INTERNATIONAL JOURNAL FOR NUMERICAL METHODS IN ENGINEERING
Int. J. Numer. Meth. Engng 0000; **00**:1–14
Published online in Wiley InterScience (www.interscience.wiley.com). DOI: 10.1002/nme

A high-order accurate particle-in-cell method

Edwards, Essex* and Bridson, Robert

Dept. Computer Science, University of British Columbia

SUMMARY

We propose the use of high-order weighted essentially non-oscillitory interpolation and moving-least-squares approximation schemes alongside high-order time integration 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 Eulerian schemes. Copyright © 0000 John Wiley & Sons, Ltd.

Received . . .

KEY WORDS: particle-in-cell; high-order accuracy; low dissipation; inviscid; incompressible; unstructured; PIC; FLIP; MPM; GIMP

1. INTRODUCTION

The Particle-in-Cell (PIC) method [1] is over half a century old, and while it has a long history of robustly handling transport problems [2], its limitation to first-order accuracy has led to its eclipse by modern Eulerian schemes and others (e.g. [3, 4]). In this paper, we demonstrate how to increase the order of accuracy of PIC, leading to Higher-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 [5], we view these as massless samples of continuous fields, rather than actual "blobs" of material. Similar to the Method-of-Lines [6], we evolve the positions and values of these particles with an arbitrary time integration scheme designed for ODE's, such as a Runge-Kutta method. To evaluate integral or differential terms that couple the values of the samples

*Correspondence to: essex@cs.ubc.ca

through space (typically including the velocity field itself), we first approximate the particle values on an auxiliary background grid using high-order moving least squares, then evaluate the terms on the grid (using whatever is most convenient: finite differences, pseudo-spectral methods, etc.), and finally use a high-order method to interpolate those terms from the grid back to the particles. We also 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 2 we place HOPIC in the context of previous work before delineating the details of the algorithm in section 3 for generic equations. Sections 4, 5 and 6 provide numerical examples of HOPIC applied to a variety of equations in 2D and 3D, demonstrating fourth-order accuracy. The focus in this paper 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.

2. RELATED WORK

Particle-in-cell methods [1] 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 [1] 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. In its earliest incarnation, only the nearest grid point was used, but over time smoother schemes were adopted. These high-order weighting functions improve the results, but do not raise the convergence of the algorithms beyond first order.

The early PIC schemes interpolate the grid values to the particles after each step. This causes significant numerical diffusion, and prevents convergence with respect to the time step independent of grid spacing. FLIP [7, 8] essentially eliminates 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) [9] is a finite element method developed from PIC and FLIP and applied to solid mechanics problems; the Generalized Interpolation Material Point (GIMP) method [10] later improved upon 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 [11].

Within the context of MPM and GIMP, recent investigations [11, 12, 13] have analysed the errors due to spatial operators and time integration, and made various improvements. Wallstedt and Guilkey [14] recently used weighted least squares (WLS) to improve the accuracy of the spatial interpolation in GIMP, and achieved second-order accuracy away from boundaries. Their use of WLS and their model of the particles as point samples of the continuum is similar to HOPIC. Their work is focused on solid mechanics, while we present our work as a more general framework for solving transport problems, with an emphasis on fluids.

DOI: 10.1002/nme

In the context of geophysical simulation, Moresi et al. [15] interpreted MPM as a type of finite element method 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.

In parallel to the development of MPM and GIMP, modern PIC schemes [16] have also been developed from vortex-particle methods [17]. Cottet and Weynans [18] have analyzed the high-order convergence properties of these methods in 1D. In 2D and 3D, the method converges at high-order to a smoothed version of the problem, but high-order convergence to the true solution requires that the particle resolution increases faster than the grid resolution [16], i.e. the number of particles must grow very quickly to see convergence. Regardless of this theoretical requirement, high-order convergence is sometimes observed in practice [19].

A common theme across all of these PIC methods (with the exception of the WLS method [14]) is that the interpolation accuracy depends on how regular the particle distribution is. Since accuracy is quickly lost when the particle distribution deviates from a regular grid, vortex methods use a remeshing scheme that frequently (typically every step) replaces all of the particles with new particles located on a regular grid. This approach also solves the problem with noise that develops in PIC simulations, but adds some numerical diffusion.

In this context, we present several contributions:

- A method which extends the trend of increasing spatial accuracy in GIMP by using MLS and weighted essentially non-oscillitory (WENO) interpolation for arbitrarily high-order spatial interpolation
- High-order convergence to the PDE solution with simple convergence requirements
- Insensitivity to the regularity of the particle distribution, with no loss of convergence with irregular distributions
- A simple particle reseeding strategy, admitted by our insensitivity to irregular distributions
- A regularization to handle the noise that accumulates in PIC schemes, without disrupting high-order convergence.

3. THE GENERIC METHOD

Consider a transport PDE of the form:

$$\frac{\partial \vec{q}}{\partial t} + \vec{u} \cdot \nabla \vec{q} = \mathcal{L}\vec{q} \tag{1}$$

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 $\mathcal{L}$ 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, 5 and 6 provide specific examples: advection-diffusion, shallow water, and vorticity respectively. In the present work we do not admit

DOI: 10.1002/nme

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 3.3. 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. $\vec{q}_i = \vec{q}(\vec{x}_i)$, and initial conditions are set this way. These particles are the complete representation of the problem.

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

$$\frac{d}{dt}\begin{bmatrix}\mathbf{x}\\ \mathbf{q}\end{bmatrix} = \begin{bmatrix}\mathbf{u}\\ \mathcal{L}\vec{q} + \mathcal{R}\end{bmatrix} = \mathbf{f}(\mathbf{x}, \mathbf{q}) \tag{2}$$

where $\mathbf{u} = (\vec{u}_1, \ldots, \vec{u}_m)$ is the vector of all velocities for the particles. This is advanced 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 an approximation from the particles described in section 3.2. On the grid, any technique can be applied to compute $\mathbf{f}$ at the grid vertices (e.g. finite differences); this result is then interpolated back to the particles, as discussed in section 3.4. Finally, a regularization term $\mathcal{R}$ is included in the evolution to prevent sub-grid noise (section 3.5).

3.1. Convergence Requirements

The interpolation steps, evaluation of $\mathbf{f}$, and time integration must all be high-order accurate for the entire process to be high-order accurate. However, this alone is not sufficient for high-order convergence.

