

# Documentation of the TLM and adjoint models of the Lin-Rood spherical shallow water finite volume model

Santha Akella\* and I. M. Navon<sup>†‡</sup>

July 20, 2004

## Abstract

A brief computational description of the Lin-Rood shallow water finite volume (SWfv) model is provided. Along side, salient features of the model are described. 4D-VAR data assimilation experiments using this model provided us avenues to study model error introduced by different high order advection schemes (such as van Leer and PPM) in the framework of 4D-variational data assimilation. This motivated us to derive the tangent linear model (TLM) and the adjoint model of the SWfv model. Details of the development of these models along with detailed description and flowcharts of the subroutines and results are provided in this technical document.

## 1 Introduction

Numerical weather prediction and climate models have traditionally been based on either finite differences or spectral methods. Semi-implicit, semi-Lagrangian methods are known to be computationally efficient. Monotonicity preserving conservative difference schemes, such as van Leer and PPM methods, (which are used to model fluid flows that involve shocks) are mass conserving and more consistent with the physical process that we desire to simulate. The SWfv model of S.-J. Lin and R. Rood is a semi-implicit, semi-Lagrangian flux-form conserving solver of the shallow water equations on the sphere, see [1], [2], [3] and [4] for additional details.

The shallow water equations model the flow of a thin layer of fluid in the atmosphere and the general circulation models couple such shallow water models vertically, using pressure as the vertical coordinate, see [5]. The (shallow water) finite volume dynamical core is used at the NASA Data Assimilation Office (DAO), now Global Modeling and Assimilation Office (GMAO) for numerical weather prediction and climate simulations, the physics has been added to the model by a cooperative effort between GMAO and National Center for Atmospheric Research (NCAR).

Here we deal with the core of the model, comprising of the shallow water equations model only. It is

---

\*email: [sakella@math.fsu.edu](mailto:sakella@math.fsu.edu)

<sup>†</sup>[navon@csit.fsu.edu](mailto:navon@csit.fsu.edu)

<sup>‡</sup>CSIT and Department of Mathematics, FSU, Tallahassee, FL-3206.

to be noted that future efforts to use the full physics 3-D finite volume model for data assimilation requires both the TLM and adjoint of the dynamical core to be developed. Also, our effort is entirely independent of those underway at GMAO.

## 2 Brief description of the Lin-Rood SWfv model

The SWfv model integrates the familiar shallow water equations on the sphere,

$$\frac{\partial h}{\partial t} + \nabla \cdot (\mathbf{V}h) = \mathbf{0} \quad (1)$$

$$\frac{\partial u}{\partial t} = \Omega v - \frac{1}{a \cos\theta} \frac{\partial}{\partial \lambda} [\kappa + \varphi] \quad (2)$$

$$\frac{\partial v}{\partial t} = -\Omega u - \frac{1}{a} \frac{\partial}{\partial \theta} [\kappa + \varphi] \quad (3)$$

where  $h$  represents the fluid height (above the surface height),  $u$  and  $v$  represent the zonal and meridional wind velocities respectively,  $\omega$  is the angular velocity of the earth,  $a$  is the radius of the earth,  $\theta$  is the latitude.  $\lambda$  is the longitude.

The free surface potential is given by

$$\varphi = \varphi_s + g h,$$

$\kappa = \frac{1}{2} \mathbf{V} \cdot \mathbf{V}$  is the kinetic energy, and  $\Omega = 2\omega \sin\theta + \nabla X \mathbf{V}$  is the absolute vorticity.

The computation proceeds with a combination of the C-grid and D-grid. In the first half of the time step, the advective winds are updated on the C-grid, and in the other half of the time step, the prognostic variables are updated on the D-grid. For the advection scheme, one can use either first order differencing, Piece-Wise Parabolic (PPM), and van Leer schemes (either unconstrained or constrained). The poles have been treated in the same fashion as that in [6] using a polar Fourier filter.

In particular, it is to be noted that the algorithm conserves total mass for all the time of the numerical integration and, after a 60 day integration of the model, the loss in total energy is approximately lost by 0.1%, and the loss in total enstrophy is 1%. More details are available in Lin and Rood, 1997[1]. This is an excellent conservation of integral invariants of the shallow water equations.

For our studies, we choose a resolution of 128 X 64 cells along the longitude,  $\lambda$  and latitude,  $\theta$  ( $2.8^\circ$  X  $2.8^\circ$ ) directions respectively, and a time step of 600 seconds. The wavenumber 4 Rossby-Haurwitz initial condition test case has been used[8]. Fig. 3 shows the initial height field, and we show the DAY-1 height field in fig. 4, obtained using the unconstrained van Leer on C-grid and monotonicity preserving PPM on the D-grid.

Appendix A describes the flow of variables and a flowchart of the forward model.

### 3 TLM of the SWfv model

We have used TAMC [7] and hand intervention to generate the TLM. Briefly, each line of the TLM is generated by linearizing the non-linear statements within the forward model, further details about writing a TLM can be found in [9].

To obtain initial conditions for the TLM, perturbations were taken as random fields over all the model's grid points with the magnitude of the winds ranging from -3 to 3  $ms^{-1}$  and height field from -300 to 300  $m^2s^{-2}$ .

If we consider a state vector  $\mathbf{x}$ , its perturbation will be denoted by  $\delta\mathbf{x}$  (these perturbations, provide the initial conditions to the TLM).

Perturbations for the velocity components have been generated as,

$$\delta\mathbf{x} = \alpha * 6 * [RAND() - \frac{1}{2}], \quad (4)$$

where  $\mathbf{x} = u, v$ ;  $\alpha$  ranges from 1.0 to  $10^{-10}$ , and  $RAND()$  is the DOUBLE PRECISION pseudo random number generator provided by FORTRAN90 compiler on SGI Origin 200, such that the pseudo random numbers  $\in [0, 1]$ .

In a similar fashion, perturbations for the height field are obtained as,

$$\delta\mathbf{h} = \alpha * 600 * [RAND() - \frac{1}{2}]. \quad (5)$$

#### 3.1 Verification of the TLM

Let  $\mathbf{x}(t_0)$  be integrated to a state  $\mathbf{x}^n$  (here, the model state is a vector of dimension  $n \times 1$ , where  $n$  is equal to three times the total number of grid points, mathematically, it can be written as  $\mathbf{x}^T = [u, v, h]$ ) using a nonlinear model  $\mathcal{M}$ , i.e.,

$$\mathbf{x}^n = \mathcal{M}[\mathbf{x}(t_0)],$$

and let

$$\mathbf{x}'^n = \mathcal{M}[\mathbf{x}(t_0) + \delta\mathbf{x}(t_0)],$$

and finally, let

$$\Delta\mathbf{x} = \mathbf{x}'^n - \mathbf{x}^n = \mathcal{M}[\mathbf{x}(t_0) + \delta\mathbf{x}(t_0)] - \mathcal{M}[\mathbf{x}(t_0)].$$

Therefore, Taylor expansion around model state,  $\mathbf{x}(t_0)$  yields

$$\Delta\mathbf{x} = \mathcal{M}[\mathbf{x}(t_0)] + \mathcal{R}[\delta\mathbf{x}(t_0)] + \mathbf{O}([\delta\mathbf{x}(t_0)]^2) - \mathcal{M}[\mathbf{x}(t_0)],$$

where  $\mathcal{R}$  is the Jacobian of  $\mathcal{M}$ , with respect to the model state, thus it is the TLM.

Hence,

$$\Delta \mathbf{x} = \mathcal{R}[\delta \mathbf{x}(t_0)] + \mathbf{O}([\delta \mathbf{x}(t_0)]^2).$$

Independently the TLM,  $\mathcal{L}$  can be integrated to generate,

$$\delta \mathbf{x}^n = \mathcal{L}[\delta \mathbf{x}(t_0)].$$

Let us consider the following two ratios,

$$f(\alpha) = \frac{\|\Delta \mathbf{x}\|}{\|\delta \mathbf{x}^n\|},$$

and

$$r(\alpha) = \frac{\|\Delta \mathbf{x}\| - \|\delta \mathbf{x}^n\|}{\|\delta \mathbf{x}^n\|}.$$

Since  $\Delta \mathbf{x}$  must be very well approximated by the result of integration of our TLM,  $\delta \mathbf{x}^n$ , which implies that as  $\alpha$  decreases,  $f(\alpha)$  must tend to 1.0 and  $r(\alpha)$  (see equations 4 and 5) must tend to 0%. Indeed integration of our TLM for 1 day (the validity of the TLM for the shallow water equations model is at-most 24-36 hours) indicates such a behavior as shown by figure 5. For integrating the TLM, we have used the same value of the time-step as was used to integrate the non-linear model.

