Appendix: MATLAB Software used for Simulation

Drew Canfield's Analysis of Various Cannons

```
clear;
clc;
hold off

%% Define Initial Parameters
g = 32.2; %ft/s^2
CB = zeros(6,5);
CB(:,1) = [6;12;18;24;32;36];  %Cannon Ball mass lbs
CB(:,2) = [3.49;4.40;5.04;5.55;6.10;6.34];%Cannon Ball Dia inches
CB(:,3) = CB(:,2)*25/24; %Cannon Bore inches
CB(:,4) = CB(:,3)/12; %Bore in ft.
CB(:,5) = (pi/4).*CB(:,4).^2; %Bore Area sq ft
CB(:,6) = CB(:,1)/g ;%mass of CB in slugs
CB(:,7) = (pi/4).*(CB(:,2)/12).^2; %Ball cross section Area sq ft
CB(:,8) = CB(:,2)/12; % Cannon ball dia. feet
CWt     = [1.25;1.87;2.5;6;8;9];% Charge Wt in lbs gunpowder
L_ft    =  CB(:,4)*18;%length of the bore in Feet
Patm = 14.7*144; %DC from Robin 1804
R = 1600; %DC based on 19th century, Collins
GPden = 55; %Gun powder density lbs/ft cu
C_ft = (CWt./GPden)./CB(:,5); %Length of charge in ft

F = zeros(1000,6);
FD = zeros(1000,6);
V = zeros(1000,6);
Vmach = zeros(1000,6);
X = zeros(1000,6);
dx = L_ft(:)/1000; % DC divides up the length of the cannon into 1000th's
CD = 5; %Coefficient of Drag
Fkn = zeros(1000,6);
Vkn = zeros(1000,6);

%% Estimation of Muzzle V including drag
K1 = zeros(6,1);
K2 = zeros(6,1);
K1(:) = R*Patm*CB(:,5).*C_ft(:);
K2(:) = -0.0871*0.5*CD*CB(:,1)/g;

%% Simulation within the Cannon
for k=1:6
for n=1:1000
   X(n,k) = n*dx(k); %DC defines 1000 positions in the canon
   if(X(n,k)>C_ft(k))
      F(n,k) =  R*(C_ft(k)/(n*dx(k)))*Patm*CB(k,5)-.5*.08071*V(n-1).*V(n-
1)*CD*CB(k,5);
        %DC First term  gunpowder force, second term drag
```

```
        if(n>1) %DC allows velocity at bottom of cannon to be zero
           V(n,k) =
1991*sqrt(((CWt(k)./((CB(k,1))+(CWt(k)/3)))*log(X(n,k)./C_ft(k))));
%            V(n,k) =
1991*sqrt(((CWt(k)./((CB(k,1))))*log(X(n,k)./C_ft(k))));

            Vmach(n,k) = V(n,k)./1087.4;
            %DC Current mach number of cannonball

        end
   end
end
end


MuzzleV = V(1000,:)'; %DC exit velocity of cannonball
Re(:,1) = ((CB(:,2).*MuzzleV(:)/(12*1.69))*10000)'; %DC Based on 80 F temp,
true

%% Parameters for Air Simulation
ang = 4/57.3 ;% cannon elevation radians, where 4 is degree of cannon
MN(1:6,1) = 1;
MN(1:6,2) = 2;
x = zeros(12000,6);
y = zeros(12000,6);
Vx = zeros(12001,6);
Vy = zeros(12001,6);
Vt = zeros(12001,6);
Ret = zeros(12000,6);
Ret(1,:)=Re(:,1);
AngA = zeros(12001,6);
CD = zeros(12001,6);
VxFps = Vx(1,:)'; %DC first entry in Vx for each cannonball
VyFps = Vy(1,:)'; %DC first entry in Yx for each cannonball
Table =MN(:,1);
CB_mass = CB(:,1)/g; %Mass of Cannonball in slugs
rho = .074; %density of air lbs/ft^3(1.225kg/m^3)
y0  = 10;  % Height of cannon
nb = zeros(1,6);
dt = .001;
En = zeros(12000,6);
MACH = zeros(12000,6);
Mach = zeros(12000,6);
t = zeros(12000,1);

CannonAlt = 112; %Altitude of Cannon
TgtAltitude = 176; %Altitude of Target
TgtAltD = 176-112; %Altitude difference in feet
EOF = zeros(3,1);

% Drag Coef Model
%SphereCD
hold off
CDA = zeros(2500,2);
M  = zeros(2500,1);
```

```
M = (1:1:2500)/1000;
n  = 1:1:2500;
CDA(1:300,1)= .1;
CDA(301:500,1) = .1+ n(1:200)./6000;
for n=1:400
CDA((500+n),1) = CDA(500,1)+ n*.5/400 + .1*(1+(sin(-pi/2+n*(2*3.1416/400))));
end
for n=1:700
CDA((900+n),1) = CDA(900,1) +.38*(1-exp(-.005*n));
end
CDA(1601:2500,1) = CDA(1600,1);
CDA(700:2500,2) = CDA(700:2500,1);
CDA(1:600,2 )= .47;
for n=1:99
CDA(600+n,2)= CDA(600,2)+ (n*.001);
end

figure(15)
%Plot Drag Coef
semilogx(M(200:2500),CDA(200:2500,1),'k');
hold on;
semilogx(M(200:2500),CDA(200:2500,2),'--');
Limits = [.2,2.5,0,1.5];
axis(Limits);
grid on;
grid minor;
title('CD vs MACH ')
ylabel('CD');
xlabel('MACH');
legend('Re > Transition', 'RE < Transition');
hold off
%end Drag coef++++++++++++++++++++++++++++++++++++++++++++++++++++++++++

%% In Air Simulation
for k=1:6
    x(1,k) = 0;  % Range
    y(1,k) = y0;   % altitude
    Vy(1,k)= sin(ang)*MuzzleV(k);
    Vx(1,k)= cos(ang)*MuzzleV(k);
    Vt(1,k) = sqrt(((Vx(1,k))^2)+((Vy(1,k))^2));
    MACH(1,k)   = MuzzleV(k)/1125.33;

for n=2:12000
    t(n) = n*dt; %time in msec
   MACH(n,k) = Vt(n-1,k)./1125.33;
   Mach(n,k) = 1000*MACH(n,k);% used in drag cal.
    Icd = round(Mach(n,k));% index for drag cal.
    if(Icd>2500)
        Icd = 2500;% Limit of CD
    end
    CD(n,k) = CDA(Icd,1);%Drag coef from function

    Fdrx = .5*rho*(Vx(n-1).^2)*CD(n,k)*CB(k,5);%Drag Force
    dvx = (Fdrx/CB(k,1));
    Fdry = .5*rho*(Vy(n-1).^2)*CD(n,k)*CB(k,5);%Drag Force
    dvy = (Fdry/CB(k,1));
```

```
    Vy(n,k) = Vy(n-1,k)- g*dt-dvy*dt;
    Vx(n,k) = Vx(n-1,k)-dvx*dt;
    En(n,k) = 0.5*CB_mass(k).*Vx(n,k).*Vx(n,k); %DC neglects y energy
    Vt(n,k) = sqrt(Vx(n,k).^2+Vy(n,k).^2);
    Ret(n,k) = (((CB(k,2))*Vt(n,k))/(12*1.69))*10000'; %DC Based on 80 F
temp, true
    AngA(n,k) = atan(Vy(n,k)/Vx(n,k));
    dy = ((Vy(n-1,k)+ Vy(n,k))*dt/2); %average velocity *time
    dx = ((Vx(n-1,k)+ Vx(n,k))*dt/2);
   % CD = .2; % Tupbulent flow, Re ~10^6
    x(n,k)= x(n-1,k)+dx;
    y(n,k) = y(n-1,k)+dy;
    if((y(n,k)<-200)* nb(k)<1) %DC finds the location of when the altitude
        %is less than -200 ft from the starting height
        nb(k) = n;
    end
end
end
 nbp = 0;
for n=1:6
    if nbp<nb(n);
        nbp = nb(n); %finds the latest element position for when the altitude
        %of the cannonball is less than -200 ft from starting altitude
        x_land(n)=nbp;
    end
end
xp = zeros(1,length(x));
xp(:) = x(:,1)'/3;

%% Penetration
A = CB(:,7);%Cross section area ft^2
d = CB(:,2);%penitrator dia in inches
W = CB(:,1); %Cannon ball Wt. lbs.
