Calculating Lagrangian Trajectories Using Time-Dependent Velocity Fields

PEDRO DE VRIES

Royal Netherlands Meteorological Institute, De Bilt, Netherlands

KRISTOFER DÖÖS

Department of Meteorology, Stockholm University, Stockholm, Sweden

(Manuscript received 2 March 2000, in final form 16 October 2000)

ABSTRACT

The authors report on the development of an efficient and accurate tool for computing Lagrangian trajectories using time-varying velocities. By linearly interpolating the velocities both in space and time one can obtain analytical expressions for the trajectory inside each model grid box. In combination with a numerical determination of the transit times through each grid box, these expressions allow for the implementation of expedient algorithms for offline analyses of large datasets resulting from general circulation models. The authors demonstrate the efficacy of this approach on a time-varying two-dimensional model gyre.

1. Introduction

Ocean and atmosphere general circulation models (GCMs) can provide a huge amount of data on velocities, temperature, and other characteristics. A versatile technique to analyse these data and derive further information is to consider Lagrangian trajectories. These trajectories can be calculated using the three-dimensional velocity fields from the GCMs and essentially represent the streamlines of the particular model. Usually, in the case of ocean studies, water masses are distinguished in terms of temperature and salinity. Employing Lagrangian trajectories, water masses can also be distinguished in terms of origin and/or destination and be traced, while its characteristics are continually changing.

Trajectories can be computed online, that is, they are integrated in the GCM at each model time step. A considerable reduction in computational effort can be obtained by employing the model output after the model integration has been finished. Trajectories are then computed offline from these model outputs.

Recently, new tools for calculating and interpreting trajectories have been developed (Döös 1995; Blanke and Raynaud 1997). For stationary velocity fields, it was shown by Blanke and Raynaud (1997) that the trajectory inside one grid box can be given analytically, assuming a linear interpolation of the velocities. Thereby, the need for expensive numerical schemes was relaxed. An entire trajectory then consists of subsequent analytical calculations of positions on the edges of grid boxes. Forward and backward integrations in time are both possible. In this way, very large quantities of trajectories can be generated accurately and efficiently, providing a powerful tool to carry out offline investigations of large datasets from GCMs. In the case of sampled time-dependent velocity fields, the above can also be applied by assuming the velocities to be constant over (part of) the time-sampling interval. The seasonality of the Pacific Equatorial Undercurrent could be treated in this manner (Blanke and Raynaud 1997).

In addition, by assigning certain mass transports to particular trajectories (Döös 1995; Blanke and Raynaud 1997), quantitative measurements of water mass exchanges between different ocean basins or sections can be obtained. For a proper analysis, one generally needs tens of thousand trajectories, which would be impossible to do online in the case of high-resolution GCMs. Within this approach a specific water mass is represented by a large ensemble of smaller water parcels, each representing in an averaged sense a collection of water molecules. Temperature and salinity budgets along trajectories can be computed in order to assess water mass transformations (Drijfhout et al. 1996). Convection can be taken into account by assigning a water parcel a random depth whenever it enters a convectively unstable water column (Döös 1995; Drijfhout et al. 1996). Like the velocities used to calculate the trajectories, these convection events also follow from the ocean GCM.

Here, we develop an extension to the approach fol-

Corresponding author address: Dr. Pedro de Vries, Royal Netherlands Meteorological Institute (KNMI), P.O. Box 201, 3730 AE De Bilt, Netherlands.
E-mail: pvriesde@knmi.nl

lowed by Döös (1995) and Blanke and Raynaud (1997). A method to calculate Lagrangian trajectories from a sampled set of time-varying velocity fields is presented. By simultaneously interpolating linearly in space and time, it is possible to derive analytical solutions for the trajectory inside a grid box. The solutions are such that the transit time through a single grid box must now be determined numerically. The accuracy and efficiency of our method is demonstrated by applying it to a time-dependent two-dimensional model gyre. Hence, powerful algorithms can indeed be devised to calculate trajectories from time-dependent velocity fields.

We start by discussing the case of stationary velocity fields in section 2. The general outline of a trajectory calculation is described. In the next section, we show how to treat the case of time-varying velocity fields. First, the analytical solutions pertaining to linear velocity interpolations in space and time are given. This is followed by a description how to utilize the exact results in order to calculate numerically the appropriate transit time through a single grid box. Finally, our method is illustrated in section 4 by calculating trajectories using the velocity fields from a particular time-varying two-dimensional model gyre. By comparing analytical and numerical trajectories, we investigate the magnitude and behavior of the errors introduced by using the approximation of interpolating velocities linearly in space and time. We also consider some comparisons with Runge–Kutta integration schemes.

2. Stationary velocity fields

We first discuss the case when the velocity fields do not depend on time (Döös 1995; Blanke and Raynaud 1997). An example of such a case is an annual-mean dataset. Velocities are considered to be known on the corners of the grid boxes (''B'' grid) of the particular model. From these velocities, volume transports are derived. The volume transport through the eastern wall of the *ijk* grid box is given by

$$U_{i,j,k} = \frac{1}{2}(u_{i,j,k} + u_{i,j-1,k})\Delta y \Delta z_k, \tag{1}$$

in which *i, j, k* denote the discretized longitude, latitude, and depth, respectively; *u* is the zonal velocity; and $\Delta y \Delta z_k$ defines the meridional–vertical area. Meridional transports are defined analogously, while vertical transports simply follow from the nondivergency of the velocities.

Inside a grid box (see, e.g., Fig. 1), volume transports are obtained by interpolating linearly between the values of the opposite walls. For the zonal direction, for example, using $F_i = U_{i,j,k}$ and $r = x/\Delta x$, one obtains

$$F(r) = F_{i-1} + (r - r_{i-1})(F_i - F_{i-1}). \tag{2}$$

Local transport and position are related by $F = dr/ds$, where the scaled time variable $s = t/(\Delta x \Delta y \Delta z_k)$ the denominator being the volume of the particular grid box.

