A Comparison of a Family of Eulerian and Semi-Lagrangian Finite Element Methods for the Advection-Diffusion Equation

Francis X. Giraldo [*†]
Naval Research Laboratory, Marine Meteorology Division, Monterey, CA 93943

Beny Neta
Naval Postgraduate School, Department of Mathematics, Monterey, CA 93943

Abstract

Eulerian and semi-Lagrangian finite element methods are analyzed for stability and accuracy for the one-dimensional advection-diffusion equation. The methods studied are a class of schemes called theta algorithms that yield the explicit ($\theta = 0$), semi-implicit ($\theta = \frac{1}{2}$), and implicit ($\theta = 1$) methods. The stability analysis shows that the semi-Lagrangian method is unconditionally stable for all values of θ while the Eulerian method is only unconditionally stable for $\frac{1}{2} \leq \theta \leq 1$. The accuracy analysis shows that the semi-Lagrangian and Eulerian methods are second order accurate in both space and time only for $\theta = \frac{1}{2}$. This analysis shows that the best methods are the $\theta = \frac{1}{2}$ which are the semi-implicit methods. In essence this paper compares a semi-implicit Eulerian method with a semi-implicit semi-Lagrangian method, analytically and numerically. The analysis shows that the semi-implicit semi-Lagrangian method exhibits better amplitude, dispersion and group velocity errors than the semi-implicit Eulerian method thereby achieving better results. Numerical experiments are performed on the two-dimensional advection and advection-diffusion equations having known analytic solutions. The numerical results corroborate the analysis by demonstrating that the semi-Lagrangian method is superior to the Eulerian method while using time steps two to four times greater. This property makes them more attractive than Eulerian methods particularly for integrating atmospheric and ocean equations because long time histories are sought for such problems.

1 Introduction

Eulerian and semi-Lagrangian finite element models for the advection and advection-diffusion equations are presented. The best methods are found to be the semi-implicit methods ($\theta = \frac{1}{2}$). Therefore this paper essentially compares a semi-implicit Eulerian method with a semi-implicit semi-Lagrangian method. The majority of the numerical models developed in the past have used Eulerian methods. In numerical weather prediction, attention has recently shifted towards semi-Lagrangian methods because they are not bound by the CFL restrictions of Eulerian methods and as a result can use time steps four times greater. In short, they offer increased efficiency without a decrease in accuracy. The analysis performed in this paper shows that the maximum allowable Courant number should not exceed four. For values larger than four, dispersion errors can adversely affect the accuracy of the semi-Lagrangian solution.

Semi-Lagrangian methods and other related methods such as Characteristic Galerkin and Eulerian-Lagrangian methods have been studied using the advection equation in two-dimensions [14] and the advection-diffusion equation in one [11] and two-dimensions [13]. In [11] a class of schemes similar to

[*]This research was conducted while the author was an NRC associate at the Naval Postgraduate School.
[†]Author to whom all correspondence should be addressed.

semi-Lagrangian methods are studied for amplification errors but only for Lagrange interpolation. In this paper, we analyze a family of two-time-level semi-Lagrangian methods for amplification, dispersion and group velocity errors. In addition, this paper compares semi-Lagrangian methods in two-dimensions using Lagrange, Hermite, and spline interpolation.

Semi-Lagrangian methods have been implemented successfully for numerical weather prediction models by Bates and McDonald [1], Robert [16], and Staniforth and Temperton [17]. However, most of these methods have used finite difference spatial discretizations but finite elements have many advantages over finite difference methods including optimality and generalization to unstructured grids. In section 2, the finite element discretization of the two-dimensional advection-diffusion equation using Eulerian and semi-Lagrangian methods is introduced. Bilinear rectangular finite elements are used for the spatial discretization. For a comparison of various triangular and rectangular finite element discretizations see Neta and Williams [10] who recommend either bilinear rectangular or isosceles triangular elements. Section 3 contains the stability and accuracy analyses of these methods. Section 3 also discusses the properties of the operators discretized by the finite element method for the Eulerian and semi-Lagrangian methods, and how the structure of the resulting matrices affects the choice of matrix solvers. Section 4 presents the numerical experiments performed on the two-dimensional advection and advection-diffusion equations to validate the methods and corroborate the one-dimensional analysis. Finally, section 5 contains the concluding remarks and a discussion of the direction of future work.

2 Discretization

The differential form of the $2D$ advection-diffusion equation is

$$\frac{\partial \varphi}{\partial t} + \vec{u} \cdot \nabla \varphi = K \nabla^2 \varphi \tag{1}$$

where φ is some conservation variable, $\vec{u}$ is the velocity vector, and K is the diffusion coefficient.

2.1 Eulerian

In Eulerian schemes the evolution of the system is monitored from fixed positions in space and as a consequence, are the easiest methods to implement as all variable properties are computed at fixed grid points in the domain. Discretizing this equation by the finite element method, we arrive at the following elemental equations

$$M\dot{\varphi} + (A + D)\varphi = R$$

where M is the mass matrix, A the advection, D the diffusion, and R the boundary terms which are given by

$$M_{ij} = \int_\Omega \psi_i \psi_j d\Omega,$$

$$A_{ij} = \int_\Omega \sum_{\ell=1}^{4} \left(u_\ell \psi_\ell \psi_i \frac{\partial \psi_j}{\partial x} + v_\ell \psi_\ell \psi_i \frac{\partial \psi_j}{\partial y} \right) d\Omega,$$

$$D_{ij} = K \int_\Omega \left(\frac{\partial \psi_i}{\partial x}\frac{\partial \psi_j}{\partial x} + \frac{\partial \psi_i}{\partial y}\frac{\partial \psi_j}{\partial y} \right) d\Omega,$$

$$R_i = K \int_{\partial\Omega} \psi_i (\nabla \varphi \cdot \vec{n}) dS,$$

where ψ are the bilinear shape functions and $\vec{n}$ is the outward pointing normal vector of the boundaries. Discretizing this relation in time gives the theta algorithm

$$[M + \Delta t \theta (A + D)]\varphi^{n+1} = [M - \Delta t (1 - \theta)(A + D)]\varphi^n + \Delta t (\theta R^{n+1} + (1 - \theta) R^n) \tag{2}$$

where $\theta = 0, \frac{1}{2}, 1$ gives the explicit, semi-implicit, and implicit methods, respectively [7]. For other possible time discretizations see [18].

2.2 Semi-Lagrangian

Semi-Lagrangian methods belong to the general class of upwinding methods. These methods incorporate characteristic information into the numerical scheme. The Lagrangian form of Equation (1) is

$$\frac{d\varphi}{dt} = K\nabla^2\varphi \tag{3}$$

$$\frac{d\vec{x}}{dt} = \vec{u}(\vec{x},t) \tag{4}$$

where $\frac{d}{dt}$ denotes the total derivative. Discretizing this equation by the two-time level theta semi-Lagrangian method yields

$$\varphi^{n+1} - \Delta t\theta K\nabla^2\varphi^{n+1} = \varphi_d^n + \Delta t(1-\theta)K\nabla^2\varphi_d^n \tag{5}$$

where $\varphi^{n+1} = \varphi(\vec{x}, t+\Delta t)$ and $\varphi_d^n = \varphi(\vec{x}-\vec{\alpha}, t)$ are the solutions at the arrival and departure (d) points, respectively and (integrating (4) by the mid-point rule)

$$\vec{\alpha} = \Delta t\,\vec{u}\left(\vec{x} - \frac{\vec{\alpha}}{2}, t + \frac{\Delta t}{2}\right) \tag{6}$$

defines a recursive relation for the semi-Lagrangian departure points. Discretizing this relation in space by the finite element method, we get

$$[M + \Delta t\theta D]\,\varphi^{n+1} = [M - \Delta t(1-\theta)D]\,\varphi_d^n + \Delta t(\theta R^{n+1} + (1-\theta)R_d^n) \tag{7}$$

where the matrices are defined as in the Eulerian case.

3 Stability and Accuracy Analysis

In order to determine the linear stability of the methods, let us turn our attention toward the one-dimensional advection-diffusion equation

$$\varphi_t + u\varphi_x - K\varphi_{xx} = 0. \tag{8}$$

3.1 Eulerian

The discretization of the theta algorithm gives

$$\left(\frac{1}{\Delta t}\left[\frac{1}{6},\frac{2}{3},\frac{1}{6}\right] - \frac{\theta u}{2\Delta x}[1,0,-1] - \frac{\theta K}{\Delta x^2}[1,-2,1]\right)\cdot[\varphi_{j-1}^{n+1},\varphi_j^{n+1},\varphi_{j+1}^{n+1}] =$$
$$\left(\frac{1}{\Delta t}\left[\frac{1}{6},\frac{2}{3},\frac{1}{6}\right] + \frac{(1-\theta)u}{2\Delta x}[1,0,-1] + \frac{(1-\theta)K}{\Delta x^2}[1,-2,1]\right)\cdot[\varphi_{j-1}^{n},\varphi_j^{n},\varphi_{j+1}^{n}] \tag{9}$$

where linear finite elements are used. This discretization is obtained by constructing the global equations from the element equations and is now defined at the grid points. Note that linear finite elements are very similar to second order centered finite differences; the only difference being the consistent mass matrix. If the mass matrix were to be lumped, then we would arrive at an identical second order centered finite difference discretization. Let us introduce the Fourier mode

$$\varphi_j^{n+1} = G^{n+1}\exp^{ij\phi} \tag{10}$$

where G is the amplification factor, j is the grid point, $i = \sqrt{-1}$, $\phi = k\Delta x$ is the phase angle, k is the wave number, and n denotes the time level. Substituting (10) into (9) and letting

$$\sigma = \frac{\Delta t u}{\Delta x}(\text{Courant number}) \quad \text{and} \quad \mu = \frac{\Delta t K}{\Delta x^2}\ (\text{diffusion coefficient})$$

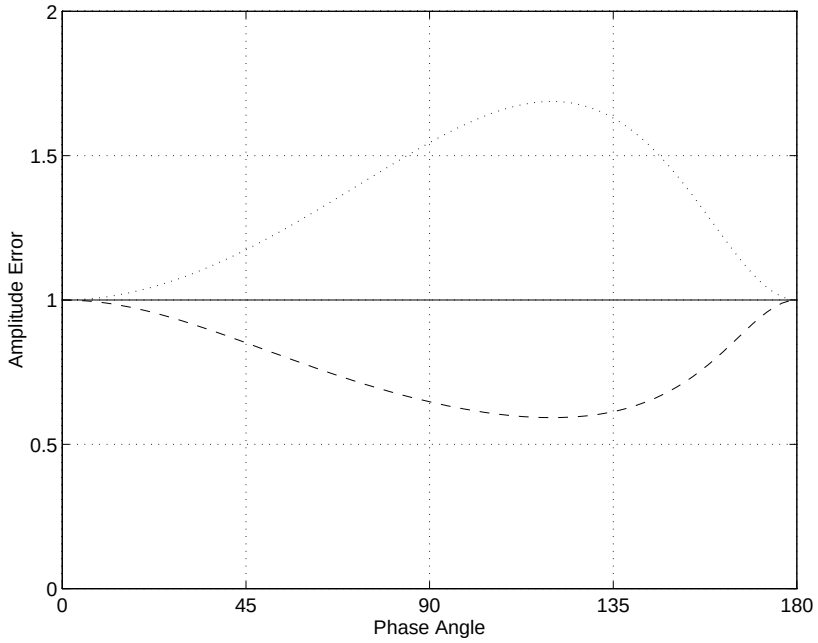

Figure 1: The Amplification Error for the Eulerian method for $K = 0$ and $\sigma = \frac{\pi}{4}$. The three values for θ illustrated are 0 (dotted), $\frac{1}{2}$ (solid) and 1 (dashed).

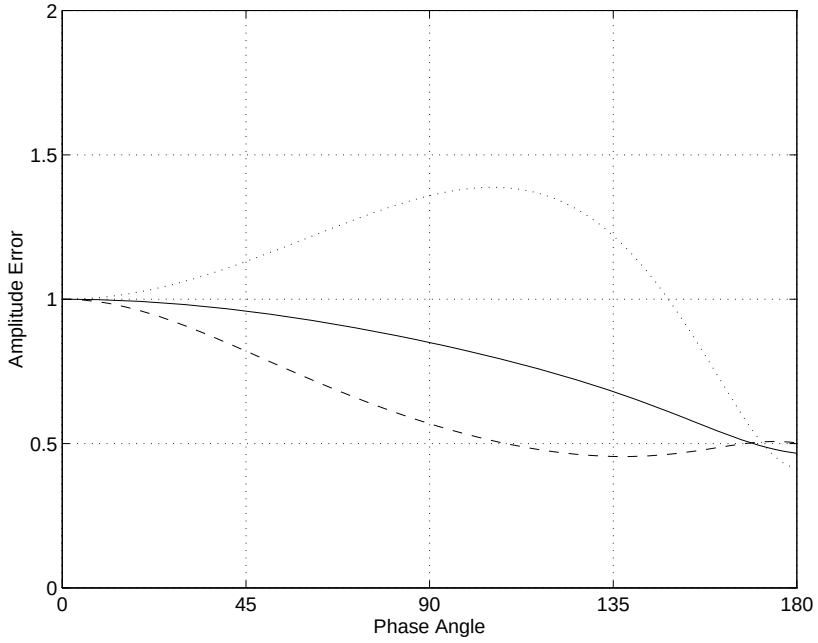

Figure 2: The Amplification Error for the Eulerian method for $K = 7 \times 10^4$ and $\sigma = \frac{\pi}{4}$. The three values for θ illustrated are 0 (dotted), $\frac{1}{2}$ (solid) and 1 (dashed).

