Numerical approximation of Elliptic PDE : 2D

Numerical approximation is not an elegant subject. It is a collection of technical details and dirty work. However, is the more convenient way to solve real world problems.

The aim here is to define a numerical scheme for the 2D Laplace equation with Dirichlet boundary conditions :

$$-\frac{\partial^2 T}{\partial x^2} - \frac{\partial^2 T}{\partial y^2} = F(\boldsymbol{x}, T) \quad \text{for any} \quad \boldsymbol{x} = (x, y)^T \in \Omega = (0,1) \times (0,1)$$

$$\text{with} \quad T(x=0, y) = T_{x0}(y), \quad T(x=1, y) = T_{x1}(y), \quad \text{for any} \quad y \in [0,1]$$

$$\text{with} \quad T(x, y=0) = T_{y0}(x), \quad T(x, y=1) = T_{y1}(x), \quad \text{for any} \quad x \in]0,1[$$

Function $F(\boldsymbol{x}, T)$, $T_{x0}(y)$, $T_{x1}(y)$, $T_{y0}(x)$ and $T_{y1}(x)$ are given.
The numerical approximation use discrete points $\boldsymbol{x}_{i,j}$, localized in the domain Ω, defined as

$$\boldsymbol{x}_{i,j} = \begin{pmatrix} i * \delta x \\ j * \delta y \end{pmatrix} \quad \text{for} \quad i = 1, \cdots, N_x \quad \text{and} \quad j = 1, \cdots, N_y \quad \text{with} \quad \delta x = \frac{1}{N_x + 1}, \quad \delta y = \frac{1}{N_y + 1},$$

The numerical solution is defined by the values $\tilde{T}_{i,j}$ solutions of :

$$-\frac{\tilde{T}_{i+1,j} - 2\tilde{T}_{i,j} + \tilde{T}_{i-1,j}}{\delta x^2} - \frac{\tilde{T}_{i,j+1} - 2\tilde{T}_{i,j} + \tilde{T}_{i,j-1}}{\delta y^2} = F(\boldsymbol{x}_{i,j}, \tilde{T}_{i,j}) \quad \text{for} \quad \begin{cases} i = 1, \cdots, N_x \\ j = 1, \cdots, N_y \end{cases} \tag{0.1}$$

For $j = 0, \cdots, N_y + 1$ we have $\tilde{T}_{0,j} = T_{x0}(y_j)$ and $\tilde{T}_{N_x+1,j} = T_{x1}(y_j)$.
For $i = 1, \cdots, N_x$ we have $\tilde{T}_{i,0} = T_{y0}(x_i)$ and $\tilde{T}_{i,N_y+1} = T_{y1}(x_i)$.

1. Compute the truncation error of (0.1) and show that it is of order of δh^2 where $\delta h = \max(\delta x, \delta y)$
2. Arrange the system (0.1) into a linear system of the form

   $$\underline{A}\tilde{\boldsymbol{T}} = \boldsymbol{S}, \quad \text{where} \quad \tilde{\boldsymbol{T}} = \left(\tilde{T}_1, \cdots, \tilde{T}_{is}, \cdots \tilde{T}_{Ns}\right)^T, \quad \text{with} \quad \tilde{T}_{is} \equiv \tilde{T}_{i,j} \quad \text{where} \quad is(i,j) = i + (i-1) * N_x$$

   and define $\underline{A}$ and $\boldsymbol{S}$. As a simple example, consider the case with $N_x = N_y = 3$.
3. The solution of the previous system is assume to be defined as $\tilde{\boldsymbol{T}} = \lim_{k \to \infty} \tilde{\boldsymbol{T}}^k$, $\quad k > 0 \in \mathbb{N}$, where $\tilde{\boldsymbol{T}}^{k+1}$ are defined by Jacobi relaxations :

   $$\tilde{\boldsymbol{T}}_{is}^{k+1} = \frac{\delta x^2 \delta y^2}{2(\delta x^2 + \delta y^2)} \left(\frac{\tilde{T}^k_{i+1,j} + \tilde{T}^k_{i-1,j}}{\delta x^2} + \frac{\tilde{T}^k_{i,j+1} + \tilde{T}^k_{i,j-1}}{\delta y^2} + F(\boldsymbol{x}_{i,j}, \tilde{T}^k_{i-1,j}) \right)$$

   for any initial $\tilde{\boldsymbol{T}}^0$. Consider the case of $F(\boldsymbol{x}, T) = \frac{\pi^2}{2} \sin(\pi x) \sin(\pi y)$ and $\tilde{T}_{i,0} = T_{y0}(x_i) = \tilde{T}_{i,N_y+1} = T_{y1}(x_i) \equiv 0$. Then, for $\tilde{\boldsymbol{T}}^0 \equiv 0$, Compute (with a fortran or scilab program) program $\tilde{\boldsymbol{T}}^K$ for $N_x = N_y = 100$ and $N_x = N_y = 200$, then gives a numerical estimation of the truncation error when $K = 10$, $K = 100$, $K = 10000$. Make the same analysis when $\tilde{\boldsymbol{T}}^{k+1}$ is computed by Gauss-Seidel, Conjugate-Gradient methods.

Radiatif transfert case

In this case fte function $F(\boldsymbol{x},T)$ is defined as

$$F(\boldsymbol{x},T) = -\alpha\,(T-\theta_e)^4\,, \quad \tilde{T}_{i,0} = T_{y0}(x_i) = \tilde{T}_{i,N_y+1} = T_{y1}(x_i) = \theta_i$$

where θ_e, θ_i and α are given constants. For example use $\theta_e = 20$ for different values of $\frac{\theta_i}{\theta_e}$ and α.

$$\frac{\theta_i}{\theta_e} = 2, \quad 4, \quad 6 \quad 8, ... \quad \text{and} \quad \alpha = 10^{-1}, \quad 10^{-2}, \quad 10^{-4}, \quad 10^{-6}, ...$$

Gauss-Seidel Relaxation

$$\tilde{\boldsymbol{T}}_{is}^{k+1} = \frac{\delta x^2 \delta y^2}{2(\delta x^2 + \delta y^2)} \left(\frac{\tilde{T}_{i+1,j}^{k} + \tilde{T}_{i-1,j}^{k+1}}{\delta x^2} + \frac{\tilde{T}_{i,j+1}^{k} + \tilde{T}_{i,j-1}^{k+1}}{\delta y^2} + F(\boldsymbol{x}_{i,j}) \right)$$

Conjugate Gradient relaxation

$$\tilde{\boldsymbol{T}}^{k+1} = \tilde{\boldsymbol{T}}^{k} + \alpha_k \boldsymbol{d}_k, \quad \text{and} \quad \boldsymbol{d}^{k+1} = \boldsymbol{r}^{k+1} + \beta_k \boldsymbol{d}_k,$$
$$\text{where} \quad \alpha_k = \frac{\boldsymbol{r}^k \cdot \boldsymbol{r}^k}{\boldsymbol{d}^k \cdot \underline{A}\boldsymbol{d}^k} \quad \text{and} \quad \beta_k = \frac{\boldsymbol{r}^{k+1} \cdot \boldsymbol{r}^{k+1}}{\boldsymbol{r}^k \cdot \boldsymbol{r}^k}$$

with $\boldsymbol{r}^k = \boldsymbol{S} - \underline{A}\tilde{\boldsymbol{T}}^k$ (the residual) and the initial descent direction $\boldsymbol{d}^0 = \boldsymbol{r}^0$