Simulation, Analysis, and Validation of Computational Models

— 5. Numerics of Non-Linear Systems —

Lecturer: Michael Herrmann
School of Informatics, University of Edinburgh
michael.herrmann@ed.ac.uk, +44 131 6 517177

Overview

- Numerical integration
- Collision
- Percolation

Recap: Chaos

- Asymptotic behavior of a nonlinear system, when (stable) fixed points, limit cycles and tori can be excluded in a deterministic, simple dynamical system.
- Sensitivity to initial conditions implies noise-like features, but short-term predictions are possible.
- Contains typically many unstable periodic trajectories.
- Occurs when the period of the trajectory of the system's state diverges to infinity.
- Occurs everywhere in nature, and in social and engineered systems.
- Mild chaos can add a bit of realism to a simulation.

H. Sayama (2015) Introduction to the Modeling and Analysis of Complex Systems. SUNY.

Last time: Nonlinear systems

- Non-linear dynamics is predominant in all aspects nature, in society, and in complex technical systems.
- A major challenge in many fields of application of the theory is the identification of imminent qualitative changes ("tipping points").
- While linear systems are well understood, many problem in non-linear dynamics remain to be solved, to be formulated or to be discovered.
- Numerical simulation of non-linear systems are still improvable.

Numerical integration: Simulating non-linear dynamics

- Goal: If there is no analytical solution, we need to find a numerical solution $\hat{x}(t)$ for $t \geq t_0$ for an ordinary differential equation $\dot{x} = f(x)$, given $x(t_0) = \hat{x}_0(0) = x_0$
- An algorithm that calculates $\hat{x}$ can be evaluated by the
  - integral
    $$\int_{t_0}^{t_1} (x(t) - \hat{x}(t))^2 \, dt$$
  - maximum of the difference $|x - \hat{x}(t)|$
  - maximal order of a polynomial that is perfectly be integrated.
- General idea: Find $\dot{x}$ at some points across the interval $[t, t + \Delta t]$ using the given ODE and Taylor expansion. Resulting values are weighted and added to get an estimate for $t + \Delta t$.
- Integration formulas depend on the specific method and differ in number and weights and order of points along time axis.

Numerical integration

Euler method is the simplest integration formula

$$x(t+\Delta t) = x(t) + \dot{x}(t)\,\Delta t \text{ where } \dot{x}(t) = f(x(t))$$

Perfect only for constant functions, but can be sufficient in other cases (for small Δt),
Errors can be relatively large especially for exponential dynamics.

Example: $\dot{x}(t) = x(t)$ starting at $x(0) = 1$ using $\Delta t = 1$

$x(1) = x(0) + x(0) \cdot 1 = 2$

$x(2) = x(1) + x(1) \cdot 1 = 2 + 2 \cdot 1 = 4$

$x(3) = x(2) + x(2) \cdot 1 = 4 + 4 \cdot 1 = 8 \Rightarrow x(n) = 2^n$

At $n = 4$ Euler: 16, exp(4) =54.60: abs. error of 38.60

$\Delta t = 1$ is generally too large, but reducing step width to 0.1 still gives an abs. error of 9.34

Many better methods available.

Numerical integration: Predictor–corrector method

Result from a simple formula can be corrected by another method.

Combinations of any methods are possible, but for simplicity we start again with the Euler method for the Predictor step:

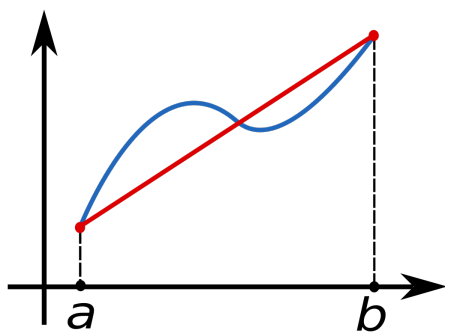
$$\tilde{x}(t+\Delta t) = x(t) + \dot{x}(t)\,\Delta t$$

Corrector step to interpolate the initial attempt with new result from the predictor using the differential equation $\dot{\tilde{x}} = f(\tilde{x}(t+\Delta t))$

$$\hat{x}(t+\Delta t) = x(t) + \tfrac{1}{2}\Delta t\,(\dot{x}(t) + f(\tilde{x}(t+\Delta t)))$$

which is called trapezoidal rule.