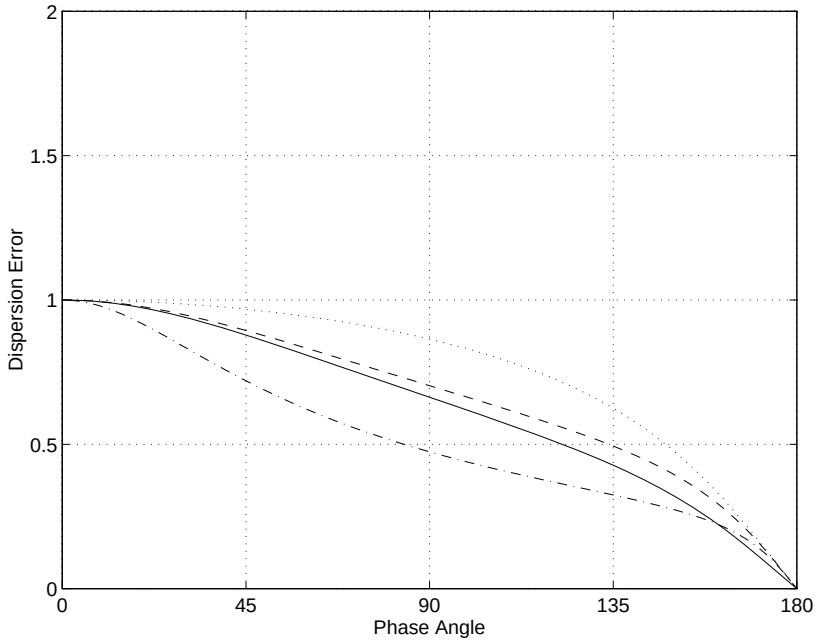

Figure 3: The Dispersion Error for the Eulerian methods for $K = 7 \times 10^4$. The Courant numbers for $\theta = \frac{1}{2}$ illustrated are $\sigma = \frac{\pi}{4}$ (dotted), $\frac{\pi}{2}$ (dashed), and π (dashed-dotted). The Courant number for $\theta = 1$ illustrated is $\sigma = \frac{\pi}{4}$ (solid).

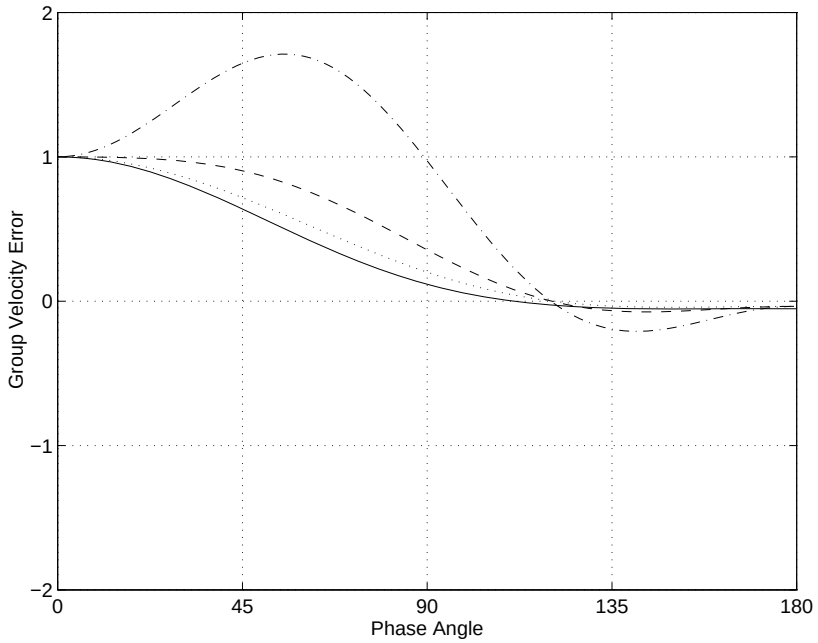

Figure 4: The Group Velocity Error for the Eulerian methods for $K = 7 \times 10^4$. The Courant numbers for $\theta = \frac{1}{2}$ illustrated are $\sigma = \frac{\pi}{4}$ (dotted), $\frac{\pi}{2}$ (dashed), and π (dashed-dotted). The Courant number for $\theta = 1$ illustrated is $\sigma = \frac{\pi}{4}$ (solid).

we get the amplification factor

$$G = \left[\frac{\frac{2}{3} + \frac{1}{3}\cos\phi + 2\mu(1-\theta)(\cos\phi - 1) - i\sigma(1-\theta)\sin\phi}{\frac{2}{3} + \frac{1}{3}\cos\phi - 2\mu\theta(\cos\phi - 1) + i\sigma\theta\sin\phi} \right] \tag{11}$$

and the amplification error

$$\epsilon_G = |G| \exp^{-\mu\phi^2}. \tag{12}$$

For advection ($K = \mu = 0$) the method is unconditionally unstable for $0 \leq \theta < \frac{1}{2}$ and unconditionally stable for $\frac{1}{2} \leq \theta \leq 1$. For advection-diffusion the method is conditionally stable for $0 \leq \theta < \frac{1}{2}$ and unconditionally stable for $\frac{1}{2} \leq \theta \leq 1$. Writing $G = |G| \exp^{-i\Phi}$ we get the dispersion relation

$$\Phi = \arctan\left[\frac{a}{b}\right] \tag{13}$$

where

$$a = \sigma\left(\frac{2}{3} + \frac{1}{3}\cos\phi\right)\sin\phi,$$

$$b = \left[\frac{2}{3} + \frac{1}{3}\cos\phi + 2\mu(1-\theta)(\cos\phi - 1)\right]\left[\frac{2}{3} + \frac{1}{3}\cos\phi - 2\mu\theta(\cos\phi - 1)\right] - \sigma^2\theta(1-\theta)\sin^2\phi$$

and the dispersion error is given by

$$\epsilon_\Phi = \frac{\Phi}{\sigma\phi}. \tag{14}$$

The group velocity is defined as the derivative of the frequency ω with respect to the wave number k and is given by

$$\frac{d\omega}{dk} = \frac{\Delta x}{\Delta t}\frac{d\tan\Phi}{d\phi}\frac{1}{\sec^2\Phi} \tag{15}$$

where $\Phi = \omega\Delta t$. The derivative of the tan function is given by

$$\frac{d\tan\Phi}{d\phi} = \frac{a'\cdot b - b'\cdot a}{b^2} \tag{16}$$

where

$$a' = \sigma\left[\left(\frac{2}{3} + \frac{1}{3}\cos\phi\right)\cos\phi - \frac{1}{3}\sin^2\phi\right]$$

and

$$\begin{aligned} b' = &\left(-\frac{1}{3}\sin\phi - 2\mu(1-\theta)\sin\phi\right)\left(\frac{2}{3} + \frac{1}{3}\cos\phi - 2\mu\theta(\cos\phi - 1)\right) \\ &+ \left(-\frac{1}{3}\sin\phi + 2\mu\theta\sin\phi\right)\left(\frac{2}{3} + \frac{1}{3}\cos\phi + 2\mu(1-\theta)\right) \\ &-2\sigma^2\theta(1-\theta)\sin\phi\cos\phi \end{aligned}$$

which yields the group velocity error

$$\epsilon_{gv} = \frac{1}{\sigma}\frac{d\tan\Phi}{d\phi}\frac{1}{\sec^2\Phi}. \tag{17}$$

Figures 1 and 2 show the behavior of the amplitude errors for advection and advection-diffusion for the three Eulerian methods. Figures 3 and 4 show the dispersion and group velocity errors for advection-diffusion for the semi-implicit and fully-implicit Eulerian methods - the curves for the explicit method are not included because this method has stability problems and hence is a poor choice. By virtue of figures 1 and 3 we can see that for advection the $\theta = \frac{1}{2}$ algorithm yields the best solution; one which has no amplitude (damping) error. In addition, for advection-diffusion

the $\theta = \frac{1}{2}$ algorithm also yields the best algorithm, albeit it has an associated damping error due to the diffusion introduced by the governing equation. However, the dispersion and group velocity errors associated with the $\theta = \frac{1}{2}$ algorithm are quite large as $\phi \to \pi$ (short waves) and increase with increasing σ. These figures tell us that for advection (figures 1, 3 and 4), the semi-implicit Eulerian method ($\theta = \frac{1}{2}$) suffers from dispersion and group velocity errors because these waves are not damped. On the other hand, it performs better for advection-diffusion (see figures 2, 3, and 4) because the short dispersive waves are damped by the diffusion terms in the governing equation. Figure 4 shows that the group velocity approaches the actual advection speed u for the long waves ($\phi = 0$) but goes to zero for the short waves ($\phi \to \pi$). As σ increases, the group velocity error becomes greater than one for some phase angles and negative for other phase angles meaning that the information is propagating faster than the theoretical wave speed or in the wrong direction. As an example, for the Courant number $\sigma = \pi$ we get

$$\frac{d\omega}{dk} > u \quad \text{for} \quad 0 < \phi < \frac{2\pi}{3}$$

and

$$\frac{d\omega}{dk} \leq 0 \quad \text{for} \quad \phi > \frac{2\pi}{3}.$$

Note that dispersion and group velocity plots for the fully-implicit method are given only for $\sigma = \frac{\pi}{4}$. By comparing these curves with those for the semi-implicit method it is evident that this method is much more dispersive than the semi-implicit method and consequently not as accurate.

For the accuracy analysis we expand via a Taylor series to fourth order in space and third order in time about the point $(j\Delta x, t)$ and get

$$\varphi_j^n = \varphi$$

$$\varphi_{j+1}^n = \varphi + \Delta x \varphi_x + \frac{\Delta x^2}{2}\varphi_{xx} + \frac{\Delta x^3}{6}\varphi_{xxx} + O(\Delta x)^4,$$

$$\varphi_{j-1}^n = \varphi - \Delta x \varphi_x + \frac{\Delta x^2}{2}\varphi_{xx} - \frac{\Delta x^3}{6}\varphi_{xxx} + O(\Delta x)^4$$

$$\varphi_j^{n+1} = \varphi + \Delta t \varphi_t + \frac{\Delta t^2}{2}\varphi_{tt} + O(\Delta t)^3,$$

$$\begin{aligned}\varphi_{j+1}^{n+1} = \varphi &+ \Delta t \varphi_t + \frac{\Delta t^2}{2}\varphi_{tt} + \Delta x \varphi_x + \frac{\Delta x^2}{2}\varphi_{xx} + \frac{\Delta x^3}{6}\varphi_{xxx} \\ &+ \Delta t \Delta x \varphi_{tx} + \frac{\Delta t \Delta x^2}{2}\varphi_{txx} + \frac{\Delta t \Delta x^3}{6}\varphi_{txxx} \\ &+ \frac{\Delta t^2 \Delta x}{2}\varphi_{ttx} + \frac{\Delta t^2 \Delta x^2}{4}\varphi_{ttxx} + \frac{\Delta t^2 \Delta x^3}{12}\varphi_{ttxxx} + O(\Delta t^3, \Delta x^4),\end{aligned}$$

$$\begin{aligned}\varphi_{j-1}^{n+1} = \varphi &+ \Delta t \varphi_t + \frac{\Delta t^2}{2}\varphi_{tt} - \Delta x \varphi_x + \frac{\Delta x^2}{2}\varphi_{xx} - \frac{\Delta x^3}{6}\varphi_{xxx} \\ &- \Delta t \Delta x \varphi_{tx} + \frac{\Delta t \Delta x^2}{2}\varphi_{txx} - \frac{\Delta t \Delta x^3}{6}\varphi_{txxx} \\ &- \frac{\Delta t^2 \Delta x}{2}\varphi_{ttx} + \frac{\Delta t^2 \Delta x^2}{4}\varphi_{ttxx} - \frac{\Delta t^2 \Delta x^3}{12}\varphi_{ttxxx} + O(\Delta t^3, \Delta x^4),\end{aligned}$$

which when substituted into (9) gives the local discretization error

$$L(x,t) = \varphi_t + u\varphi_x - K\varphi_{xx} + \left[\frac{\Delta t}{2}\varphi_{tt} + \Delta t\, u\, \theta\, \varphi_{tx} - \Delta t\, \theta\, K\, \varphi_{txx}\right] + O(\Delta t^2, \Delta x^2).$$

Differentiating the original equation

$$K\varphi_{xx} = \varphi_t + u\varphi_x$$

with respect to t we obtain

$$K\varphi_{txx} = \varphi_{tt} + u\varphi_{tx}$$

which can be substituted into the local discretization error to yield

$$L(x,t) = \varphi_t + u\varphi_x - K\varphi_{xx} + \left[\Delta t\varphi_{tt}\left(\frac{1}{2} - \theta\right)\right] + O(\Delta t^2, \Delta x^2). \tag{18}$$

This relation shows that the method is second order accurate in both time and space for $\theta = \frac{1}{2}$. For all other values, the method is only first order accurate in time. For $K = 0$ (advection) we can write

$$\varphi_{tt} = u^2\varphi_{xx}$$

and we then get

$$L(x,t) = \varphi_t + u\varphi_x + \left[\Delta t\, u^2\, \varphi_{xx}\left(\frac{1}{2} - \theta\right)\right] + O(\Delta t^2, \Delta x^2) \tag{19}$$

which yields a diffusion-like term for $\theta > \frac{1}{2}$ which explains why implicit methods are unconditionally stable. However, this diffusion term dissipates the solution and hence diminishes its accuracy. For $\theta < \frac{1}{2}$ the scheme adds this quantity thereby explaining the reason for the instabilities encountered by this second order explicit method. In section 4 (numerical experiments) the explicit method used in the numerical experiments is a first order upwind method.

