A High-Order Accurate Particle-in-Cell Method

by

Essex Edwards

BCIS, The University of the Fraser Valley, 2007

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

Master of Science

in

THE FACULTY OF GRADUATE STUDIES
(Computer Science)

The University Of British Columbia
(Vancouver)

June 2010

Abstract

We propose the use of high-order accurate interpolation and approximation schemes alongside high-order accurate time integration methods to enable high-order accurate Particle-in-Cell methods. The key insight is to view the unstructured set of particles as the underlying representation of the continuous fields; the grid used to evaluate integro-differential coupling terms is purely auxiliary. We also include a novel regularization term to avoid the accumulation of noise in the particle samples without harming the convergence rate. We include numerical examples for several model problems: advection-diffusion, shallow water, and incompressible Navier-Stokes in vorticity formulation. The implementation demonstrates fourth-order convergence, shows very low numerical dissipation, and is competitive with high-order accurate Eulerian schemes.

Table of Contents

List of Figures

Glossary

CFL Courant-Friedrichs-Lewy condition

DOF degrees of freedom

ENO Essentially Non-Oscillatory

FEM Finite Element Method

FFT Fast Fourier Transform

FFTW Fastest Fourier Transform in the West

FLIP Fluid Implicit Particle

GIMP Generalized Interpolation Material Point

HOPIC High-Order Particle-in-Cell

LAPACK Linear Algebra Package

LES Large-Eddy Simulation

MLS Moving Least Squares

MPM Material Point Method

ODE Ordinary Differential Equation

PDE Partial Differential Equation

PIC Particle-in-Cell

RK4 classical fourth-order Runge-Kutta time integration

WENO Weighted Essentially Non-Oscillatory

WLS Weighted Least Squares

Acknowledgments

This thesis would not have existed without the support of a great many people.

At UBC, I have had the pleasure of working with many brilliant people. My supervisor, Robert Bridson, has been wonderful to work with. I have learned much from him in the last year. I would also like to thank my lab-mates, both in the Scientific Computing Lab and in Imager Lab. I appreciate all the discussions with them, from academic to personal, and serious to light-hearted.

I would also like to thank Noham Weinberg for the opportunities he gave me at UFV and for his encouragement to go to grad school. I look back fondly on my time working with him, Elna, Liam, Jeff, Johan, and everyone else at UFV.

Most of all, I appreciate the encouragement and support I have received from from my family, friends, and especially Elna. They were always there, unwaveringly.

Chapter 1

Introduction

The Particle-in-Cell (PIC) method [17] is over half a century old, and while it has a long history of robustly handling transport problems [18], its limitation to first-order accuracy has led to its eclipse by modern Eulerian schemes and others (e.g., [2, 13]). In this thesis, we demonstrate how to increase the order of accuracy of PIC, leading to High-Order Particle-in-Cell (HOPIC), illustrated with fourth-order accurate discretizations of a variety of equations. We believe this provides a fruitful avenue of exploration for robust, efficient, and high-order accurate schemes with remarkably low numerical diffusion and dispersion.

The fundamental representation of the solution for HOPIC is a set of unstructured (albeit well-distributed) particles that store the primary variables of the equations along with their position. Unlike many particle methods [7], we view these as massless samples of continuous fields, rather than actual "blobs" of material. Similar to the Method-of-Lines [25], we evolve the positions and values of these particles with an arbitrary time integration scheme designed for Ordinary Differential Equations (ODEs), such as a Runge-Kutta method [1] . To evaluate integral or differential terms that couple the values of the samples through space (including, typically, the velocity field itself), we first transfer the particle values to an auxiliary background grid using high-order Moving Least Squares (MLS), evaluate the terms

[1] A symplectic method may be more appropriate for conservation and Hamiltonian problems, however not all problems we consider fit into those categories, and our spatial discretizations do not preserve those properties.

on the grid (using whatever is most convenient: finite differences, pseudo-spectral methods, etc.), and use a high-order method to interpolate those terms from the grid back to the particles. Finally, we include an artificial regularization term on the particles that dampens components of the solution not resolved on the grid, i.e., noise, without disturbing the convergence rate.

In Section 1.1 we place HOPIC in the context of previous work before delineating the details of the algorithm in Chapter 2 for generic equations. In Chapter 3 we discuss some details of our implementation. Chapter 4 provides numerical examples of HOPIC applied to a variety of equations in 2D and 3D, demonstrating fourth-order accuracy. The focus here is on the core particle-grid transfer operations and related regularization; we defer an investigation of the interaction between HOPIC and boundary conditions to future work, assuming periodicity for now.

Our primary contribution is the first globally high-order accurate PIC method. This includes the use of high-order approximation and interpolation schemes to transfer to and from the grid, and novel regularization term to avoid the accumulation of noise, without disrupting the convergence rate. We also suggest a simple reseeding strategy that maintains a quality distribution of particles, ensuring that a high-order approximation using MLS is always possible. Finally, we validate a fourth-order version of HOPIC on several example problems with analytic solutions, as well as comparison to real-world experiments.

1.1 Related Work

PIC methods [17] attempt to combine the advantages of both grid and particle methods. Particles are used to advect all transported quantities, while a more convenient grid is used to solve the non-advection portion of the problem.

The original PIC [17] and subsequent schemes treat the particles as "blobs" of material. To approximate the particles on a grid, the extensive properties of each particle are distributed as a weighted combination to nearby grid points so as to conserve mass and other extensive properties. In its earliest incarnation, only the nearest gridpoint was used, but over time smoother schemes were used. These high-order weighting functions improve the results, but do not raise the convergance of the algorithms beyond first order.

The early PIC schemes interpolated the grid values to the particles after each step. This caused significant numerical diffusion, and prevented convergence with respect to the time step independent of grid spacing. Fluid Implicit Particle (FLIP) [6, 8], a PIC-derived scheme, essentially eliminated this dissipation by interpolating the change in grid values to the particles, rather than the new value itself. With the exception of the extensive property conservation principles described above, our HOPIC scheme reduces to FLIP when using Forward Euler and appropriate first-order interpolation schemes.

The Material Point Method (MPM) [27] is a finite element method developed from PIC and FLIP and applied to solid mechanics problems; the Generalized Interpolation Material Point (GIMP) method [3] later improved and provided a general framework for MPM. GIMP builds a model of the particles as localized weighted integrals of the underlying continuum. However, the GIMP method for converting between particles and the grid is only first-order accurate beyond 1D [26].

Wallstedt and Guilkey [29] recently extended GIMP to second-order convergence by using Weighted Least Squares (WLS) to interpolate between the particles and the grid. The use of WLS, and their model of the particles as samples of the continuum, is similar to HOPIC. While their work is focused on solid mechanics, we present a more general framework for solving transport problems with a concentration on fluids, demonstrate higher-order accuracy, include particle reseeding to avoid WLS's regression to first order where particle distribution becomes too sparse, and crucially address the sub-grid noise issue with a novel regularization term.

In the context of geophysical simulation, Moresi [23] reinterpreted MPM as a type of Finite Element Method (FEM) with the particle locations as quadrature points. They compute quadrature weights such that affine functions would be reconstructed exactly, giving a second-order accurate reconstruction. However, their entire algorithm does not achieve second-order accuracy, due to other low-order approximations. This interpretation of MPM, and the reweighting technique, do not appear to have been picked up by the MPM community at large.

For incompressible fluid flow, an alternative particle formulation is the vortex particle method [11] which is potentially high-order accurate [4]. The fundamental representation is particles carrying a kernel of vorticity with compact support. This produces a smooth global velocity field, but reveals how specific this method is to the vorticity formulation; there is no natural mechanism to handle equations besides vorticity including highly viscous flow, or entirely different problems such as the shallow-water equations.

