SPEEDY DA (3D-VAR)

Prepared by Junjie Liu, Takemasa Miyoshi and Juan Ruiz

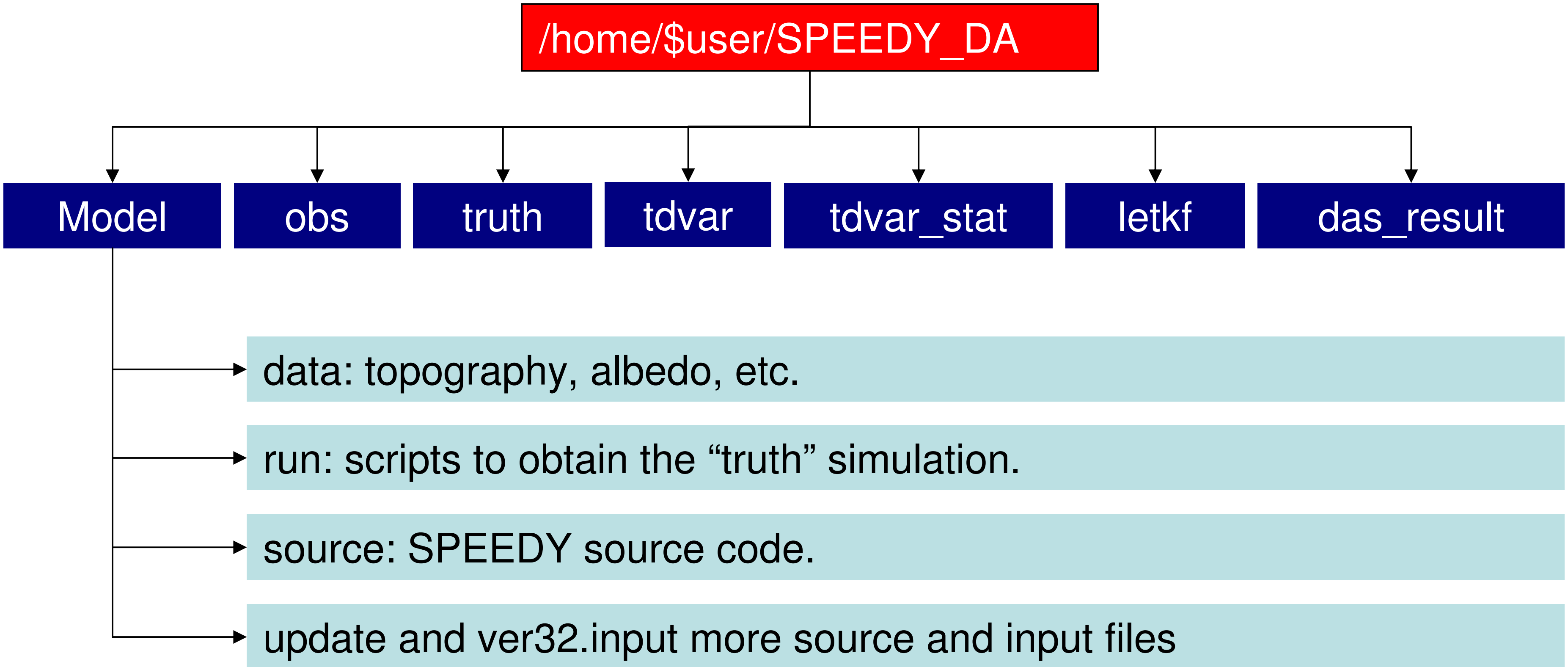
Buenos Aires, Argentina

27 October – 7 November 2008

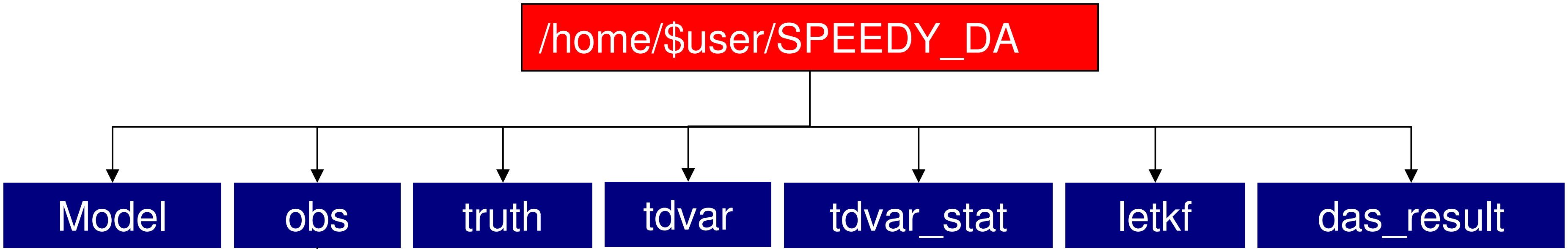
What will we learn in this lesson:

- Speedy general circulation model.
- 3D-VAR implementation formula.
- How to construct the error covariance matrix (the NMC method)
- Characteristics of the error structure
- Response test with 3D-VAR (assimilating only one observation)
- Run 3DVAR for a realistic rawindsonde observation network

Directory description:



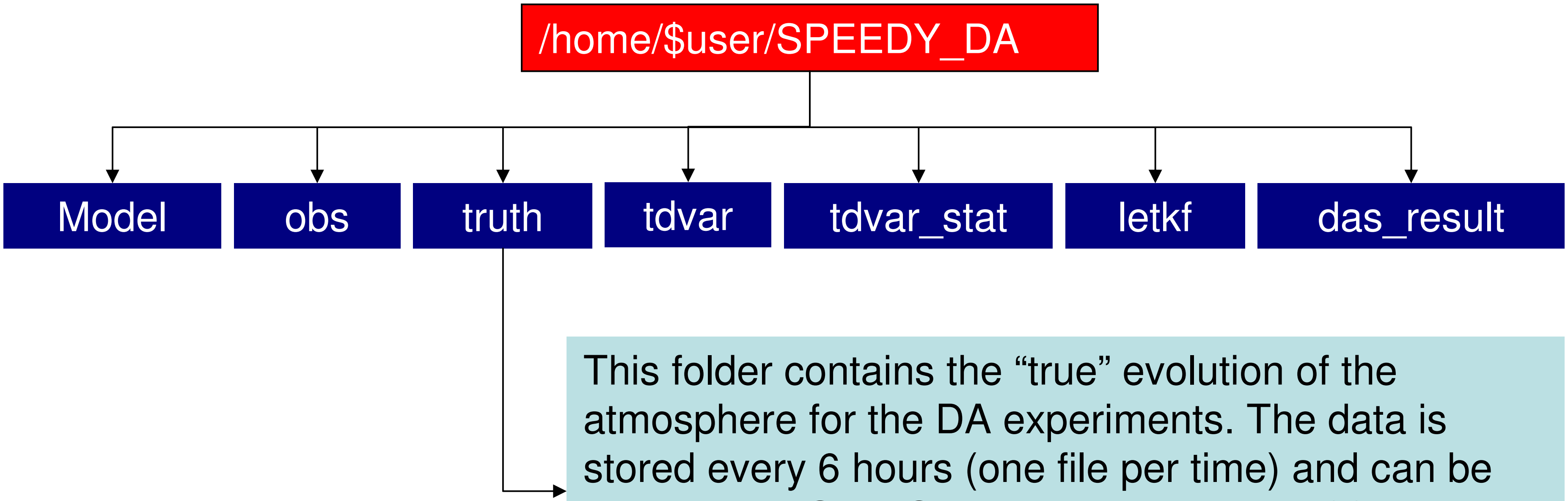
Directory description:



This folder contains the "observation" files. The observations are generated adding a random noise to the truth simulation. The random is added on each variable at each vertical level and grid point. The amplitud of the random noise is different for each variable (and is defined in the fortran program letkf/common_speedy.f90)

