**Physics 160FP**

**Kinnard Hockenhull**

**Models of Molecular Motions**

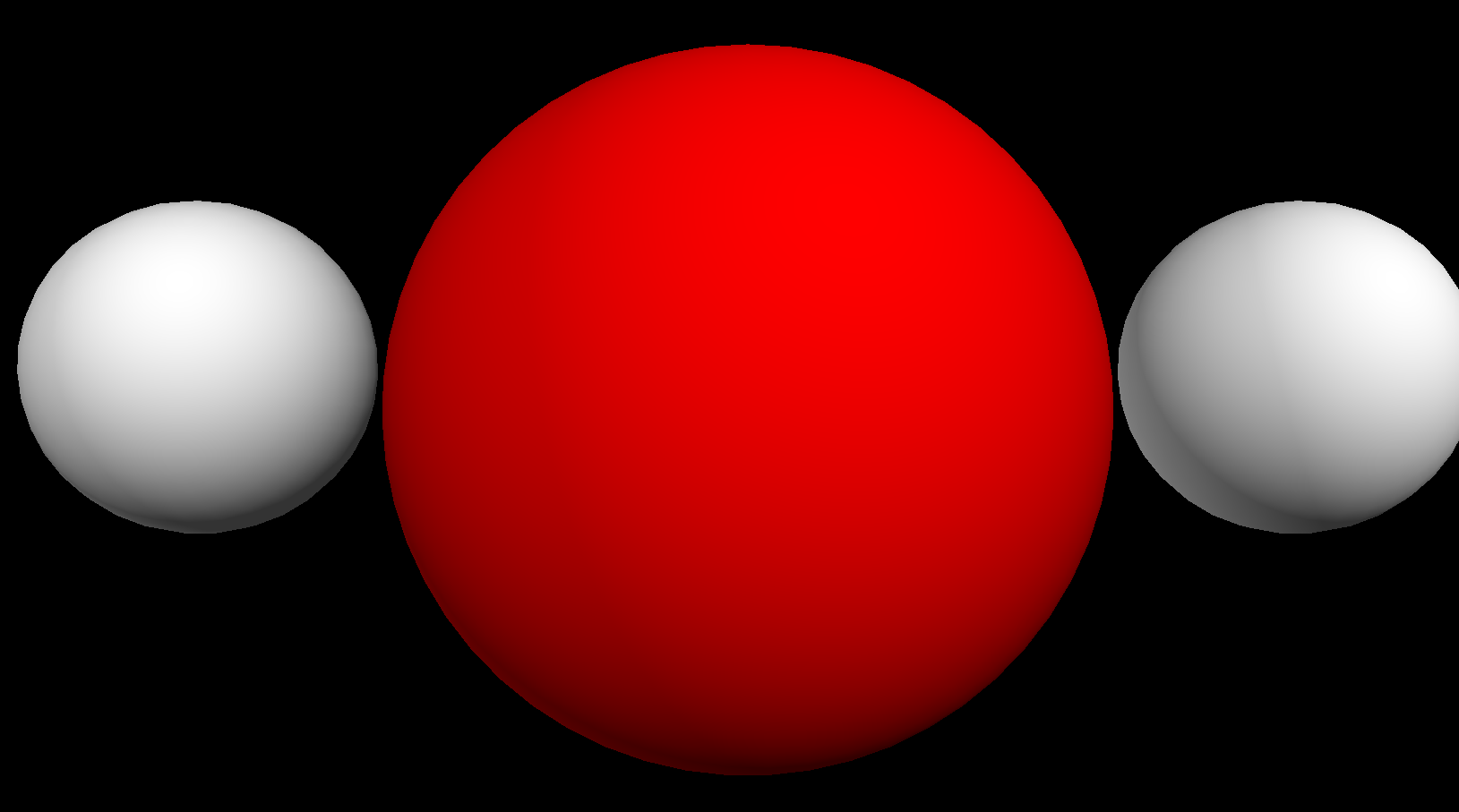
Of all the fundamental forces governing our world, those that govern intramolecular and intermolecular interaction are perhaps the most pervasive. They appear as the force of magnets, and the electronics which depend on them, but also as the magnetosphere of the Earth and the beautiful aurora it produces. They appears as the forces that governs chemical interactions. All of these forces are actually all incarnations of the electrostatic force. Curiously, it is the force which governs springs on the macroscopic level and chemical bonds on the nanoscopic level. This curious coincidence of physical phenomena is one of great consequence for it allows us to simply yet elegantly describe the mechanics of molecules and glean much other understanding from the Universe. In this paper I will model interparticle interactions using three different but related vPython set-ups.

