Advection Solver Performance with Long Time Steps, and Strategies for Fast and Accurate Numerical Implementation

Jerry Tessendorf*
Digital Production Arts
School of Computing
Clemson University

February 4, 2015

1 Introduction

This note addresses the subject of advection around the theme of applying Characteristic Maps as a method of constructing solvers. Typically the accuracy of solvers is characterized as an asymptotic function of a short time step, short usually defined by restricting the value of the CFL parameter. Within the range of the CFL limit the asymptotic function is expressed as a formula in terms of a power of the time step, i.e. $O(\Delta t^p)$, for some power p.

A key issue is the implementation of solvers that are accurate for "long" time steps. Long time steps might be described as situations that violate CFL restrictions, or at least situations in which it is ambiguous whether the solver can be expected to work based on the CFL restriction. Of course, an advection solver can be used well beyond its CFL limit, with consequent error in the advection. Characterizing the magnitude of that error is of interest. A common approach for computing error is to advect a field over a period of time with some number of steps, then reverse the advection so that, in principle, the field should return to its original form. The error measure is related to the disparity between the original field and the advected one. Defining advection solvers in terms of characteristic maps offers another measure of accuracy. Characteristic maps depend only on the velocity field, and exist independent of the material subject to advection. Given a "reliable" characteristic map, the error of a particular solver could be assessed without invoking any particular material model. This approach is employed in section 7, along with "traditional" evaluation methods.

These issues are addressed here in three stages:

*email: jtessen@clemson.edu

1. Build accurate and fast advection solvers for long time steps using the composition property of characteristic maps. This process is somewhat similar to the Geometric Integration approach for dynamical systems, in that the algorithm starts with a solver that is accurate for short times and builds up one for long times. The product solver has less error than the original. In the case of Geometric Integration, the product solver would have less asymptotic error, i.e. if the original solver has asymptotic error $O(\Delta t^p)$ the product solver would be $O(\Delta t^{p+2})$ or higher. But in this case, the asymptotic error for the product solver is the same as the original, but with a significantly smaller proportionality. However, the construction of the long time advection solver is fast. Typically when a long advection time $T \gg \Delta t$ is desired, the object must be advected over the short time step Δt for $N = T/\Delta t$ times. However, the composition property of the CM generates a solver in $\log(T/\Delta t)$ steps. For example, if $T = 1024\Delta t$, the typical method would require 1024 advections, while the composition approach requires only 10, and has an error 10/1024 times smaller. This is presented in section 3.

2. Build the exact solution of the characteristic map equation. In section 4 the exact solution of the characteristic map equation is derived, apparently for the first time. The solution is explicit, but not analytic. However, a general numerical algorithm, named here "Gradient Stretch," follows in section 6, that is relatively simple but potentially time consuming because accurate exponentiation of 3×3 matrices is required. The appendix includes a discussion of accurate matrix exponentiation. One of the "traditional" test cases for advection solvers employs a velocity field corresponding to a rigid rotation. This test has the fortunate property that the exact characteristic map can be completely evaluated analytically, and as should be anticipated, the map is a rotation transformation in standard form. The general numerical implementation accurately generates this result also.

3. Characterize solver error relative to the exact characteristic map. Typical tests of solver accuracy include two ingredients: a specific velocity field, and an initial object to be advected. The test conducts advection in a way that the exact outcome is known. The measure of solver error follows from the disparity between the advected and expected object. Three of these tests are reproduced in section 7, comparing four different solvers: Semi-Lagrangian, BFECC, Modified MacCormack, and Gradient Stretch. All of these tests involve multiple steps of advection and storing intermediate data on grids. The errors of the solvers cannot be determined directly because the grid storage and interpolation requirement adds errors to the tests. The coupled impact of gridding with multiple advections is demonstrated qualitatively and visually. We also see a set of quantitative error measures by comparing the Gradient Stretch solver directly to other solvers. At short time steps, the classic asymptotic error forms

are reproduced, and at long time steps the solvers transition to a different behavior that has less, but still substantial, error than projected from the asymptotic formulae. This long time error behavior may hint at a "universal" error behavior because the solvers seem to converge on nearly identical behavior.

Before carrying out the outlined stages, the next section reviews the characteristic map formulation of the advection problem. Solvers for Semi-Lagrangian, BFECC, and Modified MacCormack are written in terms of characteristic maps. The approach is phrased for a wide range of velocity fields, including those that are not divergence-free.

2 Characteristic Map Solvers

Advection problems can formally be solved in terms of a vector field called the Characteristic Map (CM) [1], which maps a point in space back to the originating point from which material advected. Generally the CM tracks the time of origination, t', and the current time, t. If the CM is labelled $\mathbf{X}(\mathbf{x}, t, t')$, a field ϕ advected over a time interval Δt updates to the value

$$\phi(\mathbf{x}, t + \Delta t) \;=\; \det\left(\nabla \mathbf{X}(\mathbf{x}, t + \Delta t, t)\right)\; \phi(\mathbf{X}(\mathbf{x}, t + \Delta t, t), t) \tag{1}$$

Scaling by the determinant accounts for changes in field magnitude due to concentration or rarifaction by the underlying velocity field, which may be either compressible or incompressible. This is an exact expression for advection. The time interval Δt is not assumed to be small or large, although it is assumed here to be positive.

The prototypical advection problem in PDE form is

$$\frac{\partial \phi(\mathbf{x}, t)}{\partial t} \;+\; \nabla \cdot (\mathbf{u}(\mathbf{x})\; \phi(\mathbf{x}, t)) \;=\; S(\mathbf{x}, t) \tag{2}$$

where ϕ is the material, and S is an external source/sink of material. The material is assumed to have an initial distribution

$$\phi(\mathbf{x}, t = 0) \;=\; \phi_0(\mathbf{x}) \tag{3}$$

The advection equation has an conservation law for the total amount of the material. Integrating this equation over all 3D space, and assuming there is no material located at spatial infinity, the conservation law is

$$\frac{d}{dt} \int d^3x\; \phi(\mathbf{x}, t) \;=\; \int d^3x\; S(\mathbf{x}, t) \tag{4}$$

which shows that the total amount of the material varies over time due solely to sources and sinks. Advection does not cause net gain or loss of material.

The characteristic map solution of equation 2 is

$$\begin{aligned}\phi(\mathbf{x},t) &= \det\left(\nabla\mathbf{X}(\mathbf{x},t,0)\right)\ \phi_0(\mathbf{X}(\mathbf{x},t,0)) \\ &+ \int_0^t dt'\ \det\left(\nabla\mathbf{X}(\mathbf{x},t,t')\right)\ S(\mathbf{X}(\mathbf{x},t,t'),t') \end{aligned} \tag{5}$$

with the Characteric Map satisfying

$$\frac{\partial\mathbf{X}(\mathbf{x},t,t')}{\partial t} + \mathbf{u}(\mathbf{x})\cdot\nabla\mathbf{X}(\mathbf{x},t,t') = 0 \tag{6}$$

and initial condition $\mathbf{X}(\mathbf{x},t',t') = \mathbf{x}$. This solution also explicitly satisfies the conservation property in equation 4. Integrating this solution over all of 3D space, we have

$$\begin{aligned}\int d^3x\ \phi(\mathbf{x},t) &= \int d^3x\ \det\left(\nabla\mathbf{X}(\mathbf{x},t,0)\right)\ \phi_0(\mathbf{X}(\mathbf{x},t,0)) \\ &+ \int_0^t dt'\ \int d^3x\ \det\left(\nabla\mathbf{X}(\mathbf{x},t,t')\right)\ S(\mathbf{X}(\mathbf{x},t,t'),t') \end{aligned} \tag{7}$$

The combination

$$\int d^3x\ \det\left(\nabla\mathbf{X}(\mathbf{x},t,t')\right) \tag{8}$$

signals a change of integration variable from $\mathbf{x}$ to $\mathbf{X}$. Applying this change,

$$\begin{aligned}\int d^3x\ \phi(\mathbf{x},t) &= \int d^3x\ \phi_0(\mathbf{x}) \\ &+ \int_0^t dt'\ \int d^3x\ S(\mathbf{x},t') \end{aligned} \tag{9}$$

which is equivalent to equation 4.

