Run the steady response test run

Ziming Chen (2023.10.27)

1. Unzip and install the LBM model: Model-Linear-Baroclinic-Model-master.zip

   Check the icc: chenzm@: icc -v

   Set the $LNHOME in ~/.bashrc

   export LNHOME=~/my_data/Models/LBM/LBM_NetCDF

2. Edit the file $LNHOME/Lmake.inc

   1) Set the ARC (Architecture):

```
###### Architecture          ################
ARC  = N000
#ARC  = sun
#ARC  = alpha
#ARC  = sgi
#ARC  = sr8000
#ARC  = linux
```

   2) Set the PROJECT

```
###### Model type            ################

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

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

### accerelated iterative solver (AIM)
#PROJECT         = aim

### nonlinear, dynamical core
#PROJECT         = dcore

### barotropic model
#PROJECT         = baro

### coupled mLBM-CZ
#PROJECT         = cz

### orographic forcing
#PROJECT         = wvfrc.topo
```

   tintgr indicates time-advance method. For others, please see the LBM guidance

   3) Set the resolution

   (We will set the zonal wave truncation, only when PROJECT is mkamat)

```
###### 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
#ZWTRN  = m6
```

4) Model options

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

This is for the dry model

3. Making model library

1) Set the compiler

$LNHOME/model/src/sysdep/Makedef.N000

```
16 # MAKE          = make
17 CC              = icc
18 #FC             = g77
19 #LD             = g77
20 FC              = ifort
21 LD              = ifort
```

Or you could use gcc compiler and gfortran compiler:

```
# MAKE          = make
CC              = cc
#FC             = g77
#LD             = g77
#CC             = icc
FC              = f77
LD              = f77
#FC             = ifort
#LD             = ifort
```

cc for gcc; icc for icc

2) Create lib and bin dictionary

```
[Perlmutter:LBM_NetCDF>mkdir -p $LNHOME/model/bin/N000
 (py39)
[Perlmutter:LBM_NetCDF>mkdir -p $LNHOME/model/lib/N000
 (py39)
```

3) Make model library

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

And then a lib such as liblbm2t42m120c.a will be created under:

$LNHOME/model/lib/N000

PS: Only if you change the model resolution you have to re-make the lib

4. Preparing basic states

1) Modify the compiler in $LNHOME/solver/include/make.inc.N000

ifort compiler:

```
11 #  Modify the FORTRAN and OPTS definitions to refer to the
12 #  compiler and desired compiler options for your machine.  NOOPT
13 #  refers to the compiler options desired when NO OPTIMIZATION is
14 #  selected.  Define LOADER and LOADOPTS to refer to the loader and
15 #  desired load options for your machine.
16 #
17 FORTRAN  = ifort
18 OPTS     = -O -u -convert big_endian -fpe3 -no-vec
19 NOOPT    = -u -convert big_endian -fpe3 -no-vec
20 LOADER   = ifort
```

If you are using other compiler or other version of ifort, it would be fine to modify these code.

LAPACK location:

```
34 #  The location of the libraries to which you will link.  (The
35 #  machine-specific, optimized BLAS library should be used whenever
36 #  possible.)
37 #
38 # 2022.11.04: Ziming Chen
39 BLASLIB      = $(LNHOME)/solver/lib/$(ARC)/librefblas.a
40 LAPACKLIB    = $(LNHOME)/solver/lib/$(ARC)/liblapack.a
41 TMGLIB       = $(LNHOME)/solver/lib/$(ARC)/libtmglib.a
```

If you have not installed LAPACK in your computer, please see 2) below. Otherwise, skip 2)

2) Install LAPACK lib if it has not been installed before

   In the code of this LBM, I have installed the LAPACK, so normally you could skip this step and compile the basic state code in 3).

   Download lapack in https://codeload.github.com/Reference-LAPACK/lapack/tar.gz/v3.9.0

   Unzip it in $LNHOME/solver/lib/N000

   Set the compiler in $LNHOME/solver/lib/linux/lapack-3.10.1/make.inc.example, and then rename it as make.inc

   Modify the name of CC compiler and FC compiler

```
#  CC is the C compiler, normally invoked with options CFLAGS.
#
CC = icc
CFLAGS = -O3

#  Modify the FC and FFLAGS definitions to the desired compiler
#  and desired compiler options for your machine.  NOOPT refers to
#  the compiler options desired when NO OPTIMIZATION is selected.
#
#  Note: During a regular execution, LAPACK might create NaN and Inf
#  and handle these quantities appropriately. As a consequence, one
#  should not compile LAPACK with flags such as -ffpe-trap=overflow.
#
FC = ifort
FFLAGS = -O2 -frecursive
FFLAGS_DRV = $(FFLAGS)
FFLAGS_NOOPT = -O0 -frecursive
```

