Grand Valley State University

Padnos College of Engineering and Computing

Golf Ball Projectile

EGR 312 Dynamics

Chad Paffhausen

December 8, 2006

Fall 2006

1. **Derive the differential equations of motion.**

A free body diagram of a golf ball is shown in Figure 1.

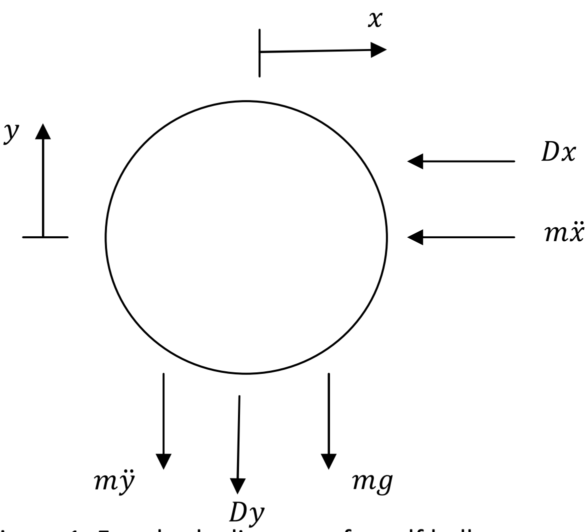

Figure 1: Free body diagram of a golf ball

The sum of the forces in the x direction is shown below.

$$\sum F_x = m\ddot{x} = -Dx$$

$$m\ddot{x} = -\frac{1}{2}C_D\rho AV^2$$

$$\ddot{x} = \frac{-C_D\rho AV^2}{2m}$$

The equation of for tau is listed below and is solved for mass so it can be inserted into the equation above.

$$\tau = \frac{m}{3\pi d\mu}$$

$$m = 3\tau\pi d\mu$$

$$\ddot{x} = \frac{-C_d\rho AV^2}{6\tau\pi d\mu}$$

The equation for area is listed below and is solved for diameter so it cab be inserted into the equation above.

$$A = \frac{\pi d^2}{4}$$

$$\ddot{x} = \frac{-C_d \rho \frac{\pi d^2}{4} V^2}{6\tau\pi d\mu}$$

$$\ddot{x} = \frac{-C_d \rho d V^2}{24\tau\mu}$$

The equation for Reynolds number is listed below and is solved for rho over mu so it can be inserted into the equation above.

$$R_e = \frac{\rho V d}{\mu}$$

$$\frac{\rho}{\mu} = \frac{R_e}{Vd}$$

$$\ddot{x} = \frac{-C_d V^2 R_e}{24\tau V}$$

$$\ddot{x} = -\frac{C_d R_e}{24\tau} V$$

The equation below is used to get velocity in terms of x.

$$V = \dot{x}$$

$$\ddot{x} = -\frac{C_d R_e}{24\tau} \dot{x}$$

The equation below is used to get position in terms of velocity.

$$\dot{x} = u$$

$$\dot{u} = -\frac{C_d R_e}{24\tau} u$$

The two equations listed below are the state equations for the horizontal direction.

$$\boxed{\frac{dx}{dt} = u}$$

$$\boxed{\frac{du}{dt} = -\frac{C_d R_e}{24\tau} u}$$

The sum of the forces in the y direction are shown below.

$$\sum F_y = m\ddot{y} = -Dy - mg$$

$$\ddot{y} = \frac{-D_y}{m} - g$$

$$\ddot{y} = \frac{-C_D \rho A V^2}{2m} - g$$

The equation of for tau is listed below and is solved for mass so it can be inserted into the equation above.

$$\tau = \frac{m}{3\pi d\mu}$$

$$m = 3\tau\pi d\mu$$

$$\ddot{y} = \frac{-C_d \rho A V^2}{6\tau\pi d\mu} - g$$

The equation for area is listed below and is solved for diameter so it cab be inserted into the equation above.

$$A = \frac{\pi d^2}{4}$$

$$\ddot{y} = \frac{-C_d \rho \frac{\pi d^2}{4} V^2}{6\tau\pi d\mu} - g$$

$$\ddot{y} = \frac{-C_d \rho d V^2}{24\tau\mu} - g$$

The equation for Reynolds number is listed below and is solved for rho over mu so it can be inserted into the equation above.

