Technical Note for the Community Recharge Oscillator (CRO) Model Package for the ENSO Winter School 2025

<CRO Code Distribution Team>

27 February 2025

1. Introduction

The CRO code package is an easy-to-use Python/MATLAB software for solving and fitting the ENSO RO model. The CRO code is currently under development and is planned for release in 2025. The distributed version for the ENSO Winter School 2025 is a light Python version that includes only the essential features.

2. Master RO

The CRO code supports the master RO equation, which can consider various types of RO models. The master RO equation is given as:

$$\frac{dT}{dt} = RT + F_1 h + b_T T^2 - c_T T^3 + d_T Th + \sigma_T N_T(1 + g_T) + E_T$$

$$\frac{dh}{dt} = -F_2 T - \varepsilon h - b_h T^2 + \sigma_h N_h + E_h$$

$$\frac{d\xi_T}{dt} = -m_T \xi_T + \sqrt{2m_T} w_T$$

$$\frac{d\xi_h}{dt} = -m_h \xi_h + \sqrt{2m_h} w_h$$

$$N_T = \begin{cases} \xi_T, if \;\; n_T = 0 \\ w_T, if \;\; n_T = 1 \end{cases}$$

$$N_h = \begin{cases} \xi_h, if \;\; n_h = 0 \\ w_h, if \;\; n_h = 1 \end{cases}$$

$$g_T = \begin{cases} BT, if \;\; n_g = 0 \\ BH(T)T, if \;\; n_g = 1 \end{cases}.$$

T and h are monthly SST and thermocline depth anomalies, respectively. w_T and w_h are uncorrelated Gaussian white noise forcing with zero mean and unit variance. E_T and E_h

are external forcing. The RO parameters and variables are listed in the table below.

Class		Notation (equation)	Notation (code)	Unit
Deterministic Parameter	Linear parameter	R	R	month^{-1}
		F_1	F1	K m^{-1} month^{-1}
		F_2	F2	m K^{-1} month^{-1}
		ε	epsilon	month^{-1}
	Nonlinear parameter	b_T	b_T	K^{-1} month^{-1}
		c_T	c_T	K^{-2} month^{-1}
		d_T	d_T	m^{-1} month^{-1}
		b_h	b_h	K^{-2} m month^{-1}
Noise parameter		σ_T	sigma_T	K month$^{-1/2}$ ($n_T = 1$) K month^{-1} ($n_T = 0$)
		σ_h	sigma_h	m month$^{-1/2}$ ($n_h = 1$) K month^{-1} ($n_h = 0$)
		B	B	K^{-1}
		m_T	m_T	month^{-1}
		m_h	m_h	month^{-1}
Noise option parameter		n_T	n_T	no unit
		n_h	n_h	no unit
		n_g	n_g	no unit
Noise forcing	Red noise	ξ_T	xi_T	no unit
		ξ_h	xi_h	no unit
	White noise	w_T	w_T	month$^{-1/2}$
		w_h	w_h	month$^{-1/2}$
External forcing		E_T	E_T	K month^{-1}
		E_h	E_h	m month^{-1}
Variable		T	T	K
		h	h	m

The different combinations of parameter choices above allow us to consider a total of 24 types of RO by varying the deterministic terms (linear or nonlinear), noise coefficient (additive or multiplicative or multiplicative with Heaviside function), noise color (white or red), and inclusion of seasonality in parameters (fixed or seasonal). A detailed physical explanation of this master RO equation is not provided in this document, as it is beyond its scope.

2.1 Type of deterministic terms

The linear RO does not include nonlinear parameters at all ($b_T = c_T = d_T = b_h = 0$). The nonlinear RO includes at least one of the nonlinear parameters.

2.2 Type of noise color

The white noise RO includes white noise forcing for both T and h ($N_T = w_T,\ N_h = w_h$). The noise option parameter is set to $n_T = n_h = 1$. The red noise RO includes red noise forcing for both T and h ($N_T = \xi_T,\ N_h = \xi_h$). The noise option parameter is set to $n_T = n_h = 0$. We do not consider mixed white/red noise for T and h (e.g., $N_T = w_T,\ N_h = \xi_h$) for simplicity.

2.3 Type of noise coefficient

The additive noise RO includes additive noise forcing of T ($B = 0$). The multiplicative noise RO includes state-dependent noise forcing of T ($B \neq 0$). We consider two different types of state-dependent noise forcing: linear multiplicative forcing ($g_T = BT$) with noise option parameter of $n_g = 0$, and Heaviside linear multiplicative forcing ($g_T = BH(T)T$) with noise option parameter of $n_g = 1$.