Chapter 2

The Generic Method

Consider a transport Partial Differential Equation (PDE) of the form:

$$\frac{\partial \vec{q}}{\partial t} + \vec{u} \cdot \nabla \vec{q} = \mathscr{L} \vec{q} \tag{2.1}$$

for time $t > 0$ and over some spatial domain Ω, where $\vec{q}$ is a vector-valued field containing all transported quantities, $\vec{u}$ is the velocity field (possibly derived from $\vec{q}$), and $\mathscr{L}$ is an operator which provides any spatial coupling or forcing terms such as diffusion. We neglect boundary conditions on Ω, and assume initial conditions at $t = 0$ are given. Sections 4.1, 4.2 and 4.3 provide specific examples: advection-diffusion, shallow water, and vorticity respectively. In the present work we do not admit additional constraint equations, such as incompressibility in the velocity-pressure form of the Navier-Stokes equations.

Our method begins by sampling the entire domain with particles. Their distribution is unstructured, but must be sufficiently dense, as described in section 2.1. A particle carries a value $\vec{q}_i$ which is interpreted as the value of the continuous field at the particle's position, i.e. $\vec{q}_i = \vec{q}(\vec{x}_i)$. Initial conditions are set this way from a specification of $\vec{q}$ at time 0. These particles are the complete representation of the problem, and constitute the entire state.

Concatenating all m particle values together, we arrive at two long vectors, $\mathbf{x} = (\vec{x}_1, \ldots, \vec{x}_m)$ and $\mathbf{q} = (\vec{q}_1, \ldots, \vec{q}_m)$ representing the state, which we evolve according to the system of ODEs

$$\frac{d}{dt}\begin{bmatrix}\mathbf{x}\\ \mathbf{q}\end{bmatrix} = \begin{bmatrix}\mathbf{u}\\ \mathscr{L}\vec{q}+\mathscr{R}\end{bmatrix} = \mathbf{f}(\mathbf{x},\mathbf{q}) \tag{2.2}$$

where $\mathbf{u} = (\vec{u}_1,\dots,\vec{u}_m)$ is the vector of all velocities for the particles. This is advanced discretely using an explicit time-integration scheme.

The core of our algorithm is in how $\mathbf{f}(\mathbf{x},\mathbf{q})$ is evaluated. We first approximate the field $\vec{q}(\vec{x})$ on a regular rectangular grid, using scattered data approximation from the particles described in section 2.2. On the grid, any technique can be applied to compute $\vec{f}$ at the grid vertices (e.g. finite differences); this result is then interpolated back to the particles, as discussed in section 2.3. Finally, a regularization term $\mathscr{R}$ is included in the evolution to prevent sub-grid noise (section 2.4).

Each of these steps must be high-order accurate for the entire scheme to be high-order accurate. Additionally, for convergence with shrinking Δt and fixed Δx, the time-integration scheme requires $\mathbf{f}$ to be sufficiently smooth; for example, using fourth-order Runge-Kutta (RK4), $\mathbf{f}$ should be several times differentiable. The $\mathbf{f}$ defined in this paper is only C^0 (due to the grid-to-particle interpolation) and, consequently, we observe only lower-order convergence with a fixed Δx and shrinking Δt. However, subject to a Courant-Friedrichs-Lewy (CFL) condition, $\Delta t \propto \Delta x$, the smoothness requirement appears to be unnecessary and fourth-order convergence is observed. We always act under this regime, and thus do not need additional smoothness beyond continuity of $\mathbf{f}$.

2.1 Particle Seeding and Reseeding

To reliably reconstruct a continuous field from scattered particles, those particles must be sufficiently well distributed: in particular, we need a sufficient number in the support of the MLS kernel at each grid point, arranged non-degenerately, to attain a well-posed least-squares problem. However, quality distributions can be quickly destroyed by advection through a distorting velocity field. Compressible flow fields naturally cause some areas to become under-sampled, and even incompressible flows with shearing can cause clumps and sparse areas to develop over time. Even though our interpretation of particles as mere samples of the underlying fields, as opposed to discrete blobs of material, gives a method that is robust to minor variations in sampling density, significant gaps must be filled and extreme oversampling is computationally wasteful.

The particle-to-grid approximation requires a minimum number of particles in the neighborhood of each grid point. This is easily maintained by adding particles to any cell which has too few. However, reseeding can only be done between timesteps, not before the evaluation of **f** by each sub-step of RK4 (as that would change the length of the state-vector being accumulated). So, the distribution must be good enough to maintain this minimum across multiple sub-steps. To that end, we maintain a higher particle density than strictly necessary; this also helps with the robustness of the particle-to-grid approximation process.

On the simulation grid, we delete random particles from any cell with too many ($n > 10$ in 3D), and add particles at jittered positions near the center of any cell which has too few ($n < 6$ in 3D). Field values at the newly added particles are interpolated from the grid. With incompressible 3D flow, typical turnover is 0.5% of particles per step.

We emphasize the use of random sampling to seed particles, unlike earlier PIC schemes which employ structured distributions. While structured distributions offer slightly higher quality reconstruction for a single time step, they are much less robust to deformation. In particular, even a simple shearing flow may compress a structured distribution along one axis while opening up large gaps along another; a uniform random distribution remains uniform under the same flow. See figure 2.1 for an illustration of this phenomenon.

2.2 Particle-to-Grid Approximation

Existing PIC methods approximate the particles with a grid using several approaches. One common approach is to conserve properties of the particles, such as total mass or momentum, when transfered to the grid [6, 8, 17], consistent with the interpretation of the particles as blobs of material. Another approach [9] approximates the inverse of the grid-to-particle interpolation by solving for grid values which minimize the mean squared error when interpolated back to the particles. The accuracy of this approach depends on the grid-to-particle interpolant, and may be of high-order accuracy; however, it comes at the cost of a global (and potentially nonlinear, depending upon the interpolation) solve.

In this thesis, we instead use MLS [20] to approximate on the grid the field sampled by the particles. MLS provides a general purpose algorithm for fitting to scattered data, of arbitrary order of accuracy and degree of smoothness depending on the choice of basis and weighting function.

We interpret the grid vertices to be point samples of the continuum, analogous to the particles. Given these interpretations, interpolating MLS would be a natural choice. However, as particles may accumulate some noise over time (see section 2.4) and interpolating MLS overfits this noise, we instead use a simpler approximating MLS to fit through the noise.

The particular choice of weighting function appears to have little effect on the results. Because of its simplicity, we use the following C^3 spline

$$w(r) \quad = \quad \begin{cases} \left[1-\left(\frac{r}{\Delta x}\right)^2\right]^4 & \text{if } |r| \leq \Delta x \\ 0 & \text{otherwise} \end{cases} \tag{2.3}$$

in 1D and tensor products of the same in higher dimensions. Only the particles in cells adjacent to a grid vertex are in the support of the MLS kernel, both simplifying the implementation and making it faster than broader or more irregular supports. In our experiments, results were essentially the same with non-grid-aligned kernels and radially symetric kernels.

To achieve fourth-order accuracy in our examples, we use cubic polynomials for the least-squares fit. This leads to four degrees of freedom (DOF) in the 1D local least-squares problem, 10 DOF in 2D, and 20 DOF in 3D; clearly we need at least this many particles in the support of the kernel, and not in a degenerate position (e.g. all in a straight line), to get a well-posed least-squares problem. Our reseeding strategy is designed to always achieve this.

