A classical model for the $H_2{}^+$ ion

Connah Johnson and Sam Brind

89398520 and 8926382

School of Physics and Astronomy
The University of Manchester

Theory computing project report

April 2015

Abstract

The motion of a classical electron under the influence of the Coulomb force has been numerically modelled, leading to a numerical model of the $H_2{}^+$ ion. The origin of the Coulomb force has been varied alongside the initial system conditions. Systems investigated include; a free electron with a stationary proton, two stationary protons and two free protons. The last system being the most accurate classical $H_2{}^+$ ion model investigated. Using the model, the conditions for the stability of the ion have been explored. A discussion on the numerical methods used has also been performed.

1. Introduction

The $H_2{}^+$ ion is formed through the ionisation of a H_2 molecule. The ion consists of two protons and an electron interacting through the Coulomb force, which is attractive between the electron and a proton and repulsive between the two protons. A classical inspection assumes that the bodies are point charges orbiting each other with negligible gravitational interactions, quantum mechanical effects are also ignored. This $H_2{}^+$ ion system is shown in Figure 1.

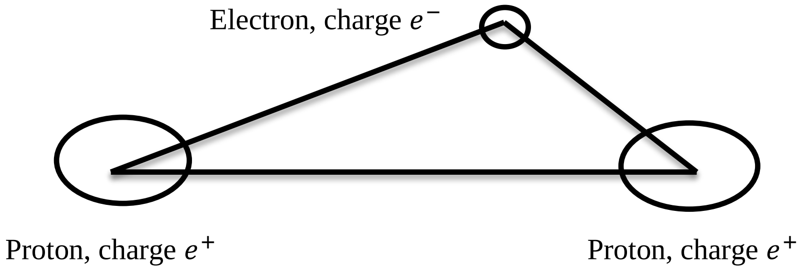

Figure 1. The $\boldsymbol{H_2}{}^+$ ion system in the x-y plane, showing the lines of action of the electromagnetic interactions.

The stability of the ion is dependent on many factors. For stability the protons must remain approximately motionless. This is due to the Coulomb force being weaker at large distances, so as a proton's separation in the system increases the force keeping the system bound decreases leading to the disintegration of the ion. The electron will then subsequently collide with one of the protons as the electron, according to a classical treatment. In nature this condition is answered through the large relative mass of the proton when compared to the electron, leading to the centre of mass of the system at the origin. It is also necessary that the ion is undisturbed by external collisions and that there is an absence of photons interacting with the system. [1]

The main difference between the classical and quantum mechanical approaches for the $H_2{}^+$ ion is the issue of the radiation of energy and the definition of the position of the particles. The issue of the radiation of energy occurs due to the acceleration experienced by the charge particles through a process called Bremsstrahlung. This process causes a loss in the kinetic energy of a charged particle when deflected by another charged particle leading to the emittance of a photon. The power loss due to Bremsstrahlung is given by Larmor's formula;

$$P = \frac{e^2 a^2}{6\pi\varepsilon_0 c^3}$$

where P is the power loss and a is the acceleration of the charged particle with the other quantities having their usual definitions [2]. As the kinetic energy of the system diminishes the classical orbiting electron will eventually collide with the proton, a phenomenon which lead to the development of Quantum Mechanics. The introduction of the wavefunction to the quantum mechanical description of particles renders the concept of tracking the position of the particles nonsensical while the symmetric and antisymmetric electron wavefunctions lead to different covalent bond stabilities.

2. Theory

The motions of the charged particles are affected by electromagnetic processes as was discussed in section 1. Two charged particles interact through the Coulomb force;

$$\underline{F_{12}} = \frac{Q_1 Q_2}{4\pi\varepsilon_0} \frac{\hat{r}_{21}}{\left|\underline{r_{21}}\right|^2} \qquad (1)$$

where $\underline{r_{21}} = \underline{r_2} - \underline{r_1}$ and $\hat{r}_{21}$ is the unit vector direction of $\underline{r_{21}}$. $\underline{F_{12}}$ is the force acting on particle 1 due to particle 2. The force acting on the particle is related to the acceleration experienced through Newton's 2nd law, $\underline{F} = m\underline{a}$

(2), where m is the mass of the particle and $\underline{a}$ is the acceleration of the particle. For a system of N charges Equation 1 may be written;

$$\underline{F} = \frac{q}{4\pi\varepsilon_0}\sum_{i=1}^{N} q_i \frac{\underline{r}-\underline{r}_i}{|\underline{r}-\underline{r}_i|^3} \qquad (3)$$

where $\underline{F}$ is the force acting on a particle at the position $\underline{r}$ and the summation is taking into account the contributions to the force from each other particle in the system. Equation 3 shows that the motion of each particle may be considered separately when the instantaneous positions of all other particles are known. [3]

2.1 Two stationary protons and an electron system

Consider a system where two protons are located a fixed distance of d from the origin. The displacement of the electron from the protons and the coordinate system is shown in Figure 2.

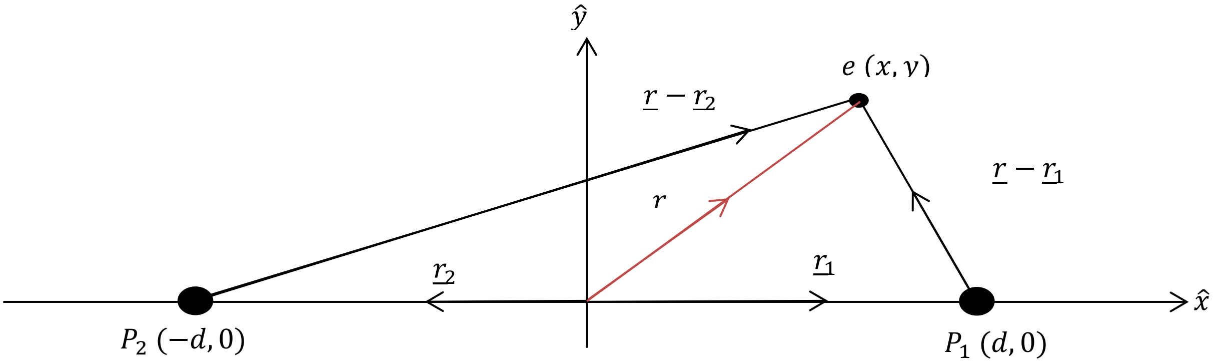

Figure 2. The stationary proton set up in the x-y plane, showing the positions of the particles and the displacement vector directions.

