EuroMinSci
A EUROCORES PROGRAMME
EUROPEAN SCIENCE FOUNDATION COLLABORATIVE RESEARCH

Introduction into numerical modelling

- **Paul Bons**
- **Institut für Geowissenschaften**
- **Eberhard Karls Universität Tübingen**

EuroMinSci
A EUROCORES PROGRAMME
EUROPEAN SCIENCE FOUNDATION COLLABORATIVE RESEARCH

Contents

What is a model?

- **Mannequin, photo model**
- **(Small) Example for an artwork or building**
- **Wooden or gypsum design to make a cast**
- **Example for an artist (person, fruit tray)**
- **Example, fixed design (the Citroën DX was a successful model)**
- **Small-scale imitation (model airplane)**
- **Exemplary person (e.g. Ghandi, Mandela)**

- **Empirical interpretation of a mathematical-logical system**

(Dutch van Dale dictionary)

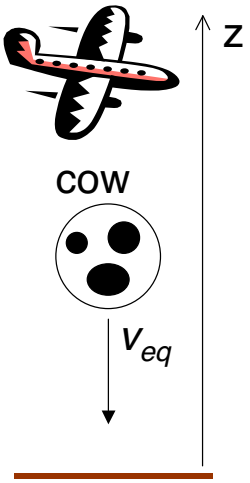
My view of a "model"

- **A simplification (interpretation) of a system or sub-system defined by set of (mathematical) rules**
  - physical
  - chemical
  - Geometrical

Example: cow falling from an airplane

- model cow: sphere with density ρ_{cow}
- equilibrium velocity: Stoke's law

$$v_{eq} = \frac{2g\rho_{cow}R_{cow}^2}{9\eta_{air}}$$



What is a simulation?

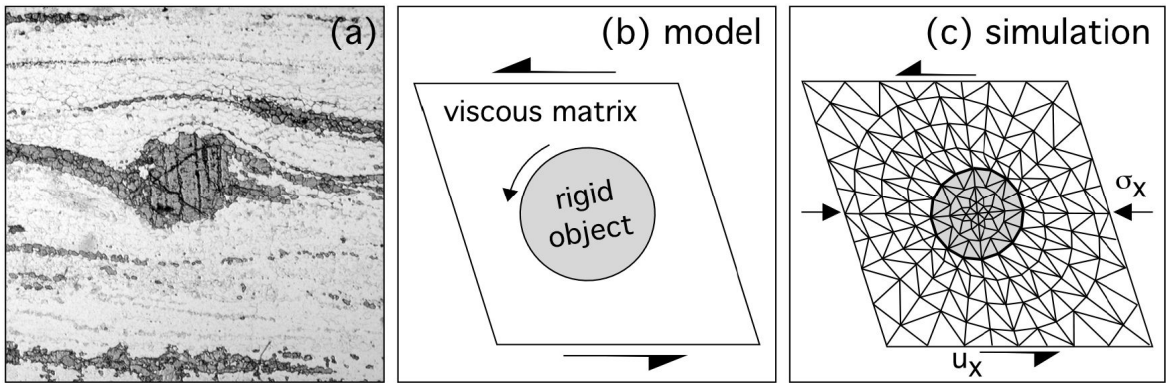
Dictionary

sim•u•late |ˈsimyəˌlāt|
verb [trans.]
imitate the appearance or character of : *red ocher intended to simulate blood* |
[as adj.] (**simulated**) *a simulated leather handbag.*
• pretend to have or feel (an emotion) : *it was impossible to force a smile, to simulate pleasure.*
• produce a computer model of : *future population changes were simulated by computer.*

DERIVATIVES
sim•u•la•tion |ˌsimyəˈlā SH ən| noun
sim•u•la•tive |-ˌlātiv| adjective

ORIGIN mid 17th cent.: from Latin ***simulat- 'copied, represented,'*** from the verb ***simulare***, from ***similis 'like.'***

Reality - model - simulation



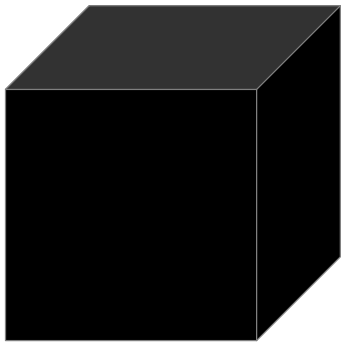
- **The development of the winged porphyroclast can be simulated with the FEM program BASIL**
  - First level model: A high-viscosity object in a low-viscosity matrix, subjected to stress/strain rate boundary conditions
  - Second level model: Regions of the material can be "modelled" as polygons with certain physical properties. Stress and strain are governed by continuum-mechanical laws

Why do modelling?

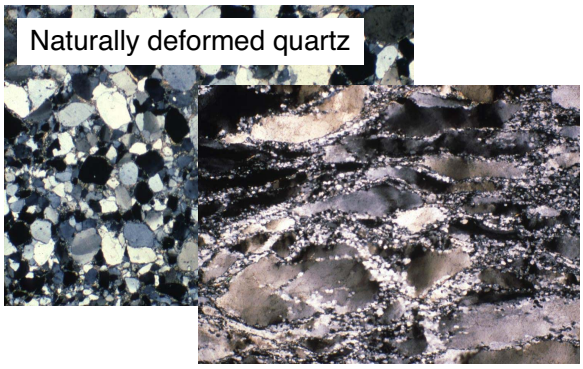
- **Popular - quick publications**
- **For the sake of it**
  - repeat what we know with a new method, preferably with nice computer graphics
- **To gain knowledge**
  - to test hypotheses
  - to develop hypotheses
  - to determine/constrain values of parameters that are difficult to measure in nature or in experiments

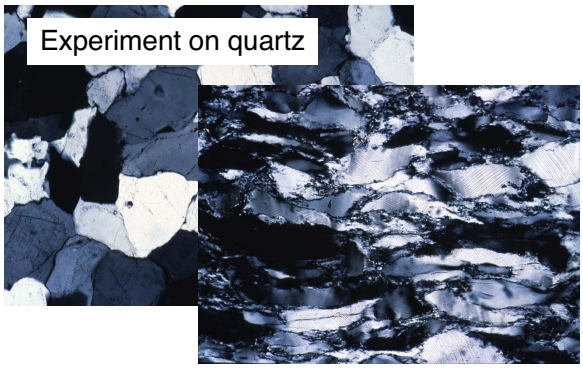
Advantages of numerical modelling

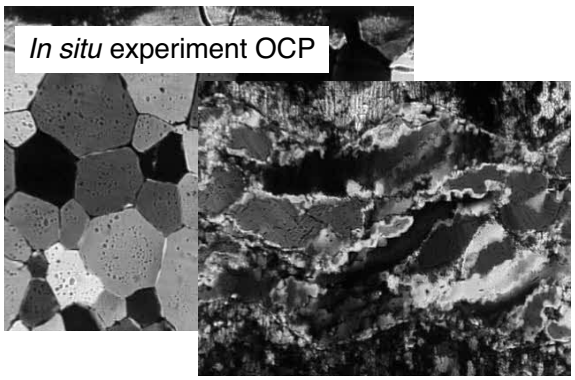
- **We know exactly what we do**
- **Hypothesis → set of rules**
  - Only these rules apply
  - nothing else
- **What are the results?????**
  - confirm/reject hypothesis
  - learn about the system

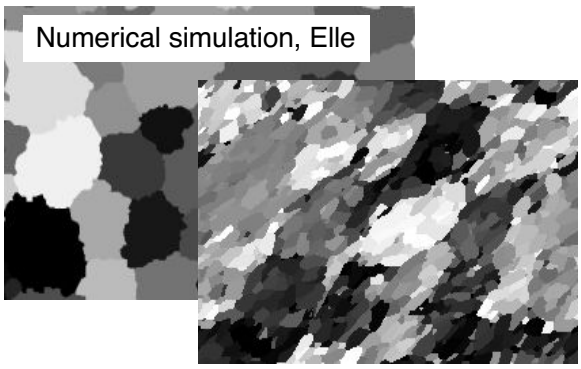


Simulation of microstructures









Why model microstructures?

- **To find out how microstructures develop**
  - how do crystallographic preferred orientations form by growth competition in veins?
- **To find out what microstructures mean**
  - what do strain fringes tell us on the kinematics of deformation?
  - How much information is lost by static recrystallisation?
- **To determine the properties of rocks**
  - Will strain localise or not?