In this thesis we restrict our attention to problems with smooth solutions. In the presence of shocks, kinks, or other sharp features, plain MLS can lead to severe overshoots and thence instability; a nonlinear approach to limit overshoot would be required, such as in Quasi-ENO MLS [16].

2.3 Grid-to-Particle Interpolation

Using a regular rectangular auxiliary grid admits a wide variety of existing interpolation schemes. The Weighted Essentially Non-Oscillatory (WENO) schemes are particularly attractive for their simplicity and robust behaviour around sharp features [1]. In our examples we use the fourth-order WENO scheme as described, for example, by Macdonald and Ruuth [21]. This uses a 4-point stencil in 1D, and in higher dimensions, is computed one dimension at a time, giving a 16-point stencil in 2D, and 64 points in 3D. Because it is nonlinear, the order in which the dimensions are applied has an effect, but the difference is $\mathrm{O}(\Delta x^4)$ and thus may be neglected.

2.4 Regularization

In any non-trivial situation, the particles in FLIP and similar variants of PIC may accumulate noise: variations of $\vec{q}$ that are not resolved on the grid, and thus are invisible to the physics $\mathscr{L}$ of the problem, may persist, grow, and otherwise behave non-physically. Some PIC algorithms suffer additional noise due to instabilities and artifacts of the method [5]; however, noise is unavoidable even in an ideal method simply due to the greater number of particles than grid points.

To prevent noise from continuing to accumulate, we include a regularization term in equation (2.1) inspired by the original noise-free PIC method [17], creating the modified problem

$$\frac{\partial \vec{q}}{\partial t} + \vec{u} \cdot \nabla \vec{q} = \mathscr{L}\vec{q} + \theta(\mathscr{G}(\vec{q}) - \vec{q}) \qquad (2.4)$$

where $\mathscr{G}$ samples the field on the grid and interpolates it back (to the particles). The modification adds a decay at rate θ to features on the particles that are not represented on the grid, i.e., are not interpolated from the grid values. This modification does not affect the rate of spatial convergence because the interpolation and approximation schemes are high-order accurate: $\mathscr{G}(q) - q = \mathrm{O}(\Delta x^n)$. However, it does have the potential to affect convergence with respect to the time step. By

[1] While we restrict our attention to smooth problems, we believe that WENO is an important step towards handling non-smooth problems.

scaling θ as $\mathrm{O}(\Delta t^{n-1})$, the effect is no more than $\mathrm{O}(\Delta t^{n})$ and high-order convergence is not affected. In particular, we use the form $\theta = k\Delta t^{3}$ with user-supplied constant parameter k. In practice, we choose k such that the half-life of the decay ($\lambda = \ln(2)/\theta$) is on the order of the time-scale of the interesting dynamics. Qualitatively, this choice keeps the extra dissipation introduced by the regularization from being too strong. We never found this to be too weak to combat noise, although the rate at which noise grows may be problem and resolution dependent.

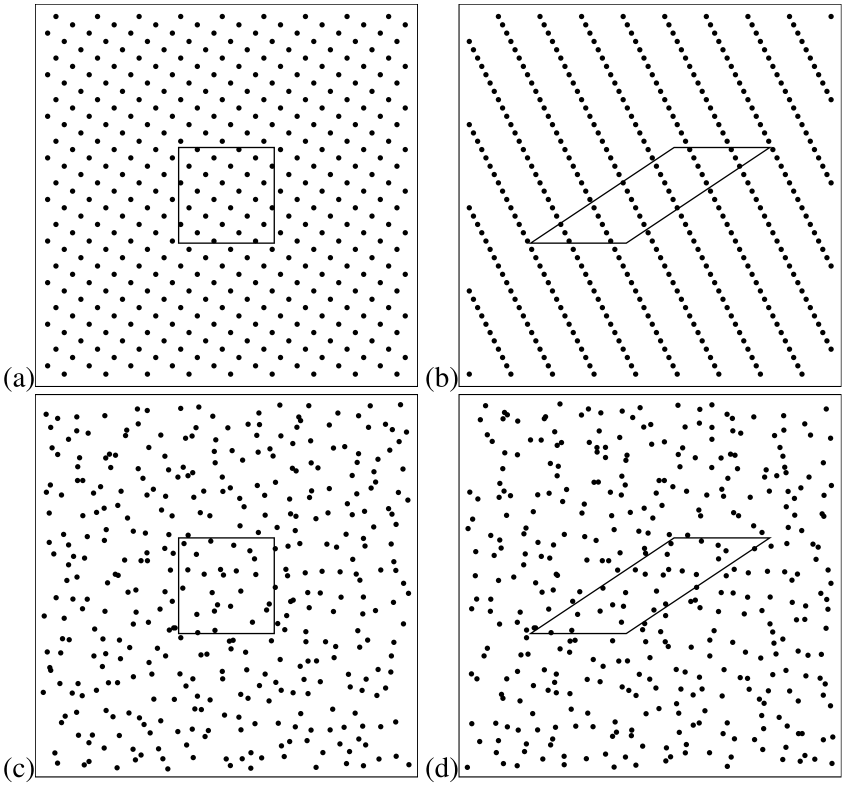


Figure 2.1: A comparison of structured and random sampling. In (a), a structured grid sampling is used, which when deformed by a uniform shear flow, results in (b), exhibiting severe degradation of sampling quality. In (c), the structured grid samples are randomly jittered, eliminating all directional bias. Under the same shear flow, (d) illustrates how the random distribution remains reasonable and unbiased.

Chapter 3

Implementation

We implemented a fourth-order accurate HOPIC for a variety of problems. The implementation was programmed in C++ and parallelized using OpenMP [12]. WLS computations were done using the *dgels* routine from LAPACK [1], and Fourier transforms for the pseudo-spectral calculations were done with FFTW [15].

All timed experiments were run on a 4-core Intel i5 machine with 3 GB of RAM, running openSUSE 11.2. Our implemenation was compiled with the GNU C++ compiler version 4.4.1 and optimization flag '-O2'.

Each time the particles were moved, we computed an acceleration structure that records which particles are in each cell. During MLS approximation at a vertex, the kernel support is exactly the the volume of the cells surrounding that vertex, so the particles needed were available from the acceleration structure. Likewise, when interpolating from the grid to the particles, we iterated over the grid cells, and interpolated to all the particles in a particular cell at once. This also allowed precomputation and reuse of intermediate WENO calculations.

Presumably, better cache coherence could be achieved by sorting the particles in memory according to which grid cell they were in. However, we did not observe this in practice. We started with particles in this sorted order, but did not resort during the simulation, and did not observe any slowdown as the particles become mixed.

We also present a calculation of the 1D WENO interpolant that is more computationally efficient than the expressions from Macdonald and Ruuth [21]. Our formu-

```
function f = weno4(f1,f2,f3,f4,x)
  [w1,p1] = wenoParabola(f1,f2,f3,   x);
  [w2,p2] = wenoParabola(f4,f3,f2, 1-x);
  f = (w1*p1 + w2*p2)/(w1+w2);

function [w,g] = wenoParabola(f1,f2,f3,x)
  d  = (f3-f1)*0.5;          % 1st derivative at 0
  dd = (f1-2*f2+f3);         % 2nd derivative
  S = d*(d+dd) + 4/3*dd^2;   % smoothness over x=[0,1]
  w = (2-x)/(1e-6 + S)^2;    % relative weight
  g = f2 + x*(d + 0.5*x*dd); % value of parabola at x
```

Listing 3.1: Efficient calculation of WENO interpolant.

lation uses 44 FLOPs (21 adds and 23 multiplies), compared to 73 FLOPs (28 adds and 45 multiplies) from the naive expression. The key is to rewrite the smoothness measures in terms of the polynomial derivatives, and then reuse these terms in the other expressions. For 4 points located at $-1,0,1,2$ with values f_1,f_2,f_3,f_4 and $x\in[0,1]$, we calculate the WENO interpolant at x by the algorithm in Listing 3.1.

