

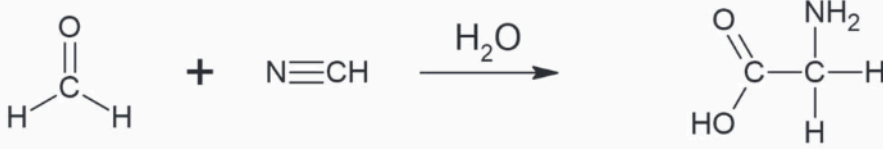
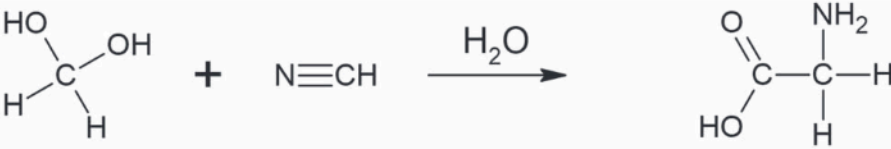
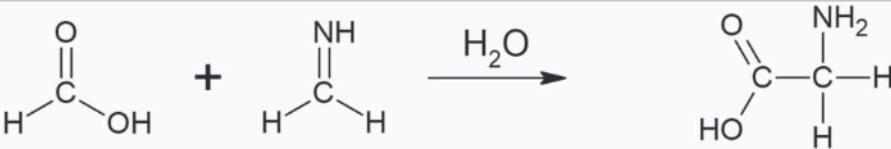
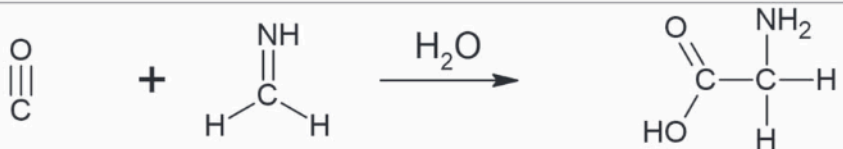
# Practicing Meta-dynamics on the Müller-Brown Potential Energy Surface

## 1. Introduction to Meta-dynamics

Based on my interest in the **molecular dynamics**(MD)<sup>[1]</sup>, my advisor recommended that I learn the **PLUMED**<sup>[2]</sup> plugin to deepen my understanding of MD capabilities and become familiar with state-of-the-art enhanced sampling techniques.

After hearing this advice, I did some research and discovered the PLUMED **Masterclasses**<sup>[3]</sup> which gather experts to explain cutting-edge methods and how to solve challenging problems. In this post, I will explain some parts of Masterclass 9. In some blog post in the future I will provide some challenges with the production of the state-of-art pipeline for the drug testing of the preclinical phase. That will only be our concern. In the multiple posts I will try to provide the path to getting the understanding of the highly accurate **meta dynamics-based calculation**<sup>[4]</sup> of protein-protein and protein-ligand binding potencies.

As a motivation, I was inspired by recent research on glycine formation. In that work, glycine was hypothesized to form stepwise from smaller molecules and was investigated using exploratory **CPMD**<sup>[5]</sup> and **free energy simulations**<sup>[6]</sup>. [Ab Initio Study of Glycine Formation in the Condensed Phase: Carbon Monoxide, Formaldimine, and Water Are Enough](#) provides the basis for much of this blog post. This study served as my entry point into the field.

System	Reaction		
S0			
S1			
S2			
S3			

**Fig 1**

Stepwise reactions observed in the glycine formation study (Francisco Carrascoza *et al* 2023). This diagram from the article shows a sequence of reactions (system S3V0) starting from carbon monoxide and formaldimine in water and ultimately forming glycine. (We will not delve into these specific reactions here; Figure 1 is meant to illustrate the kind of complex pathways meta-dynamics can discover.)

When I came across the lecture on **reaction pathways**<sup>[7]</sup>, I was able to put this idea into practice, but first I need to work through the simpler examples.

## 2. Enhanced-Sampling Tasks

### 2.1 Key Concepts

**Path-meta-dynamics (PMD)**<sup>[8]</sup> is an enhanced sampling method that simultaneously converges a transition pathway and computes the free-energy profile along that path. It achieves this by performing meta-dynamics on an adaptive reaction path connecting two stable states – thereby yielding either an average transition path or the **minimum free-energy path** (MFEP)<sup>[9]</sup>, depending on parameters.

To simplify computation and visualization, we will employ a basic model system. We'll use PLUMED's PESMD module standalone without coupling it to any **MD engine**<sup>[10]</sup> or full

molecular system so GROMACS won't be required for this task. These enhanced-sampling techniques can be applied to existing studies with minimal modifications, which bodes well for investigations of glycine formation.

We don't want to wait a lot of time for those things to do some interesting things so we just use few key descriptive degrees of freedom of the transition, commonly called **collective variables**.

The choice of CV's is a task of great importance. In the glycine system I would have to come up with the CV's for example in this task the **free energy surface**<sup>[11]</sup> (FES) is fully known and described by a set of  $N$  CV's,  $z_i(q)$  with  $i = 1..N$ , which are all functions of system's particle positions  $q(t)$ . We will also assume that these particle positions  $q(t)$  evolve in time with a **canonical equilibrium distribution**<sup>[12]</sup> at temperature  $T$  under a potential  $V(q)$ .

For example in the [research](#) authors explicitly describe several CVs that biased or restrained during their meta-dynamics/CP-MD runs:

- In the exploration stage - drive the system from reactants toward glycine they used the path CVs  $s$  and  $z$ .
  - $s$  measures progress along a reference pathway built from reactant to product structures;  $z$  is the orthogonal deviation from that path. The distance metric for the path was the distance-mean-square deviation (DMSD) in exploration and the MSD in FES runs.
- When they calculated free-energy-surfaces (FES) calculations, the same path CV  $z$  was kept within 0.14 Angstrom
- The cartesian inter-atomic distances between selected atoms(e.g, the two carbon atoms that form the new C-C bond, and the C of CO with the O of a nearby water). Which is very similar to the example that was used in the master class that used similar distances and the angles between the atoms.
- In the proton-transfer sub-reaction which is a usual thing in the molecular dynamics(water to formalimine formulation) the  $N-H$  cartesian distance was measured between the nitrogen of formalimine and a proton from a water molecule.
  - Only two reference points (neutral and protonated) were needed; bias was applied directly to this distance.

# Rare events

In real life (experiment)

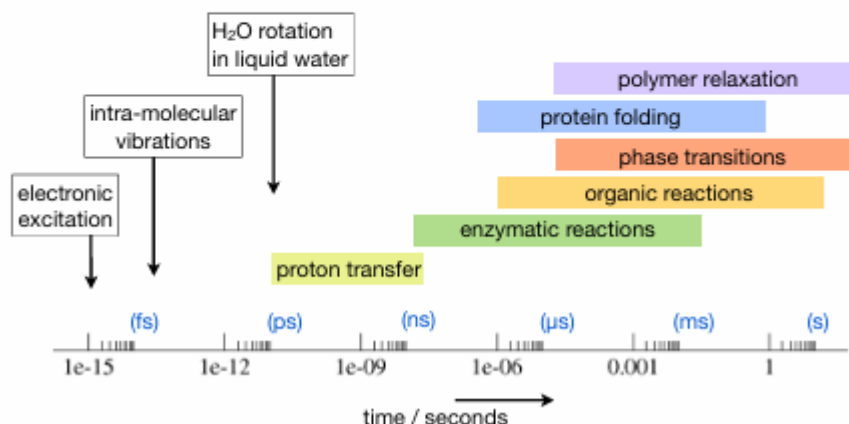


Fig 2

Representation of the time scale for the "rare events" which we describe in the manner of changes in the chemical reaction. For example we're interested in the nanosecond scale. Slide from representation the plumed masterclass 9.

## 2.2 Rare Events in MD

In molecular systems, the experimentally observed timescales on which they occur, and why each regime is painful to simulate with straightforward MD.

Each type of those timescales give us different properties.

- Electronic excitation and non-radiative decay (photo-excitation, internal conversion). Which is on timescale of  $1 - 100\text{fs}$  ( $10^{-15} - 10^{-13}\text{s}$ ). The nuclear motion is slower than electronic motion, because of that classical MD cannot follow electrons. Needs sub-fs steps. In my example simulations that are in my [github repository](#) is an example of a simulation that was done in timestep  $dt$  in picoseconds( $ps$ ) as a  $dt = 0.0002ps$  is a  $0.0002ps \times \frac{1000fs}{1ps} = 2fs$ . in that case full quantum treatment of electrons and nuclei(nonadiabatic dynamics), tiny time steps, high cost of excited-state potential energy surfaces are key simulation challenges. For those problems we have mixed quantum-classical techniques like [surface hopping](#) but that's not for now to talk about.

The last was contact control during FES runs. The coordination-number CV between heavy-atom groups ( $G_1, G_2$ ) were used. They restrained or monitored the number of contacts. Defined with the **standard smooth switching function**<sup>[13]</sup> in PLUMED.

For the meta-dynamics we are interested in exploring the space spanned by the CVs.  $z_i$  and quantifying the free energy. In order to do so, in meta-dynamics we exert a history-dependent

repulsive potential by summing **Gaussian kernels**<sup>[14]</sup> over time. That's the summation

$$V_{bias}(z, t) = \sum_{k\tau < t} H(k\tau) \exp \left( - \sum_{i=1}^N \frac{(z_i - z_i(q(k\tau)))^2}{2W_i^2} \right)$$

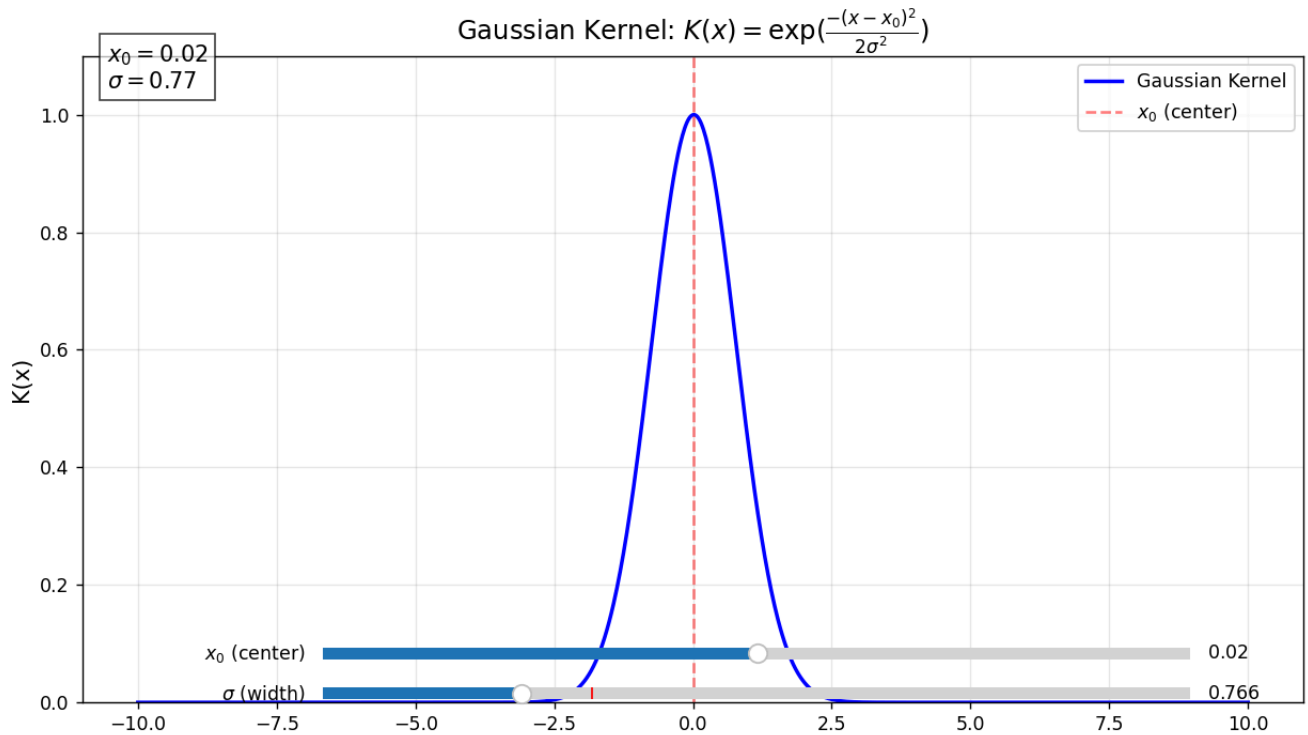
## 2.3 Gaussian Kernel: Derivation & Interpretation

Lets start with the easy examples. We should think of gaussian kernel as a smooth, bell-shaped hill that you drop at a specific point in space( or along a variable). Here's a breakdown:

- **The “Bell Curve” Shape** in Gaussian is just the familiar bell curve you’ve seen in statistics class. It’s highest at its centre and falls off smoothly as you move away. In many algorithms (like meta-dynamics, or in machine learning methods such as [kernel density estimation](#)), we place (“kernel”) a little bell-shaped hill wherever the system visits. Each hill nudges the system away from places it’s already been, encouraging exploration. Mathematically, in one dimension it looks like

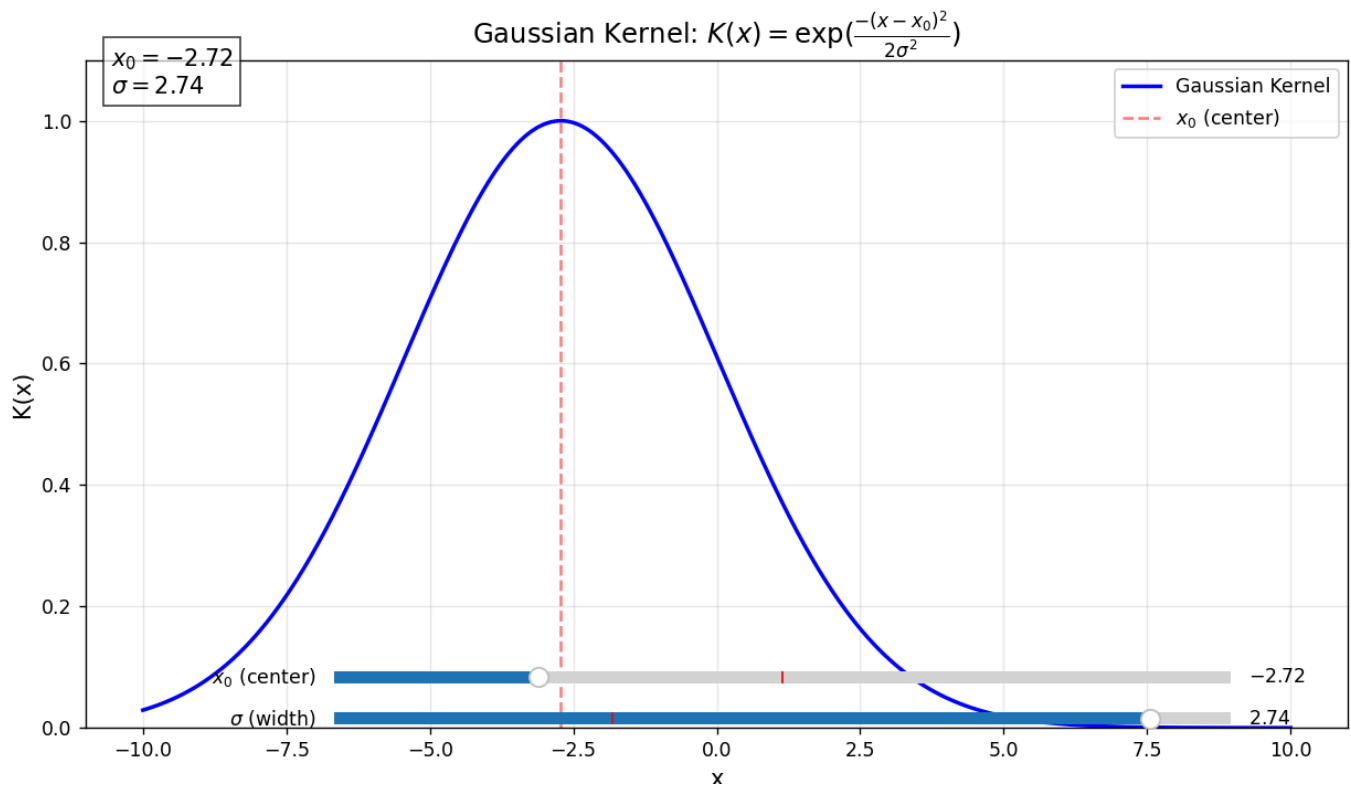
$$K(x) = \exp \left( - \frac{x - x_0^2}{2\sigma^2} \right)$$

where  $x_0$  is the centre of the hill and  $\sigma$  controls the ditch (a larger  $\sigma$  means a fatter, wider hill)



**Fig. 3**

*With the centre almost on the origin ( $x_0 \approx 0$ ) and a small width  $\sigma \approx 0.77$ , the Gaussian forms a tall, needle-like peak that falls to near-zero within just a few units of  $x$ . This setting illustrates a kernel that gives very local influence: it strongly weights points right at  $x_0$  while rapidly discounting anything farther away.*

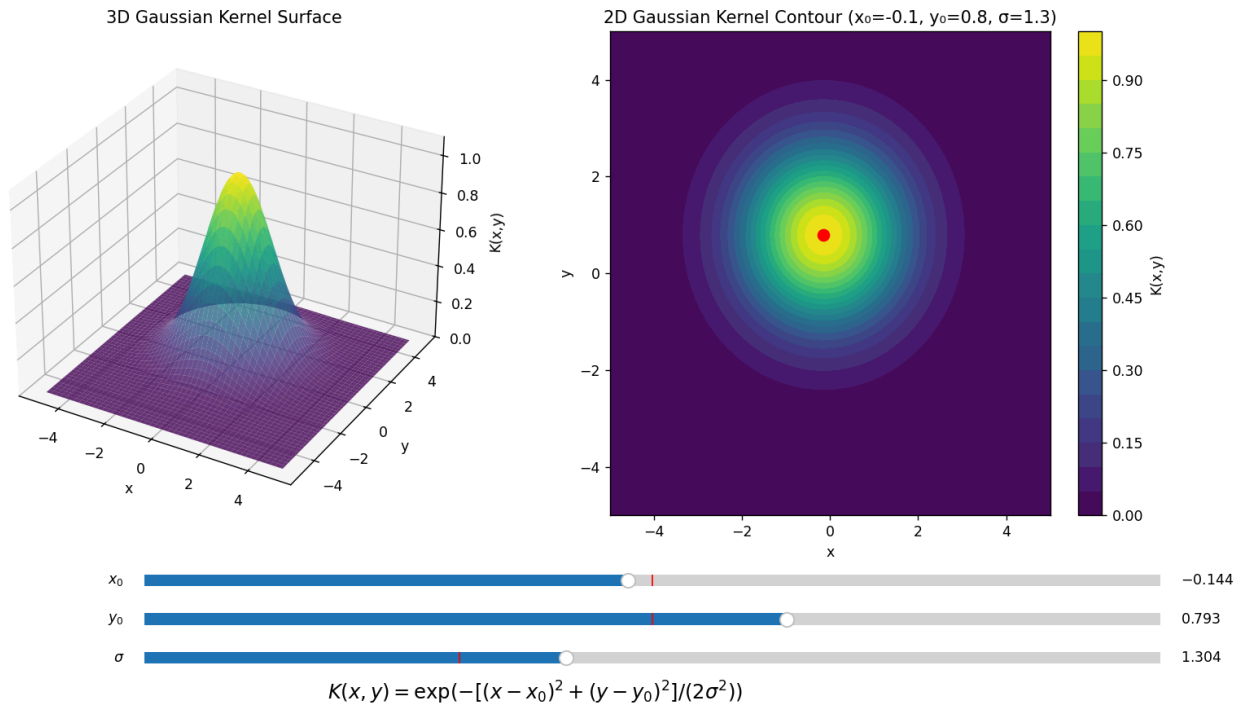


**Fig. 4**

Here the bell's centre is moved to  $x_0 \approx -2.7$  and the width is widened to  $\sigma \approx 2.7$ , producing a lower-slope mound that stays above zero across much of the axis. The larger  $\sigma$  spreads the kernel's influence over a wide range, while the shift in  $x_0$  simply translates the entire curve without altering its shape.

