Published in final edited form as:

J Comput Chem. 2011 January 15; 32(1): 134–141. doi:10.1002/jcc.21609.

Accurate Estimation of Solvation Free Energy Using Polynomial Fitting Techniques

Conrad Shyu and

Department of Physics, University of Idaho, Moscow, ID 83844-0903

F. Marty Ytreberg

Department of Physics, University of Idaho, Moscow, ID 83844-0903

Conrad Shyu: conrads@uidaho.edu; F. Marty Ytreberg: ytreberg@uidaho.edu

Abstract

This report details an approach to improve the accuracy of free energy difference estimates using thermodynamic integration data (slope of the free energy with respect to the switching variable λ) and its application to calculating solvation free energy. The central idea is to utilize polynomial fitting schemes to approximate the thermodynamic integration data to improve the accuracy of the free energy difference estimates. Previously, we introduced the use of polynomial regression technique to fit thermodynamic integration data (Shyu and Ytreberg, *J Comput Chem* 30: 2297–2304, 2009). In this report we introduce polynomial and spline interpolation techniques. Two systems with analytically solvable relative free energies are used to test the accuracy of the interpolation approach. We also use both interpolation and regression methods to determine a small molecule solvation free energy. Our simulations show that, using such polynomial techniques and non-equidistant λ values, the solvation free energy can be estimated with high accuracy without using soft-core scaling and separate simulations for Lennard-Jones and partial charges. The results from our study suggest these polynomial techniques, especially with use of non-equidistant λ values, improve the accuracy for ΔF estimates without demanding additional simulations. We also provide general guidelines for use of polynomial fitting to estimate free energy. To allow researchers to immediately utilize these methods, free software and documentation is provided via <http://www.phys.uidaho.edu/ytreberg/software>.

1 Introduction

The Helmholtz or Gibbs free energy constitutes an important thermodynamic quantity to understand how chemical species recognize each other, associate or react. Examples of such problems include conformational equilibria and molecular association, partitioning between immiscible liquids, receptor-drug interaction, protein-protein and protein-DNA association, and protein stability^{1,2}. Thermodynamic integration (TI) is a commonly used approach for the calculation of free energy differences (ΔF) between two systems with potential energy functions U_1 and U_0 , respectively^{3,4,5,6,7}. The free energy difference, $\Delta F = F_1 - F_0$, is equivalent to the reversible work to switch from U_0 to U_1 , and can be determined by estimating the integral

$$\Delta F = \int_{\lambda=0}^1 \left\langle \frac{\partial U_\lambda}{\partial \lambda} \right\rangle_\lambda d\lambda. \quad (1)$$

The notation $\langle \partial U_\lambda / \partial \lambda \rangle_\lambda$ indicates an ensemble average at a particular value of λ . The variable λ permits continuously switching from one energy function to another. Switching potential

energies requires a continuously variable energy function U_λ such that $U_{\lambda=0} = U_0$ and $U_{\lambda=1} = U_1$. In addition, the free energy function U_λ must be differentiable with respect to λ for $0 \leq \lambda \leq 1$ ⁸.

The relationship in Equation 1 is exact. Numerical uncertainty comes into play because the integral must be approximated by performing simulations at various fixed, discrete, values of λ . Typically, these discrete λ values are used to convert the integral to a sum (e.g., using quadrature). If the estimate of $\langle \partial U_\lambda / \partial \lambda \rangle_\lambda$ is slow to converge, then it is necessary to perform very long simulations in order to reliably estimate the average. In addition, $\langle \partial U_\lambda / \partial \lambda \rangle_\lambda$ may heavily depend on λ so that a large number of fixed λ simulations is needed in order to estimate the integral with sufficient accuracy.

Previously, we reported the use of regression techniques to calculate the free energy from TI data⁹. Using analytical solvable examples, we were able to achieve highly accurate estimates. We note that regression does not produce a curve that goes through every data point exactly. Instead the regression model only describes the tendency of the data and does not represent the true functional form.

The purpose of this current study is to introduce polynomial and spline interpolation techniques to estimate ΔF and to use both interpolation and regression methods to estimate a small molecule solvation free energy. This differs from our previous regression study⁹ in that these polynomials interpolate the slope of the free energy $dF/d\lambda = \langle \partial U_\lambda / \partial \lambda \rangle_\lambda$ as a function of λ . The derived curve is forced to pass through each data point. Regression, by contrast, constructs the curve that minimizes the errors between the data and extrapolated points. The key motivation for these polynomial fitting approaches is that, even if the averages $\langle \partial U_\lambda / \partial \lambda \rangle_\lambda$ are fully converged (i.e., infinitely long sampling), there will be error in the ΔF estimates due to the fact that one must estimate the integral. Here we present methods which reduce this error and construct the polynomial that represents the functional form of TI data. We also examine the use of both equidistant and non-equidistant λ values. Two test systems previously used by us with analytical ΔF values are utilized to quantify the accuracy of the interpolation techniques⁹. We also use both regression and interpolation techniques to estimate a small molecule solvation free energy. The results from our simulations suggest that polynomial fitting, especially with use of non-equidistant λ values, improves the accuracy for ΔF estimates without demanding additional simulations. In addition, polynomial fitting techniques permit accurate estimates of free energy difference from TI without the need for soft-core scaling and separate simulations for partial charges and Lennard-Jones potentials.