Remark:

The norms in the above equations are weighted (total) energy norms, defined by,  $W$  which has been chosen to be a block diagonal matrix, such that the velocity components are weighted by the unit matrix,  $\mathbf{I}$ , and the height field by  $W_1$ , such that  $W_1 = w_1 * \mathbf{I}$ .

$$w_1 = g * \left[ \frac{1}{N_x \cdot N_y} \sum_{i,j} |h_{i,j}| \right]^{-1},$$

where  $N_x = 128$ , and  $N_y = 65$  cells, and acceleration due to gravity,  $g \approx 9.81$ . Infact if  $\mathbf{h}, \mathbf{u}$ , and  $\mathbf{v}$  are respectively the vectors of height and velocity fields, then the weighted norm,

$$\|\mathbf{x}\|_w^2 = w_1 \|\mathbf{h}\|^2 + \|\mathbf{u}\|^2 + \|\mathbf{v}\|^2,$$

which is proportional to the total energy. In the above equation,  $\|\underline{x}\|^2 = \sum_{i,j} x_{i,j}^2$ .

Figures 6 and 7 provide the flowchart of the TLM (all the linearized variables have the same nomenclature as their non-linear counterparts, and have "g\_" preceding the variable, for e.g., if  $z$  is a non-linear variable, the corresponding linearized variable would be  $g_{-z}$ ).

## 4 Adjoint Model

It may be of interest that to our knowledge we have not come across an adjoint model that is based on flux-form conserving advection schemes. Also transposing the statements, in other words, deriving the adjoint statements for the PPM and van Leer advection schemes has been laborious and time consuming.

The adjoint model satisfies two properties. It is the transpose of the TLM and also it provides us the means to obtain the gradient of a cost functional (which uses the prognostic variables of the model) with respect to the model initial conditions.

#### 4.1 Transpose Check

This test checks whether the TLM and the adjoint model are transpose of each other. Let  $Q$  represent the input to the linearized model,  $A$  be the linearization operator. Then  $A^T$  represents the adjoint operator. Hence  $AQ$  is the output of the linearized model, which leads to the following identity:

$$(AQ)^T(AQ) = Q^T[A^T(AQ)]. \quad (6)$$

It is obvious from the above equation that the left hand side (lhs) is entirely from the TLM, whereas the right hand side (rhs) has a contribution from the adjoint model.

We have checked our adjoint model for the satisfaction of the above property for all the model subroutines. The above equality is satisfied up-to 12 decimal digits for all of them.

#### 4.2 Gradient check

This check is based on a Taylor series expansion of the cost functional, accurate up to first order,

$$\mathcal{J}(\underline{x} + \epsilon \underline{\nabla} \mathcal{J}) = \mathcal{J}(\underline{x}) + \epsilon \underline{\nabla} \mathcal{J}^T \underline{\nabla} \mathcal{J} + O(\epsilon^2), \quad (7)$$

where  $\epsilon$  is a scalar, and  $\mathcal{J}$  is the cost functional based upon the  $\mathbf{u}$ ,  $\mathbf{v}$ ,  $\mathbf{h}$  fields.

Therefore, as  $\epsilon \rightarrow 0$ , we expect that

$$\gamma(\epsilon) = \frac{\mathcal{J}(\underline{x} + \epsilon \underline{\nabla} \mathcal{J}) - \mathcal{J}(\underline{x})}{\epsilon \|\underline{\nabla} \mathcal{J}\|^2} \rightarrow 1, \quad (8)$$

where the euclidean norm is used for evaluating  $\|\cdot\|^2$ . Also we have used the weighted energy norm described in section 3.1 to evaluate the above norm.

In Figures 8- 12, we provide the verification of the equation (8) for various advection schemes using both norms. Hence, our adjoint model indeed provides us the gradient of the cost functional  $\mathcal{J}$  with respect to the initial conditions.

Figures 13 and 14 provides a flowchart of the adjoint model (all the adjoint variables have their names preceded by "ad", for e.g., if  $x$  is a non-linear variable, then the corresponding adjoint variable would be  $adx$ ); the final values of the adjoint variables provide us the gradient of the cost functional with respect to the initial condition.

Efforts to use the adjoint model for various data assimilation and mathematical applications are underway. We expect to submit for publication such studies shortly.

### **Acknowledgments:**

We would like to thank Drs. S.-J. Lin and R. B. Rood for providing us their spherical shallow-water finite volume model.

The authors acknowledge the support from the NSF grant number ATM-9731472 managed by Dr. Linda Peng whom we would like to thank for her support.

### **Appendix A**

1. Declare all the integer constants: (all of these are to be input)

IMR-resolution of grid along longitudinal direction (inside rslv.com),

JMR-resolution of grid along latitudinal direction (inside rslv.com),

n.time\_steps- total number of time steps, for e.g., at a resolution of 128 X 64 cells, 600 seconds time step, for a 1 day run, n.time\_steps = 144,

IREAD and IOUT- input and output files respectively,

IORD and JORD- advection scheme on D-grid, along the longitude and latitudes respectively,

ICD and JCD- advection scheme on C-grid, along the longitude and latitudes respectively,

NDT- time step in seconds, NHT- time step for C-grid, NHT= NDT/2.

2. Declare all the geometric constants and variables: (these are calculated within the driver: drv.f or by subroutine, calling setrig.f)

PI -  $\pi$ , Grav- $g$ , acceleration due to gravity, AE- radius of the earth, OMEGA-angular velocity of the earth,

DP- increment along latitude,  $\Delta\theta$ , DL- increment along longitude,  $\Delta\lambda$ ,

$RDY = \frac{1}{AE*DP}$ , RCAP, ACAP, RDX- scaled polar cap areas (see drv.f)

UMAX- magnitude of maximum velocity, that is expected ( a suitable upper bound),

arrays, SINE, COSP, SINP- all of length JNP, all are calculated inside subroutine setrig.f, they store sines and cosines of the latitudes.

arrays SINLON, COSLON- both of length IMR, are calculated inside drv.f, they store sines and cosines of the longitudes.

hs- array of dimension (IMR,JNP), surface height field, it is a constant (based on bottom topography, input from either IREAD or appropriate initial conditions),

$f_0$  and  $f_c$ - arrays of dimension (IMR,JNP), coriolis parameters either at cell centers or cell corners (constants, calculated inside drv.f)

3. Time integrated variables:

u- array of dimension (IMR,JNP), U-wind, on D-grid,

v- array of dimension (IMR,JNP), V-wind, on D-grid,  
h- array of dimension (IMR,JNP), height field, is not updated and neither it is used on C-grid (instead *hm* is used as a proxy for *h* on the C-grid),  
vort- array of dimension (IMR,JNP), vorticity field (not a prognostic variable), initial value of the vorticity is due to the coriolis force.  
uc, vc- arrays of dimension (IMR,JNP) are the U- and V- winds on the C-grid,  
ua, va- arrays of dimension (IMR,JNP) are the U- and V- winds on the A-grid,

4. Subroutines called by the driver (drv.f):

SUBROUTINE setrig- to obtain the values of DP,DL,COSP,COSE,SINP,SINE using values of IMR and JNP,

If the initial conditions (u,v,h) are read from a file, IREAD, then they are to be read on the A-grid, these variables are then interpolated to D-grid using SUBROUTINE atod.

SUBROUTINE call\_sw uses all the constants and parameters and, the initial conditions to integrate the variables (u,v,h,vort) forward in time up-to the n.time\_steps.

SUBROUTINE swout is used to output the variables in binary format (to be visualized using sw.ctf, for GrADS), the output file is controlled by the integer IOUT.