$$R_e = \frac{\rho V d}{\mu}$$

$$\frac{\rho}{\mu} = \frac{R_e}{Vd}$$

$$\ddot{y} = \frac{-C_d V^2 R_e}{24\tau V} - g$$

$$\ddot{y} = -\frac{C_d R_e}{24\tau} V - g$$

The equation below is used to get velocity in terms of x.

$$V = \dot{y}$$

$$\ddot{y} = -\frac{C_d R_e}{24\tau}\dot{y} - g$$

The equation below is used to get position in terms of velocity.

$$\dot{y} = v$$

$$\dot{v} = -\frac{C_d R_e}{24\tau}v - g$$

The two equations listed below are the state equations for the vertical direction.

$$\boxed{\frac{dy}{dt} = v}$$

$$\boxed{\frac{dv}{dt} = -\frac{C_d R_e}{24\tau}v - g}$$

2. Assume no friction, solve x, y, u, and v symbolically, and plot the path of the golf ball.

Solving symbolically for velocity (u).

$$\frac{du}{dt} = 0$$

$$\int du = \int 0 dt$$

$$\boxed{u = u_0 = constant}$$

Solving symbolically for position (x).

$$\frac{dx}{dt} = u$$

$$\int dx = \int u dt$$

$$\boxed{x = x_o + ut}$$

Solving symbolically for velocity (v).

$$\frac{dv}{dt} = -g$$

$$\int dv = \int -g dt$$

$$\boxed{v = v_0 - gt}$$

Solving symbolically for position (y).

$$\frac{dy}{dt} = v$$

From above $v = v_0 - gt$.

$$\frac{dy}{dt} = v - gt$$

$$\int dy = \int (v - gt)dt$$

$$\boxed{y = y_o + v_0 t - \frac{1}{2}gt^2}$$

Now the horizontal and vertical position needs to be plotted. The initial position for both the x and y direction is zero. The initial velocity in the vertical direction is $120\frac{ft}{sec}sin30° = 60\frac{ft}{sec}$. The initial velocity in the horizontal direction is $120\frac{ft}{sec}cos30° = 103.9\frac{ft}{sec}$. The velocity in the horizontal direction is constant. The equation for position in the horizontal direction is $x = 103.9t$. The equation for the position in the vertical direction is $y = 60t - \frac{1}{2}32.2t^2$. Figure 2 is a plot of projectile motion with no frictional force.

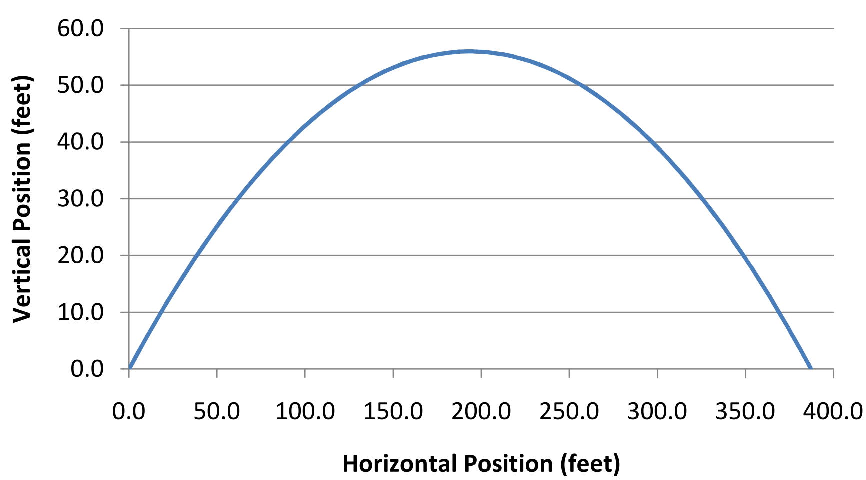

Figure 2: Projectile Motion without Air Friction

3. Assume the product C_D R_e in equation 1 is constant $R_e = 9 \times 10^4$ and $C_D = 0.25$, solve the system of differential equations for a closed form solution, and plot the path of the golf ball.

$$\ddot{x} = -\frac{C_d R_e}{24\tau}\dot{x}$$

C_d and R_e is given and τ needs to be calculated.

