An Unconditionally Stable MacCormack Method*

Andrew Selle[†] Ronald Fedkiw[†] ByungMoon Kim[‡]
Yingjie Liu[§] Jarek Rossignac[‡]

June 29, 2007

Abstract

The back and forth error compensation and correction (BFECC) method advects the solution forward and then backward in time. The result is compared to the original data to estimate the error. Although inappropriate for parabolic and other non-reversible partial differential equations, it is useful for often troublesome advection terms. The error estimate is used to correct the data before advection raising the method to second order accuracy, even though each individual step is only first order accurate. In this paper, we rewrite the MacCormack method to illustrate that it estimates the error in the same exact fashion as BFECC. The difference is that the MacCormack method uses this error estimate to correct the already computed forward advected data. Thus, it does not require the third advection step in BFECC reducing the cost of the method while still obtaining second order accuracy in space and time. Recent work replaced each of the three BFECC advection steps with a simple first order accurate unconditionally stable semi-Lagrangian method yielding a second order accurate unconditionally stable BFECC scheme. We use a similar approach to create a second order accurate unconditionally stable MacCormack method.

[*] Stanford University, 353 Serra Mall Room 206, Stanford, CA 94305, Phone: 650-714-4426, Fax: 650-723-0033

[†] Compute Science Department, Stanford University, Stanford, CA, email: aselle@cs.stanford.edu

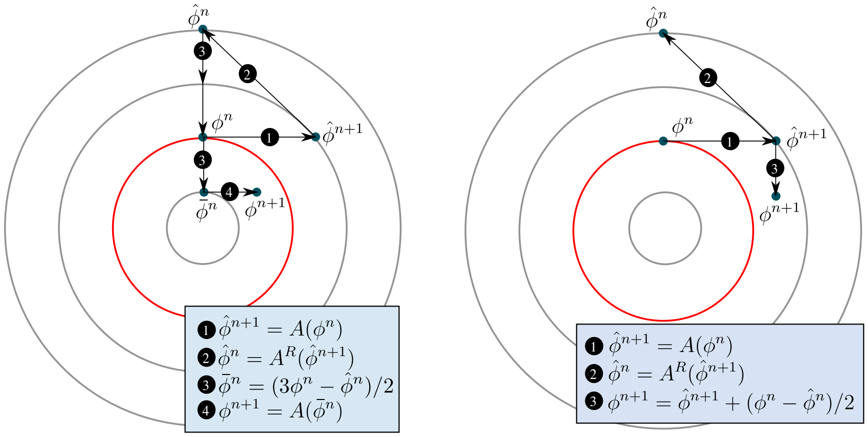
[‡] College of Computing, Georgia Institute of Technology, Atlanta, GA

[§] School of Mathematics, Georgia Institute of Technology, Atlanta, GA

1 Introduction

Courant et al. [2] proposed a simple method of characteristics scheme for discretizing advection equations. These semi-Lagrangian type schemes are popular in many areas (for example in the atmospheric sciences community [25]), because they can be made unconditionally stable. The simplest semi-Lagrangian scheme traces back a straight line characteristic and uses trilinear interpolation to estimate the data, and thus is first order accurate in space and time. The order of accuracy can be improved by tracing back curved characteristics and using higher order interpolation (e.g. [19]). However, this can significantly increase the complexity and computational cost of the method especially since high order polynomial interpolants require limiters to avoid oscillations, new extrema and possible instability. See for example the appendix of [9] which illustrates the use of a non-oscillatory cubic spline interpolant. Another way to improve the fidelity of semi-Lagrangian schemes is via auxiliary information. For example, after [24] popularized the semi-Lagrangian method in the field of computer graphics, [9] showed that vorticity confinement [26] could be used to alleviate the high amount of dissipation allowing for visually intricate, albeit non-physical flows (see also [22] which used vortex particles). Similarly, [6] showed that the simple first order accurate semi-Lagrangian scheme can be used to obtain very accurate level set tracking as long as particles are tracked with higher order accuracy. The original particle level set method [5] used fifth order accurate Hamilton-Jacobi WENO [12] and third order accurate TVD Runge-Kutta [23] making it difficult to extend the method to more complicated data structures. In contrast, the simple semi-Lagrangian approach proposed in [6] allowed for straightforward extension to octree [17, 16] (see also [27]) and run length encoded [11] data structures.

Back and forth error compensation and correction (BFECC) was first proposed in [3] with the aim of reducing mass loss in level set methods (see [21, 20]). The key idea was to realize that a reversible differential equation could be evolved forward and then backward in time to obtain an error estimate. The difference between the final result and the original data is approximately twice the advection error. While inappropriate for non-reversible differential equations such as the heat equation or level set reinitialization [28], it is useful for problematic advection terms in hyperbolic differential equations. First, the forward advection operator A is applied to get $\hat{\phi}^{n+1} = A(\phi^n)$, and then the backward advection operator A^R is applied to get $\hat{\hat{\phi}}^n = A^R(\hat{\phi}^{n+1})$. The result is used to estimate the error in an advection step as $\hat{\hat{\phi}}^n - \phi^n = 2e$ or $e = (\hat{\hat{\phi}}^n - \phi^n)/2$. Next, this error estimate is used to adjust the initial condition of the final advection via $\bar{\phi}^n = \phi^n - e$ and then $\phi^{n+1} = A(\bar{\phi}^n)$. If A was linear, this could be viewed as evolving both the equation and the error forward in


$\hat{\phi}^n$
3
2
ϕ^n
1
$\hat{\phi}^{n+1}$
3
4
$\bar{\phi}^n$
ϕ^{n+1}
❶ $\hat{\phi}^{n+1} = A(\phi^n)$
❷ $\hat{\phi}^n = A^R(\hat{\phi}^{n+1})$
❸ $\bar{\phi}^n = (3\phi^n - \hat{\phi}^n)/2$
❹ $\phi^{n+1} = A(\bar{\phi}^n)$
$\hat{\phi}^n$
2
ϕ^n
1
$\hat{\phi}^{n+1}$
3
ϕ^{n+1}
❶ $\hat{\phi}^{n+1} = A(\phi^n)$
❷ $\hat{\phi}^n = A^R(\hat{\phi}^{n+1})$
❸ $\phi^{n+1} = \hat{\phi}^{n+1} + (\phi^n - \hat{\phi}^n)/2$

(a) Original BFECC

(b) Cheaper BFECC

Figure 1: A conceptual diagram of the steps of the original BFECC method and the cheaper modified BFECC method (which is actually a modified MacCormack scheme).

time, i.e. $\phi^{n+1} = A(\phi^n - e) = A(\phi^n) - A(e)$. Intuitively, this treats e as if it were a time n quantity that needs to be advected forward in time to time $n + 1$. Since there is no strong evidence that e is a time n quantity, one could just as well add it directly to the time $n + 1$ state to obtain $\phi^{n+1} = A(\phi^n) - e = \hat{\phi}^{n+1} - e$. And since $\hat{\phi}^{n+1}$ has already been computed in the error estimation step, this strategy alleviates the need to carry out a third advection step making the method significantly cheaper. Figure 1 shows a conceptual diagram of these two methods applied.

Originally, [3] focused on the application of BFECC to the level set equation $\phi_t + V \cdot \nabla\phi = 0$ using forward Euler time integration combined with first order accurate upwinding and downwinding for the forward and backward time evolution, respectively. More recently, a series of papers [13, 14, 4] showed that the forward and backward advection operators could be replaced by a first order accurate unconditionally stable semi-Lagrangian scheme without changing the desirable properties of the method. This produces an unconditionally stable, fully second order accurate semi-Lagrangian method composed of simple first order accurate building blocks. This has the desirable simplicity of TVD Runge-Kutta methods [23], and interestingly contains backward time evolution similar to the higher order accurate TVD Runge-Kutta methods. Thus, the order of accuracy of the simple semi-Lagrangian scheme can be raised from one to two by increasing the amount of work by a factor of three, since three semi-Lagrangian advections are required. Alternatively, as pointed out above, the error estimate could be added directly to $\hat{\phi}^{n+1}$ removing the last advection step, thus requiring only twice the effort of the first order accurate scheme.

The motivation for this cheaper version of the BFECC scheme came from the MacCormack method [18], which uses a combination of upwinding and downwinding to achieve second order accuracy in space and time. Consider the cheaper version of the BFECC scheme applied to the one dimensional wave equation $\phi_t + \phi_x = 0$, with $\lambda = \Delta t/\Delta x$, $\triangle^- \phi = \phi_i - \phi_{i-1}$ and $\triangle^+ \phi = \phi_{i+1} - \phi_i$. The forward advection step is $\hat{\phi}_i^{n+1} = \phi_i^n - \lambda \triangle^- \phi^n$, the backward advection step is $\hat{\phi}_i^n = \hat{\phi}_i^{n+1} + \lambda \triangle^+ \hat{\phi}^{n+1}$, and the error estimate is $e_i = (\hat{\phi}_i^n - \phi_i^n)/2 = (\hat{\phi}_i^{n+1} - \phi_i^n + \lambda \triangle^+ \hat{\phi}^{n+1})/2$. Finally, $\phi_i^{n+1} = \hat{\phi}_i^{n+1} - e = (\phi_i^n + \hat{\phi}_i^{n+1} - \lambda \triangle^+ \hat{\phi}^{n+1})/2$. This is the same as equation (2b) from the original [18] if one switches $\triangle^-$ with $\triangle^+$ throughout. That is, [18] proposed unstable downwind differencing for the forward step, and unstable upwind differencing for the backward step, whereas we propose the stable versions for both steps. Note that this slight modification is also typically referred to as a MacCormack method or modified MacCormack method, see e.g. [29] and [1][1]. Thus while this particular modification of BFECC is not novel, it adds insight to the (modified) MacCormack method allowing us to extend it to be unconditionally stable via simple semi-Lagrangian building blocks. Moreover, when viewed in this fashion many other improvements can be made. For example, one can automatically revert to the first order accurate scheme when the upwind and downwind building blocks pull data from non-commensurate regions (e.g. if one gets information from the fluid and the other from a solid wall boundary). It also becomes straightforward to apply limiters to prevent new extrema, which is important since the error correction step of both BFECC and our newly proposed MacCormack scheme can produce new extrema leading to instability.

