Introduction to Building Geometries into Arrays & Grid Setup

Outline

• Concept of Building Geometries into Arrays
• How MATLAB Indexes Arrays & Why
• Grid Concepts Setup
Suppose it is desired to simulate a wave scattering from a triangle. How is the triangle communicated to the numerical algorithm?
Geometrical Structures in Numerical Algorithms

Geometrical structures must be built into data arrays.

These Techniques are NOT Meant for Graphics!

This Topic teaches techniques that are NOT intended for generating graphics. Instead, the techniques in this lecture are intended to create arrays containing different shapes and geometries so that numerical computation can be performed onto those shapes and geometries.
How MATLAB Indexes Arrays & Why

Recall How Matrix Elements are Indexed

Recall from linear algebra matrix notation for indexing arrays.

\[
\begin{bmatrix}
a_{11} & a_{12} & a_{13} & a_{14} \\
a_{21} & a_{22} & a_{23} & a_{24} \\
a_{31} & a_{32} & a_{33} & a_{34} \\
a_{41} & a_{42} & a_{43} & a_{44} \\
\end{bmatrix}
\]

- Increasing values of \( m \)
- Increasing values of \( n \)

\( a_{mn} \) is the row number
\( n \) is the column number
MATLAB Treats Arrays at Matrices

Recall from linear algebra matrix notation for indexing arrays.

\[
\begin{bmatrix}
a_{11} & a_{12} & a_{13} & a_{14} \\
a_{21} & a_{22} & a_{23} & a_{24} \\
a_{31} & a_{32} & a_{33} & a_{34} \\
a_{41} & a_{42} & a_{43} & a_{44}
\end{bmatrix}
\]

\(a_{mn}\) is the row number
\(a_{mn}\) is the column number

In MATLAB, \(a_{mn}\) is indexed as \(A(m,n)\).

The Problem 😞

We like to think of arrays as \(f(i,j)\).

In MATLAB, \(a_{mn}\) is indexed as \(A(m,n)\).
The Solution in This Course

In this course, arrays will be treated in the more intuitive sense as

$f(i,j)$

Horizontal position in array starting at $i=1$.

Vertical position in array starting at $j=1$.

This will have to considered when arrays are visualized and creating meshgrids because MATLAB thinks it is working with matrices instead of arrays.

```matlab
imagesc(xa,ya,f.');
[Y,X] = meshgrid(ya,xa);
```

Grid Concepts & Setup
The Grid

The grid in the context of this course refers to the following set of information:

- Physical size of the grid ($S_x$ & $S_y$)
- Numerical size of the grid ($N_x$ & $N_y$)
- Grid resolution ($dx$ & $dy$)
- Grid axes ($x_a$ and $y_a$)
- Arrays containing data on the grid

Physical Size of the Grid ($S_x$ & $S_y$)

The physical size of the grid refers to how much physical space the grid represents.
Numerical Size of the Grid (\(N_x \& N_y\))

The *numerical size* of the grid refers to how many points are on the grid.

Note that \(N_x\) and \(N_y\) must be integers.

Grid Resolution (\(dx \& dy\))

Space is divided into an array of *grid cells*. 

Numerical Height \(N_y\) or \(N_y\) 

Numerical Width \(N_x\) or \(N_x\)
Space is divided into an array of grid cells.

**Cell Height**

Δ\(y\) or \(dy\)

**Cell Width**

Δ\(x\) or \(dx\)

It is preferred to setup the grid so that the grid cells are as square as possible.
Relation Between Grid Parameters

The grid parameters $S_x, N_x, dx, S_y, N_y$ and $dy$ must satisfy:

$$S_x = N_x \Delta x$$
$$S_y = N_y \Delta y$$

$$S_x = N_x \cdot dx$$
$$S_y = N_y \cdot dy$$

Typically $S_x, N_y, S_y$ and $N_y$ are specified at the start of a program and $dx$ and $dy$ are calculated from them later in the code.