**Models**

http://upload.wikimedia.org/wikipedia/commons/thumb/b/b7/H2O_2D_labelled.svg/135px-H2O_2D_labelled.svg.pngIn the interest of computational parsimony, the elegance of the model, and my own sanity several assumptions have been made. Some will be shed as we progress through the models. Atoms are treated as massive, sometimes charged point particles represented by spheres. This contrasts with the currently understood reality explained by quantum physics, not only in complexity but also in computability. Bonds are modeled variously using springs, the coulumb force, and the Lennard-Jones Potential.

I chose to model water because of its simplicity, its pervasiveness, and it centrality to human experience and also because of its role as a solvent in easily(or so I thought) modeled ionic interactions.

**Hooke Model**

My water model is composed of three particles, one oxygen and two hydrogens and three springs. The hydrogens are attached to the oxygens by springs with 95.84pm as the equilibrium length, as this is the equilibrium length of the O-H bond in water. There is a third imaginary spring, which does not represent a bond running from Hydrogen to Hydrogen. This spring, I believe, accounts for the lone pairs on oxygen which give water its bond angle, but which would not appear in a model treating atoms as point particles. 

The force of each spring is given by a Hook’s Law: . The spring constants of the springs were derived from their respective angular frequencies of oscillation, which were available from experimentation and from this semesters lecture notes, using the relation . This model allows for intramolecular behavior such as bond length and angle oscillation to be understood but it does not account for intermolecular interactions.

**Coulumb Model**

In order to model interatomic interactions, the particles of water were given partial charges which sum to 0 (arbitrarily, δO=-.8, δH=+.4) and the force due to these charges was modeled using Coulumb’s Law:  rather than springs. This model accounts for attractive forces between differently charged particles, and repulsive forces between similarly charged particles but it does not account for restoring forces in each case.

In order to account for both the attractive and repulsive forces between particles, I extended my model to include forces of temporary intratomic shifts in charge distribution know as dipole moments known as Van der Waals forces using the Lennard Jones Potential:. This model was developed by John Lennard-Jones in the 1920’s at Cambridge. It is widely used because of its analytical and computational simplicity.