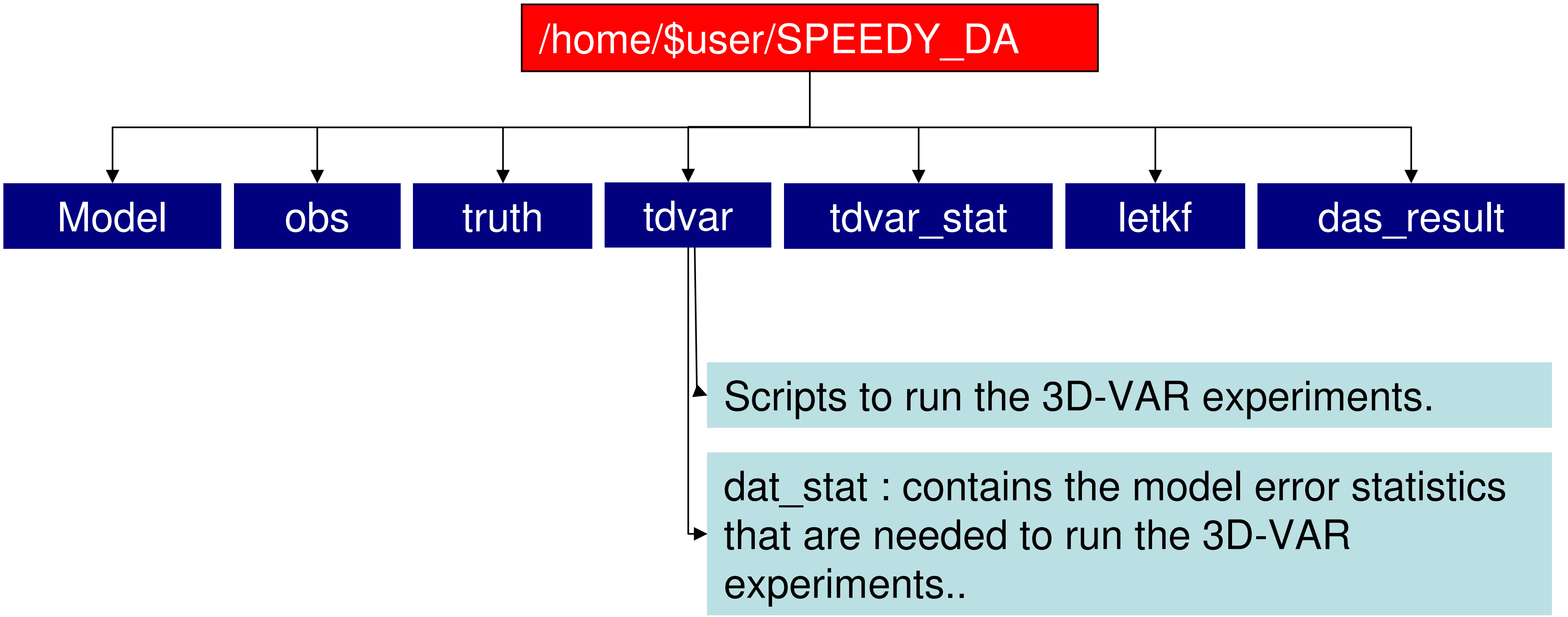
A new set of observations can be generated using the script letkf/create_obs.sh

Directory description:

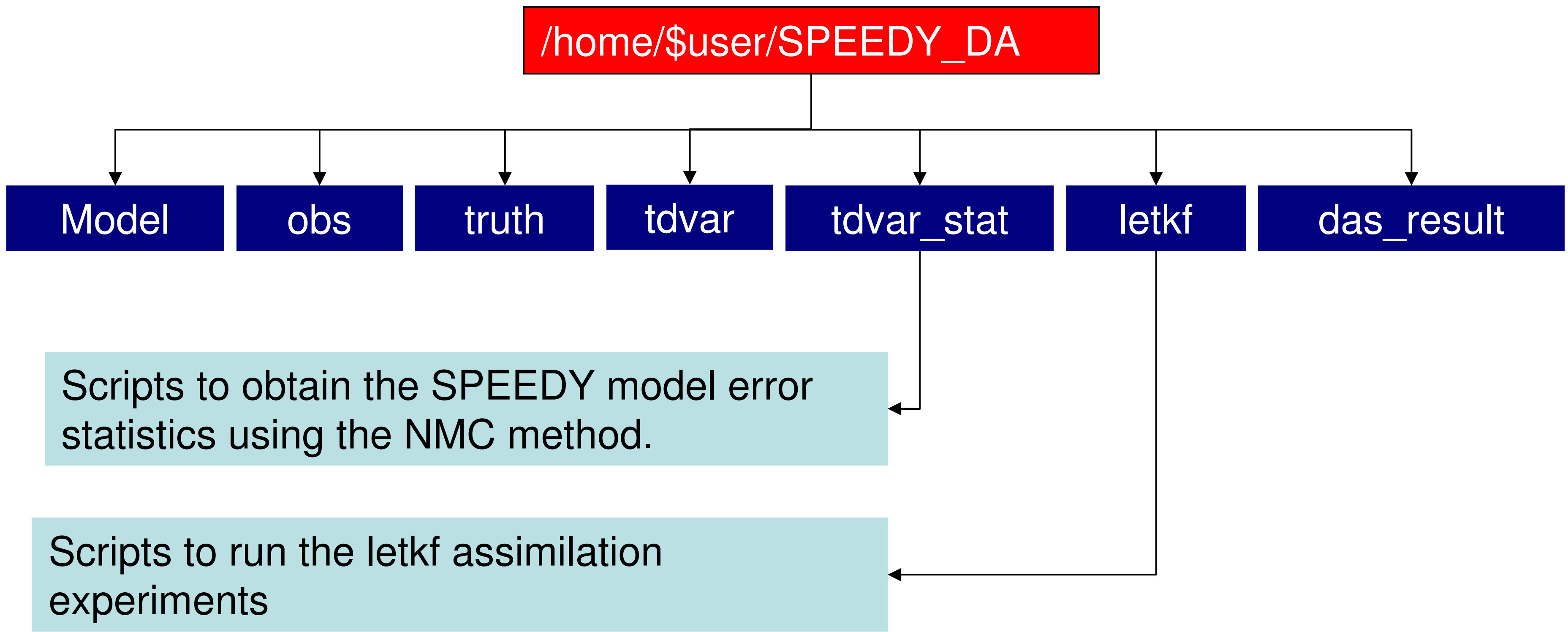


This folder contains the “true” evolution of the atmosphere for the DA experiments. The data is stored every 6 hours (one file per time) and can be viewed with GrADS. There are two sets of files one contains the data in the original sigma levels the other contains the data in pressure levels (*_p)

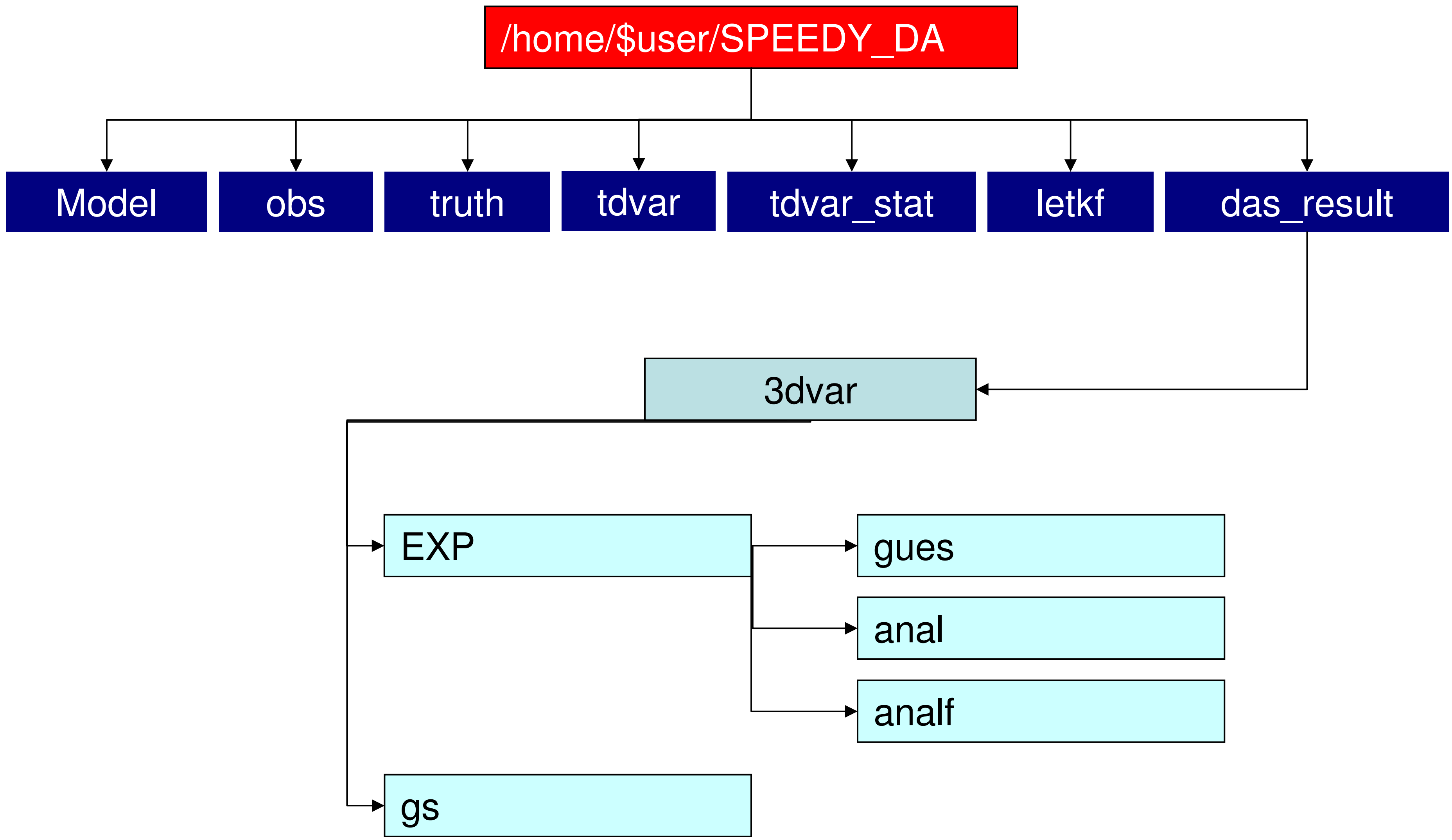
Directory description:



