Numerical Solution of Steady-State Heat Equation

Jiaqi Li

November 2018

1 Governing Equations

In this document, we attempt to solve the steady-state heat equation in one- and two- dimensions. In 1D, we have

$$-kT''(x) = q(x), \quad x \in \Omega \tag{1}$$

In 2D, we have

$$-k\nabla^2 T(x,y) = q(x,y), \quad (x,y) \in \Omega \tag{2}$$

where k is the thermal conductivity, T is temperature, and q is a given heat source term. k is assumed to be constant throughout the domain. To simplify the problem, we assume $\Omega = (0,1)$ in 1D and $\Omega = (0,1) \times (0,1)$ in 2D. We further assume Dirichlet boundary condition:

$$T|_{\partial\Omega} = g \tag{3}$$

where $\partial\Omega$ denotes the boundary of domain Ω, and g is a given function defined on $\partial\Omega$. From PDE theory we know that the combination of equation (1)/(2) and (3) forms a well-posed problem provided that q and g are regular.

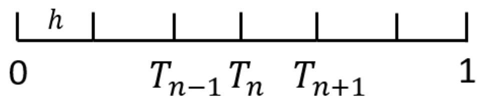
2 Discretization Schemes

Central difference is used to discretize the second-order derivative in the heat equation.

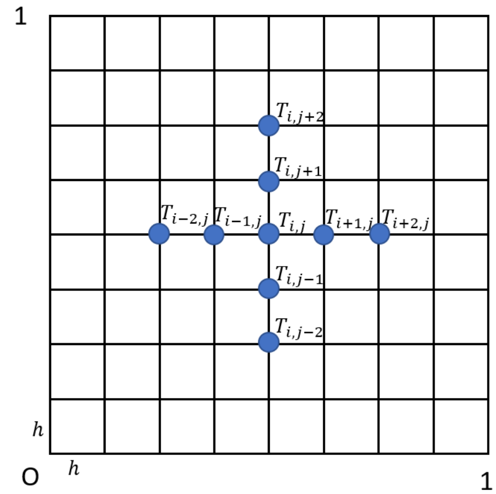
2.1 1D case

In 1D case, we partition the domain into uniform intervals. Assume $0 = x_0 < x_1 < ... < x_m = 1$, and $x_n - x_{n-1} = h,\ n = 1, 2, ..., m$. The second-order approximation to the second-order derivative is

$$T''(x_n) = \frac{1}{h^2}\left(T_{n-1} - 2T_n + T_{n+1}\right) + O(h^2)$$



(a) 1D mesh



(b) 2D mesh

Figure 1: Representative figures of 1D and 2D mesh

where $T_n = T(x_n)$. Then the discrete approximation to equation (1) is

$$\frac{1}{h^2}\left(T_{n-1} - 2T_n + T_{n+1}\right) + O(h^2) = -\frac{q_n}{k}$$

where $q_n = q(x_n)$. In order to get an approximate solution, we neglect the second-order term, and obtain

$$T_{n-1} - 2T_n + T_{n+1} = -\frac{q_n h^2}{k}, \quad n = 1, 2, ..., m-1 \tag{4}$$

The boundary condition (3) implies

$$T_0 = g(0), \quad T_m = g(1) \tag{5}$$

Equations (4) and (5) form a solvable linear system.

The fourth-order scheme is

$$T''(x_n) = \frac{1}{h^2}\left(-\frac{1}{12}T_{n-2} + \frac{4}{3}T_{n-1} - \frac{5}{2}T_n + \frac{4}{3}T_{n+1} - \frac{1}{12}T_{n+2}\right) + O(h^4)$$

The corresponding discrete approximation to equation (1) is

$$\frac{1}{h^2}\left(-\frac{1}{12}T_{n-2} + \frac{4}{3}T_{n-1} - \frac{5}{2}T_n + \frac{4}{3}T_{n+1} - \frac{1}{12}T_{n+2}\right) + O(h^4) = -\frac{q_n}{k}$$