2 Theory

The primary focus of this study is to present a mathematical framework for the analysis of fixed TI simulation data using polynomial and spline interpolation techniques. The objective is then to construct a polynomial that best fits the TI data. The functional form of the simulation data is represented by a series of data points $\{\lambda, dF/d\lambda\}$, thus a polynomial is constructed through these data. Interpolation refers to the problem of determining a function that exactly represents given data points. Mathematically the polynomial obtained from interpolation is considered to be the one that best represents all data points within the given interval ($\lambda = [0, 1]$)^{10,11}.

Polynomial interpolation and regression techniques generally give a similar result, especially when high degrees of polynomials are employed to fit the TI data. However, the curve obtained from polynomial interpolation passes through each data point whereas the curved generated via regression does not. If TI simulations are fully converged (e.g., infinite sampling), then polynomial interpolation should give the most accurate estimate.

Regression, on the other hand, utilizes the least squares method that minimizes the sum of the square of the difference between the actual value and the approximated one. Regression does have an advantage over interpolation in that the approximation is independent of the number of data points. However, we note that regression may not give the best estimates if the TI data include several inflection points due to the fact that regression does not attempt to construct a curve that passes through every data point.

2.1 Mathematical Notation

To simplify the mathematical expressions and to be consistent with the most commonly used notation in polynomial and spline interpolation papers, we denote the switching variable λ by x , and, $dF/d\lambda$ by y . We also denote the actual continuous function $dF/d\lambda$ by $f(x)$. The functional form of the simulation data is thus represented by a series of data points $\{\lambda, dF/d\lambda\} = \{x, y\}$, and the approximating function $p(x)$ is constructed through these data points. The following sections briefly introduce mathematical definitions and properties of both Lagrange and Newton polynomials, and the cubic spline function.

2.2 Lagrange and Newton Polynomial Interpolation

The classical Weierstrass theorem establishes the mathematical foundation for polynomial approximation¹². The theorem asserts that there exists a polynomial $p(x)$ for approximating the continuous function $f(x)$ defined within a closed interval $[a, b] = [0, 1]$, and the polynomial approximation $p(x)$ can get arbitrarily close to the function $f(x)$ as the degree of polynomial is increased¹³. The most straightforward method to obtain the polynomial $p(x)$ of degree n is to calculate the values of $f(x)$ for $n + 1$ distinct fitting data points within the interval of $a \leq x \leq b$, and satisfy the simultaneous equation $y_i = f(x_i) = p(x_i)$, for $i = 0, 1, 2, \dots, n$. Lagrange and Newton polynomials are the most commonly used methods for interpolation^{13,15}.

The coefficients of the above polynomial $p(x)$ then form the basis of the Lagrange interpolation formula. The Weierstrass theorem contends that there is always a unique polynomial $p(x)$ of degree n that satisfies these conditions. The Lagrange interpolation polynomial $L(x)$ is a linear combination of Lagrange basis polynomials $l_i(x)$ such that

$$L(x) = \sum_{i=0}^n y_i l_i(x) = \sum_{i=0}^n y_i \left[\prod_{j \neq i, j=0}^n \frac{x - x_j}{x_i - x_j} \right]. \quad (2)$$

It is worth noting that the degree of polynomial n cannot be chosen arbitrarily but is determined by the number of data points $n + 1$. This is important since the degree of interpolating polynomials will thus be determined by the number of λ values used to estimate ΔF .

Mathematically, the Lagrange form of the interpolation polynomial is generally preferred in proof and theoretical arguments because the derivatives of the polynomial are continuous, and maxima or minima always exist^{10,11}. The construction of Lagrange polynomials, however, is computationally demanding because all Lagrange basis polynomials have to be reevaluated each time y_i is updated. By contrast, Newton interpolation utilizes the divided difference (more on this below) which is generally more computationally efficient¹⁴.

A more practical form of interpolation polynomial for computational purposes is given by Newton polynomials that utilize the divided difference. The divided difference is defined as the ratio of the difference in the function values, y_i or $f(x_i)$, at two points divided by the

difference in the values of the corresponding independent variable, x_i . Divided differences do not require the recalculations of coefficients if new data points are included. As a consequence, it is generally more computationally efficient to use the divided differences method for interpolation¹⁵. Similar to the Lagrange polynomial, the Newton polynomial is a linear combination of basis polynomials

$$N(x) = \sum_{i=0}^n a_i \left[\prod_{j=0}^{i-1} (x - x_j) \right], \quad (3)$$

where the coefficients $a_i = g[x_0, \dots, x_i]$ is the notation for divided differences. The first divided difference, for example, between data points x_0 and x_1 is given by

$$a_1 = g[x_0, x_1] = \frac{f(x_1) - f(x_0)}{x_1 - x_0} = \frac{y_1 - y_0}{x_1 - x_0}. \quad (4)$$

The second divided difference between data points x_0 , x_1 , and x_2 is given by

$$a_2 = g[x_0, x_1, x_2] = \frac{g[x_1, x_2] - g[x_0, x_1]}{x_2 - x_0}. \quad (5)$$

Accordingly the n -th divided difference between data points $x_0, x_1, \dots, x_n$ is given by

$$a_n = g[x_0, x_1, \dots, x_n] = \frac{g[x_1, x_2, \dots, x_n] - g[x_0, x_1, \dots, x_{n-1}]}{x_n - x_0}. \quad (6)$$

The Newton interpolation polynomial is then given by

$$N(x) = f(x_0) + g[x_0, x_1](x - x_0) + g[x_0, x_1, x_2](x - x_0)(x - x_1) + \dots + g[x_0, x_1, \dots, x_i](x - x_0)(x - x_1) \dots (x - x_{i-1}). \quad (7)$$

The calculations of the divided differences form a successive, recurrent relationship between the previous two coefficients. Computationally, the divided differences can be written in the form of a table that significantly simplifies the algorithmic implementation¹⁵.

