Introduction to Computational Physics

Pascal Debus *
Thomas Gersdorf†

ETH Zurich, Fall 2014

Abstract

This is a short summary of the lecture *Introduction to Computational Physics* given by Professor Hans J. Herrmann at ETH Zurich in fall 2013. It is strongly focuses around the expected exam questions and hence not complete.

1 General

1.1 Relevant questions

- Congruential and lagged-Fibonacci RN
- Definition of percolation
- Fractal dimension and sand-box method
- Hoshen-Kopelman algorithm
- Finite size scaling
- Integration with Monte-Carlo
- Detailed Balance and MR^{2}T^2
- Ising model
- Simulate random walk
- Euler method
- 2nd order Runge-Kutta
- 2nd order predictor-corrector
- Jacobi and Gauss-Seidel relaxation
- Gradient methods
- Strategy of finite elements, finite volume and spectral methods

2 Random number generators

Random numbers:

- Sequence of numbers in random or uncorrelated order
- Probability for occurrence of a number is always the same
- Computers are completely deterministic
- Use a deterministic algorithm such that numbers are
  - almost homogeneously
  - randomly distributed
- → Computers generate pseudo-random-numbers

2.1 Congruential (multiplicative) generators

Create a sequence x_i with two integer numbers c, p and **seed** x_0:

$$\boxed{x_i = (cx_{i-1}) \mod p} \tag{1}$$

creates random numbers in the interval $[0, p-1]$
Divide by p to map to $[0, 1[$ and obtain **normalized** pseudo random numbers:

$$\boxed{0 \leq z_i = \frac{x_i}{p} < 1} \quad z_i \in \mathbb{Q}$$

All integers are smaller than p, hence, the sequence must repeat atfer at least $(p-1)$ iterations, which is the **maximal period** of this RNG.

Carmichael: Maximal period is $p-1$, can be obtained if p is a Mersenne prime number $M = 2^{\text{prime}} - 1$ and smallest number with $c^{p-1} \mod p = 1$.
Park/Miller: $p = 2^{31} - 1 = 21474834647$, $c = 16807$ for 32-bit integers.

Testing of RNGs: plot in 2D or 3D to show correlations between two(x_i, x_{i+1}) or three(x_i, x_{i+1}, x_{i+2}) consecutive random numbers.

Marsaglia-Theorem: For a congruential RNG the random numbers in an n-cube test lie on parallel $(n-1)$-dimensional hyperplanes.

cRNG are in general very fast but do not produce good random numbers.

*pdebus@student.ethz.ch
†thomas@gersdorf.tel

2.2 Lagged Fibonacci (additive) generators

- permit very large periods
- allow for adventageous predcitions about correlations

Consider a sequence $x_i \in \{0,1\}$ of binary numbers, $1 \leq i \leq b$:

$$\boxed{x_{b+1} = \left(\sum_{j \in \mathcal{J}} x_{b+1-j} \right) \mod 2, \quad \mathcal{J} \subset \{1, \ldots, b\}}$$

The generator performs a certain operation on the previous numbers.
Example: Random number is result of some binary operation of two previous numbers (two element lagged Fibonacci generator):

$$x_i = x_{i-c} \oplus x_{i-d} := (x_{i-c} + x_{i-d}) \mod 2 \tag{2}$$

Note: Initial sequence is necessary, for $d \leq c$ at least c bits long.

Zierler-Trinomial condition for c and d: If

$$T_{c,d}(r) = 1 + z^c + z^d \tag{3}$$

cannot be factorized in subpolynomials, then maximal period $2^c - 1$, where z is binary and c ususally chosen up to 10000.
This statement is part of a theorem of **Compagner**, stating also:

$$\langle x_i x_{i-k} - \langle x_i \rangle^2 \rangle = 0$$

The smallest numbers satisfying the above relation are $(c,d) = (250, 103)$(Kirkpatrick and Stoll, 1981).

Conversion from binary number to natural number e.g. uint32

- 32 LFRNG paralell (very efficient) BUT also 32 initial sequences needed which must be uncorrelated each by itself and among each other.
- Extract 32 bit long part of a sequence: Slow and strong correlations.

Lagged Fibonacci RNGs are much slower than cRNGs but produce good random number sequences.

2.3 Testing of RNGs

Most important tests:

- n-dimensional **cube test** (square in 2D, cube in 3D) should show no correlations between n consecutive random numbers, should be homogeneous
- average value $\bar{s} = \lim_{N \to \infty} \frac{1}{N} \sum_i s_i \overset{!}{=} \frac{1}{2}$
- fluctions of mean value (χ^2) test: distribution around mean value should be Gaussian-like (large k limit of chi-squared-distribution $N(k, 2k)$)

$$\chi^2 = \sum_{i=1}^{k} \frac{(N_i - np_i)^2}{np_i} \tag{4}$$

- spectral test: FFT, Fourier spectrum should correspond to white noise (no peaks, no correlations)
- no correlations $\langle s_i * s_{i+d} \rangle - \langle s_i^2 \rangle \overset{!}{=} 0$
- Marsaglia's Diehard test battery

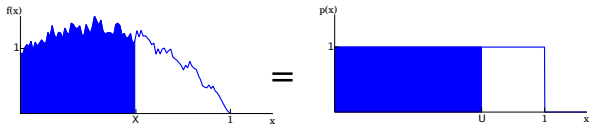
2.4 Non-uniform distributions

Congruential and lagged Fibonacci produce numbers in $\mathbb{N}$ following a uniform distribution.

$$P_u(z) = \begin{cases} 1, & \text{if } z \in [0,1) \\ 0, & \text{else} \end{cases}$$

Assume z is uniformly distributed, and y according to the desired distribution, then we can obtain the analytic mapping by the condition

$$\int_0^y P(y')\,\mathrm{d}y \overset{!}{=} \int_0^z P_u(z')\,\mathrm{d}z'$$



2.4.1 Analytic mapping

Transform random numbers from uniform distribution to some distribution given by $P(y)$:

$$y = \left[\int_0^y P(y')\,\mathrm{d}y' \right]^{-1} (z) \tag{5}$$

if a closed-form analytical integral expression and its inverse exists.

Examples:

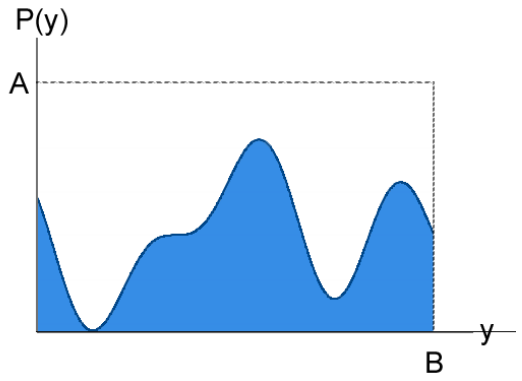
- Poisson $P(y) = ke^{-ky}$ such that $y = -\frac{1}{k}\log(1-z)$
- Gaussian $P(y) = \frac{1}{\sqrt{\pi\sigma}} e^{-\frac{y^2}{\sigma}}$
  by **Box-Muller-Method**, which transforms two *uncorrelated* uniformly distributed random variables

z_1, z_2 by

$$y_1 = \sqrt{-\sigma \log(1 - z_2)} \sin(2\pi z_1)$$
$$y_2 = \sqrt{-\sigma \log(1 - z_2)} \cos(2\pi z_2)$$

2.4.2 Rejection method

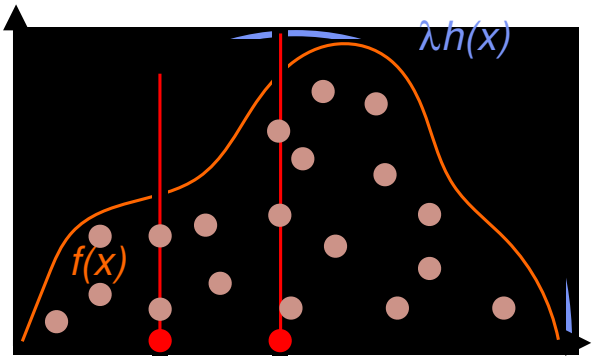
For distributions that cannot be inverted analytically:



Square around distribution with lengths A an B, generate random 2D points (Bz_1, Az_2).
Let $P(y)$ be the distribution of interest with the necessary condition the it is well-behaved:(i.e. finite over the domain of interest):

$$P(y) < A \quad \forall y \in [0, B] \quad A, B \in \mathbb{R}$$

- Generate $(z_1, z_2) \in [0, 1)$ and map to bounding box $\Rightarrow (Bz_1, Az_2)$
- Reject if $\boxed{Az_2 > P(Bz_1)}$ (Point above bounding box)
- Generalization: Use bounding distribution $Q(y)$ with $P(y) < \lambda Q(y)$ instead of box.



3 Percolation

3.1 Concept of percolation

Definition(Material Science and Chemistry):
Percolation describes movement and filtering of fluids through porous media.

Percolation model has some **universal features of critical phenomena**, these features do not depend on the practical model that is used.

Important point in that context: system-spanning cluster (the **percolating cluster**) at the critical point (e.g. polymerization). **Percolation threshold** occurs at some critical occupation probability p_c (or multiple parameter $p_{i,c}$) such that infinite connectivity (**percolation**) is achieved. (Once we have a spanning cluster the percolation transition occurs)

Definition Universality (statistical mechanics): Properties for a large class of systems are independent of dynamical details of the system. Systems display universality in the **scaling limit**, a large number of interacting parts.

Definition of scaling: A function $f(x, y)$ that can be expressed as a function $f(x')/$

3.2 The percolation model

Typical modeling: Occupy some lattice with probability p. Check for connected paths.

3.2.1 Burning method

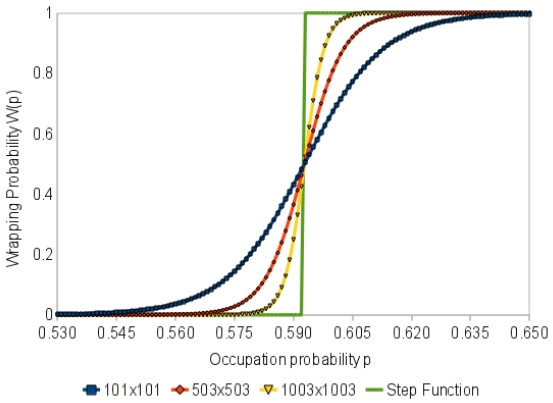
Define square lattice, set one side "on fire" and continue by burning next neighbors.

- Provides boolean feedback for the existence of a cluster
- also calculates the **minimal path length**

1. Label all occupied cells in the top line with the marker $t = 2$
2. Iteration step $t + 1$
   (a) Go through all cells and find the cells which have label t
   (b) For each of the found t-label cells do
      i. Check if any direct neighbor (N,E,S,W) is occupied and not burning (label 1)
      ii. Set found neighbors to label $t + 1$
3. Repeat step 2 (with $t = t + 1$) until either there are no neighbors to burn any more or the bottom line has been reached, then the latest label minus 1 defines the shortest path.

3.2.2 Percolation threshold

Probability to obtain spanning-cluster depends on occupation probability. For a **critical probability** p_c ($= 0.5927$ for 2D site-lattice) percolation is achieved.
p_c actually defined for an infinite cluster.
Finite Case: p_c is interpreted as *average probability* at which the first percolating cluster appears.



The transition from low to high wrapping probability becomes more abrupt with the lattice size: Wrapping probability approaches step function for infinite lattice size.

