

Gromacs Workshop

Spring 2007 @ CSC

Erik Lindahl

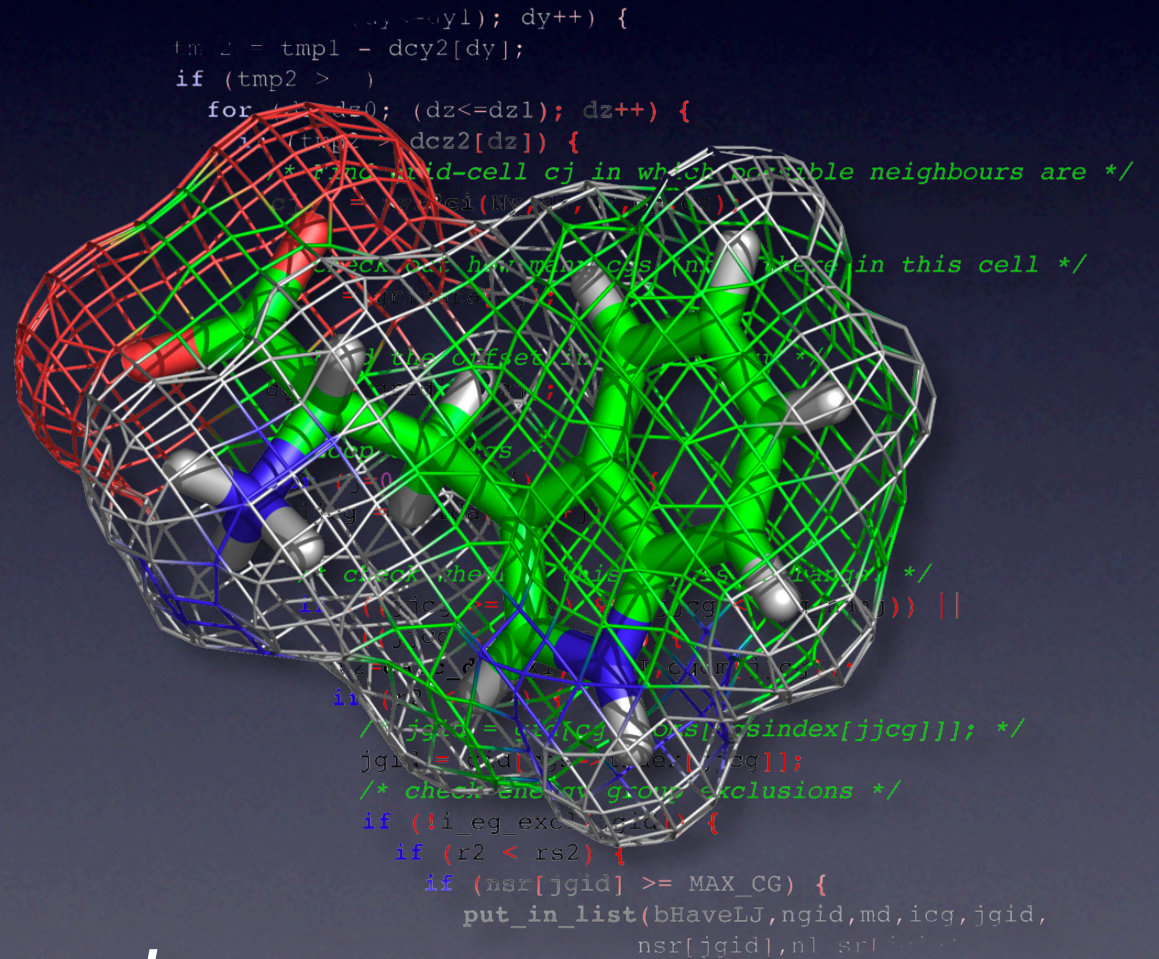
*Center for Biomembrane Research
Stockholm University, Sweden*

David van der Spoel

*Dept. Cell & Molecular Biology
Uppsala University, Sweden*

Berk Hess

*Max-Planck-Institut for Polymer Research
Mainz, Germany*



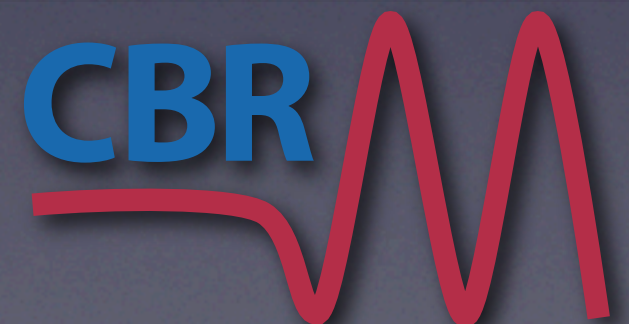
For CSC: Atte Sillanpää

Introduction to Molecular Simulation and Gromacs

Erik Lindahl

lindahl@cbr.su.se

lindahl@gromacs.org



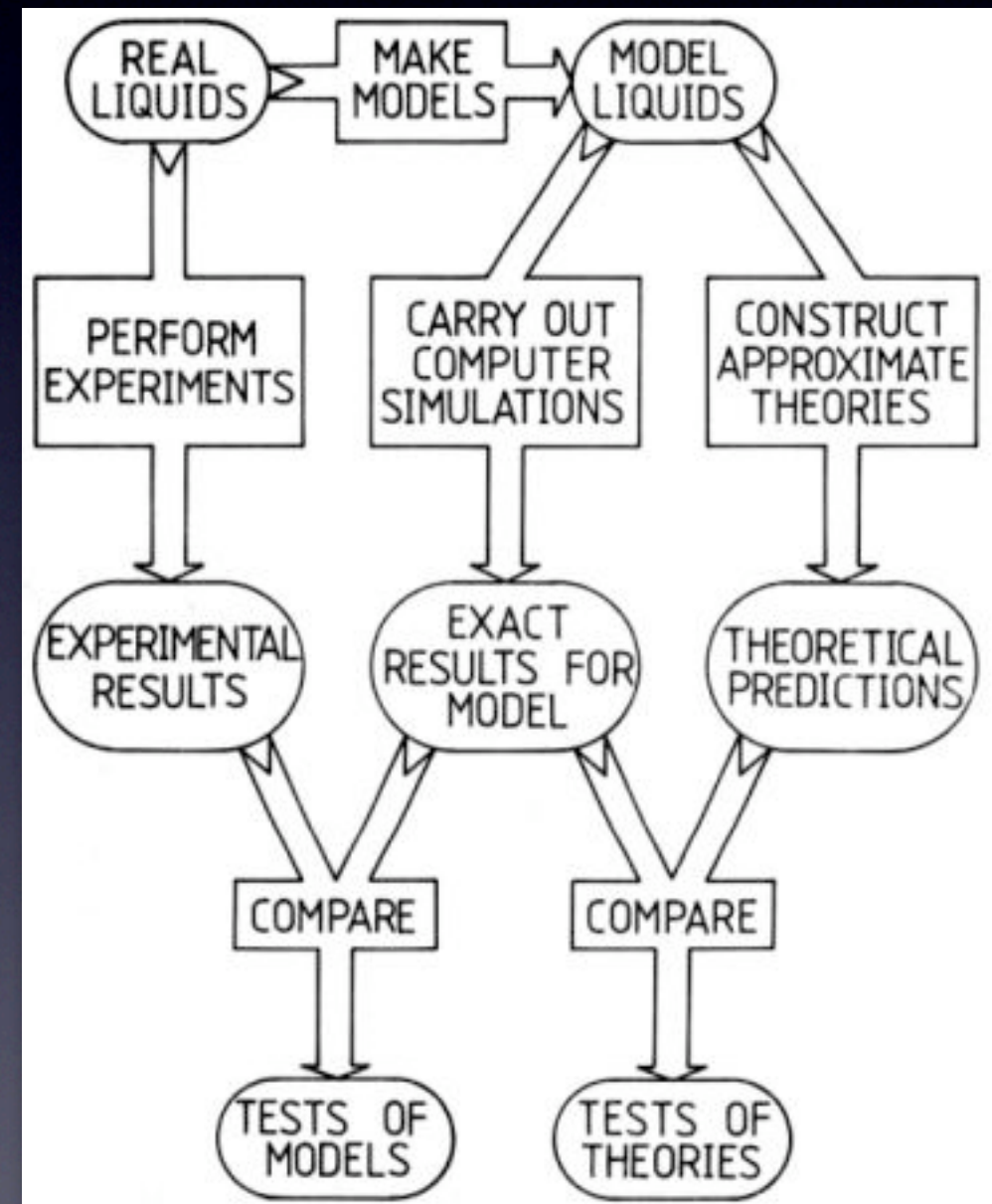
[illegible]

- [illegible]


```

        _dy1); dy++) {
    tmp1 = tmp1 - dcy2[dy];
    if (tmp2 > 0)
        for (dz=0; (dz<=dz1); dz++) {
            for (daz=0; daz2[dz1] {
                /* find cell of in which possible neighbors
                are located. Note that this is not the cell
                of the grid point (dx,dz)
                */
                int i,j,k;
                int ix=dx,dx2,dx3;
                int iz=dz,dz2,dz3;
                int i1,i2,i3;
                int j1,j2,j3;
                int k1,k2,k3;
                int n1,n2,n3;
                int n11,n12,n13;
                int n21,n22,n23;
                int n31,n32,n33;
                int n111,n112,n113;
                int n121,n122,n123;
                int n131,n132,n133;
                int n211,n212,n213;
                int n221,n222,n223;
                int n231,n232,n233;
                int n311,n312,n313;
                int n321,n322,n323;
                int n331,n332,n333;
                int n1111,n1112,n1113;
                int n1121,n1122,n1123;
                int n1131,n1132,n1133;
                int n1211,n1212,n1213;
                int n1221,n1222,n1223;
                int n1231,n1232,n1233;
                int n1311,n1312,n1313;
                int n1321,n1322,n1323;
                int n1331,n1332,n1333;
                int n2111,n2112,n2113;
                int n2121,n2122,n2123;
                int n2131,n2132,n2133;
                int n2211,n2212,n2213;
                int n2221,n2222,n2223;
                int n2231,n2232,n2233;
                int n2311,n2312,n2313;
                int n2321,n2322,n2323;
                int n2331,n2332,n2333;
                int n3111,n3112,n3113;
                int n3121,n3122,n3123;
                int n3131,n3132,n3133;
                int n3211,n3212,n3213;
                int n3221,n3222,n3223;
                int n3231,n3232,n3233;
                int n3311,n3312,n3313;
                int n3321,n3322,n3323;
                int n3331,n3332,n3333;
                int n11111,n11112,n11113;
                int n11121,n11122,n11123;
                int n11131,n11132,n11133;
                int n11211,n11212,n11213;
                int n11221,n11222,n11223;
                int n11231,n11232,n11233;
                int n11311,n11312,n11313;
                int n11321,n11322,n11323;
                int n11331,n11332,n11333;
                int n12111,n12112,n12113;
                int n12121,n12122,n12123;
                int n12131,n12132,n12133;
                int n12211,n12212,n12213;
                int n12221,n12222,n12223;
                int n12231,n12232,n12233;
                int n12311,n12312,n12313;
                int n12321,n12322,n12323;
                int n12331,n12332,n12333;
                int n13111,n13112,n13113;
                int n13121,n13122,n13123;
                int n13131,n13132,n13133;
                int n13211,n13212,n13213;
                int n13221,n13222,n13223;
                int n13231,n13232,n13233;
                int n13311,n13312,n13313;
                int n13321,n13322,n13323;
                int n13331,n13332,n13333;
                int n21111,n21112,n21113;
                int n21121,n21122,n21123;
                int n21131,n21132,n21133;
                int n21211,n21212,n21213;
                int n21221,n21222,n21223;
                int n21231,n21232,n21233;
                int n21311,n21312,n21313;
                int n21321,n21322,n21323;
                int n21331,n21332,n21333;
                int n22111,n22112,n22113;
                int n22121,n22122,n22123;
                int n22131,n22132,n22133;
                int n22211,n22212,n22213;
                int n22221,n22222,n22223;
                int n22231,n22232,n22233;
                int n22311,n22312,n22313;
                int n22321,n22322,n22323;
                int n22331,n22332,n22333;
                int n23111,n23112,n23113;
                int n23121,n23122,n23123;
                int n23131,n23132,n23133;
                int n23211,n23212,n23213;
                int n23221,n23222,n23223;
                int n23231,n23232,n23233;
                int n23311,n23312,n23313;
                int n23321,n23322,n23323;
                int n23331,n23332,n23333;
                int n31111,n31112,n31113;
                int n31121,n31122,n31123;
                int n31131,n31132,n31133;
                int n31211,n31212,n31213;
                int n31221,n31222,n31223;
                int n31231,n31232,n31233;
                int n31311,n31312,n31313;
                int n31321,n31322,n31323;
                int n31331,n31332,n31333;
                int n32111,n32112,n32113;
                int n32121,n32122,n32123;
                int n32131,n32132,n32133;
                int n32211,n32212,n32213;
                int n32221,n32222,n32223;
                int n32231,n32232,n32233;
                int n32311,n32312,n32313;
                int n32321,n32322,n32323;
                int n32331,n32332,n32333;
                int n33111,n33112,n33113;
                int n33121,n33122,n33123;
                int n33131,n33132,n33133;
                int n33211,n33212,n33213;
                int n33221,n33222,n33223;
                int n33231,n33232,n33233;
                int n33311,n33312,n33313;
                int n33321,n33322,n33323;
                int n33331,n33332,n33333;
                int n111111,n111112,n111113;
                int n111121,n111122,n111123;
                int n111131,n111132,n111133;
                int n111211,n111212,n111213;
                int n111221,n111222,n111223;
                int n111231,n111232,n111233;
                int n111311,n111312,n111313;
                int n111321,n111322,n111323;
                int n111331,n111332,n111333;
                int n112111,n112112,n112113;
                int n112121,n112122,n112123;
                int n112131,n112132,n112133;
                int n112211,n112212,n112213;
                int n112221,n112222,n112223;
                int n112231,n112232,n112233;
                int n112311,n112312,n112313;
                int n112321,n112322,n112323;
                int n112331,n112332,n112333;
                int n113111,n113112,n113113;
                int n113121,n113122,n113123;
                int n113131,n113132,n113133;
                int n113211,n113212,n113213;
                int n113221,n113222,n113223;
                int n113231,n113232,n113233;
                int n113311,n113312,n113313;
                int n113321,n113322,n113323;
                int n113331,n113332,n113333;
                int n121111,n121112,n121113;
                int n121121,n121122,n121123;
                int n121131,n121132,n121133;
                int n121211,n121212,n121213;
                int n121221,n121222,n121223;
                int n121231,n121232,n121233;
                int n121311,n121312,n121313;
                int n121321,n121322,n121323;
                int n121331,n121332,n121333;
                int n122111,n122112,n122113;
                int n122121,n122122,n122123;
                int n122131,n122132,n122133;
                int n122211,n122212,n122213;
                int n122221,n122222,n122223;
                int n122231,n122232,n122233;
                int n122311,n122312,n122313;
                int n122321,n122322,n122323;
                int n122331,n122332,n122333;
                int n123111,n123112,n123113;
                int n123121,n123122,n123123;
                int n123131,n123132,n123133;
                int n123211,n123212,n123213;
                int n123221,n123222,n123223;
                int n123231,n123232,n123233;
                int n123311,n123312,n123313;
                int n123321,n123322,n123323;
                int n123331,n123332,n123333;
                int n13
```

