# A Successive Substitution Method for the Evaluation of Trajectories Approximating the Parcel Path by a Linear Function of Space and Time

VALDIR INNOCENTINI

*Instituto Nacional de Pesquisas Espaciais, Sao Jose dos Campos, Brazil*

(Manuscript received 25 August 1997, in final form 4 May 1998)

ABSTRACT

Parcel trajectories can be determined by the time integration of the velocity using the Picard successive substitution procedure. However, this scheme can rarely be applied in numerical problems because it implies (in general) an increasing difficulty in solving the analytical solution in each iteration. This strong restriction to its practical use can be overcome by approximating the velocity to a simple function where the analytical integration can be performed as the iteration advances.

The usual procedure adopted in meteorological research is to compute the trajectories using the Petterssen scheme. In this work it is shown that approximating the velocity by a linear time function along the trajectory in the Picard method yields a numerical algorithm formally identical to the Petterssen scheme. An alternative method is proposed following the philosophy of approximating the velocity in the Picard method in order to simplify the integration. In this approach, the parcel path is assumed to be a linear function of space and time, and the velocity along this path at each iteration is computed using an interpolating polynomial. Only one analytical integration is necessary, and the successive integration depends upon the coefficients of the interpolating function adopted for the velocity. For cubic spline, the alternative algorithm can be represented in a suitable vectorial form convenient in designing the numerical code.

The method is tested on idealized situations with high time and space velocity variations using either Lagrangian or Eulerian coordinates. The results for the alternative and Petterssen schemes are similar in the Eulerian coordinate, but the superiority of the alternative scheme is evident in the Lagrangian coordinate.

## 1. Introduction

Determination of air parcel trajectories is a tool frequently employed in meteorology for at least three different purposes: 1) to enrich the description of a parcel motion subject to some kind of dynamic instability, which requires clear and elegant interpretation (Kuo et al. 1992; Innocentini and Caetano Neto 1992); 2) to feed models designed to point out regions favorable to the incidence of acid rain, deposition of pollutants, and air quality (Jakobs et al. 1995), with trajectories computed with the wind fields produced by regional models; and 3) to perform advection using the semi-Lagrangian technique, efficiently employed in numerical weather prediction models.

Usually the trajectories must be determined from data, produced by either observations objectively analyzed or numerical models, available in a gridded domain (in space and time). The results depend on the time interval, grid length, and interpolating procedure adopted. The time interval can vary from seconds to hours. For example, the rawinsonde network in Europe and North America has a rich spatial resolution at 12-h intervals; these data blended with products from a numerical model can effectively increase the temporal resolution. On the other hand, the semi-Lagrangian technique used in a numerical model can define a time step on the order of minutes. The impact on the accuracy of the trajectory using forecast products provided by numerical models and rawinsonde data at 6 h and 12 h has been investigated by Draxler (1991).

The great variety of situations and necessities requires quite different numerical approaches; they can vary from the simplest, with velocity constant, to more complex, with the velocity assuming a temporal and spatial dependence. For example, a situation with time step $\Delta t$, grid length $\Delta x$, and maximum velocity $U$ satisfying the relation for the Courant number $C \equiv \Delta t \times U \times (\Delta x)^{-1} \ll 1$ offers reasonable conditions for assuming a constant velocity (in time and space). Cases with $C \approx 1$ require spatial interpolation. Lagrange interpolation of polynomials in space (which can be linear, quadratic, cubic, quartic, etc., depending on the number of grid points values used to define the coefficients) has been proposed in a semi-Lagrangian scheme (McDonald

*Corresponding author address:* Dr. Valdir Innocentini, Instituto Nacional de Pesquisas Espaciais—DCM, Av. dos Astronautas, 1758, Caixa Postal 515, CEP 12227-010, São Jose dos Campos-SP, Brazil. E-mail: valdir@met.inpe.br

1984). Recently, Makar and Karpik (1996) have presented the cubic basis splines as an alternative method for interpolation on the sphere.

Kuo et al. (1985) examined the sensitivity of trajectory computation to spatial and temporal resolution using the North American synoptic network. The available wind fields were first linearly interpolated to a time step of 10 min, and the trajectory integration was performed with a simple explicit forward integration assuming constant velocity during the small time step. This is a reasonable assumption, since they use $\Delta x = 80$ km, resulting in $C \approx 0.2$ for $U = 100$ km h$^{-1}$.

In the semi-Lagrangian integration scheme the trajectory is only an intermediary step, since the final objective is to determine the values of all advected variables at the nodes of the grid mesh. An excellent review of this numerical technique can be found in Staniforth and Côté (1991). As a recent example, the cascade interpolation technique, suggested by Purser and Leslie (1991), and extended later to a mass conservative scheme (Leslie and Purser 1995), has been used in the semi-Lagrangian advection scheme. It has the advantage of small computation cost in the spatial interpolation of the advected field. They also maintain the velocities fixed at the small time step for tracing the parcel pathway, but once the departure point is obtained, higher-order spatial interpolation is evoked to determine the value of the advected variable at its upstream origin. Sun and Yeh (1997) applied the cascade technique in a forward semi-Lagrangian scheme.

Even a temporal interpolation may be necessary when the velocity experiences a substantial time variation. A time dependence for the wind speed is assumed by Maryon and Heasman (1988). They apply a trajectory model to the winds provided each 6-h period by the U.K. Meteorological Office operational numerical models. They obtain an analytical formulation of the velocity by linear interpolation in space and time. Although the interpolation used is very simple, the system of differential equations is rather complex to solve analytically. The solution is obtained using the Runge–Kutta integration technique.

Seibert (1993) examines the convergence and accuracy of trajectories, in several cases with known analytical solutions, performed with the iterative schemes proposed by Petterssen (1940) and Pudykiewicz et al. (1985). The Petterssen scheme avoids the analytical difficulty with the time integral using a second-order implicit quadrature formula, which results in a fixed-point problem solved by iterative substitution. The velocities are given at discrete time steps, but are spatially continuous. In practical applications the accuracy of an algorithm rests upon both the spatial and temporal assumption for the velocity, since the variables are commonly available only at discrete nodes; some spatial interpolating procedure must be adopted in order to compute the velocities at the extremities of the path, which are not, in general, coincident with the grid points.

This research proposes an alternative scheme based on the Petterssen scheme. The method replaces the linear time dependence of the velocity along the trajectory assumed in the Petterssen algorithm by an approach that takes into account the spatial interpolating polynomial as the parcel travels along the trajectory. Section 2 describes the trajectory problem, the Picard, the Petterssen, and the alternative substitution algorithms. The alternative method with the bicubic spatial polynomial and some relevant properties are presented in section 3. Section 4 illustrates the method applied to the slotted cylinder advection and to the fast-moving pendulum problems, respectively, a semi-Lagrangian and a Lagrangian scheme. Finally, section 5 resumes the discussion of the practical use of the alternative approach.

## 2. The alternative solution

The trajectory of a parcel, given its initial position $x_0$ at time $t_0$, is the function $x(t)$ satisfying the first-order ordinary differential equation:

$$\frac{dx}{dt} = u(x, t), \qquad x(t_0) = x_0, \tag{1}$$

where $u(x, t)$ is the velocity. The existence and uniqueness of a solution for this problem is demonstrated in texts dealing with ordinary differential equations using the method of successive approximations or Picard theorem (see, e.g., Boyce and DiPrima 1977, p. 72); it defines a convergent sequence of functions where the limit is unique and represents the solution of problem (1). Except in particular cases, the analytical solution of (1) and the sequence of functions defined by the Picard method are difficult to find, and in practical applications quadrature formulas are required.

In meteorological research the Petterssen scheme is usually employed to determine the arrival location of a parcel when the velocity is given at two time steps. In the following it is shown that this scheme can be presented as an approximation of the Picard method. Then, an alternative and more accurate approximation is described based on the same method. To fully understand this new approach, it is convenient to present a brief description of the Picard and Petterssen methods.