2 Temporal Accuracy and Stability

Consider the model ordinary differential equation $y' = \lambda y$ and the Taylor expansion $y^{n+1} = y^n + \lambda \Delta t y^n + \Delta t^2 \lambda^2 y^n/2 + \Delta t^3 \lambda^3 y^n/6 + O(\Delta t^4)$. Forward Euler time integration is $y_{fe}^{n+1} = (1 + \Delta t \lambda) y^n$ with a leading order truncation error of $\Delta t^2 \lambda^2 y^n/2$. The subsequent step backwards in time from y_{fe}^{n+1} to $\hat{y}^n$ is $\hat{y}^n = (1 - \Delta t \lambda) y_{fe}^{n+1} = (1 - \Delta t^2 \lambda^2) y^n$, so the error is $e = (\hat{y}^n - y^n)/2 = -\Delta t^2 \lambda^2 y^n/2$. BFECC computes a new initial value

[1] page 224

	Forward Euler		BFECC		MacCormack (RK2)	
Δt	Error	Order	Error	Order	Error	Order
1.000	-6.61e-02	–	-3.35e-02	–	1.73e-02	–
.500	-3.35e-02	1.0	-7.13e-03	2.2	3.40e-03	2.3
.250	-1.67e-02	1.0	-1.59e-03	2.2	7.66e-04	2.2
.125	-8.36e-03	1.0	-3.72e-04	2.1	1.82e-04	2.1

Table 1: $y' = -y/2$ with $y_0 = 1.3$. As predicted the error of the MacCormack scheme is about half in magnitude and positive compared the to the back and forth scheme.

$\bar{y}^n = y^n - e$ which is advanced forward in time via $y_{fb}^{n+1} = (1+\Delta t\lambda)\bar{y}^n$, i.e.

$$y_{fb}^{n+1} = \left(1 + \Delta t\lambda + \frac{1}{2}\Delta t^2\lambda^2 + \frac{1}{2}\Delta t^3\lambda^3\right) y^n$$

with a leading order truncation error of $-\Delta t^3\lambda^3 y^n/3$. The version of the MacCormack scheme we consider in this paper is

$$y_m^{n+1} = y_{fe}^{n+1} - e = (1 + \Delta t\lambda + \frac{1}{2}\Delta t^2\lambda^2)y^n$$

with a leading order truncation error of $\Delta t^3\lambda^3 y^n/6$. That is, the MacCormack method is identical to second order accurate Runge-Kutta for ordinary differential equations. All this can be illustrated by solving $y' = -y/2$ with $y_0 = 1.3$ as shown in Table 1.

Next, we consider the regions of stability. As usual, forward Euler requires $|1 + \Delta t\lambda| \leq 1$ so $\Delta t\lambda = x + yi$ must satisfy $(x + 1)^2 + y^2 \leq 1$. Similarly, BFECC requires $(1 + x + \frac{1}{2}x^2 - \frac{1}{2}y^2 + \frac{1}{2}x^3 - \frac{3}{2}xy^2)^2 + (xy + y + \frac{3}{2}x^2y - \frac{1}{2}y^3)^2 \leq 1$. The graphs of the stability regions are shown in Figure 2. Note that the BFECC method includes a significant portion of the imaginary axis similar to third order TVD Runge-Kutta.

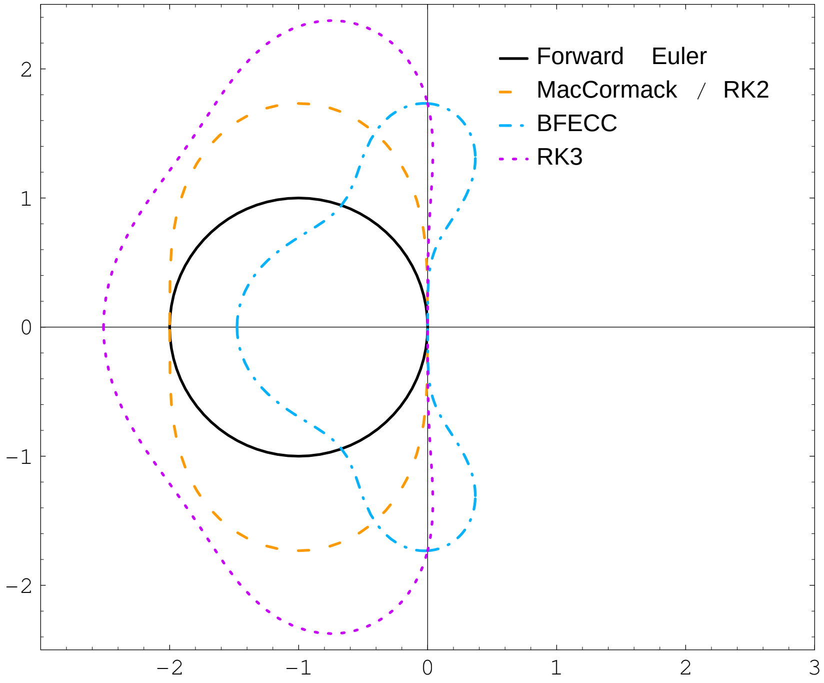


Figure 2: Stability regions for ordinary differential equations.

3 The Wave Equation

Consider the model problem $u_t + u_x = 0$ discretized with forward Euler in time. First order accurate spatial upwinding is given by

$$u_i^{n+1} = u_i^n - \frac{\Delta t}{\Delta x}(u_i^n - u_{i-1}^n) = (1-\lambda)u_i^n + \lambda u_{i-1}^n$$

with $\lambda = \Delta t/\Delta x$. BFECC can be written as a step forward in time

$$\hat{u}_i^{n+1} \quad = \quad (1-\lambda)u_i^n + \lambda u_{i-1}^n$$

followed by a step backward in time

$$\hat{u}_i^n \quad = \quad (1-\lambda)\hat{u}_i^{n+1} + \lambda\hat{u}_{i+1}^{n+1} = ((1-\lambda)^2 + \lambda^2)u_i^n + \lambda(1-\lambda)(u_{i-1}^n + u_{i+1}^n)$$

followed by a correction of the original data using the estimated error

$$\tilde{u}_i^n \quad = \quad u_i^n - (\hat{u}_i^n - u_i^n)/2 = u_i^n - \lambda(1-\lambda)(u_{i+1}^n - 2u_i^n + u_{i-1}^n)/2$$

followed by a step forward in time using this error corrected data

$$\begin{aligned} u_i^{n+1} &= (1-\lambda)\tilde{u}_i^n + \lambda\tilde{u}_{i-1}^n \\ &= \left(-\frac{1}{2}\lambda^2 + \frac{1}{2}\lambda^3\right)u_{i-2}^n + \left(\frac{1}{2}\lambda + 2\lambda^2 - \frac{3}{2}\lambda^3\right)u_{i-1}^n \\ &+ \left(1 - \frac{5}{2}\lambda^2 + \frac{3}{2}\lambda^3\right)u_i^n + \left(-\frac{1}{2}\lambda + \lambda^2 - \frac{1}{2}\lambda^3\right)u_{i+1}^n. \end{aligned}$$

Taylor expanding u_{i-2}^n, u_{i-1}^n and u_{i+1}^n, and using $u_t + u_x = 0$ leads to

$$\begin{aligned} u_i^{n+1} &= u_i^n - \Delta x(u_i^n)_x\lambda + \frac{\Delta x^2}{2}(u_i^n)_{xx}\lambda^2 - \frac{\Delta x^3}{6}(u_i^n)_{xxx}(-3\lambda^2 + \lambda + 3\lambda^3) \\ &\quad + \sum_{i=4}^{\infty}\xi_{\mathrm{bf}}(i, \Delta x, \Delta t) \\ &= u_i^n + \Delta t(u_i^n)_t + \frac{\Delta t^2}{2}(u_i^n)_{tt} + \frac{\Delta t^3}{6}(u_i^n)_{ttt}\left(-\frac{3}{\lambda} + \frac{1}{\lambda^2} + 3\right) \\ &\quad + \sum_{i=4}^{\infty}\xi_{\mathrm{bf}}(i, \Delta x, \Delta t) \end{aligned}$$

where $\xi_{\mathrm{bf}}(i, \Delta x, \Delta t) = O(\Delta x^{i-2}\Delta t^2 + \Delta x^{i-3}\Delta t^3 + (i \bmod 2)\Delta x^{i-1}\Delta t)$. Thus, the final terms can be simplified to $\sum_{i=4}^{\infty}\xi_{\mathrm{bf}}(i, \Delta x, \Delta t) = O(\Delta x^2\Delta t^2 + \Delta x\Delta t^3 + \Delta x^4\Delta t)$.

The MacCormack method instead uses the error estimate to correct the already advected data resulting in

$$\begin{aligned} u_i^{n+1} &= \hat{u}_i^{n+1} - (\hat{u}_i^n - u_i^n)/2 \\ &= \left(\frac{1}{2}\lambda + \frac{1}{2}\lambda^2\right) u_{i-1}^n + (1-\lambda^2)u_i^n + \left(-\frac{1}{2}\lambda + \frac{1}{2}\lambda^2\right) u_{i+1}^n. \end{aligned}$$

Taylor expansions and $u_t + u_x = 0$ can be used to rewrite this as

$$\begin{aligned} u_i^{n+1} &= u_i^n - \Delta x (u_i^n)_x \lambda + \frac{\Delta x^2}{2}(u_i^n)_{xx}\lambda^2 - \frac{\Delta x^3}{6}(u_i^n)_{xxx}\lambda + \sum_{i=4}^{\infty} \xi_m(i, \Delta x, \Delta t) \\ &= u_i^n + \Delta t (u_i^n)_t + \frac{\Delta t^2}{2}(u_i^n)_{tt} + \frac{\Delta t^3}{6}(u_i^n)_{ttt}\frac{1}{\lambda^2} + O(\Delta x^2 \Delta t^2 + \Delta x^4 \Delta t) \end{aligned}$$

since $\xi_m(i, \Delta x, \Delta t) = O(\Delta x^{i-2}\Delta t^2)$ when i even and $\xi_m(i, \Delta x, \Delta t) = O(\Delta x^{i-1}\Delta t)$ when i is odd.