2.4 Inclusion of seasonality

The seasonal RO includes seasonal variation in the deterministic and noise parameters. The CRO supports three different types of seasonality. For any deterministic and noise parameter X:

(1) Constant: X.

(2) Seasonal cycle: $X+X_a*\sin(\omega_a t+\varphi_{Xa})$ (where $\omega_a=2\pi/12$).

2.5 Summary table

By a combination of these different parameter settings, a total of 24 different types of RO are considered. The parameter for each RO type is summarized in the table below.

Num	RO Type	Acronym	Deterministic Parameters	Noise Parameters
1	Linear-White-Additive	LWA	$R,\ F_1,\ F_2,\ \varepsilon$	$\sigma_T,\ \sigma_h$
2	Linear-White-Multi	LWM	$R,\ F_1,\ F_2,\ \varepsilon$	$\sigma_T,\ \sigma_h,\ B$
3	Linear-White-Multi-Heaviside	LWM(H)	$R,\ F_1,\ F_2,\ \varepsilon$	$\sigma_T,\ \sigma_h,\ B$
4	Linear-Red-Additive	LRA	$R,\ F_1,\ F_2,\ \varepsilon$	$\sigma_T,\ \sigma_h,\ m_T,\ m_h$
5	Linear-Red-Multi	LRM	$R,\ F_1,\ F_2,\ \varepsilon$	$\sigma_T,\ \sigma_h,\ m_T,\ m_h,\ B$
6	Linear-Red-Multi-Heaviside	LRMH	$R,\ F_1,\ F_2,\ \varepsilon$	$\sigma_T,\ \sigma_h,\ m_T,\ m_h,\ B$
7	Nonlinear-White-Additive	NWA	$R,\ F_1,\ F_2,\ \varepsilon,\ b_T,\ c_T,\ d_T,\ b_h$	$\sigma_T,\ \sigma_h$
8	Nonlinear-White-Multi	NWM	$R,\ F_1,\ F_2,\ \varepsilon,\ b_T,\ c_T,\ d_T,\ b_h$	$\sigma_T,\ \sigma_h,\ B$
9	Nonlinear-White-Multi-Heaviside	NWM(H)	$R,\ F_1,\ F_2,\ \varepsilon,\ b_T,\ c_T,\ d_T,\ b_h$	$\sigma_T,\ \sigma_h,\ B$
10	Nonlinear-Red-Additive	NRA	$R,\ F_1,\ F_2,\ \varepsilon,\ b_T,\ c_T,\ d_T,\ b_h$	$\sigma_T,\ \sigma_h,\ m_T,\ m_h$
11	Nonlinear-Red-Multi	NRM	$R,\ F_1,\ F_2,\ \varepsilon,\ b_T,\ c_T,\ d_T,\ b_h$	$\sigma_T,\ \sigma_h,\ m_T,\ m_h,\ B$
12	Nonlinear-Red-Multi-Heaviside	NRM(H)	$R,\ F_1,\ F_2,\ \varepsilon,\ b_T,\ c_T,\ d_T,\ b_h$	$\sigma_T,\ \sigma_h,\ m_T,\ m_h,\ B$
13	Seasonal-Linear-White-Additive	SLWA	$R,\ R_a,\ \phi_{Ra},\ F_1,\ F_2,\ \varepsilon$	$\sigma_T,\ \sigma_h$
14	Seasonal-Linear-White-Multi	SLWM	$R,\ R_a,\ \phi_{Ra},\ F_1,\ F_2,\ \varepsilon$	$\sigma_T,\ \sigma_h,\ B$
15	Seasonal-Linear-White-Multi-Heaviside	SLWM(H)	$R,\ R_a,\ \phi_{Ra},\ F_1,\ F_2,\ \varepsilon$	$\sigma_T,\ \sigma_h,\ B$
16	Seasonal-Linear-Red-Additive	SLRA	$R,\ R_a,\ \phi_{Ra},\ F_1,\ F_2,\ \varepsilon$	$\sigma_T,\ \sigma_h,\ m_T,\ m_h$
17	Seasonal-Linear-Red-Multi	SLRM	$R,\ R_a,\ \phi_{Ra},\ F_1,\ F_2,\ \varepsilon$	$\sigma_T,\ \sigma_h,\ m_T,\ m_h,\ B$
18	Seasonal-Linear-Red-Multi-Heaviside	SLRM(H)	$R,\ R_a,\ \phi_{Ra},\ F_1,\ F_2,\ \varepsilon$	$\sigma_T,\ \sigma_h,\ m_T,\ m_h,\ B$
19	Seasonal-Nonlinear-White-Additive	SNWA	$R,\ R_a,\ \phi_{Ra},\ F_1,\ F_2,\ \varepsilon,\ b_T,\ c_T,\ d_T,\ b_h$	$\sigma_T,\ \sigma_h$
20	Seasonal-Nonlinear-White-Multi	SNWAM	$R,\ R_a,\ \phi_{Ra},\ F_1,\ F_2,\ \varepsilon,\ b_T,\ c_T,\ d_T,\ b_h$	$\sigma_T,\ \sigma_h,\ B$
21	Seasonal-Nonlinear-White-Multi-Heaviside	SNWM(H)	$R,\ R_a,\ \phi_{Ra},\ F_1,\ F_2,\ \varepsilon,\ b_T,\ c_T,\ d_T,\ b_h$	$\sigma_T,\ \sigma_h,\ B$
22	Seasonal-Nonlinear-Red-Additive	SNRA	$R,\ R_a,\ \phi_{Ra},\ F_1,\ F_2,\ \varepsilon,\ b_T,\ c_T,\ d_T,\ b_h$	$\sigma_T,\ \sigma_h,\ m_T,\ m_h$
23	Seasonal-Nonlinear-Red-Multi	SNRM	$R,\ R_a,\ \phi_{Ra},\ F_1,\ F_2,\ \varepsilon,\ b_T,\ c_T,\ d_T,\ b_h$	$\sigma_T,\ \sigma_h,\ m_T,\ m_h,\ B$
24	Seasonal-Nonlinear-Red-Multi-Heaviside	SNRM(H)	$R,\ R_a,\ \phi_{Ra},\ F_1,\ F_2,\ \varepsilon,\ b_T,\ c_T,\ d_T,\ b_h$	$\sigma_T,\ \sigma_h,\ m_T,\ m_h,\ B$