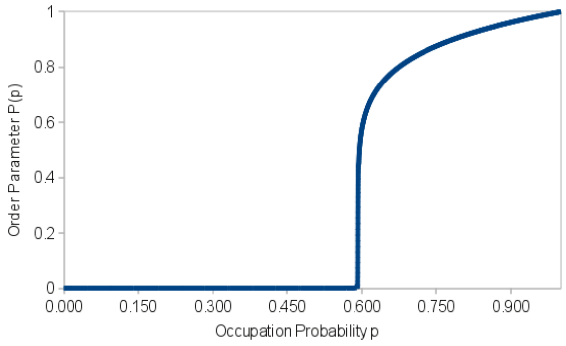
The threshold is a characteristic value for a given **latice type**(e.g. honey comb lattice has the highest 2D probability)

3.2.3 The order parameter

Consider probabilities $p > p_c$.

Order parameter
is **fraction of occupied sites** that belong to the **largest spanning cluster.**

$$P(p) \propto= \begin{cases} (p-p_c)^{\beta} & p > p_c \\ 0 & p < p_c \end{cases} \tag{6}$$



$\beta = \frac{5}{36}(2D), \approx 0.41(3D)$

- **Power law** behavior close to percolation threshold
- dimension-dependent exponent β

(universal criticality)
Example:

- phase transition
- magnetization
- percolation

3.3 Hoshen-Kopelman algorithm

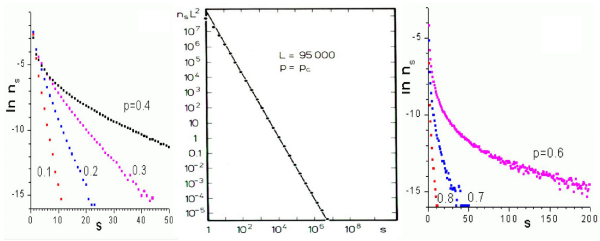
Define lattice as matrix N_{ij} and occupy randomly where a value of 0 means unoccupied and 1 occupied. Obtain cluster size distribution algorithmically, scan lattice left-then-down:

1. start with mass-of-cluster array M_i and number of clusters starting at $k = 2$ (0,1 are already taken)
2. move through lattice N_{ij} and for each (i, j):
   (a) if both top and left are empty, new cluster: $k \leftarrow k+1, N_{ij} \leftarrow k, M_k \leftarrow 1$
   (b) if occupied and only one of (top,left) occupied with k_0, append to cluster: $N_{ij} \leftarrow k_0, M(k_0) \leftarrow M(k_0)+1$
   (c) if occupied and top and left occupied, append site to one of them $N_{ij} \leftarrow k_1, M_{k_1} \leftarrow M_{k_1} + M_{k_2} + 1$ and reference second cluster to first cluster $M_{k_2} = -k_1$
   (d) if cluster with negative mass occurs, use reference to original cluster
3. for $k \leq 2 \ldots k_{\max}$ do:
   sum up cluster masses to get cluster size distributions (only positive numbers) if $M_k > 0$ then $n(M_{k_i}) = n(M_{k_i}) + 1$

- Recursive detection(Stop of we have found a k_0 with $M_{k_0 \geq 0}$)
- Algorithm scales linearly $\mathcal{O}(n)$

3.3.1 Cluster size distribution

Behavior of the relative cluster size n_s



(a) $n_{p<p_c}(s) \propto s^{-\theta} e^{as}$
Subcritical occupation: power law X expontial

(b) $n_{p_c}(s) = s^{-\tau}$
Critical occupation: exponential

(c) $n_{p>p_c}(s) \propto e^{-bs^{1-1/d}}$
Overcritical occupation expontial decay

$$n_p(s) \propto \begin{cases} s^{-\theta} e^{-as} & p < p_c \\ s^{-\tau} & p = p_c \\ e^{bs^{1-\frac{1}{d}}} & p > p_c \end{cases} \tag{7}$$

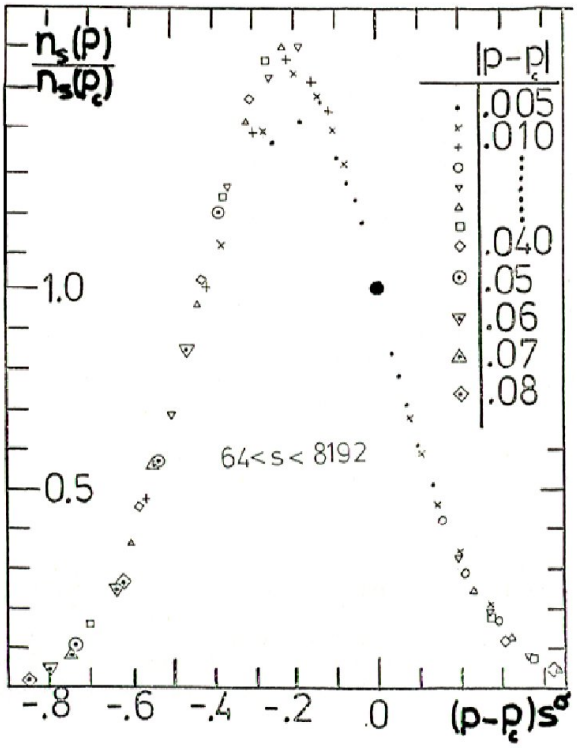
where $\tau = \frac{187}{91}(2D), \approx 2.18(3D)$ **Rescaled distribution**

$$\boxed{\tilde{n}_p(s) = \frac{n_p(s)}{n_{p_c}(s)}}$$

Comparison non-critical distributions n_p to cluster size distribution for p_c (n_{p_c}):

$$\boxed{n_p(s) = s^{-\tau}\mathfrak{R}_{\pm}\left[(p-p_c)s^{\sigma}\right]}$$

with the help of the **scaling function** $\mathfrak{R}_{\pm}$, plot $\tilde{n}_p(s)$ against $(p-p_c)s^{\sigma}$



Second moment of cluster size distribution is a power law:

$$\chi = \left\langle \sum_{s \text{ w.o. biggest}} s^2 n(s) \right\rangle \propto C\,|p-p_c|^{-\gamma} \tag{8}$$

where $\gamma = \frac{43}{18} \approx 2.39(2D), 1.8(3D)$
Due to the divergence aorund p_c, χ is a strong indicator of p_c

Connection to the Ising model:
Magnetic susceptibility diverges near critical temperature.

Scaling exponent relation:

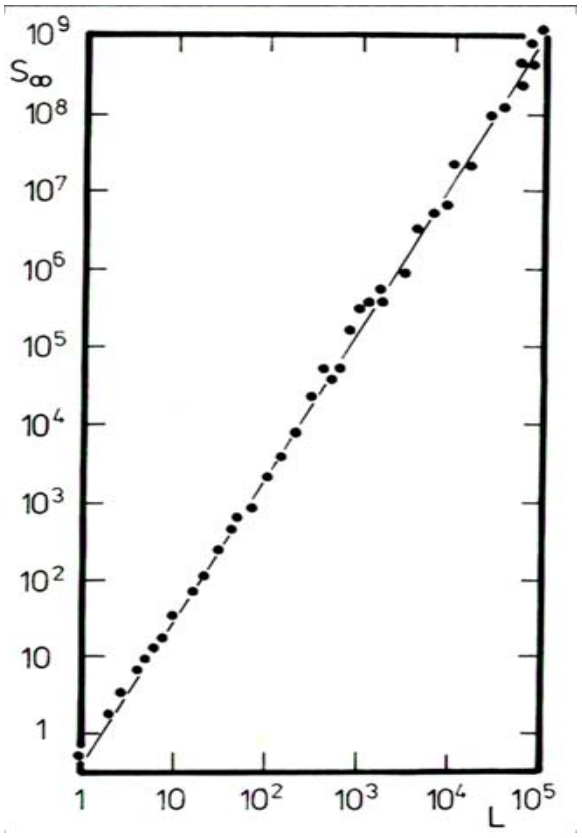
$$\boxed{\gamma = \frac{3-\tau}{\sigma}}$$

3.3.2 Size dependence of the order parameter

Size of the larges cluster **at percolation threshold** ($p = p_c$) is power law in the lattice size

$$\boxed{s_{\infty} \propto L^{d_f}} \tag{9}$$

d_f is the **fractal dimension**!



Derivation in finite-size effects chapter, yielding

$$\boxed{d_f = d - \frac{\beta}{\nu}} \tag{10}$$

$d_f = \frac{91}{48}(2D), \approx 2.51(3D)$

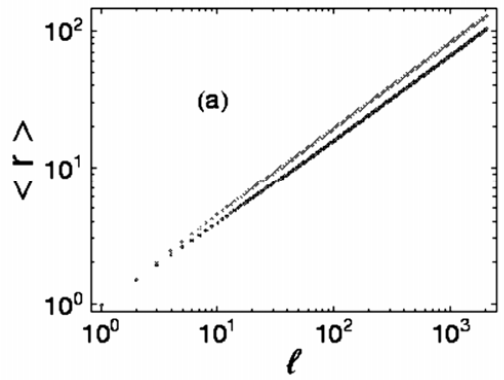
Universality: Coefficients β and ν are universal, hence d_f, **the fractal dimension**, is universal.

3.3.3 The shortest path

Shortest path of spanning cluster is a **power law**

$$t^s \propto L^{d_{min}} \tag{11}$$

$$d_{\min} = \begin{cases} 1.13, & 2D \\ 1.33, & 3D \\ 1.61, & 4D \end{cases} \tag{12}$$



4 Fractals

How well does a fractal object fills a certain space

Self-Similiarity: Object that is built up of smaller copies of itself.

4.1 Fractal dimension

A fractal dimension is **statistical index of complexity** that measures how details in a pattern changes with the measurement scale.

4.1.1 Formal definition

Fractal dimension of an object
Cover object with d-dimensional spheres of radius $r_i < \epsilon$. Consider all possible covering by spheres and let $N_\epsilon(c)$ be the number of spheres in the covering c. The resulting volume is given by:

$$V_\epsilon(c) = \sum_i^{N_i(c)} r_i^d \tag{13}$$

Minimize number of spheres and volume containing the object

$$V_\epsilon^* = \min_{V_\epsilon(c)} \left(\min_{N_\epsilon(c)} \left(V_\epsilon(c) \right) \right) \tag{14}$$

Then fractal dimension is

$$\boxed{d_f := \lim_{\epsilon \to 0} \frac{\log \frac{V_\epsilon^*}{\epsilon^d}}{\log \frac{L}{\epsilon}}} \tag{15}$$

where L is the length of the system.

Infinitesimal limit ($\epsilon \to 0$) of the mathematical definition:

$$\frac{V_\epsilon^*}{\epsilon^d} = \left(\frac{L}{\epsilon} \right)^{d_f} \tag{16}$$

Interpretation of the fractal dimension(s): When length is stretched by a factor a, its volume or mass grows by a factor of a^{d_f}.

Example: Consider the Sierpinski-triangle and stretch one side by the factor 2. This increases the volume(area) by a factor 3.