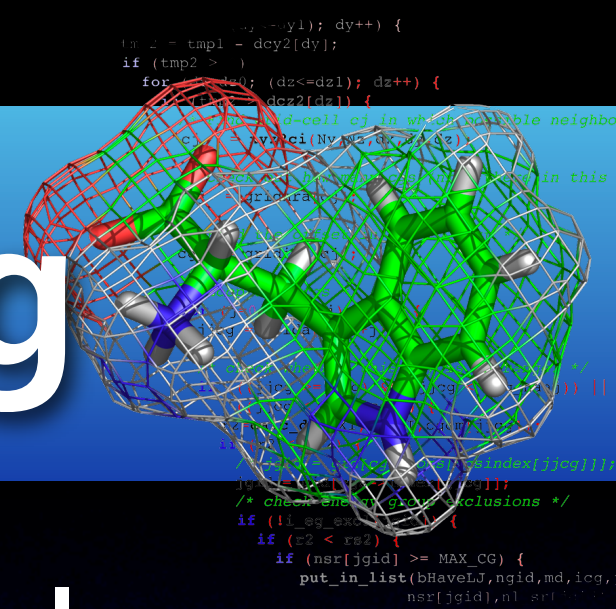
- # Allen & Tildesley



MODEL2
1E212 OF

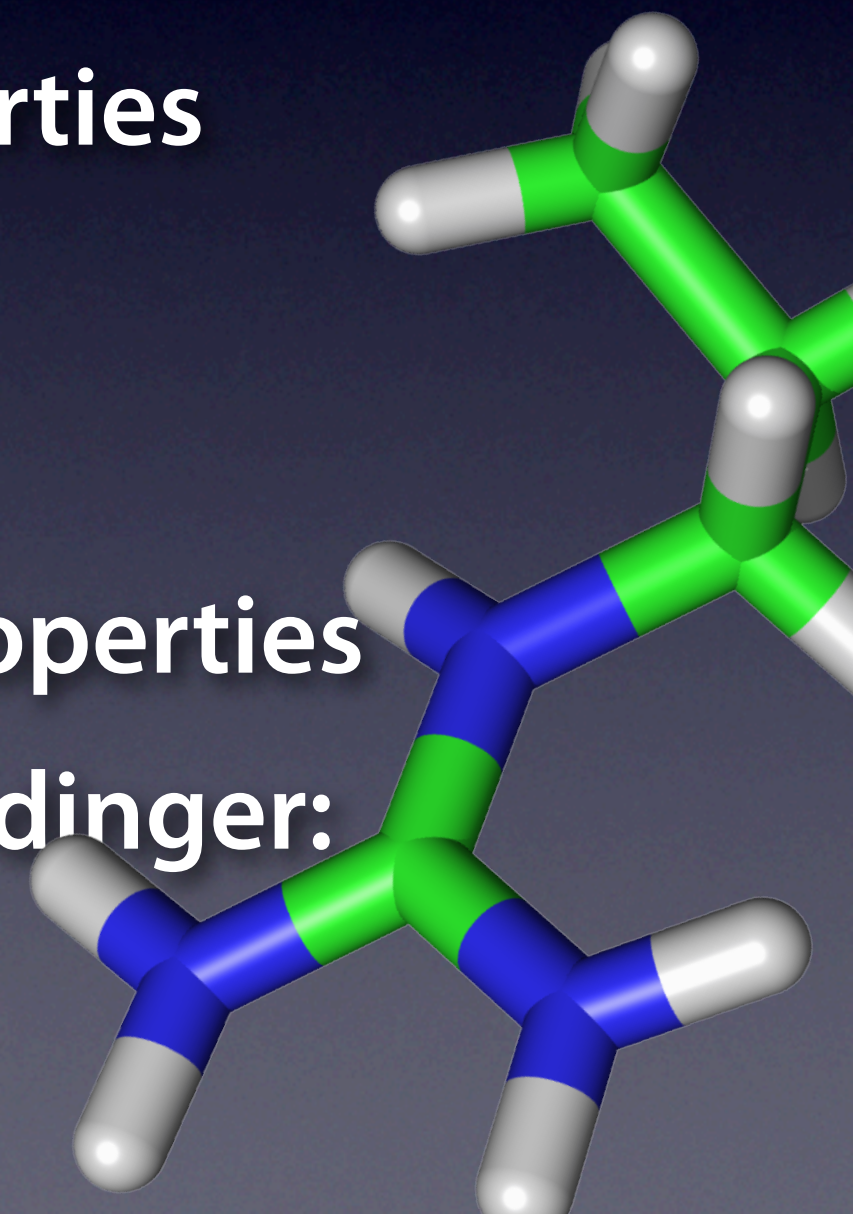
THEORIES
OF

Molecular Modeling



- The art of describing complex chemical systems from atomic/quantum models
- Prediction of macroscopic properties
 - Ensemble averages
 - Static / Equilibrium properties
 - Dynamic / Non-equilibrium properties
 - Everything described by Schrödinger:

$$H\Psi(x,t) = E\Psi(x,t)$$



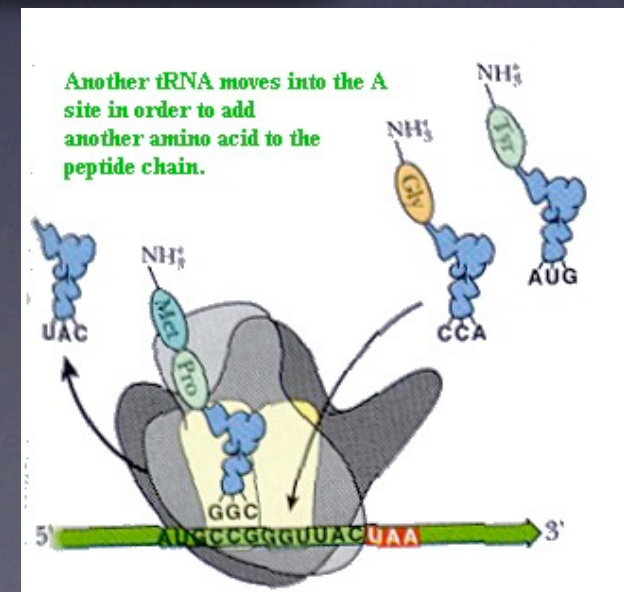
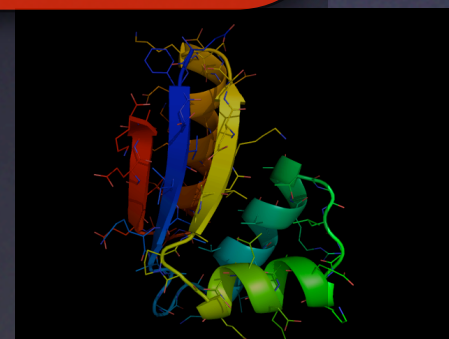
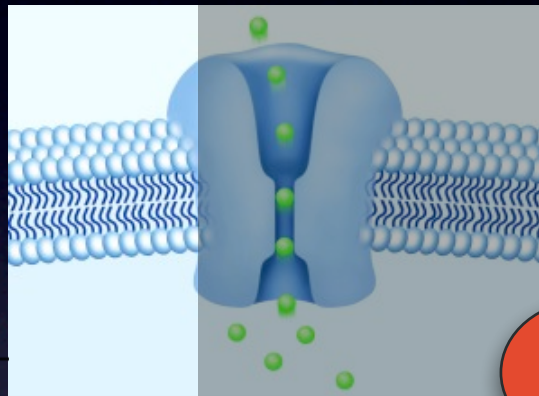
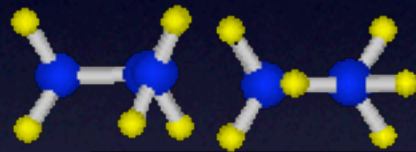
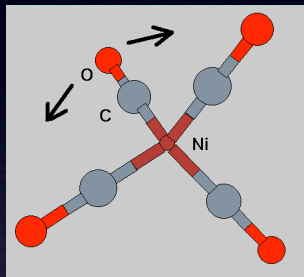
[illegible]

Less detail

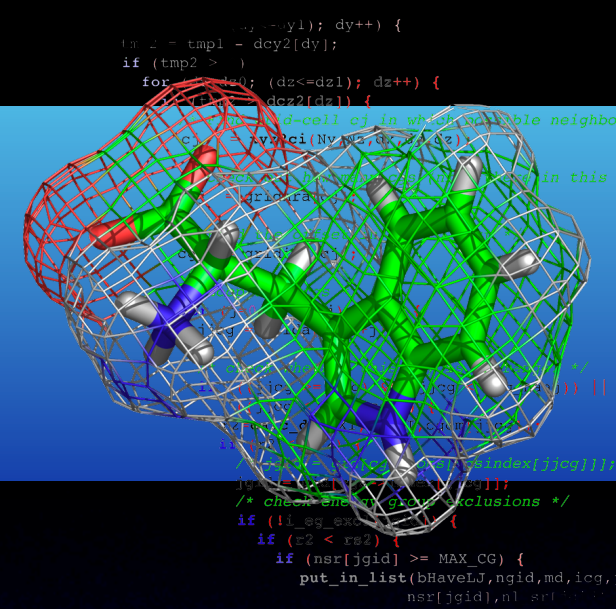
Where we are

Where we want to be

Parameter quality?

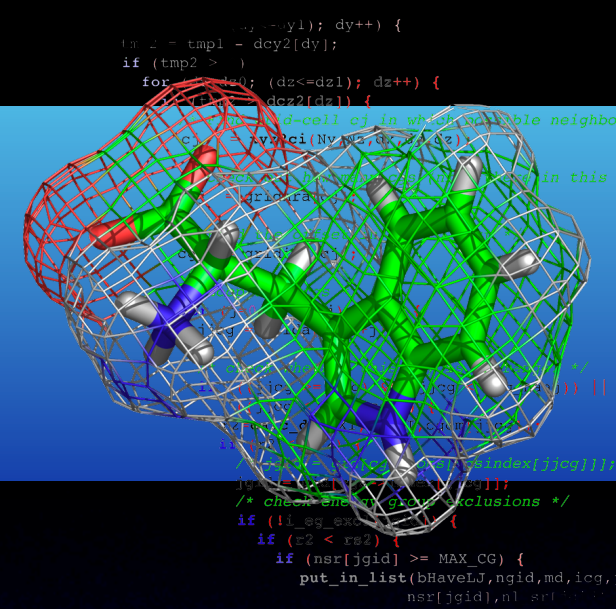


Limitations of QM



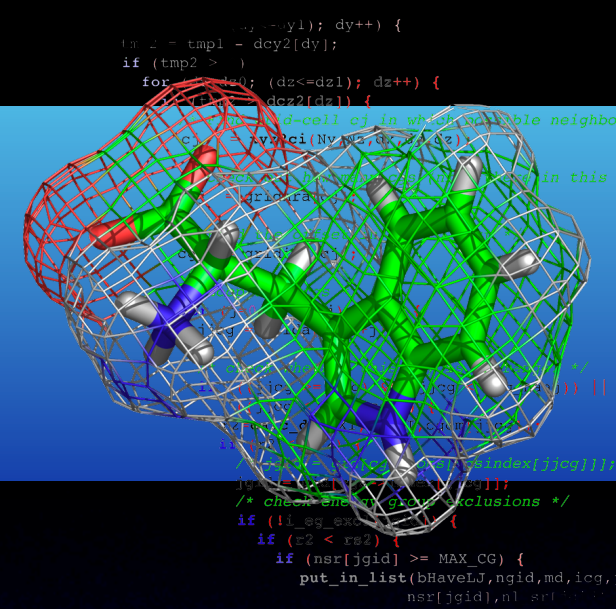
- QM is necessary to describe electronic, and sometimes hydrogen, motion
- Heavy atoms can be treated classically
- Can only solve up to ~100 atoms accurately
- No time dependence
- Several approximations exist, but they are only accurate on QM scale - you cannot extrapolate 10 orders of magnitude!

Empirical models



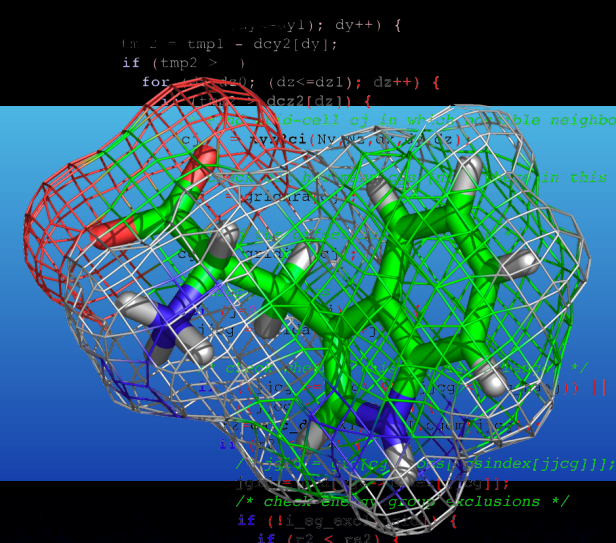
- Classical particles
- Parameterize to *reproduce experiments*
- Some properties like charge, bond parameters can be obtained from QM (Semi-empirical models)
- Atomic simulation is *not* always the best choice
 - coarse-grained methods (Marrink)
 - QSAR, chemoinformatics

Example properties



- Free energy (of binding, solvation, interaction)
- Diffusion coefficients, viscosity
- Reaction rates, phase transition properties
- Protein folding times
- Structure refinement
- All of these are *ensemble averages* over huge numbers of molecules ($N_{\text{avogadro}} \sim 6E23$)

Example properties

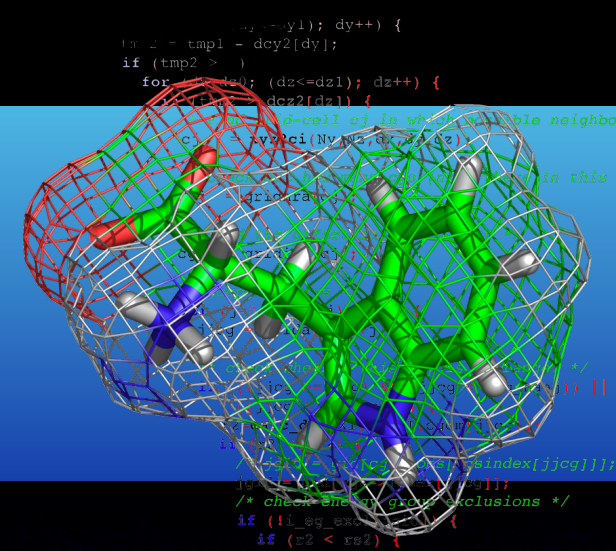


- Free energy (of binding, solvation, interaction)
- Diffusion coefficients
- Reaction rates
- Protein folding times
- Structure refinement
- All of these are *ensemble averages* of many molecules ($N_{\text{avogadro}} \sim 6\text{E}23$)
- Thus, our goal is to sample equilibrium ensembles, not simulate individual particles!