5. SUBROUTINE call\_sw:

The input to SUBROUTINE call\_sw are all the constants, parameters and initial values of all dependent variables (from initial condition). Call\_sw updates these initial values of *u, v, h, vort* to the final time of integration, in the following manner,

obtains the array *hm*, which serves as a proxy to *h* within the scope of the SUBROUTINE ccore,

filters *hm* using the polar Fourier filter, by calling SUBROUTINE pft0 (the SUBROUTINES within filter.f serve as an interface between the FFT routines of ecmfft.f and call\_sw.f; pft0 is one of the subroutines inside filter.f),

for the first half of the time step, obtain the values of the mid-time level prognostic winds: UC and VC on the C-grid, variables *u, v, h* are not changed after a call to ccore; only UC, VC and vort are updated,

in the other half of the time step, calls the SUBROUTINE dcore, which uses the (above updated values of) UC, VC and *vort* to update *u, v, h, vort* on the D-grid; SUBROUTINE dcore does not change values of UC and VC.

Figures 1 and 2 provide a flow chart of the forward model.

## References

- [1] S.-J. Lin and R. B. Rood: An explicit flux-form semi-Lagrangian shallow-water model on the sphere, Q. J. R. Meteorological society, volume 123, pp. 2477-2498, 1997.
- [2] S.-J. Lin and R. B. Rood: Multidimensional flux-form semi-Lagrangian transport schemes, Monthly Weather Review, volume 124, pp. 2046-2070, 1996.
- [3] Lin, S.-J., Chao, W. C., Sud, Y.C. and Walker, G. K.: A class of the van Leer transport schemes and its applications to the moisture transport in a general circulation model. Monthly Weather Review, volume 122, pp. 1575-1593, 1994.
- [4] Rood, R.B: Numerical advection algorithms and their role in atmospheric transport and chemistry models, Reviews of geophysics, volume 25, pp. 71-100, 1987.
- [5] S.-J. Lin, A finite-volume integration method for computing pressure gradient force in general vertical coordinates, Q. J. R. Meteorological society, volume 123, pp. 1749- 1762, 1997.
- [6] Suarez, M.J. and Takacs, L.L: Documentation of the ARIES/GEOS dynamical core: version 2, NASA Technical Memorandum 104606, volume 5.
- [7] Tangent and Adjoint Model Compiler, <http://www.autodiff.com/tamc>
- [8] D. L. Williamson, J.B. Drake, J.J. Hack, Rudiger Jakob and P.N. Swarztrauber: A Standard Test Set for Numerical Approximations to the Shallow Water Equations on the Sphere, Journal of Computational Physics, volume 102, pp. 211-224, 1992.
- [9] Kalnay, E.: Atmospheric Modeling, Data Assimilation and Predictability, Cambridge University Press, 2003.



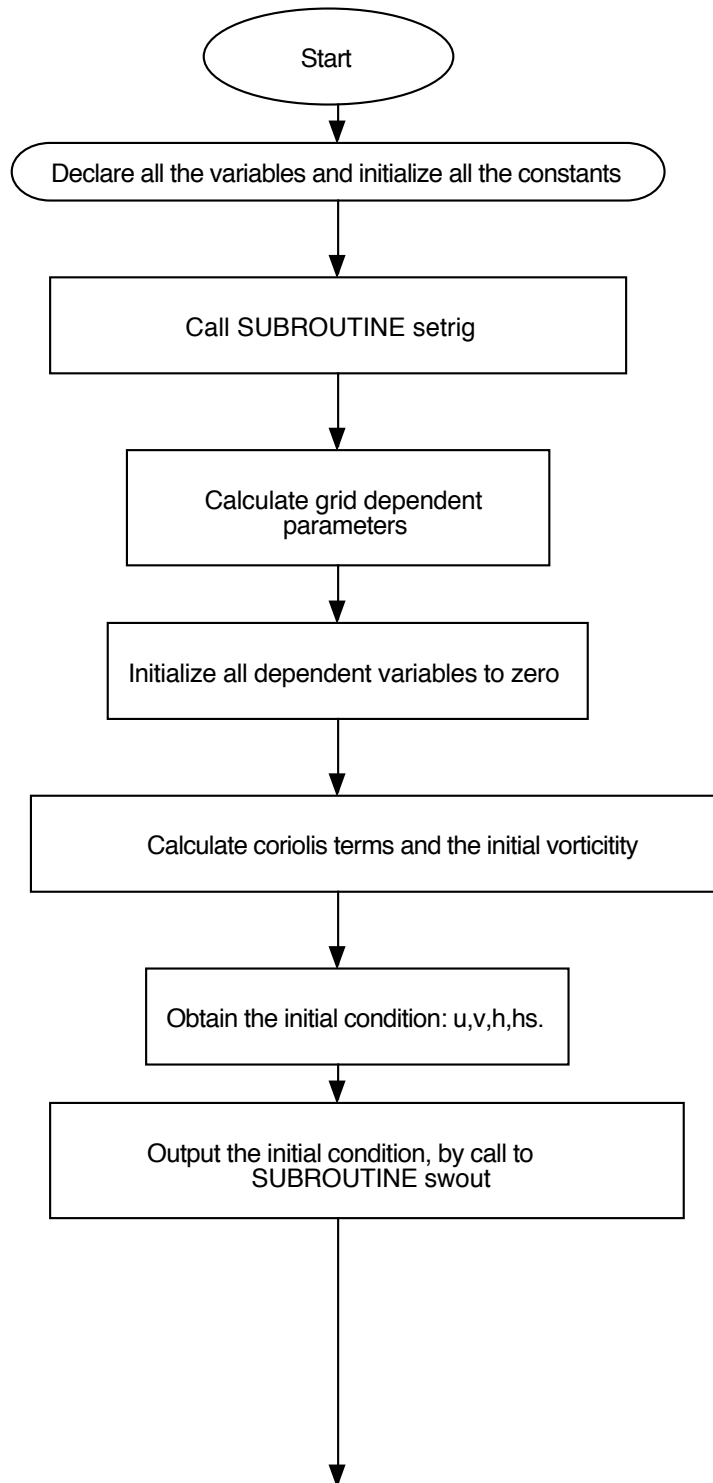


Figure 1: Flow chart of the forward model

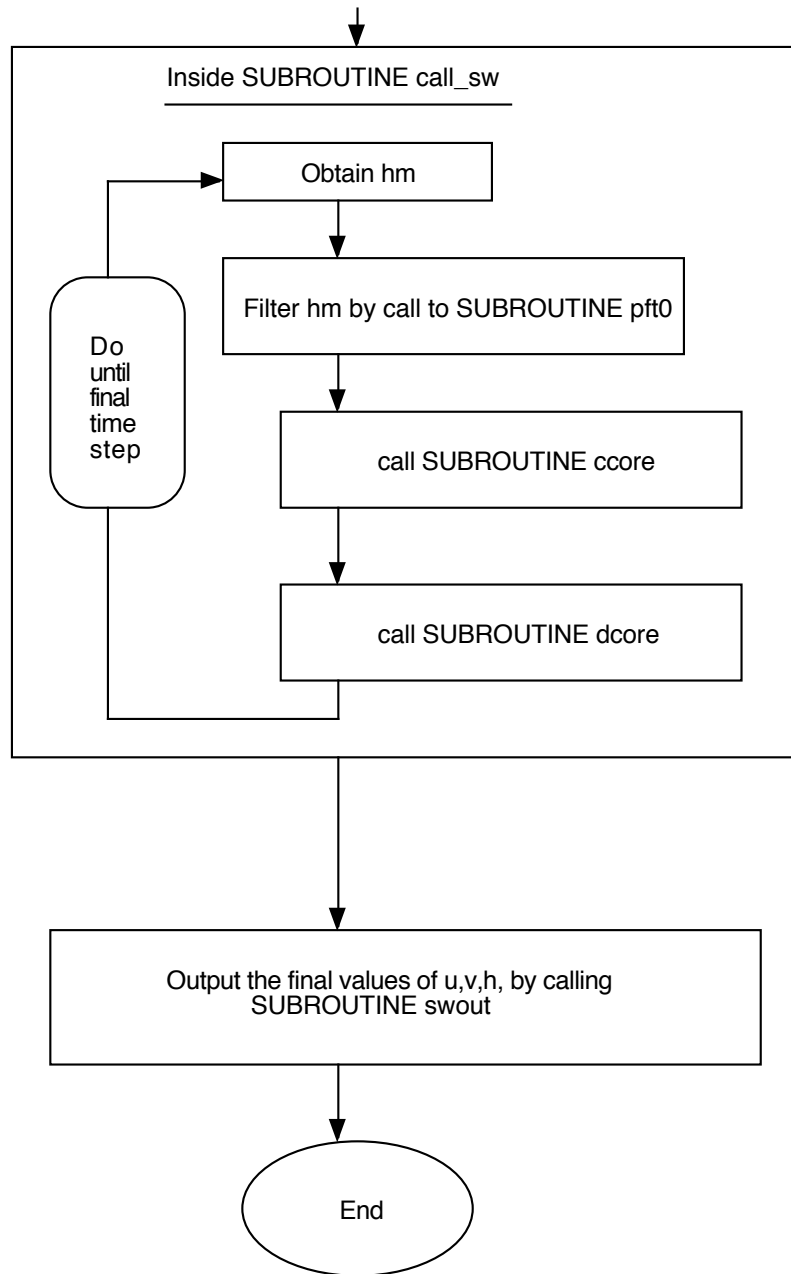


Figure 2: Flow chart of the forward model (continued)

