

# String method and time-independent free energy estimator

C. J. F. Solano

March 19, 2017

The string method (SM) [1,2] has been implemented for finding minimum free energy paths (MFEPs) on a given free energy surface (FES). A MFEP is defined as a curve whose tangent is everywhere parallel to the gradient of the free energy calculated in some appropriate metric. Since metadynamics [3] allows for estimating the FES and deriving its gradient, MFEPs can be computed efficiently using the SM. It should be noted, however, that other enhanced sampling methods could be also combined with the SM by including minor modifications in this code. Given an initial guess for a curve on the FES, the SM finds the closest MFEP by moving a discrete set of points on the curve by the steepest descent on the free energy landscape. At the same time, the points are kept at constant distance from each other.

A time-independent and locally convergent FES estimator [4] for metadynamics has been also implemented in this code. Thereby, this FES estimator can be calculated at any position in the collective variable (CV) space.

A run is driven through a set of commands which are usually specified in an input file. This file is composed by several lines of characters which are internally divided into keywords and variables. The input file is not subjected to any format, except that punctuation marks are not allowable so that different words should be separated at least by one blank space. A command can be split in several lines. To this end, a `–` symbol should be included at the end of a line to indicate that next one belongs to the same command.

Run program as: executable file [*input* [*output*]] [-h]

## 1 General Commands

Some conventions are used in the description on the different commands. On the one hand, keywords are written in bold characters. On the other hand, data provided by the user are written in italic characters. Optional keywords and data are encapsulated between square brackets. Those keywords and data to be selected among several options are separated by vertical lines and encapsulated between parenthesis. In some cases, a set of keywords and data can be abbreviated by the symbol `(...)`.

- **TITLE** *name* *runtitle*: Input title.
- **OPEN** *unit* *name* *filename* (**read|write**): Open a file named as *filename* linked to an integer *unit* unit to read (i.e., input file) or write (i.e., output file) some specific information.
- **CLOSE** *unit*: Close a file linked to an integer *unit* unit.
- **STRM** *ncv* *N<sub>CV</sub>* [**metric** (...)] [**string** (...)] [**stmfe** (...)]: Invoked the SM module for a *N<sub>CV</sub>*-dimensional CV space. The keywords and data are divided into three different

categories: (i) metric tensor definition, (ii) MFEP estimation, and (iii) free energy calculation along a given MFEP.

- **LOCAL ncv**  $N_{CV}$  (...): Invoked the time-independent and local FES estimator module for a  $N_{CV}$ -dimensional CV space.
- **FES ncv**  $N_{CV}$  (...): Print a new FES file evaluated in a mesh grid with a different number of grid points.
- **EXIT**: End of a run.

## 2 STRM Command

### 2.1 Metric tensor definition

A metric tensor as defined in [2] is specified in the SM module by:

**metric munit**  $imunit$ ,

where  $imunit$  is an unit linked to an input file that contains the elements of this tensor (see Section 5).

### 2.2 MFEP estimation

A MFEP calculation is specified in the SM module by:

**string** (...),

where (...) includes

- **mig**  $N$ : Number  $N > 4$  of images (or points) used to discretize the string (or curve).
- (**rimag ufts**  $imunit$  | **gimag iufts**  $iendmunit$ ): Read initial images from file linked to an  $imunit$  unit (see Section 5). Alternatively, read initial end images from file linked to an  $iendmunit$  unit (see Section 5) and generate the rest of initial images on a straight line connecting both ends.
- (**gauss** (...) | **Bspline** (...) | **bicubic** (...)): Specify how the FES and its gradient are calculated. Further details about this set of keywords and data are given in Subsection 2.4.
- **dt**  $\Delta t$ : Input time step  $\Delta t > 0$  for evolving the position of the images (0.0001 by default).
- **kappa**  $\kappa$ : Input parameter  $\kappa \in [0, 1)$  to smooth the string evolution (0.1 by default).
- **ftsit**  $M_{it}$ : Input maximum number of iterations  $M_{it} > 0$  (10000 by default).
- [**evolution ustring**  $iustring$ ]: The evolution of the string is monitored by computing an specific distance magnitude  $d(t)$  (see [2]) that is stored in an output file linked to an  $iustring$  unit. A plateau in the profile of  $d(t)$  is an indication that the MFEP calculation has converged.
- [**conv ftstol**  $\alpha$ ]: Check the convergence of a MFEP calculation by using a tolerance value  $\alpha > 0$  (default =  $\max(N^{-4}, 10^{-10})$ ) for the maximum distance of images between consecutive iterations divided by the time step and stop automatically the iteration process when convergence is achieved.

