A comparison of a family of Eulerian and semi-Lagrangian finite element methods for the advection-diffusion equation

Francis X. Giraldo* & Beny Neta

* Naval Research Laboratory, Monterey, CA 93943
b Naval Postgraduate School, Department of Mathematics, Monterey, CA 93943

Abstract

This paper compares a family of Eulerian and semi-Lagrangian methods. Numerical experiments are performed on the two-dimensional advection-diffusion equations having a known analytic solution. The numerical results demonstrate 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-diffusion equations are presented. The best methods are found to be the semi-implicit methods (θ = 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.
Semi-Lagrangian methods and other related methods such as Characteristic Galerkin and Eulerian-Lagrangian methods have been studied using the advection-diffusion equation in one [6] and two-dimensions [7]. In [6] a class of schemes similar to semi-Lagrangian methods are studied for amplification errors but only for Lagrange interpolation. 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 [8], and Staniforth and Temperton [9]. 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, see Neta and Williams [5]. 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-diffusion equations to validate the methods and corroborate the one-dimensional analysis in [3]. Finally, section 5 contains the conclusions drawn from this study.

2 Discretization

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

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

(1)

where $\varphi$ 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

$$M \varphi + (A + D) \varphi = R$$
where $M$ is the mass matrix, $A$ the advection, $D$ the diffusion, and $R$ the boundary terms which are given by

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

$$D_{ij} = K \int_\Omega \left( \frac{\partial \psi_i}{\partial x} \frac{\partial \psi_j}{\partial x} + \frac{\partial \psi_i}{\partial y} \frac{\partial \psi_j}{\partial y} \right) d\Omega, \quad R_i = K \int_{\partial \Omega} \psi_i (\nabla \varphi \cdot \bar{n}) dS,$$

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

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

(2)

where $\theta = 0, \frac{1}{2}, 1$ gives the explicit, semi-implicit, and implicit methods, respectively [4]. For other possible time discretizations see [10].

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

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

(3)

$$\frac{d \mathbf{x}}{dt} = \mathbf{u}(\mathbf{x}, t)$$

(4)

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

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

(5)

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

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

(6)

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

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

(7)

where the matrices are defined as in the Eulerian case.
3 Stability Analysis

Giraldo and Neta [3] among others, have discussed the stability and accuracy of the one dimensional advection and advection-diffusion equations. It was shown 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.

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.

3.1 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.
4 Numerical Experiments

Numerical experiments are performed on the two-dimensional advection-diffusion equation. The domain is defined as

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

where

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

\[ \Delta x = \Delta y = 10^5 \quad \text{and} \quad N \text{ is the number of points in the } x \text{ and } y \text{ directions.} \]

The initial wave is centered at

\[ x_0 = x_{\text{min}} + \frac{x_{\text{max}} - x_{\text{min}}}{4}, \quad y_0 = y_{\text{min}} + \frac{y_{\text{max}} - y_{\text{min}}}{2} \]

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

\[ u = +\Omega y \quad \text{and} \quad v = -\Omega x \]  

(8)

with \( \Omega = 10^{-5} \).

4.1 Semi-Lagrangian Interpolation

For the semi-Lagrangian method, four different methods of computing the trajectories are studied. Many others are possible, e.g. Runge-Kutta methods. Exact trajectory calculation is compared with 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 trajectory computation methods use the same interpolator for the departure point.

4.1.1 Exact Trajectories

Using the relations for the Lagrangian trajectories (4) and the velocity (8), we write

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

which can be integrated to yield the equations

\[ x(t) = x_o \cos \Omega t + y_o \sin \Omega t \]

\[ y(t) = -x_o \sin \Omega t + y_o \cos \Omega t \]

where

\[ x_o = x_a \cos \Omega(t + \Delta t) - y_a \sin \Omega(t + \Delta t) \]

\[ y_o = x_a \sin \Omega(t + \Delta t) + y_a \cos \Omega(t + \Delta t) \]

(9)

