Gaussian Processes in Molecular Dynamics

Electronic Structure Discussion Group

July 27, 2005

Sebastian Ahnert (TCM)

Gábor Csányi (TCM)
Risi Kondor (Columbia)

Molecular Dynamics

In order to do Molecular Dynamics, we need to (at least approximately) calculate the force on each atom due to all other atoms. This is the gradient of the total energy of the system with respect to the coordinates of the atom in question.

Empirical Potentials
Pro: $O(N)$, parallelizable
Con: inaccurate, non-transferable, hard to build

Tight-Binding Methods
Pro:accurate, much faster than DFT
Con:$O(N^3)$, not parallelizable, non-transferable

DFT
Pro:very accurate, transferable
Con:$O(N^3)$ and very slow

Our aim

Our aim is to develop a universal method for constructing Empirical Potentials by fitting a nearest neighbour energy potential to QM data.

Pro:
$O(N)$
parallelizable
transferable
easy to build
maximum accuracy*

Con:
Nearest neighbour
Size of phase space

*within nearest neighbour approximation

Empirial Potentials

Task: fit parametrized energy potential function empirically.

This is an “art” (Brenner), which involves

some level of intuitive chemical insight, considerable trial-and-error, and significant tenacity

E.g. Stillinger-Weber potential:

$$\Phi_{SW}(\mathbf{r_1}, \ldots, \mathbf{r_N}) = \sum_{\substack{i,j \\ j>i}}^{N} f_2(\frac{r_{ij}}{\sigma}) + \sum_{\substack{i,j,k \\ k>j>i}} f_3(\frac{\mathbf{r}_i}{\sigma}, \frac{\mathbf{r}_j}{\sigma}, \frac{\mathbf{r}_k}{\sigma})$$

where

$$\begin{aligned} f_2(r) &= A(Br^{-p} - r^{-q})e^{\frac{1}{r-a}} \\ f_3(\mathbf{r}_i, \mathbf{r}_j, \mathbf{r}_k) &= \sum_{\pi(i,j,k)} \lambda e^{\frac{\gamma}{r_{ij}-a} + \frac{\gamma}{r_{ik}-a}} (\cos\theta_{jik} + k)^2 \end{aligned}$$

Tight-Binding Methods

In the Tight-Binding scheme the total energy is given by:

$$E_{tot} = E_{coh} + E_{rep}$$

where the cohesive part is calculated using an approximate Hamiltonian matrix $\mathbf{H}$:

$$E_{coh} = \mathrm{tr}(\rho \mathbf{H})$$

The elements of $\mathbf{H}$ only depend on a parametrized function of the internuclear distance.

Local Tight-Binding energies (Csányi)

Recall and rewrite E_{coh}:

$$E_{coh} = \mathrm{tr}(\rho\mathbf{H}) = \sum_{i=1}^{N_{orb}} (\rho\mathbf{H})_{ii}$$

where N_{orb} is the number of orbitals. Now we introduce ϵ_j such that:

$$\epsilon_j = \sum_{k_j} (\rho\mathbf{H})_{k_j k_j}$$

where k_j runs over the orbitals of atom j. Thus we can write:

$$E = \sum_{j=1}^{N_{atoms}} \epsilon_j$$

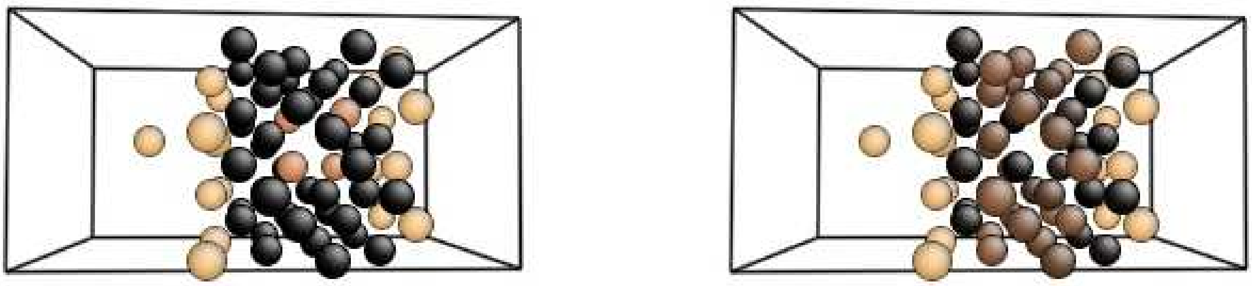
so that ϵ_j is a form of local energy.

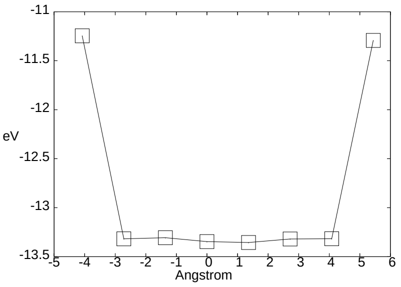
How local are these energies?

