

PyBasin

Decompaction

Porosity is calculated assuming an exponential decay of porosity with depth [Athy [1930];Fowler2008]:

$$n = n_0 e^{-cz} \quad (1)$$

The thickness of the pore space b_w is related to porosity n by [Allen and Allen, 2005]:

$$b_w = \int_{z1}^{z2} n_0 e^{-cz} \quad (2)$$

which after integration gives:

$$b_w = \frac{n_0}{c} (e^{-cz_1} - e^{-cz_2}) \quad (3)$$

The total thickness is equal to the thickness of the pore space plus the thickness of the matrix: $b_t = b_w + b_m$

The initial, decompacted thickness can be calculated by setting $z_1 = 0$ and $z_2 = b_0$, where b_0 is the initial thickness:

$$b_0 = b_m + \frac{n_0}{c} (1 - e^{-cb_0}) \quad (4)$$

This equation cannot be solved analytically. We use iteration to find the value of b_0 :

$$b_0^{i+1} = b_m + \frac{n_0}{c} (1 - e^{-cb_0^i}) \quad (5)$$

where i denotes a step in the iteration. In general 10 to 20 iterations are required to find a value within 0.01 m of the ‘real’ decompacted thickness.

Decompaction with exhumation phases

Things get significantly more complicated if we included exhumation phases in the simulated burial history.

One solution is to first calculate the decompacted thickness assuming maximum present-day burial. Then go back to each of the exhumation phases and calculate the estimated burial depth of each unit. If the burial depth exceeds present-day burial depths then the thickness of the pore space b_w and the matrix thickness b_m need to be recalculated. To keep things simple we assume no thickness change during exhumation, only during burial.

In this case the pore thickness can be calculated as:

$$b_w = \frac{n_0}{c} (e^{-cz_{1,m}} - e^{-cz_{2,m}}) \quad (6)$$

where $z_{1,m}$ and $z_{2,m}$ are the top and the base of the unit at maximum burial depth. Because we assume no thickness changes during uplift, the thickness at maximum burial is the same as the present-day thickness (b_p):

$$z_{2,m} = z_{1,m} + b_p \quad (7)$$

However, we do not know in advance if the burial depth of a unit was deeper than the present-day burial or not.

The workflow is to calculate the initial and matrix thickness first on the basis of the present-day burial depth. Subsequently we go back to the exhumation phase and add the eroded section on top. Then we stack all the other units below the eroded unit and calculate their porosity and thickness. ...

For the first unit the thickness is simply the eroded thickness:

For the subsequent units, denoted by j :

$$b_{w,j} = \frac{n_0}{c} (e^{-cz_{b,j-1}} - e^{-cz_{b,j}}) \quad (8)$$

and the total thickness is again the sum of the matrix and the pore water thickness:

$$b_j = b_{w,j} + b_{m,j} \quad (9)$$

and the base depth of unit j is equal to the top plus the thickness:

$$z_j = z_{j-1} + b_j \quad (10)$$

again this leaves us with an equation that cannot be solved analytically:

$$b_j = b_m + \frac{n_0}{c} (e^{-cz_{b,j-1}} - e^{-c(z_{b,j-1} + b_j)}) \quad (11)$$

Heat flow

Heat flow is governed by the following equation

$$\rho c \frac{\partial T}{\partial t} = \nabla q + Q \quad (12)$$

where ρ is density (kg m^{-3}), c is heat capacity ($\text{J kg}^{-1} \text{K}^{-1}$), T is temperature (K), t is time (sec), q is heat flux (W m^{-2}) and Q is heat production (W m^{-3}).

We discretize the derivative of temperature in time as

$$\frac{\partial T}{\partial t} = \frac{T^{t+\Delta t} - T^t}{\Delta t} \quad (13)$$

heat flow is defined as

$$q = K \nabla T \quad (14)$$

where K is thermal conductivity ($\text{W m}^{-1} \text{K}^{-1}$).