Consider any method, such as the one above, in which $d\mathbf{r}/dt = \mathbf{g}(\mathbf{r})$ is solved using an $O(\Delta x^p)$ approximation to $\mathbf{g}$ and an $O(\Delta t^p)$ time integration scheme. Letting the timestep Δt go to zero, with fixed grid resolution $\Delta x = O(1)$, the result converges to the discrete-space-continuous-time solution which is $O(\Delta x^p) = O(1)$ different from the true solution. Likewise, converging only with respect to Δx leaves an $O(\Delta t^p) = O(1)$ error due to discrete time integration. Thus, for the method to converge to the true solution at high-order, a necessary condition is that Δx and Δt converge together, subject to $\Delta x \propto \Delta t$. In practice, we use a CFL condition to enforce this, since it is an intuitive way to set the ratio. However, the dependence on $\|\mathbf{u}\|$ is not required by this argument.

The time integration scheme also puts certain requirements on $\mathbf{f}$. We use a fourth-order Runge Kutta method (RK4), which requires $\mathbf{f}$ to be several times differentiable. Subject to the CFL-like condition above, this smoothness condition does not appear to apply to the numerical approximation to $\mathbf{f}$, but only to the true $\mathbf{f}$ that it converges to. Thus, we restrict our attention to smooth problems.

3.2. Particle-to-Grid Approximation

Existing PIC methods use several approaches to approximate the particles with a grid. One common approach is to conserve extensive properties of the particles, such as total mass or momentum, when transferred to the grid [1, 7, 8]. Another approach [20] 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

DOI: 10.1002/nme

interpolant, and could be of high order, but it comes at the cost of a global and potentially nonlinear solve.

In this paper, we use moving least squares (MLS) [21] to approximate the field sampled by the particles and evaluate it at the grid points. From scattered particles with positions x_i and values $q_i = q(x_i)$, MLS defines a field which approximates the function $q(x)$. It depends on two parameters: a weighting function $w(r)$ and the basis P of functions to approximate with. The space of k^{th} order polynomials is a standard choice. The MLS approximant at a point x is $p(x)$ where $p \in \text{span}\{P\}$ minimizes the error term

$$E_{\text{error}} = \sum_{i=0}^{n} [w(x - x_i)(p(x) - q_i)]^2 \tag{3}$$

When a concrete choice of basis and weight function are determined, this results in an overdetermined linear system $A(x)c = b$ where $A(x)_{i,j} = P_j(x_i)w(x - x_i)$, $b_i = q_i w(x - x_i)$, and c is the coordinate of p with respect to the basis P. For weight function with compact support, this linear system is small and easily solved in a least squares sense. The resulting MLS approximant is of arbitrary order of accuracy and degree of smoothness depending on the choice of basis, the weighting function, and the smoothness of the field sampled by the particles.

The behaviour of the weight function at $r = 0$ determines whether MLS interpolates the data points, or smoothly approximates them. Since we interpret the grid vertices as point samples of the continuum, analogous to the particles, interpolating MLS would seem to be a natural choice. However, we use a simpler approximating MLS since interpolating would be very sensitive to any noise that the particles may accumulate over time (see section 3.5).

The particular choice of weight function has little effect on our results. Because of its simplicity, we use the following C^3 spline in 1D and tensor products of the same in higher dimensions.

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

Only the particles in cells adjacent to a grid vertex are in the support of that vertex's MLS kernel, simplifying the implementation. In our experiments, results were essentially the same with other kernels, including larger and spherically-symmetric kernels. Some performance benefit might be attainable by increasing the size of the kernel while decreasing the particle number density, but we chose the simplicity of this compact kernel.

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

In the presence of shocks, kinks, or other sharp features, MLS can lead to severe overshoots and thence instability; a nonlinear approach to limit overshoot would be required, such as in Quasi-ENO MLS [22]. However, in this paper we restrict our attention to problems with smooth solutions.

DOI: 10.1002/nme

3.3. 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 moving least-squares 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. 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 still be filled and extreme oversampling is computationally wasteful.

The minimum number of particles in the neighborhood of each grid point is maintained by adding particles to any cell that approaches having too few. However, reseeding can only be done between time-steps, not before each evaluation of f by RK4 (as that would change the length of the state-vector being accumulated). So, to maintain this minimum density across multiple sub-steps, 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 within any cell which has too few ($n < 6$ in 3D). New particle values $\vec{q}_i$ are interpolated from the grid. With incompressible 3D flow, typical turnover in our examples is 0.5% of particles per step.

Because the interpolation procedure is fourth-order accurate, the new particles values are also fourth order accurate. However, if the number of timesteps taken is proportional to the grid resolution, and a constant fraction of the particles are updated on each reseeding, then these fourth-order perturbations can add up to a third-order error over the course of the simulation. In practice, we did not observe this problem with divergence-free flows, where the reseeding rate is so low that this is not usually the largest error term. With divergent flows, such as shallow water, reseeding is much more significant, and a more careful reseeding strategy might be necessary here. One simple solution might be to increase the order of approximation and interpolation when reseeding.

We emphasize the use of random sampling to seed particles, unlike other PIC schemes which employ structured distributions. While for a single time step, structured distributions offer slightly higher quality reconstruction, 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. Figure 1 illustrates this phenomenon.

3.4. Grid-to-Particle Interpolation

Using a regular rectangular auxiliary grid admits a wide variety of existing interpolation schemes. The Weighted Essentially Non-Oscillitory (WENO) schemes are particularly attractive for their simplicity and robust behaviour around sharp features. While we restrict our attention to smooth problems, we believe that WENO is an important step towards handling non-smooth problems.

WENO works by building two lower-order interpolants and computing a convex combination of them. The combination weights are determined from smoothness measures of the low-order

DOI: 10.1002/nme

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

function [w,g] = weno_parabola(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 1: Matlab code for efficient calculation of WENO interpolant.

interpolants such that the result is high-order accurate in smooth regions, and has minimal overshoot in non-smooth regions.

In our examples, we use the fourth-order WENO scheme described, for example, by Macdonald and Ruuth [23]. This uses a four-point stencil in 1D, and in higher dimensions it is computed one dimension at a time, giving a 16-point stencil in 2D, and 64 points in 3D. Because of nonlinearities in WENO, 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.

We present code for calculating the 1D interpolant in listing 1. For four points located at $\{-1,0,1,2\}$ with values $\{f_1,f_2,f_3,f_4\}$ and $x \in [0,1]$, *weno4* calculates the interpolant at x. This function uses roughly half as many floating-point operations as a direct implementation of the expressions of Macdonald and Ruuth [23] (44 FLOPS versus approximately 73 FLOPS).

