Motion in more than one dimension

Vector equations of motion

$$\dot{\vec{x}}(t) = \vec{v}(t)$$
$$\dot{\vec{v}}(t) = \frac{1}{m}\vec{F}[\vec{x}(t), \vec{v}(t), t]$$

Different components (dimensions) coupled through F

Example: Planetary motion (in a 2D plane)

Gravitational force:

$$\vec{F}(r) = -\frac{GMm}{r^3}\vec{r}$$

Two-body problem; can be reduced to one-body problem for effective mass: $\mu = \frac{Mm}{(m+M)}$

Consider M >> m, assume M stationary

Equations of motion for the x and y coordinates

$$\begin{aligned} \dot{x} &= v_x \\ \dot{v}_x &= -GMx/r^3 \\ \dot{v}_y &= -GMy/r^3 \\ \dot{y} &= v_y \end{aligned}$$

$$r = \sqrt{x^2 + y^2}$$

The leapfrog algorithm is

$$\begin{aligned} x(n+1) &= x(n) + \Delta_t v_x(n+1/2) \\ y(n+1) &= y(n) + \Delta_t v_y(n+1/2) \\ v_x(n+1/2) &= v_x(n-1/2) - \Delta_t GMx(n)[x^2(n) + y^2(n)]^{-3/2} \\ v_y(n+1/2) &= v_y(n-1/2) - \Delta_t GMy(n)[x^2(n) + y^2(n)]^{-3/2} \end{aligned}$$

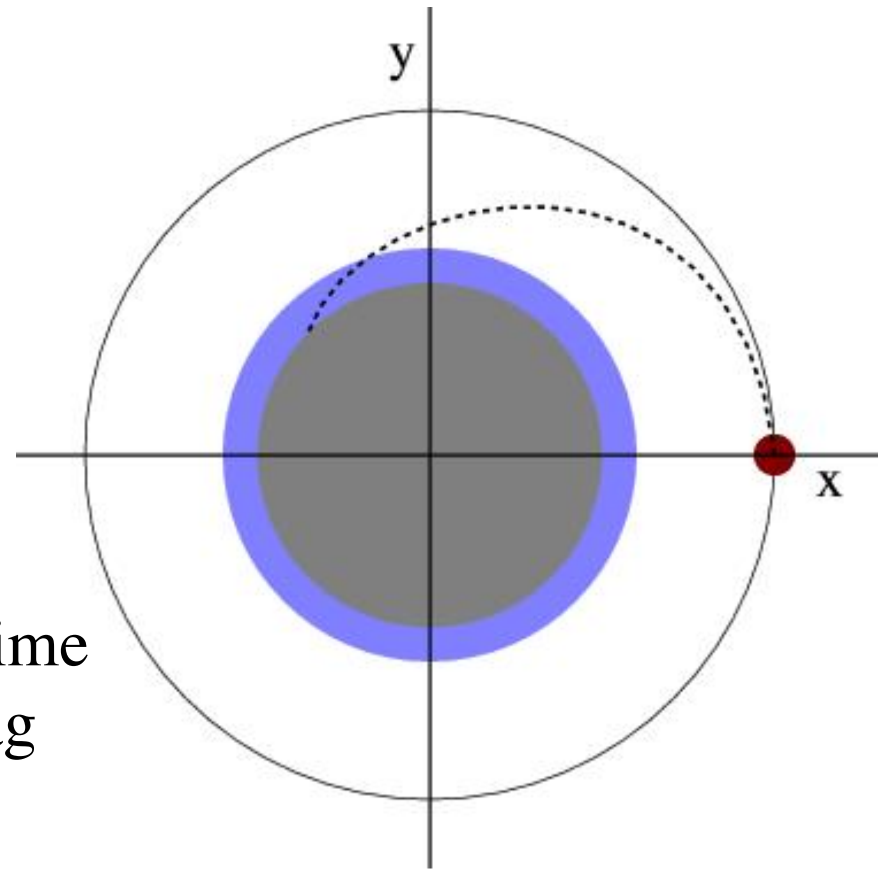
Not much harder than 1D

Runge-Kutta also easily generalizes to D>1

Program example: de-orbiting a satellite

Program ‘crash.f90’ on course web site:

➢ Solves equations of motion for a satellite, including forces of
  - gravitation
  - atmospheric drag
  - thrust of rocket motor for de-orbiting