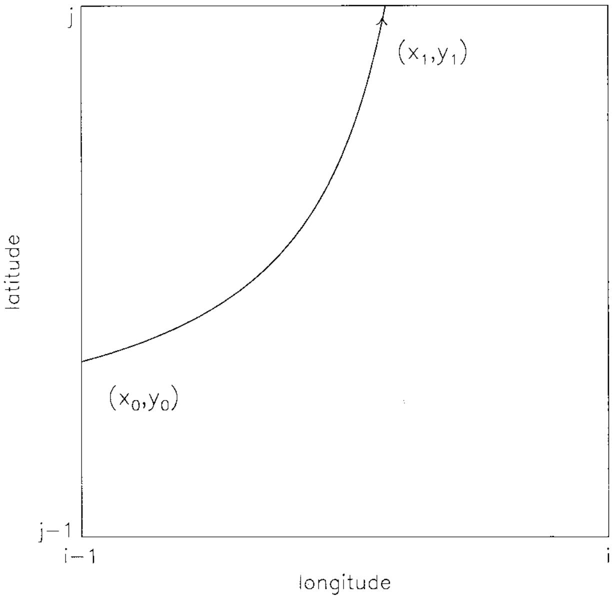

FIG. 1. Illustration of a trajectory [$x(t)$, $y(t)$] through one grid box. The model velocities are defined at the corners of the box.

The approximation (2) can now be written in terms of the following differential equation:

$$\frac{dr}{ds} + \beta r + \delta = 0, \tag{3}$$

with $\beta = F_{i-1} - F_i$ and $\delta = -F_{i-1} - \beta r_{i-1}$. Using the initial condition $r(s_0) = r_0$, the zonal displacement of the trajectory inside the considered grid box can be solved analytically and is given by

$$r(s) = \left(r_0 + \frac{\delta}{\beta}\right)e^{-\beta(s-s_0)} - \frac{\delta}{\beta}. \tag{4}$$

The time s_1 when the trajectory reaches a zonal wall can be determined explicitly:

$$s_1 = s_0 - \frac{1}{\beta}\log\left[\frac{r_1 + \delta/\beta}{r_0 + \delta/\beta}\right], \tag{5}$$

where $r_1 = r(s_1)$ is given by either r_{i-1} or r_i. With the use of (2), the logarithmic factor can be expressed as $\log[F(r_1)/F(r_0)]$. For a trajectory reaching the wall $r = r_i$, for instance, the transport $F(r_1)$ must necessarily be positive, so in order for Eq. (5) to have a solution, the transport $F(r_0)$ must then be positive also. If this is not the case, then the trajectory either reaches the other wall at r_{i-1} or the signs of the transports are such that there is a zero zonal transport somewhere inside the grid box that is reached exponentially slow.

For the meridional and vertical directions, a similar calculation of s_1 is performed determining the meridional and vertical displacements of the trajectory, respectively, inside the considered grid box. The smallest transit time $s_1 - s_0$ and the corresponding r_1 denote at which wall of the grid box the trajectory will exit and move into the adjacent one. The exact displacements in the other two directions are then computed using the smallest s_1 in the corresponding expressions (4). The entire procedure is then repeated for as long as is desired. The above considerations can easily be translated into an efficient numerical algorithm.

Note that interpolations different from Eq. (2) are possible. For example, in a two-dimensional formulation in which the velocities are defined at the grid box corners, the zonal velocity $u(x, y)$ within the *ij* grid box can also be given by

$$u(x, y) = a_0 + a_1(x - x_{i-1}) + a_2(y - y_{i-1}) + a_3(x - x_{i-1})(y - y_{i-1}), \qquad (6)$$

where the four coefficients $\{a_0, a_1, a_2, a_3\}$ can be expressed in terms of the four velocities $\{u_{i-1,j-1}, u_{i,j-1}, u_{i-1,j}, u_{i,j}\}$. The last two terms in (6) represent the linear y dependence of the zonal velocity within that grid box and may be of importance in the case of strongly curved streamlines. This Lagrange interpolation in space is correct up to *linear* order in Δx and Δy. This is also true for the approximation (2), albeit the second-order error may be larger compared to the one corresponding with the use of Eq. (6). To increase the order of accuracy, one should include in the interpolation scheme velocities that lie outside the *ij* grid box. The differential equations that correspond with the interpolation (6) are coupled and nonlinear. It seems impossible to give a general analytical solution for this case. One could optionally drop the nonlinear terms. We continue, however, to make use of (2), since it allows for straightforward, independent analytical solutions in the stationary case and provides an adequate framework to treat time-dependent fields.

3. Time-dependent velocity fields

a. Analytical solutions

The approach presented in the previous section can be extended to the case of time-varying velocity fields. Given a set of velocities $\{\mathbf{v}_n\}$ for each model point, where n represents a discretized time, a linear interpolation of transports in space as well as in time leads to

$$\begin{aligned} F(r, s) = {} & F_{i-1,n-1} + (r - r_{i-1})(F_{i,n-1} - F_{i-1,n-1}) \\ & + \frac{s - s_{n-1}}{\Delta s} \\ & \times [F_{i-1,n} - F_{i-1,n-1} + (r - r_{i-1}) \\ & \times (F_{i,n} - F_{i-1,n} - F_{i,n-1} + F_{i-1,n-1})], \end{aligned} \qquad (7)$$

where, as before, r and s are scaled space and time variables, respectively; $\Delta s = s_n - s_{n-1}$; and i signifies either a longitudinal, meridional, or vertical direction. Connecting the local transport to the time derivative of the position, we get the differential equation

$$\frac{dr}{ds} + \alpha rs + \beta r + \gamma s + \delta = 0, \qquad (8)$$

where the coefficients are defined by

$$\alpha = -\frac{1}{\Delta s}(F_{i,n} - F_{i-1,n} - F_{i,n-1} + F_{i-1,n-1}), \qquad (9)$$

$$\beta = F_{i-1,n-1} - F_{i,n-1} - \alpha s_{n-1}, \qquad (10)$$

$$\gamma = -\frac{1}{\Delta s}(F_{i-1,n} - F_{i-1,n-1}) - \alpha r_{i-1}, \qquad (11)$$

$$\delta = -F_{i-1,n-1} + r_{i-1}(F_{i,n-1} - F_{i-1,n-1}) - \gamma s_{n-1}. \qquad (12)$$