Directory description:



Directory description:



The SPEEDY (Simplified Parameterizations, primitivE-Equation DYnamics) model, main characteristics:

- General circulation model.
- Resolution: Spectral T30 (aprox 4º) and 7 sigma vertical levels.
- Available parameterizations (simple schemes):
    - Radiation (long wave and short wave)
    - Cumulus (mass flux scheme)
    - PBL
    - Large scale condensation
    - Land model

    - More information available in http://www.ictp.trieste.it/~moltenif/speedy-doc.html

3D-Var implementation formula

Cost function:

$$J(\delta\mathbf{x}) = \frac{1}{2}\delta\mathbf{x}^T\mathbf{B}^{-1}\delta\mathbf{x} + \frac{1}{2}(\mathbf{H}\delta\mathbf{x} - \mathbf{d})^T\mathbf{R}^{-1}(\mathbf{H}\delta\mathbf{x} - \mathbf{d})$$

$$\delta\mathbf{x} = \mathbf{x} - \mathbf{x}^b \qquad \mathbf{d} = \mathbf{y} - H(\mathbf{x}^b)$$

Background error covariance is too large to get the inverse, so we define a variable transformation (Barker et al., 2004):

$$\delta\mathbf{x} = \mathbf{U}\delta\mathbf{v}$$

$$J(\delta\mathbf{v}) = \frac{1}{2}\delta\mathbf{v}^T\mathbf{U}^T\mathbf{B}^{-1}\mathbf{U}\delta\mathbf{v} + \frac{1}{2}(\mathbf{H}\mathbf{U}\delta\mathbf{v} - \mathbf{d})^T\mathbf{R}^{-1}(\mathbf{H}\mathbf{U}\delta\mathbf{v} - \mathbf{d})$$

3D-Var implementation (continued)

In order to make the covariance matrix to be the identity:

$$\mathbf{B} = \mathbf{U}\mathbf{U}^T$$

$$J(\delta\mathbf{v}) = \frac{1}{2}\delta\mathbf{v}^T\delta\mathbf{v} + \frac{1}{2}(\mathbf{H}\mathbf{U}\delta\mathbf{v} - \mathbf{d})^T\mathbf{R}^{-1}(\mathbf{H}\mathbf{U}\delta\mathbf{v} - \mathbf{d})$$

The gradient of the cost function is:

$$\nabla J(\delta\mathbf{v}) = \delta\mathbf{v} + \mathbf{U}^T\mathbf{H}^T\mathbf{R}^{-1}(\mathbf{H}\mathbf{U}\delta\mathbf{v} - \mathbf{d})$$

Using cost function and the gradient of the cost function, we use a quasi-Newton minimizer to find the solution $\delta\mathbf{v}$, and then convert $\delta\mathbf{v}$ to $\delta\mathbf{x}$

Quasi-Newton Minimizer

Quasi-Newton's method:

$$\delta \mathbf{v}_{k+1} = \delta \mathbf{v}_k - \alpha \mathbf{P}^{-1}(\delta \mathbf{v}_k) \nabla J(\delta \mathbf{v}_k)$$

Central idea underlying quasi-Newton method is to use an approximation of the inverse Hessian $\mathbf{P}^{-1}$

Inverse Hessian matrix here is the second derivative of the cost function, which is the analysis error covariance in the data assimilation framework

Construction of the variable transformation $\mathbf{U}$

$\delta\mathbf{x} = \mathbf{U}\delta\mathbf{v}$ $\quad$ $\mathbf{B} = \mathbf{U}\mathbf{U}^T$

The essential problem is to construct the variable transformation:

$$\mathbf{U} = \mathbf{VCA}$$

A: Error standard deviation. Each variable is scaled by its own error standard deviation (wich is a function of lat, lon and sigma level).

C: Horizontal and vertical error correlation. The horizontal error correlation is assumed to be Gaussian so only the characteristic length scale has to be determined (the X and Y length scales are computed independantly). The errors at different levels aren´t strongly correlated because the vertical resolution of the SPEEDY model is too coarse. So background errors are assumed to be uncorrelated in the vertical direction.

V: Intervariable correlation. A certain group of non-correlated variables should be selected to descrive the atmosphere state. In this case Ps, T, q, Uu and Vu are used. Where Uu and Vu are the umbalanced components of the wind defined as:

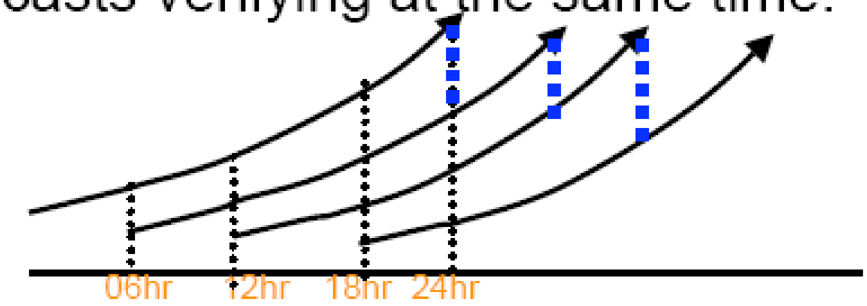
$$u_u = u - r_1 u_g(p_s, T)$$

$$v_u = v - r_2 v_g(p_s, T)$$

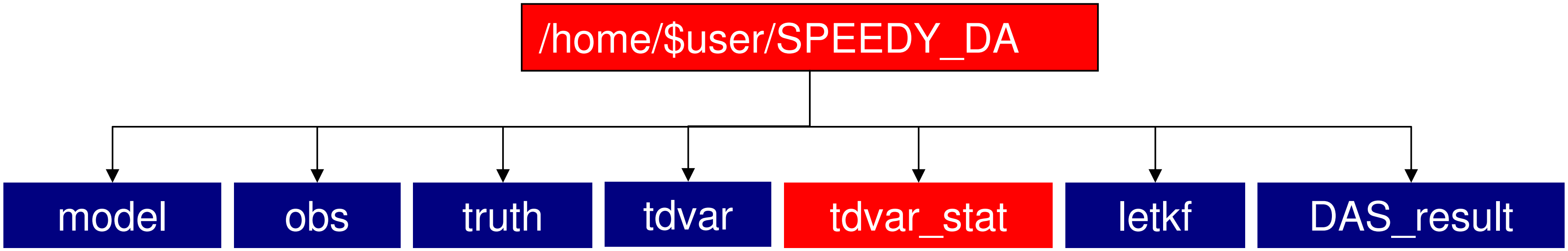
Where r1 and r2 are the correlation coefficients between the U and V components of the wind and the geostrophyc wind.

Creation of the transformation matrix U based on NMC method (Parrish and Derber, 1992)

The structure of the forecast error covariance is estimated as the average over many differences between two short-range model forecasts verifying at the same time.



1. The blue dashed lines represent the difference between 18hr forecast and 24hr forecast.
2. Based on these forecast differences, we can calculate all the statistics (A, C, V) need to construct the background error covariance B



Go to the tdvar_stat folder, you will find the following scripts:

-**tdvar_nmc.sh** : This script runs a data assimilation cycle for one month using 3DVAR. It also computes 24 hour forecast started from the analysis that will be used to get the error statistics using the NMC method.