- We know gravitational force
- Rocket motor causes a constant deceleration for limited (given) time
- We will create a model for air drag

$$F = F_{\text{gravity}}(r)\vec{e}_r + F_{\text{rocket}}(t)\vec{e}_v + F_{\text{drag}}(r,v)\vec{e}_v$$

Gravitation

$$\frac{\vec{F}_{\mathrm{gravity}}}{m} = \frac{GM}{r^2}\vec{e}_r$$

Braking using rocket motor during given time, starting at t=0

$$\frac{\vec{F}_{\mathrm{rocket}}}{m} = \Theta(T_{\mathrm{brake}} - t)B\vec{e}_v$$

Assuming constant deceleration B, e.g., B=5 m/s^2

Atmospheric drag; depends on density of air

$$\frac{\vec{F}_{\mathrm{drag}}}{m} = \frac{C_d}{m}\rho(h)v^2\vec{e}_v \qquad \rho(0) \approx 1.2kg/m^3$$

Adjusting drag-coefficient to give reasonable terminal velocity

$$\frac{F_{\mathrm{drag}}}{m} = g \rightarrow \frac{C_d}{m} = \frac{g}{\rho(h)v_t^2} \qquad \text{(at h=0)}$$

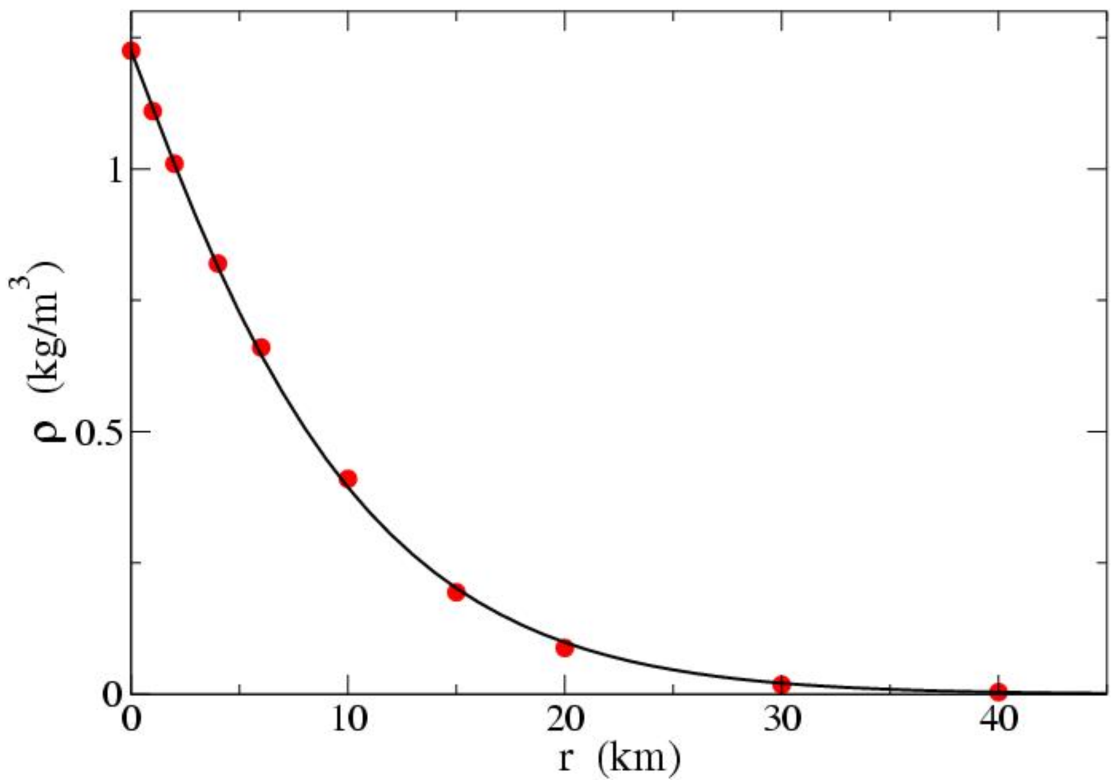
$$\frac{C_d}{m} = 8 \cdot 10^{-4}\frac{m^2}{kg} \quad \text{gives} \quad v_t \approx 100m/s$$