3.2 Semi-Lagrangian

The Lagrangian form of the one-dimensional advection-diffusion equation is

$$\frac{d\varphi}{dt} - K\varphi_{xx} = 0 \tag{20}$$

$$\frac{dx}{dt} = u(x,t) \tag{21}$$

and the discretized form is

$$\begin{aligned}\left(\frac{1}{\Delta t}\left[\frac{1}{6},\frac{2}{3},\frac{1}{6}\right] - \frac{\theta K}{\Delta x^2}[1,-2,1]\right)\cdot[\varphi_{j-1}^{n+1},\varphi_j^{n+1},\varphi_{j+1}^{n+1}] = \\ \left(\frac{1}{\Delta t}\left[\frac{1}{6},\frac{2}{3},\frac{1}{6}\right] + \frac{(1-\theta)K}{\Delta x^2}[1,-2,1]\right)\cdot[\tilde{\varphi}_{d-1}^{n},\tilde{\varphi}_d^{n},\tilde{\varphi}_{d+1}^{n}]\end{aligned} \tag{22}$$

$$\alpha = \Delta t\ u\left(x - \frac{\alpha}{2}, t + \frac{\Delta t}{2}\right)$$

where d is the departure point and $\tilde{\varphi}_d^n$ is the interpolation of φ_d^n using grid point values. Introducing the Fourier modes we obtain the amplification factor

$$G = [f_d]\left[\frac{\frac{2}{3} + \frac{1}{3}\cos\phi + 2\mu\theta(\cos\phi - 1)}{\frac{2}{3} + \frac{1}{3}\cos\phi - 2\mu(1-\theta)(\cos\phi - 1)}\right] \tag{23}$$

where

$$f_d = \frac{\tilde{\varphi}_d^n}{\varphi_j^n} \tag{24}$$

which is a generalized stability criteria and is valid for any type of approximation used for $\tilde{\varphi}_d^n$. The amplification error is again defined by (12). Assuming no interpolation is required because we know the value at the departure point, then the interpolation function is

$$\tilde{\varphi}_d^n = \varphi^n(j\Delta x - \alpha)$$

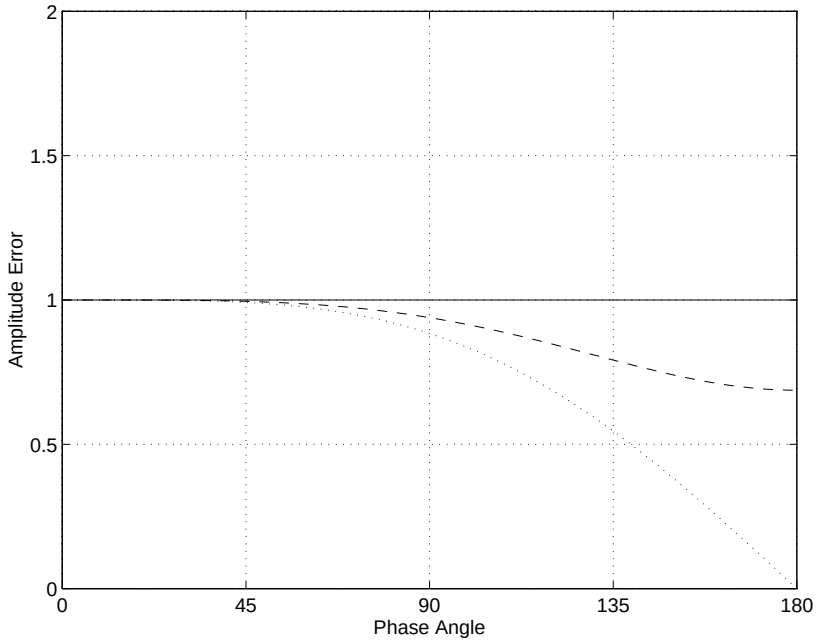

Figure 5: The Amplification Error for the semi-Lagrangian method using cubic spline interpolation for $K = 0$. The four values for $\hat{\alpha}$ illustrated are 0.25 and 0.75 (dashed), 0.50 (dotted), and 1 (solid). This figure is valid for all values of p and θ.

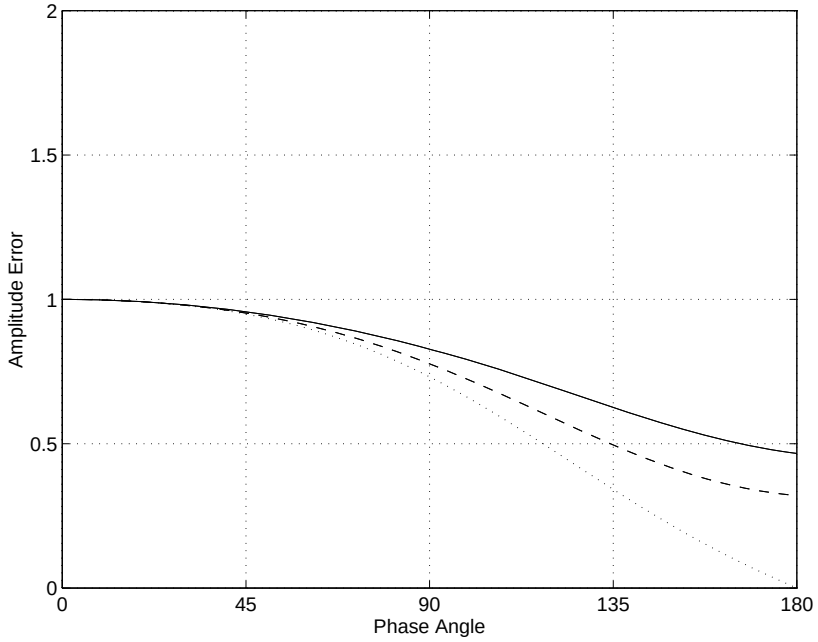

Figure 6: The Amplification Error for the semi-Lagrangian method using cubic spline interpolation for $K = 7 \times 10^4$. The four values for $\hat{\alpha}$ illustrated are 0.25 and 0.75 (dashed), 0.50 (dotted), and 1 (solid). This figure is valid for all values of p and θ.

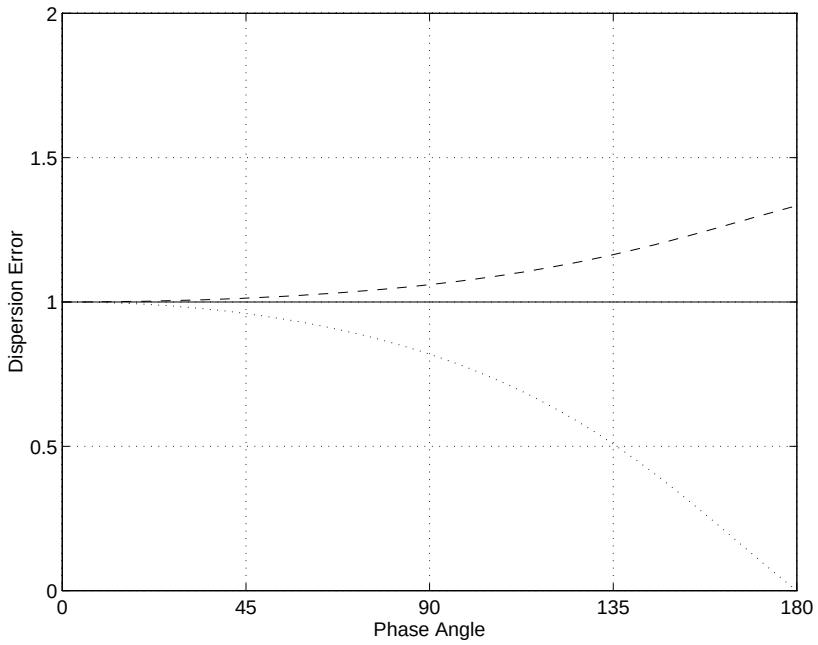

Figure 7: The Dispersion Error for the semi-Lagrangian method using cubic spline interpolation. The four values for $\hat{\alpha}$ illustrated are 0.25 (dotted), 0.75 (dashed), 0.5 and 1 (solid) where $p = 0$. Thus $\sigma = 0.25, 0.50, 0.75$ and 1. This figure is valid for all values of K and θ.

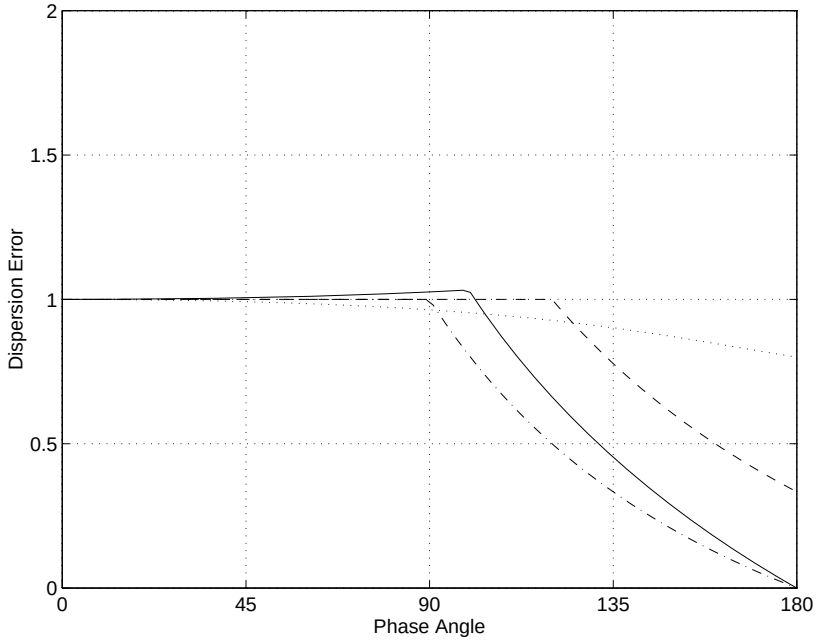

Figure 8: The Dispersion Error for the semi-Lagrangian method using cubic spline interpolation. The four values for $\hat{\alpha}$ illustrated are 0.25 (dotted), 0.50 (dashed), 0.75 (solid) and 1 (dashed-dotted) where $p = 1$. Thus $\sigma = 1.25, 1.50, 1.75$ and 2. This figure is valid for all values of K and θ.

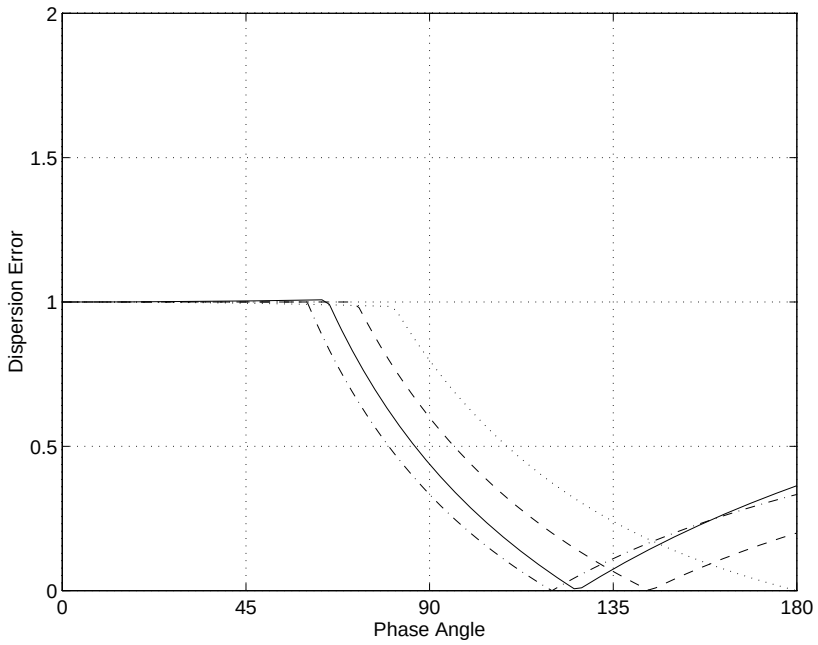

Figure 9: The Dispersion Error for the semi-Lagrangian method using cubic spline interpolation. The four values for $\hat{\alpha}$ illustrated are 0.25 (dotted), 0.50 (dashed), 0.75 (solid) and 1 (dashed-dotted) where $p = 2$. Thus $\sigma = 2.25, 2.50, 2.75$ and 3. This figure is valid for all values of K and θ.

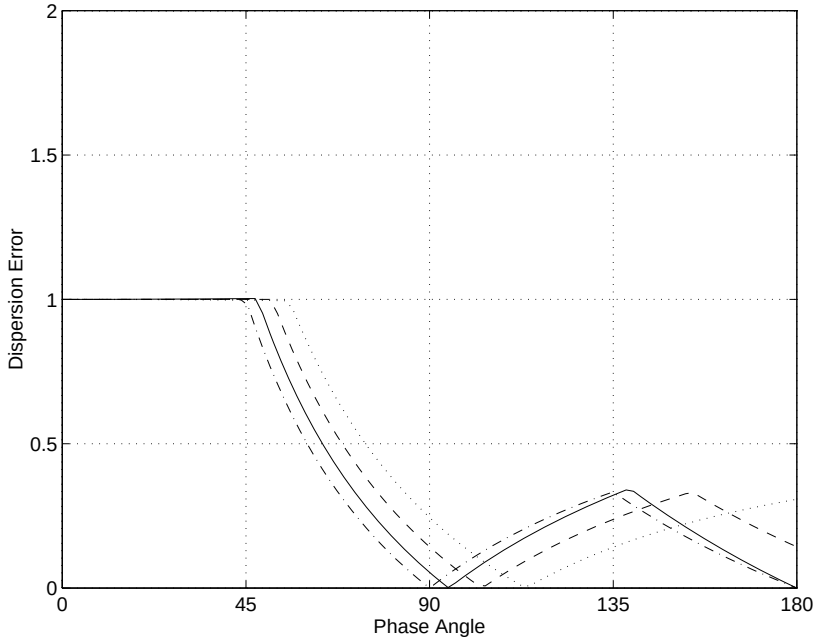

Figure 10: The Dispersion Error for the semi-Lagrangian method using cubic spline interpolation. The four values for $\hat{\alpha}$ illustrated are 0.25 (dotted), 0.50 (dashed), 0.75 (solid) and 1 (dashed-dotted) where $p = 3$. Thus $\sigma = 3.25, 3.50, 3.75$ and 4. This figure is valid for all values of K and θ.

