University of Glasgow

EARTH4072 – Igneous Geology

Introduction to Computational Geosciences

WKSHP 4 | Introduction to Comp Modelling II

Dr. Tobias Keller
Tobias.Keller@glasgow.ac.uk
@magmamatters

WORLD CHANGING GLASGOW

Intro Comp Geosci | Programme

Week	WKSHP I	WKSHP II	WKSHP III	WKSHP IV
20/10/2020	First Steps Coding	Comp Data Analysis	Comp Modelling I	Comp Modelling II

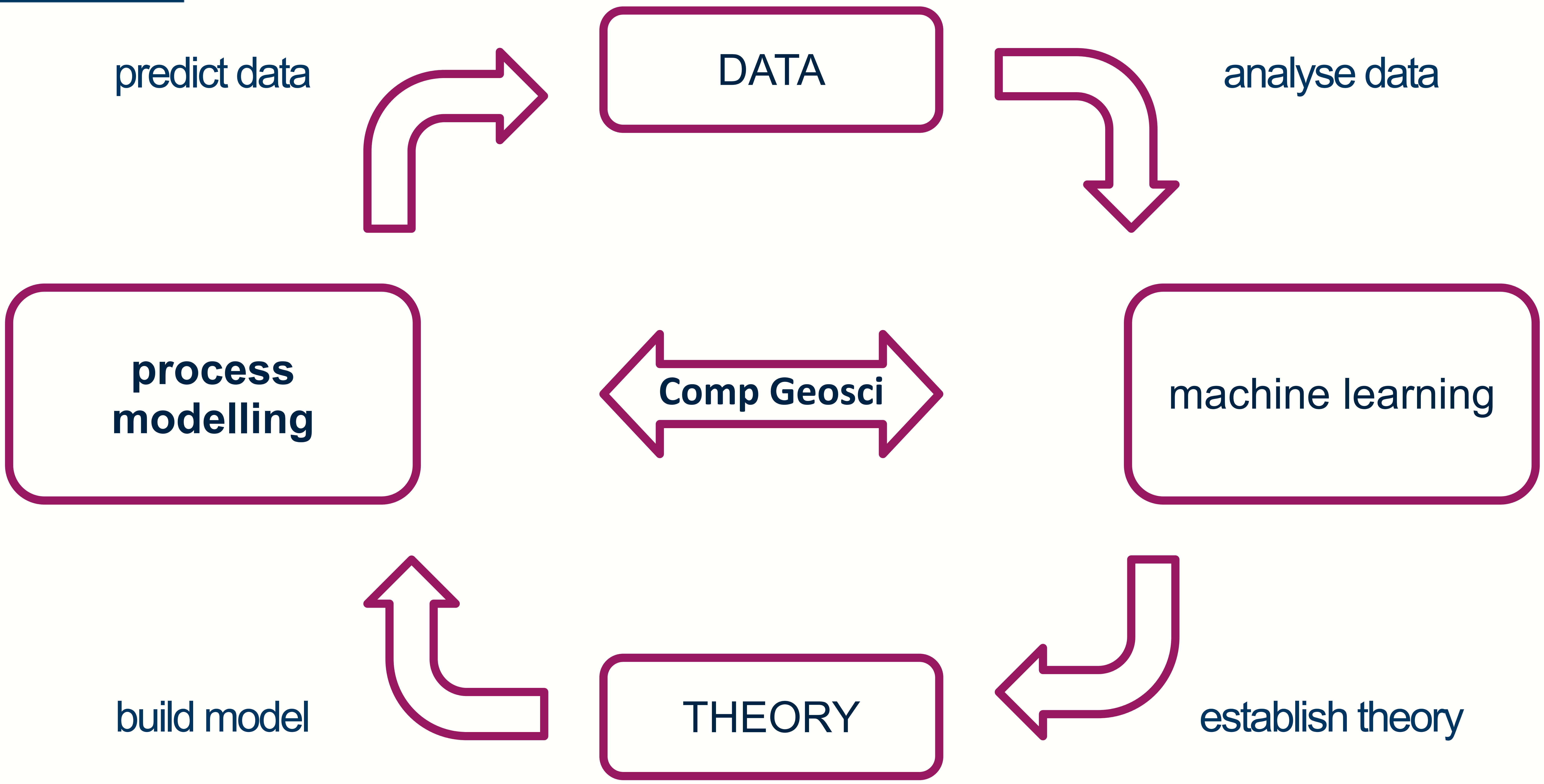
Comp GeoSci | Intended Learning Outcomes