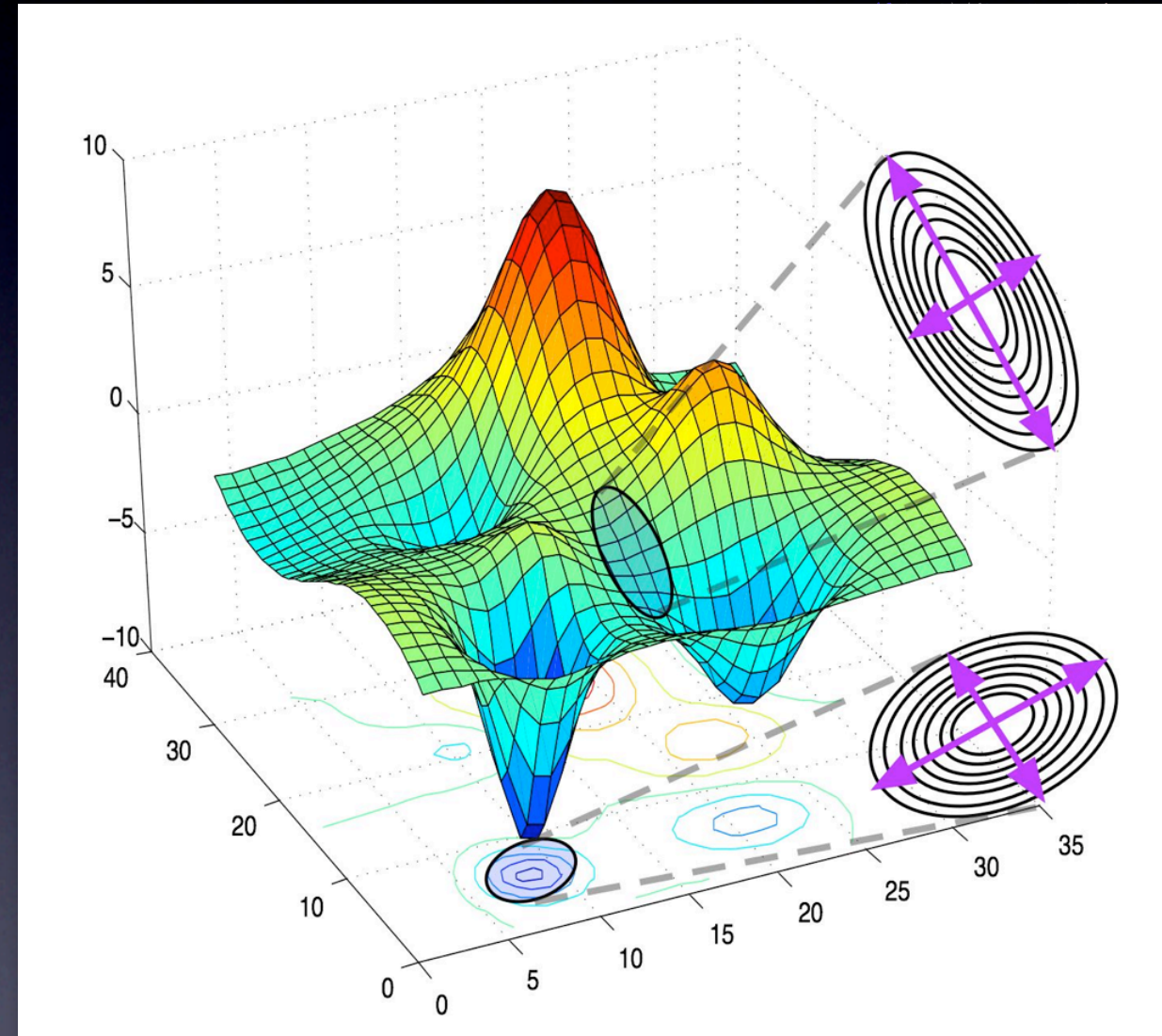
[illegible]

-

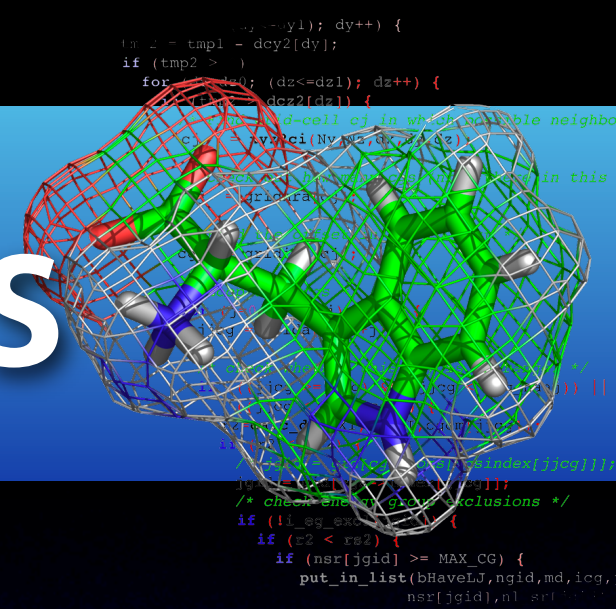
Energy landscapes



- $3N$ -dimensional space
- Native structure is the free energy minimum
- Ideally, we would sample all of phase space exhaustively
- In practice we have to make do with the most populated parts



Molecular Dynamics



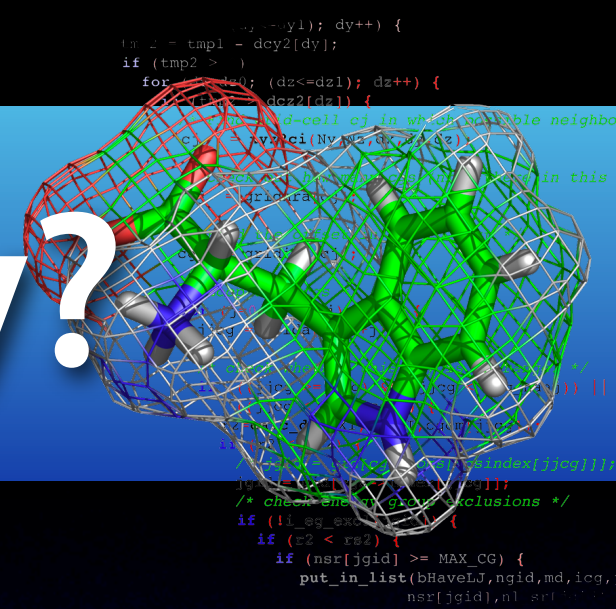
- Solve Newton's equations of motion:

$$m_i \frac{\partial^2 r_i}{\partial t^2} = F_i \quad i = 1..N$$

$$F_i = -\frac{\partial V(r)}{\partial r_i}$$

- Timestep has to be small (femtoseconds)
- Forces depend on all particle coordinates in the system (expensive to calculate)
- Generates a system *trajectory* over time


What is a trajectory?



- *NOT* an accurate prediction of the motion of an individual particle
- Molecular dynamics is a chaotic process - differences grow exponentially
- But trajectories aren't random either?
- *Shadow trajectory*: Simulated path stays close to (some) real path for shorter times

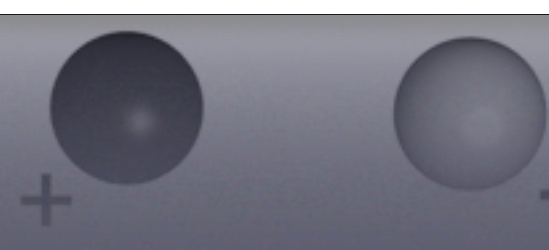
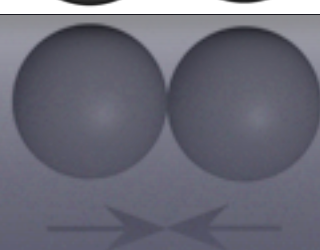
[illegible]

A diagram of a diatomic molecule consisting of two grey spheres connected by a grey rod. A double-headed black arrow is drawn above the rod, pointing in opposite directions from the center of the bond, representing the vibrational motion of the molecule.

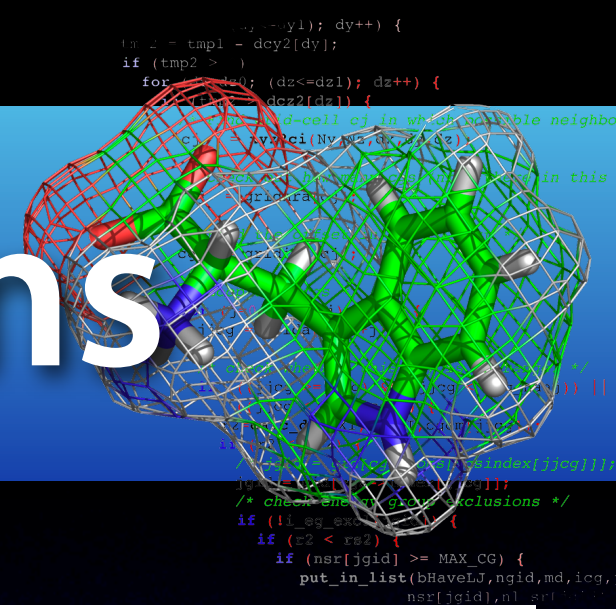


A diagram of a bent triatomic molecule, represented by three gray spheres (atoms) connected by gray rods (bonds). The central atom is at the bottom, and the two other atoms are positioned to its left and right, forming a V-shape. A curved double-headed arrow is drawn between the two outer atoms, indicating the bond angle. Above the diagram, the Greek letter θ is written, representing the bond angle.

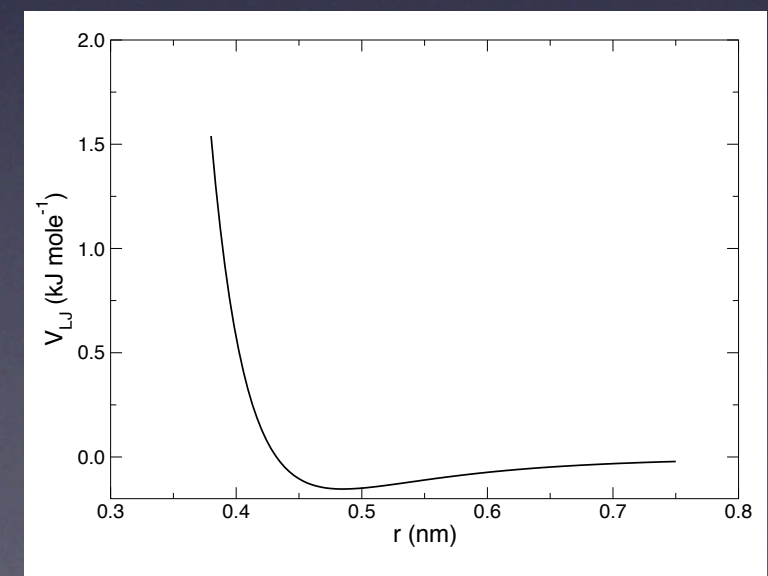
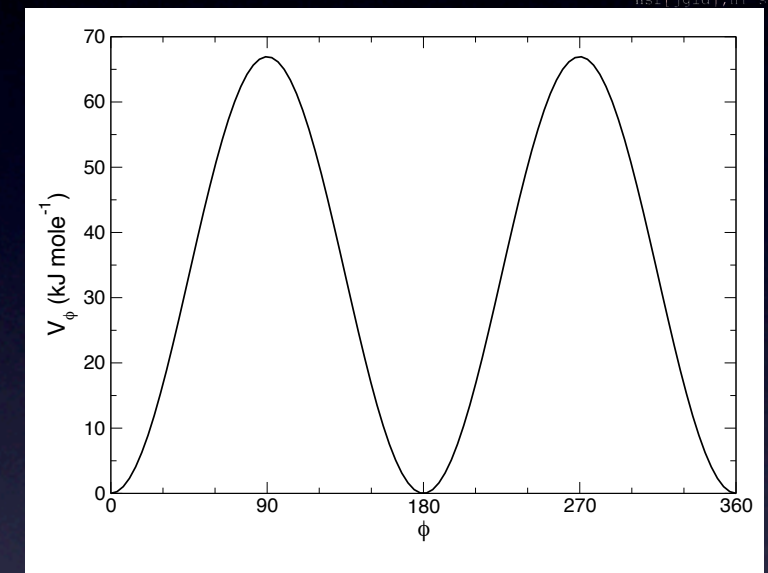
A diagram showing two identical grey spheres in contact. Above them, two black arrows point directly toward each other, representing the initial velocities of the two objects before the collision.