$$\tau = \frac{m}{3\pi d\mu}$$

$$\tau = \frac{1.5oz\left(\frac{1lb}{16oz}\right)\left(\frac{1}{32.2\frac{ft}{s^2}}\right)}{3\pi(1.75in)\left(\frac{1ft}{12in}\right)0.375\times10^{-6}\frac{(lb-s)}{ft}}$$

$$\tau = 5648.8$$

$$\ddot{x} = -\frac{0.25\times9\times10^4}{24(5648.8)}\dot{x}$$

$$\ddot{x} = -0.1660\dot{x}$$

The homogeneous equation is solved below.

$$\ddot{x} + 0.1660\dot{x} = 0$$

$$R^2 + 0.1660R = 0$$

$$R(R + 0.1660) = 0$$

$$R = 0, -0.1660$$

$$x = c_1e^{0t} + c_2e^{-0.1660t}$$

$$x = c_1 + c_2e^{-0.1660t}$$

Since there is no particular solution the derivative is taken to get the velocity equation. Using the initial conditions for position and velocity the constant can be found.

$$\dot{x} = -0.1660c_2e^{-0.1660t}$$

$$x(0) = c_1 + c_2 = 0$$

$$\dot{x}(0) = -0.1660c_2 = v(0) = V_0 = 120cos30° = 103.9$$

$$c_2 = -626.04$$

$$c_1 = -c_2 = 626.04$$

The equation below is the equation for horizontal position as a function of time.

$$\boxed{x = 626.04 - 626.04e^{-0.1660t}}$$

The vertical direction can be solved the same way as the horizontal direction was.

$$\ddot{y} = -\frac{C_d R_e}{24\tau}\dot{y} - g$$

Tau is the same as calculated before with the horizontal position and C_d and R_e are also the same.

$$\tau = 5648.8$$

$$\ddot{y} = -\frac{0.25 \times 9 \times 10^4}{24(5648.8)}\dot{y} - g$$

$$\ddot{y} = -0.1660\dot{y}$$

$$\ddot{y} + 0.1660\dot{y} = 32.2$$

The homogeneous equation is solve below.

$$R^2 + 0.1660R = 0$$

$$R(R + 0.1660) = 0$$

$$R = 0, -0.1660$$

$$y_h = c_1 e^{0t} + c_2 e^{-0.1660t}$$

$$y_h = c_1 + c_2 e^{-0.1660t}$$

Now the particular solution needs to be found. A guess of $y_p = At + B$ is used to solve for the particular solution.

$$y_p = At + B$$

$$\dot{y_p} = A$$

$$\ddot{y_p} = 0$$

$$0 + 0.1660A = -32.2$$

$$A = -193.98$$

$$y_p = -193.98t$$

The derivative of y was taken to solve for the constants.

$$y = c_1 + c_2 e^{-0.1660t} - 193.98t$$

$$\dot{y} = -0.1660c_2 e^{-0.1660t} - 193.98$$

$$y(0) = c_1 + c_2 = 0$$

$$\dot{y}(0) = -0.1660c_2 - 193.98 = v(0) = V_0 = 120sin30° = 60$$

$$c_2 = -1530$$

$$c_1 = -c_2 = 1530$$

The equation below is the equation for vertical position as a function of time.

$$\boxed{y = 1530 - 1530e^{-0.1660t} - 193.98t}$$

Figure 3 is a plot of projectile motion with fixed values for C_D and R_e.