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

\[ \alpha_1 = x_o[\cos \Omega(t + \Delta t) - \cos \Omega t] + y_o[\sin \Omega(t + \Delta t) - \sin \Omega t] \]

\[ \alpha_2 = -x_o[\sin \Omega(t + \Delta t) - \sin \Omega t] + y_o[\cos \Omega(t + \Delta t) - \cos \Omega t] \]

(10)

(11)
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 $\alpha$ 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

$$\psi_1(\xi) = -\frac{9}{16}(\xi - 1)(\xi^2 - \frac{1}{9}), \quad \psi_2(\xi) = \frac{27}{16}(\xi - \frac{1}{3})(\xi^2 - 1)$$

$$\psi_3(\xi) = -\frac{27}{16}(\xi + \frac{1}{3})(\xi^2 - 1), \quad \psi_4(\xi) = \frac{9}{16}(\xi + 1)(\xi^2 - \frac{1}{9})$$

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

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

where

$$a_0 = u(0), \quad a_1 = \frac{\partial u}{\partial \xi}(0),$$

$$a_2 = 3[u(1) - u(0)] - 2\frac{\partial u}{\partial \xi}(0) - \frac{\partial u}{\partial \xi}(1), \quad a_3 = 2[u(0) - u(1)] + \frac{\partial u}{\partial \xi}(0) + \frac{\partial u}{\partial \xi}(1).$$

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 [2] 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

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

which defines a tridiagonal system that can be solved efficiently.

4.2 2D Advection-Diffusion

The initial condition is given as in [7] by the following exponential function

\[
\varphi_o = 100e^{-\frac{r^2}{4 \Delta x^2}}.
\]

All of the variables are defined as before, the diffusion coefficient \( K \) assumes the values \( 1 \times 10^4 \), \( 5 \times 10^4 \), and \( 7 \times 10^4 \). 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 [7] by

\[
\varphi_{\text{exact}}(x, y, t) = \frac{100}{1 + \frac{Kt}{\Delta x^2}} e^{-\frac{\ddot{x}^2 + \ddot{y}^2}{4 \Delta x^2 + 4Kt}}
\]

where

\[
\ddot{x} = x - x_o \cos \Omega t - y_o \sin \Omega t \quad \text{and} \quad \ddot{y} = y + x_o \sin \Omega t - y_o \cos \Omega t.
\]

Table 1 shows the results for the Eulerian and semi-Lagrangian methods for various values of \( K \) and \( \theta \). The \( \ell_2 \) norm of the error and the first and second moments \( M_1 = \frac{\sum \varphi_{ij}}{\sum \varphi_{\text{exact},ij}} \) and \( M_2 = \frac{\sum \varphi_{ij}^2}{\sum \varphi_{\text{exact},ij}^2} \) are used for comparison. 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. The best solutions are given by the semi-implicit methods (\( \theta = \frac{1}{2} \)). For the lower diffusion coefficient (\( 1 \times 10^4 \)), the semi-implicit Eulerian method is affected by dispersion as is illustrated.
### Table 1: The Eulerian and semi-Lagrangian methods for the advection-diffusion equation.