2.3 Spline Function Interpolation

Studies have suggested that the spline function is typically preferred over polynomials because the interpolation error can be reduced by using low degrees of polynomials^{16,17,18}. Spline interpolation also avoids the Runge's phenomenon which occurs when using high degree polynomials¹⁹. Runge's phenomenon is when approximation errors increase dramatically as the degree of interpolation polynomials increases.

Spline functions are piecewise polynomials of the same degree^{17,20}. Unlike the Lagrange and Newton interpolation polynomials, the degree of the piecewise polynomials does not depend on the number of data points. The piecewise polynomials join together at the data points $\{\lambda, dF/d\lambda\}$ called knots and must be continuous¹⁰. The main advantages of spline interpolation are stability and simplicity²¹. Moreover, the use of polynomials of lower

degree offers the possibility to avoid the oscillatory behavior that commonly arises from fitting a single polynomial exactly to a large number of data points¹⁰.

We utilize the cubic spline which is the most commonly used type. A cubic spline function is a set of third degree piecewise polynomials that provide a smooth curve passing through all data points. Because the segments join with matching derivatives up to order two, the curvature of the polynomials changes smoothly along the knots. The fundamental idea behind cubic spline interpolation is based on the engineer's tool used to draw smooth curves through a number of points. This spline consists of weights attached to a flat surface at the points to be connected. A flexible strip is then bent across each of these weights, resulting in a smooth curve. The mathematical spline is similar in principle. The points, in this case, are TI data. The weights are the coefficients of the cubic polynomials used to interpolate the data²².

Mathematically, the cubic spline function $S_i(x)$ is defined as

$$S_i(x) = \frac{z_{i+1}}{6h_i}(x - x_i)^3 + \frac{z_i}{6h_i}(x_{i+1} - x)^3 + \left(\frac{y_{i+1}}{h_i} - \frac{h_i}{6}z_{i+1}\right)(x - x_i) + \left(\frac{y_i}{h_i} - \frac{h_i}{6}z_i\right)(x_{i+1} - x), \quad (8)$$

where $h_i = x_{i+1} - x_i$ is the size of the i -th interval $[x_i, x_{i+1}]$. The spline function $S_i(x)$ provides a cubic polynomial on the interval $x \in [x_i, x_{i+1}]$. The values of z_i are the second derivatives at the endpoints such that $z_i = S_i''(x_i)$. To compute the values of z_i , it is necessary to solve the recurrent equation

$$h_{i-1}z_{i-1} + 2(h_{i-1} + h_i)z_i + h_i z_{i+1} = 6 \left(\frac{y_{i+1} - y_i}{h_i} - \frac{y_i - y_{i-1}}{h_{i-1}} \right), \quad (9)$$

for $i = 0$ to $n - 1$.

In the current study, natural spline boundary conditions were implemented to construct the interpolation polynomials. There are two boundary condition types which could be utilized. The natural cubic spline imposes the boundary conditions $z_0 = 0$ and $z_n = 0$, and the clamped cubic spline requires z_0 and z_n for a given function^{16,17}. When the natural boundary conditions are used, the spline assumes the shape that a long flexible curve would take if it is forced to go through all the data points. A natural spline permits the slope at the ends be free to equilibrate to the position that minimizes the oscillatory behavior of the curve. A clamped cubic spline, on the other hand, further requires a piecewise cubic function which passes through the given set of knots with a given function or value. The derivatives and second derivatives of adjacent cubic polynomials must also agree at the interior abscissae. It is worth noting that spline interpolation polynomials are piecewise and do not always approximate the $dF/d\lambda$ completely. This is particularly evident when the choice of λ neglects the inflection points along the curve.

2.4 Algebraic Formulation

To minimize the numerical errors in the ΔF estimation process, interpolating polynomials were first transformed into their analytical forms, and then all integrals were evaluated analytically. The derivation of the analytical integration form for the interpolating polynomials involves the expansion of every coefficient in the polynomial equation. The

rearrangement of coefficients permits analytical integration of the interpolating polynomials. The analytical integral for the Lagrange polynomial is expressed as

$$\int L(x)dx = \sum_{i=0}^n y_i \int l_i(x)dx = \sum_{i=0}^n y_i \int \prod_{j=0, j \neq i}^n \frac{x-x_j}{x_i-x_j} dx. \quad (10)$$

Similarly, the analytical integral for the Newton polynomial is expressed as

$$\int N(x)dx = \sum_{i=0}^n a_i \int \prod_{j=0}^{i-1} (x-x_j)dx, \quad (11)$$

where the coefficient a_i is the notation for divided differences; see Equation 3.

