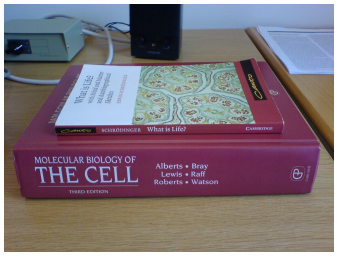
Numerical methods in practice
some examples

FYTN03, HT 2009

Henrik Jönsson
Computational Biology & Biological Physics,
Department of Theoretical Physics
Lund University, Lund, Sweden

What is life?



Morphogens, Turing

THE CHEMICAL BASIS OF MORPHOGENESIS

By A. M. TURING, F.R.S. *University of Manchester*

(*Received* 9 *November* 1951—*Revised* 15 *March* 1952)

It is suggested that a system of chemical substances, called morphogens, reacting together and diffusing through a tissue, is adequate to account for the main phenomena of morphogenesis. Such a system, although it may originally be quite homogeneous, may later develop a pattern or structure due to an instability of the homogeneous equilibrium, which is triggered off by random disturbances. Such reaction-diffusion systems are considered in some detail in the case of an isolated ring of cells, a mathematically convenient, though biologically unusual system.

Morphogens, Turing

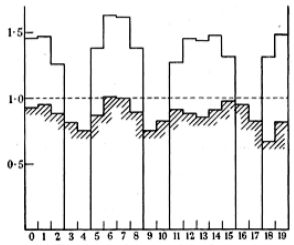
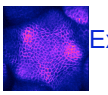


FIGURE 3. Concentrations of Y in the development of the first specimen (taken from table 1). ------- original homogeneous equilibrium; //// incipient pattern; ——— final equilibrium.

Early auxin transport model

…cal mod…tical

Methodology



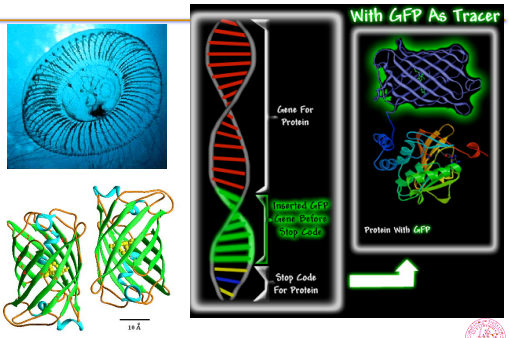
Experiments

Hypotheses

Mathematical models

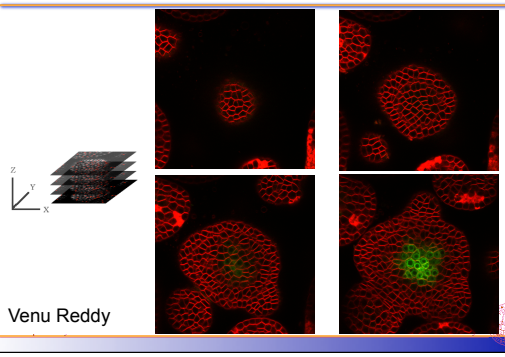
Computer simulation

Green Fluorescent Protein



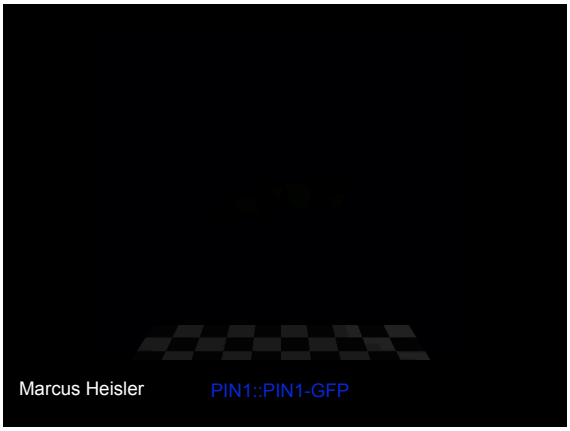
Lund University

WUS::GFP and red membrane stain



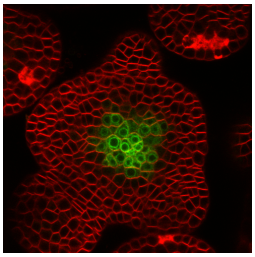
Venu Reddy

Lund University

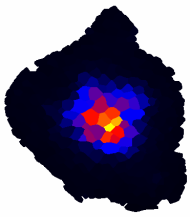


Marcus Heisler PIN1::PIN1-GFP

Quantitative measures from image



Green WUS::GFP
Red membrane stain

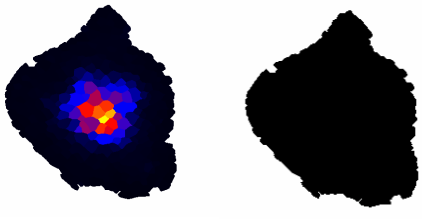


WUS "concentration"

Lund University

WUS network simulation (2D template)

Cell volumes, wall areas, and neighbors from template



Template **Simulation**

Lund University

Outline

- Protein folding, introducing all atoms and simulated tempering
- (Random walks for a DNA -> ask Tobias)
- TGFb-pathway, example of solving ODEs
- Finding parameters, χ^2 optimization
- Multicellular models with diffusion - example of solving PDEs

Lund University

HP model

Do you like G&Ns metropolis implementation?

