Documentation of the Tangent Linear and Adjoint Models of New MPI Version of the FSU Global Spectral Model

Zhuo Liu and I.Michael Navon
School of Computational Science and Information Technology
Florida State University
Tallahassee, FL 32306

22 March 2002

1 Introduction

Due to its advantages over the optimal interpolation and 3-D data assimilation and considering the ever increasing types of observational data, especially the asynoptic data, 4-D variational data assimilation has attracted more and more attention during the past years. It was first implemented by Courtier (1985), Derber (1985), Lewis and Derber (1985), LeDimet and Talagrand (1986), Courtier and Talagrand (1987) and Zou, et al. (1990) on shallow water models as well as quasi-geostrophic models. Later, this method was applied to more complex models by Thépaut and Courtier (1991), Navon, et al. (1992), Chao and Chang (1992), just to name but a few. The forecast models are assumed to be perfect in these studies. Derber (1989) developed a variational continuous assimilation method to take into account the forecast model error. Recently, further studies have been carried out with special emphasis on the physical processes of the NWP model(Li, Navon and Zhu 2000).

The objective of the 4-D variational data assimilation is to fit the model forecasts to the observations over some time and space interval. It can combine the previous data with currently available data in the full model dynamics and imposes no restriction on the data type using used in the data assimilation and simply employs these data in a straightforward manner. It is possible, therefore, to utilize the available observation data, especially asynoptic data, as much as possible, and to retrieve information not only of the variables whose observations are available but also of the related variables via the full model dynamics.

The FSU Global Spectral Model (FSU GSM) has been successfully applied for the numerical weather forecast, especially for the tropics. The effort to establish a 4-D variational data assimilation system was begun with the FSU spectral model in 1993. The adjoint code of the dry adiabatic version of the FSU GSM was developed by Zhi Wang (Wang 1993). Later, Tsuyuki (1996a, 1996b, 1997) incorporated the moisture variable, the smoothed parameterization of moist process, horizontal diffusion and a simplified surface friction, and carried out the experiments using the precipitation data. The work was continued to incorporate the radiation processes and the boundary layer processes into the data assimilation system(see Yanqiu Zhu and Navon, 1998, Li, Navon and Zhu, 2000) because of the important roles they play in simulating various large-scale and mesoscale phenomena, especially in the tropical weather system. This effort will make the adjoint model consistent with the nonlinear forecast model. Moreover, the adjoint technique allows using the variational approach for assimilating various types of observations in meteorology, including satellite radiation observations, to be treated exactly and at a reasonable cost. For instance, outgoing longwave radiation (OLR) currently is commonly used to retrieve precipitation information; however, the limitations of this algorithm degrade the accuracy of the retrieved data. The variational approach makes it possible to assimilate this type of data directly, instead of its being used to retrieve other meteorological quantities.

In this study, a T126L14 of MPI version of FSU GSM is used, i.e., the horizontal resolution is of a triangular truncation type with total wavenumber 126 and 14 levels in the vertical. The physical processes include orography, planetary boundary layer processes, vertical diffusion, dry adjustment, large-scale condensation and evaporation, deep cumulus condensation, horizontal diffusion and radiation processes. For a detailed description of FSU GSM, see T. N. Krishnamurti et al (1988). The model is run on a IBM SP3 machine with MPI parallel environment.

As a first stage, we have coded the adjoint of the dynamical core of the model(MPI version). We are in the process of deriving the code of the full physics version of the model.

2 Description of FSU Global Spectrum Model

2.1 Model Description

The FSUGSM is a global hydrostatic primitive equation model. The prognostic variables are vorticity, divergence, virtual temperature, moisture and log surface pressure. The model uses the spectral technique in the horizontal direction, and second order finite difference in the vertical. The wave number truncation used ranges from T42 to T63 for climate simulations and from T126 to T170 for real time forecasts. Higher resolutions may be used for research purposes. A σ coordinate is used in the vertical. The model physics include long and shortwave radiation, boundary layer processes, large scale precipitation, shallow and deep cumulus convection. The GCM physics routines have been developed to adhere to the plug-compatible guidelines developed by the U.S. Modeling Infrastructure Working Group. As a result, the model has available to it a large array of selectable physical parameterizations. Currently six cumulus parameterizations have been implemented, along with three radiation packages and a couple of boundary layer schemes. Most of the NCAR CCM 3.6 atmospheric physics has been incorporated as an option in the model. In addition to the current simple land surface scheme, two detailed land biosphere schemes are in the process of being implemented: BATS and SSiB.

2.2 Governing Equations

In FSU GSM, the σ vertical coordinate is defined as

$$\sigma = p/p_s \tag{1}$$

where p is the pressure and p_s the surface pressure. Thus, $\sigma = 0$ is at the top of the atmosphere and $\sigma = 1$ at the earth's surface. The boundary conditions are $\dot{\sigma} = 0$ at $\sigma = 1$ and $\sigma = 0$. The original governing equations of FSU GSM are as follows.

the vorticity equation:

$$\frac{\partial \zeta}{\partial t} = -\nabla \cdot (\zeta + f)\mathbf{V} - \mathbf{k} \cdot \nabla \times (RT\nabla q + \dot{\sigma}\frac{\partial \mathbf{V}}{\partial \sigma} - \mathbf{F}) \tag{2}$$

the divergence equation:

$$\frac{\partial D}{\partial t} = \mathbf{k} \cdot \times(\zeta + f)\mathbf{V} - \nabla \cdot (RT\nabla q + \dot{\sigma}\frac{\partial \mathbf{V}}{\partial \sigma} - \mathbf{F}) - \nabla^2(\phi + \frac{\mathbf{V} \cdot \mathbf{V}}{2}) \tag{3}$$

the thermodynamic equation:

$$\frac{\partial T_v}{\partial t} = -\nabla \cdot \mathbf{V}T_v + T_v D + \dot{\sigma}\gamma - \frac{RT_v}{C_p}(D + \frac{\partial \dot{\sigma}}{\partial \sigma}) + H_T \tag{4}$$

the continuity equation:

$$\frac{\partial q}{\partial t} = -D - \frac{\partial \dot{\sigma}}{\partial \sigma} - \mathbf{V} \cdot \nabla q \tag{5}$$

the hydrostatic equation:

$$\sigma\frac{\partial \phi}{\partial \sigma} = -RT_v \tag{6}$$

the moisture equation:

$$\frac{\partial S}{\partial t} = -\nabla \cdot \mathbf{V}S + SD - \dot{\sigma}\frac{\partial S}{\partial \sigma} + H_T - H_M - [\frac{RT}{C_p} - \frac{RT_d^2}{\epsilon L(T_d)}]\ [D + \frac{\partial \dot{\sigma}}{\partial \sigma} - \frac{\dot{\sigma}}{\sigma}] \tag{7}$$

In the above equations, the terms are defined as follows:

- f = Coriolis parameter,
- $\mathbf{V}$ = horizontal vector wind,
- ζ = vertical component of vorticity = $\mathbf{k} \cdot \nabla \times \mathbf{V}$,
- D = horizontal divergence = $\nabla \cdot \mathbf{V}$,
- T is the absolute temperature,
- T_v is the virtual temperature,
- $q = lnp_s$,
- γ = static stability = $\frac{RT}{C_P\sigma} - \frac{\partial T}{\partial \sigma}$,
- $\dot{\sigma}$ = vertical velocity in sigma coordinates,
- $= (\sigma - 1)\ (\hat{D} + \hat{\mathbf{V}} \cdot \nabla q) + \hat{D}^\sigma + \hat{\mathbf{V}}^\sigma \cdot \nabla q$,

- $\phi =$ geopotential height,
- $\mathbf{F}$ is the horizontal frictional force per unit mass,
- $H_T =$ the diabatic heating,
- R is the gas constant for dry air,
- C_P is the specific heat of dry air at constant pressure,
- T_d is the dewpoint temperature,
- $S = T - T_d$ is the dewpoint depression,
- ϵ is the ratio of the molecular weight of water vapor to effective molecular weight of dry air (0.622),
- $L(T_d)$ is the latent heat of vaporization of water or ice,
- H_M represents moisture sources or sinks,
- $\hat{F} =$ integral operator $=\int_0^1 Fd\sigma$,
- $\hat{F}^\sigma$ =integral operator $= \int_\sigma^1 Fd\sigma$.

Following Robert (1966) the horizontal wind velocity u and v components are converted into scalars $U = u\cos\theta/a$ and $V = v\cos\theta/a$, where a is the radius of the earth, u is the zonal wind component, v is the meridional wind component, θ is the latitude and λ is the longitude. The temperature field and γ are decomposed into their initial horizontal means and deviations respectively, i.e., $T = T^* + T'$ and $\gamma = \gamma^* + \gamma'$ for the semi-implicit algorithm.

Furthermore, with the definition of a horizontal differential operator α which is a function of A and B:

$$\alpha(A, B) = \frac{1}{\cos^2\theta}[\frac{\partial A}{\partial\lambda} + \cos\theta\frac{\partial B}{\partial\theta}] \tag{8}$$

the governing equations (2)-(7) become:

$$\frac{\partial\zeta}{\partial t} = -\alpha(A, B) \tag{9}$$

$$\frac{\partial D}{\partial t} + \nabla^2(\phi + RT_v^* q) = \alpha(B, -A) - a^2\nabla^2 E \tag{10}$$

$$\frac{\partial T_v}{\partial t} - \gamma^*\dot{\sigma} - \frac{RT^*}{C_p}\frac{\partial q}{\partial t} = -\alpha(UT', VT') + B_T \tag{11}$$

$$\frac{\partial q}{\partial t} + \hat{G} + \hat{D} = 0 \tag{12}$$

$$\frac{\partial S}{\partial t} = -\alpha(US, VS) + B_s \tag{13}$$

where

$$A = (\zeta + f)U + \dot{\sigma}\frac{\partial V}{\partial \sigma} + \frac{RT'}{a^2}\cos\theta\frac{\partial q}{\partial \theta} - \cos\theta\frac{F_\theta}{a} \tag{14}$$

$$B = (\zeta + f)V + \dot{\sigma}\frac{\partial U}{\partial \sigma} + \frac{RT'}{a^2}\frac{\partial q}{\partial \lambda} - \cos\theta\frac{F_\lambda}{a} \tag{15}$$

$$G = \frac{1}{\cos^2\theta}[U\frac{\partial q}{\partial \lambda} + V\cos\theta\frac{\partial q}{\partial \theta} \tag{16}$$

$$\frac{E = (U^2 + V^2)}{2\cos^2\theta} \tag{17}$$

$$B_T = T'D + \gamma'\dot{\sigma} - \frac{RT'}{C_p}(\hat{G} + \hat{D}) + \frac{RT}{C_p}G + H_T \tag{18}$$

$$\dot{\sigma} = (\sigma - 1)(\hat{G} + \hat{D}) + \hat{G}^\sigma + \hat{D}^\sigma \tag{19}$$

$$B_s = SD - \dot{\sigma}\frac{\partial S}{\partial \sigma} + [\frac{RT_v}{C_p} - \frac{RT_d^2}{\epsilon L(T_d)}]\ [\frac{\partial \dot{\sigma}}{\partial \sigma} + G - \hat{G} - \hat{D}] + H_T - H_M \tag{20}$$

The dependent variables are expanded in the triangularly truncated series of spherical harmonics as following

$$F = \sum_{m=-J}^{J}\sum_{l=|m|}^{J} F_l^m Y_l^m \tag{21}$$

where F_l^m are complex expansion coefficients, functions of σ and t.

By introducing the following two new variables

$$P = \phi + RT^*q \tag{22}$$

and

$$W = \dot{\sigma} - \sigma(\hat{G} + \hat{D}) \tag{23}$$

the spectral form of the governing equations assumes the form

$$\frac{\partial \zeta_l^m}{\partial t} = -\{\alpha(A, B)\}_l^m \tag{24}$$

$$\frac{\partial D_l^m}{\partial t} - a^{-2}l(l+1)P_l^m = \{\alpha(B, -A) - a^2\nabla^2 E\}_l^m \tag{25}$$

$$\sigma\frac{\partial^2 P_l^m}{\partial \sigma \partial t} + R\gamma^* W_l^m = R\{\alpha(UT', VT') - B_T\}_l^m \tag{26}$$

$$\frac{\partial W_l^m}{\partial \sigma} + D_l^m = \{B_w\}_l^m \tag{27}$$

$$\frac{\partial q_l^m}{\partial t} - {W_s}_l^m = 0 \tag{28}$$

$$\frac{\partial S_l^m}{\partial t} = \{-\alpha(US, VS) + B_s\}_l^m \tag{29}$$

where B_T, B_s, B_w, UT', VT', SU, SV, E, A, B and $\alpha(A, B)$ are all nonlinear expressions. All variables required in the calculation of the right-hand side of equations (24) to (29) are first synthesized onto the real-space transform grid. From the real-space form of the variables, these nonlinear expressions can be calculated. All of the integrals on the right-hand side of equations (24) to (29) can be calculated by one of the three forms,

$$\{B_T\}_l^m = \frac{1}{2\pi}\int_{-\pi/2}^{\pi/2}\int_0^{2\pi} B_T Y_l^{cm} \cos\theta d\lambda d\theta \tag{30}$$

$$\{a^2\nabla^2 E\}_l^m = -\frac{l(l+1)}{2\pi}\int_{-\pi/2}^{\pi/2}\int_0^{2\pi} \frac{U^2+V^2}{2\cos\theta} Y_l^{cm} d\lambda d\theta \tag{31}$$

$$\begin{aligned}\{\alpha(A, B)\}_l^m &= \tfrac{1}{2\pi}\textstyle\int_{-\pi/2}^{\pi/2}\int_0^{2\pi} \tfrac{1}{\cos\theta}[\tfrac{\partial A}{\partial\lambda} + \cos\theta\dfrac{\partial B}{\partial\theta}] Y_l^{cm} d\lambda d\theta \\ &= \textstyle\int_{-\pi/2}^{\pi/2}[imA_m P_l^m - B_m \cos\theta\dfrac{\partial P_l^m}{\partial\theta}]\dfrac{d\theta}{\cos\theta}\end{aligned} \tag{32}$$

With the aid of the transform technique, the nonlinear terms can be calculated efficiently by computing the products in real grid space and transforming them back to the spectral coefficient space.

