A simple semi-Lagrangian scheme for advection equations

By WEN-YIH SUN*, KAO-SAN YEH and RUEEN-YUH SUN
Purdue University, USA

(Received 19 January 1995; revised 28 December 1995)

SUMMARY

A semi-Lagrangian scheme is proposed by using forward trajectories and a splitting approach to the interpolation procedure. Several polynomial interpolation schemes are tested, including cubic spline and Lagrange polynomials of degree 3, 5 and 7. A simple filter is also proposed to eliminate spurious short waves and to achieve positive-definiteness. Uniform, rotational and Smolarkiewicz’s deformational flows are tested, a solution to the inviscid Burger’s equation is also provided. This new algorithm employing cubic spline interpolation and the new filter yields efficient and accurate short-term simulations. The advantage in efficiency of adopting split trajectories for interpolation is accompanied by a restriction which is slightly more stringent than the stability criterion for conventional semi-Lagrangian advection schemes. It is also found that the simple filter can eliminate spurious short waves very effectively without degrading the solutions.

KEYWORDS: Numerical methods Semi-Lagrangian model Trajectories

1. INTRODUCTION

In search of an efficient and accurate advection scheme for the Purdue mesoscale model (Sun and Chern 1993, 1994; Hsu and Sun 1994), we have tested several existing methods. Sun (1993) introduced a new scheme by combining the Crowley (1968) fourth-order scheme and the Gadd (1978, 1979) scheme. Because of its greater accuracy in phase speed, Sun’s scheme performs better than the original Crowley or Gadd schemes, but the Courant number of Sun’s scheme is still limited by unity, as are other Eulerian schemes. Here, we propose a semi-Lagrangian scheme which is simple, efficient and accurate in the cases we have studied.

Since Wiin-Nielsen’s (1959) paper, many versions of semi-Lagrangian schemes employing backward trajectories have been introduced. Some of them have been discussed in a review article by Staniforth and Côté (1991). Iterations are often required to solve for the departure points of the backward trajectories if the velocity field is not constant, as discussed by Bates (1984), Kuo and Williams (1990), and Bermejo and Staniforth (1992). Since the iteration process is time-consuming, we propose using forward trajectories instead of backward trajectories. For further efficiency, we propose a splitting approach to the interpolation procedure as described in section 2(b).

The simplest way to achieve positive-definiteness is probably to reset the undershoots and overshoots to the global minimum and maximum (Ostiguy and Laprise 1990). This, however, cannot eliminate spurious short waves effectively. Bermejo and Staniforth (1992) propose to suppress spurious short waves with local extrema, but we have found that their method also suppresses realistic long waves. By modifying Bermejo and Staniforth’s method and combining it with Ostiguy and Laprise’s method, we obtain a simple filter which is able to eliminate spurious short waves very effectively without degrading the solutions.

2. ALGORITHMS AND EQUATIONS

For convenience, the points of a regular Cartesian grid will simply be called the regular grid points, an irregularly spaced set of points will be called a set of irregular grid points.

* Corresponding author: Purdue University, Department of Earth and Atmospheric Sciences, 1397 Civil Engineering Bldg., West Lafayette, IN 47907-1397, USA.
(a) **One-dimensional passive advection**

Let us consider the one-dimensional advection equation

$$ \frac{df}{dt} = \frac{\partial f}{\partial t} + u \frac{\partial f}{\partial x} = 0, $$

(1)

where $u = dx/dt$. Assume that we know $f(x, t)$ at all regular grid points at time $t_n$, we wish to obtain values at all regular grid points at time $t_{n+1}$. We shall employ forward trajectories to solve Eq. (1) numerically. Let $\alpha_i$ be the distance that the $i$th particle travels from time $t_n$ to $t_{n+1}$, then

$$ \frac{f(x_i + \alpha_i, t_{n+1}) - f(x_i, t_n)}{\Delta t} = 0, $$

(2)

i.e. the value of $f$ at the arrival point $x_i + \alpha_i$ at time $t_{n+1}$ is just its value at the upstream point $x_i$ at time $t_n$, since $f$ is conserved according to Eq. (1). The values of $f$ at the regular grid points $x_i$ at time $t_{n+1}$ can then be easily obtained by interpolations from the values of $f$ at the irregular grid points $x_i + \alpha_i$. The remaining problem is to determine the displacements $\alpha_i$. Given a velocity field $u(x, t)$, the displacements can be calculated by integration:

