Calculation of Heat Conduction Utilizing Neural Networks

Aleksandr Bogdanov, Milan Holec, Pavel Váchal

(and whoever will work on this)

Notes compiled by PV (pavel.vachal@fjfi.cvut.cz)

April 28, 2023

Contents

1 Governing Equations

We want to solve

$$\frac{d\,\varepsilon(x)}{dt} = -\vec{\nabla}\cdot\vec{q}(x), \tag{1}$$

where

$$\vec{q}(x) = -\alpha(x)\kappa(x)T(x)^{\beta(x)}\vec{\nabla}T(x), \tag{2}$$

and

$$\varepsilon(x) = C_V(x)\,T(x) = \frac{3}{2}n(x)\,k_B\,T(x). \tag{3}$$

The Gauss units are as follwoing $\varepsilon\left[\frac{\mathrm{erg}}{\mathrm{cm}^3}\right], T[\mathrm{eV}], q[\frac{\mathrm{erg}}{\mathrm{cm}^2}]$ with $k_B = 1.380649\times10^{-16}\frac{\mathrm{erg}}{\mathrm{K}} = 1.602178\times10^{-12}\frac{\mathrm{erg}}{\mathrm{eV}}$ and α,β are unitless. The conductivity κ in our model is defined in Gauss units as

$$\kappa(x) = \frac{Z(x)+0.24}{Z(x)+4.2}\;\frac{1.31\times10^{10}}{Z(x)\lambda_{ei}(x)}\;\tau^{\beta(x)-\frac{5}{2}}, \tag{4}$$

where Coulomb logarithm $\lambda_{ei}(x) = 23 - \ln\left(\frac{\sqrt{n(x)}Z(x)}{T(x)^{3/2}}\right)$ (see Plasma Formularly) and $\tau = \frac{1}{T_{\text{preheat}}}$. **We use $T_{\text{preheat}} = 1000$ eV to safely include the preheat region** of the hohlraum wall simulation. Note that $1\,\text{erg} = 1\,\frac{\text{g}\cdot\text{cm}^2}{\text{s}^2} = 10^{-7}$ J, which might be used to convert heat flux for benchmarking.

In order to solve (1), (2), and (3), we define

$$F := C_V \frac{dT}{dt} - \frac{d}{dx}\underbrace{\left(\alpha\kappa T^\beta \frac{d}{dx}T\right)}_{q} \tag{5}$$

and our task will be to find T that solves

$$F(T) = 0. \tag{6}$$

2 Discretization

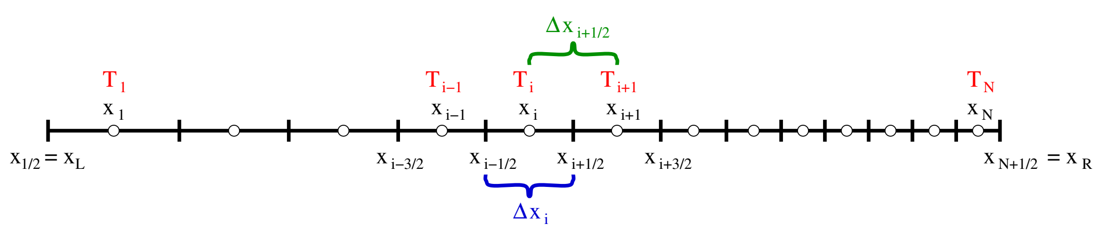

Figure 1: *Discretization of the 1D problem*

We discretize the problem on a general 1D mesh with cell-related variables indexed by integers and node-related ones by half-integers, as shown in Fig. 1.

Thus the cell volume is

$$\Delta x_i = x_{i+\frac{1}{2}} - x_{i-\frac{1}{2}}, \tag{7}$$

while the volume of the node-assigned dual cell (distance between neighboring cell centers) is

$$\Delta x_{i+\frac{1}{2}} = x_{i+1} - x_i = \frac{\Delta x_i + \Delta x_{i+1}}{2}. \tag{8}$$

