Calculus Identities for Generalized Simplex Gradients: Rules and Applications

by

Gabriel Jarry-Bolduc

B.Sc. Hons., Université du Québec à Trois-Rivières, 2017

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

MASTER OF SCIENCE

in

The College of Graduate Studies

(Mathematics)

THE UNIVERSITY OF BRITISH COLUMBIA

(Okanagan)

June 2019

The following individuals certify that they have read, and recommend to the College of Graduate Studies for acceptance, a thesis/dissertation entitled:

Calculus Identities for Generalized Simplex Gradients: Rules and Applications

submitted by Gabriel Jarry-Bolduc in partial fulfilment of the requirements of the degree of Master of Science

Warren Hare, I. K. Barber School of Arts & Sciences
Supervisor

Yves Lucet, I. K. Barber School of Arts & Sciences
Supervisory Committee Member

Andreas Holger, I. K. Barber School of Arts & Sciences
Supervisory Committee Member

Julien Cheng, School of Engineering
Supervisory Committee Member

Abstract

Simplex gradients, essentially the gradient of a linear approximation, are a popular tool in derivative-free optimization (DFO). In 2015, a product rule, a quotient rule and a sum rule for simplex gradients were introduced by Regis [33]. Unfortunately, those calculus rules only work under a restrictive set of assumptions. The purpose of this thesis is to provide new calculus rules that work in a wider setting. The rules place minimal assumptions on the functions involved and the interpolation sets. The rules further lead to an alternative approach to gradient approximation in situations where the rules could be applied. We analyze the new approach, provide error bounds, include some testing on numerical stability and accuracy.

Lay Summary

Differential calculus studies instantaneous rate of change of a function. For a function of several variables, the instantaneous rate of change of a function is captured in a *vector* called the *gradient*. The gradient of a function is highly valued to solve an optimization problem. Unfortunately, in many modern optimization problems, gradients are not available or simply not reliable. For those reasons, we approximate the latter using accessible information about our optimization problem. One method that was proven to be efficient is called the *simplex gradient*. The main contributions of this thesis are to introduce calculus rules for simplex gradients such as a product rule, a quotient rule, and a chain rule. Those rules have the same structure as those for true gradients plus a term that can be viewed as an error term. Removing the error terms from the formulas lead to new techniques to gradient approximations. We demonstrate that it is advantageous to use these new techniques in certain situations.

Preface

This thesis has been adapted from a manuscript co-authored with Dr. Warren Hare at the University of British Columbia, Okanagan Campus. The manuscript has been submitted for publication to SIAM Journal on Optimization [22].

Table of Contents

List of Tables

List of Figures

Acknowledgements

Thank you to Dr. Warren Hare, my co-author and supervisor. You have inspired me and helped me become a better researcher.

Thank you to NSERC and UBC for funding this research.

Dedication

À mes parents:

Merci pour tout ce que vous avez fait pour moi.
Merci d'avoir continuer de croire en moi malgré toutes ces années difficiles.
Merci de m'avoir appris à ne jamais abandonner.
Je ne pourrai jamais assez vous remercier.

Je vous aime.

Chapter 1

Introduction

Calculus is considered to have been developed in the seventeenth century by Isaac Newton and Gottfried Wilhelm Leibniz. It is the mathematical study of continuous change. One major branch is called *differential calculus.* It studies instantaneous rate of change of a function f. When a function involves several variables, the instantaneous rate of change can be computed through use of a vector called the *gradient* of f.

Gradients are an extremely powerful tool in *optimization.* Before saying why gradients are valuable, let us define the branch of mathematics named optimization. Optimization is the study of mathematical problem of minimizing or maximizing a function f possibly subject to some constraints [4]. In this thesis, we always assume we want to minimize a problem. With this definition in mind, it is now clear why optimization is such a popular branch of mathematics in modern research. Indeed, a lot of recent mathematical problems involve minimizing the cost of some procedures or projects.

To solve an optimization problem, we cannot deny the importance of gradients. The gradient of our function f at a point x represents the direction of the greatest rate of increase of our function f at x. Thus, the negative gradient represents the greatest rate of decrease of f at x. Hence, one rudimentary strategy is to follow the direction of the negative gradient to reduce the value of a function. Unfortunately, the gradient of our *objective function* f, the function we want to minimize, is not always available or even reliable. Let us provide situations where this is the case.

First, a common optimization problem where the gradient is not available to the optimizer is when the objective function comes from a computer simulation [4]. Even though the objective function could be *fully differentiable* (and so the gradient of f exists everywhere), the output does not include derivatives. For instance, the simulation could only return the function value $f(x)$ for some input x. In this situation, it is not possible to use directly gradient-based method such as the well-known *Quasi-Newton method* [5, Chapter 4].

Second, *Noisy* optimization problems are a classic example where the gradient of the objective function f is not reliable. Roughly speaking, a noisy function is a function with small perturbations that produces several quick increases and decreases of the function values. Figure 1 illustrates such a function. This noisy function is the sum of the well-behaved function $f(x) = x^2$ and small perturbations coming from normally distributed random numbers. In real-life applications, these small oscillations could be caused by experimental or measurement errors [8]. We can see that gradient-based methods may easily become trapped in one of the several local *minimizers*, the values of the independent variable x associated to the minimum function values $f(x)$.

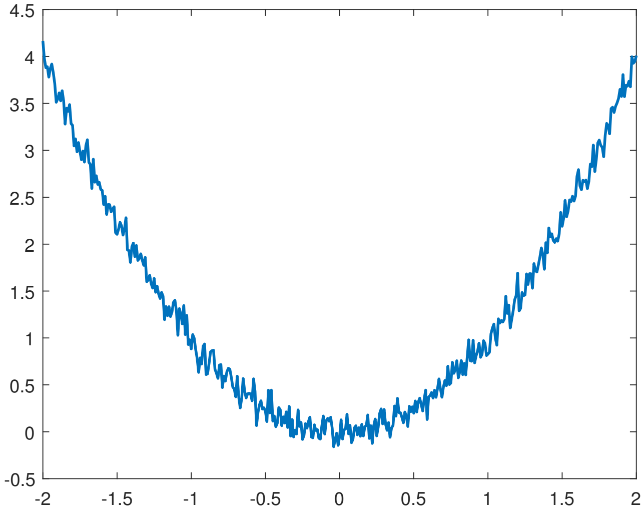

Figure 1: An Example of Noisy Function

To solve the previous examples, it should now be clear that we cannot use the true gradient of the objective function f. The modern branch of mathematics that is interested by those previous problems is called *derivative-free optimization.*

Derivative-free optimization is the mathematical study of optimization algorithms that do not employ first order information such as gradients [4]. Note that we did not claim first order information does not exist. Indeed, The objective function can be fully differentiable but for some justifiable reasons, we do not use first order information directly.

In Derivative-free optimization, it is crucial to understand that, more often than not, the explicit functions are unknown to the optimizer. The functions are hidden in what we refer to a *blackbox*. A blackbox is any process that returns an output whenever we provide an input, but the inner workings of the process are not analytically available [4]. Indeed, a computer simulation is a type of blackbox. A second type of blackbox is laboratory experiments. As we have seen, conducting a laboratory experiment can return a noisy function. Moreover, we often do not know the explicit mathematical functions involved in our experiment.

A legitimate question to ask is how can we solve those optimization problems?

Instead of using the gradient of our objective function f directly, we can approximate the latter with available and reliable information we possess on f. One method that was proven to be efficient is to build the *simplex gradient* of f. Simplex gradients are essentially the gradient of a linear approximation.

Suppose we want to approximate the gradient of the function $f(x) = x^2$ at the point $x = 1$. Assume that the only information we possess is that $f(1) = 1$ and $f(2) = 4$. We can build a linear function passing through $f(1) = 1$ and $f(2) = 4$. The slope of this linear function, which is equal to 3, represents the simplex gradient of f at $x = 1$. Note that the true gradient of f at $x = 1$ is equal to 2. Figure 2 illustrates this example. We see that the simplex gradient of f at $x = 1$ is not extremely accurate. To obtain an accurate simplex gradient, we need to obtain information about our function f closer to $x = 1$. This simple example demonstrates the importance of the *sample set*, the set of points used to build our approximation of the gradient.

Derivative-free optimization methods have been effectively applied to a wide range of fields: oil production problems [15, 20], molecular geometry [2, 26], helicopter rotor blade design [6, 7, 35], research on water resources [1, 17, 27, 29], alloy and process design [9, 18, 19] and engineering applications [3, 13] to name a few. Undeniably, the value of derivative-free methods and simplex gradients is now well-established.

Now that we have a better understanding of derivative-free optimization and simplex gradients, let us present a brief history of this branch of mathematics and introduce specific details about derivative-free optimization and simplex gradients.

We can classify derivative-free methods into two main categories: model-based methods and direct search methods. In both categories, simplex gradients can play an important role.

In derivative-free optimization, a model-based method approximates the objective function with a model function and then utilizes the model function

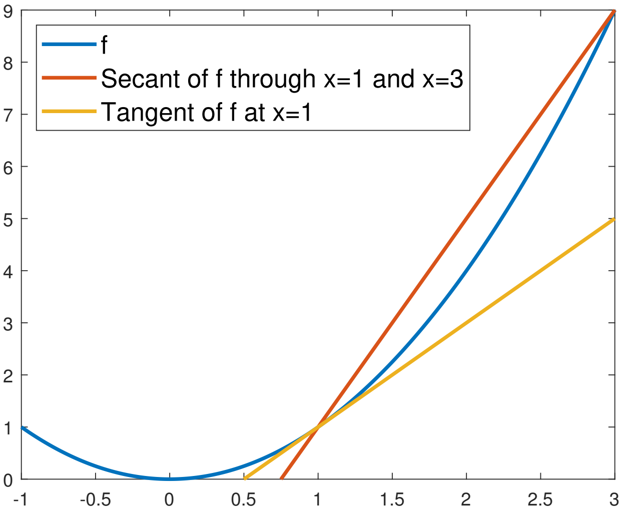

Figure 2: An Example of Simplex Gradient

to guide the optimization. The beginning of model-based methods occurred in 1969 when Winfield presented his Ph.D thesis *Function and functional optimization by interpolation in data tables* [37]. However, model-based methods in derivative-free optimization were generally considered too computationally expensive until the mid 1990's when Powell developed rigorous analysis for a method based on linear interpolation [32]. This led to the development of simplex gradients. Simplex gradients are now frequently used in derivative-free optimization. More recently, a meticulous theory on building models was developed by Conn, Scheinberg and Vicente [10, 11]. The main value of simplex gradients in model-based methods is to determine a *descent direction* of the true function [4, Chapter 10]. Even when the objective function is nonsmooth, simplex gradients can be defined and can help solve the optimization problem [12].

A direct search method is a type of method that works from an incumbent solution and analyzes a collection of trial points to find improvement in the objective function. If no improvement is found then a step size parameter is adjusted. Initial works on direct search methods include Hooke and Jeeves [23] and Nelder and Mead [30] published in the 1960's. In 2007, Custódio and Vicente suggested various strategies to improve the performance of direct search methods using the simplex gradient [14]. A year after, Custódio et al. demonstrated that the efficiency of direct search methods can be improved by reordering the poll directions according to descent indicators built from simplex gradients. Moreover, they defined a new stopping criterion for direct search methods involving the simplex gradient [12].

Several properties of simplex gradients were analyzed in *Iterative Methods for Optimization* [25]. Error bounds for the simplex gradient were provided and the notion of *centered simplex gradient* was introduced. Thereafter, the importance of the sample set geometry (*poisedness*) which is used to build the simplex gradient was deeply investigated [11**?**]. In 2010, the strong dependence between the geometry of the sample set and global convergence of a model-based algorithm was revealed [34]. The utility of simplex gradients in nonsmooth optimization is an active area of research. On that topic, Bortz and Kelley presented some benefits of using simplex gradients to solve noisy optimization problems [8].

In 2015, a publication by Regis proposed some calculus rules for the simplex gradient: a product rule, a quotient rule and a sum rule [33]. Unfortunately, those rules only work under a restrictive set of assumptions. Moreover, those rules did not have the same structure as the calculus rules for the true gradient. In 2017, Hare began investigating compositions of functions [21].

There are two main achievements presented in this thesis. First, we introduce a rigorous repertoire of calculus rules for the *generalized simplex gradient*. These rules work in a wider setting than the one introduced by Regis [33]. Indeed, the following calculus rules can be used regardless of the number of points in the sample set and place minimal assumptions on the sample set. It also places minimal assumptions on the functions involved. It turns out that the calculus rules introduce in this thesis have the same structure as the true gradient plus a term E that can be viewed as an error term. Removing the term E from the formula leads to new techniques to approximate gradients. This represents the second main achievement of this thesis. This new approach, named *generalized simplex calculus gradient*, has interesting benefits. For instance, in the quotient rule and the power rule, it allows us to remove some assumptions on the functions involved. Also, under a certain assumption on the sample set, this new approach suits perfectly linear functions as it returns the true gradient of the objective function.

Since limited information about the functions involved is available to the optimizer, it can be difficult to decide on which gradient approximation technique to use on a specific problem. Information about the Lipschitz constants may be useful. In that sense, we propose an algorithm to make this decision based on an approximation of the Lipschitz constants in Chapter 7.

To support the relevance of this thesis, let us present an example where one could use this new technique to approximate the gradient. Suppose the functions $f_i(x)$ for all $i \in \{1, 2, \dots, n\}$ are blackboxes returning the probability of an event taking place depending on a variable $x \in \mathbb{R}^d$. Assume we are interested in minimizing the probability of these n events happening simultaneously. In other words, we want to minimize $F(x) = f_1(x) \cdot f_2(x) \cdots f_n(x)$. Then the generalized calculus gradient could be employed to solve this problem. The product rule developed allows us to consider the blackboxes separately rather than considering this problem as one function $F = f_1 \cdot f_2 \cdots f_n$.

This thesis is structured as follows. In Chapter 2, we define the generalized simplex gradient and introduce some basic definitions. We also provide some preliminary results and definitions that are useful in the subsequent chapters. In Chapter 3, we provide a product rule, a quotient rule and a power rule. In Chapter 4, we introduce a general chain rule for the generalized simplex gradient. In Chapter 5, we explore the potential of the calculus rules. The role of the term E in the formulas is examined. We demonstrate that novel gradient approximation techniques can be achieved by removing this error term. The behaviour of linear functions when used in the generalized simplex calculus gradient formulas is analyzed. Error bounds for the generalized simplex calculus gradients are also developed. In Chapter 6, we compare the numerical stability of the generalized simplex gradient, the generalized simplex gradient using the calculus rule, and the generalized simplex calculus gradient. Also, numerical experiments are conducted using Moré, Garbow and Hillstrom's Test Set [28]. Lastly, Chapter 7 summarizes the work we have accomplished, proposes an algorithm to approximate Lipschitz constants, and suggests some topics to explore in future research.

Chapter 2

Preliminary Definitions

In this chapter, we present the definition of generalized simplex gradient and an error bound for the latter. First, we start with some basic definitions that are useful in the next chapters and we also introduce some of the notation used throughout this thesis.

2.1 Basic Notation and Definitions

Throughout this thesis, we use the following notation. Note that, in general, we follow the notation of [4].

$\mathbb{N}$	Set of all natural numbers: $\{0, 1, 2, \dots\}$
$\mathbb{N}_+$	Set of all positive natural numbers: $\{1, 2, \dots\}$
$\mathbb{R}$	Set of all real numbers.
$\mathbb{R}_+$	Set of all positive real numbers.
$\mathbb{R}^d$	Set of all d dimensional real vectors.

Note that we represent an element of $\mathbb{R}^d$ as a column vector.

Unless stated otherwise, we use d, k, m, n and p to denote elements in $\mathbb{N}_+$.

The *open ball* centered at x_0 with radius $\Delta > 0$, denoted $B(x_0, \Delta)$, is defined as

$$B(x_0, \Delta) = \{x \in \mathbb{R}^d : \|x - x_0\| < \Delta\}.$$

The *closed ball* of the same type is denoted $\bar{B}(x_0, \Delta)$.

A $m \times d$ real matrix A, denoted $A \in \mathbb{R}^{m \times d}$, is an array of scalars consisting of m rows and d columns. The element of A in row i and column j is called the i, j*-entry* of A and is denoted $A_{i,j}$.

For a matrix $A \in \mathbb{R}^{m \times d}$, we denote the *transpose* of A by A^T.

The norm of a vector $x = \begin{bmatrix} x_1 & x_2 & \dots & x_d \end{bmatrix}^T$ is denoted by $\|x\|$ and should be taken as the ℓ_2 norm (Euclidean norm) unless stated otherwise. We have

$$\|x\| = \|x\|_2 = \sqrt{(x_1)^2 + (x_2)^2 + \cdots + (x_d)^2}.$$

In $\mathbb{R}$, we use the notation $|\cdot|$ to denote the norm of a scalar.

Given a matrix A, we use the *induced matrix norm*

$$\|A\| = \|A\|_2 = \max\{\|Ax\|_2 : \|x\| = 1\}.$$

The identity matrix in $\mathbb{R}^{d\times d}$ is denoted $\mathtt{I}_d$.

If $A \in \mathbb{R}^{d\times d}$, that is A is a real square matrix, then we denote the *determinant* of A by $\det(A)$. When $\det(A) \neq 0$, we say A is *invertible* and we denote the inverse of A by A^{-1}.

For nonsquare matrices, we can use a generalization of the inverse matrix which is called *pseudoinverse*. The most known type of matrix pseudoinverse is the *Moore-Penrose Pseudoinverse* [24].

Definition 2.1 (Moore-Penrose Pseudoinverse)**.** Let $A \in \mathbb{R}^{m\times d}$. A matrix, denoted $A^\dagger$, is called the Moore-Penrose pseudoinverse of A and satisfies the following four equations:

i. $AA^\dagger A = A$

ii. $A^\dagger AA^\dagger = A^\dagger$

iii. $(AA^\dagger)^T = AA^\dagger$

iv. $(A^\dagger A)^T = A^\dagger A$.

Note that every matrix $A \in \mathbb{R}^{m\times d}$ has a unique Moore-Penrose pseudoinverse $A^\dagger$ [24].

Definition 2.2 (Linearly Independent)**.** A set of vectors $\{v_1, v_2, \dots, v_m\}$ in $\mathbb{R}^d$ is *linearly independent* if and only if the only solution to

$$\sum_{i=1}^{m} \lambda_i v_i = 0, \quad \lambda_i \in \mathbb{R},$$

is $\lambda_i = 0$ for all $i \in \{1, 2, \dots, m\}$.

When $A \in \mathbb{R}^{m\times d}$ has linearly independent columns, a formula to compute $A^\dagger$ is

$$A^\dagger = \left(A^T A\right)^{-1} A^T \in \mathbb{R}^{d\times m}.$$

This particular pseudoinverse is a *left inverse* as $A^\dagger A = \mathtt{I}_d$.

When $A \in \mathbb{R}^{m\times d}$ has linearly independent rows, a formula to compute $A^\dagger$ is

$$A^\dagger = A^T \left(AA^T\right)^{-1} \in \mathbb{R}^{d\times m}.$$

This particular pseudoinverse is a *right inverse* as $AA^\dagger = \mathtt{I}_m$.

The *rank* of a matrix $A \in \mathbb{R}^{m\times d}$, denoted $\operatorname{rank}(A)$, is the maximal number of linearly independent columns of A. We say a matrix $A \in \mathbb{R}^{m\times d}$ has *full rank* if and only if its rank equals the largest possible rank for a matrix of the same dimension.

Furthermore, we say a matrix $A \in \mathbb{R}^{m\times d}$ has *full column rank* if and only if every columns of A are linearly independent. Similarly, we say a matrix $A \in \mathbb{R}^{m\times d}$ has *full row rank* if and only if every rows of A are linearly independent.

We now turn our attention to some basic notations and definitions related to functions.

The notation $f \in \mathcal{C}^0$ means that f is *continuous* and $f \in \mathcal{C}^k$ means that f is k times *differentiable* and all *partial derivatives* are continuous up to the kth order. We use the term *smooth* to refer to the situation where $f \in \mathcal{C}^2$. Note that if $f \in \mathcal{C}^k$ where $k \geq 2$, then f is smooth.

We now introduce the notion of *Lipschitz continuity* which is a stronger concept than continuity.

Definition 2.3 (Lipschitz Continuity)**.** Let $f : \mathbb{R}^d \to \mathbb{R}$. We say the function f is Lipschitz continuous on the set $\Omega \subseteq \mathbb{R}^d$ if and only if there exists a scalar $L_f \geq 0$ for which

$$\|f(x) - f(y)\| \leq L_f \|x - y\| \quad \text{for all } x, y \in \Omega. \tag{2.1}$$

If it exists, the smallest scalar L_f satisfying Equation (2.1) is called the Lipschitz constant of f on the set Ω.

Furthermore, we say f is locally Lipschitz continuous at $x_0 \in \mathbb{R}^d$ if and only if f is Lipschitz continuous on some open ball centered at x_0. That is, there exists $L_f \geq 0$ and $\Delta > 0$ such that

$$\|f(x) - f(y)\| \leq L_f \|x - y\| \quad \text{for all } x, y \in B\left(x_0, \Delta\right).$$

Writing $f \in \mathcal{C}^{0+}$ with constant L_f near x_0 means that f is Lipschitz continuous with Lipschitz constant L_f on some open ball $B\left(x_0, \Delta\right)$.

2.2 Generalized Simplex Gradient

The purpose of this section is to defined the generalized simplex gradient and present an error bound between the generalized simplex gradient and the true gradient.

First, we introduce some basic definitions and investigate why the word simplex shows up in the name of this method to approximate a gradient.

Definition 2.4 (Convex set)**.** A set Ω is convex if and only if given any two points $x, y \in \Omega$ and any $\theta \in [0,1]$, we have

$$\theta x + (1-\theta)y \in \Omega.$$

In other words, the line segment joining any two points in Ω is entirely contained in Ω.

Definition 2.5 (Convex Hull)**.** The convex hull of a set $\mathcal{X}$, denoted $\operatorname{conv}(\mathcal{X})$, is the smallest convex set containing $\mathcal{X}$.

In $\mathbb{R}^2$, the convex hull of a certain set $\mathcal{X}$ may be visualized as the shape enclosed by a rubber band stretched around $\mathcal{X}$. We can now define a simplex.

Definition 2.6 (Simplex)**.** A simplex in $\mathbb{R}^d$ is the convex hull of exactly $d+1$ distinct points that has nonempty interior.

For example, in $\mathbb{R}$ a simplex is an interval. In $\mathbb{R}^2$, a simplex is a triangle. In $\mathbb{R}^3$, it is a triangular-based pyramid.

Given a set of sample points $\mathcal{X} = \{x_0, x_1, \dots, x_d\} \subset \mathbb{R}^d$, we work with the convex hull of $\mathcal{X}$ to verify if $\operatorname{conv}(\mathcal{X})$ forms a simplex. Figure 3 illustrates the convex hull of the set $\mathcal{X} = \{[1\ 1]^T, [3\ 3]^T, [3\ 1]^T\} \subset \mathbb{R}^2$. In this example, $\operatorname{conv}(\mathcal{X})$ clearly forms a simplex.

Now, let us consider the set $\mathcal{X} = \{[1\ \ 1]^T, [2\ \ 1]^T, [3\ \ 1]^T\} \subset \mathbb{R}^2$. Then $\operatorname{conv}(\mathcal{X})$ does not form a simplex since $\operatorname{conv}(\mathcal{X})$ has an empty interior.

In higher dimensions of $\mathbb{R}^d$, it can be difficult to tell if $\operatorname{conv}(\mathcal{X})$ forms a simplex or not. Fortunately, the following proposition introduces a simple test to check if the convex hull of a set forms a simplex.

Proposition 2.7 (Simplex Test)**.** *[4, Theorem 2.5]*

Let $\mathcal{X} = \{x_0, x_1, \dots, x_d\}$ be a set of $d+1$ points in $\mathbb{R}^d$. Let $S(\mathcal{X}) = \begin{bmatrix} x_1 - x_0 & x_2 - x_0 & \dots & x_d - x_0 \end{bmatrix} \in \mathbb{R}^{d \times d}$. Then $\operatorname{conv}(\mathcal{X})$ forms a simplex if and only if $S(\mathcal{X})$ is invertible.