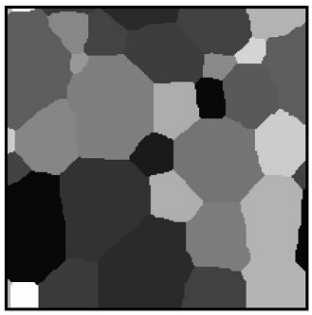
Ingredients of a model

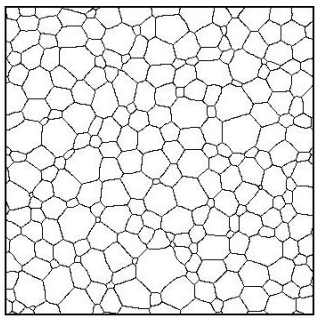
MODEL

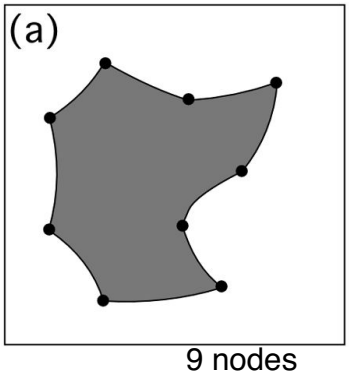
- **Description of the system**
  - Regular grid, polygonal elements, etc.
- **Variables and parameters**
  - Independent variables
  - State / dependent variables
  - Physical parameters
- **Equations**
  - State equations
  - Evolution equations
- **Boundary & initial-value conditions**

Description of the system

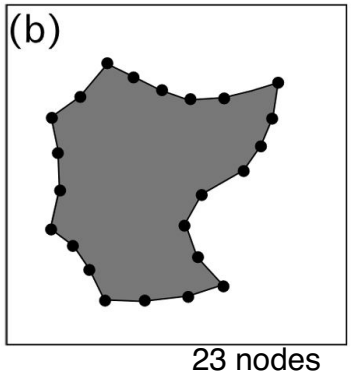
- **Main types**
  - Regular grid
  - Polygonal elements



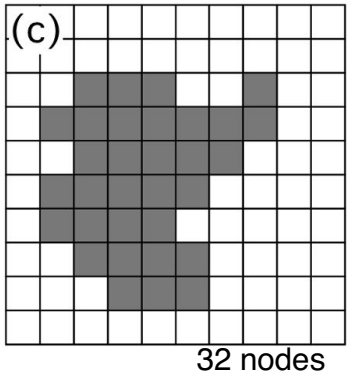




9 nodes

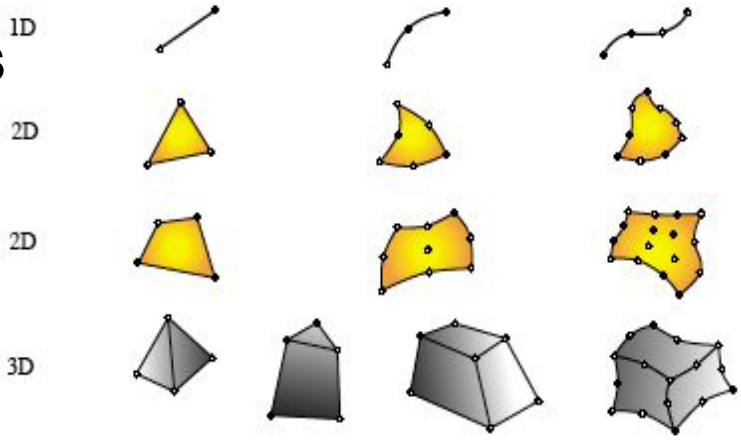


23 nodes



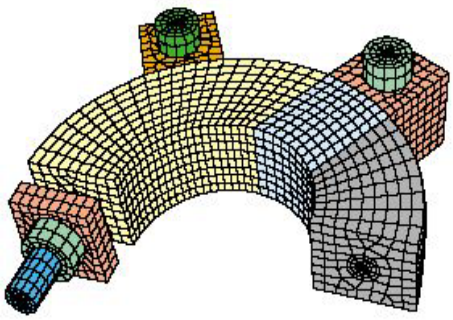
32 nodes

Elements

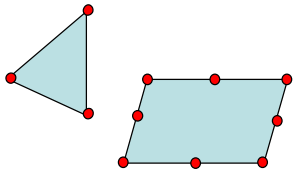


- **Elements are defined by nodes**
- **Linked by straight or curved boundary segments**
- **Elements are disjoint = non-overlapping**
- **The nodes have two roles:**
  - store the values of the state variables
  - define the shape of the elements

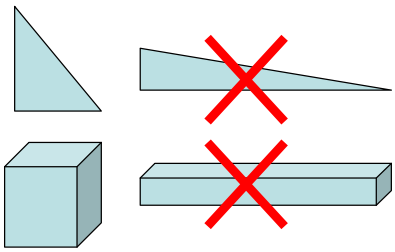
Example of FEM-mesh in structural model of machine part



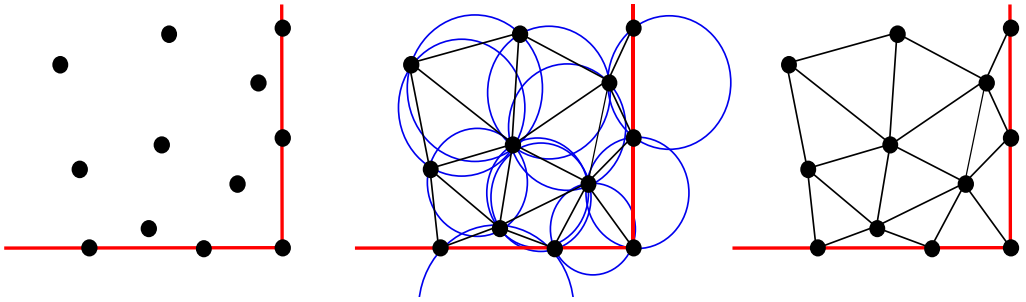
- **Many (FEM-) programs come with a mesh-generator**



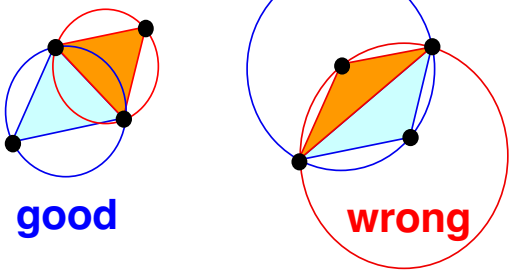
- **Optimal mesh:**
  - Least number of elements (nodes)
  - Element shape & number of nodes adapted to complexity of function
  - Preferably low aspect ratio:
    no sharp points



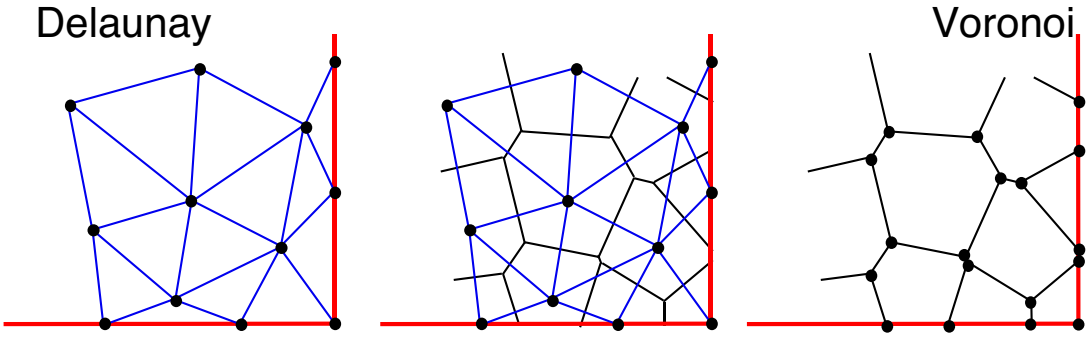
Mesh generation 1: Delaunay triangulation



- Each circle going through 3 nodes of a triangles does not contain any more nodes



Mesh generation 2: Voronoi polygons



- Lines perpendicular to sides of triangles form polygons:

  Voronoi polygons
- Each polygon contains one node
- Any point inside the polygon is closer to that node than to any other node

Variables and parameters

- **Independent variables**
  - Base of the system
  - Time, space (x,y,z)
- **State / dependent variables**
  - Define state of system or parts of system
  - Dislocation density, stress, strain rate
- **Physical parameters**
  - Boltzman's constant, Avagadro number
  - Lattice parameters, Young's modulus