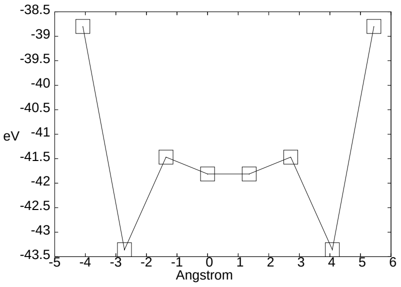
These local energies satisfy the strong locality assumption [Csányi]:

$$\nabla^n_{\mathbf{x}_j}\nabla_{\mathbf{x}_i}E \to 0 \;\; as \;\; |\mathbf{x}_i - \mathbf{x}_j| \to 0 \;\; \forall n, i \neq j$$

But this depends on TB model:



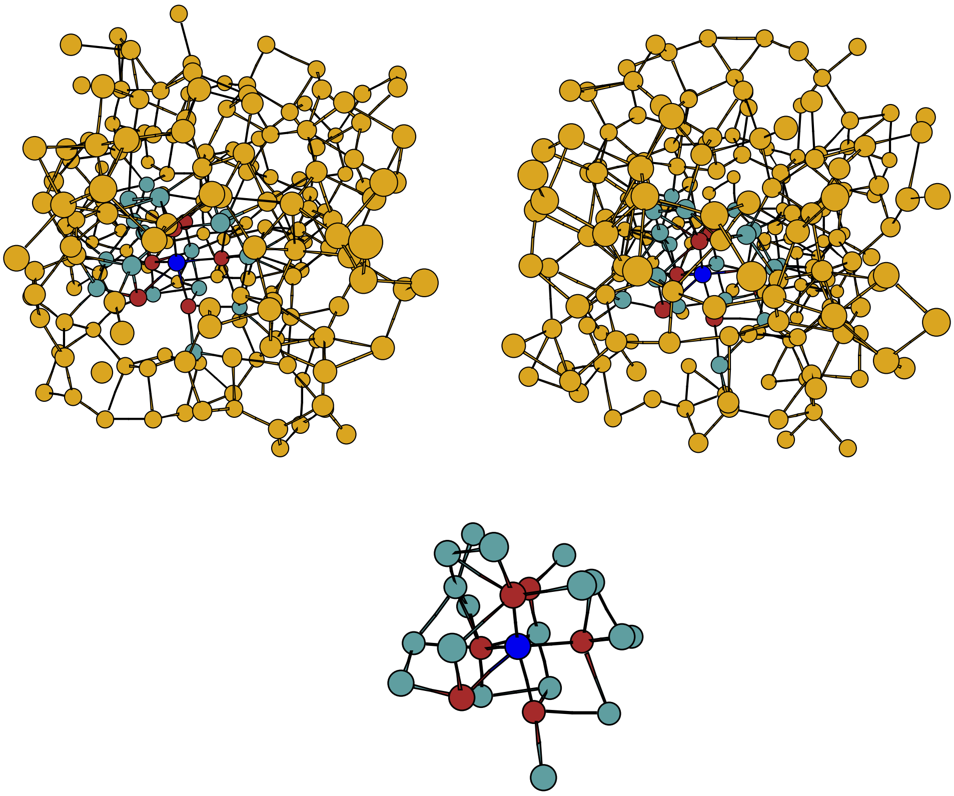




Nearest Neighbour Localization Errors

How dependent are these local energies on the neighbourhood?

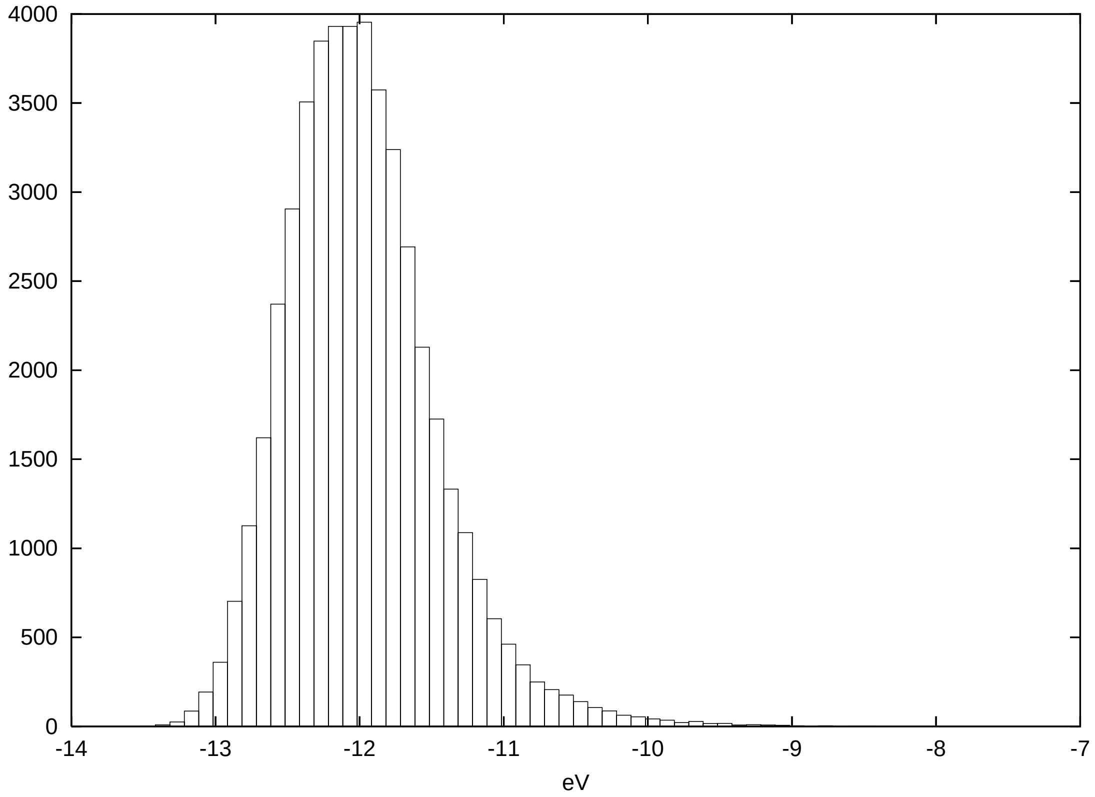
Keep given neighbourhood fixed, allow the rest to evolve freely:



Results: Keeping nearest four neighbours fixed the local energy varies less than 2% ($\approx 0.2\mathrm{eV}$).