3.5. 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 $\mathcal{L}$ 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 [24]; 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 (1) inspired by the original noise-free PIC method [1], creating the modified problem

$$\frac{\partial \vec{q}}{\partial t} + \vec{u} \cdot \nabla \vec{q} = \mathcal{L}\vec{q} + \theta(\mathcal{G}(\vec{q}) - \vec{q}) \tag{4}$$

where $\mathcal{G}$ applies MLS and WENO to sample the field on the grid and interpolate 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: $\mathcal{G}(q) - q = O(\Delta x^n)$. However, it does have the potential to affect convergence with respect to the time step. By scaling θ as $O(\Delta t^{n-1})$, the effect is no more than $O(\Delta t^n)$ and high-order convergence is not affected. In particular, we use the form $\theta = k\Delta t^3$ with user-supplied constant parameter k.

DOI: 10.1002/nme

In practice, we choose a half-life λ for the decay which is approximately the same as the time-scale of the interesting dynamics. This gives a rate $\theta = \ln(2)/\lambda$ which we use in the coarsest simulation of a convergence study. Higher resolution runs consequently have even weaker regularization. We never found this to be too weak to combat noise, although the rate at which noise grows may be problem and resolution dependent.

This regularization contributes relatively minor numerical diffusion because the regularization rate is chosen to be small, and the spatial term is a high-order approximation to zero. The maximum numerical diffusion occurs when the regularization rate is very large, effectively forcing the particles to agree with the grid at all times. In this case, the regularization makes HOPIC behave like a high-order Eulerian scheme. This is demonstrated in figure 2 by advecting a smooth pulse ($f = \tanh(3\sin(2\pi(x - 0.25)))$) across the periodic unit interval ten times. The simulations use a grid of twenty samples and three particles per cell.

4. ADVECTION-DIFFUSION

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

$$\frac{\partial q}{\partial t} + \vec{u} \cdot \nabla q = D \nabla \cdot \nabla q \tag{5}$$

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

We measured convergence against the analytic solution of advecting the Fourier mode $q_0(x, y) = \sin(x + y)$ in an arbitrary constant velocity field $\vec{u} = (\sqrt{2}, \sqrt{5})$. We let $D = 10^{-3}$ and simulated time $t = 0$ to $t = 1$. Consistent with the convergence requirements specified in section 3.5, $\Delta t = 1/N$, $\Delta x = 2\pi/N$ and $\theta = 1/N^3$. The results show the expected fourth-order convergence (Figure 3).

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, and the transported field is shown in figure 5.

5. SHALLOW WATER

Our second demonstration of HOPIC is on the shallow water equations

$$\frac{\mathrm{D}}{\mathrm{D}t} \begin{bmatrix} \vec{u} \\ h \end{bmatrix} = - \begin{bmatrix} g \nabla h \\ h(\nabla \cdot \vec{u}) \end{bmatrix} + S \tag{6}$$

DOI: 10.1002/nme

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

We measured convergence using the method of manufactured solutions with solution

$$\begin{aligned} u &= \nabla\times\phi + (t-1/2)\nabla\phi \\ h &= 10+\phi \\ \phi &= \sin^2(x)\sin^2(y+t) \end{aligned}$$

defined on the periodic domain $[0,\pi)^2$. These functions make all of the terms in the shallow-water equations non-zero and have moderately small divergence in the velocity field, but are otherwise arbitrary. Inital conditions and source terms were derived to be consistent with this solution.

Simulation parameters were $\Delta t = N$, $\Delta x = \pi/N$ and $\theta = 1/N^3$. Without reseeding (simply using a high particle density from the start), convergence is shown in figure 6. The results confirm the fourth-order accuracy of the method. When reseeding is done every step, convergence slows down to third-order at higher resolutions due to the error term mentioned in the section 3.3.

Although the shallow water equations are capable of generating and propagating shocks, our present method is not designed to handle them and our example does not include 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.

6. VORTICITY

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

$$\frac{\mathrm{D}\vec{\omega}}{\mathrm{D}t} = \vec{\omega}\cdot\nabla\vec{u} + \nu\nabla\cdot\nabla\vec{\omega} \tag{7}$$

$$\vec{\omega} = \nabla\times\vec{u} \tag{8}$$

with viscosity coefficient ν. In 2D, equation 7 simplifies to scalar vorticity and no vortex-stretching term

$$\frac{\mathrm{D}\omega}{\mathrm{D}t} = \nu\nabla\cdot\nabla\omega \tag{9}$$

For this problem, we computed the right-hand-side derivatives on the grid using a pseudo-spectral method with 3/2-rule dealiasing [25]. 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.

DOI: 10.1002/nme

6.1. 2D Test Problems

In 2D, we measured convergence against the 2D Taylor-Green [26] analytic solution

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

This represents four 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 7, and display fourth-order convergence over a wide range of resolutions.

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

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

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

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 8). We used inviscid ($\nu = 0$) conditions for this problem, so both enstrophy and kinetic energy should be conserved, and the vortices should stay roughly the same size.

On this test problem, we compared HOPIC to an Eulerian scheme using fifth-order upwinding WENO [27] 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-discretizations.

We simulated from time $t = 0$ to $t = 50$, in which time the dipole crossed half of the $[0, 1)^2$ domain. All simulations used $N \times N$ grids, with time steps restricted to CFL number 1.0, and $\theta = (10/N)^3$. Drift in kinetic energy and enstrophy were used to detect numerical dissipation, with the results shown in figure 9.

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 with OpenMP-parallelized [28] C++ implementations of both running on a 4-core 3.2GHz Intel i5 machine with 3GB RAM, using the FFTW library [29] for pseudo-spectral grid evaluations. HOPIC is more time-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 (since there are $O(1)$ particles per cell). However, the work done on the grid by both methods is the $O(m \log m)$ time of a FFT. Because of this, we expect that on problems requiring higher resolution grids, Eulerian schemes will slow down even more relative to HOPIC. For a problem requiring more than the $O(m \log m)$ time of a FFT for solving the grid-terms, we expect the particle overhead of HOPIC to be better amortized, increasing its advantage.