The divergence of q in (1) is discretized on the primary cell by the finite difference

$$\vec{\nabla}\cdot\vec{q}\Big|_i \overset{1D}{=} \frac{dq}{dx}\Big|_i \approx \frac{q_{i+\frac{1}{2}} - q_{i-\frac{1}{2}}}{\Delta x_i}, \tag{9}$$

with the nodal value of the flux (2) being approximated as

$$q_{i+\frac{1}{2}} = \overline{(\alpha\kappa T^\beta)}_{i+\frac{1}{2}} \frac{T_{i+1} - T_i}{\Delta x_{i+\frac{1}{2}}}, \tag{10}$$

where $\overline{(\alpha\kappa T^\beta)}_{i+\frac{1}{2}}$ is obtained by some kind of averaging from the two connected cells, for example

$$\overline{(\alpha\kappa T^\beta)}_{i+\frac{1}{2}} = \frac{\left(\alpha\kappa T^\beta\right)_i + \left(\alpha\kappa T^\beta\right)_{i+1}}{2}, \tag{11a}$$

$$\overline{(\alpha\kappa T^\beta)}_{i+\frac{1}{2}} = \frac{\Delta x_i \left(\alpha\kappa T^\beta\right)_i + \Delta x_{i+1} \left(\alpha\kappa T^\beta\right)_{i+1}}{\Delta x_i + \Delta x_{i+1}}, \quad \text{or} \tag{11b}$$

$$\overline{(\alpha\kappa T^\beta)}_{i+\frac{1}{2}} = \frac{\frac{1}{\Delta x_i} \left(\alpha\kappa T^\beta\right)_i + \frac{1}{\Delta x_{i+1}} \left(\alpha\kappa T^\beta\right)_{i+1}}{\frac{1}{\Delta x_i} + \frac{1}{\Delta x_{i+1}}}, \tag{11c}$$

where we denoted

$$\left(\alpha\kappa T^\beta\right)_j = \alpha_j \kappa_j T_j^{\beta_j}. \tag{12}$$

Discretizing (6), resp. (1), (2), and (3) over the i-th cell, we have

$$F_i := C_{Vi} \frac{dT_i}{dt} - \frac{\overline{(\alpha\kappa T^\beta)}_{i+\frac{1}{2}} \frac{T_{i+1}-T_i}{\Delta x_{i+\frac{1}{2}}} - \overline{(\alpha\kappa T^\beta)}_{i-\frac{1}{2}} \frac{T_i - T_{i-1}}{\Delta x_{i-\frac{1}{2}}}}{\Delta x_i} \tag{13}$$

with space-dependent α and β being provided by the neural network and k, C_V being also functions of x:

$$\alpha = \alpha(\mathrm{NN}(x)), \qquad \beta = \beta(\mathrm{NN}(x)), \qquad k = k(Z(x)), \qquad C_V = C_V(n(x)). \tag{14}$$

At this point let us remark, that classical heat conductivity in plasma uses constant $\beta = 5/2$, which further simplifies the equations. This is the case for example in [Silar, vyzkumak, 2009]. However, there the problem is transformed using $\theta = T^{7/2}$ and solved by a mimetic scheme, whereas here we are going to proceed by Newton's iterative method.

For a regular mesh (i.e., with equidistant nodes), we have

$$\Delta x = \Delta x_i = \Delta x_{i+\frac{1}{2}}, \ \forall i, \tag{15}$$

and thus (13) simplifies to

$$F_i = C_{Vi} \frac{dT_i}{dt} - \frac{1}{\Delta x^2} \left(\overline{(\alpha\kappa T^\beta)}_{i+\frac{1}{2}} (T_{i+1} - T_i) - \overline{(\alpha\kappa T^\beta)}_{i-\frac{1}{2}} (T_i - T_{i-1}) \right) \tag{16}$$

and all three types of averaging (11) are equivalent:

$$\overline{(\alpha\kappa T^\beta)}_{i+\frac{1}{2}} = \frac{\left(\alpha\kappa T^\beta\right)_i + \left(\alpha\kappa T^\beta\right)_{i+1}}{2}. \tag{17}$$

There are several ways to solve (6), that is, in the discrete case

$$F_i(\boldsymbol{T}) = 0, \ \forall i. \tag{18}$$

Replacing also the time derivative by a finite difference, (16) becomes

$$F_i(\boldsymbol{T}) = C_{V_i} \frac{T_i - T_i^{[t-\Delta t]}}{\Delta t} - \frac{1}{\Delta x^2} \left(\overline{(\alpha\kappa T^\beta)}_{i+\frac{1}{2}} (T_{i+1} - T_i) - \overline{(\alpha\kappa T^\beta)}_{i-\frac{1}{2}} (T_i - T_{i-1}) \right), \tag{19}$$

where $T_i^{[t-\Delta t]}$ is the temperature at the previous time level $t - \Delta t$. Note that by using temperature at the actual time level t in the spatial difference (the term in parentheses), we are aiming at implicit schemes, so that the time step Δt is not overrestricted by stability requirements.

