

## Towards Linear Scaling MD COMP3320/6464

### Code Kernel

```
for (i=1; i<total_atoms; i++)
  for (j=0; j<i; j++)
    r2 = (at[i].x - at[j].x)^2 + (at[i].y - at[j].y)^2 +
          (at[i].z - at[j].z)^2
    PE += 1.0/r2^6 - 2.0/r2^3
    f_x = (12.0/r2^7 - 12.0/r2^4) (at[i].x - at[j].x)
    f_y = (12.0/r2^7 - 12.0/r2^4) (at[i].y - at[j].y)
    f_z = (12.0/r2^7 - 12.0/r2^4) (at[i].z - at[j].z)
    at[i].fx+=f_x
    at[i].fy+=f_y
    at[i].fz+=f_z
    at[j].fx-=f_x
    at[j].fy-=f_y
    at[j].fz-=f_z
  next j
next i
```

## Code Kernel

```
for (i=1; i<total_atoms; i++)
  for (j=0; j<i; j++)
    r2 = (at[i].x - at[j].x)^2 + (at[i].y - at[j].y)^2 +
          (at[i].z - at[j].z)^2
    PE += 1.0/r2^6 - 2.0/r2^3
    f_x = (12.0/r2^7 - 12.0/r2^4) (at[i].x - at[j].x)
    f_y = (12.0/r2^7 - 12.0/r2^4) (at[i].y - at[j].y)
    f_z = (12.0/r2^7 - 12.0/r2^4) (at[i].z - at[j].z)
    at[i].fx+=f_x
    at[i].fy+=f_y
    at[i].fz+=f_z
    at[j].fx-=f_x
    at[j].fy-=f_y
    at[j].fz-=f_z
  next j
next i
```

How many CPU cycles per loop?

## Typical Performance (Iwaki)

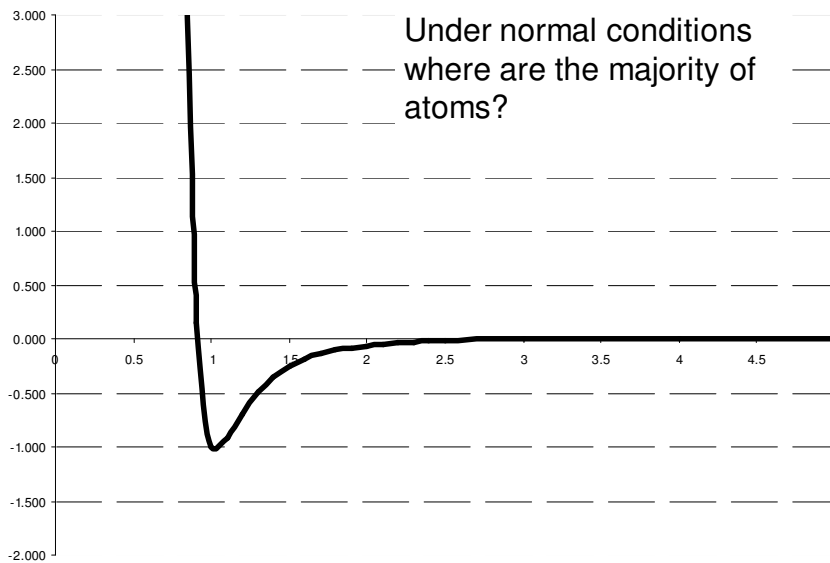
Atoms	Time (sec)
125	0
1000	0
3375	0
8000	3
15625	11
27000	34
42875	87
64000	193
91125	391
125000	736
166375	1304
216000	2198
274625	3553

- To simulate 300,000 atoms it takes about 1 hour to evaluate the energy and forces ....once!
- Is this consistent (sanity check)
  - Estimated 50cycles per loop iteration
  - $50 * 300,000 * 300,000 / 2 / 1\text{GHz} = 2250\text{s}$
  - Approximately right!
- 1 hour per energy/force
  - We can't afford to do too many of these!

