Node Ordering, Numbering and Boundary conditions.

Matt Ueckermann, Pierre Lermusiaux and Tapovan Lolla.

March 11, 2013

Some of you may have difficulties with the implementation of solution techniques for differential equations involving two or more spatial directions (e.g. for Problem Set 3). Here is a short description of a possible solution procedure.

When differential equations involving one spatial dimension (1D in space) are discretized, the final numerical problem to be solved is often of the form $\mathbf{Ax} = \mathbf{b}$. In this case, $\mathbf{x}$ is a vector that contains the values/integrals defined on the 1D spatial grid.

When differential equations involving two or more spatial dimensions (2D or more in space) are discretized, the numerical problem to be solved can still often be written in the form $\mathbf{Ax} = \mathbf{b}$. In these higher dimensional cases, $\mathbf{x}$ is a vector that contains the values/integrals defined on the 2D (or more) spatial grid. To do this, one can define a pointer array that allows to map the positions on the 2D (or higher) spatial grid to the positions in the vector. For example, for *imax* and *jmax* points in a 2D structured spatial domain, ordering along the i direction first, then the j direction:

```
do $j = 1$, $jmax$
  do $i = 1$, $imax$
    xTvector($(j-1)*imax+i$) = $T(i,j)$
  enddo
enddo
```

The xTvector array contains a vector form of the $T(i, j)$ values.

The above is all one would need for a structured grid and it may suffice for several of you. However, below, we provide more detail on a possible implementation technique. Feel free to contact us or attend our office hours for any other difficulties.

1 Boundary Conditions

There are a number of ways in which boundary conditions can be incorporated in a matrix. The major difference is if the equation for the boundary conditions is part of the $\mathbf{A}$ matrix, or if the contributions due to the boundary condition is directly added to the RHS vector $\mathbf{b}$.

Consider the 1D problem shown in Figure (1) with the following equation in the interior: $\frac{T_{i+1}-2T_i+T_{i-1}}{\Delta x^2} = f_i$, and Dirichlet boundary conditions (i.e. $T_{1,4} = g_D$).

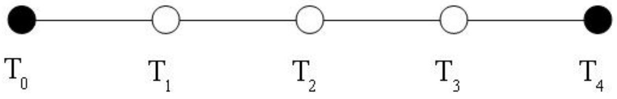


Figure 1: 1D grid

1.1 Boundary conditions included in A matrix

The equations governing this problem is can be written as:

$$\begin{aligned} T_0 &= g_D \\ T_2 - 2T_1 + T_0 &= \Delta x^2 f_1 \\ T_3 - 2T_2 + T_1 &= \Delta x^2 f_2 \\ T_4 - 2T_3 + T_2 &= \Delta x^2 f_3 \\ T_4 &= g_D \end{aligned}$$

We can write these equations directly as a matrix as follows:

$$\mathbf{A} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 1 & -2 & 1 & 0 & 0 \\ 0 & 1 & -2 & 1 & 0 \\ 0 & 0 & 1 & -2 & 1 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} T_0 \\ T_1 \\ T_2 \\ T_3 \\ T_4 \end{bmatrix} = \begin{bmatrix} g_D \\ \Delta x^2 f_1 \\ \Delta x^2 f_2 \\ \Delta x^2 f_3 \\ g_D \end{bmatrix}$$

As you can see, the **A** matrix is somewhat larger than it needs to be. This is the method that is followed in the 'Node numbering' section of this document.

1.2 Boundary conditions included in b

We can re-write the governing equations as:

$$\begin{aligned} T_2 - 2T_1 &= \Delta x^2 f_1 - T_0 \\ T_3 - 2T_2 + T_1 &= \Delta x^2 f_2 \\ -2T_3 + T_2 &= \Delta x^2 f_3 - T_4 \end{aligned}$$

Then we can, again, write these equations directly as a matrix as follows:

$$\mathbf{A} = \begin{bmatrix} -2 & 1 & 0 \\ 1 & -2 & 1 \\ 0 & 1 & -2 \end{bmatrix} \begin{bmatrix} T_1 \\ T_2 \\ T_3 \end{bmatrix} = \begin{bmatrix} \Delta x^2 f_1 - g_D \\ \Delta x^2 f_2 \\ \Delta x^2 f_3 - g_D \end{bmatrix}$$

This is the form you are asked to implement in the Problem Set.

