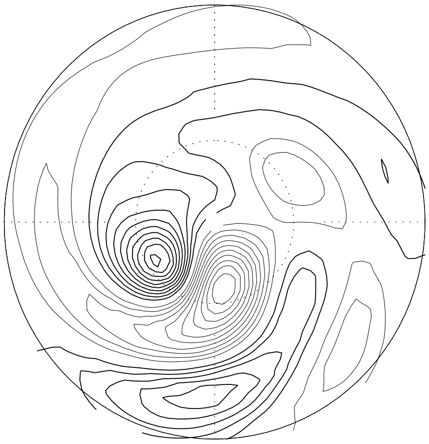
$$
\begin{matrix}
B & M & 0 & 0 & 0 & 0 & 0 & 0 \\
L & B & M & 0 & 0 & 0 & 0 & 0 \\
0 & L & B & M & 0 & 0 & 0 & 0 \\
0 & 0 & L & B & M & 0 & 0 & 0 \\
0 & 0 & 0 & L & B & M & 0 & 0 \\
0 & 0 & 0 & 0 & L & B & M & 0 \\
0 & 0 & 0 & 0 & 0 & L & B & M \\
0 & 0 & 0 & 0 & 0 & 0 & L & B
\end{matrix}
$$



Linear Baroclinic Model (LBM) Package

Users' Guide

Version 2.2
31 August, 2005

Masahiro Watanabe
Faculty of Environmental Earth Science,
Hokkaido University, Japan
hiro@ees.hokudai.ac.jp

Contents

1 About this package

1.1 Purpose and application

This numerical package has been built up in order to examine a linear dynamics in the atmosphere, such as to compute steady linear response to a prescribed forcing, eigenanalysis, and so on.

Fully nonlinear climate models, like atmospheric general circulation models (AGCMs), are recently improved more and more realistic, indicating that they can be used to simulate and understand current climate which involves a number of feedback processes. However, such models are enough complicated, perhaps as well as in nature, making the interpretation — what is going on in the model — difficult without carring out thorough analyses of the outputs.

The linearized atmospheric model contained in this package was aimed to help understanding the complicated sequence of feedback in the dynamical atmosphere, by removing nonlinearity in their processes. The dynamical framework is simplified in this model, so that the results would be much easily interpreted. Besides, the package supplies several techniques to deal with linear dynamics, which may be useful to investigate various aspects of the large–scale atmospheric circulation. On this regard, the linear model is expected to be a powerful tool for climate, in particular meteorological, studies. Models and diagnostic techniques provided by this package include;

- staedy linear response to prescribed forcing (cf. section 3), solved with
  - time integration of the linear model (cf. section 3.1)
  - direct method for full matrix (cf. section 3.2)
  - direct method as a stationary wave model (cf. section 3.3)
  - iterative method for full matrix (cf. section 3.4)
- storm track model (cf. section 4)
- eigenmode and singular mode computation (cf. section 5)
- time integration of the nonlinear dynamical model (cf. section 6)
- moist linear model (cf. section 7)
- barotropic model (cf. section 8)

Note that the latter three of the above application have been available in the version 2.0 while the iterative method in section 3 first appeas in the present version.

1.2 Mathematical basis of the linear model

This document is primarily directed to explain how to use this numerical package, so that the mathematical and/or physical aspects of the linear model is not fully described (brief description is given in Appendix). The user may need to refer to appropriate research articles, most of which are cited in this document, for details of the model physics.

1.3 About the numerics

The package is archived into three compressed tar files:

- `ln_solver2.2.tar.gz` (about 103MB)
- `ln_solver2.2.ncepdata.tar.gz` (about 45MB)
- `ln_solver2.2.ecmdata.tar.gz` (about 133MB)

The first archive contains all the model code while the latter two are the NCEP/NCAR and ECMWF (ERA40) reanalysis data to make basic states (see section 2.4).

Contents in the main package (`ln_solver2.2.tar.gz`) is further divided into two components; dynamical model and matrix solver. All the programs contained is written in Fortran, and the model part has been constructed based on a dynamical core of AGCM cooperatively developed at Center for Climate System Research (CCSR), University of Tokyo, and National Institute for Environmental Studies (NIES), called CCSR/NIES AGCM, version 5.4g. The spectral representation of primitive equations with a vertical sigma coordinate has been explicitly linearized about a basic state. Since the model is used, in most cases, with multiple levels although a single-layer model (barotropic model) is available, it is hereafter referred to as the **linear baroclinic model (LBM)** in this document. The form of linear equations are given by Watanabe and Kimoto (2000, 2001) and also briefly described in the Appendix of this manual.

Linear solutions for the steady LBM are computed in the second component of the package, i.e., matrix solver part. The matrix solver needs help of a freely distributed mathematical library, LAPACK. To increase the portability of this package, LAPACK library for several architectures has been included. The systems on which the package has been tested are:

- Sun OS 4.2–5.8 (Solaris)
- DEC (alpha)
- SGI Origin
- SR8000 (HITACHI super computer)

If you are going to use this package on a different system, you need to install LAPACK and its base library BLAS. They are available via web site `[http://www.netlib.org/]`.

Both in the model and matrix solver parts, file names and parameters are given from an external file using the Fortran `NAMELIST` function. They are usually specified in a shell script for running the linear model, while set in a file **SETPAR** for using the matrix manipulation. The detail will be described in the subsequent sections. Note that all the file names must be shorter than **90 characters** in full path.

1.4 File format

For convenience, all the data files is read and written with unformatted, sequential access. Most of data files can be opened by a familiar graphical tool of **GrADS**. Sample GrADS `.ctl` files are provided to look at the results on a geographical map. Only the exception is several input files to run LBM since they need to have a data header according to a graphic software locally developed at CCSR, Univ. of Tokyo, called **Gtool**. However, users who are not familiar with Gtool would not feel inconvenience, because the package has been developed for those who do not have knowladge about the Gtool format so that they can also use the package without any trouble. The data format will be unified to GrADS in the next version.

1.5 Feedback to the developer

This package is the updated second version in which prenty of changes has been made from the first version. Although this is not a beta–version and all the program has been thoroughly tested, there may remain bugs somewhere. The developer (MW) appreciates if you could send a bug report to `hiro@ees.hokudai.ac.jp`. Also if you made any progressive improvement for this model, please let him know what was changed, in order to reflect the improvement in the third version.

2 Getting started

2.1 Unpack

When you got the LBM package, `ln_solver2.2.tar.gz`, you must uncompress and carry out initial set up. Note that the package requires at minimum 90 MB disk space.

Let `%>` be the command prompt on a Unix terminal, enter

```
%> zcat ln_solver2.2.tar.gz | tar xvf -
```

or

```
%> gunzip ln_solver2.2.tar.gz
%> tar xvf ln_solver2.2.tar
```

It will create a top–level directory `ln_solver`, followed by several sub–directories.

`ln_solver`	`/bs`		basic state files
		`/gt3`	Gtool format data
		`/grads`	GrADS format data
		`/ncep`	GrADS format NCEP data (option)
		`/ecmwf`	GrADS format ECMWF data (option)
		`/etc`	miscellaneous data (option)
	`/doc`		documents (including this file)
	`/model`		linear model
		`/bin`	model executable files
		`/doc`	AGCM change log (in Japanese)
		`/lib`	model library files
		`/sh`	shell scripts
		`/src`	model source code
	`/sample`		sample data and `.ctl` files
	`/solver`		matrix solver
		`/custom`	customized routines
		`/etc`	post processors for STM (cf. section 4)
		`/expert`	modal expansion (cf. section 5)
		`/include`	dimension files
		`/lib`	LAPACK library files
		`/steady`	steady response (cf. section 3)
		`/util`	utilities

The directory which you stand will be the home for LBM. After the package was unpacked, the environmental variable which defines the LBM home, `LNHOME`, must be specified (user name *hiro* is assumed throughout the document).

```
%> setenv LNHOME /home/hiro/ln_solver
```

or if you use `bash`,

```
%> export LNHOME='/home/hiro/ln_solver'
```

The above line can be written in your `.cshrc` or `.bashrc` to avoid an annoyance to type on every terminal. In the document, for example, moving to the LBM home is expressed as

```
%> cd $LNHOME
```

2.2 Edit the file `Lmake.inc`

First you edit a file `$LNHOME/Lmake.inc`, which defines most of setting as below. You can readily specify the system and model resolution, but others such as model type, truncation, and model option, depend on what you are going to do. Temporarily you choose one for them. The current version supports four vertical resolutions (5, 8, 11, and 20 levels) while horizontal resolution of T21 is ordinary recommended. Here let's suppose you will use T21L5 model on Sun WS.

```
# include file for Makefile for linear solver
#
# set environment LNHOME in your .cshrc
#
###### Architecture ################
ARC = sun
#ARC = alpha
#ARC = sgi
#ARC = sr8000

###### Model type ################

### time-advance linear model (incl.  storm track model)
PROJECT = tintgr

### standard, making linear matrix (incl.  stationary wave model)
#PROJECT = mkamat

### accelerated iterative method (AIM)
#PROJECT = aim

### nonlinear, dynamical core
#PROJECT = dcore
```

```
### barotropic model
#PROJECT = baro

### coupled mLBM-CZ
#PROJECT = cz

### orographic forcing
#PROJECT = wvfrc.topo

###### Horizontal Resolution ################
#HRES = t10
HRES = t21
#HRES = t42
#HRES = t63
#HRES = t106

###### Vertical Resolution ################
#VRES = l1
VRES = l5
#VRES = l8
#VRES = l11
#VRES = l20

###### zonal wave truncation ################
ZWTRN =
#ZWTRN = m15
#ZWTRN = m10
#ZWTRN = m5

###### options for model ####################

### time-advance linear model (incl.  storm track model)
######## dry model
MODELOPT = -DOPT_CLASSIC
######## moist model
#MODELOPT =

### lbm/swm
######## dry model
#MODELOPT = -DOPT_MKMAT -DOPT_CLASSIC
#MODELOPT = -DOPT_MKMAT -DOPT_OWALL -DOPT_CLASSIC
```

```
#MODELOPT = -DOPT_WVFRC -DOPT_ORHS -DOPT_CLASSIC
#MODELOPT = -DOPT_CLASSIC
######## moist model
#MODELOPT = -DOPT_MKMAT
#MODELOPT = -DOPT_MKMAT -DOPT_OWALL
#MODELOPT = -DOPT_WVFRC -DOPT_ORHS

######## dry AIM
#MODELOPT = -DOPT_CLASSIC

### nonlinear dynamical core
#MODELOPT =
#MODELOPT = -DOPT_RWRIT

### barotropic model
#MODELOPT = -DOPT_MKMAT
#MODELOPT = -DOPT_MKMAT -DOPT_OWALL
#MODELOPT = -DOPT_WVFRC

### orographic forcing
#MODELOPT = -DOPT_CLASSIC
```

2.3 Making model library

Before running the model, you need to compile the model library as

```
%> cd $LNHOME/model/src
%> make lib
```

After a while, a library such as `liblbm2t21ml5c.a` will be created under `$LNHOME/model/lib/sun/`. Since many of routines in LBM can be shared regardless of the model type and options, they are archived into the library file for convenience. The library for different system will also be stored separately. Thus, only if you change the model resolution you have to re–make the library as

```
%> cd $LNHOME/model/src
%> make clean
%> make lib
```

2.4 Preparing basic states

Several types of basic state have been provided in `$LNHOME/bs/grads` and `$LNHOME/bs/gt3`. For example, `ncepwin.t21l20.grd` is a basic state for T21L20 model, obtained from winter (DJF) average of the climatology for 1958–97 in the NCEP/NCAR reanalysis data, while `erawin.t21l20.grd` is a basic state for T21L20 model as well but from the winter climatology for 1961–90 in the ECMWF reanalysis (ERA40) data. You can use an utility to make such observational climatological fields as a basic state either derived from the NCEP–NCAR or ECMWF reanalysis.

2.4.1 NCEP–NCAR basic states

For using the NCEP data, copy the NCEP data archive `ln_solver2.2.ncepdata.tar.gz` to `$LNHOME`, then

```
%> cd $LNHOME
%> zcat ln_solver2.2.ncepdata.tar.gz | tar xvf -
```

or

```
%> cd $LNHOME
%> gunzip ln_solver2.2.ncepdata.tar.gz
%> tar xvf ln_solver2.2.ncepdata.tar
```

This results in a new directory `$LNHOME/bs/ncep` that contains 6 GrADS files and corresponding `.ctl` files. They are three sets of climatology data for different horizontal resolution (T10, T21, and T42).

Second, compile the utility

```
%> cd $LNHOME/solver/util
%> make bs
```

A resultant command, `ncepsbs`, will make a basic state by time averaging and transforming vertical coordinate from pressure to σ. To do that, you need to specify file names and parameters. Edit following lines in a parameter file `$LNHOME/solver/util/SETPAR`.

```
    &nmncp   cncep='/home/hiro/ln_solver/bs/ncep/ncep.clim.y58-97.t21.grd',
             cncep2='/home/hiro/ln_solver/bs/ncep/ncep.clim.y58-97.ps.t21.grd',
             calt='/home/hiro/ln_solver/bs/gt3/grz.t21',
             kmo=12, navg=3, ozm=f, osw=f, ousez=t
    &end
    &nmbs    cbs0='/home/hiro/ln_solver/bs/gt3/ncepwin.t21l5',
             cbs='/home/hiro/ln_solver/bs/grads/ncepwin.t21l5.grd'
    &end
```

Note that the vertical levels for the basic state has already been specified in `Lmake.inc`. When you want to examine linear response in summer (JJA), you can prepare the summer basic state as

```
&nmncp  cncep='/home/hiro/ln_solver/bs/ncep/ncep.clim.y58-97.t21.grd',
        cncep2='/home/hiro/ln_solver/bs/ncep/ncep.clim.y58-97.ps.t21.grd',
        calt='/home/hiro/ln_solver/bs/gt3/grz.t21',
        kmo=6, navg=3, ozm=f, osw=f, ousez=t
&end
&nmbs   cbs0='/home/hiro/ln_solver/bs/gt3/ncepsum.t21l5',
        cbs='/home/hiro/ln_solver/bs/grads/ncepsum.t21l5.grd'
&end
```

Alternatively, by selecting `ozm=t` the zonally uniform basic state will be obtained. After you specified the above parameters,

```
%> ncepsbs
```

will create the basic states with Gtool and GrADS format named by `cbs0` and `cbs`, respectively.

As for the barotropic model (cf. section 8), a basic state only contains stream function. If you use the barotropic model, refer to section 8 which describes how to make the basic state.

2.4.2 ERA40 basic states

For using the ERA40 data, copy the ECMWF data archive `ln_solver2.2.ecmdata.tar.gz` to `$LNHOME`, then

```
%> cd $LNHOME
%> zcat ln_solver2.2.ecmdata.tar.gz | tar xvf -
```

or

```
%> cd $LNHOME
%> gunzip ln_solver2.2.ecmdata.tar.gz
%> tar xvf ln_solver2.2.ecmdata.tar
```

This results in a new directory `$LNHOME/bs/ecmwf` that contains 3 GrADS files and corresponding `.ctl` files. They are three sets of climatology data for different horizontal resolution (T21, T42, and T63).

Second, compile the utility (you can skip if you have done before)

```
%> cd $LNHOME/solver/util
%> make bs
```

A resultant command, `ecmsbs`, will make a basic state by time averaging and transforming vertical coordinate from pressure to σ. To do that, you need to specify file names and parameters. Edit following lines in a parameter file `$LNHOME/solver/util/SETPAR`.

```
&nmecm  cecm='/home/hiro/ln_solver/bs/ecmwf/ERA40.clim.t21.grd',
        calt='/home/hiro/ln_solver/bs/gt3/grz.t21',
        kmo=12, navg=3, ozm=f, osw=f
&end
&nmbs   cbs0='/home/hiro/ln_solver/bs/gt3/erawin.t21l5',
        cbs='/home/hiro/ln_solver/bs/grads/erawin.t21l5.grd'
&end
```

Note that the vertical levels for the basic state has already been specified in `Lmake.inc`. When you want to examine linear response in summer (JJA), you can prepare the summer basic state as

```
&nmecm  cecm='/home/hiro/ln_solver/bs/ecmwf/ERA40.clim.t21.grd',
        calt='/home/hiro/ln_solver/bs/gt3/grz.t21',
        kmo=6, navg=3, ozm=f, osw=f
&end
&nmbs   cbs0='/home/hiro/ln_solver/bs/gt3/erasum.t21l5',
        cbs='/home/hiro/ln_solver/bs/grads/erasum.t21l5.grd'
&end
```

Alternatively, by selecting `ozm=t` the zonally uniform basic state will be obtained. After you specified the above parameters,

```
%> ecmsbs
```

will create the basic states with Gtool and GrADS format named by `cbs0` and `cbs`, respectively.

3 Steady response

A major use of LBM is to calculate a steady, atmospheric response to prescribed forcing such as the diabatic heating anomaly. In this package, we provide three methods to obtain the steady response. The first method described in 3.1 is a time integration of the model. This method has been often used to obtain steady response (e.g. Jin and Hoskins 1995). Another two methods are both based on a matrix inversion for the steady version of the dynamical equation. While they definitely ensure that the response is an exactly steady solution to a forcing, the matrix inversion may be computationally expensive in particular for the second method presented in section 3.2. Since the best way to obtain the steady solution depends on problem which you want to address, you should choose a suitable method by taking your purpose into account.

3.1 Time integration

The time integration of LBM can be carried out in a similar manner to the AGCM, except that the model requires a basic state and initial perturbation or steady forcing but not the boundary and initial conditions. The output file contains the following 8 quantities (you cannot add or remove the variable).

· stream function	[m^2 s^{-1}]
· velocity potential	[m^2 s^{-1}]
· zonal wind	[m s^{-1}]
· meridional wind	[m s^{-1}]
· pressure vertical velocity (ω)	[hPa s^{-1}]
· temperature	[K]
· geopotential height	[m]
· surface pressure	[hPa]

For the computation, there are four steps, as explained below.

3.1.1 Making model binary

Choose `'PROJECT = tintgr'` in `$LNHOME/Lmake.inc`. Note that the zonal wave truncation and model option in `$LNHOME/Lmake.inc` should be specified as

```
###### zonal wave truncation ###############
ZWTRN =
#ZWTRN = m15
#ZWTRN = m10
#ZWTRN = m5

###### options for model ###################
```

```
### time-advance linear model (incl.  storm track model)
######## dry model
MODELOPT = -DOPT_CLASSIC
######## moist model
#MODELOPT =
```

Then, enter the following command.

```
%> cd $LNHOME/model/src
%> make clean.special
%> make lbm
```

Suppose now you selected the horizontal resolution of T21 and vertical 20 levels on Sun WS, the above procedure will generate an executable file `lbm2.t21ml20ctintgr` under the directory `$LNHOME/model/bin/sun/`. This is the executable file to run LBM in time.

3.1.2 Preparing forcing

Before running LBM, you have to prepare a GrADS file of initial perturbation or steady forcing data (both have the same format). A sample steady forcing with T21L20 resolution is found in `$LNHOME/sample/`. The forcing file `frc.t21l20.classic.grd` mimics a diabatic heting anomaly associated with El Niño, and can be used to test your model. As readily understood by looking at the corresponding `.ctl` file, the file contains all the prognostic variables, so that you can make other types of steady forcing, e.g., vorticity forcing instead of heating.

We also provide a tool, `mkfrcng`, to easily make an idealized simple forcing. A forcing data can be created after you edit the parameter file `$LNHOME/solver/util/SETPAR`. For example, the sample heating data `frc.t21l20.classic.grd` was made by specifying parameters as follows.

```
&nmfin  cfm='/home/hiro/ln_solver/data/frc.t21l20.classic.mat',
        cfg='/home/hiro/ln_solver/sample/frc.t21l20.classic.grd',
        fact=1.0,1.0,1.0,1.0,1.0
&end
&nmvar  ovor=f, odiv=f, otmp=t, ops=f, osph=t
&end
&nmhpr  khpr=1,
        hamp=1.,
        xdil=40.,
        ydil=12.,
        xcnt=210.,
        ycnt=0.
&end
```

```
&nmvpr  kvpr=2,
        vamp=8,
        vdil=20.,
        vcnt=0.45
&end
&nmall  owall=t
&end
&nmcls  oclassic=t
&end
```

The file defined by `cfg` is the GrADS forcing data while `cfm` specifies corresponding spherical coefficient data, which is not used here but necessary for sections 3.2 and 3.3. Other parameters are briefly described, but see a description file `$LNHOME/solver/util/param_list` for more details. `&nmvar` indicates a forcing for which variable (consider only temperature forcing in the above example). The horizontal and vertical shapes of the forcing are determined by `&nmhpr` and `&nmvpr`, in which:

`khpr`	horizontal shape: 1=elliptic, 2=zonal uniform
`hamp`	amplitude of horizontal function
`xdil`	zonal extent (half radius)
`ydil`	meridional extent (half radius)
`xcnt`	center longitude from 0°to 360°
`ycnt`	center latitude from -90°to 90°

`kvpr`	vertical profile: 1=sinusoidal, 2=gamma, 3=uniform
`vamp`	amplitude of vertical profile (unit is per day)
`vdil`	dilation parameter (only for `kvpr=2`)
`vcnt`	center level in σ

After you modified `$LNHOME/solver/util/SETPAR` following the above explanation, then

```
%> cd $LNHOME/solver/util
%> make clean
%> make
%> mkfrcng
```

will make a new forcing file. You should check, by looking at with GrADS, if the forcing structure is consistent with what you thought.