In a numerical setting in which the material and velocity data may exist on a grid, evaluating advection using the CM means that it is necessary to interpolate the gridded data using whatever interpolation algorithm is of interest. Assuming the interpolation algorithm is bounded, advection via equation 5 is unconditionally stable for the same reason that Semi-Lagrangian advection is unconditionally stable, i.e., the update is bounded by gridded values of the field. But unlike Semi-Lagrangtian advection, we might want our CM advection solver to be valid for higher orders of Δt than linear.

For the remainder of this note we only look at advection over a "single timestep", meaning effectively that the velocity field is constant-in-time during the advection. In this situation, the CM has a time shift symmetry, i.e.

$$\mathbf{X}(\mathbf{x},t,t') = \mathbf{X}(\mathbf{x},t-t',0) \tag{10}$$

This lets us simplify the notation from $\mathbf{X}(\mathbf{x},t,0)$ to $\mathbf{X}(\mathbf{x},t)$.

The CM solver corresponding to Semi-Lagrangian advection is

$$\mathbf{X}_{SL}(\mathbf{x},\Delta t) = \mathbf{x} - \mathbf{u}(\mathbf{x},t)\Delta t \tag{11}$$

Although Semi-Lagrangian error is $O(\Delta t^2)$, other advection schemes have formally smaller error. The BFECC[1] algorithm is constructed from the semi-lagrangian advection as

$$\mathbf{X}_{BFECC}(\mathbf{x}, \Delta t) \;=\; \mathbf{X}_{SL}\left(\mathbf{x} + \frac{1}{2}\left(\mathbf{x} - \mathbf{X}_{SL}\left(\mathbf{X}_{SL}(\mathbf{x}, \Delta t), -\Delta t\right)\right), \Delta t\right) \tag{12}$$

and Modified MacCormack is

$$\mathbf{X}_{MM}(\mathbf{x}, \Delta t) \;=\; \mathbf{X}_{SL}(\mathbf{x}, \Delta t) \;+ \frac{1}{2}\left(\mathbf{x} - \mathbf{X}_{SL}\left(\mathbf{X}_{SL}(\mathbf{x}, \Delta t), -\Delta t\right)\right) \tag{13}$$

Both of these advection schemes have asymptotic error of $O(\Delta t^3)$.

The ideal advection scheme would provide a method to insure as much accuracy as needed for a particular problem. One solution is to employ even higher order advection schemes. But an issue of interest is maintaining accuracy when the time step is large, and it is not known whether high order advection schemes hurt or help.

These issues occur in related dynamical problems in classical mechanics, where the approach of Geometric Integration (GI) fruitfully guides better quality and flexibility in solver construction. GI provides explicit strategies for constructing solvers with high order accuracy from simpler, less-accurate solvers. For example, given a solver $S(\Delta t)$ that updates a dynamical system over time interval Δt with asymptotic error $O(\Delta t^{2p})$, the solver

$$S(\gamma \Delta t)\; S((1-2\gamma)\Delta t)\; S(\gamma \Delta t) \tag{14}$$

where

$$\gamma = \frac{1}{2 - 2^{\frac{1}{2p+1}}} \tag{15}$$

has asymptotic error[2] $O(\Delta t^{2p+2})$. This property is valid universally for Hamiltonian systems of all type, and there are many more similar relationships between solvers. Collectively these universial relationships are valuable tools for selecting a range of accuracy and efficiency criteria for any particular application. No such tools are known for advection.

3 Logarithmic Evaluation of the Characteristic Map

The CM enjoys a composition-in-time property following from the fact that advection conceptually can be deconstructed into a sequence of small advections.

[1] BFECC [3] and Modified MacCormack [5] were not originally constructed as characteristic map solvers. Here they are rebuilt in the language of characteristic maps. Although the error properties and overal structure of these versions follow the original logic, the exact solver is not identical to the originals.

[2] Robert McLachlan and Reinout Quispel, "Six Lectures on the Geometric Integration of ODEs."

Advection solvers that are valid only for small time steps are still useful when long time steps are desired, because a long time advection can be built from iteratively advecting with time steps small enough for the solver to be valid. If the long time step is T, and the solver is valid for time steps $\Delta t < T$, then the material must be advected $N = \text{int}(T/\Delta t)$ steps. For a solver with error $O(\Delta t^p)$, the error for taking a single long step T is $O(T^p)$, whereas taking multiple small steps, the error is $N\ O((T/N)^p)$, or $N^{1-p}\ O(T^p)$. As long as the error exponent p is larger than 1, multiple small advections are more accurate than a single long one. This kind of error analysis does not account for information loss if the data is regridded at each advection step.

The CM has the same composition property for building long-time advections from a sequence of small ones. But for the CM there is an additional fact that the advected CM is also a CM suitable for longer time steps. This is exploitable to reduce the number of advections required from $\text{int}(T/\Delta t)$ to $\text{int}(log_2(T/\Delta t))$. This is a substantial reduction in the number of operations that much be performed, meaning that the advection is faster to execute and there is less regridding loss.

The composition property is the following: given a CM for a time step t_1 and a CM for a time step t_2, the CM for the time step $t_1 + t_2$ follows from the composition of the two:

$$\mathbf{X}(\mathbf{x}, t_1 + t_2) \ = \ \mathbf{X}\left(\mathbf{X}(\mathbf{x}, t_1), t_2\right) \tag{16}$$

In particular, if $t_1 = t_2 = t$,

$$\mathbf{X}(\mathbf{x}, 2t) \ = \ \mathbf{X}\left(\mathbf{X}(\mathbf{x}, t), t\right) \tag{17}$$

This propery sets up the following procedure to construct the CM $\mathbf{X}(\mathbf{x}, T)$ for long time T:

1. Define a short time step $\Delta t \ = \ T/2^M$, for M sufficiently large that Δt is small enough to build an accurate solver.

2. Construct a CM for a chosen short time step Δt, i.e. $\mathbf{X}_0(\mathbf{x}) \ \equiv \ \mathbf{X}(\mathbf{x}, \Delta t)$, using an advection scheme that is accurate for that time step.