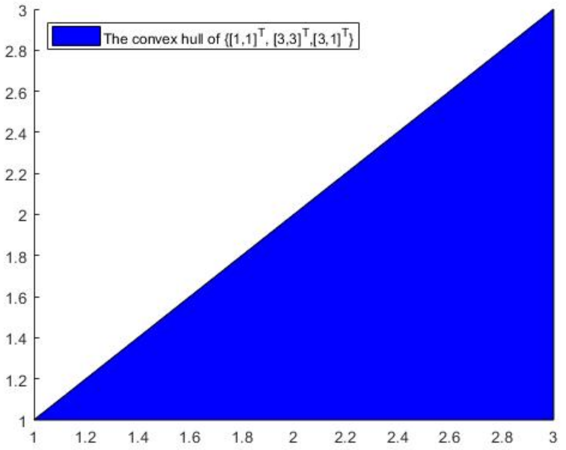

Figure 3: The Convex Hull of $\mathcal{X}$ Forms a Simplex

We are now ready to define the simplex gradient of f over the set $\mathcal{X}$. Consider a set of points $\mathcal{X} = \{x_0, x_1, \dots, x_d\} \subset \mathbb{R}^d$. Treating $\mathcal{X}$ as *interpolation points*, we can build an *interpolation model* $L_{\mathcal{X}}(x) = \alpha_0 + \alpha^T x$, where $\alpha_0 \in \mathbb{R}$ and $\alpha^T = \begin{bmatrix}\alpha_1 & \alpha_2 & \dots & \alpha_d\end{bmatrix} \in \mathbb{R}^d$. We are seeking the values of α_0 and α such that

$$L_{\mathcal{X}}(x_i) = f(x_i), \quad \text{for all } x_i \in \mathcal{X}.$$

We get a system of $d+1$ linear equations and $d+1$ unknowns. In matrix form, we have

$$\begin{bmatrix} 1 & (x_0)^T \\ 1 & (x_1)^T \\ \vdots & \vdots \\ 1 & (x_d)^T \end{bmatrix} \begin{bmatrix} \alpha_0 \\ \alpha_1 \\ \vdots \\ \alpha_d \end{bmatrix} = \begin{bmatrix} f(x_0) \\ f(x_1) \\ \vdots \\ f(x_d) \end{bmatrix}. \tag{2.2}$$

To tighten notation, let

$$f(\mathcal{X}) = \begin{bmatrix} f(x_0) \\ f(x_1) \\ \vdots \\ f(x_d) \end{bmatrix} \quad \text{and} \quad X = \begin{bmatrix} x_0 & x_1 & \dots & x_d \end{bmatrix}.$$

We can rewrite Equation (2.2) as

$$\begin{bmatrix}\mathbf{1} & X^T\end{bmatrix}\begin{bmatrix}\alpha_0 \\ \alpha\end{bmatrix} = f(\mathcal{X}), \tag{2.3}$$

where $\mathbf{1} \in \mathbb{R}^{d+1}$ is the vector of all ones. We want Equation (2.3) to have a unique solution. Hence, we require $\begin{bmatrix}\mathbf{1} & X^T\end{bmatrix}$ to be invertible. The term *poised* refers to this particular situation.

Definition 2.8 (Poised for Linear Interpolation)**.** We say the set $\mathcal{X} = \{x_0, x_1, \dots, x_d\} \subset \mathbb{R}^d$ is poised for linear interpolation if and only if the matrix $\begin{bmatrix}\mathbf{1} & X^T\end{bmatrix} \in \mathbb{R}^{d+1\times d+1}$ is invertible.

The next proposition demonstrates the relation between being poised for linear interpolation and simplices.

Proposition 2.9 (Simplex Test Continued)**.** *[4, Proposition 9.1]*
Let $\mathcal{X} = \{x_0, x_1, \dots, x_d\} \subset \mathbb{R}^d$. Then the following are equivalent.

i. The sample set $\mathcal{X}$ is poised for linear interpolation.

ii. The convex hull $\operatorname{conv}(\mathcal{X})$ *forms a simplex.*

iii. The matrix $S(\mathcal{X}) = \begin{bmatrix}x_1 - x_0 & x_2 - x_0 & \dots & x_d - x_0\end{bmatrix}$ is invertible.

We can now see the intimate link between being poised and simplices. For this reason, the gradient of the linear interpolation function $L_{\mathcal{X}}$ is called the *simplex gradient.*

Definition 2.10 (Simplex Gradient)**.** Let $f : \mathbb{R}^d \to \mathbb{R}$ and let the sample set $\mathcal{X} = \{x_0, x_1, \dots, x_d\} \subset \mathbb{R}^d$ be poised for linear interpolation. The simplex gradient of f over $\mathcal{X}$, denoted $\nabla_S f(\mathcal{X})$, is the gradient of the linear interpolation function $L_{\mathcal{X}}$ of f over $\mathcal{X}$. That is

$$\nabla_S f(\mathcal{X}) = \nabla L_{\mathcal{X}}(x) = \alpha.$$

Note that the notation for the simplex gradient uses the symbol $\mathcal{X}$ since the simplex gradient is defined through the elements of the sample set $\mathcal{X}$. Also, note that the order of the elements in the set $\mathcal{X}$ does not matter. Indeed, changing the order of the elements in $\mathcal{X}$ only swap rows in Equation (2.2) and so the solution of the linear system is unchanged.

Let us introduce a simple formula for computing the simplex gradient.

Proposition 2.11 (Computing the Simplex Gradient)**.** *[4, Proposition 9.2]* *Let $f : \mathbb{R}^d \to \mathbb{R}$, and let $\mathcal{X} = \{x_0, x_1, \dots, x_d\} \subset \mathbb{R}^d$ be poised for linear interpolation. Also, let*

$$S = S(\mathcal{X}) = \begin{bmatrix} x_1 - x_0 & x_2 - x_0 & \dots & x_d - x_0 \end{bmatrix} \in \mathbb{R}^{d \times d},$$

and

$$\delta_f = \delta_f(\mathcal{X}) = \begin{bmatrix} (f(x_1) - f(x_0))^T \\ (f(x_2) - f(x_0))^T \\ \dots \\ (f(x_d) - f(x_0))^T \end{bmatrix} \in \mathbb{R}^d.$$

Then the simplex gradient of f over $\mathcal{X}$ is

$$\nabla_s f(\mathcal{X}) = S^{-T} \delta_f.$$

It is worth mentioning that the usual definition of δ_f does not need a transpose operator in every row. But for a function f where the output is in a higher dimension than $\mathbb{R}$ (such functions are considered in Chapter 4), it is important to include the transpose operator in every row so that δ_f is well-defined.

Example 2.12 (Computing a Simplex Gradient)**.** Let $f : \mathbb{R}^2 \to \mathbb{R} : y \mapsto y_1^2 + y_2^2$, and let the sample set of points $\mathcal{X} = \{[0\ 0]^T, [\delta\ 0]^T, [0\ \delta]^T\}$, where $\delta \in \mathbb{R}_+$. Then

$$\begin{aligned} \nabla_s f(\mathcal{X}) &= S^{-T} \delta_f \\ &= \begin{bmatrix} \delta & 0 \\ 0 & \delta \end{bmatrix}^{-T} \begin{bmatrix} \delta^2 - 0 \\ \delta^2 - 0 \end{bmatrix} \\ &= \begin{bmatrix} \delta \\ \delta \end{bmatrix}. \end{aligned}$$

Note that

$$\nabla f(x_0) = \begin{bmatrix} 0 \\ 0 \end{bmatrix}.$$

In the previous example, we see that the simplex gradient of f over $\mathcal{X}$ is not an accurate approximation of the true gradient of f at x_0 when δ is far from 0. We get an accurate simplex gradient by choosing δ near 0.

Two assumptions to calculate simplex gradients are that the sample set $\mathcal{X}$ contains exactly $d+1$ elements and $\mathcal{X}$ is poised for linear interpolation. One may wonder if we can extend this concept of simplex gradients to the case where $\mathcal{X}$ does not contain exactly $d+1$ elements. Also, can we remove the assumption that $\mathcal{X}$ is well-poised? Generalized simplex gradients allow us to remove those two restrictions on $\mathcal{X}$. To accomplish this, we take advantage of the properties of the Moore-Penrose pseudoinverse. In the rest of this thesis, we use the notation $\langle \cdot \rangle$ to denote an ordered set.

Definition 2.13 (Generalized Simplex Gradient)**.** Let $\mathcal{X} = \langle x_0, x_1, \ldots, x_k \rangle$ be an ordered set of $k+1$ points in $\mathbb{R}^d$. The generalized simplex gradient of f over $\mathcal{X}$ is

$$\nabla_s f(\mathcal{X}) = \left(S^T\right)^{\dagger} \delta_f. \tag{2.4}$$

Note that the set $\mathcal{X}$ is now an ordered set of points. It has been proved that the order of the set $\mathcal{X}$ affects our calculation of the generalized simplex gradient when the set $\mathcal{X}$ contains more than $d+1$ points in $\mathbb{R}^d$ [33, Proposition 3]. In particular, the position of x_0 must be fixed. For this reason, we now consider an ordered set of points $\mathcal{X}$ in the rest of this thesis. Also, note that the sample set $\mathcal{X}$ does not necessarily form a simplex anymore.

We use the term *determined* to refer to the case where $k = d$ and $\operatorname{rank}(S) = d$. We use the term *underdetermined* to refer to the case where $\operatorname{rank}(S) < d$. Lastly, we use the term *overdetermined* to refer to the case where $k > d$ and $\operatorname{rank}(S) = d$.

If we are in the determined or overdetermined cases, and $f \in \mathcal{C}^{1+}$, an error bound between the generalized simplex gradient and the true gradient can be defined [33]. The accuracy of this bound is measured in term of $\Delta = \max_{1 \le i \le k} \|x_i - x_0\|$, the radius of $\mathcal{X}$.

Proposition 2.14 (Error Bound)**.** *Let $\mathcal{X}$ be an ordered set of $k+1$ points in $\mathbb{R}^d$ where $k \ge d$, and let $S(\mathcal{X})$ have full rank. Assume that $f \in \mathcal{C}^{1+}$ with Lipschitz constant $L_{\nabla f} \ge 0$ on an open domain $\mathcal{O}$ containing* $\operatorname{conv}(\mathcal{X})$*. Define*

$$\varepsilon_s f(\mathcal{X}) = \frac{\sqrt{k}}{2} L_{\nabla f} \left\| \left(\widehat{S}\,(\mathcal{X})^T \right)^{\dagger} \right\| \Delta,$$

where $\widehat{S}(\mathcal{X}) = S(\mathcal{X})/\Delta$. Then

$$\|\nabla_s f(\mathcal{X}) - \nabla f(x_0)\| \le \varepsilon_s f(\mathcal{X}).$$

The proof can be done in a similar way than the one in [33, Proposition 7]. Note that the above Proposition 2.14 consider the convex hull $\operatorname{conv}(\mathcal{X})$ instead of the closed ball $\bar{B}\left(x_0, \Delta\right)$ in [33, Proposition 7] . The proof is still valid since it holds that $\nabla f\left(x_0+t(x_i-x_0)\right), t \in [0,1]$, is Lipschitz continuous for all $i \in \{1,2,\dots,k\}$. Considering $\operatorname{conv}(\mathcal{X})$ instead of $\bar{B}\left(x_0, \Delta\right)$ provides a more accurate error bound since it can decrease the value of the Lipschitz constant $L_{\nabla f}$.

Also, note that the previous proposition assumes $k \geq d$. Indeed, if $k < d$ we can drive the absolute error $\|\nabla_s f(\mathcal{X}) - \nabla f(x_0)\|$ to infinity.

Example 2.15 (The Absolute Error Goes to Infinity)**.** Let $f : \mathbb{R}^2 \to \mathbb{R} : y \mapsto y_1 + \beta y_2$, where $\beta \in \mathbb{R}$, and let $\mathcal{X} = \langle [0\ 0]^T, [1\ 0]^T \rangle$.
Note that $L_{\nabla f} = 0$ on $\mathbb{R}^2$ for any $\beta \in \mathbb{R}$.
We have

$$\begin{aligned} \nabla_s f(\mathcal{X}) &= \left(S^T\right)^\dagger \delta_f \\ &= [1\ 0]^\dagger\ [1] \\ &= \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \end{aligned}$$

and

$$\nabla f(x_0) = \begin{bmatrix} 1 \\ \beta \end{bmatrix}.$$

Hence, driving $\beta \longrightarrow \infty$, we get

$$\|\nabla_s f(\mathcal{X}) - \nabla f(x_0)\| \longrightarrow \infty.$$

Therefore, the error bound defined in Proposition 2.14 is not valid in the underdetermined case.

Now that we have an equation to compute the generalized simplex gradient of a function f over a sample set $\mathcal{X}$ and presented the significant definitions and notation, we are ready to introduce calculus rules for the generalized simplex gradient.

Chapter 3

Product, Quotient and Power Rules

Throughout this chapter let f and g be functions from $\mathbb{R}^d$ to $\mathbb{R}$.

A product rule for the simplex gradient of two functions was introduced by Regis in 2015 [33, Proposition 10]. However, the rule only works when the number of points in the set $\mathcal{X}$ is equal to $d+1$ and under the strong assumption that the matrices $\left(S(\mathcal{X})^T\right)^\dagger$ and $diag\left(g(x_1), \ldots, g(x_k)\right)$ commute. We begin by providing a product rule that can be used regardless of the number of points $k+1$ in the sample set $\mathcal{X}$ (obviously we need $k \geq 1$ for the matrix S and the vector δ_f to be well-defined). Thereafter, we introduce a product rule for n functions, a quotient rule and a power rule that also work regardless of the number of elements in the sample set $\mathcal{X}$.

First, let us define the *product difference vector*.

Definition 3.1 (Product Difference Vector)**.** Let $\mathcal{X} = \langle x_0, x_1, \ldots, x_k \rangle$ be an ordered set of $k+1$ points in $\mathbb{R}^d$. The product difference vector of f and g over $\mathcal{X}$ is

$$\delta_{f|g} = \delta_{f|g}(\mathcal{X}) = \begin{bmatrix} (f(x_1) - f(x_0))(g(x_1) - g(x_0)) \\ (f(x_2) - f(x_0))(g(x_2) - g(x_0)) \\ \vdots \\ (f(x_k) - f(x_0))(g(x_k) - g(x_0)) \end{bmatrix}.$$

Note that the product difference vector is the componentwise multiplication $\delta_f \odot \delta_g = \delta_{f|g}$. The product difference vector allows us to create a product rule for the generalized simplex gradient of fg over $\mathcal{X}$. The resulting rule is structurally similar to the product rule for the true gradient plus a term denoted E_{fg}. Without further ado, let us introduce the product rule.

Theorem 3.2 (Product Rule)**.** *Let $\mathcal{X} = \langle x_0, x_1, \ldots, x_k \rangle$ be an ordered set of $k+1$ points in $\mathbb{R}^d$. Then*

$$\nabla_s (fg)(\mathcal{X}) = f(x_0)\nabla_s g(\mathcal{X}) + g(x_0)\nabla_s f(\mathcal{X}) + E_{fg},$$

where $E_{fg} = \left(S^T\right)^\dagger \delta_{f|g}$.

Proof. We have

$$\begin{aligned}
\nabla_s(fg)(\mathcal{X}) &= \left(S^T\right)^\dagger \delta_{fg} \\
&= \left(S^T\right)^\dagger \begin{bmatrix} f(x_1)g(x_1) - f(x_0)g(x_0) \\ \vdots \\ f(x_k)g(x_k) - f(x_0)g(x_0) \end{bmatrix} \\
&= \left(S^T\right)^\dagger \left(\begin{bmatrix} f(x_0)g(x_1) - f(x_0)g(x_0) \\ \vdots \\ f(x_0)g(x_k) - f(x_0)g(x_0) \end{bmatrix} + \begin{bmatrix} f(x_1)g(x_0) - f(x_0)g(x_0) \\ \vdots \\ f(x_k)g(x_0) - f(x_0)g(x_0) \end{bmatrix} \right. \\
&\qquad \left. + \begin{bmatrix} f(x_1)g(x_1) + f(x_0)g(x_0) - f(x_0)g(x_1) - f(x_1)g(x_0) \\ \vdots \\ f(x_k)g(x_k) + f(x_0)g(x_0) - f(x_0)g(x_k) - f(x_k)g(x_0) \end{bmatrix} \right) \\
&= f(x_0) \left(S^T\right)^\dagger \begin{bmatrix} g(x_1) - g(x_0) \\ \vdots \\ g(x_k) - g(x_0) \end{bmatrix} + g(x_0) \left(S^T\right)^\dagger \begin{bmatrix} f(x_1) - f(x_0) \\ \vdots \\ f(x_k) - f(x_0) \end{bmatrix} \\
&\qquad + \left(S^T\right)^\dagger \begin{bmatrix} (f(x_1) - f(x_0))(g(x_1) - g(x_0)) \\ \vdots \\ (f(x_k) - f(x_0))(g(x_k) - g(x_0)) \end{bmatrix} \\
&= f(x_0)\nabla_s g(\mathcal{X}) + g(x_0)\nabla_s f(\mathcal{X}) + \left(S^T\right)^\dagger \delta_{f|g} \ .
\end{aligned}$$

□

Notice that the product rule is *symmetric*, in the sense that the formula for $\nabla_s(fg)(\mathcal{X})$ is identical to the formula created from $\nabla_s(gf)(\mathcal{X})$[1]. Also, we point out that, in the specific case where $k = d$, the product rule presented in this thesis does not have the same structure than the product rule for simplex gradients introduced by Regis in [33].

The product rule immediately produces the following corollary for the generalized simplex gradient of f^n over $\mathcal{X}$. Note that, in this thesis, we define 0^0 to be equal to 1.

[1] We mention this as the product rule presented in [33, Proposition 10] does not share this property.

Corollary 3.3 (Power Rule for a Positive Integer Exponent)**.**

Let $\mathcal{X} = \langle x_0, x_1, \dots, x_k \rangle$ be an ordered set of $k+1$ points in $\mathbb{R}^d$ and $n \in \mathbb{N}_+$. Then

$$\nabla_s f^n(\mathcal{X}) = n[f(x_0)]^{n-1} \nabla_s f(\mathcal{X}) + E_{f^n},$$

where

$$E_{f^n} = \left(S^T\right)^\dagger \left(\sum_{i=1}^{n-1} [f(x_0)]^{n-1-i} \delta_{f|f^i} \right).$$

Proof. We prove this by induction on n. When $n = 1$, we have

$$\begin{aligned}\nabla_s f(\mathcal{X}) &= 1 \nabla_s f(\mathcal{X}) + 0 \\ &= 1[f(x_0)]^0 \nabla_s f(\mathcal{X}) + \left(S^T\right)^\dagger \left(\sum_{i=1}^{0} [f(x_0)]^{-1} \delta_{f|f^1} \right),\end{aligned}$$

since an empty sum is equal to zero.

Next, assume the equation is true for $n = \ell$ for some integer $\ell \geq 1$. Considering $\nabla_s f^{\ell+1}(\mathcal{X})$, we see that

$$\begin{aligned}\nabla_s f^{\ell+1}(\mathcal{X}) &= \nabla_s \left(f^\ell(\mathcal{X}) f(\mathcal{X}) \right) \\ &= [f(x_0)]^\ell \nabla_s f(\mathcal{X}) + f(x_0) \nabla_s f^\ell(\mathcal{X}) + \left(S^T\right)^\dagger \delta_{f|f^\ell} \\ &= [f(x_0)]^\ell \nabla_s f(\mathcal{X}) \\ &\quad + f(x_0) \left(\ell [f(x_0)]^{\ell-1} \nabla_s f(\mathcal{X}) + \left(S^T\right)^\dagger \sum_{i=1}^{\ell-1} [f(x_0)]^{\ell-1-i} \delta_{f|f^i} \right) + \left(S^T\right)^\dagger \delta_{f|f^\ell} \\ &= [f(x_0)]^\ell \nabla_s f(\mathcal{X}) + \ell [f(x_0)]^\ell \nabla_s f(\mathcal{X}) \\ &\quad + \left(S^T\right)^\dagger f(x_0) \sum_{i=1}^{\ell-1} [f(x_0)]^{\ell-1-i} \delta_{f|f^i} + \left(S^T\right)^\dagger \delta_{f|f^\ell} \\ &= (\ell+1)[f(x_0)]^\ell \nabla_s f(\mathcal{X}) + \left(S^T\right)^\dagger \left(\sum_{i=1}^{\ell-1} [f(x_0)]^{\ell-i} \delta_{f|f^i} + \delta_{f|f^\ell} \right) \\ &= (\ell+1)[f(x_0)]^\ell \nabla_s f(\mathcal{X}) + \left(S^T\right)^\dagger \left(\sum_{i=1}^{\ell} [f(x_0)]^{\ell-i} \delta_{f|f^i} \right).\end{aligned}$$

Hence, the equation is also true for $n = \ell+1$. The induction is complete. □

Next, we extend the product rule for the general case of n functions. To prove the product rule for the simplex gradient of n functions, we require the following Lemma 3.4.

Lemma 3.4. *Let $f_i : \mathbb{R}^d \to \mathbb{R}$ for all $i \in \{1, 2\}$. Then*

$$\delta_{f_1 f_2} = f_1(x_0)\delta_{f_2} + f_2(x_0)\delta_{f_1} + \delta_{f_1|f_2}.$$

Proof. We have

$$\begin{aligned}
\delta_{f_1 f_2} &= \begin{bmatrix} (f_1 f_2)(x_1) - (f_1 f_2)(x_0) \\ \vdots \\ (f_1 f_2)(x_k) - (f_1 f_2)(x_0) \end{bmatrix} \\
&= \begin{bmatrix} (f_1 f_2)(x_1) - (f_1 f_2)(x_0) + f_1(x_0) f_2(x_1) - f_1(x_0) f_2(x_1) + f_1(x_1) f_2(x_0) - f_1(x_1) f_2(x_0) \\ \vdots \\ (f_1 f_2)(x_k) - (f_1 f_2)(x_0) + f_1(x_0) f_2(x_k) - f_1(x_0) f_2(x_k) + f_1(x_k) f_2(x_0) - f_1(x_k) f_2(x_0) \end{bmatrix} \\
&= f_1(x_0) \begin{bmatrix} f_2(x_1) - f_2(x_0) \\ \vdots \\ f_2(x_k) - f_2(x_0) \end{bmatrix} + f_2(x_0) \begin{bmatrix} f_1(x_1) - f_1(x_0) \\ \vdots \\ f_1(x_k) - f_1(x_0) \end{bmatrix} \\
&\quad + \begin{bmatrix} \left(f_1(x_1) - f_1(x_0)\right)\left(f_2(x_1) - f_2(x_0)\right) \\ \vdots \\ \left(f_1(x_k) - f_1(x_0)\right)\left(f_2(x_k) - f_2(x_0)\right) \end{bmatrix} \\
&= f_1(x_0)\delta_{f_2} + f_2(x_0)\delta_{f_1} + \delta_{f_1|f_2}.
\end{aligned}$$

□

Proposition 3.5 (Product Rule for n Functions)**.** *Let $f_i : \mathbb{R}^d \to \mathbb{R}$ for all $i \in \{1, 2, \ldots, n\}$ where $n \geq 2$. Then*

$$\nabla_s \left(f_1 f_2 \cdots f_n\right)(\mathcal{X}) = \sum_{i=1}^{n} \left(\prod_{j \neq i} f_j(x_0) \right) \nabla_s f_i(\mathcal{X}) + E_{f_1 f_2 \cdots f_n},$$

where

$$E_{f_1 f_2 \cdots f_n} = \left(S^T\right)^{\dagger} \left(\delta_{f_1 f_2 \cdots f_n} - \sum_{i=1}^{n} \left(\prod_{j \neq i} f_j(x_0) \right) \delta_{f_i} \right).$$

Proof. We prove this by induction on n. When $n = 2$, using Lemma 3.4, we have

$$\begin{aligned}
\nabla_s \left(f_1 f_2\right)(\mathcal{X}) &= f_2(x_0)\nabla_s f_1(\mathcal{X}) + f_1(x_0)\nabla_s f_2(\mathcal{X}) + \left(S^T\right)^{\dagger} \delta_{f_1|f_2} \\
&= f_2(x_0)\nabla_s f_1(\mathcal{X}) + f_1(x_0)\nabla_s f_2(\mathcal{X}) \\
&\qquad + \left(S^T\right)^{\dagger} \left(\delta_{f_1 f_2} - f_1(x_0)\delta_{f_2} - f_2(x_0)\delta_{f_1}\right) \\
&= \sum_{i=1}^{2} \left(\prod_{j\neq i} f_j(x_0)\right) \nabla_s f_i(\mathcal{X}) + \left(S^T\right)^{\dagger} \left(\delta_{f_1 f_2} - \sum_{i=1}^{2} \left(\prod_{j\neq i} f_j(x_0)\right) \delta_{f_i}\right).
\end{aligned}$$

Next, suppose the equation is true for $n = \ell$ for some integer $\ell \geq 2$.