## Real Simulations

- Timestep corresponds to  $10^{-15}$  seconds (Femto-second)
  - Determined by vibrational frequency of atoms involved (typically hydrogens have highest frequency bonds)
- Protein folding occurs in  $10^{-6}$  seconds or longer
  - Implies  $10^{-6}/10^{-15} = 10^9$  timesteps
  - Implies > 100,000 years on 1 CPU computer if each energy/gradient evaluation takes 1 hour!

## Back to the Potential!



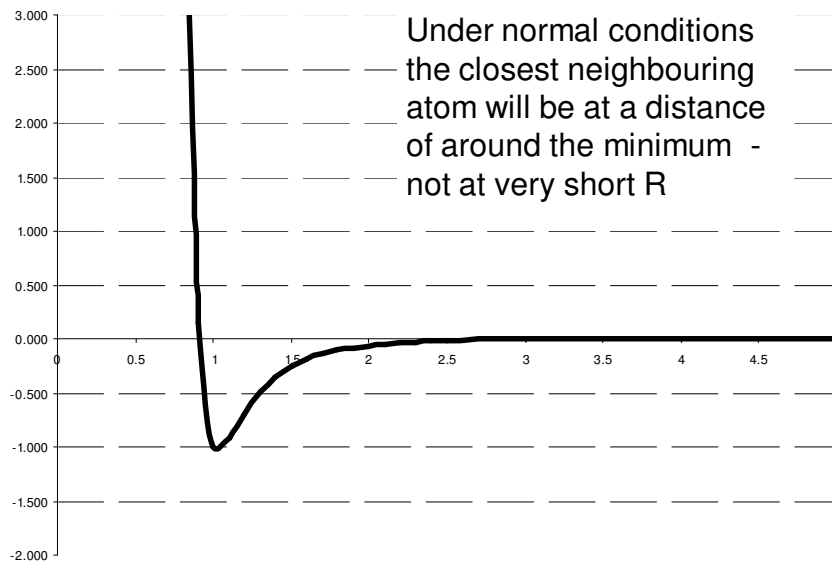
## Back to the Potential!

Minimum at  $r_{\min} = 1.0$

For particles separated by 50 units the interaction energy is small  $10^{-10}$

R	1/R <sup>12</sup>	-2/R <sup>6</sup>	TOTAL
0.4	5.96E+04	-4.883E+02	5.912E+04
0.5	4.10E+03	-1.280E+02	3.968E+03
0.6	4.59E+02	-4.287E+01	4.165E+02
0.7	7.22E+01	-1.700E+01	5.525E+01
0.8	1.46E+01	-7.629E+00	6.923E+00
0.9	3.54E+00	-3.763E+00	-2.226E-01
1	1.00E+00	-2.000E+00	-1.000E+00
1.1	3.19E-01	-1.129E+00	-8.103E-01
1.2	1.12E-01	-6.698E-01	-5.576E-01
1.3	4.29E-02	-4.144E-01	-3.714E-01
1.4	1.76E-02	-2.656E-01	-2.480E-01
1.5	7.71E-03	-1.756E-01	-1.679E-01
1.6	3.55E-03	-1.192E-01	-1.157E-01
1.7	1.72E-03	-8.286E-02	-8.114E-02
1.8	8.64E-04	-5.880E-02	-5.794E-02
1.9	4.52E-04	-4.251E-02	-4.206E-02
2	2.44E-04	-3.125E-02	-3.101E-02
2.1	1.36E-04	-2.332E-02	-2.318E-02
2.2	7.78E-05	-1.764E-02	-1.756E-02
2.3	4.56E-05	-1.351E-02	-1.346E-02
2.4	2.74E-05	-1.047E-02	-1.044E-02
2.5	1.68E-05	-8.192E-03	-8.175E-03
3	1.88E-06	-2.743E-03	-2.742E-03
4	5.96E-08	-4.883E-04	-4.882E-04
5	4.10E-09	-1.280E-04	-1.280E-04
10	1.00E-12	-2.000E-06	-2.000E-06
15	7.71E-15	-1.756E-07	-1.756E-07
20	2.44E-16	-3.125E-08	-3.125E-08
50	4.10E-21	-1.280E-10	-1.280E-10

## Our Potential!



## Evaluating the Potential Energy

```

icount=0;
for (i=0; i<natoms; i++){
  for (j=0; j<i; j++){
    PE = PE + Interaction(i,j);
    icount++;
  }
}

```

- A contribution to PE will only occur if  
 $\text{Interaction}(i,j)/PE > \text{Machine\_Precision}$   
 else it will have no effect
  - Why?
  - Is this due to truncation or rounding error?
- How does value of PE change as?