In this assimilation cycle we don´t know anything yet about error correlations so the standard deviation of the errors is assumed to be constant for each variable.

The horizontal correlation is assumed to be Gaussian with a prescribed length scale (the same for X and Y direction).

The intervariable correlation is assumed to be 0 (however this is not true as we will see later).

- **nmc_stat.sh** : This scripts compiles and runs the nmc_stat.f90 program that will compute the error standard deviation, the horizontal length scale and the correlation between the wind and the geostrophic wind from the forecast obtained with the previous script. The results will be available in tdvar_stat/dat_stat and can be viewed with GrADS.

Exercises: Background error structure

Make plots of the results (use grads commands):

In the folder tdvar_stat/dat_stat you will find GrADS scripts to plot the results

⇒grads

=>reg.gs → The regression coefficient between v and geostrophic wind background errors

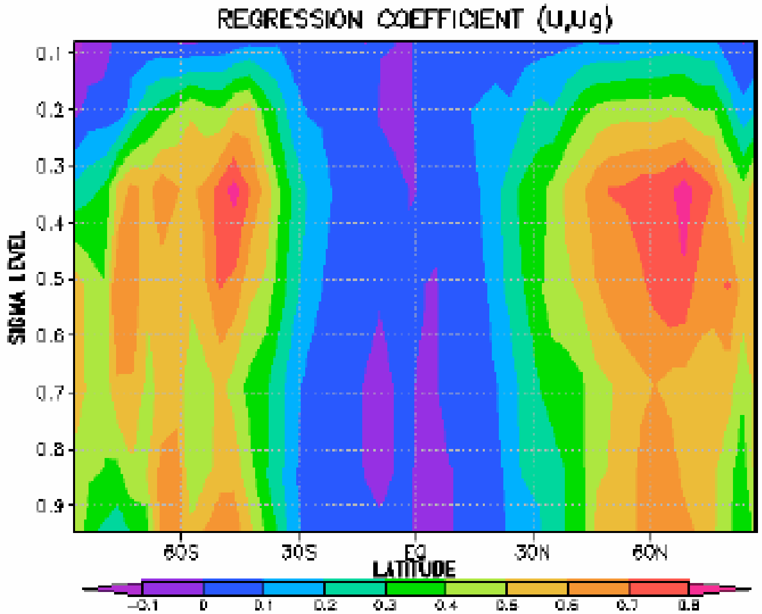
=>xcorr.gs

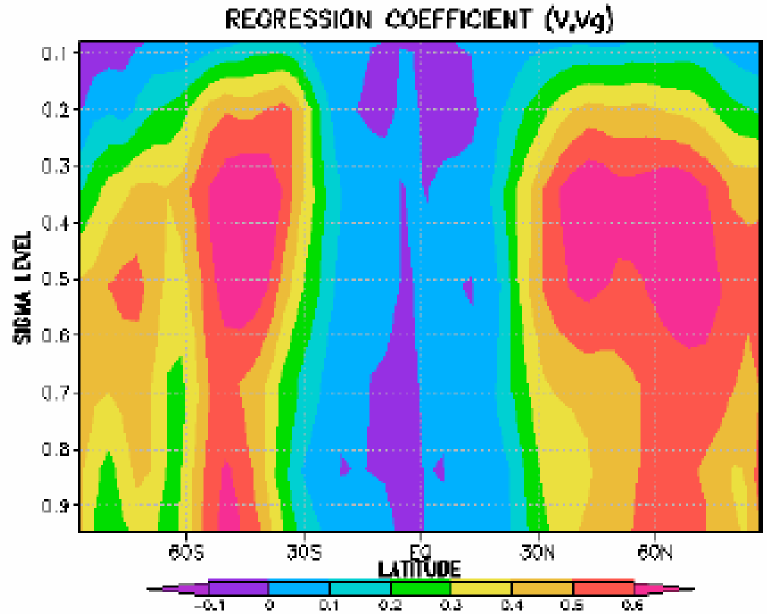
=>ycorr.gs

→ Correlation of the background errors in the x and y direction

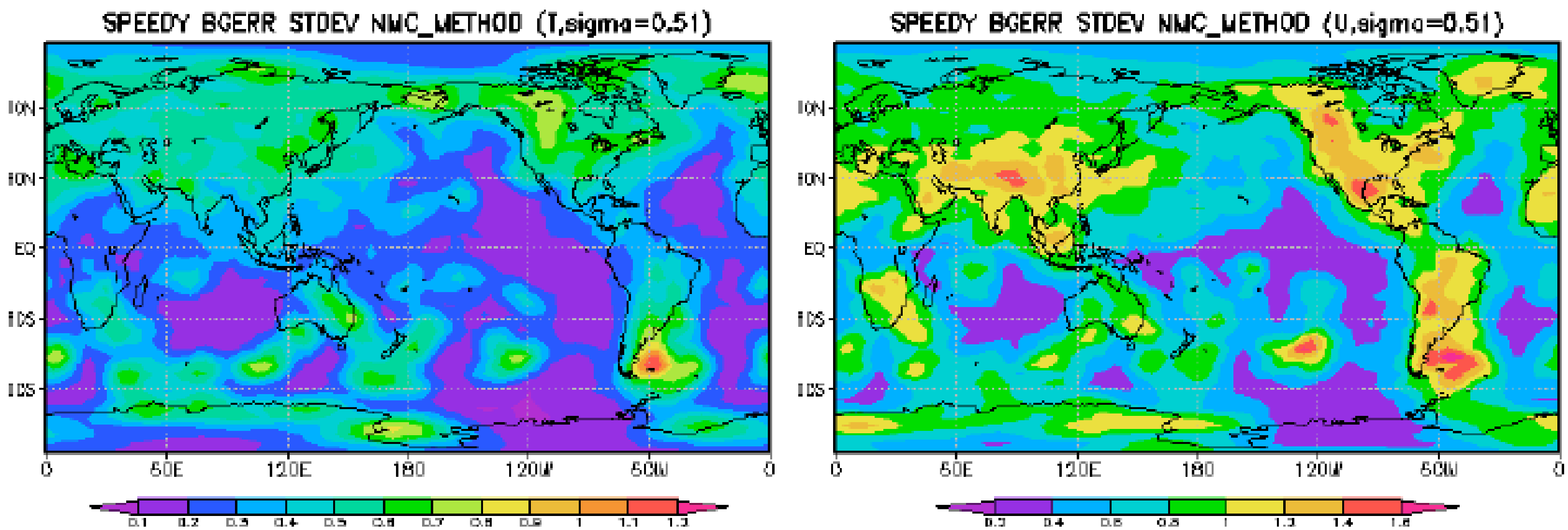
=>stdev.gs → Background error standard deviation

Inter-variable background error relationship (geostrophic balance)