Vnew = zeros(6,1);
CBN = zeros(6,1);
N = zeros(6,1);
D = zeros(6,1);
S = 10;%Penitrability of Tgt
for G =1:6
Vnew(G) = Vt(nbp,G);%Velocity at Impact ft/sec
CBN(G) = .5/1; %Ln/d for cannon ball nose length r, devided by diameter
N(G) = 0.188*(CBN(G)+.56);  %Tangent Ogives nose performance
K = 0.2*((W(G)).^4);% Soft soil and W<60 lbs
D(G) = .00178.*S.*N(G).*((W(G)./A(G))^.7).*(V(G)-100); %Penitration for soil
in feet
end
Range = x_land';
table(Range, D)

%% Figures/Tables from internal ballistics
Bore = CB(:,3);
CB_dia = CB(:,2);
adeg = [4,4,4,4,4,4]';
Table =MN(:,1);
table(Table,CB_mass,Bore,CB_dia,CWt)
Table = MN(:,2);
```

```
table(Table,adeg,L_ft,C_ft, MuzzleV,VxFps,VyFps, Re)

figure(7)
for n=1:6
plot(X(:,n),F(:,n))
if n == 1;
    hold;
end
plot(X(:,n),F(:,n))
end
title('Force vs Length ')
xlabel('Distance ft')
ylabel('Force lbs')
legend('6lb','12lb','18lb','24lb','32lb','36lb');
hold off

figure(8)
for n=1:6

plot(X(:,n),V(:,n))
if n == 1;
    hold;
end
plot(X(:,n),V(:,n));
end
title('Velocity vs Length  ')
xlabel('Distance ft')
ylabel('Velocity ft/sec')
legend('6lb','12lb','18lb','24lb','32lb','36lb');
hold off

figure(9)
for n=1:6

plot(X(:,n),Vmach(:,n))
if n == 1;
    hold;
end
plot(X(:,n),Vmach(:,n));
end
title('Velocity vs Length  ')
xlabel('Distance ft')
ylabel('Velocity Mach')
legend('6lb','12lb','18lb','24lb','32lb','36lb');
hold off

%% Figures/Tables for Exterior Ballistics
figure(1)
for k=1:6
AngAp(k,:) = AngA(1:12000,k)'*57.3;
plot(xp(1:nbp),AngAp(k,1:nbp))

if k == 1;
    hold;
end
```

```
end
title('AngA vs Distance ')
ylabel('AngA');
xlabel('Range Yds');
legend('6lb','12lb','18lb','24lb','32lb','36lb');
hold off;

MaxAngA = max(AngA(1:12000,1))*57.3;
disp('Maximum Angle of Attack in 1');
disp(MaxAngA)

figure(2)
for k=1:6
Enp(k,:) = En(1:12000,k)';
plot(xp(1:nbp),Enp(k,1:nbp))
if k == 1;
    hold;
end
end
title('En vs Distance ')
ylabel('En ft-lb');
xlabel('Range Yds');
legend('6lb','12lb','18lb','24lb','32lb','36lb');
hold off

figure(3)
for k=1:6
Vxp(k,:) = Vx(1:12000,k)';
plot(xp(1:nbp),Vxp(k,1:nbp))
if k == 1;
    hold;
end
end
title('Vx vs Distance ')
ylabel('Vx ft/sec');
xlabel('Range Yds');
legend('6lb','12lb','18lb','24lb','32lb','36lb');
hold off

figure(4)
for k=1:6
Vyp = Vy(1:12000,k)';
plot(xp(1:nbp),Vyp(1:nbp))
if k == 1;
    hold;
end
end
title('Vy vs Distance ')
ylabel('Vy');
xlabel('Range Yds');
legend('6lb','12lb','18lb','24lb','32lb','36lb');
hold off

Vtp = zeros(1,12001);
figure(5)
for k=1:6
```

```
Vtp(k,:) = Vx(1:12001,k)';
plot(xp(1:nbp),Vtp(k,(1:nbp)))
if k == 1;
    hold;
end
end
title('Velocity vs Distance ')
ylabel('Velocity ft/sec');
xlabel('Range Yds');
legend('6lb','12lb','18lb','24lb','32lb','36lb');
hold off

figure(6)
for k=1:6
Atp(k,:) = y(1:12000,k)';
plot(xp(1:nbp),Atp(k,(1:nbp)))
if k == 1;
    hold;
end
end
title('Altitude vs Distance ')
ylabel('Altitude Feet');
xlabel('Range Yds');
legend('6lb','12lb','18lb','24lb','32lb','36lb');
hold off

maxAlt1 = max(y(1:12000,1));
disp('Maximum Altitude in 1 in ft')
disp(maxAlt1)

figure(10)
Ret = Ret';
for n=1:6
plot(xp(1,:),Ret(n,:));
if n == 1
    hold;
end
end
title('Reynolds Number vs Distance  ')
xlabel('Distance (ft)')
ylabel('Reynolds Number')
legend('6lb','12lb','18lb','24lb','32lb','36lb');
hold off

%% Double Check Assumptions about fluid mechanics
% Reynolds Number Minima per Cannonball
MinRe = zeros(6,1);
for n = 1:6
Min = find((Ret(n,:) == min(Ret(n,:))),1,'first');
MinRe(n,1) = Ret(n,Min);
end
ReMIN = min(MinRe(:,1));
disp('Minimum Reynolds Number');
disp(ReMIN);

MaxRe = zeros(6,1);
```

```
for n = 1:6
Max = find((Ret(n,:) == max(Ret(n,:))),1,'first');
MaxRe(n,1) = Ret(n,Max);
end
ReMAX = max(MaxRe(:,1));
disp('Maximum Reynolds Number');
disp(ReMAX);
```

Drew Canfield's Sensitivity Analysis

Analysis of Cross Wind

```
% Evaluation of Cross Wind Impact on Cannonball Ballistics
clear;
clc;
hold off

%% Define Initial Parameters
CB = zeros(6,5);
CB(:,1) = [6;12;18;24;32;36];  %Cannon Ball mass lbs
CB(:,2) = [3.49;4.40;5.04;5.55;6.10;6.34];%Cannon Ball Dia inches
CB(:,3) = CB(:,2)*25/24; %Cannon Bore inches
CB(:,4) = CB(:,3)/12; %Bore in ft.
CB(:,5) = (pi/4).*CB(:,4).^2; %Bore Area sq ft
CWt     = [1.25;1.87;2.5;6;8;9];% Charge Wt in lbs gunpowder
L_ft    =  CB(:,4)*18;%length of the bore in Feet
Patm = 14.7*144; %DC from Robin 1804
R = 1600; %DC based on 19th century, Collins
GPden = 55; %Gun powder density lbs/ft cu
C_ft = (CWt./GPden)./CB(:,5); %Length of charge in ft

F = zeros(1000,6);
FD = zeros(1000,6);
V = zeros(1000,6);
Vmach = zeros(1000,6);
X = zeros(1000,6);
dx = L_ft(:)/1000; % DC divides up the length of the cannon into 1000th's
g = 32.17; %ft/s^2
CD = 5; %Coefficient of Drag
Fkn = zeros(1000,6);
Vkn = zeros(1000,6);

%% Estimation of Muzzle V including drag
K1 = zeros(6,1);
K2 = zeros(6,1);
K1(:) = R*Patm*CB(:,5).*C_ft(:);
K2(:) = -0.0871*0.5*CD*CB(:,1)/g;

%% Simulation within the Cannon
for k=1:6
for n=1:1000
   X(n,k) = n*dx(k); %DC defines 1000 positions in the canon
   if(X(n,k)>C_ft(k))
```

```
      F(n,k) =  R*(C_ft(k)/(n*dx(k)))*Patm*CB(k,5)-.5*.08071*V(n-1).*V(n-
1)*CD*CB(k,5);
        %DC First term  gunpowder force, second term drag
         if(n>1) %DC allows velocity at bottom of cannon to be zero
            V(n,k) =
1991*sqrt(((CWt(k)./((CB(k,1))+(CWt(k)/3)))*log(X(n,k)./C_ft(k))));

            %DC changed 1928 to 1991 to allign with use of R = 1600,
                %19th century values

             Vmach(n,k) = V(n,k)./1087.4;
             %DC Current mach number of cannonball
         end
   end
end
end


MuzzleV = V(1000,:)'; %DC exit velocity of cannonball
MuzzleVdrag_2 = Vdrag_2(1000,:)';
Re(:,1) = ((CB(:,2).*MuzzleV(:)/(12*1.69))*10000)'; %DC Based on 80 F temp,
true

%% Parameters for Air Simulation
ang = 4/57.3 ;% cannon elevation radians, where 4 is degree of cannon
N(1:6,1) = 1;
N(1:6,2) = 2;
x = zeros(12000,6);
y = zeros(12000,6);
Vx = zeros(12001,6);
Vy = zeros(12001,6);
Vt = zeros(12001,6);
Ret = zeros(12000,6);
Ret(1,:)=Re(:,1);
AngA = zeros(12001,6);
CD = zeros(12001,6);
VxFps = Vx(1,:)'; %DC first entry in Vx for each cannonball
VyFps = Vy(1,:)'; %DC first entry in Yx for each cannonball
Table =N(:,1);
CB_mass = CB(:,1)/g; %Mass of Cannonball in slugs
rho = .074; %density of air lbs/ft^3(1.225kg/m^3) - based on pounds
y0  = 10;  % Height of cannon
nb = zeros(1,6);
dt = .001;
En = zeros(12000,6);
MACH = zeros(12000,6);
Mach = zeros(12000,6);
t = zeros(12000,1);

CannonAlt = 112; %Altitude of Cannon
TgtAltitude = 176; %Altitude of Target
TgtAltD = TgtAltitude - CannonAlt; %Altitude difference in feet
EOF = zeros(6,2);

% Drag Coef Model
```

```
%SphereCD
hold off
CDA = zeros(2500,2);
M  = zeros(2500,1);
M = (1:1:2500)/1000;
n  = 1:1:2500;
CDA(1:300,1)= .1;
CDA(301:500,1) = .1+ n(1:200)./6000;
for n=1:400
CDA((500+n),1) = CDA(500,1)+ n*.5/400 + .1*(1+(sin(-pi/2+n*(2*3.1416/400))));
end
for n=1:700
CDA((900+n),1) = CDA(900,1) +.38*(1-exp(-.005*n));
end
CDA(1601:2500,1) = CDA(1600,1);
CDA(700:2500,2) = CDA(700:2500,1);
CDA(1:600,2 )= .47;
for n=1:99
CDA(600+n,2)= CDA(600,2)+ (n*.001);
end

figure(15)
%Plot Drag Coef
semilogx(M(200:2500),CDA(200:2500,1),'k');
hold on;
semilogx(M(200:2500),CDA(200:2500,2),'--');
Limits = [.2,2.5,0,1.5];
axis(Limits);
grid on;
grid minor;
title('CD vs MACH ')
ylabel('CD');
xlabel('MACH');
legend('Re > Transition', 'RE < Transition');
hold off
%end Drag coef++++++++++++++++++++++++++++++++++++++++++++++++++++++++

%% In Air Simulation
for k=1:6
    x(1,k) = 0;  % Range
    y(1,k) = y0;   % altitude
    z(1,k) = 0; % Horizontal Spread
    Vy(1,k)= sin(ang)*MuzzleV(k);
    Vx(1,k)= cos(ang)*MuzzleV(k);
    Vz(1,k)=0;
    Vt(1,k) = sqrt(((Vx(1,k))^2)+((Vy(1,k))^2)+Vz(1,k)^2);
    MACH(1,k)   = MuzzleV(k)/1125.33;

for n=2:12000
    t(n) = n*dt; %time in msec
   MACH(n,k) = Vt(n-1,k)./1125.33;
   Mach(n,k) = 1000*MACH(n,k);% used in drag cal.