Model for the atmospheric density

This form turns out to give good agreement with data:

$$\rho(h) = 1.225 \cdot \exp\left[-\left(\frac{h}{k_1} + \left(\frac{h}{k_{3/2}}\right)^{3/2}\right)\right] \text{kg/m}^3$$

$$k_1 = 1.2 \cdot 10^4 \text{ m and } k_2 = 2.2 \cdot 10^4 \text{ m}$$



Difficult to model atmosphere > 40 km

Let's see what the model gives for the stability of low orbits

Some elements of the program `crash.f90`

Parameters and some variables in module `systemparam`

```
module systemparam

 real(8), parameter :: pi=3.141592653589793d0
 real(8), parameter :: gm=3.987d14  ! G times M of earth
 real(8), parameter :: arocket=5.d0 ! Deceleration due to engine
 real(8), parameter :: dragc=8.d-4  ! Air drag coefficient / m
 real(8), parameter :: re=6.378d6   ! Earth's radius

 real(8) :: dt,dt2  ! time step, half of the time step
 real(8) :: tbrake  ! run-time of rocket engine

end module systemparam
```

All program units including the statement

`use systemparam`

can access these constants and variables

Main program

Reads input data from the user:

```
print*,'Initial altitude of satellite (km)'; read*,r0
r0=r0*1.d3+re
print*,'Rocket run-time (seconds)'; read*,tbrake
print*,'Time step delta-t for RK integration (seconds)';read*,dt
dt2=dt/2.d0
print*,'Writing results every Nth step; give N';read*,wstp
print*,'Maximum integration time (hours)';read*,tmax
tmax=tmax*3600.d0
```

Sets initial conditions:

```
x=r0
y=0.d0
vx=0.d0
vy=sqrt(gm/r0)
nstp=int(tmax/dt)
```

velocity of object in a Kepler orbit of radius r

$$v = \sqrt{\frac{GM_e}{r}}$$

Opens a file to which data will be written

```
open(1,file='sat.dat',status='replace')
```

Main loop for integrations steps:

```
do i=0,nstp
  call polarposition(x,y,r,a)
  if (r > re) then
    t=dble(i)*dt
    if(mod(i,wstp)==0)write(1,1)t,a,(r-re)/1.d3,sqrt(vx**2+vy**2)
    1 format(f12.3,'  ',f12.8,'  ',f14.6,'  ',f12.4)
    call rkstep(t,x,y,vx,vy)
  else
    print*,'The satellite has successfully crashed!'
    goto 2
  end if
end do
print*,'The satellite did not crash within the specified time.'
2 close(1)
```

Polar coordinates from subroutine `polarposition(x,y,r,a)`

```
r=sqrt(x**2+y**2)
if (y >= 0.d0) then
  a=acos(x/r)/(2.d0*pi)
else
  a=1.d0-acos(x/r)/(2.d0*pi)
end if
```

Runge-Kutta integration step by `rkstep(t0,x0,y0,vx0,vy0)`

```
t1=t0+dt; th=t0+dt2
call accel(x0,y0,vx0,vy0,t0,ax,ay)
kx1=dt2*ax
ky1=dt2*ay
lx1=dt2*vx0
ly1=dt2*vy0
call accel(x0+lx1,y0+ly1,vx0+kx1,vy0+ky1,th,ax,ay)
kx2=dt2*ax; ky2=dt2*ay
lx2=dt2*(vx0+kx1)
ly2=dt2*(vy0+ky1)
call accel(x0+lx2,y0+ly2,vx0+kx2,vy0+ky2,th,ax,ay)
kx3=dt*ax
ky3=dt*ay
lx3=dt*(vx0+kx2)
ly3=dt*(vy0+ky2)
call accel(x0+lx3,y0+ly3,vx0+kx3,vy0+ky3,t1,ax,ay)
kx4=dt2*ax
ky4=dt2*ay
lx4=dt2*(vx0+kx3)
ly4=dt2*(vy0+ky3)
x1=x0+(lx1+2.d0*lx2+lx3+lx4)/3.d0
y1=y0+(ly1+2.d0*ly2+ly3+ly4)/3.d0
vx1=vx0+(kx1+2.d0*kx2+kx3+kx4)/3.d0
vy1=vy0+(ky1+2.d0*ky2+ky3+ky4)/3.d0
x0=x1; y0=y1; vx0=vx1; vy0=vy1
```

