WHAM Convergence Module

Cameron Kopp

December 2022

1 Data Generation

The unbiased probability distribution used as an example here admittedly seems arbitrary. It was selected due to its relative smoothness to test the following BFGS algorithm, which struggles when the unbiased distribution has values of zeros or extrema at the boundaries. I am completely sure whether this behavior is due to BFGS algorithm I have implemented or whether it is something intrinsic in the Weighted History Analysis Method.

1.1 Method: SampleConstructor.xp1nverseSinSq(x)

The goal here is generate random data from the probability distribution function (FIGURE 2 (a))

$$\rho\left(x\right)=\frac{1}{\left(1+\left(1-x\right)^{2}\right)}\frac{\sin^{2}\left(2x\right)+2}{3} \tag{1}$$

for a set of biased simulations $\{i\}\,(i=0,1,2,...,S)$, where

$$\rho_{i}\left(x\right)=\sqrt{\frac{k}{2\pi}}*e^{-\frac{k}{2}\left(x-\mu_{i}\right)^{2}}*\rho\left(x\right) \tag{2}$$

and

$$\mu_{i}=x_{min}+i\left(\frac{x_{max}-x_{min}}{S}\right) \tag{3}$$

with C as a normalization constant. If we define the biasing potential for a simulation i, which will be centered around μ_i (equation (3) above), as

$$V_{i}'\left(x\right)=\sqrt{\frac{k}{2\pi}}*e^{-\frac{k}{2}\left(x-\mu_{i}\right)^{2}} \tag{4}$$

then we see that:

$$\rho_{i}\left(x\right)=C\rho\left(x\right)*V_{i}'\left(x\right) \tag{5}$$

Figure 1 illustrates the relation between equations (1), (4) and (5) for different values of μ_i

2 BFGS Algorithm with Armijo Line Search

Here we are trying to minimize a function $A\left(\vec{f}\right)$ by altering the values of the indices $\{f_i\}\,(i = 0, ..., S)$, of the vector $\vec{f}$. In the context of the wham equations, we are minimizing equation(19) of the Hummer and Zhu paper by altering the vector of the logarithms of the normalization constants for a simulation i. We will use the index $l = 0, 1, 2, ...$ to label the iteration number of the BFGS algorithm, such that $\vec{f}^0$ will represent the input estimate, $l = 0$, and $\vec{f}^l$ will represent an arbitrary estimate at iteration l. We will use $A\left(\vec{f}^l\right)$ to represent equation (6) evaluated for the set of inverse partition functions $\vec{f}^l$, with the other variable (the $\vec{f}^J$ or $\vec{f}^K$ that we are not minimizing over) held constant. H^l defines an $(S \times S)$ matrix for iteration l of the BFGS algorithm, which is the inverse Hessian matrix and will be used to determine the search direction for iteration l where $H^{l=0} = I^S$, the $S \times S$ identity matrix. The following vector also has to be defined:

$$\vec{g}^l\left(\vec{f}^l\right) = \left[g_0\left(\vec{f}\right), g_1\left(\vec{f}\right), ..., g_S\left(\vec{f}\right)\right]^T, \quad g_i\left(\vec{f}^l\right) = \frac{\partial A}{\partial f_i}, \quad i = 0, 1, ..., S \tag{6}$$

where $\frac{\partial A}{\partial f}$ is defined above in equation (7) or (8). The following steps outline the BFGS algorithm for a given H^0, $\vec{f}^0$: for $l = 0, 1, ...$

(1)compute the quasi-Newtonian search direction

(2)determine the step size α^l and calculate $\vec{f}^{l+1}$

(3)compute H^{l+1}

2.1 Search Direction (Step 1)

The fist step in the BFGS algorithm is computing the search direction $\vec{p}$, which sets the direction that we will look in for the new $\vec{f}^{l+1}$. The BFGS calculates the search direction with the equation

$$\vec{p} = -H^l\vec{g}^l. \tag{7}$$

Intuitively, it should make sense that $\vec{p}$ is proportional to the gradient $\vec{g}^l$. If we were using Newton's method, our search direction would be $\vec{p} = \vec{g}\left(\frac{\partial^2 A}{\partial \vec{f}^2}\right)^{-1}$. Rather than computing the second derivative with each step, the BFGS algorithm approximates it with H^l.