Change the INT_CPU_TIME to EXT_ETIME in the TIMER

```
#  Timer for the SECOND and DSECND routines
#
#  Default:  SECOND and DSECND will use a call to the
#  EXTERNAL FUNCTION ETIME
#TIMER = EXT_ETIME
#  For RS6K:  SECOND and DSECND will use a call to the
#  EXTERNAL FUNCTION ETIME_
#TIMER = EXT_ETIME_
#  For gfortran compiler:  SECOND and DSECND will use a call to the
#  INTERNAL FUNCTION ETIME
# TIMER = INT_ETIME
#  If your Fortran compiler does not provide etime (like Nag Fortran
#  Compiler, etc...) SECOND and DSECND will use a call to the
#  INTERNAL FUNCTION CPU_TIME
TIMER = INT_CPU_TIME
#  If none of these work, you can use the NONE value.
#  In that case, SECOND and DSECND will always return 0.
```

(Original screenshot)

```
38  #  Timer for the SECOND and DSECND routines
39  #
40  #  Default:  SECOND and DSECND will use a call to the
41  #  EXTERNAL FUNCTION ETIME
42  TIMER = EXT_ETIME
43  #  For RS6K:  SECOND and DSECND will use a call to the
44  #  EXTERNAL FUNCTION ETIME_
45  # TIMER = EXT_ETIME_
46  #  For gfortran compiler:  SECOND and DSECND will use a call to the
47  #  INTERNAL FUNCTION ETIME
48  # TIMER = INT_ETIME
49  #  If your Fortran compiler does not provide etime (like Nag Fortran
50  #  Compiler, etc...) SECOND and DSECND will use a call to the
51  #  INTERNAL FUNCTION CPU_TIME
52  # TIMER = INT_CPU_TIME
53  #  If none of these work, you can use the NONE value.
54  #  In that case, SECOND and DSECND will always return 0.
```

(Update screenshot)

Modify file $LNHOME/solver/lib/linux/Makefile

```
13  .PHONY: lib
14  lib: lapacklib tmglib
15  # lib: blaslib variants lapacklib tmglib
```

(Original screenshot)

```
13  .PHONY: lib
14  # lib: lapacklib tmglib
15  lib: blaslib variants lapacklib tmglib
```

(Updated screenshot)

Annotate the tmglib, while open the lib: blaslib variants lapacklik tmglib

Compile the LAPACK lib in $LNHOME/solver/lib/N000:

chenzm@: make

3) Compile the basic state
   a) The original method to read basic state:

In $LNHOME/solver/util/:

chenzm@: make bs

modify the SETPAR (or SETPAR_Ziming if you want to read the basic state in NetCDF format, please see b))

Set the Input File at first:

```
14  &nmncp cncep='/global/cfs/cdirs/m1867/zmchen/my_data/Models/LBM/LBM_NetCDF/bs/ncep/
    ncep.clim.y58-97.t42.grd',
15
16          cncep2='/global/cfs/cdirs/m1867/zmchen/my_data/Models/LBM/LBM_NetCDF/bs/ncep/
            ncep.clim.y58-97.ps.t42.grd',
17
18          calt='/global/cfs/cdirs/m1867/zmchen/my_data/Models/LBM/LBM_NetCDF/bs/gt3/grz.t42',
19
20          kmo=6, navg=3, ozm=f, osw=f, ousez=t
```

Here we read the basic state in the ncep module and interpolate the basic static of ncep into sigma level.

cncep: the name of data including seven basic variables

cncep2: the name of pressure data

For other parameter please see the param_list.

b) If Read NetCDF, then modify the SETPAR_Ziming:

```
14  &nmncp cncep='/project/projectdirs/m1867/zmchen/Work/2022_2/NH_SummerStationaryWave/
    Data/07.LBM/LBMhist/bs/test/
    CMIP6.historical.ltm.y1995-2014.t42.Ziming.gs_2Models.grb',
15
16          cncep2='/project/projectdirs/m1867/zmchen/Work/2022_2/NH_SummerStationaryWave/
            Data/07.LBM/LBMhist/bs/test/
            CMIP6.historical.ltm.y1995-2014.t42.ps.Ziming.gs_2Models.grb',
17
18          cncep_nc='/project/projectdirs/m1867/zmchen/Work/2022_2/
            NH_SummerStationaryWave/Data/07.LBM/LBMhist/bs/test/
            CMIP6.historical.ltm.y1995-2014.t42.Ziming_2Models.nc'
19
20          calt='/project/projectdirs/m1867/zmchen/Work/2022_2/NH_SummerStationaryWave/
            Data/07.LBM/LBMhist/bs/gt3/grz.t42',
21
22          kmo=6, navg=3, ozm=f, osw=f, ousez=t
23  &end
```

* And then modify the Makefile:

Add the sentence of ncepsbs_mine in Makefile

```
NCPBS = ncepsbs.o

NCPBS_mine = ncepsbs_mine.o

NCPBBS = ncep1vbs.o

all: dec hdec bsgrd redist mkfrcng mkfrcsst fvec gt2gr ncepsbs ncepsbs_mine
1vbs ecmsbs mymkfrcng mymkfrcng222

bs: dec hdec bsgrd ncepsbs ncepsbs_mine ncep1vbs ecmsbs

ncepsbs: $(NCPBS) ; \
     $(LOADER) $(LOADOPTS) -o $@ \
     $(NCPBS) $(LAPACKLIB) $(BLASLIB) $(LDLIBS)

ncepsbs_mine: $(NCPBS_mine) ; \
     $(LOADER) $(LOADOPTS) -o $@ \
     $(NCPBS_mine) $(LAPACKLIB) $(BLASLIB) $(LDLIBS) \
     -lnetcdff -L$(NETCDF_L) -I${NETCDF_I}
     # Ziming Chen: 2022.11.18
```

* And next modify the directory of netcdf.inc in

$LNHOME/solver/util/ncepsbs_mine.f, if necessary:

```
 9   C      2022.11.18: Ziming Chen
10          include '/global/homes/c/chenzm/intel/netcdf_intel/include/netcdf.
11         *inc'
12   C      2022.11.18: Ziming Chen
```

Set the Output File:

```
25   &nmbs  cbs0='/project/projectdirs/m1867/zmchen/Work/2022_2/NH_SummerStationaryWave/Data/07.LBM/
     LBMhist/bs/gt3/ziming.sum1.t42l20',
26          cbs='/project/projectdirs/m1867/zmchen/Work/2022_2/NH_SummerStationaryWave/Data/07.LBM/
            LBMhist/bs/grads/ziming.sum1.t42l20.grd'
27   &end
```

cbs0: output file of basic state

cbs: output file in grd format

Set big endian (must):

export F_UFMTENDIAN=big

If you make clean all the compiling, export this before “make”

c) in $LNHOME/solver/util/

chenzm@: ./ncepsbs or ./ecmsbs

(./ncepsbs_mine, If you read the basic state in NetCDF)

if you see the error below:

```
(py37) [honghx@fat1 util]$ ./ncepsbs

 ### start making basic state ###

 ... selected month from:          6  ...
 ... number of month averaged:          3  ...

  @@ Input :
 /home/honghx/LBM/ln_solver/bs/ncep/ncep.clim.y58-97.t21.grd

  @@ Input :
 /home/honghx/LBM/ln_solver/bs/ncep/ncep.clim.y58-97.ps.t21.grd

  @@ Output:
 /home/honghx/LBM/ln_solver/bs/gt3/ncepsum.t21111zm

  @@ Output:
 /home/honghx/LBM/ln_solver/bs/grads/ncepsum.t21111zm.grd


 ... data read start ...
    ... month =           6
    ... month =           7
    ... month =           8
 ... data read end ...
 ... Z used for Ps ...
forrtl: severe (24): end-of-file during read, unit 30, file /home/honghx/LBM/ln_solver/bs/gt3/grz
.t21
Image              PC                Routine            Line        Source
ncepsbs            0000000000440EC4  Unknown               Unknown  Unknown
ncepsbs            00000000004585CA  Unknown               Unknown  Unknown
ncepsbs            0000000000403E99  Unknown               Unknown  Unknown
ncepsbs            00000000004027AE  Unknown               Unknown  Unknown
libc.so.6          00007FD376544B15  Unknown               Unknown  Unknown
ncepsbs            00000000004026B9  Unknown               Unknown  Unknown
```


This is because the characters of cncep and cncep2 are limited by 50. You need to increase the characters in ncepsbs.f

```
C 2022.11.03: Ziming Chen
C  extend the length of CHARACTER to 500 (original one is 90)
      CHARACTER*500 CNCEP         !! NCEP pressure level data (grads)
      CHARACTER*500 CNCEP2        !! NCEP surface pressure    (grads)
      CHARACTER*500 CALT          !! topography               (gtool3)
      CHARACTER*500 CBS0          !! basic state              (gtool)
      CHARACTER*500 CBS           !! basic state              (grads)
C 2022.11.03: Ziming Chen
```

In addition, we also need to increase the characters in the following code:

model/src/proj/tintgr/dterm-2.F

solver/util/mkfrcng.f

solver/steady/linpwf.f

solver/include/make.inc.N000

model/src/proj/tintgr/aadmn-2.F

solver/util/gt2gr.f

If read the basic state in NetCDF format:

chenzm@: ./ncepsbs_mine

Attention: Not missing value or Fill Value are allowed in the basic state interpolation code

5. Steady response

1) Preparing forcing

In $LNHOME/solver/util/SETPAR, indicate the output forcing

```
32    &nmfin cfm='/pscratch/sd/c/chenzm/my_data/Model/LBM_NetCDF/data/Forcing/
      frc.t42l20.Ziming_test.mat',
33
34          cfg='/pscratch/sd/c/chenzm/my_data/Model/LBM_NetCDF/data/Forcing/
            frc.t42l20.Ziming_test.grd'
```