For practical calculation, we neglect the fourth-order term, obtaining

$$-\frac{1}{12}T_{n-2} + \frac{4}{3}T_{n-1} - \frac{5}{2}T_n + \frac{4}{3}T_{n+1} - \frac{1}{12}T_{n+2} = -\frac{q_n h^2}{k}, \quad n = 2, 3, ..., m-2 \tag{6}$$

The boundary condition (5) still applies. However, we need two more equations to close the system. A simple choice is to use second-order difference at the nodes adjacent to the boundary.

$$T_{n-1} - 2T_n + T_{n+1} = -\frac{q_n h^2}{k}, \quad n = 1, m-1 \tag{7}$$

Equations (6) together with (5) and (7) form a solvable linear system.

2.2 2D case

In 2D case, we partition the domain uniformly in both x and y direction. Let $0 = x_0 < x_1 < ... < x_m = 1$, and $0 = y_0 < y_1 < ... < y_n = 1$. Let Δx and Δy be the mesh size in x and y direction, respectively. The second-order approximation to derivatives in the heat equation (2) is

$$\begin{aligned}\nabla^2 T(x_i, y_j) &= \left(\frac{\partial}{\partial x^2} + \frac{\partial}{\partial y^2}\right) T|_{(x_i, y_j)} \\ &= \frac{1}{\Delta x^2}(T_{i-1,j} - 2T_{i,j} + T_{i+1,j}) + \frac{1}{\Delta y^2}(T_{i,j-1} - 2T_{i,j} + T_{i,j+1}) + O(\Delta x^2) + O(\Delta y^2)\end{aligned}$$

The approximation to the equation is

$$\frac{1}{\Delta x^2}(T_{i-1,j} - 2T_{i,j} + T_{i+1,j}) + \frac{1}{\Delta y^2}(T_{i,j-1} - 2T_{i,j} + T_{i,j+1}) + O(\Delta x^2) + O(\Delta y^2) = -\frac{q_{i,j}}{k}$$

where $q_{i,j} = q(x_i, y_j)$. When $m = n$, and therefore $\Delta x = \Delta y = h$, the above equation can be simplified as

$$T_{i,j-1} + T_{i-1,j} - 4T_{i,j} + T_{i+1,j} + T_{i,j+1} = -\frac{q_{i,j} h^2}{k}, \quad i, j = 1, 2, ..., n-1 \tag{8}$$

where we have omitted the second-order error $O(h^2)$. The boundary condition (3) implies

$$T_{0,j} = g(0, y_j), \quad T_{n,j} = g(1, y_j), \quad T_{i,0} = g(x_i, 0), \quad T_{i,n} = g(x_i, 1) \tag{9}$$

for $i = 1, 2, ..., n-1$ and $j = 0, 1, ..., n$. Equations (8) and (9) form a closed linear system.

The fourth-order approximation to derivatives is

$$\begin{aligned}\nabla^2 T(x_i, y_j) =& \frac{1}{\Delta x^2}\left(-\frac{1}{12}T_{i-2,j} + \frac{4}{3}T_{i-1,j} - \frac{5}{2}T_{i,j} + \frac{4}{3}T_{i+1,j} - \frac{1}{12}T_{i+2,j}\right) + O(\Delta x^4) \\ &+ \frac{1}{\Delta y^2}\left(-\frac{1}{12}T_{i,j-2} + \frac{4}{3}T_{i,j-1} - \frac{5}{2}T_{i,j} + \frac{4}{3}T_{i,j+1} - \frac{1}{12}T_{i,j+2}\right) + O(\Delta y^4)\end{aligned}$$