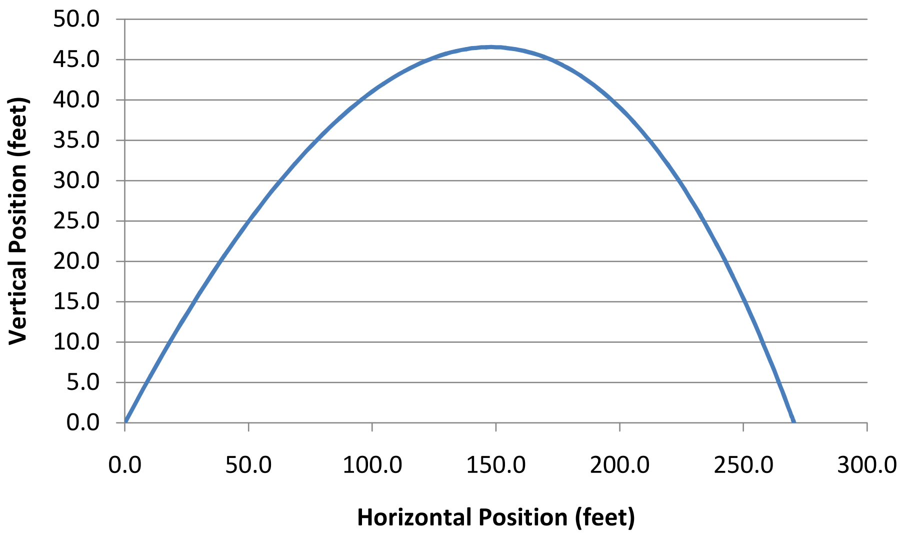

Figure 3: Projectile Motion with R_e = 9 x 10^4 and C_D = 0.25

4. **Solve the system of differential equations using the method of central differences, where the variables and their derivative are approximated as:**

$$\boldsymbol{x(t) = x(t - \Delta t) + \frac{\Delta t}{2}[u(t - \Delta t) + u(t)]}$$

$$\boldsymbol{y(t) = y(t - \Delta t) + \frac{\Delta t}{2}[v(t - \Delta t) + v(t)]}$$

$$\boldsymbol{u(t) = u(t - \Delta t) - \frac{C_D R_e}{24\tau}\left[\frac{u(t) + u(t - \Delta t)}{2}\right]\Delta t}$$

$$\boldsymbol{v(t) = v(t - \Delta t) - \frac{C_D R_e}{24\tau}\left[\frac{v(t) + v(t - \Delta t)}{2}\right]\Delta t - g\Delta t}$$

To complete the method of central differences, it was first necessary to isolate the $u(t)$ and $v(t)$ terms.

For $u(t)$:

$$u(t)+\frac{C_D \operatorname{Re}\Delta t}{48\tau}u(t)=u(t-\Delta t)-\frac{C_D \operatorname{Re}\Delta t}{48\tau}u(t-\Delta t)$$

$$u(t)\left[1+\frac{C_D \operatorname{Re}\Delta t}{48\tau}\right]=u(t-\Delta t)-\frac{C_D \operatorname{Re}\Delta t}{48\tau}u(t-\Delta t)$$

Now *u*(*t*) can be isolated,

$$u(t)=\frac{u(t-\Delta t)-\frac{C_D \operatorname{Re}\Delta t}{48\tau}u(t-\Delta t)}{\left[1+\frac{C_D \operatorname{Re}\Delta t}{48\tau}\right]}$$

Simplifying,

$$u(t)=\frac{48\tau(u(t-\Delta t))-C_D \operatorname{Re}\Delta t(u(t-\Delta t))}{48\tau+C_D \operatorname{Re}\Delta t}$$

For *v*(*t*):

$$v(t)+\frac{C_D \operatorname{Re}\Delta t}{48\tau}v(t)=v(t-\Delta t)-\frac{C_D \operatorname{Re}\Delta t}{48\tau}v(t-\Delta t)-g\Delta t$$

Isolating *v*(*t*):

$$v(t)\left[1+\frac{C_D \operatorname{Re}\Delta t}{48\tau}\right]=v(t-\Delta t)-\frac{C_D \operatorname{Re}\Delta t}{48\tau}v(t-\Delta t)-g\Delta t$$

$$v(t)=\frac{v(t-\Delta t)-\frac{C_D \operatorname{Re}\Delta t}{48\tau}v(t-\Delta t)-g\Delta t}{\left[1+\frac{C_D \operatorname{Re}\Delta t}{48\tau}\right]}$$

Simplifying,

$$v(t)=\frac{48\tau(v(t-\Delta t))-C_D \operatorname{Re}\Delta t(v(t-\Delta t))-48\tau g\Delta t}{48\tau+C_D \operatorname{Re}\Delta t}$$

The C program used to solve the system of differential equations using the method of central differences is in Appendix A. The two equations solve for above were used in the program to calculate u and v. Figure 4 was plotted in excel using the data output by the C program. The projectile motion has a drag coefficient that changes as the velocity of the ball changes. While R_e is less than or equal to $9x10^4$, C_D is equal to 0.4 and while R_e is greater than $9x10^4$, C_D is equal to 0.1.

