

Investigation 2: 1D Diffusion

Christopher Pattison

Non-Linear Diffusion

$$\nabla \cdot k \nabla T = q$$

The heat equation is solved where k is a function of T . The equation is discretized with the Finite Volume Method with k varying in space

$$\int_V \frac{dkdT}{dx^2} + q = 0$$

$$k \left. \frac{dT}{dx} \right|_{x_w}^{x_e} + q(x_e - x_w)$$

$$\Delta x_e = x_E - x_P; \quad \Delta x_w = x_P - x_W$$

$$\frac{k_e(T_E - T_P)}{\Delta x_e} - \frac{k_w(T_P - T_W)}{\Delta x_w} + q\Delta x$$

$$\frac{k_e}{\Delta x_e} T_E - \left(\frac{k_e}{\Delta x_e} + \frac{k_w}{\Delta x_w} \right) T_P + \frac{k_w}{\Delta x_w} T_W + q(x_e - x_w)$$

When some $k(T)$ is plugged in, a non-linear system is obtained that can be solved by a non-linear solver.

$$k(T) = k_0 + k_1 e^{-\frac{T}{T_0}}$$

$$k_e = \frac{2}{k(T_E)^{-1} + k(T_P)^{-1}}; \quad k_w = \frac{2}{k(T_P)^{-1} + k(T_W)^{-1}}$$

Non-Linear Solvers

Picard Iteration

Picard Iteration involves selecting an initial guess for T then iteratively reforming A and solving for T . Because it fails to acknowledge much of a relationship (other than its existence) between T and its coefficients, it performs extremely slow for strongly non-linear systems.

$$A_n x_{n+1} = b$$

If a non-linear model is to be used for b , it is also just a matter of updating b every iteration.

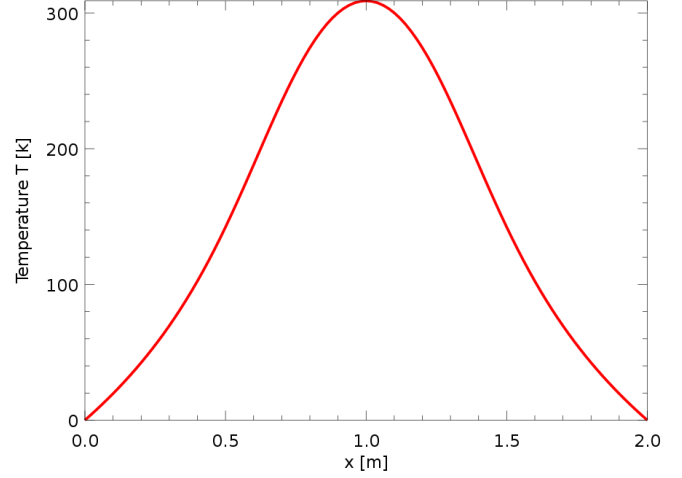


Figure 1: Heat Equation solution with $k(T)$

Newton's Method

Newton's Method solves for the zeros of a function. We can define a vector valued residual function

$$R = Ax - b$$

Since $R = \mathbf{0}$ when a valid solution is reached, finding the zeros would be the same as solving for x . Newton's Method for a scalar function is

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

This can be generalized for a vector valued function as

$$x_{n+1} = x_n - J^{-1} f(x_n)$$

Where J is the Jacobian of $f(x)$ and $f(x)$ is the previously defined residual function R . Since finding the inverse directly is expensive, we can instead put it into the form

$$J \Delta x = -f(x_n)$$

A linear system solver can then be used and x updated

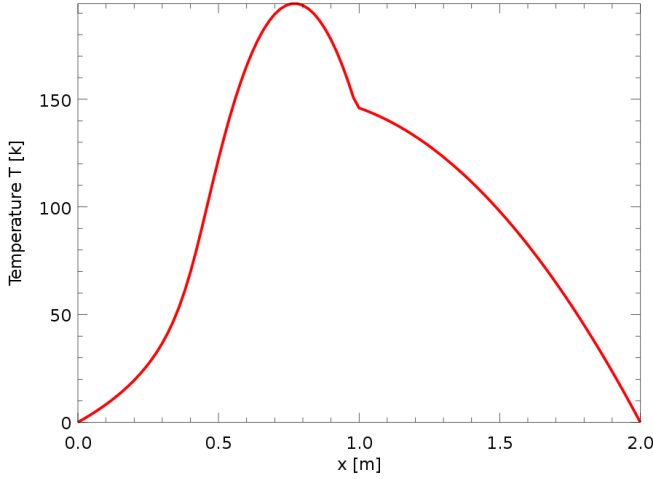


Figure 2: Solution with discontinuous k

Implementation

Boundary conditions were moved to a separate module in order to easily update all boundary conditions at once. In addition, the evaluation of k was also moved to this module; allowing for conductivity models to be implemented easily. Picard Iteration proved adequate, converging in 14 iterations for the test case.

Discontinuous Conductivity

The modular structure of the code easily lends itself to arbitrary specification of k

$$k = \begin{cases} k_0 + k_1 e^{\frac{T}{T_0}} & x \leq 1 \\ k_3 & x > 1 \end{cases}$$

Notably, the required number of iterations over the continuous model increased depending on $\frac{k_1}{k_0}$. A model with discontinuous k could model fuel rods in an enclosure or a heating element where the thermal conductivity is a strong function of temperature.

Performance Improvement

A better performing non-linear solver could be achieved by using a higher order method such as Hal-

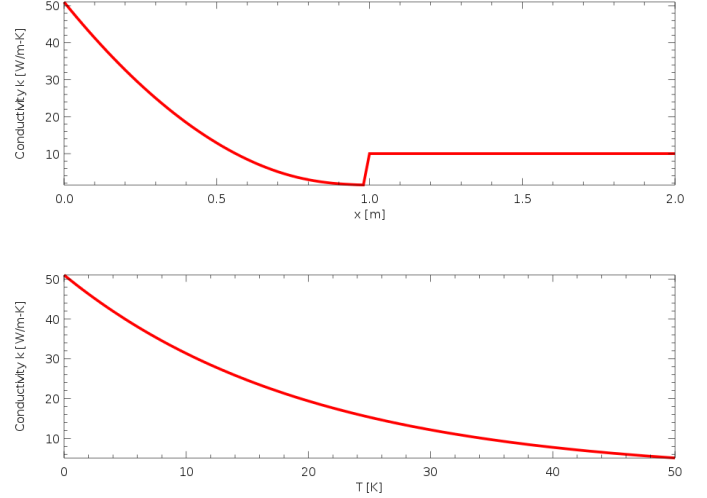


Figure 3: Discontinuous $k(x, T)$ conductivity $\frac{k_1}{k_0} = 20$; $k_3 = 10$

$\frac{k_1}{k_0}$	Discontinuous	Continuous
10	16	15
25	24	17
50	27	14
75	19	9
100	13	8
150	9	6
200	7	6

Figure 4: Conductivity Model Performance

ley's Method.

$$x_{n+1} = x_n - \frac{2f(x_n)f'(x_n)}{2f'(x_n)^2 - f(x_n)f''(x_n)}$$

$$(2J^2 - J'f(x_n))\Delta x = -2Jf(x_n)$$

Using numerical derivation, the Jacobian and derivative of the Jacobian could be constructed with ease. The convergence would be greater than Newton's Method, but the extra overhead would only be worthwhile on very non-linear problems.