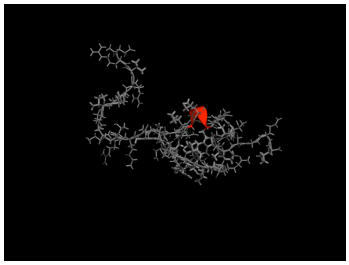
Why don't you like G&Ns metropolis implementation?

Why don't you like the local moves in G&Ns metropolis implementation?

Lund University

Peptide folding and aggregation

Anders Irbäck



All-atom Monte Carlo simulations, with sequence as the only input.

Lund University

Simulated tempering used

- Similar to simulated annealing, but system allowed to go up and down in temperature
- Temperature step is a metropolis step
- g_T temperature factor in exp(-E/T+g_T)
- g_T tuned in beginning of simulation to have the system spend equal times at different temperatures

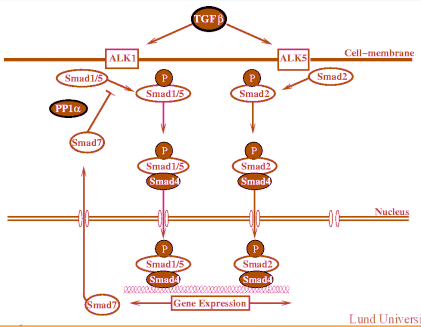
Lund University

Ordinary differential equations

- TGFb-pathway, example of solving ODEs
- State of art ODE solvers?
- Finding parameters, χ^2 optimization

Lund University

The TGFβ pathway

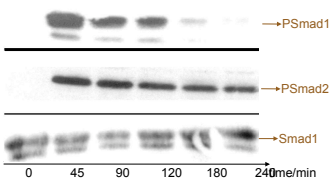


Lund University

Experimental Data

- Western blot analysis

Valdimarsdottir *et al.* 2004



- Data available for PSmad1 and Psmad2
- Different TGF-β dosages

Lund University

The Model

$$\emptyset \overset{p_0}{\underset{p_0 p_1}{\rightleftharpoons}} \text{ALK1} \quad (1)$$

$$\emptyset \overset{p_4}{\underset{p_4 p_5}{\rightleftharpoons}} \text{Smad4} \quad (2)$$

$$\emptyset \overset{p_8}{\underset{p_8 p_9}{\rightleftharpoons}} \text{ALK5} \quad (3)$$

$$\text{TGF}\beta + \text{ALK1} \overset{p_{13}}{\underset{p_{14}}{\rightleftharpoons}} \text{TA1} \quad (4)$$

$$\text{PSmad1} + \text{Smad4} \overset{p_{18}}{\underset{p_{19}}{\rightleftharpoons}} \text{PS14} \quad (5)$$

$$\text{TGF}\beta + \text{ALK5} \overset{p_{20}}{\underset{p_{21}}{\rightleftharpoons}} \text{TA5} \quad (6)$$

$$\text{P}_\text{A} + \text{TA1} \overset{\text{Smad7}}{\underset{p_{28}}{\overset{p_{27}}{\rightleftharpoons}}} \text{TA1P} \quad (7)$$

$$\text{P}_\text{B} + \text{TA5} \overset{\text{Smad7}}{\underset{p_{32}}{\overset{p_{31}}{\rightleftharpoons}}} \text{TA2P} \quad (8)$$

$$\emptyset \overset{p_2}{\underset{p_2 p_3}{\rightleftharpoons}} \text{Smad1} \quad (9)$$

$$\emptyset \overset{p_6}{\underset{p_6 p_7}{\rightleftharpoons}} \text{Smad2} \quad (10)$$

$$\emptyset \overset{\text{PS14N}}{\underset{p_{10}}{\overset{(p_{11}, p_{12})}{\rightleftharpoons}}} \text{Smad7} \quad (11)$$

$$\text{Smad1} \overset{\text{TA1}}{\underset{p_{17}}{\overset{(p_{15}, p_{16})}{\rightleftharpoons}}} \text{PSmad1} \quad (12)$$

$$\text{Smad2} \overset{\text{TA5}}{\underset{p_{24}}{\overset{(p_{22}, p_{23})}{\rightleftharpoons}}} \text{PSmad2} \quad (13)$$

$$\text{PSmad2} + \text{Smad4} \overset{p_{25}}{\underset{p_{26}}{\rightleftharpoons}} \text{PS24} \quad (14)$$

$$\text{PS14} \overset{p_{29}}{\underset{k_{30}}{\rightleftharpoons}} \text{PS14N} \quad (15)$$