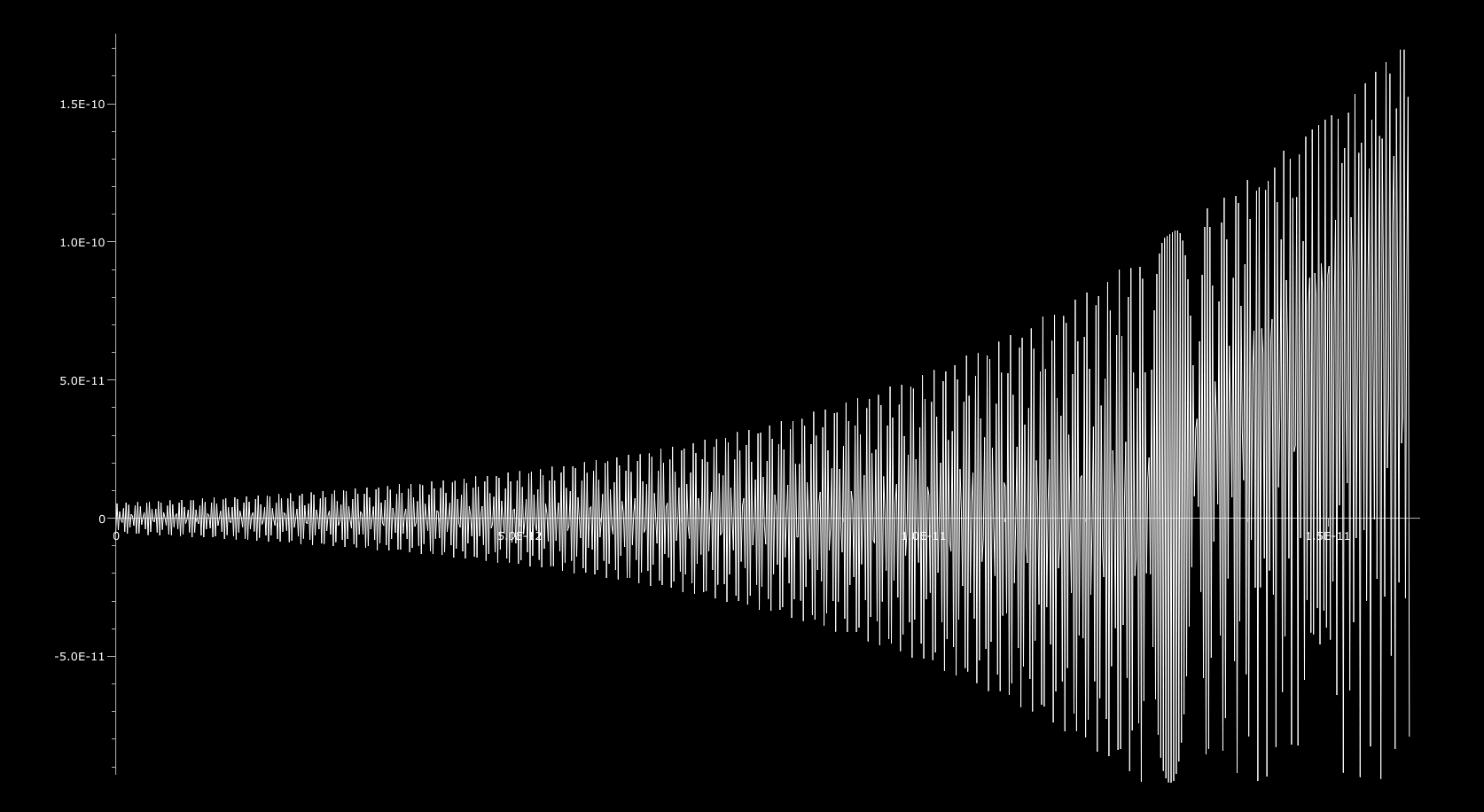
**Neutral**

δ

**+ ⇒-**

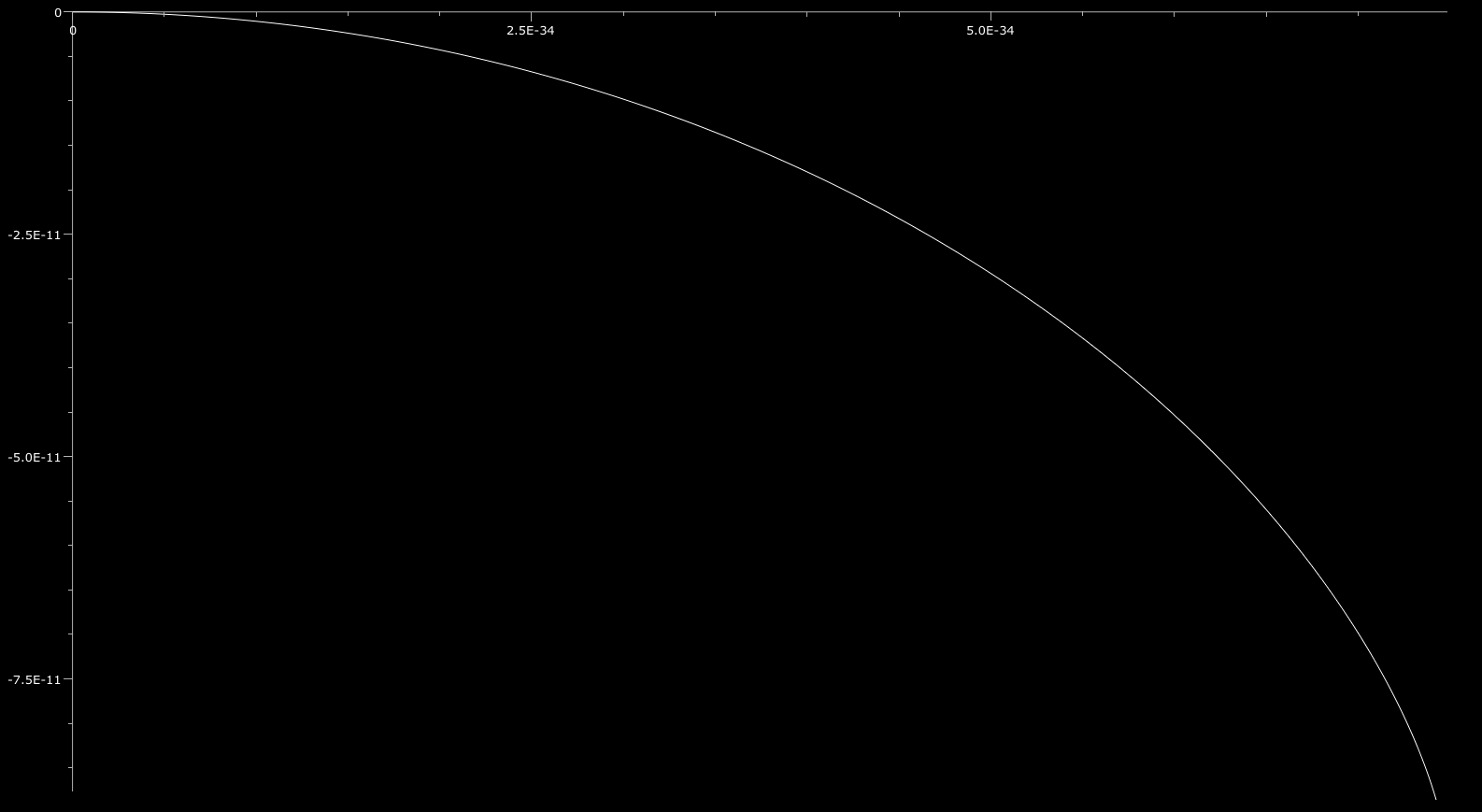
**Results**

The spring model starts off fine except that if run long enough, the oscillations of the O-H bond length grow and do so in such a way that indicates my spring water model is inaccurate as a model for nature. This may be a product of cumulative rounding error rather than modeling error and can be corrected for by using a smaller time step.



Graph of O-H bond length in Hook Model oscillating out of control.

The Coulomb model seems to pull in the hydrogens, as if in a pinball machine only for them to be shot away. This makes sense because of the absence of a restorative force.