With the condition that the protons' positions remain fixed, the forces acting on the protons may be ignored. Using Equation 3, the combined force acting on the electron is given by;

$$\underline{F} = \frac{-e}{4\pi\varepsilon_0}\left[e\frac{(x-d)\cdot\hat{x}+y\cdot\hat{y}}{[(x-d)^2+y^2]^{\frac{3}{2}}} + e\frac{(x+d)\cdot\hat{x}+y\cdot\hat{y}}{[(x+d)^2+y^2]^{\frac{3}{2}}}\right] \qquad (4)$$

where the positions of the particles has been made explicit. Equation 4 is equal to Equation 2 and the acceleration may be decomposed into the constituent directions producing;

$$a_x = \frac{-e^2}{4\pi\varepsilon_0 m_e}\left[\frac{x-d}{[(x-d)^2+y^2]^{\frac{3}{2}}} + \frac{x+d}{[(x+d)^2+y^2]^{\frac{3}{2}}}\right] \quad a_y = \frac{-e^2}{4\pi\varepsilon_0 m_e}\left[\frac{y}{[(x-d)^2+y^2]^{\frac{3}{2}}} + \frac{y}{[(x+d)^2+y^2]^{\frac{3}{2}}}\right] \qquad (5)$$

which may be integrated using a numerical method, with knowledge of the initial position and velocity of the electron, to form an equation of motion for the electron.

2.2 Covalent binding

The forces between the electron and each proton must be greater than the repulsive force between the two positively charged protons for a stable ion. This defines a covalent bond and therefore the electron binds the protons together. For this condition to be met the time averaged position of the electron must be between the protons. If this time averaged position is far from the protons then the system is unstable. As such it would be expected that the electron spends a greater amount of time between the protons than elsewhere.

As shown in Figure 2, the proton-proton repulsion force acts along the $\hat{x}$ direction therefore the $\hat{y}$ component of the electron motion is independent of the covalent bonding. Let the force acting on the first proton by both the electron and the second proton be F_1 and likewise for the second proton force F_2, as shown in Figure 3. Where both F_1 and F_2 are confined to act along the positive $\hat{x}$ direction only.

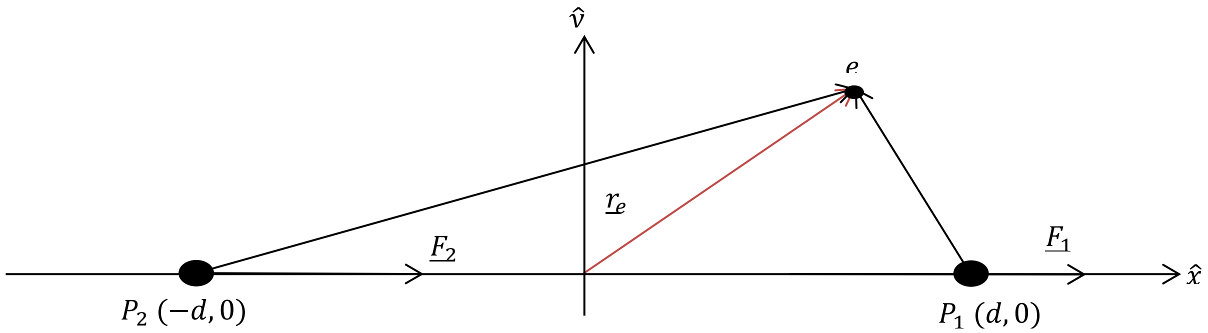

Figure 3. The covalent forces acting on each proton showing the direction of the resultant system force if the resultant forces acting on each proton are unequal.

Consider the magnitudes of these forces. If $F_1 > F_2$ the forces are not balanced with the net force acting on P_2 being in the $-\hat{x}$ direction. This is because the proton-proton repulsion is greater than the P_2-electron attraction. So the protons will separate, pulling the system apart. Therefore the system is stable if $F_1 \leq F_2$ with the equality being the point of stability. Forces F_1 and F_2 may be derived from Equation 4 while ignoring the y components;

$$F_1 = \frac{e^2}{4\pi\varepsilon_0}\left[\frac{1}{4d^2} + \frac{x-d}{[(d-x)^2+y^2]^{\frac{3}{2}}}\right] \qquad F_2 = \frac{e^2}{4\pi\varepsilon_0}\left[\frac{-1}{4d^2} + \frac{(x+d)}{[(x+d)^2+y^2]^{\frac{3}{2}}}\right] \tag{6}$$

where x is the displacement of the electron along the x-axis.

2.3 Many body systems

A many body system occurs when there are more than two bodies in an interacting system. The forces for these systems cannot be integrated to form exact solutions. Therefore either numerical models or assumption about the system are used to solve for the equations of motion. The orbiting motions of the planets in the solar system form a many body system, which is solvable in the approximation of the Sun and Jupiter having much larger masses than other bodies the system. The complexity of the system is due to the coupling of the force equations between the interacting bodies and as such the greater the number of bodies the greater the complexity. Statistical methods are used when the number of bodies becomes large because of the increasing computational cost of calculating the motion of each of the bodies. Many body systems also exhibit chaotic behaviour because a small perturbation on the conditions of one body may have an effect on all other particle in the system to some extent. [4]

When the assumption that the protons remain fixed may be relaxed a three-body system is formed, a many body system. Here the origin of the system has been taken to be at the point of the centre of mass. As the Coulomb force is an example of a central force field, the magnitudes of the forces act in the radial direction and only depend on the distances between the charges. The assumption that the system is confined to move in the x-y plane only is still enforced. [4]

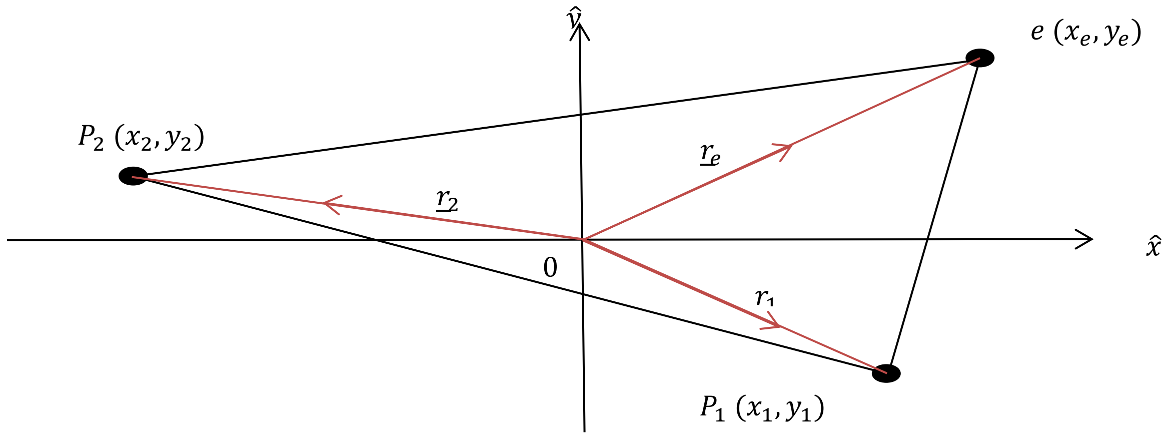

Figure 4. The many body set up for a system of two protons and an electron each interacting via the coulomb force.

