Modeling Document

- **Governing Equations**

$$\begin{aligned} -k\nabla^2 T(x,y) &= q(x,y) && \text{in } \Omega \\ T &= T_b && \text{on } \partial\Omega \end{aligned} \tag{1}$$

  where,

$$\begin{aligned} \Omega &\ \text{ is a 2D bounded domain} \\ \partial\Omega &\ \text{ is the boundary of the domain} \\ T &\ \text{ is the material's temperature field} \\ q &\ \text{ is the heat source term} \\ k &\ \text{ is the thermal conductivity} \end{aligned}$$

  We have the Dirichlet boundary conditions on the boundary.

- **Assumptions**
  - The thermal conductivity is assumed to be constant.
  - We assume a square domain $\Omega = \{\ (x,y) : x \in [0,L], y \in [0,L]\ \}$ for the 2D case. For 1D, it will obviously be a line $\Omega = \{\ (x,y) : x \in [0,L]\ \}$
  - Dirichlet boundary condition is assumed at the boundaries
  - For the fourth order scheme, we assume that the values at the points adjacent to the boundary points are known. This is to reduce the cumbersome effort required to come up with different schemes at the boundary.

- **Nomenclature for discretization** Our numerical methods are all node based (as will be reiterated later).
  - 1D We have $(N+1)$ points $\{x_0, x_1, x_2, \dots x_N\}$ in the x-direction with $x_i = i\Delta x$, where $\Delta x = L/N$
  - 2D We have $(N+1)$ points $\{x_0, x_1, x_2, \dots x_N\}$ in the x-direction with $x_i = i\Delta x$, where $\Delta x = L/N$ and $(J+1)$ points $\{y_0, y_1, y_2, \dots y_J\}$ in the y-direction with $y_j = j\Delta y$, where $\Delta j = L/J$. Hence, we have a $(N+1) \times (J+1)$ grid.
  - i is always associated with the indexing in x-direction and j is always associated with the indexing in y-direction.
  - $T(x_i, y_j)$ is given the shorthand notation $T(i,j)$ and $q(x_i, y_j)$ is given the shorthand notation $q(i,j)$

- **Numerical Method**

  Our numerical methods are all node based (as will be reiterated later).

  - $2^{\textbf{nd}}$ **order finite difference approximation**

    Definition

$$\frac{\partial^2 T}{\partial x^2} \approx \frac{T(x+\Delta x) - 2T(x) + T(x-\Delta x)}{\Delta x^2} + \mathcal{O}(\Delta x^2)$$

    Discretized heat equation

    1. 1D

$$\begin{cases} -k\left(\frac{T(i+1)-2T(i)+T(i-1)}{\Delta x^2}\right) + \mathcal{O}(\Delta x^2) = q(i), \;\; i \in \{1,2,3,\ldots,N-1\} \\ T(0) = T_b \;\text{ and }\; T(N) = T_b \end{cases} \tag{2}$$

    2. 2D

$$\begin{cases} -k\left(\frac{T(i+1,j)-2T(i,j)+T(i-1,j)}{\Delta x^2}\right) - k\left(\frac{T(i,j+1)-2T(i,j)+T(i,j-1)}{\Delta y^2}\right) \\ +\mathcal{O}(\Delta x^2) + \mathcal{O}(\Delta y^2) = q(i,j), \;\; i \in \{1,2,3,\ldots,N-1\} \;\; j \in \{1,2,3,\ldots,J-1\} \\ \\ T(0,j) = T_b \;\text{ for }\; j \in \{0,1,2,\ldots,J\} \\ T(i,0) = T_b \;\text{ for }\; i \in \{0,1,2,\ldots,N\} \\ T(N,j) = T_b \;\text{ for }\; j \in \{0,1,2,\ldots,J\} \\ T(i,J) = T_b \;\text{ for }\; i \in \{0,1,2,\ldots,N\} \end{cases} \tag{3}$$

  - $4^{\textbf{th}}$ **order finite difference approximation**

    Definition

$$\frac{\partial^2 T}{\partial x^2} \approx \frac{-T(x-2\Delta x) + 16T(x-\Delta x) - 30T(x) + 16T(x+\Delta x) - T(x+2\Delta x)}{12\Delta x^2} + \mathcal{O}(\Delta x^4)$$

    Discretized heat equation

    Near the boundaries, we have to use the second order approximation because a fourth order approximation cannot be defined.