3. Construct the following iteration of maps:

$$\mathbf{X}_n(\mathbf{x}) \ = \ \mathbf{X}_{n-1}\left(\mathbf{X}_{n-1}(\mathbf{x})\right), \quad n = 1, \ldots, M \tag{18}$$

The map $\mathbf{X}_M(\mathbf{x})$ is the CM $\mathbf{X}(\mathbf{x}, T)$, and is generated from only M advections of the maps. Normally, a field advected to time $2^M \Delta t$ using an advection scheme accurate for time step Δt requires 2^M advections. This composition rule accelerates long-time advection logarithmically.

How much error is induced by the M advections, called "folds" here, carried out this way? The error for a single short time step is $2^{-pM}\ O(T^p)$, so for the M advections the error accumulates to $M 2^{-pM}\ O(T^p)$. Compared to evaluating all 2^M advections, this error is a factor of $M 2^{-M}$ smaller. In addition, the number

of regridding events is M for this method, as opposed to 2^M previously, so losses due to regridding are reduced as well.

This logarithmic speedup is universally applicable to all advection schemes that are built in terms of characteristic maps.

4 Exact Solution for the Characteristic Map

The exact solution of equation 6 for the CM is an explicit form that depends on the velocity field, the gradient of the velocity field, and the CM at previous times. Derivation starts with constructing the gradient of the CM, $\nabla\mathbf{X}$. From equation 6, it satisfies the evolution equation

$$\frac{\partial \nabla\mathbf{X}(\mathbf{x},t)}{\partial t} \; + \; (\mathbf{u}(\mathbf{x})\cdot\nabla)\;\nabla\mathbf{X}(\mathbf{x},t) \; + (\nabla\mathbf{u}(\mathbf{x}))\cdot\;\nabla\mathbf{X}(\mathbf{x},t) \; = \; 0 \tag{19}$$

The middle term in this equation induces advection by the velocity field. If this advection term were negligible compared to the other two terms, the gradient has a clear behavior:

$$\nabla\mathbf{X}(\mathbf{x},t) \;\approx\; \exp\left(-t\;\nabla\mathbf{u}(\mathbf{x})\right) \tag{20}$$

Similarly, if the term proportional to the gradient of the velocity were negligible compared to the others, the result is advection of the initial gradient field, which is the identity matrix:

$$\nabla\mathbf{X}(\mathbf{x},t) \;\approx\; \nabla\mathbf{X}(\mathbf{X}(\mathbf{x},t),0) \;\approx\; \mathbf{1} \tag{21}$$

In all other situations, where all three terms balance each other, the solution is:

$$\nabla\mathbf{X}(\mathbf{x},t) \;=\; \exp\left(-\int_0^t dt'\;\nabla\mathbf{u}\left(\mathbf{X}(\mathbf{x},t-t')\right)\right)_+ \tag{22}$$

where the notation $(\;)_+$ means that the integral exponential is arranged as an ordered exponential. The definition of an order exponential involves dividing the time interval $(0,t)$ into N segments with time step $\Delta t = t/N$, and arranging them with smallest value of t' on the right most side:

$$\begin{aligned}\exp\left(-\int_0^t dt'\;\nabla\mathbf{u}\left(\mathbf{X}(\mathbf{x},t-t')\right)\right)_+ \;\approx\;& \exp\left(-\Delta t\;\nabla\mathbf{u}\left(\mathbf{x}\right)\right)\\ \times\;& \exp\left(-\Delta t\;\nabla\mathbf{u}\left(\mathbf{X}(\mathbf{x},\Delta t)\right)\right)\\ \times\;& \ldots\\ \times\;& \exp\left(-\Delta t\;\nabla\mathbf{u}\left(\mathbf{X}(\mathbf{x},t-\Delta t)\right)\right)\end{aligned}$$

The limit $N\to\infty$ with $N\Delta t = t$ is the exact solution for $\nabla\mathbf{X}$ in terms of the gradient of the velocity field and the advection field $\mathbf{X}$.

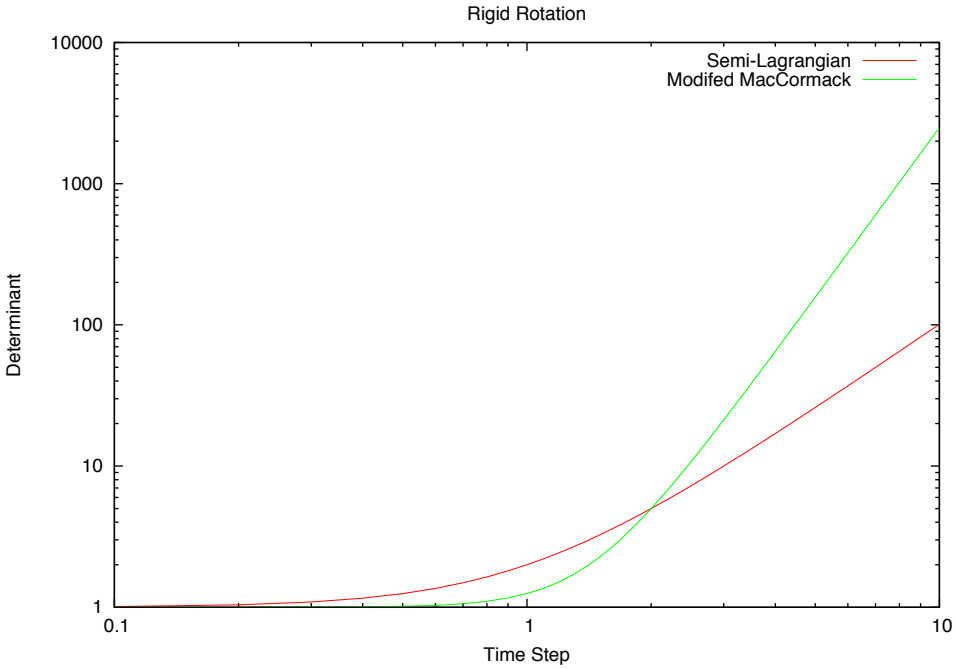

Figure 1: Value of $\det(\nabla \mathbf{X})$ under a rigid rotation, for Semi-Lagrangian and Modified MacCormack solvers. The angular speed is $|\vec{\omega}| = 1$.

This exact expression for the gradient shows several properties. The map gradient is an important quantity for advecting material because its determinant is a factor in equation 5. The determinant for the above expression is

$$\det\left(\nabla \mathbf{X}\right) \;=\; \exp\left(-\int_0^t dt' \;\nabla \cdot \mathbf{u}(\mathbf{X}(\mathbf{x}, t - t'))\right) \tag{23}$$

Two properties come from this expression: (1) For all types of flows, the determinant is positive definite, and the CM is an invertible map; (2) Incompressible flows have a determinant of one.