Define $g = f_1 f_2 \cdots f_n$. Considering $\nabla_s \left(f_1 f_2 \cdots f_n f_{n+1}\right)(\mathcal{X})$, we see that

$$\begin{aligned}
\nabla_s \left(f_1 f_2 \cdots f_{n+1}\right)(\mathcal{X}) &= \nabla_s \left(g f_{n+1}\right)(\mathcal{X}) \\
&= f_{n+1}(x_0)\nabla_s g(\mathcal{X}) + g(x_0)\nabla_s f_{n+1}(\mathcal{X}) + \left(S^T\right)^{\dagger} \delta_{g|f_{n+1}} \\
&= f_{n+1}(x_0) \left(\sum_{i=1}^{n} \left(\prod_{j\neq i} f_j(x_0)\right) \nabla_s f_i(\mathcal{X})\right. \\
&\qquad \left. + \, E_{f_1 f_2 \cdots f_n}\right) + g(x_0)\nabla_s f_{n+1}(\mathcal{X}) + \left(S^T\right)^{\dagger} \delta_{g|f_{n+1}} \\
&= \left(\sum_{i=1}^{n+1} \left(\prod_{j\neq i} f_j(x_0)\right) \nabla_s f_i(\mathcal{X})\right) \\
&\qquad + f_{n+1}(x_0) E_{f_1 f_2 \cdots f_n} + \left(S^T\right)^{\dagger} \delta_{g|f_{n+1}}.
\end{aligned}$$

To complete the proof, we must show

$$f_{n+1}(x_0) E_{f_1 f_2 \cdots f_n} + \left(S^T\right)^{\dagger} \delta_{g|f_{n+1}} = E_{f_1 f_2 \cdots f_n f_{n+1}}.$$

Indeed, we have

$$\begin{aligned}
& f_{n+1}(x_0) E_n + \left(S^T\right)^{\dagger} \delta_{g|f_{n+1}} \\
&\quad = f_{n+1}(x_0) \left(S^T\right)^{\dagger} \left(\delta_g - \sum_{i=1}^{n} \left(\prod_{j\neq i} f_j(x_0)\right) \delta_{f_i}\right) + \left(S^T\right)^{\dagger} \delta_{g|f_{n+1}} \\
&\quad = \left(S^T\right)^{\dagger} \left[f_{n+1}(x_0)\delta_g - f_{n+1}(x_0) \left(\sum_{i=1}^{n} \left(\prod_{j\neq i} f_j(x_0)\right) \delta_{f_i}\right) + \delta_{g|f_{n+1}}\right] \\
&\quad = \left(S^T\right)^{\dagger} \left[f_{n+1}(x_0)\delta_g + g(x_0)\delta_{f_{n+1}} + \delta_{g|f_{n+1}} - \left(\sum_{i=1}^{n+1} \left(\prod_{j\neq i} f_j(x_0)\right) \delta_{f_i}\right)\right]
\end{aligned}$$

$$= \left(S^T\right)^\dagger \left(\delta_{f_1\cdots f_n f_{n+1}} - \left(\sum_{i=1}^{n+1}\left(\prod_{j\neq i} f_j(x_0)\right)\delta_{f_i}\right)\right) \quad \text{(by Lemma 3.4)}$$
$$= E_{f_1 f_2\cdots f_n f_{n+1}}.$$

Therefore, the equation is true for $n = \ell + 1$ and the induction is complete. □

The following Corollary 3.6 presents an alternative formula whenever all f_i for $i \in \{1, 2, \ldots, n\}$ are equal.

Corollary 3.6. *Let $f : \mathbb{R}^d \to \mathbb{R}$ and $n \in \mathbb{N}_+$. Then*

$$\nabla_s f^n(\mathcal{X}) = n[f(x_0)]^{n-1}\nabla_s f(\mathcal{X}) + E_{f^n},$$

where

$$E_{f^n} = \left(S^T\right)^\dagger \left(\delta_{f^n} - n[f(x_0)]^{n-1}\delta_f\right).$$

Proof. The result is obtained easily by letting $f_i = f$ for all $i \in \{1, 2, \ldots, n\}$ in Proposition 3.5. □

Note that Corollary 3.3 and Corollary 3.6 provide two different formulas to calculate $\nabla_s f^n(\mathcal{X})$. Let us show that there are equivalent.

Lemma 3.7. *Let $\mathcal{X} = \langle x_0, x_1, \ldots, x_k\rangle$ be an ordered set of $k + 1$ points in $\mathbb{R}^d$ and $n \in \mathbb{N}_+$. Then*

$$\sum_{i=1}^{n-1}[f(x_0)]^{n-1-i}\delta_{f|f^i} = \delta_{f^n} - n[f(x_0)]^{n-1}\delta_f.$$

Proof. We have

$$\begin{aligned}\sum_{i=1}^{n-1}[f(x_0)]^{n-1-i}\delta_{f|f^i} &= [f(x_0)]^{n-2}\delta_{f|f} + [f(x_0)]^{n-3}\delta_{f|f^2}\\ &\quad + \cdots + f(x_0)\delta_{f|f^{n-2}} + \delta_{f|f^{n-1}}\\ &= [f(x_0)]^{n-2}\left(\delta_{f^2} - f(x_0)\delta_f - f(x_0)\delta_f\right)\\ &\quad + [f(x_0)]^{n-3}\left(\delta_{f^3} - f(x_0)\delta_{f^2} - [f(x_0)]^2\delta_f\right)\\ &\quad + \cdots + f(x_0)\left(\delta_{f^{n-1}} - f(x_0)\delta_{f^{n-2}} - [f(x_0)]^{n-2}\delta_f\right)\\ &\quad + \delta_{f^n} - f(x_0)\delta_{f^{n-1}} - [f(x_0)]^{n-1}\delta_f\end{aligned}$$

by Lemma 3.4. After cancelations we get

$$\begin{aligned}\sum_{i=1}^{n-1}[f(x_0)]^{n-1-i}\delta_{f|f^i} &= \underbrace{-[f(x_0)]^{n-1}\delta_f - [f(x_0)]^{n-1}\delta_f - \cdots - [f(x_0)]^{n-1}\delta_f}_{n \text{ times}} + \delta_{f^n}\\ &= -n[f(x_0)]^{n-1}\delta_f + \delta_{f^n}.\end{aligned}$$

□

Aware of the product rule, we can develop a quotient rule for generalized simplex gradients.

Theorem 3.8 (Quotient Rule)**.** *Let $\mathcal{X} = \langle x_0, x_1, \ldots, x_k\rangle$ be an ordered set of $k+1$ points in $\mathbb{R}^d$ for which $g(x_0)$, $g(x_1)$, …, $g(x_k)$ are all nonzero. Then*

$$\nabla_s\left(\frac{f}{g}\right)(\mathcal{X}) = \frac{g(x_0)\nabla_s f(\mathcal{X}) - f(x_0)\nabla_s g(\mathcal{X})}{[g(x_0)]^2} - E_{\frac{f}{g}},$$

where

$$E_{\frac{f}{g}} = \frac{\left(S^T\right)^\dagger}{g(x_0)}\delta_{\frac{f}{g}|g}.$$

Proof. By the product rule, we have

$$\begin{aligned}\nabla_s f(\mathcal{X}) &= \nabla_s\left(\frac{f}{g}\cdot g\right)(\mathcal{X})\\ &= \left(\frac{f}{g}\right)(x_0)\nabla_s g(\mathcal{X}) + g(x_0)\nabla_s\left(\frac{f}{g}\right)(\mathcal{X}) + \left(S^T\right)^\dagger\delta_{\frac{f}{g}|g}.\end{aligned}$$

Solving for $\nabla_s\left(\frac{f}{g}\right)(\mathcal{X})$ gives

$$\nabla_s\left(\frac{f}{g}\right)(\mathcal{X}) = \frac{g(x_0)\nabla_s f(\mathcal{X}) - f(x_0)\nabla_s g(\mathcal{X})}{[g(x_0)]^2} - \frac{\left(S^T\right)^\dagger}{g(x_0)}\delta_{\frac{f}{g}|g}.$$

□

Note that Theorem 3.8 requires $g(x_0) \neq 0, g(x_1) \neq 0, \ldots, g(x_k) \neq 0$. This is needed to ensure $\delta_{\frac{f}{g}|g}$ does not include any division by zero. The quotient rule introduced by Regis in 2015 also requires this assumption [33, Proposition 11].

The following corollary is used to prove the power rule for a negative integer exponent.

Corollary 3.9. *Let $\mathcal{X} = \langle x_0, x_1, \ldots, x_k \rangle$ be an ordered set of $k+1$ points in $\mathbb{R}^d$ for which $f(x_0)$, $f(x_1)$, $\ldots$, $f(x_k)$ are all nonzero. Then*

$$\nabla_s \left(\frac{1}{f} \right) (\mathcal{X}) = -\frac{\nabla_s f(\mathcal{X})}{[f(x_0)]^2} - \frac{\left(S^T\right)^\dagger}{f(x_0)} \delta_{\frac{1}{f}|f}.$$

Finally, we conclude this chapter by presenting the power rule for a negative integer exponent.

Proposition 3.10 (Power Rule for a Negative Integer Exponent)**.**

Let $\mathcal{X} = \langle x_0, x_1, \ldots, x_k \rangle$ be an ordered set of $k+1$ points in $\mathbb{R}^d$ for which $f(x_0)$, $f(x_1)$, …, $f(x_k)$ are all nonzero and $n \in \mathbb{N}_+$. Then

$$\nabla_s f^{-n}(\mathcal{X}) = n[f(x_0)]^{-n+1} \nabla_s \left(\frac{1}{f}\right)(\mathcal{X}) + \left(S^T\right)^\dagger \left(\sum_{i=1}^{n-1} [f(x_0)]^{-n+1+i} \delta_{f^{-1}|f^{-i}}\right).$$

It follows that

$$\nabla_s f^{-n}(\mathcal{X}) = -n[f(x_0)]^{-n-1} \nabla_s f(\mathcal{X}) - E_{f^{-n}},$$

where

$$E_{f^{-n}} = \frac{\left(S^T\right)^\dagger}{[f(x_0)]^n} \left(n\delta_{\frac{1}{f}|f} - \sum_{i=1}^{n-1} [f(x_0)]^{1+i} \delta_{f^{-1}|f^{-i}}\right).$$

Proof. By the power rule for a positive integer exponent,

$$\begin{aligned}
\nabla_s \left(\frac{1}{f}\right)^n (\mathcal{X}) &= n\left(\left(\frac{1}{f}\right)(x_0)\right)^{n-1} \nabla_s \left(\frac{1}{f}\right)(\mathcal{X}) \\
&\quad + \left(S^T\right)^\dagger \left(\sum_{i=1}^{n-1} \left(\left(\frac{1}{f}\right)(x_0)\right)^{n-1-i} \delta_{f^{-1}|f^{-i}}\right) \\
&= n[f(x_0)]^{-n+1} \nabla_s \left(\frac{1}{f}\right)(\mathcal{X}) + \left(S^T\right)^\dagger \left(\sum_{i=1}^{n-1} [f(x_0)]^{-n+1+i} \delta_{f^{-1}|f^{-i}}\right)
\end{aligned}$$

which proves our first claim.

By Corollary 3.9,

$$\begin{aligned}
\nabla_s f^{-n}(\mathcal{X}) &= n[f(x_0)]^{-n+1} \left(\frac{-\nabla_s f(\mathcal{X})}{[f(x_0)]^2} - \frac{\left(S^T\right)^\dagger}{f(x_0)} \delta_{\frac{1}{f}|f}\right) \\
&\quad + \left(S^T\right)^\dagger \left(\sum_{i=1}^{n-1} [f(x_0)]^{-n+1+i} \delta_{f^{-1}|f^{-i}}\right) \\
&= -n[f(x_0)]^{-n-1} \nabla_s f(\mathcal{X}) - \frac{n\left(S^T\right)^\dagger \delta_{\frac{1}{f}|f}}{[f(x_0)]^n}
\end{aligned}$$

$$
\begin{aligned}
&+ \left(S^T\right)^\dagger \left(\sum_{i=1}^{n-1}[f(x_0)]^{-n+1+i}\delta_{f^{-1}|f^{-i}}\right) \\
&= -n[f(x_0)]^{-n-1}\nabla_s f(\mathcal{X}) - \frac{\left(S^T\right)^\dagger}{[f(x_0)]^n}\left(n\delta_{\frac{1}{f}|f} - \sum_{i=1}^{n-1}[f(x_0)]^{1+i}\delta_{f^{-1}|f^{-i}}\right).
\end{aligned}
$$

□

We now turn out attention to a more advanced calculus rule, the chain rule.

Chapter 4

Chain Rule

Throughout this chapter, let $f : \mathbb{R}^p \to \mathbb{R}$ and $g : \mathbb{R}^d \to \mathbb{R}^p$, where

$$g(y) = \begin{bmatrix} g_1(y) \\ g_2(y) \\ \vdots \\ g_p(y) \end{bmatrix} \in \mathbb{R}^p.$$

Let $\mathcal{X} = \langle x_0, x_1, \ldots, x_k \rangle$ be an ordered set of $k+1$ points in $\mathbb{R}^d$ and define

$$g(\mathcal{X}) = \langle g(x_0), g(x_1), \ldots, g(x_k) \rangle$$

to be an ordered set of $k+1$ points in $\mathbb{R}^p$.

Before inaugurating the chain rule, let us present some matrices involved in the formula. Note that

$$S(g(\mathcal{X})) = \begin{bmatrix} g(x_1) - g(x_0) & \ldots & g(x_k) - g(x_0) \end{bmatrix} \in \mathbb{R}^{p \times k}$$

and

$$\begin{aligned} \delta_f(g(\mathcal{X})) &= \begin{bmatrix} f\left(g(x_1)\right) - f\left(g(x_0)\right) \\ \vdots \\ f\left(g(x_k)\right) - f\left(g(x_0)\right) \end{bmatrix} \\ &= \begin{bmatrix} (f \circ g)(x_1) - (f \circ g)(x_0) \\ \vdots \\ (f \circ g)(x_k) - (f \circ g)(x_0) \end{bmatrix} \\ &= \delta_{f \circ g}(\mathcal{X}) \in \mathbb{R}^k. \end{aligned}$$

In the next definition, we define the *generalized simplex Jacobian of g over $\mathcal{X}$* in order to make the notation tighter.

Definition 4.1 (Generalized Simplex Jacobian Matrix)**.** Let $g : \mathbb{R}^d \to \mathbb{R}^p : y \mapsto \begin{bmatrix} g_1(y) & g_2(y) & \dots & g_p(y) \end{bmatrix}^T$. Then the generalized simplex Jacobian $\mathbf{J_s}$ of g over $\mathcal{X}$ is a $p \times d$ real matrix defined as

$$\mathbf{J_s} g(\mathcal{X}) = \begin{bmatrix} \nabla_s g_1(\mathcal{X})^T \\ \nabla_s g_2(\mathcal{X})^T \\ \vdots \\ \nabla_s g_p(\mathcal{X})^T \end{bmatrix}.$$

Theorem 4.2 (Chain Rule)**.** *Let $f : \mathbb{R}^p \to \mathbb{R}$ and $g : \mathbb{R}^d \to \mathbb{R}^p$. Then*

$$\nabla_s (f \circ g)(\mathcal{X}) = (\mathbf{J_s} g(\mathcal{X}))^T \nabla_s f(g(\mathcal{X})) - E_{f \circ g},$$

where

$$E_{f \circ g} = \left(S(\mathcal{X})^T\right)^\dagger \left(S\left(g(\mathcal{X})\right)^T \left(S(g(\mathcal{X}))^T\right)^\dagger - \mathtt{I}_k\right) \delta_f(g(\mathcal{X})).$$

Proof. We have

$$\begin{aligned} \nabla_s (f \circ g)(\mathcal{X}) &= \left(S(\mathcal{X})^T\right)^\dagger \delta_{f \circ g}(\mathcal{X}) \\ &= \left(S(\mathcal{X})^T\right)^\dagger \left(S(g(\mathcal{X}))^T \left(S(g(\mathcal{X}))^T\right)^\dagger - \widehat{E}\right) \delta_{f \circ g}(\mathcal{X}), \end{aligned}$$

where

$$\widehat{E} = S(g(\mathcal{X}))^T \left(S(g(\mathcal{X}))^T\right)^\dagger - \mathtt{I}_k.$$

In order to make the notation tighter, let $\mathcal{Y} = g(\mathcal{X})$.

Now, using $\delta_{f \circ g}(\mathcal{X}) = \delta_f\left(g(\mathcal{X})\right) = \delta_f(\mathcal{Y})$, we find

$$\begin{aligned} \nabla_s (f \circ g)(\mathcal{X}) &= \left(\left(S(\mathcal{X})^T\right)^\dagger S(\mathcal{Y})^T \left(S(\mathcal{Y})^T\right)^\dagger - \left(S(\mathcal{X})^T\right)^\dagger \widehat{E}\right) \delta_f(\mathcal{Y}) \\ &= \left(S(\mathcal{X})^T\right)^\dagger S(\mathcal{Y})^T \left(S(\mathcal{Y})^T\right)^\dagger \delta_f(\mathcal{Y}) - \left(S(\mathcal{X})^T\right)^\dagger \widehat{E}\, \delta_f(\mathcal{Y}). \end{aligned}$$

Notice that $S(\mathcal{Y})^T = \begin{bmatrix} (g(x_1) - g(x_0))^T \\ \vdots \\ (g(x_k) - g(x_0))^T \end{bmatrix} = \delta_g(\mathcal{X})$.

Hence,

$$\begin{aligned}
\nabla_s (f \circ g)(\mathcal{X}) &= \left[\nabla_s g_1(\mathcal{X}) \quad \nabla_s g_2(\mathcal{X}) \quad \dots \quad \nabla_s g_p(\mathcal{X})\right] \nabla_s f(\mathcal{Y}) - \left(S(\mathcal{X})^T\right)^{\dagger} \widehat{E}\, \delta_f(\mathcal{Y}) \\
&= \begin{bmatrix} (\nabla_s g_1(\mathcal{X}))^T \\ (\nabla_s g_2(\mathcal{X}))^T \\ \dots \\ (\nabla_s g_p(\mathcal{X}))^T \end{bmatrix}^T \nabla_s f(\mathcal{Y}) - \left(S(\mathcal{X})^T\right)^{\dagger} \left(S(\mathcal{Y})^T \left(S(\mathcal{Y})^T\right)^{\dagger} - \mathtt{I}_k\right) \delta_f(\mathcal{Y}) \\
&= \left(\mathbf{J_s} g(\mathcal{X})\right)^T \nabla_s f(g(\mathcal{X})) - E_{f \circ g}.
\end{aligned}$$

□

The next corollary demonstrates that the term $E_{f\circ g}$ vanishes whenever $k \leq p$ and $S(g(\mathcal{X}))$ has full rank.

Corollary 4.3. *Let $f : \mathbb{R}^p \to \mathbb{R}$ and $g : \mathbb{R}^d \to \mathbb{R}^p$. Let $\mathcal{X} = \langle x_0, x_1, \dots, x_k \rangle$ be an ordered set of $k+1$ points in $\mathbb{R}^d$ and $g(\mathcal{X}) = \langle g(x_0),\, g(x_1),\, \dots,\, g(x_k) \rangle$ be an ordered set of $k+1$ points in $\mathbb{R}^p$. Suppose $S(g(\mathcal{X}))$ has full column rank (i.e. $k \leq p$ and $S\left(g(\mathcal{X})\right)$ has full rank). Then*

$$\nabla_s (f \circ g)(\mathcal{X}) = \left(\mathbf{J_s} g(\mathcal{X})\right)^T \nabla_s f(g(\mathcal{X})).$$

Proof. Since $S(g(\mathcal{X}))$ has full column rank, $S(g(\mathcal{X}))^T$ has full row rank. This implies that $S(g(\mathcal{X}))^T$ has right inverse $\left(S(g(\mathcal{X}))^T\right)^{\dagger} \in \mathbb{R}^{p \times k}$. Thus

$$\begin{aligned}
E &= \left(S(\mathcal{X})^T\right)^{\dagger} \left(S(g(\mathcal{X}))^T \left(S(g(\mathcal{X}))^T\right)^{\dagger} - \mathtt{I}_k\right) \delta_f(g(\mathcal{X})) \\
&= \left(S(\mathcal{X})^T\right)^{\dagger} \left(\mathtt{I}_k - \mathtt{I}_k\right) \delta_f(g(\mathcal{X})) \\
&= 0.
\end{aligned}$$

Therefore, $\nabla_s (f \circ g)(\mathcal{X}) = \left(\mathbf{J_s} g(\mathcal{X})\right)^T \nabla_s f(g(\mathcal{X}))$. □

We now have presented all the calculus rules. We see that all of them have the same structure as the calculus rules for the true gradient plus a term E that can be viewed as an error term. Indeed, the term E can be viewed as an error term in the sense that, when all the functions involved are linear, removing the term E from the formulas presented in Chapter 3 and Section 4 can provide an exact approximation of the true gradient.

However, in general, removing the term E from the calculus rules does not imply we get the value of the true gradient.

In Chapter 5, we explore the effects of removing the term E in the formulas. Removing E from the formulas leads to a new approach to gradient approximation in situations where the calculus rules could be applied. Moreover, removing the term E in the quotient rule and the power rule for a negative exponent allows us to remove some assumptions on the function values of the sample set $\mathcal{X}$. Therefore, this new approach can also be applied in situations where the calculus rules were not applicable.

Chapter 5

Generalized Simplex Calculus Gradient

First of all, let us summarize the work achieved so far and present all the calculus rules from Chapter 3 and Chapter 4 in Table 1.

Let us introduce new notation. We employ $\nabla_{sp}(\cdot)(\mathcal{X}), \nabla_{sq}(\cdot)(\mathcal{X})$ and $\nabla_{sc}(\cdot)(\mathcal{X})$ to denote the product rule, quotient rule and chain rule respectively that do not include the term E. We also use $\nabla_{sp}(\cdot)(\mathcal{X})$ for the power rule since it is a particular case of the product rule. We refer to these new approaches to approximate gradients as the generalized simplex calculus gradients . We formally define these in equations (5.1), (5.3),(5.4), and (5.2) and present an overview of these rules in Table 2.

The purpose of this section is to compare both approaches: the generalized simplex calculus gradient and the generalized simplex gradient.

Throughout this chapter, let $\widehat{S}(\mathcal{X}) = S(\mathcal{X})/\Delta$.

In this chapter, we provide error bounds for generalized simplex calculus gradients and examples where $\nabla_{sp}(\cdot)(\mathcal{X}), \nabla_{sq}(\cdot)(\mathcal{X}), \nabla_{sc}(\cdot)(\mathcal{X})$ are more accurate than the generalized simplex gradient $\nabla_s(\cdot)(\mathcal{X})$ and vice versa. We begin by showing that when $S(\mathcal{X})$ has full row rank, the generalized simplex

Table 1: The Calculus Rules for Generalized Simplex Gradients

Rule	**Formula**	**E**
Product fg	$f(x_0)\nabla_s g(\mathcal{X}) + g(x_0)\nabla_s f(\mathcal{X}) + E_{fg}$	$\left(S^T\right)^{\dagger} \delta_{f\mid g}$
Product $f_1 \cdots f_n$	$\sum_{i=1}^n \left(\prod_{j\neq i} f_j(x_0)\right) \nabla_s f_i(\mathcal{X}) + E_{f_1\cdots f_n}$	$\left(S^T\right)^{\dagger} \left(\delta_{f_1\cdots f_n} - \sum_{i=1}^n \left(\prod_{j\neq i} f_j(x_0)\right) \delta_{f_i}\right)$
Positive power	$n[f(x_0)]^{n-1}\nabla_s f(\mathcal{X}) + E_{f^n}$	$\left(S^T\right)^{\dagger} \left(\sum_{i=1}^{n-1} [f(x_0)]^{n-1-i} \delta_{f\mid f^i}\right)$
Negative power	$-n[f(x_0)]^{-n-1}\nabla_s f(\mathcal{X}) - E_{f^{-n}}$	$\frac{\left(S^T\right)^{\dagger}}{[f(x_0)]^n} \left(n\delta_{\frac{1}{f}\mid f} - \sum_{i=1}^{n-1} [f(x_0)]^{1+i} \delta_{f^{-1}\mid f^{-i}}\right)$
Quotient	$\frac{g(x_0)\nabla_s f(\mathcal{X}) - f(x_0)\nabla_s g(\mathcal{X})}{[g(x_0)]^2} - E_{\frac{f}{g}}$	$\frac{\left(S^T\right)^{\dagger}}{g(x_0)} \delta_{\frac{f}{g}\mid g}$
Chain	$\left(\mathbf{J_s} g(\mathcal{X})\right)^T \nabla_s f(g(\mathcal{X})) - E_{f\circ g}$	$\left(S(\mathcal{X})^T\right)^{\dagger} \left(S\left(g(\mathcal{X})\right)^T \left(S(g(\mathcal{X}))^T\right)^{\dagger} - \mathtt{I}_k\right) \delta_f(g(\mathcal{X}))$

Table 2: Generalized Simplex Calculus Gradients

Rule	**Formula**	**Relation to $\nabla_s(\cdot)(\mathcal{X})$**
Product fg	$\nabla_{sp}(fg)(\mathcal{X}) = f(x_0)\nabla_s g(\mathcal{X}) + g(x_0)\nabla_s f(\mathcal{X})$	$\nabla_s(\cdot)(\mathcal{X}) = \nabla_{sp}(\cdot)(\mathcal{X}) + E_{(\cdot)}$
Product $f_1 \cdots f_n$	$\nabla_{sp}(f_1 \cdots f_n)(\mathcal{X}) = \sum_{i=1}^{n} \left(\prod_{j \neq i} f_j(x_0) \right) \nabla_s f_i(\mathcal{X})$	$\nabla_s(\cdot)(\mathcal{X}) = \nabla_{sp}(\cdot)(\mathcal{X}) + E_{(\cdot)}$
Positive power	$\nabla_{sp}(f^n)(\mathcal{X}) = n[f(x_0)]^{n-1}\nabla_s f(\mathcal{X})$	$\nabla_s(\cdot)(\mathcal{X}) = \nabla_{sp}(\cdot)(\mathcal{X}) + E_{(\cdot)}$
Negative power	$\nabla_{sp}(f^{-n})(\mathcal{X}) = -n[f(x_0)]^{-n-1}\nabla_s f(\mathcal{X})$	$\nabla_s(\cdot)(\mathcal{X}) = \nabla_{sp}(\cdot)(\mathcal{X}) - E_{(\cdot)}$
Quotient	$\nabla_{sq}(f/g)(\mathcal{X}) = \frac{g(x_0)\nabla_s f(\mathcal{X}) - f(x_0)\nabla_s g(\mathcal{X})}{[g(x_0)]^2}$	$\nabla_s(\cdot)(\mathcal{X}) = \nabla_{sq}(\cdot)(\mathcal{X}) - E_{(\cdot)}$
Chain	$\nabla_{sc}(f \circ g)(\mathcal{X}) = (\mathbf{J_s} g(\mathcal{X}))^T \nabla_s f(g(\mathcal{X}))$	$\nabla_s(\cdot)(\mathcal{X}) = \nabla_{sc}(\cdot)(\mathcal{X}) - E_{(\cdot)}$

gradient of a linear function is equal to the true gradient.