Chapter 4

Experiments

We performed a variety of experiments using HOPIC on different problems. The primary goal of our experimentation was to confirm fourth-order convergence of HOPIC. We also compared HOPIC to other, Eulerian, high-order methods. The experiments and their results are presented here, in order of increasing complexity.

4.1 Advection-Diffusion

Our first demonstration of HOPIC is the advection-diffusion equation for a scalar quantity q and given velocity field $\vec{u}$

$$\frac{\partial q}{\partial t} + \vec{u} \cdot \nabla q = D \nabla \cdot \nabla q \tag{4.1}$$

where D is the constant diffusion coefficient.

With $D = 0$ and no regularization, HOPIC merely transports the initial particle values around the domain and the only error is in the trajectory of the particles through the velocity field. With regularized HOPIC, grid-representable solutions perform nearly this well, and sub-grid features are smoothed depending on θ. Relative to a Lagrangian approach, even high-order accurate Eulerian methods suffer from severe numerical diffusion in this case.

We measured convergence against the 2D analytic solution of advecting the Fourier mode $q_0(x,y) = \sin(x+y)$ in a constant velocity field $\vec{u} = (\sqrt{2}, \sqrt{5})$ not aligned with the grid. We let $D = 10^{-3}$ and simulated time $t = 0$ to $t = 1$. Consistent with the convergence requirements specified in section 2.4, $\Delta t = 1/N$, $\Delta x = 2\pi/N$ and $\theta = 1/N^3$. The results show the expected fourth-order convergence (Figure 4.1).

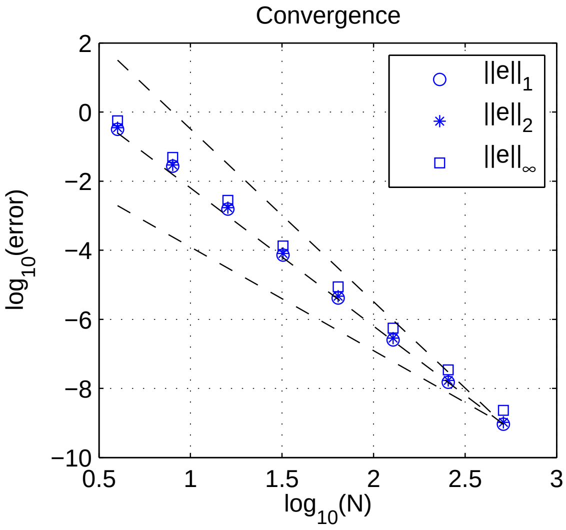


Figure 4.1: Convergence of HOPIC to the analytic solution of advecting and diffusing $\sin(x+y)$ in a uniform velocity field. Dashed lines show third, fourth, and fifth-order convergence.

In a spatially varying velocity field, we performed a numerical convergence study starting with a smooth field $1-\cos(x)$ advected and diffused within a vortex. The velocity field is given as the curl of the stream function $\psi = (1-\cos(x))(1-\cos(y))$. Simulation parameters are the same as the previous example, and error was measured against an $N = 512$ simulation. Convergence results are shown in figure 4.2, and the transported field is shown in 4.3.

4.2 Shallow Water

Our second demonstration of HOPIC is on the shallow water equations

$$\frac{\mathrm{D}}{\mathrm{D}t}\begin{bmatrix}\vec{u}\\ h\end{bmatrix} = -\begin{bmatrix}g\nabla h\\ h(\nabla\cdot\vec{u})\end{bmatrix} \tag{4.2}$$

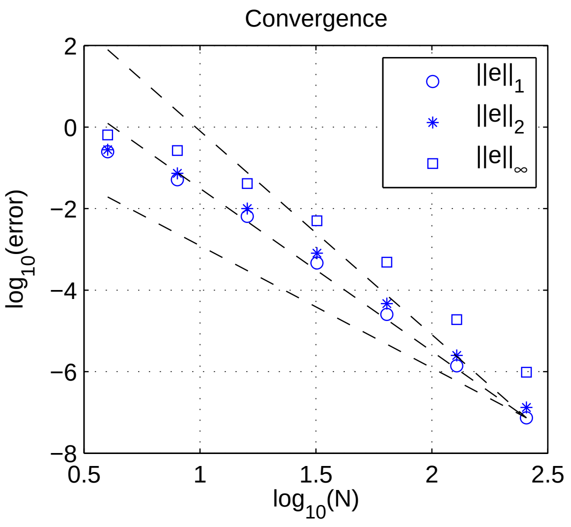


Figure 4.2: Convergence of HOPIC to a high resolution numerical solution of advection-diffusion in a vortex. Dashed lines show third, fourth, and fifth-order convergence.

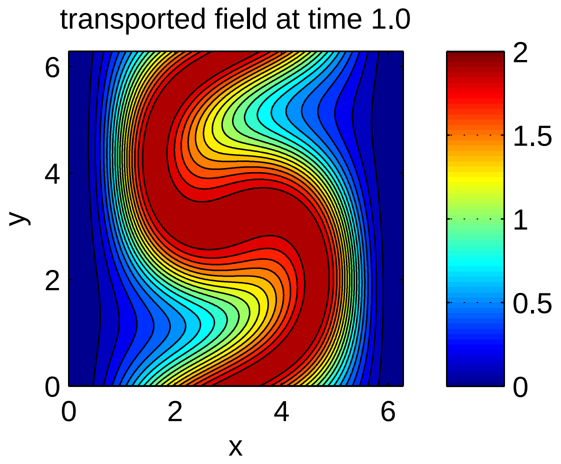


Figure 4.3: The final state at time $t = 1$ of the field $1 - \cos(x)$ following advection and diffusion in a vortex.

where h is the height of the water's surface above a flat bottom, and $\mathrm{D}/\mathrm{D}t$ is the material derivative (i.e., $\mathrm{D}\vec{q}/\mathrm{D}t = \partial\vec{q}/\partial t + \vec{u}\cdot\nabla\vec{q}$). We evaluated the partial derivatives on a collocated grid using fourth-order central finite differences.

We measured convergence of HOPIC on a test problem with initial conditions $u, v = 0$, $h = 1 + (1-\cos(x))(1-\cos(y))$ on the periodic domain $[0, 2\pi)^2$. Simulation parameters were $\Delta t = 0.5/N$, $\Delta x = 2\pi/N$ and $\theta = 1/N^3$. Error was measured relative to a $N = 512$ simulation and results are shown in figure 4.4. We observed fourth-order convergence from $N = 16$ in the L_1 and L_2 norm. In the L_∞ norm, convergence did not appear to be fourth order until $N \approx 128$.

Although the shallow water equations are capable of generating and propogating shocks, our present method is not designed to handle them. In practice, MLS produces large and noisy oscillations around shocks, and correct shock speed according to the Rankine-Hugoniot condition is not enforced as it is in a Finite Volume scheme, for example. As such, we do not consider cases in which a shock develops, and our example is chosen to avoid shocks.