    1. 1D

$$\begin{cases} -k\left(\frac{T(i+1)-2T(i)+T(i-1)}{\Delta x^2}\right) + \mathcal{O}(\Delta x^2) = q(i), \;\; i \in \{1, N-1\} \\ -k\left(\frac{-T(i-2)+16T(i-1)-30T(i)+16T(i+1)-T(i+2)}{12\Delta x^2}\right) + \mathcal{O}(\Delta x^4) = q(i), \;\; i \in \{2,3,\ldots,N-2\} \\ T(0) = T_b \;\text{ and }\; T(N) = T_b \end{cases} \tag{4}$$

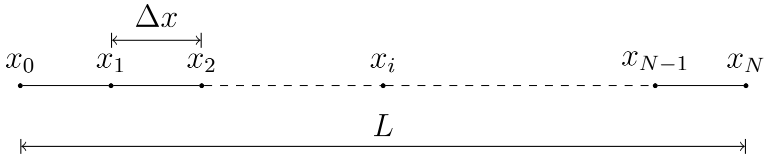
2. 2D

$$
\begin{cases}
-k\left(\frac{T(i-1,j)-2T(i,j)+T(i+1,j)}{\Delta x^2}\right)+\mathcal{O}(\Delta x^2) \\
-k\left(\frac{T(i,j-1)-2T(i,j)+T(i,j+1)}{\Delta y^2}\right)+\mathcal{O}(\Delta y^2)=q(i,j), \\
\ i\ \in\{1,N-1\}, j\ \ \in\{1,J-1\} \\
\\
-k\left(\frac{T(i-1,j)-2T(i,j)+T(i+1,j)}{\Delta x^2}\right)+\mathcal{O}(\Delta x^2) \\
-k\left(\frac{-T(i,j-2)+16T(i,j-1)-30T(i,j)+16T(i,j+1)-T(i,j+2)}{12\Delta y^2}\right)+\mathcal{O}(\Delta y^4)=q(i,j), \\
\ i\ \in\{1,N-1\}, j\ \ \in\{2,\dots,J-2\} \\
\\
-k\left(\frac{-T(i-2,j)+16T(i-1,j)-30T(i,j)+16T(i+1,j)-T(i+2,j)}{12\Delta x^2}\right)+\mathcal{O}(\Delta x^4) \\
-k\left(\frac{T(i,j-1)-2T(i,j)+T(i,j+1)}{\Delta y^2}\right)+\mathcal{O}(\Delta y^2)=q(i,j), \\
\ i\ \in\{2,\dots,N-2\}, j\ \ \in\{1,J-1\} \\
\\
-k\left(\frac{-T(i-2,j)+16T(i-1,j)-30T(i,j)+16T(i+1,j)-T(i+2,j)}{12\Delta x^2}\right)+\mathcal{O}(\Delta x^4) \\
-k\left(\frac{-T(i,j-2)+16T(i,j-1)-30T(i,j)+16T(i,j+1)-T(i,j+2)}{12\Delta y^2}\right)+\mathcal{O}(\Delta y^4)=q(i,j), \\
\ i\ \in\{2,\dots,N-2\}, j\ \ \in\{2,\dots,J-2\} \\
\\
T(0,j)=T_b\ \text{ for }\ j\in\{0,1,2,\dots,J\} \\
T(i,0)=T_b\ \text{ for }\ i\in\{0,1,2,\dots,N\} \\
T(N,j)=T_b\ \text{ for }\ j\in\{0,1,2,\dots,J\} \\
T(i,J)=T_b\ \text{ for }\ i\in\{0,1,2,\dots,N\}
\end{cases}
\tag{5}
$$