Modeling of Biological Systems

- Deterministic model using Ordinary Differential Equations (ODE)

- Law of Mass Action (well known from undergraduate chemistry)

Modeling of Biological Systems

- Deterministic model using Ordinary Differential Equations (ODE)

- Law of Mass Action (well known from undergraduate chemistry)

$$\text{X} + \text{Y} \underset{k_2}{\overset{k_1}{\rightleftharpoons}} \text{Z}$$

Modeling of Biological Systems

- Deterministic model using Ordinary Differential Equations (ODE)

- Law of Mass Action (well known from undergraduate chemistry)

$$\text{X} + \text{Y} \underset{k_2}{\overset{k_1}{\rightleftharpoons}} \text{Z} \quad k_1[\text{X}][\text{Y}]$$

Modeling of Biological Systems

- Deterministic model using Ordinary Differential Equations (ODE)

- Law of Mass Action (well known from undergraduate chemistry)

$$\text{X} + \text{Y} \underset{k_2}{\overset{k_1}{\rightleftharpoons}} \text{Z} \quad k_1[\text{X}][\text{Y}] \longrightarrow \begin{aligned} \frac{d[\text{X}]}{dt} &= -k_1[\text{X}][\text{Y}] + k_2[\text{Z}] \\ \frac{d[\text{Y}]}{dt} &= -k_1[\text{X}][\text{Y}] + k_2[\text{Z}] \\ \frac{d[\text{Z}]}{dt} &= k_1[\text{X}][\text{Y}] - k_2[\text{Z}] \end{aligned}$$

Modeling Gene Expression

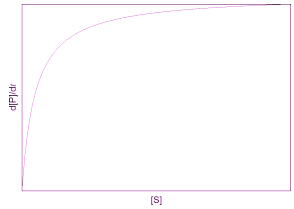
-The Michaelis-Menten Reaction

$$\text{S} + \text{E} \underset{k_b}{\overset{k_f}{\rightleftharpoons}} \text{ES} \overset{k_p}{\rightarrow} \text{P} + \text{E}$$

Modeling Gene Expression

-The Michaelis-Menten Reaction

$$\mathrm{S} + \mathrm{E} \underset{k_b}{\overset{k_f}{\rightleftharpoons}} \mathrm{ES} \overset{k_p}{\rightarrow} \mathrm{P} + \mathrm{E} \longrightarrow \frac{d[\mathrm{P}]}{dt} = \frac{v_{max}[\mathrm{S}]}{K_M + [\mathrm{S}]}$$

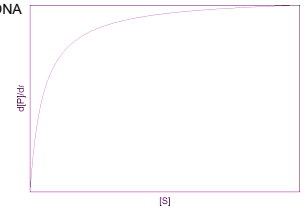


Modeling Gene Expression

-The Michaelis-Menten Reaction

$$\mathrm{S} + \mathrm{E} \underset{k_b}{\overset{k_f}{\rightleftharpoons}} \mathrm{ES} \overset{k_p}{\rightarrow} \mathrm{P} + \mathrm{E} \longrightarrow \frac{d[\mathrm{P}]}{dt} = \frac{v_{max}[\mathrm{S}]}{K_M + [\mathrm{S}]}$$

T DNA



The Model Equations

$$\frac{dA_1}{dt} = p_0(1 - p_1A_1) - p_{13}T_\beta A_1 + p_{14}T_1$$
$$\frac{dS_1}{dt} = p_2(1 - p_3S_1) - \frac{p_{15}T_1S_1}{p_{16} + S_1} + p_{17}P_1$$
$$\frac{dS_4}{dt} = p_4(1 - p_5S_4) - p_{18}P_1S_4 + p_{19}P_{14} - p_{25}P_2$$
$$\frac{dS_2}{dt} = p_6(1 - p_7S_2) - \frac{p_{22}T_1S_2}{p_{23} + S_2} + p_{24}P_2$$
$$\frac{dA_5}{dt} = p_8(1 - p_9A_5) - p_{20}T_\beta A_5 + p_{21}T_5$$
$$\frac{dS_7}{dt} = \frac{p_{11}P_{14}}{p_{12} + P_{14}} - p_{10}S_7$$
$$\frac{dP_1}{dt} = \frac{p_{15}T_1S_1}{p_{16} + S_1} - p_{17}P_1 - p_{18}P_1S_4 + p_{19}P_{14}$$
$$\frac{dP_{14}}{dt} = p_{18}P_1S_4 - p_{19}P_{14} - p_{29}P_{14} + p_{30}P_{14N}$$
$$\frac{dP_{14N}}{dt} = p_{29}P_{14} - p_{30}P_{14N}$$