Using Equations 3 and 2 in a similar method used in section 2.1 one may solve for the equations of motion. The method is repeated for all three particles which allows the production of the six equations of motion for the particles in the system, after decomposing the accelerations into the x and y components.

2.4 Orbits and energy considerations

For this model, the ${H_2}^+$ ion is considered to be isolated so the total energy of the system must be constant and equal to the summation of potential and kinetic energies of the bodies. Graphing the potential energy of the system becomes informative as shown in Figure 5.

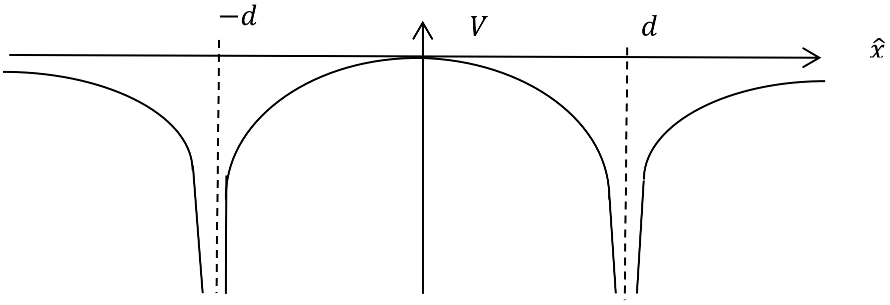

Figure 5. The one dimensional potential well for an electron located between two protons. The potential energy tends to infinity at the location of the protons whereas the potential energy is zero at the midpoint. The midpoint is a point of unstable equilibrium.

The origin of a rectilinear system of two protons and an electron shows that a small perturbation of the system may lead to the collision of the electron with a proton and the disintegration of the system. This effect demonstrates the importance of the accuracy of numerical method as an error in the calculated electron position will, in effect, perturb the system.

The total energy of the system also allows the classification of the orbits of the electron as bound or unbound. If the total energy of the system is negative then the orbits will be bound permitting the formation of a stable molecule. If the total energy of the system is positive then the orbits are unbound, as the potential energy of the system is negative due to the attractive Coulomb force this implies that the kinetic energy of the system is too great. This may occur due to a large initial position for the electron with a given velocity. The velocity dependence on the orbit classification allows orbits to be classified as circular, elliptical, parabolic and hyperbolic. Circular orbits occur in the single proton-electron system where the electron velocity is perpendicular to the Coulomb force whereas elliptical orbits are bound orbits and occur in a three body system. Parabolic orbits occur when the potential energy equals the kinetic energy which is at the point at which the system becomes unbound. Hyperbolic orbits are also unbound and occur when the kinetic energy is larger than the potential energy.

3. Numerical methods

To model the ${H_2}^+$ ion requires solving second order differential equations and as such requires the use of numerical methods. When considering the accuracy of a numerical method the step length, h, is focal to the discussion as shown in Figure 6. The smaller the value of the step length the greater the accuracy of the method at the expense of computation costs. The step length may also have a lower limit due to the increase in computational rounding errors caused by a finite variable memory length. The model was required to be able to accurately calculate multiple orbits of the electron around the protons. The required accuracy was increased when the protons were given velocities as the system was unstable in nature and high velocities produced greater error in methods using large h values. As such, methods that had an effective step length of multiple orders of h produced higher accuracies per computation cost but were increasingly more complex to implement. [5]

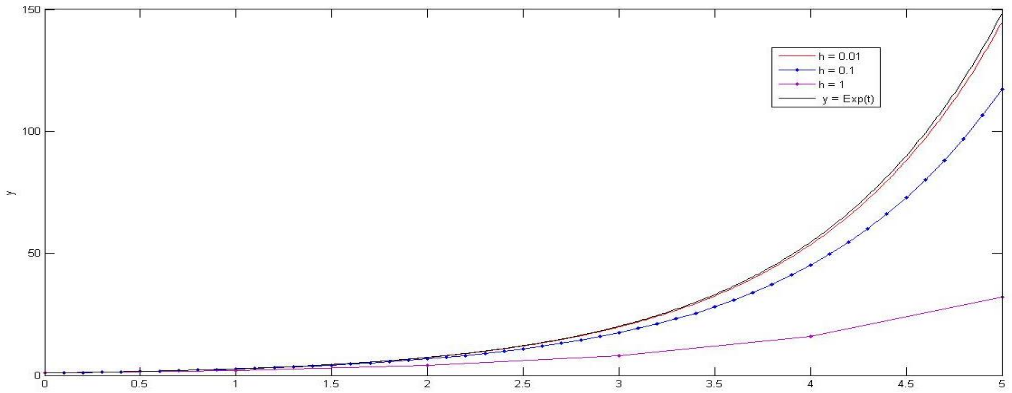

Figure 6. The step length of the Euler method is varied to show the loss in accuracy when compared to the analytic solution to the equation $\boldsymbol{y = e^{t}}$

The trialled methods included the Euler method, second order Runge-Kutta and the fourth Order Runge-Kutta (RK4) each having an effective step length of order 1, 2 and 4 respectively. These three methods were trialled against the known solution for simple harmonic motion, $\boldsymbol{\frac{d^2y}{dx^2} = -y}$, as shown in Figure 7.

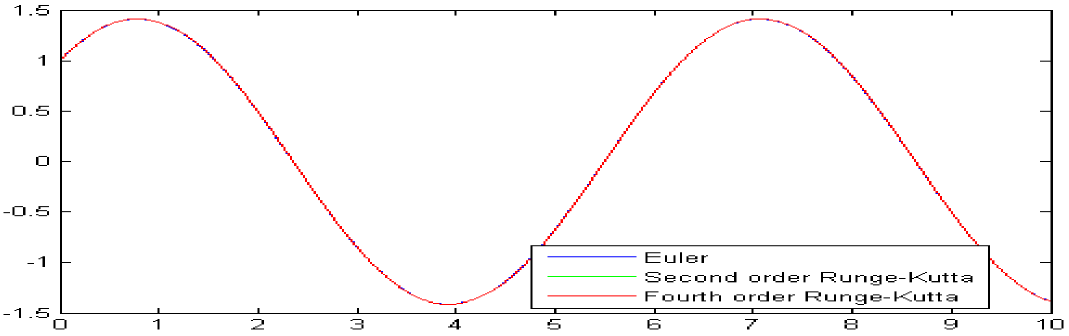

Figure 7. A graph showing simple harmonic motion of a particle with an initial displacement of 1 unit and the estimated motion as calculated using three different numerical methods. The plot shows that for a simple system with a step length $\boldsymbol{h = 0.005}$ all methods provide similar results.