Analytical solutions can be obtained for the following three cases: $\alpha > 0$, $\alpha < 0$, and $\alpha = 0$. Note that inside the grid box, the acceleration, $d^2r/ds^2 = -\alpha r - \gamma$, consists of a constant and an r-dependent term proportional to α. For $\alpha > 0$, the latter term implies a varying deceleration across the grid box. If $\alpha > 0$, we define the timelike variable $\xi = (\beta + \alpha s)/\sqrt{2\alpha}$ and get

$$\begin{aligned} r(s) = {} & \left(r_0 + \frac{\gamma}{\alpha}\right)e^{\xi_0^2-\xi^2} - \frac{\gamma}{\alpha} \\ & + \frac{\beta\gamma - \alpha\delta}{\alpha}\sqrt{\frac{2}{\alpha}}[D(\xi) - e^{\xi_0^2-\xi^2}D(\xi_0)], \end{aligned} \qquad (13)$$

using Dawson's integral $D(\xi) = \exp(-\xi^2) \int_0^\xi \exp(x^2)\, dx$ and the initial condition $r(s_0) = r_0$. Substituting $\alpha < 0$, ξ becomes imaginary. By defining $\zeta = i\xi = (\beta + \alpha s)/\sqrt{-2\alpha}$, Eq. (13) can be re-expressed as

$$\begin{aligned} r(s) = {} & \left(r_0 + \frac{\gamma}{\alpha}\right)e^{\zeta^2-\zeta_0^2} - \frac{\gamma}{\alpha} \\ & - \frac{\beta\gamma - \alpha\delta}{\alpha}\sqrt{\frac{\pi}{-2\alpha}}e^{\zeta^2}[\text{erf}(\zeta) - \text{erf}(\zeta_0)], \end{aligned} \qquad (14)$$

where $\text{erf}(\zeta) = (2/\sqrt{\pi}) \int_0^\zeta \exp(-x^2)\, dx$. The case $\alpha = 0$ will occur occasionally in practice. Its corresponding solution of (8) reads

$$\begin{aligned} r(s) = {} & \left(r_0 + \frac{\delta}{\beta}\right)e^{-\beta(s-s_0)} - \frac{\delta}{\beta} \\ & + \frac{\gamma}{\beta^2}[1 - \beta s + (\beta s_0 - 1)e^{-\beta(s-s_0)}]. \end{aligned} \qquad (15)$$

Compare this with the solution (4) for time-independent velocity fields.

A major difference compared to the time-independent case is that now the transit times $s_1 - s_0$ cannot in general be obtained explicitly. Using the solutions (13)–(15), the relevant root s_1 of

$$r(s_1) - r_1 = 0, \qquad (16)$$

has to be computed numerically for each direction. In the following subsection, we describe how the roots s_1

and the corresponding exiting wall r_1 can be determined. The displacement of the trajectory inside the considered grid box then proceeds as discussed previously for stationary velocity fields.

b. Determination of s_1 and r_1

Here we consider how to determine the roots s_1 of (16) and the corresponding r_1 needed to compute trajectories inside a grid box. In the following, $s_{n-1} \le s_0 < s_n$ and the relevant roots s_1 are to obey $s_0 < s_1 \le s_n$. We also focus on the cases $\alpha > 0$ and $\alpha < 0$, since the considerations below can easily be adapted for $\alpha = 0$. For numerical purposes, we use

$$\frac{\beta\gamma - \alpha\delta}{\alpha} = \frac{F_{i-1,n-1}F_{i,n} - F_{i,n-1}F_{i-1,n}}{F_{i,n} - F_{i-1,n} - F_{i,n-1} + F_{i-1,n-1}}, \quad (17)$$

$$\frac{\gamma}{\alpha} = \frac{F_{i-1,n} - F_{i-1,n-1}}{F_{i,n} - F_{i-1,n} - F_{i,n-1} + F_{i-1,n-1}} - r_{i-1}, \quad (18)$$

$$\xi = \frac{F_{i-1,n-1} - F_{i,n-1} + \alpha(s - s_{n-1})}{\sqrt{2\alpha}}, \quad (19)$$

and similarly for ζ.

The coefficient (17) appearing in the solutions (13) and (14) is exactly zero when either the r_{i-1} or r_{i-1} wall represents land, the transports F_{i-1} or F_i being zero for all n, respectively. In these instances, the opposite wall fixes r_1, and the root $s_1 > s_0$ can then be computed analytically. If there is no solution, we take $s_1 = s_n$. When all three transit times equal s_n, the trajectory will not move into an adjacent grid box but will remain inside the original one. Its new position is subsequently computed, and the next time interval is considered.

If $(\beta\gamma - \alpha\delta)/\alpha \ne 0$, the computation of the roots of Eq. (16) can only be done numerically. This is also true for locating the extrema of the solutions (13) and (14). Alternatively, one can consider $F(r, s) = 0$ using Eq. (7) to analyze where possible extrema are located. It follows that in the *sr* plane, extrema lie on a hyperbola of the form $r = (as + b)/(c + ds)$. Of course, only the parts defined by $s_{n-1} \le s \le s_n$ and $r_{i-1} \le r \le r_i$ are relevant. Depending on which parts of the hyperbola, if any, lie in this ‘‘box’’ and on the initial condition $r(s_0) = r_0$, the trajectory $r(s)$ exhibits none, one, or at most two extrema. In the latter case, the trajectory will not cross either the wall at r_{i-1} or the one at r_i (see Fig. 2 for an example). Hence, those trajectories $r(s)$ determining the transit time $s_1 - s_0$ will have at most one extremum, that is, there is at most one change of sign in the local transport F.

An efficient way to proceed then is as follows. First, consider the wall at r_i. For a trajectory to reach this wall, the local transport must be nonnegative, which depends on the signs of the transports $F_{i,n-1}$ and $F_{i,n}$. Four distinct configurations may arise: A) $F(r_i, s) > 0$ for $s_{n-1} < s < s_n$, B) sign of $F(r_i, s)$ changes from positive to negative at $s = \tilde{s} < s_n$, C) sign of $F(r_i, s)$ changes from negative to positive at $s = \hat{s} < s_n$, D) $F(r_i, s) < 0$ for $s_{n-1} < s < s_n$.

