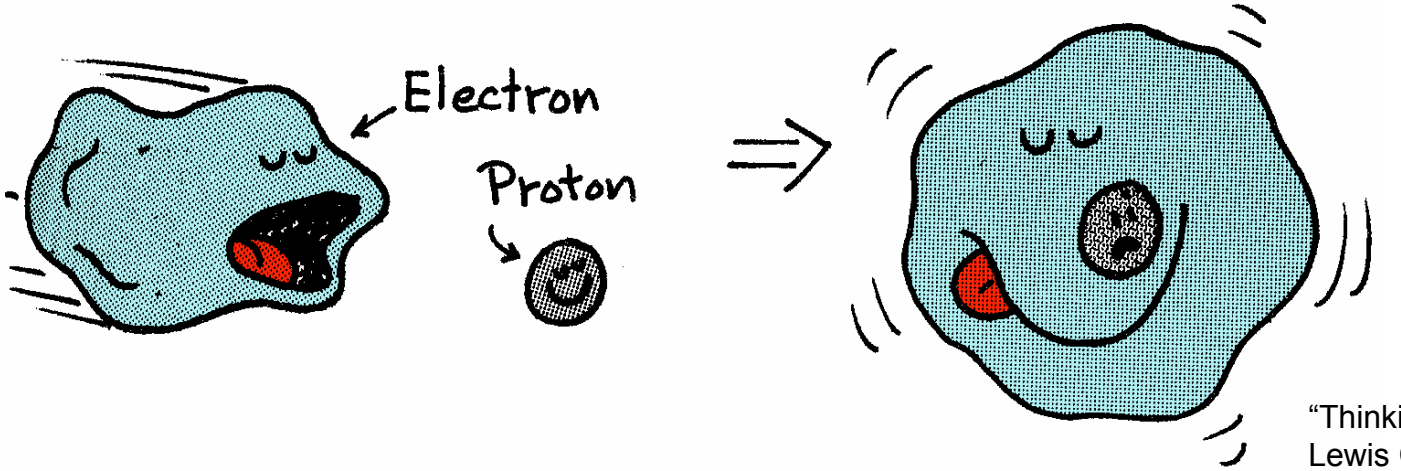
eFF, a force field with electrons



"Thinking Physics", Lewis Carrol Epstein, p. 532

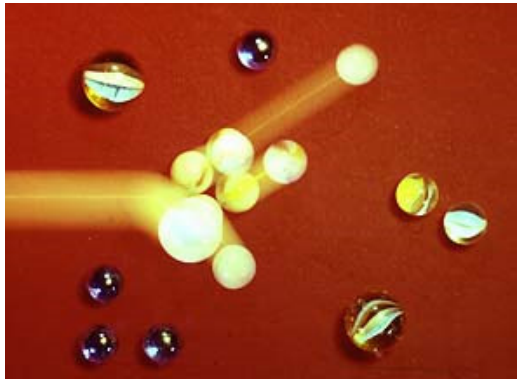
Julius Su and William A. Goddard III

Materials and Process Simulation Center,

California Institute of Technology

eFF, a force field with electrons

Let's make electrons classical objects!



Simplifies the force field, fewer parameters needed.

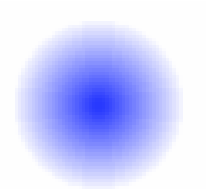
Very large simulations *including electronic effects* practical.

Applications: nuclear fusion, Auger processes, electrochemistry.

eFF, a force field with electrons

Contains only **electrons** and **nuclei** (no atom types, bond types, explicit hybridization …)

Electrons represented by floating Gaussians.



Nuclei represented by point charges.

$$\phi_i(r) \propto e^{-\left((r-r_i)/\rho_i\right)^2}$$

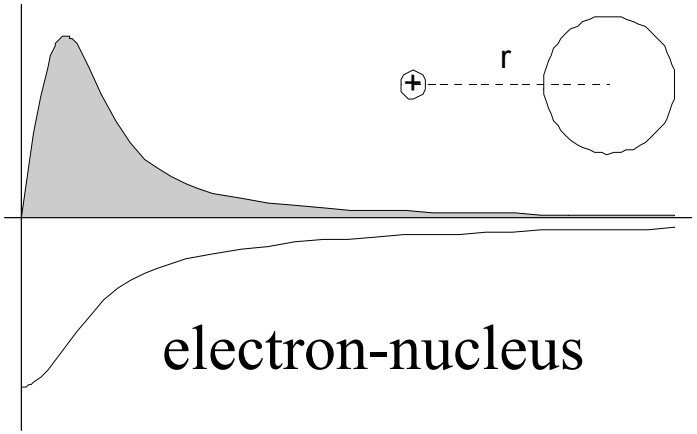
position and extent are continuous dynamical variables

Electrons can have up or down spin.