$$\frac{dP_2}{dt} = \frac{p_{22}T_1S_2}{p_{23} + S_2} - p_{24}P_2 - p_{25}P_2S_4 + p_{26}P_{24}$$
$$\frac{dP_{24}}{dt} = p_{25}P_2S_4 - p_{26}P_{24}$$
$$\frac{dT_1}{dt} = p_{13}T_\beta A_1 - p_{14}T_1 - p_{27}S_7P_AT_1 + p_{28}T_{1P}$$
$$\frac{dT_5}{dt} = p_{20}T_\beta A_5 - p_{21}T_5 - p_{31}S_7P_BT_5 + p_{32}T_{5P}$$
$$\frac{dP_A}{dt} = -p_{27}S_7P_AT_1 + p_{28}T_{1P}$$
$$\frac{dP_B}{dt} = -p_{31}S_7P_BT_1 + p_{32}T_{1P}$$
$$\frac{dT_{1P}}{dt} = p_{27}S_7P_AT_1 - p_{28}T_{1P}$$
$$\frac{dT_{5P}}{dt} = p_{31}S_7P_BT_5 - p_{32}T_{5P}$$

How would you solve the equations?

Our experience

- We used 5th order Runge-Kutta with adaptive step size
- Since system stiff for some parameter values, we also implemented an implicit method (Rosenbrock)
- Rosenbrock much faster in the stiff cases, but since we scanned parameter space and most sets were not stiff, RK5 was at least as good!

State of art? (MATLAB)

Solver	Problem Type	Order of Accuracy	When to Use
ode45	Nonstiff	Medium	Most of the time. This should be the first solver you try.
ode23	Nonstiff	Low	For problems with crude error tolerances or for solving moderately stiff problems.
ode113	Nonstiff	Low to high	For problems with stringent error tolerances or for solving computationally intensive problems.
ode15s	Stiff	Low to medium	If ode45 is slow because the problem is stiff.
ode23s	Stiff	Low	If using crude error tolerances to solve stiff systems and the mass matrix is constant.
ode23t	Moderately Stiff	Low	For moderately stiff problems if you need a solution without numerical damping.
ode23tb	Stiff	Low	If using crude error tolerances to solve stiff systems.

Algorithms

ode45 is based on an explicit Runge–Kutta (4,5) formula, the Dormand–Prince pair. It is a *one-step* solver – in computing y(tn), it needs only the solution at the immediately preceding time point, y(tn-1). In general, ode45 is the best function to apply as a *first try* for most problems. [3]

ode23 is an implementation of an explicit Runge–Kutta (2,3) pair of Bogacki and Shampine. It may be more efficient than ode45 at crude tolerances and in the presence of moderate stiffness. Like ode45, ode23 is a one-step solver. [2]

ode113 is a variable order Adams–Bashforth–Moulton PECE solver. It may be more efficient than ode45 at stringent tolerances and when the ODE file function is particularly expensive to evaluate. ode113 is a *multistep* solver -- it normally needs the solutions at several preceding time points to compute the current solution. [7]

The above algorithms are intended to solve nonstiff systems. If they appear to be unduly slow, try using one of the stiff solvers below.

ode15s is a variable order solver based on the numerical differentiation formulas (NDFs). Optionally, it uses the backward differentiation formulas (BDFs, also known as Gear's method) that are usually less efficient. Like ode113, ode15s is a multistep solver. Try ode15s when ode45 fails, or is very inefficient, and you suspect that the problem is stiff, or when solving a differential-algebraic problem. [9], [10]

ode23s is based on a modified Rosenbrock formula of order 2. Because it is a one-step solver, it may be more efficient than ode15s at crude tolerances. It can solve some kinds of stiff problems for which ode15s is not effective. [9]

ode23t is an implementation of the trapezoidal rule using a "free" interpolant. Use this solver if the problem is only moderately stiff and you need a solution without numerical damping. ode23t can solve DAEs. [10]

ode23tb is an implementation of TR-BDF2, an implicit Runge–Kutta formula with a first stage that is a trapezoidal rule step and a second stage that is a backward differentiation formula of order two. By construction, the same iteration matrix is used in evaluating both stages. Like ode23s, this solver may be more efficient than ode15s at crude tolerances. [8], [1]

Lund University

State of art? (MATLAB)

References

[1] Bank, R. E., W. C. Coughran, Jr., W. Fichtner, E. Grosse, D. Rose, and R. Smith, "Transient Simulation of Silicon Devices and Circuits," *IEEE Trans. CAD*, 4 (1985), pp 436–451.

