The Discrete Adjoint Method for High-Order Time-Stepping Methods

by

Kai Rothauge

MMath, The University of Bath, 2007

A THESIS SUBMITTED IN PARTIAL FULFILLMENT OF
THE REQUIREMENTS FOR THE DEGREE OF

DOCTOR OF PHILOSOPHY

in

The Faculty of Graduate and Postdoctoral Studies

(Mathematics)

THE UNIVERSITY OF BRITISH COLUMBIA

(Vancouver)

December 2016

Abstract

This thesis examines the derivation and implementation of the discrete adjoint method for several time-stepping methods. Our results are important for gradient-based numerical optimization in the context of large-scale model calibration problems that are constrained by nonlinear time-dependent PDEs. To this end, we discuss finding the gradient and the action of the Hessian of the data misfit function with respect to three sets of parameters: model parameters, source parameters and the initial condition. We also discuss the closely related topic of computing the action of the sensitivity matrix on a vector, which is required when performing a sensitivity analysis. The gradient and Hessian of the data misfit function with respect to these parameters requires the derivatives of the misfit with respect to the simulated data, and we give the procedures for computing these derivatives for several data misfit functions that are of use in seismic imaging and elsewhere.

The methods we consider can be divided into two categories, linear multistep (LM) methods and Runge-Kutta (RK) methods, and several variants of these are discussed. Regular LM and RK methods can be used for ODE systems arising from the semi-discretization of general nonlinear time-dependent PDEs, whereas implicit-explicit and staggered variants can be applied when the PDE has a more specialized form. Exponential time-differencing RK methods are also discussed. The implementation of the associated adjoint time-stepping methods is discussed in detail. Our motivation is the application of the discrete adjoint method to high-order time-stepping methods, but the approach taken here does not exclude lower-order methods.

All of the algorithms have been implemented in MATLAB using an object-

oriented design and are written with extensibility in mind. For exponential RK methods it is illustrated numerically that the adjoint methods have the same order of accuracy as their corresponding forward methods, and for linear PDEs we give a simple proof that this must always be the case. The applicability of some of the methods developed here to pattern formation problems is demonstrated using the Swift-Hohenberg model.

Preface

The work presented in this thesis is original research conducted while studying at the University of British Columbia under the supervision of Dr. Eldad Haber and Dr. Uri Ascher. The inspiration came from a discussion with Dr. Haber about high-order Runge-Kutta methods in the context of full waveform inversion.

I am responsible for deriving the expressions for all time-stepping methods considered in this study, aided by discussions with my supervisors regarding the discrete adjoint method and related topics. The code was implemented from scratch by myself, and I performed all the numerical tests and conducted the numerical experiments. I am responsible for the thesis manuscript preparation, with revision support from Dr. Ascher and Dr. Haber.

Some results of this work have been submitted for publication. The application of the discrete adjoint method to general linear multistep and Runge-Kutta methods for linear PDEs is presented in [94], and the discrete adjoint method for exponential time-differencing Runge-Kutta methods, including the application to a pattern formation problem, is discussed in [93]. Dr. Ascher and Dr. Haber provided fruitful discussions and helped with editing of the manuscripts. Additional manuscripts based on this work are in preparation.

Table of Contents

Appendix

List of Tables

List of Figures

List of Acronyms

Abbreviation	Description
ODE(s)	Ordinary differential equations(s)
PDE(s)	Partial differential equations(s)
MOL	Method of lines
LM	Linear multistep
BDF(s)	Backward Differentiation Formula(s)
RK	Runge-Kutta
ERK	Explicit Runge-Kutta
DIRK	Diagonally-Implicit Runge-Kutta
IMEX	Implicit-explicit
IMEX LM/RK	Implicit-explicit linear multistep/Runge-Kutta
StagLM/RK	Staggered linear multistep/Runge-Kutta
ETD	Exponential Time-Differencing
ETDRK	Exponential Time-Differencing Runge-Kutta
DO	Discretize-then-Optimize
OD	Optimize-then-Discretize
AD	Automatic Differentiation
GN	Gauss-Newton
BFGS	Broyden-Fletcher-Goldfarb-Shanno

Continued on next page...

Abbreviation	**Description**
L-BFGS	Limited-memory BFGS
FWI	Full waveform inversion

Notation

- **General**

Symbol	Description
$\mathbf{a}$	Lowercase boldface letters denote vectors
$\mathbf{A}$	Uppercase boldface letters denote matrices
t	Time variable
T	Final integration time
t_k	kth time level
τ_k	Variable time step, $\tau_k = t_k - t_{k-1}$
τ	Constant time step
K	Number of time-steps
w	Variable that is continuous is space and time
$\mathbf{w}(t)$	Semi-discretized variable, still continuous in time
$\mathbf{w}_k$	Estimated value of $\mathbf{y}$ at t_k, $\mathbf{w}_k = \mathbf{w}(t_k)$
$\mathbf{w}_0$	Initial condition
$\overline{\mathbf{w}}$	Fully discretized variable, exact structure depends on time-stepping method
$N_{\mathbf{a}}$	Number of entries in vector $\mathbf{a}$
N	Number of entries in $\mathbf{y}_k$ (usually this is the number of spatial unknowns)
$\mathbf{y}$	Forward solution

Continued on next page...

Symbol	**Description**
$\boldsymbol{\xi}$	Solution of the linearized forward problem
$\mathbf{q}$	Source term for the forward problem
$\boldsymbol{\lambda}$	Adjoint solution
$\boldsymbol{\theta}$	Source term for the adjoint problem
$\mathbf{t}$	Time-stepping vector, abstract representation of time-stepping method
$\dfrac{\partial \mathbf{t}}{\partial \mathbf{y}}$	Derivative of the time-stepping method with respect to the solution, abstract representation of the linearized time-stepping method
$\dfrac{\partial \mathbf{t}}{\partial \mathbf{y}}^{\top}$	Derivative of the time-stepping method with respect to the solution, abstract representation of the adjoint time-stepping method
$\mathbf{m}$	Model parameters
$\mathbf{s}$	Source parameters
$\mathbf{p}$	Parameters set, $\mathbf{p} = \begin{bmatrix} \mathbf{m}^{\top} & \mathbf{y}_0^{\top} & \mathbf{s}^{\top} \end{bmatrix}^{\top}$
$\mathbf{p}^{\mathrm{ref}}$	Reference parameter set (incorporates prior information)
$\mathbf{d}$	Simulated data/measurements/observations
$\mathbf{d}^{\mathrm{obs}}$	Actual/observed data/measurements/observations
M	Misfit function, $\mathrm{M} = \mathrm{M}(\mathbf{d}, \mathbf{d}^{\mathrm{obs}})$
R	Regularization function, $\mathrm{R} = \mathrm{R}(\mathbf{p}, \mathbf{p}^{\mathrm{ref}})$
β	Regularization parameter
Ω	Objective function, $\Omega = \Omega(\mathbf{p}) = \Omega(\mathbf{d}, \mathbf{d}^{\mathrm{obs}}; \mathbf{p}, \mathbf{p}^{\mathrm{ref}}) = \mathrm{M} + \beta \mathrm{R}$
$\dfrac{\partial \mathbf{x}}{\partial \mathbf{y}}$	Jacobian/sensitivity matrix of $\mathbf{x}$ with respect to $\mathbf{y}$
$\mathbf{J}$	Sensitivity matrix $\dfrac{\partial \mathbf{d}}{\partial \mathbf{p}}$

Continued on next page...

Symbol	**Description**
$\nabla_{\mathbf{p}/\mathbf{d}}\mathrm{M}$	gradient of M with respect to $\mathbf{p}/\mathbf{d}$
$\mathcal{H}_{\mathrm{M}}$	Hessian of M with respect to $\mathbf{p}$
$\dfrac{\partial^2 \mathrm{M}}{\partial \mathbf{p} \partial \mathbf{p}}$	Hessian of M with respect to $\mathbf{p}$
$\dfrac{\partial^2 \mathrm{M}}{\partial \mathbf{d} \partial \mathbf{d}}$	Hessian of M with respect to $\mathbf{d}$
$\dfrac{\partial^2 \mathbf{x}}{\partial \mathbf{y} \partial \mathbf{z}}\mathbf{w}$	$\dfrac{\partial^2 \mathbf{x}}{\partial \mathbf{y} \partial \mathbf{z}}\mathbf{w} = \dfrac{\partial}{\partial \mathbf{y}}\left(\dfrac{\partial \mathbf{x}}{\partial \mathbf{z}}\mathbf{w}\Big\vert_{\mathbf{w}}\right)$, where $\mathbf{x}$, $\mathbf{y}$, $\mathbf{z}$ and $\mathbf{w}$ are arbitrary vectors of appropriate lengths; $\mathbf{w}$ is taken to be fixed with respect to $\mathbf{y}$
$\mathbf{v}/\mathbf{w}$	Usually arbitrary vectors of appropriate size that are multiplied with some matrix
$\mathbf{v_a}/\mathbf{w_a}$	Usually arbitrary vectors with the same length and structure as $\mathbf{a}$
$\mathrm{diag}\,(\mathbf{a})$	Diagonal matrix with the entries of $\mathbf{a}$ on its diagonal
blkdiag $(\mathbf{A}_1, \cdots, \mathbf{A}_K)$	Block diagonal matrix, with matrices $\mathbf{A}_1, \cdots, \mathbf{A}_K$ on its diagonal
$\otimes$	Kronecker product
$\odot$	Kronecker dot product
$\mathbf{F}$	Discrete Fourier transform matrix (1D or 2D, depends on context)
$\widehat{\mathbf{a}}$	Discrete Fourier transform of vector $\mathbf{a}$, $\widehat{\mathbf{a}} = \mathtt{fft}(\mathbf{a}) = \mathbf{F}\,\mathbf{a}$
$\cdot^{\top}$	Transpose operator
$\cdot^{\star}$	Hermitian operator
$\bar{\cdot}$	Conjugation operator

- **Runge-Kutta Methods**

Symbol	**Description**
s	Number of internal stages
σ	Index used to denote the current internal stage
$\mathbf{Y}_{k,\sigma}$	σth internal stage at the kth time-step if the forward solution is given by $\mathbf{y}$; internal stages are generally represented by the uppercase of the letter representing the forward solution
$\mathbf{Y}_k$	Vector containing all the internal stages at the kth time-step, e.g. $\mathbf{Y}_k = \begin{bmatrix} \mathbf{Y}_{k,1}^\top & \cdots & \mathbf{Y}_{k,s} \end{bmatrix}^\top$
$\widehat{\mathbf{y}}_k$	Vector containing all the internal stages and the computed solution at the kth time-step, i.e. $\widehat{\mathbf{y}}_k = \begin{bmatrix} \mathbf{Y}_k^\top & \mathbf{y}_k^\top \end{bmatrix}^\top$
$a_{\sigma i}$	ith Runge-Kutta coefficient for the σth stage
b_σ	σth weight
c_σ	σth node
$\mathbf{\Psi}$	Runge-Kutta transition matrix, maps one time level to the next

- **Linear Multistep Methods**

Symbol	**Description**
s	Number of internal steps
σ	Index used to denote the current step
α_j, β_j, γ_j	Weights used by various linear multistep methods
$\mathbf{S}$	Source weighting matrix

- **Regular Time-Stepping Methods**

Symbol	Description
$\mathbf{f}$	Nonlinear operator, $\mathbf{f} = \mathbf{f}(\mathbf{y}, t; \mathbf{m})$

- **Implicit-Explicit Time-Stepping Methods**

Symbol	Description
$\mathbf{f}^{\mathcal{E}}$	Nonlinear operator to be handled explicitly, $\mathbf{f}^{\mathcal{E}} = \mathbf{f}^{\mathcal{E}}(\mathbf{y}, t; \mathbf{m})$
$\mathbf{f}^{\mathcal{I}}$	Nonlinear operator to be handled implicitly, $\mathbf{f}^{\mathcal{I}} = \mathbf{f}^{\mathcal{I}}(\mathbf{y}, t; \mathbf{m})$
$\mathcal{E}$	"Explicit", used only as superscript
$\mathcal{I}$	"Implicit", used only as superscript

- **Staggered Time-Stepping Methods**

Symbol	Description
t_k	As above, but referred to as integer time level in this context
$t_{k+\frac{1}{2}}$	Half-integer time level, halfway between t_k and t_{k+1}
$\mathbf{u}$	Component of forward solution at integer time levels
$\mathbf{v}$	Component of forward solution at half-integer time levels
$\mathbf{f}^{\mathbf{u}}$	Nonlinear operator mapping from the half-integer time level to the integer time level, $\mathbf{f}^{\mathbf{u}} = \mathbf{f}^{\mathbf{u}}(\mathbf{v}, t; \mathbf{m})$
$\mathbf{f}^{\mathbf{v}}$	Nonlinear operator mapping from the integer time level to the half-integer time level, $\mathbf{f}^{\mathbf{v}} = \mathbf{f}^{\mathbf{v}}(\mathbf{u}, t; \mathbf{m})$
e	"even", used as superscript or under summation symbols
o	"odd", used as superscript or under summation symbols

- **Exponential Time-Differencing Methods**

Symbol	Description
$\mathbf{L}$	Linear operator
$\mathbf{L}_k$	Linear operator as a function of the solution at the kth time level, $\mathbf{L}_k = \mathbf{L}\left(\mathbf{y}_k\right)$
$\mathbf{n}$	nonlinear operator $\mathbf{n} = \mathbf{n}(\mathbf{y}, t; \mathbf{m})$
$\varphi_\ell(z)$	$\varphi_\ell(z) = \dfrac{\varphi_{\ell-1}(z) - \varphi_{\ell-1}(0)}{z}$, with $\ell > 0$ and $\varphi_0(z) = e^z$
$\varphi_{\ell,\sigma}$	$\varphi_{\ell,\sigma} = \varphi_\ell(c_\sigma \tau_k \mathbf{L}_{k-1})$

- **Misfit Functions**

Symbol	Description
N_R	Number of receivers/traces
$\mathbf{W}$	Weighting matrix for least-squares misfit
$\mathcal{A}_i$	"Amplitude" of trace $\mathbf{d}_i$, $\mathcal{A}_i = \lVert\mathbf{d}_i\rVert$; used by the least-squares amplitude misfit
$\mathcal{T}_i$	Optimal shift of trace $\mathbf{d}_i$; used by the cross-correlation time shift misfit
$S(t')$	Discrete shift operator, shifts trace it acts on to the right by t' grid points
$\star$	Cross-correlation operator
N_ω	Number of frequencies used by Fourier transform
$\mathbf{E}$	Matrix of 0s and 1s

Acknowledgements

I am indebted to my supervisors Eldad Haber and Uri Ascher for their patience and support while completing this thesis. Their guidance and insight have been crucial during my time at UBC, and their significant expertise in a wide range of topics has made me a much better applied mathematician.

I am grateful to Chen Greif for the support and interesting discussions, and to Dan Coombs and Lee Yupitun for their help. Thank you to Michael Ward for the engaging courses.

My time at SFU as an exhange student was a pivotal moment in my life. I am grateful to Ralf Wittenberg for showing me that research can be fun and inspiring me to go to grad school.

A huge thank you to my best friend and amazing partner Rebecca for giving me so much encouragement, love and support during this stressful time.

I am very grateful to Iain for his friendship, finding a friend like him is a rare thing. A big thank you to Erin and Kyle for their generosity, they have made my life a lot easier.

Also thank you to Fred, having him around made the final few years of grad school far more bearable, and to Mike for the many fun conversations. Thank you to Pouya and Helen for being such good friends, and to everyone else who has made my time here more enjoyable over the years.

I am especially grateful for the support of my family.

To my family

Chapter 1

Introduction

Large-scale model calibration is an important type of inverse problem dealing with the recovery or approximation of the model parameters appearing in partial differential equations (PDEs). The PDE models some real-life process and observations of this process, which usually contain noise, are compared to numerically computed solutions of the PDE. Model calibration problems are very common in science and engineering, for instance geophysics [36, 51, 87], fluid dynamics [71], computer vision [116] and bioscience [21], to name just a few. For a more general treatment of such problems we mention [9–11, 49, 66, 118].

We consider a generic time-dependent nonlinear PDE. Using the method of lines (MOL) approach, it is discretized in space by applying a finite difference, finite volume, finite element or pseudspectral method, where the known boundary conditions are assumed to have already been incorporated; we do not assume that the initial condition is known, although it can be. This results in a large system of ordinary differential equations (ODEs) that can be written in generic first-order form as

$$\begin{aligned} \frac{\partial \mathbf{y}(t;\mathbf{p})}{\partial t} &= \mathbf{f}(\mathbf{y}(t;\mathbf{p}), t, \mathbf{m}) + \mathbf{q}(t, \mathbf{s}) \\ \mathbf{y}(0) &= \mathbf{y}_0, \end{aligned} \tag{1.1}$$

on an underlying discrete spatial grid, with $0 \leq t \leq T$. The vector $\mathbf{p} = \begin{bmatrix} \mathbf{m}^\top & \mathbf{y}_0^\top & \mathbf{s}^\top \end{bmatrix}^\top$ contains the parameters that we are trying to recover, where $\mathbf{m} \in \mathbb{R}^{N_\mathbf{m}}$ is the set of discretized model parameters that could, for instance, represent some physical properties of the underlying material that we want to estimate. The source parameters

$\mathbf{s} \in \mathbb{R}^{N_\mathbf{s}}$ determine the behaviour of the source term, and we also allow for the initial condition $\mathbf{y}_0$ to be unknown[1].

We refer to $\mathbf{y} = \mathbf{y}(t; \mathbf{p})$ in this context as the *forward solution*. It is a time-dependent vector of length N and depends on $\mathbf{p}$ indirectly through the discretized operator $\mathbf{f}$ and the discretized source term $\mathbf{q}$. Since $\mathbf{y}$ arises from a spatially discretized PDE, N can be very large in practice. The length of the parameter vector, $N_\mathbf{p}$, may be similarly large, even if $\mathbf{y}_0$ is known.

Given a current estimate of $\mathbf{p}$, the solution to (1.1) is approximated by using some time-stepping method, leading to a fully discretized version of (1.1). The results of this simulation are then compared to the observations of the actual process to further improve on the estimate of $\mathbf{p}$. This is often done by employing a large optimization problem constrained, among other requirements, by the discretized PDE. Gradient-based optimization procedures are commonly employed, but computing the required gradients is generally a non-trivial matter. In this context, for large-scale problems, the *adjoint method* is a well-known and efficient approach to computing the required gradients.

The goal of this study is to systematically implement the *discrete adjoint method* to the fully discretized version of (1.1). The discrete adjoint method for time-dependent problems is highly dependent on the time-stepping method being used and our motivation is its application to several high-order time-stepping methods. This is expected to be a significant practical contribution to model calibration problems where it is desirable to solve the forward model to high accuracy.

Our results are also relevant to the problem of sensitivity analysis, where the efficient computation of the action of the sensitivity matrix on a vector is required. Sensitivity analysis is an essential tool for parameter estimation, uncertainty quantification, optimization, optimal control, and the construction of reduced order models,

[1] We note that model calibration is often referred to as *parameter estimation* if one is trying to recover just $\mathbf{m}$ or (sometimes) $\mathbf{s}$. In some sources the two terms are used interchangeably, but we will try to use the term *model calibration* when recovering $\mathbf{p}$, and *parameter estimation* when recovering only $\mathbf{m}$.

among other applications.

Below we give an outline of the time-stepping methods we discuss in this thesis. This is followed by a brief overview of PDE-constrained optimization, and after that we introduce the adjoint method. We conclude the introduction with thesis contributions and related work, then outline of the rest of the thesis.

1.1 Time-Stepping Schemes

The main objective of this thesis is to implement the discrete adjoint method for the following classes of time-stepping methods that lend themselves to high-order time integration:

- Standard linear multistep (LM) and Runge-Kutta (RK) methods are commonly used to solve a wide range of ODE systems. RK methods are multistage methods that use collocation points inbetween two time levels using internal stages, whereas LM methods save a finite time history of the solution in order to extrapolate to the next step.

- Implicit-explicit (IMEX) methods can be used in cases where $\mathbf{f}$ can be split into a component that should be evaluated explicitly and another component that would be better served by an implicit treatment. IMEX methods can be divided into linear multistep (IMEX LM) and Runge-Kutta approaches (IMEX RK).

- Staggered methods are applicable when $\mathbf{y}$ can be split into two sets, say $\mathbf{u}$ and $\mathbf{v}$, where the evolution of $\mathbf{u}$ depends solely on the current value of $\mathbf{v}$, and vice versa. There are again two approaches: linear multistep (StagLM) and Runge-Kutta (StagRK).

- Lastly, exponential time-differencing methods are applicable to problems that can be written in semilinear form. Linear multistep and Runge-Kutta approaches are

again possible, but in this case we focus solely on the Runge-Kutta (ETDRK) approach.

Applying a time-stepping method to (1.1) fully discretizes the system by breaking the time interval $[0, T]$ up into a finite set of time-steps. To utilize the discrete adjoint method, this fully discretized system is abstractly represented in the form of the *time-stepping equation*

$$\mathbf{t}(\overline{\mathbf{y}}(\mathbf{p}); \mathbf{m}, \mathbf{y}_0) = \mathbf{S}\,\overline{\mathbf{q}}(\mathbf{s}), \tag{1.2}$$

where the *time-stepping vector* $\mathbf{t}$ and the *source weighting operator* $\mathbf{S}$ depend on the time-stepping method being used. $\overline{\mathbf{y}}$ represents the fully discretized forward solution, i.e. a vector containing the numerically approximated solutions at all of the time levels, and $\overline{\mathbf{q}}$ represents the fully discretized source term.

1.2 PDE-Constrained Optimization

The model calibration problem is to recover a feasible set of parameters $\mathbf{p}$ that approximately solves a PDE-constrained optimization problem of the form

$$\mathbf{p}^\star = \arg\min_{\mathbf{p}} \Omega(\mathbf{p}) \qquad \text{s.t. (1.2) holds.} \tag{1.3}$$

The *objective function*

$$\Omega(\mathbf{p}) = \Omega(\mathbf{d}, \mathbf{d}^{\text{obs}}; \mathbf{p}, \mathbf{p}^{\text{ref}}) = \mathrm{M}(\mathbf{d}, \mathbf{d}^{\text{obs}}) + \beta\,\mathrm{R}(\mathbf{p}, \mathbf{p}^{\text{ref}})$$

has two components. The *regularization function* $\mathrm{R}(\mathbf{p}, \mathbf{p}^{\text{ref}})$ penalizes straying too far away from the prior knowledge we have of the true parameter values (which is incorporated in $\mathbf{p}^{\text{ref}}$). Common examples are Tikhonov-type regularization [35, 123], including least squares and total variation (TV) [127]. The *data misfit function*

$\mathrm{M}(\mathbf{d}, \mathbf{d}^{\mathrm{obs}})$ in some way quantifies the difference between $\mathbf{d}$ and $\mathbf{d}^{\mathrm{obs}}$, with $\mathbf{d}^{\mathrm{obs}}$ a given set of observations of the true solution and $\mathbf{d} = \mathbf{d}(\mathbf{p})$ the observation of the current simulated solution, depending on $\mathbf{p}$ implicitly through the forward solution.

The most popular choice for M, corresponding to the assumption that the noise in the data is simple and white, is the least-squares function

$$\mathrm{M} = \frac{1}{2} \left\| \mathbf{d} - \mathbf{d}^{\mathrm{obs}} \right\|_2^2 .$$

However, we will not restrict our discussion to any particular misfit function. The relative importance of the data misfit and prior is adjusted using the regularization parameter β.

There are several classes of optimization procedures that can be used to solve (1.3), see for instance [35, 123, 127]. Often it is possible to reformulate (1.3) as an unconstrained optimization problem (this is the *reduced space* approach for PDE-constrained optimization), and then minimized using the gradient of Ω with respect to $\mathbf{p}$ [31, 86],

$$\nabla_{\mathbf{p}} \Omega = \nabla_{\mathbf{p}} \mathrm{M} + \beta \nabla_{\mathbf{p}} \mathrm{R}.$$

Popular gradient-based minimization procedures include steepest-descent, nonlinear conjugate-gradient and BFGS. Newton's method is a gradient-based procedure that requires the availability of the Hessian, and first-order approximations of the Hessian lead to Gauss-Newton and Levenberg-Marquardt methods.

The derivatives of the regularization function are known and are independent of the time-stepping scheme, so in this thesis we focus on M. The adjoint method is used to find expressions for both the gradient and the action of the Hessian of M.

1.3 The Adjoint Method

The sensitivity matrix $\mathbf{J} = \frac{\partial \mathbf{d}}{\partial \mathbf{p}}$ stores the first-order derivatives of the predicted data with respect to the model parameters and can be calculated explicitly in small-scale applications. However, in cases such as those considered here it is far more feasible to compute the action of the sensitivity matrix on a vector using the adjoint method. Originally developed in the optimal control community, Chavent [19] introduced the adjoint method to the theory of inverse problems to efficiently compute the gradient of a function. See [90] for a review of the method applied to geophysical problems. As we will discuss in Chapter 3, the adjoint method requires the solution of an *adjoint problem*, which will have to be integrated backward in time and requires the availability of the forward numerical solution.

There are two frameworks that one can take when computing derivatives of the misfit function, discretize-then-optimize (DO) or optimize-then-discretize (OD). In OD one forms the adjoint differential problem, for either the given PDE or its semi-discretized form (1.1), which is subsequently discretized, thus decoupling the discretization process of the problem from that of its adjoint. A disadvantage of this approach is that the gradients of the discretized problem are not the (exact) gradients of any function, because a discretization error gets introduced when moving from the continuous setting [49]. However, this approach offers flexibility in its implementation, since the numerical method and the sequence of step sizes used to solve the adjoint problem may differ from those used to solve the forward problem, and may be tuned separately to satisfy the accuracy needs of the reverse integration. This comes at the cost of having to interpolate the forward solution to the time levels required in the integration of the adjoint problem.

We therefore prefer the DO approach, where one applies the adjoint method to the fully discretized form (1.2). This provides the exact gradient (within roundoff error) of the numerical data misfit function, which can be important in numerical optimization problems. On the other hand, there is no flexibility to separately tune the adjoint

integration since the sequence of step sizes are determined by the forward numerical method. The advantage of this, of course, is that all the variables needed to form the discrete adjoint solution procedure are computed during the forward solution, and no additional interpolations are necessary. Needless to say, the time-stepping method is of central importance, which is the motivation for the work done in this thesis.

An alternative approach for computing the gradient that falls inside the DO framework is automatic differentiation (AD) [48, 85], where the exact derivatives of M with respect to $\mathbf{p}$, up to rounding error, are computed automatically by following the sequence of arithmetic commands used in the computation of M and successively applying basic differentiation rules, particularly the chain rule, to determine the dependence of M on $\mathbf{p}$.

AD is a great tool for many purposes, in particular because it can be easy to use if one is already familiar with an AD implementation, and it allows one to focus on the results rather than the differentiation procedure. However, it can also be difficult to use if the practitioner is not already familiar with it, especially if sophisticated features are required, and the generated output may be hard to follow, leading a loss of understanding of the mathematical structure.

For large-scale problems, where efficiency and memory allocation are essential, the adjoint method approach should be preferred since hard-coded and optimizable computations can lead to a significant performance increase. For instance, having an explicit algorithm for the adjoint time-stepping method, as we develop here, allows one to exploit opportunities to increase the computational efficiency, such as parallelization and the precomputation of repeated quantities.

Furthermore, it would require a very sophisticated AD code to accomplish some of the tasks considered here, since most AD codes would have difficulties in handling black-box routines, linear solvers, etc. without supervision by the user. In particular, the derivatives of the φ-functions that arise in ETD schemes, further elaborated upon in Sections 2.4.1 and 2.4.2, would present a significant challenge. Finally, when using

AD the derivatives need to be computed for each PDE and time-stepping method used to solve it, possibly for different platforms and time-stepping libraries. The discrete adjoint method is much more flexible in this regard, since the adjoint time-stepping methods and related expressions need to be derived just once for each family of time-stepping method.

For these reasons we do not consider AD any further, but see [47] for a discussion on AD vs. the continuous adjoint method, and see [95] for a joint adjoint-AD implementation to compute Hessian matrices. [99] discusses AD for the sensitivity analysis of ODE-constrained problems.

1.4 Thesis Contributions and Related Work

The main objective of this thesis is the systematic implementation of the discrete adjoint method for a variety of popular time-stepping methods used for high-order time integration. There are numerous publications that discuss the methodology behind adjoint-based gradient computation, but these are usually for time-independent problems and the details on the implementation are frequently omitted. In contrast, our work is done with the numerical implementation of the resulting expressions in mind. Therefore we present several algorithms throughout the text to assist the interested reader in the implementation of our results.

For each time-stepping method, the adjoint method requires the derivatives of (1.2) with respect to $\mathbf{p}$ and $\overline{\mathbf{y}}$, as well as the transposes of these derivative matrices. One of the main contributions of this thesis is to find the expressions for these terms. This is presented in such a way that keeps the various time-stepping methods distinct from each other, so that researchers interested in one particular scheme can easily find the relevant expressions without having to go through extra layers of discussion that are not relevant to the desired scheme.

Model calibration involving the simultaneous recovery of three distinct sets of

parameters (model parameters $\mathbf{m}$, source parameters $\mathbf{s}$ and the initial condition $\mathbf{y}_0$) appears to be a little-explored field and will be useful in applications where some or all of these parameters are poorly known. Our main contribution here is finding an expression for the action of the Hessian of M on some arbitrary vector, again using the adjoint method, where these distinct sets of parameters lead to cross-terms that complicate the derivation.

For most of the thesis we will allow M to be some generic data misfit function. As we will see, the derivatives of M with respect to $\mathbf{p}$ require the derivatives of M with respect to the simulated data $\mathbf{d}$ (i.e. observations of the forward solution). In most applications the ℓ_2 data misfit discussed above is sufficient and in this case finding the derivatives of M with respect to $\mathbf{d}$ is trivial, but there a some applications, in particular full waveform inversion (FWI) [36] or seismic tomography, where other, more specialized, misfit functions might be more sensible. While the derivatives of some of these misfit functions in the continuous setting are known, finding both the first and second derivatives in a discrete setting is novel as far as we know. In particular, finding the derivatives of the interferometric misfit is new to the best of our knowledge.

We apply some of our results to the problem of parameter estimation in pattern formation. The model parameters are allowed to vary in space, leading to different patterns forming in different regions. The number of model parameters in this case is large and the application of adjoint-based optimization procedures to this particular problem appears to be novel.

Applying the discrete adjoint method to certain time-stepping methods has of course been done before, particularly in the area of optimal control. See for instance [4, 58, 83] for applications to the Crank-Nicolson method. Adjoint methods for RK methods have already been tackled in several publications. In [32] a second order RK method for optimal control problems with ODE constraints is developed, and Hager investigated the order conditions for adjoint RK methods up to order four in [52], but

the discussion is somewhat inaccessible to practitioners working outside of the field of optimal control. [100] tackles the consistency of the methods found in [52], and see [3] for a discussion on the adjoint RK method in combination with adaptive spatial grids. The adjoint method is applied to Rosenbrock time-stepping methods (which are closely related to implicit RK methods) in [28]. The discrete adjoint method was used by Zhang [131] for a wide range of RK methods to perform sensitivity analysis with respect to model parameters and the initial condition.

Properties of adjoint RK methods have also been investigated in the context of optimal control in, for example, [12, 60, 69]. Sanz-Serna has recently [107] discussed the adjoint method in conjunction with symplectic RK methods in this context. The standard approach to developing adjoint RK schemes in the optimal control community appears to be using AD, as described in [128].

The discrete adjoint method for LM methods has been addressed as well, see [101] for the derivation and some analysis of adjoint LM methods. In this case the adjoint methods are obtained using a variational calculus approach, which differs from the approach taken in this thesis, although it leads to the same expressions in the case of constant step sizes. Note that variable step sizes are allowed in [101] (in contrast to the work in this thesis, where only constant step sizes are considered for LM methods) and it is shown that having variable step sizes may lead to the adjoint LM methods that are inconsistent. The consistency and stability properties of both adjoint LM and RK methods are discussed by Sandu in [102]. See also [2] for the adjoint method in conjunction with variable time-step integrators.

As will be shown in Chapter 3, the adjoint method can also be used to obtain expressions for second-order derivatives (often called second-order adjoints in the literature), and some work on this has already been done for LM and RK methods, see in particular [22] and [106].

Aside from optimal control problems, the work listed above has been applied to data assimilation problems [23–25], PDE-constrained optimization problems [22] and

a-posteriori error estimates [92]. See also the references in these papers, and the examples in the papers listed in the paragraphs above, for other applications.

In this thesis we re-derive the adjoint LM and RK methods. While the expressions we obtain do not differ significantly from those in the references above, our approach is geared towards the practical implementation of these methods and we give a significant number of additional details that will be of relevance when computing the gradient and Hessian of M. Re-deriving the adjoint methods of regular LM and RK methods also helps the reader in understanding the derivation of the more complicated IMEX, staggered and exponential time-stepping methods that are also tackled.

The work presented for these latter methods appears to be novel, although we should mention that the leap-frog method is a lower-order staggered time-stepping method that has been used in numerous implementations for parameter estimation in time-domain seismic imaging, electromagnetic inversion, and elsewhere (see also [115] for a discussion of the adjoint leapfrog method in meteorology). Our discussion on staggered time-stepping methods is, however, more general and geared towards higher-order methods.

There are only a few software packages available for the solution of ODEs have the capability to compute sensitivities for large-scale problems. These include `ODESSA` [75] and `CVODES` within `SUNDIALS` [110] from Lawrence Livermore National Laboratory, which are geared towards sensitivity analysis for problems solved using backward differentiation formulas (BDFs). There are also several packages available for various RK methods from the group of Adrian Sandu, such as `KPP` [30, 103, 105], `MatlODE` [104], `DENSERKS` [1], and, more recently, `FATODE` [132–134], which can handle many different types of regular RK methods, including implicit and Rosenbrock methods. These packages have been used in 3D PDE-constrained optimization solvers in atmospheric data assimilation, in particular NASA's GEOS-Chem [113, 114] and Environmental Protection Agency's CMAQ [45, 57].

None of the packages above can handle general IMEX, staggered or exponential time integration methods. The major contribution of this thesis is that it provides the background needed for the implementation of the sensitivity computations, as well as gradient and Hessian computations, not only for the time-stepping methods implemented by the packages listed above, but many more.

The strength of the approach taken in this study is that it is not limited to a specific time-stepping method since we find general formulations for each class of time-stepping method. This is in contrast to AD, where the adjoint expressions would have to be found for each specific time-stepping method. While our motivation is large-scale model calibration, our results can be applied to any application where the goal is to solve a problem in the form of (1.3) using gradient-based minimization procedures.

To avoid potential confusion, we mention at this point that it appears that in the computational fluid dynamics community, especially in regard to shape optimization, the term *discrete adjoint method* is used when working with steady-state (i.e. time-independent) problems or, more relevant to our discussion, semidiscrete problems in the form (1.1). A time-stepping method is then applied to this semidiscrete adjoint formulation. In our discussion we consider the adjoint method to be discrete only if it is applied to the fully discretized problem (1.2).

1.5 Thesis Overview and Outline

This thesis has nine chapters. After this introductory chapter, we review the time-stepping methods of interest in Chapter 2 and show how each of these can be written in the form of (1.2).

Expressions for the sensitivity matrix, as well as the gradient and the action of the Hessian of M are derived in Chapter 3 using the discrete adjoint method. The derivations of the Hessian are included in Appendix B.

The adjoint method requires the derivatives of $\mathbf{t}$ with respect to $\mathbf{p}$ and $\overline{\mathbf{y}}$. Finding expressions for these derivatives is an important contribution of this thesis, but due to the very technical nature of these derivations we have omitted them from the main text and instead included them in Appendix A. For ease of reference, Chapter 4 lists the page and equation numbers of the final expressions of the derivatives.

The computation of the action of the sensitivity matrix requires the solution of the linearized forward problem $\frac{\partial \mathbf{t}}{\partial \overline{\mathbf{y}}}\boldsymbol{\xi} = \mathbf{q}$, which is the focus of Chapter 5. The adjoint method requires the adjoint solution $\overline{\boldsymbol{\lambda}}$, which is the solution to the adjoint problem $\frac{\partial \mathbf{t}}{\partial \overline{\mathbf{y}}}^{\top} \boldsymbol{\lambda} = \boldsymbol{\theta}$, and this is discussed in detail in Chapter 6.

Chapter 7 presents the derivatives of several misfit functions that might be of use especially in the context of seismic tomography. Some of our results concerning exponential time-differencing methods are used in Section 8 to tackle a parameter estimation problem involving the Swift-Hohenberg model, a PDE problem with solutions exhibiting the interesting phenomenon of pattern formation. Some final thoughts and avenues for future work are provided in Chapter 9.

Chapter 2

Review of Time-Stepping Methods

In this chapter we review the time-stepping methods considered in this thesis, where each class of time-stepping methods can be approached using either an LM-type or an RK-type approach.

The time interval $0 \leq t \leq T$ is discretized by $0 = t_0 < t_1 < \cdots < t_K = T$, with time-step sizes $\tau_k = t_k - t_{k-1}$. When using an LM-type scheme we restrict ourselves to uniform time-steps $\tau = \tau_k$, whereas for RK-type schemes we allow variable time-steps (except for staggered schemes). We let $\mathbf{y}_k \approx \mathbf{y}(t_k)$ be the numerical approximation of the true solution at the kth time-step, and $\overline{\mathbf{y}}$ represents the fully-discretized solution containing the solutions at all of the time levels. In the case of RK-type methods we will additionally include the internal stages computed at each time level in $\overline{\mathbf{y}}$ for reasons that will become apparent in later chapters.

As discussed in the introduction, it is also shown how to abstractly represent each time-stepping method by the time-stepping equation (1.2)

$$\mathbf{t}(\overline{\mathbf{y}}; \mathbf{m}, \mathbf{y}_0) = \mathbf{S}\overline{\mathbf{q}}(\mathbf{s}).$$

Representing a time-stepping scheme in this manner is a crucial tool when used in conjunction with the discrete adjoint method, as we will see in Chapter 3, but we emphasize that $\mathbf{t}$ is purely conceptual and is never formed in practice.

For notational simplicity we suppress the dependence of all quantities on the model and source parameters $\mathbf{m}$ and $\mathbf{s}$ in this chapter. It is important for later chapters to be aware of the fact that $\mathbf{y}$ depends on $\mathbf{m}$ indirectly through $\mathbf{t}$ and on $\mathbf{s}$ indirectly through $\mathbf{q}$.

To aid the readability in what follows, we introduce the following $KN \times KN$ block template matrix that will be required by LM-type methods in the sections below:

$$\times = \times^{(s)} \otimes \mathbf{I}_N, \tag{2.1a}$$

where $\times^{(s)}$ is the placeholder of a $K \times K$ matrix of the form

$$\times^{(s)} = \begin{bmatrix} \boxed{\times^{(1:s)}} & & & & & & & & \\ \times_{s-1} & \cdots & \times_1 & \times_0 & & & & & \\ \times_s & \times_{s-1} & \cdots & \times_1 & \times_0 & & & & \\ & \ddots & \ddots & & \ddots & \ddots & & & \\ & & \times_s & \times_{s-1} & \cdots & \times_1 & \times_0 & & \\ & & & \ddots & \ddots & & \ddots & \ddots & \\ & & & & \times_s & \times_{s-1} & \cdots & \times_1 & \times_0 \end{bmatrix}. \tag{2.1b}$$

The *initialization block* $\times^{(1:s)}$ is needed to handle the first few steps of the LM method. We review several options for this in Section 2.1.1 and leave the structure of $\times^{(1:s)}$ undefined for now.

2.1 Regular Time-Stepping Methods

By "regular" time-stepping methods we mean the usual LM and RK methods that are used to compute the solution of a large number of ODEs in practice. These methods are described in numerous works on the numerical solution of ODEs, standard references include Gear [39]; Lambert [73, 74]; Butcher [15]; Hairer, Nørsett and Wanner [54]; Hairer and Wanner [56]; and Ascher and Petzold [6]. See also [5, 16, 77] for more recent treatments.

2.1.1 Linear Multistep Methods

Given a linear ODE system in the form (1.1), the general s-step method, with $s \leq k \leq K$, is written as

$$\sum_{j=0}^{s} \alpha_j^{(s)} \mathbf{y}_{k-j} = \tau \sum_{j=0}^{s} \beta_j^{(s)} \mathbf{f}_{k-j} + \tau \sum_{j=0}^{s} \beta_j^{(s)} \mathbf{q}(t_{k-j}), \tag{2.2}$$

with $\mathbf{f}_{k-j} = \mathbf{f}\left(\mathbf{y}_{k-j}, t_{k-j}\right)$. The weights $\alpha_j^{(s)}$ and $\beta_j^{(s)}$ determine the s-step method, with $\alpha_0^{(s)} = 1$ (always). There are two main classes of multistep methods in active use:

- *Backward differentiation formulas* (BDFs), for which $\beta_1^{(s)} = \ldots = \beta_s^{(s)} = 0$, and
- *Adams*-type methods, for which $\alpha_1 = -1$ and $\alpha_2 = \ldots = \alpha_s = 0$. Further, *Adams-Bashforth* methods are explicit ($\beta_0^{(s)} = 0$) and *Adams-Moulton* methods are implicit ($\beta_0^{(s)} \neq 0$).

By Dahlquist's first barrier theorem [29, 54], an explicit s-step multistep method cannot attain an order of accuracy greater than s, and an implicit method cannot attain an order greater than $s+1$ if s is odd and greater than $s+2$ if s is even.

An s-step method of course needs to have the solution at s previous time steps available, which is not the case at the start of the integration when $k < s$. There are various options to handle this:

- Specify $s-1$ additional initial conditions, for instance in addition to $\mathbf{y}_0$ also specify $\mathbf{y}_{-1}$ if $s = 2$, $\mathbf{y}_{-1}$ and $\mathbf{y}_{-2}$ if $s = 3$, etc. This is the most sensible option if $\mathbf{y}_0 = \mathbf{0}$.
- If the same order of accuracy needs to be maintained, use a Runge-Kutta method of the same order for first $s-1$ iterations, keeping the same time step τ if possible.
- Use lower-order LM methods to gradually build up a solution for the first $s-1$ time steps. One can use methods with increasingly higher orders of accuracy, for

instance employ a first-order method to integrate up to τ, then a second-order method to integrate up to 2τ (using the previously computed solution at τ), etc. If necessary, the lower-order schemes can have time steps that are smaller than τ, thereby ensuring that the solution at $k\tau$ is sufficiently accurate.

Using the block template matrix (2.1), $\mathbf{A}^{(s)}$ and $\mathbf{A}$ are simply $\times^{(s)}$, $\times^{(1:s)}$ and $\times$ when we set $\times_j = \alpha_j^{(s)}$, and likewise setting $\times_j = \beta_j^{(s)}$ gives $\mathbf{B}^{(s)}$ and $\mathbf{B}$. The *initialization blocks* $\mathbf{A}^{(1:s)}$ and $\mathbf{B}^{(1:s)}$ in $\mathbf{A}^s$ and $\mathbf{B}^s$ allow for one of the three approaches mentioned above to handle the initialization. The first approach just requires that the blocks have the same diagonal and subdiagonals as $\times^{(s)}$ and the Runge-Kutta approach can be implemented using the discussion in the next section. We focus on the third approach here, but for simplicity assume that each of the lower-order methods uses the same time step τ as the higher-order method. In this case the initialization blocks are

$$\times^{(1:s)} = \begin{bmatrix} \times_0^{(1)} & & & \\ \times_1^{(2)} & \times_0^{(2)} & & \\ \vdots & \ddots & \ddots & \\ \times_{s-2}^{(s-1)} & \cdots & \times_1^{(s-1)} & \times_0^{(s-1)} \end{bmatrix}, \tag{2.3}$$

with the superscripts (σ) indicating the highest order of accuracy that can be attained at the time step k for $0 < k < s$, with $\alpha_j^{(\sigma)}$ and $\beta_j^{(\sigma)}$ being the corresponding weights of the σ-step method.

We also define the $KN \times N$ block matrices

$$\begin{aligned} \boldsymbol{\alpha}^\top &= \begin{bmatrix} \alpha_1^{(1)} & \alpha_2^{(2)} & \cdots & \alpha_s^{(s)} & \mathbf{0}_{1\times(K-s)} \end{bmatrix} \otimes \mathbf{I}_N \\ \boldsymbol{\beta}^\top &= \begin{bmatrix} \beta_1^{(1)} & \beta_2^{(2)} & \cdots & \beta_s^{(s)} & \mathbf{0}_{1\times(K-s)} \end{bmatrix} \otimes \mathbf{I}_N. \end{aligned} \tag{2.4}$$

The blocks $\mathbf{A}^{(1:s)}$ and $\mathbf{B}^{(1:s)}$, block matrices $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$, solution $\mathbf{y}$ and source term $\mathbf{q}$ can be modified in an obvious way if lower-order methods at the start of the

integration use time steps that are smaller than τ. We will not consider this more general formulation here due to space constraints.

The time-stepping system (2.2) can then be compactly represented by

$$\mathbf{A}\,\overline{\mathbf{y}} = \tau\,\mathbf{B}\,\overline{\mathbf{f}}\,(\overline{\mathbf{y}}) + \tau\,\boldsymbol{\beta}\,\mathbf{f}_0(\mathbf{y}_0) - \boldsymbol{\alpha}\,\mathbf{y}_0 + \tau\,\begin{bmatrix}\boldsymbol{\beta} & \mathbf{B}\end{bmatrix}\overline{\mathbf{q}}, \tag{2.5}$$

with

$$\begin{aligned} \overline{\mathbf{y}} &= \begin{bmatrix}\mathbf{y}_1^\top & \cdots & \mathbf{y}_K^\top\end{bmatrix}^\top, \qquad \overline{\mathbf{q}} = \begin{bmatrix}\mathbf{q}_0^\top & \mathbf{q}_1^\top & \cdots & \mathbf{q}_K^\top\end{bmatrix}^\top, \\ \overline{\mathbf{f}}\,(\overline{\mathbf{y}}) &= \begin{bmatrix}\mathbf{f}_1^\top & \cdots & \mathbf{f}_K^\top\end{bmatrix}^\top. \end{aligned} \tag{2.6}$$

(2.5) is then obtained by letting

$$\mathbf{t}(\overline{\mathbf{y}}, \mathbf{y}_0) = \mathbf{A}\,\overline{\mathbf{y}} - \tau\,\mathbf{B}\,\overline{\mathbf{f}}\,(\overline{\mathbf{y}}) + \boldsymbol{\alpha}\,\mathbf{y}_0 - \tau\,\boldsymbol{\beta}\,\mathbf{f}_0(\mathbf{y}_0) \tag{2.7}$$

and $\mathbf{S} = \tau\,\begin{bmatrix}\boldsymbol{\beta} & \mathbf{B}\end{bmatrix}$. Notice $\mathbf{t}(\overline{\mathbf{y}}, \mathbf{y}_0)$ is a vector of length KN and the kth time-step is given by the kth subvector of length N:

$$\mathbf{t}_k(\mathbf{y}, \mathbf{y}_0) = \sum_{j=0}^{s} \alpha_j^{(s)}\,\mathbf{y}_{k-j} - \tau \sum_{j=0}^{s} \beta_j^{(s)}\,\mathbf{f}_{k-j}. \tag{2.8}$$

Examples

The s-step Adams-Bashforth methods with $s = 1, 2, 3$ are

$$\begin{aligned} s = 1: &\quad \mathbf{y}_k = \mathbf{y}_{k-1} + \tau\,\mathbf{f}_{k-1} + \mathbf{q}_k \\ s = 2: &\quad \mathbf{y}_k = \mathbf{y}_{k-1} + \tau\left(\frac{3}{2}\mathbf{f}_{k-1} - \frac{1}{2}\mathbf{f}_{k-2}\right) + \mathbf{q}_k \\ s = 3: &\quad \mathbf{y}_k = \mathbf{y}_{k-1} + \tau\left(\frac{23}{12}\mathbf{f}_{k-1} - \frac{4}{3}\mathbf{f}_{k-2} + \frac{5}{12}\mathbf{f}_{k-3}\right) + \mathbf{q}_k, \end{aligned}$$

where

$$\begin{aligned}
\mathbf{q}_k &= \tau\,\mathbf{q}(t_{k-1}) \\
\mathbf{q}_k &= \frac{\tau}{2}\left(3\mathbf{q}(t_{k-1}) - \mathbf{q}(t_{k-2})\right) \\
\mathbf{q}_k &= \frac{\tau}{12}\left(23\,\mathbf{q}(t_{k-1}) - 16\,\mathbf{q}(t_{k-2}) + 5\,\mathbf{q}(t_{k-3})\right),
\end{aligned}$$

so if using a third order Adams-Bashforth method we have

$$\mathbf{A} = \begin{bmatrix} 1 & & & & & \\ -1 & 1 & & & & \\ & -1 & 1 & & & \\ & & -1 & 1 & & \\ & & & \ddots & & \\ & & & & -1 & 1 \end{bmatrix} \otimes \mathbf{I}_N, \qquad \boldsymbol{\alpha} = \begin{bmatrix} -1 \\ 0 \\ 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix} \otimes \mathbf{I}_N,$$

$$\mathbf{B} = \begin{bmatrix} 0 & & & & & \\ \frac{3}{2} & 0 & & & & \\ -\frac{4}{3} & \frac{23}{12} & 0 & & & \\ \frac{5}{12} & -\frac{4}{3} & \frac{23}{12} & 0 & & \\ & & & \ddots & & \\ & & \frac{5}{12} & -\frac{4}{3} & \frac{23}{12} & 0 \end{bmatrix} \otimes \mathbf{I}_N, \qquad \boldsymbol{\beta} = \begin{bmatrix} 1 \\ -\frac{1}{2} \\ \frac{5}{12} \\ 0 \\ \vdots \\ 0 \end{bmatrix} \otimes \mathbf{I}_N,$$

with $\mathbf{A}$ and $\mathbf{B}$ being matrices of size $KN \times KN$.

The s-step BDFs of orders $s = 1, 2, 3$ are

$$\begin{aligned}
s = 1: &\qquad \mathbf{y}_k - \mathbf{y}_{k-1} = \tau\,\mathbf{f}_k + \tau\,\mathbf{q}(t_k) \\
s = 2: &\qquad \mathbf{y}_k - \frac{4}{3}\mathbf{y}_{k-1} + \frac{1}{3}\mathbf{y}_{k-2} = \tau\frac{2}{3}\mathbf{f}_k + \tau\frac{2}{3}\mathbf{q}(t_k) \\
s = 3: &\qquad \mathbf{y}_k - \frac{18}{11}\mathbf{y}_{k-1} + \frac{9}{11}\mathbf{y}_{k-2} - \frac{2}{11}\mathbf{y}_{k-3} = \tau\frac{6}{11}\mathbf{f}_k + \tau\frac{6}{11}\mathbf{q}(t_k),
\end{aligned}$$

Therefore a third-order BDF can be represented in matrix form using

$$\mathbf{A} = \begin{bmatrix} 1 & & & & & & \\ -\frac{4}{3} & 1 & & & & & \\ \frac{9}{11} & -\frac{18}{11} & 1 & & & & \\ -\frac{2}{11} & \frac{9}{11} & -\frac{18}{11} & 1 & & & \\ & & & \ddots & & & \\ & & -\frac{2}{11} & \frac{9}{11} & -\frac{18}{11} & 1 \end{bmatrix} \otimes \mathbf{I}_N, \qquad \boldsymbol{\alpha} = \begin{bmatrix} -1 \\ \frac{1}{3} \\ -\frac{2}{11} \\ 0 \\ \vdots \\ 0 \end{bmatrix} \otimes \mathbf{I}_N$$

$$\mathbf{B} = \begin{bmatrix} 1 & & & & & \\ & \frac{2}{3} & & & & \\ & & \frac{6}{11} & & & \\ & & & \frac{6}{11} & & \\ & & & & \ddots & \\ & & & & & \frac{6}{11} \end{bmatrix} \otimes \mathbf{I}_N \qquad \boldsymbol{\beta} = \mathbf{0}_{K\times 1} \otimes \mathbf{I}_N.$$

Again, $\mathbf{A}$ and $\mathbf{B}$ are matrices of size $KN \times KN$.

2.1.2 Runge-Kutta Methods

The family of s-stage Runge-Kutta methods is given by

$$\mathbf{y}_k = \mathbf{y}_{k-1} + \tau_k \sum_{\sigma=1}^{s} b_\sigma \mathbf{Y}_{k,\sigma}, \tag{2.9a}$$

where the internal stages are, for $\sigma = 1, \cdots, s$,

$$\mathbf{Y}_{k,\sigma} = \mathbf{f}\left(\mathbf{y}_{k-1} + \tau_k \sum_{i=1}^{s} a_{\sigma i} \mathbf{Y}_{k,i}, t_{k-1} + c_\sigma \tau_k\right) + \mathbf{q}(t_{k-1} + c_\sigma \tau_k). \tag{2.9b}$$

The procedure starts from a known initial value $\mathbf{y}_0$. Here a particular method is determined by setting the number of stages and the corresponding coefficients $a_{\sigma i}$,

weights b_σ and nodes c_σ, which are commonly summarized in a *Butcher tableau*:

$$\begin{array}{c|cccc} c_1 & a_{11} & a_{12} & \cdots & a_{1s} \\ c_2 & a_{21} & a_{22} & \cdots & a_{2s} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ c_s & a_{s1} & a_{s2} & \cdots & a_{ss} \\ \hline & b_1 & b_2 & \cdots & b_s \end{array} \quad = \quad \begin{array}{c|c} \mathbf{c}_s & \mathbf{A}_s \\ \hline & \mathbf{b}_s^\top \end{array}.$$

The method is said to be *explicit* if $a_{\sigma i} = 0$ for $\sigma \leq i$ and *diagonally-implicit* if $a_{\sigma i} = 0$ for $\sigma < i$. It is *implicit* otherwise.

We mention that for stiff problems, RK methods are susceptible to the problem of *order reduction*, where the order of accuracy of the method can be reduced significantly.

For ease of notation, define $\underline{\mathbf{y}}_{k,\sigma} := \mathbf{y}_{k-1} + \tau_k \sum_{i=1}^{s} a_{\sigma i} \mathbf{Y}_{k,i}$. The method given by (2.9a) and (2.9b) can be written in matrix form as

$$\begin{bmatrix} \mathbf{I}_N & & \\ & \mathbf{I}_{sN} & \\ -\mathbf{I}_N & \mathbf{B}^\top & \mathbf{I}_N \end{bmatrix} \begin{bmatrix} \mathbf{y}_{k-1} \\ \mathbf{Y}_k \\ \mathbf{y}_k \end{bmatrix} - \begin{bmatrix} \mathbf{0}_{N\times 1} \\ \mathbf{F}_k \\ \mathbf{0}_{N\times 1} \end{bmatrix} = \begin{bmatrix} \mathbf{y}_{k-1} \\ \mathbf{Q}_k \\ \mathbf{0}_{N\times 1} \end{bmatrix}, \tag{2.10}$$

with $\mathbf{B} = -\tau \mathbf{b}_s \otimes \mathbf{I}_N$,

$$\mathbf{Y}_k = \begin{bmatrix} \mathbf{Y}_{k,1} \\ \vdots \\ \mathbf{Y}_{k,s} \end{bmatrix}, \quad \mathbf{F}_k = \begin{bmatrix} \mathbf{F}_{k,1} \\ \vdots \\ \mathbf{F}_{k,s} \end{bmatrix} \quad \text{and} \quad \mathbf{Q}_k = \begin{bmatrix} \mathbf{Q}_{k,1} \\ \vdots \\ \mathbf{Q}_{k,s} \end{bmatrix}, \tag{2.11}$$

where

$$\mathbf{F}_{k,\sigma} = \mathbf{F}_{k,\sigma}\left(\overline{\mathbf{y}}, \mathbf{y}_0\right) = \mathbf{f}\left(\underline{\mathbf{y}}_{k,\sigma}, t_{k-1} + c_\sigma \tau_k\right)$$
$$\mathbf{Q}_{k,\sigma} = \mathbf{q}\left(\underline{\mathbf{y}}_{k,\sigma}, t_{k-1} + c_\sigma \tau_k\right).$$

The system (2.10) constitutes a single time step in the solution procedure, so the Runge-Kutta procedure as a whole can be represented by (1.2) by letting $\mathbf{S} = \mathbf{I}_{(s+1)KN}$ and

$$\mathbf{t}(\mathbf{y}, \mathbf{y}_0) = \mathbf{T}\overline{\mathbf{y}} - \overline{\mathbf{f}}, \tag{2.12}$$

with

$$\overline{\mathbf{y}} = \begin{bmatrix} \mathbf{Y}_1^\top & \mathbf{y}_1^\top & \cdots & \mathbf{Y}_K^\top & \mathbf{y}_K^\top \end{bmatrix}^\top \tag{2.13}$$

$$\overline{\mathbf{f}} = \begin{bmatrix} \mathbf{F}_1^\top & \mathbf{y}_0 & \mathbf{F}_2^\top & 0_{1\times N} & \cdots & \mathbf{F}_K^\top & \mathbf{0}_{1\times N} \end{bmatrix}^\top \tag{2.14}$$

$$\overline{\mathbf{q}} = \begin{bmatrix} \mathbf{Q}_1^\top & \mathbf{0}_{1\times N} & \cdots & \mathbf{Q}_K^\top & \mathbf{0}_{1\times N} \end{bmatrix}^\top \tag{2.15}$$

$$\mathbf{T} = \begin{bmatrix} \mathbf{I}_{sN} & & & & & & \\ \mathbf{B}^\top & \mathbf{I}_N & & & & & \\ & & \mathbf{I}_{sN} & & & & \\ & -\mathbf{I}_N & \mathbf{B}^\top & \mathbf{I}_N & & & \\ & & \ddots & \ddots & \ddots & & \\ & & & & & \mathbf{I}_{sN} & \\ & & & & -\mathbf{I}_N & \mathbf{B}^\top & \mathbf{I}_N \end{bmatrix}. \tag{2.16}$$

Here $\overline{\mathbf{y}}, \overline{\mathbf{q}}, \overline{\mathbf{f}} \in \mathbb{R}^{(s+1)KN}$ and $\mathbf{T} \in \mathbb{R}^{(s+1)KN\times(s+1)KN}$.

Notice that we have included the internal stages (2.9b) in $\overline{\mathbf{y}}$. We do this because they are required when computing derivatives, specifically when we compute the derivatives of $\mathbf{t}$, which leads to a significant increase in storage overhead for high-order Runge-Kutta schemes. This can be mitigated by the use of checkpointing, where the solution is stored only for some time steps and the solution at other time steps is then recomputed using these stored solutions as needed, although note that this effectively means that the forward solution has to be calculated twice for every adjoint computation.

Examples

The most popular fourth-order Runge-Kutta (RK4) method is given by

$$\mathbf{A}_s = \begin{bmatrix} 0 & & & \\ \frac{1}{2} & 0 & & \\ & \frac{1}{2} & 0 & \\ & & 1 & 0 \end{bmatrix}, \qquad \mathbf{b}_s = \begin{bmatrix} \frac{1}{6} \\ \frac{1}{3} \\ \frac{1}{3} \\ \frac{1}{6} \end{bmatrix} \quad \text{and} \quad \mathbf{c}_s = \begin{bmatrix} 0 \\ \frac{1}{2} \\ \frac{1}{2} \\ 1 \end{bmatrix},$$

so that at some time step k we have

$$\mathbf{B} = -\tau_k \mathbf{b}_4 \otimes \mathbf{I}_N = -\tau_k \begin{bmatrix} \frac{1}{6} \\ \frac{1}{3} \\ \frac{1}{3} \\ \frac{1}{6} \end{bmatrix} \otimes \mathbf{I}_N, \qquad \mathbf{Q}_k = \begin{bmatrix} \mathbf{q}(t_{k-1}) \\ \mathbf{q}(t_{k-1} + \frac{1}{2}\tau_k) \\ \mathbf{q}(t_{k-1} + \frac{1}{2}\tau_k) \\ \mathbf{q}(t_k) \end{bmatrix},$$

$$\mathbf{F}_k = \begin{bmatrix} \mathbf{F}_{k,1} \\ \mathbf{F}_{k,2} \\ \mathbf{F}_{k,3} \\ \mathbf{F}_{k,4} \end{bmatrix} = \begin{bmatrix} \mathbf{f}\left(\mathbf{y}_{k-1}, t_{k-1}\right) \\ \mathbf{f}\left(\mathbf{y}_{k-1} + \frac{\tau_k}{2}\mathbf{Y}_{k,1}, t_{k-1} + \frac{1}{2}\tau_k\right) \\ \mathbf{f}\left(\mathbf{y}_{k-1} + \frac{\tau_k}{2}\mathbf{Y}_{k,2}, t_{k-1} + \frac{1}{2}\tau_k\right) \\ \mathbf{f}\left(\mathbf{y}_{k-1} + \tau_k \mathbf{Y}_{k,3}, t_k\right) \end{bmatrix}.$$

A fourth order 2-stage implicit RK method is given by

$$\mathbf{A}_s = \begin{bmatrix} \frac{1}{4} & \frac{1}{24}(6 - 4\sqrt{3}) \\ \frac{1}{24}(6 + 4\sqrt{3}) & \frac{1}{4} \end{bmatrix}, \quad \mathbf{b}_s = \begin{bmatrix} \frac{1}{2} \\ \frac{1}{2} \end{bmatrix} \quad \text{and} \quad \mathbf{c}_s = \begin{bmatrix} \frac{1}{6}(3 - \sqrt{3}) \\ \frac{1}{6}(3 + \sqrt{3}) \end{bmatrix},$$

and hence at some time step k we have

$$\mathbf{B} = -\tau_k \mathbf{b}_4 \otimes \mathbf{I}_N = -\tau_k \begin{bmatrix} \frac{1}{2} \\ \frac{1}{2} \end{bmatrix} \otimes \mathbf{I}_N, \qquad \mathbf{Q}_k = \frac{1}{6} \begin{bmatrix} \mathbf{q}(t_{k-1} + (3 - \sqrt{3})\tau_k) \\ \mathbf{q}(t_{k-1} + (3 + \sqrt{3})\tau_k) \end{bmatrix},$$

$$\mathbf{F}_k = \begin{bmatrix} \mathbf{F}_{k,1} \\ \mathbf{F}_{k,2} \end{bmatrix} = \begin{bmatrix} \mathbf{f}\left(\mathbf{y}_{k-1} + \frac{\tau_k}{4}\mathbf{Y}_{k,1} + \frac{1}{24}(6-4\sqrt{3})\mathbf{Y}_{k,2}, t_{k-1} + (3-\sqrt{3})\tau_k\right) \\ \mathbf{f}\left(\mathbf{y}_{k-1} + \frac{1}{24}(6+4\sqrt{3})\mathbf{Y}_{k,1} + \frac{\tau_k}{4}\mathbf{Y}_{k,2}, t_{k-1} + (3+\sqrt{3})\tau_k\right) \end{bmatrix}.$$

2.2 Implicit-Explicit Time-Stepping Methods

For many PDEs there are often natural splittings of the right hand sides of the differential systems into two parts, one of which is a non-stiff (or mildly stiff) term that can be handled by explicit time integration, and the other part is stiff and requires implicit time integration. This leads to an ODE system of the form (cf. (1.1))

$$\begin{aligned} \frac{\partial \mathbf{y}}{\partial t} &= \mathbf{f}(\mathbf{y}, t) + \mathbf{q}(t) = \mathbf{f}^{\mathcal{E}}(\mathbf{y}, t) + \mathbf{f}^{\mathcal{I}}(\mathbf{y}, t) + \mathbf{q}(t) \\ \mathbf{y}(0) &= \mathbf{y}_0, \end{aligned} \tag{2.17}$$

where $\mathbf{f}^{\mathcal{E}}$ represents the non-stiff process, for example convection, and $\mathbf{f}^{\mathcal{I}}$ represents the stiff process, for instance chemical reaction or diffusion. Implicit-explicit (IMEX) time-stepping methods are a popular class of time-stepping methods that can be used to solve ODE systems of this type. There are main variants of IMEX methods, linear multistep (IMEX LM) and Runge-Kutta (IMEX RK) methods, which we describe below.

We treat the source term $\mathbf{q}$ separately because it depends on the model parameters $\mathbf{s}$ and not the model parameters $\mathbf{m}$, as $\mathbf{f}^{\mathcal{E}}$ and $\mathbf{f}^{\mathcal{I}}$ do. For both IMEX LM and IMEX RK methods we have taken it to be part of the explicit term $\mathbf{f}^{\mathcal{E}}$ and handle it accordingly.

IMEX methods were proposed in the late 1970's [27, 124] and have enjoyed much attention since then, especially for solving convection-diffusion-reaction problems [70]. Applications include mathematical biology [89, 96], weather forecasting [130], optimal control [59] and options pricing [98].

There is a large amount of work that has been published on IMEX methods over the last few decades, here we follow the approach taken in the pioneering work in

[7, 8].

2.2.1 IMEX Linear Multistep Methods

The s-step IMEX linear multistep (IMEX LM) scheme for (2.17), with $s > 1$, is given by

$$\sum_{j=0}^{s} \alpha_j^{(s)} \mathbf{y}_{k-j} = \tau \sum_{j=1}^{s} \beta_j^{(s)} \mathbf{f}_{k-j}^{\mathcal{E}} + \tau \sum_{j=0}^{s} \gamma_j^{(s)} \mathbf{f}_{k-j}^{\mathcal{I}} + \tau \sum_{j=1}^{s} \beta_j^{(s)} \mathbf{q}_{k-j}, \tag{2.18}$$

with $\alpha_0^{(s)} = 1$.

The order of accuracy of IMEX LM methods is equal to the number of steps s. IMEX LM methods are designed by either choosing an implicit method (usually BDF) and combining it with an explicit scheme of the same order, or by choosing an explicit scheme such as an Adams-Bashforth or total-variation bounded (TVB) method, then picking an implicit scheme with the same order of accuracy and good stability or damping properties.

We now let $\mathbf{A}$ and $\mathbf{B}$ be defined using (2.1) and (2.3), as in Section 2.1.1, with $\beta_0^{(s)} = 0$. We obtain the matrices $\mathbf{\Gamma}^{(s)}$, $\mathbf{\Gamma}^{(1:s)}$ and $\mathbf{\Gamma}$ by letting $\times_j = \gamma_j^{(s)}$ in $\times^{(s)}$, $\times^{(1:s)}$ and $\times$. We also define the $KN \times N$ block matrices

$$\begin{aligned}
\boldsymbol{\alpha} &= \begin{bmatrix} \alpha_1^{(1)} & \alpha_2^{(2)} & \cdots & \alpha_s^{(s)} & \mathbf{0}_{1\times(K-s)} \end{bmatrix}^\top \otimes \mathbf{I}_N \\
\boldsymbol{\beta}^\top &= \begin{bmatrix} \beta_1^{(1)} & \beta_2^{(2)} & \cdots & \beta_s^{(s)} & \mathbf{0}_{1\times(K-s)} \end{bmatrix}^\top \otimes \mathbf{I}_N \\
\boldsymbol{\gamma}^\top &= \begin{bmatrix} \gamma_1^{(1)} & \gamma_2^{(2)} & \ldots & \gamma_s^{(s)} & \mathbf{0}_{1\times(K-s)} \end{bmatrix}^\top \otimes \mathbf{I}_N.
\end{aligned} \tag{2.19}$$

Then (2.18) can be written as

$$\mathbf{A}\,\overline{\mathbf{y}} = \tau \mathbf{B}\,\overline{\mathbf{f}}^{\mathcal{E}}(\overline{\mathbf{y}}) + \tau \mathbf{\Gamma}\,\overline{\mathbf{f}}^{\mathcal{I}}(\overline{\mathbf{y}}) + \tau \boldsymbol{\beta}\,\mathbf{f}_0^{\mathcal{E}}(\mathbf{y}_0) + \tau \boldsymbol{\gamma}\,\mathbf{f}_0^{\mathcal{I}}(\mathbf{y}_0) - \boldsymbol{\alpha}\,\mathbf{y}_0 + \tau \begin{bmatrix} \boldsymbol{\beta} & \mathbf{B} \end{bmatrix} \overline{\mathbf{q}}, \tag{2.20}$$

with

$$\begin{aligned}\overline{\mathbf{y}} &= \begin{bmatrix}\mathbf{y}_1^\top & \cdots & \mathbf{y}_K^\top\end{bmatrix}^\top, & \overline{\mathbf{f}}^{\mathcal{E}}(\mathbf{y}) &= \begin{bmatrix}\mathbf{f}_1^{\mathcal{E}\top} & \cdots & \mathbf{f}_K^{\mathcal{E}\top}\end{bmatrix}^\top \\ \overline{\mathbf{q}} &= \begin{bmatrix}\mathbf{q}_0^\top & \mathbf{q}_1^\top & \cdots & \mathbf{q}_K^\top\end{bmatrix}^\top, & \overline{\mathbf{f}}^{\mathcal{I}}(\mathbf{y}) &= \begin{bmatrix}\mathbf{f}_1^{\mathcal{I}\top} & \cdots & \mathbf{f}_K^{\mathcal{I}\top}\end{bmatrix}^\top.\end{aligned} \tag{2.21}$$

The method can then be represented by the time-stepping equation $\mathbf{t}(\mathbf{y};\mathbf{y}_0) = \mathbf{S}\,\mathbf{q}$, with $\mathbf{S} = \tau\begin{bmatrix}\boldsymbol{\beta} & \mathbf{B}\end{bmatrix}$ and the time-stepping vector

$$\mathbf{t}(\mathbf{y},\mathbf{y}_0) = \mathbf{A}\,\overline{\mathbf{y}} - \tau\,\mathbf{B}\,\overline{\mathbf{f}}^{\mathcal{E}}(\overline{\mathbf{y}}) - \tau\,\boldsymbol{\Gamma}\,\overline{\mathbf{f}}^{\mathcal{I}}(\overline{\mathbf{y}}) - \tau\,\boldsymbol{\beta}\,\mathbf{f}_0^{\mathcal{E}} - \tau\,\boldsymbol{\gamma}\,\mathbf{f}_0^{\mathcal{I}} + \boldsymbol{\alpha}\,\mathbf{y}_0 \tag{2.22}$$

$\mathbf{t}$ is a vector of length KN and the kth time level is given by the kth subvector of length N:

$$\mathbf{t}_k(\mathbf{y},\mathbf{y}_0) = \sum_{i=0}^{s}\alpha_i^{(s)}\mathbf{y}_{k-i} - \tau\sum_{i=1}^{s}\beta_i^{(s)}\mathbf{f}_{k-i}^{\mathcal{E}} - \tau\sum_{i=0}^{s}\gamma_i^{(s)}\mathbf{f}_{k-i}^{\mathcal{I}}. \tag{2.23}$$

Examples

The coefficients for the second-order IMEX BDF2 method are

$$\alpha_{0:2}^{(2)} = \begin{bmatrix}1 & -\frac{4}{3} & \frac{1}{3}\end{bmatrix}^\top, \qquad \beta_{1:2}^{(2)} = \begin{bmatrix}\frac{4}{3} & -\frac{2}{3}\end{bmatrix}^\top \quad \text{and} \quad \gamma_{0:2}^{(2)} = \begin{bmatrix}\frac{2}{3} & 0 & 0\end{bmatrix}^\top.$$

The third-order IMEX BDF3 method is defined by

$$\begin{aligned}\alpha_{0:3}^{(3)} &= \begin{bmatrix}1 & -\frac{18}{11} & \frac{9}{11} & -\frac{2}{11}\end{bmatrix}^\top \\ \beta_{1:3}^{(3)} &= \begin{bmatrix}\frac{18}{11} & -\frac{18}{11} & \frac{6}{11}\end{bmatrix}^\top \\ \gamma_{0:3}^{(3)} &= \begin{bmatrix}\frac{6}{11} & 0 & 0 & 0\end{bmatrix}^\top\end{aligned}$$

and the third-order IMEX TVB3 method has the coefficients

$$\begin{aligned}\alpha_{0:3}^{(3)} &= \begin{bmatrix}1 & -\frac{3909}{2048} & \frac{1367}{1024} & -\frac{873}{2048}\end{bmatrix}^\top \\ \beta_{1:3}^{(3)} &= \begin{bmatrix}\frac{18463}{12288} & -\frac{1271}{768} & \frac{8233}{12288}\end{bmatrix}^\top\end{aligned}$$

$$\gamma_{0:3}^{(3)} = \begin{bmatrix} \frac{1089}{2048} & -\frac{1139}{12288} & -\frac{367}{6144} & \frac{1699}{12288} \end{bmatrix}^\top .$$

2.2.2 IMEX Runge-Kutta Methods

The family of s-stage IMEX Runge-Kutta (IMEX RK) methods [7] to solve (2.17) is

$$\mathbf{y}_k = \mathbf{y}_{k-1} + \tau_k \sum_{i=1}^{s} b_i^{\mathcal{E}} \mathbf{Y}_{k,i}^{\mathcal{E}} + \tau_k \sum_{i=1}^{s-1} b_i^{\mathcal{I}} \mathbf{Y}_{k,i}^{\mathcal{I}}. \tag{2.24a}$$

Let $t_{k-1,\sigma} = t_{k-1} + c_\sigma \tau_k$. The internal stages are

$$\mathbf{Y}_{k,1}^{\mathcal{E}} = \mathbf{f}^{\mathcal{E}} \left(\mathbf{y}_{k-1}, t_{k-1}\right) + \mathbf{q}(t_{k-1}) \tag{2.24b}$$

and for $\sigma = 1, \cdots, s-1$, solve

$$\mathbf{Y}_{k,\sigma}^{\mathcal{I}} = \mathbf{f}^{\mathcal{I}} \left(\underline{\mathbf{y}}_{k,\sigma}, t_{k-1,\sigma}\right), \tag{2.24c}$$

with

$$\underline{\mathbf{y}}_{k,\sigma} = \mathbf{y}_{k-1} + \tau_k \sum_{i=1}^{\sigma} a_{\sigma+1,i}^{\mathcal{E}} \mathbf{Y}_{k,i}^{\mathcal{E}} + \tau_k \sum_{i=1}^{\sigma} a_{\sigma,i}^{\mathcal{I}} \mathbf{Y}_{k,i}^{\mathcal{I}}, \tag{2.24d}$$

and then evaluate

$$\mathbf{Y}_{k,\sigma+1}^{\mathcal{E}} = \mathbf{f}^{\mathcal{E}} \left(\underline{\mathbf{y}}_{k,\sigma}, t_{k-1,\sigma}\right) + \mathbf{q}\left(t_{k-1,\sigma}\right). \tag{2.24e}$$

The implicit schemes are generally chosen to be diagonally-implicit, so we restrict our discussion to this case. IMEX RK methods can be represented by a pair of Butcher

tableaux, one for the explicit method and one for the implicit method:

$$
\begin{array}{c|ccccc|ccccc}
0 & 0 & & & & & 0 & & & & \\
c_2 & 0 & a^{\mathcal{I}}_{1,1} & & & & a^{\mathcal{E}}_{2,1} & 0 & & & \\
c_3 & 0 & a^{\mathcal{I}}_{2,1} & a^{\mathcal{I}}_{2,2} & & & a^{\mathcal{E}}_{3,1} & a^{\mathcal{E}}_{3,2} & 0 & & \\
\vdots & \vdots & \vdots & & \ddots & & & \vdots & & \ddots & \\
c_s & 0 & a^{\mathcal{I}}_{s-1,1} & a^{\mathcal{I}}_{s-1,2} & \cdots & a^{\mathcal{I}}_{s-1,s-1} & a^{\mathcal{E}}_{s,1} & a^{\mathcal{E}}_{s,2} & \cdots & a^{\mathcal{E}}_{s,s-1} & 0 \\
\hline
 & 0 & b^{\mathcal{I}}_1 & b^{\mathcal{I}}_2 & \cdots & b^{\mathcal{I}}_{s-1} & b^{\mathcal{E}}_1 & b^{\mathcal{E}}_2 & \cdots & b^{\mathcal{E}}_{s-1} & b^{\mathcal{E}}_s.
\end{array}
\tag{2.25}
$$

As for RK methods, for stiff problems IMEX RK methods are susceptible to the problem of *order reduction*, where the order of accuracy of the method can be reduced significantly.

Letting

$$
\mathbf{B} = -\tau_k \begin{bmatrix} b^{\mathcal{E}}_1 & b^{\mathcal{I}}_1 & b^{\mathcal{E}}_2 & \cdots & b^{\mathcal{E}}_{s-1} & b^{\mathcal{I}}_{s-1} & b^{\mathcal{E}}_s \end{bmatrix}^\top \otimes \mathbf{I}_N, \tag{2.26}
$$

the single step in (2.24) can be written in matrix form as

$$
\begin{bmatrix} \mathbf{I}_N & & \\ & \mathbf{I}_{(2s+1)N} & \\ -\mathbf{I}_N & \mathbf{B}^\top & \mathbf{I}_N \end{bmatrix} \begin{bmatrix} \mathbf{y}_k \\ \mathbf{Y}_k \\ \mathbf{y}_k \end{bmatrix} - \begin{bmatrix} \mathbf{0}_{N\times 1} \\ \mathbf{F}_k(\mathbf{y}) \\ \mathbf{0}_{N\times 1} \end{bmatrix} = \begin{bmatrix} \mathbf{y}_{k-1} \\ \mathbf{Q}_k \\ \mathbf{0}_{N\times 1} \end{bmatrix}, \tag{2.27}
$$

with

$$
\mathbf{Y}_k = \begin{bmatrix} \mathbf{Y}_{k,1} \\ \vdots \\ \mathbf{Y}_{k,s} \end{bmatrix}, \quad \mathbf{F}_k(\mathbf{y}) = \begin{bmatrix} \mathbf{F}_{k,1}(\mathbf{y}) \\ \vdots \\ \mathbf{F}_{k,s}(\mathbf{y}) \end{bmatrix} \quad \text{and} \quad \mathbf{Q}_k = \begin{bmatrix} \mathbf{Q}_{k,1} \\ \vdots \\ \mathbf{Q}_{k,s} \end{bmatrix}, \tag{2.28}
$$

where we have let

$$
\mathbf{Y}_{k,i} = \begin{cases} \mathbf{Y}^{\mathcal{E}}_{k,\lceil \frac{i}{2} \rceil} & \text{if } i \text{ odd} \\ \mathbf{Y}^{\mathcal{I}}_{k,\frac{i}{2}} & \text{if } i \text{ even} \end{cases} \tag{2.29}
$$

and

$$\mathbf{F}_{k,i} = \begin{cases} \mathbf{f}^{\mathcal{E}}\left(\underline{\mathbf{y}}_{k,\lceil\frac{i}{2}\rceil-1}, t_{k-1,\lceil\frac{i}{2}\rceil}\right) & \text{if } i \text{ odd} \\ \mathbf{f}^{\mathcal{I}}\left(\underline{\mathbf{y}}_{k,\frac{i}{2}}, t_{k-1,\frac{i}{2}+1}\right) & \text{if } i \text{ even,} \end{cases} \tag{2.30}$$

with $\underline{\mathbf{y}}_{k,\sigma} = \mathbf{y}_{k-1} + \tau_k \sum_{i=1}^{\sigma} a^{\mathcal{E}}_{\sigma+1,i}\mathbf{Y}^{\mathcal{E}}_{k,i} + \tau_k \sum_{i=1}^{\sigma} a^{\mathcal{I}}_{\sigma,i}\mathbf{Y}^{\mathcal{I}}_{k,i}$.

In order to write the IMEX RK scheme in the form of the time-stepping equation (1.2), we let the time-stepping vector be

$$\mathbf{t}(\overline{\mathbf{y}}) = \mathbf{T}\overline{\mathbf{y}} - \overline{\mathbf{f}}, \tag{2.31}$$

where

$$\overline{\mathbf{y}} = \begin{bmatrix} \mathbf{Y}_1^\top & \mathbf{y}_1^\top & \mathbf{Y}_2^\top & \mathbf{y}_2^\top & \cdots & \mathbf{Y}_K^\top & \mathbf{y}_K^\top \end{bmatrix}^\top \tag{2.32}$$

$$\overline{\mathbf{f}} = \begin{bmatrix} \mathbf{F}_1^\top & \mathbf{y}_0 & \mathbf{F}_2^\top & \mathbf{0}_{1\times N} & \cdots & \mathbf{F}_K^\top & \mathbf{0}_{1\times N} \end{bmatrix}^\top \tag{2.33}$$

$$\mathbf{T} = \begin{bmatrix} \mathbf{I}_{(2s+1)N} & & & & & & \\ \mathbf{B}^\top & \mathbf{I}_N & & & & & \\ & & \mathbf{I}_{(2s+1)N} & & & & \\ & -\mathbf{I}_N & \mathbf{B}^\top & \mathbf{I}_N & & & \\ & & \ddots & \ddots & \ddots & & \\ & & & & & \mathbf{I}_{(2s+1)N} & \\ & & & & -\mathbf{I}_N & \mathbf{B}^\top & \mathbf{I}_N \end{bmatrix}. \tag{2.34}$$

$\mathbf{T}$ is a block lower-triangular matrix. We have explicitly included the internal stages in $\overline{\mathbf{y}}$ because they will be needed later on.

The source term is given by

$$\overline{\mathbf{q}} = \begin{bmatrix} \mathbf{Q}_1^\top & \mathbf{0}_{1\times N} & \mathbf{Q}_2^\top & \mathbf{0}_{1\times N} & \cdots & \mathbf{Q}_K^\top & \mathbf{0}_{1\times N} \end{bmatrix}^\top \tag{2.35}$$

and $\mathbf{S} = \mathbf{I}_{2K(s+1)N\times 2K(s+1)N}$.

Examples

The ARS3 scheme found in [7] is a 3-stage, order 3 method that has the following Butcher tableaus:

$$\begin{array}{c|ccc} 0 & 0 & & \\ \gamma & 0 & \gamma & \\ 1-\gamma & 0 & 1-2\gamma & \gamma \\ \hline & 0 & 1/2 & 1/2 \end{array} \quad \begin{array}{c|ccc} 0 & & & \\ \gamma & \gamma & 0 & \\ 1-\gamma & \gamma-1 & 2(1-\gamma) & 0 \\ \hline & 0 & 1/2 & 1/2 \end{array},$$

where $\gamma = \frac{3+\sqrt{3}}{6}$. As mentioned above, reduction of order can be expected when the problem is stiff.

There are several other IMEX RK schemes of higher order, including an 6-stage scheme with order 4 derived in [70].

2.3 Staggered Time-Stepping Methods

Ghrist et al. [42, 43] introduced high-order staggered time-stepping methods for hyperbolic systems in first-order form

$$\frac{\partial \mathbf{y}}{\partial t} = \mathbf{f}\left(\mathbf{y}(t), t, \mathbf{m}\right) + \mathbf{q}(t) \quad \Rightarrow \quad \begin{aligned} \frac{\partial \mathbf{u}}{\partial t} &= \mathbf{f}^{\mathbf{u}}\left(\mathbf{v}(t), t\right) + \mathbf{q}^{\mathbf{u}}(t) \\ \frac{\partial \mathbf{v}}{\partial t} &= \mathbf{f}^{\mathbf{v}}\left(\mathbf{u}(t), t\right) + \mathbf{q}^{\mathbf{v}}(t). \end{aligned} \tag{2.36}$$

For ODE systems in this form it is known that staggered grids in space can increase the accuracy of finite difference and pseudospectral difference methods [37]. A similar staggering in time of the two variables can be performed, leading to staggered linear multistep (StagLM) and staggered Runge-Kutta (StagRK) methods. The analysis given in [42, 43] shows that the staggered time-stepping methods can have a significantly reduced error constant and an increased imaginary stability boundary compared to their non-staggered counterparts.

The efficiency of these methods was investigated in [125, 126] for linear wave equations, and the stability and convergence properties of StagRK schemes for semilinear wave equations was looked at in [84]. The stability of StagLM methods was revisited in [41].

To describe the staggered methods and how they can be represented by (1.2), we discretize $\mathbf{u}(t)$ in time on the integer time levels, so that $\mathbf{u}_k \approx \mathbf{u}(t_k)$, with length $N_{\mathbf{u}}$, and $\mathbf{v}(t)$ is discretized on the half-integer time levels, i.e. $\mathbf{v}_{k+\frac{1}{2}} \approx \mathbf{v}(t_{k+\frac{1}{2}})$, with length $N_{\mathbf{v}}$. Clearly $N = N_{\mathbf{u}} + N_{\mathbf{v}}$. We take it as a given that these two variables would also be staggered in space, but this is not important for the discussion that follows.

We define the following quantities:

$$\begin{aligned}
\mathbf{y}_k &= \begin{bmatrix} \mathbf{u}_k \\ \mathbf{v}_{k+\frac{1}{2}} \end{bmatrix} \\
\mathbf{f}_k &= \begin{bmatrix} \mathbf{f}^{\mathbf{u}}_{k-\frac{1}{2}} \\ \mathbf{f}^{\mathbf{v}}_{k} \end{bmatrix} = \begin{bmatrix} \mathbf{f}^{\mathbf{u}}\left(\mathbf{v}_{k-\frac{1}{2}}, t_{k-\frac{1}{2}}\right) \\ \mathbf{f}^{\mathbf{v}}\left(\mathbf{u}_k, t_k\right) \end{bmatrix} \\
\mathbf{q}_k &= \begin{bmatrix} \mathbf{q}^{\mathbf{u}}_{k-\frac{1}{2}} \\ \mathbf{q}^{\mathbf{v}}_{k} \end{bmatrix} = \begin{bmatrix} \mathbf{q}^{\mathbf{u}}\left(t_{k-\frac{1}{2}}\right) \\ \mathbf{q}^{\mathbf{v}}\left(t_k\right) \end{bmatrix}
\end{aligned} \tag{2.37}$$

for $k \geq 0$, with initial condition $\mathbf{y}_0 = \begin{bmatrix} \mathbf{u}_0^\top & \mathbf{v}_{\frac{1}{2}}^\top \end{bmatrix}^\top$. We let $\mathbf{f}_0 = \begin{bmatrix} \mathbf{0}_{1\times N_{\mathbf{u}}} & \mathbf{f}_0^{\mathbf{v}\top} \end{bmatrix}^\top$ and $\mathbf{q}_0 = \begin{bmatrix} \mathbf{0}_{1\times N_{\mathbf{u}}} & \mathbf{q}_0^{\mathbf{v}\top} \end{bmatrix}^\top$.

2.3.1 Staggered Linear Multistep Methods

A staggered s-step multistep methods may be applied to (2.36), giving us, for $k \geq s$,

$$\begin{aligned} \sum_{j=0}^{s} \alpha_j^{(s)} \mathbf{u}_{k-j} &= \tau \sum_{j=0}^{s} \beta_j^{(s)} \mathbf{f}^{\mathbf{u}}_{k-\frac{1}{2}-j} + \tau \sum_{j=0}^{s} \beta_j^{(s)} \mathbf{q}^{\mathbf{u}}_{k-\frac{1}{2}-j} \\ \sum_{j=0}^{s} \alpha_j^{(s)} \mathbf{v}_{k+1/2-j} &= \tau \sum_{j=0}^{s} \beta_j^{(s)} \mathbf{f}^{\mathbf{v}}_{k-j} + \tau \sum_{j=0}^{s} \beta_j^{(s)} \mathbf{q}^{\mathbf{v}}_{k-j} \end{aligned} \tag{2.38}$$

Recall that we are using fixed time-steps τ for LM-type methods. StagLM methods are classified into staggered backward differentiation formulas and staggered Adams-type methods, along the lines of the corresponding non-staggered methods. Due to time-staggering, even methods with $\beta_0^{(\sigma)} \neq 0$ will lead to explicit methods.

As in Section 2.1.1, let the $KN \times KN$ matrices $\mathbf{A}$ and $\mathbf{B}$ be defined using (2.1) and (2.3), and also define $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ as in that section.

The system (2.38) can then be written as

$$\mathbf{A}\overline{\mathbf{y}} = \tau \mathbf{B}\overline{\mathbf{f}} + \tau \boldsymbol{\beta} \mathbf{f}_0 + \tau \begin{bmatrix} \boldsymbol{\beta} & \mathbf{B} \end{bmatrix} \overline{\mathbf{q}} - \boldsymbol{\alpha} \mathbf{y}_0, \tag{2.39}$$

with

$$\begin{aligned} \overline{\mathbf{y}} &= \begin{bmatrix} \mathbf{y}_1^\top & \cdots & \mathbf{y}_K^\top \end{bmatrix}^\top, \qquad \overline{\mathbf{f}} = \begin{bmatrix} \mathbf{f}_1^\top & \cdots & \mathbf{f}_K^\top \end{bmatrix}^\top \\ \overline{\mathbf{q}} &= \begin{bmatrix} \mathbf{q}_0^\top & \mathbf{q}_1^\top & \cdots & \overline{\mathbf{q}}_K^\top \end{bmatrix}^\top. \end{aligned} \tag{2.40}$$

Then, setting $\mathbf{S} = \tau \begin{bmatrix} \boldsymbol{\beta} & \mathbf{B} \end{bmatrix}$, we arrive at (1.2) by letting

$$\mathbf{t}\left(\overline{\mathbf{y}}; \mathbf{y}_0\right) = \mathbf{A}\overline{\mathbf{y}} + \boldsymbol{\alpha}\mathbf{y}_0 - \tau \mathbf{B}\overline{\mathbf{f}} - \tau \boldsymbol{\beta} \mathbf{f}_0. \tag{2.41}$$

Examples

The leapfrog method is a 1-step StagLM method of order 2, with $\alpha = \begin{bmatrix} 1 & -1 \end{bmatrix}^\top$ and $\beta = 1$. This can be regarded as either a staggered BDF or a staggered Adams-Bashforth scheme.

The next staggered BDF schemes of interest are StagBDF3, with

$$\alpha = \begin{bmatrix} 1 & -\frac{21}{23} & -\frac{3}{23} & \frac{1}{23} \end{bmatrix}^\top \qquad \text{and} \qquad \beta_0 = \frac{24}{23},$$

and StagBDF4, with

$$\alpha = \begin{bmatrix} 1 & -\frac{17}{22} & -\frac{9}{22} & \frac{5}{22} & -\frac{1}{22} \end{bmatrix}^\top \qquad \text{and} \qquad \beta_0 = \frac{12}{11}.$$

It was found in [41] that there are no staggered BDF schemes of higher order.

Higher-order staggered Adams-Bashforth schemes include the third-order StagAB3 scheme, with

$$\alpha = \begin{bmatrix} 1 & -1 \end{bmatrix}^\top \qquad \text{and} \qquad \beta = \begin{bmatrix} \frac{25}{24} & -\frac{1}{12} & \frac{1}{24} \end{bmatrix}^\top,$$

and the fourth-order StagAB4 scheme, with

$$\alpha = \begin{bmatrix} 1 & -1 \end{bmatrix}^\top \qquad \text{and} \qquad \beta = \begin{bmatrix} \frac{13}{12} & -\frac{5}{24} & \frac{1}{6} & -\frac{1}{24} \end{bmatrix}^\top.$$

7th-order and 8th-order schemes are available as well.

2.3.2 Staggered Runge-Kutta Methods

StagRK schemes were derived in [42], where only explicit schemes with fixed time-steps were considered. Due to the time-staggering it is in fact unclear how variable time-steps would be handled, so we will also only consider fixed time-steps here. It is necessary to carefully distinguish between the number of stages s being odd or even, since the parity of s will lead to slightly different formulations.

To propagate from $\mathbf{y}_{k-1}$ to $\mathbf{y}_k$, we compute

- *Number of stages s is odd:*

$$
\begin{aligned}
\mathbf{u}_k &= \mathbf{u}_{k-1} + \tau \sum_{\substack{\sigma=1 \\ \sigma \text{ odd}}}^{s} b_\sigma \mathbf{U}^o_{k,\sigma} \\
\mathbf{v}_{k+\frac{1}{2}} &= \mathbf{v}_{k-\frac{1}{2}} + \tau \sum_{\substack{\sigma=1 \\ \sigma \text{ odd}}}^{s} b_\sigma \mathbf{V}^o_{k+\frac{1}{2},\sigma},
\end{aligned}
\tag{2.42a}
$$

where the internal stages are, for $\sigma = 1, \cdots, s$,

$$
\begin{aligned}
\mathbf{U}^o_{k,\sigma} &= \begin{cases} \mathbf{f}^{\mathbf{u}}\left(\underline{\mathbf{v}}^o_{\mathbf{U},k-\frac{1}{2},\sigma}, t_{k-\frac{1}{2},\sigma}\right) + \mathbf{q}^{\mathbf{u}}(t_{k-\frac{1}{2},\sigma}) & \text{if } \sigma \text{ odd} \\ \mathbf{f}^{\mathbf{v}}\left(\underline{\mathbf{u}}^o_{\mathbf{U},k-1,\sigma}, t_{k-1,\sigma}\right) + \mathbf{q}^{\mathbf{v}}(t_{k-1,\sigma}) & \text{if } \sigma \text{ even} \end{cases} \\
\mathbf{V}^o_{k+\frac{1}{2},\sigma} &= \begin{cases} \mathbf{f}^{\mathbf{v}}\left(\underline{\mathbf{u}}^o_{\mathbf{V},k,\sigma}, t_{k,\sigma}\right) + \mathbf{q}^{\mathbf{v}}(t_{k,\sigma}) & \text{if } \sigma \text{ odd} \\ \mathbf{f}^{\mathbf{u}}\left(\underline{\mathbf{v}}^o_{\mathbf{V},k-\frac{1}{2},\sigma}, t_{k-\frac{1}{2},\sigma}\right) + \mathbf{q}^{\mathbf{u}}(t_{k-\frac{1}{2},\sigma}) & \text{if } \sigma \text{ even.} \end{cases}
\end{aligned}
\tag{2.42b}
$$

- *Number of stages s is even:*

$$
\begin{aligned}
\mathbf{u}_k &= \mathbf{u}_{k-1} + \tau \sum_{\substack{\sigma=2 \\ \sigma \text{ even}}}^{s} b_\sigma \mathbf{U}^e_{k,\sigma} \\
\mathbf{v}_{k+\frac{1}{2}} &= \mathbf{v}_{k-\frac{1}{2}} + \tau \sum_{\substack{\sigma=2 \\ \sigma \text{ even}}}^{s} b_\sigma \mathbf{V}^e_{k+\frac{1}{2},\sigma},
\end{aligned}
\tag{2.43a}
$$

where the internal stages are, for $\sigma = 1, \cdots, s$,

$$
\begin{aligned}
\mathbf{U}^e_{k,\sigma} &= \begin{cases} \mathbf{f}^{\mathbf{v}}\left(\underline{\mathbf{u}}^e_{\mathbf{U},k-1,\sigma}, t_{k-1,\sigma}\right) + \mathbf{q}^{\mathbf{v}}(t_{k-1,\sigma}) & \text{if } \sigma \text{ odd} \\ \mathbf{f}^{\mathbf{u}}\left(\underline{\mathbf{v}}^e_{\mathbf{U},k-\frac{1}{2},\sigma}, t_{k-\frac{1}{2},\sigma}\right) + \mathbf{q}^{\mathbf{u}}(t_{k-\frac{1}{2},\sigma}) & \text{if } \sigma \text{ even} \end{cases} \\
\mathbf{V}^e_{k+\frac{1}{2},\sigma} &= \begin{cases} \mathbf{f}^{\mathbf{u}}\left(\underline{\mathbf{v}}^e_{\mathbf{V},k-\frac{1}{2},\sigma}, t_{k-\frac{1}{2},\sigma}\right) + \mathbf{q}^{\mathbf{u}}(t_{k-\frac{1}{2},\sigma}) & \text{if } \sigma \text{ odd} \\ \mathbf{f}^{\mathbf{v}}\left(\underline{\mathbf{u}}^e_{\mathbf{V},k,\sigma}, t_{k,\sigma}\right) + \mathbf{q}^{\mathbf{v}}(t_{k,\sigma}) & \text{if } \sigma \text{ even.} \end{cases}
\end{aligned}
\tag{2.43b}
$$

For brevity of notation we have let $t_{k-1,\sigma} = t_{k-1} + c_\sigma \tau$, $t_{k-\frac{1}{2},\sigma} = t_{k-\frac{1}{2}} + c_\sigma \tau$, and

$$\begin{aligned}
\underline{\mathbf{v}}^{o/e}_{\mathbf{U},k-\frac{1}{2},\sigma} &= \mathbf{v}_{k-\frac{1}{2}} + \tau \sum_{\substack{i=1\\ i\ \mathrm{e/o}}}^{\sigma-1} a_{\sigma i} \mathbf{U}^{o/e}_{k,i}, & \underline{\mathbf{v}}^{o/e}_{\mathbf{V},k-\frac{1}{2},\sigma} &= \mathbf{v}_{k-\frac{1}{2}} + \tau \sum_{\substack{i=1\\ i\ \mathrm{o/e}}}^{\sigma-1} a_{\sigma i} \mathbf{V}^{o/e}_{k+\frac{1}{2},i} \\
\underline{\mathbf{u}}^{o/e}_{\mathbf{U},k-1,\sigma} &= \mathbf{u}_{k-1} + \tau \sum_{\substack{i=1\\ i\ \mathrm{o/e}}}^{\sigma-1} a_{\sigma i} \mathbf{U}^{o/e}_{k,i}, & \underline{\mathbf{u}}^{o/e}_{\mathbf{V},k,\sigma} &= \mathbf{u}_{k} + \tau \sum_{\substack{i=1\\ i\ \mathrm{e/o}}}^{\sigma-1} a_{\sigma i} \mathbf{V}^{o/e}_{k+\frac{1}{2},i}.
\end{aligned} \tag{2.44}$$

With $\mathbf{Y}^{o/e}_{k,\sigma} = \begin{bmatrix} \mathbf{U}^{o/e\top}_{k,\sigma} & \mathbf{V}^{o/e\ \top}_{k+\frac{1}{2},\sigma} \end{bmatrix}^\top$, (2.42a) and (2.43a) are

$$\mathbf{y}_k = \mathbf{y}_{k-1} + \tau \sum_{\substack{\sigma=1\\ \sigma\ \mathrm{odd}}}^{s} b_\sigma \mathbf{Y}^{o}_{k,\sigma} \qquad \text{and} \qquad \mathbf{y}_k = \mathbf{y}_{k-1} + \tau \sum_{\substack{\sigma=1\\ \sigma\ \mathrm{even}}}^{s} b_\sigma \mathbf{Y}^{e}_{k,\sigma}, \tag{2.45}$$

respectively. In order to represent the procedure in the form of the time-stepping equation, we define

$$\begin{aligned}
\mathbf{B}^o &= -\tau \begin{bmatrix} b_1 & 0 & b_3 & 0 & \cdots & b_{s-2} & 0 & b_s \end{bmatrix}^\top \otimes \mathbf{I}_N, \\
\mathbf{B}^e &= -\tau \begin{bmatrix} 0 & b_2 & 0 & b_4 & \cdots & b_{s-2} & 0 & b_s \end{bmatrix}^\top \otimes \mathbf{I}_N, \\
\mathbf{Y}^{o/e}_k &= \begin{bmatrix} \mathbf{Y}^{o/e}_{k,1} \\ \vdots \\ \mathbf{Y}^{o/e}_{k,s} \end{bmatrix}, \qquad \mathbf{Q}^{o/e}_k = \begin{bmatrix} \mathbf{Q}^{o/e}_{k,1} \\ \vdots \\ \mathbf{Q}^{o/e}_{k,s} \end{bmatrix} \qquad \text{and} \qquad \mathbf{F}^{o/e}_k = \begin{bmatrix} \mathbf{F}^{o/e}_{k,1} \\ \vdots \\ \mathbf{F}^{o/e}_{k,s} \end{bmatrix}
\end{aligned} \tag{2.46}$$

with

	$\mathbf{Q}^{o}_{k,\sigma}$	$\mathbf{Q}^{e}_{k,\sigma}$
σ odd	$\begin{bmatrix} \mathbf{q}^{\mathbf{u}}(t_{k-\frac{1}{2},\sigma}) \\ \mathbf{q}^{\mathbf{v}}(t_{k,\sigma}) \end{bmatrix}$	$\begin{bmatrix} \mathbf{q}^{\mathbf{v}}(t_{k-1,\sigma}) \\ \mathbf{q}^{\mathbf{u}}(t_{k-\frac{1}{2},\sigma}) \end{bmatrix}$
σ even	$\begin{bmatrix} \mathbf{q}^{\mathbf{v}}(t_{k-1,\sigma}) \\ \mathbf{q}^{\mathbf{u}}(t_{k-\frac{1}{2},\sigma}) \end{bmatrix}$	$\begin{bmatrix} \mathbf{q}^{\mathbf{u}}(t_{k-\frac{1}{2},\sigma}) \\ \mathbf{q}^{\mathbf{v}}(t_{k,\sigma}) \end{bmatrix}$

(2.47)

and

	$\mathbf{F}^o_{k,\sigma}$	$\mathbf{F}^e_{k,\sigma}$
σ odd	$\begin{bmatrix} \mathbf{f}^{\mathbf{u}}\left(\underline{\mathbf{v}}^o_{\mathbf{U},k-\frac{1}{2},\sigma}, t_{k-\frac{1}{2},\sigma}\right) \\ \mathbf{f}^{\mathbf{v}}\left(\underline{\mathbf{u}}^o_{\mathbf{V},k,\sigma}, t_{k,\sigma}\right) \end{bmatrix}$	$\begin{bmatrix} \mathbf{f}^{\mathbf{v}}\left(\underline{\mathbf{u}}^e_{\mathbf{U},k-1,\sigma}, t_{k-1,\sigma}\right) \\ \mathbf{f}^{\mathbf{u}}\left(\underline{\mathbf{v}}^e_{\mathbf{V},k-\frac{1}{2},\sigma}, t_{k-\frac{1}{2},\sigma}\right) \end{bmatrix}$
σ even	$\begin{bmatrix} \mathbf{f}^{\mathbf{v}}\left(\underline{\mathbf{u}}^o_{\mathbf{U},k-1,\sigma}, t_{k-1,\sigma}\right) \\ \mathbf{f}^{\mathbf{u}}\left(\underline{\mathbf{v}}^o_{\mathbf{V},k-\frac{1}{2},\sigma}, t_{k-\frac{1}{2},\sigma}\right) \end{bmatrix}$	$\begin{bmatrix} \mathbf{f}^{\mathbf{u}}\left(\underline{\mathbf{v}}^e_{\mathbf{U},k-\frac{1}{2},\sigma}, t_{k-\frac{1}{2},\sigma}\right) \\ \mathbf{f}^{\mathbf{v}}\left(\underline{\mathbf{u}}^e_{\mathbf{V},k,\sigma}, t_{k,\sigma}\right) \end{bmatrix}$

(2.48)

The propagation for $\mathbf{y}_k$, for both an odd and an even number of stages, can then abstractly be represented by

$$\begin{bmatrix} \mathbf{I}_N & & \\ & \mathbf{I}_{sN} & \\ -\mathbf{I}_N & \left(\mathbf{B}^{o/e}\right)^\top & \mathbf{I}_N \end{bmatrix} \begin{bmatrix} \mathbf{y}_{k-1} \\ \mathbf{Y}^{o/e}_k \\ \mathbf{y}_k \end{bmatrix} - \begin{bmatrix} \mathbf{0}_{N\times 1} \\ \mathbf{F}^{o/e}_k \\ \mathbf{0}_{N\times 1} \end{bmatrix} = \begin{bmatrix} \mathbf{y}_{k-1} \\ \mathbf{Q}^{o/e}_k \\ \mathbf{0}_{N\times 1} \end{bmatrix}, \tag{2.49}$$

(2.49) constitutes a single time step in the solution procedure, so the StagRK procedure as a whole can be written in the form (1.2) with $\mathbf{S} = \mathbf{I}_{(s+1)KN\times(s+1)KN}$,

$$\overline{\mathbf{q}} = \overline{\mathbf{q}}^{o/e} = \begin{bmatrix} \left(\mathbf{q}^{o/e}_1\right)^\top & \mathbf{0}_{1\times N} & \left(\mathbf{q}^{o/e}_2\right)^\top & \mathbf{0}_{1\times N} & \cdots & \left(\mathbf{q}^{o/e}_K\right)^\top & \mathbf{0}_{1\times N} \end{bmatrix}^\top \tag{2.50a}$$

$$\mathbf{t}\left(\overline{\mathbf{y}}; \mathbf{y}_0\right) = \mathbf{t}^{o/e}\left(\overline{\mathbf{y}}\right) = \mathbf{T}\overline{\mathbf{y}} - \overline{\mathbf{f}}\left(\overline{\mathbf{y}}, \mathbf{y}_0\right), \tag{2.50b}$$

where

$$\overline{\mathbf{y}} = \overline{\mathbf{y}}^{o/e} = \begin{bmatrix} \left(\mathbf{Y}^{o/e}_1\right)^\top & \mathbf{y}_1^\top & \left(\mathbf{Y}^{o/e}_2\right)^\top & \mathbf{y}_2^\top & \cdots & \left(\mathbf{Y}^{o/e}_K\right)^\top & \mathbf{y}_K^\top \end{bmatrix}^\top$$

$$\overline{\mathbf{f}} = \overline{\mathbf{f}}^{o/e} = \begin{bmatrix} \left(\mathbf{F}^{o/e}_1\right)^\top & \mathbf{y}_0^\top & \left(\mathbf{F}^{o/e}_2\right)^\top & \mathbf{0}_{1\times N} & \cdots & \left(\mathbf{F}^{o/e}_K\right)^\top & \mathbf{0}_{1\times N} \end{bmatrix}^\top$$

and

$$\mathbf{T} = \mathbf{T}^{o/e} = \begin{bmatrix} \mathbf{I}_{sN} & & & & & & \\ \left(\mathbf{B}^{o/e}\right)^\top & \mathbf{I}_N & & & & & \\ & & \mathbf{I}_{sN} & & & & \\ & -\mathbf{I}_N & \left(\mathbf{B}^{o/e}\right)^\top & \mathbf{I}_N & & & \\ & & \ddots & \ddots & \ddots & & \\ & & & & & \mathbf{I}_{sN} & \\ & & & & -\mathbf{I}_N & \left(\mathbf{B}^{o/e}\right)^\top & \mathbf{I}_N \end{bmatrix}, \tag{2.51}$$

which is a block lower-triangular matrix. We have explicitly included the internal stages in $\overline{\mathbf{y}}$ because they will be needed later on.

Example

The leapfrog method is a 1-stage StagRK method of order 2. The next StagRK of interest is the 5-stage method of order 4 (StagRK4) that has the Butcher tableau

$$\begin{array}{c|c} \mathbf{c}_s & \mathbf{A}_s \\ \hline & \mathbf{b}_s^\top \end{array} = \begin{array}{c|ccccc} 0 & 0 & & & & \\ \frac{1}{4}(2-\gamma) & \frac{1}{4}(2-\gamma) & 0 & & & \\ -\frac{1}{2}\gamma & & -\frac{1}{2}\gamma & 0 & & \\ \frac{1}{4}(2+\gamma) & \frac{1}{4}(2+\gamma) & & & 0 & \\ \frac{1}{2}\gamma & & & & \frac{1}{2}\gamma & 0 \\ \hline & 1-2b_5 & 0 & b_5 & 0 & b_5 \end{array}.$$

The method is parametrized by $\gamma = (6b_5)^{-1/2}$ and the most appealing member of this family was found [42] to have the value $b_5 = 1/24$ (and therefore $\gamma = 2$). Note that $\mathbf{U}^o_{k,1} = \mathbf{V}^o_{k+\frac{1}{2},2}$ and $\mathbf{V}^o_{k+\frac{1}{2},1} = \mathbf{U}^o_{k+1,2}$, so only 4 evaluations of each of $\mathbf{f}^\mathbf{v}$ and $\mathbf{f}^\mathbf{u}$ are required at each time level.

2.4 Exponential Time-Differencing Methods

Exponential time-differencing (ETD) methods, also referred to as exponential integrators, were originally developed in the 1960s [18, 91] and have attracted much recent attention [63, 78–80, 120, 121]. They have become an important approach to numerically solving PDEs, particularly when the PDE system is time-dependent, semi-linear, and exhibits stiffness, taking on the form

$$\begin{aligned} \frac{\partial \mathbf{y}(t)}{\partial t} &= \mathbf{f}(\mathbf{y}(t), t) = \mathbf{L}\mathbf{y}(t) + \mathbf{n}(\mathbf{y}(t), t) + \mathbf{q}(t) \\ \mathbf{y}(0) &= \mathbf{y}_0. \end{aligned} \tag{2.52}$$

The operator $\mathbf{L}$ contains the leading derivatives and is linear, while $\mathbf{n}$ is generally nonlinear in $\mathbf{y}$. Many interesting PDEs have this form.

Before we review exponential integration, we first give a quick overview of Rosenbrock-type methods. Here, an ODE system resulting from a spatial semi-discretization of a fully nonlinear time-dependent PDE $\frac{\partial}{\partial t}\mathbf{y}(t) = \mathbf{f}(\mathbf{y}(t), t)$ is written in semi-linear form, after which an exponential integrator can be applied. This can be done by writing

$$\mathbf{f}(\mathbf{y}(t), t) = \mathbf{L}_{k-1}\mathbf{y}(t) + \mathbf{n}_{k-1}(\mathbf{y}(t), t) + \mathbf{q}(t), \tag{2.53}$$

where $\mathbf{n}_{k-1}(\mathbf{y}(t), t) = \mathbf{f}(\mathbf{y}(t), t) - \mathbf{L}_{k-1}\mathbf{y}(t)$. ETD schemes are applied to this linearization, involving exponentiation of the matrix $\mathbf{L}_{k-1}$ and leading to exponential Rosenbrock-type methods [64, 65]. The matrix $\mathbf{L}_{k-1}$ is meant to approximate the Jacobian $\mathbf{f}_{\mathbf{y}}(\mathbf{y}_{k-1}, t_{k-1})$, or some part of it, at the kth time-step. Occasionally it is sufficient to select $\mathbf{L}_{k-1} = \mathbf{L}$ independently of time, but when the dynamical system trajectory varies significantly and rapidly we may have to set $\mathbf{L}_{k-1} = \mathbf{L}_{k-1}(\mathbf{y}_{k-1}) = \dfrac{\partial \mathbf{f}}{\partial \mathbf{y}}(\mathbf{y}_{k-1}, t_{k-1})$. The case where the linear operator $\mathbf{L}_{k-1}$ depends on $\mathbf{y}_{k-1}$ adds a significant amount of complexity to the adjoint computations considered later.

In this section we will work with the linearized system (2.53), where $\mathbf{L}_{k-1} = \mathbf{L}$ and

$\mathbf{n}_{k-1}(\mathbf{y}(t), t) = \mathbf{n}(\mathbf{y}(t), t)$ if the system is semilinear to begin with an $\mathbf{L}$ is independent of the model parameters. As with the other methods considered so far, there are two approaches to solving (2.53), linear multistep and Runge-Kutta. Exponential Runge-Kutta (ETDRK) methods appear be far more popular, so we limit our discussion to these methods.

2.4.1 Exponential Runge-Kutta Methods

Integrating (2.53) exactly from time level t_{k-1} to $t_k = t_{k-1} + \tau_k$ gives

$$\begin{aligned}\mathbf{y}_k = e^{\tau_k \mathbf{L}_{k-1}}\mathbf{y}_{k-1} + \int_{t_{k-1}}^{t_k} e^{(t_k - t)\mathbf{L}_{k-1}}\, \mathbf{n}_{k-1}\left(\mathbf{y}(t_{k-1}+t), t_{k-1}+t\right) dt + \\ + \int_{t_{k-1}}^{t_k} e^{(t_k - t)\mathbf{L}_{k-1}} \mathbf{q}(t_{k-1}+t) dt\end{aligned} \tag{2.54}$$

The *exponential Euler method* is obtained by interpolating the integrand at the known value $\mathbf{n}_{k-1}(\mathbf{y}_{k-1}, t_{k-1})$ only,

$$\mathbf{y}_k = e^{\tau_k \mathbf{L}_{k-1}}\mathbf{y}_{k-1} + \tau_k \varphi_1(\tau_k \mathbf{L}_{k-1})\left(\mathbf{n}_{k-1}(\mathbf{y}_{k-1}, t_{k-1}) + \mathbf{q}(t_{k-1})\right) \tag{2.55}$$

where $\varphi_1(z) = \frac{e^z - 1}{z}$. This is the simplest numerical method that can be obtained for solving (2.54).

The integral in (2.54) can be approximated using some quadrature rule, leading to a class of s-stage explicit *exponential time-differencing Runge-Kutta* (ETDRK) methods with matrix coefficients $a_{\sigma j}(\tau_k \mathbf{L}_{k-1})$, weights $b_\sigma(\tau_k \mathbf{L}_{k-1})$ and nodes c_σ, so for $1 \le \sigma, j \le s$ we obtain

$$\mathbf{y}_k = e^{\tau_k \mathbf{L}_{k-1}}\mathbf{y}_{k-1} + \tau_k \sum_{\sigma=1}^{s} b_\sigma(\tau_k \mathbf{L}_{k-1})\mathbf{Y}_{k,\sigma}, \tag{2.56a}$$

with the internal stages

$$\mathbf{Y}_{k,\sigma} = \mathbf{n}_{k-1}\left(e^{c_\sigma \tau_k \mathbf{L}_{k-1}}\mathbf{y}_{k-1} + \tau_k \sum_{i=1}^{\sigma-1} a_{\sigma i}(\tau_k \mathbf{L}_{k-1})\mathbf{Y}_{k,i}, t_{k-1,\sigma}\right) + \mathbf{q}(t_{k-1,\sigma}) \tag{2.56b}$$

for $1 \leq \sigma \leq s$, where $t_{k-1,\sigma} = t_{k-1} + c_\sigma \tau_k$.

The procedure starts from a known initial condition $\mathbf{y}_0$. There are several alternate ways of writing (2.56a), but for our purposes the representation given here is most useful.

The Butcher tableau for these methods is

$$\begin{array}{c|cccc}
c_1 & & & & \\
c_2 & a_{21}(\tau \mathbf{L}_{k-1}) & & & \\
\vdots & \vdots & \ddots & & \\
c_s & a_{s1}(\tau_k \mathbf{L}_{k-1}) & \cdots & a_{s,s-1}(\tau_k \mathbf{L}_{k-1}) & \\
\hline
 & b_1(\tau_k \mathbf{L}_{k-1}) & \cdots & b_{s-1}(\tau_k \mathbf{L}_{k-1}) & b_s(\tau_k \mathbf{L}_{k-1})
\end{array}.$$

The coefficients $a_{\sigma i}$ and b_σ are linear combinations of the entire functions

$$\varphi_0(z) = e^z \qquad \varphi_\ell = \int_0^1 e^{(1-\theta)z}\frac{\theta^{\ell-1}}{(\ell-1)!}\,d\theta, \quad \ell \geq 1.$$

It is not hard to see that the φ-functions satisfy the recurrence relation

$$\varphi_\ell(z) = \frac{\varphi_{\ell-1}(z) - \varphi_{\ell-1}(0)}{z}, \quad \ell > 0, \tag{2.57}$$

and that $\varphi_\ell(z) = \sum_{i=0}^{\infty} \frac{z^i}{(i+\ell)!}$. Notice that the expansion of $\varphi_\ell(z)$ is that of the exponential function with the coefficients shifted forward.

As is evident by the structure of φ_ℓ for $\ell > 0$, for small z the evaluation of $\varphi_\ell(z)$ will be subject to cancellation error, and this could become a problem when evaluating $\varphi_\ell(\tau_k \mathbf{L}_{k-1})$ if the matrix $\tau_k \mathbf{L}_{k-1}$ has small eigenvalues. From now on, for brevity of

notation we use $\varphi_\ell = \varphi_\ell(\tau_k \mathbf{L}_{k-1})$ and $\varphi_{\ell,\sigma} = \varphi_\ell(c_\sigma \tau_k \mathbf{L}_{k-1})$.

The method given by (2.56) can be abstractly represented by

$$\begin{bmatrix} \mathbf{I}_N & & \\ & \mathbf{I}_{sN} & \\ -e^{\tau_k \mathbf{L}_{k-1}} & -\mathbf{B}_k^\top & \mathbf{I}_N \end{bmatrix} \begin{bmatrix} \mathbf{y}_{k-1} \\ \mathbf{Y}_k \\ \mathbf{y}_k \end{bmatrix} - \begin{bmatrix} \mathbf{0}_{N\times 1} \\ \mathbf{N}_k \\ \mathbf{0}_{N\times 1} \end{bmatrix} = \begin{bmatrix} \mathbf{y}_{k-1} \\ \mathbf{Q}_k \\ \mathbf{0}_{N\times 1} \end{bmatrix}, \tag{2.58}$$

with

$$\mathbf{B}_k = \tau_k \begin{bmatrix} b_1(\tau_k \mathbf{L}_{k-1}) & \cdots & b_s(\tau_k \mathbf{L}_{k-1}) \end{bmatrix}^\top, \tag{2.59a}$$

and

$$\mathbf{Y}_k = \begin{bmatrix} \mathbf{Y}_{k,1} \\ \vdots \\ \mathbf{Y}_{k,s} \end{bmatrix}, \qquad \mathbf{N}_k = \begin{bmatrix} \mathbf{N}_{k,1}, \\ \vdots \\ \mathbf{N}_{k,s} \end{bmatrix}, \qquad \mathbf{Q}_k = \begin{bmatrix} \mathbf{Q}_{k,1} \\ \vdots \\ \mathbf{Q}_{k,s} \end{bmatrix} \tag{2.59b}$$

where

$$\begin{aligned} \mathbf{N}_{k,\sigma} &= \mathbf{n}_{k-1}\left(e^{c_\sigma \tau_k \mathbf{L}_{k-1}} \mathbf{y}_{k-1} + \tau_k \sum_{j=1}^{\sigma-1} a_{\sigma j}(\tau_k \mathbf{L}_{k-1}) \mathbf{Y}_{k,j}, t_{k-1,\sigma} \right) \\ \mathbf{Q}_{k,\sigma} &= \mathbf{q}\left(t_{k-1,\sigma}\right). \end{aligned}$$

In (2.58) we have a single time step in the solution procedure, so the ETDRK procedure as a whole can be represented in the form of (1.2) with $\mathbf{S} = \mathbf{I}_{(s+1)KN\times(s+1)KN}$,

$$\overline{\mathbf{q}} = \begin{bmatrix} \mathbf{Q}_1^\top & \mathbf{0}_{1\times N} & \mathbf{Q}_2^\top & \mathbf{0}_{1\times N} & \cdots & \mathbf{Q}_K^\top & \mathbf{0}_{1\times N} \end{bmatrix}^\top, \tag{2.60a}$$

$$\mathbf{t}\left(\overline{\mathbf{y}}; \mathbf{y}_0\right) = \mathbf{T}\overline{\mathbf{y}} - \overline{\mathbf{n}}\left(\overline{\mathbf{y}}, \mathbf{y}_0\right) \tag{2.60b}$$

where

$$\overline{\mathbf{y}} = \begin{bmatrix} \mathbf{Y}_1^\top & \mathbf{y}_1^\top & \mathbf{Y}_2^\top & \mathbf{y}_2^\top & \cdots & \mathbf{Y}_K^\top & \mathbf{y}_K^\top \end{bmatrix}^\top,$$

$$\overline{\mathbf{n}} = \begin{bmatrix} \mathbf{N}_1^\top & (e^{\tau_1 \mathbf{L}_0}\mathbf{y}_0)^\top & \mathbf{N}_2^\top & \mathbf{0}_{1\times N} & \cdots & \mathbf{N}_K^\top & \mathbf{0}_{1\times N} \end{bmatrix}^\top,$$

and

$$\mathbf{T} = \begin{bmatrix} \mathbf{I}_{sN} & & & & & & \\ -\mathbf{B}_1^\top & \mathbf{I}_N & & & & & \\ & & \mathbf{I}_{sN} & & & & \\ & -e^{\tau_2 \mathbf{L}_1} & -\mathbf{B}_2^\top & \mathbf{I}_N & & & \\ & & \ddots & \ddots & \ddots & & \\ & & & & & \mathbf{I}_{sN} & \\ & & & & -e^{\tau_K \mathbf{L}_{K-1}} & -\mathbf{B}_K^\top & \mathbf{I}_N \end{bmatrix}, \tag{2.61}$$

which is a block lower-triangular matrix. We have explicitly included the internal stages in $\overline{\mathbf{y}}$ because they will be needed later on.

Examples

The four-stage ETD4RK method of Cox and Matthews [26] has the following Butcher tableau:

$$\begin{array}{c|cccc} 0 & & & & \\ \frac{1}{2} & \frac{1}{2}\varphi_{1,2} & & & \\ \frac{1}{2} & \mathbf{0}_{N\times N} & \frac{1}{2}\varphi_{1,3} & & \\ 1 & \varphi_{1,4} - \varphi_{1,3} & \mathbf{0}_{N\times N} & \varphi_{1,3} & \\ \hline & \varphi_1 - 3\varphi_2 + 4\varphi_3 & 2\varphi_2 - 4\varphi_3 & 2\varphi_2 - 4\varphi_3 & 4\varphi_3 - \varphi_2 \end{array}. \tag{2.62}$$

This method can be fourth-order accurate when certain conditions are satisfied, but in the worst case is only second-order.