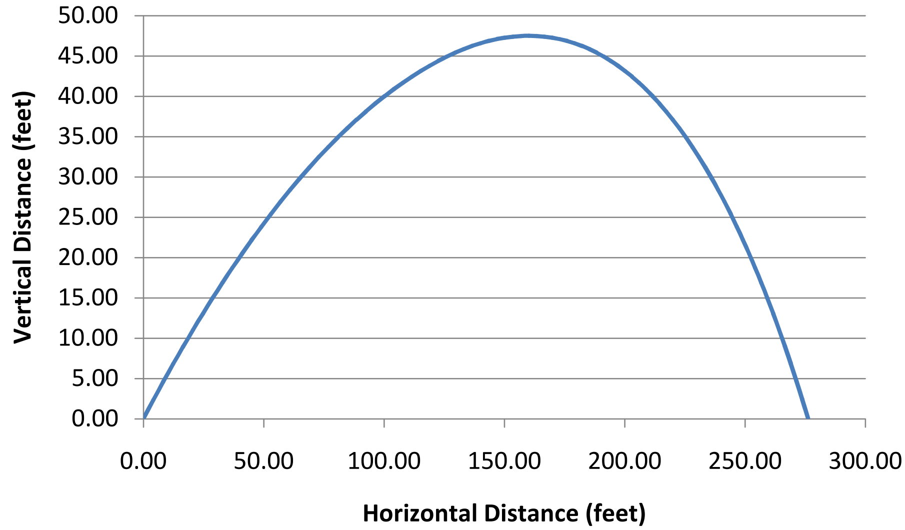

Figure 4: Projectile Motion solved using Central Differences Method

5. **Solve the system of differential equations using a fourth-order Runga-Kutta and plot the path of the ball.**

The C program used to solve the system of differential equations using the Runga-Kutta method is in Appendix A. Figure 5 was plotted in excel using the data output by the C program. The projectile motion has a drag coefficient that changes as the velocity of the ball changes. While R_e is less than or equal to $9x10^4$, C_D is equal to 0.4 and while R_e is greater than $9x10^4$, C_D is equal to 0.1.

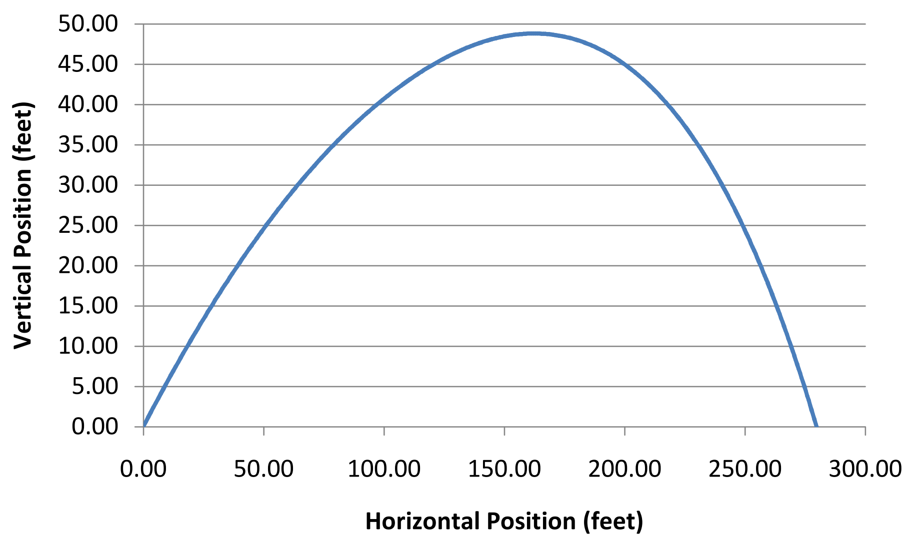

Figure 5: Projectile Motion solved using Runga-Kutta

6. **Obtain the maximum height of the trajectory and the range of the projectile for steps 2-5, and plot the trajectories on a single graph for comparison. Compare the results.**