Distribution of Local Energies

Histogram of local TB energies collected from configurations sampled during a 5000K Stillinger-Weber run in Silicon.



Gaussian Processes

Gaussian Processes allow us to estimate the value of a function $y(\mathbf{x})$ at a given point $\mathbf{x}_{N+1}$, based on a set of values $\{y_i\}$ of the function at other points $\{\mathbf{x}_i\}$, $i = 1..N$.

This is done by first expanding the function $y(\mathbf{x})$ in terms of a set of basis functions $\phi_h(\mathbf{x})$ with parameters $\{w_h\}$:

$$y(\mathbf{x}; \mathbf{w}) = \sum_{h=1}^{H} w_h \phi_h(\mathbf{x})$$

These basis functions can but do not have to be Gaussian. The reason why the Gaussian process is called Gaussian is because we place assume a Gaussian prior on the distribution of $\{w_h\}$.

Gaussian Processes (II)

We can then use Bayesian estimation to obtain an estimate

$$\hat{y}_{N+1} = \mathbf{k}^T \mathbf{C}_N^{-1} \mathbf{y}_N$$

of the value of the function at position $\mathbf{x}_{N+1}$, where $\mathbf{k}$ is a vector of basis function dot products given by

$$k_j \propto \int_H dh\, \phi_h(\mathbf{x}_{N+1}) \phi_h(\mathbf{x}_j)$$

with $j = 1..N$, and $\mathbf{C}_N$ is a covariance matrix given by

$$(\mathbf{C}_N)_{ij} \propto \int_H dh\, \phi_h(\mathbf{x}_i) \phi_h(\mathbf{x}_j)$$

with $i, j = 1..N$ and the vector $\mathbf{y}_N$ is given by

$$y_k = y(\mathbf{x}_k)$$

with $k = 1..N$.

Gaussian Processes (III)

Gaussian Processes are particularly useful as they also produce a variance for every estimate:

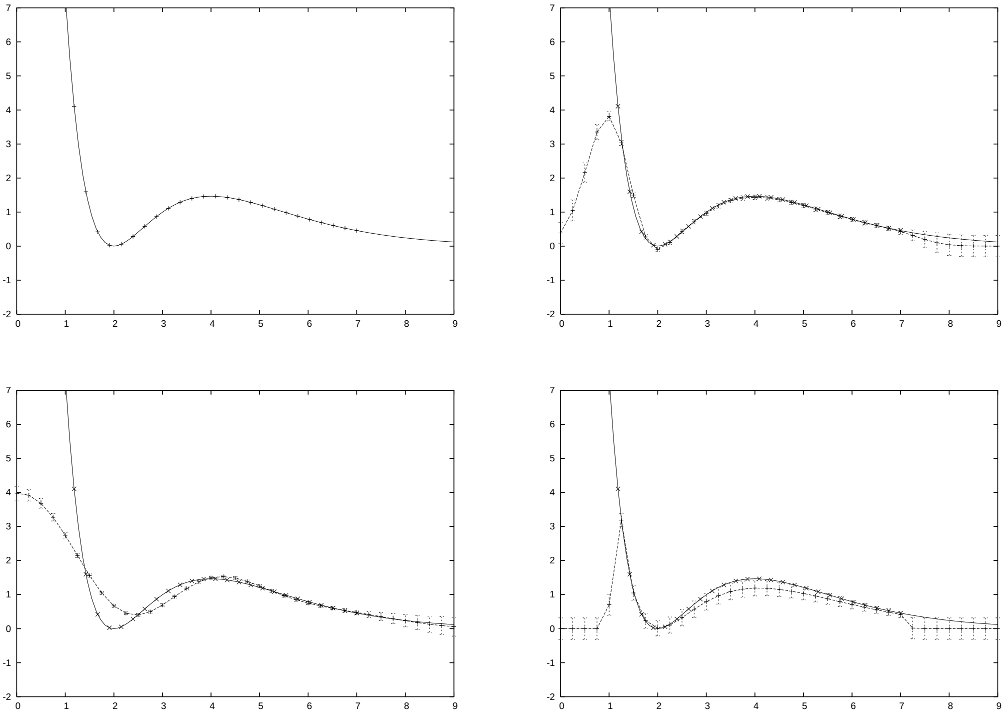
$$\sigma^2_{\hat{y}_{N+1}} = \kappa - \mathbf{k}^T \mathbf{C}_N^{-1} \mathbf{k}$$

where κ is given by:

$$\kappa \propto \int_H dh \, \phi_h(\mathbf{x}_{N+1}) \phi_h(\mathbf{x}_{N+1})$$

Note that the variance does not depend on $y(\mathbf{x})$ at all, but only on $\{\mathbf{x}_i\}$ for $i = 1..N$.

Basis Functions: Width



Gaussian Basis functions with varying widths:

top left: underlying function
top right: $\sigma = 0.5$
bottom left: $\sigma = 1.5$
bottom right: $\sigma = 0.1$

Basis function has to be narrow enough to capture features of underlying function and wide enough to cover space between 'teaching' points.

