Tangent-linear and adjoint models

By **Clive Temperton**

European Centre for Medium-Range Weather Forecasts

Abstract

Work has started on coding the tangent-linear and adjoint of the semi-Lagrangian scheme in the ECMWF model. In this paper, some initial experiences and results are described.

1. Introduction

In November 1997, a four-dimensional variational data assimilation system (4D-Var) was implemented operationally at ECMWF (*Rabier et al.*, 1998). The theoretical background is decribed by *Talagrand and Courtier* (1987). For reasons of computational economy, 4D-Var is implemented in its "incremental" form (*Courtier et al.*, 1994). The algorithm consists of a pair of nested loops. The outer loop uses the full-resolution model to compute the mismatch between the current estimate of the model trajectory and the observations, over a six-hour assimilation window. The inner loop employs the *tangent-linear* and *adjoint* of a simpler and lower-resolution model to compute the gradients used in minimizing the 4D-Var cost function.

Currently the tangent-linear and adjoint are only available for the Eulerian version of the model. This restricts the horizontal resolution of the inner loop to a spectral truncation of T63, while the outer loop uses the semi-Lagrangian version of the model with a horizontal resolution of T319. It would be highly desirable to increase the resolution of the inner loop; in practical terms this means that the tangent-linear and adjoint of the semi-Lagrangian scheme are needed.

The author is currently engaged in developing the required tangent-linear and adjoint code. At the time of the Seminar, a tangent-linear version of the semi-Lagrangian scheme in the shallow-water configuration of the ECMWF model (IFS) had been coded and tested. Although this is only the first part of the project, a good deal had already been learned. The fruits of this experience are passed on here in a pedagogical spirit. Section 2 presents a simple motivation for the use of tangent-linear and adjoint equations. Section 3 shows how to construct the tangent-linear of the semi-Lagrangian scheme in a one-dimensional example. Section 4 presents the "really two-time-level" semi-Lagrangian version of the shallow-water model, which was developed in order to simplify the early development of the tangent-linear code. Section 5 describes some coding problems and preliminary results.

2. A SIMPLE EXAMPLE

The theoretical basis of 4D-Var as presented by *Talagrand and Courtier* (1987) requires a certain amount of machinery from functional analysis: Hilbert spaces, inner products and so on. This section presents a much simpler illustrative example, taken from an article by *Rabier et al.* (1995).

Consider a system which has just two predicted values, x and y, with corresponding initial values x_0 and y_0. The nonlinear "forecast model" requires a single "timestep":

$$x = ax_0^2 + by_0^2 \tag{1}$$

$$y = y_0 \tag{2}$$

where a and b are constants. Now introduce a perturbation $(\delta x_0, \delta y_0)$ into the initial conditions. The evolution of the perturbation is:

$$\delta x = 2ax_0\delta x_0 + a(\delta x_0)^2 + 2by_0\delta y_0 + b(\delta y_0)^2$$

$$\delta y = \delta y_0.$$

The corresponding *tangent-linear* model is obtained by dropping the terms which are quadratic in the perturbation quantities:

$$\delta x = 2ax_0\delta x_0 + 2by_0\delta y_0$$

$$\delta y = \delta y_0,$$

which, in matrix form, is

$$\begin{bmatrix}\delta x\\ \delta y\end{bmatrix} = \begin{bmatrix}2ax_0 & 2by_0\\ 0 & 1\end{bmatrix}\begin{bmatrix}\delta x_0\\ \delta y_0\end{bmatrix}. \tag{3}$$