3.1.3 Running the model

`$LNHOME/model/sh/tintgr/linear-run.classic.csh` is a sample shell script for the time integration of the linear model. Copy it to any name, say, `linear-run.test.csh`, and edit suitably.

```
%> cd $LNHOME/model/sh/tintgr
%> cp linear-run.classic.csh linear-run.test.csh
%> chmod u+x linear-run.test.csh
```

First, you may need to change environmental variables set in the script.

`LNHOME`	home directory of the model package
`SYSTEM`	architecture
`RUN`	model executable file
`FDIR`	directory which contains forcing data
`DIR`	directory for first model products (Gtool format)
`BSFILE`	basic state (Gtool formatted data)
`FRC`	initial perturbation (if necessary)
`SFRC`	steady forcing (if necessary)
`TEND`	length of time integration in day

If you choose to give a steady forcing, `FRC` is not used (perhaps you can specify the same file as `SFRC` as dummy) whereas to give an initial perturbation, vice versa. The duration of the model integration depends on how long the system remains stable. If, for example, you set weak dissipations, some of eigenmodes in the system may grow faster that prevent you to integrate the model beyond certain time. When the dissipation is unrearistically strong such as not to allow any mode growing, you can continue to integrate the model without limitation. With relevant dissipation, which will be explained next, the model integration will be continued up to around 30 days (after that baroclinic waves rapidly grow and blow up). Since, at least in lower latitudes, the response is going to be equilibrated after day 10 or so, you may set `TEND` at 25 then examine the response around day 15 or 20. Note that this method results in only a good approximation to a steady solution to prescribed forcing unless the dissipation is enough strong. If you need to obtain the exact steady solution to a forcing, choose matrix inversion methods as explained in sections 3.2 and 3.3. Nevertheless, the time integration has advantages over the matrix inversion method such that you can estimate timescale of the response (see, for example, Jin and Hoskins 1995).

Before running the model, you may further check the parameters specified in the shell script. In particular, you have to be careful for setting relevant dissipations (diffusion and linear drag, see Watanabe and Kimoto 2000). The corresponding parameters are listed below to notice.

```
&nmhdif  order=4, tefold=6, tunit='HOUR'                                  &end
&nmdamp  ddragv=1,1,1,5,15,30,30,30,30,30,30,30,30,30,30,30,30,30,1,1,
         ddragd=1,1,1,5,15,30,30,30,30,30,30,30,30,30,30,30,30,30,1,1,
         ddragt=1,1,1,5,15,30,30,30,30,30,30,30,30,30,30,30,30,30,1,1,
         tunit='DAY'                                                      &end


&nmvdif  vdifv=1.d3,1.d3,1.d3,1.d3,1.d3,1.d3,1.d3,1.d3,1.d3,1.d3,
         1.d3,1.d3,1.d3,1.d3,1.d3,1.d3,1.d3,1.d3,1.d3,1.d3,
         vdifd=1.d3,1.d3,1.d3,1.d3,1.d3,1.d3,1.d3,1.d3,1.d3,1.d3,
         1.d3,1.d3,1.d3,1.d3,1.d3,1.d3,1.d3,1.d3,1.d3,1.d3,
         vdift=1.d3,1.d3,1.d3,1.d3,1.d3,1.d3,1.d3,1.d3,1.d3,1.d3,
         1.d3,1.d3,1.d3,1.d3,1.d3,1.d3,1.d3,1.d3,1.d3,1.d3,           &end
&nmfrc   ffrc='$FRC', oper=f, nfcs=1                                    &end
&nmsfrc  fsfrc='$SFRC', ofrc=t, nsfcs=1, fsend=-1,1,30,0,0,0            &end
```

`&nmhdif` defines the shape and strength of the horizontal diffusion. The linear model usually employs biharmonic (∇^4) diffusion, so that `order=4`. Since the horizontal diffusion is scale–selective, the magnitude of the diffusion coefficient is defined by the e–folding decay time for the largest wavenumber (`tefold=6, tunit='HOUR'` means 6hr decay time). It has been confirmed that a weaker diffusion up to 24hr decay time does not seriously alter the response. For reference, the CCSR/NIES AGCM, from which LBM has been developed, employs much weaker diffusion of `order=8, tefold=24, tunit='HOUR'` .

Linear drag which mimics Rayleigh friction and Newtonian damping is commonly applied to LBM. The drag term also imitates a dissipative process due to nonlinearity (Ting and Yu 1998). The response often becomes sensitive to the choice of drag coefficient. A set of parameter `&nmdamp` defines the coefficient for ζ (`ddragv`), D (`ddragd`), and T (`ddragt`) with the time unit `tunit`. In the above example, damping time scale is set at 1dy for the lowest three levels and the topmost two levels, 5 and 15dy for the fourth and fifth levels, and 30dy elsewhere.

Vertical diffusion is also included in order to suppress a vertical computational mode. You do not need to change the magnitude, set at $(1000\text{dy})^{-1}$ in the above.

`&nmfrc` and `&nmsfrc` gives initial perturbation and steady forcing, respectively. If `oper` (`ofrc`) is assigned to `f`, which means `.false.`, the initial perturbation (steady forcing) set in the line will be ignored. In the above example, the model is designated to run with steady forcing but without initial perturbation. `nfcs` and `nsfcs` are a count for the different data, which will be changed for ensemble simulation such as in the storm track model (section 4). Finally, if you want to turn off the forcing at a certain time, say, day 10, set `fsend=0,1,10,0,0,0`. When the first number of `fsend` is -1 as in the sample script, this switch is not used.

You can now run the linear model as

```
%> cd $LNHOME/model/sh/tintgr
%> linear-run.test.csh
```

All the log message to the standard output is written down to a file `$LNHOME/$DIR/SYSOUT` (see `linear-run.test.csh`). You will know the current model step by looking at the end line of `SYSOUT`.

3.1.4 Post process and visualize

The outputs for the model run (specified as `&nmhist` in `$LNHOME/model/sh/tintgr/linear-run.classic.csh`) are generated for 8 variables with the Gtool format. For convenience, they are incorporated into one GrADS file by using a tool `$LNHOME/solver/util/gt2gr`. Specify the following parameters or filenames in `$LNHOME/solver/util/SETPAR`:

```
&nmfgt  cfs='/home/hiro/ln_solver/data/out/psi',
        cfc='/home/hiro/ln_solver/data/out/chi',
        cfu='/home/hiro/ln_solver/data/out/u',
        cfv='/home/hiro/ln_solver/data/out/v',
        cfw='/home/hiro/ln_solver/data/out/w',
        cft='/home/hiro/ln_solver/data/out/t',
        cfz='/home/hiro/ln_solver/data/out/z',
        cfp='/home/hiro/ln_solver/data/out/p',
        cfq='/home/hiro/ln_solver/data/out/q',
        cfx='/home/hiro/ln_solver/data/out/dt',
        cfy='/home/hiro/ln_solver/data/out/dq',
        cfo='/home/hiro/ln_solver/data/tintgr/linear.t21l20.classic.grd',
        fact=1.0,1.0,1.0,1.0,1.0,1.0,1.0,1.0,1.0,1.0,1.0,
        opl=t,
&end
&nmbs   cbs0='/home/hiro/ln_solver/bs/gt3/ncepwin.t21l20',
        cbs='/home/hiro/ln_solver/bs/grads/ncepwin.t21l20.grd'
&end
&nmcls  oclassic=t
&end
```

where files from `cfs` to `cfo` are the Gtool data of the first products, in which `cfq, cfx, cfy`, and `cfy` are not used here, so that you can specify any name for them. They are only used in the moist linear model integration (cf. section 7.1). The GrADS data which you will get is defined by `cfo`. Basic state files defined by `&nmbs` must coincide the basic state used in the model although the Gtool file (`cbs0`) is not used in this procedure.

After you modified `$LNHOME/solver/util/SETPAR` as above, then

```
%> cd $LNHOME/solver/util
```

```
%> gt2gr
```

or, if you skipped the procedure to make forcing (3.1.2),

```
%> cd $LNHOME/solver/util
%> make clean
%> make
%> gt2gr
```

A sample control file `linear.t21l20.classic.ctl` is found in `$LNHOME/sample/`. You can copy it and modify the filename written in the `.ctl` file, so that the result can be drawn on global map using GrADS.

3.2 Matrix inversion. Part I: Solve full matrix

In many studies that use LBM as a diagnostic tool, the steady forced problem has been solved with a linear matrix inversion (Hoskins and Karoly 1981; Nigam et al. 1986; Held et al. 1989; Valdes and Hoskins 1989; Ting and Held 1990; Nigam 1994; Watanabe and Kimoto 1999; DeWeaver and Nigam 2000; Kimoto et al. 2001, among others). The method is described in this and the next sections.

A matrix for the linear dynamical operator, which consists of a set of spherical harmonic coefficients, is obtained basically by the so–called 'residual method' (Hoskins and Karoly 1981). However, amplitude of δ function in perturbation vectors is not necessary to be tiny any more since the model code has been exactly linearized. While mathematical expression for the steady equations is independent of the structure of the basic state, practical method to solve the system linearized about the zonally varying, 3D basic state is somewhat different from the case of the zonally uniform basic state. The former is considered in this section, and 3.3 describes how to deal with the latter case. Note that the latter method requires lower computational cost since it solves small block matrices (cf. 3.3), while the former case which solves one full matrix demands sufficiently large computer memory.

Suppose you use the T21L20 LBM for a steady problem. Total wave number for T21 resolution is 483, then the rank of the linear matrix N becomes

$$N = 483 \times (\ 3\ (\text{variables; } \zeta,\ \text{D, T}) \times 20\ (\text{levels}) + 1\ (\text{variable, } P_s)\ ) = 29643 \quad .$$

It implies that you have to solve a huge matrix of around 30000×30000, for which your computer must have a memory greater than 7.5GB! To avoid such a computational burden, we simply suggest to reduce the rank by selecting coarser vertical resolution (L5 or L11) besides truncating the zonal wave number of 21 at 5, 10, or 15. For your reference, solving T21L11 model with zonal waves truncated at 10 will take about 15 minutes on a HITACHI super computer, SR8000. If you would like to obtain the steady response with higher resolution, you may choose an accelerated iterative method (AIM) which we newly developed (see 3.4) but not direct method described here.

3.2.1 Making model binary

Choose `'PROJECT = mkamat'` in `$LNHOME/Lmake.inc`. To test the model, first solve a small matrix of T21L5 which truncates zonal wave number 5, i.e., edit `$LNHOME/Lmake.inc` as follows.

```
###### Vertical Resolution ################
#VRES = l1
VRES = l5
#VRES = l8
#VRES = l11
#VRES = l20
###### zonal wave truncation ################
#ZWTRN =
#ZWTRN = m15
#ZWTRN = m10
ZWTRN = m5

###### options for model ###################

### time-advance linear model (incl.  storm track model)
######## dry model
#MODELOPT = -DOPT_CLASSIC
######## moist model
#MODELOPT =

### lbm/swm
######## dry model
#MODELOPT = -DOPT_MKMAT -DOPT_CLASSIC
MODELOPT = -DOPT_MKMAT -DOPT_OWALL -DOPT_CLASSIC
#MODELOPT = -DOPT_WVFRC -DOPT_ORHS -DOPT_CLASSIC
######## moist model
#MODELOPT = -DOPT_MKMAT
#MODELOPT = -DOPT_MKMAT -DOPT_OWALL
#MODELOPT = -DOPT_WVFRC -DOPT_ORHS
```

Then, make the model executable file as in the previous section.

```
%> cd $LNHOME/model/src
%> make clean.special
%> make lbm
```

This will create an executable file `lbm2.t21ml5cmkamat`.

3.2.2 Computing linear operator matrix

`$LNHOME/model/sh/mkamat/mkamat.l5.classic.csh` is a sample shell script for the model run to make a linear operator matrix. Copy it to any name, say, `mkamat.l5.test.csh`, and edit lines if necessary. The environmental variables and parameters defined in the script are almost the same as those described in 3.1.3.

```
%> cd $LNHOME/model/sh/mkamat
%> cp mkamat.l5.classic.csh mkamat.l5.test.csh
%> chmod u+x mkamat.l5.test.csh
```

At the same time, specify the file name of the linear operator in `$LNHOME/solver/util/SETPAR`,

```
 &nmred  cdr='/home/hiro/ln_solver/matrix',
         cfo='/home/hiro/ln_solver/matrix/MAT.t21l5.ncepwin.dat'
 &end
 &nmall  owall=t
 &end
 &nmcls  oclassic=t
 &end
```

where `cfo` defines the matrix file name while `cdr` specifies a directory for temporary files of the column vectors.

The residual method performed by the above script repeat N–times one–step integration of the linear model. Our LBM actually repeats 5 (zonal wave number) $\times$ (3 (ζ, D, T) $\times$ 5 (levels) + 1 (P_s)) times loop in the script `mkamat.l5.test.csh`. Thus you will see a message on the terminal window readily after you execute the script, like

```
%> cd $LNHOME/model/sh/mkamat
%> mkamat.l5.test.csh
VOR zonal wave M is 0 and LEVEL is 1
VOR zonal wave M is 1 and LEVEL is 1
VOR zonal wave M is 2 and LEVEL is 1
VOR zonal wave M is 3 and LEVEL is 1
VOR zonal wave M is 4 and LEVEL is 1
..............
LnPs zonal wave M is 4 and LEVEL is 1
LnPs zonal wave M is 5 and LEVEL is 1
```

Each line represents an execution of LBM that yields a set of column–vector data such as `$LNHOME/matrix/MAT_v1/AMATRIX0`. After all the column–vector data is computed, the script automatically gives a command, `redist`, to gather them to one matrix file (`MAT.t21l5.ncepwin.dat` in the above example). Thus you will see a subsequent message that follows the above like

```
INPUT:/home/hiro/ln_solver/matrix/MAT_v1/AMATRIX0
INPUT:/home/hiro/ln_solver/matrix/MAT_v2/AMATRIX0
..............
INPUT:/home/hiro/ln_solver/matrix/MAT_t4/AMATRIX5
INPUT:/home/hiro/ln_solver/matrix/MAT_t5/AMATRIX5
INPUT:/home/hiro/ln_solver/matrix/MAT_p1/AMATRIX5
```

3.2.3 Preparing forcing

As in 3.1.2, the utility tool `mkfrcng` can be used to make a simplified forcing. Unlike the time integration, GrADS data file is not used to solve the matrix (but it is still useful to look at the forcing structure), instead, a vector data containing spherical coefficient of the forcing function is needed. In `$LNHOME/solver/util/SETPAR`,

```
    &nmfin  cfm='/home/hiro/ln_solver/data/frc.t21l5.classic.mat',
            cfg='/home/hiro/ln_solver/data/frc.t21l5.classic.grd',
            fact=1.0,1.0,1.0,1.0,1.0
    &end
```

where `cfm` defines the forcing data in a spectral wave space. You can just do

```
%> cd $LNHOME/solver/util
%> mkfrcng
```

to obtain that data file.

If you use more realistic or complicated forcing, you can make the GrADS data file by yourself although the data must follow the format defined in the LBM package (see `frc.t21l5.classic.ctl`). Such a user–made forcing can be converted to the spherical coefficient by using `fvec`. Let your customized forcing `frc.custom.grd`, edit `$LNHOME/solver/util/SETPAR`,

```
    &nmfin  cfm='/home/hiro/ln_solver/data/frc.t21l5.classic.mat',
            cfg='/home/hiro/ln_solver/data/frc.custom.grd',
            fact=1.0,1.0,1.0,1.0,1.0
    &end
```

Then,

```
%> cd $LNHOME/solver/util
%> fvec
```

3.2.4 Solve linear system

You can now solve the linear system by using a command `lin` and `trn` both are compiled at `$LNHOME/solver/steady`. `lin` solves the linear system while `trn` convert the response in wave space to that in a physical space, together with performing $\sigma \rightarrow$ pressure transform and calculating geopotential height.

```
%> cd $LNHOME/solver/steady
%> make clean
%> make
```

Before executing `lin` and `trn`, parameter and file names must be adequately specified as in the previous sections. Edit following lines in `$LNHOME/solver/steady/SETPAR`.

```
 &nmfin  cfm='/home/hiro/ln_solver/matrix/MAT.t21l5.ncepwin.dat',
         cfr='/home/hiro/ln_solver/data/frc.t21l5.classic.mat',
         cfs='/home/hiro/ln_solver/data/x.dat',
         cfg='/home/hiro/ln_solver/data/rsp.t21l5.classic.grd'
 &end
 &nmbs   cbs='/home/hiro/ln_solver/bs/grads/ncepwin.t21l5.grd'
 &end
 &nmuv   o2uv=f, opl=t
 &end
 &nmall  owall=t
 &end
 &nmcls  oclassic=t
 &end
```

At the first lines, `cfm` defines the file name of the linear operator matrix while `cfr` specifies the forcing vector both of which have already been prepared. `cfs` is a temporary file of the response (wave coefficient data) and `cfg` is the GrADS response data that you will look at later. The GrADS basic state `cbs` is necessary for converting $\ln P_s$ to P_s and also σ level data to the pressure data. If you want to obtain the σ, but not pressure, level data, specify `opl=f`. When a companion parameter `o2uv` is `.true.`, the output file contains zonal and meridional wind responses instead of stream function and velocity potential.

To obtain the response data defined by `cfg`, enter

```
%> lin ; trn
```

It will show a message on your terminal like

```
### SOLVE LINEAR SYSTEM ###
***********************************************
Read matrix L
```

```
   Matrix Size (left)= 3376 x 3376
Read vector F
Solve for X = L-1 F
   LAPACK routine call:  (DGESV)
   INFO code returned by DGESV = 0
***********************************************
### END OF EXECUTION ###
```

If a flag of `INFO code` in the above message is not 0, it indicates that the linear system was not correctly solved. In this case you need to check the matrix and/or forcing.

A set of GrADS data and `.ctl` file for the above example is found in `$LNHOME/sample`. First, you can look at, say, P_s and the 500 hPa height response in the sample file as (note that `ga->` denotes a GrADS prompt)

```
%> cd $LNHOME/sample
%> grads
ga-> open rsp.t21l5.classic.ctl
ga-> d p
ga-> d z(z=3)
```

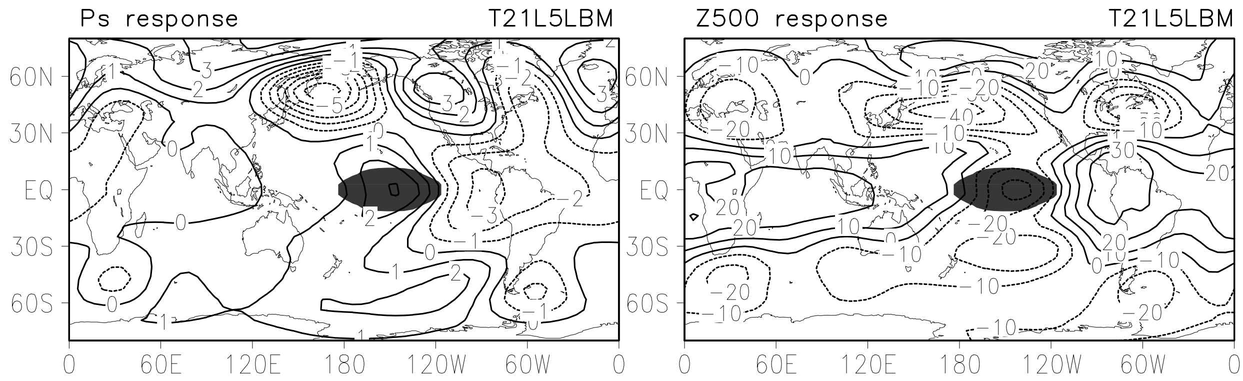


Figure 3.1 Example of the T21L5 linear response. Surface pressure (left) and 500 hPa geopotential heigh (right). Shading denotes a heating at $\sigma = 0.44$ ($Q \geq 1\text{K day}^{-1}$).

