Primary Data Description: On equilibrium fluctuations, von Storch, 2022

The data are produced by the following MATLAB scripts given below in blue. Most of them use the functions **lorz_n.m, lorz_f_n_shift.m** to get Lorenz solutions x and its differential forcing f. lorz_f_n_shift writes out not only x (as it is done in lorz_n) also the differential forcing f.

	Scripts	**Functions & data needed**
Figure 2	**plot_ts_xf** calculates and plots two different Lorenz equilibrium solutions Equilibrium solutions are obtained by - starting from a state (x1,y1,z1), which is an equilibrium state that is obtained by integrating the Lorenz model over a long time period (stored in **var.mat**) and then slightly perturbed by adding a normal random variable - integrate this slightly-perturbed state for another nts time steps to produce a spin-up integration - integrate the Lorenz model from the end of the spin-up integration to produce an equilibrium solution	**lorz_n** **lorz_f_n_shift** **var.mat** contains the initial conditions produced by **get_var**
Figure 3	**get_var** calculates variance a) by time averaging a single equilibrium solution, and b) by averaging over an ensemble of equilibrium solutions. The time averaged variance is a function of the length of the solution. The ensemble averaged variance is a function of ensemble size. Equilibrium solutions are obtained as described above. **plot_var** plots the result	**lorz_n** produces a solution of the Lorenz model using the Runge Kutta method and a time step dt=0.01. The solution starts from (x1,y1,z1) and has the length n
Figure 4	**get_c** calculates auto-correlation functions of x from a single equilibrium solution of three different lengths **plot_c** plots the time-averaged auto-correlation functions	**lorz_n** **var.mat**
Figure 5	**get_gamma** calculates auto-correlation functions of x from an ensemble of three different ensemble sizes **plot_gamma** plots the ensemble-averaged auto-correlation functions	**lorz_n** **var.mat**
Figure 6	**get_gamma_f** calculates auto-covariance functions of f from an ensemble of size en=1e+6 **get_gamma_xf** calculates cross-covariance functions between x and f from an ensemble of size en=1e+6 **plot_gamma_xf** plots the results	**lorz_f_n_shift** **var.mat**

	Scripts	Functions & data needed
Figure 7	**get_spxf_shift** calculates ensemble averaged spectra, including both the power spectra of x and those of f, and the co-spectra and the coherence and phase spectra between x and f, using one ensemble of size 1000 consisting equilibrium solutions of length 1e+7+1 **get_white_lf_var** a) identifies the frequency at which the spectrum deviates 5% from its low-frequency plateau and stores the result in **omega.mat** (needed in plot_spxf), b) calculates the variance associated with the low-frequency plateau **plot_spxf** plots power spectra of x and f and the coherence and phase spectra between x and f	**lorz_n** **lorz_f_n_shift** **var.mat** **omega.mat**
Figure 8	**plot_spf_linear** plots the spectra of f and the amplitude of the cross-spectrum between x and f (output of **get_spxf_shift**) in linear scale (rather than log scale)	
Figure9	**get_signal** calculates the spectra of two impulse-like signals. The non-zero values at the central piece of a signal is obtained from an AR1-process **plot_signal** plots the resulting spectra	**ar1**
Figure B1	**get_sp_m** calculate spectra derived from three ensembles of size 1, 10 100. The ensembles contain equilibrium solutions of size 1e+7+1. The spectra derived from an ensemble of size 1000 is calculated by get_spxf_shift. **plot_sp_m** plots spectra of the first component obtained from four ensembles of sizes 1, 10, 100, and 1000.	**lorz_n** **var.mat**
Figure B2	**get_sp_n** calculates spectra averaged over 4 ensembles of size=1000, each consisting equilibrium solutions of length 2N=1e+3, 1e+4, 1e+5, 1e+6. **plot_sp_n** plots the result	**lorz_n** **var.mat**
Figure C1	**get_sp_n_plus1** calculates the difference between two sample spectra, $S_{k,N+1}$-$S_{k,N}$, The two sample spectra are derived from two non-overlapping pieces of the same equilibrium solution, with length 2(N+1)+1 and 2N+1 **plot_sp_n_plus1** plots the result	**lorz_n** **var.mat**
Figure C2	**get_fc_1** calculates the the Fourier coefficient at the same frequency from 100 records of length T from a single equilibrium solution **plot_fc** plots the Fourier coefficients	**lorz_n** **var.mat**