| Method          | $\theta$ | $\sigma$ | $K \times 10^4$ | $||e||_2$ | $\varphi_{\text{max}}$ | $\varphi_{\text{min}}$ | $M_1$  | $M_2$  |
|-----------------|----------|----------|-----------------|-----------|--------------------------|--------------------------|--------|--------|
| Eulerian        | $1/2$    | $\pi/4$  | 1               | 0.1185    | 55.96                    | -3.47                    | 1.000  | 0.989  |
|                 |          |          | 5               | 0.0165    | 23.61                    | 0.00                     | 1.002  | 0.991  |
|                 |          |          | 7               | 0.0117    | 18.21                    | 0.00                     | 1.009  | 0.994  |
|                 | 1        | $\pi/4$  | 1               | 0.3477    | 32.69                    | 0.00                     | 1.000  | 0.555  |
|                 |          |          | 5               | 0.1318    | 17.11                    | 0.00                     | 1.006  | 0.723  |
|                 |          |          | 7               | 0.0974    | 13.97                    | 0.00                     | 1.018  | 0.768  |
|                 | 0        | $\pi/4$  | 1               | 0.7081    | 6.17                     | 0.00                     | 1.070  | 0.117  |
|                 |          |          | 5               | 0.3847    | 5.70                     | 0.00                     | 1.084  | 0.270  |
|                 |          |          | 7               | 0.3155    | 5.50                     | 0.00                     | 1.094  | 0.340  |
| SLM spline      | $1/2$    | $\pi$    | 1               | 0.0341    | 61.86                    | 0.00                     | 1.000  | 1.009  |
|                 |          |          | 5               | 0.0185    | 25.21                    | 0.00                     | 1.003  | 1.021  |
|                 |          |          | 7               | 0.0154    | 19.24                    | 0.00                     | 1.011  | 1.020  |
|                 | 1        | $\pi$    | 1               | 0.0419    | 58.02                    | 0.00                     | 1.000  | 0.976  |
|                 |          |          | 5               | 0.0133    | 23.82                    | 0.00                     | 1.004  | 0.994  |
|                 |          |          | 7               | 0.0124    | 18.42                    | 0.00                     | 1.012  | 1.000  |
|                 | 0        | $\pi$    | 1               | 0.5768    | 12.11                    | 0.00                     | 1.021  | 0.213  |
|                 |          |          | 5               | 0.2596    | 9.40                     | 0.00                     | 1.036  | 0.414  |
|                 |          |          | 7               | 0.2000    | 8.45                     | 0.00                     | 1.047  | 0.490  |

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 $33 \times 33$, and $\sigma = \frac{\pi}{4}$ for the Eulerian methods and $\sigma = \pi$ for the semi-Lagrangian methods.

In figure 1. 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 in [3].

The bottom half of table 1 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 $K$ ($1 \times 10^4$) the solution obtained by the semi-implicit semi-Lagrangian method (figure 2) is better than that obtained by the semi-implicit Eulerian method (figure 1), while using a Courant
Table 2: 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 $33 \times 33$, and $\sigma = \frac{\pi}{2}$, $\pi$ and $\frac{3\pi}{2}$.

| $\sigma$ | $10^4$ | $||e||_2$ | $\phi_{\text{max}}$ | $\phi_{\text{min}}$ | $M_1$ | $M_2$ |
|----------|--------|-----------|----------------------|----------------------|-------|-------|
| $\pi/2$  | 1      | 0.0292    | 58.17                | 0.00                 | 1.00  | 0.983 |
|          | 5      | 0.0109    | 24.90                | 0.00                 | 1.003 | 1.017 |
|          | 7      | 0.0108    | 19.11                | 0.00                 | 1.011 | 1.018 |
| $\pi$    | 1      | 0.0341    | 61.86                | 0.00                 | 1.000 | 1.009 |
|          | 5      | 0.0185    | 25.21                | 0.00                 | 1.003 | 1.021 |
|          | 7      | 0.0154    | 19.24                | 0.00                 | 1.011 | 1.020 |
| $3\pi/2$ | 1      | 0.0818    | 62.18                | 0.00                 | 1.000 | 1.016 |
|          | 5      | 0.0352    | 25.21                | 0.00                 | 1.004 | 1.024 |
|          | 7      | 0.0276    | 19.24                | 0.00                 | 1.011 | 1.023 |

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 $7 \times 10^4$ 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 $\sigma = \frac{3\pi}{2}$ (table 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.

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 2. 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 3 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.
Table 3: 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 $33 \times 33$ and $\sigma = \pi$.

5 Conclusions

A family of Eulerian and semi-Lagrangian finite element methods were presented. 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. 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.

Numerical experiments were performed on the two-dimensional advection-diffusion equation and the results demonstrate the superiority of the semi-implicit semi-Lagrangian method over the semi-implicit Eulerian method.
Figure 1: 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 $33 \times 33$ and $\sigma = \frac{\pi}{4}$.

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.

Acknowledgements

The first author would like to thank the National Research Council for
Figure 2: 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 \( 33 \times 33 \) and \( \sigma = \pi \).

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