Introduction to Computational Modelling II

- understand why and how scientist use modelling
- understand role of conceptual, analytical, numerical models
- understand model dimensionality (0-D, 1-D, 2-D, 3-D)
- understand the fundamentals of discretisation
- understand the fundamentals of numerical stability
- program and use a 1-D box model of crustal heat transport

```
% update constitutive relations
txx = eta .* exx + chi .* txxo;        % x-normal stress
tzz = eta .* ezz + chi .* tzzo;        % z-normal stress
txz = etac.* exz + chic.* txzo;        % xz-shear stress

p   = - zeta .* Div_V + xi .* po;      % compaction pressure
p([1 end],:) = p([end-1 2],:);         % periodic boundaries
p(:,[1 end]) = p(:,[end-1 2]);

w   = - (K(1:end-1,:).*K(2:end,:)).^0.5 .* (diff(P,1,1)./h + 1);
w(:,[1 end]) = w(:,[end-1 2]);

u   = - (K(:,1:end-1).*K(:,2:end)).^0.5 .* (diff(P,1,2)./h);
u([1 end],:) = u([end-1 2],:);

% update z-reference velocity
Div_tz = diff(tzz(:,2:end-1),1,1)./h + diff(txz,1,2)./h;

res_W(:,2:end-1) = - Div_tz + diff(P(:,2:end-1),1,1)./h + diff(p(:,

res_W([1 end],:) = [sum(res_W([1 end],:),1)./2;sum(res_W([1 end],:)
res_W(:,[1 end]) = res_W(:,[end-1 2]);

W = Wi - alpha.*res_W.*dtW + beta.*(Wi-Wii);

% update x-reference velocity
Div_tx  = diff(txx(2:end-1,:),1,2)./h + diff(txz,1,1)./h;

res_U(2:end-1,:) = - Div_tx + diff(P(2:end-1,:),1,2)./h + diff(p(2:

res_U([1 end],:) = res_U([end-1 2],:);
res_U(:,[1 end]) = [sum(res_U(:,[1 end]),2)./2,sum(res_U(:,[1 end])

U = Ui - alpha.*res_U.*dtU + beta.*(Ui-Uii);

% update reference pressure
Div_V(2:end-1,2:end-1) = diff(U(2:end-1,:),1,2)./h + diff(W(:,2:end
Div_v(2:end-1,2:end-1) = diff(u(2:end-1,:),1,2)./h + diff(w(:,2:end

res_P = Div_V + Div_v;

res_P([1,end],:) = res_P([end-1,2],:);
res_P(:,[1,end]) = res_P(:,[end-1,2]);

P = Pi - alpha.*res_P.*dtP + beta.*(Pi-Pii);

% update liquid evolution equation (enforce min/max limits on f)

flxdiv_fromm;  % upwind-biased advection/compaction term for liquid

res_f = (f-fo)./dt - (theta.*Div_fV + (1-theta).*Div_fVo);

res_f([1,end],:) = res_f([end-1,2],:);
res_f(:,[1,end]) = res_f(:,[end-1,2]);

if ~mod(step,nop); res_f = res_f - mean(res_f(:)); end

f = fi - alpha.*res_f.*dt/50;
f = max(0.001/f0,min(0.999/f0, f ));

% check and report convergence every nup iterations
if ~mod(it,nup); report; end
```

Comp Modelling | Basics



Comp Modelling | Basics

Definition

A simplified representation of a natural process or system aimed at interpreting, understanding, and predicting data.

“Universal Law” of Modelling

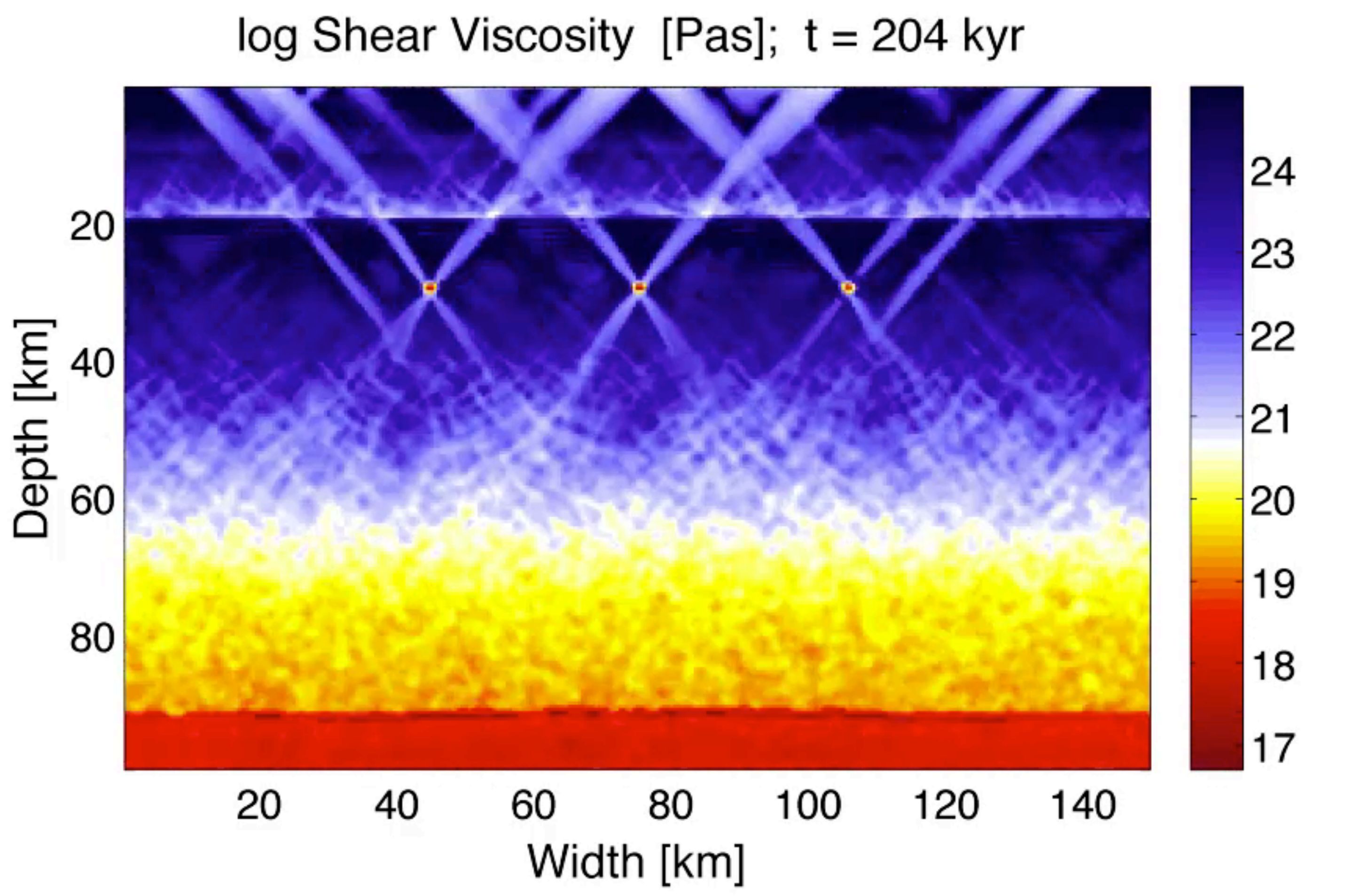
All models are wrong – some are useful.