However, many advection solvers do not enforce this result. An example is shown in figure 1, showing the value of the determinant as a function of the time for the rigid rotation flow. Rigid rotations have an incompressible velocity field $\mathbf{u}(\mathbf{x}) = \mathbf{x} \times \vec{\omega}$, where $\vec{\omega}$ is the angular velocity of the rotation. The deviation from 1 is an error that scales in the same way as the asymptotic error analysis for small time steps, but grows much larger at long times. Although Modified MacCormack has better asymptotic error, at long times it has much larger error than Semi-Lagrangian. In figure 2 the logarithmic evaluation from section 3 has been applied to the solvers with 8 folds, i.e. $M = 8$. The deviation of the

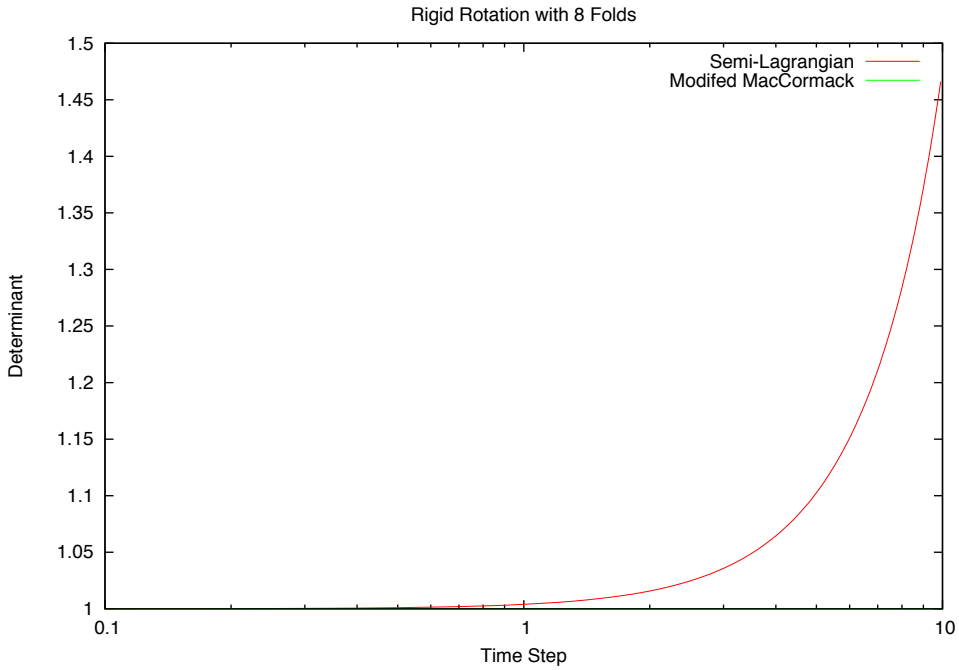

Figure 2: Value of $\det(\nabla \mathbf{X})$ under a rigid rotation, for Semi-Lagrangian and Modified MacCormack solvers and 8 logarithmic folds. Modified MacCormack is not visible in the plot because it deviates from 1 by less than 10^{-5} over this interval. The angular speed is $|\vec{\omega}| = 1$.

determinant from 1 has been reduced by two orders of magnitude for Semi-Lagrangian, and by seven orders of magnitude for Modified MacCormack, even for very long times.

Returning to solving the CM advection equation 6, the exact expression for $\nabla \mathbf{X}$ in equation 22 converts the equation to:

$$\frac{\partial \mathbf{X}(\mathbf{x},t)}{\partial t} = -\mathbf{u}(\mathbf{x}) \cdot \exp\left(-\int_0^t dt' \, \nabla \mathbf{u}\left(\mathbf{X}(\mathbf{x}, t-t')\right)\right)_+ \tag{24}$$

This integrates over time to

$$\mathbf{X}(\mathbf{x},t) = \mathbf{x} - \mathbf{u}(\mathbf{x}) \cdot \int_0^t dt' \, \exp\left(-\int_0^{t'} dt'' \, \nabla \mathbf{u}\left(\mathbf{X}(\mathbf{x}, t'-t'')\right)\right)_+ \tag{25}$$

Equation 25 is the exact solution for the CM. It is an explicit solution in that the CM appearing on the right hand side is for earlier times. This is also a starting point for constructing a new numerical advection scheme in section 6 below.

5 Analytic Solution: Constant Gradient, Rigid Rotation

There is one special case in which the CM can be calculated explicity and exactly: a constant gradient of the velocity field. When $\nabla \mathbf{u}$ is a constant matrix, the ordering of the exponential reduced to ordinary exponentiation, and the time integrals in 25 can be evaluated completely:

$$\int_0^t dt' \ \exp\left(- \int_0^{t'} dt'' \ \nabla \mathbf{u} \right) \ = \ \int_0^t dt' \ \exp\left(-t' \ \nabla \mathbf{u} \right) \ \equiv \ t \ \mathrm{sinch}\left(t \nabla \mathbf{u} \right) \quad (26)$$

If the velocity gradient is an invertible matrix, then

$$\mathrm{sinch}(\mathbf{A}) \ = \ (\mathbf{A})^{-1} \ \left(1 - e^{-\mathbf{A}} \right) \quad (27)$$

When it is not invertible, the definition of sinch follows from the Taylor expansion

$$\mathrm{sinch}(\mathbf{A}) \ = \ \sum_{n=0}^{\infty} \ \frac{(-\mathbf{A})^n}{(n+1)!} \quad (28)$$

One important example of a constant gradient is rigid body rotation. The velocity field is

$$\mathbf{u}(\mathbf{x}) \ = \ \mathbf{x} \times \vec{\omega} \quad (29)$$

where $\vec{\omega}$ is the vorticity of the flow. In this case, the gradient matrix is

$$(\nabla \mathbf{u})_{ij} = \sum_k \ \epsilon_{ijk} \ \omega_k \quad (30)$$

and ϵ_{ijk} is the Levi-Civita symbol. With this choice of velocity field, the evaluation of the sinch has an simple analytic form, and the full result is a rotation transformation:

$$\mathbf{X}(\mathbf{x}, t) \ = \ \mathbf{x} \cos(t\omega) \ + \hat{\omega}(\hat{\omega} \cdot \mathbf{x})(1 - \cos(t\omega)) \ - \ (\hat{\omega} \times \mathbf{x}) \sin(t\omega) \quad (31)$$

with $\omega = |\vec{\omega}|$ and $\hat{\omega} = \vec{\omega}/\omega$. This special exact case provides a concrete illustration of the advection solution 25. It is a useful test of solver accuracy, and can also be used as a check of numerical implementations of equation 25 in section 6.

6 Numerical Implementation of the Exact Solution

There are two alternate approaches for implmenting an advection algorithm based on the exact solution 25. One of them is to create a short-time version that

is iterated to arbitrary time using logarithmic acceleration. A good candidate for a short-time version follows from the solution for constant gradient:

$$\mathbf{X}_{GS}(\mathbf{x}, \Delta t) \;=\; \mathbf{x} \;-\; \mathbf{u}(\mathbf{x}) \cdot \Delta t \; \mathrm{sinch}(\Delta t \nabla \mathbf{u}(\mathbf{x})) \tag{32}$$

Note that (a) this short-time advection is exact for any length of time if the gradient is constant, and (2) if the gradient is constant, and $\mathbf{X}_{GS}$ is used in an iteration, the result is still the exact solution. For any time step T, a short time step can be built by selecting a desired number of folds M and setting $\Delta t = T/2^M$. The logarithmic iteration approach of section 3 would build up the full solver.