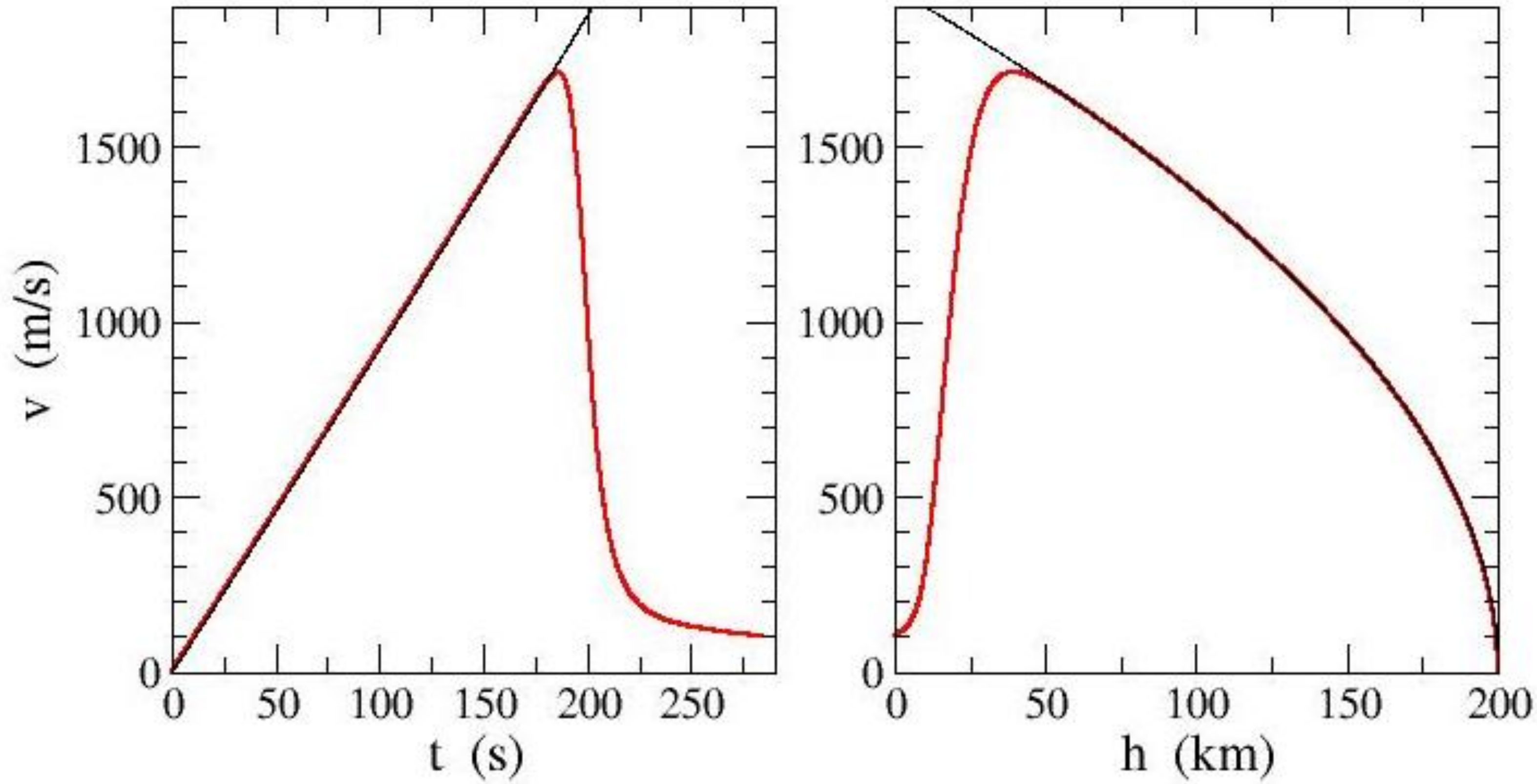
Accelarations calculated in `accel(x,y,vx,vy,t,ax,ay)`

```
r=sqrt(x**2+y**2)
v2=vx**2+vy**2
v1=sqrt(v2)

   !*** evaluates the acceleration due to gravitation
r3=1.d0/r**3
ax=-gm*x*r3
ay=-gm*y*r3

   !*** evaluates the acceleration due to air drag
if (v1 > 1.d-12) then
  ad=dragc*airdens(r)*v2
  ax=ax-ad*vx/v1
  ay=ay-ad*vy/v1
endif

   !*** evaluates the acceleration due to rocket motor thrust
if (t < tbrake .and. v1 > 1.d-12) then
  ax=ax-arocket*vx/v1
  ay=ay-arocket*vy/v1
endif
```

Let's play with the program in various ways...