Comp Modelling | Example Application

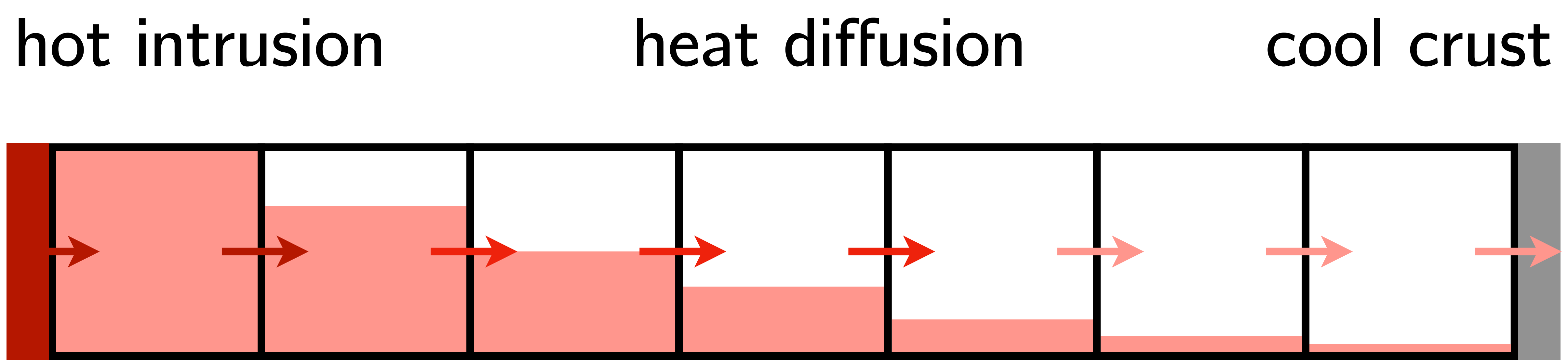
Magma Intrusion into Crust

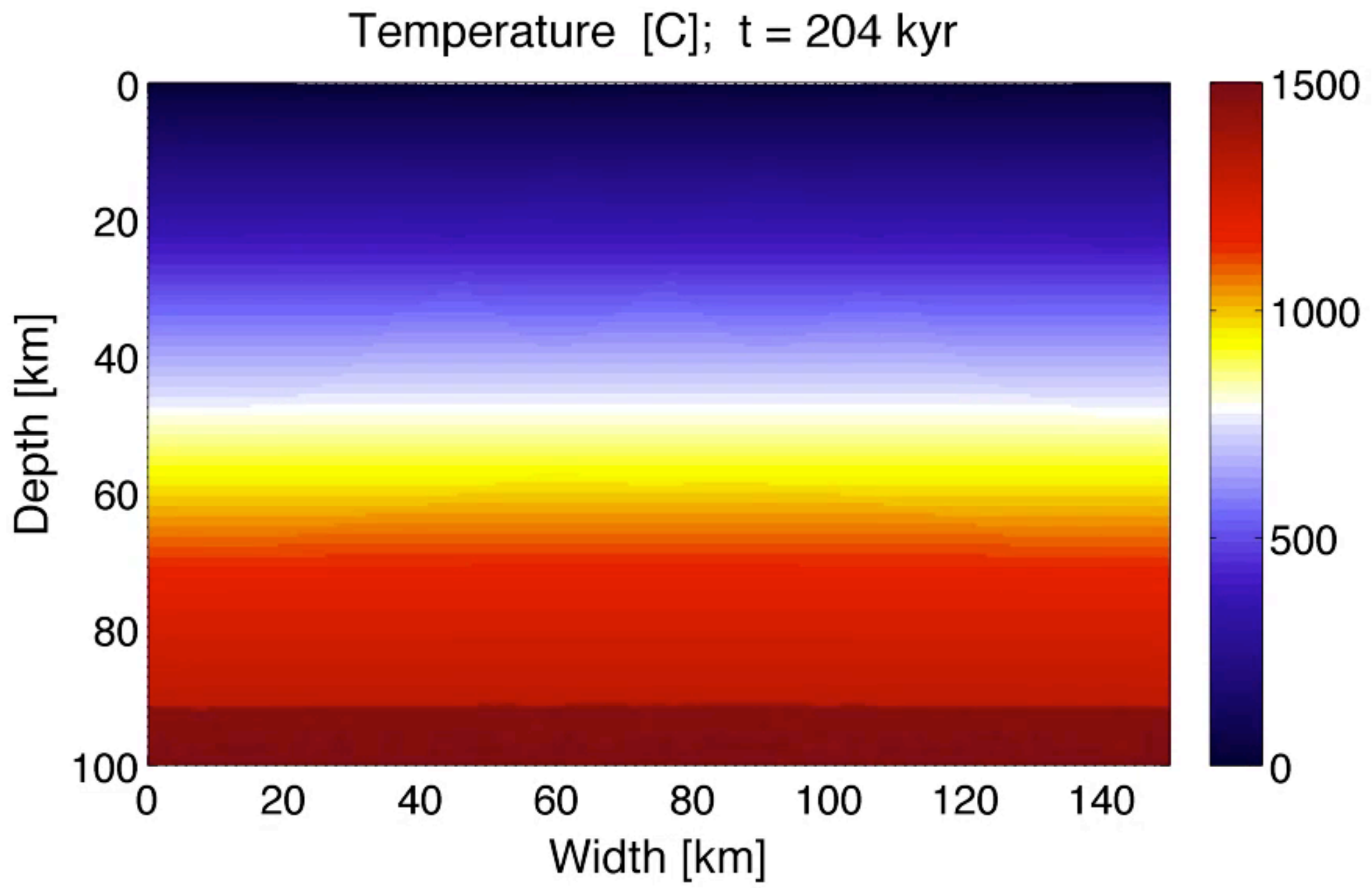
- hot magma rises into cool crust
- magma-filled tensile fractures: dyke
- heat transfer from magma to crust

en.wikipedia.org/wiki/Dike_(geology)