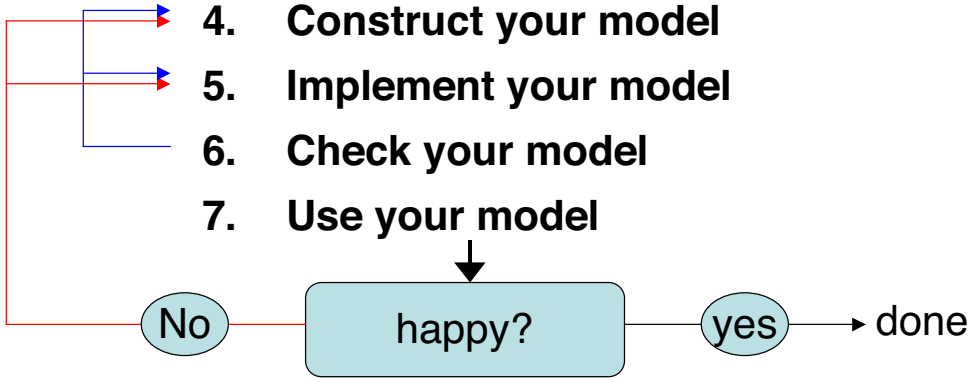
EuroMinSci
A EUROCORES PROGRAMME
EUROPEAN SCIENCE FOUNDATION COLLABORATIVE RESEARCH

Equations

- **State equations**
  - Path-independent equations that change state variables
  - Steady-state flow law: $\varepsilon = A \cdot \sigma^3$
  - Darcy's law: $J = k \cdot \partial P / \partial x$
- **Evolution equations**
  - Path-dependent equations that change state variables
  - Diffusion: $C_{t+1} = C_t + f(C_t)$
  - Special case: kinematic equations
    - Equations that change positions of particles or points in system

Modelling made easy in 7 steps

1. **What is your problem?**
2. **What do you want?**
3. **How does it work?**
4. **Construct your model**
5. **Implement your model**
6. **Check your model**
7. **Use your model**



Step 1: *what is your problem?*

- **Identify the problem**
- **Hypothesis to investigate**
  - Recrystallisation modifies a crystallographic preferred orientation (CPO)? (YES/NO)
- **Gain insight**
  - What does a CPO modified by recrystallisation look like?
- **Measurement to be made**
  - At what grain boundary mobility rate does recrystallisation modify a CPO?

EuroMinSci
A EUROCORES PROGRAMME
EUROPEAN SCIENCE FOUNDATION COLLABORATIVE RESEARCH

Step 2: *what do you want?*

- **Identify the input and output**

- **What data do you have as input?**
  - temperature
  - material properties
  - initial values
- **What do you want to get out of the model?**
  - a number or a graph
  - a picture or a movie
- **How precise do you want the results?**
  - Determines how much simplification is allowed

EuroMinSci
A EUROCORES PROGRAMME
EUROPEAN SCIENCE FOUNDATION COLLABORATIVE RESEARCH

Step 3: *how does it work?*

- **Identify the processes and mechanisms**

- **Information from experiments or theory?**
  - Darcy's flow of fluid
  - Dislocation creep
  - Diffusion
  - Elasto-viscous deformation

- **Any experience from similar problems?**
  - Trace element diffusion very similar to heat conduction

Step 4: *construct your model*

- **What tools do you need?**
  - Finite-difference method, FEM, front tracking, etc.
  - Can you use existing models & methods?

- **What are the assumptions and boundary conditions?**
- Assumptions:
  - Usually simplifications that enable certain equations etc. to be used
- Boundary conditions:
  - Literally, what happens at the boundaries of your system?
  - Also, what is the starting situation?

Step 5: *implement your model*

- **How do you turn your model into a program?**
  - Is there an analytical solution? → **do it!**
  - Otherwise, write/choose your numerical solution

- **Can you use/modify existing models?**
  - Use commercial software (FLAC, etc.)
  - Use freely available software (OOF, ELLE, Excel)
  - Or write new code?

- **How will you handle output?**
  - Display, data storage, data analysis?

Step 6: *check your model*

- **Tune and validate your model → is it correct?!**
  - sensitivity and stability analysis
  - limits of validity, application and resolution
  - compare with data
  - compare with benchmark experiments
  - compare with other models
  - calibrate

Step 7: *use your model*

EuroMinSci
A EUROCORES PROGRAMME
EUROPEAN SCIENCE FOUNDATION COLLABORATIVE RESEARCH

Main modelling techniques Chapter 2 of *The Book*

- **Monte Carlo techniques**
  - Forest fire model
  - Ising & Potts models
  - Lattice-gas/Boltzmann models → read The Book
- **Finite Difference**
- **Finite Elements**

- **Front/boundary tracking**
- **Phase field models**

- **Lattice Springs → Daniel Koehn, or read The Book**

EuroMinSci
A EUROCORES PROGRAMME
EUROPEAN SCIENCE FOUNDATION COLLABORATIVE RESEARCH

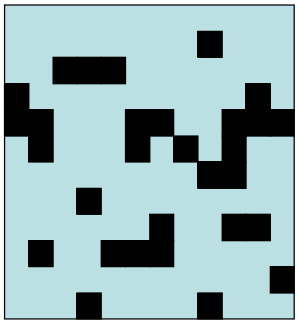
Monte Carlo methods

Forest fire
Ising model
Potts models

EuroMinSci
A EUROCORES PROGRAMME
EUROPEAN SCIENCE FOUNDATION COLLABORATIVE RESEARCH

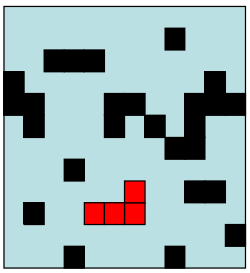
The forest fire model

- **A Monte Carlo model to simulate distribution of forest fires over time**
- **Trees are points (*x,y*) on a lattice *L***
  - $L_{(x,y)}$=1: "TREE" (black)
  - $L_{(x,y)}$=0: "NO TREE" (white)
- **Simulation of growth of forest**
- **Each time step:**
  - **Randomly select a coordinate *x,y***
    - If $L_{(x,y)}$="NO TREE", plant a tree ($L_{(x,y)}$→"TREE")

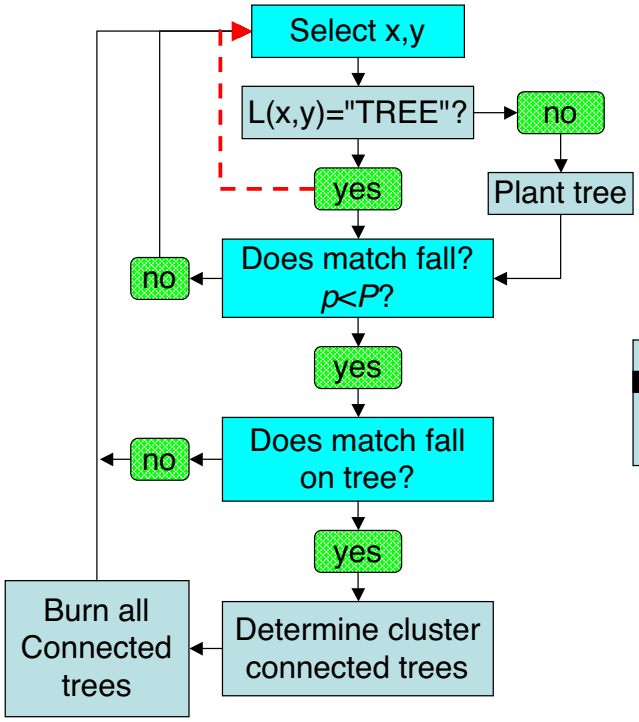


The forest fire model

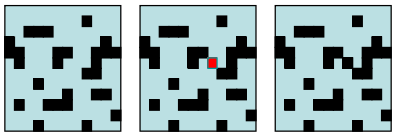
- **Simulation of forest fires**
  - Every time step there is a chance P<<1 that a match falls the area
  - If match falls on a tree, it burns and all its neighbours, and their neighbours, etc.
- **Each time step:**
  - **Randomly select a coordinate *q,r***
  - **Take random number *p* between 0 and 1**
  - If p<P and $L_{(q,r)}$="TREE"
  - Switch $L_{(q,r)}$ and all connecting neighbours with $L_{(x,y)}$="TREE" to "NO TREE"