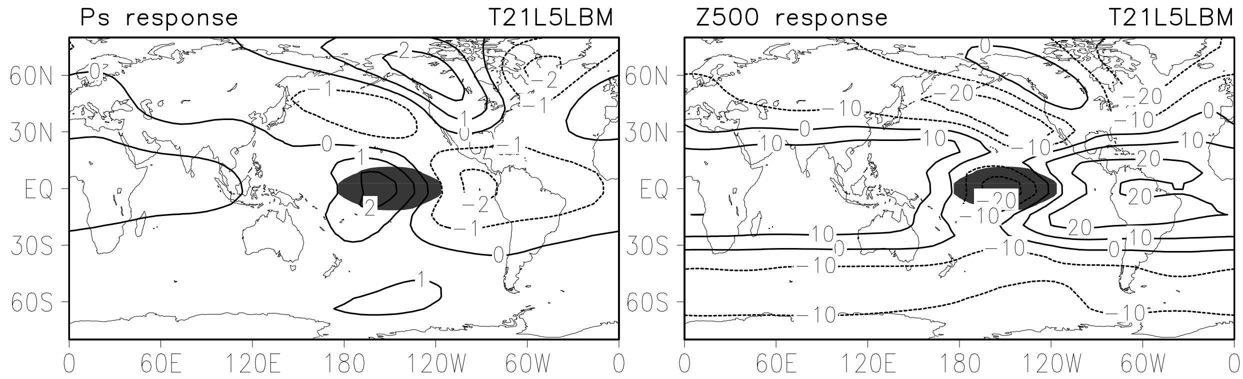


Figure 3.2 As in Fig. 3.1, except for the zonally uniform basic state.

As shown in Fig. 3.1, you will see a surface low pressure response and a trough over the North Pacific Ocean. The height response extends toward Canada and North America, which somewhat resembles the PNA pattern. Copy the `.ctl` file to your output directory (`$LNHOME/data` in the above example), and look at your response data. Did you get the same results? Also, another example of the steady response is shown in Fig. 3.2, which is exactly the same as `rsp.t21l5.classic.ctl` except that the basic state has been replaced by its zonal mean. Now the Rossby wave train from the heat source region to North America is more clearly seen. Try to reproduce the response shown in Fig. 3.2, using `$LNHOME/bs/gt3/ncepwin.t21l5zm` and `$LNHOME/bs/grads/ncepwin.t21l5zm.grd`.

If you use a different setting, see another sample `.ctl` file `rsp.t10l11.classic.ctl`, in which the T10L11 matrix is assumed to be solve with parameters

```
&nmfin  cfm='/home/hiro/ln_solver/matrix/MAT.t10l11.ncepwin.dat',
        cfr='/home/hiro/ln_solver/data/frc.t10l11.classic.mat',
        cfs='/home/hiro/ln_solver/data/x.dat',
        cfg='/home/hiro/ln_solver/data/rsp.t10l11.classic.grd'
&end
&nmbs   cbs='/home/hiro/ln_solver/bs/grads/ncepwin.t10l11.grd'
&end
&nmuv   o2uv=t, opl=f
&end
&nmall  owall=t
&end
&nmcls  oclassic=t
&end
```

The computational time of the procedures described in this section is largely devoted to the matrix preparation (running `mkamat.l5.test.csh`) and to the matrix inversion (`lin`). If you change the model basic state or dissipation (magnitude of the diffusion and linear drag terms), the former should be replicated, whereas the latter must be executed again if you altered a forcing. When you have multiple forcing data and want to solve the response to each of the forcing, it is convenient to compute an inverse matrix once. You can obtain the response by multiplying the inverse matrix with the forcing, which is much faster than using `lin`. See 5.1 for such an application.

3.3 Matrix inversion. Part II: Stationary Wave Model (SWM)

As argued in the introduction of section 3.2, solving a full matrix is computationally expensive when we select a higher model resolution. On the other hand, lower resolution is often found insufficient to discuss an issue which you want to address. An alternative way to compromise is to use a zonally uniform basic state. It means that in such a case you eliminate a wave–wave interaction in steady equations. Actually, LBM with a zonal mean basic state was quite popular until mid–1980s (e.g. Hoskins and Karoly 1981) when no one has had a computer with large enough memory. Even now, such a model (here referred to as the stationary wave model, SWM) is used to figure out a simple prototype for the complicated nature. Unless you expect a crucial role of the wave–wave interaction, you do not need to hesitate to use SWM.

In a wave space, wave–wave interaction terms are located in the off–diagonal part of the linear operator matrix. When we use the zonal mean basic state, those elements become 0, so that the full matrix is divided into several block diagonal matrices each of which corresponds to a particular zonal wave number. Unlike in section 3.2 you can solve those relatively small matrices but not a huge matrix, this reduction in the size of matrix greatly saves your computation time. For example, T21L20 SWM is now separated into 21 different matrices and a size of one matrix, say, for zonal wave number 1, will be

$$N_1 = 21 \times 2 \times (\ 3\ \text{(variables; } \zeta,\ \text{D, T)} \times 20\ \text{(levels)} + 1\ \text{(variable, } P_s)\ ) = 2562 \quad .$$

This size is one order smaller than the full matrix and might be acceptable for your computer memory.

3.3.1 Making model binary

As in 3.2.1, choose `'PROJECT = mkamat'` in `$LNHOME/Lmake.inc`. Although now you can choose higher resolution, let's consider to solve the sample matrix as in 3.2.1, namely, a T21L5 model which truncates zonal wave number 5. For SWM, model option must be set at

```
###### options for model ###################

### time-advance linear model (incl.  storm track model)
######## dry model
#MODELOPT = -DOPT_CLASSIC
######## moist model
#MODELOPT =
```

```
### lbm/swm
######## dry model
MODELOPT = -DOPT_MKMAT -DOPT_CLASSIC
#MODELOPT = -DOPT_MKMAT -DOPT_OWALL -DOPT_CLASSIC
#MODELOPT = -DOPT_WVFRC -DOPT_ORHS -DOPT_CLASSIC
```

Then, make the model executable file.

```
%> cd $LNHOME/model/src
%> make clean.special
%> make lbm
```

This will create an executable file `lbm2.t21ml5cmkamat` (same name as in section 3.2).

3.3.2 Computing linear operator matrix

You can use the same shell script as used in 3.2.2, say, mkamat.l5.test.csh. Note, however, that the basic state must be necessarily zonally uniform fields. Set the environmental variable in `$LNHOME/model/sh/mkamat/mkamat.l5.test.csh` as

```
setenv BSFILE $LNHOME/bs/gt3/ncepwin.t21l5zm       # Atm.  BS File
```

The file name of the linear operator is suggested to distinguish with a full matrix, so that edit `$LNHOME/solver/util/SETPAR`,

```
    &nmred  cdr='/home/hiro/ln_solver/matrix',
            cfo='/home/hiro/ln_solver/matrix/MATSWM.t21l5.ncepwin.dat'
    &end
    &nmall  owall=f
    &end
```

where the file defined by `cfo` is a linear matrix for SWM, while a parameter `owall=f` specifies that the matrix solver is applied to each block diagonal matrix but not to a full matrix. Execute the script

```
%> cd $LNHOME/model/sh/mkamat
%> mkamat.l5.test.csh
```

which will make a matrix file `MATSWM.t21l5.ncepwin.dat` after a while.

3.3.3 Preparing forcing

A procedure to prepare forcing is exactly the same as in 3.1.2 and 3.2.3, except for one parameter in `$LNHOME/solver/util/SETPAR`,

```
&nmall  owall=f
&end
```

As in the computation of linear matrix for SWM, the forcing vector is re–ordered for SWM, which is defined by `owall=f`. Since the GrADS data of the forcing can be shared with other types of steady response computation, if you already have a GrADS forcing file you only do

```
%> cd $LNHOME/solver/util
%> fvec
```

to obtain the wave coefficients for the forcing. See 3.1.2 and 3.2.3 for how to make the GrADS forcing data.

3.3.4 Solve linear system

Commands to solve linear matrices of SWM are the same as in used previously, i.e. `lin` and `trn`, both are compiled at `$LNHOME/solver/steady`. If you once compiled with the resolution and model option employed here, namely, if you followed the procedures in section 3.2, you do not replicate the compilation. If not, enter

```
%> cd $LNHOME/solver/steady
%> make clean
%> make
```

to obtain the commands. As in the preparation of matrix and forcing, parameter `&nmall` in `$LNHOME/solver/steady/SETPAR` is different from that specified in section 3.2. Also, you can change the file name of the steady response if you want. You must remember that the basic state is the zonally uniform state. Edit the line in `$LNHOME/solver/steady/SETPAR` as

```
&nmfin  cfm='/home/hiro/ln_solver/matrix/MATSWM.t21l5.ncepwin.dat',
        cfr='/home/hiro/ln_solver/data/frc.t21l5.classic.mat',
        cfs='/home/hiro/ln_solver/data/x.dat',
        cfg='/home/hiro/ln_solver/data/rspswm.t21l5.classic.grd'
&end
&nmbs   cbs='/home/hiro/ln_solver/bs/grads/ncepwin.t21l5zm.grd'
&end
&nmall  owall=f
&end
```

and subsequently,

```
%> lin ; trn
```

It will show a message on your terminal like

```
### SOLVE LINEAR SYSTEM ###
***********************************************
Read matrix L
Read matrix L for zonal WN = 0
   Matrix Size (left)= 336 x 336
Read vector F
Solve for X = L-1 F
   LAPACK routine call:  (DGESV)
   INFO code returned by DGESV = 0
Read matrix L for zonal WN = 1
   Matrix Size (left)= 672 x 672
Read vector F
Solve for X = L-1 F
   LAPACK routine call:  (DGESV)
   INFO code returned by DGESV = 0
........
***********************************************
### END OF EXECUTION ###

### transform wave --> grid ###
@@@ output:
/home/hiro/ln_solver/data/rspswm.t21l5.classic.grd
........
@@@ DVTMP: VIRTUAL TEMP. 93/11/08
# zonal wave:  0 check count:  336
# zonal wave:  1 check count:  672
# zonal wave:  2 check count:  640
# zonal wave:  3 check count:  608
# zonal wave:  4 check count:  576
# zonal wave:  5 check count:  544
### END OF EXECUTION ###
```

If you recall the description of the SWM, it is recognized that the size of matrix was reduced to roughly 1/10 of that for the full matrix. Therefore you will find that this step does not take long time any more.

A set of GrADS data and `.ctl` file for the above example is found in `$LNHOME/sample`. First, you can look at, say, P_s and the 500 hPa height response in the sample file as (note that ga-> denotes a GrADS prompt)

```
%> cd $LNHOME/sample
%> grads
ga-> open rspswm.t21l5.classic.ctl
ga-> d sum(p,t=1,t=6)
ga-> set z=3
ga-> d z(t=1,t=6)
```

Note that the response is obtained for each zonal wave number and stored in GrADS file as if it is a time sequence, so that the total response field can be drawn by summing up those responses (time=1 and time=6 correspond to the zonal wave number 0 and 5, respectively). The steady response as drawn by the above GrADS manipulation is shown in Fig. 3.3, which is almost identical to Fig. 3.2 by definition.

Try to reproduce the response shown in Fig. 3.3, and confirm that the SWM is equivalent to the conventional (namely solving a full matrix) steady response using the zonally uniform basic state.

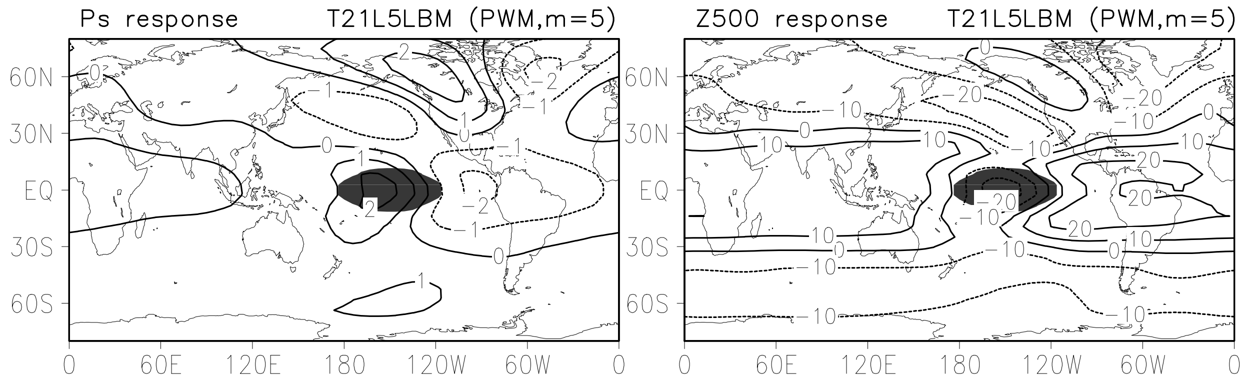


Figure 3.3 Example of the T21L5 linear response using SWM. Surface pressure (left) and 500 hPa geopotential heigh (right). Shading denotes a heating at $\sigma = 0.44$ ($Q \geq 1\text{K day}^{-1}$).

If you use a different setting, see another sample .ctl file rspswm.t21l20.classic.ctl, in which the T21L20 matrix with the wave truncation at $m = 10$ is assumed to be solve with parameters

```
&nmfin  cfm='/home/hiro/ln_solver/matrix/MATSWM.t21l20.ncepwin.dat',
        cfr='/home/hiro/ln_solver/data/frc.t21l20.classic.mat',
        cfs='/home/hiro/ln_solver/data/x.dat',
        cfg='/home/hiro/ln_solver/data/rspswm.t21l20.classic.grd'
&end
&nmbs   cbs='/home/hiro/ln_solver/bs/grads/ncepwin.t21l20zm.grd'
&end
&nmuv   o2uv=t, opl=t
&end
&nmall  owall=f
&end
&nmcls  oclassic=t
&end
```

3.3.5 Remarks on the orographic forcing

The SWM has been commonly used to diagnose the atmospheric response to orography (e.g., Hoskins and Karoly 1981). Unlike the quasi-geostrophic model in which orographic forcing is simply represented by the zonal gradient of orography multiplied by the background zonal wind, the orographic forcing in the σ–level primitive equations has a little complicated form because a part of the orographic forcing driving divergence should be solved in the semi–implicit way. To facilitate the estimate of the orographic forcing, we provide special routines as described below.

Again, suppose we use the T21L5 SWM as an example. First, choose `'PROJECT = wvfrc.topo'` in `$LNHOME/Lmake.inc` as well as the option

```
###### options for model ####################

### orographic forcing
MODELOPT = -DOPT_CLASSIC
```

Then, make the model executable file.

```
%> cd $LNHOME/model/src
%> make clean.special
%> make lbm
```

This will create an executable file `lbm2.t21ml5cwvfrc.topo`.

A sample script to run the above executable file is `$LNHOME/model/sh/wvfrc.topo/wvfrc.t21l5.topo.csh`. The output is the grid data of the orographic forcing, and the file name is specified as

```
setenv FRCGRD $LNHOME/data/frc/frc.t21l5zm.topo.grd      # topo.  forcing
```

After you edit the script as appropriate, just run it.

```
%> cd $LNHOME/model/sh/wvfrc.topo
%> wvfrc.t21l5.topo.csh
```

The procedure to convert the GrADS formatted forcing file to the spectral coefficients, you can refer to section 3.3.3. The SWM response to the orographic forcing inthe above example is shown in Fig. 3.4, which shows the 300 hPa wind and 500 hPa height response quite similar to the results in Nigam et al. (1988).

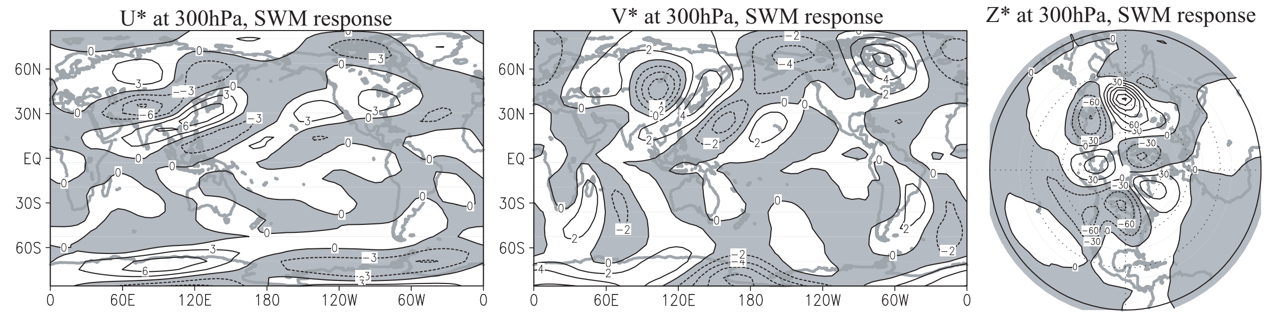


Figure 3.4 Example of the T21L5 steady stationary wave response to the orographic forcing. Shown are the zonal (left) and meridional (middle) winds at 300hPa and geopotential height at 500 hPa (right) all associated with the stationary eddies. Negative values are shaded.

3.4 Accelerated iterative method (AIM)

A simple way for solving the steady response under zonally varying basic state is the time integration (section 3.1). But the time integration is not quite efficient, so that some attempts have been made to solve the steady problems with help of advanced algorithms: for example, the so-called out-of-core solver (Branstator 1992), and parallelization of the LBM code (DeWeaver and Nigam 2000b). We have recently proposed another efficient way, referred to as the accelerated iterative method (AIM), which is shown to be one-order faster than the time integration approach. The mathematical principle is described in Watanabe et al. (2005), so in this document only the practical flows of computation is explained.

AIM is a kind of combined scheme of the relaxation and direct methods. The first part is almost the same as the SWM described in section 3.3, namely, the block matrices for each zonal wavenumber are calculated. After they are inverted by the direct method, the second part of AIM is to correct the first guess of the SWM solution by referring to the non-zonal part of the basic state. The control parameter of the AIM scheme is γ, which is denoted as the acceleration factor. The forced solution is iteratively calculated, and the solution is thought to be converged when the measure of convergence, the normalized differential norm λ, becomes smaller than a specified value.

3.4.1 Making model binary

Choose `'PROJECT = aim'` in `$LNHOME/Lmake.inc`. Although now you can choose higher resolution, let's consider to solve the sample matrix as in 3.2.1, namely, a T21L5 model. For AIM, model option is the conventional one:

```
###### options for model ####################

### dry AIM
MODELOPT = -DOPT_CLASSIC
```

Then, make the model executable file.

```
%> cd $LNHOME/model/src
%> make clean.special
%> make lbm
```

This will create an executable file `lbm2.t21ml5caim`.

3.4.2 Computing linear operator matrix

A sample script for the AIM is `$LNHOME/model/sh/aim/aimbc.l5.classic.csh`. As in the previous sections, let us copy it to `aimbc.l5.test.csh`, and edit lines if necessary. Unlike the previous sections, this shell script is to carry out all the procedures, so that the file names of the linear operator matrix and its inverse are specified here:

```
setenv MATFILE $DIR/mat/MATPWM.t21l5.zm-r.dat        # Linear matrix
setenv MATIFILE $DIR/mat/MATINV.t21l5.zm-r.dat       # Linear inv.  matrix
```

The input data for making the matrix are the 3D basic state, which will be internally truncated at $m = 0$, and topography (only requested for $\sigma \rightarrow$ pressure transform). In preparing the block matrices, set parameters for horizontal diffusion and linear drag in the first block of the script (a part beginning from `=== Making Matrix Ls ===`), and furthermore set

```
setenv OMKMATL   TRUE
setenv OMKMATR   FALSE
setenv OMKINV    FALSE
setenv OAIM      FALSE
```

then run the script

```
%> cd $LNHOME/model/sh/aim
%> aimbc.l5.test.csh
```

It should be noted that the linear operator matrix is not created at this step, instead a group of column vectors is generated.

3.4.3 Acceleration matrix and inverse

To ensure besides achieve faster convergence, we need to add an acceleration matrix to the (SWM) linear operator matrix. This procedure is fast and may need to be repeated several time for tuning the acceleration factor, so separated from the previous step. Namely, modify `aimbc.l5.test.csh` as

```
setenv OMKMATL   FALSE
setenv OMKMATR   TRUE
setenv OMKINV    TRUE
setenv OAIM      FALSE
```

and set the acceleration factor γ

```
setenv RFACT 2000     #
```

where the value of γ should be changed following the diffusion parameters employed in the LBM. For the current example, $\gamma = 2000$ is found to be appropriate. After compiling the routines `redist` and `inv`, run the script again.

```
%> cd $LNHOME/solver/util
%> make clean
%> make
%> cd $LNHOME/solver/custom
%> make clean
%> make
%> cd $LNHOME/model/sh/aim
%> aimbc.l5.test.csh
```

which will make an inverse matrix file spacified by `MATIFILE` after a while.

3.4.4 Preparing forcing

A procedure to prepare forcing is exactly the same as in 3.1.2. Note that the GrADS file of the forcing is only used in the AIM script.