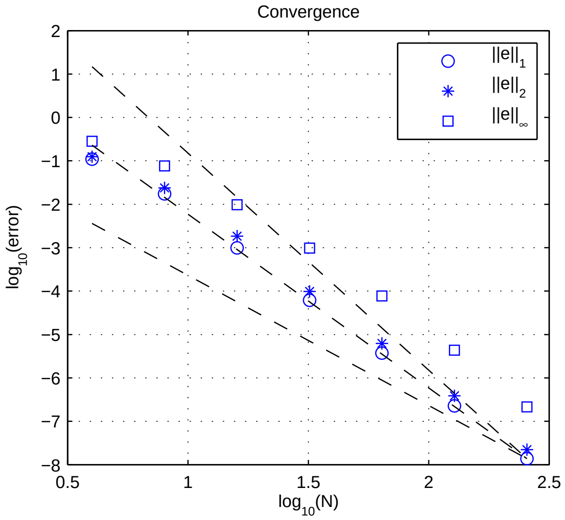


Figure 4.4: Convergence of HOPIC to a high resolution numerical result on the shallow water equations. Dashed lines show third, fourth, and fifth-order convergence.

4.3 Vorticity

Our last demonstration of HOPIC is incompressible fluid flow, in vorticity $\vec{\omega}$ formulation

$$\frac{\mathrm{D}\vec{\omega}}{\mathrm{D}t} = \vec{\omega} \cdot \nabla \vec{u} + \nu \nabla \cdot \nabla \vec{\omega} \tag{4.3}$$

$$\vec{\omega} = \nabla \times \vec{u} \tag{4.4}$$

with viscosity coefficient ν. In 2D, equation 4.3 simplifies to scalar vorticity and no vortex-stretching term

$$\frac{\mathrm{D}\omega}{\mathrm{D}t} = \nu \nabla \cdot \nabla \omega \tag{4.5}$$

For this problem, we computed the right-hand-side derivatives on the grid using a pseudo-spectral method with 3/2-rule dealiasing [24]. The 3D problem has the property that $\vec{\omega}$ should be divergence-free at all times, so every time $\vec{\omega}$ was approximated on the grid, it was projected to be divergence-free by the pseudo-spectral method. The regularization term used this projected vorticity, keeping the particles from accumulating a curl-free component.

4.3.1 2D Test Problems

In 2D, we measured convergence against the analytic solution

$$\omega = 2e^{-2\nu t} \sin(x) \sin(y) \tag{4.6}$$

of 2D Taylor Green flow due to [10]. This represents 4 alternately rotating vortices that decay exponentially over time, in the periodic domain $[0, 2\pi)^2$. Simulation parameters were $\Delta t = 1/N$, $\Delta x = 2\pi/N$, $\theta = 1/N^3$, and $\nu = 0.001$. Results are shown in figure 4.5, and display fourth-order convergence over a wide range of resolutions.

Our next test problem displays much more dynamic behaviour. The initial

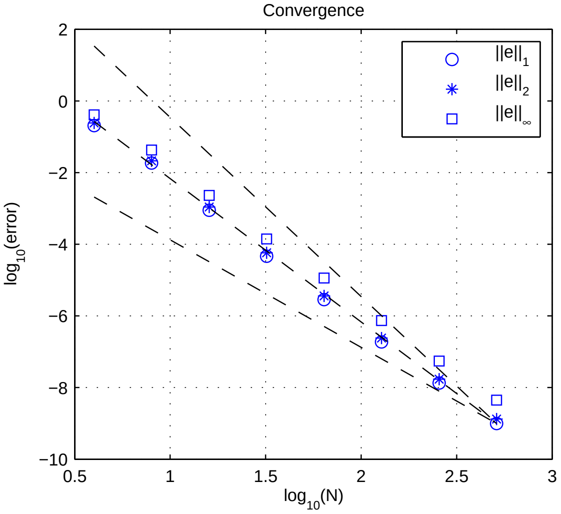


Figure 4.5: Convergence of HOPIC to the analytic Taylor-Green flow in 2D. Dashed lines show third, fourth, and fifth-order convergence.

vorticity

$$\omega(\vec{x}) = \mathrm{k}((\vec{x}-[0.7,0.2])/0.2) - \mathrm{k}((\vec{x}-[0.3,0.2])/0.2) \tag{4.7}$$

$$\mathrm{k}(\vec{r}) = \begin{cases} \exp(1-1/(1-\|\vec{r}\|^2)) & \text{if } \|\vec{r}\| < 1 \\ 0 & \text{otherwise} \end{cases} \tag{4.8}$$

describes a vortex dipole starting at one end of the simulation domain. In the true solution, the dipole self-advects along a straight line and the individual vortices wobble and stretch slightly (figure 4.6). We used inviscid ($\nu = 0$) conditions for this problem, so both enstrophy and kinetic energy should be conserved, and the vortices stay roughly the same size.

On this test problem, we compared HOPIC to an Eulerian scheme using fifth-order upwinding WENO [19] to compute the advection term. We kept the Eulerian scheme as close as possible to our HOPIC scheme by using the same pseudo-spectral approach for the right-hand-side derivatives and RK4 time-discretization.

We simulated from time $t = 0$ to $t = 50$, in which time the dipole crossed half the $[0,1)^2$ domain. All simulations used $N \times N$ grids, with time steps restricted to

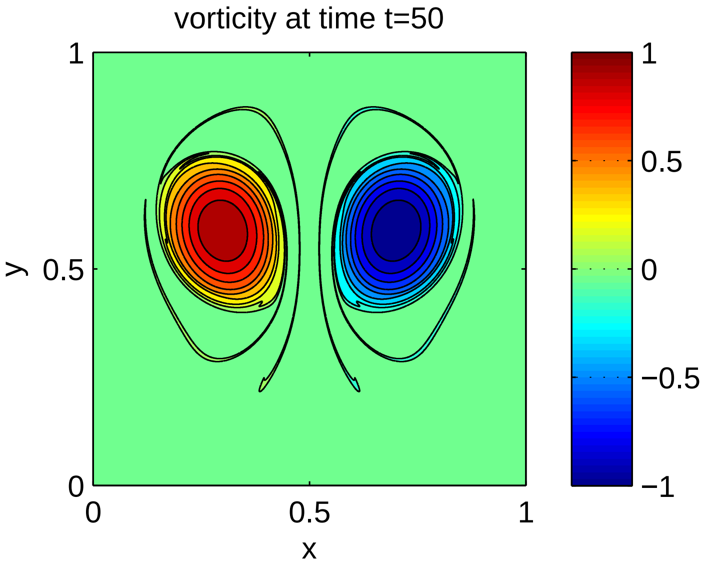


Figure 4.6: Vorticity distribution in the 2D dipole advection test problem. The initially radially-symmetric vortices wobble and distort as they move.

CFL number 1.0, and $\theta = (10/N)^3$. Drift in kinetic energy and enstrophy are used to detect numerical dissipation, with the results shown in figure 4.7.

On equal size grids, HOPIC achieves much higher accuracy than WENO, though it obviously has additional overhead due to the particles. We attempted to quantify the performance differences using our implementation. HOPIC is more efficient than the Eulerian scheme for achieving higher accuracies, despite being much more computationally expensive per step.

The work done by HOPIC due to the use of particles is $O(m)$ to transfer between m grid cells and the particles. However, the work done on the grid by both methods is the $O(m \log m)$ time of a Fast Fourier Transform (FFT). Because of this, we expect Eulerian schemes will slow down even more relative to HOPIC on problems requiring higher resolution grids. If the right-hand-side of a problem were to require more than the $O(m \log m)$ time of a FFT, we would also expect the particle overhead of HOPIC to be better amortized, increasing its advantage.