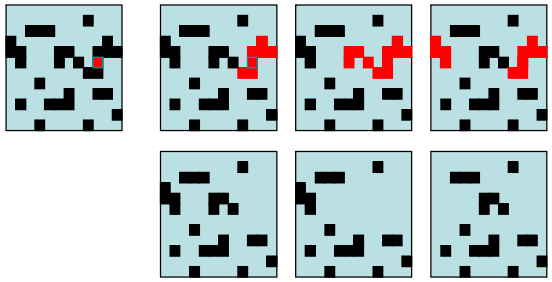
Flow diagram



- **Planting a tree**



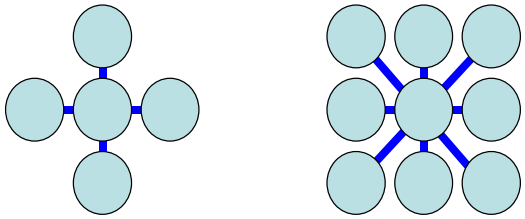
- **A forest fire**

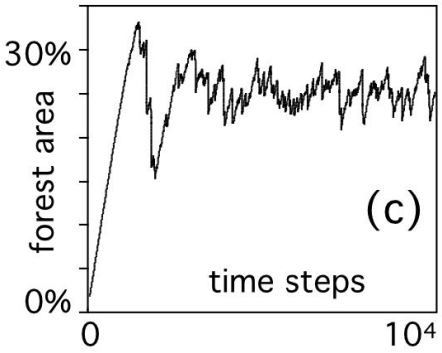


- Different possibilities

The forest fire model

- **neighbouring:**
  Von Neumann **& Moore**





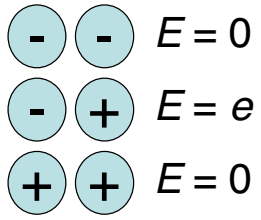
The Ising model for magnetisation

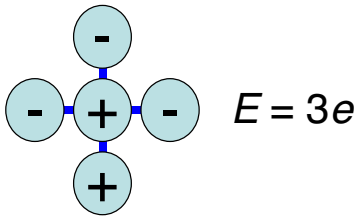
- **Magnetisation of a particle depends on the "spin" of the electrons, which can be + or -**
- **Each particle has two states +1 or -1**
- **Lowest energy state is where all particles are either +1 or -1**
  - boundaries between different spins have certain energy
- **Spins can reverse (flip) to other state**
  - a thermally activated process (T-dependent)

The “Hamiltonian”

- Each pair of particles has a certain energy *E*
- Each particle has a certain energy state depending on the spins of its neighbours
- For example
  $\delta_{ij} = 0$ when $spin_{(i)} = spin_{(j)}$
  $\delta_{ij} = e$ when $spin_{(i)} \neq spin_{(j)}$

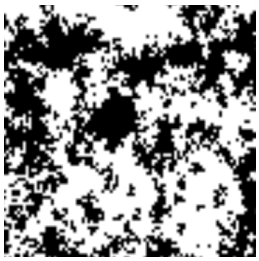
$$E_i = \sum_j \left(\delta_{ij} e\right)$$





von Neumann neighbouring

Structure of Ising model



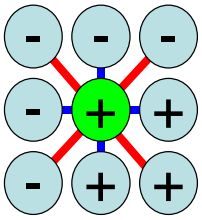
- Particles are nodes on regular grid
- Random select one particle
- If particle has different spin than at least one of the neighbours,
- calculate current energy (E_0) and energy if particle would flip to other state (E_1)
- The chance that the particle will change state depends on $\Delta E = E_1 - E_0$

loop

One implementation of Ising model

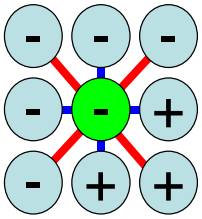
- **Hamiltonian (9-point):**

$$E_i = \sum_j \left(\delta_{ij} e\right)_{direct} + \frac{1}{\sqrt{2}} \sum_j \left(\delta_{ij} e\right)_{diagonal}$$



$E_0 = 4.12$

$\Delta E = -1.41$



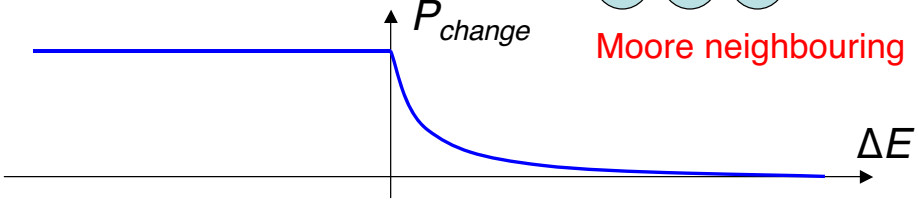
$E_1 = 2.71$

Moore neighbouring

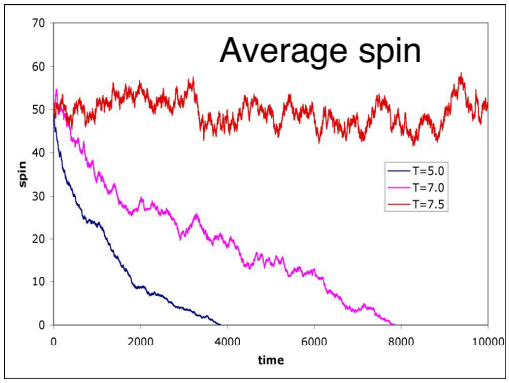
- P_{change} **rule:**

if $\Delta E \leq 0$ $P_{change} = 1$

if $\Delta E > 0$ $P_{change} = \exp^{\frac{\delta \sum E}{kT}}$



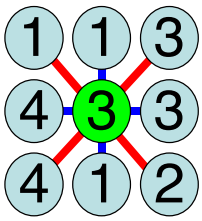
Ising model results



temperature

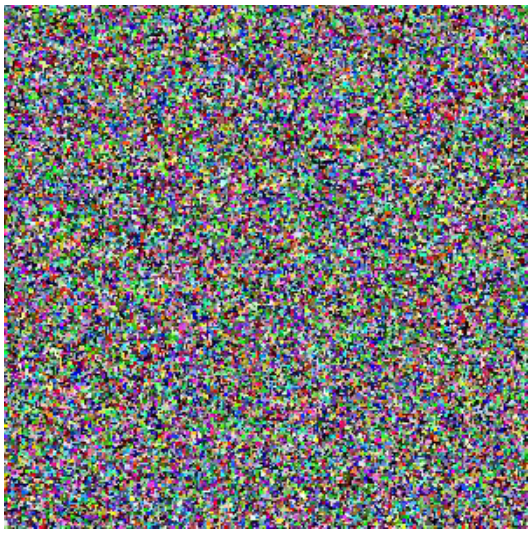
$T \ll T_{crit}$ $T < T_{crit}$ $T > T_{crit}$

q-state Potts models



q = 4

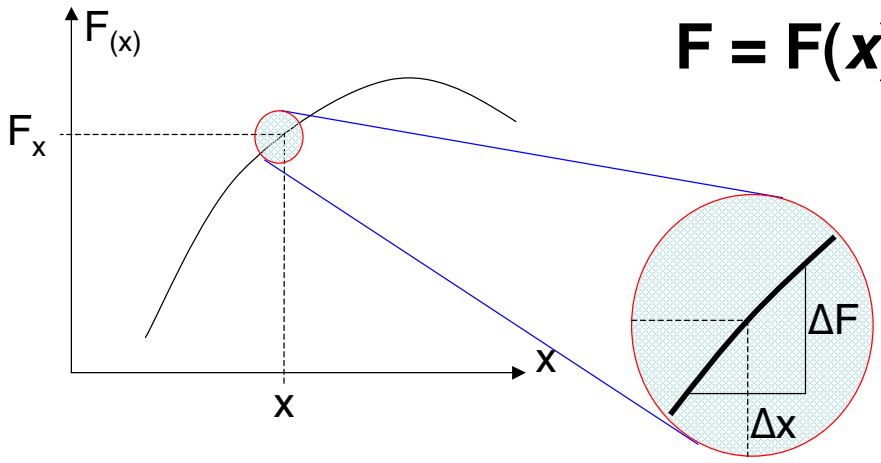
- **Same as Ising model, but each particle can have *q* states**
- **e.g. *q*=4: {1,2,3,4}**
- **Example of grain growth simulation with *q*=255**
- **Problem:**
  - Lattice effect
  - Scaling difficult



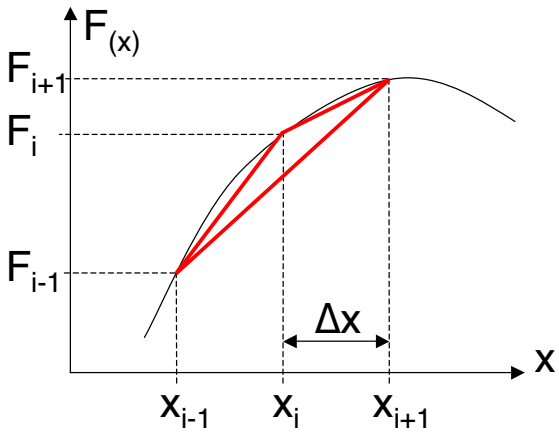
Finite difference method

The finite difference method

- **Most processes can be described by one or more (a set of) differential equations:**
  - Fick's law: $J = -D \cdot \partial C/\partial x$
  - Heat flow: $\partial T/\partial t = \kappa \partial^2 T/\partial x^2$
  - Deformation: $v_x = \partial p_x/\partial t = \partial p_x/\partial x \cdot x + \partial p_x/\partial y \cdot y$
- **We want to discretise such functions to handle them numerically**