And then we set the shape and type of forcing:

```
37    &nmvar ovor=f, odiv=f, otmp=t, ops=f, osph=f
```

ovor: vorticity forcing; div: divergence forcing; tmp: heating (temperature) forcing;

ps: surface pressure forcing; sph: specific humidity forcing

f: False; t: True

For the input data of forcing

Interpolate the forcing pattern vertically into these levels: 0.995, 0.97999, 0.94995, 0.89988, 0.82977, 0.74468, 0.64954, 0.54946, 0.45447, 0.36948, 0.2945, 0.22953, 0.17457, 0.1244, 0.084683, 0.0598005, 0.0449337, 0.0349146, 0.02488, 0.00829901

Prepared the grd format forcing data at the beginning of open file:

The data order: Vor, Div, Temp, PS, Hum

Add the following code in $LNHOME/solver/util/mkfrcng.f (I have added these code in this LBM package):

```
WRITE(*, *) 'Forcing Pattern: ', CFG_mine
OPEN ( 20, FILE = CFG_mine, status='OLD', FORM = 'UNFORMATTED',
&  ACCESS = 'direct', RECL = 128*64*20*4)
```

```
C       READ( 12) r_Temp
      DO 111 K = 1, KMAX
         READ( 20, rec = 1 ) ((r_Vor(I, J, K), I = 1, IMAX),
     &                    J = 1, JMAX)
  111  CONTINUE
      DO 112 K = 1, KMAX
         READ( 20, rec = 1 ) ((r_Div(I, J, K), I = 1, IMAX),
     &                    J = 1, JMAX)
  112  CONTINUE
      DO 113 K = 1, KMAX
         READ( 20, rec = 1 ) ((r_Temp(I, J, K), I = 1, IMAX),
     &                    J = 1, JMAX)
  113  CONTINUE
      READ( 20, rec = 1 ) ((r_PS(I, J), I = 1, IMAX), J = 1, JMAX)
      DO 114 K = 1, KMAX
         READ( 20, rec = 1 ) ((r_Hum(I, J, K), I = 1, IMAX),
     &                    J = 1, JMAX)
  114  CONTINUE
      CLOSE(20)
```

(Attention: define the r_Vor, r_Div, r_Temp, r_PS and r_Hum at first)

Prepare the NetCDF format forcing data:

Add the following code in $LNHOME/solver/util/mkfrcng.f:

- Include the NetCDF module in the code. Attention: the NetCDF module must be compiled by ifort

```
11          include 'dim.f'
12    C     2022.11.11: Ziming Chen
13          include '/opt/cray/pe/netcdf/4.8.1.1/intel/19.1/include/
14         *netcdf.inc'
15    C     2022.11.11: Ziming Chen
```

- Read the NetCDF data

```
151   *
152   *     open files
153   *
154   C
155   C     2022.10.10: Ziming Chen
156   C     Read by using NetCDF module
157         WRITE(*, *) 'Forcing Pattern: ', CFG_mine
158         retval = nf_open(CFG_mine, nf_nowrite, ncid)
159         retval = nf_inq_dimid(ncid, 'lon', ilon_dimid)
160         retval = nf_inq_dimlen(ncid, ilon_dimid, nlonin)
161         retval = nf_inq_dimid(ncid, 'lat', ilat_dimid)
162         retval = nf_inq_dimlen(ncid, ilat_dimid, nlatin)
163         retval = nf_inq_dimid(ncid, 'lev', ipre_dimid)
164         retval = nf_inq_dimlen(ncid, ipre_dimid, nprein)
165
166         nlon   = IMAX
167         nlat   = JMAX
168         if (nlonin .ne. nlon .or. nlatin .ne. nlat) then
169           print*, 'grid error: nlonin,lon,nlatin,lat', nlonin, nlon,
170        &                                               nlatin, nlat
171           stop
172         endif
173
174         np     = KMAX
175         if (nprein .ne. np) then
176           print*, 'pressure levels do not match', nprein, np
177           stop
178         endif
179
180         retval = nf_inq_varid(ncid, 'lon', ilon_varid)
181         retval = nf_get_var(ncid, ilon_varid, lonin)
182         retval = nf_inq_varid(ncid, 'lat', ilat_varid)
183         retval = nf_get_var(ncid, ilat_varid, latin)
184         retval = nf_inq_varid(ncid, 'lev', ipre_varid)
185         retval = nf_get_var(ncid, ipre_varid, pre)
186         retval = nf_inq_varid(ncid, 't', itemp_varid)
187         retval = nf_get_var_real(ncid, itemp_varid, r_Temp)
188         retval = nf_close(ncid)
189   C     Read by using NetCDF module
```

(here only read one forcing pattern)