### *a. Picard's iteration method*

The initial value problem (1) is equivalent to the integral equation:

$$x(t) = x_0 + \int_{t_0}^{t} u(x(t'), t')\, dt'. \tag{2}$$

Defining a functional L by

$$L[Z](t) \equiv x_0 + \int_{t_0}^{t} u(Z(t'), t')\, dt' \qquad t_0 \le t \le t_1, \quad (3)$$

the solution of (1) or (2) is the function $Z$ satisfying the fixed-point equation:

$$Z(t) = L[Z](t). \quad (4)$$

In the Picard method the function $Z$ is obtained by successive correction of a first-guess trajectory. The first approximation for the trajectory is prescribed with the constant velocity $u_{00} \equiv u(x_0, t_0)$:

$$x^{(1)}(t) = x_0 + (t - t_0)u_{00},$$

where the upper index denotes the iteration. This trajectory is corrected by applying the functional L on $x^{(1)}(t)$:

$$x^{(2)}(t) = L[x^{(1)}](t)$$

and similar for $x^{(3)}(t)$, so an arbitrary $k$th evaluation is given by

$$x^{(k)}(t) = L[x^{(k-1)}](t). \quad (5)$$

The sequence of functions obtained through the iterative procedure can be shown to converge to a unique solution, provided the Lipschitz condition is observed for $u$ (or $u$ is a Lipschitzian function); that is, there exists a constant $\beta$ such that

$$|u(x_N(t), t) - u(x_M(t), t)| \le \beta |x_N(t) - x_M(t)|.$$

In particular for $u$ differentiable, $\beta$ is the maximum of $|\partial u/\partial x|$. The error is given by (see Coddington 1961, p. 213)

$$|x(t) - x^{(k)}(t)| \le \frac{M}{\beta} \frac{(\beta T)^{k+1}}{(k+1)!} \exp(\beta T),$$

where

$$\beta \equiv \max\left|\frac{\partial u}{\partial x}\right|,$$
$$M \equiv \max\{|u(x, t)|\},$$
$$T \equiv (t - t_0).$$

This method, also known as the Picard theorem, can be applied in principle to any differential equation, and by this reason is proof of existence and uniqueness of a solution. It is also a particular application of the fixed-point theorem (see, e.g., Kreider et al. 1968).

It is instructive to examine and to compare the results of this method in a simple case with known analytical solution. Consider the problem

$$\frac{dx}{dt} = xt; \qquad x(0) = 1, \quad (6)$$

where one wants to determine the arrival location at $t = 1$. The analytical solution is $\exp(t^2/2)$, and the Picard method gives

TABLE 1. Numerical solution $x$ of the problem given by Eq. (6) at $t = 1$, as a function of the iteration number. The three methods depicted are described in the text.

| Iteration | Analytic Picard | Petterssen | Alternative |
|---|---|---|---|
| 1 | 1.00000 | 1.00000 | 1.00000 |
| 2 | 1.50000 | 1.50000 | 1.50000 |
| 3 | 1.62500 | 1.75000 | 1.66667 |
| 4 | 1.64583 | 1.87500 | 1.72222 |
| 5 | 1.64844 | 1.93750 | 1.74074 |
| 6 | 1.64870 | 1.96875 | 1.74691 |
| 7 | 1.64872 | 1.98438 | 1.74897 |
| 8 | 1.64872 | 1.99218 | 1.74966 |
| 9 | 1.64872 | 1.99609 | 1.74988 |
| 10 | 1.64872 | 1.99805 | 1.74996 |
| 11 | 1.64872 | 1.99902 | 1.74999 |

$$x^{(k)}(t) = 1 + \frac{t^2}{2} + \frac{1}{2!}\left(\frac{t^2}{2}\right)^2 + \cdots + \frac{1}{k!}\left(\frac{t^2}{2}\right)^k.$$

Table 1 presents the parcel position $x$ at $t = 1$ for each iteration. In the first iteration the velocity from $t = 0$ to $t = 1$ is assumed constant and equal to zero, therefore the parcel stays at the initial position $x = 1$. In the second iteration the velocity is integrated along the previous trajectory given by $x = 1$. The successive substitutions continue, so that in each iteration the arrival point is computed with the velocity along the previous trajectory.

### *b. Approximated iterative solution—The Petterssen algorithm*

The objective of approximated iterative methods is to avoid the analytical work involved in the evaluation of the integral (5), which presents an increasing difficulty as the iteration advances. Petterssen (1940, p. 222) presents a method of successive corrections to compute trajectories based on intuitive arguments. Let $t_0$ and $t_1$ be two time steps and the velocity given by two functions of space $u(x, t_0)$ and $u(x, t_1)$, respectively. First, an arrival position $x_1$ of the parcel at $t = t_1$ is computed assuming a constant velocity $u_{00} = u(x_0, t_0)$. Since the velocity at $x_1$ is $u_{11} = u(x_1, t_1)$, a new arrival position is computed with a constant mean velocity $(u_{00} + u_{11})/2$. Arrival points and mean velocities are successively corrected until a prescribed threshold value for the difference between two successive computations is achieved. Thus, the algorithm may be written by

$$x_1^{(1)} = x_0 + u_{00}\Delta t; \qquad \Delta t \equiv (t_1 - t_0),$$

$$x_1^{(2)} = x_0 + \left(\frac{u_{00} + u_{11}^{(1)}}{2}\right)\Delta t; \qquad u_{11}^{(1)} \equiv u(x_1^{(1)}, t_1)$$

$$\cdots$$

$$x_1^{(k)} = x_0 + \left(\frac{u_{00} + u_{11}^{(k-1)}}{2}\right)\Delta t; \qquad u_{11}^{(k-1)} \equiv u(x_1^{(k-1)}, t_1).$$

Note that the spatial dependence on $x$ of $u(x, t_0)$ is not required in the Petterssen algorithm, in contrast with the alternative method, which will be proposed.

The Petterssen scheme can be posed on more solid arguments, as pointed out by one referee, and following Smolarkiewicz and Pudykiewicz (1992). An implicit second-order trapezoidal quadrature of (2) is

$$x(t_1) = x_0 + \left(\frac{u_{00} + u(x(t_1), t_1)}{2}\right)\Delta t. \tag{7}$$

This is a fixed-point problem for $x(t_1)$, and can be solved by the iterative procedure

$$x^{(k+1)}(t_1) = x_0 + \left(\frac{u_{00} + u(x^{(k)}(t_1), t_1)}{2}\right)\Delta t. \tag{8}$$

Equations (8) and (3) state two fixed-point problems. Although the convergence of both iterative processes is guaranteed by the verification of the Lipschitz condition, the problems and solutions are distinct. It is worthy to note that this method is not based on the trajectory $x(t)$ for $t_0 \le t \le t_1$, but only on the arrival location $x(t_1)$.

The algorithm (8) can be obtained from another point of view, as an approximation of the Picard scheme. Consider the velocity as a linear time interpolating function of the velocity functions at the arrival and departure points; in an arbitrary $k$th iteration this is given by

$$u^{(k)}(t) = \frac{u^{(k)}(t_1)}{\Delta t}(t - t_0) + \frac{u_{00}}{\Delta t}(t_1 - t); \qquad t_0 \le t \le t_1,$$

$$u_{11}^{(k)} \equiv u^{(k)}(t_1) \equiv u(x^{(k)}(t_1), t_1). \tag{9}$$

In this case the Picard iterative algorithm (5) can be analytically solved, and one obtains

$$x^{(k+1)}(t) = x_0 + \frac{a^{(k)}}{2}t^2 + b^{(k)}t - \frac{a^{(k)}}{2}t_0^2 - b^{(k)}t_0,$$

where

$$a^{(k)} = \frac{u_1^{(k)} - u_{00}}{\Delta t}$$

$$b^{(k)} = \frac{u_{00}t_1 - u_1^{(k)}t_0}{\Delta t}.$$

For $t = t_1$, one has exactly Eq. (8). Physically, this approach is assuming that the velocity along a trajectory is linear in time. It must be noted that this is not equivalent to assuming linear interpolation of velocity for a fixed $x$.

The results of this scheme applied to the simple illustrative case proposed in Eq. (6) are provided by Table 1. Although both analytic Picard and Petterssen schemes converge, due to the $t$ linear dependence on the velocity along the trajectory assumed by the latter, the results are not the same.