2.2 Armijo Backtracking Line Search and $\vec{f}^{l+1}$ (Step 2)

The Armijo backtracking line search algorithm requires the inputs α^0, and $\tau, \beta \in (0,1)$, for which I've been using $\alpha^0 = 2$, $\tau = 0.5$, and $\beta = 0.1$. The goal is to compute a $\Delta\vec{f}$ such that $\vec{f}^{l+1} = \vec{f}^{l} + \Delta\vec{f}$, with $\Delta\vec{f} = \alpha^m \vec{p}$. This method imposes the following condition

$$A\left(\vec{f}^{l} + \alpha^m \vec{p}\right) \leq A\left(\vec{f}^{l}\right) + \alpha^m \beta \left[\vec{g}^{l}\right]^T \vec{p} \tag{8}$$

to require that we achieve a reduction in $A\left(\vec{f}^{l}\right)$ during this step that is at least a fraction β of the reduction expected in Newton's Method. Given α_0, τ, β, the algorithm is:

Until $A\left(\vec{f}^{l} + \alpha^m \vec{p}\right) \leq A\left(\vec{f}^{l}\right) + \alpha^m \beta \left[\vec{g}^{l}\right]^T \vec{p}$:

(1) set $\alpha^{m+1} = \tau\alpha^m$

(2) increment m by 1

Once α^m is found, $\vec{f}^{l+1}$ is calculated by

$$\vec{f}^{l+1} = \vec{f}^{l} + \alpha^m \vec{p} \tag{9}$$

2.3 H^{l+1} calculation (Step 3)

We have to define the following vectors:

$$\vec{s} = \vec{f}^{l+1} - \vec{f}^{l}, \quad \vec{y} = \vec{g}^{l+1} - \vec{g}^{l}. \tag{10}$$

We can then calculate the updated inverse Hessian for the BFGS directly:

$$H^{l+1} = \left(I^n - \frac{\vec{y}\,\vec{s}^T}{\vec{y}^T\vec{s}}\right) H^l \left(I^n - \frac{\vec{y}\,\vec{s}^T}{\vec{y}^T\vec{s}}\right) + \frac{\vec{s}\,\vec{s}^T}{\vec{y}^T\vec{s}} \tag{11}$$

The form of this equation is messy, but it comes from a relatively simple argument which maintains that the inverse Hessian satisfies the secant condition,$\vec{s} = H^{l+1}\vec{y}$, and then defines a quadratic approximation of the function being minimized, in our case $A\left(\vec{f}^{l}\right)$. FIGURE 2 demonstrates the convergence of this algorithm applied to histogram data generated by the method described in section 1.1.

3 Weighted Histogram Analysis Method (WHAM)

The WHAM equations are commonly used within the field of Molecular Dynamic simulations as a means of generating PMF landscapes by determining the underlying unbiased probability distribution of measurements along coordinate axis descretized into a finite number of bins. A complete derivation of the equations associated with the method can be found in Hummer and Zhu's 2012 paper - **Convergence and error estimation in free energy calculations using the weighted histogram analysis method** - but I will

quickly show the equations necessary for BFGS algorithm and explain the meaning of the parameters. The function that was being minimized in section [2], $A\left(\vec{f}\right)$, is given by the following expression:

$$\begin{aligned} A\left(\vec{f}\right) &= -\sum_{i=0}^{S} N_i \ln\left(f_i\right) - \sum_{l}^{L} M_l \ln\left(\rho_l\right) \\ &= -\sum_{i=0}^{S} N_i \ln\left(f_i\right) - \sum_{l}^{L} M_l \ln\left(\frac{M_l}{\sum_i N_i f_i c_{i,l}}\right) \end{aligned} \tag{12}$$

where the summations over $i \in [0, 1, .., S]$ and $l \in [0, 1, .., L]$ denote summations over each of simulations (total number of simulations is $S+1$) and each of the coordinate bins (total bin number $L+1$) respectively. N_i is the total number of samples taken in simulation, i. $f_i = 1/\left(\sum_l c_{i,l}\rho_l\right)$ is the normalization constant for a given simulation, i and is computed as the inverse sum of the of the biasing coefficient associated with bin l for simulation i multiplied by the estimated probability distribution at bin l, ρ_l. The vector parameter of the function $A\left(\vec{f}\right)$, is composed of the elements $[f_0, ..., f_i, ..., f_S]$. M_l is the initial histogram value associated with bin l, produced by the cumulative samples taken in each bin across all of the simulations. As seen in equation [12], the expression can be written entirely in terms of the normalization constants, noting that the estimated probability distribution associated with each bin is given by $\rho_l = \frac{M_l}{\sum_i N_i f_i c_{i,l}}$.

