# Stochastic skeleton model: quick instructions

Sulian (CIMS)

Nov 2013

## 1 Introduction

Those are quick instructions for simulating and reading the outputs of the stochastic skeleton model. For references see:

- TMS2013 (Thual, Majda, Stechmann 2013: A Stochastic Skeleton Model for the MJO, J. Atm. Sci): the article on the stochastic skeleton model.

- RLW2013: Report on solving of long-wave dynamics (Sulian Thual- A. Majda, Feb 2013): this gives more details than TMS2013 on how the long-wave equations are solved. Notably, note that the stochastic skeleton model code is intended at solving flows with greater meridional truncation (i.e. more Hermite functions/zonal strips) than in TMS2013. In particular, the scaling of Hermite functions used in the code is different from the one of TMS2013, and is the one of BM2006.

- BM2006 (Biello, J. A. and Majda, A. J. (2006). Modulating synoptic scale convective activity and boundary layer dissipation in the IPESD models of the Madden-Julian oscillation. Dynamics of Atmospheres and Oceans, 42(1-4):152215).

## 2 Files

All .m files are in a folder (called matlabfiles or other), and executed with matlab:

- The most important files are:

skelmain.m: the main file (for any operation, edit and run skelmain with matlab)

skelrun_stocha.m: runs the skeleton model

skelini_setup.m (where setup=default, WP...): setup file giving all parameters of one simulation. This file is called to initiate simulations and also for post-processing.

- Other files are:

some recurrent functions (fftispe..filterkw...ncdfgetvar...singlecolumno...)

some post-processing files (skelg_...., skelgs_....) for making graphs,

## 3 Making a simulation

We will make a brief tutorial simulation with the setup "default". The setup is unique to each simulation. The setup "default" is for testing purpose with a meridional truncation M=1, and zonally homogeneous parameters. Here the corresponding files/keys are already created, but the same protocol is followed for every new simulation :

- create skelini_default.m, containing all parameters

- create a directory datadefault.

- in skelmain.m, check that setup='default';

- we will run a simulation with this setup, from loopstart=1 to loopend=2. Those are the starting and ending restart files of the simulation. If we wanted to continue this simulation, we will resimulate with say loopstart=3 and loopend=10.

- put doverylongrun=1 in skelmain do allow (re)simulating.

- execute skelmain.m and wait for simulation to finish. This will use skelrun_stocha.m which simulates the skeleton.

- this will create two restart files in the datadefault folder.

- put doverylongrun=0 in skelmain do avoid (re)simulating, and see next section for post-processing.

- You may latter on consider the setups HM (homogeneous) or WP (warm pool) that correspond to TMS2013, or create your own.

# 4 Reading the outputs

Now go to the datadefault folder and read by hand the restart files. Those are sequential restart files in netcdf, with index i. In one restart file, say skeleton_1.nc you have (cd ncdump -h skeleton_1.nc):

- dimensions: X = 64 ; T = 2879; M = 1 ; Y = 1 ; one = 1 ;

The dimension X is zonal position, with 64 points ($\Delta x$=625 km) around the equatorial belt (40,000 km). The dimension T is time. Each restart file spans around 2000 days, with around 2879 points. The outputs are sampled around each 17 hours (i.e. each $mts\,\Delta T$ , here mts=10, where$\Delta T \approx 1.7$ hours is the simulation timestep, see the JAS article). M=1, Y=1 are dimensions to treat the meridional dependency (e.g., zonal strips or Hermite i.e. cylinder functions), but are not relevant here. one=1 is a dummy dimension

- variables: double Ks(T, X) ; double Rms(T, M, X) ; double etas(T, Y, X) ; double Zs(T, Y, X) ; double ntaus(T, Y, X) ; double ts(T, one) ;

Ks is the variable $K$ (Kelvin amplitude, non-dimensional units). Rms is the variable $R$ (Rossby amplitude, non-dimensional units). etas is the variable $\eta$ at the equator (non-dimensional units). Zs is the variable $Z$ at the equator (non-dimensional units). ts is the time $t$, in days from the beginning of the simulation.

