The '**kp_to_point**' Pyhton module provides a tool for calculation of scaled KP (Kilometer Post) of points relative to reference line, containing scaled KP data.
Points shall be defined as array in grid coordinates (Eastings and Northings).
Reference line shall be defined as broken line (polyline) array in grid coordinates (Eastings and Northings) and contain scaled KP for each vertex.

The module can be imported into Python application:

***import kp_to_point***

Once imported, the function '***go***' shall be called with two required arguments and one optional argument:

***variablename = kp_to_point.go(refarr, linarr, maxoff)***

or

***variablename = kp_to_point.go(refarr, linarr)***

Where arguments are:

***refarr: numpy array, required***

Reference line array containing vertexes coordinates and KP:
***[[easting, northing, kp],***
***…,***
***[easting, northing, kp]]***

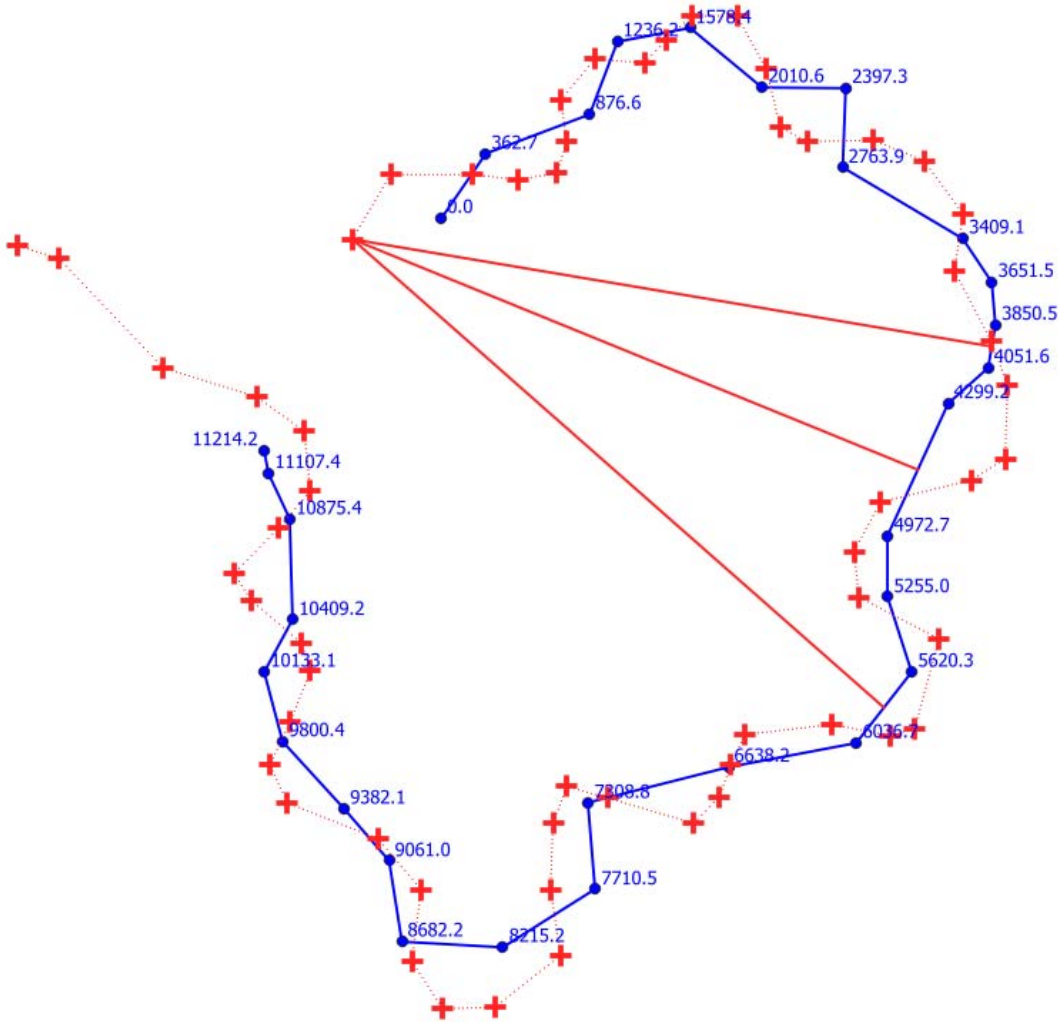
***linarr: numpy array, required***

Points array containing coordinates (line may be represented as array of points here)
***[[easting, northing],***
***…,***
***[easting, northing]]***

