

CSC / NAG Autumn School on
**Core Algorithms in High-Performance
Scientific Computing**

Case Study II

Ian Bush

Force evaluation in molecular dynamics,
structural optimisation



HECTOR

HIGH END COMPUTING TERASCALE RESOURCE

A Research Councils UK High End Computing Service

Case Study II: Force Evaluation In Molecular Dynamics And Structural Optimisation

Numerical Algorithms Group Ltd, HECTOR CSE



Purpose of The Workshop

This workshop will provide an introduction to the use of two very important libraries

- FFTs
- The BLAS

To illustrate their use we shall employ methods use in *Molecular Modelling* (MM) and *Molecular Dynamics* (MD)

- We shall use the smooth particle mesh Ewald method which use FFTs
- We shall use the conjugate gradient minimization algorithm for BLAS



So What Is Classical MD?

MD attempts to calculate the motion of atoms in a system.

Given some atoms let's assume they interact in a pairwise manner. We can then write the energy as

$$E(\vec{r}_1, \vec{r}_2, \dots, \vec{r}_N) = \sum_{i,j}^N U_{\text{pair}}(|\vec{r}_i - \vec{r}_j|)$$

So we can get the energy, and also the forces F_i by differentiation with respect to the atomic coordinates.

$$F_{ix} = -\frac{\partial E}{\partial r_{ix}} = -\sum_j \frac{\partial U_{\text{pair}}}{\partial r_{ix}}$$



So What Is Classical MD?

Given the forces, we get the acceleration by Newton's 2nd law, and so we can approximately update the position of the atoms by using a Taylor expansion about the current position

$$\mathbf{r}_i(t+\delta t) \approx \mathbf{r}_i(t) + \mathbf{v}_i(t)\delta t + \mathbf{F}_i(t)\delta t^2/2m_i$$

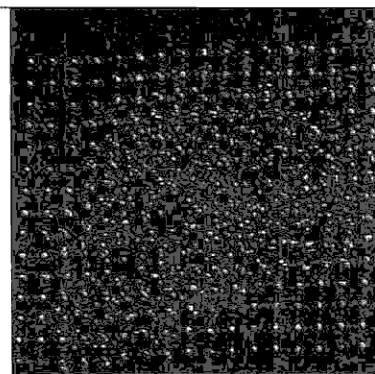
where δt is a small timestep.

(Actually better methods are available. If interested look up the Verlet algorithm)

So classical MD follows the motion of the atoms, from which, using statistical mechanics, we can calculate many properties of the system by appropriate averages.



Classical MD



Pair Potentials

Pair Potentials are usually a sum of a number of terms. Typical terms are

- Exponential $U_{pair} = A_{ij} e^{-\alpha_{ij} r_{ij}}$
 - Inverse powers $U_{pair} = \frac{B_{ij}}{r_{ij}^n}$
 - Coulomb $U_{pair} = \frac{q_i q_j}{r_{ij}}$
- e.g. For NaCl as $q_{Na} = +1$, $q_{Cl} = -1$ we might have

$$U_{NaNa}(r_{ij}) = A_{NaNa} e^{-\alpha_{NaNa} r_{ij}} + \frac{1}{r_{ij}} - \frac{C_{NaNa}}{r_{ij}^6}$$

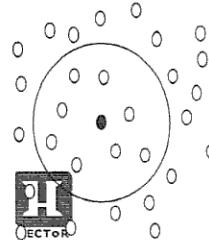
$$U_{ClCl}(r_{ij}) = A_{ClCl} e^{-\alpha_{ClCl} r_{ij}} + \frac{1}{r_{ij}} - \frac{C_{ClCl}}{r_{ij}^6}$$

$$U_{NaCl}(r_{ij}) = A_{NaCl} e^{-\alpha_{NaCl} r_{ij}} - \frac{1}{r_{ij}} - \frac{C_{NaCl}}{r_{ij}^6}$$



Screening

Pair potentials tend to be short ranged, and beyond a certain radius from a given atom we can ignore all interactions, as they are essentially zero – SCREENING
Evaluation of such terms takes $O(N)$ time



Important exception:
Can NOT do this for Coulomb

terms
 $q_i q_j / r_{ij}$



Why Coulomb Sums Are Difficult

Consider a radius σ atom interacting with a uniform "sea" by a $1/r^n$ potential. The energy of interaction is:

$$E = \int_{\sigma}^{\infty} dr \frac{1}{r^n} = 4\pi \int_{\sigma}^{\infty} dr r^2 \frac{1}{r^n}$$

This diverges for $n \leq 3$; such potentials are termed long ranged. The Coulomb potential is $n=1$. The problem is that the volume element grows faster than the potential decays



