Notes on Semi-Lagrangian Time Integration

Martin Schreiber

January 5th 2020

Contents

1 Related work **2**

2 Sidenotes **3**

3 List of symbols **3**

4 SWE equation **3**

5 Angular and Cartesian coordinate systems **4**
- 5.1 Point: Gaussian → Cartesian . . . 4
- 5.2 Point: Cartesian → Gaussian . . . 4
- 5.3 Velocity: Gaussian → Cartesian / Cartesian → Gaussian . . . 4
- 5.4 Velocity basis . . . 4

6 Two or three time level scheme **5**
- 6.1 Two time level scheme (variant LVL_two) . . . 5
- 6.2 Three time level schemes (variant LVL_three [not implemented]) . . . 5

7 Semi-Lagrangian methods **5**
- 7.1 Two-level-time SL-SI scheme (variant SL_extrapol [not implemented]) . . . 6
- 7.2 SETTLS (variant SL_SETTLS) . . . 6
- 7.3 Semi-Lagrangian EXP SETTLS (variant SL_EXP_SETTLS) . . . 7

8 Departure points **7**
- 8.1 General iterative approach to determine the midpoint . . . 7
- 8.2 Ritchies 1985 version (variant DP_ritchie [not implemented]) . . . 8
  - 8.2.1 Introduction . . . 8
  - 8.2.2 Computing midpoint of trajectory (via normalization) . . . 8
  - 8.2.3 Midpoint rule . . . 9
  - 8.2.4 Full iteration scheme (realization of variant DP_ritchie) . . . 9
- 8.3 SETTLS (variant DP_SETTLS) . . . 10

9 Sphere approximation of trajectory **10**
- 9.1 Linear approximation with normalization (SP_normalize) . . . 10
- 9.2 Other versions (variant DP_[TODO]) . . . 10

10 Coriolis effect **10**
- 10.1 Semi-implicit treatment of Coriolis effect (variant CE_implicit) . . . 10
- 10.2 SL treatment of Coriolis effect (variant CE_lagrangian) . . . 10
- 10.3 Nonlinear treatment of Coriolis effect (variant CE_nonlinear) . . . 11

11 Interpolation order **11**
- 11.1 1) Linear . . . 11
- 11.2 2) Cubic . . . 11

12 Linear or quadratic Gaussian grid for interpolation **11**
- 12.1 1) Linear Gaussian grid (variant INT_linear) . . . 11
- 12.2 2) Quadratic Gaussian grid (variant INT_quadratic) . . . 11

13 Acknowledgements **11**

References **11**

1 Related work

- First report of semi-Lagrangian formulation for weather prediction by Wiin-Nielsen [Wiin-Nielsen, 1959].
- Semi-Lagrangian formulation for barotropic atmosphere by Sawyer [Sawyer, 1963]. Formulation which is based on conserving its vorticity along the trajectories (eq. (1)) based on wind field.
- Robert studied this in combination with implicit time integration using non-divergent barotropic vorticity equation [Robert, 1981].
  - Here, he first identified that explicit time integration methods leads to 700 times smaller errors than the overall ones for the fast gravity modes, hence justifying implicit treatment of them. Similar arguments with Rossby waves lead to a 40 times smaller error for explicit treatment of the non-linear advection, hence justifying semi-Lagrangian methods.
  - A divergence diffusion and time filter were added to this model and removing them caused instabilities.
    This was accounted for by the *Coriolis effect not treated semi-implicitly* and the semi-Lagrangian formulation was only put on the vorticity equation.
- A new formulation, tackling the aforementioned issues was investigated for **primitive meteorological equations** by Robert [Robert, 1982].
  - **A three time level scheme** was used (departure, arrival and centered points)
  - This used a centered difference scheme for the velocity and geopotential prognostic variables, hence leading to a 2nd order accurate scheme.
- Numerical study in ECMWF's model by Temperton [Temperton, 1995]
  - Different treatments of the Coriolis effect in three-time-level schemes were studied.
  - Method 1) **Semi-implicit treatment of Coriolis term** using two-time-level scheme (see [Cote and Staniforth, 1988]):
    * Simply solve linear equation including Coriolis term (see also T-REXI for linear SWE on rotating sphere).
    * Advantage: Also suitable for *two-time-level schemes*.
    * Drawback in rotated or stretched coordinate systems: Solving for implicit system of equations gets *significantly more expensive*.
    * This paper also proposed a ***reduced Gaussian grid*** transformation for the velocity fields, since no anti-aliasing is required.
  - Method 2) Consider Coriolis effect before & at the end of the trajectories as part of the velocity field (see ROCHAS, M. 1990. ARPEGE Documentation, Part 2, Ch. 6 (available from Meteo-France)):
    * The Coriolis term is absorbed into the velocity field $v + 2\Omega \times r$.
    * Coriolis term is added at departure points and subtracted at arrival points.
    * Approximations of spherical geometry in determination of trajectories according to [Ritchie and Beaudoin, 1994] provided "bad" results.
    * Avoiding these approximations led to good results, leading this method to be the targeted standard three-time-level scheme in IFS (in 1995).
- ECMWF IFS model:
  - The three-level-time SL-SI scheme [Ritchie et al., 1995], operational from 1991 to Dec. 1996.
  - **"1997 version"**: **The two-level-time SL-SI scheme** [Temperton et al., 2001], operational from Dec. 1996 to 1998.
    Issues: Noise showing up for large time step sizes => Reduction of time step size
- Hortal developed the Stable Extrapolation Two-Time-Level Scheme (SETTLS) scheme [Hortal, 2002]:
  - Tackle problem of extrapolation in a two-level-time SL-SI scheme
- Richies 1986, Semi-Lagrangian Advection on a Gaussian Grid [Ritchie, 1987].
  - Interpolating and non-interpolating SL schemes.
  - TODO: more

