THE C CODE `OLIM4VAD.C`: A USER GUIDE

DAISY DAHIYA AND MARIA CAMERON

Contents

The source file `OLIM4VAD.c` implements the Ordered Line Integral Method with the midpoint quadrature rule for computing the quasi-potential for 2D SDEs with position-dependent and anisotropic diffusion,

$$d\mathbf{x} = \mathbf{b}(\mathbf{x})dt + \sigma(\mathbf{x})\sqrt{\epsilon}dW, \quad \mathbf{x} \in \Omega \subset \mathbb{R}^2, \tag{1}$$

where $\mathbf{b}(\mathbf{x})$ is a twice continuously differentiable vector field, $\sigma(\mathbf{x})$ is a nonsingular continuous matrix function (the diffusion matrix), ϵ is a small parameter, and dW is the standard 2D Brownian motion. A description of the method and the results of numerical tests, as well as applications to the Maier-Stein model [2, 3] and to a Lambda Phage genetic toggle model [4], are found in [1]. Our numerical tests have shown that `OLIM4VAD.c` is highly accurate when the ratio $\lambda(\mathbf{x})_{\max}/\lambda(\mathbf{x})_{\min}$ of the eigenvalues of $A(\mathbf{x}) = [\sigma(\mathbf{x})\sigma(\mathbf{x})^{-1}]$ does not exceed 10.

1. Compiling and running

Any other C compiler should be applicable for running `OLIM4VAD.c`. At the command line, type

```
\$ clang OLIM4VAD.c -lm -O3 -o OLIM4VAD
\$ ./OLIM4VAD
```

If the program runs normally with the default settings, you should see something like:

```
in param()
in olim()
The boundary is reached, 590724 points are Accepted, Umax = 4.1478e+00
cputime of olim() = 14.7206
NX = 1024, NY = 1024, K = 22
errmax = 5.5491e-03, erms = 1.9039e-03
Npath = 824
```

Here, `Umax` is the value of the quasi-potential at the last point that has become Accepted (i.e., the value of the quasi-potential has been finalized at it). Since the quasi-potential is computed approximately in the order of increase of its values, it is approximately the maximal value of the quasi-potential at the Accepted points. `NX` and `NY` indicate the size of the computational domain; `K` is the update factor; `errmax` is the maximal absolute error; `erms` is the root mean square error; `Npath` is the number of points at the Minimum Action Path (MAP) from the asymptotically stable equilibrium to the user-specified point.

2. The default settings

The provided version of `OLIM4VAD.c` computes the quasi-potential for the SDE

$$\begin{bmatrix} dx_1 \\ dx_2 \end{bmatrix} = \begin{bmatrix} x_2 g(r,\phi) + x_1 f(r,\phi) \\ -x_1 g(r,\phi) + x_2 f(r,\phi) \end{bmatrix} dt + \sqrt{\epsilon} \begin{bmatrix} r^{-1}x_1 & -x_2 \\ r^{-1}x_2 & x_1 \end{bmatrix} \begin{bmatrix} dw_1 \\ dw_2 \end{bmatrix}, \tag{2}$$

where $r = \sqrt{x_1^2 + x_2^2}$, ϕ is the polar angle, and the functions $f(r,\phi)$ and $g(r,\phi)$ are given by

$$f(r,\phi) = 1 - \frac{r^2}{9} + \frac{\sin\phi}{r^2}, \quad g(r,\phi) = \frac{\sin\phi}{r^2} - \left(1 - \frac{r^2}{9}\right).$$

The point $(x_1 = 3, x_2 = 0) \equiv (r = 3, \phi = 0)$ is an asymptotically stable equilibrium. The exact quasi-potential with respect to it given by

$$U(\mathbf{x}) = r^2\left(\frac{r^2}{18} - 1\right) + \frac{9}{2} + 2\,(1 - \cos\phi) = \|\mathbf{x}\|^2\left(\frac{\|\mathbf{x}\|^2}{18} - 1\right) + \frac{9}{2} + 2\left(1 - \frac{x_1}{\|\mathbf{x}\|}\right). \tag{3}$$

is well-defined in any domain of the form $\mathbb{R}^2 \backslash \{\mathbf{x} \mid \|\mathbf{x}\| < r_0\}$ where $0 < r_0 < 3$. The rotational component is given by:

$$\mathbf{l}(\mathbf{x}) = \begin{bmatrix} x_2\left(\frac{\|\mathbf{x}\|}{9} - 1\right) + \frac{x_1 x_2}{\|\mathbf{x}\|^{3/2}} \\ -x_1\left(\frac{\|\mathbf{x}\|}{9} - 1\right) + \frac{x_2^2}{\|\mathbf{x}\|^{3/2}} \end{bmatrix}. \tag{4}$$

The magnitudes of the rotational and the potential components are equal everywhere. The eigenvalues of the matrix $A(\mathbf{x}) = [\sigma(\mathbf{x})\sigma(\mathbf{x})^\top]^{-1}$ are 1 and $\|\mathbf{x}\|^{-2}$. The computational domain

$$\{-3.8 \le x_1 \le 4.2,\ -4.0 \le x_2 \le 4.0\},$$

is chosen so that the computation reaches the saddle point at $(x_1 = -3, x_2 = 0)$ prior to reaching the boundary while does not reach the origin, where the quasi-potential is not defined.