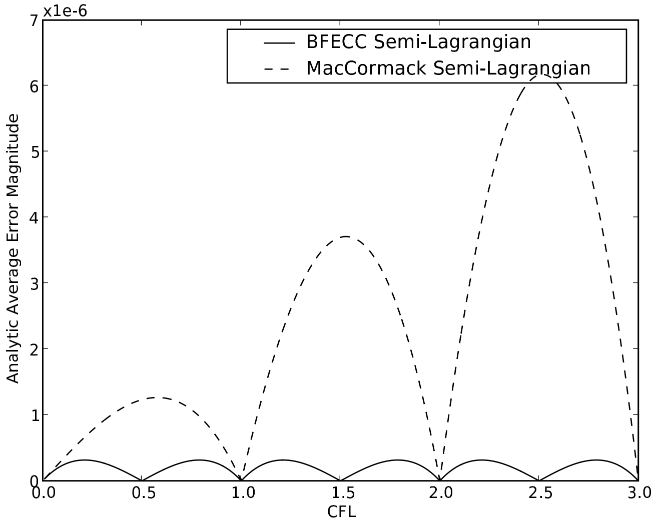
For a fixed CFL number (i.e. fixed λ) with $\Delta t = \lambda \Delta x$, the final summations in both methods become $O(\Delta x^4)$ or identically $O(\Delta t^4)$. Note that the first three terms of these expressions agree with the exact solution, so the fourth term in each expression is the leading local truncation error. We write the truncation error as $E(\lambda)\Delta t^3 u_{ttt}/6 + O(\Delta t^4)$ where $E_{bf}(\lambda) = 1/\lambda^2 - 3/\lambda + 2$ for BFECC and $E_m(\lambda) = 1/\lambda^2 - 1$ for the MacCormack method. Note that these results were derived for the wave equation under the assumption that $\lambda \leq 1$. Since an unconditionally stable approach allows for larger λ's, a similar piecewise analysis can be carried out for $\lambda \in (1, 2]$, $\lambda \in (2, 3]$, etc. to obtain

$$E_{bf}(\lambda) = \begin{cases} 2 + \frac{1}{\lambda^2} - \frac{3}{\lambda} & \lambda \leq 1 \\ 2 - \frac{6}{\lambda^3} + \frac{13}{\lambda^2} - \frac{9}{\lambda} & 1 < \lambda \leq 2 \\ 2 - \frac{30}{\lambda^3} + \frac{37}{\lambda^2} - \frac{15}{\lambda} & 2 < \lambda \leq 3 \end{cases} \quad (1)$$

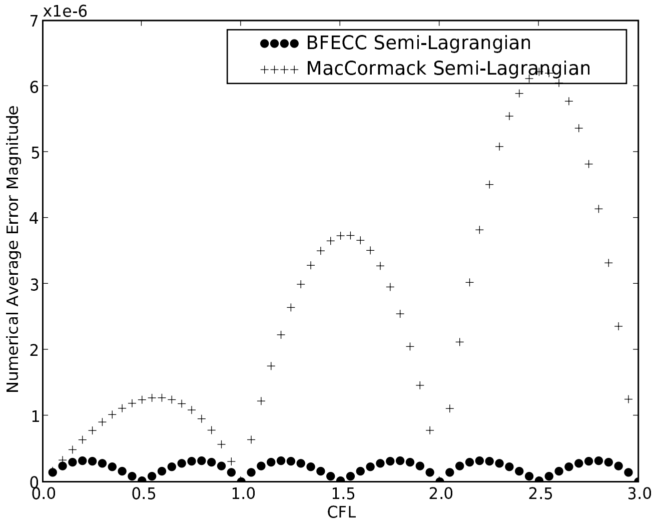
and

$$E_{mac}(\lambda) = \begin{cases} -1 + \frac{1}{\lambda^2} & \lambda \leq 1 \\ -1 - \frac{6}{\lambda^3} + \frac{7}{\lambda^2} & 1 < \lambda \leq 2 \\ -1 - \frac{30}{\lambda^3} + \frac{19}{\lambda^2} & 2 < \lambda \leq 3 \end{cases} \quad (2)$$

and so on. Using these expressions, we can compute the leading truncation error at each grid point x_i as $|\Delta t^3 u_{ttt}(x_i, t)E(\lambda)/6|$ and the average error over n grid points as $(1/n)\sum_{i=1}^{n} |\Delta t^3 u_{ttt}(x_i, t)E(\lambda)/6|$. Computing this for a single time step of $u_t + u_x = 0$ with $u(0, x) = \sin 4\pi x$ and $n = 400$ while varying the CFL number λ produces the graph depicted below.



Solving the same problem numerically for one time step and graphing the computed average error magnitude similarly yields the following graph.



Next, consider solving the wave equation with $u(0,x) = \sin 4\pi x$ and periodic boundary conditions to a final time of $t = .5$ (one period). Table 2 compares the average errors obtained using first order accurate upwinding,

	Upwinding		BFECC		MacCormack	
n	Error	Order	Error	Order	Error	Order
101	6.0e-2	–	1.3e-3	–	4.7e-3	–
201	3.1e-2	1.02	3.3e-4	2.09	1.2e-3	1.96
401	1.6e-2	1.01	8.2e-5	1.97	2.9e-4	2.01
801	7.8e-3	1.00	2.1e-5	1.99	7.2e-5	2.00

Table 2: Wave equation solved with upwind building blocks and CFL=.75.

	Semi-Lagrangian		BFECC		MacCormack	
n	Error	Order	Error	Order	Error	Order
101	2.6e-2	–	5.6e-4	–	5.3e-3	–
201	1.3e-2	.98	1.4e-4	2.09	1.3e-3	2.21
401	6.7e-3	1.02	3.5e-5	1.96	3.4e-4	2.05
801	3.3e-3	1.02	8.8e-6	1.99	8.4e-5	2.05

Table 3: Wave equation solved with semi-Lagrangian building blocks and CFL=1.75.

BFECC and the MacCormack method with a CFL of .75. Since the CFL is less than one, upwind building blocks are used for both BFECC and the MacCormack method. Spatial refinement shows the expected results for all three methods. Table 3 shows similar results for a CFL of 1.75 where semi-Lagrangian building blocks are used for BFECC and the MacCormack method. Note that the the first order accurate upwinding was replaced with the first order accurate semi-Lagrangian method as well.

4 New Extrema

Whether first order accurate upwind building blocks (and a CFL less than one) or simple semi-Lagrangian building blocks are used, the first step forward in time and the subsequent step backward in time produce monotone data. That is, $\hat{u}^n$ is bounded by $\hat{u}^{n+1}$ which is in turn bounded by u^n. However, nothing special can be stated about the manner in which the error is computed, or the result obtained when the error estimate is applied to either u^n in BFECC or $\hat{u}^{n+1}$ in the MacCormack scheme. In particular, this error correction step can lead to new extrema and possible instability. Figure 3 shows the oscillations obtained when using BFECC and the MacCormack scheme to advect a square wave for one period. A simple choice of limiter consists of limiting the final solution at each grid point to be bound by the values used in computing the first advection step from u^n to $\hat{u}^{n+1}$. For example, if the base of the semi-Lagrangian ray used in this first advection step interpolated data from the interval $[x_j, x_{j+1}]$, then we would postprocess the final result (from either BFECC or MacCormack) to be clamped between $\min(u_j^n, u_{j+1}^n)$ and $\max(u_j^n, u_{j+1}^n)$. This readily generalizes to multiple spatial dimensions, adaptive and non-Cartesian grids. Another commonly used limiter reverts to a first order accurate method when the higher order accurate method would overshoot (see e.g. [10]). For our method this requires no further computation as an out of bounds value is simply replaced with the value from the first semi-Lagrangian advection step, that is, no error correction is used. Figure 3 shows the results obtained by applying the clamping limiter to advection of the square wave initial data, and though it is not depicted we also ran the example using the reversion to first order accurate approach which yielded nearly identical results to clamping. We also compared clamping to reversion using a 2D flow example in Figure 9 in which we noticed no significant difference between the two methods. Despite this, we prefer the reversion approach as it yields better results when the semi-Lagrangian MacCormack scheme is applied to free surface flows where the reversible PDE assumption of our modified MacCormack scheme (and also BFECC) breaks down due to the linear extrapolation of velocities to outside the fluid region.

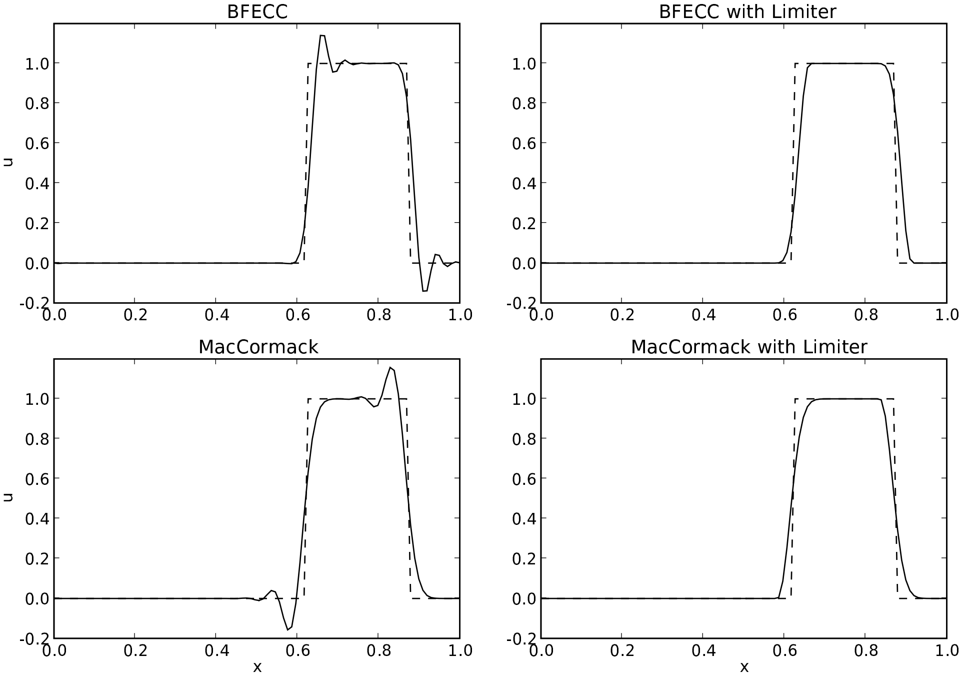


Figure 3: Both BFECC and the MacCormack method can create new extrema, as shown here where a square wave is advected for one period. Of course, these oscillations can be avoided with a simple limiting technique.