For spline functions, analytical integrations must be performed at each subinterval. Similar to that of Lagrange and Newton polynomial, the integration form requires the expansion and rearrangement of all coefficients. The analytical integral form of the cubic spline is expressed as

$$\begin{aligned} \int S_i(x)dx = & x^3 \left[\frac{1}{6h_i}(z_{i+1} - z_i) \right] \\ & + x^2 \left[\frac{1}{2h_i}(z_i x_{i+1} - z_{i+1} x_i) \right] \\ & + x \left[\frac{1}{2h_i}(z_{i+1} x_i^2 - z_i x_{i+1}^2) + \frac{1}{h_i}(y_{i+1} - y_i) - \frac{h_i}{6}(z_{i+1} - z_i) \right] \\ & + \frac{1}{6h_i}(z_i x_{i+1}^3 \\ & - z_{i+1} x_i^3) \\ & + \frac{1}{h_i}(y_i x_{i+1} - y_{i+1} x_i) \\ & + \frac{h_i}{6}(z_{i+1} x_i \\ & - z_i x_{i+1}). \end{aligned} \quad (12)$$

Similar to that of Lagrange and Newton interpolation polynomials, each x_i is replaced by λ_i and y_i by $\langle \partial U_\lambda / \partial \lambda \rangle_\lambda$. Specifically, for Lagrange interpolation

$$\Delta F = \int_0^1 L(\lambda) d\lambda = \sum_{i=0}^n \left\langle \frac{\partial U_\lambda}{\partial \lambda} \right\rangle_{\lambda_i} \int_0^1 \prod_{j=0, j \neq i}^n \frac{\lambda - \lambda_j}{\lambda_i - \lambda_j} d\lambda \quad (\text{Lagrange}) \quad (13)$$

and for Newton interpolation

$$\Delta F = \int_0^1 N(\lambda) d\lambda = \sum_{i=0}^n a_i \int_0^1 \prod_{j=0}^{i-1} (\lambda - \lambda_j) d\lambda \quad (\text{Newton}) \quad (14)$$

with the divided difference $a_i = g[\lambda_0, \lambda_1, \dots, \lambda_i]$ given by Equation 6. To calculate ΔF using cubic spline, integrals are evaluated at each subinterval $[\lambda_{i-1}, \lambda_i]$, and the total ΔF is the sum of all integrals

$$\Delta F = \sum_{i=1}^n \int_{\lambda_{i-1}}^{\lambda_i} S_i(\lambda) d\lambda \quad (\text{Cubic Spline}) \quad (15)$$

where S_i are given by Equation 8.

3 Computational Details

Two types of free energy calculations were performed in order to test our interpolation approaches. The first type utilized two test systems with analytical ΔF solutions. Simulations were performed using the same protocols as previously reported⁹. These test systems provide an unbiased measure of the accuracy of the interpolation techniques for estimating ΔF .

The second type of simulation was to compute the solvation free energy for a 4-hydroxy-2-butanone (BUQ) molecule²⁶. We used both regression⁹ and interpolation techniques and compared them to use of quadrature. Details of the solvation free energy calculation are described below.

3.1 Test Systems

Procedures for the simulations of the two test systems were previously reported⁹. Briefly, these systems involve two potential functions $U_0(\xi) = \frac{1}{2}\xi^2$ and $U_1(\xi) = 2(\xi - 5)^2$ for system one, and $U_0(\xi) = \frac{1}{2}\xi^2$ and $U_1(\xi) = \frac{1}{2}(\xi - 5)^2$ for system two with the analytical free energy $\Delta F = -\frac{1}{2}\ln\frac{1}{4}$ and $\Delta F = -\frac{1}{2}\ln 5$ respectively⁹. An equal amount of simulation time (1,000,000 Monte Carlo steps) was spent for each of six ($\lambda = 0.0, 0.2, 0.4, 0.6, 0.8$, and 1.0) and eleven ($\lambda = 0.0, 0.1, 0.2, \dots, 0.9$, and 1.0) equidistant λ values. Identical equilibrium simulations were performed on the corresponding six non-equidistant λ values ($\lambda = 0.0, 0.0955, 0.3455, 0.6545, 0.9045$, and 1.0) and eleven ($\lambda = 0.0, 0.0245, 0.0955, 0.2061, 0.3455, 0.5, 0.6545, 0.7939, 0.9045, 0.9755$, and 1.0). Averages of the slope $dF/d\lambda = \langle \partial U_\lambda / \partial \lambda \rangle_\lambda$ were collected for each value of λ . For each λ value, simulations were given 1,000 steps to equilibrate initially, then were allowed to proceed for 1,000,000 Monte Carlo steps. Each trial started with an arbitrarily chosen initial position for the particle. Simulations were performed sequentially starting at $\lambda = 0$. Trial moves for Metropolis Monte Carlo²⁵ were generated by adding a uniform random deviate between -0.5 and 0.5 to the current position resulting in a 40 to 50% acceptance ratio. A total of 1,000 independent trials were generated from each case. We recorded the biases ($\Delta F_{\text{exact}} - \Delta F_{\text{estimate}}$) for each of 1,000 independent runs and the corresponding averages and standard deviations were calculated in order to provide a measure of the accuracy of the ΔF estimates.

3.2 BUQ Solvation

We applied our interpolation and regression techniques to compute the solvation free energy for a BUQ molecule²⁶. We first obtained a reference estimate for the solvation free energy ΔF_{ref} . To obtain ΔF_{ref} with high accuracy, we performed separate simulations for changes in the Lennard-Jones parameters and the partial charges²⁷. Studies have shown that use of separate simulations provides a higher accuracy estimate and avoids possible endpoint singularity problems^{6,27,28}. Thus, the first stage to obtaining ΔF_{ref} was to compute the free energy associated with the Lennard-Jones potential. This was accomplished by setting all the

partial charges of the solute to zero. We then “grew” the BUQ molecule in the solvent using soft-core scaling to ensure a smooth potential energy curve, even for interpenetrating molecules^{28,29}. Once the neutral atoms are fully grown in the solvent, we performed the second stage to compute the free energy associated with the partial charges. This was accomplished by increasing the partial charges from zero to their final values given by the forcefield. For both stages, we employed trapezoidal quadrature to integrate the free energy slope and obtain the free energy for each of the two stages. We computed ΔF_{ref} using this two-stage approach for both inserting and deleting BUQ.