The approximation to the heat equation (2) is

$$\begin{aligned}&\frac{1}{\Delta x^2}\left(-\frac{1}{12}T_{i-2,j} + \frac{4}{3}T_{i-1,j} - \frac{5}{2}T_{i,j} + \frac{4}{3}T_{i+1,j} - \frac{1}{12}T_{i+2,j}\right) + O(\Delta x^4) \\ &+ \frac{1}{\Delta y^2}\left(-\frac{1}{12}T_{i,j-2} + \frac{4}{3}T_{i,j-1} - \frac{5}{2}T_{i,j} + \frac{4}{3}T_{i,j+1} - \frac{1}{12}T_{i,j+2}\right) + O(\Delta y^4) = -\frac{q_{i,j}}{k}\end{aligned}$$

Assume $m = n, \Delta x = \Delta y = h$. Then the above equation can be simplified:

$$-\frac{1}{12}T_{i,j-2} + \frac{4}{3}T_{i,j-1} - \frac{1}{12}T_{i-2,j} + \frac{4}{3}T_{i-1,j} - 5T_{i,j} \\ + \frac{4}{3}T_{i+1,j} - \frac{1}{12}T_{i+2,j} + \frac{4}{3}T_{i,j+1} - \frac{1}{12}T_{i,j+2} = -\frac{q_{i,j}h^2}{k} \quad (10)$$

for $i, j = 2, 3, ..., n-2$, where we have omitted the fourth-order error $O(h^4)$. The boundary condition (9) still holds, but we need $(4n - 8)$ more equations to close the system. A simple choice is to use second-order difference at the nodes adjacent to the boundary:

$$T_{i,j-1} + T_{i-1,j} - 4T_{i,j} + T_{i+1,j} + T_{i,j+1} = -\frac{q_{i,j}h^2}{k} \quad (11)$$

where $1 \leq i, j \leq n - 1$ and at least one of i, j equals 1 or $n - 1$. Equations (10) together with (9) and (11) form a closed linear system.

3 Implementation

3.1 Resulting linear system

3.1.1 1D case

Let

$$\boldsymbol{u} = \begin{bmatrix} T_0 & T_1 & \dots & T_m \end{bmatrix}^T$$

be a column vector of unknowns. The second-order finite difference scheme (4), (5) gives the following tridiagonal linear system

$$\begin{bmatrix} 1 & & & & & \\ 1 & -2 & 1 & & & \\ & 1 & -2 & 1 & & \\ & & \ddots & \ddots & \ddots & \\ & & & 1 & -2 & 1 \\ & & & & & 1 \end{bmatrix} \begin{bmatrix} T_0 \\ T_1 \\ T_2 \\ \vdots \\ T_{m-1} \\ T_m \end{bmatrix} = \begin{bmatrix} g(0) \\ -q_1 h^2/k \\ -q_2 h^2/k \\ \vdots \\ -q_{m-1} h^2/k \\ g(1) \end{bmatrix}$$

where zeros in the matrix are omitted. There are three non-zero entries on an interior row of the matrix.

The fourth-order difference scheme (6), (5), and (7) results in the following linear system

$$\begin{bmatrix} 1 & & & & & & \\ 1 & -2 & 1 & & & & \\ -\frac{1}{12} & \frac{4}{3} & -\frac{5}{2} & \frac{4}{3} & -\frac{1}{12} & & \\ & \ddots & \ddots & \ddots & \ddots & \ddots & \\ & & -\frac{1}{12} & \frac{4}{3} & -\frac{5}{2} & \frac{4}{3} & -\frac{1}{12} \\ & & & & 1 & -2 & 1 \\ & & & & & & 1 \end{bmatrix} \begin{bmatrix} T_0 \\ T_1 \\ T_2 \\ \vdots \\ T_{m-2} \\ T_{m-1} \\ T_m \end{bmatrix} = \begin{bmatrix} g(0) \\ -q_1 h^2/k \\ -q_2 h^2/k \\ \vdots \\ -q_{m-2} h^2/k \\ -q_{m-1} h^2/k \\ g(1) \end{bmatrix}$$

There are five non-zero entries on an interior row of the matrix.