Start with v=0 (change in program); the satellite drops like a rock

- What happens as it enters the atmosphere?



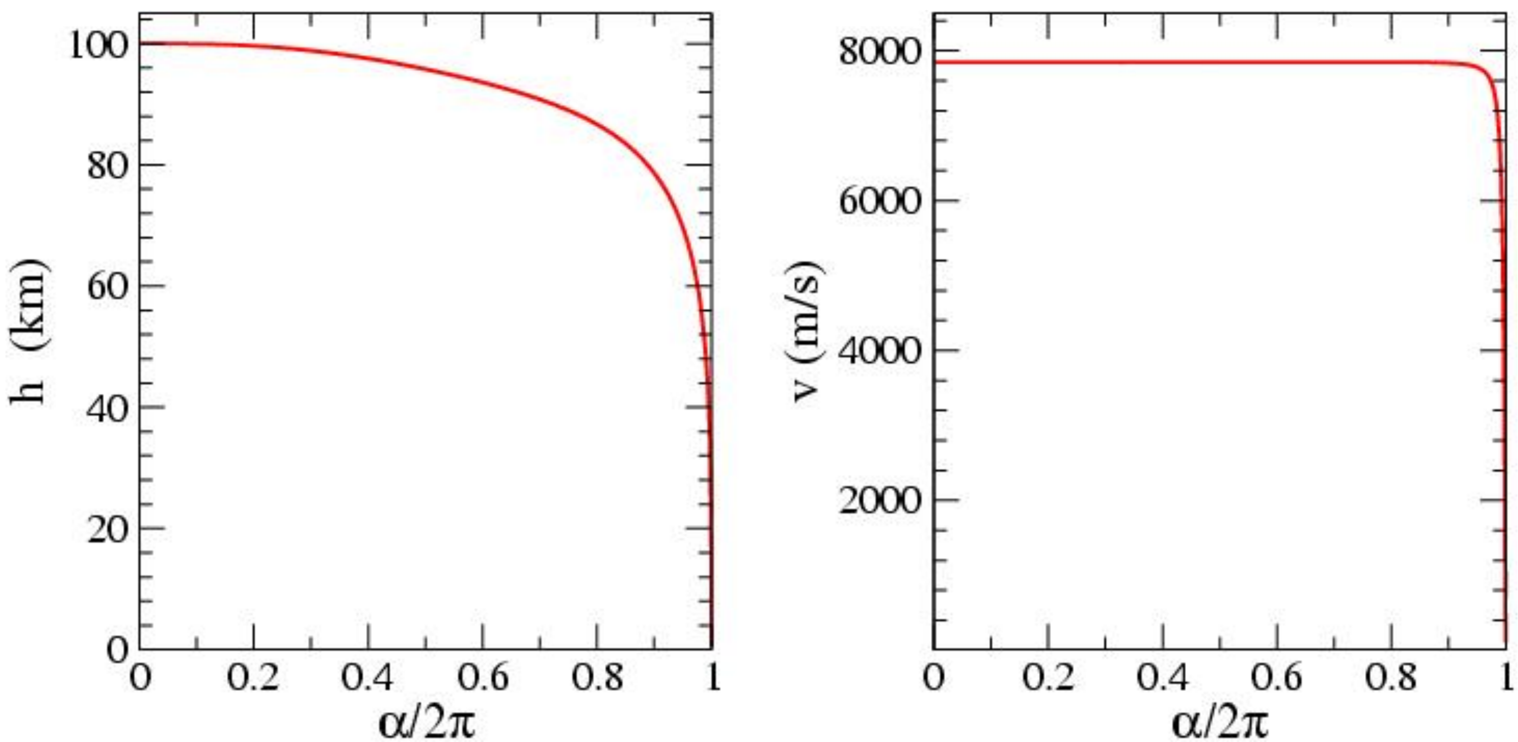
The atmosphere becomes important at 40-50 km altitude