3 Solving the Discrete Problem

3.1 Newton's Iteration

For simplicity, let's take in each equation the value of $(\alpha\, k/\beta)$ from the actual cell instead of using nodal averages at its endpoints. Then we have a system similar to (18) with the i-th equation being

$$F_i^*(\boldsymbol{T}) = C_{Vi}\,\frac{T_i - T_i^{[t-\Delta t]}}{\Delta t} - \alpha_i \kappa_i T_i^{\beta_i}\,\frac{T_{i-1} - 2\,T_i + T_{i+1}}{\Delta x^2}. \tag{20}$$

The Jacobian of such system is a tridiagonal matrix with the elements

$$J_{i,i} = \frac{\partial F_i^*}{\partial\, T_i} = \frac{C_{Vi}}{\Delta t} + 2\,\frac{\alpha_i\, k_i}{\Delta x^2}(\beta_i + 1)\,T_i^{\beta_i} \tag{21a}$$

$$J_{i,i\pm1} = \frac{\partial F_i^*}{\partial\, T_{i\pm1}} = -\,\frac{\alpha_i \kappa_i T_i^{\beta_i}}{\Delta x^2}. \tag{21b}$$

Using precise definition of nonlinear functional F given by (19) and approximate Jacobian (21), we can now perform the k-th iteration of Newton's method:

$$\boldsymbol{T}^{(k+1)} = \boldsymbol{T}^{(k)} - \boldsymbol{J}^{-1}\,\boldsymbol{F}(\boldsymbol{T}^{(k)}). \tag{22}$$

Keep in mind, that the superscript $^{(k)}$ stands for Newton's iteration, not for the evolution in time! Therefore, $T_i^{[t-\Delta t]}$ in (20) stays the same in all iterations at given time t, until the solution $\boldsymbol{T}$ at this time level has converged.

If we require an exact Jacobian, we can obtain it by deriving from the equation (19) using (11). Consequently, it can be deduced that:

$$J_{i,i} = \frac{\partial F_i}{\partial\, T_i} = \frac{C_{Vi}}{\Delta t} + \frac{1}{2}\alpha_{i+1}k_{i+1}T_{i+1}^{\beta_{i+1}} + \frac{1}{2}\alpha_{i-1}k_{i-1}T_{i-1}^{\beta_{i-1}} + (\beta_i+1)\alpha_i k_i T_i^{\beta_i} - \beta_i \alpha_i k_i T_i^{\beta_i - 1}\,(T_{i+1} + T_{i+1}) \tag{23a}$$

$$J_{i,i\pm1} = \frac{\partial F_i^*}{\partial\, T_{i\pm1}} = \frac{1}{2}\beta_{i\pm1}\alpha_{i\pm1}k_{i\pm1}T_{i\pm1}^{\beta_{i\pm1}-1}T_i - \frac{1}{2}\,(\beta_{i\pm1}+1)\,\alpha_{i\pm1}k_{i\pm1}T_{i\pm1}^{\beta_{i\pm1}} - \frac{1}{2}\alpha_i k_i T_i^{\beta_i} \tag{23b}$$

3.2 Boundary Conditions

We will enforce

$$\nabla T = 0 \tag{24}$$

on both ends of the domain. To do this on a mesh of N cells indexed from 1 to N (see Fig. 1), we formally introduce the ghost values

$$T_0 = T_1, \qquad\qquad T_{N+1} = T_N. \tag{25}$$

Inserting them into the equations (20) for the boundary cells ($i = 1$ resp. $i = N$), we get

$$F_1(\boldsymbol{T}) := C_{V_1}\,\frac{T_1 - T_1^{[t-\Delta t]}}{\Delta t} - \frac{\overline{(\alpha\kappa T^\beta)}_{\frac{3}{2}}(T_2 - T_1)}{\Delta x^2}, \tag{26a}$$