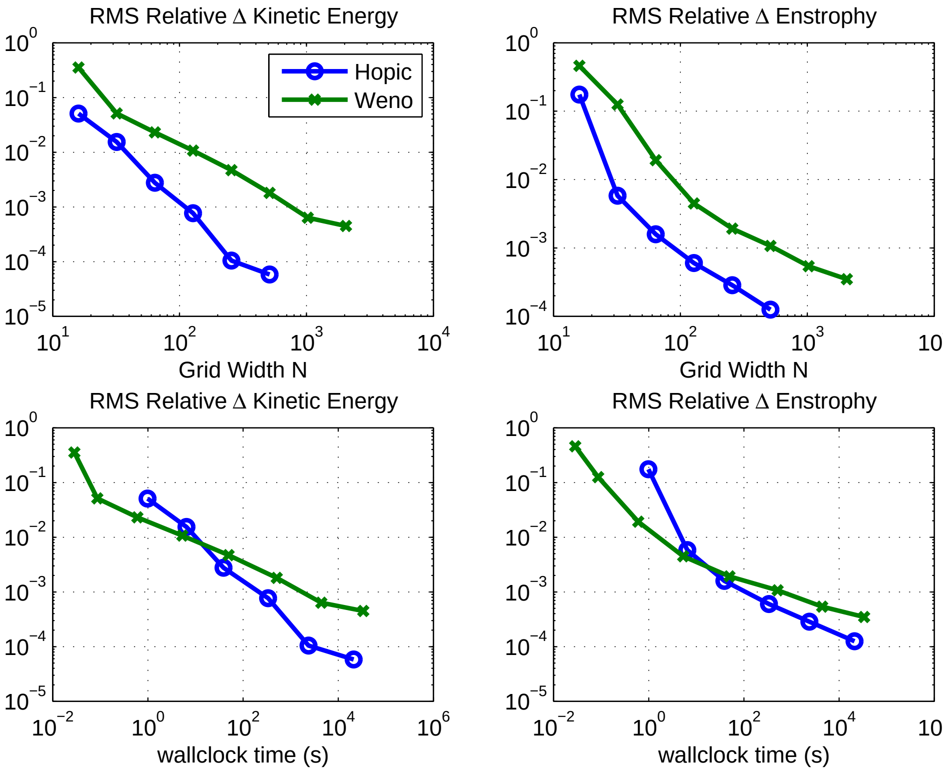


Figure 4.7: Performance comparison of HOPIC and Eulerian WENO scheme. Top row shows accuracy plotted against the size of the $N \times N$ grid. Bottom row shows accuracy plotted against total simulation runtime. Accuracy is presented in terms of drift in the conserved quantities: kinetic energy (left) and enstrophy (right).

4.3.2 3D Test Problems

Our first test problem in 3D is the Taylor-Green Vortex (TGV) [28]. The TGV decays into isotropic turbulence from an initially simple vorticity field

$$\vec{\omega} = \begin{pmatrix} \cos(x)\sin(y)\cos(z) \\ \sin(x)\cos(y)\cos(z) \\ 2\sin(x)\sin(y)\sin(z) \end{pmatrix} \tag{4.9}$$

We measured convergence of HOPIC on the TGV simulated to time $t = 1$. At this time, the flow was distorted (figure 4.9) but had not yet decayed into turbulence. Parameters for all runs were $\Delta x = 2\pi/N$, $\Delta t = 1/N$, $\theta = 43300/N^3$, and $\nu = 1/1500$. Error was measured against the result from a $N = 150$ simulation. The results, shown in figure 4.8, again display fourth-order convergence.

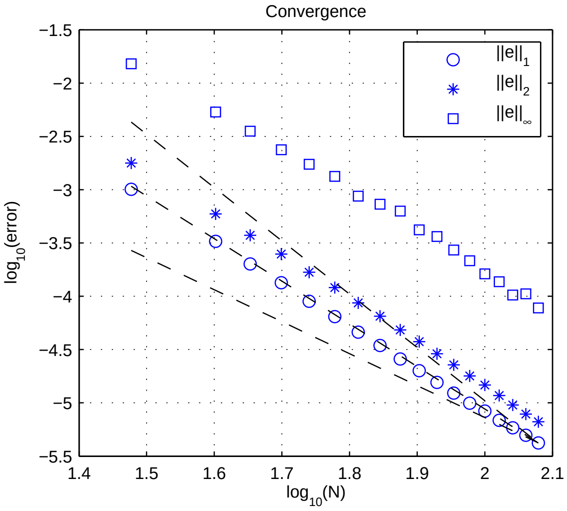


Figure 4.8: Convergence of HOPIC on the 3D Taylor Green flow. Error is measured at time $t = 1$, against a $150 \times 150 \times 150$ simulation. Dashed lines show third, fourth, and fifth-order convergence.

We also verified our results against a 3D pseudo-spectral code and the results from the Large-Eddy Simulation (LES) by Fauconnier et al.[14]. The pseudo-spectral code and LES simulation were indistinguishable to time $t = 7$, and we consider them to be following the true solution. HOPIC on coarse grids loses energy faster than the true solution, but follows the same qualitative behaviour (figure 4.10).

Our second test problem in 3D is a vortex ring. The initial vorticity is contained inside the torus constructed by tracing a tube of radius $\gamma = \pi/4$ about a circle of radius $\pi/3$. The initial vorticity was tangent to the nearest point on the circle and had magnitude $(\gamma^2 - r^2)^3/2\gamma^2$ where r was the distance to the circle. However, this construction for $\vec{\omega}$ is not divergence free, so we projected it to be divergence free using the same pseudo-spectral approach used to evaluate the coupling terms.

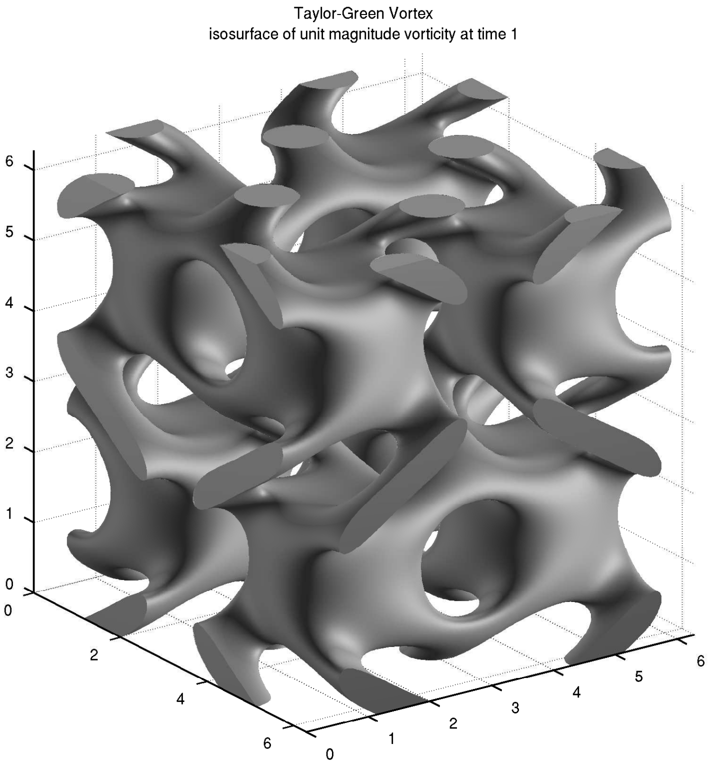


Figure 4.9: The isosurface of unit magnitude vorticity, of the 3D Taylor-Green vortex, at time $t = 1$.

We compared our results of a simulated vortex ring to the theory and experiments by Maxworthy [22] and Widnall and Sullivan [30]. Maxworthy observed that at low Reynolds number, vortex rings are stable but slow over time. With increasing Reynolds number ($600 \lesssim \mathrm{Re} \lesssim 1000$), the vortex rings become unstable and collapse in increasingly turbulent manners. At high Reynolds numbers ($\gtrsim 1000$), the ring quickly collapses into a cloud of vorticity from which another stable vortex ring is ejected. Widnall and Sullivan explain the collapse as due to a wave-like instability around the ring, with a higher wavenumber at higher Reynolds number.