The Problem: The Coulomb Sum For Periodic Systems

Given a unit cell with lattice vectors $\mathbf{a}_\mu, \mu=1,2,3$ and a set of N charge distributions $\{q_i(\mathbf{r})\}$ in the unit cell, the energy per unit cell is (note the prime!)

$$E = \frac{1}{2} \sum_{n1, n2, n3=-\infty}^{\infty} \left[\sum_{i=1}^N \sum_{j=1}^N \int d\mathbf{r}_1 \int d\mathbf{r}_2 \frac{q_i(\mathbf{r}_1) q_j(\mathbf{r}_2)}{|\mathbf{r}_1 - (\mathbf{r}_2 + \sum_k n_k \mathbf{a}_k)|} \right]$$



Ewald Sum – A "Justification"

The Coulomb potential has problems both at short range and long:

- 1) There is a singularity at $r=0$
- 2) The long range decay is very slow

Let's try splitting it into two to separate these problems:

$$\frac{1}{r} = \phi_{sr}(\mathbf{r}) + \phi_{lr}(\mathbf{r})$$



The Short Range Potential $\phi_{sr}(\mathbf{r})$

- Chosen to be short range in real space
- So can screen
- So can evaluate in $O(N)$ time

$$E_{sr} = \frac{1}{2} \sum_{i=1}^N \int d\mathbf{r}_1 q_i(\mathbf{r}_1) \left[\sum_{n1, n2, n3=-\infty}^{\infty} \sum_{j=1}^N \int d\mathbf{r}_2 \phi_{sr}(\mathbf{r}_1 - (\mathbf{r}_2 + \sum_k n_k \mathbf{a}_k)) q_j(\mathbf{r}_2) \right]$$



The Long Range Potential $\phi_{lr}(\mathbf{r})$

$$E_{lr} = \frac{1}{2} \sum_{i=1}^N \int d\mathbf{r}_1 q_i(\mathbf{r}_1) \left[\sum_{n1, n2, n3=-\infty}^{\infty} \sum_{j=1}^N \int d\mathbf{r}_2 \phi_{lr}(\mathbf{r}_1 - (\mathbf{r}_2 + \sum_k n_k \mathbf{a}_k)) q_j(\mathbf{r}_2) \right]$$

Long range in real space, so can't screen
What can we do?

Let's remind ourselves of a bit of Fourier transform theory

Some Fourier Transform Notation

$$H(f) = \int_{-\infty}^{\infty} h(t) e^{2\pi i f t} dt$$

$$h(t) = \int_{-\infty}^{\infty} H(f) e^{-2\pi i f t} df$$

$$h(t) \Leftrightarrow H(f)$$



Convolution theorem

$$\int_{-\infty}^{\infty} d\tau g(\tau) h(t-\tau) d\tau \Leftrightarrow G(f) H(f)$$

So to evaluate a convolution

- Fourier transform G
- Fourier transform H
- Multiply the transforms together
- Back Fourier transform
- $O(N \log(N))$ instead of $O(N^2)$



How does this help us?

$$E_{lr} = \frac{1}{2} \sum_{i=1}^N \int d\mathbf{r}_1 q_i(\mathbf{r}_1) \left[\sum_{n1, n2, n3=-\infty}^{\infty} \sum_{j=1}^N \int d\mathbf{r}_2 \phi_{lr}(\mathbf{r}_1 - (\mathbf{r}_2 + \sum_k n_k \mathbf{a}_k)) q_j(\mathbf{r}_2) \right]$$

A convolution!

So if we can calculate $\Phi_{lr}(\mathbf{f})$ and $Q(\mathbf{f})$ we can calculate the energy in $O(N)$ time

The Choice of Potential

We must choose how to split the Coulomb potential. By far the most common choice is

$$\phi_{sr}(\mathbf{r}; \beta) = \frac{\text{erfc}(\beta|\mathbf{r}|)}{|\mathbf{r}|} = \frac{1}{|\mathbf{r}|} \frac{2}{\sqrt{\pi}} \int_{\beta|\mathbf{r}|}^{\infty} dt e^{-t^2}$$

$$\phi_{lr}(\mathbf{r}; \beta) = \frac{\text{erf}(\beta|\mathbf{r}|)}{|\mathbf{r}|} = \frac{1}{|\mathbf{r}|} \frac{2}{\sqrt{\pi}} \int_0^{\beta|\mathbf{r}|} dt e^{-t^2}$$

where β is (in principle) arbitrary. In Fourier space

$$\Phi_{lr}(f; \beta) = \frac{1}{\pi} \frac{e^{-\frac{\pi^2 f^2}{\beta^2}}}{f^2}$$



The Structure Factor

$Q(\mathbf{f})$ is so important it

- Has a special name – the Structure Factor
- And is usually represented $S(\mathbf{f})$
- You can measure it in experiments! (almost...)

