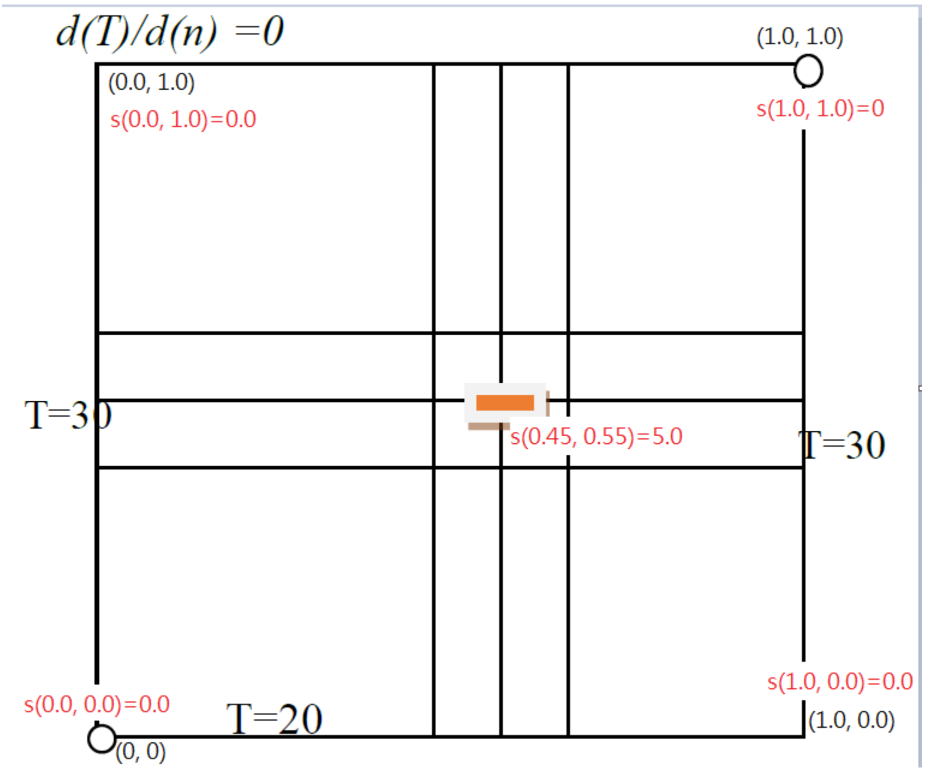
作業五說明文件



1. Boundary condition $\begin{cases} T = 20,\ y = 0 \\ T = 30, x = 0\ or\ x = 1.0 \\ \frac{dT}{dn} = 0, y = 1.0 \end{cases}$，T 代表溫度，n 代表法向量

2. Source function $s(x, y) = \begin{cases} 5, (x, y) = (0.45, 0.55) \\ 0.0, (x, y) \neq (0.45, 0.55) \end{cases}$

3. 對於所有位於 D=[0,1]x[0,1]中的點，滿足 Poisson equation:

$$\frac{\partial^2 T}{\partial x^2} + \frac{\partial^2 T}{\partial y^2} = -s(x, y)$$

以 finite difference method 表示(投影片 40_Differentiate 第 7 頁)

$$f''(x) \approx \frac{\frac{f(x+2h)-f(x+h)}{h} - \frac{f(x+h)-f(x)}{h}}{h} = \frac{f(x+2h)-2f(x+h)+f(x)}{h^2}.$$

$$\frac{T_{i+1,j}-2T_{i,j}+T_{i-1,j}}{h^2} + \frac{T_{i,j+1}-2T_{i,j}+T_{i,j-1}}{h^2} = -s(x,y)\text{，}h = 0.05$$

