Intensive Course on Data Assimilation
Buenos Aires, Argentina
27 October - 7 November 2008

SPEEDY DA (LETKF)

Prepared by Junjie Liu, Takemasa Miyoshi and Juan Ruiz

Buenos Aires, Argentina

27 October – 7 November 2008

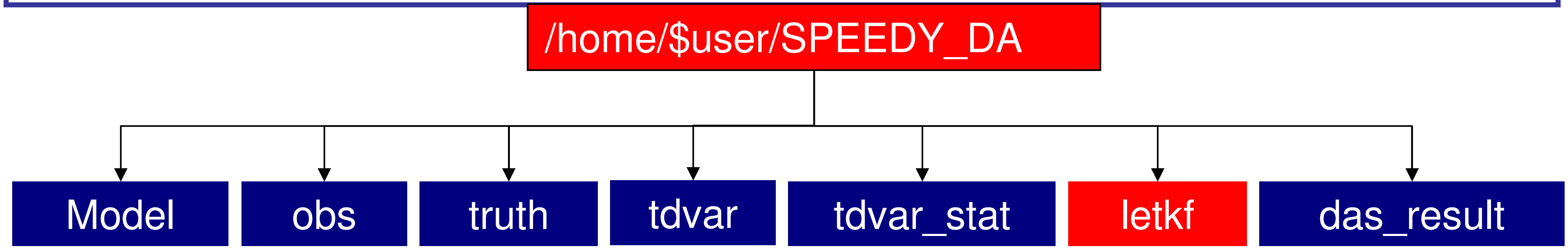
What will we learn in this lesson:

•LETKF implementation formula.

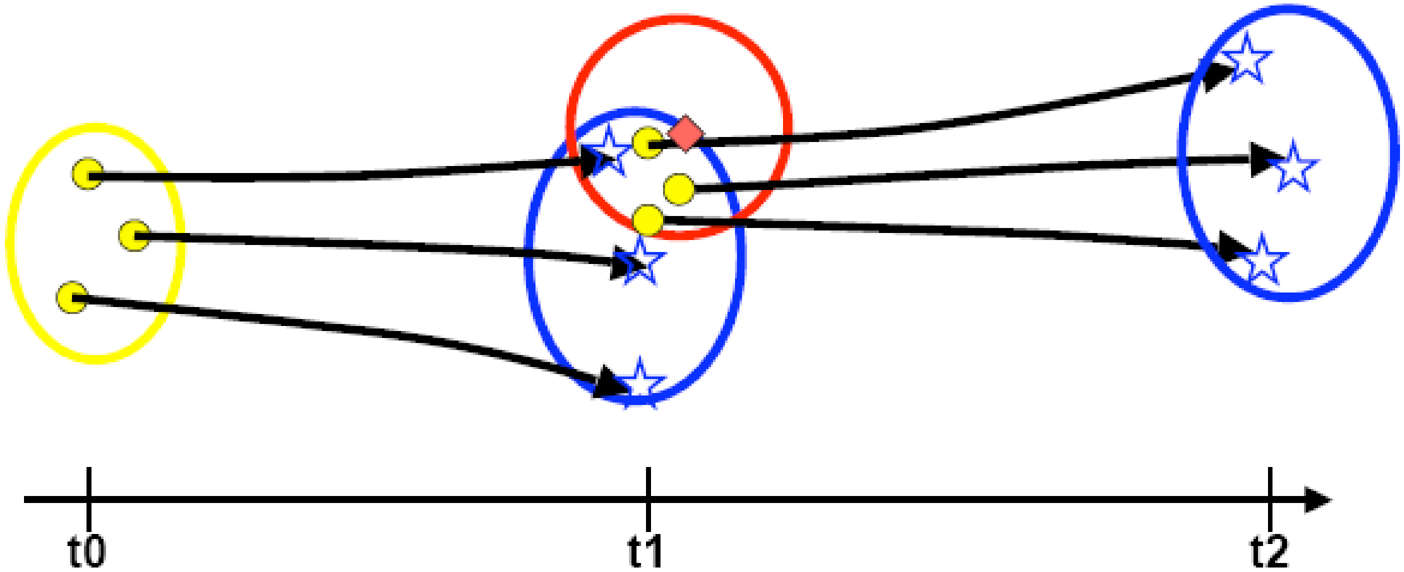
•Some LETKF parameters.

•Response test with LETKF (assimilating only one observation)

•Run LETKF for a realistic rawindsonde observation network



Local Ensemble Transform Kalman Filter
(LETKF, Ott et al., 2004, Hunt et al., 2007)



✓ Yellow: analysis ensemble and its uncertainty; Blue: background ensemble and its uncertainty; Red: observation and its uncertainty

✓ LETKF, like any other EnKF, provides background and analysis uncertainty estimation in every analysis cycle.

Liu (2007)

ETKF equations: Hunt et. al. 2007

$$\overline{\mathbf{w}}^a = \tilde{\mathbf{P}}^a (\mathbf{Y}^b)^T \mathbf{R}^{-1} (\mathbf{y}^o - \overline{\mathbf{y}}^b),$$

$$\tilde{\mathbf{P}}^a = [(k-1)\mathbf{I} + (\mathbf{Y}^b)^T \mathbf{R}^{-1} \mathbf{Y}^b]^{-1}.$$

$$\mathbf{X}^b = \left[\mathbf{x}_1^b - \overline{\mathbf{x}}^b \mid \ldots \mid \mathbf{x}_K^b - \overline{\mathbf{x}}^b\right];$$

$$\mathbf{y}_i^b = H(\mathbf{x}_i^b);\ \mathbf{Y}_n^b = \left[\mathbf{y}_1^b - \overline{\mathbf{y}}^b \mid \ldots \mid \mathbf{y}_K^b - \overline{\mathbf{y}}^b\right]$$

k is the number of ensemble members.