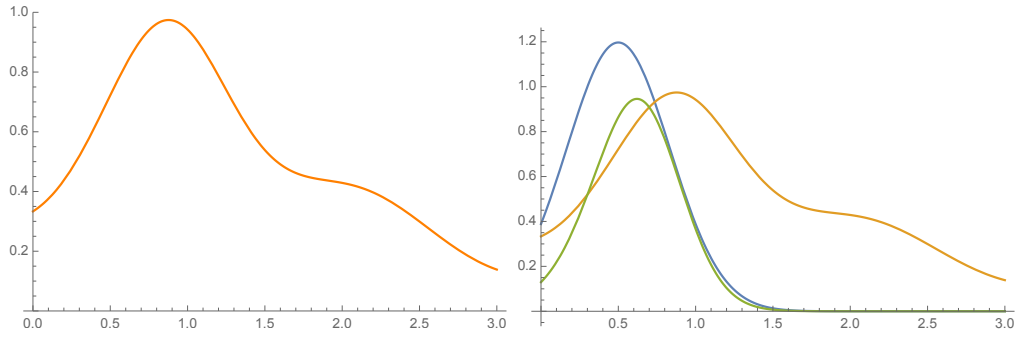
The gradient of $A\left(\vec{f}\right)$ with respect to the components $\vec{f}$, used to compute $\vec{g}$ in section [2], is given by:

$$\frac{\partial A}{\partial f_i} = N_i \left(\sum_{l=0}^{L} \frac{M_l c_{il}}{\sum_{j=0}^{S} N_j f_j c_{jl}} - \frac{1}{f_i}\right) \tag{13}$$

where the sum in the denominator is over all simulations. It should also be noted that the module here utilizes a change of variables, recommended in the above paper by Hummer and Zhu, to improve stability:

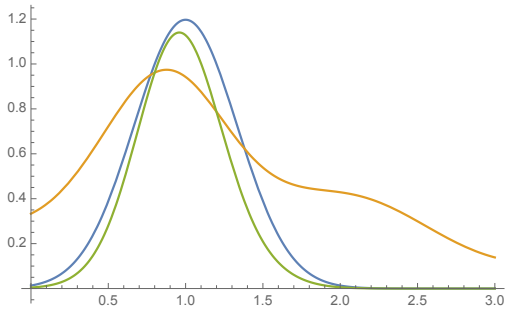
$$\begin{aligned} f_i' &= \ln\left(f_i\right) \\ f_i &= \exp\left[f_i'\right] \end{aligned} \tag{14}$$

The vector $\vec{f}$ is then replaced by the vector $\vec{f}'$ and the vector components f_i and f_j in equations [12] and [13] are replaced with $\exp\left[f_i'\right]$ and $\exp\left[f_j'\right]$.