$$ \alpha_i = \int_{t_n}^{t_{n+1}} u_i(x, t) \, dt, $$

(3)

where the subscript $i$ indicates that the parcel originated at the $i$th grid at the time $t_n$. When the velocity field is independent of time or is constant following the motion, the trajectories can be obtained exactly. In general, an $O(\Delta t^2)$-accurate extrapolator can be used to obtain sufficiently accurate $[O(\Delta t^2)]$ estimates of the trajectories (Temperton and Staniforth 1987; McDonald and Bates 1987; Staniforth and Côté 1991). We summarize the algorithm as:

(i) Estimate the displacements $\alpha_i$ at all regular grid points $x_i$.
(ii) Calculate $f$ at arrival points $x_i + \alpha_i$ at time $t_{n+1}$ with Eq. (2).
(iii) Evaluate $f$ at all regular grid points $x_i$ at time $t_{n+1}$ using an interpolation formula.

(b) **Two-dimensional passive advection**

A simple way to extend the one-dimensional algorithm to higher dimensional ones is to split the trajectories for interpolation. Consider the two-dimensional advection equation

$$ \frac{df}{dt} = \frac{\partial f}{\partial t} + u \frac{\partial f}{\partial x} + v \frac{\partial f}{\partial y} = 0, $$

(4)

where $u = dx/dt, v = dy/dt$. Let us assume that we know $f(x, y, t)$ at all regular grid points at time $t_n$, we wish to obtain values at all regular grid points at time $t_{n+1}$. Let $\alpha_{ij}$ and $\beta_{ij}$ be the $x$- and $y$-components of the displacement that the $(i, j)$th particle travels from time $t_n$ to $t_{n+1}$, which are estimated with $u$ and $v$, respectively, by the methods described in section 2(a). For convenience, we adopt the hypothetical half-time $t_{n+1/2}$ such that the particle travels from $(x_i, y_j)$ to $(x_i + \alpha_{ij}, y_j)$ in the first half-time step $[t_n, t_{n+1/2}]$ then travels from $(x_i + \alpha_{ij}, y_j)$ to $(x_i + \alpha_{ij}, y_j + \beta_{ij})$ in the second half-time step $[t_{n+1/2}, t_{n+1}]$.

Now in the beginning of the first half-time step $[t_n, t_{n+1/2}]$, i.e. $t_n$, the values of $f$ are given at all regular grid points, we wish to obtain values at all regular grid points at the end of this half-time step, i.e. $t_{n+1/2}$. Notice that all the motions involved in the first half-time step are one-dimensional in the $x$-direction, i.e. we have $v = 0$ for all particles, thus
the two-dimensional advection equation (4) is reduced to the one-dimensional advection equation (1). Consider the set of particles which are located on the line \( y = y_j \) at time \( t_n \), the one-dimensional algorithm described in section 2(a) can be used to obtain the values of \( f \) at all regular grid points on this line at time \( t_n+1/2 \), i.e. to integrate Eq. (1) along an approximate trajectory such that

\[
\frac{f(x_i + \alpha_{ij}, y_j, t_{n+1/2}) - f(x_i, y_j, t_n)}{\Delta t} = 0. \tag{5}
\]

Similarly, the values of \( f \) at the regular grid points on other lines in the \( x \)-direction at time \( t_{n+1/2} \) can be obtained. Thus the values of \( f \) at all regular grid points in the two-dimensional domain are obtained at time \( t_{n+1/2} \).

In the second half-time step \([t_{n+1/2}, t_{n+1}]\), we start with the values of \( f \) at all regular grid points in the two-dimensional domain at time \( t_{n+1/2} \) to obtain values at all regular grid points at time \( t_{n+1} \). We notice that all the motions involved in the second half-time step are one-dimensional in the \( y \)-direction, i.e. we have \( u = 0 \) for all particles, thus the two-dimensional advection equation (4) is reduced to the one-dimensional advection equation

\[
\frac{df}{dt} = \frac{\partial f}{\partial t} + v \frac{\partial f}{\partial y} = 0. \tag{6}
\]

