CSE380 TOOLS AND TECHNIQUES FOR COMPUTATIONAL SCIENCE ASSIGNMENT 4 [MODELING DOCUMENT]

MOHAMMAD AFZAL SHADAB

Abstract. This is a *modeling document* for the application to solve the steady-state heat equation in one- and two-dimensions. The document highlights the governing equations, boundary conditions, numerical approximations, and high-level pseudocode implemented in the solver.

1. Governing equations and boundary conditions

The steady-state heat equation with a constant coefficient in two dimensions is given by:

$$\boxed{-k\nabla^2 T(x,y) = q(x,y) \quad \forall \Omega \in [0,L]\times[0,H]} \tag{1}$$

where k is the thermal conductivity (W/K), $T(x,y)$ is the material temperature (K), $q(x,y)$ is a heat source term (W/m^2) and $\Omega \subset \mathbb{R}^2$ is the domain. This is Poisson equation which is a type of elliptic partial differential equations. This linear boundary value problem is subjected to either *Dirichlet boundary conditions* (T specified) following *maximum principle* or combinations of *Neumann* (∇T specified), *Dirichlet* and *Robin boundary conditions*. To begin with, let's specify Dirichlet boundary conditions:

$$\begin{aligned} T(0,y) &= T_{analytical}(0,y) \\ T(L,y) &= T_{analytical}(L,y) \\ T(x,0) &= T_{analytical}(x,0) \\ T(x,H) &= T_{analytical}(x,H) \end{aligned} \tag{2}$$

where $T_{analytical}$ is evaluated using MASA.

Date: October 29, 2020.

2. Assumptions

We'll start with the assumptions for the derivation of equation 1 from the law of conservation of energy in Eulerian framework 3 [2]:

$$\frac{\partial}{\partial t}(\rho e) + \nabla \cdot (\rho e \mathbf{u}) = \mathbf{T} : \mathbf{D} - \nabla \cdot \mathbf{q} + r \tag{3}$$

where e is internal energy per unit mass, $\mathbf{T} : \mathbf{D}$ is the strain heating, $\mathbf{q}$ is the heat flux and r is the volumetric source / sink.

- Continuum assumption
- Steady-state $(\partial(.)/\partial t = 0)$
- No advection $\mathbf{u} = \mathbf{0}$
- No source/sink term $r = 0$
- Validity of *Fourier's law of heat conduction*, i.e., $\mathbf{q} = -k\nabla T$
- No strain heating, i.e., $\mathbf{T} : \mathbf{D} = 0$, where $\mathbf{T}$ is the stress tensor and $\mathbf{D}$ is the deformation tensor
- Constant coefficient of thermal conductivity k

Moving towards the assumptions to make the numerical implementation easier:

- Square domain $L \equiv H$
- Uniform grid, i.e. $N_x = N_y = N$, where N_x and N_y are the cells in x and y directions respectively. So, $\Delta x = \Delta y = h$
- The Dirichlet boundary conditions are implemented in form of constant temperatures in the ghost cells adjacent to a corresponding boundary, i.e., $T_{0,1} = T_{analytical}(x_0, y_1)$

3. Nomenclature

For 1D, the mesh numbering is simple and straight forward, as shown in figure 1. The scheme is cell based, where cell centers (x_c, y_c) are referred to most of the times.