Let $J(x, y)$ be a function of the predicted values, and suppose we can compute $(\partial J)/(\partial x)$ and $(\partial J)/(\partial y)$. However, we want to determine the corresponding derivatives (gradients) with respect to the *initial* values, i.e., $(\partial J)/(\partial x_0)$ and $(\partial J)/(\partial y_0)$. (This is exactly analogous to what is done in 4D-Var.) Using the standard rules of partial differentiation together with Eqs. (1) and (2),

$$\frac{\partial J}{\partial x_0} = \frac{\partial x}{\partial x_0}\frac{\partial J}{\partial x} + \frac{\partial y}{\partial x_0}\frac{\partial J}{\partial y} = 2ax_0\frac{\partial J}{\partial x}$$

$$\frac{\partial J}{\partial y_0} = \frac{\partial x}{\partial y_0}\frac{\partial J}{\partial x} + \frac{\partial y}{\partial y_0}\frac{\partial J}{\partial y} = 2by_0\frac{\partial J}{\partial x} + \frac{\partial J}{\partial y}$$

which in matrix form is

$$\begin{bmatrix} \frac{\partial J}{\partial x_0} \\ \frac{\partial J}{\partial y_0} \end{bmatrix} = \begin{bmatrix} 2ax_0 & 0 \\ 2by_0 & 1 \end{bmatrix} \begin{bmatrix} \frac{\partial J}{\partial x} \\ \frac{\partial J}{\partial y} \end{bmatrix} . \tag{4}$$

Equation (4) is the *adjoint* model corresponding to the tangent-linear model in Equation (3).

Notice that the matrix in Eq. (4) is the transpose of that in Eq. (3): loosely speaking, the adjoint model is the transpose of the tangent-linear model. Notice also that the adjoint model runs "backwards in time": starting from derivatives of J with respect to the *predicted* values, it produces derivatives of J with respect to the *initial* values.

3. TANGENT-LINEAR VERSION OF A SEMI-LAGRANGIAN SCHEME

The principles behind developing the tangent-linear version of a semi-Lagrangian scheme can be illustrated using a one-dimensional spectral model of Burger's equation. In Eulerian form, the model is:

$$\frac{\partial u}{\partial t} + u\frac{\partial u}{\partial x} = \upsilon\frac{\partial^2 u}{\partial x^2} . \tag{5}$$

The corresponding Eulerian tangent-linear model is:

$$\frac{\partial}{\partial t}(\delta u) + u\frac{\partial}{\partial x}(\delta u) + (\delta u)\frac{\partial u}{\partial x} = \upsilon\frac{\partial}{\partial x^2}(\delta u) . \tag{6}$$

Equations (5) and (6) can both be time-discretized in the same way, for example using a three-time-level scheme:

$$\frac{u^+ - u^-}{2\Delta t} + \left(u\frac{\partial u}{\partial x}\right)^0 = \upsilon\left(\frac{\partial^2 u}{\partial x^2}\right)^- \tag{7}$$

$$\frac{(\delta u)^+ - (\delta u)^-}{2\Delta t} + \left(u\frac{\partial}{\partial x}(\delta u) + (\delta u)\frac{\partial u}{\partial x} \right)^0 = \upsilon \left(\frac{\partial^2}{\partial x^2}(\delta u) \right)^- \tag{8}$$

where subscripts –, 0, + refer to time-levels $(t - \Delta t)$, t, $(t + \Delta t)$ respectively. Notice that in order to run the tangent-linear model (8), we first have to run the "trajectory" model (7) and store the values of u at each timestep. (The values of $(\partial u)/(\partial x)$ may be either stored or recomputed.)

Now consider the *Lagrangian* form of Eq. (5):

$$\frac{du}{dt} = \upsilon \frac{\partial^2 u}{\partial x^2} . \tag{9}$$

For the sake of argument, we choose a three-time-level semi-Lagrangian discretization of Eq. (9):

$$u_{\mathrm{A}}^+ = u_{\mathrm{D}}^- + 2\Delta t \upsilon \left(\frac{\partial^2 u}{\partial x^2} \right)_{\mathrm{D}}^- = Z_{\mathrm{D}} \tag{10}$$

where subscripts A and D refer respectively to the arrival gridpoint and the departure point. To complete the scheme we need to find the departure point by iteratively solving the displacement equation

$$x_{\mathrm{A}} - x_{\mathrm{D}} = 2\Delta t u_{\mathrm{M}}^0 \tag{11}$$

where M is the midpoint between D and A. Assume that (as is typical) we use linear interpolation in solving (11) but cubic interpolation to obtain the value of Z at the departure point in (10).