3. Methods

3.1 Parameter Fitting

Different methods can be applied to fit the RO parameters. The distributed CRO package uses the linear regression method for the RO parameter fitting. The full CRO package, planned for a later release, will support additional fitting methods.

Linear regression is the simplest and most widely used method to estimate RO parameters. The linear regression method discretizes the time-tendency term with a forward scheme as follows:

$$\frac{T_{i+1} - T_i}{\Delta t} = RT_i + F_1 h_i + b_T {T_i}^2 - c_T {T_i}^3 + d_T T_i h_i$$

$$\frac{h_{i+1} - h_i}{\Delta t} = -F_2 T_i - \varepsilon h_i - b_h {T_i}^2$$

The deterministic parameters are determined by the coefficient of the linear regression. For the linear RO, only linear parameters can be considered for the linear regression. For nonlinear RO, nonlinear parameters are together considered for the linear regression.

For seasonal RO, the seasonal variation in parameter X is added as additional sinusoidal terms as follows:

$$X_a sin(\omega_a t + \phi_{Xa})$$

However, fitting ϕ_{Xa} to given time series is not linear regression problem anymore and involves solving a nonlinear fitting problem, which is computationally challenging. To enable the parameter estimation using the linear regression method, the sinusoidal terms are transformed as:

$$X_{as} sin(\omega_a t) + X_{ac} cos(\omega_a t)$$

where

$X_a = \sqrt{{X_{as}}^2 + {X_{ac}}^2}$

$\phi_{Xa} = atan2(\frac{X_{ac}}{X_{as}})$

The values for X_{as} and X_{ac} are estimated using linear regression method. Once these values are obtained, they are converted back to the original parameters (X_a, ϕ_{Xa}) using the relationships above.

For example, the RO with annual cycle in R:

$$\frac{T_{i+1} - T_i}{\Delta t} = RT_i + R_a sin(\omega_a t + \phi_{Ra})T_i + F_1 h_i + b_T {T_i}^2 - c_T {T_i}^3 + d_T T_i h_i$$

$$\frac{h_{i+1} - h_i}{\Delta t} = -F_2 T_i - \varepsilon h_i - b_h {T_i}^2$$

The sinusoidal function is decomposed as:

$$\frac{T_{i+1} - T_i}{\Delta t} = RT_i + R_{as} T_i sin(\omega_a t) + R_{ac} T_i cos(\omega_a t) + F_1 h_i + b_T {T_i}^2 - c_T {T_i}^3 + d_T T_i h_i$$

$$\frac{h_{i+1} - h_i}{\Delta t} = -F_2 T_i - \varepsilon h_i - b_h {T_i}^2$$

where $R_a = \sqrt{{R_{as}}^2 + {R_{ac}}^2}$ and $\phi_{Ra} = atan(\frac{R_{ac}}{R_{as}})$.