Krogstad [72] derived the method given by

$$\begin{array}{c|cccc} 0 & & & & \\ \frac{1}{2} & \frac{1}{2}\varphi_{1,2} & & & \\ \frac{1}{2} & \frac{1}{2}\varphi_{1,3} - \varphi_{2,3} & \varphi_{2,3} & & \\ 1 & \varphi_{1,4} - 2\varphi_{2,4} & \mathbf{0}_{N\times N} & 2\varphi_{2,4} & \\ \hline & \varphi_1 - 3\varphi_2 + 4\varphi_3 & 2\varphi_2 - 4\varphi_3 & 2\varphi_2 - 4\varphi_3 & 4\varphi_3 - \varphi_2 \end{array} . \tag{2.63}$$

It is usually also fourth-order accurate and has order three in the worst case.

The following five-stage method is due to Hochbruck and Ostermann [62]:

$$\begin{array}{c|ccccc} 0 & & & & & \\ \frac{1}{2} & \frac{1}{2}\varphi_{1,2} & & & & \\ \frac{1}{2} & \frac{1}{2}\varphi_{1,3} - \varphi_{2,3} & \varphi_{2,3} & & & \\ 1 & \varphi_{1,4} - 2\varphi_{2,4} & \varphi_{2,4} & \varphi_{2,4} & & \\ \frac{1}{2} & \frac{1}{2}\varphi_{1,5} - \frac{1}{4}\varphi_{2,5} - a_{5,2} & a_{5,2} & a_{5,2} & \frac{1}{4}\varphi_{2,5} - a_{5,2} & \\ \hline & \varphi_1 - 3\varphi_2 + 4\varphi_3 & \mathbf{0}_{N\times N} & \mathbf{0}_{N\times N} & -\varphi_2 + 4\varphi_3 & 4\varphi_2 - 8\varphi_3 \end{array} , \tag{2.64}$$

with $a_{5,2} = \frac{1}{2}\varphi_{2,5} - \varphi_{3,4} + \frac{1}{4}\varphi_{2,4} - \frac{1}{2}\varphi_{3,5}$. It has order four under certain mild assumptions.

2.4.2 The Action of φ_ℓ on Arbitrary Vectors

To simplify the notation in this section, let $\mathbf{L} = \tau_k \mathbf{L}_{k-1}$ and $\varphi = \varphi_\ell$ for some $k \geq 1, \ell \geq 0$. We very briefly review some methods used in practice to evaluate the product of $\varphi(\mathbf{L}) \in \mathbb{R}^{N\times N}$ with some arbitrary vector $\mathbf{w} \in \mathbb{R}^N$, where N is too large for $\varphi(\mathbf{L})$ to be computed explicitly and stored in full, or where it is impractical to first diagonalize $\mathbf{L}$.

Much has been written about the approximation of these products for large N;

see [63] and the references therein. Here we mention four of the most relevant approaches to performing these approximations, all of which also help to address the numerical cancellation error that would occur when computing φ directly using the recurrence relation (2.57).

If using a Rosenbrock-type scheme with $\mathbf{L}$ depending on $\mathbf{y}_{k-1}$, then we are interested in finding methods that lend themselves to calculating the derivatives of $\varphi(\mathbf{L}(\mathbf{y}_{k-1}, \mathbf{m}))\mathbf{w}$ with respect to $\mathbf{y}_{k-1}$ or $\mathbf{m}$. This will be explored in more detail in Section 4.4.2.

Krylov Subspace Methods

The Mth Krylov subspace with respect to a matrix $\mathbf{L}$ and a vector $\mathbf{w}$ is denoted by $\mathcal{K}_M(\mathbf{L}, \mathbf{w}) = \text{span}\{\mathbf{w}, \mathbf{L}\mathbf{w}, \ldots, \mathbf{L}^{M-1}\mathbf{w}\}$. Normalizing $\|\mathbf{w}\| = 1$, the Arnoldi process can be used to construct an orthonormal basis $\mathbf{V}_M \in \mathbb{C}^{N\times M}$ of $\mathcal{K}_M(\mathbf{L}, \mathbf{w})$ and an unreduced upper Hessenberg matrix $\mathbf{H}_M \in \mathbb{C}^{M\times M}$ satisfying the standard Krylov recurrence formula

$$\mathbf{L}\mathbf{V}_M = \mathbf{V}_M\mathbf{H}_M + h_{M+1,M}\mathbf{v}_{M+1}\mathbf{e}_M^T, \qquad \mathbf{V}_M^*\mathbf{V}_M = \mathbf{I}_M,$$

with $\mathbf{e}_M$ the Mth unit vector in $\mathbb{C}^N$. Using the orthogonality of $\mathbf{V}_M$, it can then be shown that

$$\varphi(\mathbf{L})\mathbf{w} \approx \mathbf{V}_M\varphi(\mathbf{H}_M)\mathbf{e}_1 \tag{2.65}$$

(see for instance [97]). It is assumed that $M \ll N$, so that $\varphi(\mathbf{H}_M)$ can be computed using standard methods such as diagonalization or Padé approximations.

There has been a lot of work on Krylov subspace methods for evaluating matrix functions, see for instance [34, 50] and the references therein. See in particular [61] for a discussion on Krylov subspace methods for matrix exponentials.

Polynomial Approximations

Polynomial methods approximate $\varphi(\mathbf{L})$ using some truncated polynomial series, for instance Taylor series (which is rarely used in this context), Chebyshev series for Hermitian or skew-Hermitian $\mathbf{L}$, Faber series for general $\mathbf{L}$, or Leja interpolants. See [63] and the references therein for a review of Chebyshev approximations and Leja interpolants in the context of exponential time differencing.

A polynomial approximation can generally be written in the form

$$\varphi(\mathbf{L})\mathbf{w} \approx \sum_{j=0}^{M} c_j \mathbf{L}^j \mathbf{w}, \tag{2.66}$$

although sometimes other forms are more suitable. For instance, in the case of Chebyshev polynomials it makes more sense to write

$$\varphi(\mathbf{L})\mathbf{w} \approx \sum_{j=0}^{M} c_j T_j(\mathbf{L})\mathbf{w} \tag{2.67}$$

if $\mathbf{L}$ is Hermitian or skew-Hermitian and the eigenvalues of $\mathbf{L}$ all lie inside $[-1, 1]$. The $T_j(\mathbf{L})$ satisfy the recurrence relation

$$T_{j+1}(\mathbf{L}) = 2\mathbf{L}T_j(\mathbf{L}) - T_{j-1}(\mathbf{L}), \qquad j = 1, 2, \ldots$$

initialized by $T_0(\mathbf{L}) = \mathbf{I}$ and $T_1(\mathbf{L}) = \mathbf{L}$.

Rational Approximations

The function $\varphi(z)$ can be estimated to arbitrary order using rational approximations

$$\varphi(z) \approx \varphi_{[m,n]}(z) = \sum_{i=0}^{m} a_i z^i \Big/ \sum_{k=0}^{n} b_{k-1} z^k = p_m(z)/q_n(z).$$

The polynomials $p_m(z)$ and $q_n(z)$ can be found using either Padé approximations or by using the Carathéodory-Fejér (CF) method on the negative real line, which is an efficient method for constructing near-best rational approximations. It has been applied to the problem of approximating φ-functions in [109]. The Padé approximation works for general matrices $\mathbf{L}$, but the CF method used in [109] works only if z is negative and real, so that $\mathbf{L}$ must be symmetric negative definite.

For large N it will generally be too expensive to evaluate $\varphi_{[m,n]}(\mathbf{L})$ in this form. But suppose we have that $m \leq n$, q_n has n distinct roots denoted by $s_1, \ldots, s_n$, and p_m and q_n have no roots in common. Then we can find a partial fraction expansion

$$\varphi_{[m,n]}(z) = c_0 + \sum_{i=1}^{n} \frac{c_{k-1}}{s_i - z},$$

where c_0 is some constant and $c_{k-1} = \operatorname{Res}\left[\varphi_{[m,n]}(z), \rho_{k-1}\right]$. In practice one would find these coefficients simply by clearing the denominators.

The product of $\varphi(\mathbf{L})$ with some vector $\mathbf{w}$ therefore is

$$\varphi(\mathbf{L})\,\mathbf{w} \approx \varphi_{[m,n]}(\mathbf{L})\,\mathbf{w} = c_0\,\mathbf{w} + \sum_{i=1}^{n} c_{k-1}\left(s_i\mathbf{I} - \mathbf{L}\right)^{-1}\mathbf{w}. \tag{2.68}$$

See [109] for a discussion on how a common set of poles can be used for the evaluation of different φ_ℓ. While this approach requires a higher degree n in the rational approximation to achieve a given accuracy, in the use of exponential integration this would still lead to a more efficient method overall since the same computations can be used to evaluate different φ_ℓ.

Contour Integration

The last approach we consider is based on the Cauchy integral formula

$$\phi(z) = \frac{1}{2\pi i}\int_\Gamma \frac{\phi(s)}{s - z}\,ds$$

for a fixed value of z, where $\phi(z)$ is some arbitrary function and Γ is a contour in the complex plane that encloses z and is well-separated from 0. This formula still holds when replacing z by some general matrix $\mathbf{L}$, so that

$$\phi(\mathbf{L}) = \frac{1}{2\pi i} \int_{\Gamma} \phi(s) \left(s\mathbf{I} - \mathbf{L}\right)^{-1} ds, \tag{2.69}$$

where Γ can be any contour that encloses all the eigenvalues of $\mathbf{L}$. The integral is then approximated using some quadrature rule.

There is some freedom in choosing the contour integral and the quadrature rule. Kassam and Trefethen [68] proposed the contour integral approach to circumvent the cancellation error in (2.57). For convenience, they let Γ simply be a circle in the complex plane that is large enough to enclose all the eigenvalues of $\mathbf{L}$, and then used the trapezoidal rule for the approximation. If $\mathbf{L}$ is real, then one can additionally simplify the calculations by considering only points on the upper half of a circle with its center on the real axis, and taking the real part of the result. Discretizing the contour using M points s_i and using the trapezoidal rule to evaluate (2.69), we have

$$\phi(\mathbf{L}) \approx \frac{1}{M} \Re \left(\sum_{i=1}^{M} s_i \phi(s_i) \left(s_i\mathbf{I} - \mathbf{L}\right)^{-1} \right), \tag{2.70}$$

where M must be chosen large enough to give a good approximation. Then let $\phi = \varphi$ and multiply by $\mathbf{w}$ to get an approximation of the product $\varphi(\mathbf{L})\mathbf{w}$.

Since the same contour integral will be used throughout the procedure, the quadrature points s_i remain the same for all φ-functions and one can therefore use the same solutions $\mathbf{v}_i = \left(s_i\mathbf{I} - \mathbf{L}\right)^{-1} \mathbf{w}$ of the resolvent systems when computing the product of different φ-functions with some vector $\mathbf{w}$.

A contour integral that specifically applies to φ-functions is

$$\varphi_\ell(z) = \frac{1}{2\pi i} \int_{\Gamma} \frac{e^s}{s^\ell} \frac{1}{s - z} ds.$$

Again, different contour integrals and quadrature rules can be used. If z is on the negative real line or close to it, then we can use a Hankel contour (see also [122]). Letting $z = \mathbf{L}$ and using the trapezoidal rule, we get

$$\varphi_\ell(\mathbf{L}) \approx \frac{1}{M} \Re \left(\sum_{i=1}^{M} \frac{e^{s_i}}{s_i^\ell} \left(s_i \mathbf{I} - \mathbf{L}\right)^{-1} \right), \tag{2.71}$$

for quadrature points s_i on Γ. This integral representation has the advantage that the integrand is exponentially decaying and therefore fewer quadrature points need to be used [63, 109].

Incidentally, (2.70), (2.71) and (2.68) all require an efficient procedure for solving linear systems of the form

$$(s_i \mathbf{I} - \mathbf{L})\, \mathbf{v}_i = \mathbf{w}.$$

This can be achieved, for instance, using sparse direct solvers or preconditioned Krylov subspace methods. Solving M different linear systems might seem prohibitive, but when using Krylov methods one has the advantage that $\mathcal{K}_M(\mathbf{L}, \mathbf{w}) = \mathcal{K}_M(s\mathbf{I} - \mathbf{L}, \mathbf{w})$ for all $s \in \mathbb{C}^N$, so that the same Krylov subspace can theoretically be used for all s_i. The computation also allows for parallelization since each system can be solved independently.

Chapter 3

Derivatives of the Misfit Function

The time-stepping equation

$$\mathbf{t}(\overline{\mathbf{y}}\,(\mathbf{p})\,, \mathbf{m}, \mathbf{y}_0) = \mathbf{S}\,\overline{\mathbf{q}}(\mathbf{s}), \tag{3.1}$$

is now used in conjunction with the adjoint method to systematically find the procedures for computing the action of the sensitivity matrix $\mathbf{J} = \frac{\partial \mathbf{d}}{\partial \mathbf{p}}$ and its transpose, as well as the action of the Hessian of the misfit function $\mathrm{M} = \mathrm{M}\,(\mathbf{d}(\mathbf{p}))$ with respect to the parameters $\mathbf{p} = \left[\mathbf{m}^\top, \mathbf{y}_0^\top, \mathbf{s}^\top\right]^\top$. From now on we explicitly include the dependence of $\mathbf{y}$ on the component(s) of $\mathbf{p}$ we are currently interested in.

We assume that the data $\mathbf{d}$ depends on the forward solution $\mathbf{y}$ in some known and (twice-)differentiable way, so that the computations of $\dfrac{\partial \mathbf{d}}{\partial \mathbf{y}}$ (and $\dfrac{\partial^2 \mathbf{d}}{\partial \mathbf{y} \partial \mathbf{y}}$) do not present a significant challenge. In most applications the data at a given time level k, $\mathbf{d}_k$, depends only on $\mathbf{y}_k$, i.e. the data at a given point in time depends on the solution only at that time, and we will assume that this is the case here too. The modifications are straightforward in case the data depends on the solution from previous time-steps, for instance if the data consists of some time-averaged measurements of the solution.

In this chapter we let $\mathbf{y}_k$ include, for ease of notation, the internal stages at the kth time level when the time-stepping method is an RK-type method. In this case we abuse notation slightly and let N denote the length of $\mathbf{y}_k$ when in fact the actual length is $(s+1)N$, with s denoting the number of stages of the RK method. This should not cause any serious confusion and we will not further elaborate on resulting implementation details.

3.1 The Sensitivity Matrix

The sensitivity matrix (or Jacobian) is an $N_{\mathbf{d}} \times N_{\mathbf{p}}$ matrix defined by

$$\mathbf{J} := \frac{\partial \mathbf{d}}{\partial \mathbf{p}} = \begin{bmatrix} \frac{\partial \mathbf{d}}{\partial \mathbf{m}} & \frac{\partial \mathbf{d}}{\partial \mathbf{y}_0} & \frac{\partial \mathbf{d}}{\partial \mathbf{s}} \end{bmatrix} = \frac{\partial \mathbf{d}}{\partial \overline{\mathbf{y}}} \begin{bmatrix} \frac{\partial \overline{\mathbf{y}}}{\partial \mathbf{m}} & \frac{\partial \overline{\mathbf{y}}}{\partial \mathbf{y}_0} & \frac{\partial \overline{\mathbf{y}}}{\partial \mathbf{s}} \end{bmatrix}, \tag{3.2}$$

where we have used the chain rule. The components of each column are the partial derivatives of every data point with respect to one of the unknown parameters; similarly, each row is the partial derivative of a given data point with respect to all of the unknown parameters. Sensitivity analysis involves the investigation of the structure of $\mathbf{J}$ to determine the relative influence of various parameters on the measured data.

As mentioned in the introduction of this chapter, we consider the computation of $\frac{\partial \mathbf{d}}{\partial \overline{\mathbf{y}}}$ to be known. We therefore focus on the computation of the derivative of the forward solution with respect to the parameters and use the adjoint method to find computationally tractable expressions for these derivatives:

- *Derivative of* $\overline{\mathbf{y}}$ *with respect to* $\mathbf{m}$:

  Taking the derivative with respect to the model parameters on both sides of the discrete time-stepping system (1.2), we get

$$\mathbf{0}_{N \times N_{\mathbf{m}}} = \frac{\partial}{\partial \mathbf{m}} \left(\mathbf{t} \left(\overline{\mathbf{y}}(\mathbf{m}), \mathbf{m} \right) \right) = \frac{\partial \mathbf{t} \left(\mathbf{m} \right)}{\partial \mathbf{m}} + \frac{\partial \mathbf{t} \left(\overline{\mathbf{y}} \right)}{\partial \overline{\mathbf{y}}} \frac{\partial \overline{\mathbf{y}} \left(\mathbf{m} \right)}{\partial \mathbf{m}} \tag{3.3}$$

  and therefore

$$\frac{\partial \overline{\mathbf{y}}}{\partial \mathbf{m}} = - \left(\frac{\partial \mathbf{t}}{\partial \overline{\mathbf{y}}} \right)^{-1} \frac{\partial \mathbf{t}}{\partial \mathbf{m}}. \tag{3.4}$$

- *Derivative of* $\overline{\mathbf{y}}$ *with respect to* $\mathbf{y}_0$:

  Now taking the derivative with respect to the initial condition $\mathbf{y}_0$ on both sides of (1.2) leads to:

$$\mathbf{0}_{N \times N_{\mathbf{y}_0}} = \frac{\partial}{\partial \mathbf{y}_0} \mathbf{t} \left(\overline{\mathbf{y}}(\mathbf{y}_0), \mathbf{y}_0 \right) = \frac{\partial \mathbf{t} \left(\mathbf{y}_0 \right)}{\partial \mathbf{y}_0} + \frac{\partial \mathbf{t} \left(\overline{\mathbf{y}} \right)}{\partial \overline{\mathbf{y}}} \frac{\partial \overline{\mathbf{y}} \left(\mathbf{y}_0 \right)}{\partial \mathbf{y}_0}, \tag{3.5}$$

so that

$$\frac{\partial \overline{\mathbf{y}}}{\partial \mathbf{y}_0} = -\left(\frac{\partial \mathbf{t}}{\partial \overline{\mathbf{y}}}\right)^{-1} \frac{\partial \mathbf{t}}{\partial \mathbf{y}_0}. \tag{3.6}$$

- *Derivative of* $\overline{\mathbf{y}}$ *with respect to* $\mathbf{s}$:

  Finally, we differentiate (1.2) with respect to the source parameters $\mathbf{s}$ on both sides:

$$\mathbf{S}\frac{\partial \mathbf{q}}{\partial \mathbf{s}} = \frac{\partial}{\partial \mathbf{s}}\mathbf{t}\left(\overline{\mathbf{y}}(\mathbf{s})\right) = \frac{\partial \mathbf{t}\left(\overline{\mathbf{y}}\right)}{\partial \overline{\mathbf{y}}}\frac{\partial \overline{\mathbf{y}}\left(\mathbf{s}\right)}{\partial \mathbf{s}} \tag{3.7}$$

  and hence

$$\frac{\partial \overline{\mathbf{y}}}{\partial \mathbf{s}} = \left(\frac{\partial \mathbf{t}}{\partial \overline{\mathbf{y}}}\right)^{-1}\left(\mathbf{S}\frac{\partial \mathbf{q}}{\partial \mathbf{s}}\right). \tag{3.8}$$

It follows that

$$\mathbf{J} = \frac{\partial \mathbf{d}}{\partial \overline{\mathbf{y}}}\left(\frac{\partial \mathbf{t}}{\partial \overline{\mathbf{y}}}\right)^{-1}\left[-\frac{\partial \mathbf{t}}{\partial \mathbf{m}} \quad -\frac{\partial \mathbf{t}}{\partial \mathbf{y}_0} \quad \mathbf{S}\frac{\partial \mathbf{q}}{\partial \mathbf{s}}\right]. \tag{3.9}$$

The derivatives of $\mathbf{t}$ are derived in Appendix A and references to the relevant expressions are given in Chapter 4. The solution of the linearized forward problem

$$\boldsymbol{\xi} = \frac{\partial \mathbf{t}}{\partial \overline{\mathbf{y}}}^{-1}\mathbf{q} \tag{3.10}$$

is given in Chapter 5 for each time-stepping method. It is neither desirable nor necessary to compute the sensitivity matrix $\mathbf{J}$ explicitly, what one really needs is to be able to quickly compute the product of $\mathbf{J}$ with some arbitrary vector $\mathbf{w_p}$ of length $N_\mathbf{p}$, and we outline this procedure in Algorithm 3.1. To simplify the notation we have let $\mathbf{R} = \mathbf{S}\dfrac{\partial \mathbf{q}}{\partial \mathbf{s}}$, which has dimensions $NK \times N_\mathbf{s}$. $\mathbf{R}_{k,:}$ represents the kth $N \times N_\mathbf{s}$ block-row of $\mathbf{R}$.

Algorithm 3.1. Computing $\mathbf{u} = \mathbf{J}\,\mathbf{w_p}$

Let $\mathbf{w_p} = \begin{bmatrix}\mathbf{w}_\mathbf{m}^\top & \mathbf{w}_{\mathbf{y}_0}^\top & \mathbf{w}_\mathbf{s}^\top\end{bmatrix}^\top$ *be an arbitrary vector of length* $N_\mathbf{m} + N_\mathbf{s} + N_{\mathbf{y}_0}$. *The product* $\mathbf{u} = \mathbf{J}\,\mathbf{w_p}$ *is computed as follows:*

1. *If the forward solution is not available already, solve for and store* $\overline{\mathbf{y}}$.

2. *For* $k = 1, \cdots, K$*:*

   (a) *Compute the source term* $\mathbf{q}_k = \mathbf{R}_{k,:}\mathbf{w}_\mathbf{s} - \dfrac{\partial \mathbf{t}_k}{\partial \mathbf{m}}\mathbf{w}_\mathbf{m} - \dfrac{\partial \mathbf{t}_k}{\partial \mathbf{y}_0}\mathbf{w}_{\mathbf{y}_0}$*. Chapter 4 outlines how the last two terms on the right-hand side can be computed.*

   (b) *Solve for* $\boldsymbol{\xi}_k$ *using one step of the linearized forward time-stepping method* $\dfrac{\partial \mathbf{t}}{\partial \overline{\mathbf{y}}}$ *as shown in Chapter 5, with* $\mathbf{q}_k$ *acting as the source term and* $\boldsymbol{\xi}_0 = \mathbf{0}_{N\times 1}$*.*

   (c) *Compute* $\mathbf{u}_k = \dfrac{\partial \mathbf{d}_k}{\partial \overline{\mathbf{y}}_k}\boldsymbol{\xi}_k$*.*

3. *Set* $\mathbf{u} = \begin{bmatrix}\mathbf{u}_1^\top & \cdots & \mathbf{u}_K^\top\end{bmatrix}^\top$*.*

In the following two sections we will also need to compute the action of the transpose of the sensitivity matrix

$$\mathbf{J}^\top = \begin{bmatrix} -\dfrac{\partial \mathbf{t}}{\partial \mathbf{m}}^\top \\ -\dfrac{\partial \mathbf{t}}{\partial \mathbf{y}_0}^\top \\ \dfrac{\partial \mathbf{q}}{\partial \mathbf{s}}^\top \mathbf{S}^\top \end{bmatrix} \left(\frac{\partial \mathbf{t}}{\partial \overline{\mathbf{y}}}\right)^{-\top} \frac{\partial \mathbf{d}}{\partial \overline{\mathbf{y}}}^\top \tag{3.11}$$

on some vector $\mathbf{w}_\mathbf{d}$ of length $N_\mathbf{d}$. This will involve the solution of the *adjoint problem*

$$\boldsymbol{\lambda} = \frac{\partial \mathbf{t}}{\partial \overline{\mathbf{y}}}^{-\top} \boldsymbol{\theta} \tag{3.12}$$

where $\boldsymbol{\lambda}$ is the *adjoint solution* and $\boldsymbol{\theta}$ is the *adjoint source*. Solving the adjoint problem is discussed in detail in Chapter 6 for each time-stepping method. The transposes of the derivatives of $\mathbf{t}$ with respect to $\mathbf{m}$ and $\mathbf{y}_0$ are given in Chapter 4 and derived in Appendix A. Algorithm 3.2 presents the procedure for computing the product of $\mathbf{J}^\top$ with $\mathbf{w}_\mathbf{d}$.

Algorithm 3.2. Computing $\mathbf{u} = \mathbf{J}^\top \mathbf{w_d}$

Let $\mathbf{w_d} = \begin{bmatrix} \mathbf{w}_1^\top & \cdots & \mathbf{w}_K^\top \end{bmatrix}^\top$ *be an arbitrary vector of length* $N_\mathbf{d} = \sum_{k=1}^{K} N_{\mathbf{d}_k}$. *The product* $\mathbf{u} = \mathbf{J}^\top \mathbf{w_d}$ *is computed as follows:*

1. *Set* $\mathbf{u_m} = \mathbf{0}_{N_\mathbf{m}\times 1}$, $\mathbf{u_s} = \mathbf{0}_{N_\mathbf{s}\times 1}$ *and* $\mathbf{u}_{\mathbf{y}_0} = \mathbf{0}_{N_{\mathbf{y}_0}\times 1}$. *These will store* $\mathbf{u}$.

2. *If the forward solution is not available already, solve for and store* $\overline{\mathbf{y}}$.

3. *For* $k = K, \cdots, 1$:

   (a) *Compute* $\boldsymbol{\theta}_k = \dfrac{\partial \mathbf{d}_k}{\partial \mathbf{y}_k}^\top \mathbf{w}_k$.

   (b) *Solve for* $\boldsymbol{\lambda}_k$ *using one step of the adjoint time-stepping method, with* $\boldsymbol{\theta}_k$ *acting as the adjoint source term and* $\boldsymbol{\lambda}_{K+1} = \mathbf{0}_{N\times 1}$. *See Chapter 6.*

   (c) *Update* $\mathbf{u_s} = \mathbf{u_s} + \mathbf{R}_{k,:}^\top \boldsymbol{\lambda}_k$, *and (see Chapter 4)* $\mathbf{u_m} = \mathbf{u_m} - \dfrac{\partial \mathbf{t}_k}{\partial \mathbf{m}}^\top \boldsymbol{\lambda}_k$ *and* $\mathbf{u}_{\mathbf{y}_0} = \mathbf{u}_{\mathbf{y}_0} - \dfrac{\partial \mathbf{t}_k}{\partial \mathbf{y}_0}^\top \boldsymbol{\lambda}_k$.

4. *Set* $\mathbf{u} = \begin{bmatrix} \mathbf{u_m}^\top & \mathbf{u}_{\mathbf{y}_0}^\top & \mathbf{u_s}^\top \end{bmatrix}^\top$.

Note the following:

- The forward solution $\overline{\mathbf{y}}$ is required to compute both the action of $\mathbf{J}$ and its transpose.

- The product $\mathbf{u} = \mathbf{J}\,\mathbf{w_p}$ requires the solution of the linearized forward problem, and the product $\mathbf{u} = \mathbf{J}^\top \mathbf{w_d}$ requires the adjoint solution. This is in addition to the computation of $\overline{\mathbf{y}}$ (if $\overline{\mathbf{y}}$ is not already available). If $\overline{\mathbf{y}}$ is already available, computing the action of the sensitivity matrix, or its transpose, is about as expensive as computing $\overline{\mathbf{y}}$, not counting any extra computational effort that might be required to prepare the source terms.

- As mentioned, in the case of RK methods the forward solution $\mathbf{y}_k$ at time-step k includes the internal stages. There is further abuse in notation, as $\boldsymbol{\theta}_k$, $\mathbf{v}_k$ and

$\boldsymbol{\lambda}_k$ are assumed to also include their respective internal stages for that time step, therefore these vectors are also of length $(s+1)N$. This does not apply to the initial condition $\mathbf{v}_0$ and the final condition $\boldsymbol{\lambda}_{K+1}$, which are both only of length N.

- We do not address here the derivatives of the source $\mathbf{q}$ with respect to the source parameters $\mathbf{s}$ since the source terms are application-dependent and are expected to be independent of the time-stepping method.

3.2 The Gradient

The gradient of M with respect to $\mathbf{p}$, $\nabla_{\mathbf{p}}\mathrm{M}$, is used by all iterative gradient-based optimization methods, such as steepest descent, nonlinear conjugate gradient methods, quasi-Newton methods such as BFGS, etc. It gives the direction in which the directional derivative of M has the largest value, i.e. the direction in which the largest increase in the value of M is to be expected at a given point in parameter space. To compute the gradient of the misfit function, we use the chain rule to immediately get

$$\nabla_{\mathbf{p}}\mathrm{M} = \frac{\partial\,\mathbf{d}}{\partial\,\mathbf{p}}^{\top}\nabla_{\mathbf{d}}\mathrm{M} = \mathbf{J}^{\top}\nabla_{\mathbf{d}}\mathrm{M}.$$

The gradient $\nabla_{\mathbf{p}}\mathrm{M}$ is therefore easily obtained using Algorithm 3.2 with $\mathbf{w} = \nabla_{\mathbf{d}}\mathrm{M}$. The gradient of M with respect to $\mathbf{d}$ depends on the misfit function that is being used and is independent of the time-stepping method. We discuss some misfit functions that might be of interest, and their derivatives with respect to $\mathbf{d}$, in Chapter 7.

3.3 The Hessian

The Newton method is a gradient-based optimization procedure that, in addition to the gradient, requires the action of the full Hessian of M with respect to $\mathbf{p}$

$$\mathcal{H}_{\mathrm{M}}\mathbf{w} := \frac{\partial^2 \mathrm{M}}{\partial \mathbf{p} \partial \mathbf{p}}\mathbf{w} = \nabla_{\mathbf{p}}\left(\nabla_{\mathbf{p}}\mathrm{M}^\top \mathbf{w}\right), \tag{3.13}$$

on an arbitrary vector $\mathbf{w} = \begin{bmatrix}\mathbf{w}_{\mathbf{m}}^\top & \mathbf{w}_{\mathbf{y}_0}^\top & \mathbf{w}_{\mathbf{s}}^\top\end{bmatrix}^\top$ of length $N_{\mathbf{p}} = N_{\mathbf{m}} + N_{\mathbf{y}_0} + N_{\mathbf{s}}$. The adjoint method can again be used to compute the product $\mathcal{H}_{\mathrm{M}}\mathbf{w}$.

The Hessian has the following structure

$$\mathcal{H}_{\mathrm{M}} = \frac{\partial^2 \mathrm{M}}{\partial \mathbf{p} \partial \mathbf{p}} = \begin{bmatrix} \frac{\partial^2 \mathrm{M}}{\partial \mathbf{m} \partial \mathbf{m}} & \frac{\partial^2 \mathrm{M}}{\partial \mathbf{m} \partial \mathbf{y}_0} & \frac{\partial^2 \mathrm{M}}{\partial \mathbf{m} \partial \mathbf{s}} \\ \frac{\partial^2 \mathrm{M}}{\partial \mathbf{y}_0 \partial \mathbf{m}} & \frac{\partial^2 \mathrm{M}}{\partial \mathbf{y}_0 \partial \mathbf{y}_0} & \frac{\partial^2 \mathrm{M}}{\partial \mathbf{y}_0 \partial \mathbf{s}} \\ \frac{\partial^2 \mathrm{M}}{\partial \mathbf{s} \partial \mathbf{m}} & \frac{\partial^2 \mathrm{M}}{\partial \mathbf{s} \partial \mathbf{y}_0} & \frac{\partial^2 \mathrm{M}}{\partial \mathbf{s} \partial \mathbf{s}} \end{bmatrix} \tag{3.14}$$

and the presence of the cross-terms in (3.14) makes the application of the adjoint method quite cumbersome, so we have left the calculations out of the main text and put it in Appendix B.

We use the following notation: if $\mathbf{x}$, $\mathbf{y}$, $\mathbf{z}$ and $\mathbf{w}$ are some vector quantities of appropriate size, we let

$$\frac{\partial^2 \mathbf{x}}{\partial \mathbf{y} \partial \mathbf{z}}\mathbf{w} = \frac{\partial}{\partial \mathbf{y}}\left(\frac{\partial \mathbf{x}}{\partial \mathbf{z}}\mathbf{w}\bigg|_{\mathbf{w}}\right),$$

where the $\cdot|_{\mathbf{w}}$ on the right-hand side means that $\mathbf{w}$ is taken to be fixed when performing the differentiation with respect to $\mathbf{y}$. Note that $\frac{\partial^2 \mathbf{x}}{\partial \mathbf{y} \partial \mathbf{z}}$ is actually a three-dimensional tensor and its product with a vector is ambiguously defined, so it is important to keep our convention in mind.

The expressions that we found are (see (B.15))

$$
\begin{aligned}
\mathcal{H}_{\mathrm{M}}\mathbf{w_p} &= \frac{\partial^2 \mathrm{M}}{\partial \mathbf{p}\partial \mathbf{p}}\mathbf{w_p} \\
&= \begin{bmatrix} -\dfrac{\partial \mathbf{t}}{\partial \mathbf{m}}^\top \\ -\dfrac{\partial \mathbf{t}}{\partial \mathbf{y}_0}^\top \\ \dfrac{\partial \mathbf{q}}{\partial \mathbf{s}}^\top \mathbf{S}^\top \end{bmatrix} \boldsymbol{\mu} - \begin{bmatrix} \dfrac{\partial^2 \mathbf{t}}{\partial \mathbf{m}\partial \mathbf{m}}\mathbf{w_m}^\top + \dfrac{\partial^2 \mathbf{t}}{\partial \mathbf{m}\partial \mathbf{y}_0}\mathbf{w_{y_0}}^\top + \dfrac{\partial^2 \mathbf{t}}{\partial \mathbf{m}\partial \mathbf{y}}\mathbf{x}^\top \\ \dfrac{\partial^2 \mathbf{t}}{\partial \mathbf{y}_0\partial \mathbf{m}}\mathbf{w_m}^\top + \dfrac{\partial^2 \mathbf{t}}{\partial \mathbf{y}_0\partial \mathbf{y}_0}\mathbf{w_{y_0}}^\top + \dfrac{\partial^2 \mathbf{t}}{\partial \mathbf{y}_0\partial \mathbf{y}}\mathbf{x}^\top \\ -\dfrac{\partial^2 \mathbf{q}}{\partial \mathbf{s}\partial \mathbf{s}}\mathbf{w_s}^\top \mathbf{S}^\top \end{bmatrix} \boldsymbol{\lambda}
\end{aligned}
$$

with

$$
\begin{aligned}
\boldsymbol{\mu} = \frac{\partial \mathbf{t}}{\partial \mathbf{y}}^{-\top} \Bigg(& \frac{\partial \mathbf{d}}{\partial \mathbf{y}}^\top \frac{\partial^2 \mathrm{M}}{\partial \mathbf{d}\partial \mathbf{d}} \frac{\partial \mathbf{d}}{\partial \mathbf{y}}\mathbf{x} + \left(\frac{\partial^2 \mathbf{d}}{\partial \mathbf{y}\partial \mathbf{y}}\mathbf{x}\right)^\top \nabla_{\mathbf{d}}\mathrm{M} + \\
& - \left(\frac{\partial^2 \mathbf{t}}{\partial \mathbf{y}\partial \mathbf{m}}\mathbf{w_m}^\top + \frac{\partial^2 \mathbf{t}}{\partial \mathbf{y}\partial \mathbf{y}_0}\mathbf{w_{y_0}}^\top + \frac{\partial^2 \mathbf{t}}{\partial \mathbf{y}\partial \mathbf{y}}\mathbf{x}^\top\right)\boldsymbol{\lambda}\Bigg),
\end{aligned}
\tag{3.15a}
$$

and

$$
\mathbf{x} = \frac{\partial \mathbf{y}}{\partial \mathbf{p}}\mathbf{w_p} = \frac{\partial \mathbf{t}}{\partial \mathbf{y}}^{-1}\left(\mathbf{S}\frac{\partial \mathbf{q}}{\partial \mathbf{s}}\mathbf{w_s} - \frac{\partial \mathbf{t}}{\partial \mathbf{m}}\mathbf{w_m} - \frac{\partial \mathbf{t}}{\partial \mathbf{y}_0}\mathbf{w_{y_0}}\right). \tag{3.15b}
$$

As far as we know the derivation of the Hessian with respect to three distinct sets of parameters ($\mathbf{m}$, $\mathbf{s}$ and $\mathbf{y}_0$) using the discrete adjoint method is new.

We do not give an algorithm detailing the computation of the Hessian, but we have derived the expressions of the second derivatives of $\mathbf{t}$ for each of the time-stepping methods, except for ETDRK. The derivations are presented in Appendix A, and the relevant equations and pages are pointed to in Chapter 4 for ease of reference.

The computation of the action of the Hessian requires the availability of the forward solution $\mathbf{y}$, the computation of two adjoint solutions $\boldsymbol{\lambda}$ and $\boldsymbol{\mu}$, and the computation of one linearized forward solution $\mathbf{x}$. The implementation of (3.15) can be done in such a way that only $\mathbf{y}$ and $\mathbf{x}$ need to be stored in full.

The full Hessian is actually rarely used in practice though, in large part because of the added expense of computing the second derivatives and the difficulty in coding

them. It will also generally not lead to improved convergence rates in numerical minimization procedures unless one is fairly close to the minimum already, so using the full Hessian might not always be beneficial even if an efficient implementation is at hand.

As a result, the action of the Hessian is often approximated, for instance in the case of the (generalized) Gauss-Newton method we instead use the symmetric positive semi-definite approximation

$$\mathcal{H}_{\mathrm{M}}\mathbf{w}_{\mathbf{p}} \approx \mathcal{H}_{\mathrm{M}}^{\mathrm{GN}}\mathbf{w}_{\mathbf{p}} = \mathbf{J}^{\top}\frac{\partial^2 \mathrm{M}}{\partial \mathbf{d}\partial \mathbf{d}}\mathbf{J}\,\mathbf{w}_{\mathbf{p}}, \tag{3.16}$$

which is obtained by omitting all the second derivatives of $\mathbf{t}$ and $\mathbf{d}$. A related method is the Levenberg-Marquardt method, where we add a damping term $\delta > 0$ to get a symmetric positive definite approximation

$$\mathcal{H}_{\mathrm{M}}\mathbf{w}_{\mathbf{p}} \approx \mathcal{H}_{\mathrm{M}}^{\mathrm{LM}}\mathbf{w}_{\mathbf{p}} = \left(\mathbf{J}^{\top}\frac{\partial^2 \mathrm{M}}{\partial \mathbf{d}\partial \mathbf{d}}\mathbf{J} + \delta\,\mathbf{I}_{N_{\mathbf{p}}}\right)\mathbf{w}_{\mathbf{p}}. \tag{3.17}$$

Both of these approximations are computed using Algorithm 3.1 followed by Algorithm 3.2, so that in addition to $\mathbf{y}$ we will have to compute one linearized forward solution and one adjoint solution.

Chapter 4

Derivatives of $\mathbf{t}$

As we saw in the previous chapter, the discrete adjoint method requires the derivatives of the time-stepping vector $\mathbf{t}(\overline{\mathbf{y}}, \mathbf{m}, \mathbf{y}_0)$ with respect to $\overline{\mathbf{y}}$, $\mathbf{m}$ and $\mathbf{y}_0$. Only the first derivatives of $\mathbf{t}$ are required if computing the gradient or the action of the sensitivity matrix, but when computing the action of the Hessian the second derivatives are required as well.

The expressions for all of these derivatives are of course highly dependent on the time-stepping method being used. We are in fact not so much interested in the expressions of the derivative matrices themselves, but on the action that they, and their transposes, have on arbitrary vectors. The derivations can be quite technical, so to aid the readability of this thesis we have opted to include them in Appendix A. In this chapter, for ease of reference, we simply present the page and equation numbers where expressions that assist in computing these derivatives and products can be found for each of the time-stepping methods. We emphasize that even though the derivations of the derivative terms are in the appendix, they form an important part of the contribution of this thesis.

Due to space constraints we do not give the products of the second derivatives with arbitrary vectors, but the expressions of the derivatives themselves that we have found will help the interested reader in the implementation.

Recall that for RK-type methods we defined the solution vector to be (with a small change of notation)

$$\overline{\mathbf{y}} = \begin{bmatrix} \widehat{\mathbf{y}}_1^\top & \cdots & \widehat{\mathbf{y}}_K^\top \end{bmatrix}^\top$$

where $\widehat{\mathbf{y}}_k = \begin{bmatrix} \mathbf{Y}_k^\top & \mathbf{y}_k^\top \end{bmatrix}^\top$. $\mathbf{Y}_k$ are the internal stages of the kth time-step. In Chapter

3 we abused notation slightly by letting $\mathbf{y}_k$ also include the internal stages of the solution at the kth time-step, so that what we were actually referring to was $\widehat{\mathbf{y}}_k$. We did this for notational convenience, but from now on we will keep $\widehat{\mathbf{y}}_k$ and $\mathbf{y}_k$ distinct.

4.1 Regular Time-Stepping Methods

4.1.1 Linear Multistep Methods

References to the first derivatives of $\mathbf{t}$ are given in Table 4.1, and references to the second derivatives can be found in Table 4.2.

We let $\overline{\mathbf{w}} = \begin{bmatrix} \mathbf{w}_1^\top & \cdots & \mathbf{w}_K^\top \end{bmatrix}^\top$ be an arbitrary vector of length KN, $\mathbf{w_m}$ an arbitrary vector of length $N_\mathbf{m}$, and $\mathbf{w}_0$ an arbitrary vector of length N.

Derivative	**Product**	**Page**	**Equation**
$\dfrac{\partial \mathbf{t}}{\partial \overline{\mathbf{y}}}$	$\dfrac{\partial \mathbf{t}_k}{\partial \mathbf{y}}\overline{\mathbf{w}}$	p.177	(A.4)
	$\dfrac{\partial \mathbf{t}_k}{\partial \mathbf{y}_j}^\top \mathbf{w}_k$	p.177	(A.5)
$\dfrac{\partial \mathbf{t}}{\partial \mathbf{m}}$	$\dfrac{\partial \mathbf{t}_k}{\partial \mathbf{m}}\mathbf{w_m}$	p.178	(A.10)
	$\dfrac{\partial \mathbf{t}_k}{\partial \mathbf{m}}^\top \mathbf{w}_k$	p.178	(A.11)
$\dfrac{\partial \mathbf{t}}{\partial \mathbf{y}_0}$	$\dfrac{\partial \mathbf{t}_k}{\partial \mathbf{y}_0}\mathbf{w}_0$	p.178	(A.15)
	$\dfrac{\partial \mathbf{t}_k}{\partial \mathbf{y}_0}^\top \mathbf{w}_k$	p.179	(A.16)

Table 4.1: References to the first derivatives of $\mathbf{t}$ for linear multistep methods.

$\frac{\partial^2 \mathbf{t}}{\partial \mathbf{m} \partial \overline{\mathbf{y}}} \overline{\mathbf{w}}$ p. 179, (A.18)	$\frac{\partial^2 \mathbf{t}}{\partial \mathbf{m} \partial \mathbf{m}} \mathbf{w}$ p. 181, (A.26)	$\frac{\partial^2 \mathbf{t}}{\partial \mathbf{m} \partial \mathbf{y}_0} \mathbf{w}_0$ p. 184, (A.36)
$\frac{\partial^2 \mathbf{t}}{\partial \mathbf{y}_0 \partial \overline{\mathbf{y}}} \overline{\mathbf{w}}$ p. 180, (A.21)	$\frac{\partial^2 \mathbf{t}}{\partial \mathbf{y}_0 \partial \mathbf{m}} \mathbf{w}$ p. 182, (A.29)	$\frac{\partial^2 \mathbf{t}}{\partial \mathbf{y}_0 \partial \mathbf{y}_0} \mathbf{w}_0$ p. 184, (A.39)
$\frac{\partial^2 \mathbf{t}}{\partial \overline{\mathbf{y}} \partial \overline{\mathbf{y}}} \overline{\mathbf{w}}$ *p.* 180, (A.22)	$\frac{\partial^2 \mathbf{t}}{\partial \overline{\mathbf{y}} \partial \mathbf{m}} \mathbf{w}$ p. 182, (A.32)	$\frac{\partial^2 \mathbf{t}}{\partial \overline{\mathbf{y}} \partial \mathbf{y}_0} \mathbf{w}_0$ p. 185, (A.42)

Table 4.2: References to the second derivatives of $\mathbf{t}$ for linear multistep methods.

4.1.2 Runge-Kutta Methods

References to the first derivatives of $\mathbf{t}$ are given in Table 4.3, and references to the second derivatives can be found in Table 4.4.

Let $\mathbf{w_m}$ be an arbitrary vector of length $N_\mathbf{m}$ and $\mathbf{w}_0$ an arbitrary vector of length N. Also let $\overline{\mathbf{w}} = \begin{bmatrix} \widehat{\mathbf{w}}_1^\top & \widehat{\mathbf{w}}_2^\top & \cdots & \widehat{\mathbf{w}}_K^\top \end{bmatrix}^\top$ be an arbitrary vector of length $(s+1)KN$ defined analogously to $\overline{\mathbf{y}}$.

Derivative	**Product**	**Page**	**Equation**
$\frac{\partial \mathbf{t}}{\partial \overline{\mathbf{y}}}$	$\frac{\partial \mathbf{t}_k}{\partial \overline{\mathbf{y}}} \overline{\mathbf{w}}$	p.188	(A.49)
	$\frac{\partial \mathbf{t}_k}{\partial \mathbf{y}_j}^\top \mathbf{w}_k$	p.188	(A.52)
$\frac{\partial \mathbf{t}}{\partial \mathbf{m}}$	$\frac{\partial \mathbf{t}_k}{\partial \mathbf{m}} \mathbf{w_m}$	p.189	(A.57)
	$\frac{\partial \mathbf{t}_k}{\partial \mathbf{m}}^\top \mathbf{w}_k$	p.189	(A.58)

Continued on next page...

Derivative	**Product**	**Page**	**Equation**
$\dfrac{\partial \mathbf{t}}{\partial \mathbf{y}_0}$	$\dfrac{\partial \mathbf{t}_k}{\partial \mathbf{y}_0}\mathbf{w}_0$	p.190	(A.61)
	$\dfrac{\partial \mathbf{t}_k}{\partial \mathbf{y}_0}^\top \mathbf{w}_k$	p.190	(A.62)

Table 4.3: References to the first derivatives of $\mathbf{t}$ for Runge-Kutta methods.

$\dfrac{\partial^2 \mathbf{t}}{\partial \mathbf{m} \partial \overline{\mathbf{y}}}\overline{\mathbf{w}}$ p. 193, (A.68)	$\dfrac{\partial^2 \mathbf{t}}{\partial \mathbf{m} \partial \mathbf{m}}\mathbf{w_m}$ p. 195, (A.72)	$\dfrac{\partial^2 \mathbf{t}}{\partial \mathbf{m} \partial \mathbf{y}_0}\mathbf{w}_0$ p. 197, (A.78)
$\dfrac{\partial^2 \mathbf{t}}{\partial \mathbf{y}_0 \partial \overline{\mathbf{y}}}\overline{\mathbf{w}}$ p. 194, (A.71)	$\dfrac{\partial^2 \mathbf{t}}{\partial \mathbf{y}_0 \partial \mathbf{m}}\mathbf{w_m}$ p. 195, (A.73)	$\dfrac{\partial^2 \mathbf{t}}{\partial \mathbf{y}_0 \partial \mathbf{y}_0}\mathbf{w}_0$ p. 196, (A.77)
$\dfrac{\partial^2 \mathbf{t}}{\partial \overline{\mathbf{y}} \partial \overline{\mathbf{y}}}\overline{\mathbf{w}}$ *p.* 191, (A.64)	$\dfrac{\partial^2 \mathbf{t}}{\partial \overline{\mathbf{y}} \partial \mathbf{m}}\mathbf{w_m}$ p. 195, (A.75)	$\dfrac{\partial^2 \mathbf{t}}{\partial \overline{\mathbf{y}} \partial \mathbf{y}_0}\mathbf{w}_0$ p. 197, (A.79)

Table 4.4: References to the second derivatives of $\mathbf{t}$ for Runge-Kutta methods.

4.2 Implicit-Explicit Time-Stepping Methods

4.2.1 IMEX Linear Multistep Methods

References to the first derivatives of $\mathbf{t}$ are given in Table 4.5, and references to the second derivatives can be found in Table 4.6.

We let $\overline{\mathbf{w}} = \begin{bmatrix} \mathbf{w}_1^\top & \cdots & \mathbf{w}_K^\top \end{bmatrix}^\top$ be an arbitrary vector of length KN, $\mathbf{w_m}$ an arbitrary vector of length $N_\mathbf{m}$, and $\mathbf{w}_0$ an arbitrary vector of length N.

Derivative	**Product**	**Page**	**Equation**
$\dfrac{\partial \mathbf{t}}{\partial \overline{\mathbf{y}}}$	$\dfrac{\partial \mathbf{t}}{\partial \mathbf{y}}\overline{\mathbf{w}}$	p.199	(A.84)
	$\dfrac{\partial \mathbf{t}}{\partial \mathbf{y}}^\top \overline{\mathbf{w}}$	p.199	(A.85)
$\dfrac{\partial \mathbf{t}}{\partial \mathbf{m}}$	$\dfrac{\partial \mathbf{t}}{\partial \mathbf{m}}\mathbf{w_m}$	p.200	(A.90)
	$\dfrac{\partial \mathbf{t}}{\partial \mathbf{m}}^\top \mathbf{w}$	p.200	(A.91)
$\dfrac{\partial \mathbf{t}}{\partial \mathbf{y}_0}$	$\dfrac{\partial \mathbf{t}}{\partial \mathbf{y}_0}\mathbf{w}_0$	p.201	(A.95)
	$\dfrac{\partial \mathbf{t}}{\partial \mathbf{y}_0}^\top \overline{\mathbf{w}}$	p.201	(A.96)

Table 4.5: References to the first derivatives of $\mathbf{t}$ for IMEX linear multistep methods.

$\dfrac{\partial^2 \mathbf{t}}{\partial \mathbf{m} \partial \overline{\mathbf{y}}}\overline{\mathbf{w}}$	$\dfrac{\partial^2 \mathbf{t}}{\partial \mathbf{m} \partial \mathbf{m}}\mathbf{w_m}$	$\dfrac{\partial^2 \mathbf{t}}{\partial \mathbf{m} \partial \mathbf{y}_0}\mathbf{w}_0$
p. 202, (A.98)	p. 204, (A.106)	p. 207, (A.116)

Continued on next page...

$\frac{\partial^2 \mathbf{t}}{\partial \mathbf{y}_0 \partial \overline{\mathbf{y}}}\overline{\mathbf{w}}$ p. 202, (A.101)	$\frac{\partial^2 \mathbf{t}}{\partial \mathbf{y}_0 \partial \mathbf{m}}\mathbf{w}_{\mathbf{m}}$ p. 205, (A.109)	$\frac{\partial^2 \mathbf{t}}{\partial \mathbf{y}_0 \partial \mathbf{y}_0}\mathbf{w}_0$ p. 208, (A.119)
$\frac{\partial^2 \mathbf{t}}{\partial \overline{\mathbf{y}} \partial \overline{\mathbf{y}}}\overline{\mathbf{w}}$ *p.* 203, (A.102)	$\frac{\partial^2 \mathbf{t}}{\partial \overline{\mathbf{y}} \partial \mathbf{m}}\mathbf{w}_{\mathbf{m}}$ p. 206, (A.112)	$\frac{\partial^2 \mathbf{t}}{\partial \overline{\mathbf{y}} \partial \mathbf{y}_0}\mathbf{w}_0$ p. 208, (A.122)

Table 4.6: References to the second derivatives of $\mathbf{t}$ for IMEX linear multistep methods.

4.2.2 IMEX Runge-Kutta Methods

References to the first derivatives of $\mathbf{t}$ are given in Table 4.7, and references to the second derivatives can be found in Table 4.8.

Let $\mathbf{w}_{\mathbf{m}}$ be an arbitrary vector of length $N_{\mathbf{m}}$ and $\mathbf{w}_0$ an arbitrary vector of length N. Also let $\overline{\mathbf{w}} = \begin{bmatrix} \widehat{\mathbf{w}}_1^\top & \widehat{\mathbf{w}}_2^\top & \cdots & \widehat{\mathbf{w}}_K^\top \end{bmatrix}^\top$ be an arbitrary vector of length $(s+1)KN$ defined analogously to $\overline{\mathbf{y}}$.

Derivative	**Product**	**Page**	**Equation**
$\dfrac{\partial \mathbf{t}}{\partial \overline{\mathbf{y}}}$	$\dfrac{\partial \mathbf{t}_k}{\partial \overline{\mathbf{y}}}\overline{\mathbf{w}}$	p.212	(A.129)
	$\dfrac{\partial \mathbf{t}}{\partial \overline{\mathbf{y}}}^{\top}\mathbf{w}$	p.214	(A.133)
$\dfrac{\partial \mathbf{t}}{\partial \mathbf{m}}$	$\dfrac{\partial \mathbf{t}}{\partial \mathbf{m}}\mathbf{w}$	p.215	(A.137)
	$\dfrac{\partial \mathbf{t}}{\partial \mathbf{m}}^{\top}\overline{\mathbf{w}}$	p.215	(A.138)
$\dfrac{\partial \mathbf{t}}{\partial \mathbf{y}_0}$	$\dfrac{\partial \mathbf{t}}{\partial \mathbf{y}_0}\mathbf{w}_0$	p.216	(A.141)
	$\dfrac{\partial \mathbf{t}}{\partial \mathbf{y}_0}^{\top}\overline{\mathbf{w}}$	p.216	(A.142)

Table 4.7: References to the first derivatives of $\mathbf{t}$ for IMEX Runge-Kutta methods.

$\dfrac{\partial^2 \mathbf{t}}{\partial \mathbf{m}\partial \overline{\mathbf{y}}}\overline{\mathbf{w}}$ p. 217, (A.143)	$\dfrac{\partial^2 \mathbf{t}}{\partial \mathbf{m}\partial \mathbf{m}}\mathbf{w_m}$ p. 219, (A.148)	$\dfrac{\partial^2 \mathbf{t}}{\partial \mathbf{m}\partial \mathbf{y}_0}\mathbf{w}_0$ p. 222, (A.152)
$\dfrac{\partial^2 \mathbf{t}}{\partial \mathbf{y}_0\partial \overline{\mathbf{y}}}\overline{\mathbf{w}}$ p. 217, (A.144)	$\dfrac{\partial^2 \mathbf{t}}{\partial \mathbf{y}_0\partial \mathbf{m}}\mathbf{w_m}$ p. 219, (A.149)	$\dfrac{\partial^2 \mathbf{t}}{\partial \mathbf{y}_0\partial \mathbf{y}_0}\mathbf{w}_0$ p. 222, (A.153)
$\dfrac{\partial^2 \mathbf{t}}{\partial \overline{\mathbf{y}}\partial \overline{\mathbf{y}}}\overline{\mathbf{w}}$ *p.* 218, (A.147)	$\dfrac{\partial^2 \mathbf{t}}{\partial \overline{\mathbf{y}}\partial \mathbf{m}}\mathbf{w_m}$ p. 220, (A.150)	$\dfrac{\partial^2 \mathbf{t}}{\partial \overline{\mathbf{y}}\partial \mathbf{y}_0}\mathbf{w}_0$ p. 222, (A.154)

Table 4.8: References to the second derivatives of $\mathbf{t}$ for IMEX Runge-Kutta methods.

4.3 Staggered Time-Stepping Methods

4.3.1 Staggered Linear Multistep Methods

References to the first derivatives of $\mathbf{t}$ are given in Table 4.9, and references to the second derivatives can be found in Table 4.10.

We let $\overline{\mathbf{w}} = \begin{bmatrix} \mathbf{w}_1^\top & \cdots & \mathbf{w}_K^\top \end{bmatrix}^\top$ be an arbitrary vector of length KN, $\mathbf{w_m}$ an arbitrary vector of length $N_\mathbf{m}$, and $\mathbf{w}_0$ an arbitrary vector of length N.

Derivative	**Product**	**Page**	**Equation**
$\dfrac{\partial \mathbf{t}}{\partial \overline{\mathbf{y}}}$	$\dfrac{\partial \mathbf{t}}{\partial \mathbf{y}}\overline{\mathbf{w}}$	p.225	(A.159)
	$\dfrac{\partial \mathbf{t}}{\partial \mathbf{y}}^\top \overline{\mathbf{w}}$	p.225	(A.160)
$\dfrac{\partial \mathbf{t}}{\partial \mathbf{m}}$	$\dfrac{\partial \mathbf{t}}{\partial \mathbf{m}}\mathbf{w_m}$	p.226	(A.165)
	$\dfrac{\partial \mathbf{t}}{\partial \mathbf{m}}^\top \mathbf{w}$	p.226	(A.166)
$\dfrac{\partial \mathbf{t}}{\partial \mathbf{y}_0}$	$\dfrac{\partial \mathbf{t}}{\partial \mathbf{y}_0}\mathbf{w}_0$	p.227	(A.170)
	$\dfrac{\partial \mathbf{t}}{\partial \mathbf{y}_0}^\top \overline{\mathbf{w}}$	p.227	(A.171)

Table 4.9: References to the first derivatives of $\mathbf{t}$ for staggered linear multistep methods.

$\dfrac{\partial^2 \mathbf{t}}{\partial \mathbf{m} \partial \overline{\mathbf{y}}}\overline{\mathbf{w}}$	$\dfrac{\partial^2 \mathbf{t}}{\partial \mathbf{m} \partial \mathbf{m}}\mathbf{w_m}$	$\dfrac{\partial^2 \mathbf{t}}{\partial \mathbf{m} \partial \mathbf{y}_0}\mathbf{w}_0$
p. 228, (A.173)	p. 229, (A.176)	p. 230, (A.180)

Continued on next page...

$\frac{\partial^2 \mathbf{t}}{\partial \mathbf{y}_0 \partial \overline{\mathbf{y}}}\overline{\mathbf{w}}$ p. 228, (A.174)	$\frac{\partial^2 \mathbf{t}}{\partial \mathbf{y}_0 \partial \mathbf{m}}\mathbf{w_m}$ p. 229, (A.177)	$\frac{\partial^2 \mathbf{t}}{\partial \mathbf{y}_0 \partial \mathbf{y}_0}\mathbf{w}_0$ p. 231, (A.181)
$\frac{\partial^2 \mathbf{t}}{\partial \overline{\mathbf{y}} \partial \overline{\mathbf{y}}}\overline{\mathbf{w}}$ *p.* 228, (A.175)	$\frac{\partial^2 \mathbf{t}}{\partial \overline{\mathbf{y}} \partial \mathbf{m}}\mathbf{w_m}$ p. 229, (A.178)	$\frac{\partial^2 \mathbf{t}}{\partial \overline{\mathbf{y}} \partial \mathbf{y}_0}\mathbf{w}_0$ p. 231, (A.182)

Table 4.10: References to the second derivatives of $\mathbf{t}$ for staggered linear multistep methods.

4.3.2 Staggered Runge-Kutta Methods

References to the first derivatives of $\mathbf{t}$ are given in Table 4.11, and references to the second derivatives can be found in Table 4.12.

Let $\mathbf{w_m}$ be an arbitrary vector of length $N_\mathbf{m}$ and $\mathbf{w}_0$ an arbitrary vector of length N. Also let $\overline{\mathbf{w}} = \begin{bmatrix} \widehat{\mathbf{w}}_1^\top & \widehat{\mathbf{w}}_2^\top & \cdots & \widehat{\mathbf{w}}_K^\top \end{bmatrix}^\top$ be an arbitrary vector of length $(s+1)KN$ defined analogously to $\overline{\mathbf{y}}$.

Derivative	Product	Page	Equation
$\frac{\partial \mathbf{t}}{\partial \overline{\mathbf{y}}}$	$\frac{\partial \mathbf{t}_k}{\partial \overline{\mathbf{y}}}\overline{\mathbf{w}}$	p.238	(A.197)
	$\frac{\partial \mathbf{t}}{\partial \overline{\mathbf{y}}}^\top \mathbf{w}$	p.240	(A.201)
$\frac{\partial \mathbf{t}}{\partial \mathbf{m}}$	$\frac{\partial \mathbf{t}}{\partial \mathbf{m}}\mathbf{w}$	p.242	(A.205)
	$\frac{\partial \mathbf{t}}{\partial \mathbf{m}}^\top \overline{\mathbf{w}}$	p.243	(A.206)
$\frac{\partial \mathbf{t}}{\partial \mathbf{y}_0}$	$\frac{\partial \mathbf{t}}{\partial \mathbf{y}_0}\mathbf{w}_0$	p.245	(A.210)
	$\frac{\partial \mathbf{t}}{\partial \mathbf{y}_0}^\top \overline{\mathbf{w}}$	p.246	(A.211)

Table 4.11: References to the first derivatives of $\mathbf{t}$ for staggered Runge-Kutta methods.

$\frac{\partial^2 \mathbf{t}}{\partial \mathbf{m}\partial \overline{\mathbf{y}}}\overline{\mathbf{w}}$ p. 247, (A.212)	$\frac{\partial^2 \mathbf{t}}{\partial \mathbf{m}\partial \mathbf{m}}\mathbf{w_m}$ p. 253, (A.217)	$\frac{\partial^2 \mathbf{t}}{\partial \mathbf{m}\partial \mathbf{y}_0}\mathbf{w}_0$ p. 260, (A.222)
$\frac{\partial^2 \mathbf{t}}{\partial \mathbf{y}_0\partial \overline{\mathbf{y}}}\overline{\mathbf{w}}$ p. 248, (A.213)	$\frac{\partial^2 \mathbf{t}}{\partial \mathbf{y}_0\partial \mathbf{m}}\mathbf{w_m}$ p. 254, (A.218)	$\frac{\partial^2 \mathbf{t}}{\partial \mathbf{y}_0\partial \mathbf{y}_0}\mathbf{w}_0$ p. 261, (A.223)
$\frac{\partial^2 \mathbf{t}}{\partial \overline{\mathbf{y}}\partial \overline{\mathbf{y}}}\overline{\mathbf{w}}$ *p.* 252, (A.216)	$\frac{\partial^2 \mathbf{t}}{\partial \overline{\mathbf{y}}\partial \mathbf{m}}\mathbf{w_m}$ p. 259, (A.221)	$\frac{\partial^2 \mathbf{t}}{\partial \overline{\mathbf{y}}\partial \mathbf{y}_0}\mathbf{w}_0$ p. 263, (A.224)

Table 4.12: References to the second derivatives of $\mathbf{t}$ for staggered Runge-Kutta methods.

4.4 Exponential Time-Stepping Methods

4.4.1 Exponential Runge-Kutta Methods

References to the first derivatives of $\mathbf{t}$ are given in Table 4.13. We do not consider second derivatives of ETDRK methods in this thesis.

Let $\mathbf{w}$ be an arbitrary vector of length $N_{\mathbf{m}}$ and $\mathbf{w}_0$ an arbitrary vector of length N. Also let $\overline{\mathbf{w}} = \begin{bmatrix} \widehat{\mathbf{w}}_1^\top & \cdots & \widehat{\mathbf{w}}_K^\top \end{bmatrix}^\top$ be an arbitrary vector of length $(s+1)KN$ defined analogously to $\overline{\mathbf{y}}$.

Derivative	**Product**	**Page**	**Equation**
$\dfrac{\partial \mathbf{t}}{\partial \overline{\mathbf{y}}}$	$\dfrac{\partial \mathbf{t}_k}{\partial \overline{\mathbf{y}}}\overline{\mathbf{w}}$	p.267	(A.232)
	$\dfrac{\partial \mathbf{t}}{\partial \overline{\mathbf{y}}}^\top \mathbf{w}$	p.268	(A.235)
$\dfrac{\partial \mathbf{t}}{\partial \mathbf{m}}$	$\dfrac{\partial \mathbf{t}}{\partial \mathbf{m}}\mathbf{w}$	p.270	(A.239)
	$\dfrac{\partial \mathbf{t}}{\partial \mathbf{m}}^\top \overline{\mathbf{w}}$	p.270	(A.240)
$\dfrac{\partial \mathbf{t}}{\partial \mathbf{y}_0}$	$\dfrac{\partial \mathbf{t}}{\partial \mathbf{y}_0}\mathbf{w}_0$	p.271	(A.241)
	$\dfrac{\partial \mathbf{t}}{\partial \mathbf{y}_0}^\top \overline{\mathbf{w}}$	p.272	(A.243)

Table 4.13: References to the first derivatives of $\mathbf{t}$ for exponential time-differencing Runge-Kutta methods.

4.4.2 The Derivatives of φ_ℓ

If the linear operator $\mathbf{L}_k$ depends on either $\mathbf{y}_k$ or $\mathbf{m}$ we have to be able to take the derivatives of the products $a_{\sigma j}(\tau_k \mathbf{L}_k(\mathbf{y}_k, \mathbf{m}))\mathbf{w}$, $b_\sigma(\tau_k \mathbf{L}_k(\mathbf{y}_k, \mathbf{m}))\mathbf{w}$ and $e^{c_\sigma \tau_k \mathbf{L}_k(\mathbf{y}_k, \mathbf{m})}\mathbf{w}$, where $\mathbf{w}$ is an arbitrary vector of length N, with respect to $\mathbf{y}_k$ or $\mathbf{m}$.

Since the $a_{\sigma j}$ and b_σ are linear combinations of the $\varphi_{\ell,\sigma}$-and φ_ℓ-functions respectively, we will simply consider the derivatives of $\varphi(\mathbf{L})\mathbf{w}$ for some arbitrary φ-function and matrix $\mathbf{L} = \tau_k \mathbf{L}_k(\mathbf{y}_k, \mathbf{m})$. Without loss of generality we assume in this subsection that the differentiation is with respect to $\mathbf{y}_k$, but all the results also apply when taking the derivative with respect to $\mathbf{m}$.

The calculation of the derivatives of the products of the φ_ℓ-functions will depend on the way these terms are evaluated numerically. Recall the approaches for evaluating φ_ℓ reviewed in Section 2.4.2.

Krylov Subspace Methods

Using Krylov subspace methods is unfortunately unsuitable for our purposes since if $\mathbf{L}$ depends on $\mathbf{y}_k$ or $\mathbf{m}$, it is extremely difficult to find the dependence of the right-hand side of (2.65) on these variables. For this reason we will not consider this approach further here, although it can of course be used for parameter estimation problems where $\mathbf{L}$ is independent of $\mathbf{y}_k$ and $\mathbf{m}$.

Polynomial Approach

If we represent φ by a truncated polynomial series as in (2.66) and multiply by $\mathbf{w}$, we have

$$\varphi(\mathbf{L}(\mathbf{y}_k))\mathbf{w} \approx \sum_{j=0}^{M} \omega_j \, \mathbf{L}^j(\mathbf{y}_k)\mathbf{w}.$$

Let $\mathbf{L}$ depend on a single parameter z first. The derivative of $\mathbf{L}^j(z)\mathbf{w}$ then is

$$\frac{\partial \, \mathbf{L}^j(z)}{\partial \, z}\mathbf{w} = \sum_{i=1}^{j} \mathbf{L}^{i-1} \frac{\partial \, \mathbf{L}(z)}{\partial \, z} \mathbf{L}^{j-i}\mathbf{w} = \sum_{i=1}^{j} \mathbf{L}^{i-1} \frac{\partial \, \mathbf{L}(z)}{\partial \, z} \mathbf{v}_{j-i}$$

with $\mathbf{v}_{j-i} = \mathbf{L}^{j-i}\mathbf{w}$. In the case of $\mathbf{L} = \mathbf{L}(\mathbf{y}_k)$ we therefore have

$$\frac{\partial \, \mathbf{L}^j(\mathbf{y}_k)\mathbf{w}}{\partial \, \mathbf{y}_k} = \sum_{i=1}^{j} \mathbf{L}^{i-1} \frac{\partial \, \mathbf{L}(\mathbf{y}_k)\mathbf{v}_{j-i}}{\partial \, \mathbf{y}_k}.$$

This implies that we need to have $j-1$ derivatives $\dfrac{\partial\,\mathbf{L}(\mathbf{y}_k)\mathbf{v}_{j-i}}{\partial\,\mathbf{y}_k}$ available for each j, but of course a lot of these derivatives can also be reused for different values of j.

In the special case where the matrices $\mathbf{L}$ and $\dfrac{\partial\,\mathbf{L}}{\partial\,y}$ commute for each element y of $\mathbf{y}_k$, we can simplify the above by using the matrix analogue of the usual power rule:

$$\frac{\partial\,\mathbf{L}^j(\mathbf{y}_k)}{\partial\,\mathbf{y}_k}\mathbf{w} = \mathbf{L}^{j-1}\frac{\partial\,\mathbf{L}(\mathbf{y}_k)\mathbf{w}}{\partial\,\mathbf{y}_k}.$$

This is much less cumbersome to implement, but it is hard to think of realistic scenarios where this commutativity property would hold.

The drawback of considering each monomial term on its own is that this does not necessarily reflect how the polynomial approximation of φ is actually computed. For instance, the Chebyshev approximation (2.67) is

$$\varphi(\mathbf{L})\mathbf{w} \approx \sum_{j=0}^{M} \omega_j\, T_j(\mathbf{L})\mathbf{w}$$

assuming $\mathbf{L}(\mathbf{m})$ is Hermitian or skew-Hermitian and the eigenvalues of $\mathbf{L}(\mathbf{y}_k)$ all lie inside $[-1,1]$. Each $T_j(\mathbf{y}_k)\mathbf{w}$ is computed using the recurrence relation

$$T_{j+1}(\mathbf{L}(\mathbf{y}_k))\mathbf{w} = 2\mathbf{L}\,T_j(\mathbf{L}(\mathbf{y}_k))\mathbf{w} - T_{j-1}(\mathbf{L}(\mathbf{y}_k))\mathbf{w}, \qquad j = 1, 2, \ldots$$

initialized by $T_0(\mathbf{L}(\mathbf{y}_k))\mathbf{w} = \mathbf{w}$ and $T_1(\mathbf{L}(\mathbf{y}_k))\mathbf{w} = \mathbf{L}(\mathbf{y}_k)\mathbf{w}$. Taking the derivative of (2.67) with respect to $\mathbf{y}_k$,

$$\frac{\partial\,\varphi(\mathbf{L})\mathbf{w}}{\partial\,\mathbf{y}_k} \approx \sum_{j=0}^{M} \omega_j \frac{\partial\,T_j(\mathbf{L})\mathbf{w}}{\partial\,\mathbf{y}_k},$$

with the recurrence relation

$$\frac{\partial\,T_{j+1}(\mathbf{L})\mathbf{w}}{\partial\,\mathbf{y}_k} = 2\frac{\partial\,(\mathbf{L}\,\mathbf{v})}{\partial\,\mathbf{y}_k} + 2\mathbf{L}\frac{\partial\,T_j(\mathbf{L})\mathbf{w}}{\partial\,\mathbf{y}_k} - \frac{\partial\,T_{j-1}(\mathbf{L})\mathbf{w}}{\partial\,\mathbf{y}_k}, \qquad j = 1, 2, \ldots$$

where $\mathbf{v} = T_j\mathbf{w}$, $\dfrac{\partial\, T_0(\mathbf{L})\mathbf{w}}{\partial\, \mathbf{y}_k} = \mathbf{0}_{N\times N}$ and $\dfrac{\partial\, T_1(\mathbf{L})\mathbf{w}}{\partial\, \mathbf{y}_k} = \dfrac{\partial\,(\mathbf{L}\,\mathbf{w})}{\partial\, \mathbf{y}_k}$.

We also need to be able to compute the products of the transposes of $\dfrac{\partial\,\varphi(\mathbf{L})\mathbf{w}}{\partial\, \mathbf{y}_k}$ with some vector; this is straightforward to derive from the equations above.

Rational Approximations and Contour Integration

We saw in Section 2.4.2 that the action of a φ-function can be computed by both rational approximations (under certain conditions) and contour integrals in the form

$$\varphi(\mathbf{L})\mathbf{w} \approx \sum_{i=1}^{M} \omega_i \left(s_i\mathbf{I} - \mathbf{L}\right)^{-1} \mathbf{w},$$

for some complex scalars ω_i and s_i, where the s_i do not coincide with the eigenvalues of $\mathbf{L}$.

To find the derivative of $\varphi(\mathbf{L})\mathbf{w}$ in this case, let $\mathbf{v}_i = \left(s_i\mathbf{I} - \mathbf{L}\right)^{-1}\mathbf{w}$, so that

$$\frac{\partial\,\varphi(\mathbf{L})\,\mathbf{w}}{\partial\, \mathbf{y}_k} \approx \sum_{i=1}^{M} \omega_i \frac{\partial\,\mathbf{v}_i}{\partial\, \mathbf{y}_k}. \tag{4.1}$$

To find an expression for $\dfrac{\partial\,\mathbf{v}_i}{\partial\, \mathbf{y}_k}$, consider $\left(s_i\mathbf{I} - \mathbf{L}\right)\mathbf{v}_i = \mathbf{w}$ and take the derivative with respect to $\mathbf{y}_k$ on both sides,

$$\begin{aligned}
\frac{\partial}{\partial\, \mathbf{y}_k}\left(s_i\mathbf{I} - \mathbf{L}\right)\mathbf{v}_i = \mathbf{0}_{N\times N} \qquad &\Rightarrow \qquad \left(s_i\mathbf{I} - \mathbf{L}\right)\frac{\partial\,\mathbf{v}_i}{\partial\, \mathbf{y}_k} - \frac{\partial\,(\mathbf{L}\,\mathbf{v}_i)}{\partial\, \mathbf{y}_k} = \mathbf{0}_{N\times N} \\
&\Rightarrow \qquad \frac{\partial\,\mathbf{v}_i}{\partial\, \mathbf{y}_k} = \left(s_i\mathbf{I} - \mathbf{L}\right)^{-1}\frac{\partial\,(\mathbf{L}\,\mathbf{v}_i)}{\partial\, \mathbf{y}_k},
\end{aligned}$$

where the $\mathbf{v}_i$ on the right-hand side is taken to be fixed.

For each term in the sum in (4.1) we thus require two matrix solves, one to find $\mathbf{v}_i$ and an additional one to then find $\left(s_i\mathbf{I} - \mathbf{L}\right)^{-1}\frac{\partial\,(\mathbf{L}\mathbf{v}_i)}{\partial\, \mathbf{y}_k}$. As mentioned previously, the $\mathbf{v}_i$ can be computed using the same Krylov subspace. However, with $\mathbf{z}$ an arbitrary

vector of length N, the linear systems

$$(s_i\mathbf{I} - \mathbf{L})\,\mathbf{u}_i = \frac{\partial\,(\mathbf{L}\,\mathbf{v}_i)}{\partial\,\mathbf{y}_k}\mathbf{z}$$

each have a different right-hand side, and therefore we are no longer working in the same Krylov subspace. This means that we need to solve M different linear systems just for a single evaluation of the derivative of a given φ-function, which is not ideal. The process is fortunately highly parallelizable and given the ease of access to a large number of processors these days, we do not consider this to be the bottleneck it might have been just a few years ago. Nonetheless it is a significant inconvenience for many a mathematician, and the polynomial approach does not suffer from this limitation.

We also should be able to compute the transpose of (4.1). In this case we have

$$\frac{\partial\,\varphi(\mathbf{L})\,\mathbf{w}}{\partial\,\mathbf{y}_k}^{\top} \approx \sum_{i=1}^{M} \omega_i \frac{\partial\,\mathbf{v}_i}{\partial\,\mathbf{y}_k}^{\top},$$

with $\frac{\partial\,\mathbf{v}_i}{\partial\,\mathbf{y}_k}^{\top} = \frac{\partial\,(\mathbf{L}\,\mathbf{v}_i)}{\partial\,\mathbf{y}_k}^{\top}(s_i\mathbf{I} - \mathbf{L})^{-\top} = \frac{\partial\,(\mathbf{L}\,\mathbf{v}_i)}{\partial\,\mathbf{y}_k}^{\top}\left(s_i\mathbf{I} - \mathbf{L}^{\top}\right)^{-1}$, so when multiplying by some vector $\mathbf{z}$ we can work within the same Krylov subspace $\mathcal{K}_M(\mathbf{L}^{\top}, \mathbf{z})$ for any $s_i \in \mathbb{C}^N$.

Chapter 5

The Linearized Forward Problem

In Chapter 3 we showed that computing the action of the sensitivity matrix $\mathbf{J}$ on some vector $\mathbf{w_p}$ of length $N_\mathbf{p}$ requires the solution of the linearized forward problem

$$\overline{\boldsymbol{\xi}} = \left(\frac{\partial \mathbf{t}}{\partial \overline{\mathbf{y}}}\right)^{-1} \overline{\mathbf{q}}, \tag{5.1}$$

where we will refer to $\overline{\boldsymbol{\xi}}$ as the *linearized forward solution* and $\overline{\mathbf{q}}$ in this chapter is taken to be some arbitrary source term; it is not the same source used by the forward problem. In the context of sensitivity analysis we will have

$$\overline{\mathbf{q}} = \begin{bmatrix} -\dfrac{\partial \mathbf{t}}{\partial \mathbf{m}} & -\dfrac{\partial \mathbf{t}}{\partial \mathbf{y}_0} & \mathbf{S}\dfrac{\partial \mathbf{q}}{\partial \mathbf{s}} \end{bmatrix} \mathbf{w_p},$$

see (3.9). The linearization of the time-stepping equation, $\dfrac{\partial \mathbf{t}}{\partial \overline{\mathbf{y}}}$, is derived in Appendix A for each of the time-stepping methods considered in this thesis.

Solving the linearized forward problem amounts to simply solving the standard forward problem with the nonlinear terms having been linearized about the value of the forward solution $\overline{\mathbf{y}}$ at the current time level. Therefore the methods in this chapter are not particularly interesting and we present them here simply for the sake of having a complete discussion; we also give examples of the linearizations of the particular time-stepping schemes given as examples in Chapter 2.

The exception to the above are exponential Rosenbrock-type methods, where a significant number of additional terms appear from the linearization. In this case the linearized forward problem is not at all straightforward, so the example given for this case is especially important to illustrate the implementation.

We make the following remarks before we start:

- The linearization of the nonlinear operators about the current values of the forward solution means that we must have the forward solution $\overline{\mathbf{y}}$ available, either in storage or computed on the fly using checkpointing.
- The structure of (5.1) dictates that the initial value of the linearized forward solution must be $\boldsymbol{\xi}_0 = \mathbf{0}_{N\times 1}$.
- We do not have a source weighting operator in (5.1), so for LM-type methods we do not need to combine source values from different time levels.
- We do not assume any structure on $\mathbf{q}$, so it is possible that RK-type methods will have a source term appearing in the update step.
- In the case of the operator $\mathbf{f}$ being linear, the methods below are identical to the standard forward solution procedure for linear problems, apart from how the source terms are treated.

5.1 Regular Time-Stepping Methods

Here we present the linearized LM and RK methods, where, for LM methods, the nonlinear operator $\mathbf{f}$ is linearized about the solution at the kth time level, $\mathbf{y}_k$, giving us the linear operator $\dfrac{\partial \mathbf{f}_k}{\partial \mathbf{y}}$. For RK methods the operator is linearized about the σth internal stage at the kth time-step, $\dfrac{\partial \mathbf{F}_{k,\sigma}}{\partial \mathbf{y}}$.

5.1.1 Linear Multistep Methods

Let $\overline{\boldsymbol{\xi}} = \begin{bmatrix} \boldsymbol{\xi}_1^\top & \cdots & \boldsymbol{\xi}_K^\top \end{bmatrix}^\top$ represent the solution to the linearized forward problem, and let $\overline{\mathbf{q}} = \begin{bmatrix} \mathbf{q}_1^\top & \cdots & \mathbf{q}_K^\top \end{bmatrix}^\top$ be the source.

The $K \times K$ block matrix $\dfrac{\partial \mathbf{t}}{\partial \overline{\mathbf{y}}}$ represents the linearized time-stepping procedure and was found in Section A.1.1 to have the form of (2.1) and (2.3), with the (k, j)th $N \times N$ block being (see (A.3))

$$\frac{\partial \mathbf{t}_k}{\partial \mathbf{y}_j} = \begin{cases} \alpha_{k-j}^{(\sigma)} \mathbf{I}_N - \tau \, \beta_{k-j}^{(\sigma)} \dfrac{\partial \mathbf{f}_j}{\partial \mathbf{y}} & \text{if } k^\star \leq j \leq k \\ \mathbf{0} & \text{otherwise,} \end{cases} \tag{5.2}$$

where $\sigma = \min(k, s)$ and $k^\star = \max(1, k - s)$.

To solve (5.1) we therefore must solve, for $1 \leq k \leq K$,

$$\frac{\partial \mathbf{t}_k}{\partial \mathbf{y}_k} \boldsymbol{\xi}_k = \mathbf{q}_k - \sum_{j=k^\star}^{k-1} \frac{\partial \mathbf{t}_k}{\partial \mathbf{y}_j} \boldsymbol{\xi}_j.$$

The solution procedure is summarized in Algorithm 5.1.

Algorithm 5.1. The Linearized Linear Multistep Method

Let $\sigma = \min(k, s)$ *and* $k^\star = \max(1, k - s)$.

For $k = 1, \cdots, K$, *solve for* $\boldsymbol{\xi}_k$*:*

$$\left(\mathbf{I}_N - \tau \, \beta_0^{(\sigma)} \frac{\partial \mathbf{f}_k}{\partial \mathbf{y}} \right) \boldsymbol{\xi}_k = \mathbf{q}_k - \sum_{j=k^\star}^{k-1} \left(\alpha_{k-j}^{(\sigma)} \mathbf{I}_N - \tau \, \beta_{k-j}^{(\sigma)} \frac{\partial \mathbf{f}_j}{\partial \mathbf{y}} \right) \boldsymbol{\xi}_j,$$

with $\dfrac{\partial \mathbf{f}_k}{\partial \mathbf{y}} = \dfrac{\partial \mathbf{f}(\mathbf{y}_k)}{\partial \mathbf{y}} = \dfrac{\partial \mathbf{f}(\mathbf{y}_k, t_k)}{\partial \mathbf{y}}$.

Examples

We find the linearized versions of the IMEX linear multistep methods mentioned in 2.2.1.

The linearized s-step Adams-Bashforth methods with $s = 1, 2, 3$ are, for $k = 1, \cdots, K$,

$$
\begin{aligned}
s = 1: \quad & \boldsymbol{\xi}_k = \mathbf{q}_k + \left(\mathbf{I}_N + \tau \frac{\partial \mathbf{f}\left(\mathbf{y}_{k-1}\right)}{\partial \mathbf{y}}\right) \boldsymbol{\xi}_{k-1} \\
s = 2: \quad & \boldsymbol{\xi}_k = \mathbf{q}_k + \left(\mathbf{I}_N + \tau \frac{3}{2} \frac{\partial \mathbf{f}\left(\mathbf{y}_{k-1}\right)}{\partial \mathbf{y}}\right) \boldsymbol{\xi}_{k-1} - \frac{\tau}{2} \frac{\partial \mathbf{f}\left(\mathbf{y}_{k-2}\right)}{\partial \mathbf{y}} \boldsymbol{\xi}_{k-2} \\
s = 3: \quad & \boldsymbol{\xi}_k = \mathbf{q}_k + \left(\mathbf{I}_N + \tau \frac{23}{12} \frac{\partial \mathbf{f}\left(\mathbf{y}_{k-1}\right)}{\partial \mathbf{y}}\right) \boldsymbol{\xi}_{k-1} - \tau \frac{4}{3} \frac{\partial \mathbf{f}\left(\mathbf{y}_{k-2}\right)}{\partial \mathbf{y}} \boldsymbol{\xi}_{k-2} + \\
& \qquad + \tau \frac{5}{12} \frac{\partial \mathbf{f}\left(\mathbf{y}_{k-3}\right)}{\partial \mathbf{y}} \boldsymbol{\xi}_{k-3}.
\end{aligned}
$$

The linearized s-step BDFs of orders $s = 1, 2, 3$ are, for $k = 1, \cdots, K$,

$$
\begin{aligned}
s = 1: \quad & \left(\mathbf{I}_N - \tau \frac{\partial \mathbf{f}\left(\mathbf{y}_k\right)}{\partial \mathbf{y}}\right) \boldsymbol{\xi}_k = \mathbf{q}_k + \boldsymbol{\xi}_{k-1} \\
s = 2: \quad & \left(\mathbf{I}_N - \tau \frac{2}{3} \frac{\partial \mathbf{f}\left(\mathbf{y}_k\right)}{\partial \mathbf{y}}\right) \boldsymbol{\xi}_k = \mathbf{q}_k + \frac{4}{3} \boldsymbol{\xi}_{k-1} - \frac{1}{3} \boldsymbol{\xi}_{k-2} \\
s = 3: \quad & \left(\mathbf{I}_N - \tau \frac{6}{11} \frac{\partial \mathbf{f}\left(\mathbf{y}_k\right)}{\partial \mathbf{y}}\right) \boldsymbol{\xi}_k = \mathbf{q}_k + \frac{18}{11} \boldsymbol{\xi}_{k-1} - \frac{9}{11} \boldsymbol{\xi}_{k-2} + \frac{2}{11} \boldsymbol{\xi}_{k-3}.
\end{aligned}
$$

5.1.2 Runge-Kutta Methods

For RK methods we have $\overline{\boldsymbol{\xi}} = \begin{bmatrix} \widehat{\boldsymbol{\xi}}_1^\top & \cdots & \widehat{\boldsymbol{\xi}}_K^\top \end{bmatrix}^\top$, where $\widehat{\boldsymbol{\xi}}_k = \begin{bmatrix} \boldsymbol{\Xi}_k^\top & \boldsymbol{\xi}_k^\top \end{bmatrix}^\top$. $\boldsymbol{\Xi}_k$ consists of the internal stages at the kth time-step. The source term $\overline{\mathbf{q}} = \begin{bmatrix} \widehat{\mathbf{q}}_1^\top & \cdots & \widehat{\mathbf{q}}_K^\top \end{bmatrix}^\top$ is defined similarly.

The $(s+1)KN \times (s+1)KN$ matrix $\dfrac{\partial \mathbf{t}}{\partial \overline{\mathbf{y}}}$ was found in A.1.2 to be (see (A.48))

$$
\frac{\partial \mathbf{t}}{\partial \overline{\mathbf{y}}} = \begin{bmatrix}
\mathbf{A}_1 & & & & & & \\
\mathbf{B}^\top & \mathbf{I}_N & & & & & \\
& \mathbf{C}_2 & \mathbf{A}_2 & & & & \\
& -\mathbf{I}_N & \mathbf{B}^\top & \mathbf{I}_N & & & \\
& & \ddots & \ddots & \ddots & & \\
& & & & \mathbf{C}_K & \mathbf{A}_K & \\
& & & & -\mathbf{I}_N & \mathbf{B}^\top & \mathbf{I}_N
\end{bmatrix},
$$

where (see (A.47))

$$\mathbf{A}_k = \mathbf{I}_{sN} - \tau_k \frac{\partial \mathbf{F}_k}{\partial \mathbf{y}} \left(\mathbf{A}_s \otimes \mathbf{I}_N\right)$$
$$\mathbf{C}_k = -\frac{\partial \mathbf{F}_k}{\partial \mathbf{y}} \left(\mathbf{1}_s \otimes \mathbf{I}_N\right)$$

and $\mathbf{B} = -\tau_k \mathbf{b}_s \otimes \mathbf{I}_N$. $\dfrac{\partial \mathbf{F}_k}{\partial \mathbf{y}}$ is given by

$$\frac{\partial \mathbf{F}_k}{\partial \mathbf{y}} = \text{blkdiag}\left(\frac{\partial \mathbf{F}_{k,1}}{\partial \mathbf{y}}, \cdots, \frac{\partial \mathbf{F}_{k,s}}{\partial \mathbf{y}}\right)$$

with $\mathbf{F}_{k,\sigma} = \mathbf{f}\left(\underline{\mathbf{y}}_{k,\sigma}, t_{k-1} + c_\sigma \tau_k\right)$ and $\underline{\mathbf{y}}_{k,\sigma} = \mathbf{y}_{k-1} + \tau_k \sum_{i=1}^{s} a_{\sigma i}\, \mathbf{Y}_{k,i}$.

The system (5.1) is then obtained by solving, for $1 \leq k \leq K$, the internal stages

$$\mathbf{A}_k \boldsymbol{\Xi}_k = \mathbf{Q}_k - \mathbf{C}_k \boldsymbol{\xi}_{k-1}$$
$$\Rightarrow \qquad \left(\mathbf{I}_{sN} - \tau_k \frac{\partial \mathbf{F}_k}{\partial \mathbf{y}} \left(\mathbf{A}_s \otimes \mathbf{I}_N\right)\right) \boldsymbol{\Xi}_k = \mathbf{Q}_k + \frac{\partial \mathbf{F}_k}{\partial \mathbf{y}} \left(\mathbf{1}_s \otimes \mathbf{I}_N\right) \boldsymbol{\xi}_{k-1}$$

and then computing the next step of the linearized solution,

$$\boldsymbol{\xi}_k = \mathbf{q}_k + \boldsymbol{\xi}_{k-1} - \mathbf{B}^\top \boldsymbol{\Xi}_k.$$

The detailed solution procedure is presented in Algorithm 5.2.

Algorithm 5.2. The Linearized Runge-Kutta Method

Set $\boldsymbol{\xi}_0 = \mathbf{0}_{N\times 1}$ *and recall that we defined*

$$\underline{\mathbf{y}}_{k,\sigma} = \mathbf{y}_{k-1} + \tau_k \sum_{i=1}^{\sigma} a_{\sigma i} \mathbf{Y}_{k,i}.$$

For $k = 1, \cdots, K$*:*

- *For $\sigma = 1, \cdots, s$, evaluate the internal stages:*

$$\boldsymbol{\Xi}_{k,\sigma} = \mathbf{Q}_{k,\sigma} + \frac{\partial \mathbf{f}\left(\underline{\mathbf{y}}_{k,\sigma}\right)}{\partial \mathbf{y}} \left(\boldsymbol{\xi}_{k-1} + \tau_k \sum_{i=1}^{s} a_{\sigma i} \boldsymbol{\Xi}_{k,i}\right).$$

- *Update $\boldsymbol{\xi}$:*

$$\boldsymbol{\xi}_k = \mathbf{q}_k + \boldsymbol{\xi}_{k-1} + \tau_k \sum_{\sigma=1}^{s} b_\sigma \boldsymbol{\Xi}_{k,\sigma}.$$

Examples

The linearized RK4 method is, for $k = 1, \cdots, K$:

- Compute the internal stages for $\boldsymbol{\xi}_k$:

$$\begin{aligned}
\boldsymbol{\Xi}_{k,1} &= \mathbf{Q}_{k,1} + \frac{\partial \mathbf{f}\left(\mathbf{y}_{k-1}\right)}{\partial \mathbf{y}} \boldsymbol{\xi}_{k-1} \\
\boldsymbol{\Xi}_{k,2} &= \mathbf{Q}_{k,2} + \frac{\partial \mathbf{f}\left(\mathbf{y}_{k-1} + \frac{\tau_k}{2}\mathbf{Y}_{k,1}\right)}{\partial \mathbf{y}} \left(\boldsymbol{\xi}_{k-1} + \frac{\tau_k}{2}\boldsymbol{\Xi}_{k,1}\right) \\
\boldsymbol{\Xi}_{k,3} &= \mathbf{Q}_{k,3} + \frac{\partial \mathbf{f}\left(\mathbf{y}_{k-1} + \frac{\tau_k}{2}\mathbf{Y}_{k,2}\right)}{\partial \mathbf{y}} \left(\boldsymbol{\xi}_{k-1} + \frac{\tau_k}{2}\boldsymbol{\Xi}_{k,2}\right) \\
\boldsymbol{\Xi}_{k,4} &= \mathbf{Q}_{k,4} + \frac{\partial \mathbf{f}\left(\mathbf{y}_{k-1} + \tau_k \mathbf{Y}_{k,3}\right)}{\partial \mathbf{y}} \left(\boldsymbol{\xi}_{k-1} + \tau_k \boldsymbol{\Xi}_{k,3}\right).
\end{aligned}$$

- Update $\boldsymbol{\xi}$:

$$\boldsymbol{\xi}_k = \mathbf{q}_k + \boldsymbol{\xi}_{k-1} + \frac{\tau_k}{6}\left(\boldsymbol{\Xi}_{k,1} + 2\boldsymbol{\Xi}_{k,2} + 2\boldsymbol{\Xi}_{k,3} + \boldsymbol{\Xi}_{k,4}\right).$$

The linearized version of the fourth-order 2-stage implicit RK method mentioned in Section 2.1.2 is

- Let $\mathbf{G}_1 = \dfrac{\partial \mathbf{f}\left(\underline{\mathbf{y}}_{k,1}\right)}{\partial \mathbf{y}}$ and $\mathbf{G}_2 = \dfrac{\partial \mathbf{f}\left(\underline{\mathbf{y}}_{k,2}\right)}{\partial \mathbf{y}}$, with

$$\begin{aligned}
\underline{\mathbf{y}}_{k,1} &= \mathbf{y}_{k-1} + \frac{\tau_k}{4}\mathbf{Y}_{k,1} + \frac{\tau_k}{24}(6 - 4\sqrt{3})\,\mathbf{Y}_{k,2} \\
\underline{\mathbf{y}}_{k,2} &= \mathbf{y}_{k-1} + \frac{\tau_k}{24}(6 + 4\sqrt{3})\,\mathbf{Y}_{k,1} + \frac{\tau_k}{4}\mathbf{Y}_{k,2}.
\end{aligned}$$

Solve for the internal stages of $\boldsymbol{\xi}_k$:

$$\begin{bmatrix} \mathbf{I}_N - \frac{\tau_k}{4}\mathbf{G}_1 & -\frac{\tau_k}{24}(6-4\sqrt{3})\mathbf{G}_1 \\ -\frac{\tau_k}{24}(6+4\sqrt{3})\mathbf{G}_2 & \mathbf{I}_N - \frac{\tau_k}{4}\mathbf{G}_2 \end{bmatrix} \begin{bmatrix} \boldsymbol{\Xi}_{k,1} \\ \boldsymbol{\Xi}_{k,2} \end{bmatrix} = \begin{bmatrix} \mathbf{Q}_{k,1} \\ \mathbf{Q}_{k,2} \end{bmatrix} + \begin{bmatrix} \mathbf{G}_1 \\ \mathbf{G}_2 \end{bmatrix} \boldsymbol{\xi}_{k-1}$$

- Update $\boldsymbol{\xi}$:

$$\boldsymbol{\xi}_k = \mathbf{q}_k + \boldsymbol{\xi}_{k-1} + \frac{\tau_k}{2}\left(\boldsymbol{\Xi}_{k,1} + \boldsymbol{\Xi}_{k,2}\right).$$

5.2 Implicit-Explicit Time-Stepping Methods

We now apply the same procedure to find the linearized IMEX LM and RK methods.

5.2.1 IMEX Linear Multistep Methods

Let $\overline{\boldsymbol{\xi}} = \begin{bmatrix} \boldsymbol{\xi}_1^\top & \cdots & \boldsymbol{\xi}_K^\top \end{bmatrix}^\top$ represent the solution to the linearized forward problem, and let $\overline{\mathbf{q}} = \begin{bmatrix} \mathbf{q}_1^\top & \cdots & \mathbf{q}_K^\top \end{bmatrix}^\top$ be the source.

The $K \times K$ block matrix $\dfrac{\partial \mathbf{t}}{\partial \overline{\mathbf{y}}}$ represents the linearized time-stepping procedure and was found in Section A.2.1 to have the form of (2.1) and (2.3), with the (k,j)th $N \times N$ block being (see (A.83))

$$\frac{\partial \mathbf{t}_k}{\partial \mathbf{y}_j} = \begin{cases} \alpha_{k-j}^{(\sigma)}\mathbf{I}_N - \tau\,\beta_{k-j}^{(\sigma)}\dfrac{\partial \mathbf{f}_j^{\mathcal{E}}}{\partial \mathbf{y}} - \tau\,\gamma_{k-j}^{(\sigma)}\dfrac{\partial \mathbf{f}_j^{\mathcal{I}}}{\partial \mathbf{y}} & \text{if } k^\star \le j \le k \\ \mathbf{0}_{N\times N} & \text{otherwise,} \end{cases} \tag{5.3}$$

where $\beta_0^{(\sigma)} = 0$. We have let $\sigma = \min(k,s)$ and $k^\star = \max(1, k-s)$.

To solve (5.1) we compute, for $1 \le k \le K$,

$$\frac{\partial \mathbf{t}_k}{\partial \mathbf{y}_k}\boldsymbol{\xi}_k = \mathbf{q}_k - \sum_{j=k^\star}^{k-1} \frac{\partial \mathbf{t}_k}{\partial \mathbf{y}_j}\boldsymbol{\xi}_j.$$

The solution procedure is summarized in Algorithm 5.3.

Algorithm 5.3. The Linearized Implicit-Explicit Linear Multistep Method

Let $\sigma = \min(k, s)$ *and* $k^\star = \max(1, k-s)$.

For $k = 1, \cdots, K$, *solve for* $\boldsymbol{\xi}_k$:

$$\left(\mathbf{I}_N - \tau\gamma_0^{(\sigma)}\frac{\partial \mathbf{f}_k^{\mathcal{I}}}{\partial \mathbf{y}}\right)\boldsymbol{\xi}_k = \mathbf{q}_k - \sum_{j=k^\star}^{k-1}\left(\alpha_{k-j}^{(\sigma)}\mathbf{I}_N - \tau\beta_{k-j}^{(\sigma)}\frac{\partial \mathbf{f}_j^{\mathcal{E}}}{\partial \mathbf{y}} - \tau\gamma_{k-j}^{(\sigma)}\frac{\partial \mathbf{f}_j^{\mathcal{I}}}{\partial \mathbf{y}}\right)\boldsymbol{\xi}_j,$$

with $\dfrac{\partial \mathbf{f}_k^{\mathcal{I}/\mathcal{E}}}{\partial \mathbf{y}} = \dfrac{\partial \mathbf{f}^{\mathcal{I}/\mathcal{E}}(\mathbf{y}_k)}{\partial \mathbf{y}} = \dfrac{\partial \mathbf{f}^{\mathcal{I}/\mathcal{E}}(\mathbf{y}_k, t_k)}{\partial \mathbf{y}}$.

Examples

We find the linearized versions of the IMEX linear multistep methods mentioned in 2.2.1.

For the linearized IMEX BDF2 method one needs to solve, for $k = 2, \cdots, K$,

$$\left(\mathbf{I}_N - \tau\frac{2}{3}\frac{\partial \mathbf{f}^{\mathcal{I}}(\mathbf{y}_k)}{\partial \mathbf{y}}\right)\boldsymbol{\xi}_k = \mathbf{q}_k + \frac{4}{3}\left(\mathbf{I}_N + \tau\frac{\partial \mathbf{f}^{\mathcal{E}}(\mathbf{y}_{k-1})}{\partial \mathbf{y}}\right)\boldsymbol{\xi}_{k-1} + \\ - \frac{1}{3}\left(\mathbf{I}_N + 2\tau\frac{\partial \mathbf{f}^{\mathcal{E}}(\mathbf{y}_{k-2})}{\partial \mathbf{y}}\right)\boldsymbol{\xi}_{k-2}.$$

For the linearized IMEX BDF3 method one needs to solve, for $k = 3, \cdots, K$,

$$\left(\mathbf{I}_N - \tau\frac{6}{11}\frac{\partial \mathbf{f}^{\mathcal{I}}(\mathbf{y}_k)}{\partial \mathbf{y}}\right)\boldsymbol{\xi}_k = \mathbf{q}_k + \frac{18}{11}\left(\mathbf{I}_N + \tau\frac{\partial \mathbf{f}^{\mathcal{E}}(\mathbf{y}_{k-1})}{\partial \mathbf{y}}\right)\boldsymbol{\xi}_{k-1} + \\ - \frac{9}{11}\left(\mathbf{I}_N + 2\tau\frac{\partial \mathbf{f}^{\mathcal{E}}(\mathbf{y}_{k-2})}{\partial \mathbf{y}}\right)\boldsymbol{\xi}_{k-2} + \frac{2}{11}\left(\mathbf{I}_N + 3\tau\frac{\partial \mathbf{f}^{\mathcal{E}}(\mathbf{y}_{k-3})}{\partial \mathbf{y}}\right)\boldsymbol{\xi}_{k-3}.$$

Finally, for the linearized IMEX TVB3 method one needs to solve, for $k = 3, \cdots, K$,

$$\left(\mathbf{I}_N - \tau\frac{1089}{2048}\frac{\partial \mathbf{f}^{\mathcal{I}}(\mathbf{y}_k)}{\partial \mathbf{y}}\right)\boldsymbol{\xi}_k = \mathbf{q}_k + \\ + \left(\frac{3909}{2048}\mathbf{I}_N + \tau\frac{18463}{12288}\frac{\partial \mathbf{f}^{\mathcal{E}}(\mathbf{y}_{k-1})}{\partial \mathbf{y}} - \tau\frac{1139}{12288}\frac{\partial \mathbf{f}^{\mathcal{I}}(\mathbf{y}_{k-1})}{\partial \mathbf{y}}\right)\boldsymbol{\xi}_{k-1} +$$

$$+\left(-\frac{1367}{1024}\mathbf{I}_N-\tau\frac{1271}{768}\frac{\partial\mathbf{f}^{\mathcal{E}}\left(\mathbf{y}_{k-2}\right)}{\partial\mathbf{y}}-\tau\frac{367}{6144}\frac{\partial\mathbf{f}^{\mathcal{I}}\left(\mathbf{y}_{k-2}\right)}{\partial\mathbf{y}}\right)\boldsymbol{\xi}_{k-2}+$$
$$+\left(\frac{873}{2048}\mathbf{I}_N+\tau\frac{8233}{12288}\frac{\partial\mathbf{f}^{\mathcal{E}}\left(\mathbf{y}_{k-3}\right)}{\partial\mathbf{y}}+\tau\frac{1699}{12288}\frac{\partial\mathbf{f}^{\mathcal{I}}\left(\mathbf{y}_{k-3}\right)}{\partial\mathbf{y}}\right)\boldsymbol{\xi}_{k-3}.$$

5.2.2 IMEX Runge-Kutta Methods

We let $\overline{\boldsymbol{\xi}}=\begin{bmatrix}\widehat{\boldsymbol{\xi}}_1^\top & \cdots & \widehat{\boldsymbol{\xi}}_K^\top\end{bmatrix}^\top$ represent the linearized forward solution, with $\widehat{\boldsymbol{\xi}}_k=\begin{bmatrix}\boldsymbol{\Xi}_k^\top & \boldsymbol{\xi}_k^\top\end{bmatrix}^\top$, where

$$\boldsymbol{\Xi}_k=\begin{bmatrix}{\boldsymbol{\Xi}_{k,1}^{\mathcal{E}}}^\top & {\boldsymbol{\Xi}_{k,1}^{\mathcal{I}}}^\top & {\boldsymbol{\Xi}_{k,2}^{\mathcal{E}}}^\top & \cdots & {\boldsymbol{\Xi}_{k,s-1}^{\mathcal{I}}}^\top & {\boldsymbol{\Xi}_{k,s}^{\mathcal{E}}}^\top\end{bmatrix}^\top$$

represents the $2s-1$ internal stages of the linearized forward solution at the kth time step. The source $\overline{\mathbf{q}}$ is defined similarly, with internal stages $\mathbf{Q}_{k,i}$.

The $(s+1)KN\times(s+1)KN$ matrix $\dfrac{\partial\mathbf{t}}{\partial\overline{\mathbf{y}}}$ was found in A.1.2 to be (see (A.48))

$$\frac{\partial\mathbf{t}}{\partial\overline{\mathbf{y}}}=\begin{bmatrix}\mathbf{A}_1 & & & & & & \\ \mathbf{B}^\top & \mathbf{I}_N & & & & & \\ & \mathbf{C}_2 & \mathbf{A}_2 & & & & \\ & -\mathbf{I}_N & \mathbf{B}^\top & \mathbf{I}_N & & & \\ & & \ddots & \ddots & \ddots & & \\ & & & & \mathbf{C}_K & \mathbf{A}_K & \\ & & & & -\mathbf{I}_N & \mathbf{B}^\top & \mathbf{I}_N\end{bmatrix},$$

where (see (A.127))

$$\mathbf{A}_k=\mathbf{I}_{(2s-1)N}-\tau_k\frac{\partial\mathbf{f}_k}{\partial\mathbf{y}}\left(\mathbf{A}_s\otimes\mathbf{I}_N\right)\qquad\text{and}\qquad\mathbf{C}_k=-\frac{\partial\mathbf{f}_k}{\partial\mathbf{y}}\left(\mathbf{1}_{2s-1}\otimes\mathbf{I}_N\right)$$

and

$$\mathbf{B}=-\tau_k\begin{bmatrix}b_1^{\mathcal{E}} & b_1^{\mathcal{I}} & b_2^{\mathcal{E}} & \cdots & b_{s-1}^{\mathcal{E}} & b_{s-1}^{\mathcal{I}} & b_s^{\mathcal{E}}\end{bmatrix}^\top\otimes\mathbf{I}_N.$$

See (A.125) for the definition of $\mathbf{A}_s$. $\dfrac{\partial \mathbf{f}_k}{\partial \mathbf{y}}$ was defined to be

$$\frac{\partial \mathbf{f}_k}{\partial \mathbf{y}} = \text{blkdiag}\left(\frac{\partial \mathbf{f}_{k,1}^{\mathcal{E}}}{\partial \mathbf{y}}, \frac{\partial \mathbf{f}_{k,2}^{\mathcal{I}}}{\partial \mathbf{y}}, \frac{\partial \mathbf{f}_{k,2}^{\mathcal{E}}}{\partial \mathbf{y}}, \frac{\partial \mathbf{f}_{k,3}^{\mathcal{I}}}{\partial \mathbf{y}}, \cdots, \frac{\partial \mathbf{f}_{k,s}^{\mathcal{E}}}{\partial \mathbf{y}}\right),$$

with $\mathbf{f}_{k,\sigma}^{\mathcal{E}} = \mathbf{f}^{\mathcal{E}}\left(\underline{\mathbf{y}}_{k,\sigma-1}, t_{k-1} + c_\sigma \tau_k\right)$, $\mathbf{f}_{k,\sigma}^{\mathcal{I}} = \mathbf{f}^{\mathcal{I}}\left(\underline{\mathbf{y}}_{k,\sigma-1}, t_{k-1} + c_\sigma \tau_k\right)$ and $\underline{\mathbf{y}}_{k,\sigma} = \mathbf{y}_{k-1} + \tau_k \sum_{i=1}^{\sigma} a_{\sigma+1,i}^{\mathcal{E}} \mathbf{Y}_{k,i}^{\mathcal{E}} + \tau_k \sum_{i=1}^{\sigma} a_{\sigma,i}^{\mathcal{I}} \mathbf{Y}_{k,i}^{\mathcal{I}}$.

The system (5.1) is then obtained by solving, for $1 \leq k \leq K$, the internal stages

$$\begin{aligned} \mathbf{A}_k \boldsymbol{\Xi}_k &= \mathbf{Q}_k - \mathbf{C}_k \boldsymbol{\xi}_{k-1} \\ \Rightarrow \qquad \left(\mathbf{I}_{(2s-1)N} - \tau_k \frac{\partial \mathbf{f}_k}{\partial \mathbf{y}} \left(\mathbf{A}_s \otimes \mathbf{I}_N\right)\right) \boldsymbol{\Xi}_k &= \mathbf{Q}_k + \frac{\partial \mathbf{f}_k}{\partial \mathbf{y}} \left(\mathbf{1}_{2s-1} \otimes \mathbf{I}_N\right) \boldsymbol{\xi}_{k-1} \end{aligned}$$

and then computing the next step of the linearized solution,

$$\boldsymbol{\xi}_k = \mathbf{q}_k + \boldsymbol{\xi}_{k-1} - \mathbf{B}^\top \boldsymbol{\Xi}_k.$$

The detailed solution procedure is presented in Algorithm 5.4.

Algorithm 5.4. The Linearized Implicit-Explicit Runge-Kutta Method

Set $\boldsymbol{\xi}_0 = \mathbf{0}_{N\times 1}$ *and recall that we defined*

$$\underline{\mathbf{y}}_{k,\sigma} = \mathbf{y}_{k-1} + \tau_k \sum_{i=1}^{\sigma} a_{\sigma+1,i}^{\mathcal{E}} \mathbf{Y}_{k,i}^{\mathcal{E}} + \tau_k \sum_{i=1}^{\sigma} a_{\sigma,i}^{\mathcal{I}} \mathbf{Y}_{k,i}^{\mathcal{I}}.$$

For $k = 1, \cdots, K$*:*

- *For* $i = 1, \cdots, 2s-1$*, evaluate the internal stages:*

- *if i odd, set $\sigma = \lceil \frac{i}{2} \rceil$ and compute:*

$$\boldsymbol{\Xi}^{\mathcal{E}}_{k,\sigma} = \mathbf{Q}_{k,\sigma} + \frac{\partial \mathbf{f}^{\mathcal{E}}\left(\underline{\mathbf{y}}_{k,\sigma-1}\right)}{\partial \mathbf{y}} \left(\boldsymbol{\xi}_{k-1} + \tau_k \sum_{j=1}^{\sigma-1} a^{\mathcal{E}}_{\sigma j}\, \boldsymbol{\Xi}^{\mathcal{E}}_{k,j} + \tau_k \sum_{j=1}^{\sigma-1} a^{\mathcal{I}}_{\sigma-1,j}\, \boldsymbol{\Xi}^{\mathcal{I}}_{k,j} \right).$$

- *if i even, set $\sigma = \frac{i}{2}$ and solve:*

$$\left(\mathbf{I}_N - \tau_k a^{\mathcal{I}}_{\sigma\sigma} \frac{\partial \mathbf{f}^{\mathcal{I}}\left(\underline{\mathbf{y}}_{k,\sigma}\right)}{\partial \mathbf{y}} \right) \boldsymbol{\Xi}^{\mathcal{I}}_{k,\sigma} =$$

$$= \mathbf{Q}_{k,\sigma} + \frac{\partial \mathbf{f}^{\mathcal{I}}\left(\underline{\mathbf{y}}_{k,\sigma}\right)}{\partial \mathbf{y}} \left(\boldsymbol{\xi}_{k-1} + \tau_k \sum_{j=1}^{\sigma} a^{\mathcal{E}}_{\sigma+1,j}\, \boldsymbol{\Xi}^{\mathcal{E}}_{kj} + \tau_k \sum_{j=1}^{\sigma-1} a^{\mathcal{I}}_{\sigma j}\, \boldsymbol{\Xi}^{\mathcal{I}}_{kj} \right).$$

- *Compute*

$$\boldsymbol{\xi}_k = \mathbf{q}_k + \boldsymbol{\xi}_{k-1} + \tau_k \sum_{\sigma=1}^{s} b^{\mathcal{E}}_{\sigma}\, \boldsymbol{\Xi}^{\mathcal{E}}_{k,\sigma} + \tau_k \sum_{\sigma=1}^{s-1} b^{\mathcal{I}}_{\sigma}\, \boldsymbol{\Xi}^{\mathcal{I}}_{k,\sigma}.$$

Example

The linearized ARS3 scheme is, with $\gamma = \frac{3+\sqrt{3}}{6}$,

$$\begin{aligned}
\underline{\mathbf{y}}_{k,1} &= \mathbf{y}_{k-1} + \tau_k \gamma \mathbf{Y}^{\mathcal{E}}_{k,1} + \tau_k \gamma \mathbf{Y}^{\mathcal{I}}_{k,1} \\
\underline{\mathbf{y}}_{k,2} &= \mathbf{y}_{k-1} + \tau_k \left((\gamma - 1)\mathbf{Y}^{\mathcal{E}}_{k,1} + 2(1-\gamma)\mathbf{Y}^{\mathcal{E}}_{k,2} + (1-2\gamma)\mathbf{Y}^{\mathcal{I}}_{k,1} + \gamma \mathbf{Y}^{\mathcal{I}}_{k,2} \right),
\end{aligned}$$

and for $k = 1, \cdots, K$:

- Compute the internal stages for $\boldsymbol{\xi}$:

  - Compute $\boldsymbol{\Xi}^{\mathcal{E}}_{k,1}$:

$$\boldsymbol{\Xi}^{\mathcal{E}}_{k,1} = \mathbf{Q}_{k,1} + \frac{\partial \mathbf{f}^{\mathcal{E}}\left(\mathbf{y}_{k-1}\right)}{\partial \mathbf{y}} \boldsymbol{\xi}_{k-1};$$

- Solve for $\boldsymbol{\Xi}^{\mathcal{I}}_{k,1}$:

$$\left(\mathbf{I}_N - \tau_k\gamma\frac{\partial \mathbf{f}^{\mathcal{I}}\left(\underline{\mathbf{y}}_{k,1}\right)}{\partial \mathbf{y}}\right)\boldsymbol{\Xi}^{\mathcal{I}}_{k,1} = \mathbf{Q}_{k,1} + \frac{\partial \mathbf{f}^{\mathcal{I}}\left(\underline{\mathbf{y}}_{k,1}\right)}{\partial \mathbf{y}}\left(\boldsymbol{\xi}_{k-1} + \tau_k\gamma\boldsymbol{\Xi}^{\mathcal{E}}_{k,1}\right);$$

- Compute $\boldsymbol{\Xi}^{\mathcal{E}}_{k,2}$:

$$\boldsymbol{\Xi}^{\mathcal{E}}_{k,2} = \mathbf{Q}_{k,2} + \frac{\partial \mathbf{f}^{\mathcal{E}}\left(\underline{\mathbf{y}}_{k,1}\right)}{\partial \mathbf{y}}\left(\boldsymbol{\xi}_{k-1} + \tau_k\gamma\boldsymbol{\Xi}^{\mathcal{E}}_{k,1} + \tau_k\gamma\boldsymbol{\Xi}^{\mathcal{I}}_{k,1}\right);$$

- Solve for $\boldsymbol{\Xi}^{\mathcal{I}}_{k,2}$:

$$\left(\mathbf{I}_{N\times 1} - \tau_k\gamma\frac{\partial \mathbf{f}^{\mathcal{I}}\left(\underline{\mathbf{y}}_{k,2}\right)}{\partial \mathbf{y}}\right)\boldsymbol{\Xi}^{\mathcal{I}}_{k,2} = \mathbf{Q}_{k,2} + \frac{\partial \mathbf{f}^{\mathcal{I}}\left(\underline{\mathbf{y}}_{k,2}\right)}{\partial \mathbf{y}}\left(\boldsymbol{\xi}_{k-1} + \right.$$
$$\left. + \tau_k\left((\gamma-1)\boldsymbol{\Xi}^{\mathcal{E}}_{k,1} + 2(1-\gamma)\boldsymbol{\Xi}^{\mathcal{E}}_{k,2} + (1-2\gamma)\boldsymbol{\Xi}^{\mathcal{I}}_{k,1}\right)\right);$$

- Compute $\boldsymbol{\Xi}^{\mathcal{E}}_{k,2}$:

$$\boldsymbol{\Xi}^{\mathcal{E}}_{k,3} = \mathbf{Q}_{k,3} + \frac{\partial \mathbf{f}^{\mathcal{E}}\left(\underline{\mathbf{y}}_{k,2}\right)}{\partial \mathbf{y}}\left(\boldsymbol{\xi}_{k-1} + \right.$$
$$\left. + \tau_k\left((\gamma-1)\boldsymbol{\Xi}^{\mathcal{E}}_{k,1} + 2(1-\gamma)\boldsymbol{\Xi}^{\mathcal{E}}_{k,2} + (1-2\gamma)\boldsymbol{\Xi}^{\mathcal{I}}_{k,1} + \gamma\boldsymbol{\Xi}^{\mathcal{I}}_{k,2}\right)\right).$$

- Update $\boldsymbol{\xi}$:

$$\boldsymbol{\xi}_k = \mathbf{q}_k + \boldsymbol{\xi}_{k-1} + \frac{\tau_k}{2}(\boldsymbol{\Xi}^{\mathcal{E}}_{k,2} + \boldsymbol{\Xi}^{\mathcal{E}}_{k,3} + \boldsymbol{\Xi}^{\mathcal{I}}_{k,1} + \boldsymbol{\Xi}^{\mathcal{I}}_{k,2}).$$

5.3 Staggered Time-Stepping Methods

We use the same approach used in the previous subsections to find the linearized StagLM and StagRK methods.

5.3.1 Staggered Multistep Methods

Let $\overline{\boldsymbol{\xi}} = \begin{bmatrix} \boldsymbol{\xi}_1^\top & \cdots & \boldsymbol{\xi}_K^\top \end{bmatrix}^\top$ represent the linearized forward solution and $\overline{\mathbf{q}} = \begin{bmatrix} \mathbf{q}_1^\top & \cdots & \mathbf{q}_K^\top \end{bmatrix}^\top$ the source. We have $\boldsymbol{\xi}_k = \begin{bmatrix} \boldsymbol{\xi}_k^{\mathbf{u}\top} & \boldsymbol{\xi}_{k+\frac{1}{2}}^{\mathbf{v}\top} \end{bmatrix}^\top$ and likewise for $\boldsymbol{\theta}_k$.

The $K \times K$ block matrix $\dfrac{\partial \mathbf{t}}{\partial \overline{\mathbf{y}}}$ is a lower block triangular matrix that has the form of the template matrices (2.1) and (2.3). Its (k, j)th $N \times N$ block is (see (A.158))

$$\frac{\partial \mathbf{t}_k}{\partial \mathbf{y}_j} = \begin{cases} \alpha_{k-j}^{(\sigma)} \mathbf{I}_N - \tau\, \beta_{k-j}^{(\sigma)} \dfrac{\partial \mathbf{f}_j}{\partial \mathbf{y}_j} - \tau\, \beta_{k-j-1}^{(\sigma)} \dfrac{\partial \mathbf{f}_{j+1}}{\partial \mathbf{y}_j} & \text{if } k^\star \leq j \leq k \\ \mathbf{0}_{N \times N} & \text{otherwise,} \end{cases} \tag{5.4}$$

with $\beta_\ell^{(\sigma)} = 0$ for $\ell < 0$, $\sigma = \min(k, s)$, $k^\star = \max(1, k - s)$, and $\dfrac{\partial \mathbf{f}_k}{\partial \mathbf{y}_k}$ and $\dfrac{\partial \mathbf{f}_{k+1}}{\partial \mathbf{y}_k}$ defined as in (A.157).

To solve (5.1) we compute, for $1 \leq k \leq K$,

$$\frac{\partial \mathbf{t}_k}{\partial \mathbf{y}_k} \boldsymbol{\xi}_k = \mathbf{q}_k - \sum_{j=k^\star}^{k-1} \frac{\partial \mathbf{t}_k}{\partial \mathbf{y}_j} \boldsymbol{\xi}_j.$$

The solution procedure is summarized in Algorithm 5.5.

Algorithm 5.5. The Linearized Staggered Linear Multistep Method

Let $\sigma = \min(k, s)$, $k^\star = \max(1, k - s)$.