- **The derivative of function F(*x*) to x is the rate of change of F with a change of *x***
- **$\partial F/\partial x = f(x) = \Delta F/\Delta x$ when limit $\Delta x \downarrow 0$**



F = F(x)

- **discrete approximations**
  - forward: $\frac{\partial F}{\partial x} \approx \frac{F_{x+1} - F_x}{\Delta x}$
  - backward: $\frac{\partial F}{\partial x} \approx \frac{F_x - F_{x-1}}{\Delta x}$
  - central: $\frac{\partial F}{\partial x} \approx \frac{F_{x+1} - F_{x-1}}{2\Delta x}$

Each approximation is wrong, except when $\Delta x \downarrow 0$

Choose small Δx

Compromise with calculation time & memory use

EuroMinSci
A EUROCORES PROGRAMME
EUROPEAN SCIENCE FOUNDATION COLLABORATIVE RESEARCH

Higher-order differentials

- **Higher order differentials can be discretised the same way**
- **For example: $\partial^2 F/\partial x^2$:**

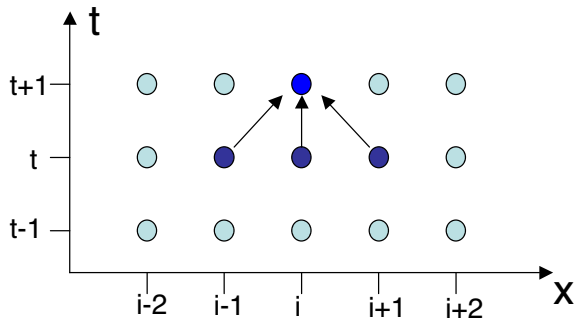
$$\frac{\partial^2 F}{\partial x^2} = \frac{\partial\left(\frac{\partial F}{\partial x}\right)}{\partial x} \approx \frac{\left(\frac{F_{i+1} - F_i}{\Delta x}\right) - \left(\frac{F_i - F_{i-1}}{\Delta x}\right)}{\Delta x}$$

$$= \frac{F_{i+1} - 2F_i + F_{i-1}}{(\Delta x)^2}$$

Modelling processes in time

- **For example, diffusion: $\partial C/\partial t = D \cdot \partial^2 C/\partial x^2$**

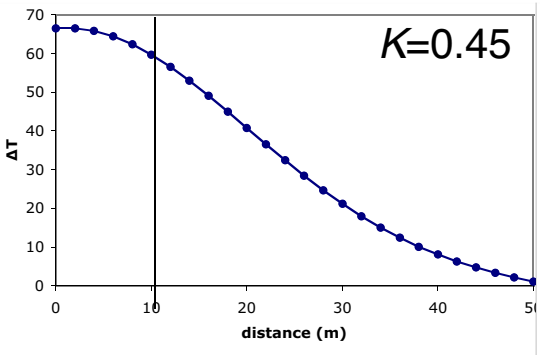
$$C_i^{t+1} \;\delta\; C_i^t + D\frac{\sum^2 C_i^t}{\sum x^2}\Delta t \;\delta\; C_i^t + D\frac{C_{i+1}^t - 2C_i^t + C_{i-1}^t}{(\Delta x)^2}\Delta t$$

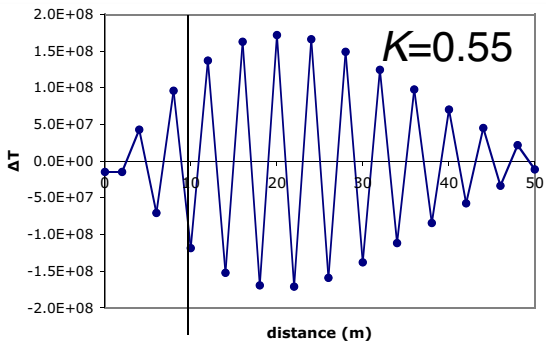


This method is called

Explicit

Instability: oscillations





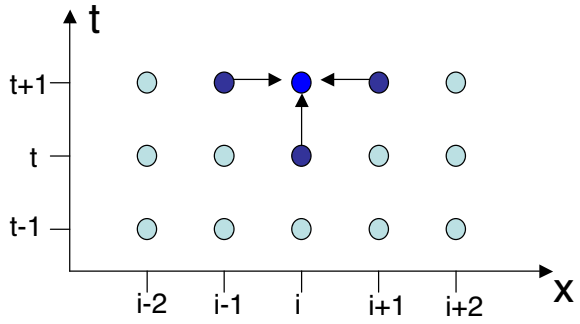
$$T_i^{t+1} \approx T_i^t + \left(\frac{\kappa\Delta t}{(\Delta x)^2}\right)\cdot\left(T_{i+1}^t - 2T_i^t + T_{i-1}^t\right) \qquad K = \frac{\kappa\Delta t}{(\Delta x)^2}$$

- **The system breaks down (instability), when $K \geq 0.5$**
- **Therefore, make sure that $K < 0.5$**

Modelling processes in time

- **Again, diffusion: $\partial C/\partial t = D \cdot \partial^2 C/\partial x^2$**

$$C_i^{t+1} = C_i^t + D\frac{\sum^2 C_i^{t+1}}{\sum x^2}\Delta t \;\delta\; C_i^t + D\frac{C_{i+1}^{t+1} - 2C_i^{t+1} + C_{i-1}^{t+1}}{(\Delta x)^2}\Delta t$$



This method is called

Implicit

Implicit method: how to solve?

$$C_i^{t+1}\;\sum C_i^t + K\left(C_{i+1}^{t+1}\;\delta\; 2C_i^{t+1} + C_{i\delta 1}^{t+1}\right) \qquad K = \frac{\kappa\Delta t}{(\Delta x)^2}$$

- **To determine C_i^{t+1}, we already need to know what it will be....**
- **Suppose we have *N* points on our grid**
- **This means we have *N* unknown variables (C_i^{t+1})**
- **If we can find *N* linear equations, we can solve for these unknown variables**

A set of *N* linear equations

$$C_i^{t+1} \approx C_i^t + K\left(C_{i+1}^{t+1} - 2C_i^{t+1} + C_{i-1}^{t+1}\right)$$

$$C_i^{t+1} \approx C_i^t + K \cdot C_{i+1}^{t+1} - 2K \cdot C_i^{t+1} + K \cdot C_{i-1}^{t+1}$$

- For each point, we have one equation → *N* equations
- Therefore we can solve for these *N* unknown variables
- But there are two problem cases on the left and right C_1 and C_N:

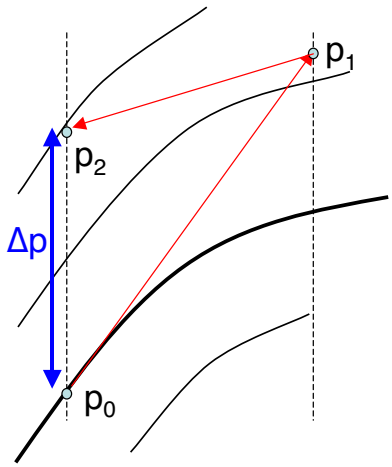
$$C_1^{t+1} \; \sum C_1^t + K\,\Delta C_2^{t+1} \; \delta \; 2K\,\Delta C_1^{t+1} + K\,\Delta C_0^{t+1}$$

$$C_N^{t+1} \approx C_N^t + K \cdot C_{N+1}^{t+1} - 2K \cdot C_N^{t+1} + K \cdot C_{N-1}^{t+1}$$

- We have to give the model the two values at the limits of our model
- These are the **boundary conditions**

Other methods

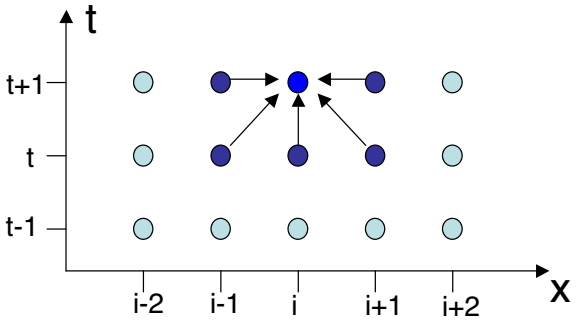
- **Various methods have been developed, based on the principles described before**
- **Iterative**
  - go forward $p_0 \rightarrow p_1$
  - then backwards $p_1 \rightarrow p_2$
  - use $\Delta p = p_2 - p_0$ to improve estimate of p_1