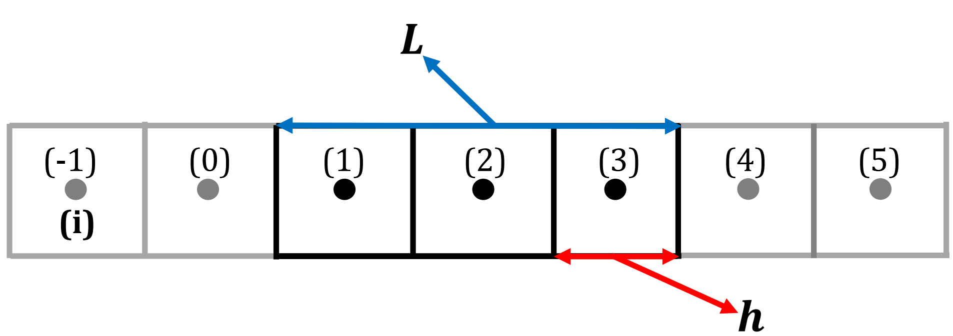

Figure 1. 1D mesh with 3 cells, grey ghost cells, and i indexing

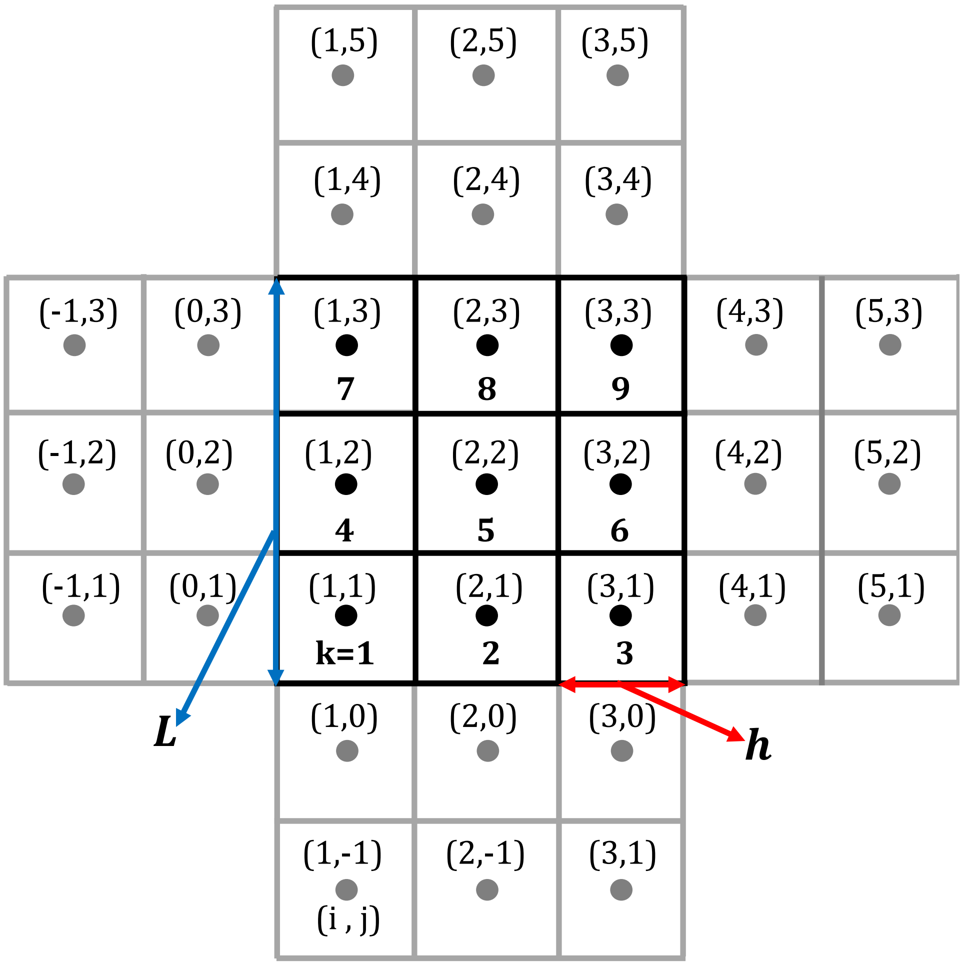

FIGURE 2. 2D 3×3 uniform mesh with grey ghost cells, (i, j) indexing, and universal k indexing

For 2D mesh, the situation is slightly sophisticated as two indices (i, j) come to picture correspondingly in x and y directions illustrated in figure 2. So, using a new numbering system for converting (i, j) into one index k which spans x direction cells first then marches in y direction,

$$
\begin{aligned}
k &= i + (j-1)N_x, \mathrm{k} \in \{1, 2, ..., N_x * N_y\} \\
k\%N_x &= i \qquad \text{(Remainder)} \\
k/N_x &= j - 1 \qquad \text{(Integer division)}
\end{aligned}
\tag{4}
$$

Be careful as the ghost cells are not considered in the k numbering and (i, j) points referring to ghost cell centers indices.

