Derivation of tri-diagonal matrix for a General Implicit Heat Flow problem

T.J. Heimovaara

April 30, 2019

1 Introduction

In the lecture slides you can find a derivation of the implicit tridiagonal matrix for the heat flow problem using a Dirichlet problem. A small problem you face with this approach is that this solution is not very easy to develop in to a general implicit solver where we can use so-called higher order implicit solvers such as the Crank Nicholson approach based on the trapezoidal rule, which is second order accurate compared to the first order implict Euler method.

A general approach is possible when we use the Robin boundary condition in stead for the lower and the top boundary conditions which for the top boundary can be written as:

$$q_{h_{top}} = -k_{rob_{top}}(T_{top} - T)$$

and for the bottom boundary as:

2 General solution for 1D flow

The general heat flow equation is

$$\boldsymbol{q}_h = -\boldsymbol{\lambda}\nabla T$$

which combined with the heat balance equation is gives

$$\zeta\frac{\partial T}{\partial t} = -\nabla \cdot \boldsymbol{q}_H.$$

For a 1D vertical implementation on a staggered grid we can discretize these equations as follows.

$$q_{h_{i-\frac{1}{2}}} = -\lambda_{i-\frac{1}{2}}\frac{\Delta T_{i-1}}{\Delta z_{i-1}} = -\lambda_{i-\frac{1}{2}}\frac{T_i - T_{i-1}}{\Delta z_{i-1}}$$

and

$$\zeta_i \frac{T_i^{m+1} - T_i^m}{\Delta t} = -\frac{q_{h_{i+\frac{1}{2}}} - q_{h_{i-\frac{1}{2}}}}{\Delta z_{i-\frac{1}{2}}} =$$

$$-\left(\frac{\frac{-\lambda_{i+\frac{1}{2}}(T_{i+1}-T_i)}{\Delta z_i} - \frac{-\lambda_{i-\frac{1}{2}}(T_i - T_{i-1})}{\Delta z_{i-1}}}{\Delta z_{i-\frac{1}{2}}} \right) =$$

$$\frac{\lambda_{i-\frac{1}{2}}}{\Delta z_{i-1}\Delta z_{i-\frac{1}{2}}} T_{i-1} - \left(\frac{\lambda_{i-\frac{1}{2}}}{\Delta z_{i-1}\Delta z_{i-\frac{1}{2}}} + \frac{\lambda_{i+\frac{1}{2}}}{\Delta z_i \Delta z_{i-\frac{1}{2}}}\right) T_i + \frac{\lambda_{i+\frac{1}{2}}}{\Delta z_i \Delta z_{i-\frac{1}{2}}} T_{i+1}$$

$$\alpha_i T_{i-1} + \beta_i T_i + \gamma_i T_{i+1}$$

vectorizing this for all inner nodes leads to

$$\boldsymbol{M}(\boldsymbol{T}^{m+1} - \boldsymbol{T}^m) = \boldsymbol{K}\boldsymbol{T}.$$

3 Top and bottom boundary condition

For the top and the bottom boundary condition we use the Robin condition defined above. Discretizing for the bottom condition we obtain

$$\zeta_1 \frac{T_1^{m+1} - T_1^m}{\Delta t} = -\frac{q_{H_{\frac{1}{2}}} - q_{H_{-\frac{1}{2}}}}{\Delta z_{\frac{1}{2}}} = -\left(\frac{\frac{-\lambda_{1\frac{1}{2}}(T_2 - T_1)}{\Delta z_1} + k_{rob_{bot}}(T_1 - T_{bnd_{bot}})}{\Delta z_{\frac{1}{2}}} \right) =$$

$$-\left(\frac{k_{rob_{bot}}}{\Delta z_{\frac{1}{2}}} + \frac{\lambda_{1\frac{1}{2}}}{\Delta z_1 \Delta z_{\frac{1}{2}}}\right) T_1 + \frac{\lambda_{1\frac{1}{2}}}{\Delta z_1 \Delta z_{\frac{1}{2}}} T_2 + \frac{k_{rob_{bot}}}{\Delta z_{\frac{1}{2}}} T_{bnd_{bot}}$$

which can be written as

$$\zeta_1 \frac{T_1^{m+1} - T_1^m}{\Delta t} = \beta_1 T_1 + \gamma_1 T_2 + y_1$$

for a zero flow boundary condition we can set $k_{rob_{bot}} = 0$.
A similar derivation occurs for the top boundary condition leading to

$$\zeta_n \frac{T_n^{m+1} - T_n^m}{\Delta t} = -\left(\frac{-k_{rob_{top}}(T_{bnd_{top}} - T_n) - \frac{-\lambda_{n-\frac{1}{2}}(T_n - T_{n-1})}{\Delta z_{n-1}}}{\Delta z_{n-\frac{1}{2}}} \right) =$$

$$\left(\frac{\lambda_{n-\frac{1}{2}}}{\Delta z_{n-1}\Delta z_{n-\frac{1}{2}}} T_{n-1} - \left(\frac{\lambda_{n-\frac{1}{2}}}{\Delta z_{n-1}\Delta z_{n-\frac{1}{2}}} + \frac{k_{rob_{top}}}{\Delta z_{n-\frac{1}{2}}}\right) T_n + \frac{k_{rob_{top}}}{\Delta z_{n-\frac{1}{2}}} T_{bnd_{top}}\right.$$

$$\alpha_n T_{n-1} + \beta_n T_n + y_n.$$

4 Explicit, implicit and semi-implicit solutions

It is clear that the general solution including top and boundary conditions is the following matrix equation

$$\boldsymbol{M}(\boldsymbol{T}^{m+1} - \boldsymbol{T}^m) = \boldsymbol{K}\boldsymbol{T} + \boldsymbol{y}$$

which lead to the following explicit Euler solution:

$$\boldsymbol{T}^{m+1} = \boldsymbol{T}^m + \frac{\boldsymbol{K}\boldsymbol{T}^m}{\boldsymbol{M}} + \boldsymbol{y}^m.$$

We also can develop an implicit solution as follows

$$(\boldsymbol{M} + \boldsymbol{K})\boldsymbol{T}^{m+1} = \boldsymbol{M}\boldsymbol{T}^m + \boldsymbol{y}^{m+1}.$$

Finally the half implicit (also known as the Crank Nicholson solution) which is based on the trapezoidal rule can be developed as:

$$\boldsymbol{M}(\boldsymbol{T}^{m+1} - \boldsymbol{T}^m) = \boldsymbol{K}\frac{\boldsymbol{T}^{m+1} + \boldsymbol{T}^m}{2} + \frac{\boldsymbol{y}^{m+1} + \boldsymbol{y}^m}{2}$$

leading to

$$(\boldsymbol{M} - \frac{\boldsymbol{K}}{2})\boldsymbol{T}^{m+1} = (\boldsymbol{M} + \frac{\boldsymbol{K}}{2})\boldsymbol{T}^m + \frac{\boldsymbol{y}^{m+1} + \boldsymbol{y}^m}{2}$$

which is considered to be the most efficient and stable solution for this problem. A challenge however is that for large grid problems the matrices tend to become very large. Luckily they are sparse matrices for which most modern programming languages have efficient tools available.