DOI: 10.1002/nme

6.2. 3D Test Problems

Our first test problem in 3D is the Taylor-Green Vortex (TGV) [30]. The TGV decays into turbulence from an initially simple vorticity field:

$$\vec{\omega} = \begin{pmatrix} \cos(x)\sin(y)\cos(z) \\ \sin(x)\cos(y)\cos(z) \\ 2\sin(x)\sin(y)\sin(z) \end{pmatrix} \tag{10}$$

We measured convergence of HOPIC on the TGV simulated to time $t = 1$. At this time, the flow was distorted (figure 11) 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 an $N = 150$ simulation, as higher resolution required too much memory. The results, shown in figure 10, again display fourth-order convergence.

We also verified our results against a 3D pseudo-spectral code and the results from the large-eddy simulation by Fauconnier et al. [31]. The pseudo-spectral code and LES simulation were indistinguishable to time 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 12).

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. Finally, we projected this field to be divergence free using the same pseudo-spectral approach used to evaluate the coupling terms.

We compared our results of a simulated vortex ring to the theory and experiments by Maxworthy [32] and Widnall and Sullivan [33]. 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 developed a wave-like instability with four periods around the ring and subsequently collapsed at around time 750. At $\nu = 1/50000$, the ring devloped 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. The $75 \times 75 \times 150$ grid used in these simulations is too coarse be a direct numerical simulation of the small turbulent details during high Re collapse. However, these results are suggestive, and demonstrate the stability, robustness, and low numerical diffusion of the method.

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

DOI: 10.1002/nme

high Reynolds number, $\nu = 1/50,000$. This appeared as 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.

In a simulation with m grid cells, the interpolation and approximation steps are all $O(m)$, while the physics step is $O(m \ln m)$ using the FFT. 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.

7. 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 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 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 and 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 are investigating those. Similarly, extending the time integration to include implicit methods could be critical for some applications. For problems with divergent flow fields, a more careful reseeding strategy is necessary to maintain convergence. We also did not address shocks, free surfaces, or other discontinuities.

ACKNOWLEDGEMENTS

This work has been performed as a part of the first author's Master's thesis, which was supported by the Natural Sciences and Engineering Research Council of Canada.

REFERENCES

DOI: 10.1002/nme

1. Harlow FH. The particle-in-cell method for numerical solution of problems in fluid dynamics. *Methods in Computational Physics* 1964; :319–343.
2. Harlow FH. PIC and its progeny. *Computer Physics Communications* 1988; **48**(1):1–10.
3. Babuska I, Banerjee U, Osborn JE. Generalized Finite Element Methods: Main Ideas, Results, and Perspective. *International Journal of Computational Methods* 2004; **1**(1):67–103, doi:10.1142/S0219876204000083.
4. Eymard R, Gallouet T, Herbin R. Finite volume methods. *Handbook of numerical analysis* 2000; **7**:713–1018.
5. Brackbill JU. Particle methods. *International Journal for Numerical Methods in Fluids* March 2005; **47**(8-9):693–705, doi:10.1002/fld.912.
6. Schiesser WE. *The Numerical Method of Lines: Integration of Partial Differential Equations*. Academic Press, 1991.
7. Brackbill J, Ruppel H. FLIP: A method for adaptively zoned, particle-in-cell calculations of fluid flows in two dimensions. *Journal of Computational Physics* August 1986; **65**(2):314–343, doi:10.1016/0021-9991(86)90211-1.
8. Brackbill JU, Kothe DB, Ruppel HM. Flip: A low-dissipation, particle-in-cell method for fluid flow. *Computer Physics Communications* January 1988; **48**(1):25–38, doi:10.1016/0010-4655(88)90020-3.
9. Sulsky D. Application of a particle-in-cell method to solid mechanics. *Computer Physics Communications* May 1995; **87**(1-2):236–252, doi:10.1016/0010-4655(94)00170-7.
10. Bardenhagen SG, Kober EM. The generalized interpolation material point method. *Computer Modeling in Engineering and Sciences* 2004; **5**(6):477–495.
11. Steffen M, Kirby RM, Berzins M. Analysis and reduction of quadrature errors in the material point method (MPM). *International Journal for Numerical Methods in Engineering* November 2008; **76**(6):922–948, doi: 10.1002/nme.2360.
12. Wallstedt P, Guilkey J. Improved velocity projection for the material point method. *Computer Modeling in Engineering and Sciences* 2007; **19**(3):223.
13. Wallsteft P, Guilkey J. An evaluation of explicit time integration schemes for use with the generalized interpolation material point method. *Journal of Computational Physics* November 2008; **227**(22):9628–9642, doi: 10.1016/j.jcp.2008.07.019.
14. Wallstedt P, Guilkey J. A weighted least squares particle-in-cell method for solid mechanics. *International Journal for Numerical Methods in Engineering* 2010; doi:10.1002/nme.3041.
15. Moresi L, Dufour F, HB M. A Lagrangian integration point finite element method for large deformation modeling of viscoelastic geomaterials. *Journal of Computational Physics* January 2003; **184**(2):476–497, doi:10.1016/S0021-9991(02)00031-1.
16. Cottet GH, Koumoutsakos PD. *Vortex Methods: Theory and Practice*. Cambridge University Press: Cambridge CB2 2RU, UK, 2000.
17. Chorin AJ. Numerical study of slightly viscous flow. *Journal of Fluid Mechanics Digital Archive* 1973; **57**(04):785–796, doi:10.1017/S0022112073002016.
18. Cottet GH, Weynans L. Particle methods revisited: a class of high order finite-difference methods. *C. R. Math.* Jul 2006; **343**(1):51–56.
19. Bergdorf M, Koumoutsakos P. A lagrangian particle-wavelet method. *Multiscale Modeling and Simulation* 2006; **5**:980–995, doi:10.1137/060652877.
20. Burgess D, Sulsky D, Brackbill J. Mass matrix formulation of the FLIP particle-in-cell method. *Journal of Computational Physics* November 1992; **103**(1):1–15, doi:10.1016/0021-9991(92)90323-Q.
21. Lancaster P, Salkauskas K. Surfaces generated by moving least squares methods. *Mathematics of Computation* 1981; **37**(155):141—158, doi:10.2307/2007507.
22. Gooch CFO. Quasi-ENO Schemes for Unstructured Meshes Based on Unlimited Data-Dependent Least-Squares Reconstruction. *Journal of Computational Physics* May 1997; **133**(1):6–17, doi:10.1006/jcph.1996.5584.
23. Macdonald CB, Ruuth SJ. Level Set Equations on Surfaces via the Closest Point Method. *Journal of Scientific Computing* March 2008; **35**(2-3):219–240, doi:10.1007/s10915-008-9196-6.
24. Brackbill J. The ringing instability in particle-in-cell calculations of low-speed flow. *Journal of Computational Physics* 1988; **75**(2):469–492.
25. Peyret R. *Spectral Methods for Incompressible Viscous Flow*. Springer, 2002.
26. Chorin AJ. Numerical Solution of the Navier-Stokes Equations. *Mathematics of Computation* 1968; **22**(104):745 – 762.
27. Jiang G. A High-Order WENO Finite Difference Scheme for the Equations of Ideal Magnetohydrodynamics. *Journal of Computational Physics* April 1999; **150**(2):561–594, doi:10.1006/jcph.1999.6207.
28. Dagum L, Menon R. OpenMP: an industry standard API for shared-memory programming. *IEEE Computational Science and Engineering* 1998; **5**(1):46–55.