5 General Stability and Accuracy Results for Linear Equations in Multi Dimensions

Consider the equation

$$u_t + \mathbf{v} \cdot \nabla u = 0, \quad \mathbf{x} \in [0,1]^d, \tag{3}$$

with periodic boundary conditions, where $\mathbf{v} \in R^d$ is a constant vector. Assume a uniform partition of the domain so that $\mathbf{x_j} = (j_1, j_2, \cdots, j_d)h$, $j_l \in \{0, 1, \cdots, J\}$, and $h = 1/J$ is the mesh size. Let $u^n_{\mathbf{j}}$ be a numerical solution at $(\mathbf{x_j}, t_n)$. A linear scheme A (such as the first order accurate semi-Lagrangian scheme) is a linear operator $\hat{u}^{n+1} = A(u^n)$ such that

$$\hat{u}^{n+1}_{\mathbf{j}} = \sum_{\mathbf{i}} \alpha_{\mathbf{i}} u^n_{\mathbf{j}+\mathbf{i}},$$

where $\{\alpha_{\mathbf{i}}\}$ are constants depending on $\Delta t/h$, $\Delta t = t_{n+1} - t_n$. We can obtain stability and accuracy results by applying similar arguments as in [4]. Expanding $u^n_{\mathbf{j}}$ into the finite Fourier series

$$u^n_{\mathbf{j}} = \sum_{\mathbf{k}} c^n_{\mathbf{k}} e^{2\pi i \mathbf{k} \cdot \mathbf{x_j}},$$

where $\mathbf{k} = (k_1, k_2, \cdots, k_d)$, $k_l \in \{0, \pm 1, \cdots, \pm(J-1)\}$, we obtain a relation between two adjacent time steps $c^{n+1}_{\mathbf{k}} = \rho_A(\mathbf{k}) c^n_{\mathbf{k}}$, where $\rho_A(\mathbf{k})$ is called the Fourier symbol of A and $\max_{\mathbf{k}} |\rho_A(\mathbf{k})|$ is called the amplification factor of A. Let $A^R : \ \hat{u}^n = A^R(\hat{u}^{n+1})$ be the corresponding linear scheme that solves the time reversed equation of (3). Then the modified MacCormack scheme acting on A is

$$F : \ u^{n+1} = F(u^n) = \left(A + \frac{1}{2}(I - A^R A) \right) (u^n),$$

where I is the identity operator. The Fourier symbol of F is

$$\rho_F = \rho_A + \frac{1}{2}(1 - \rho_{A^R} \rho_A).$$

We have the following theorem.

Theorem 1. Suppose $\rho_{A^R} = \overline{\rho_A}$, then $|\rho_F| \le 1$ if $|\rho_A| \le 1$.

Most conventional first order accurate linear schemes satisfy the condition of Theorem 1, e.g., the upwind scheme, Lax-Friedrichs scheme and in particular the Courant-Isaacson-Rees scheme (CIR or first order accurate semi-Lagrangian scheme).

A linear scheme L is said to be accurate of order r if for any solution u of (3) having continuous derivatives up to order $r+1$,

$$u(\mathbf{x_j}, t_{n+1}) - L(u(\cdot, t_n))|_{\mathbf{x_j}} = O(h^{r+1}),$$

where $\Delta t/h$ is fixed during the mesh refinement. A convenient way to check the accuracy of scheme F is to compare the discrete Fourier symbol ρ_F to the corresponding Fourier symbol for the exact solution u by using the Lax's theorem [15]. Expanding u into the Fourier series

$$u(\mathbf{x}, t) = \sum_{\mathbf{k} \in Z^d} c_{\mathbf{k}}(t) e^{2\pi i \mathbf{k} \cdot \mathbf{x}}$$

and substituting it into (3) to obtain $\frac{dc_{\mathbf{k}}(t)}{dt} = P(i\mathbf{k})c_{\mathbf{k}}(t)$, we obtain

$$c_{\mathbf{k}}(t_n + \Delta t) = e^{\Delta t P(i\mathbf{k})} c_{\mathbf{k}}(t_n),$$

where P is a homogeneous polynomial of degree one with real coefficients and $e^{\Delta t P(i\mathbf{k})}$ is the Fourier symbol for the exact solution u. Therefore for an r-th order linear scheme A, following the Lax's theorem

$$\rho_A(\mathbf{k}) = e^{\Delta t P(i\mathbf{k})} + Q_{r+1}(i\mathbf{k}h) + O(|\mathbf{k}h|^{r+2}),$$

where Q_{r+1} is a homogeneous polynomial of degree $r+1$ with real coefficients. If r is an odd positive integer, then $Q_{r+1}(i\mathbf{k}h)$ is a real number. In addition, if $\rho_{A^R} = \overline{\rho_A}$, then

$$\begin{aligned}
\rho_F(\mathbf{k}) &= \rho_A(\mathbf{k}) + \tfrac{1}{2}[1 - \rho_{A^R}(\mathbf{k})\rho_A(\mathbf{k})] \\
&= \rho_A(\mathbf{k}) - \tfrac{1}{2}[e^{-\Delta t P(i\mathbf{k})} + e^{\Delta t P(i\mathbf{k})}]Q_{r+1}(i\mathbf{k}h) + O(|\mathbf{k}h|^{r+2}) \\
&= e^{\Delta t P(i\mathbf{k})} + Q_{r+1}(i\mathbf{k}h) - \tfrac{1}{2}[e^{-\Delta t P(i\mathbf{k})} + e^{\Delta t P(i\mathbf{k})}]Q_{r+1}(i\mathbf{k}h) + O(|\mathbf{k}h|^{r+2})] \\
&= e^{\Delta t P(i\mathbf{k})} + O(|\mathbf{k}h|^{r+2}).
\end{aligned}$$

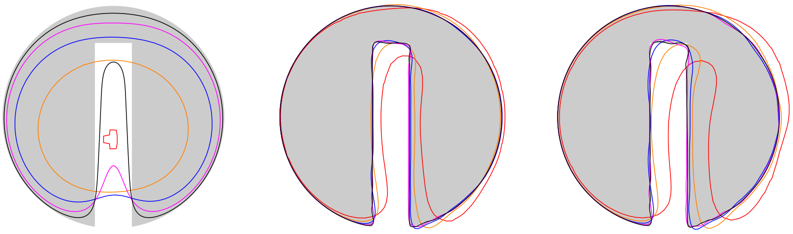
We have the following theorem.

Theorem 2. Suppose $\rho_{A^R} = \overline{\rho_A}$ and scheme A is accurate of order r for equation (3), where r is an odd positive integer, then scheme F is accurate of order $r+1$.

6 Zalesask's Disc

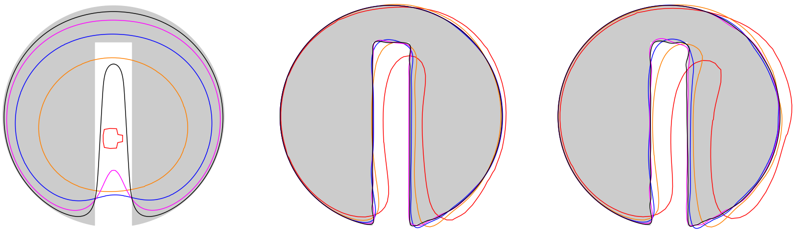
We use the the standard Zalesask's Disc test to illustrate the behavior of these methods for multidimensional advection. The initial data is a slotted circle centered at $(50, 75)$ with radius 15 and a 5 by 25 size slot. We show the results after advecting one rotation (to $t = 628$) in a rotational velocity field $V = \frac{\pi}{314}(50 - y, x - 50)$. Although BFECC was originally proposed for level set methods, this particular example is not meant to test these methods ability to treat interfaces. Instead, we are focusing on how these methods behave for multidimensional advection. Thus, we do not consider reinitialization [28] or the addition of particles [5]. In fact, [6] showed that the addition of accurately advected particles allows one to reduce high order accurate level set advection to first order accuracy without adverse results. Still, accurate multidimensional advection is quite useful for various other problems such as for incompressible flows. In the figures, we plot the zero isocontours for visualization purposes but stress that the errors are computed across the entire domain (i.e. all isocontours). In particular, we compute the order of accuracy by comparing the error e on the coarse grid to the error e' on the fine grid at coincident node locations as $\log_2 \frac{e(i,j)}{e'(2i-1,2j-1)}$ and average all of these values to obtain the values shown in the tables below.

Figure 4 compares first order accurate upwinding with the versions of BFECC and the MacCormack scheme obtained using first order accurate upwind building blocks. Figure 5 compares the simple first order accurate semi-Lagrangian method with the versions of BFECC and the MacCormack scheme obtained using simple first order accurate semi-Lagrangian building blocks. Note that the semi-Lagrangian method is fully multidimensional, so even though the CFL is .75 in both Figure 4 and Figure 5 the resulting numerical methods are all different (in contrast to one spatial dimension where semi-Lagrangian and upwind building blocks are identical for CFL's less than one). Finally, Figure 6 repeats the calculations from Figure 5 using a higher CFL of 1.75.