    Icd = round(Mach(n,k));% index for drag cal.
    if(Icd>2500)
        Icd = 2500;% Limit of CD
    end
```

```
    CD(n,k) = CDA(Icd,1);%Drag coef from function

    % Cross Wind Incorporation
    % Average Max Wind Speed in August - 15mph
    Ws = (15*5028)*(1/60)*(1/60); % ft/s
    Ws_Icd = round(abs(Vz(n-1,k)-Ws)*1000);
    if (Ws_Icd>2500)
        Ws_Icd = 2500;
    end
    CD_cw = CDA(Ws_Icd,1);
    Fdrz = 0.5*rho*((Vz(n-1,k)-Ws)^2)*CD_cw*CB(k,5); % Cross Wind Drag
    dvz = (Fdrz/(CB(k,1)));
    Vz(n,k) = Vz(n-1,k)+dvz*dt;
    dz = ((Vz(n-1,k)+Vz(n,k))*dt/2);
    z(n,k)=z(n-1,k)+dz;

    % 2D Ballistics
    Fdrx = .5*rho*(Vx(n-1).^2)*CD(n,k)*CB(k,5);%Drag Force
    dvx = (Fdrx/(CB(k,1)));
    Fdry = .5*rho*(Vy(n-1).^2)*CD(n,k)*CB(k,5);%Drag Force
    dvy = (Fdry/(CB(k,1)));
    Vy(n,k) = Vy(n-1,k)- g*dt-dvy*dt;
    Vx(n,k) = Vx(n-1,k)-dvx*dt;
    En(n,k) = 0.5*CB_mass(k).*Vx(n,k).*Vx(n,k); %DC neglects y energy
    Vt(n,k) = sqrt(Vx(n,k).^2+Vy(n,k).^2+Vz(n,k).^2);
    Ret(n,k) = (((CB(k,2))*Vt(n,k))/(12*1.69))*10000'; %DC Based on 80 F
temp, true
    AngA(n,k) = atan(Vy(n,k)/Vx(n,k));
    dy = ((Vy(n-1,k)+ Vy(n,k))*dt/2); %average velocity *time
    dx = ((Vx(n-1,k)+ Vx(n,k))*dt/2);
   % CD = .2; % Tupbulent flow, Re ~10^6
    x(n,k)= x(n-1,k)+dx;
    y(n,k) = y(n-1,k)+dy;

    if(y(n,k)>TgtAltD)
    EOF(k,1) = x(n,k);
    EOF(k,2) = n;
    end
      if((y(n,k)<-200)* nb(k)<1) %DC finds the location of when the altitude
        %is less than -200 ft from the starting height
        nb(k) = n;
    end
end
end

 nbp = 0;
for n=1:6
    if nbp<nb(n);
        nbp = nb(n); %finds the latest element position for when the altitude
        %of the cannonball is less than -200 ft from starting altitude
    end
    X_land(1,n)=nbp;
end
xp = zeros(1,length(x));
xp(:) = x(:,1)'/3;
```

```
Np = n;
Range = EOF./3;% Range yards
MaxRange = max(Range);

%% Figures/Tables from internal ballistics
Bore = CB(:,3);
CB_dia = CB(:,2);
adeg = [4,4,4,4,4,4]';
Table =N(:,1);
table(Table,CB_mass,Bore,CB_dia,CWt)
Table = N(:,2);
table(Table,adeg,L_ft,C_ft, MuzzleV,VxFps,VyFps, Re)

figure(7)
for n=1:6
plot(X(:,n),F(:,n))
if n == 1;
    hold;
end
plot(X(:,n),F(:,n))
end
title('Force vs Length ')
xlabel('Distance ft')
ylabel('Force lbs')
legend('6lb','12lb','18lb','24lb','32lb','36lb');
hold off

figure(8)
for n=1:6

plot(X(:,n),V(:,n))
if n == 1;
    hold;
end
plot(X(:,n),V(:,n));
end
title('Velocity vs Length  ')
xlabel('Distance ft')
ylabel('Velocity ft/sec')
legend('6lb','12lb','18lb','24lb','32lb','36lb');
hold off

figure(9)
for n=1:6

plot(X(:,n),Vmach(:,n))
if n == 1;
    hold;
end
plot(X(:,n),Vmach(:,n));
end
title('Velocity vs Length  ')
xlabel('Distance ft')
ylabel('Velocity Mach')
legend('6lb','12lb','18lb','24lb','32lb','36lb');
hold off
```

```
%% Figures/Tables for Exterior Ballistics
nbp = max(EOF(:,2))-1;

figure(1)
for k=1:6
AngAp(k,:) = AngA(1:12000,k)'*57.3;
plot(xp(1:nbp),AngAp(k,1:nbp))

if k == 1;
    hold;
end
end
title('AngA vs Distance ')
ylabel('AngA');
xlabel('Range Yds');
legend('6lb','12lb','18lb','24lb','32lb','36lb');
hold off;

figure(2)
for k=1:6
Enp(k,:) = En(1:12000,k)';
plot(xp(1:nbp),Enp(k,1:nbp))
if k == 1;
    hold;
end
end
title('En vs Distance ')
ylabel('En');
xlabel('Range Yds');
legend('6lb','12lb','18lb','24lb','32lb','36lb');
hold off

figure(3)
for k=1:6
Vxp(k,:) = Vx(1:12000,k)';
plot(xp(1:nbp),Vxp(k,1:nbp))
if k == 1;
    hold;
end
end
title('Vx vs Distance ')
ylabel('Vx ft/sec');
xlabel('Range Yds');
legend('6lb','12lb','18lb','24lb','32lb','36lb');
hold off

figure(4)
for k=1:6
Vyp = Vy(1:12000,k)';
plot(xp(1:nbp),Vyp(1:nbp))
if k == 1;
    hold;
end
end
title('Vy vs Distance ')
```

```
ylabel('Vy');
xlabel('Range Yds');
legend('6lb','12lb','18lb','24lb','32lb','36lb');
hold off

Vtp = zeros(1,12001);
figure(5)
for k=1:6
Vtp(k,:) = Vx(1:12001,k)';
plot(xp(1:nbp),Vtp(k,(1:nbp)))
if k == 1;
    hold;
end
end
title('Velocity vs Distance ')
ylabel('Velocity ft/sec');
xlabel('Range Yds');
legend('6lb','12lb','18lb','24lb','32lb','36lb');
hold off

figure(6)
for k=1:6
Atp(k,:) = y(1:12000,k)';
plot(xp(1:nbp),Atp(k,(1:nbp)))
if k == 1;
    hold;
end
end
title('Altitude vs Distance ')
ylabel('Altitude Feet');
xlabel('Range Yds');
legend('6lb','12lb','18lb','24lb','32lb','36lb');
hold off

figure(10)
Ret = Ret';
for n=1:6
plot(xp(1,:),Ret(n,:));
if n == 1
    hold;
end
end
title('Reynolds Number vs Distance  ')
xlabel('Distance (ft)')
ylabel('Reynolds Number')
legend('6lb','12lb','18lb','24lb','32lb','36lb');
hold off

figure(14)
for k=1:6
plot(xp(1:nbp),z(1:nbp,k))
    if (k == 1)
    hold;
end
end
title('Horizontal Spread vs Distance ')
```

```
ylabel('Horizontal Distance (ft)');
xlabel('Range (Yds)');
legend('6lb','12lb','18lb','24lb','32lb','36lb');
hold off

%% Double Check Assumptions about fluid mechanics
% Reynolds Number Minima per Cannonball
MinRe = zeros(6,1);
for n = 1:6
Min = find((Ret(n,:) == min(Ret(n,:))),1,'first');
MinRe(n,1) = Ret(n,Min);
end
disp('Minimum Reynolds Numbers');
disp(MinRe);
ReMIN = min(MinRe(:,1));

MaxRe = zeros(6,1);
for n = 1:6
Max = find((Ret(n,:) == max(Ret(n,:))),1,'first');
MaxRe(n,1) = Ret(n,Max);
end
ReMAX = max(MaxRe(:,1));
%% Analysis of Impact

Hspread = z(end,1);
Land = (X_land(1));

disp('Horizontal Spread in ft');
disp(Hspread);
disp('Maximum Reynolds Number');
disp(ReMAX);
disp('Minimum Reynolds Number');
disp(ReMIN);
disp('Landing Position in ft');
disp(Land);


% Data from Original Simulation

% Minimum Reynolds Number
%      9.224075405632702e+05
%
% Maximum Reynolds Number
%      4.883343867833446e+06
% Range(1,1)
%
% ans =
%
%         7855
% Maximum Angle of Attack in 1
%    4.302281140389114
% Maximum Altitude in 1 in ft
%      1.671552540999079e+02
```

```
DiffReMIN = ReMIN - 9.224075405632702e+05;
DiffReMAX = ReMAX - 4.883343867833446e+06;
DiffLandPos = Land - 7855;
DiffAlt = max(y(1:12000,1))- 1.671552540999079e+02;
DiffAngA = max(AngA(1:12000,1))*57.3 - 4.302281140389114;

disp('Difference in Min. RE')
disp(DiffReMIN)
disp('Difference in Max. RE')
disp(DiffReMAX)
disp('Difference in Landing Position in ft')
disp(DiffLandPos)
disp('Difference in Altitude in ft');
disp(DiffAlt)
disp('Difference in Angle of Attack in degrees')
disp(DiffAngA)
```

Analysis of Vertical Angle

```
% Evaluation of change of Angle on ballistics of 18 lb Cannonball
clear;
clc;
hold off

%% Define Initial Parameters
CB = zeros(1,5);
CB(1,1) = 18;  %Cannon Ball mass lbs
CB(1,2) = 5.04;%Cannon Ball Dia inches
CB(1,3) = CB(1,2)*25/24; %Cannon Bore inches
CB(1,4) = CB(1,3)/12; %Bore in ft.