$$\overline{\mathbf{x}}^a = \overline{\mathbf{x}}^b + \mathbf{X}^b \overline{\mathbf{w}}^a$$

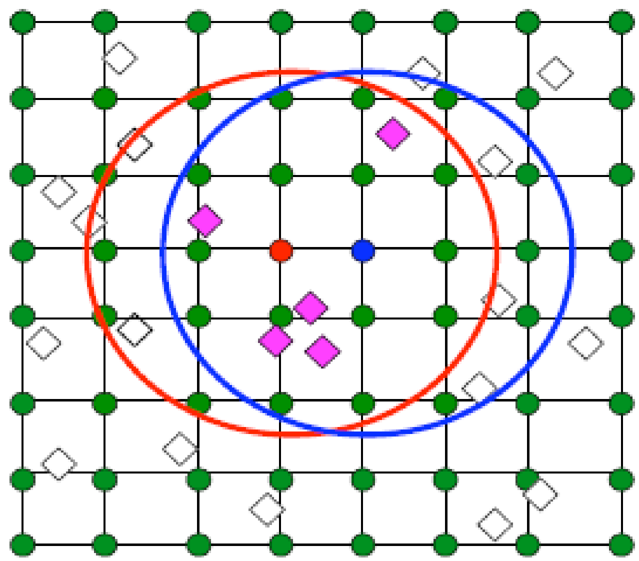
The analysis is a weighed average of the background ensemble members.

$$\mathbf{W}^a = [(k-1)\tilde{\mathbf{P}}^a]^{1/2} \qquad \mathbf{X}^a = \mathbf{X}^b \mathbf{W}^a$$

X^a, X^b are n*k matrices where n is the number of model variables.

Local Ensemble Transform Kalman Filter
(LETKF, Ott et al., 2004, Hunt et al., 2007)

Schematic of 2-dimension local patch

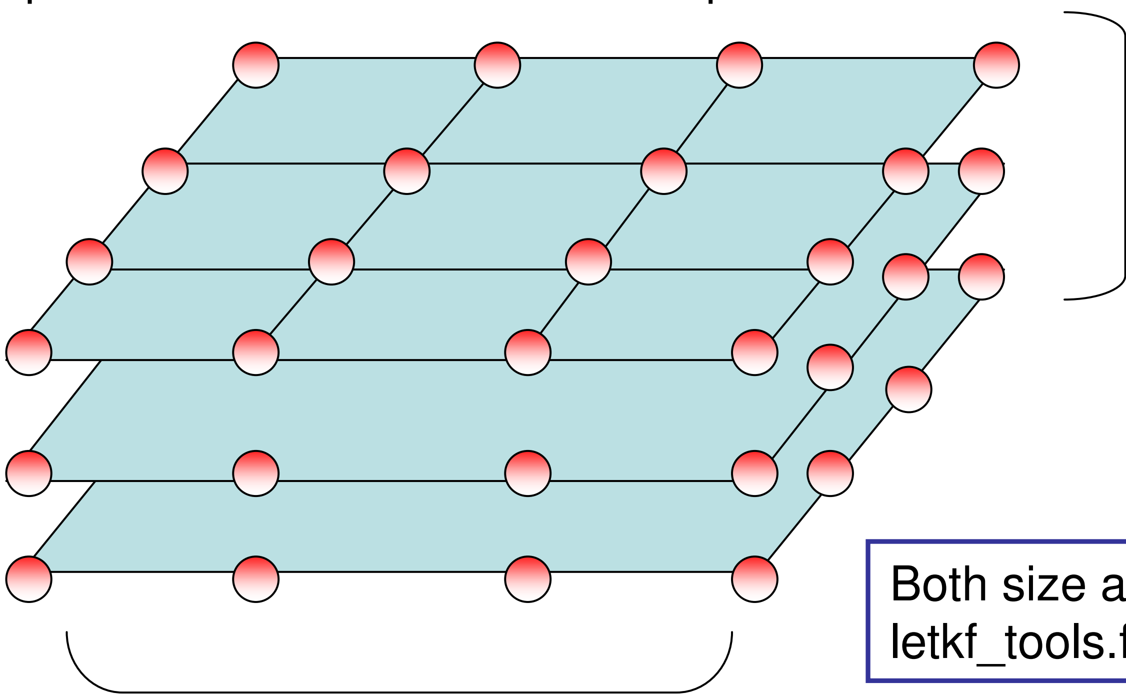


- Different local volumes have a great overlap.
- Each observation is used more than once in the data assimilation.
- The analysis in each grid point is highly parallel.

Liu (2007)

Analysis localization in the SPEEDY model.

The shape of the localization boxes is square.



Both size are controled in the letkf_tools.f90 file.

```
INTEGER,PARAMETER :: nlocal=4          ! horizontal local patch size
INTEGER,PARAMETER :: nlocal2=4         ! size of the patch that will be added to the analysis.
INTEGER,PARAMETER :: nlocalv=1         ! vertical local patch size, 0 means doint assimilatin at each vertical levels
INTEGER,PARAMETER :: nskip_h=2        !J 1=no skip, 2 skip 1 point at a time
LOGICAL,PARAMETER :: test=.TRUE.
```

nlocal=4 means a total horizontal size of 9 grid points.

Error covariance inflation:

To avoid filter divergence the background error covariance or the analysis error covariance can be “inflated” by different methods.

