

```

% constructGrid(m) : construct a grid of (m+1)x(m+1) cells and populate the
% related values of the grid based on the value of m.
%
% Output parameters:
%   grid = the output grid with all its populated fields
%
% Input parameters:
%   m = the size of the matrix of phi what will be contained in the grid.

function grid = constructGrid(m)

% Where does the grid start and end? This corresponds to the domain of
% omega, i.e (0,0) to (1,1).
grid.lowerLeftCorner = [0 0];
grid.upperRightCorner = [1 1];

% The size of the grid
grid.gridSize = m;

% Calculate dx (step in x direction), and dy (step in y direction)
grid.step = (grid.upperRightCorner - grid.lowerLeftCorner) ./ (grid.gridSize - 1);

% Assign the values along both the x-axis and y-axis. We're getting the
% transpose here so that we get a nice one-column vector instead of a one-row
% vector.
grid.cells = cell(2, 1);
grid.cells{1} = (grid.lowerLeftCorner(1) : grid.step(1) : grid.upperRightCorner(1))';
grid.cells{2} = (grid.lowerLeftCorner(2) : grid.step(2) : grid.upperRightCorner(2))';

% Now generate the x-axis and y-axis. The values in grid.axes will
% determine the values that will be used for the grid lines along both
% axes.
grid.axes = cell(2, 1);
[ grid.axes{:} ] = ndgrid(grid.cells{:});

```