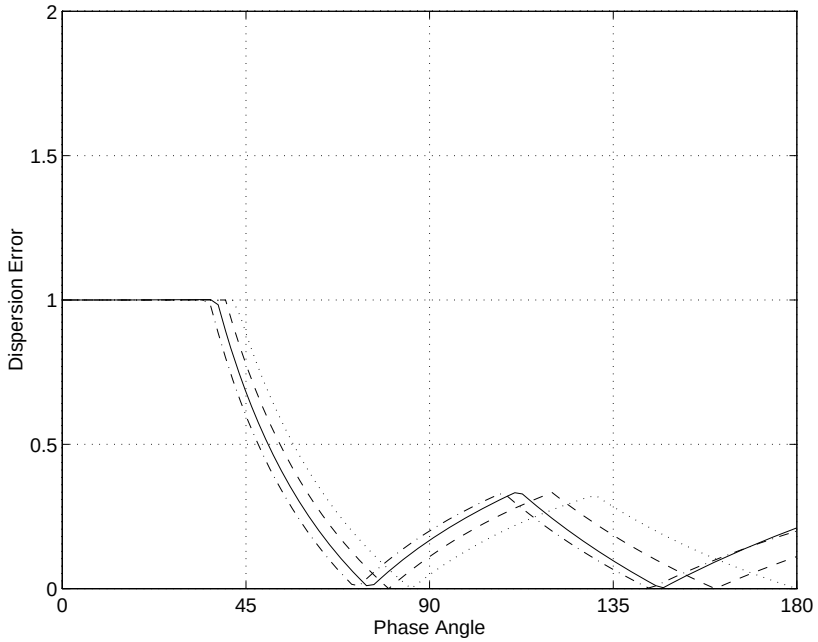

Figure 11: The Dispersion Error for the semi-Lagrangian method using cubic spline interpolation. The four values for $\hat{\alpha}$ illustrated are 0.25 (dotted), 0.50 (dashed), 0.75 (solid) and 1 (dashed-dotted) where $p = 4$. Thus $\sigma = 4.25, 4.50, 4.75$ and 5. This figure is valid for all values of K and θ.

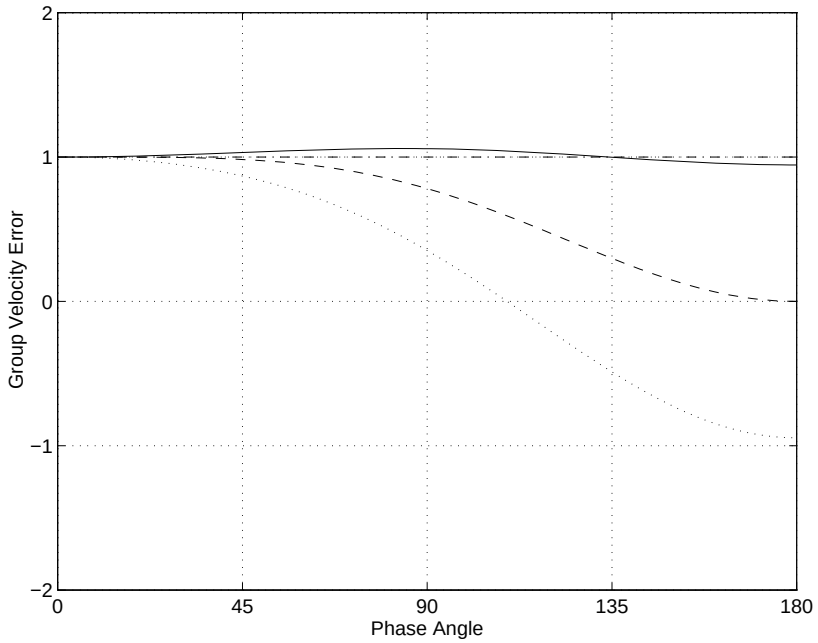

Figure 12: The Group Velocity Error for the semi-Lagrangian method using cubic spline interpolation. The four values for $\hat{\alpha}$ illustrated are 0.25 (dotted), 0.50 (dashed), 0.75 (solid) and 1 (dashed-dotted) where $p = 0$. Thus $\sigma = 0.25, 0.50, 0.75$ and 1. This figure is valid for all values of K and θ.

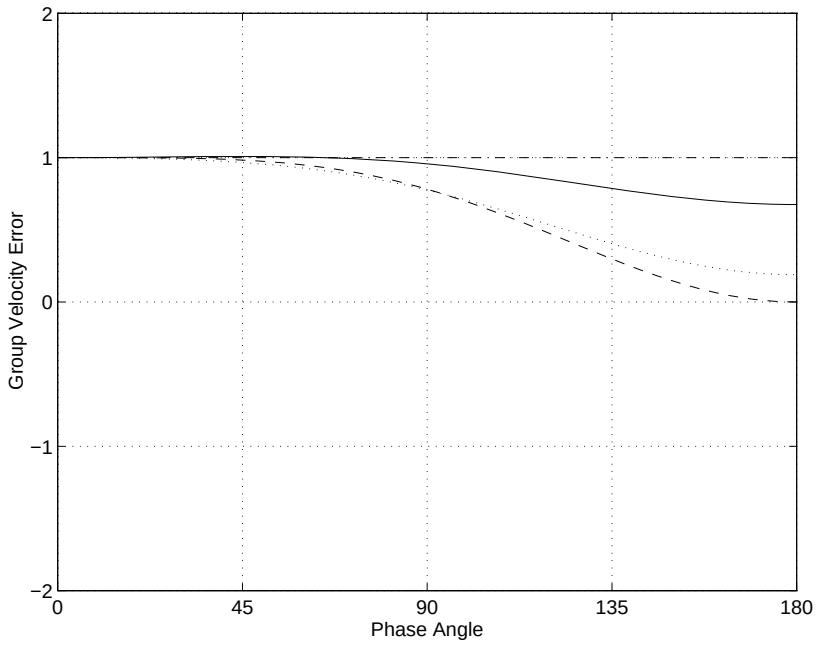

Figure 13: The Group Velocity Error for the semi-Lagrangian method using cubic spline interpolation. The four values for $\hat{\alpha}$ illustrated are 0.25 (dotted), 0.50 (dashed), 0.75 (solid) and 1 (dashed-dotted) where $p = 1$. Thus $\sigma = 1.25, 1.50, 1.75$ and 2. This figure is valid for all values of K and θ.

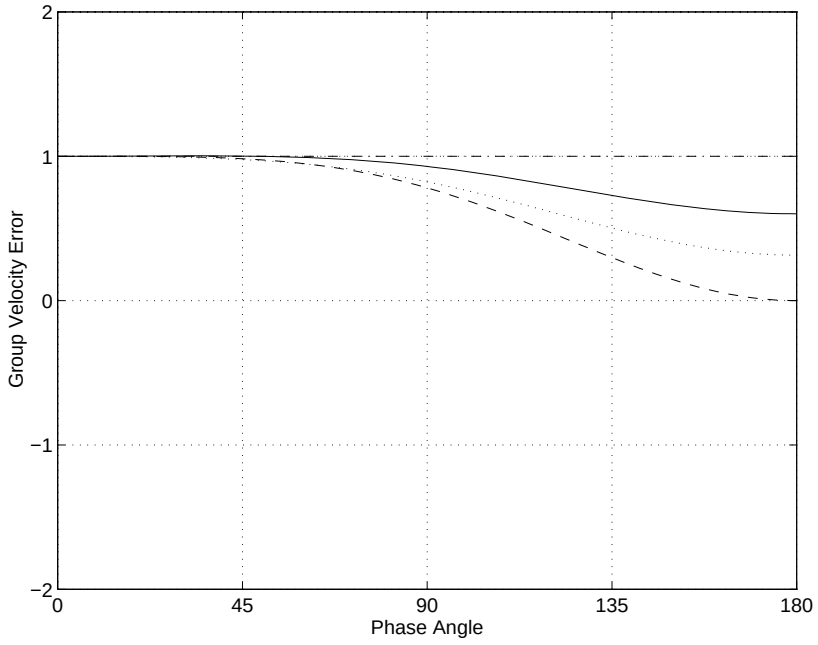

Figure 14: The Group Velocity Error for the semi-Lagrangian method using cubic spline interpolation. The four values for $\hat{\alpha}$ illustrated are 0.25 (dotted), 0.50 (dashed), 0.75 (solid) and 1 (dashed-dotted) where $p = 2$. Thus $\sigma = 2.25, 2.50, 2.75$ and 3. This figure is valid for all values of K and θ.

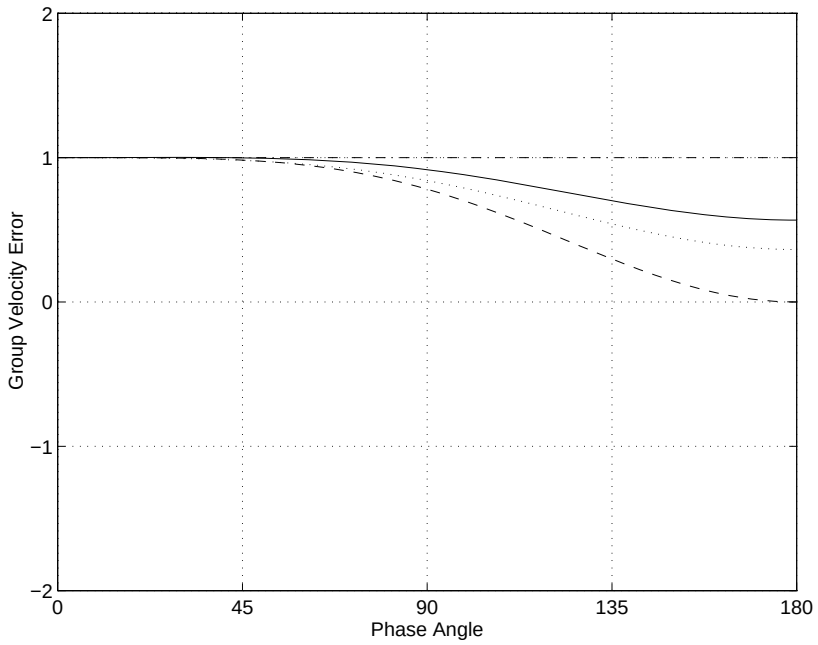

Figure 15: The Group Velocity Error for the semi-Lagrangian method using cubic spline interpolation. The four values for $\hat{\alpha}$ illustrated are 0.25 (dotted), 0.50 (dashed), 0.75 (solid) and 1 (dashed-dotted) where $p = 3$. Thus $\sigma = 3.25, 3.50, 3.75$ and 4. This figure is valid for all values of K and θ.

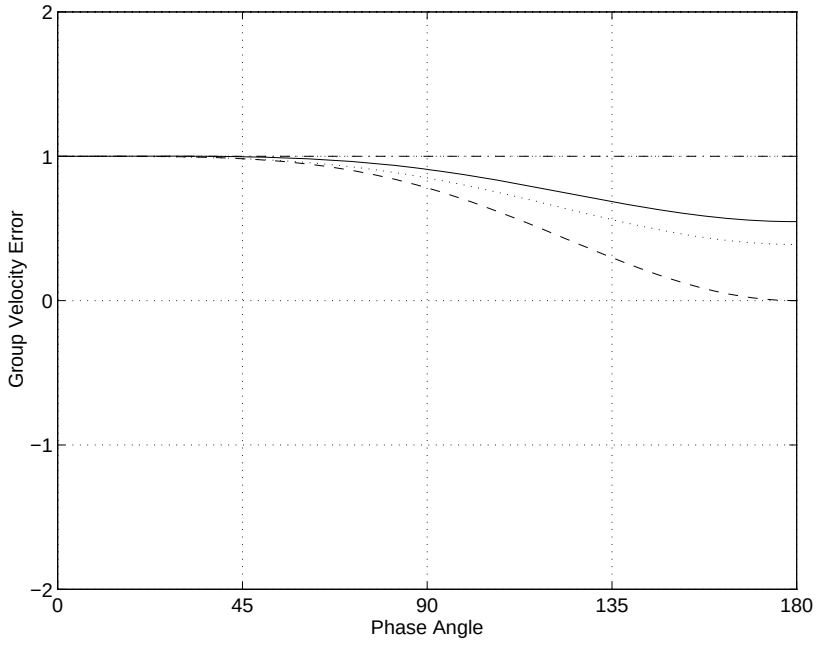

Figure 16: The Group Velocity Error for the semi-Lagrangian method using cubic spline interpolation. The four values for $\hat{\alpha}$ illustrated are 0.25 (dotted), 0.50 (dashed), 0.75 (solid) and 1 (dashed-dotted) where $p = 4$. Thus $\sigma = 4.25, 4.50, 4.75$ and 5. This figure is valid for all values of K and θ.