CB(1,5) = (pi/4).*CB(1,4).^2; %Bore Area sq ft
CWt     = 2.5;% Charge Wt in lbs gunpowder
L_ft    =  CB(1,4)*18;%length of the bore in Feet
Patm = 14.7*144; %DC from Robin 1804
R = 1600; %DC based on 19th century, Collins
GPden = 55; %Gun powder density lbs/ft cu
C_ft = (CWt./GPden)./CB(1,5); %Length of charge in ft

F = zeros(1000,1);
FD = zeros(1000,1);
V = zeros(1000,1);
Vmach = zeros(1000,1);
X = zeros(1000,1);
dx = L_ft(:)/1000; % DC divides up the length of the cannon into 1000th's
g = 32.2; %ft/s^2
CD = 5; %Coefficient of Drag
Fkn = zeros(1000,1);
Vkn = zeros(1000,1);

%% Simulation within the Canon
for n=1:1000
   X(n) = n*dx; %DC defines 1000 positions in the cannon
   if(X(n)>C_ft)
```

```
        F(n) =  R*(C_ft/(n*dx))*Patm*CB(1,5)-.5*.08071*V(n-1).*V(n-
1)*CD*CB(1,5);
            %DC First term  gunpowder force, second term drag
            if(n>1) %DC allows velocity at bottom of cannon to be zero
               V(n) = 1991*sqrt(((CWt./((CB(1,1))+(CWt/3)))*log(X(n)./C_ft)));
                   %DC changed 1928 to 1991 to allign with use of R = 1600,
                   %19th century values
                Vmach(n) = V(n)./1087.4;
                %DC Current mach number of cannonball
            end
      end
end
MuzzleV = V(1000,1)'; %DC exit velocity of cannonball
Re(:,1) = ((CB(:,2).*MuzzleV(:)/(12*1.69))*10000)'; %DC Based on 80 F temp,
true

%% Parameters for Air Simulation
AngV = 3.1:0.1:5;
AngV = AngV./57.3;
x = zeros(12000,20);
y = zeros(12000,20);
X_land = zeros(1,20);
Vx = zeros(12001,20);
Vy = zeros(12001,20);
Vt = zeros(12001,20);
Vx(1,:) = MuzzleV*cos(AngV(:));
Vy(1,:) = MuzzleV*sin(AngV(:));
Ret = zeros(12000,20);
Ret(1,:)=Re(:,1);
AngA = zeros(12001,20);
CD = zeros(12001,20);
VxFps = Vx(1,:)'; %DC first entry in Vx for each cannonball
VyFps = Vy(1,:)'; %DC first entry in Yx for each cannonball
CB_mass = CB(1,1)/g; %Mass of Cannonball in slugs
rho = .074; %density of air lbs/ft^3(1.225kg/m^3)
y0  = 10;  % Height of cannon
nb = zeros(20,1);
dt = .001;
En = zeros(12000,20);
I = 0;

%% In Air Simulation for Varying Angles
for H = 1:20
    ang = AngV(H);
    x(1,H) = 0;  % Range
    y(1,H) = y0;   % altitude
    Vy(1,H)= sin(ang)*MuzzleV;
    Vx(1,H)= cos(ang)*MuzzleV;
    Vt(1,H) = sqrt(Vx(1,H).^2+Vy(1,H).^2);
for n=2:12001
    t = n*dt; %time in msec
    Mach = Vt(n-1,H)/1125.33;
    CD(n,H) = .58+0.58*tanh(2*(Mach -.8));%Drag coef
    Fdrx = .5*rho*(Vx(n-1).^2)*CD(n,H)*CB(1,5);%Drag Force
    dvx = (Fdrx/CB(1,1));
    Fdry = .5*rho*(Vy(n-1).^2)*CD(n,H)*CB(1,5);%Drag Force
```

```
    dvy = (Fdry/CB(1,1));
    Vy(n,H) = Vy(n-1,H)- g*dt-dvy*dt;
    Vx(n,H) = Vx(n-1,H)-dvx*dt;
    En(n,H) = 0.5*CB_mass.*Vx(n,H).*Vx(n,H); %DC neglects y energy
    Vt(n,H) = sqrt(Vx(n,H).^2+Vy(n,H).^2);
    Ret(n,H) = (((CB(1,2))*Vt(n,H))/(12*1.69))*10000'; %DC Based on 80 F
temp, true
    AngA(n,H) = atan(Vy(n,H)/Vx(n,H));
    dy = ((Vy(n-1,H)+ Vy(n,H))*dt/2); %average velocity *time
    dx = ((Vx(n-1,H)+ Vx(n,H))*dt/2);
   % CD = .2; % Tupbulent flow, Re ~10^6
    x(n,H)= x(n-1,H)+dx;
    y(n,H) = y(n-1,H)+dy;
    if((y(n,H)<-200)* nb(H)<1) %DC finds the location of when the altitude
        %is less than -200 ft from the starting height
        nb(H) = n;
    end
    if y(n,H)<-200
        I = I +1;
        if I == 1
            X_land(1,H)=x(n,H);
        end
    end

end
    I = 0;
end

 nbp = 0;
for n=1:6
    if nbp<nb(n);
        nbp = nb(n); %finds the latest element position for when the altitude
        %of the cannonball is less than -200 ft from starting altitude
    end
end
xp = zeros(1,length(x));
xp(:) = x(:,1)'/3;

E_land = X_land(1,20)-X_land(1,1);

%% Figures/Tables from internal ballistics

figure(1)
plot(X,F);
title('Force vs Length ')
xlabel('Distance ft')
ylabel('Force lbs')

figure(2)
plot(X,V);
title('Velocity vs Length  ')
xlabel('Distance ft')
ylabel('Velocity ft/sec')

figure(3)
```

```
plot(X,Vmach);
title('Velocity vs Length  ')
xlabel('Distance ft')
ylabel('Velocity Mach')

%% Figures/Tables for Exterior Ballistics

AngV = AngV.*57.3;
AngA = AngA.*57.3;
figure(4)
for k=1:20
AngAp(k,:) = AngA(1:12000,k)'*57.3;
plot(xp(1:nbp),AngAp(k,1:nbp))

if k == 1;
    hold;
end
end
title('AngA vs Distance ')
ylabel('AngA');
xlabel('Range Yds');
legend(num2str(AngV(1)),num2str(AngV(2)),num2str(AngV(3)),num2str(AngV(4))...
    ,num2str(AngV(5)),num2str(AngV(6)),num2str(AngV(7)),num2str(AngV(8))...

,num2str(AngV(9)),num2str(AngV(10)),num2str(AngV(11)),num2str(AngV(12))...

,num2str(AngV(13)),num2str(AngV(14)),num2str(AngV(15)),num2str(AngV(16))...

,num2str(AngV(17)),num2str(AngV(18)),num2str(AngV(19)),num2str(AngV(20)));
hold off;

figure(5)
for k=1:20
Enp(k,:) = En(1:12000,k)';
plot(xp(1:nbp),Enp(k,1:nbp))
if k == 1;
    hold;
end
end
title('En vs Distance ')
ylabel('En');
xlabel('Range Yds');
legend(num2str(AngV(1)),num2str(AngV(2)),num2str(AngV(3)),num2str(AngV(4))...
    ,num2str(AngV(5)),num2str(AngV(6)),num2str(AngV(7)),num2str(AngV(8))...

,num2str(AngV(9)),num2str(AngV(10)),num2str(AngV(11)),num2str(AngV(12))...

,num2str(AngV(13)),num2str(AngV(14)),num2str(AngV(15)),num2str(AngV(16))...

,num2str(AngV(17)),num2str(AngV(18)),num2str(AngV(19)),num2str(AngV(20)));
hold off

figure(6)
for k=1:20
Vxp(k,:) = Vx(1:12000,k)';
plot(xp(1:nbp),Vxp(k,1:nbp))
```

```
if k == 1;
    hold;
end
end
title('Vx vs Distance ')
ylabel('Vx ft/sec');
xlabel('Range Yds');
legend(num2str(AngV(1)),num2str(AngV(2)),num2str(AngV(3)),num2str(AngV(4))...
    ,num2str(AngV(5)),num2str(AngV(6)),num2str(AngV(7)),num2str(AngV(8))...

,num2str(AngV(9)),num2str(AngV(10)),num2str(AngV(11)),num2str(AngV(12))...

,num2str(AngV(13)),num2str(AngV(14)),num2str(AngV(15)),num2str(AngV(16))...

,num2str(AngV(17)),num2str(AngV(18)),num2str(AngV(19)),num2str(AngV(20)));
hold off

figure(7)
for k=1:20
Vyp = Vy(1:12000,k)';
plot(xp(1:nbp),Vyp(1:nbp))
if k == 1;
    hold;
end
end
title('Vy vs Distance ')
ylabel('Vy');
xlabel('Range Yds');
legend(num2str(AngV(1)),num2str(AngV(2)),num2str(AngV(3)),num2str(AngV(4))...
    ,num2str(AngV(5)),num2str(AngV(6)),num2str(AngV(7)),num2str(AngV(8))...

,num2str(AngV(9)),num2str(AngV(10)),num2str(AngV(11)),num2str(AngV(12))...

,num2str(AngV(13)),num2str(AngV(14)),num2str(AngV(15)),num2str(AngV(16))...

,num2str(AngV(17)),num2str(AngV(18)),num2str(AngV(19)),num2str(AngV(20)));
hold off

Vtp = zeros(12001,20);
figure(8)
for k=1:20
Vtp(:,k) = Vx(:,k)';
plot(xp(1:nbp),Vtp(1:nbp,:))
if k == 1;
    hold;
end
end
title('Velocity vs Distance ')
ylabel('Velocity ft/sec');
xlabel('Range Yds');
legend(num2str(AngV(1)),num2str(AngV(2)),num2str(AngV(3)),num2str(AngV(4))...
    ,num2str(AngV(5)),num2str(AngV(6)),num2str(AngV(7)),num2str(AngV(8))...