- **[pimag [ufts imunit]]**: Print final images in an output file linked to an *imunit* unit. This unit should be specified if **rimag ufts imunit** is not defined previously. Otherwise, it overwrites a file that includes the initial images.

It should be noted that a metric tensor (see Subsection 2.1) should be defined before a MFEP calculation. The evolution of the string is the preferred option to evaluate the convergence of a MFEP calculation.

### 2.3 Free energy calculation along a MFEP

A free energy calculation along a given MFEP is specified in the SM module by:

**stmfe** (...),  
where (...) includes

- **[mig N ufts imunit (gauss (...) | Bspline (...) | bicubic (...))]**: Some keywords and data defined in Subsections 2.2 and 2.4 should be specified if a MFEP calculation is not previously performed.
- **ufe iufe**: Print free energy along a given MFEP in an output file linked to an *iufe* unit.

### 2.4 FES (and its gradient) calculation

There are three different options for calculating a FES (and its gradient): (i) from a biased potential (i.e., Gaussians hills) file ( $N_{CV} \geq 2$ ), (ii) from a B-spline interpolation ( $N_{CV} = 2, 3$ ), and (iii) from a bicubic interpolation ( $N_{CV} = 2$ ).

#### 2.4.1 Gaussian hills

A FES (and its gradient) can be calculated from a Gaussian hills file (see Section 5) using the keywords and data:

**gauss ugauss iugauss ngauss N<sub>g</sub> nskip N<sub>0</sub> ctgauss C [nigauss N<sub>av</sub>]**,

where

- *iugauss* is an unit linked to an input file that stores the Gaussian hills from a metadynamics simulation (see Section 5).
- $N_g > 0$  is the number of Gaussians.
- $N_0 \geq 0$  is the number of lines to be skipped in the hills file (0 by default).
- $C$  is a constant factor to pass from bias potential to FES (1.0 by default).
- $0 \leq N_{av} \leq N_g$  is a number of initial Gaussians for an average FES (0 by default).

#### 2.4.2 FES in a mesh grid

If a FES is evaluated and stored in a mesh grid covering the region of interest (see Section 5), then B-spline and bicubic interpolation can be used to calculate the FES (and its gradient) in any position in this CV domain. Therefore, **Bspline** (...) and **bicubic** (...) share keywords and data, where (...) includes

- **ugrid iugrid**: Input a file linked to an *iugrid* unit that stores FES in a mesh grid (see Section 5).

- **ng1**  $N_g(1)$ : Number of grid points  $N_g(1) > 0$  in first CV direction.
- **ng2**  $N_g(2)$ : Number of grid points  $N_g(2) > 0$  in second CV direction.
- [**ng3**  $N_g(3)$ ]: Number of grid points  $N_g(3) > 0$  in third CV direction.

### 3 LOCAL Command

The time-independent FES estimator is evaluated in a position in the CV space using the keywords and data:

**LOCAL ncv**  $N_{CV}$  **ugauss** *iugauss* **ngauss**  $N_g$  **nskip**  $N_0$  **ctgauss**  $C$  (...),

where  $0 < N_{CV} \leq 3$ , Gaussian hills file and related keywords and data are specified as in the Subsection 2.4, and (...) includes