which gives the amplification factor

$$f_d = \exp^{-im\alpha}$$

which is stable for any value of α. Generally speaking, the departure points do not lie on grid points thereby requiring some form of interpolation. In this paper, cubic spline interpolation is used to approximate the departure points. Using anything lower than cubic interpolation eliminates any advantages that the semi-Lagrangian method might offer [13]. In addition, using Lagrange interpolation as opposed to Hermite or spline interpolation also greatly diminishes the accuracy of the solution. For cubic spline interpolation we obtain

$$\begin{aligned}\tilde{\varphi}_d^n &= \varphi_{j-p} - \hat{\alpha}\dot{\varphi}_{j-p} + \hat{\alpha}^2[3(\varphi_{j-p-1} - \varphi_{j-p}) + \dot{\varphi}_{j-p-1} + 2\dot{\varphi}_{j-p}] \\ &\quad -\hat{\alpha}^3[2(\varphi_{j-p-1} - \varphi_{j-p}) + \dot{\varphi}_{j-p-1} + \dot{\varphi}_{j-p}]\end{aligned}$$

where we have defined the departure point as in [1] to be p grid intervals away from the arrival point j and

$$\hat{\alpha} = \frac{\alpha}{\Delta x} - p \tag{25}$$

is the residual Courant number and $\sigma = \dfrac{\alpha}{\Delta x}$ is the Courant number. In addition, the terms $\dot{\varphi}_{j-p}$ are obtained from the recurrence relation

$$\dot{\varphi}_{j-p-1} + 4\dot{\varphi}_{j-p} + \dot{\varphi}_{j-p+1} = 3(\varphi_{j-p+1} - \varphi_{j-p-1}) \quad j - p = 1, ..., n$$

where n is the number of gridpoints in the mesh. This interpolation yields

$$\begin{aligned}f_d = \left[\exp^{-ip\phi}\right][1 + \hat{\alpha}^2(2\hat{\alpha} - 3)(1 - \cos\phi) - \hat{\alpha}^2(\hat{\alpha} - 1)(1 - \cos^2\phi)] \\ -i\sin\phi[\hat{\alpha}(\hat{\alpha} - 1)(\hat{\alpha}(1 + \cos\phi) - 1) - \hat{\alpha}^2(2\hat{\alpha} - 3)]\end{aligned}$$

which says that the method is stable for all p, since $0 \leq \hat{\alpha} \leq 1$ by definition. Thus the two-time level semi-Lagrangian method is unconditionally stable for advection ($K = \mu = 0$) and advection-diffusion. The dispersion relation is given by

$$\Phi = \arctan\left[\frac{a}{b}\right] \tag{26}$$

where

$$\begin{aligned}a &= B\sin p\phi + A\cos p\phi\,, \\ b &= B\cos p\phi - A\sin p\phi\,,\end{aligned}$$

and

$$\begin{aligned}A &= [\hat{\alpha}(\hat{\alpha} - 1)(\hat{\alpha}(1 + \cos\phi) - 1) - \hat{\alpha}^2(2\hat{\alpha} - 3)]\sin\phi\,, \\ B &= [1 + \hat{\alpha}^2(2\hat{\alpha} - 3)(1 - \cos\phi) - \hat{\alpha}^2(\hat{\alpha} - 1)(1 - \cos^2\phi)]\,,\end{aligned}$$

and the dispersion error is defined by (14). The group velocity and group velocity error are defined once again by equations (15) and (17) where the derivative of the tan function is given by (16) where

$$\begin{aligned}a' &= \sin p\phi(B' - Ap) + \cos p\phi(Bp + A')\,, \\ b' &= \cos p\phi(B' - Ap) - \sin p\phi(Bp + A')\,,\end{aligned}$$

and

$$\begin{aligned}A' &= \cos\phi\left\{\hat{\alpha}(\hat{\alpha} - 1)[\hat{\alpha}(1 + \cos\phi) - 1] - \hat{\alpha}^2(2\hat{\alpha} - 3)\right\} - \sin^2\phi\hat{\alpha}^2(\hat{\alpha} - 1) \\ B' &= \hat{\alpha}^2\sin\phi\,[2\hat{\alpha} - 3 - 2(\hat{\alpha} - 1)\cos\phi]\,.\end{aligned}$$

Figures 5 and 6 show the amplitude errors for different values of $\hat{\alpha}$ for advection and advection-diffusion, respectively. Figures 7 - 15 show the dispersion and group velocity errors for different values of $\hat{\alpha}$ for advection-diffusion. By looking at the dispersion and group velocity relations for the semi-Lagrangian method, we can see that these errors are associated only with advection, because

the diffusion terms drop out from these relations. Therefore, these errors are independent of K and θ because θ only affects the diffusion terms. The four different values of $\hat{\alpha}$ shown are 0.25, 0.50, 0.75 and 1 which correspond to the departure point lying one-quarter, one-half, three-quarters, and one grid point distance away from the p grid point. First considering advection, figures 5 and 7 - 10 show that while the dispersion error is large for $\phi \to \pi$ (short waves) these short dispersive waves are dampened for $\sigma \leq 4$ as long as the dispersion occurs at phase angles $\phi > \frac{\pi}{4}$. This damping for the short dispersive waves keeps the solution stabilized and accurate. Note that figures 1 and 3 show that the short dispersive waves are not damped for the Eulerian case and as a result, the solution is adversely affected by dispersion errors. Figure 11 shows that for $\sigma > 4$, there is dispersion associated with some phase angles $\phi < \frac{\pi}{4}$ which include long waves. Thus as the Courant number increases beyond 4, the semi-Lagrangian method suffers from inaccuracies due to dispersion errors that are not damped by the amplitude error. Figures 12-16 show that the group velocities approach the actual advection speed u for all waves for $\hat{\alpha} = 0.25, 0.75$ and 1, but goes to zero for the short waves ($\phi = \pi$) for $\hat{\alpha} = 0.5$ and moves in the opposite direction for the short waves for $\hat{\alpha} = 0.25$ and $p = 0$. However there are two points which salvage the semi-Lagrangian method: first, by looking at figure 5 (amplitude plot), we can see that these waves are damped and so do not affect the solution, and second, that as p and hence the Courant number σ increases, the group velocities approach the advection speed u for all values of $\hat{\alpha}$ except for 0.5. At $\hat{\alpha} = 0.5$, the numerical method exhibits dispersion errors accompanied by amplitude errors thereby eliminating any ill effects that the short dispersive waves may have caused. Figure 6 shows that for advection-diffusion, the diffusion associated with the governing equations eliminates one of the advantages of the semi-Lagrangian method over the Eulerian method, namely, the diffusion of the short dispersive waves. This means that for advection-diffusion, the semi-implicit Eulerian method will be competitive with the semi-implicit semi-Lagrangian method. However, the semi-Lagrangian method clearly exhibits a more realistic group velocity behavior than the Eulerian method.

For the order of accuracy analysis we expand to fourth order in space and third order in time about $(j\Delta x, t)$, as in the Eulerian case, to get

$$\varphi^n_{j-p-3} = \varphi - [p+3]\Delta x\varphi_x + \frac{([p+3]\Delta x)^2}{2}\varphi_{xx} - \frac{([p+3]\Delta x)^3}{6}\varphi_{xxx} + O(\Delta x)^4,$$

$$\varphi^n_{j-p-2} = \varphi - [p+2]\Delta x\varphi_x + \frac{([p+2]\Delta x)^2}{2}\varphi_{xx} - \frac{([p+2]\Delta x)^3}{6}\varphi_{xxx} + O(\Delta x)^4,$$

$$\varphi^n_{j-p-1} = \varphi - [p+1]\Delta x\varphi_x + \frac{([p+1]\Delta x)^2}{2}\varphi_{xx} - \frac{([p+1]\Delta x)^3}{6}\varphi_{xxx} + O(\Delta x)^4,$$

$$\varphi^n_{j-p} = \varphi - p\Delta x\varphi_x + \frac{(p\Delta x)^2}{2}\varphi_{xx} - \frac{(p\Delta x)^3}{6}\varphi_{xxx} + O(\Delta x)^4,$$

$$\varphi^n_{j-p+1} = \varphi - [p-1]\Delta x\varphi_x + \frac{([p-1]\Delta x)^2}{2}\varphi_{xx} - \frac{([p-1]\Delta x)^3}{6}\varphi_{xxx} + O(\Delta x)^4,$$

$$\varphi^n_{j-p+2} = \varphi - [p-2]\Delta x\varphi_x + \frac{([p-2]\Delta x)^2}{2}\varphi_{xx} - \frac{([p-2]\Delta x)^3}{6}\varphi_{xxx} + O(\Delta x)^4,$$

which, after substituting the Taylor series expansions and the relation $p + \hat{\alpha} = \dfrac{\alpha}{\Delta x}$ into (22), gives the local discretization error

$$L(x,t) = \varphi_t + u\varphi_x - K\varphi_{xx} + \left[\frac{\Delta t}{2}\varphi_{tt} - \frac{\Delta t u^2}{2}\varphi_{xx} - \Delta t\theta K\varphi_{txx} + \Delta t(1-\theta)uK\varphi_{xxx}\right] + O(\Delta t^2, \Delta x^2). \tag{27}$$

For pure advection (K=0) we may use the original pde

$$\varphi_t = -u\varphi_x$$

to obtain

$$\varphi_{tt} = -u\varphi_{xt}$$

and

$$\varphi_{tx} = -u\varphi_{xx}$$

thereby yielding the discretization error

$$L(x,t) = \varphi_t + u\varphi_x - K\varphi_{xx} + [0] + O(\Delta t^2, \Delta x^2). \tag{28}$$

For advection-diffusion ($K > 0$) we may use the original equation

$$K\varphi_{xx} = \varphi_t + u\varphi_x$$

to obtain the relations

$$K\varphi_{txx} = \varphi_{tt} + u\varphi_{tx}$$

and

$$K\varphi_{xxx} = \varphi_{tx} + u\varphi_{xx}$$

which, when substituted into the local discretization error, yields

$$L(x,t) = \varphi_t + u\varphi_x - K\varphi_{xx} + [\Delta t(\frac{1}{2} - \theta)(\varphi_{tt} + 2u\varphi_{tx} + u^2\varphi_{xx})] + O(\Delta t^2, \Delta x^2). \tag{29}$$

This relation shows that the method is second order accurate in both space and time for $\theta = \frac{1}{2}$. The semi-Lagrangian method itself is second order accurate in space and time but the accuracy of the numerical scheme is dependent on the order of the interpolation functions used to determine the departure point and on the time discretization, such as explicit, implicit or semi-implicit. In order to obtain second order accuracy, the interpolation functions have to be at least second order accurate, and the time discretization must be semi-implicit for advection-diffusion. In addition, the interpolation functions need not be Hermite or spline, but can also be Lagrange interpolation functions.

3.3 Operator and Matrix Properties

By looking at the Eulerian differential form of the advection-diffusion equation (1) we can see that the operator is not self-adjoint. The self-adjointness of the operator has significant implications for the finite element discretization. If the operator is not self-adjoint, then we cannot obtain classical variational principles for the problem. The finite element method can still be used but finite element equations can only be obtained through the method of weighted residuals. For the semi-Lagrangian differential form (5) the operator is self-adjoint which means that we can obtain variational principles. Because the finite element method is optimal for the discretization of operators having variational principles, the combination of the semi-Lagrangian time integration with the finite element space discretization yields a complementary and powerful numerical technique.

Looking at it another way, consider the Eulerian discretization in (2) with the semi-Lagrangian discretization in (7). From the definitions of the matrices $M, A,$ and D we can see that the resulting coefficient matrix for the Eulerian method is not symmetric while it is for the semi-Lagrangian method. Let us now see how this affects the manner in which we solve the resulting system of linear equations. Firstly, we can try to solve the Eulerian matrix using Jacobi-type iterative methods but we are not guaranteed to converge to a solution. On the other hand, the symmetric property of the semi-Lagrangian matrix not only ensures convergence but also only necessitates the storage of half the matrix. Therefore, by taking advantage of these properties we can solve the semi-Lagrangian matrix by conjugate gradient methods using an incomplete Choleski factorization. This yields a

very powerful and efficient method for solving this class of matrices. For the Eulerian method, we have to resort to either Krylov subspace methods, such as GMRES, or to direct methods which are relatively inefficient. GMRES is a generalized conjugate gradient method that is applicable to non-symmetric matrices. These methods are the best of the unsymmetric solvers but are not as reliable as their symmetric counterparts, namely, the conjugate gradient methods.

4 Numerical Experiments

Numerical experiments are performed on the two-dimensional advection and advection-diffusion equations. For both test cases the domain is defined as

$$x_{min} \leq x \leq x_{max} \quad \text{and} \quad y_{min} \leq y \leq y_{max}$$

where

$$x_{min} = y_{min} = -\frac{(N-1)10^5}{2}, \quad x_{max} = y_{max}\frac{(N-1)10^5}{2},$$

$\Delta x = \Delta y = 10^5$ and N is the number of points in the x and y directions. The initial waves are centered at

$$x_o = x_{min} + \frac{x_{max} - x_{min}}{4}, \quad y_o = y_{min} + \frac{y_{max} - y_{min}}{2}$$

and the velocity field rotates about the center of the domain and is defined as

$$u = +\Omega y \text{ and } v = -\Omega x \tag{30}$$

with $\Omega = 10^{-5}$.

4.1 Semi-Lagrangian Interpolation

For the semi-Lagrangian method, four different methods of computing the trajectories are studied. These are exact trajectory calculation and trajectory interpolation using cubic spline, cubic Hermite, and cubic Lagrange polynomials. The exact trajectory computation uses cubic spline interpolation for the departure point interpolation. The other methods use the same method for interpolation used for the trajectory computation.