Our simulations captured all of these qualitative behaviours. At $\nu = 1/500 \approx 1/\mathrm{Re}$, our simulated vortex ring was stable. At $\nu = 1/5000$, our vortex ring de-

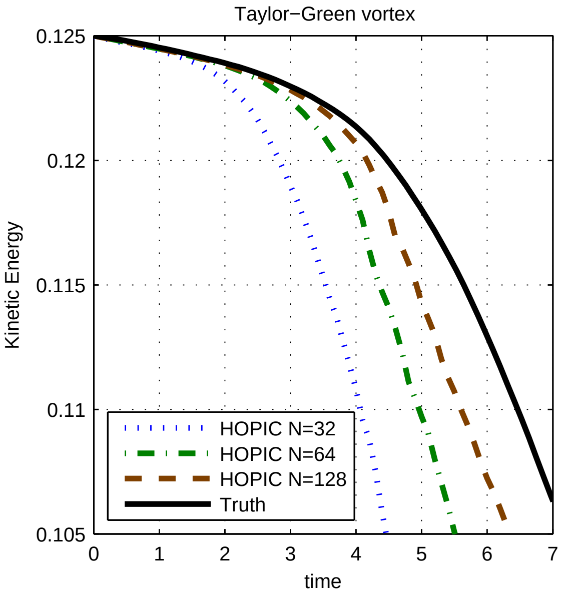


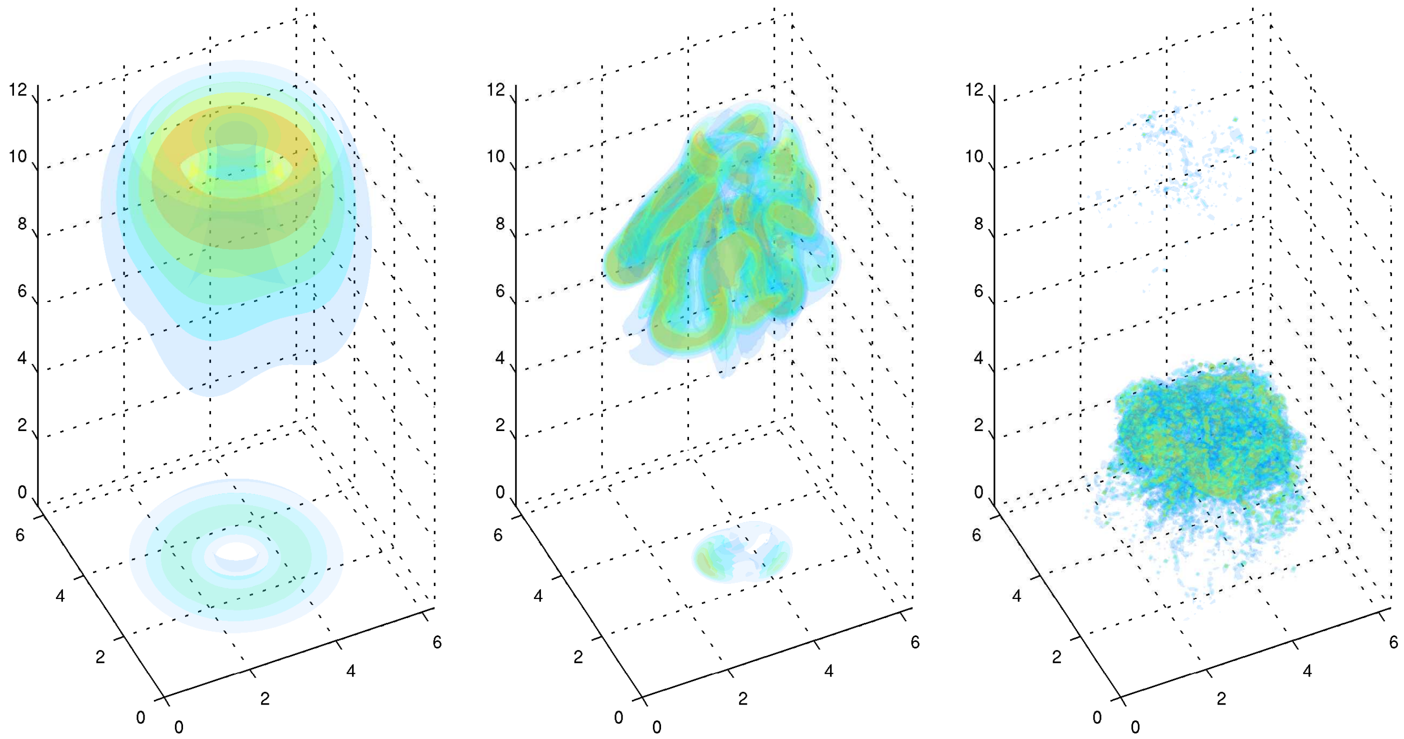
Figure 4.10: Kinetic energy of the decaying 3D Taylor Green Vortex. ‘Truth’ is from a high resolution large eddy simulation [14]

.

veloped a wave-like instability with four periods around the ring and subsequently collapsed at around time 750. At $\nu = 1/50000$, the ring developed an instability with seven periods and collapsed at approximately time 450. Production of a second vortex ring after collapse was observed at $\nu = 1/50000$, but was sensitive to the other simulation parameters. Differences in the initial conditions and the ambiguities in the definition of Reynolds number make a more detailed comparison to the literature impossible.

We compared HOPIC to two Eulerian methods in 3D. First, the same upwind WENO scheme as in 2D and, second, a fully pseudo-spectral scheme. However, we encountered problems with both of these methods. The WENO method was unstable when simulating the Taylor-Green vortex beyond the beginning of the turbulence cascade ($t \approx 2$), with unbounded growth in kinetic energy. On the vortex-ring test problem, the pseudo-spectral code produced qualitatively wrong results at high Reynolds number, $\nu = 1/50,000$. This appeared as excess isotropic noise growing from the ring to fill the domain, presumably due to the grid not being refined enough for true direct numerical simulation. HOPIC was robust under all tests we tried.

In a simulation with m grid cells, the interpolation and approximation steps are all $O(m)$. The physics step is $O(m \ln m)$ using the FFT, and could be worse with general linear solvers. An $m = 75^3$ 3D vorticity simulation is illustrative of typical timings. The average step took 70 seconds to compute, with 50% of the time spent in MLS approximation, 20% in WENO interpolation, 14% computing physics on the grid, 6% precomputing acceleration structures, and the remaining time in IO and other auxiliary functions.



(a) $\nu = 1/500, t = 600$

(b) $\nu = 1/5000, t = 800$

(c) $\nu = 1/50000, t = 500$

Figure 4.11: Vortex rings of varying vorticity. Plots show volume renderings of the magnitude of vorticity. At high viscosity, the ring is stable. Decreasing viscosity produces unstable vortex rings which collapse more quickly and violently.

Chapter 5

Conclusions

We present the HOPIC method that extends PIC techniques to high-order accuracy for general transport problems. The core idea is to consider the particles as a sampling of the underlying continuous field. MLS approximation and WENO interpolation provide a high-order means to transfer information between the particles and grid. Coupled with any high-order accurate scheme to compute differential terms on the grid, the result is global high-order spatial accuracy. Temporal accuracy is supplied by a standard explicit time-integration method for ODEs. Furthermore, a regularization that decays particle values towards the grid-interpolated values removes noise without affecting convergence.