The fully explicit time integration algorithm is employed in vorticity and moisture equations while the semi-implicit formulation is used in the remaining equations. After eliminating $\overline{D_l^m}^t$ and $\overline{W_l^m}^t$, we get a diagnostic equation for $\overline{P_l^m}^t$

$$\frac{\partial}{\partial\sigma}(\frac{\sigma}{\gamma^*}\frac{\overline{\partial P_l^m}^t}{\partial\sigma}) - \frac{R\Delta t^2}{a^2} l(l+1)\overline{P_l^m}^t = \frac{\partial}{\partial\sigma}\{\frac{C_T}{\gamma^*}\}_l^m + \{C_D\}_l^m \tag{33}$$

where $(^{-t})$ is defined by $\overline{F}^t = \frac{1}{2}[F(t+\Delta t) + F(t-\Delta t)]$, and

$$\{C_T\}_l^m = R\Delta t\{\alpha(UT', VT') - B_T\}_l^m + \sigma\frac{\partial}{\partial\sigma}P_l^m(t-\Delta t) \tag{34}$$

$$\{C_D\}_l^m = -R\Delta t\{B_w - \Delta t\alpha(B, -A) + a^2\Delta t\nabla^2 E\}_l^m + R\Delta t D_l^m(t-\Delta t) \tag{35}$$

Thus if $\{C_D\}_l^m$ and $\{C_T\}_l^m$ are known, the $\overline{P_l^m}^t$ can be calculated for each l and m. Then the remaining variables $\overline{D_l^m}^t$, $\overline{W_l^m}^t$, $\overline{W_{sl}^m}^t$, $\overline{q_l^m}^t$ and consequently the same variables at $t + \Delta t$ can be obtained.

2.3 Vertical discretization

Suppose there are N vertical levels in the FSU GSM, they are defined as $\sigma = \sigma_n = (2n-1)/2N$, $1 \le n \le N$. $\sigma_{N+1} = 1$. The variables U, V, ζ, D, ϕ and P are carried on these levels. The vorticity equation and divergence equation are applied at these levels. q, ϕ_s and W_s are carried on σ_{N+1} level.

The variables $\tilde{T}$,$\tilde{S}$, $\tilde{\dot{\sigma}}$ and $\tilde{W}$ are defined on levels intermediate to the levels of the geopotential, which are defined as $\tilde{\sigma}_n = \sqrt{\sigma_n \sigma_{n+1}}$. The tilde notation is used to denote the variables which are carried in the layers ($\sigma = \tilde{\sigma}_n$).

It is convenient to define $d_n = ln(\sigma_{n+1}/\sigma_n)$ and vertical increments $\tilde{\delta}_n = \tilde{\sigma}_{n+1} - \tilde{\sigma}_n$, $1 \leq n \leq N-2$. For the top and bottom increments $\tilde{\delta}_0 = \tilde{\sigma}_1$ and $\tilde{\delta}_{N-1} = 1 - \tilde{\sigma}_{N-1}$.

In the program, we use the spectrum coefficients $\{\xi, D, P, q, S\}_l^m$ of the grid variables as the basic variables.

2.4 New features related to the MPI version of the FSU spectral model

With the development of massively parallel processing computer technology, today's mainframe supercomputers and workstations tend to have multiprocessors which allow a single job to be split into pieces which execute on different processors concurrently. Such a technique is called parallel processing or multi-tasking. Its main aim is to make the fullest use of the computer resources and reduce run time of the job. In the present day data assimilation and forecast operations, which is tapping the capacity of the most powerful computers available, multi-tasking of the models becomes an inevitable trend.

There are different kinds of algorithms implementing parallelization, depending on the architecture of the hardware platforms and software systems provided. some algorithms are multi-tasked by inserting compiling directives into the code to make the compiler produce the executable file which can run on multi-processors in parallel. This kind of multi-tasking involves less programmer effort to rewrite the code than some other algorithms such as MPI (Message-Passing Interface). In accordance with the original model, the TLM and ADJ were also multi-tasked with the same algorithm.

2.5 General concepts and algorithmic aspects of multi-tasking

The processes that participate in the parallel execution of a task are arranged in a master/slave organization. The original process is the master. It creates several (can be zero) slave processes at the beginning of execution of the job according to the user's specification and the resources available. The master and each of the slave processes are called a thread of execution or simply a thread.

The technique of multi-tasking we employed works at the FORTRAN do loop level. Whenever a parallel do loop is encountered, the master asks the slaves for help. The iterations of the do loop is automatically divided among them and each slave executes different indexes of the do loop. For each multitasked do loop, the processors work independently and concurrently and the programmer can not control the sequence of the execution of the iterations. Therefore, the basic principle is that for multi-processing to work properly, the iterations of the loop must not depend on each other; each iteration must stand alone and produce the same answer regardless of whether any other iteration of the loop is executed. This property is called data independence. Otherwise, the loop is called to have data dependency and can not be correctly executed in parallel without modification.

For a loop to be data-independent, no iterations of the loop can write a value into a memory location that is read or written by any other iteration of that loop. It is all right if the same iteration reads and/or writes a memory location repeatedly as long as no others do; it is also all right if many iterations read the same location, as long as none of them write to it. In a FORTRAN program, memory locations are represented by variable names. So we need to examine the way variables are used in the loop, paying particular attention to variables that appear on the left-hand side of assignment statements. If a variable (including elements of an array) is both read and written within a loop, there is possibility of data dependency associated with it.

Our main task in parallelizing the model is to analyze the data dependencies of each do loop and if a do loop is data dependent, to locate the statement(s) in it which can not be made parallel and try to find another way to express so that it doesn't depend on any other iteration of the loop. If this fails, we have to try to separate this statement from the remainder of the original loop.

2.6 Implementation of Parallelization process with the FSU spectral model

In this section we begin by first discussing first the organization of the FSU global spectral model. Initially, the spectral prognostic variables and their derivatives are transformed to the Gaussian grid, one latitude at a time. One routine computes the nonlinear dynamical tendencies for each grid point, while another routine computes the physical tendencies. Each of these routines are designed to operate on a single latitude at a time. The main physics routine calls a number of other routines which may act either on a single vertical column, or on a latitude band. When the tendencies have been computed, they are spectrally analyzed, again one latitude band at a time. These tendencies are accumulated from one band to the next. The spectral and grid point calculations are thus done in one large loop across the latitudes. Once the spectral tendencies have been summed up, they are then used by the semi-implicit algorithm to obtain the prognostic spectral variables at the next time step. The process is then repeated until the forecast is complete.

The latitude band structure has a number of advantages: only one latitude band of the grid point variables for the current time step is required in memory at any given time. This significantly reduces the amount of memory required by the model. Another advantage is the ease of carrying out the spectral transformation. This transformation is done in two steps: first, the spectral variables are Legendre-transformed to the latitude band, and then a Fourier transform is applied. In order to use the Fast Fourier Transform, the whole latitude is done at once.

Since the Legendre transform is rather slow, it can easily be done one latitude at a time. Given the band structure of the model, and that the calculations for each band are independent of each other, it is straightforward to parallelize the model by dividing the latitude bands across processors. Once the calculations are carried out for all processors, a reduction is done to obtain the total spectral global tendencies. The semi-implicit scheme is fast, so there is no need to implement it in parallel.

While this one dimensional partitioning is not the most efficient parallel algorithm, it is easy to implement, and can be carried out on both shared memory (SMP) architecture and distributed memory architecture (DMP) in a very similar manner. For SMP parallelization, the main latitude loop is split amongst processors by using compiler directives for SMP-enabled FORTRAN compilers. The model was first parallelized on an SGI Origin 2000 using the "doacross" directive. The directives have now been replaced with the OpenMP standard "parallel do" directives.

After the main latitude loop, a reduction is done by the master task, and the semi-implicit scheme is solved on a single processor. To achieve optimal load balance, any latitude band can be assigned to any processor. The main latitude loop is consecutively partitioned across logical latitudes, each of which may be mapped to any physical latitude by a mapping function. Although a number of scheduling algorithms are available, we use primarily static scheduling.

Parallelization on DMP architecture is done by running a copy of the model on each processor and using Message Passing Interface (MPI) for communication. Each processor only operates on its portion of the latitude bands, which is determined at the beginning of the program execution. When all processors have completed their latitude calculations, a reduction is done using the MPI allreduce routine. Thus all processors have a copy of the total spectral tendencies, and each solves the semi- implicit scheme to obtain the global spectral variables at the next time step. While this computation is redundant, it is fast and minimizes communication.

For the perturbations we use a Double Fast Fourier Transform for the north-south and east-west directions. This transform can be done in parallel. For SMP machines, such as the SGI Origin 2000, there are parallel 2D FFT libraries available. For DMP architecture, the Double FFT Transpose algorithm can be used (*cf.* Foster,1994).

All Spectral Transforms, Grid Dynamics and Physics are computed in one large latitude loop. A simple one dimension partitioning of the latitudes is done. Each partition is distributed across the nodes. The partitions are further repartitioned across CPUs within the node. Each sub-partition may either be an MPI task or OpenMP thread. MPI is used to communicate spectral tendencies (reduction) across the nodes. For OpenMP, a simple summation loop is used for reduction. Serial "overhead" includes solving the semi-implicit algorithm and array initialization. Load balancing s achieved by distributing the latitudes evenly geographically across each sub-partition. In other words, each partition

(CPU) has latitudes that are evenly distributed from North to South Pole.

The following is an example for illustrating the MPI process:

```
#if defined(MPI)
      include 'mpif.h'
#endif

#if defined(MPI)

c
c  initial MPI environment
c
      call MPI_init(ierr)
c
c   obtain the I.D. of current processor
c
      call MPI_comm_rank( MPI_COMM_WORLD, myid, ierr)
c
c   obtain the total numbers of processors which involved in the calculation
c
      call MPI_comm_size( MPI_COMM_WORLD, numprocs, ierr)
#else

      myid=0
      numprocs=1

#endif

c   read initial data for P,C,T,ES,PS
c
       call read_his(nhis_in,ifmt,P,C,T,ES,PS,len1,ilev,levs,ieof)
c
c  if using multi CPU, the inputed initial data should be broadcasted to
c  each processors
c
#if defined(MPI)
       if (numprocs .gt. 1) then
        call MPI_bcast(P,icoef2*ilevv,MPI_REAL8,0,MPI_COMM_WORLD,ierr)
        call MPI_bcast(C,icoef2*ilevv,MPI_REAL8,0,MPI_COMM_WORLD,ierr)
        call MPI_bcast(T,icoef2*ilevv,MPI_REAL8,0,MPI_COMM_WORLD,ierr)
        call MPI_bcast(ES,icoef2*levs,MPI_REAL8,0,MPI_COMM_WORLD,ierr)
        call MPI_bcast(PS,icoef2,MPI_REAL8,0,MPI_COMM_WORLD,ierr)
       endif
#endif

       ... ...

c
c  divide the latitude to several bands according to the number of
c  processors. on each processor, only one latitude band was distributed
c
       latproc=float(ilatv)/float(numprocs) + 0.5

       latstart(1)=1
       latend(1)=latproc

       do i=2,numprocs
        latstart(i)=latstart(i-1)+latproc
        latend(i)=latend(i-1)+latproc
       enddo
       latend(numprocs)=ilatv
c
c  calculate the tendency pt,ct,peet,est,pst at each latitude
c
       DO 750 jj=latstart(myid+1),latend(myid+1)

         ... ...

 750  continue
 c
 c   after carrying out the calculation of the tendency over each
 c   processor, a reduction should be carried out to redistribute the
c    updated tendency to all latitudes.

      If (numprocs .gt. 1) then

      call MPI_allreduce(ct,summpi,2*icofp,MPI_REAL8,
     + MPI_SUM, MPI_COMM_WORLD,ierr)

      do i=1,icofp
```

```
        ct(i,1)=summpi(i)
      enddo

      call MPI_allreduce(peet,summpi,2*icofp,MPI_REAL8,
     + MPI_SUM, MPI_COMM_WORLD,ierr)

      do i=1,icofp
        peet(i,1)=summpi(i)
      enddo

     call MPI_allreduce(est,summpi,2*icofs,MPI_REAL8,
    + MPI_SUM, MPI_COMM_WORLD,ierr)

     do i=1,icofs
       est(i,1)=summpi(i)
     enddo

     call MPI_allreduce(pst,summpi,2*icoef,MPI_REAL8,
    + MPI_SUM, MPI_COMM_WORLD,ierr)

     do i=1,icoef
       pst(i,1)=summpi(i)
     enddo

     Endif

     ...

#if defined(MPI)
     call MPI_finalize(rc)
#endif
     stop
```

Fig. 2.6 is the flowchart of dynamic core of FSU Spectral model

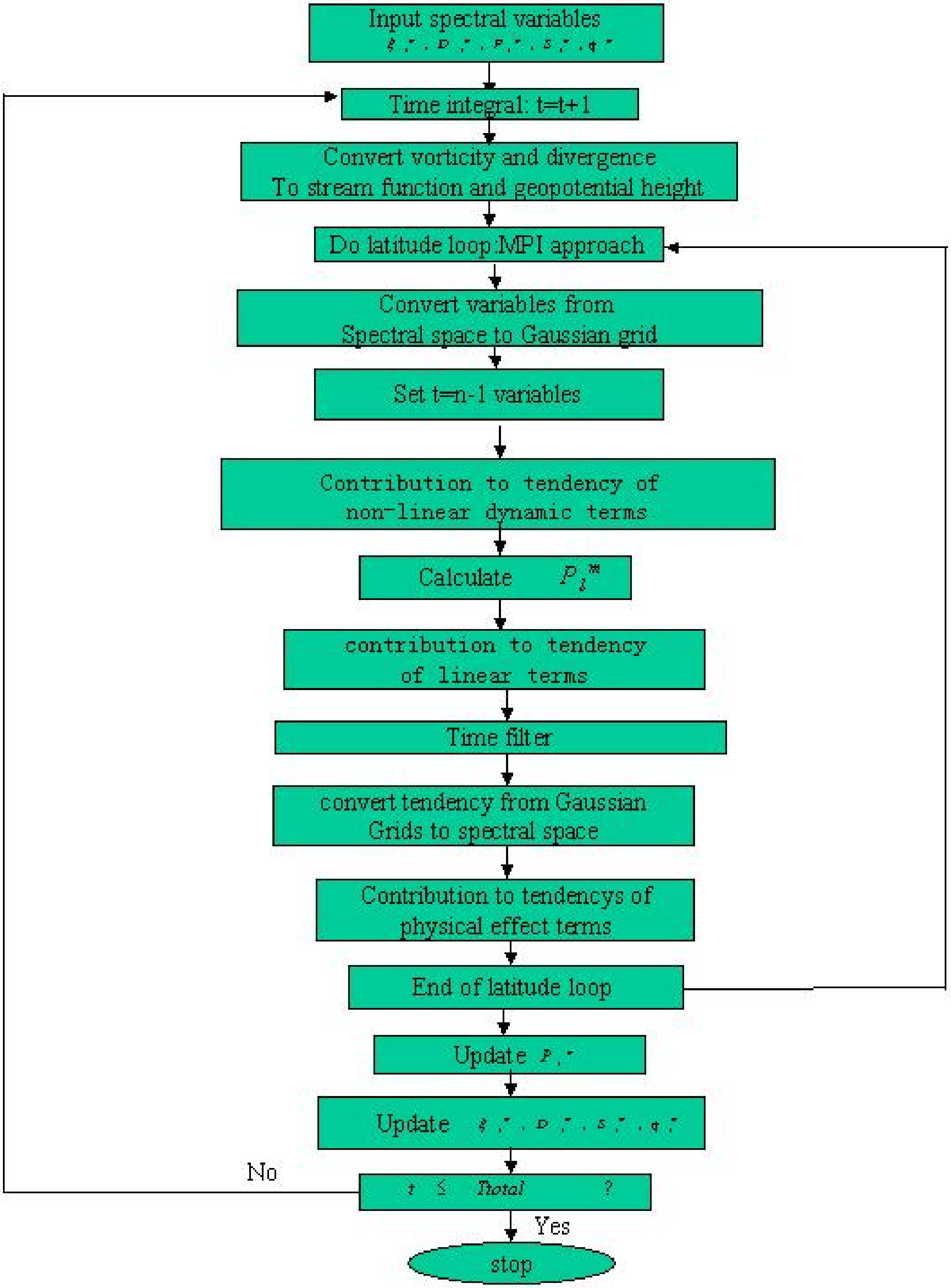
Input spectral variables
$\xi_l^m, D_l^m, P_l^m, S_l^m, q_l^m$
Time integral: t=t+1
Convert vorticity and divergence
To stream function and geopotential height
Do latitude loop:MPI approach
Convert variables from
Spectral space to Gaussian grid
Set t=n-1 variables
Contribution to tendency of
non-linear dynamic terms
Calculate P_l^m
contribution to tendency
of linear terms
Time filter
convert tendency from Gaussian
Grids to spectral space
Contribution to tendencys of
physical effect terms
End of latitude loop
Update P_l^m
Update $\xi_l^m, D_l^m, S_l^m, q_l^m$
No
$t \leq Ttotal$?
Yes
stop