For 1D domains, the heat flow equation can be discretized spatially as

$$\nabla q_i = \frac{q_{i+1/2} - q_{i-1/2}}{\Delta x_i} \quad (15)$$

$$q_{i+1/2} = K_{i+1/2} \frac{T_{i+1} - T_i}{\Delta x_{i+1/2}} \quad (16)$$

$$q_{i-1/2} = K_{i-1/2} \frac{T_i - T_{i-1}}{\Delta x_{i-1/2}} \quad (17)$$

And writing everything using x coordinates:

$$\Delta x_{i+1/2} = x_{i+1} - x_i \quad (18)$$

$$q_{i+1/2} = K_{i+1/2} \frac{T_{i+1} - T_i}{x_{i+1} - x_i} \quad (19)$$

$$q_{i-1/2} = K_{i-1/2} \frac{T_i - T_{i-1}}{x_i - x_{i-1}} \quad (20)$$

therefore

$$\nabla q_i = \frac{2}{x_{i+1} - x_{i-1}} \left(K_{i+1/2} \frac{T_{i+1} - T_i}{x_{i+1} - x_i} - K_{i-1/2} \frac{T_i - T_{i-1}}{x_i - x_{i-1}} \right) \quad (21)$$

Boundaries

For the left-hand boundary, with a fixed flux q_l from the left:

$$\nabla q_0 = \frac{q_{1/2} - q_l}{\Delta x_0} \quad (22)$$

$$\nabla q_0 = \frac{q_{1/2} - q_l}{\frac{x_1 - x_0}{2}} \quad (23)$$

$$\nabla q_0 = \frac{2}{x_1 - x_0} (q_{1/2} - q_l) \quad (24)$$

$$\nabla q_0 = \frac{2}{x_1 - x_0} \left(\left(K_{1/2} \frac{T_1 - T_0}{x_1 - x_0} \right) - q_l \right) \quad (25)$$

For the right-hand boundary at the last node n , with a fixed flux q_r from the right:

$$\nabla q_n = \frac{q_r - q_{n-1/2}}{\Delta x_0} \quad (26)$$

$$\nabla q_n = \frac{2}{x_n - x_{n-1}} (q_r - q_{n-1/2}) \quad (27)$$

$$q_{n-1/2} = K_{n-1/2} \frac{T_n - T_{n-1}}{x_n - x_{n-1}} \quad (28)$$

$$\nabla q_n = \frac{2}{x_n - x_{n-1}} \left(q_r - \left(K_{n-1/2} \frac{T_n - T_{n-1}}{x_n - x_{n-1}} \right) \right) \quad (29)$$

Constant K and Δx

To keep things simple we first try to solve this for a simple case where thermal conductivity (K) and grid size (Δx) are constant.

in the simple case with constant Δx and constant K we get:

$$\nabla q = \frac{2}{\Delta x} \left(K \frac{T_{i+1} - T_i}{\Delta x} - K \frac{T_i - T_{i-1}}{\Delta x} \right) \quad (30)$$

$$\nabla q = \frac{2K}{\Delta x^2} (T_{i+1} + T_{i-1} - 2T_i) \quad (31)$$

and for the full heat flow equation

$$\rho c \frac{T_i^{t+1} - T_i^t}{\Delta T} = \frac{2K}{\Delta x^2} (T_{i+1} + T_{i-1} - 2T_i) + Q \quad (32)$$

implicit FD:

$$\rho c \frac{T_i^{t+1} - T_i^t}{\Delta T} = \frac{2K}{\Delta x^2} (T_{i+1}^{t+1} + T_{i-1}^{t+1} - 2T_i^{t+1}) + Q \quad (33)$$

where T^t is known and T^{t+1} is unknown

reorganizing:

$$T_i^{t+1} - T_i^t = s (T_{i+1}^{t+1} + T_{i-1}^{t+1} - 2T_i^{t+1}) + \frac{Q\Delta t}{\rho c} \quad (34)$$

where

$$s = \frac{2K\Delta t}{\rho c \Delta x^2} \quad (35)$$

and putting all unknowns ($T_{...}^{t+1}$) on the left hand side

$$s (T_{i+1}^{t+1} + T_{i-1}^{t+1} - 2T_i^{t+1}) - T_i^{t+1} = -\frac{Q\Delta t}{\rho c} - T_i^t \quad (36)$$

$$-sT_{i-1}^{t+1} + (1 + 2s)T_i^{t+1} - sT_{i+1}^{t+1} = T_i^t + \frac{Q\Delta t}{\rho c} \quad (37)$$