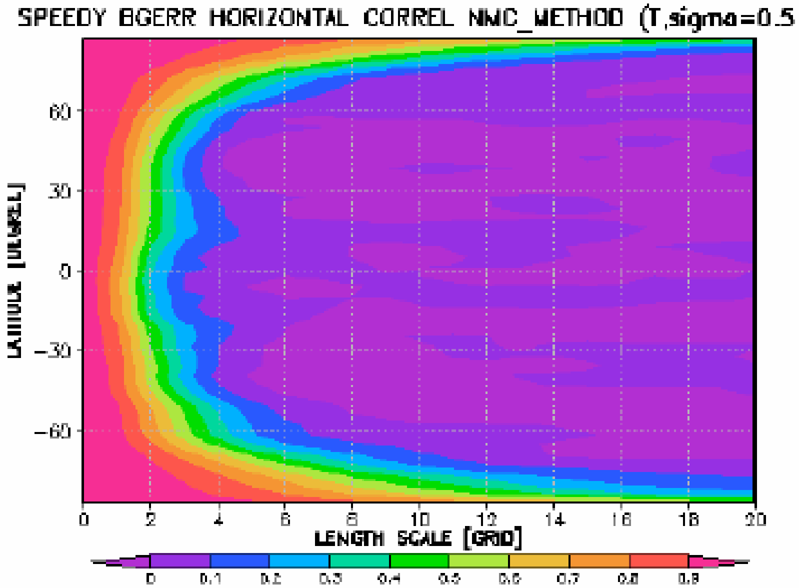


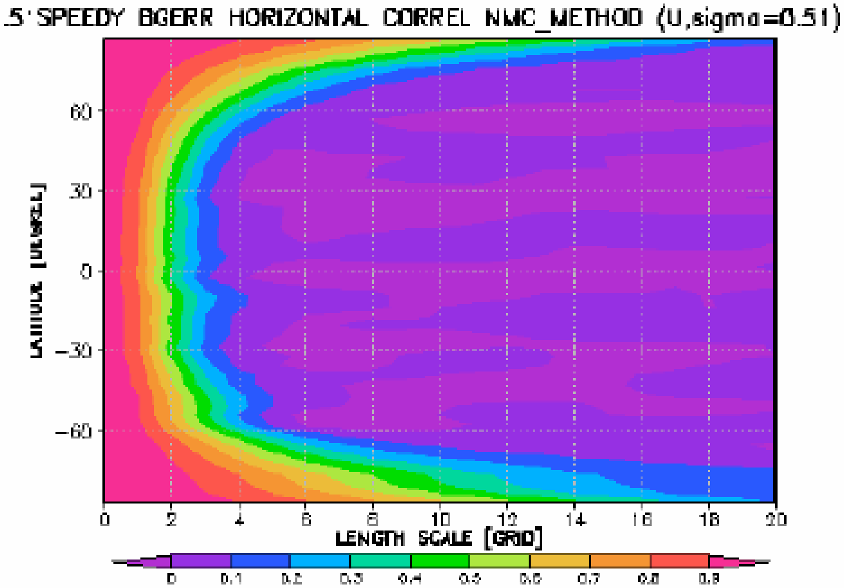
3D-Var estimated background error standard deviation



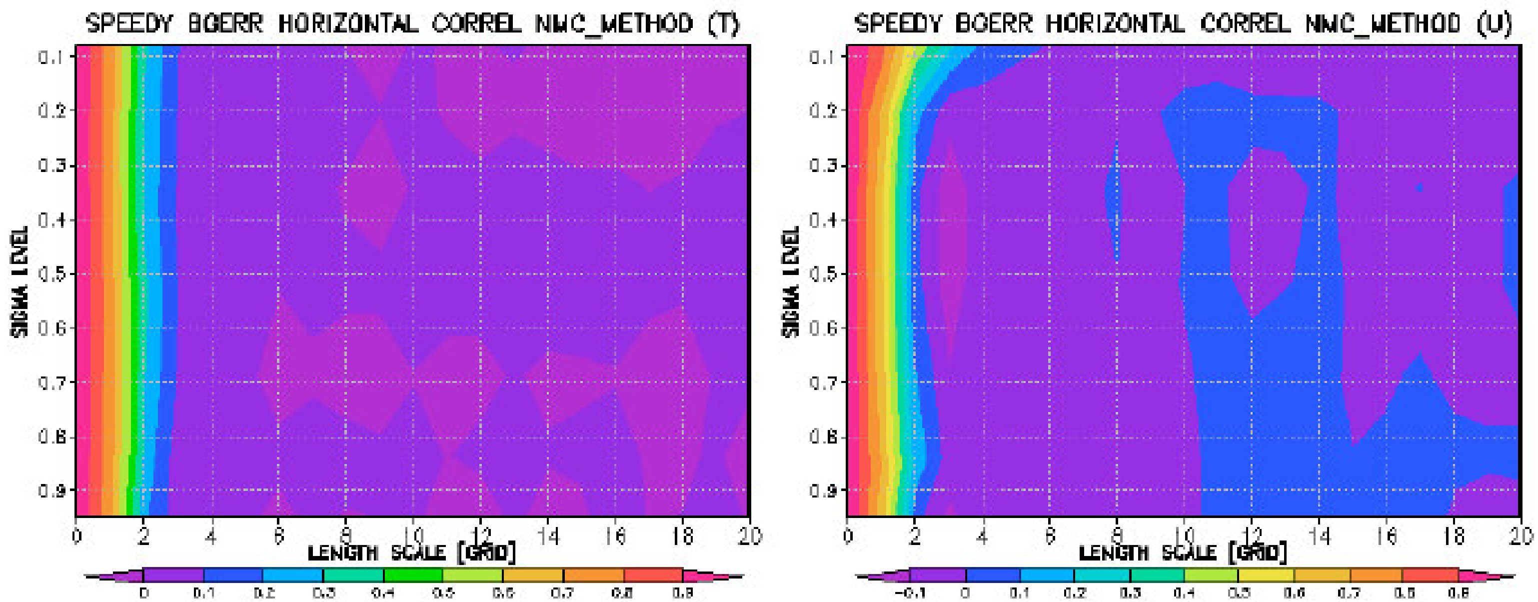
Strong spatial dependence of the background error standard deviation for both temperature (left panel) and zonal wind (right panel).