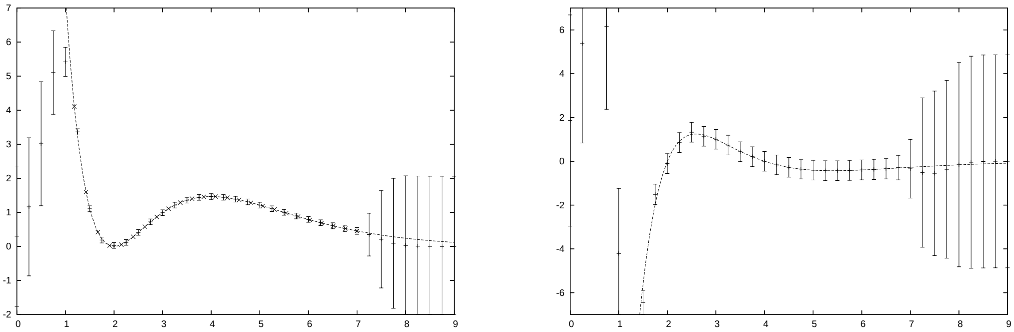
Inferring (From) Derivatives

By using the derivatives of the original basis functions $\phi_h(\mathbf{x})$, we can:

(a) infer a derivative of a function $y(\mathbf{x})$ from a set of values of this function $\{y_k\}$.

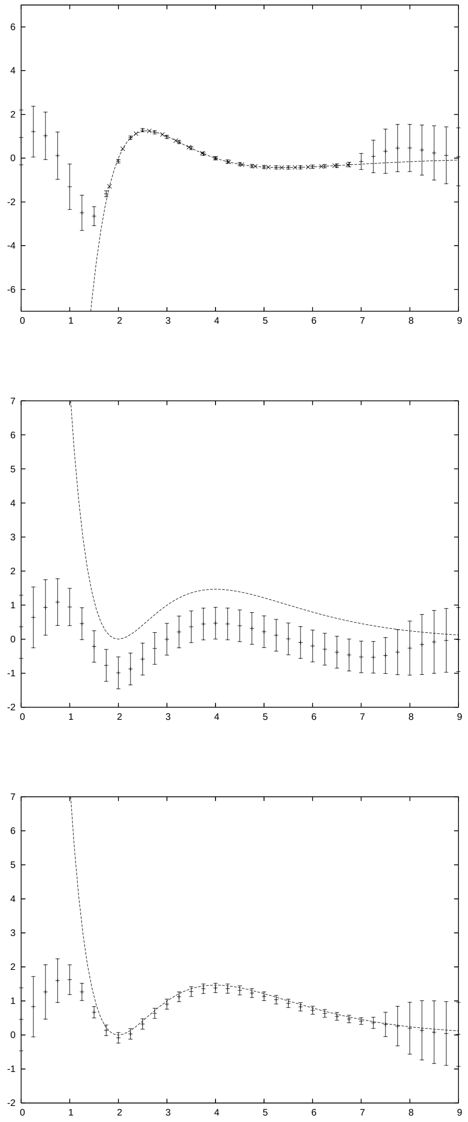
(b) infer a function from values of its derivative, albeit with a constant shift.

The constant shift in (b) can be removed by adding a single value of the function to be inferred to the teaching set.



An inferred derivative

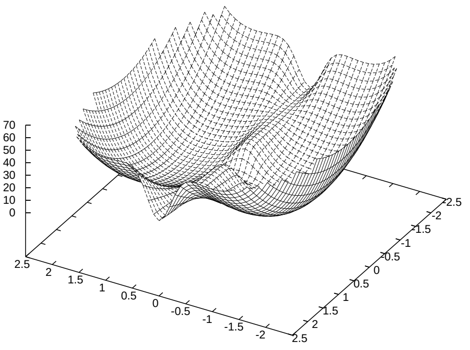
Inferring (From) Derivatives (II)

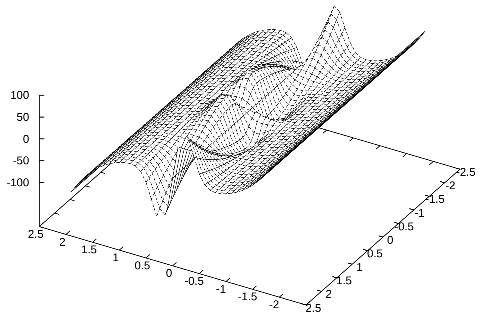


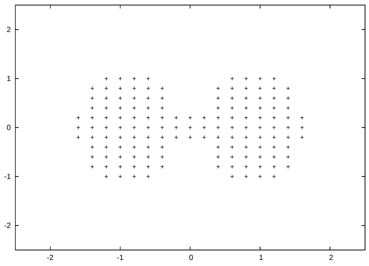
A function inferred from a derivative

Teaching Point Distributions

An example of how to choose teaching point distributions efficiently, for a model 2D double-well potential:

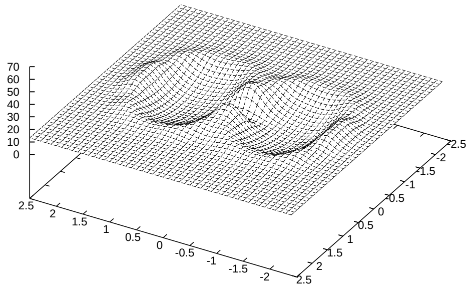


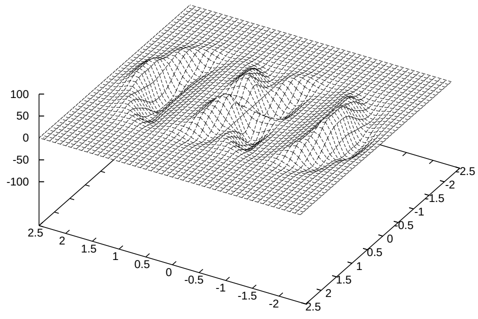


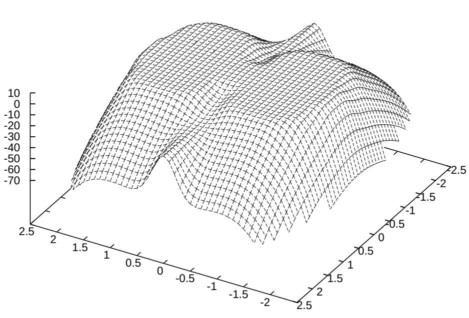


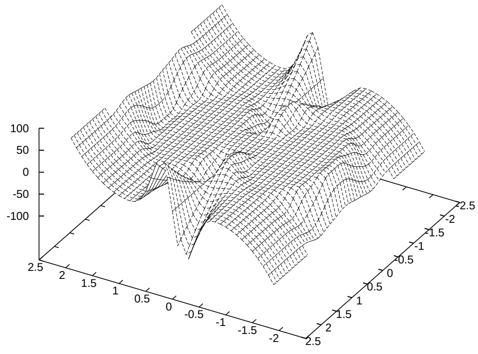
Teaching Point Distributions (II)

The results:









GP in MD: Coordinate System

We want to fit the potential in the configuration space of the four nearest neighbours.

