Generalised Finite Difference Method for linear heatconduction problems in 2D

Contents

1 Basic Equations

for the calculation of the temperature field $T(\boldsymbol{x}, t)$ on an closed area $\Omega \subseteq \mathbb{R}^2$ with the boundary $\partial\Omega$, the following equations apply:

$$\begin{aligned}
\dot{T} - a\,(T_{xx} + T_{yy}) &= 0 \quad && \boldsymbol{x} \in \Omega \\
T &= \bar{T} \quad && \boldsymbol{x} \in \partial\Omega_d \quad \text{(Dirichlet boundary condition)} \\
-\lambda\,(T_x\,n_x + T_y\,n_y) &= \dot{q} \quad && \boldsymbol{x} \in \partial\Omega_n \quad \text{(Neumann boundary condition)}
\end{aligned}$$

The **thermal diffusivity** a is composed as follows:

$$a = \frac{\lambda}{\rho\,c}$$

Here is ρ the **material density**, λ the **thermal conductivity** and c the specific heat capacity of the material.

2 Moving-Least-Square Approximation of the temperature field

The temperature field is approximated at the point $\boldsymbol{x}_i$ by a second degree Taylor polynomial $\tilde{T}_i$. At each position x_j with $j \in S_i$, the value of the series expansion is given by

$$\begin{aligned}\tilde{T}_i(\boldsymbol{x}_j) = {} & T(\boldsymbol{x}_i) + T_{,x}(\boldsymbol{x}_i)\,(x_j - x_i) + T_{,y}(\boldsymbol{x}_i)\,(y_j - y_i) + \\ & \frac{1}{2}\,T_{,xx}(\boldsymbol{x}_i)\,(x_j - x_i)^2 + T_{,xy}(\boldsymbol{x}_i)\,(x_j - x_i)\,(y_j - y_i) + \frac{1}{2}\,T_{,yy}(\boldsymbol{x}_i)\,(y_j - y_i)^2.\end{aligned}$$

This gives for all x_j with $j \in S_i$ and $|S| = n$ the numerical error:

$$e_{ij}^T = \tilde{T}_i(\boldsymbol{x}_j) - T(\boldsymbol{x}_j)$$

With the definitions

$$\boldsymbol{e}_i^T = \begin{pmatrix} e_{i1}^T \\ \vdots \\ e_{in}^T \end{pmatrix} \in \mathbb{R}^n$$

$$\boldsymbol{a}_i = \begin{pmatrix} T_i \\ T_{i,x} \\ T_{i,y} \\ T_{i,xx} \\ T_{i,xy} \\ T_{i,yy} \end{pmatrix} \in \mathbb{R}^6$$

$$\boldsymbol{b}_i^T = \begin{pmatrix} T_{j1} \\ \vdots \\ T_{jn} \end{pmatrix} \in \mathbb{R}^n$$

$$\boldsymbol{D}_i = \begin{pmatrix} 1 & \Delta x_{i1} & \Delta y_{i1} & \frac{1}{2}\,\Delta x_{i1}^2 & \Delta x_{i1}\,\Delta y_{i1} & \frac{1}{2}\,\Delta y_{i1}^2 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & \Delta x_{in} & \Delta y_{in} & \frac{1}{2}\,\Delta x_{in}^2 & \Delta x_{in}\,\Delta y_{in} & \frac{1}{2}\,\Delta y_{in}^2 \end{pmatrix} \in \mathbb{R}^{n\times 6}$$

follows:

$$\boldsymbol{e}_i^T = \boldsymbol{D}_i \cdot \boldsymbol{a}_i - \boldsymbol{b}_i^T$$

3 Discretisation of inner area points

Considering a 1st order time discretisation

$$\dot{T}_i = f\,\left(T_i - T_i^{k-1}\right),$$

the numerical error can be described as follows:

$$f\,T_i - a\,(T_{i,xx} + T_{i,yy}) - f\,T_i^{k-1} = e_{i\Omega}$$

This leads to the following minimisation problem:

$$\min J_i = \sum_{j \in S_i} W_{ij}^2\,\left(e_{ij}^T\right)^2 + W_\Omega^2\,e_{i\Omega}^2$$

