Manual for Running the Tropical Stochastic Skeleton GCM Model

Sulian Thual, (CIMS)

May 9, 2018

This manual provides instructions for simulating and reading the outputs of the Tropical Stochastic Skeleton GCM model (TSS-GCM model). For references see TMC2018:

- Thual, Majda, Chen 2018: A Tropical Stochastic Skeleton Model for the MJO, El Nino and Dynamic Walker Circulation: A Simplified GCM. Submitted to Journal of Climate.

The manual is organized as follows. Section 1 provides the model formulation. Section 2 is details how to run the model.

1 Model Formulation

This section details the formulation of the TSS-GCM model. This includes a presentation of the model equations, their truncated version for numerical solving, and all parameter definitions and values. See TMC2018 for more details.

1.1 TSS-GCM Model

The TSS-GCM model reads:

Intraseasonal Atmosphere

$$\begin{aligned}
&(\partial_t + d_u)u' - yv' - \partial_x\theta' = 0\\
&yu' - \partial_y\theta' = 0\\
&(\partial_t + d_u)\theta' - (\partial_x u' + \partial_y v') = \overline{H}a'\\
&(\partial_t + d_q)q' + \overline{Q}(\partial_x u' + \partial_y v') = -\overline{H}a' + \sigma_q \dot{W}_q\\
&\partial_t a' = \Gamma q'(\overline{a} + a') - \lambda a' + \sqrt{\lambda(\overline{a} + a')\overline{a}}\dot{W}_a.
\end{aligned} \tag{1}$$

Interannual atmosphere

$$\begin{aligned}
&-y\overline{v} - \partial_x\overline{\theta} = 0\\
&y\overline{u} - \partial_y\overline{\theta} = 0\\
&-(\partial_x\overline{u} + \partial_y\overline{v}) = \overline{H}\overline{a} - s^\theta\\
&-\overline{Q}(\partial_x\overline{u} + \partial_y\overline{v}) = \overline{H}\overline{a} + s^q + E_q\\
&\overline{H}\overline{a} = (E_q + s^q - \overline{Q}s^\theta)/(1 - \overline{Q})
\end{aligned} \tag{2}$$

Ocean

$$\begin{aligned}
&\partial_t U - \epsilon c_1 YV + \epsilon c_1 \partial_x H = \epsilon c_1 \tau_x\\
&YU + \partial_Y H = 0\\
&\partial_t H + \epsilon c_1(\partial_x U + \partial_Y V) = 0
\end{aligned} \tag{3}$$

SST

$$\partial_t T = -\epsilon c_1 \zeta E_q + \epsilon c_1 \eta H \tag{4}$$

Couplings

$$\begin{aligned} \tau_x &= \gamma(\overline{u} + u') \\ E_q &= \alpha_q T \end{aligned} \tag{5}$$

In the atmosphere (Eq. 1-2), the flow u is decomposed into an interannual component $\overline{u}$ and intraseasonal component u', with $u = \overline{u} + u'$, etc. The x is zonal direction, y is meridional direction and t is intraseasonal time. The u, v are zonal and meridional winds, θ is potential temperature, q is lower lever moisture and a is the planetary envelope of convective activity. The E_q is latent heat release from the ocean.

In the ocean (Eq. 3), Y is meridional direction. The U, V, are zonal and meridional currents, H is thermocline depth, τ_x is zonal wind stress. In the SST budget (Eq. 4), T is SST. The Eq. 5 provides couplings between the ocean and atmosphere.

The atmosphere extends over the entire equatorial belt $0 \leq x \leq L_A$, while the ocean Pacific extends from $0 \leq x \leq L_O$, with $L_O < L_A$. There are periodic boundary conditions in the atmosphere $u(0, y, t) = u(L_A, y, t)$, etc and reflection boundary conditions in the ocean $\int_{-\infty}^{+\infty} U(0, Y, t)dY = 0$ and $U(L_O, Y, t) = 0$.