Empirical Interactions

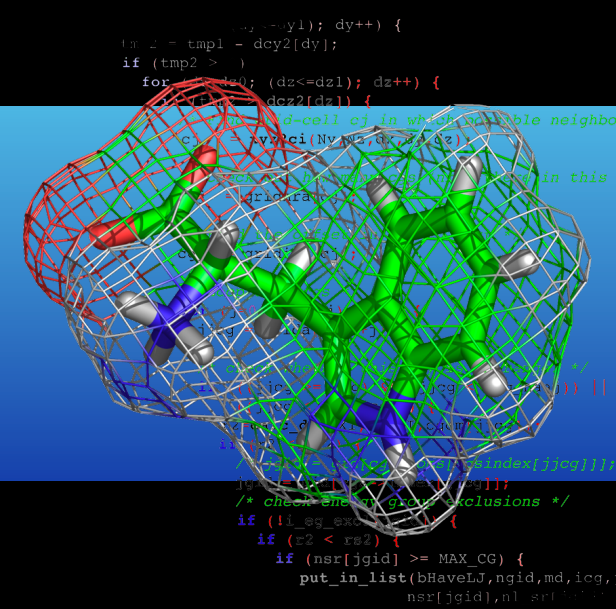
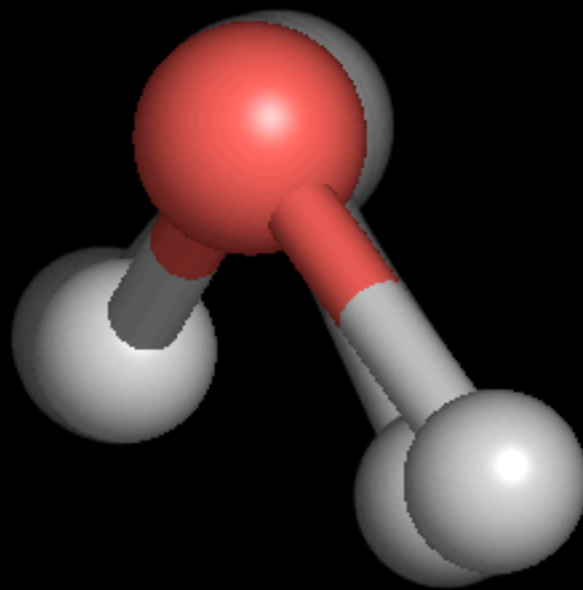


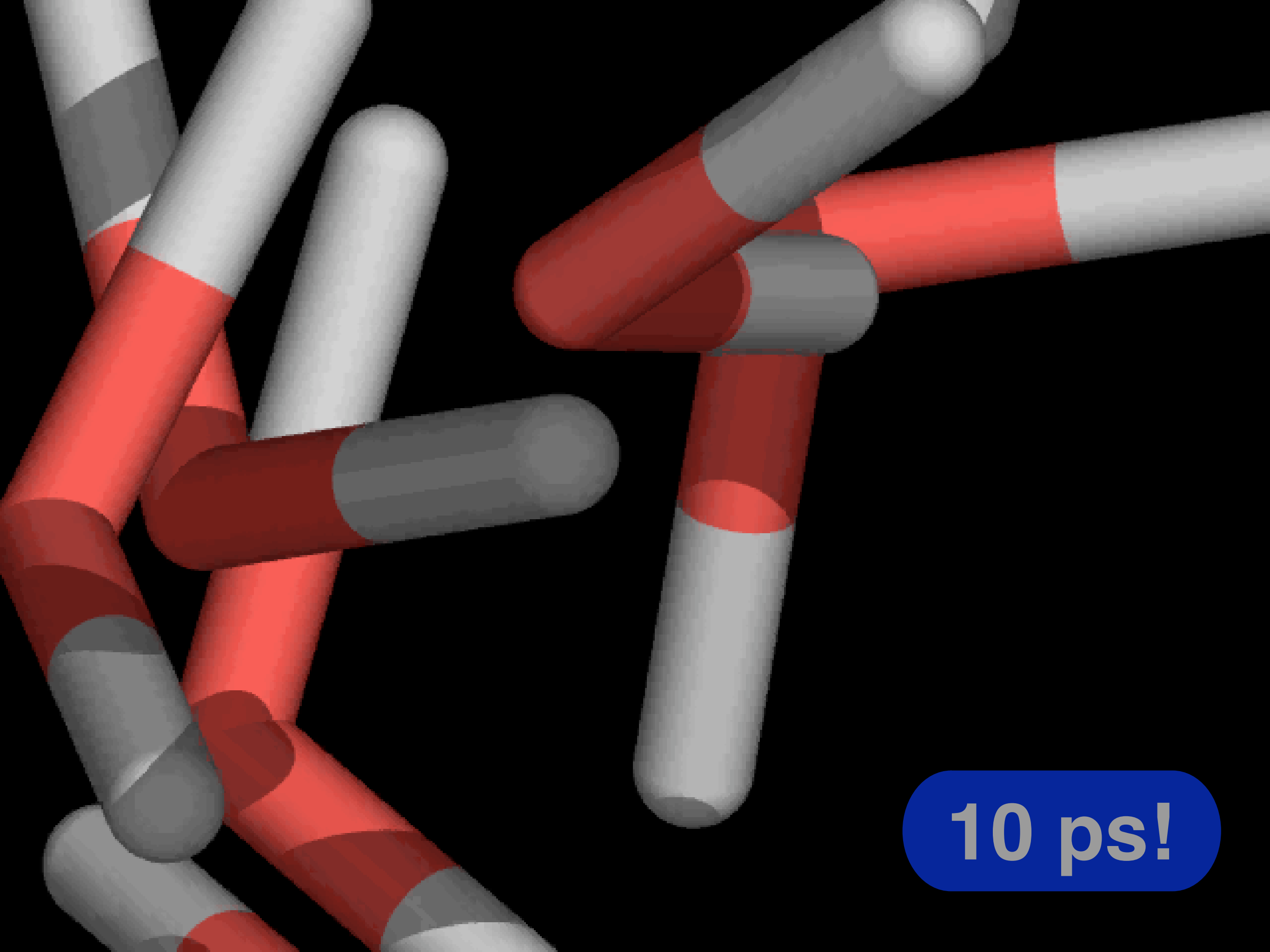
$$\begin{aligned}
 V(r) = & \sum_{bonds} \frac{1}{2} k_{ij}^b (r_{ij} - r_{ij}^0)^2 \\
 & + \sum_{angles} \frac{1}{2} k_{ijk}^\theta (\theta_{ijk} - \theta_{ijk}^0)^2 \\
 & + \sum_{torsions} \left\{ \sum_n k_\theta [1 + \cos(n\phi - \phi_0)] \right\} \\
 & + \sum_{impropers} k_\xi (\xi_{ijkl} - \xi_{ijkl}^0) \\
 & + \sum_{i,j} \frac{q_i q_j}{4\pi\epsilon_0 r_{ij}} \\
 & + \sum_{i,j} \left[\frac{C_{12}}{r_{ij}^{12}} - \frac{C_6}{r_{ij}^6} \right]
 \end{aligned}$$



Taking a timestep

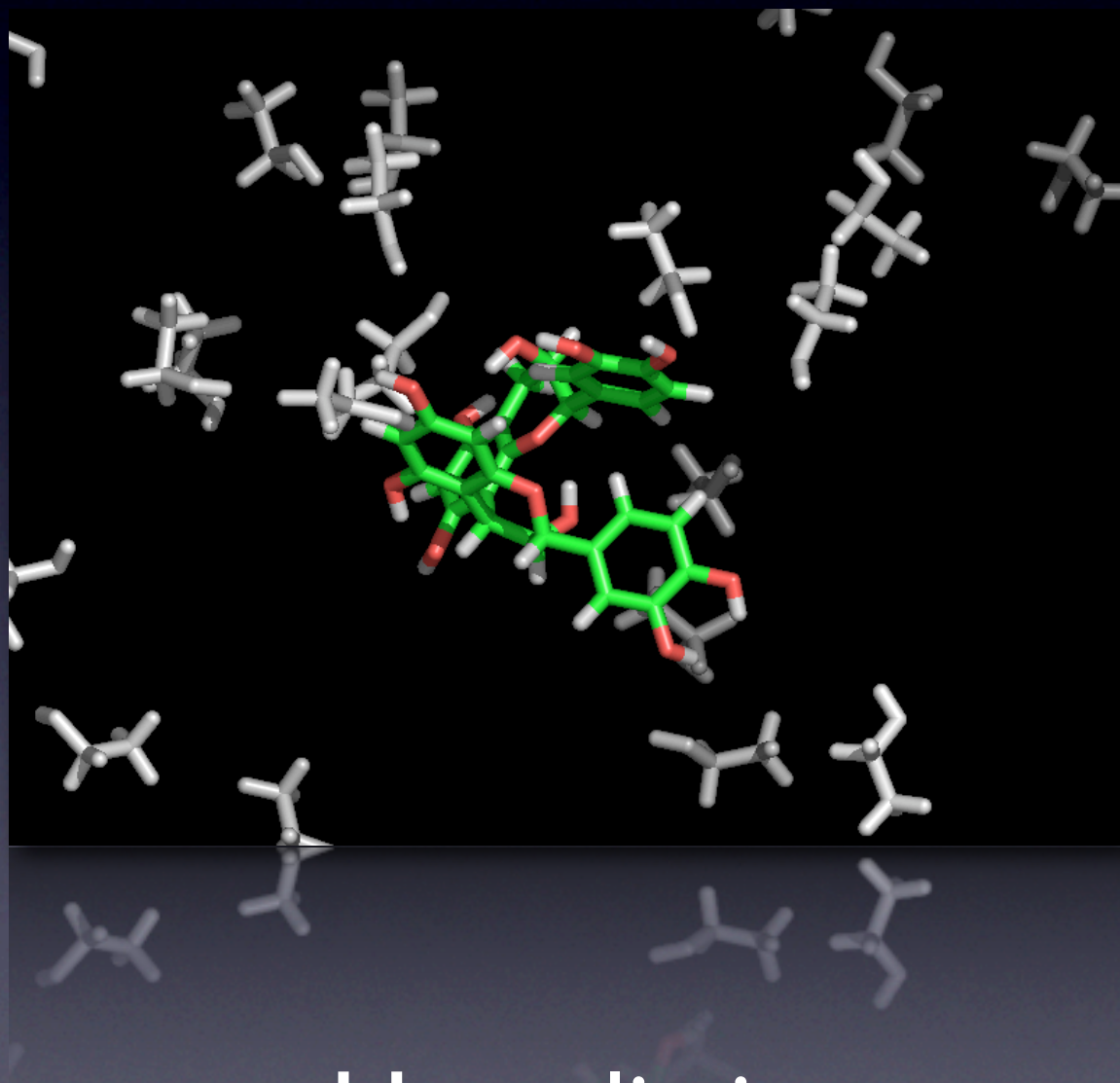
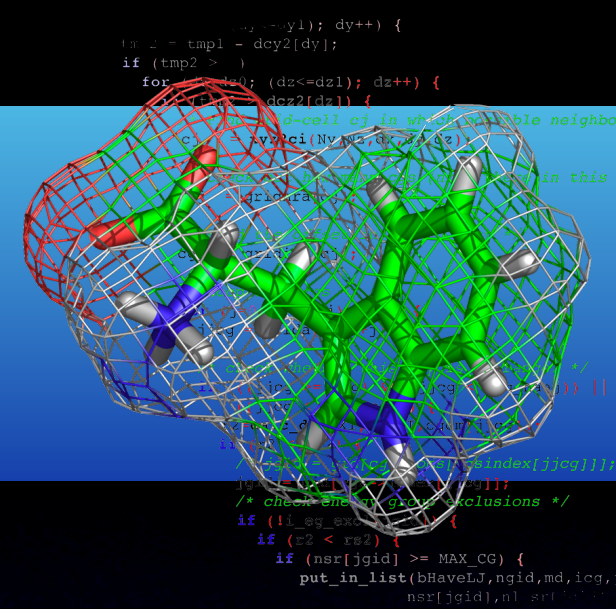
8 fs



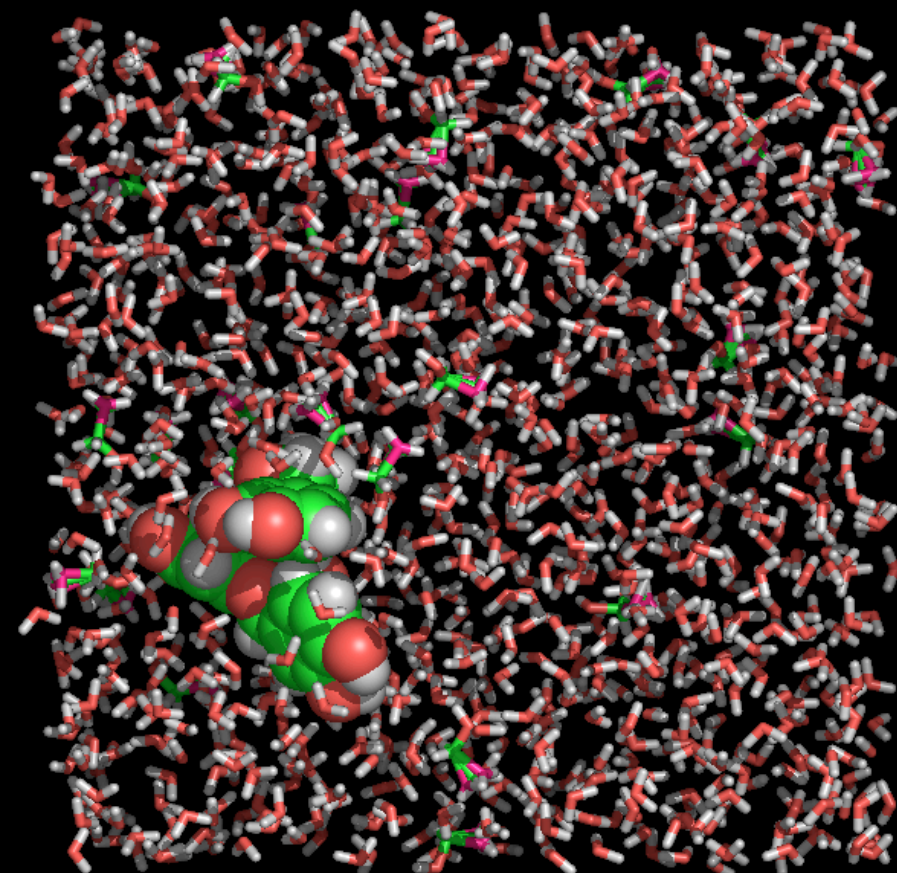


10 ps!

