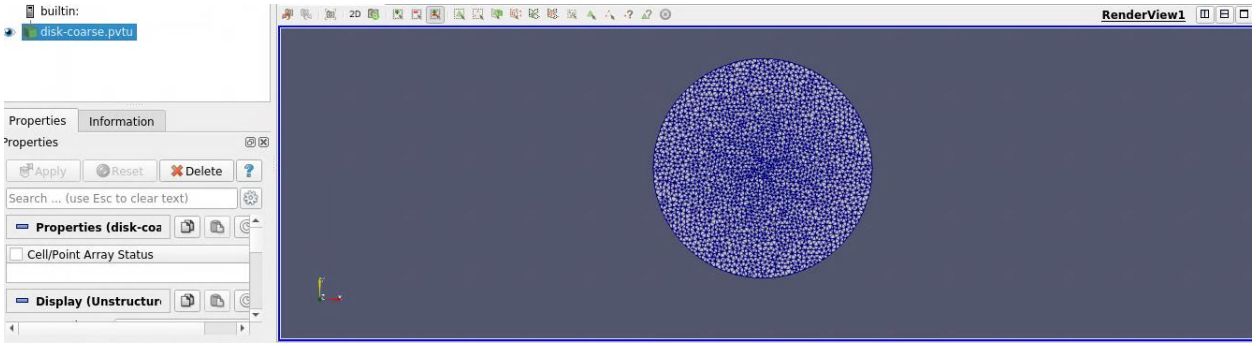
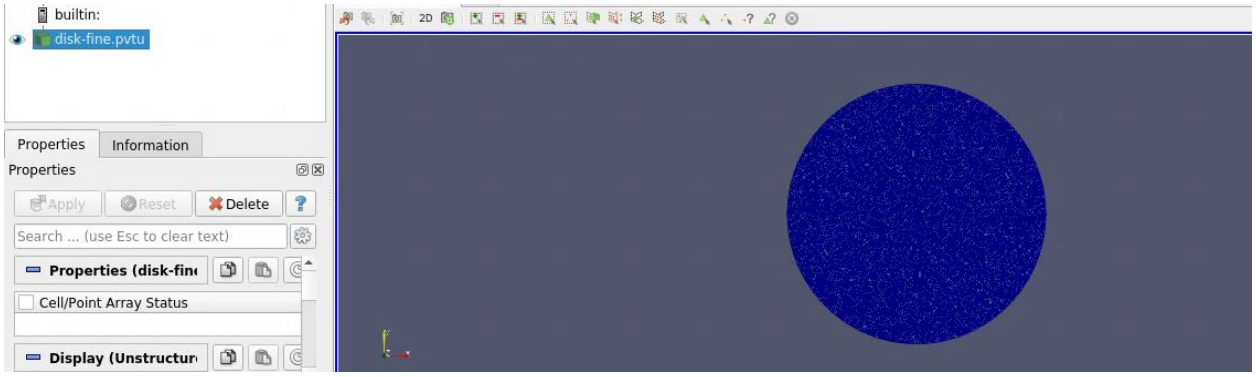
Homework 1 PCCM Report Rohan Krishna Balaji 403596

1) One Picture per Mesh with edge visible

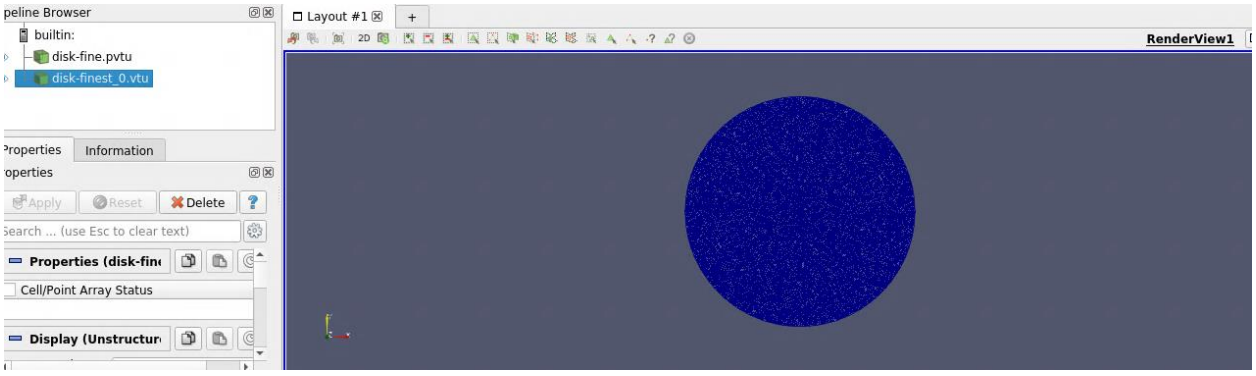
Coarse:



Fine:



Finest:



2) Complete Code in tri.ccp to obtain the Picture

```
#include "tri.h"
/*****************************************************************************
*************************
void preProcessor::prepareMesh()
*****************************************************************************
*************************
Does everything related to mesh generation and transfers between
processors.
*****************************************************************************
************************/
void triMesh::prepareMesh(inputSettings* settings)
{
    cout << endl << "====================== MESH ======================="
<< endl;

    readMeshFiles(settings);

    return;
}

/*****************************************************************************
*************************
void preProcessor::readMeshFiles()
*****************************************************************************
*************************
File read procedure :
1- Name of the file to be opened is retrieved from the inputSetting obj.
2- File is opened in appropriate format, this is ascii format for minf or
   other text files and binary format for binary mesh files.
3- Read operation for minf file is straight forward. Binary files are read
   as size of a double or int and stored in readStream. Then swapbytes
   function is called to swap the bytes for the correct endianness.
4- Finally obtained data is deep-copied to the mesh data structure.
*****************************************************************************
************************/
void triMesh::readMeshFiles(inputSettings* settings)
{
    ifstream file;        // file name object for serial file read
    string   dummy;       // dummy string to hold file names etc.
    char*    readStream;  // temperory var used for strings read from
files
    double   dummyDouble; // temperory var used for double values read
from files


/*****************************************************************************
*********************
    * READ THE MINF FILE
    * This file should hold the number of elements, nodes, space
dimensions, element nodes and
    * element faces.
```

```
**************************************************************************
*********************/
    // Add code here
       dummy= settings->getMinfFile();
       file.open(dummy.c_str(),ios::in);
       file.seekg(0,ios::beg);
       if(!file.good())
       {
       cerr << "Can't Open Minf" <<file <<endl;
       exit(EXIT_FAILURE);
       }
       readStream = new char[20];
       file >> readStream;
       file >> readStream;
       nn = atoi(readStream);
       file >> readStream;
       file >> readStream;
       ne = atoi(readStream);

       cout<<"nn=" <<nn<<","<< "ne="<<ne <<"nsd="<<","<<"nen="<<nen<<endl;

       delete [] readStream;
       file.close();
    //Allocation of memory for the mesh data structure
    xyz = new double [nn*nsd];
    node = new triNode[nn];
    elem = new triElement[ne];


/**************************************************************************
**********************
    * READ THE MXYZ FILE
    * This file contains the node coordinates

**************************************************************************
*********************/
    // Add code here
       dummy = settings->getMxyzFile();
       file.open(dummy.c_str(),ios::in|ios::binary|ios::ate);
       file.seekg(0,ios::beg);
       if(!file.good())
       {
       cerr<<"can't open Mxyz"<< file <<endl;
       exit(EXIT_FAILURE);
       }
       readStream = new char [nsd * sizeof(double)];
       int i;
       for (i=0; i< nn; i++)
       {
       file.read(readStream, nsd * sizeof(double));
       swapBytes(readStream, nsd, sizeof(double));

       xyz = (double*)readStream;
```

```
        dummyDouble= *xyz;
        node[i].setX(dummyDouble);
        xyz = xyz+1;

        xyz= (double*)(readStream+sizeof(double));
        dummyDouble = *xyz;
        node[i].setY(dummyDouble);
        xyz=xyz+1;

        if(i<10)
        cout<<i<<"Node:"<<node[i].getX()<<","<<node[i].getY()<<endl;
        }
        file.close();

/**************************************************************************
**********************
    * READ THE MIEN FILE
    * This file contains the element connectivity

**************************************************************************
**********************/
    // Add code here
        dummy = settings -> getMienFile();
        file.open(dummy.c_str(), ios::in| ios::binary|ios::ate);
        file.seekg(0,ios::beg);
        if(!file.good())
        {
        cerr<<"cant open mien"<<file<<endl;
        exit(EXIT_FAILURE);
        }
        int j;
        int k;
        int *eleconn;
        int dummyInt;
        readStream = new char[nen * sizeof(int)];
        eleconn = new int[nen*ne];
        for(j=0; j<ne ;j++)
        {
        file.read(readStream, nen*sizeof(int));
        swapBytes(readStream, nen,sizeof(int));

        for (k=0; k<nen; k++)
        {
        eleconn = (int*)(readStream+sizeof(int)*k);
        dummyInt = *eleconn;
        elem[j].setConn(k, dummyInt-1);
        eleconn = eleconn+1;
        }
        }
        file.close();
    return;
}

void triMesh::swapBytes (char *array, int nelem, int elsize)
```

```
{
    register int sizet, sizem, i, j;
    char *bytea, *byteb;
    sizet = elsize;
    sizem = sizet - 1;
    bytea = new char [sizet];
    byteb = new char [sizet];
    for (i = 0; i < nelem; i++)
    {
        memcpy((void *)bytea, (void *)(array+i*sizet), sizet);
        for (j = 0; j < sizet; j++)
            byteb[j] = bytea[sizem - j];
        memcpy((void *)(array+i*sizet), (void *)byteb, sizet);
    }
    free(bytea);
    free(byteb);

    return;
}
```

3) Answer to the Question:

Not sure