Figure 1: Flowchart of dynamic core of FSU Spectral model

3 Tangent Linear Model of FSU GSM

The TLM is the linear approximation to the original nonlinear model. Its accuracy determines the accuracy of the adjoint model and the accuracy of the gradient of cost function calculated from the adjoint model in 4-D variational assimilation. Therefore we need to carefully evaluate its accuracy and its validity range.

3.1 Verification of Tangent Linear Model

To verify the correctness of the TLM, we first checked each subroutine by comparing the result of the TLM with the difference of the twice GCM call, with and without perturbations, respectively. After that,in order to check the whole GCM, we employed a more quantitative algorithm, as described in Yang et al. (1996).

The evolution of the vector of the atmospheric variables $\mathbf{X}$ is given by the integration of the model M between times t_0 and t_n as:

$$\mathbf{X}(t_n) = M\left(t_n, t_0\right)\left(\mathbf{X}\left(t_0\right)\right). \tag{36}$$

If at initial time $\mathbf{X}(t_0)$ has a perturbation $\delta\mathbf{X}(t_0)$, it would evolve to:

$$\mathbf{X}^{'}(t_n) = M\left(t_n, t_0\right)\left(\mathbf{X}_0\left(t_0\right) + \delta\mathbf{X}\left(t_0\right)\right) \tag{37}$$

The perturbation

$$\begin{aligned} \Delta\mathbf{X} &= \mathbf{X}^{'}(t_n) - \mathbf{X}(t_n) \\ &= M\left(t_n, t_0\right)\left(\mathbf{X}_0\left(t_0\right) + \delta\mathbf{X}\left(t_0\right)\right) - M\left(t_n, t_0\right)\left(\mathbf{X}\left(t_0\right)\right) \\ &= M\left(t_n, t_0\right)\left(\mathbf{X}_0\left(t_0\right)\right) + R\left(t_n, t_0\right)\delta\mathbf{X}(t_0) + O(\delta\mathbf{X}(t_0)^2) - M\left(t_n, t_0\right)\left(\mathbf{X}_0\left(t_0\right)\right), \end{aligned}$$

where R is the first derivative of M, or the tangent-linear model.

In the first order, the perturbation of $\mathbf{X}$ can be approximated by:

$$\delta\mathbf{X}\left(t_n\right) = R\left(t_n, t_0\right)\delta\mathbf{X}\left(t_0\right) \tag{38}$$

The difference is denoted as:

$$D(\delta\mathbf{X}) = \Delta\mathbf{X} - \delta\mathbf{X}. \tag{39}$$

To quantify the comparison, a norm is defined by:

$$\|\mathbf{X}\|^2 = \mathbf{X}^T W \mathbf{X} \tag{40}$$

where W is a diagonal matrix which allocates a proper weighting to each field of variable.

The relative difference between the TLM and the nonlinear model is defined as

$$r = \frac{\|D\|}{\|\delta\mathbf{X}\|}. \tag{41}$$

It provides a quantitative measure of the accuracy of the linear approximation.

The basic trajectory used to carry out the verification is the model integration using the initial condition from December 11, 1997 at 01GMT. To produce the perturbations, we simply used a random number generator to obtain a set of random numbers, which multiplied the initial data to form a set of perturbations, These perturbations multiplied by different factors α serve as the perturbations for a series of our experiments.

For each of the experiments, we calculated the correlation coefficients as well as the relative differences r between D and $\delta\mathbf{X}$ for each model variable. Table 1 gives the correlation coefficients between D field and $\delta\mathbf{X}$ field while Table 2 shows the relative errors. The length of the integration period was 6 hours.

Table 1: Correlation Coefficients Between D Field and $\delta\mathbf{X}$ Field(6 hours window):

α	ξ_l^m	D_l^m	P_l^m	S_l^m	q_l^m
10^{-1}	0.9752774696863	0.9586570786112	0.9995981826444	0.9902851180289	0.9995611838382
10^{-2}	0.9998145881311	0.9997563770319	0.9999980279475	0.9999836182357	0.9999969428590
10^{-3}	0.9999984587540	0.9999979156388	0.9999999623921	0.9999996348523	0.9999999624367
10^{-4}	0.9999999825436	0.9999999653958	0.9999999995609	0.9999999646956	0.9999999992486
10^{-5}	0.9999999997527	0.9999999996616	0.9999999999975	0.9999999968876	0.9999999999957
10^{-6}	0.9999999999980	0.9999999999970	0.9999999999999	0.9999999995488	0.9999999999999
10^{-7}	0.9999999999923	0.9999999999907	0.9999999999984	0.9999999999890	0.9999999999999
10^{-8}	0.9999999995993	0.9999999995447	0.9999999994902	0.9999999999393	0.9999999999924
10^{-9}	0.9999999374050	0.9999999225539	0.9999999619972	0.9999999942594	0.9999999990478
10^{-10}	0.9999948661435	0.9999940157752	0.9999964451702	0.9999996596388	0.9999998907478

Table 2: Relative Error Between D Field and $\delta\mathbf{X}$ Field(6 hours window):

α	ξ_l^m	D_l^m	P_l^m	S_l^m	q_l^m	Total
10^{-1}	0.233984	0.307175	0.283554E-01	0.141929	0.296302E-01	0.283554E-01
10^{-2}	0.197709E-01	0.221351E-01	0.198696E-02	0.573723E-02	0.247941E-02	0.198696E-02
10^{-3}	0.175702E-02	0.204176E-02	0.274501E-03	0.855306E-03	0.274481E-03	0.274501E-03
10^{-4}	0.186891E-03	0.264308E-03	0.297036E-04	0.265817E-03	0.387869E-04	0.297036E-04
10^{-5}	0.228556E-04	0.261336E-04	0.222050E-05	0.788966E-04	0.291837E-05	0.222052E-05
10^{-6}	0.201098E-05	0.250696E-05	0.344050E-06	0.300378E-04	0.207467E-06	0.344071E-06
10^{-7}	0.390822E-05	0.430068E-05	0.177315E-05	0.467600E-05	0.376353E-06	0.177315E-05
10^{-8}	0.283220E-04	0.301847E-04	0.319380E-04	0.110153E-04	0.388815E-05	0.319380E-04
10^{-9}	0.353907E-03	0.394679E-03	0.276283E-03	0.108248E-03	0.438597E-04	0.276283E-03
10^{-10}	0.322239E-02	0.346072E-02	0.267074E-02	0.825069E-03	0.467469E-03	0.267074E-02

From the above tables we see that as α decreases from $10.^{-1}$ to 10^{-10}, the correlation coefficients approach 1.0 with a very high level of accuracy. The relative error decreases almost linearly between $\alpha = 10^{-1}$ and $\alpha = 10^{-6}$.

The relative errors between D and $\delta\mathbf{X}$ are also shown in Figure. 2-4 for integration windows of one, three and six hours respectively. For all the windows the minimum of the relative errors for $10^{-1} \leq \alpha \leq 10^{-10}$ was attained when $\alpha = 10^{-6}$.

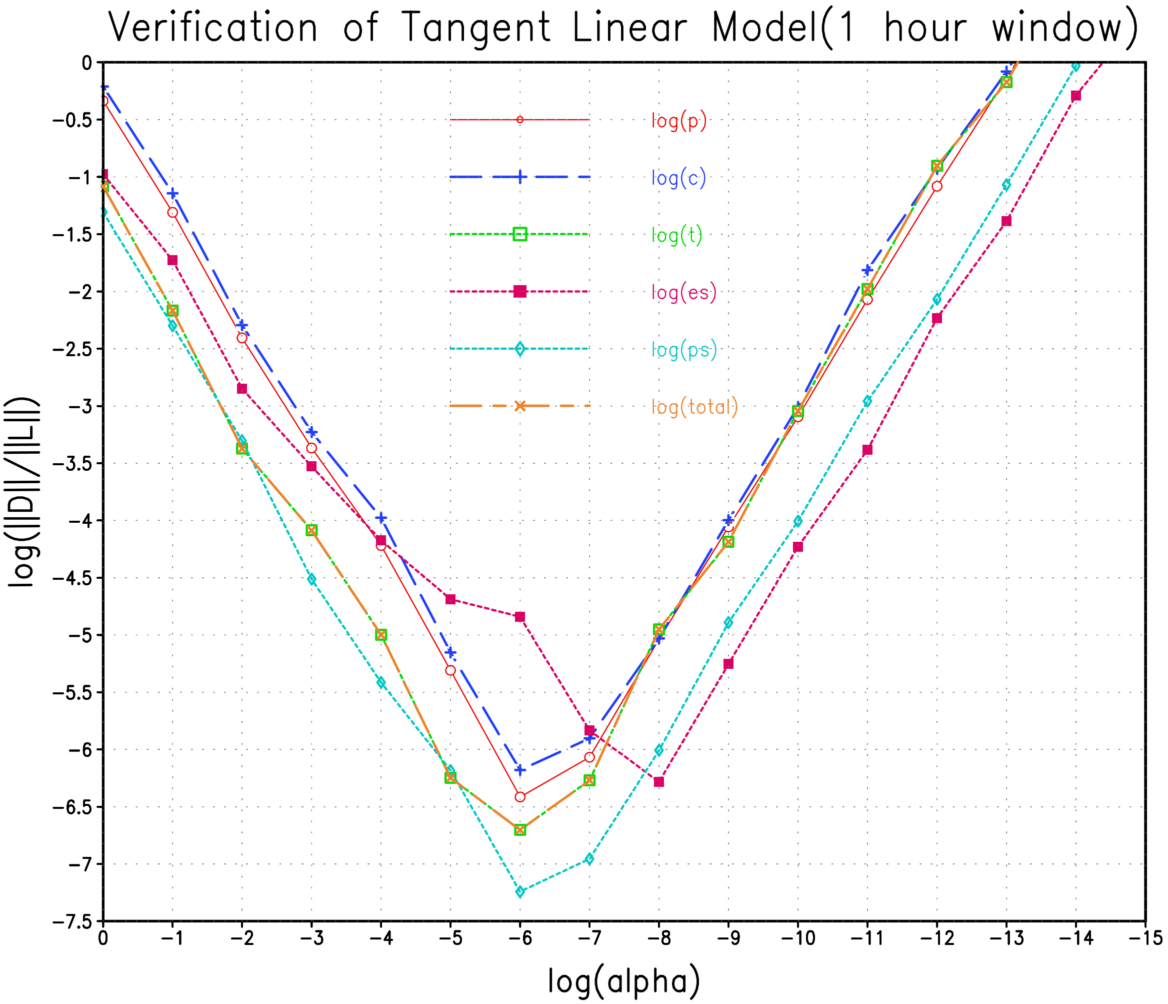

Figure 2: Verification of TLM for one hours data assimilation window.