As we expect radial and angular contributions, we use radii and direction cosines of the bonds between the central atom and the four nearest neighbours.

In the space of radii we use **Gaussian** basis functions, in the space of direction cosines we use **Spherical Harmonics** as basis functions.

Using Spherical Harmonics means that the angular part of the basis function dot product is only dependent on the angle between the direction cosines.

Symmetries

Permutation Symmetry

In our configuration space of radii and direction cosines, 24 points are equivalent to one another, as we can relabel the four neighbours and should still get the same energies and forces.

Rotational Symmetry

Our potential function should be invariant under rotation of the entire configuration. Hence pre-rotate all configurations such that the first neighbour lies along the x-axis and the second neighbour lies in the positive x-y plane.

Mirror Symmetry

We can mirror the system in the x-y plane so that the third neighbour always lies in the positive z space.

Implementing Symmetries

The symmetries are implemented inside the basis functions by generating all 24 permutations of neighbour labels, pre-rotating & mirroring the configurations, calculating the dot product and then summing the 24 contributions:

$$\begin{aligned}
\phi(\mathbf{z_1})\phi^T(\mathbf{z_2}) &= \frac{1}{(n!)^2} \sum_{\xi_1,\xi_2\in\Xi} \sqrt{\pi}^n \sigma_z^n \\
&\times \exp\left(\frac{-(|\mathbf{z_1}^{r,\xi_1} - \mathbf{z_2}^{r,\xi_2}|^2)}{4\sigma_z^2}\right) \\
&\times \prod_{k=1}^{n} \sum_{l=0}^{\infty} \frac{2l+1}{4\pi} P_l(\cos\gamma_{k,\mathbf{R}}^{\xi_1,\xi_2})
\end{aligned}$$

$$\cos\gamma_{k,\mathbf{R}}^{\xi_1,\xi_2} = (\mathbf{R_1}\mathbf{z_1}_k^a)^{\xi_1}.(\mathbf{R_2}\mathbf{z_2}_k^a)^{\xi_2}$$

How To Fit A Potential

We can fit a potential by doing the following:

- Run a Stillinger-Weber MD run at high temperature
- At intervals, calculate local Tight-Binding energies as well as Gaussian Process estimates of configurations in the simulation cell
- Compare the two for each configuration: If the difference is greater than a given threshold, teach the configuration

Modelling the Fitting Process for an Exact Potential

If we assume that the physically accessible part of configuration space is filled by randomly distributed smaller 'taught' volumes, we can derive a model in which the number of teaching points obeys:

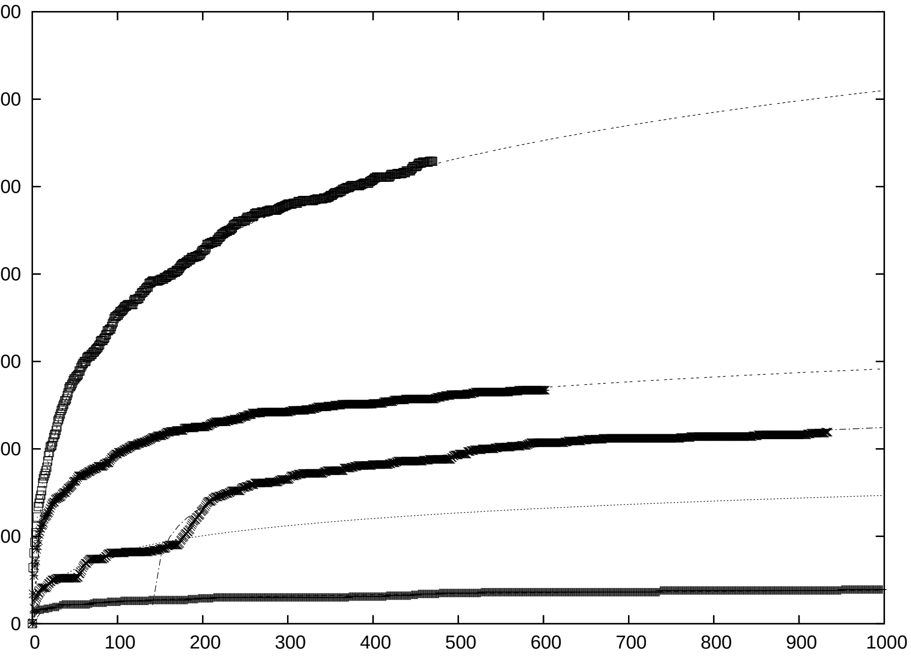
$$N_{TP} = -\frac{\ln\left(1 - xc\ln(1-\frac{v}{V})\right)}{\ln(1-\frac{v}{V})}$$

where x is the number of teaching iterations and $\frac{v}{V}$ is the fraction of the configuration space volume filled by every teaching point.