Summarised in matrix vector notation:

$$\boldsymbol{G}_\Omega = \begin{pmatrix} f & 0 & 0 & -a & 0 & -a \end{pmatrix} \in \mathbb{R}^{1\times 6}$$

$$\underbrace{\begin{pmatrix} \boldsymbol{e}_i^T \\ e_{i\Omega} \end{pmatrix}}_{\boldsymbol{e}_i \in \mathbb{R}^{n+1}} = \underbrace{\begin{pmatrix} \boldsymbol{D}_i \\ \boldsymbol{G}_\Omega \end{pmatrix}}_{\boldsymbol{M}_i \in \mathbb{R}^{(n+1)\times 6}} \cdot \boldsymbol{a}_i - \underbrace{\begin{pmatrix} \boldsymbol{b}_i^T \\ f\,T_i^{k-1} \end{pmatrix}}_{\boldsymbol{b}_i \in \mathbb{R}^{n+1}}$$

The solution of the minimisation problem with the weight matrix

$$\boldsymbol{W} = \operatorname{diag}\left(W_{i1}, \cdots, W_{in}, W_\Omega\right) \in \mathbb{R}^{(n+1)\times(n+1)} \tag{3.1}$$

is:

$$\boldsymbol{a}_i = \underbrace{\left[\left(\boldsymbol{M}_i^T \cdot \boldsymbol{W}_i^2 \cdot \boldsymbol{M}_i\right)^{-1} \cdot \boldsymbol{M}_i^T \cdot \boldsymbol{W}_i^2\right]}_{\boldsymbol{C}_i \in \mathbb{R}^{6\times(n+1)}} \cdot \boldsymbol{b}_i \tag{3.2}$$

This results in the system of equations for determining the temperature T_i at the timestep k:

$$T_i = \sum_{j \in S_i} C_i \left[0, 0 : n\right] \, T_j + C_i \left[0, n\right] \, f \, T_i^{k-1}$$

Or:

$$T_i \left(1 - C_i \left[0, 0\right]\right) - \sum_{\substack{j \in S_i \\ j \neq i}} C_i \left[0, 1 : n\right] \, T_j = C_i \left[0, n\right] \, f \, T_i^{k-1}$$

4 Discretisation of Dirichlet boundary conditions

For Dirichelt boundary conditions($\boldsymbol{x} \in \partial\Omega_d$), the following numerical errors are minimised:

$$\begin{aligned} f\,T_i - a\,(T_{i,xx} + T_{i,yy}) - f\,T_i^{k-1} &= e_{i\Omega} \\ T_i - \bar{T}_i &= e_{i\partial\Omega_d} \end{aligned}$$

This leads to the following minimisation problem:

$$\min J_i = \sum_{j \in S_i} W_{ij}^2\,\left(e_{ij}^T\right)^2 + W_\Omega^2\,e_{i\Omega}^2 + W_{\partial\Omega_d}^2\,e_{i\partial\Omega_d}^2$$

Summarised in matrix vector notation:

$$\boldsymbol{G}_{\partial\Omega_d} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 \end{pmatrix} \in \mathbb{R}^{1\times 6}$$

$$\underbrace{\begin{pmatrix} \boldsymbol{e}_i^T \\ e_{i\Omega} \\ e_{i\partial\Omega_d} \end{pmatrix}}_{\boldsymbol{e}_i \in \mathbb{R}^{n+2}} = \underbrace{\begin{pmatrix} \boldsymbol{D}_i \\ \boldsymbol{G}_\Omega \\ \boldsymbol{G}_{\partial\Omega_d} \end{pmatrix}}_{\boldsymbol{M}_i \in \mathbb{R}^{(n+2)\times 6}} \cdot \boldsymbol{a}_i - \underbrace{\begin{pmatrix} \boldsymbol{b}_i^T \\ f\,T_i^{k-1} \\ \bar{T}_i \end{pmatrix}}_{\boldsymbol{b}_i \in \mathbb{R}^{n+2}}$$

The solution of the minimisation problem with the weight matrix

$$\boldsymbol{W} = \operatorname{diag}\left(W_{i1}, \cdots, W_{in}, W_\Omega, W_{\partial\Omega_d}\right) \in \mathbb{R}^{(n+2)\times(n+2)} \tag{4.1}$$

is:

$$\boldsymbol{a}_i = \underbrace{\left[\left(\boldsymbol{M}_i^T \cdot \boldsymbol{W}_i^2 \cdot \boldsymbol{M}_i\right)^{-1} \cdot \boldsymbol{M}_i^T \cdot \boldsymbol{W}_i^2\right]}_{\boldsymbol{C}_i \in \mathbb{R}^{6\times(n+2)}} \cdot \boldsymbol{b}_i \tag{4.2}$$