The noise parameters are determined by the residual of the linear regressions (T_{res} and h_{res}). A different approach for noise parameter fitting is required for different types of RO.

3.1 White-Additive types

The regression residuals are parameterized as:

$$T_{res} = \sigma_T w_T$$

$$h_{res} = \sigma_h w_h$$

σ_T and σ_h can be simply estimated by the standard deviation of the residual.

$$\sigma_T = std(T_{res})$$

$$\sigma_h = std(h_{res})$$

3.2 White-Multiplicative types

The regression residuals are parameterized as:

$$T_{res} = \sigma_T w_T (1 + BT)$$

$$h_{res} = \sigma_h w_h$$

σ_h can be simply estimated as:

$$\sigma_h = std(h_{res})$$

Estimating σ_T and B has been a challenging problem. Many studies attempted different approaches to this. The key is finding the value of σ_T and B that can best explain the relationship between T_{res} and T.

The statistical relationship between T_{res} and T is (derivation is documented below):

$$var(T_{res}) = \sigma_T^2 + \sigma_T^2 B^2 var(T)$$

To determine σ_T and B, at least two or more $var(T_{res})$ and $var(T)$ should be given. We randomly sample non-consecutive segments of T_{res} and T from their original time series, and construct an ensemble of $var(T_{res})$ and $var(T)$. The length of the segment is 120 and the number of ensemble member is 10^5. σ_T and B are determined from the coefficient of the linear regression of Y against X, given by:

$$Y = pX + q$$

where $Y = var(T_{res})$ and $X = var(T)$. The expressions for σ_T and B are:

$$\sigma_T = \sqrt{q}$$

$$B = \sqrt{p/q}.$$

∗ Derivation

Variance of T_{res} is:

$$\begin{aligned} var(T_{res}) &= var(\sigma_T w_T + \sigma_T B w_T T) = var(\sigma_T w_T) + var(\sigma_T B w_T T) + 2cov(\sigma_T w_T, \sigma_T B w_T T) \\ &= \sigma_T^2 + \sigma_T^2 B^2 var(w_T T) + 2\sigma_T^2 B cov(w_T, w_T T) \end{aligned}$$

Here,

$$var(w_T T) = mean(w_T^2) mean(T^2) - \left(mean(T)\right)^2 \left(mean(w_T)\right)^2 = var(T)$$

$$\begin{aligned} cov(w_T, w_T T) &= mean(w_T^2 T) - mean(w_T) mean(w_T T) = mean(w_T^2 T) = mean(w_T^2) mean(T) \\ &= 0 \end{aligned}$$

Finally,

$$var(T_{res}) = \sigma_T^2 + \sigma_T^2 B^2 var(T)$$

3.3 White-Multi-Heaviside types

For the White-Multi-Heaviside noise type RO, the same fitting method is used as for the White-Multi type RO, but with a modified relationship to estimate σ_T and B:

$$var(T_{res}) = \sigma_T^2 + \sigma_T^2 B^2 var(H(T)T)$$

3.4 Red-Additive types

The regression residual of T is parameterized as:

$$T_{res} = \sigma_T \xi_T$$

$$\frac{d\xi_T}{dt} = -m_T \xi_T + \sqrt{2m_T} w_T$$

Note that the variance of ξ_T is 1.

σ_T is simply given as:

$$\sigma_T = std(T_{res})$$

To easily estimate m_T, we multiply the red noise equation by σ_T:

$$\sigma_T \frac{d\xi_T}{dt} = -m_T \sigma_T \xi_T + \sqrt{2m_T} \sigma_T w_T$$

Finally,

$$\frac{dT_{res}}{dt} = -m_T T_{res} + \sqrt{2m_T} \sigma_T w_T$$

m_T can be determined by the linear regression:

$$\frac{T_{res\ i+1} - T_{res\ i}}{\Delta t} = -m_T T_{res\ i}$$

Alternatively, m_T can be determined by lagged autocorrelation of T_{res}. It is nearly identical to the linear regression, and we do not discuss it here for conciseness.