Matrices

We now have discretized the heat flow equation for a uniform grid. We have one equation for each grid cell, and one unknown for each grid cell: the temperature at the next timestep T_i^{t+1} . Recall from your math class that if you have an equal number of unknowns and equations one can a set of equations to find the unknowns.

The most efficient way to solve a system of equations is to use matrices. First we convert equation 37 to matrix notation:

$$A\vec{x} = \vec{b} \quad (38)$$

where A is a matrix and \vec{x} and \vec{v} are vectors. We can fit equation 37 in this equation by moving all the coefficients to matrix A , the unknowns (T_0^{t+1} , T_1^{t+1} , T_2^{t+1} , etc..) to vector \vec{x} and the known variables ($T_i^t + \frac{Q\Delta t}{\rho c}$) to vector \vec{b} .

For an example grid containing 6 cells, the matrix A takes the shape of:

$$A = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ -s & (1+2s) & -s & 0 & 0 & 0 \\ 0 & -s & (1+2s) & -s & 0 & 0 \\ 0 & 0 & -s & (1+2s) & -s & 0 \\ 0 & 0 & 0 & -s & (1+2s) & -s \\ 0 & 0 & 0 & 0 & -s & (1+s) \end{bmatrix} \quad (39)$$

Note that the top and bottom line are different from what you would expect from eq. 37. The reason for this is that we have to account for boundary conditions, more on this in the next section.

The vector \vec{b} is:

$$\vec{b} = \begin{bmatrix} T_s \\ T_1^t + \frac{Q\Delta t}{\rho c} \\ T_2^t + \frac{Q\Delta t}{\rho c} \\ T_3^t + \frac{Q\Delta t}{\rho c} \\ T_4^t + \frac{Q\Delta t}{\rho c} \\ T_5^t + \frac{(Q+w/\Delta x)\Delta t}{\rho c} \end{bmatrix} \quad (40)$$

and the vector containing the unknowns is:

$$\vec{x} = \begin{bmatrix} T_0^{t+1} \\ T_1^{t+1} \\ T_2^{t+1} \\ T_3^{t+1} \\ T_4^{t+1} \\ T_5^{t+1} \end{bmatrix} \quad (41)$$

Solving system of equations

One way to solve a system of equations is to calculate the inverse matrix. If we multiply the left and right hand side of equation 38 with the inverse matrix we get:

$$A^{-1}Ax = A^{-1}b \quad (42)$$

multiplying a matrix with its inverse results in the identity matrix $A^{-1}A = I$:

$$I = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \quad (43)$$

and therefore we can simplify equation 42:

$$x = A^{-1}b \quad (44)$$

more on how to find the inverse matrix.... with python ...

Checking solutions with analytical equations

We check whether our numerical model is correct by comparing our solution to an analytical solution for the cooling of an intrusive in the subsurface. The solution for temperature change of an initially perturbed temperature field is [Carslaw and Jaeger, 1959]:

$$T(z, t) = T_b + \frac{T_i - T_b}{2} \left(\operatorname{erf} \left(\frac{L - z}{2\sqrt{\kappa t}} \right) + \operatorname{erf} \left(\frac{L + z}{2\sqrt{\kappa t}} \right) \right) \quad (45)$$

where T_b is the background temperature, T_i is the temperature of the intrusive, L is the length of the intrusive.

Following [Ehlers, 2005] we use this equation to simulate cooling of an intrusive in the subsurface. The numerical and analytical solutions for cooling are shown in Figure 1. The Intrusive body has an initial temperature of 700 degrees C, and stretches from 0 to 500 m distance. The background temperature is 50 degrees C. The solutions match to within ... degree C and show the gradual decrease in temperatures.