2 Sidenotes

- Gaussian grids:
  - linear Gaussian grid: No anti-aliasing
  - quadratic Gaussian Grid: Alias-free computation of quadratic terms

3 List of symbols

Symbol	Description
$\frac{d}{dt}$	Total derivative
F	PDE $F = N + L$ in total derivative formulation
N	Nonlinearities without nonlinear advection
L	Linearities
$R = R(t)$	Position
$V = V(t)$	Velocity
$\cdot_D$	Subscript D for value at departure point
$\cdot_A$	Subscript A for value at arrival point

4 SWE equation

Given the velocity-based formulation

$$\frac{\partial \Phi}{\partial t} = -\nabla \cdot (\Phi V)$$
$$\frac{\partial V}{\partial t} = -\nabla \Phi - fk \times V - V \cdot \nabla V$$

and using the total material derivative

$$\frac{D}{Dt} = \frac{\partial}{\partial t} + V \cdot \nabla$$

we can write

$$\frac{D\Phi}{Dt} = -\Phi \nabla \cdot V$$
$$\frac{DV}{Dt} = -\nabla \Phi - fk \times V.$$

Using a linearization around $\Phi = \overline{\Phi} + \Phi'$, we get

$$\frac{D\Phi'}{Dt} = -\overline{\Phi} \nabla \cdot V - \Phi' \nabla \cdot V$$
$$\frac{DV}{Dt} = -\nabla \Phi' - fk \times V$$

and we can split this equation into linear and non-linear parts.

For a compact notation, let $U = (\Phi', V)^T = (\Phi', u, v)^T$ be the vector of the vectors of variables. We can then write

$$\frac{DU}{Dt} = L_g(U) + L_f(U) + N_d(U)$$

with the gravity modes

$$L_g(U) = \begin{bmatrix} -\overline{\Phi} \nabla \cdot V \\ -\nabla \Phi' \end{bmatrix}$$

the Coriolis effect

$$L_f(U) = \begin{bmatrix} 0 \\ -fk \times V \end{bmatrix}$$

and the nonlinear divergence

$$N_d(U) = \begin{bmatrix} -\Phi' \nabla \cdot V \\ 0 \end{bmatrix}$$

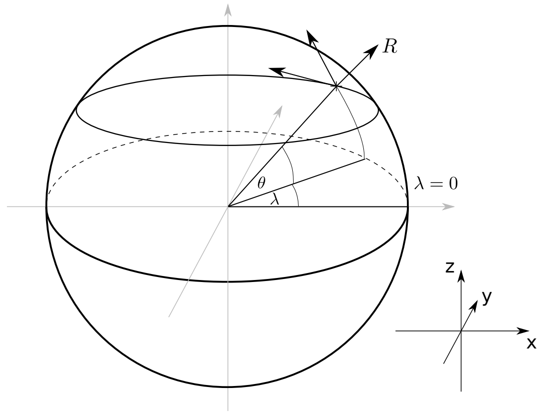

Figure 1: Gaussian and Cartesian coordinate system displaying points and velocity basis for given Gaussian angles.