initial height field

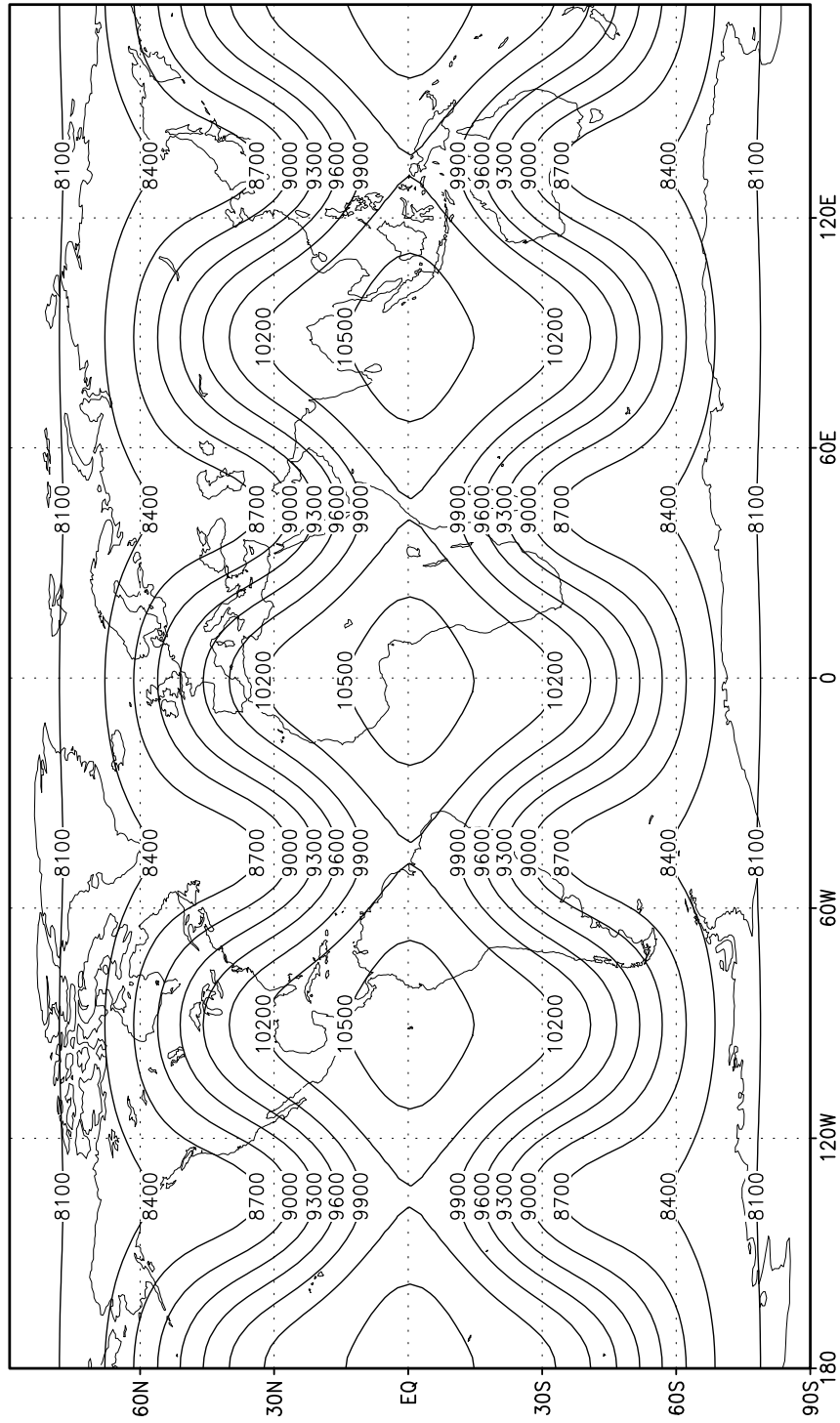


Figure 3: Initial height field

# DAY-1 height field

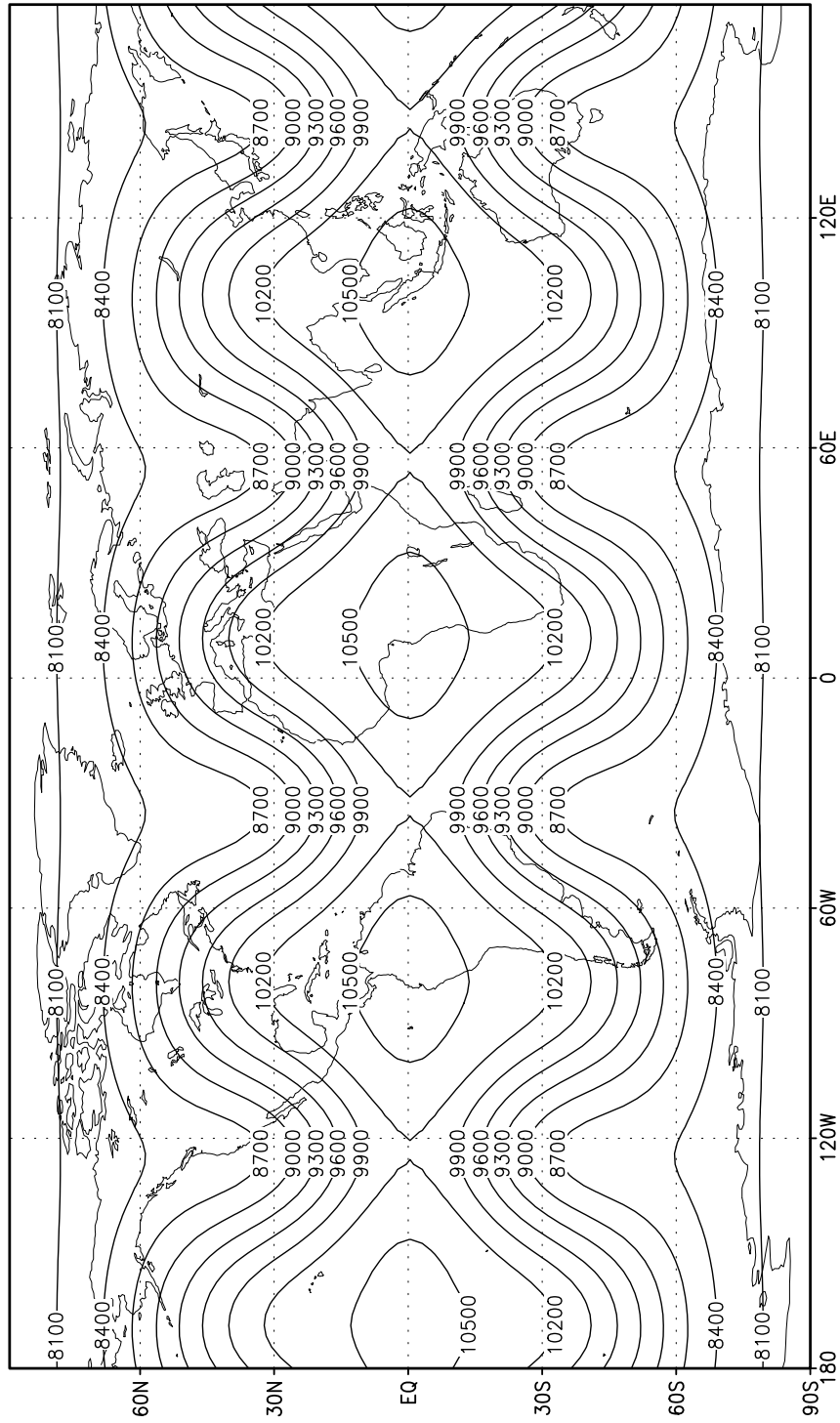


Figure 4: DAY-1 height field

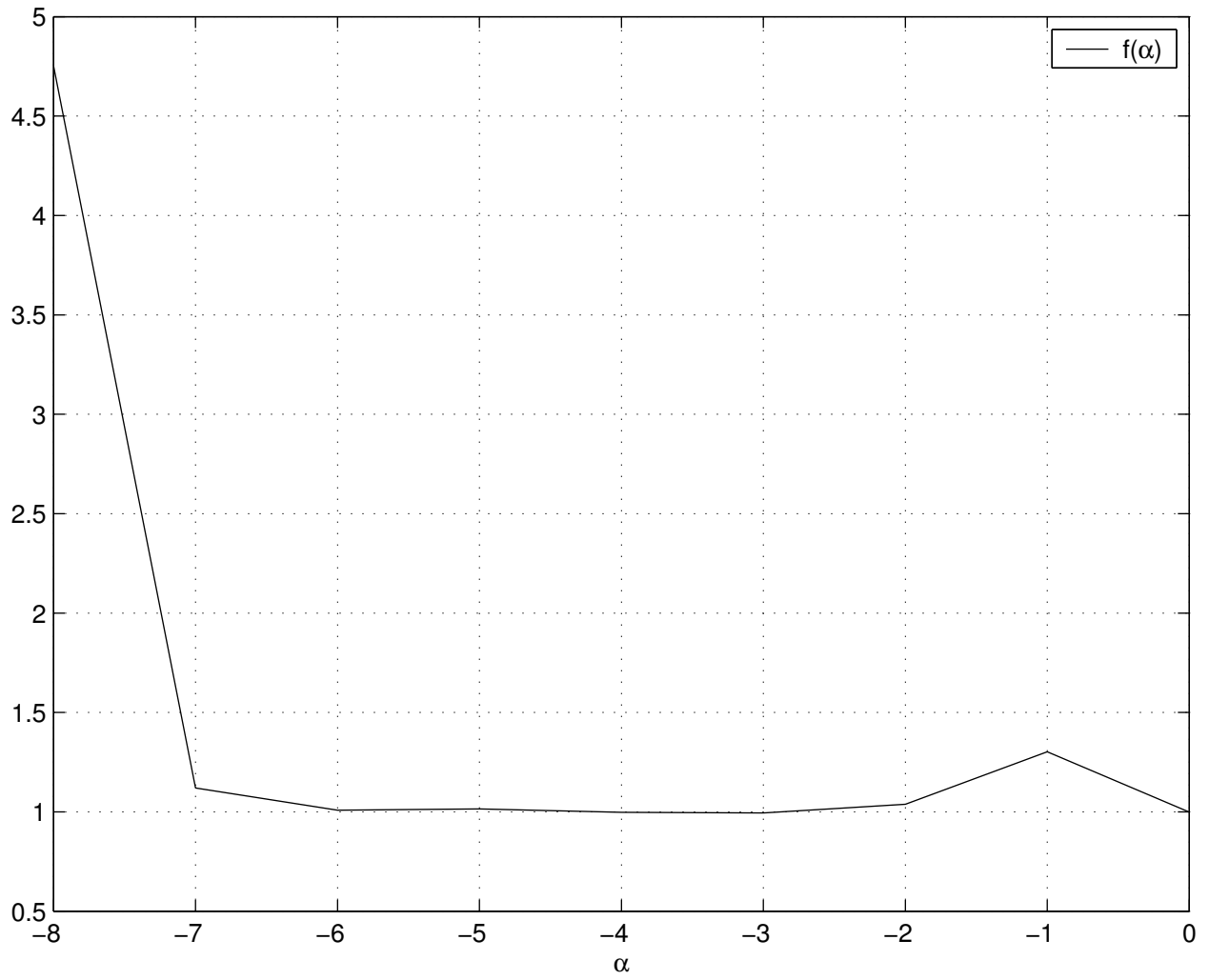