,num2str(AngV(9)),num2str(AngV(10)),num2str(AngV(11)),num2str(AngV(12))...

,num2str(AngV(13)),num2str(AngV(14)),num2str(AngV(15)),num2str(AngV(16))...
```

```
,num2str(AngV(17)),num2str(AngV(18)),num2str(AngV(19)),num2str(AngV(20)));
hold off

figure(9)
for k=1:20
Atp(k,:) = y(1:12000,k)';
plot(xp(1:nbp),Atp(k,(1:nbp)))
if k == 1;
    hold;
end
end
title('Altitude vs Distance ')
ylabel('Altitude Feet');
xlabel('Range Yds');
legend(num2str(AngV(1)),num2str(AngV(2)),num2str(AngV(3)),num2str(AngV(4))...
    ,num2str(AngV(5)),num2str(AngV(6)),num2str(AngV(7)),num2str(AngV(8))...

,num2str(AngV(9)),num2str(AngV(10)),num2str(AngV(11)),num2str(AngV(12))...

,num2str(AngV(13)),num2str(AngV(14)),num2str(AngV(15)),num2str(AngV(16))...

,num2str(AngV(17)),num2str(AngV(18)),num2str(AngV(19)),num2str(AngV(20)));
hold off

figure(10)
Ret = Ret';
for n=1:20
plot(xp(1,:),Ret(n,:));
if n == 1;
    hold;
end
plot(xp(1,:),Ret(n,:));
end
title('Reynolds Number vs Distance  ')
xlabel('Distance (ft)')
ylabel('Reynolds Number')
legend(num2str(AngV(1)),num2str(AngV(2)),num2str(AngV(3)),num2str(AngV(4))...
    ,num2str(AngV(5)),num2str(AngV(6)),num2str(AngV(7)),num2str(AngV(8))...

,num2str(AngV(9)),num2str(AngV(10)),num2str(AngV(11)),num2str(AngV(12))...

,num2str(AngV(13)),num2str(AngV(14)),num2str(AngV(15)),num2str(AngV(16))...

,num2str(AngV(17)),num2str(AngV(18)),num2str(AngV(19)),num2str(AngV(20)));
hold off

%% Double Check Assumptions about fluid mechanics
% Reynolds Number Minima per Cannonball
MinRe = zeros(20,1);
for n = 1:20
Min = find((Ret(n,:) == min(Ret(n,:))),1,'first');
MinRe(n,1) = Ret(n,Min);
end
ReMIN = min(MinRe(:,1));
```

```
MaxRe = zeros(20,1);
for n = 1:20
Max = find((Ret(n,:) == max(Ret(n,:))),1,'first');
MaxRe(n,1) = Ret(n,Max);
end
ReMAX = max(MaxRe(:,1));

%% Calculation of Standard Deviation of each measure by the end
Vtmax = max(Vt(:,:));
Vymax = max(Vy(:,:));
Vxmax = max(Vx(:,:));
Emax = max(En(:,:));
Atp = Atp';
Altmax = max(Atp(:,:));
AngMax = max(AngA(:,:));

Vstd = std(Vtmax(:));
Vystd = std(Vymax(:));
Vxstd = std(Vxmax(:));
Enstd = std(Emax(:));
altstd = std(Altmax(:));
Angstd = std(AngMax(:));
nbstd = std(nb(:));
Restd = std(MinRe(:,1));


X = java_array('java.lang.String', 6);
X(1) = java.lang.String('Velocity');
X(2) = java.lang.String('Y_Velocity');
X(3) = java.lang.String('X_Velocity');
X(4) = java.lang.String('Energy');
X(5) = java.lang.String('Reynolds_Number');
X(6) = java.lang.String('nbp');
D = cell(X);
X = java_array('java.lang.String',2);
X(1) = java.lang.String('Altitude');
X(2) = java.lang.String('Angle_Of_Attack');
D2 = cell(X);
StandardDeviations =
table(Vstd,Vystd,Vxstd,Enstd,Restd,nbstd,'variablenames',D)
StandardDeviations2 = table(altstd,Angstd,'variablenames',D2)

disp('Max Re');
disp(ReMAX);
disp('Min Re');
disp(ReMIN);
disp('Max Change in Landing Position');
disp(E_land);
```

Analysis of Horizontal Spread

```
% Evaluation of change of Angle on ballistics of 18 lb Cannonball
clear;
clc;
hold off
```

```
%% Define Initial Parameters
CB = zeros(1,5);
CB(1,1) = 18;  %Cannon Ball mass lbs
CB(1,2) = 5.04;%Cannon Ball Dia inches
CB(1,3) = CB(1,2)*25/24; %Cannon Bore inches
CB(1,4) = CB(1,3)/12; %Bore in ft.
CB(1,5) = (pi/4).*CB(1,4).^2; %Bore Area sq ft
CWt     = 2.5;% Charge Wt in lbs gunpowder
L_ft    =  CB(1,4)*18;%length of the bore in Feet
Patm = 14.7*144; %DC from Robin 1804
R = 1600; %DC based on 19th century, Collins
GPden = 55; %Gun powder density lbs/ft cu
C_ft = (CWt./GPden)./CB(1,5); %Length of charge in ft

F = zeros(1000,1);
FD = zeros(1000,1);
V = zeros(1000,1);
Vmach = zeros(1000,1);
X = zeros(1000,1);
dx = L_ft(:)/1000; % DC divides up the length of the cannon into 1000th's
g = 32.2; %ft/s^2
CD = 5; %Coefficient of Drag
Fkn = zeros(1000,1);
Vkn = zeros(1000,1);

%% Simulation within the Canon
for n=1:1000
   X(n) = n*dx; %DC defines 1000 positions in the cannon
   if(X(n)>C_ft)
      F(n) =  R*(C_ft/(n*dx))*Patm*CB(1,5)-.5*.08071*V(n-1).*V(n-
1)*CD*CB(1,5);
        %DC First term  gunpowder force, second term drag
         if(n>1) %DC allows velocity at bottom of cannon to be zero
            V(n) = 1991*sqrt(((CWt./((CB(1,1))+(CWt/3)))*log(X(n)./C_ft)));
                %DC changed 1928 to 1991 to allign with use of R = 1600,
                %19th century values
             Vmach(n) = V(n)./1087.4;
             %DC Current mach number of cannonball
         end
   end
end
MuzzleV = V(1000,1)'; %DC exit velocity of cannonball
Re(:,1) = ((CB(:,2).*MuzzleV(:)/(12*1.69))*10000)'; %DC Based on 80 F temp,
true

%% Parameters for Air Simulation
AngV = 0.1:0.1:2;
AngV = AngV./57.3;
ang = 4/57.3;
x = zeros(12000,20);
y = zeros(12000,20);
z = zeros(12000,20);
Fdry= zeros(12000,20);
Fdrx= zeros(12000,20);
Fdrz= zeros(12000,20);
```

```
dvy= zeros(12000,20);
dvx= zeros(12000,20);
dvz= zeros(12000,20);
dy= zeros(12000,20);
dx= zeros(12000,20);
dz= zeros(12000,20);
X_land = zeros(1,20);
Vx = zeros(12001,20);
Vy = zeros(12001,20);
Vz = zeros(12001,20);
Vt = zeros(12001,20);
Ret = zeros(12000,20);
t = zeros(12000,20);
Ret(1,:)=Re(:,1);
AngA = zeros(12001,20);
CD = zeros(12001,20);
VxFps = Vx(1,:)'; %DC first entry in Vx for each cannonball
VyFps = Vy(1,:)'; %DC first entry in Yx for each cannonball
CB_mass = CB(1,1)/g; %Mass of Cannonball in slugs
rho = .074; %density of air lbs/ft^3(1.225kg/m^3)
y0  = 10;  % Height of cannon
nb = zeros(20,1);
dt = .001;
En = zeros(12000,20);

% Drag Coef Model
%SphereCD
hold off
CDA = zeros(2500,2);
M  = zeros(2500,1);
M = (1:1:2500)/1000;
n  = 1:1:2500;
CDA(1:300,1)= .1;
CDA(301:500,1) = .1+ n(1:200)./6000;
for n=1:400
CDA((500+n),1) = CDA(500,1)+ n*.5/400 + .1*(1+(sin(-pi/2+n*(2*3.1416/400))));
end
for n=1:700
CDA((900+n),1) = CDA(900,1) +.38*(1-exp(-.005*n));
end
CDA(1601:2500,1) = CDA(1600,1);
CDA(700:2500,2) = CDA(700:2500,1);
CDA(1:600,2 )= .47;
for n=1:99
CDA(600+n,2)= CDA(600,2)+ (n*.001);
end

figure(15)
%Plot Drag Coef
semilogx(M(200:2500),CDA(200:2500,1),'k');
hold on;
semilogx(M(200:2500),CDA(200:2500,2),'--');
Limits = [.2,2.5,0,1.5];
axis(Limits);
grid on;
grid minor;
```

```
title('CD vs MACH ')
ylabel('CD');
xlabel('MACH');
legend('Re > Transition', 'RE < Transition');
hold off

%% In Air Simulation for Varying Angles
for H = 1:20
    ang2 = AngV(H);

    x(1,H) = 0;  % Range
    y(1,H) = y0;   % altitude
    z(1,H) = 0; %Horizontal Spread

    Vy(1,H) = cos(ang2)*sin(ang)*MuzzleV;
    Vx(1,H) = cos(ang)*cos(ang2)*MuzzleV;
    Vz(1,H) = cos(ang)*sin(ang2)*MuzzleV;

    Vt(1,H) = sqrt(Vx(1,H).^2+Vy(1,H).^2+Vz(1,H).^2);

for n=2:12001
    t(n,H) = n*dt; %time in msec
   MACH(n,H) = Vt(n-1,H)./1125.33;
   Mach(n,H) = 1000*MACH(n,H);% used in drag cal.