5 Angular and Cartesian coordinate systems

In this section, we gather the equations required to get from Gaussian space to Cartesian space for points and velocities, see Fig. 1.

5.1 Point: Gaussian → Cartesian

Given a Gaussian coordinate (λ, θ) the Cartesian coordinate R can be computed via

$$R = \begin{bmatrix} \cos\lambda\cos\theta \\ \sin\lambda\cos\theta \\ \sin\theta \end{bmatrix}.$$

5.2 Point: Cartesian → Gaussian

Given a Cartesian coordinate R, the Gaussian coordinate is given by

$$\lambda = \arctan\left(R_y/R_x\right) + \begin{cases} \pi & \text{if } R_x < 0 \\ 2\pi & \text{if } R_y < 0 \\ 0 & \text{else} \end{cases}$$

and

$$\theta = \arccos\left(-R_z\right) - \pi/2.$$

5.3 Velocity: Gaussian → Cartesian / Cartesian → Gaussian

Since there's no scaling involved, the transformation matrices for vectors (such as velocity) are the same as for point transformations.

5.4 Velocity basis

For a velocity given in Cartesian coordinates we first determine the basis at the particular lat/lon coordinate R. On a unit sphere, the first basis vector perpendicular to the surface is trivially given by $B_R = R$. The vector along the longitude is then given by

$$B_{Lon} = \begin{bmatrix} -\sin\lambda \\ \cos\lambda \\ 0 \end{bmatrix}$$

where we also see that

$$\begin{aligned} R \cdot B_{Lon} &= \begin{bmatrix} \cos\lambda\cos\theta \\ \sin\lambda\cos\theta \\ \sin\theta \end{bmatrix} \cdot \begin{bmatrix} -\sin\lambda \\ \cos\lambda \\ 0 \end{bmatrix} \\ &= -\sin\lambda\cos\lambda\cos\theta + \cos\lambda\sin\lambda\cos\theta = 0 \end{aligned}$$

and using the cross product, we get

$$\begin{aligned}
B_{Lat} &= B_{Lon} \times R \\
&= \begin{bmatrix} -\sin\lambda \\ \cos\lambda \\ 0 \end{bmatrix} \times \begin{bmatrix} \cos\lambda\cos\theta \\ \sin\lambda\cos\theta \\ \sin\theta \end{bmatrix} \\
&= \begin{bmatrix} \cos\lambda\sin\theta \\ \sin\lambda\sin\theta \\ -\sin\lambda\sin\lambda\cos\theta - \cos\lambda\cos\lambda\cos\theta \end{bmatrix} \\
&= \begin{bmatrix} \cos\lambda\sin\theta \\ \sin\lambda\sin\theta \\ -\cos\theta \end{bmatrix}
\end{aligned}$$

where we check that $B_{Lat} \cdot B_{Lon} = 0$ and

$$\begin{aligned}
B_{Lat} \cdot R = &\begin{bmatrix} \cos\lambda\sin\theta \\ \sin\lambda\sin\theta \\ -2\cos\theta \end{bmatrix} \begin{bmatrix} \cos\lambda\cos\theta \\ \sin\lambda\cos\theta \\ \sin\theta \end{bmatrix} \\
&= \cos^2\lambda\sin\theta\cos\theta + \sin^2\lambda\sin\theta\cos\theta - \cos\theta\sin\theta \\
&= \left(\cos^2\lambda + \sin^2\lambda\right)\sin\theta\cos\theta - \cos\theta\sin\theta \\
&= 0.
\end{aligned}$$

Using dot products, we can gain the lon/lat velocities from the Cartesian ones by computing

$$V^A(\lambda, \theta, t) = \begin{bmatrix} B_{Lon}^T \\ B_{Lat}^T \\ B_R^T \end{bmatrix} V^C(\lambda, \theta, t)$$

with $B_R^T V^C = 0$.

6 Two or three time level scheme

6.1 Two time level scheme (variant LVL_two)

Following [Hortal, 2002], we write for the variables X

$$\frac{dX}{dt} = L + N$$

with d/dt the total derivative, L the linearized part and N all terms treated as nonlinearities.