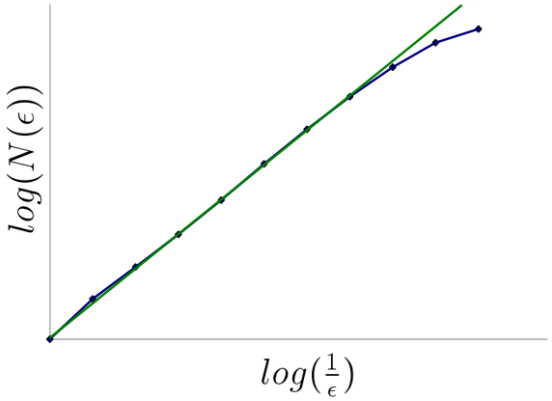
$$\Rightarrow d_f = \frac{\log 3}{\log 2} \approx 1.585$$

4.2 Box counting method

Determine fractal dimension of a fractal object by **superimposing** lattice of lattice constant ϵ and plot number of non-empty $N(\epsilon)$ sites as a function of $\frac{1}{\epsilon}$ in a log-log-plot and determine d_f with

$$d_f = \frac{\log N(\epsilon)}{\log \frac{1}{\epsilon}} \tag{17}$$

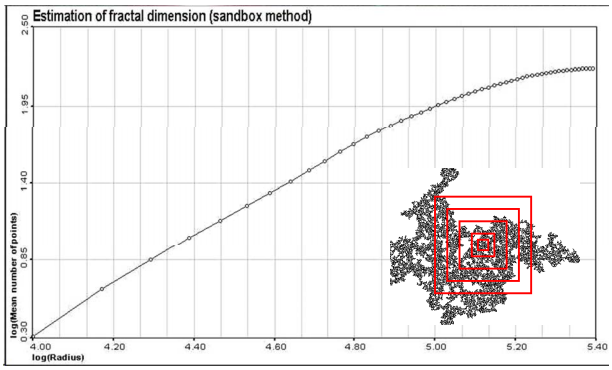
in a region with constant slope.



4.3 Sandbox method

Put a box around the center of a fractal object, increase box size and measure mass of fractal object part inside the box.

$$d_f = \frac{\log M(R)}{\log R} \tag{18}$$



4.4 Correlation function method

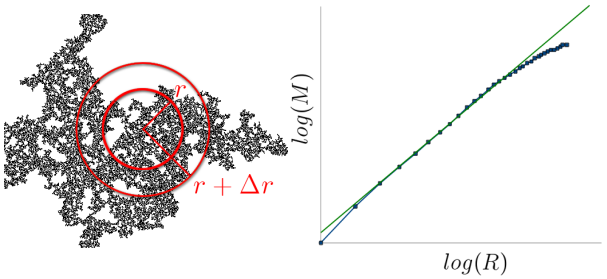
The **correlation function** is a measure for the amount of order in a system and describes the correlation of microscopic variables over distance.
Intuitive Idea: A Large value implies that two quantities strongly influence each other.

Correlation function of the density at the origin and at radius r.

$$\boxed{c(r) = \langle \rho(0)\rho(r) \rangle_{\text{av}}} \tag{19}$$

with some suitable averaging (for instance different origins).

The correlation function method counts the number of filled site within bandsize Δr and normalizes the expression with the surface area at r.



Fractal dimension is obtained by fitting

$$c(r) \propto r^{d_f - d} \tag{20}$$

Correlation function can also be written as

$$c(r) = \underbrace{\frac{\Gamma\left(\frac{d}{2}\right)}{2\pi^{d/2} r^{d-1} \Delta r}}_{\text{surface area at radius r}} \underbrace{[M(r+\Delta r) - M(r)]}_{\text{number of sites in } \delta r} \tag{21}$$

4.5 Correlation Length ξ

Correlation length is the typical length scale over which the correlation functions decays

$$c(r) \propto C + e^{-\frac{r}{\xi}} \tag{22}$$

with an **offset** C, vanishing in the subcritical regime. Furthermore, in this regime, the correlation length is propoertional to the radius of a typical cluster.

Correlation length is singular at p_c

$$\xi \propto |p - p_c|^{-\nu} \tag{23}$$

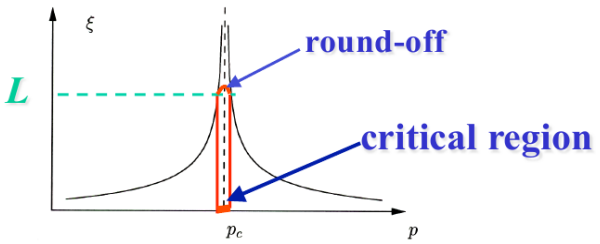
where $\nu = \frac{42}{89}$(2D), 0.88(3D) At $p = p_c$ correlation function decays like a power law:

$$c(r) \propto r^{-(d-2-\eta)} \tag{24}$$

with $\eta = \frac{5}{24}$(2D),-0.05(3D)

4.6 Finite size effects

Correlation length ξ cannot be larger than system size L, therefore maximum instead of a singularity:



The correlation length gets cut at the size L of the system Use two points p_1, p_2 bounding the critical region, then

$$L = \xi(p_1) \propto (p_1 - p_c)^{-\nu}$$
$$p_1 - p_2 \approx 2(p_1 - P - c)$$

assuming p_c lies approximately in the center of the region.

It follows for the size of the critical region:

$$(p_1 - p_2) = L^{-\frac{1}{\nu}} \tag{25}$$

Conclusion: If $L \to \infty$, the critical region vanishes, which is impossible with a finite PC.
Hence we need to extrapolate the behavior

Close to p_c (extrapolation not scaling):

$$\boxed{p_{\text{eff}}(L) = p_c\left(1 - aL^{-\frac{1}{\nu}}\right)} \tag{26}$$

4.7 Finite size scaling

Consider the second moment χ of the cluster size distribution as a function of p and L.
$\to$ can be reduced to a one variable function.
Self-similarity of percolating clusters near critical point

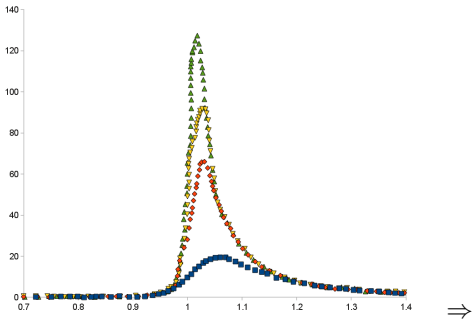
$$\boxed{\chi(p, L) = L^{\frac{\gamma}{\nu}} \mathcal{N}_\chi\left[(p - p_c) L^{\frac{1}{\nu}}\right]} \tag{27}$$

where $\mathcal{N}_\chi$ is the scaling function.

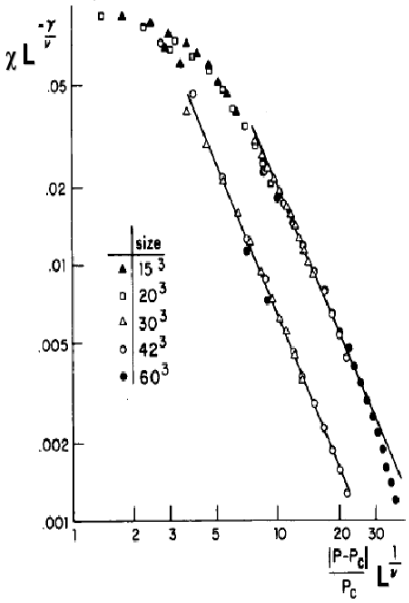
Plotting χ against p for several L-values, leads to differences at the critical value(peak height)

At $p = p_c$ the scaling function approaches a constant and

$$\boxed{\chi_{\max} = L^{\frac{\gamma}{\nu}}} \tag{28}$$



⇒



If we find an expression for the size of the peak depending only on L as well as introducing new parameters, based on previous one, a **data collapse** happens:
Only one parameter is necessary to describe the data.

Size depence of the Order parameter
Fraction of sites in the spanning cluster at p_c:

$$s_\infty \propto L^{d_f}$$
$$\Rightarrow \boxed{PL^d = s_\infty \propto L^{d_f}}$$

4.7.1 Fractal dimension in percolation

Fraction of sites in spanning cluster (order parameter):

$$P(p) = (p - p_c)^\beta \tag{29}$$

consider P as function of p and L, then **finite size scaling**

$$\boxed{P(p, L) = L^{-\frac{\beta}{\nu}} \mathcal{N}_P \left[(p - p_c) L^{\frac{1}{\nu}}\right]} \tag{30}$$

At $p = p_c$ order parameter

$$P = L^{-\frac{\beta}{\nu}} \tag{31}$$

and number of sites of the spanning cluster

$$s_\infty = M \propto L^{d_f} \tag{32}$$

depends on the system size.

We know

$$M = PL^d = L^{-\frac{\beta}{\nu}+d} \overset{!}{=} L^{d_f} \tag{33}$$
$$d_f = d - \frac{\beta}{\nu} \tag{34}$$

4.8 Cellular automata

Discrete model of grid cells, each can have a finite number of states, in some finite dimension. **Completely deterministic discrete time evolution**: After each time step, calculate new state as function of previous state of all grid cells.

5 Monte-Carlo-Methods

Main advantage: Error decreases with number of samples N like

$$\boxed{\Delta \propto \frac{1}{\sqrt{N}}} \tag{35}$$

5.1 Applications of Monte-Carlo

Calculation of π: Use quarter of circle $r = 1$ to approximate $\frac{\pi}{4}$.

$$\frac{\pi}{4} = \int_0^1 \sqrt{1 - x^2}\, \mathrm{d}x \tag{36}$$

Monte-Carlo algorithm: count number of random points inside the circle:

$$\pi(N) = 4\frac{N_{\text{inside}}}{N} \tag{37}$$

5.2 Computation of integrals

Basic method: Pick N random points x_i in the integral interval and approximate integral

$$\boxed{\int_a^b g(x)\, \mathrm{d}x \approx \frac{(b-a)}{N} \sum_{i=1}^{N} g(x_i)} \tag{38}$$

One doesn't need to integrate analytically!

Simple sampling, good for smooth functions.

Importance sampling:

$$\int_a^b g(x)\, \mathrm{d}x = \int_a^b \frac{g(x)p(x)}{p(x)}\, \mathrm{d}x \approx \frac{(b-a)}{N} \sum_{i=1}^{N} \frac{g(x_i)}{p(x_i)} \tag{39}$$

$p(x)$ is the distribution functions which is the distribution we use to choose the random points (transform uniform random number distribution to this distribution function).

Remaining requirement (less strong): Just $\frac{g(x)}{p(x)}$ needs to be smooth.

5.2.1 Error of integration

Convetional methods:e.g. trapezium rule over $[x_0, x_0 + \Delta x]$

$$\begin{aligned}\int_{x_0}^{x_0+\Delta x} \mathrm{d}x &= f(x_0)\Delta x + \frac{1}{2}f'(x_0)\Delta x^2 + \frac{1}{6}f''(x_0)\Delta x^3 + \cdots \\ &= \underbrace{\frac{1}{2}(f(x_0) + f(x_0 + \Delta x))\Delta x}_{\text{trapezium rule}} + \mathcal{O}(\Delta x^3)\end{aligned}$$

Hence, the error behaves like $\propto \Delta x^3$

Compound trapezium rule: $[x_0, x_1]$ will be subdivided into N pieces with length $\Delta x = \frac{x_1 - x_0}{N}$:

$$\begin{aligned}\int_{x_0}^{x_1} &\approx \frac{\Delta x}{2} \sum_{j=0}^{N-1} [f(x_0 + j\Delta x) + f(x_0 + (j+1)\Delta x)] \\ &= \frac{\Delta x}{2} [f(x_0) + 2f(x_0 + \Delta x) + 2f(x_0 + 2\Delta x) + \cdots \\ &\quad + 2f(x_0 + (N-1)\Delta x) + f(x_1)]\end{aligned}$$

- Error for each step $\propto \Delta x^3$ (as calculated above)
- Cumulative Error $N \times \mathcal{O}(\Delta x^3) = N\mathcal{O}(N^{-3})\mathcal{O}(N^{-2})$

Generalization for $d \geq 2$

- Segment: Δx^{d+2}, $T \propto N \propto \frac{1}{\Delta x^d}$,$\Delta x \propto T^{-\frac{1}{d}}$
- Cumulative: $\propto \Delta x^2 \propto T^{-\frac{2}{d}}$

Monte-Carlo-Error
Assume N equidistant points in $[a, b]$ with $h = \frac{b-a}{N}$, then

$$\boxed{I = \int_a^b g(x)\,\mathrm{d}x \approx \frac{b-a}{N}\sum_{i=1}^{N} g(x_i) = (b-a)\langle g\rangle =: Q}$$

where $\langle g\rangle$ is the sample mean of the integrand.