To apply the one-dimensional algorithm, we need to find all the \( y \)-displacements \( \gamma_{ij} \) of the particles which are located at the regular grid points at time \( t_{n+1/2} \). The advection in the \( y \)-direction is then to integrate (6) along an approximate trajectory such that

\[
\frac{f(x_i, y_j + \gamma_{ij}, t_{n+1}) - f(x_i, y_j, t_{n+1/2})}{\Delta t} = 0. \tag{7}
\]

Note that it is not appropriate to calculate the \( y \)-displacements \( \gamma_{ij} \) directly with \( v \), they must be interpolated from the displacements \( \beta_{ij} \) of the particles which were located at the regular grid points \( (x_i, y_j) \) at time \( t_n \) and have been advected to the irregular grid points \( (x_i + \alpha_{ij}, y_j) \) at time \( t_{n+1/2} \), as is illustrated in Fig. 1. Now for each set of particles which are located on a line in the \( y \)-direction, we can apply the one-dimensional algorithm in the \( y \)-direction to evaluate the values of \( f \) at the regular grid points on this line at time \( t_{n+1} \). Consequently, we obtain the values of \( f \) at all regular grid points in the two-dimensional domain at time \( t_{n+1} \). We summarize the two-dimensional algorithm as
(i) Estimate the displacements $\alpha_{ij}$ and $\beta_{ij}$ at all regular grid points $(x_i, y_j)$.

(ii) Evaluate $f$ and $\gamma_{ij}$ at all regular grid points $(x_i, y_j)$ at time $t_{n+1/2}$, using Eq. (5) and the one-dimensional scheme.

(iii) Evaluate $f$ at all regular grid points $(x_i, y_j)$ at time $t_{n+1}$, using Eq. (7) and the one-dimensional scheme.

Note that the displacements $(\alpha_{ij}, \beta_{ij})$ are calculated in an unsplit manner, the scheme is therefore not a time-splitting scheme. We also note that this interpolation procedure is not symmetric, in the sense that different answers will be obtained if the method is applied in the $x$- and then the $y$-direction, rather than in the $y$- and then the $x$-direction. The stability condition on this 'split interpolation method' is slightly more stringent than the condition on the conventional semi-Lagrangian schemes. The standard semi-Lagrangian criterion is that trajectories should not cross, while the condition on the method is that neither the $x$- nor $y$-projections of the trajectories should cross. We note that the stability condition on our method is exactly the same as that of the 'cascade interpolation method' of Purser and Leslie (1991).

(c) A simple short-wave filter

Since the interpolation around areas of sharp gradients often leads to the occurrence of spurious short waves, it is desirable to develop a filter that is able to eliminate the spurious short waves effectively without degrading the solutions. Ouistiguy and Laprise (1990) propose to reset the undershoots and overshoots to the global minimum and maximum at the beginning of the computation for passive advection, and to the global minimum and maximum at the beginning of each time step for forced advection; this method will be referred to as Filter A. Bermejo and Staniforth (1992) propose to suppress spurious short waves with local extrema in each time step; their method will be referred to as Filter B and is described as follows.

Suppose that we wish to determine the value of $f$ at the regular grid point $x_k$ which is located between the irregular data points $x_i + \alpha_i$ and $x_{i+1} + \alpha_{i+1}$, i.e.,

$$x_i + \alpha_i < x_k < x_{i+1} + \alpha_{i+1}.$$  

Define the local minimum and maximum surrounding $x_k$ as

$$f^- = \min\{ f(x_i + \alpha_i), f(x_{i+1} + \alpha_{i+1}) \}$$

$$f^+ = \max\{ f(x_i + \alpha_i), f(x_{i+1} + \alpha_{i+1}) \}. \quad \text{(8)}$$

Filter B is then to reset the value $f(x_k)$ obtained from an interpolation such that

$$f(x_k) = f^- \text{ if } f(x_k) < f^-$$

$$f(x_k) = f^+ \text{ if } f(x_k) > f^+. \quad \text{(9)}$$

Our experiments indicate that Filter A cannot eliminate the spurious waves effectively, while Filter B suppresses realistic long waves as well. Thus we propose to apply Filter B only when the following monotonicity condition holds:

$$\{ f(x_i + \alpha_i) - f(x_{i+1} + \alpha_{i+1}) \} \{ f(x_{i+2} + \alpha_{i+2}) - f(x_{i+1} + \alpha_{i+1}) \} \geq 0, \quad \text{(10)}$$