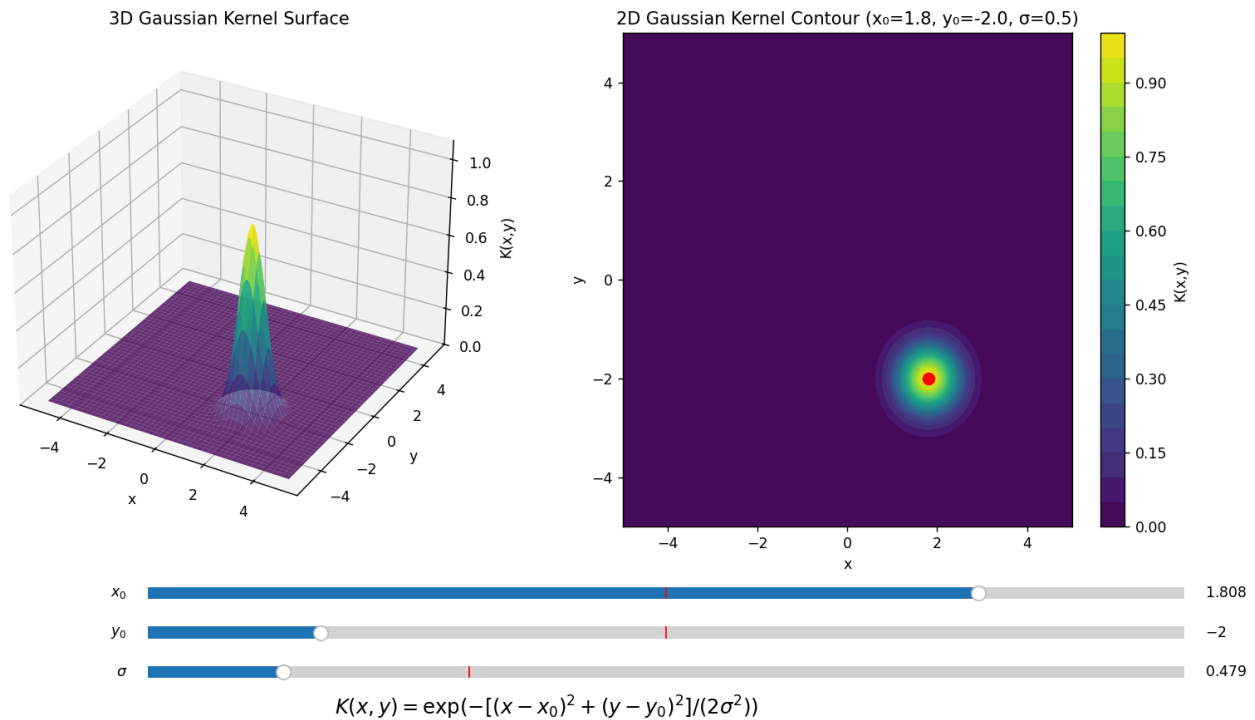
after that we have several variables(let's say  $x$  and  $y$ ), you just combine two such bells into one 2D hill

$$K(x, y) = \exp\left(-\frac{[(x - x_0)^2 + (y - y_0)^2]}{2\sigma^2}\right)$$



**Fig. 5**

The 3D surface shows a smooth, gently rounded Gaussian that tops out near  $(x_0, y_0) \approx (-0.14, 0.79)$ . With a width of  $\sigma \approx 1.3$  the peak is neither needle-thin nor extremely flat, so the kernel still dominates close to its centre but fades only gradually as you move a few units away. The concentric rings in the contour plot confirm this: their spacing widens slowly, signalling a fairly broad region of influence around the red centre point.



**Fig. 6**

Here the kernel is pushed to  $(x_0, y_0) \approx (1.8, -2.0)$  and squeezed to a narrow  $\sigma \approx 0.48$ . The 3D view reveals a steep, slender spike whose base occupies just a small patch of the plane. Correspondingly, the contour plot displays only a handful of closely packed rings around the red dot, illustrating how quickly the kernel's weight plummets: it grants strong emphasis to points very near  $(1.8, -2.0)$  and almost none elsewhere.

The hill is infinitely smooth, no sharp corners. It won't **jerk**<sup>[15]</sup> the system around. The bias only affects the nearby region (because it decays exponentially), so we only push the system out of the places it's already been sampled well.

Now when we know what the Gaussian kernel is, we can take the natural next step in the meta-dynamics formalism: first we recall that in the long-time limit the accumulated bias exactly cancels out the underlying FES. In other words, if you keep dropping these Gaussian hills forever, the sum of all those hills fills in every free-energy well so that

$$\lim_{t \rightarrow \infty} V_{\text{bias}}(z, t) = -F(z) + \text{const.}$$

where  $F(z)$  is the true free energy as a function of the collective variables  $z$

So putting it all together we:

1. **Deposit Gaussians** every  $\tau$  time units at the current CVs  $z(q(k\tau))$ , each with height  $H$  and widths  $W_i$ .
2. **Bias builds up** because you avoid revisiting already-sampled regions - the system is "pushed" into new territory.
3. **Free-energy estimate** emerges in the long-time limit, since

$$V_{\text{bias}}(z, t) = \sum_{k\tau < t} H(k\tau) \exp \left( - \sum_{i=1}^N \frac{(z_i - z_i(q(k\tau)))^2}{2W_i^2} \right) \rightarrow -F(z)$$

In standard meta-dynamics, all hills have the same height  $H$ . In **well-tempered meta-dynamics**<sup>[16]</sup>, one lets the effective hill height decay as more bias accumulates:

$$H_k = H_0 \exp \left( - \frac{V_{\text{bias}}(z(q(k\tau)), k\tau)}{k_B \Delta T} \right),$$

so that deep wells get filled more slowly. The resulting bias converges to a fraction of the true free energy:

$$\lim_{t \rightarrow \infty} V_{\text{bias}}(z, t) = - \frac{\Delta T}{T + \Delta T} F(z) + \text{const.}$$

so by choosing the "bias temperature"  $\Delta T$ , you control the trade-off between exploration speed and accuracy of  $F(z)$ . Meta-dynamics also has several extensions to accelerate



convergence and improve computational performance, such as the **multiple-walker**<sup>[17]</sup> and said well-tempered versions.

Meta-dynamics can handle a few CVs simultaneously in a trivial manner, which spares us from having to find a single perfect CV. Instead, an appropriate set of CVs allows us to converge an insightful multidimensional free energy landscape, in which (meta)stable states and connecting MFEPs can be identified. In practice however, especially when investigating complex transitions, the number of CVs is limited to  $N \approx 3$ , because of the exponential growth of computational cost with CV dimensionality.

For the glycine what would be not enough we need to find another way

## 2.4 Path-Meta-dynamics

In order to tackle complex transitions that require many CVs, path-based methods were developed([\*parallel-tempering meta-dynamics\*](#), [\*bias-exchange meta-dynamics\*](#), [\*transition-tempered meta-dynamics\*](#), [\*infrequent meta-dynamics\*](#), [\*funnel meta-dynamics\*](#), [\*adaptive gaussian meta-dynamics\*](#), [\*on-the-fly probability enhanced sampling\*](#)).

For example, if we want to make [\*SN2 nucleophilic substitution\*](#) in solution:

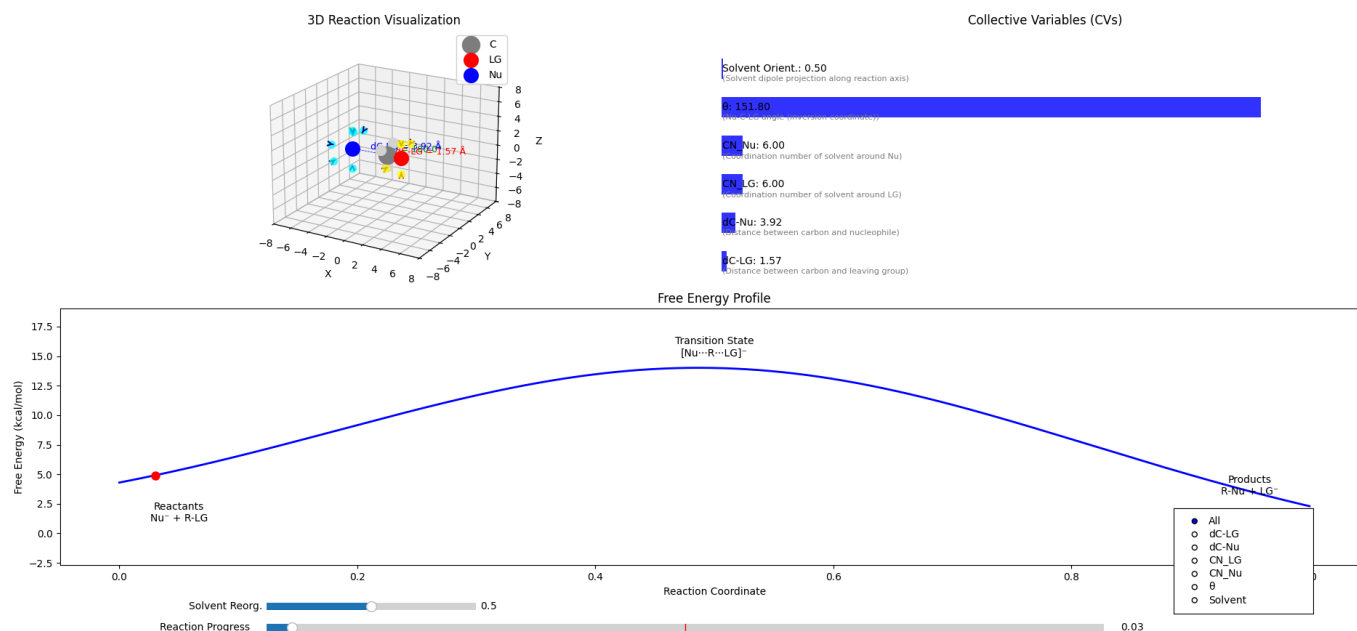
Key chemical events are concerted bond-forming to the nucleophile (Nu) and bond-breaking from the leaving group (LG), plus solvent reorganization around both.

Typical CVs( $N \approx 5 \rightarrow 8$ ) is that reaction would be:

1.  $(d_C)_{LG}$  the distance between the central carbon and LG
2.  $(d_C)_{NU}$  the distance between the central carbon and Nu
3.  $CN_{LG}$  the coordination number of solvent oxygens around LG
4.  $CN_{NU}$  the coordination number of solvent oxygens around Nu
5.  $\theta_{LG-C-Nu}$ , the Nu-C-LG angle(inversion coordinate)
6. A solvent-orientation collective coordinate (e.g. projection of water dipole along the reaction axis)
7. Optional dihedral(s) if the Nu or LG has internal rotors that gate the approach

Those many CVs are necessary for the SN2. The SN2 barrier and mechanism depend not just on the two bond distances but also on how the solvent shell reorganizes to stabilize charge build-up on LG and Nu in the transition state.

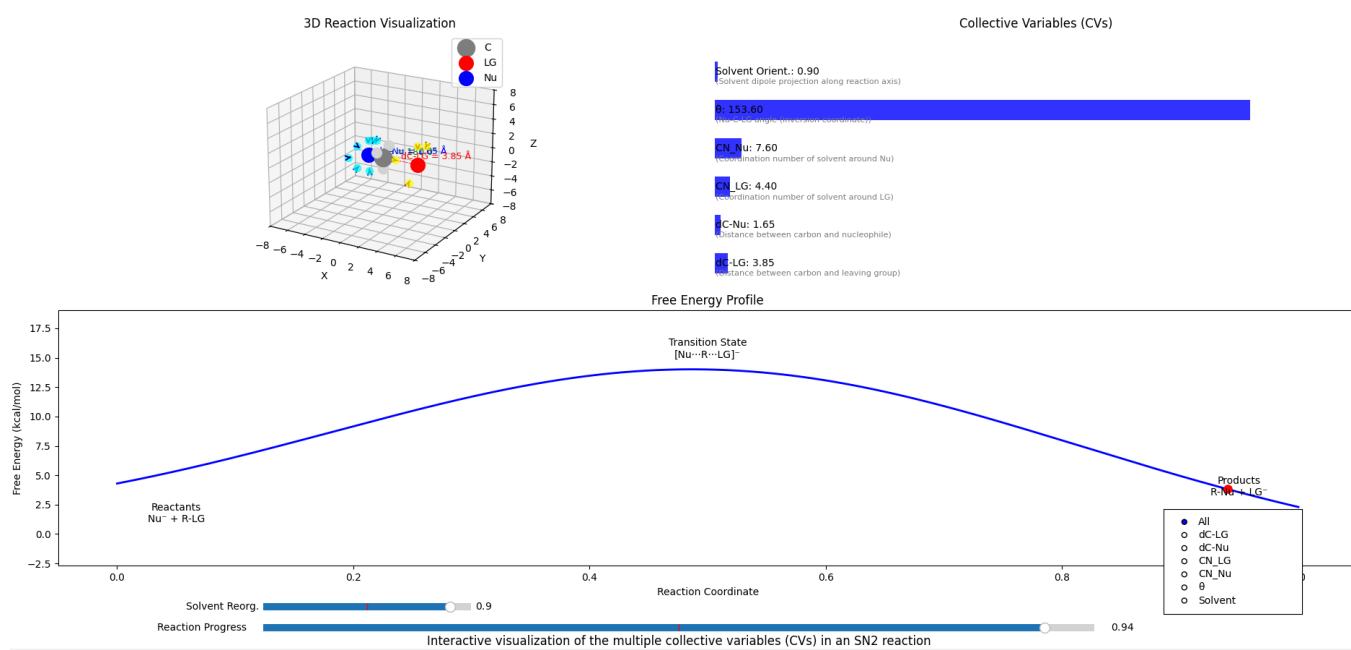
## SN2 Nucleophilic Substitution in Solution



**Fig. 7**

The scene captures early, “reactant-side” snapshot where the  $SN2$  system just after the nucleophile (blue) begins its approach. In the 3D view the  $Nu - C$  distance is still long, while the  $C - LG$  bond (red) remains intact. The bar chart of collective variables (CVs) shows moderate solvent re-orientation and only partial solvent coordination around both  $Nu$  and  $LG$ , consistent with an early stage. In the free-energy profile the marker sits near  $\sigma \approx 0.3$ , on the rising limb that leads to the transition state, and the slider beneath confirms that reaction progress is still low ( $\approx 3\%$ ).

## SN2 Nucleophilic Substitution in Solution



**Fig. 8**

Here the nucleophile has almost completed the substitution: the  $Nu - C$  bond is short, the  $LG - C$  bond is greatly elongated, and solvent molecules have reorganised to stabilise the

emerging charges. The CV bars reflect this shift - large coordination to  $Nu$ , diminished coordination to  $LG$ , and an increased solvent-orientation value. On the free-energy plot the marker now lies far to the right ( $\sigma \approx 0.94$ ) on the downhill path from the transition state toward products, matching the reaction-progress slider that reads ( $\approx 94\%$ ).

### In these schemes, a path-CV is introduced.

Those paths are simply a continuous, one-dimensional curve through the high-dimensional CV space that links two end states  $A$  and  $B$ . Concretely, a path is a parametrized mapping  $s(\sigma) : [0, 1] \rightarrow R^N$  such that,  $s(0)$  sits in basin  $A$  (reactant or initial state) and  $s(1)$  sits in basin  $B$  (product or final state)

Then we got the initial guesses ("schemes") which consists of:

1. [Linear interpolation](#) in CV-space. We simply draw a straight-line between the CV-vectors of  $A$  and  $B$  and discretize it into equal segments.
2. String method guess, which use a rough preliminary path from a coarse string-method or from a meta-dynamics run.
3. **Nudged Elastic Band**<sup>[18]</sup> (NEB) is a chain of replicas (images) connected by springs; you can take those images as your initial path nodes.
4. **Committer-based sampling**<sup>[19]</sup>, a short run of unbiased MD bursts started along some CV-interpolated pathway and stitch together average transition points to form a path.

And once we picked an initial path  $s^0(\sigma)$ , path-meta-dynamics iteratively "bends" it toward regions of **high transition flux**<sup>[20]</sup> by sampling points  $z_k$  near the current path via meta-dynamics bias on the progress coordinate  $\sigma(z)$ . Computing the average positions of those samples in each  $\sigma$  slice and updating the discrete path nodes toward those averages (while re-equidistantly reparametrizing)

**We would like to ask, why to bother with that?** But the answer is clear. *This technique collapses an  $N$ -dimensional problem onto a single coordinate  $\sigma$  along the path!*

We now don't need to fill an exponentially large **grid in CV-space**<sup>[21]</sup>, only to refine a one dimensional curve. In the narrow "tube" limit, the same machinery recovers the minimum free-energy path (MFEP); in the wider limit, you get the average reactive flux path.

This progress parameter emulates the reaction coordinate and can in principle be connected to the committor value. Since the transition path curve is not known [a priori](#), the path-CV must be adaptive. If we assume that, in the vicinity of the path, the **iso-committor planes**<sup>[22]</sup>  $S_\sigma$  are perpendicular to  $s(\sigma)$ , and that the configurational probability is a good indicator of the transition flux density, then we can converge the average transition path by iteratively adapting a guess path to the cumulative average density:  $s_g(\sigma_g) = \langle z \rangle_{\sigma_g}$

We slice our MD samples into "bins" according to their current value of the progress coordinate  $\sigma(z)$ . For each bin(labelled  $\sigma_g$ ), we compute the arithmetic mean of the z-vectors in that bin:  $\langle z \rangle_{\sigma_g}$  then we move our path node  $s_g(\sigma_g)$  to that mean position.

Like it was said the key advantage of PMD is that the meta-dynamics biasing can be performed on the 1D progress parameter along the adaptive path-CV, instead of the  $N$ -dimensional CV-space. PMD is able to converge a path, and the free energy along it, with a [sublinear rise](#) in computational cost with respect to the growth in CV dimensionality (instead of the exponential relation when not using a path-CV)

### How it is calculated?

Numerically, the adaptive path-CV is implemented as a set of  $M$  ordered nodes (coordinates in CV-space)  $s_g(\sigma_g, t) \rightarrow s_j^{t_i}$  with  $j = 1, \dots, M$ , where the  $t_i$  is the discrete time of path updates. The [projection](#) of any point  $z$  onto the path is done considering the closest two nodes, and  $\sigma$  is obtained by interpolation. This approach requires [equidistant](#) nodes which are imposed by a reparameterization step after each path update. The update step for the path nodes is done by

$$s_j^{t_{i+1}} = s_j^{t_i} + \frac{\sum_k w_k \times (z_k - s^{t_i}(\sigma(z_k)))}{\sum_k \xi^{t_i-k} w_{j,k}}$$

where

$$w_k = \max \left[ 0, \left( 1 - \frac{\| s_j^{t_i} - s^{t_i}(\sigma(z_k)) \|}{\| s_j^{t_i} - s_{j+1}^{t_i} \|} \right) \right]$$

, where  $k$  is the current MD step and  $w$  is the weight of the adjustment, which is non-zero only for the two nodes closest to the average CV density.

This is very hard to unpack at the first glance, but I will try. Let's unpack the numerical update of the path-nodes  $s_j^{t_i} \rightarrow s_j^{t_{i+1}}$  in a series of clear steps

1. **Discretize the path into nodes.** At update time  $t_i$ , the continuous path  $s(\sigma)$  is represented by  $M$  discrete nodes

$$\{s_1^{t_i}, s_2^{t_i}, \dots, s_M^{t_i}\}$$

These are spaced evenly in the  $\sigma$ -parameter (after the previous reparameterization).

2. **Collect MD samples  $\{z_k\}$  since the last update.** As the MD runs (with meta-dynamics bias on  $\sigma$ ), you accumulate configurations  $z_k$  at each MD step  $k$ .
3. **Project each sample onto the current path.** For each sample  $z_k$ , we need to find it's projection  $s^{t_i}(\sigma(z_k))$  on the discrete path(by locating the two nearest nodes and linearly interpolating). After that we compute the vector displacement  $\Delta z_k = z_k - s^{t_i}(\sigma(z_k))$
4. **Compute the local weight  $w_{j,k}$  of sample  $k$  on node  $j$ .** In our case only the two nodes bracketing the projection get nonzero weight. For node  $j$

$$w_k = \max \left[ 0, \left( 1 - \frac{\| s_j^{t_i} - s^{t_i}(\sigma(z_k)) \|}{\| s_j^{t_i} - s_{j+1}^{t_i} \|} \right) \right]$$

geometrically, this tapers linearly from 1 (if the sample projects exactly onto node  $j$ ) down to 0 at the next node.