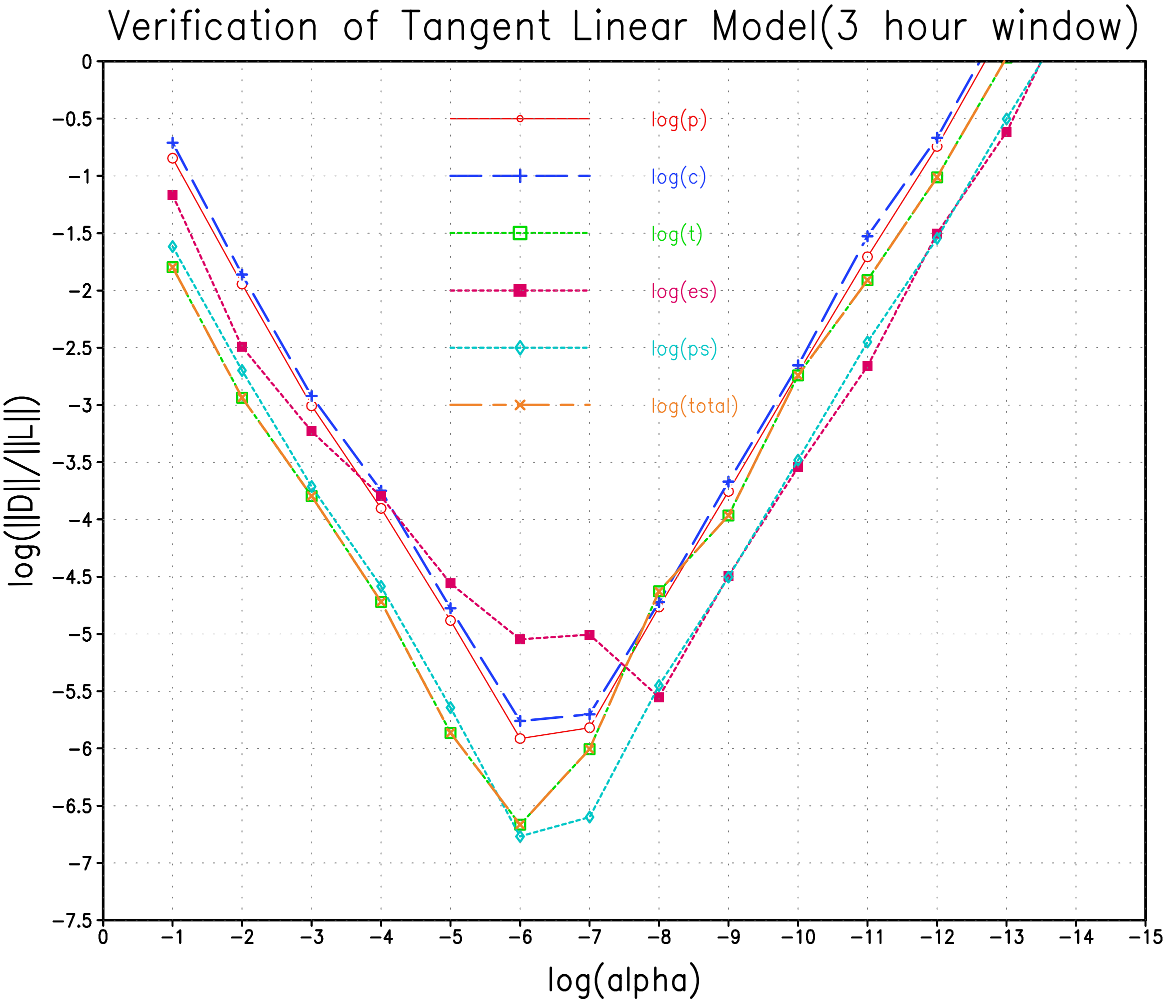

Figure 3: Verification of TLM for 3 hours data assimilation window.

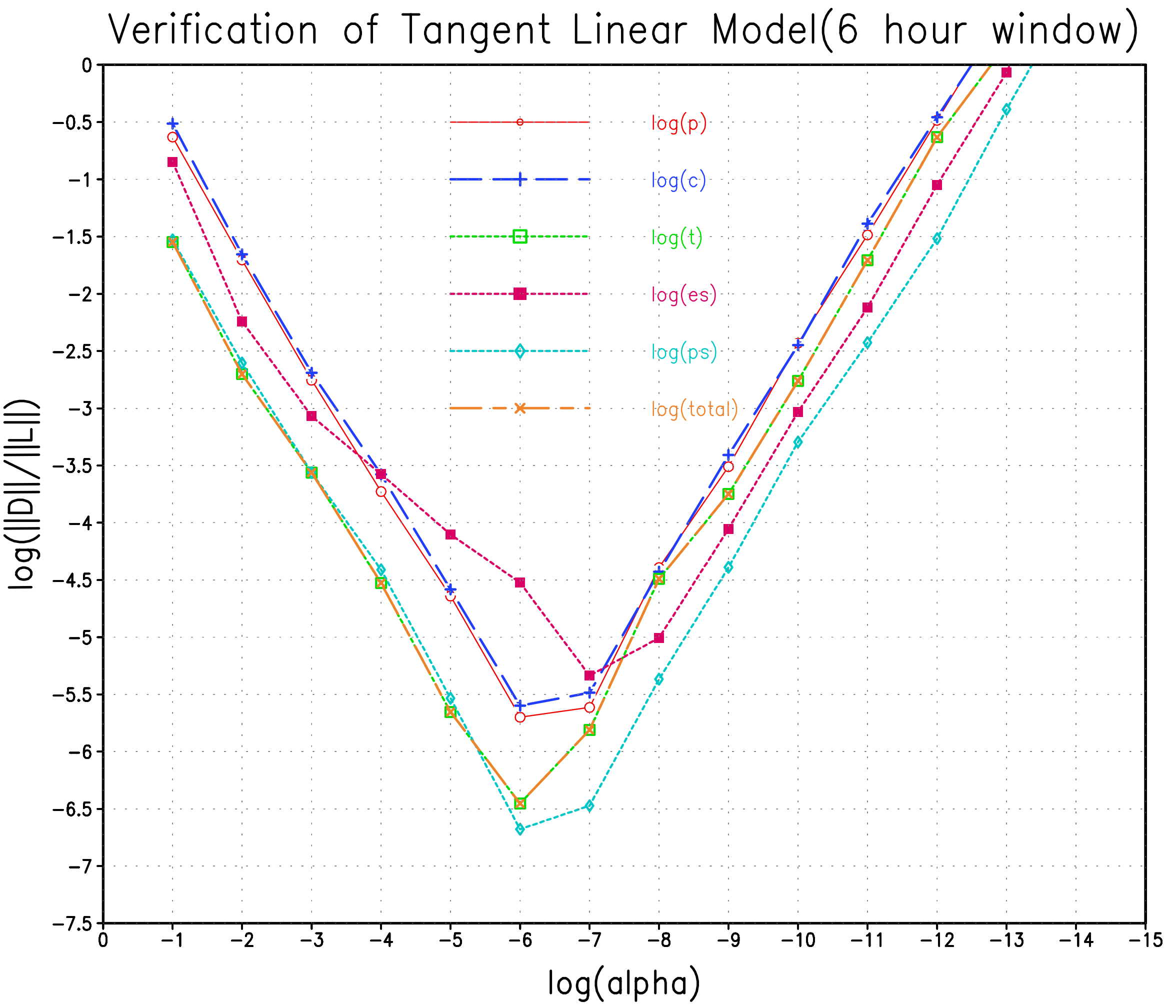

Figure 4: Verification of TLM for 6 hours data assimilation window.

4 Adjoint Model of FSU GSM

4.1 Variational data assimilation problem

The cost function for 4-D variational data assimilation will assume the form :

$$J(\mathbf{X}_0) = \frac{1}{2}(\mathbf{X_0} - \mathbf{X_b})^T\mathbf{B}^{-1}(\mathbf{X_0} - \mathbf{X_b}) + \frac{1}{2}\sum_{r=0}^{R}(\mathbf{HX}(t_r) - \mathbf{Z}^{obs}(t_r))^T\mathbf{W}(\mathbf{HX}(t_r) - \mathbf{Z}^{obs}(t_r))$$

where $\mathbf{X}_0$ is the spectral coefficients of control variable, a vector of dimension N representing the initial state of the model; $\mathbf{X}(t_r)$ is a vector of dimension N containing all the spectral coefficients of model variables; $\mathbf{Z}^{obs}(t_r)$ represents observational data in spectral space used for the assimilation purposes; $\mathbf{H}$ is a transformation matrix that maps the model variables to the observations; $\mathbf{W}$ is an $N \times N$ diagonal matrix of weighting coefficients which represents the covariance matrix of the observation errors. The first term on the RHS of the equation is the background term, $\mathbf{X_0} - \mathbf{X_b}$ represents the departures of the model variables at the start of the analysis from the spectral coefficients of background field $\mathbf{X_b}$. $\mathbf{B}$ is an approximation to the covariance matrix of background error. Here we simply omit the background error terms. The values of the elements are usually determined by the dimensional scaling of various variables, relative importance and quality of the data set and other considerations. Generally, $\mathbf{W}(t_r)$ can be taken as the inverse covariance matrix of the observation errors.

Almost all of the large-scale unconstrained minimization algorithms require the information of the gradient of the cost function $J(\mathbf{X_0})$ with respect to the control variables. The most efficient way to obtain the gradient is to deploy adjoint method.

For the purpose of the following discussion, the governing equation of a numerical model will be written as

$$\frac{\partial \mathbf{X}}{\partial t} = \mathbf{F}(\mathbf{X}, t) \tag{42}$$

$$\mathbf{X}(0) = \mathbf{X}_0 \tag{43}$$

Where $\mathbf{X}$ represents the set of spectral coefficients of the model variables. $\mathbf{F}$ is the nonlinear operator matrix. The values of $\mathbf{X}(t)$ can be obtained by integrating the model from the initial condition $\mathbf{X}_0$.

The tangent linear equation may be solved as the solution of the following system

$$\frac{\partial \hat{\mathbf{X}}}{\partial t} = [\frac{\partial \mathbf{F}}{\partial \mathbf{X}}](\mathbf{X}, t)\ \hat{\mathbf{X}} \tag{44}$$

$$\hat{\mathbf{X}}(0) = \hat{\mathbf{U}} \tag{2.39}$$

where the notation $(\hat{\ })$ indicates a perturbation vector, $\mathbf{X}$ is the model state vector variable while $\hat{\mathbf{X}}$ is the perturbation vector variable, and $\hat{\mathbf{U}}$ is a small perturbation of initial condition.

The adjoint model is defined as:

$$-\frac{\partial \mathbf{P}}{\partial t} - [\frac{\partial \mathbf{F}}{\partial \mathbf{X}}]^T \mathbf{P} = [\nabla_{\mathbf{X}} \mathbf{H}]^T \mathbf{W}(\mathbf{H}(\mathbf{X})(t_r) - \mathbf{Z}^{obs}(t_r)) \tag{45}$$

$$\mathbf{P}(t_R) = 0 \tag{46}$$

where P is the adjoint variables, the superscript T denotes the transpose operator.

Therefore, in order to construct the adjoint code, the nonlinear terms in the forward nonlinear model need first to be linearized locally. Then the adjoint code is developed directly from the basic direct code, the tangent linear code. If we view the linear model as the result of the multiplication of a number of operator matrices:

$$P_r = A_1 A_2 \dots A_N \tag{47}$$

where each A_i represents a subroutine or Do loop. Then the adjoint model can be viewed as

$$P_r^T = A_N^T \dots A_2^T A_1^T \tag{48}$$

i.e., the adjoint model is written backward from the tangent linear model line by line, subroutine be subroutine (Navon et al. 1992).

Since a minor error may result in the incorrect gradient of cost function with respect to the control variables, it is necessary to verify the correctness of the tangent linear code and the adjoint code segment by segment. The final gradient check should also be performed. For a detailed description of the verification of correctness, see Navon *et al.* (1992) ,Zou et al (1993c), Li *et al.*(1993,1994).

4.2 Verification of the adjoint model

We tested the correctness of the adjoint model by applying the following identity (see Navon et al., 1992):

$$(AQ)^T (AQ) = Q^T (A^T (AQ)) \tag{49}$$

where Q represents the input of the original code, A represents the original code or a segment of it, say a subroutine, a do loop or even a single statement. A^T is the adjoint of A. If (49) holds within the machine accuracy, it can be said that the adjoint is correct versus the TLM code.

One has to be cautious when using (49) to check the adjoint code. Sometimes different variables are involved in Q and the identity represents an integral of all the spectral coefficients of the model variables in spectral space. If spectral coefficients of some variables are very small in magnitude as compared to other ones, their error may not show up in the integral. In this case we need to change its magnitude in the ADJ code (also change the TLM accordingly) in order to make sure that each and every variable is checked.

We checked the adjoint model segment by segment, do loop by do loop and subroutine by subroutine. With double precision, the identity (49) was always accurate within 13 digits or better. This verified the correctness of the adjoint model against TLM.

4.3 Verification of Gradient

In the 4-D variational data assimilation, the adjoint model is used to calculate the gradient of a cost function J with respect to the initial disturbance $\delta\mathbf{X}(t_0)$. J can be defined as:

$$J(\mathbf{X}(t_0)) = \frac{1}{2}\sum_{r=0}^{R}\left(\mathbf{X}(t_r) - \mathbf{X}^{obs}(t_r)\right)^T \mathbf{W}(t_r)\left(\mathbf{X}(t_r) - \mathbf{X}^{obs}(t_r)\right) \tag{50}$$

where $\mathbf{X}(t_r)$ is the spectral coefficients of model state at time t_r, $\mathbf{X}^{obs}(t_r)$ is the observation; R is the number of time levels for the analyzed fields in the assimilation window; $\mathbf{W}(t_r)$ is an $N \times N$ diagonal weighting matrix, where $\mathbf{W}_p$, $\mathbf{W}_c$, $\mathbf{W}_T$, $\mathbf{W}_{Ps}$ and $\mathbf{W}_{es}$ are diagonal sub-matrices consisting of weighting factors for each variable, respectively. Their respective values can be calculated, *e.g*, by(see Navon, 1992)

$$\mathbf{W}_T = \frac{1}{\max_{l,m}|\mathbf{X}^{obs}_{T_l^m}(t_0) - \mathbf{X}^{obs}_{T_l^m}(t_r)|^2}$$

The others can be calculated in the same method.

Now suppose the initial $\mathbf{X}(t_0)$ has a perturbation $\alpha\mathbf{h}$, where α is a small scalar and $\mathbf{h}$ is a vector of unit length (such as the normalized vector of the adjoint model output). According to Taylor expansion we get:

$$J(\mathbf{X}(t_0) + \alpha\mathbf{h}) = J(\mathbf{X}(t_0)) + \alpha\mathbf{h}^T\nabla J(\mathbf{X}(t_0)) + O(\alpha^2), \tag{51}$$

We can define a function of α as:

$$\Phi(\alpha) = \frac{J(\mathbf{X}(t_0) + \alpha\mathbf{h}) - J(\mathbf{X}(t_0))}{\alpha\mathbf{h}^T\nabla J(\mathbf{X}(t_0))} = 1 + O(\alpha). \tag{52}$$

For values of α which are small but not too close to the machine zero, one should expect to obtain values of $\Phi(\alpha)$ which are close to unity. Here the gradient $\nabla J(\mathbf{X}(t_0))$ is calculated by the adjoint model.

We first generated the $\mathbf{X}^{obs}(t_r)$ by integrate the original GCM for 6 hours starting from the analysis data at 01GMT on December 11st, 1997. The difference between this result and the initial field, multiplied by 0.1, was then added to the initial field to be used as the initial condition to generate the trajectory $\mathbf{X}(t_r)$. Fig. 5-7 show the variation of $\Phi(\alpha)$ and $log|1 - \Phi(\alpha)|$ with α for different time windows, respectively. From them one can see that as α decreases from 10^{-7} to 10^{-9}, $\Phi(\alpha)$ approaches unity almost linearly and then stays close to unity with a high degree of accuracy until $\alpha \sim 10^{-17}$. This means that for perturbations within this range, the gradient calculated with the adjoint model is reliable.