A second approach is to divide the time interval into N segments to evaluate the integral in 25. The result is algorithm 1. This second approach also enjoys

Algorithm 1 Gradient Stretch Characteristic Map

procedure GradientStretchCharacteristicMap($\mathbf{u}(\mathbf{x})$, T, N)
  $\Delta t \leftarrow T/N$
  $\mathbf{X}_{GS} \leftarrow \mathbf{x}$
  $\mathbf{M} \leftarrow 0$
  $\mathbf{Q} \leftarrow \Delta t \; \mathrm{sinch}\left(\Delta t \; \nabla \mathbf{u}(\mathbf{x})\right)$
  for $i \leftarrow 0, i < N$ **do**
    $\mathbf{M} \leftarrow \mathbf{M} + \mathbf{Q}$
    $\mathbf{X}_{GS} \leftarrow \mathbf{x} - \mathbf{u}(\mathbf{x}) \cdot \mathbf{M}$
    $\mathbf{Q} \leftarrow \mathbf{Q} \cdot \exp\left(-\Delta t \; \nabla \mathbf{u}(\mathbf{X}_{GS})\right)$
    $i \leftarrow$i + 1
  end for
  return $\mathbf{X}_{GS}$
end procedure

the property that if the gradient is constant, the result is exact, regardless of the choice of N. Note that the choice $N = 1$ reduces algorithm 1 to equation 32.

A third algorithm follows from combining these two. A fold value M and time division N can be used to evaluate algorithm 1 with arguments $(\mathbf{u}, T/2^M, N)$, then iterate that map through M folds.

Despite the assembly of this algorithm from the exact CM, and the robustness of the algorithm for rigid rotations, it has very limited utility for many practical advection scenarios. Accurate computation of the sinch($\mathbf{A}$) function and matrix exponentiation is time consuming. The appendix gives some suggestions based on the effort to produce accurate error analyses in section 7. In comparison, it is much faster to use the CM form of other algorithms, for example Semi-Lagrangian, BFECC, Modified MacCormack, or others, along with the logarithmic iteration process to improve accuracy.

Test	Velocity Field	Domain	$N_i \times N_j \times N_k$	GS M, N
Rigid Rotation	$\frac{\pi}{314}\mathbf{x} \times \vec{\omega}$ $\vec{\omega} = (0, 1, 0)$	$(-100, -100, -100) \times (100, 100, 100)$	$20 \times 20 \times 20$	0, 256
Shear	$(\sin^2(\pi x)\sin(2\pi y)$, $-\sin^2(\pi y)\sin(2\pi x)$, $(1 - r/R)^2)$ $r = ((x - 0.5)^2 + (y - 0.5)^2)^{1/2}$ $R = 0.5$	$(0, 0, 0) \times (1, 1, 1)$	$20 \times 20 \times 20$	11, 20
LeVeque Twist	$(2\sin^2(\pi x)\sin(\pi y)\sin(2\pi z)$, $-\sin^2(\pi y)\sin(\pi z)\sin(2\pi x)$, $-\sin^2(\pi z)\sin(\pi x)\sin(2\pi y))$	$(0, 0, 0) \times (1, 1, 1)$	$100 \times 100 \times 100$	11, 20

Table 1: Velocity fields and calculation domains for the error statistics.

7 Solver Error

One of the standard tests of advection schemes is advection of shapes in a chosen flow, usually reversing the flow after a time and advecting backward for an equal amount of time. The shape is compared before and after advection to assess the accuracy of the advection scheme. These tests, applied to the Semi-Lagrangian, BFECC, Modified MacCormack, and Gradient Stretch algorithms for three different flows listed in table 1, illustrate their relative qualitative performance.

The gradient stretch advection algorithm also provides a reference for estimating the error of other algorithms. Choosing a large value for the fold parameter M for logarithmic iteration, and a large value for the time division parameter N, the value of $\mathbf{X}_{GS}$ can be taken as "ground truth" for comparison with other solvers. An error field defined as

$$\mathcal{E}(\mathbf{x}) \equiv \mathbf{X}_{solver}(\mathbf{x}) - \mathbf{X}_{GS}(\mathbf{x}) \tag{33}$$

is the source of spatially-sampled error statistics for the mean error

$$\langle \mathcal{E} \rangle \equiv \frac{1}{N_i\ N_j\ N_k} \sum_{ijk} \mathcal{E}(\mathbf{x}_{ijk}) \tag{34}$$

and the rms error

$$\sigma_{\mathcal{E}} \equiv \left\{ \frac{1}{N_i\ N_j\ N_k} \sum_{ijk} \left(\mathcal{E}(\mathbf{x}_{ijk}) - \langle \mathcal{E} \rangle\right)^2 \right\}^{1/2} \tag{35}$$

using a grid of points $\mathbf{x}_{ijk}$ from a relevant rectangular domain, as listed in table 1 for each test case. The GS solver parameters M and N were chosen for each case to make sure that the statistics are accurate.

7.1 Rigid Rotation

For rigid rotations, the gradient stretch solver is exact and completely preserves the shape without loss, even for $M = 0$ and $N = 1$. The visual test case [2]

consists of a sphere with a notch removed from it rotating around a point outside the sphere, as shown in figure 3.

For the error statistics, the value of N was 256 in order to insure the accuracy of the matrix sinch and exponentiation operations even for large timesteps. Figure 4 shows the rms error $\sigma_{\mathcal{E}}$ for the three solvers as functions of timestep. Through the entire four decades of time step, the error is the power-law predicted by asymptotic analysis: $O(\Delta t^2)$ for Semi-Lagrangian and $O(\Delta t^3)$ for BFECC and Modified MacCormack.

7.2 Shear

Visually, BFECC, Modified MacCormack, and Gradient Stretch produce essentially identical results for the shear test [4], with a small amount of distortion of the sphere after the advections, and Semi-Lagrangian distortion is substantially greater, as seen in figure 5. For the visual test, the simplest form of gradient stretch was used, i.e. $N = 1$, $M = 0$ corresponding to equation 32.

Figure 6 displays the rms error for the solvers. Quantitatively the asymptotic error holds for BFECC and Modified MacCormack over the range of time scales tested. But near $\Delta t \sim 1$ the Semi-Lagangian error transitions from $O(\Delta t^2)$ to $O(\Delta t)$ at large time steps, with overall lower error.

7.3 LeVeque Twist

As with the rigid rotation and shear tests, Semi-Lagrangian advection produces noticeably larger errors than the other solvers for the LeVeque Twist [4] test, visually demonstrated in figure 7. At small time steps the rms errors exhibit the asymptotic behavior. Near $\Delta t \sim 1$ all three solvers deviate to an error $O(\Delta t)$, in figure 8.

8 Conclusion

The Characteristic Map has been employed systematically in this note as a tool for creating practical, stable, efficient, and accurate advection schemes. The use of a logarithmic folding of advections accelerates the creation of long time step schemes from short time step ones, with much less error. A new "gradient stretch" advection scheme has been derived from the exact expression for the Characteristic Map, although this scheme is computationally heavy due to the need to evaluate matrix-valued exponentials and sinch functions. However, the gradient stretch algorithm enjoys accuracy benefits not shared by other advection schemes, such as being the exact solution in the case of rigid rotation, and highly accurate computation of the determinant of the gradient of the map, with consequent elimination of numerically-induced volume loss or gain.