Above example: $\tilde{x}(1) = 2$, $f(\tilde{x}) = 2 \Rightarrow$
$\hat{x}(1) = 1 + \frac{1}{2}\cdot 1\cdot(1+2) = 2.5$ $(\Delta t = 1)$
(compare to $e = 2.71828$).



Numerical integration: Various classifications

- Explicit methods determine the state for a later time from the state of the system at present
- Implicit methods determine the solution by solving an equation involving the current state of the system and later ones.

- Direct methods: Use intermediate values to calculate next step
- Multi-step method: Calculate new values based on several previous values (needs to get started by simple method)

Numerical integration: Explicit methods

- E.g. RK4 (4th order Runge-Kutta method, explicit)

$$x(t+\Delta t) = x(t) + \frac{\Delta t}{6}\left(\underbrace{k_1 + 2k_2 + 2k_3 + k_4}_{\text{mixture of derivatives}}\right),$$

$$k_1 = f(x(t)),$$

$$k_2 = f\left(x\left(t+\frac{\Delta t}{2}\right) + \Delta t\frac{k_1}{2}\right),$$

$$k_3 = f\left(x\left(t+\frac{\Delta t}{2}\right) + \Delta t\frac{k_2}{2}\right),$$

$$k_4 = f\left(x(t+\Delta t) + \Delta t\, k_3\right).$$

- 4th order: The error of the result scales with $(\Delta t)^4 \times$ 4th derivative of the function f. I.e. it solves ODEs with polynomial f of order not greater than 3 precisely.

Numerical integration: Implicit methods

- For highly non-linear ODEs, implicit methods may work better, i.e. moving forwards and backwards in time (like predictor corrector). Example: Adams-Moulton method.

$$\begin{aligned} x(t+\Delta t) = x(t) + \tfrac{h}{720} \; & (251 f(x(t+\Delta t)) + 646 f(x(t)) \\ & -264 f(x(t-\Delta t)) + 106 f(x(t-2\Delta t)) \\ & -19 f(x(t-3\Delta t))) \end{aligned}$$

- Need to get started (multi-step method), by calculating the first steps by more simple methods
- Solves ODEs with polynomial f of order not greater than 4 precisely (order 5).

Numerical integration: Points to remember

- Because analytical solutions are mostly unavailable, numerical integration is generally used.
- Trade-off between complexity of the integration formula and a small step width: Simply reducing step widths can give an idea that the system is problematic, but may be inefficient for solution.
  - Complex integration methods are usually available in numerical modelling
  - Practically, 4th order Runge-Kutta (RK4) is often sufficient.
  - Adaptive methods, step width control (not discussed here)
  - For complex or chaotic systems, implicit methods should be considered.
- Check: Is the energy conserved also in the simulated system?

Collisions

- Collisions are essential in particle dynamics
- Should be avoided in robotics (i.e. occur at zero speed), but needed in robot simulator
- Assumptions: Elastic, point contact, no spin, no friction, no complex shapes, no deformation, no internal degrees of freedom.
- Trajectory at the collision is not smooth, because of instantaneous effect is assumed.
- Need to stop integration and handle collision separately.

Collisions in 1D

Two masses m_1 and m_2 on a line collide with velocities v_1 and v_2 and depart with velocities v_1' and v_2' (masses remain unchanged)

For an elastic collision, and total momentum

$$m_1 v_1 + m_2 v_2 = m_1 v_1' + m_2 v_2'$$

and total energy

$$\frac{1}{2} m_1 {v_1}^2 + \frac{1}{2} m_2 {v_2}^2 = \frac{1}{2} m_1 {v_1'}^2 + \frac{1}{2} m_2 {v_2'}^2$$

are conserved, although momentum and energy can be transferred from one object to the other one

$$\begin{aligned} v_1' &= \frac{m_1 v_1 + m_2 v_2}{m_1 + m_2} + \frac{m_2}{m_1 + m_2}(v_2 - v_1) = v_1 + \frac{2m_2}{m_1 + m_2}(v_2 - v_1) \\ v_2' &= \frac{m_1 v_1 + m_2 v_2}{m_1 + m_2} + \frac{m_1}{m_1 + m_2}(v_1 - v_2) = v_2 + \frac{2m_1}{m_1 + m_2}(v_1 - v_2) \end{aligned}$$