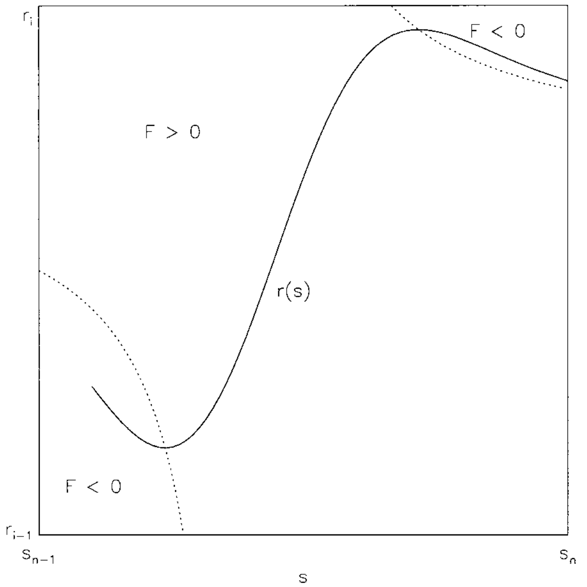

FIG. 2. Example of trajectory $r(s)$ exhibiting two extrema (zero-transport points) inside the relevant *rs* ‘‘box.’’ Regions with positive and negative transports are shown. Extrema for trajectories with differing initial conditions must lie on the hyperbola (dotted curves).

For case A, evaluate $r(s_n)$ using the appropriate analytical solution. If $r(s_n) \ge r_i$, the trajectory has crossed the grid-box wall for $s_1 \le s_n$. If the initial transport $F(r_0, s_0) < 0$, the trajectory may have crossed the opposite wall at an earlier time. The latter is only possible if case C applies for the wall at r_{i-1} and $\hat{s} > s_0$, in which case one checks whether $r(\hat{s}) \le r_{i-1}$. If this is not so, then there is a solution to $r(s_1) - r_1 = 0$ for $r_1 = r_i$ and $s_0 < s_1 \le s_n$. Subsequently, this root can be calculated numerically using a root-solving algorithm. On the other hand, if $r(s_n) < r_i$ or, if applicable, $r(\hat{s}) \le r_{i-1}$ continue with considering the other wall. The arguments for the wall at r_{i-1} are similar to those relating to r_i.

If case B applies and $s_0 < \tilde{s}$, follow the considerations given for case A using $\tilde{s}$ instead of s_n. If there is a root for $r_1 = r_i$, then $s_0 < s_1 \le \tilde{s}$. For case C, follow the considerations given for case A. If there is a root for $r_1 = r_i$, then $\hat{s} < s_1 \le s_n$. For case D, no solution of (16) is possible for $r_1 = r_i$. Turn attention to the other wall.

All these considerations, to be applied to each direction, can be programmed and implemented quite straightforwardly.

4. Application to a two-dimensional gyre

In this section, we present results of Lagrangian trajectory calculations using velocity fields describing a time-varying two-dimensional gyre. Employing a model that allows analytical solutions for the trajectories gives

TABLE 1. Characteristics of trajectories studied. The amplitude a, period T_a, and mean flow strength Ψ_{mean} are given, respectively. The maximum and/or minimal values of the flow strength are $\Psi_{\text{mean}} \times [1 \pm \lambda]$. Typical amplitude deviations $|\Delta a(t)/a|$ and rms values at $t = 200$ yr are also shown (see Figs. 5 and 6).

Amplitude	Period	Mean flow	Typical	rms at
$a/\Delta x$	T_a (years)	Ψ_{mean} (Sv m^{-1})	$\lvert\Delta a(t)\rvert/a$	$t = 200$ yr
9.5	12.5	0.49	14×10^{-4}	0.391
19.5	13.3	0.46	33×10^{-5}	0.095
29.5	14.6	0.42	14×10^{-5}	0.037
39.5	16.8	0.37	8×10^{-5}	0.011

the possibility to test and demonstrate the usefulness of the tools developed in section 3.

The stream function Ψ, defining the gyre, is given by

$$\Psi(x, y, t) = \tilde{\Psi}_\lambda(t) \exp\left\{-\frac{1}{2L^2}[(x - x_0)^2 + (y - y_0)^2]\right\}, \tag{20}$$

where (x_0, y_0) is the stationary center of the gyre and L its size. The explicit time dependence is described by the prefactor

$$\tilde{\Psi}_\lambda(t) = \tilde{\Psi}_0\left[1 + \lambda \cos\left(\frac{2\pi t}{t_0}\right)\right]. \tag{21}$$

The term Ψ defines circular flows with strengths exhibiting a Gaussian dependence on the distance from the gyre center. In addition, these strengths have a sinusoidal time dependence with a period t_0. If $|\lambda|$ is larger than unity, the flows also reverse direction periodically. The time-varying horizontal velocities or transports follow from $-d\Psi/dy$ and $d\Psi/dx$. Note that these transports depend both on x and y in contrast to the ones derived from $\Psi \propto [(x - x_0)^2 + (y - y_0)^2]$. In the latter case, $\alpha = 0$ and $\beta = 0$ for each grid box and direction. The gyre [(20)] obviously allows for a more extensive test of our implemented algorithm.

The trajectories $\mathbf{x}(t) = [x(t), y(t)]$ can be found analytically by employing $d\mathbf{x}/dt = \mathbf{v}$ and read

$$x(t) = x_0 + a \sin\left[\frac{2\pi t}{T_a} + \phi_\lambda(t)\right], \tag{22}$$

$$y(t) = y_0 - a \cos\left[\frac{2\pi t}{T_a} + \phi_\lambda(t)\right], \tag{23}$$

where the time-dependent phase factor ϕ_λ with $T_0 = T_{a=0}$ has the form

$$\phi_\lambda(t) = \phi_0 + \frac{\lambda t_0}{T_0} \sin\left(\frac{2\pi t}{t_0}\right). \tag{24}$$

The initial conditions determine the phase constant ϕ_0 and the amplitude a. For a stationary gyre ($\lambda = 0$), the trajectories describe a periodic circular orbit around the center of the gyre with a period T_a given by

$$T_a = \frac{2\pi L^2}{|\tilde{\Psi}_0|} \exp\left(\frac{a^2}{2L^2}\right). \tag{25}$$

In the case of a time-varying gyre, the trajectories follow the instantaneous velocity and move on a circle with a fixed amplitude, resulting in a nontrivial time-dependent phase behavior of $\mathbf{x}(t)$.