1.2 Meridional Truncation

1.2.1 Parabolic Cylinder Functions

For numerical solving, the TSS-GCM from Eq. 1-5 is projected meridionally and truncated using the parabolic cylinder functions in the ocean and atmosphere. The parabolic cylinder functions in the ocean and atmosphere differ by their meridional extent, or Rossby radius,as shown in Fig, 1. The first atmosphere parabolic cylinder functions read $\phi_0(y) = (\pi)^{-1/4} exp(-y^2/2)$, $\phi_2 = (4\pi)^{-1/4}(2y^2 - 1)exp(-y^2/2)$, while the ocean parabolic cylinder functions read $\psi_m(Y)$, identical to the previous expression except depending here on the Y axis, with $Y = y/\sqrt{c}$. In order to couple the ocean and atmosphere basis, projection coefficients are introduced that reads $\chi_A = \int_{-\infty}^{+\infty} \phi_0(y)\phi_0(y/\sqrt{c})dy$ and $\chi_O = \int_{-\infty}^{+\infty} \psi_0(Y)\psi_0(\sqrt{c}Y)dY$.

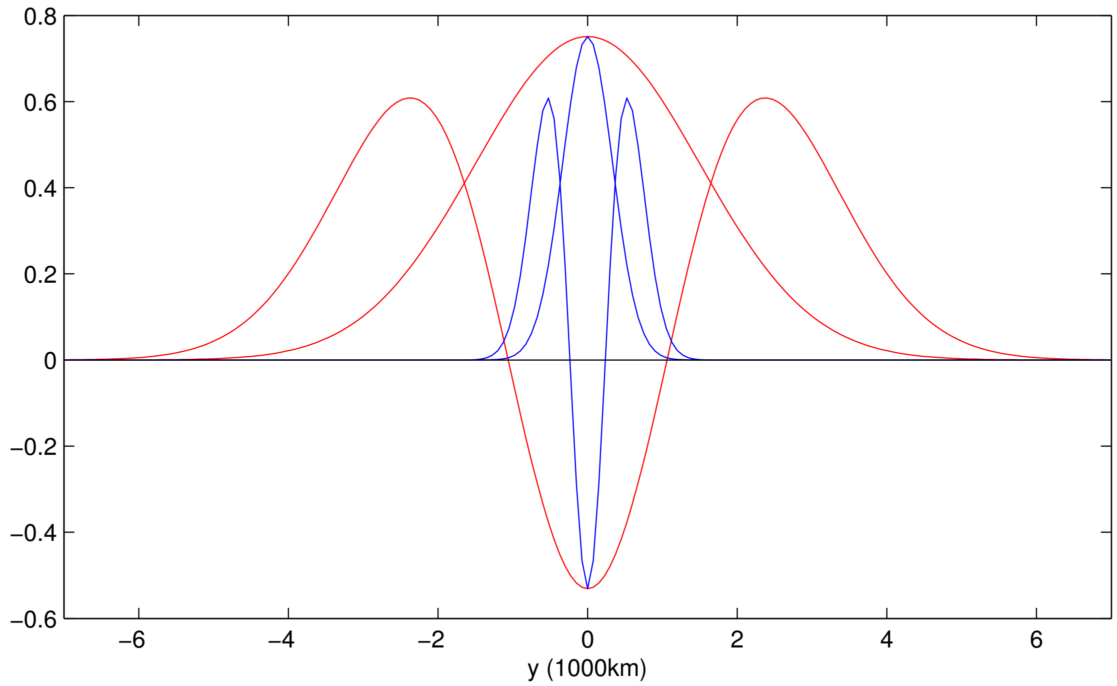

Figure 1: Meridional profiles of atmosphere parabolic cylinder functions ϕ_0, ϕ_2(red) and ocean parabolic cylinder functions ψ_0, ψ_2(blue), as a function of meridional position y (1000km).

1.2.2 Projection and Truncation

In the atmosphere we assume a truncation to the first parabolic cylinder function ϕ_0, $\{a, q, s^\theta, s^q\} = \{a, q, s^\theta, s^q\}\phi_0(y)$ (with a slight abuse of notations), known to excite the Kelvin and first Rossby atmospheric equatorial waves of amplitude K_A and R_A. The winds and potential temperature decompose as $u = K_A - R_A$ and $\theta = -K_A - R_A$. In the ocean we assume a truncation $\tau_x = \tau_x\psi_0$ known to excite only the the Kelvin and first Rossby oceanic waves of amplitude K_O and R_O. The current and thermocline depth deocmpose as $U = K_O - R_O$ and $H = K_O + R_O$. Similarly, for the SST we assume $T = T\psi_0$.