Conceptual Model





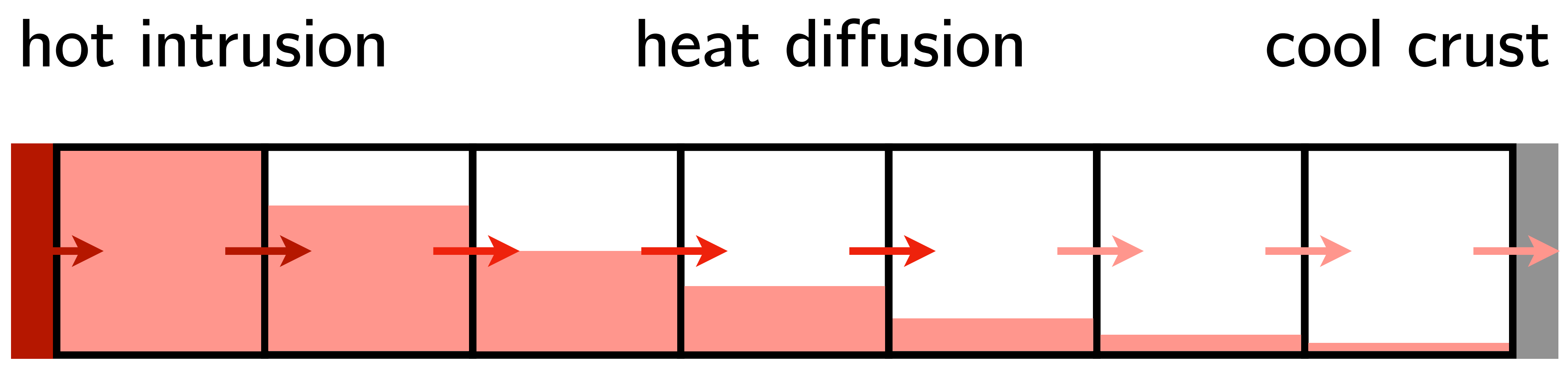
2-D model of magma ascent, heat transfer

Comp Modelling | Model to Solution

Dimensionality

- decide how many spatial dimensions and/or time to represent model in
- 1-dimensional model, time-dependence, one spatial dimension

Intrusive Contact Heating Model



crustal layer as row of ‘storage heaters’

change in temperature through time

$$\frac{\partial T(t,x)}{\partial t} = -\frac{\partial q(x)}{\partial x}$$

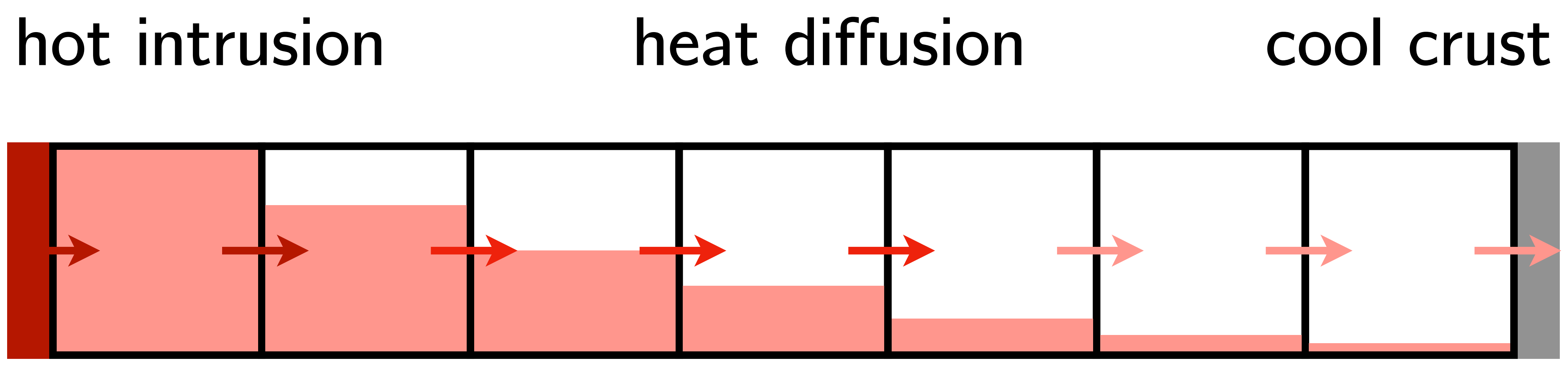
how to describe heat flux?

Comp Modelling | Model to Solution

Constitutive Relations

- relating fluxes to driving forces: use empirical law or derive from theory
- general constitutive relation: *flux* = *driving force* x *material resistance*
- Fourier's Law: heat flows from high to low T, resisted by thermal diffusivity, K

Intrusive Contact Heating Model



crustal layer as row of 'storage heaters'