4. 若想求得各網格點的溫度$T_{i,j}$，也就是$T_{1,1}$~$T_{19,20}$，需要求解以下方程式

$$T_{0,1} + T_{2,1} + T_{1,0} + T_{1,2} - 4T_{1,1} = -s(0.05,0.05)h^2$$

$$T_{1,1} + T_{3,1} + T_{2,0} + T_{2,2} - 4T_{2,1} = -s(0.1,0.05)h^2$$

$$T_{2,1} + T_{4,1} + T_{3,0} + T_{3,2} - 4T_{3,1} = -s(0.15,0.05)h^2$$

$$\vdots$$

$$T_{18,19} + T_{20,19} + T_{19,18} + T_{19,20} - 4T_{19,19} = -s(0.95,0.95)h^2$$

上述等式僅足以提供 19 ∗ 19 個等式，但實際上有 19 ∗ 20 個變數

因此加入 boundary condition $\frac{dT}{dn} = 0$，*並套用* $finite\ differentiate\ method$

$$D_f = f'(x) \approx \frac{f(x)-f(x-h)}{h}.$$

(投影片 40_$Differentiate$第 4 頁)

$$\frac{T_{1,20} - T_{1,19}}{h} = 0$$

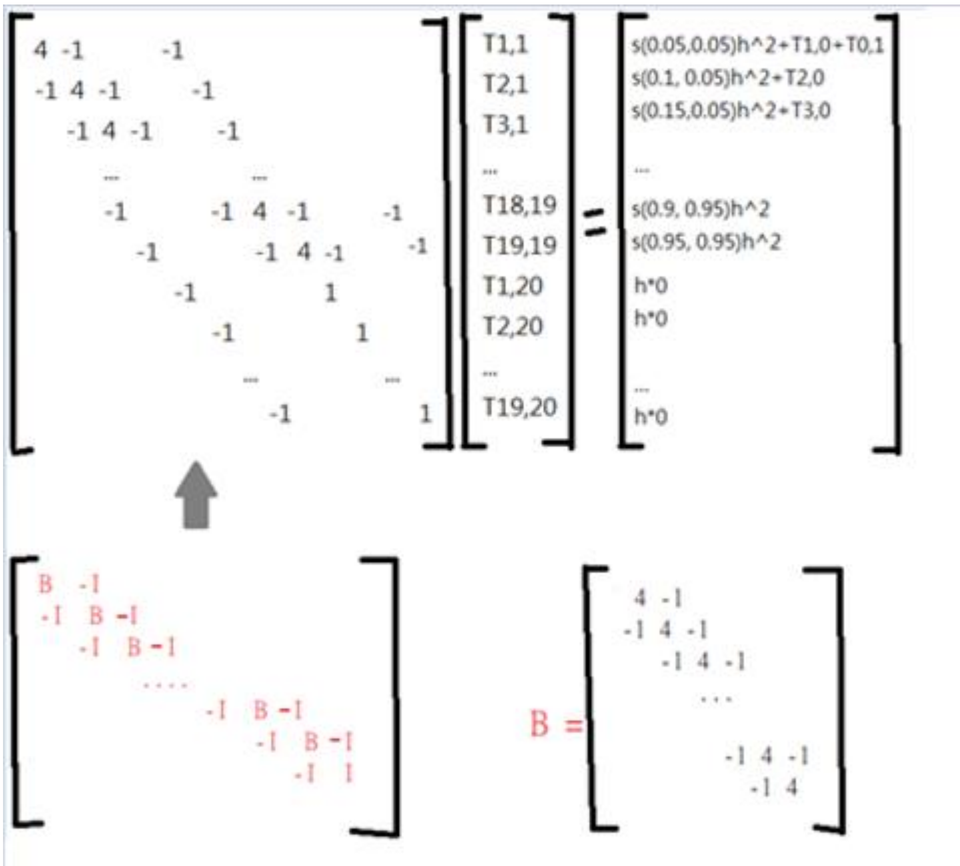
$$\frac{T_{2,20} - T_{2,19}}{h} = 0$$

$$\vdots$$

$$\frac{T_{19,20} - T_{19,19}}{h} = 0$$

矩陣表達式如下($T_{0,1}$ 、$T_{1,0}$ 、$T_{2,0}$ 、$T_{20,19}$ *為已知，因為* $boundary\ condition$)