Other methods

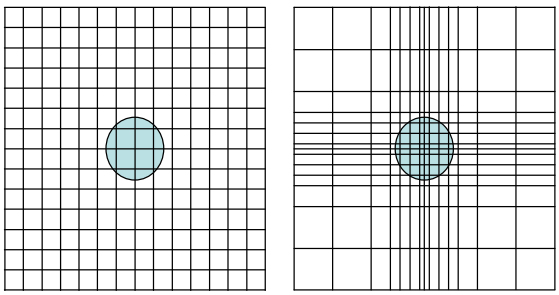
- **Crank-Nicholson scheme**
  - Combine implicit and explicit method

$$C_i^{t+1} \approx C_i^t + \tfrac{1}{2}K\left(C_{i+1}^t - 2C_i^t + C_{i-1}^t\right) + \tfrac{1}{2}K\left(C_{i+1}^{t+1} - 2C_i^{t+1} + C_{i-1}^{t+1}\right)$$



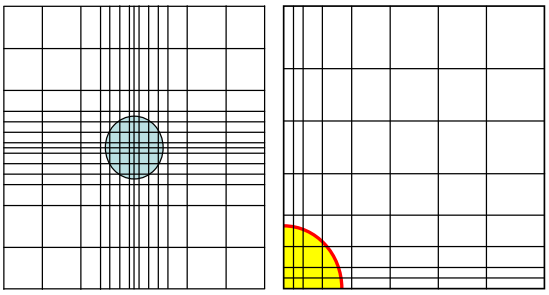
Variable grid spacing

- **Necessary resolution may vary in space**
- **Use variable grid spacing, with dense grid in areas with strong gradients**

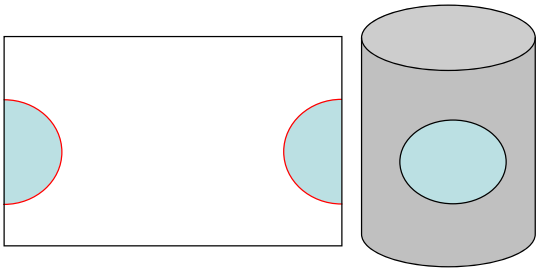


$$\frac{\partial^2 F}{\partial x^2} \approx \frac{\left(\frac{F_{i+1} - F_i}{\Delta x_{(i\ldots i+1)}}\right) - \left(\frac{F_i - F_{i-1}}{\Delta x_{(i-1\ldots i)}}\right)}{\frac{1}{2}\left(\Delta x_{(i\ldots i+1)} + \Delta x_{(i-1\ldots i)}\right)}$$

Selecting boundaries



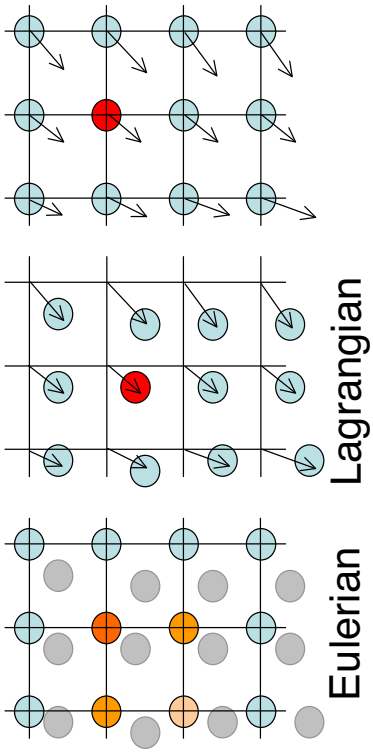
- **Use symmetry to select your system**



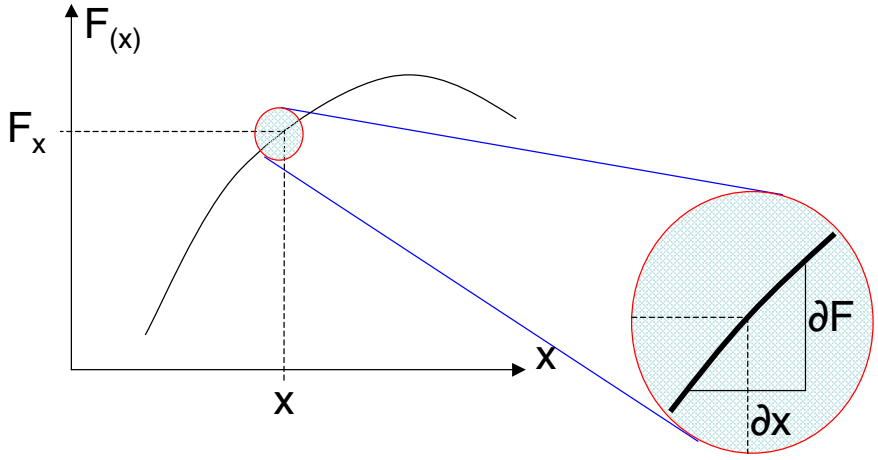
- **Sometimes it is possible to avoid boundaries by having wrapping boundaries**

Numerical diffusion

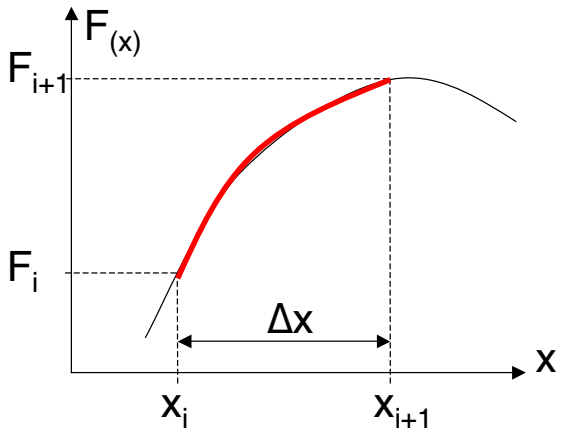
- In the FD description values of variables are stored on grid nodes
  - velocity, chemical composition, etc.
- these are properties of a material particle at each grid point
- properties in between grid points are interpolated
- If material is moving, the properties should move along with the material particle
- This can lead to "numerical diffusion" if the particle is not moved to exactly another grid node



Finite element method



- **Finite Difference Method**
- **Using**
  $\partial F/\partial x = f(x) = \Delta F/\Delta x$ when limit $\Delta x \downarrow 0$
- **Basic assumption of FDM is that sections of a curve can be approximated with a linear function**
- **when Δx is small enough**



- **It is also possible to use other functions**
  - e.g. polynomials $F_{(x)} = a + b \cdot x + c \cdot x^2$
- **Distance between nodes (Δx) may be larger**

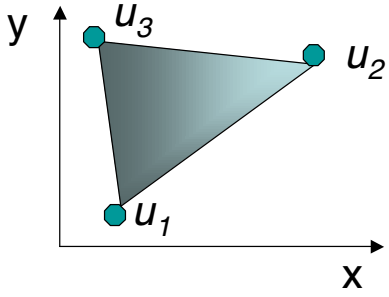
  = First essential of FEM

The Finite Element Method

- **Second characteristic of FEM:**
  - **Divide the system in elements of finite size**
- **For each element assume a function* to describe the values of the state variables**
- **Solve all functions, taking into account**
  - the state- and kinematic equations
  - continuity conditions between elements
  - the boundary equations

* **Linear function FEM ≈ implicit FDM**

Finding the solution



true function: $u = \kappa_{(f)}$

trial function: $u = k_{(f)}$

Minimise error $\sum(\kappa_{(f)} - k_{(f)})^2)_i \downarrow 0$

- **Each node *i* has number of state variables or degrees of freedom (u_i): e.g. velocity**
- **These are a continuous function of *x* & *y* and "forces" (*f*) acting on the system: e.g. stress**

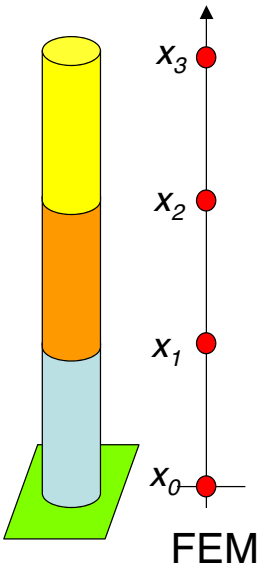
Forces and state variables

Application	State variables (*u*)	Force (*f*)
solid mechanics	**displacement or velocity**	**mechanical forces**
heat conduction	**temperature**	**heat flux**
chemical dispersion	**trace element concentration**	**diffusion**

Finding the solution, continued

- for each node we can write set of equations: $k_i \cdot u_i = f_i$
- Combining all equations + boundary equations, we get: $K_{ij} \cdot u_i = f_j$
- K_{ij} is often called the stiffness matrix
- But we know f_j and want to know u_i:
- Solution: $u_i = K^{-1}_{ij} \cdot f_j$