For $k = 1, \cdots, K$*:*

- *Update* $\boldsymbol{\xi}^{\mathbf{u}}$*:*

$$\boldsymbol{\xi}_k^{\mathbf{u}} = \mathbf{q}_k^{\mathbf{u}} - \sum_{j=k^\star}^{k-1} \alpha_{k-j}^{(\sigma)} \boldsymbol{\xi}_j^{\mathbf{u}} + \tau \sum_{j=k^\star-1}^{k-1} \beta_{k-j-1}^{(\sigma)} \frac{\partial \mathbf{f}^{\mathbf{u}}\left(\mathbf{v}_{j+\frac{1}{2}}, t_{j+\frac{1}{2}}\right)}{\partial \mathbf{v}} \boldsymbol{\xi}_{j+\frac{1}{2}}^{\mathbf{v}}$$

- *Update* $\boldsymbol{\xi}^{\mathbf{v}}$*:*

$$\boldsymbol{\xi}^{\mathbf{v}}_{k+\frac{1}{2}} = \mathbf{q}^{\mathbf{v}}_{k+\frac{1}{2}} - \sum_{j=k^\star}^{k-1} \alpha^{(\sigma)}_{k-j} \boldsymbol{\xi}^{\mathbf{v}}_{j+\frac{1}{2}} + \tau \sum_{j=k^\star}^{k} \beta^{(\sigma)}_{k-j} \frac{\partial \mathbf{f}^{\mathbf{v}}\left(\mathbf{u}_j, t_j\right)}{\partial \mathbf{u}} \boldsymbol{\xi}^{\mathbf{u}}_j.$$

Examples

We find the linearized versions of the staggered linear multistep methods mentioned in 2.3.1. The linearized leapfrog method for $k = 1, \cdots, K$ is

$$\begin{aligned}
\boldsymbol{\xi}^{\mathbf{u}}_k &= \mathbf{q}^{\mathbf{u}}_k + \boldsymbol{\xi}^{\mathbf{u}}_{k-1} + \tau \frac{\partial \mathbf{f}^{\mathbf{u}}\left(\mathbf{v}_{k-\frac{1}{2}}, t_{k-\frac{1}{2}}\right)}{\partial \mathbf{v}} \boldsymbol{\xi}^{\mathbf{v}}_{k-\frac{1}{2}} \\
\boldsymbol{\xi}^{\mathbf{v}}_{k+\frac{1}{2}} &= \mathbf{q}^{\mathbf{v}}_{k+\frac{1}{2}} + \boldsymbol{\xi}^{\mathbf{v}}_{k-\frac{1}{2}} + \tau \frac{\partial \mathbf{f}^{\mathbf{v}}\left(\mathbf{u}_k, t_k\right)}{\partial \mathbf{u}} \boldsymbol{\xi}^{\mathbf{u}}_k.
\end{aligned}$$

The linearized StagBDF3 method for $k = 3, \cdots, K$ is

$$\begin{aligned}
\boldsymbol{\xi}^{\mathbf{u}}_k &= \mathbf{q}^{\mathbf{u}}_k + \frac{21}{23}\boldsymbol{\xi}^{\mathbf{u}}_{k-1} + \frac{3}{23}\boldsymbol{\xi}^{\mathbf{u}}_{k-2} - \frac{1}{23}\boldsymbol{\xi}^{\mathbf{u}}_{k-3} + \tau \frac{24}{23} \frac{\partial \mathbf{f}^{\mathbf{u}}\left(\mathbf{v}_{k-\frac{1}{2}}, t_{k-\frac{1}{2}}\right)}{\partial \mathbf{v}} \boldsymbol{\xi}^{\mathbf{v}}_{k-\frac{1}{2}} \\
\boldsymbol{\xi}^{\mathbf{v}}_{k+\frac{1}{2}} &= \mathbf{q}^{\mathbf{v}}_{k+\frac{1}{2}} + \frac{21}{23}\boldsymbol{\xi}^{\mathbf{v}}_{k-\frac{1}{2}} + \frac{3}{23}\boldsymbol{\xi}^{\mathbf{v}}_{k-\frac{3}{2}} - \frac{1}{23}\boldsymbol{\xi}^{\mathbf{v}}_{k-\frac{5}{2}} + \tau \frac{24}{23} \frac{\partial \mathbf{f}^{\mathbf{v}}\left(\mathbf{u}_k, t_k\right)}{\partial \mathbf{u}} \boldsymbol{\xi}^{\mathbf{u}}_k.
\end{aligned}$$

The linearized StagBDF4 method for $k = 4, \cdots, K$ is

$$\begin{aligned}
\boldsymbol{\xi}^{\mathbf{u}}_k &= \mathbf{q}^{\mathbf{u}}_k + \frac{17}{22}\boldsymbol{\xi}^{\mathbf{u}}_{k-1} + \frac{9}{22}\boldsymbol{\xi}^{\mathbf{u}}_{k-2} - \frac{5}{22}\boldsymbol{\xi}^{\mathbf{u}}_{k-3} + \frac{1}{22}\boldsymbol{\xi}^{\mathbf{u}}_{k-4} + \tau \frac{12}{11} \frac{\partial \mathbf{f}^{\mathbf{u}}\left(\mathbf{v}_{k-\frac{1}{2}}, t_{k-\frac{1}{2}}\right)}{\partial \mathbf{v}} \boldsymbol{\xi}^{\mathbf{v}}_{k-\frac{1}{2}} \\
\boldsymbol{\xi}^{\mathbf{v}}_{k+\frac{1}{2}} &= \mathbf{q}^{\mathbf{v}}_{k+\frac{1}{2}} + \frac{17}{22}\boldsymbol{\xi}^{\mathbf{v}}_{k-\frac{1}{2}} + \frac{9}{22}\boldsymbol{\xi}^{\mathbf{v}}_{k-\frac{3}{2}} - \frac{5}{22}\boldsymbol{\xi}^{\mathbf{v}}_{k-\frac{5}{2}} + \frac{1}{22}\boldsymbol{\xi}^{\mathbf{v}}_{k-\frac{7}{2}} + \tau \frac{12}{11} \frac{\partial \mathbf{f}^{\mathbf{v}}\left(\mathbf{u}_k, t_k\right)}{\partial \mathbf{u}} \boldsymbol{\xi}^{\mathbf{u}}_k.
\end{aligned}$$

The linearized StagAB3 scheme for $k = 3, \cdots, K$ is

$$\begin{aligned}
\boldsymbol{\xi}^{\mathbf{u}}_k = \mathbf{q}^{\mathbf{u}}_k + \boldsymbol{\xi}^{\mathbf{u}}_{k-1} + \tau \frac{25}{24} \frac{\partial \mathbf{f}^{\mathbf{u}}\left(\mathbf{v}_{k-\frac{1}{2}}, t_{k-\frac{1}{2}}\right)}{\partial \mathbf{v}} \boldsymbol{\xi}^{\mathbf{v}}_{k-\frac{1}{2}} - \tau \frac{1}{12} \frac{\partial \mathbf{f}^{\mathbf{u}}\left(\mathbf{v}_{k-\frac{3}{2}}, t_{k-\frac{3}{2}}\right)}{\partial \mathbf{v}} \boldsymbol{\xi}^{\mathbf{v}}_{k-\frac{3}{2}} + \\
+ \tau \frac{1}{24} \frac{\partial \mathbf{f}^{\mathbf{u}}\left(\mathbf{v}_{k-\frac{5}{2}}, t_{k-\frac{5}{2}}\right)}{\partial \mathbf{v}} \boldsymbol{\xi}^{\mathbf{v}}_{k-\frac{5}{2}}
\end{aligned}$$

$$\boldsymbol{\xi}^{\mathbf{v}}_{k+\frac{1}{2}} = \mathbf{q}^{\mathbf{v}}_{k+\frac{1}{2}} + \boldsymbol{\xi}^{\mathbf{v}}_{k-\frac{1}{2}} + \tau\frac{25}{24}\frac{\partial \mathbf{f}^{\mathbf{v}}\left(\mathbf{u}_k, t_k\right)}{\partial \mathbf{u}}\boldsymbol{\xi}^{\mathbf{u}}_k - \tau\frac{1}{12}\frac{\partial \mathbf{f}^{\mathbf{v}}\left(\mathbf{u}_{k-1}, t_{k-1}\right)}{\partial \mathbf{u}}\boldsymbol{\xi}^{\mathbf{u}}_{k-1} + \\ + \tau\frac{1}{24}\frac{\partial \mathbf{f}^{\mathbf{v}}\left(\mathbf{u}_{k-2}, t_{k-2}\right)}{\partial \mathbf{u}}\boldsymbol{\xi}^{\mathbf{u}}_{k-2}.$$

Finally, the linearized StagAB4 scheme for $k = 4, \cdots, K$ is

$$\boldsymbol{\xi}^{\mathbf{u}}_k = \mathbf{q}^{\mathbf{u}}_k + \boldsymbol{\xi}^{\mathbf{u}}_{k-1} + \tau\frac{13}{12}\frac{\partial \mathbf{f}^{\mathbf{u}}\left(\mathbf{v}_{k-\frac{1}{2}}, t_{k-\frac{1}{2}}\right)}{\partial \mathbf{v}}\boldsymbol{\xi}^{\mathbf{v}}_{k-\frac{1}{2}} - \tau\frac{5}{24}\frac{\partial \mathbf{f}^{\mathbf{u}}\left(\mathbf{v}_{k-\frac{3}{2}}, t_{k-\frac{3}{2}}\right)}{\partial \mathbf{v}}\boldsymbol{\xi}^{\mathbf{v}}_{k-\frac{3}{2}} + \\ + \tau\frac{1}{6}\frac{\partial \mathbf{f}^{\mathbf{u}}\left(\mathbf{v}_{k-\frac{5}{2}}, t_{k-\frac{5}{2}}\right)}{\partial \mathbf{v}}\boldsymbol{\xi}^{\mathbf{v}}_{k-\frac{5}{2}} - \tau\frac{1}{24}\frac{\partial \mathbf{f}^{\mathbf{u}}\left(\mathbf{v}_{k-\frac{7}{2}}, t_{k-\frac{7}{2}}\right)}{\partial \mathbf{v}}\boldsymbol{\xi}^{\mathbf{v}}_{k-\frac{7}{2}}$$

$$\boldsymbol{\xi}^{\mathbf{v}}_{k+\frac{1}{2}} = \mathbf{q}^{\mathbf{v}}_{k+\frac{1}{2}} + \boldsymbol{\xi}^{\mathbf{v}}_{k-\frac{1}{2}} + \tau\frac{13}{12}\frac{\partial \mathbf{f}^{\mathbf{v}}\left(\mathbf{u}_k, t_k\right)}{\partial \mathbf{u}}\boldsymbol{\xi}^{\mathbf{u}}_k - \tau\frac{5}{24}\frac{\partial \mathbf{f}^{\mathbf{v}}\left(\mathbf{u}_{k-1}, t_{k-1}\right)}{\partial \mathbf{u}}\boldsymbol{\xi}^{\mathbf{u}}_{k-1} + \\ + \tau\frac{1}{6}\frac{\partial \mathbf{f}^{\mathbf{v}}\left(\mathbf{u}_{k-2}, t_{k-2}\right)}{\partial \mathbf{u}}\boldsymbol{\xi}^{\mathbf{u}}_{k-2} - \tau\frac{1}{24}\frac{\partial \mathbf{f}^{\mathbf{v}}\left(\mathbf{u}_{k-3}, t_{k-3}\right)}{\partial \mathbf{u}}\boldsymbol{\xi}^{\mathbf{u}}_{k-3}.$$

5.3.2 Staggered Runge-Kutta Methods

We let $\overline{\boldsymbol{\xi}} = \begin{bmatrix}\widehat{\boldsymbol{\xi}}_1^\top & \cdots & \widehat{\boldsymbol{\xi}}_K^\top\end{bmatrix}^\top$ represent the adjoint solution, with $\widehat{\boldsymbol{\xi}}_k = \begin{bmatrix}\boldsymbol{\Xi}_k^\top & \boldsymbol{\xi}_k^\top\end{bmatrix}^\top$, where

$$\boldsymbol{\xi}_k = \begin{bmatrix}\boldsymbol{\xi}^{\mathbf{u}\top}_k & \boldsymbol{\xi}^{\mathbf{v}}_{k+\frac{1}{2}}{}^\top\end{bmatrix}^\top \qquad \text{and} \qquad \boldsymbol{\Xi}_{k,\sigma} = \begin{bmatrix}\boldsymbol{\Xi}^{\mathbf{u}}_{k,\sigma}{}^\top & \boldsymbol{\Xi}^{\mathbf{v}}_{k+\frac{1}{2},\sigma}{}^\top\end{bmatrix}^\top,$$

the latter representing the s internal stages of the linearized forward solution at the kth time-step. The source $\overline{\mathbf{q}}$ is defined similarly, with internal stages $\mathbf{Q}_{k,\sigma}$.

The linearization of the time-stepping equation, $\dfrac{\partial \mathbf{t}}{\partial \overline{\mathbf{y}}}$, was derived in Section A.3.2, see (A.196):

$$\frac{\partial \mathbf{t}}{\partial \overline{\mathbf{y}}}^\top = \begin{bmatrix} \mathbf{A}_1 & \mathbf{C}_{1,1} & & & & & \\ \mathbf{B}^\top & \mathbf{I}_N & & & & & \\ & \mathbf{C}_{2,1} & \mathbf{A}_2 & \mathbf{C}_{2,2} & & & \\ & -\mathbf{I}_N & \mathbf{B}^\top & \mathbf{I}_N & & & \\ & & \ddots & \ddots & \ddots & & \\ & & & & \mathbf{C}_{K,K-1} & \mathbf{A}_K & \mathbf{C}_{K,K} \\ & & & & -\mathbf{I}_N & \mathbf{B}^\top & \mathbf{I}_N \end{bmatrix}.$$

$\mathbf{A}_k$ and $\mathbf{C}_{k,j}$ were defined in (A.195) to be

$$\begin{aligned}\mathbf{A}_k &= \mathbf{I}_{sN} - \tau\,\mathbf{G}_k^{\mathrm{o/e}}\left(\mathbf{A}_s \otimes \mathbf{I}_N\right)\\ \mathbf{C}_{k,j} &= -\frac{\partial \mathbf{F}_k^{\mathrm{o/e}}}{\partial \mathbf{y}_j}\\ \mathbf{B} &= \mathbf{B}^{\mathrm{o/e}};\end{aligned}$$

see Section A.3.2 for the definitions of $\mathbf{G}_k^{\mathrm{o/e}}$ and $\dfrac{\partial \mathbf{F}_k^{\mathrm{o/e}}}{\partial \mathbf{y}_j}$, which contain the derivatives of $\mathbf{f}^{\mathbf{u}}$ and $\mathbf{f}^{\mathbf{v}}$ with respect to $\mathbf{u}$ and $\mathbf{v}$. Important for this is the shorthand

$$\underline{\mathbf{v}}^{o/e}_{\mathbf{U},k-\frac{1}{2},\sigma} = \mathbf{v}_{k-\frac{1}{2}} + \tau \sum_{\substack{i=1\\ i\text{ even}}}^{\sigma-1} a_{\sigma i}\mathbf{U}^{o/e}_{k,i}, \qquad \underline{\mathbf{v}}^{o/e}_{\mathbf{V},k-\frac{1}{2},\sigma} = \mathbf{v}_{k-\frac{1}{2}} + \tau \sum_{\substack{i=1\\ i\text{ odd}}}^{\sigma-1} a_{\sigma i}\mathbf{V}^{o/e}_{k+\frac{1}{2},i}$$

$$\underline{\mathbf{u}}^{o/e}_{\mathbf{U},k-1,\sigma} = \mathbf{u}_{k-1} + \tau \sum_{\substack{i=1\\ i\text{ odd}}}^{\sigma-1} a_{\sigma i}\mathbf{U}^{o/e}_{k,i}, \qquad \underline{\mathbf{u}}^{o/e}_{\mathbf{V},k,\sigma} = \mathbf{u}_{k} + \tau \sum_{\substack{i=1\\ i\text{ even}}}^{\sigma-1} a_{\sigma i}\mathbf{V}^{o/e}_{k+\frac{1}{2},i},$$

where o/e denotes whether the number of stages s is odd or even.

While not apparent from the form above, $\dfrac{\partial \mathbf{t}}{\partial \overline{\mathbf{y}}}$ is essentially lower (block) triangular because we can swap rows and columns to make it lower block triangular, which is in fact what we do to get to the algorithm below. Conceptually, we solve (5.1) as follows:

$$\begin{aligned}\mathbf{A}_k\,\boldsymbol{\Xi}_k &= \mathbf{Q}_k + \mathbf{C}_{k,k}\,\boldsymbol{\xi}_k + \mathbf{C}_{k,k-1}\,\boldsymbol{\xi}_{k-1}\\ \boldsymbol{\xi}_k &= \mathbf{q}_k + \boldsymbol{\xi}_{k-1} + \mathbf{B}^\top\boldsymbol{\Xi}_k\end{aligned} \tag{5.5}$$

for $i, \cdots, K$. The detailed solution procedure for s odd is summarized in Algorithm 6.6.

Algorithm 5.6. The Linearized Staggered Runge-Kutta Method, with s odd

Set $\boldsymbol{\xi}_0 = \mathbf{0}_{N\times 1}$. *Recall that (see* (2.44)*)*

$$\underline{\mathbf{v}}_{\mathbf{U},k-\frac{1}{2},\sigma} = \mathbf{v}_{k-\frac{1}{2}} + \tau \sum_{\substack{i=1\\ i \text{ even}}}^{\sigma-1} a_{\sigma i}\mathbf{U}_{k,i}, \qquad \underline{\mathbf{v}}_{\mathbf{V},k-\frac{1}{2},\sigma} = \mathbf{v}_{k-\frac{1}{2}} + \tau \sum_{\substack{i=1\\ i \text{ odd}}}^{\sigma-1} a_{\sigma i}\mathbf{V}_{k+\frac{1}{2},i}$$

$$\underline{\mathbf{u}}_{\mathbf{U},k-1,\sigma} = \mathbf{u}_{k-1} + \tau \sum_{\substack{i=1\\ i \text{ odd}}}^{\sigma-1} a_{\sigma i}\mathbf{U}_{k,i}, \qquad \underline{\mathbf{u}}_{\mathbf{V},k,\sigma} = \mathbf{u}_{k} + \tau \sum_{\substack{i=1\\ i \text{ even}}}^{\sigma-1} a_{\sigma i}\mathbf{V}_{k+\frac{1}{2},i}.$$

For $k = 1, \cdots, K$*:*

- *For* $\sigma = 1, \cdots, s$*, evaluate the internal stages for* $\boldsymbol{\xi}^{\mathbf{u}}$*:*

$$\boldsymbol{\Xi}^{\mathbf{u}}_{k,\sigma} = \mathbf{Q}^{\mathbf{u}}_{k,\sigma} + \begin{cases} \dfrac{\partial \mathbf{f}^{\mathbf{u}}\left(\underline{\mathbf{v}}_{\mathbf{U},k-1/2,\sigma}\right)}{\partial \mathbf{v}} \left(\boldsymbol{\xi}^{\mathbf{v}}_{k-\frac{1}{2}} + \tau \sum\limits_{\substack{i=1\\ i \text{ even}}}^{\sigma-1} a_{\sigma i}\,\boldsymbol{\Xi}^{\mathbf{u}}_{k,i}\right) & \text{if } \sigma \text{ odd} \\ \dfrac{\partial \mathbf{f}^{\mathbf{v}}\left(\underline{\mathbf{u}}_{\mathbf{U},k-1,\sigma}\right)}{\partial \mathbf{u}} \left(\boldsymbol{\xi}^{\mathbf{u}}_{k-1} + \tau \sum\limits_{\substack{i=1\\ i \text{ odd}}}^{\sigma-1} a_{\sigma i}\,\boldsymbol{\Xi}^{\mathbf{u}}_{k,i}\right) & \text{if } \sigma \text{ even} \end{cases}$$

- *Compute*

$$\boldsymbol{\xi}^{\mathbf{u}}_{k} = \mathbf{q}^{\mathbf{u}}_{k} + \boldsymbol{\xi}^{\mathbf{u}}_{k-1} + \tau \sum_{\substack{\sigma=1\\ \sigma \text{ odd}}}^{s} b_\sigma \boldsymbol{\Xi}^{\mathbf{u}}_{k,\sigma}.$$

- *For* $\sigma = 1, \cdots, s$*, evaluate the internal stages for* $\boldsymbol{\xi}^{\mathbf{v}}$*:*

$$\boldsymbol{\Xi}^{\mathbf{v}}_{k+\frac{1}{2},\sigma} = \mathbf{Q}^{\mathbf{v}}_{k+\frac{1}{2},\sigma} + \begin{cases} \dfrac{\partial \mathbf{f}^{\mathbf{v}}\left(\underline{\mathbf{u}}_{\mathbf{V},k,\sigma}\right)}{\partial \mathbf{u}} \left(\boldsymbol{\xi}^{\mathbf{u}}_{k} + \tau \sum\limits_{\substack{i=1\\ i \text{ even}}}^{\sigma-1} a_{\sigma i}\,\boldsymbol{\Xi}^{\mathbf{v}}_{k+\frac{1}{2},i}\right) & \text{if } \sigma \text{ odd} \\ \dfrac{\partial \mathbf{f}^{\mathbf{u}}\left(\underline{\mathbf{v}}_{\mathbf{V},k-\frac{1}{2},\sigma}\right)}{\partial \mathbf{v}} \left(\boldsymbol{\xi}^{\mathbf{v}}_{k-\frac{1}{2}} + \tau \sum\limits_{\substack{i=1\\ i \text{ odd}}}^{\sigma-1} a_{\sigma i}\,\boldsymbol{\Xi}^{\mathbf{v}}_{k+\frac{1}{2},i}\right) & \text{if } \sigma \text{ even} \end{cases}$$

- *Compute*

$$\boldsymbol{\xi}^{\mathbf{v}}_{k+\frac{1}{2}} = \mathbf{q}^{\mathbf{v}}_{k+\frac{1}{2}} + \boldsymbol{\xi}^{\mathbf{v}}_{k-\frac{1}{2}} + \tau \sum_{\substack{\sigma=1\\ \sigma \text{ odd}}}^{s} b_\sigma \boldsymbol{\Xi}^{\mathbf{v}}_{k+\frac{1}{2},\sigma}.$$

The solution procedure for even s is similar, see Section 2.3.2 and Appendix A.3.2 for

guidance.

Example

The linearized forward time-stepping method corresponding to the 5-stage StagRK4 method with parameter b_5 and $\gamma = (6\,b_5)^{-1/2}$ (see Section 2.3.2) is:
For $k = K, \cdots, 1$:

- For $\sigma = 1, \cdots, s$, evaluate the internal stages for $\boldsymbol{\xi}^{\mathbf{u}}$:

$$
\begin{aligned}
\boldsymbol{\Xi}^{\mathbf{u}}_{k,1} &= \mathbf{Q}^{\mathbf{u}}_{k,1} + \frac{\partial \mathbf{f}^{\mathbf{u}}\left(\mathbf{v}_{k-\frac{1}{2}}\right)}{\partial \mathbf{v}} \boldsymbol{\xi}^{\mathbf{v}}_{k-\frac{1}{2}} \\
\boldsymbol{\Xi}^{\mathbf{u}}_{k,2} &= \mathbf{Q}^{\mathbf{u}}_{k,2} + \frac{\partial \mathbf{f}^{\mathbf{v}}\left(\mathbf{u}_{k-1} + \frac{\tau}{4}(2-\gamma)\mathbf{U}_{k,1}\right)}{\partial \mathbf{u}} \left(\boldsymbol{\xi}^{\mathbf{u}}_{k-1} + \frac{\tau}{4}(2-\gamma)\,\boldsymbol{\Xi}^{\mathbf{u}}_{k,1}\right) \\
\boldsymbol{\Xi}^{\mathbf{u}}_{k,3} &= \mathbf{Q}^{\mathbf{u}}_{k,3} + \frac{\partial \mathbf{f}^{\mathbf{u}}\left(\mathbf{v}_{k-\frac{1}{2}} - \frac{\tau}{2}\gamma\mathbf{U}_{k,2}\right)}{\partial \mathbf{v}} \left(\boldsymbol{\xi}^{\mathbf{v}}_{k-\frac{1}{2}} - \frac{\tau}{2}\gamma\,\boldsymbol{\Xi}^{\mathbf{u}}_{k,2}\right) \\
\boldsymbol{\Xi}^{\mathbf{u}}_{k,4} &= \mathbf{Q}^{\mathbf{u}}_{k,4} + \frac{\partial \mathbf{f}^{\mathbf{v}}\left(\mathbf{u}_{k-1} + \frac{\tau}{4}(2+\gamma)\mathbf{U}_{k,1}\right)}{\partial \mathbf{u}} \left(\boldsymbol{\xi}^{\mathbf{u}}_{k-1} + \frac{\tau}{4}(2+\gamma)\,\boldsymbol{\Xi}^{\mathbf{u}}_{k,1}\right) \\
\boldsymbol{\Xi}^{\mathbf{u}}_{k,5} &= \mathbf{Q}^{\mathbf{u}}_{k,5} + \frac{\partial \mathbf{f}^{\mathbf{u}}\left(\mathbf{v}_{k-\frac{1}{2}} + \frac{\tau}{2}\gamma\mathbf{U}_{k,4}\right)}{\partial \mathbf{v}} \left(\boldsymbol{\xi}^{\mathbf{v}}_{k-\frac{1}{2}} + \frac{\tau}{2}\gamma\,\boldsymbol{\Xi}^{\mathbf{u}}_{k,4}\right)
\end{aligned}
$$

- Update $\boldsymbol{\xi}^{\mathbf{u}}$:

$$
\boldsymbol{\xi}^{\mathbf{u}}_{k} = \mathbf{q}^{\mathbf{u}}_{k} + \boldsymbol{\xi}^{\mathbf{u}}_{k-1} + \tau\,(1-2\,b_5)\boldsymbol{\Xi}^{\mathbf{u}}_{k,1} + \tau\,b_5\boldsymbol{\Xi}^{\mathbf{u}}_{k,3} + \tau\,b_5\boldsymbol{\Xi}^{\mathbf{u}}_{k,5}.
$$

- For $\sigma = 1, \cdots, s$, evaluate the internal stages for $\boldsymbol{\xi}^{\mathbf{v}}$:

$$
\begin{aligned}
\boldsymbol{\Xi}^{\mathbf{v}}_{k+\frac{1}{2},1} &= \mathbf{Q}^{\mathbf{v}}_{k+\frac{1}{2},1} + \frac{\partial \mathbf{f}^{\mathbf{v}}\left(\mathbf{u}_k\right)}{\partial \mathbf{u}} \boldsymbol{\xi}^{\mathbf{u}}_{k} \\
\boldsymbol{\Xi}^{\mathbf{v}}_{k+\frac{1}{2},2} &= \mathbf{Q}^{\mathbf{v}}_{k+\frac{1}{2},2} + \frac{\partial \mathbf{f}^{\mathbf{u}}\left(\mathbf{v}_{k-\frac{1}{2}} + \frac{\tau}{4}(2-\gamma)\,\mathbf{V}_{k+\frac{1}{2},1}\right)}{\partial \mathbf{v}} \left(\boldsymbol{\xi}^{\mathbf{v}}_{k-\frac{1}{2}} + \frac{\tau}{4}(2-\gamma)\,\boldsymbol{\Xi}^{\mathbf{v}}_{k+\frac{1}{2},1}\right) \\
\boldsymbol{\Xi}^{\mathbf{v}}_{k+\frac{1}{2},3} &= \mathbf{Q}^{\mathbf{v}}_{k+\frac{1}{2},3} + \frac{\partial \mathbf{f}^{\mathbf{v}}\left(\mathbf{u}_k - \frac{\tau}{2}\gamma\,\mathbf{V}_{k+\frac{1}{2},2}\right)}{\partial \mathbf{u}} \left(\boldsymbol{\xi}^{\mathbf{u}}_{k} - \frac{\tau}{2}\gamma\,\boldsymbol{\Xi}^{\mathbf{v}}_{k+\frac{1}{2},2}\right)
\end{aligned}
$$

$$\Xi^{\mathbf{v}}_{k+\frac{1}{2},4} = \mathbf{Q}^{\mathbf{v}}_{k+\frac{1}{2},4} + \frac{\partial \mathbf{f}^{\mathbf{u}}\left(\mathbf{v}_{k-\frac{1}{2}} + \frac{\tau}{4}(2+\gamma)\,\mathbf{V}_{k+\frac{1}{2},1}\right)}{\partial \mathbf{v}}\left(\boldsymbol{\xi}^{\mathbf{v}}_{k-\frac{1}{2}} + \frac{\tau}{4}(2+\gamma)\,\Xi^{\mathbf{v}}_{k+\frac{1}{2},1}\right)$$
$$\Xi^{\mathbf{v}}_{k+\frac{1}{2},5} = \mathbf{Q}^{\mathbf{v}}_{k+\frac{1}{2},5} + \frac{\partial \mathbf{f}^{\mathbf{v}}\left(\mathbf{u}_k + \frac{\tau}{2}\gamma\,\mathbf{V}_{k+\frac{1}{2},4}\right)}{\partial \mathbf{u}}\left(\boldsymbol{\xi}^{\mathbf{u}}_{k} + \frac{\tau}{2}\gamma\,\Xi^{\mathbf{v}}_{k+\frac{1}{2},4}\right)$$

- Update $\boldsymbol{\xi}^{\mathbf{v}}$:

$$\boldsymbol{\xi}^{\mathbf{v}}_{k+\frac{1}{2}} = \mathbf{q}^{\mathbf{v}}_{k+\frac{1}{2}} + \boldsymbol{\xi}^{\mathbf{v}}_{k-\frac{1}{2}} + \tau(1-2\,b_5)\Xi^{\mathbf{v}}_{k+\frac{1}{2},1} + \tau\,b_5\Xi^{\mathbf{v}}_{k+\frac{1}{2},3} + \tau\,b_5\Xi^{\mathbf{v}}_{k+\frac{1}{2},5}.$$

Letting $b_5 = 1/24$ (and therefore $\gamma = 2$), the internal stages for $\boldsymbol{\xi}^{\mathbf{u}}$ are

$$\Xi^{\mathbf{u}}_{k,1} = \mathbf{Q}^{\mathbf{u}}_{k,1} + \frac{\partial \mathbf{f}^{\mathbf{u}}\left(\mathbf{v}_{k-\frac{1}{2}}\right)}{\partial \mathbf{v}}\boldsymbol{\xi}^{\mathbf{v}}_{k-\frac{1}{2}}$$
$$\Xi^{\mathbf{u}}_{k,2} = \mathbf{Q}^{\mathbf{u}}_{k,2} + \frac{\partial \mathbf{f}^{\mathbf{v}}\left(\mathbf{u}_{k-1}\right)}{\partial \mathbf{u}}\boldsymbol{\xi}^{\mathbf{u}}_{k-1}$$
$$\Xi^{\mathbf{u}}_{k,3} = \mathbf{Q}^{\mathbf{u}}_{k,3} + \frac{\partial \mathbf{f}^{\mathbf{u}}\left(\mathbf{v}_{k-\frac{1}{2}} - \tau\,\mathbf{U}_{k,2}\right)}{\partial \mathbf{v}}\left(\boldsymbol{\xi}^{\mathbf{v}}_{k-\frac{1}{2}} - \tau\,\Xi^{\mathbf{u}}_{k,2}\right)$$
$$\Xi^{\mathbf{u}}_{k,4} = \mathbf{Q}^{\mathbf{u}}_{k,4} + \frac{\partial \mathbf{f}^{\mathbf{v}}\left(\mathbf{u}_{k-1} + \tau\mathbf{U}_{k,1}\right)}{\partial \mathbf{u}}\left(\boldsymbol{\xi}^{\mathbf{u}}_{k-1} + \tau\,\Xi^{\mathbf{u}}_{k,1}\right)$$
$$\Xi^{\mathbf{u}}_{k,5} = \mathbf{Q}^{\mathbf{u}}_{k,5} + \frac{\partial \mathbf{f}^{\mathbf{u}}\left(\mathbf{v}_{k-\frac{1}{2}} + \tau\,\mathbf{U}_{k,4}\right)}{\partial \mathbf{v}}\left(\boldsymbol{\xi}^{\mathbf{v}}_{k-\frac{1}{2}} + \tau\,\Xi^{\mathbf{u}}_{k,4}\right)$$

and the internal stages for $\boldsymbol{\xi}^{\mathbf{v}}$ are

$$\Xi^{\mathbf{v}}_{k+\frac{1}{2},1} = \mathbf{Q}^{\mathbf{v}}_{k+\frac{1}{2},1} + \frac{\partial \mathbf{f}^{\mathbf{v}}\left(\mathbf{u}_{k}\right)}{\partial \mathbf{u}}\boldsymbol{\xi}^{\mathbf{u}}_{k}$$
$$\Xi^{\mathbf{v}}_{k+\frac{1}{2},2} = \mathbf{Q}^{\mathbf{v}}_{k+\frac{1}{2},2} + \frac{\partial \mathbf{f}^{\mathbf{u}}\left(\mathbf{v}_{k-\frac{1}{2}}\right)}{\partial \mathbf{v}}\boldsymbol{\xi}^{\mathbf{v}}_{k-\frac{1}{2}}$$
$$\Xi^{\mathbf{v}}_{k+\frac{1}{2},3} = \mathbf{Q}^{\mathbf{v}}_{k+\frac{1}{2},3} + \frac{\partial \mathbf{f}^{\mathbf{v}}\left(\mathbf{u}_{k} - \tau\,\mathbf{V}_{k+\frac{1}{2},2}\right)}{\partial \mathbf{u}}\left(\boldsymbol{\xi}^{\mathbf{u}}_{k} - \tau\,\Xi^{\mathbf{v}}_{k+\frac{1}{2},2}\right)$$
$$\Xi^{\mathbf{v}}_{k+\frac{1}{2},4} = \mathbf{Q}^{\mathbf{v}}_{k+\frac{1}{2},4} + \frac{\partial \mathbf{f}^{\mathbf{u}}\left(\mathbf{v}_{k-\frac{1}{2}} + \tau\,\mathbf{V}_{k+\frac{1}{2},1}\right)}{\partial \mathbf{v}}\left(\boldsymbol{\xi}^{\mathbf{v}}_{k-\frac{1}{2}} + \tau\,\Xi^{\mathbf{v}}_{k+\frac{1}{2},1}\right)$$
$$\Xi^{\mathbf{v}}_{k+\frac{1}{2},5} = \mathbf{Q}^{\mathbf{v}}_{k+\frac{1}{2},5} + \frac{\partial \mathbf{f}^{\mathbf{v}}\left(\mathbf{u}_{k} + \tau\,\mathbf{V}_{k+\frac{1}{2},4}\right)}{\partial \mathbf{u}}\left(\boldsymbol{\xi}^{\mathbf{u}}_{k} + \tau\,\Xi^{\mathbf{v}}_{k+\frac{1}{2},4}\right).$$

The update steps are obvious.

5.4 Exponential Time-Differencing Methods

The linearized ETDRK method is somewhat more difficult to obtain if the linear operator $\mathbf{L}_k$ happens to depend on $\mathbf{y}_k$, as is the case with exponential Rosenbrock methods. Therefore we illustrate the linearized ETDRK method using an example.

5.4.1 Exponential Runge-Kutta Method

The linear system to be solved is

$$\begin{bmatrix} \mathbf{A}_1 & & & & & & \\ -\mathbf{B}_1^\top & \mathbf{I}_N & & & & & \\ & \mathbf{C}_2 & \mathbf{A}_2 & & & & \\ & \mathbf{D}_2 & -\mathbf{B}_2^\top & \mathbf{I}_N & & & \\ & & \ddots & \ddots & \ddots & & \\ & & & & \mathbf{C}_{k-1} & \mathbf{A}_{k-1} & \\ & & & & \mathbf{D}_{k-1} & -\mathbf{B}_{k-1}^\top & \mathbf{I}_N \end{bmatrix} \begin{bmatrix} \boldsymbol{\Xi}_1 \\ \boldsymbol{\xi}_1 \\ \boldsymbol{\Xi}_2 \\ \vdots \\ \boldsymbol{\xi}_{K-1} \\ \boldsymbol{\Xi}_{k-1} \\ \boldsymbol{\xi}_{k-1} \end{bmatrix} = \begin{bmatrix} \mathbf{Q}_1 \\ \mathbf{q}_1 \\ \mathbf{Q}_2 \\ \vdots \\ \mathbf{q}_{K-1} \\ \mathbf{Q}_{k-1} \\ \mathbf{q}_{k-1} \end{bmatrix}, \tag{5.6}$$

with $\mathbf{A}_k$, $\mathbf{C}_k$ and $\mathbf{D}_k$ defined in (A.229) and $\mathbf{B}_k^\top = \tau_k \begin{bmatrix} b_1(\tau_k \mathbf{L}_k) & \cdots & b_s(\tau_k \mathbf{L}_k) \end{bmatrix}$. Since the system is block-lower triangular we use forward substitution to get

$$\boldsymbol{\xi}_k = \mathbf{q}_k + \mathbf{B}_k^\top \boldsymbol{\Xi}_k - \mathbf{D}_k \boldsymbol{\xi}_k$$

for $k = 1, \ldots, K$ and internal stages

$$\mathbf{A}_k \boldsymbol{\Xi}_k = \mathbf{Q}_k - \mathbf{C}_k \boldsymbol{\xi}_k,$$

with $\boldsymbol{\xi}_0 = \mathbf{0}_{N \times 1}$.

Substituting (A.229) then gives

$$\boldsymbol{\xi}_k = \mathbf{q}_k + \mathbf{B}_k^\top \boldsymbol{\Xi}_k + \frac{\partial \left(e^{\tau_k \mathbf{L}_k} \mathbf{y}_k\right)}{\partial \mathbf{y}_k} \boldsymbol{\xi}_k + \frac{\partial \left(\mathbf{B}_k^\top \mathbf{Y}_k\right)}{\partial \mathbf{y}_k} \boldsymbol{\xi}_k$$

for $k = 1, \cdots, K$, with the internal stages

$$\boldsymbol{\Xi}_k = \mathbf{Q}_k + \frac{\partial \mathbf{N}_k}{\partial \mathbf{y}_k} \boldsymbol{\xi}_k + \tau_k \frac{\partial \mathbf{N}_k}{\partial \mathbf{y}} \mathbf{A}_k^s \boldsymbol{\Xi}_k.$$

The detailed solution procedure is summarized in Algorithm 5.7.

Algorithm 5.7. The Linearized Exponential Runge-Kutta Method

Ignore the terms in braces if $\mathbf{L}_k$ *does not depend on* $\mathbf{y}_k$.

Set $\boldsymbol{\xi}_0 = \mathbf{0}_{N\times 1}$. *For* $k = 1, \cdots, K$:

- *For* $\sigma = 1, \cdots, s$, *compute the internal stages*

$$\begin{aligned}\boldsymbol{\Xi}_{k,\sigma} = \mathbf{Q}_{k,i} + \frac{\partial \mathbf{n}_{k,i}}{\partial \mathbf{y}} \left(e^{c_\sigma \tau_k \mathbf{L}_{k-1}} \boldsymbol{\xi}_{k-1} + \tau_k \sum_{j=1}^{i-1} a_{\sigma i}(\tau_k \mathbf{L}_{k-1}) \boldsymbol{\Xi}_{k,j} \right) + \\ + \left\{ \frac{\partial \mathbf{n}_{k,i}}{\partial \mathbf{y}_{k-1}} \boldsymbol{\xi}_{k-1} + \frac{\partial \mathbf{n}_{k,i}}{\partial \mathbf{y}} \left(\frac{\partial \left(e^{c_\sigma \tau_k \mathbf{L}_{k-1}(\mathbf{y}_{k-1})} \mathbf{y}_{k-1}^{\text{fixed}}\right)}{\partial \mathbf{y}_{k-1}} \boldsymbol{\xi}_{k-1} + \right.\right. \\ \left.\left. + \tau_k \sum_{j=1}^{i-1} \frac{\partial (a_{\sigma i}(\tau_k \mathbf{L}_{k-1}(\mathbf{y}_{k-1})) \mathbf{Y}_{k,j})}{\partial \mathbf{y}_{k-1}} \boldsymbol{\xi}_{k-1} \right) \right\}. \end{aligned} \tag{5.7a}$$

- *Then compute the update:*

$$\begin{aligned}\boldsymbol{\xi}_k = \mathbf{q}_k + e^{\tau_k \mathbf{L}_{k-1}}\boldsymbol{\xi}_{k-1} + \tau_k \sum_{\sigma=1}^{s} b_\sigma(\tau_k \mathbf{L}_{k-1})\boldsymbol{\Xi}_{k,i} + \\ + \left\{ \frac{\partial\left(e^{\tau_k \mathbf{L}_{k-1}(\mathbf{y}_{k-1})}\mathbf{y}_{k-1}^{\text{fixed}}\right)}{\partial \mathbf{y}_{k-1}}\boldsymbol{\xi}_{k-1} + \right. \\ \left. + \tau_k \sum_{\sigma=1}^{s} \frac{\partial(b_\sigma(\tau_k \mathbf{L}_{k-1}(\mathbf{y}_{k-1}))\mathbf{Y}_{k,i})}{\partial \mathbf{y}_{k-1}}\boldsymbol{\xi}_{k-1} \right\}.\end{aligned} \tag{5.7b}$$

The derivatives in the braces in (5.7) can be evaluated by writing $a_{\sigma i}$ and b_σ in terms of φ-functions and then differentiating the φ-functions as in Section 4.4.2. Without the terms in braces we simply have the standard ETDRK method where the nonlinear term has been linearized.

5.4.2 Application to Krogstad's Scheme

We show how the above algorithm would look when applied to the scheme proposed in [72]. Recall that the matrix coefficients $a_{\sigma i}$ and b_σ are given in (2.63) in terms of the φ-functions. The linearized Krogstad scheme is presented in Algorithm 5.8.

Algorithm 5.8. The Linearized Krogstad Scheme

Set $\boldsymbol{\xi}_0 = \mathbf{0}_{N\times 1}$. *Let* $\varphi_\ell = \varphi_\ell(\tau_k \mathbf{L}_{k-1}(\mathbf{y}_{k-1}))$ *and* $\varphi_{\ell,i} = \varphi_{\ell,i}(\tau_k \mathbf{L}_{k-1}(\mathbf{y}_{k-1}))$, *and ignore the terms in braces if* $\mathbf{L}_{k-1}$ *does not depend on* $\mathbf{y}_k$. *For* $k = 1, \cdots, K$:

- *Compute the internal stages*

$$
\begin{aligned}
\boldsymbol{\Xi}_{k,1} &= \mathbf{Q}_{k,1} + \frac{\partial \mathbf{n}_{k,1}}{\partial \mathbf{y}}\boldsymbol{\xi}_{k-1} + \left\{\frac{\partial \mathbf{n}_{k,1}}{\partial \mathbf{y}_{k-1}}\boldsymbol{\xi}_{k-1}\right\} \\
\boldsymbol{\Xi}_{k,2} &= \mathbf{Q}_{k,2} + \frac{\partial \mathbf{n}_{k,2}}{\partial \mathbf{y}}\left(\varphi_{0,2}\boldsymbol{\xi}_{k-1} + \frac{\tau_k}{2}\varphi_{1,2}\boldsymbol{\Xi}_{k,1}\right) + \\
&\quad + \left\{\frac{\partial \mathbf{n}_{k,2}}{\partial \mathbf{y}}\left(\frac{\partial\left(\varphi_{0,2}\,\mathbf{y}_{k-1}^{\text{fixed}}\right)}{\partial \mathbf{y}_{k-1}} + \frac{\tau_k}{2}\frac{\partial(\varphi_{1,2}\,\mathbf{Y}_{k,1})}{\partial \mathbf{y}_{k-1}}\right)\boldsymbol{\xi}_{k-1} + \frac{\partial \mathbf{n}_{k,2}}{\partial \mathbf{y}_{k-1}}\boldsymbol{\xi}_{k-1}\right\} \\
\\
\boldsymbol{\Xi}_{k,3} &= \mathbf{Q}_{k,3} + \frac{\partial \mathbf{n}_{k,3}}{\partial \mathbf{y}}\left(\varphi_{0,3}\boldsymbol{\xi}_{k-1} + \frac{\tau_k}{2}\varphi_{1,3}\boldsymbol{\Xi}_{k,1} + \tau_k\varphi_{2,3}\left(\boldsymbol{\Xi}_{k,2} - \boldsymbol{\Xi}_{k,1}\right)\right) + \\
&\quad + \left\{\frac{\partial \mathbf{n}_{k,3}}{\partial \mathbf{y}_{k-1}}\boldsymbol{\xi}_{k-1} + \frac{\partial \mathbf{n}_{k,3}}{\partial \mathbf{y}}\left(\frac{\partial(\varphi_{0,3}\mathbf{y}_{k-1}^{\text{fixed}})}{\partial \mathbf{y}_{k-1}} + \frac{\tau_k}{2}\frac{\partial(\varphi_{1,3}\mathbf{Y}_{k,1})}{\partial \mathbf{y}_{k-1}}\right)\boldsymbol{\xi}_{k-1}\right\} \\
&\quad + \left\{\tau_k\frac{\partial \mathbf{n}_{k,3}}{\partial \mathbf{y}}\left(\frac{\partial(\varphi_{2,3}(\mathbf{Y}_{k,2} - \mathbf{Y}_{k,1}))}{\partial \mathbf{y}_{k-1}}\right)\boldsymbol{\xi}_{k-1}\right\} \\
\boldsymbol{\Xi}_{k,4} &= \mathbf{Q}_{k,4} + \frac{\partial \mathbf{n}_{k,4}}{\partial \mathbf{y}}\left(\varphi_0\boldsymbol{\xi}_{k-1} + \tau_k\varphi_{1,4}\boldsymbol{\Xi}_{k,1} + 2\tau_k\varphi_{2,4}\left(\boldsymbol{\Xi}_{k,3} - \boldsymbol{\Xi}_{k,1}\right)\right) + \\
&\quad + \left\{\frac{\partial \mathbf{n}_{k,4}}{\partial \mathbf{y}_{k-1}}\boldsymbol{\xi}_{k-1} + \frac{\partial \mathbf{n}_{k,4}}{\partial \mathbf{y}}\left(\frac{\partial\left(\varphi_0\mathbf{y}_{k-1}^{\text{fixed}}\right)}{\partial \mathbf{y}_{k-1}} + \tau_k\frac{\partial(\varphi_{1,4}\mathbf{Y}_{k,1})}{\partial \mathbf{y}_{k-1}}\right)\boldsymbol{\xi}_{k-1}\right\} + \\
&\quad + \left\{2\tau_k\frac{\partial \mathbf{n}_{k,4}}{\partial \mathbf{y}}\left(\frac{\partial(\varphi_{2,4}(\mathbf{Y}_{k,3} - \mathbf{Y}_{k,1}))}{\partial \mathbf{y}_{k-1}}\right)\boldsymbol{\xi}_{k-1}\right\}
\end{aligned}
$$

- *Compute the update:*

$$
\begin{aligned}
\boldsymbol{\xi}_k = \mathbf{q}_k &+ \varphi_0\boldsymbol{\xi}_{k-1} + \tau_k\varphi_2(-3\boldsymbol{\Xi}_{k,1} + 2\boldsymbol{\Xi}_{k,2} + 2\boldsymbol{\Xi}_{k,3} - \boldsymbol{\Xi}_{k,4}) + \\
&+ \tau_k\varphi_1\boldsymbol{\Xi}_{k,1} + 4\tau_k\varphi_3(\boldsymbol{\Xi}_{k,1} - \boldsymbol{\Xi}_{k,2} - \boldsymbol{\Xi}_{k,3} + \boldsymbol{\Xi}_{k,4}) + \\
&+ \left\{\frac{\partial\left(\varphi_0\mathbf{y}_{k-1}^{\text{fixed}}\right)}{\partial \mathbf{y}_{k-1}}\boldsymbol{\xi}_{k-1} + \tau_k\frac{\partial(\varphi_1\mathbf{Y}_{k,1})}{\partial \mathbf{y}_{k-1}}\boldsymbol{\xi}_{k-1} + \right. \\
&\quad + \tau_k\frac{\partial(\varphi_2(-3\mathbf{Y}_{k,1} + 2\mathbf{Y}_{k,2} + 2\mathbf{Y}_{k,3} - \mathbf{Y}_{k,4}))}{\partial \mathbf{y}_{k-1}}\boldsymbol{\xi}_{k-1} + \\
&\quad \left. + 4\tau_k\frac{\partial(\varphi_3(\mathbf{Y}_{k,1} - \mathbf{Y}_{k,2} - \mathbf{Y}_{k,3} + \mathbf{Y}_{k,4}))}{\partial \mathbf{y}_{k-1}}\boldsymbol{\xi}_{k-1}\right\}.
\end{aligned}
$$

The way the scheme is written here may of course not represent the most computationally efficient implementation. For instance, there are some opportunities for parallelization and one should also exploit the fact that $c_2 = c_3 = 1/2$.

Chapter 6

The Adjoint Problem

The major ingredient needed when calculating the action of the transpose of the sensitivity matrix is the solution of the adjoint problem,

$$\boldsymbol{\lambda} = \left(\frac{\partial \mathbf{t}}{\partial \mathbf{y}}\right)^{-\top} \boldsymbol{\theta}, \tag{6.1}$$

where $\boldsymbol{\lambda}$ is the adjoint solution and $\boldsymbol{\theta}$ is the adjoint source. In this chapter the adjoint source is taken to be some arbitrary adjoint source, but for the gradient computation we will have $\boldsymbol{\theta} = \frac{\partial \mathbf{d}}{\partial \mathbf{y}}^{\top} \nabla_{\mathbf{d}} \mathrm{M}$. The adjoint solution is a crucial component in the computation of the gradient and the action of the Hessian.

The following remarks are in order:

- The structure of the linearization of the time-stepping vector is lower block triangular (or essentially lower block triangular in the case of staggered methods), so the adjoint problem will require the solution of an upper block triangular system. This is done using backsubstitution, which corresponds to solving the adjoint problem backward in time, starting from a final condition.

- The structure of (6.1) dictates that the final value of the adjoint solution must be $\boldsymbol{\lambda}_{K+1} = \mathbf{0}_{N\times 1}$. The value of $\boldsymbol{\lambda}_K$ is that of the adjoint source at the final time-step, so in the literature this is generally considered to be the final condition.

- As was the case when computing the linearized forward solution, the linearization of the nonlinear operators about the current values of the forward solution means that we must have the forward solution available, either in storage or computed on the fly using checkpointing.

- We do not have a source weighting operator in (6.1), so for LM-type methods we do not need to combine source values from different time levels.

- We do not assume any structure on $\mathbf{q}$, so it is possible that RK-type methods will have a source term appearing in the update step as well as in the internal stages.

We will now show the details of the adjoint time-stepping procedure to solve (6.1) for each of the time-stepping methods under consideration. For each of the time-stepping methods we will also give examples of particular schemes, based on the examples given in Chapter 2.

6.1 Regular Time-Stepping Methods

In this section we present the adjoint LM and RK methods, and give examples of particular adjoint schemes.

As discussed in the introduction to the thesis, the results presented in this section are not new. Adjoint LM methods were discussed in [101, 102] and in fact variable time-steps were considered in those papers; we only consider constant time-steps due to related to simplicity and consistency of the resulting adjoint methods. Adjoint RK methods are discussed in several references, for instance [3, 52, 100, 131] and several others.

Nonetheless, the adjoint LM and RK methods are included here for several reasons:

- The approach used to derive the adjoint methods is different from the one used in the references above. The use of an abstract representation of a time-stepping method using a time-stepping vector $\mathbf{t}$ allows for a general approach to finding adjoint time-stepping methods; a large number of different time-stepping methods can be tackled in this way.

- For the sake of completeness and readability it is useful to have the adjoint LM and RK methods in the same chapter as the other adjoint time-stepping methods.

- Connected to the above point, having the adjoint LM and RK methods available will help the reader better understand the derivation of the more complicated adjoint IMEX, staggered and exponential time-stepping methods.

6.1.1 Linear Multistep Methods

Let $\overline{\boldsymbol{\lambda}} = \begin{bmatrix} \boldsymbol{\lambda}_1^\top & \cdots & \boldsymbol{\lambda}_K^\top \end{bmatrix}^\top$ represent the adjoint solution and $\overline{\boldsymbol{\theta}} = \begin{bmatrix} \boldsymbol{\theta}_1^\top & \cdots & \boldsymbol{\theta}_K^\top \end{bmatrix}^\top$ the adjoint source.

The transpose of the $K \times K$ block matrix $\dfrac{\partial \mathbf{t}}{\partial \overline{\mathbf{y}}}$ is an upper block triangular matrix that has the form of the transposes of the template matrices (2.1) and (2.3). Its (k, j)th $N \times N$ block is (see (A.3))

$$\frac{\partial \mathbf{t}_j}{\partial \mathbf{y}_k}^\top = \begin{cases} \alpha_{j-k}^{(\sigma)} \mathbf{I}_N - \tau \beta_{j-k}^{(\sigma)} \dfrac{\partial \mathbf{f}_k}{\partial \mathbf{y}}^\top & \text{if } k \leq j \leq k^\star \\ \mathbf{0}_{N \times N} & \text{otherwise,} \end{cases} \tag{6.2}$$

where $\sigma = \min(j, s)$ and $k^\star = \min(K, k + s)$.

To solve (6.1) we must solve, for $1 \leq k \leq K$,

$$\frac{\partial \mathbf{t}_k}{\partial \mathbf{y}_k}^\top \boldsymbol{\lambda}_k = \boldsymbol{\theta}_k - \sum_{j=k+1}^{k^\star} \frac{\partial \mathbf{t}_j}{\partial \mathbf{y}_k}^\top \boldsymbol{\lambda}_j.$$

The solution procedure is summarized in Algorithm 6.1.

Algorithm 6.1. The Adjoint Linear Multistep Method

Let $\sigma = \min(k, s)$ *and* $k^\star = \max(1, k - s)$.

For $k = K, \cdots, 1$, *solve for* $\boldsymbol{\lambda}_k$:

$$\left(\mathbf{I}_N - \tau \beta_0^{(\sigma)} \frac{\partial \mathbf{f}_k}{\partial \mathbf{y}}^\top \right) \boldsymbol{\lambda}_k = \boldsymbol{\theta}_k - \sum_{j=k+1}^{k^\star} \alpha_{j-k}^{(\sigma)} \boldsymbol{\lambda}_j + \tau \frac{\partial \mathbf{f}_k}{\partial \mathbf{y}}^\top \left(\sum_{j=k+1}^{k^\star} \beta_{j-k}^{(\sigma)} \boldsymbol{\lambda}_j \right)$$

with $\dfrac{\partial \mathbf{f}_k}{\partial \mathbf{y}} = \dfrac{\partial \mathbf{f}\left(\mathbf{y}_k\right)}{\partial \mathbf{y}} = \dfrac{\partial \mathbf{f}\left(\mathbf{y}_k, t_k\right)}{\partial \mathbf{y}}.$

For explicit methods the left-hand side is simply $\boldsymbol{\lambda}_k$, whereas for implicit methods one needs to solve a $N \times N$ linear system at each time-step. Note that the right-hand side requires just one matrix-vector product, in contrast to the linearized forward linear multistep method, where s matrix-vector products are needed on the right-hand side at each time level.

Examples

We find the adjoint time-stepping methods corresponding to the linear multistep methods mentioned in Section 2.1.1. In all cases we have $\boldsymbol{\lambda}_k = \mathbf{0}_{N\times 1}$ for $k > K$.

The adjoint s-step Adams-Bashforth methods, with $s = 1, 2, 3$ and for $k = K, \cdots, \max 1, s-1$, are

$$
\begin{aligned}
s = 1: &\quad \boldsymbol{\lambda}_k = \boldsymbol{\theta}_k + \boldsymbol{\lambda}_{k+1} + \tau \frac{\partial \mathbf{f}\left(\mathbf{y}_k\right)}{\partial \mathbf{y}}^\top \boldsymbol{\lambda}_{k+1} \\
s = 2: &\quad \boldsymbol{\lambda}_k = \boldsymbol{\theta}_k + \boldsymbol{\lambda}_{k+1} + \frac{\tau}{2} \frac{\partial \mathbf{f}\left(\mathbf{y}_k\right)}{\partial \mathbf{y}}^\top \left(3\,\boldsymbol{\lambda}_{k+1} - \boldsymbol{\lambda}_{k+2}\right) \\
s = 3: &\quad \boldsymbol{\lambda}_k = \boldsymbol{\theta}_k + \boldsymbol{\lambda}_{k+1} + \frac{\tau}{12} \frac{\partial \mathbf{f}\left(\mathbf{y}_k\right)}{\partial \mathbf{y}}^\top \left(23\,\boldsymbol{\lambda}_{k+1} - 16\,\boldsymbol{\lambda}_{k+2} + 5\,\boldsymbol{\lambda}_{k+3}\right).
\end{aligned}
$$

For the adjoint s-step BDFs, with $s = 1, 2, 3$, we solve, for $k = K, \cdots, \max 1, s-1$,

$$
\begin{aligned}
s = 1: &\quad \left(\mathbf{I}_N - \tau \frac{\partial \mathbf{f}\left(\mathbf{y}_k\right)}{\partial \mathbf{y}}\right) \boldsymbol{\lambda}_k = \boldsymbol{\theta}_k + \boldsymbol{\lambda}_{k+1} \\
s = 2: &\quad \left(\mathbf{I}_N - \tau \frac{2}{3} \frac{\partial \mathbf{f}\left(\mathbf{y}_k\right)}{\partial \mathbf{y}}\right) \boldsymbol{\lambda}_k = \boldsymbol{\theta}_k + \frac{4}{3}\boldsymbol{\lambda}_{k+1} - \frac{1}{3}\boldsymbol{\lambda}_{k+2} \\
s = 3: &\quad \left(\mathbf{I}_N - \tau \frac{6}{11} \frac{\partial \mathbf{f}\left(\mathbf{y}_k\right)}{\partial \mathbf{y}}\right) \boldsymbol{\lambda}_k = \boldsymbol{\theta}_k + \frac{18}{11}\boldsymbol{\lambda}_{k+1} - \frac{9}{11}\boldsymbol{\lambda}_{k+2} + \frac{2}{11}\boldsymbol{\lambda}_{k+3}.
\end{aligned}
$$

Note that the BDFs are essentially the same as the linearized forward BDFs except that each $\boldsymbol{\xi}_{k-i}$ is replaced by $\boldsymbol{\lambda}_{k+i}$, i.e. we compute the linearized forward solution, just backward in time.

6.1.2 Runge-Kutta Methods

We let $\overline{\boldsymbol{\lambda}} = \begin{bmatrix} \widehat{\boldsymbol{\lambda}}_1^\top & \cdots & \widehat{\boldsymbol{\lambda}}_K^\top \end{bmatrix}^\top$ represent the adjoint solution, with $\widehat{\boldsymbol{\lambda}}_k = \begin{bmatrix} \boldsymbol{\Lambda}_k^\top & \boldsymbol{\lambda}_k^\top \end{bmatrix}^\top$, where $\boldsymbol{\Lambda}_k^\top = \begin{bmatrix} \boldsymbol{\Lambda}_{k,1}^\top & \cdots & \boldsymbol{\Lambda}_{k,s}^\top \end{bmatrix}^\top$ represents the internal stages of the adjoint solution at the kth time step. The adjoint source $\overline{\boldsymbol{\theta}}$ is defined similarly, with internal stages $\boldsymbol{\Theta}_{k,\sigma}$.

The linearization of the time-stepping equation, $\frac{\partial \mathbf{t}}{\partial \overline{\mathbf{y}}}$, was derived in Section A.1.2. Its transpose is (see (A.51))

$$\frac{\partial \mathbf{t}}{\partial \overline{\mathbf{y}}}^\top = \begin{bmatrix} \mathbf{A}_1^\top & \mathbf{B} & & & & & \\ & \mathbf{I}_N & \mathbf{C}_2^\top & -\mathbf{I}_N & & & \\ & & \mathbf{A}_2^\top & \mathbf{B} & & & \\ & & & \ddots & \ddots & \ddots & \\ & & & & \mathbf{I}_N & \mathbf{C}_K^\top & -\mathbf{I}_N \\ & & & & & \mathbf{A}_K^\top & \mathbf{B} \\ & & & & & & \mathbf{I}_N \end{bmatrix}.$$

where (see (A.47))

$$\mathbf{A}_k = \mathbf{I}_{sN} - \tau_k \frac{\partial \mathbf{F}_k}{\partial \mathbf{y}} \left(\mathbf{A}_s \otimes \mathbf{I}_N\right) \qquad \text{and} \qquad \mathbf{C}_k = -\frac{\partial \mathbf{F}_k}{\partial \mathbf{y}} \left(\mathbf{1}_s \otimes \mathbf{I}_N\right)$$

and $\mathbf{B} = -\tau_k \mathbf{b}_s \otimes \mathbf{I}_N$. $\dfrac{\partial \mathbf{F}_k}{\partial \mathbf{y}}$ was defined to be

$$\frac{\partial \mathbf{F}_k}{\partial \mathbf{y}} = \text{blkdiag}\left(\frac{\partial \mathbf{F}_{k,1}}{\partial \mathbf{y}}, \cdots, \frac{\partial \mathbf{F}_{k,s}}{\partial \mathbf{y}}\right)$$

with $\mathbf{F}_{k,\sigma} = \mathbf{f}\left(\underline{\mathbf{y}}_{k,\sigma}, t_{k-1} + c_\sigma \tau_k\right)$ and $\underline{\mathbf{y}}_{k,\sigma} = \mathbf{y}_{k-1} + \tau_k \sum_{i=1}^{s} a_{\sigma,i} \mathbf{Y}_{k,i}$.

Since $\dfrac{\partial \mathbf{t}}{\partial \overline{\mathbf{y}}}^\top$ is upper (block) triangular, we solve the adjoint system (6.1) by back-

substitution:

$$
\begin{aligned}
\boldsymbol{\lambda}_k &= \boldsymbol{\theta}_k + \boldsymbol{\lambda}_{k+1} - \mathbf{C}_{k+1}^{\top}\boldsymbol{\Lambda}_{k+1} \\
&= \boldsymbol{\theta}_k + \boldsymbol{\lambda}_{k+1} + \left(\mathbf{1}_s^{\top} \otimes \mathbf{I}_N\right) \frac{\partial \mathbf{F}_{k+1}}{\partial \mathbf{y}}^{\top} \boldsymbol{\Lambda}_{k+1} \\
\boldsymbol{\Lambda}_k &= \mathbf{A}_k^{-\top} \left(\boldsymbol{\Theta}_k - \mathbf{B}\boldsymbol{\lambda}_k\right) \\
&= \left(\mathbf{I}_{sN} - \tau_k \frac{\partial \mathbf{F}_k}{\partial \mathbf{y}} \left(\mathbf{A}_s \otimes \mathbf{I}_N\right)\right)^{-\top} \left(\boldsymbol{\Theta}_k - \mathbf{B}\boldsymbol{\lambda}_k\right).
\end{aligned}
\tag{6.3}
$$

The detailed solution procedure is summarized in Algorithm 6.2.

Algorithm 6.2. The Adjoint Runge-Kutta Method

Let $\boldsymbol{\lambda}_{K+1} = \mathbf{0}_{N\times 1}$ *and* $\boldsymbol{\Lambda}_{K+1} = \mathbf{0}_{sN\times 1}$*, and recall that we defined*

$$\underline{\mathbf{y}}_{k,\sigma} = \mathbf{y}_{k-1} + \tau_k \sum_{i=1}^{\sigma} a_{\sigma i} \mathbf{Y}_{k,i}.$$

Denote $\widehat{\boldsymbol{\Lambda}}_{k,\sigma} = \dfrac{\partial \mathbf{f}\left(\underline{\mathbf{y}}_{k,\sigma}\right)}{\partial \mathbf{y}}^{\top} \boldsymbol{\Lambda}_{k,\sigma}$*. For* $k = K, \cdots, 1$*:*

- *Update* $\boldsymbol{\lambda}$*:*

$$\boldsymbol{\lambda}_k = \boldsymbol{\theta}_k + \boldsymbol{\lambda}_{k+1} + \sum_{\sigma=1}^{s} \widehat{\boldsymbol{\Lambda}}_{k+1,\sigma}$$

- *Compute, or solve for, the (modified) internal stages for* $\boldsymbol{\lambda}_{k-1}$*. For* $\sigma = s, \cdots, 1$*:*

$$
\begin{aligned}
\boldsymbol{\Lambda}_{k,\sigma} &= \boldsymbol{\Theta}_{k,\sigma} + \tau_k b_\sigma \boldsymbol{\lambda}_k + \tau_k \sum_{i=1}^{s} a_{i\sigma} \widehat{\boldsymbol{\Lambda}}_{k,i} \\
\widehat{\boldsymbol{\Lambda}}_{k,\sigma} &= \frac{\partial \mathbf{f}\left(\underline{\mathbf{y}}_{k,\sigma}\right)}{\partial \mathbf{y}}^{\top} \boldsymbol{\Lambda}_{k,\sigma}.
\end{aligned}
$$

The adjoint method is explicit, diagonally implicit or fully implicit in accordance with the corresponding forward method. For explicit methods the summation ranges from $\sigma + 1$ to s.

Example

The adjoint RK4 method is, for $k = K, \cdots, 1$ and with $\boldsymbol{\lambda}_{K+1} = \mathbf{0}_{N\times 1}$ and $\boldsymbol{\Lambda}_{K+1} = \mathbf{0}_{sN\times 1}$:

- Update $\boldsymbol{\lambda}$:

$$\boldsymbol{\lambda}_k = \boldsymbol{\theta}_k + \boldsymbol{\lambda}_{k+1} + \widehat{\boldsymbol{\Lambda}}_{k+1,1} + \widehat{\boldsymbol{\Lambda}}_{k+1,2} + \widehat{\boldsymbol{\Lambda}}_{k+1,3} + \widehat{\boldsymbol{\Lambda}}_{k+1,4};$$

- Compute the (modified) internal stages for $\boldsymbol{\lambda}_{k-1}$. For $\sigma = s, \cdots, 1$:

$$\begin{aligned}
\widehat{\boldsymbol{\Lambda}}_{k,4} &= \frac{\partial \mathbf{f}\left(\mathbf{y}_{k-1} + \tau_k \mathbf{Y}_{k,3}\right)}{\partial \mathbf{y}}^{\top} \left(\boldsymbol{\Theta}_{k,4} + \frac{\tau_k}{6}\boldsymbol{\lambda}_k\right) \\
\widehat{\boldsymbol{\Lambda}}_{k,3} &= \frac{\partial \mathbf{f}\left(\mathbf{y}_{k-1} + \frac{\tau_k}{2} \mathbf{Y}_{k,2}\right)}{\partial \mathbf{y}}^{\top} \left(\boldsymbol{\Theta}_{k,3} + \frac{\tau_k}{3}\boldsymbol{\lambda}_k + \tau_k \widehat{\boldsymbol{\Lambda}}_{k,4}\right) \\
\widehat{\boldsymbol{\Lambda}}_{k,2} &= \frac{\partial \mathbf{f}\left(\mathbf{y}_{k-1} + \frac{\tau_k}{2} \mathbf{Y}_{k,1}\right)}{\partial \mathbf{y}}^{\top} \left(\boldsymbol{\Theta}_{k,2} + \frac{\tau_k}{3}\boldsymbol{\lambda}_k + \frac{\tau_k}{2} \widehat{\boldsymbol{\Lambda}}_{k,3}\right) \\
\widehat{\boldsymbol{\Lambda}}_{k,1} &= \frac{\partial \mathbf{f}\left(\mathbf{y}_{k-1}\right)}{\partial \mathbf{y}}^{\top} \left(\boldsymbol{\Theta}_{k,1} + \frac{\tau_k}{6}\boldsymbol{\lambda}_k + \frac{\tau_k}{2} \widehat{\boldsymbol{\Lambda}}_{k,2}\right)
\end{aligned}$$

Note that we do not actually require the 4th internal stage $\mathbf{Y}_{k,4}$ of $\mathbf{y}_k$ in the computation due to the fact that the RK4 method is explicit.

The adjoint time-stepping method corresponding to the fourth-order 2-stage implicit RK method mentioned in Section 2.1.2 is, for $k = K, \cdots, 1$ and with $\boldsymbol{\lambda}_{K+1} = \mathbf{0}_{N\times 1}$ and $\boldsymbol{\Lambda}_{K+1} = \mathbf{0}_{sN\times 1}$:

- Update $\boldsymbol{\lambda}$:

$$\boldsymbol{\lambda}_k = \boldsymbol{\theta}_k + \boldsymbol{\lambda}_{k+1} + \widehat{\boldsymbol{\Lambda}}_{k+1,1} + \widehat{\boldsymbol{\Lambda}}_{k+1,2};$$

- Let $\mathbf{G}_1 = \dfrac{\partial \mathbf{f}\left(\underline{\mathbf{y}}_{k,1}\right)}{\partial \mathbf{y}}$ and $\mathbf{G}_2 = \dfrac{\partial \mathbf{f}\left(\underline{\mathbf{y}}_{k,2}\right)}{\partial \mathbf{y}}$, with

$$\underline{\mathbf{y}}_{k,1} = \mathbf{y}_{k-1} + \frac{\tau_k}{4}\mathbf{Y}_{k,1} + \frac{\tau_k}{24}(6 - 4\sqrt{3})\,\mathbf{Y}_{k,2}$$

$$\underline{\mathbf{y}}_{k,2} = \mathbf{y}_{k-1} + \frac{\tau_k}{24}(6+4\sqrt{3})\,\mathbf{Y}_{k,1} + \frac{\tau_k}{4}\mathbf{Y}_{k,2}.$$

Solve for the internal stages of $\boldsymbol{\lambda}_{k-1}$:

$$\begin{bmatrix} \mathbf{I}_N - \frac{\tau_k}{4}\mathbf{G}_1^\top & -\frac{\tau_k}{24}(6+4\sqrt{3})\mathbf{G}_2^\top \\ -\frac{\tau_k}{24}(6-4\sqrt{3})\mathbf{G}_1^\top & \mathbf{I}_N - \frac{\tau_k}{4}\mathbf{G}_2^\top \end{bmatrix} \begin{bmatrix} \boldsymbol{\Lambda}_{k,1} \\ \boldsymbol{\Lambda}_{k,2} \end{bmatrix} = \begin{bmatrix} \boldsymbol{\Theta}_{k,1} \\ \boldsymbol{\Theta}_{k,2} \end{bmatrix} + \frac{\tau_k}{2}\begin{bmatrix} \boldsymbol{\lambda}_k \\ \boldsymbol{\lambda}_k \end{bmatrix},$$

then let $\widehat{\boldsymbol{\Lambda}}_{k,1} = \mathbf{G}_1^\top \boldsymbol{\Lambda}_{k,1}$ and $\widehat{\boldsymbol{\Lambda}}_{k,2} = \mathbf{G}_2^\top \boldsymbol{\Lambda}_{k,2}$.

6.2 Implicit-Explicit Time-Stepping Methods

This section presents the adjoint IMEX LM and IMEX RK time-stepping methods.

6.2.1 IMEX Linear Multistep Methods

Let $\overline{\boldsymbol{\lambda}} = \begin{bmatrix} \boldsymbol{\lambda}_1^\top & \cdots & \boldsymbol{\lambda}_K^\top \end{bmatrix}^\top$ represent the adjoint solution and $\overline{\boldsymbol{\theta}} = \begin{bmatrix} \boldsymbol{\theta}_1^\top & \cdots & \boldsymbol{\theta}_K^\top \end{bmatrix}^\top$ the adjoint source.

The transpose of the $K \times K$ block matrix $\dfrac{\partial \mathbf{t}}{\partial \overline{\mathbf{y}}}$ is an upper block triangular matrix that has the form of the transposes of the template matrices (2.1) and (2.3). Its (k,j)th $N \times N$ block is (see (A.83))

$$\frac{\partial \mathbf{t}_j}{\partial \mathbf{y}_k} = \begin{cases} \alpha_{j-k}^{(\sigma)}\mathbf{I}_N - \tau\,\beta_{j-k}^{(\sigma)}\dfrac{\partial \mathbf{f}_k^{\mathcal{E}}}{\partial \mathbf{y}} - \tau\,\gamma_{j-k}^{(\sigma)}\dfrac{\partial \mathbf{f}_k^{\mathcal{I}}}{\partial \mathbf{y}} & \text{if } k \le j \le k^\star \\ \mathbf{0}_{N\times N} & \text{otherwise,} \end{cases} \tag{6.4}$$

with $\beta_0^{(\sigma)} = 0$, $\sigma = \min(j,s)$ and $k^\star = \min(K, k+s)$.

To solve (6.1) we must solve, for $1 \le k \le K$,

$$\frac{\partial \mathbf{t}_k}{\partial \mathbf{y}_k}^\top \boldsymbol{\lambda}_k = \boldsymbol{\theta}_k - \sum_{j=k+1}^{k^\star} \frac{\partial \mathbf{t}_j}{\partial \mathbf{y}_k}^\top \boldsymbol{\lambda}_j.$$

The solution procedure is summarized in Algorithm 6.3.

Algorithm 6.3. The Adjoint Implicit-Explicit Linear Multistep Method

Let $\sigma = \min(k, s)$ *and* $k^\star = \max(1, k-s)$.

For $k = K, \cdots, 1$, *solve for* $\boldsymbol{\lambda}_k$:

$$\left(\mathbf{I}_N - \tau\gamma_0^{(\sigma)}\frac{\partial \mathbf{f}_k^{\mathcal{I}}}{\partial \mathbf{y}}^\top\right)\boldsymbol{\lambda}_k = \boldsymbol{\theta}_k - \sum_{j=k+1}^{k^\star}\alpha_{j-k}^{(\sigma)}\boldsymbol{\lambda}_j + \tau\frac{\partial \mathbf{f}_k^{\mathcal{E}}}{\partial \mathbf{y}}^\top\left(\sum_{j=k+1}^{k^\star}\beta_{j-k}^{(\sigma)}\boldsymbol{\lambda}_j\right) + \\ + \tau\frac{\partial \mathbf{f}_k^{\mathcal{I}}}{\partial \mathbf{y}}^\top\left(\sum_{j=k+1}^{k^\star}\gamma_{j-k}^{(\sigma)}\boldsymbol{\lambda}_j\right)$$

with $\dfrac{\partial \mathbf{f}_k^{\mathcal{I}/\mathcal{E}}}{\partial \mathbf{y}} = \dfrac{\partial \mathbf{f}^{\mathcal{I}/\mathcal{E}}(\mathbf{y}_k)}{\partial \mathbf{y}} = \dfrac{\partial \mathbf{f}^{\mathcal{I}/\mathcal{E}}(\mathbf{y}_k, t_k)}{\partial \mathbf{y}}$.

The right-hand side requires just two matrix-vector products, in contrast to the linearized forward IMEX linear multistep method, where $2s$ matrix-vector products are needed on the right-hand side at each time level.

Examples

We find the adjoint time-stepping methods corresponding to the IMEX linear multistep methods mentioned in Section 2.2.1. In all cases we have $\boldsymbol{\lambda}_k = \mathbf{0}_{N\times 1}$ for $k > K$.

For the adjoint IMEX BDF2 method one needs to solve, for $k = 2, \cdots, K$,

$$\left(\mathbf{I}_N - \tau\frac{2}{3}\frac{\partial \mathbf{f}^{\mathcal{I}}(\mathbf{y}_k)}{\partial \mathbf{y}}\right)\boldsymbol{\lambda}_k = \boldsymbol{\theta}_k + \left(\frac{4}{3}\boldsymbol{\lambda}_{k+1} - \frac{1}{3}\boldsymbol{\lambda}_{k+2}\right) + \\ + \tau\frac{\partial \mathbf{f}^{\mathcal{E}}(\mathbf{y}_k)}{\partial \mathbf{y}}^\top\left(\frac{4}{3}\boldsymbol{\lambda}_{k+1} - \frac{2}{3}\boldsymbol{\lambda}_{k+2}\right).$$

For the adjoint IMEX BDF3 method one needs to solve, for $k = 3, \cdots, K$,

$$\left(\mathbf{I}_N - \tau\frac{2}{3}\frac{\partial \mathbf{f}^{\mathcal{I}}(\mathbf{y}_k)}{\partial \mathbf{y}}\right)\boldsymbol{\lambda}_k = \boldsymbol{\theta}_k + \left(\frac{18}{11}\boldsymbol{\lambda}_{k+1} - \frac{9}{11}\boldsymbol{\lambda}_{k+2} + \frac{2}{11}\boldsymbol{\lambda}_{k+3}\right) +$$

$$+\tau\frac{\partial\mathbf{f}^{\mathcal{E}}\left(\mathbf{y}_k\right)}{\partial\mathbf{y}}^{\top}\left(\frac{18}{11}\boldsymbol{\lambda}_{k+1}-\frac{18}{11}\boldsymbol{\lambda}_{k+2}+\frac{6}{11}\boldsymbol{\lambda}_{k+3}\right).$$

Finally, for the adjoint IMEX TVB3 method one needs to solve, for $k=3,\cdots,K$,

$$\begin{aligned}\left(\mathbf{I}_N-\tau\frac{1089}{2048}\frac{\partial\mathbf{f}^{\mathcal{I}}\left(\mathbf{y}_k\right)}{\partial\mathbf{y}}\right)\boldsymbol{\lambda}_k=\boldsymbol{\theta}_k&+\left(\frac{3909}{2048}\boldsymbol{\lambda}_{k+1}-\frac{1367}{1024}\boldsymbol{\lambda}_{k+2}+\frac{873}{2048}\boldsymbol{\lambda}_{k+3}\right)+\\&+\frac{\partial\mathbf{f}^{\mathcal{E}}\left(\mathbf{y}_k\right)}{\partial\mathbf{y}}^{\top}\left(\frac{18463}{12288}\boldsymbol{\lambda}_{k+1}-\frac{1271}{768}\boldsymbol{\lambda}_{k+2}+\frac{8233}{12288}\boldsymbol{\lambda}_{k+3}\right)+\\&+\frac{\partial\mathbf{f}^{\mathcal{I}}\left(\mathbf{y}_k\right)}{\partial\mathbf{y}}^{\top}\left(-\frac{1139}{12288}\boldsymbol{\lambda}_{k+1}-\frac{367}{6144}\boldsymbol{\lambda}_{k+2}+\frac{1699}{12288}\boldsymbol{\lambda}_{k+3}\right).\end{aligned}$$

6.2.2 IMEX Runge-Kutta Methods

We let $\overline{\boldsymbol{\lambda}}=\left[\widehat{\boldsymbol{\lambda}}_1^{\top}\quad\cdots\quad\widehat{\boldsymbol{\lambda}}_K^{\top}\right]^{\top}$ represent the adjoint solution, with $\widehat{\boldsymbol{\lambda}}_k=\left[\boldsymbol{\Lambda}_k^{\top}\quad\boldsymbol{\lambda}_k^{\top}\right]^{\top}$, where

$$\boldsymbol{\Lambda}_k=\left[\boldsymbol{\Lambda}_{k,1}^{\mathcal{E}}{}^{\top}\quad\boldsymbol{\Lambda}_{k,1}^{\mathcal{I}}{}^{\top}\quad\boldsymbol{\Lambda}_{k,2}^{\mathcal{E}}{}^{\top}\quad\cdots\quad\boldsymbol{\Lambda}_{k,s-1}^{\mathcal{I}}{}^{\top}\quad\boldsymbol{\Lambda}_{k,s}^{\mathcal{E}}{}^{\top}\right]^{\top}$$

represents the $2s-1$ internal stages of the adjoint solution at the kth time step. The adjoint source $\overline{\boldsymbol{\theta}}$ is defined similarly, with internal stages $\boldsymbol{\Theta}_{k,i}$.

The linearization of the time-stepping equation, $\dfrac{\partial\mathbf{t}}{\partial\overline{\mathbf{y}}}$, was derived in Section A.2.2. Its transpose is (see (A.131))

$$\frac{\partial\mathbf{t}}{\partial\overline{\mathbf{y}}}^{\top}=\begin{bmatrix}\mathbf{A}_1^{\top}&\mathbf{B}&&&&&\\&\mathbf{I}_N&\mathbf{C}_2^{\top}&-\mathbf{I}_N&&&\\&&\mathbf{A}_2^{\top}&\mathbf{B}&&&\\&&&\ddots&\ddots&\ddots&\\&&&&\mathbf{I}_N&\mathbf{C}_K^{\top}&-\mathbf{I}_N\\&&&&&\mathbf{A}_K^{\top}&\mathbf{B}\\&&&&&&\mathbf{I}_N\end{bmatrix}.$$

where (see (A.127))

$$\mathbf{A}_k = \mathbf{I}_{(2s-1)N} - \tau_k \frac{\partial \mathbf{f}_k}{\partial \mathbf{y}} \left(\mathbf{A}_s \otimes \mathbf{I}_N\right) \qquad \text{and} \qquad \mathbf{C}_k = -\frac{\partial \mathbf{f}_k}{\partial \mathbf{y}} \left(\mathbf{1}_{2s-1} \otimes \mathbf{I}_N\right)$$

and

$$\mathbf{B} = -\tau_k \begin{bmatrix} b_1^{\mathcal{E}} & b_1^{\mathcal{I}} & b_2^{\mathcal{E}} & \cdots & b_{s-1}^{\mathcal{E}} & b_{s-1}^{\mathcal{I}} & b_s^{\mathcal{E}} \end{bmatrix}^\top \otimes \mathbf{I}_N.$$

See (A.125) for the definition of $\mathbf{A}_s$. $\dfrac{\partial \mathbf{f}_k}{\partial \mathbf{y}}$ was defined to be

$$\frac{\partial \mathbf{f}_k}{\partial \mathbf{y}} = \text{blkdiag}\left(\frac{\partial \mathbf{f}_{k,1}^{\mathcal{E}}}{\partial \mathbf{y}}, \frac{\partial \mathbf{f}_{k,2}^{\mathcal{I}}}{\partial \mathbf{y}}, \frac{\partial \mathbf{f}_{k,2}^{\mathcal{E}}}{\partial \mathbf{y}}, \frac{\partial \mathbf{f}_{k,3}^{\mathcal{I}}}{\partial \mathbf{y}}, \cdots, \frac{\partial \mathbf{f}_{k,s}^{\mathcal{E}}}{\partial \mathbf{y}}\right),$$

with $\mathbf{f}_{k,\sigma}^{\mathcal{E}} = \mathbf{f}^{\mathcal{E}}\left(\underline{\mathbf{y}}_{k,\sigma-1}, t_{k-1} + c_\sigma \tau_k\right)$, $\mathbf{f}_{k,\sigma}^{\mathcal{I}} = \mathbf{f}^{\mathcal{I}}\left(\underline{\mathbf{y}}_{k,\sigma-1}, t_{k-1} + c_\sigma \tau_k\right)$ and $\underline{\mathbf{y}}_{k,\sigma} = \mathbf{y}_{k-1} + \tau_k \sum_{i=1}^{\sigma} a_{\sigma+1,i}^{\mathcal{E}} \mathbf{Y}_{k,i}^{\mathcal{E}} + \tau_k \sum_{i=1}^{\sigma} a_{\sigma,i}^{\mathcal{I}} \mathbf{Y}_{k,i}^{\mathcal{I}}$.

Since $\dfrac{\partial \mathbf{t}}{\partial \overline{\mathbf{y}}}^\top$ is upper (block) triangular, we solve the adjoint system (6.1) by back-substitution:

$$\begin{aligned}
\boldsymbol{\lambda}_k &= \boldsymbol{\theta}_k + \boldsymbol{\lambda}_{k+1} - \mathbf{C}_{k+1}^\top \boldsymbol{\Lambda}_{k+1} \\
&= \boldsymbol{\theta}_k + \boldsymbol{\lambda}_{k+1} + \left(\mathbf{1}_{2s-1}^\top \otimes \mathbf{I}_N\right) \frac{\partial \mathbf{F}_{k+1}}{\partial \mathbf{y}}^\top \boldsymbol{\Lambda}_{k+1} \\
\boldsymbol{\Lambda}_k &= \mathbf{A}_k^{-\top} \left(\boldsymbol{\Theta}_k - \mathbf{B}\boldsymbol{\lambda}_k\right) \\
&= \left(\mathbf{I}_{(2s-1)N} - \tau_k \frac{\partial \mathbf{f}_k}{\partial \mathbf{y}} \left(\mathbf{A}_s \otimes \mathbf{I}_N\right)\right)^{-\top} \left(\boldsymbol{\Theta}_k - \mathbf{B}\boldsymbol{\lambda}_k\right).
\end{aligned} \tag{6.5}$$

The detailed solution procedure is summarized in Algorithm 6.4.

Algorithm 6.4. The Adjoint Implicit-Explicit Runge-Kutta Method

Let $\boldsymbol{\lambda}_{K+1} = \mathbf{0}_{N\times 1}$ *and* $\boldsymbol{\Lambda}_{K+1} = \mathbf{0}_{(2s-1)N\times 1}$*, and recall that we defined*

$$\underline{\mathbf{y}}_{k,\sigma} = \mathbf{y}_{k-1} + \tau_k \sum_{i=1}^{\sigma} a_{\sigma+1,i}^{\mathcal{E}} \mathbf{Y}_{k,i}^{\mathcal{E}} + \tau_k \sum_{i=1}^{\sigma} a_{\sigma,i}^{\mathcal{I}} \mathbf{Y}_{k,i}^{\mathcal{I}}.$$

Denote $\widehat{\mathbf{\Lambda}}^{\mathcal{E}}_{k,\sigma} = \dfrac{\partial \mathbf{f}^{\mathcal{E}}\left(\underline{\mathbf{y}}_{k,\sigma-1}\right)}{\partial \mathbf{y}}^{\top} \mathbf{\Lambda}^{\mathcal{E}}_{k,\sigma}$ *and* $\widehat{\mathbf{\Lambda}}^{\mathcal{I}}_{k,\sigma} = \dfrac{\partial \mathbf{f}^{\mathcal{I}}\left(\underline{\mathbf{y}}_{k,\sigma}\right)}{\partial \mathbf{y}}^{\top} \mathbf{\Lambda}^{\mathcal{I}}_{k,\sigma}$.
For $k = K, \cdots, 1$*:*

- *Update* $\boldsymbol{\lambda}$*:*

$$\boldsymbol{\lambda}_k = \boldsymbol{\theta}_k + \boldsymbol{\lambda}_{k+1} + \sum_{\sigma=1}^{s} \widehat{\mathbf{\Lambda}}^{\mathcal{E}}_{k+1,\sigma} + \sum_{\sigma=1}^{s-1} \widehat{\mathbf{\Lambda}}^{\mathcal{I}}_{k+1,\sigma}$$

- *Compute the internal stages for* $\boldsymbol{\lambda}_k$*. For* $i = 2s-1, \cdots, 1$*,*

  - *if* i *odd, let* $\sigma = \lceil \frac{i}{2} \rceil$ *and compute*

$$\mathbf{\Lambda}^{\mathcal{E}}_{k,\sigma} = \mathbf{\Theta}^{\mathcal{E}}_{k,\sigma} + \tau_k b^{\mathcal{E}}_{\sigma} \boldsymbol{\lambda}_k + \tau_k \sum_{j=\sigma+1}^{s} a^{\mathcal{E}}_{j\sigma} \widehat{\mathbf{\Lambda}}^{\mathcal{I}}_{k,j-1} + \tau_k \sum_{j=\sigma+1}^{s} a^{\mathcal{E}}_{j\sigma} \widehat{\mathbf{\Lambda}}^{\mathcal{E}}_{k,j}$$

$$\widehat{\mathbf{\Lambda}}^{\mathcal{E}}_{k,\sigma} = \frac{\partial \mathbf{f}^{\mathcal{E}}\left(\underline{\mathbf{y}}_{k,\sigma-1}\right)}{\partial \mathbf{y}}^{\top} \mathbf{\Lambda}^{\mathcal{E}}_{k,\sigma};$$

  - *if* i *even, let* $\sigma = \frac{i}{2}$ *and solve*

$$\left(\mathbf{I}_N - \tau_k a^{\mathcal{I}}_{\sigma\sigma} \frac{\partial \mathbf{f}^{\mathcal{I}}\left(\underline{\mathbf{y}}_{k,\sigma}\right)}{\partial \mathbf{y}}^{\top}\right) \mathbf{\Lambda}^{\mathcal{I}}_{k,\sigma} = \mathbf{\Theta}^{\mathcal{I}}_{k,\sigma} + \tau_k b^{\mathcal{I}}_{\sigma} \boldsymbol{\lambda}_k + \\ + \tau_k \sum_{j=\sigma+1}^{s-1} a^{\mathcal{I}}_{j\sigma} \widehat{\mathbf{\Lambda}}^{\mathcal{I}}_{k,j} + \tau_k \sum_{j=\sigma}^{s-1} a^{\mathcal{I}}_{j\sigma} \widehat{\mathbf{\Lambda}}^{\mathcal{E}}_{k,j+1}$$

$$\widehat{\mathbf{\Lambda}}^{\mathcal{I}}_{k,\sigma} = \frac{\partial \mathbf{f}^{\mathcal{I}}\left(\underline{\mathbf{y}}_{k,\sigma}\right)}{\partial \mathbf{y}}^{\top} \mathbf{\Lambda}^{\mathcal{I}}_{k,\sigma}.$$

Example

The adjoint ARS3 method is, with $\gamma = \frac{3+\sqrt{3}}{6}$,

$$\underline{\mathbf{y}}_{k,1} = \mathbf{y}_{k-1} + \tau_k \gamma \mathbf{Y}^{\mathcal{E}}_{k,1} + \tau_k \gamma \mathbf{Y}^{\mathcal{I}}_{k,1}$$
$$\underline{\mathbf{y}}_{k,2} = \mathbf{y}_{k-1} + \tau_k \left((\gamma-1)\mathbf{Y}^{\mathcal{E}}_{k,1} + 2(1-\gamma)\mathbf{Y}^{\mathcal{E}}_{k,2} + (1-2\gamma)\mathbf{Y}^{\mathcal{I}}_{k,1} + \gamma \mathbf{Y}^{\mathcal{I}}_{k,2}\right),$$

and for $k = K, \cdots, 1$:

- Update $\boldsymbol{\lambda}$:

$$\boldsymbol{\lambda}_k = \boldsymbol{\theta}_k + \boldsymbol{\lambda}_{k+1} + \widehat{\boldsymbol{\Lambda}}^{\mathcal{E}}_{k+1,1} + \widehat{\boldsymbol{\Lambda}}^{\mathcal{E}}_{k+1,2} + \widehat{\boldsymbol{\Lambda}}^{\mathcal{E}}_{k+1,3} + \widehat{\boldsymbol{\Lambda}}^{\mathcal{I}}_{k+1,1} + \widehat{\boldsymbol{\Lambda}}^{\mathcal{I}}_{k+1,2}.$$

- Compute the (modified) internal stages for $\boldsymbol{\lambda}$:
  - Compute $\widehat{\boldsymbol{\Lambda}}^{\mathcal{E}}_{k,3}$:

$$\widehat{\boldsymbol{\Lambda}}^{\mathcal{E}}_{k,3} = \frac{\partial \mathbf{f}^{\mathcal{E}}\left(\underline{\mathbf{y}}_{k,2}\right)}{\partial \mathbf{y}}^{\top} \left(\boldsymbol{\Theta}^{\mathcal{E}}_{k,3} + \frac{\tau_k}{2}\boldsymbol{\lambda}_k\right);$$

  - Solve for $\widehat{\boldsymbol{\Lambda}}^{\mathcal{I}}_{k,2}$:

$$\left(\mathbf{I}_N - \tau_k\gamma\frac{\partial \mathbf{f}^{\mathcal{I}}\left(\underline{\mathbf{y}}_{k,2}\right)}{\partial \mathbf{y}}^{\top}\right)\boldsymbol{\Lambda}^{\mathcal{I}}_{k,2} = \boldsymbol{\Theta}^{\mathcal{I}}_{k,2} + \frac{\tau_k}{2}\boldsymbol{\lambda}_k + \tau_k\gamma\widehat{\boldsymbol{\Lambda}}^{\mathcal{E}}_{k,3}$$

$$\widehat{\boldsymbol{\Lambda}}^{\mathcal{I}}_{k,2} = \frac{\partial \mathbf{f}^{\mathcal{I}}\left(\underline{\mathbf{y}}_{k,2}\right)}{\partial \mathbf{y}}^{\top} \boldsymbol{\Lambda}^{\mathcal{I}}_{k,2};$$

  - Compute $\widehat{\boldsymbol{\Lambda}}^{\mathcal{E}}_{k,2}$:

$$\widehat{\boldsymbol{\Lambda}}^{\mathcal{E}}_{k,2} = \frac{\partial \mathbf{f}^{\mathcal{E}}\left(\underline{\mathbf{y}}_{k,1}\right)}{\partial \mathbf{y}}^{\top} \left(\boldsymbol{\Theta}^{\mathcal{E}}_{k,2} + \frac{\tau_k}{2}\boldsymbol{\lambda}_k + 2\tau_k\left(1-\gamma\right)\left(\widehat{\boldsymbol{\Lambda}}^{\mathcal{I}}_{k,2} + \widehat{\boldsymbol{\Lambda}}^{\mathcal{E}}_{k,3}\right)\right);$$

  - Solve for $\widehat{\boldsymbol{\Lambda}}^{\mathcal{I}}_{k,1}$:

$$\left(\mathbf{I}_N - \tau_k\gamma\frac{\partial \mathbf{f}^{\mathcal{I}}\left(\underline{\mathbf{y}}_{k,1}\right)}{\partial \mathbf{y}}^{\top}\right)\boldsymbol{\Lambda}^{\mathcal{I}}_{k,1} = \boldsymbol{\Theta}^{\mathcal{I}}_{k,1} + \frac{\tau_k}{2}\boldsymbol{\lambda}_k + \tau_k\gamma\widehat{\boldsymbol{\Lambda}}^{\mathcal{E}}_{k,2} +$$

$$+ \tau_k(1-2\gamma)\left(\widehat{\boldsymbol{\Lambda}}^{\mathcal{I}}_{k,2} + \widehat{\boldsymbol{\Lambda}}^{\mathcal{E}}_{k,3}\right)$$

$$\widehat{\boldsymbol{\Lambda}}^{\mathcal{I}}_{k,1} = \frac{\partial \mathbf{f}^{\mathcal{I}}\left(\underline{\mathbf{y}}_{k,1}\right)}{\partial \mathbf{y}}^{\top} \boldsymbol{\Lambda}^{\mathcal{I}}_{k,1};$$

- Compute $\widehat{\boldsymbol{\Lambda}}^{\mathcal{E}}_{k,1}$:

$$\widehat{\boldsymbol{\Lambda}}^{\mathcal{E}}_{k,1} = \frac{\partial \mathbf{f}^{\mathcal{E}}\left(\mathbf{y}_{k-1}\right)}{\partial \mathbf{y}}^{\top} \left(\boldsymbol{\Theta}^{\mathcal{E}}_{k,1} + \tau_k \gamma \left(\widehat{\boldsymbol{\Lambda}}^{\mathcal{I}}_{k,1} + \widehat{\boldsymbol{\Lambda}}^{\mathcal{E}}_{k,2}\right) + \tau_k(\gamma - 1)\left(\widehat{\boldsymbol{\Lambda}}^{\mathcal{I}}_{k,2} + \widehat{\boldsymbol{\Lambda}}^{\mathcal{E}}_{k,3}\right)\right).$$

6.3 Staggered Time-Stepping Methods

In this section we discuss the adjoint StagLM and StagRK time-stepping methods.

6.3.1 Staggered Linear Multistep Methods

Let $\overline{\boldsymbol{\lambda}} = \begin{bmatrix}\boldsymbol{\lambda}_1^\top & \cdots & \boldsymbol{\lambda}_K^\top\end{bmatrix}^\top$ represent the adjoint solution and $\overline{\boldsymbol{\theta}} = \begin{bmatrix}\boldsymbol{\theta}_1^\top & \cdots & \boldsymbol{\theta}_K^\top\end{bmatrix}^\top$ the adjoint source. We have $\boldsymbol{\lambda}_k = \begin{bmatrix}\boldsymbol{\lambda}_k^{\mathbf{u}\top} & \boldsymbol{\lambda}_{k+\frac{1}{2}}^{\mathbf{v}\top}\end{bmatrix}^\top$ and likewise for $\boldsymbol{\theta}_k$.

The transpose of the $K \times K$ block matrix $\dfrac{\partial \mathbf{t}}{\partial \overline{\mathbf{y}}}$ is an upper block triangular matrix that has the form of the transposes of the template matrices (2.1) and (2.3). Its (k,j)th $N \times N$ block is (see (A.158))

$$\frac{\partial \mathbf{t}_j}{\partial \mathbf{y}_k} = \begin{cases} \alpha^{(\sigma)}_{j-k}\mathbf{I}_N - \tau\,\beta^{(\sigma)}_{j-k}\dfrac{\partial \mathbf{f}_k}{\partial \mathbf{y}_k}^{\top} - \tau\,\beta^{(\sigma)}_{j-k-1}\dfrac{\partial \mathbf{f}_{k+1}}{\partial \mathbf{y}_k}^{\top} & \text{if } k \le j \le k^\star \\ \mathbf{0}_{N\times N} & \text{otherwise,} \end{cases} \tag{6.6}$$

with $\beta^{(\sigma)}_\ell = 0$ for $\ell < 0$, $\sigma = \min(j,s)$, $k^\star = \min(K, k+s)$, and $\dfrac{\partial \mathbf{f}_k}{\partial \mathbf{y}_k}$ and $\dfrac{\partial \mathbf{f}_{k+1}}{\partial \mathbf{y}_k}$ defined as in (A.157).

To solve (6.1) we must solve, for $1 \le k \le K$,

$$\frac{\partial \mathbf{t}_k}{\partial \mathbf{y}_k}^{\top} \boldsymbol{\lambda}_k = \boldsymbol{\theta}_k - \sum_{j=k+1}^{k^\star} \frac{\partial \mathbf{t}_j}{\partial \mathbf{y}_k}^{\top} \boldsymbol{\lambda}_j.$$

The solution procedure is summarized in Algorithm 6.5.

Algorithm 6.5. The Adjoint Staggered Linear Multistep Method

Let $\sigma = \min(j, s)$ *and* $k^\star = \min(K, k+s)$.

For $k = K, \cdots, 1$*:*

- *Update* $\boldsymbol{\lambda}^{\mathbf{v}}$*:*

$$\boldsymbol{\lambda}^{\mathbf{v}}_{k+\frac{1}{2}} = \boldsymbol{\theta}^{\mathbf{v}}_{k+\frac{1}{2}} - \sum_{j=k+1}^{k^\star} \alpha^{(\sigma)}_{j-k} \boldsymbol{\lambda}^{\mathbf{v}}_{j+\frac{1}{2}} + \tau \frac{\partial \mathbf{f}^{\mathbf{u}}\left(\mathbf{v}_{k+\frac{1}{2}}, t_{k+\frac{1}{2}}\right)}{\partial \mathbf{v}}^{\top} \left(\sum_{j=k+1}^{k^\star} \beta^{(\sigma)}_{j-k-1} \boldsymbol{\lambda}^{\mathbf{u}}_{j} \right).$$

- *Update* $\boldsymbol{\lambda}^{\mathbf{u}}$*:*

$$\boldsymbol{\lambda}^{\mathbf{u}}_{k} = \boldsymbol{\theta}^{\mathbf{u}}_{k} - \sum_{j=k+1}^{k^\star} \alpha^{(\sigma)}_{j-k} \boldsymbol{\lambda}^{\mathbf{u}}_{j} + \tau \frac{\partial \mathbf{f}^{\mathbf{v}}\left(\mathbf{u}_k, t_k\right)}{\partial \mathbf{u}}^{\top} \left(\sum_{j=k}^{k^\star} \beta^{(\sigma)}_{j-k} \boldsymbol{\lambda}^{\mathbf{v}}_{j+\frac{1}{2}} \right)$$

The right-hand side requires just two matrix-vector products, in contrast to the linearized forward staggered linear multistep method, where $2s$ matrix-vector products are needed on the right-hand side at each time level.

Examples

We find the adjoint time-stepping methods of the staggered linear multistep methods mentioned in Section 2.3.1. In all cases we have $\boldsymbol{\lambda}^{\mathbf{v}}_{k+\frac{1}{2}} = \mathbf{0}_{N\times 1}$ and $\boldsymbol{\lambda}^{\mathbf{u}}_{k} = \mathbf{0}$ for $k > K$.

The adjoint leapfrog method for $k = K, \cdots, 1$ is

$$\begin{aligned}
\boldsymbol{\lambda}^{\mathbf{v}}_{k+\frac{1}{2}} &= \boldsymbol{\theta}^{\mathbf{v}}_{k+\frac{1}{2}} + \boldsymbol{\lambda}^{\mathbf{v}}_{k+\frac{3}{2}} + \tau \frac{\partial \mathbf{f}^{\mathbf{u}}\left(\mathbf{v}_{k+\frac{1}{2}}, t_{k+\frac{1}{2}}\right)}{\partial \mathbf{v}}^{\top} \boldsymbol{\lambda}^{\mathbf{u}}_{k+1} \\
\boldsymbol{\lambda}^{\mathbf{u}}_{k} &= \boldsymbol{\theta}^{\mathbf{u}}_{k} + \boldsymbol{\lambda}^{\mathbf{u}}_{k+1} + \tau \frac{\partial \mathbf{f}^{\mathbf{v}}\left(\mathbf{u}_k, t_k\right)}{\partial \mathbf{u}}^{\top} \boldsymbol{\lambda}^{\mathbf{v}}_{k+\frac{1}{2}}.
\end{aligned}$$

The adjoint StagBDF3 method for $k = K, \cdots, 3$ is

$$\begin{aligned}
\boldsymbol{\lambda}^{\mathbf{v}}_{k+\frac{1}{2}} &= \boldsymbol{\theta}^{\mathbf{v}}_{k+\frac{1}{2}} + \frac{21}{23}\boldsymbol{\lambda}^{\mathbf{v}}_{k+\frac{3}{2}} + \frac{3}{23}\boldsymbol{\lambda}^{\mathbf{v}}_{k+\frac{5}{2}} - \frac{1}{23}\boldsymbol{\lambda}^{\mathbf{v}}_{k+\frac{7}{2}} + \tau\frac{24}{23}\frac{\partial \mathbf{f}^{\mathbf{u}}\left(\mathbf{v}_{k+\frac{1}{2}}, t_{k+\frac{1}{2}}\right)}{\partial \mathbf{v}}^{\top}\boldsymbol{\lambda}^{\mathbf{u}}_{k+1} \\
\boldsymbol{\lambda}^{\mathbf{u}}_{k} &= \boldsymbol{\theta}^{\mathbf{u}}_{k} + \frac{21}{23}\boldsymbol{\lambda}^{\mathbf{u}}_{k+1} + \frac{3}{23}\boldsymbol{\lambda}^{\mathbf{u}}_{k+2} - \frac{1}{23}\boldsymbol{\lambda}^{\mathbf{u}}_{k+3} + \tau\frac{24}{23}\frac{\partial \mathbf{f}^{\mathbf{v}}\left(\mathbf{u}_k, t_k\right)}{\partial \mathbf{u}}^{\top}\boldsymbol{\lambda}^{\mathbf{v}}_{k+\frac{1}{2}}.
\end{aligned}$$

The adjoint StagBDF4 method for $k = K, \cdots, 4$ is

$$\begin{aligned}
\boldsymbol{\lambda}^{\mathbf{v}}_{k+\frac{1}{2}} &= \boldsymbol{\theta}^{\mathbf{v}}_{k+\frac{1}{2}} + \frac{17}{22}\boldsymbol{\lambda}^{\mathbf{v}}_{k+\frac{3}{2}} + \frac{9}{22}\boldsymbol{\lambda}^{\mathbf{v}}_{k+\frac{5}{2}} - \frac{5}{22}\boldsymbol{\lambda}^{\mathbf{v}}_{k+\frac{7}{2}} + \frac{1}{22}\boldsymbol{\lambda}^{\mathbf{v}}_{k+\frac{9}{2}} + \\
&\qquad + \tau\frac{12}{11}\frac{\partial \mathbf{f}^{\mathbf{u}}\left(\mathbf{v}_{k+\frac{1}{2}}, t_{k+\frac{1}{2}}\right)}{\partial \mathbf{v}}^{\top}\boldsymbol{\lambda}^{\mathbf{u}}_{k+1} \\
\boldsymbol{\lambda}^{\mathbf{u}}_{k} &= \boldsymbol{\theta}^{\mathbf{u}}_{k} + \frac{17}{22}\boldsymbol{\lambda}^{\mathbf{u}}_{k+1} + \frac{9}{22}\boldsymbol{\lambda}^{\mathbf{u}}_{k+2} - \frac{5}{22}\boldsymbol{\lambda}^{\mathbf{u}}_{k+3} + \frac{1}{22}\boldsymbol{\lambda}^{\mathbf{u}}_{k+4} + \tau\frac{12}{11}\frac{\partial \mathbf{f}^{\mathbf{v}}\left(\mathbf{u}_k, t_k\right)}{\partial \mathbf{u}}^{\top}\boldsymbol{\lambda}^{\mathbf{v}}_{k+\frac{1}{2}}.
\end{aligned}$$

The adjoint StagAB3 scheme for $k = K, \cdots, 3$ is

$$\begin{aligned}
\boldsymbol{\lambda}^{\mathbf{v}}_{k+\frac{1}{2}} &= \boldsymbol{\theta}^{\mathbf{v}}_{k+\frac{1}{2}} + \boldsymbol{\lambda}^{\mathbf{v}}_{k+\frac{3}{2}} + \tau\frac{\partial \mathbf{f}^{\mathbf{u}}\left(\mathbf{v}_{k+\frac{1}{2}}, t_{k+\frac{1}{2}}\right)}{\partial \mathbf{v}}^{\top}\left(\frac{25}{24}\boldsymbol{\lambda}^{\mathbf{u}}_{k+1} - \frac{1}{12}\boldsymbol{\lambda}^{\mathbf{u}}_{k+2} + \frac{1}{24}\boldsymbol{\lambda}^{\mathbf{u}}_{k+3}\right) \\
\boldsymbol{\lambda}^{\mathbf{u}}_{k} &= \boldsymbol{\theta}^{\mathbf{u}}_{k} + \boldsymbol{\lambda}^{\mathbf{u}}_{k+1} + \tau\frac{\partial \mathbf{f}^{\mathbf{v}}\left(\mathbf{u}_k, t_k\right)}{\partial \mathbf{u}}^{\top}\left(\frac{25}{24}\boldsymbol{\lambda}^{\mathbf{v}}_{k+\frac{1}{2}} - \frac{1}{12}\boldsymbol{\lambda}^{\mathbf{v}}_{k+\frac{3}{2}} + \frac{1}{24}\boldsymbol{\lambda}^{\mathbf{v}}_{k+\frac{5}{2}}\right).
\end{aligned}$$

Finally, the adjoint StagAB4 scheme for $k = K, \cdots, 4$ is

$$\begin{aligned}
\boldsymbol{\lambda}^{\mathbf{v}}_{k+\frac{1}{2}} &= \boldsymbol{\theta}^{\mathbf{v}}_{k+\frac{1}{2}} + \boldsymbol{\lambda}^{\mathbf{v}}_{k+\frac{3}{2}} + \frac{\tau}{24}\frac{\partial \mathbf{f}^{\mathbf{u}}\left(\mathbf{v}_{k+\frac{1}{2}}, t_{k+\frac{1}{2}}\right)}{\partial \mathbf{v}}^{\top}\left(26\boldsymbol{\lambda}^{\mathbf{u}}_{k+1} - 5\boldsymbol{\lambda}^{\mathbf{u}}_{k+2} + 4\boldsymbol{\lambda}^{\mathbf{u}}_{k+3} - \boldsymbol{\lambda}^{\mathbf{u}}_{k+4}\right) \\
\boldsymbol{\lambda}^{\mathbf{u}}_{k} &= \boldsymbol{\theta}^{\mathbf{u}}_{k} + \boldsymbol{\lambda}^{\mathbf{u}}_{k+1} + \frac{\tau}{24}\frac{\partial \mathbf{f}^{\mathbf{v}}\left(\mathbf{u}_k, t_k\right)}{\partial \mathbf{u}}^{\top}\left(26\boldsymbol{\lambda}^{\mathbf{v}}_{k+\frac{1}{2}} - 5\boldsymbol{\lambda}^{\mathbf{v}}_{k+\frac{3}{2}} + 4\boldsymbol{\lambda}^{\mathbf{v}}_{k+\frac{5}{2}} - \boldsymbol{\lambda}^{\mathbf{v}}_{k+\frac{7}{2}}\right).
\end{aligned}$$

6.3.2 Staggered Runge-Kutta Methods

We let $\overline{\boldsymbol{\lambda}} = \begin{bmatrix}\widehat{\boldsymbol{\lambda}}_1^{\top} & \cdots & \widehat{\boldsymbol{\lambda}}_K^{\top}\end{bmatrix}^{\top}$ represent the adjoint solution, with $\widehat{\boldsymbol{\lambda}}_k = \begin{bmatrix}\boldsymbol{\Lambda}_k^{\top} & \boldsymbol{\lambda}_k^{\top}\end{bmatrix}^{\top}$, where

$$\boldsymbol{\lambda}_k = \begin{bmatrix}\boldsymbol{\lambda}_k^{\mathbf{u}\top} & \boldsymbol{\lambda}^{\mathbf{v}}_{k+\frac{1}{2}}{}^{\top}\end{bmatrix}^{\top} \qquad \text{and} \qquad \boldsymbol{\Lambda}_{k,\sigma} = \begin{bmatrix}\boldsymbol{\Lambda}^{\mathbf{u}}_{k,\sigma}{}^{\top} & \boldsymbol{\Lambda}^{\mathbf{v}}_{k+\frac{1}{2},\sigma}{}^{\top}\end{bmatrix}^{\top},$$

the latter representing the s internal stages of the adjoint solution at the kth time step. The adjoint source $\overline{\boldsymbol{\theta}}$ is defined similarly, with internal stages $\boldsymbol{\Theta}_{k,\sigma}$.

The linearization of the time-stepping equation, $\dfrac{\partial \mathbf{t}}{\partial \overline{\mathbf{y}}}$, was derived in Section A.3.2. Its transpose is (see (A.200))

$$\frac{\partial \mathbf{t}}{\partial \overline{\mathbf{y}}}^{\top} = \begin{bmatrix} \mathbf{A}_1^\top & \mathbf{B} & & & & & \\ \mathbf{C}_{1,1}^\top & \mathbf{I}_N & \mathbf{C}_{2,1}^\top & -\mathbf{I}_N & & & \\ & & \mathbf{A}_2^\top & \mathbf{B} & & & \\ & & \ddots & \ddots & \ddots & \ddots & \\ & & & \mathbf{C}_{K-1,K-1}^\top & \mathbf{I}_N & \mathbf{C}_{K,K-1}^\top & -\mathbf{I}_N \\ & & & & & \mathbf{A}_K^\top & \mathbf{B} \\ & & & & & \mathbf{C}_{K,K}^\top & \mathbf{I}_N \end{bmatrix}.$$

$\mathbf{A}_k$ and $\mathbf{C}_{k,j}$ were defined in (A.195) to be

$$\begin{aligned} \mathbf{A}_k &= \mathbf{I}_{sN} - \tau\, \mathbf{G}_k^{\mathrm{o/e}} \left(\mathbf{A}_s \otimes \mathbf{I}_N\right) \\ \mathbf{C}_{k,j} &= -\frac{\partial \mathbf{F}_k^{\mathrm{o/e}}}{\partial \mathbf{y}_j} \\ \mathbf{B} &= \mathbf{B}^{\mathrm{o/e}}. \end{aligned}$$

See Section A.3.2 for the definitions of $\mathbf{G}_k^{\mathrm{o/e}}$ and $\dfrac{\partial \mathbf{F}_k^{\mathrm{o/e}}}{\partial \mathbf{y}_j}$, which contain the derivatives of $\mathbf{f}^{\mathbf{u}}$ and $\mathbf{f}^{\mathbf{v}}$ with respect to $\mathbf{u}$ and $\mathbf{v}$. Important for this is the shorthand

$$\underline{\mathbf{v}}^{o/e}_{\mathbf{U},k-\frac{1}{2},\sigma} = \mathbf{v}_{k-\frac{1}{2}} + \tau \sum_{\substack{i=1 \\ i \text{ even}}}^{\sigma-1} a_{\sigma i} \mathbf{U}^{o/e}_{k,i}, \qquad \underline{\mathbf{v}}^{o/e}_{\mathbf{V},k-\frac{1}{2},\sigma} = \mathbf{v}_{k-\frac{1}{2}} + \tau \sum_{\substack{i=1 \\ i \text{ odd}}}^{\sigma-1} a_{\sigma i} \mathbf{V}^{o/e}_{k+\frac{1}{2},i}$$

$$\underline{\mathbf{u}}^{o/e}_{\mathbf{U},k-1,\sigma} = \mathbf{u}_{k-1} + \tau \sum_{\substack{i=1 \\ i \text{ odd}}}^{\sigma-1} a_{\sigma i} \mathbf{U}^{o/e}_{k,i}, \qquad \underline{\mathbf{u}}^{o/e}_{\mathbf{V},k,\sigma} = \mathbf{u}_k + \tau \sum_{\substack{i=1 \\ i \text{ even}}}^{\sigma-1} a_{\sigma i} \mathbf{V}^{o/e}_{k+\frac{1}{2},i},$$

where o/e denotes whether the number of stages s is odd or even.

While not apparent from the form above, $\dfrac{\partial \mathbf{t}}{\partial \overline{\mathbf{y}}}^{\top}$ is essentially upper (block) triangular (we can swap rows and columns to make it upper block diagonal), and therefore we solve the adjoint system (6.1) by backsubstitution:

$$\begin{aligned}
\boldsymbol{\lambda}_k &= \boldsymbol{\theta}_k + \boldsymbol{\lambda}_{k+1} - \mathbf{C}_{k,k}^{\top}\boldsymbol{\Lambda}_k - \mathbf{C}_{k+1,k}^{\top}\boldsymbol{\Lambda}_{k+1} \\
&= \boldsymbol{\theta}_k + \boldsymbol{\lambda}_{k+1} + \frac{\partial \mathbf{F}_k^{\mathrm{o/e}}}{\partial \mathbf{y}_k}^{\top} \boldsymbol{\Lambda}_k + \frac{\partial \mathbf{F}_{k+1}^{\mathrm{o/e}}}{\partial \mathbf{y}_k}^{\top} \boldsymbol{\Lambda}_{k+1} \\
\boldsymbol{\Lambda}_k &= \mathbf{A}_k^{-\top}\left(\boldsymbol{\Theta}_k - \mathbf{B}\boldsymbol{\lambda}_k\right) \\
&= \left(\mathbf{I}_{sN} - \tau\,\mathbf{G}_k^{\mathrm{o/e}}\left(\mathbf{A}_s \otimes \mathbf{I}_N\right)\right)^{-\top}\left(\boldsymbol{\Theta}_k - \mathbf{B}\boldsymbol{\lambda}_k\right).
\end{aligned} \tag{6.7}$$

The detailed solution procedure for s odd is summarized in Algorithm 6.6.

Algorithm 6.6. The Adjoint Staggered Runge-Kutta Method, for s odd

Let $\boldsymbol{\lambda}_{K+1} = \mathbf{0}_{N\times 1}$ *and* $\boldsymbol{\Lambda}_{K+1} = \mathbf{0}_{sN\times 1}$. *Recall that (see* (2.44)*)*

$$\underline{\mathbf{v}}^o_{\mathbf{U},k-\frac{1}{2},\sigma} = \mathbf{v}_{k-\frac{1}{2}} + \tau \sum_{\substack{i=1\\ i\text{ even}}}^{\sigma-1} a_{\sigma i}\mathbf{U}^o_{k,i}, \qquad \underline{\mathbf{v}}^o_{\mathbf{V},k-\frac{1}{2},\sigma} = \mathbf{v}_{k-\frac{1}{2}} + \tau \sum_{\substack{i=1\\ i\text{ odd}}}^{\sigma-1} a_{\sigma i}\mathbf{V}^o_{k+\frac{1}{2},i}$$

$$\underline{\mathbf{u}}^o_{\mathbf{U},k-1,\sigma} = \mathbf{u}_{k-1} + \tau \sum_{\substack{i=1\\ i\text{ odd}}}^{\sigma-1} a_{\sigma i}\mathbf{U}^o_{k,i}, \qquad \underline{\mathbf{u}}^o_{\mathbf{V},k,\sigma} = \mathbf{u}_k + \tau \sum_{\substack{i=1\\ i\text{ even}}}^{\sigma-1} a_{\sigma i}\mathbf{V}^o_{k+\frac{1}{2},i}.$$

For $k = K, \cdots, 1$*:*

- *Compute*

$$\boldsymbol{\lambda}^{\mathbf{v}}_{k+\frac{1}{2}} = \mathbf{q}^{\mathbf{v}}_{k+\frac{1}{2}} + \boldsymbol{\lambda}^{\mathbf{v}}_{k+\frac{3}{2}} + \sum_{\substack{\sigma=1\\ \sigma\text{ odd}}}^{s} \widehat{\boldsymbol{\Lambda}}^{\mathbf{u}}_{k+1,\sigma} + \sum_{\substack{\sigma=1\\ \sigma\text{ even}}}^{s} \widehat{\boldsymbol{\Lambda}}^{\mathbf{v}}_{k+\frac{3}{2},\sigma}$$

- *Compute the (modified) internal stages for $\boldsymbol{\lambda}^{\mathbf{v}}$, for $\sigma = s, \cdots, 1$:*

$$\widehat{\boldsymbol{\Lambda}}^{\mathbf{v}}_{k+\frac{1}{2},\sigma} = \boldsymbol{\Theta}^{\mathbf{v}}_{k+\frac{1}{2},\sigma} + \tau \begin{cases} \dfrac{\partial \mathbf{f}^{\mathbf{v}}\left(\underline{\mathbf{u}}^{\mathrm{o}}_{\mathbf{V},k,\sigma}\right)}{\partial \mathbf{u}}^{\top} \left(\sum\limits_{\substack{i=\sigma+1 \\ i \text{ even}}}^{s} a_{i\sigma} \widehat{\boldsymbol{\Lambda}}^{\mathbf{v}}_{k+\frac{1}{2},i} + b_{\sigma} \boldsymbol{\lambda}^{\mathbf{v}}_{k+\frac{1}{2}} \right) & \text{if } \sigma \text{ odd} \\ \dfrac{\partial \mathbf{f}^{\mathbf{u}}\left(\underline{\mathbf{v}}^{\mathrm{o}}_{\mathbf{V},k-\frac{1}{2},\sigma}\right)}{\partial \mathbf{v}}^{\top} \left(\sum\limits_{\substack{i=\sigma+1 \\ i \text{ odd}}}^{s} a_{i\sigma} \widehat{\boldsymbol{\Lambda}}^{\mathbf{v}}_{k+\frac{1}{2},i} \right) & \text{if } \sigma \text{ even} \end{cases}$$

- *Compute*

$$\boldsymbol{\lambda}^{\mathbf{u}}_{k} = \mathbf{q}^{\mathbf{u}}_{k} + \boldsymbol{\lambda}^{\mathbf{u}}_{k+1} + \sum_{\substack{\sigma=1 \\ \sigma \text{ odd}}}^{s} \widehat{\boldsymbol{\Lambda}}^{\mathbf{v}}_{k+\frac{1}{2},\sigma} + \sum_{\substack{\sigma=1 \\ \sigma \text{ even}}}^{s} \widehat{\boldsymbol{\Lambda}}^{\mathbf{u}}_{k+1,\sigma}$$

- *Compute the internal stages for $\boldsymbol{\lambda}^{\mathbf{u}}_{k}$, for $\sigma = s, \cdots, 1$:*

$$\widehat{\boldsymbol{\Lambda}}^{\mathbf{u}}_{k,\sigma} = \boldsymbol{\Theta}^{\mathbf{u}}_{k,\sigma} + \tau \begin{cases} \dfrac{\partial \mathbf{f}^{\mathbf{u}}\left(\underline{\mathbf{v}}^{\mathrm{o}}_{\mathbf{U},k-\frac{1}{2},\sigma}\right)}{\partial \mathbf{v}}^{\top} \left(\sum\limits_{\substack{i=\sigma+1 \\ i \text{ even}}}^{s} a_{i\sigma} \widehat{\boldsymbol{\Lambda}}^{\mathbf{u}}_{k,i} + b_{\sigma} \boldsymbol{\lambda}^{\mathbf{u}}_{k} \right) & \text{if } \sigma \text{ odd} \\ \dfrac{\partial \mathbf{f}^{\mathbf{v}}\left(\underline{\mathbf{u}}^{\mathrm{o}}_{\mathbf{U},k-1,\sigma}\right)}{\partial \mathbf{u}}^{\top} \left(\sum\limits_{\substack{i=\sigma+1 \\ i \text{ odd}}}^{s} a_{i\sigma} \widehat{\boldsymbol{\Lambda}}^{\mathbf{u}}_{k,i} \right) & \text{if } \sigma \text{ even} \end{cases}$$

The solution procedure for even s is similar, see Section 2.3.2 and Appendix A.3.2 for guidance.