The 1997 version discretized this equation with an averaging as

$$\frac{X_A^{t+\Delta t} - X_D^t}{\Delta t} = \frac{1}{2}\left(L_A^{t+\Delta t} + L_D^t\right) + \frac{1}{2}\left(N_D^{t+\frac{\Delta t}{2}} + N_A^{t+\frac{\Delta t}{2}}\right).$$

Subscript A marks the arrival points which are aligned at grid cells and subscript D marks the departure points which typically require an interpolation.

6.2 Three time level schemes (variant LVL_three [not implemented])

We just consider a two-time level scheme here, since the three-level scheme don't seem to be used anymore.

7 Semi-Lagrangian methods

For all Semi-Lagrangian methods we assume the departure point to be already known. Calculating the departure point is discussed in Section (5) since the computation of departure points is based on an implicit semi-Lagrangian way, hence on the following methods.

7.1 Two-level-time SL-SI scheme (variant SL_extrapol [not implemented])

(See [Temperton et al., 2001]) Following [Hortal, 2002], we write for the variables X

$$\frac{dX}{dt} = L + N$$

with d/dt the total derivative, L the linearized part and N all nonlinearly treated terms.

The 1997 version discretized this equation with an averaging as

$$\frac{X_A^{t+\Delta t} - X_D^t}{\Delta t} = \frac{1}{2}\left(L_A^{t+\Delta t} + L_D^t\right) + \frac{1}{2}\left(N_D^{t+\frac{\Delta t}{2}} + N_A^{t+\frac{\Delta t}{2}}\right). \tag{1}$$

Subscripts A mark the arrival points which are aligned at grid cells and subscripts D mark the departure points which typically require an interpolation.

Superscripts $t + \Delta t$ mark the unknowns at the next time step and $t + \frac{\Delta t}{2}$ mark quantities computed by linear extrapolation in time from variables at time t and $t - \Delta t$.

The movement along the trajectories for a position R and velocity V is given by

$$\frac{dR}{dt} = V.$$

In the 1997 version, the arrival point is approximated by

$$R_A^{t+\Delta t} \approx R_D^t + \Delta t \cdot V_M^{t+\frac{\Delta t}{2}}$$

where the departure positions R_D are to be computed. Here, also the velocities between the time steps is computed by an extrapolation. Given the departure points, Eq. (1) can be solved. However, for non-constant V, this method turned out to be unstable [Hortal, 2002, Durran, 1999].

7.2 SETTLS (variant SL_SETTLS)

See [Hortal, 2002].

As a first step, the approximation of the arrival point is based on a 2nd order Taylor expansion

$$\begin{aligned} R_A^{t+\Delta t} &\approx R_D^t + \Delta t \cdot \left[\frac{dR}{dt}\right]_D^t + \frac{(\Delta t)^2}{2} \cdot \left[\frac{d^2R}{dt^2}\right]_{AV} \\ &= R_D^t + \Delta t \cdot V_D^t + \frac{(\Delta t)^2}{2} \cdot \left[\frac{dV}{dt}\right]_{AV} \end{aligned}$$

with AV some average. Based on numerical studies, the averaging was chosen to be

$$\left[\frac{dV}{dt}\right]_{AV} \approx \left[\frac{dV}{dt}\right]^{t-\frac{\Delta t}{2}} \approx \frac{V_A^t - V_D^{t-\Delta t}}{\Delta t}.$$

Finally, the equation reads

$$\begin{aligned} R_A^{t+\Delta t} &\approx R_D^t + \Delta t \cdot V_D^t + \frac{(\Delta t)^2}{2} \cdot \frac{V_A^t - V_D^{t-\Delta t}}{\Delta t} \\ &= R_D^t + \frac{\Delta t}{2} \cdot \left(\left[2V^t - V^{t-\Delta t}\right]_D + V_A^t\right). \end{aligned}$$

Note, that this equation solely depends on the velocity field at time $t - \Delta t$ and at t and that is solved iteratively, similar to [Temperton et al., 2001] to determine the departure points. Based on the information in Section (8) on departure point, we can use the approximation of the departure point to evaluate quantities with $[]_D$.

$$R_D^t \approx R_A^{t+\Delta t} - \frac{\Delta t}{2} \cdot \left(\left[\underbrace{2V^t - V^{t-\Delta t}}_{\text{extrapolated velocities}}\right]_D + V_A^t\right). \tag{2}$$

Again, this is solved iteratively until the departure point has been determined. Based on this, we can use the interpolation at the departure point and the same equation to advect other state variables along the Lagrangian trajectory with the same equation.