5. **Apply a "fade" factor for older samples.** To let the path "forget" old history, each past sample at MD step  $k$  is further multiplied by

$$\xi^{t_i-k}, \xi = e^{-\ln(2)/\tau_{1/2}}$$

so that samples older than one half-life  $\tau_{1/2}$  count half as much.

6. **Accumulate weighted displacements and normalize.** For each node  $j$ , build the numerator and denominator

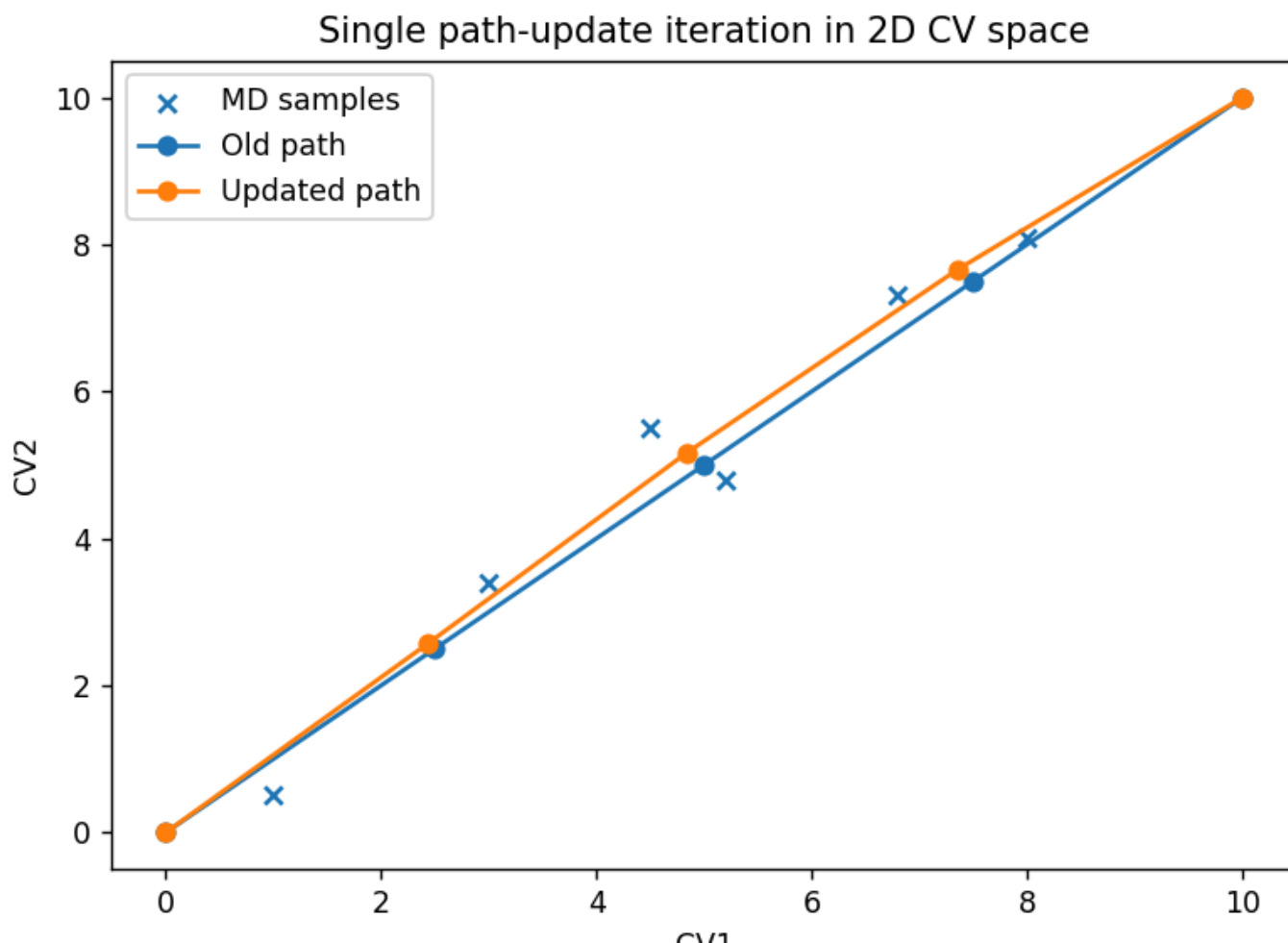
$$\text{numerator}_j = \sum_k (\xi^{t_i-k} w_{j,k}) \Delta z_k$$

$$\text{denominator}_j = \sum_k \xi^{t_i-k} w_{j,k}$$

7. **Update the node position.** Shift each node by the average weighted displacement:

$$s_j^{t_{i+1}} = s_j^{t_i} + \frac{\text{numerator}_j}{\text{denominator}_j}.$$

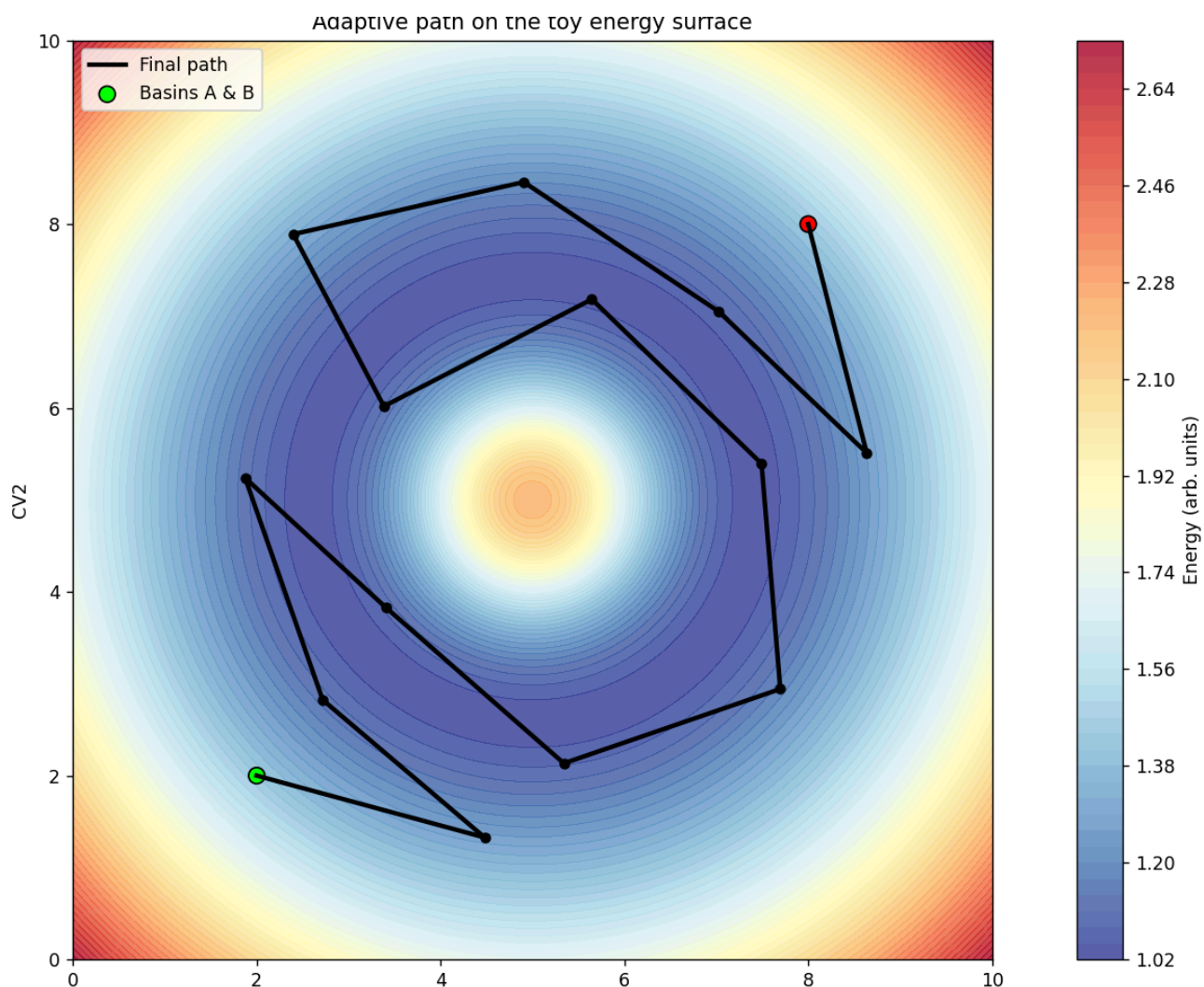
After those steps finally, the  $M$  updated nodes are "re-evened out" so that they again sit at equal  $\sigma$ -intervals along the new curve  $s(\sigma)$ . So the discrete path steadily bends toward the true, flux weighted transition tube.



**Fig. 9**

One adaptive-update step in a toy 2D CV space. Blue crosses mark the MD configurations gathered since the last update; the blue line with dots is the original straight-line guess connecting reactant and product. Applying the node-update rule pulls each node toward the weighted mean of the samples in its neighbourhood, yielding the orange path. Notice how every node has shifted slightly upward, exactly where the samples are denser, illustrating how path-meta-dynamics bends an initial guess toward the true, flux-weighted transition tube while keeping the nodes evenly spaced along the progress coordinate  $\sigma$ .

Additionally, we can improve the method by placing trailing nodes at both ends of the path to better capture the free-energy basins. For the basic code, the results look like this:

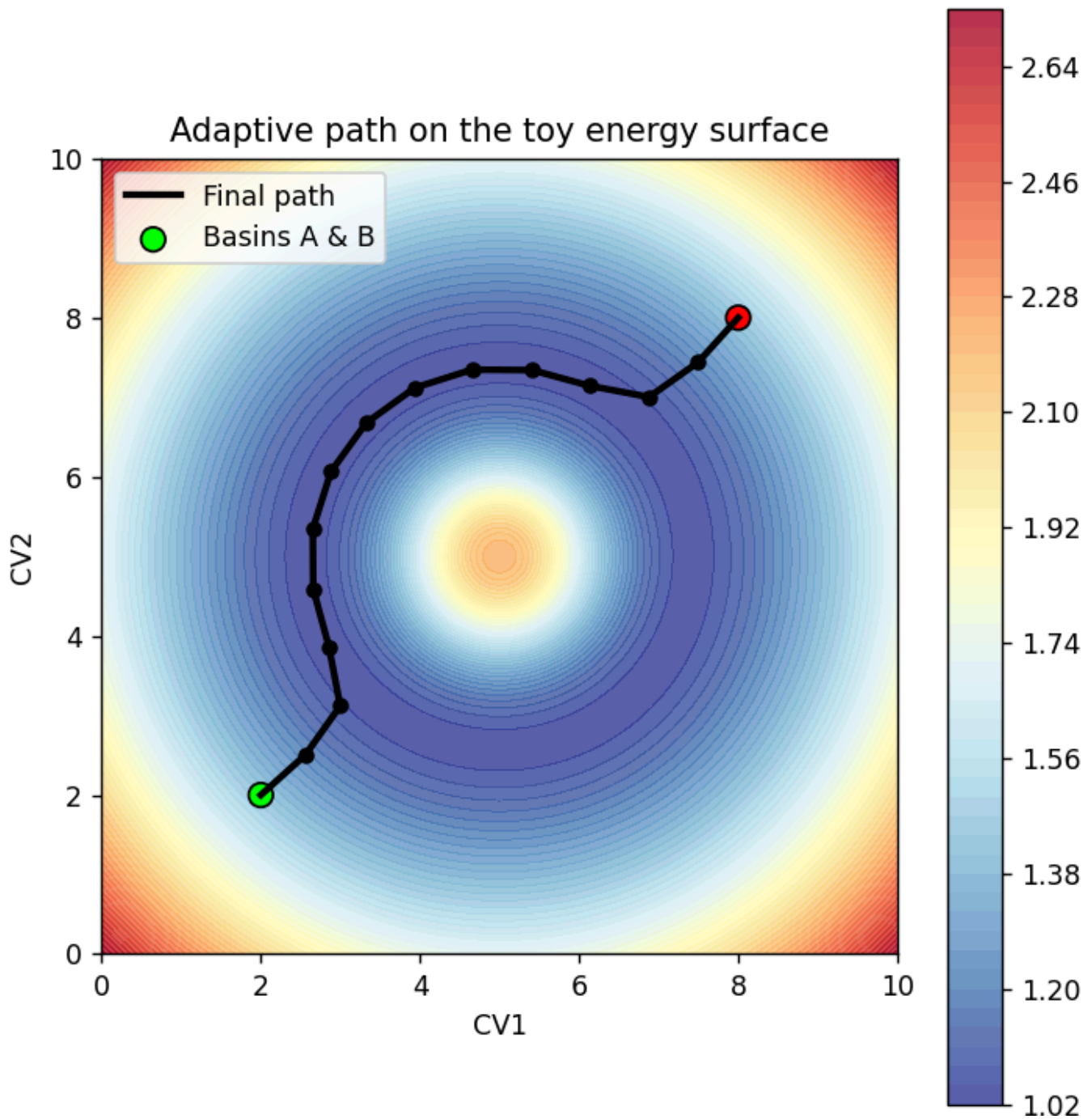


**Fig. 10**

*Straight-ish polyline, with no tube potential( or a very soft one), path cuts straight across contour rings, even wandering over higher-energy regions. The algorithm was density-driven. It found a shortcut that balances flux coming from different directions - good for the average reactive current, not optimal in free energy*



For the adaptive implementation using the tube mechanism, the results are as follows:



**Fig. 11**

*Smooth arc hugging the blue valley, with tube potential applied(probably fairly stiff), where path bends around the central barrier, consistently following the lowest-energy lane. The narrow tube forced sampling right on the curve, this updates followed local  $\nabla F$ , which recovered the MFEP*

The restraint we add is a [harmonic spring](#) on the distance between every instantaneous configuration  $z_k$  and its projection onto the current path  $s^{t_i}(\sigma(z_k))$

$$U_{\text{tube}}(z_k) = \frac{1}{2} k_{\text{tube}} \| z_k - s^{t_i}(\sigma(z_k)) \|^2$$

So in wide tube (small  $k_{\text{tube}}$ ) samples are free to stray a fair distance from the path. The path-



update rule therefore averages over a *broad* density of configurations you converge to the average transition path (sometimes called the reactive flux tube). With narrow tube (large  $k_{tube}$  "infinitesimally narrow" in the limit) sampling is confined to a thin corridor around the current curve. The only points that matter sit very close to the path, so the update step is dominated by the local free-energy gradient rather than by a global density average. In this limit the algorithm chases the minimum-free-energy path (MFEP) - the steepest-descent trajectory that traces the valley floor between A and B.

So because we can tune  $k_{tube}$  we have a knob that continuously interpolates between "density-driven" (average path) and "gradient-driven" (MFEP) behaviour.

Then we would work on the trailing nodes at both ends. Those are just extra, *flexible* points tacked onto the fixed endpoints in basins A and B. Because the meta-dynamics bias keeps pushing the system out of the wells, the trailing nodes can *slide down* the funnel and settle exactly where the free-energy valley floor is, giving the main section of the path a cleaner launch/landing zone. Without them, the first movable node would often sit too high up the wall of the basin and the path would start with an awkward kink. Something we can indeed spot in the first plot but not in the second.

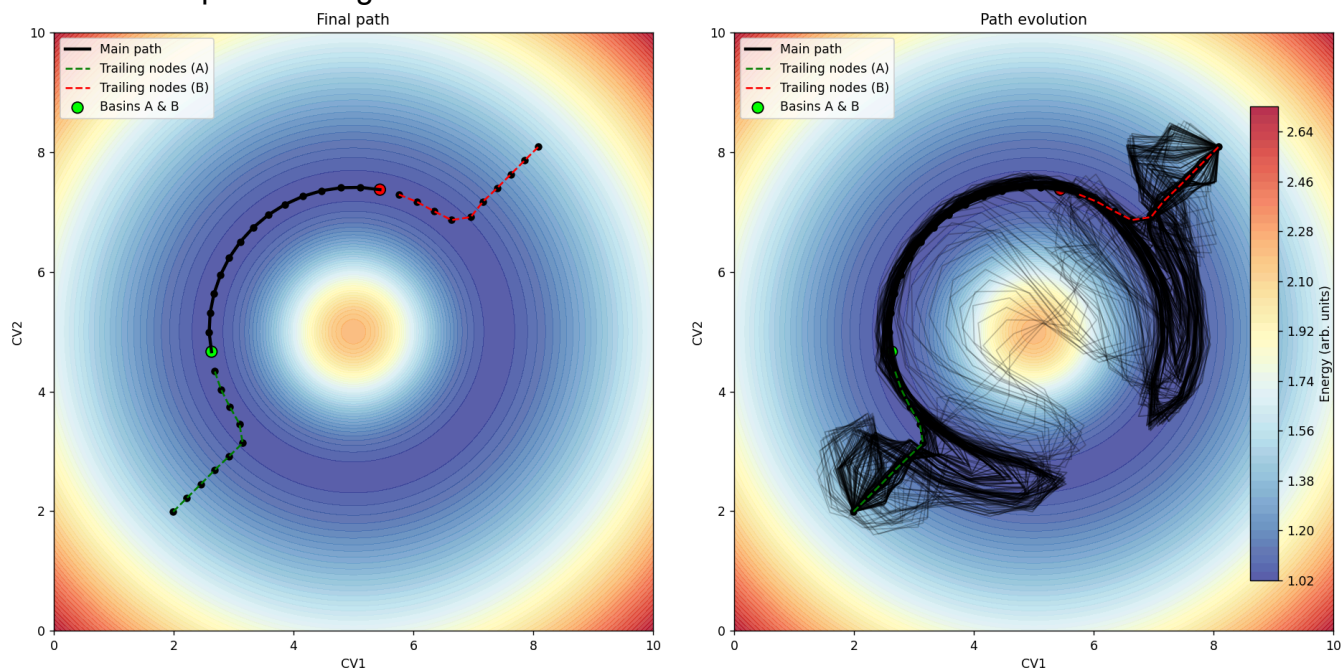
Meta-dynamics then steps in as the workhorse that breathes life into the adaptive path-CV: by depositing a time-dependent series of Gaussian hills directly on the progress coordinate  $\sigma$ , it repeatedly pushes the system back and forth along the curve, ensuring that every segment of the transition tube is visited many times.

Because each new hill partially overwrites the ones laid down earlier, the bias continually "self-heals," and in the long-time limit the accumulated potential converges to the negative of the true one-dimensional free-energy profile,  $V_{bias}(\sigma, t \rightarrow \infty) = -F(\sigma)$ . In practice this means that a single PMD run yields both the optimal pathway (average path or MFEP, depending on how tight you make the tube) *and* the quantitative free-energy barrier along it.

All of the usual meta-dynamics refinements - well-tempered damping of hill heights, transition-tempered focusing on the barrier region, multiple-walker parallelisation - plug in without modification, and one can even swap meta-dynamics for **umbrella sampling**<sup>[23]</sup>, **steered MD**<sup>[24]</sup>, or perpendicular biasing whenever a particular segment of the mechanism demands finer control.

Thus the path algorithm and meta-dynamics form a self-consistent loop: meta-dynamics drives exploration and reconstructs the free energy, the path update bends the curve toward the physically relevant channel, and the process iterates until both geometry and energetics have converged.