Variance

$$\mathrm{var}(g) =: \sigma^2 = \frac{1}{N-1}\sum_{i=1}^{N}\left(g(x_i) - \langle g\rangle\right)^2$$

where the denominator $N-1$ is due to unbiasedness of the estimator.
By the central limit theorem follows then the variance of our integral Q:

$$\mathrm{var}(Q) = (b-a)^2\frac{operatornamevar(g)}{N} = (b-a)^2\frac{\sigma^2}{N}$$
$$\Rightarrow \boxed{\delta Q \approx \sqrt{\mathrm{var}(Q)} = (b-a)\frac{\sigma}{\sqrt{N}}}$$

Summary:
Conventional numerical integration with N equidistant points (distance $h = \frac{(b-a)}{N}$) in 1D:

$$\text{area} = A \propto \frac{1}{N^2} \propto \frac{1}{T^2} \tag{40}$$

Error goes as

$$\Delta \propto (AN)^2 \propto \frac{1}{N^2} \propto (\Delta x)^2 \tag{41}$$

Trapezian rule:

$$\Delta \propto (\Delta x)^2 \propto \frac{1}{N^2} \propto (T)^{\frac{2}{d}} \tag{42}$$

since

$$T \propto N \propto \frac{1}{(\Delta x)^d} \tag{43}$$

$\Rightarrow$ error independent of dimension, but computation time not.

For d dimensions conventional:

$$T \propto N \propto \left(\frac{1}{\Delta x}\right)^d \tag{44}$$
$$\Rightarrow \Delta x \propto T^{-\frac{1}{d}} \tag{45}$$
$$\Delta \propto \left(N\Delta x\Delta x^d\right)^2 = T^{-\frac{2}{d}} \tag{46}$$

For d dimensions Monte-Carlo:

$$\Delta \propto \frac{1}{\sqrt{N}} \propto T^{-\frac{1}{2}} \tag{47}$$

Comparison of errors
There is a critical dimension, where MC becomes more efficient

$$T^{-\frac{2}{d}} \stackrel{!}{=} \frac{1}{\sqrt{T}} \qquad d_{\text{crit}} = 4$$

5.3 Higher-dimensional integrals

Example: Hard Spheres placed in 3D Volume: Consider N hard spheres with radiu R placed in a 3D box with Volume V.
The distance between two points is given as

$$\boxed{r_{ij} = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2 + (z_i - z_j)^2}}$$

From the hard sphere/no overlap condition follows that $\boxed{r_{ij} > 2R}$.
We want to compute the **average distance** $\langle r_{ij}\rangle$ between two centers:

$$\langle r_{ij}\rangle = \underbrace{\frac{1}{\int \mathrm{d}^3\mathbf{r}_1\,\mathrm{d}^3\mathbf{r}_2\ldots\mathrm{d}^3\mathbf{r}_N}}_{=:Z^{-1}}\int \frac{2}{N(N-1)}\times \tag{48}$$
$$\times\sum_{i<j} r_{ij}\,\mathrm{d}^3\mathbf{r}_1\,\mathrm{d}^3\mathbf{r}_2\ldots\mathrm{d}^3\mathbf{r}_N \tag{49}$$

The MC-approach to solve this problem is as follows:

- Choose particle position (center of the sphere)
- If sphere overlaps with already existing one: retry
- after placement of all spheres calculate the average r_{ij}

5.4 Canonical Monte-Carlo

Ensemble: Large number of identical systems

Microcanoncial ensemble: Closed system with constant N, T, U (inner energy).

Canonical ensemble: Closed system with heat reservoir and fixed N, T, V

Grand canonical ensemble: Open system (exchange heat and particles with environment) with fixed μ, T, V

Ensemble average over phase space Λ with probability measure $\mathrm{d}\mu$ (normalization with partition function)

$$\langle f\rangle = \int_\Lambda f\,\mathrm{d}\mu = \overline{f}_t = \lim_{T\to\infty}\frac{1}{T}\int_0^T f(x(t))\,\mathrm{d}t \tag{50}$$

- The normalizing factor of the measure is called **partition function**
- From the ergodic hypothesis follows that all microstates are **equiprobable**
- The energy of configuration X is $E(X)$
- Probability (at thermal equilibrium) given by

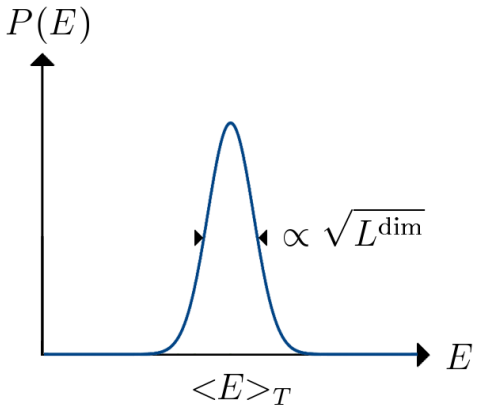
$$\boxed{p_{eq} = \frac{1}{Z_T}e^{-\frac{E(X)}{k_B T}}}$$

with the partition function $Z_T = \sum_X e^{-\frac{E(X)}{k_B T}}$.

Discrete ensemble average

$$\langle Q \rangle = \sum_X Q(X) \underbrace{p_{\text{eq}}}_{\text{Boltzmann}} (X) \tag{51}$$

Problem of Sampling
It is inefficient to calculate ensemble averages in an equally distributed system. Hand-waving argument: Peak of energy increase as $\sqrt{L^{\text{d}}}$, but system size increases as L^d, therefore relative peak width **decreases** with increasing system size.



5.4.1 Markov chains

Start in configuration X and propose new configuration Y with probability $T(X \to Y)$.

Properties for proposing a new state:

- **Ergodicity**: reach any possible configuration after finite number of steps (A state is *ergodic* if it is aperiodic and positively recurrent)
- **Normalization**: $\sum_Y T(X \to Y) = 1$
- **Reversibility**: $T(X \to Y) = T(Y \to X)$

$\to$ not every new configuration is also **accepted**.

Accept a new configuration with some acceptance probability to control dynamics (e.g. temperature dependence), therefore **total Markov chain acceptance probability** (Overall probability of a configuration making it through both steps.)

$$W(X \to Y) = \underbrace{T(X \to Y)}_{\text{Transition Prob}} \cdot \underbrace{A(X \to Y)}_{\text{Acceptance Prob}} \tag{52}$$

(Can also be interpreted as conditional probability of acceptance for given y)

Master equation

$$\boxed{\frac{\mathrm{d}p(X,t)}{\mathrm{d}t} = \sum_Y p(Y)W(Y \to X) - \sum_X p(X)W(X \to Y)} \tag{53}$$

where $p(x,t)$ is the probability to find x in time t

Properties of $W(x \to Y)$:

- **Ergodicity**: $\forall X, Y \quad W(X \to Y) > 0$
- **Normalization**: $\sum_Y W(X \to Y) = 1$
- **Homogeneity**: $\sum_Y p_{\text{st}}(Y)W(Y \to X) = p_{\text{st}}(X)$

5.4.2 Detailed Balance

The stationary states of the Markov chains,

$$\frac{\mathrm{d}p(X,t)}{\mathrm{d}t} = 0 \tag{54}$$

should model Boltzmann equilibrium distribution:

$$p_{\text{st}}(X) = p_{\text{eq}}(X) = \frac{1}{Z_T} e^{-\frac{E(X)}{k_B T}} \quad \forall X \tag{55}$$

$$\Rightarrow \sum_Y p_{eq}(Y)W(Y \to X) = \sum_Y P_{eq}(X)W(X \to Y) \tag{56}$$

One finds the **detailed balance condition**

$$\boxed{p_{\text{eq}}(X)W(X \to Y) = p_{\text{eq}}(Y)W(Y \to X) \quad \forall X, Y} \tag{57}$$

such that **the steady state is the thermal equilibrium.**

Since $W(X \to Y) = T(X \to Y) \cdot A(X \to Y)$ and $T(X \to Y) = T(Y \to X)$ one can rewrite the detailed balance condition to

$$p_{\text{eq}}(X)A(X \to Y) = p_{\text{eq}}(Y)A(Y \to X) \quad \forall X, Y \tag{58}$$

5.4.3 MR^{2}T^2

Basic Idea: Carry out importance sampling through a Markov Chain. Acceptance probability is

$$A(X \to Y) = \min\left(1, \frac{p_{\text{eq}}(Y)}{p_{\text{eq}}(X)}\right) \tag{59}$$

$$= \min\left(1, \frac{\frac{1}{Z} e^{-\frac{E(Y)}{kT}}}{\frac{1}{Z} e^{-\frac{E(X)}{kT}}}\right) \tag{60}$$

$$= \min\left(1, e^{-\frac{(E(Y)-E(X))}{kT}}\right) \tag{61}$$

$$\tag{62}$$

$$\boxed{A(X \to Y) = \min\left(1, e^{\frac{\Delta}{k_B T}}\right)}$$

Always accept transitions to lower energy. Thermal equilibrium is enforced by detailed balance.

5.4.4 Glauber dynamics

Acceptance probability is

$$\boxed{A(X \to Y) = \frac{e^{-\frac{\Delta E}{kT}}}{1 - e^{-\frac{\Delta E}{kT}}}} \tag{63}$$

Glauber dynamics are superior at low temperatures due to different acceptance formulation.

5.5 Ising model

Consider a discrete collection of N binary variables(spins) $\sigma_i \in \{-1, +1\}$ Hamiltonian

$$\mathcal{H} = E = -\sum_{i,j} J_{ij}\sigma_i\sigma_j - H_i\sigma_i \tag{64}$$

Coupling $J_{ij} = J$ is typically just for nearest neighbors and H_i usually homogeneous external field.

Example: 1D ferromagnetic Ising: $E = \sum_i \sigma_i\sigma_{i+1}$.

5.5.1 Monte-Carlo-Algorithm

1. Choose randomly site i having spin state σ_i
2. Calculate

$$\begin{aligned}\Delta E = E(Y) - E(X) &= \sum_{<i,j>\text{n.n.}} 2J\sigma_i\sigma_j \\ &= 2J\sigma_i h_i \\ h_i &= \sum_{\text{n.n of}i} \sigma_j\end{aligned}$$

3. If $\Delta E < 0$ flip spin
4. If $\Delta E \geq 0$ flip spin with probability $e^{-\frac{\Delta E}{kT}}$

Sweep: Group of N steps.

Magnetization Let M be the magenetization, χ magnetic susceptibility and H the magnetic field strength. Then

$$M = \chi H \tag{65}$$

$$M(T) = \frac{1}{N}\lim_{H\to 0}\sum_{i=1}^{N}\sigma_i \tag{66}$$

$$\propto \begin{cases} |T_c - T|^\beta & T < T_c \\ 0 & T > T_c \end{cases} \tag{67}$$

where $\beta = \frac{1}{8}$(2D), 0.326(3D). We have a singularity at the critical temperature(or a maximum if the system is finite).

Magnetic susceptibility

$$M = \chi H \tag{68}$$

$$\chi \propto (T - T_c)^{-\gamma} \tag{69}$$

Energy and heat capacity Energy increases with T (S-like curve), heat capacity has peak at T_c

5.6 Binary mixtures and Kawasaki dynamics

Now two species with constant N_a and N_b.