2 Node Numbering

In our experience, confusion often arises on how the equations from a spatially discretized set of equations is transferred to a matrix. The confusion arises due to striking a similarity between the

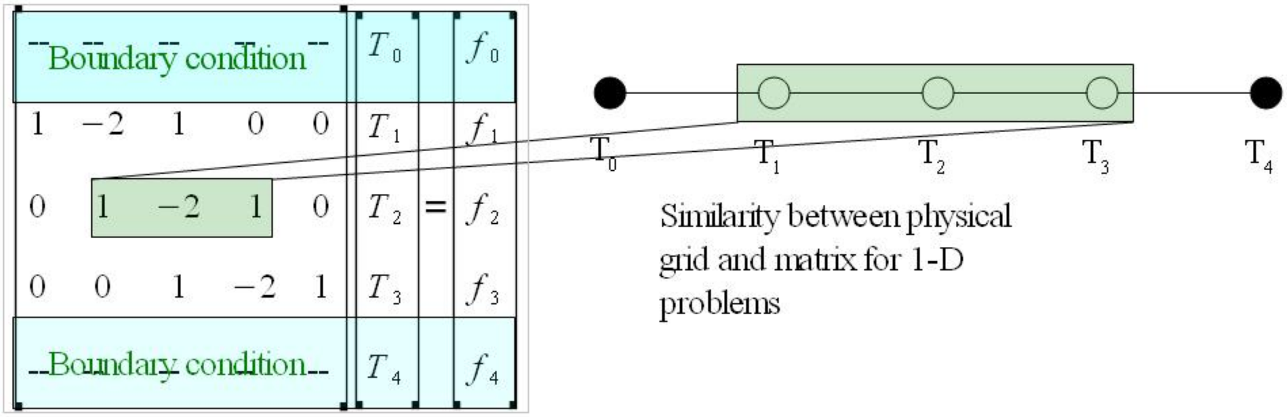

Boundary condition
1 −2 1 0 0
0 1 −2 1 0
0 0 1 −2 1
Boundary condition
T_0 T_1 T_2 T_3 T_4
=
f_0 f_1 f_2 f_3 f_4
T_0 T_1 T_2 T_3 T_4
Similarity between physical grid and matrix for 1-D problems

Figure 2: Similarity between matrix structure and computational cell for 1D discretization

numerical footprint and the location of coefficients in the matrix for one-dimensional problems. For this example, we will ignore the boundary conditions. Take the following 1 Dimensional grid (see Figure 1), with the governing equation $\frac{T_{i+1}-2T_i+T_{i-1}}{\Delta x^2} = f'_i$ for the interior.

We have a set of linear equations, one for each node in the interior. Note, up to this point we have not discussed any matrices! The matrices come from set of simultaneous equations, which are as follows:

$$T_2 - 2T_1 + T_0 = \Delta x^2 f'_1 = f_1$$
$$T_3 - 2T_2 + T_1 = \Delta x^2 f'_2 = f_2$$
$$T_4 - 2T_3 + T_2 = \Delta x^2 f'_3 = f_3$$

Referring to Figure 2, one can see the similarity between the physical problem, and the location of variables in the matrix. Unfortunately, this direct similarity stops at 1D problems. Now

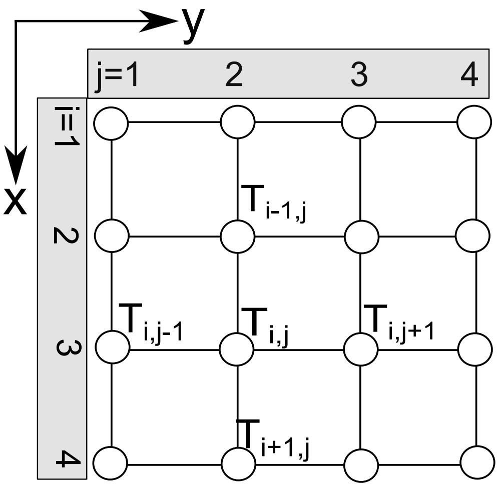


Figure 3: 2D Grid of interior node points

consider the 2D grid in Figure 3 with the following discretized governing equation:

$$\frac{T_{i+1,j} - 2T_{i,j} + T_{i-1,j}}{\Delta x^2} + \frac{T_{i,j+1} - 2T_{i,j} + T_{i,j-1}}{\Delta y^2} = f'_{i,j} \tag{1}$$

for the interior nodes. To simplify matters, let $\Delta x = \Delta y = h$, and the equation reduces to:

$$T_{i+1,j} + T_{i,j+1} - 4T_{i,j} + T_{i-1,j} + T_{i,j-1} = f'_{i,j}h^2 = f_{i,j} \tag{2}$$

Here we will have 16 equations for the interior, that is for $i = [1, 2, 3, 4]$, $j = [1, 2, 3, 4]$. At this point, some may ask: "Do I now need a three-dimensional **A** matrix because I have a matrix of unknowns instead of a vector of unknowns?" This is not the case because the solution matrix and the physical grid are NOT related. The matrix is only used to write the the simultaneous set of equations in an elegant form. In this problem we have 16 unknowns. The first step to constructing a matrix, now, is to label the nodes. The following describes a simple method for node numbering in MATLAB.

Step 1: Create a "`Node`" matrix for the interior nodes that resembles the physical grid.

```
Node=zeros(4,4)
```

$$\texttt{Node} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

The coordinates of this matrix now resembles the coordinates of the physical space. [Aside: if the problem was three-dimensional, you would create a 3-dimensional 'Node' matrix]

Step 2: Number the unknowns

```
Node(:)  = 1:16
```

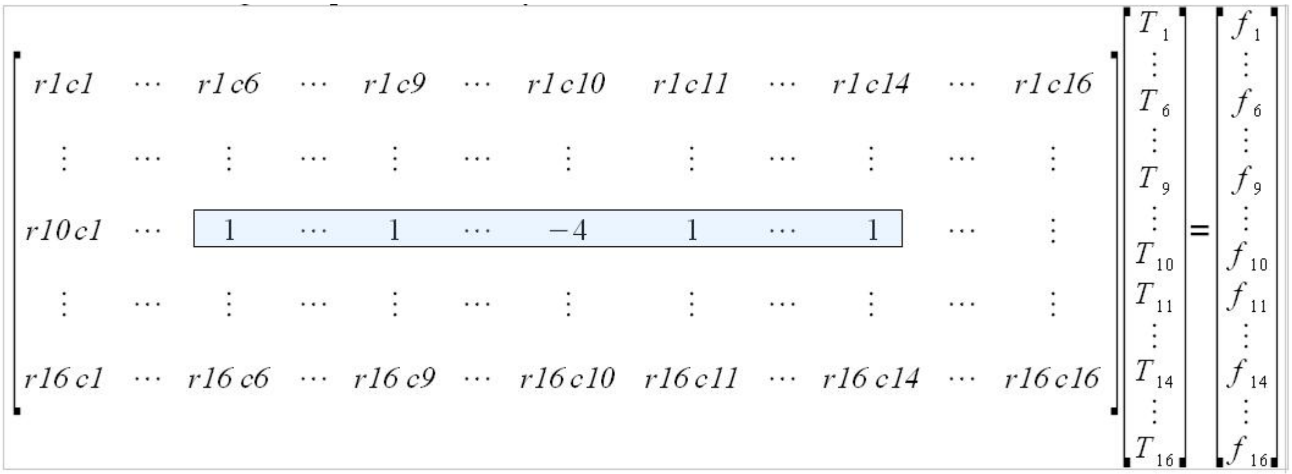
$$\texttt{Node} = \begin{bmatrix} 1 & 5 & 9 & 13 \\ 2 & 6 & 10 & 14 \\ 3 & 7 & 11 & 15 \\ 4 & 8 & 12 & 16 \end{bmatrix}$$

We could have numbered the nodes in a completely different way. There is no specific reason why we chose to number it this way. But now, each of the unknown temperature values has a number. A few examples: $T_{1,1} \sim \texttt{Node}(1,1) = 1$, $T_{2,3} \sim \texttt{Node}(2,3) = 10$, $T_{4,2} \sim \texttt{Node}(4,2) = 8$, $T_{2,4} \sim \texttt{Node}(2,4) = 14$. Look at these carefully, it is easy to confuse the rows and columns! Essentially, we have created a mapping $(i, j) \rightarrow \texttt{Node\#}$. So if we are trying to find the number associated with with coordinate ($i = 3$, $j = 2$) you can simply use: `Node(3,2)` (Note the way we have defined coordinates i and j, so that it matches with MATLAB's assignment of rows and columns).