The Petterssen scheme is fundamentally a form to solve the fixed-point problem, when the integral (2) is approximated by a second-order trapezoidal quadrature formula, by an iterative procedure. However, it is convenient in this research to assume this method as the analytic Picard method with the velocity being linearly interpolated along the trajectory computed in the previous iteration: this (more physical) assumption will supply the heuristic argument of the alternative method proposed in the following discussion.

### c. Alternative approximated iterative solution

One can heuristically argue that an improved solution of the Petterssen scheme could be achieved if in each iteration the parcel could *feel* the actual spatial variation of the velocity. Let $u(x, t_0)$ and $u(x, t_1)$, represented by $u_0(x)$ and $u_1(x)$, be two prescribed velocity fields at $t = t_0$ and $t = t_1$, respectively. A parcel traveling from $t_0$ to $t_1$ must have the velocity close to $u_0(x)$ for $t \to t_0$ and to $u_1(x)$ for $t \to t_1$. Then, one can use the linear interpolation

$$u(x, t) = \frac{u_1(x)}{\Delta t}(t - t_0) + \frac{u_0(x)}{\Delta t}(t_1 - t) \tag{10}$$

(the linear interpolation serves only to illustrate; another temporal interpolation can be evoked, in principle). Consider the trajectory a linear time-dependent function:

$$x(t) = x_0 + \frac{x(t_1) - x_0}{\Delta t}(t - t_0) \tag{11}$$

and (10) becomes a function of only $t$, which defines the velocity following the parcel and allows the analytical solution of integral (2).

The method can be summarized as follows. The first iteration computes the arrival position $x_1^{(1)}$ as in the Petterssen scheme:

$$x_1^{(1)} = x_0 + \int_{t_0}^{t_1} u(x_0, t_0)\, dt = x_0 + u_0\Delta t. \tag{12}$$

The second iteration computes a corrected arrival position $x_1^{(2)}$ by assuming a linear (in $t$) pathway connecting $x_1^{(1)}$ and $x_0$. Thus,

$$x^{(1)}(t) = \frac{\Delta x^{(1)}}{\Delta t}t + \left[x_0 - \frac{\Delta x^{(1)}}{\Delta t}t_0\right] \equiv a^{(1)}t + b^{(1)} \tag{13}$$

with $\Delta x^{(1)} \equiv x^{(1)}(t_1) - x_0$, is substituted into

$$x_1^{(2)} = x_0 + \int_{t_0}^{t_1} u(x^{(1)}(t), t)\, dt, \tag{14}$$

so the integral can be analytically determined. For a general $k$th iteration,

$$x_1^{(k+1)} = x_0 + \int_{t_0}^{t_1} u(x^{(k)}(t), t)\, dt \tag{15a}$$

with

$$x^{(k)}(t) = \frac{\Delta x^{(k)}}{\Delta t}t + \left[x_0 - \frac{\Delta x^{(k)}}{\Delta t}t_0\right] \equiv a^{(k)}t + b^{(k)}. \quad (15b)$$

Supposing $u_0(x)$ and $u_1(x)$ have the usual form employed in spatial interpolations (e.g., sine, cosine, or a polynomial) the integral in (15b) can be analytically evaluated. Equations (15a) and (15b) define a fixed-point problem for $x(t_1)$ and therefore converge provided the Lipschitz condition is satisfied. In summary, the alternative method is performed in two stages. First, the $k$th estimate of the linear trajectory as a function of time is calculated with (15b) using the $k$th arrival position given by (15a) and the known departure location. In the second stage, the $(k + 1)$th arrival position is estimated with (15a), making use of the $k$th trajectory function given by (15b) and a predefined interpolating function of velocity. One must note that the alternative method makes uses of the spline fit for the velocity at $t_0$ and $t_1$, while the Petterssen scheme uses only this spline at $t_1$.

Consider the case

$$u(t_1) = Ax, \qquad u(t_0) = 0,$$

which is reduced to the illustrative case given by (6) when $x_0 = 1$, $t_0 = 0$, $t_1 = 1$, and $A = 1$. The alternative procedure approximates $u(x, t)$ by the relation

$$u(x, t) = A\frac{x}{\Delta t}(t - t_0)$$

with $x(t)$ computed at each iteration. Then

$$x_1^{(1)} = x_0 + u_{00}\Delta t$$

and a general $x_1^{(k+1)}$ is computed with a time linear function for $x^{(k)}(t)$:

$$x_1^{(k+1)} = x_0 + \int_{t_0}^{t_1} u[x^{(k)}(t), t]\, dt,$$

$$x^{(k)}(t) = \frac{x_1^{(k)} - x_0}{\Delta t}t + \left(x_0 - \frac{x_1^{(k)} - x_0}{\Delta t}t_0\right),$$

$$\equiv a^{(k)}t + b^{(k)}$$

$$x_1^{(k+1)} = x_0 + \int_{t_0}^{t_1} \frac{A(a^{(k)}t + b^{(k)})}{\Delta t}(t - t_0)\, dt,$$

$$= x_0 + \frac{A}{\Delta t}\left[\frac{a^{(k)}}{3}t^3 + \left(\frac{b^{(k)}}{2} - \frac{a^{(k)}t_0}{2}\right)t^2 + b^{(k)}t_0 t\right]_{t_0}^{t_1}.$$

The numerical results of the simple example are given in the last column of Table 1. The practical advantage of this method in relation to the Picard algorithm is that the analytical form of the integral is the same for all iterations. In relation to the Petterssen scheme, the advantage is that the spatial variation of the velocities supplied at $t = t_0$ and $t = t_1$ is carried out by the parcel along the trajectory.

The alternative method is based on the Picard substitution scheme with two independent approaches:

(i) temporal interpolation on the velocity fields so that the space and time variables are separated;
(ii) parcel displacement given by a linear time dependence.

The error due to the approximation (i) depends on the interpolation adopted. For practical reasons, the interpolation must result in a simple expression so that the analytical integral in (15b) can be obtained. For a fixed $x$, the Taylor's expansion of $u$ at $t = t_0$ and $t = t_1$ is

$$u_0 = u(t) + (t_0 - t)u'(t) + (t_0 - t)^2u''(t)/2! + \ldots$$

$$u_1 = u(t) + (t_1 - t)u'(t) + (t_1 - t)^2u''(t)/2! + \ldots,$$

and a combination of these expansions gives

$$u(t) = u_0\frac{(t_1 - t)}{(t_1 - t_0)} + u_1\frac{(t - t_0)}{(t_1 - t_0)} + \frac{1}{2}(t_0 - t)(t_1 - t)u''(t) + O(\Delta t^3).$$

If the $(\Delta t)^2$ and higher-order terms are neglected, the linear interpolation results in a first-order accuracy approximation. In the kind of problem that this research is addressing the value of $u$ is given at discrete time steps, and therefore there is no information about the time dependence of $u$ for each fixed $x$. The linear interpolation seems a reasonable assumption that has been used in the past (Smolarkiewicz and Pudykiewicz 1992), or at least, better than a constant in time velocity usually adopted in semi-Lagrangian advection schemes. It is worthwhile to mention that linear time dependence for $u$ at a fixed $x$ is not equivalent to either linear velocity along the displacement (the alternative scheme) or linear displacement (the Petterssen scheme).

The error introduced by the approach (ii) is difficult to examine, and in this research the use of the alternative scheme is motivated by the heuristic arguments given above, the results of Table 1, and the numerical tests carried out in the next sections.

## 3. Two-dimensional applications

Seibert (1993) investigated the accuracy of the Petterssen scheme in rotational and deformational flows. Appendix C shows the equivalence between Petterssen and the alternative schemes in stationary flows with the velocity given by a linear function of $x$ and $y$. Therefore the alternative method will be illustrated in situations where the velocity does not have these properties.

