A USER GUIDE FOR `QPOT4LORENZ63` PACKAGE

MARIA CAMERON AND SHUO YANG

Contents

The package contains C and Matlab source codes for visualization and analysis of stochastic Lorenz'63 model:

$$d\mathbf{x} = \begin{bmatrix} \sigma(x_2 - x_1) \\ x_1(\rho - x_3) - x_2 \\ x_1x_2 - \beta x_3 \end{bmatrix} dt + \sqrt{\epsilon} d\mathbf{w}, \quad \text{where} \quad \mathbf{x} \equiv \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}, \tag{1}$$

$d\mathbf{w}$ is the three-dimensional standard Brownian motion, and ϵ is a small parameter. We set $\sigma = 10$ and $\beta = 8/3$, and let ρ range from zero to the critical value $\rho_2 \approx 24.74$ at which the strangle attractor becomes the only attractor of the corresponding deterministic ODE. Below are the list of C and Matlab programs with short instructions included in the package.

1. C source codes

- `olim3D4Lorenz63.c`, a C source code implementing the 3D ordered line integral method with the midpoint quadrature rule [5]. the vector field is the Lorenz vector field. No input is required. Output files:
  - `LorenzQpot_rho<...>.txt` contains a column of values of the quasipotential at 3D mesh points. Its name is assigned in line 1440.

  - `parameters_rho<...>.txt` contains a column vector with the following parameters:
    `XMIN, XMAX, YMIN, YMAX, ZMIN, ZMAX, NX, NY, NZ,`
    `sigma, beta, rho, x_ipoint.x, x_ipoint.y, x_ipoint.z, K, SSF,`
    where `struct myvector x_ipoint` is the asymptotically stable equilibrium with respect to which the quasipotential is computed, and SSF is the subsampling parameter: if `NX = NY = NX = SSF·n + 1` then the number of output values in `LorenzQpot_rho<...>.txt` is $(n+1)^3$. The name of this file is assigned in line 1478.

  To run `olim3D4Lorenz63.c`, type in the terminal window:

```
gcc -Wall LinLorenz.c olim3D4Lorenz63.c -lm -O3
```

  The level sets of the computed quasipotential are visualized by the code `olim3Dvisualize.m`.

- `LinLorenz.c` finds the quasipotential decomposition for linearized Lorenz'63 near the origin if $0 < \rho < 1$ or near $C_+ = (\sqrt{\beta(\rho-1)}, \sqrt{\beta(\rho-1)}, \rho-1)$ if $1 < \rho < \rho_2 \approx 24.74$. It also contains a number of linear algebra functions. It is an auxiliary file that does not contain function `main`. It implements an algorithm similar to Bartels-Stewart [1] but customized and simplified for finding linearized Lorenz'63. A description of the algorithm is found in [2].

- `findQmatrix.h` is a header file for `LinLorenz.c`.

- `olim2DEquilibLimitCycle.c`, a C source code implementing the 2D ordered line integral method with the midpoint quadrature rule on a radial mesh [3]. It requires the following input files:
  - `Mesh2Ddata_rho<...>.txt` containing two numbers, N_r and N_a, where $N_r \times N_a$ is the size of the radial mesh. Its name is assigned in line 752.
  - `Mesh2Drho<...>.txt` contains x_1, x_2, and x_3 coordinates of the mesh points. Its name is assigned in line 770.

  These input files are generated by the Matlab code `make2Dmesh.m`.
  `olim2DEquilibLimitCycle.c` produces one output file `Qpot2Drho<...>.txt` with the values of the quasipotential at the mesh points. Its name is assigned in line 816.
  To run `olim2DEquilibLimitCycle.c`, type in the terminal window:

```
gcc olim2DEquilibLimitCycle.c -lm -O3
```

  The quasipotential computed by `olim2DEquilibLimitCycle.c` is visualized by `MyPolarPlot.m`.

- `ShootMAPs.c` implements 4-stage 4-th order Runge-Kutta method for shooting MAPs. Input files for it are produced by `olim3D4Lorenz63.c`:
  - `parameters_rho<...>.txt`. The name is assigned in line 561.

– LorenzQpot_rho<...>.txt. The name is assigned in line 580.