3.4.5 Iteration

The final procedure of AIM is the iterative solver using the inverse matrices and steady forcing. Again, set the environmental variables in `aimbc.l5.test.csh`

```
setenv OMKMATL   FALSE
setenv OMKMATR   FALSE
setenv OMKINV    FALSE
setenv OAIM      TRUE
```

and set the maximum number of iteration and threshold for the differential norm λ,

```
setenv MAXITE 3000
setenv ERRMN 1.d-8
```

Before starting iteration, it is suggested to check whether the parameters `&nmdelt` , `&nmhdif` , `&nmdamp` , `&nmzmfct` , and `&nmvdif` specified in the last block in the script (beginning from `AIM`) are exactly the same as those given in the first block, i.e., used in section 3.4.2. Otherwise, AIM does not lead to the correct solution. The input forcing file and output file of the solution are given as

```
setenv FRCFILE   $FRDIR/frc.t21l5.tst.grd     # forcing
setenv RSPFILT   $FRDIR/aim.t21l5.tst.grd     # response
```

Also, if you would like to compare the AIM solution with the true solution that should be calculated by the direct method in advance, you may specify the files of the true response (input) and the RMS error (output),

```
setenv TRUEFILE   $FRDIR/rsp.t21l5.tst.grd     # true response
setenv RMSEFILE   $FRDIR/rmse.t21l5.tst.grd    # RMS error
```

then set the namelist parameter

```
&nmerr oerr=t ...
```

The default output contains the first guess and the steady soloution (obtained at the last iteration step). If you do not need to have the first guess,

```
&nmrsp frsp='$RSPFILT', oinit=f ...
```

or alternatively if you need to have all the solutions at every iteration step,

```
&nmrsp frsp='$RSPFILT', ofdump=t
```

The output file `$RSPFILT` contains 8 variables (see sample `.ctl` file `sample/aim.t42l20.ctl`):

·	stream function	[m^2 s^{-1}]
·	velocity potential	[m^2 s^{-1}]
·	zonal wind	[m s^{-1}]
·	meridional wind	[m s^{-1}]
·	pressure vertical velocity (ω)	[hPa s^{-1}]
·	temperature	[K]
·	geopotential height	[m]
·	surface pressure	[hPa]

all of which are transformed into pressure corrdinate if you set

```
&nms2p os2p=t
```

in the script `aimbc.l5.tst.csh`.

Now, you are ready to compute the iterative solutions, by running the script again.

```
%> cd $LNHOME/model/sh/aim
%> aimbc.l5.test.csh
```

The resultant log file, `$FRDIR/SYSOUT.aim`, stores the extent to which the solution approaches convergence,

```
===== ITE: 0 DNORM= 845058.153954282 LAMBDA= 9.99E+35
===== ITE: 1 DNORM= 349007.2598010577 LAMBDA= 1.0
===== ITE: 2 DNORM= 247727.48374276334 LAMBDA= 0.7098061051336692
===== ITE: 3 DNORM= 196303.15673701846 LAMBDA= 0.5624615283043564
........
===== ITE: 1413 DNORM= 698.3571292681396 LAMBDA= 2.0009816691670526E-3
===== ITE: 1414 DNORM= 697.9744875512099 LAMBDA= 1.999885297369091E-3
```

where the number at the right indicates λ. If you set the threshold for λ at larger value (`setenv ERRMN 2.d-3` in the above example), you will finally have the message

```
@@@ AIM CONVERGED AT: 1414
```

otherwise,

```
### AIM NOT CONVERGED AT: 3000
```

It should be emphasized, while AIM is much efficient than other iterative methods, that you need to know the best value of γ before starting iteration; in particular, the solution diverges for γ smaller than the appropriate value. Moreover, the system has to be stable or near neutral. When the AIM solution blows up due to either of the above causes, you can step back to section 3.4.3 (NOT section 3.4.2!), and then change γ or damping parameters to repeat the calculation.

One of the most useful application of AIM is to solve a number of steady response to different forcing, with the identical basic state. An example of such a computation is the hindcast of the wintertime circulation anomalies by solving the steady responses to diabatic forcing and transient eddy forcing provided for individual winter. See `aimbc.t42l20.hindcast.csh` is the sample shell script for the hindcast diagnosis, in which additional namelist parameters at the last block are modified:

```
&nmamat ofmult=t,
        nffst=1,
        nflst=43, ...
```

4 Storm Track Model (STM)

4.1 What is STM?

Storm track model (STM) is developed in order to parameterize a statistical property of the midlatitude transient, such as defined by 2–8 day bandpassed components and is often referred to as the storm track, in terms of a function of large–scale stationary fields. The STM provided here is on a basis of the Branstator's (1995) method. That is, eddy covariance statistics like $\overline{u'v'}$ are represented by an ensemble of a large number of linear model integrations given random initial perturbations. Since some of normal modes which are allowed to grow in the linear model will blow up beyond the timescale of baroclinic instability (less than 10 days), each time integration must be stopped by that time. The STM does not require a matrix inversion, implying that the heigher horizontal resolution would be possible unlike the computation of steady linear response (section 3.2).

The basic procedures to make the model binary file and to run the model are similar to the time integration of LBM (section 3.1).

It is noted that we are trying to construct several different types of the new storm track model. They will appear in the forthcoming version, but are currently not available.

4.2 Making model binary

Choose `'PROJECT = tintgr'` in `$LNHOME/Lmake.inc`. Note that the zonal wave truncation and model option in `$LNHOME/Lmake.inc` should be specified as

```
###### zonal wave truncation ################
ZWTRN =
#ZWTRN = m15
#ZWTRN = m10
#ZWTRN = m5

###### options for model ###################

### time-advance linear model (incl.  storm track model)
######## dry model
MODELOPT = -DOPT_CLASSIC
######## moist model
#MODELOPT =
```

Then, enter the following command.

```
%> cd $LNHOME/model/src
%> make clean.special
```

```
%> make lbm
```

It generates the executable file of STM (same name as in section 3.1). Also several post–processor files must be compiled from `$LNHOME/solver/etc/`.

```
%> cd $LNHOME/solver/etc
%> make clean
%> make stm
```

It creates three executable files, `random`, `outstr`, and `chkval`.

4.3 Preparing initial perturbations

As mentioned, STM is run without forcing, but with spatially random, initial perturbations. You thus need to prepare the initial fields. In the previous model package, a huge file of random sequences has been optionally supplied, but in the present version a utility which internally generates a random number with zero-mean, unit variance, and following Gaussian distribution is provided.

At `$LNHOME/solver/etc`, specify the lines in `SETPAR`. Suppose now you use a T21L11 STM,

```
 &nmrdm  crdmg='/home/hiro/ln_solver/data/frc.random.l11.grd',
         nexp=300,
         fact=6.7e-6, 6.7e-7, 6.7, 0.0, 0.0,
 &end
 &nmcls  oclassic=t
 &end
```

where `crdmg` defines the random sequence data which you would like to give to the STM as initial perturbation fields. The file `crdmg` has exactly the same format as the forcing files used in section 3. `nexp` is the total number of samples, corresponding to the ensemble menber, while the amplitude of each perturbation is determined by `fact`, where the first three are only referred for ζ, D, and T. The above number is set at default according to the result of STM experiments (cf. Watanabe and Kimoto 2000). The initial value data (`frc.random.l11.grd` in the above example) can be obtained by

```
%> cd $LNHOME/solver/etc
%> random
```

Since the eddy statistics are obtained from an ensemble, many different initial fields are needed. Therefore the above command creates 1000 fields although you may not need to run the model 1000 times.

4.4 Running STM

A sample shell script to run STM is `$LNHOME/model/sh/tintgr/s.track.csh`. As in 3.1, the environmental variable `BSFILE` in that shell indicates a basic state while `FRC` magnifies initial perturbation but not forcing. In the STM, three variable

`ILNGTH`	length of each integration [days]
`NINTG`	number of ensemble member
`NINTV`	sampling interval [days]

can be altered as tuning parameters. Another tuning parameters for the STM are a strength of the horizontal diffusion and linear drag, both set in a shell script as

```
&nmhdif  order=8, tefold=12, tunit='HOUR'     &end
&nmdamp  ddragv=2,2,2,4,4,-1,-1,-1,-1,-1,-1,
         ddragd=2,2,2,4,4,-1,-1,-1,-1,-1,-1,
         ddragt=2,2,2,4,4,-1,-1,-1,-1,-1,-1,
         tunit='DAY'
&end
```

The meaning of the above parameters has been explained in 3.1.3. Since STM arrows an unstable growth of perturbations, those dissipations are generally set weaker than used in the steady response. Note that the damping coefficient of -1 in `&nmdamp` means that the linear damping is not applied to the level.

Primary products of the STM are 7 quantities (ψ, ζ, u, v, T, Z, and P_s) averaged for the time interval `NINTV`. They are used to compute eddy variance and covariance by a post–processor command `outstr`. NAMELIST near the end of the sample shell file defines parameters for `outstr`.

```
&nmrec   nfcs=$iintg, nfday=2, nlday=$ILNGTH, tint=$NINTV, tunit='DAY'  &end
&nmfili  cfs='$DIR/s',
         cfr='$DIR/r',
         cfu='$DIR/u',
         cfv='$DIR/v',
         cft='$DIR/t',
         cfz='$DIR/z',
         cfp='$DIR/p',
&end
&nmfilo  cfo='$LNHOME/data/out1000.grd',
         cfo2='$LNHOME/data/eddy.stm.grd',
         cbs='$LNHOME/bs/grads/ncepwin.t21l11.grd',
         ofo=f
&end
```

You don't need to change the first set of parameters, `&nmfili`, while you can specify the output filenames which you want in a second set `&nmfilo`. Note that the file `cfo` is a raw eddy fields (not variance/covariance) while `cfo2` is the eddy statistics that will be the final product of STM. Because `cfo` becomes huge, choose in most cases `ofo=f` which prevents to write down `cfo` unless you hope to see the time evolution of simulated baroclinic instability waves.

The eddy statistics defined by `cfo2` are kept on the original σ surface since they are occasionally used to compute a forcing due to eddy feedback (cf. Watanabe and Kimoto 2000). On the other hand, you may want to convert the STM results to pressure–level data in order to compare them with observed data provided on a pressure level. For that purpose, the command `outs2p` can be used after you finished the STM computation. Given the following parameters in `$LNHOME/solver/etc/SETPAR`,

```
&nmfin  cfi='/home/hiro/ln_solver/data/eddy.stm.grd',
        cfo='/home/hiro/ln_solver/data/eddy.stm.pres.grd'
&end
&nmbs   cbs='/home/hiro/ln_solver/bs/grads/ncepwin.t21l11.grd'
&end
```

enter

```
%> cd $LNHOME/solver/etc
%> outs2p
```

The file defined by `cfo` will be the eddy data interporated on the pressure levels.

An example to show the extent to which the STM can reproduce the storm track activity in the CCSR/NIES AGCM is illustrated in Fig. 4.1. Except for the momentum flux is stronger (weaker) in a northern (southern) flank of the storm track, STM reproduces an overall feature of the storm track in the AGCM. While a practical idea to use STM in a midlatitude air–sea interaction issue has been described in Peng and Whitaker (1999) and Watanabe and Kimoto (2000), the STM may also be applied to other problems.

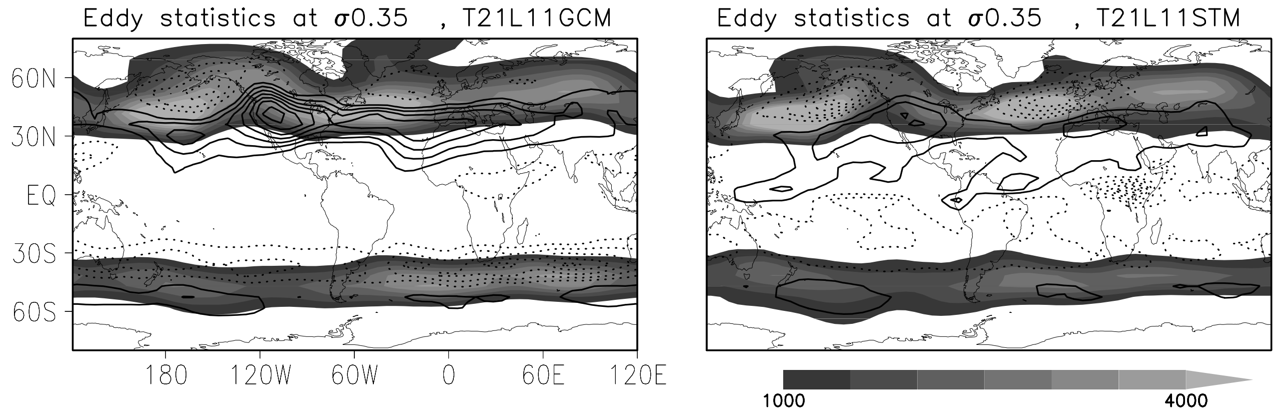


Figure 4.1 Example of the STM simulation. Winter climatology of the transient eddy height variance and momentum flux at $\sigma = 0.35$ in the T21L11 CCSR/NIES GCM (left) and STM with the same resolution (right). The unit of the height variance (shade) is m^2 while the momentum flux (contour) is drawn for every 3 m^2 s^{-2}.

5 Modal decomposition

In this section, we present three methods to detect structure of free modes in the dynamical operator. This is an attempt to seek a preferred mode or a recurrent mode in the atmospheric circulation fields specified as the basic state in LBM. Note that procedures described in this section do not require neither a time integration nor a linear matrix calculation, so that all you have to do is to execute commands in `$LNHOME/solver/`. We assume that you have already prepared a linear operator matrix following 3.2.2 or 3.3.2.

5.1 Preferred mode detection

An easiest way to detect a preferred mode is to compute an ensemble of steady atmospheric response to spatially random forcing. Regarding the ensemble as if it consists of a time sequence of anomaly fields, you can detect a leading structure by a conventional orthogonal expansion such as the EOF analysis. As mathematically derived in A.3, a leading EOF in that analysis approximately corresponds to a leading singular mode which will be obtained in 5.2.

We provide two ways to deal with the above computation: one with a number of forcing which has a spatial δ–function (one–point forcing), and another with a number of spatially random forcing. The former case is discussed in A.3 while the latter case has been carried out by Branstator (1990). In both cases, you first need to have an inverse linear matrix in order to compute multiple responses quickly. For this purpose, compile commands `inv` and `cov` as

```
%> cd $LNHOME/solver/custom
%> make clean
%> make
```

Then, edit `$LNHOME/solver/custom/SETPAR` to specify the file name of matrices. Suppose that you use a T21L5 LBM (not SWM) with zonal waves truncated at 5 as in 3.2.

```
&nmfin  cfm='/home/hiro/ln_solver/matrix/MAT.t21l5.ncepwin.dat',
        cfr='/home/hiro/ln_solver/data/frc.mat',
        cfs='/home/hiro/ln_solver/matrix/MATINV.t21l5.ncepwin.dat',
        cfg='/home/hiro/ln_solver/data/rsp.grd',
&end
&nmall  owall=t
&end
&nmcls  oclassic=t
&end
```

where `cfm` and `cfs` denote a linear matrix data which you already made and an inverse matrix for it, respectively. Files `cfr` and `cfg` are not used at this moment, so that you can specify any dummy file name for them. If you use SWM, modify `&nmall` as `owall=f`. However, most important source of preferred modes is considered to lie in a zonally symmetric component of the basic state, suggesting that your choice of SWM will create a trivial result.

An execution of `inv` will take a certain time after showing a following message.

```
%> inv

### COMPUTE INVERSE MATRIX ###
*********************************************
Read matrix L
   Matrix Size (left)= 3376 x 3376
*********************************************
   LAPACK routine call:  (DGESV)
```

You will obtain the correct inverse matrix `MATINV.t21l5.ncepwin.dat` if a flag `INFO code` is 0 (see 3.2.2).

A forcing file is computed as follows. For on–point forcing data, edit `$LNHOME/solver/custom/SETPAR` as follows.

```
 &nmfin  cfm='/home/hiro/ln_solver/matrix/MAT.t21l5.ncepwin.dat',
         cfr='/home/hiro/ln_solver/data/frc.random1.mat',
         cfs='/home/hiro/ln_solver/matrix/MATINV.t21l5.ncepwin.dat',
         cfg='/home/hiro/ln_solver/data/rsp.grd',
 &end
 &nmvar  ovor=t, odiv=f, otmp=f, ops=f, osph=f
 &end
 &nmfrc  fact=1.0D-8,1.0D-9,1.0D-1,1.0D-9,1.0D-9,
         xlonw=120.,
         xlone=150.,
         ylats=20.,
         ylatn=60.
 &end
 &nmall  owall=t
 &end
 &nmcls  oclassic=t
 &end
```

In `&nmfin`, `cfr` that defines a forcing data is only used. Note that the GrADS forcing data is not created. `&nmvar` specifies which variable is forced, say, in the above example, one–point vorticity forcing is considered. If you want to apply forcing to all the variable, set `odiv, otmp`, and `ops` at `.true.`. The amplitude of forcing is fixed by `fact` while the region where the forcing is given is defined by a boundary west and east longitudes from 0°to 360°(`xlonw` and `xlone`) and a southern and northern boundary latitudes from -90°to 90°(`ylats` and `ylatn`). If you remove these lines, a default set of global coverage will be used. The forcing data `frc.random1.mat` is then obtained by executing `ofrc`.

```
%> ofrc

### MAKE FORCING MATRIX ###
 precision check ..  1.1000000000000
....
Total # forcing:  175
................
Forcing file (output):
Make one-point forcing file
................
grid --> wave (vorticity)
# forcing:  175
................
grid --> wave (divergence)
# forcing:  0
................
grid --> wave (temperature)
# forcing:  0
................
grid --> wave (sfc.  pressure)
# forcing:  0
................
grid --> wave (humidity)
# forcing:  0
................
Written to matrix file (all)
................
### END OF EXECUTION ###
```

It is noted that the total number of forcing is automatically determined with the above parameters. In the above example, you have to compute 175 responses using the forcing data.

Likewise, a forcing file for the second method, namely, a spatially random forcing, is computed as follows. Two parameters are only the difference from the case of one–point forcing in `$LNHOME/solver/custom/SETPAR`.

```
&nmfin  cfm='/home/hiro/ln_solver/matrix/MAT.t21l5.ncepwin.dat',
        cfr='/home/hiro/ln_solver/data/frc.random2.mat',
        cfs='/home/hiro/ln_solver/matrix/MATINV.t21l5.ncepwin.dat',
        cfg='/home/hiro/ln_solver/bs/etc/random1000.l5.grd',
&end
&nmvar  ovor=t, odiv=f, otmp=f, ops=f, osph=f
&end
&nmfrc  fact=1.0D-8,1.0D-9,1.0D-1,1.0D-9,1.0D-9,
        xlonw=120.,
        xlone=150.,
        ylats=20.,
        ylatn=60.
&end
&nmfno  nftype=100
&end
&nmall  owall=t
&end
&nmcls  oclassic=t
&end
```

In `&nmfin`, you can use the same name of `cfr` as in one–point forcing. You should now specify a pre–supplied random sequence data (see 4.3) in `cfg`. Also, unlike `ofrc`, you have to fix the total number of forcing (0–1000) by `nftype`. Execute `rfrc` to obtain the random forcing data.

```
%> rfrc

### MAKE FORCING MATRIX ###
Random sequence (input) : /home/hiro/ln_solver/bs/etc/random1000.l5.grd
Forcing file (output):
/home/hiro/ln_solver/data/frc.random2.mat
................
Total # forcing:  100
................
 precision check ..  1.1000000000000
....
Make random forcing file
................
# case:  1
# case:  2
....
# case:  100
```

```
Written to matrix file (all)
................
### END OF EXECUTION ###
```

The responses are then easily computed using the inverse matrix, one–point (or random) forcing, and a command `cov`. Edit `$LNHOME/solver/custom/SETPAR` as

```
&nmfin  cfm='/home/hiro/ln_solver/matrix/MAT.t21l5.ncepwin.dat',
        cfr='/home/hiro/ln_solver/data/frc.random1.mat',
        cfs='/home/hiro/ln_solver/matrix/MATINV.t21l5.ncepwin.dat',
        cfg='/home/hiro/ln_solver/data/rsp.random.t21l5.classic.grd',
&end
&nmbs   cbs='/home/hiro/ln_solver/bs/grads/ncepwin.t21l5.grd'
&end
&nmuv   o2uv=t, opl=t
&end
&nmfno  nftype=175
&end
&nmall  owall=t
&end
&nmcls  oclassic=t
&end
```

where `cfg` is a file name of the response data while `cfm` is not used. You have to give a total number of ensemble by `nftype`.

Then,

```
%> cov
```

will create the ensemble (175 members in the above example) response fields. Although we do not support further analysis to that response data, you will readily compute EOFs for the responses if you are familiar with the observational data analyses.

5.2 Eigenvalue analysis

Eigenanalysis has often been performed to identify normal modes of the atmosphere. It is typically based on a barotropic vorticity equation (Branstator 1985b), which will be described elsewhere (cf. 8.2). In this section, the eigensolver is applied to the multi-level model operator.