A schematic version of the model code is given below, together with a running commentary. For each gridpoint x_i:

(1) $x^* = x_i - u_i^0 \Delta t$! first guess of midpoint x^*

Iteration loop:

(2) $j = \mathrm{INT}(x^*/(\Delta x))$! index of gridpoint to left

(3) $\omega = (x^* - x_j)/(\Delta x)$! weight for linear interpolation

(4) $u^* = u_j^0 + \omega(u_{j+1}^0 - u_j^0)$! u at x^*

(5) $x^* = x_i - u^* \Delta t$! updated estimate of midpoint

[loop back to (2)]

(6) $x^{**} = 2x^* - x_i$! departure point

(7) $j = \mathrm{INT}(x^{**}/(\Delta x))$! index of gridpoint to left

(8) $\omega = (x^{**} - x_j)/(\Delta x)$! weight for interpolation

(9) $Z^{**} = Z_{j-1} + \sum_{k=0}^{2} (Z_{j+k} - Z_{j-1}) P_k(\omega)$! cubic Lagrange interpolation

where $Z = u^- + 2\Delta t \upsilon \left(\frac{\partial^2 u}{\partial x^2} \right)^-$

(10) $u_i^+ = Z^{**}$! end of timestep.

The corresponding tangent-linear code is obtained by differentiating the above line by line:

(T1) $(\delta x)^* = -(\delta u)_i^0 \Delta t$! first guess of $(\delta x)^*$

Iteration loop:

(T2) Do nothing ! (2) is undifferentiable!

(T3) $(\delta\omega) = (\delta x)^*/(\Delta x)$

(T4) $(\delta u)^* = (\delta u)_j^0 + \omega\{(\delta u)_{j+1}^0 - (\delta u)_j^0\} + (\delta\omega)(u_{j+1}^0 - u_j^0)$

(T5) $(\delta x)^* = -(\delta u)^* \Delta t$! updated estimate

[loop back to (T2)]

(T6) $(\delta x)^{**} = 2(\delta x)^*$

(T7) Do nothing ! (7) is undifferentiable!

(T8) $(\delta\omega) = (\delta x)^{**}/(\Delta x)$

(T9) $(\delta Z)^{**} = (\delta Z)_{j-1} + \sum_{k=0}^{2} \{(\delta Z)_{j+k} - (\delta Z)_{j-1}\} P_k(\omega)$

$$+(\delta\omega) \sum_{k=0}^{2} (Z_{j+k} - Z_{j-1}) \frac{d}{d\omega} P_k(\omega)$$

where $(\delta Z) = (\delta u)^- + 2\Delta t \upsilon \frac{\partial^2}{\partial x^2} (\delta u)^-$

(T10) $(\delta u)_i^+ = (\delta Z)^{**}$! end of timestep.

Notice that the tangent-linear of the interpolations in (T4) and (T9) contains two parts. The first part is simply an interpolation of the perturbation quantity at the original point x^* or x^{**} as defined by the "trajectory" run of the model. The second part is a correction to account for a perturbation $(\delta x)^*$ or $(\delta x)^{**}$ in this location, induced by the perturbation quantity (δu).

The significance of the "do nothing" statements (T2) and (T7) can now be appreciated: consider for example the linear interpolation in (T4). The "correction" term $(\delta\omega)(u_{j+1}^0 - u_j^0)$ is only correct if the original midpoint x^* and the perturbed midpoint $x^* + (\delta x)^*$ lie within the same interval $[x_j, x_{j+1}]$. *Polavarapu et al.* (1996) show that for infinitesimal perturbations, the tangent-linear of the interpolation is correct if and only if the first derivative of the interpolator is continuous (which implies for example cubic spline interpolation). The more reassuring result from their analysis is that even if the tangent-linear is occasionally incorrect, the error may be tolerable.