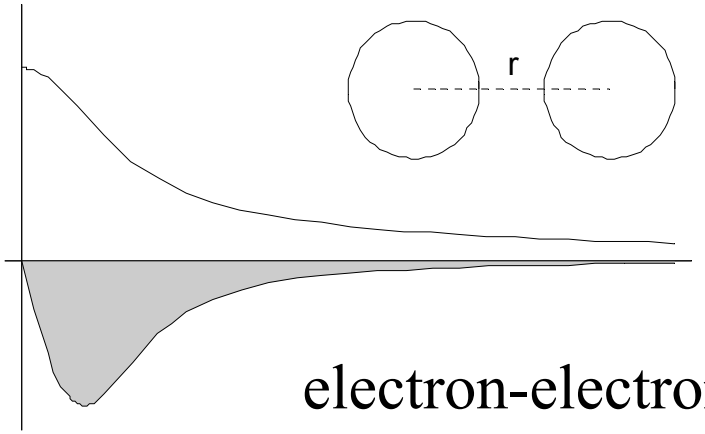
A basic model of electron interaction

Kinetic energy is given by $\frac{3}{2}\frac{1}{r_e^2}$

Electrostatics are screened Coulomb potentials



electron-nucleus

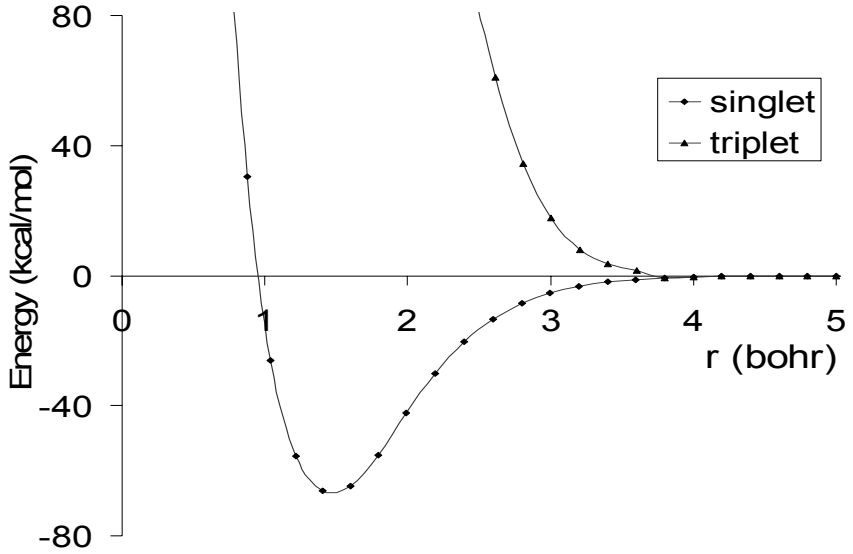


electron-electron

Electrostatic interactions and between same spin electrons, a Pauli repulsion.

A valence bond model for Pauli repulsion between electrons

Pauli from E(triplet)-E(singlet) for H_2 molecule (valence bond wavefunction).



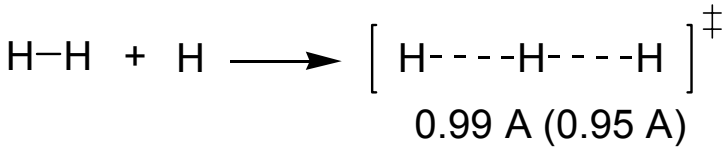
2 parameters added (scaling factors on separation distance and orbital size).

Consider only kinetic energy difference:

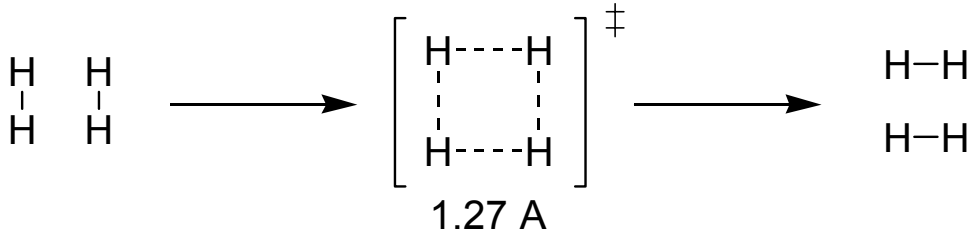
$$\Delta E_{deform} = \frac{\langle 12-21|\hat{H}|12-21\rangle}{\langle 12-21|12-21\rangle} - \frac{\langle 12+21|\hat{H}|12+21\rangle}{\langle 12+21|12+21\rangle}$$

$$= \left(3\left(\frac{1}{r_1^2}+\frac{1}{r_2^2}\right) - 4\left(\frac{3\left(r_1^2+r_2^2\right)-4r^2}{\left(r_1^2+r_2^2\right)^2}\right)\right)\cdot\frac{S^2}{1-S^4}$$

Allowed vs. forbidden reactions

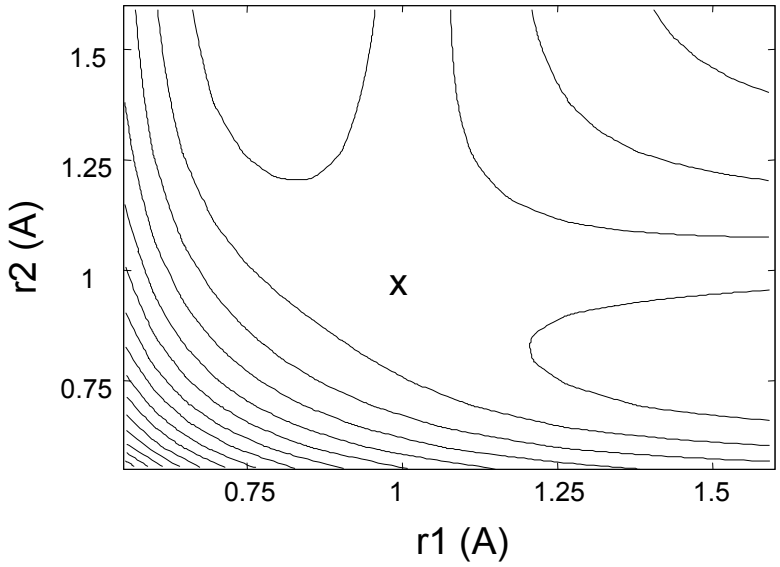


$\Delta E^{\dagger} = 18$ kcal/mol (9.7)

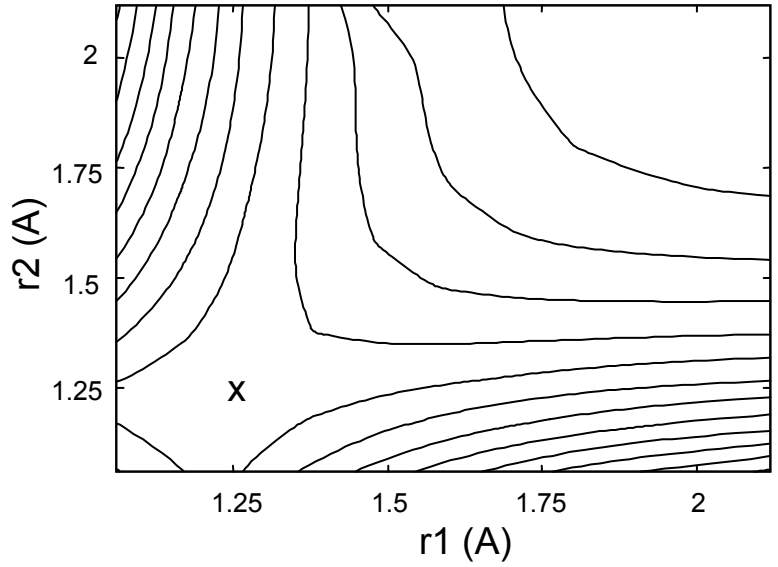


$\Delta E^{\dagger} = 105$ kcal/mol (forbidden)