4. Numerical Methods

Finite difference approximation has been implemented considering the ease in implementation when compared with other discretization methods such as finite volumes, finite element, etc [3]. Rewriting equation 1 in expanded form for 1D

$$-k\frac{\partial^2 T(x)}{\partial x^2} = q(x) \tag{5}$$

and for 2D

$$-k\left(\frac{\partial^2 T(x,y)}{\partial x^2} + \frac{\partial^2 T(x,y)}{\partial y^2}\right) = q(x,y) \tag{6}$$

4.1. **Second order finite difference approximation.**

Using the Taylor's expansion, the second order central difference approximation for second order derivative in x direction can be written

$$\frac{\partial^2 T}{\partial x^2} = \frac{T_{i+1} - 2T_i + T_{i-1}}{\Delta x^2} + \mathcal{O}(h^2) \tag{7}$$

where subscript i is the cell centered value of cell i.

4.1.1. *1D.*

Equation 7 can be substituted into equation 5, to give

$$-k\left(\frac{T_{i+1} - 2T_{i,j} + T_{i-1}}{\Delta x^2}\right) + \mathcal{O}(h^2) = q_i \tag{8}$$

Inserting $\lambda = -k/\Delta x^2$ and dividing by λ, we get

$$T_{i+1} - 2T_i + T_{i-1} + \mathcal{O}(h^4) = q_i/\lambda \tag{9}$$

Neglecting the truncation error $\mathcal{O}(h^4)$ and writing in matrix form for a 3 cell grid shown in figure 1 after implementing the penalty approach for the boundary terms, we get

$$\begin{pmatrix} -2 & 1 & \\ 1 & -2 & 1 \\ & 1 & -2 \end{pmatrix} \begin{bmatrix} T_1 \\ T_2 \\ T_3 \end{bmatrix} = \begin{bmatrix} q_1/\lambda - T_0 \\ q_2/\lambda \\ q_3/\lambda - T_4 \end{bmatrix} \Rightarrow \mathbf{AT{=}B} \tag{10}$$

For a N-cell grid, the matrix equation takes the following form,

$$\mathbf{A} = \begin{pmatrix} -2 & 1 & & & & \\ 1 & -2 & 1 & & & \\ & 1 & -2 & 1 & & \\ & & \ddots & \ddots & \ddots & \\ & & & 1 & -2 & 1 \\ & & & & 1 & -2 \end{pmatrix}, \mathbf{T} = \begin{bmatrix} T_1 \\ T_2 \\ \vdots \\ T_{N-1} \\ T_N \end{bmatrix}, \mathbf{B} = \begin{bmatrix} q_1/\lambda - T_0 \\ q_2/\lambda \\ \vdots \\ q_{N-1}/\lambda \\ q_N/\lambda - T_{N+1} \end{bmatrix} \tag{11}$$

So, the resulting matrix $\mathbf{A}$ is tridiagonal (3 diagonals).

$$\mathbf{A} = \begin{pmatrix} \times & \times & & & & \\ \times & \times & \times & & & \\ & \times & \times & \times & & \\ & & \ddots & \ddots & \ddots & \\ & & & \times & \times & \times \\ & & & & \times & \times \end{pmatrix} \tag{12}$$

4.1.2. *2D.* We can write

$$-k\left(\frac{T_{i+1,j} - 2T_{i,j} + T_{i-1,j}}{\Delta x^2} + \frac{T_{i,j+1} - 2T_{i,j} + T_{i,j-1}}{\Delta y^2}\right) + \mathcal{O}(h^2) = q_{i,j} \tag{13}$$

where i and j are the positions of cell center in x and y directions. Implementing, $\Delta x = \Delta y = h$ from assumptions and $-k/h^2 = \lambda$

$$T_{i+1,j} + T_{i-1,j} + T_{i,j+1} + T_{i,j-1} - 4T_{i,j} + \mathcal{O}(h^4) = q_{i,j}/\lambda \tag{14}$$

Since we have disconnected elements at $i = 1, N$, using k notation for same row, i.e, when $(k+1)/N = (k-1)/N = k/N = i$

$$T_{k,k+1} + T_{k,k-1} + T_{k,k+N_x} + T_{k,k-N_x} - 4T_{k,k} + \mathcal{O}(h^4) = q_k/\lambda \quad \text{(for same row)} \tag{15}$$