The initial coordinates for the BUQ were extracted from the FKBP-ligand complexes (1D7J)²⁶ which were retrieved from the Protein Data Bank. The topologies for BUQ were then generated by the PRODRG server³⁰. The simulations were performed using the GRO-MACS package 3.3.3³¹ at constant temperature in a periodic cubic box with an edge length of approximately 2.4 nm and the default GROMOS-96 43A1 forcefield. This corresponded to approximately 500 SPC³² water molecules. The neighbor list for nonbonded interactions was updated every 10 steps. The bond distances and bond angles of water were constrained using the LINCS algorithm³³. To obtain a isothermal-isobaric ensemble at 300 K, a leap-frog stochastic dynamics³⁴ was used to integrate the equations of motion with a 2.0 fs timestep. The temperature was maintained using Langevin dynamics with a friction coefficient of 1.0 amu/ps. The pressure was maintained at 1.0 atm using the Parrinello-Rahman algorithm³⁵. The pressure coupling time was set to 1.0 ps, and the isothermal compressibility was set to $4.5 \times 10^{-5} \text{ bar}^{-1}$. Particle-mesh summation Ewald algorithm^{36,37} was used for electrostatics with a real-space cutoff of 1.0 nm and a Fourier spacing of 0.1 nm. Van der Waals interactions used a cutoff with a smoothing function such that the interactions smoothly decayed to zero between 0.75 nm and 0.9 nm. Dispersion corrections for the energy and pressure were also utilized³⁸.

The BUQ system was first minimized using steepest descent for 500 steps. 100 ps of constant volume simulation were performed for system equilibration, followed by 100 ps of constant pressure simulation. Then 10 ns constant pressure simulations were performed with eleven equidistant λ values ($\lambda = 0.0, 0.1, 0.2, \dots, 0.9$, and 1.0). Simulations were performed sequentially starting at $\lambda = 0$ and averages of the slope $dF/d\lambda = \langle \partial U_\lambda / \partial \lambda \rangle_\lambda$ were collected for each value of λ . For the first stage, we computed the free energy for the Lennard-Jones interactions using a soft-core scaling parameter of $\alpha = 0.5$. For the second stage, we computed the free energy for the partial charge interactions using $\alpha = 0.0$ ²⁷. Four additional λ values ($\lambda = 0.65, 0.75, 0.85, 0.95$ for deleting BUQ and $\lambda = 0.05, 0.15, 0.25, 0.35$ for inserting) were included in the simulations for Lennard-Jones potentials to further refine the ΔF soft-core estimate where the free energy slopes undergo the most change.

The total free energy is the sum of the free energies from simulations in which the Lennard-Jones parameters and partial charges were changing. We determined that $\Delta F_{\text{ref}}^{\text{del}} = 31.6 \text{ kJ/mol}$ for deleting BUQ and $\Delta F_{\text{ref}}^{\text{ins}} = -31.5 \text{ kJ/mol}$ for inserting BUQ.

4 Results and Discussion

We now summarize the results from our polynomial fitting methods applied to two test systems and to BUQ solvation. The analytical test systems provide an objective assessment of the accuracy of the ΔF estimates. The BUQ solvation provides a demonstration of our approach on a common chemical problem. For the results below, interpolation or regression polynomials for each set of TI data were first transformed into the analytical form (Equations 13, 14, and 15) and integration was then performed analytically. Results

presented below show that use of polynomial fitting with non-equidistant λ values results in more accurate ΔF estimates than use of quadrature.

4.1 Test Systems

Table 1 summarizes the averages and standard deviations of biases for system one and two from 1,000 independent trials using six and eleven equidistant and non-equidistant λ values utilizing both interpolation and regression techniques⁹. Both the Lagrange and Newton interpolating polynomials provide the most accurate estimates, but often with larger uncertainty than that of spline or quadrature.

Using equidistant λ values, the estimates obtained from trapezoidal quadrature are most heavily biased followed by cubic spline. Lagrange and Newton polynomials give somewhat biased estimates, however, the magnitude of bias is much smaller than that of the cubic spline and trapezoidal quadrature. The estimates obtained from regression show nearly identical accuracy to that of Lagrange and Newton. As the number of λ values increases, the accuracy improve. Using eleven equidistant λ values, for example, the Lagrange and Newton polynomials give the most accurate estimates of ΔF but the uncertainty also increases. The estimates obtained from regression are slightly biased but with smaller uncertainty compared to that of Lagrange and Newton interpolation. This is due to the fact that regression does not attempt to construct an interpolating polynomial that goes through each λ value exactly. The estimates obtained from cubic spline are also slightly biased but with small uncertainty. The fact that the uncertainty remains relatively unchanged for cubic spline in both test systems reflects that fact that cubic spline utilizes a low degree piecewise polynomial and is thus less sensitive to changes of $dF/d\lambda$.

Using non-equidistant λ values, the Lagrange and Newton interpolating polynomials are able to accurately estimate ΔF . The uncertainty, however, remains similar to use of equidistant λ values. The estimates obtained from regression are more biased than that of Lagrange and Newton. Overall, these estimates are considerably more accurate than that of using equidistant λ values. These results suggest that non-equidistant λ values are more appropriate for estimating ΔF , especially when polynomial interpolation or regression techniques are used. Non-equidistant λ values significantly improve the overall estimate accuracy. Quadrature results by contrast do not improve with the use of non-equidistant λ values and remain heavily biased. We also note that, unlike quadrature, the steep free energy curvature for system two does not appear to significantly impact the accuracy of ΔF estimates obtained from the Lagrange and Newton polynomials. The curvature, however, slightly affects the accuracy of estimates obtained from regression. The results also show that cubic spline estimates are more accurate than that of trapezoidal quadrature but less accurate than Lagrange or Newton.