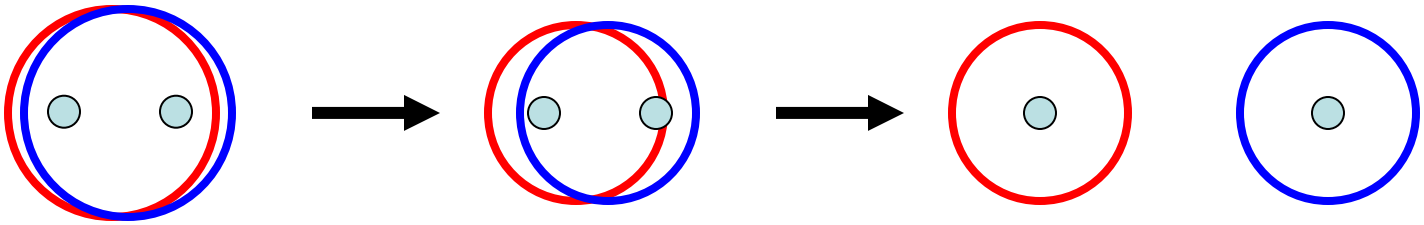
$H_2 + H \rightarrow H + H_2$

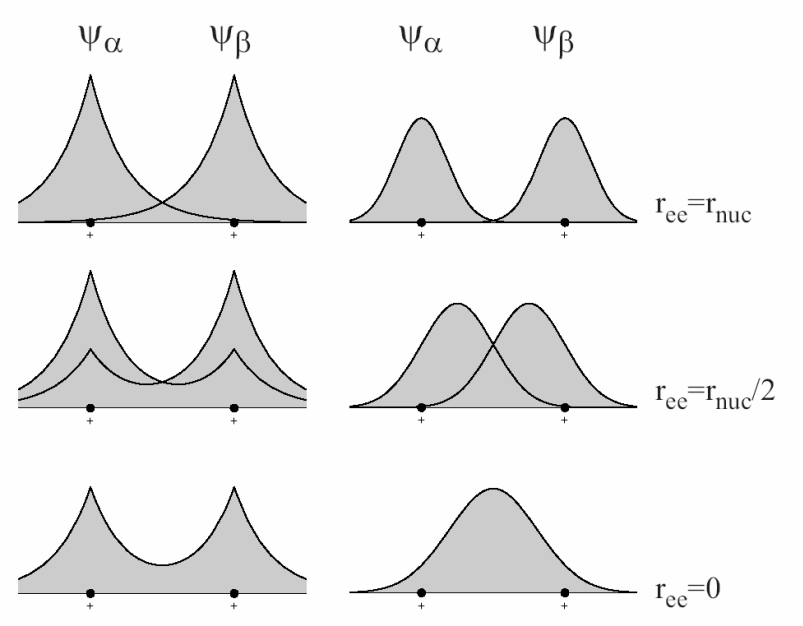


$H_2 + D_2 \rightarrow HD + HD$

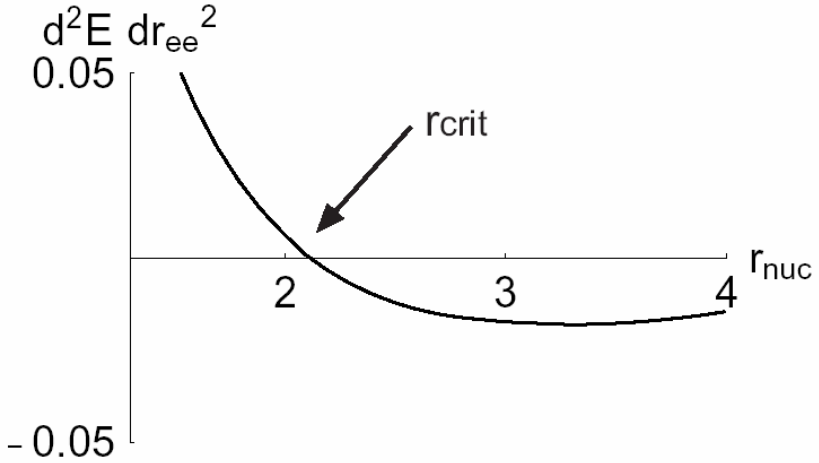


Reproducing transition from closed shell to open shell wavefunction





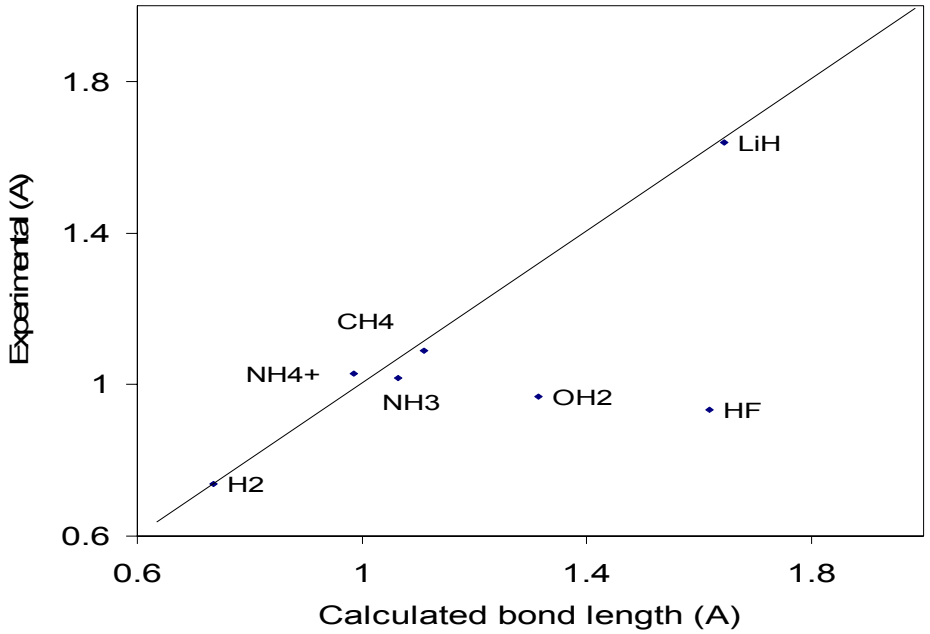
LCAO eFF

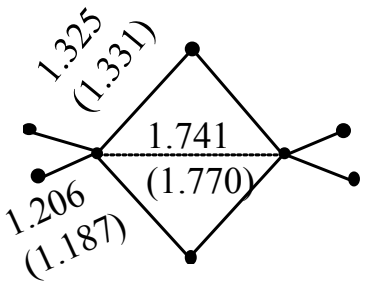


Predicts similar value for electron coalesence

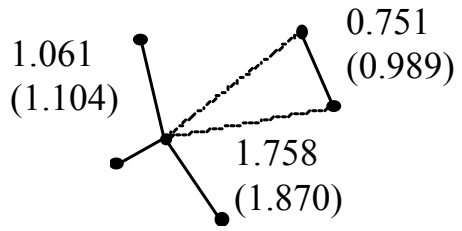
r_{crit} = 2.13 bohr (vs 2.3 bohr UHF)

