

S1. Python code for generating radiating lines

The code below was used for generating radiating lines in this study. The variables in **bold** and *italic* require users' inputs.

```
import arcpy.  
from math import radians, sin, cos  
output = "FileName.shp" # Put the name for the file of radiating lines  
# Type the full_path in the first inputs  
arcpy.CreateFeatureclass_management("full_path", output, "Polyline")  
# Type the full path\\File_name.shp here.  
cur = arcpy.InsertCursor("full_path\\FileName.shp")  
# Type starting ( $\Theta$ ) and ending angle ( $\Phi$ ) of the radiating lines in the first two inputs  
# Type radiating lines intervals ( $\Delta$ ) in the third input  
angle_range = range( $\Theta$ ,  $\Phi$ ,  $\Delta$ )  
# Type the center point coordinate systems (center_x, center_y)  
origin_x, origin_y = (center_x, center_y)  
# Type the radius in meter (r)  
distance = r  
# Type the degree of half interval here  
angle =  $\Delta/2$   
for angle in angle_range:  
    # start point  
    lineArray = arcpy.Array()  
    start = arcpy.Point()  
    (start.ID, start.X, start.Y) = (1, origin_x, origin_y)  
    lineArray.add(start)  
    # end point  
    (disp_x, disp_y) = (distance * sin(radians(angle)), distance * cos(radians(angle)))  
    (end_x, end_y) = (origin_x + disp_x, origin_y + disp_y)  
    end = arcpy.Point()  
    (end.ID, end.X, end.Y) = (2, end_x, end_y)  
    lineArray.add(end)  
    feat = cur.newRow()  
    feat.shape = lineArray  
    cur.insertRow(feat)
```

```
del feat, lineArray
else:
    print '%d' %(angle)
del cur
```