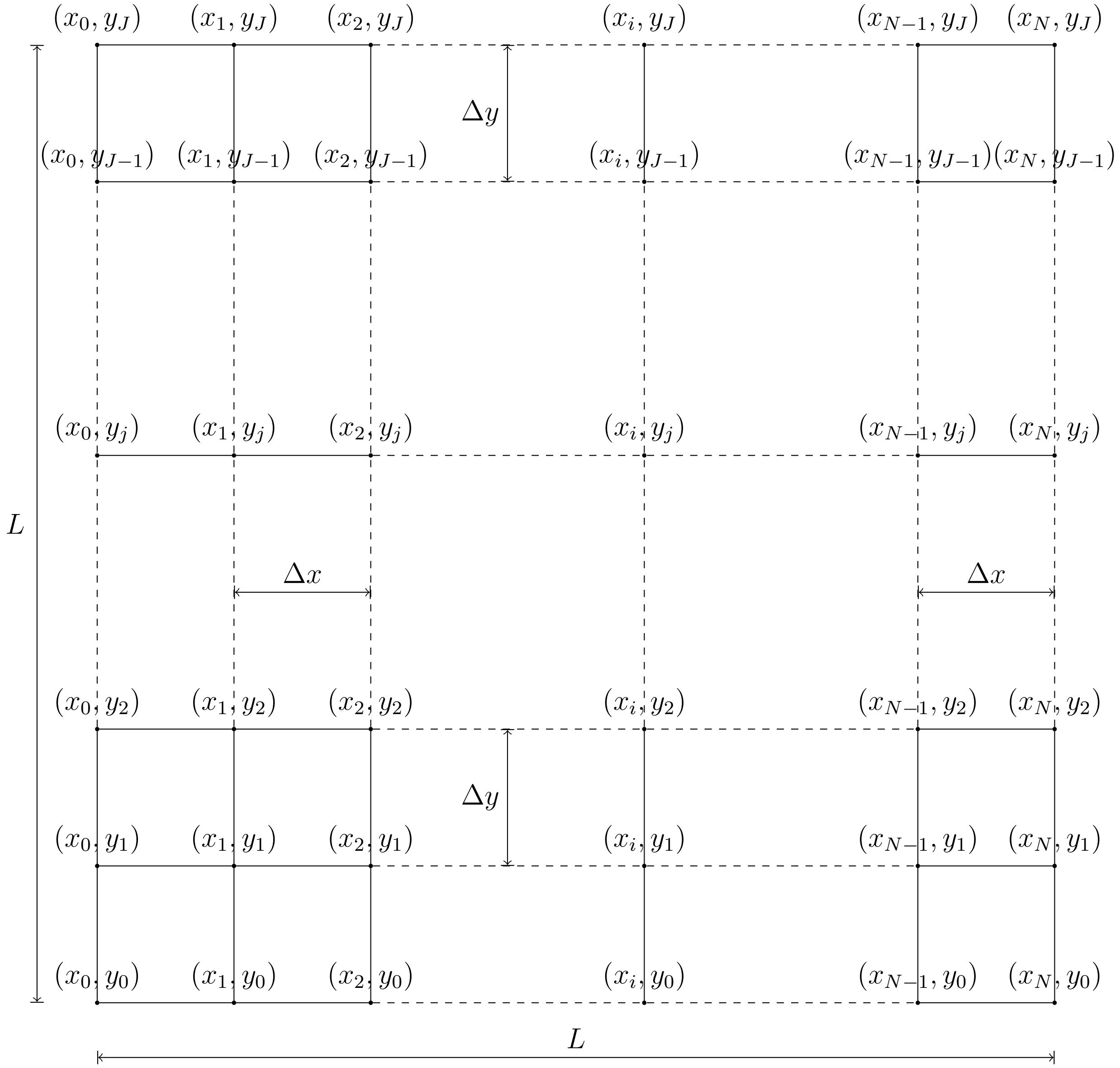
- **Mesh diagrams**

  Our schemes are node based.

  1. 1D



  2. 2D



- **Linear system of Equations**

  - **2^{nd} order finite difference approximation**

    1. 1D

       We first define the following vectors

$$\mathbf{q} = \left[\frac{k}{\Delta x^2}T_b, q(1), \ldots, q(N-1), \frac{k}{\Delta x^2}T_b\right]^T$$
$$\mathbf{T} = [T(0), \ldots, T(N)]^T$$

We now define a $(N+1)\times(N+1)$ tridiagonal matrix $\mathbf{A}$ such that,

$$\mathbf{A} = \frac{k}{\Delta x^2}\begin{bmatrix} 1 & & & & & & \\ -1 & 2 & -1 & & & & \\ & -1 & 2 & -1 & & & \\ & & \ddots & \ddots & \ddots & & \\ & & & -1 & 2 & -1 & \\ & & & & -1 & 2 & -1 \\ & & & & & & 1 \end{bmatrix}$$