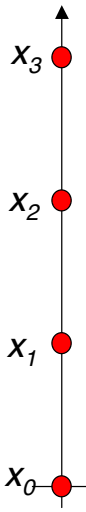
Example with 3 elements



FEM model

- Three elastic cylinders on top of each other
- How much would each shorten under the weight of the cylinders on top?
- 1-dimensional 3-element model
- Each cylinder has
  - same length L_0
  - the same elasticity coefficient E
  - and variable mass W_i
- Basic law: $\left(\frac{\Delta \sum L}{L_0}\right) \delta E = F$

Example with 3 elements



$$\frac{(\Delta x_3 - \Delta x_2)\cdot E}{L_0} = L_0 g\left(\tfrac{1}{2}W_3\right) \qquad \Delta x_3 e - \Delta x_2 e = L_0 g\left(\tfrac{1}{2}W_3\right)$$

$$\frac{(\sum x_2\ \delta\ \sum x_1)\,\Delta E}{L_0} = L_0 g\left(\tfrac{1}{2}W_2 + W_3\right) \qquad \Delta x_2 e - \Delta x_1 e = L_0 g\left(\tfrac{1}{2}W_2 + W_3\right)$$

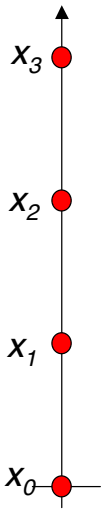
$$\frac{\Delta x_1 \cdot E}{L_0} = L_0 g\left(\tfrac{1}{2}W_1 + W_2 + W_3\right) \qquad \Delta x_1 e = L_0 g\left(\tfrac{1}{2}W_1 + W_2 + W_3\right)$$

$$use: \left(\frac{E}{L_0}\right) = e$$

$$\Delta x_0 = 0$$

EuroMinSci
A EUROCORES PROGRAMME
EUROPEAN SCIENCE FOUNDATION COLLABORATIVE RESEARCH

Reorganise the three equations



$$e\cdot\Delta x_1 + 0\cdot\Delta x_2 + 0\cdot\Delta x_3 = L_0 g\left(\tfrac{1}{2}W_1 + W_2 + W_3\right)$$

$$-e\cdot\Delta x_1 + e\cdot\Delta x_2 + 0\cdot\Delta x_3 = L_0 g\left(\tfrac{1}{2}W_2 + W_3\right)$$

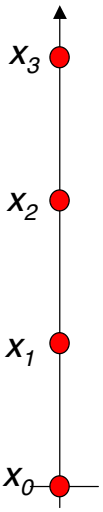
$$0\cdot\Delta x_1 - e\cdot\Delta x_2 + e\cdot\Delta x_3 = L_0 g\left(\tfrac{1}{2}W_3\right)$$

- **Write as vector-function**

$$\begin{pmatrix}\Delta x_1\\ \Delta x_2\\ \Delta x_3\end{pmatrix}\begin{pmatrix} e & 0 & 0\\ -e & e & 0\\ 0 & -e & e\end{pmatrix} = \begin{pmatrix} L_0 g\left(\tfrac{1}{2}W_1 + W_2 + W_3\right)\\ L_0 g\left(\tfrac{1}{2}W_2 + W_3\right)\\ L_0 g\left(\tfrac{1}{2}W_3\right)\end{pmatrix}$$

$$u_i \cdot K_{ij} = f_j$$

Reorganise the three equations



- **We have:** $u_i \cdot K_{ij} = f_j$

$$\begin{pmatrix} \sum x_1 \\ \sum x_2 \\ \sum x_3 \end{pmatrix} \begin{pmatrix} e & 0 & 0 \\ \delta e & e & 0 \\ 0 & \delta e & e \end{pmatrix} = \begin{pmatrix} L_0 g\left(\frac{1}{2}W_1 + W_2 + W_3\right) \\ L_0 g\left(\frac{1}{2}W_2 + W_3\right) \\ L_0 g\left(\frac{1}{2}W_3\right) \end{pmatrix}$$

- **We need:** $u_i = K^{-1}{}_{ij} \cdot f_j$
- **Use:** $K \cdot K^{-1} = I \quad \Leftrightarrow \begin{pmatrix} e & 0 & 0 \\ -e & e & 0 \\ 0 & -e & e \end{pmatrix} \begin{pmatrix} e & 0 & 0 \\ e & e & 0 \\ e & e & e \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$
- **Giving:** $\begin{pmatrix} \delta \sum x_1 \\ \sum x_2 \\ \sum x_3 \end{pmatrix} \begin{pmatrix} \delta e & 0 & 0 \\ e & e & 0 \\ e & e & e \end{pmatrix} = \begin{pmatrix} \delta L_0 g\left(\frac{1}{2}W_1 + W_2 + W_3\right) \\ L_0 g\left(\frac{1}{2}W_2 + W_3\right) \\ L_0 g\left(\frac{1}{2}W_3\right) \end{pmatrix}$

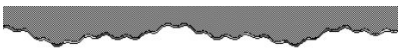
Boundary/front tracking

Principles

Implementation in Elle

Boundary tracking

- **In some cases, we are only interested in boundaries between regions**
  - Grain boundaries
  - Subgrain boundaries



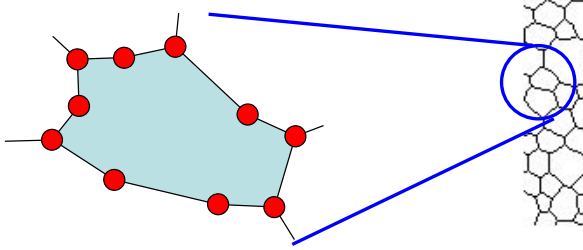
- **No need to bother about the interior**
  - Only track what the **boundaries** or **fronts** do

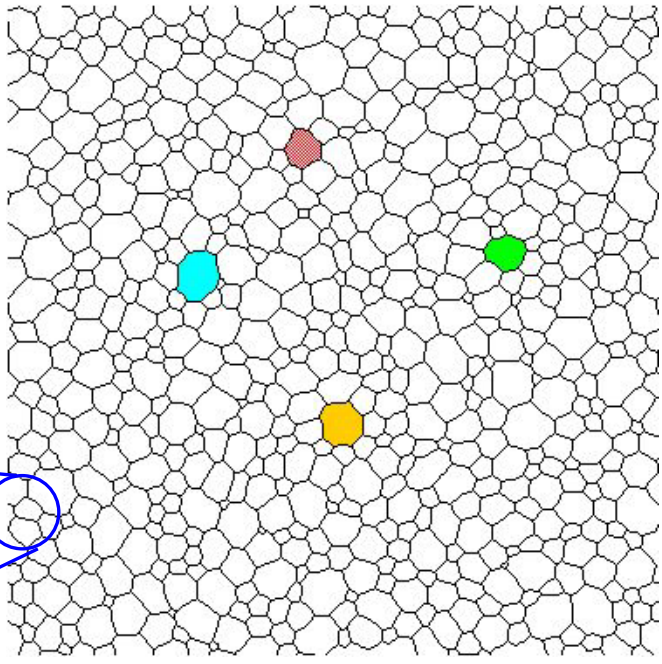
- **Typical application in microstructures**
  - Dynamic recrystallisation
  - Static grain growth
  - Metamorphic reactions
  - Partial melts

Boundary tracking (in Elle)

- **Boundaries are defined by nodes that link boundary segments**

- **To change the microstructure:**
  Move the nodes

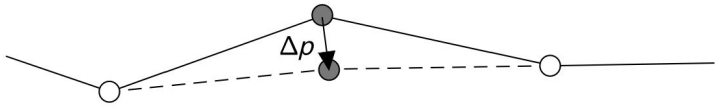




Boundary movement = evolution equations

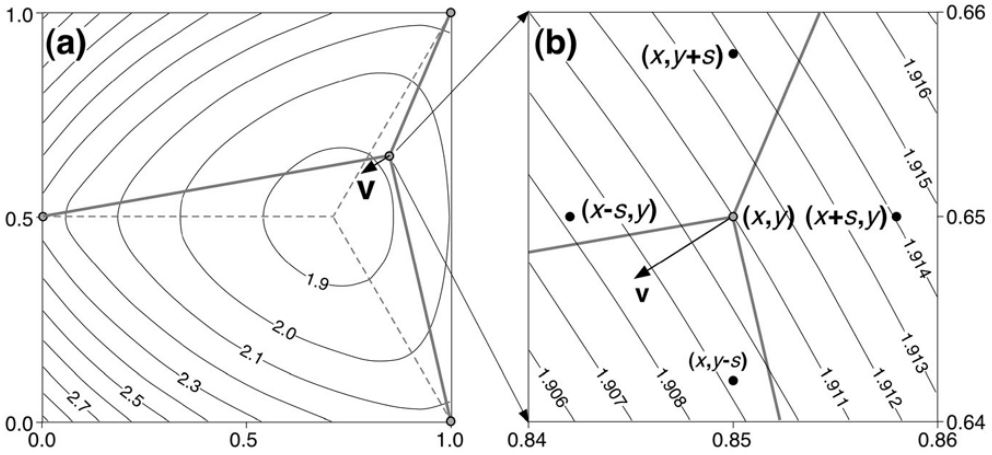
- **The evolution equations that determine node movement can be anything**
  - Monte Carlo
  - Finite Elements
  - Finite Difference, etc.