Eigenmodes can be computed both for the LBM and SWM operators as in sections 3.2 and 3.3, but the former is generally hard to solve due to huge size of the matrix. Here, suppose computing the eigen functions for T21L20 SWM. First, compile commands `eof` and `etrn` in `$LNHOME/solver/expert`.

```
%> cd $LNHOME/solver/expert
%> make clean
%> make
```

You may see that the compiler sais some messages like

```
eof.f:
     MAIN eof:
        "eof.f", line 40:  Error:  array "a" has too many elements
        "eof.f", line 40:  Warning:  overflow in expression
```

This happens due to insufficient memory of your computer (see also B.3). Since the dimension in the program has been kept for LBM, you can reduce it in case of SWM by modifying the top part of `eof.f` as

```
*
*     standard
CC      PARAMETER ( MAXN=2*NMAX*NTR*(NVAR*KMAX+1) )
*
*     reduced memory
      PARAMETER ( MAXN=2*NMAX*(NVAR*KMAX+1) )
```

then re–compile the programs. You will obtain `eof` and `etrn` under `$LNHOME/solver/expert`.

Before executing them, edit `$LNHOME/solver/expert/SETPAR`.

```
&nmefin  cfm='/home/hiro/ln_solver/matrix/MAT.t21l20.ncepwinzm.dat',
         cfe='/home/hiro/ln_solver/data/evalswm.t21l20.grd',
         cfs='/home/hiro/ln_solver/data/evec.dat',
         cfg='/home/hiro/ln_solver/data/evecswm.t21l20.grd'
&end
&nmbs    cbs='/home/hiro/ln_solver/bs/grads/ncepwin.t21l20zm.grd'
&end
&nmuv    o2uv=t, opl=t
&end
&nmwave  nwave=0
&end
&nmmod   modmin=1, modmax=1281
&end
&nmall   owall=f
&end
&nmcls   oclassic=t
&end
```

where `cfe` and `cfg` are GrADS files which contain complex eigenvalues and associated eigenvectors, respectively. The file `cfs` is a temporary data of spherical coefficient for `cfg`. These outputs will be made only for modes from `modmin` to `modmax` defined by `&nmmod`. For SWM, the eigenmodes are computed for the matrix with a specific zonal wave number, that is specified by `&nmwave`. If you compute eigenmodes of LBM, this parameter is ignored and instead you must specify `&nmall owall=t`. The above example is to calculate eigenmodes for the zonal wave number 0 of SWM, i.e., eigenfunctions for the axisymmetric atmosphere.

After you edit `SETPAR`, enter

```
%> eof ; etrn
```

It will produce files of `cfe` and `cfg` after a while. Sample `.ctl` files for the eigenvalues and eigenvectors are found in `$LNHOME/sample`.

5.3 Singular vector analysis

A singular vector analysis using the singular value decomposition (SVD) is essentially similar to the eigenanalysis in the previous section, but has several advantages over it. The theory of singular mode and mathematical expressions are given by Navarra (1993), Itoh and Kimoto (1999), Kimoto et al. (2001), and also in A.2.

As in 5.2, singular modes can be computed both for the LBM and SWM operators. Here, again suppose carrying out SVD for T21L20 SWM. Executable commands `svd` and `strn` are compiled at `$LNHOME/solver/expert`.

```
%> cd $LNHOME/solver/expert
%> make clean
%> make
```

See previous section if you are troubled with the error due to a memory problem.

Then edit `$LNHOME/solver/expert/SETPAR`.

```
 &nmsfin  cfm='/home/hiro/ln_solver/matrix/MAT.t21l20.ncepwinzm.dat',
          cfe='/home/hiro/ln_solver/data/svalswm.t21l20.grd',
          cfl='/home/hiro/ln_solver/data/svec-u.dat',
          cfr='/home/hiro/ln_solver/data/svec-v.dat',
          cflg='/home/hiro/ln_solver/data/svecswm.t21l20.u1.grd',
          cfrg='/home/hiro/ln_solver/data/svecswm.t21l20.v1.grd'
 &end
 &nmbs    cbs='/home/hiro/ln_solver/bs/grads/ncepwin.t21l20zm.grd'
 &end
 &nmuv    o2uv=t, opl=t
 &end
```

```
&nmwave  nwave=0
&end
&nmmod   modmin=1281, modmax=1281
&end
&nmall   owall=f
&end
&nmcls   oclassic=t
&end
```

They are different from parameters for eigenanalysis only for input/output files specification, namely, `&nmsfin`. Since the SVD generates two sets of orthogonal functions (left and right singular vectors, see A.2), the output GrADS files for singular vectors are specified by two parameters of `cflg` and `cfrg` which defines left and right vectors, respectively. Other parameters are defined in the same way as the eigensolver. The above example is to calculate eigenmodes for the zonal wave number 0 of SWM that has a rank of 1281, but the last mode is only written down to the GrADS file (`modmin=1281, modmax=1281`). Since the modes are internally sorted in order of the singular values, the 1281th mode represents a singular mode with the smallest singular value. You will know the implication of this mode by referring to Navarra (1993) or Kimoto et al. (2001).

After you edit `SETPAR`, enter

```
%> svd ; strn
```

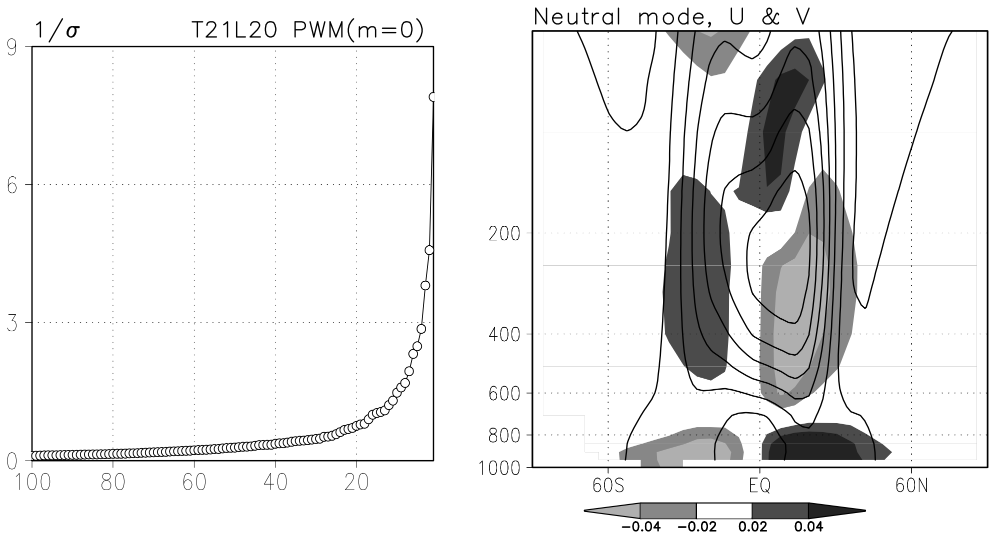


Figure 5.1 Example of singular vector analysis to zonal mean components of the T21L20 SWM. Inverse singular value for leading 100 modes (left) and structure of the leading mode (u: contour, v: shading). The horizontal (vertical) axis of the left panel corresponds to number of mode (time scale of unit 1×10^{11} s^{-1}).

It will produce files of `cfe` and `cfg` after a while. Sample `.ctl` files for the singular values and singular vectors are found in `$LNHOME/sample`. An example of the singular mode computation is shown in Fig. 5.1, that illustrates a singular value spectrum and the neutral mode structure for the axisymmetric atmosphere. See Watanabe et al. (2002) for the interpretation of the neutral mode.

6 Nonlinear dynamical model

The nonlinear dynamical model has been incorporated into the LBM package from version 2.0, but it has an extended option in the present version. The nonlinear atmospheric model, which is often called the dynamical core, is a useful tool to address processes responsible for mean climate (cf. Hoskins and Rodwell 1995; Rodwell and Hoskins 1995) as well as to diagnose an anomalous response to steady forcing (Jin and Hoskins 1995). The model is also used to compare results of LBM in order to assess an importance of nonlinearity (e.g. Ting and Yu 1998).

The nonlinear model developed here is nearly identical to the dynamics of the CCSR/NIES AGCM, furthermore, the Mellor-Yamada turbulent mixing and surface Bulk fluxes are incorplated to the momentum equation with an implicit scheme. While the variables are not decomposed into mean and anomaly unlike LBM, either total or anomaly (defined as deviations from initial fields) can be obtained as output. The nonlinear model does not include an interative moist process, but will do in the forthcoming version.

An important procedure of the nonlinear dry model is to prepare forcing that ensures the model equilibrium state remaining close to the basic state. Two different methods are provided: one is to calculate the residual diabatic heating using the model (section 6.1), and another is to apply temperature restoring to a simple (e.g. convective–radiative) equilibrium profile (section 6.2). The residual forcing in the former way, denoted as R, is obtained by integrating the nonlinear operator one step forward. An example of the nonlinear response to R calculated with respect to the summer (JJA) climatological state is shown in Fig. 6.1. The Asian monsoon circulation over the Indian Ocean and anticyclonic circulation associated with subtropical highs are both well reproduced. The structure of R is presented in Fig. 6.2, which reveals the residual heating is noisier than we anticipate. While the calculation with R is not very bad, it is more desirable to calculate the heating as in the conventional Q1 by yourself and then give it to the model.

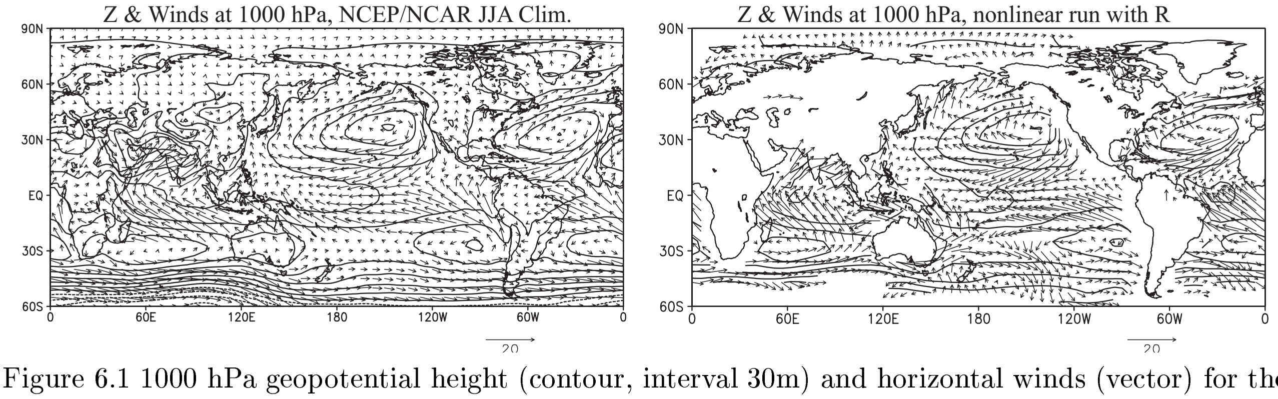


Figure 6.1 1000 hPa geopotential height (contour, interval 30m) and horizontal winds (vector) for the NCEP/NCAR JJA climatology (left) and the results of the T21L20 nonlinear integration averaged for days 21–30 (right). The forcing R to compute the nonlinear response is shown in Fig. 6.2. Note that the integration is initialized by the JJA zonal-mean state.

6.1 Nonlinear run with the residual forcing R

6.1.1 Preparing R

The Time integration of the nonlinear model is carried out in a similar manner to the LBM (cf. section 3.1) except that the prescribed forcing R is first computed.

Choose `'PROJECT = dcore'` in `$LNHOME/Lmake.inc`. Note that the zonal wave truncation and model option in `$LNHOME/Lmake.inc` should be specified as

```
###### zonal wave truncation ################
ZWTRN =
#ZWTRN = m15
#ZWTRN = m10
#ZWTRN = m5

###### options for model ####################

### nonlinear dynamical core
#MODELOPT =
MODELOPT = -DOPT_RWRIT
```

Then, enter compile

```
%> cd $LNHOME/model/src
%> make clean.special
%> make lbm
```

Suppose now you selected the horizontal resolution of T21 and vertical 20 levels on Sun WS, the above procedure will generate an executable file `lbm2.t21ml20cdcore` under the directory `$LNHOME/model/bin/sun/`. This is the executable file to make R. Note that you need to remake the model executable when running the nonlinear model later.

A sample script, `$LNHOME/model/sh/dcore/nonlinear-run.csh` has been provided. It can be used both to run the model and to compute R. Copy it to any name, say, `nonlinear-run.test.csh`, and edit suitably.

```
%> cd $LNHOME/model/sh/dcore
%> cp nonlinear-run.csh nonlinear-run.test.csh
%> chmod u+x nonlinear-run.test.csh
```

Many of environmental variables defined in the script are the same as in section 3.1.3, namely,

`LNHOME`	home directory of the model package
`SYSTEM`	architecture
`RUN`	model executable file
`FDIR`	directory which contains forcing data
`DIR`	directory for first model products (Gtool format)
`INITFILE`	initial state (Gtool formatted data)
`SFRC`	steady forcing (if necessary)
`TEND`	length of time integration in day

but additionally,

`DATZ`	topography (Gtool format)
`RFRC`	constant forcing R (GrADS format)

should be specified. Here suppose to simulate summer circulation as in Fig. 6.1, so that define NCEP summer basic state, say `ncepsum.t21l20`, as `INITFILE`.

The forcing R specified as `RFRC` is readily obtained by

```
%> cd $LNHOME/model/sh/dcore
%> nonlinear-run.test.csh
```

The forcing data has the same format as anomalous steady forcing made in 3.1.2, therefore you can look at the data by referring to a sample `.ctl` file `$LNHOME/sample/frc.t21l20.classic.ctl`. It is noted that the residual forcing is only considered for temperature field. If you would like to give the forcing to other fields, additional namelist parameter should be given in the script

```
&nmrwrit ORVOR=t, ORDIV=t, ORT=t, ORPI=t
```

A procedure to make steady anomalous forcing is the same as in 3.1.2. In the nonlinear model, the steady *anomalous* forcing is not necessarily applied.

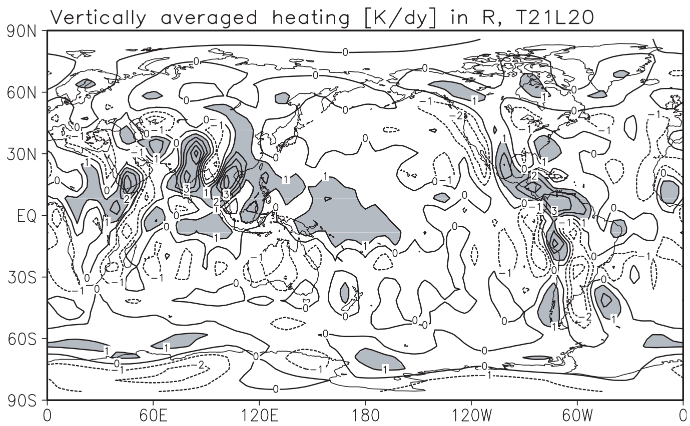


Figure 6.2 Example of the residual thermal forcing in the T21L20 nonlinear model, as shown by the vertical average, with the contour interval of 1 K/dy. The heating greater than 1 K/dy is shaded.

6.1.2 Running the model

Given R, and steady anomalous forcing if necessary, you can run the nonlinear model using the same script `nonlinear-run.test.csh`. Before running the model, you need to remake the model executable file.

Edit again options in `$LNHOME/Lmake.inc` as

```
###### options for model ###################

### nonlinear dynamical core
MODELOPT =
#MODELOPT = -DOPT_RWRIT
```

then, compile the source code.

```
%> cd $LNHOME/model/src
%> make clean.special
%> make lbm
```

In the script `nonlinear-run.test.csh`, the variable `RFRC` is now input but not output. Also you can change the initial state `INITFILE` arbitrarily; ordinally the initial field is given by the zonal mean state of the climatology (see Hoskins and Rodwell 1995), so that let us specify `ncepsun.t21l20zm` as `INITFILE` (see section 2.4 for how to make the zonal–mean basic state). If you want to obtain anomaly fields, as defined by difference from initial state, modify a parameter in the script file as

```
    &nmanm  oanm=t  &end
```

You can now run the nonlinear model as

```
%> cd $LNHOME/model/sh/dcore
%> nonlinear-run.test.csh
```

All the log message to the standard output is written down to a file `$LNHOME/$DIR/SYSOUT` (see the script file). You will know the current model step by looking at the end line of `SYSOUT`.

6.2 Nonlinear run with the temperature restoring

Another way of running the nonlinear model is to restore temperature field to an equilibrium state given externally, but not to prescribe the heating. The given reference temperature is practically arbitrary, but should be one of the stationary solution of the system from the physical point of view.

A sample script, `$LNHOME/model/sh/dcore/nonlinear-run.restore.csh` has been provided. It can be used to run the model with the reference state. Copy it to any name, say, `nonlinear-run.test2.csh`, and edit suitably.

```
%> cd $LNHOME/model/sh/dcore
%> cp nonlinear-run.restore.csh nonlinear-run.test2.csh
%> chmod u+x nonlinear-run.test2.csh
```

The file of the reference state should have the same format as the basic state. In the above example, the summer climatology is used as reference:

```
setenv SFRC $LNHOME/bs/grads/ncepsum.t21l20.grd  # restoring state
```

where temperature is only used in the model. The timescale of restoration at each σ level is determined by the namelist parameter `nmret`.

```
&nmret drstt=
2.,2.,2.5,3.,4.5,
6.,8.,10.,15.,15.,
15.,15.,15.,15.,15.,
15.,15.,15.,15.,15., tunit='DAY'
&end
```

The above timescale is subject to tuning, so that you can change and search the appropriate parameter. With the above set of parameters, the nonlinear model produces the summer circulation similar to that shown in Fig. 6.1 (right) but with smaller magnitude. The restoring may also be suitable for idealized simulations, such as to investigate the baroclinic adjustment (example shown in Fig. 6.3).

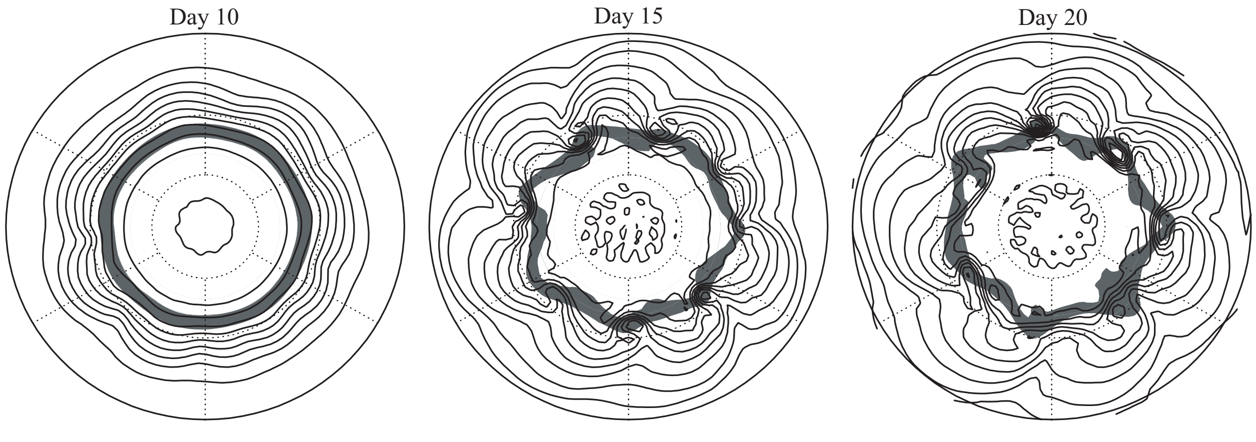


Figure 6.3 Example of the baroclinic adjustment experiment using the T42L20 nonlinear model with restoring temperature to the radiative–convective equilibrium state. Shown are the surface potential temperatuere (contour) and potential vorticity at θ = 330 K (shade) at day 10, 15, and 20, respectively.

6.3 Post process and visualize

A post process that combine Gtool data files to one GrADS file is exactly the same as in section 3.1.4. Namely, edit lines in `$LNHOME/solver/util/SETPAR` and perform `gt2gr`.

```
%> cd $LNHOME/solver/util
%> make clean ; make (if necessary)
%> gt2gr
```

7 Moist LBM

So far, most of linear baroclinic model simulation had employed a prescribed forcing, which was anomalous diabatic heating in many cases, and has been assumed steady in time. When one argues a midlatitude response to a remote, equatorial heating, this assumption is justified since cumulus convectionc which result in the anomalous heating do not, or less, interact with the circulation response. However, if one considers a response within the tropics, the assumption may not be valid any more. This was the reason that previous simple model studies for CISK (e.g. Hayashi 1971; Stevens and Lindzen 1978) have dealt with an interaction between the dynamics and convection. Ting (1991) also tried to parameterize a heating as a function of temperature response (assuming constant relative humidity), although the scheme is not sufficient to discuss a full coupling between dynamics and convection.

An attempt to incorporate an interactive convection into LBM with an explicit treatment of atmospheric moisture is relatively new approach and fewly conducted, except for a mean climate simulation by Seager and Zebiak (1995). In this section, we describe the LBM which include an interactive convection based on the Betts–Miller scheme (Betts 1986; Betts and Miller 1986) which has been linearized by Neelin and Yu (1994) and Yu and Neelin (1994). Several modifications have been applied to their scheme as detailed by Watanabe and Jin (2003) and referred it to as the moist LBM (mLBM). An example to apply the mLBM to climate study can be found in Watanabe and Jin (2002, 2003).