Figure 7 shows that for a simple system with a step length $h = 0.005$ all the trialled methods provided similar results. However the errors were great at the extremes of the graphs, as shown in Figure 8, where the response of the equation varied considerably over a short step length. This large error was found to occur with the high acceleration experienced by the electron when in close proximity to a proton.

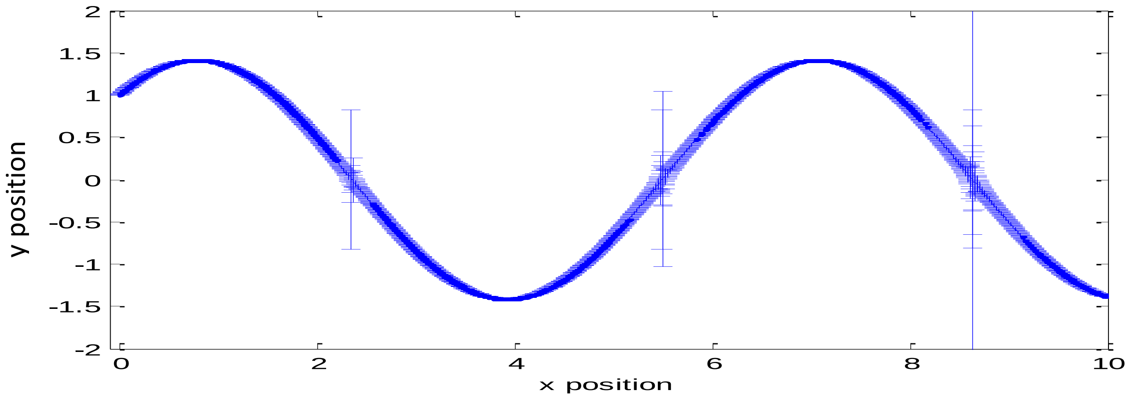

Figure 8. The second order Runge-Kutta method used to solve the simple harmonic motion equation. The error bars show the percentage error of the solution when compared to the analytic solution $\boldsymbol{y = \cos(x)}$.

Examining the total energy of the system may be used as a method of validating the accuracy of a numerical method to model a given situation. The RK4 has high accuracy but, due to the investigated time variation as shown in Figure 9, would be better suited to model where the energy decays, such as for damped oscillations.[6]

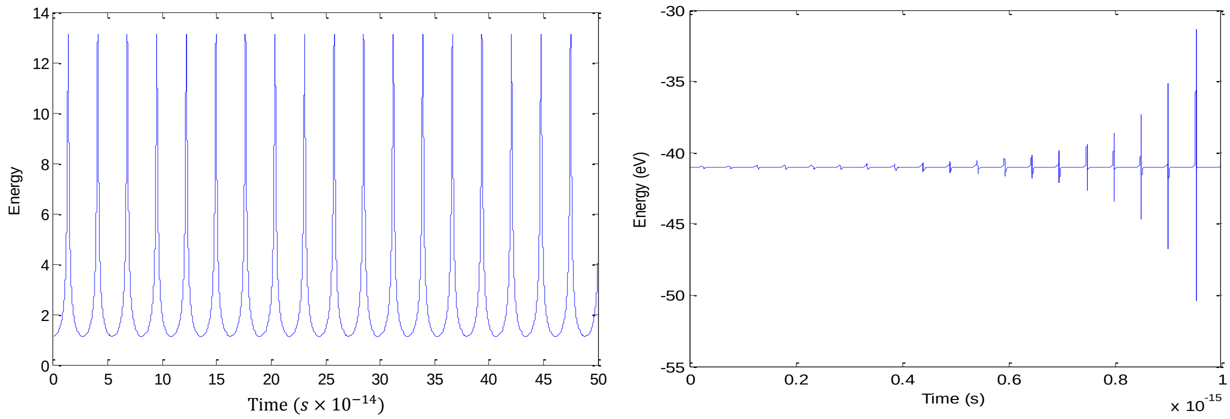

Figure 9. The shapes of the total energy variations: on the left the variation for a single proton-electron system using the RK4 showing a drift in the energy, on the right a two stationary proton and an electron system using the leapfrog method which may be averaged to a constant value. Energy is expected to be conserved in both cases.

The RK4 provided a highly efficient and accurate model with the lowest percentage error for a given step length. However when these numerical methods were applied to the ${H_2}^+$ ion model they were shown to violate the conservation of energy principle producing incorrect trajectories for simple models such as an electron orbiting a single proton. The stability of the numerical method directly affects whether the energy of the system is conserved. The previously trialled methods were not reversible in time due asymmetric time dependence and, as such, allowed the total system energy to drift, Figure 9.

The leapfrog method is second order with respect to the step length, like the second order Runge-Kutta method, as well as permitting the time averaged energy to be conserved through a periodic oscillation of the system energy. Therefore the leapfrog method was introduced to improve the model. [6]

This method involves calculating the half-step velocities;

$$v_{i+\frac{1}{2}} = v_i + \frac{h}{2}a_i$$

where v_i and a_i are the previous iterations velocity and acceleration calculated through Equations 5 for two stationary protons etc. The iterative position values are then calculated through $x_{i+1} = x_i + hv_{i+\frac{1}{2}}$, time is then incremented. [1]

4. Results

The results for the different systems allowed the ${H_2}^+$ ion model to be progressively built upon. The results for the one proton system allow the stability of the code to be examined. The two stationary proton model demonstrated the effect of the electron energy on the electron's orbit, the covalent binding and different types of orbit. The free proton model demonstrated the three body system and the importance of the accuracy of the numerical method through small perturbations from a point of instability. The trajectories of the electrons were plotted from Equations 5 when integrated twice using the numerical methods to produce position values with the initial proton positions being Bohr radii in magnitude.

4.1 Single proton system

The single proton system would produce elliptical or circular orbits. For these orbits to be maintained the method needed to be stable, this was shown in Figure 10. Here the circular orbit was maintained for 317 full rotations without any significant deviation from a circular path.