Lemma 5.1. *Let $f : \mathbb{R}^d \to \mathbb{R} : y \mapsto a^T y + c$ where $a \in \mathbb{R}^d$ and $c \in \mathbb{R}$. Suppose $S(\mathcal{X})$ has full row rank. Then $\nabla_s f(\mathcal{X}) = \nabla f(x_0)$.*

Proof. We have

$$\begin{aligned}
\nabla_s f(\mathcal{X}) &= \left(S^T\right)^\dagger \begin{bmatrix} f(x_1) - f(x_0) \\ \vdots \\ f(x_k) - f(x_0) \end{bmatrix} \\
&= \left(S^T\right)^\dagger \begin{bmatrix} (a^T x_1 + c) - (a^T x_0 + c) \\ \vdots \\ (a^T x_k + c) - (a^T x_0 + c) \end{bmatrix} \\
&= \begin{bmatrix} (x_1 - x_0)^T \\ \vdots \\ (x_k - x_0)^T \end{bmatrix}^\dagger \begin{bmatrix} (x_1 - x_0)^T \\ \vdots \\ (x_k - x_0)^T \end{bmatrix} a.
\end{aligned}$$

Since S has full row rank, $\left(S^T\right)^\dagger$ is a left inverse. Therefore, $\nabla_s f(\mathcal{X}) = a$. □

We point out that it is not possible for the matrix S to have full row rank in the underdetermined case (rank$(S) < d$). This makes sense since underdetermined simplex gradients do not capture enough information to guaranteed a perfect approximation of $\nabla f(x_0)$. Example 2.15 can be used to illustrate this fact.

5.1 Product Rule

Let us define the *generalized simplex product gradient of fg over $\mathcal{X}$*

$$\nabla_{sp}(fg)(\mathcal{X}) = f(x_0)\nabla_s g(\mathcal{X}) + g(x_0)\nabla_s f(\mathcal{X}). \tag{5.1}$$

The next corollary shows that $\nabla_{sp}(fg)(\mathcal{X})$ is perfectly accurate when f and g are linear functions.

Corollary 5.2. *Let f and g be linear functions from $\mathbb{R}^d$ to $\mathbb{R}$. Suppose $S(\mathcal{X})$ has full row rank. Then*

$$\nabla_{sp}(fg)(\mathcal{X}) = \nabla(fg)(x_0).$$

Proof. This follows from Lemma 5.1. □

Hence, $\nabla_{sp}(fg)(\mathcal{X})$ is always as good or better than $\nabla_s(fg)(\mathcal{X})$ whenever f and g are linear functions and $S(\mathcal{X})$ has full row rank. The following example illustrates this fact.

Example 5.3. Let $f : \mathbb{R}^2 \to \mathbb{R} : y \mapsto y_1 - y_2$ and $g : \mathbb{R}^2 \to \mathbb{R} : y \mapsto y_1 + y_2$. Consider the ordered set $\mathcal{X} = \langle [1\ 1]^T, [2\ 1]^T, [1\ 2]^T \rangle$. We get

$$\begin{aligned}\nabla_{sp}(fg)(\mathcal{X}) &= 0 \cdot \nabla_s g(\mathcal{X}) + 2 \cdot \nabla_s f(\mathcal{X}) \\ &= \begin{bmatrix} 2 \\ -2 \end{bmatrix} \\ &= \nabla(fg)(x_0),\end{aligned}$$

and

$$\begin{aligned}\nabla_s(fg)(\mathcal{X}) &= \begin{bmatrix} 2 \\ -2 \end{bmatrix} + E_{fg} \\ &= \begin{bmatrix} 2 \\ -2 \end{bmatrix} + \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} (1-0)(3-2) \\ (-1-0)(3-2) \end{bmatrix} \\ &= \begin{bmatrix} 3 \\ -3 \end{bmatrix}.\end{aligned}$$

We can extend $\nabla_{sp}(fg)(\mathcal{X})$ to the case of n functions. We get

$$\begin{aligned}\nabla_{sp}(f_1 f_2 \cdots f_n)(\mathcal{X}) = &(f_1 f_2 \cdots f_{n-1})(x_0)\nabla_s f_n(\mathcal{X}) \\ &+ (f_1 f_2 \cdots f_{n-2} f_n)(x_0)\nabla_s f_{n-1}(\mathcal{X}) + \dots \\ &+ (f_1 f_3 f_4 \cdots f_n)(x_0)\nabla_s f_2(\mathcal{X}) \\ &+ (f_2 f_3 \cdots f_n)(x_0)\nabla_s f_1(\mathcal{X}).\end{aligned}$$

Recall that by *the Fundamental Theorem of Algebra*, every single variable degree n polynomial with complex coefficients has, counted with multiplicity, exactly n complex roots [36, Theorem 2.4]. Hence, $\nabla_{sp}(f_1 f_2 ... f_n)(\mathcal{X})$ is an exact approximation whenever $f = f_1 f_2 \cdots f_n$ (where $f_i : \mathbb{R} \to \mathbb{C}, i \in \{1, 2, \ldots, n\}$ are linear functions) is a real polynomial of degree n and $\mathcal{X}$ has full row rank. In the next example, let $\mathfrak{i}$ denote the imaginary number such that $\mathfrak{i}^2 = -1$.

Example 5.4 (Factorizing a Single Variable Polynomial)**.** Let $f : \mathbb{R} \to \mathbb{R} : y \mapsto y^4 - 1$ and $\mathcal{X} = \langle 1, 2 \rangle$. Then

$$\begin{aligned} f &= f_1 f_2 f_3 f_4 \\ &= (y-1)(y+1)(y-\mathfrak{i})(y+\mathfrak{i}). \end{aligned}$$

We get

$$\begin{aligned} \nabla_{sp}(f_1 f_2 f_3 f_4)(\mathcal{X}) &= 0 \cdot \nabla_s f_4(\mathcal{X}) + 0 \cdot \nabla_s f_3(\mathcal{X}) + 0 \cdot \nabla_s f_2(\mathcal{X}) \\ &\quad + 2(1-i)(1+i)\nabla_s f_1(\mathcal{X}) \\ &= 4 \\ &= \nabla f(x_0). \end{aligned}$$

Note that

$$\begin{aligned} \nabla_s f(\mathcal{X}) &= 4 + E_f \\ &= 15. \end{aligned}$$

Once again, the previous example shows the enormous improvement in accuracy that can be made using the generalized simplex product gradient when the functions involved are linear.

Now, let us present an error bound for the generalized simplex product gradient $\nabla_{sp}(fg)(\mathcal{X})$.

Theorem 5.5 (Error Bound for $\nabla_{sp}(fg)(\mathcal{X})$)**.** *Let $\mathcal{X}$ be an ordered set of $k+1$ points in $\mathbb{R}^d$. Assume that $S(\mathcal{X})$ has full row rank, $f, g \in \mathcal{C}^{1+}$ with Lipschitz constant $L_{\nabla f} \geq 0$ and $L_{\nabla g} \geq 0$ in an open domain $\mathcal{O}$ containing* $\operatorname{conv}(\mathcal{X})$. *Define*

$$\varepsilon_{sp}(fg)(\mathcal{X}) = \frac{\sqrt{k}}{2} \left(|f(x_0)| L_{\nabla g} + |g(x_0)| L_{\nabla f} \right) \left\| \left(\widehat{S}\,(\mathcal{X})^T \right)^{\dagger} \right\| \Delta.$$

Then

$$\|\nabla_{sp}(fg)(\mathcal{X}) - \nabla(fg)(x_0)\| \leq \varepsilon_{sp}(fg)(\mathcal{X}).$$

Proof. We have $\|\nabla_{sp}(fg)(\mathcal{X}) - \nabla(fg)(x_0)\|$

$$\begin{aligned}
&= \|f(x_0)\nabla_s g(\mathcal{X}) + g(x_0)\nabla_s f(\mathcal{X}) - (f(x_0)\nabla g(x_0) + g(x_0)\nabla f(x_0))\| \\
&\leq |f(x_0)|\,\|\nabla_s g(\mathcal{X}) - \nabla g(x_0)\| + |g(x_0)|\,\|\nabla_s f(\mathcal{X}) - \nabla f(x_0)\| \\
&\leq \frac{\sqrt{k}}{2}\left(|f(x_0)|L_{\nabla g} + |g(x_0)|L_{\nabla f}\right)\left\|\left(\widehat{S}\,(\mathcal{X})^T\right)^\dagger\right\|\Delta
\end{aligned}$$

by Proposition 2.14. □

Analyzing this error bound, we get the following Corollary 5.6 that provides sufficient conditions to obtain an exact approximation of $\nabla(fg)(x_0)$.

Corollary 5.6. *Let the assumptions of Theorem 5.5 hold. If any of the following cases hold, then*

$$\nabla_{sp}(fg)(\mathcal{X}) = \nabla(fg)(x_0).$$

i. The function values $f(x_0) = 0$ and $g(x_0) = 0$.

ii. The functions f and g are linear.

iii. The function f is linear and $f(x_0) = 0$.

iv. The function g is linear and $g(x_0) = 0$.

We would like to answer the following question: when is the generalized simplex product gradient $\nabla_{sp}(fg)(\mathcal{X})$ more accurate than the generalized simplex gradient $\nabla_s(fg)(\mathcal{X})$?

Comparing their respective error bound gives us a good indicator to answer this question. Recall that the product of Lipschitz continuous functions on a bounded domain Ω is Lipschitz continuous on that domain Ω [16, Section 12.7].

Corollary 5.7. *Let the assumptions of Theorem 5.5 hold and so $\nabla(fg)$ is Lipschitz continuous on $\mathcal{O}$ with Lipschitz constant $L_{\nabla(fg)} \geq 0$. If*

$$|f(x_0)|L_{\nabla g} + |g(x_0)|L_{\nabla f} < L_{\nabla(fg)}$$

then the error bound $\varepsilon_{sp}(fg)(\mathcal{X})$ is smaller than the error bound $\varepsilon_s(fg)(\mathcal{X})$.

In practice, we do not know the value of the Lipschitz constants $L_{\nabla f}, L_{\nabla g}$ and $L_{\nabla(fg)}$. For this reason, it is improbable that we know which gradient approximation has a smaller error bound. A technique to approximate the

Lipschitz constants is presented in Chapter 7. Based on the approximation of the Lipschitz constants, a decision can be made.

Example 5.8 provides an example where the true absolute error for the generalized simplex product gradient $\nabla_{sp}(fg)(\mathcal{X})$ is smaller than $\nabla_s(fg)(\mathcal{X})$ and Example 5.9 provides an example where the true absolute error for the generalized simplex product gradient $\nabla_{sp}(fg)(\mathcal{X})$ is greater than $\nabla_s(fg)(\mathcal{X})$.

Example 5.8. Let $f : \mathbb{R} \to \mathbb{R} : y \mapsto e^y$ and $g : \mathbb{R} \to \mathbb{R} : y \mapsto 2e^y$. Consider the ordered set $\mathcal{X} = \langle 0, 1 \rangle$. First, let us find the error bounds for $\nabla_{sp}(fg)(\mathcal{X})$ and $\nabla_s(fg)(\mathcal{X})$. Note that $L_{\nabla f} = e, L_{\nabla g} = 2e$ and $L_{\nabla(fg)} = 8e^2$ on [0,1]. It follows that

$$|f(x_0)|L_{\nabla g} + |g(x_0)|L_{\nabla f} = 4e \approx 10.87.$$

The error bounds are $\varepsilon_{sp}(fg)(\mathcal{X}) = 2e \approx 5.44$ and $\varepsilon_s(fg)(\mathcal{X}) = 4e^2 \approx 29.56$. The true absolute errors are

$$\|\nabla_{sp}(fg)(\mathcal{X}) - \nabla(fg)(x_0)\| \approx 2.87 \leq \varepsilon_{sp}(fg)(\mathcal{X})$$

and

$$\|\nabla_s(fg)(\mathcal{X}) - \nabla(fg)(x_0)\| \approx 8.78 \leq \varepsilon_s(fg)(\mathcal{X}).$$

Example 5.9. Let $f : \mathbb{R} \to \mathbb{R} : y \mapsto e^{-y^2}$ and $g : \mathbb{R} \to \mathbb{R} : y \mapsto e^{-y^3}$. Consider the ordered set $\mathcal{X} = \langle 0, 1 \rangle$. We have $L_{\nabla f} = 2$, $L_{\nabla g} = 3/e$ and $L_{\nabla(fg)} \approx 2.81$ on [0,1]. It follows that

$$|f(x_0)|L_{\nabla g} + |g(x_0)|L_{\nabla f} = \frac{3}{e} + 2 \approx 3.10.$$

The error bounds are $\varepsilon_{sp}(fg)(\mathcal{X}) = \frac{3}{2e} + 1 \approx 1.55$ and $\varepsilon_s(fg)(\mathcal{X}) \approx 1.40$. The true absolute errors are

$$\|\nabla_{sp}(fg)(\mathcal{X}) - \nabla(fg)(x_0)\| \approx 1.26 \leq \varepsilon_{sp}(fg)(\mathcal{X})$$

and

$$\|\nabla_s(fg)(\mathcal{X}) - \nabla(fg)(x_0)\| \approx 0.86 \leq \varepsilon_s(fg)(\mathcal{X}).$$

Note that all the results obtained for $\nabla_{sp}(fg)(\mathcal{X})$ (Corollary 5.2, Theorem 5.5, Corollary 5.6 and Corollary 5.7) can be extended to the case of the product of n functions. Let us present an error bound for the generalized simplex product gradient $\nabla_{sp}(f_1 f_2 \cdots f_n)(\mathcal{X})$.

Proposition 5.10 (Error Bound for $\nabla_{sp}(f_1f_2\cdots f_n)(\mathcal{X})$)**.** *Let $\mathcal{X}$ be an ordered set of $k+1$ points in $\mathbb{R}^d$. Assume that $S(\mathcal{X})$ has full row rank and $f_i \in \mathcal{C}^{1+}$ are Lipschitz continuous with Lipschitz constant $L_{\nabla f_i} \geq 0, i \in \{1,2,\ldots,n\}$, in an open domain $\mathcal{O}$ containing* $\operatorname{conv}(\mathcal{X})$*.*

Define

$$\varepsilon_{sp}(f_1f_2\cdots f_n)(\mathcal{X}) = \frac{\sqrt{k}}{2}\left(\sum_{i=1}^{n}\left(\prod_{j\neq i}|f_j(x_0)|\right)L_{\nabla f_i}\right)\left\|\left(\widehat{S}\left(\mathcal{X}\right)^T\right)^{\dagger}\right\|\Delta.$$

Then

$$\|\nabla_{sp}(f_1f_2\cdots f_n)(\mathcal{X}) - \nabla(f_1f_2\cdots f_n)(x_0)\| \leq \varepsilon_{sp}(f_1f_2\cdots f_n)(\mathcal{X}).$$

Proof. We have

$$\begin{aligned}
&\|\nabla_{sp}(f_1f_2\cdots f_n)(\mathcal{X}) - \nabla(f_1f_2\cdots f_n)(x_0)\| \\
&= \big\|(f_1f_2\cdots f_{n-1})(x_0)\nabla_s f_n(\mathcal{X}) - (f_1f_2\cdots f_{n-1})(x_0)\nabla f_n(\mathcal{X}) \\
&\qquad + \cdots + (f_2f_3\cdots f_n)(x_0)\nabla_s f_1(x_0) - (f_2f_3\cdots f_n)(x_0)\nabla f_1(x_0)\big\| \\
&\leq |(f_1f_2\cdots f_{n-1})(x_0)|\,\|\nabla_s f_n(\mathcal{X}) - \nabla f_n(x_0)\| \\
&\qquad + \cdots + |(f_2f_3\cdots f_n)(x_0)|\,\|\nabla_s f_1(\mathcal{X}) - \nabla f_1(x_0)\| \\
&\leq |(f_1f_2\cdots f_{n-1})(x_0)|\,\frac{\sqrt{k}}{2}L_{\nabla f_n}\left\|\,\left(\widehat{S}\left(\mathcal{X}\right)^T\right)^{\dagger}\right\|\Delta \\
&\qquad + \cdots + |(f_2f_3\cdots f_n)(x_0)|\,\frac{\sqrt{k}}{2}L_{\nabla f_1}\left\|\,\left(\widehat{S}\left(\mathcal{X}\right)^T\right)^{\dagger}\right\|\Delta \\
&= \frac{\sqrt{k}}{2}\left(\sum_{i=1}^{n}\left(\prod_{j\neq i}|f_j(x_0)|\right)L_{\nabla f_i}\right)\left\|\left(\widehat{S}\left(\mathcal{X}\right)^T\right)^{\dagger}\right\|\Delta.
\end{aligned}$$

□

Once again, analyzing the previous error bound, we get Corollary 5.11 that provides sufficient conditions for $\nabla_{sp}(f_1f_2\cdots f_n)(\mathcal{X})$ to be an exact approximation of the true gradient $\nabla(f_1f_2\cdots f_n)(x_0)$.

Corollary 5.11. *Let the assumptions of Proposition 5.10 hold. If any of the following cases hold, then*

$$\nabla_{sp}(f_1f_2\cdots f_n)(\mathcal{X}) = \nabla(f_1f_2\cdots f_n)(x_0).$$

i. The function values $f_i(x_0) = 0$ and $f_j(x_0) = 0$ for $i,j \in \{1,2,\ldots,n\}$.

ii. The function f_i is linear for all $i \in \{1, 2, \ldots, n\}$.

iii. The function f_i is linear and $f_i(x_0) = 0$ for $i \in \{1, 2, \ldots, n\}$.

Corollary 5.6 and Corollary 5.11 are highly restrictive. Nevertheless, it provides two cases where the generalized simplex product gradient is a better or as good approximation than the generalized simplex gradient $\nabla_s(\cdot)(\mathcal{X})$.

Next we investigate the particular case where all f_i, for $i \in \{1, 2, \ldots, n\}$, are equal.

5.2 Power Rule

Let us define the *generalized simplex product gradient of f^n over $\mathcal{X}$*

$$\nabla_{sp} f^n(\mathcal{X}) = n[f(x_0)]^{n-1} \nabla_s f(\mathcal{X}), \tag{5.2}$$

where n is a nonzero integer and $f(x_0)$ is nonzero whenever n is negative.

Note that $\nabla_{sp} f^n(\mathcal{X})$ only requires $f(x_0)$ to be nonzero when n is a negative integer which is not sufficient in Proposition 3.10. Hence, Equation (5.2) is less restrictive than the power rule for a negative integer. First, we introduce an error bound for $\nabla_{sp} f^n(\mathcal{X})$.

Proposition 5.12 (Error Bound for $\nabla_{sp} f^n(\mathcal{X})$)**.** *Let $f : \mathbb{R}^d \to \mathbb{R}$ and let n be a nonzero integer. Let $\mathcal{X}$ be an ordered set of $k+1$ points in $\mathbb{R}^d$ for which $f(x_0)$ is nonzero whenever n is a negative integer. Assume that $S(\mathcal{X})$ has full row rank and $f \in \mathcal{C}^{1+}$ with Lipschitz constant $L_{\nabla f} \geq 0$ in an open domain $\mathcal{O}$ containing* $\operatorname{conv}(\mathcal{X})$*. Define*

$$\varepsilon_{sp} f^n(\mathcal{X}) = \frac{\sqrt{k}}{2} \left(|n| |f(x_0)|^{n-1} L_{\nabla f} \right) \left\| \left(\widehat{S}\left(\mathcal{X}\right)^T \right)^\dagger \right\| \Delta.$$

Then

$$\| \nabla_{sp} f^n(\mathcal{X}) - \nabla f^n(x_0) \| \leq \varepsilon_{sp} f^n(\mathcal{X}).$$

Proof. We have

$$\begin{aligned} \| \nabla_{sp} f^n(\mathcal{X}) - \nabla f^n(x_0) \| &= \left\| n f(x_0)^{n-1} \nabla_s f(\mathcal{X}) - n f(x_0)^{n-1} \nabla f(x_0) \right\| \\ &\leq |n| |f(x_0)|^{n-1} \left\| \nabla_s f(\mathcal{X}) - \nabla f(x_0) \right\| \\ &\leq \frac{\sqrt{k}}{2} \left(|n| |f(x_0)|^{n-1} L_{\nabla f} \right) \left\| \left(\widehat{S}\left(\mathcal{X}\right)^T \right)^\dagger \right\| \Delta \end{aligned}$$

by Proposition 2.14. □

Corollary 5.13. *Let the assumptions of Proposition 5.12 hold. If any of the following cases hold, then*

$$\nabla_{sp} f^n(\mathcal{X}) = \nabla f^n(x_0).$$

i. The function f is linear.

ii. The function value $f(x_0) = 0$ and $n \in \mathbb{N}_+$.

Next, we provide an example where the error bound $\varepsilon_{sp} f^n(\mathcal{X})$ is smaller than $\varepsilon_s f^n(\mathcal{X})$ independent of the ordered set $\mathcal{X}$ and an example where $\nabla_{sp} f^n(\mathcal{X})$ is less accurate than $\nabla_s f^n(\mathcal{X})$.

Example 5.14. Let $f : \mathbb{R} \to \mathbb{R} : y \mapsto y^2 + 1, n = 2$ and so $f^2 : \mathbb{R} \to \mathbb{R} : y \mapsto y^4 + 2y^2 + 1$. Let $\mathcal{X} = \langle x_0, x_1 \rangle$. Without loss of generality, assume that $x_0 < x_1$ (since we are in the determined case, the elements x_0 and x_1 can be rearranged if necessary). Note that the Lipschitz constants are $L_{\nabla f} = 2$ and $L_{\nabla f^2} = \max_{y \in [x_0, x_1]} |4(3y^2 + 1)|$ on $[x_0, x_1]$. It follows that

$$n[f(x_0)]^{n-1} L_{\nabla f} = 4(x_0^2 + 1).$$

Since

$$\begin{aligned} 4(x_0^2 + 1) &\leq L_{\nabla f^2} \\ &= \max 4(3y^2 + 1) \quad \text{for any } y \in [x_0, x_1], \end{aligned}$$

the error bound $\varepsilon_{sp} f^2(\mathcal{X})$ is smaller than the error bound $\varepsilon_s f^2(\mathcal{X})$. Notice, if $\mathcal{X} = \langle 1, 2 \rangle$, the error bounds are $\varepsilon_{sp} f^2(\mathcal{X}) = 4$ and $\varepsilon_s f^2(\mathcal{X}) = 26$. The true absolute errors are

$$\left\| \nabla_{sp} f^2(\mathcal{X}) - \nabla f^2(x_0) \right\| = 4 \leq \varepsilon_{sp} f^2(\mathcal{X})$$

and

$$\left\| \nabla_s f^2(\mathcal{X}) - \nabla f^2(x_0) \right\| = 13 \leq \varepsilon_s f^2(\mathcal{X}).$$

Example 5.15. Let $f : \mathbb{R} \to \mathbb{R} : y \mapsto -y^2 + 10, n = 2$ and so $f^2 : \mathbb{R} \to \mathbb{R} : y \mapsto y^4 - 20y^2 + 100$. Let $\mathcal{X} = \langle 1, 2 \rangle$. Note that the Lipschitz constants are $L_{\nabla f} = 2$ and $L_{\nabla f^2} = 28$ on $[1, 2]$. It follows that

$$n[f(x_0)]^{n-1} L_{\nabla f} = 36.$$

The error bounds are $\varepsilon_{sp}f^2(\mathcal{X}) = 18$ and $\varepsilon_s f^2(\mathcal{X}) = 14$. The true absolute errors are

$$\left\|\nabla_{sp}f^2(\mathcal{X}) - \nabla f^2(x_0)\right\| = 18 \leq \varepsilon_{sp}f^2(\mathcal{X})$$

and

$$\left\|\nabla_s f^2(\mathcal{X}) - \nabla f^2(x_0)\right\| = 9 \leq \varepsilon_s f^2(\mathcal{X}).$$

Lastly, let us provide two examples for a negative integer n. Example 5.16 illustrates the benefit of using $\nabla_{sp}f^n(\mathcal{X})$ and Example 5.17 provides an example where $\nabla_{sp}f^n(\mathcal{X})$ is not better than $\nabla_s f^n(\mathcal{X})$.