Fourier's Law of heat transfer

$$q(x) = -K\frac{\partial T(x)}{\partial x}$$

flux

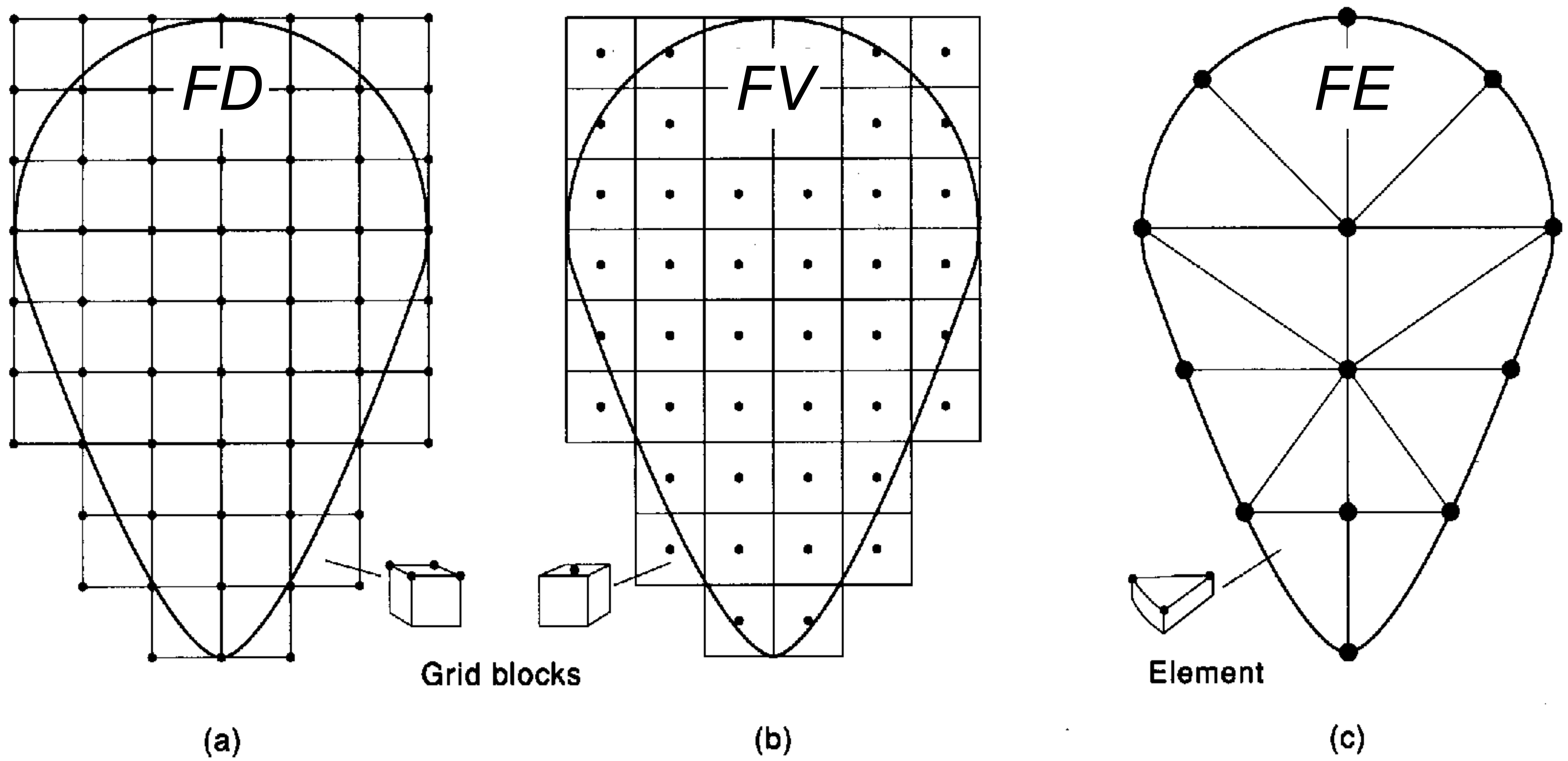
material resistance

driving force

Comp Modelling | Model to Solution

Spatial Discretisation

- different discretisation schemes, ways to divide space into cells, elements, etc.
- most common schemes: finite difference (FD), finite volume (FV), finite element (FE)



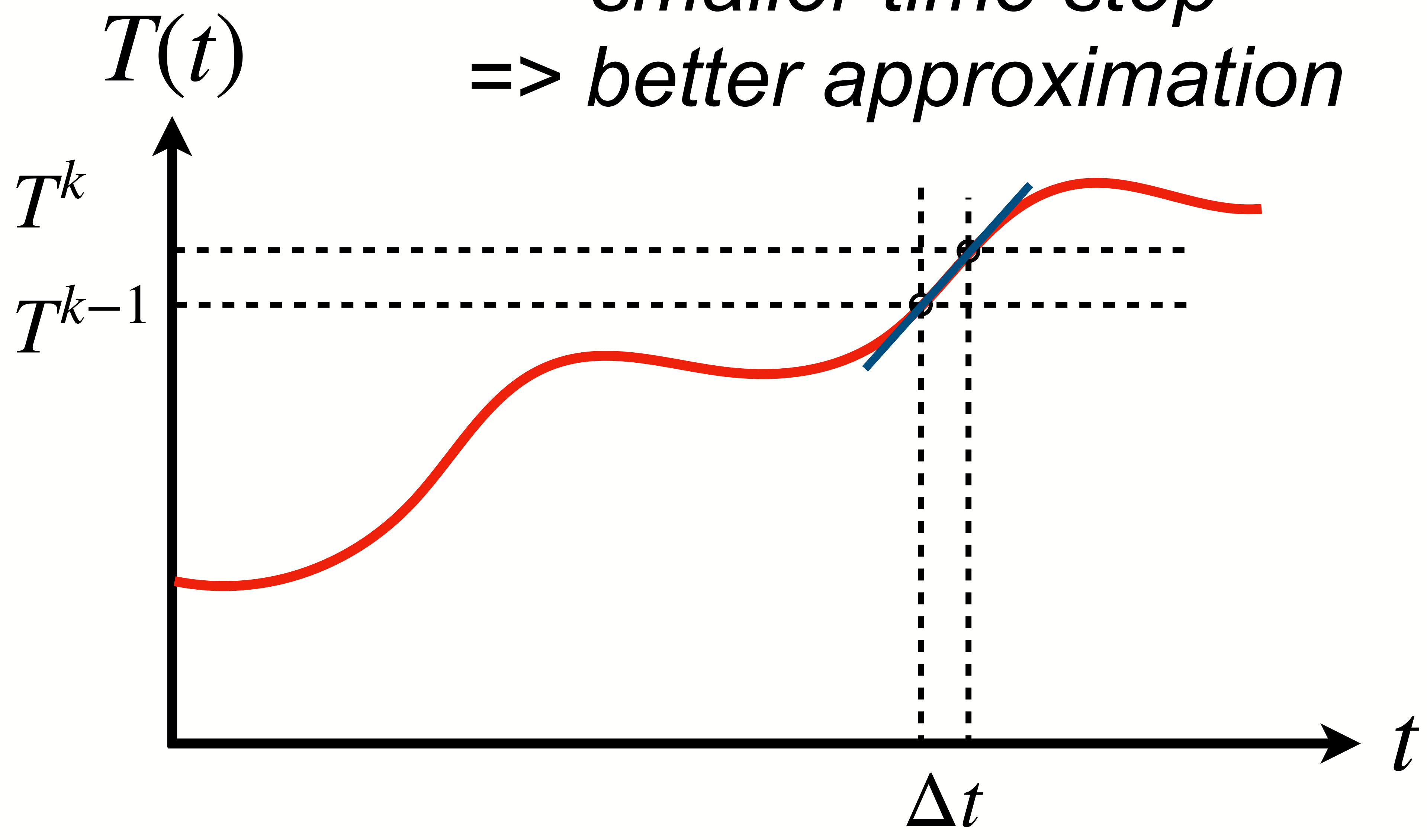
Temporal Discretisation (1-D layer model)