Please note that the accuracy of the adjoint gradient not only depends on the accuracy of the tangent linear and the adjoint models, but also on the approximation involved in linearizing (50), which in turn depends, to some extent, on the relative magnitude of the distance $\mathbf{X}(t_r) - \mathbf{X}^{obs}(t_r)$ versus the perturbation $\alpha\mathbf{h}$. If we did not multiply the difference by 0.1 when constructing the initial condition for $\mathbf{X}(t_r)$, the result should be even better.

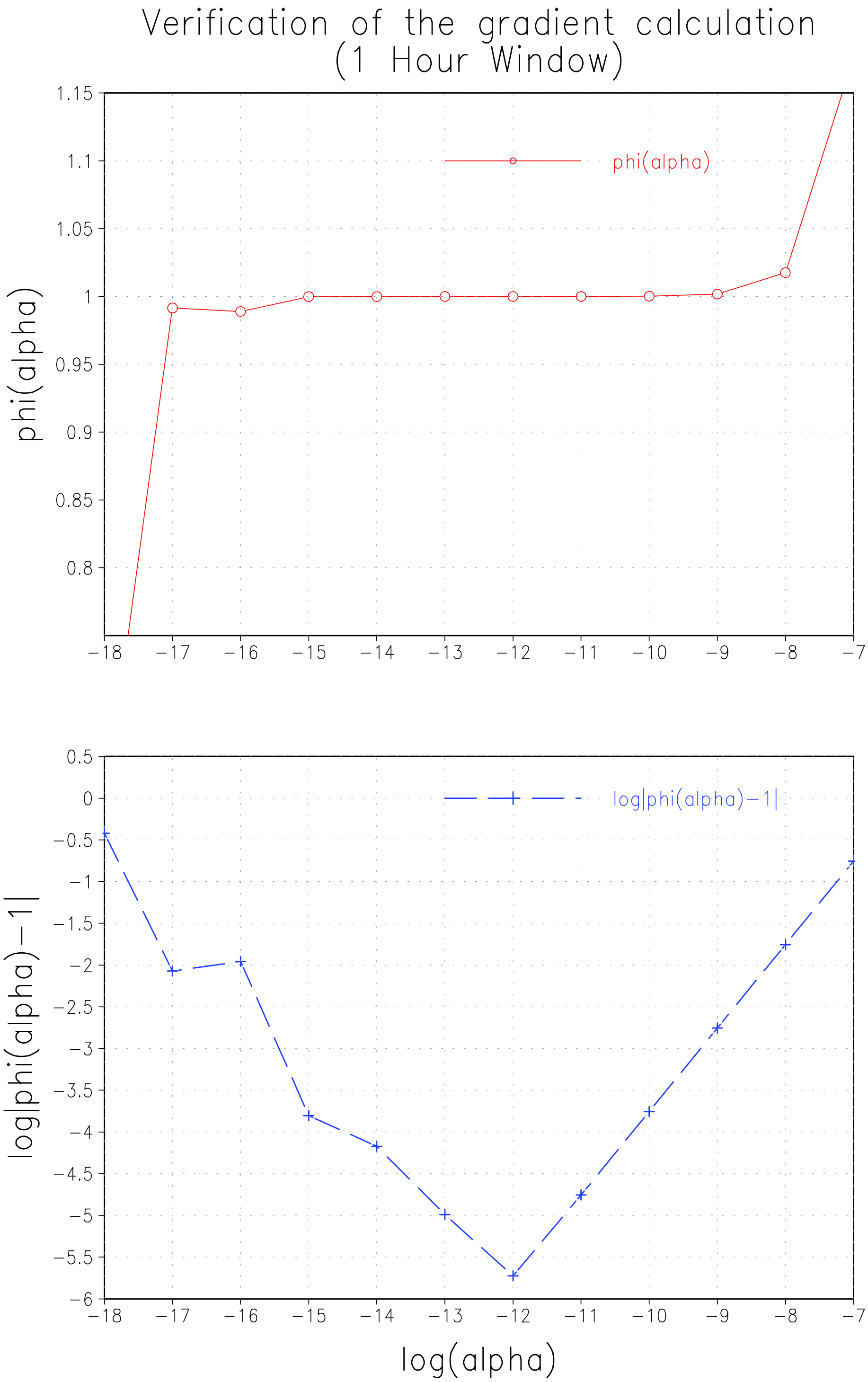

Figure 5: Verification of gradient for one hours data assimilation window. Upper:$\phi(\alpha)$; Lower: $\log|\phi(\alpha) - 1|$.

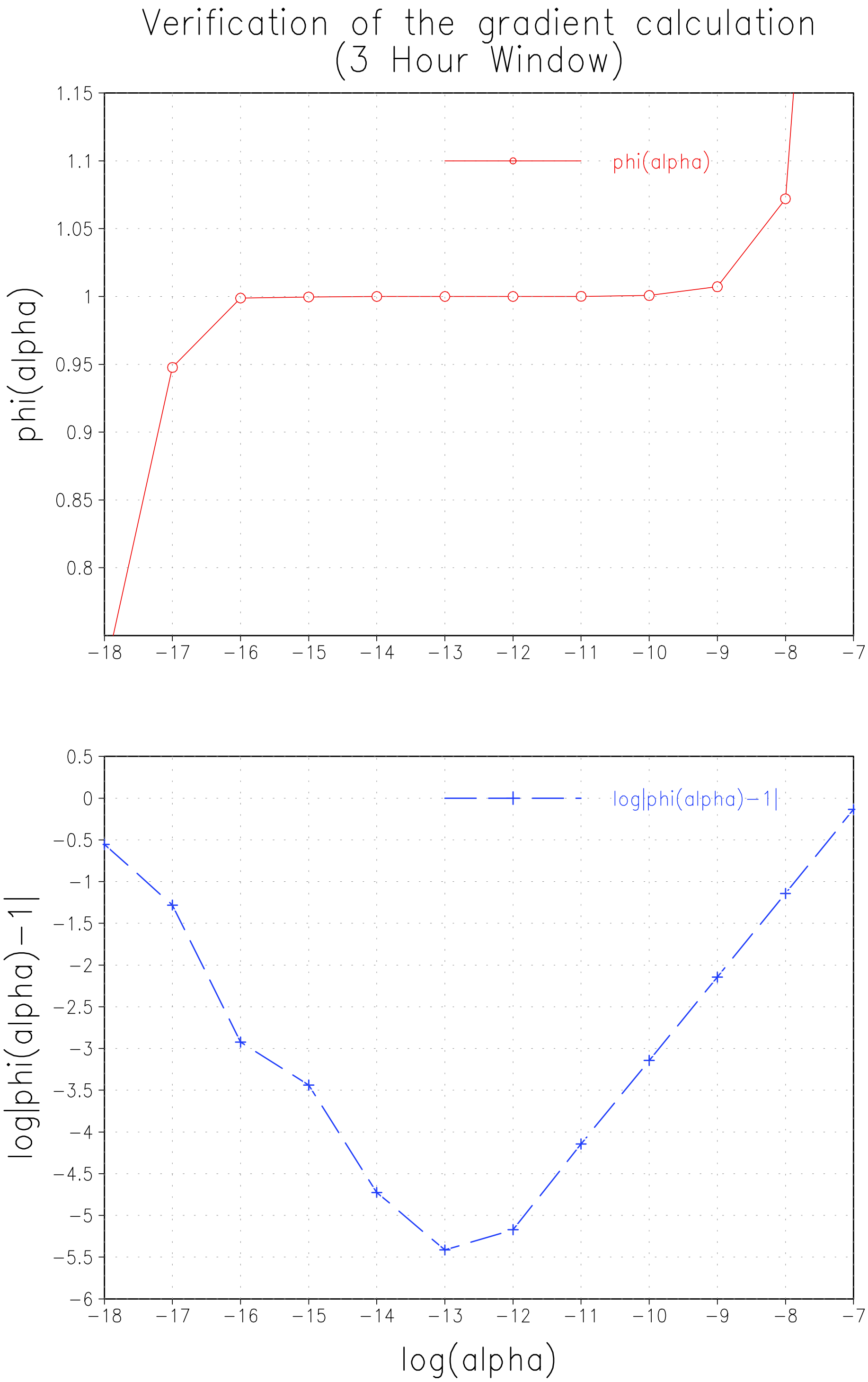

Figure 6: Same as Figure. 5 but for three hours data assimilation window. Upper:$\phi(\alpha)$; Lower: $\log|\phi(\alpha) - 1|$.

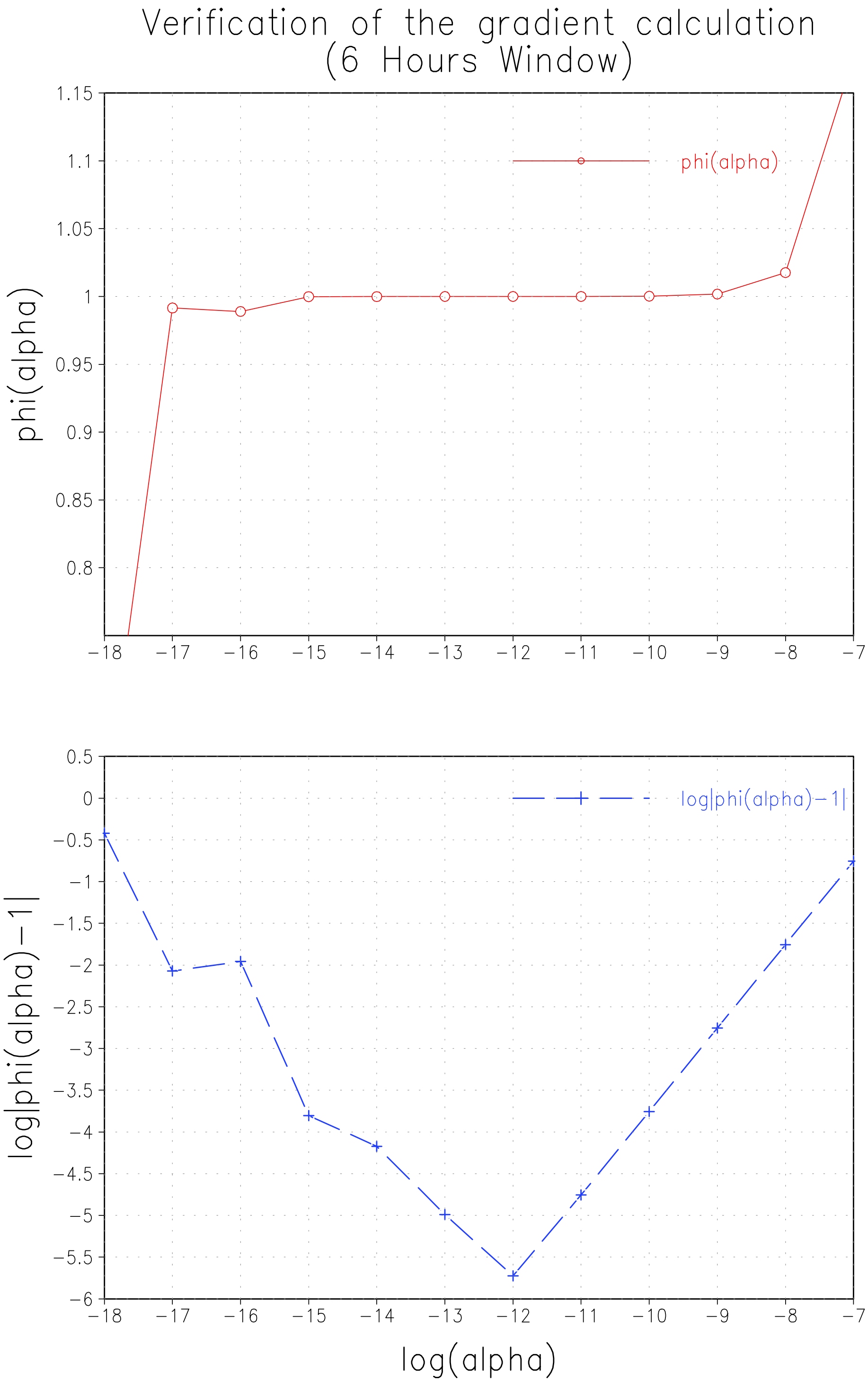

Figure 7: Same as Figure. 5 but for six hours data assimilation window. Upper:$\phi(\alpha)$; Lower: $\log|\phi(\alpha) - 1|$.

4.4 Experiments with minimization of cost functional

Once the gradient of the cost function is available, we can minimize the cost function by carrying out a large-scale minimization process. Here we used the limited-memory large-scale minimization quasi-Newton method LBFGS algorithm which is a robust method(see [18] and [20]).

Fig. 8 shows the variation of the cost function ($\log(||f||/||f_0||)$) and the gradient ($\log(||g||/||g_0||)$) with respect to the number of iterations. It exhibits a very fast rate of convergence.

For the 3 hours window, although the cost function decreased monotonously, the gradient did not decrease during the initial stages(see Fig. 9).

We plotted out the variation of the each component of the cost function, and found out that some components decreased in a very fast manner, while other components experienced a very slow decrease and during initial stages, the former even increased. This led the entire gradient to exhibit an unsatisfactory stages rate of convergence for these stages. This means that the scaling scheme used was not suitable. Initially we adopted the scaling method described by (see Navon, 1992):

$$S_T = \frac{1}{2} \max_{l,m} |\mathbf{X}^{obs}_{T_l^m}(t_0) - \mathbf{X}^{obs}_{T_l^m}(t_r)|$$

A similar procedure was implemented for the other variables.

A change to a more suitable scaling scheme assuming the form :

$$S_{T_l^m} = \frac{1}{2} |\mathbf{X}^{obs}_{T_l^m}(t_0) - \mathbf{X}^{obs}_{T_l^m}(t_r)|$$

was implemented for all prognostic variables in the cost functional. This means that for each spectral coefficient we used a different scaling. For this scaling scheme, all the components of the cost functional exhibited a satisfactory rate of decrease (see Fig. 11). One thing should be pointed out that even for the former scaling scheme the gradient does not decrease monotonically, the cost functional decrease even better than the later scling scheme.

Fig. 12 shows the variation of the cost function and gradient with respect to the number of minimization iterations for the new scaling scheme. Using the new scaling scheme, both cost function and gradient exhibited a monotonic rate of decrease.See also Zupanski (1993,1996).

Fig. 13 shows the variation of cost function and gradient with the number of minimization iterations with the new scaling scheme for a 6h window of assimilation. The results are also very satisfactory.

The individual components of the cost function and gradient are also shown in Fig. 14.