The whole output looks right now like this

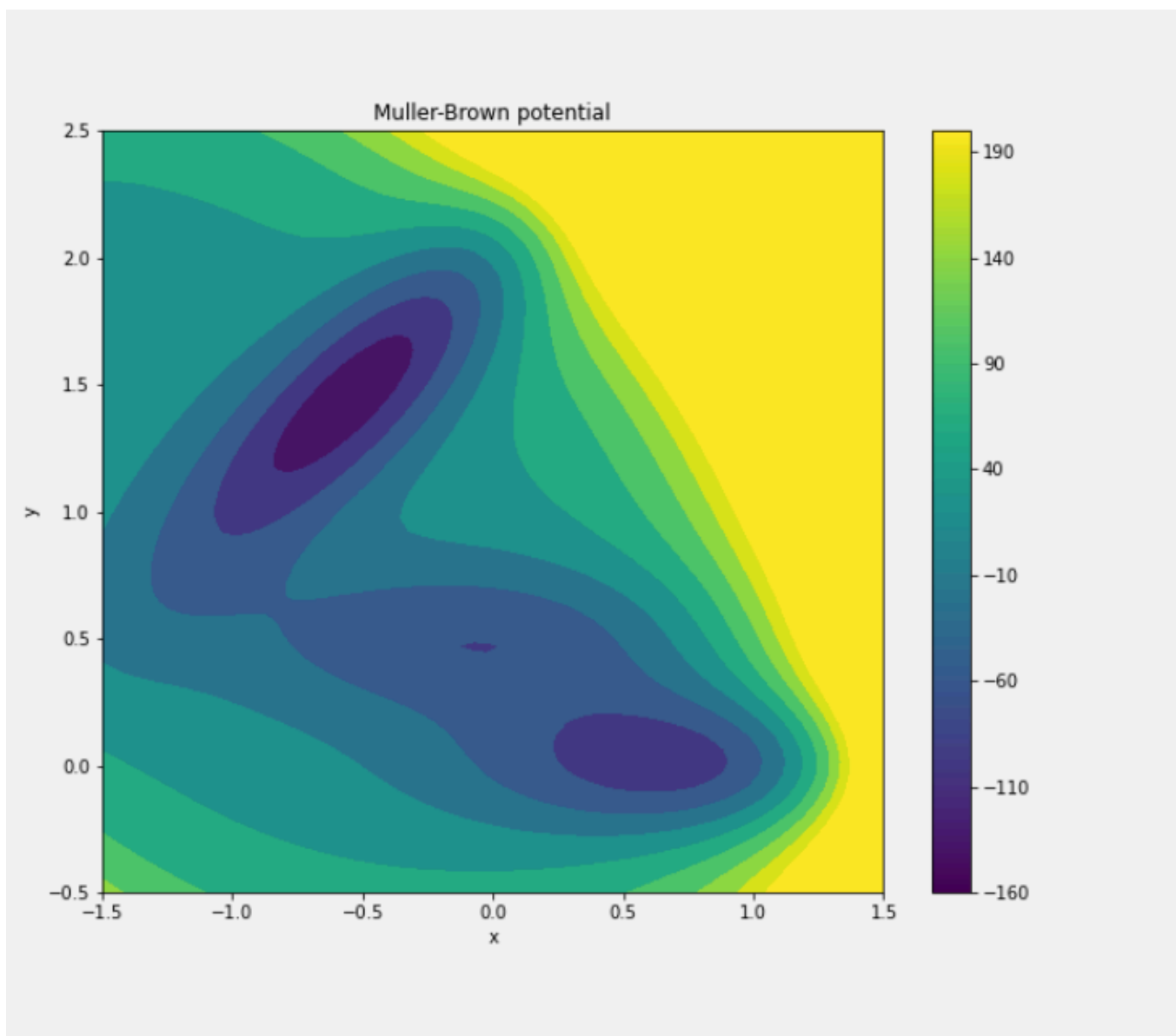


**Fig. 12**

*Left: the **final path** (black) connecting basins A and B winds smoothly along the valley floor of the 2D free-energy surface (contours and colour map). Green and red dots are the flexible trailing nodes that sit inside the basins, giving the main section a kink-free launch and landing. Right: a time-trace of the **path evolution**: every faint poly-line is one intermediate node set during successive updates. Beginning with a crude straight guess, the nodes first fan out within the wide harmonic tube, then tighten and slide downhill as the tube is narrowed, ultimately collapsing onto the single, high-flux channel shown on the left. The two panels together illustrate how the spring-tube restraint and iterative updates steer the curve from a density-dominated average toward the minimum-free-energy (MFEP) trajectory.*

Alright, that concludes our introduction to the topic. Let's now move on to the examples presented in the PLUMED Masterclass.

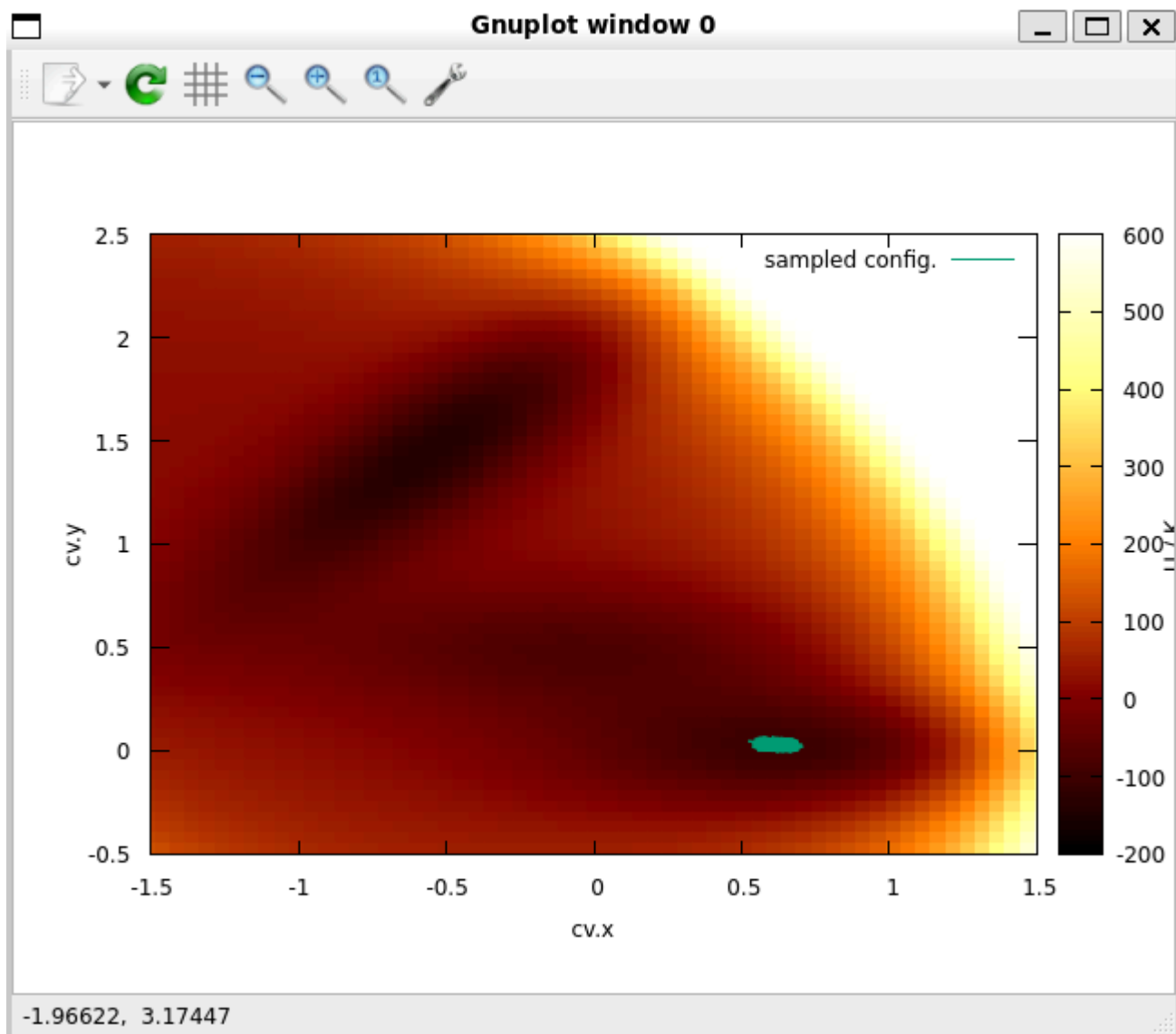
## 2.5 Müller–Brown Potential Exploration



**Fig. 13**

*Müller–Brown potential energy surface with contour shading reveals the rugged two–dimensional landscape used in the exercise: two deep minima (dark purple) sit in the lower-right and upper-left, separated by a broad saddle and flanked by a shallow mid-basin. Energies rise steeply toward the yellow plateau at the far right, creating  $\approx 5k_B T$  barriers between wells at the chosen simulation temperature.*

In our first exercise, we ran a straightforward MD simulation on the classic **Müller–Brown (MB)** [25] potential energy surface (PES), using PLUMED’s `pesmd` engine. Below is the snapshot of our sampling:

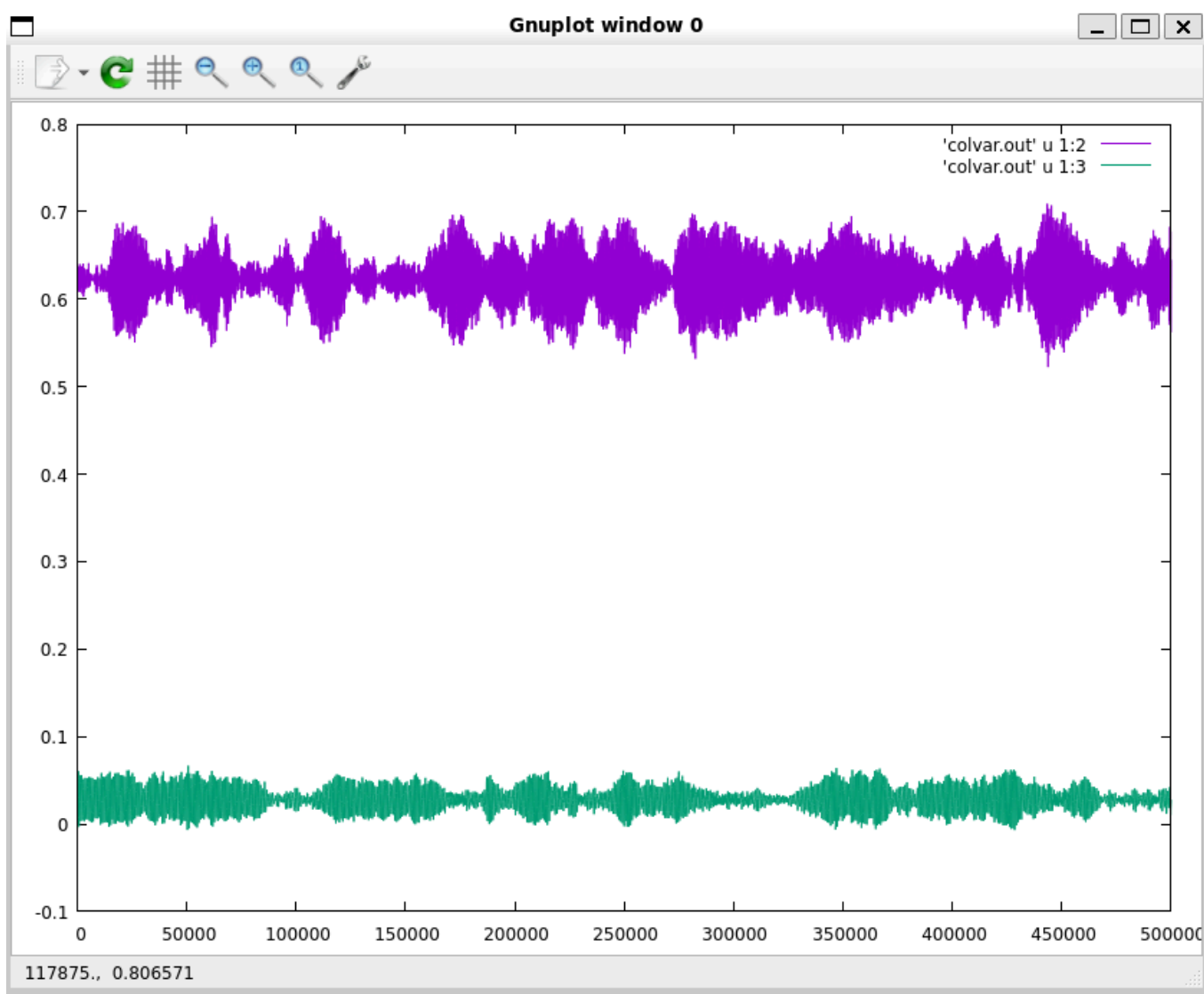


**Fig. 14**

*Plain-MD sampling footprint with the green trajectory remains trapped in the lower-right well, oscillating within a tight oval around  $(x \approx 0.6, y \approx 0)$ . No excursions reach the neighbouring basins or the ridge: with  $T = 0.5$  and friction  $\gamma = 0.1$  the system never acquires enough thermal energy to surmount the Müller-Brown barriers.*

We can see that the sampling is confined. The trajectory (green line) sits tightly around one minimum (around  $cv.x \approx 0.6$ ,  $cv.y \approx 0$ ). You never see it wander into the upper-left basin or cross the ridge to the right, it's “stuck” in that well. We can also see that the cloud of points is narrow ( $\Delta x, \Delta y \leq 0.05$ ), reflecting the low temperature ( $T = 0.5$ ) and moderate friction ( $\gamma = 0.1$ ). The [thermal fluctuations](#) are relatively small.

With these plain MD settings the system doesn't gain enough energy to hop the  $5k_B T$  barrier to the other wells.



**Fig. 15**

*Time traces of the collective variables. Purple ( $cv.x$ ) and teal ( $cv.y$ ) traces fluctuate around fixed mean values with amplitudes  $< 0.05$ , confirming the visual impression from **Fig. 14**. Over half-a-million integration steps the coordinates stay confined, illustrating the fundamental limitation of plain MD on rough landscapes: without biasing, rare barrier-crossing events are exceedingly unlikely on affordable time scales.*

## 2.5.1 Why Plain MD Gets “Stuck”?

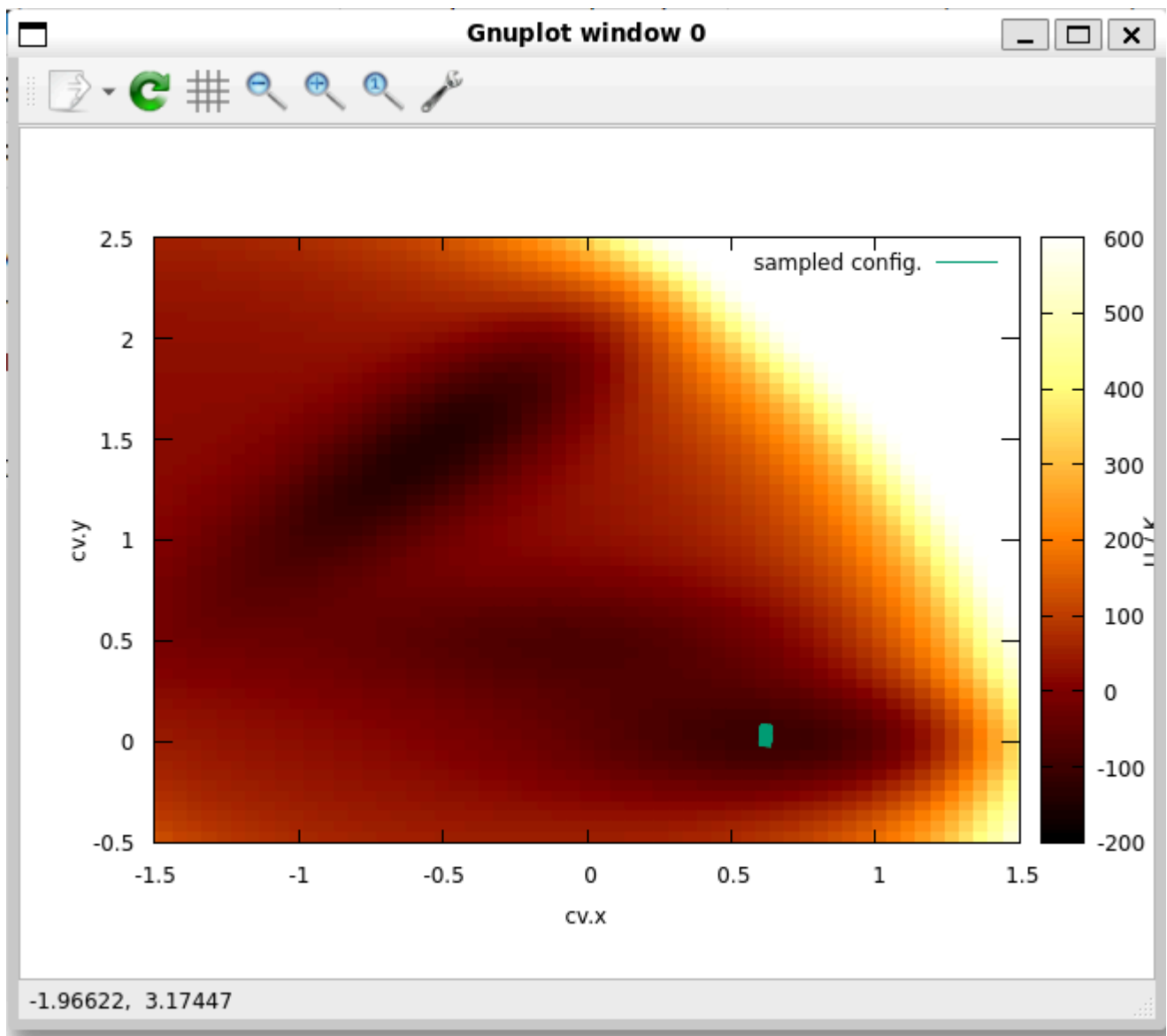
*(following the Müller-Brown exercise from the PLUMED Masterclass)*

The MB surface has multiple minima separated by significant barriers. In plain [Langevin dynamics](#), the temperature controls the average kinetic energy. Friction damps motion, making long “ballistic” hops rare. Timestep and simulation length merely scale how many small fluctuations you collect. At  $T = 0.5$ , the barrier height of  $\approx 5k_B T$  is prohibitive: rare events on MD timescales. If we want more exploration **without** invoking meta-dynamics or path-CV, we can tweak the MD “knobs”:

Parameter	Effect	Suggested Change
Temperature	Increases barrier hops	Raise $T$ to 1.0 or 1.5
Friction	Reduces damping	Lower $\gamma$ to 0.01 – 0.05
Timestep	Speeds up exploration	Increase $\Delta t$ (0.002–0.005) ps
Simulation length	More sampling	Double or triple <code>nstep</code>
Random seed / start	Different noise / basin	Change <code>idum</code> , <code>posx_init</code>

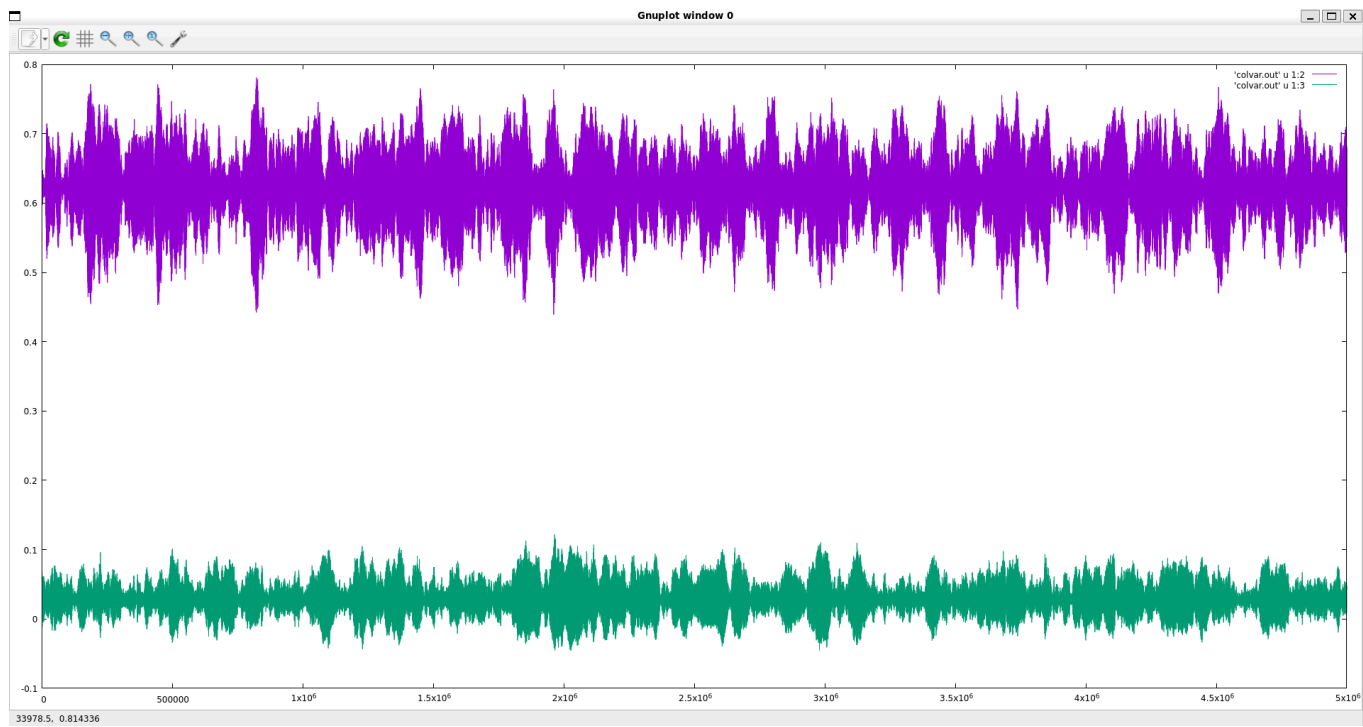
By increasing the temperature or reducing friction, the particle gains more kinetic energy and fewer **dissipative losses**<sup>[26]</sup> - making barrier crossings more probable. Extending the run length or using a larger timestep further improves the odds of hopping to other minima. These modifications are reflected in the input file.

```
nstep 5000000 ; # (was 500000)
temperature 1.5 ; # (was 0.5)
friction 0.05 ; # (was 0.1)
Δt 0.002 ; # (slightly increased from 0.001)
... (remaining inputs unchanged) ...
```