4. THE "REALLY TWO-TIME-LEVEL" SCHEME

The existing tangent-linear framework in the IFS model was based on a three-time-level Eulerian scheme. In this case, stepping the tangent-linear model forward from time-level $(n-1)$ to $(n+1)$ requires values of the trajectory model only at time-level (n). The situation is a little more complicated in the semi-Lagrangian case, where trajectory model values are required at two time-levels, both $(n-1)$ and (n). The same is true even in the so-called two-time-level scheme (*Temperton et al.*, 1999) which steps from time-level (n) to time-level $(n+1)$, since some values from time-level $(n-1)$ are in fact used to estimate quantities at time-level $(n+1/2)$.

To circumvent this difficulty in the early stages of coding the tangent-linear of the semi-Lagrangian scheme, a "really two-time-level" version of the shallow-water code was implemented. The semi-implicit semi-Lagrangian scheme is:

$$\underset{\sim}{v}^{n+1} = \underset{\sim}{v}_*^n + ((\Delta t)/2)\{(f\underset{\sim}{k}\times\underset{\sim}{v})^{n+1} + (f\underset{\sim}{k}\times\underset{\sim}{v})_*^n\}$$

$$-((\Delta t)/2)\{(\underset{\sim}{\nabla}\phi)^{n+1} + (\underset{\sim}{\nabla}\phi)_*^n\} \qquad (12)$$

$$\phi^{n+1} = \phi^n - ((\Delta t)/2)\Phi\{D^{n+1} + D_*^n\}$$

$$-((\Delta t)/2)\{(\phi' D)^n + (\phi' D)_*^n\} \qquad (13)$$

where Φ is the mean geopotential height, D is divergence, $\phi' = \phi - \Phi$ and subscript * means

evaluation at the departure point. The only unconventional feature in the above equations is that the small term $\phi' D$ is evaluated at time-level (n) rather than extrapolated to time-level $(n+1/2)$.

The most unconventional aspect of the scheme lies in the algorithm to find the departure point, which is inspired by the scheme of *Hortal* (1999). Let $\underset{\sim}{z}$ be a position vector, and consider a Taylor series expansion about the departure point $\underset{\sim}{z}_{\mathrm{D}}^{n}$. The arrival point $\underset{\sim}{z}_{\mathrm{A}}^{n+1}$ can be written as

$$\underset{\sim}{z}_{\mathrm{A}}^{n+1} = \underset{\sim}{z}_{\mathrm{D}}^{n} + \Delta t \underset{\sim}{v}_{\mathrm{D}}^{n} + \frac{1}{2}(\Delta t)^2 \left(\frac{d}{dt}\underset{\sim}{v}\right)_{\mathrm{D}}^{n} + \ldots \tag{14}$$

In the case of the shallow-water equations we can easily compute the total derivative in the last term of Eq. (14): it is simply the right-hand side of the momentum equations. Thus, (14) becomes

$$\underset{\sim}{z}_{\mathrm{A}}^{n+1} = \underset{\sim}{z}_{\mathrm{D}}^{n} + \Delta t \underset{\sim}{v}_{\mathrm{D}}^{n} + \frac{1}{2}(\Delta t)^2 \{ f \underset{\sim}{k} \times \underset{\sim}{v} - \underset{\sim}{\nabla}\phi \}_{\mathrm{D}}^{n} \tag{15}$$

which can be solved iteratively in the usual way to find the departure point D.

Equations (12), (13) and (15) constitute a *genuinely* two-time-level semi-Lagrangian scheme which, as discussed above, leads to a tangent-linear form in which the required trajectory model values are the same as for the Eulerian model.

5. CODING PROBLEMS AND PRELIMINARY RESULTS

Coding the tangent-linear of the semi-Lagrangian scheme was not always straightforward. A nice example comes from the algorithm which finds the coordinates of the departure point in spherical geometry. The original model code to find the longitude of the departure point is of the form

```
ZYY = ...
ZZW = ...
λD = MOD(π+SIGN(1., ZYY)*(ACOS(ZZW)-π),2π).
```

The last statement contains two non-differentiable functions, MOD and SIGN. The first part of the solution is to recognize that the core of this statement is simply

```
λD = ACOS(ZZW) ;
```

the rest of the code decides whether λ_D is east (>0) or west (<0) of the zero meridian. However, the tangent-linear of this simpler statement is

$$(\delta\lambda)_D = -\delta(\mathrm{ZZW}) / \mathrm{SIN}(\lambda_D)$$

so that if the departure point happens to be on the zero meridian, the perturbation $(\delta\lambda)_D$ is undefined.