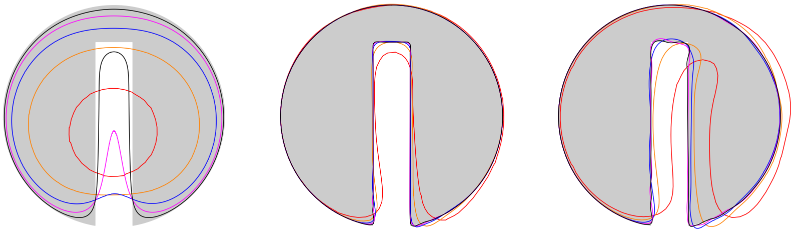
	Upwinding		BFECC		MacCormack	
Grid Size	Error	Order	Error	Order	Error	Order
🟥 101^2	1.8	–	8.9e-2	–	1.6e-1	–
🟧 201^2	9.6e-1	1.05	3.9e-2	1.79	6.3e-2	1.93
🟦 401^2	5.2e-1	1.04	1.8e-2	1.77	3.2e-2	1.59
🟪 801^2	2.8e-1	1.02	7.6e-3	1.98	1.3e-2	2.08
⬛ 1601^2	1.5e-1	1.01	3.1e-3	2.04	5.5e-3	2.08

Figure 4: Zalesask's disc rotation comparing simple upwinding with the versions of BFECC and the MacCormack scheme obtained using first order accurate upwind building blocks (CFL=.75).



	Semi-Lagrangian		BFECC		MacCormack	
Grid Size	Error	Order	Error	Order	Error	Order
🟥 101^2	1.8	–	9.9e-2	–	1.7e-1	–
🟧 201^2	1.0	1.04	4.1e-2	1.73	6.7e-2	1.95
🟦 401^2	5.4e-1	1.03	2.0e-2	1.67	3.3e-2	1.60
🟪 801^2	2.9e-1	1.02	8.4e-3	1.93	1.4e-2	2.06
⬛ 1601^2	1.5e-1	1.01	3.5e-3	2.00	5.9e-3	2.06

Figure 5: Zalesask's disc rotation comparing the simple first order accurate semi-Lagrangian method with the versions of BFECC and the MacCormack scheme obtained using simple first order accurate semi-Lagrangian blocks (CFL=.75).



	Semi-Lagrangian		BFECC		MacCormack	
Grid Size	Error	Order	Error	Order	Error	Order
🟥 101^2	2.0	–	5.4e-2	–	1.5e-1	–
🟧 201^2	1.1	1.02	2.4e-2	1.81	6.0e-2	1.82
🟦 401^2	5.6e-1	1.02	1.0e-2	1.84	3.0e-2	1.62
🟪 801^2	3.0e-1	1.01	4.0e-3	1.98	1.2e-2	1.99
⬛ 1601^2	1.5e-1	1.02	1.5e-3	2.07	5.2e-3	2.01

Figure 6: Zalesask's disc rotation comparing the simple first order accurate semi-Lagrangian method with the versions of BFECC and the MacCormack scheme obtained using simple first order accurate semi-Lagrangian blocks (CFL=1.75).

7 Incompressible Flow

Consider inviscid incompressible flow on a standard MAC grid. First the advection terms are updated to obtain an intermediate velocity, and then a Poisson equation is solved for the pressure which is subsequently used to make the intermediate velocity divergence free.

Consider a $[0, 3] \times [0, 1]$ domain divided into 300×100 MAC grid cells with a radius .125 circle centered at $(.5, .5)$. The horizontal velocity is set to one at the left and right boundaries, and the vertical velocity is set to zero at the top and bottom boundaries. Zero derivative Neumann boundary conditions are used for the pressure on all four boundaries. Figure 7 compares the first order accurate semi-Lagrangian method to BFECC and the MacCormack method using first order accurate semi-Lagrangian building blocks. The MacCormack method is shown both with and without the extrema clamping procedure discussed earlier. Note that each component of the velocity field is separately advected and limited. Note that BFECC and the MacCormack method both achieve similarly higher Reynolds number flows than the first order accurate semi-Lagrangian method. Whereas Figure 7 was obtained with a CFL of .75, comparable results are shown in Figure 8 with a higher CFL of 1.75. In Figure 9 we depict the results of applying the extrema clamping strategies discussed in Section 4.

Both BFECC and the MacCormack method do not perform well when information is mixed from disparate regions of the flow. This happens frequently near domain boundaries leading to poor error estimates in these regions. Thus, for both BFECC and the MacCormack method, we revert to the first order accurate semi-Lagrangian scheme at any grid point whose semi-Lagrangian characteristic could source from a solid wall or solid circle boundary (i.e. CFL $\times \max(\Delta x, \Delta y)$).

Now consider free surface flows which combine incompressible flow with level set advection for interface tracking. We use the basic free surface model described in [7] with the improved second order accurate free surface pressure boundary condition proposed in [8] and the improved fast marching method from [16]. In addition, we use first order accurate semi-Lagrangian level set advection together with second order accurate Runge-Kutta advection for the particles because [6] showed that the Particle Level Set Method with a second order accurate particle advection scheme obviates the need for high order level set advection.

As [7] does, we extrapolate a divergence free velocity field across the interface for the level set and incompressible advection, resulting in lower order values in those regions. The BFECC or semi-Lagrangian MacCormack error computation then depends on these less accurate values which creates new extrema which we could clamp. Unfortunately, clamping tends to pollute the solution with the incorrect error correction, thus when new

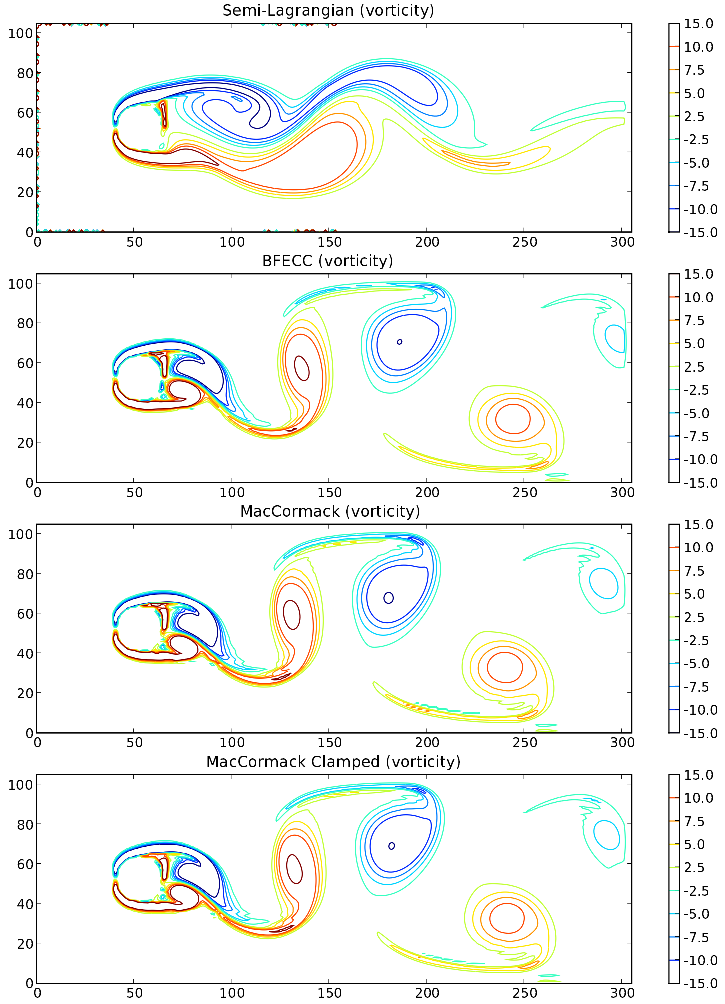


Figure 7: Inviscid incompressible flow comparing the first order accurate semi-Lagrangian method to BFECC and the MacCormack method using first order accurate semi-Lagrangian building blocks (CFL=.75).

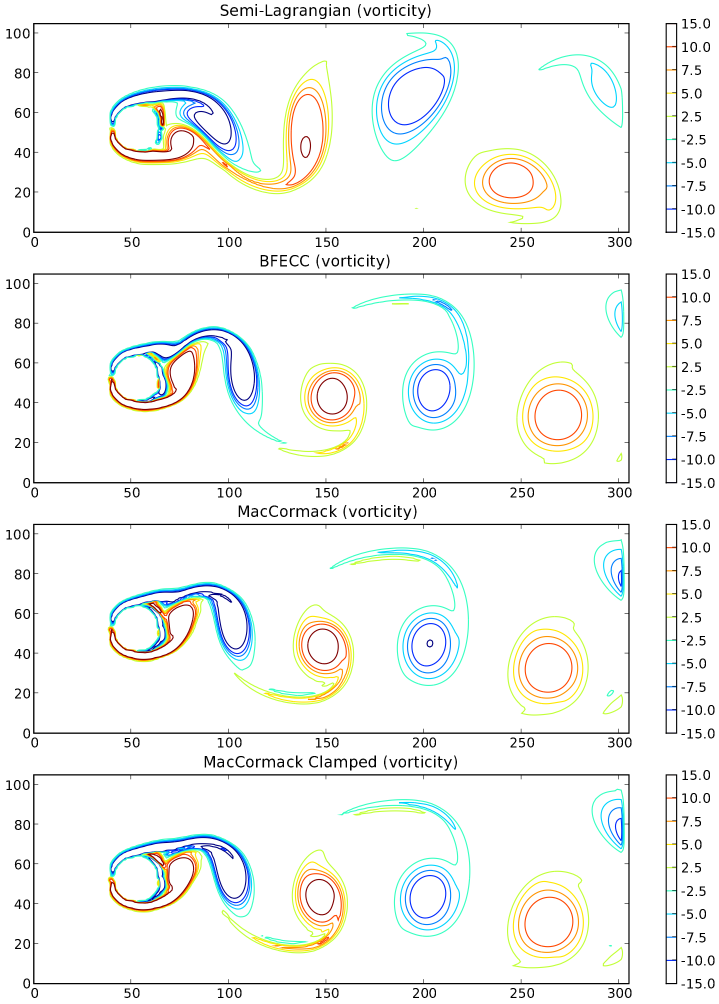


Figure 8: Inviscid incompressible flow comparing the first order accurate semi-Lagrangian method to BFECC and the MacCormack method using first order accurate semi-Lagrangian building blocks (CFL=1.75).