Using penalty approach for boundary terms, for a 3×3 uniform grid, the resulting matrix equation takes the form

$$\begin{pmatrix} -4 & 1 & & 1 & & & & & \\ 1 & -4 & 1 & & 1 & & & & \\ & 1 & -4 & & & 1 & & & \\ 1 & & & -4 & 1 & & 1 & & \\ & 1 & & 1 & -4 & 1 & & 1 & \\ & & 1 & & 1 & -4 & & & 1 \\ & & & 1 & & & -4 & 1 & \\ & & & & 1 & & 1 & -4 & 1 \\ & & & & & 1 & & 1 & -4 \end{pmatrix} \begin{bmatrix} T_1 \\ T_2 \\ T_3 \\ T_4 \\ T_5 \\ T_6 \\ T_7 \\ T_8 \\ T_9 \end{bmatrix} = \begin{bmatrix} q_1/\lambda - T_{1,0} - T_{0,1} \\ q_2/\lambda - T_{2,0} \\ q_3/\lambda - T_{4,1} - T_{3,0} \\ q_4/\lambda - T_{0,2} \\ q_5/\lambda \\ q_6/\lambda - T_{4,2} \\ q_7/\lambda - T_{0,3} - T_{1,4} \\ q_8/\lambda - T_{2,4} \\ q_9/\lambda - T_{4,3} - T_{3,4} \end{bmatrix}$$

$$\Rightarrow \mathbf{AT} = \mathbf{B} \tag{16}$$

For a $N \times N$ grid, the matrix equation takes the following form 17,

$$
\begin{pmatrix}
-4 & 1 & \Leftarrow (N-1) \Rightarrow & 1 & & & \\
1 & -4 & 1 & \Leftarrow (N-1) \Rightarrow & 1 & & \\
 & 1 & -4 & 1 & & \ddots & \\
1 & \Leftarrow (N-1) \Rightarrow & 1 & \ddots & \ddots & & 1 \\
 & 1 & & \ddots & \ddots & 1 & \\
 & & \ddots & & 1 & -4 & 1 \\
 & & & 1 & \Leftarrow (N-1) \Rightarrow & 1 & -4
\end{pmatrix}
\begin{bmatrix} T_1 \\ T_2 \\ T_3 \\ \vdots \\ T_N \\ T_{N+1} \\ \vdots \\ T_{N*N-1} \\ T_{N*N} \end{bmatrix}
=
\begin{bmatrix} q_1/\lambda - T_{1,0} - T_{0,1} \\ q_2/\lambda - T_{2,0} \\ q_3/\lambda - T_{3,0} \\ \vdots \\ q_N/\lambda - T_{N+1,1} - T_{N-1,0} \\ q_{N+1}/\lambda - T_{0,2} \\ \vdots \\ q_{N*N-1}/\lambda - T_{N-1,N+1} \\ q_{N*N}/\lambda - T_{N,N+1} - T_{N+1,N} \end{bmatrix}
\tag{17}
$$

$$
\Rightarrow \mathbf{AT} = \mathbf{B} \tag{18}
$$

where "$\Leftarrow (N-1) \Rightarrow$" is the distances between the elements. So, the resulting matrix $\mathbf{A}$ is banded with 5 diagonals: principal diagonal, offdiagonals at distance 1 and at distance N from principal diagonal as shown below:

$$
\mathbf{A} =
\begin{pmatrix}
\times & \times & \Leftarrow (N-1) \Rightarrow & \times & & & \\
\times & \times & \times & \Leftarrow (N-1) \Rightarrow & \times & & \\
 & \times & \times & \times & & \ddots & \\
\times & \Leftarrow (N-1) \Rightarrow & \times & \ddots & \ddots & & \times \\
 & \times & & \ddots & \ddots & 1 & \\
 & & \ddots & & \times & \times & \times \\
 & & & \times & \Leftarrow (N-1) \Rightarrow & \times & \times
\end{pmatrix}
$$