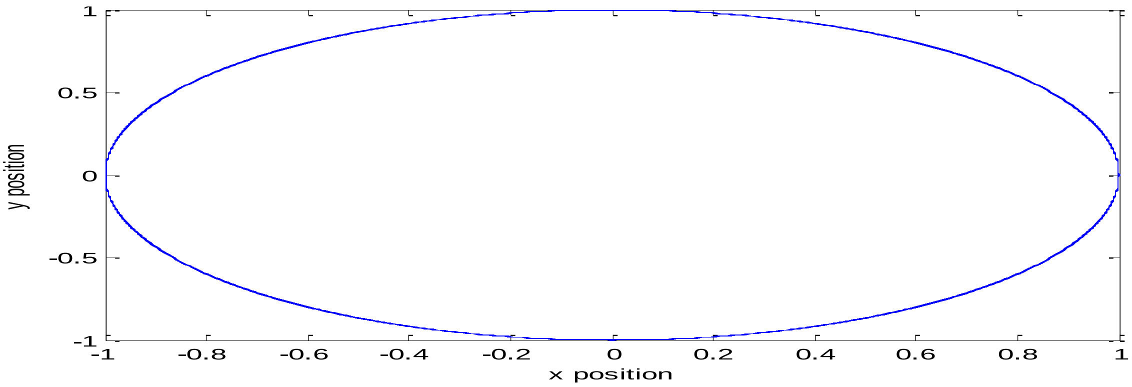

Figure 10. The orbit of an electron with an initial velocity of 1 unit in the y diection and an initial position of 1 unit in the exdirection. The orbited proton is located at the origin of the system.

4.2 Two stationary protons and an electron system

As discussed in section 2.4 the classification of the orbits on the electron around the protons is determined by the total energy of the system. Figure 11 shows the trajectory of the electron motion changing with the increasing system energy, with orbits changing from elliptical to hyperbolic, bound to unbound.

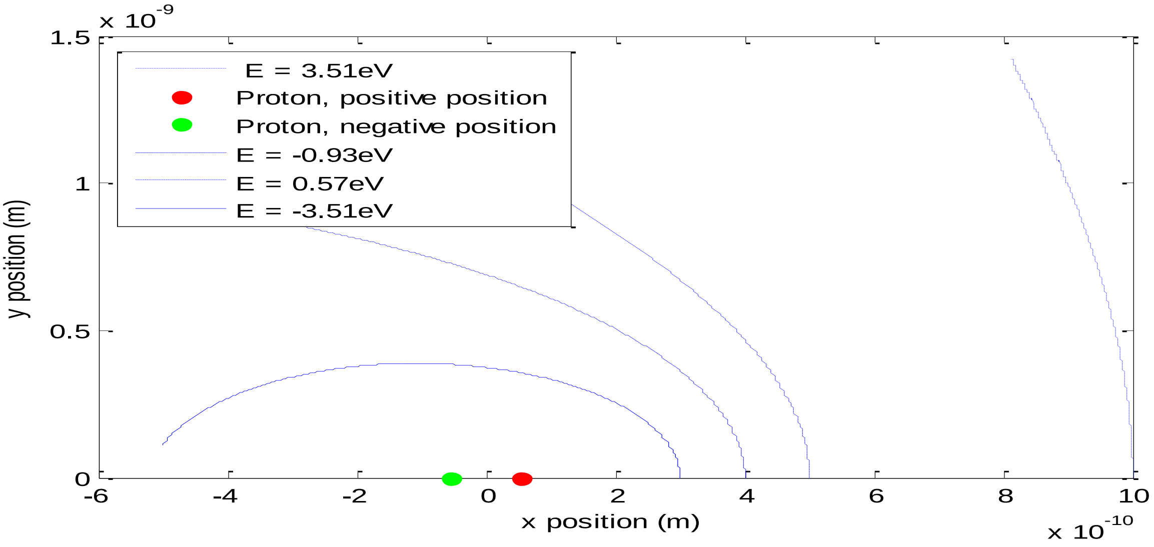

Figure 11. The trajectories of the electron for different positions along the x axis and therefore different potential energies. The initial velocity of each electron is $\mathbf{1.5 \times 10^{6}}\ \boldsymbol{ms^{-1}}$ in the $\boldsymbol{y}$ direction.

For the two stationary proton system a graph of the trajectory and the individual energies that the electron possesses for given initial conditions is shown in Figure 12. The graph of the potential energy is similair to the graph shown in Figure 5.

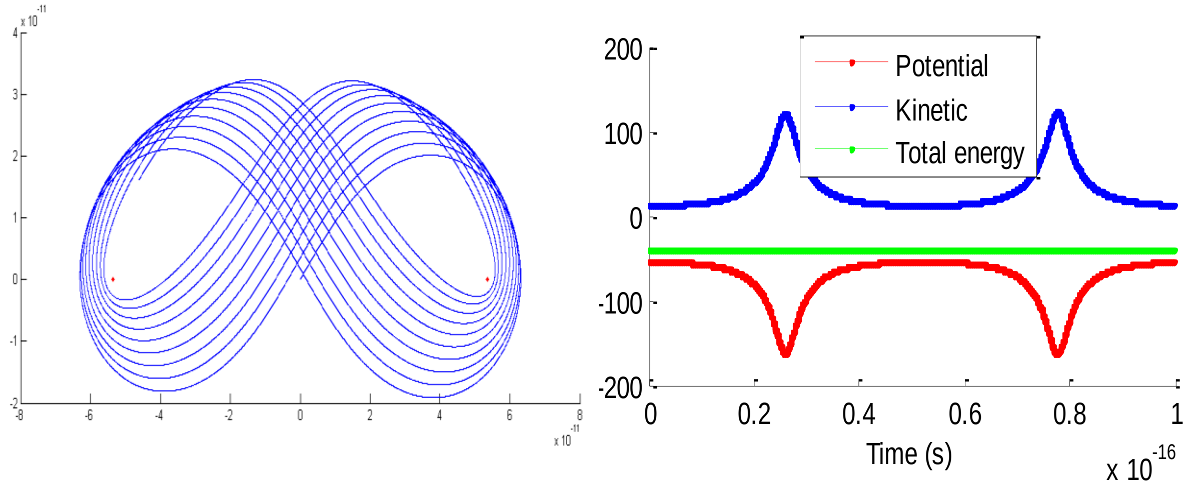

Figure 12. The orbit on the left shows an example of a path of the electron with initial velocity in the x and y directions with magnitude $\mathbf{1.5 \times 10^{6}}\ \boldsymbol{ms^{-1}}$ with the protons being located at the Bohr radius from the origin. The energy graph on the right shows the variation in the energies.

Covalent bonding was modelled through Equations 6 and by calculating the forces at each point in the path of the electron. The forces were then equated and the region for which the stability inequality was satisfied can be seen in Figure 13. The lines seen in the bounding region of Figure 13 that are not located on the elctron trajectory are numerical errors that in accurately satisfy the binding condition but do not detract from the clear boundary to the binding region.