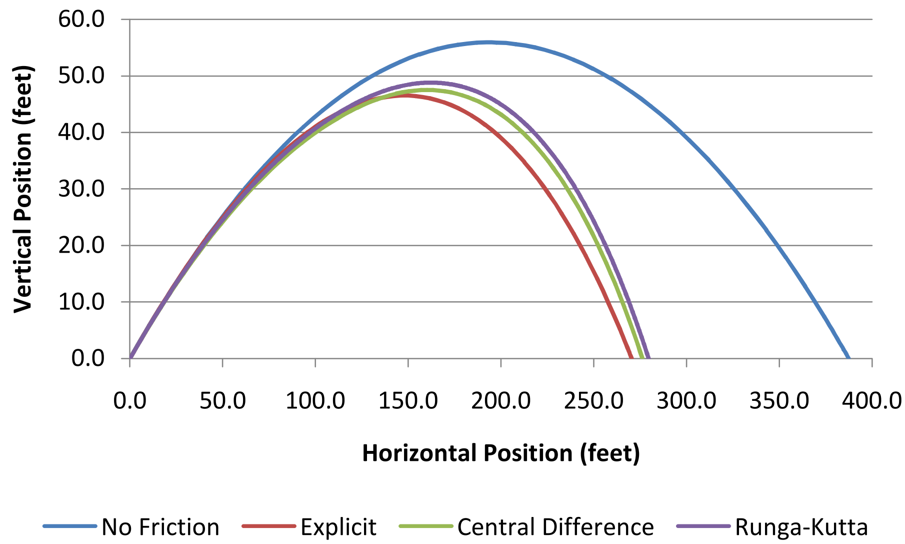

Figure 6: Comparison of Projectile Motion

Table 1 gives the maximum height and range of the golf ball for the four different methods described above. The Runga-Kutta and Central Difference method calculate the Reynolds Number and drag coefficient the same way. The curve, maximum height, and range are very close to each other. The graph from problem 1 does not include friction and therefore the golf ball has a greater range and maximum height than all of the other methods.

Table 1: Maximum Height and Range

Type	Maximum Height (feet)	Range (feet)
No Friction	55.9	387.2
Explicit	46.5	270.4
Central Difference	47.5	276.5
Runga-Kutta	48.8	279.5

The explicit solution has the lowest range and maximum height because it uses a constant drag coefficient and Reynolds number, where the central difference and Runga-Kutta have a drag coefficient and Reynolds number that changes depending on the velocity. Runga-Kutta and the central difference method are the best ways to model projectile motion because of this. A constant drag is not an accurate way to model drag because drag changes depending on the velocity of the golf ball. The best way to model projectile motion with drag is using Runga-Kutta because it uses derivatives where the central difference method uses step sizes and averages.

Appendix A

```
// Chad Paffhausen
// 312 Project Problem 4
// This program solves the system of differential equations
// using the method of central differences.  The values of
// t, x, u, y, and v are printed to a text file.

#include <stdio.h>

// Global Variables

double v, v_new,
       x, x_new,
       u, u_new,
       y, y_new;

double Cdx, Cdy,
       Rex, Rey;

// Defined Global Variables

double      h = 0.04,
      g = 32.2,
      tau = 5648.8,
      mu = 0.000000375,
      rho = 0.002378,
      d = 1.75/12;

// Global Definitions

void step_v(void);
void step_u(void);
void step_x(void);
void step_y(void);

void step_v(void)
{
      v_new = ((v*(48*tau-Cdy*Rey*h)) - (48*tau*g*h))/(48*tau+Cdy*Rey*h);
}

void step_u(void)
{
      u_new = ((u)*((48*tau)-(Cdx*Rex*h)))/((48*tau)+(Cdx*Rex*h));
}

void step_x(void)
{
      x_new = x + (h/2) * (u+u_new);
}

void step_y(void)
{
      y_new = y + (h/2) * (v+v_new);
}


int main()
{
      FILE *fp;
      double t;

      fp = fopen("out.txt", "w");
```

```
        // Initial Conditions

        x = 0.0;
        u = 103.923;
        y = 0.0;
        v = 60;

        fprintf(fp, "t, x, u, y, v\n");     // Heading

        for( t = 0.0; t < 10.0; t += h )
        {
              fprintf(fp, "%f, %f, %f, %f, %f\n", t, x, u, y, v);

              Rex = (rho*u*d)/mu;
              Rey = (rho*v*d)/mu;

              if(Rex <= 90000) Cdx = 0.4;
              if(Rex > 90000) Cdx = 0.1;

              if(Rey <= 90000) Cdy = 0.4;
              if(Rey > 90000) Cdy = 0.1;

              // Calculate u
              step_u();
              u = u_new;

              //Calculate v
              step_v();
              v = v_new;

              // Calculate x
              step_x();
              x = x_new;

              // Calculate y
              step_y();
              y = y_new;
        }
        fclose(fp);
        return 1;
}
```