Now (2) can be written as,

$$\mathbf{AT} = \mathbf{q}$$

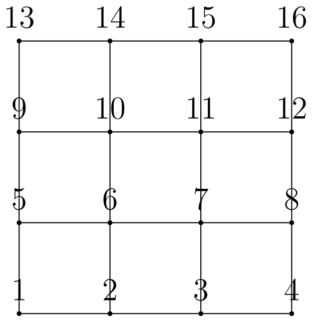
The first and last rows are different because they account for boundary conditions. The sparsity pattern of A can be given by,

$$\mathbf{A} = \begin{bmatrix} \times & & & & & & \\ \times & \times & \times & & & & \\ & \times & \times & \times & & & \\ & & \times & \times & \times & & \\ & & & \times & \times & \times & \\ & & & & \times & \times & \times \\ & & & & & & \times \end{bmatrix}$$

`Number of non-zero elements on an interior row of the matrix =` 3

2. 2D

As an illustration, the elements will be numbered in the following fashion



So we can imagine how the matrix is going to look: the terms of second derivative in x-direction will be adjacent to each other, but they terms of the second derivative in y-direction will be offset by N points. This will become clear in the visual representation below. We first define the following $(N+1)(J+1)\times 1$ vectors

$$\mathbf{q}(i+j(N+1)) = \begin{cases} q(i,j) & \text{if } i \in \{1,2,\ldots,N-1\},\ j \in \{1,2,\ldots,J-1\} \\ \frac{k}{\Delta x^2}T_b & \text{otherwise i.e. at boundaries} \end{cases}$$

$$\mathbf{T} = [T(0), T(1), \ldots, T((N+1)(J+1)))]^T$$

We now define $(N+1)(J+1) \times (N+1)(J+1)$ matrices $\mathbf{A_x}$ (interior x-direction derivatives), $\mathbf{A_y}$ (interior y-direction derivatives) and $\mathbf{A_b}$ (boundary elements) such that, (refer to Appendix A.1 for the individual sparsity patterns.)

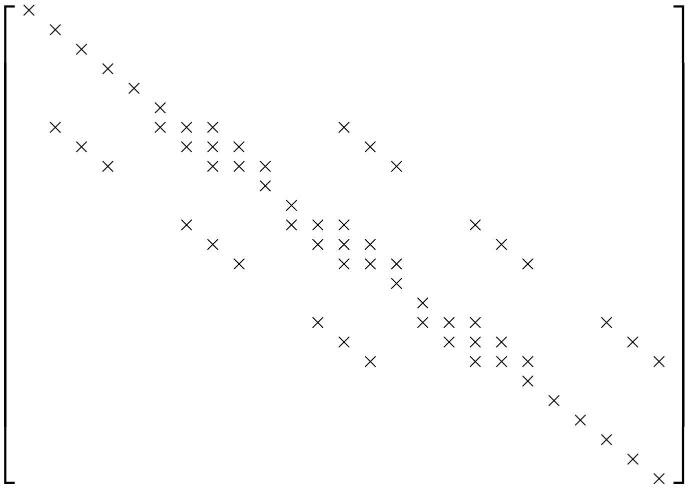
$$\mathbf{A_x} = \begin{cases} \mathbf{A_x}(i,i) = \begin{cases} 2\frac{k}{\Delta x^2} & \text{if } i \text{ corresponds to an interior point} \\ 0 & \text{otherwise} \end{cases} \\ \mathbf{A_x}(i,i-1) = \begin{cases} -\frac{k}{\Delta x^2} & \text{if } i \text{ corresponds to an interior point} \\ 0 & \text{otherwise} \end{cases} \\ \mathbf{A_x}(i,i+1) = \begin{cases} -\frac{k}{\Delta x^2} & \text{if } i \text{ corresponds to an interior point} \\ 0 & \text{otherwise} \end{cases} \end{cases}$$

$$\mathbf{A_y} = \begin{cases} \mathbf{A_y}(j,j) = \begin{cases} 2\frac{k}{\Delta y^2} & \text{if } j \text{ corresponds to an interior point} \\ 0 & \text{otherwise} \end{cases} \\ \mathbf{A_x}(j,j-N) = \begin{cases} -\frac{k}{\Delta y^2} & \text{if } j \text{ corresponds to an interior point} \\ 0 & \text{otherwise} \end{cases} \\ \mathbf{A_x}(j,j+N) = \begin{cases} -\frac{k}{\Delta y^2} & \text{if } j \text{ corresponds to an interior point} \\ 0 & \text{otherwise} \end{cases} \end{cases}$$

$$\mathbf{A_b} = \begin{cases} \mathbf{A_b}(i,i) = \begin{cases} \frac{k}{\Delta x^2} & \text{if } i \text{ corresponds to an interior point} \\ 0 & \text{otherwise} \end{cases} \end{cases}$$