完整版



拆分成兩個矩陣的形式如下

$$\begin{bmatrix} 4 & -1 & & & -1 & & \\ -1 & 4 & -1 & & & & \\ & & \ddots & & & & \\ & & & & -1 & 4 & -1 \\ & & & & & -1 & 4 \end{bmatrix} \begin{bmatrix} T_{1,1} \\ T_{2,1} \\ \vdots \\ T_{18,19} \\ T_{19,19} \end{bmatrix} = \begin{bmatrix} s(0.05,0.05)h^2 + T_{0,1} + T_{1,0} \\ s(0.1,0.05)h^2 + T_{2,0} \\ \vdots \\ s(0.95,0.95)h^2 + T_{19,20} + T_{20,19} \end{bmatrix}$$

$$
\begin{bmatrix} 1 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & 1 \end{bmatrix}
\begin{bmatrix} T_{1,20} \\ \vdots \\ T_{19,20} \end{bmatrix}
=
\begin{bmatrix} 0 * h + T_{1,19} \\ \vdots \\ 0 * h + T_{19,19} \end{bmatrix}
$$

⊗完整版或拆分兩個矩陣的形式都可以用來求解

5. 試著用以下方式針對以上矩陣求解$T_{1,1}$~$T_{19,20}$，終止條件為$|T^{n+1} - T^{n}| \leq 0.00001$，所有未知的$T_{i,j}$初始值皆設為 0，並將最終收斂的計算結果、迭代數(iteration)列印出來。
   A. Gauss-Seidol method, w = 1.0(30%)
   B. SOR method with w = 1.2(30%)
6. 試著代入不同的 boundary condition 或 source function，並展示其結果。(15%)
7. 將$w = 1.0$~1.95一一代入 SOR method，看看 w 為何者時，會最快收斂得到答案，並列印出所有的迭代數，試著比較其結果。(15%)
8. (二選一)試著將各網格點溫度轉換成顏色資訊，並用工具渲染出來。(15%)
9. (二選一)試著將網格大小調整成 10x10、30x30、40x40，並找出最佳的 w。(10%)
10. 如果對題目有不了解或寫作業時有遇到困難，都可以到實驗室詢問助教。
11. Gaussial Seidol method、SOR 程式可參考
    \SampleCodes\LinearSystem\iterative\GaussSeidel_SOR
12. 繳交期限 – 1/26(三)
13. Demo 期限 – 1/26(三)
14. 實作流程解釋
    - SOR 公式

$$
x_i^{(k+1)} = x_i^{(k)} + \frac{\omega}{a_{ii}} \left[b_i - \sum_{j=0}^{i-1} a_{ij} x_j^{(k+1)} - \sum_{j=i}^{n-1} a_{ij} x_j^{(k)} \right]
$$

    - 欲求解的矩陣

$$
\left[\begin{array}{cccccccccccc}
4 & -1 & & & -1 & & & & & & & \\
-1 & 4 & -1 & & & -1 & & & & & & \\
 & -1 & 4 & -1 & & & -1 & & & & & \\
 & & & \cdots & & & & \cdots & & & & \\
 & & & -1 & & & -1 & 4 & -1 & & -1 & \\
 & & & & -1 & & & -1 & 4 & -1 & & -1 \\
 & & & & & -1 & & & & 1 & & \\
 & & & & & & -1 & & & & 1 & \\
 & & & & & & & \cdots & & & \cdots & \\
 & & & & & & & & -1 & & & 1
\end{array}\right]
\begin{bmatrix} T1,1 \\ T2,1 \\ T3,1 \\ \cdots \\ T18,19 \\ T19,19 \\ T1,20 \\ T2,20 \\ \cdots \\ T19,20 \end{bmatrix}
=
\begin{bmatrix} s(0.05,0.05)h^2+T1,0+T0,1 \\ s(0.1, 0.05)h^2+T2,0 \\ s(0.15,0.05)h^2+T3,0 \\ \cdots \\ s(0.9, 0.95)h^2 \\ s(0.95, 0.95)h^2 \\ h*0 \\ h*0 \\ \cdots \\ h*0 \end{bmatrix}
$$

