Differential Equations : Oscillators

モデル化とシミュレーション特論
2023 年度前期
佐賀大学理工学研究科　只木進一

Differential equations and their solutions

- First order differential Equations
  t : independent variable, $\vec{y}$: dependent variables

$$\frac{\mathrm{d}}{\mathrm{d}t}\vec{y} = \vec{f}(t, \vec{y}) \tag{1.1}$$

- Solving (integrating) differential equations: finding solutions of Eq. (1.1)
  - Needs initial conditions.
  - Analytical solutions: finding functions $\vec{y}(t)$ of t satisfying Eq. (1.1).
  - What do numerical solutions mean?

Numerical methods for Differential Equations

- Returning to the definition of differentiation.

$$\frac{\mathrm{d}}{\mathrm{d}t}\vec{y} = \lim_{h \to 0} \frac{\vec{y}(t+h) - \vec{y}(t)}{h} \tag{1.2}$$

- Euler method: simplest numerical method: advance t with h.

$$\vec{y}(t+h) = \vec{y}(t) + h\vec{y'}(t) \tag{1.3}$$

Example 1.1: Motions under a constant force

- Second order differential equation for a falling object

$$\frac{\mathrm{d}^2 y}{\mathrm{d}t^2} = -g \tag{1.4}$$

- Converting to a set of first order differential equations

$$\frac{\mathrm{d}y}{\mathrm{d}t} = v \tag{1.5}$$

$$\frac{\mathrm{d}v}{\mathrm{d}t} = -g \tag{1.6}$$

Some initial steps

- Initial values: $y(0)$ and $v(0)$
- $t = h$

$$y(h) = y(0) + hv(0), \quad v(h) = v(0) - hg$$

- $t = 2h$

$$y(2h) = y(h) + hv(h) = y(0) + 2hv(0) - h^2 g$$
$$v(2h) = v(h) - hg = v(0) - 2hg$$

Matrix formulation

$$\begin{pmatrix} y(h) \\ v(h) \\ g \end{pmatrix} = \begin{pmatrix} 1 & h & 0 \\ 0 & 1 & -h \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} y(0) \\ v(0) \\ g \end{pmatrix}$$

$$\begin{aligned} \begin{pmatrix} y(2h) \\ v(2h) \\ g \end{pmatrix} &= \begin{pmatrix} 1 & h & 0 \\ 0 & 1 & -h \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} y(h) \\ v(h) \\ g \end{pmatrix} = \begin{pmatrix} 1 & h & 0 \\ 0 & 1 & -h \\ 0 & 0 & 1 \end{pmatrix}^2 \begin{pmatrix} y(0) \\ v(0) \\ g \end{pmatrix} \\ &= \begin{pmatrix} 1 & 2h & -h^2 \\ 0 & 1 & -2h \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} y(0) \\ v(0) \\ g \end{pmatrix} \end{aligned}$$

At n-th step

$$
\begin{aligned}
\begin{pmatrix} y(nh) \\ v(nh) \\ g \end{pmatrix} &= \begin{pmatrix} 1 & h & 0 \\ 0 & 1 & -h \\ 0 & 0 & 1 \end{pmatrix}^n \begin{pmatrix} y(0) \\ v(0) \\ g \end{pmatrix} \\
&= \begin{pmatrix} 1 & nh & -\frac{n(n-1)}{2}h^2 \\ 0 & 1 & -nh \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} y(0) \\ v(0) \\ g \end{pmatrix} \qquad (1.7)
\end{aligned}
$$

Solutions

- General terms

$$y(nh) = y(0) + nhv(0) - \frac{n(n-1)}{2}h^2 g \tag{1.8}$$

$$v(nh) = v(0) - nhg \tag{1.9}$$

- Taking limits $h \to 0$ with fixed $t = nh$

$$y(t) = y(0) + tv(0) - \frac{t^2}{2}g \tag{1.10}$$

$$v(t) = v(0) - tg \tag{1.11}$$

- Obtaining the well-known analytical solution.

Numerical solutions of differential equations

- Given initial values $\vec{y}(0)$
- Obtaining sequences $\vec{y}(nh)$ numerically

Runge-Kutta method

$$\vec{k}_1 = h\vec{f}\,(t_n, \vec{y}_n)$$
$$\vec{k}_2 = h\vec{f}\left(t_n + \frac{h}{2}, \vec{y}_n + \frac{\vec{k}_1}{2}\right)$$
$$\vec{k}_3 = h\vec{f}\left(t_n + \frac{h}{2}, \vec{y}_n + \frac{\vec{k}_2}{2}\right)$$
$$\vec{k}_4 = h\vec{f}\left(t_n + h, \vec{y}_n + \vec{k}_3\right)$$

$$\vec{y}_{n+1} = \vec{y}_n + \frac{1}{6}\left(\vec{k}_1 + 2\vec{k}_2 + 2\vec{k}_3 + \vec{k}_4\right) + O\left(h^5\right) \qquad (1.12)$$