In this case a multiplicative scheme with fixed parameter is used to inflate the background error covariance.

$$X^{i}{}_{b} = (x^{i}{}_{b} - \bar{x}_{b})\sqrt{(1 + parameter)}$$

The paremeter inflation factor has to be tuned for each particular system. The value for the SPEEDY model is set in the file letkf_tools.f90

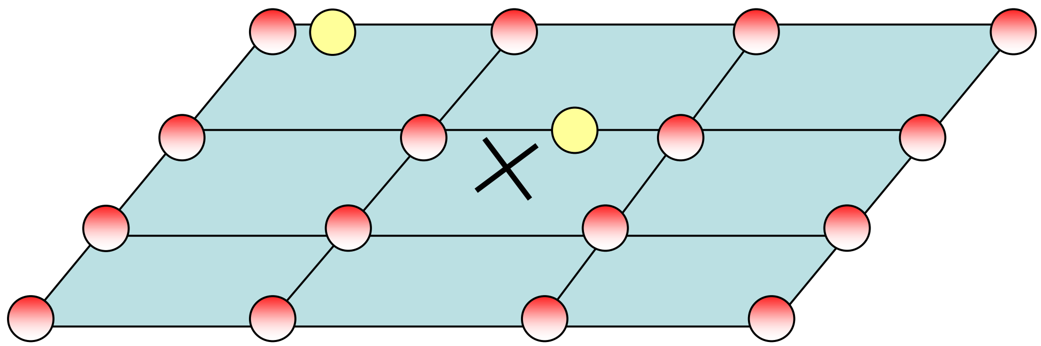
```
IMPLICIT NONE

PRIVATE
PUBLIC :: set_letkf_tools,das_letkf

INTEGER,PARAMETER :: nlocal=4           ! horizontal local patch size
INTEGER,PARAMETER :: nlocal2=4          ! size of the patch that will be added to th
INTEGER,PARAMETER :: nlocalv=1          ! vertical local patch size, 0 means doint a
INTEGER,PARAMETER :: nskip_h=2          !J 1=no skip, 2 skip 1 point at a time
LOGICAL,PARAMETER :: test=.TRUE.
INTEGER,SAVE :: ndiml_h
REAL(r_size),SAVE :: parm_inflation=0.05d0    ! inflation factor
INTEGER,PARAMETER :: msw_inflation = 0
! mode switch msw_inflation: inflation type
! 0: multiplicative with fixed parameter
INTEGER,PARAMETER :: msw_localize =  2
! mode switch msw_localize: covariance localization type
! 0: no localization
! 1: localization by B : does not work
! 2: localization by R
REAL(r_size),PARAMETER :: sigma_obs = 2.5d0 !observational localization scale
REAL(r_size),PARAMETER :: tref = 273.0d0
REAL(r_size),PARAMETER :: pref = 1000.0d2
REAL(r_size),SAVE :: uorder(nlev),vorder(nlev),torder(nlev),qorder(nlev),psorder
```

The optimun value for the SPEEDY model is 0.05

Localization by R

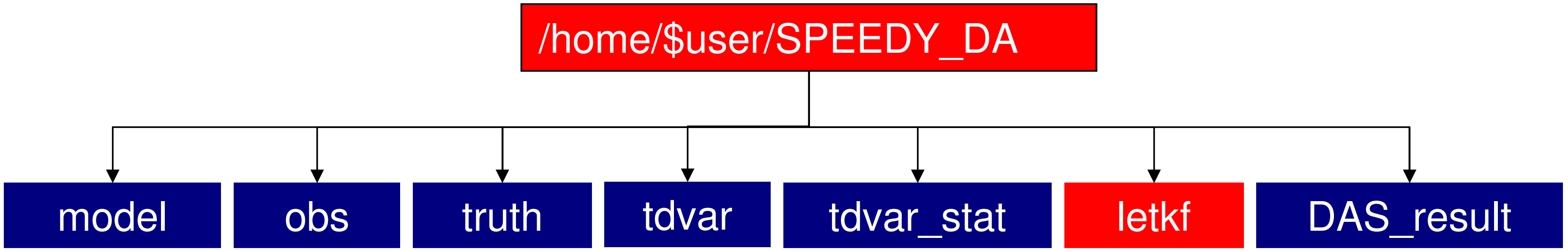
Inside the local domain the observations that are closer to the domain center will have greater impact than that which are near the domain border.

To achieve this the observation error of every observation within the local domain is increased by a factor which depends on the inverse of the distance to the domain center.