Figure 5: Verification of the TLM

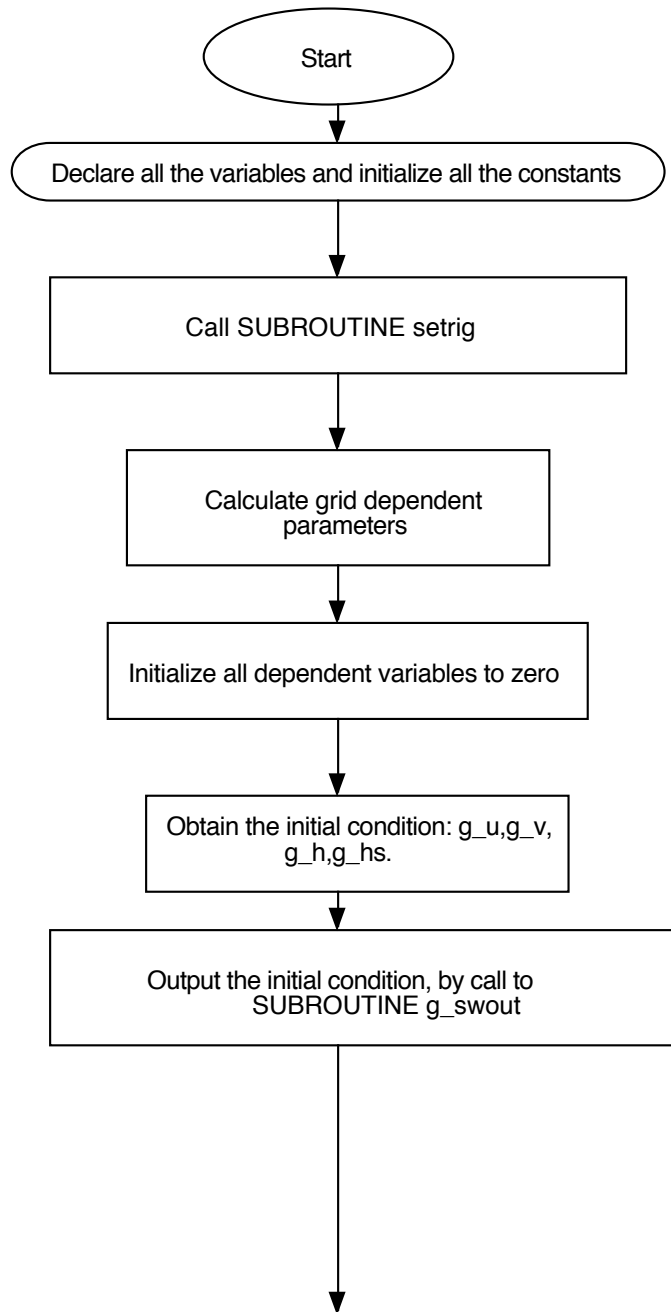


Figure 6: Flow chart of the TLM

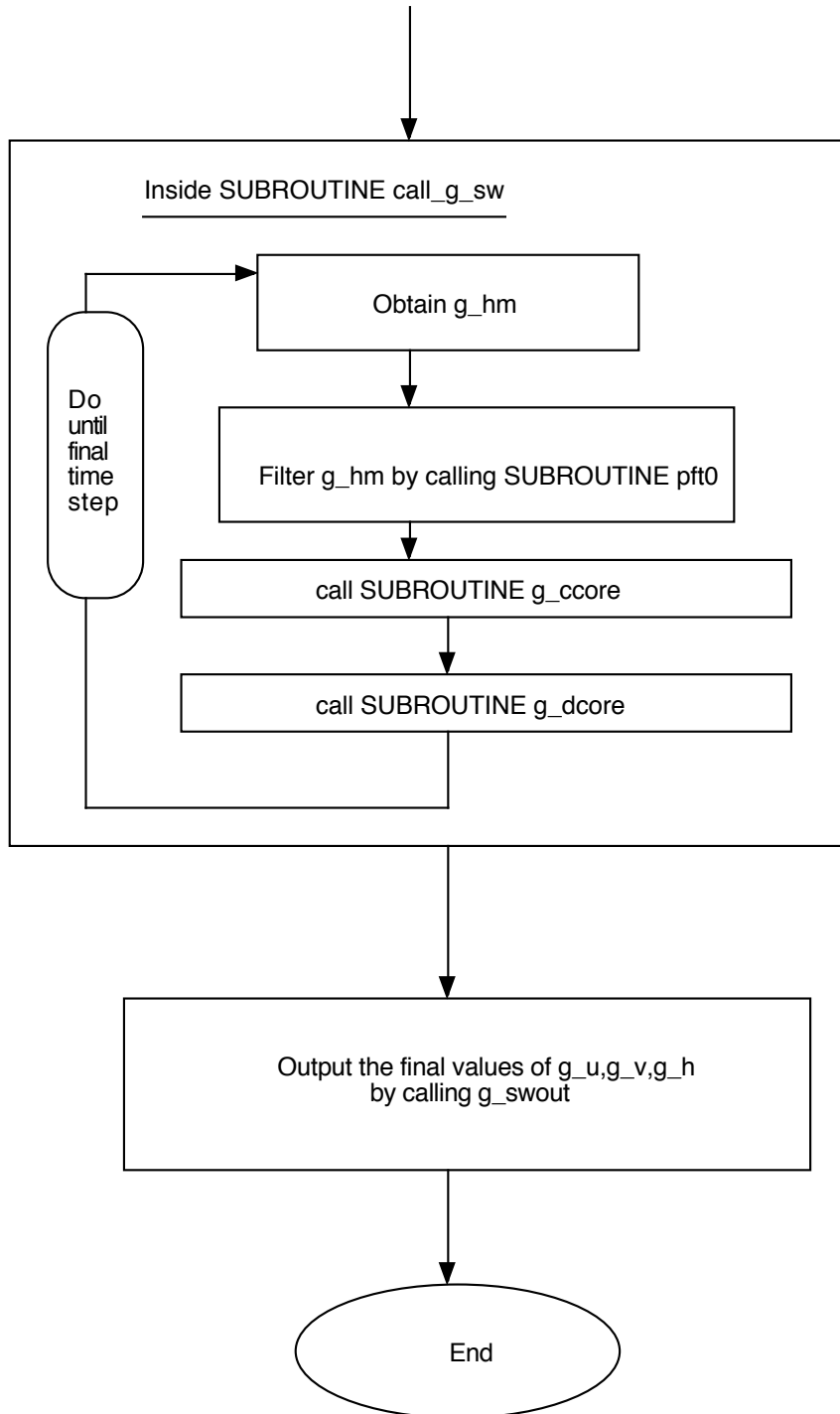


Figure 7: Flow chart of the TLM (continued)