3.1.2 2D case

Assume $m = n$. Let

$$\boldsymbol{\alpha}_j = \begin{bmatrix} T_{0,j} & T_{1,j} & \dots & T_{n,j} \end{bmatrix}^T, \quad j = 0, 1, \dots, n$$

The second-order difference scheme (8), (9) gives

$$\boldsymbol{C}\boldsymbol{\alpha}_{j-1} + \boldsymbol{B}\boldsymbol{\alpha}_j + \boldsymbol{C}\boldsymbol{\alpha}_{j+1} = \boldsymbol{f}_j, \quad j = 1, 2, \dots, n-1 \tag{12}$$

where

$$\boldsymbol{B} = \begin{bmatrix} 1 & & & & \\ 1 & -4 & 1 & & \\ & \ddots & \ddots & \ddots & \\ & & 1 & -4 & 1 \\ & & & & 1 \end{bmatrix}, \quad \boldsymbol{C} = \begin{bmatrix} 0 & & & & \\ & 1 & & & \\ & & \ddots & & \\ & & & 1 & \\ & & & & 0 \end{bmatrix}$$

are $(n+1) \times (n+1)$ matrices, and

$$\boldsymbol{f}_j = \begin{bmatrix} T_{0,j} & -q_{1,j} h^2/k & \dots & -q_{n-1,j} h^2/k & T_{n,j} \end{bmatrix}^T$$

is column vector with length $(n+1)$. Rewriting equation (12) using block matrices, we obtain

$$\begin{bmatrix} \boldsymbol{I} & & & & \\ \boldsymbol{C} & \boldsymbol{B} & \boldsymbol{C} & & \\ & \ddots & \ddots & \ddots & \\ & & \boldsymbol{C} & \boldsymbol{B} & \boldsymbol{C} \\ & & & & \boldsymbol{I} \end{bmatrix} \begin{bmatrix} \boldsymbol{\alpha}_0 \\ \boldsymbol{\alpha}_1 \\ \vdots \\ \boldsymbol{\alpha}_{n-1} \\ \boldsymbol{\alpha}_n \end{bmatrix} = \begin{bmatrix} \boldsymbol{\alpha}_0 \\ \boldsymbol{f}_1 \\ \vdots \\ \boldsymbol{f}_{n-1} \\ \boldsymbol{\alpha}_n \end{bmatrix}$$

Note that $\boldsymbol{\alpha}_0$ and $\boldsymbol{\alpha}_n$ on the right hand side is given by boundary condition (9). There are five non-zero entries on an interior row of the matrix.

The second-order scheme (10), (9), (11) results in the following equations for $\boldsymbol{\alpha}_j$:

$$\boldsymbol{F}\boldsymbol{\alpha}_{j-2} + \boldsymbol{E}\boldsymbol{\alpha}_{j-1} + \boldsymbol{D}\boldsymbol{\alpha}_j + \boldsymbol{E}\boldsymbol{\alpha}_{j+1} + \boldsymbol{F}\boldsymbol{\alpha}_{j+2} = \boldsymbol{f}_j, \quad j = 2, 3, ..., n-2$$

where

$$\boldsymbol{D} = \begin{bmatrix} 1 & & & & & & \\ 1 & -4 & 1 & & & & \\ -\frac{1}{12} & \frac{4}{3} & -5 & \frac{4}{3} & -\frac{1}{12} & & \\ & \ddots & \ddots & \ddots & \ddots & \ddots & \\ & & -\frac{1}{12} & \frac{4}{3} & -5 & \frac{4}{3} & -\frac{1}{12} \\ & & & & 1 & -4 & 1 \\ & & & & & & 1 \end{bmatrix}$$