Correct up to $O(h^4)$

Differential Equations with Java

- Runge-Kutta method
  Obtain values of dependent variables $\vec{y}(t+h)$ at $t+h$ from $\vec{y}(t)$ and $\mathrm{d}\vec{y}/\mathrm{d}t = \vec{f}(t, \vec{y})$
- Runge-Kutta method can be described as subprograms
  - Implement as static method which does not affect the properties of the instance.
- Sample programs
  `https://github.com/modeling-and-simulation-mc-saga/DifferentialEquations`

Function as an argument of methods

- Java does not have *pointers*
  - In C, functions are passed as pointers to other functions.
- In Java, functions can be passed to methods as an instance of interface.
- However, how to create an instance of interface
  - Instances of an interface can not be created because interfaces do not have implementations of methods.

Instances of Interface

- Observe the case of `DoubleFunction` interface.
- Anonymous class : Implements method `apply()` at the construction.

```
DoubleFunction<Double> function
    = new DoubleFunction<Double>(){
        @Override
        public Double apply(double v){
            return v*v;
        }
};
```

- Lambda expression

```
DoubleFunction<Double> function = x -> x * x;
```

`rungeKutta` package

- `DifferentialEquation.java`
  - An interface
  - Describe right hand side of differential equations
- `RungeKutta.java`
  - Implement Runge-Kutta method
  - Advance time with h
  - Advance time with given n steps with h
- `State.java`
  - Record class for keeping state
    - independent variable x
    - dependent variables $\vec{y}$
  - Record class: simple data carrier

DifferentialEquation interface

```
@FunctionalInterface
public interface DifferentialEquation {
    /**
     *
     * @param t independent variable
     * @param y dependent variables
     * @return RHSs of differential equations
     */
    public double[] rhs(double t, double y[]);
}
```

State class implemented as Record class

```
public record State(double x, double[] y) {};
```

⇕

```
public final class State{
    private final double x;
    private final double[] y;

    public State(double x, double[] y){
        this.x = x;
        this.y = y;
    }

    public double x() {return this.x;}
    public double[] y() {return this.y;}

    public boolean equals(){...}
    public int hashCode(){...}
    public String toString(){...}
}
```

Harmonic Oscillators

- Harmonic Oscillators

$$m\frac{\mathrm{d}^2x}{\mathrm{d}t^2} = -kx \tag{3.1}$$

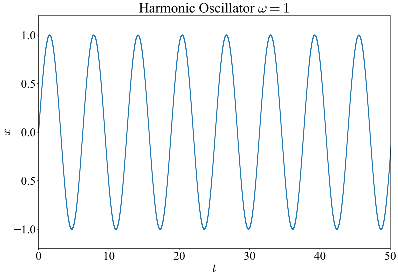
$$x(t) = A\cos(\omega t + \delta), \quad \omega^2 = \frac{k}{m} \tag{3.2}$$

- In a form of first-order differential equations

$$\frac{\mathrm{d}x}{\mathrm{d}t} = v \tag{3.3}$$

$$\frac{\mathrm{d}v}{\mathrm{d}t} = -\frac{k}{m}x \tag{3.4}$$

```java
public HarmonicOscillator(double x, double v, double k) {
    super(x,v);
    //differential equation
    equation = (double xx, double[] yy) -> {
        double dy[] = new double[2];
        dy[0] = yy[1];// dx/dt = v
        dy[1] = -k * yy[0];// dv/dt = - (k/m) x
        return dy;
    };
}
```



Periodic External Force

- Interesting phenomena such as resonance appear under periodic external forces.

$$\frac{\mathrm{d}^2 x}{\mathrm{d}t^2} = -\omega^2 x + \frac{1}{m} F(t) \tag{4.1}$$

$$F(t) = f \cos\left(\gamma t + \beta\right) \tag{4.2}$$

Homogeneous and Inhomogeneous equations

- Homogeneous equations : equal degree of x in both sides

$$\frac{\mathrm{d}^2 x}{\mathrm{d}t^2} = G(t)x \tag{4.3}$$

- general solutions

$$x(t) = Ax_+(t) + Bx_-(t) \tag{4.4}$$

Homogeneous and Inhomogeneous equations

- Homogeneous equations plus an inhomogeneous term $F(t)$

$$\frac{\mathrm{d}^2 x}{\mathrm{d}t^2} = G(t)x + F(t) \tag{4.5}$$

- special solution $x_0(t)$

$$\frac{\mathrm{d}^2 x_0}{\mathrm{d}t^2} = G(t)x_0 + F(t) \tag{4.6}$$

- General solutions for inhomogeneous equations

$$x(t) = Ax_+(t) + Bx_-(t) + x_0(t) \tag{4.7}$$

Special Solutions

$$\frac{\mathrm{d}^2 x}{\mathrm{d}t^2} = -\omega^2 x + \frac{1}{m} F(t) \tag{4.8}$$

$$F(t) = f \cos(\gamma t + \beta) \tag{4.9}$$

- Assume a form of the special solution as $x_0(t) = B\cos(\gamma t + \beta)$

$$-\gamma^2 B \cos(\gamma t + \beta) = -\omega^2 B \cos(\gamma t + \beta) + \frac{f}{m}\cos(\gamma t + \beta) \tag{4.10}$$