The TSS-GCM Model from Eq. 1-5 projected and truncated on the first parabolic cylinder functions reads:

Intraseasonal Atmosphere

$$\begin{aligned}
&(\partial_t + d_u)K'_A + \partial_x K'_A = -\overline{H}a'/2 \\
&(\partial_t + d_u)R'_A - \partial_x R'_A/3 = -\overline{H}a'/3 \\
&(\partial_t + d_q)q' + \overline{Q}(\partial_x K'_A - \partial_x R'_A/3) = -\overline{H}a'(1 - \overline{Q}/6) + \sigma_q \dot{W}_q \\
&\partial_t a' = \Gamma q'(\overline{a} + a') - \lambda a' + \sqrt{\lambda(\overline{a} + a')\overline{a}}\dot{W}_a
\end{aligned} \tag{6}$$

Interannual atmosphere

$$\begin{aligned}
&\partial_x \overline{K}_A = -\chi_A(E_q - < E_q >)(2 - 2\overline{Q})^{-1} \\
&-\partial_x \overline{R}_A/3 = -\chi_A(E_q - < E_q >)(3 - 3\overline{Q})^{-1} \\
&\overline{K}_A(0, \tau) = \overline{K}_A(L_A, \tau) \\
&\overline{R}_A(0, \tau) = \overline{R}_A(L_A, \tau) \\
&\overline{H}\overline{a} = (E_q - < E_q > + s^q - \overline{Q}s^\theta)/(1 - \overline{Q})
\end{aligned} \tag{7}$$

Ocean

$$\begin{aligned}
&\partial_t K_O + \epsilon c_1 \partial_x K_O = \epsilon \chi_O c_1 \tau_x/2 \\
&\partial_t R_O - \epsilon(c_1/3)\partial_x R_O = -\epsilon \chi_O c_1 \tau_x/3 \\
&K_O(0, t) = r_W R_O(0, t) \\
&R_O(L_O, t) = r_E K_O(L_O, t)
\end{aligned} \tag{8}$$

SST

$$\partial_t T = -\epsilon c_1 \zeta E_q + \epsilon c_1 \eta(K_O + R_O) \tag{9}$$

Couplings

$$\begin{aligned}
&\tau_x = \gamma(\overline{K}_A - \overline{R}_A + K'_A - R'_A) \\
&E_q = \alpha_q T
\end{aligned} \tag{10}$$

In the above Eq. 6, we consider in practice the variable change $Z' = q' + \overline{Q}\theta'$ (i.e. $Z' = q' - \overline{Q}(K'_A + R'_A)$) which leads to the modified equations (using $d_u = d_q$):

$$\begin{aligned}
&(\partial_t + d_q)Z' = -\overline{H}a'(1 - \overline{Q}) + \sigma_q \dot{W}_q \\
&\partial_t a' = \Gamma(Z' + \overline{Q}(K'_A + R'_A))(\overline{a} + a') - \lambda a' + \sqrt{\lambda(\overline{a} + a')\overline{a}}\dot{W}_a
\end{aligned} \tag{11}$$

In the above Eq. 7, the absence of dissipation in the interannual atmosphere imposes a peculiar solvability condition of a zero equatorial zonal mean of latent heating. This is enforced by removing the zonal mean:

$$< E_q > = \frac{1}{L_A}\int_0^{L_A} E_q dx \tag{12}$$

1.2.3 Flow Reconstruction

In order to reconstruct back the physical flow from the projected and truncated system in Eq. 6-10, we use $\{a, q\} = \{a, q\}\phi_0(y)$, $\tau_x = \tau_x\psi_0$ and $T = T\psi_0$ (with a slight abuse of notations), as well as:

$$\begin{aligned}
u' &= (K'_A - R'_A)\phi_0 + (R'_A/\sqrt{2})\phi_2 \\
\theta' &= -(K'_A + R'_A)\phi_0 - (R'_A/\sqrt{2})\phi_2 \\
\overline{u} &= (\overline{K}_A - \overline{R}_A)\phi_0 + (\overline{R}_A/\sqrt{2})\phi_2 \\
\overline{\theta} &= -(\overline{K}_A + \overline{R}_A)\phi_0 - (\overline{R}_A/\sqrt{2})\phi_2 \\
U &= (K_O - R_O)\psi_0 + (R_O/\sqrt{2})\psi_2 \\
H &= (K_O + R_O)\psi_0 + (R_O/\sqrt{2})\psi_2
\end{aligned} \tag{13}$$

where the use of ϕ_2 and ψ_2 is optional in the reconstruction.