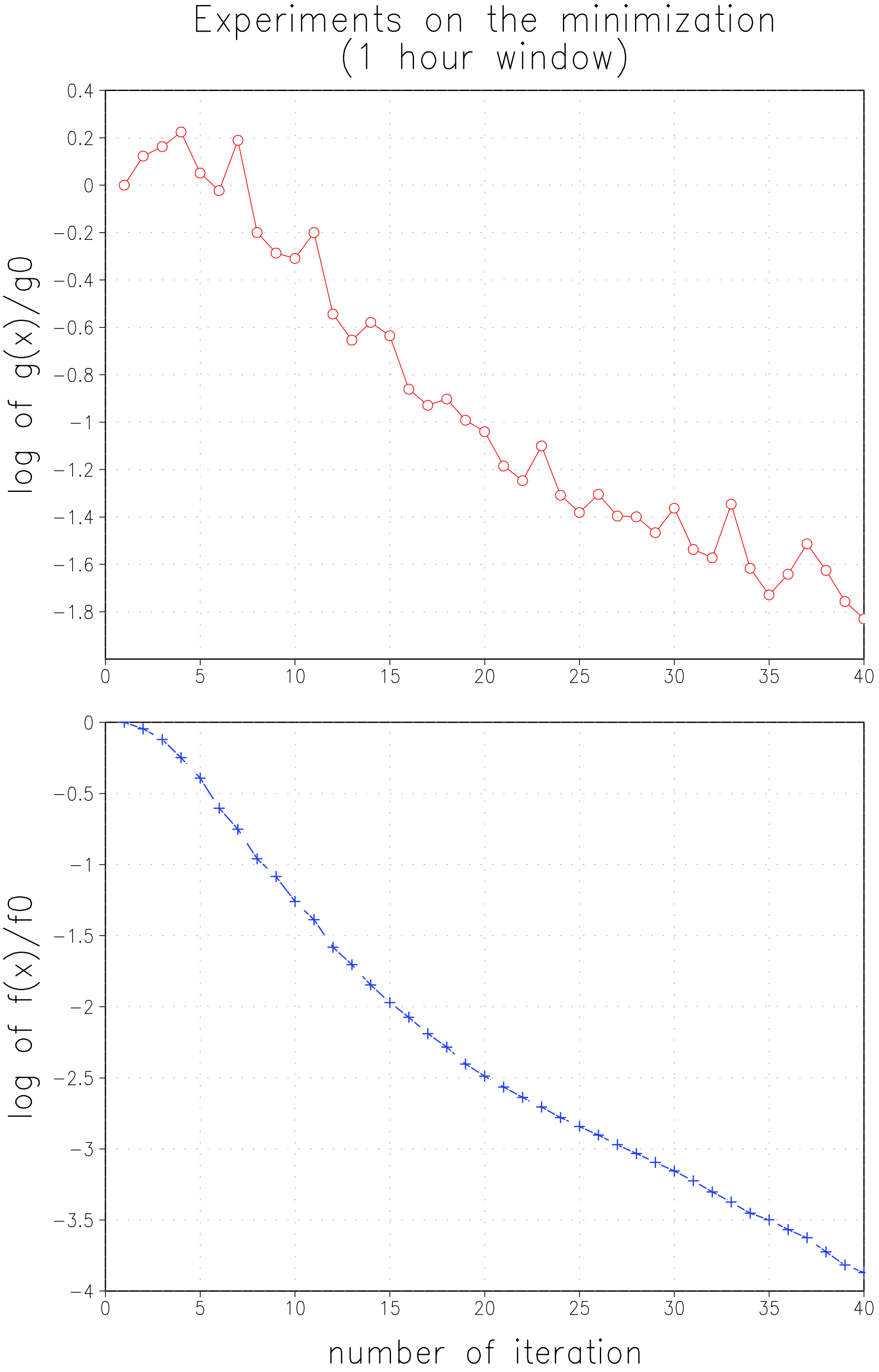

Figure 8: experiment of minimization for one hours DA window. Upper: $\log(||g||/||g_0||)$; Lower: $\log(||f||/||f_0||)$

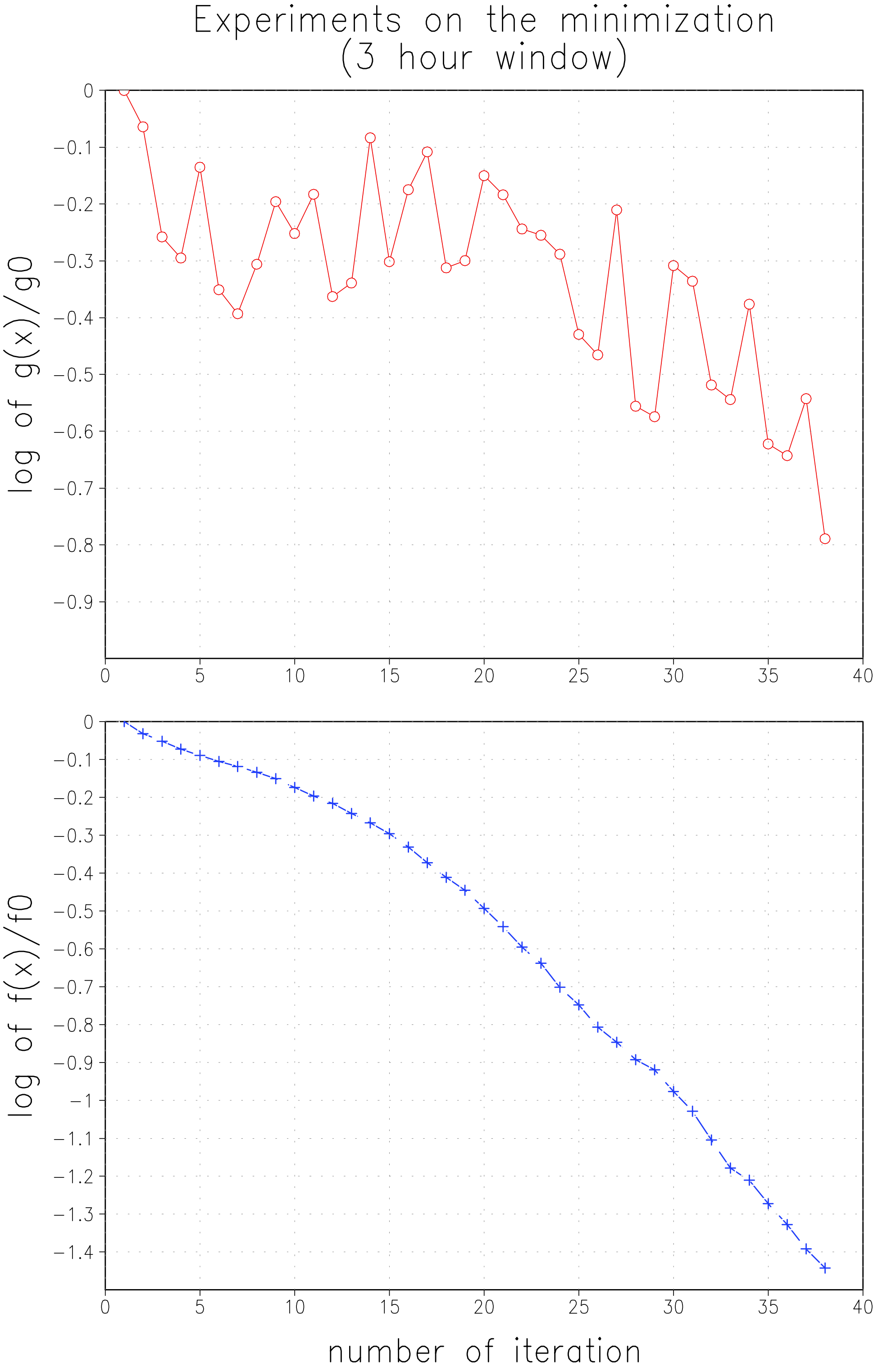

Figure 9: Same as Figure. 8 but for three hours DA window. Upper: $\log(||g||/||g_0||)$; Lower: $\log(||f||/||f_0||)$

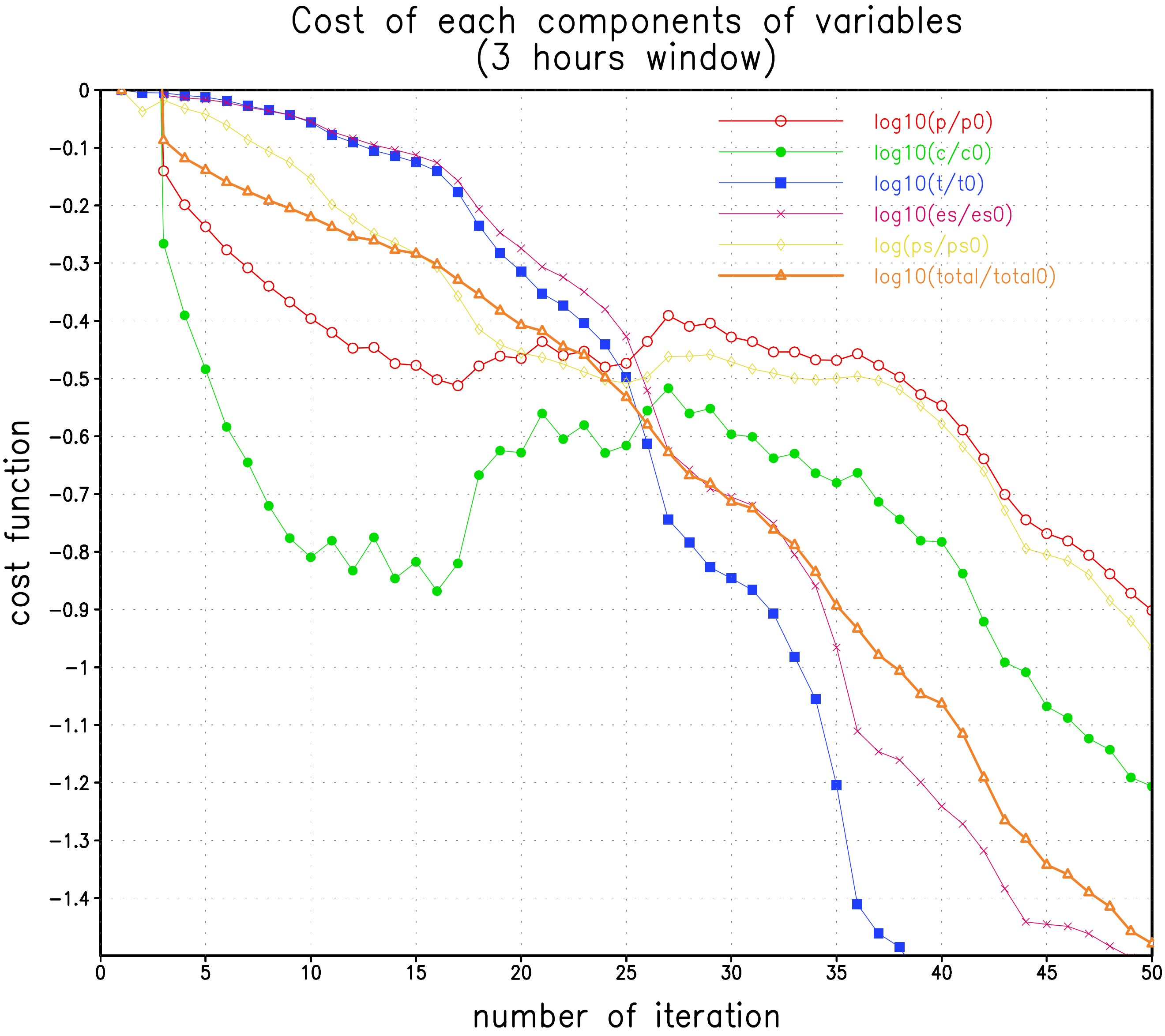

Figure 10: Cost of each component of variables

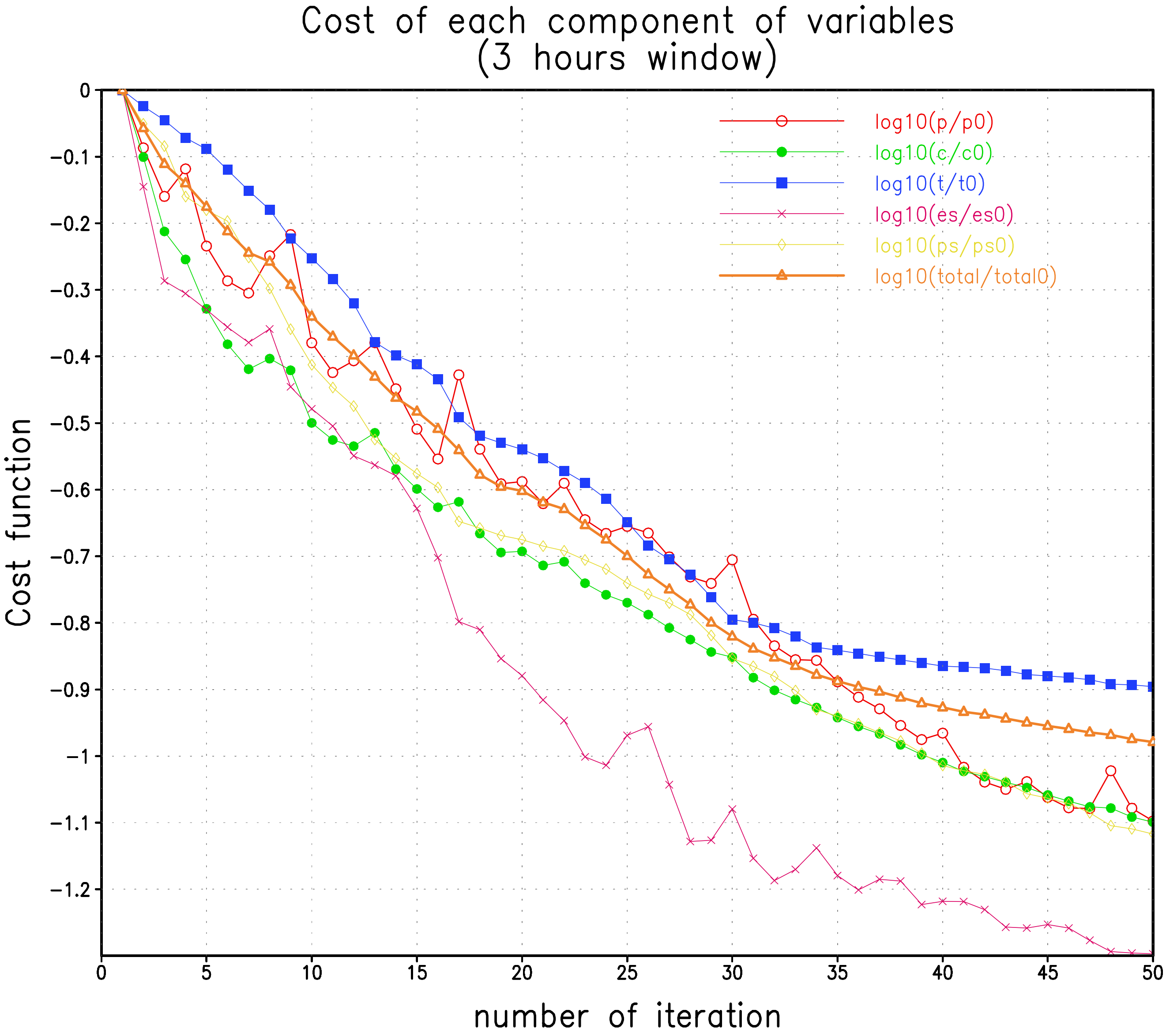

Figure 11: Same as Figure. 10 but with new scaling scheme.