Write forcing data

```
*       write GrADS file
*
        DO 100 K = 1, KMAX
           IF( OVOR ) THEN
              DO 110 J = 1, JMAX
                 DO 110 I = 1, IMAX
                    IJ = (J-1)*IDIM + I
C                   2022.11.11: Ziming Chen
C                     DUMX( I, J) = GFRCT( IJ, K)
                    DUMX(I, J) = r_Vor(I, J, K)
C                   2022.11.11: Ziming Chen
 110             CONTINUE
                 WRITE( IFG ) ((SNGL(DUMX(I,J)),I=1,IMAX),J=1,JMAX)
           ELSE
                 WRITE( IFG ) ((SNGL(DUM(I,J)),I=1,IMAX),J=1,JMAX)
           ENDIF
 100    CONTINUE
```

(Other forcing patterns should be written in the same way)

Then modify the cfg_mine which indicates the input forcing name in SETPAR:

This is the forcing data in NetCDF that I should prepare:

```
&nmfin cfm='/pscratch/sd/c/chenzm/my_data/Model/LBM_NetCDF/data/Forcing/
frc.t42l20.Ziming_test.mat',

       cfg='/pscratch/sd/c/chenzm/my_data/Model/LBM_NetCDF/data/Forcing/
       frc.t42l20.Ziming_test.grd'

       cfg_mine='/pscratch/sd/c/chenzm/my_data/Model/LBM_NetCDF/data/Forcing/
       frc.t42l20.Ziming.Heating_CP20N150W.nc'

       fact=1.0,1.0,1.0,1.0,1.0
&end
```

For more details of the parameters, please see $LNHOME/solver/util/param_list

2) Compile and ouput the forcing pattern

```
[Perlmutter:util>make clean
rm -f *.o *~ dec hdec bsgrd redist mkfrcng mkfrcsst fvec gt2gr ncepsbs ncep1vbs ec
g222
(py39)
[Perlmutter:util>make
```

```
\
     ifort  -o redist \
     redist.o /pscratch/sd/c/chenzm/my_data/Model/LBM_NetCDF/solver/lib/N000/liblapack.a /pscratch/sd/c/c
henzm/my_data/Model/LBM_NetCDF/solver/lib/N000/librefblas.a /pscratch/sd/c/chenzm/my_data/Model/LBM_NetCD
F/model/lib/N000/liblbm2t42ml20c.a
ifort -O -u -convert big_endian -fpe3 -no-vec -c mkfrcng.f
\
     ifort  -o mkfrcng \
     mkfrcng.o /pscratch/sd/c/chenzm/my_data/Model/LBM_NetCDF/solver/lib/N000/liblapack.a /pscratch/sd/c/
chenzm/my_data/Model/LBM_NetCDF/solver/lib/N000/librefblas.a /pscratch/sd/c/chenzm/my_data/Model/LBM_NetC
DF/model/lib/N000/liblbm2t42ml20c.a \
     -lnetcdff -L/global/homes/c/chenzm/intel/netcdf_intel/lib -I/global/homes/c/chenzm/intel/netcdf_inte
l/include
ifort -O -u -convert big_endian -fpe3 -no-vec -c mkfrcsst.f
\
     ifort  -o mkfrcsst \
     mkfrcsst.o /pscratch/sd/c/chenzm/my_data/Model/LBM_NetCDF/solver/lib/N000/liblapack.a /pscratch/sd/c
/chenzm/my_data/Model/LBM_NetCDF/solver/lib/N000/librefblas.a /pscratch/sd/c/chenzm/my_data/Model/LBM_Net
CDF/model/lib/N000/liblbm2t42ml20c.a
ifort -O -u -convert big_endian -fpe3 -no-vec -c fvec.f
\
     ifort  -o fvec \
     fvec.o /pscratch/sd/c/chenzm/my_data/Model/LBM_NetCDF/solver/lib/N000/liblapack.a /pscratch/sd/c/che
nzm/my_data/Model/LBM_NetCDF/solver/lib/N000/librefblas.a /pscratch/sd/c/chenzm/my_data/Model/LBM_NetCDF/
model/lib/N000/liblbm2t42ml20c.a
ifort -O -u -convert big_endian -fpe3 -no-vec -c gt2gr.f
\
     ifort  -o gt2gr \
     gt2gr.o /pscratch/sd/c/chenzm/my_data/Model/LBM_NetCDF/solver/lib/N000/liblapack.a /pscratch/sd/c/ch
enzm/my_data/Model/LBM_NetCDF/solver/lib/N000/librefblas.a /pscratch/sd/c/chenzm/my_data/Model/LBM_NetCDF
/model/lib/N000/liblbm2t42ml20c.a
ifort -O -u -convert big_endian -fpe3 -no-vec -c ncepsbs.f
\
     ifort  -o ncepsbs \
     ncepsbs.o /pscratch/sd/c/chenzm/my_data/Model/LBM_NetCDF/solver/lib/N000/liblapack.a /pscratch/sd/c/
chenzm/my_data/Model/LBM_NetCDF/solver/lib/N000/librefblas.a /pscratch/sd/c/chenzm/my_data/Model/LBM_NetC
DF/model/lib/N000/liblbm2t42ml20c.a
ifort -O -u -convert big_endian -fpe3 -no-vec -c ncepsbs_mine.f
\
     ifort  -o ncepsbs_mine \
     ncepsbs_mine.o /pscratch/sd/c/chenzm/my_data/Model/LBM_NetCDF/solver/lib/N000/liblapack.a /pscratch/
sd/c/chenzm/my_data/Model/LBM_NetCDF/solver/lib/N000/librefblas.a /pscratch/sd/c/chenzm/my_data/Model/LBM
_NetCDF/model/lib/N000/liblbm2t42ml20c.a \
     -lnetcdff -L/global/homes/c/chenzm/intel/netcdf_intel/lib -I/global/homes/c/chenzm/intel/netcdf_inte
l/include
ifort -O -u -convert big_endian -fpe3 -no-vec -c ncep1vbs.f
\
     ifort  -o ncep1vbs \
     ncep1vbs.o /pscratch/sd/c/chenzm/my_data/Model/LBM_NetCDF/solver/lib/N000/liblapack.a /pscratch/sd/c
/chenzm/my_data/Model/LBM_NetCDF/solver/lib/N000/librefblas.a /pscratch/sd/c/chenzm/my_data/Model/LBM_Net
CDF/model/lib/N000/liblbm2t42ml20c.a
ifort -O -u -convert big_endian -fpe3 -no-vec -c ecmsbs.f
\
     ifort  -o ecmsbs \
     ecmsbs.o /pscratch/sd/c/chenzm/my_data/Model/LBM_NetCDF/solver/lib/N000/liblapack.a /pscratch/sd/c/c
henzm/my_data/Model/LBM_NetCDF/solver/lib/N000/librefblas.a /pscratch/sd/c/chenzm/my_data/Model/LBM_NetCD
F/model/lib/N000/liblbm2t42ml20c.a
ifort -O -u -convert big_endian -fpe3 -no-vec -c mymkfrcng.f
\
     ifort  -o mymkfrcng \
     mymkfrcng.o /pscratch/sd/c/chenzm/my_data/Model/LBM_NetCDF/solver/lib/N000/liblapack.a /pscratch/sd/
c/chenzm/my_data/Model/LBM_NetCDF/solver/lib/N000/librefblas.a /pscratch/sd/c/chenzm/my_data/Model/LBM_Ne
tCDF/model/lib/N000/liblbm2t42ml20c.a
ifort -O -u -convert big_endian -fpe3 -no-vec -c mymkfrcng222.f
\
     ifort  -o mymkfrcng222 \
     mymkfrcng222.o /pscratch/sd/c/chenzm/my_data/Model/LBM_NetCDF/solver/lib/N000/liblapack.a /pscratch/
sd/c/chenzm/my_data/Model/LBM_NetCDF/solver/lib/N000/librefblas.a /pscratch/sd/c/chenzm/my_data/Model/LBM
_NetCDF/model/lib/N000/liblbm2t42ml20c.a
(py39)
```

```
(py39)
[Perlmutter:util>./mkfrcng
  ### MAKE FORCING MATRIX ###
   *** precision check        1.100
```