For a general matrix,

$$A = \begin{cases} A_{l-N_x,l} = \begin{cases} 1 & l/N_x = 0 \\ 0 & \text{else (bottom boundary)} \end{cases} \\ A_{l-1,l} = \begin{cases} 1 & l\%N_x > 1 \\ 0 & \text{else (left boundary)} \end{cases} \\ A_{l,l} = -4 \\ A_{l+1,l} = \begin{cases} 1 & l\%N_x \ = 0 \\ 0 & \text{else (right boundary)} \end{cases} \\ A_{l-N_x,l} = \begin{cases} 1 & l/N_x = N_y - 1 \\ 0 & \text{else (top boundary)} \end{cases} \end{cases} \tag{19}$$

4.2. **Fourth order finite difference approximation.** Using the Taylor's expansion, the fourth order central difference approximation for second order derivative in x direction can be written as:

$$\frac{\partial^2 T}{\partial x^2} = \frac{-T_{i+2} + 16T_{i+1} - 30T_i + 16T_{i-1} - T_{i-2}}{\Delta 12x^2} + \mathcal{O}(h^4) \tag{20}$$

where subscript i is the cell centered value of cell i.

4.2.1. *1D.* Equation 20 can be substituted into equation 5, to give

$$-k\left(\frac{-T_{i+2} + 16T_{i+1} - 30T_i + 16T_{i-1} - T_{i-2}}{\Delta 12x^2}\right) + \mathcal{O}(h^4) = q_i \tag{21}$$

Inserting $\lambda = -k/\Delta x^2$ and dividing by λ, we get

$$-T_{i+2} + 16T_{i+1} - 30T_i + 16T_{i-1} - T_{i-2} + \mathcal{O}(h^6) = q_i/\lambda \tag{22}$$

Neglecting the truncation error $\mathcal{O}(h^6)$ and writing in matrix form for a 3 cell grid shown in figure 1 after implementing the penalty approach for the boundary terms, we get

$$\begin{pmatrix} -30 & 16 & -1 \\ 16 & -30 & 16 \\ -1 & 16 & -30 \end{pmatrix} \begin{bmatrix} T_1 \\ T_2 \\ T_3 \end{bmatrix} = \begin{bmatrix} q_1/\lambda - (16T_0 - T_{-1}) \\ q_2/\lambda - (-T_{-1} - T_4) \\ q_3/\lambda - (16T_4 - T_5) \end{bmatrix} \Rightarrow \mathbf{AT{=}B} \tag{23}$$

For a N cell grid, the matrix equation takes the form

$$\begin{pmatrix} -30 & 16 & -1 & & & & \\ 16 & -30 & 16 & -1 & & & \\ -1 & 16 & \ddots & \ddots & \ddots & & \\ & -1 & \ddots & \ddots & 16 & -1 & \\ & & \ddots & 16 & -30 & 16 & -1 \\ & & & -1 & 16 & -30 & 16 \\ & & & & -1 & 16 & -30 \end{pmatrix} \begin{bmatrix} T_1 \\ T_2 \\ T_3 \\ T_4 \\ T_5 \\ \vdots \\ T_{N-1} \\ T_N \end{bmatrix} = \begin{bmatrix} q_1/\lambda - (16T_0 - T_{-1}) \\ q_2/\lambda - (-T_{-0}) \\ q_3/\lambda \\ q_4/\lambda \\ q_5/\lambda \\ \cdots \\ q_{N-1}/\lambda - (-T_{N+1}) \\ q_N/\lambda - (16T_{N+1} - T_{N+2}) \end{bmatrix} \tag{24}$$

$$\Rightarrow \mathbf{AT} = \mathbf{B}$$

So, the resulting matrix $\mathbf{A}$ is pentadiagonal (5 diagonals)

$$\mathbf{A} = \begin{pmatrix} \times & \times & \times & & & & \\ \times & \times & \times & \times & & & \\ \times & \times & \ddots & \ddots & \ddots & & \\ & \times & \ddots & \ddots & \times & \times & \\ & & \ddots & \times & \times & \times & \times \\ & & & \times & \times & \times & \times \\ & & & & \times & \times & \times \end{pmatrix} \tag{25}$$