then apply Filter A for positive-definiteness. This method will be referred to as Filter C. Note that a local overshoot or undershoot, i.e. $f(x_k) > f^-$ or $f(x_k) < f^-$, under the
monotonicity condition (10) implies the existence of an unresolvable short wave whose wavelength is essentially less than two grid intervals, as shown in Fig. 2(a). We also note that condition (10) is adopted to avoid potentially excessive truncation of Filter B. For example, when

\[ \{ f(x_i + \alpha_i) - f(x_{i-1} + \alpha_{i-1}) \} > 0 \text{ and } \{ f(x_{i+2} + \alpha_{i+2}) - f(x_{i+1} + \alpha_{i+1}) \} < 0 \]

we can expect a local maximum in the interval between \( x_i + \alpha_i \) and \( x_{i+1} + \alpha_{i+1} \), as shown in Fig. 2(b). Consequently, local extrema can be represented better with Filter C than with Filter B.

3. Numerical Simulations

Several flows are simulated by the new scheme, including one- and two-dimensional uniform flows, the rotational flow and Smolarkiewicz’s deformational flow (Smolarkiewicz 1982). A numerical solution to the inviscid Burger’s equation is also provided. For convenience, we adopt the following abbreviations:
CSPL: cubic spline,
LAG3: Lagrange polynomials of degree 3,
LAG5: Lagrange polynomials of degree 5,
LAG7: Lagrange polynomials of degree 7.

The total mass and total energy for one- and two-dimensional cases are defined with respect to the initial conditions:

\[
\text{Mass} = \frac{\left\{ \sum_{i=1}^{N} f_i \right\}}{\left\{ \sum_{i=1}^{N} g_i \right\}},
\]

\[
\text{Energy} = \frac{\left\{ \sum_{i=1}^{N} f_i^2 \right\}}{\left\{ \sum_{i=1}^{N} g_i^2 \right\}},
\]

\[
\text{Mass} = \frac{\left\{ \sum_{i=1}^{N} \sum_{j=1}^{M} f_{ij} \right\}}{\left\{ \sum_{i=1}^{N} \sum_{j=1}^{M} g_{ij} \right\}},
\]

\[
\text{Energy} = \frac{\left\{ \sum_{i=1}^{N} \sum_{j=1}^{M} f_{ij}^2 \right\}}{\left\{ \sum_{i=1}^{N} \sum_{j=1}^{M} g_{ij}^2 \right\}},
\]

where \( N \) and \( M \) are the total number of grid intervals in the \( x \)- and \( y \)-direction, respectively, \( \{ f_i \} \) and \( \{ f_{ij} \} \) are the numerical solutions, and \( \{ g_i \} \) and \( \{ g_{ij} \} \) are the initial conditions. When analytic solutions are available, the root-mean-square errors (Error) are defined as usual:

\[
\text{Error} = \left\{ \frac{1}{N} \sum_{i=1}^{N} (f_i - g_i)^2 \right\}^{1/2},
\]

\[
\text{Error} = \left\{ \frac{1}{NM} \sum_{i=1}^{N} \sum_{j=1}^{M} (f_{ij} - g_{ij})^2 \right\}^{1/2},
\]

where \( \{ g_i \} \) and \( \{ g_{ij} \} \) are the analytic solutions. The maximum Courant numbers for one- and two-dimensional cases are defined, respectively, as

\[
\text{CN} = \max\{ |u \Delta t / \Delta x|, \}
\]

\[
\text{CN} = \max\{ |u \Delta t / \Delta x|, |v \Delta t / \Delta y| \}.
\]

We set \( \Delta x = \Delta y = 1 \) for all of our simulations, except for the solutions to the inviscid Burger's equation.