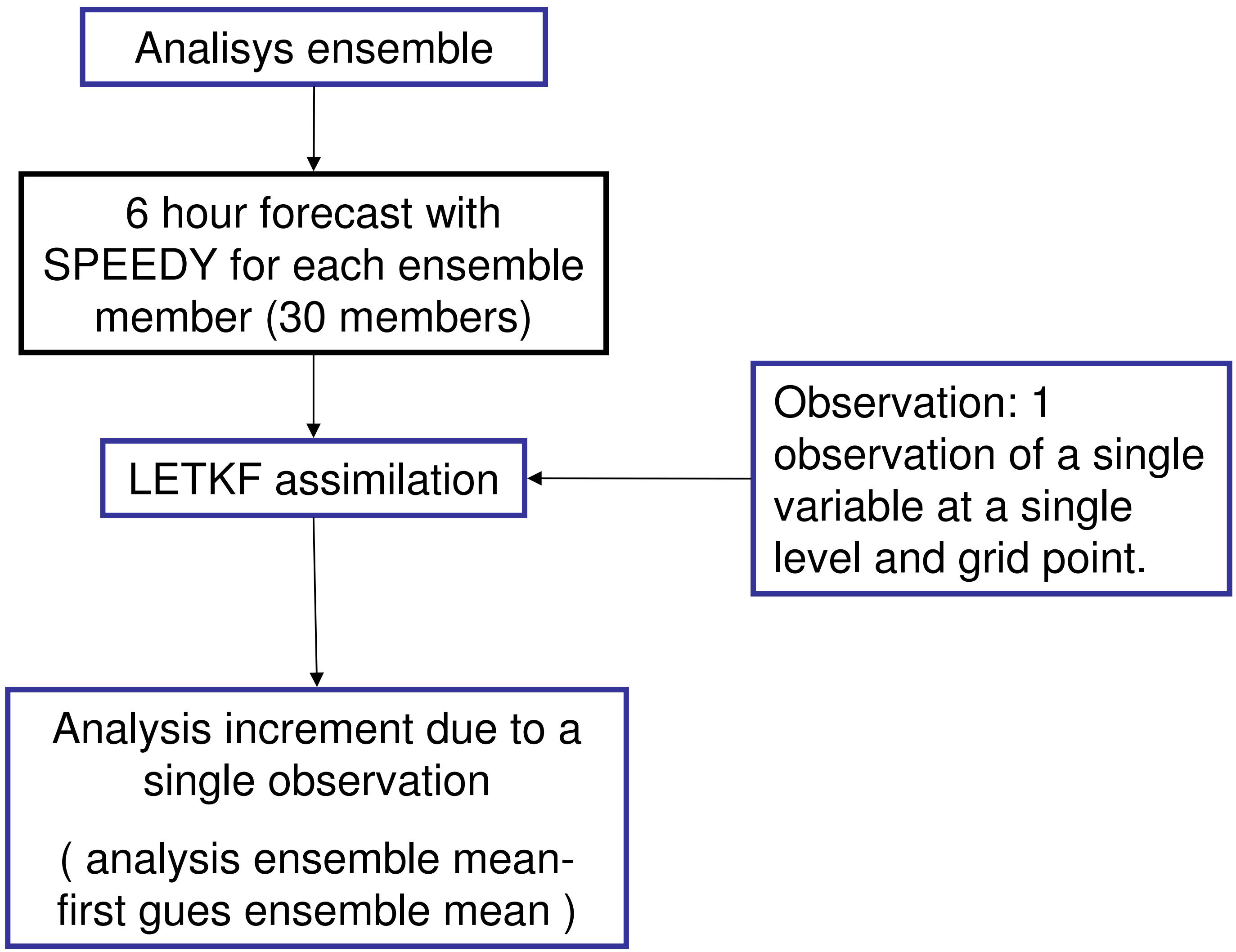


This helps to avoid the undesirable consequences of sampling errors, reducing the influence of observations that are far away from the patch center. This is especially important in the situation where we have sparce observations.

LETKF Response to a single observation



Response experiment:



Observation location: (as in 3D-VAR experiment)

This is controled by the ex_obs.f90 file.

```fortran
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 LETKF that the observational increment is going to be fixed. (edit the letkf_tools.f90 file)

```
MODULE letkf_tools
!
! Module for LETKF on the SPEEDY model
!
  USE common
  USE common_speedy
  USE common_letkf

  IMPLICIT NONE

  PRIVATE
  PUBLIC :: set_letkf_tools,das_letkf

  INTEGER,PARAMETER :: nlocal=4          ! horizontal local patch size
  INTEGER,PARAMETER :: nlocal2=4         ! size of the patch that will be added to the analysis.
  INTEGER,PARAMETER :: nlocalv=1         ! vertical local patch size, 0 means doint assimilatin at each vertical leve
  INTEGER,PARAMETER :: nskip_h=2        !J 1=no skip, 2 skip 1 point at a time
  LOGICAL,PARAMETER :: test=.TRUE.
  INTEGER,SAVE :: ndiml_h
  REAL(r_size),SAVE :: parm_inflation=0.05d0   ! Inflation factor
  INTEGER,PARAMETER :: msw_inflation = 0
  ! mode switch msw_inflation: inflation type
  ! 0: multiplicative with fixed parameter
  INTEGER,PARAMETER :: msw_localize =  2
```

Set the test variable as .TRUE.

Set the “compiler” variable in the scripts letkf_response.sh (for the PC in the lab use the Intel option).

Run the letkf_response.sh:

➢ ./letkf_response.sh response

```
>>>END BUILDING SPEEDY MODEL
>>>BEGIN COMPUTATION OF 1982/01/01/00
finish forecast of ensemble number, 01
finish forecast of ensemble number, 02
finish forecast of ensemble number, 03
finish forecast of ensemble number, 04
finish forecast of ensemble number, 05
finish forecast of ensemble number, 06
finish forecast of ensemble number, 07
finish forecast of ensemble number, 08
finish forecast of ensemble number, 09
finish forecast of ensemble number, 10
finish forecast of ensemble number, 11
finish forecast of ensemble number, 12
finish forecast of ensemble number, 13
finish forecast of ensemble number, 14
finish forecast of ensemble number, 15
finish forecast of ensemble number, 16
finish forecast of ensemble number, 17
finish forecast of ensemble number, 18
finish forecast of ensemble number, 19
finish forecast of ensemble number, 20
finish forecast of ensemble number, 21
finish forecast of ensemble number, 22
finish forecast of ensemble number, 23
finish forecast of ensemble number, 24
finish forecast of ensemble number, 25
finish forecast of ensemble number, 26
finish forecast of ensemble number, 27
finish forecast of ensemble number, 28
finish forecast of ensemble number, 29
finish forecast of ensemble number, 30
calculate mean state and spread for
1982010100
calculate mean state and spread for
1982010106
 LETKF assimilation...