Appendix B

```
// Chad Paffhausen
// 312 Project Problem 5
// This program solves the system of differential equations
// using Runga-Kutta.  The values of
// t, x, u, y, and v are printed to a text file.

#include <stdio.h>

// Global Variables

#define SIZE 4        // the length of the state vector
#define rho 0.002378  // Air density
#define d 0.1458 // Ball diameter (feet)
#define mu 0.000000375     // Air viscocity
#define m 0.0029115   // Ball mass (slug)
#define g 32.2        // Acceleration due to gravity (ft/s)

// Global Definitions

void multiply(double, double[], double[]);
void add(double[], double[], double[]);
void step(double, double, double[]);
void derivative(double, double[], double[]);

void step(double t, double h, double X[])
{
     double    tmp[SIZE],
          dX[SIZE],
          F1[SIZE],
          F2[SIZE],
          F3[SIZE],
          F4[SIZE];

     // Calculate F1
     derivative(t, X, dX);
     multiply(h, dX, F1);

     // Calculate F2
     multiply(0.5, F1, tmp);
     add(X, tmp, tmp);
     derivative(t+h/2.0, tmp, dX);
     multiply(h, dX, F2);

     // Calculate F3
     multiply(0.5, F2, tmp);
     add(X, tmp, tmp);
     derivative(t+h/2.0, tmp, dX);
     multiply(h, dX, F3);

     // Calculate F4
     add(X, F3, tmp);
     derivative(t+h, tmp, dX);
     multiply(h, dX, F4);

     // Calculate the weighted sum
     add(F2, F3, tmp);
```

```
	multiply(2.0, tmp, tmp);
	add(F1, tmp, tmp);
	add(F4, tmp, tmp);
	multiply(1.0/6.0, tmp, tmp);
	add(tmp, X, X);
}

// State Equations Calculated Here

void derivative(double t, double X[], double dX[])
{
	double Cdx, Cdy, Rex, Rey, tau;

	// calculate Re
	Rex = X[1] * ( (rho * d) / mu);
	Rey = X[3] * ( (rho * d) / mu);

	if(Rex <= 90000) // Determine Cd
		Cdx = 0.4;
	else
		Cdx = 0.1;

	if(Rey <= 90000) // Determine Cd
		Cdy = 0.4;
	else
		Cdy = 0.1;

	tau = m / (3 * 3.14 * d * mu);   // Calculate tau

	dX[0] = X[1];
	dX[1] = (-Cdx * Rex) / (24 * tau) * X[1];
	dX[2] = X[3];
	dX[3] = (-Cdy * Rey) / (24 * tau) * X[3] - g;
}

// A subroutine to add vectors to simplify other equations

void add(double X1[], double X2[], double R[])
{
	int i;

	for(i = 0; i < SIZE; i++)
		R[i] = X1[i] + X2[i];
}

// A subroutine to multiply a vector by a scalar to simplify other
equations

void multiply(double X, double V[], double R[])
{
	int i;

	for(i = 0; i < SIZE; i++) R[i] = X*V[i];
}

int main()
{
	FILE *fp;
```

```
    double h = 0.001;
    double t;
    int j = 0;
    double X[SIZE]; // create state variable list

    X[0] = 0;  // set initial condition for x
    X[1] = 103.9;    // set initial condition for u
    X[2] = 0;  // set initial condition for y
    X[3] = 60; // set initial condition for v

    if( ( fp = fopen("out2.txt", "w")) != NULL)
    {
         fprintf(fp, " t(s) x u y v \n\n");

         for( t = 0.0; t < 5.0; t += h )
         {
              if(j == 0) fprintf(fp, "%9.2f %9.2f %9.2f %9.2f
%9.2f\n", t, X[0], X[1], X[2], X[3]);
              step(t, h, X);

              j++; if(j >= 10) j = 0;
         }
    }
    fclose(fp);

    return 1;
}
```