We implemented a fourth-order version of HOPIC and demonstrated it on a variety of problems, in both 2D and 3D. The results showed the designed fourth-order convergence and the low numerical dissipation and dispersion expected from its similarity to FLIP. Quantifying and characterizing the nature of the error in more detail is left for future work. Although it comes with no guarantees about conservation or stability, we found that HOPIC robustly handled all of our test problems. It produced qualitatively reasonable results, even when the Eulerian schemes were unstable. Compared to high-order accurate Eulerian schemes, HOPIC produced superior results on the same size grid, and for high accuracies, HOPIC also had lower compute time.

This initial investigation into HOPIC produced promising results but leaves many avenues for future work. Handling more general classes of problems, especially boundary conditions and constraints (such as incompressibility in a velocity-pressure formulation of flow) is clearly important and we hope to investigate those. Similarly, extending the time integration to include implicit methods could be critical for some applications. We also did not address shocks, free surfaces, or other discontinuities; this would require an extension of MLS (which is unable to handle discontinuities). Besides handling more general problems, there is also room for performance improvements both in terms of computational speed and constant-factor gains in accuracy. For example, replacing MLS, WENO, and the grid-solve with three methods that agree on the representation of the field could provide a practical gain in accuracy.

Bibliography

[1] E. Anderson, Z. Bai, C. Bischof, S. Blackford, J. Demmel, J. Dongarra, J. Du Croz, A. Greenbaum, S. Hammarling, A. McKenney, and D. Sorensen. *LAPACK Users' Guide*. Society for Industrial and Applied Mathematics, Philadelphia, PA, third edition, 1999. ISBN 0-89871-447-8.

[2] I. Babuska, U. Banerjee, and J. E. Osborn. Generalized Finite Element Methods: Main Ideas, Results, and Perspective. *International Journal of Computational Methods*, 1(1):67–103, 2004. doi:10.1142/S0219876204000083.

[3] S. G. Bardenhagen and E. M. Kober. The generalized interpolation material point method. *Computer Modeling in Engineering and Sciences*, 5(6): 477–495, 2004.

[4] J. T. Beale and A. Majda. Vortex Methods. II: Higher Order Accuracy in Two and Three Dimensions. *Mathematics of Computation*, 39(159):29–52, 1982. ISSN 00255718.

[5] J. Brackbill. The ringing instability in particle-in-cell calculations of low-speed flow. *Journal of Computational Physics*, 75(2):469–492, 1988.

[6] J. Brackbill and H. Ruppel. FLIP: A method for adaptively zoned, particle-in-cell calculations of fluid flows in two dimensions. *Journal of Computational Physics*, 65(2):314–343, August 1986. doi:10.1016/0021-9991(86)90211-1.

[7] J. U. Brackbill. Particle methods. *International Journal for Numerical Methods in Fluids*, 47(8-9):693–705, March 2005. ISSN 0271-2091. doi:10.1002/fld.912.

[8] J. U. Brackbill, D. B. Kothe, and H. M. Ruppel. FLIP: A low-dissipation, particle-in-cell method for fluid flow. *Computer Physics Communications*, 48(1):25–38, January 1988. doi:10.1016/0010-4655(88)90020-3.

[9] D. Burgess, D. Sulsky, and J. Brackbill. Mass matrix formulation of the FLIP particle-in-cell method. *Journal of Computational Physics*, 103(1): 1–15, November 1992. ISSN 00219991. doi:10.1016/0021-9991(92)90323-Q.

[10] A. J. Chorin. Numerical Solution of the Navier-Stokes Equations. *Mathematics of Computation*, 22(104):745–762, 1968.

[11] A. J. Chorin. Numerical study of slightly viscous flow. *Journal of Fluid Mechanics Digital Archive*, 57(04):785–796, 1973. doi:10.1017/S0022112073002016.

[12] L. Dagum and R. Menon. OpenMP: an industry standard API for shared-memory programming. *IEEE Computational Science and Engineering*, 5(1):46–55, 1998.

[13] R. Eymard, T. Gallouet, and R. Herbin. Finite volume methods. *Handbook of numerical analysis*, 7:713–1018, 2000.

[14] D. Fauconnier, C. De Langhe, and E. Dick. Construction of explicit and implicit dynamic finite difference schemes and application to the large-eddy simulation of the Taylor-Green vortex. *Journal of Computational Physics*, 228(21):8053–8084, 2009. doi:10.1016/j.jcp.2009.07.028.

[15] M. Frigo and S. G. Johnson. The design and implementation of FFTW3. *Proceedings of the IEEE*, 93(2):216–231, 2005.

[16] C. F. O. Gooch. Quasi-ENO Schemes for Unstructured Meshes Based on Unlimited Data-Dependent Least-Squares Reconstruction. *Journal of Computational Physics*, 133(1):6–17, May 1997. ISSN 00219991. doi:10.1006/jcph.1996.5584.

[17] F. H. Harlow. The particle-in-cell method for numerical solution of problems in fluid dynamics. *Methods in Computational Physics*, pages 319–343, 1964.

[18] F. H. Harlow. PIC and its progeny. *Computer Physics Communications*, 48 (1):1–10, 1988.

[19] G. Jiang. A high-order WENO finite difference scheme for the equations of ideal magnetohydrodynamics. *Journal of Computational Physics*, 150(2): 561–594, April 1999. ISSN 00219991. doi:10.1006/jcph.1999.6207.

[20] P. Lancaster and K. Salkauskas. Surfaces generated by moving least squares methods. *Mathematics of Computation*, 37(155):141–158, 1981. doi:10.2307/2007507.

[21] C. B. Macdonald and S. J. Ruuth. Level Set Equations on Surfaces via the Closest Point Method. *Journal of Scientific Computing*, 35(2-3):219–240, March 2008. ISSN 0885-7474. doi:10.1007/s10915-008-9196-6.

[22] T. Maxworthy. The structure and stability of vortex rings. *Journal of Fluid Mechanics*, 51(01):15–32, 1972. doi:10.1017/S0022112072001041.

[23] L. Moresi. A Lagrangian integration point finite element method for large deformation modeling of viscoelastic geomaterials. *Journal of Computational Physics*, 184(2):476–497, January 2003. doi:10.1016/S0021-9991(02)00031-1.

[24] R. Peyret. *Spectral Methods for Incompressible Viscous Flow*. Springer, 2002. ISBN 0387952217.

[25] W. E. Schiesser. *The Numerical Method of Lines: Integration of Partial Differential Equations*. Academic Press, 1991. ISBN 0126241309.

[26] M. Steffen, R. M. Kirby, and M. Berzins. Analysis and reduction of quadrature errors in the material point method (MPM). *International Journal for Numerical Methods in Engineering*, 76(6):922–948, November 2008. ISSN 00295981. doi:10.1002/nme.2360.

[27] D. Sulsky. Application of a particle-in-cell method to solid mechanics. *Computer Physics Communications*, 87(1-2):236–252, May 1995. ISSN 00104655. doi:10.1016/0010-4655(94)00170-7.

[28] G. I. Taylor and A. E. Green. Mechanism of the production of small eddies from large ones. *Proceedings of the Royal Society of London. Series A, Mathematical and Physical Sciences*, 158(895):499–521, 1937. ISSN 00804630.

[29] P. C. Wallstedt and J. E. Guilkey. A weighted least squares particle-in-cell method for solid mechanics. *International Journal for Numerical Methods in Engineering*, in press.

[30] S. E. Widnall and J. P. Sullivan. On the stability of vortex rings. *Proceedings of the Royal Society of London. Series A, Mathematical and Physical Sciences*, 332(1590):335–353, 1973.