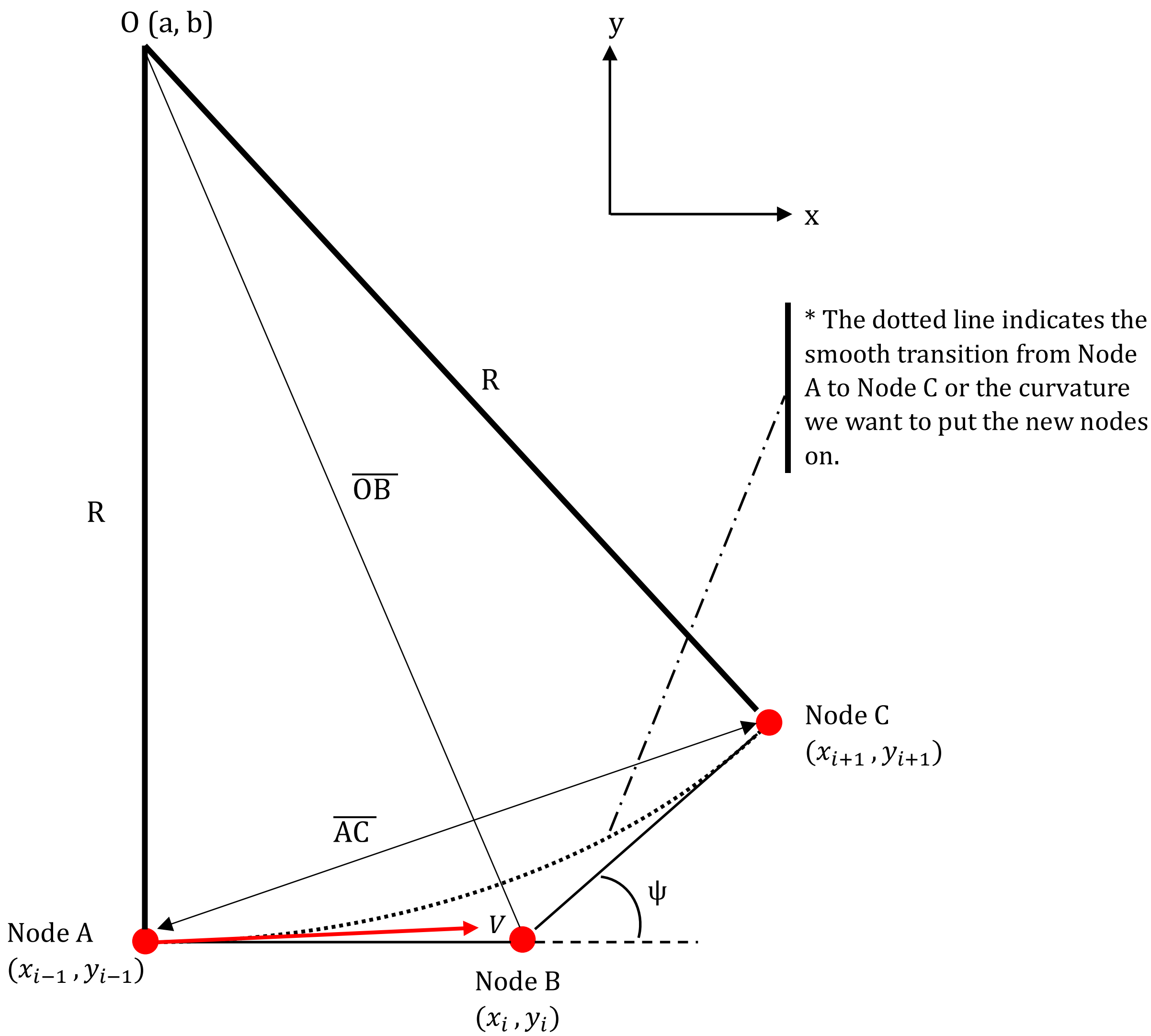
Mathematical Interpretation of Path Smoothing using Curvature

When we have three consecutive nodes A, B, and C inside the node path list with the format [[node],...], we will smooth the path $\overline{ABC}$ so that it becomes the circumference of the curvature from Node A to Node C as depicted in the figure below.



In order to find the unknown variables, R, a (the center x-position of the curvature), and b (the center y-position of the curvature) we use the following relations to come up with 3 equations to solve 3 unknown variables.

$$R^2 = (x_{i-1} - a)^2 + (y_{i-1} - b)^2 = (x_{i+1} - a)^2 + (y_{i+1} - b)^2 \tag{1}$$

$$\overline{AC} \cdot \overline{OB} = |\overline{AC}||\overline{OB}| cos\frac{\pi}{2} = 0 \tag{2}$$

By solving this nonlinear equation (perhaps by using Scipy function fsolve) we can obtain the values of R, a, and b. Then, from the polar representation of the curvature we can describe the (x, y) values on the curvature.

$$x = R\cos\theta + a \tag{3}$$

$$y = R\sin\theta + b \tag{4}$$

From this relation, we can compute the angles at Node A and Node C to generate points for the curvature only between the two nodes.

$$\theta_{i-1} = \arccos\left(\frac{x_{i-1} - a}{R}\right) \tag{5}$$

$$\theta_{i+1} = \arccos\left(\frac{x_{i+1} - a}{R}\right) \tag{6}$$

Now, we can generate points representing θ. For example,

```
# Assume numpy is already imported
theta = np.linspace(theta1, theta2, 20)
```

We are also able to compute the radius R from equation (1). Then, we can obtain the points for (x, y) on the curvature by plugging in the θs into equations (3) and (4). Finally, insert these points into a new node path.