Environments

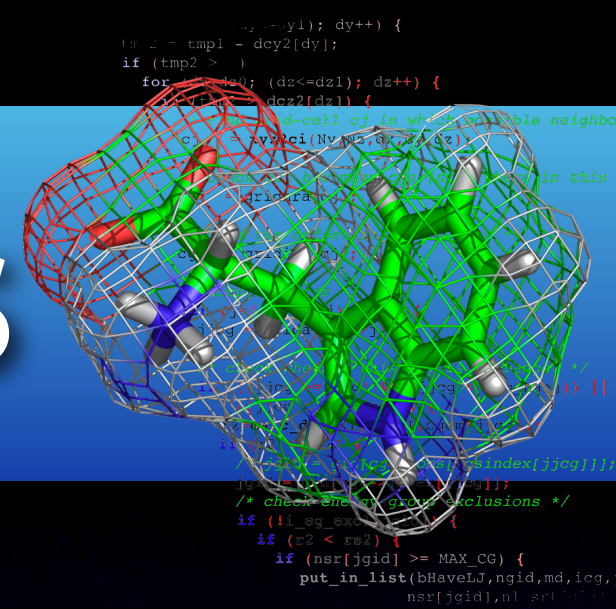


Unrealistic

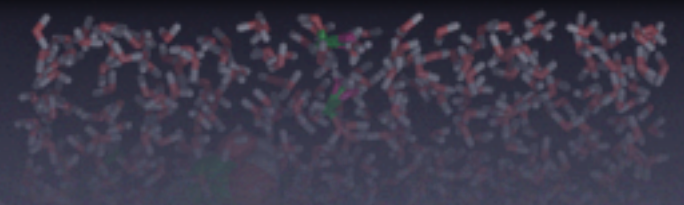
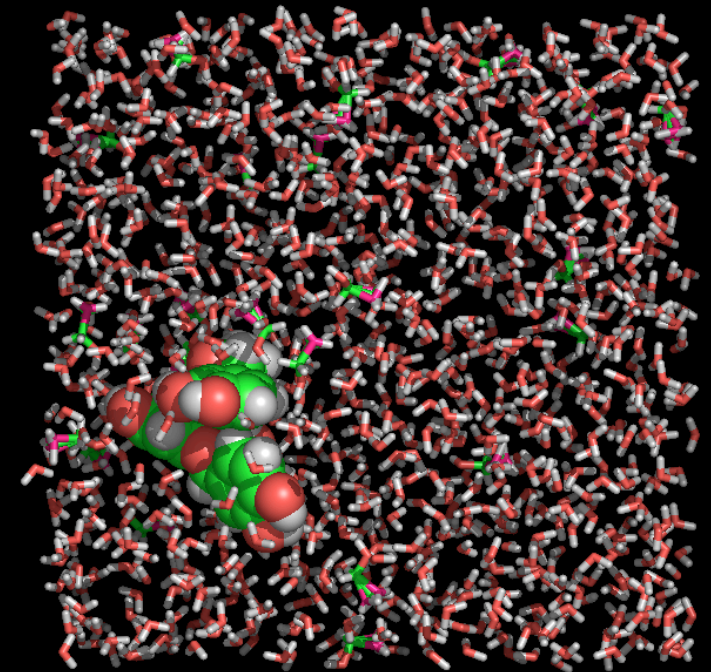


Realistic

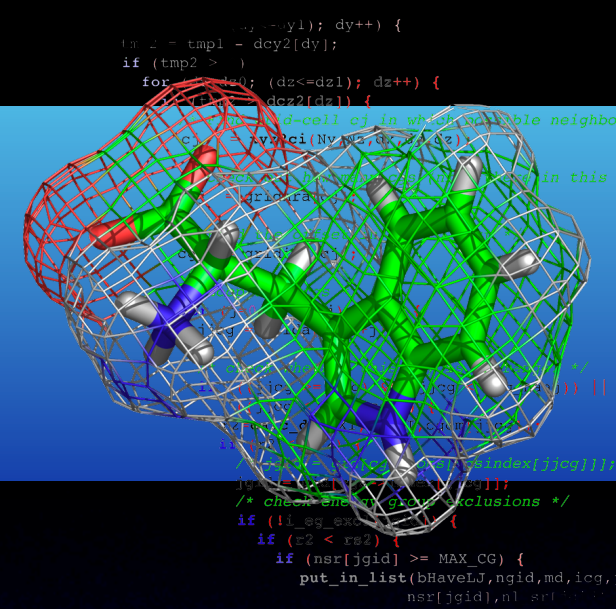
Boundary conditions



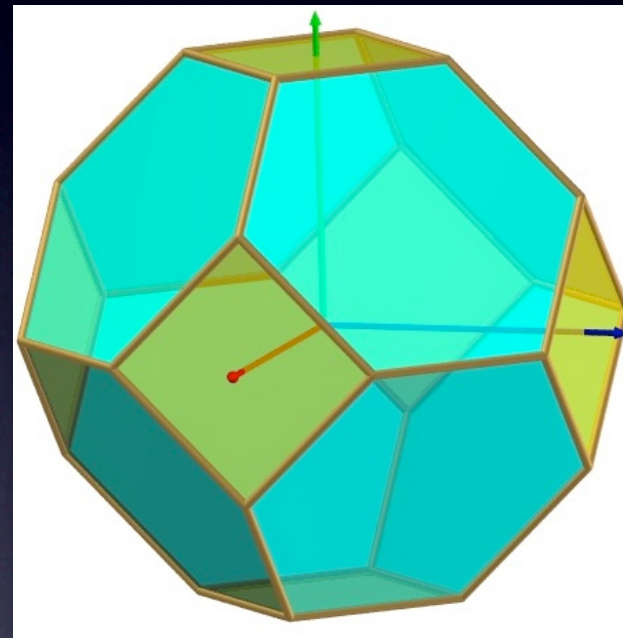
- Vacuum: No solvent
- Droplet: Spherical layer of water around protein, solvent molecules constrained with random forces (not in Gromacs)
- Periodic Boundary Conditions (PBC) - a water that exits on the right reappears on the left



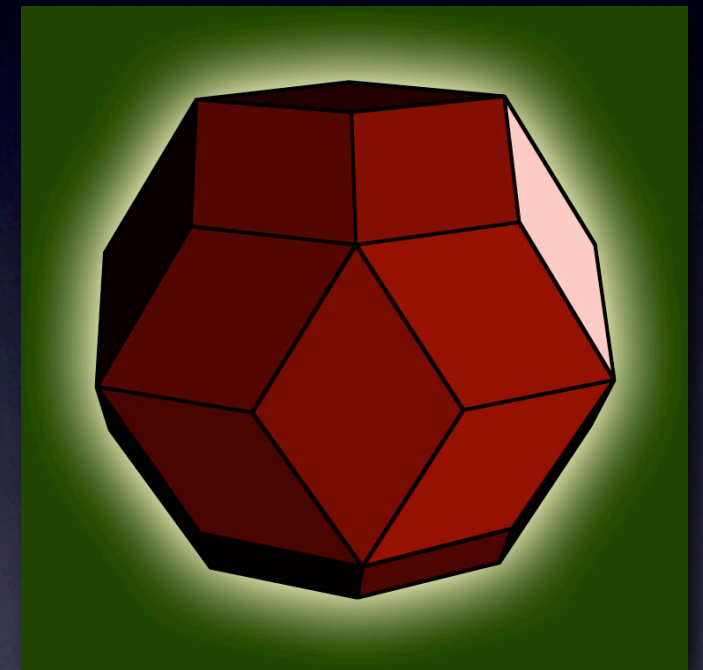
Periodic cell shapes



- Cubic / rectangular
- Truncated octahedron (more spherical)
- Rhombic dodecahedron (most spherical cell)
- Octahedron volume is only 77% of cube, and dodecahedron 71% at same periodic distance!



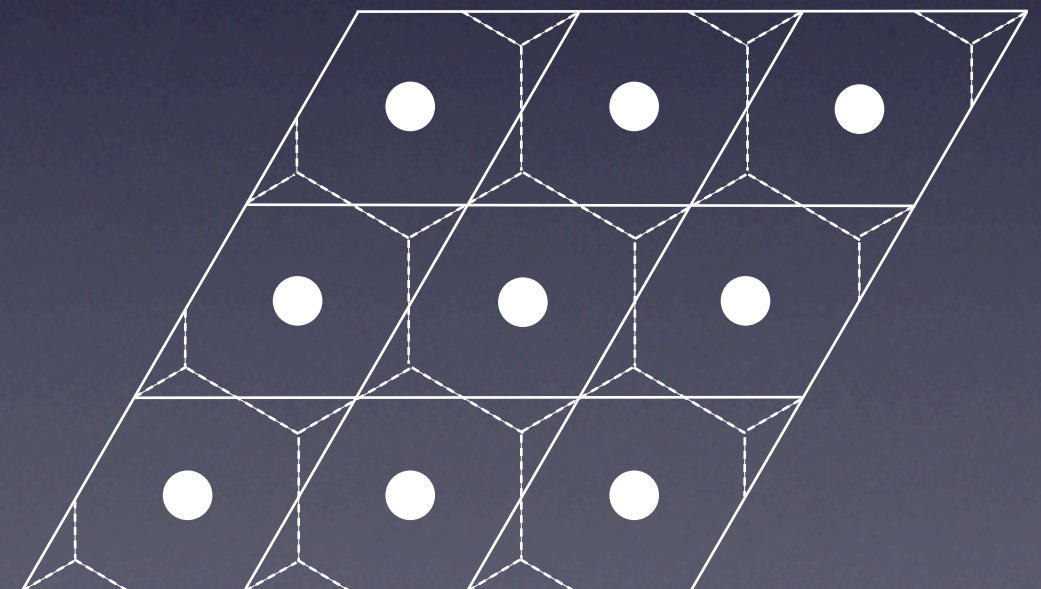
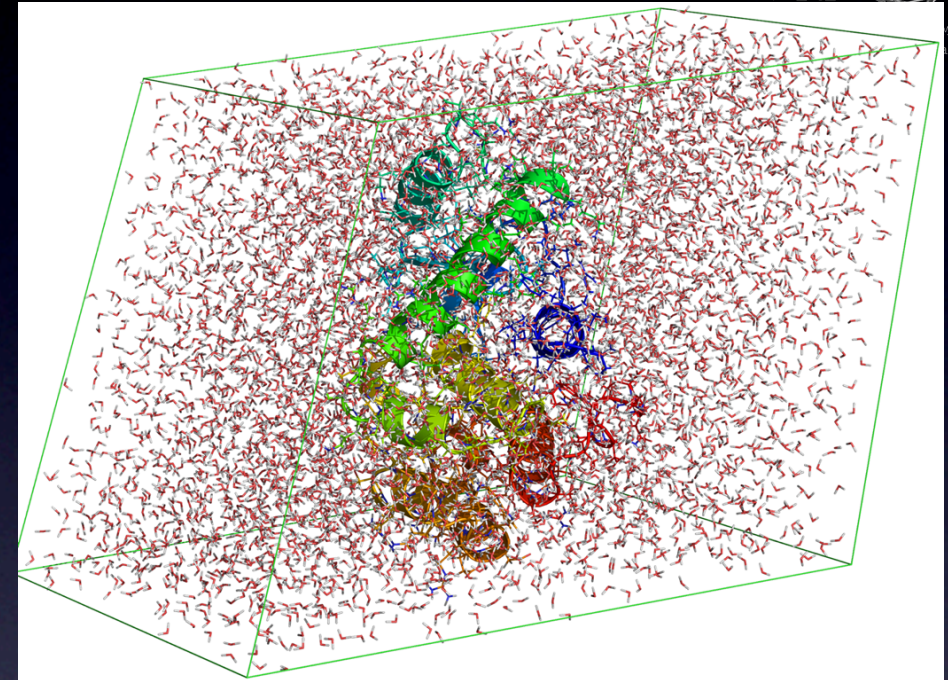
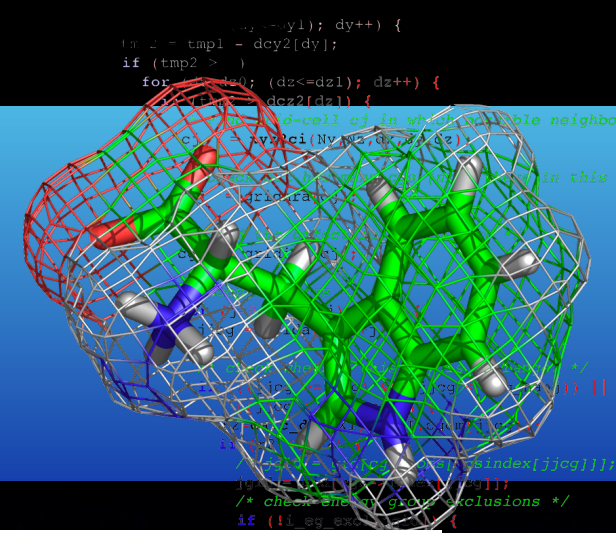
Truncated
octahedron