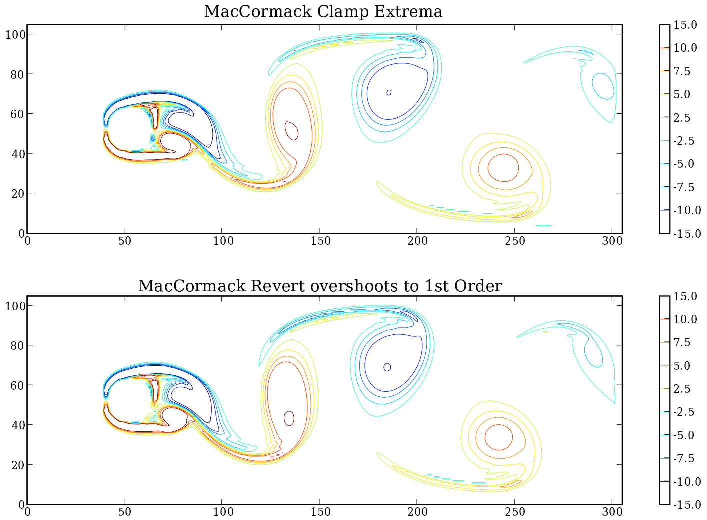
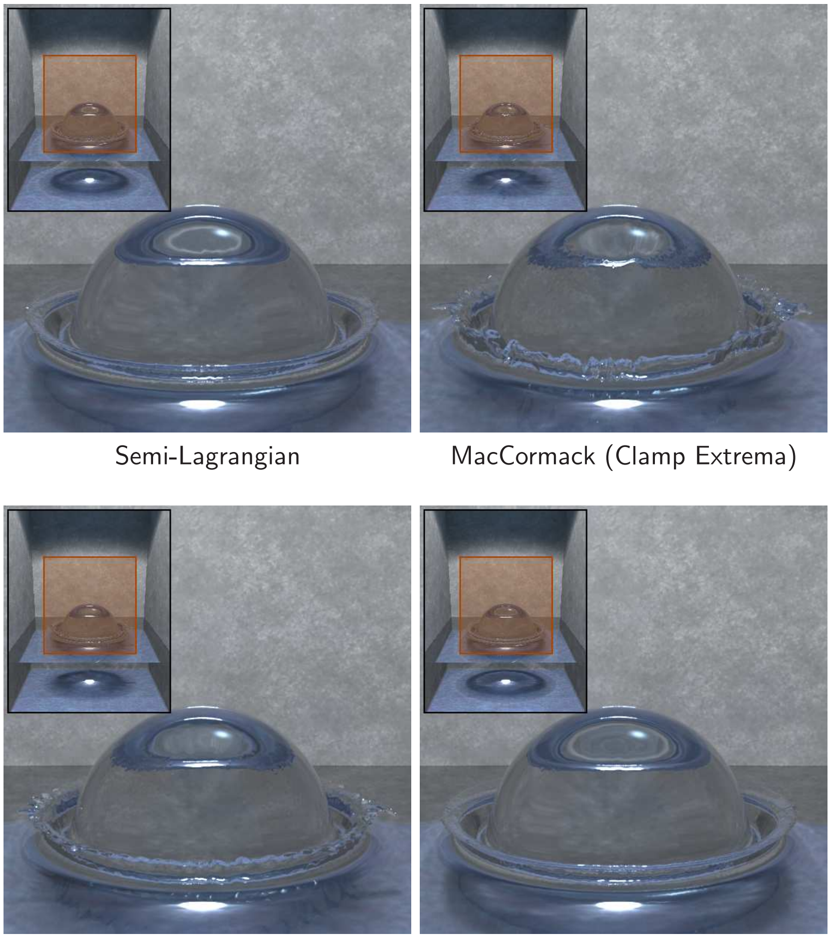


Figure 9: Inviscid incompressible flow comparing the semi-Lagrangian MacCormack method with clamping to the semi-Lagrangian MacCormack method with reversion to first order when new extrema are detected (CFL=.75).

extrema are detected we advocate reverting to first order accurate semi-Lagrangian advection instead.

Consider a spherical radius .5 drop located at $(.5, .75, .5)$ falling (with gravity vector $(0, -9.8, 0)$) into a pool at height .412134 contained in a walled domain of $[0, 1] \times [0, 1.5] \times [0, 1]$ as shown in Figure 10, 11 and 12. As with the flow past circle example, advection was reverted to the first order accurate semi-Lagrangian method in cells close to the domain boundaries. Without any limiter the simulation becomes unstable so we apply the following limiter variants: (1) clamping of new extrema, (2) clamping of new extrema but always using the first order accurate semi-Lagrangian method near the interface and (3) reverting to the first order accurate semi-Lagrangian method at cells where new extrema are found. (1) and (3) do well at producing higher Reynolds number flows while (2) mitigates most of the benefit of the second order accurate approach.

Finally, consider a radius .2 solid ball splashing into a pool of water in a $[0, 1.5] \times [0, 1] \times [0, 1]$ domain. The ball is kinematically moved from $(1.25, .55, .5)$ to $(.8, .1, .5)$ between time 0 and .075 using linear interpolation, and the fluid has the gravity vector $(0, -9.8, 0)$. Figure 13, 14 and 15 show a comparison between the first order accurate semi-Lagrangian method and the three limiter strategies described above. Again we note that without a limiter, the simulation is unstable. Also, as with the flow past circle example and the falling drop example, advection near the domain boundaries and the object boundaries is reverted to the first order accurate semi-Lagrangian method.



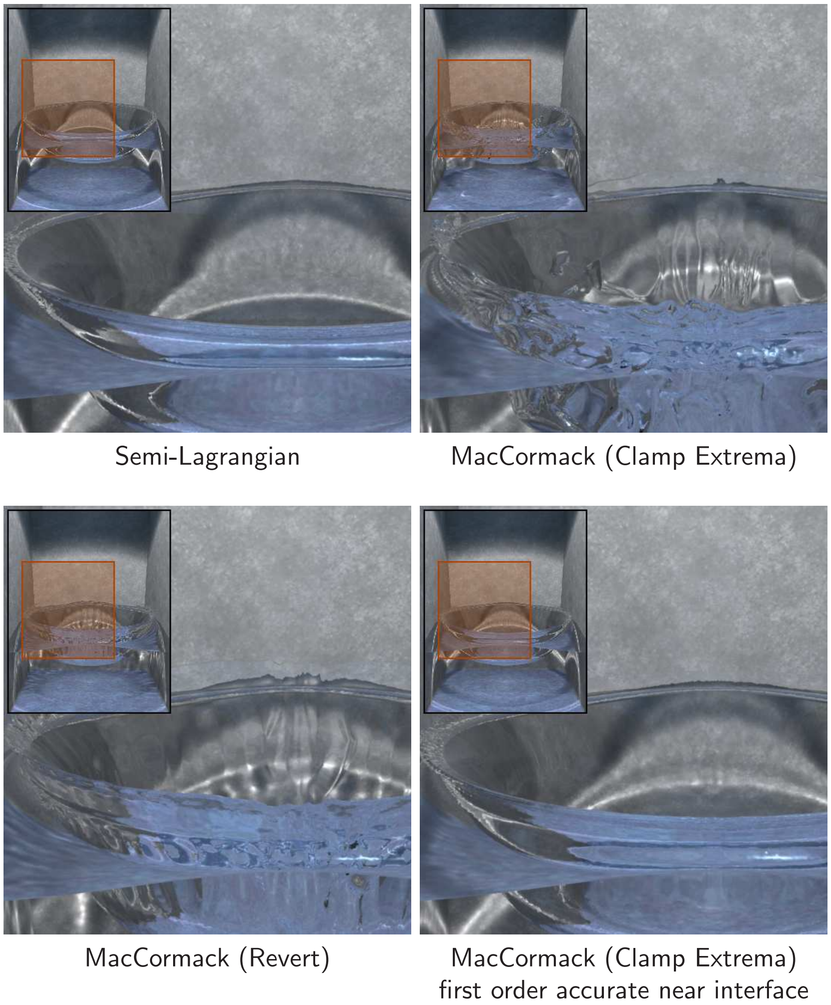
Semi-Lagrangian

MacCormack (Clamp Extrema)

MacCormack (Revert)

MacCormack (Clamp Extrema)
first order accurate near interface

Figure 10: Inviscid incompressible free surface flow of a drop falling into a pool. Notice how the MacCormack methods (top right, bottom left) produce additional detail. Using the first order accurate semi-Lagrangian advection near the interface (bottom right) removes much of the advantage of the second order accurate method (bottom right). ($200 \times 300 \times 200$ grid, $t = 6/24$, CFL=1.75).