I.e. weighted mean velocity + proportional slowdown or speedup

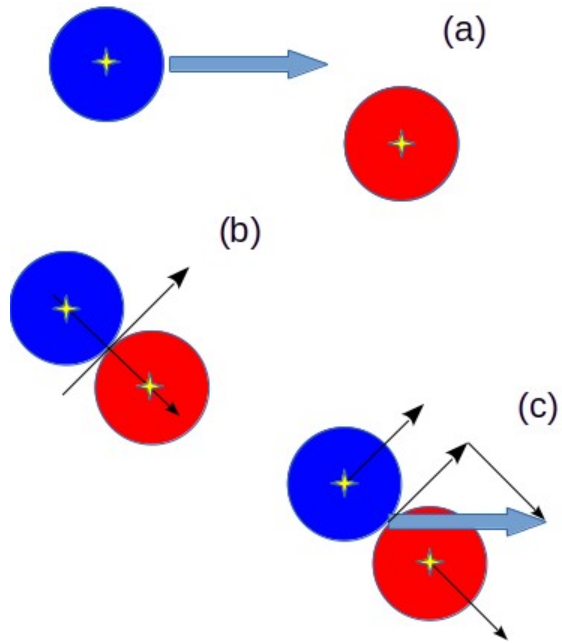
(JFYI: https://www.youtube.com/watch?v=HEfHFsfGXjs)

Collision in 2D

(a) Relative motion v_1 of a moving ball against resting one $v_2 = 0$

(b) Contact between the two convex objects defines coordinate system

(c) After collision, original velocity splits according to contact coordinates



New v_2' will be in direction from x_1 to x_2 (at collision): $x_2 - x_1$.

Project v_1 on unit vector $\frac{x_2-x_1}{\|x_2-x_1\|}$: $\left\langle v_1, \frac{x_2-x_1}{\|x_2-x_1\|} \right\rangle \frac{x_2-x_1}{\|x_2-x_1\|}$ for $v_2 = 0$

or $\left\langle v_1 - v_2, \frac{x_2-x_1}{\|x_2-x_1\|} \right\rangle \frac{x_2-x_1}{\|x_2-x_1\|}$ for any v_2

Collision in 2D

v_1' direction is obtained from the vector orthogonal to $(x_2 - x_1)$.

$$\tilde{v}_1 \propto \frac{\left\langle v_1 - v_2, (x_2 - x_1)^\perp \right\rangle}{\|x_2 - x_1\|^2} (x_2 - x_1)^\perp \qquad \begin{pmatrix} a \\ b \end{pmatrix}^\perp = \begin{pmatrix} -b \\ a \end{pmatrix}$$

$$\tilde{v}_2 \propto \frac{\langle v_1 - v_2, x_2 - x_1 \rangle}{\|x_2 - x_1\|^2} (x_2 - x_1)$$

The respective lengths are found as in the 1D case (see above)

$$v_1' = v_1 + \frac{2m_2}{m_1 + m_2} (v_2 - v_1)$$

$$v_2' = v_2 + \frac{2m_1}{m_1 + m_2} (v_1 - v_2)$$

So that we arrive at

$$v_1' = v_1 + \frac{2m_2}{m_1 + m_2} \frac{\langle v_2 - v_1, (x_2 - x_1)^\perp \rangle}{\|x_2 - x_1\|^2} (x_2 - x_1)^\perp$$

$$v_2' = v_2 + \frac{2m_1}{m_1 + m_2} \frac{\langle v_1 - v_2, x_2 - x_1 \rangle}{\|x_2 - x_1\|^2} (x_2 - x_1)$$

Check conservation of momentum and energy!

Simple Collision Detection

- Sort objects lexicographically by coordinate
- Check first coordinate (then second coordinate etc.) for any close objects (or hierarchically for object parts) to fill active list
- Calculate any position increments $\dot{x}$ (see previous lecture) and check active list for penetration
- Revert increment of relevant object parts and invoke collision handler
- Calculate forces to other parts of the same object