Although the alternative method proposed here can be implemented in conjunction with any interpolating function, provided it can be analytically integrated, in recent years the use of cubic interpolation and cubic basis spline (B-cubic) have been increased due to their extreme accuracy and computational facilities. Makar

and Karpik (1996) discuss these approaches in the context of a semi-Lagrangian advective scheme. Consider a bicubic interpolation for the velocity along $x$ at $t = t_0$ and $t = t_1$:

$$u_0(x, y) = \sum_{i=1}^{4}\sum_{j=1}^{4} c_{0ij}x^{i-1}y^{i-1},$$

$$u_1(x, y) = \sum_{i=1}^{4}\sum_{j=1}^{4} c_{1ij}x^{i-1}y^{i-1}, \qquad (16)$$

where $c_{0ij}$ and $c_{1ij}$ are constants defined by the bicubic interpolation. This formulation can be found in many texts dealing with numerical methods, as, for example, Press et al. (1992). Using a linear time interpolation they can be reduced to a single expression:

$$u(x, y, t) = \sum_{i=1}^{4}\sum_{j=1}^{4} c(t)_{ij}x^{i-1}y^{i-1}, \qquad (17)$$

with

$$c(t)_{ij} = \frac{c_{0ij}(t_1 - t)}{(t_1 - t_0)} + \frac{c_{1ij}(t - t_0)}{(t_1 - t_0)}\Bigg\} \qquad t_0 \le t \le t_1,$$

or

$$c(t)_{ij} = a_{ij}t + b_{ij},$$

with

$$a_{ij} = \frac{c_{1ij} - c_{0ij}}{(t_1 - t_0)}; \qquad b_{ij} = \frac{c_{0ij}t_1 - c_{1ij}t_0}{(t_1 - t_0)};$$

$$c_{0ij} = c(t_0)_{ij}; \quad \text{and} \quad c_{1ij} = c(t_1)_{ij}.$$

The first iteration computes $x_1^{(1)}$ using (12) and a similar expression for $y_1^{(1)}$. Then, with (13) one obtains $a^{(1)}$, $b^{(1)}$, and $x^{(1)}(t)$ and similarly $c^{(1)}$, $d^{(1)}$, and $y^{(1)}(t)$, which are substituted in (17) resulting in

$$x_1^{(2)} = x_0 + \int_{t_0}^{t_1}\left[\sum_{i=1}^{4}\sum_{j=1}^{4}(a_{ij}t + b_{ij})(a^{(1)}t + b^{(1)})^{i-1} \times (c^{(1)}t + d^{(1)})^{j-1}\right] dt.$$

Note that $a_{ij}$ and $b_{ij}$ are determined by the values of $u$ at the grid points (or the coefficients of the polynomial), while $a^{(1)}$, $b^{(1)}$, $c^{(1)}$, and $d^{(1)}$ are dependent on $x^{(1)}$ and $y^{(1)}$ obtained in the previous iteration. This means that only $a^{(k-1)}$, $b^{(k-1)}$, $c^{(k-1)}$, and $d^{(k-1)}$ must be recomputed at the k*th* iteration. Supposing $a^{(1)} \neq 0$, a transformation of variable reduces this expression to

$$x_1^{(2)} = x_0 + \int_{a^{(1)}t_0+b^{(1)}}^{a^{(1)}t_1+b^{(1)}}\left[\sum_{i=1}^{4}\sum_{j=1}^{4}(e_{ij}^{(1)}t + f_{ij}^{(1)})t^{i-1} \times (g^{(1)}t + h^{(1)})^{j-1}\right] dt,$$

where

$$g^{(1)} = \frac{c^{(1)}}{a^{(1)}}; \qquad h^{(1)} = d^{(1)} - \frac{b^{(1)}c^{(1)}}{a^{(1)}};$$

$$e_{ij}^{(1)} = \frac{a_{ij}}{(a^{(1)})^2}; \quad \text{and} \quad f_{ij}^{(1)} = \frac{b_{ij}}{a^{(1)}} - \frac{a_{ij}^{(1)}b_1}{(a^{(1)})^2}.$$

Then

$$x_1^{(2)} = x_0 + \sum_{j=1}^{7}\frac{1}{(j+1)}r_j^{(1)}T^{j+1} + \sum_{j=1}^{7}\frac{1}{j}s_j^{(1)}T^j + C^{(1)},$$

where

$$T = a^{(1)}t_1 + b^{(1)}; \quad \text{and}$$

$$C^{(1)} = -\sum_{j=1}^{7}\frac{1}{j}r_j^{(1)}(a^{(1)}t_0 + b^{(1)})^j - \sum_{j=1}^{7}\frac{1}{(j+1)}s_j^{(1)}(a^{(1)}t_0 + b^{(1)})^{j+1},$$

with $r_j^{(1)}$ and $s_j^{(1)}$ given in appendix A; the index (1) means this value was obtained with $x_1^{(1)}$ and $y_1^{(1)}$ (from the first iteration). If $a^{(1)} = 0$ and $c^{(1)} \neq 0$, the transformation of variable must be done with $c^{(1)}t + d^{(1)}$. The case $a^{(1)} = c^{(1)} = 0$, does not allow any transformation of variable; the integration is easier, resulting in

$$x_1^{(2)} = x_0 + \sum_{i=1}^{4}\sum_{j=1}^{4}\frac{a'_{ij}}{2}(t_1^2 + t_0^2) + b'_{ij}(t_1 - t_0),$$

with

$$a'_{ij} = a_{ij}(b^{(1)}d^{(1)})^{j-1}, \quad \text{and} \quad b'_{ij} = b_{ij}(b^{(1)}d^{(1)})^{j-1}.$$

For the $(k + 1)$th iteration, the algorithm can be summarized in the following manner: (i) compute $a^{(k)}$, $b^{(k)}$, $c^{(k)}$, and $d^{(k)}$ using (15a); (ii) with $c_{0ij}$ and $c_{1ij}$ computed before the first iteration (they are not modified in the forthcoming iterations) a number of operations are performed in order to compute $r_j^{(k)}$ and $s_j^{(k)}$ as given in appendix A; (iii) finally $x_1^{(k+1)}$ and $y_1^{(k+1)}$ are calculated.

Although the computation of the coefficients given above seems very complicated, making the alternative method unreliable, in practical problems once the subroutines have been written, the method can be easily applied. The step-by-step numerical procedure is summarized in appendix B.

## 4. Numerical tests

Although Table 1 constitutes an example revealing the superiority of the alternative approach over the Petterssen scheme in a case with linear time variation of the velocity, it is necessary to test the applicability of the proposed method in more complex situations.

The accuracy of the advection of a solid body solved by a semi-Lagrangian technique is determined by (i) the computation of the trajectory and (ii) the interpolation procedure carried out to evaluate the advected scalar function at the arrival and departure points. Obviously the interpolating procedure is a numerical step inde-

pendent of the interpolation regarding the description of the velocity, which is used to determine the parcel trajectory. For example, Sun and Yeh (1997) recover the value of the advected scalar to the Eulerian grid using a sequence of one-dimensional interpolation, but the Lagrangian grid is determined assuming constant velocities prescribed at the previous time step.

A severe Eulerian (or semi-Lagrangian) test employed in research discussing numerical methods for transport of particles is the slotted cylinder problem (Zalesch 1979; Bermejo 1990). Certainly the success of the test depends on the ability of the continuous function to represent the solid body that will be advected. In the case of a slotted cylinder, there is an abrupt variation of the height of the cylinder and interpolating continuous polynomials tend to create holes and peaks in regions around the sharp spatial gradient. This is the case of the cubic function. Despite this limitation, the cubic spline interpolation will be employed in the cylinder test carried out here, since this interpolation is frequently evoked in meteorology (Purnell 1976; Pudykiewickz and Staniforth 1984; Staniforth and Côté 1991). The main purpose of this test is to show the feasibility of the alternative scheme rather than its accuracy, as justified below in the description of the results.

A Lagrangian test (focused only on the computation of the trajectory) will be considered for the two methods, and the results will be compared. It consists of computation of the trajectory of parcels attached on an oscillating pendulum. It is more illustrative for this research, which is mainly concerned with the trajectory evaluation.

There are remarkable distinctions between the two tests above concerning the numerical procedure. In the first, at each time step the parcel trajectory is computed *from where it comes* (backward), while the second is *to where it goes* (forward). In the numerical code, the first case computes the departure position of the parcel located at each grid point in the given time step. In the second neither the departure nor the arrival locations must be on a grid point. The second test is more effective for highlighting the advantages of the alternative procedure because there is no influence of the interpolation adopted in finding the scalar field at regular grid points (the interpolating polynomial error is eliminated).

