Goals for This Lecture:

- Learn how to integrate dynamical equations of motion
- Understand how these equations form a coupled set of first-order ordinary differential equations (ODEs)
- Develop a simple numerical algorithm (Euler's method) to solve such systems of ODEs

Integrating Equations of Motion

- In this lecture we are going to develop several methods for solving dynamical equations of motion
- Our focus in this lecture is on the numerical method itself.
- In the next lecture we will look at how to package up the method in a subprogram
- We'll consider an example of a body in orbit around the sun

The Equations of Motion

- We know that the equations of motion of a point-like body with mass m << M_{sun} are given by:

$$\frac{d\mathbf{x}}{dt} = \mathbf{v} \qquad \frac{d\mathbf{v}}{dt} = \mathbf{a}$$

- In order to specify the dynamics of this body

$$\mathbf{x}(t) \qquad \mathbf{v}(t)$$

we also need to know the position and velocity at time t = 0

$$\mathbf{x}(0) = \mathbf{x}_0 \qquad \mathbf{v}(0) = \mathbf{v}_0$$

Approximating the Equations of Motion

- If we consider a time interval Δt that is sufficiently short we can approximate the time differential by

$$dt \approx \Delta t$$

- We can then approximate the time derivative of the position by $$\frac{d\mathbf{x}}{dt} \approx \frac{\mathbf{x}(t+\Delta t) - \mathbf{x}(t)}{\Delta t}$$

- Similarly the time derivative of the velocity can be approximated by $$\frac{d\mathbf{v}}{dt} \approx \frac{\mathbf{v}(t+\Delta t) - \mathbf{v}(t)}{\Delta t}$$

Euler’s Method for Integrating the Equations of Motion

- We can then substitute the approximated derivatives into the equations of motion to obtain

$$\frac{\mathbf{x}(t+\Delta t)-\mathbf{x}(t)}{\Delta t}\approx\mathbf{v}(t) \qquad \frac{\mathbf{v}(t+\Delta t)-\mathbf{v}(t)}{\Delta t}\approx\mathbf{a}(t)$$

- We can then solve for the new values of the position and velocity

$$\mathbf{x}(t+\Delta t)\approx\mathbf{x}(t)+\mathbf{v}(t)\Delta t \qquad \mathbf{v}(t+\Delta t)\approx\mathbf{v}(t)+\mathbf{a}(t)\Delta t$$

- This is Euler’s method for integrating the Equations of motion forward in time

A Body Orbiting the Sun

- For a body orbiting the Sun we have

$$\mathbf{a}(t) = \frac{-GM_{sun}\mathbf{x(t)}}{r^3}$$

where (if we consider the Sun's location to be at the origin and if we consider the plane of the orbit to be the x-y plane)

$$r^2 = x^2 + y^2$$

- The components of the acceleration are then given by

$$a_x(t) = \frac{-GM_{sun}x(t)}{r^3} \quad a_y(t) = \frac{-GM_{sun}y(t)}{r^3}$$

Euler’s Method Applied to A Body Orbiting the Sun

- Euler’s method then reduces to

$$x(t+\Delta t) = x(t) + v_x(t)\Delta t$$

$$y(t+\Delta t) = y(t) + v_y(t)\Delta t$$

$$v_x(t+\Delta t) = v_x(t) - \frac{GM_{sun}x(t)}{\left(x(t)^2 + y(t)^2\right)^{3/2}}$$

$$v_y(t+\Delta t) = v_y(t) - \frac{GM_{sun}y(t)}{\left(x(t)^2 + y(t)^2\right)^{3/2}}$$

Solving the Problem Computationally

- Let’s now use a computer program to integrate the equations of motion using Euler’s method

- We’ll use units of solar masses, and Astronomical Units for distance
- In these units M_{sun} = 1 and G = 39.47