3. Output and visualization

`OLIM4VAD.c` does not require any input files. There are two output files:

- `Qpot.txt` contains the `NX`×`NY` array of computed quasi-potential values. The quasi-potential values at all non-Accepted mesh points get values $1e{+}6$ as they are printed to the file `Qpot.txt`.
- `Instanton.txt` contains the `Npath`×`2` array of points of the MAP (Minimum Action Path, a.k.a. the instanton) from the saddle at $(-3, 0)$ to the asymptotically stable equilibrium at $(3, 0)$ (see Section 1).

The names of the output files are specified in lines 154–155 of `OLIM4VAD.c`:

```
const char *f_qpot_name = "Qpot.txt"; // output file with the quasipotential
const char *ifname = "Instanton.txt"; // output file with the MAP (the instanton)
```

The quasi-potential and the MAP can be visualized with Matlab. The file `OLIM4VADvisualize.m` plots the level sets of the computed quasi-potential, the MAP, and the direction of the vector field.

4. Changing the SDE, the parameters, and the computational domain

- The mesh size NX×NY and the update parameter K are specified in lines 24-26:

```
#define NX 1024
#define NY 1024
#define K 22
```

We recommend picking the value of the update parameter K according to the following Rule-of-Thumb proposed in [5], provided that the ratio $\lambda(\mathbf{x})_{\max}/\lambda(\mathbf{x})_{\min}$ of the eigenvalues of the matrix $A(\mathbf{x}) = [\sigma(\mathbf{x})\sigma(\mathbf{x})^{-1}]$ does not exceed 10

$$\boxed{K(N) = 10 + 4(\mathsf{round}[\log_2 N] - 7), \quad 2^7 \le N \le 2^{12}.} \tag{5}$$

- The vector field $\mathbf{b}(\mathbf{x})$ needs to be specified in the function `struct myvector myfield(struct myvector x)` (lines 159–172).
- The diffusion matrix $\sigma(\mathbf{x})$ needs to be specified in the function `struct mymatrix Sigma_matrix(struct myvector x)` (lines 176–185).
- The exact solution, if available, needs to be specified in the function `double exact_solution(struct myvector x)` (lines 201–208). If it is unavailable, you need to comment out all lines where it is used: lines 141, 201–208, 233, 1061–1067, and 1075.
- The asymptotically stable equilibrium with respect to which the quasi-potential needs to be computed is specified by the variable `x_ipoint` of type `struct myvector` in the function `param()` in line 218.
- The computational domain is specified by the variables `XMIN, XMAX, YMIN, YMAX` of type `double` in the function `param()` in lines 219–220.
- For initialization, you need to provide the matrix `Bi` (type `struct mymatrix`) (line 221). `Bi` is the Jacobian matrix of the vector field $\mathbf{b}(\mathbf{x})$ evaluated at the asymptotically stable equilibrium given by `x_ipoint`.
- If you would like to compute a MAP, specify the point where it should end in the variable `x_ShootMAP` of type `struct myvector` in the function `param()` (line 222). The function `void ShootMAP()` starting on line 919 computes the MAP by shooting it backwards from the point `x_ShootMAP` using the standard 4th order 4-stage Runge-Kutta method and linear interpolation. The integration stops either when the asymptotically stable equilibrium is reached or when the number of the computed MAP points exceeds the maximum specified in the variable `Npathmax` in line 131. If you do not want to compute a MAP, comment out lines 1078–1083.

Acknowledgement

We thank Mr. Samuel Potter (UMD, CS) for valuable advice.

References

[1] D. Dahiya and M. Cameron, An Ordered Line Integral Method for Computing the Quasi-potential in the case of Variable Anisotropic Diffusion, *submitted to Physica D*, 2018, arXiv:

[2] R.S. Maier, and D.L. Stein. Escape problem for irreversible systems. Phys. Rev. E 48(2), 931 (1993)

[3] R.S. Maier, and D.L. Stein. A Scaling Theory of Bifurcations in the Symmetric Weak-Noise Escape Problem, J. Stat. Phys. **83** 3/4 291-357 (1996)

[4] E. Aurell and K. Sneppen. Epigenetics as a First Exit Problem. Physical Review Letters. 88: 048101(1-4), (2002)

[5] D. Dahiya, and M. Cameron, Ordered Line Integral Methods for Computing the Quasi-potential. *J. Sci. Comput.,* Nov. 2017.