1.3 Model Parameters

All reference scales and parameters are listed in the matlab file ensoini_params_tssgcm.m. The dimensional reference scales are x: 15000 km, y : 1500km, Y : 330km, t: 3.3 days, τ : 33 days u: $5\,m.s^{-1}$, θ, q: 1.5 K. Table 1 lists all parameters definitions and their nondimensional values.

Parameter	nondimensional value
c ratio of ocean/atmosphere phase speed	0.05
ϵ Froude number	0.1
$c_1 = c/\epsilon$	0.5
L_A equatorial belt length	8/3
L_O equatorial Pacific length	1.2
$\overline{H}$ convective heating rate factor	22
$\overline{Q}$ mean vertical moisture gradient	0.9
Γ convective growth/decay rate	1.66
α_q latent heating factor	0.2
γ wind stress coefficient	6.53
ζ latent heating exchange coefficient	8.7
η profile of thermocline feedback	$\eta(x) = 1.5 + (0.5\tanh(7.5(x - L_O/2))$
$d_a, d_q, d_\theta, \lambda$ atmosphere dissipations	0.11
σ_q moisture noise amplitude	0.4
s^q external moistening source	$s^q = 2.2(1 + 0.6\cos(2\pi x/L_A))$
s^θ external cooling source	$s^\theta = s^q$ except Walker circulation: $s^\theta = 2.2(1 + 0.6\cos(2\pi x/L_A - 0.1))$
r_E eastern boundary reflection	1
r_W western boundary reflection	0.5

Table 1: Model parameter definitions and nondimensional values.

2 Running The Code

2.1 Overview of Files

The code is written in matlab, and all .m files are in the “matlabfiles” folder. The type of files differ by their prefix:

ensoini_launch: The master file with calls to all other functions.

ensoini_params: Parameter files (one for each simulation). There are currently four parameter files for each type of model setup (TSS-GCM, Crude Interannual, Crude Intraseasonal, and TSS-GCM with Walker Circulation).

ensonum: Include all files needed to run a simulation and writting outputs. This includes ensonum_initial (computes or read initial conditions for a given restart file), ensonum_linear (computes the linear matrix for the interannual atmosphere and ocean, this is used for initial conditions), ensonum_loop (computes solutions for one restart file), ensonum_oldatm (computes the balanced interannual atmosphere), ensonum_wavefourier (computes the evolution of equatorial waves in the intraseasonal atmosphere)

ensoplot: All files useful for post-processing and making analysis figures of a simulation.

ensospe: technical files used

2.2 A Quick Example

We will make a brief tutorial for running and analyzing a simulation with the setup "tssgcm". The setup "tssgcm" corresponds to the complete TSS-GCM model from TMC2018. Most of the steps below are already prepared in the provided files, but they apply if a new setup is created with a different name.

1) Creating a new setup:

-Create a setup file ensoini_params_tssgcm: The file contains all parameters for the setup.

- Create a folder within the matlabfiles folder called data_tssgcm: this will contain all outputs from the simulation in netcdf format.

2) Editing the main file:

- Open and edit ensoini_launch: in the first paragraph "Setup", uncomment the "tssgcm" setup and comment all other setups. Set firstrestart=1 and choose a positive value for lastrestart (here 3). The setup is then loaded.

- In the second paragraph "Numerical Simulations" , set dolaunch=1 to recompute solutions. The code computes and store model solutions sequentially in a serie of netcdf files called "restart files", indexed from 1 to infinity. Each restart file contains around 4 years of simulation data (4000 timesteps). For dolaunch=1 numerical solutions will be (re)computed from firstrestart to lastrestart.

- In the third paragraph "Analysis", make sure to uncomment everything for now.

3) Running a simulation:

- open matlab in the matlabfiles directory (on linux, you can use "matlab -nodesktop" from a terminal to use the command line system only)

- run ensoini_launch and wait for simulation to compute (around 10 minutes). The restart files are stored in the folder data_tssgcm.

4) Analysing a simulation:

- Reopen and edit ensoini_launch: In the second paragraph "Numerical Simulations" set dolaunch=0 such that solutions are not recomputed anymore

- In the third paragraph "Analysis", uncomment ensoplot_timeserie_efficient: this analysis code will plot timeseries for the current numerical simulation.

- Run ensoini_launch again to do the analysis.

5) Extending a simulation.