real    3m34.064s
user    3m33.457s
sys     0m0.275s
NORMAL END
[juan@localhost letkf]$
```

6 hour ensemble forecast using SPEEDY

LETKF assimilation

The results will be available in DAS_result/letkf/response

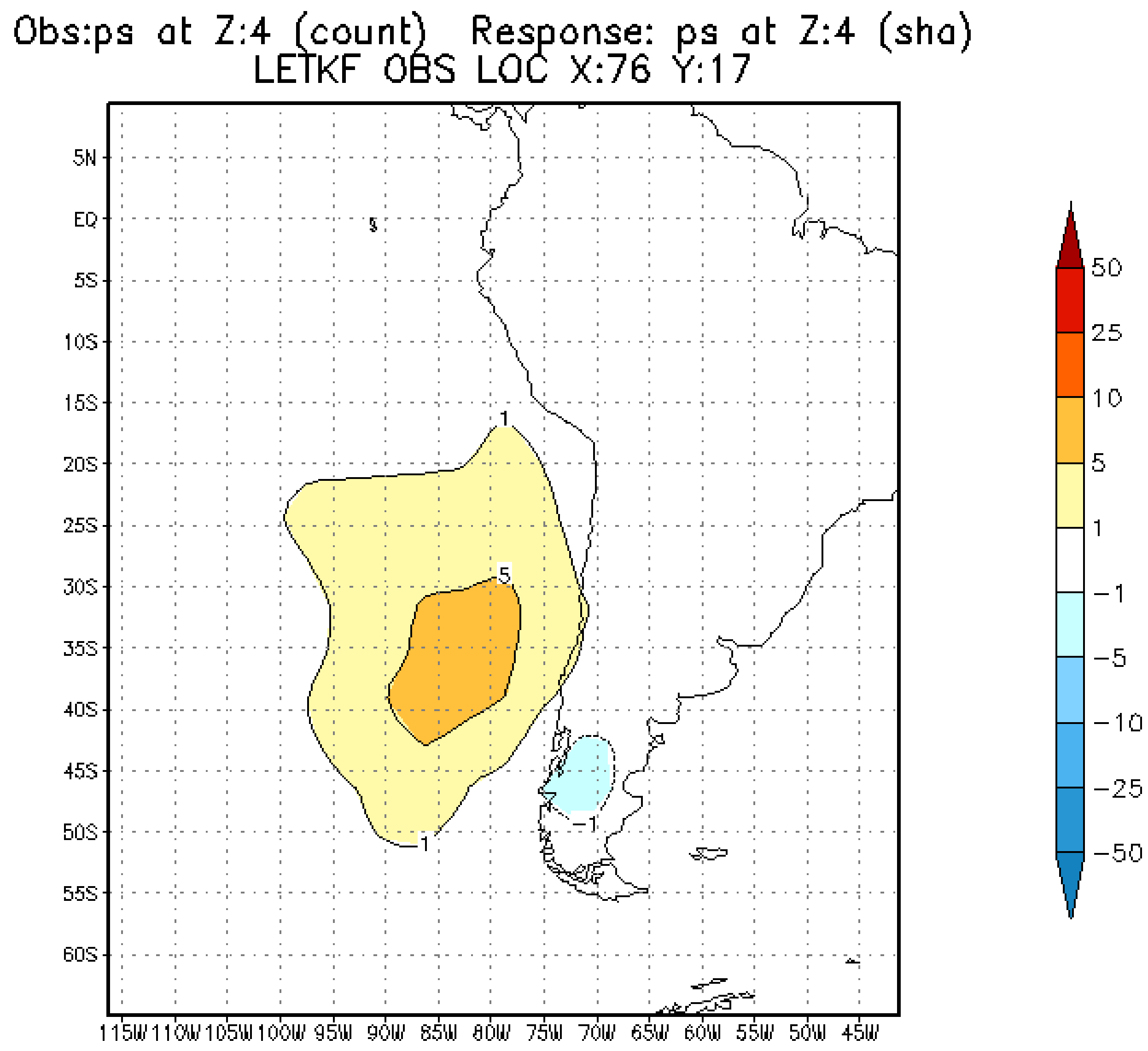
The files analysis.grd and gues.grd contains the analysis ensemble mean and first gues ensemble mean respectively and they can be opened with GrADS.

In this case you will also have a gues_spread.grd file which contains the first gues ensemble spread data and also can be opened with GrADS.

In the folder DAS_result/letkf/response you will also find a grads script (DA2008_response.gs) to plot the results.

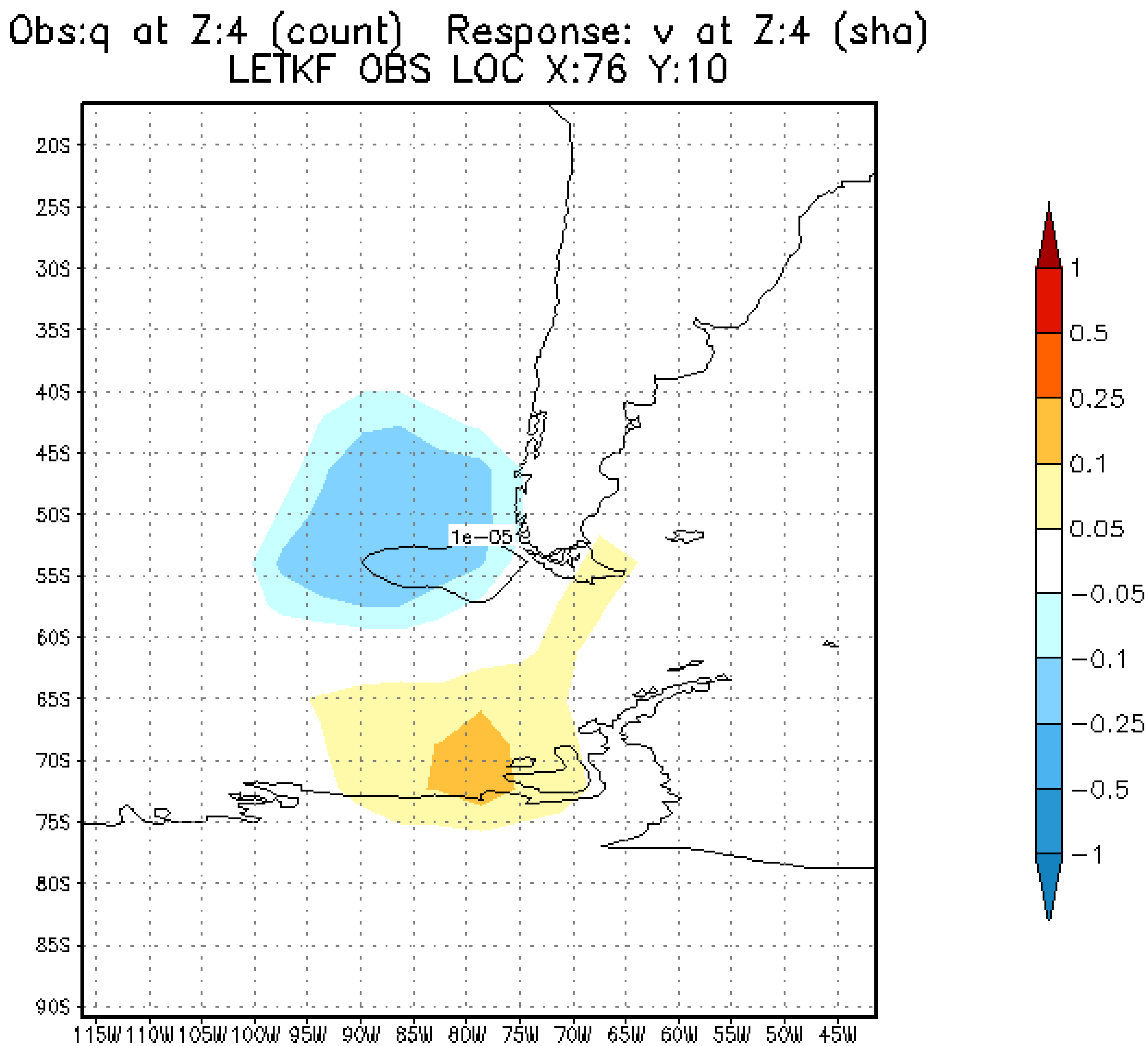
Results…

Response in PS for a single PS observation.

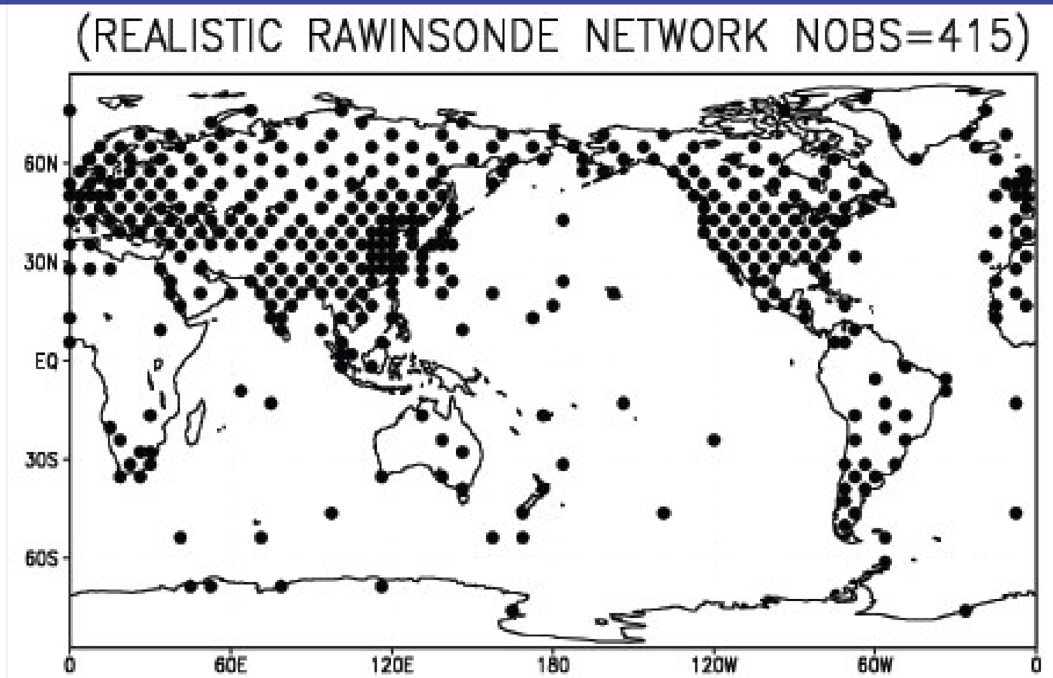


Results…

Response in V (z=4) for a single q(z=4) observation.



Run LETKF of rawinsonde observation network



Edit ex_obs.f90 in the letkf folder

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