## Variation of PE with icount

- Typical system has input  $R=N$ 
  - i.e. atoms roughly spaced by 1 unit, or at minimum

R/Na	5/5	10/10	15/15	20/20	25/25	30/30	35/35	40/40
Atoms	125	1000	3375	8000	15625	27000	42875	64000
Value of PE								
1.00E-01	32	46	55	55	55	55	55	55
1.00E+00	946	1317	1378	1485	1653	1801	1953	2145
1.00E+01	7750	45264	61060	91378	94830	94770	94360	95227
1.00E+02		499500	3014516	3044259	3244255	3500181	3904615	4388164
1.00E+03			5693625	31996000	122062500	243995095	240384477	242385153
1.00E+04						364486500	919111375	2047968000

## Interaction Energy Distribution

R/Na	5/5	10/10	15/15	20/20	25/25	30/30	35/35	40/40
Atoms	125	1000	3375	8000	15625	27000	42875	64000
<b>Interaction</b>								
1.0E-01	9	9	9	9	9	9	9	9
1.0E-02	80	84	80	77	78	79	79	80
1.0E-03	380	765	807	872	876	873	871	867
1.0E-04	778	4383	8090	7852	7662	7555	7703	7833
1.0E-05	1700	9547	25918	44300	67158	84346	83570	82505
1.0E-06	2734	25439	62513	104317	163895	243443	379573	442099
1.0E-07	1942	61327	168846	307202	494631	698699	818771	1054717
1.0E-08	127	124606	419352	839435	1326361	1895494	2604072	3283569
1.0E-09	0	180371	952292	2122767	3593634	5270487	7255807	9527764
1.0E-10	0	90650	1757298	4881452	9040268	14154078	19850114	26250839
1.0E-11	0	2319	1901207	9300736	20519481	34658943	51373283	70367790
1.0E-12	0	0	396534	11202045	37691917	75111314	121463548	175374832
1.0E-13	0	0	679	3170040	40756939	125098254	242766207	387798903
1.0E-14	0	0	0	14896	8399591	107262926	472507768	1373776193
<b>TOTAL</b>	7750	499500	5693625	31996000	122062500	364486500	919111375	2047968000
<b>CPUs</b>	0	0.04	0.47	2.87	11.37	34.2	86.59	257.58

## Distance Based Cutoff

- If pairs of atoms are separated by a distance greater than some cutoff we ignore the contribution

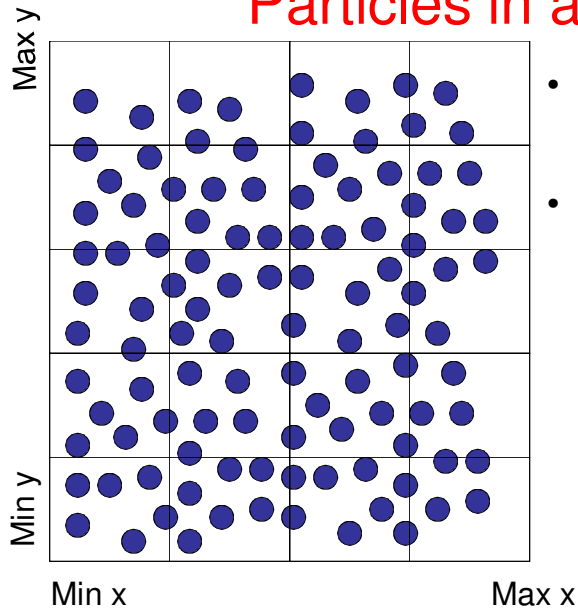
```

icount=0;
for (i=0; i<natoms; i++){
  for (j=0; j<i; j++){
    if (distance(i,j) < cutoff){
      PE = PE + Interaction(i,j);
      icount++;
    }
  }
}

```

- Is this likely to run any better?