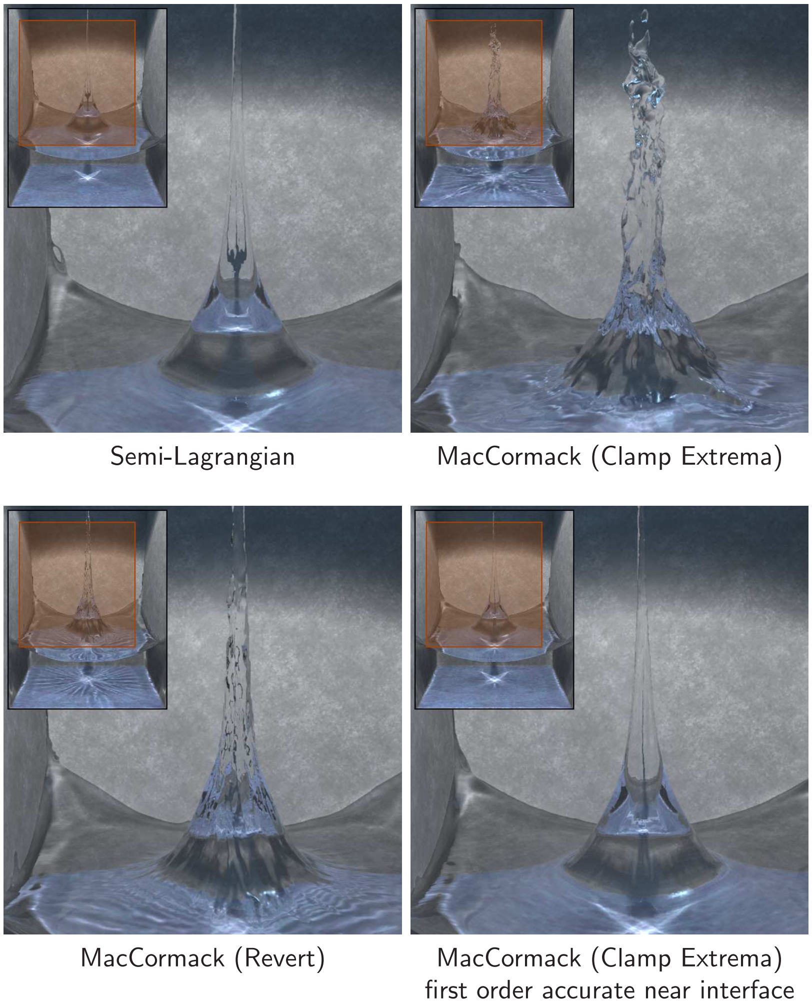
Semi-Lagrangian

MacCormack (Clamp Extrema)

MacCormack (Revert)

MacCormack (Clamp Extrema) first order accurate near interface

Figure 11: Inviscid incompressible free surface flow of a drop falling into a pool ($200 \times 300 \times 200$ grid, $t = 10/24$, CFL=1.75).



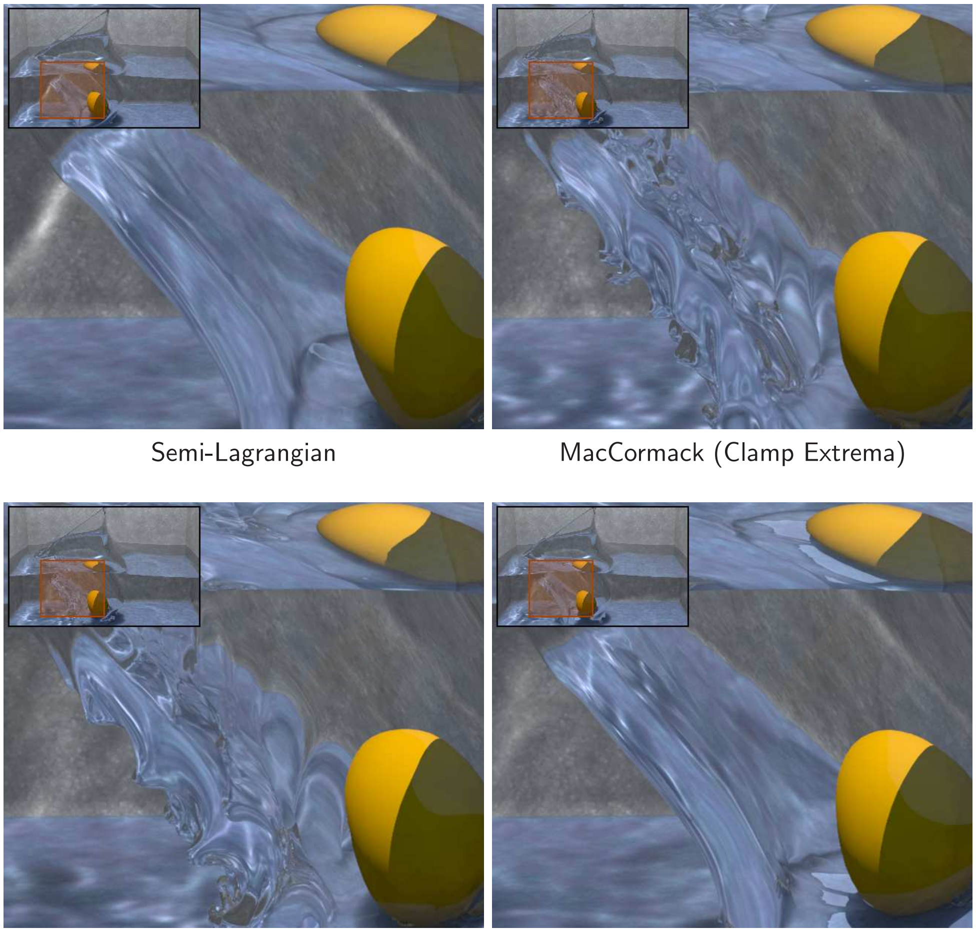
Semi-Lagrangian

MacCormack (Clamp Extrema)

MacCormack (Revert)

MacCormack (Clamp Extrema)
first order accurate near interface

Figure 12: Inviscid incompressible free surface flow of a drop falling into a pool ($200 \times 300 \times 200$ grid, $t = 18/24$, CFL=1.75).



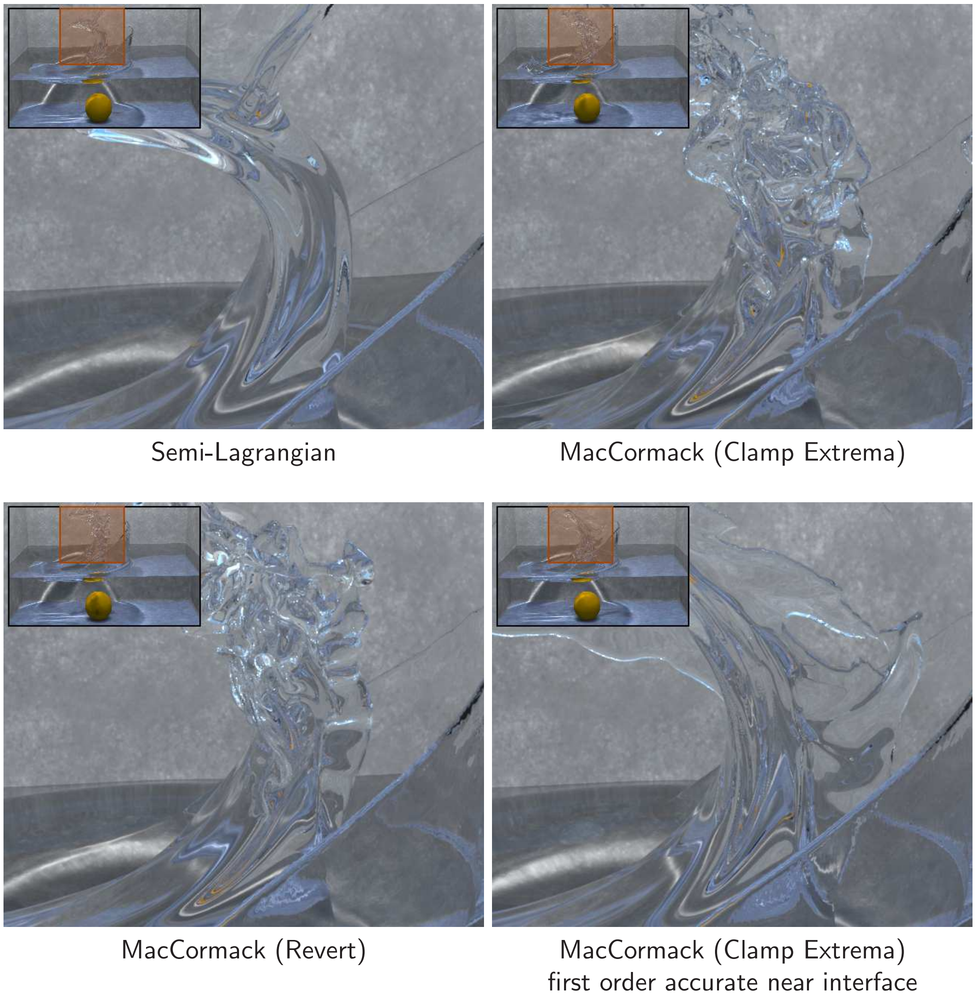
Semi-Lagrangian

MacCormack (Clamp Extrema)

MacCormack (Revert)

MacCormack (Clamp Extrema)
first order accurate near interface

Figure 13: Inviscid incompressible free surface flow of a solid ball thrown into a pool. Note that the clamp and reversion MacCormack variants (upper right, lower left) capture the vortex sheeting due to the ball's penetration that is missed when using the semi-Lagrangian method or the clamped MacCormack method that uses the first order accurate semi-Lagrangian method near the interface (upper left, lower right). (450 × 300 × 300 grid, CFL=1.75, t=4/24)