4.2.2. *2D.* Equation 20 can be substituted into equation 7, to give

$$-k\left(\frac{-T_{i+2,j} + 16T_{i+1,j} - 30T_{i,j} + 16T_{i-1,j} - T_{i-2,j}}{\Delta 12x^2} + \frac{-T_{i,j+2} + 16T_{i,j+1} - 30T_{i,j} + 16T_{i,j-1} - T_{i,j-2}}{\Delta 12y^2}\right) + \mathcal{O}(h^4) = q_{i,j} \tag{26}$$

where i and j are the positions of cell center in x and y directions. Implemententing, $\Delta x = \Delta y = h$ from assumptions and $-k/12h^2 = \lambda$

$$-T_{i+2,j} + 16T_{i+1,j} + 16T_{i-1,j} - T_{i-2,j} - T_{i,j+2} + 16T_{i,j+1} + 16T_{i,j-1} - T_{i,j-2} - 60T_{i,j} + \mathcal{O}(h^6) = q_{i,j}/\lambda \tag{27}$$

Using k indexing for blocks in same row,

$$-T_{k+2,k} + 16T_{k+1,k} + 16T_{k-1,k} - T_{k-2,k} - T_{k+2N_x,k} + 16T_{k+N_x,k} + 16T_{k-N_x,k} - T_{k-2N_x,k} - 60T_{k,k} + \mathcal{O}(h^6) = q_k/\lambda \tag{28}$$

Neglecting the truncation error $\mathcal{O}(h^6)$ and writing in matrix form for 3×3 uniform grid shown in figure 2 after implementing the penalty approach for the boundary terms, we get

$$
\begin{pmatrix}
-30 & 16 & -1 & -1 & & & -1 & & \\
16 & -30 & 16 & & -1 & & & -1 & \\
-1 & 16 & -30 & & & -1 & & & -1 \\
-1 & & & -30 & 16 & -1 & -1 & & \\
& -1 & & 16 & -30 & 16 & & -1 & \\
& & -1 & -1 & 16 & -30 & & & -1 \\
-1 & & & -1 & & & -30 & 16 & -1 \\
& -1 & & & -1 & & 16 & -30 & 16 \\
& & -1 & & & -1 & -1 & 16 & -30
\end{pmatrix}
\begin{bmatrix} T_1 \\ T_2 \\ T_3 \\ T_4 \\ T_5 \\ \vdots \\ T_8 \\ T_9 \end{bmatrix}
=
\begin{bmatrix}
q_1/\lambda - (16T_{0,1} - T_{-1,1} + 16T_{1,0} - T_{1,-1}) \\
q_2/\lambda - (-T_{0,1} - T_{4,1} + 16T_{2,0} - T_{2,-1}) \\
q_3/\lambda - (16T_{4,1} - T_{5,1} + 16T_{3,0} - T_{3,-1}) \\
q_4/\lambda - (-T_{-1,2} + 16T_{0,2} - T_{1,0} - T_{1,4}) \\
q_5/\lambda - (-T_{0,2} - T_{4,2} - T_{2,0} - T_{2,4}) \\
\vdots \\
q_8/\lambda - (16T_{2,4} - T_{2,5} - T_{0,3} - T_{4,3}) \\
q_9/\lambda - (16T_{3,4} - T_{3,5} + 16T_{4,3} - T_{5,3})
\end{bmatrix}
$$

$$\Rightarrow \mathbf{AT} = \mathbf{B} \tag{29}$$

For a general $N \times N$ grid ($N > 3$), the matrix equation takes the form

$$
\begin{pmatrix}
-30 & 16 & -1 & \Leftarrow (N-2) \Rightarrow & -1 & \Leftarrow N \Rightarrow & -1 & & \\
16 & -30 & 16 & -1 & \Leftarrow (N-2) \Rightarrow & -1 & & \ddots & \\
-1 & 16 & -30 & 16 & -1 & & \ddots & & -1 \\
& -1 & 16 & -30 & \ddots & -1 & \Leftarrow (N-2) \Rightarrow & -1 & \\
-1 & \Leftarrow (N-2) \Rightarrow & -1 & \ddots & \ddots & 16 & -1 & \Leftarrow (N-2) \Rightarrow & -1 \\
& \ddots & & -1 & 16 & -30 & 16 & -1 & \\
-1 & \Leftarrow N \Rightarrow & -1 & \Leftarrow (N-2) \Rightarrow & -1 & 16 & -30 & 16 & -1 \\
& \ddots & & -1 & \Leftarrow (N-2) \Rightarrow & -1 & 16 & -30 & 16 \\
& & -1 & \Leftarrow N \Rightarrow & -1 & \Leftarrow (N-2) \Rightarrow & -1 & 16 & -30
\end{pmatrix}
$$

