```
function MeshData = generateRecMesh(dx1,dx2,x1_lim1,x1_lim2,x2_lim1,x2_lim2)
% $Author : Vignesh Ramakrishnan$
% $RIN : 662028006$  $Date : November 21, 2021$
% $Code Version: 1.0$
% Inputs: dx1 - discretization along x1-axis
%         dx2 - discretization along x2-axis
%         x1_lim1 - lower limit of x1 dimension of the domain to mesh
%         x1_lim2 - upper limit of x1 dimension of the domain to mesh
%         x2_lim1 - lower limit of x2 dimension of the domain to mesh
%         x2_lim2 - upper limit of x2 dimension of the domain to mesh
% Outputs: struct mesh
%          mesh.dim        - holds the dimension of the domain = 2
%          mesh.num_elem   - Number of rectangular elements present in the
%                            domain
%          mesh.num_node   - Number of nodal elements present in the domain
%          mesh.DOF        - 2D Matrix with each node holding its DOF value
%          mesh.CornerDOF  - 1D array holding the DOF values of domain
%                            corners
%          mesh.BoundaryDOF- 1D array holding the DOF values of domain
%                            boundary
%          mesh.GridFn     - 2D cell array with each cell holding the
%                            domain location of each nodal DOF
%          mesh.DimLen     - 1x2 array that holds total number of points
%                            along x1 and x2 direction
%          mesh.DX         - [dx1 dx2]: discretization along x1 and x2

    % Generate [X1 X2] - Values along which to generate rectangular mesh
    x1 = x1_lim1:dx1:x1_lim2;
    x2 = x2_lim1:dx2:x2_lim2;

    % dimension of domain
    dim = 2;

    Nnodes = length(x1)*length(x2);
    Nelem = (length(x1)-1)*(length(x2)-1);

    MeshDOF = zeros(length(x2),length(x1));
    GridFn = cell(length(x2),length(x1));
    k = 1;
    b_dof = 1;
    c_dof = 1;
    corner_dof = zeros(2*dim,1);
    per = 2*(length(x1)-1) + 2*(length(x2)-1);
    boundary_dof = zeros(per,1);
    for i=1:length(x2)
        for j=1:length(x1)
            MeshDOF(i,j) = k;
            GridFn{i,j} = [x1(j),x2(i)];
            if i==1 && j==1 % corner 1
                corner_dof(c_dof) = k;
                c_dof = c_dof + 1;
            elseif i==1 && j== length(x1) % corner 2
                corner_dof(c_dof) = k;
                c_dof = c_dof + 1;
            elseif i==length(x2) && j==1 % corner 3
                corner_dof(c_dof) = k;
                c_dof = c_dof + 1;
            elseif i== length(x2) && j==length(x1) % corner 4
                corner_dof(c_dof) = k;
```

```
                c_dof = c_dof + 1;
            end
            if i==1
                boundary_dof(b_dof) = k;
                b_dof = b_dof + 1;
            elseif j==1 || j==length(x1)
                boundary_dof(b_dof) = k;
                b_dof = b_dof + 1;
            elseif i == length(x2)
                boundary_dof(b_dof) = k;
                b_dof = b_dof + 1;
            end

            k = k + 1;
        end
    end
    MeshData.dim = dim;
    MeshData.num_elem = Nelem;
    MeshData.num_node = Nnodes;
    MeshData.DOF = MeshDOF;
    MeshData.CornerDOF = corner_dof;
    MeshData.BoundaryDOF = boundary_dof;
    MeshData.GridFn = GridFn;
    MeshData.DimLen = [length(x1) length(x2)];
    MeshData.DX = [dx1 dx2];
end
```

Published with MATLAB® R2021a