Output information

```
Perlmutter:util>./mkfrcng
 ### MAKE FORCING MATRIX ###
  *** precision check ..    1.10000000000000
  *** precision check ..    1.01000000000000
  *** precision check ..    1.00100000000000
  *** precision check ..    1.00010000000000
  *** precision check ..    1.00001000000000
  *** precision check ..    1.00000100000000
  *** precision check ..    1.00000010000000
  *** precision check ..    1.00000001000000
  *** precision check ..    1.00000000100000
  *** precision check ..    1.00000000010000
  *** precision check ..    1.00000000001000
  *** precision check ..    1.00000000000100
  *** precision check ..    1.00000000000010
  *** precision check ..    1.00000000000001
  *** precision check ..    1.00000000000000
  *** precision check ..    1.00000000000000

 .................
 Selected shape:
   Horizontal:Elliptic
   Vertical  :Gamma
 .................

 Forcing Pattern:
 /pscratch/sd/c/chenzm/my_data/Model/LBM_NetCDF/data/Forcing/frc.t42l20.Ziming.H
 eating_CP20N150W.nc




 Matrix file         :
 /pscratch/sd/c/chenzm/my_data/Model/LBM_NetCDF/data/Forcing/frc.t42l20.Ziming_t
 est.mat




 Matrix file (GrADS) :
 /pscratch/sd/c/chenzm/my_data/Model/LBM_NetCDF/data/Forcing/frc.t42l20.Ziming_t
 est.grd




 .................

  *** precision check ..    1.10000000000000
  *** precision check ..    1.01000000000000
  *** precision check ..    1.00100000000000
  *** precision check ..    1.00010000000000
  *** precision check ..    1.00001000000000
  *** precision check ..    1.00000100000000
  *** precision check ..    1.00000010000000
  *** precision check ..    1.00000001000000
  *** precision check ..    1.00000000100000
  *** precision check ..    1.00000000010000
  *** precision check ..    1.00000000001000
  *** precision check ..    1.00000000000100
  *** precision check ..    1.00000000000010
  *** precision check ..    1.00000000000001
  *** precision check ..    1.00000000000000
  *** precision check ..    1.00000000000000
  ### check SUMGW =   0.999999999999945
 Set vertical shape
 .................

 Set horizontal shape
 .................

 Written to GrADS file
 .................

 Written to matrix file (all)
 .................


 ### END OF EXECUTION ###
```