The regression residual of h is given as:

$$h_{res} = \sigma_h \xi_h$$

$$\frac{d\xi_h}{dt} = -m_h \xi_h + \sqrt{2m_h} w_h$$

σ_h and m_h are estimated in the exact same way as the σ_T and m_T:

$$\sigma_h = std(h_{res})$$

$$\frac{h_{res\ i+1} - h_{res\ i}}{\Delta t} = -m_h h_{res\ i}$$

3.5 Red-Multi types

The regression residual of T is parameterized as:

$$T_{res} = \sigma_T \xi_T (1 + BT)$$

$$\frac{d\xi_T}{dt} = -m_T \xi_T + \sqrt{2m_T} w_T$$

Note that the variance of ξ_T is 1.

Same as the white-multiplicative RO case, B can be estimated as:

$$var(T_{res}) = \sigma_T^2 + \sigma_T^2 B^2 var(T)$$

We apply a random sampling method to estimate σ_T and B same as the white-multiplicative RO case.

Using the estimated B, we introduce the new variable, T'_{res}:

$$T'_{res} = \frac{T_{res}}{1 + BT}$$

Then, the red noise equation is written as:

$$\frac{dT_{res}{}'}{dt} = -m_T T_{res}{}' + \sigma_T \sqrt{2m_T} w_T$$

The m_T can be determined by the linear regression:

$$\frac{T_{res\ i+1} - T_{res\ i}}{\Delta t} = -m_T T_{res\ i}$$

The regression residual of h is given as:

$$h_{res} = \sigma_h \xi_h$$

$$\frac{d\xi_h}{dt} = -m_h \xi_h + \sqrt{2m_h} w_h$$

σ_h and m_h are estimated in the exact same way as the red-additive RO case:

$$\sigma_h = std(h_{res})$$

$$\frac{h_{res\ i+1} - h_{res\ i}}{\Delta t} = -m_h h_{res\ i}$$

3.6 Red-Multi-Heaviside types

For the Red-Multi-Heaviside noise type RO, the same fitting method is used as for the Red-Multi type RO, but with a modified relationship to estimate σ_T and B:

$$var(T_{res}) = \sigma_T^2 + \sigma_T^2 B^2 var(H(T)T)$$

3.2 RO Solving

The CRO numerically integrates the RO equation using standard methods for solving stochastic differential equations.

3.1 Numerical formulation

To perform the numerical integration of the master RO, we first convert the RO equation to the canonical form of a stochastic differential equation. The general canonical form is written as:

$$dX = f(X)dt + g(X)dW$$

where $f(X)$ is a deterministic term and $g(X)$ is a stochastic term. dW is an increment of the Wiener process. dW is a random variable with a normal distribution of zero mean and variance of dt.

(1) White Noise RO

A canonical formulation of white noise RO ($n_T = 1$ and $n_h = 1$) is written as:

$$dT = (RT + F_1 h + b_T T^2 - c_T T^3 + d_T Th + E_T)dt + [\sigma_T(1 + g_T)]dW$$

$$dh = (-F_2 T - \varepsilon h - b_h T^2 + E_h)dt + (\sigma_h)dW$$

(2) Red Noise RO

A canonical formulation of red noise RO ($n_T = 0$ and $n_h = 0$) is written as:

$$dT = [RT + F_1 h + b_T T^2 - c_T T^3 + d_T Th + E_T + \sigma_T(1 + g_T)\xi_T]dt$$

$$dh = (-F_2 T - \varepsilon h - b_h T^2 + E_h + \sigma_h \xi_h)dt$$

$$d\xi_T = (-m_T \xi_T)dt + \sqrt{2m_T}dW$$

$$d\xi_h = (-m_h \xi_h)dt + \sqrt{2m_h}dW$$

3.2 Numerical method for solving the master RO equation

The CRO offers two different numerical integration methods for the RO.

(1) Euler-Maruyama (EM) method

The EM method is based on the "Ito calculus" of a stochastic process. This is a stochastic extension of the Euler method for an ordinary differential equation. It is written as:

$$X_{i+1} = X_i + f(X_i)\Delta t + g(X_i)\Delta W_i$$

i denotes i th time step. Δt is a numerical time step. ΔW_i is a normal random variable with zero mean and variance of Δt (standard deviation of $\sqrt{\Delta t}$).