Set the test parameter as .FALSE. In the letkf_tools.f90 program

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

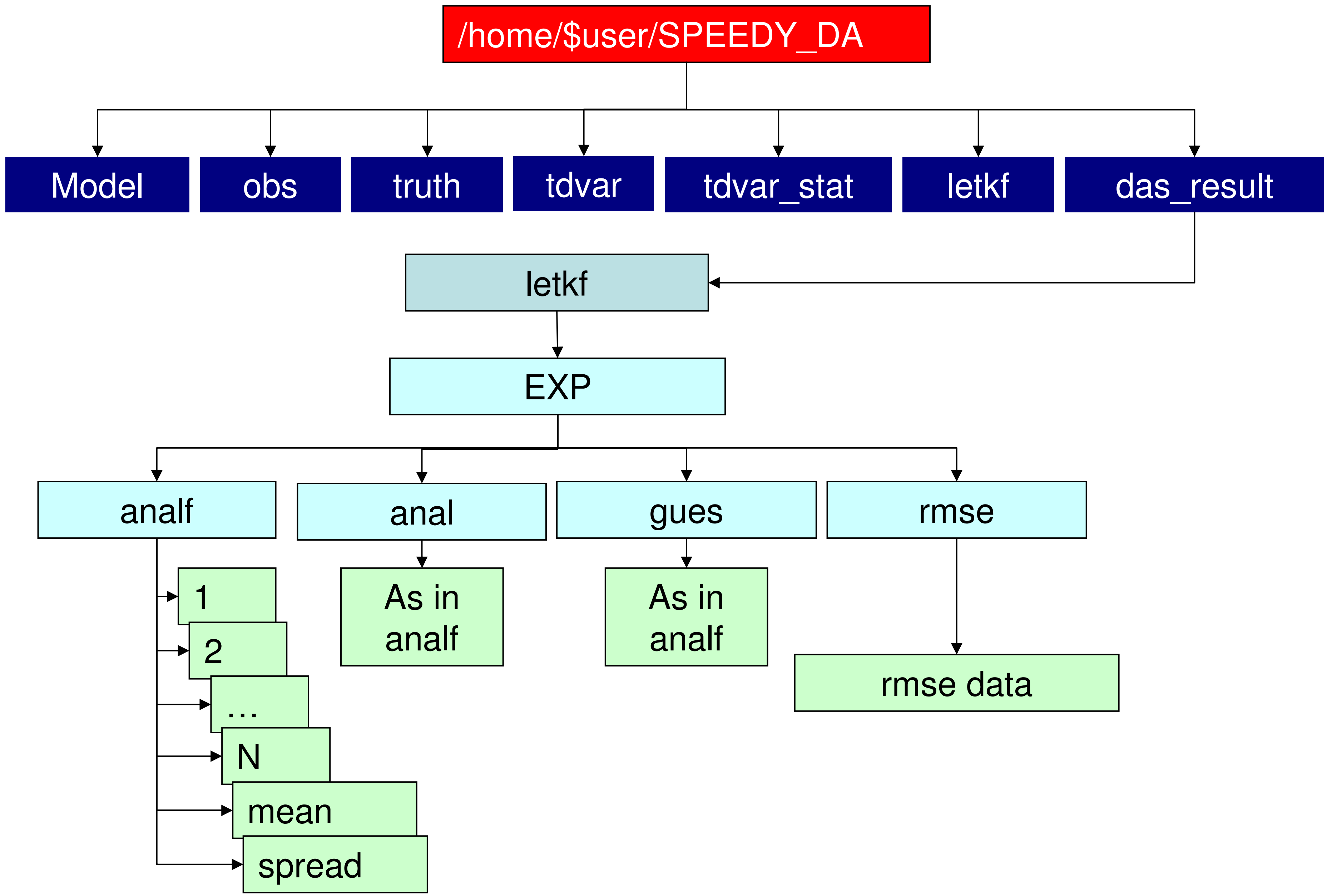
To run an assimilation cycle type:

./letkf.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/letkf.

In the letkf.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 mean and the analysis mean in pressure levels will be stored, else first gues ensemble members, analysis and filtered analysis ensemble members will be stored in sigma and pressure levels. (do not set store = 1 in the lab computers since we don´t have enough storage capacity)

Directory description:



Zonally averaged RMSE computation:

After the completion of a DA experiment you can use the rmse_zm.sh and rmse_energy.sh scripts to compute the zonally averaged rmse for each variable and for each time.

Under “letkf” type:

- rmse_zm.sh “EXP_NAME”

Where “EXP_NAME” is the name of the experiment. This will create a folder inside the experiment results folder with the name “rmse”, inside this folder a GrADS ctl file and a data file will be created.

As in the previous example the script rmse_energy.sh will compute the RMSE for the total energy.

The time control of the RMSE computation (to change the start and end date for the RMSE computation you will have to edit the rmse_zm.f90 and rmse_energy.f90 files.

The rmse_zm.sh and rmse_energy.sh scripts under “tdvar” computes the RMSE for the 3DVAR experiments.

Data Assimilation Cycle Intercomparisson

To compare the results obtained with different configurations you can use two grads scripts located in /home/user/SPEEDY_DA/DAS_results/letkf/gs and in

/home/user/SPEEDY_DA/DAS_results/tdvar/gs

- DA2008_ave.gs Computes the time average of the zonally averaged RMSE for a particular variable at a particular level. Can be used to compare multiple experiments.
- DA2008_timec.gs Shows the temporal evolution of the zonally averaged RMSE at a particular level, latitude and for a particular variable.