[2] Bogacki, P. and L. F. Shampine, "A 3(2) pair of Runge–Kutta formulas," *Appl. Math. Letters*, Vol. 2, 1989, pp 1–9.

[3] Dormand, J. R. and P. J. Prince, "A family of embedded Runge–Kutta formulae," *J. Comp. Appl. Math.*, Vol. 6, 1980, pp 19–26.

[4] Forsythe, G. , M. Malcolm, and C. Moler, *Computer Methods for Mathematical Computations*, Prentice–Hall, New Jersey, 1977.

[5] Kahaner, D. , C. Moler, and S. Nash, *Numerical Methods and Software*, Prentice–Hall, New Jersey, 1989.

[6] Shampine, L. F. , *Numerical Solution of Ordinary Differential Equations*, Chapman & Hall, New York, 1994.

[7] Shampine, L. F. and M. K. Gordon, *Computer Solution of Ordinary Differential Equations: the Initial Value Problem*, W. H. Freeman, San Francisco, 1975.

[8] Shampine, L. F. and M. E. Hosea, "Analysis and Implementation of TR-BDF2," *Applied Numerical Mathematics 20*, 1996.

[9] Shampine, L. F. and M. W. Reichelt, "The MATLAB ODE Suite," *SIAM Journal on Scientific Computing*, Vol. 18, 1997, pp 1–22.

[10] Shampine, L. F., M. W. Reichelt, and J.A. Kierzenka, "Solving Index-1 DAEs in MATLAB and Simulink," *SIAM Review*, Vol. 41, 1999, pp 538–552.

Lund University

State of art? (LSODA)

- In systems biology, LSODA is often quoted as the solver to use
- It combines two solvers Adams for non-stiff and BDF for stiff regions, and automatically switches in-between

See e.g. www.copasi.org

Lund University

Next problem, optimization

- Parameters in the TGFb-model unknown
- Adjust parameters to fit experimental data
- Minimize χ^2, the quadratic difference between model and data points

Lund University

How would you minimize χ^2 ?

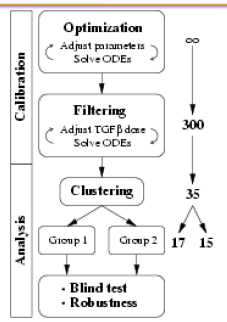
Lund University

Parameter optimization

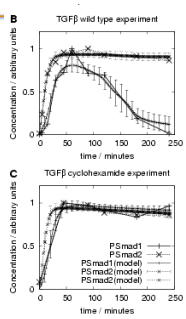
- Need a global optimization algorithm
- Adjust parameters to fit experimental data
- We used simulated annealing
- Minimize χ^2, the quadratic difference between model and data points
- Note: T is a parameter, E=χ^2 , one simulation to extract one E (i.e. takes time)

Lund University

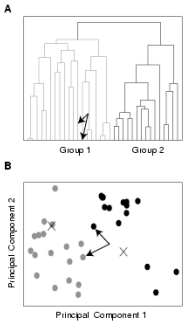
Procedure



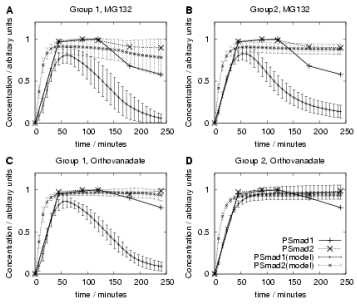
Reference experiments



Clustering of multiple solutions (more numerical methods)

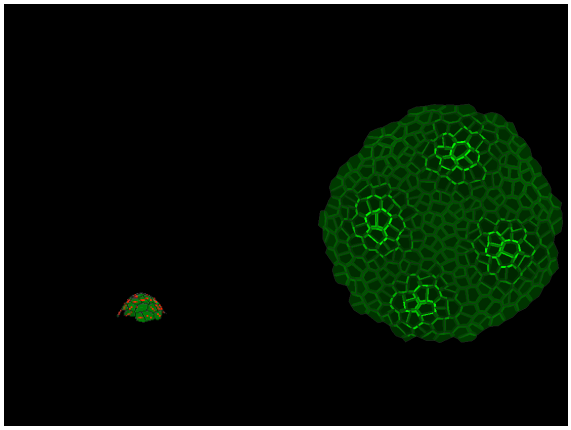


"Blind test" experiments



Multiple cells

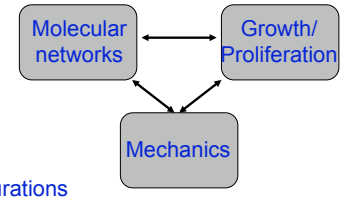
from ODEs to PDEs (and back)



Models need to take care of…

- Gene regulatory network
- Molecular reactions
- Molecular signalling
- Molecular transport
- Growth
- Cell proliferation
- Cell neighbourhood
- Mechanics