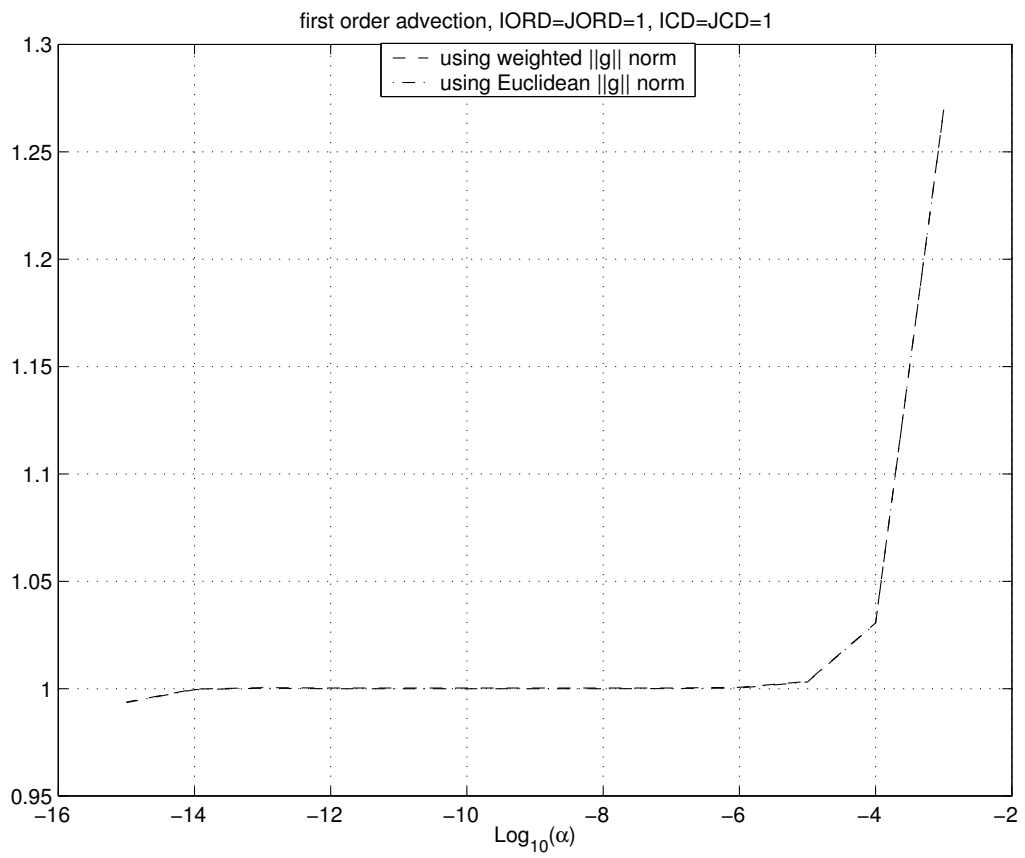


Figure 8: Gradient verification for first order advection scheme



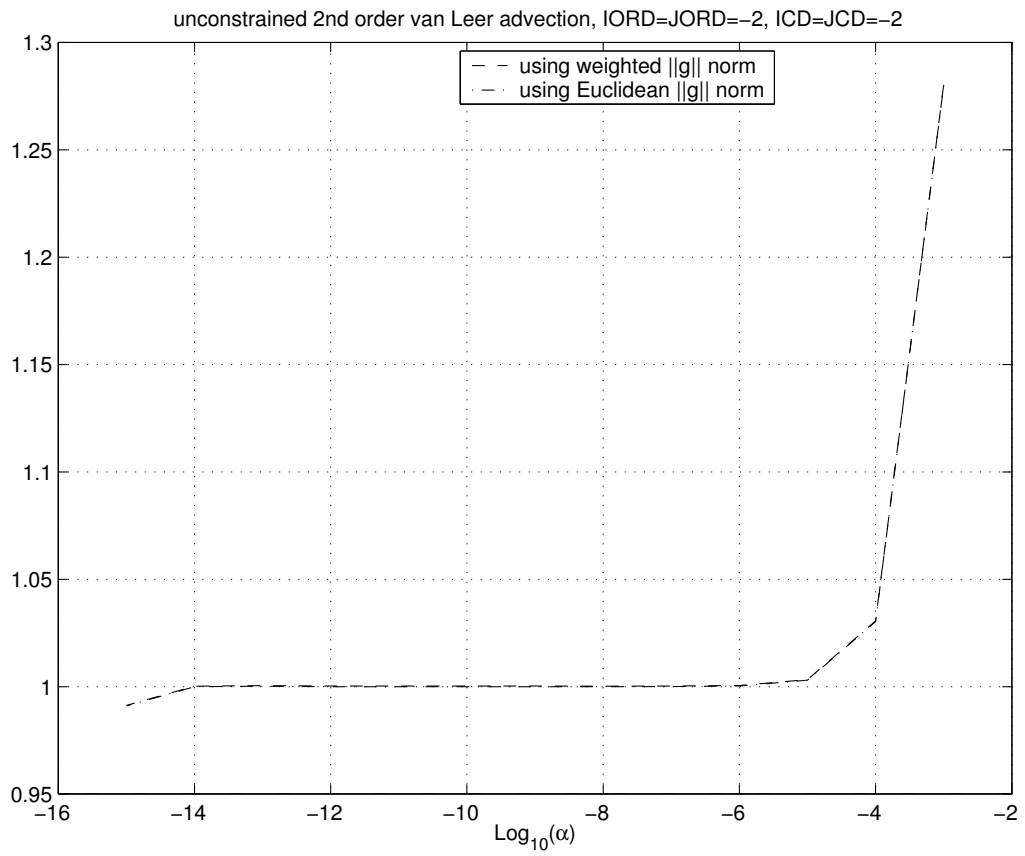


Figure 9: Gradient verification for unconstrained van Leer advection scheme