**Reconstruction:**

Typically, some good variables to reconstruct are $u$, $\theta$, $q$ and $\overline{H}a$ taken at the equator ($y = 0$), in dimensional units (those are the variables plotted in TMS2013). Here is how you could reconstruct them by hand. Otherwise, there are some post-processing files (see next section) where those computations are already done:

- To reconstruct $\overline{H}a$ in dimensional units (Kday-1): Had=HH*DDA*oa/(ta/oneday)*etas; (this is because $a = \Delta a\,\eta$ in non-dimensional units). Here: oneday=24*3600, ta=8*onehour with onehour=3600 (time dimensional scale, in seconds), oa=15 (temperature dimensional scale, in Kelvin), DDA=0.001 amplitude of stochastic transition $\Delta a$), and HH=0.22 ($\overline{H}$ in non-dimensional units)

- To reconstruct $u$ in dimensional units (ms-1): use $u = (K/\sqrt{2} - R/4)\phi_0 + R\phi_2/4\sqrt{2}$ taken at $y = 0$, in nondimensional units. Then dimensionalize using ud=u*ua with ua=50 (in ms-1). Beware this formula of the code does not correspond to the formula of TMS2013, because the scalings of $K$ and $R$ used in practice are rather the ones from BM2006, with notably $\phi_0(y) = \sqrt{2}(4\pi)^{-1/4}exp(-y^2/2)$ and $\phi_2(y) = (2y^2-1)(4\pi)^{-1/4}exp(-y^2/2)$. Also in the code the $\phi_2$ component is omitted when reconstructing.

- To reconstruct $\theta$ in dimensional units (K): use $\theta = (-K/\sqrt{2} - R/4)\phi_0 - R\phi_2/4\sqrt{2}$ at $y = 0$, in nondimensional units. Then dimensionalize using thetad=theta*oa with oa=15 (in K). In the code the $\phi_2$ component is omitted when reconstructing.

- To reconstruct $q$ in dimensional units (K): use $q = Z - \overline{Q}\theta$ in nondimensional units, with $\overline{Q} = 0.9$. Then dimensionalize using qd=q*oa with oa=15 (in K).

# 5 Post-Processing

**Main programs:**

The file skelmain.m contains a list of progams for post-processing (i.e. making graphs). Once the simulation is complete, edit skelmain.m with doverylongrun=0 and a call to the corresponding program. The most useful ones are:

- skelg_stabnew: computes linear stability (figure 1 in TMS2013).

- skelg_hovquick: does x-t hovmuller of variables at the equator and data projection (see TMS2013).

- skelg_plotquick: plots timeseries at a single location (useful to see if very long runs reach a statistical equilibria).

- skelg_kwcontourconca: plots k-w spectras of variables at the equator (see TMS2013).

- skelg_stabprojconcanew: hovmuller of data projection envelope (figure 5 in TMS2013).

Those programs should work with no adaptation for the setup default, HM and WP. Note that those programs recurrently call the skelini file for parameters, and need to have indexrestart (or ilpmin, ilpmax) specified in the skelmain file in order to know which restart files to use.

**Recurrent functions:**

There are some reccurent functions that are used, here are some notes on what they do:

- ffspe.m....: zonal Fourier transform.

- filterkw: filters data in the k-w domain.

- singlecolumnf and singlecolumnfm: this transforms data from spectral space (Hermite coefficients) to physical space (values on zonal strips), and inversely. Associated files are hermitefunc.m, hermitegauss.m, etc... that define the spectral and physical space, and singlecolumno, singlecolumnu... that permit to recontruct variables. See also the skelini file for grids definitions. This builds on a complex formalism between spectral/physical space that is detailed in RLW2013. This formalism is intended for solving flows with high meridional truncation (e.g. $M = 5$). It is nevertheless used for the simpler meridional truncation $M = 1$ (one Hermite coefficient/zonal strip) of TMS2013.

**Side-notes for $M = 1$:**

There are some additional programs that I let in the folder in case (skelg..skelgs...varsmjo...). Note that those are mostly intended at analysing outputs for a meridional truncation $M = 5$, which I currently work on, and may need some adaptation for the case $M = 1$. In particular, when using a truncation $M = 1$ (nyk=1 in skelini...) we reconstruct using only the $\phi_0$ component for all variables. Therefore, to reconstruct $v$ (with $\phi_1$ component) or the complete $u$ and $\theta$ (with $\phi_0$ and $\phi_2$ components) some adaptation is required, by playing with the singlecolumnf and singlecolumnfm functions (e.g. create a new Hermite space psima with nyka=3 and use singlecolumnu or singlecolumnum with this new set).