We implemented the gyre [(20)] on a grid having a 100° × 100° flat geometry with a 1° × 1° horizontal resolution and consisting of one vertical layer. The following grid-box dimensions are set: $\Delta x = \Delta y = 111.18$ km. The gyre characteristics are $\Psi_0 = -0.5$ Sv m^{-1} (1 Sverdrup $= 10^6$ m^3 s^{-1}), $L = 50\Delta x$ and $t_0 = 1$ yr. The typical long timescale $\approx T_0$ of the circulation is of the order of 12 to 13 yr. The center of the gyre is at 50° latitude and 50° longitude. We have studied trajectories with initial conditions such that $\phi_0 = 0$ and $a/\Delta x = 9.5$, 19.5, 29.5, and 39.5, respectively. See also Table 1. Lagrangian trajectories using these initial conditions and fields derived from (20) were calculated and compared with the theoretical solutions. The time dependency of the periodic fields was sampled at times $t_n = nt_0/N$ ($n = 0, \ldots, N-1$). Unless stated otherwise, $N = 16$. (Results obtained below do not essentially change when employing instead a set of sampled fields that are averages over time intervals of t_0/N.)

The roots of Eq. (16) are computed numerically using the well-known Newton–Raphson method. Near the root, this method approximately doubles the number of significant digits with each iteration step. Writing $f(z) = r(z) - r_1$ and choosing a starting value z_1, one employs

$$z_{l+1} = z_l - f(z_l)\left[\frac{df(z_l)}{dz}\right]^{-1}. \tag{26}$$

To ensure convergence for all possible starting values, we use a type of bisection technique whenever the computed iteration value falls outside a time interval bracketing the root. The same is done when large values of $f(z_l)$ are obtained due to the exponential nature of the solutions (13)–(15). Convergence is achieved when $|z_{l+1} - z_l| < \varepsilon(z_{l+1} - s_0)$ giving $s_1 = z_{l+1}$, while the parameter ε controls the relative error in the computed transit time $s_1 - s_0$. Generally, the resulting relative errors are on average one order of magnitude smaller

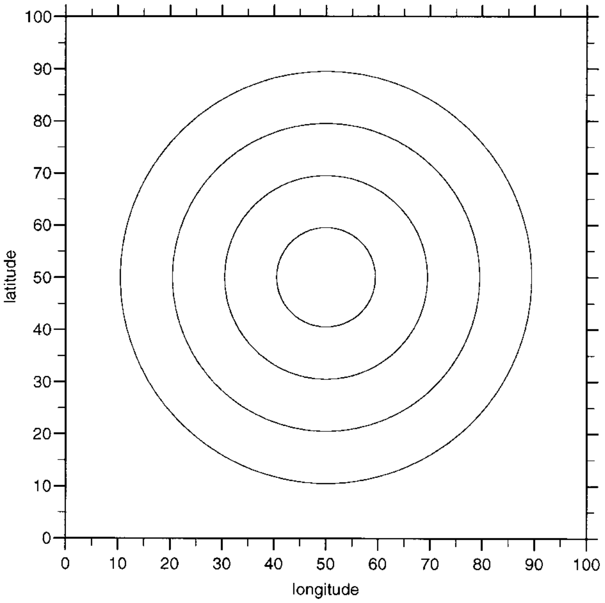

FIG. 3. Computed trajectories [$x(t)$, $y(t)$] for $\lambda = 3.4$ and the four values of a employed. The trajectories are drawn from $t = 0$ to $t = 200$ yr. The contour lines of the stream function [(20)] are also circular. The trajectories remain circular for both the time-independent ($\lambda = 0$) and time-dependent ($\lambda \neq 0$) case, although the motion on the circle is markedly different (see, e.g., Fig. 4).

than the control parameter. Unless stated otherwise, $\varepsilon = 10^{-7}$.

Computed trajectories (x, y) from $t = 0$ to $t = 200$ years are shown in Fig. 3 for $\lambda = 3.4$. It is seen that the computed trajectories remain on a circle with ampitude a. The meridional position y as a function of time is presented in Fig. 4 for $\lambda = 3.4$. Both the analytical and numerically computed values are given. A slow oscillation, described by the period T_a and amplitude a, is superimposed with yearly oscillations. These become larger for larger λ. The computed trajectories do not differ from the exact ones for short times, indicating that the time-sampling interval Δt of about three weeks is very adequate for obtaining convergency. For much larger Δt, deviations at the subyearly scale start to appear, which are larger in the case of the smaller amplitudes a. These deviations are also larger upon increasing λ. At very large times, it is seen that the computed trajectories develop varying phase differences relative to the analytical ones. Trajectories alternatingly dephase and rephase.

To gain some insight into the amplitude and phase deviations of the numerical results $\mathbf{x}_{\text{num}}(t)$, we consider the amplitude defined by $a(t) = \|\mathbf{x}_{\text{num}}(t)\|$ and the overall root-mean-square (rms) error defined by

$$\text{rms}(t) = \|\mathbf{x}_{\text{num}}(t) - \mathbf{x}(t)\|/a. \qquad (27)$$

Linear interpolations in space and time imply a ‘‘local’’ error of the order of $(\Delta x)^2$ and $(\Delta t)^2$. As the trajectories evolve in time, these errors either add up and/or may (partially) compensate each other. In the former case, errors are proportional to the number of time steps and thus grow linearly in time. Smaller errors occur for the trajectories having a smaller curvature (e.g., larger amplitude). This can already be seen in Fig. 4. Also see the discussion on interpolating in space at the end of section 2.