## Particles in a Box



- Assign particles to boxes of size cutoff
  - Scales linearly
- Each box will only interact with its direct neighbors
  - 8 other boxes in 2d, 26 in 3d
  - Still have  $r$  based cutoff, but only between boxes
  - Evaluation now scales as  $O(n)$

## Box Based MD Program Walk Through Code

## Performance and box size

Cutoff	PE at Step 0	PE at Step 10	PE at Step 10	CPU(s)
5	-1.980713E+05	-1.980713E+05	6.407134E-04	23
10	-1.989768E+05	-1.989768E+05	6.441947E-04	134
15	-1.990421E+05	-1.990421E+05	6.443475E-04	374
20	-1.990540E+05	-1.990540E+05	6.443699E-04	784
25	-1.990570E+05	-1.990570E+05	6.443751E-04	952
30	-1.990578E+05	-1.990578E+05	6.443766E-04	1115
35	-1.990581E+05	-1.990581E+05	6.443772E-04	1266
40	-1.990582E+05	-1.990582E+05	6.443773E-04	1428
45	-1.990582E+05	-1.990582E+05	6.443774E-04	1573
Orig	-1.990582E+05	-1.990582E+05	6.443774E-04	943

- Box 35/35 10 timesteps
- With 10 unit cutoff
  - Calculation is  $\approx 7$  times faster
  - Energies accurate to about 1%

## Scaling with Problem Size

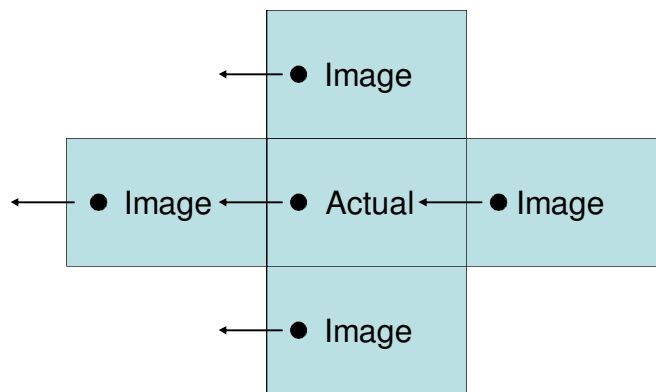
- Fix box size at 20 units and scale problem
  - Do we see linear scaling??

Input Na	Total No Atoms	Measured CPU(s)	Predicted $O(N^2)$	Predicted $O(N)$
20	8000	7.5	N/A	N/A
25	15625	23.7	28.6	14.6
30	27000	62.7	70.7	40.9
35	42875	140.2	158.0	99.5
40	64000	231.6	312.5	209.3
45	91125	375.3	469.5	329.7
50	125000	609.9	706.2	514.9
55	166375	959.1	1080.5	811.8
60	216000	1266.3	1616.6	1245.2
65	274625	1662.6	2047.0	1610.0
70	343000	2272.0	2593.5	2076.5
75	421875	3071.1	3437.0	2794.4

- Yes around  $N_a \approx 60$  but then not so good for  $N_a \approx 70$ 
  - Probably cache effect at  $N_a \approx 70$
  - $6 \times 343000 \times 8 \approx 16\text{MB}$  (x,y,z coord and force array)

## Periodic Boundary Conditions

- To give a more realistic simulation of condensed phases it is usual to implement periodic boundary conditions
  - With cutoffs every particle sees at most one image of every other atom in the system (*minimum image convention* – make box larger than cutoff radius)



## Dynamics Methods: Conclusions

- Methods that integrate Newton's equations of motion are widely used in computational science
  - From simulation of galaxies to simulations of proteins
  - Use of high performance parallel computers is widespread
- Two major challenges are
  - Number of timesteps required
  - Evaluation of long range interaction potentials
- Linear scaling or near linear scaling methods are very important
  - A basic knowledge of numerical representation of floating point data and errors is essential to understanding these methods
- Linear scaling is only possible now that we can do calculations on small clusters sufficiently fast
  - The calculations within each box scale as  $O(n_b^2)$  where  $n_b$  is the number of atoms in the box