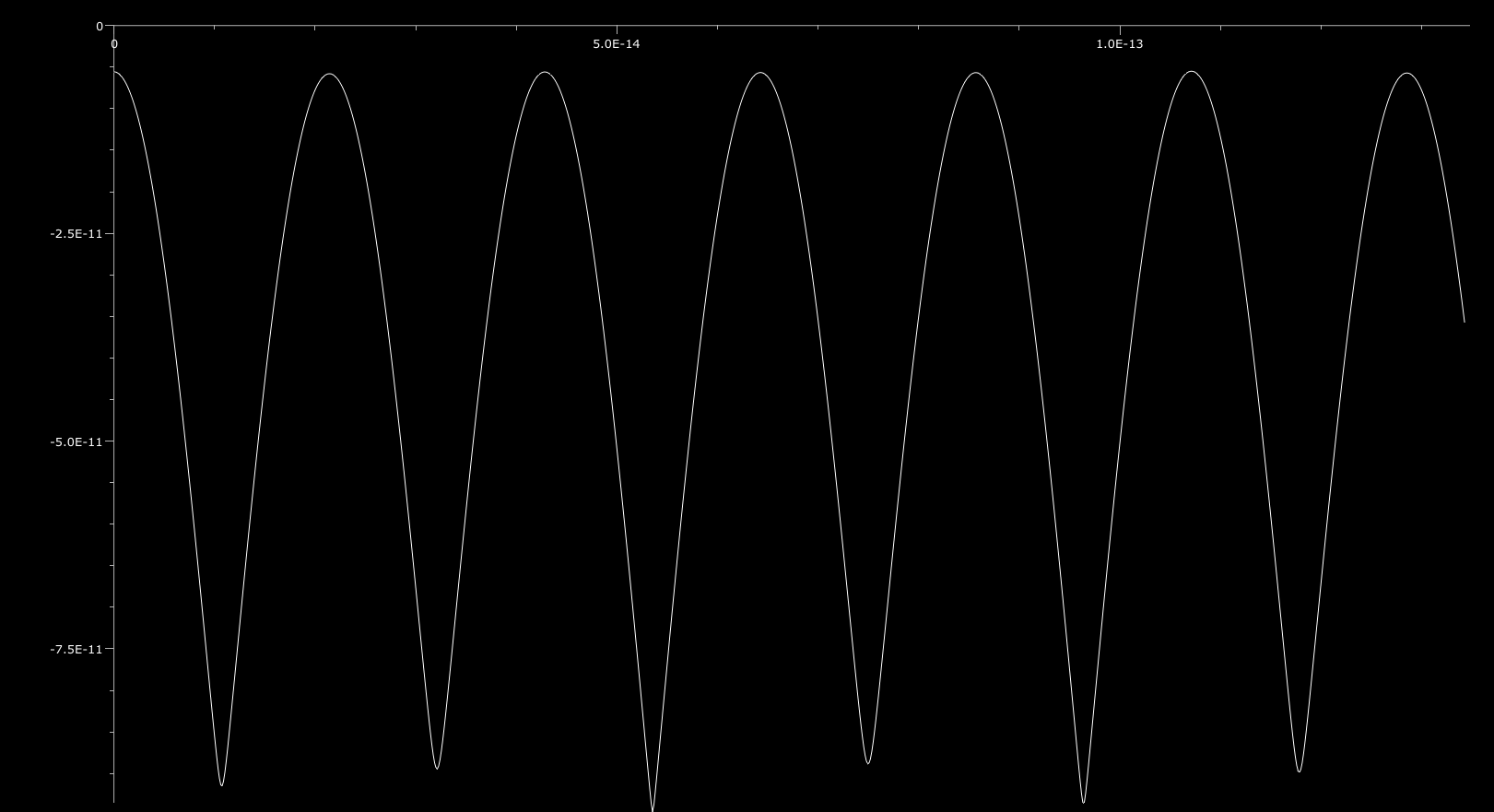


Falling in . . .



Flying away!

The final model, using the Lennard-Jones Potential seems to be the best, as it displays neither the instability of the Coulumb model or the eventual decay of the spring model. I had hoped to also include a model involving an ionic interaction but this hope was snuffed out by the pernicious n-body problem.

Oscillations of the Lennard-Jones Potential Model.

**Code**

**Hooke**

#Tweaked for visuals

from \_\_future\_\_ import division

from visual import \*

from visual.graph import \* # import graphing features

O=Oxygen=sphere(pos=vector(0,0,0),radius=10\*.66e-11, color=color.red)

#H1=Hydrogen1=sphere(pos=vector(95.564e-12,7.2661e-12,0),radius=10\*.31e-11)

#H2=Hydrogen2=sphere(pos=vector(-95.564e-12,7.2661e-12,0),radius=10\*.31e-11)

#With a Kick

H1=Hydrogen1=sphere(pos=vector(90e-12,7e-12,0),radius=10\*.31e-11)

H2=Hydrogen2=sphere(pos=vector(-90e-12,7e-12,0),radius=10\*.31e-11)

mO=15.99/(6.022e26)

mH=1.0079/(6.022e26)

#Given from experimentation

k=50

upp=5e-5

pO=mO\*vector(0, 0, 0)

pH1=mH\*vector(0, 0, 0)

pH2=mH\*vector(0, 0, 0)

graph1 = gcurve(color=color.white)

#Force Modeled by a spring

t=0

dt=15e-18

while t<5e-11:

t=t+dt

Fl1=-k\*(mag(H1.pos-O.pos)-95.84e-12)\*norm(H1.pos-O.pos)

Fl2=-k\*(mag(H2.pos-O.pos)-95.84e-12)\*norm(H2.pos-O.pos)

Fa=upp\*(mag(H1.pos-H2.pos)-191.128e-12)\*norm(H1.pos-H2.pos)

H1.pos=H1.pos+((pH1)/mH)\*dt

H2.pos=H2.pos+((pH2)/mH)\*dt

pH1=pH1+(Fl1-Fa)\*dt

pH2=pH2+(Fl2+Fa)\*dt

graph1.plot(pos=(t,mag(H1.pos-O.pos)-95.84e-12))

**Coulomb**

from \_\_future\_\_ import division

from visual import \*

from visual.graph import \* # import graphing features

O=Oxygen=sphere(pos=vector(0,0,0),radius=10\*.66e-11, color=color.red)