Relative amplitude deviations $\Delta a(t)/a = |a(t) - a|/a$ are shown in Fig. 5 for $a = 9.5\Delta x$ and $\lambda = 3.4$. The behavior is irregular, and the maximum values attained do not grow as a function of time but are constant. Typical (absolute) deviations, which are presented in Table 1, scale as $(1/a)^2$. This is in accordance with a second-order ‘‘local’’ error due to the linear interpolation in space. To a very good approximation, the amplitude is conserved, implying that trajectories follow the instantaneous transports very well (also see Fig. 3). When numerical trajectories are sufficiently close to the analytical ones, amplitude deviations are independent of the value of λ, indicating that these are indeed determined by the errors introduced by the linear interpolation in space. When reducing the accuracy with which the roots of (16) are computed, an additional error comes into play, and the overall deviations increase. For $\varepsilon = 0.1$ e.g., $|\Delta a(t)|/a$ exhibits a linear time dependence corresponding with a slow continuous decrease of $\|\mathbf{x}_{\text{num}}(t)\|$. Such a linear contribution, albeit with a much smaller proportionality constant, also appears at intermediate values of ε when the number of time samplings N is sufficiently lowered.

The rms errors as a function of time are presented in Fig. 6 for $\lambda = 3.4$. The overall error displays a linear contribution superimposed with oscillations with a period of 1 yr. The latter are solely due to the sampling of the time dependency of the fields. Increasing λ or decreasing N will give larger oscillations with the same period. The magnitude of the oscillations roughly scales with $(\Delta t)^2$ or $(1/N)^2$. On the other hand, the linear contributions in Fig. 6 equal the ones when using $\lambda = 0$. Therefore, these originate from interpolating in space, while the oscillating contributions are due to interpolations in time. The rms values at $t = 200$ yr are given in Table 1. Except for the largest amplitude, these scale as $(1/a)^2$. The slope of the linear contributions depends on the initial position within the grid box. When relaxing the time interpolation and computing the trajectories using constant velocities over time intervals of size t_0/N, the rms errors have qualitatively the same behavior as above. In contrast, both the slope of the linear contribution and the magnitude of the superimposed oscillations have increased. For the smallest and largest amplitude, the slope increases by 10% and 400%, respectively. The oscillations are a factor of 10 larger. Thus, applying a linear interpolation in time reduces rms errors in this case.

Since deviations in the amplitude were shown to be very small, the rms error is basically determined by inaccuracies in the phase. Note that for constant am-

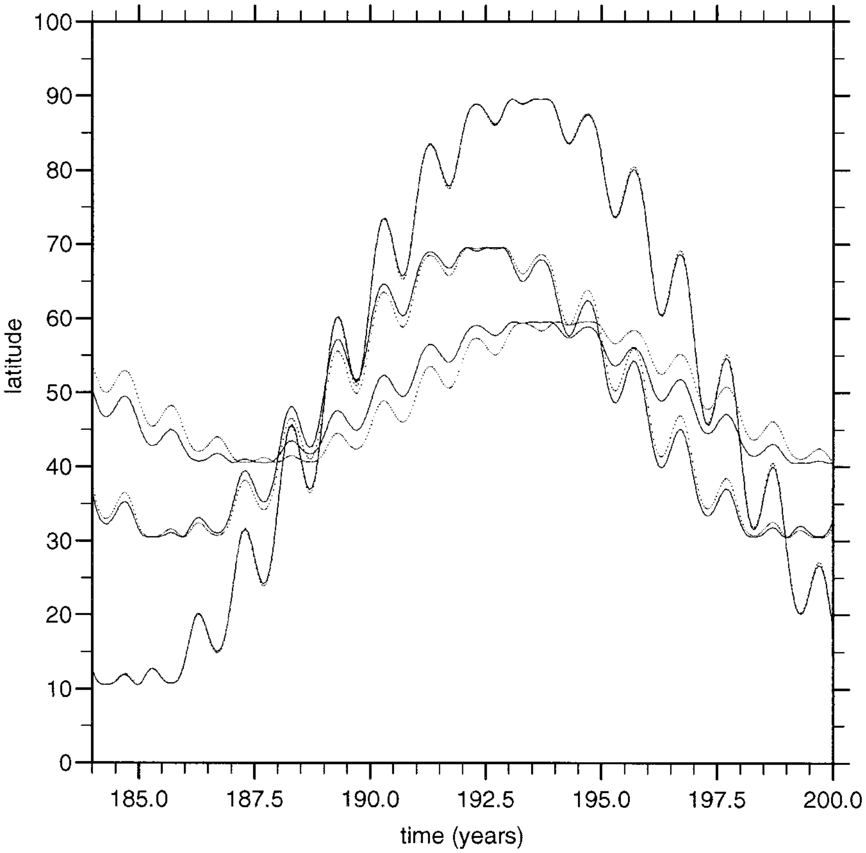

FIG. 4. Meridional position y as a function of time t for $\lambda = 3.4$ and various amplitude values. Drawn (dotted) lines are the analytical (computed) results. The slow oscillation is described by the period T_a. From top to bottom at $t = 184$ yr; $a/\Delta x = 9.5$, 19.5, and 39.5, respectively.

plitude, maximal dephasing ($= \pi$) between computed and analytical trajectories leads to rms = 2. The phase deviations are the largest for the trajectory with the smallest amplitude (also see Fig. 4). Errors introduced due to linear interpolations in time are seen to compensate one another on average. This is due to the fact that for (20), transports periodically increase and decrease while flows remain circular at all times. For other types of flow fields, therefore, interpolations in time may very well contribute significantly to the overall error.

The time reversibility of the numerical code was tested by first integrating trajectories forward in time for a thousand years and then integrating them backward in time. The largest differences between initial and final positions was of the order of $10^{-5}\Delta x$, essentially representing a very small phase difference. The latter shouldn't be too surprising, since the employed flows are very regular, and associated trajectories will therefore not exhibit much chaoticity.

The average number of iterations required to solve 1 root of Eq. (16) numerically was 3 or 4, depending on the configuration of the transports involved (see section 2b). At each time step, there isn't necessarily always a solution for each direction, so that the average number of iterations per time step is lower than 6 to 8. For $\lambda = 3.4$, this number was 2.8. Compared to the case of stationary velocities, the total computational effort increased by a factor of 3. Applying our approach to the more complex case of a three-dimensional seasonal dataset from the OCCAM project (Webb et al. 1997) resulted in a 80% increase in CPU time and an average number of 4.6 iterations per time step for solving the roots.