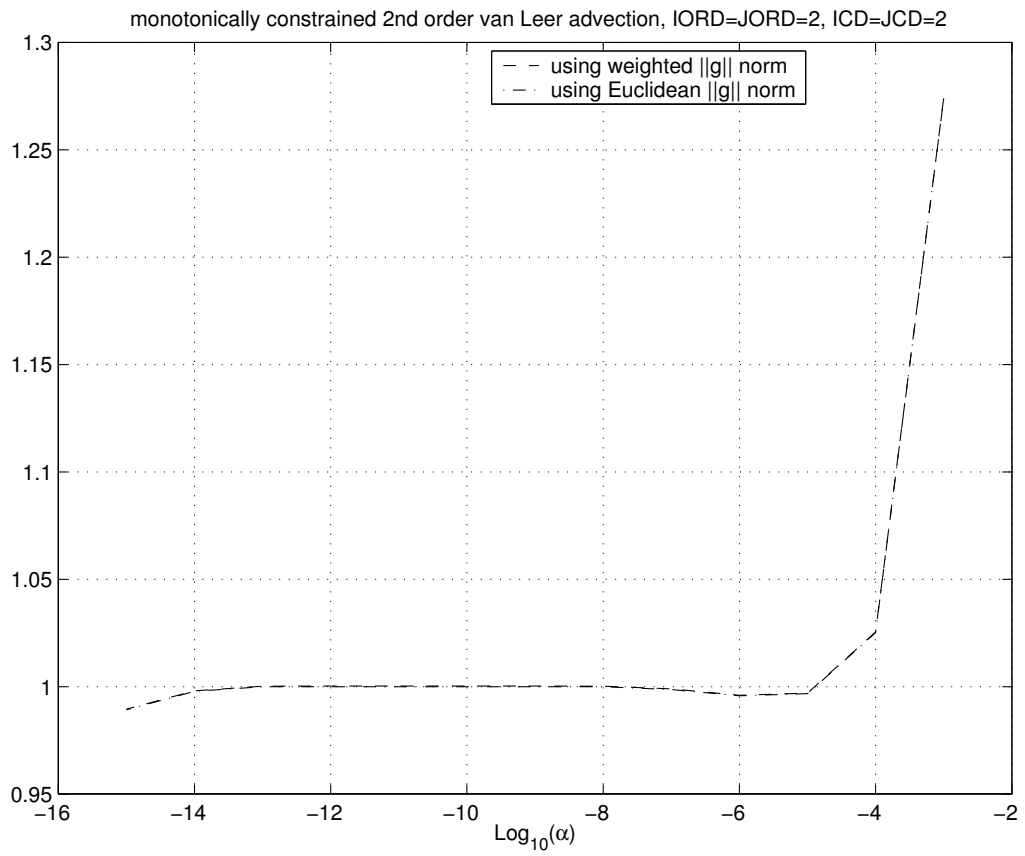


Figure 10: Gradient verification for constrained van Leer advection scheme

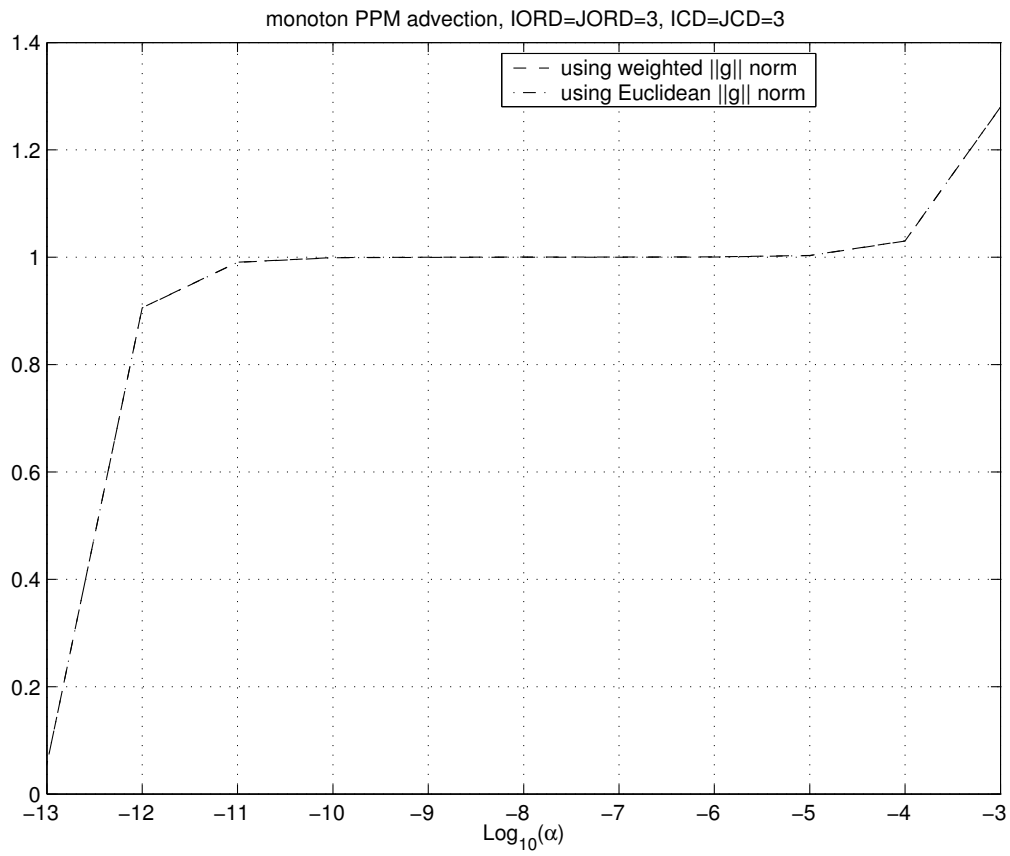


Figure 11: Gradient verification for monotonically constrained PPM advection scheme