(a) One-dimensional uniform flow

A tri-rectangular wave in a one-dimensional flow with the uniform velocity \( u = 2/3(\text{CN} = 2/3) \) is simulated after one revolution (120 time steps) with periodic boundary conditions. The tri-rectangular wave is composed of a triangle with the base width of 6 \( \Delta x \) atop a rectangular wave with width of 20 \( \Delta x \) in an 80 \( \Delta x \) domain. Statistics for the simulations using CSPL interpolation with and without the filters are shown in Table 1. We see that Filter C is able to achieve positive-definiteness while retaining the maximum very well. Figure 3(a) shows the simulation by the new scheme using CSPL interpolation without any filter. Figure 3(b) shows the simulation using CSPL interpolation with Filter C, we note that the spurious short waves are filtered out effectively without degrading the solutions. Similar results were obtained for using LAG3, LAG5 or LAG7 interpolation.
TABLE 1. One-dimensional uniform flow test for a tri-rectangular wave with Courant number 2/3 after 120 time steps (1 revolution)

<table>
<thead>
<tr>
<th>Interpolation</th>
<th>Filter</th>
<th>Max</th>
<th>Min</th>
<th>Mass (%)</th>
<th>Error</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cubic spline</td>
<td>A</td>
<td>0.904</td>
<td>-0.042</td>
<td>100.0</td>
<td>0.040</td>
</tr>
<tr>
<td>Cubic spline</td>
<td>B</td>
<td>0.904</td>
<td>0</td>
<td>102.3</td>
<td>0.042</td>
</tr>
<tr>
<td>Cubic spline</td>
<td>C</td>
<td>0.785</td>
<td>0</td>
<td>98.7</td>
<td>0.053</td>
</tr>
<tr>
<td>Cubic spline</td>
<td></td>
<td>0.902</td>
<td>0</td>
<td>100.9</td>
<td>0.045</td>
</tr>
</tbody>
</table>

See text for explanation of filter.

Figure 3. A simulation (solid lines) of a tri-rectangular wave in a uniform flow after 120 time steps (one revolution) with Courant number = 2/3 by the new scheme using cubic-spline interpolation. The analytical solution is plotted with dashed lines. (a) Without any filter. (b) With Filter C (see text).

(b) Inviscid Burger's equation

For the inviscid Burger's equation

\[ \frac{du}{dt} = \frac{\partial u}{\partial t} + u \frac{\partial u}{\partial x} = 0, \]  

with the initial condition

\[ u(x, 0) = U - \tan^{-1}(x), -\infty < x < \infty, \]  

where \( U \) is the velocity of the mean flow, we have the analytic solution

\[ u(x, t) = U - \tan^{-1}(x - ut) \]  

(Kuo and Williams 1990). A numerical solution for \( U = 0.5 \) at the collapse time \( t = 1 \) using CSPL interpolation with Filter C is shown in Fig. 4, the analytic solution almost coincides with the numerical solution. The domain is of 32 \( \Delta x \) with \( \Delta x = 1/16 \) and the time step is \( \Delta t = 1/8 \) (8 time steps, \( \text{CN} = 3.3 \)).

(c) Two-dimensional uniform flow

A square block in a two-dimensional flow with the uniform velocity \( u = v = 0.2 \) (\( \text{CN} = 0.2 \)) is simulated after 800 time steps (four revolutions) with periodic boundary
conditions. The square block is of unit amplitude and is 7 $\Delta x$ wide on each side in a domain of dimension $40 \times 40 \Delta x^2$. The statistics are shown in Table 2, and the simulation using CSPL interpolation with Filter C is shown in Fig. 5. We note that the phase and shape are both well preserved. In comparison with those of Sun (1993), we see that the simulations by the new scheme are better than Crowley’s fourth-order ($\text{max} = 1.44$, $\text{min} = -0.185$), Gadd’s ($\text{max} = 1.50$, $\text{min} = -0.36$), and Sun’s New1 ($\text{max} = 1.17$, $\text{min} = -0.187$) or New2 ($\text{max} = 1.30$, $\text{min} = -0.132$) schemes (see Fig. 1 of Sun (1993)).

(d) Rotational flow

A square block of unit amplitude in a rotational flow with the maximum $CN = 1.3$ is simulated after 288 time steps (two revolutions). The square block is 10 $\Delta x$ wide, centred 15 $\Delta x$ from the axis of rotation which is located at the centre of a domain of dimension $60 \times 60 \Delta x^2$. The value of the scalar distribution is fixed at zero on the boundaries, and exact trajectories are used. The statistics are shown in Table 3, and the simulation using CSPL interpolation with Filter C is presented in Fig. 6 which shows only the lower one-quarter of the domain. We note that the diffusion and distortion are small, and the simulations are comparable with those of Williamson and Rasch (1989).
Figure 5. A simulation (solid contours) of a square block in a uniform flow after four revolutions (800 time steps, Courant number = 0.2) using cubic-spline interpolation with Filter C (see text). The analytical solution is plotted with stippled contours.