Now, the net matrix is $\mathbf{A} = \mathbf{A_x} + \mathbf{A_y} + \mathbf{A_b}$. The sparsity pattern of $\mathbf{A}$ is given by, (illustration for a 5 × 5 grid)

$\mathbf{A} =$



`Number of non-zero elements on an interior row of the matrix = 5.`

Now (3) can be written as,

$$\mathbf{AT} = \mathbf{q}$$

– 4^{th} **order finite difference approximation**

1. 1D

   We first define the following vectors

$$\mathbf{q} = \left[\frac{k}{12\Delta x^2}T_b, \frac{k}{12\Delta x^2}T_{true}(1), q(3)\ldots, q(N-2), \frac{k}{12\Delta x^2}T_{true}(N-1), \frac{k}{12\Delta x^2}T_b\right]^T$$

$$\mathbf{T} = [T(0), \ldots, T(N)]^T$$

   Since our purpose is code verification, we assume that $T(1)$ and $T(N-1)$ are known to us from the analytical solution, just to avoid using a different scheme there.

   We now define a $(N+1)\times(N+1)$ pentadiagonal matrix $\mathbf{A}$ such that (blank elements are 0),

$$\mathbf{A} = \frac{k}{12\Delta x^2}\begin{bmatrix} 1 & & & & & & & & & \\ & 1 & & & & & & & & \\ 1 & -16 & 30 & -16 & 1 & & & & & \\ & 1 & -16 & 30 & -16 & 1 & & & & \\ & & 1 & -16 & 30 & -16 & 1 & & & \\ & & & \ddots & \ddots & \ddots & \ddots & \ddots & & \\ & & & & 1 & -16 & 30 & -16 & 1 & \\ & & & & & 1 & -16 & 30 & -16 & 1 \\ & & & & & & & & 1 & \\ & & & & & & & & & 1 \end{bmatrix}$$

   Now (4) can be written as,

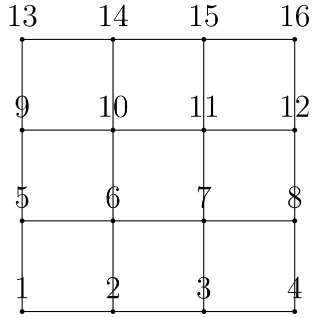
$$\mathbf{AT} = \mathbf{q}$$

   Note that the first and last row in $\mathbf{A}$ looks different because we accounted for the boundary conditions. Similarly, the second and second-last rows look different because we use a second order approximation close to the boundary. The sparsity pattern of the matrix is given by,

$$\mathbf{A} = \begin{bmatrix} \times & & & & & & & & \\ & \times & & & & & & & \\ \times & \times & \times & \times & \times & & & & \\ & \times & \times & \times & \times & \times & & & \\ & & \times & \times & \times & \times & \times & & \\ & & & \times & \times & \times & \times & \times & \\ & & & & \times & \times & \times & \times & \times \\ & & & & & & & \times & \\ & & & & & & & & \times \end{bmatrix}$$

```
Number of non-zero elements on an interior row of the matrix = 5.
```

2. 2D As an illustration, the elements will be numbered in the following fashion



So we can imagine how the matrix is going to look: the terms of second derivative in x-direction will be adjacent to each other, but they terms of the second derivative in y-direction will be offset by N points. This will become clear in the visual representation below. We first define the following $(N+1)(J+1) \times 1$ vectors

$$\mathbf{q}(i + j(N+1)) = \begin{cases} q(i,j) \ \text{ if } \ i \in \{1, 2, \ldots, N-1\}, \ j \in \{1, 2, \ldots, J-1\} \\ \frac{k}{12\Delta x^2} T_b \quad \text{at boundaries} \\ \frac{k}{12\Delta x^2} T_{true}(i,j) \quad \text{i = 1 or i = N-1 or j = 1 or j = J-1} \\ \text{Assumption that these values are known from analytical solution} \end{cases}$$

$$\mathbf{T} = [T(0), T(1), \ldots, T((N+1)(J+1))]^T$$

We now define $(N+1)(J+1) \times (N+1)(J+1)$ matrices $\mathbf{A_x}$ (interior x-direction derivatives), $\mathbf{A_y}$ (interior y-direction derivatives) and $\mathbf{A_b}$ (boundary and close to boundary elements) such that, (refer to Appendix A.2 for the individual sparsity patterns.)