Collisions: Points to remember

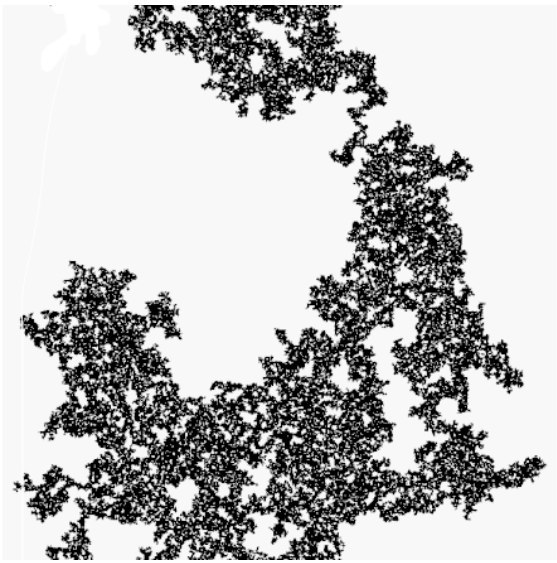
- In a collision the (possibly) linear microscopic dynamics is abstracted to a nonlinear macroscopic effect
- Using heuristics and simple elementary shapes can be useful
- Realistic collision of deformable material is still subject of research
- More than two objects: Chaining, jamming, percolation

Phase transitions

- Last lecture: Non-linear equations describing a qualitative change of a macroscopic system
- In phase transitions, order-parameters are used to describe typical phenomena
- *Universality*: Order parameter effects implied by typical equations (such as $\dot{x} = cx - x^3$)
- Percolation as a simple example of a phase transition in microscopic description

Percolation

- Classical example: Mix an in increasing fraction of iron particle with sand, and check a whether the mixture conducts electricity
- Mathematical problem: Formation of a giant component



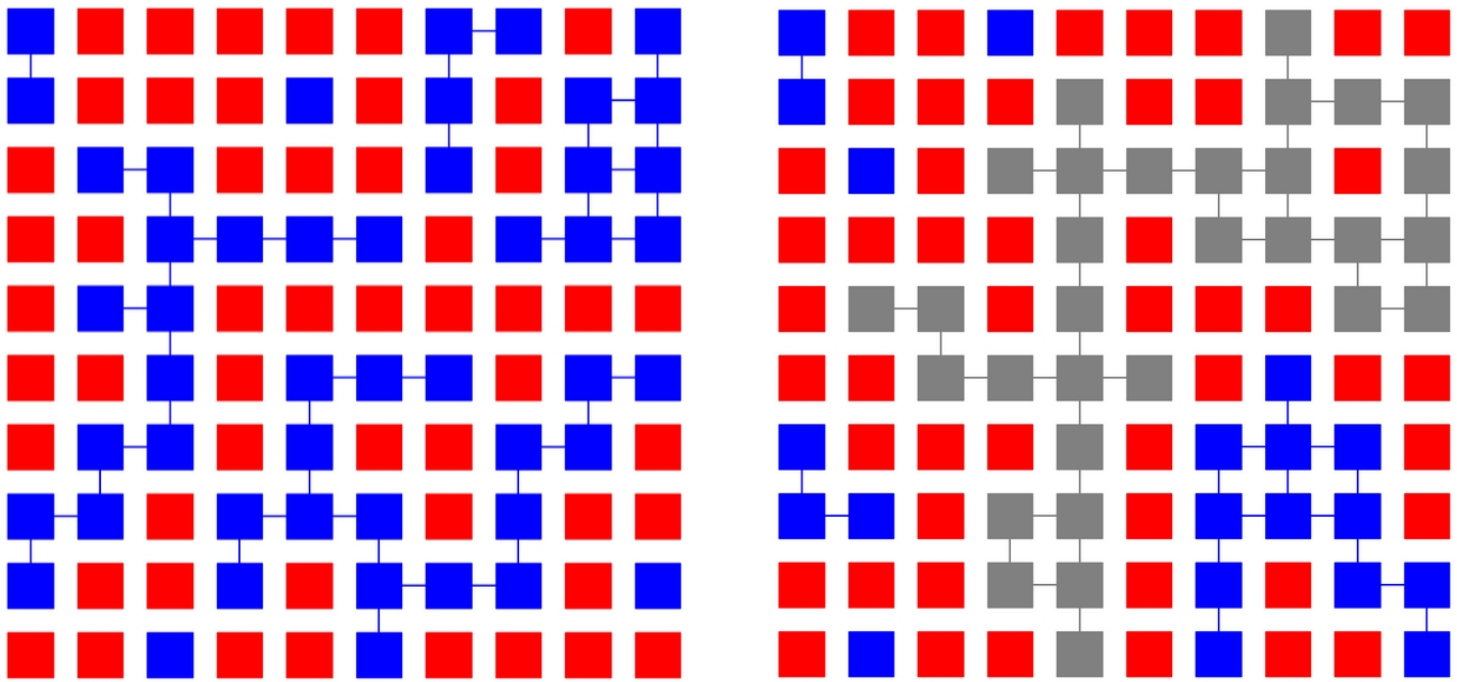
Other examples: Transport in porous media, underground CO_2 storage, drip-through coffee making, groundwater recharge, wear and tear, forest fires, neural avalanches, spread of diseases (later)

