2024/11/16

Final Project: 图像插值拟合(用python实现)

# 插值算法

本项目插值算法的核心是cubic spline, 即三次样条插值， 这种插值方法精度较高， 严格过给定点， 可以指定两端的导数，灵活性较高

## CubicSpline方法及其改进

### 本方法来自于scipy.interpolate， <https://docs.scipy.org/doc/scipy/reference/generated/scipy.interpolate.CubicSpline.html> (official manual) ***class*CubicSpline(*x*, *y*, *axis****=0***, *bc\_type****='not-a-knot'***, *extrapolate****=None***)** **x, y 为numpy数组， 其中x必须严格增大 bc\_type可以指定两端点的导数值， extraplote在本项目中没有使用**

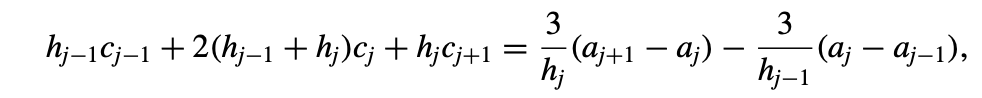
### 如何规避CubicSpline中对x严格增大的要求而不影响pyplot： **在class interResults中，我们假设x单调，可以是单调减，新建一个dataSorted,进行插值，然后将原数据进行代入求点**

### **X不单调的情况下，我们进行极坐标变换 注意到interResults的构造函数中的circle参数，置1时采用极坐标变换。 通过调用xy2circle(position) circle2xy(positon)方法，我们很容易就能进行极坐标变换。 借助极坐标变换，只要theta单调，我们就可以调用CubicSpline方法**