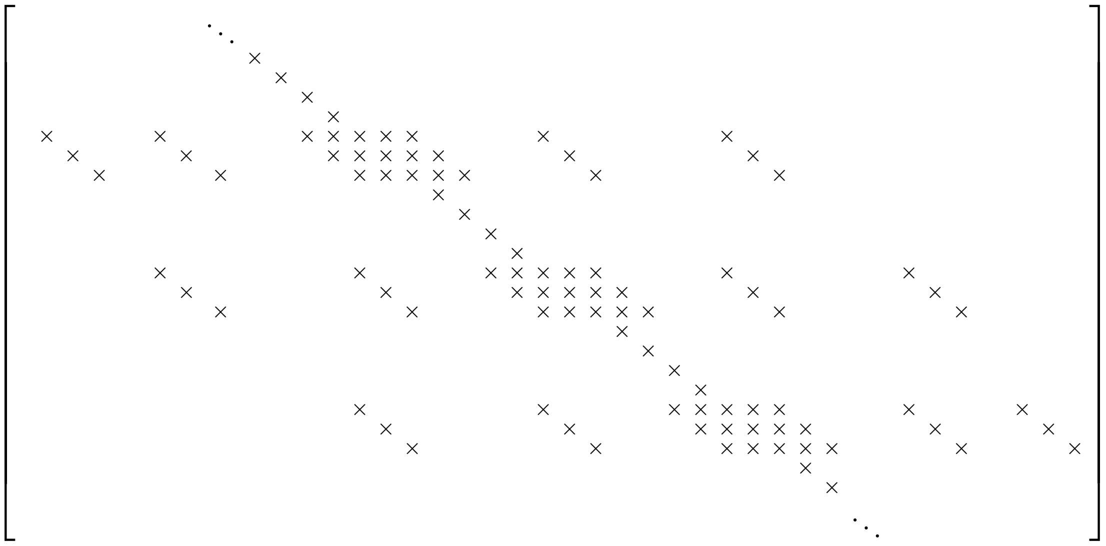
$$\mathbf{A_x} = \begin{cases} \mathbf{A_x}(i,i) = \begin{cases} 30\frac{k}{12\Delta x^2} & \text{if } i \text{ corresponds to an interior point} \\ 0 & \text{otherwise} \end{cases} \\ \mathbf{A_x}(i,i-1) = \begin{cases} -16\frac{k}{12\Delta x^2} & \text{if } i \text{ corresponds to an interior point} \\ 0 & \text{otherwise} \end{cases} \\ \mathbf{A_x}(i,i+1) = \begin{cases} -16\frac{k}{12\Delta x^2} & \text{if } i \text{ corresponds to an interior point} \\ 0 & \text{otherwise} \end{cases} \\ \mathbf{A_x}(i,i+2) = \begin{cases} \frac{k}{12\Delta x^2} & \text{if } i \text{ corresponds to an interior point} \\ 0 & \text{otherwise} \end{cases} \\ \mathbf{A_x}(i,i-2) = \begin{cases} \frac{k}{12\Delta x^2} & \text{if } i \text{ corresponds to an interior point} \\ 0 & \text{otherwise} \end{cases} \end{cases}$$

$$\mathbf{A_y} = \begin{cases} \mathbf{A_y}(j,j) = \begin{cases} 30\frac{k}{12\Delta y^2} & \text{if } j \text{ corresponds to an interior point} \\ 0 & \text{otherwise} \end{cases} \\ \mathbf{A_y}(j,j-N) = \begin{cases} -16\frac{k}{12\Delta y^2} & \text{if } j \text{ corresponds to an interior point} \\ 0 & \text{otherwise} \end{cases} \\ \mathbf{A_y}(j,j+N) = \begin{cases} -16\frac{k}{12\Delta y^2} & \text{if } j \text{ corresponds to an interior point} \\ 0 & \text{otherwise} \end{cases} \\ \mathbf{A_y}(j,j+2N) = \begin{cases} \frac{k}{12\Delta y^2} & \text{if } j \text{ corresponds to an interior point} \\ 0 & \text{otherwise} \end{cases} \\ \mathbf{A_y}(j,j-2N) = \begin{cases} \frac{k}{12\Delta y^2} & \text{if } j \text{ corresponds to an interior point} \\ 0 & \text{otherwise} \end{cases} \end{cases}$$