Modelling

- $N \times N$ grid of cells
- Choose randomly whether a cell is conductive (probability p) or not (probability $1 - p$)
- Use flooding algorithm to identify conductive cluster: All conductive cells with with a conductive cell next to them
- Repeat many times to determine probability of conductivity of the material as a whole
  - practically zero for small p
  - steeply increasing for a range of p
  - practically one for large p
- Steepness of increase increases with N and becomes infinite for $N \to \infty$: Unique value p_{crit}
- Quantitative: Throughput increases for $p > p_{\text{crit}}$, similar to pitchfork bifurction (see last lecture)

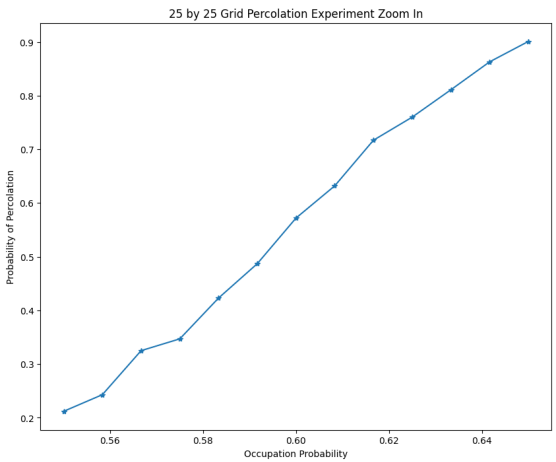
Percolation near critical value

Percolation in a 10×10 square grid (top to bottom) near critical occupation probability $p = 0.492$

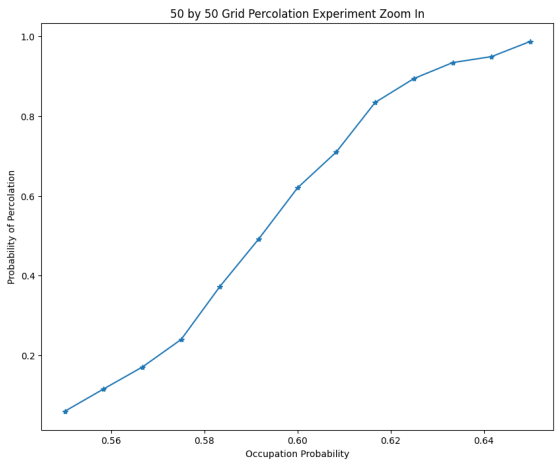


Scaling of the critical range

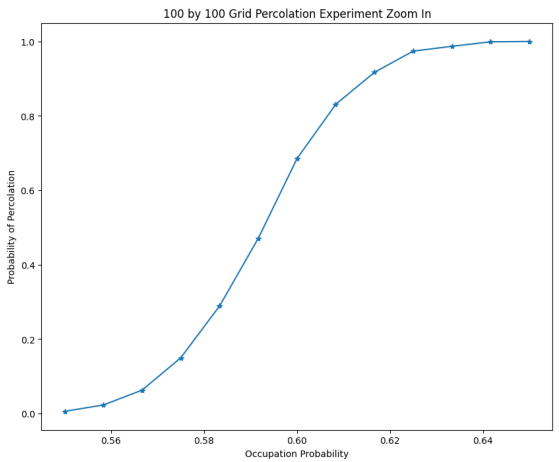
Percolation probability in an $N \times N$ square grid over occupation probability



25



50



100

Conclusion on percolation

- Practically interesting with a many applications
- Shows a common type of phase transition
- Distribution of the size of clusters at criticality decays as a power law with the same exponent for 2D lattices
- Percolation clusters are self-similar (*fractals*)
- When modelling disease spreading, percolation with be studied on a graph

Next topics

- Noise
- System modelling
- Simulations