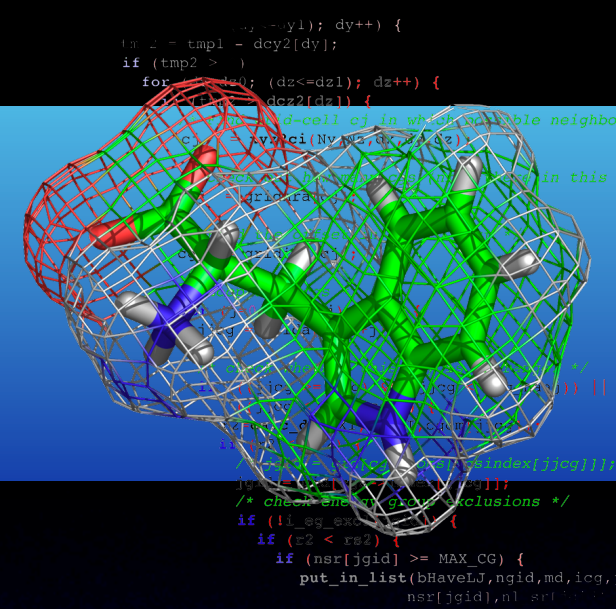
Rhombic
dodecahedron

Triclinic cells

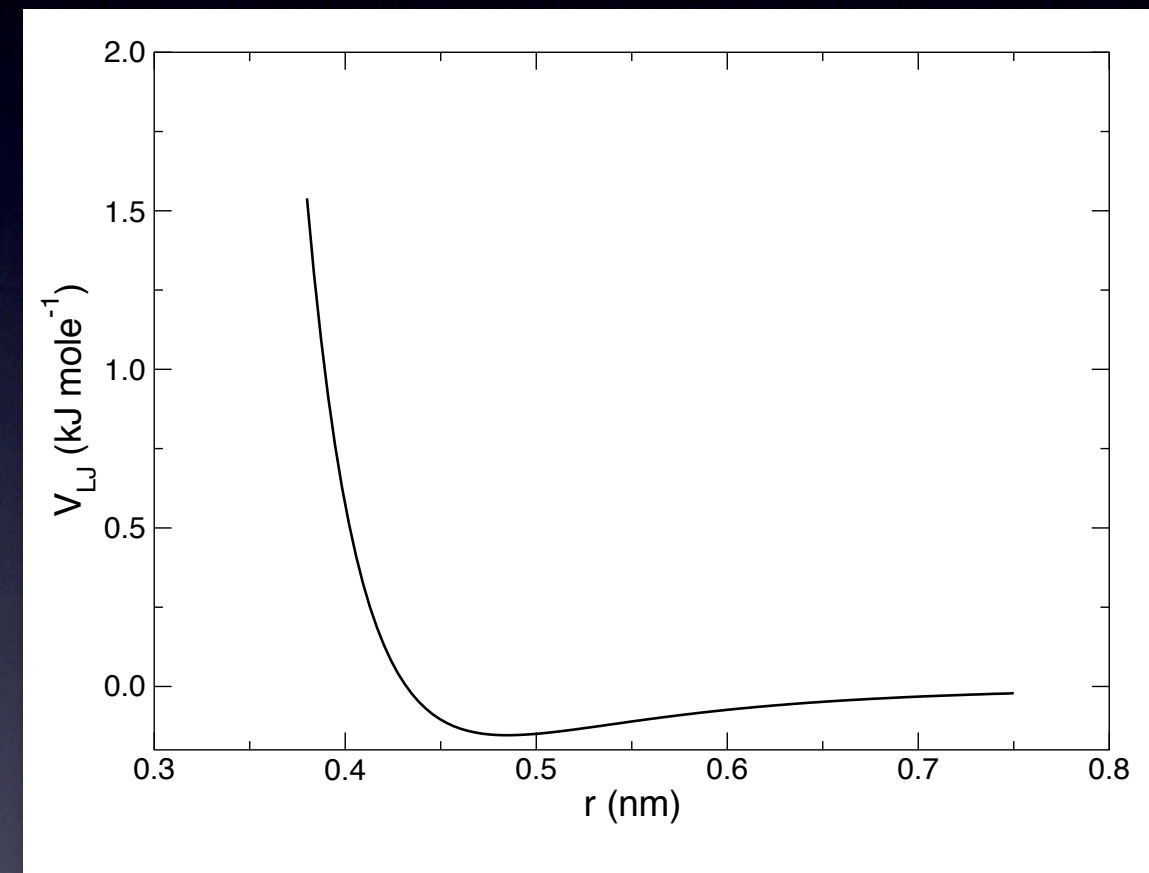
- Gromacs uses triclinic cells internally
- These can represent any of the cells mentioned, and also e.g. hexagonal boxes
- Not trivial to see in 3D, but see the hexagonal example in 2D to the right!



Lennard-Jones



- Interactions between all particles in a system
- Long range dipole dispersions: $1/r^6$
- Short range repulsion: Approx. $1/r^{12}$ (Really exp)
- Decays fast - cutoff fine

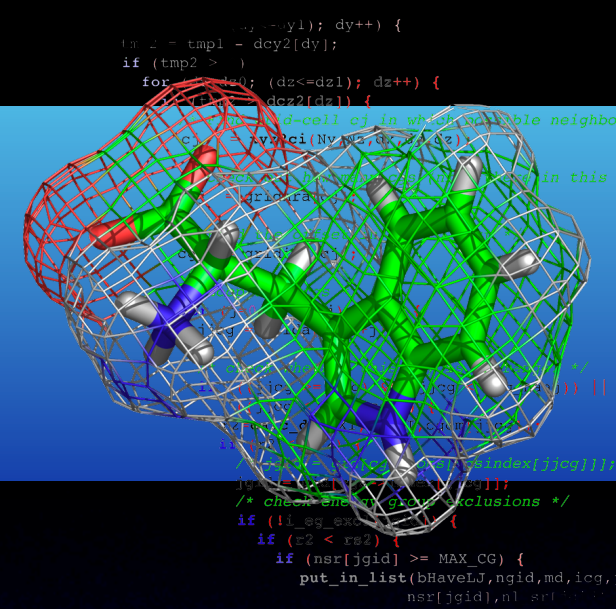


$$4\epsilon \left[\left(\frac{\sigma}{r} \right)^{12} - \left(\frac{\sigma}{r} \right)^6 \right]$$

$$\frac{C_{12}}{r_{ij}^{12}} - \frac{C_6}{r_{ij}^6}$$

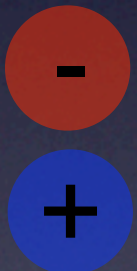
or

Electrostatics

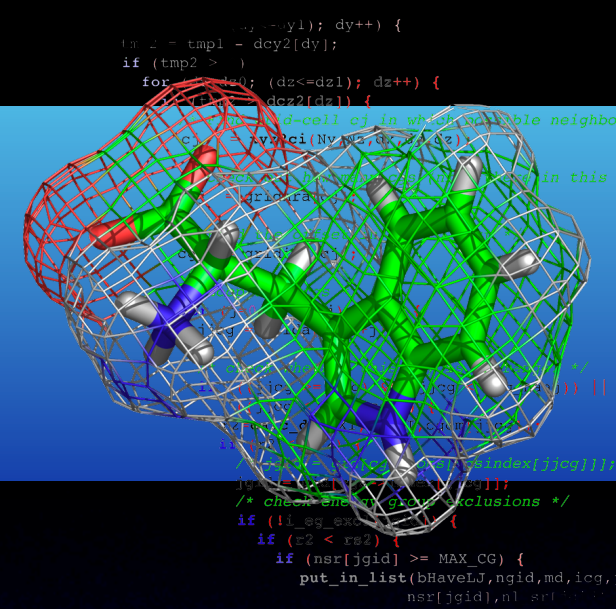


- Interactions between charged particles
- Strong interaction - sensitive to partial charges!
- Decays very slowly; cut-offs do not even always converge!
- Charge groups can be used to guarantee convergence

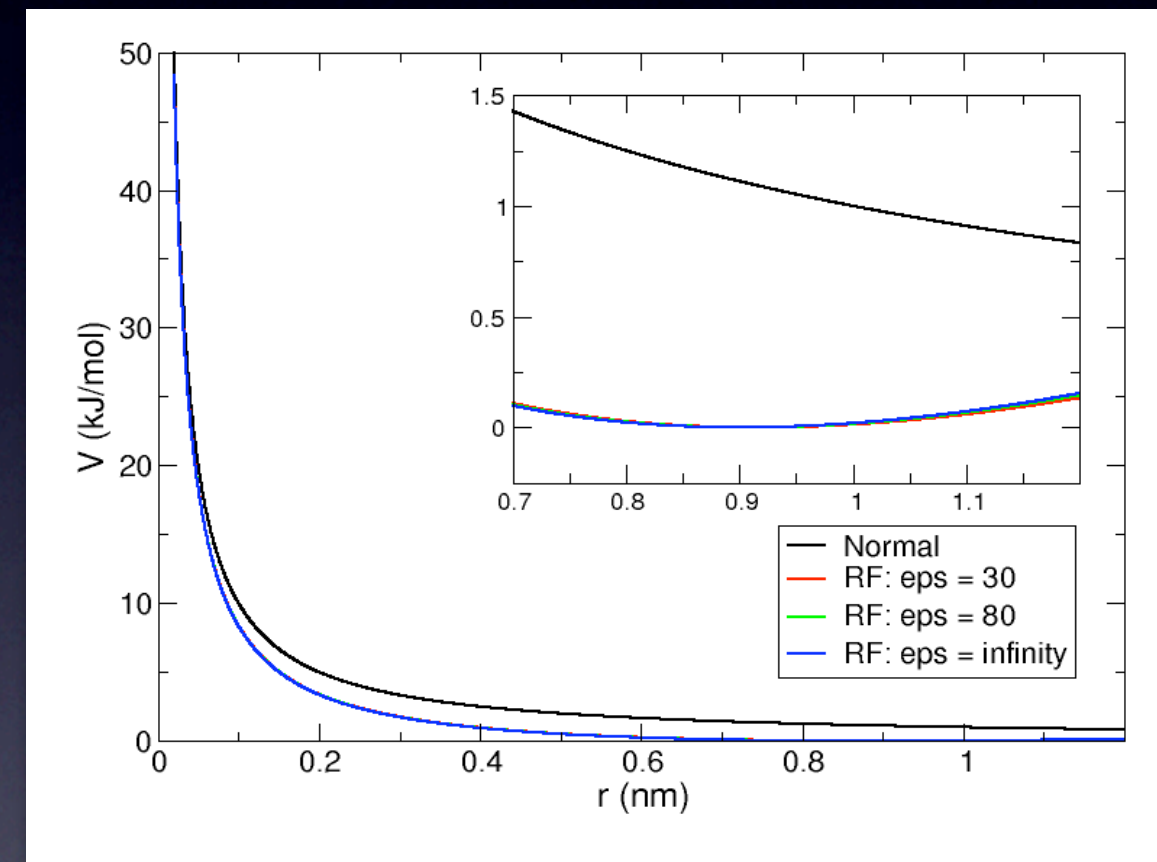
$$\frac{1}{4\pi\epsilon_0} \frac{q_i q_j}{r_{ij}}$$



Reaction-field

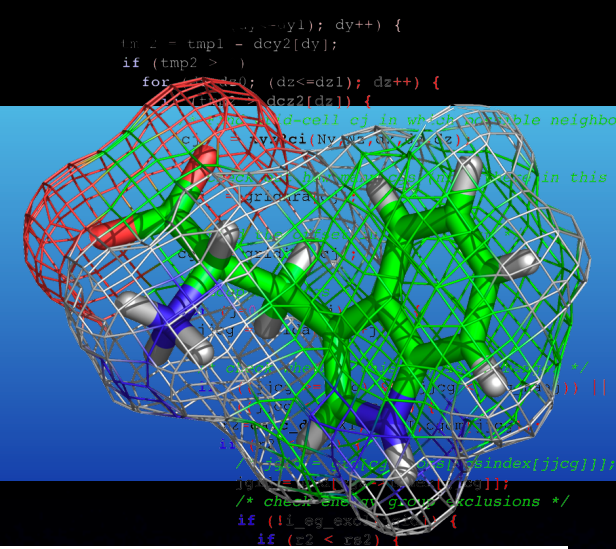


- Assume interactions are damped by ϵ_r beyond cut-off
- Homogeneous & isotropic
- Really cheap to calculate
- Good for dipoles, not ions

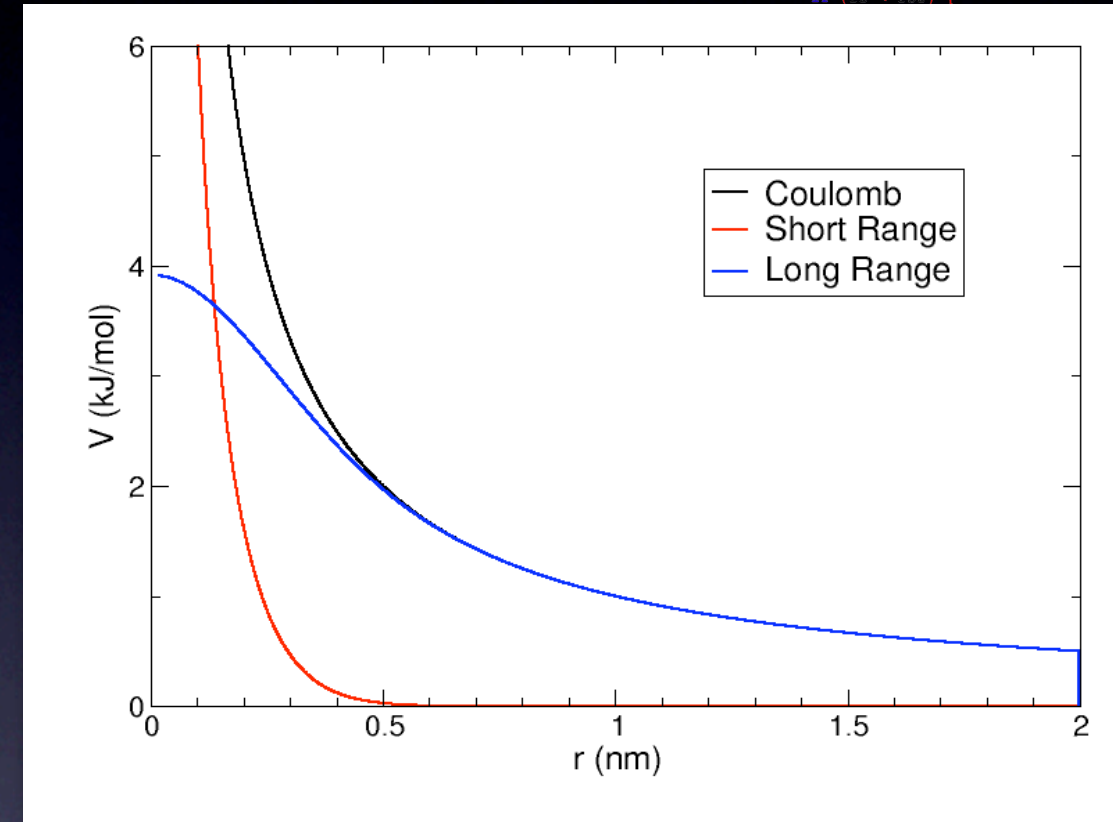


$$V = V_{coulomb} * \left(1 + \frac{\epsilon_{rf} - 1}{2\epsilon_{rf} + 1} \right) \frac{r^3}{r_{cut}^3}$$

Ewald summation



- Separate electrostatics into long & short range parts
- Short range decays fast (cut-off can be applied)
- Long range solved in reciprocal space, just as for crystals - corresponds to sum over the infinite periodic copies!