- approximate continuous rate of change as discrete steps of change
- approximate *infinitesimal derivative* by *finite difference*

finite difference approximation

$$\frac{T^k - T^{k-1}}{\Delta t} \approx \frac{\partial T}{\partial t} \text{ for } \Delta t \to 0$$

smaller time step => better approximation



Comp Modelling | Model to Solution

Discretisation: 1-D FV heat transfer model

- cell centre coordinates: x^i $(i = 1,\ldots,n)$; cell face coordinates: $x^{i\pm1/2} = (x^i + x^{i\pm1})/2$
- discrete temperatures: $T^{i,k} = T(x = x^i, t = t^k)$, located at cell centres
- discrete heat fluxes: $q^{i\pm1/2} = q(x = x^{i\pm1/2})$

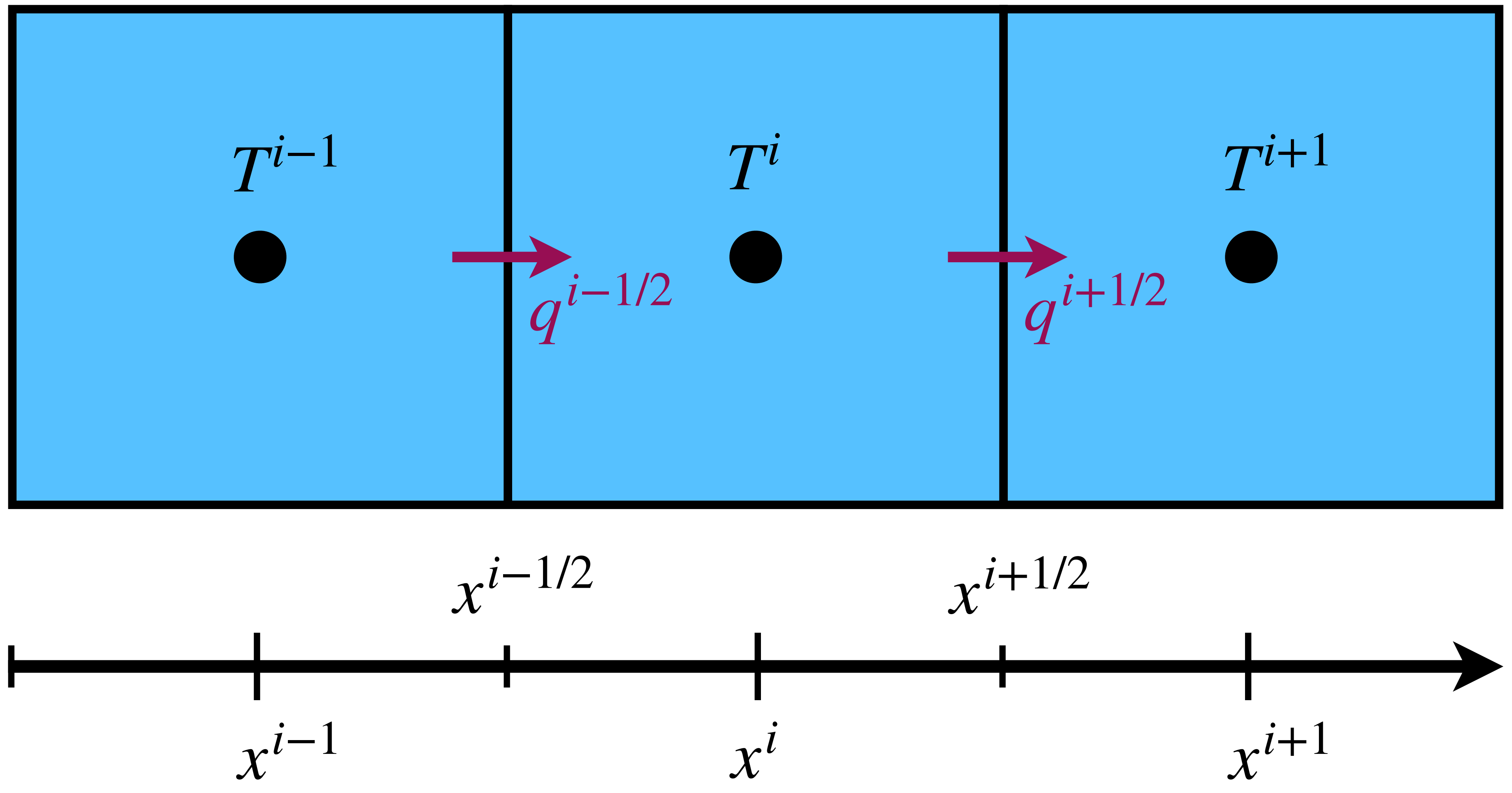
finite difference approximation

$$\frac{T^{i,k} - T^{i,k-1}}{\Delta t} = -\frac{q^{i+1/2} - q^{i-1/2}}{\Delta x}$$