So if we can evaluate the structure factor

$$S(f) = \int d\mathbf{r} \sum_j q_j(\mathbf{r}) e^{2\pi i \mathbf{f} \cdot \mathbf{r}}$$

in $O(N)$ time, we can evaluate E_{lr} in $O(N)$ time.

Classical MD

To make progress let's consider a specific set of charge distributions

$$q_i(\mathbf{r}) = q_i \delta(\mathbf{r} - \mathbf{r}_i)$$

i.e. point charges. This is the case for classical MD; \mathbf{r}_i is the position of atom i.



The Structure Factor for Point Charges

So

$$S(\mathbf{f}) = \sum_j \int d\mathbf{r} q_j(\mathbf{r}) e^{2\pi i \mathbf{f} \cdot \mathbf{r}} = \sum_j q_j e^{2\pi i \mathbf{f} \cdot \mathbf{r}_j}$$

This would be a 3D discrete Fourier transform if the atoms were so kind as to lie on a regular grid within the unit cell.

But they very, very rarely do...

So this looks like it will take $O(N^2)$ time



Factorising And Scaling

To make progress let's follow Darden *et al.* and factorise the structure factor and scale the coordinates:

$$S(\mathbf{f}(m_1, m_2, m_3)) = \prod_{\mu=1}^3 \sum_j q_j e^{2\pi i \frac{m_\mu}{K_\mu} u_{j\mu}}$$

where $u_{j\mu} = K_\mu \mathbf{b}_\mu \cdot \mathbf{r}_j$

Note $0 \leq u_\mu < K_\mu$

Now we need only deal with a 1D problem



What I Want

It would be nice if we could write

$$e^{2\pi i \frac{m_\mu}{K_\mu} u} \simeq c(m_\mu) \sum_{\kappa=-\infty}^{\infty} T(u - \kappa) e^{2\pi i \frac{m_\mu}{K_\mu} \kappa}$$

The complex exponentials are now only evaluated on a regular grid And we can think about using a fast fourier transform. Then

$$S(\mathbf{f}) \simeq \left[\prod_{\mu=1}^3 c(m_\mu) \right] \sum_{\kappa_1=-\infty}^{\infty} \sum_{\kappa_2=-\infty}^{\infty} \sum_{\kappa_3=-\infty}^{\infty} e^{2\pi i \sum_{\mu=1}^3 \frac{m_\mu}{K_\mu} \kappa_\mu} \sum_j q_j \prod_{\mu=1}^3 T(u_{j\mu} - \kappa_\mu)$$



What I Want

Still following Darden *et al.*, after a bit of maths one gets

$$S(\mathbf{f}) \simeq \left[\prod_{\mu=1}^3 c(m_\mu) \right] \sum_{\kappa_1=0}^{K_1-1} \sum_{\kappa_2=0}^{K_2-1} \sum_{\kappa_3=0}^{K_3-1} e^{2\pi i \sum_{\mu=1}^3 \frac{m_\mu}{K_\mu} \kappa_\mu} \left[\sum_j q_j \sum_{n_1, n_2, n_3} \prod_{\mu=1}^3 T(u_{j\mu} - \kappa_\mu - n_\mu K_\mu) \right]$$

which *is* a discrete three dimensional Fourier Transform.

So we can now use our hyper efficient FFT technology to evaluate the energy



Cardinal B-Splines

The best choice for $T(u)$ known currently are the n^{th} order Cardinal B-Splines, $M_n(u)$

They are

- 1) Short ranged; $M_n(u)$ is non-zero only for $0 \leq u \leq n$
- 2) Analytic, being differentiable $n-2$ times. Other methods are only piecewise differentiable.

This choice is known as the

Smooth Particle-Mesh Ewald (SPME) algorithm



Complex Exponentials and Cardinal B-Splines

$$e^{2\pi i \frac{m}{K} u} \simeq b(m) \sum_{\kappa = \text{int}(u) - n + 1}^{\text{int}(u)} M_n(u - \kappa) e^{2\pi i \frac{m}{K} \kappa}$$

$$b(m) = \frac{e^{2\pi i (n-1) \frac{m}{K}}}{\sum_{\kappa=0}^{n-2} M_n(\kappa+1) e^{2\pi i \frac{m}{K} \kappa}}$$



The SPME Algorithm Step by Step

Firstly we need to calculate

$$w(r) = \sum_j \sum_{n_1=-\infty}^{\infty} \sum_{n_2=-\infty}^{\infty} \sum_{n_3=-\infty}^{\infty} q_j \prod_{\mu=1}^3 M_n(u_{j\mu} - \kappa_{\mu} - n_{\mu} K_{\mu})$$

Note also that each atom contributes only to a small portion of the grid **near that atom**. It therefore takes $O(N)$ time.