$$
\times
\begin{bmatrix} T_1 \\ T_2 \\ T_3 \\ T_4 \\ T_5 \\ \vdots \\ T_8 \\ T_9 \end{bmatrix}
=
\begin{bmatrix}
q_1/\lambda - (16T_{0,1} - T_{-1,1} + 16T_{1,0} - T_{1,-1}) \\
q_2/\lambda - (-T_{0,1} - T_{4,1} + 16T_{2,0} - T_{2,-1}) \\
q_3/\lambda - (16T_{4,1} - T_{5,1} + 16T_{3,0} - T_{3,-1}) \\
\vdots \\
q_N/\lambda - (16T_{N+1,1} - T_{N+2,1} + 16T_{N,0} - T_{N,-1}) \\
q_{N+1}/\lambda - (16T_{0,2} - T_{-1,2} - T_{1,0}) \\
\vdots \\
q_{N*N-1}/\lambda - (16T_{N-1,N+1} - T_{N-1,N+2} - T_{N+1,N}) \\
q_{N*N}/\lambda - (16T_{N,N+1} - T_{N,N+2} + 16T_{N+1,N} - T_{N+2,N})
\end{bmatrix}
\tag{30}
$$

$$\Rightarrow \mathbf{AT} = \mathbf{B} \tag{31}$$

where "$\Leftarrow N \Rightarrow$" and "$\Leftarrow (N-2) \Rightarrow$" are the distances between the elements. So, the resulting matrix $\mathbf{A}$ is banded with 9 diagonals: principal diagonal, offdiagonals at distance 1, at distance N, and at distance $2N$ from principal diagonal as shown below:

$$\mathbf{A} = \begin{pmatrix}
\times & \times & -1 & \Leftarrow (N-2) \Rightarrow & \times & \Leftarrow N \Rightarrow & \times & & \\
\times & \times & \times & \times & \Leftarrow (N-2) \Rightarrow & \times & & \ddots & \\
\times & \times & \times & \times & \times & & \ddots & & \times \\
 & \times & \times & \times & \ddots & \times & \Leftarrow (N-2) \Rightarrow & \times & \\
\times & \Leftarrow (N-2) \Rightarrow & \times & \ddots & \ddots & \times & \times & \Leftarrow (N-2) \Rightarrow & \times \\
 & \ddots & & \times & \times & \times & \times & \times & \\
\times & \Leftarrow N \Rightarrow & \times & \Leftarrow (N-2) \Rightarrow & \times & \times & \times & \times & \times \\
 & \ddots & & \times & \Leftarrow (N-2) \Rightarrow & \times & \times & \times & \times \\
 & & \times & \Leftarrow N \Rightarrow & \times & \Leftarrow (N-2) \Rightarrow & \times & \times & \times
\end{pmatrix} \tag{32}$$

5. Pseudocodes [1]

5.1. **Jacobi.** The algorithm is given below:

```
Input: A= A_{i,j}, B= B_j, TO= TO_j=T^(0), tolerance TOL, max
number of iterations N_iter
STEP 1 Set k = 1
STEP 2 While (k ≤ N) do Steps 3-6
       STEP 3 For i = 1, 2, ..., n
```

$$x_i = \frac{1}{a_{i,i}}\Big[\sum_{j=1, j\neq i}^{n} (-A_{i,j} TO_j) + B_i\Big]$$

```
       STEP 4 If ||T − TO|| < TOL or ||T^(k) − T^(k−1)|| / ||T^(k)|| < TOL, then
                                OUTPUT(T_1, T_2, T_3, T_4, ..., T_n); STOP
       STEP 5 Set k = k + 1
       STEP 6 For i = 1, 2, ..., n
              Set TO_i = T_i
STEP 7 OUTPUT (T_1, T_2, T_3, T_4, ..., T_n); STOP
```