From this model we can also derive a distribution of the errors for an exact potential with its width as a function of teaching iterations given by:

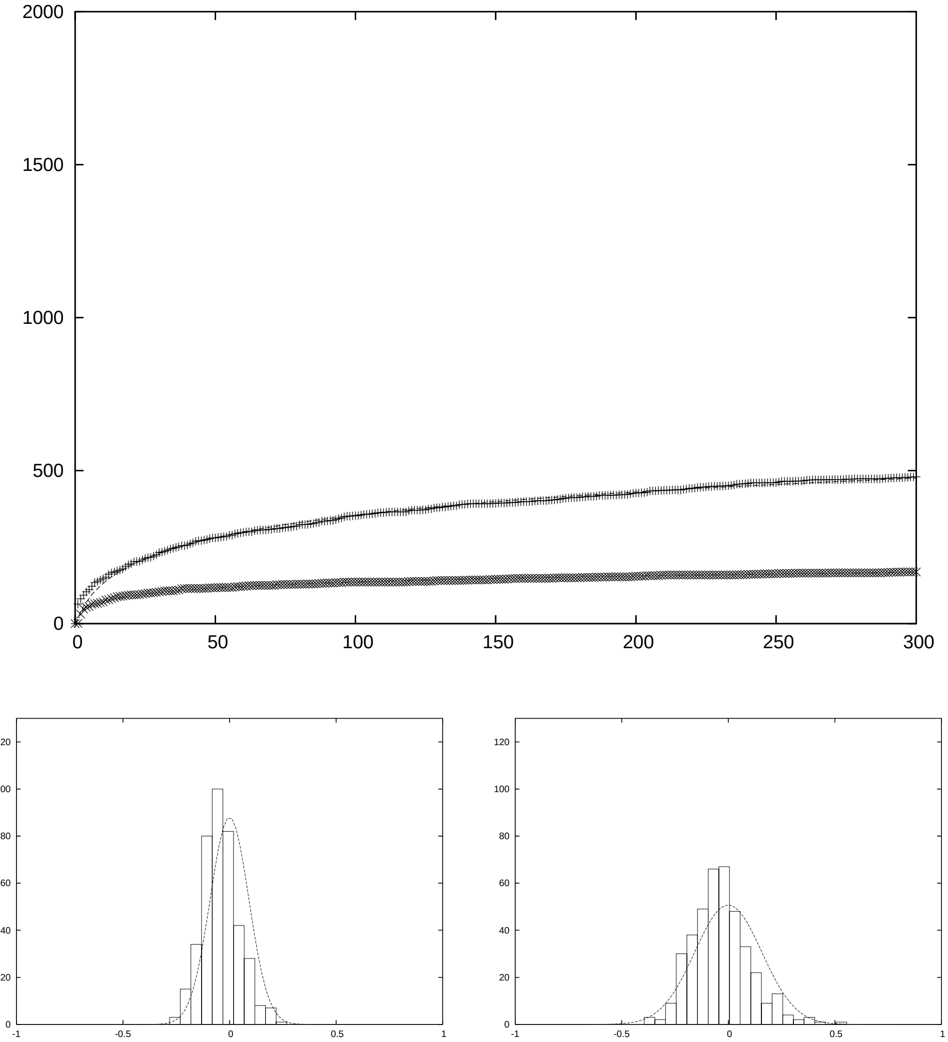
$$\sigma = \left[\frac{\sqrt{2}}{t}\,\mathrm{erfc}^{-1}\left(\frac{c}{N_a[1 - xc\ln(1-\frac{v}{V})]}\right)\right]^{-1}$$

Results (I)



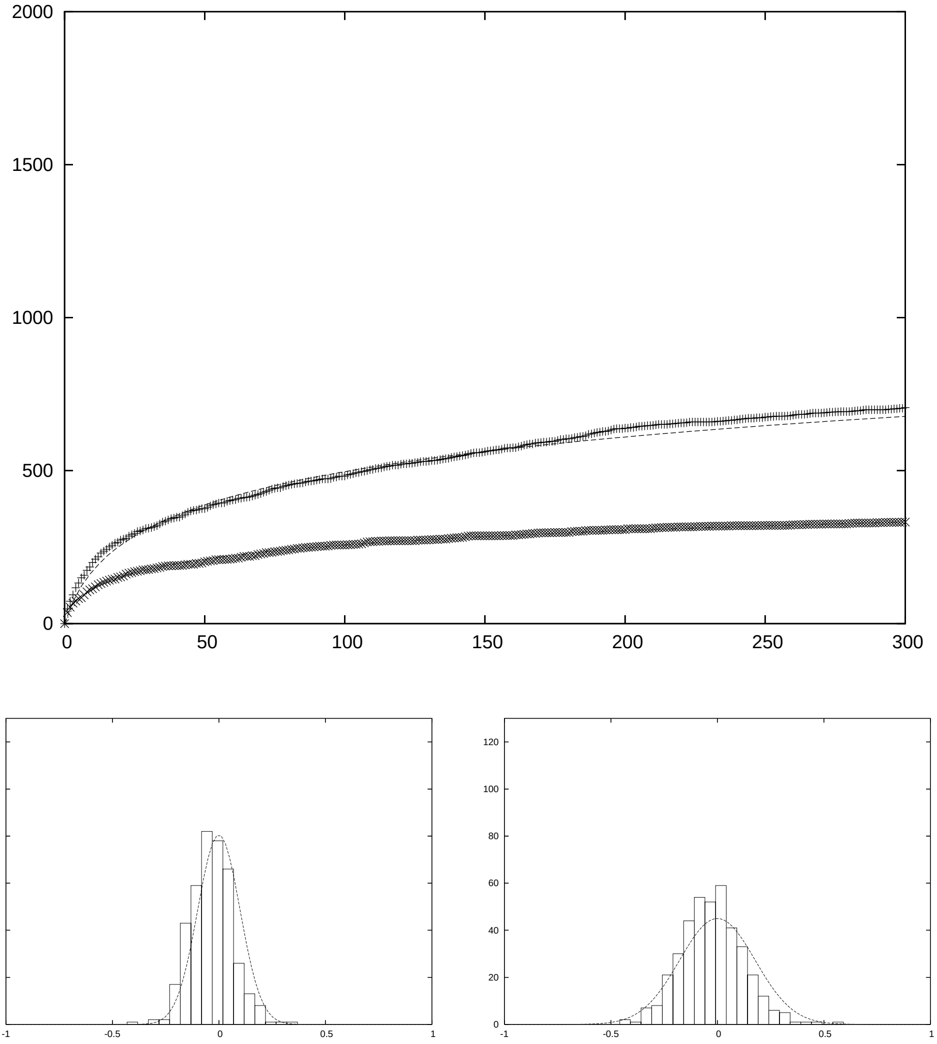
Teaching point number as a function of teaching iterations for bulk density Silicon at 2000K, 3000K, 4000K and 5000K with a 0.25eV teaching threshold.