In the shear and LeVeque Twist test cases, the behavior of the rms error underwent a transition at large time steps. In the shear test, the Semi-Lagrangian scheme transitioned from the short time behavior of $O(\Delta t^2)$ to $O(\Delta t)$, and in

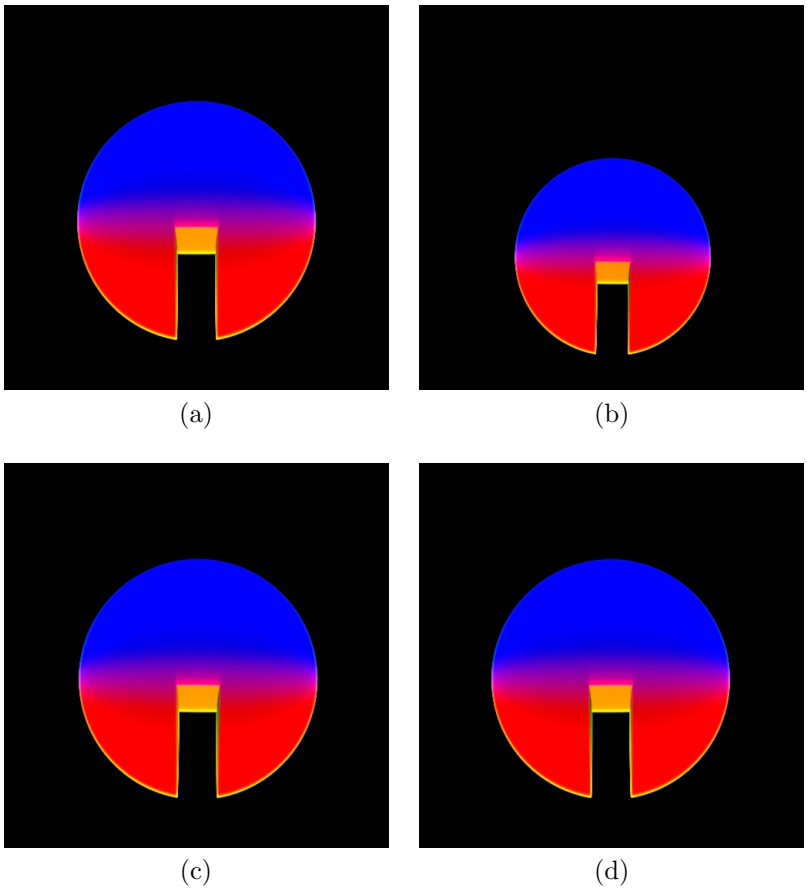

Figure 3: Notched sphere prior to rotation (a); after 360° rotation by Semi-Lagrangian advection (b); BFECC (c); and Modifed MacCormack (d). The time step is 6.28 and there were 100 advections.

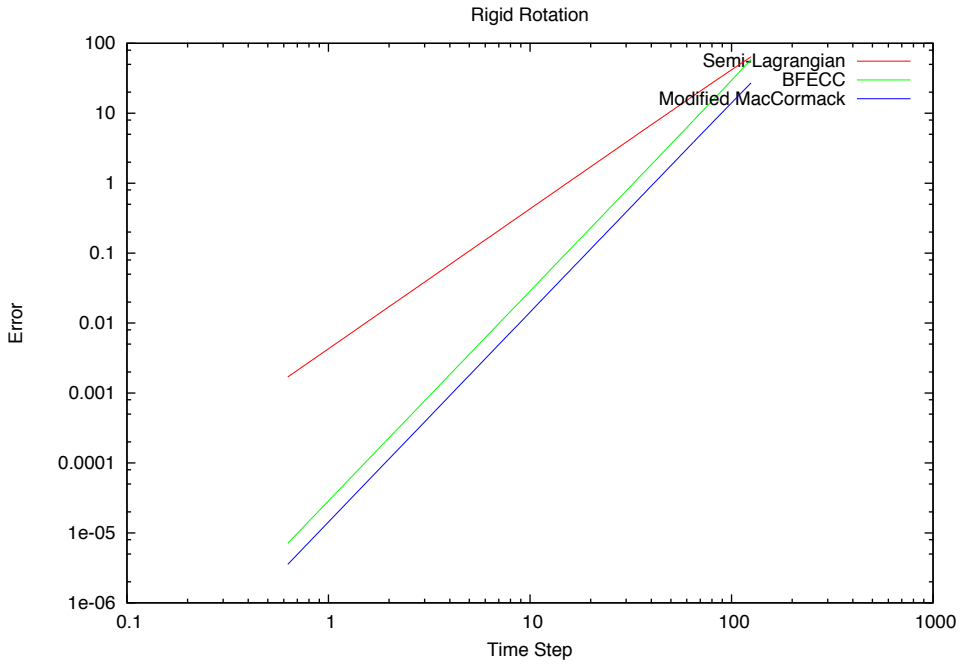

Figure 4: The rms error $\sigma_{\mathcal{E}}$ for the three solvers Semi-Lagrangian, BFECC, and Modified MacCormack, as a function of time step, for the Rigid Rotation case. The CM $\mathbf{X}_{GS}$ was calculated using $M = 0$ and $N = 256$.