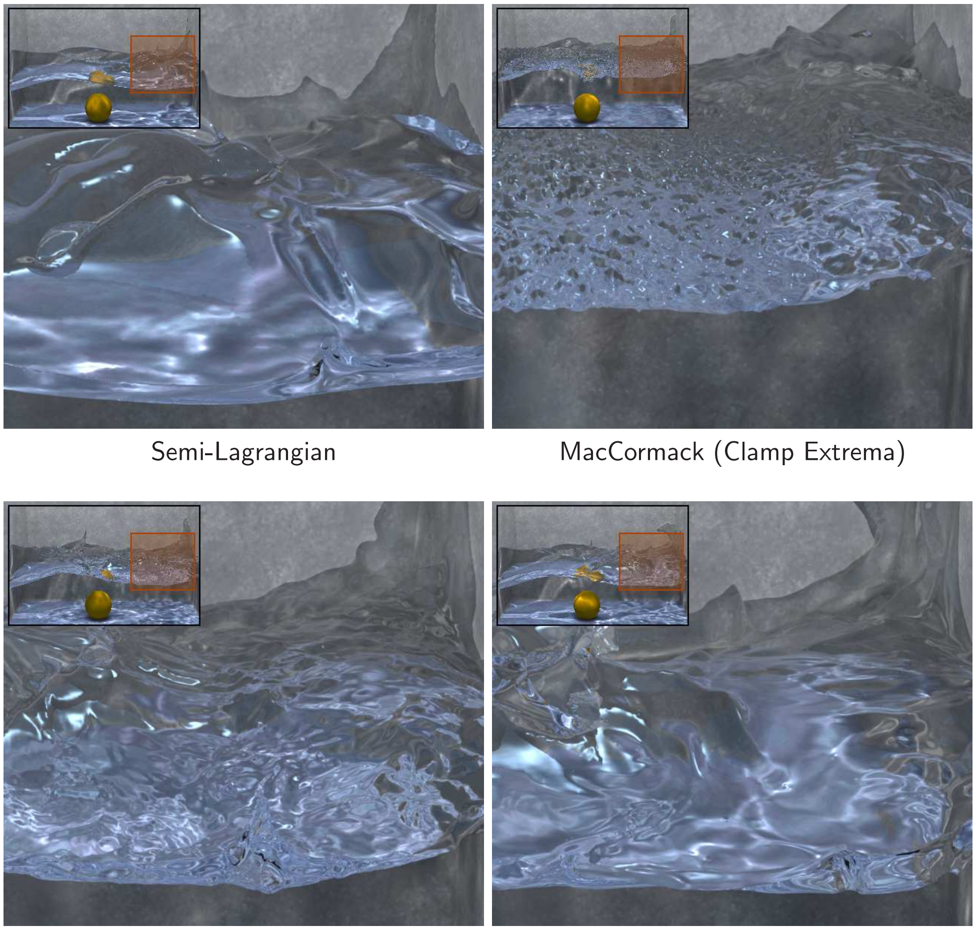
Semi-Lagrangian

MacCormack (Clamp Extrema)

MacCormack (Revert)

MacCormack (Clamp Extrema)
first order accurate near interface

Figure 14: Inviscid incompressible free surface flow of a solid ball thrown into a pool. ($450 \times 300 \times 300$ grid, CFL=1.75, t=10/24)



Semi-Lagrangian

MacCormack (Clamp Extrema)

MacCormack (Revert)

MacCormack (Clamp Extrema)
first order accurate near interface

Figure 15: Inviscid incompressible free surface flow of a solid ball thrown into a pool. ($450 \times 300 \times 300$ grid, CFL=1.75, t=40/24)

8 Conclusion

Motivated by the BFECC method, we rewrote a version of the MacCormack method into a similar form facilitating its extension to unconditional stability while still preserving second order accuracy in space in time. The MacCormack method only requires two first order accurate advection steps whereas BFECC is 50% more expensive requiring three advection steps. Although the semi-Lagrangian versions of BFECC and the MacCormack scheme are not fully monotone, we proposed a simple limiter that preserves monotonicity. We illustrated the expected behavior of the method on both simple and more complex problems, noting that difficulties could arise if the semi-Lagrangian rays mix information from disparate regions such as from both solid wall boundaries and fluid regions. In those instances, it is simple to revert to the first order accurate semi-Lagrangian method. We also saw that using an increasing number of simple semi-Lagrangian rays can improve accuracy. In particular one ray yields the first order accurate semi-Lagrangian method, two rays yield the second order accurate MacCormack method and three rays yield the second order accurate BFECC method, which has a lower truncation error magnitude than the MacCormack method in some problems. Thus an obvious question is whether one can do better than the BFECC method with three rays, perhaps even obtaining third order accuracy.

Acknowledgments

A. Selle and R. Fedkiw were supported in part by an ONR YIP award and a PECASE award (ONR N00014-01-1-0620), a Packard Foundation Fellowship, a Sloan Research Fellowship, ONR N0014-06-1-0393, ONR N00014-03-1-0071, ONR N00014-02-1-0720, ONR N00014-05-1-0479 (for a SUN computing cluster), ARO DAAD19-03-1-0331, NSF IIS-0326388, NSF ITR-0205671, NSF ITR-0121288, NSF ACI-0323866 and NIH U54-GM072970. Y. Liu was supported by NSF grant DMS-0511815.

References

[1] J.D. Anderson. *Computational Fluid Dynamics: The Basics With Applications.* McGraw-Hill, 1995. New York, NY.

[2] R. Courant, E. Issacson, and M. Rees. On the solution of nonlinear hyperbolic differential equations by finite differences. *Comm. Pure and Applied Math*, 5:243–255, 1952.

[3] T. Dupont and Y. Liu. Back and forth error compensation and correction methods for removing errors induced by uneven gradients of the level set function. *J. Comput. Phys.*, 190/1:311–324, 2003.

[4] T. Dupont and Y. Liu. Back and forth error compensation and correction methods for semi-Lagrangian schemes with application to level set interface computations. *Math. Comp.*, In Press., 2006.

[5] D. Enright, R. Fedkiw, J. Ferziger, and I. Mitchell. A hybrid particle level set method for improved interface capturing. *J. Comput. Phys.*, 183:83–116, 2002.

[6] D. Enright, F. Losasso, and R. Fedkiw. A fast and accurate semi-Lagrangian particle level set method. *Computers and Structures*, 83:479–490, 2005.

[7] D. Enright, S. Marschner, and R. Fedkiw. Animation and rendering of complex water surfaces. *ACM Trans. Graph. (SIGGRAPH Proc.)*, 21(3):736–744, 2002.

[8] D. Enright, D. Nguyen, F. Gibou, and R. Fedkiw. Using the particle level set method and a second order accurate pressure boundary condition for free surface flows. In *Proc. 4th ASME-JSME Joint Fluids Eng. Conf.*, number FEDSM2003–45144. ASME, 2003.

[9] R. Fedkiw, J. Stam, and H. Jensen. Visual simulation of smoke. In *Proc. of ACM SIGGRAPH 2001*, pages 15–22, 2001.

[10] M. Herrmann and G. Blanquart. Flux corrected finite volume scheme for preserving scalar boundedness in reacting large-eddy simulations. *AIAA J.*, 44(12):2879–2886, 2006.

[11] G. Irving, E. Guendelman, F. Losasso, and R. Fedkiw. Efficient simulation of large bodies of water by coupling two and three dimensional techniques. *ACM Trans. Graph. (SIGGRAPH Proc.)*, 25(3):805–811, 2006.

[12] G.-S. Jiang and D. Peng. Weighted ENO schemes for Hamilton-Jacobi equations. *SIAM J. Sci. Comput.*, 21:2126–2143, 2000.

[13] B.-M. Kim, Y. Liu, I. Llamas, and J. Rossignac. Using BFECC for fluid simulation. In *Eurographics Workshop on Natural Phenomena 2005*, 2005.

[14] B.-M. Kim, Y. Liu, I. Llamas, and J. Rossignac. Advections with significantly reduced dissipation and diffusion. *IEEE Trans. on Vis. and Comput. Graph.*, In Press., 2006.

[15] P.D. Lax. On the stability of difference approximations to solutions of hyperbolic equations with variable coefficients. *Comm. Pure Appl. Math.*, 14:497–520, 1961.

[16] F. Losasso, R. Fedkiw, and S. Osher. Spatially adaptive techniques for level set methods and incompressible flow. *Computers and Fluids*, 35:995–1010, 2006.

[17] F. Losasso, F. Gibou, and R. Fedkiw. Simulating water and smoke with an octree data structure. *ACM Trans. Graph. (SIGGRAPH Proc.)*, 23:457–462, 2004.

[18] R. MacCormack. The effect of viscosity in hypervelocity impact cratering. In *AIAA Hypervelocity Impact Conference*, 1969. AIAA paper 69-354.

[19] C. Min and F. Gibou. A second order accurate projection method for the incompressible Navier-Stokes equation on non-graded adaptive grids. *J. Comput. Phys.*, 219:912–929, 2006.

[20] S. Osher and R. Fedkiw. *Level Set Methods and Dynamic Implicit Surfaces*. Springer-Verlag, 2002. New York, NY.

[21] S. Osher and J. Sethian. Fronts propagating with curvature-dependent speed: Algorithms based on Hamilton-Jacobi formulations. *J. Comput. Phys.*, 79:12–49, 1988.

[22] A. Selle, N. Rasmussen, and R. Fedkiw. A vortex particle method for smoke, water and explosions. *ACM Trans. Graph. (SIGGRAPH Proc.)*, 24(3):910–914, 2005.

[23] C.-W. Shu and S. Osher. Efficient implementation of essentially non-oscillatory shock capturing schemes. *J. Comput. Phys.*, 77:439–471, 1988.

[24] J. Stam. Stable fluids. In *Proc. of SIGGRAPH 99*, pages 121–128, 1999.

[25] A. Staniforth and J. Cote. Semi-Lagrangian integration schemes for atmospheric models: A review. *Monthly Weather Review*, 119:2206–2223, 1991.

[26] J. Steinhoff and D. Underhill. Modification of the Euler equations for "vorticity confinement": Application to the computation of interacting vortex rings. *Phys. of Fluids*, 6(8):2738–2744, 1994.

[27] J. Strain. Tree methods for moving interfaces. *J. Comput. Phys.*, 151:616–648, 1999.

[28] M. Sussman, P. Smereka, and S. Osher. A level set approach for computing solutions to incompressible two-phase flow. *J. Comput. Phys.*, 114:146–159, 1994.

[29] R.F. Warming and R.M. Beam. Upwind second-order difference schemes and applications in aerodynamic flows. *AIAA J.*, 14(9):1241–1249, 1976.