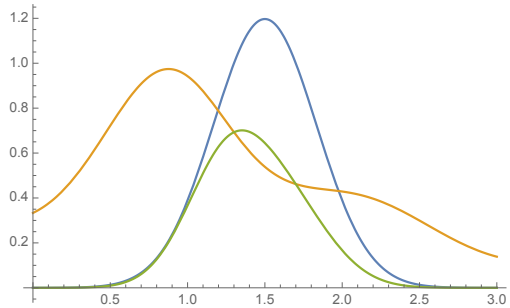


(a) $\rho(x)$

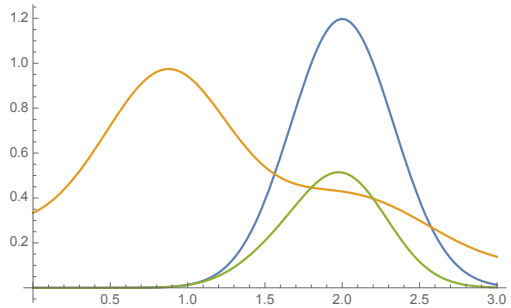
(b) $\mu_i = 0.5$



(c) $\mu_i = 1.0$

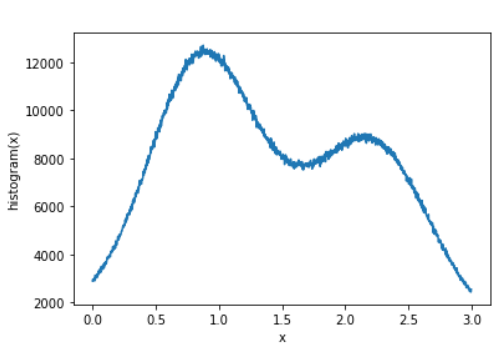


(d) $\mu_i = 1.5$

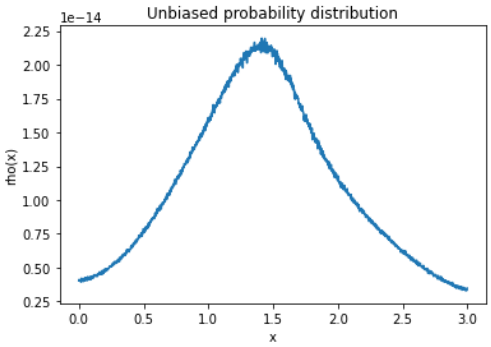


(e) $\mu_i = 2.0$

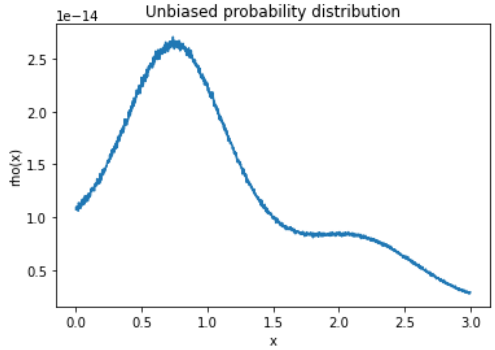
Figure 1: (a): Plot of the known probability distribution, $\rho(x)$ (equation [1]). (b-e): Plots of ORANGE : $\rho(x)$ (equation [1]), BLUE : the biasing potential $V_i'(x)$ (equation [4]), and GREEN : the biasing potential applied to the known probability distribution $\rho_i(x) = \rho(x) * V_i'(x)$ (equation [5]) for simulations with incremented values of μ_i (spring constant: $k = 9$). The green curve represents the probability distribution that the successive simulations discussed in section [1.1] sample from.



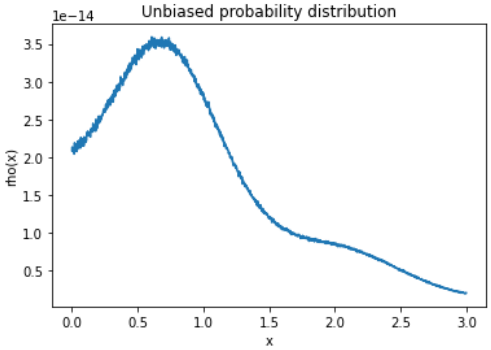
(a) Initial Histogram Data



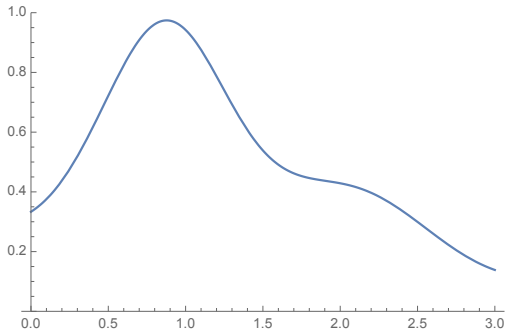
(b) $l = 1$



(c) $l = 5$



(d) $l = 10$



(e) $\rho\,(x)$

Figure 2: (a): The biased histogram generated by the method discussed in section [1.1], which is the starting point of the BFGS convergence. (b-c): The estimated unbiased distribution given after BFGS loop iteration l, illustrating the successive convergence toward (e), the known unbiased probability distribution (equation [1.1])