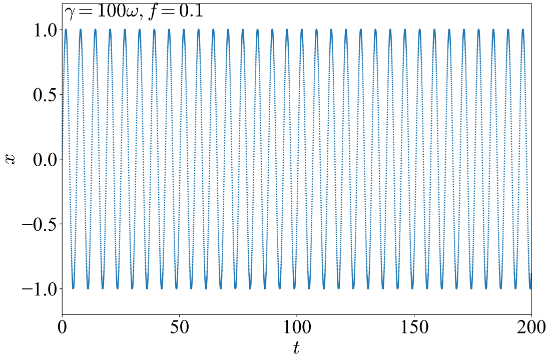
$$x_0(t) = \frac{f}{m(\omega^2 - \gamma^2)} \cos(\gamma t + \beta) \tag{4.11}$$

General Solutions

- general solutions for homogeneous equation plus special solution

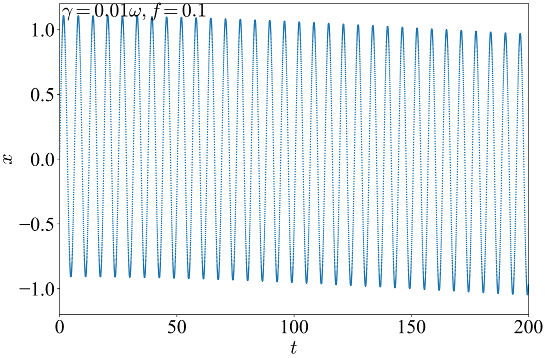
$$x(t) = A\cos(\omega t + \delta) + \frac{f}{m(\omega^2 - \gamma^2)}\cos(\gamma t + \beta) \tag{4.12}$$

Fast External Force : $\gamma \gg \omega$



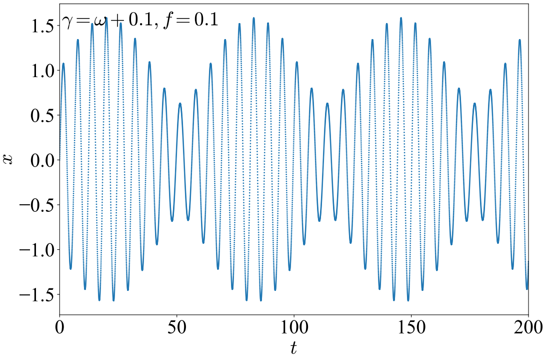
- the external force changes faster than one period of the oscillator
- the external force changes too fast to affect the oscillator

Slow External Force : $\gamma \ll \omega$

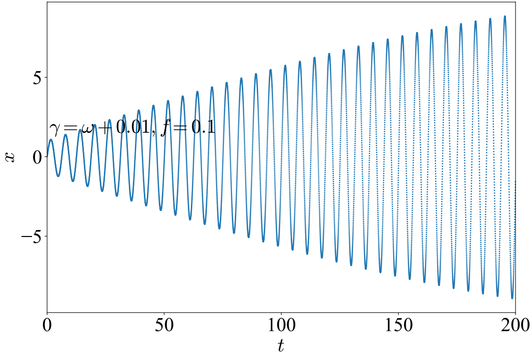


- Oscillation under slow external force are almost not affected.
- External force only affects the outline.

External Force with Close Frequency: resonance



External Force with Very Close Frequency: howling



- howling : the amplitude grows linearly with time

Approximated solutions

- $\gamma = \omega + \epsilon$, $\epsilon \ll 1$

$$\cos(\gamma t + \beta) - \cos(\omega t + \beta) = -t\epsilon \sin(\omega t + \beta) + O\left(\epsilon^2\right) \tag{4.13}$$

$$\frac{1}{\omega^2 - \gamma^2} = -\frac{1}{2\omega}\left(1 + O(\epsilon)\right) \tag{4.14}$$

- howling

$$x(t) = A' \cos(\omega t + \alpha') + t\frac{f}{2m\omega} \sin(\omega t + \beta) + O(\epsilon) \tag{4.15}$$

Defining equation

```java
public HarmonicOscillatorWithExternalForce(
    double x, double v, double k,
    DoubleFunction<Double> exForce) {
    super(x, v);
    equation = (double t, double[] yy) -> {
        double dy[] = new double[2];
        dy[0] = yy[1];// dx/dt = v
        // dv/dt = - (k/m) x + exF(t)
        dy[1] = -k * yy[0] + exForce.apply(t);
        return dy;
    };
}
```

```java
//defining the external force with Lambda expression
DoubleFunction<Double> exForce
        = t -> f * Math.cos(gamma * t + beta);
```