$$\mathbf{A_b} = \begin{cases} \mathbf{A_b}(j,j) = \begin{cases} \frac{k}{12\Delta x^2} & \text{if } j \text{ is a boundary point or an adjacent to boundary point} \\ 0 & \text{otherwise} \end{cases} \end{cases}$$

Now, the net matrix is $\mathbf{A} = \mathbf{A_x} + \mathbf{A_y} + \mathbf{A_b}$. The sparsity pattern of $\mathbf{A}$ is given by, (illustration for a 7 × 7 grid, which means the interior matrix is 3 × 3)

$\mathbf{A} =$



Number of non-zero elements on an interior row of the matrix = 9.

Now (5) can be written as,

$$\mathbf{AT} = \mathbf{q}$$

- **Iterative solvers**

1. **Jacobi**

```
subroutine jacobi (A,q, TOL, max_iter)
!!! Find T = inv(A)*q using Jacobi iteration

K = size(q)
iters = 0 !!! Number of iterations
error = 0 !!! Compare to tolerance
T(1:K) = 0 !!! Initialize entire T array to zero

do while (iters <= max_iters)

    T_old(:) = T(:)

    do i = 1,...,K
        T(i) =  1/A(i,i) * (q(i)-$\Sigma_{j\neq i}^{K}$A(i,j)T_old(j))
    end do

    error = $\frac{\|T\_old - T\|}{\|T\|}$

    iters = iters + 1

    if (error <= TOL)
        break
    end if
end do

return T
end subroutine
```

2. **Gauss-Seidel**

```
subroutine gauss_seidel (A,q, TOL, max_iter)
!!! Find T = inv(A)*q using Gauss Seidel iteration

K = size(q)
iters = 0 !!! Number of iterations
error = 0 !!! Compare to tolerance
T(1:K) = 0 !!! Initialize entire T array to zero

do while (iters <= max_iters)

    T_old(:) = T(:)

    do i = 1,...,K
        T(i) =  1/A(i,i) * (q(i)-$\Sigma_{j\neq i}^{K}$A(i,j)T(j))
    end do

    error = $\frac{\|T\_\mathtt{old}-T\|}{\|T\|}$
    iters = iters + 1
    if (error <= TOL)
        break
    end if



end do

return T
end subroutine
```

- **Estimate of memory requirements**
  - Second order approximation

Variable	Dimension	Memory	Comments
T	N+1	8×(N+1)	
q	N+1	8×(N+1)	
A	(N+1)×(N+1)	8×3	
N	1	4	
L	1	8	
Δx	1	8	
k	1	8	
	Total	4(4N+17)	

- 2D

Variable	Dimension	Memory	Comments
T	(N+1)(J+1)	8×(N+1)(J+1)	
q	(N+1)(J+1)	8×(N+1)(J+1)	
A	(N+1)(J+1)×(N+1)(J+1)	8×5	
N	1	4	
J	1	4	
L	1	8	
Δx	1	8	
Δy	1	8	
k	1	8	
	Total	16(N+1)(J+1)+80	

- Fourth order approximation
  - 1D

Variable	Dimension	Memory	Comments
T	N+1	8×(N+1)	
q	N+1	8×(N+1)	
A	(N+1)×(N+1)	8×5	
N	1	4	
L	1	8	
Δx	1	8	
k	1	8	
	Total	4(4N+21)	

  - 2D

Variable	Dimension	Memory	Comments
T	(N+1)(J+1)	8×(N+1)(J+1)	
q	(N+1)(J+1)	8×(N+1)(J+1)	
A	(N+1)(J+1)×(N+1)(J+1)	8×9	
N	1	4	
J	1	4	
L	1	8	
Δx	1	8	
Δy	1	8	
k	1	8	
	Total	16(N+1)(J+1)+120	