H1=Hydrogen1=sphere(pos=vector(95.564e-12,7.2661e-12,0),radius=10\*.31e-11)

H2=Hydrogen2=sphere(pos=vector(-95.564e-12,7.2661e-12,0),radius=10\*.31e-11)

#With a Kick

#H1=Hydrogen1=sphere(pos=vector(90e-12,7e-12,0),radius=10\*.31e-11)

#H2=Hydrogen2=sphere(pos=vector(-90e-12,7e-12,0),radius=10\*.31e-11)

mO=15.99/(6.022e26)

mH=1.0079/(6.022e26)

qO=.8

qH=.4

qNa=1

qCl=-1

pO=mO\*vector(0, 0, 0)

pH1=mH\*vector(0, 0, 0)

pH2=mH\*vector(0, 0, 0)

graph1 = gcurve(color=color.white)

#Force modeled electrostatically

#Fe=K\*qO\*qH/(H.pos-O.pos)^2

ke=8.987551e+9 #Coulomb's Constant

t=0

dt=1.15e-38

while t<5e-12:

t=t+dt

FOH1=((ke\*qO\*qH)/(mag(H1.pos-O.pos))\*\*2)\*-norm(H1.pos-O.pos) #Force of Oxygen on Hydrogen1

FOH2=((ke\*qO\*qH)/(mag(H1.pos-O.pos))\*\*2)\*-norm(H2.pos-O.pos)

FH1O=((ke\*qO\*qH)/(mag(O.pos-H1.pos))\*\*2)\*-norm(O.pos-H1.pos) #=-FOH1

FH2O=-((ke\*qO\*qH)/(mag(O.pos-H2.pos))\*\*2)\*norm(O.pos-H2.pos)

FH1H2=-((ke\*qH\*qH)/(mag(H2.pos-H1.pos))\*\*2)\*norm(H2.pos-H1.pos)

FH2H1=-((ke\*qH\*qH)/(mag(H1.pos-H2.pos))\*\*2)\*norm(H1.pos-H2.pos)

H1.pos=H1.pos+((pH1)/mH)\*dt

H2.pos=H2.pos+((pH2)/mH)\*dt

O.pos=O.pos+((pO)/mO)\*dt

pH1=pH1+(FOH1+FH2H1)\*dt

pH2=pH2+(FOH2+FH1H2)\*dt

pO=pO+(FH1O+FH2O)\*dt

graph1.plot(pos=(t,mag(H1.pos-O.pos)-95.84e-12))

**Lennard-Jones**

from \_\_future\_\_ import division

from visual import \*

from visual.graph import \* # import graphing features

O=Oxygen=sphere(pos=vector(0,0,0),radius=10\*.66e-11, color=color.red)

#H1=Hydrogen1=sphere(pos=vector(95.564e-12,7.2661e-12,0),radius=10\*.31e-11)

#H2=Hydrogen2=sphere(pos=vector(-95.564e-12,7.2661e-12,0),radius=10\*.31e-11)

#With a Kick

H1=Hydrogen1=sphere(pos=vector(90e-12,7e-12,0),radius=10\*.31e-11)

H2=Hydrogen2=sphere(pos=vector(-90e-12,7e-12,0),radius=10\*.31e-11)

mO=15.99/(6.022e26)

mH=1.0079/(6.022e26)

qO=.8

qH=.4

qNa=1

qCl=-1

pO=mO\*vector(0, 0, 0)

pH1=mH\*vector(0, 0, 0)

pH2=mH\*vector(0, 0, 0)

graph1 = gcurve(color=color.white)

#Force modeled using Lennard-Jones potential

#F=-12Ar^-13+6Br^-7

A=1

B=2

t=0

dt=1e-18

while t<5e-12:

t=t+dt

FOH1=(-1)\*(-12\*A\*(mag(H1.pos-O.pos)\*\*13)+6\*B\*(mag(H1.pos-O.pos))\*\*(-7))\*norm(H1.pos-O.pos) #Force of Oxygen on Hydrogen1

FOH2=-1\*(-12\*A\*(mag(H2.pos-O.pos)\*\*13)+6\*B\*(mag(H2.pos-O.pos))\*\*(-7))\*norm(H2.pos-O.pos)