First-row hydrides and hydrocarbons





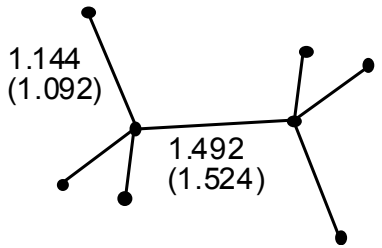
B_2H_6

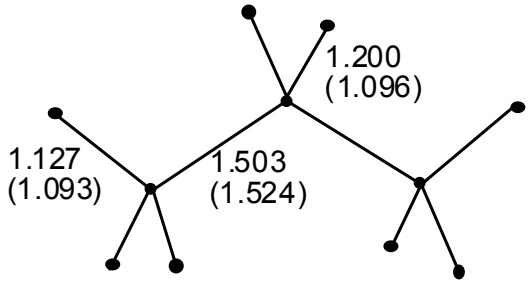


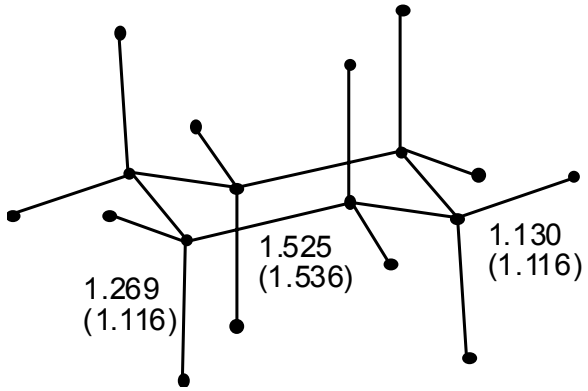
CH_5^+

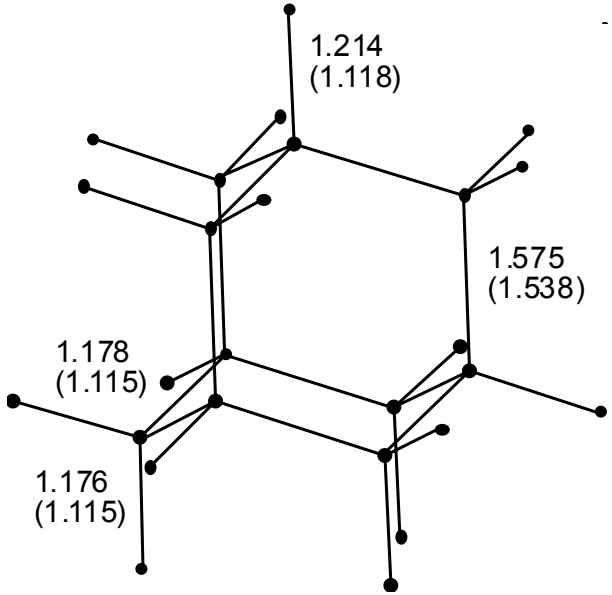
- Molecules without lone pairs work well.
- Hydrides work well too.

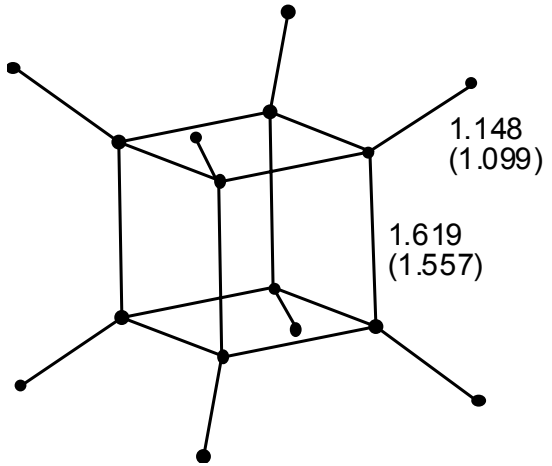
First-row hydrides and hydrocarbons







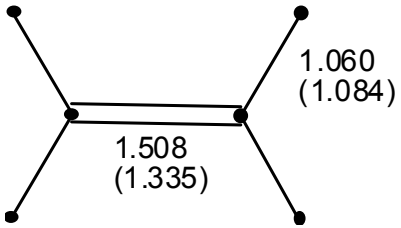


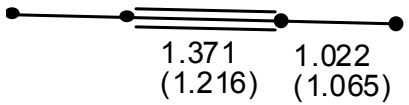


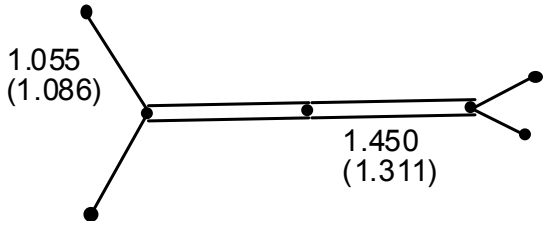
- Good, but axial bonds too long.