How does our method compare with Runge–Kutta integration techniques? Let us first point out that numerically solving equations like $dr/ds = F(r, s)$ generally entails two types of approximations. The time derative must be represented such that an accurate time integration can be performed and the transport or velocity $F(r, s)$ must be defined in terms of the sampled data in space and time. A particular choice for the latter will determine the subgrid description of the streamlines. It then remains to choose an algorithm to calculate trajectories that accurately follow these streamlines.

A Runge–Kutta algorithm uses a fourth-order finite difference method. For given spatial resolution Δx and typical velocities v, the time step τ must be small enough so that displacements do not ''overshoot'' in the normal

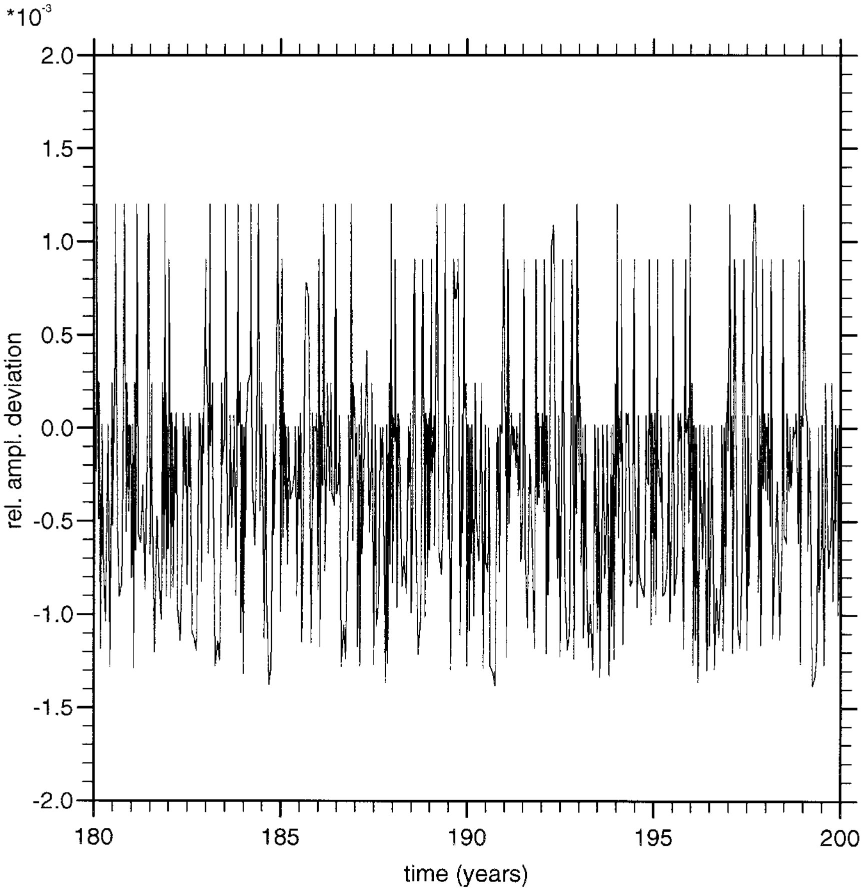

FIG. 5. Relative amplitude deviation $\Delta a/a$ as a function of time t for $\lambda = 3.4$ and $a = 9.5\Delta x$. Typical deviations scale as $(1/a)^2$.

direction for curved trajectories, such as shown in Fig. 1. For trajectories near coasts particles could otherwise end up on land. This can be avoided (Drijfhout et al. 1996; Fujio et al. 1992) at the expense of an increased computational effort. On the other hand, for streamlines that strongly converge on the scale of Δx particles may then cross to another streamline and eventually end up in a wrong region of the model. These considerations apply to the cases of stationary velocities and time-dependent velocities.

Employing a first-order scheme and data from an eddy-resolving ocean model, Bönig and Cox (1988) adaptively scaled the time step such that a gridbox is traversed in n steps: $v\tau = \Delta x/n$. For typical trajectories, they found that $n = 200$ was adequate for their purposes. Drijfhout et al. (1996) determined an appropriate time step by requiring an idealized two-dimensional velocity field that integrated circular trajectories remain unchanged for at least 100 loops. Using a Runge–Kutta scheme, a 5° × 5° model resolution, and a typical velocity $v = 0.1$ m s^{-1}, a time step of a half day ($n \approx 100$) was obtained.

In contrast, our approach is based on analytical time-dependent solutions. No additional code is required for strongly converging streamlines or near land boundaries. Our method is presently restricted to linear interpolations in space and time for representing the subgrid velocities. As shown for the model gyre above, the computed trajectories differ from the analytical ones mostly due to the interpolation in space. We have discussed other interpolations at the end of section 2. The implementation of these interpolations is rather straightforward for Runge–Kutta based algorithms.

To compare CPU performances, we also calculated trajectories using Runge–Kutta schemes in which the velocities are interpolated according to Eqs. (7) and the extension of (6) to first order in Δt. Per time step, the latter required 26% more computational effort compared to the former, but the number of steps n for traversing a grid box was about one order smaller due to being a more accurate interpolation in space for curved streamlines. In the following, we discuss results for the more efficient Runge–Kutta scheme.

To investigate the flow regime determined by the smallest spatial scale, that is, strongly curved streamlines, we chose the trajectory with parameters $\lambda = 3.4$, $a = 2.5\Delta x$, $T_a = 20.2$ days and velocity $v = 2\pi a/T_a = 1$ m s^{-1} [see Eqs. (22) and (23)]. Note that due to