```
*WHITE
exp1='EXP_FULL'
*RED
exp2='EXP_NOT'
*GREEN
exp3='EXP_NOUV'
*BLUE
exp4='EXP_NOQ'
*YELLOW
exp5='EXP_SUPUPPER'
*BROWN
exp6='EXP_NOERRSTAT'
```

Select the experimets that you´d like to compare

```
var='t'
level=4
range1=0
range2=8
```

Select the variable (U, V, T, PS, Q) and the level (1 to 7), and the Y axis start and end values.

```
'open  ../'exp1'/rmse/rmse_zm.ctl'
'open  ../'exp2'/rmse/rmse_zm.ctl'
'open  ../'exp3'/rmse/rmse_zm.ctl'
'open  ../'exp4'/rmse/rmse_zm.ctl'
'open  ../'exp5'/rmse/rmse_zm.ctl'
'open  ../'exp6'/rmse/rmse_zm.ctl'



'set z 'level
'set grads off'
'set t 1 230'
'set lat -90 90'
```

The rest of the script opens the selected experiments, compute and plots the time average.

```
reinit
* Draw a line plot comparing the RMSE evolution for different experiments at a fixed lattitude.

*WHITE
exp1='EXP_FULL'
*RED
exp2='EXP_NOT'
*GREEN
exp3='EXP_NOUV'
*BLUE
exp4='EXP_NOQ'
*YELLOW
exp5='EXP_SUPUPPER'
*BROWN
exp6='EXP_NOERRSTAT'

var='u'
level=4
lat=20
range1=0
range2=15

'open  ../'exp1'/rmse/rmse_zm.ctl'
'open  ../'exp2'/rmse/rmse_zm.ctl'
'open  ../'exp3'/rmse/rmse_zm.ctl'
'open  ../'exp4'/rmse/rmse_zm.ctl'
'open  ../'exp5'/rmse/rmse_zm.ctl'
'open  ../'exp6'/rmse/rmse_zm.ctl'

'set z 'level
'set grads off'
'set t 1 230'
'set lat 'lat
```

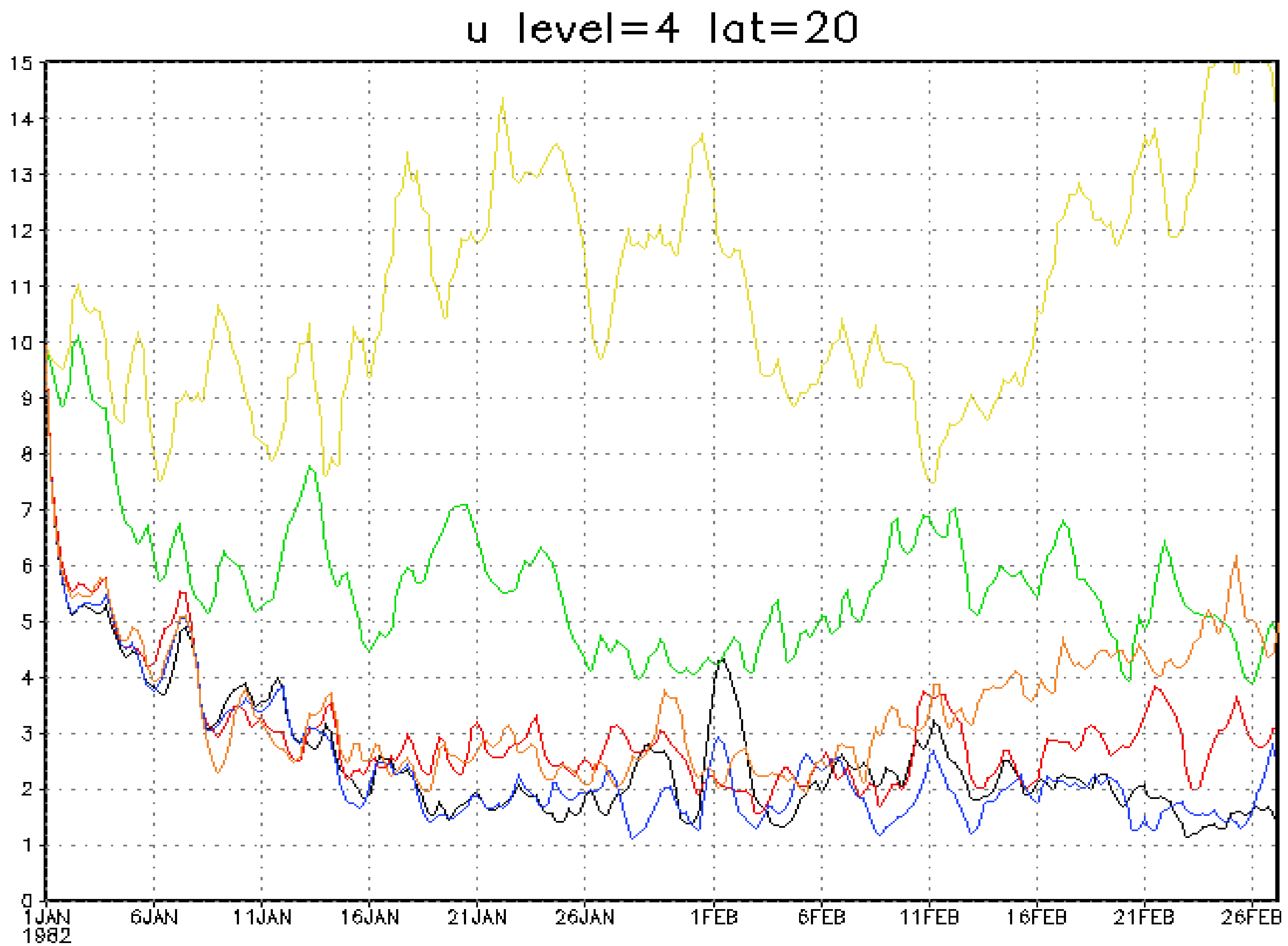
Chose the experiments.

Chose the desidered variable, level, and latitude.

Set the Y axis range.

Example 3DVAR: U at level 4 and latitude 20 N.

EXP_FULL (black), EXP_NOT (red), EXP_NOUV (green), EXP_NOQ (blue), EXP_SUPUPPER (yellow), EXP_NOERRSTAT (brown)



Example 3DVAR: T at level 4.

EXP_FULL (black), EXP_NOT (red), EXP_NOUV (green), EXP_NOQ (blue), EXP_SUPUPPER (yellow), EXP_NOERRSTAT (brown)