The actual final velocity is 102.6 m/s (exactly 100 m/s terminal velocity results when assuming constant sea-level air density)

Looking at low-altitude orbits without starting the motor

Atmospheric drag brings down the satellite for h < 200 km

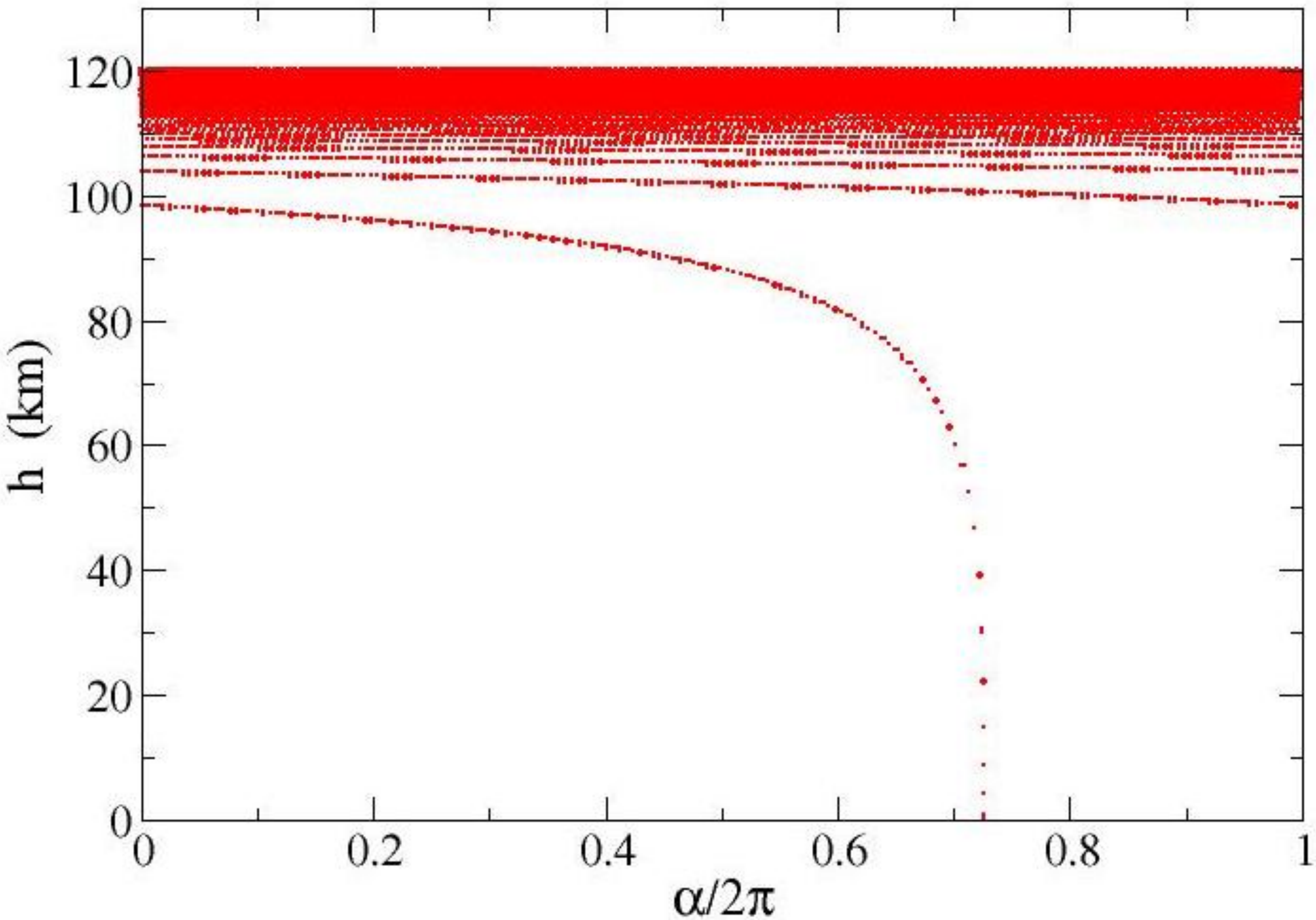
Starting from 100 km, the satellite completes just 1 revolution
(relative air density is $1.5*10^{-8}$ of sea-level density at 100 km)



Velocity little changed until 40 km altitude (direction changes)