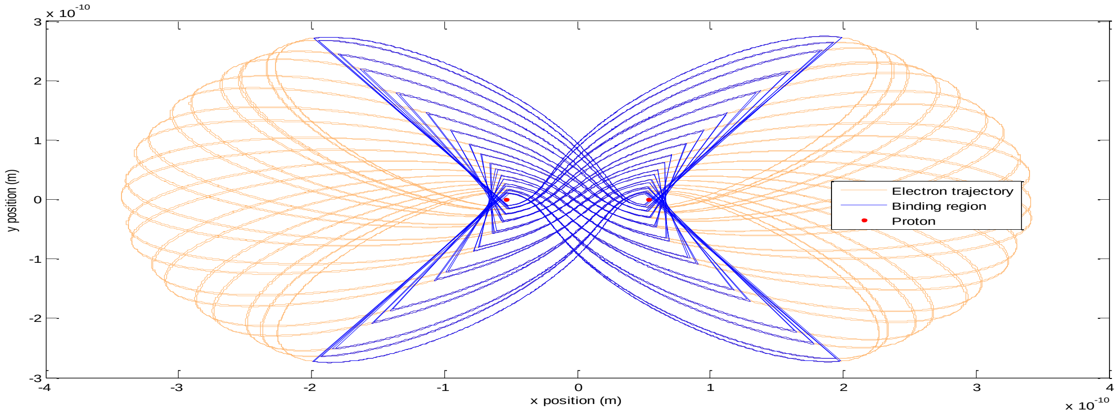

Figure 13. The region for which covalent bonding takes place has been highlighted for an electron with initial velocity of $\mathbf{-2.8 \times 10^{5}}$ in both the x and y directions and initial position of $\mathbf{-2.4 \times 10^{-10}}$ and $\mathbf{2.4 \times 10^{-10}}$ in the x and y directions respectively.

4.3 The three body problem

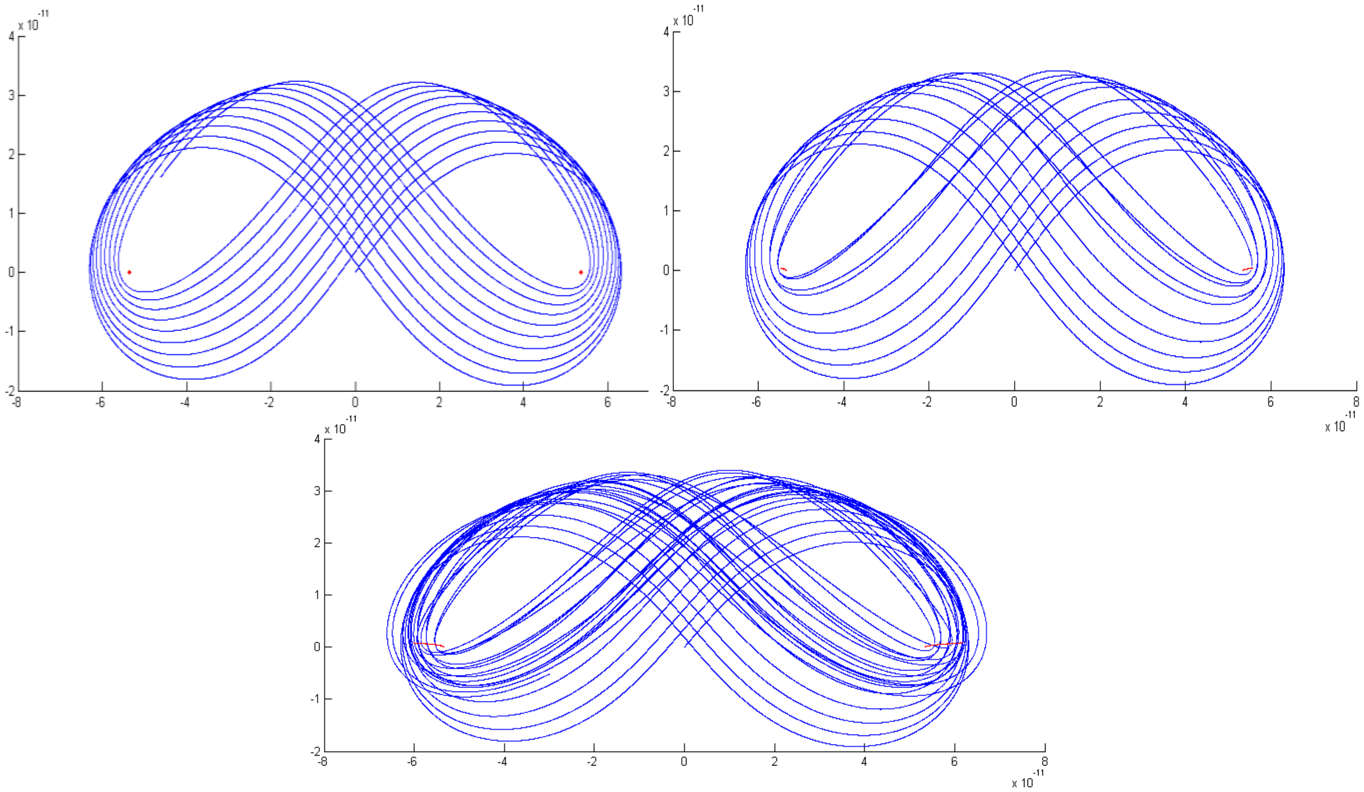

Figure 14. An example demonstrating the effect of allowing the protons to move. Top left is an example of the previous two stationary protons system while top right is the same initial conditions but allowing the protons to move. The lower plot is for double the time and clearly shows the instability of this system due to the protons moving apart. [1]

Through progressive iterations of the model, the code was set up to solve the three body problem involving two moving protons. Figure 14 shows an example of how introducing two extra bodies changes the calculated motion. The amount the protons move from their initial condition provides an indication of the system's stability.