5.2. **Gauss-Seidel.** The algorithm is given below:

```
Input: A= A_{i,j}, B= B_j, TO= TO_j=T^(0), tolerance TOL, max
number of iterations N_iter
STEP 1 Set k = 1
STEP 2 While (k ≤ N) do Steps 3-6
```

```
STEP 3 For i = 1, 2, ..., n
```

$$x_i = \frac{1}{a_{i,i}}\left[-\sum_{j=1}^{i-1}(A_{i,j}T_j) - \sum_{j=i+1}^{n}(A_{i,j}TO_j) + B_i\right]$$

STEP 4 If $||\mathbf{T} - \mathbf{TO}|| < TOL$ or $\frac{||\mathbf{T}^{(k)} - \mathbf{T}^{(k-1)}||}{||\mathbf{T}^{(k)}||} < TOL$, then
OUTPUT$(T_1, T_2, T_3, T_4, ..., T_n)$; STOP

STEP 5 Set $k = k + 1$

STEP 6 For $i = 1, 2, ..., n$
Set $\mathbf{TO}_i = T_i$

STEP 7 OUTPUT $(T_1, T_2, T_3, T_4, ..., T_n)$; STOP

6. Memory requirement

Let's have a look at the memory requirement on case-by-case in the tables. Also, neglecting the elements at the top left and bottom right corners of the matrix **A**.

Table 1. Memory requirement for 1D, second order central difference

Variable	Dimension	Memory	Remarks
T	N	$8 \times N$	8-byte double precision, column vector
B	N	$8 \times N$	8-byte double precision, column vector
A	$N \times N$	$8 \times 3 \times N$	8-byte double precision, square tridiagonal matrix
N	1	4×1	4-byte single precision, scalar
L	1	8×1	8-byte double precision, scalar
Δx	1	8×1	8-byte double precision, scalar
k	1	8×1	8-byte double precision, scalar
Total		$40N + 28$	

Table 2. Memory requirement for 2D, second order central difference

Variable	Dimension	Memory	Remarks
$\mathbf{T}$	N^2	$8 \times N^2$	8-byte double precision, column vector
$\mathbf{B}$	N^2	$8 \times N^2$	8-byte double precision, column vector
$\mathbf{A}$	$N^2 \times N^2$	$8 \times 5 \times N^2$	8-byte double precision, banded, 5 diagonals
N	1	4×1	4-byte single precision, scalar
$L \equiv H$	1	8×1	8-byte double precision, scalar
$\Delta x = \Delta y = h$	1	8×1	8-byte double precision, scalar
k	1	8×1	8-byte double precision, scalar
Total		$56N^2 + 28$	

Table 3. Memory requirement for 1D, fourth order central difference

Variable	Dimension	Memory	Remarks
$\mathbf{T}$	N	$8 \times N$	8-byte double precision, column vector
$\mathbf{B}$	N	$8 \times N$	8-byte double precision, column vector
$\mathbf{A}$	$N \times N$	$8 \times 5 \times N$	8-byte double precision, pentadiagonal square matrix
N	1	4×1	4-byte single precision, scalar
L	1	8×1	8-byte double precision, scalar
Δx	1	8×1	8-byte double precision, scalar
k	1	8×1	8-byte double precision, scalar
Total		$56N + 28$	

Table 4. Memory requirement for 2D, fourth order central difference

Variable	Dimension	Memory	Remarks
$\mathbf{T}$	N^2	$8 \times N^2$	8-byte double precision, column vector
$\mathbf{B}$	N^2	$8 \times N^2$	8-byte double precision, column vector
$\mathbf{A}$	$N^2 \times N^2$	$8 \times 9 \times N^2$	8-byte double precision, banded, 9 diagonals
N	1	4×1	4-byte single precision, scalar
L	1	8×1	8-byte double precision, scalar
Δx	1	8×1	8-byte double precision, scalar
k	1	8×1	8-byte double precision, scalar
Total		$90N^2 + 28$	

References

[1] Kendall E Atkinson and Weimin Han. *Elementary numerical analysis*. Wiley New York, 1985.

[2] J Tinsley Oden. *An introduction to mathematical modeling: a course in mechanics*. John Wiley & Sons, 2011.

[3] Alfio Quarteroni, Riccardo Sacco, and Fausto Saleri. *Numerical mathematics*, volume 37. Springer Science & Business Media, 2010.

Email address: `mashadab@utexas.edu`