Results (II)



Bulk Sillicon at 5000K with 0.25eV and 0.5eV thresholds, exact potential

Results (III)



75% *Bulk density Sillicon at 5000K with 0.25eV and 0.5eV thresholds, exact potential*

Results (IV)

t/eV	T	$\frac{\rho}{\rho_{bulk}}$	$\frac{v}{V}$	c
0.25	2000K	1	0.161706	4.83070
0.25	3000K	1	0.0336771	4.41856
0.25	4000K	1	0.0236521	44.7633
0.25	5000K	1	0.00881727	24.9542
0.25	5000K	0.75	0.00586319	29.8565
0.5	5000K	1	0.0341717	32.3293
0.5	5000K	0.75	0.0143356	27.6467

Table of the exact potential model parameters for various thresholds, temperatures and densities

t/eV	$\frac{\rho}{\rho_{bulk}}$	σ_{data}/eV	σ_{theory}/eV
0.25	1	0.0856813	0.09059
0.25	0.75	0.0979888	0.09905
0.5	1	0.135018	0.1574
0.5	0.75	0.148017	0.1772

Table of the error distribution widths of the bare local energy data for 5000K, and the theoretical predictions derived from the exact potential model

Modelling a Non-Exact Potential

A potential function with a finite error, such as the nearest neighbour localization error in Tight-Binding local energies obeys a different equation for the teaching point number, namely:

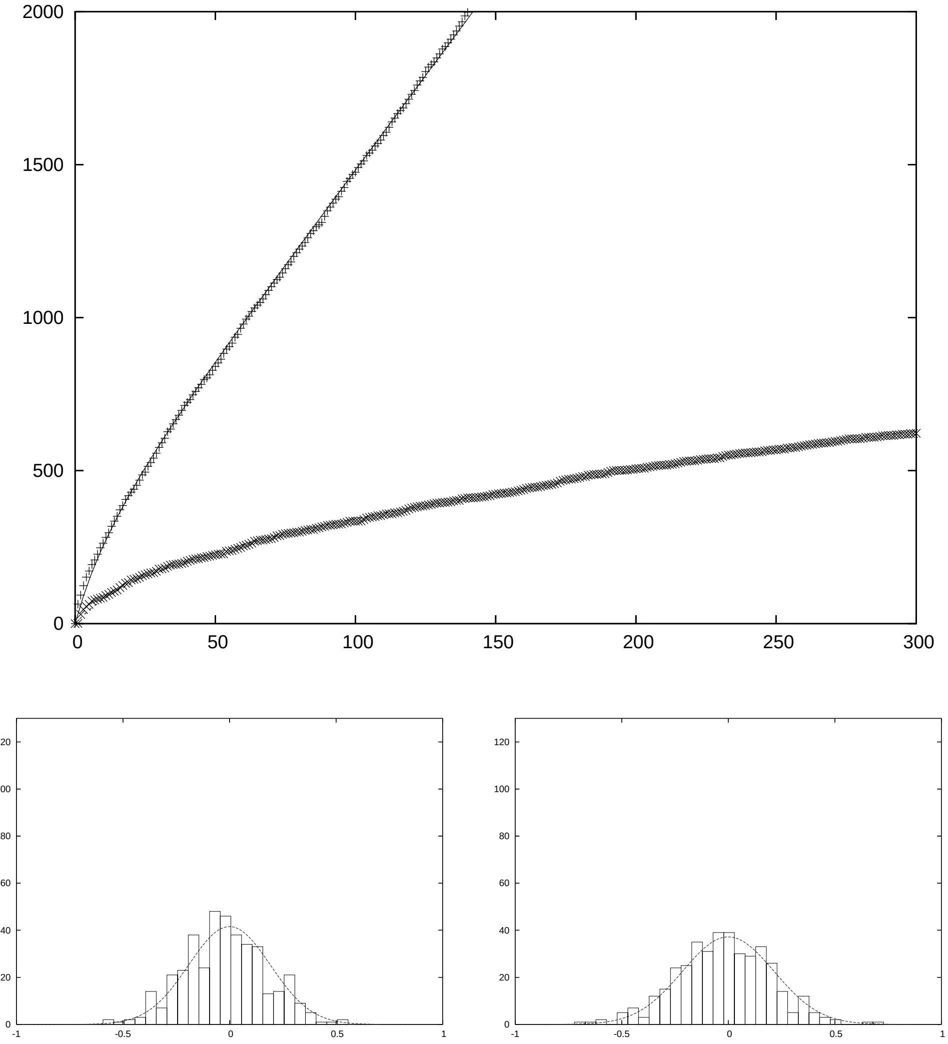
$$N_{TP} = -\frac{\ln[(k+c)(1-\frac{v}{V})^{-kx} - c] + \ln k}{\ln(1-\frac{v}{V})}$$

the gradient of which tends towards k (instead of zero as in the exact case) as $x \to \infty$. The width of the error distribution is given by:

$$\sigma = \left[\frac{\sqrt{2}}{t} \operatorname{erfc}^{-1} \left(\frac{k}{N_a[1 - \frac{c}{k+c}(1-\frac{v}{V})^{kx}]} \right) \right]^{-1}$$