Example 5.16. Let $f : \mathbb{R} \to \mathbb{R} : y \mapsto y, n = -2$ and so $f^{-2} : \mathbb{R} \to \mathbb{R} : y \mapsto y^{-2}$. Let $\mathcal{X}= \langle 10^{-6}, 1 + 10^{-6}\rangle$. The Lipschitz constants are $L_{\nabla f} = 0$ and $L_{\nabla f^{-2}} = 6 \times 10^{24}$ on $[10^{-6}, 1 + 10^{-6}]$. We have

$$|n|[f(x_0)]^{n-1}L_{\nabla f} = 0.$$

It follows that the error bound for $\nabla_{sp}f^{-2}(\mathcal{X})$ is enormously smaller than the error bound for $\nabla_s f^{-2}(\mathcal{X})$ as $\varepsilon_{sp}f^{-2}(\mathcal{X}) = 0$ and $\varepsilon_s f^{-2}(\mathcal{X}) = 3 \times 10^{24}$. The true absolute errors are

$$\left\|\nabla_{sp}f^{-2}(\mathcal{X}) - \nabla f^{-2}(x_0)\right\| = 0 \leq \varepsilon_{sp}f^{-2}(\mathcal{X})$$

and

$$\left\|\nabla_s f^{-2}(\mathcal{X}) - \nabla f^{-2}(x_0)\right\| = 2 \times 10^{18} \leq \varepsilon_s f^{-2}(\mathcal{X}).$$

Example 5.17. Let $f : \mathbb{R} \to \mathbb{R} : y \mapsto e^y, n = -2$ and so $f^{-2} : \mathbb{R} \to \mathbb{R} : y \mapsto e^{-2y}$. Let $\mathcal{X} = \langle 0, 1\rangle$. The Lipschitz constants are $L_{\nabla f} = e$ and $L_{\nabla f^{-2}} = 4$ on [0,1]. We have

$$|n|[f(x_0)]^{n-1}L_{\nabla f} = 2e.$$

It follows that the error bound for $\nabla_{sp}f^{-2}(\mathcal{X})$ is bigger than the error bound for $\nabla_s f^{-2}(\mathcal{X})$ as $\varepsilon_{sp}f^{-2}(\mathcal{X}) = e$ and $\varepsilon_s f^{-2}(\mathcal{X}) = 2$. The true absolute errors are

$$\left\|\nabla_{sp}f^{-2}(\mathcal{X}) - \nabla f^{-2}(x_0)\right\| = 1.4365 \leq \varepsilon_{sp}f^{-2}(\mathcal{X})$$

and

$$\left\|\nabla_s f^{-2}(\mathcal{X}) - \nabla f^{-2}(x_0)\right\| = 1.1353 \leq \varepsilon_s f^{-2}(\mathcal{X}).$$

We continue our investigation by looking at the *generalized simplex quotient gradient of* $\left(\frac{f}{g}\right)$ *over* $\mathcal{X}$, denoted $\nabla_{sq}\left(\frac{f}{g}\right)(\mathcal{X})$.

5.3 Quotient Rule

Let $f : \mathbb{R}^d \to \mathbb{R}$ and $g : \mathbb{R}^d \to \mathbb{R}$. Let $\mathcal{X}$ be an ordered set of $k+1$ points in $\mathbb{R}^d$ for which $g(x_0)$ is nonzero. Define the *generalized simplex quotient gradient of* $\left(\frac{f}{g}\right)$ *over* $\mathcal{X}$

$$\nabla_{sq}\left(\frac{f}{g}\right)(\mathcal{X}) = \frac{g(x_0)\nabla_s f(\mathcal{X}) - f(x_0)\nabla_s g(\mathcal{X})}{[g(x_0)]^2}. \tag{5.3}$$

Notice that $\nabla_{sq}\left(\frac{f}{g}\right)(\mathcal{X})$ does not require $g(x_1), g(x_2), \ldots, g(x_k)$ to be all nonzero which is the case for $\nabla_s\left(\frac{f}{g}\right)(\mathcal{X})$ in Theorem 3.8. Hence, the generalized simplex quotient gradient is less restrictive than the quotient rule. This provides a good motive to use the generalized simplex quotient gradient over the generalized simplex gradient in certain situations.

Theorem 5.18 (Error Bound for $\nabla_{sq}\left(\frac{f}{g}\right)(\mathcal{X})$)**.** *Let* $f : \mathbb{R}^d \to \mathbb{R}$ *and* $g : \mathbb{R}^d \to \mathbb{R}$*. Let* $\mathcal{X}$ *be an ordered set of* $k+1$ *points in* $\mathbb{R}^d$ *for which* $g(x_0)$ *is nonzero. Assume that* $S(\mathcal{X})$ *has full row rank,* $f \in \mathcal{C}^{1+}$ *and* $g \in \mathcal{C}^{1+}$ *with Lipschitz constant* $L_{\nabla f} \geq 0$ *and* $L_{\nabla g} \geq 0$ *in an open domain* $\mathcal{O}$ *containing* $\operatorname{conv}(\mathcal{X})$*. Define*

$$\varepsilon_{sq}\left(\frac{f}{g}\right)(\mathcal{X}) = \frac{\sqrt{k}}{2}\left(\left|\frac{1}{g(x_0)}\right| L_{\nabla f} + \left|\frac{f(x_0)}{[g(x_0)]^2}\right| L_{\nabla g}\right)\left\|\left(\widehat{S}\,(\mathcal{X})^T\right)^\dagger\right\|\Delta.$$

Then

$$\left\|\nabla_{sq}\left(\frac{f}{g}\right)(\mathcal{X}) - \nabla\left(\frac{f}{g}\right)(x_0)\right\| \leq \varepsilon_{sq}\left(\frac{f}{g}\right)(\mathcal{X}).$$

Proof. We have

$$\begin{aligned}
&\left\|\nabla_{sq}\left(\frac{f}{g}\right)(\mathcal{X}) - \nabla\left(\frac{f}{g}\right)(x_0)\right\| \\
&\quad= \left\|\frac{g(x_0)\nabla_s f(x_0) - f(x_0)\nabla_s g(\mathcal{X}) - (g(x_0)\nabla f(x_0) - f(x_0)\nabla g(x_0))}{[g(x_0)]^2}\right\| \\
&\quad\leq \left|\frac{1}{g(x_0)}\right| \|\nabla_s f(\mathcal{X}) - \nabla f(x_0)\| + \left|\frac{f(x_0)}{[g(x_0)]^2}\right| \|\nabla_s g(\mathcal{X}) - \nabla g(x_0)\| \\
&\quad\leq \frac{\sqrt{k}}{2}\left(\left|\frac{1}{g(x_0)}\right| L_{\nabla f} + \left|\frac{f(x_0)}{[g(x_0)]^2}\right| L_{\nabla g}\right)\left\|\left(\widehat{S}\,(\mathcal{X})^T\right)^\dagger\right\|\Delta
\end{aligned}$$

by Proposition 2.14. □

Corollary 5.19. *Let the assumptions of Theorem 5.18 hold. If f and g are linear functions, then*

$$\nabla_{sq}\left(\frac{f}{g}\right)(\mathcal{X}) = \nabla\left(\frac{f}{g}\right)(x_0).$$

Now, let us compare the error bound $\varepsilon_{sq}\left(\frac{f}{g}\right)(\mathcal{X})$ and $\varepsilon_s\left(\frac{f}{g}\right)(\mathcal{X})$. Assume that $\frac{f}{g}$ is Lipschitz continuous with Lipschitz constant $L_{\nabla\frac{f}{g}}$ on $\mathcal{O}$. We see that $\varepsilon_{sq}\left(\frac{f}{g}\right)(\mathcal{X})$ is smaller than $\varepsilon_s\left(\frac{f}{g}\right)(\mathcal{X})$ whenever

$$\left|\frac{1}{g(x_0)}\right| L_{\nabla f} + \left|\frac{f(x_0)}{[g(x_0)]^2}\right| L_{\nabla g} < L_{\nabla\frac{f}{g}}.$$

Next we provide one example where $\nabla_{sq}\left(\frac{f}{g}\right)(\mathcal{X})$ is more accurate than $\nabla_s\left(\frac{f}{g}\right)(\mathcal{X})$ and one example where $\nabla_{sq}\left(\frac{f}{g}\right)(\mathcal{X})$ is worse than $\nabla_s\left(\frac{f}{g}\right)(\mathcal{X})$.

Example 5.20. Let $f:\mathbb{R}\to\mathbb{R}: y\mapsto 1$ and $g:\mathbb{R}\to\mathbb{R}: y\mapsto y$.

First, let us consider the ordered set $\mathcal{X}_1 = \langle 10^{-6}, 0\rangle$. The Lipschitz constants are $L_{\nabla f} = 0$ and $L_{\nabla g} = 0$. Note that the Lipschitz constant $L_{\nabla\frac{f}{g}}$ is not defined on $[0, 10^{-6}]$. Also, $\nabla_s\left(\frac{f}{g}\right)(\mathcal{X}_1)$ is not defined since there is a division by zero in the vector $\delta_{\frac{f}{g}}$. On the other hand, the error bound $\varepsilon_{sq}\left(\frac{f}{g}\right)(\mathcal{X}_1) = 0$, and so the true absolute error is

$$\left\|\nabla_{sq}\left(\frac{f}{g}\right)(\mathcal{X}_1) - \nabla\left(\frac{f}{g}\right)(x_0)\right\| = 0 \le \varepsilon_{sq}\left(\frac{f}{g}\right)(\mathcal{X}_1).$$

Second, consider the ordered set $\mathcal{X}_2 = \langle 10^{-6}, 1+10^{-6}\rangle$. Note that the Lipschitz constant $L_{\nabla\frac{f}{g}}$ is now define on $[10^{-6}, 1+10^{-6}] : L_{\nabla\frac{f}{g}} = 2\times 10^{18}$. It follows that the error bounds are $\varepsilon_{sq}\left(\frac{f}{g}\right)(\mathcal{X}_2) = 0$ and $\varepsilon_s\left(\frac{f}{g}\right)(\mathcal{X}_2) = 10^{18}$. The true absolute errors are

$$\left\|\nabla_{sq}\left(\frac{f}{g}\right)(\mathcal{X}_2) - \nabla\left(\frac{f}{g}\right)(x_0)\right\| = 0 \le \varepsilon_{sq}\left(\frac{f}{g}\right)(\mathcal{X}_2)$$

and

$$\left\|\nabla_s\left(\frac{f}{g}\right)(\mathcal{X}_2) - \nabla\left(\frac{f}{g}\right)(x_0)\right\| = 10^{12} \le \varepsilon_s\left(\frac{f}{g}\right)(\mathcal{X}_2).$$

Example 5.21. Let $f : \mathbb{R} \to \mathbb{R} : y \mapsto y^3$, $g : \mathbb{R} \to \mathbb{R} : y \mapsto y^2$ and $\mathcal{X} = \langle 1, 2 \rangle$. The Lipschitz constants are $L_{\nabla f} = 12$, $L_{\nabla g} = 2$ and $L_{\nabla \frac{f}{g}} = 0$ on [1,2]. It follows that the error bounds are $\varepsilon_{sq}\left(\frac{f}{g}\right)(\mathcal{X}) = 7$ and $\varepsilon_{s}\left(\frac{f}{g}\right)(\mathcal{X}) = 0$. The true absolute errors are

$$\left\| \nabla_{sq}\left(\frac{f}{g}\right)(\mathcal{X}) - \nabla\left(\frac{f}{g}\right)(x_0) \right\| = 3 \leq \varepsilon_{sq}\left(\frac{f}{g}\right)(\mathcal{X})$$

and

$$\left\| \nabla_{s}\left(\frac{f}{g}\right)(\mathcal{X}) - \nabla\left(\frac{f}{g}\right)(x_0) \right\| = 0 \leq \varepsilon_{s}\left(\frac{f}{g}\right)(\mathcal{X}).$$

5.4 Chain Rule

We now turn attention to the chain rule. Let us begin by focusing on compositions of linear functions. The next proposition shows that the term $E_{f \circ g}$ vanishes in $\nabla_s(f \circ g)(\mathcal{X})$.

Proposition 5.22 (Chain Rule for Linear Functions)**.** *Let $f : \mathbb{R}^p \to \mathbb{R} : y \mapsto a^T y + c_1$ and $g : \mathbb{R}^d \to \mathbb{R}^p : y \mapsto By + c_2$ where $a \in \mathbb{R}^p, B \in \mathbb{R}^{p \times d}, c_1 \in \mathbb{R}$ and $c_2 \in \mathbb{R}^p$. Suppose $S(g(\mathcal{X}))$ has full rank. Then*

$$\nabla_s(f \circ g)(\mathcal{X}) = (\mathbf{J_s} g(\mathcal{X}))^T \, \nabla_s f(g(\mathcal{X})).$$

Proof. If $k \leq p$, then the result follows from Corollary 4.3. Now, suppose $k > p$. We have

$$\begin{aligned} E_{f \circ g} &= \left(S(\mathcal{X})^T\right)^{\dagger} \left(S\left(g(\mathcal{X})\right)^T \left(S(g(\mathcal{X}))^T\right)^{\dagger} - I_k\right) \delta_f(g(\mathcal{X})) \\ &= \left(S(\mathcal{X})^T\right)^{\dagger} S(g(\mathcal{X}))^T \nabla_s f(g(\mathcal{X})) - \left(S(\mathcal{X})^T\right)^{\dagger} \delta_f(g(\mathcal{X})) \\ &= \left(S(\mathcal{X})^T\right)^{\dagger} S(g(\mathcal{X}))^T \nabla f(g(x_0)) - \left(S(\mathcal{X})^T\right)^{\dagger} \delta_f(g(\mathcal{X})), \end{aligned}$$

as $\nabla_s f(g(\mathcal{X})) = \nabla f(g(x_0))$ whenever f is a linear function and $S(g(\mathcal{X}))$ has full row rank (Proposition 5.1).

Finally,

$$\begin{aligned} E_{f \circ g} &= \left(S(\mathcal{X})^T\right)^{\dagger} S(g(\mathcal{X}))^T a - \left(S(\mathcal{X})^T\right)^{\dagger} \delta_f(g(\mathcal{X})) \quad (\text{since } \nabla f(g(x_0)) = a) \\ &= \left(S(\mathcal{X})^T\right)^{\dagger} S(g(\mathcal{X}))^T a - \left(S(\mathcal{X})^T\right)^{\dagger} S(g(\mathcal{X}))^T a \\ &= 0 \end{aligned}$$

as $\delta_f(g(\mathcal{X})) = S(g(\mathcal{X}))^T a$ for linear functions. □

In the previous proposition, note that even though the term $E_{f\circ g}$ is equal to 0 does not necessarily mean that $\nabla_s(f\circ g)(\mathcal{X}) = \nabla(f\circ g)(x_0)$. The following example illustrates this situation.

Example 5.23. Let $f : \mathbb{R} \to \mathbb{R} : y \mapsto y$ and $g : \mathbb{R}^2 \to \mathbb{R} : y \mapsto y_1 + y_2$. Let

$$\mathcal{X} = \left\langle \begin{bmatrix} 0 & 0 \end{bmatrix}^T, \begin{bmatrix} 1 & 0 \end{bmatrix}^T, \begin{bmatrix} 2 & 0 \end{bmatrix}^T \right\rangle.$$

Then $S\left(g(\mathcal{X})\right)$ has full rank and $\nabla(f\circ g)(x_0) \neq \nabla_s(f\circ g)(\mathcal{X})$.

Details We have $g(\mathcal{X}) = \langle 0, 1, 2\rangle$. It follows that $S\left(g(\mathcal{X})\right) = \begin{bmatrix} 1 & 2 \end{bmatrix}$ has full rank.

The gradient of $(f\circ g)$ at x_0

$$\begin{aligned} \nabla(f\circ g)(x_0) &= \left(\mathbf{J}g(x_0)\right)^T \nabla f\left(g(x_0)\right) \\ &= \begin{bmatrix} 1 \\ 1 \end{bmatrix}. \end{aligned}$$

But the simplex gradient of $(f\circ g)$ over $\mathcal{X}$

$$\begin{aligned} \nabla_s(f\circ g)(\mathcal{X}) &= \left(\mathbf{J_s}g(\mathcal{X})\right)^T \nabla_s f\left(g(\mathcal{X})\right) \\ &= \begin{bmatrix} 1 \\ 0 \end{bmatrix}. \end{aligned}$$

In the previous example, note that $S(\mathcal{X}) = \begin{bmatrix} 1 & 2 \\ 0 & 0 \end{bmatrix}$ does not have full row rank. By adding the assumption that both $S(\mathcal{X})$ and $S\left(g(\mathcal{X})\right)$ have full row rank, we get the following result.

Proposition 5.24 ($\nabla_s(f\circ g)(\mathcal{X}) = \nabla(f\circ g)(x_0)$)**.** *Let $f : \mathbb{R}^p \to \mathbb{R} : y \mapsto a^T y + c_1$ and $g : \mathbb{R}^d \to \mathbb{R}^p : y \mapsto By + c_2$ where $a \in \mathbb{R}^p, B \in \mathbb{R}^{p\times d}, c_1 \in \mathbb{R}$ and $c_2 \in \mathbb{R}^p$. Suppose $S(\mathcal{X})$ and $S(g(\mathcal{X}))$ have full row rank. Then*

$$\nabla_s(f\circ g)(\mathcal{X}) = \nabla(f\circ g)(x_0).$$

Proof. We have

$$\begin{aligned}
&\nabla_s(f\circ g)(\mathcal{X})\\
&=(\mathbf{J_s}g(\mathcal{X}))^T\,\nabla_s f(g(\mathcal{X})) && \text{(by Proposition 5.22)}\\
&=\begin{bmatrix}\nabla_s g_1(\mathcal{X})^T\\ \nabla_s g_2(\mathcal{X})^T\\ \vdots\\ \nabla_s g_p(\mathcal{X})^T\end{bmatrix}^T \nabla_s f(g(\mathcal{X}))\\
&=\begin{bmatrix}\nabla_s g_1(\mathcal{X})^T\\ \nabla_s g_2(\mathcal{X})^T\\ \vdots\\ \nabla_s g_p(\mathcal{X})^T\end{bmatrix}^T \nabla f\left(g(x_0)\right) && \text{(as } S\left(g(\mathcal{X})\right) \text{ has full row rank and } f \text{ linear)}\\
&=\begin{bmatrix}\nabla g_1(x_0)^T\\ \nabla g_2(x_0)^T\\ \vdots\\ \nabla g_p(x_0)^T\end{bmatrix}^T \nabla f\left(g(x_0)\right) && \text{(as } S(\mathcal{X}) \text{ has full row rank and } g \text{ is linear)}\\
&=\nabla(f\circ g)(x_0).
\end{aligned}$$

□

Let us provide an example for the previous proposition.

Example 5.25. Let $g:\mathbb{R}^3\to\mathbb{R}^2:y\mapsto By$ where $B=\begin{bmatrix}1&1&1\\1&-1&-1\end{bmatrix}$ and let $f:\mathbb{R}^2\to\mathbb{R}:y\mapsto a^Ty$ where $a=\begin{bmatrix}2\\-2\end{bmatrix}$. Let $\mathcal{X}=\left\langle\begin{bmatrix}0\\0\\0\end{bmatrix},\begin{bmatrix}1\\0\\0\end{bmatrix},\begin{bmatrix}0\\1\\0\end{bmatrix},\begin{bmatrix}0\\0\\1\end{bmatrix}\right\rangle$. Then $g(\mathcal{X})=\left\langle\begin{bmatrix}0\\0\end{bmatrix},\begin{bmatrix}1\\1\end{bmatrix},\begin{bmatrix}1\\-1\end{bmatrix},\begin{bmatrix}1\\-1\end{bmatrix}\right\rangle$. Note that $\mathcal{X}$ and $g(\mathcal{X})$ have full row rank. We have

$$\begin{aligned}
\nabla_s(f\circ g)(\mathcal{X})&=(\mathbf{J_s}g(\mathcal{X}))^T\,\nabla_s f(g(\mathcal{X}))\\
&=\begin{bmatrix}1&1&1\\1&-1&-1\end{bmatrix}^T\begin{bmatrix}2\\-2\end{bmatrix}\\
&=\begin{bmatrix}0&4&4\end{bmatrix}^T\\
&=\nabla(f\circ g)(x_0).
\end{aligned}$$

Now, we define the *generalized simplex chain gradient of* $(f \circ g)$ *over* $\mathcal{X}$

$$\nabla_{sc}(f \circ g)(\mathcal{X}) = \left(\mathbf{J_s}g(\mathcal{X})\right)^T \nabla_s f(g(\mathcal{X})). \tag{5.4}$$

The next theorem provides an error bound for $\nabla_{sc}(f \circ g)(\mathcal{X})$.