Example

The adjoint time-stepping method corresponding to the 5-stage StagRK4 method with parameter b_5 and $\gamma = (6\,b_5)^{-1/2}$ (see Section 2.3.2) is:
For $k = K, \cdots, 1$:

- Compute the update for $\boldsymbol{\lambda}^{\mathbf{v}}$:

$$\boldsymbol{\lambda}^{\mathbf{v}}_{k+\frac{1}{2}} = \mathbf{q}^{\mathbf{v}}_{k+\frac{1}{2}} + \boldsymbol{\lambda}^{\mathbf{v}}_{k+\frac{3}{2}} + \widehat{\boldsymbol{\Lambda}}^{\mathbf{u}}_{k+1,1} + \widehat{\boldsymbol{\Lambda}}^{\mathbf{v}}_{k+\frac{3}{2},2} + \widehat{\boldsymbol{\Lambda}}^{\mathbf{u}}_{k+1,3} + \widehat{\boldsymbol{\Lambda}}^{\mathbf{v}}_{k+\frac{3}{2},4} + \widehat{\boldsymbol{\Lambda}}^{\mathbf{u}}_{k+1,5}$$

- Compute the (modified) internal stages for $\boldsymbol{\lambda}^{\mathbf{v}}$:

$$
\begin{aligned}
\widehat{\boldsymbol{\Lambda}}^{\mathbf{v}}_{k+\frac{1}{2},5} &= \frac{\partial \mathbf{f}^{\mathbf{v}}\left(\mathbf{u}_k + \frac{\tau}{2}\gamma \mathbf{V}^o_{k+\frac{1}{2},4}\right)^\top}{\partial \mathbf{u}} \left(\boldsymbol{\Theta}^{\mathbf{v}}_{k+\frac{1}{2},5} + \tau\, b_5 \boldsymbol{\lambda}^{\mathbf{v}}_{k+\frac{1}{2}}\right) \\
\widehat{\boldsymbol{\Lambda}}^{\mathbf{v}}_{k+\frac{1}{2},4} &= \frac{\partial \mathbf{f}^{\mathbf{u}}\left(\mathbf{v}_{k-\frac{1}{2}} + \frac{\tau}{4}(2+\gamma) \mathbf{V}^o_{k+\frac{1}{2},1}\right)^\top}{\partial \mathbf{v}} \left(\boldsymbol{\Theta}^{\mathbf{v}}_{k+\frac{1}{2},4} + \frac{\tau}{2}\gamma\, \widehat{\boldsymbol{\Lambda}}^{\mathbf{v}}_{k+\frac{1}{2},5}\right) \\
\widehat{\boldsymbol{\Lambda}}^{\mathbf{v}}_{k+\frac{1}{2},3} &= \frac{\partial \mathbf{f}^{\mathbf{v}}\left(\mathbf{u}_k - \frac{\tau}{2}\gamma \mathbf{V}^o_{k+\frac{1}{2},2}\right)^\top}{\partial \mathbf{u}} \left(\boldsymbol{\Theta}^{\mathbf{v}}_{k+\frac{1}{2},3} + \tau\, b_5 \boldsymbol{\lambda}^{\mathbf{v}}_{k+\frac{1}{2}}\right) \\
\widehat{\boldsymbol{\Lambda}}^{\mathbf{v}}_{k+\frac{1}{2},2} &= \frac{\partial \mathbf{f}^{\mathbf{u}}\left(\mathbf{v}_{k-\frac{1}{2}} + \frac{\tau}{4}(2-\gamma) \mathbf{V}^o_{k+\frac{1}{2},1}\right)^\top}{\partial \mathbf{v}} \left(\boldsymbol{\Theta}^{\mathbf{v}}_{k+\frac{1}{2},2} - \frac{\tau}{2}\gamma\, \widehat{\boldsymbol{\Lambda}}^{\mathbf{v}}_{k+\frac{1}{2},3}\right) \\
\widehat{\boldsymbol{\Lambda}}^{\mathbf{v}}_{k+\frac{1}{2},1} &= \frac{\partial \mathbf{f}^{\mathbf{v}}\left(\mathbf{u}_k\right)^\top}{\partial \mathbf{u}} \left(\boldsymbol{\Theta}^{\mathbf{v}}_{k+\frac{1}{2},1} + \frac{\tau}{4}(2-\gamma)\, \widehat{\boldsymbol{\Lambda}}^{\mathbf{v}}_{k+\frac{1}{2},2} + \frac{\tau}{4}(2+\gamma)\, \widehat{\boldsymbol{\Lambda}}^{\mathbf{v}}_{k+\frac{1}{2},4} + \right. \\
&\qquad\qquad \left. + \tau\,(1-2\,b_5) \boldsymbol{\lambda}^{\mathbf{v}}_{k+\frac{1}{2}}\right).
\end{aligned}
$$

- Compute the update for $\boldsymbol{\lambda}^{\mathbf{u}}$:

$$
\boldsymbol{\lambda}^{\mathbf{u}}_k = \mathbf{q}^{\mathbf{u}}_k + \boldsymbol{\lambda}^{\mathbf{u}}_{k+1} + \widehat{\boldsymbol{\Lambda}}^{\mathbf{v}}_{k+\frac{1}{2},1} + \widehat{\boldsymbol{\Lambda}}^{\mathbf{u}}_{k+1,2} + \widehat{\boldsymbol{\Lambda}}^{\mathbf{v}}_{k+\frac{1}{2},3} + \widehat{\boldsymbol{\Lambda}}^{\mathbf{u}}_{k+1,4} + \widehat{\boldsymbol{\Lambda}}^{\mathbf{v}}_{k+\frac{1}{2},5}.
$$

- Compute the (modified) internal stages for $\boldsymbol{\lambda}^{\mathbf{u}}$:

$$
\begin{aligned}
\widehat{\boldsymbol{\Lambda}}^{\mathbf{u}}_{k,5} &= \frac{\partial \mathbf{f}^{\mathbf{u}}\left(\mathbf{v}_{k-\frac{1}{2}} + \frac{\tau}{2}\gamma \mathbf{U}^o_{k,4}\right)^\top}{\partial \mathbf{v}} \left(\boldsymbol{\Theta}^{\mathbf{u}}_{k,5} + \tau\, b_5 \boldsymbol{\lambda}^{\mathbf{u}}_k\right) \\
\widehat{\boldsymbol{\Lambda}}^{\mathbf{u}}_{k,4} &= \frac{\partial \mathbf{f}^{\mathbf{v}}\left(\mathbf{u}_{k-1} + \frac{\tau}{4}(2+\gamma) \mathbf{U}^o_{k,1}\right)^\top}{\partial \mathbf{u}} \left(\boldsymbol{\Theta}^{\mathbf{u}}_{k,4} + \frac{\tau}{2}\gamma\, \widehat{\boldsymbol{\Lambda}}^{\mathbf{u}}_{k,5}\right) \\
\widehat{\boldsymbol{\Lambda}}^{\mathbf{u}}_{k,3} &= \frac{\partial \mathbf{f}^{\mathbf{u}}\left(\mathbf{v}_{k-\frac{1}{2}} - \frac{\tau}{2}\gamma \mathbf{U}^o_{k,2}\right)^\top}{\partial \mathbf{v}} \left(\boldsymbol{\Theta}^{\mathbf{u}}_{k,3} + \tau\, b_5 \boldsymbol{\lambda}^{\mathbf{u}}_k\right) \\
\widehat{\boldsymbol{\Lambda}}^{\mathbf{u}}_{k,2} &= \frac{\partial \mathbf{f}^{\mathbf{v}}\left(\mathbf{u}_{k-1} + \frac{\tau}{4}(2-\gamma) \mathbf{U}^o_{k,1}\right)^\top}{\partial \mathbf{u}} \left(\boldsymbol{\Theta}^{\mathbf{u}}_{k,2} - \frac{\tau}{2}\gamma\, \widehat{\boldsymbol{\Lambda}}^{\mathbf{u}}_{k,3}\right) \\
\widehat{\boldsymbol{\Lambda}}^{\mathbf{u}}_{k,1} &= \frac{\partial \mathbf{f}^{\mathbf{u}}\left(\mathbf{v}_{k-\frac{1}{2}}\right)^\top}{\partial \mathbf{v}} \left(\boldsymbol{\Theta}^{\mathbf{u}}_{k,1} + \frac{\tau}{4}(2-\gamma)\, \widehat{\boldsymbol{\Lambda}}^{\mathbf{u}}_{k,2} + \frac{\tau}{4}(2+\gamma)\, \widehat{\boldsymbol{\Lambda}}^{\mathbf{u}}_{k,4} + \tau\,(1-2\,b_5) \boldsymbol{\lambda}^{\mathbf{u}}_k\right).
\end{aligned}
$$

Letting $b_5 = 1/24$ (and therefore $\gamma = 2$), the modified internal stages for $\boldsymbol{\lambda}^{\mathbf{v}}$ are

$$
\begin{aligned}
\widehat{\boldsymbol{\Lambda}}^{\mathbf{v}}_{k+\frac{1}{2},5} &= \frac{\partial \mathbf{f}^{\mathbf{v}}\left(\mathbf{u}_k + \tau \mathbf{V}^o_{k+\frac{1}{2},4}\right)}{\partial \mathbf{u}}^{\top} \left(\boldsymbol{\Theta}^{\mathbf{v}}_{k+\frac{1}{2},5} + \frac{\tau}{24}\boldsymbol{\lambda}^{\mathbf{v}}_{k+\frac{1}{2}}\right) \\
\widehat{\boldsymbol{\Lambda}}^{\mathbf{v}}_{k+\frac{1}{2},4} &= \frac{\partial \mathbf{f}^{\mathbf{u}}\left(\mathbf{v}_{k-\frac{1}{2}} + \tau \mathbf{V}^o_{k+\frac{1}{2},1}\right)}{\partial \mathbf{v}}^{\top} \left(\boldsymbol{\Theta}^{\mathbf{v}}_{k+\frac{1}{2},4} + \tau \widehat{\boldsymbol{\Lambda}}^{\mathbf{v}}_{k+\frac{1}{2},5}\right) \\
\widehat{\boldsymbol{\Lambda}}^{\mathbf{v}}_{k+\frac{1}{2},3} &= \frac{\partial \mathbf{f}^{\mathbf{v}}\left(\mathbf{u}_k - \tau \mathbf{V}^o_{k+\frac{1}{2},2}\right)}{\partial \mathbf{u}}^{\top} \left(\boldsymbol{\Theta}^{\mathbf{v}}_{k+\frac{1}{2},3} + \frac{\tau}{24}\boldsymbol{\lambda}^{\mathbf{v}}_{k+\frac{1}{2}}\right) \\
\widehat{\boldsymbol{\Lambda}}^{\mathbf{v}}_{k+\frac{1}{2},2} &= \frac{\partial \mathbf{f}^{\mathbf{u}}\left(\mathbf{v}_{k-\frac{1}{2}}\right)}{\partial \mathbf{v}}^{\top} \left(\boldsymbol{\Theta}^{\mathbf{v}}_{k+\frac{1}{2},2} - \tau \widehat{\boldsymbol{\Lambda}}^{\mathbf{v}}_{k+\frac{1}{2},3}\right) \\
\widehat{\boldsymbol{\Lambda}}^{\mathbf{v}}_{k+\frac{1}{2},1} &= \frac{\partial \mathbf{f}^{\mathbf{v}}\left(\mathbf{u}_k\right)}{\partial \mathbf{u}}^{\top} \left(\boldsymbol{\Theta}^{\mathbf{v}}_{k+\frac{1}{2},1} + \tau \widehat{\boldsymbol{\Lambda}}^{\mathbf{v}}_{k+\frac{1}{2},4} + \frac{11\tau}{12}\boldsymbol{\lambda}^{\mathbf{v}}_{k+\frac{1}{2}}\right).
\end{aligned}
$$

and the modified internal stages for $\boldsymbol{\lambda}^{\mathbf{u}}$ are

$$
\begin{aligned}
\widehat{\boldsymbol{\Lambda}}^{\mathbf{u}}_{k,5} &= \frac{\partial \mathbf{f}^{\mathbf{u}}\left(\mathbf{v}_{k-\frac{1}{2}} + \tau \mathbf{U}^o_{k,4}\right)}{\partial \mathbf{v}}^{\top} \left(\boldsymbol{\Theta}^{\mathbf{u}}_{k,5} + \frac{\tau}{24}\boldsymbol{\lambda}^{\mathbf{u}}_{k}\right) \\
\widehat{\boldsymbol{\Lambda}}^{\mathbf{u}}_{k,4} &= \frac{\partial \mathbf{f}^{\mathbf{v}}\left(\mathbf{u}_{k-1} + \tau \mathbf{U}^o_{k,1}\right)}{\partial \mathbf{u}}^{\top} \left(\boldsymbol{\Theta}^{\mathbf{u}}_{k,4} + \tau \widehat{\boldsymbol{\Lambda}}^{\mathbf{u}}_{k,5}\right) \\
\widehat{\boldsymbol{\Lambda}}^{\mathbf{u}}_{k,3} &= \frac{\partial \mathbf{f}^{\mathbf{u}}\left(\mathbf{v}_{k-\frac{1}{2}} - \tau \mathbf{U}^o_{k,2}\right)}{\partial \mathbf{v}}^{\top} \left(\boldsymbol{\Theta}^{\mathbf{u}}_{k,3} + \frac{\tau}{24}\boldsymbol{\lambda}^{\mathbf{u}}_{k}\right) \\
\widehat{\boldsymbol{\Lambda}}^{\mathbf{u}}_{k,2} &= \frac{\partial \mathbf{f}^{\mathbf{v}}\left(\mathbf{u}_{k-1}\right)}{\partial \mathbf{u}}^{\top} \left(\boldsymbol{\Theta}^{\mathbf{u}}_{k,2} - \tau \widehat{\boldsymbol{\Lambda}}^{\mathbf{u}}_{k,3}\right) \\
\widehat{\boldsymbol{\Lambda}}^{\mathbf{u}}_{k,1} &= \frac{\partial \mathbf{f}^{\mathbf{u}}\left(\mathbf{v}_{k-\frac{1}{2}}\right)}{\partial \mathbf{v}}^{\top} \left(\boldsymbol{\Theta}^{\mathbf{u}}_{k,1} + \tau \widehat{\boldsymbol{\Lambda}}^{\mathbf{u}}_{k,4} + \frac{11\tau}{12}\boldsymbol{\lambda}^{\mathbf{u}}_{k}\right).
\end{aligned}
$$

6.4 Exponential Time-Differencing Methods

Lastly, we discuss the adjoint ETDRK method, and apply it to Krogstad's scheme as an illustration.

6.4.1 Exponential Runge-Kutta Methods

We let $\overline{\boldsymbol{\lambda}} = \begin{bmatrix} \widehat{\boldsymbol{\lambda}}_1^\top & \cdots & \widehat{\boldsymbol{\lambda}}_K^\top \end{bmatrix}^\top$ represent the adjoint solution, with $\widehat{\boldsymbol{\lambda}}_k = \begin{bmatrix} \boldsymbol{\Lambda}_k^\top & \boldsymbol{\lambda}_k^\top \end{bmatrix}^\top$. The adjoint source $\overline{\boldsymbol{\theta}}$ is defined similarly, with internal stages $\boldsymbol{\Theta}_{k,\sigma}$.

The linearization of the time-stepping equation, $\dfrac{\partial \mathbf{t}}{\partial \overline{\mathbf{y}}}$, was derived in Section A.4.1. Its transpose is (see (A.234))

$$\frac{\partial \mathbf{t}}{\partial \overline{\mathbf{y}}}^\top = \begin{bmatrix} \mathbf{A}_1^\top & -\mathbf{B}_1 & & & & & \\ & \mathbf{I}_N & \mathbf{C}_2^\top & \mathbf{D}_2^\top & & & \\ & & \mathbf{A}_2^\top & -\mathbf{B}_2 & & & \\ & & & \ddots & \ddots & \ddots & \\ & & & & \mathbf{I}_N & \mathbf{C}_K^\top & \mathbf{D}_K^\top \\ & & & & & \mathbf{A}_K^\top & -\mathbf{B}_K \\ & & & & & & \mathbf{I}_N \end{bmatrix}.$$

$\mathbf{A}_k$, $\mathbf{C}_k$ and $\mathbf{D}_k$ were defined in (A.229):

$$\mathbf{A}_k = \mathbf{I}_{sN} - \tau_k \frac{\partial \mathbf{N}_k}{\partial \overline{\mathbf{y}}} \mathbf{A}_{k-1}^s \qquad \mathbf{C}_k = -\frac{\partial \mathbf{N}_k}{\partial \overline{\mathbf{y}}_{k-1}},$$
$$\mathbf{D}_k = -\frac{\partial \left(e^{\tau_k \mathbf{L}_{k-1}} \overline{\mathbf{y}}_{k-1} \right)}{\partial \overline{\mathbf{y}}_{k-1}} - \frac{\partial \left(\mathbf{B}_k^\top \overline{\mathbf{y}}_k \right)}{\partial \overline{\mathbf{y}}_{k-1}}.$$

One step of the solution procedure is

$$\boldsymbol{\lambda}_k = \boldsymbol{\theta}_k - \mathbf{C}_{k+1}^\top \boldsymbol{\Lambda}_{k+1} - \mathbf{D}_{k+1}^\top \boldsymbol{\lambda}_{k+1} \tag{6.8a}$$

followed by the internal stages

$$\mathbf{A}_k^\top \boldsymbol{\Lambda}_k = \boldsymbol{\Theta}_k + \mathbf{B}_k \boldsymbol{\lambda}_k, \tag{6.8b}$$

where $\boldsymbol{\lambda}_{K+1} = \mathbf{0}_{N\times 1}$ and $\boldsymbol{\Lambda}_{K+1} = \mathbf{0}_{sN\times 1}$.

Using the general formulas for $\mathbf{A}_k$, $\mathbf{C}_k$ and $\mathbf{D}_k$ gives

$$\boldsymbol{\lambda}_k = \boldsymbol{\theta}_k + \left(\frac{\partial \mathbf{N}_{k+1}}{\partial \mathbf{y}}\right)^\top \boldsymbol{\Lambda}_{k+1} + \left(\frac{\partial \left(e^{\tau_k \mathbf{L}_k}\mathbf{y}_k\right)}{\partial \mathbf{y}_k}^\top + \frac{\partial \left(\mathbf{B}_{k+1}^\top \mathbf{Y}_{k+1}\right)}{\partial \mathbf{y}_k}^\top\right) \boldsymbol{\lambda}_{k+1} \tag{6.9a}$$

with internal stages

$$\boldsymbol{\Lambda}_k = \boldsymbol{\Theta}_k + \tau_{k-1}\left(\mathbf{A}_{k-1}^s\right)^\top \left(\frac{\partial \mathbf{N}_k}{\partial \mathbf{y}}\right)^\top \boldsymbol{\Lambda}_k + \mathbf{B}_k \boldsymbol{\lambda}_k. \tag{6.9b}$$

The detailed solution procedure is summarized in Algorithm 6.7.

Algorithm 6.7. The Adjoint Exponential Runge-Kutta Method

Let $\boldsymbol{\lambda}_{K+1} = \mathbf{0}_{N\times 1}$ *and* $\boldsymbol{\Lambda}_{K+1} = \mathbf{0}_{sN\times 1}$. *For* $k = K, \cdots, 1$*:*

- *Compute*

$$\begin{aligned}
\boldsymbol{\lambda}_k = \boldsymbol{\theta}_k &+ e^{\tau_{k+1}\mathbf{L}_k^\top}\boldsymbol{\lambda}_{k+1} + \sum_{\sigma=1}^{s} e^{c_\sigma \tau_{k+1}\mathbf{L}_k^\top}\left(\frac{\partial \mathbf{n}_{k,\sigma}}{\partial \mathbf{y}}\right)^\top \boldsymbol{\Lambda}_{k+1,\sigma} + \\
&+ \left\{\sum_{i=1}^{s}\left[\frac{\partial \mathbf{n}_{k,i}}{\partial \mathbf{y}_k}^\top + \left(\frac{\partial e^{c_i \tau_{k+1}\mathbf{L}_k(\mathbf{y}_k)}\mathbf{y}_k^{\text{fixed}}}{\partial \mathbf{y}_k}^\top + \right.\right.\right. \\
&+ \left.\left.\tau_{k+1}\sum_{j=1}^{i-1}\frac{\partial \left(a_{ij}(\tau_{k+1}\mathbf{L}_k(\mathbf{y}_k))\mathbf{Y}_{k+1,j}\right)}{\partial \mathbf{y}_k}^\top\right)\frac{\partial \mathbf{n}_{k,i}}{\partial \mathbf{y}}^\top\right]\boldsymbol{\Lambda}_{k+1,i} + \\
&+ \left.\left(\frac{\partial \left(e^{\tau_{k+1}\mathbf{L}_k(\mathbf{y}_k)}\mathbf{y}_k^{\text{fixed}}\right)}{\partial \mathbf{y}_k}^\top + \tau_{k+1}\sum_{i=1}^{s}\frac{\partial \left(b_i(\tau_{k+1}\mathbf{L}_k(\mathbf{y}_k))\mathbf{Y}_{k+1,i}\right)}{\partial \mathbf{y}_k}^\top\right)\boldsymbol{\lambda}_{k+1}\right\},
\end{aligned}$$

  where the terms in braces are ignored if $\mathbf{L}_k$ *does not depend on* $\mathbf{y}_k$.

- *For* $\sigma = s, \cdots, 1$,

$$\boldsymbol{\Lambda}_{k,\sigma} = \boldsymbol{\Theta}_{k,\sigma} + \tau_k b_\sigma \left(\tau_k \mathbf{L}_{k-1}\right)^\top \boldsymbol{\lambda}_k + \tau_k \sum_{j=\sigma+1}^{s} a_{j\sigma}\left(\tau_k \mathbf{L}_{k-1}^\top\right)\left(\frac{\partial \mathbf{n}_{k-1,j}}{\partial \mathbf{y}}\right)^\top \boldsymbol{\Lambda}_{k,j}.$$

The derivatives in the braces in (6.10a) *are computed as discussed in Section 4.4.2.*

Example

We continue the example from Section 5.4 and apply the above algorithm to Krogstad's scheme. Recall the matrix coefficients $a_{\sigma j}$ and b_σ given in (2.63). The adjoint scheme is given in Algorithm 6.8.

Algorithm 6.8. The Adjoint Krogstad Scheme

Let $\widehat{\mathbf{\Lambda}}_{k,\sigma} = \left(\frac{\partial \mathbf{n}_{k-1,\sigma}}{\partial \mathbf{y}}\right)^\top \mathbf{\Lambda}_{k,\sigma}$. *For* $k = K, \ldots, 1$*:*

- *In this step, let* $\varphi_\ell = \varphi_\ell(\tau_{k+1}\mathbf{L}_k(\mathbf{y}_k))$ *and* $\varphi_{\ell,i} = \varphi_\ell(c_\sigma \tau_{k+1}\mathbf{L}_k(\mathbf{y}_k))$. *Compute*

$$
\begin{aligned}
\boldsymbol{\lambda}_k = \boldsymbol{\theta}_k &+ e^{\tau_{k+1}\mathbf{L}_k^\top}\left(\boldsymbol{\lambda}_{k+1} + \widehat{\mathbf{\Lambda}}_{k+1,4}\right) + e^{\frac{1}{2}\tau_{k+1}\mathbf{L}_k^\top}\left(\widehat{\mathbf{\Lambda}}_{k+1,2} + \widehat{\mathbf{\Lambda}}_{k+1,3}\right) + \\
&+ \widehat{\mathbf{\Lambda}}_{k+1,1} + \left\{\sum_{\sigma=1}^{4} \frac{\partial \mathbf{n}_{k,i}}{\partial \mathbf{y}_k}^\top \mathbf{\Lambda}_{k+1,i} + \frac{\partial e^{\tau_{k+1}\mathbf{L}_k(\mathbf{y}_k)}\mathbf{y}_k^{\text{fixed}}}{\partial \mathbf{y}_k}^\top \widehat{\mathbf{\Lambda}}_{k+1,4} + + \right. \\
&+ \frac{\partial e^{\frac{1}{2}\tau_{k+1}\mathbf{L}_k(\mathbf{y}_k)}\mathbf{y}_k^{\text{fixed}}}{\partial \mathbf{y}_k}^\top \left(\widehat{\mathbf{\Lambda}}_{k+1,2} + \widehat{\mathbf{\Lambda}}_{k+1,3}\right) + \\
&+ \frac{\partial \left(e^{\tau_{k+1}\mathbf{L}_k(\mathbf{y}_k)}\mathbf{y}_k^{\text{fixed}}\right)}{\partial \mathbf{y}_k}^\top + \frac{\tau_{k+1}}{2}\frac{\partial(\varphi_{1,2}\mathbf{Y}_{k+1,1})}{\partial \mathbf{y}_k}^\top \widehat{\mathbf{\Lambda}}_{k+1,2} + \\
&+ \tau_k \left(\left(\frac{1}{2}\frac{\partial(\varphi_{1,3}\mathbf{Y}_{k+1,1})}{\partial \mathbf{y}_k}^\top + \frac{\partial(\varphi_{2,3}(\mathbf{Y}_{k+1,2} - \mathbf{Y}_{k+1,1}))}{\partial \mathbf{y}_k}^\top\right) \widehat{\mathbf{\Lambda}}_{k+1,3} + \right. \\
&+ \left(\frac{\partial(\varphi_{1,4}\mathbf{Y}_{k+1,1})}{\partial \mathbf{y}_k}^\top + 2\frac{\partial(\varphi_{2,4}(\mathbf{Y}_{k+1,3} - \mathbf{Y}_{k+1,1}))}{\partial \mathbf{y}_k}^\top\right) \widehat{\mathbf{\Lambda}}_{k+1,4} + \\
&+ \left(\frac{\partial(\varphi_1\mathbf{Y}_{k+1,1})}{\partial \mathbf{y}_k}^\top - \frac{\partial(\varphi_2\left(3\mathbf{Y}_{k+1,1} - 2\mathbf{Y}_{k+1,2} - 2\mathbf{Y}_{k+1,3} + \mathbf{Y}_{k+1,4}\right))}{\partial \mathbf{y}_k}^\top + \right. \\
&\left.\left.\left. + \; 4\frac{\partial(\varphi_3(\mathbf{Y}_{k+1,1} - \mathbf{Y}_{k+1,2} - \mathbf{Y}_{k+1,3} + \mathbf{Y}_{k+1,4}))}{\partial \mathbf{y}_k}^\top\right) \boldsymbol{\lambda}_{k+1}\right)\right\}
\end{aligned}
$$

  with $\boldsymbol{\lambda}_{K+1} = \widehat{\mathbf{\Lambda}}_{K+1,i} = \mathbf{0}_{N\times 1}$. *Ignore the terms in braces if* $\mathbf{L}_k$ *is independent of* $\mathbf{y}_k$.

- *Now let* $\varphi_\ell = \varphi_\ell(\tau_k\mathbf{L}_{k-1}(\mathbf{y}_{k-1}))$ *and* $\varphi_{\ell,i} = \varphi_\ell(c_\sigma\tau_k\mathbf{L}_{k-1}(\mathbf{y}_{k-1}))$. *The internal*

stages are computed as follows:

$$
\begin{aligned}
\boldsymbol{\Lambda}_{k,4} &= \tau_{k-1} b_4^\top \boldsymbol{\lambda}_k = \tau_k \left(4\varphi_3^\top - \varphi_2^\top\right) \boldsymbol{\lambda}_k \\
\boldsymbol{\Lambda}_{k,3} &= \tau_k b_3^\top \boldsymbol{\lambda}_k + \tau_k a_{4,3}^\top \widehat{\boldsymbol{\Lambda}}_{k,4} = \tau_k \left(2\varphi_2^\top - 4\varphi_3^\top\right) \boldsymbol{\lambda}_k + \tau_k 2\varphi_{2,4}^\top \widehat{\boldsymbol{\Lambda}}_{k,4} \\
\boldsymbol{\Lambda}_{k,2} &= \tau_k b_2^\top \boldsymbol{\lambda}_k + \tau_k a_{4,2}^\top \widehat{\boldsymbol{\Lambda}}_{k,4} + \tau_k a_{3,2}^\top \widehat{\boldsymbol{\Lambda}}_{k,3} \\
&= \tau_k \left(2\varphi_2^\top - 4\varphi_3^\top\right) \boldsymbol{\lambda}_k + \tau_k \varphi_{2,3}^\top \widehat{\boldsymbol{\Lambda}}_{k,3} \qquad (6.11) \\
\boldsymbol{\Lambda}_{k,1} &= \tau_k b_1^\top \boldsymbol{\lambda}_k + \tau_k a_{4,1}^\top \widehat{\boldsymbol{\Lambda}}_{k,4} + \tau_k a_{3,1}^\top \widehat{\boldsymbol{\Lambda}}_{k,3} + \tau_k a_{2,1}^\top \widehat{\boldsymbol{\Lambda}}_{k,2} \\
&= \tau_k \left(\varphi_1^\top - 3\varphi_2^\top + 4\varphi_3^\top\right) \boldsymbol{\lambda}_k + \tau_k \left(\varphi_{1,4}^\top - 2\varphi_{2,4}^\top\right) \widehat{\boldsymbol{\Lambda}}_{k,4} + \\
&\qquad + \frac{\tau_k}{2} \left(\varphi_{1,3}^\top - 2\varphi_{2,3}^\top\right) \widehat{\boldsymbol{\Lambda}}_{k,3} + \frac{\tau_k}{2} \varphi_{1,2}^\top \widehat{\boldsymbol{\Lambda}}_{k,2}.
\end{aligned}
$$

Other schemes can be handled similarly. We note that the procedure gives ample opportunity for parallelization and precomputing quantities, for instance when computing the products of $\boldsymbol{\lambda}_k$ with $\varphi_\ell^\top$ or the products of $\widehat{\boldsymbol{\Lambda}}_{k,i}$ with $\varphi_{\ell,i}^\top$.

6.5 Stability, Convergence, and Order of Accuracy for Linear Problems

Let us briefly discuss some properties of interest of the adjoint RK-type methods if the PDE happens to be linear. The consistency and order of accuracy of adjoint time-stepping methods corresponding to regular RK methods for nonlinear problems have been investigated by Sandu in [100], and an analysis of the order of accuracy of these adjoint RK methods (up to order 4) in the context of optimal control was given by Hager in [52]. The approach taken in these papers is using Taylor expansions.

However, we find that for linear problems the simple argument given in the theorem below, which is relevant to not only regular RK methods, but also IMEX, staggered RK and ETDRK methods, should suffice.

Theorem 6.9. *For linear PDEs, the adjoint RK-type method inherits the convergence and stability properties from the corresponding forward method. It also has the same*

order of accuracy.

Proof. A single step of any RK method applied to a homogeneous problem can be written as

$$\mathbf{y}^k = \mathbf{\Psi}\left(\tau_k \mathbf{L}\right)\mathbf{y}^{k-1}, \tag{6.12}$$

where $\mathbf{\Psi}$ is the *transfer operator* that represents the propagation from one time level to the next. For instance, after some manipulation we obtain for explicit, regular RK methods

$$\mathbf{\Psi}\left(\tau_k \mathbf{L}\right) = \mathbf{I} + \sum_{\sigma=1}^{s} b_\sigma \mathbf{C}_{k,\sigma}\left(\tau_k \mathbf{L}\right)$$

with

$$\begin{aligned}\mathbf{C}_{k,\sigma}\left(\tau_k \mathbf{L}\right) = \tau_k \mathbf{L}\Bigg(&\mathbf{I}_N + \tau_k \sum_{\sigma_1=1}^{\sigma-1} a_{\sigma\sigma_1}\mathbf{L} + \tau_k^2 \sum_{\sigma_1=2}^{\sigma-1}\sum_{\sigma_2=1}^{\sigma_1-1} a_{\sigma\sigma_1}a_{\sigma_1\sigma_2}\mathbf{L}^2 + \\ &+\tau_k^3 \sum_{\sigma_1=3}^{\sigma-1}\sum_{\sigma_2=2}^{\sigma_1-1}\sum_{\sigma_3=1}^{\sigma_2-1} a_{\sigma\sigma_1}a_{\sigma_1\sigma_2}a_{\sigma_2\sigma_3}\mathbf{L}^3 + \cdots + \tau_k^{\sigma-1}\prod_{i=2}^{\sigma}(a_{i,i-1}\mathbf{L})\Bigg).\end{aligned}$$

Slightly different representations of $\mathbf{\Psi}\left(\tau_k \mathbf{L}\right)$ for regular RK methods can be found in the literature. It is possible to find expressions similar to this for IMEX RK and StagRK methods, although we do not give them here.

Letting $\mathbf{\Psi}_k = \mathbf{\Psi}\left(\tau_k \mathbf{L}\right)$, the homogeneous forward problem can be written as

$$\begin{bmatrix} \mathbf{I}_N & & & & \\ -\mathbf{\Psi}_2 & \mathbf{I}_N & & & \\ & -\mathbf{\Psi}_3 & \mathbf{I}_N & & \\ & & \ddots & \ddots & \\ & & & -\mathbf{\Psi}_K & \mathbf{I}_N \end{bmatrix} \begin{bmatrix} \mathbf{y}_1 \\ \mathbf{y}_2 \\ \mathbf{y}_3 \\ \vdots \\ \mathbf{y}_K \end{bmatrix} = \begin{bmatrix} \mathbf{\Psi}_1\mathbf{y}_0 \\ \mathbf{0} \\ \mathbf{0} \\ \vdots \\ \mathbf{0} \end{bmatrix},$$

and therefore the homogeneous adjoint system is

$$\begin{bmatrix} \mathbf{I}_N & -\mathbf{\Psi}_2^\top & & & \\ & \mathbf{I}_N & -\mathbf{\Psi}_3^\top & & \\ & & \ddots & \ddots & \\ & & & \mathbf{I}_N & -\mathbf{\Psi}_K^\top \\ & & & & \mathbf{I}_N \end{bmatrix} \begin{bmatrix} \boldsymbol{\lambda}_1 \\ \boldsymbol{\lambda}_2 \\ \vdots \\ \boldsymbol{\lambda}_{K-1} \\ \boldsymbol{\lambda}_K \end{bmatrix} = \begin{bmatrix} \mathbf{0} \\ \mathbf{0} \\ \vdots \\ \mathbf{0} \\ \mathbf{\Psi}_{K+1}\boldsymbol{\lambda}_{K+1} \end{bmatrix}.$$

Hence $\boldsymbol{\lambda}_{k-1} = \mathbf{\Psi}_k^\top(\mathbf{L})\,\boldsymbol{\lambda}_k$. It is fairly straightforward, albeit tedious, to show that this condition holds for the adjoint RK method we found in earlier in this chapter. Note that $\mathbf{\Psi}_k^\top(\mathbf{L}) = \mathbf{\Psi}_k\left(\mathbf{L}^\top\right)$.

By the Lax equivalence theorem, a consistent linear method in the form (6.12) is convergent if and only if it is Lax-Richtmyer stable. Letting $\tau_k = \tau$ for simplicity, Lax-Richtmyer stability means that for each time T, there is a constant $C_T > 0$ such that

$$\left\|\mathbf{\Psi}\left(\tau\mathbf{L}\right)^K\right\| \leq C_T$$

for all $\tau > 0$ and integers K for which $\tau K < T$. We know that this must hold for $\mathbf{\Psi}\left(\tau\mathbf{L}\right)$ (because the forward method is stable), and hence the adjoint time-stepping method is Lax-Richtmyer stable as well since matrix transposition is invariant under, e.g., the 2-norm.

Now consider a continuous forward solution $\mathbf{y}(t)$ and a continuous adjoint solution $\boldsymbol{\lambda}(t)$. An arbitrary RK method applied to the scalar test problem $\dot{\mathbf{y}}(t) = \mu\,\mathbf{y}(t)$, where μ is some scalar (it represents one of the eigenvalues of $\mathbf{L}$). Then we have that over a single time step $\mathbf{y}_k = \mathbf{\Psi}(z)\,\mathbf{y}_{k-1}$, where $z = \tau_k\mu$, and if the method is pth order accurate we must have that

$$\mathbf{\Psi}(z) - e^z = O(z^{p+1}).$$

Since the same $\mathbf{\Psi}$ applies to the corresponding scalar adjoint RK method, it follows

that for a single time step the same condition must hold for the solution of the continuous adjoint scalar test problem, and therefore the adjoint RK method must have the same order of accuracy as the forward method. □

This result also applies to IMEX RK and StagRK methods, if the PDE is linear. Nonlinear PDEs require a much more cumbersome approach involving Taylor expansions of $\mathbf{\Psi}$ and lie outside the scope of this thesis, but, as mentioned above, see [52, 100].

We will not discuss here the order reduction phenomenon [13, 88, 108] that can happen close to the boundary when RK methods are applied to nonhomogeneous problems, but it is possible that the absence of source terms in the internal stages of the adjoint RK methods would lessen this effect. At the least it seems reasonable to assume that the reduction of order suffered by the adjoint RK method would not be worse than that of the RK method, but a more careful analysis needs to be done to support this statement.

Chapter 7

Data Misfit Function Examples

An important application of large-scale distributed parameter estimation is that of seismic imaging. Seismic waves, from either natural or man-made sources, are measured at *seismic receivers*, with each receiver generating a *trace* that displays the waveform passing through that point in space. The observed data is generated using a number of receivers, and from these measurements one wants to infer characteristics about the structure of the subsurface. The problem is commonly referred to as *full waveform inversion* (FWI) if it is treated as a parameter estimation problem and numerical opimization procedures are used; see [36] for a good introduction to the subject.

There are two main approaches to obtaining seismic data, either *reflection seismology*, where seismic waves are generated using sources that are usually close to the surface, and their reflections from the interfaces between different strata in the subsurface are measured using receivers that are also usually located close to the surface. Alternatively, *seismic tomography* measures seismic waves that have propagated through the subsurface without having been reflected off of interfaces. Usually both the seismic sources and receivers are located underground, with the sources located in one borehole and the receivers in a different one, and the properties of the subsurface between these two boreholes is investigated in this way.

Over the years different misfit functions have been developed to help improve the quality of the recovered model parameters[2]. In this chapter we present the derivations of the derivatives of some of these misfits, namely the least-squares amplitude misfit,

[2]Recall that the misfit function gives a scalar value that measures the discrepancy between the simulated data $\mathbf{d} = \mathbf{d}(\mathbf{p})$ and the actual observed data $\mathbf{d}^{\text{obs}}$.

the cross-correlation time shift misfit and the more recent interferometric misfit, the latter being the main motivation for the work in this chapter. It is not yet well-known, but could potentially become more important in the future. As before, we will use $\mathrm{M} = \mathrm{M}(\mathbf{d}, \mathbf{d}^{\mathrm{obs}})$ to denote the misfit function. While all of these misfits can be used for reflection seismology, they are more useful, and used more often, in the context of seismic tomography.

The gradient $\nabla_{\mathbf{p}}\mathrm{M}$ requires $\nabla_{\mathbf{d}}\mathrm{M}$ and the Hessian $\dfrac{\partial^2 \mathrm{M}}{\partial \mathbf{p} \partial \mathbf{p}}$, as well as its Gauss-Newton (3.16) and Levenberg-Marquardt (3.17) approximations, additionally require $\dfrac{\partial^2 \mathrm{M}}{\partial \mathbf{d} \, \partial \mathbf{d}}$. We will compute both of these derivatives for each of the misfit functions under consideration here.

We let $\mathbf{d}_i$ be the simulated measurements taken at the ith of N_R receivers, so that the simulated data is $\mathbf{d} = \begin{bmatrix} \mathbf{d}_1^\top & \cdots & \mathbf{d}_{N_R}^\top \end{bmatrix}^\top$. Similarly, the actual observations at the ith receiver are given by $\mathbf{d}_i^{\mathrm{obs}}$. These measurements are naturally discrete in practice, but in what follows it will be beneficial to at first treat them as continuous in order to mathematically justify the misfit function under consideration.

Not discussed at length is the least-squares (or ℓ_2) waveform difference misfit, or simply least-squares misfit, which is by far the most commonly used misfit because of its simplicity, although often it may not necessarily be the best choice, especially in seismic imaging. The misfit is given by

$$\mathrm{M} = \frac{1}{2} \left\| \mathbf{W} \left(\mathbf{d} - \mathbf{d}^{\mathrm{obs}} \right) \right\|^2 = \frac{1}{2} \left(\mathbf{d} - \mathbf{d}^{\mathrm{obs}} \right)^\top \mathbf{W}^\top \mathbf{W} \left(\mathbf{d} - \mathbf{d}^{\mathrm{obs}} \right),$$

where $\mathbf{W}$ is a weighting matrix, and its derivatives are

$$\nabla_{\mathbf{d}}\mathrm{M} = \mathbf{W}^\top \mathbf{W} \left(\mathbf{d} - \mathbf{d}^{\mathrm{obs}} \right)$$

and

$$\frac{\partial^2 \mathrm{M}}{\partial \mathbf{d} \, \partial \mathbf{d}} = \mathbf{W}^\top \mathbf{W}.$$

For the least-squares amplitude misfit and cross-correlation time shift misfit, see [36] and the references therein for both background information and their gradients in a continuous setting. Our contribution here is to derive the gradient of M with respect to $\mathbf{d}$ in a discrete setting (as much as possible), and to derive the expression for the product of the Hessian $\frac{\partial^2 \mathrm{M}}{\partial \mathbf{d} \partial \mathbf{d}}$ with some arbitrary vector $\mathbf{v}$ of length $N_{\mathbf{d}}$, the latter not having been done before as far as we know. The derivatives of the interferomentric are also novel to the best of our knowledge. For all three misfits we give algorithms for computing the gradient and the action of the Hessian.

A misfit that is not discussed in this thesis is the time-frequency misfit, which is more versatile than the least-squares amplitude and cross-correlation time shift misfits, but also signifcantly more complicated. See [36] for a discussion.

7.1 Least-Squares Amplitudes Misfit

The least squares amplitude measures the relative difference in total energy in the simulated waveform compared to the observed waveform[3]. It has been used in large-scale seismic tomography applications in for instance, [111, 112, 119]. Our discussion is based on [36], where the gradient of the M was computed in a continuous setting.

Letting $\mathcal{A}_i = \|\mathbf{d}\| = \sqrt{\mathbf{d}_i^\top \mathbf{d}_i}$ and $\mathcal{A}_i^{\mathrm{obs}} = \left\|\mathbf{d}_i^{\mathrm{obs}}\right\| = \sqrt{\mathbf{d}_i^{\mathrm{obs}\top} \mathbf{d}_i^{\mathrm{obs}}}$, then the misfit function that we want to minimize is

$$\mathrm{M} = \frac{1}{2} \sum_i \frac{\left(\mathcal{A}_i - \mathcal{A}_i^{\mathrm{obs}}\right)^2}{\left(\mathcal{A}_i^{\mathrm{obs}}\right)^2} = \frac{1}{2} \sum_i \frac{\left(\sqrt{\mathbf{d}_i^\top \mathbf{d}_i} - \sqrt{\mathbf{d}_i^{\mathrm{obs}\top} \mathbf{d}_i^{\mathrm{obs}}}\right)^2}{\mathbf{d}_i^{\mathrm{obs}\top} \mathbf{d}_i^{\mathrm{obs}}}. \tag{7.1}$$

Using the chain rule gives

$$\nabla_{\mathbf{d}_i} \mathrm{M} = \frac{\left(\sqrt{\mathbf{d}_i^\top \mathbf{d}_i} - \sqrt{\mathbf{d}_i^{\mathrm{obs}\top} \mathbf{d}_i^{\mathrm{obs}}}\right)}{\left(\mathbf{d}_i^{\mathrm{obs}\top} \mathbf{d}_i^{\mathrm{obs}}\right) \sqrt{\mathbf{d}_i^\top \mathbf{d}_i}} \mathbf{d}$$

[3]The "amplitude" refers to the fact that if just a single wave is observed then the misfit will measure the relative difference in the simulated and actual wave amplitudes.

$$
\begin{aligned}
&= \frac{1}{\mathbf{d}_i^{\mathrm{obs}\top}\mathbf{d}_i^{\mathrm{obs}}}\mathbf{d}_i - \frac{1}{\sqrt{\mathbf{d}_i^{\mathrm{obs}\top}\mathbf{d}_i^{\mathrm{obs}}}\sqrt{\mathbf{d}_i^{\top}\mathbf{d}_i}}\mathbf{d}_i \\
&= \frac{\mathcal{A}_i - \mathcal{A}_i^{\mathrm{obs}}}{\left(\mathcal{A}_i^{\mathrm{obs}}\right)^2 \mathcal{A}_i}\mathbf{d}_i, \qquad (7.2)
\end{aligned}
$$

so that

$$
\nabla_{\mathbf{d}}\mathrm{M} = \begin{bmatrix} \dfrac{\mathcal{A}_1 - \mathcal{A}_1^{\mathrm{obs}}}{\left(\mathcal{A}_1^{\mathrm{obs}}\right)^2 \mathcal{A}_1}\mathbf{d}_1 \\ \vdots \\ \dfrac{\mathcal{A}_{N_R} - \mathcal{A}_{N_R}^{\mathrm{obs}}}{\left(\mathcal{A}_{N_R}^{\mathrm{obs}}\right)^2 \mathcal{A}_{N_R}}\mathbf{d}_{N_R} \end{bmatrix}.
$$

The numerical implementation of this gradient is straightforward and is given in Algorithm 7.1.

Algorithm 7.1. Gradient of the Least-Squares Amplitude Misfit

For $i = 1, \cdots, N_R$,

- *If not already available, compute the norm of each observed and simulated trace.*
- *Set*

$$
\boldsymbol{\theta}_i = \frac{\mathcal{A}_i - \mathcal{A}_i^{\mathrm{obs}}}{\left(\mathcal{A}_i^{\mathrm{obs}}\right)^2 \mathcal{A}_i}\mathbf{d}_i.
$$

Finally, set

$$
\nabla_{\mathbf{d}}\mathrm{M} = \begin{bmatrix} \boldsymbol{\theta}_1^{\top} & \cdots & \boldsymbol{\theta}_{N_R}^{\top} \end{bmatrix}^{\top}.
$$

The Hessian is also easy to find, we have

$$
\frac{\partial^2 \mathrm{M}}{\partial \mathbf{d}\,\partial \mathbf{d}} = \begin{bmatrix} \dfrac{\partial^2 \mathrm{M}}{\partial \mathbf{d}_1\,\partial \mathbf{d}_1} & & \\ & \ddots & \\ & & \dfrac{\partial^2 \mathrm{M}}{\partial \mathbf{d}_{N_R}\,\partial \mathbf{d}_{N_R}} \end{bmatrix}
$$

with

$$\begin{aligned}\frac{\partial^2 \mathrm{M}}{\partial \mathbf{d}_i \partial \mathbf{d}_i} &= \frac{1}{\mathbf{d}_i^{\mathrm{obs}\top}\mathbf{d}_i^{\mathrm{obs}}}\mathbf{I} - \frac{1}{\sqrt{\mathbf{d}_i^{\mathrm{obs}\top}\mathbf{d}_i^{\mathrm{obs}}}\sqrt{\mathbf{d}_i^\top \mathbf{d}_i}}\mathbf{I} + \frac{1}{\sqrt{\mathbf{d}_i^{\mathrm{obs}\top}\mathbf{d}_i^{\mathrm{obs}}}\sqrt{\mathbf{d}_i^\top \mathbf{d}_i}\left(\mathbf{d}_i^\top \mathbf{d}_i\right)}\mathbf{d}_i\mathbf{d}_i^\top \\ &= \frac{\mathcal{A}_i - \mathcal{A}_i^{\mathrm{obs}}}{\left(\mathcal{A}_i^{\mathrm{obs}}\right)^2 \mathcal{A}_i}\mathbf{I} + \frac{1}{\mathcal{A}_i^{\mathrm{obs}}\mathcal{A}_i^3}\mathbf{d}_i\mathbf{d}_i^\top. \end{aligned} \tag{7.3}$$

The procedure for computing the product of the Hessian of M with some arbitrary vector $\mathbf{v}$ of length $N_{\mathbf{d}}$, defined analogously to $\mathbf{d}$, is given in Algorithm 7.2.

Algorithm 7.2. Product of the Hessian of the Least-Squares Amplitude Misfit

Procedure for computing the product of the Hessian $\dfrac{\partial^2 \mathrm{M}}{\partial \mathbf{d}\,\partial \mathbf{d}}$ *with an arbitrary vector* $\mathbf{v} = \begin{bmatrix}\mathbf{v}_1^\top & \cdots & \mathbf{v}_{N_R}^\top\end{bmatrix}^\top$, *where each* $\mathbf{v}_i$ *has length* $N_{\mathbf{d}_i}$.

For $i = 1, \cdots, N_R$,

- *If not already available, compute the norm of each observed and simulated trace.*
- *Set*

$$\boldsymbol{\theta}_i = \frac{\mathcal{A}_i - \mathcal{A}_i^{\mathrm{obs}}}{\left(\mathcal{A}_i^{\mathrm{obs}}\right)^2 \mathcal{A}_i}\mathbf{v}_i + \frac{\mathbf{d}_i^\top \mathbf{v}_i}{\mathcal{A}_i^{\mathrm{obs}}\mathcal{A}_i^3}\mathbf{d}_i.$$

Finally, set

$$\frac{\partial^2 \mathrm{M}}{\partial \mathbf{d}\,\partial \mathbf{d}}\mathbf{v} = \begin{bmatrix}\boldsymbol{\theta}_1^\top & \cdots & \boldsymbol{\theta}_{N_R}^\top\end{bmatrix}^\top.$$

7.2 Cross-Correlation Time Shift Misfit

Explicitly including phase information of the observed and simulated waveforms makes it possible to extract more information from the data, and this is the goal of the cross-correlation time shift misfit. The misfit was introduced in [81] and is similar in spirit to ideas discussed previously in [33], [17] and [76]. Further development was done in [40] and the misfit was applied to actual data in [135] and [20]. We

again follow the discussion in [36], where the gradient of the misfit was derived in a continuous setting.

The cross-correlation time shift misfit assumes that there is only a phase difference between the observed waveform and the simulated one, and measures the magnitude of this shift by cross-correlating the observed and simulated data.

To start, let us assume that the observed and simulated data at the ith receiver are both continuous time series $\mathbf{d}_i^{\text{obs}}(t)$ and $\mathbf{d}_i(t)$. We do this in order to define the *optimal* or *cross-correlation time shift* for the ith receiver by

$$\mathcal{T}_i := \arg\max_{t'} \mathcal{C}\left(t';\, \mathbf{d}_i, \mathbf{d}_i^{\text{obs}}\right) = \arg\max_{t'} \int_0^T \mathbf{d}_i^{\text{obs}}(t)\, \mathbf{d}_i(t+t')\, \mathrm{d}t \tag{7.4}$$

and the corresponding optimality condition below. For simplicity we assume that $\mathcal{T}_i$ is a multiple of the discrete time step τ. Note that the optimal time shift $\mathcal{T}_i$ depends on $\mathbf{d}_i$. The data misfit is given by

$$\mathrm{M} = \frac{1}{2} \sum_i \mathcal{T}_i^2, \tag{7.5}$$

and its derivatives in terms of $\mathcal{T}_i$ are

$$\nabla_{\mathbf{d}} \mathrm{M} = \begin{bmatrix} \mathcal{T}_1 \nabla_{\mathbf{d}_1} \mathcal{T}_1 \\ \vdots \\ \mathcal{T}_{N_R} \nabla_{\mathbf{d}_{N_R}} \mathcal{T}_{N_R} \end{bmatrix}$$

and

$$\frac{\partial^2 \mathrm{M}}{\partial \mathbf{d} \partial \mathbf{d}} = \begin{bmatrix} \dfrac{\partial^2 \mathrm{M}}{\partial \mathbf{d}_1 \partial \mathbf{d}_1} & & \\ & \ddots & \\ & & \dfrac{\partial^2 \mathrm{M}}{\partial \mathbf{d}_{N_R} \partial \mathbf{d}_{N_R}} \end{bmatrix}$$

with $\dfrac{\partial^2 \mathrm{M}}{\partial \mathbf{d}_i \partial \mathbf{d}_i} = \mathcal{T}_1 \dfrac{\partial^2 \mathcal{T}_i}{\partial \mathbf{d}_i \partial \mathbf{d}_i} + \nabla_{\mathbf{d}_i} \mathcal{T}_i \left(\nabla_{\mathbf{d}_i} \mathcal{T}_i\right)^\top$.

We therefore need to find expressions for the terms $\nabla_{\mathbf{d}_i}\mathcal{T}_i$ and $\dfrac{\partial^2\mathcal{T}_i}{\partial\mathbf{d}_i\partial\mathbf{d}_i}$. Using integration by parts we get the following condition for $\mathcal{T}_i$:

$$0 = \frac{\mathrm{d}}{\mathrm{d}t'}\mathcal{C}\left(t';\mathbf{d}_i,\mathbf{d}_i^{\mathrm{obs}}\right)\Big|_{t'=\mathcal{T}_i} = \int_0^T \mathbf{d}_i^{\mathrm{obs}}(t)\,\dot{\mathbf{d}}_i(t+\mathcal{T}_i)\,\mathrm{d}t$$
$$= -\int_0^T \dot{\mathbf{d}}_i^{\mathrm{obs}}(t-\mathcal{T}_i)\,\mathbf{d}_i(t)\,\mathrm{d}t,$$

where we have assumed $\mathbf{d}_i(0) = \mathbf{d}_i^{\mathrm{obs}}(0) = 0$ and $\mathbf{d}_i(T) = \mathbf{d}_i^{\mathrm{obs}}(T) = 0$.

Now consider the corresponding discrete setting, where this optimality condition becomes

$$0 = -\left(S\left(\mathcal{T}_i\right)\dot{\mathbf{d}}_i^{\mathrm{obs}}\right)^\top \mathbf{d}_i, \tag{7.6}$$

with $S\left(t'\right)$ being a discrete shift operator that has the effect of moving whatever time series it acts on by an amount t' to the right and $\dot{\mathbf{d}}_i^{\mathrm{obs}}$ being the discretized derivative of the continuous observed data.

Differentiating (7.6) with respect to $\mathbf{d}$ then gives

$$0 = -S\left(\mathcal{T}_i\right)\dot{\mathbf{d}}_i^{\mathrm{obs}} - \mathbf{d}_i^\top\left(\dot{S}\left(\mathcal{T}_i\right)\dot{\mathbf{d}}_i^{\mathrm{obs}}\right)\nabla_{\mathbf{d}_i}\mathcal{T}_i \tag{7.7}$$
$$\Rightarrow \qquad \nabla_{\mathbf{d}_i}\mathcal{T}_i = -\frac{S\left(\mathcal{T}_i\right)\dot{\mathbf{d}}_i^{\mathrm{obs}}}{\mathbf{d}_i^\top\left(\dot{S}\left(\mathcal{T}_i\right)\dot{\mathbf{d}}_i^{\mathrm{obs}}\right)}.$$

At this point note that

$$\dot{S}\left(\mathcal{T}_i\right) \approx \frac{S\left(\mathcal{T}_i+\tau\right) - S\left(\mathcal{T}_i\right)}{\tau} = S\left(\mathcal{T}_i\right)\frac{S\left(\tau\right)-1}{\tau}$$

and that

$$\frac{S\left(\tau\right)-1}{\tau}\mathbf{d}_i \approx \dot{\mathbf{d}}_i$$

since $S\left(\tau\right)$ has the effect of shifting $\mathbf{d}_i$ to the right by one time step. Therefore

$$\mathbf{d}_i^\top\left(\dot{S}\left(\mathcal{T}_i\right)\dot{\mathbf{d}}_i^{\mathrm{obs}}\right) \approx \dot{\mathbf{d}}_i^\top\left(S\left(\mathcal{T}_i\right)\dot{\mathbf{d}}_i^{\mathrm{obs}}\right).$$

This relationship can be shown to be exact in the continuous setting using integration by parts.

Next, since $\mathcal{T}_i$ is the optimal time shift and we assume that the observed and simulated data vary only in the arrival time and not in the shape or amplitude of the waveform, we have that $\mathbf{d}_i = S(\mathcal{T}_i)\,\mathbf{d}_i^{\text{obs}}$ and hence

$$\dot{\mathbf{d}}_i^\top \left(S(\mathcal{T}_i)\,\dot{\mathbf{d}}_i^{\text{obs}} \right) = \dot{\mathbf{d}}_i^\top \dot{\mathbf{d}}_i.$$

As a result of all of this we have

$$\nabla_{\mathbf{d}_i} \mathcal{T}_i = -\frac{S(\mathcal{T}_i)\,\dot{\mathbf{d}}_i^{\text{obs}}}{\mathbf{d}_i^\top \left(\dot{S}(\mathcal{T}_i)\,\dot{\mathbf{d}}_i^{\text{obs}} \right)} = -\frac{1}{\left\| \dot{\mathbf{d}}_i \right\|^2} \dot{\mathbf{d}}_i. \tag{7.8}$$

It follows that

$$\nabla_{\mathbf{d}_i} \mathrm{M} = \mathcal{T}_i \nabla_{\mathbf{d}_i} \mathcal{T}_i = -\frac{\mathcal{T}_i}{\left\| \dot{\mathbf{d}}_i \right\|^2} \dot{\mathbf{d}}_i. \tag{7.9}$$

The algorithm for computing the gradient of M with respect to the simulated data is given below.

Algorithm 7.3. Gradient of the Cross-Correlation Time-Shift Misfit

For $i = 1, \cdots, N_R$,

- *If not already available, compute the optimal time shift* $\mathcal{T}_i$ *by finding the shift that maximizes the value of the cross-correlation in* (7.4).
- *Use a finite difference approximation to find the approximate derivative* $\dot{\mathbf{d}}_i$ *of the simulated data* $\mathbf{d}_i$.
- *Set*

$$\boldsymbol{\theta}_i = -\frac{\mathcal{T}_i}{\left\| \dot{\mathbf{d}}_i \right\|^2} \dot{\mathbf{d}}_i.$$

Finally, set

$$\nabla_{\mathbf{d}}\mathrm{M} = \begin{bmatrix} \boldsymbol{\theta}_1^\top & \cdots & \boldsymbol{\theta}_{N_R}^\top \end{bmatrix}^\top .$$

Let us now compute the Hessian of $\mathcal{T}_i$ with respect to its corresponding trace $\mathbf{d}_i$. For this we go back to (7.7). Taking the derivative with respect to $\mathbf{d}_i$ on both sides gives

$$\begin{aligned}
\mathbf{0} &= \frac{\partial}{\partial\,\mathbf{d}_i}\left(S\left(\mathcal{T}_i\right)\dot{\mathbf{d}}_i^{\mathrm{obs}} + \mathbf{d}_i^\top\left(\dot{S}\left(\mathcal{T}_i\right)\dot{\mathbf{d}}_i^{\mathrm{obs}}\right)\nabla_{\mathbf{d}_i}\mathcal{T}_i\right) \\
&= \left(2\dot{S}\left(\mathcal{T}_i\right)\dot{\mathbf{d}}_i^{\mathrm{obs}} + \mathbf{d}_i^\top\left(\ddot{S}\left(\mathcal{T}_i\right)\dot{\mathbf{d}}_i^{\mathrm{obs}}\right)\nabla_{\mathbf{d}_i}\mathcal{T}\right)\nabla_{\mathbf{d}}\mathcal{T}_i^\top + \mathbf{d}^\top\left(\dot{S}\left(\mathcal{T}_i\right)\dot{\mathbf{d}}_i^{\mathrm{obs}}\right)\frac{\partial^2\mathcal{T}_i}{\partial\mathbf{d}_i\,\partial\mathbf{d}_i}
\end{aligned}$$

so that

$$\begin{aligned}
\frac{\partial^2\mathcal{T}_i}{\partial\mathbf{d}_i\,\partial\mathbf{d}_i} &= -\frac{1}{\mathbf{d}_i^\top\left(\dot{S}\left(\mathcal{T}_i\right)\dot{\mathbf{d}}_i^{\mathrm{obs}}\right)}\left(2\dot{S}\left(\mathcal{T}_i\right)\dot{\mathbf{d}}_i^{\mathrm{obs}} + \mathbf{d}_i^\top\left(\ddot{S}\left(\mathcal{T}_i\right)\dot{\mathbf{d}}_i^{\mathrm{obs}}\right)\nabla_{\mathbf{d}_i}\mathcal{T}_i\right)\nabla_{\mathbf{d}_i}\mathcal{T}_i^\top \\
&= -\frac{1}{\dot{\mathbf{d}}_i^\top\left(S\left(\mathcal{T}_i\right)\dot{\mathbf{d}}_i^{\mathrm{obs}}\right)}\left(2S\left(\mathcal{T}_i\right)\ddot{\mathbf{d}}_i^{\mathrm{obs}} + \dot{\mathbf{d}}_i^\top\left(S\left(\mathcal{T}_i\right)\ddot{\mathbf{d}}_i^{\mathrm{obs}}\right)\nabla_{\mathbf{d}_i}\mathcal{T}_i\right)\nabla_{\mathbf{d}_i}\mathcal{T}_i^\top \\
&= -\frac{1}{\left\|\dot{\mathbf{d}}_i\right\|^2}\left(2\ddot{\mathbf{d}}_i + \dot{\mathbf{d}}_i^\top\ddot{\mathbf{d}}_i\nabla_{\mathbf{d}_i}\mathcal{T}_i\right)\nabla_{\mathbf{d}_i}\mathcal{T}_i^\top \\
&= -\frac{1}{\left\|\dot{\mathbf{d}}_i\right\|^4}\left(-2\ddot{\mathbf{d}}_i\,\dot{\mathbf{d}}_i^\top + \frac{\dot{\mathbf{d}}_i^\top\ddot{\mathbf{d}}_i\,\dot{\mathbf{d}}_i\,\dot{\mathbf{d}}_i^\top}{\left\|\dot{\mathbf{d}}_i\right\|^2}\right)
\end{aligned} \tag{7.10}$$

where we have again used the property that the derivatives of $S(\mathcal{T}_i)$ can be moved to the vectors and also the assumption that the optimally shifted simulated data is the same as the observed data, which must also hold for their derivatives. Therefore

$$\begin{aligned}
\frac{\partial^2\mathrm{M}}{\partial\mathbf{d}_i\,\partial\mathbf{d}_i} &= \mathcal{T}_i\frac{\partial^2\mathcal{T}_i}{\partial\mathbf{d}_i\,\partial\mathbf{d}_i} + \nabla_{\mathbf{d}_i}\mathcal{T}\left(\nabla_{\mathbf{d}_i}\mathcal{T}_i\right)^\top \\
&= \frac{\mathcal{T}_i}{\left\|\dot{\mathbf{d}}_i\right\|^4}\left(2\ddot{\mathbf{d}}_i - \frac{\dot{\mathbf{d}}_i^\top\ddot{\mathbf{d}}_i\,\dot{\mathbf{d}}_i}{\left\|\dot{\mathbf{d}}_i\right\|^2}\right)\dot{\mathbf{d}}_i^\top + \frac{\dot{\mathbf{d}}_i\,\dot{\mathbf{d}}_i^\top}{\left\|\dot{\mathbf{d}}_i\right\|^4}.
\end{aligned} \tag{7.11}$$

The procedure for computing the product of the Hessian of M with some arbitrary vector $\mathbf{v}$ of length $N_{\mathbf{d}}$, defined analogously to $\mathbf{d}$, is given in Algorithm 7.4.

Algorithm 7.4. Product of the Hessian of the Cross-Correlation Time Shift Misfit

Procedure for computing the product of the Hessian $\dfrac{\partial^2 \mathrm{M}}{\partial \mathbf{d}\, \partial \mathbf{d}}$ *with an arbitrary vector* $\mathbf{v} = \begin{bmatrix} \mathbf{v}_1^\top & \cdots & \mathbf{v}_{N_R}^\top \end{bmatrix}^\top$, *where each* $\mathbf{v}_i$ *has length* $N_{\mathbf{d}_i}$.

For $i = 1, \cdots, N_R$,

- *If not already available, compute the optimal time shift* $\mathcal{T}_i$ *by finding the shift that maximizes the value of the cross-correlation in* (7.4).
- *Use finite difference approximations to find the approximate derivatives* $\dot{\mathbf{d}}_i$ *and* $\ddot{\mathbf{d}}_i$ *of the trace* $\mathbf{d}_i$.
- *Set*

$$\boldsymbol{\theta}_i = \mathcal{T}_i \frac{\dot{\mathbf{d}}_i^\top \mathbf{v}_i}{\left\|\dot{\mathbf{d}}_i\right\|^4} \left(2\ddot{\mathbf{d}}_i - \frac{\dot{\mathbf{d}}_i^\top \ddot{\mathbf{d}}_i\, \dot{\mathbf{d}}_i}{\left\|\dot{\mathbf{d}}_i\right\|^2} \right) + \frac{\dot{\mathbf{d}}_i^\top \mathbf{v}_i}{\left\|\dot{\mathbf{d}}_i\right\|^4} \dot{\mathbf{d}}_i.$$

Finally, set

$$\frac{\partial^2 \mathrm{M}}{\partial \mathbf{d}\, \partial \mathbf{d}} \mathbf{v} = \begin{bmatrix} \boldsymbol{\theta}_1^\top & \cdots & \boldsymbol{\theta}_{N_R}^\top \end{bmatrix}^\top.$$

Note: Testing the correctness of the numerical computation of $\nabla_{\mathbf{d}}\mathrm{M}$ and $\dfrac{\partial^2 \mathrm{M}}{\partial \mathbf{d}\, \partial \mathbf{d}} \mathbf{v}$ may prove tricky since a perturbation of the simulated data $\mathbf{d}$ will usually not lead to a change in any of the $\mathcal{T}_i$; the best approach for testing the implementation is to test the gradient $\nabla_{\mathbf{p}}\mathrm{M}$ (which of course depends on $\nabla_{\mathbf{d}}\mathrm{M}$) and using a very small measurement interval τ; similarly for $\dfrac{\partial^2 \mathrm{M}}{\partial \mathbf{d}\, \partial \mathbf{d}} \mathbf{v}$.

7.3 Interferometric Misfit

The interferometric misfit has shown some potential [67, 129] in application to seismic imaging problems. It does not appear to be too well-known, but providing the derivatives of this misfit with respect to the data will hopefully facilitate the use of gradient-based optimization procedures with this misfit.

The idea is to measure the phase difference, using cross-correlation, between the simulated data measured by a given pair of receivers, then comparing it to the cross-correlation of the actual data from those receivers. This serves to give a relative measure of discrepancy between the simulated and observed data [67], rather than an absolute one, as would be the case for the least-squares misfit.

The form of the interferometric misfit that we are interested in measures the differences between the cross-correlations of pairs of simulated and observed data as follows:

$$\frac{1}{2}\sum_{i,j}\mathbf{E}_{ij}\left\|\mathbf{d}_i(t)\star\mathbf{d}_j(t)-\mathbf{d}_i^{\mathrm{obs}}(t)\star\mathbf{d}_j^{\mathrm{obs}}(t)\right\|^2.$$

The matrix $\mathbf{E}$ is a $N_R\times N_R$ matrix of zeros and ones that determines whether a given pair is included in the summation or not. This is a mechanism that is introduced so that only particular combinations of traces are included in the misfit computation, allowing us to easily exclude receiver pairs that are, for instance, too far apart to provide useful information. The cross-correlation is defined to be

$$\mathbf{d}_i(t)\star\mathbf{d}_j(t)=\int_{\mathbb{R}}\overline{\mathbf{d}}_i(t)\,\mathbf{d}_j(t+t')\,\mathrm{d}t',$$

and similarly for $\mathbf{d}_i^{\mathrm{obs}}(t)$.

For discrete measurements the cross-correlation can be efficiently computed using the Fast Fourier Transform (`fft`) and its inverse (`ifft`). The MATLAB pseudocode

looks as follows:

$$\mathbf{d}_i \star \mathbf{d}_j = \texttt{fftshift}\left(\texttt{ifft}\left(\overline{\texttt{fft}\left(\mathbf{d}_i, N_\omega\right)} .* \texttt{fft}\left(\mathbf{d}_j, N_\omega\right)\right)\right). \tag{7.12}$$

Each trace $\mathbf{d}_i$ is assumed to have a length of $N_{\mathbf{d}_i}$. For simplicity, we assume that there is a fixed number of frequencies N_ω used by the Fourier transform, with $N_\omega \geq N_{\mathbf{d}_i} + N_{\mathbf{d}_j} - 1$ for all i, j.

The `fft` above can be represented using the $N_\omega \times N_\omega$ discrete Fourier transform matrix $\mathbf{F}$. The `ifft` is represented by $\mathbf{F}^{-1} = \frac{1}{N_\omega}\overline{\mathbf{F}}$. The discrete cross-correlation computation (7.12) can therefore also be written as

$$\mathbf{d}_i \star \mathbf{d}_j = \mathbf{J}\,\mathbf{F}^{-1}\left(\operatorname{diag}\left(\overline{\mathbf{F}\,\mathbf{U}_{ij}\,\mathbf{d}_i}\right)\,\mathbf{F}\,\mathbf{U}_{ji}\,\mathbf{d}_j\right) = \mathbf{J}\,\mathbf{F}^{-1}\left(\operatorname{diag}\left(\overline{\mathbf{F}}\,\mathbf{U}_{ij}\,\mathbf{d}_i\right)\,\mathbf{F}\,\mathbf{U}_{ji}\,\mathbf{d}_j\right) \tag{7.13}$$

which is the form we will work with here. We have introduced

$$\mathbf{J} = \begin{bmatrix} \mathbf{0}_{\lfloor N_\omega/2\rfloor \times \lceil N_\omega/2\rceil} & \mathbf{I}_{\lfloor N_\omega/2\rfloor} \\ \mathbf{I}_{\lceil N_\omega/2\rceil} & \mathbf{0}_{\lceil N_\omega/2\rceil \times \lfloor N_\omega/2\rfloor} \end{bmatrix} \qquad \text{and} \qquad \mathbf{U}_i = \begin{bmatrix} \mathbf{I}_{N_{\mathbf{d}_i}} \\ \mathbf{0}_{\left(N_\omega - N_{\mathbf{d}_i}\right)\times N_{\mathbf{d}_i}} \end{bmatrix}$$

where $\mathbf{J}$ acts as the matrix representation of `fftshift` and $\mathbf{U}_i$ appends $N_\omega - N_{\mathbf{d}_i}$ zeros to a vector of length $N_{\mathbf{d}_i}$. $\mathbf{J}^\top$ acts as the matrix representation of `ifftshift`.

The discrete cross-correlation misfit is then

$$\mathrm{M} = \frac{1}{2}\sum_{i,j} \mathbf{E}_{ij}\left\|\mathbf{d}_i \star \mathbf{d}_j - \mathbf{d}_i^{\text{obs}} \star \mathbf{d}_j^{\text{obs}}\right\|^2 \tag{7.14}$$

with $\star$ as in (7.13). However, we need to express M as a function of $\mathbf{d}$ in order to take derivatives with respect to $\mathbf{d}$. To this end, let

$$\begin{aligned} \mathbf{G}_1 &:= \mathbf{1}_{N_R} \otimes \left(\left(\mathbf{I}_{N_R} \otimes \mathbf{F}\right)\mathbf{U}\right), \\ \mathbf{G}_2 &:= \left(\widetilde{\mathbf{E}} \otimes \mathbf{I}_{N_\omega}\right)\left(\left(\mathbf{I}_{N_R} \otimes \overline{\mathbf{F}}\right)\mathbf{U}\right) \\ \mathbf{G}_3 &:= \mathbf{I}_{N_R^2} \otimes \left(\mathbf{J}\,\mathbf{F}^{-1}\right) \end{aligned} \tag{7.15}$$

with

$$\mathbf{U} = \begin{bmatrix} \mathbf{U}_1 & & \\ & \ddots & \\ & & \mathbf{U}_{N_R} \end{bmatrix}, \qquad \widetilde{\mathbf{E}} = \begin{bmatrix} \mathbf{E}_{1,:} & & \\ & \ddots & \\ & & \mathbf{E}_{N_R,:} \end{bmatrix}$$

and $\mathbf{1}_{N_R}$ a vector of ones of length N_R. It is easy to see that $\mathbf{G}_1$ and $\mathbf{G}_2$ are matrices of size $N_R^2 N_\omega \times N_\mathbf{d}$ and $\mathbf{G}_3$ is a matrix of size $N_R^2 N_\omega \times N_R^2 N_\omega$. Note that

$$\mathbf{G}_3^\star \mathbf{G}_3 = \left(\mathbf{I}_{N_R^2} \otimes \left(\left(\mathbf{F}^{-1}\right)^\star \mathbf{J}^\top\right)\right)\left(\mathbf{I}_{N_R^2} \otimes \left(\mathbf{J}\,\mathbf{F}^{-1}\right)\right) = \frac{1}{N_\omega}\mathbf{I}_{N_R^2 N_\omega}$$

since $\mathbf{J}^\top\mathbf{J} = \mathbf{I}_{N_\omega}$ and $\left(\mathbf{F}^{-1}\right)^\star \mathbf{F}^{-1} = \frac{1}{N_\omega}\mathbf{F}\,\mathbf{F}^{-1} = \frac{1}{N_\omega}\mathbf{I}_{N_\omega}$, where we have used the property of the discrete Fourier transform matrix that $\mathbf{F}^\star = \overline{\mathbf{F}} = N_\omega\,\mathbf{F}^{-1}$.

(7.14) can therefore be rewritten as

$$\mathrm{M} = \frac{1}{2}\left\|\mathbf{G}_3\left(\mathrm{diag}\left(\mathbf{G}_2\mathbf{d}\right)\mathbf{G}_1\mathbf{d} - \mathrm{diag}\left(\mathbf{G}_2\mathbf{d}^{\mathrm{obs}}\right)\mathbf{G}_1\mathbf{d}^{\mathrm{obs}}\right)\right\|^2. \tag{7.16}$$

Then the gradient with respect to the data $\mathbf{d}$ is

$$\begin{aligned}\nabla_\mathbf{d}\mathrm{M} &= \left(\mathrm{diag}\left(\mathbf{G}_2\mathbf{d}\right)\mathbf{G}_1 + \mathrm{diag}\left(\mathbf{G}_1\mathbf{d}\right)\mathbf{G}_2\right)^\star \cdot \\ &\qquad \cdot \mathbf{G}_3^\star\mathbf{G}_3\left(\mathrm{diag}\left(\mathbf{G}_2\mathbf{d}\right)\mathbf{G}_1\mathbf{d} - \mathrm{diag}\left(\mathbf{G}_2\mathbf{d}^{\mathrm{obs}}\right)\mathbf{G}_1\mathbf{d}^{\mathrm{obs}}\right) \\ &= \frac{1}{N_\omega}\left(\mathbf{G}_1^\star\,\mathrm{diag}\left(\overline{\mathbf{G}_2}\mathbf{d}\right) + \mathbf{G}_2^\star\,\mathrm{diag}\left(\overline{\mathbf{G}_1}\mathbf{d}\right)\right) \cdot \\ &\qquad \cdot \left(\mathrm{diag}\left(\mathbf{G}_2\mathbf{d}\right)\mathbf{G}_1\mathbf{d} - \mathrm{diag}\left(\mathbf{G}_2\mathbf{d}^{\mathrm{obs}}\right)\mathbf{G}_1\mathbf{d}^{\mathrm{obs}}\right)\end{aligned} \tag{7.17}$$

It is beneficial to bring this expression into a form that is easier to compute by rewriting them in terms of Fourier transforms and cross-correlations. We start by looking at the products $\mathbf{G}_1\mathbf{x}$, $\mathbf{G}_2\mathbf{x}$ and $\mathbf{G}_1^\top\mathbf{y}$, $\mathbf{G}_2^\top\mathbf{y}$, where $\mathbf{x} = \begin{bmatrix}\mathbf{x}_1^\top & \cdots & \mathbf{x}_{N_R}^\top\end{bmatrix}^\top$ is a vector of size $N_\mathbf{d} = \sum_{i=1}^{N_R} N_{\mathbf{d}_i}$ and $\mathbf{y} = \begin{bmatrix}\mathbf{y}_1^\top & \cdots & \mathbf{y}_{N_R}^\top\end{bmatrix}^\top$ is a vector of size $N_R^2 N_\omega$, with $\mathbf{y}_i = \begin{bmatrix}\mathbf{y}_{i,1}^\top & \cdots & \mathbf{y}_{i,N_R}^\top\end{bmatrix}^\top$ and each $\mathbf{y}_{i,j}$ having length N_ω:

- $\mathbf{G}_1\mathbf{x}$:

$$\mathbf{G}_1\mathbf{x} = \mathbf{1}_{N_R} \otimes \left(\left(\mathbf{I}_{N_R} \otimes \mathbf{F}\right)\mathbf{U}\right)\mathbf{x} = \mathbf{1}_{N_R} \otimes \begin{bmatrix} \mathbf{F}\mathbf{U}_1\mathbf{x}_1 \\ \vdots \\ \mathbf{F}\mathbf{U}_{N_R}\mathbf{x}_{N_R} \end{bmatrix} =: \mathbf{1}_{N_R} \otimes \mathbf{z}\left(\mathbf{x}\right) \tag{7.18}$$

- $\mathbf{G}_2\mathbf{x}$:

$$\mathbf{G}_2\mathbf{x} = \left(\widetilde{\mathbf{E}} \otimes \mathbf{I}_{N_\omega}\right) \begin{bmatrix} \overline{\mathbf{F}}\mathbf{U}_1\mathbf{x}_1 \\ \vdots \\ \overline{\mathbf{F}}\mathbf{U}_{N_R}\mathbf{x}_{N_R} \end{bmatrix} = \begin{bmatrix} \mathbf{E}_{1,:} \otimes \overline{\mathbf{F}}\mathbf{U}_1\mathbf{x}_1 \\ \vdots \\ \mathbf{E}_{N_R,:} \otimes \overline{\mathbf{F}}\mathbf{U}_{N_R}\mathbf{x}_{N_R} \end{bmatrix} =: \begin{bmatrix} \mathbf{w}_1\left(\mathbf{x}_1\right) \\ \vdots \\ \mathbf{w}_{N_R}\left(\mathbf{x}_{N_R}\right) \end{bmatrix} \tag{7.19}$$

with $\mathbf{w}_{i,j}\left(\mathbf{x}_i\right) = \mathbf{E}_{i,j}\,\overline{\mathbf{F}}\mathbf{U}_i\mathbf{x}_i$.

- $\mathbf{G}_1^\star\mathbf{y}$:

$$\begin{aligned} \mathbf{G}_1^\star\mathbf{y} &= \mathbf{1}_{N_R}^\top \otimes \left(\mathbf{U}^\top\left(\mathbf{I}_{N_R} \otimes \mathbf{F}^\star\right)\right)\mathbf{y} \\ &= \begin{bmatrix} \mathbf{U}_1^\top & & \\ & \ddots & \\ & & \mathbf{U}_{N_R}^\top \end{bmatrix} \overline{\mathbf{F}} \sum_{i=1}^{N_R} \mathbf{y}_i = \begin{bmatrix} \mathbf{U}_1^\top\overline{\mathbf{F}} \sum_{i=1}^{N_R} \mathbf{y}_{i,1} \\ \vdots \\ \mathbf{U}_{N_R}^\top\overline{\mathbf{F}} \sum_{i=1}^{N_R} \mathbf{y}_{i,N_R} \end{bmatrix} \end{aligned} \tag{7.20}$$

- $\mathbf{G}_2^\star\mathbf{y}$:

$$\begin{aligned} \mathbf{G}_2^\star\mathbf{y} &= \mathbf{U}^\top\left(\mathbf{I}_{N_R} \otimes \left(\overline{\mathbf{F}}\right)^\star\right)\left(\widetilde{\mathbf{E}}^\top \otimes \mathbf{I}_{N_\omega}\right)\mathbf{y} \\ &= \mathbf{U}^\top\left(\mathbf{I}_{N_R} \otimes \mathbf{F}\right) \begin{bmatrix} \sum_{i=1}^{N_R} \mathbf{E}_{1,i}\,\mathbf{y}_{1,i} \\ \vdots \\ \sum_{i=1}^{N_R} \mathbf{E}_{N_R,i}\,\mathbf{y}_{N_R,i} \end{bmatrix} = \begin{bmatrix} \mathbf{U}_1^\top\mathbf{F} \sum_{i=1}^{N_R} \mathbf{E}_{1,i}\,\mathbf{y}_{1,i} \\ \vdots \\ \mathbf{U}_{N_R}^\top\mathbf{F} \sum_{i=1}^{N_R} \mathbf{E}_{N_R,i}\,\mathbf{y}_{N_R,i} \end{bmatrix}. \end{aligned} \tag{7.21}$$

The conjugates of the above products are formed by conjugating $\mathbf{F}$ and its inverse. We also need the following terms:

- $\overline{\mathbf{G}_1}\mathbf{d} \odot \mathbf{G}_1\mathbf{d}$:

$$\begin{aligned}\overline{\mathbf{G}_1}\mathbf{d} \odot \mathbf{G}_1\mathbf{d} &= \mathbf{1}_{N_R} \otimes \left(\overline{\mathbf{z}}\left(\mathbf{d}\right) \odot \mathbf{z}\left(\mathbf{d}\right)\right) \\ &= \mathbf{1}_{N_R} \otimes \begin{bmatrix} \left(\overline{\mathbf{F}}\mathbf{U}_1\mathbf{d}_1\right) \odot \left(\mathbf{F}\mathbf{U}_1\mathbf{d}_1\right) \\ \vdots \\ \left(\overline{\mathbf{F}}\mathbf{U}_{N_R}\mathbf{d}_{N_R}\right) \odot \left(\mathbf{F}\mathbf{U}_{N_R}\mathbf{d}_{N_R}\right) \end{bmatrix}\end{aligned} \tag{7.22}$$

- $\overline{\mathbf{G}_2}\mathbf{d} \odot \mathbf{G}_2\mathbf{d}$:

$$\overline{\mathbf{G}_2}\mathbf{d} \odot \mathbf{G}_2\mathbf{d} = \begin{bmatrix} \overline{\mathbf{w}}_1\left(\mathbf{d}_1\right) \odot \mathbf{w}_1\left(\mathbf{d}_1\right) \\ \vdots \\ \overline{\mathbf{w}}_{N_R}\left(\mathbf{d}_{N_R}\right) \odot \mathbf{w}_{N_R}\left(\mathbf{d}_{N_R}\right) \end{bmatrix} \tag{7.23}$$

with $\overline{\mathbf{w}}_{i,j}\left(\mathbf{d}_i\right) \odot \mathbf{w}_{i,j}\left(\mathbf{d}_i\right) = \mathbf{E}_{i,j}\left(\mathbf{F}\mathbf{U}_i\mathbf{d}_i\right) \odot \left(\overline{\mathbf{F}}\mathbf{U}_i\mathbf{d}_i\right)$.

- $\mathbf{G}_2\mathbf{d} \odot \mathbf{G}_1\mathbf{d}$:

$$\mathbf{G}_2\mathbf{d} \odot \mathbf{G}_1\mathbf{d} = \begin{bmatrix} \mathbf{w}_1\left(\mathbf{d}_1\right) \odot \mathbf{z}\left(\mathbf{d}\right) \\ \vdots \\ \mathbf{w}_{N_R}\left(\mathbf{d}_{N_R}\right) \odot \mathbf{z}\left(\mathbf{d}\right) \end{bmatrix} \tag{7.24}$$

with $\mathbf{w}_{i,j}\left(\mathbf{d}_i\right) \odot \mathbf{z}_j\left(\mathbf{d}\right) = \mathbf{E}_{i,j}\left(\overline{\mathbf{F}}\mathbf{U}_i\mathbf{d}_i\right) \odot \left(\mathbf{F}\mathbf{U}_j\mathbf{d}_j\right)$, and similarly for $\mathbf{G}_2\mathbf{d}^{\text{obs}} \odot \mathbf{G}_1\mathbf{d}^{\text{obs}}$.

- $\overline{\mathbf{G}_2}\mathbf{d} \odot \mathbf{G}_1\mathbf{d}$:

$$\overline{\mathbf{G}_2}\mathbf{d} \odot \mathbf{G}_1\mathbf{d} = \begin{bmatrix} \overline{\mathbf{w}}_1\left(\mathbf{d}_1\right) \odot \mathbf{z}\left(\mathbf{d}\right) \\ \vdots \\ \overline{\mathbf{w}}_{N_R}\left(\mathbf{d}_{N_R}\right) \odot \mathbf{z}\left(\mathbf{d}\right) \end{bmatrix} \tag{7.25}$$

with $\overline{\mathbf{w}}_{i,j}\left(\mathbf{d}_i\right) \odot \mathbf{z}_j\left(\mathbf{d}\right) = \mathbf{E}_{i,j}\left(\mathbf{F}\mathbf{U}_i\mathbf{d}_i\right) \odot \left(\mathbf{F}\mathbf{U}_j\mathbf{d}_j\right)$.

- $\mathbf{G}_2\mathbf{d} \odot \overline{\mathbf{G}_1}\mathbf{d}$:

$$\mathbf{G}_2\mathbf{d} \odot \overline{\mathbf{G}_1}\mathbf{d} = \begin{bmatrix} \mathbf{w}_1\left(\mathbf{d}_1\right) \odot \overline{\mathbf{z}}\left(\mathbf{d}\right) \\ \vdots \\ \mathbf{w}_{N_R}\left(\mathbf{d}_{N_R}\right) \odot \overline{\mathbf{z}}\left(\mathbf{d}\right) \end{bmatrix} \tag{7.26}$$

with $\mathbf{w}_{i,j}\left(\mathbf{d}_i\right) \odot \overline{\mathbf{z}}_j\left(\mathbf{d}\right) = \mathbf{E}_{i,j}\left(\overline{\mathbf{F}}\mathbf{U}_i\mathbf{d}_i\right) \odot \left(\overline{\mathbf{F}}\mathbf{U}_j\mathbf{d}_j\right)$.

Using the above equations, we will now simplify the expression for the gradient in (7.17). Let

$$\mathbf{r} = \mathbf{G}_2\mathbf{d} \odot \mathbf{G}_1\mathbf{d} - \mathbf{G}_2\mathbf{d}^{\text{obs}} \odot \mathbf{G}_1\mathbf{d}^{\text{obs}}. \tag{7.27}$$

Then, using (7.19) and (7.20),

$$\mathbf{G}_1^\star \operatorname{diag}\left(\overline{\mathbf{G}_2}\mathbf{d}\right)\mathbf{r} = \begin{bmatrix} \mathbf{U}_1^\top \overline{\mathbf{F}} \sum_{i=1}^{N_R} \left(\mathbf{F}\mathbf{U}_i\mathbf{d}_i\right) \odot \mathbf{r}_{i1} \\ \vdots \\ \mathbf{U}_{N_R}^\top \overline{\mathbf{F}} \sum_{i=1}^{N_R} \left(\mathbf{F}\mathbf{U}_i\mathbf{d}_i\right) \odot \mathbf{r}_{iN_R} \end{bmatrix}; \tag{7.28}$$

we left out the $\mathbf{E}_{ij}$ since $\mathbf{E}_{ij} = \mathbf{E}_{ij}^2$ and there already is a $\mathbf{E}_{ij}$ appearing in $\mathbf{r}_{ij}$.

Similarly, using (7.18) and (7.21)

$$\mathbf{G}_2^\star \operatorname{diag}\left(\overline{\mathbf{G}_1}\mathbf{d}\right)\mathbf{r} = \begin{bmatrix} \mathbf{U}_1^\top \mathbf{F} \sum_{i=1}^{N_R} \mathbf{E}_{1i}\,\mathbf{y}_{1i} \\ \vdots \\ \mathbf{U}_{N_R}^\top \mathbf{F} \sum_{i=1}^{N_R} \mathbf{E}_{N_R i}\,\mathbf{y}_{N_R i} \end{bmatrix},$$

with $\mathbf{y} = \overline{\mathbf{G}_1}\mathbf{d} \odot \mathbf{r} = \left(\mathbf{1}_{N_R} \otimes \overline{\mathbf{z}}\left(\mathbf{d}\right)\right) \odot \mathbf{r}$, where

$$\mathbf{y}_i = \overline{\mathbf{z}}_i\left(\mathbf{d}\right) \odot \mathbf{r}_i = \begin{bmatrix} \overline{\mathbf{F}}\mathbf{U}_1\mathbf{d}_1 \odot \mathbf{r}_{i1} \\ \vdots \\ \overline{\mathbf{F}}\mathbf{U}_{N_R}\mathbf{d}_{N_R} \odot \mathbf{r}_{iN_R} \end{bmatrix}.$$

Therefore

$$\mathbf{G}_2^\star \operatorname{diag}\left(\overline{\mathbf{G}_1}\mathbf{d}\right)\mathbf{r} = \begin{bmatrix} \mathbf{U}_1^\top \mathbf{F} \sum_{i=1}^{N_R} \left(\overline{\mathbf{F}}\mathbf{U}_i\mathbf{d}_i\right) \odot \mathbf{r}_{1i} \\ \vdots \\ \mathbf{U}_{N_R}^\top \mathbf{F} \sum_{i=1}^{N_R} \left(\overline{\mathbf{F}}\mathbf{U}_i\mathbf{d}_i\right) \odot \mathbf{r}_{N_R i} \end{bmatrix}. \tag{7.29}$$

The algorithm for computing the gradient is given below.

Algorithm 7.5. Gradient of the Interferometric Misfit

- *If not already available, compute the discrete Fourier transform of each trace* $\mathbf{d}_i$*:*

$$\widehat{\mathbf{d}}_i = \texttt{fft}\left(\texttt{pad}\left(\mathbf{d}_i\right)\right)$$

  for $i = 1, \cdots, N_R$*, where* `pad` *pads* $\mathbf{d}_i$ *with* $N_\omega - N_{\mathbf{d}_i}$ *zeros. Likewise,*

$$\widehat{\mathbf{d}}_i^{\text{obs}} = \texttt{fft}\left(\texttt{pad}\left(\mathbf{d}_i^{\text{obs}}\right)\right).$$

- *Set* $\boldsymbol{\theta}_i = \mathbf{0}_{N_{\mathbf{d}_i}\times 1}$ *for* $i = 1, \cdots, N_R$*.*

- *For* $j = 1, \cdots, N_R$ *compute:*

  - *Set* $\boldsymbol{\phi}_1 = \mathbf{0}_{N_\omega \times 1}$ *and* $\boldsymbol{\phi}_2 = \mathbf{0}_{N_\omega \times 1}$ *(these are temporary storage arrays).*
  - *For* $j = 1, \cdots, N_R$*:*

$$\boldsymbol{\phi}_1 = \boldsymbol{\phi}_1 + \mathbf{E}_{ij}\left(\widehat{\mathbf{d}}_i \odot \left(\overline{\widehat{\mathbf{d}}_j} \odot \widehat{\mathbf{d}}_i - \overline{\widehat{\mathbf{d}}_j^{\text{obs}}} \odot \widehat{\mathbf{d}}_i^{\text{obs}}\right)\right)$$

  and

$$\boldsymbol{\phi}_2 = \boldsymbol{\phi}_2 + \mathbf{E}_{ji}\left(\overline{\widehat{\mathbf{d}}_j} \odot \left(\overline{\widehat{\mathbf{d}}_i} \odot \widehat{\mathbf{d}}_j - \overline{\widehat{\mathbf{d}}_i^{\text{obs}}} \odot \widehat{\mathbf{d}}_j^{\text{obs}}\right)\right).$$

- *Compute*

$$\boldsymbol{\theta}_i = \Re\left(\texttt{restrict}\left(\texttt{ifft}\left(\boldsymbol{\phi}_1\right) + \frac{1}{N_\omega}\texttt{fft}\left(\boldsymbol{\phi}_2\right)\right)\right),$$

  where `restrict` *restricts the vector to the first* $N_{\mathbf{d}_i}$ *entries.*

- *Finally, set*

$$\nabla_{\mathbf{d}}\mathrm{M} = \begin{bmatrix}\boldsymbol{\theta}_1^\top & \cdots & \boldsymbol{\theta}_{N_R}^\top\end{bmatrix}^\top.$$

Differentiating (7.17) with respect to $\mathbf{d}$ gives the Hessian

$$\begin{aligned}\frac{\partial^2 \mathrm{M}}{\partial \mathbf{d}\,\partial \mathbf{d}} = \frac{1}{N_\omega}\big[&\mathbf{G}_1^\star \operatorname{diag}\left(\overline{\mathbf{G}_2}\mathbf{d} \odot \mathbf{G}_2\mathbf{d}\right)\mathbf{G}_1 + \mathbf{G}_2^\star \operatorname{diag}\left(\overline{\mathbf{G}_1}\mathbf{d} \odot \mathbf{G}_1\mathbf{d}\right)\mathbf{G}_2 + \\ &+ \mathbf{G}_1^\star \operatorname{diag}\left(\overline{\mathbf{G}_2}\mathbf{d} \odot \mathbf{G}_1\mathbf{d}\right)\mathbf{G}_2 + \mathbf{G}_2^\star \operatorname{diag}\left(\overline{\mathbf{G}_1}\mathbf{d} \odot \mathbf{G}_2\mathbf{d}\right)\mathbf{G}_1 + \\ &+ \mathbf{G}_1^\star \operatorname{diag}\left(\mathbf{G}_2\mathbf{d} \odot \mathbf{G}_1\mathbf{d} - \mathbf{G}_2\mathbf{d}^{\mathrm{obs}} \odot \mathbf{G}_1\mathbf{d}^{\mathrm{obs}}\right)\overline{\mathbf{G}_2} + \\ &+ \mathbf{G}_2^\star \operatorname{diag}\left(\mathbf{G}_2\mathbf{d} \odot \mathbf{G}_1\mathbf{d} - \mathbf{G}_2\mathbf{d}^{\mathrm{obs}} \odot \mathbf{G}_1\mathbf{d}^{\mathrm{obs}}\right)\overline{\mathbf{G}_1}\,\big]\,. \end{aligned} \tag{7.30}$$

Our goal now is to find the expression for the product of the Hessian with an arbitrary vector $\mathbf{v}$ of length $N_{\mathbf{d}}$. Letting $\mathbf{r}$ be defined as in (7.27), we look at each of the terms in (7.30) separately:

- $\mathbf{G}_1^\star \operatorname{diag}\left(\overline{\mathbf{G}_2}\mathbf{d} \odot \mathbf{G}_2\mathbf{d}\right)\mathbf{G}_1\mathbf{v}$:
  Use (7.20) and let $\mathbf{y} = \left(\overline{\mathbf{G}_2}\mathbf{d} \odot \mathbf{G}_2\mathbf{d}\right) \odot \mathbf{G}_1\mathbf{v}$, with

$$\begin{aligned}\mathbf{y}_{i,j} &= \overline{\mathbf{w}}_{i,j}\left(\mathbf{d}_i\right) \odot \mathbf{w}_{i,j}\left(\mathbf{d}_i\right) \odot \mathbf{z}_j\left(\mathbf{v}\right) \\ &= \mathbf{E}_{ij}\left(\mathbf{F}\mathbf{U}_i\mathbf{d}_i\right) \odot \left(\overline{\mathbf{F}}\mathbf{U}_i\mathbf{d}_i\right) \odot \left(\mathbf{F}\mathbf{U}_j\mathbf{v}_j\right),\end{aligned}$$

  where we have used (7.23) and (7.18). Then

$$\begin{aligned}&\mathbf{G}_1^\star \operatorname{diag}\left(\overline{\mathbf{G}_2}\mathbf{d} \odot \mathbf{G}_2\mathbf{d}\right)\mathbf{G}_1\mathbf{v} = \\ &= \begin{bmatrix} \mathbf{U}_1^\top\overline{\mathbf{F}}\sum_{i=1}^{N_R}\mathbf{E}_{i1}\left(\mathbf{F}\mathbf{U}_i\mathbf{d}_i\right) \odot \left(\overline{\mathbf{F}}\mathbf{U}_i\mathbf{d}_i\right) \odot \left(\mathbf{F}\mathbf{U}_1\mathbf{v}_1\right) \\ \vdots \\ \mathbf{U}_{N_R}^\top\overline{\mathbf{F}}\sum_{i=1}^{N_R}\mathbf{E}_{iN_R}\left(\mathbf{F}\mathbf{U}_i\mathbf{d}_i\right) \odot \left(\overline{\mathbf{F}}\mathbf{U}_i\mathbf{d}_i\right) \odot \left(\mathbf{F}\mathbf{U}_{N_R}\mathbf{v}_{N_R}\right) \end{bmatrix}\end{aligned} \tag{7.31}$$

- $\mathbf{G}_2^\star \operatorname{diag}\left(\overline{\mathbf{G}_1}\mathbf{d} \odot \mathbf{G}_1\mathbf{d}\right)\mathbf{G}_2\mathbf{v}$:
  Use (7.21) and let $\mathbf{y} = \left(\overline{\mathbf{G}_1}\mathbf{d} \odot \mathbf{G}_1\mathbf{d}\right) \odot \mathbf{G}_2\mathbf{v}$, with

$$\begin{aligned}\mathbf{y}_{i,j} &= \overline{\mathbf{z}}_j\left(\mathbf{d}\right) \odot \mathbf{z}_j\left(\mathbf{d}\right) \odot \mathbf{w}_{i,j}\left(\mathbf{v}_i\right) \\ &= \mathbf{E}_{ij}\left(\overline{\mathbf{F}}\mathbf{U}_j\mathbf{d}_j\right) \odot \left(\mathbf{F}\mathbf{U}_j\mathbf{d}_j\right) \odot \left(\overline{\mathbf{F}}\mathbf{U}_i\mathbf{v}_i\right),\end{aligned}$$

where we have used (7.22) and (7.19). Then

$$\mathbf{G}_2^\star \operatorname{diag}\left(\overline{\mathbf{G}_1}\mathbf{d} \odot \mathbf{G}_1\mathbf{d}\right) \mathbf{G}_2\mathbf{v} = \\ = \begin{bmatrix} \mathbf{U}_1^\top \mathbf{F} \sum_{i=1}^{N_R} \mathbf{E}_{1i} \left(\overline{\mathbf{F}}\mathbf{U}_i\mathbf{d}_i\right) \odot \left(\mathbf{F}\mathbf{U}_i\mathbf{d}_i\right) \odot \left(\overline{\mathbf{F}}\mathbf{U}_1\mathbf{v}_1\right) \\ \vdots \\ \mathbf{U}_{N_R}^\top \mathbf{F} \sum_{i=1}^{N_R} \mathbf{E}_{N_R i} \left(\overline{\mathbf{F}}\mathbf{U}_i\mathbf{d}_i\right) \odot \left(\mathbf{F}\mathbf{U}_i\mathbf{d}_i\right) \odot \left(\overline{\mathbf{F}}\mathbf{U}_{N_R}\mathbf{v}_{N_R}\right) \end{bmatrix} \tag{7.32}$$

- $\mathbf{G}_1^\star \operatorname{diag}\left(\overline{\mathbf{G}_2}\mathbf{d} \odot \mathbf{G}_1\mathbf{d}\right) \mathbf{G}_2\mathbf{v}$:

  Use (7.20) and let $\mathbf{y} = \left(\overline{\mathbf{G}_2}\mathbf{d} \odot \mathbf{G}_1\mathbf{d}\right) \odot \mathbf{G}_2\mathbf{v}$, with

$$\begin{aligned} \mathbf{y}_{i,j} &= \overline{\mathbf{w}}_{i,j}\left(\mathbf{d}_i\right) \odot \mathbf{z}_j\left(\mathbf{d}\right) \odot \mathbf{w}_{i,j}\left(\mathbf{v}_i\right) \\ &= \mathbf{E}_{ij}\left(\mathbf{F}\mathbf{U}_i\mathbf{d}_i\right) \odot \left(\mathbf{F}\mathbf{U}_j\mathbf{d}_j\right) \odot \left(\overline{\mathbf{F}}\mathbf{U}_i\mathbf{v}_i\right), \end{aligned}$$

  where we have used (7.25) and (7.19). Then

$$\mathbf{G}_1^\star \operatorname{diag}\left(\overline{\mathbf{G}_2}\mathbf{d} \odot \mathbf{G}_1\mathbf{d}\right) \mathbf{G}_2\mathbf{v} = \\ = \begin{bmatrix} \mathbf{U}_1^\top \overline{\mathbf{F}} \sum_{i=1}^{N_R} \mathbf{E}_{i1} \left(\mathbf{F}\mathbf{U}_i\mathbf{d}_i\right) \odot \left(\mathbf{F}\mathbf{U}_1\mathbf{d}_1\right) \odot \left(\overline{\mathbf{F}}\mathbf{U}_i\mathbf{v}_i\right) \\ \vdots \\ \mathbf{U}_{N_R}^\top \overline{\mathbf{F}} \sum_{i=1}^{N_R} \mathbf{E}_{iN_R} \left(\mathbf{F}\mathbf{U}_i\mathbf{d}_i\right) \odot \left(\mathbf{F}\mathbf{U}_{N_R}\mathbf{d}_{N_R}\right) \odot \left(\overline{\mathbf{F}}\mathbf{U}_i\mathbf{v}_i\right) \end{bmatrix} \tag{7.33}$$

- $\mathbf{G}_2^\star \operatorname{diag}\left(\overline{\mathbf{G}_1}\mathbf{d} \odot \mathbf{G}_2\mathbf{d}\right) \mathbf{G}_1\mathbf{v}$:

  Use (7.21) and let $\mathbf{y} = \left(\overline{\mathbf{G}_1}\mathbf{d} \odot \mathbf{G}_2\mathbf{d}\right) \odot \mathbf{G}_1\mathbf{v}$, with

$$\begin{aligned} \mathbf{y}_{i,j} &= \overline{\mathbf{z}}_j\left(\mathbf{d}\right) \odot \mathbf{w}_{i,j}\left(\mathbf{d}_i\right) \odot \mathbf{z}_j\left(\mathbf{v}\right) \\ &= \mathbf{E}_{ij}\left(\overline{\mathbf{F}}\mathbf{U}_j\mathbf{d}_j\right) \odot \left(\overline{\mathbf{F}}\mathbf{U}_i\mathbf{d}_i\right) \odot \left(\mathbf{F}\mathbf{U}_j\mathbf{v}_j\right), \end{aligned}$$

  where we have used (7.26) and (7.18). Then

$$\mathbf{G}_2^\star \operatorname{diag}\left(\overline{\mathbf{G}_1}\mathbf{d} \odot \mathbf{G}_2\mathbf{d}\right) \mathbf{G}_1\mathbf{v} =$$

$$= \begin{bmatrix} \mathbf{U}_1^\top \mathbf{F} \sum_{i=1}^{N_R} \mathbf{E}_{1i} \left(\overline{\mathbf{F}}\mathbf{U}_i\mathbf{d}_i\right) \odot \left(\overline{\mathbf{F}}\mathbf{U}_1\mathbf{d}_1\right) \odot \left(\mathbf{F}\mathbf{U}_i\mathbf{v}_i\right) \\ \vdots \\ \mathbf{U}_{N_R}^\top \mathbf{F} \sum_{i=1}^{N_R} \mathbf{E}_{N_R i} \left(\overline{\mathbf{F}}\mathbf{U}_i\mathbf{d}_i\right) \odot \left(\overline{\mathbf{F}}\mathbf{U}_{N_R}\mathbf{d}_{N_R}\right) \odot \left(\mathbf{F}\mathbf{U}_i\mathbf{v}_i\right) \end{bmatrix} \tag{7.34}$$

- $\mathbf{G}_1^\star \operatorname{diag}\left(\mathbf{G}_2\mathbf{d} \odot \mathbf{G}_1\mathbf{d} - \mathbf{G}_2\mathbf{d}^{\text{obs}} \odot \mathbf{G}_1\mathbf{d}^{\text{obs}}\right) \overline{\mathbf{G}_2}\mathbf{v}$:

  Using (7.20), (7.19), (7.24) and (7.27):

$$\begin{aligned} \mathbf{G}_1^\star \operatorname{diag}\left(\mathbf{G}_2\mathbf{d} \odot \mathbf{G}_1\mathbf{d} - \mathbf{G}_2\mathbf{d}^{\text{obs}} \odot \mathbf{G}_1\mathbf{d}^{\text{obs}}\right) \overline{\mathbf{G}_2}\mathbf{v} &= \\ &= \mathbf{G}_1^\star \operatorname{diag}\left(\mathbf{r}\right) \overline{\mathbf{G}_2}\mathbf{v} \\ &= \mathbf{G}_1^\star \operatorname{diag}\left(\overline{\mathbf{G}_2}\mathbf{v}\right) \mathbf{r} \\ &= \begin{bmatrix} \mathbf{U}_1^\top \overline{\mathbf{F}} \sum_{i=1}^{N_R} \left(\mathbf{F}\mathbf{U}_i\mathbf{v}_i\right) \odot \mathbf{r}_{i1} \\ \vdots \\ \mathbf{U}_{N_R}^\top \overline{\mathbf{F}} \sum_{i=1}^{N_R} \left(\mathbf{F}\mathbf{U}_i\mathbf{v}_i\right) \odot \mathbf{r}_{iN_R} \end{bmatrix}. \end{aligned} \tag{7.35}$$

  Also look at (7.28).

- $\mathbf{G}_2^\star \operatorname{diag}\left(\mathbf{G}_2\mathbf{d} \odot \mathbf{G}_1\mathbf{d} - \mathbf{G}_2\mathbf{d}^{\text{obs}} \odot \mathbf{G}_1\mathbf{d}^{\text{obs}}\right) \overline{\mathbf{G}_1}\mathbf{v}$:

  Using (7.21), (7.18), (7.24) and (7.27):

$$\begin{aligned} \mathbf{G}_2^\star \operatorname{diag}\left(\mathbf{G}_1\mathbf{d} \odot \mathbf{G}_1\mathbf{d} - \mathbf{G}_1\mathbf{d}^{\text{obs}} \odot \mathbf{G}_1\mathbf{d}^{\text{obs}}\right) \overline{\mathbf{G}_2}\mathbf{v} &= \\ &= \mathbf{G}_2^\star \operatorname{diag}\left(\mathbf{r}\right) \overline{\mathbf{G}_1}\mathbf{v} \\ &= \mathbf{G}_2^\star \operatorname{diag}\left(\overline{\mathbf{G}_1}\mathbf{v}\right) \mathbf{r} \\ &= \begin{bmatrix} \mathbf{U}_1^\top \mathbf{F} \sum_{i=1}^{N_R} \left(\overline{\mathbf{F}}\mathbf{U}_i\mathbf{v}_i\right) \odot \mathbf{r}_{1i} \\ \vdots \\ \mathbf{U}_{N_R}^\top \mathbf{F} \sum_{i=1}^{N_R} \left(\overline{\mathbf{F}}\mathbf{U}_i\mathbf{v}_i\right) \odot \mathbf{r}_{N_R i} \end{bmatrix}. \end{aligned} \tag{7.36}$$

  Also look at (7.29).

We are now ready to give the procedure for computing the product of the Hessian of M with some arbitrary vector $\mathbf{v}$ of length $N_{\mathbf{d}}$, defined analogously to $\mathbf{d}$.

Algorithm 7.6. Product of the Hessian of the Interferometric Misfit

Procedure for computing the product of the Hessian $\dfrac{\partial^2 \mathrm{M}}{\partial \mathbf{d}\, \partial \mathbf{d}}$ *with an arbitrary vector* $\mathbf{v} = \begin{bmatrix} \mathbf{v}_1^\top & \cdots & \mathbf{v}_{N_R}^\top \end{bmatrix}^\top$*, where each* $\mathbf{v}_i$ *has length* $N_{\mathbf{d}_i}$*.*

For $i = 1, \cdots, N_R$*,*

- *If not already available, compute the discrete Fourier transform of each trace* $\mathbf{d}_i$*:*

$$\widehat{\mathbf{d}}_i = \texttt{fft}\left(\texttt{pad}\left(\mathbf{d}_i\right)\right)$$

  for $i = 1, \cdots, N_R$*, where* `pad` *pads* $\mathbf{d}_i$ *with* $N_\omega - N_{\mathbf{d}_i}$ *zeros. Likewise,*

$$\widehat{\mathbf{d}}_i^{\text{obs}} = \texttt{fft}\left(\texttt{pad}\left(\mathbf{d}_i^{\text{obs}}\right)\right).$$

- *Similarly, compute the discrete Fourier transform of each trace* $\mathbf{v}_i$*:*

$$\widehat{\mathbf{v}}_i = \texttt{fft}\left(\texttt{pad}\left(\mathbf{v}_i\right)\right)$$

  for $i = 1, \cdots, N_R$*.*

- *Set* $\boldsymbol{\theta}_i = \mathbf{0}_{N_{\mathbf{d}_i} \times 1}$ *for* $i = 1, \cdots, N_R$*.*

- *For* $i = 1, \cdots, N_R$*:*

  - *Set* $\boldsymbol{\phi}_1 = \mathbf{0}_{N_\omega \times 1}$ *and* $\boldsymbol{\phi}_2 = \mathbf{0}_{N_\omega \times 1}$ *(these are temporary storage arrays).*
  - *For* $j = 1, \cdots, N_R$ *compute:*

$$\boldsymbol{\phi}_1 = \boldsymbol{\phi}_1 + \mathbf{E}_{ij}\left(\widehat{\mathbf{d}}_j \odot \left(\overline{\widehat{\mathbf{d}}_j} \odot \widehat{\mathbf{v}}_i + \widehat{\mathbf{d}}_i \odot \overline{\widehat{\mathbf{v}}_j}\right) + \widehat{\mathbf{v}}_j \odot \left(\overline{\widehat{\mathbf{d}}_j} \odot \widehat{\mathbf{d}}_i - \overline{\widehat{\mathbf{d}}_j^{\text{obs}}} \odot \widehat{\mathbf{d}}_i^{\text{obs}}\right)\right)$$

and

$$\boldsymbol{\phi}_2 = \boldsymbol{\phi}_2 + \mathbf{E}_{ji}\left(\overline{\widehat{\mathbf{d}}_j} \odot \left(\overline{\widehat{\mathbf{d}}_i} \odot \widehat{\mathbf{v}}_j + \widehat{\mathbf{d}}_j \odot \overline{\widehat{\mathbf{v}}_i}\right) + \overline{\widehat{\mathbf{v}}_j} \odot \left(\overline{\widehat{\mathbf{d}}_i} \odot \widehat{\mathbf{d}}_j - \overline{\widehat{\mathbf{d}}_i^{\text{obs}}} \odot \widehat{\mathbf{d}}_j^{\text{obs}}\right)\right).$$

- *Compute*

$$\boldsymbol{\theta}_i = \Re\left(\texttt{restrict}\left(\texttt{ifft}\left(\boldsymbol{\phi}_1\right) + \frac{1}{N_\omega}\texttt{fft}\left(\boldsymbol{\phi}_2\right)\right)\right),$$

  where `restrict` *restricts the vector to the first* $N_{\mathbf{d}_i}$ *entries.*

- *Finally, set*

$$\frac{\partial^2 \mathrm{M}}{\partial \mathbf{d}\,\partial \mathbf{d}}\,\mathbf{v} = \begin{bmatrix}\boldsymbol{\theta}_1^\top & \cdots & \boldsymbol{\theta}_{N_R}^\top\end{bmatrix}^\top.$$

Chapter 8

Numerical Experiment

As an interesting and simple application of the methods derived in this thesis for the ETDRK method, we consider the parameter estimation problem applied to the following version of the fourth-order Swift-Hohenberg model on the torus $\Omega = \mathbb{T}^2 : [0, L_x) \times [0, L_y)$:

$$\frac{\partial y}{\partial t} = r\,y - \left(1 + \nabla^2\right)^2 y + g\,y^2 - y^3, \tag{8.1}$$

where $r > 0$ and g are parameter functions that determine the behaviour of the solution. After spatially discretizing in some appropriate way we have

$$\frac{\partial \mathbf{y}}{\partial t} = \operatorname{diag}(\mathbf{r})\,\mathbf{y} - \left(\mathbf{I} + \nabla_h^2\right)^2 \mathbf{y} + \operatorname{diag}(\mathbf{g})\,\mathbf{y}^2 - \mathbf{y}^3, \tag{8.2}$$

with $\mathbf{y} = \mathbf{y}(t)$, $\mathbf{r}$ and $\mathbf{g}$ being the spatial discretizations of r and g respectively, and ∇_h^2 the spatially discretized Laplace operator. The Swift-Hohenberg model is an example of a PDE whose solutions exhibit pattern formation, a phenomenon that occurs in many different branches of science, for instance biology (morphogenesis, vegetation patterns, animal markings, growth of bacterial colonies, etc.), physics (liquid crystals, nonlinear waves, Bénard cells, etc.) and chemical kinetics (e.g. the Belousov-Zhabotinsky, CIMA and PA-MBO relations). The field is enormous; see [44, 82] for some interesting applications, but there are many others. The Swift-Hohenberg model itself was derived from the equations of thermal convection [117], and it is possible to use other nonlinear terms than the one used here.

The fourth-order derivative term implies that this equation is very stiff, making it a good candidate for use with an exponential integrator. The obtained patterns

depend on the parameters $\mathbf{m} = \begin{bmatrix}\mathbf{r}^\top & \mathbf{g}^\top\end{bmatrix}^\top$. It is possible to obtain different patterns in different regions of the domain if these parameters exhibit spatial variability, which is the case that we consider here. This is therefore a distributed parameter estimation problem.

We take the linear part to be $\mathbf{L} = -\left(\mathbf{I} + \nabla_h^2\right)^2$, hence $\mathbf{n}(\mathbf{y}, \mathbf{m}, t) = \operatorname{diag}(\mathbf{r})\,\mathbf{y}(t) + \operatorname{diag}(\mathbf{g})\,\mathbf{y}(t)^2 - \mathbf{y}(t)^3$. Notice that we have included the linear term $\operatorname{diag}(\mathbf{r})\,\mathbf{y}$ in the nonlinear part of the equation: the dominant differential term is in $\mathbf{L}$ anyway.

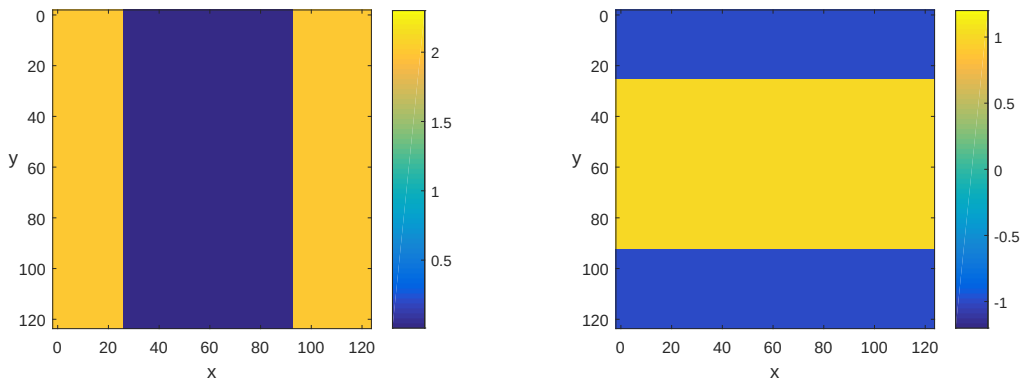
Experiment setup

Using a finite difference spatial discretization with $N_x \times N_y$ grid points, the periodic boundary conditions allow us to diagonalize the linear term using the pseudospectral method, where we compute the product with the linear part in the frequency domain and then switch back to the real domain to compute the nonlinear part. Hence

$$\frac{\partial \widehat{\mathbf{y}}(t)}{\partial t} = \widehat{\mathbf{L}}\widehat{\mathbf{y}}(t) + \mathbf{F}\left(\operatorname{diag}(\mathbf{r})\,\mathbf{F}^{-1}\widehat{\mathbf{y}}(t) + \operatorname{diag}(\mathbf{g})\left(\mathbf{F}^{-1}\widehat{\mathbf{y}}(t)\right)^2 - \left(\mathbf{F}^{-1}\widehat{\mathbf{y}}(t)\right)^3\right), \quad (8.3)$$

where $\mathbf{F}$ represents the 2D Fourier transform in space, $\mathbf{F}^{-1}$ is its inverse, $\widehat{\mathbf{y}}(t) = \mathbf{F}\mathbf{y}(t)$ and $\widehat{\mathbf{L}}$ is the diagonalized differential operator. The wave numbers on this grid are $\mathbf{k_x} = \frac{2\pi}{L_x}(-\frac{N_x}{2} : \frac{N_x}{2} - 1)$ and $\mathbf{k_y} = \frac{2\pi}{L_y}(-\frac{N_y}{2} : \frac{N_y}{2} - 1)$, so ∇_h^2 is then diagonalized with each diagonal entry the negative of the sum of the square of an element of $\mathbf{k_x}$ and the square of an element of $\mathbf{k_y}$. Now $\widehat{\mathbf{y}}(t)$ is taken to be the forward solution instead of $\mathbf{y}(t)$.

In our experiment we let $L_x = L_y = 40\pi$, $N_x = N_y = 2^7$, and the initial condition $\mathbf{y}_0$ is given by a field of Gaussian noise. We employ contour integration to evaluate the φ-functions, using a parabolic contour with 32 quadrature points. The actual parameter fields $\mathbf{r}$ and $\mathbf{g}$ are taken to be piecewise constant, as shown in Figure 8.1. The value of $\mathbf{r}$ is 2 in the outer strips and 0.04 in the inner strip. The value of $\mathbf{g}$ is -1 in the outer strips and 1 in the inner strip. The solution $\mathbf{y}(t)$ at time $t = 50s$



(a) $\mathbf{r}$ (b) $\mathbf{g}$

Figure 8.1: Actual parameter values

is shown in Figure 8.2. The outer vertical strips in the solution evolve fairly quickly relative to the central vertical strip, which evolves on a much slower time scale due to the small value of $\mathbf{r}$ there.

Derivatives of n

The adjoint solution procedure and sensitivity computations require the derivatives of

$$\mathbf{n}(\widehat{\mathbf{y}}, \mathbf{m}, t) = \mathbf{F}\left(\operatorname{diag}(\mathbf{r})\,\mathbf{F}^{-1}\widehat{\mathbf{y}}(t) + \operatorname{diag}(\mathbf{g})\left(\mathbf{F}^{-1}\widehat{\mathbf{y}}(t)\right)^2 - \left(\mathbf{F}^{-1}\widehat{\mathbf{y}}(t)\right)^3\right)$$

with respect to $\widehat{\mathbf{y}}(t)$ and $\mathbf{m}$, as well as their transposes. We have

$$\begin{aligned}
\frac{\partial \mathbf{n}(t)}{\partial \widehat{\mathbf{y}}} &= \mathbf{F}\left(\operatorname{diag}(\mathbf{r}) + 2\operatorname{diag}(\mathbf{g})\left(\mathbf{F}^{-1}\widehat{\mathbf{y}}(t)\right) - 3\left(\mathbf{F}^{-1}\widehat{\mathbf{y}}(t)\right)^2\right)\mathbf{F}^{-1} \\
\frac{\partial \mathbf{n}(t)}{\partial \mathbf{m}} &= \left[\operatorname{diag}(\widehat{\mathbf{y}}(t)) \quad \mathbf{F}\operatorname{diag}\left(\left(\mathbf{F}^{-1}\widehat{\mathbf{y}}(t)\right)^2\right)\right] \\
\frac{\partial \mathbf{n}(t)}{\partial \widehat{\mathbf{y}}}^{\top} &= \mathbf{F}^{-\top}\left(\operatorname{diag}(\mathbf{r}) + 2\operatorname{diag}(\mathbf{g})\left(\mathbf{F}^{-1}\widehat{\mathbf{y}}(t)\right) - 3\left(\mathbf{F}^{-1}\widehat{\mathbf{y}}(t)\right)^2\right)\mathbf{F}^{\top} \\
\frac{\partial \mathbf{n}(t)}{\partial \mathbf{m}}^{\top} &= \begin{bmatrix} \operatorname{diag}(\widehat{\mathbf{y}}(t)) \\ \operatorname{diag}\left(\left(\mathbf{F}^{-1}\widehat{\mathbf{y}}(t)\right)^2\right)\mathbf{F}^{\top} \end{bmatrix}.
\end{aligned}$$

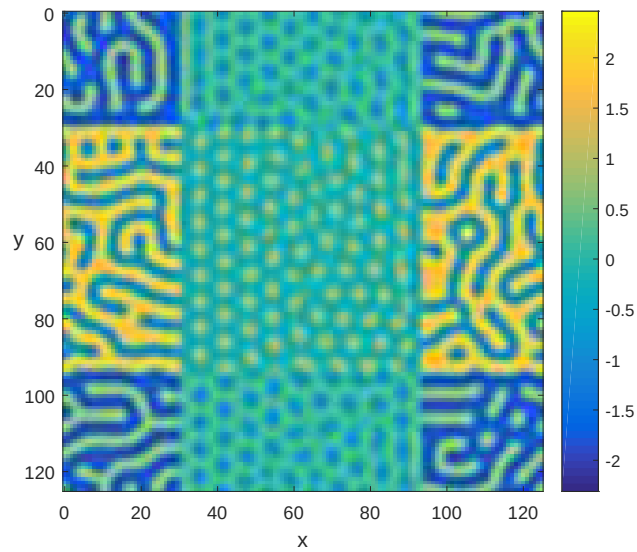


Figure 8.2: Ground truth solution at $t = 50s$.

The transposes of $\mathbf{F}$ and $\mathbf{F}^{-1}$ are $\mathbf{F}^\top = N\mathbf{F}^{-1}$ and $\mathbf{F}^{-\top} = \frac{1}{N}\mathbf{F}$, with $N = N_x N_y$, so that

$$\frac{\partial \mathbf{n}(t)}{\partial \widehat{\mathbf{y}}}^\top = \mathbf{F}\left(\operatorname{diag}(\mathbf{r}) + 2\operatorname{diag}(\mathbf{g})\left(\mathbf{F}^{-1}\widehat{\mathbf{y}}(t)\right) - 3\left(\mathbf{F}^{-1}\widehat{\mathbf{y}}(t)\right)^2\right)\mathbf{F}^{-1}$$

$$\frac{\partial \mathbf{n}(t)}{\partial \mathbf{m}}^\top = \begin{bmatrix} \operatorname{diag}(\widehat{\mathbf{y}}(t)) \\ N\operatorname{diag}\left(\left(\mathbf{F}^{-1}\widehat{\mathbf{y}}(t)\right)^2\right)\mathbf{F}^{-1} \end{bmatrix}.$$

Note that $\frac{\partial \mathbf{n}(t)}{\partial \widehat{\mathbf{y}}}^\top = \frac{\partial \mathbf{n}(t)}{\partial \widehat{\mathbf{y}}}$.

