Simple Harmonic Motion

Methods for Ordinary Differential Equations

Dr. McDonnell

Fall 2018

Simple Harmonic Motion

Simple harmonic motion occurs for simple, linear restoring forces:

$$m\frac{d^2x}{dt^2} = -kx,$$

where m is the mass of the object, $x(t)$ is its position at a given time t, and k is some positive constant describing the strength of the restoring force.

Preliminary: Finite Differences

- Forward difference:

$$\frac{dx}{dt} \approx \frac{x(t + \Delta t) - x(t)}{\Delta t}$$

- Backward difference:

$$\frac{dx}{dt} \approx \frac{x(t) - x(t - \Delta t)}{\Delta t}$$

- Central difference:

$$\frac{dx}{dt} \approx \frac{x(t + \frac{1}{2}\Delta t) - x(t - \frac{1}{2}\Delta t)}{\Delta t}$$

Which one we use depends on the context. For stepping forward in time, the forward difference makes sense.

Preliminary: Reduce Second Order to First Order

It is convenient to transform our second order equation into a system of two first order equations by defining an intermediate function,

$$v(t) = \frac{dx}{dt}$$

We then have two equations:

$$\begin{aligned} \frac{dx}{dt} &= v \\ m\frac{dv}{dt} &= -kx \end{aligned}$$

ODE Method 1: The Euler Method

The goal is to solve for the system variables at the next time step, in terms of the information we have at each "current" time step. The **Euler method** uses the forward-difference scheme above,

$$\begin{aligned}\frac{x(t+\Delta t)-x(t)}{\Delta t} &= v(t) \\ \frac{v(t+\Delta t)-v(t)}{\Delta t} &= -\frac{k}{m}x(t).\end{aligned}$$

Solve each of these equations algebraically for the system variables at the next time step:

$$\begin{aligned}x(t+\Delta t) &= x(t)+v(t)\Delta t \\ v(t+\Delta t) &= v(t)-\frac{k}{m}x(t)\Delta t.\end{aligned}$$

ODE Method 1: The Euler Method

Advantages:

- It's intuitive!
- It's easy to apply to many systems.

Disadvantages:

- It is very sensitive to the size of the time step Δt.
- It is easy to obtain unphysical results.

Theory: System Vector

Define a "system vector"

$$\vec{y}(t) = \begin{bmatrix} x(t) \\ v(t) \end{bmatrix}$$

We can package the previous system of two equations into a single vector equation,

$$\frac{d\vec{y}}{dt} = \vec{f}(t,\ \vec{y}(t)),$$

where

$$\vec{f}(t,\ \vec{y}(t)) = \begin{bmatrix} v \\ -\frac{k}{m}x \end{bmatrix}$$

ODE Method 2: The Runge-Kutta Method

A very useful method is obtained via integrating a Taylor polynomial approximation to the system vector $\vec{y}(t)$. For what is called the fourth-order **Runge-Kutta** algorithm, it looks like making four "mini" time steps:

$$
\begin{aligned}
\vec{k}_1 &= \Delta t\, \vec{f}(t,\ \vec{y}(t)) \\
\vec{k}_2 &= \Delta t\, \vec{f}\left(t + \frac{\Delta t}{2},\ \vec{y}(t) + \frac{\vec{k}_1}{2}\right) \\
\vec{k}_3 &= \Delta t\, \vec{f}\left(t + \frac{\Delta t}{2},\ \vec{y}(t) + \frac{\vec{k}_2}{2}\right) \\
\vec{k}_4 &= \Delta t\, \vec{f}\left(t + \Delta t,\ \vec{y}(t) + \vec{k}_3\right) \\
\vec{y}(t + \Delta t) &= \vec{y}(t) + \frac{1}{6}\left(\vec{k}_1 + 2\vec{k}_2 + 2\vec{k}_3 + \vec{k}_4\right)
\end{aligned}
$$

ODE Method 3: A Pre-packaged Solver

Check out the `scipy.integrate.odeint` library routine...

Review

- Solving the dynamics of many systems requires solving an ODE, which involves solving for the variables at the next time step in terms of information from the current time step.
- The Euler method is a straightforward application of the forward-difference scheme for approximating derivatives.
  - For problems that cannot be handled with Euler, Runge-Kutta is a powerful "workhorse".
- Explore these methods with Project 1: Simple Harmonic Motion.