In both methods (alternative and Petterssen) the required two-dimensional interpolation (for both the velocity and scalar field in the first test, and only for velocity in the second test) is performed using 16 points, that is, nine grid squares centered around the one where the parcel departure is located. Situations with a great velocity demand a certain caution; if the distance between the arrival position and the boundary of the central grid square is larger than $0.4\Delta x$ (or $0.4\Delta y$), the computation is performed again with the time step divided by two. Such procedure avoids the use of velocity values, in the computation of the polynomial, at grid points far from the parcel trajectory traced during the time step. This procedure is applied to both methods.

### *a. Advection of the slotted cylinder*

A two-dimensional grid is defined by $101 \times 101$ points, with the center at the point (51, 51). At the grid point (26, 51) a cylinder of radius 15 grid points and height $H = 1$ is imposed. A slot is applied in the cylinder, 6 grid points wide and 24 grid points deep, as can be seen in Fig. 1a. The advected velocities are

$$u = -\Omega(J - 51), \qquad v = -\Omega(I - 51), \quad (18)$$

with

$$\Omega = at, \qquad I, J = 1, 2, 3, \ldots, 101,$$

where $u$ and $v$ are the velocities along the $I$ (or $x$) and $J$ (or $y$) axes, respectively. The time necessary to complete one revolution is $2(\pi/a)^{0.5}$ or 100 time steps for $a = 4\pi \times 10^{-4}$ and $\Delta t = 1$ s. At this time the velocity in the most distant border of the cylinder is about five grid lengths per second.

Figures 1b and 1c present the results at 10 and 100 time steps, respectively, for the alternative scheme. In the beginning of the integration peaks and holes are noticeable because the cubic spline is not adequate to represent sharp gradients as sharp as those imposed by the cylinder configuration. This shows that the test is highly contaminated by the error of the interpolating method applied to the scalar field. As the advection proceeds the gradients are smoothed, as can be noted after 100 time steps in Fig. 1c. The result of this figure is very similar to those presented by Bermejo (1990).

The difference between the two schemes after one revolution is very small (not shown). Although the velocity is linear in space, a time variation is imposed in order to avoid a linear steady case, or both schemes would be equivalent, as demonstrated in appendix C. However this experiment (which is very important for showing the feasibility of the alternative method in a test considered severe), cannot emphasize the difference between the two methods because it is inferior to $0.5\Delta x$ at each time step, and the scalar interpolation has great error at the cylinder border, as one can note in Fig. 1b. Therefore, the results are a combination between interpolation and trajectory errors. The distinction between the two methods can be enhanced when the interpolation of the scalar field is eliminated, which is achieved in the next test.

### *b. Fast-moving pendulum*

Consider a pendulum with a motion defined by the angle with a horizontal axis

$$\theta(t) = \frac{\theta_0}{2}\left(\sin(\omega t - \pi/2) + 1\right), \quad (19)$$

and associated angular velocity

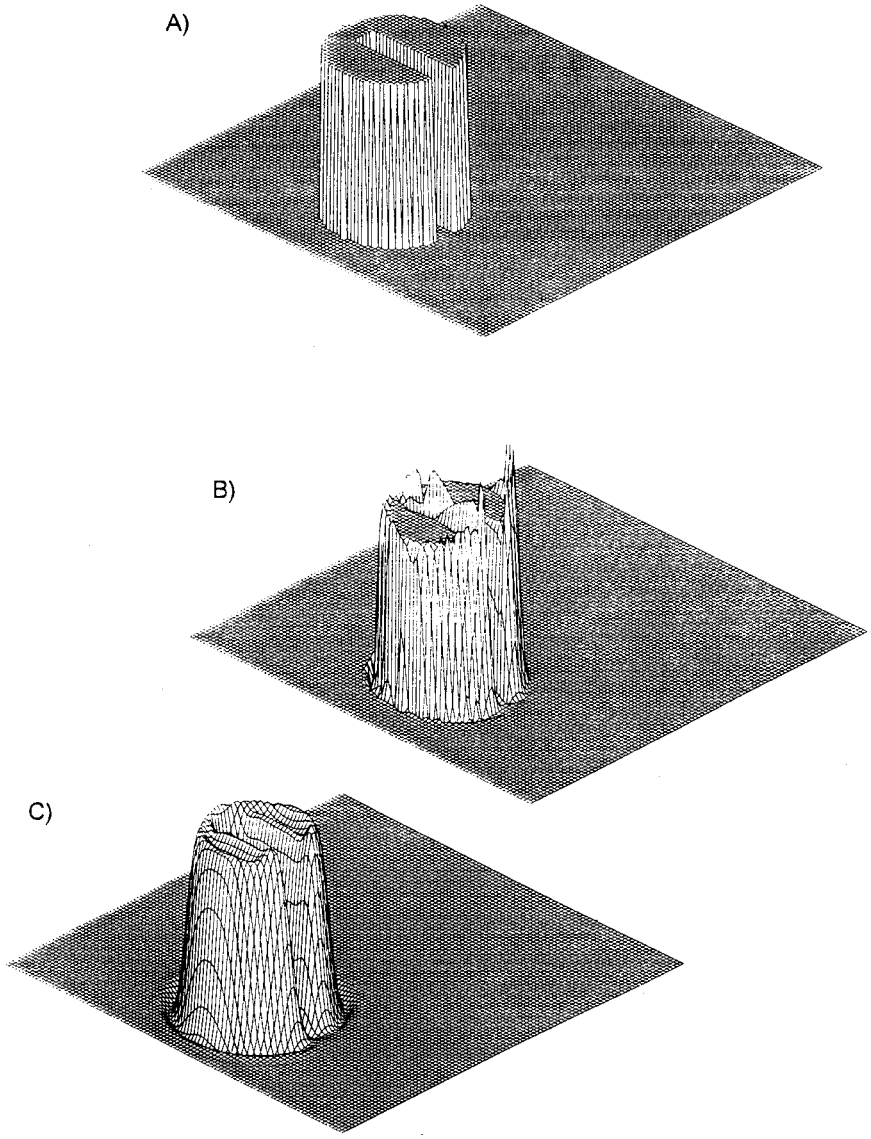


FIG. 1. The advected slotted cylinder at step (a) 0, (b) 10, and (c) 100.

$$\frac{d\theta}{dt} = \Omega = \frac{\theta_0}{2}\omega \cos(\omega t - \pi/2). \qquad (20)$$

The speed varies from 0 at $\theta = 0$, achieves a maximum value at $\theta = \theta_0/2$, and becomes 0 again at $\theta = \theta_0$; then the pendulum starts its backward motion, decreasing its $\theta$ value. From (20) one can note the variation of velocity is controlled by $\theta_0$ and $\omega$.

The experiment illustrated here is carried out with $\omega = 2\pi/10$ and $\theta_0 = \pi/18$. The pendulum, initially along the horizontal axis, is placed on the same grid mesh defined for the slotted cylinder. Its fixed point arises on the center of the domain, that is, at $I = J = 51$. The components of the velocity used to compute the trajectory are given by (18) with $\Omega$ defined by (20). The analytical solution of any parcel attached to the pendulum is given by (19) and its initial distance from the central point. The time step and grid length are the same used in the cylinder experiment. However the advection is Lagrangian in contrast with the semi-Lagrangian advection of the previous test.

The error (defined by the difference with the analytical solution of the distance to the center) of both methods increases with the distance to the center, but always is smaller for the alternative scheme. Figure 2 depicts the error (relative to a grid length) evolution from time step 90 to 100 for the point initially located at 40 grid lengths from the center. Note the error of the Petterssen scheme is never inferior to 0.19 and reaches a maximum of 0.51, while for the alternative method these values are 0.05 and 0.44, respectively.