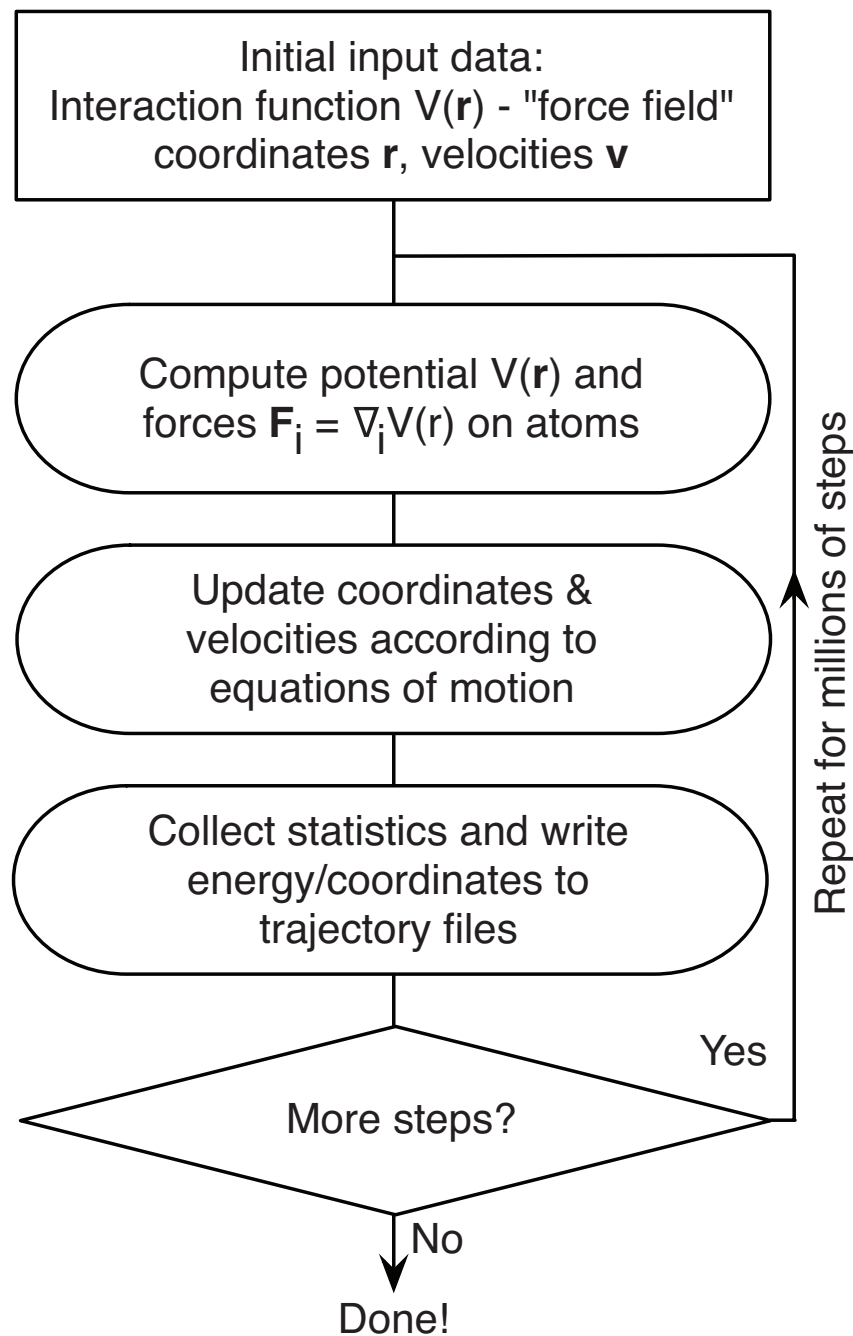
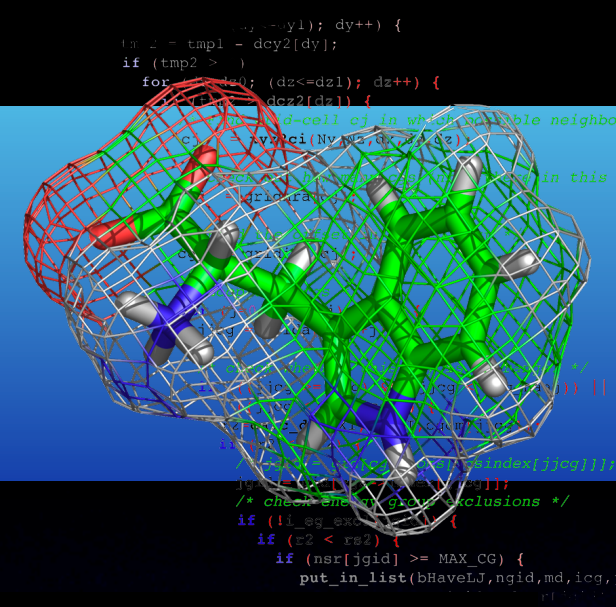


$$V_{sr} = V_{coulomb} * \operatorname{erfc}(r/\beta)$$
$$V_{lr} = V_{coulomb} * (1 - \operatorname{erfc}(r/\beta))$$

[illegible]

-

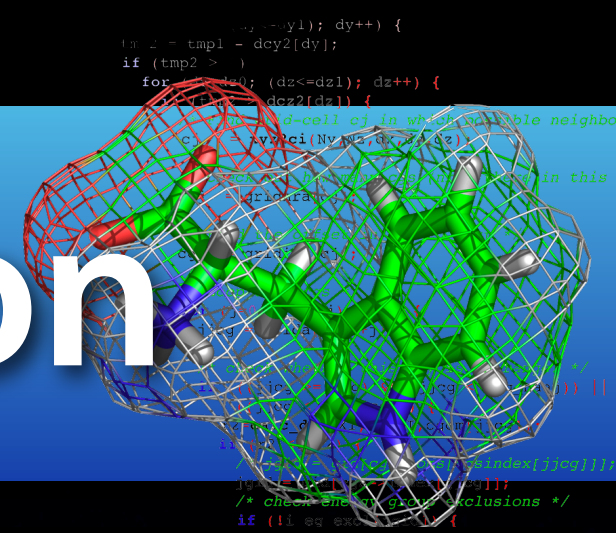
MD algorithm



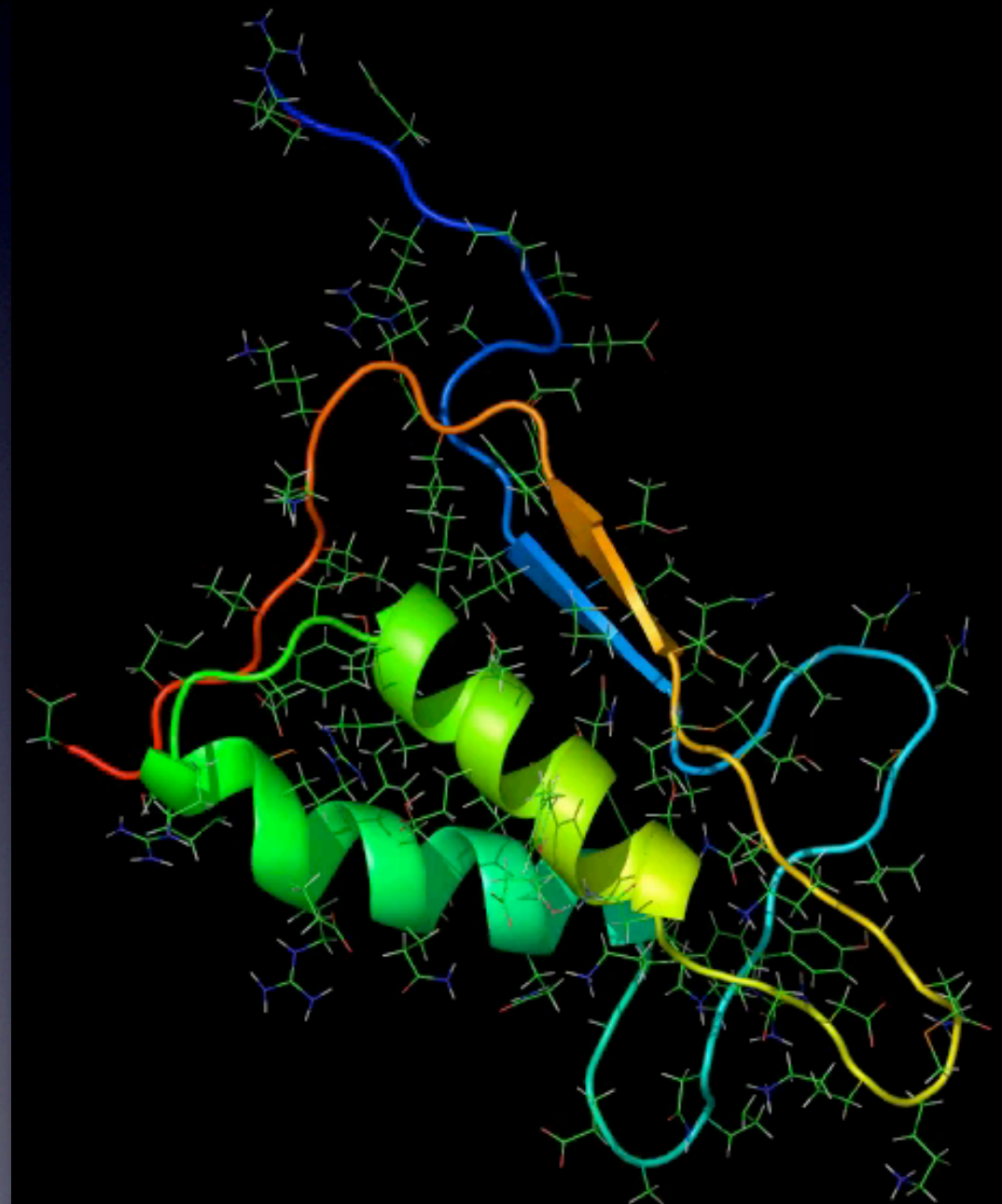
[illegible]

- **Get a structure (usually from PDB)**
- **Fix missing segments, sidechains, etc.**
- **Prepare a Gromacs topology (parameters)**
- **Add solvent water**
- **Energy minimization**
- **Equilibration simulation**
- **Run production simulation**
- **Analyze the output trajectory data**

Energy Minimization



- Find local energy minima
- ...or rather: avoid maxima
- Forces are gradient of V
- Example algorithms:
Steepest descent
Conjugate gradients
L-BFGS
- Normally small changes