The output file MAP_rho<...>.txt is $N_{MAP} \times 3$ array of points along the computed MAP. This program is set up for $1 < \rho < 13$ or $\rho = 15$, or $\rho = 20$. If you need to shoot MAPs for other values of ρ, provide terminal points for the MAPs as it is done in lines 611–625. To run ShootMAPs.c, type in the terminal window:

```
gcc -Wall LinLorenz.c ShootMAPs.c -lm -O3
```

2. Matlab source codes

- make2Dmesh.m generates a radial mesh on the manifold defined by the characteristics going from an unstable limit cycle to an asymptotically stable spiral point [3] (Appendix F). Important parameters are all set in lines 5 – 26.

- thickness.m implements a procedure for measuring the thickness of the Lorenz attractor as described in [3] (Apprendix G). Call it as:
  `w = thickness(y0),`
  where y0 must be a point lying on the Lorenz attractor, and w will be the estimate for the thickness at near y0.

- thickness_map.m generates a collection of points lying on the Lorenz attractor, calls thickness.m to estimate thickness near these points, and uses colorcoding to visualize the thickness map. It is convenient to call StrangeAttractorMesh.m first to depict the Lorenz attractor, and then plot the thickness map in the same figure.

- StrangeAttractorMesh.m generates a mesh on a union of four manifolds approximating the Lorenz attractor. Call it as
  `handle = StrangeAttractorMesh();`
  if you would like to output a handle for the plotted objects.

- olim3Dvisualize.m visualizes level sets of the quasipotential in 3D. It requires two input files, parameters_rho<...>.txt and LorenzQpot_rho<...>.txt produced by olim3Dlorenz63.c. If $0 < \rho < 1$, it calls gmam_lorenz.m to find a collection of MAPs. gmam_lorenz.m implements the geometric minimum action method (GMAM) [4]. If $\rho_0 \approx 13.926 < \rho < \rho_2 \approx 24.74$, it calls find_saddle_cycle.m to find the saddle cycles. olim3Dvisualize.m outputs handle, a handle for plotted objects for the case if you would like to make a movie using make_movie.m.

- find_saddle_cycle.m finds saddle cycles. Call it as
  `[Y,lY,l] = find_saddle_cycle(rho).`
  Y is a $N_{cycle} \times 3$ arrays of points of the saddle cycle, lY is the arclength along it, while l is the total length.

- gmam_lorenz.m implements GMAM [4] for finding minimum action paths (MAPs). It is handy when the paths do not spiral like for $\rho < 1$. Call it as
  MAP = gmam_lorenz(xi,xf,sigma,beta,rho).
  xi and xf are 3×1 arrays of coordinates of the initial and final points of the desired MAP. For $\rho > 1$, we recommend to use the C code ShootMAPs.c.

- make_movie.m makes a movie in which the plotted objects given by the input argument handle are rotated around the z-axis. Call it as
  make_movie(handle,rho).
  It outputs an avi file.

- MyPolarPlot.m visualizes the quasipotential computed on a radial mesh. It requires two input files: Qpot2Drho<...>.txt produced by olim2DEquilibLimitCycle.c, and Mesh2Drho<...>.txt produced by make2Dmesh.m.

- lorenz_diagram.m plots a bifurcation diagram for the deterministic Lorenz system in the (ρ, x_1)-plane (see [3]).

References

[1] R. Bartels and G. W. Stewart, *Solution of the matrix equation* $AX + XB = C$, Comm A.C.M., 15 (1972), 9, 820–826

[2] M. Cameron, Construction of the quasi-potential for linear SDEs using false quasi-potentials and a geometric recursion, arXiv:1801.00327

[3] M. Cameron and S. Yang, Computing the quasipotential for highly dissipative and chaotic SDEs. An application to stochastic Lorenz'63, *in revision*, arXiv:

[4] M. Heymann, E. Vanden-Eijnden, Pathways of maximum likelihood for rare events in non-equilibrium systems, application to nucleation in the presence of shear, Phys. Rev. Lett. **100**, 14, 140601 (2007)

[5] S. Yang, S. Potter, and M. Cameron, Computing the quasipotential for non-gradient SDEs in 3D, *submitted to* Journal of Computational Physics, 2018, arXiv:1808.00562