This results in the system of equations for determining the temperature T_i at the timestep k:

$$T_i = \sum_{j \in S_i} C_i\left[0, 0:n\right]\, T_j + C_i\left[0, n\right]\, f\, T_i^{k-1} + C_i\left[0, n+1\right]\, \bar{T}_i$$

Or:

$$T_i \left(1 - C_i\left[0, 0\right]\right) - \sum_{\substack{j \in S_i \\ j \neq i}} C_i\left[0, 1:n\right]\, T_j = C_i\left[0, n\right]\, f\, T_i^{k-1} + C_i\left[0, n+1\right]\, \bar{T}_i$$

5 Discretisation of Neumann boundary conditions

For Neumann boundary conditions($\boldsymbol{x} \in \partial\Omega_n$), the following numerical errors are minimised:

$$\begin{aligned} f\,T_i - a\,(T_{i,xx} + T_{i,yy}) - f\,T_i^{k-1} &= e_{i\Omega} \\ -\lambda\,(T_{i,x}\,n_x + T_{i,y}\,n_y) - \dot{q}_i &= e_{i\partial\Omega_n} \end{aligned}$$

This leads to the following minimisation problem:

$$\min J_i = \sum_{j\in S_i} W_{ij}^2\,\left(e_{ij}^T\right)^2 + W_\Omega^2\,e_{i\Omega}^2 + W_{\partial\Omega_d}^2\,e_{i\partial\Omega_n}^2$$

Summarised in matrix vector notation:

$$\boldsymbol{G}_{\partial\Omega_n} = \begin{pmatrix} 0 & -\lambda\,n_x & -\lambda\,n_y & 0 & 0 & 0 \end{pmatrix} \in \mathbb{R}^{1\times 6}$$

$$\underbrace{\begin{pmatrix} \boldsymbol{e}_i^T \\ e_{i\Omega} \\ e_{i\partial\Omega_n} \end{pmatrix}}_{\boldsymbol{e}_i\in\mathbb{R}^{n+2}} = \underbrace{\begin{pmatrix} \boldsymbol{D}_i \\ \boldsymbol{G}_\Omega \\ \boldsymbol{G}_{\partial\Omega_n} \end{pmatrix}}_{\boldsymbol{M}_i\in\mathbb{R}^{(n+2)\times 6}} \cdot\,\boldsymbol{a}_i - \underbrace{\begin{pmatrix} \boldsymbol{b}_i^T \\ f\,T_i^{k-1} \\ \dot{q}_i \end{pmatrix}}_{\boldsymbol{b}_i\in\mathbb{R}^{n+2}}$$

The solution of the minimisation problem with the weight matrix

$$\boldsymbol{W} = \mathrm{diag}\,(W_{i1},\cdots,W_{in},W_\Omega,W_{\partial\Omega_n}) \in \mathbb{R}^{(n+2)\times(n+2)} \tag{5.1}$$

is:

$$\boldsymbol{a}_i = \underbrace{\left[\left(\boldsymbol{M}_i^T\cdot\boldsymbol{W}_i^2\cdot\boldsymbol{M}_i\right)^{-1}\cdot\boldsymbol{M}_i^T\cdot\boldsymbol{W}_i^2\right]}_{\boldsymbol{C}_i\in\mathbb{R}^{6\times(n+2)}}\cdot\boldsymbol{b}_i \tag{5.2}$$

This results in the system of equations for determining the temperature T_i at the timestep k:

$$T_i = \sum_{j \in S_i} C_i\left[0, 0:n\right]\, T_j + C_i\left[0, n\right]\, f\, T_i^{k-1} + C_i\left[0, n+1\right]\, \dot{q}_i$$

Or:

$$T_i\left(1 - C_i\left[0, 0\right]\right) - \sum_{\substack{j \in S_i \\ j \neq i}} C_i\left[0, 1:n\right]\, T_j = C_i\left[0, n\right]\, f\, T_i^{k-1} + C_i\left[0, n+1\right]\, \dot{q}_i$$