實際計算的流程如下(這裡用$\omega = 1$，也就是 Gausial Seidol method):

$$
T_{1,1}{}^{1} = T_{1,1}{}^{0} + \frac{1}{4}\left[-s(0.05,0.05)h^2 + T_{1,0} + T_{0,1} + T_{2,1} + T_{1,2} - 4T_{1,1} \right]
$$

$$
T_{2,1}{}^{1} = T_{2,1}{}^{0} + \frac{1}{4}\left[-s(0.1,0.05)h^2 + T_{1,1} + T_{3,1} + T_{2,0} + T_{2,2} - 4T_{2,1} \right]
$$

$$T_{3,1}{}^{1} = T_{3,1}{}^{0} + \frac{1}{4}\left[-s(0.15,0.05)h^2 + T_{2,1} + T_{4,1} + T_{3,0} + T_{3,2} - 4T_{3,1}\right]$$

...

仔細觀察後，可以得到一個通式

$$T_{i,j}{}^{k+1} =$$

$$T_{i,j}{}^{k} + \frac{1}{4}\left[-s(0.05*(i+1),0.05*(j+1))h^2 + T_{i-1,j} + T_{i+1,j} + T_{i,j-1} + T_{i,j+1} - 4T_{i,j}\right]$$

以上是針對 T1,1~T19,19 而得的通式，而計算 T1,20~T19,20 的流程如下

$$T_{1,20}{}^{1} = T_{1,20}{}^{0} + \frac{1}{1}[0*h + T_{1,19} - T_{1,20}]$$

$$T_{2,20}{}^{1} = T_{2,20}{}^{0} + \frac{1}{1}[0*h + T_{2,19} - T_{2,20}]$$

...

可以得到另一個通式

$$T_{i,j}{}^{k+1} = T_{i,j}{}^{k} + 0*h + T_{i,j-1} - T_{i,j}$$

- 因此實際寫程式的 Pesudo code 如下

SOR Algorithm

$$x_i{}^{(k+1)} = x_i{}^{(k)} + \frac{\omega}{a_{ii}}\left[b_i - \sum_{j=0}^{i-1} a_{ij}x_j{}^{(k+1)} - \sum_{j=i}^{n-1} a_{ij}x_j{}^{(k)}\right]$$

```
//Initialize the solution.
 x[] = {0.0};
 err = ∞;
//Iterate until being converged
 while(err>ε){
    for(i=0;i≤n-1;i++){
       sum = b[i];
       for(j=0;j≤n-1;j++)  sum = sum – A[i][j]*x[j];
       x[i] = x[i] + ω*sum/A[i][i];
    }
    //Compute the residual
    r[] = b[] – A[][]*x[];
    err = norm_inf(r, n); //Compute the norm of the residual
 }
 return (x[]);
```

這裡就可以直接置換成右邊兩個通式了

j != 20

$$T_{i,j}{}^{k+1} = T_{i,j}{}^{k} + \frac{1}{4}\left[s(0.05*(i+1),0.05*(j+1))h^2 + T_{i-1,j} + T_{i+1,j} + T_{i,j-1} + T_{i,j+1} - 4T_{i,j}\right]$$

$T_{i,j}{}^{k+1} = T_{i,j}{}^{k} + 0*h + T_{i,j-1} - T_{i,j}$ j=20的情況

r可以設成 $T_{i,j}{}^{k+1} - T_{i,j}{}^{k}$

err依樣找inf-norm就行