$$F_N(\boldsymbol{T}) := C_{V_N}\,\frac{T_N - T_N^{[t-\Delta t]}}{\Delta t} + \frac{\overline{(\alpha\kappa T^\beta)}_{N-\frac{1}{2}}(T_N - T_{N-1})}{\Delta x^2}, \tag{26b}$$

which immediately yields the first and last row of the Jacobian matrix

$$J_{1,1} = \frac{\partial F_1^*}{\partial T_1} = \frac{C_{V_1}}{\Delta t} + \frac{\alpha_1\kappa_1}{\Delta x^2}(\beta_1 + 1)\,T_1^{\beta_1}, \tag{27a}$$

$$J_{1,2} = \frac{\partial F_1^*}{\partial T_2} = -\,\frac{\alpha_1\kappa_1 T_1^{\beta_1}}{\Delta x^2}, \tag{27b}$$

$$J_{N,N-1} = \frac{\partial F_N^*}{\partial T_{N-1}} = -\,\frac{\alpha_N\kappa_N T_N^{\beta_N}}{\Delta x^2}, \tag{27c}$$

$$J_{N,N} = \frac{\partial F_N^*}{\partial T_N} = \frac{C_{V_N}}{\Delta t} + \frac{\alpha_N\kappa_N}{\Delta x^2}(\beta_N + 1)\,T_N^{\beta_N}. \tag{27d}$$

4 The Linear Case

Let us now look at the special case of linear heat conduction, that is,

$$\beta_i \equiv 0\ ,\forall i \tag{28}$$

and for simplicity let us also use a constant coefficient

$$a = \alpha_i\,k_i,\ \forall i \tag{29}$$

and constant heat capacity. In other words, we are solving

$$C_V\,\frac{d\,T}{dt} = a\,\frac{d^2\,T}{dx^2}, \tag{30}$$

which we discretize (again aiming at the implicit scheme), define the functional

$$F_i = C_V\,\frac{T_i - T_i^{[t-\Delta t]}}{\Delta t} - a\,\frac{T_{i-1} - 2\,T_i + T_{i+1}}{\Delta x^2} \tag{31}$$

and calculate

$$F_i = 0,\ \forall i. \tag{32}$$

If Newton's iteration was to be used, the Jacobian would be a constant tridiagonal matrix with elements

$$J_{i,i} = \frac{\partial F_i}{\partial T_i} = \frac{C_V}{\Delta t} + 2\,\frac{a}{\Delta x^2} \tag{33a}$$

$$J_{i,i\pm1} = \frac{\partial F_i}{\partial T_{i\pm1}} = -\,\frac{a}{\Delta x^2}. \tag{33b}$$

However, for such a simple problem it is not necessary to iterate or even explicitly formulate the Jacobian. We can simply rewrite (basically just reorder) (32) with (31) as

$$T_i - \frac{a}{C_V}\,\frac{\Delta t}{\Delta x^2}\,(T_{i-1} - 2\,T_i + T_{i+1}) = T_i^{[t-\Delta t]}, \tag{34}$$

that is,

$$-\frac{a}{C_V}\,\frac{\Delta t}{\Delta x^2}\,T_{i-1} + \left(1 + 2\frac{a}{C_V}\,\frac{\Delta t}{\Delta x^2}\right)T_i - \frac{a}{C_V}\,\frac{\Delta t}{\Delta x^2}\,T_{i+1} = T_i^{[t-\Delta t]}, \tag{35}$$

and denoting

$$b = \frac{a}{C_V} \tag{36}$$

we recover

$$-b\,\frac{\Delta t}{\Delta x^2}\,T_{i-1} + \left(1 + 2\,b\,\frac{\Delta t}{\Delta x^2}\right)T_i - b\,\frac{\Delta t}{\Delta x^2}\,T_{i+1} = T_i^{[t-\Delta t]}, \tag{37}$$

which yields a simple linear system

$$\boldsymbol{A}\,\boldsymbol{T} = \boldsymbol{T}^{[t-\Delta t]}, \tag{38}$$

with a tridiagonal constant matrix $\boldsymbol{A}$ as it can be found in many textbooks on implicit methods for parabolic problems.