Order of Accuracy

We numerically illustrate that the adjoint solution and the gradient have the same order of accuracy as the forward solution, meaning that pth-order forward schemes are expected to lead to pth-order adjoint schemes. In this subsection we let $\mathbf{y}(n\tau)$, $\boldsymbol{\lambda}(n\tau)$ and $\nabla(n\tau)$ denote the quantity of interest one gets when performing the computations with a time-step $n\tau$, where we let $\tau = \frac{1}{80}s$. We let $\mathbf{y}_{\text{exact}}$, $\boldsymbol{\lambda}_{\text{exact}}$ and ∇_{exact} denote

	Euler	Cox-Matthews	Krogstad	Hochbruck-Ostermann
$p_{\mathbf{y}}(2\tau, \tau)$	0.9914	4.0644	4.0699	4.0275
$p_{\mathbf{y}}(4\tau, 2\tau)$	0.9908	3.9726	3.9732	3.9719
$p_{\mathbf{y}}(8\tau, 4\tau)$	–	3.9375	3.9378	3.9387
$p_{\mathbf{y}}(16\tau, 8\tau)$	–	3.8849	3.8835	3.8770

Table 8.1: Approximate order of accuracy $p_{\mathbf{y}}$ for the forward solution, computed using $p_{\mathbf{y}}(2^i\tau, 2^{i-1}\tau) = \log_2(\|\epsilon_{\mathbf{y}}(2^i\tau)\|/\|\epsilon_{\mathbf{y}}(2^{i-1}\tau)\|)$ for $i = 1, 2, 3, 4$, with $\epsilon_{\mathbf{y}}(2^i\tau) = \mathbf{y}(2^i\tau) - \mathbf{y}_{\text{exact}}$.

the "exact" values attained by performing the computations with a fine time-step $\frac{1}{160}s$ using the Krogstad scheme. The error between the computed and exact forward solution is $\epsilon_{\mathbf{y}}(n\tau) = \mathbf{y}(n\tau) - \mathbf{y}_{\text{exact}}$, and we must have $\|\epsilon_{\mathbf{y}}(n\tau)\| \approx O(n\tau)^p$ for a pth-order method, from which it follows that $(\|\epsilon_{\mathbf{y}}(2^i\tau)\|/\|\epsilon_{\mathbf{y}}(2^{i-1}\tau)\|) \approx 2^p$. Computing $p_{\mathbf{y}}(2^i\tau, 2^{i-1}\tau) := \log_2(\|\epsilon_{\mathbf{y}}(2^i\tau)\|/\|\epsilon_{\mathbf{y}}(2^{i-1}\tau)\|)$ for different values of $i = 1, 2, \ldots$ then gives an approximation of the order of accuracy p, with the estimate being more accurate for smaller values of τ and i. The results for the four ETD schemes discussed in this thesis are given in Table 8.1. We note the following:

- The simulations were run from 0s to 20s for the forward solution, and from 20s to 0s for the adjoint solution.

- The initial condition is Gaussian noise and the approximate orders of accuracy actually differ from simulation to simulation because of this. These differences are only slight for the higher-order methods, but can be quite pronounced for the lower-order Euler method. Therefore the results shown are the averages of 10 simulations using different initial conditions.

- The larger time steps are too large for the Euler method, and even the smaller time-steps can lead to inaccurate results on occasion. In computing the average order of accuracy we have therefore only included the computed $p_{\mathbf{y}}$'s that are in the interval $(p - 0.5, p + 0.5)$.

- Incidentally, for the Swift-Hohenberg equation with periodic boundary conditions,

	Euler	Cox-Matthews	Krogstad	Hochbruck-Ostermann
$p_{\boldsymbol{\lambda}}(2\tau, \tau)$	0.9976	4.0383	4.0588	3.9902
$p_{\boldsymbol{\lambda}}(4\tau, 2\tau)$	0.9434	3.9516	3.9568	3.9679
$p_{\boldsymbol{\lambda}}(8\tau, 4\tau)$	–	3.8969	3.9041	3.9343
$p_{\boldsymbol{\lambda}}(16\tau, 8\tau)$	–	3.8027	3.8100	3.8616

Table 8.2: Approximate order of accuracy $p_{\boldsymbol{\lambda}}$ for the adjoint solution, computed using $p_{\boldsymbol{\lambda}}(2^i\tau, 2^{i-1}\tau) = \log_2(\|\epsilon_{\boldsymbol{\lambda}}(2^i\tau)\|/\|\epsilon_{\boldsymbol{\lambda}}(2^{i-1}\tau)\|)$ for $i = 1, 2, 3, 4$, with $\epsilon_{\boldsymbol{\lambda}}(2^i\tau) = \boldsymbol{\lambda}(2^i\tau) - \boldsymbol{\lambda}_{\text{exact}}$.

	Euler	Cox-Matthews	Krogstad	Hochbruck-Ostermann
$p_{\nabla}(2\tau, \tau)$	1.0260	4.0430	4.0627	3.9947
$p_{\nabla}(4\tau, 2\tau)$	0.9270	3.9546	3.9575	3.9684
$p_{\nabla}(8\tau, 4\tau)$	–	3.9022	3.9056	3.9347
$p_{\nabla}(16\tau, 8\tau)$	–	3.8103	3.8136	3.8622

Table 8.3: Approximate order of accuracy p_{∇} for the gradient, computed using $p_{\nabla}(2^i\tau, 2^{i-1}\tau) = \log_2(\|\epsilon_{\nabla}(2^i\tau)\|/\|\epsilon_{\nabla}(2^{i-1}\tau)\|)$ for $i = 1, 2, 3, 4$, with $\epsilon_{\nabla}(2^i\tau) = \nabla(2^i\tau) - \nabla_{\text{exact}}$.

we see that Cox-Matthews and Krogstad schemes do indeed attain fourth order accuracy.

The quantities $p_{\boldsymbol{\lambda}}(2^i\tau, 2^{i-1}\tau)$ and $p_{\nabla}(2^i\tau, 2^{i-1}\tau)$ are defined analogously to $p_{\mathbf{y}}$ and are given in Tables 8.2 and 8.3, respectively. We have again averaged the approximate orders of accuracy from 10 simulations.

The crucial observation here is that the orders of accuracy of the adjoint ETD schemes and the resulting gradient are the same as that of the corresponding forward scheme.

Testing the Rosenbrock Approach

To test the order of accuracy of adjoint exponential Rosenbrock methods, and simulataneously check that our derivations and implementations for these methods are correct, the Swift-Hohenberg equation was reformulated by finding the Jacobian of

	Euler	Cox-Matthews	Krogstad	Hochbruck-Ostermann
$p_{\mathbf{y}}(4\tau, 2\tau)$	1.9505	4.1970	4.1967	4.1213
$p_{\mathbf{y}}(8\tau, 4\tau)$	1.9713	4.0774	4.0811	4.0561
$p_{\mathbf{y}}(16\tau, 8\tau)$	1.9846	3.7946	3.7991	3.7884

Table 8.4: Approximate order of accuracy $p_{\mathbf{y}}$ for the forward solution using a Rosenbrock approach, computed using $p_{\mathbf{y}}(2^i\tau, 2^{i-1}\tau) = \log_2(\|\epsilon_{\mathbf{y}}(2^i\tau)\|/\|\epsilon_{\mathbf{y}}(2^{i-1}\tau)\|)$ for $i = 1, 2, 3, 4$, with $\epsilon_{\mathbf{y}}(2^i\tau) = \mathbf{y}(2^i\tau) - \mathbf{y}_{\text{exact}}$.

	Euler	Cox-Matthews	Krogstad	Hochbruck-Ostermann
$p_{\boldsymbol{\lambda}}(4\tau, 2\tau)$	1.8885	4.2572	4.2482	4.2244
$p_{\boldsymbol{\lambda}}(8\tau, 4\tau)$	1.9414	4.0531	4.0498	4.0411
$p_{\boldsymbol{\lambda}}(16\tau, 8\tau)$	1.9702	3.1043	3.1029	3.0998

Table 8.5: Approximate order of accuracy $p_{\boldsymbol{\lambda}}$ for the adjoint solution using a Rosenbrock approach, computed using $p_{\boldsymbol{\lambda}}(2^i\tau, 2^{i-1}\tau) = \log_2(\|\epsilon_{\boldsymbol{\lambda}}(2^i\tau)\|/\|\epsilon_{\boldsymbol{\lambda}}(2^{i-1}\tau)\|)$ for $i = 1, 2, 3, 4$, with $\epsilon_{\boldsymbol{\lambda}}(2^i\tau) = \boldsymbol{\lambda}(2^i\tau) - \boldsymbol{\lambda}_{\text{exact}}$.

the right-hand side of (8.3),

$$\frac{\partial \mathbf{f}}{\partial \widehat{\mathbf{y}}} = \widehat{\mathbf{L}} + \mathbf{F}\left(\operatorname{diag}(\mathbf{r}) + 2\operatorname{diag}(\mathbf{g})\operatorname{diag}\left(\mathbf{F}^{-1}\widehat{\mathbf{y}}(t)\right) - 3\operatorname{diag}\left(\mathbf{F}^{-1}\widehat{\mathbf{y}}(t)\right)\right)\mathbf{F}^{-1},$$

and then setting $\mathbf{L}_k = \dfrac{\partial \mathbf{f}(\widehat{\mathbf{y}}_k)}{\partial \widehat{\mathbf{y}}}$ at the kth time-step. Consequently, with $\mathbf{y}_k = \mathbf{F}^{-1}\widehat{\mathbf{y}}_k$,

$$\mathbf{L}_k = \widehat{\mathbf{L}} + \mathbf{F}\operatorname{diag}\left(\mathbf{r} + 2\mathbf{g} \odot \mathbf{y}_k - 3\mathbf{y}_k^2\right)\mathbf{F}^{-1},$$

and hence

$$\mathbf{n}_k = \mathbf{f} - \mathbf{L}_k\widehat{\mathbf{y}}(t) = \mathbf{F}\operatorname{diag}\left(\mathbf{g} \odot \left(\mathbf{F}^{-1}\widehat{\mathbf{y}}(t) - 2\mathbf{y}_k\right) - \left(\mathbf{F}^{-1}\widehat{\mathbf{y}}(t)\right)^2 - 3\mathbf{y}_k^2\right)\mathbf{F}^{-1}\widehat{\mathbf{y}}(t).$$

The experiment from the previous subsection is repeated, but due to the significant increase in computational effort required by the additional terms in the derivatives we have run the simulations for only 2 different random initial conditions, and for a smallest time-step of 2τ, with the "exact" equations computed using a time-step of τ.

	Euler	Cox-Matthews	Krogstad	Hochbruck-Ostermann
$p_{\boldsymbol{\lambda}}(4\tau, 2\tau)$	1.8730	4.2892	4.2779	4.2580
$p_{\boldsymbol{\lambda}}(8\tau, 4\tau)$	1.9345	4.0911	4.0839	4.0764
$p_{\boldsymbol{\lambda}}(16\tau, 8\tau)$	1.9669	3.4815	3.4758	3.4754

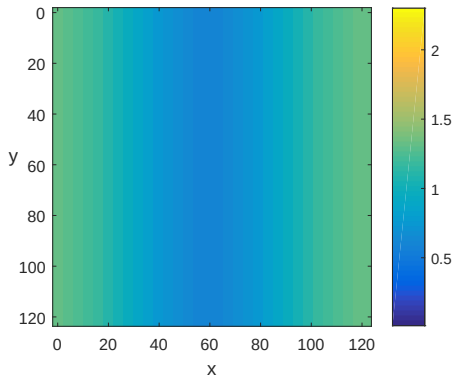
Table 8.6: Approximate order of accuracy p_∇ for the gradient using a Rosenbrock approach, computed using $p_\nabla(2^i\tau, 2^{i-1}\tau) = \log_2(\|\epsilon_\nabla(2^i\tau)\|/\|\epsilon_\nabla(2^{i-1}\tau)\|)$ for $i = 1, 2, 3, 4$, with $\epsilon_\nabla(2^i\tau) = \nabla(2^i\tau) - \nabla_{\text{exact}}$.

We have also run the simulations for just 10s instead of 20s. The results in tables 8.4-8.6 suggest that the adjoint exponential Rosenbrock method does indeed also attain the same (numerical) order of accuracy as the corresponding forward method. Oddly the Euler method seems to have an order of accuracy of around 2 instead of the expected value of 1, but this result should be viewed with reservation. The results for the largest time-step suggest that this time-step was too large, with a noticeable decrease in the order of accuracy especially for the adjoint solution.

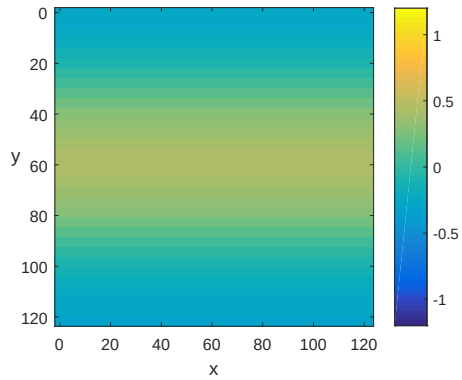
Parameter Estimation

Although outside the stated scope of this study, we show a possible parameter estimate attained using a gradient-based optimization method, where the gradient is computed using the results from this thesis. Recovering estimates that are closer to the ground truth parameters takes a more sophisticated approach than the one used here. In particular, the regularization term $\mathrm{R}(\mathbf{m}, \mathbf{m}^{\text{ref}})$ in (1.3), which is not the focus of this example, may have to be altered, or a level set method may be introduced; this is the subject of a future investigation. The initial guesses for the parameters are shown in Figures 8.3a and 8.3b, and the recovered parameters are shown in Figures 8.3c and 8.3d. Here are details of the setup:

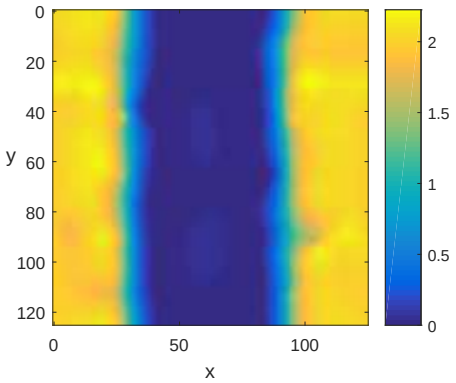
- The simulations were run using the Krogstad scheme with a time-step of 0.25s.
- Observations were taken every 0.5s up to 25s, and 5% noise was added to each observation.



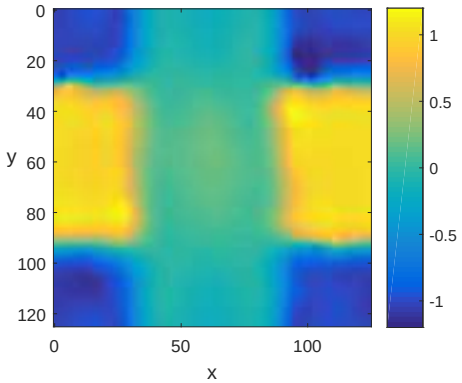
(a) Initial guess for **r**



(b) Initial guess for **g**



(c) Recovered **r**



(d) Recovered **g**

Figure 8.3: Recovered parameter values (bottom) and initial guesses (top)

- The standard least-squares misfit function $\mathrm{M} = \|\mathbf{d}(\mathbf{m}) - \mathbf{d}^{\mathrm{obs}}\|$ was employed.
- TV regularization using the smoothed Huber norm was used with $\beta = 10$.
- The parameters were recovered using 400 iterations of L-BFGS (keeping 20 previous updates in storage) with cubic line search. The results after 200 iterations were already very similar to those in Figure 8.3.
- We used a projected gradient method, where the value of $\mathbf{r}$ at each point was constrained to always lie in the interval $[0.01, 2.3]$ and the value of $\mathbf{g}$ at each point was in $[-1.2, 1.2]$.

We make the following remarks on the results:

- The recovered values of $\mathbf{r}$ are quite acceptable, as are the values of $\mathbf{g}$ on the two outside vertical strips, whereas the estimate of the central vertical strip of $\mathbf{g}$ is poor.
- This is the result of the observations having only been taken for up to 25s, a time after the patterns on the outside vertical strips have formed but before the central pattern, which evolves on a much slower time scale, has formed.
- Interestingly, including observations at later times - after the spotty pattern has formed - actually leads to parameter estimates that are qualitatively worse overall. This is connected to the fact that the central spotty pattern evolves on a much slower time-scale. Presumably it is because the later observation times contain the spotty pattern in the central region instead of the stripe-like pattern occurring in this region in the initial simulated data, which is based on the initial guess for the parameters and contains the stripe-like pattern in the central region at all times. When only a relatively small number of observations are included that have a spotty pattern there is a bias towards a non-spotty pattern in the central region, leading to the recovery above.

  Once more late observations are included, which all contain the spotty pattern, there will be a bias towards finding parameters that recreate this pattern and for

that the parameters used as an initial guess are simply too far from the true values. As a result, the minimization procedure ends up at a local minimum that is too far from the actual minimum, leading to a poor recovery.

Chapter 9

Conclusion

We have systematically derived the expressions required by the discrete adjoint method for several high-order time-stepping methods. Our results for IMEX, staggered and exponential time-stepping methods are new and can potentially play an important role in the field of large-scale model calibration problems involving nonlinear time-dependent PDEs that are solved using any of these methods. While the adjoint methods for regular LM and RK methods have already been found previously, we have provided details on their implementation and that of several associated expressions.

The discrete adjoint method was used to derive algorithms for computing the sensitivity matrix with respect to three distinct sets of parameters: the model parameters, source parameters, and the initial condition. The expression for the Hessian of the data misfit with respect to these parameters was also found. These formulas require the solution of the adjoint problem, which is solved using an adjoint time-stepping scheme corresponding to the forward scheme. The methods also require the first and, when computing the Hessian, second derivatives of the time-stepping vector, and these have been addressed in detail.

Our formulations assumed some generic data misfit M, but we also derived the derivatives of three discrete data misfit functions that are applicable to seismic imaging and related fields. In particular, the interferometric misfit is an interesting idea that might be well-suited to certain types of seismic imaging problems. We hope that having the gradient of with respect to the data available will aid in the further study of this misfit.

Not discussed in this thesis but an important contribution nonetheless is that

we have implemented the results from this study in MATLAB. We use an object-oriented approach, and the modular design allows for easy addition of new time-stepping methods and PDEs. This code will be used to apply our results to model calibration problems in a variety of applications.

One such application of interest to us is pattern formation when the model parameters exhibit spatial variability, and we have applied the techniques from this study to a simple problem involving the Swift-Hohenberg model. A more in-depth examination into parameter estimation in pattern formation problems will be the subject of future work.

A simple experiment reveals that the adjoint exponential integrator and the computed gradient of the misfit function have the same order of accuracy as the corresponding exponential integrator. This also holds for other time-stepping methods when applied to other simple problems, although we have not shown these results due to space constraints. This of course does not constitute a general proof that the order of accuracy of the adjoint time-stepping method matches that of the corresponding forward method. However, for the case when the PDE is linear we have given a simple argument for Runge-Kutta methods to show that this is indeed the case in general.

For exponential time-stepping methods we have found that if the linear operator depends on the solution $\mathbf{y}_k$ at the current time level (as is often the case with Rosenbrock-type methods) or the model parameters, the derivatives of the φ-functions with respect to $\mathbf{y}_k$ and $\mathbf{m}$ introduce significant computational overhead. The extra expense introduced in this case could be prohibitive in some applications, and it might then be more reasonable to instead apply an IMEX method to the semi-linearized PDE. The use of IMEX methods in the context of parameter estimation will be the subject of a future investigation.

The time-stepping methods we considered can broadly be classified as either linear multistep type methods or Runge-Kutta type methods. The internal stages of the RK method are needed when computing the derivatives of M, so these will have to

be stored, leading to an increase in storage overhead. For explicit RK methods the last internal stage does not need to be stored since it will not be used. LM methods do not have internal stages that need to be stored, however explicit higher-order RK methods allow for larger time steps than Adams methods so they may still lead to less required storage space overall.

Regular, IMEX, staggered and exponential time integration methods are four of the main time-stepping methods that lend themselves to high-order time-stepping, but our treatment is not exhaustive and there are of course others. General linear methods (GLMs) [14, 38, 46, 55] combine the multistage approach of RK methods with the multistep approach of LM methods. In predictor-corrector methods [16] an initial prediction step estimates the solution at a given time-step and is followed by a correction step that then refines the approximated solution. Geometric integrators [53] are an important method for solving Hamiltonian ODE systems since they preserve the structure of the phase space of the ODE. These methods can be tackled with the same techniques we have used here.

As mentioned, our results can of course also be applied to lower-order methods in the time-stepping families we have discussed. There are several other lower-order methods, usually specialized to certain types of applications, that we do not consider here, for instance Verlet integration and the Newmark-beta method. We also only consider time-stepping methods for ODE systems in first-order form, so this excludes methods designed for use with second-order ODE systems such as Beeman's algorithm and Runge-Kutta-Nyström methods.

Bibliography

[1] M. Alexe and A. Sandu. Forward and Adjoint Sensitivity Analysis with Continuous Explicit Runge-Kutta Schemes. *Applied Mathematics and Computation*, 208(2):328–334, 2009.

[2] M. Alexe and A. Sandu. On the Discrete Adjoints of Variable Step Time Integrators. *Journal of Computational and Applied Mathematics*, 233:1005–1020, 2009.

[3] M. Alexe and A. Sandu. Space-time Adaptive Solution of Inverse Problems with the Discrete Adjoint Method. *Journal of Computational Physics*, 270:21–39, 2014.

[4] T. Apel and T. G. Flaig. Crank-Nicolson schemes for optimal control problems with evolution equations. *SIAM Journal on Numerical Analysis*, 50:1484–1512, 2012.

[5] U. M. Ascher. *Numerical Methods for Evolutionary Differential Equations*. SIAM, 2008.

[6] U. M. Ascher and L. Petzold. *Computer Methods for Ordinary Differential Equations and Differential-Algebraic Equations*. SIAM, 1998.

[7] U.M. Ascher, S.J. Ruuth, and R.J. Spiteri. Implicit-Explicit Runge-Kutta methods for time-dependent partial differential equations. *Applied Numerical Mathematics*, 25:151–167, 1997.

[8] U.M. Ascher, S.J. Ruuth, and B.T.R. Wetton. Implicit-Explicit methods for time-dependent PDE's. *SIAM J. Numer. Anal.*, 32:797–823, 1995.

[9] R. C. Aster, B. Borchers, and C. H. Thurber. *Parameter Estimation and Inverse Problems*. Academic Press, 2 edition, 2012.

[10] L. Biegler, G. Biros, O. Ghattas, and M. Heinkenschloss et al., editors. *Large Scale Inverse Problems and Quantification of Uncertainty*. John Wiley & Sons, 2010.

[11] H. G. Bock, T. Carraro, W. Jäger, S. Körkel, R. Rannacher, and J. Schlöder, editors. *Model Based Parameter Estimation: Theory and Applications*. Springer, 2013.

[12] J. F. Bonnans and J. Laurent-Varin. Computation of order conditions for symplectic partitioned Runge-Kutta schemes with application to optimal control. *Numerische Mathematik*, 103:1–10, 2006.

[13] P. Brenner, M. Crouzeix, and V. Thomée. Single step methods for inhomogeneous linear differential equations in Banach space. *RAIRO Analyse Numerique*, 16:5–26, 1982.

[14] J. Butcher. A Modified Multistep Method for the Numerical Integration of Ordinary Differential Equations. *Journal of the ACM (JACM)*, (12):124–135, 1965.

[15] J. C. Butcher. *The Numerical Analysis of Ordinary Differential Equations: Runge-Kutta and General Linear Methods*. John Wiley, Chichester, UK, 1987.

[16] J. C. Butcher. *Numerical Methods for Ordinary Differential Equations*. John Wiley, Chichester, UK, 2003.

[17] M. Cara and J. J. Leveque. Waveform inversion using secondary observables. *Geophysical Research Letters*, 14(10):1046–1049, 1987.

[18] J. Certaine. The solution of ordinary differential equations with large time constants. In A. Ralston and H.S. Wilf, editors, *Mathematical Methods for Digital Computers*. Wiley, New York, 1960.

[19] G. Chavent. Identification of function parameters in partial differential equations. In R.E. Goodson and American Automatic Control Council, editors, *Identification of Parameters in Distributed Systems: Symposium, Joint Automatic Control Conference, Austin, June 1974, Papers*. American Society of Mechanical Engineers, New York, 1974.

[20] P. Chen, L. Zhao, and T. H. Jordan. Full 3D tomography for the crustal structure of the Los Angeles region. *Bulletin of the Seismological Society of America*, 97:1094–1120, 2007.

[21] I-C. Chou and E. O. Voit. Recent developments in parameter estimation and structure identification of biochemical and genomic systems. *Mathematical Biosciences*, 219:57–83, 2009.

[22] A. Cioaca, M. Alexe, and A. Sandu. Second Order Adjoints for Solving PDE-Constrained Optimization Problems. *Optimization Methods and Software*, 27:625–653, 2012.

[23] A. Cioaca and A. Sandu. An Optimization Framework to Improve 4DVar Data Assimilation System Performance. *Journal of Computational Physics*, 275:377–389, 2014.

[24] A. Cioaca, A. Sandu, and E. de Sturler. Efficient Methods for Computing Observation Impact in 4D-Var Data Assimilation. *Computational Geosciences*, 17:975–990, 2013.

[25] A. Cioaca, A. Sandu, E. de Sturler, and E. Constantinescu. Efficient computation of observation impact in 4D-Var data assimilation. In A.M. Dienstfrey

and R.E. Boisvert, editors, *Uncertainty Quantification in Scientific Computing*, IFIP Advances in Information and Communication Technology, pages 250–263. Springer, 2012.

[26] S. M. Cox and P. C. Matthews. Exponential Time Differencing for stiff systems. *Journal of Computational Physics*, 176:430–455, 2002.

[27] M. Crouzeix. Une méthode multipas implicite-explicite pour l'approximation des équations d'évolution paraboliques. *Numerische Mathematik*, 35:257–276, 1980.

[28] S. Daescu, G.R. Carmichael, and A. Sandu. Adjoint Implementation of Rosenbrock Methods Applied to Variational Data Assimilation. In S.E. Gryning and F.A. Schiermeier, editors, *Air Pollution Modeling and its Application XIV*, pages 361–369. Springer, 2000.

[29] G. Dahlquist. Convergence and stability in the numerical integration of ordinary differential equations. *Mathematica Scandinavica*, (3):33–53, 1956.

[30] V. Damian, A. Sandu, M. Damian, F. Potra, and G.R. Carmichael. The Kinetic PreProcessor KPP: A Software Environment for Solving Chemical Kinetics. *Computers and Chemical Engineering*, 26:1567–1579, 2002.

[31] J. E. Dennis and R. B. Schnabel. *Numerical Methods for Unconstrained Optimization and Nonlinear Equations*. SIAM, 1996.

[32] A. L. Dontchev, W. W. Hager, and V. M. Veliov. Second-order Runge- Kutta approximations in control constrained optimal control. *SIAM Journal on Numerical Analysis*, 38:202–226, 2000.

[33] A. M. Dziewonski, J. Mills, and S. Bloch. Residual dispersion measurement - a new method of surface wave analysis. *Bulletin of the Seismological Society of America*, 62:129–139, 1972.

[34] M. Eiermann and O. G. Ernst. A restarted Krylov subspace method for the evaluation of matrix functions. *SIAM Journal of Numerical Analysis*, 44:2481–2504, 2006.

[35] H. W. Engl, M. Hanke, and A. Neubauer. *Regularization of Inverse Problems*. Kluwer, 1996.

[36] A. Fichtner. *Full Seismic Waveform Modeling and Inversion*. Springer, 2011.

[37] B. Fornberg. High-order finite differences and the pseudospectral method on staggered grids. *SIAM Journal of Numerical Analysis*, 27:904–918, 1990.

[38] C. W. Gear. Hybrid Methods for Initial Value Problems in Ordinary Differential Equations. *Society for Industrial and Applied Mathematics*, (2):69–86, 1965.

[39] C. W. Gear. *Numerical Initial Value Problems in Ordinary Differential Equations*. Prentice-Hall, 1971.

[40] L. S. Gee and T. H. Jordan. Generalized seismological data functionals. *Geophysical Journal International*, 111:363–390, 1992.

[41] M. Ghrist and B. Fornberg. Two Results Concerning the Stability of Staggered Multistep Methods. *SIAM Journal of Numerical Analysis*, 50(4):1849–1860, 2012.

[42] M. Ghrist, B. Fornberg, and T. A. Driscoll. Staggered Time Integrators for Wave Equations. *SIAM Journal of Numerical Analysis*, 38(3):718–741, 2000.

[43] M. L. Ghrist. *High-Order Finite Difference Methods for Wave Equations*. PhD thesis, University of Colorado, Boulder, Colorado, 2000.

[44] J. P. Gollub and J. S. Langer. Pattern formation in nonequilibrium physics. *Revisions of Modern Physics*, 71:S396–S403, 1999.

[45] T. Gou, K. Singh, A. Sandu, A. Hakami, P. Percell, T. Chai, D. Byun, and J. Seinfeld. CMAQ_ADJ v4.5.4: An adjoint model for EPA's Community Multiscale Air Quality (CMAQ). `http://people.cs.vt.edu/~asandu/Software/CMAQ_ADJ`, 2008–2010.

[46] W. B. Gragg and H. J. Stetter. Generalized Multistep Predictor-Corrector Methods. *Journal of the ACM (JACM)*, (11):188–209, 1964.

[47] R. Griesse and A. Walther. Evaluating Gradients in Optimal Control: Continuous Adjoints versus Automatic Differentiation. *Journal of Optimization Theory and Applications*, 122:63–86, 2004.

[48] A. Griewank and A. Walther. *Evaluating Derivatives: Principles and Techniques of Algorithmic Differentiation*, volume 105 of *Other Titles in Applied Mathematics*. SIAM, 2nd edition, 2008.

[49] M. D. Gunzburger. *Perspectives in Flow Control and Optimization*. SIAM, 2002.

[50] S. Güttel. Rational krylov approximation of matrix functions: Numerical methods and optimal pole selection. *GAMM Mitteilungen*, 36:8–31, 2013.

[51] E. Haber. *Computational Methods in Geophysical Electromagnetics*. SIAM, 2014.

[52] W. W. Hager. Runge-Kutta methods in optimal control and the transformed adjoint system. *Numerische Mathematik*, 87:247–282, 2000.

[53] E. Hairer, C. Lubich, and G. Wanner. *Geometric Numerical Integration*. Number 31 in Springer Series in Computational Mathematics. Springer, 2002.

[54] E. Hairer, S. P. Nørsett, and G. Wanner. *Solving Ordinary Differential Equations I: Nonstiff Problems*. Springer, 2 edition, 1993.

[55] E. Hairer and G. Wanner. Multistep-multistage-multiderivative methods for ordinary differential equations. *Computing*, (11):287–303, 1973.

[56] E. Hairer and G. Wanner. *Solving Ordinary Differential Equations II: Stiff and Differential-Algebraic Problems*. Springer, 2 edition, 1996.

[57] A. Hakami, D.K. Henze, J.H. Seinfeld, K. Singh, A. Sandu, S. Kim, D. Byun, and Q. Li. The Adjoint of CMAQ. *Environmental Science and Technology*, 41:7807–7817, 2007.

[58] M. Heinkenschloss. Numerical solution of implicitly constrained optimization problems. Technical Report TR08-05, Rice University, Department of Computational and Applied Mathermatics, 2013.

[59] M. Herty, L. Pareschi, and S. Steffensen. Implicit-Explicit Runge-Kutta Schemes for Numerical Discretization of Optimal Control Problems. *SIAM Journal on Numerical Analysis*, 51:1875–1899, 2013.

[60] M. Herty and V. Schleper. Time discretizations for numerical optimization of hyperbolic problems. *App. Math. Comp.*, 218:183–194, 2011.

[61] M. Hochbruck and C. Lubich. On Krylov subspace approximations to the matrix exponential operator. *SIAM Journal of Numerical Analysis*, 34:1911–1925, 1997.

[62] M. Hochbruck and A. Ostermann. Explicit exponential RungeâĂŞ Kutta methods for semilinear parabolic problems. *SIAM Journal of Numerical Analysis*, 43:1069–1090, 2005.

[63] M. Hochbruck and A. Ostermann. Exponential Integrators. *Acta Numerica*, 19:209–286, 2010.

[64] M. Hochbruck, A. Ostermann, and J. Schweitzer. Exponential Rosenbrocktype methods. *SIAM Journal of Numerical Analysis*, 47:786–803, 2009.

[65] M. Hochbruck and J. van den Eshof. Explicit integrators of Rosenbrock-type. *Oberwolfach Reports*, 3:1107–1110, 2006.

[66] V. Isakov. *Inverse Problems for Partial Differential Equations*. Springer, 2nd edition, 2006.

[67] V. Jugnon and L. Demanet. Interferometric inversion: A robust approach to linear inverse problems. In *SEG Technical Program Expanded Abstracts 2013*, pages 5180–5184. SEG, 2013.

[68] A.-K. Kassam and L. N. Trefethen. Fourth-order time-stepping for stiff PDEs. *SIAM Journal of Scientific Computing*, 26:1214–1233, 2005.

[69] C. Y. Kaya. Inexact Restoration for Runge-Kutta Discretization of Optimal Control Problems. *SIAM Journal of Numerical Analysis*, 48:1492–1517, 2010.

[70] C. A. Kennedy and M. H. Carpenter. Additive Runge-Kutta schemes for convection-diffusion-reaction equations. *Applied Numerical Mathematics*, 44:139–181, 2003.

[71] J. B. Kool, J. C. Parker, and M. T. van Genuchten. Parameter estimation for unsaturated flow and transport models: A review. *Journal of Hydrology*, 91:255–293, 1986.

[72] S. Krogstad. Generalized integrating factor methods for stiff PDEs. *Journal of Computational Physics*, 203:72–88, 2005.

[73] J. D. Lambert. *Computational Methods in Ordinary Differential Equations*. John Wiley, 1973.

[74] J. D. Lambert. *Numerical Methods for Ordinary Differential Equations*. John Wiley, 1991.

[75] J.R. Leis and M.A. Kramer. ODESSA - an ordinary differential equation solver with explicit simultaneous sensitivity analysis. *ACM Transactions on Mathematical Software*, 14(1):61–75, 1986.

[76] A. L. Lerner-Lam and T. H. Jordan. Earth structure from fundamental and higher-mode waveform analysis. *Geophysical Journal of the Royal Astronomical Society*, 75:759–797, 1983.

[77] R. J. LeVeque. *Finite Difference Methods for Ordinary and Partial Differential Equations: Steady State and Time Dependent Problems*. SIAM, Philedelphia, 2007.

[78] V. T. Luan and A. Ostermann. Exponential B-series: The stiff case. *SIAM Journal of Numerical Analysis*, 51:3431–3445, 2013.

[79] V. T. Luan and A. Ostermann. Explicit exponential Runge-Kutta methods of high order for parabolic problems. *Journal of Computational and Applied Mathematics*, 256:168–179, 2014.

[80] V. T. Luan and A. Ostermann. Stiff order conditions for exponential Runge-Kutta methods of order five. In H. G. Bock et al., editor, *Modeling, Simulation and Optimization of Complex Processes - HPSC 2012*. Springer, 2014.

[81] Y. Luo and G. T. Schuster. Wave-equation traveltime inversion. *Geophysics*, 56:645–653, 1991.

[82] P. K. Maini, K. J. Painter, and H. N. P. Chau. Spatial pattern formation in chemical and biological systems. *Faraday Transactions*, 93:3601–3610, 1997.

[83] D. Meidner and B. Vexler. A priori error analysis of the Petrov-Galerkin Crank-Nicolson scheme for Parabolic Optimal Control Problems. *SIAM Journal on Control and Optimization*, 49:2183–2211, 2011.

[84] D. Murai and T. Koto. On Time Staggering for Wave Equations. *Journal of Computational and Applied Mathematics*, 235(14):4251–4264, 2011.

[85] R. Neidinger. Introduction to Automatic Differentiation and MATLAB Object-Oriented programming. *SIAM Review*, 52:545–563, 2010.

[86] J. Nocedal and S. J. Wright. *Numerical Optimization*. Springer, 2nd edition, 2006.

[87] D. Oldenburg, E. Haber, and R. Shekhtman. Three dimensional inversion of multi-source time domain electromagnetic data. *J. Geophysics*, 78:E47–E57, 2013.

[88] A. Ostermann and M. Roche. Runge-Kutta methods for partial differential equations and fractional orders of convergence. *Mathematics of Computing*, 59:403–420, 1992.

[89] K. M. Owolabi and K. C. Patidar. Higher-order time-stepping methods for time-dependent reactionâĂŞdiffusion equations arising in biology. *Applied Mathematics and Computation*, 240:30–50, 2014.

[90] F.-E. Plessix. A review of the adjoint-state method for computing the gradient of a functional with geophysical applications. *Geophys. J. Int.*, 167:495–503, 2006.

[91] D. A. Pope. An exponential method of numerical integration of ordinary differential equations. *Communications of the Association for Computing Machinery*, 6:491–493, 1963.

[92] V. Rao and A. Sandu. A-posteriori Error Estimates for Inverse Problems. *SIAM/ASA Journal on Uncertainty Quantification*, 3:737–761, 2015.

[93] K. Rothauge, E. Haber, and U.M. Ascher. The Discrete Adjoint Method for Exponential Integration. 2016. Submitted.

[94] K. Rothauge, E. Haber, and U.M. Ascher. The Discrete Adjoint Method for Large-Scale Optimization Problems with Linear Time-Dependent PDE Constraints. 2016. Submitted.

[95] M. P. Rumpfkeil and D. J. Mavriplis. Efficient Hessian Calculations using Automatic Differentiation and the Adjoint Method with applications. *AIAA Journal*, 48:2406–2417, 2010.

[96] S. J. Ruuth. Implicit-explicit methods for reaction-diffusion problems in pattern-formation. *J. Math. Biol.*, 34:148–176, 1995.

[97] Y. Saad. *Iterative Methods for Sparse Linear Systems*. SIAM, 2nd edition, 2003.

[98] S. Salmia and J. Toivanena. IMEX schemes for pricing options under jump-diffusion models. *Applied Numerical Mathematics*, 84:33–45, 2014.

[99] A. Sandu. Sensitivity Analysis of ODE via Automatic Differentiation. Master's thesis, University of Iowa, Iowa City, Iowa, 1997.

[100] A. Sandu. On the Properties of Runge-Kutta Discrete Adjoints. In *International Conference for Computational Science (ICCS-2006)*, volume 3994 of *Lecture Notes in Computer Science*, pages 550–557, Berlin, Germany, 2006. Springer.

[101] A. Sandu. Reverse Automatic Differentiation of Linear Multistep Methods. In C.H. Bischof, H.M. Bucker, P. Hovland, U. Naumann, and J. Utke, editors, *Advances in Automatic Differentiation*, volume 10 of *Lecture Notes in Computational Science and Engineering*, pages 1–11. Springer, 2008.

[102] A. Sandu. Solution of Inverse Problems using Discrete ODE Adjoints. In L. Biegler, G. Biros, O. Ghattas, M. Heinkenschloss, D. Keyes, B. Mallick, L. Tenorio, B. van Bloemen Waanders, and K. Willcox, editors, *Large Scale Inverse Problems and Quantification of Uncertainty*, Advances in Automatic Differentiation, chapter 16, pages 345–364. John Wiley & Sons, 2010.

[103] A. Sandu, D. Daescu, and G.R. Carmichael. Direct and Adjoint Sensitivity Analysis of Chemical Kinetic Systems with KPP: I - Theory and Software Tools. *Atmospheric Environment*, 37:5083–5096, 2003. URL: `http://dx.doi.org/10.1016/j.atmosenv.2003.08.019`.

[104] A. Sandu, J. Linford, and V. Rao. MatlODE: Matlab Integration Package for Stiff ODEs. `http://people.cs.vt.edu/~asandu/Software/MatlODE/matlode.html`, 2011–2015.

[105] A. Sandu and P. Miehe. Forward, Tangent Linear, and Adjoint Runge-Kutta Methods in KPP-2.2 for Efficient Chemical Kinetic Simulations. *International Journal of Computer Mathematics*, 87:2458–2479, 2010.

[106] A. Sandu and L. Zhang. Discrete Second Order Adjoints in Atmospheric Chemical Transport Modeling. *Journal of Computational Physics*, 227:5949–5983, 2008.

[107] J. M. Sanz-Serna. Symplectic Runge-Kutta Schemes for Adjoint Equations, Automatic Differentiation, Optimal Control and More. *SIAM Review*, 58:3–33, 2016.

[108] J. M. Sanz-Serna, J. G. Verwer, and W. H. Hundsdorfer. Convergence and order reduction of Runge-Kutta schemes applied to evolutionary problems in partial differential equations. *Numerische Mathematik*, 50:405–418, 1986.

[109] T. Schmelzer and L. N. Trefethen. Evaluating matrix functions for exponential integrators via Carathéodory-Fejér approximation and contour integral. *Electronic Transactions on Numerical Analysis*, 29:1–18, 2007.

[110] R. Serban and A.C. Hindmarsh. CVODES, the sensitivity-enabled ODE solver in SUNDIALS. Technical Report UCRL-PROC-210300, Lawrence Livermore National Laboratory, 2003.

[111] K. Sigloch, N. McQuarrie, and G. Nolet. Two-stage subduction history under north america inferred from multiple-frequency tomography. *Nature Geoscience*, 1:458–462, 2008.

[112] K. Sigloch and G. Nolet. Measuring finite-frequency body-wave amplitudes and traveltimes. *Geophysical Journal International*, 167:271–287, 2006.

[113] K. Singh, P. Eller, A. Sandu, D. Henze, K. Bowman, M. Kopacz, and M. Lee. Towards the Construction of a Standard Adjoint GEOS-Chem Model. In *Proceedings of the 2009 Spring Simulation Multiconference, High Performance Computing Symposium*, San Diego, CA, USA, 2009. Society for Computer Simulation International.

[114] K. Singh, P. Eller, A. Sandu, D. Henze, M. Kopacz, and K. Bowman. GEOS-Chem ADJOINT v7: An adjoint model for Harvard's GEOS-Chem. `http://people.cs.vt.edu/~asandu/Software/GC_AD`, 2008–2010.

[115] Z. Sirkes and E. Tziperman. Finite difference of adjoint or adjoint of finite difference? *Monthly Weather Review*, 125:3373–3378, 1997.

[116] C. V. Stewart. Robust parameter estimation in computer vision. *SIAM Review*, 41(3):513–537, 1999.

[117] J. Swift and P.C. Hohenberg. Hydrodynamic fluctuations at the convective instability. *Phys. Rev. A.*, 15:319–328, 1977.

[118] A. Tarantola. *Inverse Problem Theory and Methods for Model Estimation*. SIAM, 2004.

[119] I. M. Tibuleac, G. Nolet, C. Michaelson, and I. Koulakov. P wave amplitudes in a 3-D Earth. *Geophysical Journal International*, 155:1–10, 2003.

[120] M. Tokman. Efficient integration of large stiff systems of ODEs with exponential propagation iterative (EPI) methods. *Journal of Computational Physics*, 213(2):748–776, 2005.

[121] M. Tokman. A new class of exponential propagation iterative methods of Runge-Kutta type (EPIRK). *Journal of Computational Physics*, 230(24):8762–8778, 2011.

[122] L. N. Trefethen, J. A. C. Weideman, and T. Schmelzer. Talbot Quadratures and Rational Approximations. *BIT*, 46:653–670, 2006.

[123] K. van den Doel, U. Ascher, and E. Haber. The lost honour of ℓ_2-based regularization. *Radon Series in Computational and Applied Math*, 2013. M. Cullen, M. Freitag, S. Kindermann and R. Scheinchl (Eds).

[124] J. M. Varah. Stability restrictions on second order, three-level finite-difference schemes for parabolic equations. *SIAM J. Numer. Anal.*, 17:300–309, 1980.

[125] J. G. Verwer. On Time Staggering for Wave Equations. *Journal of Scientific Computing*, 33(2):139–154, 2007.

[126] J. G. Verwer. Time Staggering for Wave Equations Revisited. CWI-report MAS-E0710, 2007.

[127] C. R. Vogel. *Computational Methods for Inverse Problems*. SIAM, Philadelphia, 2002.

[128] A. Walther. Automatic differentiation of explicit Runge-Kutta methods for optimal control. *Computational Optimization and Applications*, 36:83–108, 2007.

[129] R. Wang, O. Yilmaz, and F. J. Herrmann. Full waveform inversion with interferometric measurements. Technical Report TR-EOAS-2014-5, 04 2014. URL: `https://www.slim.eos.ubc.ca/Publications/Public/TechReport/2014/wang2014SEGfwi/wang2014SEGfwi.html`.

[130] H. Weller, S.-J. Lock, and N. Wood. Runge-Kutta IMEX schemes for the Horizontally Explicit/Vertically Implicit (HEVI) solution of wave equations. *Journal of Computational Physics*, 252:365–381, 2013.

[131] H. Zhang. *Efficient Time Stepping Methods and Sensitivity Analysis for Large Scale Systems of Differential Equations*. PhD thesis, Virginia Polytechnic Institute and State University, Blacksburg, Virginia, 8 2014.

[132] H. Zhang and A. Sandu. FATODE: A Library for Forward, Adjoint, and Tangent Linear Integration of ODEs. Technical Report TR-11-25, Department of Computer Science, Virginia Polytechnic Institute and State University, 2011.

[133] H. Zhang and A. Sandu. FATODE: A Library for Forward, Adjoint, and Tangent Linear Integration of ODEs. `http://people.cs.vt.edu/~asandu/Software/FATODE/index.html`, 2011–2015.

[134] H. Zhang and A. Sandu. FATODE: A Library for Forward, Adjoint, and Tangent Linear Integration of ODEs. *SIAM Journal on Scientific Computing*, 36:504–523, 2014. URL: `http://dx.doi.org/10.1137/130912335`.

[135] C. Zhou, W. Cai, Y. Luo, G. Schuster, and S. Hassanzadeh. Acoustic wave-equation traveltime and waveform inversion of crosshole seismic data. *Geophysics*, 60:765–773, 1995.

Appendix A

Derivations of the Derivatives of $\mathbf{t}$

In this appendix we give the derivations of the derivatives of the time-stepping vector

$$\mathbf{t}(\overline{\mathbf{y}}, \mathbf{m}, \mathbf{y}_0) \tag{A.1}$$

with respect to the forward solution $\mathbf{y}$, the initial condition $\overline{\mathbf{y}}_0$ and the model parameters $\mathbf{m}$. We also find expressions for the products of the first derivatives, and their transposes, with appropriately-sized arbitrary vectors. These expressions constitute an important part in the calculations of the action of the sensitivity matrix or the computation of the gradient or the action of the Hessian. In this chapter $\overline{\mathbf{y}}$ is taken to be fixed with respect to $\mathbf{y}_0$ and $\mathbf{m}$.

It aid readability, if $\mathbf{x}$, $\mathbf{y}$, $\mathbf{z}$ and $\mathbf{w}$ are some vector quantities of appropriate size, we let

$$\frac{\partial^2 \mathbf{x}}{\partial \mathbf{y} \partial \mathbf{z}} \mathbf{w} = \frac{\partial}{\partial \mathbf{y}} \left(\frac{\partial \mathbf{x}}{\partial \mathbf{z}} \mathbf{w} \bigg|_{\mathbf{w}} \right),$$

At times we will use the convention

$$\mathbf{x}_{\mathbf{z}} = \frac{\partial \mathbf{x}}{\partial \mathbf{z}}.$$

A.1 Regular Time-Stepping Methods

A.1.1 Linear Multistep Methods

The time-stepping vector for a generic s-step LM method was found in (2.7) in Section 2.1.1:

$$\mathbf{t}\left(\overline{\mathbf{y}}, \mathbf{m}, \mathbf{y}_0\right) = \mathbf{A}\,\overline{\mathbf{y}} - \tau\,\mathbf{B}\,\overline{\mathbf{f}}\left(\overline{\mathbf{y}}, \mathbf{m}, \mathbf{y}_0\right) + \boldsymbol{\alpha}\,\mathbf{y}_0 - \tau\,\boldsymbol{\beta}\,\mathbf{f}_0\left(\mathbf{m}, \mathbf{y}_0\right).$$

We have now included the dependence of $\mathbf{t}$ on $\mathbf{m}$; $\overline{\mathbf{y}}$ is independent of $\mathbf{m}$ and $\mathbf{y}_0$. Recall the definitions of the matrices $\mathbf{A}$ and $\mathbf{B}$, defined in (2.1) and (2.3), and $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$, defined in (2.4). Also recall that $\overline{\mathbf{y}} = \begin{bmatrix}\mathbf{y}_1^\top & \cdots & \mathbf{y}_K^\top\end{bmatrix}^\top$. Let $\sigma = \min\left(k, s\right)$ and $k^\star = \max\left(1, k-s\right)$. In this subsection we also let $\overline{\mathbf{w}} = \begin{bmatrix}\mathbf{w}_1^\top & \cdots & \mathbf{w}_K^\top\end{bmatrix}^\top$ and $\overline{\mathbf{v}} = \begin{bmatrix}\mathbf{v}_1^\top & \cdots & \mathbf{v}_K^\top\end{bmatrix}^\top$ be arbitrary vectors of length KN, $\mathbf{w}_\mathbf{m}$ and $\mathbf{v}_\mathbf{m}$ arbitrary vectors of length $N_\mathbf{m}$, and $\mathbf{w}_0$ and $\mathbf{v}_0$ arbitrary vectors of length N.

Let us now take the derivative of $\mathbf{t} = \mathbf{t}\left(\overline{\mathbf{y}}, \mathbf{m}, \mathbf{y}_0\right)$ with respect to $\overline{\mathbf{y}}$, $\mathbf{m}$ and $\mathbf{y}_0$ in turn:

- $\dfrac{\partial \mathbf{t}}{\partial \overline{\mathbf{y}}}$

  We have

$$\begin{aligned}\frac{\partial \mathbf{t}}{\partial \overline{\mathbf{y}}} &= \begin{bmatrix}\dfrac{\partial \mathbf{t}}{\partial \mathbf{y}_1} & \dfrac{\partial \mathbf{t}}{\partial \mathbf{y}_2} & \cdots & \dfrac{\partial \mathbf{t}}{\partial \mathbf{y}_K}\end{bmatrix} \\ &= \mathbf{A} - \tau\,\mathbf{B}\begin{bmatrix}\dfrac{\partial \mathbf{f}_1}{\partial \mathbf{y}} & & \\ & \ddots & \\ & & \dfrac{\partial \mathbf{f}_K}{\partial \mathbf{y}}\end{bmatrix}.\end{aligned} \tag{A.2}$$

  It is easy to see that $\dfrac{\partial \mathbf{t}}{\partial \overline{\mathbf{y}}}$ has the form of (2.1) and (2.3), with the (k, j) $N \times N$ block being

$$\frac{\partial \mathbf{t}_k}{\partial \mathbf{y}_j} = \begin{cases}\alpha_{k-j}^{(\sigma)}\mathbf{I}_N - \tau\,\beta_{k-j}^{(\sigma)}\dfrac{\partial \mathbf{f}_j}{\partial \mathbf{y}} & \text{if } k^\star \le j \le k \\ \mathbf{0}_{N\times N} & \text{otherwise.}\end{cases} \tag{A.3}$$

- The product of $\dfrac{\partial \mathbf{t}_k}{\partial \overline{\mathbf{y}}}$ with $\overline{\mathbf{w}}$ is

$$\frac{\partial \mathbf{t}_k}{\partial \overline{\mathbf{y}}}\overline{\mathbf{w}} = \sum_{j=k^\star}^{k} \alpha_{k-j}^{(\sigma)} \mathbf{w}_j - \tau \sum_{j=k^\star}^{k} \beta_{k-j}^{(\sigma)} \frac{\partial \mathbf{f}_j}{\partial \mathbf{y}} \mathbf{w}_j. \tag{A.4}$$

- The product of $\dfrac{\partial \mathbf{t}_k}{\partial \mathbf{y}_j}^\top$ with $\mathbf{w}_k$ is

$$\frac{\partial \mathbf{t}_k}{\partial \mathbf{y}_j}^\top \mathbf{w}_k = \alpha_{k-j}^{(\sigma)} \mathbf{w}_k - \tau \beta_{k-j}^{(\sigma)} \frac{\partial \mathbf{f}_j}{\partial \mathbf{y}}^\top \mathbf{w}_k \tag{A.5}$$

  and therefore the product of $\dfrac{\partial \mathbf{t}}{\partial \mathbf{y}_j}^\top$ with $\overline{\mathbf{w}}$ is

$$\frac{\partial \mathbf{t}}{\partial \mathbf{y}_j}^\top \overline{\mathbf{w}} = \sum_{k=j}^{\min(K,j+s)} \alpha_{k-j}^{(\sigma)} \mathbf{w}_k - \tau \sum_{k=j}^{\min(K,j+s)} \beta_{k-j}^{(\sigma)} \frac{\partial \mathbf{f}_j}{\partial \mathbf{y}}^\top \mathbf{w}_k. \tag{A.6}$$

- $\dfrac{\partial \mathbf{t}}{\partial \mathbf{m}}$

  Differentiating $\mathbf{t}$ with respect to $\mathbf{m}$ gives

$$\frac{\partial \mathbf{t}}{\partial \mathbf{m}} = \frac{\partial \mathbf{t}}{\partial \mathbf{m}} = -\tau \left(\mathbf{B} \frac{\partial \overline{\mathbf{f}}}{\partial \mathbf{m}} + \boldsymbol{\beta} \frac{\partial \mathbf{f}_0}{\partial \mathbf{m}} \right) \tag{A.7}$$

  with

$$\frac{\partial \overline{\mathbf{f}}}{\partial \mathbf{m}} = \begin{bmatrix} \dfrac{\partial \mathbf{f}_1(\mathbf{m})}{\partial \mathbf{m}} \\ \vdots \\ \dfrac{\partial \mathbf{f}_K(\mathbf{m})}{\partial \mathbf{m}} \end{bmatrix}. \tag{A.8}$$

  Therefore the kth $N \times N_{\mathbf{m}}$ block of (A.7) is

$$\frac{\partial \mathbf{t}_k}{\partial \mathbf{m}} = -\tau \sum_{j=k^\star}^{k} \beta_{k-j}^{(\sigma)} \frac{\partial \mathbf{f}_j}{\partial \mathbf{m}} - \tau \beta_k^{(k)} \frac{\partial \mathbf{f}_0}{\partial \mathbf{m}} \tag{A.9}$$

with $\beta_k^{(k)} = 0$ if $k > s$.

- The product of (A.9) with $\mathbf{w_m}$ is

$$\frac{\partial \mathbf{t}_k}{\partial \mathbf{m}} \mathbf{w_m} = -\tau \sum_{j=k^\star}^{k} \beta_{k-j}^{(\sigma)} \frac{\partial \mathbf{f}_j}{\partial \mathbf{m}} \mathbf{w_m} - \tau \, \beta_k^{(k)} \frac{\partial \mathbf{f}_0}{\partial \mathbf{m}} \mathbf{w_m}. \tag{A.10}$$

- The product of the transpose with the k the block of $\mathbf{w}$ is

$$\frac{\partial \mathbf{t}_k}{\partial \mathbf{m}}^\top \mathbf{w}_k = -\tau \sum_{j=k^\star}^{k} \beta_{k-j}^{(\sigma)} \frac{\partial \mathbf{f}_j}{\partial \mathbf{m}}^\top \mathbf{w}_k - \tau \, \beta_k^{(k)} \frac{\partial \mathbf{f}_0}{\partial \mathbf{m}}^\top \mathbf{w}_k. \tag{A.11}$$

The product of the transpose of $\dfrac{\partial \mathbf{t}_k}{\partial \mathbf{m}}$ with $\overline{\mathbf{w}}$ hence is

$$\frac{\partial \mathbf{t}}{\partial \mathbf{m}}^\top \mathbf{w} = -\tau \sum_{k=1}^{K} \sum_{j=k^\star}^{k} \beta_{k-j}^{(\sigma)} \frac{\partial \mathbf{f}_j}{\partial \mathbf{m}}^\top \mathbf{w}_k - \tau \frac{\partial \mathbf{f}_0}{\partial \mathbf{m}}^\top \sum_{k=1}^{s} \beta_k^{(k)} \mathbf{w}_k. \tag{A.12}$$

- $\dfrac{\partial \mathbf{t}}{\partial \mathbf{y}_0}$

This derivative $\mathbf{t}$ with respect to $\mathbf{y}_0$ is

$$\frac{\partial \mathbf{t}}{\partial \mathbf{y}_0} = \frac{\partial \mathbf{t}}{\partial \mathbf{y}_0} = \boldsymbol{\alpha} - \tau \boldsymbol{\beta} \frac{\partial \mathbf{f}_0}{\partial \mathbf{y}} \tag{A.13}$$

and the kth $N \times N$ block is

$$\frac{\partial \mathbf{t}_k}{\partial \mathbf{y}_0} = \alpha_k^{(k)} \mathbf{I}_N - \tau \, \beta_k^{(k)} \frac{\partial \mathbf{f}_0}{\partial \mathbf{y}}, \tag{A.14}$$

with $\alpha_k^{(k)} = \beta_k^{(k)} = 0$ if $k > s$.

- The product of (A.14) with $\mathbf{w}_0$ is

$$\frac{\partial \mathbf{t}_k}{\partial \mathbf{y}_0} \mathbf{w}_0 = \alpha_k^{(k)} \mathbf{w}_0 - \tau \, \beta_k^{(k)} \frac{\partial \mathbf{f}_0}{\partial \mathbf{y}} \mathbf{w}_0. \tag{A.15}$$

- The product of the transpose with the kth block of $\mathbf{w}$ is

$$\frac{\partial \mathbf{t}_k}{\partial \mathbf{y}_0}^\top \mathbf{w}_k = \alpha_k^{(k)} \mathbf{w}_k - \tau \beta_k^{(k)} \frac{\partial \mathbf{f}_0}{\partial \mathbf{y}}^\top \mathbf{w}_k. \tag{A.16}$$

The product of the transpose of $\dfrac{\partial \mathbf{t}}{\partial \mathbf{y}_0}$ with $\mathbf{w}$ therefore is

$$\frac{\partial \mathbf{t}}{\partial \mathbf{y}_0}^\top \mathbf{w} = \sum_{k=1}^{s} \alpha_k^{(k)} \mathbf{w}_k - \tau \frac{\partial \mathbf{f}_0}{\partial \mathbf{y}}^\top \sum_{k=1}^{s} \beta_k^{(k)} \mathbf{w}_k. \tag{A.17}$$

Derivatives of $\dfrac{\partial \mathbf{t}}{\partial \overline{\mathbf{y}}}$

We will now find expressions for the derivatives of (A.4)

$$\frac{\partial \mathbf{t}_k}{\partial \overline{\mathbf{y}}} \overline{\mathbf{w}} = \sum_{j=k}^{k^\star} \alpha_{k-j}^{(\sigma)} \mathbf{w}_j - \tau \sum_{j=k}^{k^\star} \beta_{k-j}^{(\sigma)} \frac{\partial \mathbf{f}_j}{\partial \mathbf{y}} \mathbf{w}_j$$

with respect to $\mathbf{m}$, $\mathbf{y}_0$ and $\overline{\mathbf{y}}$.

- $\dfrac{\partial^2 \mathbf{t}}{\partial \mathbf{m} \partial \overline{\mathbf{y}}} \overline{\mathbf{w}}$

Differentiating (A.4) with respect to $\mathbf{m}$ gives the kth $N \times N_{\mathbf{m}}$ block of $\dfrac{\partial^2 \mathbf{t}}{\partial \mathbf{m} \partial \overline{\mathbf{y}}} \overline{\mathbf{w}}$:

$$\frac{\partial^2 \mathbf{t}_k}{\partial \mathbf{m} \partial \overline{\mathbf{y}}} \overline{\mathbf{w}} = -\tau \sum_{j=k^\star}^{k} \beta_{k-j}^{(\sigma)} \frac{\partial^2 \mathbf{f}_j}{\partial \mathbf{m} \partial \mathbf{y}} \mathbf{w}_j. \tag{A.18}$$

- The product of (A.18) with $\mathbf{v}_{\mathbf{m}}$ is

$$\left(\frac{\partial^2 \mathbf{t}_k}{\partial \mathbf{m} \partial \overline{\mathbf{y}}} \overline{\mathbf{w}} \right) \mathbf{v}_{\mathbf{m}} = -\tau \sum_{j=k^\star}^{k} \beta_{k-j}^{(\sigma)} \left(\frac{\partial^2 \mathbf{f}_j}{\partial \mathbf{m} \partial \mathbf{y}} \mathbf{w}_j \right) \mathbf{v}_{\mathbf{m}}. \tag{A.19}$$

- The product of its transpose with the kth block of $\overline{\mathbf{v}}$ is

$$\left(\frac{\partial^2 \mathbf{t}_k}{\partial \mathbf{m} \partial \overline{\mathbf{y}}} \overline{\mathbf{w}}\right)^\top \mathbf{v}_k = -\tau \sum_{j=k^\star}^{k} \beta_{k-j}^{(\sigma)} \left(\frac{\partial^2 \mathbf{f}_j}{\partial \mathbf{m} \partial \mathbf{y}} \mathbf{w}_j\right)^\top \mathbf{v}_k. \tag{A.20}$$

- $\dfrac{\partial^2 \mathbf{t}}{\partial \mathbf{y}_0 \partial \overline{\mathbf{y}}} \overline{\mathbf{w}}$

The derivative of (A.4) with respect to $\mathbf{y}_0$ is

$$\frac{\partial^2 \mathbf{t}_k}{\partial \mathbf{y}_0 \partial \overline{\mathbf{y}}} \overline{\mathbf{w}} = \mathbf{0}_{N \times N}. \tag{A.21}$$

- $\dfrac{\partial^2 \mathbf{t}}{\partial \overline{\mathbf{y}} \partial \overline{\mathbf{y}}} \overline{\mathbf{w}}$

Consider the jth $N \times N$ block of

$$\frac{\partial^2 \mathbf{t}_k}{\partial \overline{\mathbf{y}} \partial \overline{\mathbf{y}}} \overline{\mathbf{w}} = \begin{bmatrix} \frac{\partial^2 \mathbf{t}_k}{\partial \mathbf{y}_1 \partial \overline{\mathbf{y}}} \overline{\mathbf{w}} & \frac{\partial^2 \mathbf{t}_k}{\partial \mathbf{y}_2 \partial \overline{\mathbf{y}}} \overline{\mathbf{w}} & \cdots & \frac{\partial^2 \mathbf{t}_k}{\partial \mathbf{y}_K \partial \overline{\mathbf{y}}} \overline{\mathbf{w}} \end{bmatrix}.$$

Taking the derivative of (A.4) with respect to $\mathbf{y}_j$ gives the (k, j)th $N \times N$ block of $\dfrac{\partial^2 \mathbf{t}}{\partial \overline{\mathbf{y}} \partial \overline{\mathbf{y}}} \overline{\mathbf{w}}$,

$$\frac{\partial^2 \mathbf{t}_k}{\partial \mathbf{y}_j \partial \overline{\mathbf{y}}} \overline{\mathbf{w}} = \begin{cases} -\tau \beta_{k-j}^{(\sigma)} \dfrac{\partial^2 \mathbf{f}_j}{\partial \mathbf{y} \partial \mathbf{y}} \mathbf{w}_j & \text{if } k^\star \leq j \leq k \\ \mathbf{0}_{N \times N} & \text{otherwise.} \end{cases} \tag{A.22}$$

- The product of $\dfrac{\partial^2 \mathbf{t}_k}{\partial \mathbf{y}_j \partial \overline{\mathbf{y}}} \overline{\mathbf{w}}$ with the jth block of $\overline{\mathbf{v}}$ trivially is

$$\left(\frac{\partial^2 \mathbf{t}_k}{\partial \mathbf{y}_j \partial \overline{\mathbf{y}}} \overline{\mathbf{w}}\right) \mathbf{v}_j = \begin{cases} -\tau \beta_{k-j}^{(\sigma)} \left(\dfrac{\partial^2 \mathbf{f}_j}{\partial \mathbf{y} \partial \mathbf{y}} \mathbf{w}_j\right) \mathbf{v}_j & \text{if } k^\star \leq j \leq k \\ \mathbf{0}_{N \times 1} & \text{otherwise,} \end{cases} \tag{A.23}$$

so that

$$\left(\frac{\partial^2 \mathbf{t}_k}{\partial \overline{\mathbf{y}} \partial \overline{\mathbf{y}}} \overline{\mathbf{w}}\right) \overline{\mathbf{v}} = -\tau \sum_{j=k^\star}^{k} \beta_{k-j}^{(\sigma)} \left(\frac{\partial^2 \mathbf{f}_j}{\partial \mathbf{y} \partial \mathbf{y}} \mathbf{w}_j\right) \mathbf{v}_j. \tag{A.24}$$

- The product of the transpose of $\left(\frac{\partial^2 \mathbf{t}_k}{\partial \mathbf{y}_j \partial \overline{\mathbf{y}}} \overline{\mathbf{w}}\right)$ with the kth block of $\overline{\mathbf{v}}$ is

$$\left(\frac{\partial^2 \mathbf{t}_k}{\partial \mathbf{y}_j \partial \overline{\mathbf{y}}} \overline{\mathbf{w}}\right)^\top \mathbf{v}_k = \begin{cases} -\tau \beta_{k-j}^{(\sigma)} \left(\frac{\partial^2 \mathbf{f}_j}{\partial \mathbf{y} \partial \mathbf{y}} \mathbf{w}_j\right)^\top \mathbf{v}_k & \text{if } j \leq k \leq j^\star \\ \mathbf{0}_{N \times 1} & \text{otherwise.} \end{cases} \tag{A.25}$$

with $j^\star = \min(K, j+s)$.

Derivatives of $\frac{\partial \mathbf{t}}{\partial \mathbf{m}}$

The derivatives of (A.10),

$$\begin{aligned} \frac{\partial \mathbf{t}_k}{\partial \mathbf{m}} \mathbf{w}_\mathbf{m} &= -\tau \sum_{j=k^\star}^{k} \beta_{k-j}^{(\sigma)} \frac{\partial \mathbf{f}_j}{\partial \mathbf{m}} \mathbf{w}_\mathbf{m} - \tau \beta_k^{(k)} \frac{\partial \mathbf{f}_0}{\partial \mathbf{m}} \mathbf{w}_\mathbf{m} \\ &= -\tau \sum_{j=\min(0,k-s)}^{k} \beta_{k-j}^{(\sigma)} \frac{\partial \mathbf{f}_j}{\partial \mathbf{m}} \mathbf{w}_\mathbf{m}, \end{aligned}$$

with respect to $\mathbf{m}$, $\mathbf{y}_0$ and $\overline{\mathbf{y}}$ are:

- $\frac{\partial^2 \mathbf{t}}{\partial \mathbf{m} \partial \mathbf{m}} \mathbf{w}_\mathbf{m}$

  Taking the derivative of (A.10) with respect to $\mathbf{m}$ gives the kth $N \times N_\mathbf{m}$ block of $\frac{\partial^2 \mathbf{t}}{\partial \mathbf{m} \partial \mathbf{m}} \mathbf{w}_\mathbf{m}$

$$\frac{\partial^2 \mathbf{t}_k}{\partial \mathbf{m} \partial \mathbf{m}} \mathbf{w}_\mathbf{m} = -\tau \sum_{j=\min(0,k-s)}^{k} \beta_{k-j}^{(\sigma)} \frac{\partial^2 \mathbf{f}_j}{\partial \mathbf{m} \partial \mathbf{m}} \mathbf{w}_\mathbf{m}. \tag{A.26}$$

  - The product with $\mathbf{v}_\mathbf{m}$ is

$$\left(\frac{\partial^2 \mathbf{t}_k}{\partial \mathbf{m} \partial \mathbf{m}} \mathbf{w}_\mathbf{m}\right) \mathbf{v}_\mathbf{m} = -\tau \sum_{j=\min(0,k-s)}^{k} \beta_{k-j}^{(\sigma)} \left(\frac{\partial^2 \mathbf{f}_j}{\partial \mathbf{m} \partial \mathbf{m}} \mathbf{w}_\mathbf{m}\right) \mathbf{v}_\mathbf{m}. \tag{A.27}$$

- The product of its transpose with the kth block of $\overline{\mathbf{v}}$ is

$$\left(\frac{\partial^2 \mathbf{t}_k}{\partial \mathbf{m} \partial \mathbf{m}} \mathbf{w}_\mathbf{m}\right)^\top \mathbf{v}_k = -\tau \sum_{j=\min(0,k-s)}^{k} \beta_{k-j}^{(\sigma)} \left(\frac{\partial^2 \mathbf{f}_j}{\partial \mathbf{m} \partial \mathbf{m}} \mathbf{w}_\mathbf{m}\right)^\top \mathbf{v}_k. \tag{A.28}$$

- $\dfrac{\partial^2 \mathbf{t}}{\partial \mathbf{y}_0 \partial \mathbf{m}} \mathbf{w}_\mathbf{m}$

Differentiating (A.10) with respect to $\mathbf{y}_0$ immediately gives

$$\frac{\partial^2 \mathbf{t}_k}{\partial \mathbf{y}_0 \partial \mathbf{m}} \mathbf{w}_\mathbf{m} = -\tau \, \beta_k^{(k)} \frac{\partial^2 \mathbf{f}_0}{\partial \mathbf{y} \partial \mathbf{m}} \mathbf{w}_\mathbf{m} \tag{A.29}$$

with $\beta_k^{(k)} = 0$ if $k > s$.

- The product with $\mathbf{v}_0$ is

$$\left(\frac{\partial^2 \mathbf{t}_k}{\partial \mathbf{y}_0 \partial \mathbf{m}} \mathbf{w}_\mathbf{m}\right) \mathbf{v}_0 = -\tau \, \beta_k^{(k)} \left(\frac{\partial^2 \mathbf{f}_0}{\partial \mathbf{y} \partial \mathbf{m}} \mathbf{w}_\mathbf{m}\right) \mathbf{v}_0. \tag{A.30}$$

- The product of its transpose with the kth block of $\mathbf{v}_\mathbf{y}$ is

$$\left(\frac{\partial^2 \mathbf{t}_k}{\partial \mathbf{y}_0 \partial \mathbf{m}} \mathbf{w}_\mathbf{m}\right)^\top \mathbf{v}_k = -\tau \, \beta_k^{(k)} \left(\frac{\partial^2 \mathbf{f}_0}{\partial \mathbf{y} \partial \mathbf{m}} \mathbf{w}_\mathbf{m}\right)^\top \mathbf{v}_k \tag{A.31}$$

with $\beta_k^{(k)} = 0$ if $k > s$.

- $\dfrac{\partial^2 \mathbf{t}}{\partial \overline{\mathbf{y}} \partial \mathbf{m}} \mathbf{w}_\mathbf{m}$

Consider the jth $N \times N$ block of

$$\frac{\partial^2 \mathbf{t}_k}{\partial \overline{\mathbf{y}} \partial \mathbf{m}} \mathbf{w}_\mathbf{m} = \left[\frac{\partial^2 \mathbf{t}_k}{\partial \mathbf{y}_1 \partial \mathbf{m}} \mathbf{w}_\mathbf{m} \quad \frac{\partial^2 \mathbf{t}_k}{\partial \mathbf{y}_2 \partial \mathbf{m}} \mathbf{w}_\mathbf{m} \quad \cdots \quad \frac{\partial^2 \mathbf{t}_k}{\partial \mathbf{y}_K \partial \mathbf{m}} \mathbf{w}_\mathbf{m}\right]. \tag{A.32}$$

The (k,j)th $N \times N$ block of $\dfrac{\partial^2 \mathbf{t}}{\partial \overline{\mathbf{y}} \partial \mathbf{m}} \mathbf{w_m}$ is

$$\frac{\partial^2 \mathbf{t}_k}{\partial \mathbf{y}_j \partial \mathbf{m}} \mathbf{w_m} = \begin{cases} -\tau \beta_{k-j}^{(\sigma)} \dfrac{\partial^2 \mathbf{f}_j}{\partial \mathbf{y} \partial \mathbf{m}} \mathbf{m} & \text{if } k^\star \leq j \leq k \\ \mathbf{0}_{N \times N} & \text{otherwise.} \end{cases} \tag{A.33}$$

- The product of $\dfrac{\partial^2 \mathbf{t}_k}{\partial \mathbf{y}_j \partial \mathbf{m}} \mathbf{w_m}$ with the jth block of $\overline{\mathbf{v}}$ is

$$\left(\frac{\partial^2 \mathbf{t}_k}{\partial \mathbf{y}_j \partial \mathbf{m}} \mathbf{w_m} \right) \mathbf{v}_j = \begin{cases} -\tau \beta_{k-j}^{(\sigma)} \left(\dfrac{\partial^2 \mathbf{f}_j}{\partial \mathbf{y} \partial \mathbf{m}} \mathbf{m} \right) \mathbf{v}_j & \text{if } k^\star \leq j \leq k \\ \mathbf{0}_{N \times 1} & \text{otherwise.} \end{cases} \tag{A.34}$$

- The product of the transpose of $\dfrac{\partial^2 \mathbf{t}_k}{\partial \mathbf{y}_j \partial \mathbf{m}} \mathbf{w_m}$ with the kth block of $\overline{\mathbf{v}}$ is

$$\left(\frac{\partial^2 \mathbf{t}_k}{\partial \mathbf{y}_j \partial \mathbf{m}} \mathbf{w_m} \right)^\top \mathbf{v}_k = \begin{cases} -\tau \beta_{k-j}^{(\sigma)} \left(\dfrac{\partial^2 \mathbf{f}_j}{\partial \mathbf{y} \partial \mathbf{m}} \mathbf{m} \right)^\top \mathbf{v}_k & \text{if } j \leq k \leq j^\star \\ \mathbf{0}_{N \times 1} & \text{otherwise.} \end{cases} \tag{A.35}$$

with $j^\star = \min(K, j+s)$.

Derivatives of $\dfrac{\partial \mathbf{t}}{\partial \mathbf{y}_0} \mathbf{w}_0$

Finally, we find expressions for the derivatives of (A.15),

$$\frac{\partial \mathbf{t}_k}{\partial \mathbf{y}_0} \mathbf{w}_0 = \alpha_k^{(k)} \mathbf{w}_0 - \tau \, \beta_k^{(k)} \frac{\partial \mathbf{f}_0}{\partial \mathbf{y}} \mathbf{w}_0,$$

where $\alpha_k^{(k)} = \beta_k^{(k)} = 0$ if $k > s$, with respect to $\mathbf{m}$, $\mathbf{y}_0$ and $\overline{\mathbf{y}}$:

- $\dfrac{\partial^2 \mathbf{t}}{\partial \mathbf{m} \partial \mathbf{y}_0} \mathbf{w}_0$

Differentiating (A.15) with respect to $\mathbf{m}$ gives the kth $N \times N_{\mathbf{m}}$ block of $\dfrac{\partial^2 \mathbf{t}}{\partial \mathbf{m} \partial \mathbf{y}_0} \mathbf{w}_0$:

$$\frac{\partial^2 \mathbf{t}_k}{\partial \mathbf{m} \partial \mathbf{y}_0} \mathbf{w}_0 = -\tau \, \beta_k^{(k)} \frac{\partial^2 \mathbf{f}_0}{\partial \mathbf{m} \partial \mathbf{y}} \mathbf{w}_0 \tag{A.36}$$

with $\beta_k^{(k)} = 0$ if $k > s$.

- The product with $\mathbf{v}_{\mathbf{m}}$ is

$$\left(\frac{\partial^2 \mathbf{t}_k}{\partial \mathbf{m} \partial \mathbf{y}_0} \mathbf{w}_0 \right) \mathbf{v}_{\mathbf{m}} = -\tau \, \beta_k^{(k)} \left(\frac{\partial^2 \mathbf{f}_0}{\partial \mathbf{m} \partial \mathbf{y}} \mathbf{w} \right) \mathbf{v}_{\mathbf{m}}. \tag{A.37}$$

- The product of the transpose with the kth block of $\overline{\mathbf{v}}$ is

$$\left(\frac{\partial^2 \mathbf{t}_k}{\partial \mathbf{m} \partial \mathbf{y}_0} \mathbf{w}_0 \right)^\top \mathbf{v}_k = -\tau \, \beta_k^{(k)} \left(\frac{\partial^2 \mathbf{f}_0}{\partial \mathbf{m} \partial \mathbf{y}} \mathbf{w} \right)^\top \mathbf{v}_k. \tag{A.38}$$

- $\dfrac{\partial^2 \mathbf{t}}{\partial \mathbf{y}_0 \partial \mathbf{y}_0} \mathbf{w}_0$

The kth $N \times N$ block of $\dfrac{\partial^2 \mathbf{t}}{\partial \mathbf{y}_0 \partial \mathbf{y}_0} \mathbf{w}_0$ is obtained by taking the derivative of (A.15) with respect to $\mathbf{y}_0$:

$$\frac{\partial^2 \mathbf{t}_k}{\partial \mathbf{y}_0 \partial \mathbf{y}_0} \mathbf{w}_0 = -\tau \, \beta_k^{(k)} \frac{\partial^2 \mathbf{f}_0}{\partial \mathbf{y} \partial \mathbf{y}} \mathbf{w}_0 \tag{A.39}$$

with $\beta_k^{(k)} = 0$ if $k > s$.

- The product with $\mathbf{v}_{\mathbf{m}}$ is

$$\left(\frac{\partial^2 \mathbf{t}_k}{\partial \mathbf{y}_0 \partial \mathbf{y}_0} \mathbf{w}_0 \right) \mathbf{v}_0 = -\tau \, \beta_k^{(k)} \left(\frac{\partial^2 \mathbf{f}_0}{\partial \mathbf{y} \partial \mathbf{y}} \mathbf{w} \right) \mathbf{v}_0. \tag{A.40}$$

- The product of the transpose with the kth block of $\overline{\mathbf{v}}$ is

$$\left(\frac{\partial^2 \mathbf{t}_k}{\partial \mathbf{y}_0 \partial \mathbf{y}_0} \mathbf{w}_0 \right)^\top \mathbf{v}_k = -\tau \, \beta_k^{(k)} \left(\frac{\partial^2 \mathbf{f}_0}{\partial \mathbf{y} \partial \mathbf{y}} \mathbf{w} \right)^\top \mathbf{v}_k. \tag{A.41}$$

- $\dfrac{\partial^2 \mathbf{t}}{\partial \overline{\mathbf{y}} \partial \mathbf{y}_0} \mathbf{w}_0$

  Differentiating (A.15) with respect to $\overline{\mathbf{y}}$ immediately gives

$$\frac{\partial^2 \mathbf{t}_k}{\partial \overline{\mathbf{y}} \partial \mathbf{y}_0} \mathbf{w}_0 = \mathbf{0}_{N \times KN}. \tag{A.42}$$

A.1.2 Runge-Kutta Methods

In Section 2.1.2 we found the following time-stepping equation for RK methods:

$$\mathbf{t}\left(\overline{\mathbf{y}}, \mathbf{y}_0\right) = \underbrace{\begin{bmatrix} \mathbf{I}_{sN} & & & & & & \\ \mathbf{B}^\top & \mathbf{I}_N & & & & & \\ & & \mathbf{I}_{sN} & & & & \\ & -\mathbf{I}_N & \mathbf{B}^\top & \mathbf{I}_N & & & \\ & & \ddots & \ddots & \ddots & & \\ & & & & & \mathbf{I}_{sN} & \\ & & & & -\mathbf{I}_N & \mathbf{B}^\top & \mathbf{I}_N \end{bmatrix}}_{\mathbf{T}} \underbrace{\begin{bmatrix} \mathbf{Y}_1 \\ \mathbf{y}_1 \\ \mathbf{Y}_2 \\ \mathbf{y}_2 \\ \vdots \\ \mathbf{Y}_K \\ \mathbf{y}_K \end{bmatrix}}_{\overline{\mathbf{y}}} - \underbrace{\begin{bmatrix} \mathbf{F}_1 \\ \mathbf{y}_0 \\ \mathbf{F}_2 \\ \mathbf{0}_{N\times 1} \\ \vdots \\ \mathbf{F}_K \\ \mathbf{0}_{N\times 1} \end{bmatrix}}_{\overline{\mathbf{f}}\left(\overline{\mathbf{y}}, \mathbf{y}_0\right)},$$

where $\mathbf{B} = -\tau_k \mathbf{b}_s \otimes \mathbf{I}_N \in \mathbb{R}^{sN \times N}$,

$$\mathbf{Y}_k = \begin{bmatrix} \mathbf{Y}_{k,1} \\ \vdots \\ \mathbf{Y}_{k,s} \end{bmatrix} \quad \text{and} \quad \mathbf{F}_k = \begin{bmatrix} \mathbf{F}_{k,1} \\ \vdots \\ \mathbf{F}_{k,s} \end{bmatrix},$$

with $\mathbf{F}_{k,\sigma} = \mathbf{F}_{k,\sigma}\left(\overline{\mathbf{y}}, \mathbf{y}_0\right) = \mathbf{f}\left(\underline{\mathbf{y}}_{k,\sigma}, t_{k-1} + c_\sigma \tau_k\right)$. We have let $\underline{\mathbf{y}}_{k,\sigma} := \mathbf{y}_{k-1} + \tau_k \sum_{i=1}^{s} a_{\sigma,i} \mathbf{Y}_{k,i}$. Note that $\mathbf{T} \in \mathbb{R}^{(s+1)KN \times (s+1)KN}$ and $\mathbf{t}, \overline{\mathbf{f}}, \overline{\mathbf{y}} \in \mathbb{R}^{(s+1)KN \times 1}$. The kth time-step is represented by

$$\mathbf{t}_k = \begin{bmatrix} \mathbf{Y}_k - \mathbf{F}_k \\ -\mathbf{y}_{k-1} + \mathbf{B}^\top \mathbf{Y}_k + \mathbf{y}_k \end{bmatrix}.$$

We let $\widehat{\mathbf{y}}_k = \begin{bmatrix} \mathbf{Y}_k^\top & \mathbf{y}_k^\top \end{bmatrix}^\top$. We also let $\overline{\mathbf{w}} = \begin{bmatrix} \widehat{\mathbf{w}}_1^\top & \cdots & \widehat{\mathbf{w}}_K^\top \end{bmatrix}^\top$ and $\overline{\mathbf{v}} = \begin{bmatrix} \widehat{\mathbf{v}}_1^\top & \cdots & \widehat{\mathbf{v}}_K^\top \end{bmatrix}^\top$ be arbitrary vectors of length $(s+1)KN$, where $\widehat{\mathbf{w}}_k^\top$ and $\widehat{\mathbf{v}}_k^\top$ are defined analogously to $\widehat{\mathbf{y}}_k^\top$. Also, $\mathbf{w_m}$ and $\mathbf{v_m}$ are arbitrary vectors of length $N_\mathbf{m}$, and $\mathbf{w}_0$ and $\mathbf{v}_0$ are arbitrary vectors of length N.

We will now take the derivative of $\mathbf{t}$ with respect to each of $\overline{\mathbf{y}}$, $\mathbf{m}$ and $\mathbf{y}_0$ in turn. For this we need the derivatives of $\underline{\mathbf{y}}_{k,\sigma}$ with respect to $\mathbf{y}_{k-1}$ and $\mathbf{Y}_{k,i}$, which are $\dfrac{\partial \underline{\mathbf{y}}_{k,\sigma}}{\partial \mathbf{y}_{k-1}} = \mathbf{I}_N$ and $\dfrac{\partial \underline{\mathbf{y}}_{k,\sigma}}{\partial \mathbf{Y}_{k,i}} = \tau_k a_{\sigma i} \mathbf{I}_N$.

- $\dfrac{\partial \mathbf{t}}{\partial \overline{\mathbf{y}}}$

  We break the derivative of $\mathbf{t}$ with respect to $\overline{\mathbf{y}}$ into parts:

$$\frac{\partial \mathbf{t}}{\partial \overline{\mathbf{y}}} = \begin{bmatrix} \dfrac{\partial \mathbf{t}}{\partial \widehat{\mathbf{y}}_1} & \dfrac{\partial \mathbf{t}}{\partial \widehat{\mathbf{y}}_2} & \cdots & \dfrac{\partial \mathbf{t}}{\partial \widehat{\mathbf{y}}_K} \end{bmatrix} = \mathbf{T} - \begin{bmatrix} \dfrac{\partial \overline{\mathbf{f}}}{\partial \widehat{\mathbf{y}}_1} & \dfrac{\partial \overline{\mathbf{f}}}{\partial \widehat{\mathbf{y}}_2} & \cdots & \dfrac{\partial \overline{\mathbf{f}}}{\partial \widehat{\mathbf{y}}_K} \end{bmatrix}. \tag{A.43}$$

  If

$$\frac{\partial \mathbf{F}_k}{\partial \widehat{\mathbf{y}}_j} = \begin{bmatrix} \dfrac{\partial \mathbf{F}_k}{\partial \mathbf{Y}_{j,1}} & \cdots & \dfrac{\partial \mathbf{F}_k}{\partial \mathbf{Y}_{j,s}} & \dfrac{\partial \mathbf{F}_k}{\partial \mathbf{y}_j} \end{bmatrix}$$

  denotes the derivative of $\mathbf{F}_k$ with respect to $\widehat{\mathbf{y}}_j$, with $1 \le k, j \le K$, and $\dfrac{\partial \mathbf{F}_{k,\sigma}}{\partial \widehat{\mathbf{y}}_{j,i}}$ denotes the (σ, i)th $N \times N$ block of $\dfrac{\partial \mathbf{F}_k}{\partial \widehat{\mathbf{y}}_j}$, with $1 \le \sigma \le s$ and $1 \le i \le s+1$, then, using the chain rule,

$$\frac{\partial \mathbf{F}_{k,\sigma}}{\partial \widehat{\mathbf{y}}_j} = \begin{cases} \tau_k \dfrac{\partial \mathbf{F}_{k,\sigma}}{\partial \mathbf{y}} \left(\begin{bmatrix} a_{\sigma,1} & \cdots & a_{\sigma,s} & 0 \end{bmatrix} \right) \otimes \mathbf{I}_N & \text{if } j = k \\ \dfrac{\partial \mathbf{F}_{k,\sigma}}{\partial \mathbf{y}} \left(\begin{bmatrix} 0 & \cdots & 0 & 1 \end{bmatrix} \right) \otimes \mathbf{I}_N & \text{if } j = k-1 \\ \mathbf{0}_{N \times (s+1)N} & \text{otherwise.} \end{cases} \tag{A.44}$$

Letting

$$\frac{\partial \mathbf{F}_k}{\partial \mathbf{y}} = \begin{bmatrix} \frac{\partial \mathbf{F}_{k,1}}{\partial \mathbf{y}} & & \\ & \ddots & \\ & & \frac{\partial \mathbf{F}_{k,s}}{\partial \mathbf{y}} \end{bmatrix}, \tag{A.45}$$

the (k,j)th $(s+1)N \times (s+1)N$ block of $\frac{\partial \mathbf{t}}{\partial \overline{\mathbf{y}}}$ therefore is

$$\frac{\partial \mathbf{t}_k}{\partial \widehat{\mathbf{y}}_j} = \begin{cases} \begin{bmatrix} \mathbf{I}_{sN} - \tau_k \frac{\partial \mathbf{F}_k}{\partial \mathbf{y}} (\mathbf{A}_s \otimes \mathbf{I}_N) & \mathbf{0}_{sN \times N} \\ \mathbf{B}^\top & \mathbf{I}_N \end{bmatrix} & \text{if } k = j \\ \begin{bmatrix} \mathbf{0}_{sN \times sN} & -\frac{\partial \mathbf{F}_k}{\partial \mathbf{y}} (\mathbf{1}_s \otimes \mathbf{I}_N) \\ \mathbf{0}_{N \times sN} & -\mathbf{I}_N \end{bmatrix} & \text{if } j = k-1 \\ \mathbf{0}_{(s+1)N \times (s+1)N} & \text{otherwise.} \end{cases} \tag{A.46}$$

Defining

$$\begin{aligned} \mathbf{A}_k &:= \mathbf{I}_{sN} - \tau_k \frac{\partial \mathbf{F}_k}{\partial \mathbf{y}} (\mathbf{A}_s \otimes \mathbf{I}_N) \\ \mathbf{C}_k &:= -\frac{\partial \mathbf{F}_k}{\partial \mathbf{y}} (\mathbf{1}_s \otimes \mathbf{I}_N), \end{aligned} \tag{A.47}$$

we can then write

$$\frac{\partial \mathbf{t}}{\partial \overline{\mathbf{y}}} = \begin{bmatrix} \mathbf{A}_1 & & & & & & \\ \mathbf{B}^\top & \mathbf{I}_N & & & & & \\ & \mathbf{C}_2 & \mathbf{A}_2 & & & & \\ & -\mathbf{I}_N & \mathbf{B}^\top & \mathbf{I}_N & & & \\ & & \ddots & \ddots & \ddots & & \\ & & & & \mathbf{C}_K & \mathbf{A}_K & \\ & & & & -\mathbf{I}_N & \mathbf{B}^\top & \mathbf{I}_N \end{bmatrix}, \tag{A.48}$$

which is a lower block-triangular matrix.

- The kth $(s+1)N \times 1$ block of the product $\dfrac{\partial \mathbf{t}}{\partial \overline{\mathbf{y}}}\overline{\mathbf{w}}$ is

$$\frac{\partial \mathbf{t}_k}{\partial \overline{\mathbf{y}}}\overline{\mathbf{w}} = \begin{bmatrix} \mathbf{A}_k \mathbf{W}_k + \mathbf{C}_k \mathbf{w}_{k-1} \\ \mathbf{B}^\top \mathbf{W}_k + \mathbf{w}_k - \mathbf{w}_{k-1} \end{bmatrix} \tag{A.49}$$

with $\mathbf{w}_0 = \mathbf{0}_{N\times 1}$. If $\dfrac{\partial \mathbf{t}_{k,\sigma}}{\partial \overline{\mathbf{y}}}\overline{\mathbf{w}}$ is the σth $N \times 1$ block of $\dfrac{\partial \mathbf{t}_k}{\partial \overline{\mathbf{y}}}\overline{\mathbf{w}}$, with $1 \le \sigma \le s+1$, we have

$$\frac{\partial \mathbf{t}_{k,\sigma}}{\partial \overline{\mathbf{y}}}\overline{\mathbf{w}} = \begin{cases} \mathbf{W}_{k,\sigma} - \dfrac{\partial \mathbf{F}_{k,\sigma}}{\partial \mathbf{y}} \underline{\mathbf{w}}_{k,\sigma} & \text{if } \sigma \le s \\ -\tau_k \sum\limits_{i=1}^{s} b_i \mathbf{W}_{k,i} + \mathbf{w}_k - \mathbf{w}_{k-1} & \text{if } \sigma = s+1, \end{cases} \tag{A.50}$$

where $\underline{\mathbf{w}}_{k,\sigma} = \mathbf{w}_{k-1} + \tau_k \sum\limits_{i=1}^{s} a_{\sigma i} \mathbf{W}_{k,i}$.

- The transpose of $\dfrac{\partial \mathbf{t}}{\partial \overline{\mathbf{y}}}$ is upper block-triangular:

$$\frac{\partial \mathbf{t}}{\partial \overline{\mathbf{y}}}^\top = \begin{bmatrix} \mathbf{A}_1^\top & \mathbf{B} & & & & & \\ & \mathbf{I}_N & \mathbf{C}_2^\top & -\mathbf{I}_N & & & \\ & & \mathbf{A}_2^\top & \mathbf{B} & & & \\ & & & \ddots & \ddots & \ddots & \\ & & & & \mathbf{I}_N & \mathbf{C}_K^\top & -\mathbf{I}_N \\ & & & & & \mathbf{A}_K^\top & \mathbf{B} \\ & & & & & & \mathbf{I}_N \end{bmatrix}. \tag{A.51}$$

The kth $(s+1)N\times 1$ block of the product of $\dfrac{\partial \mathbf{t}}{\partial \overline{\mathbf{y}}}^\top$ with some arbitrary $\overline{\mathbf{w}}$ therefore is

$$\frac{\partial \mathbf{t}}{\partial \widehat{\mathbf{y}}_k}^\top \overline{\mathbf{w}} = \begin{bmatrix} \mathbf{A}_k^\top \mathbf{W}_k + \mathbf{B}\mathbf{w}_k \\ \mathbf{C}_{k+1}^\top \mathbf{W}_{k+1} + \mathbf{w}_k - \mathbf{w}_{k+1} \end{bmatrix}, \tag{A.52}$$

with $\mathbf{W}_{K+1,\sigma} = \mathbf{w}_{K+1} = \mathbf{0}_{N\times 1}$. The σth $N \times 1$ block $\dfrac{\partial \mathbf{t}}{\partial \widehat{\mathbf{y}}_{k\sigma}}^\top \overline{\mathbf{w}}$ of $\dfrac{\partial \mathbf{t}}{\partial \widehat{\mathbf{y}}_k}^\top \overline{\mathbf{w}}$, with

$1 \leq \sigma \leq s+1$, is

$$\frac{\partial \mathbf{t}}{\partial \widehat{\mathbf{y}}_{k\sigma}}^{\top} \overline{\mathbf{w}} = \begin{cases} \mathbf{W}_{k\sigma} - \tau_k \sum_{i=1}^{s} a_{i\sigma} \frac{\partial \mathbf{F}_{ki}}{\partial \mathbf{y}}^{\top} \mathbf{W}_{ki} - \tau_k b_\sigma \mathbf{w}_k & \text{if } \sigma \leq s \\ -\sum_{i=1}^{s} \frac{\partial \mathbf{F}_{ki}}{\partial \mathbf{y}}^{\top} \mathbf{W}_{k+1,i} + \mathbf{w}_k - \mathbf{w}_{k+1} & \text{if } \sigma = s+1. \end{cases} \tag{A.53}$$

- $\dfrac{\partial \mathbf{t}}{\partial \mathbf{m}}$

Taking the derivative of $\mathbf{t}$ with respect to $\mathbf{m}$,

$$\frac{\partial \mathbf{t}}{\partial \mathbf{m}} = \frac{\partial}{\partial \mathbf{m}} \left(\mathbf{T}\overline{\mathbf{y}} - \overline{\mathbf{f}} \right) = -\frac{\partial \overline{\mathbf{f}}}{\partial \mathbf{m}} \tag{A.54}$$

with

$$\frac{\partial \overline{\mathbf{f}}}{\partial \mathbf{m}} = \begin{bmatrix} \frac{\partial \mathbf{F}_1}{\partial \mathbf{m}}^{\top} & \mathbf{0}_{N_\mathbf{m} \times N} & \cdots & \frac{\partial \mathbf{F}_k}{\partial \mathbf{m}}^{\top} & \mathbf{0}_{N_\mathbf{m} \times N} \end{bmatrix}^{\top}, \tag{A.55}$$

where

$$\frac{\partial \mathbf{F}_k}{\partial \mathbf{m}} = \begin{bmatrix} \frac{\partial \mathbf{F}_{k,1}}{\partial \mathbf{m}}^{\top} & \frac{\partial \mathbf{F}_{k,2}}{\partial \mathbf{m}}^{\top} & \cdots & \frac{\partial \mathbf{F}_{k,s}}{\partial \mathbf{m}}^{\top} \end{bmatrix}^{\top}. \tag{A.56}$$

  - The product $\dfrac{\partial \mathbf{t}_k}{\partial \mathbf{m}} \mathbf{w}_\mathbf{m}$ is simply

$$\frac{\partial \mathbf{t}_k}{\partial \mathbf{m}} \mathbf{w}_\mathbf{m} = -\begin{bmatrix} \frac{\partial \mathbf{F}_k}{\partial \mathbf{m}} \mathbf{w}_\mathbf{m} \\ \mathbf{0}_{N \times 1} \end{bmatrix}. \tag{A.57}$$

  - The product $\dfrac{\partial \mathbf{t}_k}{\partial \mathbf{m}}^{\top} \widehat{\mathbf{w}}_k$ is

$$\frac{\partial \mathbf{t}_k}{\partial \mathbf{m}}^{\top} \widehat{\mathbf{w}}_k = -\sum_{\sigma=1}^{s} \frac{\partial \mathbf{F}_{k\sigma}}{\partial \mathbf{m}}^{\top} \mathbf{W}_{k\sigma}. \tag{A.58}$$

- $\dfrac{\partial \mathbf{t}}{\partial \mathbf{y}_0}$

Taking the derivative of $\mathbf{t}$ with respect to $\mathbf{y}_0$,

$$\frac{\partial \mathbf{t}}{\partial \mathbf{y}_0} = \frac{\partial}{\partial \mathbf{y}_0}\left(\mathbf{T}\overline{\mathbf{y}} - \overline{\mathbf{f}}\right) = -\frac{\partial \overline{\mathbf{f}}}{\partial \mathbf{y}_0} \tag{A.59}$$

with

$$\frac{\partial \overline{\mathbf{f}}}{\partial \mathbf{y}_0} = \begin{bmatrix} \dfrac{\partial \mathbf{F}_1}{\partial \mathbf{y}} \dfrac{\partial \underline{\mathbf{y}}_1}{\partial \mathbf{y}_0} \\ \mathbf{I}_N \\ \mathbf{0}_{(s+1)(K-1)N\times N} \end{bmatrix}, \quad \text{where} \quad \frac{\partial \underline{\mathbf{y}}_1}{\partial \mathbf{y}_0} = \begin{bmatrix} \dfrac{\partial \underline{\mathbf{y}}_{1,1}}{\partial \mathbf{y}_0} \\ \vdots \\ \dfrac{\partial \underline{\mathbf{y}}_{1,s}}{\partial \mathbf{y}_0} \end{bmatrix} = \mathbf{1}_s \otimes \mathbf{I}_N. \tag{A.60}$$

- The product $\dfrac{\partial \mathbf{t}_k}{\partial \mathbf{y}_0}\mathbf{w}_0$ is

$$\frac{\partial \mathbf{t}_k}{\partial \mathbf{y}_0}\mathbf{w}_0 = \begin{cases} -\begin{bmatrix} \dfrac{\partial \mathbf{F}_1}{\partial \mathbf{y}}\left(\mathbf{1}_s \otimes \mathbf{w}_0\right) \\ \mathbf{w}_0 \end{bmatrix} & \text{if } \ k = 1 \\ \mathbf{0}_{N\times 1} & \text{otherwise,} \end{cases}. \tag{A.61}$$

- The product $\dfrac{\partial \mathbf{t}_k}{\partial \mathbf{y}_0}^\top \widehat{\mathbf{w}}_k$ is

$$\frac{\partial \mathbf{t}_k}{\partial \mathbf{y}_0}^\top \widehat{\mathbf{w}}_k = \begin{cases} -\displaystyle\sum_{\sigma=1}^{s} \frac{\partial \mathbf{F}_{1,\sigma}}{\partial \mathbf{y}}^\top \mathbf{W}_{1,\sigma} - \mathbf{w}_1 & \text{if } \ k = 1 \\ \mathbf{0}_{N\times 1} & \text{otherwise.} \end{cases} \tag{A.62}$$

Derivatives of $\dfrac{\partial \mathbf{t}}{\partial \overline{\mathbf{y}}}$

We will now find expressions for the derivatives of the vector $\dfrac{\partial \mathbf{t}_{k,\sigma}}{\partial \overline{\mathbf{y}}}\overline{\mathbf{w}}$ (A.50),

$$\frac{\partial \mathbf{t}_{k,\sigma}}{\partial \overline{\mathbf{y}}}\overline{\mathbf{w}} = \begin{cases} \mathbf{W}_{k,\sigma} - \dfrac{\partial \mathbf{F}_{k,\sigma}}{\partial \mathbf{y}}\underline{\mathbf{w}}_{k,\sigma} & \text{if } \sigma \leq s \\ -\tau_k \sum\limits_{i=1}^{s} b_i \mathbf{W}_{k,i} + \mathbf{w}_k - \mathbf{w}_{k-1} & \text{if } \sigma = s+1, \end{cases}$$

with respect to $\mathbf{m}$, $\mathbf{y}_0$ and $\overline{\mathbf{y}}$. $\dfrac{\partial \mathbf{t}_{k,\sigma}}{\partial \overline{\mathbf{y}}}\overline{\mathbf{w}}$ is the σth $N \times 1$ block of $\dfrac{\partial \mathbf{t}_{k}}{\partial \overline{\mathbf{y}}}\overline{\mathbf{w}}$, with $1 \leq \sigma \leq s+1$, which in turn is the kth $(s+1)N \times 1$ block of $\dfrac{\partial \mathbf{t}}{\partial \overline{\mathbf{y}}}\overline{\mathbf{w}}$.

- $\dfrac{\partial^2 \mathbf{t}}{\partial \overline{\mathbf{y}}\partial \overline{\mathbf{y}}}\overline{\mathbf{w}}$

  Consider the jth term of

$$\frac{\partial^2 \mathbf{t}_k}{\partial \overline{\mathbf{y}}\partial \overline{\mathbf{y}}}\overline{\mathbf{w}} = \begin{bmatrix} \dfrac{\partial^2 \mathbf{t}_k}{\partial \widehat{\mathbf{y}}_1 \partial \overline{\mathbf{y}}}\overline{\mathbf{w}} & \dfrac{\partial^2 \mathbf{t}_k}{\partial \widehat{\mathbf{y}}_2 \partial \overline{\mathbf{y}}}\overline{\mathbf{w}} & \cdots & \dfrac{\partial^2 \mathbf{t}_k}{\partial \widehat{\mathbf{y}}_K \partial \overline{\mathbf{y}}}\overline{\mathbf{w}} \end{bmatrix}. \tag{A.63}$$

  Taking the derivative of (A.50) with respect to $\widehat{\mathbf{y}}_j$,

$$\frac{\partial^2 \mathbf{t}_{k,\sigma}}{\partial \widehat{\mathbf{y}}_j \partial \overline{\mathbf{y}}}\overline{\mathbf{w}} = \begin{cases} -\dfrac{\partial^2 \mathbf{F}_{k,\sigma}}{\partial \widehat{\mathbf{y}}_j \partial \overline{\mathbf{y}}}\underline{\mathbf{w}}_{k,\sigma} & \text{if } 1 \leq \sigma \leq s \\ \mathbf{0}_{N\times(s+1)N} & \text{if } \sigma = s+1, \end{cases} \tag{A.64}$$

  where $\underline{\mathbf{w}}_{k,\sigma} = \mathbf{w}_{k-1} + \tau_k \sum\limits_{i=1}^{s} a_{\sigma i}\mathbf{W}_{k,i}$, $\mathbf{w}_0 = \mathbf{0}_{N\times 1}$ and the term $\dfrac{\partial^2 \mathbf{F}_{k,\sigma}}{\partial \widehat{\mathbf{y}}_j \partial \mathbf{y}}\underline{\mathbf{w}}_{k,\sigma}$ is

$$\frac{\partial^2 \mathbf{F}_{k,\sigma}}{\partial \widehat{\mathbf{y}}_j \partial \mathbf{y}}\underline{\mathbf{w}}_{k,\sigma} = \begin{cases} \tau_k \dfrac{\partial^2 \mathbf{F}_{k,\sigma}}{\partial \mathbf{y}\partial \mathbf{y}}\underline{\mathbf{w}}_{k,\sigma}\left(\begin{bmatrix} a_{\sigma,1:s} & 0 \end{bmatrix} \otimes \mathbf{I}_N\right) & \text{if } j = k \\ \dfrac{\partial^2 \mathbf{F}_{k,\sigma}}{\partial \mathbf{y}\partial \mathbf{y}}\underline{\mathbf{w}}_{k,\sigma}\left(\begin{bmatrix} \mathbf{0}_{1\times s} & 1 \end{bmatrix} \otimes \mathbf{I}_N\right) & \text{if } j = k-1 \\ \mathbf{0}_{N\times(s+1)N} & \text{otherwise.} \end{cases} \tag{A.65}$$

- The product of $\dfrac{\partial^2 \mathbf{t}_k}{\partial \widehat{\mathbf{y}}_j \partial \overline{\mathbf{y}}}\overline{\mathbf{w}}$ with $\widehat{\mathbf{v}}_j$ is

$$\left(\frac{\partial^2 \mathbf{t}_k}{\partial \widehat{\mathbf{y}}_j \partial \overline{\mathbf{y}}}\overline{\mathbf{w}}\right)\widehat{\mathbf{v}}_j = \begin{bmatrix} \left(\dfrac{\partial^2 \mathbf{t}_{k,1}}{\partial \widehat{\mathbf{y}}_j \partial \overline{\mathbf{y}}}\overline{\mathbf{w}}\right)\widehat{\mathbf{v}}_j \\ \vdots \\ \left(\dfrac{\partial^2 \mathbf{t}_{k,s+1}}{\partial \widehat{\mathbf{y}}_j \partial \overline{\mathbf{y}}}\overline{\mathbf{w}}\right)\widehat{\mathbf{v}}_j \end{bmatrix}, \tag{A.66}$$

with

$$\left(\frac{\partial^2 \mathbf{t}_{k,\sigma}}{\partial \widehat{\mathbf{y}}_k \partial \overline{\mathbf{y}}}\overline{\mathbf{w}}\right)\widehat{\mathbf{v}}_k = \begin{cases} -\dfrac{\partial^2 \mathbf{F}_{k,\sigma}}{\partial \mathbf{y} \partial \mathbf{y}}\underline{\mathbf{w}}_{k,\sigma}\left(\tau_k \sum\limits_{i=1}^{s} a_{\sigma,i}\mathbf{V}_{k,i}\right) & \text{if } 1 \le \sigma \le s \\ \mathbf{0}_{N\times 1} & \text{if } \sigma = s+1, \end{cases}$$

$$\left(\frac{\partial^2 \mathbf{t}_{k,\sigma}}{\partial \widehat{\mathbf{y}}_{k-1} \partial \overline{\mathbf{y}}}\overline{\mathbf{w}}\right)\widehat{\mathbf{v}}_{k-1} = \begin{cases} -\dfrac{\partial^2 \mathbf{F}_{k,\sigma}}{\partial \mathbf{y} \partial \mathbf{y}}\underline{\mathbf{w}}_{k,\sigma}\mathbf{v}_{k-1} & \text{if } 1 \le \sigma \le s \\ \mathbf{0}_{N\times 1} & \text{if } \sigma = s+1, \end{cases}$$

$$\left(\frac{\partial^2 \mathbf{t}_{k,\sigma}}{\partial \widehat{\mathbf{y}}_j \partial \overline{\mathbf{y}}}\overline{\mathbf{w}}\right)^\top \widehat{\mathbf{v}}_j = \mathbf{0}_{N\times 1} \qquad \text{for all other values of } j,$$

where $\mathbf{v}_0 = \mathbf{0}_{N\times 1}$.

- The product of the transpose of $\dfrac{\partial^2 \mathbf{t}_k}{\partial \widehat{\mathbf{y}}_j \partial \overline{\mathbf{y}}}\overline{\mathbf{w}}$ with $\widehat{\mathbf{v}}_k$ is

$$\left(\frac{\partial^2 \mathbf{t}_k}{\partial \widehat{\mathbf{y}}_j \partial \overline{\mathbf{y}}}\overline{\mathbf{w}}\right)^\top \widehat{\mathbf{v}}_k = \begin{bmatrix} \left(\dfrac{\partial^2 \mathbf{t}_k}{\partial \widehat{\mathbf{y}}_{j,1} \partial \overline{\mathbf{y}}}\overline{\mathbf{w}}\right)^\top \widehat{\mathbf{v}}_k \\ \vdots \\ \left(\dfrac{\partial^2 \mathbf{t}_k}{\partial \widehat{\mathbf{y}}_{j,s+1} \partial \overline{\mathbf{y}}}\overline{\mathbf{w}}\right)^\top \widehat{\mathbf{v}}_k \end{bmatrix}, \tag{A.67}$$

with

$$\left(\frac{\partial^2 \mathbf{t}_j}{\partial \widehat{\mathbf{y}}_{j,\sigma} \partial \overline{\mathbf{y}}}\overline{\mathbf{w}}\right)^\top \widehat{\mathbf{v}}_j = \begin{cases} -\tau_j \sum\limits_{i=1}^{s} a_{i\sigma}\left(\dfrac{\partial^2 \mathbf{F}_{j,\sigma}}{\partial \widehat{\mathbf{y}}_j \partial \overline{\mathbf{y}}}\underline{\mathbf{w}}_{j,\sigma}\right)^\top \mathbf{V}_{j,i} & \text{if } 1 \le \sigma \le s \\ \mathbf{0}_{N\times 1} & \text{if } \sigma = s+1, \end{cases}$$

$$\left(\frac{\partial^2 \mathbf{t}_{j+1}}{\partial \widehat{\mathbf{y}}_{j,\sigma} \partial \overline{\mathbf{y}}} \overline{\mathbf{w}}\right)^\top \widehat{\mathbf{v}}_{j+1} = \begin{cases} \mathbf{0}_{N\times 1} & \text{if } 1 \le \sigma \le s \\ -\sum\limits_{i=1}^{s} \left(\frac{\partial^2 \mathbf{F}_{j+1,i}}{\partial \widehat{\mathbf{y}}_{j,\sigma} \partial \overline{\mathbf{y}}} \underline{\mathbf{w}}_{j+1,i}\right)^\top \mathbf{V}_{j+1,i} & \text{if } \sigma = s+1, \end{cases}$$

$$\left(\frac{\partial^2 \mathbf{t}_{k}}{\partial \widehat{\mathbf{y}}_{j,\sigma} \partial \overline{\mathbf{y}}} \overline{\mathbf{w}}\right)^\top \widehat{\mathbf{v}}_{k} = \mathbf{0}_{N\times 1} \qquad \text{for all other values of } k,$$

where $\mathbf{V}_{K+1,i} = \mathbf{0}_{N\times 1}$.

- $\dfrac{\partial^2 \mathbf{t}}{\partial \mathbf{m} \partial \overline{\mathbf{y}}} \overline{\mathbf{w}}$

  Taking the derivative of (A.50) with respect to $\mathbf{m}$,

$$\frac{\partial^2 \mathbf{t}_{k,\sigma}}{\partial \mathbf{m} \partial \overline{\mathbf{y}}} \overline{\mathbf{w}} = \begin{cases} -\dfrac{\partial^2 \mathbf{F}_{k,\sigma}}{\partial \mathbf{m} \partial \mathbf{y}} \underline{\mathbf{w}}_{k,\sigma} & \text{if } \sigma \le s \\ \mathbf{0}_{N\times N_{\mathbf{m}}} & \text{if } \sigma = s+1, \end{cases} \tag{A.68}$$

  where $\underline{\mathbf{w}}_{k,\sigma} = \mathbf{w}_{k-1} + \tau_k \sum\limits_{i=1}^{s} a_{\sigma i} \mathbf{W}_{k,i}$ and $\mathbf{w}_0 = \mathbf{0}_{N\times 1}$.

  - The product of $\dfrac{\partial^2 \mathbf{t}_{k}}{\partial \mathbf{m} \partial \overline{\mathbf{y}}} \overline{\mathbf{w}}$ with $\mathbf{v}_{\mathbf{m}}$ is

$$\left(\frac{\partial^2 \mathbf{t}_{k}}{\partial \mathbf{m} \partial \overline{\mathbf{y}}} \overline{\mathbf{w}}\right) \mathbf{v}_{\mathbf{m}} = \begin{bmatrix} \left(\dfrac{\partial^2 \mathbf{t}_{k,1}}{\partial \mathbf{m} \partial \overline{\mathbf{y}}} \overline{\mathbf{w}}\right) \mathbf{v}_{\mathbf{m}} \\ \vdots \\ \left(\dfrac{\partial^2 \mathbf{t}_{k,s+1}}{\partial \mathbf{m} \partial \overline{\mathbf{y}}} \overline{\mathbf{w}}\right) \mathbf{v}_{\mathbf{m}} \end{bmatrix}, \tag{A.69}$$

    with

$$\left(\frac{\partial^2 \mathbf{t}_{k,\sigma}}{\partial \mathbf{m} \partial \overline{\mathbf{y}}} \overline{\mathbf{w}}\right) \mathbf{v}_{\mathbf{m}} = \begin{cases} -\left(\dfrac{\partial^2 \mathbf{F}_{k,\sigma}}{\partial \mathbf{m} \partial \mathbf{y}} \underline{\mathbf{w}}_{k,\sigma}\right) \mathbf{v}_{\mathbf{m}} & \text{if } \sigma \le s \\ \mathbf{0}_{N\times 1} & \text{if } \sigma = s+1. \end{cases}$$

- The product of the transpose of $\dfrac{\partial^2 \mathbf{t}_k}{\partial \mathbf{m} \partial \overline{\mathbf{y}}} \overline{\mathbf{w}}$ with $\widehat{\mathbf{v}}_k$ is

$$\left(\frac{\partial^2 \mathbf{t}_k}{\partial \mathbf{m} \partial \overline{\mathbf{y}}} \overline{\mathbf{w}} \right)^\top \widehat{\mathbf{v}}_k = - \sum_{\sigma=1}^{s} \left(\frac{\partial^2 \mathbf{F}_{k,\sigma}}{\partial \mathbf{m} \partial \mathbf{y}} \underline{\mathbf{w}}_{k,\sigma} \right)^\top \mathbf{V}_{k,\sigma}. \tag{A.70}$$

- $\dfrac{\partial^2 \mathbf{t}}{\partial \mathbf{y}_0 \partial \overline{\mathbf{y}}}$

  Differentiating (A.50) with respect to $\mathbf{y}_0$,

$$\frac{\partial^2 \mathbf{t}_{1,\sigma}}{\partial \mathbf{y}_0 \partial \overline{\mathbf{y}}} \overline{\mathbf{w}} = \begin{cases} -\dfrac{\partial^2 \mathbf{F}_{1,\sigma}}{\partial \mathbf{y} \partial \mathbf{y}} \underline{\mathbf{w}}_{1,\sigma} & \text{if } 1 \leq \sigma \leq s \text{ and } k = 1 \\ \mathbf{0}_{N \times N} & \text{otherwise} \end{cases} \tag{A.71}$$

  where $\underline{\mathbf{w}}_{1,\sigma} = \tau_k \sum_{i=1}^{s} a_{\sigma i} \mathbf{W}_{1,i}$. We used the fact that

$$\frac{\partial^2 \mathbf{F}_{1,\sigma}}{\partial \mathbf{y}_0 \partial \mathbf{y}} \underline{\mathbf{w}}_{1,\sigma} = \frac{\partial^2 \mathbf{F}_{1,\sigma}}{\partial \mathbf{y} \partial \mathbf{y}} \underline{\mathbf{w}}_{1,\sigma} \frac{\partial \underline{\mathbf{y}}_1}{\partial \mathbf{y}_0} = \frac{\partial^2 \mathbf{F}_{1,\sigma}}{\partial \mathbf{y} \partial \mathbf{y}} \underline{\mathbf{w}}_{1,\sigma}.$$

Derivatives of $\dfrac{\partial \mathbf{t}}{\partial \mathbf{m}}$

Expressions for the derivatives of

$$\frac{\partial \mathbf{t}_{k,\sigma}}{\partial \mathbf{m}} \mathbf{w}_{\mathbf{m}} = \begin{cases} -\dfrac{\partial \mathbf{F}_{k,\sigma}}{\partial \mathbf{m}} \mathbf{w}_{\mathbf{m}} & \text{if } 1 \leq \sigma \leq s \\ \mathbf{0}_{N \times 1} & \text{otherwise,} \end{cases}$$

(see (A.57)) with respect to $\mathbf{m}$, $\mathbf{y}_0$ and $\overline{\mathbf{y}}$ are derived below. $\dfrac{\partial \mathbf{t}_{k,\sigma}}{\partial \mathbf{m}} \mathbf{w}_{\mathbf{m}}$ is the σth $N \times 1$ block of $\dfrac{\partial \mathbf{t}_k}{\partial \mathbf{m}} \mathbf{w}_{\mathbf{m}}$, with $1 \leq \sigma \leq s+1$, which in turn is the kth $(s+1)N \times 1$ block of $\dfrac{\partial \mathbf{t}}{\partial \mathbf{m}} \mathbf{w}_{\mathbf{m}}$.

- $\dfrac{\partial^2 \mathbf{t}}{\partial \mathbf{m} \partial \mathbf{m}} \mathbf{w}_{\mathbf{m}}$

Taking the derivative of (A.57) with respect to $\mathbf{m}$ gives

$$\frac{\partial^2 \mathbf{t}_{k,\sigma}}{\partial \mathbf{m} \partial \mathbf{m}} \mathbf{w}_{\mathbf{m}} = \begin{cases} -\dfrac{\partial^2 \mathbf{F}_{k,\sigma}}{\partial \mathbf{m} \partial \mathbf{m}} \mathbf{w}_{\mathbf{m}} & \text{if } 1 \leq \sigma \leq s \\ \mathbf{0}_{N \times 1} & \text{otherwise.} \end{cases} \tag{A.72}$$

- $\dfrac{\partial^2 \mathbf{t}}{\partial \mathbf{y}_0 \partial \mathbf{m}} \mathbf{w}_{\mathbf{m}}$

Similarly, differentiating (A.57) with respect to $\mathbf{y}_0$ gives

$$\frac{\partial^2 \mathbf{t}_{k,\sigma}}{\partial \mathbf{y}_0 \partial \mathbf{m}} \mathbf{w}_{\mathbf{m}} = \begin{cases} -\dfrac{\partial^2 \mathbf{F}_{k,\sigma}}{\partial \mathbf{y} \partial \mathbf{m}} \mathbf{w}_{\mathbf{m}} & \text{if } 1 \leq \sigma \leq s \text{ and } k = 1 \\ \mathbf{0}_{N \times 1} & \text{otherwise,} \end{cases} \tag{A.73}$$

since $\dfrac{\partial \mathbf{F}_{1,\sigma}}{\partial \mathbf{y}_0} = \dfrac{\partial \mathbf{F}_{1,\sigma}}{\partial \mathbf{y}} \dfrac{\partial \underline{\mathbf{y}}_{1,\sigma}}{\partial \mathbf{y}_0} = \dfrac{\partial \mathbf{F}_{1,\sigma}}{\partial \mathbf{y}}$.

- $\dfrac{\partial^2 \mathbf{t}}{\partial \overline{\mathbf{y}} \partial \mathbf{m}} \mathbf{w}_{\mathbf{m}}$

Differentiating (A.57) with respect to $\overline{\mathbf{y}}$ gives

$$\frac{\partial^2 \mathbf{t}_{k,\sigma}}{\partial \overline{\mathbf{y}} \partial \mathbf{m}} \mathbf{w}_{\mathbf{m}} = \left[\frac{\partial^2 \mathbf{t}_k}{\partial \widehat{\mathbf{y}}_1 \partial \mathbf{m}} \mathbf{w}_{\mathbf{m}} \quad \cdots \quad \frac{\partial^2 \mathbf{t}_k}{\partial \widehat{\mathbf{y}}_K \partial \mathbf{m}} \mathbf{w}_{\mathbf{m}} \right]. \tag{A.74}$$

The jth $(s+1)N \times N$ block of $\dfrac{\partial^2 \mathbf{t}_{k,\sigma}}{\partial \overline{\mathbf{y}} \partial \mathbf{m}} \mathbf{w}_{\mathbf{m}}$ is

$$\frac{\partial^2 \mathbf{t}_{k,\sigma}}{\partial \widehat{\mathbf{y}}_j \partial \mathbf{m}} \mathbf{w}_{\mathbf{m}} = \begin{cases} -\dfrac{\partial^2 \mathbf{F}_{k,\sigma}}{\partial \widehat{\mathbf{y}}_j \partial \mathbf{m}} \mathbf{w}_{\mathbf{m}} & \text{if } 1 \leq \sigma \leq s \\ \mathbf{0}_{N \times 1} & \text{otherwise,} \end{cases} \tag{A.75}$$

As before, using the chain rule,

$$\frac{\partial^2 \mathbf{F}_{k,\sigma}}{\partial \widehat{\mathbf{y}}_j \partial \mathbf{m}} \mathbf{w_m} = \begin{cases} \tau_k \dfrac{\partial^2 \mathbf{F}_{k,\sigma}}{\partial \mathbf{y} \partial \mathbf{m}} \mathbf{w_m} \left(\begin{bmatrix} a_{\sigma,1:s} & 0 \end{bmatrix} \otimes \mathbf{I}_N \right) & \text{if } j = k \\ \dfrac{\partial^2 \mathbf{F}_{k,\sigma}}{\partial \mathbf{y} \partial \mathbf{m}} \mathbf{w_m} \left(\begin{bmatrix} \mathbf{0}_{1\times s} & 1 \end{bmatrix} \otimes \mathbf{I}_N \right) & \text{if } j = k-1 \\ \mathbf{0}_{N\times(s+1)N} & \text{otherwise.} \end{cases} \tag{A.76}$$

Derivatives of $\dfrac{\partial \mathbf{t}}{\partial \mathbf{y}_0}$

Lastly, we find expressions for the derivatives of the product (A.61) for $k = 1$:

$$\frac{\partial \mathbf{t}_{1,\sigma}}{\partial \mathbf{y}_0} \mathbf{w}_0 = \begin{cases} -\dfrac{\partial \mathbf{F}_{1,\sigma}}{\partial \mathbf{y}} \mathbf{w}_0 & \text{if } 1 \le \sigma \le s \\ \mathbf{w}_0 & \text{otherwise.} \end{cases}$$

For $k > 1$, $\dfrac{\partial \mathbf{t}_{1,\sigma}}{\partial \mathbf{y}_0} \mathbf{w}_0 = \mathbf{0}_{N\times 1}$. With respect to $\mathbf{m}$, $\mathbf{y}_0$ and $\overline{\mathbf{y}}$ are derived below. $\dfrac{\partial \mathbf{t}_{k,\sigma}}{\partial \mathbf{y}_0} \mathbf{w}_0$ is the σth $N \times 1$ block of $\dfrac{\partial \mathbf{t}_k}{\partial \mathbf{y}_0} \mathbf{w}_0$, with $1 \le \sigma \le s+1$, which in turn is the kth $(s+1)N \times 1$ block of $\dfrac{\partial \mathbf{t}}{\partial \mathbf{y}_0} \mathbf{w}_0$.