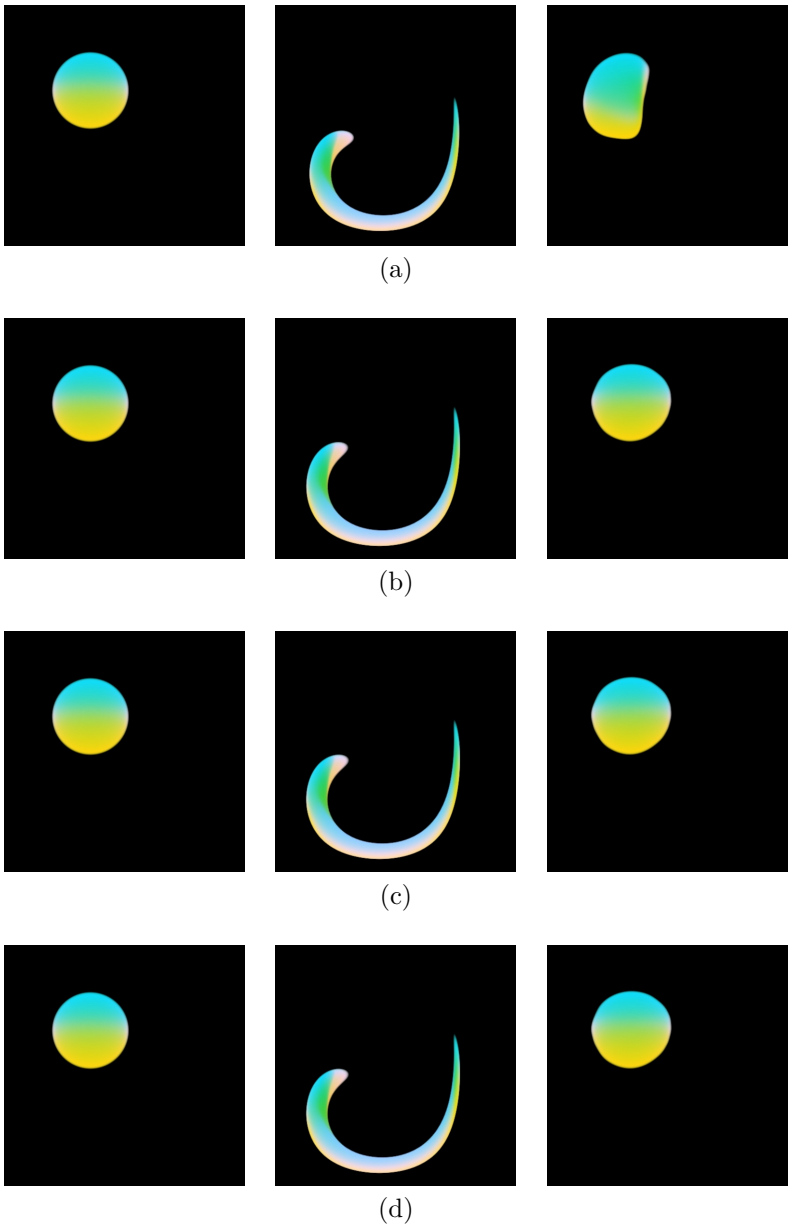

Figure 5: Sphere prior to shear (left) and with maximum shear (center), and returned to original position (right). (a) Semi-Lagrangian; (b) BFECC; (c) Modifed MacCormack; and (d) Gradient Stretch with $N = 1$. The time step was 3/150 and there were 150 advections. The Gradient Stretch case (d) used $N = 1$ and $M = 0$.

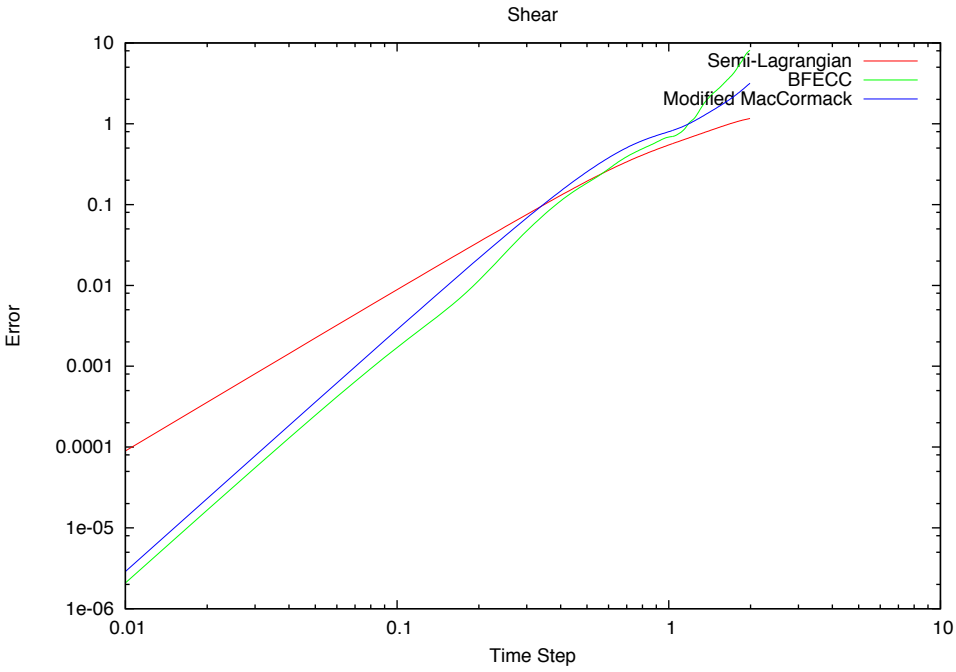

Figure 6: The rms error $\sigma_{\mathcal{E}}$ for the three solvers Semi-Lagrangian, BFECC, and Modified MacCormack, as a function of time step, for the Shear case. The CM $\mathbf{X}_{GS}$ was calculated using $M = 11$ and $N = 20$.

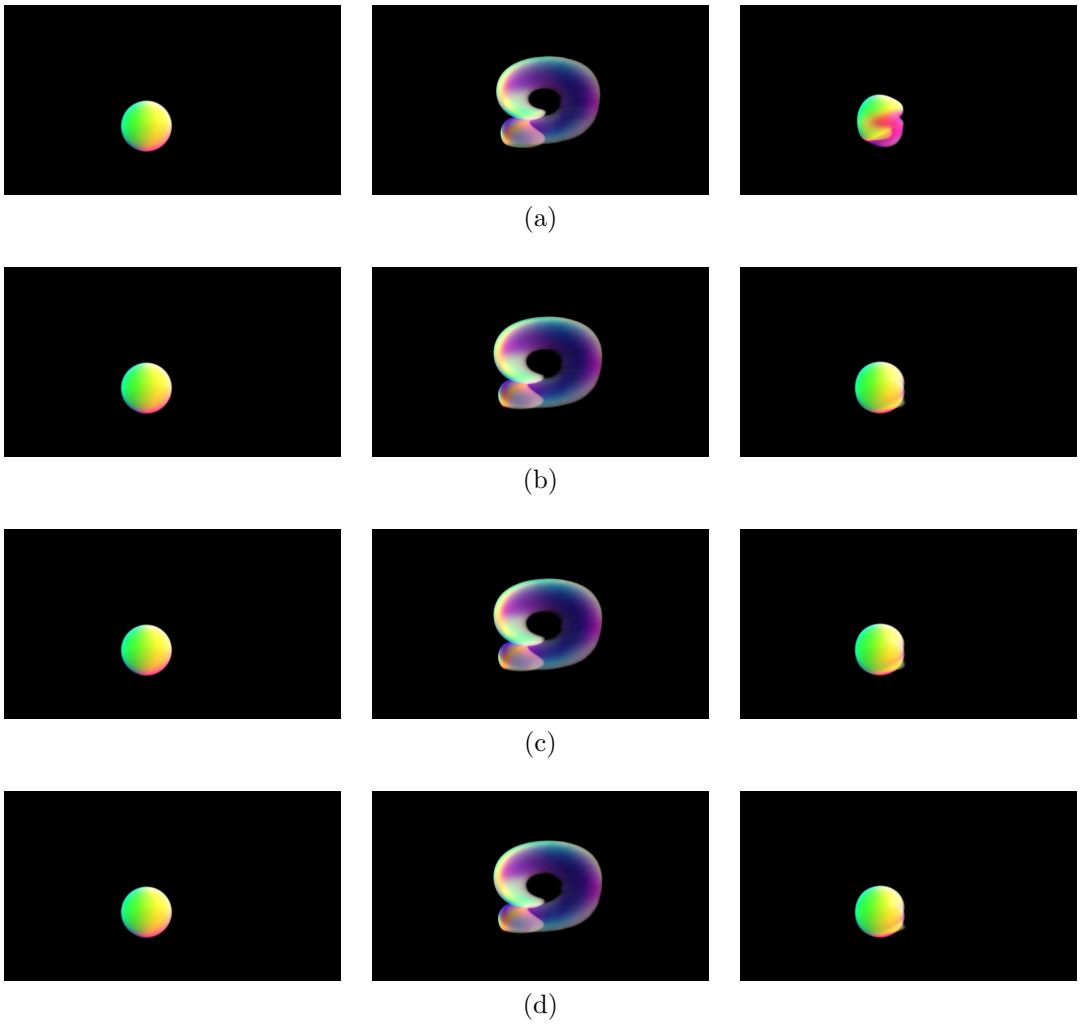

Figure 7: Sphere prior to advection (left) and with maximum advection (center), and returned to original position (right). (a) Semi-Lagrangian; (b) BFECC; (c) Modifed MacCormack; and (d) Gradient Stretch with $N = 6$. The time step was 30/150 and there were 15 advections. Each solver used $M = 4$ folds.