4.2 BUQ Solvation

We first performed a two-stage simulation for changing Lennard-Jones parameters and partial charges in order to obtain the reference solvation free energies $\Delta F_{\text{ref}}^{\text{ins}}$ and $\Delta F_{\text{ref}}^{\text{del}}$. We then determined the free energy estimate with no soft-core scaling utilizing a single stage, i.e., Lennard-Jones parameters and partial charges were simultaneously changed. Figure 1 shows the free energy curves from the simulations using eleven equidistant (Figure 1a) and non-equidistant λ values (Figure 1b). The curves are all well-behaved except when λ is close to either 0.0 or 1.0. As expected, the figures show clear evidence of the endpoint singularity when the vanishing or growing atoms become close to the other atoms^{5,27}. The rapid changes of the free energy curves at the endpoint introduce significant errors into the numerical integration using trapezoidal quadrature. It is worth noting that the use of non-

equidistant λ significantly alleviates the endpoint singularity problem, even for quadrature, because more data points are used near $\lambda = 0.0$ and 1.0 .

Table 2 summarizes the ΔF estimates obtained using both interpolation and regression techniques with equidistant and non-equidistant λ values. The reference free energies $\Delta F_{\text{ref}}^{\text{ins}}$ and $\Delta F_{\text{ref}}^{\text{del}}$ were used to estimate the accuracy of these polynomial fitting techniques. For equidistant λ values, while none of the integration techniques give an accurate estimate of ΔF , Lagrange and Newton polynomials provide the best results. The use of non-equidistant λ values, significantly improves the accuracy of all ΔF estimates. The improvement is particularly evident with the estimates obtained using Lagrange and Newton polynomials. It is worth noting that the estimates obtained from regression are nearly identical to that of Lagrange and Newton interpolation. These results clearly demonstrate that the use of a continuous function to approximate the $dF/d\lambda$ curve is a very promising approach for accurate estimate of ΔF . Cubic spline interpolation delivers slightly biased estimates but is still considerably more accurate than that of trapezoidal quadrature.

For BUQ solvation, our study has shown that using the polynomial fitting techniques for interpolation and extrapolation and use of non-equidistant λ values, ΔF can be estimated within the accuracy of ± 0.4 kJ/mol *without using soft-core scaling and performing separate simulations for Lennard-Jones or partial charges*. Steinbrecher et al.²⁷ showed that the soft-core potentials are most promising for alchemical TI simulations and deliver ΔF estimates with accuracies of 0.4 kJ/mol. However, our simulations suggest that similar accuracy can be achieved using non-equidistant λ values and polynomial fitting techniques without soft-core scaling.

4.3 Guidelines for Use of Polynomial Fitting

To better facilitate the use of the approaches in this study, we propose guidelines for estimating ΔF using polynomial fitting techniques. These suggestions are largely based on the results from our well-converged simulations and also on the mathematical theories that describe the behavior of each technique. We note that the construction of an interpolating polynomial requires complex and recursive calculations that involve a large number of parameters. These calculations can quickly accumulate significant rounding errors. Thus, we recommend that one should always look at the interpolating polynomial to ensure that estimates are reasonable.

For well-converged TI simulations, we recommend Lagrange or Newton polynomial interpolation technique for the estimate of ΔF provided that the number of λ values is eleven or less. Lagrange and Newton showed the least bias in our study. We believe that this is because the interpolating polynomial is forced to pass through each data point exactly. For more than eleven λ values, cubic spline is recommended. The estimates from the cubic spline are not always as accurate as that of Lagrange and Newton, however the use of low degree polynomials offers protection against the oscillatory behavior that commonly arises from fitting a single polynomial exactly to a large number of data points. With cubic spline, the accuracy should improve with use of more λ values. Lagrange or Newton polynomial interpolation technique is not suitable for the estimates of ΔF with more than eleven λ values because high degree polynomials have been shown to be numerically unstable^{10,39,40,41}. This effect is particularly evident for equidistant λ values¹⁰. This instability is due to the fact that high degree polynomials are more sensitive to changes of $dF/d\lambda$ and the interpolating will oscillate significantly toward the endpoints. This oscillation is commonly referred to as Runge's phenomenon (more on this below)^{10,39}.

If the TI data is noisy or convergence is unknown then regression should be used instead of interpolation. Regression does not attempt to construct a curve that passes through all data points exactly. This property permits regression to construct a polynomial that is less sensitive to fluctuations in $\Delta F/d\lambda$. In addition, regression also possesses a unique advantage that the degree of fitting polynomial does not depend on the number of λ values⁹. Therefore one can choose the polynomial degree low enough to avoid Runge's Phenomenon. We caution that the degree of polynomial should not exceed the number of λ values to avoid overfitting.