- It is possible to extend the length of a simulation. Here for example edit ensoini_launch with dolaunch=1, firstrestart=3 and lastrestart=10. This will extend the current simulation "tssgcm" from the restart file 3 to 10.

2.3 Analysing Simulation outputs

The data files created when running the above simulation are in the folder data_tssgcm. They are in netcdf format. In linux shell, the netcdf files can be read using ncdump (e.g. "ncdump -h ensomjo_1.nc"). Each ensomjo_i.nc contains the data from restart, with the following structure:

- Xs: matrix containing the entire system state except $\overline{a}$, with the structure $X_s = \{K'_A, R'_A, a', Z', K_O, R_O, T\}$. The files ensospe_ncdfgetvar.m and ensospe_ncdmakevar.m are useful to read and write the netcdf files. To read Z' in the restart file 1 for example, use the commands: Xs=ncdfgetvar('ensomjo_1.nc','Xs'); Z=Xs(ixZ:nxZ,:); The index ixZ, nxZ can be found in ensonum_loop.m for each variable.

- ts: time axis for the restart file, in nondimensional units. The system is updated each 0.8 hours (timestep dt), but data is saved each 8 hours (=dt*mts with mts=10).

- MMAs: vector containg the variable $\overline{a}$.

For analysing the model outputs, some examples are provided in the ensoplot files. The provided files include a computation of time-averaged solutions, power spectra and hovmollers. All of those files do not modify the simulation files in any way. Other analysis files can be created and launched (using the ensoplot_ prefix for consistency) by the user.

3 Numerical Algorithm

This section details the numerical algorithm used to solve the system, as well as the associated structure of the code.

3.1 Overview

The numerical algorithm for the complete TSS-GCM model is performed in ensonum_loop (over the length of one restart file). We use a simple split method to update the system. The interannual atmosphere and ocean are solved using the method of lines in space and Euler in time. The atmosphere variables A' and Z' are solved using an Euler method in time. Finally, the intraseasonal equatorial waves K'_A, R'_A are solved exactly using a Fourier method.

In the file ensonum_loop, there are also options to solve instead cruder versions of the TSS-GCM model (crude interannual or crude intraseasonal), which is not detailed here. The option simutype (=1,2, or 3) in ensoini_param determines the version used (=3 for the TSS-GCM model).

3.2 Numerical Algorithm Intraseasonal Atmosphere

The numerical algorithm for the intraseasonal atmosphere (Eq. 6 with variable change from Eq. 11) is performed in ensonum_loop.m and ensonum_wavefourier.m.

3.2.1 Intraseasonal Atmosphere Waves

For solving the intraseasonal atmospheric waves K'_A and R'_A (that satisfy periodic boundary conditions on the equatorial belt of length L_A) we consider a zonal Fourier Transform. The interannual atmosphere model is discretized spatially on the grid $i = 1, ..., n_A$ $(n_A = 64)$, which reads $A'(x_i) = A'_i$, etc with $x_i = i\Delta x$ and $\Delta x = 625\,km$. For any variable $f(x,t)$defined on the zonal grid $x_i = i\Delta x$ with $f(0,t) = f(L_A,t)$, the Discrete Fourier Transform and Discrete Inverse Fourier Transform read:

$$F(k,t) = \sum_{i=1}^{N} f(x_i,t)\exp(-j2\pi ki/N) \tag{14}$$

$$f(x_i,t) = \frac{1}{N}\sum_{k=0}^{N-1} F(k,t)\exp(j2\pi ki/N) \tag{15}$$

where j is the complex number and $k = 0, ...N-1$ is an integer. In practice we use the Fast Fourier Transform (with N even).

The evolution of the equatorial waves K'_A and R'_A have the following generic form:

$$(\partial_t + e)f + c\partial_x f = p \tag{16}$$

where f is either K'_A or R'_A. The value of f at time $t+\Delta t$ solved exactly in Fourier space using:

$$F(k,t+\Delta t) = F(k,t)\exp(-e\Delta t - 2\pi jkc\Delta t) + \frac{P(k,t)}{e+2\pi jkc}(1-\exp(-e\Delta t - 2\pi jkc\Delta t)) \tag{17}$$

$$F(0,t+\Delta t) = F(0,t) + P(0,t)\Delta t, \text{ for } k=0 \text{ and } e=0 \tag{18}$$

where P is the Discrete Fourier Transform of p.