(2) Euler-Heun (EH) method

The EH method is based on the "Stratonovich calculus" of a stochastic process. This is a stochastic extension of the Heun's method for an ordinary differential equation. It is written as:

$$X_{i+1} = X_i + \frac{1}{2}\left(f(X_i) + f(\hat{X}_{i+1})\right)\Delta t + \frac{1}{2}\left(g(X_i) + g(\hat{X}_{i+1})\right)\Delta W_i$$

$$\hat{X}_{i+1} = X_i + f(X_i)\Delta t + g(X_i)\Delta W_i$$

i denotes i th time step. Δt is a numerical time step. ΔW_i is a normal random variable with zero mean and variance of Δt (standard deviation of $\sqrt{\Delta t}$). $\hat{X}_{i+1}$ is a supporting value for the numerical integral.

(3) Difference between the methods

The EM and EH methods are based on different stochastic calculus. The EM method is based on "Itô calculus" while the EH method is based on "Stratonovich calculus". The EH method has higher numerical accuracy and stability than the EM, but it is computationally more expensive.

Ito and Stratonovich calculus give an identical analytical solution of a stochastic differential equation if the noise is an additive (i.e., $B = 0$). However, if the noise is multiplicative (i.e., $B \neq 0$), Ito and Stratonovich calculus give a different analytical solution. Therefore, the numerical method should be carefully chosen in this case.

In a nutshell, Stratonovich calculus is more physically relevant while Ito is more mathematically relevant. Thus, for the ENSO modelling, Stratonovich calculus provides a mathematical framework that is better able to describe the ocean-atmosphere system than

Ito. Following this reason, we set EH as a default method for solving RO in the CRO. A detailed explanation of the difference between Itô and Stratonovich calculus can be found at https://www.robots.ox.ac.uk/~lsgs/posts/2018-09-30-ito-strat.html.

(4) Cautionary note

In practice, the most common mistake for numerical integration of RO is to treat the noise forcing equally as the deterministic term. For example, using the EM method, linear-white-additive RO is solved as:

$$T_{i+1} = T_i + (RT_i + F_1 h_i)\Delta t + \sigma_T (w_T *)_i \sqrt{\Delta t}$$
$$h_{i+1} = h_i + (-F_2 T_i - \varepsilon h_i)\Delta t + \sigma_h (w_h *)_i \sqrt{\Delta t}$$

where $w_T *$ and $w_h *$ are normal random variable with zero mean and unit variance. On the random noise term, $\sqrt{\Delta t}$ is multiplied not Δt. When $\Delta t = 1$, this mistreatment gives identical results with the correct method, thus it does not cause a problem. However, when $\Delta t \neq 1$, this mistreatment produces incorrect results.

4. CRO Code Structure

The CRO package consists of two main functions: the parameter fitting function (RO_fitting) and the solver function (RO_solver).

5. CRO Main Functions

5.1 Parameter Fitting

The RO_fitting function takes the T and h time series, as well as the RO type choices, as inputs, and generates the fitted values of all parameters.

Syntax

par=RO_fitting(T, h, T_option, h_option, noise_option)
par=RO_fitting(T, h, T_option, h_option, noise_option, method_fitting)
par=RO_fitting(T, h, T_option, h_option, noise_option, method_fitting, dt_fitting)

Description

• par=RO_fitting(T, h, T_option, h_option, noise_option) returns the fitted RO parameter from the given time series, T and h. The RO type can be specified by par_option_T, par_option_h, and par_option_noise. The fitting is performed with the default method (specified by the table at the end of Section 3.1)

• par=RO_fitting(T, h, T_option, h_option, noise_option, method_fitting) defines the fitting method specified by method_fitting.

• par=RO_fitting(T, h, T_option, h_option, noise_option, method_fitting, dt_fitting) specifies the temporal interval of input T and h time series, dt_fitting. When dt_fitting is 1.0 (default), the inputs T and h are monthly time series.

Input Arguments

• T-Input SST anomaly time series (vector; Nx1 or 1xN)
Input SST anomaly time series, specified as a vector.

• h-Input h anomaly time series (vector; Nx1 or 1xN)
Input h anomaly time series, specified as a vector.

• T_option-RO equation form for deterministic dT/dt part (dictionary with 5 key-value pairs)
Option for specifying the type of RO equation in the deterministic dT/dt part, specified as dictionary with 5 key-value pairs. Each element in the structure/dictionary (X_i) specifies the type of parameter in order of R, F1, b_T, c_T, and d_T.
T_option={"R": X_1, "F1": X_2, "b_T": X_3, "c_T": X_4, "d_T": X_5}
The elements of the dictionary (X_i, i=1,2,3,4,5) are one of these values:
 0-not considered (X_i=0)
 1-seasonally-constant (X_i=X)
 3-seasonally-varying-annual ($X_i=X+X_a\sin(wt+\varphi_a)$) ($w=2\pi/12$)