4.1.1 Exact Trajectories

Using the relations for the Lagrangian trajectories (4) and the velocity (30), we write

$$\frac{dx}{dt} = +\Omega y \quad \text{and} \quad \frac{dy}{dt} = -\Omega x$$

which can be integrated to yield the equations

$$x(t) = x_o \cos\Omega t + y_o \sin\Omega t \quad \text{and} \quad y(t) = -x_o \sin\Omega t + y_o \cos\Omega t$$

where

$$x_o = x_a \sin\Omega(t+\Delta t) + y_a \cos\Omega(t+\Delta t) \quad \text{and} \quad y_o = x_a \cos\Omega(t+\Delta t) - y_a \sin\Omega(t+\Delta t) \tag{31}$$

and $x_a = x(t+\Delta t)$ and $y_a = y(t+\Delta t)$ are the arrival points. The semi-Lagrangian midpoint trajectories then become

$$\alpha_1 = x_o[\cos\Omega(t+\Delta t) - \cos\Omega t] + y_o[\sin\Omega(t+\Delta t) - \sin\Omega t] \tag{32}$$

$$\alpha_2 = -x_o[\sin\Omega(t+\Delta t) - \sin\Omega t] + y_o[\cos\Omega(t+\Delta t) - \cos\Omega t]. \tag{33}$$

4.1.2 Cubic Lagrange

From the one-dimensional version of the trajectory equation (4) we obtain the relation

$$\alpha = \Delta t \; u\left(x_j - \frac{\alpha}{2}, t + \frac{\Delta t}{2}\right).$$

Since we do not know the departure interval α a priori, we can iterate this relation to obtain it. However, because the iterated midpoint departure point $x_m = x_j - \alpha/2$ generally falls between grid points, we need to interpolate these non-grid point values. Lagrange interpolation yields C^0 approximations. A cubic Lagrange interpolation of the velocity u can be written as

$$u(x_m) = \sum_{i=1}^{4} \psi_i(\xi_m) u_i$$

where $\psi_i(\xi),\ i = 1, ..., 4$ are the Lagrange polynomials and are defined by

$$\psi_1(\xi) = -\tfrac{9}{16}(\xi - 1)(\xi^2 - \tfrac{1}{9})$$

$$\psi_2(\xi) = \tfrac{27}{16}(\xi - \tfrac{1}{3})(\xi^2 - 1)$$

$$\psi_3(\xi) = -\tfrac{27}{16}(\xi + \tfrac{1}{3})(\xi^2 - 1)$$

$$\psi_4(\xi) = \tfrac{9}{16}(\xi + 1)(\xi^2 - \tfrac{1}{9})$$

where $\xi = \dfrac{2(x - x_c)}{\Delta x}$, $x_c = \dfrac{x_4 + x_1}{2}$ and $\Delta x = x_4 - x_1$.

4.1.3 Cubic Hermite

Hermite interpolation uses not only the values at the grid points but also the derivatives, which makes the interpolation C^1. A Hermite interpolant for the velocity can be written as

$$u(x_m) = a_0 + a_1\xi_m + a_2\xi_m^2 + a_3\xi_m^3$$

where

$$a_0 = u(0)$$

$$a_1 = \frac{\partial u}{\partial \xi}(0)$$

$$a_2 = 3[u(1) - u(0)] - 2\frac{\partial u}{\partial \xi}(0) - \frac{\partial u}{\partial \xi}(1)$$

$$a_3 = 2[u(0) - u(1)] + \frac{\partial u}{\partial \xi}(0) + \frac{\partial u}{\partial \xi}(1).$$

Notice that generally, only the values of u at the grid points are known whereas the derivatives at the grid points are not. These derivatives are computed locally using the procedure described in [5] where all of the surrounding elements are used to compute the derivatives at each grid point, thereby using a three-point stencil in one dimension and a nine-point stencil in two dimensions.

4.1.4 Cubic Spline

Cubic splines use the same interpolating functions as in cubic Hermite interpolation; however, they are C^2. Once again, only the grid point values of u are known and it remains to compute the derivatives. The manner in which these derivatives are computed for spline interpolation differs from

Hermite interpolation. In spline interpolation, the derivatives are computed globally by enforcing slope and curvature continuity at all grid points yielding the following relations

$$\left(\frac{\partial u}{\partial \xi}\right)_{i-1} + 4\left(\frac{\partial u}{\partial \xi}\right)_{i} + \left(\frac{\partial u}{\partial \xi}\right)_{i+1} = 3(u_{i+1} - u_{i-1}) \quad \text{for} \quad i = 1, ..., n$$

which defines a tridiagonal system that can be solved efficiently.

4.2 2D Advection

The advection equation

$$\frac{\partial \varphi}{\partial t} + u\frac{\partial \varphi}{\partial x} + v\frac{\partial \varphi}{\partial y} = 0$$

is given by setting $K = 0$ in (1). The initial condition is given as in [13] by the cosine wave

$$\varphi_o = \frac{100}{2}\left[1 + \cos\frac{\pi r}{R}\right]$$

$$\text{where} \quad r = \sqrt{(x - x_o)^2 + (y - y_o)^2} \quad \text{and} \quad R = 4\Delta x.$$

The boundary conditions are assumed periodic in all directions. The analytic solution of this problem is

$$\varphi_{\text{exact}}(x, y, t) = \varphi_o(x - ut, y - vt, t)$$

which is the rotation of the cosine wave about the center of the domain without any dissipation or deformation. The ℓ^2 error norm is defined in the following way

$$||e||_{\ell^2} = \sqrt{\frac{\int\int[\varphi(x,y,t) - \varphi_{\text{exact}}(x,y,t)]^2 dx dy}{\int\int[\varphi_{\text{exact}}(x,y,t)]^2 dx dy}}. \tag{34}$$

Table 1 shows the results obtained using the Eulerian methods for $\theta = \frac{1}{2}$, 1 and 0. The results are illustrated for up to five revolutions. It is clear that the best Eulerian method is obtained by $\theta = \frac{1}{2}$. This is the semi-implicit Eulerian method which is second order accurate in both time and space. The superiority of the $\theta = \frac{1}{2}$ method over the other two can be seen not only by comparing the ℓ^2 error norm but also by the values of φ_{max}. In addition, this method best conserves the first and second moments of the conservation variable. However, one discernible problem with this method is its minimum values. The analytic maximum and minimum should be 100 and 0. The $\theta = \frac{1}{2}$ method clearly yields unwanted dispersion errors in the form of large negative numbers for the minimum values (see table 1).

Table 2 shows the results obtained using the semi-Lagrangian method. For pure advection, this method is second order accurate in both time and space for all values of θ. This table shows the results for the semi-Lagrangian method using four different types of interpolation methods. This test was carried out to determine the amount of error introduced by iteratively computing the trajectories and interpolating the departure point values. In this case we have the luxury of a known velocity field, but in practice, it is unknown. The four methods of interpolation are: exact, cubic Lagrange, cubic Hermite, and cubic spline interpolation. In the exact method, the trajectories are computed exactly by virtue of equations 31 - 33 but the departure point values are interpolated using cubic splines. This test serves to show the error introduced by using cubic splines to compute the trajectories. Table 2 shows that very little error is introduced by numerically computing the trajectories using cubic splines. This table also shows that there is little difference between the cubic Lagrange and Hermite interpolation, however, they yield much less accurate solutions than the cubic spline. Nonetheless, all of these semi-Lagrangian methods yield much better results than the Eulerian methods, including the semi-implicit Eulerian method. This can be seen not just by comparing the ℓ^2 norms but by comparing the maximum and minimum values as well. Note that the semi-Lagrangian method introduces very little dispersion error while the Eulerian methods are hampered by dispersion and/or damping errors. However, the semi-Lagrangian method does seem to suffer from non-conservation.

θ	Revolutions	$\|\|e\|\|_{\ell^2}$	φ_{max}	φ_{min}	$\sum \frac{\varphi_{ij}}{\varphi_{\text{exact}_{ij}}}$	$\sum \frac{\varphi_{ij}^2}{\varphi_{\text{exact}_{ij}}^2}$
1/2	1	0.2442	96.66	-13.63	1.001	1.000
	2	0.4052	91.24	-20.01	0.998	1.000
	3	0.5136	84.61	-23.43	1.000	1.000
	4	0.5913	78.76	-24.35	1.004	1.001
	5	0.6505	75.83	-25.41	0.996	1.002
1	1	0.6478	37.93	-0.38	1.000	0.347
	2	0.7486	27.67	-0.41	1.000	0.253
	3	0.7951	22.88	-0.42	1.000	0.208
	4	0.8230	19.87	-0.34	0.999	0.181
	5	0.8422	17.79	-0.40	0.999	0.163
0	1	0.9374	4.83	0.00	1.042	0.039
	2	0.9744	2.26	0.00	1.056	0.027
	3	0.9838	1.67	0.00	1.061	0.026
	4	0.9863	1.51	0.00	1.062	0.026
	5	0.9870	1.47	0.00	1.063	0.026

Table 1: The Eulerian method for the advection equation. $\theta = \frac{1}{2}$ is the semi-implicit method, $\theta = 1$ is the implicit method, and $\theta = 0$ is the explicit method. The grid is 33×33 and $\sigma = \frac{\pi}{4}$.

Interpolation	Revolutions	$\|\|e\|\|_{\ell^2}$	φ_{max}	φ_{min}	$\sum \frac{\varphi_{ij}}{\varphi_{\text{exact}_{ij}}}$	$\sum \frac{\varphi_{ij}^2}{\varphi_{\text{exact}_{ij}}^2}$
Exact	1	0.0459	98.45	-1.26	0.998	0.968
	2	0.0732	95.79	-1.75	1.001	0.941
	3	0.0970	92.98	-1.98	1.004	0.918
	4	0.1186	90.33	-2.12	1.008	0.897
	5	0.1383	87.91	-2.26	1.010	0.878
Spline	1	0.0674	98.28	-1.36	1.000	0.968
	2	0.1210	95.05	-1.81	0.999	0.942
	3	0.1714	91.37	-2.03	1.000	0.918
	4	0.2194	87.67	-1.97	1.001	0.897
	5	0.2652	85.33	-2.09	1.003	0.878
Hermite	1	0.1846	83.91	-3.11	0.999	0.863
	2	0.2913	71.99	-3.36	1.000	0.778
	3	0.3672	64.47	-3.86	1.003	0.717
	4	0.4252	59.74	-3.67	1.007	0.669
	5	0.4719	55.86	-3.56	1.013	0.631
Lagrange	1	0.1973	80.98	-1.65	0.999	0.813
	2	0.3054	68.88	-1.91	1.001	0.713
	3	0.3797	61.00	-1.91	1.004	0.646
	4	0.4354	55.30	-1.89	1.009	0.597
	5	0.4798	51.26	-1.89	1.014	0.558

Table 2: The semi-Lagrangian method for the advection equation. Four types of interpolation methods are illustrated. These are exact, cubic spline, cubic Hermite, and cubic Lagrange. The grid is 33×33 and $\sigma = \pi$.

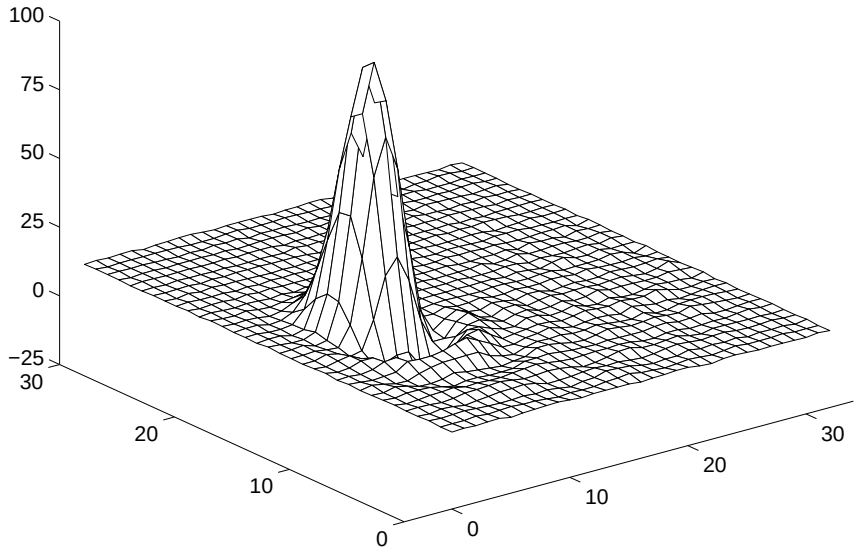

Figure 17: The semi-implicit **Eulerian** ($\theta = \frac{1}{2}$) solution after one revolution for the advection equation. The grid dimension is 33×33 and $\sigma = \frac{\pi}{4}$.

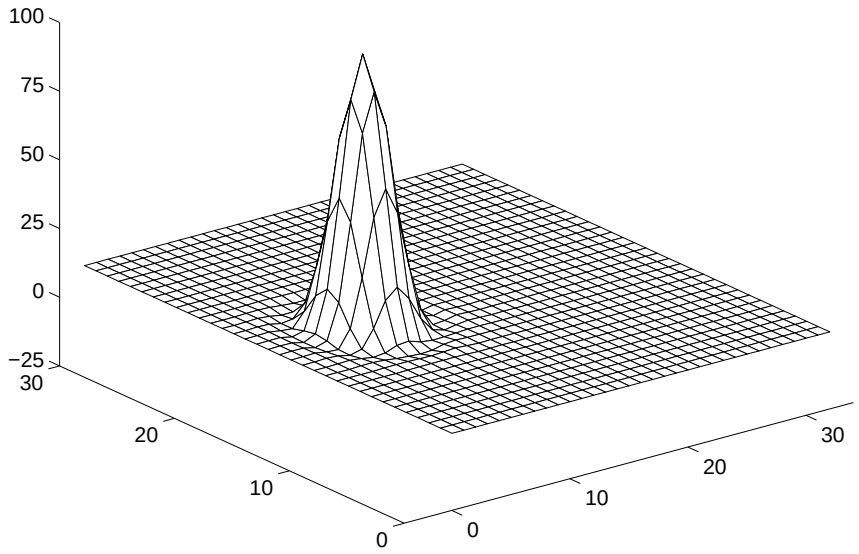

Figure 18: The semi-implicit **Semi-Lagrangian** ($\theta = \frac{1}{2}$) cubic spline solution after one revolution for the advection equation. The grid dimension is 33×33 and $\sigma = \pi$.