<table>
<thead>
<tr>
<th>Initial conditions</th>
<th>Interpolation</th>
<th>Filter</th>
<th>Max</th>
<th>Min</th>
<th>Mass (%)</th>
<th>Error</th>
</tr>
</thead>
<tbody>
<tr>
<td>Square block</td>
<td>CSPL</td>
<td>C</td>
<td>1</td>
<td>0</td>
<td>102.2</td>
<td>0.059</td>
</tr>
<tr>
<td>Square block</td>
<td>LAG3</td>
<td>C</td>
<td>1</td>
<td>0</td>
<td>104.5</td>
<td>0.076</td>
</tr>
<tr>
<td>Square block</td>
<td>LAG5</td>
<td>C</td>
<td>1</td>
<td>0</td>
<td>103.2</td>
<td>0.062</td>
</tr>
<tr>
<td>Square block</td>
<td>LAG7</td>
<td>C</td>
<td>0.99</td>
<td>0</td>
<td>102.6</td>
<td>0.056</td>
</tr>
<tr>
<td>Slotted cylinder</td>
<td>CSPL</td>
<td>C</td>
<td>1</td>
<td>0</td>
<td>100.4</td>
<td>0.062</td>
</tr>
<tr>
<td>Slotted cylinder</td>
<td>LAG3</td>
<td>C</td>
<td>1</td>
<td>0</td>
<td>100.9</td>
<td>0.078</td>
</tr>
<tr>
<td>Slotted cylinder</td>
<td>LAG5</td>
<td>C</td>
<td>1</td>
<td>0</td>
<td>100.7</td>
<td>0.065</td>
</tr>
<tr>
<td>Slotted cylinder</td>
<td>LAG7</td>
<td>C</td>
<td>1</td>
<td>0</td>
<td>100.6</td>
<td>0.059</td>
</tr>
</tbody>
</table>

The simulations of the square block are after 288 time steps (two revolutions) with maximum Courant number 1.3. The simulations of the slotted cylinder are after 220 time steps (five revolutions) with maximum Courant number 7.1. See text for explanation of interpolation and filter.

A solid-body rotation of the slotted cylinder of Zalesak (1979) is also tested. The rotation axis is at the centre, and the groove of the slotted cylinder is 6 $\Delta x$ wide in a domain of dimension 100 $\times$ 100 $\Delta x^2$, as shown in Fig. 7. The value of the scalar distribution is fixed at zero on the boundaries, and exact trajectories are used. The statistics of the simulations with Filter C after 220 time steps (five rotations, CN = 7.1) are shown in Table 3, and the simulation using CSPL interpolation with Filter C is shown in Fig. 8. We note that the phase and shape are both well preserved. In comparison with the simulations with traditional semi-Lagrangian schemes by Liu (1993), error = 0.79, max = 1.11, and min = $-0.17$, the proposed method shows better accuracy again.
Figure 6. A simulation (solid contours) of a square block in a rotational flow after two revolutions (288 time steps, Courant number = 1.3) using cubic-spline interpolation with Filter C (see text). The analytical solution is plotted with stippled contours.

Figure 7. The slotted cylinder as an initial condition for a rotational flow.
ADVECTION EQUATIONS

(e) Smolarkiewicz's deformational flow

The deformational flow of Smolarkiewicz (1982) is described by the stream function

$$\psi(x) = A \sin(kx) \cos(ky),$$

where $A = 8$, $k = 4\pi / L$ and $L = 100 \Delta x$ is the width of a square domain. The initial condition is a circular cone of radius 15 $\Delta x$ centred at the centre of the domain, as shown in Fig. 9. Both the short-term and long-term simulations are tested with exact trajectories and periodic boundary conditions. The statistics are shown in Table 4, where 'revolution' is referred to that at the vortex centres with period