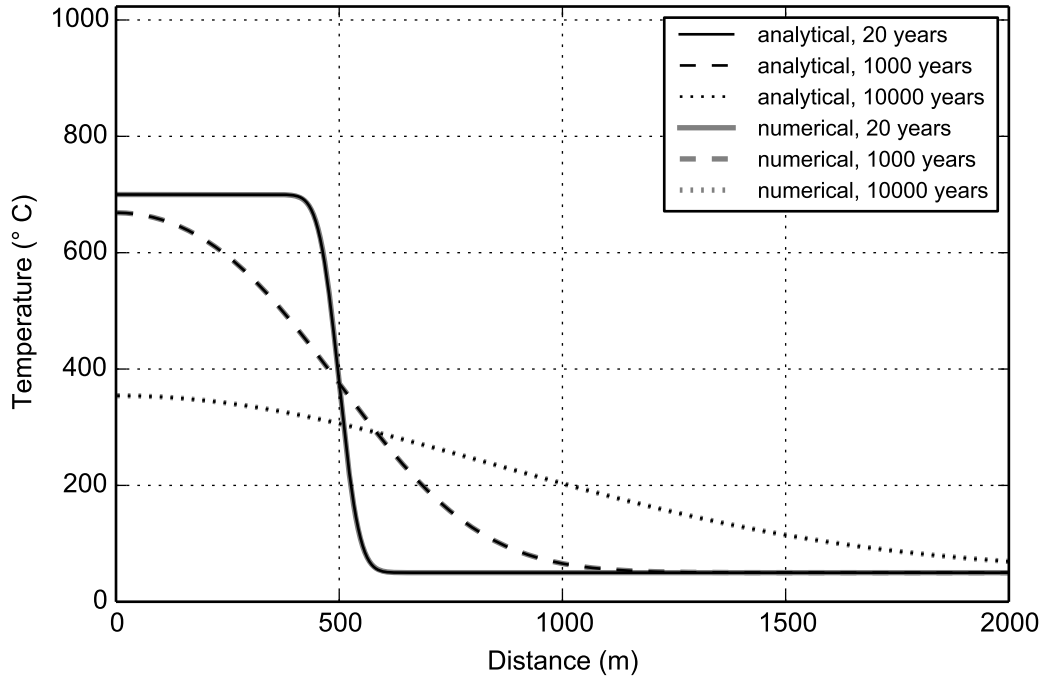


Figure 1: Comparison of numerical and analytical solution for the cooling of an intrusive in the subsurface. The analytical and numerical solution overlap, the maximum difference is 1 degree C. $\Delta x = 5$ m, $\Delta t = 10.0$ years, $\kappa = 1.01 \times 10^{-6} \text{ m}^2 \text{ s}^{-1}$ ($=32 \text{ km}^2 \text{ Ma}^{-1}$).

Variable K and Δx

Now we have successfully solved heat flow for a simple case with constant thermal conductivity, lets make things more complex and realistic by introducing non-constant grid spacing and thermal conductivity.

Recall that the discretized equation for ∇q is:

$$\nabla q = \frac{2}{x_{i+1} - x_{i-1}} \left(K_{i+1/2} \frac{T_{i+1} - T_i}{x_{i+1} - x_i} - K_{i-1/2} \frac{T_i - T_{i-1}}{x_i - x_{i-1}} \right) \quad (46)$$

if we define

$$s = \frac{2}{x_{i+1} - x_{i-1}} \quad (47)$$

$$t = \frac{K_{i+1/2}}{x_{i+1} - x_i} \quad (48)$$

$$u = \frac{K_{i-1/2}}{x_i - x_{i-1}} \quad (49)$$

we end up with a substituted equation:

$$\nabla q = suT_{i-1}^{t+1} - (st + su)T_i^{t+1} + stT_{i+1}^{t+1} \quad (50)$$

adding this to the full heat flow equation yields:

$$\rho_i c_i \frac{T_i^{t+1} - T_i^t}{\Delta t} = suT_{i-1}^{t+1} - (st + su)T_i^{t+1} + stT_{i+1}^{t+1} + Q \quad (51)$$

if we define v as:

$$v = \frac{\Delta t}{\rho_i c_i} \quad (52)$$

we can rewrite this as

$$T_i^{t+1} - T_i^t = suvT_{i-1}^{t+1} - (stv + suv)T_i^{t+1} + stvT_{i+1}^{t+1} + Qv \quad (53)$$

which after rearranging gives:

$$T_i^t + Qv = -suvT_{i-1}^{t+1} + (1 + (stv + suv))T_i^{t+1} - stvT_{i+1}^{t+1} \quad (54)$$

Lower fixed flux boundary:

For the lower node n with a specified flux q_r the flux is given by:

$$\nabla q_n = \frac{2}{x_n - x_{n-1}} \left(q_r - \left(K_{n-1/2} \frac{T_n - T_{n-1}}{x_n - x_{n-1}} \right) \right) \quad (55)$$

and therefore the full equation for the last node is:

$$\rho_n c_n \frac{T_n^{t+1} - T_n^t}{\Delta t} = \frac{2}{x_n - x_{n-1}} \left(q_r - \left(K_{n-1/2} \frac{T_n - T_{n-1}}{x_n - x_{n-1}} \right) \right) + Q \quad (56)$$

if we define:

$$s_n = \frac{2}{x_n - x_{n-1}} \quad (57)$$

$$u_n = \frac{K_{n-1/2}}{x_n - x_{n-1}} \quad (58)$$

$$T_n^{t+1} - T_n^t = s_n v (q_r - (u_n (T_n - T_{n-1}))) + Q v \quad (59)$$

$$T_n^{t+1} - T_n^t = s_n v q_r - (s_n v u_n (T_n - T_{n-1})) + Q v \quad (60)$$

$$T_n^{t+1} - T_n^t = s_n v q_r - s_n v u_n T_n^{t+1} + s_n v u_n T_{n-1}^{t+1} + Q v \quad (61)$$

$$-T_n^t - Q v - s_n v q_r = -(1 + s_n v u_n T_n^{t+1}) + s_n v u_n T_{n-1}^{t+1} \quad (62)$$

$$T_n^t + Q v + s_n v q_r = -s_n v u_n T_{n-1}^{t+1} + (1 + s_n v u_n T_n^{t+1}) \quad (63)$$

and therefore the last term in the matrix is

Final matrix

casting this equation into matrix form we end up with a slightly more complex matrix A :

$$A = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ -suv & (1 + stv + suv) & -stv & 0 & 0 & 0 \\ 0 & -suv & (1 + stv + suv) & -stv & 0 & 0 \\ 0 & 0 & -suv & (1 + stv + suv) & -stv & 0 \\ 0 & 0 & 0 & -suv & (1 + stv + suv) & -stv \\ 0 & 0 & 0 & 0 & -suv & (1 + suv) \end{bmatrix} \quad (64)$$

The vector \vec{b} is:

$$\vec{b} = \begin{bmatrix} T_s \\ T_1^t + Qv \\ T_2^t + Qv \\ T_3^t + Qv \\ T_4^t + Qv \\ \dots \\ T_n^t + Qv + svq_r \end{bmatrix} \quad (65)$$

References

- Allen, P. A., and J. R. Allen (2005), *Basin analysis: principles and applications*, Blackwell publishing, Oxford.
- Athy, L. F. (1930), Density, porosity, and compaction of sedimentary rocks, *AAPG Bulletin*, 14(1), 1–24.
- Carslaw, H. S., and J. C. Jaeger (1959), *Conduction of heat in solids*, Oxford University Press, New York.
- Ehlers, T. A. (2005), Computational Tools for Low-Temperature Thermochronometer Interpretation, *Reviews in Mineralogy and Geochemistry*, 58(1), 589–622, doi:[10.2138/rmg.2005.58.22](https://doi.org/10.2138/rmg.2005.58.22).