Robot Motion Planning:
An Efficient Algorithm for Line-Polygon Intersection

Jin Seob Kim, Ph.D.
Senior Lecturer, ME Dept., LCSR, JHU

In this note, an efficient algorithm to compute the intersection of a line and a convex polygon is presented, which will be useful in the motion planning of a point robot. This is from [1] where you can find more details. Here the pseudocode is only shown.

Pseudocode

Input: a 2D segment S from point $\mathbf{p}_0$ to point $\mathbf{p}_1$, and a 2D convex polygon Q with n vertices $\{\mathbf{q}_0, \mathbf{q}_1, \cdots, \mathbf{q}_{n-1}\}$.

Output: b is true if intersection, otherwise, b is false.

function b = isintersect_linepolygon (S, Q)
  Expand Q so that the end point is the first point: $Q = \{\mathbf{q}_0, \mathbf{q}_1, \cdots, \mathbf{q}_{n-1}, \mathbf{q}_n = \mathbf{q}_0\}$.
  if $\mathbf{p}_0 = \mathbf{p}_1$ (i.e., S is a single point) **then**:
    Use inpolygon to check whether the point is in/on or outside of the polygon.
  else
    initialize as $t_E = 0$ and $t_L = 1$
    $d\mathbf{s} = \mathbf{p}_1 - \mathbf{p}_0$.
    for each edge $e_i = \mathbf{q}_i\mathbf{q}_{i+1}$, $i = 0, 1, \cdots, n-1$ **do**:
      Compute the outward normal vector $\mathbf{n}_i$ of the edge e_i.
      $N = -(\mathbf{p}_0 - \mathbf{q}_i) \cdot \mathbf{n}_i$
      $D = d\mathbf{s} \cdot \mathbf{n}_i$
      if $D = 0$ **then**:
        if $N < 0$ **then**:
          Return: b is false
        end if
      end if
      $t = N/D$
      if $D < 0$ **then**:
        $t_E \leftarrow \max(t_E, t)$
        if $t_E > t_L$ **then**:
          Return: b is false
        end if
      else if $D > 0$ **then**:
        $t_L \leftarrow \min(t_L, t)$
        if $t_L < t_E$ **then**:
          Return: b is false

```
            end if
        end if
    end for
    if t_E ≤ t_L then:
        Return: b is true
    else
        Return: b is false
    end if
end if
```

References

[1] http://geomalgorithms.com/a13-_intersect-4.html.