- At t= 0 we’ll place the body along the x-axis at a distance of 1 AU from the sun and give it the Earth’s velocity in the y-direction

$$v_y = 6.29$$

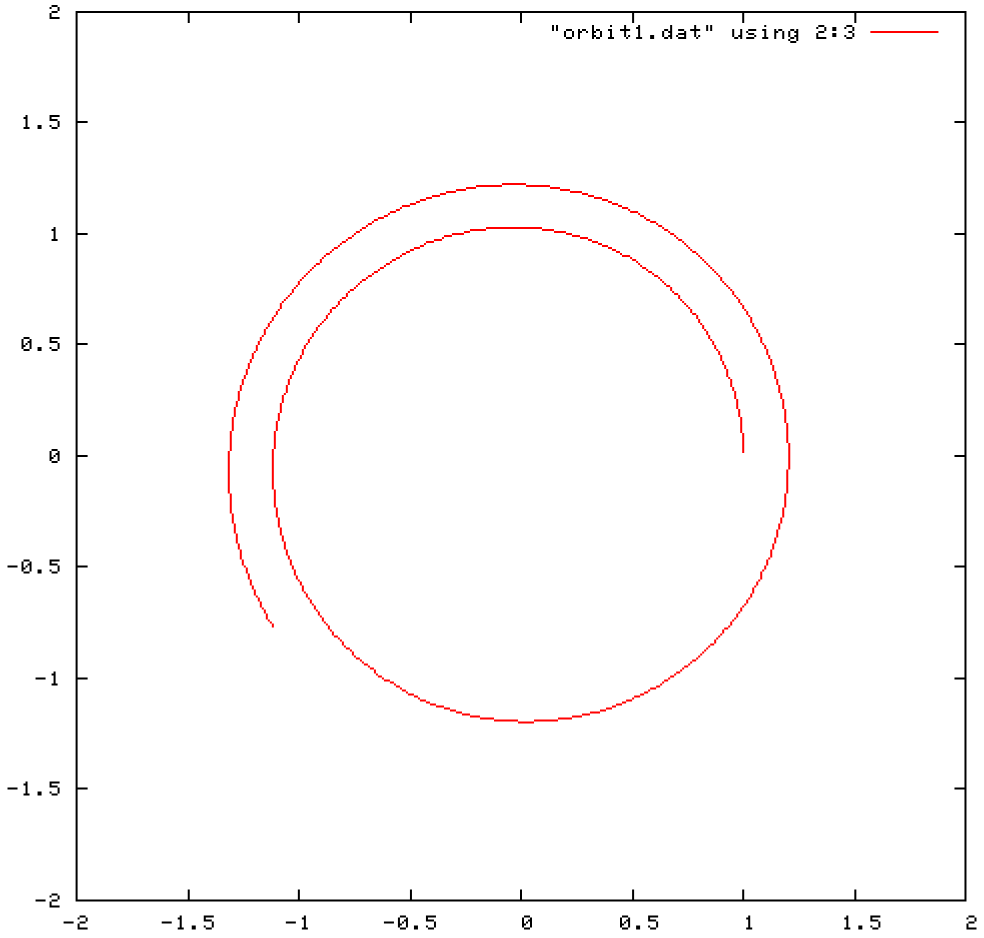
- We’ll take timesteps of 1 day Δt = 1/365 of a year

The Orbit Program

```
program orbit
implicit none
real :: x=1.0d0, y=0.0d0, vx=0.0d0, vy=6.29d0 ! Initial pos. & vel.
real :: t=0.0d0,dt=1.0d0/365.0d0              ! Time & Timestep
real :: xn, yn, vxn, vyn      ! Temporary variables to hold new values
real :: grav=39.47d0, msun=1.0d0
do while(t < 2.0)     ! Integrate for two years
  xn = x+vx*dt        ! X coordinate eqn.
  yn = y+vy*dt        ! Y coordinate eqn.
  vxn = vx-dt*grav*msun*x/(x**2 + y**2)**1.5d0   ! X vel. equation
  vyn = vy-dt*grav*msun*y/(x**2 + y**2)**1.5d0   ! Y vel. equation
  x = xn              ! Copy new values to x, y, vx, & vy
  y = yn
  vx = vxn
  vy = vyn
  t = t+dt            ! Increment time
  write(*,*) t,x,y    ! Write out results
enddo
stop
end program orbit
```