- **Basic movement routine in Elle:**
  - Finite difference
  - Central
  - Explicit
  - Work done to move boundary = free energy gained



$$f_{gbm} = \frac{-dE}{dp} \Leftrightarrow f_{gbm} v = \frac{-dE}{dt} = \frac{dW}{dt}$$

Finding gradient in free energy



- **A node has a certain energy as a function of position**
- **Map the energy field: find the gradient**
- **Central finite difference: use 4 trial positions**

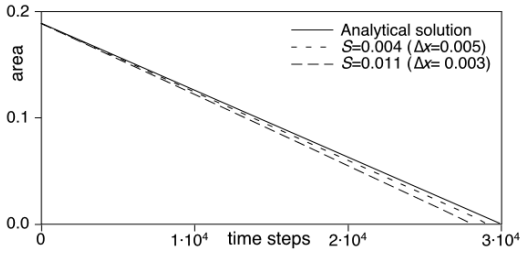
$$\Leftrightarrow |v| = \frac{-2(dE/du)}{\sum \frac{L_i \sin^2(\alpha_i)}{m_i}}.$$

(see Ch. 3.5)

Advantages of approach

$$|v| = \frac{-2(dE/du)}{\sum \frac{L_i \sin^2(\alpha_i)}{m_i}} \qquad p_{t+\Delta t} = p_t + v\Delta t$$

- ***E* in *dE/du* can be anything:**
  - Surface energy
  - Strain energy (dislocations, etc)
  - Chemical energy: metamorphic reactions
- **Energy balance is maintained**
  - Except for errors in explicit finite difference approximation

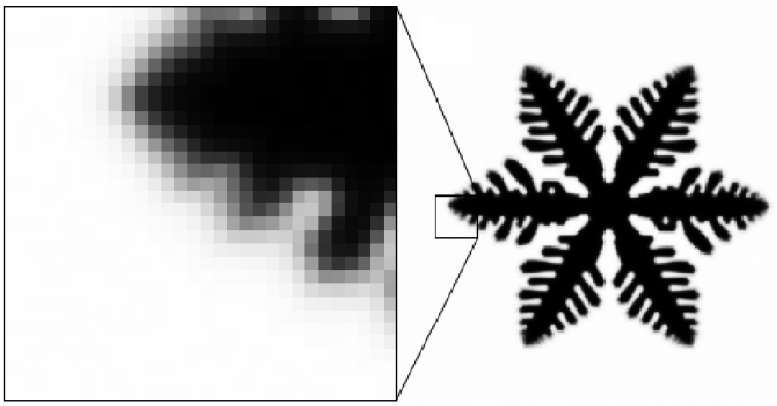


$$S = \frac{\delta m \sum t}{\left(\sum x\right)^2} << 0.01$$

Phase field modelling

Phase field modelling

- **So far, regions had discrete boundaries**
  - Grain boundaries
  - Phase boundaries
- **Why not make the boundaries fuzzy?**
  - We can use "simple" differential equations to solve what is happening (FEM, FD, Monte Carlo)



Rapidly growing crystal:

Crystal has sharp boundary

But: growth rate determined by heat diffusion: gradient field

State of a phase and evolution

- **Each point in the system has certain "state"**
  - With a certain energy (*W*)
- **System evolves to reduce *W* for each point**
  - Obeying rules (mass/energy conservation, diffusion rates, etc.)

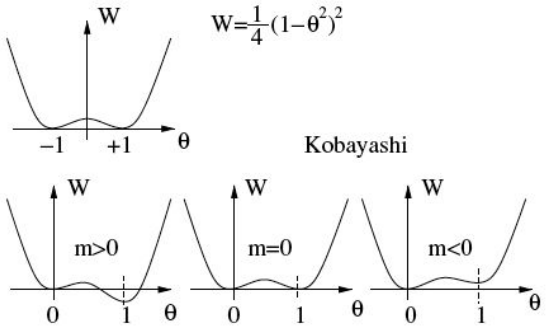


Figure 1. Schematic view of the Landau–Ginzburg free-energy functions. The top graph represents the function used for multiphase flows while the bottom graphs correspond to Kobayashi's prescription for dendritic growth.

Results (Biben paper)

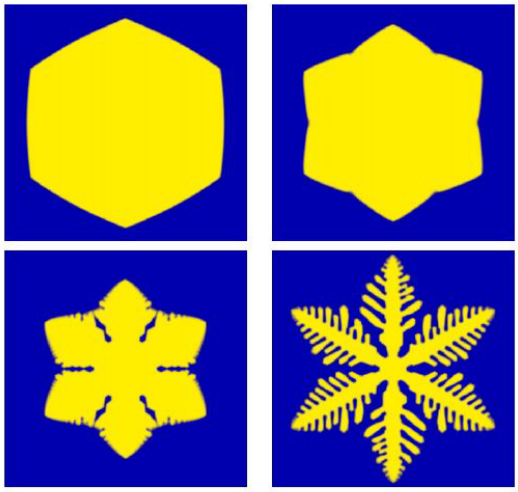


Figure 2. Different growth regimes as obtained from Kobayashi's model. The four images correspond to a dimensionless latent heat $K = 0.8, 1.0, 1.2$ and 1.8 from left to right and top to bottom; the dimensionless relaxation time is $\tau = 3 \times 10^{-4}$. The values of the more technical parameters are specified at the beginning of the 'C' program listed in the appendix.

INSTITUTE OF PHYSICS PUBLISHING — EUROPEAN JOURNAL OF PHYSICS
Eur. J. Phys. **26** (2005) S47–S55 — doi:10.1088/0143-0807/26/5/S06

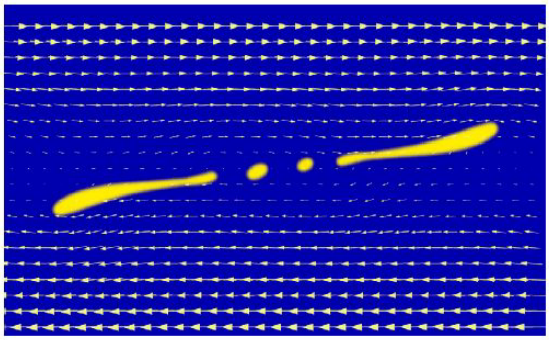
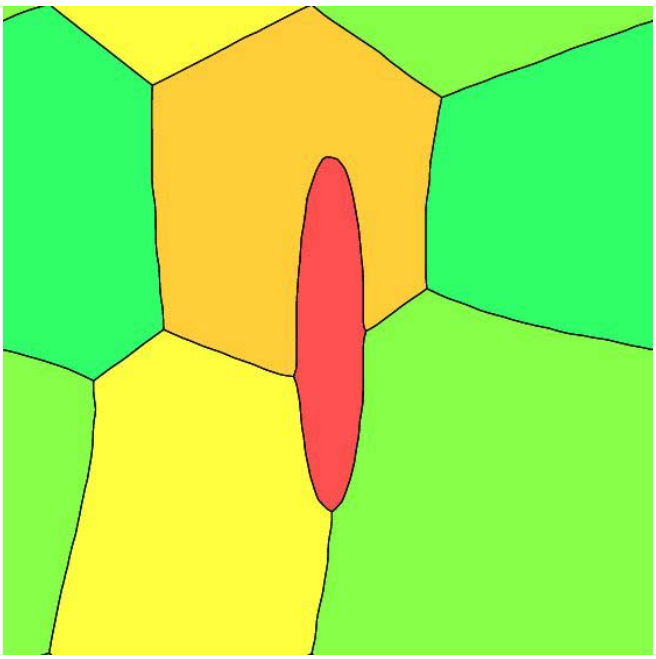


Figure 3. Break-up of a drop in a shear flow for a viscosity ratio $\eta_{\text{drop}}/\eta_{\text{solvent}} = 0.01$.

$$\rho(\theta)\left(\frac{\partial \mathbf{v}}{\partial t} + \mathbf{v}\cdot\nabla\mathbf{v}\right) = \nabla\cdot(2\eta(\theta)\mathbf{D}) - \nabla P + \mathbf{F}^{\text{int}}(\theta),$$

That is all: $\dfrac{\partial \theta}{\partial t} + \mathbf{v}\cdot\nabla\theta = -\Gamma\dfrac{\delta \mathrm{F}}{\delta\theta}.$



Thank you!

Enjoy the workshop