$$q^{i+1/2} = -K\frac{T^{i+1,k-1} - T^{i,k-1}}{\Delta x}$$

$$q^{i-1/2} = -K\frac{T^{i,k-1} - T^{i-1,k-1}}{\Delta x}$$

FV stencil



Comp Modelling | Model to Solution

Discretisation: 1-D FV heat transfer model, variable parameter

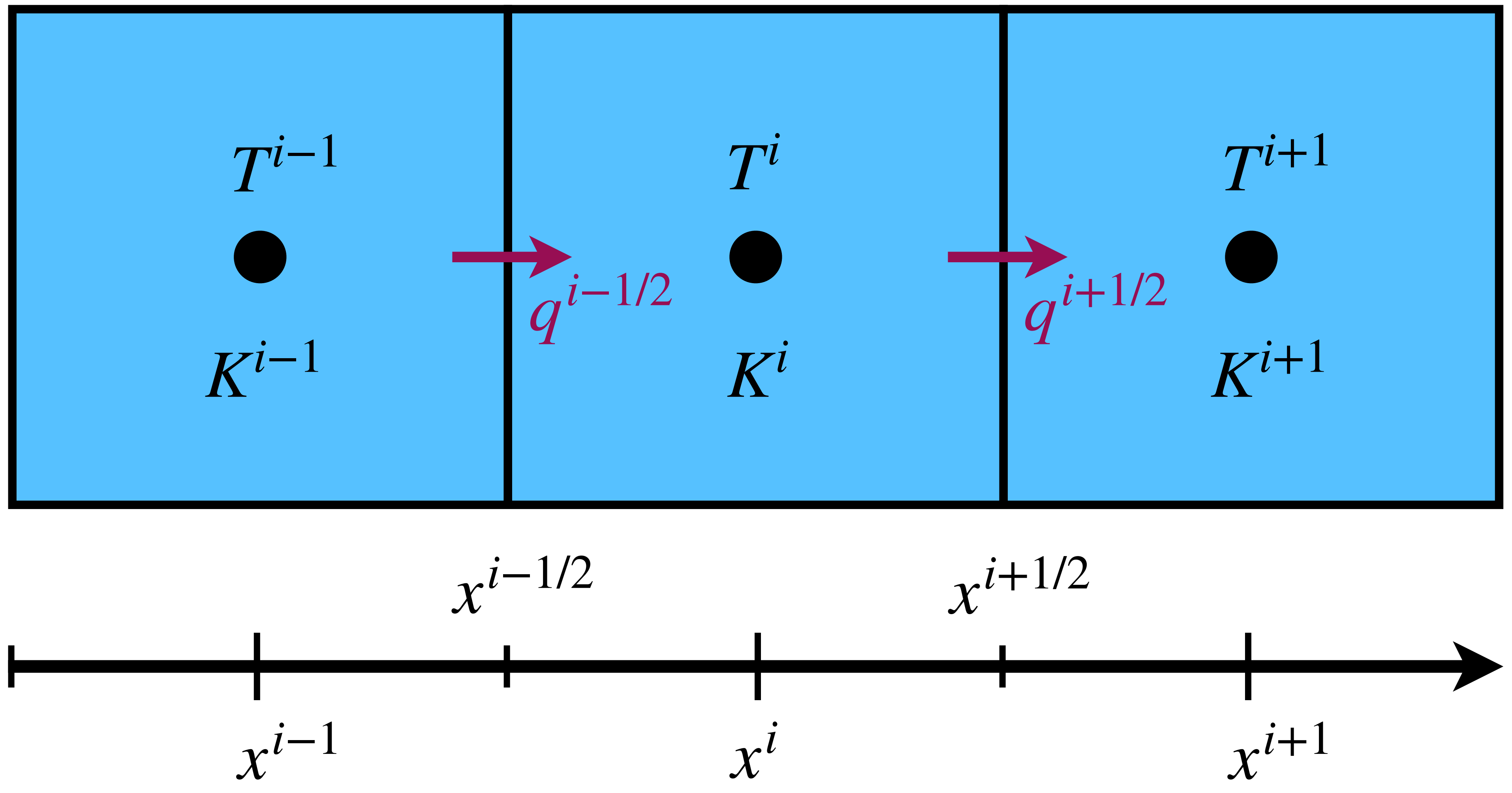
- discrete diffusivities: $K^i = K(x = i^k)$, located at cell centres
- discrete heat fluxes: $q^{i\pm 1/2} = q(x = x^{i\pm 1/2})$, located on cell faces

finite difference approximation

$$q^{i+1/2} = -\frac{(K^i + K^{i+1})}{2}\frac{T^{i+1,k-1} - T^{i,k-1}}{\Delta x}$$

$$q^{i-1/2} = -\frac{(K^i + K^{i-1})}{2}\frac{T^{i,k-1} - T^{i-1,k-1}}{\Delta x}$$

FV stencil



Comp Modelling | Model to Solution

Numerical Stability

- rate of change depends on flux, which depends on spatial gradient, which depends on rate of change: *feedback loop can run out of control*!
- stable time step: limit diffusive flux advance to half grid cell per time step

diffusivity	*limit distance*	*limit time step?*
$K \quad [\mathrm{m^2/s}]$	$\Delta x/2 \quad [\mathrm{m}]$	