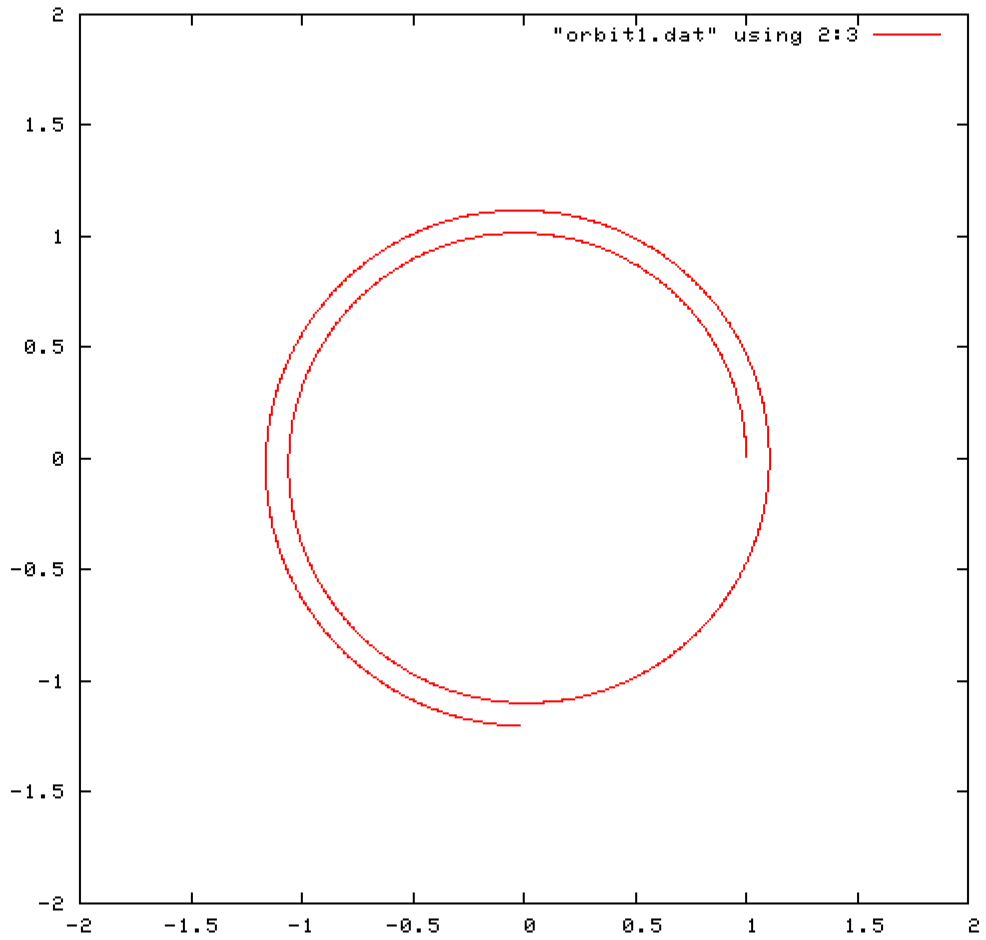
The Results

- Here's what we get:
- Not a very good orbit!
  - Should be circular w/ 2 complete orbits
- Perhaps the Δt is too large?