Theorem 5.26 (Error Bound for $\nabla_{sc}(f \circ g)$)**.** *Let* $g : \mathbb{R}^d \to \mathbb{R}^p$*,* $f : \mathbb{R}^p \to \mathbb{R}$ *and* $\mathcal{X}$ *be an ordered set of* $k+1$ *points in* $\mathbb{R}^d$*. Assume that* $S(\mathcal{X})$*,* $S(g(\mathcal{X}))$ *have full row rank. Also, assume* $g \in \mathcal{C}^{1+}$ *with Lipschitz constant* $L_{\nabla g} \geq 0$ *in an open domain* $\mathcal{O}_1$ *containing* $\operatorname{conv}(\mathcal{X})$ *and* $f \in \mathcal{C}^{1+}$ *with Lipschitz constant* $L_{\nabla f} \geq 0$ *in an open domain* $\mathcal{O}_2$ *containing* $\operatorname{conv}(g(\mathcal{X}))$*. Denote* $L_{g_i} \geq 0$ *to be the Lipschitz constant for* g_i *in* $\mathcal{O}_1$ *for all* $i \in \{1, 2, \ldots, p\}$*. Define*

$$\varepsilon_{sc}(f \circ g)(\mathcal{X}) = \frac{\sqrt{k}\,p}{2}\left(\sqrt{k}\,L_{g_*}\,L_{\nabla f}\left\|\left(\widehat{S}\left(g(\mathcal{X})\right)^T\right)^\dagger\right\| + \|\nabla f(g(x_0))\|L_{\nabla g_*}\right)\left\|\left(\widehat{S}\left(\mathcal{X}\right)^T\right)^\dagger\right\|\Delta_*,$$

where

$$\begin{aligned}
\Delta_* &= \max\left\{\Delta_{\mathcal{X}}, \Delta_{g(\mathcal{X})}\right\}\\
L_{g_*} &= \max\{L_{g_i} : i = 1, \ldots, p\}\\
L_{\nabla g_*} &= \max\{L_{\nabla g_i} : i = 1, \ldots, p\}\\
\widehat{S}(\mathcal{X}) &= S(\mathcal{X})/\Delta_{\mathcal{X}}\\
\widehat{S}(g(\mathcal{X})) &= S(g(\mathcal{X}))/\Delta_{g(\mathcal{X})}.
\end{aligned}$$

Then

$$\|\nabla_{sc}(f \circ g)(\mathcal{X}) - \nabla(f \circ g)(x_0)\| \leq \varepsilon_{sc}(f \circ g)(\mathcal{X}).$$

Proof. We have

$$\begin{aligned}
&\|\nabla_{sc}(f \circ g)(\mathcal{X}) - \nabla(f \circ g)(x_0)\|\\
&= \|\left(\mathbf{J_s}g(\mathcal{X})\right)^T \nabla_s f\left(g(\mathcal{X})\right) - \left(\mathbf{J}g(x_0)\right)^T \nabla f\left(g\left(x_0\right)\right)\|\\
&= \|\left(\mathbf{J_s}g(\mathcal{X})\right)^T \nabla_s f\left(g(\mathcal{X})\right) - \left(\mathbf{J_s}g(\mathcal{X})\right)^T \nabla f\left(g\left(x_0\right)\right) + \left(\mathbf{J_s}g(\mathcal{X})\right)^T \nabla f\left(g\left(x_0\right)\right) - \left(\mathbf{J}g(x_0)\right)^T \nabla f\left(g\left(x_0\right)\right)\|\\
&\leq \|\left(\mathbf{J_s}g(\mathcal{X})\right)^T\|\|\nabla_s f\left(g(\mathcal{X})\right) - \nabla f\left(g\left(x_0\right)\right)\| + \|\nabla f\left(g\left(x_0\right)\right)\|\|\left(\mathbf{J_s}g(\mathcal{X}) - \mathbf{J}g(x_0)\right)^T\|.
\end{aligned}$$

Note that

$$\|\nabla_s f\left(g(\mathcal{X})\right) - \nabla f\left(g\left(x_0\right)\right)\| \leq \frac{\sqrt{k}}{2}L_{\nabla f}\left\|\left(\widehat{S}\left(g(\mathcal{X})\right)^T\right)^\dagger\right\|\Delta_{g(\mathcal{X})}$$

by Proposition 2.14, and

$$\begin{aligned}
\| \left(\mathbf{J_s} g(\mathcal{X}) - \mathbf{J} g(x_0)\right)^T \| &= \left\| \begin{bmatrix} (\nabla_s g_1(\mathcal{X}) - \nabla g_1(x_0))^T \\ \vdots \\ (\nabla_s g_p(\mathcal{X}) - \nabla g_p(x_0))^T \end{bmatrix}^T \right\| \\
&\leq \|\nabla_s g_1(\mathcal{X}) - \nabla g_1(x_0)\| \\
&\quad + \cdots + \|\nabla_s g_p(\mathcal{X}) - \nabla g_p(x_0)\| \\
&\leq \frac{\sqrt{k}}{2} \Delta_{\mathcal{X}} \left\| \left(\widehat{S}\,(\mathcal{X})^T\right)^\dagger \right\| \left(L_{\nabla g_1} + \cdots + L_{\nabla g_p}\right) \\
&\leq \frac{\sqrt{k}\, p}{2} L_{\nabla g_*} \left\| \left(\widehat{S}\,(\mathcal{X})^T\right)^\dagger \right\| \Delta_{\mathcal{X}}.
\end{aligned}$$

Also,

$$\begin{aligned}
\| \left(\mathbf{J_s} g(\mathcal{X})\right)^T \| &= \left\| \begin{bmatrix} \nabla_s g_1(\mathcal{X})^T \\ \vdots \\ \nabla_s g_p(\mathcal{X})^T \end{bmatrix}^T \right\| \\
&\leq \|\nabla_s g_1(\mathcal{X})\| + \cdots + \|\nabla_s g_p(\mathcal{X})\| \\
&\leq \left\| \left(\widehat{S}\,(\mathcal{X})^T\right)^\dagger \right\| \left\| \frac{\delta_{g_1}}{\Delta_{\mathcal{X}}} \right\| \\
&\quad + \cdots + \left\| \left(\widehat{S}\,(\mathcal{X})^T\right)^\dagger \right\| \left\| \frac{\delta_{g_p}}{\Delta_{\mathcal{X}}} \right\| \\
&\leq \left\| \left(\widehat{S}\,(\mathcal{X})^T\right)^\dagger \right\| \sqrt{k}\, L_{g_1} \\
&\quad + \cdots + \left\| \left(\widehat{S}\,(\mathcal{X})^T\right)^\dagger \right\| \sqrt{k}\, L_{g_p} \\
&\leq \sqrt{k}\, p\, L_{g_*} \left\| \left(\widehat{S}\,(\mathcal{X})^T\right)^\dagger \right\|.
\end{aligned}$$

All together,

$$\begin{aligned}
&\|\nabla_{sc}(f \circ g)(\mathcal{X}) - \nabla(f \circ g)(x_0)\| \\
&\leq \sqrt{k}\, p\, L_{g_*} \left\| \left(\widehat{S}\,(\mathcal{X})^T\right)^\dagger \right\| \frac{\sqrt{k}}{2} L_{\nabla f} \left\| \left(\widehat{S}\,(g(\mathcal{X}))^T\right)^\dagger \right\| \Delta_{g(\mathcal{X})} \\
&\quad + \|\nabla f(g(x_0))\| \frac{\sqrt{k}\, p}{2} L_{\nabla g_*} \left\| \left(\widehat{S}\,(\mathcal{X})^T\right)^\dagger \right\| \Delta_{\mathcal{X}}
\end{aligned}$$

$$\leq \frac{\sqrt{k}\,p}{2}\left(\sqrt{k}\,L_{g_*}\,L_{\nabla f}\left\|\left(\widehat{S}\,(g(\mathcal{X}))^T\right)^\dagger\right\| + \|\nabla f(g(x_0))\| L_{\nabla g_*}\right)\left\|\left(\widehat{S}\,(\mathcal{X})^T\right)^\dagger\right\|\Delta_*.$$

□

The following corollary provides an alternative error bound for the generalized simplex chain gradient $\nabla_{sc}(f\circ g)(\mathcal{X})$.

Corollary 5.27. *Let the assumptions of Theorem 5.26 hold. Let $L_f \geq 0$ denote the Lipschitz constant of f on $\mathcal{O}_2$. Then*

$$\begin{aligned}&\|\nabla_{sc}(f\circ g)(\mathcal{X}) - \nabla(f\circ g)(x_0)\|\\ &\leq \frac{kp}{2}\left(L_{g_*}\,L_{\nabla f} + L_f\,L_{\nabla g_*} + L_{\nabla f}\,L_{\nabla g_*}\frac{\Delta_*}{2}\right)\left\|\left(\widehat{S}\,(g(\mathcal{X}))^T\right)^\dagger\right\|\left\|\left(\widehat{S}\,(\mathcal{X})^T\right)^\dagger\right\|\Delta_*.\end{aligned}$$

Proof. Let us consider the term

$$\left(\sqrt{k}\,L_{g_*}\,L_{\nabla f}\left\|\left(\widehat{S}\,(g(\mathcal{X}))^T\right)^\dagger\right\| + \|\nabla f(g(x_0))\| L_{\nabla g_*}\right)$$

in the error bound presented in Theorem 5.26. By adding and substracting $\|\nabla_s f(g(\mathcal{X}))\|\,L_{\nabla g_*}$,

$$\begin{aligned}&\sqrt{k}\,L_{g_*}\,L_{\nabla f}\left\|\left(\widehat{S}\,(g(\mathcal{X}))^T\right)^\dagger\right\| + \|\nabla f(g(x_0))\| L_{\nabla g_*}\\
&= \sqrt{k}\,L_{g_*}\,L_{\nabla f}\left\|\left(\widehat{S}\,(g(\mathcal{X}))^T\right)^\dagger\right\| + \|\nabla_s f(g(\mathcal{X}))\|\,L_{\nabla g_*} - \|\nabla_s f(g(\mathcal{X}))\|\,L_{\nabla g_*}\\
&\qquad + \|\nabla f(g(x_0))\| L_{\nabla g_*}\\
&\leq \sqrt{k}\,L_{g_*}\,L_{\nabla f}\left\|\left(\widehat{S}\,(g(\mathcal{X}))^T\right)^\dagger\right\| + \|\nabla_s f(g(\mathcal{X}))\|\,L_{\nabla g_*}\\
&\qquad + \|\nabla f(g(x_0)) - \nabla_s f(g(\mathcal{X}))\|\,L_{\nabla g_*}\\
&\leq \sqrt{k}\,L_{g_*}\,L_{\nabla f}\left\|\left(\widehat{S}\,(g(\mathcal{X}))^T\right)^\dagger\right\| + \|\nabla_s f(g(\mathcal{X}))\|\,L_{\nabla g_*}\\
&\qquad + \frac{\sqrt{k}}{2}L_{\nabla f}\left\|\left(\widehat{S}\,(g(\mathcal{X}))^T\right)^\dagger\right\|\Delta_{g(\mathcal{X})}L_{\nabla g_*}\\
&\leq \sqrt{k}\,L_{g_*}\,L_{\nabla f}\left\|\left(\widehat{S}\,(g(\mathcal{X}))^T\right)^\dagger\right\| + \left\|\left(\widehat{S}\,(g(\mathcal{X}))^T\right)^\dagger\right\|\left\|\frac{\delta_{f\circ g}}{\Delta_{g(\mathcal{X})}}\right\|L_{\nabla g_*}\\
&\qquad + \frac{\sqrt{k}}{2}L_{\nabla f}\left\|\left(\widehat{S}\,(g(\mathcal{X}))^T\right)^\dagger\right\|\;\Delta_{g(\mathcal{X})}L_{\nabla g_*}\\
&\leq \sqrt{k}\,L_{g_*}\,L_{\nabla f}\left\|\left(\widehat{S}\,(g(\mathcal{X}))^T\right)^\dagger\right\| + \left\|\left(\widehat{S}\,(g(\mathcal{X}))^T\right)^\dagger\right\|\sqrt{k}\,L_f\,L_{\nabla g_*}\end{aligned}$$

$$+\frac{\sqrt{k}}{2}L_{\nabla f}\left\|\left(\widehat{S}\left(g(\mathcal{X})\right)^T\right)^{\dagger}\right\|\ \Delta_{g(\mathcal{X})}L_{\nabla g_*}$$

$$=\sqrt{k}\left\|\left(\widehat{S}\left(g(\mathcal{X})\right)^T\right)^{\dagger}\right\|\left(L_{g_*}\ L_{\nabla f}+L_f\ L_{\nabla g_*}+L_{\nabla f}\ L_{\nabla g_*}\frac{\Delta_{g(\mathcal{X})}}{2}\right).$$

It follows that

$$\begin{aligned}&\|\nabla_{sc}(f\circ g)(\mathcal{X})-\nabla(f\circ g)(x_0)\|\\ &\leq\frac{kp}{2}\left(L_{g_*}\ L_{\nabla f}+L_f\ L_{\nabla g_*}+L_{\nabla f}\ L_{\nabla g_*}\frac{\Delta_*}{2}\right)\left\|\left(\widehat{S}\left(g(\mathcal{X})\right)^T\right)^{\dagger}\right\|\left\|\left(\widehat{S}\left(\mathcal{X}\right)^T\right)^{\dagger}\right\|\Delta_*.\end{aligned}$$

□

Analyzing the previous error bounds, we find when the generalized simplex chain gradient $\nabla_{sc}(f\circ g)(\mathcal{X})$ is an exact approximation of the true gradient $\nabla(f\circ g)(x_0)$.

Corollary 5.28. *Let the assumptions of Theorem 5.26 hold. If f and g are linear functions then*

$$\nabla_{sc}(f\circ g)(\mathcal{X})=\nabla(f\circ g)(x_0).$$

Note that if g is Lipschitz continuous on a set $\mathcal{O}_1$ and f is Lipschitz continuous on a set $\mathcal{O}_2$ such that $g(\mathcal{O}_1)\subseteq\mathcal{O}_2$ then $f\circ g$ is Lipschitz continuous on $\mathcal{O}_1$. Thus, $L_{(f\circ g)}$ is well-defined on $\mathcal{O}_1$.

Let us give an example where $\nabla_{sc}(f\circ g)(\mathcal{X})$ is a better approximation than $\nabla_s(f\circ g)(\mathcal{X})$ and one example where $\nabla_{sc}(f\circ g)(\mathcal{X})$ is not a better approximation. By Corollary 4.3, recall that $S(g(\mathcal{X}))$ cannot have full column rank if we want the term $E_{f\circ g}$ not equal to 0 in the formula for $\nabla_s(f\circ g)(\mathcal{X})$. Hence, the following examples consider sample sets of points where $S(g(\mathcal{X}))$ does not have full column rank.

Example 5.29. Let $f:\mathbb{R}\to\mathbb{R}:y\mapsto 1/(y+1)$ and $g:\mathbb{R}\to\mathbb{R}:y\mapsto y^2$. First, consider the ordered set $\mathcal{X}_1=\langle 0,0.5,1\rangle$. Note that $L_{g*}=2$, $L_{\nabla g*}=2$, $L_{\nabla f}=2$ and $L_{\nabla(f\circ g)}=2$ on $[0,1]$. Using Theorem 5.26 and Proposition 2.14, the error bounds are $\varepsilon_{sc}(f\circ g)(\mathcal{X}_1)\approx 3.48$ and $\varepsilon_s(f\circ g)(\mathcal{X}_1)\approx 1.27$. The true absolute errors are

$$\|\nabla_{sc}(f\circ g)(\mathcal{X}_1)-\nabla(f\circ g)(x_0)\|\approx 0.47\leq\varepsilon_{sc}(f\circ g)(\mathcal{X}_1)$$

and

$$\|\nabla_s(f\circ g)(\mathcal{X}_1)-\nabla(f\circ g)(x_0)\|\approx 0.48\leq\varepsilon_s(f\circ g)(\mathcal{X}_1).$$

Now let us consider $\mathcal{X}_2 = \langle 0, 0.5, -0.5, 1\rangle$. The Lipschitz constants are now $L_{g*} = 2, L_{\nabla f} = 16$, $L_{\nabla g*} = 2$ and $L_{\nabla(f\circ g)} = 2$ on $[-0.5, 1]$. The error bounds are $\varepsilon_{sc}(f\circ g)(\mathcal{X}_2) \approx 36.96$ and $\varepsilon_s(f\circ g)(\mathcal{X}_2) \approx 1.42$.

The true absolute errors are

$$\|\nabla_{sc}(f\circ g)(\mathcal{X}_2) - \nabla(f\circ g)(x_0)\| \approx 0.36 \leq \varepsilon_{sc}(f\circ g)(\mathcal{X}_2)$$

and

$$\|\nabla_s(f\circ g(\mathcal{X}_2) - \nabla(f\circ g)(x_0)\| \approx 0.33 \leq \varepsilon_s(f\circ g)(\mathcal{X}_2).$$

Chapter 6

Numerical Experiments

In this chapter, we first explore the *numerical stability* of the formulas introduced previously. Second, we investigate the numerical accuracy of generalized simplex calculus gradients on Moré, Garbow and Hillstrom's Set [28].

Note that all the functions f considered in the next sections are differentiable at x_0. Indeed, the gradient of f at x_0 must exist for the absolute error and the relative error to be well-defined.

All numerical calculations are performed using MATLAB 2018a. Note that MATLAB constructs the double-precision data type according to IEEE Standard 754.

6.1 Numerical Stability

6.1.1 Product Rule

We begin by investigating the numerical stability of the product rule. Let us denote the generalized simplex gradient that uses the product rule by $\nabla_{spe}(\cdot)(\mathcal{X})$. We have presented three approaches to build an approximation of the gradient. Indeed, we can approximate $\nabla(fg)(x_0)$ with:

$$(i) \quad \nabla_s \left(fg\right)\left(\mathcal{X}\right) = \left(S^T\right)^{\dagger} \delta_{fg} \qquad \text{(Def. 2.10)}$$

$$(ii) \quad \nabla_{spe} \left(fg\right)\left(\mathcal{X}\right) = f(x_0)\nabla_s g(\mathcal{X}) + g(x_0)\nabla_s f(\mathcal{X}) + E_{fg} \qquad \text{(Theorem 3.2)}$$

and

$$(iii) \quad \nabla_{sp} \left(fg\right)\left(\mathcal{X}\right) = f(x_0)\nabla_s g(\mathcal{X}) + g(x_0)\nabla_s f(\mathcal{X}) \qquad \text{(Eq. (5.1)).}$$

Let us provide three examples. We include the table for the first example in the current section. For the remaining examples, tables are provided in the appendix. We present one example where each approach is the most stable.

Let us clarify some details about the tables. The parameter β_m is defined as

$$\beta_m = 10^{-m}, \quad m \in \mathbb{N}. \tag{6.1}$$

The role of β_m is to shrink $S(\mathcal{X})$: $S(\mathcal{X})_m = \beta_m S(\mathcal{X})$.

The absolute error for a gradient approximation technique, for instance $\nabla_s(\cdot)(\mathcal{X})$, is denoted by AE $\nabla_s(\cdot)(\mathcal{X})$ and equal to

$$\mathrm{AE} = \|\nabla_s(\cdot)(\mathcal{X}) - \nabla(\cdot)(x_0)\|.$$

The relative error for $\nabla_s(\cdot)(\mathcal{X})$, denoted RE $\nabla_s(\cdot)(\mathcal{X})$, is

$$\mathrm{RE} = \frac{\mathrm{AE}}{\|\nabla(\cdot)(x_0)\|}.$$

For the purpose of this thesis, we say that the formula becomes *unstable* at $m = \ell$ whenever the relative error at ℓ is greater than the relative error at $\ell - 1$, or, the relative error at ℓ is equal to zero and all the relative errors are not equal to zero for $m = 1, 2, \dots, \ell - 1$. Essentially, we are looking for points where any pattern in the value of the relative error changes drastically. This concept of stability is an indicator of how small Δ can be before numerical errors occur. Since Δ is intimately linked to the accuracy of our approximation, stability also provides information about the maximal accuracy that an approximation technique can reach on the functions considered.

In the following tables, a boldfaced number exhibits when a formula becomes unstable.

Example 6.1. Let $f : \mathbb{R} \to \mathbb{R} : y \mapsto \log y$ and $g : \mathbb{R} \to \mathbb{R} : y \mapsto e^y$. Consider $\mathcal{X} = \langle 2, 3 \rangle$. Table 3 shows that $\nabla_s(fg)(\mathcal{X})$ is the most stable. Indeed, $\nabla_s(fg)(\mathcal{X})$ becomes unstable when $\beta_m = 10^{-10}$. Using the approaches $\nabla_{spe}(fg)(\mathcal{X})$ and $\nabla_{sp}(fg)(\mathcal{X})$, the formulas become unstable at a bigger value of β_m: at $\beta_m = 10^{-9}$. For this reason, we say that $\nabla_s(fg)(\mathcal{X})$ is the most stable.

Example 6.2. Let $f : \mathbb{R} \to \mathbb{R} : y \mapsto y$ and $g : \mathbb{R} \to \mathbb{R} : y \mapsto y$. Consider $\mathcal{X} = \langle 4, 5 \rangle$. Appendix Table 6 shows that $\nabla_{spe}(fg)(\mathcal{X})$ is the most stable.

Theoretically, $\nabla_{sp}(fg)(\mathcal{X})$ is perfectly accurate since f and g are linear functions and $S(\mathcal{X})$ has full row rank. However, numerical errors show up at $\beta_m = 10^{-11}$.

Example 6.3. Let $f : \mathbb{R} \to \mathbb{R} : y \mapsto \sqrt{y}$ and $g : \mathbb{R} \to \mathbb{R} : y \mapsto \sqrt{y}$. Consider $\mathcal{X} = \langle 1, 2 \rangle$. Appendix Table 7 shows that $\nabla_{sp}(fg)(\mathcal{X})$ is the most stable.

Note that $\nabla_s(fg)(\mathcal{X})$ and $\nabla_{spe}(fg)(\mathcal{X})$ are the most accurate approximation since fg is a linear function and $S(\mathcal{X})$ has full row rank.

Table 3: An Example Where $\nabla_s(fg)(\mathcal{X})$ is the Most Stable

β_m	**RE** $\nabla_s(fg)(\mathcal{X})$	**RE** $\nabla_{spe}(fg)(\mathcal{X})$	**RE** $\nabla_{sp}(fg)(\mathcal{X})$
10^{0}	9.2197e-01	9.2197e-01	3.3805e-01
10^{-1}	6.2907e-02	6.2907e-02	1.9900e-02
10^{-2}	6.0714e-03	6.0714e-03	1.8702e-03
10^{-3}	6.0500e-04	6.0500e-04	1.8584e-04
10^{-4}	6.0479e-05	6.0479e-05	1.8572e-05
10^{-5}	6.0477e-06	6.0477e-06	1.8571e-06
10^{-6}	6.0474e-07	6.0474e-07	1.8568e-07
10^{-7}	6.0080e-08	6.0313e-08	1.8407e-08
10^{-8}	9.1121e-09	9.5759e-09	5.3853e-09
10^{-9}	8.8912e-10	**1.7861e-08**	**1.8281e-08**
10^{-10}	**5.0283e-07**	1.2142e-07	1.2138e-07

6.1.2 Power Rule for a Positive Integer

Let us denote the generalized simplex gradient that uses the power rule for a positive integer n by $\nabla_{spe}(\cdot)(\mathcal{X})$. We use the same notation as the product rule since the power rule is a particular case of the product rule. We have three approaches to build an approximation of the gradient. Indeed, we can approximate $\nabla(fg)(x_0)$ with:

$$(i) \quad \nabla_s f^n(\mathcal{X}) = \left(S^T\right)^{\dagger} \delta_{f^n} \qquad \text{(Definition 2.10)}$$

$$(ii) \quad \nabla_{spe} f^n(\mathcal{X}) = n[f(x_0)]^{n-1}\nabla_s f(\mathcal{X}) + E_{f^n} \qquad \text{(Corollary 3.3)}$$

and

$$(iii) \quad \nabla_{sp} f^n(\mathcal{X}) = n[f(x_0)]^{n-1}\nabla_s f(\mathcal{X}) \qquad \text{(Equation (5.2))}.$$

As before, we provide three examples.

Example 6.4. Let $f\colon \mathbb{R}^2 \to \mathbb{R}\colon y \mapsto 5ey_1 - 4\pi y_2 + 0.00001\pi e$ and let $n = 3$. Consider $\mathcal{X} = \left\langle \begin{bmatrix} 0 \\ \sqrt{\pi} \end{bmatrix}, \begin{bmatrix} e^2 \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ \pi^2 \end{bmatrix} \right\rangle$. Appendix Table 8 shows that $\nabla_s f^3(\mathcal{X})$ is the most stable.

Note that f is a linear function and $S(\mathcal{X})$ has full row rank but the relative error for $\nabla_{sp} f^3(\mathcal{X})$ is not equal to 0 due to numerical errors.

Example 6.5. Let $f : \mathbb{R} \to \mathbb{R} : y \mapsto y + 2$ and let $n = 4$. Consider $\mathcal{X} = \langle 4, 5\rangle$. Appendix Table 9 shows that $\nabla_{spe} f^4(\mathcal{X})$ is the most stable and

$\nabla_{sp} f^4(\mathcal{X})$ is the most accurate since f is a linear function and $S(\mathcal{X})$ has full row rank. It is worth mentioning that the values of the relative errors change drastically from $\beta_m = 10^{-15}$ to $\beta_m = 10^{-16}$. This is likely due from rounding errors with tiny numbers rounded to the nearest integer.

For this last example, we compare the absolute errors since the true gradient of the function at x_0 is equal to zero.

Example 6.6. Let $f\colon \mathbb{R}^2 \to \mathbb{R}\colon y \mapsto y_1^{10} + y_2^{10}$ and let $n = 5$.

Consider $\mathcal{X} = \left\langle \begin{bmatrix} 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ 1 \end{bmatrix} \right\rangle$. Appendix Table 10 shows that $\nabla_{sp} f^5(\mathcal{X})$ is the most stable.

Theoretically, note that $\nabla_{sp} f^5(\mathcal{X})$ is an exact approximation of $\nabla f^5(x_0)$ as $f(x_0) = 0$ (Corollary 5.13).

Also, note that this table contains numbers extremely close to 0. This can be explained by the fact that MATLAB allows *subnormal numbers* [31].

6.1.3 Quotient Rule

We denote the generalized simplex gradient that uses the quotient rule by $\nabla_{sqe}(\cdot)(\mathcal{X})$. We have introduced three approaches to build an approximation of the gradient. The gradient of f over g at x_0 can be approximated with:

$$(i) \quad \nabla_s \left(\frac{f}{g}\right)(\mathcal{X}) = \left(S^T\right)^\dagger \delta_{\frac{f}{g}} \qquad \text{(Def. 2.10)}$$

$$(ii) \quad \nabla_{sqe} \left(\frac{f}{g}\right)(\mathcal{X}) = \frac{g(x_0)\nabla_s f(\mathcal{X}) - f(x_0)\nabla_s g(\mathcal{X})}{[g(x_0)]^2} - E_{\frac{f}{g}} \qquad \text{(Theorem 3.8)}$$

and

$$(iii) \quad \nabla_{sq} \left(\frac{f}{g}\right)(\mathcal{X}) = \frac{g(x_0)\nabla_s f(\mathcal{X}) - f(x_0)\nabla_s g(\mathcal{X})}{[g(x_0)]^2} \qquad \text{(Eq. (5.3)).}$$

Next we provide three examples.