Zonal error correlation between adjacent grid points





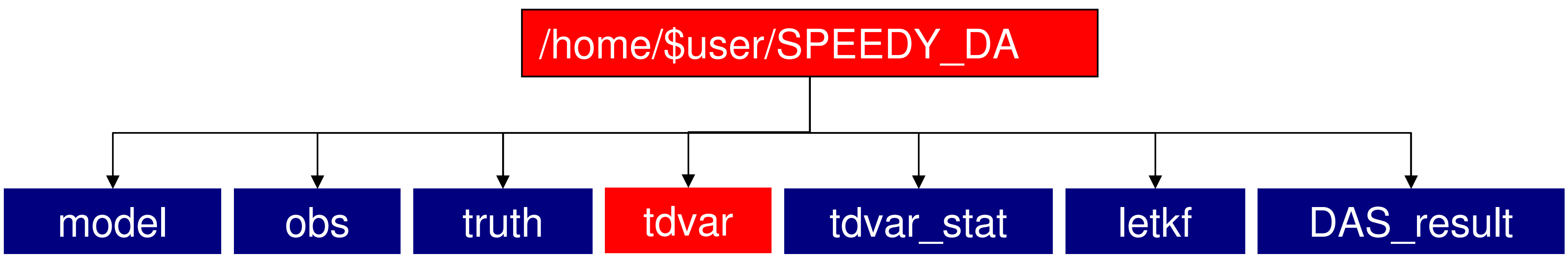
meridional error correlation



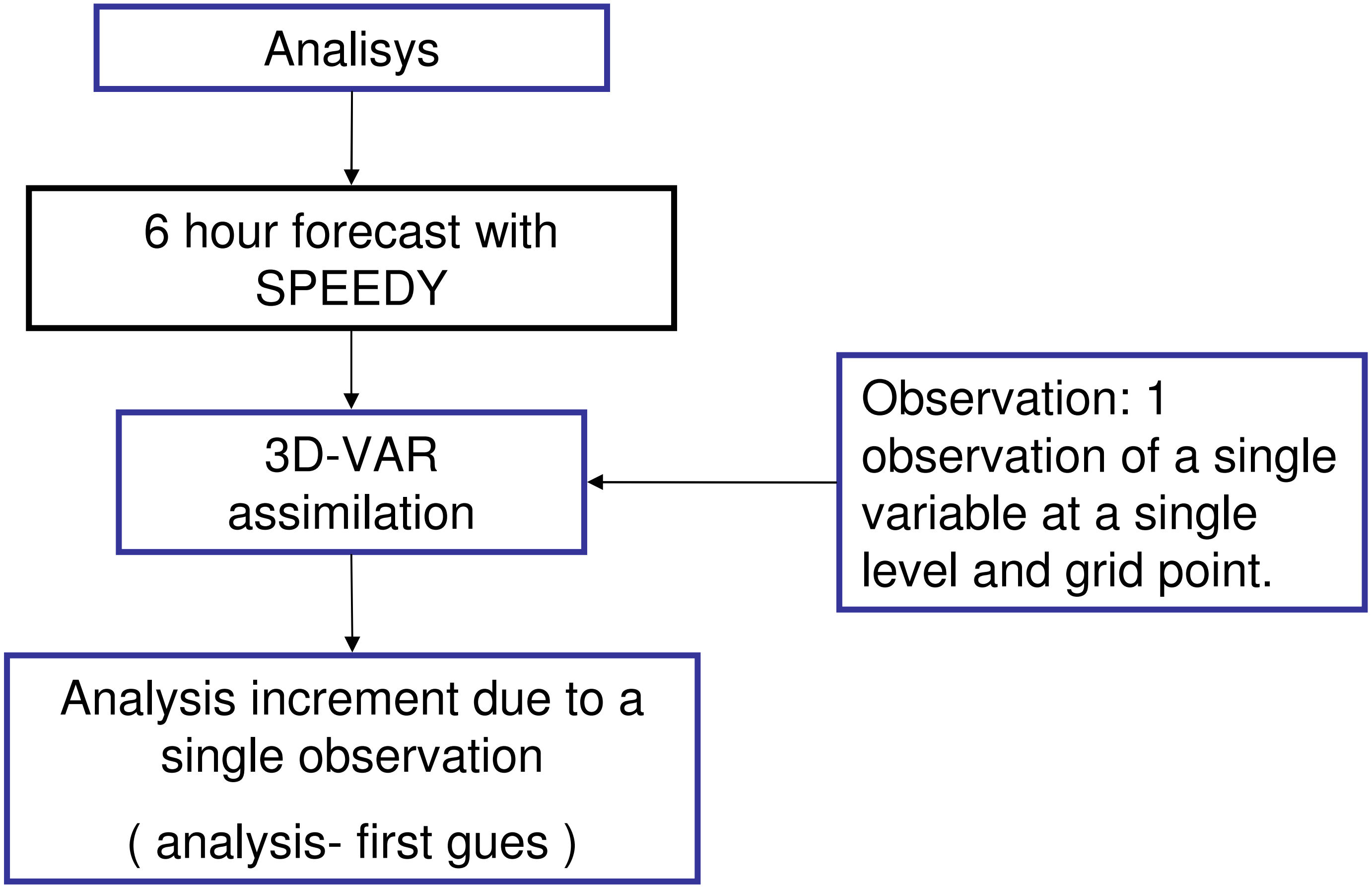
Almost has no dependence on height

The correlation decreases to zero beyond the second grid points

3D-VAR Response to a single observation



Response experiment:



Observation location:

This is controled by the ex_obs.f90 file.

```
PROGRAM ex_obs
  IMPLICIT NONE

  LOGICAL,PARAMETER :: msw_test=.TRUE.
  LOGICAL,PARAMETER :: msw_real=.FALSE.
  LOGICAL,PARAMETER :: msw_dnsobs=.FALSE.
  INTEGER,PARAMETER :: nlon=96
  INTEGER,PARAMETER :: nlat=48
  INTEGER,PARAMETER :: nlev=7

  INTEGER :: ilon,ilat,ios
  CHARACTER(10) :: ctmp1,ctmp2
  INTEGER :: i,j,k

  INTEGER :: ex_u(nlon,nlat,nlev) = 0
  INTEGER :: ex_v(nlon,nlat,nlev) = 0
  INTEGER :: ex_t(nlon,nlat,nlev) = 0
  INTEGER :: ex_q(nlon,nlat,nlev) = 0
  INTEGER :: ex_ps(nlon,nlat) = 0
  INTEGER :: ex_prec(nlon,nlat) = 0

  IF (msw_test) THEN
    ex_u(76,35,4) = 1
  ELSE IF (msw_real) THEN
    OPEN(10,file='obsmark.gs')
    READ(10,'(A)')
    READ(10,'(A)')
    READ(10,'(A)')
    READ(10,'(A)')
    READ(10,'(A)')
    READ(10,'(A)')
    DO
```

Sets a single observation experiment

Sets the observation location

SPEEDY has 96x48 grid points starting at the South Pole in Y and at Greenwich in X. It also has 7 sigma vertical levels.

5 possible variables U, V, T, Ps and q.

Once the location is set we must tell the 3D-VAR that the observational increment is going to be fixed. (edit the tdvar.f90 file)

```
PROGRAM tdvar
!=======================================================================
!
! [PURPOSE:] Main program of 3DVAR for the SPEEDY model
!
! [HISTORY:]
!   10/21/2004 Takemasa Miyoshi  created
!
!=======================================================================
  USE common
  USE common_speedy
  USE tdvar_tools
  USE minimizelib

  IMPLICIT NONE
  LOGICAL,PARAMETER :: msw_test=.TRUE.
  INTEGER,PARAMETER :: maxiter = 100
  REAL(r_size) :: u(nlon,nlat,nlev)
  REAL(r_size) :: v(nlon,nlat,nlev)
  REAL(r_size) :: t(nlon,nlat,nlev)
  REAL(r_size) :: q(nlon,nlat,nlev)
  REAL(r_size) :: ps(nlon,nlat)
```

Set the variable msw_test as .TRUE.

```
!!! For test
  IF (msw_test) THEN
    u = 1.0d0
    v = 1.0d0
    t = 1.0d0
    q = 1.0d-4
    ps = 1.0d2
  ELSE
    obs = obs - gues
    CALL reshape_gpv(obs,u,v,t,q,ps)
```

Set the value of the observational increments for each variable (only one of this variables is actually being observed, the one selected in ex_obs.f90)

Before running the experiment :

Edit the tdvar_response.sh script

```
#!/bin/sh
# for 3dvar cycle
set -e



#
export F_UFMTENDIAN=big
export GFORTRAN_CONVERT_UNIT="big_endian"
EXP=$1                                                      #Experiment name
#Compiler 1) ifort 2) gfortran 3) pg
compiler=1
TDVAR=/home/${USER}/SPEEDY_DA/tdvar                         #Script path
DATA=/home/${USER}/SPEEDY_DA/DAS_result/3dvar/${EXP}        #Output folder
ini=/home/${USER}/SPEEDY_DA/tdvar/initial             #Initial gues folder

OBSDIR=/home/${USER}/SPEEDY_DA/obs
GUESDIR=$DATA/gues
ANALDIR=$DATA/anal
ANALFDIR=$DATA/analf
mkdir -p $ANALFDIR
mkdir -p $ANALDIR
mkdir -p $GUESDIR

cp ${ini}/ctl/*.ctl ${DATA}/


cd $TDVAR
source timeinc.sh

# Initial date of analysis cycle
IYYYY=1982
IMM=01
IDD=01
IHH=00
# Final date of analysis cycle
EYYYY=1982
EMM=01
EDD=01
EHH=00
```

Compiler number:

1) Ifort 2)gfortran 3)PG

Use Ifort in the lab machines

To run the experiment :

./tdvar_response.sh name_of_the_experiment   (for example “response”).

You will see something like this:

```
>>>END BUILDING SPEEDY MODEL
>>>BEGIN COMPUTATION OF 1982/01/01/00

real    0m0.677s
user    0m0.621s
sys     0m0.054s
 TEST MODE (tdvar)
==========================================================================
|  N  | BG COST |OBS COST |   COST    | MAX(|GRAD|) | MAX(|CTRL|) | IFLAG |
--------------------------------------------------------------------------
     1  0.00E+00  5.00E-01  5.00E-01      9.85E-01      6.69E-01     1
     2  5.00E-01  1.11E-01  6.11E-01      1.13E+00      3.11E-01     1
     3  1.08E-01  4.99E-02  1.58E-01      2.23E-07      3.11E-01     0
==========================================================================

real    0m0.640s
user    0m0.252s
sys     0m0.063s
NORMAL END
```

The results will be available in DAS_result/3dvar/”name_of_the_experiment”

The files analysis.grd and gues.grd contains the analysis and first gues respectively and they can be opened with GrADS.

Plotting the results:

In DAS_result/3dvar/response you will find a GrADS script to plot the results (DA2008_response.gs)