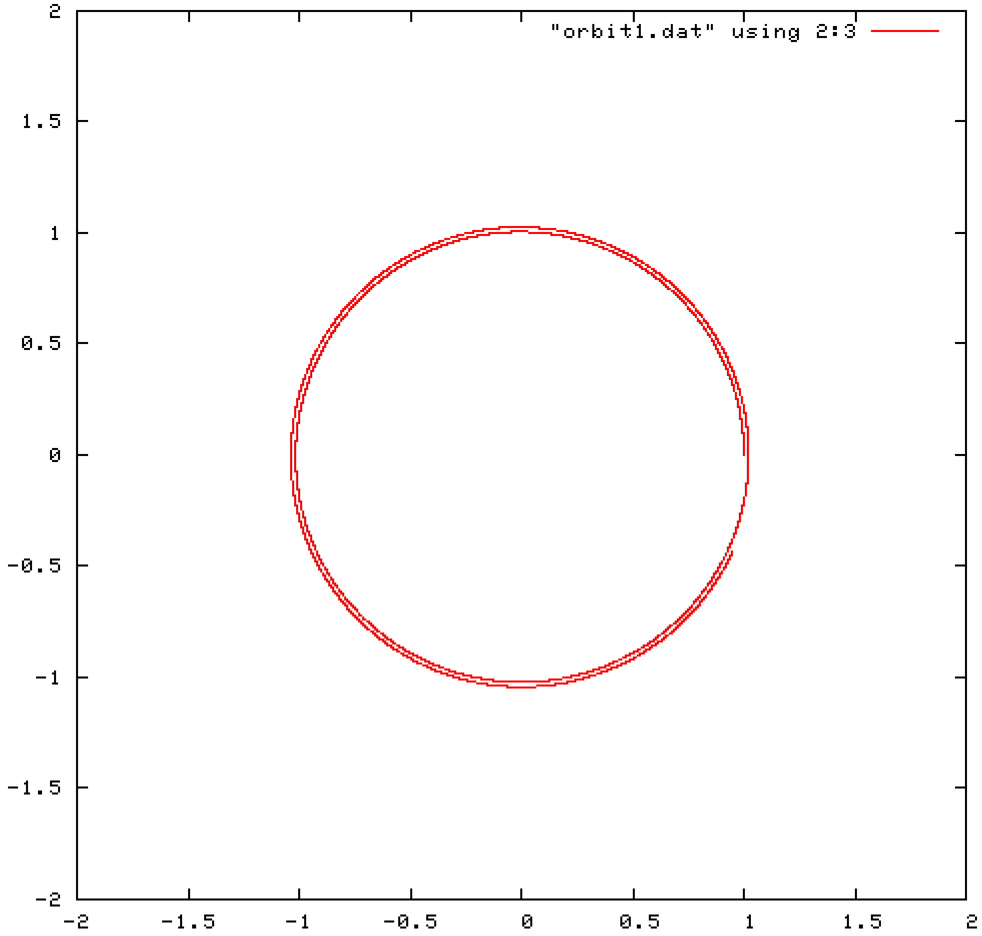
The Results

- Cut the timestep size Δt to 0.5 days
- Here's what we get:
- Look's better but still not great
  - Orbits are tighter but still less than two complete orbits
- Let's make Δt smaller…



The Results

- Cut the timestep size Δt to 0.1 days
- Here’s what we get:
- Now it looks better but far from perfect
- We could improve accuracy with smaller timesteps but perhaps we can find a better numerical method!



Coupled Systems of Ordinary Differential Equations

- The equations we were solving with Euler's method were of the form

$$
\begin{aligned}
\frac{dy_1}{dt} &= f_1(y_1, y_2, \ldots, y_N, t) \\
\frac{dy_2}{dt} &= f_2(y_1, y_2, \ldots, y_N, t) \\
&\vdots \\
\frac{dy_N}{dt} &= f_N(y_1, y_2, \ldots, y_N, t)
\end{aligned}
$$

- This is a set of coupled first-order Ordinary Differential Equations (ODEs)
- In the orbit problem we had four equatios and our four variables were: x, y, v_x v_y

Euler’s Method for Coupled Systems of ODEs

- Let us develop a shorthand notation for the time at the nth step t^n we will denote $y_i(t^n)$ as y_i^n
- Then we can approximate the derivatives as: $\frac{dy_i}{dt} \approx \frac{y_i^{n+1} - y_i^n}{\Delta t}$
- Thus Euler’s method for a set of couple ODEs is:

$$\begin{aligned} y_1^{n+1} &= y_1^n + \Delta t f_1(y_1^n, y_2^n, \ldots, y_N^n, t^t) \\ y_2^{n+1} &= y_2^n + \Delta t f_2(y_1^n, y_2^n, \ldots, y_N^n, t^t) \\ &\vdots \\ y_N^{n+1} &= y_N^n + \Delta t f_N(y_1^n, y_2^n, \ldots, y_N^n, t^t) \end{aligned}$$

- Exercise: Convince yourself that the orbital equations fit the form described in the previous slide and that the numerical method we used fits into the form shown above

Assignment

- Get your own version of the orbit code working and make the plots shown in the previous slides using gnuplot

- We will recode this and other algorithms in C++ shortly