The present version is also include the large-scale condensation (LSC) process as represented by a linear scheme similar to the convective adjustment.

7.1 Time integration

The time integration of the mLBM can be carried out in a similar manner to that of dry model (cf. section 3.1), except that the prescribed forcing is SST anomaly but not heating any more. The output file contains the following 14 quantities (you cannot add or remove the variable).

· stream function	[m^2 s^{-1}]
· velocity potential	[m^2 s^{-1}]
· zonal wind	[m s^{-1}]
· meridional wind	[m s^{-1}]
· pressure vertical velocity (ω)	[hPa s^{-1}]
· temperature	[K]
· geopotential height	[m]
· surface pressure	[hPa]
· specific humidity	[kg kg^{-1}]

·	convective heat source Q1c	[K s^{-1}]
·	convective moisture source -Q2c	[kg kg^{-1} s^{-1}]
·	LSC heat source Q1l	[K s^{-1}]
·	LSC moisture source -Q2l	[kg kg^{-1} s^{-1}]
·	precipitation	[mm/day]

For the computation, there may be five steps, as explained below.

7.1.1 Making model binary

Choose `'PROJECT = tintgr'` in `$LNHOME/Lmake.inc`. Note that the zonal wave truncation and model option in `$LNHOME/Lmake.inc` should be specified as

```
###### zonal wave truncation ################
ZWTRN =
#ZWTRN = m15
#ZWTRN = m10
#ZWTRN = m5

###### options for model ###################

### time-advance linear model (incl.  storm track model)
######## dry model
#MODELOPT = -DOPT_CLASSIC
######## moist model
MODELOPT =
```

Then, enter the following command.

```
%> cd $LNHOME/model/src
%> make clean.special
%> make lbm
```

Suppose now you selected the horizontal resolution of T21 and vertical 20 levels on Sun WS, and the above procedure will generate an executable file `lbm2.t21ml20ctintgr` under the directory `$LNHOME/model/bin/sun/`. This is the executable file to run mLBM in time.

7.1.2 Preparing basic state SST

In the mLBM, mean fields for sea surface temperature (SST) and ground wetness (W_g) are also prescribed as basic states. These data are actually prepared in a separate file from the standard basic state data that contain atmospheric variables. Both SST and W_g are used to compute anomalous surface heat fluxes in the moist model (W_g is converted to the evaporative efficiency over the land surface).

While winter basic states for T21 model have been provided by the package, named `sstwin.t21` and `wgwin.t21` in `$LNHOME/bs/gt3/`, you may want to apply the mLBM in other season or month. For this purpose, we supply a command `ncep1vbs` to make a basic state for SST and W_g using observed climatology data packed in a data archive `ln_solver2.0.ncepdata.tar.gz` (see section 2.4 for unpacking the archive).

First, edit following lines in `$LNHOME/solver/util/SETPAR` to specify file names and season or month.

```
&nmncp  cncep='/home/hiro/ln_solver/bs/ncep/sst.clim.t21.grd',
        kmo=6, navg=3, ozm=f, osw=f, cvar='SST'
&end
&nmbs   cbs0='/home/hiro/ln_solver/bs/gt3/sstsum.t21',
        cbs='/home/hiro/ln_solver/bs/grads/sstsum.t21.grd'
&end
```

then compile and execute the command as

```
%> cd $LNHOME/solver/util
%> make clean
%> make
%> ncep1vbs
```

The above example makes summer (JJA) SST basic state `sstsum.t21`. The wetness data can be calculated in the same way except for the above parameters modified as

```
&nmncp  cncep='/home/hiro/ln_solver/bs/ncep/wg.clim.t21.grd',
        kmo=6, navg=3, ozm=f, osw=f, cvar='WG'
&end
&nmbs   cbs0='/home/hiro/ln_solver/bs/gt3/wgsum.t21',
        cbs='/home/hiro/ln_solver/bs/grads/wgsum.t21.grd'
&end
```

Check whether the SST and W_g data are made correctly by looking at the corresponding GrADS data, `sstsum.t21.grd` and `wgsum.t21.grd`. The GrADS `.ctl` files can be copied from those for winter that have been prepared in advance. Note that the SST basic state over land has been filled by the NCEP–derived skin temperature climatology.

7.1.3 Preparing forcing

Before running moist LBM, you have to prepare a GrADS file of initial perturbation or steady SST forcing data. Note that both data have a different format unlike the conventional LBM. A procedure to prepare an initial perturbations has been explained in section 3.1.2. To make a simplified steady SST forcing, you can use a tool `mkfrcsst`. The simple forcing data can be created after you edit the parameter file `$LNHOME/solver/util/SETPAR`. The parameters as specified below are similar to that for `mkfrcng` (see section 3.1.2) but here you only need to define the horizontal profile and the forcing function over land is automatically removed by using a land/sea grid index `cfm`.

```
&nmfin  cfm='/home/hiro/ln_solver/bs/gt3/gridx.t21',
        cfg='/home/hiro/ln_solver/sample/frcsst.t21.grd'
&end
&nmhpr  khpr=1,
        hamp=2.,
        xdil=65.,
        ydil=15.,
        xcnt=240.,
        ycnt=0.
&end
```

The file defined by `cfg` is the GrADS forcing data while `cfm` specifies grid index file that has been prepared in the package. Note that the amplitude `hamp` now has a unit of K. You will find a sample forcing file as defined by the above parameters in `$LNHOME/sample/`, which imitates a typical El Niño condition.

To make a forcing that you wish to have, compile and execute the command `mkfrcsst` as

```
%> cd $LNHOME/solver/util
%> make clean
%> make
%> mkfrcsst
```

In `$LNHOME/sample`, you can find another sample SST forcing file `frcsst.nino.t21.grd`. This is made from observed SST composite for El Niño (difference between 5 El Niños and 6 La Niñas) for tropical band of 20°S–20°N. The LBM package currently does not support a tool to make such a realistic forcing data.

7.1.4 Running the model

`$LNHOME/model/sh/tintgr/linear-run.csh` is a sample shell script for the time integration of the mLBM. Copy it to any name, say, `linear-run.moist-test.csh`, and edit suitably.

```
%> cd $LNHOME/model/sh/tintgr
%> cp linear-run.csh linear-run.moist-test.csh
%> chmod u+x linear-run.moist-test.csh
```

Many of environmental variables defined in the script are the same as in section 3.1.3, namely,

`LNHOME`	home directory of the model package
`SYSTEM`	architecture
`RUN`	model executable file
`FDIR`	directory which contains forcing data
`DIR`	directory for first model products (Gtool format)
`BSFILE`	basic state (Gtool formatted data)
`FRC`	initial perturbation (if necessary)
`SFRC`	steady forcing (if necessary)
`TEND`	length of time integration in day

but additionally,

`DATZ`	topography (Gtool format)
`DATS`	basic state SST (Gtool format)
`DATW`	basic state soil wetness (Gtool format)
`DATI`	surface index (Gtool format)

are specified.

If you choose to give a steady forcing, `FRC` is not used (perhaps you can specify the same file as `SFRC` as dummy) whereas to give an initial perturbation, vice versa. The duration of the model integration is generally set less than a month (see section 3.1.3 for more detail).

The parameters which define strength of the dissipation terms such as the horizontal and vertical diffusion, Rayleigh friction, and Newtonian damping, are similarly specified following the explanation in the dry model (section 3.1.3). Note that the diffusion terms are also applied to the moisture equation to remove a small–scale noise while no drag is included except for the planetary boundary layer since there is no physical ingredient that represents such a damping process. Parameters which control the cumulus parameterization and surface fluxes are denoted as `&nmmca` and `&nmsfcm`, respectively.

```
&nmmca   ocum=t, ttau0=3, qtau0=3, ttauc=6, tunit='HOUR', sigkb=0.9D0  &end
&nmsfcm  owes=t, expw=0.5                                              &end
```

where `ttau0` and `qtau0` defines a relaxation time (with the time unit `tunit`) due to convective adjustment while `sigkb` determines a top of the cloud bottom layer. If you set `owes=f`, a wind–evaporation feedback term is turned off (do not change `expw`).

Similarly, parameters for the LSC process are

```
&nmlsc  olsc=t, ttaul0=2, qtaul0=2, tunit='HOUR', dqrat=1.D-2  &end
```

where `ttaul0` and `qtaul0` are the relaxation time for temperature and specific humidity. When you would like ot turn off either the convection or LSC, set

```
&nmmca  ocum=f, ...&end
```

or

```
&nmlsc  olsc=f, ...&end
```

You can now run the linear model as

```
%> cd $LNHOME/model/sh/tintgr
%> linear-run.moist-test.csh
```

All the log message to the standard output is written down to a file `$LNHOME/$DIR/SYSOUT` (see `linear-run.moist-test.csh`). You will know the current model step by looking at the end line of `SYSOUT`.

7.1.5 Post process and visualize

The outputs for the model run (specified as `&nmhist` in `$LNHOME/model/sh/tintgr/linear-run.csh`) consist of 14 variables with the Gtool format. As in section 3.1.4, they are incorporated into one GrADS file by using a tool `$LNHOME/solver/util/gt2gr`. Specify the following parameters or filenames in `$LNHOME/solver/util/SETPAR`:

```
&nmfgt  cfs='/home/hiro/ln_solver/data/out/psi',
        cfc='/home/hiro/ln_solver/data/out/chi',
        cfu='/home/hiro/ln_solver/data/out/u',
        cfv='/home/hiro/ln_solver/data/out/v',
        cfw='/home/hiro/ln_solver/data/out/w',
        cft='/home/hiro/ln_solver/data/out/t',
        cfz='/home/hiro/ln_solver/data/out/z',
        cfp='/home/hiro/ln_solver/data/out/p',
        cfq='/home/hiro/ln_solver/data/out/q',
        cftc='/home/hiro/ln_solver/data/out/dtc',
        cfqc='/home/hiro/ln_solver/data/out/dqc',
        cftl='/home/hiro/ln_solver/data/out/dtl',
        cfql='/home/hiro/ln_solver/data/out/dql',
        cfpr='/home/hiro/ln_solver/data/out/pr',
        cfo='/home/hiro/ln_solver/data/tintgr/linear.t21l20.grd',
        fact=1.,1.,1.,1.,1.,1.,1.,1.,1.,1.,1.,1.,1.,1.
        opl=t,
&end
&nmbs   cbs0='/home/hiro/ln_solver/bs/gt3/ncepwin.t21l20',
        cbs='/home/hiro/ln_solver/bs/grads/ncepwin.t21l20.grd'
&end
&nmcls  oclassic=f
&end
```

See section 3.1.4 for the description of parameters. Only the difference from the dry model is to set `oclassic=f`. After you modified `$LNHOME/solver/util/SETPAR` as above, enter

```
%> cd $LNHOME/solver/util
%> gt2gr
```

or, if you skipped the procedure to make forcing (section 3.1.2),

```
%> cd $LNHOME/solver/util
%> make clean
%> make
%> gt2gr
```

A sample control file `linear.t21l20.ctl` is found in `$LNHOME/sample/`. You can copy it and modify the filename written in the `.ctl` file, so that the result can be drawn on global map using GrADS. A sample response at day 15 is shown in Fig. 7.1 which compares a 500 hPa height response obatined in the conventional dry model and the current moist model. The heating internally computed in the moist model looks similar to the observed OLR–based heating in the dry model except for a slight overestimate of the magnitude. Besides, the response over the North Pacific is located in more realistic position in the moist LBM.

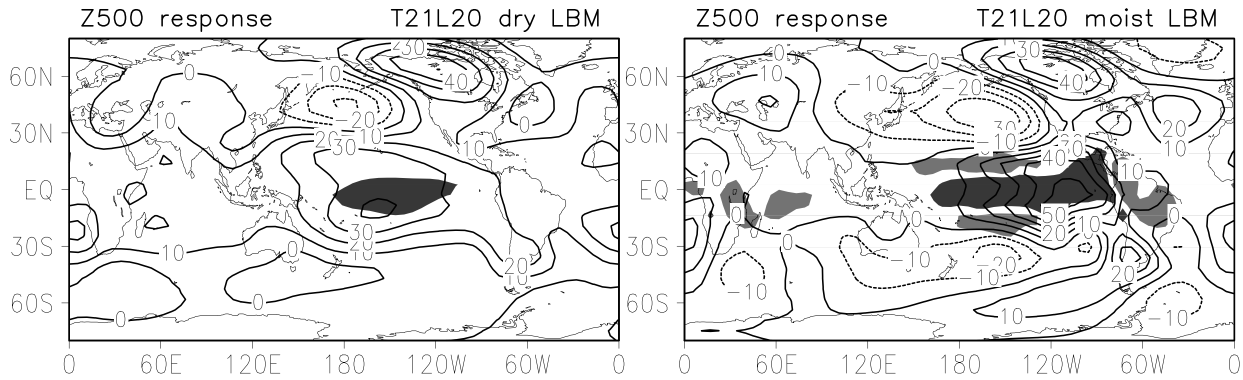


Figure 7.1 Comparison of dry and moist linear model integrations as revealed in a 500 hPa height response at day 15. Conventional dry model (left) and moist model (right). The contour interval is 10 m while light (dark) shading denotes a vertically averaged heating is less than -1 (greater than 1) K day−1. The heating and SST anomaly in dry and moist models are derived from observed composite of OLR and SST for EL Niño winters.

7.2 Steady response

A steady linear response with interacting moist processes using the direct method can also be obtained. Overall procedure is similar to the dry LBM and SWM (sections 3.2 and 3.3), but there may be several differences as explained below.

7.2.1 Making model binary

Choose 'PROJECT = mkamat' in $LNHOME/Lmake.inc. Since a linearized cumulus scheme in this mLBM has been designed as the heating being a function of boundary layer moist energy (cf. Watanabe and Jin 2003), the model must have at least one level below the boundary layer top (typically $\sigma = 0.90$). The 5–level model as exemplified in section 3.2 which does not resolve the boundary layer is not appropriate for the moist model. Therefore, suppose using a secondary small matrix of T21L8 which truncates zonal wave number at m =5, i.e., edit $LNHOME/Lmake.inc as follows.

```
###### Vertical Resolution ################
#VRES = l1
#VRES = l5
VRES = l8
#VRES = l11
#VRES = l20
###### zonal wave truncation ################
#ZWTRN =
#ZWTRN = m15
#ZWTRN = m10
ZWTRN = m5

###### options for model ####################

### time-advance linear model (incl.  storm track model)
######## dry model
#MODELOPT = -DOPT_CLASSIC
######## moist model
#MODELOPT =

### lbm/pwm
######## dry model
#MODELOPT = -DOPT_MKMAT -DOPT_CLASSIC
#MODELOPT = -DOPT_MKMAT -DOPT_OWALL -DOPT_CLASSIC
#MODELOPT = -DOPT_WVFRC -DOPT_ORHS -DOPT_CLASSIC
######## moist model
#MODELOPT = -DOPT_MKMAT
MODELOPT = -DOPT_MKMAT -DOPT_OWALL
#MODELOPT = -DOPT_WVFRC -DOPT_ORHS
```

If you employ zonally uniform basic state, the model can be reduced to SWM (see section 3.3) by specifying MODELOPT = -DOPT_MKMAT. To compile the model, enter the following command.

```
%> cd $LNHOME/model/src
```

```
%> make clean.special
%> make lbm
```

This will create an executable file `lbm2.t21ml8cmkamat`. If you have not made the model library `liblbm2t21ml8c.a`, compile it altogether.

```
%> cd $LNHOME/model/src
%> make clean
%> make lib lbm
```

7.2.2 Preparing basic state and anomalous SST

The moist model is forced by SST anomaly but not by diabatic heating because the heating is internally generated in the model. As in the time integration, therefore, SST data for both basic state and anomaly are required to run the model. See section 7.1.2 for how to make those files.

You also need to prepare a basic state for T21L8 model. Use the command `ncepsbs` in `$LNHOME/solver/util` (see section 2.4).

7.2.3 Computing linear operator matrix

`$LNHOME/model/sh/mkamat/mkamat.l8.csh` is a sample shell script for making a linear operator matrix. Copy it to any name, say, `mkamat.l8.test.csh`, and edit lines if necessary.

```
%> cd $LNHOME/model/sh/mkamat
%> cp mkamat.l8.csh mkamat.l8.test.csh
%> chmod u+x mkamat.l8.test.csh
```

The environmental variables and parameters defined in the script have already been described in sections 3.2.2 and 7.1.4. At the same time, specify the file name of the linear operator in `$LNHOME/solver/util/SETPAR`,

```
&nmred  cdr='/home/hiro/ln_solver/matrix',
        cfo='/home/hiro/ln_solver/matrix/MAT.t21l8.ncepwin.dat'
&end
&nmall  owall=t
&end
&nmcls  oclassic=f
&end
```

where `cfo` defines the matrix file name while `cdr` specifies a directory for temporary files of the column vectors. Before running `mkamat.l8.test.csh`, you might need to compile a command `redist` under `$LNHOME/solver/util/`, which refers to the above parameters.

```
%> cd $LNHOME/solver/util
%> make clean
%> make
```

The residual method performed by the above script repeat N–times one–step integration of the linear model. In the current example, our LBM actually repeats 5 (zonal wave number) $\times$ (4 (ζ, D, T, q) $\times$ 8 (levels) + 1 (P_s)) times loop in the script `mkamat.l8.test.csh`. Thus when you run the script, you will see a message on the terminal window like

```
%> cd $LNHOME/model/sh/mkamat
%> mkamat.l8.test.csh
VOR zonal wave M is 0 and LEVEL is 1
VOR zonal wave M is 1 and LEVEL is 1
VOR zonal wave M is 2 and LEVEL is 1
VOR zonal wave M is 3 and LEVEL is 1
VOR zonal wave M is 4 and LEVEL is 1
..............
SPH zonal wave M is 4 and LEVEL is 8
SPH zonal wave M is 5 and LEVEL is 8
```

Each line represents an execution of LBM that yields a set of column–vector data such as `$LNHOME/matrix/MAT_v1/AMATRIX0`. After all the column–vector data is computed, the script automatically gives the command `redist` to gather them to one matrix file (`MAT.t21l5.ncepwin.dat` in the above example). Thus you will see, right after the above message, a subsequent log message like

```
INPUT:/home/hiro/ln_solver/matrix/MAT_v1/AMATRIX0
INPUT:/home/hiro/ln_solver/matrix/MAT_v2/AMATRIX0
..............
INPUT:/home/hiro/ln_solver/matrix/MAT_t8/AMATRIX5
INPUT:/home/hiro/ln_solver/matrix/MAT_p1/AMATRIX5
INPUT:/home/hiro/ln_solver/matrix/MAT_q1/AMATRIX5
..............
```

7.2.4 Computing forcing vector

Exactly speaking, a forcing given to solve the steady mLBM is surface heat flux anomalies associated with the anomalous SST. In case of the time integration, they are internally computed and hold in the model during the whole integration. However, for the current use you need to calulate, in advance, anomalous flux forcing which is a function of both SST anomaly and the basic state. To do this, first recompile the model executable file with another model option, namely,

```
###### options for model ###################

### lbm/pwm
######## dry model
#MODELOPT = -DOPT_MKMAT -DOPT_CLASSIC
#MODELOPT = -DOPT_MKMAT -DOPT_OWALL -DOPT_CLASSIC
#MODELOPT = -DOPT_WVFRC -DOPT_ORHS -DOPT_CLASSIC
######## moist model
#MODELOPT = -DOPT_MKMAT
#MODELOPT = -DOPT_MKMAT -DOPT_OWALL
MODELOPT = -DOPT_WVFRC -DOPT_ORHS
```

and

```
%> cd $LNHOME/model/src
%> make clean.special
%> make lbm
```

You will find a sample shell script to compute the forcing, `mksfrc.moist.csh`, in `$LNHOME/model/sh/mkamat`. Edit only the environmental variables set at the script, for example, system and executable file name. The variable `FRCGRD` is an input GrADS file for the SST anomaly while `FRCMAT` denotes an output file which contains spherical coefficients for the surface flux forcing. Note that `NTRN` must be the same as the zonal wave number truncated in `Lmake.inc`.

After you rewritten the script with relevant configuration, run it as

```
%> cd $LNHOME/model/sh/mkamat
%> mksfrc.moist.csh
```

It will readily create the forcing data.