**Fig. 16**

*"Pushed" plain-MD still hugs the same well and even after cranking the knobs ( $T = 1.5$ ,  $\gamma = 0.05$ ,  $10 \times$  longer run) the cyan trajectory merely fattens its oval around the lower-right Müller-Brown minimum; no points reach the mid-basin or the upper-left well. The hotter, less damped particle explores a wider patch of the valley floor but the  $\approx 5k_B T$  ridge remains insurmountable without bias.*



**Fig. 17**

*Time traces confirm localisation.  $Cv.x$  (purple) now wanders by  $\pm 0.15$  and  $Cv.y$  (green) by  $\pm 0.5$ -roughly double the earlier amplitudes-yet both signals stay centred near their original means for five million steps. The absence of level-shifts or large excursions underscores the lesson: simply raising temperature or extending simulation time yields diminishing returns against **multi-kilojoule barriers**<sup>[27]</sup>, highlighting the need for dedicated biasing techniques.*

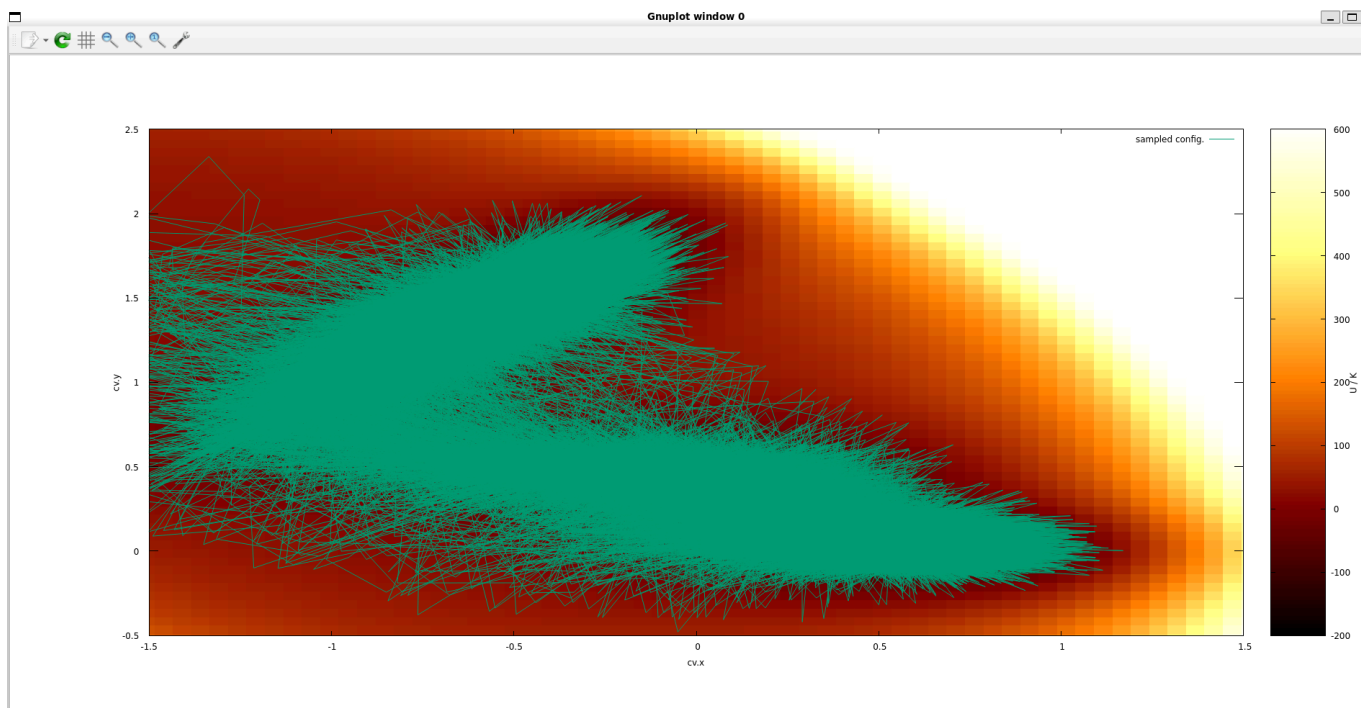
## 2.5.2 New basin escape approach

*(following the Müller-Brown exercise from the PLUMED Masterclass)*

Well, we've got some learning on that. **Plain MD - even when you push it - struggles to overcome free-energy barriers that are several  $k_B T$  high.** What I have demonstrated is that extending time or raising temperature yields diminishing returns and can distort the physics<sup>[28]</sup> if you go too far. Also to efficiently explore multiple basins, you need biasing methods that “fill in” barriers, rather than hoping for a lucky thermal kick.

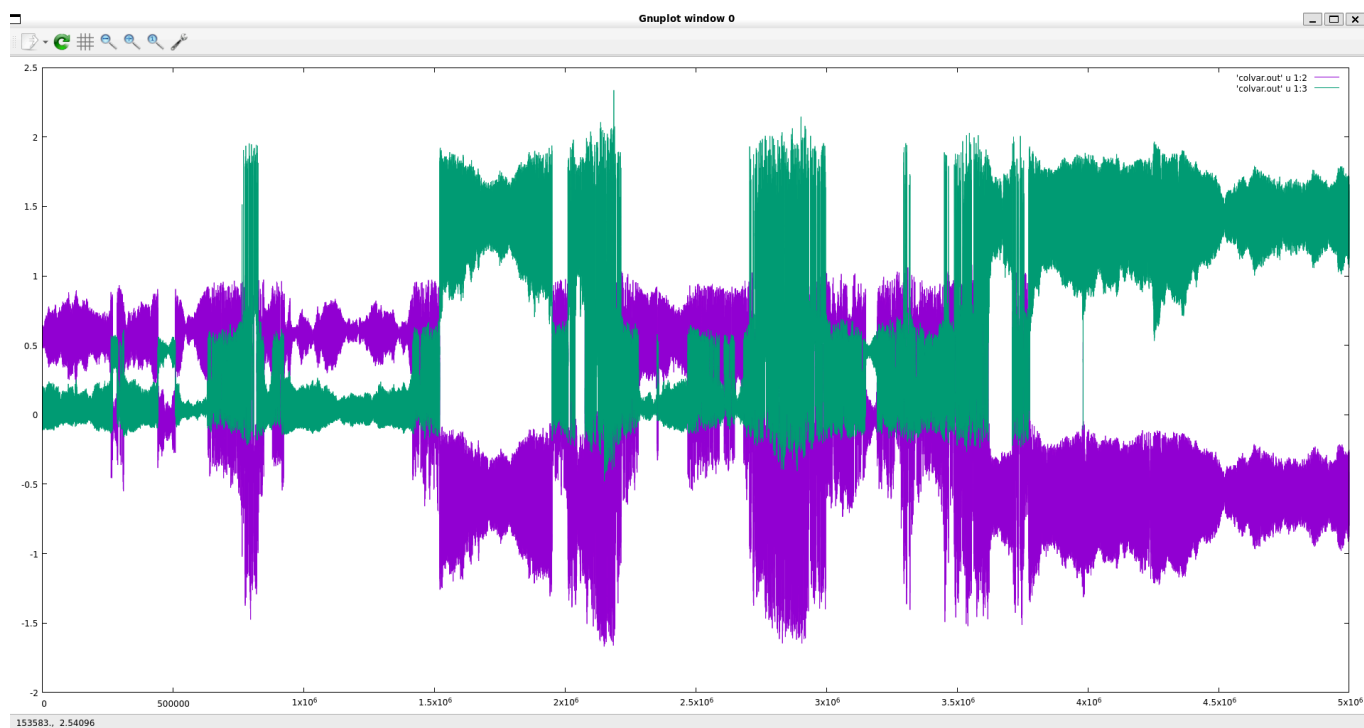
The final method involves increasing the temperature tenfold. Let's examine the results now, along with the corresponding step sizes.





**Fig. 18**

*Barrier-crossing achieved at the price of “over-heating.” Raising the temperature by an order of magnitude unleashes the walker: the turquoise trajectory now shuttles back and forth between the lower-right well and the mid-basin at ( $y \approx 1.5$ ), repeatedly traversing the Müller–Brown saddle. Although the run finally escapes the original minimum, the path fans out into high-energy (yellow) regions that would be thermally inaccessible at physically realistic  $T$ , illustrating how brute-force heating trades accuracy for reach.*



**Fig. 19**

*Collective-variable traces reveal erratic, high- $T$  dynamics. The purple  $Cv.x$  and green  $Cv.y$*

series exhibit large, abrupt jumps and long dwell periods in two distinct value ranges, confirming frequent basin-to-basin hops. Between these plateaus the signals spike to extreme values, signalling excursions onto steep ridges of the surface. Such jagged, high-amplitude motion underscores the drawback of the temperature-boost strategy: enhanced exploration comes with distorted kinetics and a loss of fidelity to the true free-energy landscape.

## 2.5.3 Introducing Meta-dynamics to the problem

(following the Müller-Brown exercise from the PLUMED Masterclass)

Let's take this insight forward by adding a simple **meta-dynamics** block to our `plumed.dat` file. Here's a minimal example that I will "build up" bias on  $cv_x$  and  $cv_y$

```
UNITS LENGTH=nm TIME=fs ENERGY=kcal/mol          # keep nm/fs because pesmd
expects it

# ---- collective variables -----
-
cv:      DISTANCE ATOMS=1,2 COMPONENTS              # gives cv.x cv.y (= the 2-D
coords)

# ---- harmonic "walls" to keep us on the grid -----
LOWER_WALLS ARG=cv.x,cv.y AT={-1.5,-0.5} KAPPA={1000,1000}
UPPER_WALLS ARG=cv.x,cv.y AT={ 1.5, 2.5} KAPPA={1000,1000}

# ---- well-tempered meta-dynamics -----
-
metad:  METAD ...
        ARG=cv.x,cv.y                # bias both coordinates
        SIGMA=0.10,0.10              # Gaussian widths      (hint says ≈0.1)
        HEIGHT=0.1                   # Gaussian height      (≈1 kBT at T=0.5)
        PACE=500                     # deposit every 500 steps
        BIASFACTOR=10                # well-tempered flavour
        FILE=HILLS                   # where Gaussians are stored
    ... METAD

# ---- output -----
-
PRINT ARG=cv.x,cv.y,metad.bias STRIDE=100 FILE=colvar.out
```

We may ask right now, how to choose meta-dynamics parameters? Well, When selecting parameters for meta-dynamics, a good approach is to match the Gaussian width (**SIGMA**) to the natural fluctuation scale of your collective variables -which usually is around 0.05 – 0.15 nm. If this width is too small, your energy surface becomes fragmented or "pixelated"; if it's too

large, you lose important details. The Gaussian height (**HEIGHT**) should be about  $1 k_B T$  (approximately  $0.1 \frac{\text{kcal}}{\text{mol}}$  at room temperature), which is gentle enough not to distort your system's kinetics yet effective enough to speed up sampling. Depositing Gaussians every 0.5 ps (**PACE** = 500 steps at 1 fs per step) strikes a balance: it's frequent enough to fill energy basins efficiently without excessive overlap. A typical **BIASFACTOR** of around 10 ensures your simulation remains "well-tempered," meaning it gradually smooths out the free-energy landscape without losing critical features.

In practice, **keep in mind that your Gaussians don't have to be spherical** - if your energy landscape has more pronounced features in one direction, consider using different widths for each collective variable. Additionally, aim for Gaussian heights roughly equal to one contour interval of your energy map; you don't want them so small they're negligible, nor so large they overwhelm the basin. For example, during a typical 5 ns simulation, you'll deposit around 10,000 Gaussians, enough to thoroughly explore a two-dimensional landscape without over-smoothing it.

These parameter choices will help you achieve publication-quality results by accelerating exploration without compromising the fundamental dynamics of your system.

- **SIGMA** should match the natural fluctuation scale ( $\approx 0.05 - 0.15 \text{ nm}$  here)
- **HEIGHT**  $k_B T$  biases gently without distorting kinetics
- **PACE** sets how often you add hills, 500 steps  $\approx 0.5 \text{ ps}$  with  $\Delta t = 1 \text{ fs}$

Here have the HILLS file, which shows where the Gaussians are located.

```

try more help for more information.
konrad@konrad:~/plumed_masterclass/masterclass-22-09/exercises/1_metadynamic
s$ more HILLS
#! FIELDS time cv.x cv.y sigma_cv.x sigma_cv.y height biasf
#! SET multivariate false
      500      0.6134572527167506      0.05216753593333578
    0.1      0.1      1
1
      1000      0.6119534368390872      0.04162710354708714
    0.1      0.1      1
1
      1500      0.650100578976523      0.001274386356911975
    0.1      0.1      1
1
      2000      0.6146880740398433      0.006460644182507157
    0.1      0.1      1
1
      2500      0.5755353961996159      0.01400800485418734
    0.1      0.1      1
1
      3000      0.5612959789567544      0.02059892241639384
    0.1      0.1      1
1
      3500      0.6309623995409657      0.02699995782594922
    0.1      0.1      1
1
      4000      0.7055951224971628      0.02639026267346495
    0.1      0.1      1
1
      4500      0.648318453545444      0.01599555968367015
    0.1      0.1      1

```

**Fig. 20**

*First few HILLS records in plain-text HILLS file lists every Gaussian that well-tempered meta-dynamics has deposited so far. For each entry you see the simulation time (steps), the centre in ( $Cv.x, Cv.y$ ), the widths  $\sigma_x, \sigma_y (= 0.1)$ , the Gaussian height (initially  $0.1 \frac{\text{kcal}}{\text{mol}}$ ), and the running bias factor. The gradual fall-off of the “height” column confirms that the bias is being tempered as intended.*

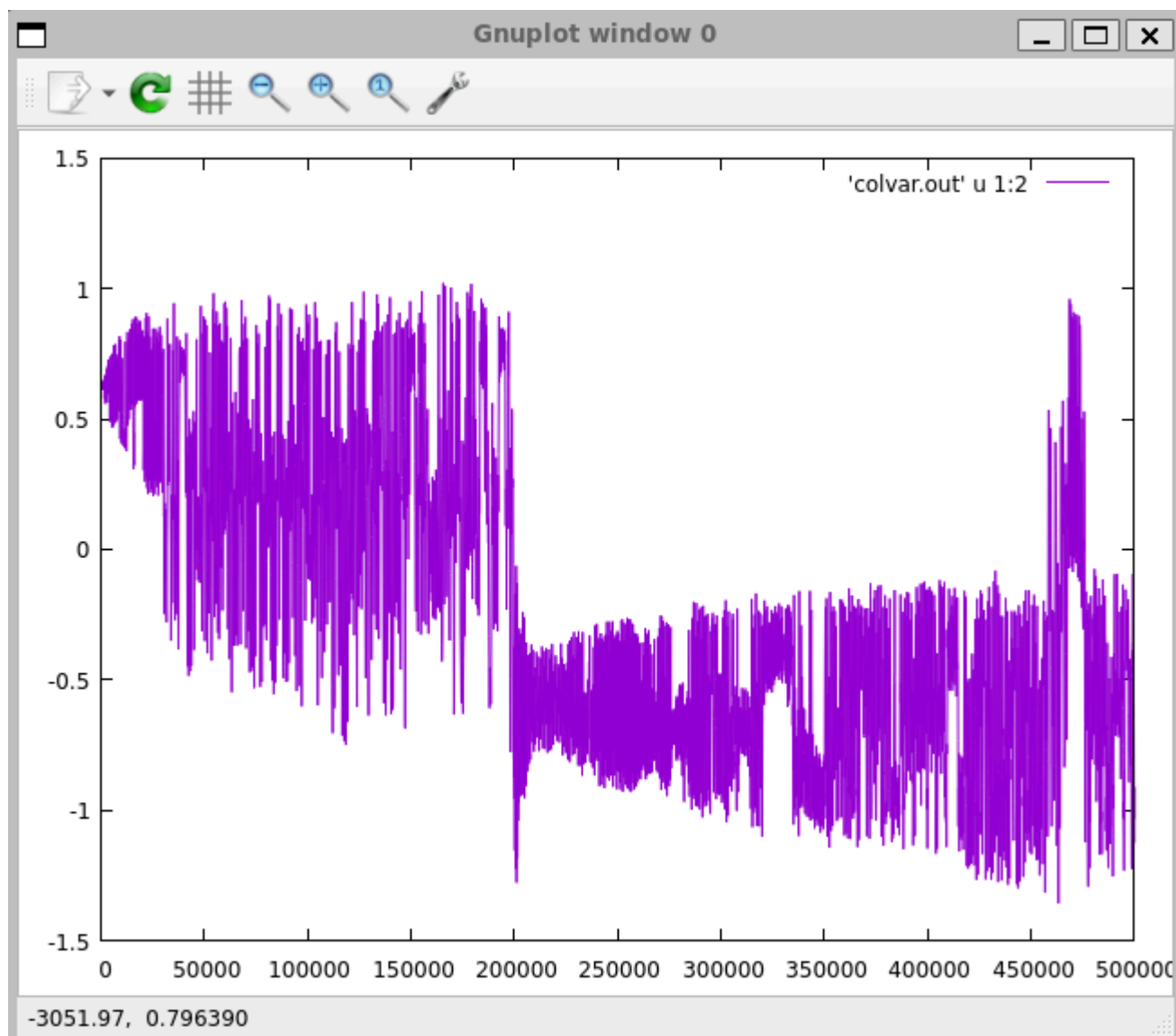
Next, we need to sum those hills to reconstruct the energy landscape.

```
konrad@konrad:~/plumed_masterclass/masterclass-22-09/exercises/1_metadynamics$ plumed sum_hills --hills HILLS --mintozero
PLUMED: PLUMED is starting
PLUMED: Version: 2.9.3 (git: Unknown) compiled on May  8 2025 at 14:37:59
PLUMED: Please cite these papers when using PLUMED [1][2]
PLUMED: For further information see the PLUMED web page at http://www.plumed.org
PLUMED: Root: /home/konrad/opt/plumed-2.9.3/lib/plumed
PLUMED: For installed feature, see /home/konrad/opt/plumed-2.9.3/lib/plumed/src/config/config.txt
PLUMED: Molecular dynamics engine:
PLUMED: Precision of reals: 8
PLUMED: Running over 1 node
PLUMED: Number of threads: 1
PLUMED: Cache line size: 512
PLUMED: Number of atoms: 1
PLUMED: File suffix:
PLUMED: Timestep: 0.000000
PLUMED: KbT has not been set by the MD engine
PLUMED: It should be set by hand where needed
PLUMED: Relevant bibliography:
PLUMED:   [1] The PLUMED consortium, Nat. Methods 16, 670 (2019)
PLUMED:   [2] Tribello, Bonomi, Branduardi, Camilloni, and Bussi, Comput. Phys. Commun. 185, 604 (2014)
PLUMED: Please read and cite where appropriate!
PLUMED: Finished setup
PLUMED: Action FAKE
PLUMED:   with label cv
PLUMED:   added component to this action:  cv.x
PLUMED:   added component to this action:  cv.y
PLUMED: Action FUNCSUMHILLS
PLUMED:   with label @1
PLUMED:   with arguments cv.x cv.y
PLUMED:   Output format is %14.9f
PLUMED:   hillsfile   : HILLS
PLUMED:
PLUMED:   No boundaries defined: need to do a prescreening of hills
PLUMED:   doing serialread
PLUMED:   opening file HILLS
```