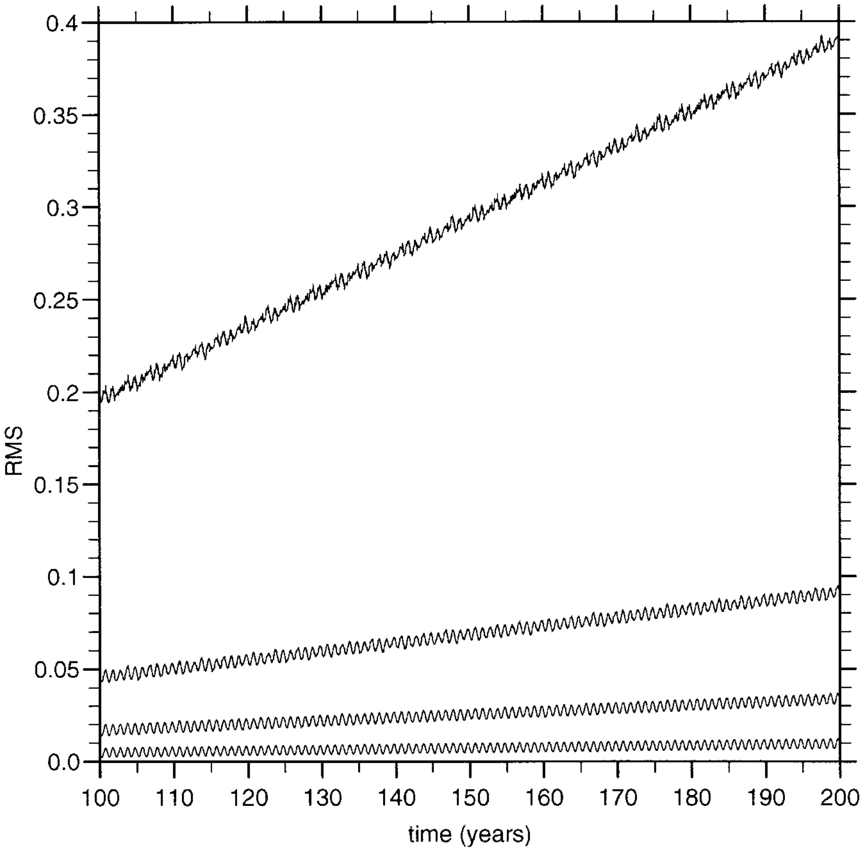

FIG. 6. Root-mean-square (rms) error $\|\Delta\mathbf{x}(t)\|/a$ for computed trajectories as a function of time t for $\lambda = 3.4$ and various amplitude values. From top to bottom: $a/\Delta x = 9.5$, 19.5, 29.5, and 39.5, respectively. Errors are mainly reflected in terms of phase differences between computed and analytical trajectories. Contributions to error: linear contribution is due to interpolation in space; oscillations with yearly period are due to interpolation in time.

the time dependence, the magnitude of the velocity can become smaller and larger [order of $\upsilon(1 + \lambda T_a/T_0)$]. For our method, the average time step to cross a grid box is 0.44 days, with an average number of 4.6 iterations per time step for solving the roots. On an SGI workstation, the required CPU time for 1 million time steps is 35.2 s. Applying the same criterion as used by Drijfhout et al. (1996), we estimated $\tau = 0.05$ days ($n \approx 10$) for the Runge–Kutta method. The CPU time for n million Runge–Kutta time steps measured 36.2 s. It should be noted that for the latter calculations, we did not include any provisions for adapting the time step or to prevent ‘‘overshootings’’ ending up on land. In an application to a GCM, these should be included resulting in a substantial increase of CPU time.

For the four trajectories above (see Table 1), the velocities $\upsilon = 2\pi a/T_a$ are of the order of a few centimeters, since the periods T_a are much larger. For our method, the average traversal time was 10 to 16 days, which is of the order of the sampling time interval. The appropriate time step τ was such that also $n \approx 10$. Thus, also for flow regimes determined by Δt variations, comparable CPU times (not taking into account previously discussed provisions) are found for the Runge–Kutta scheme and the method developed here.

If the latter time variations are relaxed, the time step τ can be increased. Generally, the Runge–Kutta scheme using the first-order Lagrangian spatial interpolation is expected to be more efficient than the presented method in the case of stationary velocity fields with weakly to moderately curved streamlines.

In conclusion, the numerical algorithm developed for calculating trajectories from time-varying fields has been shown to be both accurate and efficient.

5. Summary

We have presented a method to calculate Lagrangian trajectories from time-varying velocity fields. A linear interpolation in space and time allows for the derivation of analytical expressions for the trajectory inside a model grid box. Contrary to the case of stationary fields, the transit time through each grid box must be determined numerically. A description of how to apply and implement the analytical results has been given.

With the use of a particular time-varying two-dimensional model gyre, we have tested our method and investigated the behavior of errors introduced by interpolating linearly in space and time. The errors due to the approximation of interpolating linearly in time mainly average out because of the sinusoidal time dependency and the gyre flows being circular. The main difference between analytical and numerically computed trajectories can be expressed in terms of phase differences, which are dominantly determined by errors due to the linear interpolation in space. Trajectories with a large curvature exhibit larger phase differences. For more general flow fields, the time-interpolation errors will also contribute significantly to these phase deviations.

We have shown that accurate and efficient algorithms can be devised for calculating Lagrangian trajectories that may prove useful for the offline analysis of large datasets derived from GCMs. The present method will now be used to trace the thermohaline conveyor belt and other water masses in the high-resolution ocean GCM OCCAM.

Acknowledgments. This work is financially supported by the European MAST-III project TRACMASS (contract MAS3-CT97-0142). One of us (PdV) acknowledges the kind hospitality of the International Meteorological Institute in Stockholm, where part of this work was performed.

REFERENCES

Blanke, B., and S. Raynaud, 1997: Kinematics of the Pacific Equatorial Undercurrent: An Eulerian and Lagrangian approach from GCM results. *J. Phys. Oceanogr.,* **27,** 1038–1053.

Bönig, C. W., and M. D. Cox, 1988: Particle dispersion and mixing of conservative properties in an eddy-resolving model. *J. Phys. Oceanogr.,* **18,** 320–338.

Döös, K., 1995: Interocean exchange of water masses. *J. Geophys. Res.,* **100,** 13 499–13 514.

Drijfhout, S. S., E. Maier-Reimer, and U. Mikolajewicz, 1996: Tracing the conveyor belt in the Hamburg large-scale geostrophic ocean general circulation model. *J. Geophys. Res.,* **101,** 22 563–22 575.

Fujio, S., T. Kadowaki, and N. Imasato, 1992: World ocean circulation diagnostically derived from hydrographic and wind stress fields. 2: The water movement. *J. Geophys. Res.,* **97,** 14 439–14 452.

Webb, D. J., and Coauthors, 1997: A multiprocessor ocean general circulation model using message passing. *J. Atmos. Oceanic Technol.,* **14,** 175–183.