- $\dfrac{\partial^2 \mathbf{t}}{\partial \mathbf{y}_0 \partial \mathbf{y}_0} \mathbf{w}_0$

  Differentiating (A.61) with respect to $\mathbf{y}_0$ gives

$$\frac{\partial^2 \mathbf{t}_{1,\sigma}}{\partial \mathbf{y}_0 \partial \mathbf{y}_0} \mathbf{w}_0 = \begin{cases} -\dfrac{\partial \mathbf{F}_{1,\sigma}}{\partial \mathbf{y} \partial \mathbf{y}} \mathbf{w}_0 & \text{if } 1 \le \sigma \le s \\ \mathbf{0}_{N\times N} & \text{otherwise.} \end{cases} \tag{A.77}$$

  and $\dfrac{\partial^2 \mathbf{t}_{k,\sigma}}{\partial \mathbf{y}_0 \partial \mathbf{y}_0} \mathbf{w}_0 = \mathbf{0}_{N\times N}$ for $k > 1$.

- $\dfrac{\partial^2 \mathbf{t}}{\partial \mathbf{m} \partial \mathbf{y}_0} \mathbf{w}_0$

Similarly, taking the derivative of (A.61) with respect to $\mathbf{m}$ gives

$$\frac{\partial^2 \mathbf{t}_{1,\sigma}}{\partial \mathbf{m} \partial \mathbf{y}_0} \mathbf{w}_0 = \begin{cases} -\dfrac{\partial \mathbf{F}_{1,\sigma}}{\partial \mathbf{m} \partial \mathbf{y}} \mathbf{w}_0 & \text{if } 1 \leq \sigma \leq s \\ \mathbf{0}_{N \times N_{\mathbf{m}}} & \text{otherwise.} \end{cases} \tag{A.78}$$

and $\dfrac{\partial^2 \mathbf{t}_{k,\sigma}}{\partial \mathbf{y}_0 \partial \mathbf{y}_0} \mathbf{w}_0 = \mathbf{0}_{N \times N_{\mathbf{m}}}$ for $k > 1$.

- $\dfrac{\partial^2 \mathbf{t}}{\partial \overline{\mathbf{y}} \partial \mathbf{y}_0} \mathbf{w}_0$

  Differentiating (A.61) with respect to $\overline{\mathbf{y}}$ gives

$$\frac{\partial^2 \mathbf{t}_k}{\partial \overline{\mathbf{y}} \partial \mathbf{y}_0} \mathbf{w}_0 = \begin{bmatrix} \dfrac{\partial^2 \mathbf{t}_k}{\partial \widehat{\mathbf{y}}_1 \partial \mathbf{y}_0} \mathbf{w}_0 & \cdots & \dfrac{\partial^2 \mathbf{t}_k}{\partial \widehat{\mathbf{y}}_K \partial \mathbf{y}_0} \mathbf{w}_0 \end{bmatrix}. \tag{A.79}$$

  Clearly $\dfrac{\partial^2 \mathbf{t}_k}{\partial \widehat{\mathbf{y}}_j \partial \mathbf{y}_0} \mathbf{w}_0 = \mathbf{0}_{N \times N}$ for $j > 1$. For $j = 1$ we have

$$\frac{\partial^2 \mathbf{t}_{1,\sigma}}{\partial \widehat{\mathbf{y}}_1 \partial \mathbf{y}_0} \mathbf{w}_0 = \begin{cases} \dfrac{\partial^2 \mathbf{F}_{1,\sigma}}{\partial \widehat{\mathbf{y}}_1 \partial \mathbf{y}_0} \mathbf{w}_0 & \text{if } 1 \leq \sigma \leq s. \end{cases} \tag{A.80}$$

  Using the chain rule we have

$$\frac{\partial^2 \mathbf{F}_{1,\sigma}}{\partial \widehat{\mathbf{y}}_1 \partial \mathbf{y}_0} \mathbf{w}_0 = \frac{\partial^2 \mathbf{F}_{1,\sigma}}{\partial \mathbf{y} \partial \mathbf{y}_0} \mathbf{w}_0 \left(\begin{bmatrix} a_{\sigma,1:s} & 0 \end{bmatrix} \otimes \mathbf{I}_N \right). \tag{A.81}$$

A.2 Implicit-Explicit Time-Stepping Methods

A.2.1 IMEX Linear Multistep Methods

The time-stepping vector for a generic s-step IMEX LM method was found in (2.22) in Section 2.2.1:

$$\mathbf{t}(\mathbf{y}, \mathbf{m}, \mathbf{y}_0) = \mathbf{A}\overline{\mathbf{y}} - \tau \mathbf{B} \overline{\mathbf{f}}^{\mathcal{E}} (\overline{\mathbf{y}}, \mathbf{m}) - \tau \boldsymbol{\Gamma} \overline{\mathbf{f}}^{\mathcal{I}} (\overline{\mathbf{y}}, \mathbf{m}) - \tau \boldsymbol{\beta} \mathbf{f}_0^{\mathcal{E}} - \tau \boldsymbol{\gamma} \mathbf{f}_0^{\mathcal{I}} + \boldsymbol{\alpha} \mathbf{y}_0$$

$\mathbf{t}$ is a vector of length KN and the kth time vector is given by the kth subvector of length N:

$$\mathbf{t}_k(\mathbf{y}, \mathbf{m}, \mathbf{y}_0) = \sum_{i=0}^{s} \alpha_i^{(s)} \mathbf{y}_{k-i} - \tau \sum_{i=1}^{s} \beta_i^{(s)} \mathbf{f}_{k-i}^{\mathcal{E}} - \tau \sum_{i=0}^{s} \gamma_i^{(s)} \mathbf{f}_{k-i}^{\mathcal{I}}.$$

We have now included the dependence of $\mathbf{t}$ on $\mathbf{m}$; $\overline{\mathbf{y}}$ is independent of $\mathbf{m}$ and $\mathbf{y}_0$. Recall the definitions of the matrices $\mathbf{A}$, $\mathbf{B}$ and $\mathbf{\Gamma}$, defined in (2.1) and (2.3), and $\boldsymbol{\alpha}$, $\boldsymbol{\beta}$ and $\boldsymbol{\gamma}$, defined in (2.19). Also recall that $\overline{\mathbf{y}} = \begin{bmatrix} \mathbf{y}_1^\top & \cdots & \mathbf{y}_K^\top \end{bmatrix}^\top$. Let $\sigma = \min(k, s)$ and $k^\star = \max(1, k-s)$. In this subsection we also let $\overline{\mathbf{w}} = \begin{bmatrix} \mathbf{w}_1^\top & \cdots & \mathbf{w}_K^\top \end{bmatrix}^\top$ and $\overline{\mathbf{v}} = \begin{bmatrix} \mathbf{v}_1^\top & \cdots & \mathbf{v}_K^\top \end{bmatrix}^\top$ be arbitrary vectors of length KN, $\mathbf{w}_\mathbf{m}$ and $\mathbf{v}_\mathbf{m}$ arbitrary vectors of length $N_\mathbf{m}$, and $\mathbf{w}_0$ and $\mathbf{v}_0$ arbitrary vectors of length N.

Let us now take the derivative of $\mathbf{t} = \mathbf{t}(\overline{\mathbf{y}}, \mathbf{m}, \mathbf{y}_0)$ with respect to $\overline{\mathbf{y}}$, $\mathbf{m}$ and $\mathbf{y}_0$ in turn:

- $\dfrac{\partial \mathbf{t}}{\partial \overline{\mathbf{y}}}$

  We have

$$\begin{aligned} \frac{\partial \mathbf{t}}{\partial \overline{\mathbf{y}}} &= \begin{bmatrix} \dfrac{\partial \mathbf{t}}{\partial \mathbf{y}_1} & \dfrac{\partial \mathbf{t}}{\partial \mathbf{y}_2} & \cdots & \dfrac{\partial \mathbf{t}}{\partial \mathbf{y}_K} \end{bmatrix} \\ &= \mathbf{A} - \tau \mathbf{B} \begin{bmatrix} \dfrac{\partial \mathbf{f}_1^{\mathcal{E}}}{\partial \mathbf{y}} & & \\ & \ddots & \\ & & \dfrac{\partial \mathbf{f}_K^{\mathcal{E}}}{\partial \mathbf{y}} \end{bmatrix} - \tau \mathbf{\Gamma} \begin{bmatrix} \dfrac{\partial \mathbf{f}_1^{\mathcal{I}}}{\partial \mathbf{y}} & & \\ & \ddots & \\ & & \dfrac{\partial \mathbf{f}_K^{\mathcal{I}}}{\partial \mathbf{y}} \end{bmatrix}. \end{aligned} \tag{A.82}$$

  It is easy to see that $\dfrac{\partial \mathbf{t}}{\partial \overline{\mathbf{y}}}$ has the form of (2.1) and (2.3), with the (k, j) $N \times N$ block being

$$\frac{\partial \mathbf{t}_k}{\partial \mathbf{y}_j} = \begin{cases} \alpha_{k-j}^{(\sigma)} \mathbf{I}_N - \tau \beta_{k-j}^{(\sigma)} \dfrac{\partial \mathbf{f}_j^{\mathcal{E}}}{\partial \mathbf{y}} - \tau \gamma_{k-j}^{(\sigma)} \dfrac{\partial \mathbf{f}_j^{\mathcal{I}}}{\partial \mathbf{y}} & \text{if } k^\star \le j \le k \\ \mathbf{0}_{N \times N} & \text{otherwise,} \end{cases} \tag{A.83}$$

with $\beta_0^{(\sigma)} = 0$.

- The product of $\dfrac{\partial \mathbf{t}_k}{\partial \overline{\mathbf{y}}}$ with $\overline{\mathbf{w}}$ is

$$\frac{\partial \mathbf{t}_k}{\partial \overline{\mathbf{y}}}\overline{\mathbf{w}} = \sum_{j=k^\star}^{k} \alpha_{k-j}^{(\sigma)} \mathbf{w}_j - \tau \sum_{j=k^\star}^{k-1} \beta_{k-j}^{(\sigma)} \frac{\partial \mathbf{f}_j^{\mathcal{E}}}{\partial \mathbf{y}} \mathbf{w}_j - \tau \sum_{j=k^\star}^{k} \gamma_{k-j}^{(\sigma)} \frac{\partial \mathbf{f}_j^{\mathcal{I}}}{\partial \mathbf{y}} \mathbf{w}_j. \tag{A.84}$$

- The product of $\dfrac{\partial \mathbf{t}_k}{\partial \mathbf{y}_j}^\top$ with $\mathbf{w}_k$ is

$$\frac{\partial \mathbf{t}_k}{\partial \mathbf{y}_j}^\top \mathbf{w}_k = \alpha_{k-j}^{(\sigma)} \mathbf{w}_k - \tau\, \beta_{k-j}^{(\sigma)} \frac{\partial \mathbf{f}_j^{\mathcal{E}}}{\partial \mathbf{y}}^\top \mathbf{w}_k - \tau\, \gamma_{k-j}^{(\sigma)} \frac{\partial \mathbf{f}_j^{\mathcal{I}}}{\partial \mathbf{y}}^\top \mathbf{w}_k \tag{A.85}$$

with $\beta_0^{(\sigma)} = 0$, and therefore the product of $\dfrac{\partial \mathbf{t}}{\partial \mathbf{y}_j}^\top$ with $\overline{\mathbf{w}}$ is

$$\begin{aligned} \frac{\partial \mathbf{t}}{\partial \mathbf{y}_j}^\top \overline{\mathbf{w}} = & \sum_{k=j}^{\min(K,j+s)} \alpha_{k-j}^{(\sigma)} \mathbf{w}_k - \tau \sum_{k=j+1}^{\min(K,j+s)} \beta_{k-j}^{(\sigma)} \frac{\partial \mathbf{f}_j^{\mathcal{E}}}{\partial \mathbf{y}}^\top \mathbf{w}_k + \\ & - \tau \sum_{k=j}^{\min(K,j+s)} \gamma_{k-j}^{(\sigma)} \frac{\partial \mathbf{f}_j^{\mathcal{I}}}{\partial \mathbf{y}}^\top \mathbf{w}_k. \end{aligned} \tag{A.86}$$

- $\dfrac{\partial \mathbf{t}}{\partial \mathbf{m}}$

Differentiating $\mathbf{t}$ with respect to $\mathbf{m}$ gives

$$\frac{\partial \mathbf{t}}{\partial \mathbf{m}} = -\tau \left(\mathbf{B} \frac{\partial \overline{\mathbf{f}}^{\mathcal{E}}}{\partial \mathbf{m}} + \boldsymbol{\Gamma} \frac{\partial \overline{\mathbf{f}}^{\mathcal{I}}}{\partial \mathbf{m}} + \boldsymbol{\beta} \frac{\partial \mathbf{f}_0^{\mathcal{E}}}{\partial \mathbf{m}} + \boldsymbol{\gamma} \frac{\partial \mathbf{f}_0^{\mathcal{I}}}{\partial \mathbf{m}} \right) \tag{A.87}$$

with

$$\frac{\partial \overline{\mathbf{f}}^{\mathcal{E}}}{\partial \mathbf{m}} = \begin{bmatrix} \dfrac{\partial \mathbf{f}_1^{\mathcal{E}}(\mathbf{m})}{\partial \mathbf{m}} \\ \vdots \\ \dfrac{\partial \mathbf{f}_K^{\mathcal{E}}(\mathbf{m})}{\partial \mathbf{m}} \end{bmatrix} \quad \text{and} \quad \frac{\partial \overline{\mathbf{f}}^{\mathcal{I}}}{\partial \mathbf{m}} = \begin{bmatrix} \dfrac{\partial \mathbf{f}_1^{\mathcal{I}}(\mathbf{m})}{\partial \mathbf{m}} \\ \vdots \\ \dfrac{\partial \mathbf{f}_K^{\mathcal{I}}(\mathbf{m})}{\partial \mathbf{m}} \end{bmatrix}. \tag{A.88}$$

Therefore the kth $N \times N_{\mathbf{m}}$ block of (A.87) is

$$\frac{\partial \mathbf{t}_k}{\partial \mathbf{m}} = -\tau \sum_{j=k^\star}^{k-1} \beta_{k-j}^{(\sigma)} \frac{\partial \mathbf{f}_j^{\mathcal{E}}}{\partial \mathbf{m}} - \tau \sum_{j=k^\star}^{k} \gamma_{k-j}^{(\sigma)} \frac{\partial \mathbf{f}_j^{\mathcal{I}}}{\partial \mathbf{m}} - \tau\, \beta_k^{(k)} \frac{\partial \mathbf{f}_0^{\mathcal{E}}}{\partial \mathbf{m}} - \tau\, \gamma_k^{(k)} \frac{\partial \mathbf{f}_0^{\mathcal{I}}}{\partial \mathbf{m}} \tag{A.89}$$

with $\beta_k^{(k)} = \gamma_k^{(k)} = 0$ if $k > s$.

- The product of (A.89) with $\mathbf{w}_{\mathbf{m}}$ is

$$\begin{aligned} \frac{\partial \mathbf{t}_k}{\partial \mathbf{m}} \mathbf{w}_{\mathbf{m}} = -\tau \sum_{j=k^\star}^{k-1} \beta_{k-j}^{(\sigma)} \frac{\partial \mathbf{f}_j^{\mathcal{E}}}{\partial \mathbf{m}} \mathbf{w}_{\mathbf{m}} - \tau \sum_{j=k^\star}^{k} \gamma_{k-j}^{(\sigma)} \frac{\partial \mathbf{f}_j^{\mathcal{I}}}{\partial \mathbf{m}} \mathbf{w}_{\mathbf{m}} + \\ - \tau\, \beta_k^{(k)} \frac{\partial \mathbf{f}_0^{\mathcal{E}}}{\partial \mathbf{m}} \mathbf{w}_{\mathbf{m}} - \tau\, \gamma_k^{(k)} \frac{\partial \mathbf{f}_0^{\mathcal{I}}}{\partial \mathbf{m}} \mathbf{w}_{\mathbf{m}}. \end{aligned} \tag{A.90}$$

- The product of the transpose with the k the block of $\mathbf{w}$ is

$$\begin{aligned} \frac{\partial \mathbf{t}_k}{\partial \mathbf{m}}^\top \mathbf{w}_k = -\tau \sum_{j=k^\star}^{k-1} \beta_{k-j}^{(\sigma)} \frac{\partial \mathbf{f}_j^{\mathcal{E}}}{\partial \mathbf{m}}^\top \mathbf{w}_k - \tau \sum_{j=k^\star}^{k} \gamma_{k-j}^{(\sigma)} \frac{\partial \mathbf{f}_j^{\mathcal{I}}}{\partial \mathbf{m}}^\top \mathbf{w}_k + \\ - \tau\, \beta_k^{(k)} \frac{\partial \mathbf{f}_0^{\mathcal{E}}}{\partial \mathbf{m}}^\top \mathbf{w}_k - \tau\, \gamma_k^{(k)} \frac{\partial \mathbf{f}_0^{\mathcal{I}}}{\partial \mathbf{m}}^\top \mathbf{w}_k. \end{aligned} \tag{A.91}$$

The product of the transpose of $\dfrac{\partial \mathbf{t}_k}{\partial \mathbf{m}}$ with $\overline{\mathbf{w}}$ hence is

$$\begin{aligned} \frac{\partial \mathbf{t}}{\partial \mathbf{m}}^\top \mathbf{w} = -\tau \sum_{k=1}^{K} \sum_{j=k^\star}^{k-1} \beta_{k-j}^{(\sigma)} \frac{\partial \mathbf{f}_j^{\mathcal{E}}}{\partial \mathbf{m}}^\top \mathbf{w}_k - \tau \frac{\partial \mathbf{f}_0^{\mathcal{E}}}{\partial \mathbf{m}}^\top \sum_{k=1}^{s} \beta_k^{(k)} \mathbf{w}_k + \\ - \tau \sum_{k=1}^{K} \sum_{j=k^\star}^{k} \gamma_{k-j}^{(\sigma)} \frac{\partial \mathbf{f}_j^{\mathcal{I}}}{\partial \mathbf{m}}^\top \mathbf{w}_k - \tau \frac{\partial \mathbf{f}_0^{\mathcal{I}}}{\partial \mathbf{m}}^\top \sum_{k=1}^{s} \gamma_k^{(k)} \mathbf{w}_k. \end{aligned} \tag{A.92}$$

- $\dfrac{\partial \mathbf{t}}{\partial \mathbf{y}_0}$

This derivative $\mathbf{t}$ with respect to $\mathbf{y}_0$ is

$$\frac{\partial \mathbf{t}}{\partial \mathbf{y}_0} = \boldsymbol{\alpha} - \tau \boldsymbol{\beta} \frac{\partial \mathbf{f}_0^{\mathcal{E}}}{\partial \mathbf{y}} - \tau \boldsymbol{\gamma} \frac{\partial \mathbf{f}_0^{\mathcal{I}}}{\partial \mathbf{y}} \tag{A.93}$$

and the kth $N \times N$ block is

$$\frac{\partial \mathbf{t}_k}{\partial \mathbf{y}_0} = \alpha_k^{(k)} \mathbf{I}_N - \tau \beta_k^{(k)} \frac{\partial \mathbf{f}_0^{\mathcal{E}}}{\partial \mathbf{y}} - \tau \gamma_k^{(k)} \frac{\partial \mathbf{f}_0^{\mathcal{I}}}{\partial \mathbf{y}}, \tag{A.94}$$

with $\alpha_k^{(k)} = \beta_k^{(k)} = \gamma_k^{(k)} = 0$ if $k > s$.

- The product of (A.94) with $\mathbf{w}_0$ is

$$\frac{\partial \mathbf{t}_k}{\partial \mathbf{y}_0} \mathbf{w}_0 = \alpha_k^{(k)} \mathbf{w}_0 - \tau \beta_k^{(k)} \frac{\partial \mathbf{f}_0^{\mathcal{E}}}{\partial \mathbf{y}} \mathbf{w}_0 - \tau \gamma_k^{(k)} \frac{\partial \mathbf{f}_0^{\mathcal{I}}}{\partial \mathbf{y}} \mathbf{w}_0. \tag{A.95}$$

- The product of the transpose with the kth block of $\mathbf{w}$ is

$$\frac{\partial \mathbf{t}_k}{\partial \mathbf{y}_0}^\top \mathbf{w}_k = \alpha_k^{(k)} \mathbf{w}_k - \tau \beta_k^{(k)} \frac{\partial \mathbf{f}_0^{\mathcal{E}}}{\partial \mathbf{y}}^\top \mathbf{w}_k - \tau \gamma_k^{(k)} \frac{\partial \mathbf{f}_0^{\mathcal{I}}}{\partial \mathbf{y}}^\top \mathbf{w}_k. \tag{A.96}$$

The product of the transpose of $\dfrac{\partial \mathbf{t}}{\partial \mathbf{y}_0}$ with the $\mathbf{w}$ therefore is

$$\frac{\partial \mathbf{t}}{\partial \mathbf{y}_0}^\top \mathbf{w} = \sum_{k=1}^{s} \alpha_k^{(k)} \mathbf{w}_k - \tau \frac{\partial \mathbf{f}_0^{\mathcal{E}}}{\partial \mathbf{y}}^\top \sum_{k=1}^{s} \beta_k^{(k)} \mathbf{w}_k - \tau \frac{\partial \mathbf{f}_0^{\mathcal{I}}}{\partial \mathbf{y}}^\top \sum_{k=1}^{s} \gamma_k^{(k)} \mathbf{w}_k. \tag{A.97}$$

Derivatives of $\dfrac{\partial \mathbf{t}}{\partial \overline{\mathbf{y}}}$

We will now find expressions for the derivatives of (A.84)

$$\frac{\partial \mathbf{t}_k}{\partial \overline{\mathbf{y}}} \overline{\mathbf{w}} = \sum_{j=k^\star}^{k} \alpha_{k-j}^{(\sigma)} \mathbf{w}_j - \tau \sum_{j=k^\star}^{k-1} \beta_{k-j}^{(\sigma)} \frac{\partial \mathbf{f}_j^{\mathcal{E}}}{\partial \mathbf{y}} \mathbf{w}_j - \tau \sum_{j=k^\star}^{k} \gamma_{k-j}^{(\sigma)} \frac{\partial \mathbf{f}_j^{\mathcal{I}}}{\partial \mathbf{y}} \mathbf{w}_j$$

with respect to $\mathbf{m}$, $\mathbf{y}_0$ and $\overline{\mathbf{y}}$.

- $\dfrac{\partial^2 \mathbf{t}}{\partial \mathbf{m} \partial \overline{\mathbf{y}}} \overline{\mathbf{w}}$

Differentiating (A.84) with respect to $\mathbf{m}$ gives the kth $N \times N_{\mathbf{m}}$ block of $\dfrac{\partial^2 \mathbf{t}}{\partial \mathbf{m} \partial \overline{\mathbf{y}}} \overline{\mathbf{w}}$:

$$\frac{\partial^2 \mathbf{t}_k}{\partial \mathbf{m} \partial \overline{\mathbf{y}}} \overline{\mathbf{w}} = -\tau \sum_{j=k^\star}^{k-1} \beta_{k-j}^{(\sigma)} \frac{\partial^2 \mathbf{f}_j^{\mathcal{E}}}{\partial \mathbf{m} \partial \mathbf{y}} \mathbf{w}_j - \tau \sum_{j=k^\star}^{k} \gamma_{k-j}^{(\sigma)} \frac{\partial^2 \mathbf{f}_j^{\mathcal{I}}}{\partial \mathbf{m} \partial \mathbf{y}} \mathbf{w}_j. \tag{A.98}$$

- The product of (A.98) with $\mathbf{v}_{\mathbf{m}}$ is

$$\begin{aligned} \left(\frac{\partial^2 \mathbf{t}_k}{\partial \mathbf{m} \partial \overline{\mathbf{y}}} \overline{\mathbf{w}} \right) \mathbf{v}_{\mathbf{m}} = -\tau \sum_{j=k^\star}^{k-1} \beta_{k-j}^{(\sigma)} \left(\frac{\partial^2 \mathbf{f}_j^{\mathcal{E}}}{\partial \mathbf{m} \partial \mathbf{y}} \mathbf{w}_j \right) \mathbf{v}_{\mathbf{m}} + \\ - \tau \sum_{j=k^\star}^{k} \gamma_{k-j}^{(\sigma)} \left(\frac{\partial^2 \mathbf{f}_j^{\mathcal{I}}}{\partial \mathbf{m} \partial \mathbf{y}} \mathbf{w}_j \right) \mathbf{v}_{\mathbf{m}}. \end{aligned} \tag{A.99}$$

- The product of the transpose of (A.98) with the kth block of $\overline{\mathbf{v}}$ is

$$\begin{aligned} \left(\frac{\partial^2 \mathbf{t}_k}{\partial \mathbf{m} \partial \overline{\mathbf{y}}} \overline{\mathbf{w}} \right)^\top \mathbf{v}_k = -\tau \sum_{j=k^\star}^{k-1} \beta_{k-j}^{(\sigma)} \left(\frac{\partial^2 \mathbf{f}_j^{\mathcal{E}}}{\partial \mathbf{m} \partial \mathbf{y}} \mathbf{w}_j \right)^\top \mathbf{v}_k + \\ - \tau \sum_{j=k^\star}^{k} \gamma_{k-j}^{(\sigma)} \left(\frac{\partial^2 \mathbf{f}_j^{\mathcal{I}}}{\partial \mathbf{m} \partial \mathbf{y}} \mathbf{w}_j \right)^\top \mathbf{v}_k. \end{aligned} \tag{A.100}$$

- $\dfrac{\partial \mathbf{t}_k}{\partial \mathbf{y}_0 \partial \overline{\mathbf{y}}} \overline{\mathbf{w}}$

The derivative of (A.84) with respect to $\mathbf{y}_0$ is simply

$$\frac{\partial^2 \mathbf{t}_k}{\partial \mathbf{y}_0 \partial \overline{\mathbf{y}}} \overline{\mathbf{w}} = \mathbf{0}_{N \times N}. \tag{A.101}$$

- $\dfrac{\partial^2 \mathbf{t}}{\partial \overline{\mathbf{y}} \partial \overline{\mathbf{y}}} \overline{\mathbf{w}}$

Consider the jth $N \times N$ block of

$$\frac{\partial^2 \mathbf{t}_k}{\partial \overline{\mathbf{y}} \partial \overline{\mathbf{y}}} \overline{\mathbf{w}} = \left[\frac{\partial^2 \mathbf{t}_k}{\partial \mathbf{y}_1 \partial \overline{\mathbf{y}}} \overline{\mathbf{w}} \quad \frac{\partial^2 \mathbf{t}_k}{\partial \mathbf{y}_2 \partial \overline{\mathbf{y}}} \overline{\mathbf{w}} \quad \cdots \quad \frac{\partial^2 \mathbf{t}_k}{\partial \mathbf{y}_K \partial \overline{\mathbf{y}}} \overline{\mathbf{w}} \right].$$

Taking the derivative of (A.84) with respect to $\mathbf{y}_j$ gives the (k,j)th $N \times N$ block of $\dfrac{\partial^2 \mathbf{t}}{\partial \overline{\mathbf{y}} \partial \overline{\mathbf{y}}} \overline{\mathbf{w}}$,

$$\frac{\partial^2 \mathbf{t}_k}{\partial \mathbf{y}_j \partial \overline{\mathbf{y}}} \overline{\mathbf{w}} = \begin{cases} -\tau \beta_{k-j}^{(\sigma)} \dfrac{\partial^2 \mathbf{f}_j^{\mathcal{E}}}{\partial \mathbf{y} \partial \mathbf{y}} \mathbf{w}_j - \tau \gamma_{k-j}^{(\sigma)} \dfrac{\partial^2 \mathbf{f}_j^{\mathcal{I}}}{\partial \mathbf{y} \partial \mathbf{y}} \mathbf{w}_j & \text{if } k^\star \leq j \leq k \\ \mathbf{0}_{N \times N} & \text{otherwise.} \end{cases} \tag{A.102}$$

- The product of $\dfrac{\partial^2 \mathbf{t}_k}{\partial \mathbf{y}_j \partial \overline{\mathbf{y}}} \overline{\mathbf{w}}$ with the jth block of $\overline{\mathbf{v}}$ trivially is

$$\left(\frac{\partial^2 \mathbf{t}_k}{\partial \mathbf{y}_j \partial \overline{\mathbf{y}}} \overline{\mathbf{w}} \right) \mathbf{v}_j = \begin{cases} -\tau \left(\beta_{k-j}^{(\sigma)} \dfrac{\partial^2 \mathbf{f}_j^{\mathcal{E}}}{\partial \mathbf{y} \partial \mathbf{y}} \mathbf{w}_j + \gamma_{k-j}^{(\sigma)} \dfrac{\partial^2 \mathbf{f}_j^{\mathcal{I}}}{\partial \mathbf{y} \partial \mathbf{y}} \mathbf{w}_j \right) \mathbf{v}_j & \\ & \text{if } k^\star \leq j \leq k \\ \mathbf{0}_{N \times 1} & \text{otherwise,} \end{cases} \tag{A.103}$$

so that

$$\left(\frac{\partial^2 \mathbf{t}_k}{\partial \overline{\mathbf{y}} \partial \overline{\mathbf{y}}} \overline{\mathbf{w}} \right) \overline{\mathbf{v}} = -\tau \sum_{j=k^\star}^{k} \left(\beta_{k-j}^{(\sigma)} \frac{\partial^2 \mathbf{f}_j^{\mathcal{E}}}{\partial \mathbf{y} \partial \mathbf{y}} \mathbf{w}_j + \gamma_{k-j}^{(\sigma)} \frac{\partial^2 \mathbf{f}_j^{\mathcal{I}}}{\partial \mathbf{y} \partial \mathbf{y}} \mathbf{w}_j \right) \mathbf{v}_j. \tag{A.104}$$

- The product of the transpose of $\left(\dfrac{\partial^2 \mathbf{t}_k}{\partial \mathbf{y}_j \partial \overline{\mathbf{y}}} \overline{\mathbf{w}} \right)$ with the kth block of $\overline{\mathbf{v}}$ is

$$\left(\frac{\partial^2 \mathbf{t}_k}{\partial \mathbf{y}_j \partial \overline{\mathbf{y}}} \overline{\mathbf{w}} \right)^\top \mathbf{v}_k = \begin{cases} -\tau \left(\beta_{k-j}^{(\sigma)} \dfrac{\partial^2 \mathbf{f}_j^{\mathcal{E}}}{\partial \mathbf{y} \partial \mathbf{y}} \mathbf{w}_j^{\top} + \gamma_{k-j}^{(\sigma)} \dfrac{\partial^2 \mathbf{f}_j^{\mathcal{I}}}{\partial \mathbf{y} \partial \mathbf{y}} \mathbf{w}_j^{\top} \right) \mathbf{v}_k & \\ & \text{if } j \leq k \leq j^\star \\ \mathbf{0}_{N \times 1} & \text{otherwise,} \end{cases} \tag{A.105}$$

with $j^\star = \min(K, j+s)$.

Derivatives of $\dfrac{\partial \mathbf{t}}{\partial \mathbf{m}}$

The derivatives of (A.90),

$$\begin{aligned}\frac{\partial \mathbf{t}_k}{\partial \mathbf{m}}\mathbf{w}_{\mathbf{m}} &= -\tau \sum_{j=k^\star}^{k-1} \beta_{k-j}^{(\sigma)} \frac{\partial \mathbf{f}_j^{\mathcal{E}}}{\partial \mathbf{m}}\mathbf{w}_{\mathbf{m}} - \tau \sum_{j=k^\star}^{k} \gamma_{k-j}^{(\sigma)} \frac{\partial \mathbf{f}_j^{\mathcal{I}}}{\partial \mathbf{m}}\mathbf{w}_{\mathbf{m}} + \\ &\qquad - \tau \beta_k^{(k)} \frac{\partial \mathbf{f}_0^{\mathcal{E}}}{\partial \mathbf{m}}\mathbf{w}_{\mathbf{m}} - \tau \gamma_k^{(k)} \frac{\partial \mathbf{f}_0^{\mathcal{I}}}{\partial \mathbf{m}}\mathbf{w}_{\mathbf{m}} \\ &= -\tau \sum_{j=k^\dagger}^{k-1} \beta_{k-j}^{(\sigma)} \frac{\partial \mathbf{f}_j^{\mathcal{E}}}{\partial \mathbf{m}}\mathbf{w}_{\mathbf{m}} - \tau \sum_{j=k^\dagger}^{k} \gamma_{k-j}^{(\sigma)} \frac{\partial \mathbf{f}_j^{\mathcal{I}}}{\partial \mathbf{m}}\mathbf{w}_{\mathbf{m}},\end{aligned}$$

where $k^\dagger = \min(0, k-s)$, with respect to $\mathbf{m}$, $\mathbf{y}_0$ and $\overline{\mathbf{y}}$ are:

- $\dfrac{\partial^2 \mathbf{t}}{\partial \mathbf{m} \partial \mathbf{m}}\mathbf{w}_{\mathbf{m}}$

  Taking the derivative of (A.90) with respect to $\mathbf{m}$ gives the kth $N \times N_{\mathbf{m}}$ block of $\dfrac{\partial^2 \mathbf{t}}{\partial \mathbf{m} \partial \mathbf{m}}\mathbf{w}_{\mathbf{m}}$

$$\frac{\partial^2 \mathbf{t}_k}{\partial \mathbf{m} \partial \mathbf{m}}\mathbf{w}_{\mathbf{m}} = -\tau \sum_{j=k^\dagger}^{k-1} \beta_{k-j}^{(\sigma)} \frac{\partial^2 \mathbf{f}_j^{\mathcal{E}}}{\partial \mathbf{m} \partial \mathbf{m}}\mathbf{w}_{\mathbf{m}} - \tau \sum_{j=k^\dagger}^{k} \gamma_{k-j}^{(\sigma)} \frac{\partial^2 \mathbf{f}_j^{\mathcal{I}}}{\partial \mathbf{m} \partial \mathbf{m}}\mathbf{w}_{\mathbf{m}}. \tag{A.106}$$

  - The product with $\mathbf{v}_{\mathbf{m}}$ is

$$\begin{aligned}\left(\frac{\partial^2 \mathbf{t}_k}{\partial \mathbf{m} \partial \mathbf{m}}\mathbf{w}_{\mathbf{m}}\right)\mathbf{v}_{\mathbf{m}} &= -\tau \sum_{j=k^\dagger}^{k-1} \beta_{k-j}^{(\sigma)} \left(\frac{\partial^2 \mathbf{f}_j^{\mathcal{E}}}{\partial \mathbf{m} \partial \mathbf{m}}\mathbf{w}_{\mathbf{m}}\right)\mathbf{v}_{\mathbf{m}} + \\ &\qquad - \tau \sum_{j=k^\dagger}^{k} \gamma_{k-j}^{(\sigma)} \left(\frac{\partial^2 \mathbf{f}_j^{\mathcal{I}}}{\partial \mathbf{m} \partial \mathbf{m}}\mathbf{w}_{\mathbf{m}}\right)\mathbf{v}_{\mathbf{m}}.\end{aligned} \tag{A.107}$$

- The product of its transpose with the kth block of $\overline{\mathbf{v}}$ is

$$\left(\frac{\partial^2 \mathbf{t}_k}{\partial \mathbf{m} \partial \mathbf{m}} \mathbf{w}_{\mathbf{m}}\right)^\top \mathbf{v}_k = -\tau \sum_{j=k^\dagger}^{k-1} \beta_{k-j}^{(\sigma)} \left(\frac{\partial^2 \mathbf{f}_j^{\mathcal{E}}}{\partial \mathbf{m} \partial \mathbf{m}} \mathbf{w}_{\mathbf{m}}\right)^\top \mathbf{v}_k + \\ - \tau \sum_{j=k^\dagger}^{k} \gamma_{k-j}^{(\sigma)} \left(\frac{\partial^2 \mathbf{f}_j^{\mathcal{I}}}{\partial \mathbf{m} \partial \mathbf{m}} \mathbf{w}_{\mathbf{m}}\right)^\top \mathbf{v}_k. \tag{A.108}$$

- $\dfrac{\partial^2 \mathbf{t}}{\partial \mathbf{y}_0 \partial \mathbf{m}} \mathbf{w}_{\mathbf{m}}$

Differentiating (A.90) with respect to $\mathbf{y}_0$ immediately gives

$$\frac{\partial^2 \mathbf{t}_k}{\partial \mathbf{y}_0 \partial \mathbf{m}} \mathbf{w}_{\mathbf{m}} = -\tau \, \beta_k^{(k)} \frac{\partial^2 \mathbf{f}_0^{\mathcal{E}}}{\partial \mathbf{y} \partial \mathbf{m}} \mathbf{w}_{\mathbf{m}} - \tau \, \gamma_k^{(k)} \frac{\partial^2 \mathbf{f}_0^{\mathcal{I}}}{\partial \mathbf{y} \partial \mathbf{m}} \mathbf{w}_{\mathbf{m}} \tag{A.109}$$

with $\beta_k^{(k)} = \gamma_k^{(k)} = 0$ if $k > s$.

- The product with $\mathbf{v}_0$ is

$$\left(\frac{\partial^2 \mathbf{t}_k}{\partial \mathbf{y}_0 \partial \mathbf{m}} \mathbf{w}_{\mathbf{m}}\right) \mathbf{v}_0 = -\tau \, \beta_k^{(k)} \left(\frac{\partial^2 \mathbf{f}_0^{\mathcal{E}}}{\partial \mathbf{y} \partial \mathbf{m}} \mathbf{w}_{\mathbf{m}}\right) \mathbf{v}_0 - \tau \, \gamma_k^{(k)} \left(\frac{\partial^2 \mathbf{f}_0^{\mathcal{I}}}{\partial \mathbf{y} \partial \mathbf{m}} \mathbf{w}_{\mathbf{m}}\right) \mathbf{v}_0 \tag{A.110}$$

  with $\beta_k^{(k)} = \gamma_k^{(k)} = 0$ if $k > s$.

- The product of its transpose with the kth block of $\mathbf{v}_{\mathbf{y}}$ is

$$\left(\frac{\partial^2 \mathbf{t}_k}{\partial \mathbf{y}_0 \partial \mathbf{m}} \mathbf{w}_{\mathbf{m}}\right)^\top \mathbf{v}_k = -\tau \, \beta_k^{(k)} \left(\frac{\partial^2 \mathbf{f}_0^{\mathcal{E}}}{\partial \mathbf{y} \partial \mathbf{m}} \mathbf{w}_{\mathbf{m}}\right)^\top \mathbf{v}_k + \\ - \tau \, \gamma_k^{(k)} \left(\frac{\partial^2 \mathbf{f}_0^{\mathcal{I}}}{\partial \mathbf{y} \partial \mathbf{m}} \mathbf{w}_{\mathbf{m}}\right)^\top \mathbf{v}_k \tag{A.111}$$

  with $\beta_k^{(k)} = \gamma_k^{(k)} = 0$ if $k > s$.

- $\dfrac{\partial^2 \mathbf{t}}{\partial \overline{\mathbf{y}} \partial \mathbf{m}} \mathbf{w}_{\mathbf{m}}$

Consider the jth $N \times N$ block of

$$\frac{\partial^2 \mathbf{t}_k}{\partial \overline{\mathbf{y}} \partial \mathbf{m}} \mathbf{w}_{\mathbf{m}} = \left[\frac{\partial^2 \mathbf{t}_k}{\partial \mathbf{y}_1 \partial \mathbf{m}} \mathbf{w}_{\mathbf{m}} \quad \frac{\partial^2 \mathbf{t}_k}{\partial \mathbf{y}_2 \partial \mathbf{m}} \mathbf{w}_{\mathbf{m}} \quad \cdots \quad \frac{\partial^2 \mathbf{t}_k}{\partial \mathbf{y}_K \partial \mathbf{m}} \mathbf{w}_{\mathbf{m}} \right]. \tag{A.112}$$

The (k, j)th $N \times N$ block of $\dfrac{\partial^2 \mathbf{t}}{\partial \overline{\mathbf{y}} \partial \mathbf{m}} \mathbf{w}_{\mathbf{m}}$ is

$$\frac{\partial^2 \mathbf{t}_k}{\partial \mathbf{y}_j \partial \mathbf{m}} \mathbf{w}_{\mathbf{m}} = \begin{cases} -\tau \beta_{k-j}^{(\sigma)} \dfrac{\partial^2 \mathbf{f}_j^{\mathcal{E}}}{\partial \mathbf{y} \partial \mathbf{m}} \mathbf{m} - \tau \gamma_{k-j}^{(\sigma)} \dfrac{\partial^2 \mathbf{f}_j^{\mathcal{I}}}{\partial \mathbf{y} \partial \mathbf{m}} \mathbf{m} & \text{if } k^\star \leq j \leq k \\ \mathbf{0}_{N \times N} & \text{otherwise.} \end{cases} \tag{A.113}$$

- The product of $\dfrac{\partial^2 \mathbf{t}_k}{\partial \mathbf{y}_j \partial \mathbf{m}} \mathbf{w}_{\mathbf{m}}$ with the jth block of $\overline{\mathbf{v}}$ is

$$\left(\frac{\partial^2 \mathbf{t}_k}{\partial \mathbf{y}_j \partial \mathbf{m}} \mathbf{w}_{\mathbf{m}} \right) \mathbf{v}_j = \begin{cases} -\tau \beta_{k-j}^{(\sigma)} \left(\dfrac{\partial^2 \mathbf{f}_j^{\mathcal{E}}}{\partial \mathbf{y} \partial \mathbf{m}} \mathbf{m} \right) \mathbf{v}_j + \\ \qquad -\tau \gamma_{k-j}^{(\sigma)} \left(\dfrac{\partial^2 \mathbf{f}_j^{\mathcal{I}}}{\partial \mathbf{y} \partial \mathbf{m}} \mathbf{m} \right) \mathbf{v}_j & \text{if } k^\star \leq j \leq k \\ \mathbf{0}_{N \times 1} & \text{otherwise,} \end{cases} \tag{A.114}$$

  with $\beta_0^{(\sigma)} = 0$.

- The product of the transpose of $\dfrac{\partial^2 \mathbf{t}_k}{\partial \mathbf{y}_j \partial \mathbf{m}} \mathbf{w}_{\mathbf{m}}$ with the kth block of $\overline{\mathbf{v}}$ is

$$\left(\frac{\partial^2 \mathbf{t}_k}{\partial \mathbf{y}_j \partial \mathbf{m}} \mathbf{w}_{\mathbf{m}} \right)^\top \mathbf{v}_k = \begin{cases} -\tau \beta_{k-j}^{(\sigma)} \left(\dfrac{\partial^2 \mathbf{f}_j^{\mathcal{E}}}{\partial \mathbf{y} \partial \mathbf{m}} \mathbf{m} \right)^\top \mathbf{v}_k + \\ \qquad -\tau \gamma_{k-j}^{(\sigma)} \left(\dfrac{\partial^2 \mathbf{f}_j^{\mathcal{I}}}{\partial \mathbf{y} \partial \mathbf{m}} \mathbf{m} \right)^\top \mathbf{v}_k & \text{if } j \leq k \leq j^\star \\ \mathbf{0}_{N \times 1} & \text{otherwise.} \end{cases} \tag{A.115}$$

  with $j^\star = \min(K, j + s)$ and $\beta_0^{(\sigma)} = 0$.

Derivatives of $\dfrac{\partial \mathbf{t}}{\partial \mathbf{y}_0}$

Finally, we find expressions for the derivatives of (A.95),

$$\frac{\partial \mathbf{t}_k}{\partial \mathbf{y}_0}\mathbf{w}_0 = \alpha_k^{(k)}\mathbf{w}_0 - \tau\,\beta_k^{(k)}\frac{\partial \mathbf{f}_0^{\mathcal{E}}}{\partial \mathbf{y}}\mathbf{w}_0 - \tau\,\gamma_k^{(k)}\frac{\partial \mathbf{f}_0^{\mathcal{I}}}{\partial \mathbf{y}}\mathbf{w}_0,$$

where $\alpha_k^{(k)} = \beta_k^{(k)} = \gamma_k^{(k)} = 0$ if $k > s$, with respect to $\mathbf{m}$, $\mathbf{y}_0$ and $\overline{\mathbf{y}}$:

- $\dfrac{\partial^2 \mathbf{t}}{\partial \mathbf{m}\partial \mathbf{y}_0}\mathbf{w}_0$

  Differentiating (A.95) with respect to $\mathbf{m}$ gives the kth $N \times N_{\mathbf{m}}$ block of $\dfrac{\partial^2 \mathbf{t}}{\partial \mathbf{m}\partial \mathbf{y}_0}\mathbf{w}_0$:

$$\frac{\partial^2 \mathbf{t}_k}{\partial \mathbf{m}\partial \mathbf{y}_0}\mathbf{w}_0 = -\tau\,\beta_k^{(k)}\frac{\partial^2 \mathbf{f}_0^{\mathcal{E}}}{\partial \mathbf{m}\partial \mathbf{y}}\mathbf{w}_0 - \tau\,\gamma_k^{(k)}\frac{\partial^2 \mathbf{f}_0^{\mathcal{I}}}{\partial \mathbf{m}\partial \mathbf{y}}\mathbf{w}_0 \tag{A.116}$$

  with $\beta_k^{(k)} = \gamma_k^{(k)} = 0$ if $k > s$.

  - The product with $\mathbf{v}_{\mathbf{m}}$ is

$$\left(\frac{\partial^2 \mathbf{t}_k}{\partial \mathbf{m}\partial \mathbf{y}_0}\mathbf{w}_0\right)\mathbf{v}_{\mathbf{m}} = -\tau\beta_k^{(k)}\left(\frac{\partial^2 \mathbf{f}_0^{\mathcal{E}}}{\partial \mathbf{m}\partial \mathbf{y}}\mathbf{w}\right)\mathbf{v}_{\mathbf{m}} - \tau\gamma_k^{(k)}\left(\frac{\partial^2 \mathbf{f}_0^{\mathcal{I}}}{\partial \mathbf{m}\partial \mathbf{y}}\mathbf{w}_0\right)\mathbf{v}_{\mathbf{m}}. \tag{A.117}$$

  - The product of the transpose with the kth block of $\overline{\mathbf{v}}$ is

$$\begin{aligned}\left(\frac{\partial^2 \mathbf{t}_k}{\partial \mathbf{m}\partial \mathbf{y}_0}\mathbf{w}_0\right)^{\top}\mathbf{v}_k = -\tau\,\beta_k^{(k)}&\left(\frac{\partial^2 \mathbf{f}_0^{\mathcal{E}}}{\partial \mathbf{m}\partial \mathbf{y}}\mathbf{w}\right)^{\top}\mathbf{v}_k + \\ &- \tau\,\gamma_k^{(k)}\left(\frac{\partial^2 \mathbf{f}_0^{\mathcal{I}}}{\partial \mathbf{m}\partial \mathbf{y}}\mathbf{w}\right)^{\top}\mathbf{v}_k.\end{aligned} \tag{A.118}$$

- $\dfrac{\partial^2 \mathbf{t}}{\partial \mathbf{y}_0\partial \mathbf{y}_0}\mathbf{w}_0$

  The kth $N \times N$ block of $\dfrac{\partial^2 \mathbf{t}}{\partial \mathbf{y}_0\partial \mathbf{y}_0}\mathbf{w}_0$ is obtained by taking the derivative of (A.95)

with respect to $\mathbf{y}_0$:

$$\frac{\partial^2 \mathbf{t}_k}{\partial \mathbf{y}_0 \partial \mathbf{y}_0}\mathbf{w}_0 = -\tau\,\beta_k^{(k)}\frac{\partial^2 \mathbf{f}_0^{\mathcal{E}}}{\partial \mathbf{y}\,\partial \mathbf{y}}\mathbf{w}_0 - \tau\,\gamma_k^{(k)}\frac{\partial^2 \mathbf{f}_0^{\mathcal{I}}}{\partial \mathbf{y}\,\partial \mathbf{y}}\mathbf{w}_0 \tag{A.119}$$

with $\beta_k^{(k)} = \gamma_k^{(k)} = 0$ if $k > s$.

- The product with $\mathbf{v_m}$ is

$$\left(\frac{\partial^2 \mathbf{t}_k}{\partial \mathbf{y}_0 \partial \mathbf{y}_0}\mathbf{w}_0\right)\mathbf{v}_0 = -\tau\,\beta_k^{(k)}\left(\frac{\partial^2 \mathbf{f}_0^{\mathcal{E}}}{\partial \mathbf{y}\,\partial \mathbf{y}}\mathbf{w}\right)\mathbf{v}_0 - \tau\,\gamma_k^{(k)}\left(\frac{\partial^2 \mathbf{f}_0^{\mathcal{I}}}{\partial \mathbf{y}\,\partial \mathbf{y}}\mathbf{w}\right)\mathbf{v}_0. \tag{A.120}$$

- The product of the transpose with the kth block of $\overline{\mathbf{v}}$ is

$$\left(\frac{\partial^2 \mathbf{t}_k}{\partial \mathbf{y}_0 \partial \mathbf{y}_0}\mathbf{w}_0\right)^\top \mathbf{v}_k = -\tau\beta_k^{(k)}\left(\frac{\partial^2 \mathbf{f}_0^{\mathcal{E}}}{\partial \mathbf{y}\,\partial \mathbf{y}}\mathbf{w}\right)^\top \mathbf{v}_k - \tau\gamma_k^{(k)}\left(\frac{\partial^2 \mathbf{f}_0^{\mathcal{I}}}{\partial \mathbf{y}\,\partial \mathbf{y}}\mathbf{w}\right)^\top \mathbf{v}_k. \tag{A.121}$$

- $\dfrac{\partial^2 \mathbf{t}}{\partial \overline{\mathbf{y}}\,\partial \mathbf{y}_0}\mathbf{w}_0$

Differentiating (A.95) with respect to $\overline{\mathbf{y}}$ immediately gives

$$\frac{\partial^2 \mathbf{t}_k}{\partial \overline{\mathbf{y}}\,\partial \mathbf{y}_0}\mathbf{w}_0 = \mathbf{0}_{N\times KN}. \tag{A.122}$$

A.2.2 IMEX Runge-Kutta Methods

The time-stepping vector $\mathbf{t} = \mathbf{t}(\overline{\mathbf{y}}; \mathbf{m}, \mathbf{y}_0)$ for IMEX RK methods was found in Section 2.2.2 to be

$$\mathbf{t}\left(\overline{\mathbf{y}}; \mathbf{m}, \mathbf{y}_0\right) = \mathbf{T}\,\overline{\mathbf{y}} - \overline{\mathbf{f}}\left(\overline{\mathbf{y}}, \mathbf{m}, \mathbf{y}_0\right),$$

where

$$\overline{\mathbf{y}} = \begin{bmatrix}\mathbf{Y}_1^\top & \mathbf{y}_1^\top & \mathbf{Y}_2^\top & \mathbf{y}_2^\top & \cdots & \mathbf{Y}_K^\top & \mathbf{y}_K^\top\end{bmatrix}^\top$$
$$\overline{\mathbf{f}} = \begin{bmatrix}\mathbf{F}_1^\top & \mathbf{y}_0 & \mathbf{F}_2^\top & \mathbf{0}_{1\times N} & \cdots & \mathbf{F}_K^\top & \mathbf{0}_{1\times N}\end{bmatrix}^\top$$

$$\mathbf{T} = \begin{bmatrix} \mathbf{I}_{(2s+1)N} & & & & & & \\ \mathbf{B}^\top & \mathbf{I}_N & & & & & \\ & & \mathbf{I}_{(2s+1)N} & & & & \\ & -\mathbf{I}_N & \mathbf{B}^\top & \mathbf{I}_N & & & \\ & & \ddots & \ddots & \ddots & & \\ & & & & & \mathbf{I}_{(2s+1)N} & \\ & & & & -\mathbf{I}_N & \mathbf{B}^\top & \mathbf{I}_N \end{bmatrix},$$

see (2.31). We have let

$$\mathbf{B} = -\tau_k \begin{bmatrix} b_1^{\mathcal{E}} & b_1^{\mathcal{I}} & b_2^{\mathcal{E}} & \cdots & b_{s-1}^{\mathcal{E}} & b_{s-1}^{\mathcal{I}} & b_s^{\mathcal{E}} \end{bmatrix}^\top \otimes \mathbf{I}_N.$$

and

$$\mathbf{Y}_k = \begin{bmatrix} \mathbf{Y}_{k,1} \\ \vdots \\ \mathbf{Y}_{k,2s-1} \end{bmatrix}, \quad \mathbf{F}_k = \begin{bmatrix} \mathbf{F}_{k,1} \\ \vdots \\ \mathbf{F}_{k,2s-1} \end{bmatrix},$$

with

$$\mathbf{Y}_{k,i} = \begin{cases} \mathbf{Y}^{\mathcal{E}}_{k,\lceil \frac{i}{2} \rceil} & \text{if } i \text{ odd} \\ \mathbf{Y}^{\mathcal{I}}_{k,\frac{i}{2}} & \text{if } i \text{ even} \end{cases}$$

and

$$\mathbf{F}_{k,i} = \begin{cases} \mathbf{f}^{\mathcal{E}}\left(\underline{\mathbf{y}}_{k,\lceil \frac{i}{2} \rceil - 1}, t_{k-1,\lceil \frac{i}{2} \rceil}\right) & \text{if } i \text{ odd} \\ \mathbf{f}^{\mathcal{I}}\left(\underline{\mathbf{y}}_{k,\frac{i}{2}}, t_{k-1,\frac{i}{2}+1}\right) & \text{if } i \text{ even.} \end{cases}$$

$\underline{\mathbf{y}}_{k,\sigma}$ was defined to be $\underline{\mathbf{y}}_{k,\sigma} = \mathbf{y}_{k-1} + \tau_k \sum_{i=1}^{\sigma} a^{\mathcal{E}}_{\sigma+1,i} \mathbf{Y}^{\mathcal{E}}_{k,i} + \tau_k \sum_{i=1}^{\sigma} a^{\mathcal{I}}_{\sigma,i} \mathbf{Y}^{\mathcal{I}}_{k,i}$, and we have $\underline{\mathbf{y}}_{k,0} = \mathbf{y}_{k-1}$. Let $\mathbf{f}^{\mathcal{E}}_{k,\sigma} = \mathbf{f}^{\mathcal{E}}\left(\underline{\mathbf{y}}_{k,\sigma-1}, t_{k-1,\sigma}\right)$ and $\mathbf{f}^{\mathcal{I}}_{k,\sigma} = \mathbf{f}^{\mathcal{I}}\left(\underline{\mathbf{y}}_{k,\sigma-1}, t_{k-1,\sigma}\right)$.

From now on we omit the time dependence of $\mathbf{f}^{\mathcal{E}}$ and $\mathbf{f}^{\mathcal{I}}$. We let $\widehat{\mathbf{y}}_k = \begin{bmatrix} \mathbf{Y}_k^\top & \mathbf{y}_k^\top \end{bmatrix}^\top$. We also let $\overline{\mathbf{w}} = \begin{bmatrix} \widehat{\mathbf{w}}_1^\top & \cdots & \widehat{\mathbf{w}}_K^\top \end{bmatrix}^\top$ and $\overline{\mathbf{v}} = \begin{bmatrix} \widehat{\mathbf{v}}_1^\top & \cdots & \widehat{\mathbf{v}}_K^\top \end{bmatrix}^\top$ be arbitrary vectors of length $2sKN$, where $\widehat{\mathbf{w}}_k^\top$ and $\widehat{\mathbf{v}}_k^\top$ are defined analogously to $\widehat{\mathbf{y}}_k^\top$. Also, $\mathbf{w}_{\mathbf{m}}$ and $\mathbf{v}_{\mathbf{m}}$ are arbitrary vectors of length $N_{\mathbf{m}}$, and $\mathbf{w}_0$ and $\mathbf{v}_0$ are arbitrary vectors of length N.

We will now take the derivative of $\mathbf{t}$ with respect to each of $\overline{\mathbf{y}}$, $\mathbf{m}$ and $\mathbf{y}_0$ in turn. For this we need the derivatives of $\underline{\mathbf{y}}_{k,\sigma}$ with respect to $\mathbf{y}_{k-1}$, $\mathbf{Y}^{\mathcal{E}}_{k,i}$ and $\mathbf{Y}^{\mathcal{I}}_{k,i}$, which are $\dfrac{\partial \underline{\mathbf{y}}_{k,\sigma}}{\partial \mathbf{y}_{k-1}} = \mathbf{I}_N$, $\dfrac{\partial \underline{\mathbf{y}}_{k,\sigma}}{\partial \mathbf{Y}^{\mathcal{E}}_{k,i}} = \tau_k a^{\mathcal{E}}_{\sigma+1,i}\mathbf{I}_N$ and $\dfrac{\partial \underline{\mathbf{y}}^{\mathcal{I}}_{k,\sigma}}{\partial \mathbf{Y}^{\mathcal{I}}_{k,i}} = \tau_k a^{\mathcal{I}}_{\sigma i}\mathbf{I}_N$.

- $\dfrac{\partial \mathbf{t}}{\partial \overline{\mathbf{y}}}$

  We break the derivative of $\mathbf{t}$ with respect to $\overline{\mathbf{y}}$ into parts:

$$\frac{\partial \mathbf{t}}{\partial \overline{\mathbf{y}}} = \begin{bmatrix} \dfrac{\partial \mathbf{t}}{\partial \widehat{\mathbf{y}}_1} & \dfrac{\partial \mathbf{t}}{\partial \widehat{\mathbf{y}}_2} & \cdots & \dfrac{\partial \mathbf{t}}{\partial \widehat{\mathbf{y}}_K} \end{bmatrix} = \mathbf{T} - \begin{bmatrix} \dfrac{\partial \overline{\mathbf{f}}}{\partial \widehat{\mathbf{y}}_1} & \dfrac{\partial \overline{\mathbf{f}}}{\partial \widehat{\mathbf{y}}_2} & \cdots & \dfrac{\partial \overline{\mathbf{f}}}{\partial \widehat{\mathbf{y}}_K} \end{bmatrix}. \tag{A.123}$$

  The derivative of $\overline{\mathbf{f}}$ at the kth time-step is

$$\frac{\partial \overline{\mathbf{f}}_k}{\partial \widehat{\mathbf{y}}_j} = \begin{bmatrix} \dfrac{\partial \mathbf{F}_k}{\partial \widehat{\mathbf{y}}_j} \\ \mathbf{0}_{N\times 2sN} \end{bmatrix}$$

  with

$$\frac{\partial \mathbf{F}_k}{\partial \widehat{\mathbf{y}}_j} = \begin{bmatrix} \dfrac{\partial \mathbf{F}_k}{\partial \mathbf{Y}^{\mathcal{E}}_{j,1}} & \dfrac{\partial \mathbf{F}_k}{\partial \mathbf{Y}^{\mathcal{I}}_{j,1}} & \dfrac{\partial \mathbf{F}_k}{\partial \mathbf{Y}^{\mathcal{E}}_{j,2}} & \cdots & \dfrac{\partial \mathbf{F}_k}{\partial \mathbf{Y}^{\mathcal{I}}_{j,s-1}} & \dfrac{\partial \mathbf{F}_k}{\partial \mathbf{Y}^{\mathcal{E}}_{j,s}} & \dfrac{\partial \mathbf{F}_k}{\partial \mathbf{y}_j} \end{bmatrix}$$

  where $1 \le j, k \le K$. $\dfrac{\partial \mathbf{F}_{k,i}}{\partial \widehat{\mathbf{y}}_{j,\ell}}$ denotes the (i,ℓ)th $N \times N$ block of $\dfrac{\partial \mathbf{F}_k}{\partial \widehat{\mathbf{y}}_j}$, with $1 \le i \le 2s-1$ and $1 \le \ell \le 2s$.

  For $j = k-1$,

$$\frac{\partial \mathbf{F}_{k,i}}{\partial \widehat{\mathbf{y}}_{k-1}} = \begin{cases} \begin{bmatrix} \mathbf{0}_{N\times(2s-1)N} & \dfrac{\partial \mathbf{f}^{\mathcal{E}}_{k,\lceil \frac{i}{2} \rceil}}{\partial \mathbf{y}} \end{bmatrix} & \text{if } \ i \text{ odd} \\ \begin{bmatrix} \mathbf{0}_{N\times(2s-1)N} & \dfrac{\partial \mathbf{f}^{\mathcal{I}}_{k,\frac{i}{2}+1}}{\partial \mathbf{y}} \end{bmatrix} & \text{if } \ i \text{ even} \end{cases} \tag{A.124a}$$

and for $j = k$ we have

$$\frac{\partial \mathbf{F}_{k,i}}{\partial \widehat{\mathbf{y}}_k} = \begin{cases} \begin{bmatrix} \tau_k \mathbf{a}_{\lceil \frac{i}{2} \rceil} \dfrac{\partial \mathbf{f}^{\mathcal{E}}_{k,\lceil \frac{i}{2} \rceil}}{\partial \mathbf{y}} & \mathbf{0}_{N\times N} \end{bmatrix} & \text{if } i \text{ odd} \\ \begin{bmatrix} \tau_k \mathbf{a}_{\frac{i}{2}+1} \dfrac{\partial \mathbf{f}^{\mathcal{I}}_{k,\frac{i}{2}+1}}{\partial \mathbf{y}} & \mathbf{0}_{N\times N} \end{bmatrix} & \text{if } i \text{ even,} \end{cases} \tag{A.124b}$$

where

$$\mathbf{a}_\sigma = \begin{bmatrix} a^{\mathcal{E}}_{\sigma,1} & a^{\mathcal{I}}_{\sigma-1,1} & \cdots & a^{\mathcal{E}}_{\sigma,\sigma-1} & a^{\mathcal{I}}_{\sigma-1,\sigma-1} & \mathbf{0}_{1\times 2(s-\sigma)+1} \end{bmatrix} \otimes \mathbf{I}_N.$$

For all other values of j we have $\dfrac{\partial \mathbf{F}_{k,i}}{\partial \widehat{\mathbf{y}}_j} = \mathbf{0}_{N\times 2sN}$. Now letting

$$\begin{aligned} \mathbf{A}_s &= \begin{bmatrix} \mathbf{a}_1^\top & \mathbf{a}_2^\top & \mathbf{a}_2^\top & \mathbf{a}_3^\top & \cdots & \mathbf{a}_s^\top \end{bmatrix}^\top \\ \frac{\partial \mathbf{f}_k}{\partial \mathbf{y}} &= \text{blkdiag}\left(\frac{\partial \mathbf{f}^{\mathcal{E}}_{k,1}}{\partial \mathbf{y}}, \frac{\partial \mathbf{f}^{\mathcal{I}}_{k,2}}{\partial \mathbf{y}}, \frac{\partial \mathbf{f}^{\mathcal{E}}_{k,2}}{\partial \mathbf{y}}, \frac{\partial \mathbf{f}^{\mathcal{I}}_{k,3}}{\partial \mathbf{y}}, \cdots, \frac{\partial \mathbf{f}^{\mathcal{E}}_{k,s}}{\partial \mathbf{y}} \right), \end{aligned} \tag{A.125}$$

and using the definition of $\mathbf{T}$, the (k,j)th $2sN \times 2sN$ block of $\dfrac{\partial \mathbf{t}}{\partial \mathbf{y}}$ is

$$\frac{\partial \mathbf{t}_k}{\partial \widehat{\mathbf{y}}_j} = \begin{cases} \begin{bmatrix} \mathbf{I}_{(2s-1)N} - \tau_k \dfrac{\partial \mathbf{f}_k}{\partial \mathbf{y}} (\mathbf{A}_s \otimes \mathbf{I}_N) & \mathbf{0}_{(2s-1)N\times N} \\ \mathbf{B}^\top & \mathbf{I}_N \end{bmatrix} & \text{if } j = k \\ \begin{bmatrix} \mathbf{0}_{(2s-1)N\times(2s-1)N} & -\dfrac{\partial \mathbf{f}_k}{\partial \mathbf{y}} (\mathbf{1}_{2s-1} \otimes \mathbf{I}_N) \\ \mathbf{0}_{N\times(2s-1)N} & -\mathbf{I}_N \end{bmatrix} & \text{if } j = k-1 \\ \mathbf{0}_{2(s-1)N\times 2(s-1)N} & \text{otherwise.} \end{cases} \tag{A.126}$$

Finally, let

$$\begin{aligned} \mathbf{A}_k &= \mathbf{I}_{(2s-1)N} - \tau_k \frac{\partial \mathbf{f}_k}{\partial \mathbf{y}} (\mathbf{A}_s \otimes \mathbf{I}_N) \\ \mathbf{C}_k &= -\frac{\partial \mathbf{f}_k}{\partial \mathbf{y}} (\mathbf{1}_{2s-1} \otimes \mathbf{I}_N) \end{aligned} \tag{A.127}$$

and write

$$\frac{\partial \mathbf{t}}{\partial \overline{\mathbf{y}}} = \begin{bmatrix} \mathbf{A}_1 & & & & & & \\ \mathbf{B}^\top & \mathbf{I}_N & & & & & \\ & \mathbf{C}_2 & \mathbf{A}_2 & & & & \\ & -\mathbf{I}_N & \mathbf{B}^\top & \mathbf{I}_N & & & \\ & & \ddots & \ddots & \ddots & & \\ & & & & \mathbf{C}_K & \mathbf{A}_K & \\ & & & & -\mathbf{I}_N & \mathbf{B}^\top & \mathbf{I}_N \end{bmatrix}. \tag{A.128}$$

which is a lower block-triangular matrix.

- The kth $2sN \times 1$ block of the product $\dfrac{\partial \mathbf{t}}{\partial \overline{\mathbf{y}}}\overline{\mathbf{w}}$ is

$$\frac{\partial \mathbf{t}_k}{\partial \overline{\mathbf{y}}}\overline{\mathbf{w}} = \begin{bmatrix} \mathbf{A}_k \mathbf{W}_k + \mathbf{C}_k \mathbf{w}_{k-1} \\ \mathbf{B}^\top \mathbf{W}_k + \mathbf{w}_k - \mathbf{w}_{k-1} \end{bmatrix} \tag{A.129}$$

with $\mathbf{w}_0 = \mathbf{0}_{N\times 1}$. If $\dfrac{\partial \mathbf{t}_{k,i}}{\partial \overline{\mathbf{y}}}\overline{\mathbf{w}}$ is the ith $N \times 1$ block of $\dfrac{\partial \mathbf{t}_k}{\partial \overline{\mathbf{y}}}\overline{\mathbf{w}}$, with $1 \le i \le 2s$, we have

$$\frac{\partial \mathbf{t}_{k,i}}{\partial \overline{\mathbf{y}}}\overline{\mathbf{w}} = \begin{cases} \mathbf{W}^{\mathcal{E}}_{k,\lceil \frac{i}{2} \rceil} - \dfrac{\partial \mathbf{f}^{\mathcal{E}}_{k,\lceil \frac{i}{2} \rceil}}{\partial \mathbf{y}} \underline{\mathbf{w}}_{k,\lfloor \frac{i}{2} \rfloor} & \text{if } \ i \le 2s-1 \\ & \text{and } i \text{ odd} \\ \mathbf{W}^{\mathcal{I}}_{k,\frac{i}{2}} - \dfrac{\partial \mathbf{f}^{\mathcal{I}}_{k,\frac{i}{2}+1}}{\partial \mathbf{y}} \underline{\mathbf{w}}_{k,\frac{i}{2}} & \text{if } \ i \le 2(s-1) \\ & \text{and } i \text{ even} \\ -\tau_k \left(\displaystyle\sum_{\sigma=1}^{s} b^{\mathcal{E}}_\sigma \mathbf{W}^{\mathcal{E}}_{k,\sigma} + \sum_{\sigma=1}^{s-1} b^{\mathcal{I}}_\sigma \mathbf{W}^{\mathcal{I}}_{k,\sigma} \right) + & \text{if } \ i = 2s, \\ \qquad + \mathbf{w}_k - \mathbf{w}_{k-1} & \end{cases} \tag{A.130}$$

where $\underline{\mathbf{w}}_{k,\sigma} = \mathbf{w}_{k-1} + \tau_k \sum_{i=1}^{\sigma} a^{\mathcal{E}}_{\sigma+1,i} \mathbf{W}^{\mathcal{E}}_{k,i} + \tau_k \sum_{i=1}^{\sigma} a^{\mathcal{I}}_{\sigma i} \mathbf{W}^{\mathcal{I}}_{k,i}$.

- The transpose of $\dfrac{\partial \mathbf{t}}{\partial \overline{\mathbf{y}}}$ is upper block-triangular:

$$\frac{\partial \mathbf{t}}{\partial \overline{\mathbf{y}}}^{\top} = \begin{bmatrix} \mathbf{A}_1^\top & \mathbf{B} & & & & & \\ & \mathbf{I}_N & \mathbf{C}_2^\top & -\mathbf{I}_N & & & \\ & & \mathbf{A}_2^\top & \mathbf{B} & & & \\ & & & \ddots & \ddots & \ddots & \\ & & & & \mathbf{I}_N & \mathbf{C}_K^\top & -\mathbf{I}_N \\ & & & & & \mathbf{A}_K^\top & \mathbf{B} \\ & & & & & & \mathbf{I}_N \end{bmatrix}. \tag{A.131}$$

The jth $2sN \times 1$ block of the product of $\dfrac{\partial \mathbf{t}}{\partial \overline{\mathbf{y}}}^{\top}$ with some arbitrary $\overline{\mathbf{w}}$ therefore is

$$\frac{\partial \mathbf{t}}{\partial \widehat{\mathbf{y}}_j}^{\top} \overline{\mathbf{w}} = \begin{bmatrix} \mathbf{A}_j^\top \mathbf{W}_j + \mathbf{B}\mathbf{w}_j \\ \mathbf{C}_{j+1}^\top \mathbf{W}_{j+1} + \mathbf{w}_j - \mathbf{w}_{j+1} \end{bmatrix}, \tag{A.132}$$

with $\mathbf{W}_{K+1,\sigma} = \mathbf{w}_{K+1} = \mathbf{0}_{N\times 1}$. The ith $N \times 1$ block $\dfrac{\partial \mathbf{t}}{\partial \widehat{\mathbf{y}}_{j,i}}^{\top} \overline{\mathbf{w}}$ of $\dfrac{\partial \mathbf{t}}{\partial \widehat{\mathbf{y}}_j}^{\top} \overline{\mathbf{w}}$, with

$1 \leq \sigma \leq 2s$, is

$$\frac{\partial \mathbf{t}}{\partial \widehat{\mathbf{y}}_{j,i}}^{\top} \overline{\mathbf{w}} = \begin{cases} \mathbf{W}^{\mathcal{E}}_{j,\lceil \frac{i}{2} \rceil} - \tau_j \sum\limits_{\ell=\lceil \frac{i}{2} \rceil+1}^{s} a^{\mathcal{E}}_{\ell,\lceil \frac{i}{2} \rceil} \frac{\partial \mathbf{f}^{\mathcal{I}}_{j,\ell}}{\partial \mathbf{y}}^{\top} \mathbf{W}^{\mathcal{I}}_{j,\ell-1} + & \\ - \tau_j \sum\limits_{\ell=\lceil \frac{i}{2} \rceil+1}^{s} a^{\mathcal{E}}_{\ell,\lceil \frac{i}{2} \rceil} \frac{\partial \mathbf{f}^{\mathcal{E}}_{j,\ell}}{\partial \mathbf{y}}^{\top} \mathbf{W}^{\mathcal{E}}_{j,\ell} - \tau_j b^{\mathcal{E}}_{\lceil \frac{i}{2} \rceil} \mathbf{w}_j & \text{if } \ i \leq 2s-1 \\ & \text{and } i \text{ odd} \\ \mathbf{W}^{\mathcal{I}}_{j,\frac{i}{2}} - \tau_j \sum\limits_{\ell=\frac{i}{2}}^{s-1} a^{\mathcal{I}}_{\ell,\frac{i}{2}} \frac{\partial \mathbf{f}^{\mathcal{I}}_{j,\ell+1}}{\partial \mathbf{y}}^{\top} \mathbf{W}^{\mathcal{I}}_{j,\ell} + & \\ - \tau_j \sum\limits_{\ell=\frac{i}{2}}^{s-1} a^{\mathcal{I}}_{\ell,\frac{i}{2}} \frac{\partial \mathbf{f}^{\mathcal{E}}_{j,\ell+1}}{\partial \mathbf{y}}^{\top} \mathbf{W}^{\mathcal{E}}_{j,\ell+1} - \tau_j b^{\mathcal{I}}_{\frac{i}{2}} \mathbf{w}_j & \text{if } \ i \leq 2(s-1) \\ & \text{and } i \text{ even} \\ \mathbf{w}_j - \mathbf{w}_{j+1} - \sum\limits_{\ell=1}^{s} \frac{\partial \mathbf{f}^{\mathcal{E}}_{j+1,\ell}}{\partial \mathbf{y}}^{\top} \mathbf{W}^{\mathcal{E}}_{j+1,\ell} + & \\ \quad - \sum\limits_{\ell=1}^{s-1} \frac{\partial \mathbf{f}^{\mathcal{I}}_{j+1,\ell+1}}{\partial \mathbf{y}}^{\top} \mathbf{W}^{\mathcal{I}}_{j+1,\ell} & \text{if } \ i = 2s, \end{cases} \tag{A.133}$$

- $\dfrac{\partial \mathbf{t}}{\partial \mathbf{m}}$

Taking the derivative of $\mathbf{t}$ with respect to $\mathbf{m}$,

$$\frac{\partial \mathbf{t}}{\partial \mathbf{m}} = \frac{\partial}{\partial \mathbf{m}} \left(\mathbf{T}\mathbf{y} - \overline{\mathbf{f}} \right) = -\frac{\partial \overline{\mathbf{f}}}{\partial \mathbf{m}} \tag{A.134}$$

with

$$\frac{\partial \overline{\mathbf{f}}}{\partial \mathbf{m}} = \begin{bmatrix} \dfrac{\partial \mathbf{F}_1}{\partial \mathbf{m}} \\ \mathbf{0}_{N \times N_{\mathbf{m}}} \\ \vdots \\ \dfrac{\partial \mathbf{F}_K}{\partial \mathbf{m}} \\ \mathbf{0}_{N \times N_{\mathbf{m}}} \end{bmatrix}, \quad \text{where} \quad \frac{\partial \mathbf{F}_k}{\partial \mathbf{m}} = \begin{bmatrix} \dfrac{\partial \mathbf{F}_{k,1}}{\partial \mathbf{m}} \\ \vdots \\ \dfrac{\partial \mathbf{F}_{k,2s-1}}{\partial \mathbf{m}} \end{bmatrix} \tag{A.135}$$

and

$$\frac{\partial \mathbf{F}_{k,i}}{\partial \mathbf{m}} = \begin{cases} \dfrac{\partial \mathbf{f}^{\mathcal{E}}_{k,\lceil \frac{i}{2} \rceil}}{\partial \mathbf{m}} & \text{if } \ i \text{ odd} \\ \dfrac{\partial \mathbf{f}^{\mathcal{I}}_{k,\frac{i}{2}+1}}{\partial \mathbf{m}} & \text{if } \ i \text{ even.} \end{cases} \tag{A.136}$$

- The product $\dfrac{\partial \mathbf{t}_k}{\partial \mathbf{m}} \mathbf{w}_{\mathbf{m}}$ is simply

$$\frac{\partial \mathbf{t}_k}{\partial \mathbf{m}} \mathbf{w}_{\mathbf{m}} = - \begin{bmatrix} \dfrac{\partial \mathbf{F}_k}{\partial \mathbf{m}} \mathbf{w}_{\mathbf{m}} \\ \mathbf{0}_{N \times 1} \end{bmatrix} \tag{A.137}$$

  with $\dfrac{\partial \mathbf{F}_k}{\partial \mathbf{m}} \mathbf{w}_{\mathbf{m}}$ computed in the obvious way using (A.136).

- The product $\dfrac{\partial \mathbf{t}_k}{\partial \mathbf{m}}^{\top} \widehat{\mathbf{w}}_k$ is

$$\frac{\partial \mathbf{t}_k}{\partial \mathbf{m}}^{\top} \widehat{\mathbf{w}}_k = - \sum_{\sigma=1}^{s} \frac{\partial \mathbf{f}^{\mathcal{E}}_{k,\sigma}}{\partial \mathbf{m}}^{\top} \mathbf{W}^{\mathcal{E}}_{k,\sigma} - \sum_{\sigma=1}^{s-1} \frac{\partial \mathbf{f}^{\mathcal{I}}_{k,\sigma}}{\partial \mathbf{m}}^{\top} \mathbf{W}^{\mathcal{I}}_{k,\sigma}. \tag{A.138}$$

- $\dfrac{\partial \mathbf{t}}{\partial \mathbf{y}_0}$

Taking the derivative of $\mathbf{T}$ with respect to $\mathbf{y}_0$,

$$\frac{\partial \mathbf{t}}{\partial \mathbf{y}_0} = \frac{\partial}{\partial \mathbf{y}_0} \left(\mathbf{T}\mathbf{y} - \overline{\mathbf{f}} \right) = - \frac{\partial \overline{\mathbf{f}}}{\partial \mathbf{y}_0} \tag{A.139}$$

with

$$\frac{\partial \overline{\mathbf{f}}}{\partial \mathbf{y}_0} = \begin{bmatrix} \dfrac{\partial \mathbf{F}_1}{\partial \mathbf{y}} \dfrac{\partial \underline{\mathbf{y}}_1}{\partial \mathbf{y}_0} \\ \mathbf{I}_N \\ \mathbf{0}_{2s(K-1)N \times N} \end{bmatrix}, \quad \text{where} \quad \frac{\partial \underline{\mathbf{y}}_1}{\partial \mathbf{y}_0} = \begin{bmatrix} \dfrac{\partial \underline{\mathbf{y}}_{1,1}}{\partial \mathbf{y}_0} \\ \vdots \\ \dfrac{\partial \underline{\mathbf{y}}_{1,s}}{\partial \mathbf{y}_0} \end{bmatrix} = \mathbf{1}_s \otimes \mathbf{I}_N. \tag{A.140}$$

- The product $\dfrac{\partial \mathbf{t}_k}{\partial \mathbf{y}_0}\mathbf{w}_0$ is

$$\frac{\partial \mathbf{t}_k}{\partial \mathbf{y}_0}\mathbf{w}_0 = \begin{cases} -\begin{bmatrix} \dfrac{\partial \mathbf{F}_1}{\partial \mathbf{y}}(\mathbf{1}_s \otimes \mathbf{w}_0) \\ \mathbf{w}_0 \end{bmatrix} & \text{if } \ k=1 \\ \mathbf{0}_{N\times 1} & \text{otherwise,} \end{cases} \tag{A.141}$$

with

$$\frac{\partial \mathbf{F}_1}{\partial \mathbf{y}} = \operatorname{diag}\left(\frac{\partial \mathbf{f}_{1,1}^{\mathcal{E}}}{\partial \mathbf{y}}, \frac{\partial \mathbf{f}_{1,2}^{\mathcal{I}}}{\partial \mathbf{y}}, \frac{\partial \mathbf{f}_{1,2}^{\mathcal{E}}}{\partial \mathbf{y}}, \frac{\partial \mathbf{f}_{1,3}^{\mathcal{I}}}{\partial \mathbf{y}}, \cdots, \frac{\partial \mathbf{f}_{1,s}^{\mathcal{E}}}{\partial \mathbf{y}}\right).$$

- The product $\dfrac{\partial \mathbf{t}_k}{\partial \mathbf{y}_0}^{\top}\widehat{\mathbf{w}}_k$ is

$$\frac{\partial \mathbf{t}_k}{\partial \mathbf{y}_0}^{\top}\widehat{\mathbf{w}}_k = \begin{cases} -\displaystyle\sum_{\sigma=1}^{s}\frac{\partial \mathbf{f}_{1,\sigma}^{\mathcal{E}}}{\partial \mathbf{m}}^{\top}\mathbf{W}_{1,\sigma}^{\mathcal{E}} - \sum_{\sigma=1}^{s-1}\frac{\partial \mathbf{f}_{1,\sigma}^{\mathcal{I}}}{\partial \mathbf{m}}^{\top}\mathbf{W}_{1,\sigma}^{\mathcal{I}} - \mathbf{w}_1 & \text{if } k=1 \\ \mathbf{0}_{N\times 1} & \text{otherwise.} \end{cases} \tag{A.142}$$

Derivatives of $\dfrac{\partial \mathbf{t}}{\partial \overline{\mathbf{y}}}$

We now find the derivatives of the ith $N\times 1$ block $\dfrac{\partial \mathbf{t}_{k,i}}{\partial \overline{\mathbf{y}}}\overline{\mathbf{w}}$ of the vector $\dfrac{\partial \mathbf{t}_k}{\partial \overline{\mathbf{y}}}\overline{\mathbf{w}}$, with $1 \le 1 \le 2s$, which in turn is the kth block of $\dfrac{\partial \mathbf{t}}{\partial \overline{\mathbf{y}}}\overline{\mathbf{w}} = \dfrac{\partial \mathbf{t}}{\partial \overline{\mathbf{y}}}\overline{\mathbf{w}}$.

The expression for the product $\dfrac{\partial \mathbf{t}_{k,i}}{\partial \overline{\mathbf{y}}}\overline{\mathbf{w}}$ was given in (A.130). We now find expressions for the derivatives of (A.130) with respect to $\overline{\mathbf{y}}$, $\mathbf{m}$ and $\mathbf{y}_0$.

- $\left(\dfrac{\partial \mathbf{t}}{\partial \overline{\mathbf{y}}}\overline{\mathbf{w}}\right)_{\mathbf{m}}$

We differentiate (A.130) with respect to $\mathbf{m}$:

$$\frac{\partial^2 \mathbf{t}_{k,i}}{\partial \mathbf{m} \partial \overline{\mathbf{y}}} \overline{\mathbf{w}} = \begin{cases} -\dfrac{\partial^2 \mathbf{f}^{\mathcal{E}}_{k,\lceil \frac{i}{2} \rceil}}{\partial \mathbf{m} \partial \mathbf{y}} \underline{\mathbf{w}}_{k,\lceil \frac{i}{2} \rceil} & \text{if } \ i \leq 2s-1 \text{ and } i \text{ odd} \\ -\dfrac{\partial^2 \mathbf{f}^{\mathcal{I}}_{k,\frac{i}{2}+1}}{\partial \mathbf{m} \partial \mathbf{y}} \underline{\mathbf{w}}_{k,\frac{i}{2}} & \text{if } \ i \leq 2(s-1) \text{ and } i \text{ even} \\ \mathbf{0}_{N \times N_{\mathbf{m}}} & \text{if } \ i = 2s, \end{cases} \tag{A.143}$$

where $\underline{\mathbf{w}}_{k,\sigma} = \mathbf{w}_{k-1} + \tau_k \sum_{i=1}^{\sigma} a^{\mathcal{E}}_{\sigma+1,i} \mathbf{W}^{\mathcal{E}}_{k,i} + \tau_k \sum_{i=1}^{\sigma} a^{\mathcal{I}}_{\sigma i} \mathbf{W}^{\mathcal{I}}_{k,i}$.

- $\dfrac{\partial^2 \mathbf{t}}{\partial \mathbf{y}_0 \partial \overline{\mathbf{y}}} \overline{\mathbf{w}}$

Taking the derivative of (A.130) with respect to $\mathbf{y}_0$ gives

$$\frac{\partial^2 \mathbf{t}_{1,i}}{\partial \mathbf{y}_0 \partial \overline{\mathbf{y}}} \overline{\mathbf{w}} = \begin{cases} -\dfrac{\partial^2 \mathbf{f}^{\mathcal{E}}_{1,\lceil \frac{i}{2} \rceil}}{\partial \mathbf{y} \partial \mathbf{y}} \underline{\mathbf{w}}_{1,\lceil \frac{i}{2} \rceil} & \text{if } \ i \leq 2s-1 \text{ and } i \text{ odd} \\ -\dfrac{\partial^2 \mathbf{f}^{\mathcal{I}}_{1,\frac{i}{2}+1}}{\partial \mathbf{y} \partial \mathbf{y}} \underline{\mathbf{w}}_{1,\frac{i}{2}} & \text{if } \ i \leq 2(s-1) \text{ and } i \text{ even} \\ \mathbf{0}_{N \times N_{\mathbf{m}}} & \text{if } \ i = 2s, \end{cases} \tag{A.144}$$

where $\underline{\mathbf{w}}_{1,\sigma} = \mathbf{w}_0 + \tau_1 \sum_{i=1}^{\sigma} a^{\mathcal{E}}_{\sigma+1,i} \mathbf{W}^{\mathcal{E}}_{1,i} + \tau_1 \sum_{i=1}^{\sigma} a^{\mathcal{I}}_{\sigma i} \mathbf{W}^{\mathcal{I}}_{1,i}$, and $\dfrac{\partial^2 \mathbf{t}_{k,i}}{\partial \mathbf{y}_0 \partial \overline{\mathbf{y}}} \overline{\mathbf{w}} = \mathbf{0}_{N \times N}$ for $k > 1$.

- $\dfrac{\partial^2 \mathbf{t}}{\partial \mathbf{y} \partial \overline{\mathbf{y}}} \overline{\mathbf{w}}$

Consider the ℓth term of

$$\frac{\partial^2 \mathbf{t}_{k,i}}{\partial \mathbf{y} \partial \overline{\mathbf{y}}} \overline{\mathbf{w}} = \left[\frac{\partial^2 \mathbf{t}_{k,i}}{\partial \widehat{\mathbf{y}}_1 \partial \overline{\mathbf{y}}} \overline{\mathbf{w}} \quad \frac{\partial^2 \mathbf{t}_{k,i}}{\partial \widehat{\mathbf{y}}_2 \partial \overline{\mathbf{y}}} \overline{\mathbf{w}} \quad \cdots \quad \frac{\partial^2 \mathbf{t}_{k,i}}{\partial \widehat{\mathbf{y}}_K \partial \overline{\mathbf{y}}} \overline{\mathbf{w}} \right], \tag{A.145}$$

with

$$\frac{\partial^2 \mathbf{t}_{k,i}}{\partial \widehat{\mathbf{y}}_\ell \partial \overline{\mathbf{y}}} \overline{\mathbf{w}} = \left[\frac{\partial^2 \mathbf{t}_{k,i}}{\partial \mathbf{Y}^{\mathcal{E}}_{\ell,1} \partial \overline{\mathbf{y}}} \overline{\mathbf{w}} \quad \frac{\partial^2 \mathbf{t}_{k,i}}{\partial \mathbf{Y}^{\mathcal{I}}_{\ell,1} \partial \overline{\mathbf{y}}} \overline{\mathbf{w}} \quad \cdots \quad \frac{\partial^2 \mathbf{t}_{k,i}}{\partial \mathbf{Y}^{\mathcal{E}}_{\ell,s} \partial \overline{\mathbf{y}}} \overline{\mathbf{w}} \quad \frac{\partial^2 \mathbf{t}_{k,i}}{\partial \mathbf{y}_\ell \partial \overline{\mathbf{y}}} \overline{\mathbf{w}} \right]. \tag{A.146}$$

Then, for $\ell = k-1$,

$$\frac{\partial^2 \mathbf{t}_{k,i}}{\partial \widehat{\mathbf{y}}_{k-1} \partial \overline{\mathbf{y}}} \overline{\mathbf{w}} = \begin{cases} \begin{bmatrix} \mathbf{0}_{N\times(2s-1)N} & -\dfrac{\partial^2 \mathbf{f}^{\mathcal{E}}_{k,\lceil \frac{i}{2} \rceil}}{\partial \mathbf{y} \partial \mathbf{y}} \underline{\mathbf{w}}_{k,\lceil \frac{i}{2} \rceil} \end{bmatrix} & \text{if } \ i \leq 2s-1 \\ & \quad \text{and } i \text{ odd} \\ \begin{bmatrix} \mathbf{0}_{N\times(2s-1)N} & -\dfrac{\partial^2 \mathbf{f}^{\mathcal{I}}_{k,\frac{i}{2}+1}}{\partial \mathbf{y} \partial \mathbf{y}} \underline{\mathbf{w}}_{k,\frac{i}{2}} \end{bmatrix} & \text{if } \ i \leq 2(s-1) \\ & \quad \text{and } i \text{ even} \\ \mathbf{0}_{N\times 2sN} & \text{if } \ i = 2s, \end{cases} \tag{A.147a}$$

for $\ell = k$,

$$\frac{\partial^2 \mathbf{t}_{k,i}}{\partial \widehat{\mathbf{y}}_{k} \partial \overline{\mathbf{y}}} \overline{\mathbf{w}} = \begin{cases} \begin{bmatrix} -\tau_k \, \mathbf{a}_{\lceil \frac{i}{2} \rceil} \dfrac{\partial^2 \mathbf{f}^{\mathcal{E}}_{k,\lceil \frac{i}{2} \rceil}}{\partial \mathbf{y} \partial \mathbf{y}} \underline{\mathbf{w}}_{k,\lceil \frac{i}{2} \rceil} & \mathbf{0}_{N\times N} \end{bmatrix} & \text{if } \ i \leq 2s-1 \\ & \quad \text{and } i \text{ odd} \\ \begin{bmatrix} -\tau_k \, \mathbf{a}_{\frac{i}{2}} \dfrac{\partial^2 \mathbf{f}^{\mathcal{I}}_{k,\frac{i}{2}+1}}{\partial \mathbf{y} \partial \mathbf{y}} \underline{\mathbf{w}}_{k,\frac{i}{2}} & \mathbf{0}_{N\times N} \end{bmatrix} & \text{if } \ i \leq 2(s-1) \\ & \quad \text{and } i \text{ even} \\ \mathbf{0}_{N\times 2sN} & \text{if } \ i = 2s, \end{cases} \tag{A.147b}$$

and

$$\frac{\partial^2 \mathbf{t}_{k,i}}{\partial \widehat{\mathbf{y}}_{\ell} \partial \overline{\mathbf{y}}} \overline{\mathbf{w}} = \mathbf{0}_{2sN\times 2sN} \tag{A.147c}$$

otherwise.

Derivatives of $\dfrac{\partial \mathbf{t}}{\partial \mathbf{m}}$

We consider the derivatives of the ith $N \times 1$ block $\dfrac{\partial \mathbf{t}_{k,i}}{\partial \mathbf{m}} \mathbf{w}_{\mathbf{m}}$ of the vector $\dfrac{\partial \mathbf{t}_{k}}{\partial \mathbf{m}} \mathbf{w}_{\mathbf{m}}$, which in turn is the kth $2sN \times 1$ block of $\dfrac{\partial \mathbf{t}}{\partial \mathbf{m}} \mathbf{w}_{\mathbf{m}}$.

The expression for the product $\dfrac{\partial \mathbf{t}_{k}}{\partial \mathbf{m}} \mathbf{w}_{\mathbf{m}}$ was given in (A.137). We now find expressions for the derivatives of (A.137) with respect to $\overline{\mathbf{y}}$, $\mathbf{m}$ and $\mathbf{y}_0$.

- $\dfrac{\partial^2 \mathbf{t}}{\partial \mathbf{m} \partial \mathbf{m}} \mathbf{w}_{\mathbf{m}}$

We differentiate (A.137) with respect to $\mathbf{m}$:

$$\frac{\partial^2 \mathbf{t}_k}{\partial \mathbf{m} \partial \mathbf{m}} \mathbf{w}_{\mathbf{m}} = - \begin{bmatrix} \dfrac{\partial^2 \mathbf{F}_k}{\partial \mathbf{m} \partial \mathbf{m}} \mathbf{w}_{\mathbf{m}} \\ \mathbf{0}_{N \times N_{\mathbf{m}}} \end{bmatrix} \tag{A.148}$$

with

$$\frac{\partial^2 \mathbf{F}_k}{\partial \mathbf{m} \partial \mathbf{m}} \mathbf{w}_{\mathbf{m}} = \begin{bmatrix} \dfrac{\partial^2 \mathbf{F}_{k,1}}{\partial \mathbf{m} \partial \mathbf{m}} \mathbf{w}_{\mathbf{m}} \\ \vdots \\ \dfrac{\partial^2 \mathbf{F}_{k,2s-1}}{\partial \mathbf{m} \partial \mathbf{m}} \mathbf{w}_{\mathbf{m}} \end{bmatrix},$$

where

$$\frac{\partial^2 \mathbf{F}_{k,i}}{\partial \mathbf{m} \partial \mathbf{m}} \mathbf{w}_{\mathbf{m}} = \begin{cases} \dfrac{\partial^2 \mathbf{f}^{\mathcal{E}}_{k,\lceil \frac{i}{2} \rceil}}{\partial \mathbf{m} \partial \mathbf{m}} \mathbf{w}_{\mathbf{m}} & \text{if } \ i \text{ odd} \\ \dfrac{\partial^2 \mathbf{f}^{\mathcal{I}}_{k,\frac{i}{2}+1}}{\partial \mathbf{m} \partial \mathbf{m}} \mathbf{w}_{\mathbf{m}} & \text{if } \ i \text{ even.} \end{cases}$$

- $\dfrac{\partial^2 \mathbf{t}}{\partial \mathbf{y}_0 \partial \mathbf{m}} \mathbf{w}_{\mathbf{m}}$

Taking the derivative of (A.141) with respect to $\mathbf{y}_0$ gives

$$\frac{\partial^2 \mathbf{t}_k}{\partial \mathbf{y}_0 \partial \mathbf{m}} \mathbf{w}_{\mathbf{m}} = - \begin{bmatrix} \dfrac{\partial^2 \mathbf{F}_k}{\partial \mathbf{y}_0 \partial \mathbf{m}} \mathbf{w}_{\mathbf{m}} \\ \mathbf{0}_{N \times N} \end{bmatrix} \tag{A.149}$$

with $\dfrac{\partial^2 \mathbf{F}_k}{\partial \mathbf{y}_0 \partial \mathbf{m}} = \mathbf{0}_{(2s-1)N \times N}$ for $k > 1$ and

$$\frac{\partial^2 \mathbf{F}_1}{\partial \mathbf{y}_0 \partial \mathbf{m}} \mathbf{w}_{\mathbf{m}} = \begin{bmatrix} \dfrac{\partial^2 \mathbf{F}_{1,1}}{\partial \mathbf{y}_0 \partial \mathbf{m}} \mathbf{w}_{\mathbf{m}} \\ \vdots \\ \dfrac{\partial^2 \mathbf{F}_{1,2s-1}}{\partial \mathbf{y}_0 \partial \mathbf{m}} \mathbf{w}_{\mathbf{m}} \end{bmatrix},$$

where

$$\frac{\partial^2 \mathbf{F}_{1,i}}{\partial \mathbf{y}_0 \partial \mathbf{m}} \mathbf{w}_\mathbf{m} = \begin{cases} \dfrac{\partial^2 \mathbf{f}^{\mathcal{E}}_{1,\lceil \frac{i}{2} \rceil}}{\partial \mathbf{y} \partial \mathbf{m}} \mathbf{w}_\mathbf{m} & \text{if } \ i \text{ odd} \\ \dfrac{\partial^2 \mathbf{f}^{\mathcal{I}}_{1,\frac{i}{2}+1}}{\partial \mathbf{y} \partial \mathbf{m}} \mathbf{w}_\mathbf{m} & \text{if } \ i \text{ even.} \end{cases}$$

- $\dfrac{\partial^2 \mathbf{t}}{\partial \overline{\mathbf{y}} \partial \mathbf{m}} \mathbf{w}_\mathbf{m}$

Consider the ℓth term of

$$\frac{\partial^2 \mathbf{t}_{k,i}}{\partial \mathbf{y} \partial \mathbf{m}} \mathbf{w}_\mathbf{m} = \left[\frac{\partial^2 \mathbf{t}_{k,i}}{\partial \widehat{\mathbf{y}}_1 \partial \mathbf{m}} \mathbf{w}_\mathbf{m} \quad \frac{\partial^2 \mathbf{t}_{k,i}}{\partial \widehat{\mathbf{y}}_2 \partial \mathbf{m}} \mathbf{w}_\mathbf{m} \quad \cdots \quad \frac{\partial^2 \mathbf{t}_{k,i}}{\partial \widehat{\mathbf{y}}_K \partial \mathbf{m}} \mathbf{w}_\mathbf{m} \right], \tag{A.150}$$

with

$$\frac{\partial^2 \mathbf{t}_{k,i}}{\partial \widehat{\mathbf{y}}_\ell \partial \mathbf{m}} \mathbf{w}_\mathbf{m} = \left[\frac{\partial^2 \mathbf{t}_{k,i}}{\partial \mathbf{Y}^{\mathcal{E}}_{\ell,1} \partial \mathbf{m}} \mathbf{w}_\mathbf{m} \quad \frac{\partial^2 \mathbf{t}_{k,i}}{\partial \mathbf{Y}^{\mathcal{I}}_{\ell,1} \partial \mathbf{m}} \mathbf{w}_\mathbf{m} \quad \cdots \right.$$
$$\left. \cdots \quad \frac{\partial^2 \mathbf{t}_{k,i}}{\partial \mathbf{Y}^{\mathcal{I}}_{\ell,s-1} \partial \mathbf{m}} \mathbf{w}_\mathbf{m} \quad \frac{\partial^2 \mathbf{t}_{k,i}}{\partial \mathbf{Y}^{\mathcal{E}}_{\ell,s} \partial \mathbf{m}} \mathbf{w}_\mathbf{m} \quad \frac{\partial^2 \mathbf{t}_{k,i}}{\partial \mathbf{y}_\ell \partial \mathbf{m}} \mathbf{w}_\mathbf{m} \right].$$

Then, for $\ell = k-1$,

$$\frac{\partial^2 \mathbf{t}_{k,i}}{\partial \widehat{\mathbf{y}}_{k-1} \partial \mathbf{m}} \mathbf{w}_\mathbf{m} = \begin{cases} \left[\mathbf{0}_{N \times (2s-1)N} \quad -\dfrac{\partial^2 \mathbf{f}^{\mathcal{E}}_{k,\lceil \frac{i}{2} \rceil}}{\partial \mathbf{y} \partial \mathbf{m}} \mathbf{w}_\mathbf{m} \right] & \text{if } \ i \leq 2s-1 \\ & \text{and } i \text{ odd} \\ \left[\mathbf{0}_{N \times (2s-1)N} \quad -\dfrac{\partial^2 \mathbf{f}^{\mathcal{I}}_{k,\frac{i}{2}+1}}{\partial \mathbf{y} \partial \mathbf{m}} \mathbf{w}_\mathbf{m} \right] & \text{if } \ i \leq 2(s-1) \\ & \text{and } i \text{ even} \\ \mathbf{0}_{N \times 2sN} & \text{if } \ i = 2s, \end{cases} \tag{A.151a}$$

for $\ell = k$,

$$\frac{\partial^2 \mathbf{t}_{k,i}}{\partial \widehat{\mathbf{y}}_k \partial \mathbf{m}} \mathbf{w}_{\mathbf{m}} = \begin{cases} \begin{bmatrix} \dfrac{\partial^2 \mathbf{f}^{\mathcal{E}}_{k,\lceil \frac{i}{2} \rceil}}{\partial \mathbf{y} \partial \mathbf{m}} \mathbf{w}_{\mathbf{m}} & \mathbf{0}_{N \times N} \\ -\tau_k \mathbf{a}_{\lceil \frac{i}{2} \rceil} \end{bmatrix} & \text{if } \ i \leq 2s-1 \\ & \text{and } i \text{ odd} \\ \begin{bmatrix} \dfrac{\partial^2 \mathbf{f}^{\mathcal{I}}_{k,\frac{i}{2}+1}}{\partial \mathbf{y} \partial \mathbf{m}} \mathbf{w}_{\mathbf{m}} & \mathbf{0}_{N \times N} \\ -\tau_k \mathbf{a}_{\frac{i}{2}} \end{bmatrix} & \text{if } \ i \leq 2(s-1) \\ & \text{and } i \text{ even} \\ \mathbf{0}_{N \times 2sN} & \text{if } \ i = 2s, \end{cases} \tag{A.151b}$$

and

$$\frac{\partial^2 \mathbf{t}_{k,i}}{\partial \widehat{\mathbf{y}}_\ell \partial \mathbf{m}} \mathbf{w}_{\mathbf{m}} = \mathbf{0}_{2sN \times 2sN} \tag{A.151c}$$

otherwise.

Derivatives of $\dfrac{\partial \mathbf{t}}{\partial \mathbf{y}_0}$

We consider the derivatives of the ith $N \times 1$ block $\dfrac{\partial \mathbf{t}_{k,i}}{\partial \mathbf{y}_0} \mathbf{w}_0$ of the vector $\dfrac{\partial \mathbf{t}_k}{\partial \mathbf{y}_0} \mathbf{w}_0$, which in turn is the kth $2sN \times 1$ block of $\mathbf{t}_{\mathbf{y}_0} \mathbf{w}_0 = \dfrac{\partial \mathbf{t}}{\partial \mathbf{y}_0} \mathbf{w}_0$.

The expression for the product $\dfrac{\partial \mathbf{t}_k}{\partial \mathbf{y}_0} \mathbf{w}_0$ was given in (A.141). We now find expressions for the derivatives of (A.141) with respect to $\overline{\mathbf{y}}$, $\mathbf{m}$ and $\mathbf{y}_0$.

- $\dfrac{\partial^2 \mathbf{t}}{\partial \mathbf{m} \partial \mathbf{y}_0} \mathbf{w}_0$

  We differentiate (A.141) with respect to $\mathbf{m}$. Clearly

$$\frac{\partial^2 \mathbf{t}_k}{\partial \mathbf{m} \partial \mathbf{y}_0} \mathbf{w}_0 = \mathbf{0}_{2sN \times N_{\mathbf{m}}} \tag{A.152a}$$

for $k > 1$. For $k = 1$ we have

$$\frac{\partial^2 \mathbf{t}_{1,i}}{\partial \mathbf{m} \partial \mathbf{y}_0} \mathbf{w}_0 = \begin{cases} -\dfrac{\partial^2 \mathbf{f}^{\mathcal{E}}_{1,\lceil \frac{i}{2} \rceil}}{\partial \mathbf{m} \partial \mathbf{y}} \mathbf{w}_0 & \text{if } \ 1 \leq i \leq 2s-1 \text{ and } i \text{ odd} \\ -\dfrac{\partial^2 \mathbf{f}^{\mathcal{I}}_{1,\frac{i}{2}+1}}{\partial \mathbf{m} \partial \mathbf{y}} \mathbf{w}_0 & \text{if } \ 1 \leq i \leq 2(s-1) \text{ and } i \text{ even} \\ \mathbf{0}_{N \times N_{\mathbf{m}}} & \text{if } \ i = 2s \end{cases} \tag{A.152b}$$

- $\dfrac{\partial^2 \mathbf{t}}{\partial \mathbf{y}_0 \partial \mathbf{y}_0} \mathbf{w}_0$

Taking the derivative of (A.141) with respect to $\mathbf{y}_0$ gives

$$\frac{\partial^2 \mathbf{t}_k}{\partial \mathbf{y}_0 \partial \mathbf{y}_0} \mathbf{w}_0 = \mathbf{0}_{2sN \times N} \tag{A.153a}$$

for $k > 1$. For $k = 1$ we have

$$\frac{\partial^2 \mathbf{t}_{1,i}}{\partial \mathbf{m} \partial \mathbf{y}_0} \mathbf{w}_0 = \begin{cases} -\dfrac{\partial^2 \mathbf{f}^{\mathcal{E}}_{1,\lceil \frac{i}{2} \rceil}}{\partial \mathbf{m} \partial \mathbf{y}} \mathbf{w}_0 & \text{if } \ 1 \leq i \leq 2s-1 \text{ and } i \text{ odd} \\ -\dfrac{\partial^2 \mathbf{f}^{\mathcal{I}}_{1,\frac{i}{2}+1}}{\partial \mathbf{m} \partial \mathbf{y}} \mathbf{w}_0 & \text{if } \ 1 \leq i \leq 2(s-1) \text{ and } i \text{ even} \\ \mathbf{0}_{N \times N} & \text{if } \ i = 2s \end{cases} \tag{A.153b}$$

- $\dfrac{\partial^2 \mathbf{t}}{\partial \overline{\mathbf{y}} \partial \mathbf{y}_0} \mathbf{w}_0$

We have $\dfrac{\partial^2 \mathbf{t}_{k,i}}{\partial \mathbf{y} \partial \mathbf{y}} \mathbf{w}_0 = \mathbf{0}_{N \times 2sN}$ for $k > 1$. For $k = 1$, consider the ℓth term of

$$\frac{\partial^2 \mathbf{t}_{1,i}}{\partial \mathbf{y} \partial \mathbf{y}} \mathbf{w}_0 = \left[\frac{\partial^2 \mathbf{t}_{1,i}}{\partial \widehat{\mathbf{y}}_1 \partial \mathbf{y}} \mathbf{w}_0 \quad \frac{\partial^2 \mathbf{t}_{k,i}}{\partial \widehat{\mathbf{y}}_2 \partial \mathbf{y}} \mathbf{w}_0 \quad \cdots \quad \frac{\partial^2 \mathbf{t}_{1,i}}{\partial \widehat{\mathbf{y}}_K \partial \mathbf{y}} \mathbf{w}_0 \right], \tag{A.154}$$

is $\mathbf{0}_{N\times N}$ except for $\ell = 1$, in which case we have

$$\frac{\partial^2 \mathbf{t}_{1,i}}{\partial \widehat{\mathbf{y}}_1 \partial \mathbf{y}} \mathbf{w}_0 = \left[\frac{\partial^2 \mathbf{t}_{1,i}}{\partial \mathbf{Y}^{\mathcal{E}}_{1,1} \partial \mathbf{y}} \mathbf{w}_0 \quad \frac{\partial^2 \mathbf{t}_{1,i}}{\partial \mathbf{Y}^{\mathcal{I}}_{1,1} \partial \mathbf{y}} \mathbf{w}_0 \quad \frac{\partial^2 \mathbf{t}_{1,i}}{\partial \mathbf{Y}^{\mathcal{E}}_{1,s} \partial \mathbf{y}} \mathbf{w}_0 \quad \frac{\partial^2 \mathbf{t}_{1,i}}{\partial \mathbf{y}_1 \partial \mathbf{y}} \mathbf{w}_0 \right],$$

so that

$$\frac{\partial^2 \mathbf{t}_{1,i}}{\partial \widehat{\mathbf{y}}_k \partial \mathbf{y}} \mathbf{w}_0 = \begin{cases} \left[-\tau_k \mathbf{a}_{\lceil \frac{i}{2} \rceil} \dfrac{\partial^2 \mathbf{f}^{\mathcal{E}}_{1,\lceil \frac{i}{2} \rceil}}{\partial \mathbf{y} \partial \mathbf{y}} \mathbf{w}_0 \quad \mathbf{0}_{N\times N} \right] & \text{if } \ i \le 2s-1 \\ & \text{and } i \text{ odd} \\ \left[-\tau_k \mathbf{a}_{\frac{i}{2}} \dfrac{\partial^2 \mathbf{f}^{\mathcal{I}}_{1,\frac{i}{2}+1}}{\partial \mathbf{y} \partial \mathbf{y}} \mathbf{w}_0 \quad \mathbf{0}_{N\times N} \right] & \text{if } \ i \le 2(s-1) \\ & \text{and } i \text{ even} \\ \mathbf{0}_{N\times 2sN} & \text{if } \ i = 2s. \end{cases} \tag{A.155}$$

A.3 Staggered Time-Stepping Methods

A.3.1 Staggered Linear Multistep Methods

The time-stepping vector for a generic s-step StagLM method was found in (2.41) in Section 2.3.1:

$$\mathbf{t}\left(\overline{\mathbf{y}}, \mathbf{m}, \mathbf{y}_0\right) = \mathbf{A}\,\overline{\mathbf{y}} + \boldsymbol{\alpha}\,\mathbf{y}_0 - \tau \mathbf{B}\,\overline{\mathbf{f}} - \tau \boldsymbol{\beta}\,\mathbf{f}_0,$$

where the time-stepping vector $\mathbf{t}$ has length KN. Let $\sigma = \min(k, s)$ and $k^\star = \max(1, k-s)$. The kth time vector is given by the kth subvector of length N:

$$\mathbf{t}_k\left(\overline{\mathbf{y}}, \mathbf{m}, \mathbf{y}_0\right) = \begin{bmatrix} \sum\limits_{j=k^\star}^{k} \alpha^{(\sigma)}_{k-j} \mathbf{u}_k - \tau \sum\limits_{j=k^\star}^{k} \beta^{(\sigma)}_{k-j} \mathbf{f}^{\mathbf{u}}_{k-\frac{1}{2}} \\ \sum\limits_{j=k^\star}^{k} \alpha^{(\sigma)}_{k-j} \mathbf{v}_{k+1/2} - \tau \sum\limits_{j=k^\star}^{k} \beta^{(\sigma)}_{k-j} \mathbf{f}^{\mathbf{v}}_{k} \end{bmatrix}.$$

We have now included the dependence of $\mathbf{t}$ on $\mathbf{m}$; $\overline{\mathbf{y}}$ is independent of $\mathbf{m}$ and $\mathbf{y}_0$. Recall the definitions of the matrices $\mathbf{A}$ and $\mathbf{B}$, defined in (2.1) and (2.3), and $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$, defined as in (2.4). Also recall that $\overline{\mathbf{y}} = \begin{bmatrix} \mathbf{y}_1^\top & \cdots & \mathbf{y}_K^\top \end{bmatrix}^\top$ and $\overline{\mathbf{f}} = \begin{bmatrix} \mathbf{f}_1^\top & \cdots & \mathbf{f}_K^\top \end{bmatrix}^\top$,

with

$$\mathbf{y}_k = \begin{bmatrix} \mathbf{u}_k \\ \mathbf{v}_{k+\frac{1}{2}} \end{bmatrix} \qquad \text{and} \qquad \mathbf{f}_k = \begin{bmatrix} \mathbf{f}^{\mathbf{u}}_{k-\frac{1}{2}} \\ \mathbf{f}^{\mathbf{v}}_k \end{bmatrix}.$$

In this subsection we also let $\overline{\mathbf{w}} = \begin{bmatrix} \mathbf{w}_1^\top & \cdots & \mathbf{w}_K^\top \end{bmatrix}^\top$ be an arbitrary vector of length KN, with $\mathbf{w}_k = \begin{bmatrix} \mathbf{w}_k^{\mathbf{u}\top} & \mathbf{w}_{k+1/2}^{\mathbf{v}\top} \end{bmatrix}^\top$ having length $N = N_{\mathbf{u}} + N_{\mathbf{v}}$. $\mathbf{w}_{\mathbf{m}}$ is an arbitrary vector of length $N_{\mathbf{m}}$, and $\mathbf{w}_0 = \begin{bmatrix} \mathbf{w}_0^{\mathbf{u}\top} & \mathbf{w}_{1/2}^{\mathbf{v}\top} \end{bmatrix}^\top$ is an arbitrary vector of length $N = N_{\mathbf{u}} + N_{\mathbf{v}}$.

Let us now take the derivative of $\mathbf{t} = \mathbf{t}\left(\overline{\mathbf{y}}, \mathbf{m}, \mathbf{y}_0\right)$ with respect to $\overline{\mathbf{y}}$, $\mathbf{m}$ and $\mathbf{y}_0$ in turn:

- $\dfrac{\partial \mathbf{t}}{\partial \overline{\mathbf{y}}}$

  We have

$$\frac{\partial \mathbf{t}}{\partial \overline{\mathbf{y}}} = \begin{bmatrix} \dfrac{\partial \mathbf{t}}{\partial \mathbf{y}_1} & \dfrac{\partial \mathbf{t}}{\partial \mathbf{y}_2} & \cdots & \dfrac{\partial \mathbf{t}}{\partial \mathbf{y}_K} \end{bmatrix} = \mathbf{A} - \tau \mathbf{B} \frac{\partial \overline{\mathbf{f}}}{\partial \overline{\mathbf{y}}}, \tag{A.156}$$

  where

$$\frac{\partial \mathbf{f}_k}{\partial \mathbf{y}_j} = \begin{cases} \begin{bmatrix} \mathbf{0}_{N_{\mathbf{u}} \times N_{\mathbf{u}}} & \mathbf{0}_{N_{\mathbf{u}} \times N_{\mathbf{v}}} \\ \dfrac{\partial \mathbf{f}_k^{\mathbf{v}}}{\partial \mathbf{u}} & \mathbf{0}_{N_{\mathbf{v}} \times N_{\mathbf{v}}} \end{bmatrix} & \text{if } \ j = k \\ \begin{bmatrix} \mathbf{0}_{N_{\mathbf{u}} \times N_{\mathbf{u}}} & \dfrac{\partial \mathbf{f}^{\mathbf{u}}_{k-\frac{1}{2}}}{\partial \mathbf{v}} \\ \mathbf{0}_{N_{\mathbf{v}} \times N_{\mathbf{u}}} & \mathbf{0}_{N_{\mathbf{v}} \times N_{\mathbf{v}}} \end{bmatrix} & \text{if } \ j = k-1. \end{cases} \tag{A.157}$$

  It is easy to see that $\dfrac{\partial \mathbf{t}}{\partial \overline{\mathbf{y}}}$ has the form of (2.1) and (2.3), with the (k, j)th $N \times N$ block being

$$\frac{\partial \mathbf{t}_k}{\partial \mathbf{y}_j} = \begin{cases} \alpha_{k-j}^{(\sigma)} \mathbf{I}_N - \tau \beta_{k-j}^{(\sigma)} \dfrac{\partial \mathbf{f}_j}{\partial \mathbf{y}_j} - \tau \beta_{k-j-1}^{(\sigma)} \dfrac{\partial \mathbf{f}_{j+1}}{\partial \mathbf{y}_j} & \text{if } k^\star \leq j \leq k \\ \mathbf{0}_{N \times N} & \text{otherwise,} \end{cases} \tag{A.158}$$

  with $\beta_\ell^{(\sigma)} = 0$ for $\ell < 0$.

- The product of $\dfrac{\partial \mathbf{t}_k}{\partial \overline{\mathbf{y}}}$ with $\overline{\mathbf{w}}$ is

$$\frac{\partial \mathbf{t}_k}{\partial \overline{\mathbf{y}}}\overline{\mathbf{w}} = \begin{bmatrix} \sum\limits_{j=k^\star-1}^{k-1} \alpha_{k-j}^{(\sigma)} \mathbf{w}_j^{\mathbf{u}} - \tau \sum\limits_{j=k^\star}^{k} \beta_{k-j-1}^{(\sigma)} \dfrac{\partial \mathbf{f}_{j+1/2}^{\mathbf{u}}}{\partial \mathbf{v}} \mathbf{w}_{j+1/2}^{\mathbf{v}} \\ \sum\limits_{j=k^\star}^{k} \alpha_{k-j}^{(\sigma)} \mathbf{w}_{j+1/2}^{\mathbf{v}} - \tau \sum\limits_{j=k^\star}^{k} \beta_{k-j}^{(\sigma)} \dfrac{\partial \mathbf{f}_j^{\mathbf{v}}}{\partial \mathbf{u}} \mathbf{w}_j^{\mathbf{u}} \end{bmatrix}, \tag{A.159}$$

  with $\beta_\ell^{(\sigma)} = 0$ for $\ell < 0$.

- The product of $\dfrac{\partial \mathbf{t}_k}{\partial \mathbf{y}_j}^\top$ with $\mathbf{w}_k$ is

$$\frac{\partial \mathbf{t}_k}{\partial \mathbf{y}_j}^\top \mathbf{w}_k = \begin{bmatrix} \alpha_{k-j}^{(\sigma)} \mathbf{w}_k^{\mathbf{u}} - \tau \beta_{k-j}^{(\sigma)} \dfrac{\partial \mathbf{f}_j^{\mathbf{v}}}{\partial \mathbf{u}}^\top \mathbf{w}_{k+1/2}^{\mathbf{v}} \\ \alpha_{k-j}^{(\sigma)} \mathbf{w}_{k+1/2}^{\mathbf{v}} - \tau \beta_{k-j-1}^{(\sigma)} \dfrac{\partial \mathbf{f}_{j+1/2}^{\mathbf{u}}}{\partial \mathbf{v}}^\top \mathbf{w}_k^{\mathbf{u}} \end{bmatrix} \tag{A.160}$$

  and therefore the product of $\dfrac{\partial \mathbf{t}}{\partial \mathbf{y}_j}^\top$ with $\overline{\mathbf{w}}$ is

$$\frac{\partial \mathbf{t}}{\partial \mathbf{y}_j}^\top \overline{\mathbf{w}} = \begin{bmatrix} \sum\limits_{k=j}^{\min(K,j+s)} \alpha_{k-j}^{(\sigma)} \mathbf{w}_k^{\mathbf{u}} - \tau \sum\limits_{k=j}^{\min(K,j+s)} \beta_{k-j}^{(\sigma)} \dfrac{\partial \mathbf{f}_j^{\mathbf{v}}}{\partial \mathbf{u}}^\top \mathbf{w}_{k+1/2}^{\mathbf{v}} \\ \sum\limits_{k=j}^{\min(K,j+s)} \alpha_{k-j}^{(\sigma)} \mathbf{w}_{k+1/2}^{\mathbf{v}} - \tau \sum\limits_{k=j+1}^{\min(K,j+s)} \beta_{k-j-1}^{(\sigma)} \dfrac{\partial \mathbf{f}_{j+1/2}^{\mathbf{u}}}{\partial \mathbf{v}}^\top \mathbf{w}_k^{\mathbf{u}} \end{bmatrix}. \tag{A.161}$$

- $\dfrac{\partial \mathbf{t}}{\partial \mathbf{m}}$

  Differentiating $\mathbf{t}$ with respect to $\mathbf{m}$ gives

$$\frac{\partial \mathbf{t}}{\partial \mathbf{m}} = -\tau \left(\mathbf{B} \frac{\partial \overline{\mathbf{f}}}{\partial \mathbf{m}} + \boldsymbol{\beta} \frac{\partial \mathbf{f}_0}{\partial \mathbf{m}} \right) \tag{A.162}$$

  with

$$\frac{\partial \overline{\mathbf{f}}}{\partial \mathbf{m}} = \begin{bmatrix} \dfrac{\partial \mathbf{f}_1(\mathbf{m})}{\partial \mathbf{m}}^\top & \cdots & \dfrac{\partial \mathbf{f}_K(\mathbf{m})}{\partial \mathbf{m}}^\top \end{bmatrix}^\top. \tag{A.163}$$

Therefore the kth $N \times N_{\mathbf{m}}$ block of (A.162) is

$$\frac{\partial \mathbf{t}_k}{\partial \mathbf{m}} = \begin{bmatrix} -\tau \sum\limits_{j=k^\star}^{k} \beta_{k-j}^{(\sigma)} \dfrac{\partial \mathbf{f}_{j-\frac{1}{2}}^{\mathbf{u}}}{\partial \mathbf{m}} \\ -\tau \sum\limits_{j=k^\dagger}^{k} \beta_{k-j}^{(\sigma)} \dfrac{\partial \mathbf{f}_{j}^{\mathbf{v}}}{\partial \mathbf{m}} \end{bmatrix} \tag{A.164}$$

with $\beta_k^{(k)} = 0$ if $k > s$ and $k^\dagger = \min(0, k-s)$.

- The product of (A.164) with $\mathbf{w}_{\mathbf{m}}$ is

$$\frac{\partial \mathbf{t}_k}{\partial \mathbf{m}} \mathbf{w}_{\mathbf{m}} = \begin{bmatrix} -\tau \sum\limits_{j=k^\star}^{k} \beta_{k-j}^{(\sigma)} \dfrac{\partial \mathbf{f}_{j-\frac{1}{2}}^{\mathbf{u}}}{\partial \mathbf{m}} \mathbf{w}_{\mathbf{m}} \\ -\tau \sum\limits_{j=k^\dagger}^{k} \beta_{k-j}^{(\sigma)} \dfrac{\partial \mathbf{f}_{j}^{\mathbf{v}}}{\partial \mathbf{m}} \mathbf{w}_{\mathbf{m}} \end{bmatrix} \tag{A.165}$$

  with $\beta_k^{(k)} = 0$ if $k > s$ and $k^\dagger = \min(0, k-s)$.

- The product of the transpose with the k the block of $\mathbf{w}$ is

$$\frac{\partial \mathbf{t}_k}{\partial \mathbf{m}}^\top \mathbf{w}_k = -\tau \sum_{j=k^\star}^{k} \beta_{k-j}^{(\sigma)} \frac{\partial \mathbf{f}_{j-\frac{1}{2}}^{\mathbf{u}}}{\partial \mathbf{m}} \mathbf{w}_k^{\mathbf{u}} - \tau \sum_{j=k^\dagger}^{k} \beta_{k-j}^{(\sigma)} \frac{\partial \mathbf{f}_{j}^{\mathbf{v}}}{\partial \mathbf{m}} \mathbf{w}_{k+1/2}^{\mathbf{v}}. \tag{A.166}$$

  The product of the transpose of $\dfrac{\partial \mathbf{t}_k}{\partial \mathbf{m}}$ with $\overline{\mathbf{w}}$ hence is

$$\frac{\partial \mathbf{t}}{\partial \mathbf{m}}^\top \mathbf{w} = -\tau \sum_{k=1}^{K} \sum_{j=k^\star}^{k} \beta_{k-j}^{(\sigma)} \frac{\partial \mathbf{f}_{j-\frac{1}{2}}^{\mathbf{u}}}{\partial \mathbf{m}} \mathbf{w}_k^{\mathbf{u}} - \tau \sum_{k=1}^{K} \sum_{j=k^\dagger}^{k} \beta_{k-j}^{(\sigma)} \frac{\partial \mathbf{f}_{j}^{\mathbf{v}}}{\partial \mathbf{m}} \mathbf{w}_{k+1/2}^{\mathbf{v}} \tag{A.167}$$

  with $\beta_k^{(k)} = 0$ if $k > s$ and $k^\dagger = \min(0, k-s)$.