Starting from 200 km, the satellite doesn't come down
(model likely has too little atmosphere for > 120 km)

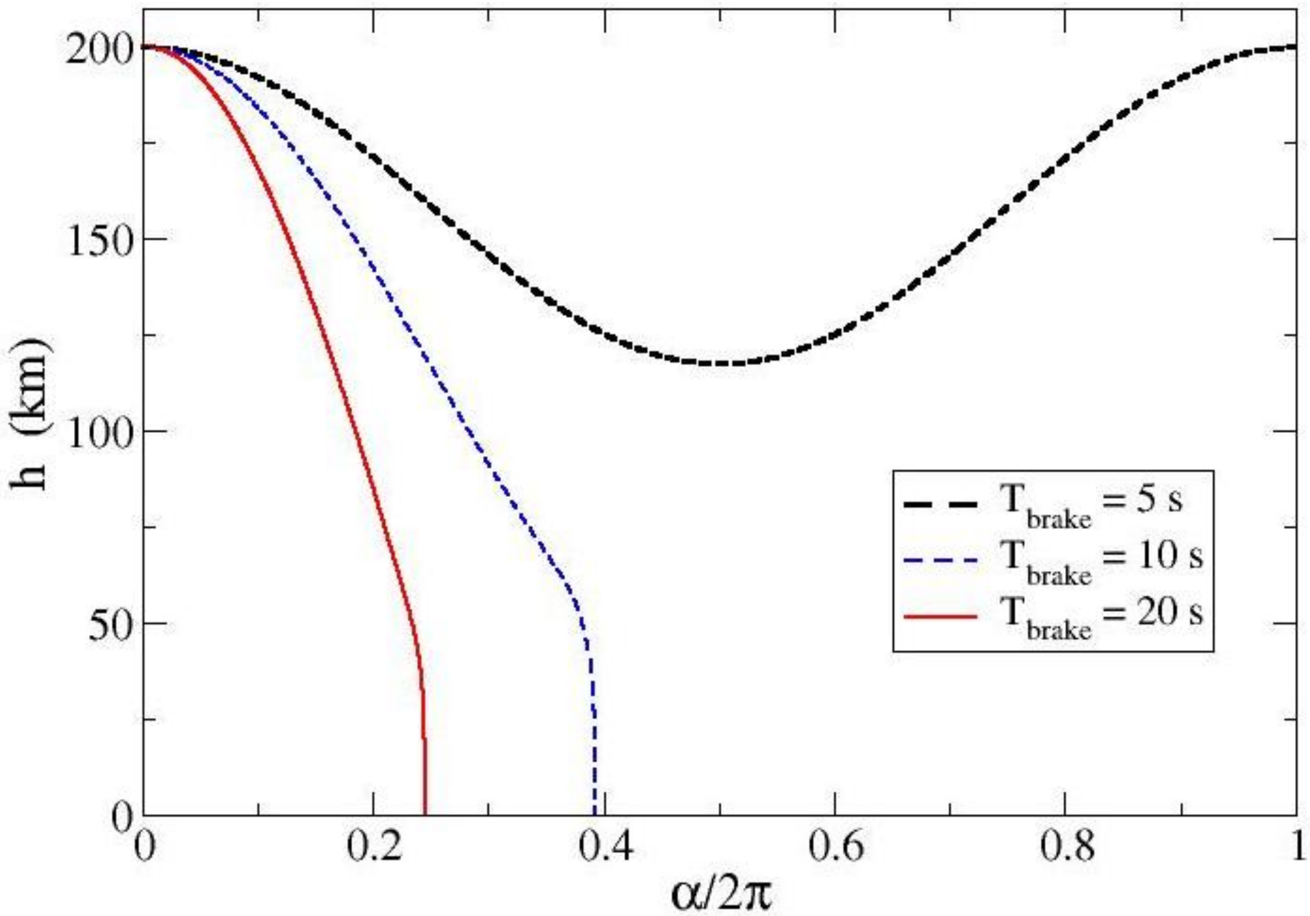
Starting from 120 km; crash in 88 hours



Path sampled every 30 s (integrated using dt = 1 s)

Finally, let's turn on the rocket motor; start at 200 km

Running the motor for 5, 10, 20 seconds



Braking for 5 s at -5 m/s^2 is not enough to bring it down during first revolution; many almost elliptic orbits before crash