Kawasaki Dynamics: Choose A-B pair (bond), calculate energy difference and flip with Metropolis or Glauber probability.

6 Random walk

Probability to go n_1 steps to the right

$$P_N(n_1) = \binom{N}{n_1} p^{n_1} q^{N-n_1} \tag{70}$$

Average distance consists of moving to the left and right:

$$\langle m\rangle = \langle n_1\rangle - \langle n_2\rangle \tag{71}$$

$$= (p-q)N \tag{72}$$

with

$$\langle n_1\rangle = \sum_{n_1} n_1 \binom{N}{n_1} p^{n_1} q^{N-n_1} \tag{73}$$

$$\langle \Delta m^2\rangle = \langle (m - \langle m\rangle)^2\rangle = \langle m^2\rangle - \langle m\rangle^2 = 4Npq \tag{74}$$

space covered:

$$\sqrt{\langle \Delta m^2\rangle} = \sqrt{N}$$

7 Solving equations

Newton method

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)} \tag{75}$$

n-dimensional case: use Jacobi matrix.

Secant method

$$x_{n+1} = x_n - (x_n - x_{n-1})\frac{f(x_n)}{f(x_n) - f(x_{n-1})} \tag{76}$$

n-dimensional case: use approximation to Jacobi matrix.

Bisection Method: Choose two starting values x_0 and x_1 below and above x-axis, calculate

$$\text{sign} f(x_m) = \text{sign} f(\frac{x_0 + x_1}{2}) \tag{77}$$

and use x_m and one of the two starting values as new values to repeat method.

False position method: Same idea, but use not arithmetic mean of x_1 and x_0 but secant.

8 Ordinary Differential Equations

General first order ODE with initial value problem:

$$\frac{\mathrm{d}y}{\mathrm{d}t} = f(y,t) \tag{78}$$

with

$$y(t_0) = y_0. \tag{79}$$

Examples include radioactive decay

$$\frac{\mathrm{d}N}{\mathrm{d}t} = -\lambda N \tag{80}$$

and cooling of coffee

$$\frac{\mathrm{d}T}{\mathrm{d}t} = -\gamma(T - T_{\text{room}} \tag{81}$$

8.1 Order of a numerical method

Definition of the order of a numerical method with discrete time steps:

A method is **locally of order** n if error at **one time step** is $\mathcal{O}((\Delta t)^n)$.

The error over a whole interval T consisting out of m time steps ($m(\Delta T) = T$)

$$m\mathcal{O}((\Delta t)^n) = \frac{T}{\Delta t}\mathcal{O}((\Delta t)^n) = \mathcal{O}((\Delta t)^{n-1}) \tag{82}$$

and hence the order is **globally** of order $n-1$.

8.2 Euler method

Simplest **finite differences method**, the **Euler method**:

1. use initial value $y(t_0) = y_0$ as starting point
2. calculate $\frac{\mathrm{d}y}{\mathrm{d}t}$ with $y(t_0) = y_0$ and $t = t_0$
3. advance linearly in t:

$$\boxed{y(t + \Delta t) = y(t) + \Delta t \frac{\mathrm{d}y(t)}{\mathrm{d}t}} \tag{83}$$

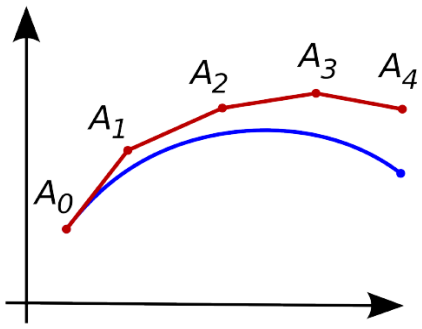
4. repeat

The Euler method are the first two terms in the Taylor expansion of the function around t_0:

$$y(t_0 + \Delta t) = y(t_0) + \Delta t \frac{\mathrm{d}y}{\mathrm{d}t}(t_0) + \mathcal{O}((\Delta t)^2) \tag{84}$$

$$= \underbrace{y(t_0) + \Delta t(y_0, t_0)}_{:= y(t_1) := y_1} + \mathcal{O}((\Delta t)^2) \tag{85}$$

Euler method is locally of order 2 and globally of order 1, error accumulates:



Improve method by making time steps Δt smaller.

8.3 Generalization to n-order ODEs

Write nth-order ODE as n coupled differential equations:

$$\frac{\mathrm{d}f_i}{\mathrm{d}t} = f_i(y_1, y_2, \ldots, y_n, t) \tag{86}$$

Euler method is then

$$\boxed{y_i(t_{n+1}) = y_i(t_n) + \Delta t f_i(y_1, y_2, \ldots, y_n, t) + \mathcal{O}((\Delta t)^2)} \tag{87}$$

8.4 Runge-Kutta Methods

Runge-Kutta methods are the q-order generalization of the Euler method, which is the first order method.

Derive from Taylor expansion:

$$y(t + \Delta t) = y(t) + \frac{(\Delta t)}{1!}\frac{\mathrm{d}y}{\mathrm{d}t} + \frac{(\Delta t)^2}{2!}\frac{\mathrm{d}^2 y}{\mathrm{d}t^2} + \ldots \tag{88}$$

$$+ \frac{(\Delta t)^q}{q!}\frac{\mathrm{d}^q y}{\mathrm{d}t^q} + \mathcal{O}((\Delta T)^q) \tag{89}$$

8.4.1 2nd order Runge-Kutta Methods

1. Perform Euler step of size $\frac{\Delta}{2}$, starting at initial vlaue $y_i(t_0)$

$$y_i(t + \frac{1}{2}\Delta t) = y_i(t) + \frac{1}{2}\Delta t f(y_i(t), t)$$

2. Calculate derivative at the reached point
3. advance full time step with calculated derivative as slope
4. Repeat previous steps

$$y_i(t + \Delta t) = y_i(t) + \Delta t \left[y_i\left(t + \frac{1}{2}\Delta t\right), t + \frac{1}{2}\Delta t \right] \tag{90}$$

$$+ \mathcal{O}((\Delta T)^3) \tag{91}$$

8.4.2 4th order Runge-Kutta Methods

Define

$$k_1 = f(y_n, t_n) \tag{92}$$

$$k_2 = f(y_n + \frac{1}{2}(\Delta t)k_1, t_{n+\frac{1}{2}(\Delta t)}) \tag{93}$$

$$k_3 = f(y_n + \frac{1}{2}(\Delta t)k_2, t_n + \frac{1}{2}(\Delta t)) \tag{94}$$

$$k_4 = f(y_n + (\Delta t)k_3, t_n + (\Delta t)) \tag{95}$$

and calculate

$$y_{n+1} = y_n + \Delta t\left(\frac{1}{6}k_1 + \frac{1}{3}k_2 + \frac{1}{3}k_3 + \frac{1}{6}k_4\right) + \mathcal{O}((\Delta T)^5) \tag{96}$$

8.4.3 q-stage Runge-Kutta method

The Runge-Kutta method can be generalized, defined by the so-called stage (number of terms in summation):

$$y_{n+1} = y_n + \Delta t \sum_{i=1}^{q} \omega_i k_i \tag{97}$$

with

$$k_i = \left[f(y_n + \Delta t \sum_{j=1}^{i-1} \beta_{ij} k_j, t + (\Delta t)\alpha_i \right] \quad \alpha_i = 0 \tag{98}$$

The coefficient are ambiguously defined in a **Butcher** array:

α_2	$\beta_{2,1}$				
α_3	$\beta_{3,1}$	$\beta_{3,2}$			
$\vdots$	$\vdots$		$\ddots$		
α_q	$\beta_{q,1}$	$\beta_{q,2}$	...	$\beta_{q,q-1}$	
	ω_1	ω_2	...	ω_{q-1}	ω_q

8.4.4 Order of q-stage Runge-Kutta

Up to RK4, the stage corresponds to the order of the method. In general not true, instead:

A Runge-Kutta method is of order p if the right side vanished up to order p:

$$y(t + \Delta t) - y(t) = \underbrace{\sum_{m=1}^{p} \frac{(\Delta t)^m}{m!} \left[\frac{\mathrm{d}^{m-1} f}{\mathrm{d}m - 1} \right]}_{\stackrel{!}{=} 0} + \mathcal{O}((\Delta t)^{p+1}) \tag{99}$$

Therefore

$$\sum_{i=1}^{q} \omega_i k_i = \sum_{m=1}^{p} \frac{(\Delta t)^m}{m!} \left[\frac{\mathrm{d}^{m-1} f}{\mathrm{d}m - 1} \right] + \mathcal{O}((\Delta t)^{p+1}) \tag{100}$$

9 Error estimation

The Runga-Kutta methods suffer from errors that are being summed up. To improve the method, the error should be estimated and the method be improved. Furthermore, the Runge-Kutta methods are **single-step methods** which include only one previous point into the calculation. Instead more previous steps can be included to improve the method $\Rightarrow$ predictor corrector.

9.1 Improvement using error estimation

Let Φ be the evolution operator of the method with order p: Calculate difference between 2x method with time step Δt(called y_2) and 1x method with time step $2(\Delta t)$ (called y_1:

$$\delta = y_1^{(2\Delta t)} - y_2^{2\times(\Delta t)} \tag{101}$$

$$y(t + 2\Delta t) = \begin{cases} y_1 + (2\Delta t)^{p+1}\Phi + \mathcal{O}((\Delta t)^{p+2}) \\ y_2 + 2(\Delta t)^{p+1}\phi + \mathcal{O}((\Delta t)^{p+2}) \end{cases} \tag{102}$$

$$\Rightarrow \delta = (2^{p+1} - 2)\Phi + \mathcal{O}((\Delta t)^{p+2}) \tag{103}$$

$$\Rightarrow y(t + \Delta t) = y_2 + \frac{2\delta}{2^{p+1} - 2} + \mathcal{O}((\Delta t)^{p+2}) \tag{104}$$

method is better since error is one order higher.

For instance for RK4:

$$y(t + \Delta t) = y_2 + \frac{\delta}{15} + \mathcal{O}((\Delta t)^6) \tag{105}$$

9.2 Adaptive time steps

Basis idea: Use error estimates to adapt time step size.

$$(\Delta t)_{\text{new}} = (\Delta t)_{\text{old}} \left(\frac{\delta_{\text{expected}}}{\delta_{\text{measured}}} \right)^{\frac{1}{p+1}} \tag{106}$$

since $\delta \propto (\Delta t)^{p+1}$.

9.3 Predictor-Corrector method

Multistep method Idea: carry out Euler step with arithmetic mean of derivative at $y(t)$ and $y(t + \Delta t)$:

$$y(t + \Delta t) \approx y(t) + \Delta t \left(\frac{f(y(t), t) + f(y(t + \Delta t), t + \Delta t)}{2} \right) \tag{107}$$

But: Implicit equation, cannot be solved directly. Use a prediction method, in this case Taylor expansion, to predict value of $f(y(t + \Delta t), t + \Delta t)$:

$$y^p(t + \Delta t) = y(t) + \Delta t \frac{\mathrm{d}y}{\mathrm{d}t}(t) + \mathcal{O}((\Delta t)^2) \tag{108}$$

Now compute corrected value of $y(t+\Delta t)$ with the corrector

$$y^c(t + \Delta t) = y(t) \tag{109}$$

$$+ \Delta t \left(\frac{f(y(t), t) + f(y^p(t + \Delta t), t + \Delta t)}{2} \right) + \mathcal{O}((\Delta t)^3) \tag{110}$$

This is referred as PEC, one can also do PECEC: the corrected value can be inserted into the corrector once more (or many times, can be done iteratively) to obtain a better approximation.