And you will see the output forcing data:

```
Forcing Pattern:
/pscratch/sd/c/chenzm/my_data/Model/LBM_NetCDF/data/Forcing/frc.t42l20.Ziming.H
eating_CP20N150W.nc



Matrix file         :
/pscratch/sd/c/chenzm/my_data/Model/LBM_NetCDF/data/Forcing/frc.t42l20.Ziming_t
est.mat



Matrix file (GrADS) :
/pscratch/sd/c/chenzm/my_data/Model/LBM_NetCDF/data/Forcing/frc.t42l20.Ziming_t
est.grd
```

3) Last before running the model

Modify the linear-run.t42l20.csh in $LNHOME/model/sh/tintgr

```
 9 #
10 setenv LNHOME   /pscratch/sd/c/chenzm/my_data/Model/LBM_NetCDF                # ROOT of model
11 setenv LBMDIR   $LNHOME/model                              # ROOT of LBM
12 setenv SYSTEM   N000                                     # execute system
13 setenv RUN      $LBMDIR/bin/$SYSTEM/lbm2.t42ml20ctintgr # Excutable file
14 setenv TDIR     $LNHOME/solver/util
15 setenv FDIR     $LNHOME/data/Forcing                        # Directory for Output
16 setenv DIR      $LNHOME/data/Output                       # Directory for Output
17 setenv BSFILE   $LNHOME/bs/gt3/ziming.sum1.t42l20           # Atm. BS File
18 setenv RSTFILE  $DIR/Restart.amat                       # Restart-Data File
19 setenv FRC      $FDIR/frc.t42l20.Ziming_test.grd              # initial perturbation
20 setenv SFRC     $FDIR/frc.t42l20.Ziming_test.grd              # steady forcing
21 setenv TRANS    gt2gr
22 setenv TEND     30
23 #
```

Modify the nmtime from the top screenshot to bottom screenshot:

```
39   &nmtime start=1,1,1,0,0,0, end=1,1,$TEND,0,0,0
```

```
39   &nmtime start=0,1,1,0,0,0, end=0,1,$TEND,0,0,0
```

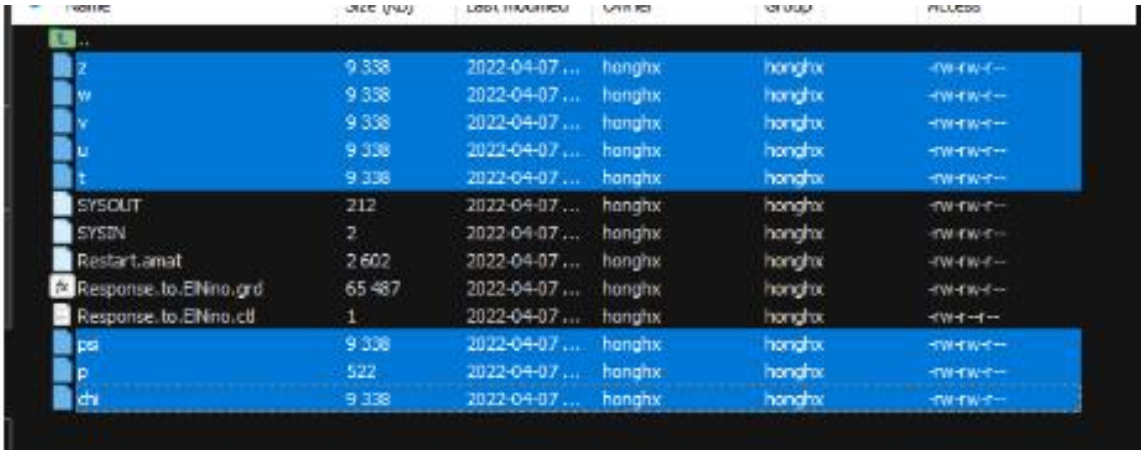
4) Run the model

```
(py37) [honghx@fat1 tintgr]$ chmod u+x linear-run.test.csh
(py37) [honghx@fat1 tintgr]$ csh linear-run.test.csh
                                                CSDN @IAP_Honghx
```

>>> chmod u+x linear-runt42l20.csh

>>> csh linear-runt42l20.csh

If we run the test successfully, then we will see in $LNHOME/data/Output:



You could also see the output information in SYSOUT when the case is running.

