```
import numpy as np
import sympy as sp


def CubicSpline(xi, yi):
    n = len(xi)
    # h values
    h = np.zeros([n-1])
    for j in range(0, n-1, 1):
        h[j] = xi[j+1]-xi[j]
    # Equations system
    A = np.zeros([n-2, n-2])
    B = np.zeros([n-2])
    S = np.zeros([n])
    A[0, 0] = 2*(h[0]+h[1])
    A[0, 1] = h[1]
    B[0] = 6*((yi[2]-yi[1])/h[1] - (yi[1]-yi[0])/h[0])
    for i in range(1, n-3, 1):
        A[i, i-1] = h[i]
        A[i, i] = 2*(h[i]+h[i+1])
        A[i, i+1] = h[i+1]
        B[i] = 6*((yi[i+2]-yi[i+1])/h[i+1] - (yi[i+1]-yi[i])/h[i])
    A[n-3, n-4] = h[n-3]
    A[n-3, n-3] = 2*(h[n-3]+h[n-2])
    B[n-3] = 6*((yi[n-1]-yi[n-2])/h[n-2] - (yi[n-2]-yi[n-3])/h[n-3])
    # Solve equations system
    r = np.linalg.solve(A, B)
    print("r: ", r)
    # S
    for j in range(1, n-1, 1):
        S[j] = r[j-1]
    S[0] = 0
    S[n-1] = 0

    # coefficients
    a = np.zeros([n-1])
    b = np.zeros([n-1])
    c = np.zeros([n-1])
    d = np.zeros([n-1])
    for j in range(0, n-1, 1):
        a[j] = (S[j+1]-S[j])/(6*h[j])
        b[j] = S[j]/2
        c[j] = (yi[j+1]-yi[j])/h[j] - (2*h[j]*S[j]+h[j]*S[j+1])/6
        d[j] = yi[j]
    # Polynomial spline
    x = sp.Symbol('x')
    polynomial = []
    for j in range(0, n-1, 1):
        pSection = a[j]*(x-xi[j])**3 + b[j]*(x-xi[j])**2 + c[j]*(x-xi[j]) + d[j]
        pSection = pSection.expand()
        polynomial.append(pSection)
    return(polynomial)

# Data inputs
xi = [0, 1, 2, 3]
fi = [0, 1, 1, 0]
resolution = 10

# Process
n = len(xi)
# Get the polynomial sections
polynomial = CubicSpline(xi, fi)

# Points of each section
xSpline = np.array([])
ySpline = np.array([])
```

```
section = 1
while not(section >= n):
    pSection = polynomial[section-1]
    pxSection = sp.lambdify('x', pSection)

    a = xi[section-1]
    b = xi[section]
    xSection = np.linspace(a, b, resolution)
    ySection = pxSection(xSection)

    xSpline = np.concatenate((xSpline, xSection))
    ySpline = np.concatenate((ySpline, ySection))
    section = section + 1

# SALIDA
print('Polinomyal sections: ')
for section in range(1, n, 1):
    print(' x = ['+str(xi[section-1])
        + ','+str(xi[section])+']')
    print(str(polynomial[section-1]))
```