Supplement of Earth Surf. Dynam., 4, 773–780, 2016
http://www.earth-surf-dynam.net/4/773/2016/
doi:10.5194/esurf-4-773-2016-supplement

Earth Surface Dynamics
Open Access
EGU

Supplement of

Long-term coastal openness variation and its impact on sediment grain-size distribution: a case study from the Baltic Sea

Wenxin Ning et al.

Correspondence to: Wenxin Ning (wenxin.ning@geol.lu.se)

S1. Python code for generating radiating lines

The code below was used for generating radial 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 (Θ) and ending angle (Φ) of the radiating lines in the first two inputs
# Type radiating lines intervals (Δ) in the third input
angle_range = range(Θ,Φ,Δ)
# 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 = Δ/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
```