Comp Modelling | Model to Solution

Numerical Stability

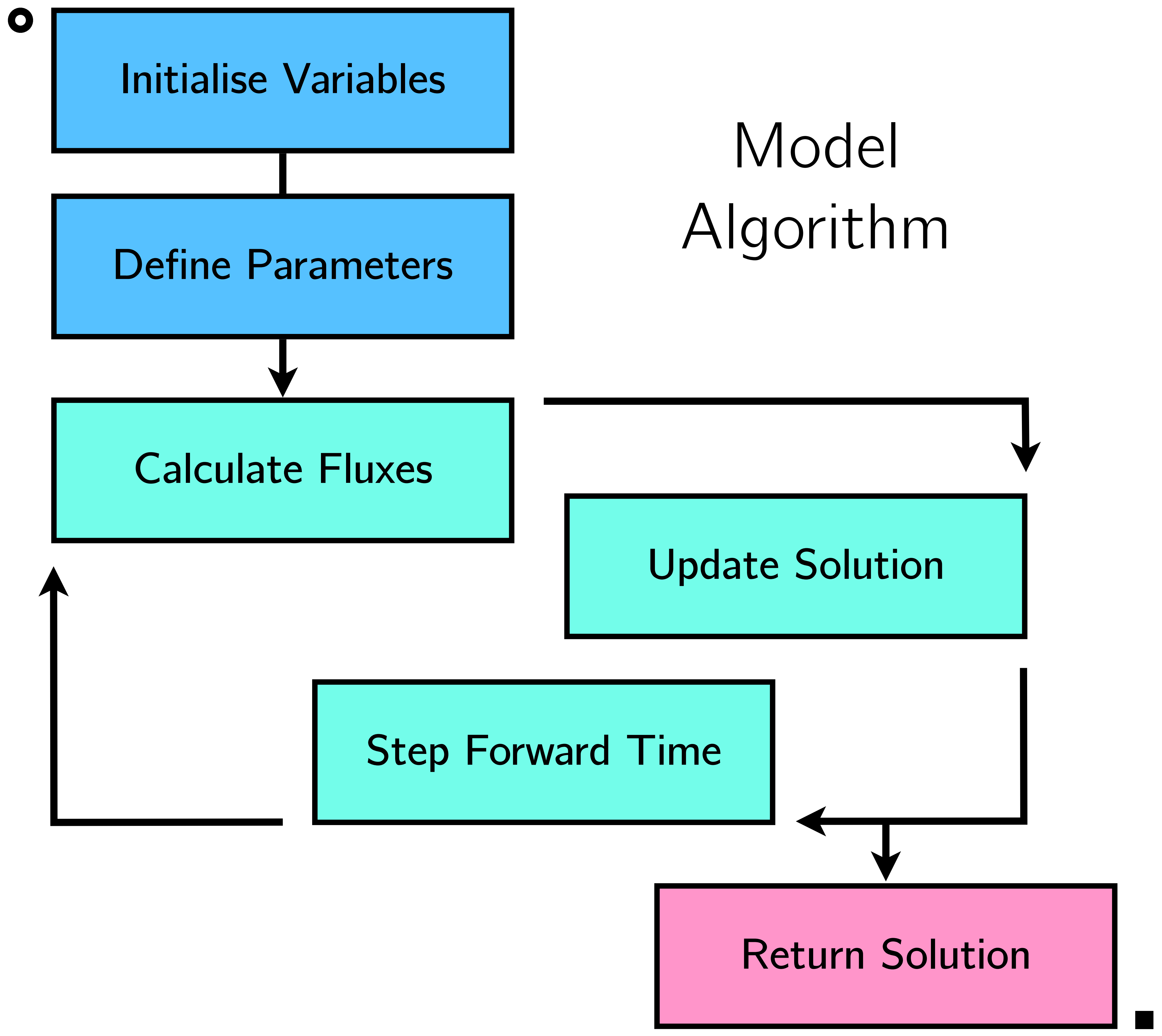
- rate of change depends on flux, which depends on spatial gradient, which depends on rate of change: *feedback loop can run out of control*!
- stable time step: limit diffusive flux advance to half grid cell per time step

diffusivity	*limit distance*	*limit time step*
$K \quad [\mathrm{m}^2/\mathrm{s}]$	$\Delta x/2 \quad [\mathrm{m}]$	$\Delta t \leq \dfrac{(\Delta x/2)^2}{K} \quad [s]$

Comp Modelling | Model to Solution

Numerical Implementation

- program algorithm to calculate solution step by step
- check if solution is robust, no numerical artefacts
- verify code to ensure accurate numerical solution



Comp Modelling | Model Benchmarking

Once programming is complete…

- test if the algorithm is programmed correctly
- most of model development spent debugging!
- frequent bugs: spelling or capitalisation, numbering or indexing, mistaken +/– signs, logic sequence

Convergence testing

- test if reduction of grid and time step size improves solution
- for simple test problems analytical (exact) solutions available
- residuals (diff. numerical to analytical solution) should reduce for $\Delta x \rightarrow 0; \; \Delta t \rightarrow 0.$
  => numerical solution *converges* to analytical solution.
- if no exact solution known (often the case!) test if solution converges towards the highest achievable numerical resolution

- ***Always benchmark a new code or new version of existing code!***

Comp Modelling | Model Documentation

A model is useless without thorough documentation

- document steps from conceptual to mathematical model
- document steps from continuous to discrete model representation
- document structure and usage of code (e.g., readme file)

Best practice for shareable, intelligible code

- use notebook format to integrate documentation with code!
- use commenting to describe every block, line of code!
- use descriptive naming convention for notebook or code files, functions, variables, parameters, etc.

- ***Highest standard is to write fully documented, open-source, reproducible code!***

Activity | Introduction to Numerical Modelling with Python

Build 1-D model of intrusive contact heating

- hot magma intrusion heats crustal wall rock
- simplified representation as row of storage heaters
- heat input vs. heat output rate

Consolidate basic numerical modelling

- complete prepared model code
- build in stable time step size
- test different initial conditions

Padlet task

- create new notebook, compose/document intrusive contact heating model from scratch, in your own code/words, post snapshot to Padlet