DOI: 10.1002/nme

29. Frigo M, Johnson SG. The design and implementation of FFTW3. *Proceedings of the IEEE* 2005; **93**(2):216–231. Special issue on "Program Generation, Optimization, and Platform Adaptation".
30. Taylor GI, Green AE. Mechanism of the production of small eddies from large ones. *Proceedings of the Royal Society of London. Series A, Mathematical and Physical Sciences* 1937; **158**(895):499–521.
31. Fauconnier D, De Langhe C, Dick E. Construction of explicit and implicit dynamic finite difference schemes and application to the large-eddy simulation of the taylor-green vortex. *J. Comput. Phys.* 2009; **228**(21):8053–8084, doi:10.1016/j.jcp.2009.07.028.
32. Maxworthy T. The structure and stability of vortex rings. *Journal of Fluid Mechanics* 1972; **51**(01):15–32, doi: 10.1017/S0022112072001041.
33. Widnall SE, Sullivan JP. On the Stability of Vortex Rings. *Proceedings of the Royal Society of London. Series A, Mathematical and Physical Sciences* 1973; **332**(1590):335–353.

DOI: 10.1002/nme

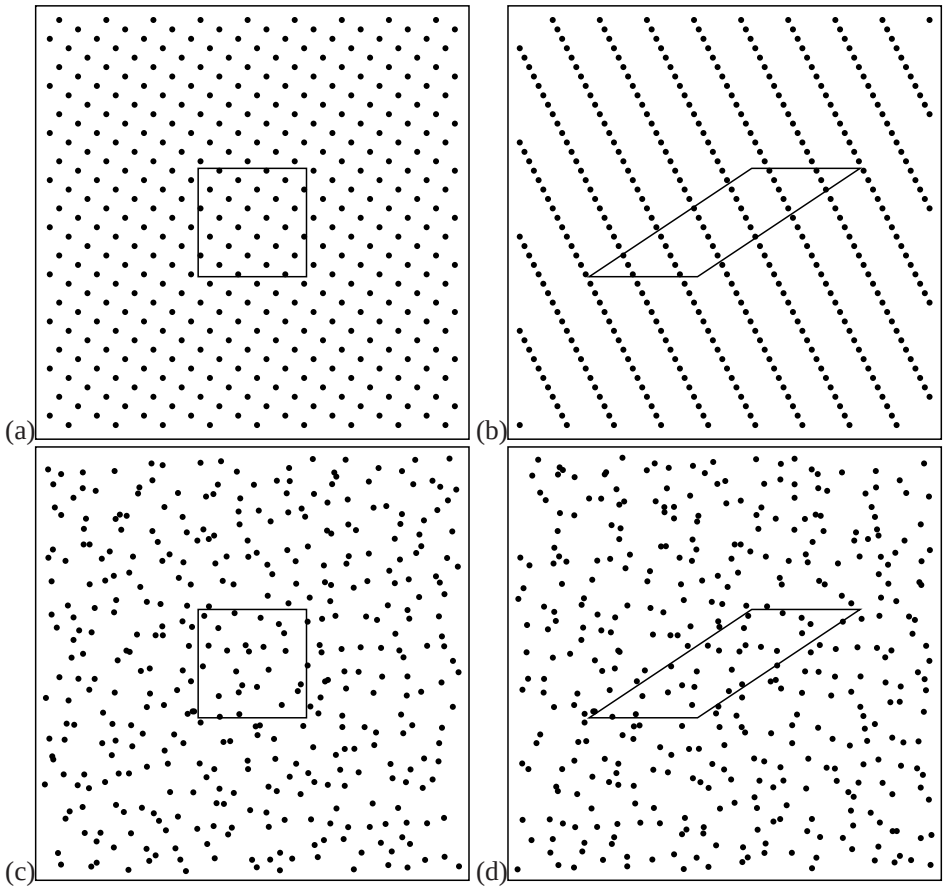


Figure 1. A comparison of structured and random sampling. In (a), a structured grid sampling is used, which when deformed by a uniform shear flow results in (b), exhibiting severe degradation of sampling quality. In (c), the structured grid samples are randomly jittered, eliminating all directional bias. Under the same shear flow, (d) illustrates how the random distribution remains reasonable and unbiased.