$$P_c = 2\pi / (k^2 |A|),$$

which is about 49.736 (Staniforth et al. 1986). Figure 10 shows the simulation by the proposed scheme using CSPL and Filter C after 1.1 revolutions (CN = 1.4, 38 time steps), which corresponds to $T/50$, where $T = 2637.6$ is the final time of integration of Smolarkiewicz (1982). Figure 11 shows the same simulation, but using LAG5 and Filter C. Comparing these simulations to the analytical solution presented by Staniforth et al. (1986), we see that both the phase and the shape are preserved quite well, although the total mass and total energy are not conserved very well. Figure 12 shows the simulation by the proposed scheme using LAG5 but without filter after 4.0 revolutions (CN = 2.0, 100 time steps). We note that the total mass lost is by only 0.6%, but the total energy maintained is only 69.6%. Figure 13 shows the simulation using LAG5 but without filter after 30.0 revolutions (CN = 2.0, 750 time steps). We note that the total mass lost is only 1.1%, but the total energy is increased by 14%. We note that these results are comparable with those presented by Huang (1994), and that the CSPL interpolation is not stable for this test, as mentioned by Huang. The instability of the CSPL interpolation in Smolarkiewicz's deformational flow test is probably due to the unresolvable nature of the analytic solution, as shown by Staniforth et al. (1986).
Figure 9. The stream function and initial condition for the deformational flow.

<table>
<thead>
<tr>
<th>Interpolation</th>
<th>Filter</th>
<th>Maximum Courant number</th>
<th>Number of time steps</th>
<th>Number of revolutions</th>
<th>Max</th>
<th>Min</th>
<th>Mass (%)</th>
<th>Energy (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>CSPL</td>
<td>C</td>
<td>0.7</td>
<td>75</td>
<td>1.1</td>
<td>1</td>
<td>0</td>
<td>129</td>
<td>136</td>
</tr>
<tr>
<td>CSPL</td>
<td>C</td>
<td>1.4</td>
<td>38</td>
<td>1.1</td>
<td>0.893</td>
<td>0</td>
<td>123</td>
<td>128</td>
</tr>
<tr>
<td>LAG3</td>
<td>C</td>
<td>0.7</td>
<td>75</td>
<td>1.1</td>
<td>0.754</td>
<td>0</td>
<td>112</td>
<td>104</td>
</tr>
<tr>
<td>LAG3</td>
<td>C</td>
<td>1.4</td>
<td>38</td>
<td>1.1</td>
<td>0.757</td>
<td>0</td>
<td>112</td>
<td>103</td>
</tr>
<tr>
<td>LAG5</td>
<td>C</td>
<td>0.7</td>
<td>75</td>
<td>1.1</td>
<td>0.805</td>
<td>0</td>
<td>116</td>
<td>113</td>
</tr>
<tr>
<td>LAG5</td>
<td>C</td>
<td>1.4</td>
<td>38</td>
<td>1.1</td>
<td>0.795</td>
<td>0</td>
<td>115</td>
<td>112</td>
</tr>
<tr>
<td>LAG7</td>
<td>C</td>
<td>0.7</td>
<td>75</td>
<td>1.1</td>
<td>0.816</td>
<td>0</td>
<td>120</td>
<td>119</td>
</tr>
<tr>
<td>LAG7</td>
<td>C</td>
<td>1.4</td>
<td>38</td>
<td>1.1</td>
<td>0.807</td>
<td>0</td>
<td>118</td>
<td>117</td>
</tr>
<tr>
<td>LAG3</td>
<td>C</td>
<td>0.5</td>
<td>400</td>
<td>4.0</td>
<td>0.371</td>
<td>-0.106</td>
<td>98.9</td>
<td>50.6</td>
</tr>
<tr>
<td>LAG3</td>
<td>C</td>
<td>2.0</td>
<td>100</td>
<td>4.0</td>
<td>0.389</td>
<td>-0.096</td>
<td>98.9</td>
<td>50.7</td>
</tr>
<tr>
<td>LAG5</td>
<td>C</td>
<td>0.5</td>
<td>400</td>
<td>4.0</td>
<td>0.468</td>
<td>-0.183</td>
<td>99.4</td>
<td>68.8</td>
</tr>
<tr>
<td>LAG5</td>
<td>C</td>
<td>2.0</td>
<td>100</td>
<td>4.0</td>
<td>0.475</td>
<td>-0.183</td>
<td>99.4</td>
<td>69.6</td>
</tr>
<tr>
<td>LAG7</td>
<td>C</td>
<td>0.5</td>
<td>400</td>
<td>4.0</td>
<td>0.523</td>
<td>-0.245</td>
<td>99.6</td>
<td>87.2</td>
</tr>
<tr>
<td>LAG7</td>
<td>C</td>
<td>2.0</td>
<td>100</td>
<td>4.0</td>
<td>0.519</td>
<td>-0.261</td>
<td>99.6</td>
<td>88.2</td>
</tr>
<tr>
<td>LAG3</td>
<td>C</td>
<td>0.5</td>
<td>3000</td>
<td>30.0</td>
<td>0.285</td>
<td>-0.118</td>
<td>95.3</td>
<td>77.9</td>
</tr>
<tr>
<td>LAG3</td>
<td>C</td>
<td>2.0</td>
<td>750</td>
<td>30.0</td>
<td>0.259</td>
<td>-0.107</td>
<td>95.5</td>
<td>62.4</td>
</tr>
<tr>
<td>LAG5</td>
<td>C</td>
<td>0.5</td>
<td>3000</td>
<td>30.0</td>
<td>0.428</td>
<td>-0.240</td>
<td>98.1</td>
<td>117</td>
</tr>
<tr>
<td>LAG5</td>
<td>C</td>
<td>2.0</td>
<td>750</td>
<td>30.0</td>
<td>0.414</td>
<td>-0.230</td>
<td>97.9</td>
<td>114</td>
</tr>
<tr>
<td>LAG7</td>
<td>C</td>
<td>0.5</td>
<td>3000</td>
<td>30.0</td>
<td>0.531</td>
<td>-0.366</td>
<td>98.7</td>
<td>160</td>
</tr>
<tr>
<td>LAG7</td>
<td>C</td>
<td>2.0</td>
<td>750</td>
<td>30.0</td>
<td>0.561</td>
<td>-0.379</td>
<td>98.7</td>
<td>180</td>
</tr>
</tbody>
</table>