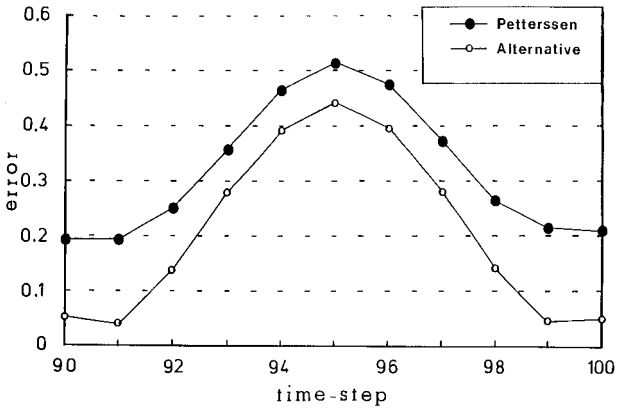


FIG. 2. Difference between the numerically and analytically calculated position (divided by a grid length). The results are plotted for the point initially at $40\Delta x$ from the pendulum fixed point.

## 5. Summary and conclusions

The evaluation of a trajectory requires the time integral of the velocity along the trajectory. The implicit nature of the problem leads us to look for a solution by approximating methods. Given the velocity as a function of space and time, the trajectory can be obtained with Picard's iterative method, represented by a time series with terms corresponding to each iteration. In many cases, Picard's method results in increasing difficulty in solving an integral as the iteration advances. This imposes a strong restriction on its practical applicability, which can be removed by computing the integral using a quadrature formula.

On the other hand, in meteorological problems, usually the velocity is given at discrete time steps. Then the second-order implicit quadrature formulation seems to be an adequate approach to compute the trajectory, since only the velocities at the departure and arrival times must be known. This formulation results in an iterative fixed-point algorithm, or the Petterssen scheme.

In this article we show the formal equivalence between the Petterssen scheme and the Picard method with the velocity, along the path obtained on the previous iteration, approximated by a linear time function.

Although in meteorological problems the velocities are usually available at grid points, their spatial dependence must be specified by some interpolating procedure in the computation of trajectories, because the velocity must be known at the extremities of the path, which rarely lie exactly on the grid points. Then it seems natural to replace the linear time interpolation of the velocity along the path, assumed in the Petterssen scheme, by an approach that does not ignore the spatial interpolating polynomial of the velocity along the previous path in the iterative method. An algorithm satisfying this requirement is proposed in this article. The analytical difficulty is removed approximating the parcel path by a linear time and space function.

The alternative method is proposed by assuming (i) the velocity is a product of a function of space by a function of time, and (ii) a linear time interpolation for the trajectory. These assumptions allow the use of an iterative method. From assumption (i) a *difference method,* defined as one where the time interval is decomposed into $n$ smaller intervals, could be applied in solving the trajectory integral problem, for example, Euler, Runge–Kutta, and predictor–corrector (see, e.g., Gear 1971). In this paper the iterative methodology is adopted, and a comparison with difference method is left to future research.

The formulation of the alternative method applied to the bicubic spatial interpolation can be conveniently represented by vectorial products, thus the numerical algorithm is easily written. Only one analytical integral is necessary. The coefficients of the interpolating polynomial are calculated before the first iteration, and the integral in the forthcoming iterative steps is updated with few additional computations. The proposed method is applied in two problems of trajectories with distinct time integration numerical schemes; one test is semi-Lagrangian and the other Lagrangian.

The semi-Lagrangian scheme is applied to the advection of a slotted cylinder where the new position is found by interpolation from gridpoint values. In the Lagrangian test, the trajectory of a fast-moving pendulum is determined without any interpolation other than the polynomial used to describe the velocity.

The slotted cylinder test provides no evidence that the alternative method has greater accuracy. The results are nearly the same obtained by the Petterssen scheme. It seems that the possible advantages of the alternative scheme are masked by the error of interpolating a scalar with sharp spatial gradient combined with trajectory error. The test with the fast-moving pendulum (which is Lagrangian and where the interpolating error is eliminated) exhibits smaller error in the alternative method.

The main conclusion of this research is that in situations with a high variation of velocity in consecutive time steps, like the case of the fast-moving pendulum, the proposed method reduces the error of the trajectories obtained with the Petterssen scheme. This increased accuracy comes at the price of decreased computational speed; approximately double the CPU time is required in the proposed method. Then the decision of what method must be adopted in a particular application has to be made in light of how much accuracy is necessary. However, in many applications the extra computational cost may not be significant. For example, in regional models, including a large number of chemical reactions, a large fraction of the CPU time is consumed by describing these chemical transformations (Saylor and Ford 1995); then the computational overhead caused by the alternative method in the computation of trajectories

may represent only a minor fraction of the total CPU time.

*Acknowledgments.* The author wishes to express his sincere thanks to Dr. Ernesto dos Santos Caetano Neto for his valuable comments throughout the development of this work. The insightful remarks and suggestions by anonymous reviewers were appreciated and were crucial to improving the paper. I am indebted to the memory of my first teacher, Mr. Onofre Innocentini, who gave me encouragement for all my life. This research was jointly supported by CNPq Grant 30.3832/86.0 and FAPESP Grant 95/4573-5.

## APPENDIX A

### Computation of the Coefficients in the Trajectory Alternative Method Using Bicubic Interpolation

The equality

$$x_1^{(n+1)} = x_0 + \int_{t_0}^{t_1} \left( \sum_{i=1}^{4} \sum_{j=1}^{4} (a_{ij}t + b_{ij})(a^{(n)}t + b^{(n)})^{i-1} \times (c^{(n)}t + d^{(n)})^{j-1} \right) dt \quad \text{(A.1)}$$

can be transformed, for $a^{(n)} \neq 0$, in

$$x_1^{(n+1)} = x_0 + \int_{a^{(n)}t_0+b^{(n)}}^{a^{(n)}t_1+b^{(n)}} \left[ \sum_{i=1}^{4} \sum_{j=1}^{4} \left( \frac{a_{ij}}{a^{(n)}}t + b_{ij} - \frac{a_{ij}b^{(n)}}{a^{(n)}} \right) t^{i-1} \left( \frac{c^{(n)}}{a^{(n)}}t + d^{(n)} - \frac{b^{(n)}c^{(n)}}{a^{(n)}} \right)^{j-1} \right] \frac{dt}{a^{(n)}}$$

$$= x_0 + \int_{a^{(n)}t_0+b^{(n)}}^{a^{(n)}t_1+b^{(n)}} \left[ \sum_{i=1}^{4} \sum_{j=1}^{4} (e_{ij}^{(n)}t + f_{ij}^{(n)})t^{i-1}(g^{(n)}t + h^{(n)})^{j-1} \right] dt \quad \text{(A.2)}$$

with

$$g^{(n)} = \frac{c^{(n)}}{a^{(n)}}; \qquad h^{(n)} = d^{(n)} - \frac{b^{(n)}c^{(n)}}{a^{(n)}}; \quad \text{(A.3a)}$$

$$e_{ij}^{(n)} = \frac{a_{ij}}{(a^{(n)})^2}; \qquad f_{ij}^{(n)} = \frac{b_{ij}}{a^{(n)}} - \frac{a_{ij}b^{(n)}}{(a^{(n)})^2}; \quad \text{(A.3b)}$$

$$a_{ij} = \frac{c_{1ij} - c_{0ij}}{(t_1 - t_0)}; \qquad b_{ij} = \frac{c_{0ij}t_1 - c_{1ij}t_0}{(t_1 - t_0)}; \quad \text{(A.4)}$$

$$a^{(n)} = \frac{x_1^{(n)} - x_0}{(t_1 - t_0)}; \qquad b^{(n)} = \frac{x_0t_1 - x_1^{(n)}t_0}{(t_1 - t_0)}; \quad \text{(A.5a)}$$

$$c^{(n)} = \frac{y_1^{(n)} - y_0}{(t_1 - t_0)}; \quad \text{and} \quad d^{(n)} = \frac{y_0t_1 - y_1^{(n)}t_0}{(t_1 - t_0)}, \quad \text{(A.5b)}$$

where $c_{1ij}$ and $c_{0ij}$ are the coefficients of the bicubic interpolation of the velocity $u(x, y)$ at $t = t_1$ and $t = t_0$, respectively. For $a^{(n)} = 0$ and $c^{(n)} \neq 0$, it is enough to replace $(a^{(n)}t + b^{(n)})$ by $(c^{(n)}t + d^{(n)})$ and vice versa. If $a^{(n)} = c^{(n)} = 0$,

$$x^{(n+1)} = x_0 + \sum_{i=1}^{4} \sum_{j=1}^{4} \frac{a'_{ij}}{2}(t_1^2 + t_0^2) + b'_{ij}(t_1 - t_0) \quad \text{(A.6)}$$

with

$$a'_{ij} = a_{ij}(b^{(n)}d^{(n)})^{j-1} \quad \text{and} \quad b'_{ij} = b_{ij}(b^{(n)}d^{(n)})^{j-1}.$$