Simulator task



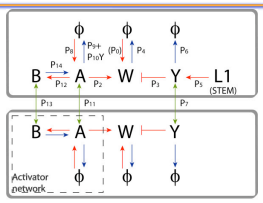
Initial configurations

Resolution

Solvers need to take care of…

- Different compartment types
- Model equations that change (dynamic neighborhood)
- Change in number of variables (cell divisions)

An example



$$\frac{dW}{dt} = \frac{1}{p_1} g(p_0 + p_2 A + p_3 Y) - p_4 W$$

$$\frac{dY}{dt} = p_5 L_1 - p_6 Y + p_7 \Delta_x Y$$

$$\frac{dA}{dt} = p_8 - p_9 A - p_{10} Y A + p_{11} \Delta_x A + p_{14} A^2 B$$

$$\frac{dB}{dt} = p_{12} A + p_{13} \Delta_x B - p_{14} A^2 B$$

How would you solve the equations?

PDE methods

Finite difference methods (space-time grid)

Finite element methods (common for mechanics fluid dyn)

An approach that is widely used, for example, in structural mechanics, is *finite-element methods*. Here, the desired function $u(\mathbf{x}, t)$ is expanded in some basis functions $\varphi_i(\mathbf{x})$,

$$u(\mathbf{x}, t) \approx \sum_{i=1}^{N} a_i(t) \varphi_i(\mathbf{x})$$

The task then is to determine the expansion coefficients $a_i(t)$, by some suitable criterion. Each basis function $\varphi_i(\mathbf{x})$ is nonzero only in a local neighborhood (an element).

Variational methods

There are also *variational methods*. Many physical PDE problems can be reformulated as variational problems; the desired function $u(\mathbf{x}, t)$ is an extremum of some integral I,

$$\delta I[u(\mathbf{x}, t)] = \delta \int d^D x F[u(\mathbf{x}, t)] = 0$$

Model with explicit carriers

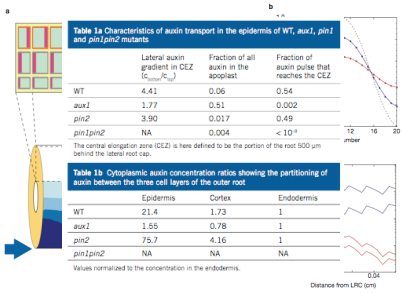


Table 1a Characteristics of auxin transport in the epidermis of WT, *aux1*, *pin1* and *pin1pin2* mutants

	Lateral auxin gradient in CEZ (k_{cross}/k_{up})	Fraction of all auxin in the apoplast	Fraction of auxin pulse that reaches the CEZ
WT	4.41	0.06	0.54
aux1	1.77	0.51	0.002
pin2	3.90	0.017	0.49
pin1pin2	NA	0.004	$< 10^{-6}$

The central elongation zone (CEZ) is here defined to be the portion of the root 500 µm behind the lateral root cap.

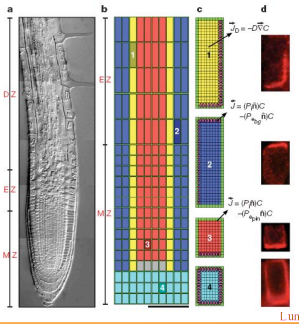
Table 1b Cytoplasmic auxin concentration ratios showing the partitioning of auxin between the three cell layers of the outer root

	Epidermis	Cortex	Endodermis
WT	21.4	1.73	1
aux1	1.55	0.78	1
pin2	75.7	4.16	1
pin1pin2	NA	NA	NA

Values normalized to the concentration in the endodermis.

(Swarup et al 2005) **Explicit Euler**

Root simulation



Root simulation

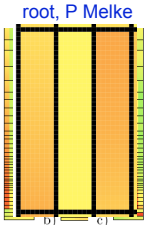
symbol	description	unit	value
Δt	time step	s (seconds)	0.1
Δx	space step	μm (microns)	2
D	auxin diffusion constant	$\mu m^2/s$	600
P_i	influx auxin permeability	μm/s	20
$P_{e_{bg}}$	background PIN efflux permeability	μm/s	1
$P_{e_{pin}}$	permeability due to basally expressed PINs, apically expressed PINs, PINs in root cap and lateral PINs of border cells.	μm/s	20

Explicit Euler and Alternating Direction Implicit (ADI) method

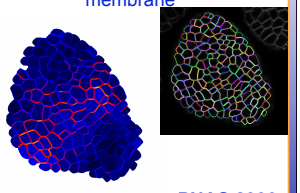
Nature 449, 1008-1013 (2007)