Higher-order predictor-corrector methods: Include higher terms in the predictive Taylor expansion:

$$y^p(t + \Delta t) = y(t) + \frac{\Delta t}{1!}\frac{\mathrm{d}y}{\mathrm{d}t}(t) + \frac{(\Delta t)^2}{2!}\frac{\mathrm{d}^2 y}{\mathrm{d}t^2}(t) \tag{111}$$

$$+ \frac{(\Delta t)^3}{3!}\frac{\mathrm{d}^3 y}{\mathrm{d}t^3}(t) + \mathcal{O}((\Delta t)^4) \tag{112}$$

For instance, include first four terms for 3rd order predictor method.

Insert Taylor predictor(new predictor instead of the one above) into

$$\left(\frac{\mathrm{d}y}{\mathrm{d}t}\right)^c(t+\Delta t) = f(y^p(t+\Delta t), t+\Delta t)) \tag{113}$$

Error is

$$\delta = \left(\frac{\mathrm{d}y}{\mathrm{d}t}\right)^c(t+\Delta t) - \left(\frac{\mathrm{d}y}{\mathrm{d}t}\right)^p(t+\Delta t) \tag{114}$$

Correct (express everything except the first derivative which is already defined):

$$y^c(t+\Delta t) = y^p + c_0\delta \tag{115}$$
$$\left(\frac{\mathrm{d}^2y}{\mathrm{d}t^2}\right)^c(t+\Delta t) = \left(\frac{\mathrm{d}^2y}{\mathrm{d}t^2}\right)^p + c_2\delta \tag{116}$$
$$\left(\frac{\mathrm{d}^3y}{\mathrm{d}t^3}\right)^c(t+\Delta t) = \left(\frac{\mathrm{d}^3y}{\mathrm{d}t^3}\right)^p + c_3\delta \tag{117}$$

with **Gear coefficients**

$$c_0 = \frac{3}{8} \quad c_2 = \frac{3}{4} \quad c_3 = \frac{1}{6} \tag{118}$$

Coefficients are obtained in a similar way to the Runge-Kutta methods, just by requiring the method to be of a certain order.

Higher order methods become very complex, e.g. 5th order Runge-Kutta method.

9.4 Sets of coupled ODEs

Straight-forward generalization to coupled ODEs:

$$\frac{\mathrm{d}f_i}{\mathrm{d}t} = f(y_1, \ldots, y_N, t) \tag{119}$$

9.5 Stiff differential equations

Success of a computational method is often not only governed by the quality of the method and the stepsize.

Some **stiff** equations are numerically unstable unless you choose a very small stepsize.

Example:

$$\frac{\mathrm{d}y}{\mathrm{d}t} = -15y(t) \tag{120}$$

becomes very unstable unless good method is used (**Adams-Moulton** in this case).

$$y(t+\Delta T) = y(t) + \frac{1}{2}\Delta t\left(f(y(t),t) - f(y(t+\Delta t), t+\Delta t)\right) + \mathcal{O}(\Delta t^2)$$

For sets of a equations a large eigenvalue is an indication that a set of equation is unstable.

10 Partial Differential Equations

Types of PDEs for two-variable PDEs:

$$a(x,t)\partial_x^2 u(x,t) + b(x,t)\partial_x\partial_t u(x,t) + c(x,t)\partial_t^2 u(x,t) \tag{121}$$
$$+d(x,t)\partial_x u(x,t) + e(x,t)\partial_t u(x,t) + f(u,x,t) = 0 \tag{122}$$

- elliptic for $a(x,t)c(x,t) - \frac{b(x,t)^2}{4} > 0$
- parabolic for $a(x,t)c(x,t) - \frac{b(x,t)^2}{4} = 0$
- hyperbolic for $a(x,t)c(x,t) - \frac{b(x,t)^2}{4} < 0$

10.1 Discretization of derivatives

Find solution of a PDE by using discrete space, i.e. a lattice.

10.1.1 First derivative in 1D

Implement first derivatives as two-point formulas

$$\frac{\mathrm{d}\Phi}{\mathrm{d}t} = \begin{cases} \frac{\Phi(x_{n+1})-\Phi(x_n)}{\Delta x} & +\mathcal{O}((\Delta x)) \\ \frac{\Phi(x_n)-\Phi(x_{n-1})}{\Delta x} & +\mathcal{O}((\Delta x)) \end{cases} \tag{123}$$

or as three-point formulas

$$\frac{\mathrm{d}\Phi}{\mathrm{d}t} = \frac{\Phi(x_{n+1}) - \Phi(x_{n-1})}{2\Delta x} + \mathcal{O}((\Delta x)^2) \tag{124}$$

10.1.2 Second derivatives

(using the two-point-formulae)

$$\frac{\mathrm{d}^2\Phi}{\mathrm{d}t^2} = \frac{\Phi(x_{n+1}) + \Phi(x_{n-1}) - 2\Phi(x_n)}{(\Delta x)^2} + \mathcal{O}((\Delta x)^2) \tag{125}$$

in two dimensions:

$$\Delta\Phi = \frac{1}{(\Delta x)^2}\left[\Phi(x_{n+1}, y_n) + \Phi(x_{n-1}, y_n)\right. \tag{126}$$
$$\left. + \Phi(x_n, y_{n+1}) + \Phi(x_n, y_{n-1}) - 4\Phi(x_n, y_n)\right] + \mathcal{O}((\Delta x)^2) \tag{127}$$

10.2 Poisson equation

Discretize space:

- x_n with $n = 1, \ldots, N$
- $\Phi_n = \Phi(x_n)$
- $\Phi_{n+1} + \Phi_{n-1} - 2\Phi_n = \Delta x^2\rho(x_n)$
- $\Phi_0 = C_0,\ \Phi_1 = C_1$ (Dirichlet BC)

$$\begin{pmatrix} -2 & 1 & 0 & \dots & \dots \\ 1 & -2 & 1 & 0 & \dots \\ 0 & 1 & -2 & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \vdots \end{pmatrix} \begin{pmatrix} \Phi_1 \\ \Phi_2 \\ \vdots \\ \Phi_{N-1} \end{pmatrix} = - \begin{pmatrix} c_0 \\ 0 \\ \vdots \\ c_1 \end{pmatrix}$$

In 1D:

$$\frac{\partial^2 \Phi}{\partial x^2} = \rho(x) \tag{128}$$

Need to solve $A\vec{\Phi} = b$!

Use linear equation solvers to solve the matrix equation. Do not use Gaussian elimination since it takes forever.

2D:

$$\Phi_{i+1,j} + \Phi_{i-1,j} + \Phi_{i,j+1} + \Phi_{i,j-1} - 4\Phi_{i,j} = \Delta x^2 \rho_{i,j}$$

10.3 Jacobi method

Simplest relaxation method (relaxation ≈ smoothing operators on the matrices)

Decompose

$$A = D + U + L \tag{129}$$

Now **Jacobi method** is

$$\boxed{\vec{\Phi}(t+1) = D^{-1}(\vec{b} - (U+L)\vec{\Phi}(t))} \tag{130}$$

Method is very slow, therefore terminate after precision is smaller than some required accuracy:

$$\delta'(t+1) = \frac{\left|\left|\vec{\Phi}(t+1) - \vec{\Phi}(t)\right|\right|}{\left|\left|\vec{\Phi}(t)\right|\right|} \leq \epsilon \tag{131}$$

Evolution operator of error and error from

$$\vec{\delta}(t+1) = A^{-1}\vec{b} - \vec{\Phi}(t+1) \tag{132}$$

$$= \underbrace{-D^{-1}(U+L)}_{\Lambda} \underbrace{\left(A^{-1}\vec{b} - \vec{\Phi}(t)\right)}_{\delta(t)} = -\Lambda\vec{\delta}(t) \tag{133}$$

Λ is evolution operator for the error and the largest eigenvalue should have a norm smaller than one for convergence of the method.
Hence

$$\Phi_n = \Phi^* + \lambda^n \vec{c}$$

10.3.1 Error in Jacobi method

How to find eigenvale λ
We pick up one factor λ from every iteration.

$$\frac{\|\vec{\Phi}(n+1) - \vec{\Phi}(n)\|}{\|\vec{\Phi}(n) - \vec{\Phi}(n-1)\|} \approx \frac{\lambda^{n+1} - \lambda^n}{\lambda^n - \lambda^{n-1}} = \lambda$$

Link between real error δ and error δ'

$$\delta(n) = \frac{\|\vec{\Phi}^* - \vec{\Phi}(n)\|}{\|\Phi(n)\|} \approx \frac{\vec{\Phi}^* - \vec{\Phi}^* - \lambda^n\vec{c}}{\|\vec{\Phi}(n)\|}$$

$$= \frac{\|\vec{c}\|}{\|\vec{\Phi}(n)\|}\lambda^n$$

$$\delta'(n) = \frac{\|\vec{\Phi}(n+1-\vec{\Phi}(n))\|}{\|\Phi(n)\|} \approx \frac{\|\vec{\Phi}^* + \lambda^{n+1}\vec{c} - \vec{\Phi}^* - \lambda^n\vec{c}\|}{\|\vec{\Phi}(n)\|}$$

$$= \underbrace{\frac{\|\vec{c}\|}{\|\vec{\Phi}(n)\|}\lambda^n}_{\delta(n)} |\lambda - 1| = \delta(n)|\lambda - 1|$$

Hence we have

$$\boxed{\delta(n) \approx \frac{\delta'(n+1)}{1-\lambda}}$$

and finally after rewriting $1 - \lambda$

$$\boxed{\delta(n) = \frac{\|\vec{\Phi}(n+1) - \vec{\Phi}(n)\|\|\vec{\Phi}(n) - \vec{\Phi}(n-1)\|}{\|\vec{\Phi}(n)\|(\|\vec{\Phi}(n) - \vec{\Phi}(n-1)\| - \|\vec{\Phi}(n+1) - \vec{\Phi}(n)\|)}}$$

Application to the Poisson equation on a 2D grid:

$$\Phi_{ij}(n+1) = \frac{1}{4}\left(\Phi_{i+1,j}(n) + \Phi_{i-1,j}(n)\right. \tag{134}$$

$$\left. + \Phi_{i,j+1}(n) + \Phi_{i,j-1}(n) - b_{ij}\right) \tag{135}$$

Important here: formular is recursive, each $\Phi_{ij}(n+1)$ depends on all previous $\Phi_{ij}(n)$. That means that one copy of $\Phi_{ij}(n)$ has to be kept in memory!!

10.4 Gauss-Seidel

$$\vec{\Phi}(t+1) = (D+U)^{-1}\left(\vec{b} - L\vec{\Phi}(t)\right) \tag{136}$$

Error evolution operator is

$$\Lambda = (D+U)^{-1}L \tag{137}$$

(**Note**: U is now in the "denominator", hence there is a faster convergence!) → The largest EV of Λ becomes smaller.

Stopping criteria

$$\delta = \frac{\left|\left|\vec{\Phi}(t+1) - \vec{\Phi}(t)\right|\right|}{(1-\lambda)\left|\left|\vec{\Phi}(t)\right|\right|} \leq \epsilon \tag{138}$$

The update scheme of Gauss-Seidel does not require backups and just overwrites old data(compare script p.116):

$$\Phi_i(t+1) = -\frac{1}{a_{ii}}\left(\sum_{j=i+1}^{N} a_{ij}\Phi_j(t) + \sum_{j=1}^{i-1} a_{ij}\Phi_j(t+1) - b_j\right)$$

10.4.1 Gauss-Seidel-Error

$$\begin{aligned}\delta(t+1) &= A^{-1}\vec{b} - (D+U)^{-1}(\vec{b} - L\vec{\Phi}(t)) \\ &= -(D+U)^{-1}L\underbrace{A^{-1}\vec{b} - \vec{\Phi}(t)}_{\delta(t)} \\ &= -\underbrace{-(D+U)^{-1}L}_{\Lambda}\delta(t)\end{aligned}$$

Method is faster and needs less memory space since the old matrix can be overwritten.