Using the midpoint trapezoidal rule, we get

$$\frac{X_A^{t+\Delta t} - X_D^t}{\Delta t} = \frac{1}{2}\left(L_A^{t+\Delta t} + L_D^t\right) + N_*^{t+\frac{1}{2}\Delta t} \tag{3}$$

with the non-linearities at the half of the trajectory approximated by extrapolation

$$N_*^{t+\frac{1}{2}\Delta t} = \frac{1}{2}\left(\left[2N^t - N^{t-\Delta t}\right]_D + N^t\right).$$

We need to solve Eq. 3 which is accomplished by rearranging it to a Helmholtz problem

$$X_A^{t+\Delta t} = X_D^t + \frac{1}{2}\Delta t\left(L_A^{t+\Delta t} + L_D^t\right) + \Delta t N_*^{t+\frac{1}{2}\Delta t}$$
$$\underbrace{X_A^{t+\Delta t} - \frac{1}{2}\Delta t L_A^{t+\Delta t}}_{\text{Helmholtz problem}} = \underbrace{X_D^t + \frac{1}{2}\Delta t L_D^t + \Delta t N_*^{t+\frac{1}{2}\Delta t}}_{\text{K: Right hand side at departure point}}.$$

The equation

$$X_A^{t+\Delta t} - \frac{1}{2}\Delta t L_A^{t+\Delta t} = K$$

can be solved with an existing backward Euler solver by considering

$$\frac{X^{t+\Delta t} - X^t}{\Delta t} = LX^{t+\Delta t},$$
$$X^{t+\Delta t} - \Delta t L X^{t+\Delta t} = X^t$$

hence we can use a half backward Euler solver for the linear parts.

7.3 Semi-Lagrangian EXP SETTLS (variant SL_EXP_SETTLS)

See [Peixoto and Schreiber, 2019]. Compute

$$U_{SLEX}^{n+1} = e^{\Delta t L} U_D^n + \Delta t e^{\Delta t L} \tilde{N}^{n+\frac{1}{2}}$$
$$= e^{\Delta t L}\left(U_D^n + \Delta t \tilde{N}^{n+\frac{1}{2}}\right)$$

using

$$\tilde{N}_e^{n+\frac{1}{2}} = \frac{1}{2}\left[2\tilde{N}^n - e^{\Delta t L}\tilde{N}^{n-1}\right]_D^n + \frac{1}{2}\tilde{N}^n.$$

- 1. Step “SL”: Compute departure points used to evaluate $[]_D$ terms.
- 2. Step “U departure”: Compute U_D^n.
- 3. Steps “nonlinear”
  - a) Compute $\tilde{N}$ and $\tilde{N}^{n-1}$ by evaluating non-linearities (maybe including the Coriolis term).
  - b) Compute $\tilde{N}_D^{n+\frac{1}{2}} = \left[2\tilde{N}^n - e^{\Delta t L}\tilde{N}^{n-1}\right]_D$.
  - c) Compute $\tilde{N}_e^{n+\frac{1}{2}} = \frac{1}{2}\left(\tilde{N}_D^{n+\frac{1}{2}} + \tilde{N}^n\right)$.
  - d) $K = U_D^n + \Delta t \tilde{N}^{n+\frac{1}{2}}$
- 4. Step “finalize”: Compute $U_{SLEX}^{n+1} = e^{\Delta t L}(K)$.

8 Departure points

In all methods, an iterative way is used to compute departure point $R(t - \Delta t)$ for given velocity $V()$ and arrival point $R(t)$. All points are given in vector space and the origin at a sphere with radius a.

8.1 General iterative approach to determine the midpoint

See [Ritchie, 1987]. All approaches are based on an iterative way to approximate the midpoint. This is described in the following section. Based on the midpoint, it’s easy to also get a 2nd order accurate method.

8.2 Ritchies 1985 version (variant DP_ritchie [not implemented])

See [Ritchie, 1987]. Richie used an approximated version by first assuming the velocity vector in 3D, also tracing it back along this linearized velocity and then projecting the resulting point back to the earth's surface.