    Icd = round(Mach(n,H));% index for drag cal.
    if(Icd>2500)
        Icd = 2500;% Limit of CD
    end
    CD(n,H) = CDA(Icd,1);%Drag coef from function

    Fdrx(n,H) = .5*rho*(Vx(n-1,H).^2)*CD(n,H)*CB(1,5);%Drag Force
    dvx(n,H) = (Fdrx(n,H)/CB(1,1));
    Fdry(n,H) = .5*rho*(Vy(n-1,H).^2)*CD(n,H)*CB(1,5);%Drag Force
    dvy(n,H) = (Fdry(n,H)/CB(1,1));
    Fdrz(n,H) = .5*rho*(Vz(n-1,H).^2)*CD(n,H)*CB(1,5);%Drag Force
    dvz(n,H) = (Fdrz(n,H)/CB(1,1));

    Vy(n,H) = Vy(n-1,H)- g*dt-dvy(n,H)*dt;
    Vx(n,H) = Vx(n-1,H)-dvx(n,H)*dt;
    Vz(n,H) = Vz(n-1,H)-dvz(n,H)*dt;
    En(n,H) = 0.5*CB_mass.*Vx(n,H).*Vx(n,H); %DC neglects y energy
    Vt(n,H) = sqrt(Vx(n,H).^2+Vy(n,H).^2+Vz(n,H).^2);

    Ret(n,H) = (((CB(1,2))*Vt(n,H))/(12*1.69))*10000'; %DC Based on 80 F
temp, true
    AngA(n,H) = atan(Vy(n,H)/Vx(n,H));

    dy(n,H) = ((Vy(n-1,H)+ Vy(n,H))*dt/2); %average velocity *time
    dx(n,H) = ((Vx(n-1,H)+ Vx(n,H))*dt/2);
    dz(n,H) = ((Vz(n-1,H)+ Vz(n,H)) *dt/2);
    x(n,H) = x(n-1,H)+dx(n,H);
    y(n,H) = y(n-1,H)+dy(n,H);
    z(n,H) = z(n-1,H)+dz(n,H);

    if((y(n,H)<-200)* nb(H)<1) %DC finds the location of when the altitude
```

```
        %is less than -200 ft from the starting height
        nb(H) = n;
    end
end
end

 nbp = 0;
for n=1:20
    if nbp<nb(n);
        nbp = nb(n); %finds the latest element position for when the altitude
        %of the cannonball is less than -200 ft from starting altitude
    end
    X_land(1,n)=nbp;
end
xp = zeros(1,length(x));
xp(:) = x(:,1)'/3;

E_land = X_land(1,20)-X_land(1,1);

%% Figures/Tables from internal ballistics

figure(1)
plot(X,F);
title('Force vs Length ')
xlabel('Distance ft')
ylabel('Force lbs')

figure(2)
plot(X,V);
title('Velocity vs Length  ')
xlabel('Distance ft')
ylabel('Velocity ft/sec')

figure(3)
plot(X,Vmach);
title('Velocity vs Length  ')
xlabel('Distance ft')
ylabel('Velocity Mach')

%% Figures/Tables for Exterior Ballistics

AngV = AngV.*57.3;
AngA = AngA.*57.3;
figure(4)
for k=1:20
AngAp(k,:) = AngA(1:12000,k)'*57.3;
plot(xp(1:nbp),AngAp(k,1:nbp))

if k == 1;
    hold;
end
end
title('AngA vs Distance ')
ylabel('AngA');
xlabel('Range Yds');
```

```
legend(num2str(AngV(1)),num2str(AngV(2)),num2str(AngV(3)),num2str(AngV(4))...
    ,num2str(AngV(5)),num2str(AngV(6)),num2str(AngV(7)),num2str(AngV(8))...

,num2str(AngV(9)),num2str(AngV(10)),num2str(AngV(11)),num2str(AngV(12))...

,num2str(AngV(13)),num2str(AngV(14)),num2str(AngV(15)),num2str(AngV(16))...

,num2str(AngV(17)),num2str(AngV(18)),num2str(AngV(19)),num2str(AngV(20)));
hold off;

figure(5)
for k=1:20
Enp(k,:) = En(1:12000,k)';
plot(xp(1:nbp),Enp(k,1:nbp))
if k == 1;
    hold;
end
end
title('En vs Distance ')
ylabel('En');
xlabel('Range Yds');
legend(num2str(AngV(1)),num2str(AngV(2)),num2str(AngV(3)),num2str(AngV(4))...
    ,num2str(AngV(5)),num2str(AngV(6)),num2str(AngV(7)),num2str(AngV(8))...

,num2str(AngV(9)),num2str(AngV(10)),num2str(AngV(11)),num2str(AngV(12))...

,num2str(AngV(13)),num2str(AngV(14)),num2str(AngV(15)),num2str(AngV(16))...

,num2str(AngV(17)),num2str(AngV(18)),num2str(AngV(19)),num2str(AngV(20)));
hold off

figure(6)
for k=1:20
Vxp(k,:) = Vx(1:12000,k)';
plot(xp(1:nbp),Vxp(k,1:nbp))
if k == 1;
    hold;
end
end
title('Vx vs Distance ')
ylabel('Vx ft/sec');
xlabel('Range Yds');
legend(num2str(AngV(1)),num2str(AngV(2)),num2str(AngV(3)),num2str(AngV(4))...
    ,num2str(AngV(5)),num2str(AngV(6)),num2str(AngV(7)),num2str(AngV(8))...

,num2str(AngV(9)),num2str(AngV(10)),num2str(AngV(11)),num2str(AngV(12))...

,num2str(AngV(13)),num2str(AngV(14)),num2str(AngV(15)),num2str(AngV(16))...

,num2str(AngV(17)),num2str(AngV(18)),num2str(AngV(19)),num2str(AngV(20)));
hold off

figure(7)
for k=1:20
Vyp = Vy(1:12000,k)';
plot(xp(1:nbp),Vyp(1:nbp))
```

```
if k == 1;
    hold;
end
end
title('Vy vs Distance ')
ylabel('Vy');
xlabel('Range Yds');
legend(num2str(AngV(1)),num2str(AngV(2)),num2str(AngV(3)),num2str(AngV(4))...
    ,num2str(AngV(5)),num2str(AngV(6)),num2str(AngV(7)),num2str(AngV(8))...

,num2str(AngV(9)),num2str(AngV(10)),num2str(AngV(11)),num2str(AngV(12))...

,num2str(AngV(13)),num2str(AngV(14)),num2str(AngV(15)),num2str(AngV(16))...

,num2str(AngV(17)),num2str(AngV(18)),num2str(AngV(19)),num2str(AngV(20)));
hold off

figure(21)
for k=1:20
Vzp = Vz(1:12000,k)';
plot(xp(1:nbp),Vzp(1:nbp))
if k == 1;
    hold;
end
end
title('Vz vs Distance ')
ylabel('Vz (ft/s)');
xlabel('Range Yds');
legend(num2str(AngV(1)),num2str(AngV(2)),num2str(AngV(3)),num2str(AngV(4))...
    ,num2str(AngV(5)),num2str(AngV(6)),num2str(AngV(7)),num2str(AngV(8))...

,num2str(AngV(9)),num2str(AngV(10)),num2str(AngV(11)),num2str(AngV(12))...

,num2str(AngV(13)),num2str(AngV(14)),num2str(AngV(15)),num2str(AngV(16))...

,num2str(AngV(17)),num2str(AngV(18)),num2str(AngV(19)),num2str(AngV(20)));
hold off

figure(22)
for k=1:20
plot(t(:,k),Vz(:,k))
if k == 1;
    hold;
end
end
title('Vz vs Time ')
ylabel('Vz (ft/s)');
xlabel('Time (s)');
legend(num2str(AngV(1)),num2str(AngV(2)),num2str(AngV(3)),num2str(AngV(4))...
    ,num2str(AngV(5)),num2str(AngV(6)),num2str(AngV(7)),num2str(AngV(8))...

,num2str(AngV(9)),num2str(AngV(10)),num2str(AngV(11)),num2str(AngV(12))...

,num2str(AngV(13)),num2str(AngV(14)),num2str(AngV(15)),num2str(AngV(16))...

,num2str(AngV(17)),num2str(AngV(18)),num2str(AngV(19)),num2str(AngV(20)));
```

```
hold off

Vtp = zeros(12001,20);
figure(8)
for k=1:20
Vtp(:,k) = Vx(:,k)';
plot(xp(1:nbp),Vtp(1:nbp,:))
if k == 1;
    hold;
end
end
title('Velocity vs Distance ')
ylabel('Velocity ft/sec');
xlabel('Range Yds');
legend(num2str(AngV(1)),num2str(AngV(2)),num2str(AngV(3)),num2str(AngV(4))...
    ,num2str(AngV(5)),num2str(AngV(6)),num2str(AngV(7)),num2str(AngV(8))...

,num2str(AngV(9)),num2str(AngV(10)),num2str(AngV(11)),num2str(AngV(12))...

,num2str(AngV(13)),num2str(AngV(14)),num2str(AngV(15)),num2str(AngV(16))...

,num2str(AngV(17)),num2str(AngV(18)),num2str(AngV(19)),num2str(AngV(20)));
hold off

figure(9)
for k=1:20
Atp(k,:) = y(1:12000,k)';
plot(xp(1:nbp),Atp(k,(1:nbp)))
if k == 1;
    hold;
end
end
title('Altitude vs Distance ')
ylabel('Altitude Feet');
xlabel('Range Yds');
legend(num2str(AngV(1)),num2str(AngV(2)),num2str(AngV(3)),num2str(AngV(4))...
    ,num2str(AngV(5)),num2str(AngV(6)),num2str(AngV(7)),num2str(AngV(8))...

,num2str(AngV(9)),num2str(AngV(10)),num2str(AngV(11)),num2str(AngV(12))...

,num2str(AngV(13)),num2str(AngV(14)),num2str(AngV(15)),num2str(AngV(16))...