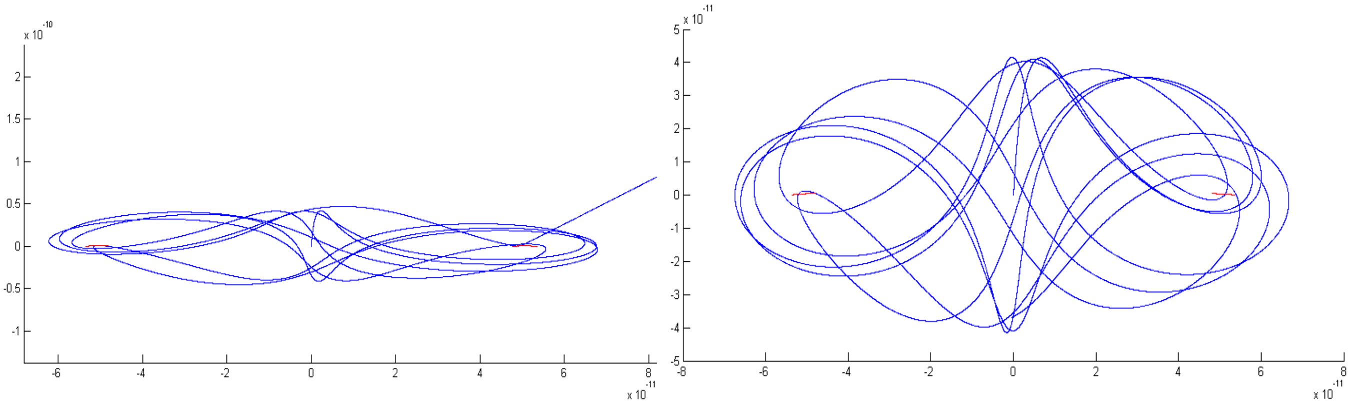

Figure 15. Two plots of perturbations from the simple case of an electron moving in the $\hat{\boldsymbol{y}}$ direction. The left plot is off axis by 1 degree and the right by 3 degrees to the $\hat{\boldsymbol{y}}$ direction.

Small perturbations from a simple orbit were also investigated. This lead to chaotic orbits, occasionally colliding with the proton as can be seen with the first graph of Figure 15. The numerical method's accuracy breaks down at areas near to the proton as the magnitude of the electron's acceleration is large, as was discussed in section 3.

5. Discussion and conclusion

The final model gave results for a variety of electron orbits however there are further additions that could be made to bring it closer to reality. As previously mentioned in section 1, an accelerating charge will experience a loss of energy described by Larmor's formula. This would produce orbits that spiral inwards and affect the electron trajectories.

The accuracy of the model may be improved by replacing the leapfrog method with a higher order method which retains the energy conserving property. A fourth order method such as the Forest-Ruth integrator would combine the accuracy of the RK4 method with the required energy property. This method uses the Hamiltonian of the system so the Newtonian dynamics would be replaced with the Lagrange formalisation. A side effect of this may be a simplification in the addition of more charged particles. [7]

The current model may be altered to accommodate more charged particles through the addition of more force terms. This would allow more complex many body systems to be investigated. A more radical change could be introduced by taking into account the quantum mechanical effects such as: the wave function describing motion instead of our equations of motion, uncertainty on the electron's position, discrete levels of energy and the effect of spin and angular momentum.

In conclusion the three body model presented provides an accurate model to the ${H_2}^+$ ion. Covalent binding alongside the response of the ion when the electron energies and velocity has been changed has been demonstrated. While small perturbations from a stable path for the three body model has been investigated. The use of classical mechanics has been contrasted against quantum mechanics while the numerical method used has been varied and tested against simple cases. The weakness of some other numerical methods has been discussed and improvements to the model have been suggested.

References

[1] De Jong, M. 1991. Introduction to computational Physics, 1st ed, Addison-Wesley, Pages 177-188

[2] Jackson, J. 1999. Classical electrodynamics, 3rd ed, Wiley

[3] Grant, I. and Philips, W. 1990. Electromagnetism, 2nd ed, Wiley

[4] Goldstein, H. 1980. Classical mechanics, 2nd ed, Addison-Wesley, ch 3

[5] Riley, K. 1978. Mathematical methods for the physical sciences, Cambridge.

[6] McMillan, S. The leapfrog integrator, http://einstein.drexel.edu/courses/Comp_Phys/Integrators/leapfrog/ Accessed on 25/04/2015

[7] Yoshida, H. 1990. Construction of higher order symplectic integrators, Physics letters A, Vol. 150, Numbers 5,6,7. http://sixtrack.web.cern.ch/SixTrack/doc/yoshida00.pdf Accessed on 25/04/2015