8.2.1 Introduction

This approach starts with a Taylor expansion and and implicit system to solve for to compute the departure point

$$R^*(t) = R(t+\Delta t) - \Delta t V(t) + O(\Delta t^2) \tag{4}$$

where $R(t+\Delta t)$ is known and $R(t)$ as well as $V(t)$ need to be computed.

8.2.2 Computing midpoint of trajectory (via normalization)

First of all, we assume $V(t)$ to be already available. However, $R^*(t)$ would still be off the sphere's surface which can be fixed with a normalization

$$\begin{aligned} R(t) &= bR^*(t) \\ &= b\left(R(t+\Delta t) - \Delta t V(t)\right) \end{aligned} \tag{5}$$

and since we require $|R^*(t)| = a$, we get

$$\begin{aligned} l &= \left|R\left(t+\Delta t\right) - \Delta t V\left(t\right)\right| \\ &= \left(\left(R\left(t+\Delta t\right) - \Delta t V\left(t\right)\right) \cdot \left(R\left(t+\Delta t\right) - \Delta t V\left(t\right)\right)\right)^{1/2} \\ &= \left(a^2 - 2\Delta t R\left(t+\Delta t\right) V(t) + \Delta t^2 V\left(t\right)^2\right)^{1/2} \end{aligned}$$

Projecting $R(t)$ on the sphere's surface yields

$$\begin{aligned} R(t) &= \left(R(t+\Delta t) - \Delta t V(t)\right) \cdot \frac{a}{l} \\ &= \left(R(t+\Delta t) - \Delta t V(t)\right) \cdot \left(1 - 2\Delta t R\left(t+\Delta t\right) V(t)/a^2 + \Delta t^2 V\left(t\right)^2/a^2\right)^{-1/2} \\ &= \left(R(t+\Delta t) - \Delta t V(t)\right) \cdot b \end{aligned}$$

with

$$b = \left(1 + \Delta t^2 V\left(t\right)^2/a^2 - 2\Delta t V(t) R\left(t+\Delta t\right)/a^2\right)^{-1/2}. \tag{6}$$

We can then use Eq. 4

$$R(t+\Delta t) = R(t) + \Delta t V(t) + O(\Delta t^2)$$

to replace $R(t+\Delta t)$ yielding

$$\begin{aligned} b &= \left(1 + \Delta t^2 V\left(t\right)^2/a^2 - 2\Delta t V(t)\left(R(t) + \Delta t V(t) + O(\Delta t^2)\right)/a^2\right)^{-1/2} \\ &= \left(1 + \Delta t^2 V\left(t\right)^2/a^2 - 2\Delta t V(t) R(t)/a^2 - 2\Delta t^2 V(t)^2/a^2 - 2\Delta t O(\Delta t^2) V(t)/a^2\right)^{-1/2} \\ &= \left(1 - \Delta t^2 V\left(t\right)^2/a^2 - 2\Delta t V(t) R(t)/a^2 - 2\Delta t O(\Delta t^2) V(t)/a^2\right)^{-1/2}. \end{aligned} \tag{7}$$

Using

$$\begin{aligned} V(t) \cdot R(t) &= \sum_n V_n(t) R_n(t) \\ &= \sum_n \frac{1}{2}\frac{d}{dt}\left(R_n(t)\right)^2 \\ &= \frac{1}{2}\frac{d}{dt}\sum_n \left(R_n(t)\right)^2 \\ &= \frac{1}{2}\frac{d}{dt}\left(R(t) \cdot R(t)\right) \\ &= \frac{1}{2}\frac{d}{dt}\left\{|R(t)|^2\right\} \end{aligned}$$

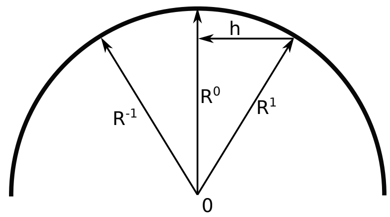

Figure 2: Sketch to compute $R^{-1} = R(t - \Delta t)$.

we can simplify the equation to

$$b = \left(1 - \Delta t^2 V\left(t\right)^2 / a^2 + O(\Delta t^3)\right)^{-1/2}.$$

Next, we use a Taylor series approximation

$$\begin{aligned} f(\Delta t) &= (1 - \Delta t)^{-1/2} \\ &\approx 1 + \Delta t \left(-\frac{1}{2}\right)(1-0)^{-\frac{3}{2}}(-1) + O(\Delta t^2) \\ &= 1 + \Delta t \frac{1}{2} + O(\Delta t^2) \end{aligned}$$

we finally get the correction factor

$$\boxed{b = 1 + \frac{1}{2}\Delta t V\left(t\right)/a + O(\Delta t^2)}$$

which we can use to compute new location $R(t) = b\left(R(t + \Delta t) - \Delta t V(t)\right)$.