A Sparsity patterns

A.1 2D Second order approximation

The sparsity pattern of $\mathbf{A_x}$ is given by, (illustration for a 5×5 grid)

$$\mathbf{A_x} = \left[\begin{array}{ccccccccccccccc}
\\
\times & \times & \times \\
& \times & \times & \times \\
& & \times & \times & \times \\
\\
& & & & & \times & \times & \times \\
& & & & & & \times & \times & \times \\
& & & & & & & \times & \times & \times \\
\\
& & & & & & & & & & \times & \times & \times \\
& & & & & & & & & & & \times & \times & \times \\
& & & & & & & & & & & & \times & \times & \times \\
\\
\end{array}\right]$$

The sparsity pattern of $\mathbf{A_y}$ is given by, (illustration for a 5×5 grid)

$$\mathbf{A_y} = \left[\begin{array}{ccccccccccccccccccc}
\\
\times & & & \times & & & & \times \\
& \times & & & \times & & & & \times \\
& & \times & & & \times & & & & \times \\
\\
& & & & \times & & & & \times & & & & \times \\
& & & & & \times & & & & \times & & & & \times \\
& & & & & & \times & & & & \times & & & & \times \\
\\
& & & & & & & & & \times & & & & \times & & & \times \\
& & & & & & & & & & \times & & & & \times & & & \times \\
& & & & & & & & & & & \times & & & & \times & & & \times \\
\\
\end{array}\right]$$

The sparsity pattern of $\mathbf{A_b}$ is given by, (illustration for a 5×5 grid)

$$\mathbf{A_b} = \left[\begin{array}{ccccccccccccccccccc}
\times \\
& \times \\
& & \times \\
& & & \times \\
& & & & \times \\
& & & & & \times \\
\\
& & & & & & & \times \\
& & & & & & & & \times \\
\\
& & & & & & & & & & \times \\
& & & & & & & & & & & \times \\
\\
& & & & & & & & & & & & & \times \\
& & & & & & & & & & & & & & \times \\
& & & & & & & & & & & & & & & \times \\
& & & & & & & & & & & & & & & & \times \\
& & & & & & & & & & & & & & & & & \times \\
& & & & & & & & & & & & & & & & & & \times
\end{array}\right]$$

A.2 2D Fourth order approximation

The sparsity pattern of $\mathbf{A_x}$ is given by, (illustration for a 7×7 grid)

$$\mathbf{A_x} = \left[\begin{array}{ccccccccccccccccccccc}
\times & \times & \times & \times & \times & & & & & & & & & & & & & & & & \\
& \times & \times & \times & \times & \times & & & & & & & & & & & & & & & \\
& & \times & \times & \times & \times & \times & & & & & & & & & & & & & & \\[4em]
& & & & & & & \times & \times & \times & \times & \times & & & & & & & & & \\
& & & & & & & & \times & \times & \times & \times & \times & & & & & & & & \\
& & & & & & & & & \times & \times & \times & \times & \times & & & & & & & \\[4em]
& & & & & & & & & & & & & & \times & \times & \times & \times & \times & & \\
& & & & & & & & & & & & & & & \times & \times & \times & \times & \times & \\
& & & & & & & & & & & & & & & & \times & \times & \times & \times & \times
\end{array}\right]$$

The sparsity pattern of $\mathbf{A_y}$ is given by, (illustration for a 7×7 grid)

$$\mathbf{A_y} = \left[\begin{array}{ccccccccccccccccccccccccccc}
\times & & & & \times & & & & \times & & & & \times & & & & \times & & & & & & & & & & \\
& \times & & & & \times & & & & \times & & & & \times & & & & \times & & & & & & & & & \\
& & \times & & & & \times & & & & \times & & & & \times & & & & \times & & & & & & & & \\[4em]
& & & & \times & & & & \times & & & & \times & & & & \times & & & & \times & & & & & & \\
& & & & & \times & & & & \times & & & & \times & & & & \times & & & & \times & & & & & \\
& & & & & & \times & & & & \times & & & & \times & & & & \times & & & & \times & & & & \\[4em]
& & & & & & & & \times & & & & \times & & & & \times & & & & \times & & & & \times & & \\
& & & & & & & & & \times & & & & \times & & & & \times & & & & \times & & & & \times & \\
& & & & & & & & & & \times & & & & \times & & & & \times & & & & \times & & & & \times
\end{array}\right]$$

The sparsity pattern of $\mathbf{A_b}$ is given by, (illustration for a 7 × 7 grid)