The SPME Algorithm Step by Step

Next we Fourier transform using a 3D FFT - $O(N \log(N))$

$$W(f) = \sum_{\kappa_1=0}^{K_1-1} \sum_{\kappa_2=0}^{K_2-1} \sum_{\kappa_3=0}^{K_3-1} e^{2\pi i \sum_{\mu=1}^3 \frac{m_{\mu}}{K_{\mu}} \kappa_{\mu}} w(f)$$



The SPME Algorithm Step by Step

Now, in a very slight variation on what has gone before - $O(N)$

$$H^{\dagger}(f; \beta) = \left[\prod_{\mu=1}^3 |b(m_{\mu})|^2 \right] \frac{1}{\pi} \frac{e^{-\frac{\pi^2 f^2}{\beta^2}}}{f^2} W^{\dagger}(f)$$

This is a local filter - $H(f)$ depends only on the same frequency in $W(f)$



The SPME Algorithm Step by Step

Then we Fourier transform back, again using a FFT - $O(N \log(N))$

$$h(r) = \sum_{m_1=0}^{K_1-1} \sum_{m_2=0}^{K_2-1} \sum_{m_3=0}^{K_3-1} e^{-2\pi i \sum_{\mu=1}^3 \frac{m_{\mu}}{K_{\mu}} \kappa_{\mu}} H(f)$$



The SPME Algorithm Step by Step

Now we can evaluate the energy

$$E = \frac{1}{2V} \int d\mathbf{r} h(\mathbf{r}) w(\mathbf{r})$$

This is simply a sum over the grid - $O(N)$
So the whole algorithm is $O(N \log(N))$, which is much faster than $O(N^2)$



SPME Algorithm - summary

- By approximating the complex exponentials we can base our algorithms on FFTs
- This reduces the scaling from $O(N^2)$ to $O(N \log(N))$ (slight lie ...)
- Which saves huge amounts of computer time
- For more details see
Darden *et al.*, J.Chem.Phys. 103 19 (1995)
Bush *et al.*, Comp.Phys.Comm. 175 5 (2006)



Using SPME

- Once we have got the SPME algorithm working we shall use it for some calculations on Sodium Chloride (common salt)
- Sodium Chloride is ionic, so coulomb interactions are important
We shall use the conjugate gradient method to find the minimum energy (or ground state) structure
 - i.e. we move the atoms around until we have minimised the energy
 - This is a simple example of Molecular Modelling which is used in drug design



The Conjugate Gradient Optimisation Method

- Closely related to the Conjugate Gradient Method we learnt for solving linear equations
- We want to find the minimum value of a function. Let's assume we are near the minimum so we can write

$$f(x) \approx c - b \cdot x + \frac{1}{2} x^T A x$$

- At the minimum the derivative is zero, and **A** is positive definite
- So differentiating once and rearranging we get at the minimum ... $Ax = b$



- But we can use a clever algorithm to avoid finding **A** explicitly!

The Conjugate Gradient Optimisation Method

To minimise our a function **f** starting at a point **P**₀

- Set $h_0 = g_0 = -\nabla f(P_0)$
- For $i = 0, 1, 2, \dots$
- Minimize along **h**_i to find point **P**_{i+1}
- $g_{i+1} = -\nabla f(P_{i+1})$
- If $|g_{i+1}| < \text{tolerance}$ the converged
- $\gamma_i = \frac{g_{i+1} \cdot g_{i+1}}{g_i \cdot g_i}$ (Fletcher-Reeves variant)



$$h_{i+1} = g_{i+1} + \gamma_i h_i$$



The Conjugate Gradient Optimisation Method

So all we need to be able to do is

- 1) Evaluate the function
- 2) Evaluate the derivative of the function
- 3) Perform linear algebra operations – e.g. dot products and vector addition – BLAS !

The function here is simply the energy expressed as a pair potential, and we will provide routines to evaluate that and its derivative.



More Information on CG

For more information on the CG algorithm have a look at

- Numerical recipes in C, chapter 10, section 6: Conjugate Gradient Methods in Multidimensions, Press *et al.*, Cambridge University Press (but don't look at the code!)
- And references therein
-



In Summary

- Lot's of stuff covered here which is really used day to day by people !
- This afternoon you will
 - 1) Call and validate a simple 1D FFT
 - 2) Interface a SPME-base molecular dynamics code to a FFT library
 - 3) Complete a CG optimiser and use this to identify the ground state structure of Sodium Chloride from a starting structure