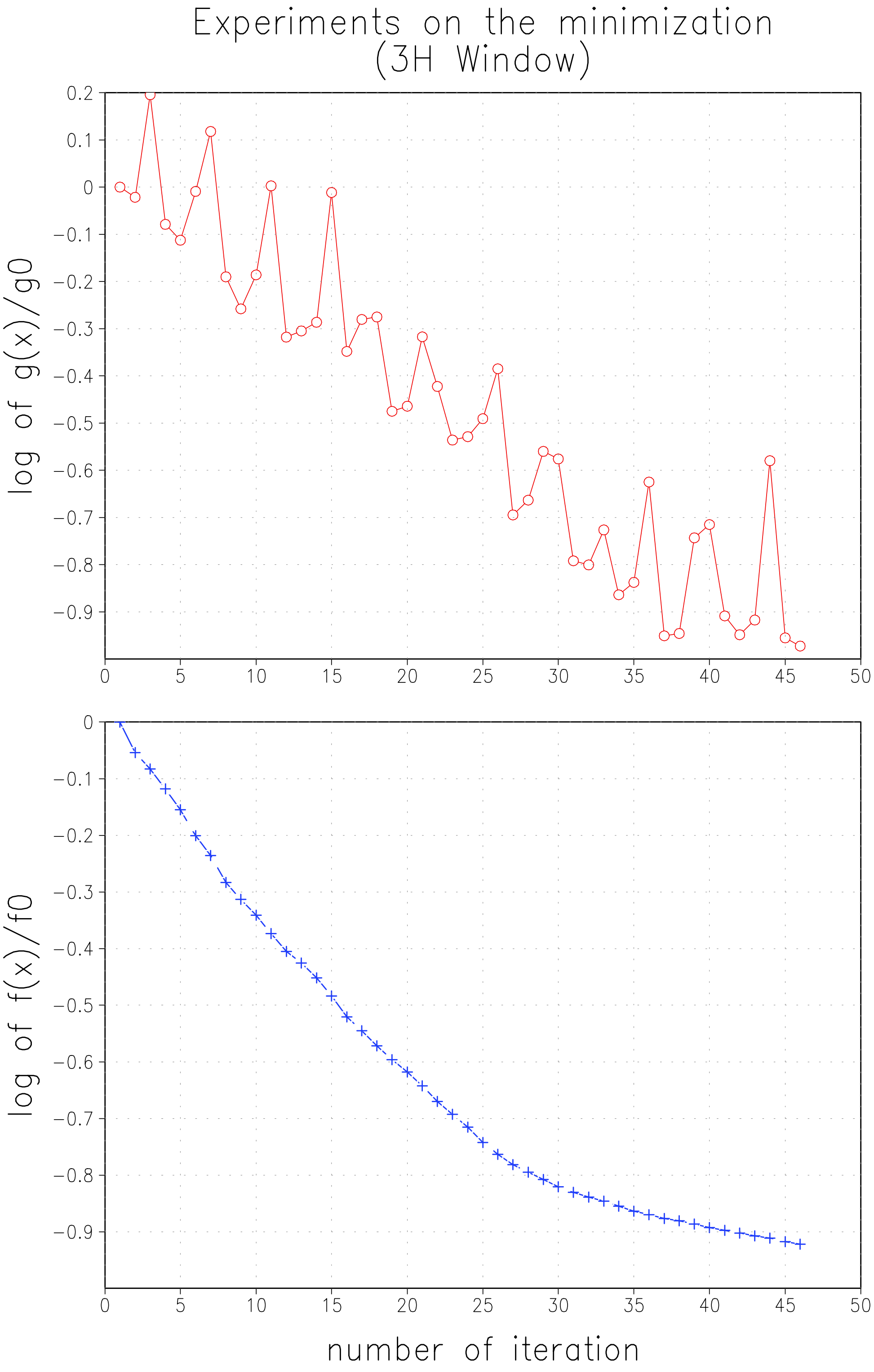

Figure 12: Same as Figure. 9 but with new scaling scheme. Upper: $\log(||g||/||g_0||)$; Lower: $\log(||f||/||f_0||)$

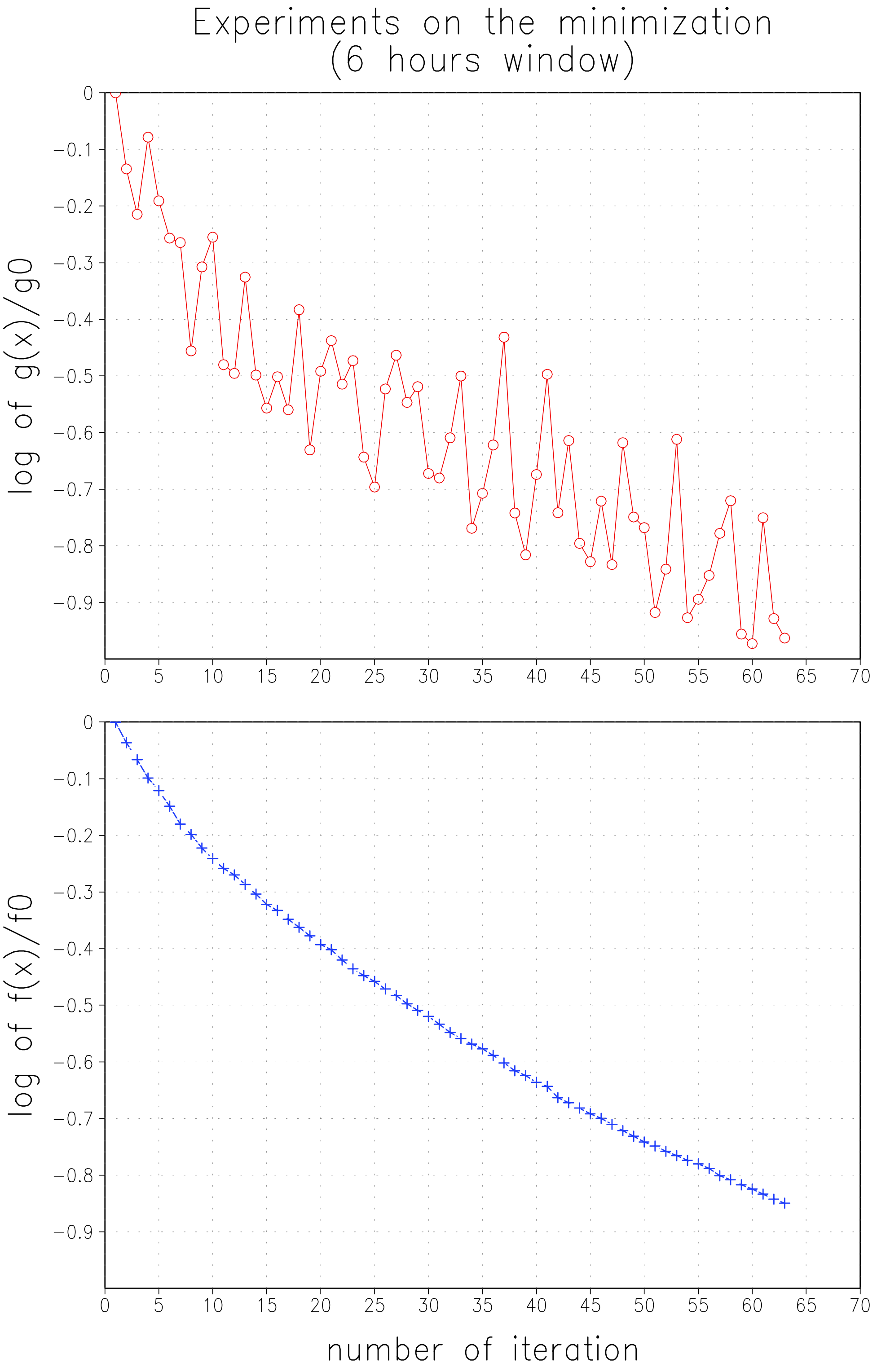

Figure 13: Same as Figure. 12 but for 6 hours window. Upper: $\log(||g||/||g_0||)$; Lower: $\log(||f||/||f_0||)$

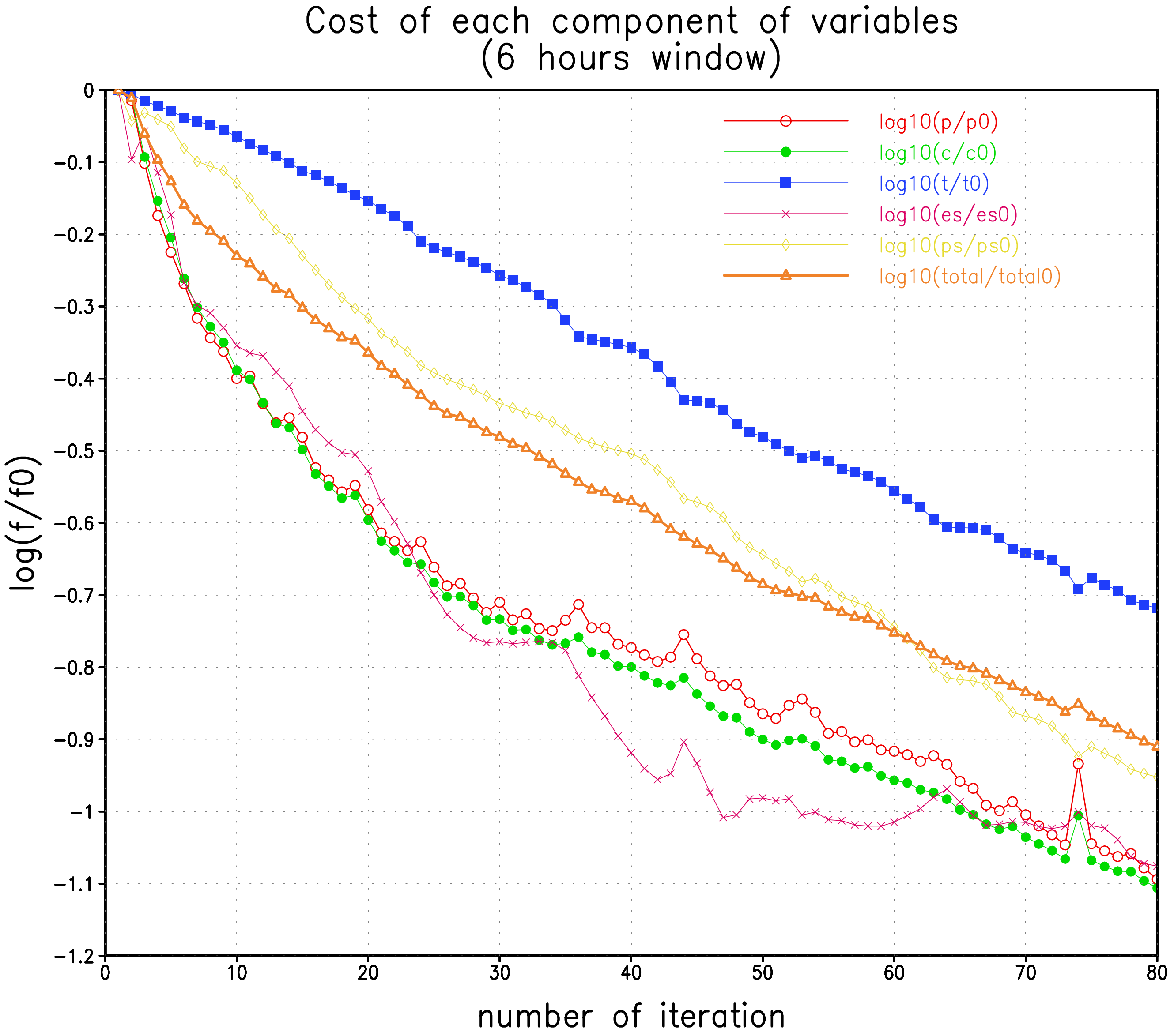

Figure 14: Same as Figure. 11 but for 6 hours window.

To evaluate the efficiency of minimization, we plot out the departure of geopotential height field from the control run. Fig. 15 shows the departure by integrate the forward model 3 hours with the perturbed initial data. While Fig. 16 shows the departure by integrate the forward model 3 hours with the optimalized initial data. It can be seen that the departure from the unperturbed data are reduced.