Appendix

Attached is the code used for the three body problem. The code when run outputs the positions of all particles for given time steps into the file, “Leapfrog.csv”.

```
//Leapfrog.cpp, Model of an electron classically orbiting around two protons, free to move, using the
leapfrog numerical method, solving the two dimensional coulomb law and writing the results to
Leapfrog.csv.

//by Connah Johnson and Sam Brind.
//19/04/2015

#include <iostream>
#include <cmath>
#include <fstream>
#include <iomanip>
using namespace std;

//define constants
double k = 8.987e9;      //constant Coulomb factor
double h = 1e-21;        //step length
double n = 1000000;      //number of iterations
double d = 5.35e-11;     //the distance of the protons from the origin along the x axis
double me = 9.109e-31;   //mass of an electron
double mp = 1.673e-27;   //mass of a proton
double q = 1.602e-19;    //electron charge
//set the initial conditions
double ex_0 = 0;
double ey_0 = 0;
double evx_0 = 1.36e6;
double evy_0 = 1.62e6;
double p1x_0 = d;
double p1y_0 = 0;
double p1vx_0 = 0;
double p1vy_0 = 0;
double p2x_0 = -d;
double p2y_0 = 0;
double p2vx_0 = 0;
double p2vy_0 = 0;
double t_0 = 0;
//the second order equations of motion are split into two first order equations for each dimension
//declare electron acceleration in the x direction
double e_ax(double ex, double ey, double evx, double evy, double p1x, double p1y, double p1vx, double
p1vy,double p2x, double p2y, double p2vx, double p2vy,double t){
    return (-k*q*q/me)*(ex-p1x)*pow(pow(ex-p1x,2)+pow(ey-p1y,2),-1.5)+ (-k*q*q/me)*(ex-
p2x)*pow(pow(ex-p2x,2)+pow(ey-p2y,2),-1.5);
}
//declare electron acceleration in the y direction
double e_ay(double ex, double ey, double evx, double evy, double p1x, double p1y, double p1vx, double
p1vy,double p2x, double p2y, double p2vx, double p2vy,double t){
    return (-k*q*q/me)*(ey-p1y)*pow(pow(ex-p1x,2)+pow(ey-p1y,2),-1.5)+ (-k*q*q/me)*(ey-
p2y)*pow(pow(ex-p2x,2)+pow(ey-p2y,2),-1.5);
}
//declare the 1st proton's acceleration in the x direction
double p1_ax(double ex, double ey, double evx, double evy, double p1x, double p1y, double p1vx, double
p1vy,double p2x, double p2y, double p2vx, double p2vy,double t){
    return (-k*q*q/mp)*(p1x-ex)*pow(pow(p1x-ex,2)+pow(p1y-ey,2),-1.5) + (k*q*q/mp)*(p1x-
p2x)*pow(pow(p1x-p2x,2)+pow(p1y-p2y,2),-1.5);
}
//declare the 1st proton's acceleration in the y direction
double p1_ay(double ex, double ey, double evx, double evy, double p1x, double p1y, double p1vx, double
p1vy,double p2x, double p2y, double p2vx, double p2vy,double t){
    return (-k*q*q/mp)*(p1y-ey)*pow(pow(p1x-ex,2)+pow(p1y-ey,2),-1.5) + (k*q*q/mp)*(p1y-
p2y)*pow(pow(p1x-p2x,2)+pow(p1y-p2y,2),-1.5);
}
//declare the 2nd proton's acceleration in the x direction
double p2_ax(double ex, double ey, double evx, double evy, double p1x, double p1y, double p1vx, double
p1vy,double p2x, double p2y, double p2vx, double p2vy,double t){
    return (-k*q*q/mp)*(p2x-ex)*pow(pow(p2x-ex,2)+pow(p2y-ey,2),-1.5) + (k*q*q/mp)*(p2x-
p1x)*pow(pow(p2x-p1x,2)+pow(p2y-p1y,2),-1.5);
}
//declare the 2nd proton's acceleration in the y direction
```

```
double p2_ay(double ex, double ey, double evx, double evy, double p1x, double p1y, double p1vx, double
p1vy,double p2x, double p2y, double p2vx, double p2vy,double t){
        return (-k*q*q/mp)*(p2y-ey)*pow(pow(p2x-ex,2)+pow(p2y-ey,2),-1.5) + (k*q*q/mp)*(p2y-
p1y)*pow(pow(p2x-p1x,2)+pow(p2y-p1y,2),-1.5);
}
double energy(double x, double y, double vx, double vy, double t){
    //energy is the sum of the kinetic energy and the potential energy of the electron
    return (0.5*me*((pow(vx,2)+pow(vy,2))) + (-k*q*q)*(pow((pow(x-d,2)+pow(y,2)),-0.5) +
pow(pow(x+d,2)+pow(y,2),-0.5)))/1.602e-19;
}
//function to measure the distance protons have moved
double proton_travel(double p1x, double p1y, double p2x, double p2y){
        //return sum of distances between proton position and initial condition
        return (pow(pow(p1x-p1x_0,2)+pow(p1y-p1y_0,2),0.5)) + (pow(pow(p2x-p2x_0,2)+pow(p2y-
p2y_0,2),0.5));
}

//main program function
int main(){
    //declare the starting points
    double ex=ex_0, ey=ey_0, evx=evx_0, evy=evy_0, t=t_0;
        double p1x=p1x_0, p1y=p1y_0, p1vx=p1vx_0, p1vy=p1vy_0;
        double p2x=p2x_0, p2y=p2y_0, p2vx=p2vx_0, p2vy=p2vy_0;
    //initialise the output file
    ofstream wfile("Leapfrog.csv");
    //file opening error check
    if (wfile.is_open()){       //file opening is successful
        //output the initial conditions
        cout<<"t0 = "<<t_0<<" x0 = "<<ex_0<<" y0 = "<<ey_0<<"\n";
        //write these initial conditions to the Leapfrog.csv file
        wfile<<t_0<<","<<ex_0<<","<<ey_0<<","<<"\n";
        //calculate/write the subsequent values to the file Leapfrog.csv
        //until the number of iterations has been exhausted
        //calculate the half step velocities
        evx = evx + (h/2)*e_ax(ex,ey,evx,evy,p1x,p1y,p1vx,p1vy,p2x,p2y,p2vx,p2vy,t);
        evy = evy + (h/2)*e_ay(ex,ey,evx,evy,p1x,p1y,p1vx,p1vy,p2x,p2y,p2vx,p2vy,t);
        p1vx = p1vx + (h/2)*p1_ax(ex,ey,evx,evy,p1x,p1y,p1vx,p1vy,p2x,p2y,p2vx,p2vy,t);
       p1vy = p1vy + (h/2)*p1_ay(ex,ey,evx,evy,p1x,p1y,p1vx,p1vy,p2x,p2y,p2vx,p2vy,t);
        p2vx = p2vx + (h/2)*p2_ax(ex,ey,evx,evy,p1x,p1y,p1vx,p1vy,p2x,p2y,p2vx,p2vy,t);
       p2vy = p2vy + (h/2)*p2_ay(ex,ey,evx,evy,p1x,p1y,p1vx,p1vy,p2x,p2y,p2vx,p2vy,t);
        //loop through n iterations
                for(int i=0; i<n; i++){
            //move to the new point
            ex = ex + h*evx;
             ey = ey + h*evy;
            evx = evx + h*e_ax(ex,ey,evx,evy,p1x,p1y,p1vx,p1vy,p2x,p2y,p2vx,p2vy,t);
             evy = evy + h*e_ay(ex,ey,evx,evy,p1x,p1y,p1vx,p1vy,p2x,p2y,p2vx,p2vy,t);
             p1x = p1x + h*p1vx;
             p1y = p1y + h*p1vy;
            p1vx = p1vx + h*p1_ax(ex,ey,evx,evy,p1x,p1y,p1vx,p1vy,p2x,p2y,p2vx,p2vy,t);
            p1vy = p1vy + h*p1_ay(ex,ey,evx,evy,p1x,p1y,p1vx,p1vy,p2x,p2y,p2vx,p2vy,t);
             p2x = p2x + h*p2vx;
             p2y = p2y + h*p2vy;
            p2vx = p2vx + h*p2_ax(ex,ey,evx,evy,p1x,p1y,p1vx,p1vy,p2x,p2y,p2vx,p2vy,t);
            p2vy = p2vy + h*p2_ay(ex,ey,evx,evy,p1x,p1y,p1vx,p1vy,p2x,p2y,p2vx,p2vy,t);
             t = t + h;
            //output the new position and distance travelled by the protons
            cout<<"t "<<t<<" x "<<ex<<" y "<<ey<<", P travel: " << proton_travel(p1x,p1y,p2x,p2y) <<
"\n";
            //write to file Leapfrog.csv
            wfile<<t<<","<<ex<<","<<ey<<","<<p1x<<","<<p1y<<","<<p2x<<","<<p2y<<"\n";
        }
        //close the file Leapfrog.csv
        wfile.close();
        cout<<"The results have been written to 'Leapfrog.csv' successfully.\n";
    }else{
        //output error message
        cout<<"Error, the file 'Leapfrog.csv' could not be opened.\n";
    }
        //arbitary return value
        return 0;
};
```