See text for explanation of interpolation and filter.
Figure 10. A simulation of the deformational flow after 1.1 revolutions (38 time steps, Courant number = 1.4) using cubic-spline interpolation with Filter C (see text).

Figure 11. Same as in Fig. 10 but using LAG5 interpolation with Filter C (see text).
Figure 12. A simulation of the deformational flow after four revolutions (100 time steps, Courant number = 2.0) using LAG5 interpolation without filter (see text).

Figure 13. A simulation of the deformational flow after 30 revolutions (750 time steps, Courant number = 2.0) using LAG5 interpolation without filter (see text).
4. Conclusions

From the results of the various tests, we may conclude that the proposed semi-Lagrangian advection scheme employing forward trajectories has several advantages. Exact trajectories can be used for great accuracy when the velocity field is independent of time, or is constant following the motion, and sufficiently accurate estimates of the trajectories can be obtained when the velocity field depends on time. The absence of iterations for the trajectories contributes to the efficiency of the scheme. The simple ‘split interpolation method’ offers an efficient and accurate way of remapping data from Lagrangian to Eulerian grids, forward trajectories are then able to be incorporated into semi-Lagrangian advection schemes. Similar ideas for using forward trajectories have also been developed by Purser and Leslie (1994).

We also show that the interpolation with cubic spline or Lagrange polynomials of degree 5 or 7 with the simple filter proposed here works very well in most of the flows tested in this paper. However, it is also noted that higher-order polynomials are better than the cubic splines to be applied to a very irregular field generated by Smolarkiewicz’s deformational flow due to the unresolvable nature of the analytic solution, as shown by Staniforth et al. (1986).

Acknowledgements

The authors would like to thank Drs H.-C. Kuo and T.-M. Liu at the National Taiwan University, and Dr J.-D. Chern at Purdue University for useful discussions. We deeply appreciate the referees for their very constructive comments. Appreciation also goes to Mr J. Gardner for proofreading of the original typescript. This work was partially funded by the National Science Foundation under grant ATMS-9213730. Additional support from IBM and DOE/Indiana University is also appreciated.

References


Liu, C.-M. 1993 ‘A study of positive definite schemes’. Ph.D. Thesis, National Taiwan University


Ostiguy, L. and Rene Laprise, J. P. 1990 On the positivity of mass in commonly used numerical transport schemes. Atmos.–Ocean, 28, 147–161

W.-Y. SUN, K.-S. YEH and R.-Y. SUN