FH1O=-1\*(-12\*A\*(mag(O.pos-H1.pos)\*\*13)+6\*B\*(mag(O.pos-H1.pos))\*\*(-7))\*norm(O.pos-H1.pos)

FH2O=-1\*(-12\*A\*(mag(O.pos-H2.pos)\*\*13)+6\*B\*(mag(O.pos-H2.pos))\*\*(-7))\*norm(O.pos-H2.pos)

FH1H2=-1\*(-12\*A\*(mag(H2.pos-H1.pos)\*\*13)+6\*B\*(mag(H2.pos-H1.pos))\*\*(-7))\*norm(H2.pos-H1.pos)

FH2H1=-1\*(-12\*A\*(mag(H1.pos-H2.pos)\*\*13)+6\*B\*(mag(H1.pos-H2.pos))\*\*(-7))\*norm(H1.pos-H2.pos)

H1.pos=H1.pos+((pH1)/mH)\*dt

H2.pos=H2.pos+((pH2)/mH)\*dt

pH1=pH1+(FOH1+FH2H1)\*dt

pH2=pH2+(FOH2+FH1H2)\*dt

pO=pO+(FH1O+FH2O)\*dt

graph1.plot(pos=(t,mag(H1.pos-O.pos)-95.84e-12))

**Ion Interaction**

#Tweaked for visuals

from \_\_future\_\_ import division

from visual import \*

from visual.graph import \* # import graphing features

O=Oxygen=sphere(pos=vector(0,0,0),radius=10\*.66e-11, color=color.red)

H1=Hydrogen1=sphere(pos=vector(95.564e-12,7.2661e-12,0),radius=10\*.31e-11)

H2=Hydrogen2=sphere(pos=vector(-95.564e-12,7.2661e-12,0),radius=10\*.31e-11)

Na=Sodium=sphere(pos=vector(0,-10e-12,195e-12), radius=116e-13, color=color.yellow)

#With a Kick

#H1=Hydrogen1=sphere(pos=vector(90e-12,7e-12,0),radius=10\*.31e-11)

#H2=Hydrogen2=sphere(pos=vector(-90e-12,7e-12,0),radius=10\*.31e-11)

mO=15.99/(6.022e26)

mH=1.0079/(6.022e26)

mNa=22.9897/(6.022e26)

qO=.8

qH=.4

qNa=1

qCl=-1

#Given from experimentation

k=50

upp=5e-5

pO=mO\*vector(0, 0, 0)

pH1=mH\*vector(0, 0, 0)

pH2=mH\*vector(0, 0, 0)

pNa=mNa\*vector(0, 0, 0)

#graph1 = gcurve(color=color.white)

graph2 = gcurve(color=color.blue)

#Intramolecular Force Modeled by springs only and intermolecular force modeled by Coulomb only

ke=8.987551e+9

t=0

dt=15e-20

while t<5e-12:

t=t+dt

Fl1=-k\*(mag(H1.pos-O.pos)-95.84e-12)\*norm(H1.pos-O.pos)

Fl2=-k\*(mag(H2.pos-O.pos)-95.84e-12)\*norm(H2.pos-O.pos)

Fa=upp\*(mag(H1.pos-H2.pos)-191.128e-12)\*norm(H1.pos-H2.pos)

FONa=((ke\*qO\*qNa)/(mag(Na.pos-O.pos))\*\*2)\*-norm(Na.pos-O.pos)

FNaO=-((ke\*qO\*qNa)/(mag(Na.pos-O.pos))\*\*2)\*-norm(Na.pos-O.pos)

FH1Na=((ke\*qNa\*qH)/(mag(Na.pos-H1.pos))\*\*2)\*-norm(Na.pos-H1.pos)

FH2Na=((ke\*qNa\*qH)/(mag(Na.pos-H2.pos))\*\*2)\*-norm(Na.pos-H2.pos)