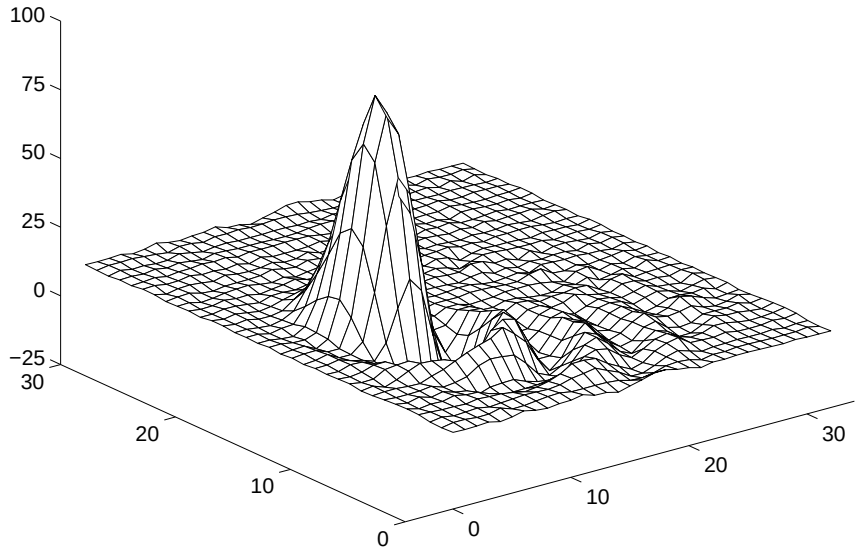

Figure 19: The semi-implicit **Eulerian** ($\theta = \frac{1}{2}$) solution after three revolutions for the advection equation. The grid dimension is 33×33 and $\sigma = \frac{\pi}{4}$.

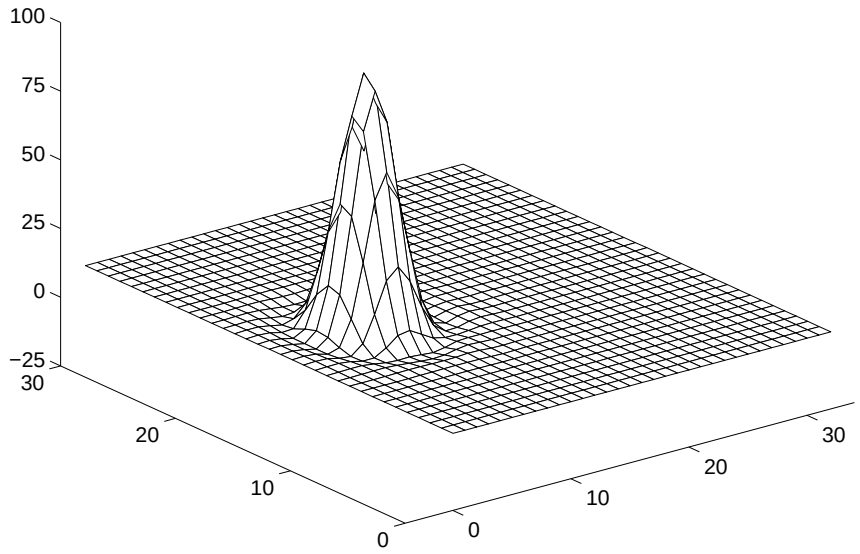

Figure 20: The semi-implicit **Semi-Lagrangian** ($\theta = \frac{1}{2}$) cubic spline solution after three revolutions for the advection equation. The grid dimension is 33×33 and $\sigma = \pi$.

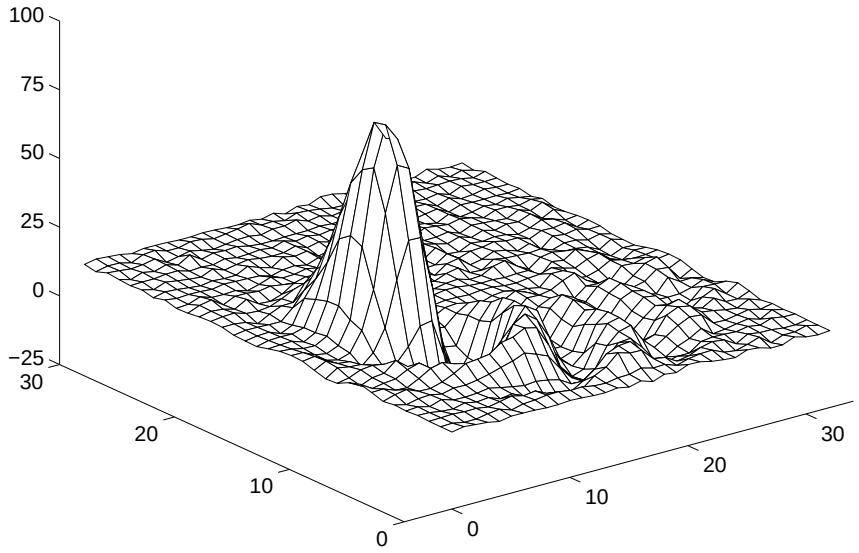

Figure 21: The semi-implicit **Eulerian** ($\theta = \frac{1}{2}$) solution after five revolutions for the advection equation. The grid dimension is 33×33 and $\sigma = \frac{\pi}{4}$.

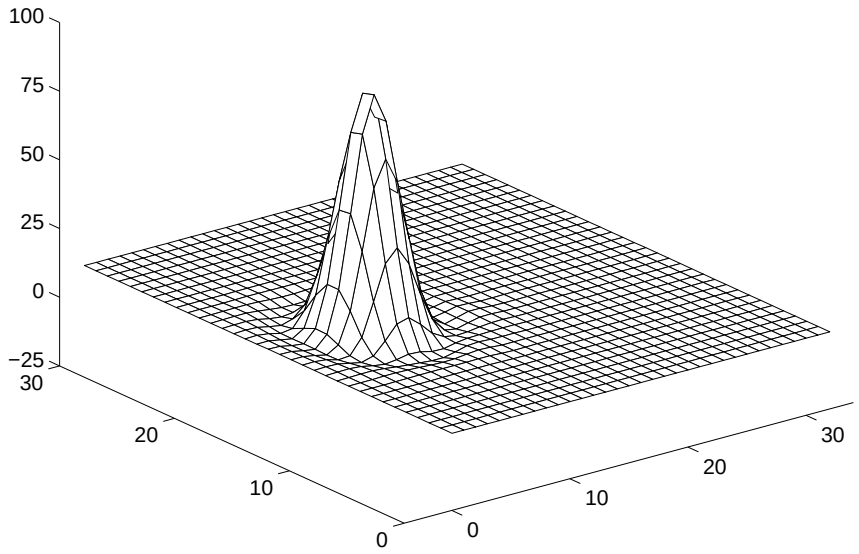

Figure 22: The semi-implicit **Semi-Lagrangian** ($\theta = \frac{1}{2}$) cubic spline solution after five revolutions for the advection equation. The grid dimension is 33×33 and $\sigma = \pi$.

σ	Revolutions	$\|\|e\|\|_{\ell^2}$	φ_{max}	φ_{min}	$\sum \frac{\varphi_{ij}}{\varphi_{\text{exact}_{ij}}}$	$\sum \frac{\varphi_{ij}^2}{\varphi_{\text{exact}_{ij}}^2}$
π	1	0.0674	98.28	-1.36	1.000	0.968
	2	0.1210	95.05	-1.81	0.999	0.942
	3	0.1714	91.37	-2.03	1.000	0.918
	4	0.2194	87.67	-1.97	1.001	0.897
	5	0.2652	85.33	-2.09	1.003	0.878
2π	1	0.1911	96.52	-1.10	1.002	0.987
	2	0.3713	97.28	-1.39	1.002	0.974
	3	0.5391	95.38	-1.35	1.002	0.961
	4	0.6919	92.13	-1.53	1.003	0.949
	5	0.8280	92.05	-1.62	1.003	0.938

Table 3: The semi-implicit semi-Lagrangian method ($\theta = \frac{1}{2}$) with cubic spline interpolation for the advection equation using different Courant numbers. The grid is 33×33, and $\sigma = \pi$ and 2π.

The semi-Lagrangian method has been known to be non-conservative ([13], [14]) and table 2 clearly shows the loss in the second moment of the conservation variable. It is important to understand that while the semi-Lagrangian method is clearly superior to the Eulerian method, it has achieved these results using Courant numbers four times greater than that used for the Eulerian method. However, as the Courant number is increased beyond four, the semi-Lagrangian method suffers inaccuracies due to dispersion errors. Table 3 compares the semi-implicit semi-Lagrangian method using Courant numbers of $\sigma = \pi$ and 2π. Clearly, the error norm has more than doubled while the time step was only increased by a factor of 2 and this trend worsens after each revolution. For increasing Courant number, the dispersion and group velocity errors become progressively worse, which is what the stability analysis of section 3 predicted thus confirming our numerical results.

Figures 17 - 22 compare the solutions graphically for the semi-implicit Eulerian method ($\theta = \frac{1}{2}$) with the semi-implict semi-Lagrangian method using cubic spline interpolation. These figures show the solution after one, three and five revolutions. These figures clearly show the unwanted dispersion errors introduced by the Eulerian method. After five revolutions, the solution obtained by the Eulerian method is no longer acceptable due to the ever increasing errors. In contrast, the semi-Lagrangian method captures the analytic solution much more accurately without suffering the dispersion errors that plague the Eulerian method.

4.3 2D Advection-Diffusion

The initial condition for the advection-diffusion problem is given as in [13] by the following exponential function

$$\varphi_o = 100 \exp^{-\frac{r^2}{4\Delta x^2}}.$$

All of the variables are defined as in the advection problem, and the diffusion coefficient K assumes the values 1×10^4, 5×10^4, and 7×10^4. The boundary condition is

$$\nabla\varphi \cdot \vec{n} = 0$$

where $\vec{n}$ is the outward pointing normal vector to the boundaries. In an infinite plane, the analytic solution of this problem is given as in [13] by

$$\varphi_{\text{exact}}(x, y, t) = \frac{100}{1 + \frac{Kt}{\Delta x^2}} \exp^{-\frac{\tilde{x}^2 + \tilde{y}^2}{4\Delta x^2 + 4Kt}}$$

Method	θ	σ	K	$\|\|e\|\|_{\ell^2}$	φ_{max}	φ_{min}	$\sum \frac{\varphi_{ij}}{\varphi_{\text{exact}_{ij}}}$	$\sum \frac{\varphi_{ij}^2}{\varphi^2_{\text{exact}_{ij}}}$
Eulerian	1/2	$\pi/4$	1×10^4	0.1185	55.96	-3.47	1.000	0.989
			5×10^4	0.0165	23.61	0.00	1.002	0.991
			7×10^4	0.0117	18.21	0.00	1.009	0.994
	1	$\pi/4$	1×10^4	0.3477	32.69	0.00	1.000	0.555
			5×10^4	0.1318	17.11	0.00	1.006	0.723
			7×10^4	0.0974	13.97	0.00	1.018	0.768
	0	$\pi/4$	1×10^4	0.7081	6.17	0.00	1.070	0.117
			5×10^4	0.3847	5.70	0.00	1.084	0.270
			7×10^4	0.3155	5.50	0.00	1.094	0.340
SLM spline	1/2	π	1×10^4	0.0341	61.86	0.00	1.000	1.009
			5×10^4	0.0185	25.21	0.00	1.003	1.021
			7×10^4	0.0154	19.24	0.00	1.011	1.020
	1	π	1×10^4	0.0419	58.02	0.00	1.000	0.976
			5×10^4	0.0133	23.82	0.00	1.004	0.994
			7×10^4	0.0124	18.42	0.00	1.012	1.000
	0	π	1×10^4	0.5768	12.11	0.00	1.021	0.213
			5×10^4	0.2596	9.40	0.00	1.036	0.414
			7×10^4	0.2000	8.45	0.00	1.047	0.490

Table 4: The Eulerian and semi-Lagrangian methods for the advection-diffusion equation. $\theta = \frac{1}{2}$ is the semi-implicit method, $\theta = 1$ is the implicit method, $\theta = 0$ is the explicit method, and SLM is the semi-Lagrangian method. The grid is 33×33, and $\sigma = \frac{\pi}{4}$ for the Eulerian methods and $\sigma = \pi$ for the semi-Lagrangian methods.

σ	K	$\|\|e\|\|_{\ell^2}$	φ_{max}	φ_{min}	$\sum \frac{\varphi_{ij}}{\varphi_{\text{exact}_{ij}}}$	$\sum \frac{\varphi_{ij}^2}{\varphi^2_{\text{exact}_{ij}}}$
$\pi/2$	1×10^4	0.0292	58.17	0.00	1.000	0.983
	5×10^4	0.0109	24.90	0.00	1.003	1.017
	7×10^4	0.0108	19.11	0.00	1.011	1.018
π	1×10^4	0.0341	61.86	0.00	1.000	1.009
	5×10^4	0.0185	25.21	0.00	1.003	1.021
	7×10^4	0.0154	19.24	0.00	1.011	1.020
$3\pi/2$	1×10^4	0.0818	62.18	0.00	1.000	1.016
	5×10^4	0.0352	25.21	0.00	1.004	1.024
	7×10^4	0.0276	19.24	0.00	1.011	1.023

Table 5: The semi-implicit semi-Lagrangian method ($\theta = \frac{1}{2}$) with cubic spline interpolation for the advection-diffusion equation using different Courant numbers. The grid is 33×33, and $\sigma = \frac{\pi}{2}$, π and $\frac{3\pi}{2}$.