6. Post process and visualize

   1) Modify the $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
```

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

A sample control file `linear.t21120.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.

2) Convert the grd data to NetCDF

Copy one of the ctl file in $LNHOME/data/Output, and then modify the information inside.

And modify the 01.GrdToNetCDF.sh.

At last run the 01.GrdToNetCDF.sh:

>>> bash 01.GrdToNetCDF.sh

3) The forcing and response are in Figure 1 and 2, respectively.

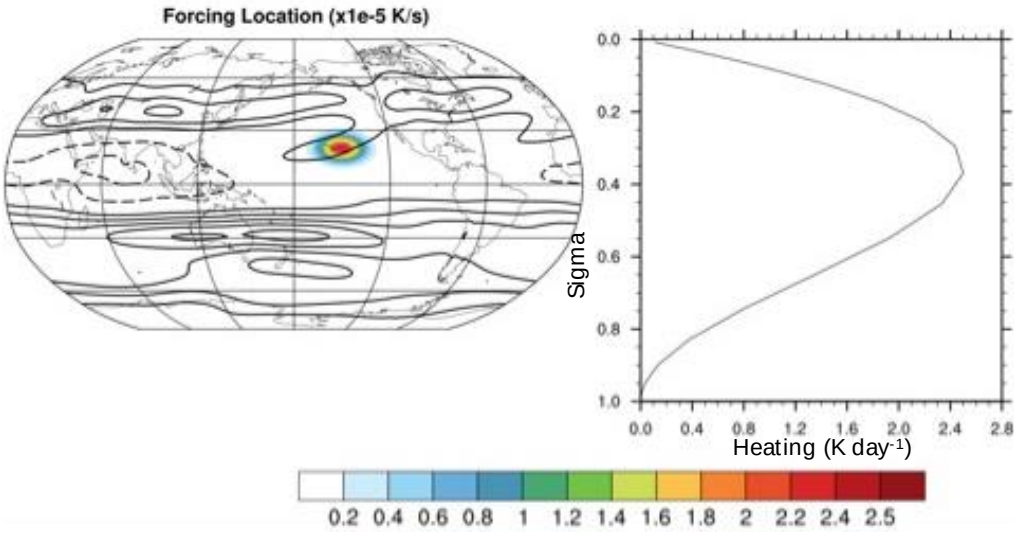


Figure 1.

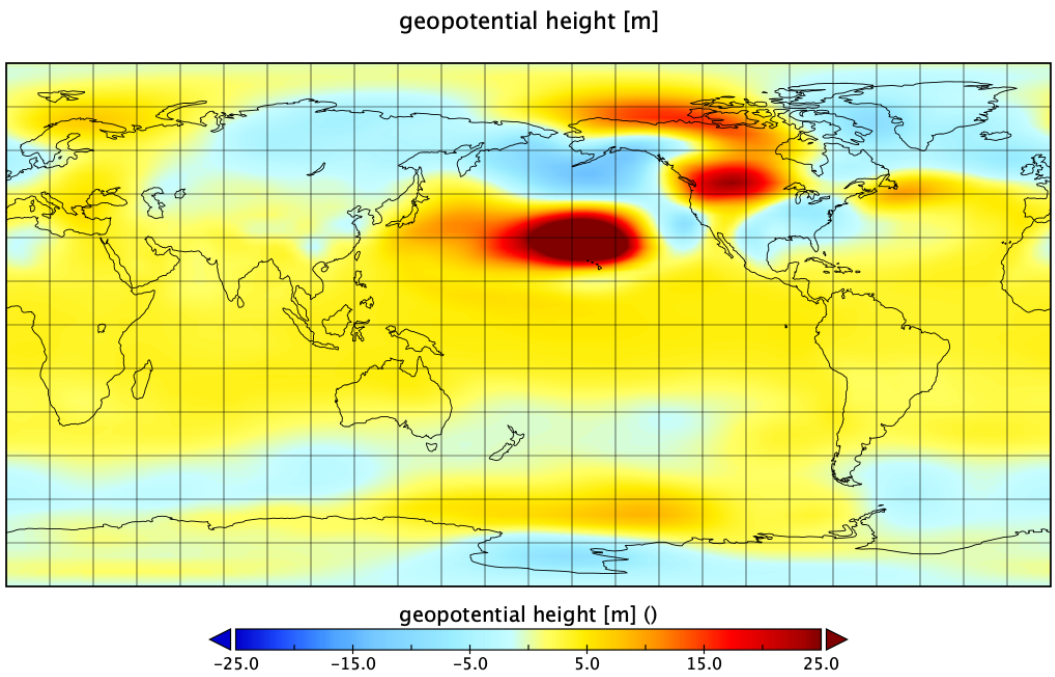

geopotential height [m]
geopotential height [m] ()
-25.0
-15.0
-5.0
5.0
15.0
25.0

Figure 2