Returning to cases with $a^{(n)} \neq 0$ or $c^{(n)} \neq 0$, one can write

$$x_1^{(n+1)} = x_0 + \sum_{j=1}^{7} \frac{1}{(j+1)} r_j^{(n)} T^{j+1} + \sum_{j=1}^{7} \frac{1}{j} s_j^{(n)} T^j + C^{(n)} \quad \text{(A.7)}$$

with

$$T = a^{(n)}t_1 + b^{(n)},$$

where

$$C^{(n)} = -\sum_{j=1}^{7} \frac{1}{(j+1)} r_j^{(n)} (a^{(n)}t_0 + b^{(n)})^{j+1} - \sum_{j=1}^{7} \frac{1}{(j+1)} s_j^{(n)} (a^{(n)}t_0 + b^{(n)})^j \quad \text{(A.8a)}$$

$$r_7^{(n)} = [e_{4,4}g^3] \quad \text{(A.8b)}$$

$$r_6^{(n)} = [e_{4,4}3g^2h + e_{4,3}g^2 + e_{3,4}g^3] \quad \text{(A.8c)}$$

$$r_5^{(n)} = [e_{4,4}3gh^2 + e_{4,3}2gh + e_{4,2}g + e_{3,4}3g^2h + e_{3,3}g^2 + e_{2,4}g^3] \quad \text{(A.8d)}$$

$$r_4^{(n)} = [e_{4,4}h^3 + e_{4,3}h^2 + e_{4,2}h + e_{4,1} + e_{3,4}3gh^2 + e_{3,3}2gh + e_{3,2}g + e_{2,4}3g^2h + e_{2,3}g^2 + e_{1,4}g^3] \quad \text{(A.8e)}$$

$$r_3^{(n)} = [e_{3,4}h^3 + e_{3,3}h^2 + e_{3,2}h + e_{3,1} + e_{2,4}3gh^2 + e_{2,3}2gh + e_{2,2}g + e_{1,4}3g^2h + e_{1,3}g^2] \quad \text{(A.8f)}$$

$$r_2^{(n)} = [e_{2,4}h^3 + e_{2,3}h^2 + e_{2,2}h + e_{2,1} + e_{1,4}3gh^2 + e_{1,3}2gh + e_{1,2}g] \quad \text{(A.8g)}$$

$$r_1^{(n)} = [e_{1,4}h^3 + e_{1,3}h^2 + e_{1,2}h + e_{1,1}] \quad \text{(A.8h)}$$

and $s_j^{(n)}$, $j = 1, \ldots, 7$, have the same formulation of $r_j^{(n)}$, except that $e_{ij}^{(n)}$ for $i, j = 1, \ldots, 4$, are replaced by $f_{ij}^{(n)}$. The $(n)$ has been omitted from the rhs of (A.8) for simplicity.

These expressions can be presented in a concise form making the numerical code easy to write. Defining the vectors

$$\mathbf{e_1} \equiv (e_{1,1}, e_{1,2}, e_{1,3}, e_{1,4}) \quad \text{(A.9a)}$$

$$\mathbf{e_2} \equiv (e_{2,1}, e_{2,2}, e_{2,3}, e_{2,4}) \quad \text{(A.9b)}$$

$$\mathbf{e_3} \equiv (e_{3,1}, e_{3,2}, e_{3,3}, e_{3,4}) \quad \text{(A.9c)}$$

$$\mathbf{e_4} \equiv (e_{4,1}, e_{4,2}, e_{4,3}, e_{4,4}) \quad \text{(A.9d)}$$

$$\mathbf{h} \equiv (1, h, h^2, h^3) \quad \text{(A.9e)}$$

$$\mathbf{gh2} \equiv (0, g, 2gh, 3gh^2) \quad \text{(A.9f)}$$

$$\mathbf{g2h} \equiv (0, 0, g^2, 3g^2h) \quad \text{(A.9g)}$$

$$\mathbf{g3} \equiv (0, 0, 0, g^3) \quad \text{(A.9h)}$$

$$\mathbf{f_1} \equiv (f_{1,1}, f_{1,2}, f_{1,3}, f_{1,4}) \quad \text{(A.9i)}$$

$$\mathbf{f_2} \equiv (f_{2,1}, f_{2,2}, f_{2,3}, f_{2,4}) \quad \text{(A.9j)}$$

$$\mathbf{f_3} \equiv (f_{3,1}, f_{3,2}, f_{3,3}, f_{3,4}) \quad \text{(A.9k)}$$

$$\mathbf{f_4} \equiv (f_{4,1}, f_{4,2}, f_{4,3}, f_{4,4}), \quad \text{(A.9l)}$$

one can write

$$r_1^{(n)} = [\mathbf{e_1} \cdot \mathbf{h}] \quad \text{(A.10a)}$$

$$r_2^{(n)} = [\mathbf{e_2} \cdot \mathbf{h} + \mathbf{e_1} \cdot \mathbf{gh2}] \quad \text{(A.10b)}$$

$$r_3^{(n)} = [\mathbf{e_3} \cdot \mathbf{h} + \mathbf{e_2} \cdot \mathbf{gh2} + \mathbf{e_1} \cdot \mathbf{g2h}] \quad \text{(A.10c)}$$

$$r_4^{(n)} = [\mathbf{e_4} \cdot \mathbf{h} + \mathbf{e_3} \cdot \mathbf{gh2} + \mathbf{e_2} \cdot \mathbf{g2h} + \mathbf{e_1} \cdot \mathbf{g3}] \quad \text{(A.10d)}$$

$$r_5^{(n)} = [\mathbf{e_4} \cdot \mathbf{gh2} + \mathbf{e_3} \cdot \mathbf{g2h} + \mathbf{e_2} \cdot \mathbf{g3}] \quad \text{(A.10e)}$$

$$r_6^{(n)} = [\mathbf{e_4} \cdot \mathbf{g2h} + \mathbf{e_3} \cdot \mathbf{g3}] \quad \text{(A.10f)}$$

$$r_7^{(n)} = [\mathbf{e_3} \cdot \mathbf{g3}]. \quad \text{(A.10g)}$$

The computation of $s_j^{(n)}$, $j = 1, \ldots, 7$ is performed with **f** replacing **e** in (A.10).