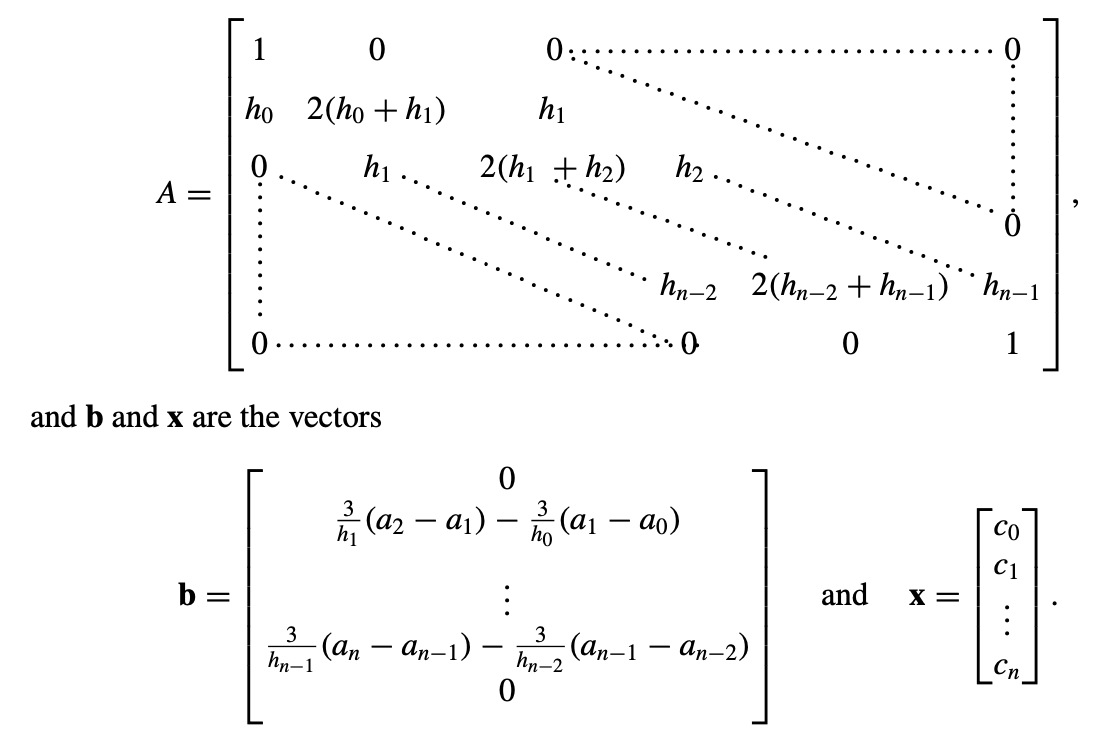
# cubic Spline的 数学依据

*Sj* (*x*)= *aj* + *bj* (*x*− *xj* ) + *cj* (*x*− *xj* )2 + *dj* (*x*− *xj* )3

依据数学推导，可以得到：

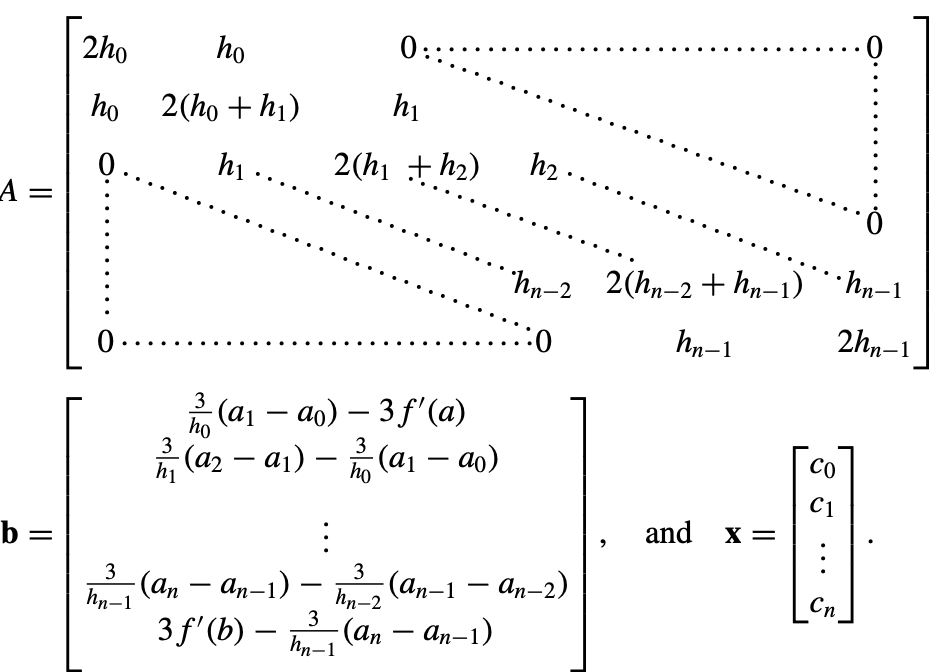


此时c为唯一的未知量， 因此我们可以构建矩阵：



这是natural spline的情况：

而对于clamped spline:



依据a, b, c, d之间的关系，我们可以求得各个composite spline

# 拟合结果 i. 结果图片

共取点28个

# 算法消耗结果：

i. samples 取点:  
samples = np.array([[1, 3], [2, 3.72], [4.53, 4], [6, 4.5],[7.5, 6.55], [11, 8.5], [14, 7.5], [17, 5], [19, 7.45], [20, 8], [22, 8.43], [23, 7.95], [24.7, 5.9], [26, 6.1] ,[28.0, 4.55], [30.5, 3.0], [29, 1.55], [28, 1.55], [24.5, 1.12], [18, 1.9], [15.5, 1.75], [13.5, 2.15], [10, 1.5], [9.2, 2], [9.4, 3], [9.3, 3.8], [8, 2.55], [5, 2.6], [2, 2.73]])   
  
共取点28个  
  
ii. 构建CubicSpline个数：  
functions += [interResults(samples[0:2], diffs[0]), interResults(samples[1:4],diffs[1]), interResults(samples[3:8], diffs[2]), interResults(samples[7:11], diffs[3]), interResults(samples[10:15], diffs[4]), interResults(samples[14:17], diffs[5], circle=1),interResults(samples[16:20], diffs[6]), interResults(samples[19:23], diffs[7]), interResults(samples[22:26], diffs[8], circle=1), interResults(samples[25: 28], diffs[9]), interResults(samples[27: 30], diffs[10])]

共11条