Step 3: Construct the A-matrix.

Here we write the equation for $i = 2$, $j = 3$. This corresponds to node number 10.

$$\text{Node} = \begin{bmatrix} 1 & 5 & \mathbf{9} & 13 \\ 2 & \mathbf{6} & [\mathbf{10}] & \mathbf{14} \\ 3 & 7 & \mathbf{11} & 15 \\ 4 & 8 & 12 & 16 \end{bmatrix}$$



$$\begin{bmatrix} r1c1 & \cdots & r1c6 & \cdots & r1c9 & \cdots & r1c10 & r1c11 & \cdots & r1c14 & \cdots & r1c16 \\ \vdots & \cdots & \vdots & \cdots & \vdots & \cdots & \vdots & \vdots & \cdots & \vdots & \cdots & \vdots \\ r10c1 & \cdots & \boxed{1} & \cdots & 1 & \cdots & -4 & 1 & \cdots & 1 & \cdots & \vdots \\ \vdots & \cdots & \vdots & \cdots & \vdots & \cdots & \vdots & \vdots & \cdots & \vdots & \cdots & \vdots \\ r16c1 & \cdots & r16c6 & \cdots & r16c9 & \cdots & r16c10 & r16c11 & \cdots & r16c14 & \cdots & r16c16 \end{bmatrix} \begin{bmatrix} T_1 \\ \vdots \\ T_6 \\ \vdots \\ T_9 \\ \vdots \\ T_{10} \\ T_{11} \\ \vdots \\ T_{14} \\ \vdots \\ T_{16} \end{bmatrix} = \begin{bmatrix} f_1 \\ \vdots \\ f_6 \\ \vdots \\ f_9 \\ \vdots \\ f_{10} \\ f_{11} \\ \vdots \\ f_{14} \\ \vdots \\ f_{16} \end{bmatrix}$$

Figure 4: Constructing **A** matrix for 2D problem

Again, the `Node` matrix corresponds to the physical grid, the **A** matrix that we are about to create is related to the set of simultaneous equations. Now, if we're writing the equation for $i = 2$, $j = 3$, we will be in row 10 of the **A** matrix, as seen in Figure 4.

Note, here we labeled the rows and the columns (r# c#), instead of writing the whole matrix. Now, how does this work in code?

```
<for loop i=2 to i=Nx-1 >
  <for loop j=2 to j=Ny-1>
    Rowij=Node(i,j)
    A(Rowij, Rowij)=-4
    A(Rowij, Node(i,j-1)=1
    A(Rowij, Node(i,j+1)=1
    A(Rowij, Node(i-1,j)=1
    A(Rowij, Node(i+1,j)=1
  <end for i>
<end for j>
```

This is the same method used in `Heated-Plate-skel.m`, and you are welcome to use this code for the current Problem Set.

Final comments:

1. Consider what happens for an unstructured grid. How would you label the nodes for the triangles? In general, there is no elegant method that would result in a consistently structured matrix.

2. Mesh generators generally have to generate reference indexes similar to the `Node` matrix we created in this document.

3. Creating this mapping or `Node` matrix is unnecessary for a structured grid, and a simple functional relationship could be used (this is what we used in the introduction to this document which is sufficient for the Problem Set). In essence, the method presented here is half-way between a structured and unstructured grid approach.

4. After solving the equations, you will have a vector of unknowns. In order to plot the solution, the reverse mapping `Node#` $\rightarrow (i, j)$ will be required. Similar plotting issues exist for unstructured grids.
5. If any of this is still unclear, please come talk to us.