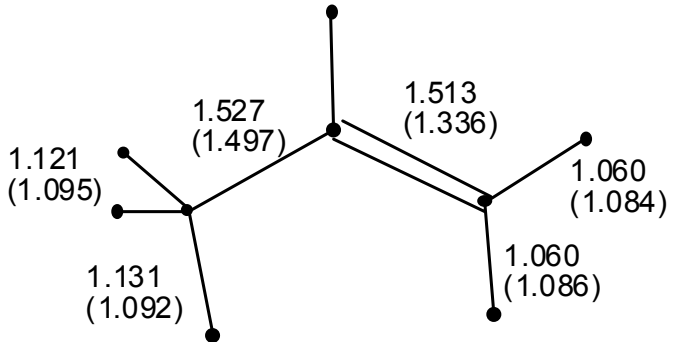
ethane barrier = 2.1 (3.0) kcal/mol
ethane BDE = 140.4 (80) kcal/mol

First-row hydrides and hydrocarbons



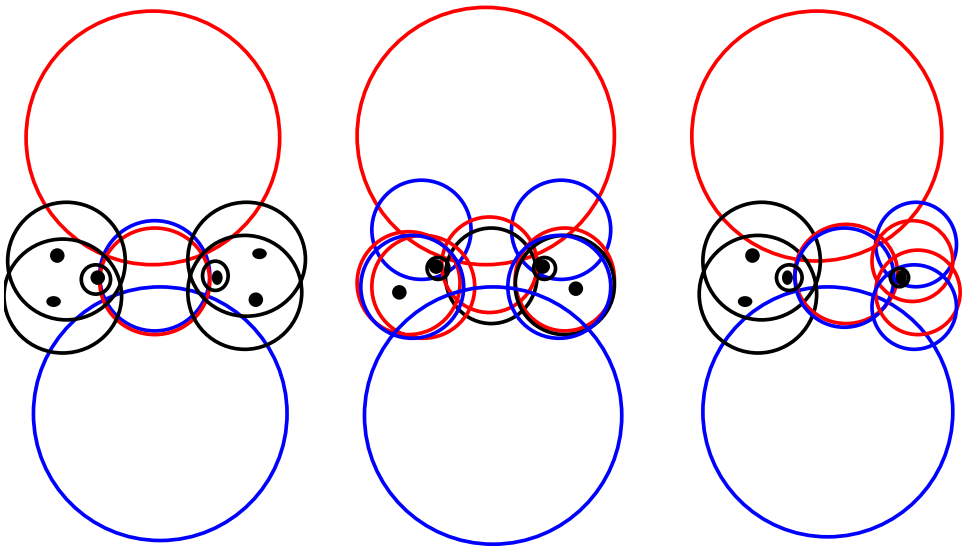






- Sigma-pi bonding preferred over banana bonds.
- Multiple bonds too long.
- Propene shows too much resonance.

Heteroatoms and multiple bonding

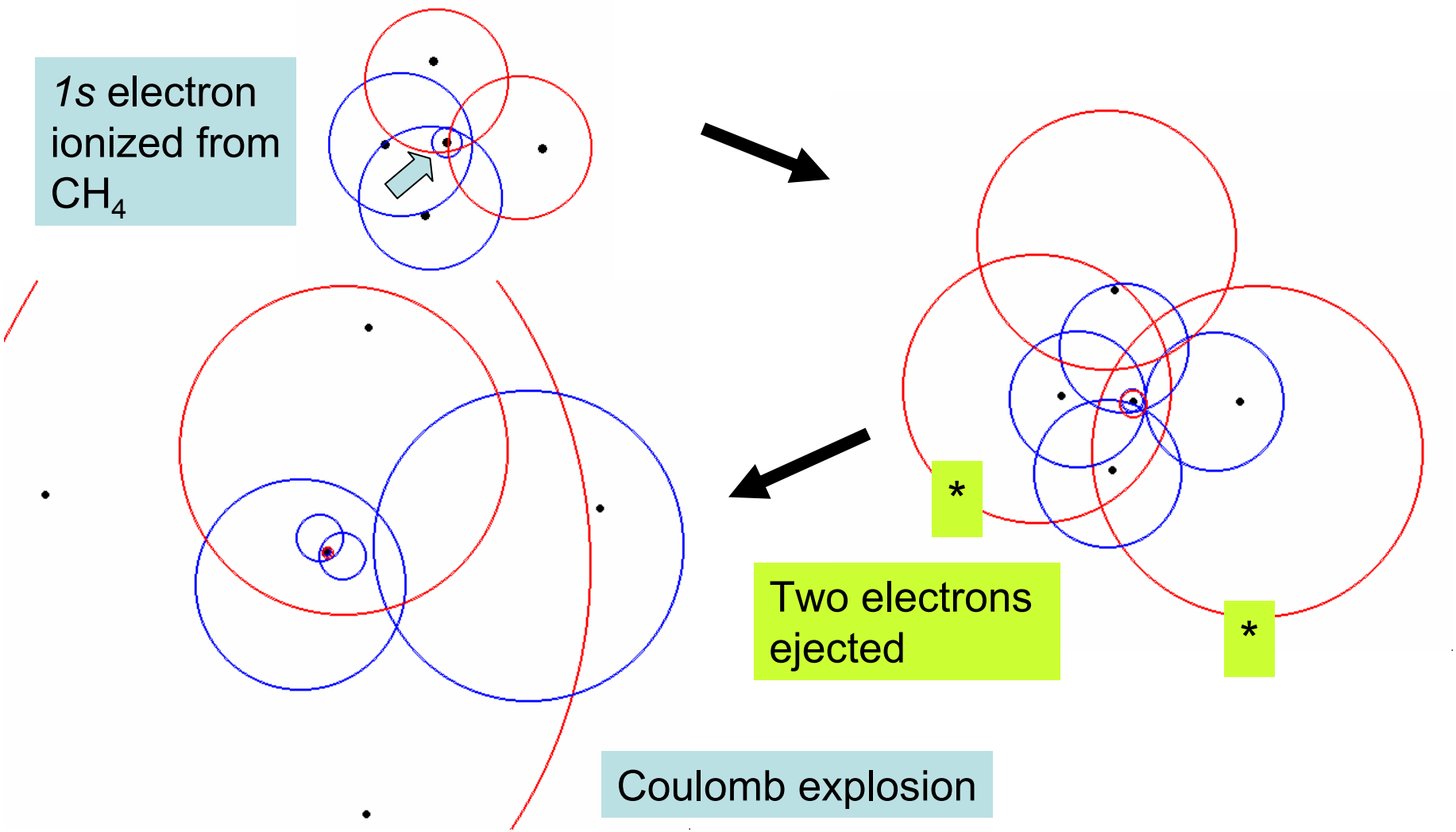


$H_2C{=}CH_2$ $HN{=}NH$ $H_2C{=}O$

	d(C-C) / Å
$H_2C{=}CH$	1.508 (1.335)
$HN{=}NH$	1.378 (1.246)
$HN{=}NH$	1.376 (1.247)
$H_2C{=}O$	1.454 (1.220)