Plant subcompartments

Subcompartments root, P Melke



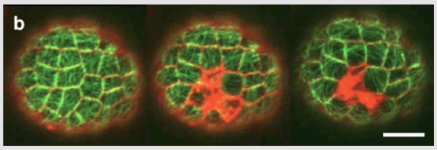
Subcompartments membrane



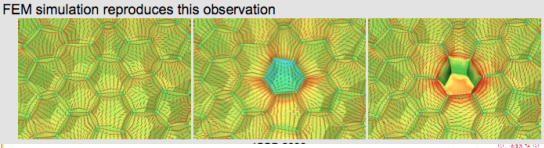
PNAS 2006

4-5 order RK w adaptive step size and Implicit Euler

FEM simulation of mechanics



FEM simulation reproduces this observation



MATLAB, the help is gone…

`sol = pdepe(m,pdefun,icfun,bcfun,xmesh,tspan)` solves initial-boundary value problems for systems of parabolic and elliptic PDEs in the one space variable x and time t. `pdefun`, `icfun`, and `bcfun` are function handles. See Function Handles in the MATLAB Programming documentation for more information. The ordinary differential equations (ODEs) resulting from discretization in space are integrated to obtain approximate solutions at times specified in `tspan`. The `pdepe` function returns values of the solution on a mesh provided in `xmesh`.

Parameterizing Functions Called by Function Functions, in the MATLAB mathematics documentation, explains how to provide additional parameters to the functions `pdefun`, `icfun`, or `bcfun`, if necessary.

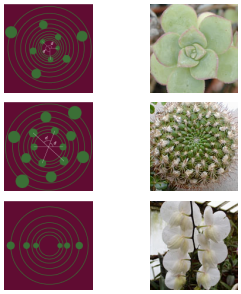
`pdepe` solves PDEs of the form:

$$c\left(x,t,u,\frac{\partial u}{\partial x}\right)\frac{\partial u}{\partial t} = x^{-m}\frac{\partial}{\partial x}\left(x^m f\left(x,t,u,\frac{\partial u}{\partial x}\right)\right) + s\left(x,t,u,\frac{\partial u}{\partial x}\right) \qquad (2\text{-}2)$$

The PDEs hold for $t_0 \le t \le t_f$ and $a \le x \le b$. The interval $[a,b]$ must be finite. m can be 0, 1, or 2, corresponding to slab, cylindrical, or spherical symmetry, respectively. If $m > 0$, then a must be $>= 0$.

Phyllotaxis, pattern formation in plants

- Spiral
- Multijugate (Decussate)
- Distichous



http://www.math.smith.edu/~phyllo/

Visible spirals, parastichies


8


13

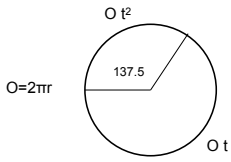

21

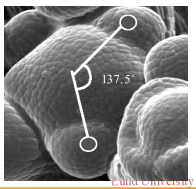

34

http://www.math.smith.edu/~phyllo

The Fibonacci sequence

- Fibonacci sequence, F_k: 1,1,2,3,5,8,13,21,34,…
- Golden mean $1/t = (\sqrt{5}+1)/2 \sim 1.6180\ldots$ $(F_{k+1}/F_{k-1}, k\to\infty)$
- Golden angle $\alpha \sim 137.5 \sim 360\ t^2$





Central zone+growth+spacing mechanism

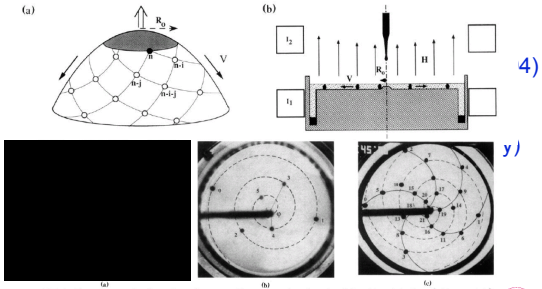


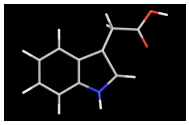
FIG. 2. Three photographs of experimental patterns. The numbers show the order of deposition of the drops (arbitrary origin). (a) $G = 1$, $\varphi = 180°$. (b) $G = 0.7$, $\varphi = 150°$ with $(i = 1, j = 2)$. (c) $G = 0.15$, $\varphi = 139°$ with $(i = 3, j = 5)$.

