Attachment 1 Python application for mesh refinement, source code

```
#Date: 2017-02-03
#Author: Jiri Galuska
#Desctiption: Boolean intersection of two cellSets
#Usage: the script must be in main direktory. Cell sets must be in
constant/polyMesh/sets/. Define appropriet name for cellSets.
#

#!/usr/bin/python

import glob
import numpy
import os
import os.path as op
import math


#************************USER-MODIFICATION-AREA*****************
#Global variables

firstSetName = "refinement"
secondSetName= "water_surface"
relPath = "constant/polyMesh/sets/"

#************************END-OF-USER-MODIFICATION-AREA*********



def readHeader(fileName):
    """Extract header from file"""
    startOfData = False
    endOfData = False
    fileRaw = open(op.join(relPath, fileName), "r")

    header = []     #initialize list, store header of the file

    for line in fileRaw:

        if line.count("(") >= 1:
            startOfData = True

        if startOfData == False:
 #It is a header, do not print the number of etries before )
            line = line.replace('\n','')        #Remove all empty
lines
            if line.rstrip().isdigit() == True:          #Here the
number of entries starts => exit loop
                break

            header.append(line)

    return header     ##Creating a tupple

def readData(fileName):
    """Extract the values"""
    startOfData = False
    fileRaw = open(op.join(relPath, fileName), "r")

    header = []     #initialize list, store header of the file
    dataBody = []   #initialize list, store data values
```

```
    for line in fileRaw:
        if line.count("(") >= 1:
            startOfData = True
            continue          #Do not print the parenthesis a start
again

        if line.count(")") >= 1:
            break

        if startOfData == True:
            tmp = line.rstrip()
            dataBody.append(int(tmp))

    return dataBody    ##Creating a tupple

def dataIntersection(firstSet, secondSet):
    intersection = []
    for lineFirst in firstSet:
        for lineSecond in secondSet:
            if lineFirst == lineSecond:
                intersection.append(lineFirst)
                break

    return intersection

def countOccurencies(list):
    numberOfEntries = 0
    for i in list:
        numberOfEntries +=1
    return numberOfEntries


firstSet = readData(firstSetName)
secondSet = readData(secondSetName)

intersection = dataIntersection(firstSet,secondSet)
header = readHeader(firstSetName)

#Create the output
with open(op.join(relPath, "vortexRefSet"), "w") as fileToWrite:
    for line in header:
        print >>fileToWrite, line

    print >> fileToWrite, countOccurencies(intersection)
    print >> fileToWrite, "("
    for line in intersection:
        print >>fileToWrite, line

    print >> fileToWrite, ")"
    print >> fileToWrite, "\n"
    print >> fileToWrite, "//
**********************************************************************
*** //"



#print firstSet
```

```
#returnMatches(a, b)
```