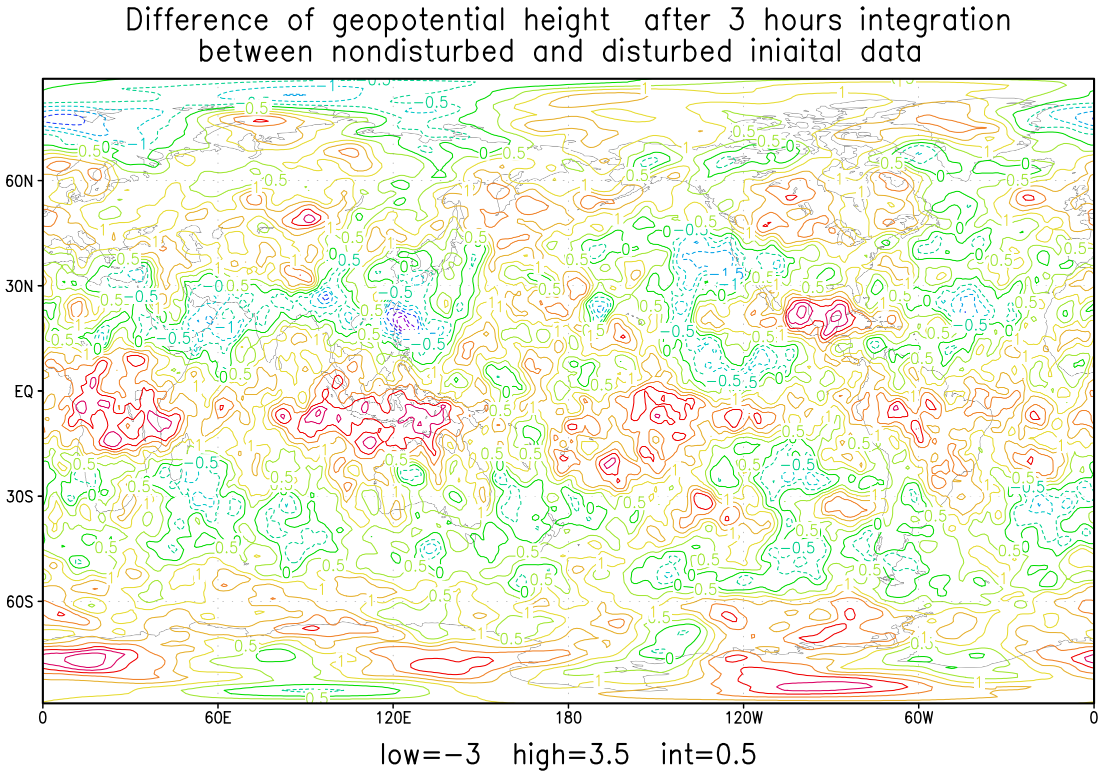
Difference of geopotential height after 3 hours integration
between nondisturbed and disturbed iniaital data
60N
30N
EQ
30S
60S
0
60E
120E
180
120W
60W
0
low=−3 high=3.5 int=0.5

Figure 15: Departure from the undisturbed data for disturbed initial data

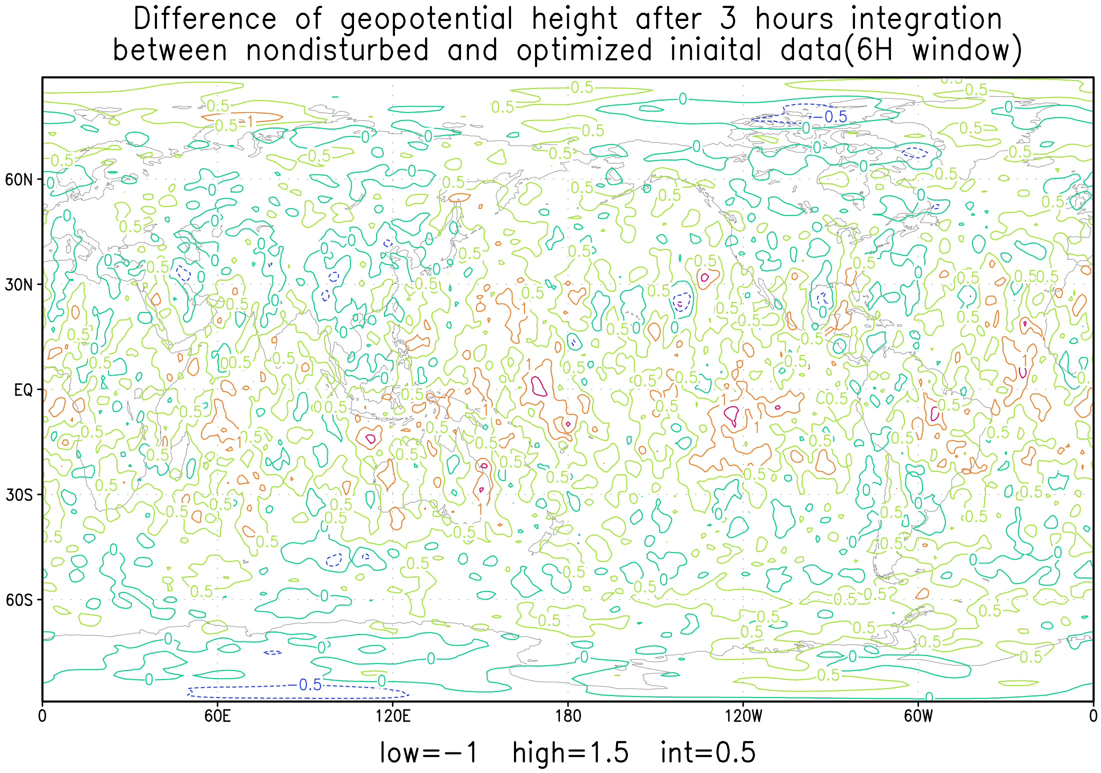
Difference of geopotential height after 3 hours integration
between nondisturbed and optimized iniaital data(6H window)
60N
30N
EQ
30S
60S
0
60E
120E
180
120W
60W
0
low=−1 high=1.5 int=0.5

Figure 16: Departure from the undisturbed data for optimized initial data

5 Summary and Conclusions

Using the new MPI version of FSU Global Spectral Model, we derived the tangent linear and adjoint models of the dynamical core of FSU model in a framework of MPI parallel processing computation.As opposed to the previous version of the adjoint model of FSU spectral model, which used the variables in real space, here the spectral coefficients of the state variables served as the basic variables.

We carried out a series of accuracy tests on the tangent linear and adjoint models.This was followed by an accuracy check of the gradient of the cost function. We then carried out a series of experiments for the minimization of the cost functional for various lengths of the assimilation window namely for 1h, 3h and 6 hours respectively.

Results obtained show that for the tangent linear model, the relative error between the nonlinear model and tangent linear model decreases linearly from 10^{-1} to 10^{-7} as the factor α of the perturbation decreases from 10^{-1} to 10^{-6}. A further reduction of the factor α to 10^{-15},leads the relative error to increase linearly slightly exceeding the identity due to the accumulation of round-off errors, which means that the tangent linear model is highly accurate within a sizable interval of perturbation.

We checked both the full adjoint model and each subroutine, since some variables are dominant while other variables are very small in magnitude compared with the dominant variables. Thus we need to select an adequate scaling very carefully for each of the control variables. In the tests, we also chose individually the factor of perturbation of one variable to be non zero, leaving all other variables to be zero,thus avoiding the occurrence to of minor errors. For double precision calculation, each result of the tests yields an accuracy of 13 digits or more, which means the adjoint is reliable versus the tangent linear model.

We also verified the accuracy of the gradient of the cost function. All the tests were carried out with different lengths of the assimilation windows and all of them yielded satisfactory results.

Since our main purpose in developing the adjoint model is to assimilate real observational data distributed in space and time, which in turns requires minimization of the cost functional with respect to the control variables we need to employ an efficient and robust minimization algorithm. We thus tested the minimization of the cost functional using the limited memory quasi-Newton method(L-BFGS) which is now used at all operational centers for 4-D VAR data assimilation. The results show that an appropriate scaling scheme is paramount to the success of the minimization, while using an unbalanced scaling scheme leads to inefficient minimization or its total failure.

To summarize the above ingredients led to a successful coding and implementation of the MPI based adjoint of the dynamical core of the FSUGSM.

6 Future work

We are now in the middle of the process of coding the tangent linear and adjoint model of the full physics FSUGSM spectral model which include a variety of on/off processes.

Once the MPI version of adjoint model with full physics is validated, we will proceed to carry out the experiments outlined in the new NSF proposal related to singular vector decomposition and use of second order adjoint to derive the Hessian singular vectors and implementation of rank deficient Kalman filter in 4-D VAR.

References

[1] Cacuci, D. G., 1988: The forward and adjoint methods of sensitivity analysis. 71-144. Chapter 3 of the book “Uncertainty Analysis” by Yigal Ronen, CRC Press, Inc., pp282.

[2] Chao W. C. & Chnag, L. P.,1992, Development of a 4-dimensional variational analysis system using the adjoint method at GLA. Part I: Dynamics, Mon. Wea. Rev., Vol. 120,pp 1661-1673

[3] Cocke, S., 1998: Case Study of Erin Using the FSU Nested Regional Spectral Model. Mon. Wea. Rev., 126, 1337-1346.

[4] Courtier P, 1985, Presentation d’une methode variationelle d’assimilation by dynamique de donnees meteorologiques reparties dans l’espace et le temps. Note de Travail No. 101, Direction de la Meteorologie, France.

[5] Courtier P, Talagrand O, 1987, Variational Assimilation of Meteorological Observations With the Adjoint Vorticity Equation .2. Numerical Results, Q. J. R. M. S. 113 (478): 1329-1347

[6] Daley Roger, Claude Girard, John Henderson and Ian Simmonds, 1976: Short- Term Forecasting with a Multi-Level Spectral Primitive Equation Model. Part I - Model Formulation. *Atmosphere*, **14, No 2**, 99–116.

[7] Derber J. C. (1985) The variational 4-D assimilation of analysis using filtered models as constraints, Ph.D These, University of Washington–Madison, Madison, Wisconsin 53706, 142pp

[8] Derber J. C., 1989, A variational continuous assimilation technique, Mon. Wea. Rev., Vol.117, pp.2437–2446

[9] Krishnamurti TN, Bedi HS, 1988, Cumulus Parameterization and Rainfall Rates .3. Mon. Weather Rev. 116 (3): 583-599

[10] Krishnamurti, T.N. ,X Jishan,H.S.Bedi, K.Ingles an d D. Osterhof,1991: Physical initialization for numerical weather prediction over tropics. Tellus, 43A, 53-81.

[11] Krishnamurti, T.N. ,H.S.Bedi, K.Ingles,1993: Physical initialization using SSM/I rainfall rates. Tellus,45A,247-269.

Robert A.J., 1966, The integration of a low-order spectral form of the primitive meteorological equations, J. Meteor. Soc. Japan, Serial 2, Vol. 44, pp.237–245

[12] LeDimet, F.X. & Talagrand, O., 1986, Variational algorithms for analysis and assimilation of Meteorological Observations: Theoretical Aspects. Tellus, 38A, 97–110

[13] Le Dimet,F. X., I. M. Navon and Dacian N. Daescu,2002: Second Order Information In Data Assimilation. Monthly Weather Review, Vol. 130,No.3 ,629-648.

[14] Lewis J.M. & Derber J.C.,1985, The use of adjoint equations to solve a variational adjustment problem with advective constrains. Tellus, 37A, Vol.4, pp.309-322

[15] Li,Y., I.M.Navon, Philippe Courtier and Pierre Gauthier,1993: Variational Data assimilation with a Semi-Implicit Semi-Lagrangian Global Shallow-Water Equations Model and its Adjoint, Monthly Weather Review, Vol. 121, No 6, 1759–1769.

[16] Yong Li, I.M.Navon, W. Yang, X.Zou J.R. Bates, S.Moorthi, R.W. Higgins, 1994: 4-D Variational Data Assimilation Experiments with a Multilevel Semi-Lagrangian Semi-Implicit GCM, Monthly Weather Review, Vol. 122, No 5, 966–983.

[17] Li, Zhijin, I.M. Navon, Yanqiu Zhu, 2000, Performance of 4D-Var with Different Strategies for the use of Adjoint Physics with the FSU Global Spectral Model, Vol. 128, pp.668–688

[18] Liu, D.C. and Jorge Nocedal, 1989, On the limited memory BFGS method for large scale minimization. Math. Prog., Vol. 45, pp.503–528

[19] Navon, I.M, X. Zou and J. Derber and J. Sela, 1992, Variational Data Assimilation with an Adiabatic Version of the NMC Spectral Model, Mon. Wea. Rev, Vol.120, No.7, pp.1433–1446

[20] J. Nocedal, 1980, Updating quasi-Newton matrices with limited storage, Mathematics of computation, Vol.35, pp. 773–782, 1980.

[21] Rabier F, P. Courtier,M Hervou, B. Strauss and A Persson, 1993:Sensitivity of Forecast error to initial conditions using the adjoint model.**ECMWF Tech Memo**, **197**,29pp.

[22] Rabier F, P. Courtier and O.Talagrand, 1992: An Application of adjoint models to sensitivity.*Beitr. Phys. Atmosph.*, **Vol 65, No 3**, 177- 192.

[23] Thépaut J.N. & Courtier P., 1991, Four-dimensional variational data assimilation using the adjoint of a multilevel primitive equation model. Q. J. R. M. S., Vol.117, 1225–1254

[24] Tsuyuki Tadashi, 1996a: Variational data assimilation in the tropics using precipitation data. Part I: Column model. *Meteor. Atmos. Phys.*, **60**, 87-104.

[25] Tsuyuki Tadashi, 1996b: Variational data assimilation in the tropics using precipitation data. Part II: 3-D model. *Mon. Wea. Rev.*, **124**, 2545-2561.

[26] Tsuyuki Tadashi, 1997: Variational data assimilation in the tropics using precipitation data. Part III: Assimilation of SSM/I precipitation rates. *Mon. Wea. Rev.*, **125**, 1447-1464.

[27] Yanqiu Zhu and I. M. Navon , 1998, FSU-GSM Forecast Error Sensitivity to Initial Conditions: Application to Indian Summer Monsoon Meteorology and Atmospheric Physics , Vol.68, No.1-2 35-41

[28] Yanqiu Zhu and I.M. Navon,1999: Impact of Key Parameters Estimation on the Performance of the FSU Spectral Model using the Full Physics Adjoint, Monthly Weather Review, Vol 127, No 7, 1497-1517.

[29] Zhi Wang, 1993: Variational Data Assimilation with 2-D Shallow Water Equations and 3-D FSU Global Spectral Models, *Ph. D. Dissertation*, Department of Mathematics, College of Arts and Sciences, The Florida State University, 235 pp.

[30] Zou, X., Navon, I. M. & LeDimet, F. X. (1992b) An optimal nudging data assimilation scheme using parameter estimation. *Q. J. of Roy. Met. Soc* ., **118**, 1163-1186.

[31] Zou, X. I.M.Navon, A Barcilon, Jeff Whittaker and Dan Cacuci, 1993b: An Adjoint Sensitivity Study of Blocking in a Two-Layer Isentropic Model. *Monthly Weather Review*, **121, No 10**, 2833–2857.

[32] Zou, X., I. M. Navon, M. Berger, M. K. Phua, T. Schlick, and F. X. LeDimet, 1993a: Numerical Experience with Limited-Memory, Quasi-Newton Methods for Large-Scale Unconstrained Nonlinear Minimization, SIAM Journal on Optimization, **3**, No 3, 582-608.

[33] Zou, X., Navon, I. M. & Sela, J. G. (1993c) Variational data assimilation with moist threshold processes using the NMC spectral model. *Tellus*,**45A**, 370–387.

[34] Zupanski M,1993: A Preconditioning Algorithm for Large-Scale Minimization Problems, Tellus A 45A (5): 478-492 .

[35] Zupanski M.,1996: A preconditioning algorithm for four-dimensional variational data assimilation, Mon Weather Rev 124 (11): 2562-2573