## APPENDIX B

### Step-by-Step Numerical Procedure in the Alternative Algorithm

1) Find the 16 grid points forming 9 grid cells, the departure position being within the central cell.
2) Compute the coefficients of the bicubic interpolation for the velocities $u$ and $v$ at $t = t_0$ and $t = t_1$:
   $cu_0$; $cv_{0ij}$; $cu_{1ij}$; $cv_{1ij}$ with $i, j = 1, 2, 3, 4$.
3) Compute with (A.4)
   $au_{ij}$; $bu_{ij}$; $av_{ij}$; $bv_{ij}$, $i, j = 1, 2, 3, 4$.
4) Find $u_0 = u(x_0, y_0, t_0)$ and $v_0 = v(x_0, y_0, t_0)$ using the coefficients obtained in step (2) above.
5) Compute $x_1^{(1)}$ and $y_1^{(1)}$ with Eq. (12).
6) Define $X = x_1^{(1)}$, $Y = y_1^{(1)}$, and $n = 0$.
7) Redefine $n = n + 1$.
8) Compute $a^{(n)}$, $b^{(n)}$, $c^{(n)}$, $d^{(n)}$ using (A.5), replacing $x^{(n)}$ and $y^{(n)}$ by $X$ and $Y$, respectively.
9) If $a^{(n)} = 0$ and $c^{(n)} \neq 0$, replace $a^{(n)}$ and $b^{(n)}$ by $c^{(n)}$ and $d^{(n)}$, respectively.
10) If $a^{(n)} = 0$ and $c^{(n)} = 0$, compute $x_1^{(n+1)}$ with (A.6) and a similar relation for $y_1^{(n+1)}$, and go to step 16 below.
11) Compute $g^{(n)}$, and $h^{(n)}$ using (A.3a).
12) For $au_{ij}$, $bu_{ij}$, using (A.3b), compute
    $e_{ij}^{(n)}$, $f_{ij}^{(n)}$, $i, j = 1, 2, 3, 4$.
13) Using (A.8), (A.9), and (A.10), compute
    $r_i^{(n)}$; $s_i^{(n)}$ and $C^{(n)}$ for $i = 1, 2, \ldots, 7$.
14) Compute $x_1^{(n+1)}$ using (A.7).
15) Repeat steps 12, 13, and 14 with $av_{ij}$ and $bv_{ij}$, and compute $y_1^{(n+1)}$.
16) Compute the difference between two consecutive iterations, and decide if a new iteration will be performed. If so, define $X = x_1^{(n+1)}$, $Y = y_1^{(n+1)}$, and return to step 7.

Observation: if the parcel arrives at a position distanced more than $0.4\Delta x$ from the central cell, the $\Delta t$ is divided by 2, and the computation is carried out in 2 smaller time steps.

## APPENDIX C

### Stationary Linear Flows

Rotational and deformational flows can be represented by

$$u(x, y) = -ry - dx; \qquad v(x, y) = rx + dy,$$

where $r = 0$ for the deformational flow, and $d = 0$ for the rotational flow. They can be generalized by a linear spatial representation in the form

$$u(x, y) = a_u x + b_u y + c_u$$

$$v(x, y) = a_v x + b_v y + c_v$$

where $a$, $b$, and $c$ (with the subscripts $u$ and $v$) are constant coefficients.

The alternative method applied to the linear velocity gives

$$x_1^{(k)} = x_0 + \int_{t_0}^{t_1} (a_u F_x(t) + b_u F_y(t) + c_u)\, dt, \quad \text{(C.1)}$$

where

$$F_x(t) = a^{(k-1)}t + b^{(k-1)}; \qquad a^{(k-1)} = \frac{x^{(k-1)} - x_0}{\Delta t};$$

$$b^{(k-1)} = x_0 - \frac{x_1^{(k-1)} - x_0}{\Delta t}t_0;$$

$$F_y(t) = c^{(k-1)}t + d^{(k-1)}; \qquad c^{(k-1)} = \frac{y_1^{(k-1)} - y_0}{\Delta t};$$

$$d^{(k-1)} = y_0 - \frac{y_1^{(k-1)} - y_0}{\Delta t}t_0,$$

and a similar relation holds for $y_1^{(k)}$. Integrating Eq. (16) and substituting the coefficients, after some manipulations one has

$$x_1^{(k)} = x_0 + [(a_u x_1^{(k-1)} + b_u y_1^{(k-1)} + c_u) + (a_u x_0 + b_u y_0 + c_u)] \times \frac{\Delta t}{2}, \quad \text{(C.2)}$$

which is exactly the Petterssen formulation. Therefore, the two methods are reduced to the same result in stationary linear flows.

## REFERENCES


Bermejo, R., 1990: On the equivalence of semi-Lagrangian schemes and particle-in-cell finite element methods. *Mon. Wea. Rev.,* **118,** 979–987.

Boyce, W. E., and R. C. DiPrima, 1977: *Elementary Differential Equations.* John Wiley and Sons, 451 pp.

Coddington, E. A., 1961: *An Introduction to Ordinary Differential Equations.* Prentice-Hall, 292 pp.

Draxler, R. R., 1991: The accuracy of trajectories during ANATEX calculated using dynamic model analyses versus rawinsonde observations. *J. Appl. Meteor.,* **30,** 1446–1467.

Gear, C. W., 1971: *Numerical Initial Value Problems in Ordinary Differential Equations.* Prentice-Hall, 253 pp.

Innocentini, V., and E. S. Caetano Neto, 1992: A numerical study of the role of humidity in the updraft driven by moist slantwise convection. *J. Atmos. Sci.,* **49,** 1092–1106.

Jakobs, H. J., H. Fieldmann, H. Hass, and M. Memmesheimer, 1995: The use of nested models for air pollution studies: An application of the FURAD model to a SANA episode. *J. Appl. Meteor.,* **34,** 1301–1319.

Kreider, D. L., R. G. Kuller, and D. R. Ostberg, 1968: *Elementary Differential Equations.* Addison-Wesley, 485 pp.

Kuo, Y.-H., M. Skumanich, P. L. Haagenson, and J. S. Chang, 1985: The accuracy of trajectory models as revealed by observing system simulation experiments. *Mon. Wea. Rev.,* **113,** 1852–1867.

——, R. J. Reed, and S. Low-Nam, 1992: Thermal structure and airflow in a model simulation of an occluded marine cyclone. *Mon. Wea. Rev.,* **120,** 2280–2297.

Leslie, L. M., and R. J. Purser, 1995: Three-dimensional mass-conserving semi-Lagrangian scheme employing forward trajectories. *Mon. Wea. Rev.,* **123,** 2551–2566.

Makar, P. A., and S. R. Karpik, 1996: Basis-spline interpolation on the sphere: Applications to semi-Lagrangian advection. *Mon. Wea. Rev.,* **124,** 182–199.

Maryon, R. H., and C. C. Heasman, 1988: The accuracy of plume trajectories forecast using the U. K. Meteorological Office operational forecasting models and their sensitivity to calculation schemes. *Atmos. Environ.,* **22,** 259–272.

McDonald, A., 1984: Accuracy of multiply-upstream, semi-Lagrangian advective schemes. *Mon. Wea. Rev.,* **112,** 1267–1275.

Petterssen, S., 1940: *Motion and Motion Systems.* Vol. I, *Weather Analysis and Forecasting,* McGraw-Hill, 505 pp.

Press, H. P., S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery, 1992: *Numerical Recipes in Fortran.* 2d. ed. Cambridge University Press, 963 pp.

Pudykiewicz, J., and A. Staniforth, 1984: Some properties and comparative performance of the semi-Lagrangian method of Robert in the solution of the advection-diffusion equation. *Atmos.–Ocean,* **22,** 283–308.

——, R. Benoit, and A. Staniforth, 1985: Preliminary results from a partial LRTAP model based on an existing meteorological forecast model. *Atmos.–Ocean,* **23,** 257–303.

Purnell, D. K., 1976: Solution of the advective equation by upstream interpolation with cubic splines. *Mon. Wea. Rev.,* **104,** 42–48.

Purser, R. J., and L. M. Leslie, 1991: An efficient interpolation procedure for high-order three-dimensional semi-Lagrangian models. *Mon. Wea. Rev.,* **119,** 2492–2498.

Saylor, R. D., and G. D. Ford, 1995: On the comparison of numerical methods for the integration of kinetic equations in atmospheric chemistry and transport models. *Atmos. Environ.,* **29,** 2585–2593.

Seibert, P., 1993: Convergence and accuracy of numerical methods for trajectory calculations. *J. Appl. Meteor.,* **32,** 558–566.

Smolarkiewicz, P. K., and J. A. Pudykiewicz, 1992: A class of semi-Lagrangian approximation for fluids. *J. Atmos. Sci.,* **49,** 2082–2096.

Staniforth, A., and J. Côté, 1991: Semi-Lagrangian integration schemes for atmospheric models—A review. *Mon. Wea. Rev.,* **119,** 2206–2223.

Sun, W.-Y., and K.-S. Yeh, 1997: A general semi-Lagrangian advection scheme employing forward trajectories. *Quart. J. Roy. Meteor. Soc.,* **123,** 2463–2476.

Zalesch, S. T., 1979: Fully multidimensional flux-corrected transport. *J. Comput. Phys.,* **31,** 335–352.