Method	θ	σ	K	$\|\|e\|\|_{\ell^2}$	φ_{max}	φ_{min}	$\sum \frac{\varphi_{ij}}{\varphi_{\text{exact}_{ij}}}$	$\sum \frac{\varphi_{ij}^2}{\varphi^2_{\text{exact}_{ij}}}$
SLM	1/2	π	1×10^4	0.0900	52.00	-0.315	1.000	0.915
Lagrange			5×10^4	0.0132	24.07	-0.098	1.001	0.999
			7×10^4	0.0131	18.70	-0.172	1.005	1.007
	1	π	1×10^4	0.1088	49.99	-0.382	1.000	0.893
			5×10^4	0.0202	22.95	-0.085	1.001	0.974
			7×10^4	0.0159	17.99	-0.161	1.008	0.988
	0	π	1×10^4	0.5790	11.97	-0.308	1.012	0.214
			5×10^4	0.2633	9.34	-0.403	1.025	0.421
			7×10^4	0.2053	8.40	-0.439	1.034	0.503
SLM	1/2	π	1×10^4	0.0841	53.55	-0.513	1.000	0.936
Hermite			5×10^4	0.0152	24.30	-0.101	1.001	1.005
			7×10^4	0.0141	18.82	-0.176	1.005	1.001
	1	π	1×10^4	0.1015	51.34	-0.588	1.000	0.912
			5×10^4	0.0193	23.14	-0.087	1.001	0.979
			7×10^4	0.0155	18.08	-0.165	1.007	0.992
	0	π	1×10^4	0.5791	12.00	-0.320	1.011	0.214
			5×10^4	0.2628	9.36	-0.418	1.021	0.420
			7×10^4	0.2049	8.41	-0.455	1.034	0.502

Table 6: The semi-Lagrangian method for the advection-diffusion equation using cubic Lagrange and Hermite interpolation. $\theta = \frac{1}{2}$ is the semi-implicit method, $\theta = 1$ is the implicit method, $\theta = 0$ is the explicit method. The grid is 33×33 and $\sigma = \pi$.

where

$$\tilde{x} = x - x_o \cos\Omega t - y_o \sin\Omega t \quad \text{and} \quad \tilde{y} = y + x_o \sin\Omega t - y_o \cos\Omega t.$$

Table 4 shows the results for the Eulerian and semi-Lagrangian methods for various values of K and θ. The integration is carried out for one revolution only because it is assumed that up to this point, the boundaries do not affect the solution and the domain can be assumed infinite. Once again, the best solutions are given by the semi-implicit methods ($\theta = \frac{1}{2}$). For the lower diffusion coefficient (1×10^4), the semi-implicit Eulerian method is still affected by dispersion as is illustrated in figure 23. But as the diffusion coefficient K increases, the equation is dominated by diffusion rather than advection and the dispersion error dissipates and as a result the accuracy of the numerical solution increases. This tells us that the numerical scheme does a much better job of capturing the diffusion effects as opposed to the advection thereby confirming the results of the one-dimensional analysis presented in section 3.

The bottom half of table 4 shows the results obtained using the semi-Lagrangian method using cubic spline interpolation. This table illustrates the poor solution quality yielded by the explicit form of the semi-Lagrangian method. It is still better than its Eulerian counterpart, but it is nonetheless not a good choice and should be avoided.

For the lower value of K (1×10^4) the solution obtained by the semi-implicit semi-Lagrangian method (figure 24) is better than that obtained by the semi-implicit Eulerian method (figure 23), while using a Courant number four times larger. But as K increases, we see that the semi-implicit Eulerian solution becomes competitive with the semi-Lagrangian method. For $K = 5 \times 10^4$ and 7×10^4 the semi-implicit Eulerian solution is slightly better than the semi-Lagrangian method for $\sigma = \pi$ and twice as accurate as the semi-Lagrangian method for $\sigma = \frac{3\pi}{2}$. This tells us two things: one, that the semi-Lagrangian method is diminishing in accuracy for Courant numbers greater than four, and that for advection-diffusion the semi-implicit Eulerian method appears to become more accurate than the semi-Lagrangian method.

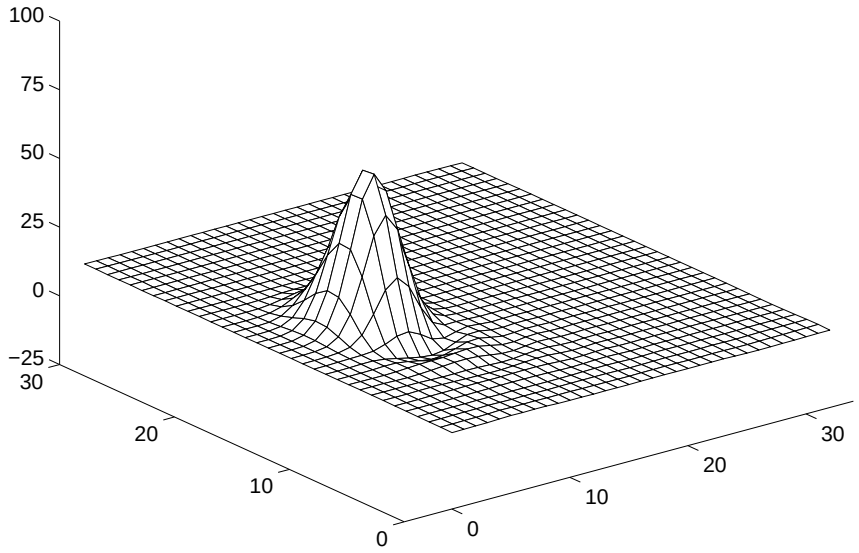

Figure 23: The semi-implicit **Eulerian** ($\theta = \frac{1}{2}$) solution after one revolution for the advection-diffusion equation with $K = 1 \times 10^4$. The grid dimension is 33×33 and $\sigma = \frac{\pi}{4}$.

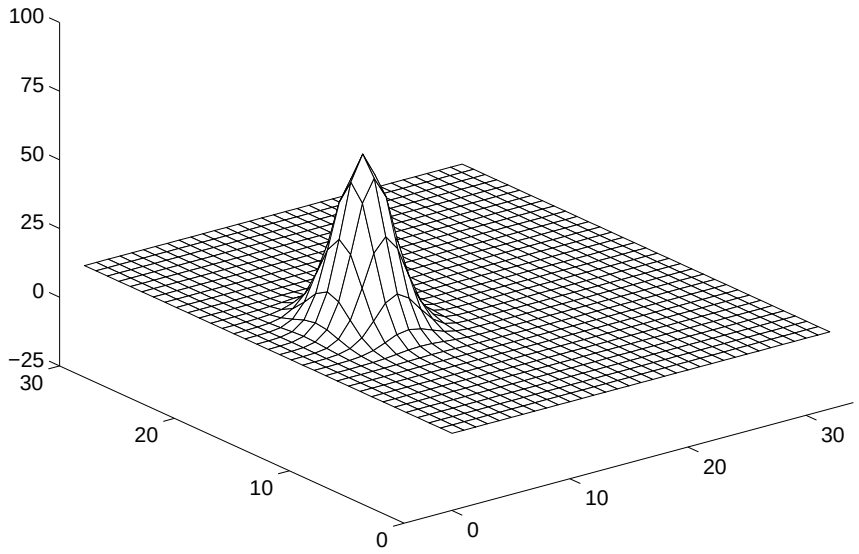

Figure 24: The semi-implicit **Semi-Lagrangian** ($\theta = \frac{1}{2}$) cubic spline solution after one revolution for the advection-diffusion equation with $K = 1 \times 10^4$. The grid dimension is 33×33 and $\sigma = \pi$.

To confirm that this is not the case and that the semi-Lagrangian method is still more accurate than the Eulerian method, we have run the semi-implicit semi-Lagrangian model using a Courant number only two times greater. These results are illustrated in table 5. In other words the semi-Lagrangian model was run at a Courant number $\sigma = \frac{\pi}{2}$ showing that the semi-Lagrangian method is still superior to the Eulerian method for all values of K.

Table 6 illustrates the results for the semi-Lagrangian method using cubic Lagrange and Hermite interpolation. There do not appear to be too many differences between the two methods. However, it is important to note that as the diffusion coefficient increases, the Lagrange and Hermite semi-Lagrangian methods compete with the cubic spline semi-Lagrangian method. Nonetheless, all of the semi-Lagrangian methods prove to be more accurate and efficient than the Eulerian methods whether for advection or advection-diffusion.

5 Conclusions and Future Work

A family of Eulerian and semi-Lagrangian finite element methods were analyzed for stability and accuracy. This included explicit, implicit, and semi-implicit methods. The semi-implicit Eulerian and semi-Lagrangian methods are second order accurate in both space and time. In addition, both methods are unconditionally stable. However, for very large time steps the accuracy of both methods diminishes but the semi-Lagrangian method still allows time steps two to four times larger than the semi-implicit Eulerian method for a given level of accuracy. This property makes semi-Lagrangian methods more attractive than Eulerian methods for integrating atmospheric and ocean equations particularly because long time histories are sought for such problems. The semi-implicit Eulerian method ($\theta = \frac{1}{2}$) was shown to be too dispersive for advection because this method has no accompanying damping for the short dispersive waves. For the semi-Lagrangian method, there is no dispersion associated with the long waves and for the short dispersive waves there is a damping associated with them thereby resulting in a more accurate solution than obtained by the Eulerian method. The analysis also shows that the accuracy of the semi-implicit semi-Lagrangian method greatly diminishes for Courant numbers greater than four because dispersion errors are introduced for the long waves ($\phi > \frac{\pi}{5}$) which are not damped by the amplitude errors of the semi-Lagrangian method. The numerical studies corroborate the amplitude, dispersion and group velocity error analyses.

Numerical experiments were performed on the two-dimensional advection and advection-diffusion equations and the results demonstrate the superiority of the semi-implicit semi-Lagrangian method over the semi-implicit Eulerian method not just in terms of accuracy but in terms of efficiency as well, as is evident by the larger time steps allowed by the semi-Lagrangian method. Because the resulting operator for the semi-Lagrangian method is self-adjoint the finite element method offers the optimal discretization. In other words, the resulting coefficient matrix for the semi-Lagrangian method is symmetric positive-definite which means that highly efficient methods of solution, such as the ICCG (incomplete Choleski conjugate gradient method) can be used. This method is extremely efficient because only half the matrix needs to be stored.

Three types of interpolation (cubic spline, cubic Hermite, and cubic Lagrange) for the trajectory and departure point calculations of the semi-Lagrangian method were compared. The numerical results show that cubic spline interpolation is superior to both Lagrange and Hermite interpolation and that very little differences are seen between the latter two types of interpolation. The numerical results also show that the cubic spline method yields results very similar to those obtained by the exact trajectory calculations. Thus, neither the trajectory calculation, nor the departure point computation are responsible for the total mass and energy losses exhibited by semi-Lagrangian methods. Ritchie [15] found that these total energy losses become negligible when the grid resolution is increased. Perhaps an efficient approach may be to employ grid refinement methods [4] or quadtree methods. In addition, mass conserving semi-Lagrangian methods for the advection equation, and mass and energy conserving semi-Lagrangian methods for the shallow water equations need to be explored further ([6], [12]).

Acknowledgements

The first author would like to thank the National Research Council for funding this work. Both authors would like to thank the Naval Research Laboratory and the Naval Postgraduate School for partial funding of this work.

References

[1] J.R. Bates and A. McDonald, 'Multiply-Upstream, Semi-Lagrangian Advective Schemes: Analysis and Applications to a Multi-Level Primite Equation Model', *Monthly Weather Review*, **110**, 1831-1842 (1982).

[2] Francis X. Giraldo and Beny Neta, 'Finite Element Approximation of Large Air Pollution Problems I: Advection', *NPS Technical Report*, # NPS-MA-95-005 (1995).

[3] Francis X. Giraldo, 'A Space Marching Adaptive Remeshing Technique Applied to the 3D Euler Equations for Supersonic Flow', *PhD Dissertation*, University of Virginia (1995).

[4] Francis X. Giraldo, 'A Finite Volume High Resolution 2D Euler Solver with Adaptive Grid Generation on High Performance Computers', *Proceedings of the Ninth International Conference on Finite Elements in Fluids*, **2**, 1031-1040 (1995).

[5] Francis X. Giraldo, 'A Parallel Domain Decomposition Method for a Semi-Lagrangian Finite Element Air Pollution Transport Model', *Proceedings of the International Symposium on Parallel and Distributed Supercomputing*, 164-171 (1995).

[6] Sylvie Gravel and Andrew Staniforth, 'A Mass-Conserving Semi-Lagrangian Scheme for the Shallow Water Equations', *Monthly Weather Review*, **122**, 243-248 (1994).

[7] Kenneth H. Huebner and Earl A. Thornton, *The Finite Element Method for Engineers*, 3rd Edition, Wiley & Sons, New York, 1995.

[8] John D. McCalpin, 'A Quantitative Analysis of the Dissipation Inherent in Semi-Lagrangian Advection', *Monthly Weather Review*, **116**, 2330-2336 (1988).

[9] A. McDonald, 'Accuracy of Multiply-Upstream, Semi-Lagrangian Advective Scheme', *Monthly Weather Review*, **112**, 1267-1275 (1984).

[10] B. Neta and R.T. Williams, 'Stability and Phase Speed for Various Finite Element Formulations of the Advection Equation', *Computers and Fluids*, **14**, 393-410 (1986).

[11] A. Oliviera and A.M. Baptista, 'A Comparison of Integration and Interpolation Eulerian-Lagrangian Methods', *International Journal for Numerical Methods in Fluids*, **21**, 183-204 (1995).

[12] A. Priestley, 'A Quasi-Conservative Version of the Semi-Lagrangian Advection Scheme', *Monthly Weather Review*, **121**, 621-629 (1993).

[13] J. Pudykiewicz and A. Staniforth, 'Some Properties and Comparative Performance of the Semi-Lagrangian Method of Robert in the Solution of the Advection-Diffusion Equation', *Atmosphere-Ocean*, **22**, 283-308 (1984).

[14] D.K. Purnell, 'Solution of the Advective Equation by Upstream Interpolation with a Cubic Spline', *Monthly Weather Review*, **104**, 42-48 (1976).

[15] Harold Ritchie, 'Application of the Semi-Lagrangian Method to a Spectral Model of the Shallow Water Equations', *Monthly Weather Review*, **116**, 1587-1598 (1988).

[16] A. Robert, 'A Stable Numerical Integration Scheme for the Primitive Meteorological Equations', *Atmosphere-Ocean*, **19**, 35-46 (1981).

[17] Andrew Staniforth and Clive Temperton, 'Semi-implicit Semi-Lagrangian Integration Schemes for a Barotropic Finite Element Regional Model', *Monthly Weather Review*, **119**, 2206-2223 (1991).

[18] J.A. Young, 'Comparative Properties of some Time Differencing for Linear and Nonlinear Oscillations', *Monthly Weather Review*, **96**, 357-364 (1968).