Differential Equations : Interacting Oscillators

モデル化とシミュレーション特論
2021 年度前期
佐賀大学理工学研究科　只木進一

Examples: Synchronization

- fire flies
  https://www.youtube.com/watch?v=WMIXp8H8364
- metronomes
  https://www.youtube.com/watch?v=JWToUATLGzs
- pendulum clocks
  Found occasionally by Christiaan Huygens in 1665.

Sample programs

Folders `CoupledOscillators2` and `kuramoto` in the downloaded files for the last week.

Interacting Harmonic Oscillators

- n oscillators with slightly different natural frequencies.
- Interactions for decreasing the difference between oscillators.
- Interactions are symmetric.

$$m_i \frac{\mathrm{d}^2 x_i}{\mathrm{d}t^2} = -k_i x_i - \sum_j \lambda_{ij} (x_i - x_j) \tag{1}$$

$$\lambda_{ij} = \lambda_{ji} > 0 \tag{2}$$

Description in CoupledOscillators2.java

```java
equation = (double tt, double yy[]) -> {
    double dy[] = new double[2 * numOscillators];
    for (int i = 0; i < numOscillators; i++) {
        int j = 2 * i;
        dy[j] = yy[j + 1];
        double dyy = -(k[i] / m[i]) * yy[j];
        for (int kk = 0; kk < numOscillators; kk++) {
            dyy -= (lambda[i][kk] / m[i]) * (yy[j] - yy[2*kk]);
        }
        dy[j + 1] = dyy;
    }
    return dy;
};
```

Energy

- potential energy

$$U = \frac{1}{2} \sum_i k_i x_i^2 + \frac{1}{2} \sum_i \sum_j \lambda_{ij} \left(x_i - x_j \right)^2 \tag{3}$$

- kinetic energy

$$T = \frac{1}{2} \sum_i m_i \left(\frac{\mathrm{d}x_i}{\mathrm{d}t} \right)^2 \tag{4}$$

Energy Conservation

Total energy

$$E = U + T \tag{5}$$

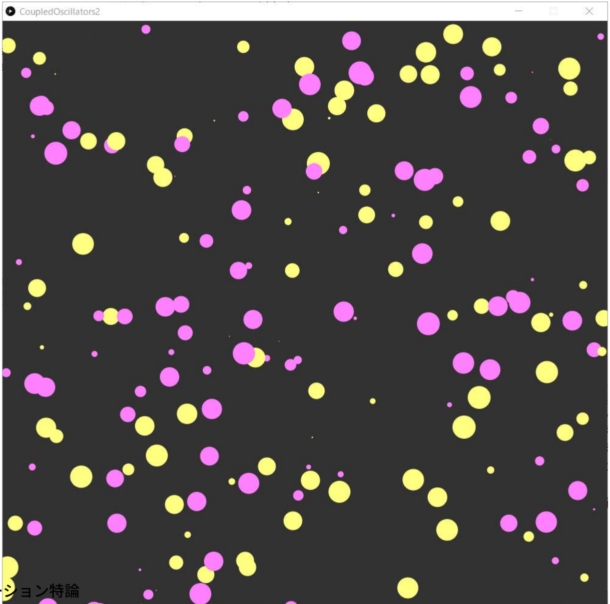
Temporal derivative of the potential and kinetic energy.

$$\frac{\mathrm{d}T}{\mathrm{d}t} = \sum_i m_i \left(\frac{\mathrm{d}x_i}{\mathrm{d}t}\right)\left(\frac{\mathrm{d}^2 x_i}{\mathrm{d}t^2}\right) \tag{6}$$

$$\frac{\mathrm{d}U}{\mathrm{d}t} = \sum_i k_i x_i \left(\frac{\mathrm{d}x_i}{\mathrm{d}t}\right) + \sum_i \sum_j \lambda_{ij} \left(\frac{\mathrm{d}x_i}{\mathrm{d}t}\right)(x_i - x_j) \tag{7}$$

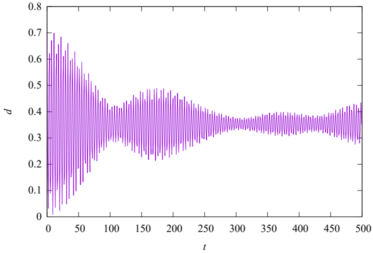
$$\frac{\mathrm{d}E}{\mathrm{d}t} = \sum_i \left(\frac{\mathrm{d}x_i}{\mathrm{d}t}\right)\left[m_i \left(\frac{\mathrm{d}^2 x_i}{\mathrm{d}t^2}\right) + k_i x_i + \sum_j \lambda_{ij}(x_i - x_j)\right] = 0 \tag{8}$$

Observation



Order Parameter

$$d = \frac{2}{n(n-1)} \sum_{i} \sum_{j \neq i} (x_i - x_j)^2 \tag{9}$$



d tends to some fixed values.

Description in CoupledOscillators2.java

```java
public List<Point2D.Double> doObserve(int tmax, double h) {
    List<Point2D.Double> plist = Utils.createList();
    for (int t = 0; t < tmax; t++) {
        Oscillator o[] = sys.update(h);
        double d = 0.;
        for (int i = 0; i < n - 1; i++) {
            for (int j = i + 1; j < n; j++) {
                d += Math.pow(o[i].y - o[j].y, 2.);
            }
        }
        d *= 2. / n / (n - 1);
        plist.add(new Point2D.Double(t * h, d));
    }
    return plist;
}
```

Kuramoto Model

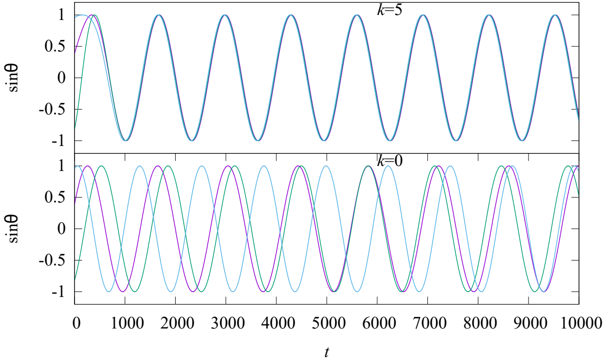
- Fundamental model for synchronization.
- N oscillators interact through their phase differences.

$$\frac{\mathrm{d}\theta_i}{\mathrm{d}t} = \omega_i + \frac{k}{N} \sum_j \sin\left(\theta_j - \theta_i\right) \tag{10}$$

Description in Kuramoto.java

```
equation = (double tt, double yy[]) -> {
    double dy[] = new double[n];
    for (int i = 0; i < n; i++) {
        dy[i] = omega[i];
        for (int j = 0; j < n; j++) {
            dy[i] += (k / n) * Math.sin(yy[j] - yy[i]);
        }
    }
    return dy;
};
```

Three oscillators



- Not Synchronize with $k = 0$
- Synchronize with $k = 5$

Order Parameter

$$R = \frac{1}{N}\sum_i e^{i\theta_i} \tag{11}$$