**Fig. 21**

*Reconstructing the bias with `plumed sum_hills` where the terminal log shows PLUMED reading the HILLS file, recognising the two collective variables, and serially adding the stored Gaussians. This command produces a grid-based free-energy surface that can be plotted or queried to monitor convergence.*

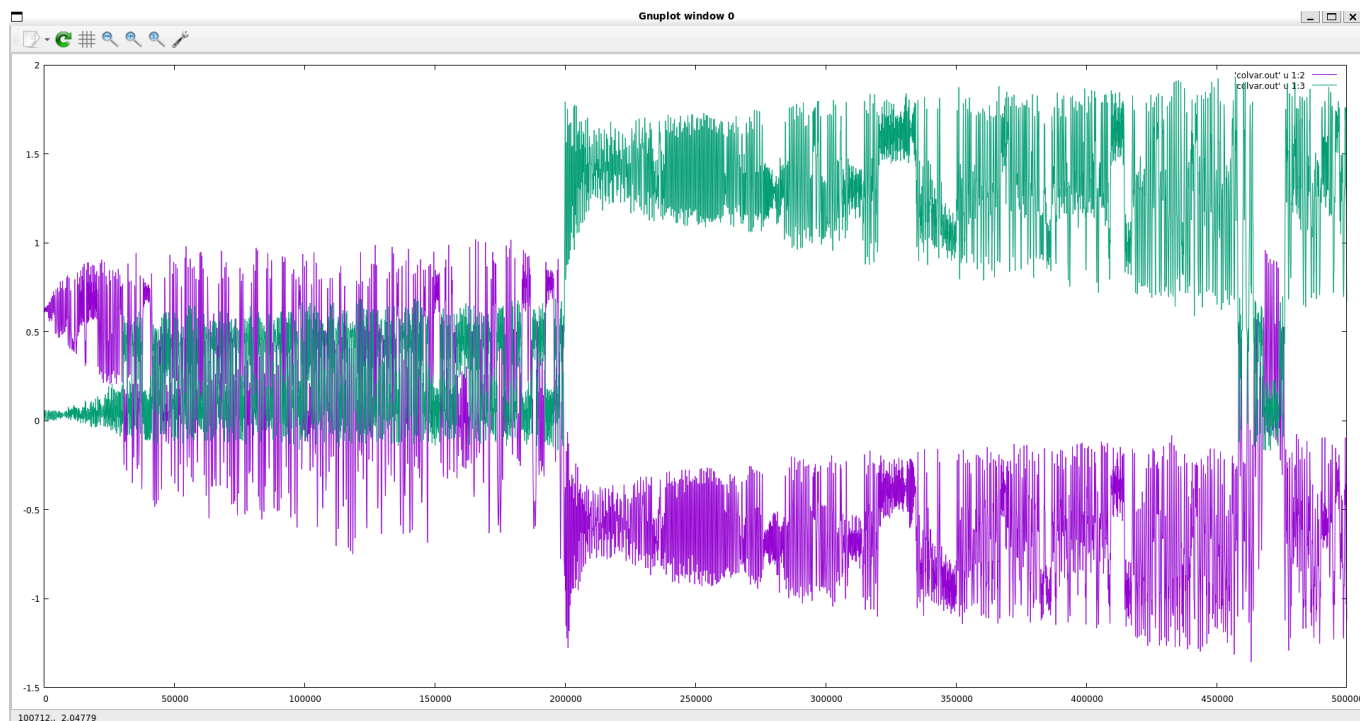
The collective variables currently appear as follows. This is promising, as the Gaussians are steadily pushed outward in circular patterns away from the minima, as shown in the plot.



**Fig. 22**

*Early meta-dynamics push-out ( $Cv.x$  only) where the purple trace starts around 0.6, oscillates narrowly, then - as Gaussians accumulate - undergoes larger excursions first to negative values and finally back toward +0.9. The widening envelope illustrates how the growing bias “forces” the system outward in concentric rings, steadily emptying the initial minimum.*

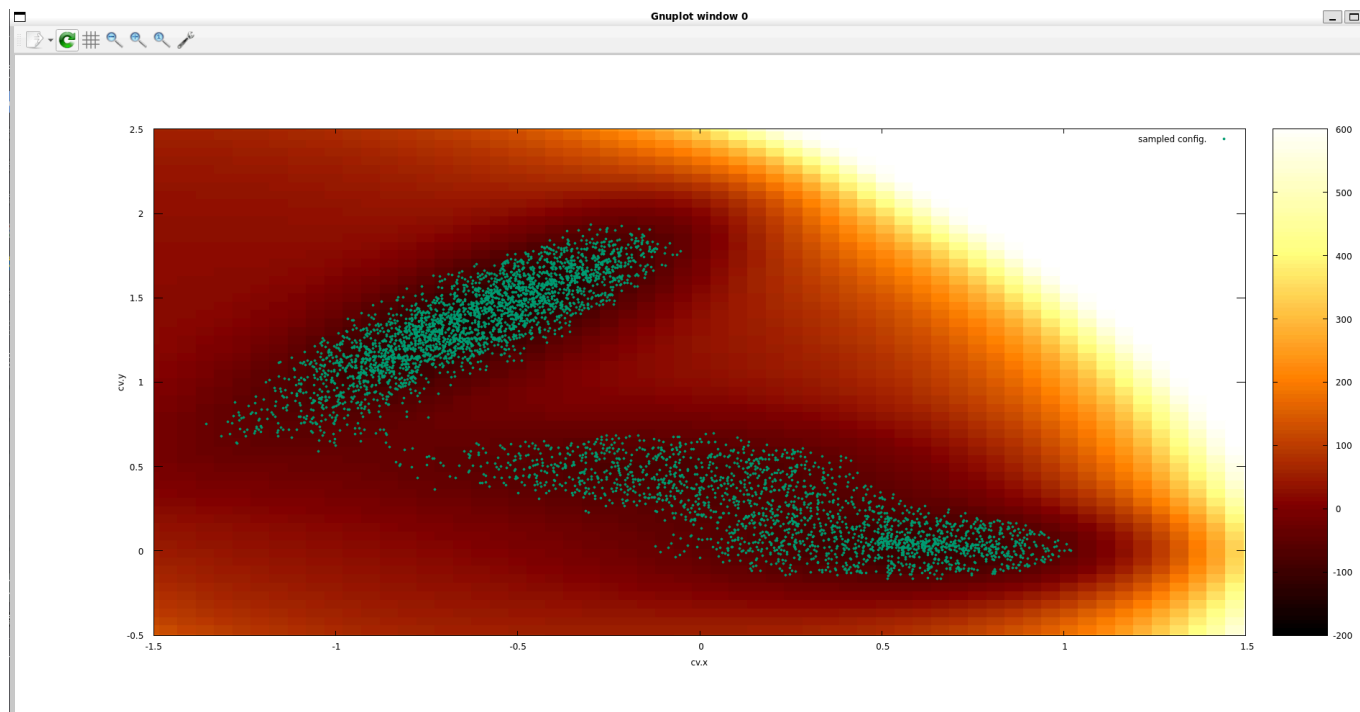
The system escapes into the intermediate region. It spends time in both the initial and intermediate states, and after roughly 200,000 steps, it overcomes the bottleneck and reaches the product state. We observe one crossing and one recrossing.



**Fig. 23**

*Two-coordinate time series reveal a successful barrier crossing. Both  $Cv.x$  (purple) and  $Cv.y$  (green) remain confined until  $\approx 2 \times 10^5$  steps, after which they jump to a new plateau that corresponds to the product basin, linger, and eventually recross. The clear plateaus separated by abrupt transitions demonstrate that the bias has lowered the  $\approx 5k_B T$  ridge enough for repeated hopping.*

And that's the result.



**Fig. 24**

*Final sampling footprint under meta-dynamics. Scatter points now populate two elongated, low-energy troughs - the reactant and product valleys - while high-energy regions remain sparsely*



visited. Compared with the brute-force high-temperature run, meta-dynamics achieves thorough exploration of the kinetically relevant pathways without driving the system into thermodynamically irrelevant, yellow plateau regions.

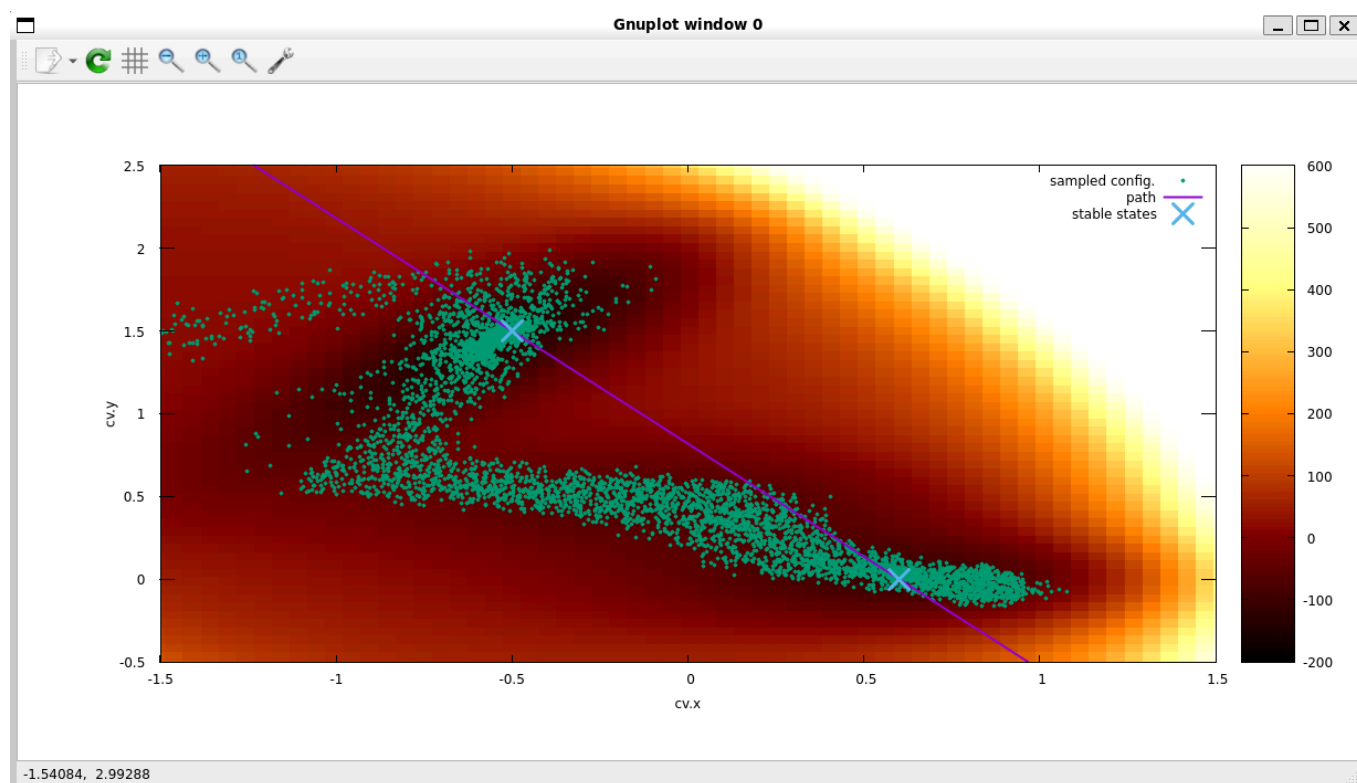
What an efficient landscape search, right? It focuses sampling on the low-energy paths. It still [scales exponentially](#), but with umbrella sampling we'd end up gridding even the regions we're not interested in. Now we need to determine whether we've converged and sampled sufficiently. How can we assess this? One approach is to monitor the relative energy differences over time. Even as we continuously add Gaussians, these differences should stabilize. We can compute a [running average](#) at each point - or simply evaluate the average once the simulation is complete - to check for convergence.

We can employ well-tempered meta-dynamics to converge the Gaussians dynamically - reducing their heights on the fly as the simulation progresses.

## 2.5.4 Well-tempered PMD

(following the Müller-Brown exercise from the PLUMED Masterclass)

When we try to get the first path on the map, we should start with the most naïve guess: draw a straight line through the Müller–Brown landscape by *linearly interpolating* the collective-variable space (**Fig. 25**). Each black dot is a *node*; between them the `genpath` routine simply connects the dots. At this point the path carries **no physics** - it is only a ruler-straight scaffold on which we will build something smarter.

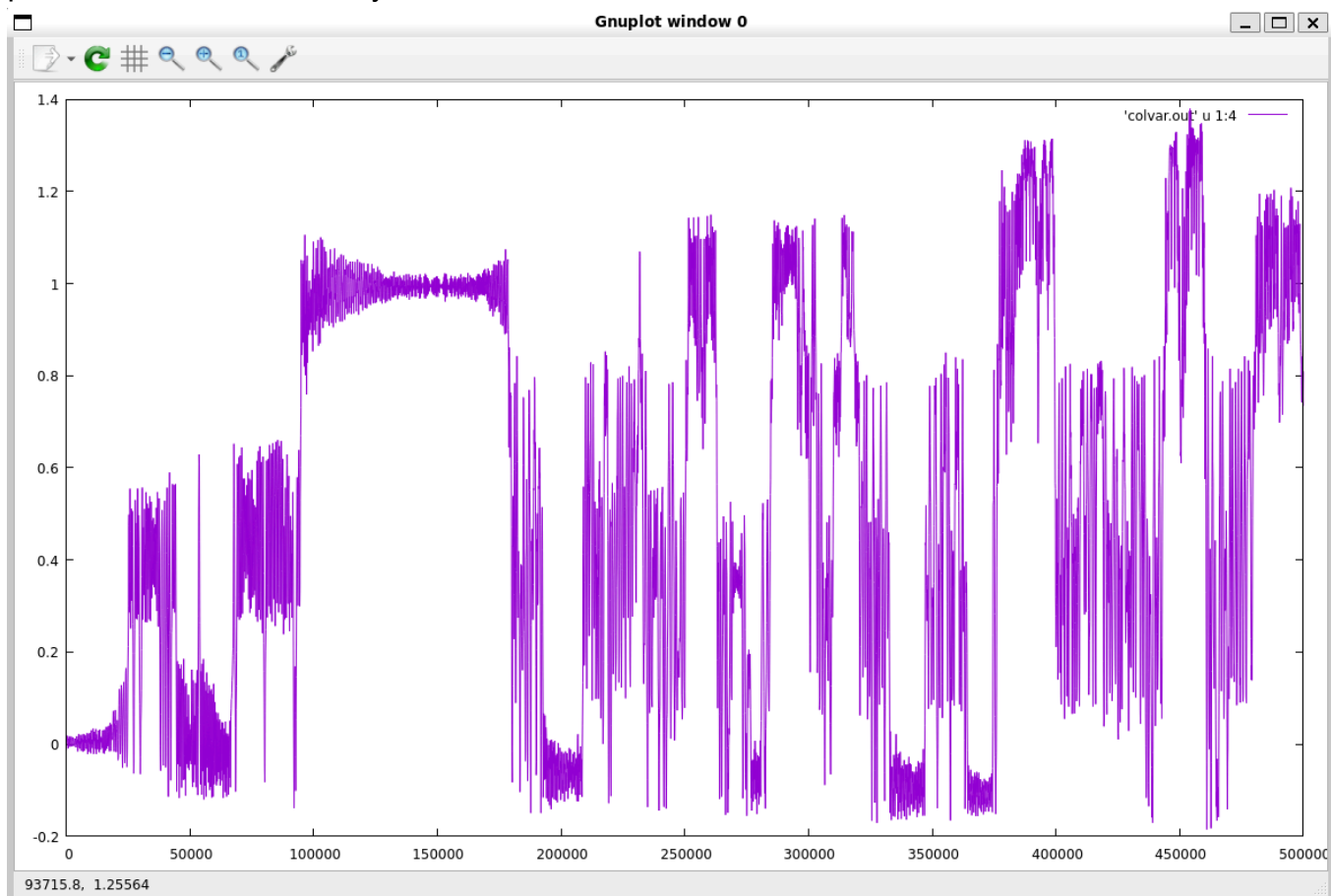


**Fig. 25**



*Naïve straight-line scaffold with the purple poly-line is nothing more than a linear interpolation between the two Müller-Brown minima. It slices straight across high-energy contour rings-geometrically shortest but thermodynamically absurd. The black dots are the equally spaced nodes that will later be allowed to move.*

Measuring how the path relaxes gives us a quick diagnostic is the *path-to-configuration distance* plotted versus simulation time (**Fig. 26**). Every time the purple trace spikes, the current configuration has drifted far away from the straight-line guess; every dip means we have slid back onto it. Large, persistent oscillations here are a red flag - the system is telling us that our path fails to track the valley floor.



**Fig. 26**

*Path-to-configuration distance before restraints. Each spike in the purple trace marks a moment when the walker has drifted far from the straight path; dips show brief returns. The large, erratic excursions warn that the initial path does not follow the energetic valley floor and offers no meaningful guidance to the dynamics.*

Switching on the tube potential keep the walker from drifting *too* far. We wrap the path in a harmonic “tube” and let PLUMED restrain the [orthogonal](#) displacement:

```
tube: RESTRAINT ARG=pcv.z AT=0.0 KAPPA=1000.0
```

Think of it as a soft pipe encasing the [poly-line](#); configurations are allowed to rattle inside, but not to escape. With the tube active, the nodes (blue dots) begin to bow toward the high-density region (**Fig. 28**). Because the path is now *adaptive*, each node shift is weighted by the local population - `z_avg.cv.x` and `z_avg.cv.z` - so that crowded slices pull harder than sparse ones.