8.2.3 Midpoint rule

Based on the new departure location $R(t)$ we use this as part of a midpoint rule to compute $R(t - \Delta t)$ using the sketch in Figure 2 where we use $R^n = R(t + n\Delta t)$.

We parametrize the vector $h(\gamma) = \gamma R^0 - R^1$ and request it to be orthogonal to b via

$$\begin{aligned} h(\gamma) \cdot R^0 &= 0 \\ \left(\gamma R^0 - R^1\right) \cdot R^0 &= 0 \\ \gamma R^0 \cdot R^0 &= R^1 \cdot R^0 \\ \gamma &= \frac{R^1 \cdot R^0}{R^0 \cdot R^0}. \end{aligned}$$

Finally, we can compute R via

$$\begin{aligned} R^{-1} &= R^1 + 2h \\ &= R^1 + 2\left(\gamma R^0 - R^1\right) \\ &= R^1 + 2\left(\frac{R^1 \cdot R^0}{R^0 \cdot R^0} R^0 - R^1\right) \\ &= R^1 + 2\left(\frac{R^1 \cdot R^0}{R^0 \cdot R^0} R^0 - R^1\right) \\ &= 2\frac{R^1 \cdot R^0}{R^0 \cdot R^0} R^0 - R^1 \end{aligned}$$

and with $R^0 \cdot R^0 = a^2$ we get

$$R(t - \Delta t) = 2\left[R(t) \cdot R(t + \Delta t)/a^2\right] R(t) - R(t + \Delta t).$$

8.2.4 Full iteration scheme (realization of variant DP_ritchie)

The iteration will solve Eq. (5) iteratively. The iterative scheme with the k-th iteration is given by

$$R_{k+1}(t) = b_{k+1}\left(R(t + \Delta t) - \Delta t V_{k+1}(t)\right)$$

with $V_{k+1}(t)$ to be computed by interpolation. The iteration is started with $R_0(t) = R(t+\Delta t)$ which is the arrival point. Given the Equations in Section 5 we can compute the velocity V^C in Cartesian space and hence solve Eq. (6)

$$b_k = \left(1 - \Delta t^2 V_k^C(t)^2 - 2\Delta t V_k^C(t) \cdot R_k(t)\right)^{-1/2}.$$

The iterations R_{k+1} can be executed until sufficient convergence is reached. However, two iterations are sufficient to gain 2nd order accuracy.

Finally, the departure position $R(t-\Delta t)$ can be calculated with a midpoint rule (see Eq. (8.2.3))

$$R(t-\Delta t) = 2\left[R(t) \cdot R(t+\Delta t)/a^2\right] R(t) - R(t+\Delta t).$$

Based on this, we can use the interpolation at the departure point and the same equation to advect other state variables along the Lagrangian trajectory with the same equation.

8.3 SETTLS (variant DP_SETTLS)

This method is based on the SETTLS SL method in Section

9 Sphere approximation of trajectory

This section describes how to approximate the movement of a point for a given velocity

$$R(t) = R(t+\Delta t) - \Delta t V(t+\Delta t)$$

on a sphere.

9.1 Linear approximation with normalization (SP_normalize)

This method assumes the velocity to act on the plane planar to the sphere. Then, the arrival point is projected back to the sphere with a normalization. Section (5) describes this method.

9.2 Other versions (variant DP_[TODO])

Other versions basically differ in the way how the velocity vector is applied on the arrival point. E.g. using exact tracing on the sphere's surface.

10 Coriolis effect

10.1 Semi-implicit treatment of Coriolis effect (variant CE_implicit)

See [Cote and Staniforth, 1988], [Ritchie et al., 1995]. 1st discussed method in [Temperton, 1995]. Simply treat it as part of the linear part L.