• h_option-RO equation form for deterministic dh/dt part (dictionary with 3 key-value pairs)
Option for specifying the type of RO equation in the deterministic dh/dt part, specified as dictionary with 3 key-value pairs. Each element in the structure/dictionary (X_i) specifies the type of parameter in order of F2, epsilon, b_h.
T_option={"F2": X_1, "epsilon": X_2, "b_h": X_3}
The elements of the dictionary (X_i, i=1,2,3) are one of these values:
 0-not considered (X=0)
 1-seasonally-constant (X=X)
 3-seasonally-varying-annual ($X=X+X_a\sin(wt+\varphi_a)$) ($w=2\pi/12$)

• noise_option-RO noise option parameter (dictionary with 3 key-value pairs)
Option for specifying noise type of the RO equation, specified as dictionary with 3 key-value pairs:
noise_option={"T": X_1, "h": X_2, "T_type": X_3}
X_1 and X_2 must be identical and are one of these values:
 "white"-white noise RO (specify n_T=n_h=1)
 "red"-red noise RO (specify n_T=n_h=0)
X_3 is one of these values:
 "additive"-additive noise RO (B is not considered)
 "multi"-multiplicative RO (specify n_g=0)
 "multi-H"-multiplicative-heaviside RO (specify n_g=1)

*Note: The RO_solver function supports different values for n_T and n_h. However, the RO_fitting function does not, as this option is not consistently available for all supported methods and is impractical in implementation.

• method_fitting-fitting method (default is specified by the table at the end of Section 3.1)

Option for specifying the fitting method for the RO. method_fitting is one of these values:

"LR-F"-linear regression (tendency with forward differencing scheme)

"LR-C"-linear regression (tendency with central differencing scheme)

• dt_fitting-temporal interval of T and h (scalar) (default=1.0)

Specify the temporal interval of T and h. The unit is a month. If dt_fitting is not specified, it uses the default value (1.0).

Output Arguments

• par-fitted parameter (dictionary; 16x1)

The fitted parameter output, specified by a dictionary size of 16,

par=[R, F1, F2, epsilon, b_T, c, d, b_h, sigma_T, sigma_h, B, m_T, m_h, n_T, n_h, n_g].

The output R, F1, b_T, c, d are:

NaN-if the parameter element corresponding to par_option_T is 0

scalar (X)-if the parameter element corresponding to par_option_T is 1

vector ([X, X_a, φ_a])-if the parameter element corresponding to par_option_T is 3

The output F2, epsilon, b_h are:

NaN-if the parameter element corresponding to par_option_h is 0

scalar (X)-if the parameter element corresponding to par_option_h is 1

vector ([X, X_a, φ_a])-if the parameter element corresponding to par_option_h is 3

The output sigma_T, and sigma_h are scalar.

The output B is:

NaN-if the noise option "T_type" is "additive"

scalar-if the noise option "T_type" is "multi" or "multi-H"

The output m_T and m_h are:

0-if the noise option "T" and "h" are "white"

scalar-if the noise option "T" and "h" are "red"

The output n_T, n_h, n_g are identical to the input n_T, n_h, n_g.

5.2 Solver

The RO_solver function numerically solves the RO equation using the given parameter inputs, and generates time series for T and h.

Syntax

[T,h]=RO_solver(par, IC, N, NE)
[T,h]=RO_solver(par, IC, N, NE, NM)
[T,h]=RO_solver(par, IC, N, NE, NM, dt)
[T,h]=RO_solver(par, IC, N, NE, NM, dt, saveat)
[T,h]=RO_solver(par, IC, N, NE, NM, dt, saveat, EF)
[T,h]=RO_solver(par, IC, N, NE, NM, dt, saveat, EF, noise_custom)
[__, noise]=RO_solver(__)

Description

• [T,h]=RO_solver(par, IC, N, NE) returns the numerically integrated T and h time series ensemble using the parameter par, initial condition IC, simulation length N, and ensemble number NE.

• [T,h]=RO_solver(par, IC, N, NE, NM) specifies the numerical integration method NM.

• [T,h]=RO_solver(par, IC, N, NE, NM, dt) specifies the time step of the numerical integration.

• [T,h]=RO_solver(par, IC, N, NE, NM, dt, saveat) specifies the time interval of the output.

• [T,h]=RO_solver(par, IC, N, NE, NM, dt, saveat, EF) specifies the external forcing (E_T and E_h).