10.4.2 Successive Over-Relaxation

Improve convergence even further with an over-relaxing parameter $1 \leq \omega \leq 2$:

$$\boxed{\vec{\Phi}(t+1) = (D+\omega U)^{-1}\left(\omega b + [(1-\omega)D - \omega L]\,\vec{\Phi}(t)\right)} \tag{139}$$

Denominator is increased with ω yielding faster convergence. For $\omega = 1$ we obtain Gauss-Seidel again.
The parameter ω must not be pushed too far, otherwise we risk blow-up.

10.5 Gradient methods

- use functionals measuring the error of a solution of a system of equations
- unique solution $\leftrightarrow$ function is paraboloid with minimum a exact solution.
- functional defined by**residual** $\vec{r}$

Define residual (something like the error estimate) as

$$\boxed{\vec{r} = A\vec{\delta} = A(A^{-1}b - \Phi) = b - A\vec{\Phi}} \tag{140}$$

Minimize functional

$$\mathcal{J} = \vec{r}^{\,T}A^{-1}\vec{r} = \begin{cases} 0, & \text{if } \Phi = \Phi_0 \\ > 0, & \text{else} \end{cases} \tag{141}$$

Substitute $\vec{r}$:

$$\mathcal{J} = (\vec{b} - A\vec{\Phi})^T A^{-1}(\vec{b} - A\vec{\Phi}) = b^T A^{-1} b + \Phi^T A\Phi - 2\,\vec{b}\,\Phi$$

Let Φ_i be the ith approximation and define $\boxed{\vec{\Phi} \to \vec{\Phi} + \alpha\,\vec{d}}$ ($\vec{d}$ is the **direction of descent**) and minimize functional $\mathcal{J}$ with respect to α. Calculate

$$\frac{\partial \mathcal{J}}{\partial \alpha} = 2\vec{d}_i^{\,T}(\bar{\alpha}_i A\vec{d}_i - \vec{r}) = 0$$

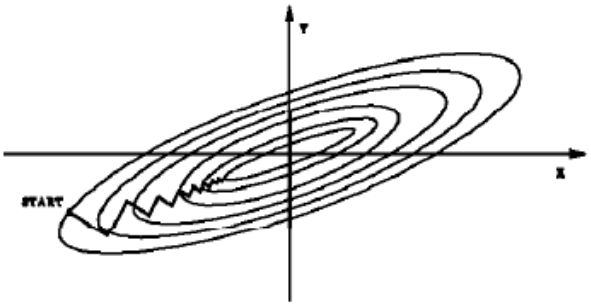
Optimal value:

$$\boxed{\bar{\alpha}_i = \frac{\vec{d}_i^{\,t}\vec{r}_i}{\vec{d}_i^{\,t}A\vec{d}_i}} \tag{142}$$

$\bar{\alpha}_i$ needs to be computed in each step and is different for different methods.

Requirements for the gradient method: Matrix should be **SPD** (used as a scalar product in CG)

10.5.1 Steepest descent method



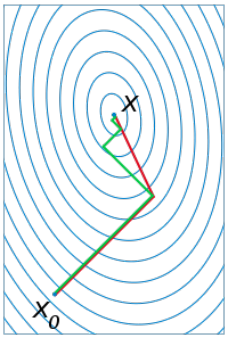
Pick the direction $\vec{d}$ in the direction of steepest descent, where gradient is minimal (largest negative gradient). In this case this is the direction of the residuum $\vec{r}$.

1. Choose $\vec{d}_i = \vec{r}_i = \vec{b} - A\vec{\Phi}_i$
2. Evaluate and store $\vec{u}_i = A\vec{r}_i$
   Needs N^2 operations for N equations, N if A is sparse.
3. Calculate length of the step $\alpha_i = \frac{\vec{r}_i^{\,t}\vec{r}_i}{\vec{r}_i \underbrace{\vec{u}_i}_{=A\vec{r}_i}}$
4. Advance $\vec{\Phi}_{i+1} = \vec{\Phi}_i + \alpha_i\vec{r}_i$
5. Update residual: $\vec{r}_{i+1} = \vec{r}_i + \alpha_i\vec{u}_i$
6. Iterate until error (use residuum to estimate error $\vec{r} = A\vec{\delta}$) is less than required precision

Problem: Steepest descent is not necessarily the shortest path to the minimum. Improve method with conjugate gradients.

10.5.2 Conjugate Gradient method

Takes functional, deforms it such that it looks like a regular paraboloid and performs steepest descent.
The new direction is chosen **conjugate** (orthogonal) to all previous direction.



Use Gram-Schmidt orthogonalization procedure to produce a direction $\vec{d}_i$ that is conjugate for all previous $\vec{d}_j$ (using A as the metric, $\vec{d}_i A \vec{d}_j = \delta_{ij}$):

$$\boxed{\vec{d}_i = \vec{r}_i - \sum_j \frac{\vec{d}_j A \vec{r}_i}{\vec{d}_j A \vec{d}_j} \vec{d}_j} \tag{143}$$

1. Initialize Φ_1 and $r_1 = \vec{b} - A\vec{\Phi}_1$, $\vec{d}_1 = \vec{r}_1$
2. Calculate temporary scalar $c = \frac{1}{\vec{d}_i^T A \vec{d}_i}$
3. Compute length of the step $\alpha_i = c\vec{r}_i^T \vec{d}_i$
4. Advance $\vec{\Phi}_{i+1} = \vec{\Phi}_i + \alpha \vec{d}_i$
5. Calculate new residual $\vec{r}_{i+1} = \vec{b} - A\vec{\Phi}_{i+1}$ and stop if residuum is smaller than required precision: $|r|^2 < \epsilon$
6. Compute direction of the next step: $\vec{d}_{i+1} = \vec{r}_{i+1} - \left(c\vec{r}_{i+1} A \vec{d}_i\right) \vec{d}_i$
7. Repeat

10.5.3 Biconjugate Gradient methods

Might be alternative when matrix is not SPD, use two residuals in parallel:

$$\vec{r} = \vec{b} - A\vec{\Phi} \quad \vec{d}_1 = \vec{r} \tag{144}$$

$$\tilde{\vec{r}} = \vec{b} - A^T\vec{\Phi} \quad \tilde{\vec{d}}_1 = \tilde{\vec{r}} \tag{145}$$

10.6 Effort for linear matrix equation solvers

The algorithms are especially efficient for sparse matrices, since the computing time is

$$\text{sparse NxN matrix-vector product} \Rightarrow \mathcal{O}(N) \tag{146}$$

$$\text{full NxN matrix-vector product} \Rightarrow \mathcal{O}(N^2) \tag{147}$$

The Gaussian procedure computing time is roughly $\mathcal{O}(N^3)$ for a $N \times N$ matrix.

10.7 Preconditioning

Numerical problems if a matrix is badly conditioned, i.e. diagonal elements are approximately the sum of the other elements in a row.

Find some matrix P^{-1} that is a good approximation of the inverse of the matrix A:

$$P^{-1}A\vec{\Phi} = P^{-1}\vec{b} \tag{148}$$

Jacobi-Preconditioner: Use diagonal elements of A for P, inverse is easy.

$$P_{ij} = A_{ij}\delta_{ij} \;\Rightarrow\; P_{ij}^{-1} = \delta_{ij}\frac{1}{A_{ij}}$$

SOR TODO

10.8 Multigrid Procedure

Dynamically switch between different resolutions to obtain better results.

11 Finite element method

11.1 Strategy of finite elements

- Use a dynamic grid (typically with triangulation), which is finer at critical points and coarser when function is well-behaving
- Instead of using a discrete grid introduce basis functions

11.2 Difference to finite differences

- Finite difference are easier to implement because of very regular grid, but is restricted to handle shapes and models with rectangular geometry
- Finite elements can handle complicated geometries and boundaries with relative ease
- FEM are so to speak a generalization of the FDM.

Example: Poisson equation in 1D, Dirichlet boundary conditions.

$$\frac{\mathrm{d}^2\Phi}{\mathrm{d}x^2}(x) = -4\rho(x), \quad \Phi(x)|_\gamma = 0$$

Expand Φ in terms of localized basis functions:

11.3 Basic idea and basis functions

Expand field function in terms of the localized basis functions:

$$\Phi(x) = \sum_i^\infty a_i u_i(x) \approx \Phi_N(x) = \sum_i^N a_i u_i(x) \tag{149}$$

and take only a **finite number of basis functions.**

Obtain expansion coefficients by introduction of so-called **weight-function** $w_i(x)$

$$\sum_{i=1}^n a_i \underbrace{\left(-\int_0^L \frac{\mathrm{d}^2 u_i}{\mathrm{d}x^2} w_j(x)\,\mathrm{d}x\right)}_{A_{ij}} = \underbrace{4\pi \int_0^L \rho(x) w_j(x)\,\mathrm{d}x}_{b_j}$$

for $j \in \{1, \ldots, N\}$

11.3.1 Galerkin method

Choose

$$\boxed{w_j(x) = u_j(x)}$$

We obtain N coupled equations and for a sensible chosen basis the derivatives of the basis fucntion need not to be found numerically.

$$A_{ij} = -\int_0^L u_i''(x) w_j(x)\,\mathrm{d}x = \int_0^L u_i'(x) w_j'(x)\,\mathrm{d}x$$
$$b_j(x) = 4\pi \int_0^L \rho(x) w_j(x)\,\mathrm{d}x$$
$$A\vec{a} = \vec{b}$$

Take hat basis functions

$$u_i(x) = \begin{cases} \frac{(x - x_{i-1})}{\Delta x} & \text{for } x \in [x_{i-1}, x_i] \\ \frac{(x - x_{i-1})}{\Delta x} & \text{for } x \in [x_i, x_{i+1}] \\ 0 & \text{otherwise} \end{cases} \tag{150}$$

and get for $\Delta x = x_i - x_{i-1}$

$$A_{ij} = \int_0^L u_i'(x) u_j'(x)\,\mathrm{d}x = \begin{cases} 2/\Delta x, & \text{if } i = j \\ -1/\Delta x, & \text{if } i = j+1 \\ 0, & \text{else} \end{cases}$$

Non-homogeneous Dirichlet BC: $\Phi(0) = \Phi_0$, $\Phi(L) = \Phi_1$
Use the following decomposition:

$$\Phi_N(x) = \frac{1}{L}(\Phi_0(L - x) + \Phi_1 x + \sum_{i=1}^{N} a_i u_i(x))$$

11.3.2 Non-Linear PDEs

Example:

$$\Phi(x)\frac{\mathrm{d}^2\Phi}{\mathrm{d}x^2}(x) = -4\pi\rho(x)$$
$$\Rightarrow \int_0^L \left[\Phi(x)\frac{\mathrm{d}^2\Phi}{\mathrm{d}x^2} + 4\pi\rho\right] w_k(x)\,\mathrm{d}x = 0$$

and we obtain the following set of equations:

$$\sum_{i,j} A_{ijk} a_i a_j = b_k, \quad A_{ijk} = \int_0^l u_i(x) u_j''(x) w_k\,\mathrm{d}x$$

Use **Picard Iteration**:

1. Initial guess Φ_0
2. Solve Linear Equation

$$\Phi_0(x)\frac{\mathrm{d}^2\Phi_1}{\mathrm{d}x^2}(x) = -4\pi\rho(x)$$

3. Iterate

$$\Phi_n\frac{\mathrm{d}^2\Phi_{n+1}}{\mathrm{d}x^2}(x) = 4\pi\rho(x)$$

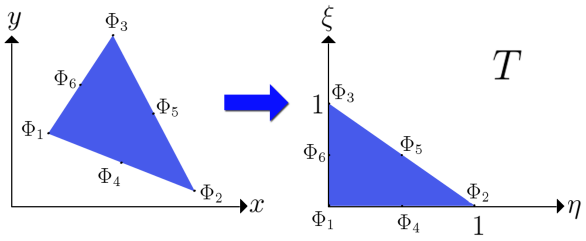
11.3.3 Basis function in higher dimension

- 2D: triangle of a triangulation for instance: piecewise continuous polynomials
- Linearization: $\Phi(r) \approx c_1 + c_2 x + c_3 y$
- Paraboloid: $\Phi(r) \approx c_1 + c_2 x + c_3 y + c_4 x^2 + c_5 xy + c_6 y^2$ $\rightarrow$ smooth transitions between elements are possible

2D Standard Form (Reference Element):

$$x = x_1 + (x_2 - x_1)\xi + (x_3 - x_1)\eta$$
$$y = y_1 + (y_2 - y_1)\xi + (y_3 - y_1)\eta$$
$$\eta = \frac{(y - y_1)(x_2 - x_1) - (x - x_1)(y_2 - y_1)}{D}$$
$$\xi = \frac{(x - x_1)(y_3 - y_1) - (y - y_1)(x_3 - x_1)}{D}$$
$$D = (y_3 - y_1)(x_2 - x_1) - (x_3 - x_1)(y_2 - y_1)$$

Coordinate Transformation



$$\nabla_x \Phi = \left(\frac{\partial\Phi}{\partial x}, \frac{\partial\phi}{\partial y}\right)$$
$$\rightarrow \nabla_\xi \Phi = \left(\frac{\partial\Phi}{\partial\xi}\frac{\partial\xi}{\partial x} + \frac{\partial\Phi}{\partial\eta}\frac{\partial\eta}{\partial x}, \frac{\partial\Phi}{\partial\xi}\frac{\partial\xi}{\partial y} + \frac{\partial\Phi}{\partial\eta}\frac{\partial\eta}{\partial y}\right)$$

Also transfer integration area $G_j \rightarrow T$ with T being the reference triangle.

$$\det J = \frac{\partial x}{\partial\xi}\frac{\partial y}{\partial\eta}\frac{\partial x}{\partial\eta}\frac{\partial y}{\partial\xi} = D$$

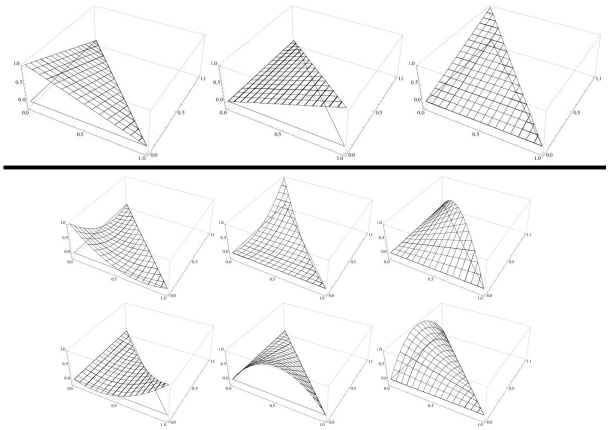
and the transformed integral become

$$\int\int_{G_j} \Phi_x^2 + \Phi_y^2\,\mathrm{d}x\,\mathrm{d}y)$$
$$= \int\int_T \left(c_1\Phi_\xi^2 + 2c_2\Phi_\xi\Phi_\eta + c_3\Phi_\eta^2\right)\mathrm{d}\eta\,\mathrm{d}\xi$$
$$c_1 = \frac{(y_3 - y_1)^2}{D} + \frac{(x_3 - x_1)^2}{D}$$
$$c_2 = \frac{(y_3 - y_1)(y_2 - y_1)}{D} + 2\frac{(x_3 - x_1)(x_2 - x_1)}{D}$$
$$c_3 = \frac{(y_2 - y_1)^2}{D} + \frac{(x_2 - x_1)^2}{D}$$

Finally we obtain the following description of the basis functions on the triangle:

Linear: determined by having value 1 in one corner and 0 in the other two corners:

$$N_1 = 1 - \xi - \eta$$
$$N_2 = \xi$$
$$N_3 = \eta$$



Quadradtiv: 6 points on each triangle, value 1 in one point, 0 in the five others.

$$N_1 = (1 - \xi - \eta)(1 - 2\xi - 2\eta)$$
$$N_2 = \xi(2\xi - 1)$$
$$N_3 = \eta(2\eta - 1)$$
$$N_4 = 4\xi(1 - \xi - \eta)$$
$$N_5 = 4\xi\eta$$
$$N_6 = 4\eta(1 - \xi - \eta)$$

Shorthand notation:

$$\Phi(\xi, \eta) = \sum_{i=1}^{6} \phi_i N_i(\xi, \eta) = \vec{\phi}\vec{N}(\xi, \eta)$$

11.3.4 Variational Approach

Basic idea: Minimization of

$$E = \int\int_G \left(\frac{1}{2}(\nabla\Phi)^2 + \frac{1}{2}a\Phi^2 + b\Phi\right) \mathrm{d}x\,\mathrm{d}y + \int_\Gamma \left(\frac{a}{2}\Phi^2 + \beta\Phi\right) \mathrm{d}s$$

where the first integral is the volume and the second is the surface contribution

Variation:

$$\delta E = \int\int_G (\nabla\Phi\delta\Phi + a\Phi\delta\Phi + b\delta\Phi)\,\mathrm{d}x\,\mathrm{d}y + \int_G (a\Phi\delta\Phi + \beta\delta\Phi)\,\mathrm{d}s$$

Greens Theorem:

$$\int\int_G \nabla u \nabla v\,\mathrm{d}x\,\mathrm{d}y = -\int\int_G v\Delta u\,\mathrm{d}x\,\mathrm{d}y + \int_\Gamma \frac{\partial u}{\partial n}\,\mathrm{d}s$$

such that the Variation becomes:

$$\delta E = \int\int_G (-\Delta\Phi + a\Phi + b)\,\delta\Phi\,\mathrm{d}x\,\mathrm{d}y = \int_\Gamma \left(\alpha\Phi + \beta + \frac{\partial\Phi}{\partial n}\right)\delta\Phi\,\mathrm{d}s = 0$$

2 different cases for Volume part $\Delta\Phi = a\Phi + b$

- $a = 0$: Poisson Equation
- $b = 0$: Helmholtz Equation

Rewrite Volume Term

$$E = \sum_{j\text{Elemente}} \int\int_{G_j} \left((\nabla\Phi)^2 + a\Phi^2 + b\Phi\right)\mathrm{d}x\,\mathrm{d}y$$
$$\Leftrightarrow E = \vec{\Phi}^T A\vec{\Phi} + b\vec{\Phi}$$

and it follows

$$\frac{\partial E}{\partial\Phi} = 0 \quad \Rightarrow \quad A\Phi + b = 0$$

11.4 Time-dependent

Semidiscretization: Kep time continuous and discretize space,
→ obtain set of coupled ODEs and evolve along time line

11.4.1 Method of Lines

Example:

$$\frac{\partial T}{\partial t}(\vec{x}, t) = \frac{\kappa}{c\rho}\Delta T(\vec{x}, t) + \frac{1}{c\rho}W(\vec{x}, t)$$

where T local temperature, c specific heat, ρ density(homogeneous), κ thermal conductivity (const), W sinks and sources.

Step formulation in 2D:

$$T(x_{ij}, t + \Delta t) = T(x_{ij}, t) + \frac{\kappa\Delta t}{C\rho\Delta x^2}\tilde{T}_{ij}(t) + \frac{\Delta t}{c\rho}W(x_{ij}, t)$$
$$\tilde{T}_{ij}(t) = T(x_{i+1,j}, t) + T(x_{i-1,j}, t) + T(x_{i,j+1}, t) + T(x_{i,j-}$$

which corresponds to explicit Euler (forward Euler) for the time step

Stability

- $\Delta x, \Delta t$ must be chosen carefully (especially too big Δt leads to blow up)
- e.g. if prefactor $\frac{\kappa\Delta t}{C\rho\Delta x^2} \geq \frac{1}{4}$: The last summand of $\tilde{T}_{ij}$ cancels the leading contribution or courses negative-positive oscillations of temperature.

Courant-Friedrich-Levvy(CFL) stability condition

$$\frac{\kappa\Delta t}{c\rho\Delta x^2} < \frac{1}{4}$$

11.4.2 Crank-Nicolson-method

$$T(x,t+\Delta t) = T(x,t) + \frac{\kappa \Delta t}{2c\rho}\left(\Delta T(x,t+\Delta T(x,t+\Delta t))\right) + \frac{\Delta t}{2c\rho}\left(W(x,t)+w(x,t+\Delta t)\right)$$

(Implicit method of second order)

Define for $n \in \{1,\dots,L^2\}$:
$\boxed{\vec{T}(t) = (T(x_n,t)) \quad \vec{W}(t) = (W(x_n,t))}$ and the **discretized Laplace operator**

$$OT(x_n,t) = \frac{\kappa \Delta t}{c\rho \Delta x^2}\left(T(x_{n+1},t) + T(x_{n-1}) + T(x_{n+L},t) + T(x_{n-L},t) - 4T(x_n,t)\right)$$

and rewrite the method equation:

$$T(x,t+\Delta t) = T(x,t) + \frac{1}{2}\left(OT(x,t) + OT(x,t+\Delta t)\right) + \frac{\Delta t}{2c\rho}\left(W(x,t) + W(x,t+\Delta t)\right)$$

Now sort with respect to time

$$(2\cdot\mathbf{1} - O)\vec{T}(t+\Delta t) = (2\cdot\mathbf{1}+O)\vec{T}(t) + \frac{\Delta t}{c\rho}(\vec{W}(t) + \vec{w}(t+\Delta t))$$

with $B = (2\mathbf{1} - O)^{-1}$ we arrive at the *formal solution*

$$\boxed{\vec{T}(t+\Delta t) = B\left[(2\mathbf{I}+O) + \frac{\Delta t}{c\rho}(\vec{W}(t) + \vec{W}(t+\Delta t))\right]}$$

This lead to an **tridiagonal matrix** which can be solved in $\mathcal{O}(n)$

11.5 Spectral Methods

basis function are globally smooth (FE: locally)
Consider an PDE given by a Differential operator L e.g. $L = \left(\frac{\partial}{\partial t} - \frac{\partial}{\partial x}\right)$:

$$Lu(x,t) = f(u(x,t))$$
$$u(0,t) = u_B \quad u(x,0) = u_1(x)$$

Expansion in terms of basis function ϕ_i

$$u(x,t) = \sum_{i=1}^{\infty} a_i(t)\phi_i(x) \approx u_N(x) = \sum_{i=1}^{N} a_i(t)\phi_i(x)$$

Pick N (orthogonal) test function w_j:

$$w_j(x) = \begin{cases} \phi_j(x), & \text{Galerkin} \\ \delta(x-x_j), & \text{Collocation} \end{cases}$$

and obtain the following eqautions

$$\boxed{\int_0^1 \left(Lu(x,t) + f(u(x,t))\right) w_j(x)\,\mathrm{d}x\,\mathrm{d}t = 0, \quad j = 1,\dots,N}$$

Other possible orthgonal systems besides fourier

- Legendre $[-1,1]$
- Chebychev $[-1,1]$
- Laguerre $[0,\infty)$
- Hermite $(-\infty,\infty)$

11.6 Finite Volume Method

TODO