$$\boldsymbol{E} = \begin{bmatrix} 0 & & & & & & \\ & 1 & & & & & \\ & & \frac{4}{3} & & & & \\ & & & \ddots & & & \\ & & & & \frac{4}{3} & & \\ & & & & & 1 & \\ & & & & & & 0 \end{bmatrix}, \boldsymbol{F} = \begin{bmatrix} 0 & & & & & & \\ & 0 & & & & & \\ & & -\frac{1}{12} & & & & \\ & & & \ddots & & & \\ & & & & -\frac{1}{12} & & \\ & & & & & 0 & \\ & & & & & & 0 \end{bmatrix},$$

are $(n+1) \times (n+1)$ matrices. Then the global system is

$$\begin{bmatrix} \boldsymbol{I} & & & & & & \\ \boldsymbol{C} & \boldsymbol{B} & \boldsymbol{C} & & & & \\ \boldsymbol{F} & \boldsymbol{E} & \boldsymbol{D} & \boldsymbol{E} & \boldsymbol{F} & & \\ & \ddots & \ddots & \ddots & \ddots & \ddots & \\ & & \boldsymbol{F} & \boldsymbol{E} & \boldsymbol{D} & \boldsymbol{E} & \boldsymbol{F} \\ & & & & \boldsymbol{C} & \boldsymbol{B} & \boldsymbol{C} \\ & & & & & & \boldsymbol{I} \end{bmatrix} \begin{bmatrix} \boldsymbol{\alpha}_0 \\ \boldsymbol{\alpha}_1 \\ \boldsymbol{\alpha}_2 \\ \vdots \\ \boldsymbol{\alpha}_{n-2} \\ \boldsymbol{\alpha}_{n-1} \\ \boldsymbol{\alpha}_n \end{bmatrix} = \begin{bmatrix} \boldsymbol{\alpha}_0 \\ \boldsymbol{f}_1 \\ \boldsymbol{f}_2 \\ \vdots \\ \boldsymbol{f}_{n-2} \\ \boldsymbol{f}_{n-1} \\ \boldsymbol{\alpha}_n \end{bmatrix}$$

There are nine non-zero entries on an interior row of the matrix.

3.2 Iterative solutions for the resulting linear system

Once the linear system is formed, we utilize iterative methods to solve the system. Suppose the system is

$$\boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}$$

where $\boldsymbol{A}$ is an $n \times n$ matrix.

3.2.1 Jacobi method

Algorithm 1 Jacobi iteration

$\boldsymbol{x}^{(0)} \leftarrow 0$
for $k = 0, 1, \dots$ **do**

$$x_i^{(k+1)} \leftarrow \frac{1}{a_{ii}} \left(b_i - \sum_{j \neq i} a_{ij} x_j^{(k)} \right)$$

  if $||\boldsymbol{x}^{(k+1)} - \boldsymbol{x}^{(k)}|| < tol$ **then**
    exit
  end if
end for

3.2.2 Gauss-Seidel method

Algorithm 2 Gauss-Seidel iteration

$\boldsymbol{x}^{(0)} \leftarrow 0$
for $k = 0, 1, \dots$ **do**

$$x_i^{(k+1)} \leftarrow \frac{1}{a_{ii}} \left(b_i - \sum_{j=1}^{i-1} a_{ij} x_j^{(k+1)} - \sum_{j=i+1}^{n} a_{ij} x_j^{(k)} \right)$$

  if $||\boldsymbol{x}^{(k+1)} - \boldsymbol{x}^{(k)}|| < tol$ **then**
    exit
  end if
end for

3.3 Required memory

We will need to store two discrete temperature fields and one source term field. Using double precision, in 1D, the memory required is $24n$ Bytes; in 2D, we need $24n^2$ Bytes. Since we are using iterative methods to solve the system, the matrix $\boldsymbol{A}$ is never explicitly formed. In order to calculate the difference between two iterations, we need keep a copy of $\boldsymbol{x}^{(k)}$ while computing $\boldsymbol{x}^{(k+1)}$.