3.2.2 Intraseasonal Atmosphere convection and moisture

Meanwhile, the convection a' and moisture variable Z' (cf Eq. 6 with variable change from Eq. 11)) are solved in ensonum_loop.m. The time derivatives are discretized using a simple Euler method, and there are no spatial derivatives thanks to the variable change. The white noise source $\dot{W}_q$ is discretize according to the Ito lemma.

The Z' conditions at time $t + \Delta t$ are computed from the conditions at time t as:

$$Z'(t+\Delta t) = Z'(t) + \Delta t(-d_q Z' - \overline{H}a'(1-\overline{Q}) + \sigma_q \frac{r_q}{\sqrt{\Delta t}}) \tag{19}$$

where r_q is a random number with Gaussian distribution. Meanwhile, the a' conditions at time $t + \Delta t$ are computed from the conditions at time t as:

$$\partial_t a'(t+\Delta t) = a'(t) + \Delta t[\Gamma(Z' + \overline{Q}(K'_A + R'_A))(\overline{a} + a') - \lambda a' + \sqrt{\lambda(\overline{a}+a')\overline{a}}\frac{r_a}{\sqrt{\Delta t}}] \tag{20}$$

where r_a is a random number with Gaussian distribution.

3.3 Numerical Algorithm Interannual Atmosphere

The numerical algorithm for the interannual atmosphere (Eq. 7) is in ensonum_loop.m and ensonum_oldatm.m. The interannual atmosphere model is discretized spatially on the grid $i = 1, ..., n_A$ ($n_A = 64$) , which reads $A'(x_i) = A'_i$, etc with $x_i = i\Delta x$ and $\Delta x = 625\,km$. The interannual winds are computed as $\overline{u} = 3W_A/2$ with:

$$d_N W_{Ai} + \frac{(W_{Ai+1} - W_{Ai})}{\Delta x} = \frac{-3\chi_A}{2(1-\overline{Q})}(E_{qi} - \sum_{i=1}^{n_A} E_{qi}\frac{\Delta x}{L_A}) \tag{21}$$

where $\overline{K}_A = (1/3)W_A$ and $\overline{R}_A = (-2/3)W_A$ (and $\overline{u} = \overline{K}_A - \overline{R}_A$). Solving the atmospheric response in equation (21) is akin to solving a matrix system $\mathbf{A}X = B$, where $X = \{W_{A1}, ..., W_{An_A}\}$. The zonal mean of E_q is removed in order to satisfy solvability condition and the small dissipation $d_N = 10^{-8}$ ensures that the matrix $\mathbf{A}$ is invertible without affecting solutions. This operation is performed in ensonum_oldatm.m (called by ensonum_loop.m).

In ensonum_loop.m , we also compute interannual convection as:

$$\overline{H}\overline{a} = (E_q - < E_q > + s^q - \overline{Q}s^\theta)/(1-\overline{Q}) \tag{22}$$

where we ensure that $\overline{a} \geq 0$ (which is enforced by imposing $\overline{a} = 10^{-5}$ if $\overline{a} \leq 0$).

3.4 Numerical Algorithm Ocean and SST

The numerical algorithm for the ocean and SST (Eq. 8-9) is in ensonum_loop.m. The ocean domain is discretized spatially on the grid $i = 1, ..., n_O$ ($n_O = 28$), with $x_i = i\Delta x$ and $\Delta x = 625\,km$. The ocean domain covers the equatorial Pacific only: the ocean and atmosphere grids overlap for $1 \leq i \leq n_O$, while the atmosphere grid for $n_O < i \leq n_A$ covers the rest of the equatorial belt with no underlying ocean.

The ocean derivatives are discretized using upwind schemes depending on the direction of propagation, $[\partial_x K_O]_i = (K_{Oi} - K_{Oi-1})/\Delta x$, $[\partial_x R_O]_i = (R_{Oi+1} - R_{Oi})/\Delta x$, with reflection boundary conditions accounted for as $[\partial_x K_O]_1 = (K_{O1} - r_w R_{O_1})/\Delta x$ and $[\partial_x R_O]_{n_O} = (r_E K_{On_O} - R_{On_O})/\Delta x$.

The time derivatives in both the ocean and SST are discretized using a simple Euler method. For example, the SST conditions at time $t + \Delta t$ is computed from the conditions at time t as:

$$T(t+\Delta t) = T(t) - \Delta t \epsilon c_1 \zeta E_q + \Delta t \epsilon c_1 \eta (K_O + R_O) \tag{23}$$