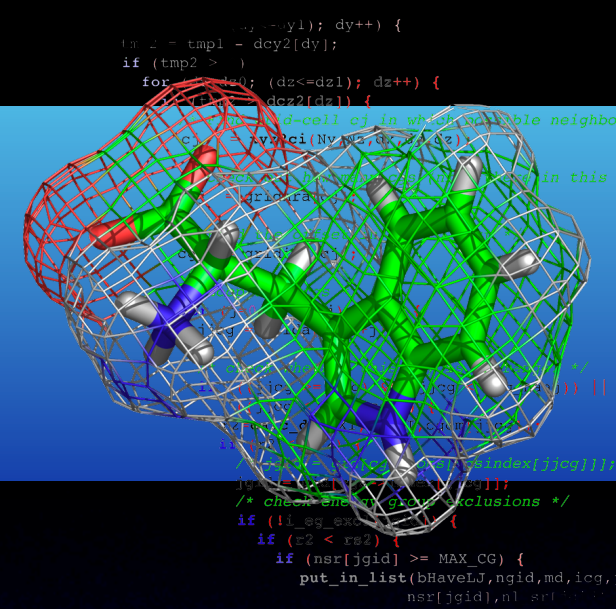



```

        _y1--y1); dy++) {
            tm2 = tmp1 - dcy2[dy];
            if (tmp2 > 0)
                for (dz=0; (dz<=dz1); dz++) {
                    for (dz2=dz1; dz2>=dz; dz2--) {
                        /* find cell of in which possible neighbors
                           are located. (Note: this is not the cell in this
                           grid range) */
                        int i,j,k;
                        int ix=dx+dx1, iy=dy+dy1, iz=dz+dz1;
                        int ix2=ix+1, iy2=iy+1, iz2=iz+1;
                        int ix1=ix-1, iy1=iy-1, iz1=iz-1;
                        int ix3=ix+2, iy3=iy+2, iz3=iz+2;
                        int ix4=ix-2, iy4=iy-2, iz4=iz-2;
                        int ix5=ix+3, iy5=iy+3, iz5=iz+3;
                        int ix6=ix-3, iy6=iy-3, iz6=iz-3;
                        int ix7=ix+4, iy7=iy+4, iz7=iz+4;
                        int ix8=ix-4, iy8=iy-4, iz8=iz-4;
                        int ix9=ix+5, iy9=iy+5, iz9=iz+5;
                        int ix10=ix-5, iy10=iy-5, iz10=iz-5;
                        int ix11=ix+6, iy11=iy+6, iz11=iz+6;
                        int ix12=ix-6, iy12=iy-6, iz12=iz-6;
                        int ix13=ix+7, iy13=iy+7, iz13=iz+7;
                        int ix14=ix-7, iy14=iy-7, iz14=iz-7;
                        int ix15=ix+8, iy15=iy+8, iz15=iz+8;
                        int ix16=ix-8, iy16=iy-8, iz16=iz-8;
                        int ix17=ix+9, iy17=iy+9, iz17=iz+9;
                        int ix18=ix-9, iy18=iy-9, iz18=iz-9;
                        int ix19=ix+10, iy19=iy+10, iz19=iz+10;
                        int ix20=ix-10, iy20=iy-10, iz20=iz-10;
                        int ix21=ix+11, iy21=iy+11, iz21=iz+11;
                        int ix22=ix-11, iy22=iy-11, iz22=iz-11;
                        int ix23=ix+12, iy23=iy+12, iz23=iz+12;
                        int ix24=ix-12, iy24=iy-12, iz24=iz-12;
                        int ix25=ix+13, iy25=iy+13, iz25=iz+13;
                        int ix26=ix-13, iy26=iy-13, iz26=iz-13;
                        int ix27=ix+14, iy27=iy+14, iz27=iz+14;
                        int ix28=ix-14, iy28=iy-14, iz28=iz-14;
                        int ix29=ix+15, iy29=iy+15, iz29=iz+15;
                        int ix30=ix-15, iy30=iy-15, iz30=iz-15;
                        int ix31=ix+16, iy31=iy+16, iz31=iz+16;
                        int ix32=ix-16, iy32=iy-16, iz32=iz-16;
                        int ix33=ix+17, iy33=iy+17, iz33=iz+17;
                        int ix34=ix-17, iy34=iy-17, iz34=iz-17;
                        int ix35=ix+18, iy35=iy+18, iz35=iz+18;
                        int ix36=ix-18, iy36=iy-18, iz36=iz-18;
                        int ix37=ix+19, iy37=iy+19, iz37=iz+19;
                        int ix38=ix-19, iy38=iy-19, iz38=iz-19;
                        int ix39=ix+20, iy39=iy+20, iz39=iz+20;
                        int ix40=ix-20, iy40=iy-20, iz40=iz-20;
                        int ix41=ix+21, iy41=iy+21, iz41=iz+21;
                        int ix42=ix-21, iy42=iy-21, iz42=iz-21;
                        int ix43=ix+22, iy43=iy+22, iz43=iz+22;
                        int ix44=ix-22, iy44=iy-22, iz44=iz-22;
                        int ix45=ix+23, iy45=iy+23, iz45=iz+23;
                        int ix46=ix-23, iy46=iy-23, iz46=iz-23;
                        int ix47=ix+24, iy47=iy+24, iz47=iz+24;
                        int ix48=ix-24, iy48=iy-24, iz48=iz-24;
                        int ix49=ix+25, iy49=iy+25, iz49=iz+25;
                        int ix50=ix-25, iy50=iy-25, iz50=iz-25;
                        int ix51=ix+26, iy51=iy+26, iz51=iz+26;
                        int ix52=ix-26, iy52=iy-26, iz52=iz-26;
                        int ix53=ix+27, iy53=iy+27, iz53=iz+27;
                        int ix54=ix-27, iy54=iy-27, iz54=iz-27;
                        int ix55=ix+28, iy55=iy+28, iz55=iz+28;
                        int ix56=ix-28, iy56=iy-28, iz56=iz-28;
                        int ix57=ix+29, iy57=iy+29, iz57=iz+29;
                        int ix58=ix-29, iy58=iy-29, iz58=iz-29;
                        int ix59=ix+30, iy59=iy+30, iz59=iz+30;
                        int ix60=ix-30, iy60=iy-30, iz60=iz-30;
                        int ix61=ix+31, iy61=iy+31, iz61=iz+31;
                        int ix62=ix-31, iy62=iy-31, iz62=iz-31;
                        int ix63=ix+32, iy63=iy+32, iz63=iz+32;
                        int ix64=ix-32, iy64=iy-32, iz64=iz-32;
                        int ix65=ix+33, iy65=iy+33, iz65=iz+33;
                        int ix66=ix-33, iy66=iy-33, iz66=iz-33;
                        int ix67=ix+34, iy67=iy+34, iz67=iz+34;
                        int ix68=ix-34, iy68=iy-34, iz68=iz-34;
                        int ix69=ix+35, iy69=iy+35, iz69=iz+35;
                        int ix70=ix-35, iy70=iy-35, iz70=iz-35;
                        int ix71=ix+36, iy71=iy+36, iz71=iz+36;
                        int ix72=ix-36, iy72=iy-36, iz72=iz-36;
                        int ix73=ix+37, iy73=iy+37, iz73=iz+37;
                        int ix74=ix-37, iy74=iy-37, iz74=iz-37;
                        int ix75=ix+38, iy75=iy+38, iz75=iz+38;
                        int ix76=ix-38, iy76=iy-38, iz76=iz-38;
                        int ix77=ix+39, iy77=iy+39, iz77=iz+39;
                        int ix78=ix-39, iy78=iy-39, iz78=iz-39;
                        int ix79=ix+40, iy79=iy+40, iz79=iz+40;
                        int ix80=ix-40, iy80=iy-40, iz80=iz-40;
                        int ix81=ix+41, iy81=iy+41, iz81=iz+41;
                        int ix82=ix-41, iy82=iy-41, iz82=iz-41;
                        int ix83=ix+42, iy83=iy+42, iz83=iz+42;
                        int ix84=ix-42, iy84=iy-42, iz84=iz-42;
                        int ix85=ix+43, iy85=iy+43, iz85=iz+43;
                        int ix86=ix-43, iy86=iy-43, iz86=iz-43;
                        int ix87=ix+44, iy87=iy+44, iz87=iz+44;
                        int ix88=ix-44, iy88=iy-44, iz88=iz-44;
                        int ix89=ix+45, iy89=iy+45, iz89=iz+45;
                        int ix90=ix-45, iy90=iy-45, iz90=iz-45;
                        int ix91=ix+46, iy91=iy+46, iz91=iz+46;
                        int ix92=ix-46, iy92=iy-46, iz92=iz-46;
                        int ix93=ix+47, iy93=iy+47, iz93=iz+47;
                        int ix94=ix-47, iy94=iy-47, iz94=iz-47;
                        int ix95=ix+48, iy95=iy+48, iz95=iz+48;
                        int ix96=ix-48, iy96=iy-48, iz96=iz-48;
                        int ix97=ix+49, iy97=iy+49, iz97=iz+49;
                        int ix98=ix-49, iy98=iy-49, iz98=iz-49;
                        int ix99=ix+50, iy99=iy+50, iz99=iz+50;
                        int ix100=ix-50, iy100=iy-50, iz100=iz-50;
                        int ix101=ix+51, iy101=iy+51, iz101=iz+51;
                        int ix102=ix-51, iy102=iy-51, iz102=iz-51;
                        int ix103=ix+52, iy103=iy+52, iz103=iz+52;
                        int ix104=ix-52, iy104=iy-52, iz104=iz-52;
                        int ix105=ix+53, iy105=iy+53, iz105=iz+53;
                        int ix106=ix-53, iy106=iy-53, iz106=iz-53;
                        int ix107=ix+54, iy107=iy+54, iz107=iz+54;
                        int ix108=ix-54, iy108=iy-54, iz108=iz-54;
                        int ix109=ix+55, iy109=iy+55, iz109=iz+55;
                        int ix110=ix-55, iy110=iy-55, iz110=iz-55;
                        int ix111=ix+56, iy111=iy+56, iz111=iz+56;
                        int ix112=ix-56, iy112=iy-56, iz112=iz-56;
                        int ix113=ix+57, iy113=iy+57, iz113=iz+57;
                        int ix114=ix-57, iy114=iy-57, iz114=iz-57;
                        int ix115=ix+58, iy115=iy+58, iz115=iz+58;
                        int ix116=ix-58, iy116=iy-58, iz116=iz-58;
                        int ix117=ix+59, iy117=iy+59, iz117=iz+59;
                        int ix118=ix-59, iy118=iy-59, iz118=iz-59;
                        int ix119=ix+60, iy119=iy+60, iz119=iz+60;
                        int ix120=ix-60, iy120=iy-60, iz120=iz-60;
                        int ix121=ix+61, iy121=iy+61, iz121=iz+61;
                        int ix122=ix-61, iy122=iy-61, iz122=iz-61;
                        int ix123=ix+62, iy123=iy+62, iz123=iz+62;
                        int ix124=ix-62, iy124=iy-62, iz124=iz-62;
                        int ix125=ix+63, iy125=iy+63, iz125=iz+63;
                        int ix126=ix-63, iy126=iy-63, iz126=iz-63;
                        int ix127=ix+64, iy127=iy+64, iz127=iz+64;
                        int ix128=ix-64, iy128=iy-64, iz128=iz-64;
                        int ix129=ix+65, iy129=iy+65, iz129=iz+65;
                        int ix130=ix-65, iy130=iy-65, iz130=iz-65;
                        int ix131=ix+66, iy131=iy+66, iz131=iz+66;
                        int ix132=ix
```

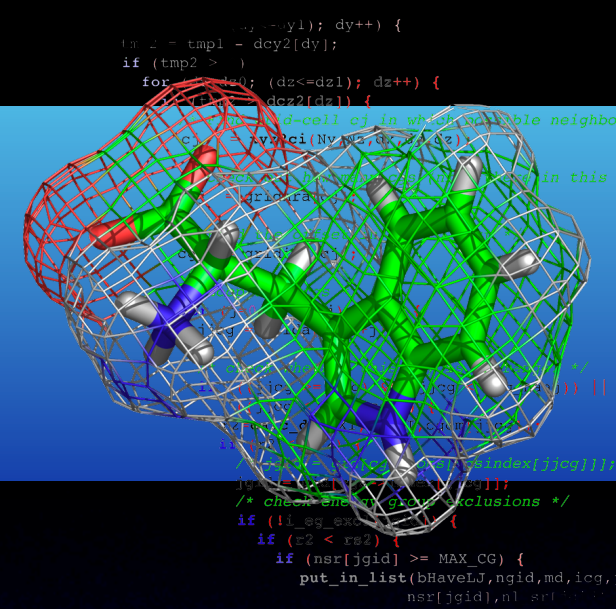
- How do we reproduce experimental conditions like temperature & pressure?
- Temperature = Kinetic energy
Set from Maxwell distribution
- Has to be controlled during simulation, since we lose energy, e.g. due to cut-offs
- Berk will cover this in detail next hour!
- Pressure: Control by adjusting cell size
- Chemical potential: Add/remove particles

Water models



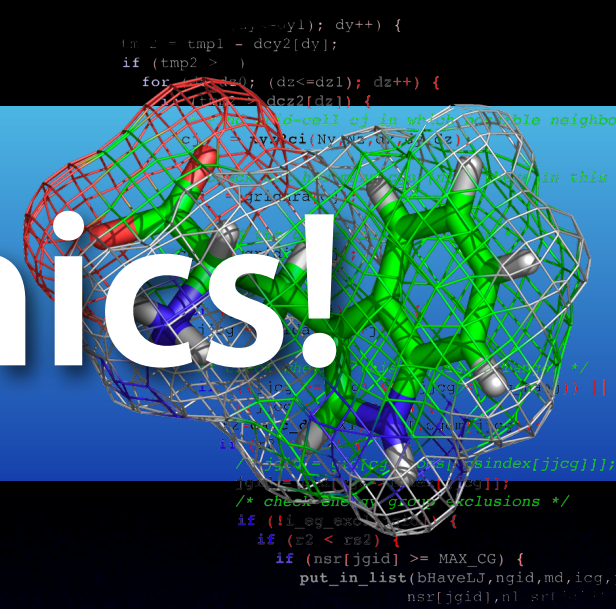
- To speed up simulation we normally use simplified models for water
- Assume water molecules to be rigid to avoid hydrogen vibrations
- Common models: SPC, TIP3P, TIP4P
- There is a wealth of advanced models when you are more interested in water properties *per se* (also polarization)

Limitations of MD



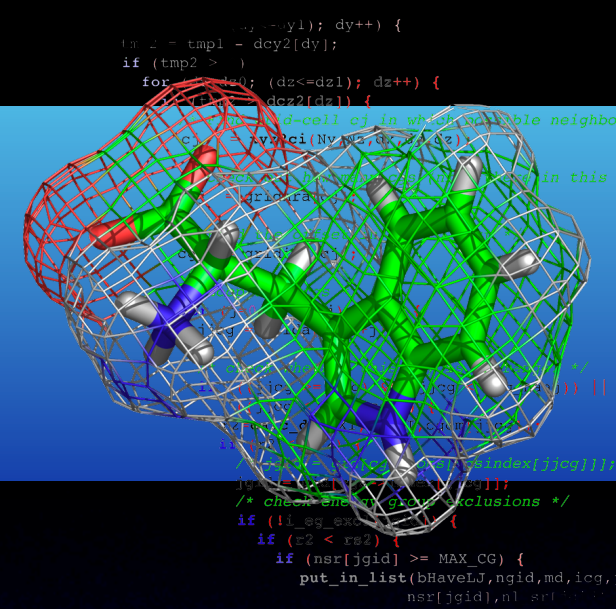
- Parameters are imperfect
- Phase space is not sampled exhaustively
- Example: Free energies of solvation for amino acids often have errors ~ 1 kJ/mol
- Likely impossible to calculate binding free energies more accurately than this
- Limited polarization effects; waters can reorient, but partial charges are fixed

It's statistical mechanics!



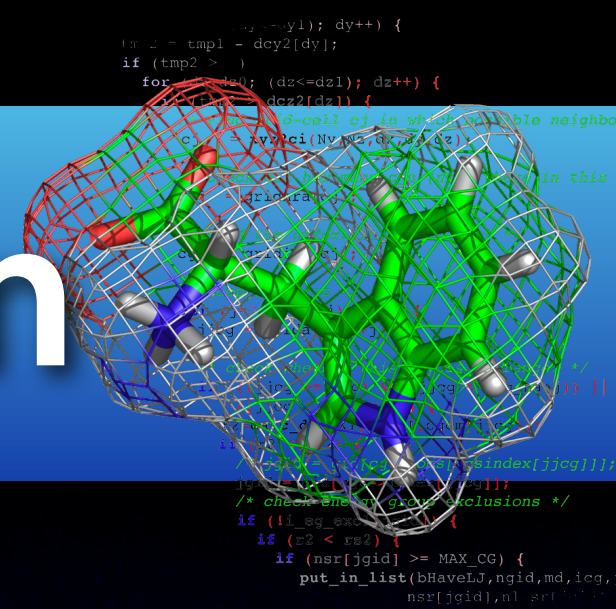
- Remember:
Just because you see something in a simulation does NOT mean it is real
- Everything is about statistics
- When you've seen it 10 times it's significant - a single event is not!
- You should always try to calculate error estimates for predicted properties

MD Programs



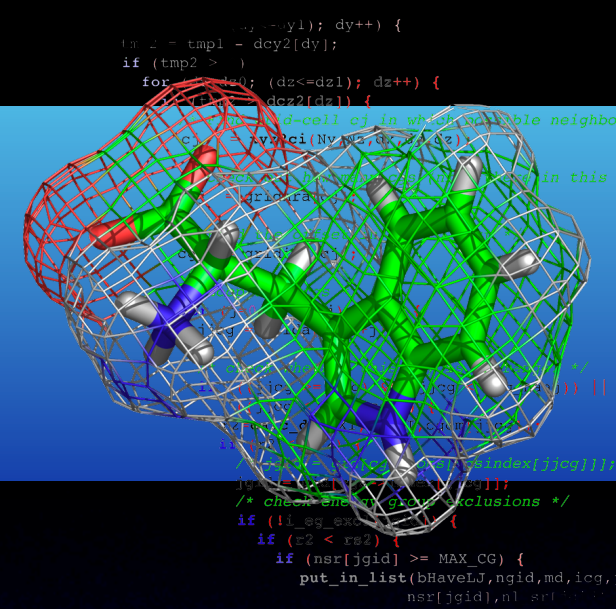
- GROMACS, Amber, Charmm, NAMD, ESPRESSO, Encad, BOSS, LAMMPS
- Programs are often intimately tied to a force field (Amber99, Charmm19, OPLS)
- Force fields supported by Gromacs: GROMOS96, Gromacs, OPLS-AA/L, Amber, Encad, Charmm (beta), etc.
- Try to stick to 1-2 programs, and learn them in detail - you need to motivate your choice of algorithms when publishing!

A typical simulation



- Get a structure (usually from PDB)
- Fix missing segments, sidechains, determine protonation states, etc.
- Prepare a Gromacs topology (parameters)
- Add solvent water
- Energy minimization
- Equilibration simulation
- Run production simulation
- Analyze the output trajectory data

Summary



- Think first, then simulate
- Timescales & limitations
- Empirical classical models
- Interaction forms
- Sampling equilibrium distributions
- Algorithms, approximations, quality
- Flowcharts of typical simulations