7.2.5 Solve linear system

You can now solve the linear system by using a command `lin` and `trn` both of which are compiled at `$LNHOME/solver/steady`. `lin` solves the linear system while `trn` convert the response in wave space to that in a physical space, together with performing $\sigma \rightarrow$ pressure transform and calculating geopotential height.

```
%> cd $LNHOME/solver/steady
%> make clean
%> make
```

Before executing `lin` and `trn`, parameter and file names must be adequately specified as in section 3.2.4. Edit following lines in `$LNHOME/solver/steady/SETPAR`.

```
 &nmfin  cfm='/home/hiro/ln_solver/matrix/MAT.moist.t21l8.ncepwin.dat',
         cfr='/home/hiro/ln_solver/data/frc.t21l8.mat',
         cfs='/home/hiro/ln_solver/data/x.dat',
         cfg='/home/hiro/ln_solver/data/rsp.t21l8.grd'
 &end
 &nmbs   cbs='/home/hiro/ln_solver/bs/grads/ncepwin.t21l8.grd'
 &end
 &nmuv   o2uv=f, opl=t
 &end
 &nmall  owall=t
 &end
 &nmcls  oclassic=f
 &end
```

The above parameters are the same as in section 3.2.4 except for `oclassic=f` which specifies the moist model. To obtain the response data defined by `cfg`, enter

```
%> lin ; trn
```

It will show a message on your terminal like

```
### SOLVE LINEAR SYSTEM ###
***********************************************
Read matrix L
   Matrix Size (left)= 6963 x 6963
Read vector F
Solve for X = L-1 F
   LAPACK routine call:  (DGESV)
   INFO code returned by DGESV = 0
***********************************************
### END OF EXECUTION ###
```

If a flag of `INFO code` in the above message is not 0, it indicates that the linear system was not correctly solved. In this case you need to check the matrix and/or forcing.

A sample `.ctl` file for the above example is found in `$LNHOME/sample`.

7.2.6 Associated heating response

After you obtained the steady response, you might want to know how the heating fields look like. Since the heat source and moisture sink is expressed by a function of the steady response which you just computed (cf. section A.4), you need to substitute the response data into the model again. For this purpose, you can use the same script as in section 7.2.4, namely, `mksfrc.moist.csh`. Before running it, you have to add the following lines in the parameters.

```
&nmwfrc  wvfrc='$ANMGRD'                   &end
&nmsfrc  fsfrc='$FRCGRD', gsfrc='$TFRC'   &end
```

where `$ANMGRD` and `$FRCGRD` are the steady response and SST anomaly, while `$TFRC` denotes the output which contains heating anomaly, respectively. All of these has been specified at the top of the script. Note that the `$ANMGRD` must contain wind, but not ζ and D, on a σ surface. If you followed the parameters listed in section 7.2.5, that response data should be again transformed from wave domain to grid domain since it contains ζ and D converted to the pressure coordinate. Thus, edit again `&nmfin` and `&nmuv` in `$LNHOME/solver/steady/SETPAR` as

```
&nmfin  cfm='/home/hiro/ln_solver/matrix/MAT.moist.t21l8.ncepwin.dat',
        cfr='/home/hiro/ln_solver/data/frc.t21l8.mat',
        cfs='/home/hiro/ln_solver/data/x.dat',
        cfg='/home/hiro/ln_solver/data/rsp.t21l8.s.grd'
&end
&nmuv   o2uv=t, opl=f
&end
```

then,

```
%> trn
```

It will readily creates another response data `rsp.t21l8.s.grd` which is to be given in `mksfrc.moist.csh`. After that you can run the script as

```
%> cd $LNHOME/model/sh/mkamat
%> mksfrc.moist.csh
```

Quantities calculated by the procedure are the heat and moisture sources, both of which have been divided into that due to anomalous state and due to SST anomaly, respectively. You will find the corresponding `.ctl` file, `heat.t21l8.ctl`, in `$LNHOME/sample`.

An example of the steady moist response is shown in Fig. 7.2, which was obtained with observed SST anomalies during El Niño given to the T21L11m10 moist linear model. It indicates that the model captures low –level circulation prevailing during ENSO (e.g. anomalous anticyclone near Philippines) and relevant heating anomalies (enhanced and suppressed convection over the central equatorial Pacific and the surrounding regions). However, the steady version tends to overestimate the cumulus heating (cf. Fig. 7.2d) thus the atmospheric response (Fig. 7.2b) with the same parameters as in the time integration (we do not yet trace the reason). Therefore you might want to reduce the impact of convective adjustment by setting slightly longer convective time scale (may be 4–6 hours).

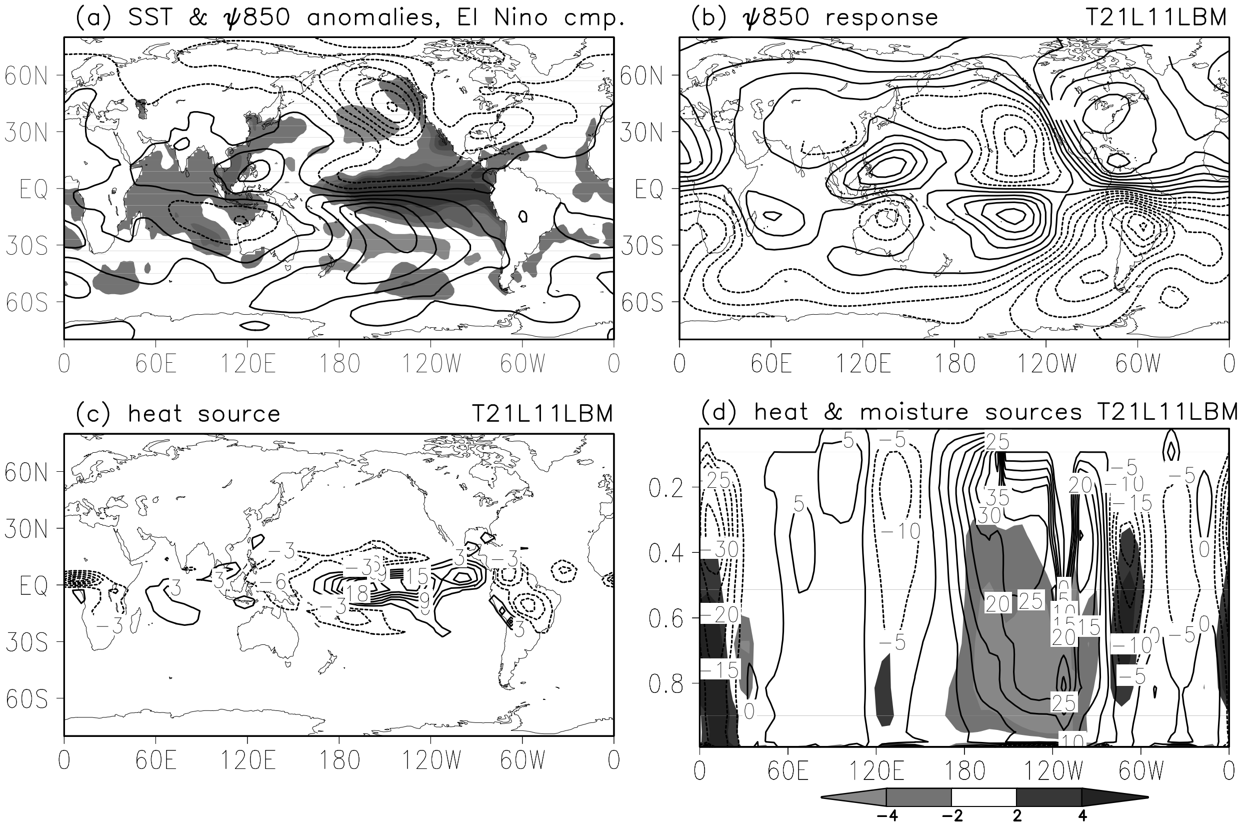


Figure 7.2 (a) Observed compisites of SST and ψ anomalies at 850 hPa based on the Niño 3 index. (b) Steady ψ response in a T21L11m10 linear model, (c) associated heating (vertically averaged, K day^{-1}), and (d) heat (contour) and moisture (shaded) sources at teh equator.

8 Barotropic model

So far, many interesting results on the atmospheric dynamics have been obtained from a simplified barotropic model which employs a linearized vorticity equation forced by divergent component $(-fD)$ in the equation (e.g. Branstator 1985a, 1985b). Although one should carefully choose the level of basic state as cautioned by Ting (1996), it has also been known that some of steady responses in LBM, especially a remote extratropical response to tropical heating, can be well reproduced by the barotropic model, indicating that the barotropic model is still a useful tool to diagnose the dynamical processes in the atmosphere.

8.1 Steady response

Choose `'PROJECT = baro'` in `$LNHOME/Lmake.inc`. The vertical resolution is, of course, 1 while no wave truncation is specified. As in sections 3.2 and 3.3, you can select either solving a full matrix or SWM.

```
###### Model type ################
### nonlinear, dynamical core
#PROJECT = dcore
### standard, making linear matrix (incl.  planetary wave model)
#PROJECT = mkamat
### time-advance linear model (incl.  storm track model)
#PROJECT = tintgr
### barotropic model
PROJECT = baro
###### Horizontal Resolution ################
#HRES = t10
HRES = t21
#HRES = t42
###### Vertical Resolution ################
VRES = l1
#VRES = l5
#VRES = l8
#VRES = l11
#VRES = l20
###### zonal wave truncation ################
ZWTRN =
#ZWTRN = m15
#ZWTRN = m10
#ZWTRN = m5
###### options for model ###################
.....
### barotropic model
```

```
#MODELOPT = -DOPT_MKMAT
MODELOPT = -DOPT_MKMAT -DOPT_OWALL
#MODELOPT = -DOPT_WVFRC
```

In the above example, you selected a full matrix. Generally you do not use SWM unless you have a special concern with the explicit zonal–wave interaction since the barotropic model is simple enough that does not require to reduce to SWM.

Then, make the model executable file (remember you need to have a model library in advance, see section 2.3).

```
%> cd $LNHOME/model/src
%> make clean.special
%> make lbm
```

This will create an executable file `lbm2.t21ml20cbaro`.

Unlike LBM described in previous sections, the barotropic model employs a basic state which only contains stream function field. It has been reported that the most relevant basic state will be chosen from an equivalent barotropic level (Ting 1996). The equivalent barotropic level is considered as 300 and 500 hPa for winter and summer cases, respectively, so that we have prepared the NCEP climatological fields for these levels. The data, `ncep.clim.y49-99.psi3.tXX.grd` and `ncep.clim.y49-99.psi5.tXX.grd` (`XX` denotes the horizontal resolution of either 10, 21, or 42), are the 300 and 500 hPa stream function, both included in a data archive `ln_solver2.2.ncepdata.tar.gz`.

The winter basic states `ncepwin.t21l1` (Gtool format) and `ncepwin.t21l1.grd` (GrADS format) have been included in the LBM package, while compile a command `ncep1vbs` if you need to make other basic state.

```
%> cd $LNHOME/solver/util
%> make clean
%> make br
```

Then edit following lines in `$LNHOME/solver/util/SETPAR`.

```
 &nmncp  cncep='/home/hiro/ln_solver/bs/ncep/ncep.clim.y49-99.psi5.t21.grd',
         kmo=6, navg=3, ozm=f, osw=f, cvar='PSI'
 &end
 &nmbs   cbs0='/home/hiro/ln_solver/bs/gt3/ncepsum.t21l1',
         cbs='/home/hiro/ln_solver/bs/grads/ncepsum.t21l1.grd'
 &end
```

and execute

```
%> ncep1vbs
```

The above example will make a summer basic state based on the 500 hPa stream function climatology. The parameters are defined in a similar manner to 2.4.

Since the barotropic model is based on the linear vorticity equation, a simplified vorticity forcing can be prepared using `mkfrcbr`. First, compile it as (skip it if you have done already)

```
%> cd $LNHOME/solver/util
%> make clean
%> make br
```

Then edit following lines in `$LNHOME/solver/util/SETPAR`.

```
 &nmfin  cfm='/home/hiro/ln_solver/data/frc.baro.mat',
         cfg='/home/hiro/ln_solver/data/frc.baro.grd',
         fact=1.0,1.0,1.0,1.0,1.0
 &end
 &nmhpr  khpr=1,
         hamp=-2.6D-5,
         xdil=30.,
         ydil=12.,
         xcnt=220.,
         ycnt=5.
 &end
 &nmall  owall=t
 &end
```

and execute

```
%> mkfrcnbr
```

Again in the above example the full matrix is considered but not SWM (if you use as SWM set `&nmall  owall=f`). The vorticity forcing defined as above mimics a forcing due to upper–level divergence associated with the El Niño heating. You will find the sample forcing file `frc.baro.grd` in `$LNHOME/sample/` which was made with the above parameters. A description of parameters has been presented in 3.1.2. Note that the present version does not provide tool which transforms a user–customized GrADS forcing file to the spherical coefficient data (like `fvec` in 3.2).

To make a linear operator matrix, we prepared a sample script `$LNHOME/model/sh/baro/mkamatbr.csh`. Copy it to any name, say, `mkamatbr.test.csh`, and edit suitably.

```
%> cd $LNHOME/model/sh/baro
%> cp mkamatbr.csh mkamatbr.test.csh
%> chmod u+x mkamatbr.test.csh
```

Meaning of environmental variables and parameters are the same as in sections 3.1.3 and 3.2.2. Only the difference from LBM is that the script can readily make a matrix file without using a post–processor `redist`. Thus the matrix file name is also defined in this script as `MATNAM`. Temporarily we make the matrix named `MATBR.ncepwin.dat`. In the above sample script, linear drag is set at $(50\mathrm{dy})^{-1}$. The barotropic response tends to be dominated by a zonally uniform component as you will see later, there is another parameter to factorize the damping for the zonal mean component, `&nmzmfct`, which specifies a factor that is multiplied to the drag coefficient set with `&nmdamp`. Now, you can make the matrix very quickly.

```
%> cd $LNHOME/model/sh/baro
%> mkamatbr.test.csh
```

A procedure to solve the steady response is similar to that in section 3.2.4. After you compile the command `linbr` and `trnbr` as

```
%> cd $LNHOME/solver/steady
%> make clean
%> make br
```

and edit `$LNHOME/solver/steady/SETPAR` as

```
 &nmfin  cfm='/home/hiro/ln_solver/matrix/MATBR.ncepwin.dat',
         cfr='/home/hiro/ln_solver/data/frc.baro.mat',
         cfs='/home/hiro/ln_solver/data/x.dat',
         cfg='/home/hiro/ln_solver/data/rsp.baro.grd'
 &end
 &nmall  owall=t
 &end
```

you can obtain the steady response by executing `linbr` and `trnbr`.

```
%> linbr ; trnbr
```

If you use the barotropic model as SWM, set `&nmall owall=f` as in making the forcing.

A sample response data `rsp.baro.grd` to the forcing `$LNHOME/sample/frc.baro.grd` is provided in `$LNHOME/sample/`. As you recognize by looking at the corresponding `.ctl` file, the response is automatically transformed from vorticity to stream function. Shown in Fig. 8.1 (right panel) is the stream function response in the sample response data. When you use a weaker damping for the zonal mean component (`&nmzmfct zmfctv=1.0` in `mkamatbr.test.csh`), the zonal mean response is much prevailing (left panel in Fig. 8.1). Try to obtain the same results using your code.

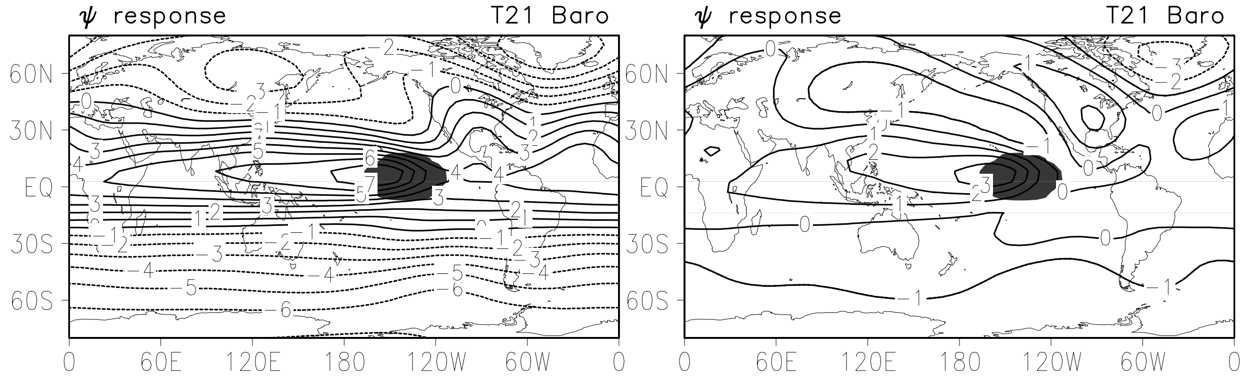


Figure 8.1 Example of the stream function response using a T21 barotropic model without (left) and with (right) an additional damping for zonal mean components. The contour interval is 1×10^7 m^2 s^{-1} while shading denotes a vorticity forcing ($F \leq -2\times 10^{-11}$ s^{-2}).

8.2 Modal decomposition

A singular value decomposition or eigenanalysis as detailed in 5.2 and 5.3 is also possible for the linear operator of the barotropic model.

Compile the commands which execute SVD and EOF to the linear operator (set of `svdbr, strnbr` and `eofbr, etrnbr`, respectively).

```
%> cd $LNHOME/solver/expert
%> make clean
%> make br
```

If you apply the SVD expansion, edit the following lines in `$LNHOME/solver/expert/SETPAR`.

```
 &nmsfin  cfm='/home/hiro/ln_solver/matrix/mat/MATBR.ncepwin.dat',
          cfe='/home/hiro/ln_solver/data/svec.br/svalbaro.grd'
          cfl='/home/hiro/ln_solver/data/svec.br/svecbaro_u.dat'
          cfr='/home/hiro/ln_solver/data/svec.br/svecbaro_v.dat'
          cflg='/home/hiro/ln_solver/data/svec.br/svecbaro_u.grd'
          cfrg='/home/hiro/ln_solver/data/svec.br/svecbaro_v.grd'
 &end
```

```
&nmwave  nwave=0
&end
&nmall   owall=t
&end
```

The file `cfm` defines an input matrix while other parameters in `&nmsfin` are all outputs. `cfe` denotes a GrADS file that contains sequence of singular values, and GrADS files for the left and right–singular vectors (U and V in A.2) are specified by `cflg` and `cfrg`, respectively. Files of `cfl` and `cfr` are temporarily made to keep singular vectors in wave space. As in 8.1, set `owall=f` when you solve a matrix for particular wave number defined by `&nmwave` in case of SWM. Otherwise, `nwave` is ignored in solving a full matrix.

To obtain the singular modes, enter

```
%> svdbr ; strnbr
```

You will find a set of sample `.ctl` files for the outputs, `svalbaro.grd` and `svecbaro_v.grd`, in `$LNHOME/sample`.

On the other hand, if you do an eigenanalysis, edit `$LNHOME/solver/expert/SETPAR` as follows.

```
&nmefin  cfm='/home/hiro/ln_solver/matrix/mat/MATBR.ncepwin.dat',
         cfe='/home/hiro/ln_solver/data/svec.br/evalbaro.grd',
         cfs='/home/hiro/ln_solver/data/svec.br/evecbaro.dat',
         cfg='/home/hiro/ln_solver/data/svec.br/evecbaro.grd'
&end
&nmwave  nwave=0
&end
&nmall   owall=t
&end
```

The difference from SVD is only that you specify a single file name for eigenvectors by `cfg`. Likewise, enter as follows to perform the eigenmode computation.

```
%> eofbr ; etrnbr
```

Again you will find a set of sample `.ctl` files for the outputs, `evalbaro.grd` and `evecbaro.grd`, in `$LNHOME/sample`. An example of eigenanalysis to the barotropic operator obtained by the above procedure is shown in Fig. 8.2 which shows the eigenspectrum and two conspicuous modes.

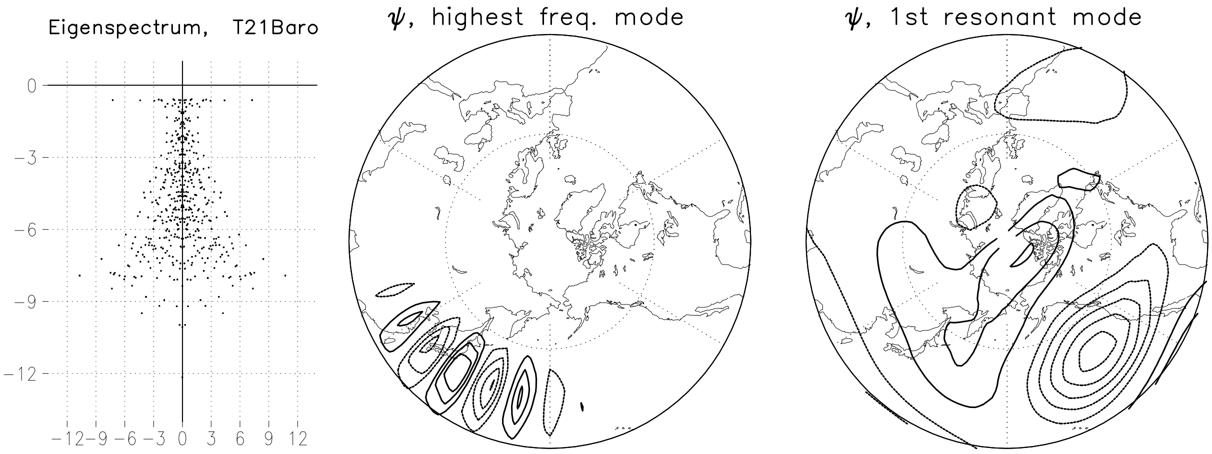


Figure 8.2 Sample eigenspectrum (left), highest frequency ($\sim$ period of 2.6hr) mode, and a first resonant mode with the decay time scale of 11.5dy. The horizontal (vertical) axis of the spectrum denotes a frequency with the unit of 1×10^{-5} s^{-1} (growth rate with 1×10^{-6} s^{-1}). The modes are represented by the stream function.