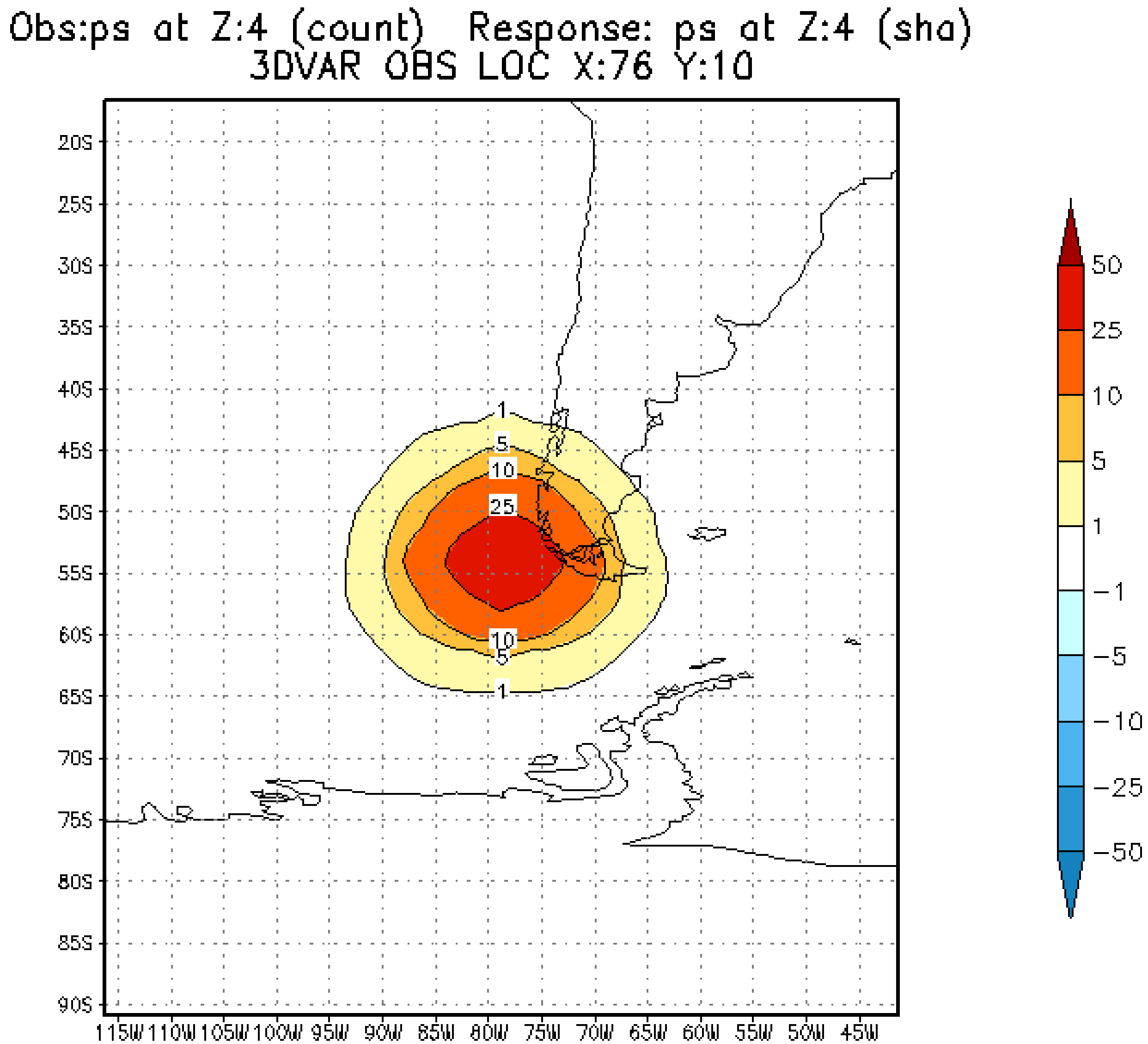
```
'reinit'

*Obs information
var_obs='ps'
lev_obs=4
x_obs=76
y_obs=24


*Response information.
var_response='ps'
lev_response=4


*The rest of the script
*Select the color scale
if(var_response=u | var_response = v )
levels_response='-1 -0.5 -0.25 -0.1 -0.05 0.05 0.1 0.25 0.5 1'
colors_response='49 47 45 43 41 0 21 23 25 27 29'
endif
if(var_response=ps )
levels_response='-50 -25 -10 -5 -1 1 5 10 25 50'
colors_response='49 47 45 43 41 0 21 23 25 27 29'
endif
```

Information about the observed variable (level, x and y location)

At which variable and level you want to plot the response

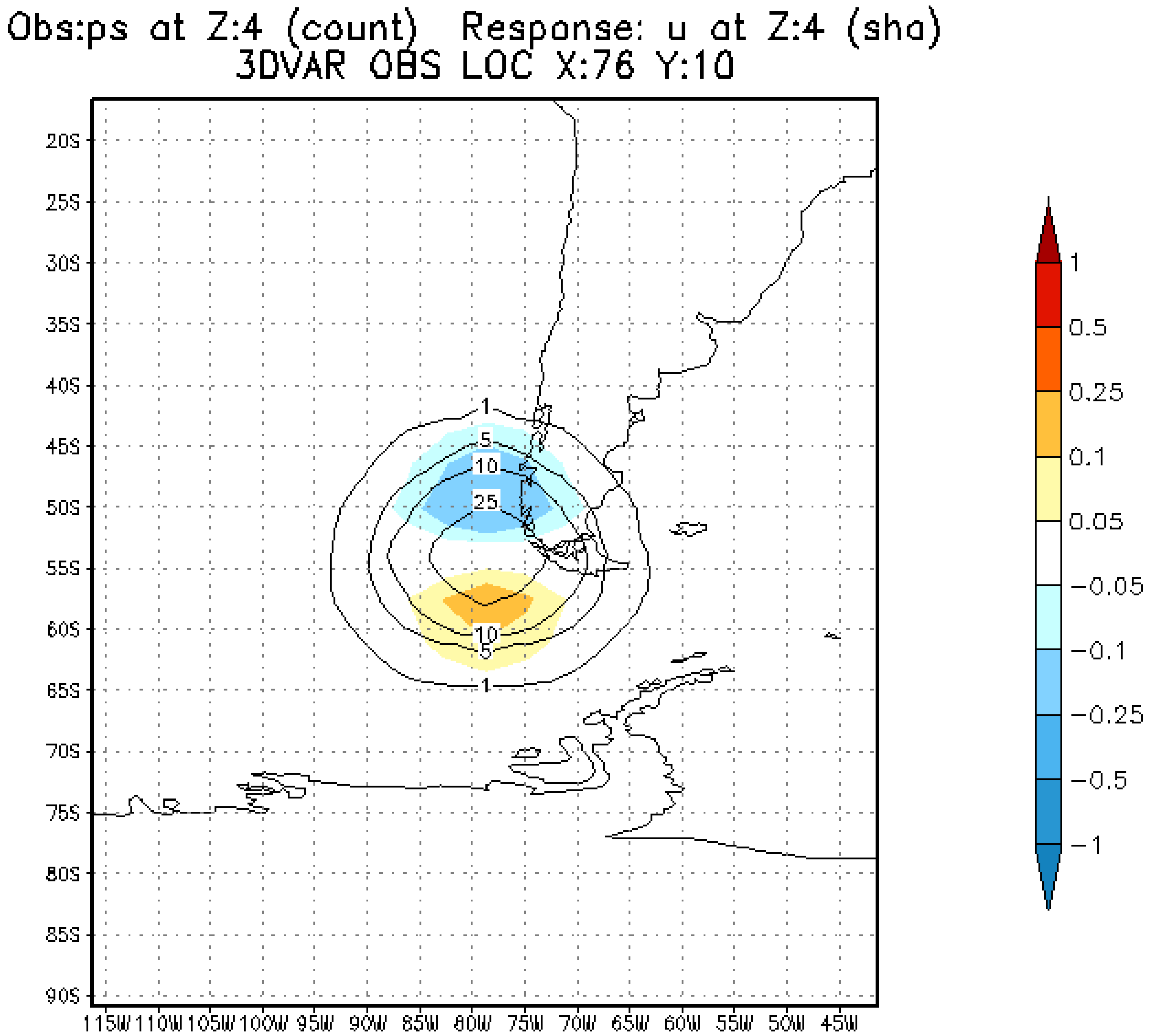
Results…

Response in PS for a single PS observation.

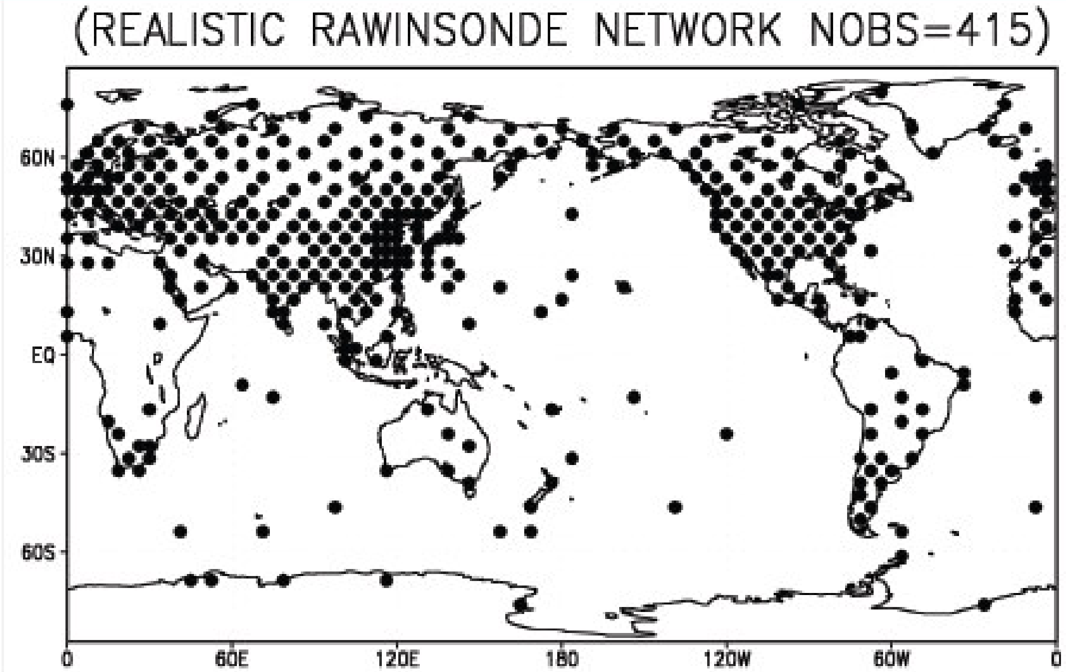


Results…

Response in U (at Z=4) for the same single PS observation.