- **temp**  $T$ : Input temperature ( $T = 300$  K by default).
- **min1**  $CV_m(1)$  [**min2**  $CV_m(2)$  [**min3**  $CV_m(3)$ ]]: Input minimum rectangular box values ( $CV_m(1)$  ( $CV_m(2)$  ( $CV_m(3)$ ))) in the CV space.
- **max1**  $CV_{mx}(1)$  [**max2**  $CV_{mx}(2)$  [**max3**  $CV_{mx}(3)$ ]]: Input maximum rectangular box values ( $CV_{mx}(1)$  ( $CV_{mx}(2)$  ( $CV_{mx}(3)$ ))) in the CV space.
- **grid1**  $N_g(1)$  [**grid2**  $N_g(2)$  [**grid3**  $N_g(3)$ ]]: Input number of grid points ( $N_g(1)$  ( $N_g(2)$  ( $N_g(3)$ ))) in the CV space.
- **CV1**  $CV(1)$  [**CV2**  $CV(2)$  [**CV3**  $CV(3)$ ]]: Input CV value ( $CV(1)$  ( $CV(2)$  ( $CV(3)$ ))) at which the FES estimator is evaluated.
- **ufes** *iufes*: Print the FES estimator in an output file linked to an *iufe* unit for a consecutive addition of bias along the simulation time.

### 4 FES command

A new FES file evaluated in a mesh grid with a different number of grid points is printed using the keywords and data:

**FES ncv**  $N_{CV}$  (**Bspline** (...) | **bicubic**) **ugrid** *iugrid* **ng1**  $N_g(1)$  **ng2**  $N_g(2)$  [**ng3**  $N_g(3)$ ] (...),

where  $N_{CV} = 2, 3$ , file for a FES evaluated in a mesh grid and related keywords and data are specified in the Subsection 2.4, and (...) includes

- **ng1w**  $N_w(1)$ : Number of new grid points  $N_w(1) > 0$  in first CV direction.
- **ng2w**  $N_w(2)$ : Number of new grid points  $N_w(2) > 0$  in second CV direction.
- [**ng3w**  $N_w(3)$ ]: Number of new grid points  $N_w(3) > 0$  in third CV direction.
- **ugridw** *iugridw*: Print FES evaluated in a new mesh grid in an output file linked to an *iugridw* unit.

### 5 Input files

In this section, one describes the format required for the input files. It should be noted that a specific format is described between parenthesis, where  $b$  indicates a boolean data type. Free format is indicated by an asterisk \*.

Figure 1: Metric tensor file for a CV space with  $N_{CV} = 2$ .

```
0.0054712938933149 0.0
0.0 0.1546498893945394
```

## 5.1 Metric tensor file

The elements of the metric tensor  $\mathbf{M}$  are specified as:

```
for  $i_1 = 1, N_{CV}$ 
  Record (*):  $M(i_1, i_2), i_2 = 1, N_{CV}$ 
end-for
```

An example is shown in Fig. 1.

## 5.2 Images files

### 5.2.1 Images file

The images of the string are specified as:

(A) *Angle CVs?*

- Record A1 (\*): *Comment*  
Comment line.
- Record A2 (b): *Qangcv*  
Boolean variable. *Qangcv = .true.* means that one or more CVs are angles obtained from a dot product between two vectors. It should be noted that metric tensor is constant only for linear CVs.
- Record A2' (\*): *Comment*  
Comment line.
- Record A2'' (\*) (only if *Qangcv = .true.*):  $d_{CV}(i), i = 1, N_{CV}$   
Constant module for angle CVs ( $d_{CV} \leq 0$  if a CV is not an angle).

(B) *Images*

- Record B1 (\*): *Comment*  
Comment line.
- Record B2 (\*):

```
for  $j = 1, N$ 
  Record (*):  $z(i, j), i = 1, N_{CV}$ 
end-for
```

Images  $\mathbf{z}_\alpha, \alpha = 1, \dots, N$

An example is shown in Fig. 2.

Figure 2: Images ( $N = 10$ ) for a CV space with  $N_{CV} = 2$ .

```
! Angle CVs?:
F
! Images (ncv dimension):
-2.50240181215030      0.675771285835378
-2.49768559934760      0.675096491225328
-2.49296540700250      0.674450120403983
-2.48824191200206      0.673828343860590
-2.48351596663152      0.673225474446073
-2.47878863421679      0.672633579634782
-2.47406125097491      0.672042090979229
-2.46933549459258      0.671437742120354
-2.46461290784417      0.670809104604351
-2.45989460665685      0.670149070876339
```

Figure 3: End images for a CV space with  $N_{CV} = 2$ .

```
.false.
-2.5 0.685
-2.23 0.7
```

### 5.2.2 End images file

The end images of the string are specified as:

(A) *Angle CVs?*

- Record A1 (b): *Qangcv*

Boolean variable. *Qangcv = .true.* means that one or more CVs are angles obtained from a dot product between two vectors. It should be noted that metric tensor is constant only for linear CVs.