,num2str(AngV(17)),num2str(AngV(18)),num2str(AngV(19)),num2str(AngV(20)));
hold off

figure(20)
for k=1:20
plot(t(1:nbp,k),z(1:nbp,k))
if k == 1;
    hold;
end
end
title('Horizontal Spread vs Time ')
ylabel('Horizontal Spread (ft)');
xlabel('Time (s)');
legend(num2str(AngV(1)),num2str(AngV(2)),num2str(AngV(3)),num2str(AngV(4))...
```

```
    ,num2str(AngV(5)),num2str(AngV(6)),num2str(AngV(7)),num2str(AngV(8))...

,num2str(AngV(9)),num2str(AngV(10)),num2str(AngV(11)),num2str(AngV(12))...

,num2str(AngV(13)),num2str(AngV(14)),num2str(AngV(15)),num2str(AngV(16))...

,num2str(AngV(17)),num2str(AngV(18)),num2str(AngV(19)),num2str(AngV(20)));
hold off

figure(10)
Ret = Ret';
for n=1:20
plot(xp(1,:),Ret(n,:));
if n == 1;
    hold;
end
plot(xp(1,:),Ret(n,:));
end
title('Reynolds Number vs Distance  ')
xlabel('Distance (ft)')
ylabel('Reynolds Number')
legend(num2str(AngV(1)),num2str(AngV(2)),num2str(AngV(3)),num2str(AngV(4))...
    ,num2str(AngV(5)),num2str(AngV(6)),num2str(AngV(7)),num2str(AngV(8))...

,num2str(AngV(9)),num2str(AngV(10)),num2str(AngV(11)),num2str(AngV(12))...

,num2str(AngV(13)),num2str(AngV(14)),num2str(AngV(15)),num2str(AngV(16))...

,num2str(AngV(17)),num2str(AngV(18)),num2str(AngV(19)),num2str(AngV(20)));
hold off

%% Double Check Assumptions about fluid mechanics
% Reynolds Number Minima per Cannonball
MinRe = zeros(20,1);
for n = 1:20
Min = find((Ret(n,:) == min(Ret(n,:))),1,'first');
MinRe(n,1) = Ret(n,Min);
end
ReMIN = min(MinRe(:,1));

MaxRe = zeros(20,1);
for n = 1:20
Max = find((Ret(n,:) == max(Ret(n,:))),1,'first');
MaxRe(n,1) = Ret(n,Max);
end
ReMAX = max(MaxRe(:,1));

%% Calculation of Standard Deviation of each measure by the end
Vtmax = max(Vt(:,:));
Vymax = max(Vy(:,:));
Vxmax = max(Vx(:,:));
Emax = max(En(:,:));
Atp = Atp';
Altmax = max(Atp(:,:));
AngMax = max(AngA(:,:));
```

```
Vstd = std(Vtmax(:));
Vystd = std(Vymax(:));
Vxstd = std(Vxmax(:));
Enstd = std(Emax(:));
altstd = std(Altmax(:));
Angstd = std(AngMax(:));
nbstd = std(nb(:));
Restd = std(MinRe(:,1));

X = java_array('java.lang.String', 6);
X(1) = java.lang.String('Velocity');
X(2) = java.lang.String('Y_Velocity');
X(3) = java.lang.String('X_Velocity');
X(4) = java.lang.String('Energy');
X(5) = java.lang.String('Reynolds_Number');
X(6) = java.lang.String('nbp');
D = cell(X);
X = java_array('java.lang.String',2);
X(1) = java.lang.String('Altitude');
X(2) = java.lang.String('Angle_Of_Attack');
D2 = cell(X);
StandardDeviations =
table(Vstd,Vystd,Vxstd,Enstd,Restd,nbstd,'variablenames',D)
StandardDeviations2 = table(altstd,Angstd,'variablenames',D2)

disp('Max Re');
disp(ReMAX);
disp('Min Re');
disp(ReMIN);
disp('Max Change in Landing Position in ft');
disp(E_land);

HSpread = z(end,20)-z(end,1);
disp('Horizontal Spread in ft');
disp(HSpread);
```

Analysis of Gunpowder Quality

```
% Evaluation of change of quality of gunpowder on ballistics of 18 lb
Cannonball
clear;
clc;
hold off

%% Define Initial Parameters
CB = zeros(1,5);
CB(1,1) = 18;  %Cannon Ball mass lbs
CB(1,2) = 5.04;%Cannon Ball Dia inches
CB(1,3) = CB(1,2)*25/24; %Cannon Bore inches
CB(1,4) = CB(1,3)/12; %Bore in ft.
CB(1,5) = (pi/4).*CB(1,4).^2; %Bore Area sq ft
CWt     = 2.5;% Charge Wt in lbs gunpowder
L_ft    =  CB(1,4)*18;%length of the bore in Feet
Patm = 14.7*144; %DC from Robin 1804
GPden = 55; %Gun powder density lbs/ft cu
C_ft = (CWt./GPden)./CB(1,5); %Length of charge in ft
```

```
F = zeros(1000,11);
FD = zeros(1000,11);
V = zeros(1000,11);
Vmach = zeros(1000,11);
X = zeros(1000,11);
dx = L_ft(:)/1000; % DC divides up the length of the cannon into 1000th's
g = 32.2; %ft/s^2
CD = 5; %Coefficient of Drag
Fkn = zeros(1000,11);
Vkn = zeros(1000,11);

R = 1500:10:1600; %Quality of Gunpowder Changes

%% Simulation within the Canon
for k = 1:11
for n=1:1000
   X(n,k) = n*dx; %DC defines 1000 positions in the cannon
   if(X(n,k)>C_ft)
         F(n,k) =  R(k)*(C_ft/(X(n,k)))*Patm*CB(1,5)-.5*.08071*V(n-1,k).*V(n-
1,k)*CD*CB(1,5);
         if(n>1)
            V(n,k) =
sqrt(2*g*R(k)*(Patm/GPden)*(CWt/CB(1,1))*log(X(n,k)./C_ft));
             Vmach(n,k) = V(n,k)./1087.4;
             %DC Current mach number of cannonball
         end
   end
end
end
MuzzleV = V(1000,:)'; %DC exit velocity of cannonball
Re(:,1) = ((CB(:,2).*MuzzleV(:)/(12*1.69))*10000)'; %DC Based on 80 F temp,
true

%% Parameters for Air Simulation
AngV = 4;
AngV = AngV./57.3;
x = zeros(12000,11);
X_land = zeros(1,11);
y = zeros(12000,11);
Vx = zeros(12001,11);
Vy = zeros(12001,11);
Vt = zeros(12001,11);
Vx(1,:) = MuzzleV*cos(AngV(:));
Vy(1,:) = MuzzleV*sin(AngV(:));
Ret = zeros(12000,11);
Ret(1,:)=Re(:,1);
AngA = zeros(12001,1);
CD = zeros(12001,11);
VxFps = Vx(1,:)'; %DC first entry in Vx for each cannonball
VyFps = Vy(1,:)'; %DC first entry in Yx for each cannonball
CB_mass = CB(1,1)/g; %Mass of Cannonball in slugs
rho = .074; %density of air lbs/ft^3(1.225kg/m^3)
y0  = 10;  % Height of cannon
nb = zeros(11,1);
dt = .001;
```

```
En = zeros(12000,11);
I = 0;

%% In Air Simulation for Varying Angles
for H = 1:11
    ang = AngV;
    x(1,H) = 0;  % Range
    y(1,H) = y0;   % altitude
    Vy(1,H)= sin(ang)*MuzzleV(H,1);
    Vx(1,H)= cos(ang)*MuzzleV(H,1);
    Vt(1,H) = sqrt(Vx(1,H).^2+Vy(1,H).^2);
for n=2:12001
    t = n*dt; %time in msec
    Mach = Vt(n-1,H)/1125.33;
    CD(n,H) = .58+0.58*tanh(2*(Mach -.8));%Drag coef
    Fdrx = .5*rho*(Vx(n-1).^2)*CD(n,H)*CB(1,5);%Drag Force
    dvx = (Fdrx/CB(1,1));
    Fdry = .5*rho*(Vy(n-1).^2)*CD(n,H)*CB(1,5);%Drag Force
    dvy = (Fdry/CB(1,1));
    Vy(n,H) = Vy(n-1,H)- g*dt-dvy*dt;
    Vx(n,H) = Vx(n-1,H)-dvx*dt;
    En(n,H) = 0.5*CB_mass.*Vx(n,H).*Vx(n,H); %DC neglects y energy
    Vt(n,H) = sqrt(Vx(n,H).^2+Vy(n,H).^2);
    Ret(n,H) = (((CB(1,2))*Vt(n,H))/(12*1.69))*10000'; %DC Based on 80 F
temp, true
    AngA(n,H) = atan(Vy(n,H)/Vx(n,H));
    dy = ((Vy(n-1,H)+ Vy(n,H))*dt/2); %average velocity *time
    dx = ((Vx(n-1,H)+ Vx(n,H))*dt/2);
   % CD = .2; % Tupbulent flow, Re ~10^6
    x(n,H)= x(n-1,H)+dx;
    y(n,H) = y(n-1,H)+dy;
    if((y(n,H)<-200)* nb(H)<1) %DC finds the location of when the altitude
        %is less than -200 ft from the starting height
        nb(H) = n;
    end
        if n <12000
        if y(n,H)<-200
        I = I +1;
        if I == 1
            X_land(1,H)=x(n,H);
        end
        else
            X_land(1,H)=x(n,H);
        end
        end
end
    I = 0;

end

E_land = X_land(1,11)-X_land(1,1);

 nbp = 0;
for n=1:6
    if nbp<nb(n);
        nbp = nb(n); %finds the latest element position for when the altitude
```

```
        %of the cannonball is less than -200 ft from starting altitude
    end
end
xp = zeros(1,length(x));
xp(:) = x(:,1)'/3;

%% Figures/Tables from internal ballistics
figure(1)
for n=1:11
plot(X(:,n),F(:,n))
if n == 1;
hold;
end
plot(X(:,n),F(:,n))
end
title('Force vs Length ')
xlabel('Distance ft')
ylabel('Force lbs')
legend(num2str(R(1)),num2str(R(2)),num2str(R(3)),num2str(R(4)),...