DOI: 10.1002/nme

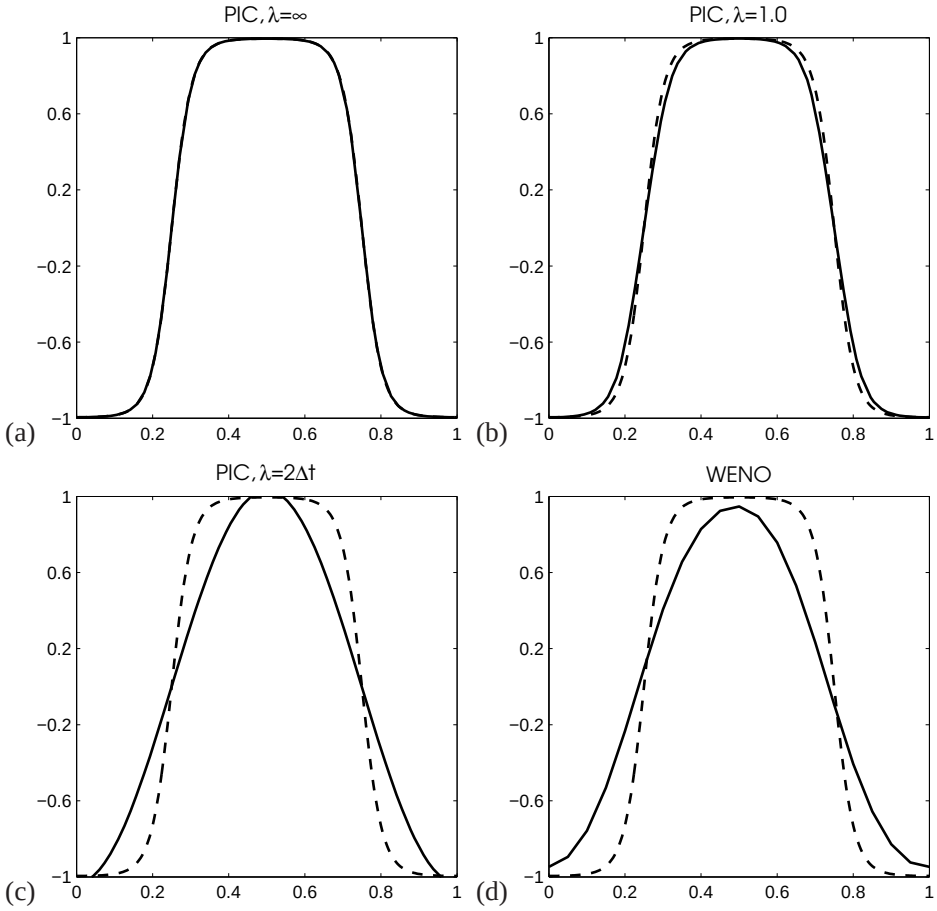


Figure 2. A smoothed pulse is advected across a periodic domain ten times to demonstrate the low numerical diffusion due to our regularization scheme. In (a), no regularization is used and PIC maintains exactly the correct solution. With a more typical half-life of one domain-crossing time, the PIC result (solid) shows minor diffusion relative to the exact result (dashed). When the regularization is increased to its effective limits (half-life on the order of a timestep) the result is very diffusive (c), but no worse than the fifth-order Eulerian WENO scheme in (d).

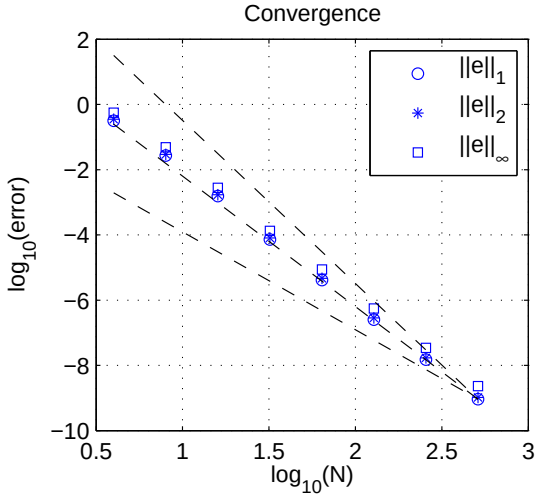


Figure 3. Convergence of HOPIC to the analytic solution of advecting and diffusing $\sin(x+y)$ in a uniform velocity field. Dashed lines show third-, fourth-, and fifth-order convergence.

Prepared using nmeauth.cls
DOI: 10.1002/nme

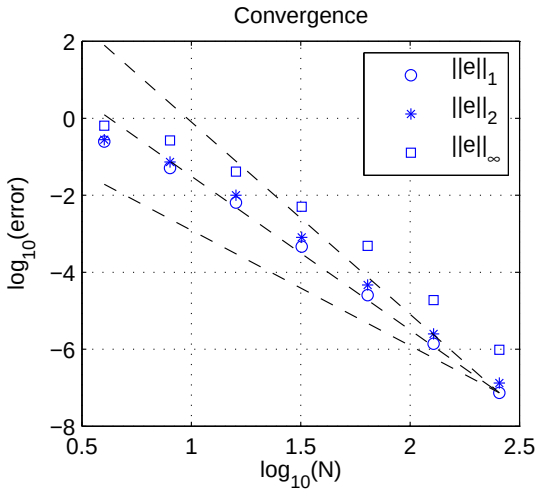


Figure 4. Convergence of HOPIC to a high resolution numerical solution of advection-diffusion in a vortex. Dashed lines show third-, fourth-, and fifth-order convergence.

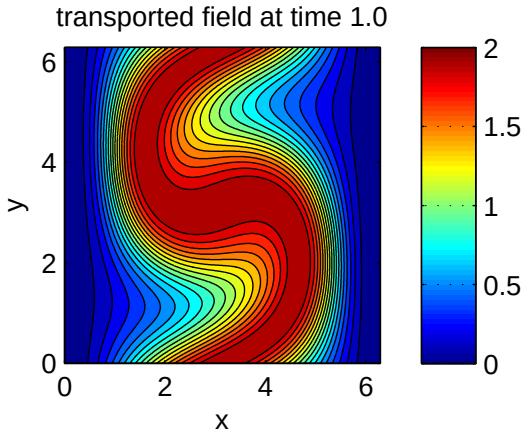


Figure 5. The final state at time $t = 1$ of the field $1 - \cos(x)$ following advection and diffusion in a vortex.

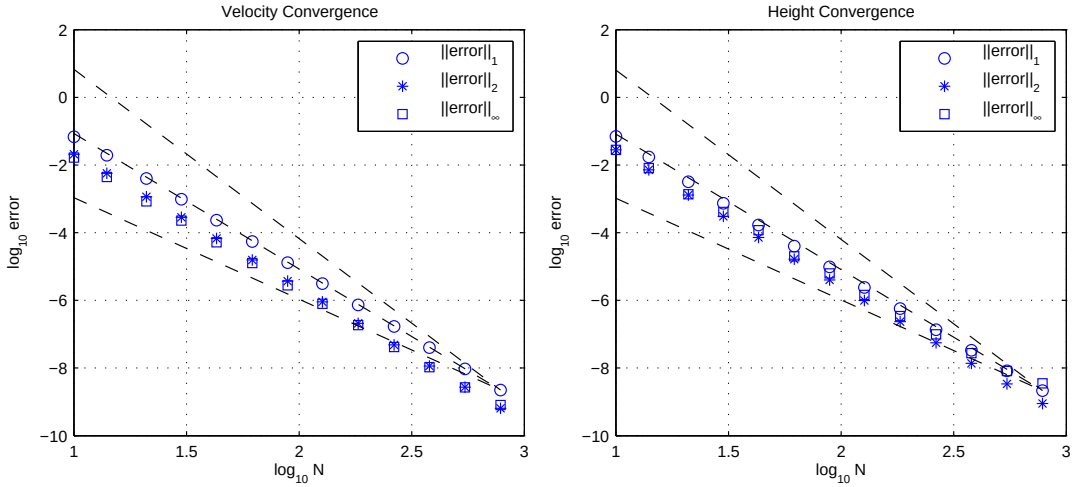


Figure 6. Convergence of HOPIC to a manufactured solution for the shallow water equations. The dashed lines show third-, fourth-, and fifth-order convergence.