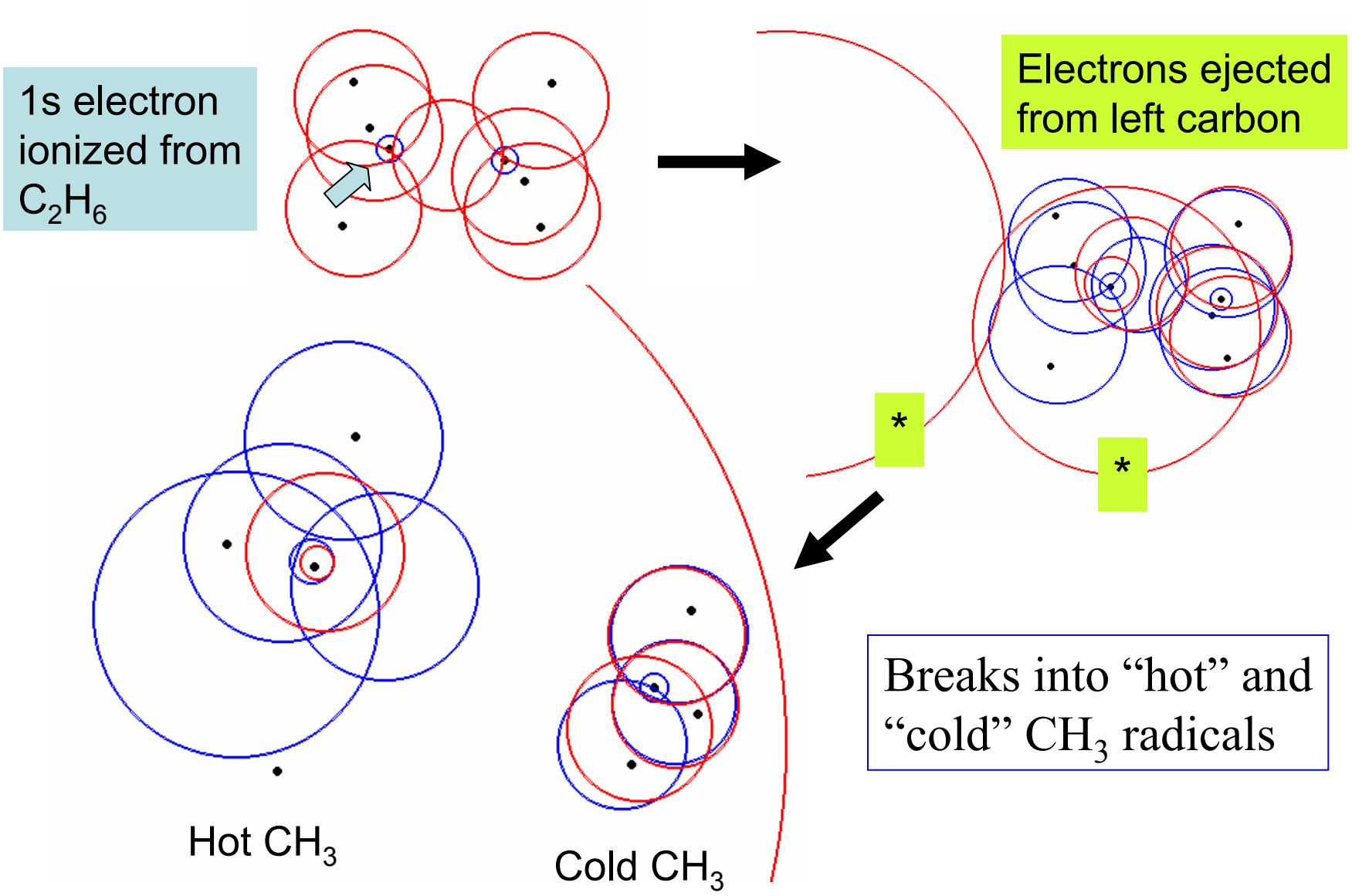
- $H_2C{=}O$ contracts + polarizes double bond.
- Diimide displays unexpected ionization.

Auger process simulated by electron force field (eFF)