Example 6.7. Let $f : \mathbb{R} \to \mathbb{R} : y \mapsto y^2$ and $g : \mathbb{R} \to \mathbb{R} : y \mapsto y$. Consider $\mathcal{X} = \langle 4, 5\rangle$. Appendix Table 11 shows that $\nabla \left(\frac{f}{g}\right)(\mathcal{X})$ is stable and accurate.

Example 6.8. Let $f : \mathbb{R} \to \mathbb{R} : y \mapsto 1$ and $g : \mathbb{R} \to \mathbb{R} : y \mapsto \log(y)$. Consider $\mathcal{X} = \langle 2, 3\rangle$. Appendix Table 12 shows that $\nabla_{sqe} \left(\frac{f}{g}\right)(\mathcal{X})$ is the most stable.

Example 6.9. Let $f : \mathbb{R} \to \mathbb{R} : y \mapsto 1$ and $g : \mathbb{R} \to \mathbb{R} : y \mapsto y$. Consider $\mathcal{X} = \langle 10^{-8}, 1 + 10^{-8}\rangle$. Appendix Table 13 shows that $\nabla_{sq} \left(\frac{f}{g}\right)(\mathcal{X})$ is stable and accurate.

6.1.4 Power Rule for a Negative Integer

Let us denote the generalized simplex gradient that uses the power rule for a negative integer $-n$ by ∇_{spe}. Once again, we have three approaches to build an approximation of the gradient. We can approximate $\nabla f^{-n}(x_0)$ with:

$$(i) \quad \nabla_s f^{-n}(\mathcal{X}) = \left(S^T\right)^\dagger \delta_{f^{-n}} \qquad \text{(Definition 2.10)}$$

$$(ii) \quad \nabla_{spe} f^{-n}(\mathcal{X}) = -n[f(x_0)]^{-n-1}\nabla_s f(\mathcal{X}) - E_{f^{-n}} \qquad \text{(Proposition 3.10)}$$

and

$$(iii) \quad \nabla_{sp} f^{-n}(\mathcal{X}) = -n[f(x_0)]^{-n-1}\nabla_s f(\mathcal{X}) \qquad \text{(Equation (5.2))}.$$

Once again, we provide three examples.

Example 6.10. Let $f\colon \mathbb{R} \to \mathbb{R}\colon y \mapsto 0.0001\pi y + 0.0001\pi$ and let $-n = -3$. Consider $\mathcal{X} = \langle 0, 3\pi, -\pi\rangle$. Appendix Table 14 shows that $\nabla_s f^{-3}(\mathcal{X})$ is the most stable. In theory, note that $\nabla_{sp} f^{-3}(\mathcal{X})$ is an exact approximation of $\nabla f^{-3}(x_0)$.

Example 6.11. Let $f : \mathbb{R} \to \mathbb{R} : y \mapsto y + 2$ and let $-n = -6$. Consider $\mathcal{X} = \langle 4, 5\rangle$. Appendix Table 15 shows that $\nabla_{spe} f^{-6}(\mathcal{X})$ is the most stable.

Theoretically, note that $\nabla_{sp} f^{-3}(\mathcal{X})$ is an exact approximation of $\nabla f^{-3}(x_0)$ since f is a linear function and $S(\mathcal{X})$ has full row rank. However, numerical errors emerge at $\beta_m = 10^{-11}$.

Example 6.12. Let $f : \mathbb{R} \to \mathbb{R} : y \mapsto y$ and let $-n = -5$. Consider $\mathcal{X} = \left\langle 2^{-8}, 1 + 2^{-8}\right\rangle$. Appendix Table 16 shows that $\nabla_{sp} f^{-n}(\mathcal{X})$ is stable and perfectly accurate since f is a linear function and $S(\mathcal{X})$ has full row rank.

6.1.5 Chain Rule

Lastly, we investigate the numerical stability of the chain rule. Let us denote the generalized simplex gradient that uses the chain rule by $\nabla_{sce}(\cdot)(\mathcal{X})$. We have three approaches to build an approximation of the gradient. We can approximate $\nabla(f \circ g)(x_0)$ with:

$$(i) \quad \nabla_s (f \circ g)(\mathcal{X}) = \left(S^T\right)^\dagger \delta_{f\circ g} \qquad \text{(Definition 2.10)}$$

$$(ii) \quad \nabla_{sce} (f \circ g)(\mathcal{X}) = \left(\mathbf{J_s} g(\mathcal{X})\right)^T \nabla_s f(g(\mathcal{X})) - E_{f\circ g} \qquad \text{(Theorem 4.2)}$$

and

$$(iii) \quad \nabla_{sc}\left(f \circ g\right)(\mathcal{X}) = \left(\mathbf{J_s} g(\mathcal{X})\right)^T \nabla_s f(g(\mathcal{X})) \qquad \text{(Equation (5.4))}.$$

Finally, we provide three examples.

Example 6.13. Let $f : \mathbb{R}^2 \to \mathbb{R} : y \mapsto (y_1 + y_2)^2$ and let $g : \mathbb{R}^2 \to \mathbb{R}^2 : y \mapsto \begin{bmatrix} \sqrt{y_1 + y_2} \\ \sqrt{y_1 + 2y_2} \end{bmatrix}$. Consider $\mathcal{X} = \left\langle \begin{bmatrix} 1 \\ 1 \end{bmatrix}, \begin{bmatrix} 2 \\ 2 \end{bmatrix}, \begin{bmatrix} 3 \\ 3 \end{bmatrix} \right\rangle$. Appendix Table 17 shows that $\nabla_s(f \circ g)(\mathcal{X})$ is the most stable.

Note that $S(\mathcal{X})$ and $S(g(\mathcal{X}))$ do not have full row rank in this example. We see that the relative errors for the three approaches do not decrease as β_m decreases.

Example 6.14. Let $f : \mathbb{R}^2 \to \mathbb{R} : y \mapsto 0.00001y_1 - 10000y_2 + 2$ and the inner function g be $g : \mathbb{R}^2 \to \mathbb{R}^2 : y \mapsto \begin{bmatrix} 0.000001y_1 - 100y_2 + 2 \\ 1000y_1 + 0.00001y_2 \end{bmatrix}$. Consider $\mathcal{X} = \left\langle \begin{bmatrix} 0 \\ 0 \end{bmatrix}, \begin{bmatrix} -9 \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ -9 \end{bmatrix}, \begin{bmatrix} 1 \\ 99999 \end{bmatrix} \right\rangle$. Appendix Table 18 shows that $\nabla_{sce}(f \circ g)(\mathcal{X})$ is the most stable.

Note that $S(\mathcal{X})$ and $S(g(\mathcal{X}))$ have full row rank. Nevertheless, due to numerical errors, we do not obtain a relative error equal to 0 using $\nabla_{sc}(f \circ g)(\mathcal{X})$.

Example 6.15. Let $f : \mathbb{R} \to \mathbb{R} : y \mapsto \sqrt{y}$ and let $g : \mathbb{R} \to \mathbb{R} : y \mapsto y^2$. Consider $\mathcal{X} = \langle 2, 3, 4, 5 \rangle$. Appendix Table 19 shows that $\nabla_{sc}(f \circ g)(\mathcal{X})$ is the most stable. Note that $(f \circ g)$ is a linear function and $S(\mathcal{X})$ have full row rank. Hence, $\nabla_s(f \circ g)(\mathcal{X})$ and $\nabla_{sce}(f \circ g)(\mathcal{X})$ are perfectly accurate. However, numerical errors emerge at $\beta_{-1} = 10^{-1}$.

6.2 Numerical Accuracy

In this section, we investigate the numerical accuracy of generalized simplex calculus gradients using the functions defined in Moré, Garbow and Hillstrom's Set [28]. First, we compare the generalized simplex gradient $\nabla_s(f \circ g)(\mathcal{X})$ with the generalized simplex chain gradient $\nabla_{sc}(f \circ g)(\mathcal{X})$. Second, we compare the generalized simplex gradient $\nabla_s(f_1 \cdots f_p)(\mathcal{X})$ with the generalized simplex product gradient $\nabla_{sp}(f_1 \cdots f_p)(\mathcal{X})$. The sample set of points used on every problem is $\mathcal{X} = \langle x_0, x_0 + e_1, \ldots, x_0 + e_d, x_0 - e_1, \ldots, x_0 - e_d \rangle$ where x_0 is the starting point defined in [28].

Let us define β to be a real number in the interval (0,1]. The role of β is to shrink the matrix $S(\mathcal{X})$. Our goal is to determine the largest value of β ncessary to obtain a relative error less than 10^{-3}. To achieve this goal, we do the following.

1. Let $\beta = 1$. Compute the approximate gradient and the resulting relative error. If the relative error is less than 10^{-3}, then stop (return $\beta = 1$).
2. Compute the approximate gradient using $\beta \in \{10^{-1}, 10^{-2}, \dots, 10^{-8}\}$ and the resulting relative error, until a value is found that gives a relative error less than 10^{-3}. If none of these values provide a relative error less than 10^{-3}, then return `error`.
3. Apply a bisection method with a tolerance of 10^{-6} to find the highest value of β that returns a relative error less than 10^{-3}.

Our findings are presented in Tables 4 and 5. In the tables, a boldfaced number is associated with the approach that does better on a certain problem. A boldfaced and underlined number means that, not only the gradient approximation technique does better but also, the value of β is at least one order of magnitude higher. In other words, the associated method does significantly better.

Table 4 provides our results regarding the chain rule. The outer function used is $f : \mathbb{R}^p \to \mathbb{R} : y \mapsto \sum_{i=1}^{p} \|y_i\|^2$. In the table, the dimension of the domain of the inner function g is given by d and the dimension of the codomain by p. In general, the generalized simplex gradient $\nabla_s(f \circ g)(\mathcal{X})$ does better as supported by the average and median calculated. However, only three values of β are boldfaced and underlined. While the generalized simplex gradient is slightly more accurate, both approaches perform similarly on 32 problems.

Some precautions are necessary to generalize these results and claim that the generalized simplex gradient is a better approximation than the generalized simplex chain gradient. Indeed, we have only considered one outer function f. Choosing another outer function could affect our conclusion regarding the accuracy of both methods.

The results of the second numerical experiment are presented in Table 5. The dimension of the domain is given by d and the number of functions f_i is given by p. In this case, we compare the generalized simplex gradient $\nabla_s F(\mathcal{X})$ where $F = f_1 \cdot f_2 \cdots f_p$, and the generalized simplex product gradient $\nabla_{sp}(f_1 f_2 \cdots f_p)(\mathcal{X})$. The latter is the clear winner. Except for the problem Box3D, the generalized simplex product gradient does better on every problem.

Note that all problems from Moré, Garbow and Hillstrom's Set contain functions that are not linear except LinearFR, LinearR1 and LinearR1W0. We see that, even when all the functions f_i for $i \in \{1, 2, \ldots, p\}$ are not linear in a certain problem, the generalized simplex product gradient perform better than the generalized simplex gradient.

While we must still be cautious about drawing any universal conclusions, since there is only really one way to create an experiment where the product rule would be applied this result seems more likely to be generalizable.

Table 4: Testing the Chain Rule on Moré, Garbow and Hillstrom's Set

Description			β to Attain $RE \leq 10^{-3}$	
Function	**d**	**p**	$\nabla_s(f \circ g)(\mathcal{X})$	$\nabla_{sc}(f \circ g)(\mathcal{X})$
1. Rosenbrock	2	2	**2.20e-02**	1.76e-02
2. Freudenstein	2	2	**4.15e-02**	1.12e-02
3. PowellBS	2	2	**1**	3.83e-04
4. BrownBS	2	3	1	1
5. Beale	2	3	3.19e-02	3.19e-02
6. Jenrich	2	4	9.56e-03	9.56e-03
7. Helical	3	3	**6.65e-02**	6.55e-02
8. Bard	3	15	4.27e-02	4.27e-02
9. Gaussian	3	15	7.48e-03	7.48e-03
10. Meyer	3	16	1	1
11. Gulf	3	20	7.18e-03	7.18e-03
12. Box3D	3	3	6.41e-01	**7.29e-01**
13. PowellS	4	4	**6.24e-02**	4.58e-02
14. Wood	4	6	1.00e-01	1.00e-01
15. Kowalik	4	11	2.65e-02	2.65e-02
16. Brown	4	4	**8.83e-01**	3.15e-01
17. Osborne1	5	33	2.09e-04	2.09e-04
18. Biggs	6	6	1.30e-01	1.30e-01
19. Osborne2	11	65	1.26e-02	1.26e-02
20. Watson	31	31	**1**	4.47e-02
21. RosenbrockE	4	4	**2.52e-02**	2.02e-02
22. PowellExt	8	8	**6.29e-02**	4.48e-02
23. Penalty1	4	5	1.47e-01	1.47e-01
24. Penalty2	6	12	2.92e-02	2.92e-02
25. VariablyDim	7	9	1.04e-01	1.04e-01
26. Trigonometric	7	7	**6.44e-03**	3.21e-03
27. BrownAlm	9	9	1	1
28. DiscreteBnd	5	5	**1.56e-02**	6.99e-03
29. DiscreteInt	3	3	**2.12e-02**	1.63e-02
30. BroydenTri	5	5	**2.74e-02**	2.02e-02
31. BroydenBan	8	8	**2.05e-02**	1.69e-02
32. LinearFR	10	13	1	1
33. LinearR1	10	10	1	1
34. LinearR1W0	10	10	1	1
35.Chebyquad	4	5	**6.31e-03**	1.98e-03
Average			**3.01e-01**	2.28e-01
Median			**4.27e-02**	3.19e-02

Table 5: Testing the Product Rule on Moré, Garbow and Hillstrom's Set

Description			β to Attain $RE \leq 10^{-3}$	
Function	**d**	**p**	$\nabla_s(f_1 \dots f_p)(\mathcal{X})$	$\nabla_{sp}(f_1 \dots f_p)(\mathcal{X})$
1. Rosenbrock	2	2	7.82e-02	$\underline{\textbf{1}}$
2. Freudenstein	2	2	1.74e-02	**4.03e-02**
3. PowellBS	2	2	2.71e-02	$\underline{\textbf{1}}$
4. BrownBS	2	3	1	1
5. Beale	2	3	2.72e-02	**8.41e-02**
6. Jenrich	2	4	1.67e-02	**2.25e-02**
7. Helical	3	3	1	1
8. Bard	3	15	7.65e-03	$\underline{\textbf{8.51e-02}}$
9. Gaussian	3	15	9.15e-06	$\underline{\textbf{4.60e-02}}$
10. Meyer	3	16	2.28e-04	$\underline{\textbf{1}}$
11. Gulf	3	3	1.03e-02	**1.09e-02**
12. Box3D	3	3	**6.91e-01**	5.95e-01
13. PowellS	4	4	3.30e-02	$\underline{\textbf{1}}$
14. Wood	4	6	1.99e-01	$\underline{\textbf{1}}$
15. Kowalik	4	11	9.03e-04	$\underline{\textbf{1.68e-02}}$
16. Brown	4	4	3.43e-01	$\underline{\textbf{1}}$
17. Osborne1	5	33	1.03e-05	$\underline{\textbf{1}}$
18. Biggs	6	6	3.39e-03	$\underline{\textbf{1}}$
19. Osborne2	11	65	3.28e-04	$\underline{\textbf{3.81e-02}}$
20. Watson	2	31	5.77e-03	$\underline{\textbf{1}}$
21. RosenbrockE	4	4	7.89e-02	$\underline{\textbf{1}}$
22. PowellExt	8	8	3.29e-02	$\underline{\textbf{1}}$
23. Penalty1	4	5	2.34e-01	$\underline{\textbf{1}}$
24. Penalty2	6	12	5.24e-02	$\underline{\textbf{1}}$
25. VariablyDim	7	9	1.19e-01	$\underline{\textbf{1}}$
26. Trigonometric	7	7	1.73e-03	$\underline{\textbf{1}}$
27. BrownAlm	9	9	8.78e-01	$\underline{\textbf{1}}$
28. DiscreteBnd	5	5	8.26e-04	$\underline{\textbf{1}}$
29. DiscreteInt	3	3	3.38e-02	$\underline{\textbf{1}}$
30. BroydenTri	5	5	3.09e-02	$\underline{\textbf{1}}$
31. BroydenBan	8	8	2.30e-02	$\underline{\textbf{1}}$
32. LinearFR	10	13	6.28e-02	$\underline{\textbf{1}}$
33. LinearR1	10	10	5.90e-02	$\underline{\textbf{1}}$
34. LinearR1W0	10	10	6.81e-02	$\underline{\textbf{1}}$
35.Chebyquad	2	2	1.05e-02	$\underline{\textbf{1}}$
Average			1.47e-01	**7.69e-01**
Median			3.09e-02	$\underline{\textbf{1}}$

Chapter 7

Conclusion

The new calculus rules introduced in this thesis provide an attractive framework that holds for underdetermined, determined and overdetermined simplex gradients and under minimal assumptions. The calculus rules for generalized simplex gradients can be written in a similar way than those for the true gradients plus a term E. Removing the term E from the calculus rules leads to several new approaches for approximating gradients. The new approaches that we named generalized simplex calculus gradients have some interesting benefits. In particular, under certain assumptions, they suit perfectly linear functions. In regards to the quotient rule and the power rule for a negative exponent, the new approaches further allow us to remove the assumption that $g(x_i) \neq 0$ for all $i \in \{1, 2, \dots, k\}$.

Error bounds for generalized simplex calculus gradients were presented. Analyzing those error bounds, we obtained results regarding when generalized simplex calculus gradients are exact approximations of the true gradients.

In regards to the chain rule, it is enchanting to see that, when $k \leq p$ and $S(g(\mathcal{X}))$ has full rank, the term $E_{f \circ g}$ vanishes and we get back the chain rule for the true gradient. In all cases, the term $E_{f \circ g}$ vanishes for linear functions whenever $S(g(\mathcal{X}))$ has full rank. Furthermore, we showed that the chain rule is perfectly accurate for linear functions whenever $S(\mathcal{X})$ and $S\left(g(\mathcal{X})\right)$ have full row rank.

The results allowed us to state three novel approaches to approximate, for instance, the true gradient $\nabla(fg)(x_0)$: the generalized simplex gradient $\nabla_s(fg)(\mathcal{X})$ (Definition 2.13), the generalized simplex gradient $\nabla_s(fg)(\mathcal{X})$ using the product rule (Theorem 3.2) and the generalized simplex product gradient $\nabla_{sp}(fg)(\mathcal{X})$ (Equation (5.1)). From a numerical perspective, we showed that, at least in some cases, the generalized simplex gradient that uses the calculus rule and the generalized simplex calculus gradient can improve the stability of our calculations. We provided three examples for each calculus rule illustrating a situation where each approach is stable. These examples also showed the important gain in accuracy we can obtain using the calculus rules or the generalized simplex calculus gradients.

Based on the numerical experiments executed on Moré, Garbow and Hillstrom's Set, the performance of the generalized simplex gradient and the generalized simplex chain gradient is almost similar. In general, the generalized simplex gradient does slightly better than the generalized simplex chain gradient. The results of the second experiment using the generalized simplex product are striking. Even when the p functions involved are not linear, the generalized simplex product performed better except on one problem, Box3D. These results demonstrate all the potential of this new approach.

In practice, it can be difficult to determine which approach does better on a certain problem. One strategy to determine the best approach could be to approximate the Lipschitz constants involved. Let us provide a rough algorithm that may be used inside an optimization routine. Algorithm 1 proposes a procedure to decide between the generalized simplex gradient $\nabla_s(fg)(\mathcal{X})$ and the generalized simplex product gradient $\nabla_{sp}(fg)(\mathcal{X})$ based on an approximation of the Lipschitz constants.

```
Algorithm 1: Approximating the Lipschitz Constants
  Given an optimization problem involving a function $F = f \cdot g$ where
   $f : \mathbb{R}^d \to \mathbb{R}$ and $g : \mathbb{R}^d \to \mathbb{R}$. At iteration $j \in \mathbb{N}_+$:
  0. Input
   $\mathcal{X}^j = \langle x_0^j, \ldots, x_k^j \rangle$    sample set of points
   $x_0^{j-1}$    $x_0$ at iteration $j-1$
   $\tilde{\nabla} f^{j-1}, \tilde{\nabla} g^{j-1}, \tilde{\nabla} F^{j-1}$    gradient approximation of $f, g$ and $F$ at $x_0^{j-1}$
  1. Approximate the Lipschitz constants
  Assume $x_0^j \neq x_0^{j-1}$. Then
   $\tilde{L}_{\nabla f} = \dfrac{\|\nabla_s f(\mathcal{X}^j) - \tilde{\nabla} f^{j-1}\|}{\|x_0^j - x_0^{j-1}\|}$
   $\tilde{L}_{\nabla g} = \dfrac{\|\nabla_s g(\mathcal{X}^j) - \tilde{\nabla} g^{j-1}\|}{\|x_0^j - x_0^{j-1}\|}$
   $\tilde{L}_{\nabla F} = \min \left\{ \dfrac{\|\nabla_s F(\mathcal{X}^j) - \tilde{\nabla} F^{j-1}\|}{\|x_0^j - x_0^{j-1}\|}, \dfrac{\|\nabla_{sp} F(\mathcal{X}^j) - \tilde{\nabla} F^{j-1}\|}{\|x_0^j - x_0^{j-1}\|} \right\}$
  2. Update
  if $|f(x_0^j)|\tilde{L}_{\nabla g} + |g(x_0^j)|\tilde{L}_{\nabla f} \leq \tilde{L}_{\nabla F}$ then
    $\tilde{\nabla} F^j = \nabla_{sp} F(\mathcal{X}^j)$
  else
    $\tilde{\nabla} F^j = \nabla_s F(\mathcal{X}^j)$
  end
```

We believe that these calculus rules are promising on several topics related to derivative-free optimization. Future research directions include the following.

i. More testing and investigation on the numerical stability and accuracy of the calculus rules for the generalized simplex gradient and the generalized simplex calculus gradients.

ii. Exploring the calculus rules of the centered difference simplex gradient [25, Definition 6.2.3].

iii. Building a framework of calculus rules for an approximation of the Hessian.

iv. Building an accurate algorithm to approximate the Lipschitz constants of funtions hidden in blackboxes.

v. Application in DFO algorithms. Using the calculus rules for the generalized simplex gradient in DFO algorithms is a wide topic that is still untouched.