    num2str(R(5)),num2str(R(6)),num2str(R(7)),num2str(R(8)),...
    num2str(R(9)),num2str(R(10)),num2str(R(11)));
hold off

figure(2)
for n=1:11
plot(X(:,n),V(:,n))
if n == 1;
hold;
end
plot(X(:,n),V(:,n));
end
title('Velocity vs Length  ')
xlabel('Distance ft')
ylabel('Velocity ft/sec')
legend(num2str(R(1)),num2str(R(2)),num2str(R(3)),num2str(R(4)),...
    num2str(R(5)),num2str(R(6)),num2str(R(7)),num2str(R(8)),...
    num2str(R(9)),num2str(R(10)),num2str(R(11)));
hold off

figure(3)
for n=1:11
plot(X(:,n),Vmach(:,n))
if n == 1;
hold;
end
plot(X(:,n),Vmach(:,n));
end
title('Velocity vs Length  ')
xlabel('Distance ft')
ylabel('Velocity Mach')
legend(num2str(R(1)),num2str(R(2)),num2str(R(3)),num2str(R(4)),...
    num2str(R(5)),num2str(R(6)),num2str(R(7)),num2str(R(8)),...
    num2str(R(9)),num2str(R(10)),num2str(R(11)));
hold off

%% Figures/Tables for Exterior Ballistics
```

```
AngV = AngV.*57.3;
AngA = AngA.*57.3;
figure(4)
for k=1:11
AngAp(k,:) = AngA(1:12001,k)'*57.3;
plot(xp(1,1:nbp),AngAp(k,1:nbp))

if k == 1;
    hold;
end
end
title('AngA vs Distance ')
ylabel('AngA');
xlabel('Range Yds');
legend(num2str(R(1)),num2str(R(2)),num2str(R(3)),num2str(R(4)),...
    num2str(R(5)),num2str(R(6)),num2str(R(7)),num2str(R(8)),...
    num2str(R(9)),num2str(R(10)),num2str(R(11)));
hold off;

figure(5)
for k=1:11
Enp(k,:) = En(1:12001,k)';
plot(xp(1:nbp),Enp(k,1:nbp))
if k == 1;
    hold;
end
end
title('En vs Distance ')
ylabel('En');
xlabel('Range Yds');
legend(num2str(R(1)),num2str(R(2)),num2str(R(3)),num2str(R(4)),...
    num2str(R(5)),num2str(R(6)),num2str(R(7)),num2str(R(8)),...
    num2str(R(9)),num2str(R(10)),num2str(R(11)));
hold off

figure(6)
for k=1:11
Vxp(k,:) = Vx(1:12001,k)';
plot(xp(1:nbp),Vxp(k,1:nbp))
if k == 1;
    hold;
end
end
title('Vx vs Distance ')
ylabel('Vx ft/sec');
xlabel('Range Yds');
legend(num2str(R(1)),num2str(R(2)),num2str(R(3)),num2str(R(4)),...
    num2str(R(5)),num2str(R(6)),num2str(R(7)),num2str(R(8)),...
    num2str(R(9)),num2str(R(10)),num2str(R(11)));
hold off

figure(7)
for k=1:11
Vyp = Vy(1:12001,k)';
plot(xp(1:nbp),Vyp(1:nbp))
```

```
if k == 1;
    hold;
end
end
title('Vy vs Distance ')
ylabel('Vy');
xlabel('Range Yds');
legend(num2str(R(1)),num2str(R(2)),num2str(R(3)),num2str(R(4)),...
    num2str(R(5)),num2str(R(6)),num2str(R(7)),num2str(R(8)),...
    num2str(R(9)),num2str(R(10)),num2str(R(11)));
hold off

Vtp = zeros(12001,20);
figure(8)
for k=1:11
Vtp(:,k) = Vx(:,k)';
plot(xp(1:nbp),Vtp(1:nbp,:))
if k == 1;
    hold;
end
end
title('Velocity vs Distance ')
ylabel('Velocity ft/sec');
xlabel('Range Yds');
legend(num2str(R(1)),num2str(R(2)),num2str(R(3)),num2str(R(4)),...
    num2str(R(5)),num2str(R(6)),num2str(R(7)),num2str(R(8)),...
    num2str(R(9)),num2str(R(10)),num2str(R(11)));
hold off

figure(9)
for k=1:11
Atp(k,:) = y(1:12001,k)';
plot(xp(1:nbp),Atp(k,(1:nbp)))
if k == 1;
    hold;
end
end
title('Altitude vs Distance ')
ylabel('Altitude Feet');
xlabel('Range Yds');
legend(num2str(R(1)),num2str(R(2)),num2str(R(3)),num2str(R(4)),...
    num2str(R(5)),num2str(R(6)),num2str(R(7)),num2str(R(8)),...
    num2str(R(9)),num2str(R(10)),num2str(R(11)));
hold off

figure(10)
Ret = Ret';
for n=1:11
plot(xp(1,:),Ret(n,:));
if n == 1;
    hold;
end
plot(xp(1,:),Ret(n,:));
end
title('Reynolds Number vs Distance  ')
xlabel('Distance (ft)')
```

```
ylabel('Reynolds Number')
legend(num2str(R(1)),num2str(R(2)),num2str(R(3)),num2str(R(4)),...
    num2str(R(5)),num2str(R(6)),num2str(R(7)),num2str(R(8)),...
    num2str(R(9)),num2str(R(10)),num2str(R(11)));
hold off

%% Double Check Assumptions about fluid mechanics
% Reynolds Number Minima per Cannonball
MinRe = zeros(11,1);
for n = 1:11
Min = find((Ret(n,:) == min(Ret(n,:))),1,'first');
MinRe(n,1) = Ret(n,Min);
end
ReMIN = min(MinRe(:,1));

MaxRe = zeros(11,1);
for n = 1:11
Max = find((Ret(n,:) == max(Ret(n,:))),1,'first');
MaxRe(n,1) = Ret(n,Max);
end
ReMAX = max(MaxRe(:,1));

%% Calculation of Standard Deviation of each measure by the end
Vtmax = max(Vt(:,:));
Vymax = max(Vy(:,:));
Vxmax = max(Vx(:,:));
Emax = max(En(:,:));
Atp = Atp';
Altmax = max(Atp(:,:));
AngMax = max(AngA(:,:));

Vstd = std(Vtmax(:));
Vystd = std(Vymax(:));
Vxstd = std(Vxmax(:));
Enstd = std(Emax(:));
altstd = std(Altmax(:));
Angstd = std(AngMax(:));
nbstd = std(nb(:));
Restd = std(MinRe(:,1));

X = java_array('java.lang.String', 6);
X(1) = java.lang.String('Velocity');
X(2) = java.lang.String('Y_Velocity');
X(3) = java.lang.String('X_Velocity');
X(4) = java.lang.String('Energy');
X(5) = java.lang.String('Reynolds_Number');
X(6) = java.lang.String('nbp');
D = cell(X);
X = java_array('java.lang.String',2);
X(1) = java.lang.String('Altitude');
X(2) = java.lang.String('Angle_Of_Attack');
D2 = cell(X);
StandardDeviations =
table(Vstd,Vystd,Vxstd,Enstd,Restd,nbstd,'variablenames',D)
StandardDeviations2 = table(altstd,Angstd,'variablenames',D2)
```

```
disp('Max Re');
disp(ReMAX);
disp('Min Re');
disp(ReMIN);
disp('Max Change in Landing Position');
disp(E_land);
```

Dr Aaron Bradshaw analysis of Soil Penetration

```
%AaronChk
hold off;
A = 20;%area sq in
D = zeros(9,2);
K = 0.65; %Compensation for light wt. penetrator and soft soil
N = 0.65;%dimentionless
S = [8;10]; %dimentionless
V = zeros(9,1);%feet/second
V = [1235;1202;1146;1051;974;911;816;745;690];
W = 18; %Wt of ball,mass slugs
R = [27;55;109;219;328;438;656;875;1094];
PH = [80.7;78.3;75.2;69.7;66.5;63.4;57.9;52.4;47.3];
for n=1:2
D(:,n) = 0.00178*S(n)*N*((W/A)^.7)*(V-100)*K;
Din = D*12;
end
Soil1 = S(1);
Soil2 = S(2);
table(W,A,Soil1,Soil2,N)
PDft1 = D(:,1);
PDft2 = D(:,2);
Din1 = D(:,1).*12;
Din2 = D(:,2).*12;
table(R,V,PDft1,PDft2, Din1,Din2,PH)
figure(1)
plot(R,Din1)
hold on;
plot(R,Din2)
plot(R,PH,'k+');
title('Penetration VS. Range');
xlabel('Range yards');
ylabel('Penitration inches');
legend('S= 8','S= 10','Historic data')
hold off;
figure(2)
plot(V,Din1)
hold on;
```

```
plot(V,Din2)
plot(V,PH,'k+');
title('Penetration VS. Velocity');
xlabel('Velocity at impact ft/sec');
ylabel('Penitration inches');
legend('S= 8','S= 10','Historic data')
```

Bibliography

1. Peterson, Harold *Round Shot and Rammers,* Bonanza Books, New York 1969
2. Map by Lt Edward Fage British Artillery, drawn then updated 1776 to 1779.
3. Manucy, Albert *Artillery Through The Ages,* Div. of Publications National Parks Service1949
4. Bailey,J.B.A. *Field Artillery and Fire Power,* Naval Inst. Press, Annapolis, 2004
5. Robins, Benjamin *New Principles of Gunnery,* London , Wingrave, 1805
   (Available from Google as a Free download)
6. Collins Dr. A. R.*Cannonball Aerodynamic Drag,Down loaded 24January,2016* http;//arc.id.au/CannonballDrag.html
7. Collins, Dr. A. R. *Smooth Boar Cannon Ballistics* Down loaded 24January,2016 http;//arc.id.au/CannonBallistics.html
8. Grice Michael *On Gunnery,* Booksurge, Charlston 2009
9. Wise, Jennings Cropper *Gunnery:An Elementary Treatise*, B.F. Johnson Publishing Co. Richmond VA. 1912