H1.pos=H1.pos+((pH1)/mH)\*dt

H2.pos=H2.pos+((pH2)/mH)\*dt

O.pos=O.pos+((pO)/mO)\*dt

Na.pos=Na.pos+((pNa)/mNa)\*dt

pH1=pH1+(Fl1-Fa-FH1Na)\*dt

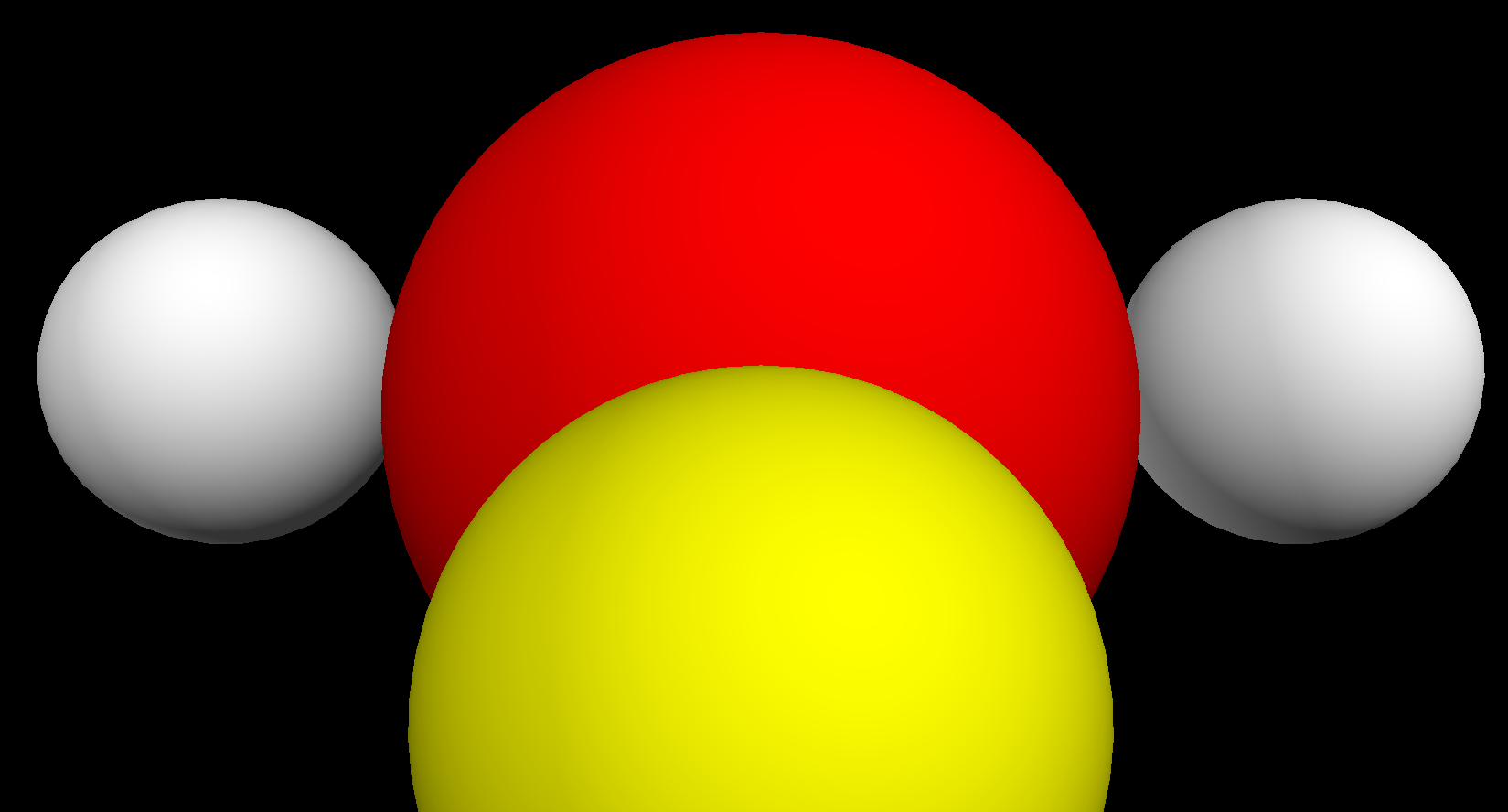
pH2=pH2+(Fl2+Fa-FH2Na)\*dt

pO=pO+(FH1O+FH2O)\*dt

pNa=pNa+(FONa+FH1Na+FH2Na)\*dt

#graph1.plot(pos=(t,mag(H1.pos-O.pos)-95.84e-12))

graph2.plot(pos=(t,mag(Na.pos-O.pos)-95.84e-12))

****