

## Maple Homework Winter 2000

A rocket-missile is launched from  $\mathbf{r}_0 = x_0 \mathbf{i} + y_0 \mathbf{j}$  with an initial velocity of  $\mathbf{v}_0 = v_{0x} \mathbf{i} + v_{0y} \mathbf{j}$ . It burns fuel at the rate of "dmdt", and the speed of the fuel ejected relative to the rocket is ufr. Resistive force on the rocket due to the atmosphere is  $-b\mathbf{v}$ , where b is a constant and  $\mathbf{v}$  is the velocity. We start with choices for the parameters of the motion and the initial conditions.

We now write Newton's second law for the motion of the rocket in the form

$$\mathbf{F}_{\text{ext}} = d\mathbf{P}/dt = (m(\mathbf{v} + d\mathbf{v}) + dm \mathbf{u}_{\text{fuel}}) - (m + dm) \mathbf{v} / dt, \quad (1)$$

where  $\mathbf{u}_{\text{fuel}}$  is the velocity of the fuel relative to the earth,  $dm$  is the mass of the fuel ejected in the time  $dt$ , and the external force is

$$\mathbf{F}_{\text{ext}} = -mg \mathbf{j} - b \mathbf{v} . \quad (2)$$

The force of air resistance acts opposite the direction of motion of the rocket and is assumed to be proportional to the velocity.

This equation will be solved numerically following the motion in time steps of  $\Delta t$ . The mass of the rocket decreases because fuel is ejected, and if we use the index  $i$  to count the steps, with  $i = 0, 1, 2, \dots, N$ , the mass is:

$$m[i] = m_0 - dm/dt * \Delta t * i. \quad (3)$$

Rearranging equation (1) allows us to find the velocity at a time  $t = i \Delta t$ :

$$-g \mathbf{j} - (b/m) \mathbf{v} = d\mathbf{v} + dm/dt (\mathbf{u}_{\text{fuel}} - \mathbf{v}), \quad (4)$$

where we take as our numerical approximation

$$d\mathbf{v} = \mathbf{v}[i] - \mathbf{v}[i-1], \text{ and } (\mathbf{u}_{\text{fuel}} - \mathbf{v}) = \text{velocity of fuel relative to the rocket} = -$$

ufr  $\mathbf{v}/v$ ,

$$v[i] = \sqrt{v_x[i-1]^2 + v_y[i-1]^2}.$$

Incorporating these assumptions into equation (4) allows us to step from the initial velocity  $\mathbf{v}_0$  to future velocities with an approximate solution that improves as  $\Delta t$  gets sufficiently small. Once the velocities are known in time steps, the x and y positions can be obtained as  $d\mathbf{r} = \mathbf{v} dt$ , starting from the initial position  $\mathbf{r}_0$ .

The program for this is given below. We start by entering the constants, and the initial conditions:

```
> N:=250:delt:=0.025:g:=32.174:mo:=2500.0/g:dmdt:=100/g:b:=30.0:ufr:=
=800.0:vo:=80.0:angle:=50.0*3.14159265/180.0:vy[0]:=vo*sin(angle):
vx[0]:=vo*cos(angle):x[0]:=0.0:y[0]:=0.0:ti[0]:=0.0:
vel[0]:=sqrt(vx[0]^2+vy[0]^2):
```

## 1. Trajectory of the rocket when the rate of burn is 100 lb/sec and the launch angle is 50°.

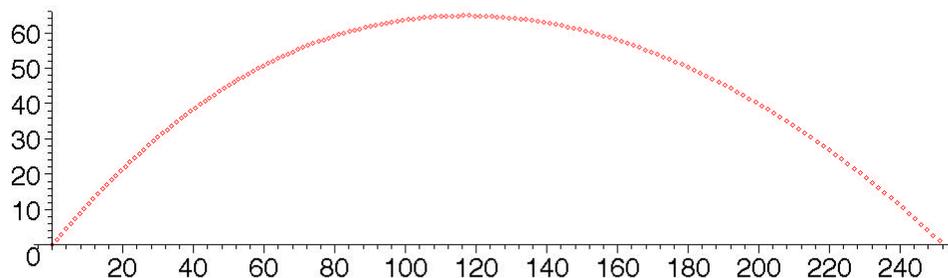
```
>
for i from 0 to N do
v[i]:=sqrt(vx[i]^2+vy[i]^2);
m[i]:=mo-dmdt*i*delt;
ti[i]:=delt*i;
vy[i+1]:=vy[i]-g*delt-b*delt/m[i]*vy[i]+dmdt/m[i]*ufr*delt*vy[i]/v[i];
vx[i+1]:=vx[i]-b*delt/m[i]*vx[i]+dmdt/m[i]*ufr*delt*vx[i]/v[i];
x[i+1]:=x[i]+vx[i]*delt;
y[i+1]:=y[i]+vy[i]*delt; if (y[i] >=0 )and (y[i+1] <=0) then
print(x[i],y[i],i);break fi;
od;
```

251.1339847, 1.208393364, 162

We plot these calculate points x versus y in a point graph. This is the trajectory of the rocket.

```
>
plot([[x[n],y[n]]$n=0..i],style=point, scaling =constrained);
```

## 2. We want to determine the Launch angle necessary for Maximum Horizontal Range.



```

v[i]:=sqrt(vx[i]^2+vy[i]^2);
m[i]:=mo-dmdt*i*delt;
ti[i]:=delt*i;
vy[i+1]:=vy[i]-g*delt-b*delt/m[i]*vy[i]+dmdt/m[i]*ufr*delt*vy[i]/v
[i];
vx[i+1]:=vx[i]-b*delt/m[i]*vx[i]+dmdt/m[i]*ufr*delt*vx[i]/v[i];
x[i+1]:=x[i]+vx[i]*delt;
y[i+1]:=y[i]+vy[i]*delt; if (y[i] >=0 )and (y[i+1] <=0) then break
fi;
od: if maxrange<x[i] then maxrange:=x[i]:maxangle:=angle fi;
od: print(maxangle),(maxrange);

```

64

264.6017474

### 3. To determine the rate of burn needed to achieve a 350 ft horizontal range.

```

> restart:
N:=400:i:=0:delt:=0.025:g:=32.174:mo:=2500.0/g:dmdt:=0.0:dwdt:=0.0
:ufr:=800.0:vo:=80.0:b:=30.0:x[0]:=0.0:y[0]:=0.0:
theta:=60*3.14159265/180.0:vy[0]:=vo*sin(theta):
vx[0]:=vo*cos(theta):
for drdt from 0 to 125 do dmdt:=drdt/g:
for i from 0 to N do
v[i]:=sqrt(vx[i]^2+vy[i]^2);
m[i]:=mo-dmdt*i*delt;
ti[i]:=delt*i;
vy[i+1]:=vy[i]- (m[i]*g+(b*vy[i])-(dmdt*ufr*vy[i]/v[i]))/m[i]*delt;
vx[i+1]:=vx[i]- (b*vx[i]-dmdt*(ufr*vx[i]/v[i]))/m[i]*delt;
x[i+1]:=x[i]+vx[i]*delt;
y[i+1]:=y[i]+vy[i]*delt;
if (y[i]>=0) and (y[i+1] <=0) then break fi;
od:
if ((x[i]-350)<1.5) and ((x[i]-350)>-1.5) then
print (drdt);print(dmdt); print(x[i]);
break
fi;
od:

```

124

3.854043638

351 3435088