- $\dfrac{\partial \mathbf{t}}{\partial \mathbf{y}_0}$

  This derivative $\mathbf{t}$ with respect to $\mathbf{y}_0$ is

$$\frac{\partial \mathbf{t}}{\partial \mathbf{y}_0} = \frac{\partial \mathbf{t}}{\partial \mathbf{y}_0} = \boldsymbol{\alpha} - \tau \mathbf{B} \frac{\partial \overline{\mathbf{f}}}{\partial \mathbf{y}_0} - \tau \boldsymbol{\beta} \frac{\partial \mathbf{f}_0}{\partial \mathbf{y}} \tag{A.168}$$

and the kth $N \times N$ block is

$$\frac{\partial \mathbf{t}_k}{\partial \mathbf{y}_0} = \begin{bmatrix} \alpha_k^{(k)} \mathbf{I}_{N_\mathbf{u}} & -\tau\, \beta_{k-1}^{(k)} \dfrac{\partial \mathbf{f}_{1/2}^\mathbf{u}}{\partial \mathbf{v}} \\ -\tau\, \beta_k^{(k)} \dfrac{\partial \mathbf{f}_0^\mathbf{v}}{\partial \mathbf{u}} & \alpha_k^{(k)} \mathbf{I}_{N_\mathbf{v}} \end{bmatrix} \tag{A.169}$$

with $\alpha_k^{(k)} = \beta_k^{(k)} = 0$ if $k > s$.

- The product of (A.169) with $\mathbf{w}_0$ is

$$\frac{\partial \mathbf{t}_k}{\partial \mathbf{y}_0} \mathbf{w}_0 = \begin{bmatrix} \alpha_k^{(k)} \mathbf{w}_0^\mathbf{u} - \tau\, \beta_{k-1}^{(k)} \dfrac{\partial \mathbf{f}_{1/2}^\mathbf{u}}{\partial \mathbf{v}} \mathbf{w}_{1/2}^\mathbf{v} \\ \alpha_k^{(k)} \mathbf{w}_{1/2}^\mathbf{v} - \tau\, \beta_k^{(k)} \dfrac{\partial \mathbf{f}_0^\mathbf{v}}{\partial \mathbf{u}} \mathbf{w}_0^\mathbf{u} \end{bmatrix}. \tag{A.170}$$

- The product of the transpose with the kth block of $\mathbf{w}$ is

$$\frac{\partial \mathbf{t}_k}{\partial \mathbf{y}_0}^\top \mathbf{w}_k = \begin{bmatrix} \alpha_k^{(k)} \mathbf{w}_k^\mathbf{u} - \tau\, \beta_{k-1}^{(k)} \dfrac{\partial \mathbf{f}_0^\mathbf{v}}{\partial \mathbf{u}}^\top \mathbf{w}_{k+1/2}^\mathbf{v} \\ \alpha_k^{(k)} \mathbf{w}_{k+1/2}^\mathbf{v} - \tau\, \beta_k^{(k)} \dfrac{\partial \mathbf{f}_{1/2}^\mathbf{u}}{\partial \mathbf{v}}^\top \mathbf{w}_k^\mathbf{u} \end{bmatrix}. \tag{A.171}$$

The product of the transpose of $\dfrac{\partial \mathbf{t}}{\partial \mathbf{y}_0}$ with $\mathbf{w}$ therefore is

$$\frac{\partial \mathbf{t}}{\partial \mathbf{y}_0}^\top \mathbf{w} = \begin{bmatrix} \sum\limits_{k=1}^{K} \alpha_k^{(k)} \mathbf{w}_k^\mathbf{u} - \tau \sum\limits_{k=0}^{K} \beta_k^{(k)} \dfrac{\partial \mathbf{f}_0^\mathbf{v}}{\partial \mathbf{u}}^\top \mathbf{w}_{k+1/2}^\mathbf{v} \\ \sum\limits_{k=1}^{K} \alpha_k^{(k)} \mathbf{w}_{k+1/2}^\mathbf{v} - \tau \sum\limits_{k=1}^{K} \beta_k^{(k)} \dfrac{\partial \mathbf{f}_{1/2}^\mathbf{u}}{\partial \mathbf{v}}^\top \mathbf{w}_k^\mathbf{u} \end{bmatrix}. \tag{A.172}$$

Derivatives of $\dfrac{\partial \mathbf{t}}{\partial \overline{\mathbf{y}}}$

We will now find expressions for the derivatives of (A.159),

$$\frac{\partial \mathbf{t}_k}{\partial \overline{\mathbf{y}}} \overline{\mathbf{w}} = \begin{bmatrix} \sum\limits_{j=k^\star}^{k} \alpha_{k-j}^{(\sigma)} \mathbf{w}_j^\mathbf{u} - \tau \sum\limits_{j=k^\star}^{k-1} \beta_{k-j-1}^{(\sigma)} \dfrac{\partial \mathbf{f}_{j+1/2}^\mathbf{u}}{\partial \mathbf{v}} \mathbf{w}_{j+1/2}^\mathbf{v} \\ \sum\limits_{j=k^\star}^{k} \alpha_{k-j}^{(\sigma)} \mathbf{w}_{j+1/2}^\mathbf{v} - \tau \sum\limits_{j=k^\star}^{k} \beta_{k-j}^{(\sigma)} \dfrac{\partial \mathbf{f}_j^\mathbf{v}}{\partial \mathbf{u}} \mathbf{w}_j^\mathbf{u} \end{bmatrix},$$

with respect to $\mathbf{m}$, $\mathbf{y}_0$ and $\overline{\mathbf{y}}$.

- $\dfrac{\partial^2 \mathbf{t}}{\partial \mathbf{m} \partial \overline{\mathbf{y}}} \overline{\mathbf{w}}$

  Differentiating (A.159) with respect to $\mathbf{m}$ gives the kth $N \times N_{\mathbf{m}}$ block of $\dfrac{\partial^2 \mathbf{t}}{\partial \mathbf{m} \partial \overline{\mathbf{y}}} \overline{\mathbf{w}}$:

$$\frac{\partial^2 \mathbf{t}_k}{\partial \mathbf{m} \partial \overline{\mathbf{y}}} \overline{\mathbf{w}} = \begin{bmatrix} -\tau \sum\limits_{j=k^\star}^{k-1} \beta_{k-j-1}^{(\sigma)} \dfrac{\partial^2 \mathbf{f}_{j+1/2}^{\mathbf{u}}}{\partial \mathbf{m} \partial \mathbf{v}} \mathbf{w}_{j+1/2}^{\mathbf{v}} \\ -\tau \sum\limits_{j=k^\star}^{k} \beta_{k-j}^{(\sigma)} \dfrac{\partial^2 \mathbf{f}_{j}^{\mathbf{v}}}{\partial \mathbf{m} \partial \mathbf{u}} \mathbf{w}_{j}^{\mathbf{u}} \end{bmatrix}. \tag{A.173}$$

- $\dfrac{\partial^2 \mathbf{t}}{\partial \mathbf{y}_0 \partial \overline{\mathbf{y}}} \overline{\mathbf{w}}$

  The derivative of (A.159) with respect to $\mathbf{y}_0$ is

$$\frac{\partial^2 \mathbf{t}_k}{\partial \mathbf{y}_0 \partial \overline{\mathbf{y}}} \overline{\mathbf{w}} = \mathbf{0}_{N \times N}. \tag{A.174}$$

- $\dfrac{\partial^2 \mathbf{t}}{\partial \overline{\mathbf{y}} \partial \overline{\mathbf{y}}} \overline{\mathbf{w}}$

  Consider the jth $N \times N$ block of

$$\frac{\partial^2 \mathbf{t}_k}{\partial \overline{\mathbf{y}} \partial \overline{\mathbf{y}}} \overline{\mathbf{w}} = \begin{bmatrix} \dfrac{\partial^2 \mathbf{t}_k}{\partial \mathbf{y}_1 \partial \overline{\mathbf{y}}} \overline{\mathbf{w}} & \dfrac{\partial^2 \mathbf{t}_k}{\partial \mathbf{y}_2 \partial \overline{\mathbf{y}}} \overline{\mathbf{w}} & \cdots & \dfrac{\partial^2 \mathbf{t}_k}{\partial \mathbf{y}_K \partial \overline{\mathbf{y}}} \overline{\mathbf{w}} \end{bmatrix}.$$

  Taking the derivative of (A.159) with respect to $\mathbf{y}_j$ gives the (k, j)th $N \times N$ block of $\dfrac{\partial^2 \mathbf{t}}{\partial \overline{\mathbf{y}} \partial \overline{\mathbf{y}}} \overline{\mathbf{w}}$,

$$\frac{\partial^2 \mathbf{t}_k}{\partial \mathbf{y}_j \partial \overline{\mathbf{y}}} \overline{\mathbf{w}} = \begin{cases} -\tau \begin{bmatrix} \mathbf{0}_{N_{\mathbf{u}} \times N_{\mathbf{u}}} & \beta_{k-j-1}^{(\sigma)} \dfrac{\partial^2 \mathbf{f}_{j+\frac{1}{2}}^{\mathbf{u}}}{\partial \mathbf{v} \partial \mathbf{v}} \mathbf{w}_{j+\frac{1}{2}}^{\mathbf{v}} \\ \beta_{k-j}^{(\sigma)} \dfrac{\partial^2 \mathbf{f}_{j}^{\mathbf{v}}}{\partial \mathbf{u} \partial \mathbf{u}} \mathbf{w}_{j}^{\mathbf{u}} & \mathbf{0}_{N_{\mathbf{v}} \times N_{\mathbf{v}}} \end{bmatrix} & \text{if } k^\star \leq j \leq k \\ \mathbf{0}_{N \times N} & \text{otherwise.} \end{cases} \tag{A.175}$$

Derivatives of $\dfrac{\partial \mathbf{t}}{\partial \mathbf{m}}$

The derivatives of (A.165),

$$\frac{\partial \mathbf{t}_k}{\partial \mathbf{m}}\mathbf{w}_{\mathbf{m}} = \begin{bmatrix} -\tau \sum\limits_{j=k^\star}^{k} \beta_{k-j}^{(\sigma)} \dfrac{\partial \mathbf{f}^{\mathbf{u}}_{j-\frac{1}{2}}}{\partial \mathbf{m}}\mathbf{w}_{\mathbf{m}} \\ -\tau \sum\limits_{j=k^\dagger}^{k} \beta_{k-j}^{(\sigma)} \dfrac{\partial \mathbf{f}^{\mathbf{v}}_{j}}{\partial \mathbf{m}}\mathbf{w}_{\mathbf{m}} \end{bmatrix}$$

where $k^\dagger = \min(0, k-s)$, with respect to $\mathbf{m}$, $\mathbf{y}_0$ and $\overline{\mathbf{y}}$ are:

- $\dfrac{\partial^2 \mathbf{t}}{\partial \mathbf{m} \partial \mathbf{m}}\mathbf{w}_{\mathbf{m}}$

  Taking the derivative of (A.165) with respect to $\mathbf{m}$ gives the kth $N \times N_{\mathbf{m}}$ block of $\dfrac{\partial^2 \mathbf{t}}{\partial \mathbf{m} \partial \mathbf{m}}\mathbf{w}_{\mathbf{m}}$

$$\frac{\partial^2 \mathbf{t}_k}{\partial \mathbf{m} \partial \mathbf{m}}\mathbf{w}_{\mathbf{m}} = \begin{bmatrix} -\tau \sum\limits_{j=k^\star}^{k} \beta_{k-j}^{(\sigma)} \dfrac{\partial^2 \mathbf{f}^{\mathbf{u}}_{j-\frac{1}{2}}}{\partial \mathbf{m} \partial \mathbf{m}}\mathbf{w}_{\mathbf{m}} \\ -\tau \sum\limits_{j=k^\dagger}^{k} \beta_{k-j}^{(\sigma)} \dfrac{\partial^2 \mathbf{f}^{\mathbf{v}}_{j}}{\partial \mathbf{m} \partial \mathbf{m}}\mathbf{w}_{\mathbf{m}} \end{bmatrix}. \tag{A.176}$$

- $\dfrac{\partial^2 \mathbf{t}}{\partial \mathbf{y}_0 \partial \mathbf{m}}\mathbf{w}_{\mathbf{m}}$

  Differentiating (A.165) with respect to $\mathbf{y}_0$ immediately gives

$$\frac{\partial^2 \mathbf{t}_k}{\partial \mathbf{y}_0 \partial \mathbf{m}}\mathbf{w}_{\mathbf{m}} = \begin{bmatrix} \mathbf{0}_{N_{\mathbf{u}} \times N_{\mathbf{u}}} & -\tau \beta_{k-1}^{(\sigma)} \dfrac{\partial^2 \mathbf{f}^{\mathbf{u}}_{\frac{1}{2}}}{\partial \mathbf{v} \partial \mathbf{m}}\mathbf{w}_{\mathbf{m}} \\ \mathbf{0}_{N_{\mathbf{v}} \times N_{\mathbf{u}}} & \mathbf{0}_{N_{\mathbf{v}} \times N_{\mathbf{v}}} \end{bmatrix} \tag{A.177}$$

  with $\beta_k^{(k)} = 0$ if $k > s$.

- $\dfrac{\partial^2 \mathbf{t}}{\partial \overline{\mathbf{y}} \partial \mathbf{m}}\mathbf{w}_{\mathbf{m}}$

  Consider the jth $N \times N$ block of

$$\frac{\partial^2 \mathbf{t}_k}{\partial \overline{\mathbf{y}} \partial \mathbf{m}}\mathbf{w}_{\mathbf{m}} = \begin{bmatrix} \dfrac{\partial^2 \mathbf{t}_k}{\partial \mathbf{y}_1 \partial \mathbf{m}}\mathbf{w}_{\mathbf{m}} & \dfrac{\partial^2 \mathbf{t}_k}{\partial \mathbf{y}_2 \partial \mathbf{m}}\mathbf{w}_{\mathbf{m}} & \cdots & \dfrac{\partial^2 \mathbf{t}_k}{\partial \mathbf{y}_K \partial \mathbf{m}}\mathbf{w}_{\mathbf{m}} \end{bmatrix}. \tag{A.178}$$

The (k,j)th $N \times N$ block of $\dfrac{\partial^2 \mathbf{t}}{\partial \overline{\mathbf{y}} \partial \mathbf{m}} \mathbf{w_m}$ is

$$\frac{\partial^2 \mathbf{t}_k}{\partial \mathbf{y}_j \partial \mathbf{m}} \mathbf{w_m} = \begin{cases} -\tau \begin{bmatrix} \mathbf{0}_{N_\mathbf{u} \times N_\mathbf{u}} & \beta_{k-j-1}^{(\sigma)} \dfrac{\partial^2 \mathbf{f}_{j+1/2}^\mathbf{u}}{\partial \mathbf{v} \partial \mathbf{m}} \mathbf{w_m} \\ \beta_{k-j}^{(\sigma)} \dfrac{\partial^2 \mathbf{f}_j^\mathbf{v}}{\partial \mathbf{u} \partial \mathbf{m}} \mathbf{w_m} & \mathbf{0}_{N_\mathbf{v} \times N_\mathbf{v}} \end{bmatrix} & \text{if } k^\star \leq j \leq k \\ \mathbf{0}_{N \times N} & \text{otherwise.} \end{cases} \tag{A.179}$$

Derivatives of $\dfrac{\partial \mathbf{t}}{\partial \mathbf{y}_0} \mathbf{w}_0$

Finally, we find expressions for the derivatives of (A.170),

$$\frac{\partial \mathbf{t}_k}{\partial \mathbf{y}_0} \mathbf{w}_0 = \begin{bmatrix} \alpha_k^{(k)} \mathbf{w}_0^\mathbf{u} - \tau \beta_{k-1}^{(k)} \dfrac{\partial \mathbf{f}_{1/2}^\mathbf{u}}{\partial \mathbf{v}} \mathbf{w}_{1/2}^\mathbf{v} \\ \alpha_k^{(k)} \mathbf{w}_{1/2}^\mathbf{v} - \tau \beta_k^{(k)} \dfrac{\partial \mathbf{f}_0^\mathbf{v}}{\partial \mathbf{u}} \mathbf{w}_0^\mathbf{u} \end{bmatrix},$$

where $\alpha_k^{(k)} = \beta_k^{(k)} = 0$ if $k > s$, with respect to $\mathbf{m}$, $\mathbf{y}_0$ and $\overline{\mathbf{y}}$:

- $\dfrac{\partial^2 \mathbf{t}}{\partial \mathbf{m} \partial \mathbf{y}_0} \mathbf{w}_0$

  Differentiating (A.170) with respect to $\mathbf{m}$ gives the kth $N \times N_\mathbf{m}$ block of $\dfrac{\partial^2 \mathbf{t}}{\partial \mathbf{m} \partial \mathbf{y}_0} \mathbf{w}_0$:

$$\frac{\partial^2 \mathbf{t}_k}{\partial \mathbf{m} \partial \mathbf{y}_0} \mathbf{w}_0 = -\tau \begin{bmatrix} \beta_{k-1}^{(k)} \dfrac{\partial^2 \mathbf{f}_{1/2}^\mathbf{u}}{\partial \mathbf{m} \partial \mathbf{v}} \mathbf{w}_{1/2}^\mathbf{v} \\ \beta_k^{(k)} \dfrac{\partial^2 \mathbf{f}_0^\mathbf{v}}{\partial \mathbf{m} \partial \mathbf{u}} \mathbf{w}_0^\mathbf{u} \end{bmatrix} \tag{A.180}$$

  with $\beta_k^{(k)} = 0$ if $k > s$.

- $\dfrac{\partial^2 \mathbf{t}}{\partial \mathbf{y}_0 \partial \mathbf{y}_0} \mathbf{w}_0$

  The kth $N \times N$ block of $\dfrac{\partial^2 \mathbf{t}}{\partial \mathbf{y}_0 \partial \mathbf{y}_0} \mathbf{w}_0$ is obtained by taking the derivative of (A.170)

with respect to $\mathbf{y}_0$:

$$\frac{\partial^2 \mathbf{t}_k}{\partial \mathbf{y}_0 \partial \mathbf{y}_0}\mathbf{w}_0 = \begin{bmatrix} \mathbf{0}_{N_\mathbf{u}\times N_\mathbf{u}} & -\tau\,\beta_{k-1}^{(k)}\dfrac{\partial^2 \mathbf{f}_{1/2}^{\mathbf{u}}}{\partial \mathbf{v}\,\partial \mathbf{v}}\mathbf{w}_{1/2}^{\mathbf{v}} \\ -\tau\,\beta_k^{(k)}\dfrac{\partial^2 \mathbf{f}_0^{\mathbf{v}}}{\partial \mathbf{u}\,\partial \mathbf{u}}\mathbf{w}_0^{\mathbf{u}} & \end{bmatrix}, \tag{A.181}$$

with $\beta_k^{(k)} = 0$ if $k > s$.

- $\dfrac{\partial^2 \mathbf{t}}{\partial \overline{\mathbf{y}}\,\partial \mathbf{y}_0}\mathbf{w}_0$

  Differentiating (A.170) with respect to $\overline{\mathbf{y}}$ immediately gives

$$\frac{\partial^2 \mathbf{t}_k}{\partial \overline{\mathbf{y}}\,\partial \mathbf{y}_0}\mathbf{w}_0 = \mathbf{0}_{N\times KN}. \tag{A.182}$$

A.3.2 Staggered Runge-Kutta Methods

The time-stepping vector $\mathbf{t} = \mathbf{t}(\overline{\mathbf{y}}; \mathbf{m}, \mathbf{y}_0)$ for StagRK methods was found in Section 2.3.2 to be

$$\mathbf{t}\left(\overline{\mathbf{y}}; \mathbf{m}, \mathbf{y}_0\right) = \mathbf{t}^{\mathrm{o/e}}\left(\overline{\mathbf{y}}; \mathbf{m}, \mathbf{y}_0\right) = \mathbf{T}\,\overline{\mathbf{y}} - \overline{\mathbf{f}}\left(\overline{\mathbf{y}}, \mathbf{m}, \mathbf{y}_0\right),$$

where

$$\overline{\mathbf{y}} = \overline{\mathbf{y}}^{\mathrm{o/e}} = \begin{bmatrix} \left(\mathbf{Y}_1^{\mathrm{o/e}}\right)^\top & \mathbf{y}_1^\top & \left(\mathbf{Y}_2^{\mathrm{o/e}}\right)^\top & \mathbf{y}_2^\top & \cdots & \left(\mathbf{Y}_K^{\mathrm{o/e}}\right)^\top & \mathbf{y}_K^\top \end{bmatrix}^\top$$

$$\overline{\mathbf{f}} = \overline{\mathbf{f}}^{\mathrm{o/e}} = \begin{bmatrix} \left(\mathbf{F}_1^{\mathrm{o/e}}\right)^\top & \mathbf{y}_0^\top & \left(\mathbf{F}_2^{\mathrm{o/e}}\right)^\top & \mathbf{0}_{1\times N} & \cdots & \left(\mathbf{F}_K^{\mathrm{o/e}}\right)^\top & \mathbf{0}_{1\times N} \end{bmatrix}^\top$$

and

$$\mathbf{T} = \mathbf{T}^{\mathrm{o/e}} = \begin{bmatrix} \mathbf{I}_{sN} & & & & & & \\ \left(\mathbf{B}^{\mathrm{o/e}}\right)^{\top} & \mathbf{I}_N & & & & & \\ & & \mathbf{I}_{sN} & & & & \\ & -\mathbf{I}_N & \left(\mathbf{B}^{\mathrm{o/e}}\right)^{\top} & \mathbf{I}_N & & & \\ & & \ddots & \ddots & \ddots & & \\ & & & & & \mathbf{I}_{sN} & \\ & & & & -\mathbf{I}_N & \left(\mathbf{B}^{\mathrm{o/e}}\right)^{\top} & \mathbf{I}_N \end{bmatrix}, \tag{A.183}$$

see (2.50b). Here we have

$$\begin{aligned} \mathbf{B}^{\mathrm{o}} &= -\tau \begin{bmatrix} b_1 & 0 & b_3 & 0 & \cdots & b_{s-2} & 0 & b_s \end{bmatrix}^{\top} \otimes \mathbf{I}_N, \\ \mathbf{B}^{\mathrm{e}} &= -\tau \begin{bmatrix} 0 & b_2 & 0 & b_4 & \cdots & b_{s-2} & 0 & b_s \end{bmatrix}^{\top} \otimes \mathbf{I}_N, \\ \mathbf{Y}_k^{\mathrm{o/e}} &= \begin{bmatrix} \mathbf{Y}_{k,1}^{\mathrm{o/e}} \\ \vdots \\ \mathbf{Y}_{k,s}^{\mathrm{o/e}} \end{bmatrix}, \qquad \text{and} \qquad \mathbf{F}_k^{\mathrm{o/e}} = \begin{bmatrix} \mathbf{F}_{k,1}^{\mathrm{o/e}} \\ \vdots \\ \mathbf{F}_{k,s}^{\mathrm{o/e}} \end{bmatrix} \end{aligned} \tag{A.184}$$

with

	$\mathbf{F}_{k,\sigma}^{\mathrm{o}}$	$\mathbf{F}_{k,\sigma}^{\mathrm{e}}$
σ odd	$\begin{bmatrix} \mathbf{f}^{\mathbf{u}}\left(\underline{\mathbf{v}}_{\mathbf{U},k-\frac{1}{2},\sigma}^{\mathrm{o}}, t_{k-\frac{1}{2},\sigma}\right) \\ \mathbf{f}^{\mathbf{v}}\left(\underline{\mathbf{u}}_{\mathbf{V},k,\sigma}^{\mathrm{o}}, t_{k,\sigma}\right) \end{bmatrix}$	$\begin{bmatrix} \mathbf{f}^{\mathbf{v}}\left(\underline{\mathbf{u}}_{\mathbf{U},k-1,\sigma}^{\mathrm{e}}, t_{k-1,\sigma}\right) \\ \mathbf{f}^{\mathbf{u}}\left(\underline{\mathbf{v}}_{\mathbf{V},k-\frac{1}{2},\sigma}^{\mathrm{e}}, t_{k-\frac{1}{2},\sigma}\right) \end{bmatrix}$
σ even	$\begin{bmatrix} \mathbf{f}^{\mathbf{v}}\left(\underline{\mathbf{u}}_{\mathbf{U},k-1,\sigma}^{\mathrm{o}}, t_{k-1,\sigma}\right) \\ \mathbf{f}^{\mathbf{u}}\left(\underline{\mathbf{v}}_{\mathbf{V},k-\frac{1}{2},\sigma}^{\mathrm{o}}, t_{k-\frac{1}{2},\sigma}\right) \end{bmatrix}$	$\begin{bmatrix} \mathbf{f}^{\mathbf{u}}\left(\underline{\mathbf{v}}_{\mathbf{U},k-\frac{1}{2},\sigma}^{\mathrm{e}}, t_{k-\frac{1}{2},\sigma}\right) \\ \mathbf{f}^{\mathbf{v}}\left(\underline{\mathbf{u}}_{\mathbf{V},k,\sigma}^{\mathrm{e}}, t_{k,\sigma}\right) \end{bmatrix}$

For brevity of notation we have let $t_{k-1,\sigma} = t_{k-1} + c_\sigma \tau$, $t_{k-\frac{1}{2},\sigma} = t_{k-\frac{1}{2}} + c_\sigma \tau$, and (see (2.44)),

$$\begin{aligned}
\underline{\mathbf{v}}^{o/e}_{\mathbf{U},k-\frac{1}{2},\sigma} &= \mathbf{v}_{k-\frac{1}{2}} + \tau \sum_{\substack{i=1\\ i\ \mathrm{e/o}}}^{\sigma-1} a_{\sigma i} \mathbf{U}^{o/e}_{k,i}\,, & \underline{\mathbf{v}}^{o/e}_{\mathbf{V},k-\frac{1}{2},\sigma} &= \mathbf{v}_{k-\frac{1}{2}} + \tau \sum_{\substack{i=1\\ i\ \mathrm{o/e}}}^{\sigma-1} a_{\sigma i} \mathbf{V}^{o/e}_{k+\frac{1}{2},i} \\
\underline{\mathbf{u}}^{o/e}_{\mathbf{U},k-1,\sigma} &= \mathbf{u}_{k-1} + \tau \sum_{\substack{i=1\\ i\ \mathrm{o/e}}}^{\sigma-1} a_{\sigma i} \mathbf{U}^{o/e}_{k,i}\,, & \underline{\mathbf{u}}^{o/e}_{\mathbf{V},k,\sigma} &= \mathbf{u}_{k} + \tau \sum_{\substack{i=1\\ i\ \mathrm{e/o}}}^{\sigma-1} a_{\sigma i} \mathbf{V}^{o/e}_{k+\frac{1}{2},i}.
\end{aligned} \tag{A.185}$$

With

$$\mathbf{y}_k = \begin{bmatrix} \mathbf{u}_k \\ \mathbf{v}_{k+\frac{1}{2}} \end{bmatrix} \qquad \text{and} \qquad \mathbf{Y}^{\mathrm{o/e}}_{k,\sigma} = \begin{bmatrix} \mathbf{U}^{\mathrm{o/e}}_{k,\sigma} \\ \mathbf{V}^{\mathrm{o/e}}_{k+\frac{1}{2},\sigma} \end{bmatrix},$$

From now on we omit the time dependence of $\mathbf{f^u}$ and $\mathbf{f^v}$. We will now take the derivative of $\mathbf{t}$ with respect to each of $\overline{\mathbf{y}}$, $\mathbf{m}$ and $\mathbf{y}_0$ in turn.

- $\dfrac{\partial \mathbf{t}}{\partial \overline{\mathbf{y}}}$

  We break the derivative of $\mathbf{t}$ with respect to $\overline{\mathbf{y}}$ into parts:

$$\frac{\partial \mathbf{t}}{\partial \overline{\mathbf{y}}} = \begin{bmatrix} \dfrac{\partial \mathbf{t}}{\partial \widehat{\mathbf{y}}_1} & \dfrac{\partial \mathbf{t}}{\partial \widehat{\mathbf{y}}_2} & \cdots & \dfrac{\partial \mathbf{t}}{\partial \widehat{\mathbf{y}}_K} \end{bmatrix} = \mathbf{T} - \begin{bmatrix} \dfrac{\partial \overline{\mathbf{f}}}{\partial \widehat{\mathbf{y}}_1} & \dfrac{\partial \overline{\mathbf{f}}}{\partial \widehat{\mathbf{y}}_2} & \cdots & \dfrac{\partial \overline{\mathbf{f}}}{\partial \widehat{\mathbf{y}}_K} \end{bmatrix}. \tag{A.186}$$

  If

$$\frac{\partial \mathbf{F}^{\mathrm{o/e}}_k}{\partial \widehat{\mathbf{y}}_j} = \begin{bmatrix} \dfrac{\partial \mathbf{F}^{\mathrm{o/e}}_k}{\partial \mathbf{Y}_{j,1}} & \cdots & \dfrac{\partial \mathbf{F}^{\mathrm{o/e}}_k}{\partial \mathbf{Y}_{j,s}} & \dfrac{\partial \mathbf{F}^{\mathrm{o/e}}_k}{\partial \mathbf{y}_j} \end{bmatrix}$$

  denotes the derivative of $\mathbf{F}^{\mathrm{o/e}}_k$ with respect to $\widehat{\mathbf{y}}_j$, with $1 \le k, j \le K$, and $\dfrac{\partial \mathbf{F}^{\mathrm{o/e}}_{k,\sigma}}{\partial \widehat{\mathbf{y}}_{j,i}}$ denotes the (σ, i)th $N \times N$ block of $\dfrac{\partial \mathbf{F}^{\mathrm{o/e}}_k}{\partial \widehat{\mathbf{y}}_j}$, with $1 \le \sigma \le s$ and $1 \le i \le s+1$, then, using the chain rule,

  - *For number of stages s odd*:

    We distinguish between the stage σ being odd or even.

For σ odd we have

$$\frac{\partial \mathbf{F}^{\mathrm{o}}_{k,\sigma}}{\partial \mathbf{U}^{\mathrm{o}}_{k,i}} = \begin{bmatrix} \tau\alpha_{\sigma i}\dfrac{\partial \mathbf{f}^{\mathbf{u}}\left(\underline{\mathbf{v}}^{\mathrm{o}}_{\mathbf{U},k-\frac{1}{2},\sigma}\right)}{\partial \underline{\mathbf{v}}^{\mathrm{o}}_{\mathbf{U},k-\frac{1}{2},\sigma}} \\ \mathbf{0}_{N_{\mathbf{v}}\times N_{\mathbf{u}}} \end{bmatrix}, \quad \frac{\partial \mathbf{F}^{\mathrm{o}}_{k,\sigma}}{\partial \mathbf{V}^{\mathrm{o}}_{k+\frac{1}{2},i}} = \begin{bmatrix} \mathbf{0}_{N_{\mathbf{u}}\times N_{\mathbf{v}}} \\ \tau\alpha_{\sigma i}\dfrac{\partial \mathbf{f}^{\mathbf{v}}\left(\underline{\mathbf{u}}^{\mathrm{o}}_{\mathbf{V},k,\sigma}\right)}{\partial \underline{\mathbf{u}}^{\mathrm{o}}_{\mathbf{V},k,\sigma}} \end{bmatrix} \tag{A.187}$$

and hence, for $j = k$, $1 \leq \sigma \leq s$ and $1 \leq i \leq \sigma - 1$, i even:

$$\begin{aligned} \frac{\partial \mathbf{F}^{\mathrm{o}}_{k,\sigma}}{\partial \mathbf{Y}^{\mathrm{o}}_{k,i}} &= \begin{bmatrix} \dfrac{\partial \mathbf{f}^{\mathbf{u}}\left(\underline{\mathbf{v}}^{\mathrm{o}}_{\mathbf{U},k-\frac{1}{2},\sigma}\right)}{\partial \mathbf{U}^{\mathrm{o}}_{k,i}} & \dfrac{\partial \mathbf{f}^{\mathbf{u}}\left(\underline{\mathbf{v}}^{\mathrm{o}}_{\mathbf{U},k-\frac{1}{2},\sigma}\right)}{\partial \mathbf{V}^{\mathrm{o}}_{k+\frac{1}{2},i}} \\ \dfrac{\partial \mathbf{f}^{\mathbf{v}}\left(\underline{\mathbf{u}}^{\mathrm{o}}_{\mathbf{V},k,\sigma}\right)}{\partial \mathbf{U}^{\mathrm{o}}_{k,i}} & \dfrac{\partial \mathbf{f}^{\mathbf{v}}\left(\underline{\mathbf{u}}^{\mathrm{o}}_{\mathbf{V},k,\sigma}\right)}{\partial \mathbf{V}^{\mathrm{o}}_{k+\frac{1}{2},i}} \end{bmatrix} \\ &= \begin{bmatrix} \dfrac{\partial \mathbf{f}^{\mathbf{u}}\left(\underline{\mathbf{v}}^{\mathrm{o}}_{\mathbf{U},k-\frac{1}{2},\sigma}\right)}{\partial \mathbf{U}^{\mathrm{o}}_{k,i}} & \mathbf{0}_{N_{\mathbf{u}}\times N_{\mathbf{v}}} \\ \mathbf{0}_{N_{\mathbf{v}}\times N_{\mathbf{u}}} & \dfrac{\partial \mathbf{f}^{\mathbf{v}}\left(\underline{\mathbf{u}}^{\mathrm{o}}_{\mathbf{V},k,\sigma}\right)}{\partial \mathbf{V}^{\mathrm{o}}_{k+\frac{1}{2},i}} \end{bmatrix} \\ &= \tau\alpha_{\sigma i} \underbrace{\begin{bmatrix} \mathbf{f}^{\mathbf{u}}_{\mathbf{v}}\left(\underline{\mathbf{v}}^{\mathrm{o}}_{\mathbf{U},k-\frac{1}{2},\sigma}\right) & \mathbf{0}_{N_{\mathbf{v}}\times N_{\mathbf{u}}} \\ \mathbf{0}_{N_{\mathbf{u}}\times N_{\mathbf{v}}} & \mathbf{f}^{\mathbf{v}}_{\mathbf{u}}\left(\underline{\mathbf{u}}^{\mathrm{o}}_{\mathbf{V},k,\sigma}\right) \end{bmatrix}}_{\mathbf{G}^{\mathrm{o}}_{k,\sigma}} \end{aligned}$$

and $\dfrac{\partial \mathbf{F}^{\mathrm{o}}_{k,\sigma}}{\partial \mathbf{Y}^{\mathrm{o}}_{j,i}} = \mathbf{0}_{N\times N}$ otherwise.

Also,

$$\frac{\partial \mathbf{F}^{\mathrm{o}}_{k,\sigma}}{\partial \mathbf{y}_j} = \begin{bmatrix} \dfrac{\partial \mathbf{f}^{\mathbf{u}}\left(\underline{\mathbf{v}}^{\mathrm{o}}_{\mathbf{U},k-\frac{1}{2},\sigma}\right)}{\partial \mathbf{u}_j} & \dfrac{\partial \mathbf{f}^{\mathbf{u}}\left(\underline{\mathbf{v}}^{\mathrm{o}}_{\mathbf{U},k-\frac{1}{2},\sigma}\right)}{\partial \mathbf{v}_{j+\frac{1}{2}}} \\ \dfrac{\partial \mathbf{f}^{\mathbf{v}}\left(\underline{\mathbf{u}}^{\mathrm{o}}_{\mathbf{V},k,\sigma}\right)}{\partial \mathbf{u}_j} & \dfrac{\partial \mathbf{f}^{\mathbf{v}}\left(\underline{\mathbf{u}}^{\mathrm{o}}_{\mathbf{V},k,\sigma}\right)}{\partial \mathbf{v}_{j+\frac{1}{2}}} \end{bmatrix}, \tag{A.188}$$

so

$$\frac{\partial \mathbf{F}^{\mathrm{o}}_{k,\sigma}}{\partial \mathbf{y}_j} = \begin{cases} \begin{bmatrix} \mathbf{0}_{N_\mathbf{u}\times N_\mathbf{u}} & \mathbf{0}_{N_\mathbf{u}\times N_\mathbf{v}} \\ \mathbf{f}^{\mathbf{v}}_{\mathbf{u}}\left(\underline{\mathbf{u}}^{\mathrm{o}}_{\mathbf{V},k,\sigma}\right) & \mathbf{0}_{N_\mathbf{v}\times N_\mathbf{v}} \end{bmatrix} & \text{if } j = k, \\ \begin{bmatrix} \mathbf{0}_{N_\mathbf{u}\times N_\mathbf{u}} & \mathbf{f}^{\mathbf{u}}_{\mathbf{v}}\left(\underline{\mathbf{v}}^{\mathrm{o}}_{\mathbf{U},k-\frac{1}{2},\sigma}\right) \\ \mathbf{0}_{N_\mathbf{v}\times N_\mathbf{u}} & \mathbf{0}_{N_\mathbf{v}\times N_\mathbf{v}} \end{bmatrix} & \text{if } j = k-1 \\ \mathbf{0}_{N\times N} & \text{otherwise.} \end{cases} \tag{A.189}$$

Similarly, for σ even:

$$\frac{\partial \mathbf{F}^{\mathrm{o}}_{k,\sigma}}{\partial \mathbf{Y}^{\mathrm{o}}_{k,i}} = \begin{cases} \tau\alpha_{\sigma i} \underbrace{\begin{bmatrix} \mathbf{f}^{\mathbf{v}}_{\mathbf{u}}\left(\underline{\mathbf{u}}^{\mathrm{o}}_{\mathbf{U},k-1,\sigma}\right) & \mathbf{0}_{N_\mathbf{v}\times N_\mathbf{v}} \\ \mathbf{0}_{N_\mathbf{u}\times N_\mathbf{u}} & \mathbf{f}^{\mathbf{u}}_{\mathbf{v}}\left(\underline{\mathbf{v}}^{\mathrm{o}}_{\mathbf{V},k-\frac{1}{2},\sigma}\right) \end{bmatrix}}_{\mathbf{G}^{\mathrm{o}}_{k,\sigma}} & \text{if } 1 \le i \le \sigma-1, \\ & \quad i \text{ odd} \\ \mathbf{0}_{N\times N} & \text{otherwise,} \end{cases} \tag{A.190a}$$

and

$$\frac{\partial \mathbf{F}^{\mathrm{o}}_{k,\sigma}}{\partial \mathbf{y}_j} = \begin{cases} \begin{bmatrix} \mathbf{f}^{\mathbf{v}}_{\mathbf{u}}\left(\underline{\mathbf{u}}^{\mathrm{o}}_{\mathbf{U},k-1,\sigma}\right) & \mathbf{0}_{N_\mathbf{v}\times N_\mathbf{v}} \\ \mathbf{0}_{N_\mathbf{u}\times N_\mathbf{u}} & \mathbf{f}^{\mathbf{u}}_{\mathbf{v}}\left(\underline{\mathbf{v}}^{\mathrm{o}}_{\mathbf{V},k-\frac{1}{2},\sigma}\right) \end{bmatrix} & \text{if } j = k-1 \\ \mathbf{0}_{N\times N} & \text{otherwise.} \end{cases} \tag{A.190b}$$

- *For number of stages s even*:

  We again differentiate between the stage σ being odd or even. Following the same procedure as for s odd, we get, for σ odd,

$$\frac{\partial \mathbf{F}^{\mathrm{e}}_{k,\sigma}}{\partial \mathbf{Y}^{\mathrm{e}}_{k,i}} = \begin{cases} \tau\alpha_{\sigma i} \underbrace{\begin{bmatrix} \mathbf{f}^{\mathbf{v}}_{\mathbf{u}}\left(\underline{\mathbf{u}}^{\mathrm{e}}_{\mathbf{U},k-1,\sigma}\right) & \mathbf{0}_{N_\mathbf{v}\times N_\mathbf{v}} \\ \mathbf{0}_{N_\mathbf{u}\times N_\mathbf{u}} & \mathbf{f}^{\mathbf{u}}_{\mathbf{v}}\left(\underline{\mathbf{v}}^{\mathrm{e}}_{\mathbf{V},k-\frac{1}{2},\sigma}\right) \end{bmatrix}}_{\mathbf{G}^{\mathrm{e}}_{k,\sigma}} & \text{if } 1 \le i \le \sigma-1, \\ & \quad i \text{ even} \\ \mathbf{0}_{N\times N} & \text{otherwise,} \end{cases} \tag{A.191a}$$

and

$$\frac{\partial \mathbf{F}^{\mathrm{e}}_{k,\sigma}}{\partial \mathbf{y}_j} = \begin{cases} \begin{bmatrix} \mathbf{f}^{\mathbf{v}}_{\mathbf{u}}\left(\underline{\mathbf{u}}^{\mathrm{e}}_{\mathbf{U},k-1,\sigma}\right) & \mathbf{0}_{N_{\mathbf{v}}\times N_{\mathbf{v}}} \\ \mathbf{0}_{N_{\mathbf{u}}\times N_{\mathbf{u}}} & \mathbf{f}^{\mathbf{u}}_{\mathbf{v}}\left(\underline{\mathbf{v}}^{\mathrm{e}}_{\mathbf{V},k-\frac{1}{2},\sigma}\right) \end{bmatrix} & \text{if } j = k-1 \\ \mathbf{0}_{N\times N} & \text{otherwise,} \end{cases} \tag{A.191b}$$

and for σ even,

$$\frac{\partial \mathbf{F}^{\mathrm{e}}_{k,\sigma}}{\partial \mathbf{Y}^{\mathrm{e}}_{k,i}} = \begin{cases} \tau\alpha_{\sigma i} \underbrace{\begin{bmatrix} \mathbf{f}^{\mathbf{u}}_{\mathbf{v}}\left(\underline{\mathbf{v}}^{\mathrm{e}}_{\mathbf{U},k-\frac{1}{2},\sigma}\right) & \mathbf{0}_{N_{\mathbf{v}}\times N_{\mathbf{u}}} \\ \mathbf{0}_{N_{\mathbf{u}}\times N_{\mathbf{v}}} & \mathbf{f}^{\mathbf{v}}_{\mathbf{u}}\left(\underline{\mathbf{u}}^{\mathrm{e}}_{\mathbf{V},k,\sigma}\right) \end{bmatrix}}_{\mathbf{G}^{\mathrm{e}}_{k,\sigma}} & \text{if } 1 \le i \le \sigma-1, \\ & i \text{ odd} \\ \mathbf{0}_{N\times N} & \text{otherwise,} \end{cases} \tag{A.192a}$$

and

$$\frac{\partial \mathbf{F}^{\mathrm{e}}_{k,\sigma}}{\partial \mathbf{y}_j} = \begin{cases} \begin{bmatrix} \mathbf{0}_{N_{\mathbf{u}}\times N_{\mathbf{u}}} & \mathbf{0}_{N_{\mathbf{u}}\times N_{\mathbf{v}}} \\ \mathbf{f}^{\mathbf{v}}_{\mathbf{u}}\left(\underline{\mathbf{u}}^{\mathrm{e}}_{\mathbf{V},k,\sigma}\right) & \mathbf{0}_{N_{\mathbf{v}}\times N_{\mathbf{v}}} \end{bmatrix} & \text{if } j = k, \\ \begin{bmatrix} \mathbf{0}_{N_{\mathbf{u}}\times N_{\mathbf{u}}} & \mathbf{f}^{\mathbf{u}}_{\mathbf{v}}\left(\underline{\mathbf{v}}^{\mathrm{e}}_{\mathbf{U},k-\frac{1}{2},\sigma}\right) \\ \mathbf{0}_{N_{\mathbf{v}}\times N_{\mathbf{u}}} & \mathbf{0}_{N_{\mathbf{v}}\times N_{\mathbf{v}}} \end{bmatrix} & \text{if } j = k-1 \\ \mathbf{0}_{N\times N} & \text{otherwise.} \end{cases} \tag{A.192b}$$

For s odd and even, the stages are combined by letting

$$\mathbf{G}^{\mathrm{o/e}}_k = \begin{bmatrix} \mathbf{G}^{\mathrm{o/e}}_{k,1} & & & \\ & \mathbf{G}^{\mathrm{o/e}}_{k,2} & & \\ & & \ddots & \\ & & & \mathbf{G}^{\mathrm{o/e}}_{k,s} \end{bmatrix}, \qquad \frac{\partial \mathbf{F}^{\mathrm{o/e}}_k}{\partial \mathbf{y}_j} = \begin{bmatrix} \dfrac{\partial \mathbf{F}^{\mathrm{o/e}}_{k,1}}{\partial \mathbf{y}_j} \\ \vdots \\ \dfrac{\partial \mathbf{F}^{\mathrm{o/e}}_{k,s}}{\partial \mathbf{y}_j} \end{bmatrix}. \tag{A.193}$$

The (k,j)th $(s+1)N \times (s+1)N$ block of $\dfrac{\partial \mathbf{t}}{\partial \mathbf{y}}$ is

$$\frac{\partial \mathbf{t}_k}{\partial \widehat{\mathbf{y}}_j} = \begin{cases} \begin{bmatrix} \mathbf{I}_{sN} - \tau \mathbf{G}_k^{\mathrm{o/e}} \left(\mathbf{A}_s \otimes \mathbf{I}_N\right) & -\dfrac{\partial \mathbf{F}_k^{\mathrm{o/e}}}{\partial \mathbf{y}_k} \\ \mathbf{B}^\top & \mathbf{I}_N \end{bmatrix} & \text{if } k = j \\ \begin{bmatrix} \mathbf{0}_{sN \times sN} & -\dfrac{\partial \mathbf{F}_k^{\mathrm{o/e}}}{\partial \mathbf{y}_{k-1}} \\ \mathbf{0}_{N \times sN} & -\mathbf{I}_N \end{bmatrix} & \text{if } j = k-1 \\ \mathbf{0}_{(s+1)N \times (s+1)N} & \text{otherwise.} \end{cases} \tag{A.194}$$

With

$$\begin{aligned} \mathbf{A}_k &= \mathbf{I}_{sN} - \tau \mathbf{G}_k^{\mathrm{o/e}} \left(\mathbf{A}_s \otimes \mathbf{I}_N\right) \\ \mathbf{C}_{k,j} &= -\frac{\partial \mathbf{F}_k^{\mathrm{o/e}}}{\partial \mathbf{y}_j} \\ \mathbf{B} &= \mathbf{B}^{\mathrm{o/e}}, \end{aligned} \tag{A.195}$$

and we can then write

$$\frac{\partial \mathbf{t}}{\partial \overline{\mathbf{y}}} = \begin{bmatrix} \mathbf{A}_1 & \mathbf{C}_{1,1} & & & & & \\ \mathbf{B}^\top & \mathbf{I}_N & & & & & \\ & \mathbf{C}_{2,1} & \mathbf{A}_2 & \mathbf{C}_{2,2} & & & \\ & -\mathbf{I}_N & \mathbf{B}^\top & \mathbf{I}_N & & & \\ & & \ddots & \ddots & \ddots & & \\ & & & & \mathbf{C}_{K,K-1} & \mathbf{A}_K & \mathbf{C}_{K,K} \\ & & & & -\mathbf{I}_N & \mathbf{B}^\top & \mathbf{I}_N \end{bmatrix}. \tag{A.196}$$

which is essentially lower block-triangular matrix since the variables can be rearranged to give a lower block-triangular matrix.

- The kth $(s+1)N \times 1$ block of the product $\mathbf{t}_{\mathbf{y}}\overline{\mathbf{w}}$ is

$$\frac{\partial \mathbf{t}_k}{\partial \overline{\mathbf{y}}}\overline{\mathbf{w}} = \begin{bmatrix} \mathbf{C}_{k,k}\mathbf{w}_k + \mathbf{A}_k\mathbf{W}_k + \mathbf{C}_{k,k-1}\mathbf{w}_{k-1} \\ \mathbf{B}^\top\mathbf{W}_k + \mathbf{w}_k - \mathbf{w}_{k-1} \end{bmatrix} \tag{A.197}$$

with $\mathbf{w}_0 = \mathbf{0}_{N\times 1}$. If $\dfrac{\partial \mathbf{t}_{k,\sigma}}{\partial \overline{\mathbf{y}}}\overline{\mathbf{w}}$ is the σth $N \times 1$ block of $\dfrac{\partial \mathbf{t}_k}{\partial \overline{\mathbf{y}}}\overline{\mathbf{w}}$, with $1 \le \sigma \le s+1$, we have

- *For number of stages s odd*

  $1 \le \sigma \le s$ and σ odd:

$$\frac{\partial \mathbf{t}_{k,\sigma}}{\partial \overline{\mathbf{y}}}\overline{\mathbf{w}} = \begin{bmatrix} \mathbf{W}^{\mathbf{u}}_{k,\sigma} - \mathbf{f}^{\mathbf{u}}_{\mathbf{v}}\left(\underline{\mathbf{v}}^{\mathrm{o}}_{\mathbf{U},k-\frac{1}{2},\sigma}\right)\left(\mathbf{w}^{\mathbf{v}}_{k-\frac{1}{2}} + \tau \sum\limits_{\substack{i=1 \\ i \text{ even}}}^{\sigma-1} a_{\sigma i}\mathbf{W}^{\mathbf{u}}_{k,i}\right) \\ \mathbf{W}^{\mathbf{v}}_{k+\frac{1}{2},\sigma} - \mathbf{f}^{\mathbf{v}}_{\mathbf{u}}\left(\underline{\mathbf{u}}^{\mathrm{o}}_{\mathbf{V},k,\sigma}\right)\left(\mathbf{w}^{\mathbf{u}}_{k} + \tau \sum\limits_{\substack{i=1 \\ i \text{ even}}}^{\sigma-1} a_{\sigma i}\mathbf{W}^{\mathbf{v}}_{k+\frac{1}{2},i}\right) \end{bmatrix}. \tag{A.198a}$$

  $1 \le \sigma \le s$ and σ even:

$$\frac{\partial \mathbf{t}_{k,\sigma}}{\partial \overline{\mathbf{y}}}\overline{\mathbf{w}} = \begin{bmatrix} \mathbf{W}^{\mathbf{u}}_{k,\sigma} - \mathbf{f}^{\mathbf{v}}_{\mathbf{u}}\left(\underline{\mathbf{u}}^{\mathrm{o}}_{\mathbf{U},k-1,\sigma}\right)\left(\mathbf{w}^{\mathbf{u}}_{k-1} + \tau \sum\limits_{\substack{i=1 \\ i \text{ odd}}}^{\sigma-1} a_{\sigma i}\mathbf{W}^{\mathbf{u}}_{k,i}\right) \\ \mathbf{W}^{\mathbf{v}}_{k+\frac{1}{2},\sigma} - \mathbf{f}^{\mathbf{u}}_{\mathbf{v}}\left(\underline{\mathbf{v}}^{\mathrm{o}}_{\mathbf{V},k-\frac{1}{2},\sigma}\right)\left(\mathbf{w}^{\mathbf{v}}_{k-\frac{1}{2}} + \tau \sum\limits_{\substack{i=1 \\ i \text{ odd}}}^{\sigma-1} a_{\sigma i}\mathbf{W}^{\mathbf{v}}_{k+\frac{1}{2},i}\right) \end{bmatrix}. \tag{A.198b}$$

  $\sigma = s+1$:

$$\frac{\partial \mathbf{t}_{k,s+1}}{\partial \overline{\mathbf{y}}}\overline{\mathbf{w}} = \begin{bmatrix} -\tau \sum\limits_{\substack{\sigma=1 \\ \sigma \text{ odd}}}^{s} b_\sigma \mathbf{W}^{\mathbf{u}}_{k,\sigma} + \mathbf{w}^{\mathbf{u}}_{k} - \mathbf{w}^{\mathbf{u}}_{k-1} \\ -\tau \sum\limits_{\substack{\sigma=1 \\ \sigma \text{ odd}}}^{s} b_\sigma \mathbf{W}^{\mathbf{v}}_{k+\frac{1}{2},\sigma} + \mathbf{w}^{\mathbf{v}}_{k+\frac{1}{2}} - \mathbf{w}^{\mathbf{v}}_{k-\frac{1}{2}} \end{bmatrix}. \tag{A.198c}$$

- *For number of stages s even*

$1 \leq \sigma \leq s$ and σ odd:

$$\frac{\partial \mathbf{t}_{k,\sigma}}{\partial \overline{\mathbf{y}}} \overline{\mathbf{w}} = \begin{bmatrix} \mathbf{W}^{\mathbf{u}}_{k,\sigma} - \mathbf{f}^{\mathbf{v}}_{\mathbf{u}} \left(\underline{\mathbf{u}}^{\mathrm{e}}_{\mathbf{U},k-1,\sigma} \right) \left(\mathbf{w}^{\mathbf{u}}_{k-1} + \tau \sum\limits_{\substack{i=1 \\ i \text{ even}}}^{\sigma-1} a_{\sigma i} \mathbf{W}^{\mathbf{u}}_{k,i} \right) \\ \mathbf{W}^{\mathbf{v}}_{k+\frac{1}{2},\sigma} - \mathbf{f}^{\mathbf{u}}_{\mathbf{v}} \left(\underline{\mathbf{v}}^{\mathrm{e}}_{\mathbf{V},k-\frac{1}{2},\sigma} \right) \left(\mathbf{w}^{\mathbf{v}}_{k-\frac{1}{2}} + \tau \sum\limits_{\substack{i=1 \\ i \text{ even}}}^{\sigma-1} a_{\sigma i} \mathbf{W}^{\mathbf{v}}_{k+\frac{1}{2},i} \right) \end{bmatrix}. \tag{A.198d}$$

$1 \leq \sigma \leq s$ and σ even:

$$\frac{\partial \mathbf{t}_{k,\sigma}}{\partial \overline{\mathbf{y}}} \overline{\mathbf{w}} = \begin{bmatrix} \mathbf{W}^{\mathbf{u}}_{k,\sigma} - \mathbf{f}^{\mathbf{u}}_{\mathbf{v}} \left(\underline{\mathbf{v}}^{\mathrm{e}}_{\mathbf{U},k-\frac{1}{2},\sigma} \right) \left(\mathbf{w}^{\mathbf{v}}_{k-\frac{1}{2}} + \tau \sum\limits_{\substack{i=1 \\ i \text{ odd}}}^{\sigma-1} a_{\sigma i} \mathbf{W}^{\mathbf{u}}_{k,i} \right) \\ \mathbf{W}^{\mathbf{v}}_{k+\frac{1}{2},\sigma} - \mathbf{f}^{\mathbf{v}}_{\mathbf{u}} \left(\underline{\mathbf{u}}^{\mathrm{e}}_{\mathbf{V},k,\sigma} \right) \left(\mathbf{w}^{\mathbf{u}}_{k} + \tau \sum\limits_{\substack{i=1 \\ i \text{ odd}}}^{\sigma-1} a_{\sigma i} \mathbf{W}^{\mathbf{v}}_{k+\frac{1}{2},i} \right) \end{bmatrix}. \tag{A.198e}$$

$\sigma = s + 1$:

$$\frac{\partial \mathbf{t}_{k,s+1}}{\partial \overline{\mathbf{y}}} \overline{\mathbf{w}} = \begin{bmatrix} -\tau \sum\limits_{\substack{\sigma=1 \\ \sigma \text{ even}}}^{s} b_\sigma \mathbf{W}^{\mathbf{u}}_{k,\sigma} + \mathbf{w}^{\mathbf{u}}_{k} - \mathbf{w}^{\mathbf{u}}_{k-1} \\ -\tau \sum\limits_{\substack{\sigma=1 \\ \sigma \text{ even}}}^{s} b_\sigma \mathbf{W}^{\mathbf{v}}_{k+\frac{1}{2},\sigma} + \mathbf{w}^{\mathbf{v}}_{k+\frac{1}{2}} - \mathbf{w}^{\mathbf{v}}_{k-\frac{1}{2}} \end{bmatrix}. \tag{A.198f}$$

From now on we let

$$\underline{\mathbf{w}}^{\mathbf{v},\mathrm{o/e}}_{\mathbf{W}^{\mathbf{u}},k-\frac{1}{2},\sigma} = \mathbf{w}^{\mathbf{v}}_{k-\frac{1}{2}} + \tau \sum_{\substack{i=1 \\ i \text{ e/o}}}^{\sigma-1} a_{\sigma i} \mathbf{W}^{\mathbf{u}}_{k,i} \qquad \underline{\mathbf{w}}^{\mathbf{u},\mathrm{o/e}}_{\mathbf{W}^{\mathbf{v}},k,\sigma} = \mathbf{w}^{\mathbf{u}}_{k} + \tau \sum_{\substack{i=1 \\ i \text{ e/o}}}^{\sigma-1} a_{\sigma i} \mathbf{W}^{\mathbf{v}}_{k+\frac{1}{2},i}$$

$$\underline{\mathbf{w}}^{\mathbf{u},\mathrm{o/e}}_{\mathbf{W}^{\mathbf{u}},k-1,\sigma} = \mathbf{w}^{\mathbf{u}}_{k-1} + \tau \sum_{\substack{i=1 \\ i \text{ o/e}}}^{\sigma-1} a_{\sigma i} \mathbf{W}^{\mathbf{u}}_{k,i} \qquad \underline{\mathbf{w}}^{\mathbf{v},\mathrm{o/e}}_{\mathbf{W}^{\mathbf{v}},k-\frac{1}{2},\sigma} = \mathbf{w}^{\mathbf{v}}_{k-\frac{1}{2}} + \tau \sum_{\substack{i=1 \\ i \text{ o/e}}}^{\sigma-1} a_{\sigma i} \mathbf{W}^{\mathbf{v}}_{k+\frac{1}{2},i}. \tag{A.199}$$

◦ The transpose of $\dfrac{\partial \mathbf{t}}{\partial \overline{\mathbf{y}}}$ is upper block-triangular:

$$\frac{\partial \mathbf{t}}{\partial \overline{\mathbf{y}}}^{\top} = \begin{bmatrix} \mathbf{A}_1^\top & \mathbf{B} & & & & & \\ \mathbf{C}_{1,1}^\top & \mathbf{I}_N & \mathbf{C}_{2,1}^\top & -\mathbf{I}_N & & & \\ & & \mathbf{A}_2^\top & \mathbf{B} & & & \\ & & \ddots & \ddots & \ddots & \ddots & \\ & & & \mathbf{C}_{K-1,K-1}^\top & \mathbf{I}_N & \mathbf{C}_{K,K-1}^\top & -\mathbf{I}_N \\ & & & & & \mathbf{A}_K^\top & \mathbf{B} \\ & & & & & \mathbf{C}_{K,K}^\top & \mathbf{I}_N \end{bmatrix}. \tag{A.200}$$

The jth $(s+1)N\times 1$ block of the product of $\dfrac{\partial \mathbf{t}}{\partial \overline{\mathbf{y}}}^{\top}$ with some arbitrary $\overline{\mathbf{w}}$ therefore is

$$\frac{\partial \mathbf{t}}{\partial \widehat{\mathbf{y}}_j}^{\top} \overline{\mathbf{w}} = \begin{bmatrix} \mathbf{A}_j^\top \mathbf{W}_j + \mathbf{B}\mathbf{w}_j \\ \mathbf{C}_{j,j}^\top \mathbf{W}_j + \mathbf{C}_{j+1,j}^\top \mathbf{W}_{j+1} + \mathbf{w}_j - \mathbf{w}_{j+1} \end{bmatrix}, \tag{A.201}$$

with $\mathbf{W}_{K+1,\sigma} = \mathbf{w}_{K+1} = \mathbf{0}_{N\times 1}$. The σth $N \times 1$ block $\dfrac{\partial \mathbf{t}}{\partial \widehat{\mathbf{y}}_{j,\sigma}}^{\top} \overline{\mathbf{w}}$ of $\dfrac{\partial \mathbf{t}}{\partial \widehat{\mathbf{y}}_j}^{\top} \overline{\mathbf{w}}$, with $1 \le \sigma \le s+1$, we have

□ *For number of stages s odd*

$1 \le \sigma \le s$ and σ odd:

$$\frac{\partial \mathbf{t}}{\partial \widehat{\mathbf{y}}_{j,\sigma}}^{\top} \overline{\mathbf{w}} = \begin{bmatrix} \mathbf{W}_{j,\sigma}^{\mathbf{u}} - \tau \sum\limits_{\substack{i=\sigma+1 \\ i \text{ even}}}^{s} a_{i\sigma} \mathbf{f}_{\mathbf{u}}^{\mathbf{v}} \left(\underline{\mathbf{u}}_{\mathbf{U},j-1,i}^{\mathrm{o}}\right)^{\top} \mathbf{W}_{j,i}^{\mathbf{u}} - \tau b_\sigma \mathbf{w}_j^{\mathbf{u}} \\ \mathbf{W}_{j+\frac{1}{2},\sigma}^{\mathbf{v}} - \tau \sum\limits_{\substack{i=\sigma+1 \\ i \text{ even}}}^{s} a_{i\sigma} \mathbf{f}_{\mathbf{v}}^{\mathbf{u}} \left(\underline{\mathbf{v}}_{\mathbf{V},j-\frac{1}{2},\sigma}^{\mathrm{o}}\right)^{\top} \mathbf{W}_{j+\frac{1}{2},i}^{\mathbf{v}} - \tau b_\sigma \mathbf{w}_{j+\frac{1}{2}}^{\mathbf{v}} \end{bmatrix}. \tag{A.202a}$$

$1 \le \sigma \le s$ and σ even:

$$\frac{\partial \mathbf{t}}{\partial \widehat{\mathbf{y}}_{j,\sigma}}^{\top} \overline{\mathbf{w}} = \begin{bmatrix} \mathbf{W}_{j,\sigma}^{\mathbf{u}} - \tau \sum\limits_{\substack{i=\sigma+1 \\ i \text{ odd}}}^{s} a_{i\sigma} \mathbf{f}_{\mathbf{v}}^{\mathbf{u}} \left(\underline{\mathbf{v}}_{\mathbf{U},j-\frac{1}{2},\sigma}^{\mathrm{o}}\right)^{\top} \mathbf{W}_{j,i}^{\mathbf{u}} \\ \mathbf{W}_{j+\frac{1}{2},\sigma}^{\mathbf{v}} - \tau \sum\limits_{\substack{i=\sigma+1 \\ i \text{ odd}}}^{s} a_{i\sigma} \mathbf{f}_{\mathbf{u}}^{\mathbf{v}} \left(\underline{\mathbf{u}}_{\mathbf{V},j,\sigma}^{\mathrm{o}}\right)^{\top} \mathbf{W}_{j+\frac{1}{2},i}^{\mathbf{v}} \end{bmatrix}. \tag{A.202b}$$

$\sigma = s+1$:

$$\frac{\partial \mathbf{t}}{\partial \widehat{\mathbf{y}}_{j,s+1}}^{\top} \overline{\mathbf{w}} = \begin{bmatrix} \mathbf{w}_j^{\mathbf{u}} - \mathbf{w}_{j+1}^{\mathbf{u}} - \sum\limits_{\substack{\sigma=1 \\ \sigma \text{ odd}}}^{s} \mathbf{f}_{\mathbf{u}}^{\mathbf{v}} \left(\underline{\mathbf{u}}_{\mathbf{V},j,\sigma}^{\mathrm{o}}\right)^{\top} \mathbf{W}_{j+\frac{1}{2},\sigma}^{\mathbf{v}} + \\ - \sum\limits_{\substack{\sigma=1 \\ \sigma \text{ even}}}^{s} \mathbf{f}_{\mathbf{u}}^{\mathbf{v}} \left(\underline{\mathbf{u}}_{\mathbf{U},j,\sigma}^{\mathrm{o}}\right)^{\top} \mathbf{W}_{j+1,\sigma}^{\mathbf{u}} \\ \mathbf{w}_{j+\frac{1}{2}}^{\mathbf{v}} - \mathbf{w}_{j+\frac{3}{2}}^{\mathbf{v}} - \sum\limits_{\substack{\sigma=1 \\ \sigma \text{ odd}}}^{s} \mathbf{f}_{\mathbf{v}}^{\mathbf{u}} \left(\underline{\mathbf{v}}_{\mathbf{U},j+\frac{1}{2},\sigma}^{\mathrm{o}}\right)^{\top} \mathbf{W}_{j+1,\sigma}^{\mathbf{u}} + \\ - \sum\limits_{\substack{\sigma=1 \\ \sigma \text{ even}}}^{s} \mathbf{f}_{\mathbf{v}}^{\mathbf{u}} \left(\underline{\mathbf{v}}_{\mathbf{V},j+\frac{1}{2},\sigma}^{\mathrm{o}}\right)^{\top} \mathbf{W}_{j+\frac{3}{2},\sigma}^{\mathbf{v}} \end{bmatrix}. \tag{A.202c}$$

☐ *For number of stages s even*

$1 \leq \sigma \leq s$ and σ odd:

$$\frac{\partial \mathbf{t}}{\partial \widehat{\mathbf{y}}_{j,\sigma}}^{\top} \overline{\mathbf{w}} = \begin{bmatrix} \mathbf{W}_{j,\sigma}^{\mathbf{u}} - \tau \sum\limits_{\substack{i=\sigma+1 \\ i \text{ even}}}^{s} a_{i\sigma} \mathbf{f}_{\mathbf{v}}^{\mathbf{u}} \left(\underline{\mathbf{v}}_{\mathbf{U},j-\frac{1}{2},\sigma}^{\mathrm{e}}\right)^{\top} \mathbf{W}_{j,i}^{\mathbf{u}} \\ \mathbf{W}_{j+\frac{1}{2},\sigma}^{\mathbf{v}} - \tau \sum\limits_{\substack{i=\sigma+1 \\ i \text{ even}}}^{s} a_{i\sigma} \mathbf{f}_{\mathbf{u}}^{\mathbf{v}} \left(\underline{\mathbf{u}}_{\mathbf{V},j,\sigma}^{\mathrm{e}}\right)^{\top} \mathbf{W}_{j+\frac{1}{2},i}^{\mathbf{v}} \end{bmatrix}. \tag{A.202d}$$

$1 \leq \sigma \leq s$ and σ even:

$$\frac{\partial \mathbf{t}}{\partial \widehat{\mathbf{y}}_{j,\sigma}}^{\top} \overline{\mathbf{w}} = \begin{bmatrix} \mathbf{W}_{j,\sigma}^{\mathbf{u}} - \tau \sum\limits_{\substack{i=\sigma+1 \\ i \text{ odd}}}^{s} a_{i\sigma} \mathbf{f}_{\mathbf{u}}^{\mathbf{v}} \left(\underline{\mathbf{u}}_{\mathbf{U},j-1,i}^{\mathrm{e}}\right)^{\top} \mathbf{W}_{j,i}^{\mathbf{u}} - \tau\, b_{\sigma} \mathbf{w}_j^{\mathbf{u}} \\ \mathbf{W}_{j+\frac{1}{2},\sigma}^{\mathbf{v}} - \tau \sum\limits_{\substack{i=\sigma+1 \\ i \text{ odd}}}^{s} a_{i\sigma} \mathbf{f}_{\mathbf{v}}^{\mathbf{u}} \left(\underline{\mathbf{v}}_{\mathbf{V},j-\frac{1}{2},\sigma}^{\mathrm{e}}\right)^{\top} \mathbf{W}_{j+\frac{1}{2},i}^{\mathbf{v}} - \tau\, b_{\sigma} \mathbf{w}_{j+\frac{1}{2}}^{\mathbf{v}} \end{bmatrix}. \tag{A.202e}$$

$\sigma = s+1$:

$$\frac{\partial \mathbf{t}}{\partial \widehat{\mathbf{y}}_{j,s+1}}^{\top} \overline{\mathbf{w}} = \begin{bmatrix} \mathbf{w}_j^{\mathbf{u}} - \mathbf{w}_{j+1}^{\mathbf{u}} - \sum\limits_{\substack{\sigma=1 \\ \sigma \text{ odd}}}^{s} \mathbf{f}_{\mathbf{u}}^{\mathbf{v}} \left(\underline{\mathbf{u}}_{\mathbf{U},j,\sigma}^{\mathrm{e}}\right)^{\top} \mathbf{W}_{j+1,\sigma}^{\mathbf{u}} + \\ - \sum\limits_{\substack{\sigma=1 \\ \sigma \text{ even}}}^{s} \mathbf{f}_{\mathbf{u}}^{\mathbf{v}} \left(\underline{\mathbf{u}}_{\mathbf{V},j,\sigma}^{\mathrm{e}}\right)^{\top} \mathbf{W}_{j+\frac{1}{2},\sigma}^{\mathbf{v}} \\ \mathbf{w}_{j+\frac{1}{2}}^{\mathbf{v}} - \mathbf{w}_{j+\frac{3}{2}}^{\mathbf{v}} - \sum\limits_{\substack{\sigma=1 \\ \sigma \text{ odd}}}^{s} \mathbf{f}_{\mathbf{v}}^{\mathbf{u}} \left(\underline{\mathbf{v}}_{\mathbf{V},j+\frac{1}{2},\sigma}^{\mathrm{e}}\right)^{\top} \mathbf{W}_{j+\frac{3}{2},\sigma}^{\mathbf{v}} + \\ - \sum\limits_{\substack{\sigma=1 \\ \sigma \text{ even}}}^{s} \mathbf{f}_{\mathbf{v}}^{\mathbf{u}} \left(\underline{\mathbf{v}}_{\mathbf{U},j+\frac{1}{2},\sigma}^{\mathrm{e}}\right)^{\top} \mathbf{W}_{j+1,\sigma}^{\mathbf{u}} \end{bmatrix}. \tag{A.202f}$$

- $\dfrac{\partial \mathbf{t}}{\partial \mathbf{m}}$

  Taking the derivative of $\mathbf{T}$ with respect to $\mathbf{m}$,

$$\frac{\partial \mathbf{t}}{\partial \mathbf{m}} = \frac{\partial \mathbf{t}}{\partial \mathbf{m}} = \frac{\partial}{\partial \mathbf{m}}\left(\mathbf{T}\overline{\mathbf{y}} - \overline{\mathbf{f}}\right) = -\frac{\partial \overline{\mathbf{f}}}{\partial \mathbf{m}} \tag{A.203}$$

  with

$$\frac{\partial \mathbf{F}^{\mathrm{o/e}}}{\partial \mathbf{m}} = \begin{bmatrix} \dfrac{\partial \mathbf{F}_1^{\mathrm{o/e}}}{\partial \mathbf{m}} \\ \mathbf{0}_{N\times N_{\mathbf{m}}} \\ \vdots \\ \dfrac{\partial \mathbf{F}_K^{\mathrm{o/e}}}{\partial \mathbf{m}} \\ \mathbf{0}_{N\times N_{\mathbf{m}}} \end{bmatrix}, \qquad \text{where} \quad \frac{\partial \mathbf{F}_k^{\mathrm{o/e}}}{\partial \mathbf{m}} = \begin{bmatrix} \dfrac{\partial \mathbf{F}_{k,1}^{\mathrm{o/e}}}{\partial \mathbf{m}} \\ \vdots \\ \dfrac{\partial \mathbf{F}_{k,s}^{\mathrm{o/e}}}{\partial \mathbf{m}} \end{bmatrix}, \tag{A.204}$$

  and

	$\dfrac{\partial \mathbf{F}_{k,\sigma}^{\mathrm{o}}}{\partial \mathbf{m}}$	$\dfrac{\partial \mathbf{F}_{k,\sigma}^{\mathrm{e}}}{\partial \mathbf{m}}$
σ odd	$\begin{bmatrix} \mathbf{f}_{\mathbf{m}}^{\mathbf{u}}\left(\underline{\mathbf{v}}_{\mathbf{U},k-\frac{1}{2},\sigma}^{\mathrm{o}}\right) \\ \mathbf{f}_{\mathbf{m}}^{\mathbf{v}}\left(\underline{\mathbf{u}}_{\mathbf{V},k,\sigma}^{\mathrm{o}}\right) \end{bmatrix}$	$\begin{bmatrix} \mathbf{f}_{\mathbf{m}}^{\mathbf{v}}\left(\underline{\mathbf{u}}_{\mathbf{U},k-1,\sigma}^{\mathrm{e}}\right) \\ \mathbf{f}_{\mathbf{m}}^{\mathbf{u}}\left(\underline{\mathbf{v}}_{\mathbf{V},k-\frac{1}{2},\sigma}^{\mathrm{e}}\right) \end{bmatrix}$
σ even	$\begin{bmatrix} \mathbf{f}_{\mathbf{m}}^{\mathbf{v}}\left(\underline{\mathbf{u}}_{\mathbf{U},k-1,\sigma}^{\mathrm{o}}\right) \\ \mathbf{f}_{\mathbf{m}}^{\mathbf{u}}\left(\underline{\mathbf{v}}_{\mathbf{V},k-\frac{1}{2},\sigma}^{\mathrm{o}}\right) \end{bmatrix}$	$\begin{bmatrix} \mathbf{f}_{\mathbf{m}}^{\mathbf{u}}\left(\underline{\mathbf{v}}_{\mathbf{U},k-\frac{1}{2},\sigma}^{\mathrm{e}}\right) \\ \mathbf{f}_{\mathbf{m}}^{\mathbf{v}}\left(\underline{\mathbf{u}}_{\mathbf{V},k,\sigma}^{\mathrm{e}}\right) \end{bmatrix}$

  - The product $\dfrac{\partial \mathbf{t}_k}{\partial \mathbf{m}}\mathbf{w}_{\mathbf{m}}$ is simply

$$\frac{\partial \mathbf{t}_k}{\partial \mathbf{m}}\mathbf{w}_{\mathbf{m}} = -\begin{bmatrix} \dfrac{\partial \mathbf{F}_k^{\mathrm{o/e}}}{\partial \mathbf{m}}\mathbf{w}_{\mathbf{m}} \\ \mathbf{0}_{N\times 1} \end{bmatrix} = \begin{bmatrix} \dfrac{\partial \mathbf{F}_{k,1}^{\mathrm{o/e}}}{\partial \mathbf{m}}\mathbf{w}_{\mathbf{m}} \\ \vdots \\ \dfrac{\partial \mathbf{F}_{k,s}^{\mathrm{o/e}}}{\partial \mathbf{m}}\mathbf{w}_{\mathbf{m}} \\ \mathbf{0}_{N\times 1} \end{bmatrix}, \tag{A.205}$$

with

	$\dfrac{\partial \mathbf{F}^{\mathrm{o}}_{k,\sigma}}{\partial \mathbf{m}}\mathbf{w_m}$	$\dfrac{\partial \mathbf{F}^{\mathrm{e}}_{k,\sigma}}{\partial \mathbf{m}}\mathbf{w_m}$
σ odd	$\begin{bmatrix} \mathbf{f^u_m}\left(\underline{\mathbf{v}}^{\mathrm{o}}_{\mathbf{U},k-\frac{1}{2},\sigma}\right)\mathbf{w_m} \\ \mathbf{f^v_m}\left(\underline{\mathbf{u}}^{\mathrm{o}}_{\mathbf{V},k,\sigma}\right)\mathbf{w_m} \end{bmatrix}$	$\begin{bmatrix} \mathbf{f^v_m}\left(\underline{\mathbf{u}}^{\mathrm{e}}_{\mathbf{U},k-1,\sigma}\right)\mathbf{w_m} \\ \mathbf{f^u_m}\left(\underline{\mathbf{v}}^{\mathrm{e}}_{\mathbf{V},k-\frac{1}{2},\sigma}\right)\mathbf{w_m} \end{bmatrix}$
σ even	$\begin{bmatrix} \mathbf{f^v_m}\left(\underline{\mathbf{u}}^{\mathrm{o}}_{\mathbf{U},k-1,\sigma}\right)\mathbf{w_m} \\ \mathbf{f^u_m}\left(\underline{\mathbf{v}}^{\mathrm{o}}_{\mathbf{V},k-\frac{1}{2},\sigma}\right)\mathbf{w_m} \end{bmatrix}$	$\begin{bmatrix} \mathbf{f^u_m}\left(\underline{\mathbf{v}}^{\mathrm{e}}_{\mathbf{U},k-\frac{1}{2},\sigma}\right)\mathbf{w_m} \\ \mathbf{f^v_m}\left(\underline{\mathbf{u}}^{\mathrm{e}}_{\mathbf{V},k,\sigma}\right)\mathbf{w_m} \end{bmatrix}$

- The product $\dfrac{\partial \mathbf{t}_k}{\partial \mathbf{m}}^{\top}\widehat{\mathbf{w}}_k$ is

$$\frac{\partial \mathbf{t}_k}{\partial \mathbf{m}}^{\top}\widehat{\mathbf{w}}_k = -\sum_{\sigma=1}^{s}\frac{\partial \mathbf{F}_{k,\sigma}}{\partial \mathbf{m}}^{\top}\mathbf{W}_{k,\sigma}$$

$$= \begin{cases} -\displaystyle\sum_{\substack{\sigma=1\\ \sigma \text{ odd}}}^{s}\left(\mathbf{f^u_m}\left(\underline{\mathbf{v}}^{\mathrm{o}}_{\mathbf{U},k-\frac{1}{2},\sigma}\right)^{\top}\mathbf{W}^{\mathbf{u}}_{k,\sigma} + \mathbf{f^v_m}\left(\underline{\mathbf{u}}^{\mathrm{o}}_{\mathbf{V},k,\sigma}\right)^{\top}\mathbf{W}^{\mathbf{v}}_{k+\frac{1}{2},\sigma}\right) + \\ -\displaystyle\sum_{\substack{\sigma=1\\ \sigma \text{ even}}}^{s}\left(\mathbf{f^v_m}\left(\underline{\mathbf{u}}^{\mathrm{o}}_{\mathbf{U},k-1,\sigma}\right)^{\top}\mathbf{W}^{\mathbf{u}}_{k,\sigma} + \mathbf{f^u_m}\left(\underline{\mathbf{v}}^{\mathrm{o}}_{\mathbf{V},k-\frac{1}{2},\sigma}\right)^{\top}\mathbf{W}^{\mathbf{v}}_{k+\frac{1}{2},\sigma}\right) \\ \qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad \text{if } s \text{ odd} \\ -\displaystyle\sum_{\substack{\sigma=1\\ \sigma \text{ odd}}}^{s}\left(\mathbf{f^v_m}\left(\underline{\mathbf{u}}^{\mathrm{e}}_{\mathbf{U},k-1,\sigma}\right)^{\top}\mathbf{W}^{\mathbf{u}}_{k,\sigma} + \mathbf{f^u_m}\left(\underline{\mathbf{v}}^{\mathrm{e}}_{\mathbf{V},k-\frac{1}{2},\sigma}\right)^{\top}\mathbf{W}^{\mathbf{v}}_{k+\frac{1}{2},\sigma}\right) + \\ \quad -\displaystyle\sum_{\substack{\sigma=1\\ \sigma \text{ even}}}^{s}\left(\mathbf{f^u_m}\left(\underline{\mathbf{v}}^{\mathrm{e}}_{\mathbf{U},k-\frac{1}{2},\sigma}\right)^{\top}\mathbf{W}^{\mathbf{u}}_{k,\sigma} + \mathbf{f^v_m}\left(\underline{\mathbf{u}}^{\mathrm{e}}_{\mathbf{V},k,\sigma}\right)^{\top}\mathbf{W}^{\mathbf{v}}_{k+\frac{1}{2},\sigma}\right) \\ \qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad \text{if } s \text{ even} \end{cases} \tag{A.206}$$

- $\dfrac{\partial \mathbf{t}}{\partial \mathbf{y}_0}$

Taking the derivative of $\mathbf{t}$ with respect to $\mathbf{y}_0$,

$$\frac{\partial \mathbf{t}}{\partial \mathbf{y}_0} = \frac{\partial \mathbf{t}}{\partial \mathbf{y}_0} = \frac{\partial}{\partial \mathbf{y}_0}\left(\mathbf{T}\overline{\mathbf{y}} - \overline{\mathbf{f}}\right) = -\frac{\partial \overline{\mathbf{f}}}{\partial \mathbf{y}_0} \tag{A.207}$$

with

$$\frac{\partial \overline{\mathbf{f}}}{\partial \mathbf{y}_0} = \begin{bmatrix} \frac{\partial \mathbf{F}_1^{\mathrm{o/e}}}{\partial \mathbf{y}_0}^\top & \mathbf{I}_N & \frac{\partial \mathbf{F}_2^{\mathrm{o/e}}}{\partial \mathbf{y}_0}^\top & \mathbf{0}_{N\times N} & \cdots & \frac{\partial \mathbf{F}_K^{\mathrm{o/e}}}{\partial \mathbf{y}_0}^\top & \mathbf{0}_{N\times N} \end{bmatrix}^\top ,$$

where

$$\frac{\partial \mathbf{F}_k^{\mathrm{o/e}}}{\partial \mathbf{y}_0} = \begin{bmatrix} \frac{\partial \mathbf{F}_{k,1}^{\mathrm{o/e}}}{\partial \mathbf{y}_0}^\top & \cdots & \frac{\partial \mathbf{F}_{k,s}^{\mathrm{o/e}}}{\partial \mathbf{y}_0}^\top \end{bmatrix}^\top .$$

Clearly $\frac{\partial \mathbf{F}_k^{\mathrm{o/e}}}{\partial \mathbf{y}_0} = \mathbf{0}_{N\times N}$ for $k > 1$. For $k = 1$ we have

- *For number of stages s odd*:

$$\frac{\partial \mathbf{F}_{1,\sigma}^{\mathrm{o}}}{\partial \mathbf{y}_0} = \begin{cases} \begin{bmatrix} \mathbf{0}_{N_\mathbf{u}\times N_\mathbf{u}} & \mathbf{f}_\mathbf{v}^\mathbf{u}\left(\underline{\mathbf{v}}_{\mathbf{U},1/2,\sigma}^{\mathrm{o}}\right) \\ \mathbf{0}_{N_\mathbf{v}\times N_\mathbf{u}} & \mathbf{0}_{N_\mathbf{v}\times N_\mathbf{v}} \end{bmatrix} & \text{if } \sigma \text{ odd} \\ \begin{bmatrix} \mathbf{f}_\mathbf{u}^\mathbf{v}\left(\underline{\mathbf{u}}_{\mathbf{U},0,\sigma}^{\mathrm{o}}\right) & \mathbf{0}_{N_\mathbf{v}\times N_\mathbf{v}} \\ \mathbf{0}_{N_\mathbf{u}\times N_\mathbf{u}} & \mathbf{f}_\mathbf{v}^\mathbf{u}\left(\underline{\mathbf{v}}_{\mathbf{V},1/2,\sigma}^{\mathrm{o}}\right) \end{bmatrix} & \text{if } \sigma \text{ even.} \end{cases} \tag{A.208}$$

- *For number of stages s even*:

$$\frac{\partial \mathbf{F}_{1,\sigma}^{\mathrm{e}}}{\partial \mathbf{y}_0} = \begin{cases} \begin{bmatrix} \mathbf{f}_\mathbf{u}^\mathbf{v}\left(\underline{\mathbf{u}}_{\mathbf{U},0,\sigma}^{\mathrm{e}}\right) & \mathbf{0}_{N_\mathbf{v}\times N_\mathbf{v}} \\ \mathbf{0}_{N_\mathbf{u}\times N_\mathbf{u}} & \mathbf{f}_\mathbf{v}^\mathbf{u}\left(\underline{\mathbf{v}}_{\mathbf{V},1/2,\sigma}^{\mathrm{e}}\right) \end{bmatrix} & \text{if } \sigma \text{ odd} \\ \begin{bmatrix} \mathbf{0}_{N_\mathbf{u}\times N_\mathbf{u}} & \mathbf{f}_\mathbf{v}^\mathbf{u}\left(\underline{\mathbf{v}}_{\mathbf{U},1/2,\sigma}^{\mathrm{e}}\right) \\ \mathbf{0}_{N_\mathbf{v}\times N_\mathbf{u}} & \mathbf{0}_{N_\mathbf{v}\times N_\mathbf{v}} \end{bmatrix} & \text{if } \sigma \text{ even.} \end{cases} \tag{A.209}$$

Then the products of $\frac{\partial \mathbf{t}_k}{\partial \mathbf{y}_0}$ are

- The product $\frac{\partial \mathbf{t}_k}{\partial \mathbf{y}_0}\mathbf{w}_0$ is

$$\frac{\partial \mathbf{t}_k}{\partial \mathbf{y}_0}\mathbf{w}_0 = \begin{cases} -\begin{bmatrix} \dfrac{\partial \mathbf{F}^{\mathrm{o/e}}_{1,1}}{\partial \mathbf{y}_0}\mathbf{w}_0 \\ \vdots \\ \dfrac{\partial \mathbf{F}^{\mathrm{o/e}}_{1,s}}{\partial \mathbf{y}_0}\mathbf{w}_0 \\ \mathbf{w}^{\mathbf{u}}_0 \\ \mathbf{w}^{\mathbf{v}}_0 \end{bmatrix} & \text{if } k = 1 \\ \mathbf{0}_{(s+1)N\times 1} & \text{if } k > 1 \end{cases} \tag{A.210}$$

with

	$\dfrac{\partial \mathbf{F}^{\mathrm{o}}_{1,\sigma}}{\partial \mathbf{y}_0}\mathbf{w}_0$	$\dfrac{\partial \mathbf{F}^{\mathrm{e}}_{1,\sigma}}{\partial \mathbf{y}_0}\mathbf{w}_0$
σ odd	$\begin{bmatrix} \mathbf{f}^{\mathbf{u}}_{\mathbf{v}}\left(\underline{\mathbf{v}}^{\mathrm{o}}_{\mathbf{U},1/2,\sigma}\right)\mathbf{w}^{\mathbf{v}}_0 \\ \mathbf{0}_{N_{\mathbf{v}}\times 1} \end{bmatrix}$	$\begin{bmatrix} \mathbf{f}^{\mathbf{v}}_{\mathbf{u}}\left(\underline{\mathbf{u}}^{\mathrm{e}}_{\mathbf{U},0,\sigma}\right)\mathbf{w}^{\mathbf{u}}_0 \\ \mathbf{f}^{\mathbf{u}}_{\mathbf{v}}\left(\underline{\mathbf{v}}^{\mathrm{e}}_{\mathbf{V},1/2,\sigma}\right)\mathbf{w}^{\mathbf{v}}_0 \end{bmatrix}$
σ even	$\begin{bmatrix} \mathbf{f}^{\mathbf{v}}_{\mathbf{u}}\left(\underline{\mathbf{u}}^{\mathrm{o}}_{\mathbf{U},0,\sigma}\right)\mathbf{w}^{\mathbf{u}}_0 \\ \mathbf{f}^{\mathbf{u}}_{\mathbf{v}}\left(\underline{\mathbf{v}}^{\mathrm{o}}_{\mathbf{V},1/2,\sigma}\right)\mathbf{w}^{\mathbf{v}}_0 \end{bmatrix}$	$\begin{bmatrix} \mathbf{f}^{\mathbf{u}}_{\mathbf{v}}\left(\underline{\mathbf{v}}^{\mathrm{e}}_{\mathbf{U},1/2,\sigma}\right)\mathbf{w}^{\mathbf{v}}_0 \\ \mathbf{0}_{N_{\mathbf{v}}\times 1} \end{bmatrix}$

- The product $\dfrac{\partial \mathbf{t}_k}{\partial \mathbf{y}_0}^{\top}\widehat{\mathbf{w}}_k$ is $\mathbf{0}_{N\times 1}$ for $k > 1$ and for $k = 1$
  - *For number of stages s odd*:

$$\frac{\partial \mathbf{t}_1}{\partial \mathbf{y}_0}^{\top}\widehat{\mathbf{w}}_1 = \begin{bmatrix} -\mathbf{w}^{\mathbf{u}}_1 - \sum\limits_{\substack{\sigma=1 \\ \sigma \text{ even}}}^{s} \mathbf{f}^{\mathbf{v}}_{\mathbf{u}}\left(\underline{\mathbf{u}}^{\mathrm{o}}_{\mathbf{U},0,\sigma}\right)^{\top}\mathbf{W}^{\mathbf{u}}_{1,\sigma} \\ -\mathbf{w}^{\mathbf{v}}_{\frac{3}{2}} - \sum\limits_{\substack{\sigma=1 \\ \sigma \text{ odd}}}^{s} \mathbf{f}^{\mathbf{u}}_{\mathbf{v}}\left(\underline{\mathbf{v}}^{\mathrm{o}}_{\mathbf{U},\frac{1}{2},\sigma}\right)^{\top}\mathbf{W}^{\mathbf{u}}_{1,\sigma} + \\ - \sum\limits_{\substack{\sigma=1 \\ \sigma \text{ even}}}^{s} \mathbf{f}^{\mathbf{u}}_{\mathbf{v}}\left(\underline{\mathbf{v}}^{\mathrm{o}}_{\mathbf{V},\frac{1}{2},\sigma}\right)^{\top}\mathbf{W}^{\mathbf{v}}_{\frac{3}{2},\sigma} \end{bmatrix} \tag{A.211a}$$

  - *For number of stages s even*:

$$\frac{\partial \mathbf{t}_1}{\partial \mathbf{y}_0}^\top \widehat{\mathbf{w}}_1 = \begin{bmatrix} -\mathbf{w}_1^{\mathbf{u}} - \sum\limits_{\substack{\sigma=1 \\ \sigma \text{ odd}}}^{s} \mathbf{f}_{\mathbf{u}}^{\mathbf{v}} \left(\underline{\mathbf{u}}^{\mathrm{e}}_{\mathbf{U},0,\sigma}\right)^\top \mathbf{W}^{\mathbf{u}}_{1,\sigma} \\ -\mathbf{w}^{\mathbf{v}}_{\frac{3}{2}} - \sum\limits_{\substack{\sigma=1 \\ \sigma \text{ odd}}}^{s} \mathbf{f}_{\mathbf{v}}^{\mathbf{u}} \left(\underline{\mathbf{v}}^{\mathrm{e}}_{\mathbf{V},\frac{1}{2},\sigma}\right)^\top \mathbf{W}^{\mathbf{v}}_{\frac{3}{2},\sigma} + \\ - \sum\limits_{\substack{\sigma=1 \\ \sigma \text{ even}}}^{s} \mathbf{f}_{\mathbf{v}}^{\mathbf{u}} \left(\underline{\mathbf{v}}^{\mathrm{e}}_{\mathbf{U},\frac{1}{2},\sigma}\right)^\top \mathbf{W}^{\mathbf{u}}_{1,\sigma} \end{bmatrix}. \tag{A.211b}$$

Derivatives of $\dfrac{\partial \mathbf{t}}{\partial \overline{\mathbf{y}}}$

We now find the derivatives of the σth $N \times 1$ block $\dfrac{\partial \mathbf{t}_{k,\sigma}}{\partial \overline{\mathbf{y}}} \overline{\mathbf{w}}$ of the vector $\dfrac{\partial \mathbf{t}_k}{\partial \overline{\mathbf{y}}} \overline{\mathbf{w}}$, which in turn is the kth block of $\mathbf{t}_{\mathbf{y}} \overline{\mathbf{w}} = \dfrac{\partial \mathbf{t}}{\partial \overline{\mathbf{y}}} \overline{\mathbf{w}}$.

The expression for the product $\dfrac{\partial \mathbf{t}_{k,\sigma}}{\partial \overline{\mathbf{y}}} \overline{\mathbf{w}}$ was given in (A.198). We now find expressions for the derivatives of (A.198) with respect to $\overline{\mathbf{y}}$, $\mathbf{m}$ and $\mathbf{y}_0$.

- $\dfrac{\partial^2 \mathbf{t}}{\partial \mathbf{m} \partial \overline{\mathbf{y}}} \overline{\mathbf{w}}$

  We differentiate (A.198) with respect to $\mathbf{m}$:

  - *For number of stages s odd*

    $1 \leq \sigma \leq s$ and σ odd:

$$\frac{\partial^2 \mathbf{t}_{k,\sigma}}{\partial \mathbf{m} \partial \overline{\mathbf{y}}} \overline{\mathbf{w}} = \begin{bmatrix} -\left(\mathbf{f}_{\mathbf{v}}^{\mathbf{u}} \left(\underline{\mathbf{v}}^{\mathrm{o}}_{\mathbf{U},k-\frac{1}{2},\sigma}\right) \underline{\mathbf{w}}^{\mathbf{v},\mathrm{o}}_{\mathbf{W}^{\mathbf{u}},k-\frac{1}{2},\sigma}\right)_{\mathbf{m}} \\ -\left(\mathbf{f}_{\mathbf{u}}^{\mathbf{v}} \left(\underline{\mathbf{u}}^{\mathrm{o}}_{\mathbf{V},k,\sigma}\right) \underline{\mathbf{w}}^{\mathbf{u},\mathrm{o}}_{\mathbf{W}^{\mathbf{v}},k,\sigma}\right)_{\mathbf{m}} \end{bmatrix}. \tag{A.212a}$$

    $1 \leq \sigma \leq s$ and σ even:

$$\frac{\partial^2 \mathbf{t}_{k,\sigma}}{\partial \mathbf{m} \partial \overline{\mathbf{y}}} \overline{\mathbf{w}} = \begin{bmatrix} -\left(\mathbf{f}_{\mathbf{u}}^{\mathbf{v}} \left(\underline{\mathbf{u}}^{\mathrm{o}}_{\mathbf{U},k-1,\sigma}\right) \underline{\mathbf{w}}^{\mathbf{u},\mathrm{o/e}}_{\mathbf{W}^{\mathbf{u}},k-1,\sigma}\right)_{\mathbf{m}} \\ -\left(\mathbf{f}_{\mathbf{v}}^{\mathbf{u}} \left(\underline{\mathbf{v}}^{\mathrm{o}}_{\mathbf{V},k-\frac{1}{2},\sigma}\right) \underline{\mathbf{w}}^{\mathbf{v},\mathrm{o}}_{\mathbf{W}^{\mathbf{v}},k-\frac{1}{2},\sigma}\right)_{\mathbf{m}} \end{bmatrix}. \tag{A.212b}$$

    $\sigma = s + 1$:

$$\frac{\partial^2 \mathbf{t}_{k,s+1}}{\partial \mathbf{m} \partial \overline{\mathbf{y}}} \overline{\mathbf{w}} = \mathbf{0}_{N \times N_{\mathbf{m}}}. \tag{A.212c}$$

- *For number of stages s even*

  $1 \leq \sigma \leq s$ and σ odd:

$$\frac{\partial^2 \mathbf{t}_{k,\sigma}}{\partial \mathbf{m} \partial \overline{\mathbf{y}}} \overline{\mathbf{w}} = \begin{bmatrix} -\left(\mathbf{f}_{\mathbf{u}}^{\mathbf{v}} \left(\underline{\mathbf{u}}_{\mathbf{U},k-1,\sigma}^{\mathrm{e}}\right) \underline{\mathbf{w}}_{\mathbf{W}^{\mathbf{u}},k-1,\sigma}^{\mathbf{u},\mathrm{e}}\right)_{\mathbf{m}} \\ -\left(\mathbf{f}_{\mathbf{v}}^{\mathbf{u}} \left(\underline{\mathbf{v}}_{\mathbf{V},k-\frac{1}{2},\sigma}^{\mathrm{e}}\right) \underline{\mathbf{w}}_{\mathbf{W}^{\mathbf{v}},k-\frac{1}{2},\sigma}^{\mathbf{v},\mathrm{e}}\right)_{\mathbf{m}} \end{bmatrix}. \tag{A.212d}$$

  $1 \leq \sigma \leq s$ and σ even:

$$\frac{\partial^2 \mathbf{t}_{k,\sigma}}{\partial \mathbf{m} \partial \overline{\mathbf{y}}} \overline{\mathbf{w}} = \begin{bmatrix} -\left(\mathbf{f}_{\mathbf{v}}^{\mathbf{u}} \left(\underline{\mathbf{v}}_{\mathbf{U},k-\frac{1}{2},\sigma}^{\mathrm{e}}\right) \underline{\mathbf{w}}_{\mathbf{W}^{\mathbf{u}},k-\frac{1}{2},\sigma}^{\mathbf{v},\mathrm{e}}\right)_{\mathbf{m}} \\ -\left(\mathbf{f}_{\mathbf{u}}^{\mathbf{v}} \left(\underline{\mathbf{u}}_{\mathbf{V},k,\sigma}^{\mathrm{e}}\right) \underline{\mathbf{w}}_{\mathbf{W}^{\mathbf{v}},k,\sigma}^{\mathbf{u},\mathrm{e}}\right)_{\mathbf{m}} \end{bmatrix}. \tag{A.212e}$$

  $\sigma = s + 1$:

$$\frac{\partial^2 \mathbf{t}_{k,s+1}}{\partial \mathbf{m} \partial \overline{\mathbf{y}}} \overline{\mathbf{w}} = \mathbf{0}_{N \times N_{\mathbf{m}}}. \tag{A.212f}$$

- $\dfrac{\partial^2 \mathbf{t}}{\partial \mathbf{y}_0 \partial \overline{\mathbf{y}}} \overline{\mathbf{w}}$

  Taking the derivative of (A.198) with respect to $\mathbf{y}_0$ gives

- *For number of stages s odd*

  $1 \leq \sigma \leq s$ and σ odd:

$$\frac{\partial^2 \mathbf{t}_{1,\sigma}}{\partial \mathbf{y}_0 \partial \overline{\mathbf{y}}} \overline{\mathbf{w}} = \begin{bmatrix} \mathbf{0}_{N_{\mathbf{u}} \times N_{\mathbf{u}}} & -\left(\mathbf{f}_{\mathbf{v}}^{\mathbf{u}} \left(\underline{\mathbf{v}}_{\mathbf{U},\frac{1}{2},\sigma}^{\mathrm{o}}\right) \underline{\mathbf{w}}_{\mathbf{W}^{\mathbf{u}},\frac{1}{2},\sigma}^{\mathbf{v},\mathrm{o}}\right)_{\mathbf{v}} \\ \mathbf{0}_{N_{\mathbf{v}} \times N_{\mathbf{u}}} & \mathbf{0}_{N_{\mathbf{v}} \times N_{\mathbf{v}}} \end{bmatrix}. \tag{A.213a}$$

  $1 \leq \sigma \leq s$ and σ even:

$$\frac{\partial^2 \mathbf{t}_{1,\sigma}}{\partial \mathbf{y}_0 \partial \overline{\mathbf{y}}} \overline{\mathbf{w}} = \begin{bmatrix} -\left(\mathbf{f}_{\mathbf{u}}^{\mathbf{v}} \left(\underline{\mathbf{u}}_{\mathbf{U},0,\sigma}^{\mathrm{o}}\right) \underline{\mathbf{w}}_{\mathbf{W}^{\mathbf{u}},0,\sigma}^{\mathbf{u},\mathrm{o}}\right)_{\mathbf{u}} & \mathbf{0}_{N_{\mathbf{v}} \times N_{\mathbf{v}}} \\ \mathbf{0}_{N_{\mathbf{u}} \times N_{\mathbf{u}}} & -\left(\mathbf{f}_{\mathbf{v}}^{\mathbf{u}} \left(\underline{\mathbf{v}}_{\mathbf{V},\frac{1}{2},\sigma}^{\mathrm{o}}\right) \underline{\mathbf{w}}_{\mathbf{W}^{\mathbf{v}},\frac{1}{2},\sigma}^{\mathbf{v},\mathrm{o}}\right)_{\mathbf{v}} \end{bmatrix}. \tag{A.213b}$$

  $\sigma = s + 1$:

$$\frac{\partial^2 \mathbf{t}_{1,s+1}}{\partial \mathbf{y}_0 \partial \overline{\mathbf{y}}} \overline{\mathbf{w}} = \mathbf{0}_{N \times N}. \tag{A.213c}$$

- *For number of stages s even*

  $1 \leq \sigma \leq s$ and σ odd:

$$\frac{\partial^2 \mathbf{t}_{1,\sigma}}{\partial \mathbf{y}_0 \partial \overline{\mathbf{y}}} \overline{\mathbf{w}} = \begin{bmatrix} -\left(\mathbf{f}_{\mathbf{u}}^{\mathbf{v}}\left(\underline{\mathbf{u}}_{\mathbf{U},0,\sigma}^{\mathrm{e}}\right) \underline{\mathbf{w}}_{\mathbf{W}^{\mathbf{u}},0,\sigma}^{\mathbf{u},\mathrm{e}}\right)_{\mathbf{u}} & \mathbf{0}_{N_{\mathbf{v}} \times N_{\mathbf{v}}} \\ \mathbf{0}_{N_{\mathbf{u}} \times N_{\mathbf{u}}} & -\left(\mathbf{f}_{\mathbf{v}}^{\mathbf{u}}\left(\underline{\mathbf{v}}_{\mathbf{V},\frac{1}{2},\sigma}^{\mathrm{e}}\right) \underline{\mathbf{w}}_{\mathbf{W}^{\mathbf{v}},\frac{1}{2},\sigma}^{\mathbf{v},\mathrm{e}}\right)_{\mathbf{v}} \end{bmatrix}. \tag{A.213d}$$

  $1 \leq \sigma \leq s$ and σ even:

$$\frac{\partial^2 \mathbf{t}_{1,\sigma}}{\partial \mathbf{y}_0 \partial \overline{\mathbf{y}}} \overline{\mathbf{w}} = \begin{bmatrix} \mathbf{0}_{N_{\mathbf{u}} \times N_{\mathbf{u}}} & -\left(\mathbf{f}_{\mathbf{v}}^{\mathbf{u}}\left(\underline{\mathbf{v}}_{\mathbf{U},\frac{1}{2},\sigma}^{\mathrm{e}}\right) \underline{\mathbf{w}}_{\mathbf{W}^{\mathbf{u}},\frac{1}{2},\sigma}^{\mathbf{v},\mathrm{e}}\right)_{\mathbf{v}} \\ \mathbf{0}_{N_{\mathbf{v}} \times N_{\mathbf{u}}} & \mathbf{0}_{N_{\mathbf{v}} \times N_{\mathbf{v}}} \end{bmatrix}. \tag{A.213e}$$

  $\sigma = s + 1$:

$$\frac{\partial^2 \mathbf{t}_{1,s+1}}{\partial \mathbf{y}_0 \partial \overline{\mathbf{y}}} \overline{\mathbf{w}} = \mathbf{0}_{N \times N}. \tag{A.213f}$$

  The derivatives $\dfrac{\partial^2 \mathbf{t}_{k,\sigma}}{\partial \mathbf{y}_0 \partial \overline{\mathbf{y}}} \overline{\mathbf{w}}$ are $\mathbf{0}_{N \times N}$ for all $k > 1$, $1 \leq \sigma \leq s + 1$.

- $\dfrac{\partial^2 \mathbf{t}}{\partial \mathbf{y} \partial \overline{\mathbf{y}}} \overline{\mathbf{w}}$

  Consider the ℓth term of

$$\frac{\partial^2 \mathbf{t}_{k,\sigma}}{\partial \mathbf{y} \partial \overline{\mathbf{y}}} \overline{\mathbf{w}} = \left[\frac{\partial^2 \mathbf{t}_{k,\sigma}}{\partial \widehat{\mathbf{y}}_1 \partial \overline{\mathbf{y}}} \overline{\mathbf{w}} \quad \frac{\partial^2 \mathbf{t}_{k,\sigma}}{\partial \widehat{\mathbf{y}}_2 \partial \overline{\mathbf{y}}} \overline{\mathbf{w}} \quad \cdots \quad \frac{\partial^2 \mathbf{t}_{k,\sigma}}{\partial \widehat{\mathbf{y}}_K \partial \overline{\mathbf{y}}} \overline{\mathbf{w}} \right], \tag{A.214}$$

  with

$$\frac{\partial^2 \mathbf{t}_{k,\sigma}}{\partial \widehat{\mathbf{y}}_\ell \partial \overline{\mathbf{y}}} \overline{\mathbf{w}} = \left[\frac{\partial^2 \mathbf{t}_{k,\sigma}}{\partial \mathbf{Y}_{\ell,1} \partial \overline{\mathbf{y}}} \overline{\mathbf{w}} \quad \cdots \quad \frac{\partial^2 \mathbf{t}_{k,\sigma}}{\partial \mathbf{Y}_{\ell,s} \partial \overline{\mathbf{y}}} \overline{\mathbf{w}} \quad \frac{\partial^2 \mathbf{t}_{k,\sigma}}{\partial \mathbf{y}_\ell \partial \overline{\mathbf{y}}} \overline{\mathbf{w}} \right], \tag{A.215}$$

  where

$$\begin{aligned} \frac{\partial^2 \mathbf{t}_{k,\sigma}}{\partial \mathbf{Y}_{\ell,i} \partial \overline{\mathbf{y}}} \overline{\mathbf{w}} &= \left[\frac{\partial^2 \mathbf{t}_{k,\sigma}}{\partial \mathbf{U}_{\ell,i} \partial \overline{\mathbf{y}}} \overline{\mathbf{w}} \quad \frac{\partial^2 \mathbf{t}_{k,\sigma}}{\partial \mathbf{V}_{\ell,i} \partial \overline{\mathbf{y}}} \overline{\mathbf{w}} \right] \qquad \text{for } 1 \leq i \leq s \\ \frac{\partial^2 \mathbf{t}_{k,\sigma}}{\partial \mathbf{y}_\ell \partial \overline{\mathbf{y}}} \overline{\mathbf{w}} &= \left[\frac{\partial^2 \mathbf{t}_{k,\sigma}}{\partial \mathbf{u}_\ell \partial \overline{\mathbf{y}}} \overline{\mathbf{w}} \quad \frac{\partial^2 \mathbf{t}_{k,\sigma}}{\partial \mathbf{v}_\ell \partial \overline{\mathbf{y}}} \overline{\mathbf{w}} \right]. \end{aligned}$$

  Then

- *For number of stages s odd*

  For $1 \leq \sigma \leq s$ and σ odd:

$$\frac{\partial^2 \mathbf{t}_{k,\sigma}}{\partial \mathbf{U}_{k,i} \partial \overline{\mathbf{y}}} \overline{\mathbf{w}} = \begin{cases} \tau \alpha_{\sigma i} \begin{bmatrix} -\left(\mathbf{f}_{\mathbf{v}}^{\mathbf{u}} \left(\underline{\mathbf{v}}^{\mathrm{o}}_{\mathbf{U},k-\frac{1}{2},\sigma}\right) \underline{\mathbf{w}}^{\mathbf{v},\mathrm{o}}_{\mathbf{W}^{\mathbf{u}},k-\frac{1}{2},\sigma}\right)_{\mathbf{v}} \\ \mathbf{0}_{N_{\mathbf{v}} \times N_{\mathbf{v}}} \end{bmatrix} & \text{if } 1 \leq i \leq \sigma, \\ & i \text{ even} \\ \mathbf{0}_{N \times N_{\mathbf{v}}} & \text{otherwise,} \end{cases} \tag{A.216a}$$

$$\frac{\partial^2 \mathbf{t}_{k,\sigma}}{\partial \mathbf{V}_{k,i} \partial \overline{\mathbf{y}}} \overline{\mathbf{w}} = \begin{cases} \tau \alpha_{\sigma i} \begin{bmatrix} \mathbf{0}_{N_{\mathbf{v}} \times N_{\mathbf{u}}} \\ -\left(\mathbf{f}_{\mathbf{u}}^{\mathbf{v}} \left(\underline{\mathbf{u}}^{\mathrm{o}}_{\mathbf{V},k,\sigma}\right) \underline{\mathbf{w}}^{\mathbf{u},\mathrm{o}}_{\mathbf{W}^{\mathbf{v}},k,\sigma}\right)_{\mathbf{u}} \end{bmatrix} & \text{if } 1 \leq i \leq \sigma, \\ & i \text{ even} \\ \mathbf{0}_{N \times N_{\mathbf{u}}} & \text{otherwise,} \end{cases} \tag{A.216b}$$

and $\dfrac{\partial^2 \mathbf{t}_{k,\sigma}}{\partial \mathbf{Y}_{\ell,i} \partial \overline{\mathbf{y}}} \overline{\mathbf{w}} = \mathbf{0}_{N \times N}$ for all other values of ℓ. Also,

$$\frac{\partial^2 \mathbf{t}_{k,\sigma}}{\partial \mathbf{u}_{\ell} \partial \overline{\mathbf{y}}} \overline{\mathbf{w}} = \begin{cases} \begin{bmatrix} \mathbf{0}_{N_{\mathbf{u}} \times N_{\mathbf{u}}} \\ -\left(\mathbf{f}_{\mathbf{u}}^{\mathbf{v}} \left(\underline{\mathbf{u}}^{\mathrm{o}}_{\mathbf{V},k,\sigma}\right) \underline{\mathbf{w}}^{\mathbf{u},\mathrm{o}}_{\mathbf{W}^{\mathbf{v}},k,\sigma}\right)_{\mathbf{u}} \end{bmatrix} & \text{if } \ \ell = k \\ \mathbf{0}_{N \times N_{\mathbf{u}}} & \text{otherwise} \end{cases} \tag{A.216c}$$

and

$$\frac{\partial^2 \mathbf{t}_{k,\sigma}}{\partial \mathbf{v}_{\ell} \partial \overline{\mathbf{y}}} \overline{\mathbf{w}} = \begin{cases} \begin{bmatrix} -\left(\mathbf{f}_{\mathbf{v}}^{\mathbf{u}} \left(\underline{\mathbf{v}}^{\mathrm{o}}_{\mathbf{U},k-\frac{1}{2},\sigma}\right) \underline{\mathbf{w}}^{\mathbf{v},\mathrm{o}}_{\mathbf{W}^{\mathbf{u}},k-\frac{1}{2},\sigma}\right)_{\mathbf{v}} \\ \mathbf{0}_{N_{\mathbf{v}} \times N_{\mathbf{v}}} \end{bmatrix} & \text{if } \ \ell = k-1 \\ \mathbf{0}_{N \times N_{\mathbf{v}}} & \text{otherwise.} \end{cases} \tag{A.216d}$$

For $1 \leq \sigma \leq s$ and σ even:

$$\frac{\partial^2 \mathbf{t}_{k,\sigma}}{\partial \mathbf{U}_{k,i} \partial \overline{\mathbf{y}}} \overline{\mathbf{w}} = \begin{cases} \tau \alpha_{\sigma i} \begin{bmatrix} -\left(\mathbf{f}_{\mathbf{u}}^{\mathbf{v}} \left(\underline{\mathbf{u}}_{\mathbf{U},k-1,\sigma}^{\mathrm{o}}\right) \underline{\mathbf{w}}_{\mathbf{W}^{\mathbf{u}},k-1,\sigma}^{\mathbf{u},\mathrm{o/e}}\right)_{\mathbf{u}} \\ \mathbf{0}_{N_{\mathbf{u}} \times N_{\mathbf{u}}} \end{bmatrix} & \text{if } 1 \leq i \leq \sigma, \\ & i \text{ odd} \\ \mathbf{0}_{N \times N_{\mathbf{v}}} & \text{otherwise,} \end{cases} \tag{A.216e}$$

$$\frac{\partial^2 \mathbf{t}_{k,\sigma}}{\partial \mathbf{V}_{k,i} \partial \overline{\mathbf{y}}} \overline{\mathbf{w}} = \begin{cases} \tau \alpha_{\sigma i} \begin{bmatrix} \mathbf{0}_{N_{\mathbf{u}} \times N_{\mathbf{v}}} \\ -\left(\mathbf{f}_{\mathbf{v}}^{\mathbf{u}} \left(\underline{\mathbf{v}}_{\mathbf{V},k-\frac{1}{2},\sigma}^{\mathrm{o}}\right) \underline{\mathbf{w}}_{\mathbf{W}^{\mathbf{v}},k-\frac{1}{2},\sigma}^{\mathbf{v},\mathrm{o}}\right)_{\mathbf{v}} \end{bmatrix} & \text{if } 1 \leq i \leq \sigma, \\ & i \text{ odd} \\ \mathbf{0}_{N \times N_{\mathbf{u}}} & \text{otherwise,} \end{cases} \tag{A.216f}$$

and $\dfrac{\partial^2 \mathbf{t}_{k,\sigma}}{\partial \mathbf{Y}_{\ell,i} \partial \overline{\mathbf{y}}} \overline{\mathbf{w}} = \mathbf{0}_{N \times N}$ for all other values of ℓ. Also,

$$\frac{\partial^2 \mathbf{t}_{k,\sigma}}{\partial \mathbf{u}_{\ell} \partial \overline{\mathbf{y}}} \overline{\mathbf{w}} = \begin{cases} \begin{bmatrix} -\left(\mathbf{f}_{\mathbf{u}}^{\mathbf{v}} \left(\underline{\mathbf{u}}_{\mathbf{U},k-1,\sigma}^{\mathrm{o}}\right) \underline{\mathbf{w}}_{\mathbf{W}^{\mathbf{u}},k-1,\sigma}^{\mathbf{u},\mathrm{o/e}}\right)_{\mathbf{u}} \\ \mathbf{0}_{N_{\mathbf{u}} \times N_{\mathbf{u}}} \end{bmatrix} & \text{if } \ \ell = k-1 \\ \mathbf{0}_{N \times N_{\mathbf{u}}} & \text{otherwise} \end{cases}. \tag{A.216g}$$

and

$$\frac{\partial^2 \mathbf{t}_{k,\sigma}}{\partial \mathbf{v}_{\ell} \partial \overline{\mathbf{y}}} \overline{\mathbf{w}} = \begin{cases} \begin{bmatrix} \mathbf{0}_{N_{\mathbf{v}} \times N_{\mathbf{v}}} \\ -\left(\mathbf{f}_{\mathbf{v}}^{\mathbf{u}} \left(\underline{\mathbf{v}}_{\mathbf{U},k-\frac{1}{2},\sigma}^{\mathrm{o}}\right) \underline{\mathbf{w}}_{\mathbf{W}^{\mathbf{u}},k-\frac{1}{2},\sigma}^{\mathbf{v},\mathrm{o}}\right)_{\mathbf{v}} \end{bmatrix} & \text{if } \ \ell = k-1 \\ \mathbf{0}_{N \times N_{\mathbf{v}}} & \text{otherwise.} \end{cases} \tag{A.216h}$$

For $\sigma = s+1$:

$$\frac{\partial^2 \mathbf{t}_{k,s+1}}{\partial \widehat{\mathbf{y}}_{\ell} \partial \overline{\mathbf{y}}} \overline{\mathbf{w}} = \mathbf{0}_{N \times (s+1)N}. \tag{A.216i}$$

- *For number of stages s even*

For $1 \leq \sigma \leq s$ and σ odd:

$$\frac{\partial^2 \mathbf{t}_{k,\sigma}}{\partial \mathbf{U}_{k,i} \partial \overline{\mathbf{y}}} \overline{\mathbf{w}} = \begin{cases} \tau \alpha_{\sigma i} \begin{bmatrix} -\left(\mathbf{f}_{\mathbf{u}}^{\mathbf{v}} \left(\underline{\mathbf{u}}_{\mathbf{U},k-1,\sigma}^{\mathrm{e}}\right) \underline{\mathbf{w}}_{\mathbf{W}^{\mathbf{u}},k-1,\sigma}^{\mathbf{u},\mathrm{e}}\right)_{\mathbf{u}} \\ \mathbf{0}_{N_{\mathbf{v}} \times N_{\mathbf{u}}} \end{bmatrix} & \text{if } 1 \leq i \leq \sigma, \\ & i \text{ even} \\ \mathbf{0}_{N \times N_{\mathbf{v}}} & \text{otherwise,} \end{cases} \tag{A.216j}$$

$$\frac{\partial^2 \mathbf{t}_{k,\sigma}}{\partial \mathbf{V}_{k,i} \partial \overline{\mathbf{y}}} \overline{\mathbf{w}} = \begin{cases} \tau \alpha_{\sigma i} \begin{bmatrix} \mathbf{0}_{N_{\mathbf{v}} \times N_{\mathbf{v}}} \\ -\left(\mathbf{f}_{\mathbf{v}}^{\mathbf{u}} \left(\underline{\mathbf{v}}_{\mathbf{V},k-\frac{1}{2},\sigma}^{\mathrm{e}}\right) \underline{\mathbf{w}}_{\mathbf{W}^{\mathbf{v}},k-\frac{1}{2},\sigma}^{\mathbf{v},\mathrm{e}}\right)_{\mathbf{v}} \end{bmatrix} & \text{if } 1 \leq i \leq \sigma, \\ & i \text{ even} \\ \mathbf{0}_{N \times N_{\mathbf{u}}} & \text{otherwise,} \end{cases} \tag{A.216k}$$

and $\dfrac{\partial^2 \mathbf{t}_{k,\sigma}}{\partial \mathbf{Y}_{\ell,i} \partial \overline{\mathbf{y}}} \overline{\mathbf{w}} = \mathbf{0}_{N \times N}$ for all other values of ℓ. Also,

$$\frac{\partial^2 \mathbf{t}_{k,\sigma}}{\partial \mathbf{u}_{\ell} \partial \overline{\mathbf{y}}} \overline{\mathbf{w}} = \begin{cases} \begin{bmatrix} -\left(\mathbf{f}_{\mathbf{u}}^{\mathbf{v}} \left(\underline{\mathbf{u}}_{\mathbf{U},k-1,\sigma}^{\mathrm{e}}\right) \underline{\mathbf{w}}_{\mathbf{W}^{\mathbf{u}},k-1,\sigma}^{\mathbf{u},\mathrm{e}}\right)_{\mathbf{u}} \\ \mathbf{0}_{N_{\mathbf{u}} \times N_{\mathbf{u}}} \end{bmatrix} & \text{if } \ \ell = k-1 \\ \mathbf{0}_{N \times N_{\mathbf{u}}} & \text{otherwise} \end{cases} \tag{A.216l}$$

and

$$\frac{\partial^2 \mathbf{t}_{k,\sigma}}{\partial \mathbf{v}_{\ell} \partial \overline{\mathbf{y}}} \overline{\mathbf{w}} = \begin{cases} \begin{bmatrix} \mathbf{0}_{N_{\mathbf{v}} \times N_{\mathbf{v}}} \\ -\left(\mathbf{f}_{\mathbf{v}}^{\mathbf{u}} \left(\underline{\mathbf{v}}_{\mathbf{V},k-\frac{1}{2},\sigma}^{\mathrm{e}}\right) \underline{\mathbf{w}}_{\mathbf{W}^{\mathbf{v}},k-\frac{1}{2},\sigma}^{\mathbf{v},\mathrm{e}}\right)_{\mathbf{v}} \end{bmatrix} & \text{if } \ \ell = k-1 \\ \mathbf{0}_{N \times N_{\mathbf{v}}} & \text{otherwise.} \end{cases} \tag{A.216m}$$

For $1 \leq \sigma \leq s$ and σ even:

$$\frac{\partial^2 \mathbf{t}_{k,\sigma}}{\partial \mathbf{U}_{k,i} \partial \overline{\mathbf{y}}} \overline{\mathbf{w}} = \begin{cases} \tau \alpha_{\sigma i} \begin{bmatrix} -\left(\mathbf{f}_{\mathbf{v}}^{\mathbf{u}} \left(\underline{\mathbf{v}}_{\mathbf{U},k-\frac{1}{2},\sigma}^{\mathrm{e}}\right) \underline{\mathbf{w}}_{\mathbf{W}^{\mathbf{u}},k-\frac{1}{2},\sigma}^{\mathbf{v},\mathrm{e}}\right)_{\mathbf{v}} \\ \mathbf{0}_{N_{\mathbf{v}} \times N_{\mathbf{v}}} \end{bmatrix} & \text{if } 1 \leq i \leq \sigma, \\ & i \text{ odd} \\ \mathbf{0}_{N \times N_{\mathbf{v}}} & \text{otherwise,} \end{cases} \tag{A.216n}$$

$$\frac{\partial^2 \mathbf{t}_{k,\sigma}}{\partial \mathbf{V}_{k,i} \partial \overline{\mathbf{y}}} \overline{\mathbf{w}} = \begin{cases} \tau \alpha_{\sigma i} \begin{bmatrix} \mathbf{0}_{N_{\mathbf{u}} \times N_{\mathbf{v}}} \\ -\left(\mathbf{f}_{\mathbf{u}}^{\mathbf{v}} \left(\underline{\mathbf{u}}_{\mathbf{V},k,\sigma}^{\mathrm{e}} \right) \underline{\mathbf{w}}_{\mathbf{W}^{\mathbf{v}},k,\sigma}^{\mathbf{u},\mathrm{e}} \right)_{\mathbf{u}} \end{bmatrix} & \text{if } 1 \leq i \leq \sigma, \\ & i \text{ odd} \\ \mathbf{0}_{N \times N_{\mathbf{u}}} & \text{otherwise,} \end{cases} \tag{A.216o}$$

and $\dfrac{\partial^2 \mathbf{t}_{k,\sigma}}{\partial \mathbf{Y}_{\ell,i} \partial \overline{\mathbf{y}}} \overline{\mathbf{w}} = \mathbf{0}_{N \times N}$ for all other values of ℓ. Also,

$$\frac{\partial^2 \mathbf{t}_{k,\sigma}}{\partial \mathbf{u}_{\ell} \partial \overline{\mathbf{y}}} \overline{\mathbf{w}} = \begin{cases} \begin{bmatrix} \mathbf{0}_{N_{\mathbf{u}} \times N_{\mathbf{u}}} \\ -\left(\mathbf{f}_{\mathbf{u}}^{\mathbf{v}} \left(\underline{\mathbf{u}}_{\mathbf{V},k,\sigma}^{\mathrm{e}} \right) \underline{\mathbf{w}}_{\mathbf{W}^{\mathbf{v}},k,\sigma}^{\mathbf{u},\mathrm{e}} \right)_{\mathbf{u}} \end{bmatrix} & \text{if } \ \ell = k \\ \mathbf{0}_{N \times N_{\mathbf{u}}} & \text{otherwise} \end{cases} \tag{A.216p}$$

and

$$\frac{\partial^2 \mathbf{t}_{k,\sigma}}{\partial \mathbf{v}_{\ell} \partial \overline{\mathbf{y}}} \overline{\mathbf{w}} = \begin{cases} \begin{bmatrix} -\left(\mathbf{f}_{\mathbf{v}}^{\mathbf{u}} \left(\underline{\mathbf{v}}_{\mathbf{U},k-\frac{1}{2},\sigma}^{\mathrm{e}} \right) \underline{\mathbf{w}}_{\mathbf{W}^{\mathbf{u}},k-\frac{1}{2},\sigma}^{\mathbf{v},\mathrm{e}} \right)_{\mathbf{v}} \\ \mathbf{0}_{N_{\mathbf{v}} \times N_{\mathbf{v}}} \end{bmatrix} & \text{if } \ \ell = k-1 \\ \mathbf{0}_{N \times N_{\mathbf{v}}} & \text{otherwise.} \end{cases} \tag{A.216q}$$

For $\sigma = s+1$:

$$\frac{\partial^2 \mathbf{t}_{k,s+1}}{\partial \widehat{\mathbf{y}}_{\ell} \partial \overline{\mathbf{y}}} \overline{\mathbf{w}} = \mathbf{0}_{N \times (s+1)N}. \tag{A.216r}$$

Derivatives of $\dfrac{\partial \mathbf{t}}{\partial \mathbf{m}}$

We consider the derivatives of the σth $N \times 1$ block $\dfrac{\partial \mathbf{t}_{k,\sigma}}{\partial \mathbf{m}} \mathbf{w}_{\mathbf{m}}$ of the vector $\dfrac{\partial \mathbf{t}_k}{\partial \mathbf{m}} \mathbf{w}_{\mathbf{m}}$, which in turn is the kth block of $\mathbf{t}_{\mathbf{y}} \overline{\mathbf{w}} = \dfrac{\partial \mathbf{t}}{\partial \mathbf{m}} \mathbf{w}_{\mathbf{m}}$.