Researchers that have not yet generated data are strongly encouraged to use non-equidistant λ values and polynomial fitting techniques. Our study revealed that polynomial interpolation together with eleven or less non-equidistant λ values always gave the most accurate estimates of ΔF . The use of non-equidistant λ values prevents the oscillatory behavior in high degree of polynomials. Runge's phenomenon is the classic illustration of polynomial interpolation that fails to provide accurate fitting. Rigorous proof requires the evaluation of complex integrals via the theory of residues^{40,41}. Epperson first presented an informal, yet comprehensive, mathematical and numerical explanation to this phenomenon¹⁹. Runge's phenomenon basically intimates that equidistant interpolation with high degree polynomial can be very sensitive to disturbances to the data within the interval. Equidistant interpolation only works well near the center of the interval. The interpolating polynomial oscillates widely at the endpoints and the approximation diverges profoundly. The divergence can also be minimized by using data points that are distributed more densely towards the ends of the interval. The most common example of such an arrangement is Chebyshev (non-equidistant) data points as was used in the current study.

5 Conclusion

We have introduced polynomial interpolation, and have implemented polynomial interpolation and regression techniques to estimate a small molecule solvation free energy using thermodynamic integration (TI) simulation data. Our study utilized two test systems with analytical ΔF values to objectively quantify the accuracy of the interpolation techniques. We also used both interpolation and regression to estimate the solvation free energy for BUQ. These polynomial approaches rely on constructing globally optimal polynomials that best fit the TI data, which are then used to estimate ΔF . Additional algebraic calculations were performed to permit analytical integration eliminating potential errors due to numerical evaluation of the integral.

Studies have shown that the use of soft-core scaling improves the accuracy of ΔF estimates to within ± 0.4 kJ/mol for alchemical TI simulations²⁷. Our results suggest that similar accuracy can also be achieved using non-equidistant λ values and polynomial fitting techniques without soft-core scaling of Lennard-Jones potentials.

We have shown that polynomial and spline interpolation techniques provide more accurate ΔF estimates than trapezoidal quadrature for the systems studied here. Lagrange or Newton polynomial interpolation technique is recommended if the TI simulations with eleven or less λ values are well converged. Our study has shown that Lagrange and Newton give the most accurate estimates with the least amount of bias. We also encourage researchers that have not yet generated data to use non-equidistant λ values and polynomial fitting techniques. For more than eleven λ values, cubic spline is recommended. Finally, regression should be used if the TI data is noisy or convergence is unknown because regression is less sensitive to fluctuations in $\Delta F/d\lambda$.

To allow researchers to immediately utilize these methods, free software and documentation is provided via <http://www.phys.uidaho.edu/ytreberg/software>.

Acknowledgments

The project described was supported by Award Number P20RR016448 from the National Institute of Health. The content is solely the responsibility of the authors and does not necessarily represent the official views of the National Institute of Health. This research was also supported by Idaho NSF-EPSCoR, Bionanoscience at University of Idaho (BANTech), and the Initiative for Bioinformatics and Evolutionary Studies (IBEST) at University of Idaho.

References

1. Chipot, C.; Pohorille, A. Free energy calculations. Springer; New York, NY: 2007.
2. Reddy, M.; Erion, M. Free energy calculations in rational drug design. Kluwer Academic/Plenum Publishers; New York, NY: 2001.
3. Kirkwood J. Journal of Chemical Physics. 1935; 3:300–313.
4. Mordasini T, McCammon J. Journal of Physical Chemistry B. 2000; 104:360–367.
5. Shirts M, Pande V. Journal of Chemical Physics. 2005; 122:144107. [PubMed: 15847516]
6. Shirts M, Pitera J, Swope W, Pande V. Journal of Chemical Physics. 2003; 119:5740.
7. Ytreberg F, Swendsen R, Zuckerman D. Journal of Chemical Physics. 2006; 125:1–11.
8. Leech, A. Molecular modeling, principle and applications. 2. Pearson Education Limited; Harlow, Essex, UK: 2001.
9. Shyu C, Ytreberg F. Journal of Computational Chemistry. 2009; 30:2297–2304. [PubMed: 19266482]
10. Faires, J.; Burden, R. Numerical methods. PWS-Kent Publishing Company; Boston, MA: 1993.
11. Henry M. The American Mathematical Monthly. 1981; 91:497–499.
12. Bishop E. Pacific Journal of Mathematics. 1961; 11:777–783.
13. Jeffreys, H.; Jeffreys, B. Methods of mathematical physics. 3. Cambridge University Press; Cambridge, England: 1988. Lagrange's interpolation formula.
14. Lee E. The American Mathematical Monthly. 1999; 96:618–622.
15. Werner W. Mathematics of Computation. 1984; 43:205–217.
16. Ahlberg, L.; Nilson, E.; Walsh, J. The theory of splines and their applications. Academic Press; New York: 1967.
17. Greville, T. Theory and applications of spline functions. Academic Press; New York: 1969.
18. Schultz M. Mathematics of Computation. 1970; 24:507–515.
19. Epperson J. The American Mathematical Monthly. 1987; 94:329–341.
20. deBoor, C. A practical guide to splines. Springer-Verlag; New York, NY: 1978.
21. Wold S. Technometrics. 1974; 16:1–11.
22. Schultz, M. Spline analysis. Prentice Hall; Englewood Cliffs, NJ: 1973.
23. Zellmer G. SIAM Review. 1977; 19:1–4.
24. Süli, E.; Mayers, D. An introduction to numerical analysis. Cambridge University Press; Cambridge, UK: 2003.
25. Metropolis N, Rosenbluth AW, Rosenbluth MN, Teller AH, Teller E. Journal of Chemical Physics. 1953; 21:1087–1092.
26. Burkhard P, Taylor P, Walkinshaw M. Journal of Molecular Biology. 2000; 295:953–962. [PubMed: 10656803]
27. Steinbrecher T, Mobley D, Case D. Journal of Chemical Physics. 2007; 127:214108. [PubMed: 18067350]
28. Zacharias M, Straatsma T, McCammon J. Journal of Chemical Physics. 1994; 100:9025.
29. Beulter T, Mark A, van Schaik R, Gerber P, van Gunsteren W. Chemical Physics Letters. 1994; 222:529–539.