***maxoff: int of float, optional***

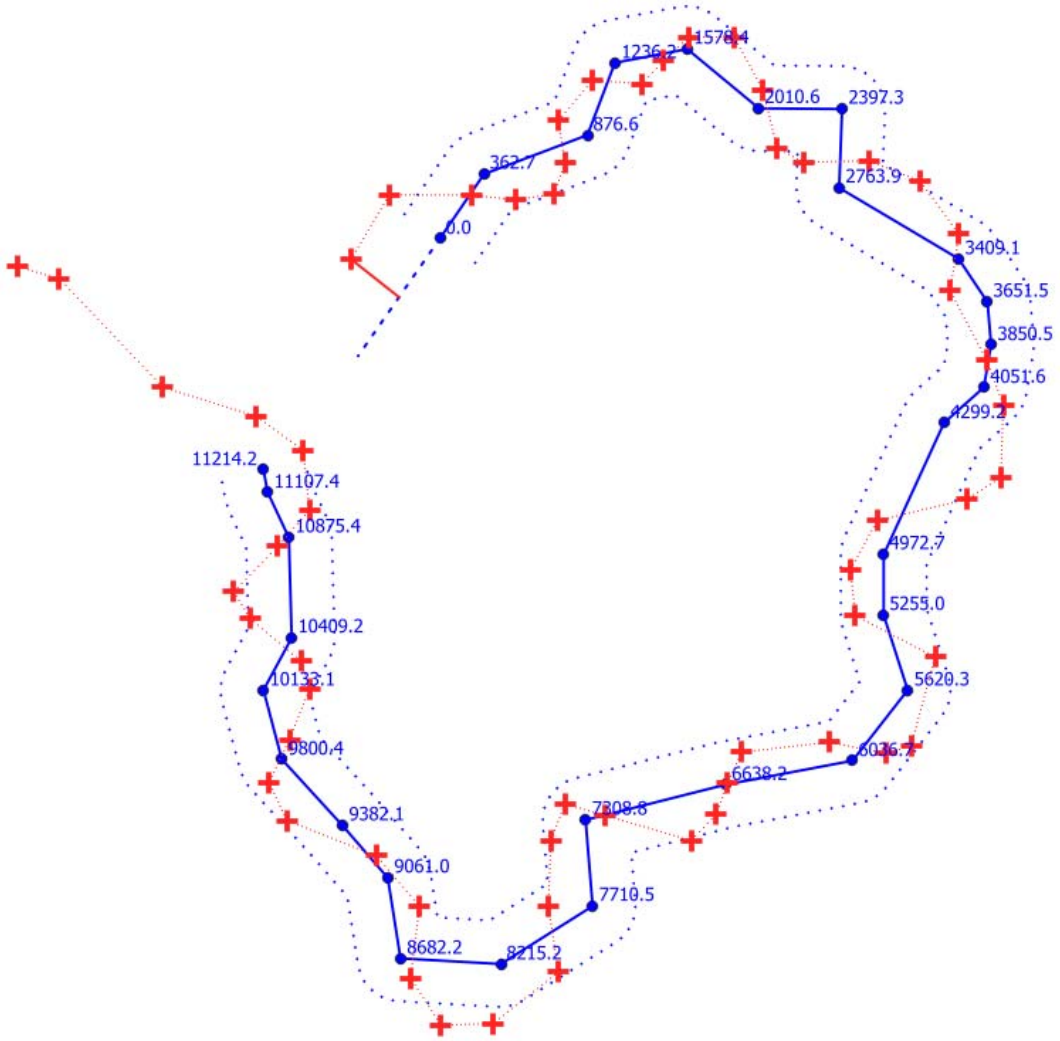
This argument specifies maximum distance of sought point from line segment where it is considered as belonging to this segment. This argument may be omitted in the majority of real survey cases. It may be used in case when reference line is closing and point may potentially fall into several segments (perpendicular line from point to reference line segment crosses the segment between ends).
See examples below.

If the argument is omitted, the point may be considered as falling into several segments (where perpendicular line from point to reference line segment crosses the segment). The closest segment will be used and point KP will be calculated based on segment start / end scaled KP.
In the figure below, the first point may fall into three segments. Point KP is calculated based on the closest segment (i.e. middle one).



If the argument is specified, the points inside the ***maxoff*** corridor will be considered as falling into closest segment (where perpendicular line from point to reference line segment crosses the segment). Points outside the ***maxoff*** corridor will be considered as falling into closest segment regardless it crosses segment or its extension

In the figure below, the first point is outside the corridor and falls into extension of the first segment . Point KP will be calculated based on the extension of the closest segment (is negative).




The function returns numpy array of points containing scaled KP:

***[[easting, northing, kp],***

***…,***

***[easting, northing, kp]]***