DOI: 10.1002/nme

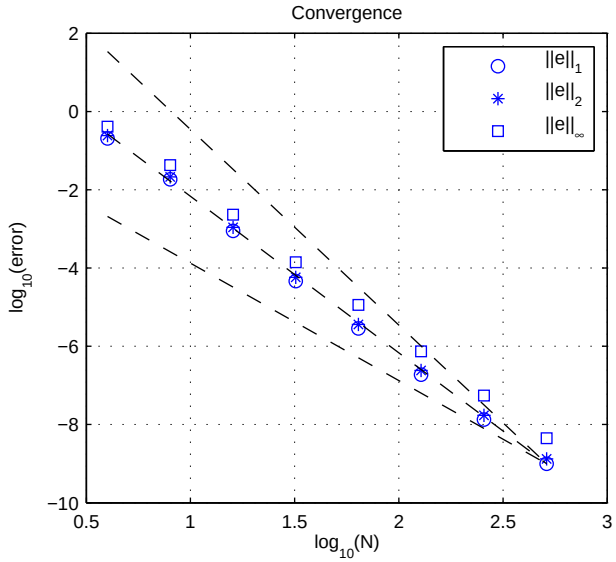


Figure 7. Convergence of HOPIC to the analytic Taylor-Green flow in 2D. Dashed lines show third-, fourth-, and fifth-order convergence.

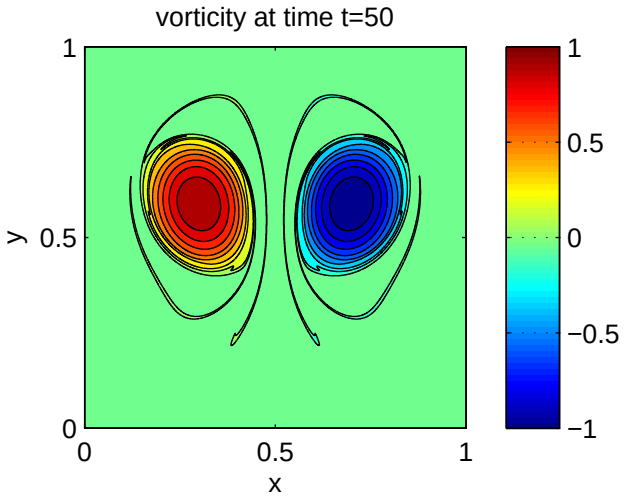


Figure 8. Vorticity distribution in the 2D dipole advection test problem. The initially radially-symmetric vortices wobble and distort as they move.

DOI: 10.1002/nme

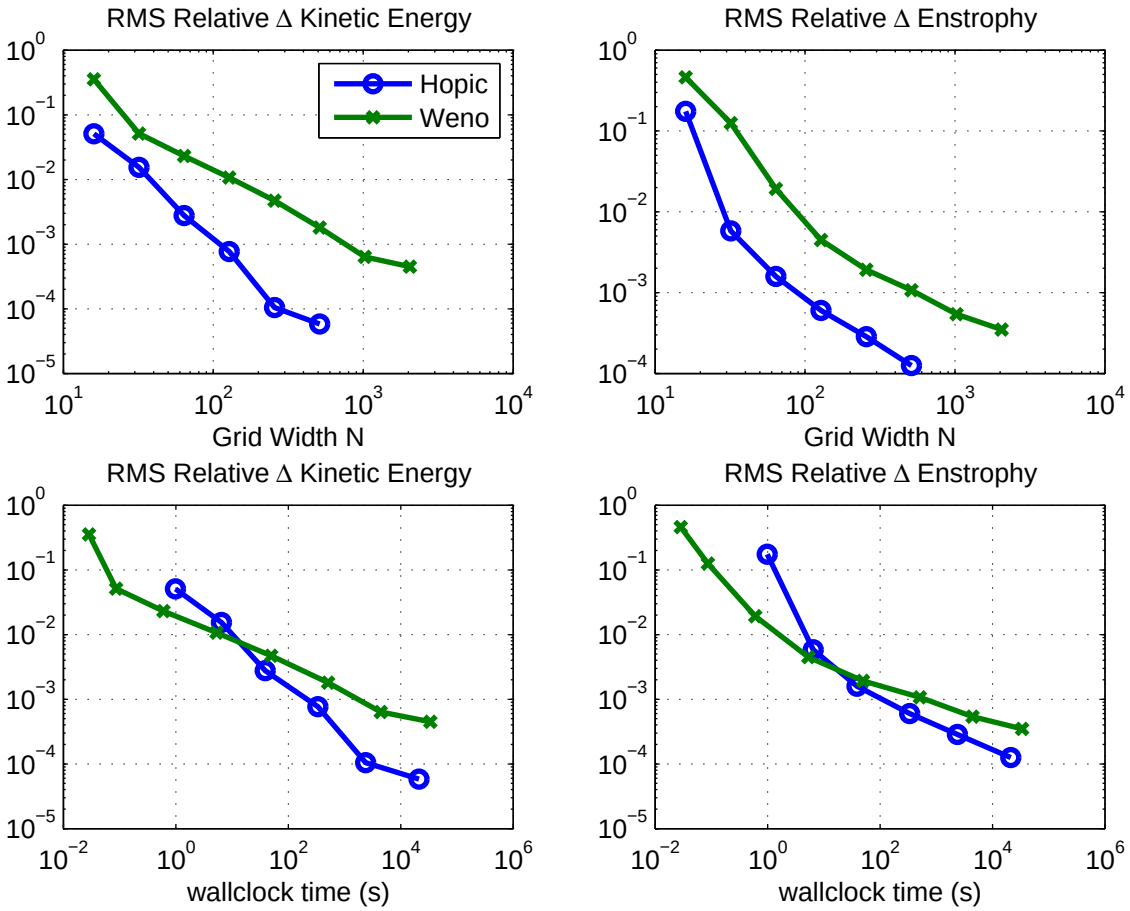


Figure 9. Performance comparison of HOPIC and Eulerian WENO scheme. Top row shows accuracy plotted against the size of the $N \times N$ grid. Bottom row shows accuracy plotted against total simulation runtime. Accuracy is presented in terms of drift in the conserved quantities: kinetic energy (left) and enstrophy (right).