30. Schüttelkopf A, van Aalten D. *Acta Crystallographica*. 2004; D60:1355–1363.
31. Lindahl E, Hess B, van der Spoel D. *Journal of Molecular Modeling*. 2001; 7:306–317.
32. Robinson, G.; Zhu, S.; Sigh, S.; Evans, M. *Water in biology, chemistry and physics: experimental overviews and computational methodologies*. World Scientific; Singapore: 1996.
33. Hess B, Bekker H, Berendsen H, Fraaije J. *Journal of Computational Chemistry*. 1997; 18:1463.
34. van Gusteren W, Berendsen H. *Molecular Simulation*. 1988; 1:173–185.
35. Parrinello M, Rahman A. *Physical Review Letters*. 1980; 45:1196–1199.
36. Darden T, York D, Pedersen L. *Journal of Chemical Physics*. 1993; 98:10089.
37. Essmann U, Perera L, Berkowitz M, Darden T, Lee H, Pedersen G. *Journal of Computational Physics*. 1995; 103:8577.
38. Allen, M.; Tildesley, D. *Computer simulation of liquids*. Oxford University Press; Oxford, UK: 1989.
39. Dahlquist, G.; Åke, B. *Numerical Methods*. Dover Publications; Mineola, NY: 1996. p. 101-103.
40. Kilgore T. *Journal of Approximation Theory*. 1978; 24:273–288.
41. deBoor C, Pinkus A. *Journal of Approximation Theory*. 1978; 24:289–303.

Figure 1.

Solvation free energy curves for BUQ with eleven equidistant (a) and non-equidistant (b) λ values. Both insertion (solid curve) and deletion (dotted curve) were computed. The fitting curves shown were constructed using Lagrange interpolating polynomials. Note that the endpoint singularity problem is present when the growing and vanishing atoms are too close to the other atoms ($\lambda = 0.0$ or 1.0). The use of non-equidistant λ values allows a smoother polynomial fit.

Table 1

Averages and standard deviations of the biases ($\Delta F_{\text{exact}} - \Delta F_{\text{estimate}}$) for system one and two using six and eleven equidistant and non-equidistant λ values. The exact solution for system one is $\Delta F_{\text{exact}}^{\text{one}} = -\frac{1}{2} \ln \frac{1}{4}$ and for system two $\Delta F_{\text{exact}}^{\text{two}} = -\frac{1}{2} \ln 5$ as previously reported⁹. Standard deviations of the biases are shown in parentheses. Results obtained from regression analysis⁹ are also included for comparison.

System One⁹				
	Equidistant		Non-equidistant	
	Six λ 's	Eleven λ 's	Six λ 's	Eleven λ 's
Trapezoidal	1.2496 (0.0210)	0.3270 (0.0152)	1.1390 (0.0212)	0.3073 (0.0150)
Cubic Spline	0.4686 (0.0198)	0.0752 (0.0152)	-0.1138 (0.0218)	-0.0047 (0.0152)
Lagrange/Newton	0.1606 (0.0200)	0.0034 (0.0377)	-0.0208 (0.0212)	0.0006 (0.0151)
Regression ⁹	0.1594 (0.0203)	0.0557 (0.0150)	-0.0201 (0.0212)	0.0161 (0.0146)
System Two⁹				
	Equidistant		Non-equidistant	
	Six λ 's	Eleven λ 's	Six λ 's	Eleven λ 's
Trapezoidal	-1.8944 (0.0232)	-0.5077 (0.0161)	-1.4944 (0.0231)	-0.4094 (0.0154)
Cubic Spline	-0.8228 (0.0212)	-0.1424 (0.0159)	0.2035 (0.0234)	0.0093 (0.0156)
Lagrange/Newton	-0.3495 (0.0210)	-0.0053 (0.0335)	0.0467 (0.0227)	-0.0006 (0.0154)
Regression ⁹	-0.3501 (0.0209)	-0.1256 (0.0150)	0.0473 (0.0218)	-0.0345 (0.0148)

Table 2

Free energy biases ($\Delta F_{\text{ref}} - \Delta F_{\text{estimate}}$) for BUQ solvation using eleven equidistant and non-equidistant λ values in kJ/mol. The reference free energy estimates ($\Delta F_{\text{ref}}^{\text{del}}$ and $\Delta F_{\text{ref}}^{\text{ins}}$) were obtained from a two-stage simulation using soft-core scaling and are utilized as reference values to evaluate the polynomial techniques. The free energy for BUQ deletion is $\Delta F_{\text{ref}}^{\text{del}} = 31.6$ kJ/mol and insertion $\Delta F_{\text{ref}}^{\text{ins}} = -31.5$ kJ/mol. Results obtained from regression analysis⁹ are also included for comparison.

	Equidistant		Non-equidistant	
	Deletion	Insertion	Deletion	Insertion
Trapezoidal	-40.2	39.6	-9.5	9.5
Cubic Spline	-26.8	27.0	1.0	-1.1
Lagrange/Newton	-14.0	14.5	-0.4	0.4
Regression ⁹	-14.3	14.5	-0.4	0.3