Run 3D-Var of rawinsonde observation network



Edit ex_obs.f90 in the tdvar folder

```
PROGRAM ex_obs
  IMPLICIT NONE

  LOGICAL,PARAMETER :: msw_test=.FALSE.
  LOGICAL,PARAMETER :: msw_real=.TRUE.
  LOGICAL,PARAMETER :: msw_dnsobs=.FALSE.
  INTEGER,PARAMETER :: nlon=96
  INTEGER,PARAMETER :: nlat=48
  INTEGER,PARAMETER :: nlev=7

  INTEGER :: ilon,ilat,ios
  CHARACTER(10) :: ctmp1,ctmp2
  INTEGER :: i,j,k

  INTEGER :: ex_u(nlon,nlat,nlev) = 0
  INTEGER :: ex_v(nlon,nlat,nlev) = 0
  INTEGER :: ex_t(nlon,nlat,nlev) = 0
  INTEGER :: ex_q(nlon,nlat,nlev) = 0
  INTEGER :: ex_ps(nlon,nlat) = 0
  INTEGER :: ex_prec(nlon,nlat) = 0

  IF (msw_test) THEN
    ex_u(76,35,4) = 1
  ELSE IF (msw_real) THEN
    OPEN(10,file='obsmark.gs')
    READ(10,'(A)')
    READ(10,'(A)')
    READ(10,'(A)')
    READ(10,'(A)')
    READ(10,'(A)')
    READ(10,'(A)')
    DO
      READ(10,'(A10,I2,A1,I2)',IOSTAT=ios) ctmp1,ilon,ctmp2,ilat
      IF( ios /= 0 ) EXIT
!      PRINT *,ilon,ilat,ios
      ex_u(ilon,ilat,1:nlev) = 1
      ex_v(ilon,ilat,1:nlev) = 1
!      ex_t(ilon,ilat,1:nlev) = 1
!      ex_q(ilon,ilat,1:nlev) = 1
      ex_ps(ilon,ilat) = 1
    END DO
    CLOSE(10)
```

Set msw_test as .FALSE. and msw_real as .TRUE.

Choose the variables that are going to be observed by uncommenting the lines corresponding to each variable

Set the msw_test parameter as .FALSE. In the tdvar.f90 program

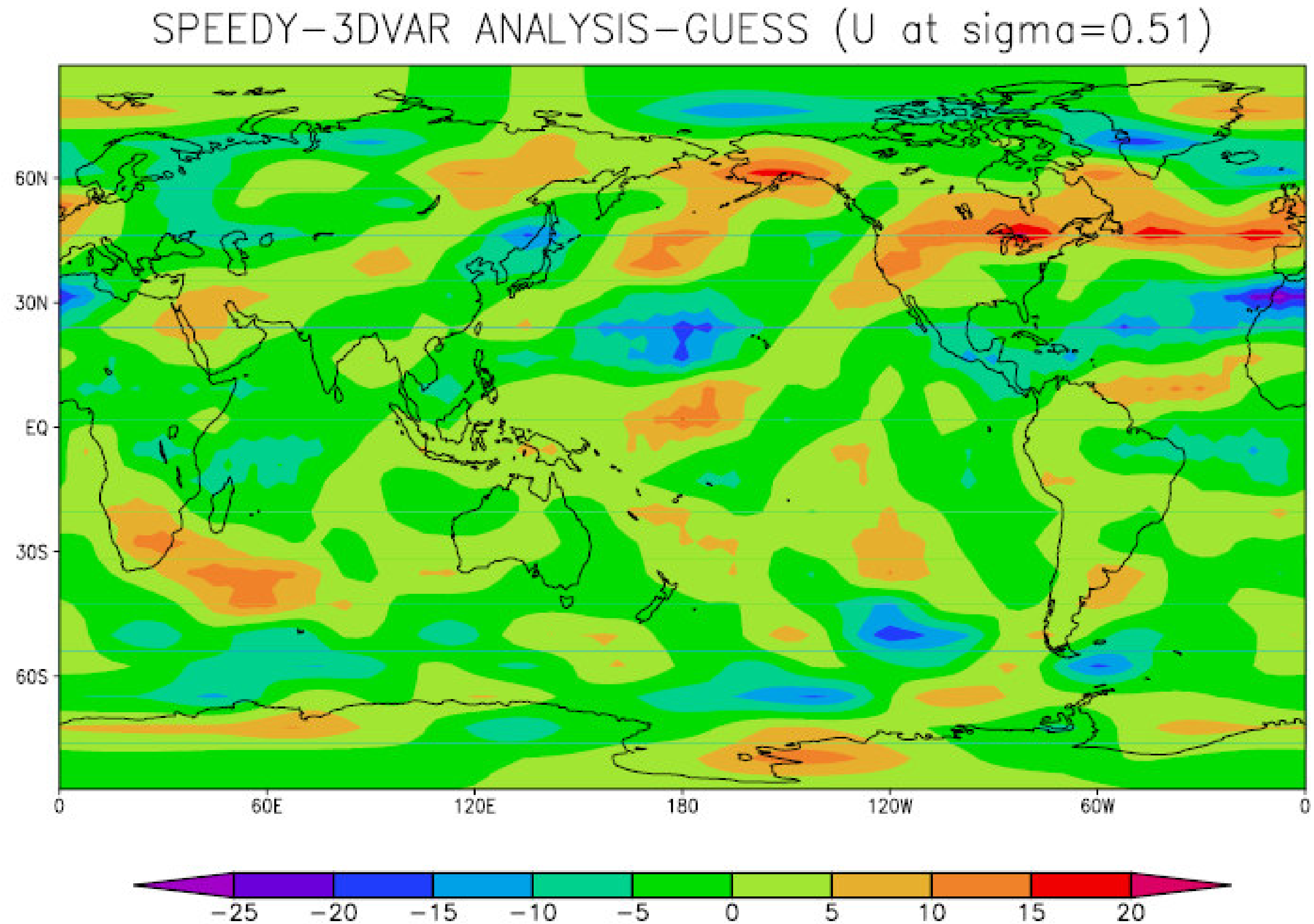
You can run the tdvar_response.sh to do only one assimilation or run the tdvar.sh script to start and assimilation cycle.

To run an assimilation cycle type:

./tdvar.sh exp_name

Where exp_name is the experiment name. The results will be stored in a folder with the name of the experiment under DAS_results/3dvar.

In the tdvar.sh script there is a variable “STORE” that controls the amount of output generated by the assimilation cycle. If it set to 0, then only the first gues and the analysis in pressure levels will be stored, else first gues, analysis and filtered analysis will be stored in sigma and pressure levels. (do not set store = 1 in the computers of the lab since we don´t have enough storage capacity)



Miyoshi (2005)

References

- Barker, D. M., W. Huang, Y.-R. Guo, A. J. Bourgeois, and Q. N. Xiao, 2004: A Three-Dimensional Variational Data assimilation system for MM5: Implementation and initial results. *Mon. Wea. Rev.*, **132**, 897-914.

- Kalnay, E., 2003: Atmospheric modeling, data assimilation and predictability. Cambridge University Press, 341pp

- Miyoshi,T., 2005: Ensemble Kalman filter experiments with a primitive-equation global model, thesis in University of Maryland

- Parrish, D. F. and J. C. Derber, 1992: The National Meteorological Center's Spectral Statistical-Interpolation Analysis system. *Mon. Wea. Rev*. **120,** 1747-1763