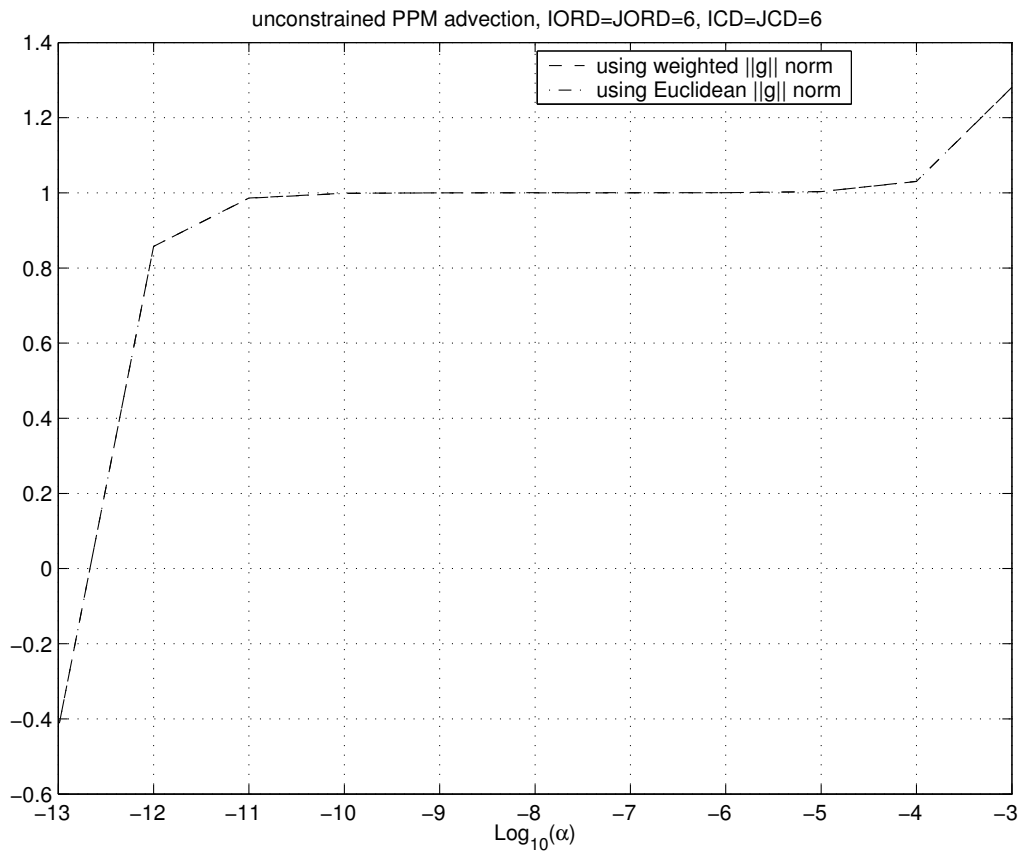


Figure 12: Gradient verification for unconstrained PPM advection scheme

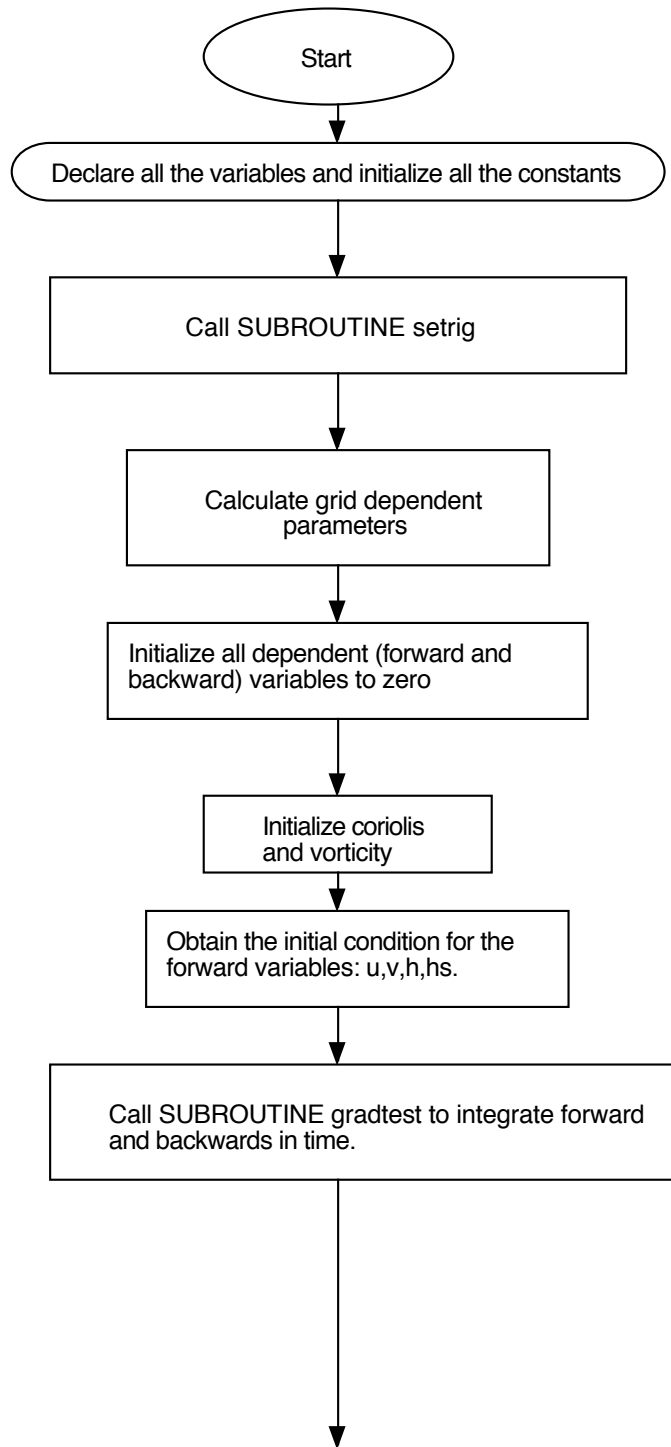


Figure 13: Flow chart of the adjoint model

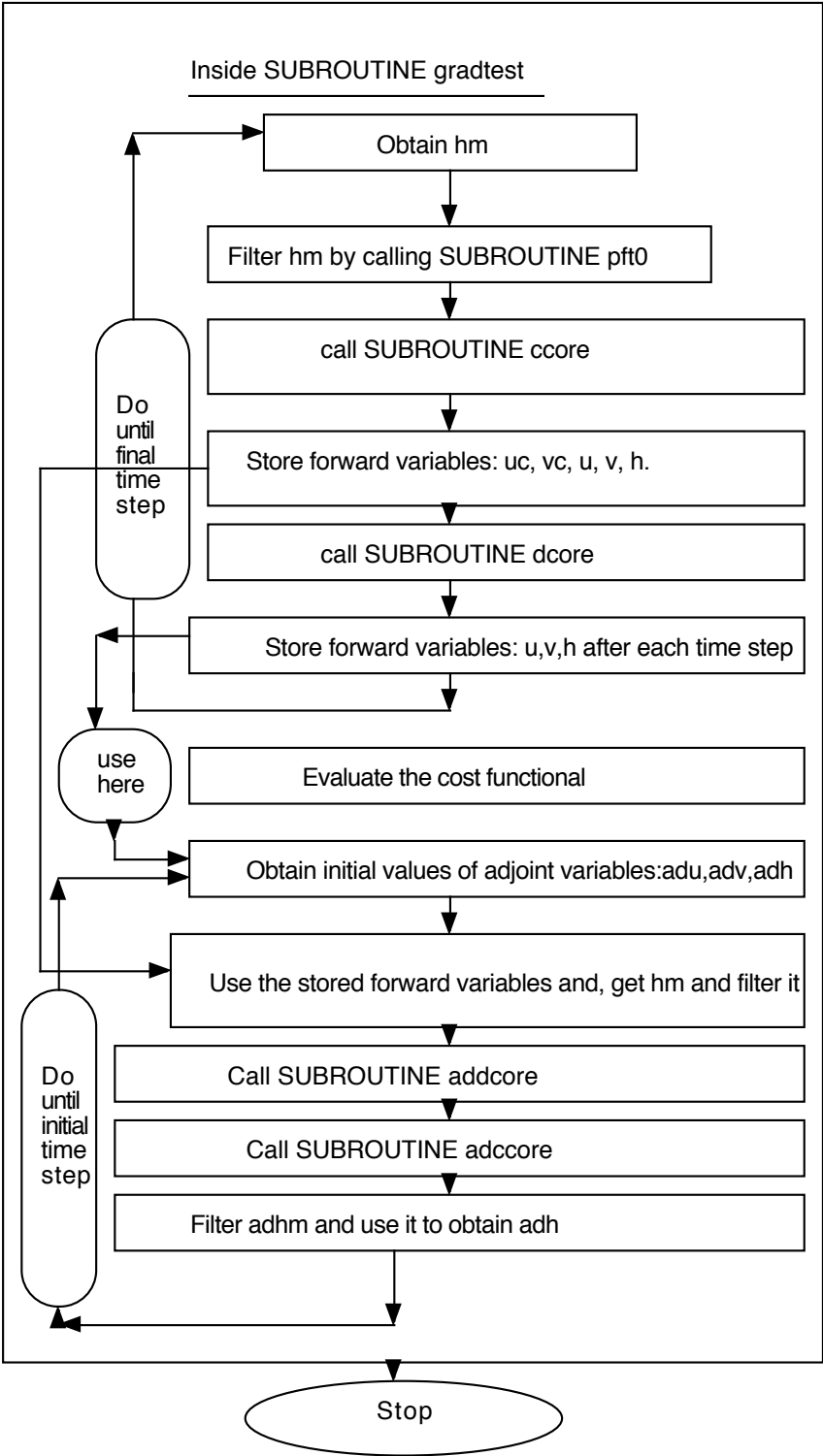


Figure 14: Flow chart of the adjoint model(continued)