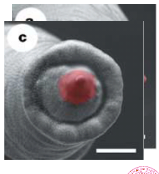
4)

y)

Auxin

- Indole-3-acetic acid (IAA)
- Plant hormone involved in several developmental activities (embryo development, tropism, primordia formation,…)
- Point addition at meristrem creates new primordia

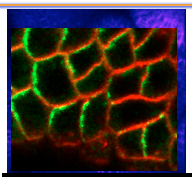




Reinhardt et al 2003

PIN-FORMED 1, PIN1 protein

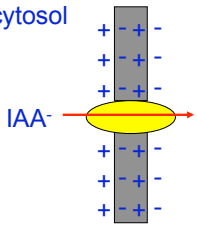
- Membrane protein
- Putative auxin efflux carrier
- Expressed in a phyllotactic pattern in epidermis, and in vascular tissue
- Polarized in cells
- Phyllotaxis fails in loss of function mutant



Ion transport across a membrane potenial

cytosol

wall



$$f(\Phi) = \frac{\Phi}{e^{\Phi} - 1}$$

$$\Phi = \frac{FVq}{RT}$$

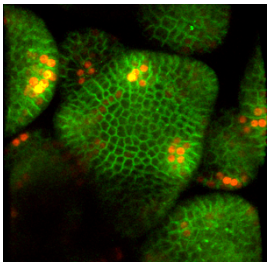
$$f_{out} \approx 4$$

$$f_{in} \approx 0.07$$

What about the sign for influx/efflux?

Lund University

Primordia is initiated by auxin peaks



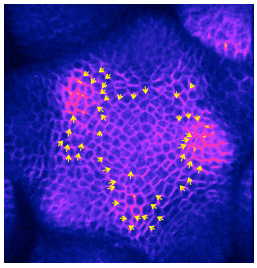
PIN1:GFP and DR5 expression

Lund University

What polarizes PIN1?

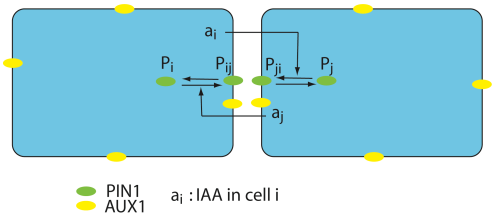
- PIN1 polarizes towards cells with high auxin content (hypothesis)

$$P_i \underset{}{\overset{f(a_j)}{\rightleftharpoons}} P_{ij}$$



Lund University

PIN1 cycling



PIN1
AUX1

a_i : IAA in cell i

Lund University

Auxin concentration model

degradation

production

active transport

passive transport

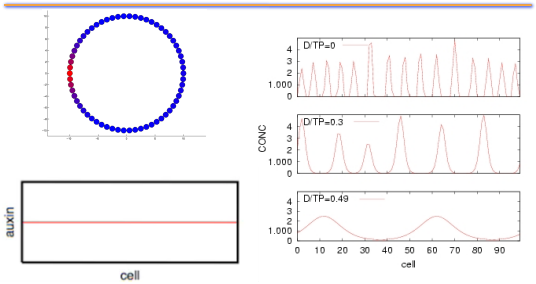
$$f_i = \frac{da_i}{dt} = K_p - K_d a_i + T\left(\sum_j^{N_i} a_j P_{ji} - a_i \sum_j^{N_i} P_{ij}\right) + D\left(\sum_j^{N_i} a_j - N_i a_i\right)$$

a_i – auxin concentration in cell i

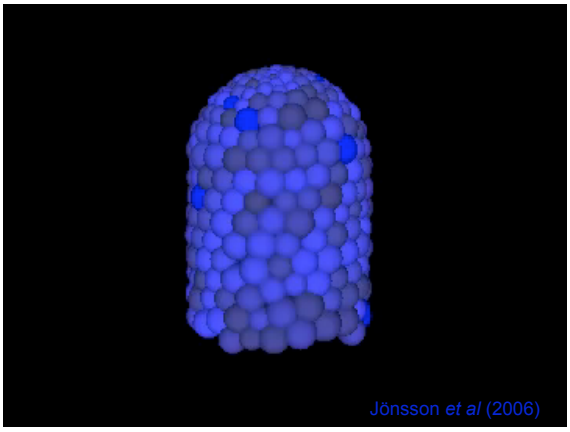
P_{ij} – polarized PIN1 ($P_{ij} \sim \frac{a_j P^{tot}}{\sum a_k}$)

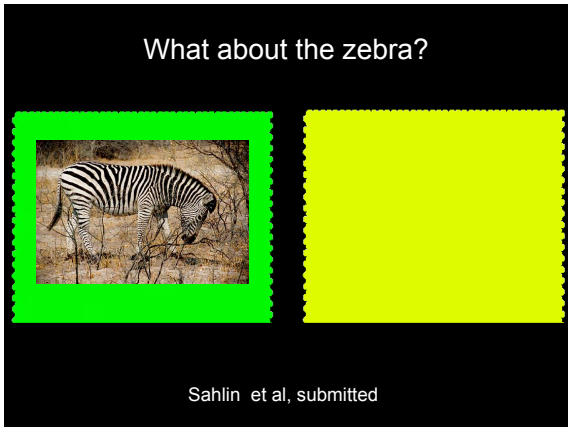
Lund University

Phyllotaxis model, Ring



Lund University


Jönsson *et al* (2006)


What about the zebra?
Sahlin et al, submitted