- Record A1' (\*) (only if *Qangcv = .true.*):  $d_{CV}(i), i = 1, N_{CV}$

Constant module for angle CVs ( $d_{CV} \leq 0$  if a CV is not an angle).

(B) *End images*

- Record B1 (\*):  $z(i, 1), i = 1, N_{CV}$
- Record B2 (\*):  $z(i, N), i = 1, N_{CV}$

An example is shown in Fig. 3.

### 5.3 Gaussian hills file

The format for Gaussian hills file from a metadynamics simulation is given bellow. It should be noted that any format is accepted for the first  $N_0$  lines since they are skipped.

for  $i_g = 1, N_g$

Record (\*):  $(z_g(i, i_g), i = 1, N_{CV}), (\delta z_g(i), i = 1, N_{CV}), w(i_g)$   
end-for

Figure 4: Gaussian hills file ( $N_0 = 1, N_g = 6$ ) for a CV space with  $N_{CV} = 2$ .

```

#!d.z theta sigma_d.z sigma_theta height
-2.905229887267241 0.6913208115403597 0.025 0.0872664626 5.172413793103448
-2.938893279983179 1.070637669780867 0.025 0.0872664626 5.172413793103448
-2.950763445524378 0.8130587248871374 0.025 0.0872664626 5.172413793103448
-2.970660829599484 0.7576791987739372 0.025 0.0872664626 5.172413793103448
-2.994136820789925 0.6644743681632246 0.025 0.0872664626 5.172413793103448
-3.070521335575753 0.7100414217403358 0.025 0.0872664626 5.172413793103448

```

where  $\mathbf{z}_g$  are the Gaussian positions in the CV space,  $\delta\mathbf{z}_g$  are the Gaussian widths (same for all different Gaussians), and  $\mathbf{w}$  are the Gaussian heights.

An example is shown in Fig. 4.

## 5.4 FES stored in a mesh grid file

The FES can be stored in a mesh grid covering the region of interest in the CV space. Then, the FES (and its gradient) can be evaluated in any specific position in this region by using B-spline ( $N_{CV} = 2, 3$ ) and bicubic interpolation ( $N_{CV} = 2$ ). Below, mesh grid file format is specified for  $N_{CV} = 2, 3$ . This file is not subjected to any format. Black lines are omitted when reading the file. Comments placed after the symbols `!`, `#` or `*` are removed.

For  $N_{CV} = 2$ :

```

for  $l_2 = 1, N_g(2)$ 
  for  $l_1 = 1, N_g(1)$ 
    Record (*,*,*)  $x_g(l_1), x_g(l_2), A(l_1, l_2)$ 
  end-for
end-for

```

For  $N_{CV} = 3$ :

```

for  $l_1 = 1, N_g(3)$ 
  for  $l_2 = 1, N_g(2)$ 
    for  $l_3 = 1, N_g(1)$ 
      Record (*,*,*,*)  $x_g(l_1), x_g(l_2), x_g(l_3), A(l_1, l_2, l_3)$ 
    end-for
  end-for
end-for

```

In previous description,  $\mathbf{x}_g$  is a grid point and  $A(\mathbf{x}_g)$  is the FES value at a given grid point.

## References

- [1] L. Maragliano *et al*, *J. Chem. Phys.* **125**, 024106 (2006).
- [2] L. Maragliano *et al*, *Chem. Phys. Lett.* **446**, 182-190 (2007).
- [3] A. Laio and M. Parrinello, *Proc. Natl. Acad. Sci. USA* **99**, 12562 (2002).
- [4] P. Tiwary and M. Parrinello, *J. Phys. Chem. B* **119**, 736-742 (2014).