The eventual solution is to recognize that the longitude of the departure point could alternatively have been computed from its sine. The final version of the tangent-linear code is of the form:

```
IF (ABS(SIN(λ_D)).GT.SQRT(0.5)) THEN
  (δλ)_D = -δ(ZZW) / SIN(λ_D)
ELSE
  (δλ)_D = -δ(ZZA) / COS(λ_D)
ENDIF
```

where ZZA is a quantity derived from the trajectory variables, which was not in the original code! (It can be verified that the result of the above is a continuous function of λ_D.) This would have been an interesting challenge for an "automatic differentiator" operating directly on the original code.

Included in the IFS framework is a configuration for testing the correctness of tangent-linear code. First, the model is run for a few timesteps from a given initial state X to produce a "trajectory" forecast $F(X)$. Secondly, the tangent-linear model is run from a random initial condition δX to produce a forecast $TL(\delta X)$. Finally for $\alpha = 10^{-\lambda}$, $(\lambda = 10, 9, \ldots, 1, 0)$, the forecast model is run from the initial condition $(X + \alpha\delta X)$ to produce $F(X + \alpha\delta X)$. For a selection of spectral coefficients, the ratio

$$R = \frac{F(X + \alpha\delta X) - F(X)}{\alpha \cdot TL(\delta X)}$$

is printed out. If the tangent-linear model is correct, this ratio should be close to 1. Typically for $\lambda = 10$ the perturbation is so small that the result can be corrupted by rounding errors, while for $\lambda = 0$ the perturbation is so large that the tangent-linear approximation becomes invalid; in between, the ratio should become very close to 1 (e.g., to within a tolerance of 10^{-6} or 10^{-7}). Using this test, the correctness of the tangent-linear version of the "really two-time-level" semi-Lagrangian scheme in the shallow-water model has been verified.

6. CONCLUDING REMARKS

The work described here represents only the first stage of the project. The next task will be to code the corresponding adjoint of the semi-Lagrangian scheme in the shallow-water equations

(in fact completed at the time of writing). These codes will then be extended to the three-dimensional model and used to enable higher-resolution inner loops within 4D-Var, as well as for other applications.

Acknowledgements

I would like to thank Florence Rabier for supplying the simple example of a tangent-linear and adjoint model outlined in Section 2. The example of the tangent-linear of a one-dimensional semi-Lagrangian scheme, given in Section 3, is based on a set of notes by Martin Janousek which I found very useful in this work.

REFERENCES

Courtier, P., Thépaut, J.-N. and Hollingsworth, A., 1994: A strategy for operational implementation of 4D-Var, using an incremental approach. *Q. J. R. Meteorol. Soc.*, **120**, 1367-1387.

Hortal, M., 1999: Aspects of the numerics of the ECMWF model. *(These Proceedings)*

Polavarapu, S., Tanguay, M., Ménard, R. and Staniforth, A., 1996: The tangent linear model for semi-Lagrangian schemes: linearizing the process of interpolation. *Tellus*, **48A**, 74-95.

Rabier, F., Klinker, E., Courtier, P. and Hollingsworth, A., 1995: Sensibilité des erreurs de prévision aux conditions initiales. *La Météorologie*, **8**, No. 12, 31-37.

Rabier, F., Thépaut, J.-N. and Courtier, P., 1998: Extended assimilation and forecast experiments with a four-dimensional variational assimilation system. *Q. J. R. Meteorol. Soc.*, **124**, 1861-1887.

Talagrand, O. and Courtier, P., 1987: Variational assimilation of meteorological observations with the adjoint vorticity equation. Part 1: Theory. *Q. J. R. Meteorol. Soc.*, **113**, 1311-1328.

Temperton, C., Hortal, M. and Simmons, A., 1999: A two-time-level semi-Lagrangian global spectral model. Submitted to *Q. J. R. Meteorol. Soc.*