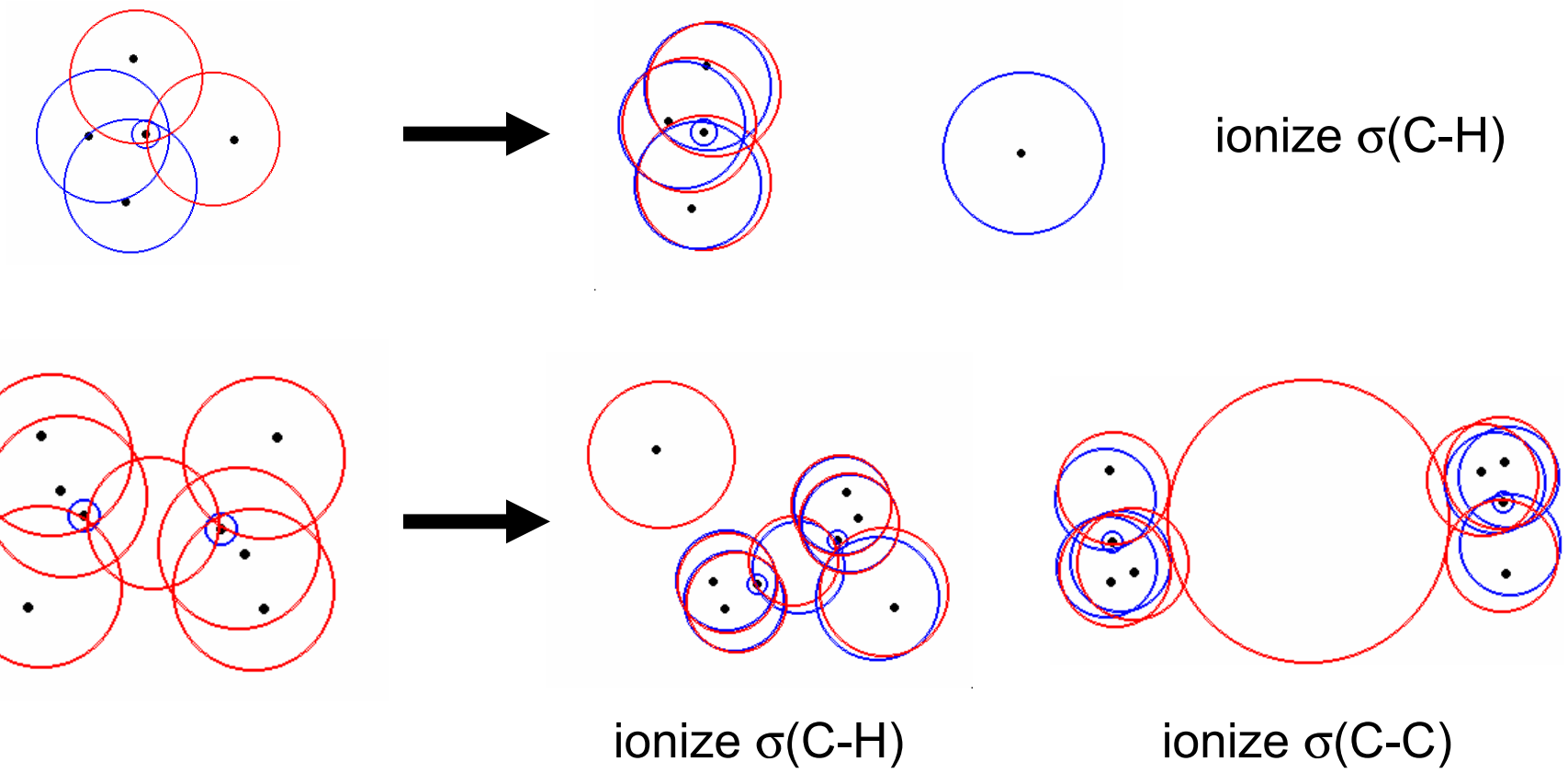


Essential features of Auger process captured

Auger ionization of C_2H_6 leads to C-C cleavage

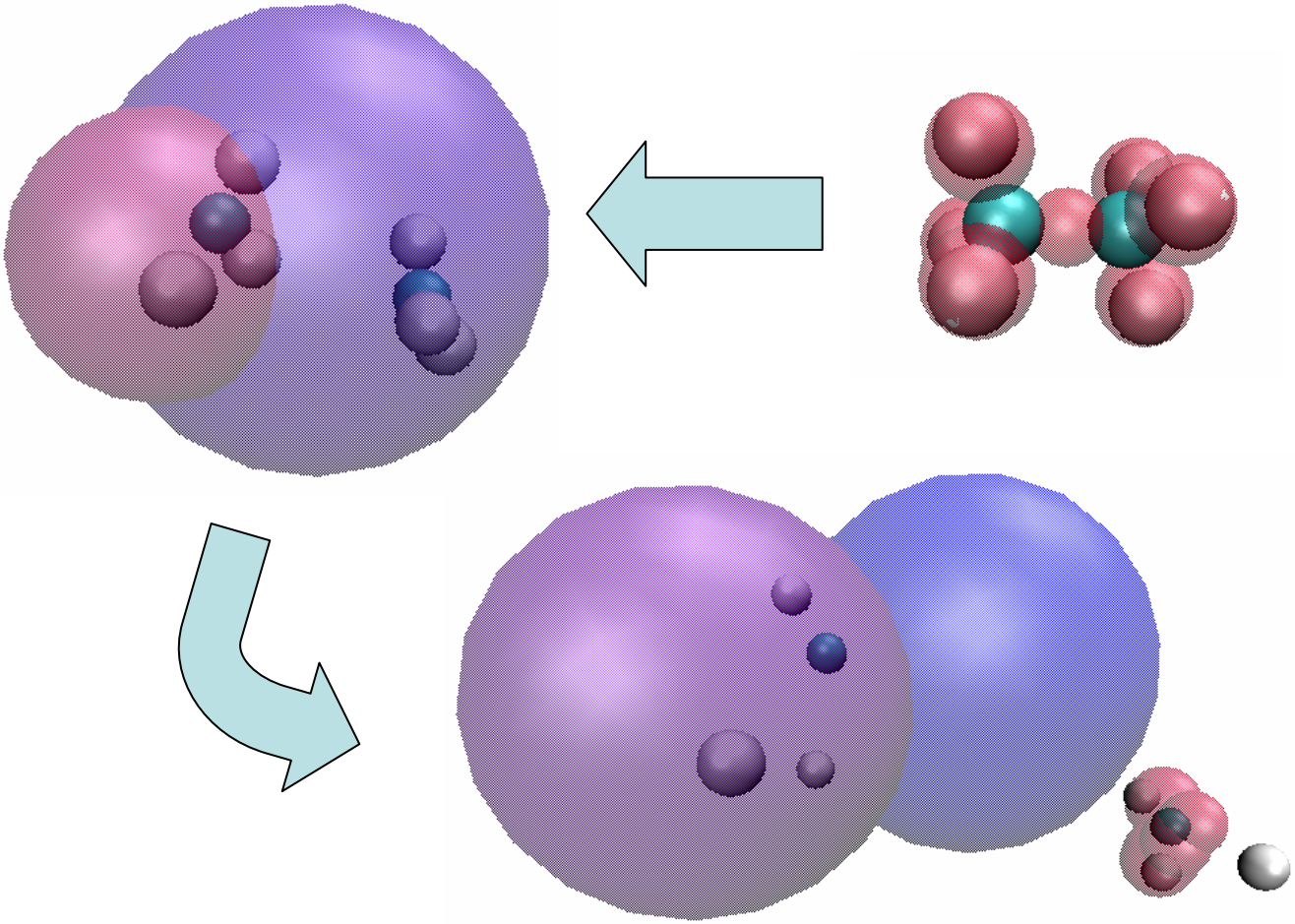


Ionization of bonding electrons in CH_4 and C_2H_6

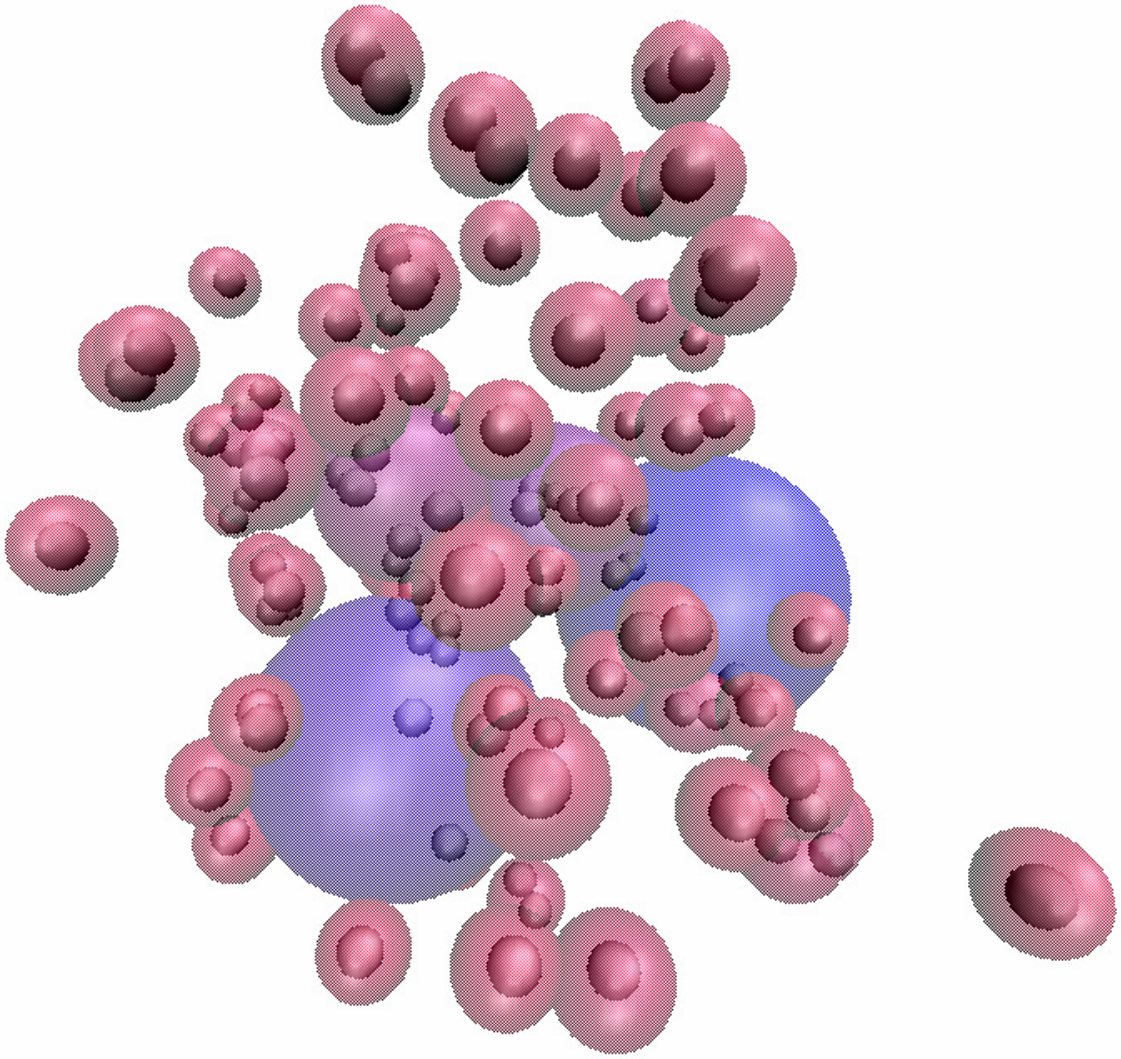


Leads to specific bond cleavage, cold fragments

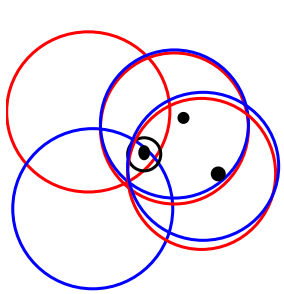
Simulation: Auger dissociation of ethane



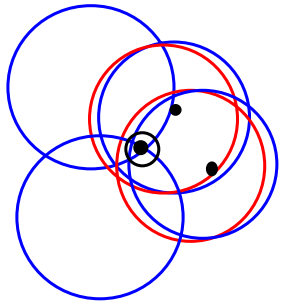
Simulation: snapshot of hydrogen plasma



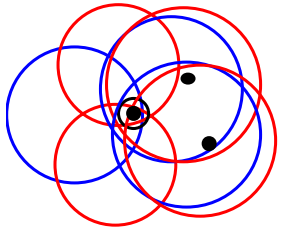
eFF has room for improvement



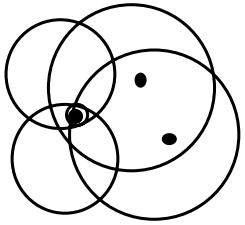
CH_2 (singlet)



CH_2 (triplet)



NH_2



OH_2

Lone pairs, particularly when small, repel other electrons too strongly. Core electrons are not tightly enough bound.

Currently looking into adding electron correlation effects.

CMDF makes it easy to use and integrate eFF

Driver: a hydrogen plasma

```
Initialize(50, 50);
for (i = 0; i < 50; i++)
{
  x = rand_uni() * 50; y = rand_uni() * 50; z = rand_uni() * 50;
  AddNucleus(x, y, z, 1);
  AddElectron(x, y, z, 1, sign(rand_uni() * 2 - 1));
}

InitializeDynamics();
SetTemperature(tempearture);
for (i = 0; i < numiter; i++)
  Dynamics();

SaveNucXYZ(xyzfile);
SaveElectrons(elecfile);
```

Accessors

```
double GetTotalEnergy();
double GetNucleusEnergy(int i);
double GetElectronEnergy(int i);
void GetNuclearForce(int i, double *fx, double *fy, double *fz);
void GetElectronForce(int i, double *fx, double *fy, double *fz, double *fr);
void GetElectronPosition(int i, double *x, double *y, double *z, double *r);
void SetElectronPosition(int i, double x, double y, double z, double r);
void GetNuclearPosition(int i, double *x, double *y, double *z);
void SetNuclearPosition(int i, double x, double y, double z);
double GetNuclearCharge(int i);
int NumNuclei();
int NumElectrons();
void AddNuclearForce(int i, double fx, double fy, double fz);
void AddElectronForce(int i, double fx, double fy, double fz, double fr);
```

Creators

```
void Initialize(int s_maxnuclei, int s_maxelectrons);
void AddNucleus(double x, double y, double z, double q);
void AddElectron(double x, double y, double z, int spin, double re);
void AddConfiguration(double x, double y, double z, char *filname);
void AddAtom(double x, double y, double z, char *name);
```