A Mathematical principles of linear dynamics

In this Appendix, a brief description of the linear atmospheric dynamics, based on primitive equations (PE), is given using a set of symbolic equations of state. Readers can refer to Watanabe and Kimoto (2000, 2001) and Watanabe and Jin (2003) for further details.

A.1 Linear dynamical system

First, let X be a vector containing prognostic variables in PE, vorticity (ζ), divergence (D), temperature (T), and logarithm of surface pressure ($\pi = \ln P_s$), namely, $\mathsf{X} \equiv \mathsf{X}(\zeta, D, T, \pi)$. A symbolic representation of the dynamical system is denoted as

$$d_t\mathsf{X} + (\mathsf{L} + \mathsf{NL})\mathsf{X} = \mathsf{F} \quad , \tag{1}$$

where L and NL are a linear and nonlinear part of the dynamical operator that consists of, for example, advection, Coriolis, pressure gradient, and dissipation terms, while F indicates forcing. Now Eq.(1) is linearized about a basic state $\overline{\mathsf{X}}$ and the nonlinear part is neglected, yielding a linear system for perturbations X'.

$$d_t\mathsf{X}' + \mathsf{L}\mathsf{X}' = \mathsf{F}' \quad . \tag{2}$$

Note that the operator L is now a function of the basic state, i.e. $\mathsf{L} \equiv \mathsf{L}(\overline{\mathsf{X}})$. Considering the steady forced problem of Eq.(2), we rewrite it as (prime is dropped for convenience)

$$\mathsf{L}\mathsf{X} = \mathsf{F} \quad . \tag{3}$$

Equation (3) corresponds to a set of linear simultaneous equations, so that it can be readily solved as

$$\mathsf{X} = \mathsf{L}^{-1}\mathsf{F} \quad . \tag{4}$$

A.2 Orthogonal basis for dynamical operator

Consider a technique to obtain the dispersion relationship of the linear vorticity equation (assuming plane wave solution), you will easily imagine that a numerical technique to decompose L into orthogonal basis functions is an expansion of such an analytical procedure. The modal decomposition is not only an approach to detect normal modes of the atmosphere but also to understand forced solution in terms of the basic characteristic of the state.

There may be, at least, two methods for the orthogonal expansion: a conventional eigen-analysis (e.g. Branstator 1985b) and a singular vector analysis (Navarra 1993; Itoh and Kimoto 1999). The eigenanalysis is carried out by decomposing L as

$$\mathsf{L}\mathsf{E} = \sigma\mathsf{E} \quad , \tag{5}$$

where σ is a set of complex eigenvalues while E is corresponding eigenvectors. Note that, E_i (i is mode) is not necessarily orthogonal since L is not self–adjoint. Instead, it has turned out (see Branstator 1985b) that the eigenvectors for the adjoint operator of L are in biorthogonal relationship with E, so that any steady solution is represented by using both E and eigenvectors of the adjoint operator.

A difficulty to preserve orthogonality in the eigenanalysis can be avoided by using another method, singular vector analysis. This employs an singular value decomposition (SVD) for L as

$$\mathsf{L} = \mathsf{U}\mathbf{\Sigma}\mathsf{V}^T \quad , \tag{6}$$

where U and V are orthonormal matrices and $\mathbf{\Sigma}$ is the diagonal matrix of real singular values. Vectors contained in U and V (u_i and v_i, respectively) is often called the u–vector and v–vector, respectively. Using Eq.(6), a steady forced problem Eq.(3) can be rewritten as

$$\mathsf{X} = \mathsf{V}\mathbf{\Sigma}^{-1}\mathsf{U}^T\mathsf{F} = \sum_i^N \mathsf{v}_i \frac{(\mathsf{u}_i, \mathsf{F})}{\sigma_i} \quad . \tag{7}$$

The above equation implies that the structure of steady response is determined by v–vector while the amplitude and sign are defined by the σ^{-1} and a projection of u–vector onto forcing. Suppose now F consists of many spatially random forcing, then perform EOF analysis to a group of steady response X. From Eq.(7) and the fact $\mathsf{F}\mathsf{F}^T = \mathsf{I}$,

$$\mathsf{X}\mathsf{X}^T = \mathsf{V}\mathbf{\Sigma}^{-2}\mathsf{V}^T \quad . \tag{8}$$

Physical implication of Eq.(8) is that v–vectors are equivalent to the EOFs of steady responses to random forcing and the leading singular vector (v_i with the smallest singular value) prevails in the stationary fields forced by random forcing. The second argument is reasonable since the smallest singular value will correspond to the mode closest to neutral and having lowest frequency in the eigenanalysis.

A.3 Detection of preferred modes

Recall a steady system of Eq.(3) with an arbitrary forcing F. At the same time, we consider a response to an one–point forcing such as

$$\mathsf{G} = \mathsf{L}^{-1}\mathsf{f} \quad , \tag{9}$$

where

$$\mathsf{f} = \delta(\lambda, \varphi, \lambda', \varphi') \qquad \begin{cases} 1, & \text{for} \quad \lambda = \lambda',\ \varphi = \varphi' \\ 0, & \text{elsewhere} \end{cases}$$

Multiplying F to Eq.(9) and integrate over the globe leads to

$$
\begin{aligned}
\int\int \mathbf{G}(\lambda,\varphi,\lambda',\varphi')\mathbf{F}(\lambda',\varphi')d\lambda' d\varphi' & \\
&= \mathbf{L}^{-1}\int\int \mathbf{f}(\lambda,\varphi,\lambda',\varphi')\mathbf{F}(\lambda',\varphi')d\lambda' d\varphi' \\
&= \mathbf{L}^{-1}\mathbf{F}(\lambda,\varphi) \\
&= \mathbf{X} \quad . \qquad (10)
\end{aligned}
$$

An appriximate form of Eq.(10) can be derived when G is decomposed into a set of orthogonal basis function by, for example, a convectional EOF technique.

$$
\begin{aligned}
\mathsf{G} &= \mathsf{EP} \\
&= \sum_i^N \mathsf{e}_i(\lambda,\varphi)\mathsf{p}_i(\lambda',\varphi') \quad , \qquad \mathsf{E} = \mathsf{E}(\mathsf{e}_1, \mathsf{e}_2, ..., \mathsf{e}_N), \qquad (11)
\end{aligned}
$$

$$
\begin{aligned}
\mathsf{X} &\simeq \int\int \mathsf{e}_1(\lambda,\varphi)\mathsf{p}_1(\lambda',\varphi')\mathsf{F}(\lambda',\varphi')d\lambda' d\varphi' \\
&= \mathsf{e}_1 \int\int \mathsf{p}_1 \mathsf{F} d\lambda' d\varphi' \quad . \qquad (12)
\end{aligned}
$$

It turns out that Eq.(12) is similar to a representation of X in terms of the singular vectors (see A.2), i.e.

$$\mathsf{L} = \mathsf{U}\mathbf{\Sigma}\mathsf{V}^T \quad ,$$

$$\mathsf{X} \simeq \mathsf{v}_1 \sigma_1^{-1} \int\int \mathsf{u}_1 \mathsf{F} d\lambda' d\varphi' \quad . \qquad (13)$$

The comparison between Eq.(12) and Eq.(13) indicates that the leading EOF of G is roughly regarded as a neutral vector v_1 while the associated principal components p_1 corresponds to an optimal forcing defined by u_1.

A.4 Linear convective interaction with dynamics

When we consider the forcing solely due to anomalous cumulus convection, Eq. (2) can be solved with a given diabatic heating as F (referred to as the dry linear model). The forcing F, which is a function of both X and a SST anomaly T_s', may be decomposed into two components, $\mathsf{F}_i(\mathsf{X})$ and $\mathsf{F}_e(T_s')$ ('i' and 'e' stand for internal and external, respectively). Then Eq. (2) is rewritten as

$$d_t \mathsf{X} + (\mathsf{L} - \mathsf{F}_i)\mathsf{X} = \mathsf{F}_e \quad . \qquad (14)$$

This equation indicates that the heating due to anomalous convection is internally determined such as to satisfy a constraint between large–scale circulation and the convective anomaly. Note that X in Eq.(14) now includes perturbation in the specific humidity q'. In the linear model represented by Eq.(14) (referred to as the moist linear model), a forcing F_e is composed by terms in linearized surface bulk fluxes that depend on T_s' and the basic state but not on the perturbation X. See Watanabe and Jin (2003) for details.

B Troubleshooting

This Appendix gives several hints for solving a problem that might troubles you in using the LBM package.

B.1 How to use the package on a different system?

If you want to use the package on an architecture other than Sun OS (Solaris), DEC alpha, SGI Origin, and SR8000, you first have to install math libraries of LAPACK and BLAS. The LAPACK pre–built archive for several machines is available at `[http://www.netlib.org/lapack/archives/]` while an optimized BLAS library is freely distributed by some of the vendors. If both are not available, go to `[http://www.netlib.org/lapack/index.html]` and `[http://www.netlib.org/blas/]` then copy compressed package files `lapack.tgz` and `blas.tgz`, respectively. Follow the respective users manual for how to install them on your machine.

After you completed the installation of LAPACK and BLAS, newly make system–dependent files as follows. We assumed in the example below that you installed on Linux.

1. move libraries under the LBM package subdirectory

```
%> cd $LNHOME/solver/lib
%> mkdir linux
```

then move library files, say, `lapack_linux.a`, `blas_linux.a`, and `tmglib_linux.a` to `$LNHOME/solver/lib/linux/`.

2. make a new system include file for the matrix solver

```
%> cd $LNHOME/solver/include
%> cp make.inc.sun make.inc.linux
```

then, edit the file `make.inc.linux` such as to match the name of architecture and compile options.

3. make new system–dependent files for the model

```
%> cd $LNHOME/model/src/sysdep
```

You may not make new files if you find `Makedef.linux` and `ylinux.F` there. If you do not (e.g. you use different system),

```
%> cp Makedef.sun Makedef.XX  (XX is your system)
%> cp ysun.F yXX.F)
```

then appropriately modify the code.

4. add your system to `Lmake.inc`

Open `$LNHOME/Lmake.inc` and add a new line as

```
# include file for Makefile for linear solver
#
# set environment LNHOME in your .cshrc
#
###### Architecture ################
ARC = linux
#ARC = sun
#ARC = alpha
#ARC = sgi
#ARC = sr8000
```

B.2 How to make basic state or forcing on a T21 Gaussian grid?

The tools such as `mkfrcng` and `ncepsbs` in `$LNHOME/solver/util` help you to prepare the basic state and forcing. However, when you want to make them using otehr sources, for example, the basic state based on ECMWF data, you need to write programs by yourself. While the package does not support such an user–dependent procedure, you can use a subroutine `$LNHOME/solver/custom/intp.f` for the spatial interporation of the original data into the Gaussian grid of the triangular truncation. By calling the subroutine `INTP` in your program together with given input/output dimensions and latitude/longitude information, you will easily obtain a T21 (either T10 or T42) Gaussian data. See top part of `intp.f` for detail.

B.3 Cannot compile commands, why?

When you use PWM (section 3.3) on a machine with relatively small memory, you may be troubled with a compilation error at `$LNHOME/solver/steady` like

```
lin.f:
     MAIN lin:
        "lin.f", line 26:  Error:  array "a" has too many elements
```

This error occurs since the array declaration has been set for LBM, that is too large for PWM matrices[1] . This error will be solved by modifying the top part of `lin.f` as

```
*
*     standard
CC      PARAMETER ( MAXN=2*NMAX*NTR*(NVAR*KMAX+1) )
*
*     reduced memory
```

[1] The above problem will be fixed in the next version in which the code is written by fortran 90.

```
      PARAMETER ( MAXN=2*NMAX*(NVAR*KMAX+1) )
```

then re–compile the program. The same problem may happen if you apply modal decomposition of EOF or SVD for PWM matrices. If you see the following error in compiling `eof` or `svd` under `$LNHOME/solver/expert`,

```
eof.f:
     MAIN eof:
        "eof.f", line 40:  Error:  array "a" has too many elements
        "eof.f", line 40:  Warning:  overflow in expression
```

modify the source files in the same way as `lin.f` shown above.

B.4 Want to make a change to the model, where should I look at?

As you will find, fortran programs stored in $LNHOME/model/src/proj/$PROJECT are files that you may want to make a change. Note that a program with the same name but in a different directry (e.g. `dterm-2.F` in `proj/tintgr/` and `proj/mkamat/`) is not the same. However, those with the same name have a similar role in the model as follows, so that you can find which you should modify for a specific purpose.

`aadmn-2.F`	master subroutine controlling I/O and time step
`dadmn-2.F`	sub–master routine managing dynamical processes
`dsetd-2.F`	subroutine which defines diffusion coefficients
`dterm-2.F`	subroutine that calculate all the tendency terms
`dgdyn-2.F`	subroutine for calculation of advection terms
`dstep-2.F`	subroutine that manages time advance
`dintg-2.F`	subroutine carrying out implicit integration

B.5 Want to submit batch jobs at SR8000

HITACHI supercomputer SR8000 has a huge memory and gives the fastest computation, so that it will much save your time to use SR8000. Because the system allows very limited computation on the TTS prompt, most jobs are going to be implemented on the batch system. If you are not familiar with the batch commands, use a sample script for making a linear matrix and solving steady response prepared in `$LNHOME/model/sh/` and `$LNHOME/solver/steady/`, respectively. For example, you may submit the script `go.csh` which executes `mkamat.l11.classic.csh` as

```
%> cd $LNHOME/model/sh
%> qsub go.csh
```

The log message is written to `message.mk`. Notice that its takes about 1hr to make the linear matrix for T21L11m10 LBM.

Similarly, submit `mat_solve` to compute steady response.

```
%> cd $LNHOME/solver/steady
%> qsub mat_solve
```

Refer to *Supercomputing News* (in Japanese) for the detail of batch commands and batch class available in SR8000.

References

Betts, A. K., 1986: A new convective adjustment scheme. Part I: Observational and theoretical basis. *Quart.J.R.Met.Soc.*, **112**, 677–691.

Betts, A. K., and M. J. Miller, 1986: A new convective adjustment scheme. Part II: Single column tests using GATE wave, BOMEX, ATEX and arctic air–mass data sets. *Quart.J.R.Met.Soc.*, **112**, 693–709.

Branstator, G., 1985a: Analysis of general circulation model sea–surface temperature anomaly simulations using a linear model. Part I: Forced solutions. *J.Atmos.Sci.*, **42**, 2225–2241.

Branstator, G., 1985b: Analysis of general circulation model sea–surface temperature anomaly simulations using a linear model. Part II: Eigenanalysis. *J.Atmos.Sci.*, **42**, 2242–2254.

Branstator, G., 1990: Low–frequency patterns induced by stationary waves. *J.Atmos.Sci.*, **47**, 629–648.

Branstator, G., 1992: The maintenance of low–frequency atmospheric anomalies. *J.Atmos.Sci.*, **49**, 1924–1945.

Branstator, G., 1995: Organization of storm track anomalies by recurring low–frequency circulation anomalies. *J.Atmos.Sci.*, **52**, 207–226.

DeWeaver, E., and S. Nigam, 2000a: Do stationary waves drive the zonal–mean Jet anomalies of the northern winter? *J.Climate*, **13**, 2160–2176.

DeWeaver, E., and S. Nigam, 2000b: Zonal–eddy dynamics of the North Atlantic Oscillation. *J.Climate*, **13**, 3893–3914.

Hayashi, Y., 1971: Instability of large–scale equatorial waves with a frequency–dependent CISK parameterization. *J.Meteor.Soc.Japan*, **49**, 59–62.

Held, I. M., S. W. Lyons, and S. Nigam, 1989: Transients and the extratropical response to El Niño. *J.Atmos.Sci.*, **46**, 163–174.

Hoskins, B. J., and D. J. Karoly, 1981: The steady linear response of a spherical atmosphere to thermal and orographical forcing. *J.Atmos.Sci.*, **38**, 1179–1196.

Hoskins, B. J., and M. J. Rodwell, 1995: A model of the Asian summer monsoon. Part I: The global scale. *J.Atmos.Sci.*, **52**, 1329–1340.

Itoh, H., and M. Kimoto, 1999: Weather regimes, low–frequency oscillations, and principal patterns of variability: A perspective of extratropical low–frequency variability. *J.Atmos.Sci.*, **56**, 2684–2705.

Jin, F. F., and B. J. Hoskins, 1995: The direct response to tropical heating in a baroclinic atmosphere. *J.Atmos.Sci.*, **52**, 307–319.

Kimoto, M., F.-F. Jin, M. Watanabe, and N. Yasutomi, 2001: Zonal–eddy coupling and a neutral mode theory for the Arctic Oscillation. *Geophys.Res.Lett.*, **28**, 737–740.

Navarra, A., 1993: A new set of orthonormal modes for linearized meteorological problems. *J.Atmos.Sci.*, **50**, 2569–2583.

Neelin, J. D., and J.-Y. Yu, 1994: Modes of tropical variability under convective adjustment and the Madden–Julian oscillation. Part I: Analytical theory. *J.Atmos.Sci.*, **51**, 1876–1894.

Nigam, S., 1994: On the dynamical basis for the Asian summer monsoon rainfall–El Niño relationship. *J.Climate*, **7**, 1750–1771.

Nigam, S., I. M. Held, and S. W. Lyons, 1986: Linear simulation of the stationary eddies in a general circulation model. Part I: The no-montain model. *J.Atmos.Sci.*, **43**, 2944–2961.

Nigam, S., I. M. Held, and S. W. Lyons, 1988: Linear simulation of the stationary eddies in a GCM. Part II: The montain model. *J.Atmos.Sci.*, **45**, 1433–1452.

Peng, S., and J. S. Whitaker, 1999: Mechanisms determining the atmospheric response to midlatitude SST anomalies. *J.Climate*, **12**, 1393–1408.

Rodwell, M. J., and B. J. Hoskins, 1995: A model of the Asian summer monsoon. Part II: Cross–equatorial flow and PV behavior. *J.Atmos.Sci.*, **52**, 1341–1356.

Seager, R., and S. E. Zebiak, 1995: Simulation of tropical climate with a linear primitive equation model. *J.Climate*, **8**, 2497–2520.

Stevens, D. E., and R. Lindzen, 1978: Tropical wave–CISK with a moisture budget and cumulus friction. *J.Atmos.Sci.*, **35**, 940–961.

Ting, M., 1991: The stationary wave response to a midlatitude SST anomaly in an idealized GCM. *J.Atmos.Sci.*, **48**, 1249–1275.

Ting, M., 1996: Steady linear response to tropical heating in barotropic and baroclinic models. *J.Atmos.Sci.*, **53**, 1698–1709.

Ting, M., and I. M. Held, 1990: The stationary wave response to a tropical SST anomaly in an idealized GCM. *J.Atmos.Sci.*, **47**, 2546–2566.

Ting, M., and L. Yu, 1998: Steady response to tropical heating in wave linear and nonlinear baroclinic model. *J.Atmos.Sci.*, **55**, 3565–3582.

Valdes, P. J., and B. J. Hoskins, 1989: Linear stationary wave simulations of the time–mean climatological flow. *J.Atmos.Sci.*, **46**, 2509–2527.

Watanabe, M., and F.-F. Jin, 2002: Role of Indian Ocean warming in the development of Philippine Sea anticyclone during ENSO. *Geophys.Res.Lett.*, **29**, doi:10.1029/2001GL014318.

Watanabe, M., and F.-F. Jin, 2003: A moist linear baroclinic model: Coupled dynamical–convective response to El Niño. *J.Climate*, **16**, 1121–1139.

Watanabe, M., F.-F. Jin, and M. Kimoto, 2002: Tropical axisymmetric mode of variability in the atmospheric circulation: Dynamics as a neutral mode. *J.Climate*, **15**, 1537–1554.

Watanabe, M., F.-F. Jin, and L.-L. Pan, 2005: Accelerated iterative method for solving steady solutions of linearized atmospheric models. *J.Atmos.Sci.* submitted.

Watanabe, M., and M. Kimoto, 1999: Tropical–extratropical connection in the Atlantic atmosphere–ocean variability. *Geophys.Res.Lett.*, **26**, 2247–2250.

Watanabe, M., and M. Kimoto, 2000: Atmosphere–ocean thermal coupling in the North Atlantic: A positive feedback. *Quart.J.R.Met.Soc.*, **126**, 3343–3369.

Watanabe, M., and M. Kimoto, 2001: Corrigendum. *Quart.J.R.Met.Soc.*, **127**, 733–734.

Yu, J.-Y., and J. D. Neelin, 1994: Modes of tropical variability under convective adjustment and the Madden–Julian Oscillation. Part II: Numerical results. *J.Atmos.Sci.*, **51**, 1895–1914.