DOI: 10.1002/nme

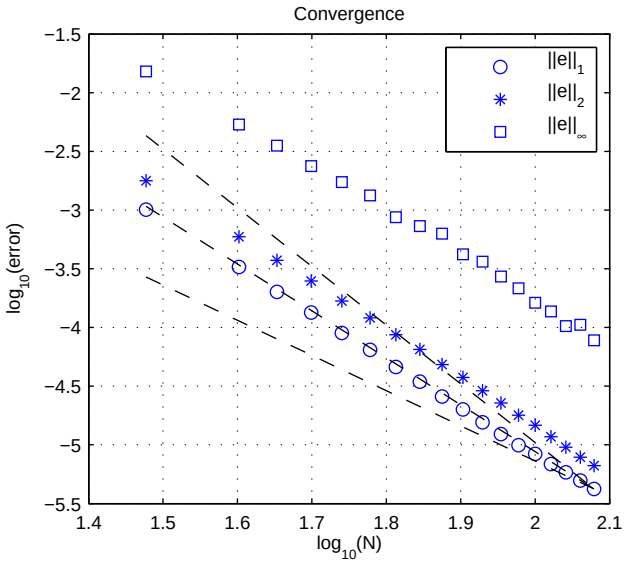


Figure 10. Convergence of HOPIC on the 3D Taylor Green flow. Error is measured at time $t = 1$, against a $150 \times 150 \times 150$ simulation. Dashed lines show third-, fourth-, and fifth-order convergence.

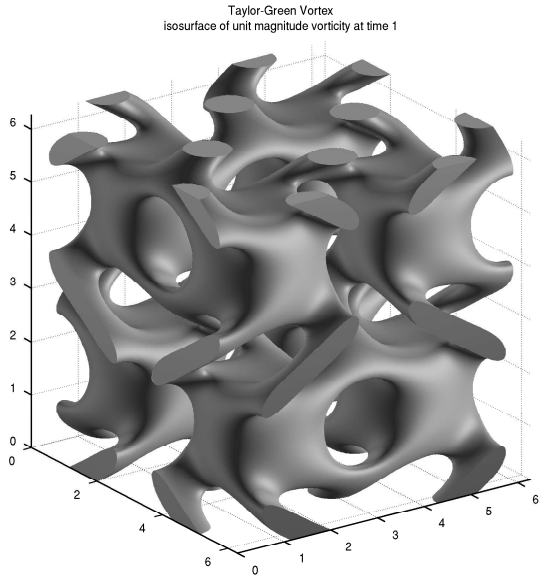


Figure 11. The isosurface of unit magnitude vorticity, of the 3D Taylor-Green vortex, at time t=1.

DOI: 10.1002/nme

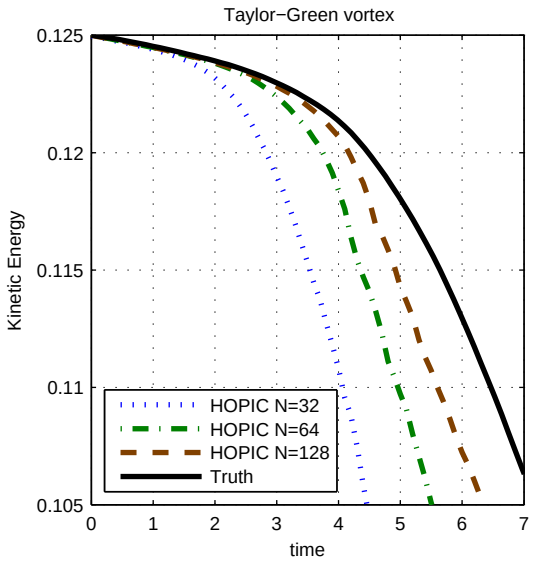
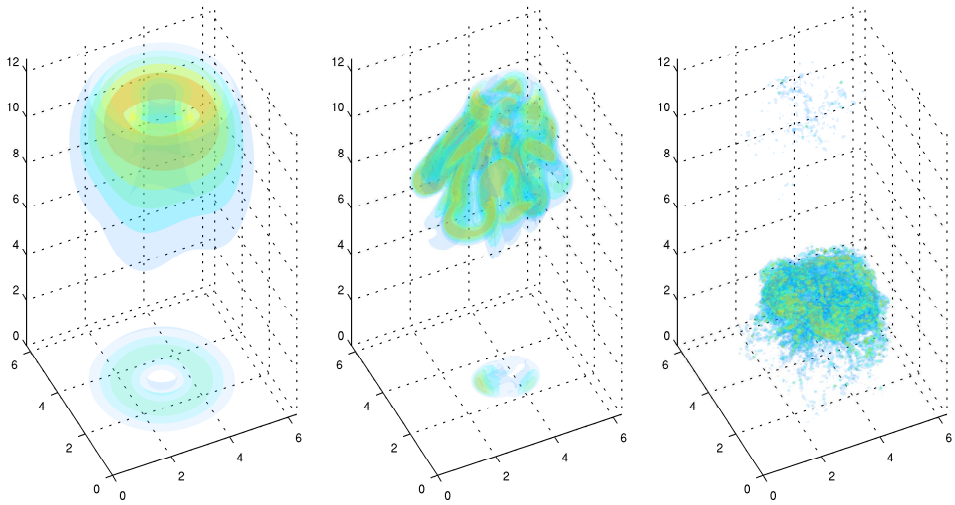


Figure 12. Kinetic energy of the decaying 3D Taylor Green Vortex. 'Truth' is from a high resolution large eddy simulation [31]

.



(a) $\nu = 1/500$, $t = 600$ (b) $\nu = 1/5000$, $t = 800$ (c) $\nu = 1/50000$, $t = 500$

Figure 13. Vortex rings of varying vorticity. Plots show volume renderings of the magnitude of vorticity. At high viscosity, the ring is stable (a). Decreasing viscosity produces unstable vortex rings (b) which collapse more quickly and violently (c).

DOI: 10.1002/nme