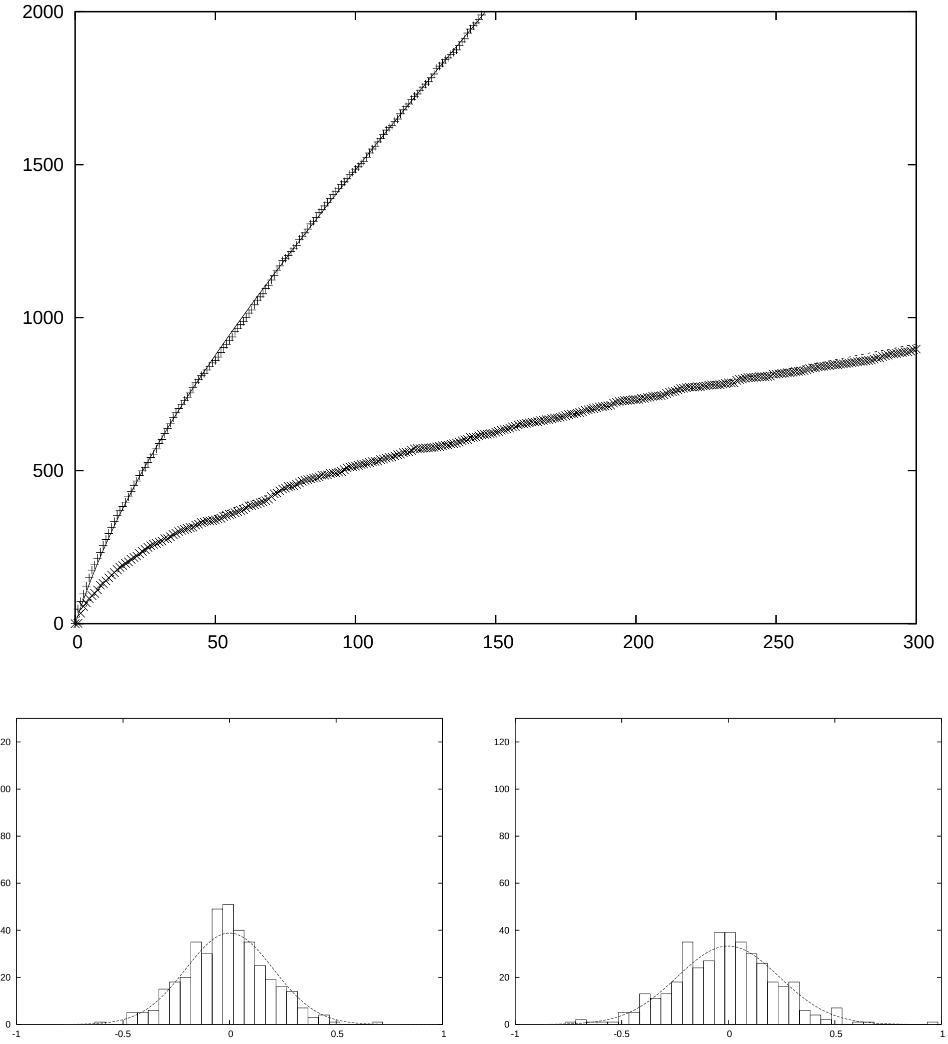
which tends towards $t[\sqrt{2}erfc^{-1}(k/N_a)]^{-1}$ (instead of zero as in the exact case) for $x \to \infty$.

Results (V)



Bulk Sillicon at 5000K with 0.25eV and 0.5eV thresholds, non-exact potential

Results (VI)



75% Bulk density Sillicon at 5000K with 0.25eV and 0.5eV thresholds, non-exact potential

Results (VII)

t/eV	T	$\frac{\rho}{\rho_{bulk}}$	k	c	$\frac{v}{V}$
0.25	5000K	1	12.3267	21.9239	0.004070
0.25	5000K	0.75	10.7345	17.1938	0.002184
0.5	5000K	1	1.19629	9.17308	0.007719
0.5	5000K	0.75	1.66906	15.4847	0.005479

Table of the non-exact potential model parameters for various thresholds, temperatures and densities

t/eV	$\frac{\rho}{\rho_{bulk}}$	σ_{data}/eV	σ_{theory}/eV
0.25	1	0.193061	0.1918
0.25	0.75	0.188321	0.2054
0.5	1	0.215667	0.2145
0.5	0.75	0.235656	0.2396

Table of the error distribution widths of the non-bare local energy data for 5000K, and the theoretical predictions derived from the non-exact potential model

Conclusion

- The task of fitting a potential to an exact potential derived from QM calculations is clearly feasible
- $< 10^3$ points are sufficient to cover the phase space of (near-)bulk Silicon at high temperatures
- This method gives an interesting tool for measuring phase space volumes explored by dynamical systems

Much remains to be done...

Further Work

- Use averaged instead of bare local energies to obtain exact potential
- Use forces (already implemented) to run full GP MD and see how well it does qualitatively and quantitatively for a wide range of systems
- If universal GP potential proves impossible, can still use this fitting approach in the context of LOTF