

```
import numpy as np
from sympy import sympify
```

```
class LinearSpline():
```

```
    def expandParams(self,X, Y):
```

```
        n = len(X)
        new_X = np.zeros(n)
        new_Y = np.zeros(n)
```

```
        for i in range(n):
```

```
            new_X[i] = X[i]
            new_Y[i] = Y[i]
```

```
        points = np.array((new_X, new_Y)).T
```

```
        return points
```

```
    def compute(self, params):
```

```
        X = params["X"]
```

```
        Y = params["Y"]
```

```
        points = self.expandParams(X, Y)
```

```
        points = np.array(points)
```

```
        n = len(points)
```

```
        splines = []
```

```
        for i in range(n-1):
```

```
            segment = self.computeSegment(points[i], points[i+1])
```

```
            splines.append(segment)
```

```
        function = []
```

```
        for i in range(n-1):
```

```
            function.append([splines[i], "{x0} <= x <= {x1}".format(
                x0=points[i][0], x1=points[i+1][0])])
```

```
        return function
```

```
    def computeSegment(self, point0, point1):
```

```
        segment = "{fx1} + ({fx1} - {fx0})/({x1} - {x0})*(x - {x1})".format(
```

```
            fx1=point1[1],
```

```
            fx0=point0[1],
```

```
            x1=point1[0],
```

```
            x0=point0[0])
```

```
        return str(sympify(segment))
```

```
points = {"X":[0,1,2,3], "Y":[0,1,1,0]}
```

```
x = LinearSpline()
```

```
print(x.compute(points))
```