The parabola on the distance of the path. It tries to now the nodes are on those positions.

```
konrad@konrad: ~/plumed_n
#! FIELDS Step Time nCV nNodes
500000 500000.000000 2 60
#! FIELDS node cv.x cv.y z_avg_cv.x z_avg_cv.y wsum
0 1.757895 -1.578947 0.000000 0.000000 0.000000
1 1.700000 -1.500000 0.000000 0.000000 0.000000
2 1.642105 -1.421053 0.000000 0.000000 0.000000
3 1.584211 -1.342105 0.000000 0.000000 0.000000
4 1.526316 -1.263158 0.000000 0.000000 0.000000
5 1.468421 -1.184211 0.000000 0.000000 0.000000
6 1.410526 -1.105263 0.000000 0.000000 0.000000
7 1.352632 -1.026316 0.000000 0.000000 0.000000
8 1.294737 -0.947368 0.000000 0.000000 0.000000
9 1.236842 -0.868421 0.000000 0.000000 0.000000
10 1.178947 -0.789474 0.000000 0.000000 0.000000
11 1.121053 -0.710526 0.000000 0.000000 0.000000
12 1.063158 -0.631579 0.000000 0.000000 0.000000
13 1.005263 -0.552632 0.000000 0.000000 0.000000
14 0.947368 -0.473684 0.000000 0.000000 0.000000
15 0.889474 -0.394737 0.000000 0.000000 0.000000
16 0.831579 -0.315789 1.464688 1.074105 12.028932
17 0.773684 -0.236842 403.792375 296.114408 4400.786918
18 0.715789 -0.157895 2743.626892 2011.993054 31337.272211
19 0.657895 -0.078947 1810.606486 1327.778089 26923.488081
20 0.600000 0.000000 585.893247 429.655048 34923.517252
21 0.542105 0.078947 -279.549461 -205.002938 24954.757178
22 0.484211 0.157895 -739.726158 -542.465849 18374.578678
23 0.426316 0.236842 -913.677071 -670.029852 18550.071778
24 0.368421 0.315789 -805.775001 -590.901668 17791.673028
25 0.310526 0.394737 -735.664212 -539.487089 17893.177890
26 0.252632 0.473684 -960.266381 -704.195346 19282.631871
27 0.194737 0.552632 -1329.959731 -975.303803 19324.480175
28 0.136842 0.631579 -1727.968542 -1267.176931 18501.171710
29 0.078947 0.710526 -2011.103615 -1474.809318 16976.052636
30 0.021053 0.789474 -1810.221896 -1327.496057 14099.106433
```

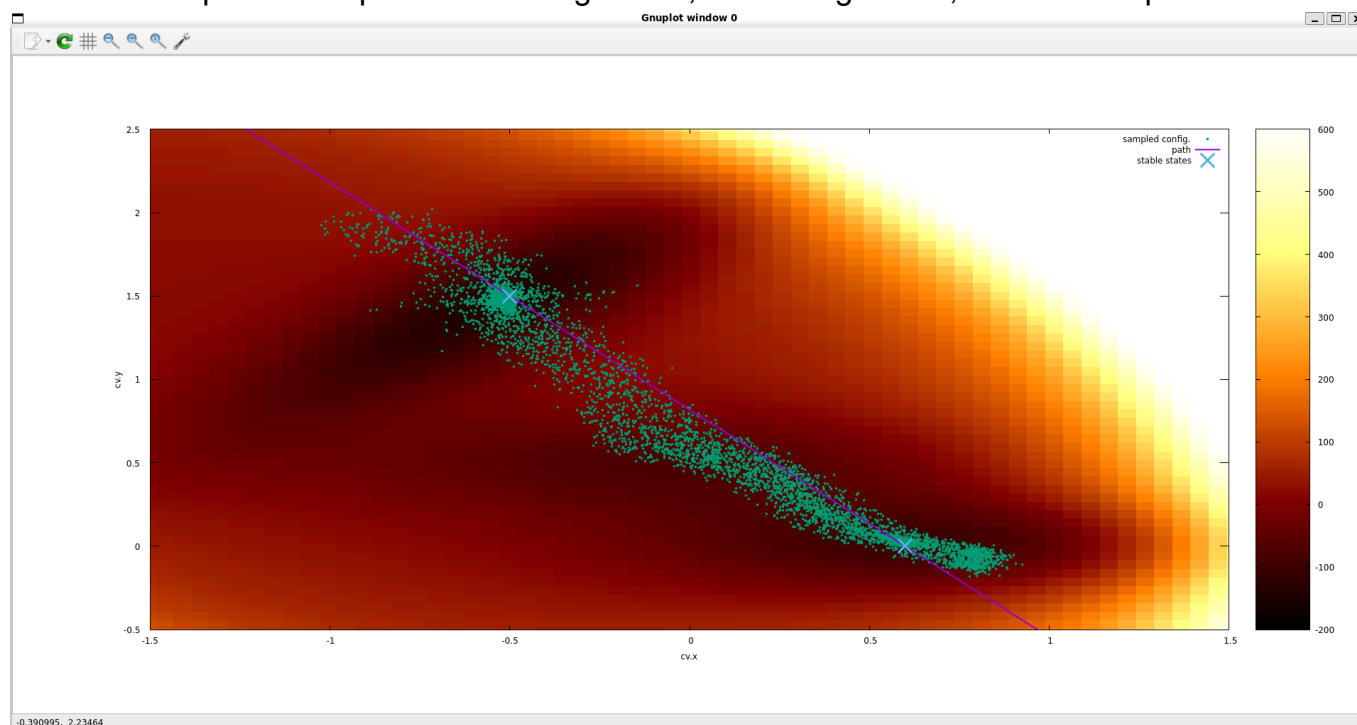
**Fig. 27**

*First adaptive update table. This excerpt of PLUMED's node log lists, for every path node, the weighted averages of the configurations that fell into its  $\sigma$ -slice (`z_avg.cv.x`, `z_avg.cv.y`) and the cumulative weight (`wsum`). Near the ends the weights remain zero - little sampling - whereas mid-path nodes have begun to pull toward populated regions, signalling the onset of density-driven bending.*

In this method, the `z_avg.cv.x` and `z_avg.cv.z` variables are weighted, and during the adaptive-path procedure we update the density and flux using a simple distance/weight

algorithm. To let the endpoints “breathe” and avoid an artificial kink where the free-energy basins meet the flexible path segment, we attach 20 trailing and 20 leading nodes that can slide into the minima, while the central 20 path nodes adjust the transition segment (**Fig. 28**). These extra nodes act like elastic shock-absorbers, anchoring the path smoothly inside each well.

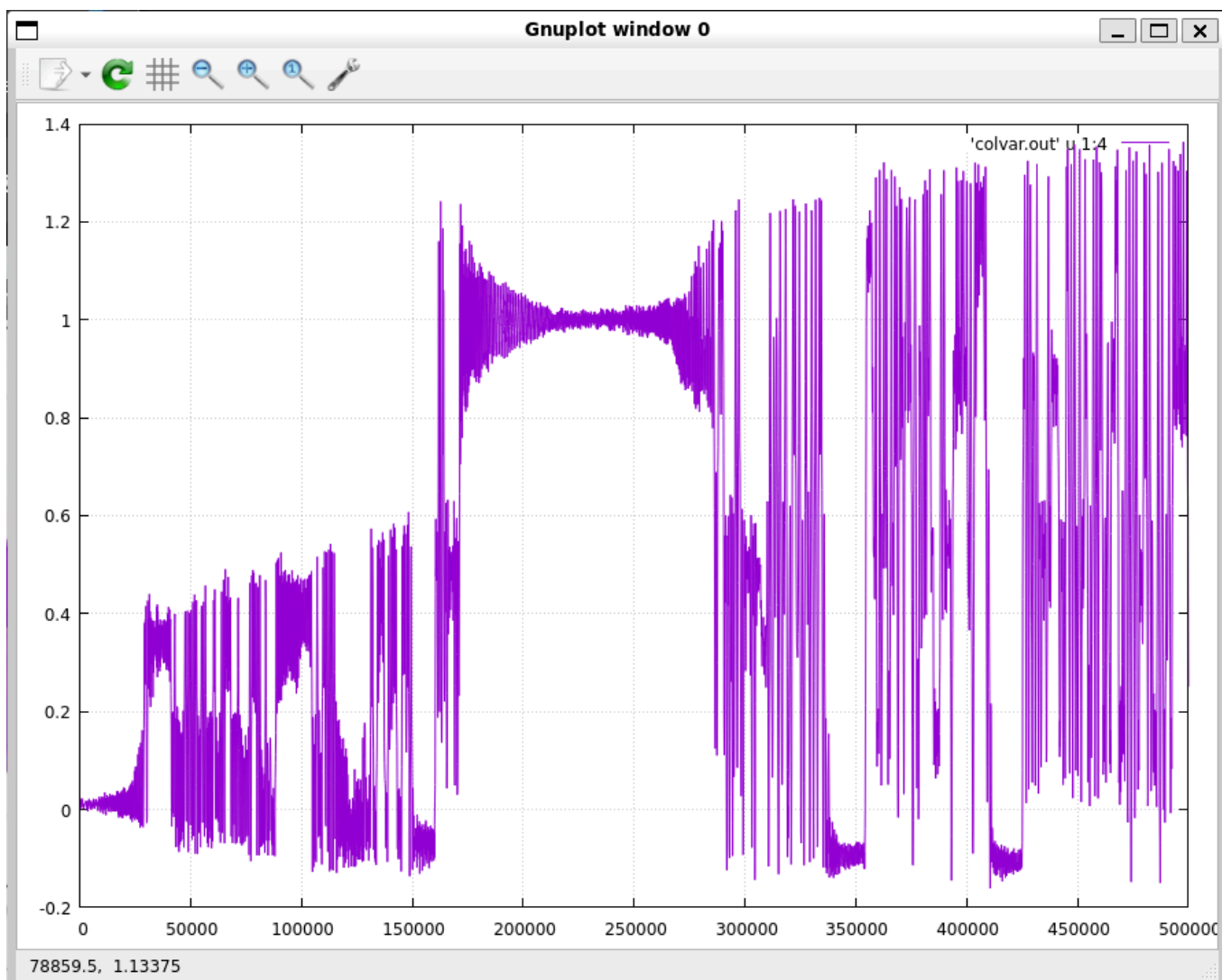
The final setup thus comprises 20 trailing nodes, 20 leading nodes, and 20 core path nodes.



**Fig. 28**

*Path after tube restraint and trailing nodes. With a harmonic tube wrapped around it and 20 flexible trailing nodes at each end, the magenta path bows gently into the low-energy gutter. Turquoise sampling points now cling to a narrow ribbon along the curve, indicating that orthogonal wanderings are tamed while the transition segment smoothly links the basins.*

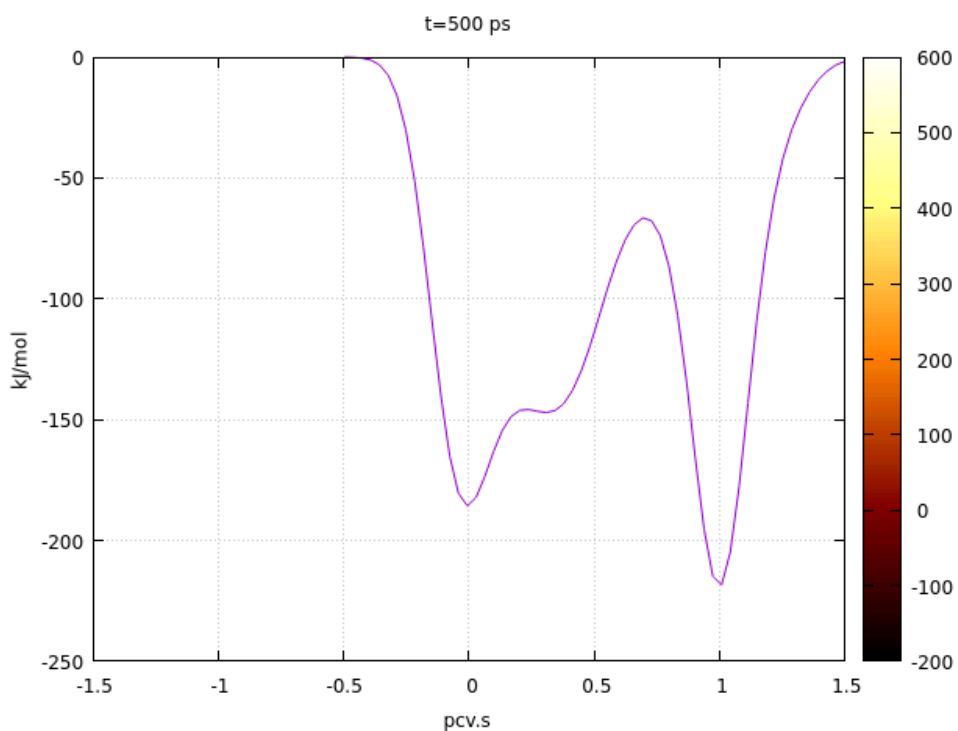
If the path behavior becomes unstable (**Fig. 29**) - “thrashing” erratically between different shapes instead of converging - it usually means the tube restraint was set too wide or the biasing was too aggressive. In such cases, there is effectively a tug-of-war between entropy (the path taking shortcuts) and energy (the path trying to stay in the valley). A chronically jagged  $\sigma(t)$  trace is a clear sign that the Gaussian hills are too large for the landscape or that the path update interval is too short, causing the algorithm to overshoot.



**Fig. 29**

*“Thrashing” diagnostic for an over-aggressive setup. Here the  $\sigma$ -coordinate flips wildly between plateaus and shows high-frequency jitter. Such behaviour means the bias hills or update stride are too large relative to the landscape: the path alternately follows entropic shortcuts and steepest-descent routes without settling, a cue to relax the parameters.*

Finally, once the path has stabilized, we can take a slice through the free energy surface along the converged path. In practice, we project the walker’s sampled density onto the  $(CV.x, CV.z)$  plane and directly visualize the valley walls carved out by the adaptive path (**Fig. 30**).

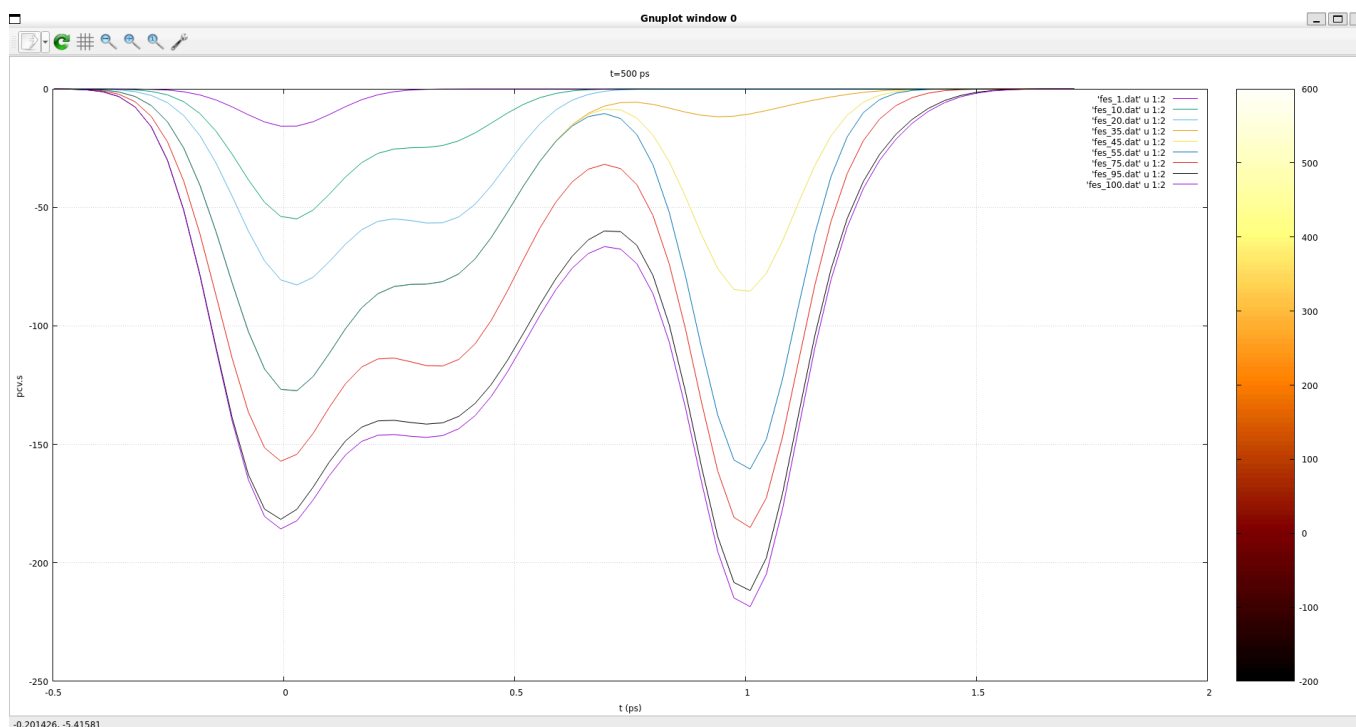


**Fig. 30**

*One-dimensional free-energy slice along the converged path ( $t = 500$  ps). Projected onto  $\sigma$ , the profile now exhibits two  $\approx -200 \frac{\text{kJ}}{\text{mol}}$  wells separated by a  $70 - 80 \frac{\text{kJ}}{\text{mol}}$  barrier at  $\text{sigma} \approx 0.5$ . The deep, smooth trough confirms that the adaptive path tracks the true valley floor rather than cresting ridges.*

Based on this, we can integrate to calculate the area under the valleys and quantify the difference - thanks to the trailing nodes.

And how many Gaussians are enough? In well-tempered mode the algorithm needed only **60 hills** to fill the basin, cross the saddle, *and* recross back to the reactant state (**Fig. 31**). The running average of the bias height plateaus after roughly 40 hills, a convenient heuristic that the free-energy difference has converged within  $\approx 1 \frac{\text{kJ}}{\text{mol}}$ . If the running average still slopes after three or four bias half-lives, either widen `SIGMA` or lower `HEIGHT` - the simulation is painting the landscape with a brush that is too fine.



**Fig. 31**

*Convergence of the well-tempered bias. Successive reconstructs of the PMD bias after 10, 20, 60 Gaussians collapse onto one another beyond  $\approx 40$  hills, showing that the free-energy difference has stabilised within  $\approx 1 \frac{\text{kJ}}{\text{mol}}$ . The plateau indicates that the chosen  $\text{SIGMA}$ ,  $\text{HEIGHT}$ , and  $\text{PACE}$  fill the basins efficiently without over-smoothing, completing the landscape in barely sixty deposits.*

Based on that, we can compute a running average to assess convergence. With the path now collapsed onto a single progress coordinate  $\sigma$ , the *one* Gaussian width you specify acquires a very tangible role: it is literally the half-width of each hill **along that axis**. A sensible rule of thumb is to match it to the natural jiggle of  $\sigma$  during the interval  $\tau$  between hill depositions *and* to the spacing between adjacent transition nodes ( $\approx 0.5$  here). A wider width blurs neighbouring basins into one another, while a narrower one leaves gaps, so the summed bias becomes ragged and the free-energy estimate noisy.

The purple  $\sigma$ -trace in (**Fig. 26**) now doubles as a seismograph for the *effective* diffusion landscape. Long horizontal plateaus at  $\sigma \approx 0$  and  $\sigma \approx 1$  mark the two Müller-Brown minima; once the bias nudges the walker past  $\approx 0.2$  or  $0.8$  it shoots almost ballistically across the mid-section, because the underlying  $2D$  valley is broad and shallow there. In other words, the apparent diffusion coefficient varies with position: it is largest where the valley is wide (centre) and smallest where it narrows into the wells (ends).

Adding the tube restraint sharpens this picture. Even a modest spring constant squeezes the turquoise cloud that surrounds the magenta path (**Fig. 27**) into a narrow ribbon, forcing configurations to slide *along* the curve instead of short-cutting through CV space. The transverse spread collapses,  $\sigma$  diffusion slows just a touch, and - critically - the bias converges

faster because no Gaussians are wasted off the path. Crank the force constant much beyond  $100 - 200 \frac{\text{kJ}}{\text{mol}}$ , however, and you over-quench orthogonal fluctuations; recrossing stall and the whole algorithm becomes sluggish.

With 20 transition nodes bracketed by 20 trailing nodes on each side, the fixed-path meta-dynamics run produces a free-energy profile that is both smooth and fully converged: two wells at  $\sigma \approx 0$  and 1 with depths of about  $-200 \frac{\text{kJ}}{\text{mol}}$ , separated by a  $70 - 80 \frac{\text{kJ}}{\text{mol}}$  barrier centred at  $\sigma \approx 0.5$ . Because the geometry of the path itself is locked, that slice is the exact line integral you would obtain by integrating the full Müller-Brown surface along the magenta poly-line-confirmation that the one-dimensional bias has recovered the thermodynamics of the original 2D system without ever leaving the adaptive loop.

## 2.5.5 Adaptive PMD

The moment we let the nodes *move* we step into the world of **adaptive PMD (Fig. 32)**.

In this variant the curve is no longer a rigid ruler but a living, breathing object that keeps a time-weighted diary of every configuration it encounters. The single knob that controls that diary is `HALFLIFE` - the number of MD steps after which the influence of a data point decays to 50%. Put differently, `HALFLIFE` sets how far into the past the path can “remember”.

A *short* memory ( $\approx 10^3$  steps) makes the path almost mercurial. It discards mistakes, chases density, and snaps into the first low-energy corridor the bias uncovers. That is perfect for the exploratory phase when you just want the curve to abandon a bad straight-line guess and latch onto any plausible transition tube.

Dial `HALFLIFE` up to  $\geq 10^4$  steps and the character flips. Now each node behaves like a cautious statistician, averaging over many recrossings before it agrees to shift. The jitter from individual Gaussians is ironed out, and what survives is the *mean* transition pathway weighted by true flux, not by fleeting noise.

The sweet spot is to **combine both moods in a two-stage workflow**:

### 1. Fast scouting.

Start with generous Gaussians, a permissive or even absent tube restraint, and `HALFLIFE`  $\approx 1000$ . In a few hundred thousand steps the bias will volley the walker back and forth until both basins are well-trodden and the path has bent into the right valley.