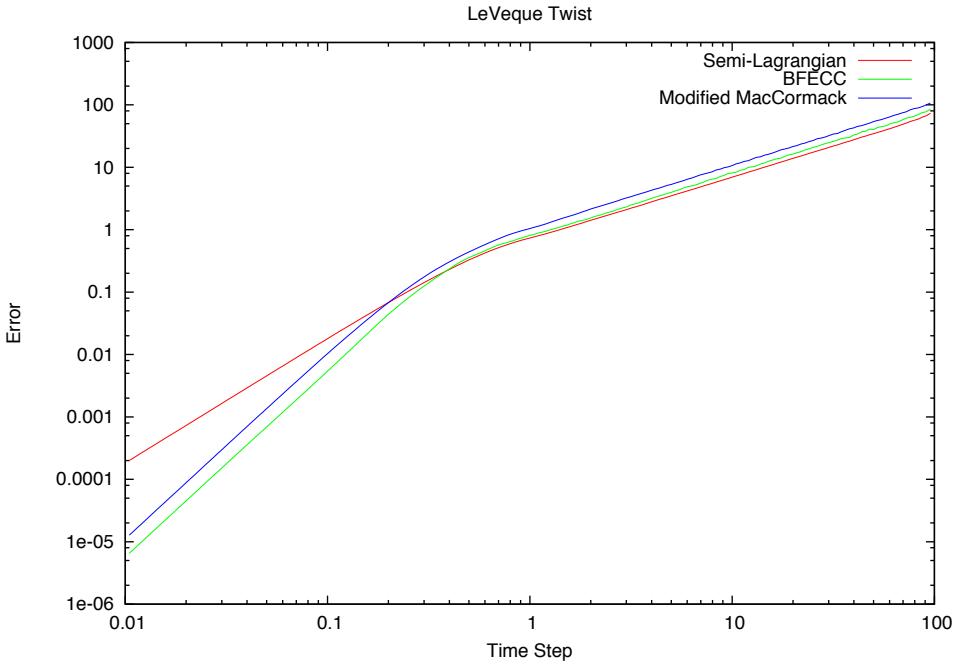

Figure 8: The rms error $\sigma_{\mathcal{E}}$ for the three solvers Semi-Lagrangian, BFECC, and Modified MacCormack, as a function of time step, for the LeVeque Twist case. The CM $\mathbf{X}_{GS}$ was calculated using $M = 11$ and $N = 20$.

the LeVeque Twist test all three schemes made a transition from their respective short time error to nearly identitical $O(\Delta t)$ behavior.

In the interest of motivating future investigation, here is a speculation on the source of this transition. The ordered exponential term can have a wide range of behaviors depending on the magnitude and specifics of its argument. In the case of the rigid rotation, the ordered exponential reduced to a purely oscillatory behavior. When there is a very large argument, i.e. large gradient or large time step or both, strong oscillations in magnitude and phase would be very difficult to reproduce via an advection scheme with an asymptotic error $O(\Delta t^p)$ with p relatively small. In such a situation, the oscillations could dominate an rms error measure such the one used here. Thinking of the ordered exponential as a rapidly fluctuating random variable, the error measure associated with the variance calculation would generate the $O(\Delta t)$ behavior seen in some of the results for the shear and LeVeque Twist cases.

Appendix: Matrix Exponential and Sinch

Matrix exponentiation has a relatively simple implementation in terms of a truncation of the Taylor series

$$e^{\mathbf{A}} \;=\; \sum_{n=0}^{\infty} \frac{\mathbf{A}^n}{n!} \tag{36}$$

to include only the first N powers of $\mathbf{A}$, and is depicted in algorithm 2. For many

Algorithm 2 Matrix exponential via Taylor expansion

```
procedure EXP(A, N)
    E ← 1
    M ← A
    for i ← 1, i ≤ N do
        E ← E + M
        M ← M * A/(i + 1)
        i ←i + 1
    end for
    return E
end procedure
```

choices of $\mathbf{A}$, this approach produces reasonable accuracy for truncations $N \sim 50$. Cases in which the exponential is oscillatory can require many more terms, $N \sim 200$ or more, to insure good reproduction of the phase and amplitude. A good strategy for reducing this load is to take advantage of the multiplicative property of the exponential

$$e^{\mathbf{A}} \;=\; \left(e^{\mathbf{A}/\ell}\right)^{\ell} \tag{37}$$

This relationship is similar to the one for logarithmic acceleration of advection in section 3, and can be exploited similarly in algorithm 3. This algorithm gives

Algorithm 3 Fast matrix exponential via Taylor expansion

```
procedure FastExp(A, N, M)
    E ← Exp(A/2^M, N)
    for i ← 0, i < M do
        E ← E * E
        i ←i + 1
    end for
    return E
end procedure
```

accurate results even for relatively small values of $N \sim 30$, $M \sim 10$ unless $\mathbf{A}$ has elements with very large magnitude.

For the sinch function, there is a simple expression in terms of the exponential when $\mathbf{A}$ is invertible:

$$\text{sinch}(\mathbf{A}) \;=\; \mathbf{A}^{-1} \left(1 \;-\; e^{-\mathbf{A}}\right) \tag{38}$$

When $\mathbf{A}$ is not invertible, there is no choice but to use a relatively time consuming truncation of the Taylor expansion

$$\text{sinch}(\mathbf{A}) \;=\; \sum_{n=0}^{\infty} \frac{(-\mathbf{A})^n}{(n+1)!} \tag{39}$$

as implemented in algorithm 4. In practice, the Taylor expansion truncation

Algorithm 4 Matrix sinch via Taylor expansion

```
procedure Sinch(A, N)
    E ← 1
    M ← −A/2
    for i ← 1, i ≤ N do
        E ← E + M
        M ← M * (−A)/(i + 2)
        i ←i + 1
    end for
    return E
end procedure
```

must be very large, i.e. $N \sim 200$, to produce reasonable accuracy over the range of cases in this note.

References

[1] JohnC. Bowman, MohammadAli Yassaei, and Anup Basu. A fully lagrangian advection scheme. *Journal of Scientific Computing*, pages 1–27, 2014.

[2] Douglas Enright, Ronald Fedkiw, Joel Ferziger, and Ian Mitchell. A hybrid particle level set method for improved interface capturing. *J. Comput. Phys.*, 183(1):83–116, November 2002.

[3] ByungMoon Kim, Yingjie Liu, Ignacio Llamas, and Jarek Rossignac. Flowfixer: Using bfecc for fluid simulation. In Pierre Poulin and Eric Galin, editors, *NPH*, pages 51–56. Eurographics Association, 2005.

[4] Peter Liovic, Murray Rudman, Jong-Leng Liow, Djamel Lakehal, and Doug Kothe. A 3d unsplit-advection volume tracking algorithm with planarity-preserving interface reconstruction. *Computers and Fluids*, 35(10):1011–1032, December 2006.

[5] Andrew Selle, Ronald Fedkiw, Byungmoon Kim, Yingjie Liu, and Jarek Rossignac. An unconditionally stable maccormack method. *J. Sci. Comput.*, 35(2-3):350–371, June 2008.