The expression for the product $\dfrac{\partial \mathbf{t}_k}{\partial \mathbf{m}} \mathbf{w}_{\mathbf{m}}$ was given in (A.205). We now find expressions for the derivatives of (A.205) with respect to $\overline{\mathbf{y}}$, $\mathbf{m}$ and $\mathbf{y}_0$.

- $\dfrac{\partial^2 \mathbf{t}}{\partial \mathbf{m} \partial \mathbf{m}} \mathbf{w}_{\mathbf{m}}$

  We differentiate (A.205) with respect to $\mathbf{m}$:

- *For number of stages s odd*

  For $1 \leq \sigma \leq s$ and σ odd:

$$\frac{\partial^2 \mathbf{t}_{k,\sigma}}{\partial \mathbf{m} \partial \mathbf{m}} \mathbf{w}_{\mathbf{m}} = \begin{bmatrix} -\left(\mathbf{f}^{\mathbf{u}}_{\mathbf{m}} \left(\underline{\mathbf{v}}^{\mathrm{o}}_{\mathbf{U},k-\frac{1}{2},\sigma}\right) \mathbf{w}_{\mathbf{m}}\right)_{\mathbf{m}} \\ -\left(\mathbf{f}^{\mathbf{v}}_{\mathbf{m}} \left(\underline{\mathbf{u}}^{\mathrm{o}}_{\mathbf{V},k,\sigma}\right) \mathbf{w}_{\mathbf{m}}\right)_{\mathbf{m}} \end{bmatrix}. \tag{A.217a}$$

  For $1 \leq \sigma \leq s$ and σ even:

$$\frac{\partial^2 \mathbf{t}_{k,\sigma}}{\partial \mathbf{m} \partial \mathbf{m}} \mathbf{w}_{\mathbf{m}} = \begin{bmatrix} -\left(\mathbf{f}^{\mathbf{v}}_{\mathbf{m}} \left(\underline{\mathbf{u}}^{\mathrm{o}}_{\mathbf{U},k-1,\sigma}\right) \mathbf{w}_{\mathbf{m}}\right)_{\mathbf{m}} \\ -\left(\mathbf{f}^{\mathbf{u}}_{\mathbf{m}} \left(\underline{\mathbf{v}}^{\mathrm{o}}_{\mathbf{V},k-\frac{1}{2},\sigma}\right) \mathbf{w}_{\mathbf{m}}\right)_{\mathbf{m}} \end{bmatrix}. \tag{A.217b}$$

  For $\sigma = s+1$:

$$\frac{\partial^2 \mathbf{t}_{k,s+1}}{\partial \mathbf{m} \partial \mathbf{m}} \mathbf{w}_{\mathbf{m}} = \mathbf{0}_{N \times N_{\mathbf{m}}}. \tag{A.217c}$$

- *For number of stages s even*

  For $1 \leq \sigma \leq s$ and σ odd:

$$\frac{\partial^2 \mathbf{t}_{k,\sigma}}{\partial \mathbf{m} \partial \mathbf{m}} \mathbf{w}_{\mathbf{m}} = \begin{bmatrix} -\left(\mathbf{f}^{\mathbf{v}}_{\mathbf{m}} \left(\underline{\mathbf{u}}^{\mathrm{e}}_{\mathbf{U},k-1,\sigma}\right) \mathbf{w}_{\mathbf{m}}\right)_{\mathbf{m}} \\ -\left(\mathbf{f}^{\mathbf{u}}_{\mathbf{m}} \left(\underline{\mathbf{v}}^{\mathrm{e}}_{\mathbf{V},k-\frac{1}{2},\sigma}\right) \mathbf{w}_{\mathbf{m}}\right)_{\mathbf{m}} \end{bmatrix}. \tag{A.217d}$$

  For $1 \leq \sigma \leq s$ and σ even:

$$\frac{\partial^2 \mathbf{t}_{k,\sigma}}{\partial \mathbf{m} \partial \mathbf{m}} \mathbf{w}_{\mathbf{m}} = \begin{bmatrix} -\left(\mathbf{f}^{\mathbf{u}}_{\mathbf{m}} \left(\underline{\mathbf{v}}^{\mathrm{e}}_{\mathbf{U},k-\frac{1}{2},\sigma}\right) \mathbf{w}_{\mathbf{m}}\right)_{\mathbf{m}} \\ -\left(\mathbf{f}^{\mathbf{v}}_{\mathbf{m}} \left(\underline{\mathbf{u}}^{\mathrm{e}}_{\mathbf{V},k,\sigma}\right) \mathbf{w}_{\mathbf{m}}\right)_{\mathbf{m}} \end{bmatrix}. \tag{A.217e}$$

  For $\sigma = s+1$:

$$\frac{\partial^2 \mathbf{t}_{k,s+1}}{\partial \mathbf{m} \partial \mathbf{m}} \mathbf{w}_{\mathbf{m}} = \mathbf{0}_{N \times N_{\mathbf{m}}}. \tag{A.217f}$$

- $\dfrac{\partial^2 \mathbf{t}}{\partial \mathbf{y}_0 \partial \mathbf{m}} \mathbf{w}_{\mathbf{m}}$

  Taking the derivative of (A.205) with respect to $\mathbf{y}_0$ gives

- *For number of stages s odd*

For $1 \leq \sigma \leq s$ and σ odd:

$$\frac{\partial^2 \mathbf{t}_{1,\sigma}}{\partial \mathbf{y}_0 \partial \mathbf{m}} \mathbf{w}_\mathbf{m} = \begin{bmatrix} \mathbf{0}_{N_\mathbf{u} \times N_\mathbf{u}} & -\left(\mathbf{f}^\mathbf{u}_\mathbf{m}\left(\underline{\mathbf{v}}^{\mathrm{o}}_{\mathbf{U},\frac{1}{2},\sigma}\right)\mathbf{w}_\mathbf{m}\right)_\mathbf{v} \\ \mathbf{0}_{N_\mathbf{v} \times N_\mathbf{u}} & \mathbf{0}_{N_\mathbf{v} \times N_\mathbf{v}} \end{bmatrix}. \tag{A.218a}$$

For $1 \leq \sigma \leq s$ and σ even:

$$\frac{\partial^2 \mathbf{t}_{1,\sigma}}{\partial \mathbf{y}_0 \partial \mathbf{m}} \mathbf{w}_\mathbf{m} = \begin{bmatrix} -\left(\mathbf{f}^\mathbf{v}_\mathbf{m}\left(\underline{\mathbf{u}}^{\mathrm{o}}_{\mathbf{U},0,\sigma}\right)\mathbf{w}_\mathbf{m}\right)_\mathbf{u} & \mathbf{0}_{N_\mathbf{v} \times N_\mathbf{v}} \\ \mathbf{0}_{N_\mathbf{u} \times N_\mathbf{u}} & -\left(\mathbf{f}^\mathbf{u}_\mathbf{m}\left(\underline{\mathbf{v}}^{\mathrm{o}}_{\mathbf{V},\frac{1}{2},\sigma}\right)\mathbf{w}_\mathbf{m}\right)_\mathbf{v} \end{bmatrix}. \tag{A.218b}$$

For $\sigma = s+1$:

$$\frac{\partial^2 \mathbf{t}_{1,s+1}}{\partial \mathbf{y}_0 \partial \mathbf{m}} \mathbf{w}_\mathbf{m} = \mathbf{0}_{N \times N}. \tag{A.218c}$$

- *For number of stages s even*

For $1 \leq \sigma \leq s$ and σ odd:

$$\frac{\partial^2 \mathbf{t}_{1,\sigma}}{\partial \mathbf{y}_0 \partial \mathbf{m}} \mathbf{w}_\mathbf{m} = \begin{bmatrix} -\left(\mathbf{f}^\mathbf{v}_\mathbf{m}\left(\underline{\mathbf{u}}^{\mathrm{e}}_{\mathbf{U},0,\sigma}\right)\mathbf{w}_\mathbf{m}\right)_\mathbf{u} & \mathbf{0}_{N_\mathbf{v} \times N_\mathbf{v}} \\ \mathbf{0}_{N_\mathbf{u} \times N_\mathbf{u}} & -\left(\mathbf{f}^\mathbf{u}_\mathbf{m}\left(\underline{\mathbf{v}}^{\mathrm{e}}_{\mathbf{V},\frac{1}{2},\sigma}\right)\mathbf{w}_\mathbf{m}\right)_\mathbf{v} \end{bmatrix}. \tag{A.218d}$$

For $1 \leq \sigma \leq s$ and σ even:

$$\frac{\partial^2 \mathbf{t}_{1,\sigma}}{\partial \mathbf{y}_0 \partial \mathbf{m}} \mathbf{w}_\mathbf{m} = \begin{bmatrix} \mathbf{0}_{N_\mathbf{u} \times N_\mathbf{u}} & -\left(\mathbf{f}^\mathbf{u}_\mathbf{m}\left(\underline{\mathbf{v}}^{\mathrm{e}}_{\mathbf{U},\frac{1}{2},\sigma}\right)\mathbf{w}_\mathbf{m}\right)_\mathbf{v} \\ \mathbf{0}_{N_\mathbf{v} \times N_\mathbf{u}} & \mathbf{0}_{N_\mathbf{v} \times N_\mathbf{v}} \end{bmatrix}. \tag{A.218e}$$

For $\sigma = s+1$:

$$\frac{\partial^2 \mathbf{t}_{1,s+1}}{\partial \mathbf{y}_0 \partial \mathbf{m}} \mathbf{w}_\mathbf{m} = \mathbf{0}_{N \times N}. \tag{A.218f}$$

The derivatives $\frac{\partial^2 \mathbf{t}_{k,\sigma}}{\partial \mathbf{y}_0 \partial \mathbf{m}} \mathbf{w}_\mathbf{m}$ are $\mathbf{0}_{N \times N}$ for all $k > 1$, $1 \leq \sigma \leq s+1$.

- $\frac{\partial^2 \mathbf{t}}{\partial \mathbf{y} \partial \mathbf{m}} \mathbf{w}_\mathbf{m}$

Consider the ℓth term of

$$\frac{\partial^2 \mathbf{t}_{k,\sigma}}{\partial \mathbf{y} \partial \mathbf{m}} \mathbf{w}_{\mathbf{m}} = \left[\frac{\partial^2 \mathbf{t}_{k,\sigma}}{\partial \widehat{\mathbf{y}}_1 \partial \mathbf{m}} \mathbf{w}_{\mathbf{m}} \quad \frac{\partial^2 \mathbf{t}_{k,\sigma}}{\partial \widehat{\mathbf{y}}_2 \partial \mathbf{m}} \mathbf{w}_{\mathbf{m}} \quad \cdots \quad \frac{\partial^2 \mathbf{t}_{k,\sigma}}{\partial \widehat{\mathbf{y}}_K \partial \mathbf{m}} \mathbf{w}_{\mathbf{m}} \right], \tag{A.219}$$

with

$$\frac{\partial^2 \mathbf{t}_{k,\sigma}}{\partial \widehat{\mathbf{y}}_\ell \partial \mathbf{m}} \mathbf{w}_{\mathbf{m}} = \left[\frac{\partial^2 \mathbf{t}_{k,\sigma}}{\partial \mathbf{Y}_{\ell,1} \partial \mathbf{m}} \mathbf{w}_{\mathbf{m}} \quad \cdots \quad \frac{\partial^2 \mathbf{t}_{k,\sigma}}{\partial \mathbf{Y}_{\ell,s} \partial \mathbf{m}} \mathbf{w}_{\mathbf{m}} \quad \frac{\partial^2 \mathbf{t}_{k,\sigma}}{\partial \mathbf{y}_\ell \partial \mathbf{m}} \mathbf{w}_{\mathbf{m}} \right], \tag{A.220}$$

where

$$\begin{aligned} \frac{\partial^2 \mathbf{t}_{k,\sigma}}{\partial \mathbf{Y}_{\ell,i} \partial \mathbf{m}} \mathbf{w}_{\mathbf{m}} &= \left[\frac{\partial^2 \mathbf{t}_{k,\sigma}}{\partial \mathbf{U}_{\ell,i} \partial \mathbf{m}} \mathbf{w}_{\mathbf{m}} \quad \frac{\partial^2 \mathbf{t}_{k,\sigma}}{\partial \mathbf{V}_{\ell,i} \partial \mathbf{m}} \mathbf{w}_{\mathbf{m}} \right] \qquad \text{for } 1 \leq i \leq s \\ \frac{\partial^2 \mathbf{t}_{k,\sigma}}{\partial \mathbf{y}_\ell \partial \mathbf{m}} \mathbf{w}_{\mathbf{m}} &= \left[\frac{\partial^2 \mathbf{t}_{k,\sigma}}{\partial \mathbf{u}_\ell \partial \mathbf{m}} \mathbf{w}_{\mathbf{m}} \quad \frac{\partial^2 \mathbf{t}_{k,\sigma}}{\partial \mathbf{v}_\ell \partial \mathbf{m}} \mathbf{w}_{\mathbf{m}} \right]. \end{aligned}$$

Then

- *For number of stages s odd*

  For $1 \leq \sigma \leq s$ and σ odd:

$$\frac{\partial^2 \mathbf{t}_{k,\sigma}}{\partial \mathbf{U}_{k,i} \partial \mathbf{m}} \mathbf{w}_{\mathbf{m}} = \begin{cases} \tau \alpha_{\sigma i} \begin{bmatrix} -\left(\mathbf{f}^{\mathbf{u}}_{\mathbf{m}} \left(\underline{\mathbf{v}}^{\mathrm{o}}_{\mathbf{U},k-\frac{1}{2},\sigma} \right) \mathbf{w}_{\mathbf{m}} \right)_{\mathbf{v}} \\ \mathbf{0}_{N_{\mathbf{v}} \times N_{\mathbf{v}}} \end{bmatrix} & \begin{array}{c} \text{if } 1 \leq i \leq \sigma, \\ i \text{ even} \end{array} \\ \mathbf{0}_{N \times N_{\mathbf{v}}} & \text{otherwise,} \end{cases} \tag{A.221a}$$

$$\frac{\partial^2 \mathbf{t}_{k,\sigma}}{\partial \mathbf{V}_{k,i} \partial \mathbf{m}} \mathbf{w}_{\mathbf{m}} = \begin{cases} \tau \alpha_{\sigma i} \begin{bmatrix} \mathbf{0}_{N_{\mathbf{v}} \times N_{\mathbf{u}}} \\ -\left(\mathbf{f}^{\mathbf{v}}_{\mathbf{m}} \left(\underline{\mathbf{u}}^{\mathrm{o}}_{\mathbf{V},k,\sigma} \right) \mathbf{w}_{\mathbf{m}} \right)_{\mathbf{u}} \end{bmatrix} & \begin{array}{c} \text{if } 1 \leq i \leq \sigma, \\ i \text{ even} \end{array} \\ \mathbf{0}_{N \times N_{\mathbf{u}}} & \text{otherwise,} \end{cases} \tag{A.221b}$$

and $\dfrac{\partial^2 \mathbf{t}_{k,\sigma}}{\partial \mathbf{Y}_{\ell,i} \partial \mathbf{m}} \mathbf{w_m} = \mathbf{0}_{N\times N}$ for all other values of ℓ. Also,

$$\frac{\partial^2 \mathbf{t}_{k,\sigma}}{\partial \mathbf{u}_\ell \partial \mathbf{m}} \mathbf{w_m} = \begin{cases} \begin{bmatrix} \mathbf{0}_{N_\mathbf{u}\times N_\mathbf{u}} \\ -\left(\mathbf{f}^\mathbf{v}_\mathbf{m}\left(\underline{\mathbf{u}}^\mathrm{o}_{\mathbf{V},k,\sigma}\right)\mathbf{w_m}\right)_\mathbf{u} \end{bmatrix} & \text{if } \ell = k \\ \mathbf{0}_{N\times N_\mathbf{u}} & \text{otherwise} \end{cases} \tag{A.221c}$$

and

$$\frac{\partial^2 \mathbf{t}_{k,\sigma}}{\partial \mathbf{v}_\ell \partial \mathbf{m}} \mathbf{w_m} = \begin{cases} \begin{bmatrix} -\left(\mathbf{f}^\mathbf{u}_\mathbf{m}\left(\underline{\mathbf{v}}^\mathrm{o}_{\mathbf{U},k-\frac{1}{2},\sigma}\right)\mathbf{w_m}\right)_\mathbf{v} \\ \mathbf{0}_{N_\mathbf{v}\times N_\mathbf{v}} \end{bmatrix} & \text{if } \ell = k-1 \\ \mathbf{0}_{N\times N_\mathbf{v}} & \text{otherwise.} \end{cases} \tag{A.221d}$$

For $1 \le \sigma \le s$ and σ even:

$$\frac{\partial^2 \mathbf{t}_{k,\sigma}}{\partial \mathbf{U}_{k,i} \partial \mathbf{m}} \mathbf{w_m} = \begin{cases} \tau\alpha_{\sigma i} \begin{bmatrix} -\left(\mathbf{f}^\mathbf{v}_\mathbf{m}\left(\underline{\mathbf{u}}^\mathrm{o}_{\mathbf{U},k-1,\sigma}\right)\mathbf{w_m}\right)_\mathbf{u} \\ \mathbf{0}_{N_\mathbf{u}\times N_\mathbf{u}} \end{bmatrix} & \text{if } 1 \le i \le \sigma, \ i \text{ odd} \\ \mathbf{0}_{N\times N_\mathbf{u}} & \text{otherwise,} \end{cases} \tag{A.221e}$$

$$\frac{\partial^2 \mathbf{t}_{k,\sigma}}{\partial \mathbf{V}_{k,i} \partial \mathbf{m}} \mathbf{w_m} = \begin{cases} \tau\alpha_{\sigma i} \begin{bmatrix} \mathbf{0}_{N_\mathbf{u}\times N_\mathbf{v}} \\ -\left(\mathbf{f}^\mathbf{u}_\mathbf{m}\left(\underline{\mathbf{v}}^\mathrm{o}_{\mathbf{V},k-\frac{1}{2},\sigma}\right)\mathbf{w_m}\right)_\mathbf{v} \end{bmatrix} & \text{if } 1 \le i \le \sigma, \ i \text{ odd} \\ \mathbf{0}_{N\times N_\mathbf{v}} & \text{otherwise,} \end{cases} \tag{A.221f}$$

and $\dfrac{\partial^2 \mathbf{t}_{k,\sigma}}{\partial \mathbf{Y}_{\ell,i} \partial \mathbf{m}} \mathbf{w_m} = \mathbf{0}_{N\times N}$ for all other values of ℓ. Also,

$$\frac{\partial^2 \mathbf{t}_{k,\sigma}}{\partial \mathbf{u}_\ell \partial \mathbf{m}} \mathbf{w_m} = \begin{cases} \begin{bmatrix} -\left(\mathbf{f}^\mathbf{v}_\mathbf{m}\left(\underline{\mathbf{u}}^\mathrm{o}_{\mathbf{U},k-1,\sigma}\right)\mathbf{w_m}\right)_\mathbf{u} \\ \mathbf{0}_{N_\mathbf{u}\times N_\mathbf{u}} \end{bmatrix} & \text{if } \ell = k-1 \\ \mathbf{0}_{N\times N_\mathbf{u}} & \text{otherwise} \end{cases}. \tag{A.221g}$$

and

$$\frac{\partial^2 \mathbf{t}_{k,\sigma}}{\partial \mathbf{v}_\ell \partial \mathbf{m}} \mathbf{w}_\mathbf{m} = \begin{cases} \begin{bmatrix} \mathbf{0}_{N_\mathbf{v} \times N_\mathbf{v}} \\ -\left(\mathbf{f}^\mathbf{u}_\mathbf{m}\left(\underline{\mathbf{v}}^{\mathrm{o}}_{\mathbf{U},k-\frac{1}{2},\sigma}\right)\mathbf{w}_\mathbf{m}\right)_\mathbf{v} \end{bmatrix} & \text{if } \ell = k-1 \\ \mathbf{0}_{N \times N_\mathbf{v}} & \text{otherwise.} \end{cases} \tag{A.221h}$$

For $\sigma = s+1$:

$$\frac{\partial^2 \mathbf{t}_{k,s+1}}{\partial \widehat{\mathbf{y}}_\ell \partial \mathbf{m}} \mathbf{w}_\mathbf{m} = \mathbf{0}_{N \times (s+1)N}. \tag{A.221i}$$

- *For number of stages s even*

For $1 \leq \sigma \leq s$ and σ odd:

$$\frac{\partial^2 \mathbf{t}_{k,\sigma}}{\partial \mathbf{U}_{k,i} \partial \mathbf{m}} \mathbf{w}_\mathbf{m} = \begin{cases} \tau\alpha_{\sigma i} \begin{bmatrix} -\left(\mathbf{f}^\mathbf{v}_\mathbf{m}\left(\underline{\mathbf{u}}^{\mathrm{e}}_{\mathbf{U},k-1,\sigma}\right)\mathbf{w}_\mathbf{m}\right)_\mathbf{u} \\ \mathbf{0}_{N_\mathbf{v} \times N_\mathbf{u}} \end{bmatrix} & \text{if } 1 \leq i \leq \sigma, \\ & i \text{ even} \\ \mathbf{0}_{N \times N_\mathbf{u}} & \text{otherwise,} \end{cases} \tag{A.221j}$$

$$\frac{\partial^2 \mathbf{t}_{k,\sigma}}{\partial \mathbf{V}_{k,i} \partial \mathbf{m}} \mathbf{w}_\mathbf{m} = \begin{cases} \tau\alpha_{\sigma i} \begin{bmatrix} \mathbf{0}_{N_\mathbf{v} \times N_\mathbf{v}} \\ -\left(\mathbf{f}^\mathbf{u}_\mathbf{m}\left(\underline{\mathbf{v}}^{\mathrm{e}}_{\mathbf{V},k-\frac{1}{2},\sigma}\right)\mathbf{w}_\mathbf{m}\right)_\mathbf{v} \end{bmatrix} & \text{if } 1 \leq i \leq \sigma, \\ & i \text{ even} \\ \mathbf{0}_{N \times N_\mathbf{v}} & \text{otherwise,} \end{cases} \tag{A.221k}$$

and $\dfrac{\partial^2 \mathbf{t}_{k,\sigma}}{\partial \mathbf{Y}_{\ell,i} \partial \mathbf{m}} \mathbf{w}_\mathbf{m} = \mathbf{0}_{N \times N}$ for all other values of ℓ. Also,

$$\frac{\partial^2 \mathbf{t}_{k,\sigma}}{\partial \mathbf{u}_\ell \partial \mathbf{m}} \mathbf{w}_\mathbf{m} = \begin{cases} \begin{bmatrix} -\left(\mathbf{f}^\mathbf{v}_\mathbf{m}\left(\underline{\mathbf{u}}^{\mathrm{e}}_{\mathbf{U},k-1,\sigma}\right)\mathbf{w}_\mathbf{m}\right)_\mathbf{u} \\ \mathbf{0}_{N_\mathbf{u} \times N_\mathbf{u}} \end{bmatrix} & \text{if } \ell = k-1 \\ \mathbf{0}_{N \times N_\mathbf{u}} & \text{otherwise} \end{cases} \tag{A.221l}$$

and

$$\frac{\partial^2 \mathbf{t}_{k,\sigma}}{\partial \mathbf{v}_\ell \partial \mathbf{m}} \mathbf{w}_\mathbf{m} = \begin{cases} \begin{bmatrix} \mathbf{0}_{N_\mathbf{v} \times N_\mathbf{v}} \\ -\left(\mathbf{f}^\mathbf{u}_\mathbf{m}\left(\underline{\mathbf{v}}^\mathrm{e}_{\mathbf{V},k-\frac{1}{2},\sigma}\right)\mathbf{w}_\mathbf{m}\right)_\mathbf{v} \end{bmatrix} & \text{if } \ell = k-1 \\ \mathbf{0}_{N \times N_\mathbf{v}} & \text{otherwise.} \end{cases} \tag{A.221m}$$

For $1 \leq \sigma \leq s$ and σ even:

$$\frac{\partial^2 \mathbf{t}_{k,\sigma}}{\partial \mathbf{U}_{k,i} \partial \mathbf{m}} \mathbf{w}_\mathbf{m} = \begin{cases} \tau\alpha_{\sigma i} \begin{bmatrix} -\left(\mathbf{f}^\mathbf{u}_\mathbf{m}\left(\underline{\mathbf{v}}^\mathrm{e}_{\mathbf{U},k-\frac{1}{2},\sigma}\right)\mathbf{w}_\mathbf{m}\right)_\mathbf{v} \\ \mathbf{0}_{N_\mathbf{v} \times N_\mathbf{v}} \end{bmatrix} & \text{if } 1 \leq i \leq \sigma, \\ & i \text{ odd} \\ \mathbf{0}_{N \times N_\mathbf{v}} & \text{otherwise,} \end{cases} \tag{A.221n}$$

$$\frac{\partial^2 \mathbf{t}_{k,\sigma}}{\partial \mathbf{V}_{k,i} \partial \mathbf{m}} \mathbf{w}_\mathbf{m} = \begin{cases} \tau\alpha_{\sigma i} \begin{bmatrix} \mathbf{0}_{N_\mathbf{u} \times N_\mathbf{v}} \\ -\left(\mathbf{f}^\mathbf{v}_\mathbf{m}\left(\underline{\mathbf{u}}^\mathrm{e}_{\mathbf{V},k,\sigma}\right)\mathbf{w}_\mathbf{m}\right)_\mathbf{u} \end{bmatrix} & \text{if } 1 \leq i \leq \sigma, \\ & i \text{ odd} \\ \mathbf{0}_{N \times N_\mathbf{u}} & \text{otherwise,} \end{cases} \tag{A.221o}$$

and $\dfrac{\partial^2 \mathbf{t}_{k,\sigma}}{\partial \mathbf{Y}_{\ell,i} \partial \mathbf{m}} \mathbf{w}_\mathbf{m} = \mathbf{0}_{N \times N}$ for all other values of ℓ. Also,

$$\frac{\partial^2 \mathbf{t}_{k,\sigma}}{\partial \mathbf{u}_\ell \partial \mathbf{m}} \mathbf{w}_\mathbf{m} = \begin{cases} \begin{bmatrix} \mathbf{0}_{N_\mathbf{u} \times N_\mathbf{u}} \\ -\left(\mathbf{f}^\mathbf{v}_\mathbf{m}\left(\underline{\mathbf{u}}^\mathrm{e}_{\mathbf{V},k,\sigma}\right)\mathbf{w}_\mathbf{m}\right)_\mathbf{u} \end{bmatrix} & \text{if } \ell = k \\ \mathbf{0}_{N \times N_\mathbf{u}} & \text{otherwise} \end{cases} \tag{A.221p}$$

and

$$\frac{\partial^2 \mathbf{t}_{k,\sigma}}{\partial \mathbf{v}_\ell \partial \mathbf{m}} \mathbf{w}_\mathbf{m} = \begin{cases} \begin{bmatrix} -\left(\mathbf{f}^\mathbf{u}_\mathbf{m}\left(\underline{\mathbf{v}}^\mathrm{e}_{\mathbf{U},k-\frac{1}{2},\sigma}\right)\mathbf{w}_\mathbf{m}\right)_\mathbf{v} \\ \mathbf{0}_{N_\mathbf{v} \times N_\mathbf{v}} \end{bmatrix} & \text{if } \ell = k-1 \\ \mathbf{0}_{N \times N_\mathbf{v}} & \text{otherwise.} \end{cases} \tag{A.221q}$$

For $\sigma = s+1$:

$$\frac{\partial^2 \mathbf{t}_{k,s+1}}{\partial \widehat{\mathbf{y}}_\ell \partial \mathbf{m}} \mathbf{w}_\mathbf{m} = \mathbf{0}_{N\times(s+1)N}. \tag{A.221r}$$

Derivatives of $\dfrac{\partial \mathbf{t}}{\partial \mathbf{y}_0}$

We consider the derivatives of the σth block $\dfrac{\partial \mathbf{t}_{k,\sigma}}{\partial \mathbf{y}_0}\mathbf{w}_0$ of the vector $\dfrac{\partial \mathbf{t}_k}{\partial \mathbf{y}_0}\mathbf{w}_0$, which in turn is the kth block of $\dfrac{\partial \mathbf{t}}{\partial \mathbf{y}_0}\mathbf{w}_0 = \dfrac{\partial \mathbf{t}}{\partial \mathbf{y}_0}\mathbf{w}_0$.

The expression for the product $\dfrac{\partial \mathbf{t}_k}{\partial \mathbf{y}_0}\mathbf{w}_0$ was given in (A.210). We now find expressions for the derivatives of (A.210) with respect to $\overline{\mathbf{y}}$, $\mathbf{m}$ and $\mathbf{y}_0$.

- $\dfrac{\partial^2 \mathbf{t}}{\partial \mathbf{m} \partial \mathbf{y}_0}\mathbf{w}_0$

  We differentiate (A.210) with respect to $\mathbf{m}$:

  - *For number of stages s odd*

    For $1 \le \sigma \le s$ and σ odd:

$$\frac{\partial^2 \mathbf{t}_{1,\sigma}}{\partial \mathbf{m} \partial \mathbf{y}_0}\mathbf{w}_0 = \begin{bmatrix} -\left(\mathbf{f}^{\mathbf{u}}_{\mathbf{v}}\left(\underline{\mathbf{v}}^{\mathrm{o}}_{\mathbf{U},\frac{1}{2},\sigma}\right)\mathbf{w}^{\mathbf{v}}_0\right)_{\mathbf{m}} \\ \mathbf{0}_{N_\mathbf{v}\times N_\mathbf{m}} \end{bmatrix}. \tag{A.222a}$$

    For $1 \le \sigma \le s$ and σ even:

$$\frac{\partial^2 \mathbf{t}_{1,\sigma}}{\partial \mathbf{m} \partial \mathbf{y}_0}\mathbf{w}_0 = \begin{bmatrix} -\left(\mathbf{f}^{\mathbf{v}}_{\mathbf{u}}\left(\underline{\mathbf{u}}^{\mathrm{o}}_{\mathbf{U},0,\sigma}\right)\mathbf{w}^{\mathbf{u}}_0\right)_{\mathbf{m}} \\ -\left(\mathbf{f}^{\mathbf{u}}_{\mathbf{v}}\left(\underline{\mathbf{v}}^{\mathrm{o}}_{\mathbf{V},k-\frac{1}{2},\sigma}\right)\mathbf{w}^{\mathbf{v}}_0\right)_{\mathbf{m}} \end{bmatrix}. \tag{A.222b}$$

    For $\sigma = s+1$:

$$\frac{\partial^2 \mathbf{t}_{1,s+1}}{\partial \mathbf{m} \partial \mathbf{y}_0}\mathbf{w}_0 = \mathbf{0}_{N\times N_\mathbf{m}}. \tag{A.222c}$$

  - *For number of stages s even*

For $1 \leq \sigma \leq s$ and σ odd:

$$\frac{\partial^2 \mathbf{t}_{1,\sigma}}{\partial \mathbf{m} \partial \mathbf{y}_0} \mathbf{w}_0 = \begin{bmatrix} -\left(\mathbf{f}_{\mathbf{u}}^{\mathbf{v}} \left(\underline{\mathbf{u}}_{\mathbf{U},0,\sigma}^{\mathrm{e}}\right) \mathbf{w}_0^{\mathbf{u}}\right)_{\mathbf{m}} \\ -\left(\mathbf{f}_{\mathbf{v}}^{\mathbf{u}} \left(\underline{\mathbf{v}}_{\mathbf{V},\frac{1}{2},\sigma}^{\mathrm{e}}\right) \mathbf{w}_0^{\mathbf{v}}\right)_{\mathbf{m}} \end{bmatrix}. \tag{A.222d}$$

For $1 \leq \sigma \leq s$ and σ even:

$$\frac{\partial^2 \mathbf{t}_{1,\sigma}}{\partial \mathbf{m} \partial \mathbf{y}_0} \mathbf{w}_0 = \begin{bmatrix} -\left(\mathbf{f}_{\mathbf{v}}^{\mathbf{u}} \left(\underline{\mathbf{v}}_{\mathbf{U},\frac{1}{2},\sigma}^{\mathrm{e}}\right) \mathbf{w}_0^{\mathbf{v}}\right)_{\mathbf{m}} \\ \mathbf{0}_{N_{\mathbf{v}} \times N_{\mathbf{m}}} \end{bmatrix}. \tag{A.222e}$$

For $\sigma = s + 1$:

$$\frac{\partial^2 \mathbf{t}_{1,s+1}}{\partial \mathbf{m} \partial \mathbf{y}_0} \mathbf{w}_0 = \mathbf{0}_{N \times N_{\mathbf{m}}}. \tag{A.222f}$$

We have $\dfrac{\partial^2 \mathbf{t}_{1,\sigma}}{\partial \mathbf{m} \partial \mathbf{y}_0} \mathbf{w}_0 = \mathbf{0}_{N \times N_{\mathbf{m}}}$ for $k > 1$.

- $\dfrac{\partial^2 \mathbf{t}}{\partial \mathbf{y}_0 \partial \mathbf{y}_0} \mathbf{w}_0$

Taking the derivative of (A.198) with respect to $\mathbf{y}_0$ gives

- *For number of stages s odd*

For $1 \leq \sigma \leq s$ and σ odd:

$$\frac{\partial^2 \mathbf{t}_{1,\sigma}}{\partial \mathbf{y}_0 \partial \mathbf{y}_0} \mathbf{w}_0 = \begin{bmatrix} \mathbf{0}_{N_{\mathbf{u}} \times N_{\mathbf{u}}} & -\left(\mathbf{f}_{\mathbf{v}}^{\mathbf{u}} \left(\underline{\mathbf{v}}_{\mathbf{U},\frac{1}{2},\sigma}^{\mathrm{o}}\right) \mathbf{w}_0^{\mathbf{v}}\right)_{\mathbf{v}} \\ \mathbf{0}_{N_{\mathbf{v}} \times N_{\mathbf{u}}} & \mathbf{0}_{N_{\mathbf{v}} \times N_{\mathbf{v}}} \end{bmatrix}. \tag{A.223a}$$

For $1 \leq \sigma \leq s$ and σ even:

$$\frac{\partial^2 \mathbf{t}_{1,\sigma}}{\partial \mathbf{y}_0 \partial \mathbf{y}_0} \mathbf{w}_0 = \begin{bmatrix} -\left(\mathbf{f}_{\mathbf{u}}^{\mathbf{v}} \left(\underline{\mathbf{u}}_{\mathbf{U},0,\sigma}^{\mathrm{o}}\right) \mathbf{w}_0^{\mathbf{u}}\right)_{\mathbf{u}} & \mathbf{0}_{N_{\mathbf{v}} \times N_{\mathbf{v}}} \\ \mathbf{0}_{N_{\mathbf{u}} \times N_{\mathbf{u}}} & -\left(\mathbf{f}_{\mathbf{v}}^{\mathbf{u}} \left(\underline{\mathbf{v}}_{\mathbf{V},\frac{1}{2},\sigma}^{\mathrm{o}}\right) \mathbf{w}_0^{\mathbf{v}}\right)_{\mathbf{v}} \end{bmatrix}. \tag{A.223b}$$

For $\sigma = s + 1$:

$$\frac{\partial^2 \mathbf{t}_{1,s+1}}{\partial \mathbf{y}_0 \partial \mathbf{y}_0} \mathbf{w}_0 = \mathbf{0}_{N \times N}. \tag{A.223c}$$

- *For number of stages s even*

  For $1 \leq \sigma \leq s$ and σ odd:

$$\frac{\partial^2 \mathbf{t}_{1,\sigma}}{\partial \mathbf{y}_0 \partial \overline{\mathbf{y}}} \overline{\mathbf{w}} = \begin{bmatrix} -\left(\mathbf{f}_{\mathbf{u}}^{\mathbf{v}}\left(\underline{\mathbf{u}}_{\mathbf{U},0,\sigma}^{\mathrm{e}}\right)\mathbf{w}_0^{\mathbf{u}}\right)_{\mathbf{u}} & \mathbf{0}_{N_{\mathbf{v}}\times N_{\mathbf{v}}} \\ \mathbf{0}_{N_{\mathbf{u}}\times N_{\mathbf{u}}} & -\left(\mathbf{f}_{\mathbf{v}}^{\mathbf{u}}\left(\underline{\mathbf{v}}_{\mathbf{V},\frac{1}{2},\sigma}^{\mathrm{e}}\right)\mathbf{w}_0^{\mathbf{v}}\right)_{\mathbf{v}} \end{bmatrix}. \tag{A.223d}$$

  For $1 \leq \sigma \leq s$ and σ even:

$$\frac{\partial^2 \mathbf{t}_{1,\sigma}}{\partial \mathbf{y}_0 \partial \overline{\mathbf{y}}} \overline{\mathbf{w}} = \begin{bmatrix} \mathbf{0}_{N_{\mathbf{u}}\times N_{\mathbf{u}}} & -\left(\mathbf{f}_{\mathbf{v}}^{\mathbf{u}}\left(\underline{\mathbf{v}}_{\mathbf{U},\frac{1}{2},\sigma}^{\mathrm{e}}\right)\mathbf{w}_0^{\mathbf{v}}\right)_{\mathbf{v}} \\ \mathbf{0}_{N_{\mathbf{v}}\times N_{\mathbf{u}}} & \mathbf{0}_{N_{\mathbf{v}}\times N_{\mathbf{v}}} \end{bmatrix}. \tag{A.223e}$$

  For $\sigma = s+1$:

$$\frac{\partial^2 \mathbf{t}_{1,s+1}}{\partial \mathbf{y}_0 \partial \mathbf{y}_0} \mathbf{w}_0 = \mathbf{0}_{N\times N}. \tag{A.223f}$$

The derivatives $\dfrac{\partial^2 \mathbf{t}_{1,\sigma}}{\partial \mathbf{y}_0 \partial \mathbf{y}_0} \mathbf{w}_0$ are $\mathbf{0}_{N\times N}$ for all $k > 1$, $1 \leq \sigma \leq s+1$.

- $\dfrac{\partial^2 \mathbf{t}}{\partial \overline{\mathbf{y}} \partial \mathbf{y}_0} \mathbf{w}_0$

It is immediately clear that

$$\frac{\partial^2 \mathbf{t}_{k,\sigma}}{\partial \overline{\mathbf{y}} \partial \mathbf{y}_0} \mathbf{w}_0 = \mathbf{0}_{(s+1)KN\times N}$$

for all $k > 1$ and that

$$\frac{\partial^2 \mathbf{t}_{1,\sigma}}{\partial \widehat{\mathbf{y}}_\ell \partial \mathbf{y}_0} \mathbf{w}_0 = \mathbf{0}_{(s+1)N\times N}$$

for $\ell > 1$. Since

$$\frac{\partial^2 \mathbf{t}_{1,\sigma}}{\partial \mathbf{Y}_{1,i} \partial \mathbf{y}_0} \mathbf{w}_0 = \begin{bmatrix} \dfrac{\partial^2 \mathbf{t}_{1,\sigma}}{\partial \mathbf{U}_{1,i} \partial \mathbf{y}_0} \mathbf{w}_0 & \dfrac{\partial^2 \mathbf{t}_{1,\sigma}}{\partial \mathbf{V}_{1,i} \partial \mathbf{y}_0} \mathbf{w}_0 \end{bmatrix} \qquad \text{for } 1 \leq i \leq s,$$

we have

- *For number of stages s odd*

For $1 \leq \sigma \leq s$ and σ odd:

$$\frac{\partial^2 \mathbf{t}_{1,\sigma}}{\partial \mathbf{Y}_{1,i} \partial \mathbf{y}_0} \mathbf{w}_0 = \begin{cases} \tau \alpha_{\sigma i} \begin{bmatrix} -\left(\mathbf{f}_{\mathbf{v}}^{\mathbf{u}} \left(\underline{\mathbf{v}}^{\mathrm{o}}_{\mathbf{U},\frac{1}{2},\sigma}\right) \mathbf{w}_0^{\mathbf{v}}\right)_{\mathbf{v}} & \mathbf{0}_{N_{\mathbf{u}} \times N_{\mathbf{u}}} \\ \mathbf{0}_{N_{\mathbf{v}} \times N_{\mathbf{v}}} & \mathbf{0}_{N_{\mathbf{v}} \times N_{\mathbf{u}}} \end{bmatrix} & \text{if } 1 \leq i \leq \sigma, \\ & i \text{ even} \\ \mathbf{0}_{N \times N} & \text{otherwise.} \end{cases} \tag{A.224a}$$

For $1 \leq \sigma \leq s$ and σ even:

$$\frac{\partial^2 \mathbf{t}_{1,\sigma}}{\partial \mathbf{U}_{1,i} \partial \mathbf{y}_0} \mathbf{w}_0 = \begin{cases} \tau \alpha_{\sigma i} \begin{bmatrix} -\left(\mathbf{f}_{\mathbf{u}}^{\mathbf{v}} \left(\underline{\mathbf{u}}^{\mathrm{o}}_{\mathbf{U},0,\sigma}\right) \mathbf{w}_0^{\mathbf{u}}\right)_{\mathbf{u}} \\ \mathbf{0}_{N_{\mathbf{u}} \times N_{\mathbf{u}}} \end{bmatrix} & \text{if } 1 \leq i \leq \sigma, \\ & i \text{ odd} \\ \mathbf{0}_{N \times N_{\mathbf{u}}} & \text{otherwise,} \end{cases} \tag{A.224b}$$

$$\frac{\partial^2 \mathbf{t}_{1,\sigma}}{\partial \mathbf{V}_{1,i} \partial \mathbf{y}_0} \mathbf{w}_0 = \begin{cases} \tau \alpha_{\sigma i} \begin{bmatrix} \mathbf{0}_{N_{\mathbf{u}} \times N_{\mathbf{v}}} \\ -\left(\mathbf{f}_{\mathbf{v}}^{\mathbf{u}} \left(\underline{\mathbf{v}}^{\mathrm{o}}_{\mathbf{V},\frac{1}{2},\sigma}\right) \mathbf{w}_0^{\mathbf{v}}\right)_{\mathbf{v}} \end{bmatrix} & \text{if } 1 \leq i \leq \sigma, \\ & i \text{ odd} \\ \mathbf{0}_{N \times N_{\mathbf{v}}} & \text{otherwise.} \end{cases} \tag{A.224c}$$

- *For number of stages s even*

For $1 \leq \sigma \leq s$ and σ odd:

$$\frac{\partial^2 \mathbf{t}_{1,\sigma}}{\partial \mathbf{U}_{1,i} \partial \mathbf{y}_0} \mathbf{w}_0 = \begin{cases} \tau \alpha_{\sigma i} \begin{bmatrix} -\left(\mathbf{f}_{\mathbf{u}}^{\mathbf{v}} \left(\underline{\mathbf{u}}^{\mathrm{e}}_{\mathbf{U},0,\sigma}\right) \mathbf{w}_0^{\mathbf{u}}\right)_{\mathbf{u}} \\ \mathbf{0}_{N_{\mathbf{v}} \times N_{\mathbf{u}}} \end{bmatrix} & \text{if } 1 \leq i \leq \sigma, \\ & i \text{ even} \\ \mathbf{0}_{N \times N_{\mathbf{v}}} & \text{otherwise,} \end{cases} \tag{A.224d}$$

$$\frac{\partial^2 \mathbf{t}_{1,\sigma}}{\partial \mathbf{V}_{1,i} \partial \mathbf{y}_0} \mathbf{w}_0 = \begin{cases} \tau \alpha_{\sigma i} \begin{bmatrix} \mathbf{0}_{N_{\mathbf{v}} \times N_{\mathbf{v}}} \\ -\left(\mathbf{f}_{\mathbf{v}}^{\mathbf{u}} \left(\underline{\mathbf{v}}^{\mathrm{e}}_{\mathbf{V},\frac{1}{2},\sigma}\right) \mathbf{w}_0^{\mathbf{v}}\right)_{\mathbf{v}} \end{bmatrix} & \text{if } 1 \leq i \leq \sigma, \\ & i \text{ even} \\ \mathbf{0}_{N \times N_{\mathbf{u}}} & \text{otherwise.} \end{cases} \tag{A.224e}$$

For $1 \leq \sigma \leq s$ and σ even:

$$\frac{\partial^2 \mathbf{t}_{1,\sigma}}{\partial \mathbf{Y}_{1,i} \partial \mathbf{y}_0} \mathbf{w}_0 = \begin{cases} \tau \alpha_{\sigma i} \begin{bmatrix} -\left(\mathbf{f}_{\mathbf{v}}^{\mathbf{u}}\left(\underline{\mathbf{v}}^{\mathrm{e}}_{\mathbf{U},\frac{1}{2},\sigma}\right)\mathbf{w}_0^{\mathbf{v}}\right)_{\mathbf{v}} & \mathbf{0}_{N_{\mathbf{u}}\times N_{\mathbf{u}}} \\ \mathbf{0}_{N_{\mathbf{v}}\times N_{\mathbf{v}}} & \mathbf{0}_{N_{\mathbf{v}}\times N_{\mathbf{u}}} \end{bmatrix} & \text{if } 1 \leq i \leq \sigma, \\ & i \text{ odd} \\ \mathbf{0}_{N\times N_{\mathbf{v}}} & \text{otherwise.} \end{cases} \tag{A.224f}$$

A.4 Exponential Time Differencing

A.4.1 Exponential Runge-Kutta Methods

The time-stepping vector $\mathbf{t} = \mathbf{t}(\overline{\mathbf{y}}; \mathbf{m}, \mathbf{y}_0)$ for ETDRK methods was found in Section 2.4.1 to be

$$\mathbf{t}\left(\overline{\mathbf{y}}; \mathbf{m}, \mathbf{y}_0\right) = \mathbf{T}\overline{\mathbf{y}} - \overline{\mathbf{n}}\left(\overline{\mathbf{y}}, \mathbf{y}_0\right),$$

where

$$\begin{aligned} \overline{\mathbf{y}} &= \begin{bmatrix} \mathbf{Y}_1^\top & \mathbf{y}_1^\top & \mathbf{Y}_2^\top & \mathbf{y}_2^\top & \cdots & \mathbf{Y}_K^\top & \mathbf{y}_K^\top \end{bmatrix}^\top, \\ \overline{\mathbf{n}} &= \begin{bmatrix} \mathbf{N}_1^\top & (e^{\tau_1 \mathbf{L}_0}\mathbf{y}_0)^\top & \mathbf{N}_2^\top & \mathbf{0}_{1\times N} & \cdots & \mathbf{N}_K^\top & \mathbf{0}_{1\times N} \end{bmatrix}^\top, \\ \mathbf{T} &= \begin{bmatrix} \mathbf{I}_{sN} & & & & & & \\ -\mathbf{B}_1^\top & \mathbf{I}_N & & & & & \\ & & \mathbf{I}_{sN} & & & & \\ & -e^{\tau_2 \mathbf{L}_1} & -\mathbf{B}_2^\top & \mathbf{I}_N & & & \\ & & \ddots & \ddots & \ddots & & \\ & & & & & \mathbf{I}_{sN} & \\ & & & & -e^{\tau_K \mathbf{L}_{K-1}} & -\mathbf{B}_K^\top & \mathbf{I}_N \end{bmatrix}, \end{aligned}$$

see (2.60b). We have let $\mathbf{B}_k = \tau_k \begin{bmatrix} b_1(\tau_k \mathbf{L}_{k-1}) & \cdots & b_s(\tau_k \mathbf{L}_{k-1}) \end{bmatrix}^\top$ and

$$\mathbf{Y}_k = \begin{bmatrix} \mathbf{Y}_{k,1} \\ \vdots \\ \mathbf{Y}_{k,s} \end{bmatrix}, \qquad \mathbf{N}_k = \begin{bmatrix} \mathbf{N}_{k,1}, \\ \vdots \\ \mathbf{N}_{k,s} \end{bmatrix},$$

with

$$\mathbf{N}_{k,\sigma} = \mathbf{n}_{k-1}\left(e^{c_\sigma \tau_k \mathbf{L}_{k-1}} \mathbf{y}_{k-1} + \tau_k \sum_{j=1}^{\sigma-1} a_{\sigma j}(\tau_k \mathbf{L}_{k-1}) \mathbf{Y}_{k,j}, t_{k-1} + c_\sigma \tau_k \right).$$

Let $\mathbf{A}_{k-1}^s$ be the $s \times s$ block matrix where the (i,j)th entry is $a_{ij}(\tau_k \mathbf{L}_{k-1})$. To ease the notation further, let

$$t_{k,\sigma} = t_{k-1} + c_\sigma \tau_k, \tag{A.225a}$$

$$\underline{\mathbf{y}}_{k,\sigma} = e^{c_\sigma \tau_k \mathbf{L}_{k-1}(\mathbf{y}_{k-1}, \mathbf{m})} \mathbf{y}_{k-1} + \tau_k \sum_{j=1}^{\sigma-1} a_{ij}(\tau_k \mathbf{L}_{k-1}(\mathbf{y}_{k-1}, \mathbf{m})) \mathbf{y}_{k,j}, \tag{A.225b}$$

and

$$\frac{\partial \mathbf{N}_k}{\partial \mathbf{y}} = \operatorname{diag}\left(\frac{\partial \mathbf{N}_{k,1}}{\partial \mathbf{y}}, \cdots, \frac{\partial \mathbf{N}_{k,s}}{\partial \mathbf{y}} \right). \tag{A.225c}$$

The solution procedure at the kth time-step is represented by

$$\mathbf{t}_k = \begin{bmatrix} \mathbf{Y}_k - \mathbf{N}_k \\ -e^{\tau_k \mathbf{L}_{k-1}} \mathbf{y}_{k-1} - \mathbf{B}_k^\top \mathbf{Y}_k + \mathbf{y}_k \end{bmatrix}. \tag{A.226}$$

We will now take the derivative of $\mathbf{t}$ with respect to each of $\overline{\mathbf{y}}$, $\mathbf{m}$ and $\mathbf{y}_0$ in turn.

- $\dfrac{\partial \mathbf{t}}{\partial \overline{\mathbf{y}}}$

Letting $\widehat{\mathbf{y}}_k = \begin{bmatrix} \mathbf{Y}_k^\top & \mathbf{y}_k^\top \end{bmatrix}^\top$, we break the derivative of $\mathbf{t}$ with respect to $\overline{\mathbf{y}}$ into parts:

$$\frac{\partial \mathbf{t}}{\partial \overline{\mathbf{y}}} = \begin{bmatrix} \frac{\partial \mathbf{t}}{\partial \widehat{\mathbf{y}}_1} & \frac{\partial \mathbf{t}}{\partial \widehat{\mathbf{y}}_2} & \cdots & \frac{\partial \mathbf{t}}{\partial \widehat{\mathbf{y}}_K} \end{bmatrix} = \mathbf{T} - \begin{bmatrix} \frac{\partial \overline{\mathbf{n}}}{\partial \widehat{\mathbf{y}}_1} & \frac{\partial \overline{\mathbf{n}}}{\partial \widehat{\mathbf{y}}_2} & \cdots & \frac{\partial \overline{\mathbf{n}}}{\partial \widehat{\mathbf{y}}_K} \end{bmatrix}. \tag{A.227}$$

The derivative of the kth time-step is

$$\frac{\partial \mathbf{t}_k}{\partial \overline{\mathbf{y}}} = \begin{bmatrix} \frac{\partial \mathbf{t}_k}{\partial \widehat{\mathbf{y}}_1} & \frac{\partial \mathbf{t}_k}{\partial \widehat{\mathbf{y}}_2} & \cdots & \frac{\partial \mathbf{t}_k}{\partial \widehat{\mathbf{y}}_K} \end{bmatrix},$$

with

$$\frac{\partial \mathbf{t}_k}{\partial \widehat{\mathbf{y}}_j} = \begin{bmatrix} \frac{\partial}{\partial \widehat{\mathbf{y}}_j}\left(\mathbf{Y}_k - \mathbf{N}_k\right) \\ \frac{\partial}{\partial \widehat{\mathbf{y}}_j}\left(-e^{\tau_k \mathbf{L}_{k-1}}\mathbf{y}_{k-1} - \mathbf{B}_k^\top \mathbf{Y}_k + \mathbf{y}_k\right) \end{bmatrix}. \tag{A.228}$$

Looking at the terms in (A.228) individually and using the chain rule, we have for $j = k$

$$\frac{\partial \mathbf{Y}_k}{\partial \widehat{\mathbf{y}}_k} = \begin{bmatrix} \mathbf{I}_{sN\times sN} & \mathbf{0}_{sN\times N} \end{bmatrix} \qquad \frac{\partial \mathbf{N}_k}{\partial \widehat{\mathbf{y}}_k} = \tau_k \frac{\partial \mathbf{N}_k}{\partial \mathbf{y}} \begin{bmatrix} \mathbf{A}_{k-1}^s & \mathbf{0}_{sN\times N} \end{bmatrix}$$

$$\frac{\partial \mathbf{y}_k}{\partial \widehat{\mathbf{y}}_k} = \begin{bmatrix} \mathbf{0}_{N\times sN} & \mathbf{I}_{N\times N} \end{bmatrix} \qquad \frac{\partial (\mathbf{B}_k^\top \mathbf{Y}_k)}{\partial \widehat{\mathbf{y}}_k} = \begin{bmatrix} \mathbf{B}_k^\top & \mathbf{0}_{N\times N} \end{bmatrix}$$

and for $j = k - 1$

$$\begin{aligned} \frac{\partial \mathbf{N}_k}{\partial \widehat{\mathbf{y}}_{k-1}} &= \begin{bmatrix} \mathbf{0}_{sN\times sN} & \frac{\partial \mathbf{N}_k}{\partial \mathbf{y}_{k-1}} \end{bmatrix} \\ \frac{\partial (e^{\tau_k \mathbf{L}_{k-1}}\mathbf{y}_{k-1})}{\partial \widehat{\mathbf{y}}_{k-1}} &= \begin{bmatrix} \mathbf{0}_{N\times sN} & \frac{\partial (e^{\tau_k \mathbf{L}_{k-1}}\mathbf{y}_{k-1})}{\partial \mathbf{y}_{k-1}} \end{bmatrix} \\ \frac{\partial (\mathbf{B}_k^\top \mathbf{Y}_k)}{\partial \widehat{\mathbf{y}}_{k-1}} &= \begin{bmatrix} \mathbf{0}_{N\times sN} & \frac{\partial (\mathbf{B}_k^\top \mathbf{Y}_k)}{\partial \mathbf{y}_{k-1}} \end{bmatrix}. \end{aligned}$$

The terms

$$\frac{\partial (\mathbf{B}_k^\top \mathbf{Y}_k)}{\partial \mathbf{y}_{k-1}} = \tau_k \sum_{\sigma=1}^{s} \frac{\partial (b_\sigma(\tau_k \mathbf{L}_{k-1}(\mathbf{y}_{k-1}))\mathbf{Y}_{k,\sigma})}{\partial \mathbf{y}_{k-1}}$$

$$\frac{\partial \mathbf{n}_{k-1,\sigma}\left(\underline{\mathbf{y}}_{k,i}(\mathbf{y}_{k-1}), \mathbf{y}_{k-1}\right)}{\partial \mathbf{y}_{k-1}} = \frac{\partial \mathbf{n}_{k-1,\sigma}}{\partial \mathbf{y}} \frac{\partial \underline{\mathbf{y}}_{k,\sigma}(\mathbf{y}_{k-1})}{\partial \mathbf{y}_{k-1}} + \frac{\partial \mathbf{n}_{k-1,\sigma}(\mathbf{y}_{k-1})}{\partial \mathbf{y}_{k-1}}$$

with

$$\frac{\partial \underline{\mathbf{y}}_{k,\sigma}(\mathbf{y}_{k-1})}{\partial \mathbf{y}_{k-1}} = \frac{\partial e^{c_\sigma \tau_k \mathbf{L}_{k-1}(\mathbf{y}_{k-1})}\mathbf{y}_{k-1}}{\partial \mathbf{y}_{k-1}} + \tau_k \sum_{j=1}^{\sigma-1} \frac{\partial (a_{\sigma j}(\tau_k \mathbf{L}_{k-1}(\mathbf{y}_{k-1}))\mathbf{Y}_{k,j})}{\partial \mathbf{y}_{k-1}},$$

where

$$\frac{\partial \left(e^{c_\sigma \tau_k \mathbf{L}_{k-1}(\mathbf{y}_{k-1})}\mathbf{y}_{k-1}\right)}{\partial \mathbf{y}_{k-1}} = e^{c_\sigma \tau_k \mathbf{L}_{k-1}(\mathbf{y}_{k-1})} + \frac{\partial \left(e^{c_\sigma \tau_k \mathbf{L}_{k-1}(\mathbf{y}_{k-1})}\mathbf{y}_{k-1}^{\text{fixed}}\right)}{\partial \mathbf{y}_{k-1}},$$

are discussed in Section 4.4.2.

Now let

$$\begin{aligned} \mathbf{A}_k &= \mathbf{I}_{sN} - \tau_k \frac{\partial \mathbf{N}_k}{\partial \mathbf{y}} \mathbf{A}_{k-1}^s \qquad \mathbf{C}_k = -\frac{\partial \mathbf{N}_k}{\partial \mathbf{y}_{k-1}}, \\ \mathbf{D}_k &= -\frac{\partial \left(e^{\tau_k \mathbf{L}_{k-1}}\mathbf{y}_{k-1}\right)}{\partial \mathbf{y}_{k-1}} - \frac{\partial \left(\mathbf{B}_k^\top \mathbf{Y}_k\right)}{\partial \mathbf{y}_{k-1}} \end{aligned} \tag{A.229}$$

so that the (k,j)th $(s+1)N \times (s+1)N$ block of $\dfrac{\partial \mathbf{t}}{\partial \overline{\mathbf{y}}}$ is

$$\frac{\partial \mathbf{t}_k}{\partial \widehat{\mathbf{y}}_j} = \begin{cases} \begin{bmatrix} \mathbf{A}_k & \mathbf{0}_{sN\times N} \\ -\mathbf{B}_k^\top & \mathbf{I}_N \end{bmatrix} & \text{if } j = k \\ \begin{bmatrix} \mathbf{0}_{sN\times sN} & \mathbf{C}_k \\ \mathbf{0}_{N\times sN} & \mathbf{D}_k \end{bmatrix} & \text{if } j = k-1 \\ \mathbf{0}_{(s+1)N\times(s+1)N} & \text{otherwise.} \end{cases} \tag{A.230}$$

and we can therefore write

$$\frac{\partial \mathbf{t}}{\partial \overline{\mathbf{y}}} = \begin{bmatrix} \mathbf{A}_1 & & & & & & \\ -\mathbf{B}_1^\top & \mathbf{I}_N & & & & & \\ & \mathbf{C}_2 & \mathbf{A}_2 & & & & \\ & \mathbf{D}_2 & -\mathbf{B}_2^\top & \mathbf{I}_N & & & \\ & & \ddots & \ddots & \ddots & & \\ & & & & \mathbf{C}_K & \mathbf{A}_K & \\ & & & & \mathbf{D}_K & -\mathbf{B}_K^\top & \mathbf{I}_N \end{bmatrix}, \tag{A.231}$$

which is a block lower-triangular matrix representing the linearized forward time-stepping scheme.

- The kth $(s+1)N \times 1$ block of the product $\frac{\partial \mathbf{t}}{\partial \overline{\mathbf{y}}}\overline{\mathbf{w}}$, where $\overline{\mathbf{w}}$ is an arbitrary vector of the form $\overline{\mathbf{w}} = \begin{bmatrix} \mathbf{W}_1^\top & \mathbf{w}_1^\top & \cdots & \mathbf{W}_K^\top & \mathbf{w}_K^\top \end{bmatrix}^\top$ defined analogously to $\overline{\mathbf{y}}$, is

$$\frac{\partial \mathbf{t}_k}{\partial \overline{\mathbf{y}}}\overline{\mathbf{w}} = \begin{bmatrix} \mathbf{A}_k\mathbf{W}_k + \mathbf{C}_k\mathbf{w}_{k-1} \\ -\mathbf{B}_k^\top\mathbf{W}_k + \mathbf{w}_k + \mathbf{D}_k\mathbf{w}_{k-1} \end{bmatrix} \tag{A.232}$$

with $\mathbf{w}_0 = \mathbf{0}_{N\times 1}$. If $\frac{\partial \mathbf{t}_{k,\sigma}}{\partial \overline{\mathbf{y}}}\overline{\mathbf{w}}$ is the σth $N \times 1$ block of $\frac{\partial \mathbf{t}_k}{\partial \overline{\mathbf{y}}}\overline{\mathbf{w}}$, with $1 \leq \sigma \leq s+1$,

we have

$$\frac{\partial \mathbf{t}_{k,\sigma}}{\partial \overline{\mathbf{y}}}\overline{\mathbf{w}} = \begin{cases} \mathbf{W}_{k,\sigma} - \dfrac{\partial \mathbf{n}_{k-1,\sigma}}{\partial \mathbf{y}}\underline{\mathbf{w}}_{k,\sigma} - \left\{ \dfrac{\partial \mathbf{n}_{k-1,\sigma}}{\partial \mathbf{y}_{k-1}}\mathbf{w}_{k-1} \right. & \\ \quad + \dfrac{\partial \mathbf{n}_{k-1,\sigma}}{\partial \mathbf{y}} \left(\dfrac{\partial \left(e^{c_\sigma \tau_k \mathbf{L}_{k-1}(\mathbf{y}_{k-1})}\mathbf{y}_{k-1}^{\text{fixed}}\right)}{\partial \mathbf{y}_{k-1}}\mathbf{w}_{k-1} + \right. & \text{if } \sigma \leq s \\ \quad \left. \left. + \ \tau_k \displaystyle\sum_{i=1}^{\sigma-1} \dfrac{\partial (a_{\sigma i}(\tau_k \mathbf{L}_{k-1}(\mathbf{y}_{k-1}))\mathbf{Y}_{k,i})}{\partial \mathbf{y}_{k-1}}\mathbf{w}_{k-1} \right) \right\} & \\ \mathbf{w}_k - e^{\tau_k \mathbf{L}_{k-1}}\mathbf{w}_{k-1} - \tau_k \displaystyle\sum_{\sigma=1}^{s} b_\sigma(\tau_k \mathbf{L}_{k-1})\mathbf{W}_{k,\sigma} + & \\ \quad - \left\{ \dfrac{\partial \left(e^{\tau_k \mathbf{L}_{k-1}(\mathbf{y}_{k-1})}\mathbf{y}_{k-1}^{\text{fixed}}\right)}{\partial \mathbf{y}_{k-1}}\mathbf{w}_{k-1} + \right. & \text{if } \sigma = s+1, \\ \quad \left. \tau_k \displaystyle\sum_{i=1}^{s} \dfrac{\partial (b_i(\tau_k \mathbf{L}_{k-1}(\mathbf{y}_{k-1}))\mathbf{W}_{k,i})}{\partial \mathbf{y}_{k-1}}\mathbf{w}_{k-1} \right\} & \end{cases} \tag{A.233}$$

where $\underline{\mathbf{w}}_{k,\sigma} = e^{c_\sigma \tau_k \mathbf{L}_{k-1}}\mathbf{w}_{k-1} + \tau_k \sum_{i=1}^{\sigma-1} a_{\sigma i}(\tau_k \mathbf{L}_{k-1})\mathbf{W}_{k,i}$.

- The transpose of $\dfrac{\partial \mathbf{t}}{\partial \overline{\mathbf{y}}}$ is upper block-triangular:

$$\frac{\partial \mathbf{t}}{\partial \overline{\mathbf{y}}}^\top = \begin{bmatrix} \mathbf{A}_1^\top & -\mathbf{B}_1 & & & & & \\ & \mathbf{I}_N & \mathbf{C}_2^\top & \mathbf{D}_2^\top & & & \\ & & \mathbf{A}_2^\top & -\mathbf{B}_2 & & & \\ & & & \ddots & \ddots & \ddots & \\ & & & & \mathbf{I}_N & \mathbf{C}_K^\top & \mathbf{D}_K^\top \\ & & & & & \mathbf{A}_K^\top & -\mathbf{B}_K \\ & & & & & & \mathbf{I}_N \end{bmatrix}. \tag{A.234}$$

The jth $(s+1)N \times 1$ block of the product of $\dfrac{\partial \mathbf{t}}{\partial \overline{\mathbf{y}}}^\top$ with some arbitrary $\overline{\mathbf{w}}$ therefore is

$$\frac{\partial \mathbf{t}}{\partial \widehat{\mathbf{y}}_j}^\top \overline{\mathbf{w}} = \begin{bmatrix} \mathbf{A}_j^\top \mathbf{W}_j - \mathbf{B}_j \mathbf{w}_j \\ \mathbf{C}_{j+1}^\top \mathbf{W}_{j+1} + \mathbf{w}_j + \mathbf{D}_{j+1}^\top \mathbf{w}_{j+1} \end{bmatrix}, \tag{A.235}$$

with $\mathbf{W}_{K+1,\sigma} = \mathbf{w}_{K+1} = \mathbf{0}_{N\times 1}$. The σth $N \times 1$ block $\frac{\partial \mathbf{t}}{\partial \widehat{\mathbf{y}}_{j,\sigma}}^{\top} \overline{\mathbf{w}}$ of $\frac{\partial \mathbf{t}}{\partial \widehat{\mathbf{y}}_{j}}^{\top} \overline{\mathbf{w}}$, with $1 \leq \sigma \leq s+1$, is

$$
\frac{\partial \mathbf{t}}{\partial \widehat{\mathbf{y}}_{j,\sigma}}^{\top} \overline{\mathbf{w}} =
$$

$$
= \begin{cases}
\mathbf{W}_{k,\sigma} - \tau_k b_\sigma \left(\tau_k \mathbf{L}_{k-1}\right)^\top \mathbf{w}_k + \\
\qquad - \tau_k \sum_{j=\sigma+1}^{s} a_{j\sigma} \left(\tau_k \mathbf{L}_{k-1}^\top\right) \left(\frac{\partial \mathbf{n}_{k-1,j}}{\partial \mathbf{y}}\right)^\top \mathbf{W}_{k,j} & \text{if } \sigma \leq s \\
\mathbf{w}_k - e^{\tau_{k+1}\mathbf{L}_k^\top} \mathbf{w}_{k+1} - \sum_{\sigma=1}^{s} e^{c_\sigma \tau_{k+1}\mathbf{L}_k^\top} \left(\frac{\partial \mathbf{n}_{k,\sigma}}{\partial \mathbf{y}}\right)^\top \mathbf{W}_{k+1,\sigma} + \\
- \left\{ \sum_{i=1}^{s} \left[\frac{\partial \mathbf{n}_{k,i}}{\partial \mathbf{y}_k}^\top + \left(\frac{\partial e^{c_i \tau_{k+1}\mathbf{L}_k(\mathbf{y}_k)} \mathbf{y}_k^{\text{fixed}}}{\partial \mathbf{y}_k}^\top + \right.\right.\right. \\
+ \left.\left.\left. \tau_{k+1} \sum_{j=1}^{i-1} \frac{\partial (a_{ij}(\tau_{k+1}\mathbf{L}_k(\mathbf{y}_k))\mathbf{Y}_{k+1,j})}{\partial \mathbf{y}_k}^\top \right) \frac{\partial \mathbf{n}_{k,i}}{\partial \mathbf{y}}^\top \right] \mathbf{W}_{k+1,i} + & \text{if } \sigma = s+1. \\
+ \left(\frac{\partial (e^{\tau_{k+1}\mathbf{L}_k(\mathbf{y}_k)} \mathbf{y}_k^{\text{fixed}})}{\partial \mathbf{y}_k}^\top + \right. \\
\left. + \tau_{k+1} \sum_{i=1}^{s} \frac{\partial (b_i(\tau_{k+1}\mathbf{L}_k(\mathbf{y}_k))\mathbf{Y}_{k+1,i})}{\partial \mathbf{y}_k}^\top \right) \mathbf{w}_{k+1} \Bigg\},
\end{cases}
$$

- $\dfrac{\partial \mathbf{t}}{\partial \mathbf{m}}$

  We now turn our attention to computing the derivative of $\mathbf{t}$ with respect to $\mathbf{m}$. Consider the derivative of the kth time-step:

$$
\frac{\partial \mathbf{t}_k}{\partial \mathbf{m}} = \begin{bmatrix} -\dfrac{\partial \mathbf{N}_k(\overline{\mathbf{y}}, \mathbf{m})}{\partial \mathbf{m}} \\ -\dfrac{\partial}{\partial \mathbf{m}} \left(e^{\tau_k \mathbf{L}_{k-1}} \mathbf{y}_{k-1} - \mathbf{B}_k^\top \mathbf{Y}_k\right) \end{bmatrix}. \tag{A.236}
$$

  The individual terms are

$$
\frac{\partial (\mathbf{B}_k^\top \mathbf{Y}_k)}{\partial \mathbf{m}} = \tau_k \sum_{\sigma=1}^{s} \frac{\partial (b_i(\tau_k \mathbf{L}_{k-1}(\mathbf{m}))\mathbf{Y}_{k,\sigma})}{\partial \mathbf{m}}, \tag{A.237a}
$$

and

$$\frac{\partial \mathbf{n}_{k-1,\sigma}\left(\underline{\mathbf{y}}_{k-1,\sigma}, \mathbf{m}\right)}{\partial \mathbf{m}} = \frac{\partial \mathbf{n}_{k,i}(\mathbf{m})}{\partial \mathbf{y}} \frac{\partial \underline{\mathbf{y}}_{k,\sigma}}{\partial \mathbf{m}} + \frac{\partial \mathbf{n}_{k,\sigma}(\mathbf{m})}{\partial \mathbf{m}} \tag{A.237b}$$

with

$$\frac{\partial \underline{\mathbf{y}}_{k,\sigma}}{\partial \mathbf{m}} = \frac{\partial \left(e^{c_i \tau_k \mathbf{L}_{k-1}(\mathbf{m})} \mathbf{y}_{k-1}\right)}{\partial \mathbf{m}} + \tau_k \sum_{j=1}^{\sigma-1} \frac{\partial (a_{\sigma j}(\tau_k \mathbf{L}_{k-1}(\mathbf{m})) \mathbf{Y}_{k,j})}{\partial \mathbf{m}}.$$

If the linear terms $\tau_k \mathbf{L}_{k-1}$ are independent of $\mathbf{m}$ then the only term that depends on $\mathbf{m}$ is $\mathbf{N}_k$, so trivially we have

$$\frac{\partial \mathbf{t}_k}{\partial \mathbf{m}} = \begin{bmatrix} -\dfrac{\partial \mathbf{N}_k(\mathbf{m})}{\partial \mathbf{m}} \\ \mathbf{0}_{N \times N_{\mathbf{m}}} \end{bmatrix}. \tag{A.238}$$

See Section 4.4.2 if $\tau_k \mathbf{L}_{k-1}$ depends on $\mathbf{m}$.

- The product of $\dfrac{\partial \mathbf{t}_k}{\partial \mathbf{m}}$ with some vector $\mathbf{w}_{\mathbf{m}}$ of length $N_{\mathbf{m}}$ is

$$\frac{\partial \mathbf{t}_k}{\partial \mathbf{m}} \mathbf{w}_{\mathbf{m}} = \begin{bmatrix} -\dfrac{\partial \mathbf{N}_k(\overline{\mathbf{y}}, \mathbf{m})}{\partial \mathbf{m}} \mathbf{w}_{\mathbf{m}} \\ -\dfrac{\partial \left(e^{\tau_k \mathbf{L}_{k-1}} \mathbf{y}_{k-1} - \mathbf{B}_k^\top \mathbf{Y}_k\right)}{\partial \mathbf{m}} \mathbf{w}_{\mathbf{m}} \end{bmatrix}, \tag{A.239}$$

  with the bottom block ignored if $\mathbf{L}_k$ does not depend on $\mathbf{m}$.

- The product $\dfrac{\partial \mathbf{t}_k}{\partial \mathbf{m}}^\top \widehat{\mathbf{w}}_k$, with $\widehat{\mathbf{w}}_k = \begin{bmatrix} \mathbf{W}_k^\top & \mathbf{w}_k^\top \end{bmatrix}^\top$ an arbitrary vector of length $(s+1)N$ defined analogously to $\widehat{\mathbf{y}}_k$ (replacing $\mathbf{Y}_k$ by $\mathbf{W}_k$), is

$$\begin{aligned} \frac{\partial \mathbf{t}_k}{\partial \mathbf{m}}^\top \widehat{\mathbf{w}}_k = -\sum_{\sigma=1}^{s} \frac{\partial \mathbf{n}_{k-1,\sigma}}{\partial \mathbf{m}}^\top \mathbf{W}_{k,\sigma} - \Bigg\{ & \frac{\partial \left(e^{\tau_k \mathbf{L}_{k-1}(\mathbf{m})} \mathbf{y}_{k-1}\right)}{\partial \mathbf{m}}^\top \mathbf{w}_k + \\ & + \sum_{\sigma=1}^{s} \left(\frac{\partial \underline{\mathbf{y}}_{k,\sigma}}{\partial \mathbf{m}}^\top \mathbf{v}_{k,\sigma} + \tau_k \frac{\partial (b_\sigma(\tau_k \mathbf{L}_{k-1}(\mathbf{m})) \mathbf{y}_{k,\sigma})}{\partial \mathbf{m}}^\top \mathbf{w}_k \right) \Bigg\}, \end{aligned}$$

where $\mathbf{v}_{k,\sigma} = \dfrac{\partial \mathbf{n}_{k-1,\sigma}}{\partial \mathbf{y}}^{\top} \mathbf{W}_{k,\sigma}$ and

$$\frac{\partial \underline{\mathbf{y}}_{k,\sigma}}{\partial \mathbf{m}}^{\top} \mathbf{v}_{k,\sigma} = \frac{\partial \left(e^{c_i \tau_k \mathbf{L}_{k-1}(\mathbf{m})} \mathbf{y}_{k-1}\right)}{\partial \mathbf{m}}^{\top} \mathbf{v}_{k,\sigma} + \tau_k \sum_{j=1}^{i-1} \frac{\partial (a_{\sigma j}(\tau_k \mathbf{L}_{k-1}(\mathbf{m})) \mathbf{Y}_{k,\sigma})}{\partial \mathbf{m}}^{\top} \mathbf{v}_{k,\sigma}.$$

The terms in braces in (A.240) are ignored if $\mathbf{L}_{k-1}$ is independent of $\mathbf{m}$.

- $\dfrac{\partial \mathbf{t}}{\partial \mathbf{y}_0}$

It is clear that the derivative of $\mathbf{t}_k$ with respect to $\mathbf{y}_0$ is $\mathbf{0}_{sN\times N}$ for all $k > 1$ since there is no explicit dependence on $\mathbf{y}_0$ in later time-steps.

For $k = 1$ we have

$$\frac{\partial \mathbf{t}_1}{\partial \mathbf{y}_0} = \begin{bmatrix} -\dfrac{\partial \mathbf{N}_1(\overline{\mathbf{y}})}{\partial \mathbf{y}_0} \\ -\dfrac{\partial}{\partial \mathbf{y}_0}\left(e^{\tau_1 \mathbf{L}_0} \mathbf{y}_0 - \mathbf{B}_1^{\top} \mathbf{Y}_1\right) \end{bmatrix}. \tag{A.240}$$

The individual terms are

$$\frac{\partial (\mathbf{B}_1^{\top} \mathbf{Y}_1)}{\partial \mathbf{y}_0} = \tau_1 \sum_{\sigma=1}^{s} \frac{\partial (b_\sigma(\tau_1 \mathbf{L}_0(\mathbf{y}_0)) \mathbf{Y}_{1,\sigma})}{\partial \mathbf{y}_0}$$

$$\frac{\partial \mathbf{n}_{0,\sigma}\left(\underline{\mathbf{y}}_{1,i}(\mathbf{y}_0), \mathbf{y}_0\right)}{\partial \mathbf{y}_0} = \frac{\partial \mathbf{n}_{0,\sigma}}{\partial \mathbf{y}} \frac{\partial \underline{\mathbf{y}}_{1,\sigma}(\mathbf{y}_0)}{\partial \mathbf{y}_0} + \frac{\partial \mathbf{n}_{0,\sigma}(\mathbf{y}_0)}{\partial \mathbf{y}_0}$$

with

$$\frac{\partial \underline{\mathbf{y}}_{1,\sigma}(\mathbf{y}_0)}{\partial \mathbf{y}_0} = \frac{\partial e^{c_\sigma \tau_1 \mathbf{L}_0(\mathbf{y}_0)} \mathbf{y}_0}{\partial \mathbf{y}_0} + \tau_1 \sum_{i=1}^{\sigma-1} \frac{\partial (a_{\sigma i}(\tau_1 \mathbf{L}_0(\mathbf{y}_0)) \mathbf{Y}_{1,i})}{\partial \mathbf{y}_0},$$

where

$$\frac{\partial \left(e^{c_\sigma \tau_1 \mathbf{L}_0(\mathbf{y}_0)} \mathbf{y}_0\right)}{\partial \mathbf{y}_0} = e^{c_\sigma \tau_k \mathbf{L}_{k-1}(\mathbf{y}_0)} + \frac{\partial \left(e^{c_\sigma \tau_1 \mathbf{L}_0(\mathbf{y}_0)} \mathbf{y}_0^{\text{fixed}}\right)}{\partial \mathbf{y}_0}.$$

- The product $\dfrac{\partial \mathbf{t}_k}{\partial \mathbf{y}_0} \mathbf{w}_0$, where $\mathbf{w}_0$ is an arbitrary vector of length N, is $\mathbf{0}_{N\times 1}$ if $k > 1$. If $k = 1$,

$$\frac{\partial \mathbf{t}_k}{\partial \mathbf{y}_0} \mathbf{w}_0 = \begin{bmatrix} \mathbf{C}_1 \mathbf{w}_0 \\ \mathbf{D}_1 \mathbf{w}_0 \end{bmatrix}. \tag{A.241}$$

If $\dfrac{\partial \mathbf{t}_{1,\sigma}}{\partial \mathbf{y}_0}\mathbf{w}_0$ is the σth $N \times 1$ block of $\dfrac{\partial \mathbf{t}_1}{\partial \mathbf{y}_0}\mathbf{w}_0$, with $1 \leq \sigma \leq s+1$, we have

$$\frac{\partial \mathbf{t}_{k,\sigma}}{\partial \mathbf{y}_0}\mathbf{w}_0 = \begin{cases} -\dfrac{\partial \mathbf{n}_{0,\sigma}}{\partial \mathbf{y}} e^{c_\sigma \tau_k \mathbf{L}_{k-1}}\mathbf{w}_0 - \left\{ \dfrac{\partial \mathbf{n}_{0,\sigma}}{\partial \mathbf{y}_0}\mathbf{w}_0 \right. & \\ \quad + \dfrac{\partial \mathbf{n}_{0,\sigma}}{\partial \mathbf{y}} \left(\dfrac{\partial \left(e^{c_\sigma \tau_1 \mathbf{L}_0(\mathbf{y}_0)} \mathbf{y}_0^{\text{fixed}}\right)}{\partial \mathbf{y}_0}\mathbf{w}_0 + \right. & \text{if } \sigma \leq s \\ \quad \left. \left. + \tau_1 \displaystyle\sum_{i=1}^{\sigma-1} \dfrac{\partial (a_{\sigma i}(\tau_1 \mathbf{L}_0(\mathbf{y}_0))\mathbf{Y}_{1,i})}{\partial \mathbf{y}_0}\mathbf{w}_0 \right) \right\} & \\ -e^{\tau_1 \mathbf{L}_0}\mathbf{w}_0 - \left\{ \dfrac{\partial \left(e^{\tau_1 \mathbf{L}_0(\mathbf{y}_0)} \mathbf{y}_0^{\text{fixed}}\right)}{\partial \mathbf{y}_0}\mathbf{w}_0 + \right. & \\ \quad \left. \tau_1 \displaystyle\sum_{i=1}^{s} \dfrac{\partial (b_i(\tau_1 \mathbf{L}_0(\mathbf{y}_0))\mathbf{W}_{1,i})}{\partial \mathbf{y}_0}\mathbf{w}_0 \right\} & \text{if } \sigma = s+1. \end{cases} \tag{A.242}$$

- The transpose of $\dfrac{\partial \mathbf{t}_1}{\partial \mathbf{y}_0}$ with some arbitrary vector $\widehat{\mathbf{w}}_1$ of length $(s+1)N$ is

$$\frac{\partial \mathbf{t}_k}{\partial \mathbf{y}_0}^\top \widehat{\mathbf{w}}_1 = \left[\mathbf{C}_1^\top \mathbf{W}_1 + \mathbf{D}_1^\top \mathbf{w}_1 \right]. \tag{A.243}$$

which gives

$$\begin{aligned} \frac{\partial \mathbf{t}}{\partial \mathbf{y}_0}^\top \mathbf{w}_0 = -e^{\tau_1 \mathbf{L}_0^\top}\mathbf{w}_1 - \Bigg\{ \sum_{i=1}^{s} \Bigg[\frac{\partial \mathbf{n}_{0,i}}{\partial \mathbf{y}_0}^\top + \Bigg(\frac{\partial e^{c_i \tau_1 \mathbf{L}_0(\mathbf{y}_0)} \mathbf{y}_0^{\text{fixed}}}{\partial \mathbf{y}_0}^\top + \\ + \tau_1 \sum_{j=1}^{i-1} \frac{\partial (a_{ij}(\tau_1 \mathbf{L}_0(\mathbf{y}_0))\mathbf{Y}_{1,j})}{\partial \mathbf{y}_0}^\top \Bigg) \frac{\partial \mathbf{n}_{0,i}}{\partial \mathbf{y}}^\top \Bigg] \mathbf{W}_{1,i} + \\ + \left(\frac{\partial \left(e^{\tau_1 \mathbf{L}_0(\mathbf{y}_0)} \mathbf{y}_0^{\text{fixed}}\right)}{\partial \mathbf{y}_0}^\top + \tau_1 \sum_{i=1}^{s} \frac{\partial (b_i(\tau_1 \mathbf{L}_0(\mathbf{y}_0))\mathbf{Y}_{1,i})}{\partial \mathbf{y}_0}^\top \right) \mathbf{w}_1 \Bigg\}. \end{aligned}$$

Appendix B

Derivation of the Hessian

In this appendix we use the discrete adjoint method to find the expression (3.15) for the product of the Hessian of the data misfit function M multiplied by some arbitrary vector $\mathbf{w_p} = \begin{bmatrix} \mathbf{w_m}^\top & \mathbf{w_{y_0}}^\top & \mathbf{w_s}^\top \end{bmatrix}^\top$ of length $N_\mathbf{m} + N_{\mathbf{y}_0} + N_\mathbf{s}$.

We use the following notation: if $\mathbf{x}$, $\mathbf{y}$, $\mathbf{z}$ and $\mathbf{w}$ are some vector quantities of appropriate lengths, we let

$$\frac{\partial^2 \mathbf{x}}{\partial \mathbf{y} \partial \mathbf{z}} \mathbf{w} = \frac{\partial}{\partial \mathbf{y}} \left(\frac{\partial \mathbf{x}}{\partial \mathbf{z}} \mathbf{w} \bigg|_\mathbf{w} \right),$$

where the $\cdot|_\mathbf{w}$ on the right-hand side means that $\mathbf{w}$ is taken to be fixed when performing the differentiation with respect to $\mathbf{y}$. Note that $\dfrac{\partial^2 \mathbf{x}}{\partial \mathbf{y} \partial \mathbf{z}}$ is actually a three-dimensional tensor and its product with a vector is ambiguously defined, so it is important to keep our convention in mind.

From Section 3.2 we know that

$$\nabla_\mathbf{p} \mathrm{M} = \frac{\partial \mathbf{d}}{\partial \mathbf{p}}^\top \nabla_\mathbf{d} \mathrm{M} = \mathbf{J}^\top \nabla_\mathbf{d} \mathrm{M},$$

so by the product rule

$$\begin{aligned} \frac{\partial^2 \mathrm{M}}{\partial \mathbf{p} \partial \mathbf{p}} \mathbf{w_p} = \nabla_\mathbf{p} \left(\nabla_\mathbf{p} \mathrm{M}^\top \mathbf{w_p} \right) &= \nabla_\mathbf{p} \left(\nabla_\mathbf{d} \mathrm{M}^\top \mathbf{J}\, \mathbf{w_p} \right) \\ &= \overset{①}{\frac{\partial^2 \mathrm{M}}{\partial \mathbf{p} \partial \mathbf{d}}}^\top \mathbf{J}\, \mathbf{w_p} + \overset{②}{\left(\frac{\partial}{\partial \mathbf{p}} \mathbf{J}\, \mathbf{w_p} \right)}^\top \nabla_\mathbf{d} \mathrm{M}\,. \end{aligned} \tag{B.1}$$

By the chain rule, 1 is just

$$\frac{\partial^2 \mathrm{M}}{\partial \mathbf{m} \partial \mathbf{d}}^{\top} \mathbf{J}\,\mathbf{w_p} = \frac{\partial \mathbf{d}}{\partial \mathbf{m}}^{\top} \frac{\partial^2 \mathrm{M}}{\partial \mathbf{d} \partial \mathbf{d}} \mathbf{J}\,\mathbf{w_p} = \mathbf{J}^{\top} \frac{\partial^2 \mathrm{M}}{\partial \mathbf{d} \partial \mathbf{d}} \mathbf{J}\,\mathbf{w_p}. \tag{B.2}$$

Letting

$$\mathbf{x} = \mathbf{x}(\mathbf{p}) = \frac{\partial \mathbf{y}}{\partial \mathbf{p}} \mathbf{w_p} = \frac{\partial \mathbf{y}}{\partial \mathbf{m}} \mathbf{w_m} + \frac{\partial \mathbf{y}}{\partial \mathbf{s}} \mathbf{w_s} + \frac{\partial \mathbf{y}}{\partial \mathbf{y}_0} \mathbf{w_{y_0}} \tag{B.3}$$

for brevity, we have $\mathbf{J}\,\mathbf{w_p} = \frac{\partial \mathbf{d}}{\partial \mathbf{p}} \mathbf{w_p} = \frac{\partial \mathbf{d}}{\partial \mathbf{y}} \mathbf{x}$. 2 becomes

$$\begin{aligned}\left(\frac{\partial}{\partial \mathbf{p}} \mathbf{J}\,\mathbf{w_p}\right)^{\top} \nabla_{\mathbf{d}} \mathrm{M} &= \left(\frac{\partial}{\partial \mathbf{p}} \left(\frac{\partial \mathbf{d}}{\partial \mathbf{y}} \mathbf{x}\right)\right)^{\top} \nabla_{\mathbf{d}} \mathrm{M} \\ &= \frac{\partial \mathbf{y}}{\partial \mathbf{p}}^{\top} \overset{3}{\left(\frac{\partial^2 \mathbf{d}}{\partial \mathbf{y} \partial \mathbf{y}} \mathbf{x}\right)}^{\top} \nabla_{\mathbf{d}} \mathrm{M} + \frac{\partial \mathbf{x}}{\partial \mathbf{p}}^{\top} \overset{4}{\frac{\partial \mathbf{d}}{\partial \mathbf{y}}}^{\top} \nabla_{\mathbf{d}} \mathrm{M}\,.\end{aligned} \tag{B.4}$$

To compute 4, we need to consider each term of

$$\frac{\partial^2 \mathbf{y}}{\partial \mathbf{p} \partial \mathbf{p}} \mathbf{w_p} = \left[\frac{\partial^2 \mathbf{y}}{\partial \mathbf{m} \partial \mathbf{p}} \mathbf{w_p} \quad \frac{\partial^2 \mathbf{y}}{\partial \mathbf{y}_0 \partial \mathbf{p}} \mathbf{w_p} \quad \frac{\partial^2 \mathbf{y}}{\partial \mathbf{s} \partial \mathbf{p}} \mathbf{w_p}\right]. \tag{B.5}$$

We apply the following labels:

$$\frac{\partial^2 \mathbf{y}}{\partial \mathbf{m} \partial \mathbf{p}} \mathbf{w_p} = \left[\overset{\mathrm{A}}{\frac{\partial^2 \mathbf{y}}{\partial \mathbf{m} \partial \mathbf{m}} \mathbf{w_m}} + \overset{\mathrm{B}}{\frac{\partial^2 \mathbf{y}}{\partial \mathbf{m} \partial \mathbf{y}_0} \mathbf{w_{y_0}}} + \overset{\mathrm{C}}{\frac{\partial^2 \mathbf{y}}{\partial \mathbf{m} \partial \mathbf{s}} \mathbf{w_s}}\right], \tag{B.6a}$$

$$\frac{\partial^2 \mathbf{y}}{\partial \mathbf{y}_0 \partial \mathbf{p}} \mathbf{w_p} = \left[\overset{\mathrm{D}}{\frac{\partial^2 \mathbf{y}}{\partial \mathbf{y}_0 \partial \mathbf{m}} \mathbf{w_m}} + \overset{\mathrm{E}}{\frac{\partial^2 \mathbf{y}}{\partial \mathbf{y}_0 \partial \mathbf{y}_0} \mathbf{w_{y_0}}} + \overset{\mathrm{F}}{\frac{\partial^2 \mathbf{y}}{\partial \mathbf{y}_0 \partial \mathbf{s}} \mathbf{w_s}}\right], \tag{B.6b}$$

$$\frac{\partial^2 \mathbf{y}}{\partial \mathbf{s} \partial \mathbf{p}} \mathbf{w_p} = \left[\overset{\mathrm{G}}{\frac{\partial^2 \mathbf{y}}{\partial \mathbf{s} \partial \mathbf{m}} \mathbf{w_m}} + \overset{\mathrm{H}}{\frac{\partial^2 \mathbf{y}}{\partial \mathbf{s} \partial \mathbf{y}_0} \mathbf{w_{y_0}}} + \overset{\mathrm{I}}{\frac{\partial^2 \mathbf{y}}{\partial \mathbf{s} \partial \mathbf{s}} \mathbf{w_s}}\right]. \tag{B.6c}$$

Each of these terms will now be considered in turn.

- Ⓐ, Ⓓ, Ⓖ: Multiply (3.3) by $\mathbf{w_m}$,

$$\frac{\partial \mathbf{t}}{\partial \mathbf{m}}\mathbf{w_m} + \frac{\partial \mathbf{t}}{\partial \mathbf{y}}\frac{\partial \mathbf{y}}{\partial \mathbf{m}}\mathbf{w_m} = \mathbf{0}_{N\times 1}, \tag{B.7}$$

with $\mathbf{t} = \mathbf{t}\left(\mathbf{y}\left(\mathbf{p}\right), \mathbf{m}, \mathbf{y}_0\right)$, and let $\mathbf{x_m} = \dfrac{\partial \mathbf{y}}{\partial \mathbf{m}}\mathbf{w_m}$, $\mathbf{z_m} = \dfrac{\partial \mathbf{t}}{\partial \mathbf{m}}\mathbf{w_m}$ and $\mathbf{v_m} = \dfrac{\partial \mathbf{t}}{\partial \mathbf{y}}\mathbf{x_m}$.

  - Ⓐ: Differentiate (B.7) with respect to $\mathbf{m}$:

$$\frac{\partial \mathbf{z_m}}{\partial \mathbf{m}} + \frac{\partial \mathbf{z_m}}{\partial \mathbf{y}}\frac{\partial \mathbf{y}}{\partial \mathbf{m}} + \frac{\partial \mathbf{v_m}}{\partial \mathbf{m}} + \frac{\partial \mathbf{v_m}}{\partial \mathbf{y}}\frac{\partial \mathbf{y}}{\partial \mathbf{m}} + \frac{\partial \mathbf{t}}{\partial \mathbf{y}}\frac{\partial \mathbf{x_m}}{\partial \mathbf{m}} = \mathbf{0}_{N\times N_\mathbf{m}}.$$

    Therefore

$$\text{Ⓐ} = \frac{\partial \mathbf{x_m}}{\partial \mathbf{m}} = -\left(\frac{\partial \mathbf{t}}{\partial \mathbf{y}}\right)^{-1}\left(\frac{\partial \mathbf{z_m}}{\partial \mathbf{m}} + \frac{\partial \mathbf{z_m}}{\partial \mathbf{y}}\frac{\partial \mathbf{y}}{\partial \mathbf{m}} + \frac{\partial \mathbf{v_m}}{\partial \mathbf{m}} + \frac{\partial \mathbf{v_m}}{\partial \mathbf{y}}\frac{\partial \mathbf{y}}{\partial \mathbf{m}}\right).$$

  - Ⓓ: Differentiate (B.7) with respect to $\mathbf{y}_0$:

$$\frac{\partial \mathbf{z_m}}{\partial \mathbf{y}_0} + \frac{\partial \mathbf{z_m}}{\partial \mathbf{y}}\frac{\partial \mathbf{y}}{\partial \mathbf{y}_0} + \frac{\partial \mathbf{v_m}}{\partial \mathbf{y}_0} + \frac{\partial \mathbf{v_m}}{\partial \mathbf{y}}\frac{\partial \mathbf{y}}{\partial \mathbf{y}_0} + \frac{\partial \mathbf{t}}{\partial \mathbf{y}}\frac{\partial \mathbf{x_m}}{\partial \mathbf{y}_0} = \mathbf{0}_{N\times N}.$$

    Therefore

$$\text{Ⓓ} = \frac{\partial \mathbf{x_m}}{\partial \mathbf{y}_0} = -\left(\frac{\partial \mathbf{t}}{\partial \mathbf{y}}\right)^{-1}\left(\frac{\partial \mathbf{z_m}}{\partial \mathbf{y}_0} + \frac{\partial \mathbf{z_m}}{\partial \mathbf{y}}\frac{\partial \mathbf{y}}{\partial \mathbf{y}_0} + \frac{\partial \mathbf{v_m}}{\partial \mathbf{y}_0} + \frac{\partial \mathbf{v_m}}{\partial \mathbf{y}}\frac{\partial \mathbf{y}}{\partial \mathbf{y}_0}\right).$$

  - Ⓖ: Differentiate (B.7) with respect to $\mathbf{s}$:

$$\frac{\partial \mathbf{z_m}}{\partial \mathbf{y}}\frac{\partial \mathbf{y}}{\partial \mathbf{s}} + \frac{\partial \mathbf{v_m}}{\partial \mathbf{y}}\frac{\partial \mathbf{y}}{\partial \mathbf{s}} + \frac{\partial \mathbf{t}}{\partial \mathbf{y}}\frac{\partial \mathbf{x_m}}{\partial \mathbf{s}} = \mathbf{0}_{N\times N_\mathbf{s}}.$$

    Therefore

$$\text{Ⓖ} = \frac{\partial \mathbf{x_m}}{\partial \mathbf{s}} = -\left(\frac{\partial \mathbf{t}}{\partial \mathbf{y}}\right)^{-1}\left(\frac{\partial \mathbf{z_m}}{\partial \mathbf{y}}\frac{\partial \mathbf{y}}{\partial \mathbf{s}} + \frac{\partial \mathbf{v_m}}{\partial \mathbf{y}}\frac{\partial \mathbf{y}}{\partial \mathbf{s}}\right).$$

- Ⓑ, Ⓔ, Ⓗ: Multiply (3.5) by $\mathbf{w}_{\mathbf{y}_0}$,

$$\frac{\partial \mathbf{t}}{\partial \mathbf{y}_0}\mathbf{w}_{\mathbf{y}_0} + \frac{\partial \mathbf{t}}{\partial \mathbf{y}}\frac{\partial \mathbf{y}}{\partial \mathbf{y}_0}\mathbf{w}_{\mathbf{y}_0} = \mathbf{0}_{N\times N}, \tag{B.8}$$

with $\mathbf{t} = \mathbf{t}\left(\mathbf{y}\left(\mathbf{p}\right), \mathbf{m}, \mathbf{y}_0\right)$, and let $\mathbf{x}_{\mathbf{y}_0} = \dfrac{\partial \mathbf{y}}{\partial \mathbf{y}_0}\mathbf{w}_{\mathbf{y}_0}$, $\mathbf{z}_{\mathbf{y}_0} = \dfrac{\partial \mathbf{t}}{\partial \mathbf{y}_0}\mathbf{w}_{\mathbf{y}_0}$ and $\mathbf{v}_{\mathbf{y}_0} = \dfrac{\partial \mathbf{t}}{\partial \mathbf{y}}\mathbf{x}_{\mathbf{y}_0}$.

- Ⓑ: Differentiate (B.8) with respect to $\mathbf{m}$:

$$\frac{\partial \mathbf{z}_{\mathbf{y}_0}}{\partial \mathbf{m}} + \frac{\partial \mathbf{z}_{\mathbf{y}_0}}{\partial \mathbf{y}}\frac{\partial \mathbf{y}}{\partial \mathbf{m}} + \frac{\partial \mathbf{v}_{\mathbf{y}_0}}{\partial \mathbf{m}} + \frac{\partial \mathbf{v}_{\mathbf{y}_0}}{\partial \mathbf{y}}\frac{\partial \mathbf{y}}{\partial \mathbf{m}} + \frac{\partial \mathbf{t}}{\partial \mathbf{y}}\frac{\partial \mathbf{x}_{\mathbf{y}_0}}{\partial \mathbf{m}} = \mathbf{0}.$$

  Therefore

$$\text{Ⓑ} = \frac{\partial \mathbf{x}_{\mathbf{y}_0}}{\partial \mathbf{m}} = -\left(\frac{\partial \mathbf{t}}{\partial \mathbf{y}}\right)^{-1}\left(\frac{\partial \mathbf{z}_{\mathbf{y}_0}}{\partial \mathbf{m}} + \frac{\partial \mathbf{z}_{\mathbf{y}_0}}{\partial \mathbf{y}}\frac{\partial \mathbf{y}}{\partial \mathbf{m}} + \frac{\partial \mathbf{v}_{\mathbf{y}_0}}{\partial \mathbf{m}} + \frac{\partial \mathbf{v}_{\mathbf{y}_0}}{\partial \mathbf{y}}\frac{\partial \mathbf{y}}{\partial \mathbf{m}}\right).$$

- Ⓔ: Differentiate (B.8) with respect to $\mathbf{y}_0$:

$$\frac{\partial \mathbf{z}_{\mathbf{y}_0}}{\partial \mathbf{y}_0} + \frac{\partial \mathbf{z}_{\mathbf{y}_0}}{\partial \mathbf{y}}\frac{\partial \mathbf{y}}{\partial \mathbf{y}_0} + \frac{\partial \mathbf{v}_{\mathbf{y}_0}}{\partial \mathbf{y}_0} + \frac{\partial \mathbf{v}_{\mathbf{y}_0}}{\partial \mathbf{y}}\frac{\partial \mathbf{y}}{\partial \mathbf{y}_0} + \frac{\partial \mathbf{t}}{\partial \mathbf{y}}\frac{\partial \mathbf{x}_{\mathbf{y}_0}}{\partial \mathbf{y}_0} = \mathbf{0}.$$

  Therefore

$$\text{Ⓔ} = \frac{\partial \mathbf{x}_{\mathbf{y}_0}}{\partial \mathbf{y}_0} = -\left(\frac{\partial \mathbf{t}}{\partial \mathbf{y}}\right)^{-1}\left(\frac{\partial \mathbf{z}_{\mathbf{y}_0}}{\partial \mathbf{y}_0} + \frac{\partial \mathbf{z}_{\mathbf{y}_0}}{\partial \mathbf{y}}\frac{\partial \mathbf{y}}{\partial \mathbf{y}_0} + \frac{\partial \mathbf{v}_{\mathbf{y}_0}}{\partial \mathbf{y}_0} + \frac{\partial \mathbf{v}_{\mathbf{y}_0}}{\partial \mathbf{y}}\frac{\partial \mathbf{y}}{\partial \mathbf{y}_0}\right).$$

- Ⓗ: Differentiate (B.8) with respect to $\mathbf{s}$:

$$\frac{\partial \mathbf{z}_{\mathbf{y}_0}}{\partial \mathbf{y}}\frac{\partial \mathbf{y}}{\partial \mathbf{s}} + \frac{\partial \mathbf{v}_{\mathbf{y}_0}}{\partial \mathbf{y}}\frac{\partial \mathbf{y}}{\partial \mathbf{s}} + \frac{\partial \mathbf{t}}{\partial \mathbf{y}}\frac{\partial \mathbf{x}_{\mathbf{y}_0}}{\partial \mathbf{s}} = \mathbf{0}.$$

  Therefore

$$\text{Ⓗ} = \frac{\partial \mathbf{x}_{\mathbf{y}_0}}{\partial \mathbf{s}} = -\left(\frac{\partial \mathbf{t}}{\partial \mathbf{y}}\right)^{-1}\left(\frac{\partial \mathbf{z}_{\mathbf{y}_0}}{\partial \mathbf{y}}\frac{\partial \mathbf{y}}{\partial \mathbf{s}} + \frac{\partial \mathbf{v}_{\mathbf{y}_0}}{\partial \mathbf{y}}\frac{\partial \mathbf{y}}{\partial \mathbf{s}}\right).$$

• Ⓒ, Ⓕ, Ⓘ: Multiply (3.7) by $\mathbf{w}_{\mathbf{s}}$,

$$\frac{\partial \mathbf{t}}{\partial \mathbf{y}}\frac{\partial \mathbf{y}}{\partial \mathbf{s}}\mathbf{w}_{\mathbf{s}} = \mathbf{S}\,\frac{\partial \mathbf{q}}{\partial \mathbf{s}}\mathbf{w}_{\mathbf{s}}, \tag{B.9}$$

with $\mathbf{t} = \mathbf{t}\left(\mathbf{y}\left(\mathbf{p}\right), \mathbf{m}, \mathbf{y}_0\right)$, and let $\mathbf{x}_{\mathbf{s}} = \dfrac{\partial \mathbf{y}}{\partial \mathbf{s}}\mathbf{w}_{\mathbf{s}}$ and $\mathbf{v}_{\mathbf{s}} = \dfrac{\partial \mathbf{t}}{\partial \mathbf{y}}\mathbf{x}_{\mathbf{s}}$.

- Ⓒ: Differentiate (B.9) with respect to $\mathbf{m}$:

$$\frac{\partial \mathbf{v}_{\mathbf{s}}}{\partial \mathbf{m}} + \frac{\partial \mathbf{v}_{\mathbf{s}}}{\partial \mathbf{y}}\frac{\partial \mathbf{y}}{\partial \mathbf{m}} + \frac{\partial \mathbf{t}}{\partial \mathbf{y}}\frac{\partial \mathbf{x}_{\mathbf{s}}}{\partial \mathbf{m}} = \mathbf{0}_{N\times N_{\mathbf{m}}}.$$

Therefore

$$\text{Ⓒ} = \frac{\partial \mathbf{x_s}}{\partial \mathbf{m}} = -\left(\frac{\partial \mathbf{t}}{\partial \mathbf{y}}\right)^{-1}\left(\frac{\partial \mathbf{v_s}}{\partial \mathbf{m}} + \frac{\partial \mathbf{v_s}}{\partial \mathbf{y}}\frac{\partial \mathbf{y}}{\partial \mathbf{m}}\right). \tag{B.10}$$

- Ⓕ: Differentiate (B.9) with respect to $\mathbf{y}_0$:

$$\frac{\partial \mathbf{v_s}}{\partial \mathbf{y}_0} + \frac{\partial \mathbf{v_s}}{\partial \mathbf{y}}\frac{\partial \mathbf{y}}{\partial \mathbf{y}_0} + \frac{\partial \mathbf{t}}{\partial \mathbf{y}}\frac{\partial \mathbf{x_s}}{\partial \mathbf{y}_0} = \mathbf{0}.$$

Therefore

$$\text{Ⓕ} = \frac{\partial \mathbf{x_s}}{\partial \mathbf{y}_0} = -\left(\frac{\partial \mathbf{t}}{\partial \mathbf{y}}\right)^{-1}\left(\frac{\partial \mathbf{v_s}}{\partial \mathbf{y}_0} + \frac{\partial \mathbf{v_s}}{\partial \mathbf{y}}\frac{\partial \mathbf{y}}{\partial \mathbf{y}_0}\right).$$

- Ⓘ: Differentiate (B.9) with respect to $\mathbf{s}$:

$$\frac{\partial \mathbf{v_s}}{\partial \mathbf{y}}\frac{\partial \mathbf{y}}{\partial \mathbf{s}} + \frac{\partial \mathbf{t}}{\partial \mathbf{y}}\frac{\partial \mathbf{x_s}}{\partial \mathbf{s}} = \mathbf{S}\,\frac{\partial^2 \mathbf{q}}{\partial \mathbf{s}\partial \mathbf{s}}\mathbf{w_s}.$$

Therefore

$$\text{Ⓘ} = \frac{\partial \mathbf{x_s}}{\partial \mathbf{m}} = \left(\frac{\partial \mathbf{t}}{\partial \mathbf{y}}\right)^{-1}\left(\mathbf{S}\,\frac{\partial^2 \mathbf{q}}{\partial \mathbf{s}\partial \mathbf{s}}\mathbf{w_s} - \frac{\partial \mathbf{v_s}}{\partial \mathbf{y}}\frac{\partial \mathbf{y}}{\partial \mathbf{s}}\right). \tag{B.11}$$

Hence, after some simplification,

$$\begin{aligned}
\frac{\partial^2 \mathbf{y}}{\partial \mathbf{m}\partial \mathbf{p}}\mathbf{w_p} &= \text{Ⓐ} + \text{Ⓑ} + \text{Ⓒ} \\
&= -\left(\frac{\partial \mathbf{t}}{\partial \mathbf{y}}\right)^{-1}\left(\frac{\partial^2 \mathbf{t}}{\partial \mathbf{m}\partial \mathbf{m}}\mathbf{w_m} + \frac{\partial^2 \mathbf{t}}{\partial \mathbf{m}\partial \mathbf{y}_0}\mathbf{w_{y_0}} + \frac{\partial^2 \mathbf{t}}{\partial \mathbf{m}\partial \mathbf{y}}\frac{\partial \mathbf{y}}{\partial \mathbf{p}}\mathbf{w_p} + \right. \\
&\qquad \left. + \left(\frac{\partial^2 \mathbf{t}}{\partial \mathbf{y}\partial \mathbf{m}}\mathbf{w_m} + \frac{\partial^2 \mathbf{t}}{\partial \mathbf{y}\partial \mathbf{y}_0}\mathbf{w_{y_0}} + \frac{\partial^2 \mathbf{t}}{\partial \mathbf{y}\partial \mathbf{y}}\frac{\partial \mathbf{y}}{\partial \mathbf{p}}\mathbf{w_p}\right)\frac{\partial \mathbf{y}}{\partial \mathbf{m}}\right),
\end{aligned} \tag{B.12}$$

$$\begin{aligned}
\frac{\partial^2 \mathbf{y}}{\partial \mathbf{y}_0\partial \mathbf{p}}\mathbf{w_p} &= \text{Ⓓ} + \text{Ⓔ} + \text{Ⓕ} \\
&= -\left(\frac{\partial \mathbf{t}}{\partial \mathbf{y}}\right)^{-1}\left(\frac{\partial^2 \mathbf{t}}{\partial \mathbf{y}_0\partial \mathbf{m}}\mathbf{w_m} + \frac{\partial^2 \mathbf{t}}{\partial \mathbf{y}_0\partial \mathbf{y}_0}\mathbf{w_{y_0}} + \frac{\partial^2 \mathbf{t}}{\partial \mathbf{y}_0\partial \mathbf{y}}\frac{\partial \mathbf{y}}{\partial \mathbf{p}}\mathbf{w_p} + \right. \\
&\qquad \left. + \left(\frac{\partial^2 \mathbf{t}}{\partial \mathbf{y}\partial \mathbf{m}}\mathbf{w_m} + \frac{\partial^2 \mathbf{t}}{\partial \mathbf{y}\partial \mathbf{y}_0}\mathbf{w_{y_0}} + \frac{\partial^2 \mathbf{t}}{\partial \mathbf{y}\partial \mathbf{y}}\frac{\partial \mathbf{y}}{\partial \mathbf{p}}\mathbf{w_p}\right)\frac{\partial \mathbf{y}}{\partial \mathbf{y}_0}\right),
\end{aligned} \tag{B.13}$$

$$\begin{aligned}
\frac{\partial^2 \mathbf{y}}{\partial \mathbf{s}\partial \mathbf{p}}\mathbf{w_p} &= \text{Ⓖ} + \text{Ⓗ} + \text{Ⓘ} \\
&= -\left(\frac{\partial \mathbf{t}}{\partial \mathbf{y}}\right)^{-1}\left(-\mathbf{S}\,\frac{\partial^2 \mathbf{q}}{\partial \mathbf{s}\partial \mathbf{s}}\mathbf{w_s} + \right.
\end{aligned}$$

$$+\left(\frac{\partial^2\mathbf{t}}{\partial\mathbf{y}\partial\mathbf{m}}\mathbf{w_m}+\frac{\partial^2\mathbf{t}}{\partial\mathbf{y}\partial\mathbf{y_0}}\mathbf{w_{y_0}}+\frac{\partial^2\mathbf{t}}{\partial\mathbf{y}\,\partial\mathbf{y}}\frac{\partial\mathbf{y}}{\partial\mathbf{p}}\mathbf{w_p}\right)\frac{\partial\mathbf{y}}{\partial\mathbf{s}}\Bigg). \quad \text{(B.14)}$$

$\frac{\partial^2\mathbf{y}}{\partial\mathbf{p}\partial\mathbf{p}}\mathbf{w_p}$ then is

$$\begin{aligned}
\frac{\partial^2\mathbf{y}}{\partial\mathbf{p}\partial\mathbf{p}}\mathbf{w_p}^\top &= \begin{bmatrix} \frac{\partial^2\mathbf{y}}{\partial\mathbf{m}\partial\mathbf{p}}\mathbf{w_p}^\top \\ \frac{\partial^2\mathbf{y}}{\partial\mathbf{y_0}\partial\mathbf{p}}\mathbf{w_p}^\top \\ \frac{\partial^2\mathbf{y}}{\partial\mathbf{s}\partial\mathbf{p}}\mathbf{w_p}^\top \end{bmatrix} \\
&= -\begin{bmatrix} \frac{\partial^2\mathbf{t}}{\partial\mathbf{m}\partial\mathbf{m}}\mathbf{w_m}^\top + \frac{\partial^2\mathbf{t}}{\partial\mathbf{m}\partial\mathbf{y_0}}\mathbf{w_{y_0}}^\top + \frac{\partial^2\mathbf{t}}{\partial\mathbf{m}\partial\mathbf{y}}\frac{\partial\mathbf{y}}{\partial\mathbf{p}}\mathbf{w_p}^\top \\ \frac{\partial^2\mathbf{t}}{\partial\mathbf{y_0}\partial\mathbf{m}}\mathbf{w_m}^\top + \frac{\partial^2\mathbf{t}}{\partial\mathbf{y_0}\partial\mathbf{y_0}}\mathbf{w_{y_0}}^\top + \frac{\partial^2\mathbf{t}}{\partial\mathbf{y_0}\partial\mathbf{y}}\frac{\partial\mathbf{y}}{\partial\mathbf{p}}\mathbf{w_p}^\top \\ -\frac{\partial^2\mathbf{q}}{\partial\mathbf{s}\partial\mathbf{s}}\mathbf{w_s}^\top\mathbf{S}^\top \end{bmatrix}\frac{\partial\mathbf{t}}{\partial\mathbf{y}}^{-\top} + \\
&\quad -\frac{\partial\mathbf{y}}{\partial\mathbf{p}}^\top\left(\frac{\partial^2\mathbf{t}}{\partial\mathbf{y}\partial\mathbf{m}}\mathbf{w_m}^\top + \frac{\partial^2\mathbf{t}}{\partial\mathbf{y}\partial\mathbf{y_0}}\mathbf{w_{y_0}}^\top + \frac{\partial^2\mathbf{t}}{\partial\mathbf{y}\,\partial\mathbf{y}}\frac{\partial\mathbf{y}}{\partial\mathbf{p}}\mathbf{w_p}^\top\right)\frac{\partial\mathbf{t}}{\partial\mathbf{y}}^{-\top}.
\end{aligned}$$

Plugging all of this into ④ and using the definition of the adjoint solution $\boldsymbol{\lambda} = \frac{\partial\mathbf{t}}{\partial\mathbf{y}}^{-\top}\frac{\partial\mathbf{d}}{\partial\mathbf{y}}^\top\nabla_\mathbf{d}\mathrm{M}$, we therefore have:

$$\begin{aligned}
④ &= -\begin{bmatrix} \frac{\partial^2\mathbf{t}}{\partial\mathbf{m}\partial\mathbf{m}}\mathbf{w_m}^\top + \frac{\partial^2\mathbf{t}}{\partial\mathbf{m}\partial\mathbf{y_0}}\mathbf{w_{y_0}}^\top + \frac{\partial^2\mathbf{t}}{\partial\mathbf{m}\partial\mathbf{y}}\mathbf{x}^\top \\ \frac{\partial^2\mathbf{t}}{\partial\mathbf{y_0}\partial\mathbf{m}}\mathbf{w_m}^\top + \frac{\partial^2\mathbf{t}}{\partial\mathbf{y_0}\partial\mathbf{y_0}}\mathbf{w_{y_0}}^\top + \frac{\partial^2\mathbf{t}}{\partial\mathbf{y_0}\partial\mathbf{y}}\mathbf{x}^\top \\ -\frac{\partial^2\mathbf{q}}{\partial\mathbf{s}\partial\mathbf{s}}\mathbf{w_s}^\top\mathbf{S}^\top \end{bmatrix}\boldsymbol{\lambda} + \\
&\quad -\frac{\partial\mathbf{y}}{\partial\mathbf{p}}^\top\left(\frac{\partial^2\mathbf{t}}{\partial\mathbf{y}\partial\mathbf{m}}\mathbf{w_m}^\top + \frac{\partial^2\mathbf{t}}{\partial\mathbf{y}\partial\mathbf{y_0}}\mathbf{w_{y_0}}^\top + \frac{\partial^2\mathbf{t}}{\partial\mathbf{y}\,\partial\mathbf{y}}\mathbf{x}^\top\right)\boldsymbol{\lambda},
\end{aligned}$$

with $\mathbf{x} = \dfrac{\partial \mathbf{y}}{\partial \mathbf{p}} \mathbf{w_p}$. Therefore the action of the Hessian is given by

$$
\begin{aligned}
\mathcal{H}_{\mathrm{M}} \mathbf{w_p} &= \frac{\partial^2 \mathrm{M}}{\partial \mathbf{p} \partial \mathbf{p}} \mathbf{w_p} \\
&= \textcircled{1} + \textcircled{2} = \textcircled{1} + \textcircled{3} + \textcircled{4} \\
&= \mathbf{J}^\top \frac{\partial^2 \mathrm{M}}{\partial \mathbf{d} \partial \mathbf{d}} \mathbf{J}\, \mathbf{w} + \frac{\partial \mathbf{y}}{\partial \mathbf{p}}^\top \left(\frac{\partial^2 \mathbf{d}}{\partial \mathbf{y} \partial \mathbf{y}} \mathbf{x} \right)^\top \nabla_{\mathbf{d}} \mathrm{M} + \\
&\quad - \begin{bmatrix} \frac{\partial^2 \mathbf{t}}{\partial \mathbf{m} \partial \mathbf{m}} \mathbf{w_m}^\top + \frac{\partial^2 \mathbf{t}}{\partial \mathbf{m} \partial \mathbf{y_0}} \mathbf{w_{y_0}}^\top + \frac{\partial^2 \mathbf{t}}{\partial \mathbf{m} \partial \mathbf{y}} \mathbf{x}^\top \\ \frac{\partial^2 \mathbf{t}}{\partial \mathbf{y_0} \partial \mathbf{m}} \mathbf{w_m}^\top + \frac{\partial^2 \mathbf{t}}{\partial \mathbf{y_0} \partial \mathbf{y_0}} \mathbf{w_{y_0}}^\top + \frac{\partial^2 \mathbf{t}}{\partial \mathbf{y_0} \partial \mathbf{y}} \mathbf{x}^\top \\ -\frac{\partial^2 \mathbf{q}}{\partial \mathbf{s} \partial \mathbf{s}} \mathbf{w_s}^\top \mathbf{S}^\top \end{bmatrix} \boldsymbol{\lambda} + \\
&\quad - \frac{\partial \mathbf{y}}{\partial \mathbf{p}}^\top \left(\frac{\partial^2 \mathbf{t}}{\partial \mathbf{y} \partial \mathbf{m}} \mathbf{w_m}^\top + \frac{\partial^2 \mathbf{t}}{\partial \mathbf{y} \partial \mathbf{y_0}} \mathbf{w_{y_0}}^\top + \frac{\partial^2 \mathbf{t}}{\partial \mathbf{y} \partial \mathbf{y}} \mathbf{x}^\top \right) \boldsymbol{\lambda} \\
&= \begin{bmatrix} -\frac{\partial \mathbf{t}}{\partial \mathbf{m}}^\top \\ -\frac{\partial \mathbf{t}}{\partial \mathbf{y_0}}^\top \\ \frac{\partial \mathbf{q}}{\partial \mathbf{s}}^\top \mathbf{S}^\top \end{bmatrix} \boldsymbol{\mu} - \begin{bmatrix} \frac{\partial^2 \mathbf{t}}{\partial \mathbf{m} \partial \mathbf{m}} \mathbf{w_m}^\top + \frac{\partial^2 \mathbf{t}}{\partial \mathbf{m} \partial \mathbf{y_0}} \mathbf{w_{y_0}}^\top + \frac{\partial^2 \mathbf{t}}{\partial \mathbf{m} \partial \mathbf{y}} \mathbf{x}^\top \\ \frac{\partial^2 \mathbf{t}}{\partial \mathbf{y_0} \partial \mathbf{m}} \mathbf{w_m}^\top + \frac{\partial^2 \mathbf{t}}{\partial \mathbf{y_0} \partial \mathbf{y_0}} \mathbf{w_{y_0}}^\top + \frac{\partial^2 \mathbf{t}}{\partial \mathbf{y_0} \partial \mathbf{y}} \mathbf{x}^\top \\ -\frac{\partial^2 \mathbf{q}}{\partial \mathbf{s} \partial \mathbf{s}} \mathbf{w_s}^\top \mathbf{S}^\top \end{bmatrix} \boldsymbol{\lambda} \qquad \text{(B.15a)}
\end{aligned}
$$

with

$$
\begin{aligned}
\boldsymbol{\mu} = \frac{\partial \mathbf{t}}{\partial \mathbf{y}}^{-\top} \Bigg(& \frac{\partial \mathbf{d}}{\partial \mathbf{y}}^\top \frac{\partial^2 \mathrm{M}}{\partial \mathbf{d} \partial \mathbf{d}} \mathbf{J}\, \mathbf{w} + \left(\frac{\partial^2 \mathbf{d}}{\partial \mathbf{y} \partial \mathbf{y}} \mathbf{x} \right)^\top \nabla_{\mathbf{d}} \mathrm{M} + \\
& - \left(\frac{\partial^2 \mathbf{t}}{\partial \mathbf{y} \partial \mathbf{m}} \mathbf{w_m}^\top + \frac{\partial^2 \mathbf{t}}{\partial \mathbf{y} \partial \mathbf{y_0}} \mathbf{w_{y_0}}^\top + \frac{\partial^2 \mathbf{t}}{\partial \mathbf{y} \partial \mathbf{y}} \mathbf{x}^\top \right) \boldsymbol{\lambda} \Bigg). \qquad \text{(B.15b)}
\end{aligned}
$$