Bibliography

[1] S. Alarie, C. Audet, V. Garnier, S. Le Digabel, and L.-A. Leclaire, *Snow water equivalent estimation using blackbox optimization*, Pacific Journal of Optimization, 9 (2013), pp. 1–21. → pages 3

[2] P. Alberto, F. Nogueira, H. Rocha, and L. Vicente, *Pattern search methods for user-provided points: Application to molecular geometry problems*, SIAM Journal on Optimization, 14 (2004), pp. 1216–1236. → pages 3

[3] C. Audet, *A survey on direct search methods for blackbox optimization and their applications*, in Mathematics without boundaries, Springer, 2014, pp. 31–56. → pages 3

[4] C. Audet and W. Hare, *Derivative-free and blackbox optimization*, Springer Series in Operations Research and Financial Engineering, Springer, Switzerland, 2017. → pages 1, 2, 3, 4, 7, 10, 12, 13

[5] J. Bonnans, J. Gilbert, C. Lemaréchal, and C. Sagastizábal, *Numerical optimization: theoretical and practical aspects*, Springer Science & Business Media, 2006. → pages 1

[6] A. Booker, J. Dennis, P. Frank, D. Serafini, and V. Torczon, *Optimization using surrogate objectives on a helicopter test example*, in Computational Methods for Optimal Design and Control, Springer, 1998, pp. 49–58. → pages 3

[7] A. Booker, P. Frank, J. Dennis, D. Moore, and D. Serafini, *Managing surrogate objectives to optimize a helicopter rotor design-further experiments*, in 7th AIAA/USAF/NASA/ISSMO Symposium on Multidisciplinary Analysis and Optimization, 1998, p. 4717. → pages 3

[8] D. Bortz and C. Kelley, *The simplex gradient and noisy optimization problems*, in Computational methods for optimal design and control, Springer, 1998, pp. 77–90. → pages 2, 5

[9] P. Chartrand and A. Pelton, *Thermodynamic evaluation and optimization of the li, na, k, mg, ca, f, cl reciprocal system using the modified quasi-chemical model*, Metallurgical and Materials Transactions A, 32 (2001), pp. 1417–1430. → pages 3

[10] A. Conn, K. Scheinberg, and L. Vicente, *Geometry of interpolation sets in derivative-free optimization*, Mathematical Programming. A Publication of the Mathematical Programming Society, 111 (2008), pp. 141–172. → pages 4

[11] ——, *Geometry of sample sets in derivative-free optimization: polynomial regression and underdetermined interpolation*, IMA journal of numerical analysis, 28 (2008), pp. 721–748. → pages 4, 5

[12] A. Custódio, J. Dennis, and L. Vicente, *Using simplex gradients of nonsmooth functions in direct search methods*, IMA journal of numerical analysis, 28 (2008), pp. 770–784. → pages 4, 5

[13] A. Custódio, K. Scheinberg, and L. Vicente, *Methodologies and software for derivative-free optimization*, Advances and Trends in Optimization with Engineering Applications, (2017), pp. 495–506. → pages 3

[14] A. Custódio and L. Vicente, *Using sampling and simplex derivatives in pattern search methods*, SIAM Journal on Optimization, 18 (2007), pp. 537–555. → pages 5

[15] D. Echeverria Ciaurri, O. Isebor, and L. Durlofsky, *Application of derivative-free methodologies to generally constrained oil production optimisation problems*, International Journal of Mathematical Modelling and Numerical Optimisation, 2 (2011), pp. 134–161. → pages 3

[16] K. Eriksson, D. Estep, and C. Johnson, *Applied mathematics: Body and soul: Volume 1: Derivatives and geometry in IR3*, Springer Science & Business Media, 2013. → pages 34

[17] K. Fowler, J. Reese, C. Kees, J. Dennis, C. Kelley, C. Miller, C. Audet, A. Booker, G. Couture, R. Darwin, et al., *Comparison of derivative-free optimization methods for groundwater supply and*

hydraulic capture community problems, Advances in Water Resources, 31 (2008), pp. 743–757. → pages 3

[18] A. Gheribi, C. Audet, S. Le Digabel, E. Bélisle, C. Bale, and A. Pelton, *Calculating optimal conditions for alloy and process design using thermodynamic and property databases, the factsage software and the mesh adaptive direct search algorithm*, Calphad, 36 (2012), pp. 135–143. → pages 3

[19] A. Gheribi, C. Robelin, S. Le Digabel, C. Audet, and A. Pelton, *Calculating all local minima on liquidus surfaces using the factsage software and databases and the mesh adaptive direct search algorithm*, The Journal of Chemical Thermodynamics, 43 (2011), pp. 1323–1330. → pages 3

[20] C. Giuliani and E. Camponogara, *Derivative-free methods applied to daily production optimization of gas-lifted oil fields*, Computers & Chemical Engineering, 75 (2015), pp. 60–64. → pages 3

[21] W. Hare, *Compositions of convex functions and fully linear models*, Optimization Letters, 11 (2017), pp. 1217–1227. → pages 5

[22] W. Hare and G. Jarry-Bolduc, *Calculus identities for generalized simplex gradients: Rules and applications*, Submitted to SIAM Journal on Optimization, (2018). → pages v

[23] R. Hooke and T. Jeeves, *Direct search solution of numerical and statistical problems*, Journal of the ACM (JACM), 8 (1961), pp. 212–229. → pages 4

[24] R. Horn and C. Johnson, *Matrix analysis*, Cambridge university press, 1990. → pages 8

[25] C. Kelley, *Iterative methods for optimization*, vol. 18, SIAM, 1999. → pages 5, 62

[26] M. Martinez and J. Meza, *On the use of direct search methods for the molecular conformation problem*, Journal of Computational Chemistry, 15, pp. 627–632. → pages 3

[27] M. Minville, D. Cartier, C. Guay, L.-A. Leclaire, C. Audet, S. Le Digabel, and J. Merleau, *Improving process representation in conceptual hydrological model calibration using climate simulations*, Water Resources Research, 50 (2014), pp. 5044–5073. → pages 3

[28] J. Moré, B. Garbow, and K. Hillstrom, *Testing unconstrained optimization software*, Association for Computing Machinery. Transactions on Mathematical Software, 7 (1981), pp. 17–41. → pages 6, 50, 55

[29] P. Muguntharan, C. Shoemaker, and R. Regis, *Comparison of function approximation, heuristic, and derivative-based methods for automatic calibration of computationally expensive groundwater bioremediation models*, Water Resources Research, 41 (2005). → pages 3

[30] J. Nelder and R. Mead, *A simplex method for function minimization*, The computer journal, 7 (1965), pp. 308–313. → pages 4

[31] D. Patterson and J. Hennessy, *Computer Organization and Design: The Hardware/Software Interface*, Newnes, 2013. → pages 53

[32] M. Powell, *A direct search optimization method that models the objective and constraint functions by linear interpolation*, in Advances in optimization and numerical analysis, Springer, 1994, pp. 51–67. → pages 4

[33] R. Regis, *The calculus of simplex gradients*, Optimization Letters, 9 (2015), pp. 845–865. → pages iii, 5, 14, 15, 16, 17, 22

[34] K. Scheinberg and P. Toint, *Self-correcting geometry in model-based algorithms for derivative-free unconstrained optimization*, SIAM Journal on Optimization, 20 (2010), pp. 3512–3532. → pages 5

[35] D. Serafini, *A framework for managing models in nonlinear optimization of computationally expensive functions*, PhD thesis, Rice University, 1999. → pages 3

[36] I. Stewart, *Galois theory*, CRC Press, Boca Raton, FL, fourth ed., 2015. → pages 33

[37] D. Winfield, *Function and functional optimization by interpolation in data tables*, PhD thesis, Harvard University, 1970. → pages 4

Appendix

Table 6: An Example Where $\nabla_{spe}(fg)(\mathcal{X})$ is the Most Stable

β_m	**RE** $\nabla_s(fg)(\mathcal{X})$	**RE** $\nabla_{spe}(fg)(\mathcal{X})$	**RE** $\nabla_{sp}(fg)(\mathcal{X})$
10^{0}	1.2500e-01	1.2500e-01	0.0000e+00
10^{-1}	1.2500e-02	1.2500e-02	0.0000e+00
10^{-2}	1.2500e-03	1.2500e-03	0.0000e+00
10^{-3}	1.2500e-04	1.2500e-04	0.0000e+00
10^{-4}	1.2500e-05	1.2500e-05	0.0000e+00
10^{-5}	1.2500e-06	1.2500e-06	0.0000e+00
10^{-6}	1.2479e-07	1.2500e-07	0.0000e+00
10^{-7}	1.3323e-08	1.2500e-08	0.0000e+00
10^{-8}	**0.0000e+00**	1.2500e-09	0.0000e+00
10^{-9}	0.0000e+00	1.2500e-10	0.0000e+00
10^{-10}	0.0000e+00	1.2500e-11	0.0000e+00
10^{-11}	1.1102e-16	1.2499e-12	**1.1102e-16**
10^{-12}	0.0000e+00	1.2501e-13	0.0000e+00
10^{-13}	0.0000e+00	1.2434e-14	0.0000e+00
10^{-14}	0.0000e+00	1.3323e-15	0.0000e+00
10^{-15}	0.0000e+00	**0.0000e+00**	0.0000e+00

Table 7: An Example Where $\nabla_{sp}(fg)(\mathcal{X})$ is the Most Stable

β_m	**RE** $\nabla_s(fg)(\mathcal{X})$	**RE** $\nabla_{spe}(fg)(\mathcal{X})$	**RE** $\nabla_{sp}(fg)(\mathcal{X})$
10^{0}	4.4409e-16	2.2204e-16	1.7157e-01
10^{-1}	**0.0000e+00**	**8.8818e-16**	2.3823e-02
10^{-2}	2.2204e-14	1.6320e-14	2.4876e-03
10^{-3}	0.0000e+00	1.0814e-13	2.4988e-04
10^{-4}	2.2206e-12	1.8795e-12	2.4999e-05
10^{-5}	2.2205e-11	1.2707e-11	2.5000e-06
10^{-6}	2.2204e-10	1.9969e-10	2.4980e-07
10^{-7}	0.0000e+00	5.7509e-10	2.4425e-08
10^{-8}	0.0000e+00	2.5000e-09	**0.0000e+00**

Table 8: An Example Where $\nabla_s f^n(\mathcal{X})$ is the Most Stable

β_m	**RE** $\nabla_s f^n(\mathcal{X})$	**RE** $\nabla_{spe} f^n(\mathcal{X})$	**RE** $\nabla_{sp} f^n(\mathcal{X})$
10^0	8.1428e+00	8.1428e+00	4.1759e-16
10^{-1}	6.0580e-01	6.0580e-01	1.3205e-16
10^{-2}	6.4301e-02	6.4301e-02	**1.3787e-15**
10^{-3}	6.4721e-03	6.4721e-03	3.2365e-14
10^{-4}	6.4763e-04	6.4763e-04	1.5178e-13
10^{-5}	6.4767e-05	6.4767e-05	1.7094e-13
10^{-6}	6.4768e-06	6.4768e-06	3.1848e-11
10^{-7}	6.4764e-07	6.4762e-07	6.7907e-11
10^{-8}	6.7528e-08	6.6897e-08	2.1322e-09
10^{-9}	2.8393e-08	2.1361e-08	1.5138e-08
10^{-10}	2.2519e-08	**5.3541e-08**	5.4180e-08
10^{-11}	**2.4845e-06**	1.5436e-06	1.5436e-06

Table 9: An Example Where $\nabla_{spe} f^n(\mathcal{X})$ is the Most Stable

β_m	**RE** $\nabla_s f^n(\mathcal{X})$	**RE** $\nabla_{spe} f^n(\mathcal{X})$	**RE** $\nabla_{sp} f^n(\mathcal{X})$
10^0	2.7894e-01	2.7894e-01	0.0000e+00
10^{-1}	2.5279e-02	2.5279e-02	0.0000e+00
10^{-2}	2.5028e-03	2.5028e-03	0.0000e+00
10^{-3}	2.5003e-04	2.5003e-04	0.0000e+00
10^{-4}	2.5000e-05	2.5000e-05	0.0000e+00
10^{-5}	2.5000e-06	2.5000e-06	0.0000e+00
10^{-6}	2.4997e-07	2.5000e-07	0.0000e+00
10^{-7}	2.4672e-08	2.5000e-08	0.0000e+00
10^{-8}	9.8686e-09	2.5000e-09	0.0000e+00
10^{-9}	**1.3158e-07**	2.5000e-10	0.0000e+00
10^{-10}	6.5791e-07	2.5000e-11	0.0000e+00
10^{-11}	3.2895e-06	2.4999e-12	**1.3158e-16**
10^{-12}	6.5785e-05	2.5001e-13	0.0000e+00
10^{-13}	1.6388e-03	2.5132e-14	0.0000e+00
10^{-14}	3.3670e-03	2.5001e-15	0.0000e+00
10^{-15}	1.8519e-01	2.6316e-16	0.0000e+00
10^{-16}	1.0000e+00	**1.0000e+00**	1.0000e+00

Table 10: An Example Where $\nabla_{sp} f^n(\mathcal{X})$ is the Most Stable

β_m	**AE** $\nabla_s f^n(\mathcal{X})$	**AE** $\nabla_{spe} f^n(\mathcal{X})$	**AE** $\nabla_{sp} f^n(\mathcal{X})$
10^0	1.4142e+00	1.4142e+00	0.0000e+00
10^{-1}	1.4142e-49	1.4142e-49	0.0000e+00
10^{-2}	1.4142e-98	1.4142e-98	0.0000e+00
10^{-3}	1.4142e-147	1.4142e-147	0.0000e+00
10^{-4}	1.4142e-196	1.4142e-196	0.0000e+00
10^{-5}	1.4142e-245	1.4142e-245	0.0000e+00
10^{-6}	1.4142e-294	1.4142e-294	0.0000e+00
10^{-7}	**0.0000e+00**	**0.0000e+00**	0.0000e+00

Table 11: An Example Where $\nabla_s \left(\frac{f}{g}\right)(\mathcal{X})$ is Stable and Accurate

β_m	**RE** $\nabla_s(\frac{f}{g})(\mathcal{X})$	**RE** $\nabla_{sqe}(\frac{f}{g})(\mathcal{X})$	**RE** $\nabla_{sq}(\frac{f}{g})(\mathcal{X})$
10^0	0.0000e+00	0.0000e+00	2.5000e-01
10^{-1}	0.0000e+00	**3.9968e-15**	2.5000e-02
10^{-2}	0.0000e+00	5.7732e-15	2.5000e-03
10^{-3}	0.0000e+00	2.5691e-13	2.5000e-04
10^{-4}	0.0000e+00	2.0690e-12	2.5000e-05
10^{-5}	0.0000e+00	4.4202e-11	2.5000e-06
10^{-6}	0.0000e+00	4.2186e-10	2.4958e-07
10^{-7}	0.0000e+00	1.6454e-09	2.6645e-08
10^{-8}	0.0000e+00	2.5000e-09	**0.0000e+00**
10^{-9}	0.0000e+00	2.5000e-10	0.0000e+00
10^{-10}	0.0000e+00	2.5000e-11	0.0000e+00
10^{-11}	**1.1102e-16**	2.5001e-12	1.1102e-16

Table 12: An Example Where $\nabla_{sqe}\left(\frac{f}{g}\right)(\mathcal{X})$ is the Most Stable

β_m	**(1) RE** $\nabla_s(\frac{f}{g})(\mathcal{X})$	**(2) RE** $\nabla_{sqe}(\frac{f}{g})(\mathcal{X})$	**(3) RE** $\nabla_{sq}(\frac{f}{g})(\mathcal{X})$
10^{0}	4.8836e-01	4.8836e-01	1.8907e-01
10^{-1}	8.8366e-02	8.8366e-02	2.4197e-02
10^{-2}	9.6180e-03	9.6180e-03	2.4917e-03
10^{-3}	9.7038e-04	9.7038e-04	2.4992e-04
10^{-4}	9.7125e-05	9.7125e-05	2.4999e-05
10^{-5}	9.7134e-06	9.7134e-06	2.5000e-06
10^{-6}	9.7151e-07	9.7137e-07	2.5002e-07
10^{-7}	9.7287e-08	9.6560e-08	2.4425e-08
10^{-8}	1.1768e-08	7.2135e-09	**0.0000e+00**
10^{-9}	**1.4326e-07**	7.2135e-10	0.0000e+00
10^{-10}	1.4234e-06	7.2135e-11	0.0000e+00
10^{-11}	5.6907e-06	7.2134e-12	0.0000e+00
10^{-12}	5.1831e-05	7.2138e-13	0.0000e+00
10^{-13}	6.5773e-04	7.2117e-14	0.0000e+00
10^{-14}	2.6846e-03	7.4677e-15	0.0000e+00
10^{-15}	3.9094e-02	6.4009e-16	0.0000e+00
10^{-16}	1.0000e+00	**1.0000e+00**	1.0000e+00

Table 13: An Example Where $\nabla_{sq}\left(\frac{f}{g}\right)(\mathcal{X})$ is Stable and Accurate

β_m	**(1) RE** $\nabla_s(\frac{f}{g})(\mathcal{X})$	**(2) RE** $\nabla_{sqe}(\frac{f}{g})(\mathcal{X})$	**(3) RE** $\nabla_{sq}(\frac{f}{g})(\mathcal{X})$
10^0	1.0000e+00	1.0000e+00	2.0000e-16
10^{-1}	1.0000e+00	1.0000e+00	2.0000e-16
10^{-2}	1.0000e+00	1.0000e+00	2.0000e-16
10^{-3}	9.9999e-01	9.9999e-01	2.0000e-16
10^{-4}	9.9990e-01	9.9990e-01	2.0000e-16
10^{-5}	9.9900e-01	9.9900e-01	2.0000e-16
10^{-6}	9.9010e-01	9.9010e-01	2.0000e-16
10^{-7}	9.0909e-01	9.0909e-01	2.0000e-16
10^{-8}	5.0000e-01	5.0000e-01	2.0000e-16
10^{-9}	9.0909e-02	9.0909e-02	2.0000e-16
10^{-10}	9.9010e-03	9.9010e-03	2.0000e-16
10^{-11}	9.9900e-04	9.9900e-04	2.0000e-16
10^{-12}	9.9990e-05	9.9990e-05	2.0000e-16
10^{-13}	9.9999e-06	9.9999e-06	2.0000e-16
10^{-14}	1.0000e-06	1.0000e-06	2.0000e-16
10^{-15}	1.0016e-07	1.0000e-07	2.0000e-16
10^{-16}	1.5220e-08	1.0000e-08	2.0000e-16
10^{-17}	**4.3380e-08**	1.0000e-09	2.0000e-16
10^{-18}	3.8723e-07	1.0000e-10	2.0000e-16
10^{-19}	6.8405e-06	1.0000e-11	2.0000e-16
10^{-20}	4.4430e-05	1.0002e-12	2.0000e-16
10^{-21}	6.3422e-04	1.0020e-13	2.0000e-16
10^{-22}	5.8039e-03	1.0200e-14	2.0000e-16
10^{-23}	5.0840e-02	1.2000e-15	2.0000e-16
10^{-24}	9.9280e-02	4.0000e-16	2.0000e-16
10^{-25}	1.0000e+00	**1.0000e+00**	**1.0000e+00**

Table 14: An Example Where $\nabla_s f^{-n}(\mathcal{X})$ is the Most Stable

β_m	**RE** $\nabla_s f^{-n}(\mathcal{X})$	**RE** $\nabla_{spe} f^{-n}(\mathcal{X})$	**RE** $\nabla_{sp} f^{-n}(\mathcal{X})$
10^{0}	9.7989e-01	9.7989e-01	1.5771e-16
10^{-1}	5.0233e-01	5.0233e-01	1.5771e-16
10^{-2}	1.3970e-01	1.3970e-01	**9.4624e-16**
10^{-3}	1.6070e-02	1.6070e-02	1.1039e-15
10^{-4}	1.6309e-03	1.6309e-03	9.4466e-14
10^{-5}	1.6334e-04	1.6334e-04	2.5911e-13
10^{-6}	1.6336e-05	1.6336e-05	7.3993e-12
10^{-7}	1.6336e-06	1.6336e-06	2.3877e-11
10^{-8}	1.6446e-07	1.6444e-07	1.0746e-09
10^{-9}	1.3732e-08	1.3566e-08	2.7702e-09
10^{-10}	1.1368e-08	**1.7614e-08**	1.9248e-08
10^{-11}	**2.6473e-07**	3.6540e-07	3.6524e-07

Table 15: An Example where $\nabla_{spe} f^{-n}(\mathcal{X})$ is the Most Stable

β_m	**RE** $\nabla_s f^{-n}(\mathcal{X})$	**RE** $\nabla_{spe} f^{-n}(\mathcal{X})$	**RE** $\nabla_{sp} f^{-n}(\mathcal{X})$
10^{0}	3.9657e-01	3.9657e-01	0.0000e+00
10^{-1}	5.5835e-02	5.5835e-02	0.0000e+00
10^{-2}	5.8075e-03	5.8075e-03	0.0000e+00
10^{-3}	5.8307e-04	5.8307e-04	0.0000e+00
10^{-4}	5.8331e-05	5.8331e-05	0.0000e+00
10^{-5}	5.8333e-06	5.8333e-06	0.0000e+00
10^{-6}	5.8329e-07	5.8333e-07	0.0000e+00
10^{-7}	5.6712e-08	5.8333e-08	0.0000e+00
10^{-8}	2.7539e-09	5.8333e-09	0.0000e+00
10^{-9}	**8.7820e-08**	5.8333e-10	0.0000e+00
10^{-10}	8.7820e-07	5.8333e-11	0.0000e+00
10^{-11}	8.7820e-06	5.8335e-12	**1.5808e-16**
10^{-12}	6.2228e-05	5.8346e-13	0.0000e+00
10^{-13}	1.7198e-03	5.8330e-14	0.0000e+00
10^{-14}	3.1516e-03	5.2165e-15	0.0000e+00
10^{-15}	6.7871e-02	6.3231e-16	0.0000e+00
10^{-16}	1.0000e+00	**1.0000e+00**	1.0000e+00

Table 16: An Example where $\nabla_{sp} f^{-n}(\mathcal{X})$ is Stable and Accurate

β_m	**RE** $\nabla_s f^{-n}(\mathcal{X})$	**RE** $\nabla_{spe} f^{-n}(\mathcal{X})$	**RE** $\nabla_{sp} f^{-n}(\mathcal{X})$
10^{0}	9.9922e-01	9.9922e-01	0.0000e+00
10^{-1}	9.9219e-01	9.9219e-01	0.0000e+00
10^{-2}	9.2201e-01	9.2201e-01	0.0000e+00
10^{-3}	4.6869e-01	4.6869e-01	0.0000e+00
10^{-4}	7.2437e-02	7.2437e-02	0.0000e+00
10^{-5}	7.6344e-03	7.6344e-03	0.0000e+00
10^{-6}	7.6754e-04	7.6754e-04	0.0000e+00
10^{-7}	7.6795e-05	7.6795e-05	0.0000e+00
10^{-8}	7.6800e-06	7.6800e-06	0.0000e+00
10^{-9}	7.6796e-07	7.6800e-07	0.0000e+00
10^{-10}	7.7195e-08	7.6800e-08	0.0000e+00
10^{-11}	8.6736e-09	7.6800e-09	0.0000e+00
10^{-12}	**0.0000e+00**	7.6800e-10	0.0000e+00
10^{-13}	0.0000e+00	7.6800e-11	0.0000e+00
10^{-14}	0.0000e+00	7.6799e-12	0.0000e+00
10^{-15}	0.0000e+00	7.6810e-13	0.0000e+00
10^{-16}	0.0000e+00	7.6561e-14	0.0000e+00
10^{-17}	0.0000e+00	7.9936e-15	0.0000e+00
10^{-18}	0.0000e+00	7.1054e-16	0.0000e+00
10^{-19}	1.0000e+00	**1.0000e+00**	**1.0000e+00**

Table 17: An Example Where $\nabla_s(f \circ g)(\mathcal{X})$ is the Most Stable

β_m	**RE** $\nabla_s(f \circ g)(\mathcal{X})$	**RE** $\nabla_{sce}(f \circ g)(\mathcal{X})$	**RE** $\nabla_{sc}(f \circ g)(\mathcal{X})$
10^0	1.8049e-01	1.8350e-01	1.8401e-01
10^{-1}	1.8049e-01	**1.8376e-01**	1.8089e-01
10^{-2}	1.8049e-01	1.8375e-01	**1.8375e-01**
10^{-3}	1.8049e-01	1.8354e-01	1.8350e-01
10^{-4}	1.8049e-01	1.8350e-01	1.8351e-01
10^{-5}	1.8049e-01	1.8389e-01	1.8389e-01
10^{-6}	1.8049e-01	1.8350e-01	1.8350e-01
10^{-7}	1.8049e-01	1.8350e-01	1.8350e-01
10^{-8}	1.8049e-01	1.8350e-01	1.8350e-01
10^{-9}	1.8049e-01	1.8350e-01	1.8350e-01
10^{-10}	1.8049e-01	1.8350e-01	1.8350e-01
10^{-11}	1.8049e-01	1.8350e-01	1.8350e-01
10^{-12}	1.8049e-01	1.8352e-01	1.8352e-01
10^{-13}	1.8049e-01	1.8361e-01	1.8361e-01
10^{-14}	**1.8089e-01**	1.8619e-01	1.8619e-01

Table 18: An Example Where $\nabla_{sce}(f \circ g)(\mathcal{X})$ is the Most Stable

β_m	**RE** $\nabla_s(f \circ g)(\mathcal{X})$	**RE** $\nabla_{sce}(f \circ g)(\mathcal{X})$	**RE** $\nabla_{sc}(f \circ g)(\mathcal{X})$
10^0	1.8626e-16	1.8626e-16	3.7253e-16
10^{-1}	**1.8627e-16**	5.4817e-22	**5.5879e-16**
10^{-2}	2.1233e-21	**1.8626e-16**	1.8626e-16

Table 19: An Example Where $\nabla_{sc}(f \circ g)(\mathcal{X})$ is the Most Stable

β_m	**RE** $\nabla_s(f \circ g)(\mathcal{X})$	**RE** $\nabla_{sce}(f \circ g)(\mathcal{X})$	**RE** $\nabla_{sc}(f \circ g)(\mathcal{X})$
10^0	0.0000e+00	0.0000e+00	8.9792e-03
10^{-1}	**2.2204e-16**	**2.2204e-16**	2.1395e-04
10^{-2}	0.0000e+00	1.1102e-16	2.3926e-06
10^{-3}	1.1102e-16	0.0000e+00	2.4204e-08
10^{-4}	2.2204e-16	2.2204e-16	2.4232e-10
10^{-5}	4.4409e-16	3.3307e-16	2.4238e-12
10^{-6}	0.0000e+00	0.0000e+00	2.4203e-14
10^{-7}	1.1102e-16	2.2204e-16	2.2204e-16
10^{-8}	2.2204e-16	2.2204e-16	**0.0000e+00**

Index