### 2. Precision carving.

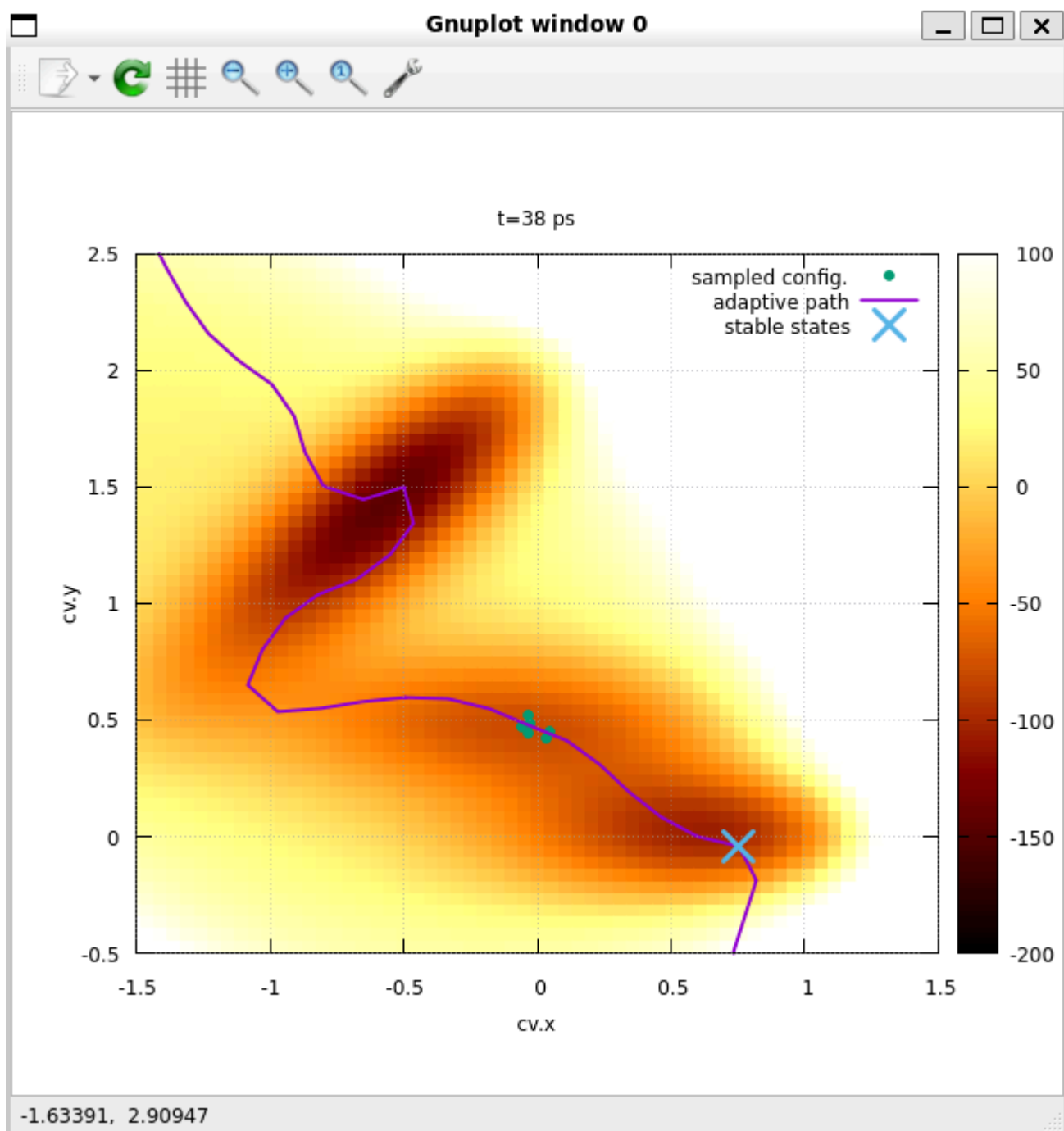
Freeze the simulation, read the *last* set of nodes with `INFILE`, shrink `SIGMA`, tighten the tube spring, and raise `HALFLIFE` two orders of magnitude. From this point on the nodes drift only when multiple independent excursions agree, while well-tempered meta-dynamics quietly files the residual bumps in the one-dimensional free-energy profile to below  $\approx kT$ .

### 2.5.5.1 Dealing with multiple product basins

Real chemical reactions rarely follow a simple, linear path from state A  $\rightarrow$  B. Often, a reaction can end up in multiple distinct final states (for example, two crystallographically different forms of product B). If an adaptive path simulation begins with only one initial guess, it typically diverges early, settling into whichever final state it encounters first. To avoid this limitation, two straightforward methods can be used. First, you can introduce a second reaction coordinate, for instance by running two parallel paths starting from the same reactant but ending in separate product states, or by using an additional coordinate that identifies which final state you're approaching. Alternatively, you can initiate several independent simulations (walkers) on separate GPUs, each slightly biased toward different product states, and afterward merge their resulting data once both have reached convergence.

Either strategy keeps PMD agnostic and lets the free energy, not a lucky first hop, decide which channel prevails.





**Fig. 32**

*First 38 ps of adaptive PMD. The magenta poly-line - now free to bend - has already abandoned the straight scaffold and is curling into the dark, low-energy gutter that arcs between the reactant well (large blue X at  $Cv.x \approx 0.6$ ,  $Cv.y \approx 0$ ) and the upper-left basin. Because HALFLIFE is still short, each newly visited configuration (turquoise dot) pulls strongly on the nearest node, giving the curve its kinked, exploratory shape. High-energy ridges (yellow) are skirted, while the path hugs the reddish trough, indicating that the algorithm has begun to recognize and follow the true transition tube within a few tens of picoseconds.*

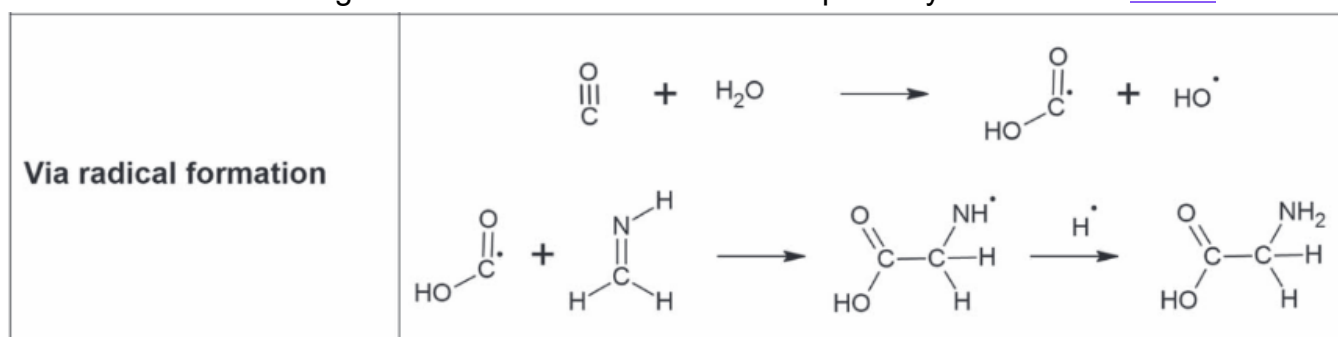
## 2.6 Multiple-Walker Path-Meta-dynamics

That topic isn't a concern for me right now, but I may explore it in the future.

These were all the exercises, accompanied by my explanations and insights from Professor Bernd Ensing, who - with his team - developed these methods and put them into practice.

### 3. Case Study: Glycine-1-13C

"With the Müller–Brown example completed, we are ready to apply the adaptive path approach to a real chemical problem: the stepwise formation of glycine in water, as reported by Francisco Carrascoza *et al* 2023. In a forthcoming post, I plan to import the reactant and product snapshots from that study, sketch an initial multi-CV path, and let our two-stage adaptive protocol reveal the molecular choreography—how formaldehyde, carbon monoxide, and a proton stitch together into the simplest amino acid (glycine). I will also outline a three-module simulation scheme designed to mirror the radical reaction pathway illustrated in [article](#)



**Fig. 33**

*Reaction pathway for glycine formation (from Francisco Carrascoza et al 2023). The reaction starts from carbon monoxide and formaldimine in water and proceeds through several intermediates (not detailed here) to form glycine. (This figure illustrates the kind of multi-step reaction the adaptive path approach would need to capture.)*

## 4. Appendix

Topics encountered while studying Prof. Ensing's lectures / exercises

### 4.1 Path Collective Variables (Path-CV) & Reaction Coordinates

#### 1. Foundations

- Cluster model; defining path collective variables
- Defining reaction coordinates; different reactant states

- Two stable states  $\rightarrow$  path A  $\rightarrow$  B  $\rightarrow$  average transition path; different channels
- From A to B in free-energy space: *can I say where I am between A & B?* (projection)

## 2. Algorithmic workflow

- Start from a *guess path*  $\rightarrow$  bias dynamics along the path  $\rightarrow$  move nodes to mean density (transition flux)  $\rightarrow$  maintain equidistant nodes
- Expressed in structures: compute RMSD for reactant structures, optimise those structures to refine the path
- Method with torsion angles  $\rightarrow$  density sampled  $\rightarrow$  free-energy samples  $\rightarrow$  *bottleneck somewhere here*

## 3. Path-CV mathematics

- $s$  = position on the path (Euclidean projection)
- $z$  = distance from the path
- Distance to mean density; two nodes + trailing nodes

## 4. Simulation strategies

- Single simulation  $\rightarrow$  reaction path & free-energy profile
- Step-wise path
- 8 simultaneous simulations, each  $\approx 10$  ns
- Torsion-angle rotation has little barrier  $\rightarrow$  zipper-like mechanism
- Leads to 3 factorial pathways
- Sub-linear convergence
- Repellers simulation: move paths in different directions

## 5. Path-CV project tools

- *Path-map* creation
- Initialise Path-CV: equilibrate, compute CV-averages at endpoints, build initial path (linear interpolation or previous trajectory)
- Choose biasing method (steered MD); enable multiple walker; monitor path & free-energy profile evolution; tube potential; scale Gaussians by hand

## 6. Specialised methods

- Path meta-dynamics ( $\sigma$  function; more CVs)
- Single-walker PMD, multiple-walker PMD, multi-PMD
- MuWaMuPaMetaDyn – *Multiple-Walker Multiple-Path Meta-dynamics*

# 4.2 Enhanced-Sampling & Rare-Event Techniques

## 1. General concept – enhanced sampling

- “All possible bonds are produced and broken”  $\rightarrow$  explore full free-energy landscape

## 2. Temperature-based acceleration

- Raising temperature to boost barrier crossing
- Parallel tempering / replica exchange
- Temperature-accelerated dynamics (mix low-T & high-T)

### 3. Meta-dynamics family

- No meta-dynamics ▶ plain meta-dynamics ▶ well-tempered meta-dynamics
- Assessing meta-dynamics accuracy: decrease hills upon every recrossing for convergence
- Flood the landscape with hills; localise minimum FE path, then switch to 1 D to converge further
- “Six collective variables is *not* doable” ▶ *biased* six CVs (all-at-once vs independent)
- 8-dimensional landscape - not just 2-D projections
- Lowest free-energy pathway; find hidden profile (reaction coordinates)

### 4. Rare-event frameworks

- Transition-interface sampling
- Rare-event methods (general)
- Committed probability: where is the TS?
- Halfway between A and B → iso-committed surface → hyperplane 0 → 0.5 → 1.0
- Average transition pathway (in CV space); symmetric channel; *painfully expensive*

### 5. Accuracy & diagnostics

- How to calculate the projection?
- What if underlying landscape unknown → how to set  $\sigma$ ?
- Defining diffusivity in meta-dynamics

## 4.3 Reaction-Path Sampling & Free-Energy Methods

### 1. Finding / sampling a reaction path; SN2 reaction example

### 2. Thermodynamic integration

### 3. Flooding vs path-based approaches

### 4. Reaction channels & paths: good reaction coordinate often hard to define

### 5. Rare events in real life; time-scales:

- Proton transfer, enzymatic reactions, organic reactions, phase transitions, protein folding, polymer relaxation
- Electronic excitation; intra-molecular vibrations; H<sub>2</sub>O rotation in liquid water

### 6. Ab-Initio, Force-Fields & Electronic Processes

- DFT / defining force fields
- Optimisation strategies in *ab initio* methods
- Electron-transfer / proton-transfer cascades → many CVs
- Force-field simulations using Path-CV

### 7. Collective-Variable Selection & Machine Learning

- “As inputs you need good collective variables.”
- ML for CV discovery:
  - Find CVs with ML → test with Path-CV → sample → better data

- **Typical CVs: distances within molecule; torsion angles (periodic, high-energy crossings)**
- **Ramachandran plot for alanine peptide in vacuum (torsion angles periodic → high-energy crossings)**

## 8. Simulation Paradigms & Model Hierarchies

- **Quantum methods**
- **MD / MC**
- **Coarse-grained models**
- **Fluid-mechanics & stochastic networks**

## 9. Practical Tips

- **Transition-interface sampling**
- **Mimic environment for single particle (Langevin bath)**
  - Two-particle distance;  $\kappa$  = force constant
- **Lennard-Jones units (restrain everything); defining temperature in energy units**
- **PLUMED-Nest example: polyproline (Alberto) – ID of the egg**
- **Using Gnuplot on Ubuntu – full command syntax for plotting**

1. **Molecular dynamics (MD)** is a computational simulation technique that models the physical movements of atoms and molecules over time using Newtonian mechanics. It allows researchers to study the structural, thermodynamic, and dynamic properties of molecular systems at the atomic level. ↩
2. **PLUMED** is an open-source plugin that can be interfaced with many molecular dynamics engines (e.g., GROMACS, AMBER, LAMMPS). It provides advanced methods for enhanced sampling, free-energy calculations, and analysis of molecular simulations. ↩
3. **Masterclasses** are intensive workshops or seminars led by experts, designed to deliver in-depth training on specialized topics; here, **PLUMED masterclasses** focus on advanced sampling methods, free-energy calculations and analysis techniques in molecular dynamics. ↩
4. **Meta-dynamics** is an enhanced sampling method in molecular dynamics that adds a history-dependent bias to overcome energy barriers, enabling exploration of rare events and accurate free-energy landscapes. ↩
5. **CPMD** (Car–Parrinello Molecular Dynamics) is an *ab initio* molecular dynamics method that integrates electronic structure calculations (via density functional theory) “on the fly” with nuclear motion, allowing for reactive simulations without predefined force fields. ↩
6. **Free energy simulations** are computational techniques (e.g., umbrella sampling, meta-dynamics, thermodynamic integration) that estimate the free energy differences between molecular states or along reaction coordinates, providing quantitative insights into reaction mechanisms and stabilities. ↩
7. **Reaction pathways** are sequences of molecular configurations that connect reactants to products through a potential energy surface, often identified by minimum-energy paths or transition states, and used to understand the mechanism and kinetics of chemical reactions. ↩

8. **Path-meta-dynamics (PMD)** is an enhanced-sampling method that applies meta-dynamics bias along a predefined path, allowing the system to explore and optimize transition pathways between states. ↩
9. **Minimum free-energy path (MFEP)** is the trajectory on the multidimensional free-energy surface that requires the least work to move from reactants to products, often passing through saddle points (transition states). ↩
10. **MD engine** (molecular dynamics engine) is a software package that numerically integrates Newton's equations of motion for atoms and molecules - handling force evaluations, neighbor lists, thermostats/barostats, and trajectories - to simulate the time evolution of molecular systems. ↩
11. **Free energy surface (FES)** is a multidimensional landscape defined over collective variables that assigns to each point  $z$  the free energy  $F(z) = -k_B T \ln P(z)$ , where  $P(z)$  is the probability of observing the system at  $z$ . It summarizes the thermodynamic stability and possible transition pathways between states. ↩
12. **Canonical equilibrium distribution** describes the probability density  $P(q) = \frac{1}{Z} e^{-\beta V(q)}$ , where  $V(q)$  is the potential energy,  $\beta = 1/(k_B T)$ ,  $k_B$  is Boltzmann's constant,  $T$  the temperature, and  $Z$  the partition function ensuring normalization. ↩
13. **Standard smooth switching function** is a continuous, differentiable function  

$$s(r) = \frac{1 - \left(\frac{r - r_0}{r_c - r_0}\right)^n}{1 - \left(\frac{r - r_0}{r_c - r_0}\right)^m}$$
used to count contacts by smoothly turning "on" when interatomic distance  $r$  is below cutoff  $r_0$  and "off" above it, with exponents  $n < m$  controlling the sharpness of the switch. ↩
14. **Gaussian kernels** in meta-dynamics are bias potentials of the form  $G(z; t') = w \exp\left(-\frac{\|z - z(t')\|^2}{2\sigma^2}\right)$ , where  $w$  is the height,  $\sigma$  the width of each Gaussian deposited at past CV value  $z(t')$ , ensuring smooth, localized biasing of the free-energy surface. ↩
15. **Jerk** in the context of molecular dynamics refers to the rate of change of acceleration (third derivative of position with respect to time), quantifying sudden changes in forces that can introduce numerical instability. ↩
16. **Well-tempered meta-dynamics** is a variant in which the deposited bias height at time ( $t$ ) scales as  $H_{\text{eff}}(t) = H \exp\left(-\frac{V(s, t)}{k_B \Delta T}\right)$  where  $V(s, t)$  is the bias already added at collective variable  $s$ ,  $\Delta T$  is a "bias-temperature" parameter, and  $k_B$  is Boltzmann's constant - ensuring gradual convergence of the free-energy estimate. ↩
17. **Multiple-walker meta-dynamics** is an approach where several independent simulations ("walkers") run in parallel, each depositing bias on a shared global history-dependent potential. This coordination speeds up exploration of the free-energy surface by combining sampling efforts across all walkers. ↩
18. **Nudged elastic band** is a method for finding saddle points and minimum energy paths between known reactants and products. The method works by optimizing a number of intermediate images along the reaction path. Each image finds the lowest energy possible while maintaining equal spacing to neighbouring images. This constrained optimization is done by adding spring forces along the band between images and by projecting out the component of the force due to the potential perpendicular to the band. ↩
19. **Committor-based sampling** relies on the committor function  $p_B(q)$ , the probability that a configuration  $q$  will reach product state  $B$  before reactant state  $A$ . By launching multiple short unbiased MD trajectories from points along a trial path and estimating their committor values, one identifies configurations with  $p_B(q) \approx 0.5$  (transition states) to refine the reaction pathway. ↩
20. **High transition flux** refers to regions along the progress coordinate where the rate (or probability per unit time) of reactive trajectories crossing a given hypersurface is maximal, indicating the most probable transition pathways between metastable states. ↩

21. **Grid in CV-space** refers to a discretized mesh covering the high-dimensional space of collective variables (CVs), where each grid point corresponds to a specific set of CV values. Sampling or biasing on this full grid becomes infeasible as the number of CVs grows, due to the “curse of dimensionality.” ↩
22. **Iso-committor planes** are hypersurfaces in configuration space where the committor function  $p_B(q)$  (the probability of reaching product before reactant) is constant. They are orthogonal to the ideal reaction coordinate and partition configurations by equal “commitment” to product or reactant basins. ↩
23. **Umbrella sampling** is a biased-simulation technique where the system is restrained to sample overlapping “windows” along a chosen reaction coordinate by applying harmonic potentials (umbrellas). By combining the histograms from each window via the Weighted Histogram Analysis Method (WHAM) or similar, one reconstructs the unbiased free-energy profile. ↩
24. **Steered MD (SMD)** is a method where an external time-dependent bias (e.g., a harmonic spring moving at constant velocity) is applied to selected atoms or collective variables, “pulling” the system along a predetermined path; analysis of the work performed yields insights into free-energy barriers and mechanical response. ↩
25. **Müller–Brown potential** is a two-dimensional analytic model PES defined as a sum of Gaussian wells. It features multiple minima and saddle points, making it a standard testbed for exploring and validating enhanced-sampling and path-finding algorithms. ↩
26. **Dissipative loss** refers to the loss of mechanical energy (kinetic and potential) due to frictional or viscous forces in the system, which is converted into heat and removed from the sampled degrees of freedom, thus damping the motion. ↩
27. **Multi-kilojoule barriers** are energy barriers on the order of tens of kilojoules per mole (e.g., 15–30 kJ/mol). For comparison, at room temperature  $k_B T \approx 2.5$  kJ/mol, so a 20 kJ/mol barrier is roughly eight times the thermal energy, making spontaneous crossing extremely rare without enhanced sampling. ↩
28. **Distorting the physics** refers to altering the system’s natural ensemble and dynamics - elevated temperature or reduced friction can change reaction pathways, violate detailed balance, and sample configurations that are not representative of the target thermodynamic conditions, leading to inaccurate kinetics and equilibrium properties. ↩