

```

%=====
%Copyright (c) 2018 by Georgia Tech Research Corporation.
%All rights reserved.
%
%The files contain code and data associated with the paper titled
%"A Deep Learning Approach to Estimate Stress Distribution: A Fast and
%Accurate Surrogate of Finite Element Analysis".
%
%The paper is authored by Liang Liang, Minliang Liu, Caitlin Martin,
%and Wei Sun, and published at Journal of The Royal Society Interface, 2018.
%
%The file list: ShapeData.mat, StressData.mat, DLStress.py, im2patch.m,
%UnsupervisedLearning.m, ReadMeshFromVTKFile.m, ReadPolygonMeshFromVTKFile.m,
%WritePolygonMeshAsVTKFile.m, Visualization.m, TemplateMesh3D.vtk, TemplateMesh2D.vtk.
%Note: *.m and *.py files were converted to pdf files for documentation purpose.
%
%THIS SOFTWARE IS PROVIDED ``AS IS'' AND WITHOUT ANY EXPRESS OR IMPLIED WARRANTIES,
%INCLUDING, WITHOUT LIMITATION, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS
%FOR A PARTICULAR PURPOSE.
%=====
%%
function Mesh = ReadMeshFromVTKFile (FilePathAndName)
%Mesh.Point
%Mesh.Element
Mesh.Point=[];
Mesh.Element={};

fid=fopen(FilePathAndName, 'r');
if fid == -1
    error('can not open vtk file')
    return
end

LineStr = fgets(fid); %# vtk DataFile Version 3.0
tempIndex = strfind(LineStr, '#');
if isempty(tempIndex)
    disp('Can not find the key char: #')
    fclose(fid);
end

LineStr = fgets(fid); %vtk output

LineStr = fgets(fid); %ASCII
tempIndex = strfind(LineStr, 'ASCII');
if isempty(tempIndex)
    disp('Can not find the key word: ASCII')
    fclose(fid);
end

LineStr = fgets(fid);
tempIndex = strfind(LineStr, 'DATASET');
if isempty(tempIndex)
    disp('Can not find the key word: DATASET')

```

```

    fclose(fid);
end

LineStr = fgets(fid);
tempIndex = strfind(LineStr, 'POINTS');
if isempty(tempIndex)
    disp('Can not find the key word: POINTS')
    fclose(fid);
end

PointCounter=0;
Mesh.Point=zeros(3,1000);%pre-allocate memory

while 1
    LineStr = fgets(fid);
    if ~ischar(LineStr)
        break;
    end
    temp = textscan(LineStr, '%f ');
    temp=temp{1}';

    if PointCounter+1 > length(Mesh.Point(1,:))
        tempPoint=zeros(3, PointCounter+1000);
        tempPoint(:,1:PointCounter)=Mesh.Point;
        Mesh.Point=tempPoint;
    end

    if length(temp) == 9
        Mesh.Point(:,PointCounter+1)=temp([1,2,3]);
        Mesh.Point(:,PointCounter+2)=temp([4,5,6]);
        Mesh.Point(:,PointCounter+3)=temp([7,8,9]);
        PointCounter=PointCounter+3;
    elseif length(temp) == 6
        Mesh.Point(:,PointCounter+1)=temp([1,2,3]);
        Mesh.Point(:,PointCounter+2)=temp([4,5,6]);
        PointCounter=PointCounter+2;
    elseif length(temp) == 3
        Mesh.Point(:,PointCounter+1)=temp([1,2,3]);
        PointCounter=PointCounter+1;
    else
        break;
    end
end
end

```

```

while ischar(LineStr)
    ....
    tempIndex = strfind(LineStr,'POLYGONS');
    if ~isempty(tempIndex)
        break;
    end
    ....
    tempIndex = strfind(LineStr,'CELLS');
    if ~isempty(tempIndex)
        break;
    end
    ....
    LineStr = fgets(fid);
end

ElementCounter=0;

while 1
    ....
    LineStr = fgets(fid);
    if ~ischar(LineStr)
        break;
    end
    ....
    temp = textscan(LineStr,'%f ');
    temp=temp{1}';
    ....
    if length(temp) <= 1
        break;
    end
    ....
    ElementCounter=ElementCounter+1;
    %temp(1) is PointCount in this Element
    Mesh.Element{ElementCounter}=temp(2:end) + 1; % "+1" change 0-index to 1_index
end

fclose(fid);

Mesh.Point=Mesh.Point(:,1:PointCounter);

```