10.2 SL treatment of Coriolis effect (variant CE_lagrangian)

See ROCHAS, M. 1990, 2nd method in [Temperton, 1995]. The underlying idea is to use the definition of the material derivative

$$\frac{D}{Dt} = \frac{\partial}{\partial t} + V \cdot \nabla$$

and

$$\frac{\partial}{\partial t} fk \times R = fk \times V$$

to rewrite the momentum equation to

$$
\begin{aligned}
\frac{DV}{Dt} &= -\nabla\Phi - fk\times V \\
\frac{\partial}{\partial t}V + V\cdot\nabla V &= -\nabla\Phi - \frac{\partial}{\partial t}fk\times R \\
\frac{\partial}{\partial t}V + V\cdot\nabla\left(V + fk\times R\right) + \frac{\partial}{\partial t}fk\times R &= -\nabla\Phi \\
\frac{\partial}{\partial t}\left(V + fk\times R\right) + V\cdot\nabla(V + fk\times R) &= -\nabla\Phi \\
\frac{DV + fk\times R}{Dt} &= -\nabla\Phi.
\end{aligned}
$$

According to the aforementioned related work, this takes the calculated departure points in the same way as from the previous work.

However, for the velocities, the the Coriolis effect is added at the departure point and simply subtracted again at the arrival points. Note, that the R coordinate at the departure and arrival point must be used for this.

10.3 Nonlinear treatment of Coriolis effect (variant CE_nonlinear)

11 Interpolation order

The big question here is where to use which order. Some particular orders have been inferred for the computation of the trajectory and for the interpolation of the state variables

11.1 1) Linear

First order accurate interpolation

11.2 2) Cubic

Third order accurate interpolation. Limiters might be used.

12 Linear or quadratic Gaussian grid for interpolation

12.1 1) Linear Gaussian grid (variant INT_linear)

Reduced computation costs since no anti-aliasing is required

12.2 2) Quadratic Gaussian grid (variant INT_quadratic)

Additional computation costs because of anti-aliasing strategy

13 Acknowledgements

Discussions with Nils Wedi, Michail Diamantakis and Pedro Peixoto.

References

[Cote and Staniforth, 1988] Cote, J. and Staniforth, A. (1988). A two-time-level semi-Lagrangian semi-implicit scheme for spectral models.

[Durran, 1999] Durran, D. (1999). *Numerical Methods for wave equations*. Springer.

[Hortal, 2002] Hortal, M. (2002). The development and testing of a new two-time-level semi-Lagrangian scheme (SETTLS) in the ECMWF forecast model. *Q. J. R. Meteorol. Soc.*, 2:1671–1687.

[Peixoto and Schreiber, 2019] Peixoto, P. S. and Schreiber, M. (2019). Semi-Lagrangian Exponential Integration With Application To the Rotating Shallow Water Equations. *SIAM Journal on Scientific Computing*, 41(5):903–928.

[Ritchie, 1987] Ritchie, H. (1987). Semi-Lagrangian Advection on a Gaussian Grid. *Monthly Weather Review*, 115(2):608–619.

[Ritchie and Beaudoin, 1994] Ritchie, H. and Beaudoin, C. (1994). Approximations and sensitivity experiments with a baroclinic semi- Lagrangian spectral model.

[Ritchie et al., 1995] Ritchie, H., Temperton, C., Simmons, A., Hortal, M., Davies, T., Dent, D., and Hamrud, M. (1995). Implementation of the Semi-Lagrangian Method in a High-Resolution Version of the ECMWF Forecast Model.

[Robert, 1981] Robert, A. (1981). A stable numerical integration scheme for the primitive meteorological equations. *Atmosphere - Ocean*, 19(1):35–46.

[Robert, 1982] Robert, A. (1982). A Semi-Lagrangian Scheme for and Semi-Implicit Numerical Integration the Primitive Meteorological Equations. *Journal of the Meteorological Society of Japan*, (February):319–325.

[Sawyer, 1963] Sawyer, J. S. (1963). A semi-Lagrangian method of solving the vorticity advection equation. *Tellus*, 15(4):336–342.

[Temperton, 1995] Temperton, C. (1995). Treatment of the Coriolis Terms in Semi-Lagrangian Spectral Models. (March 1994).

[Temperton et al., 2001] Temperton, C., Hortal, M., and Simmons, A. (2001). A two-time-level semi-Lagrangian global spectral model. *Quarterly Journal of the Royal Meteorological Society*, 127(571):111–127.

[Wiin-Nielsen, 1959] Wiin-Nielsen, A. (1959). On the Application of Trajectory Methods in Numerical Forecasting. *Tellus*, 11(2):180–196.