• [T,h]=RO_solver(par, IC, N, NE, NM, dt, saveat, EF, noise_custom) specifies custom noise time series

• [__, noise]=RO_solver(__) returns noise time series used for the integration.

Input arguments

• par-Input RO parameter (dictionary; 16x1)
The input parameter, specified by dictionary array size of 16:

par=[R, F1, F2, epsilon, b_T, c, d, b_h, sigma_T, sigma_h, B, m_T, m_h, n_T, n_h, n_g].

The input parameter (X), R, F1, F2, epsilon, b_T, c, d, b_h, sigma_T, sigma_h, B, m_T, m_h are one of these values:

scalar (X)-seasonally-constant parameter (X=X)

vector ([X, X_a, φ_a])-seasonally-varying parameter with annual cycle (X= X+ $X_a$sin(wt+φ_a))

vector (Nx1)-time-varying parameter, specified by the user. The input parameter is a monthly time series length of N. If the numerical time step of the integration (dt) is smaller than 1.0, the linear interpolation is performed to fill the unspecified value.

vector (NTx1)-time-varying parameter, specified by the user. The input parameter is the time series length of NT where NT=floor((N-1)/dt)+1.

The input parameter n_T, n_h, n_g are scalar.

• IC-Input initial conditions for T and h (vector; 2x1 or 1x2).

Initial condition array for T and h, specified by [T_0, h_0].

• N-Input simulation length in month (scalar).

Numerical integration length in a month, specified by a scalar value.

• NE-Input ensemble number (scalar)

Number of ensembles with different noise realization, specified by scalar value.

• NM-Input numerical integration method (default="EH")

Numerical integration method for solving RO. The values are one of these.

"EM"-Euler-Maruyama method

"EH"-Euler-Huen method

• dt-Input numerical time step of the integration (scalar) (default=0.1)

The time step of numerical integration (unit: month)

• saveat-Input numerical simulation output (scalar) (default=1.0)

Sampling intervals to save the output solution (unit: month). saveat should be an integer multiple of dt.

• EF-Input external forcing (dictionary; 2x1) (default={'E_T':0.0, 'E_h':0.0})

Specifies external forcing terms, {'E_T':E_T, 'E_h': E_h}. The input forcing, E_T and E_h are one of these values:

scalar (X)-time-constant forcing (X=X)

vector ([X, X_a, φ_a])-seasonally-varying forcing with annual cycle (X= X+ $X_a$sin(wt+φ_a))

vector (Nx1)-time-varying forcing, specified by the user. Input forcing is a monthly time series length of N. If the numerical time step of the integration (dt) is smaller than 1.0, the linear interpolation is performed to fill the unspecified value.

vector (NTx1)-time-varying forcing, specified by the user. Input forcing is the time series length of NT where NT=floor((N-1)/dt)+1.

• noise_custom-Input custom white noise (list of length 4) (default=[])

Specify the white noise time series for the RO. If noise_custom is specified as an empty value (default), the solver uses internally generated noise. If noise_custom is specified as a list, the solver uses the specified noise time series. Each element of the list (length: 4) is a white noise time series for the dT/dt, dh/dt, dxi_T/dt, and dxi_h/dt terms. The length of each white noise time series is NT. For the white noise RO (n_T=n_h=1), the only first and second elements in the list are used, because the dxi_T/dt, and dxi_h/dt terms are not included in the RO. For the red noise RO (n_T=n_h=0), the only third and fourth elements in the list are used because there are no white noise terms in dT/dt and dh/dt.

Output arguments

• T-Output ensemble of T time series (matrix; NE x NT)

The output T time series ensemble. T is the NE x NT matrix where NE is a number of ensembles and NT is the number of the time step (NT=floor((N-1)/saveat)+1).

• h-Output ensemble of h time series (matrix; NE x NT)

The output h time series ensemble. h is NE x NT matrix where NE is a number of ensembles and NT is the number of the time step (NT=floor((N-1)/saveat)+1).

• noise-Output ensemble of noise time series (matrix; NE x 4 x NT)

The returned white noise time series, which are used in dT/dt, dh/dt, dxi_T/dt, dxi_h/dt terms. noise is NE x 4 x NT where NE is the number of the ensemble and NT is the number of the time step (NT=floor((N-1)/dt)+1).

6. Acknowledgment

The CRO code distribution team members are Soong-Ki Kim, Sen Zhao, Bastien Pagli, Samantha Stevenson, Priyamvada Priya, and Sooman Han. We thank Bastien Pagli for preparing the Python CRO package for the ENSO Winter School 2025.