

# OPEN CASCADE BSpline Curve Interpolation

[eryar@163.com](mailto:eryar@163.com)

**Abstract.** Global curve interpolation to point data is a way to construct curves. The paper focus on the interpolate algorithm in OPEN CASCADE, and give a simple example to demonstrate the usage of the `GeomAPI_Interpolate` class.

**Key Words.** Interpolate, NURBS, BSpline, OPEN CASCADE

## 1.Introduction

曲线曲面拟合 Curve and Surface Fitting 的方式可以分为两类: 插值 interpolation 和逼近 approximation。采用插值的方式时, 所创建的曲线或曲面必须精确地满足所给的数据条件, 例如曲线通过所给的插值点。采用逼近的方式时, 创建的曲线或曲面不必精确地满足所给的数据条件, 只要在一定的误差范围内接近即可, 如下图所示:

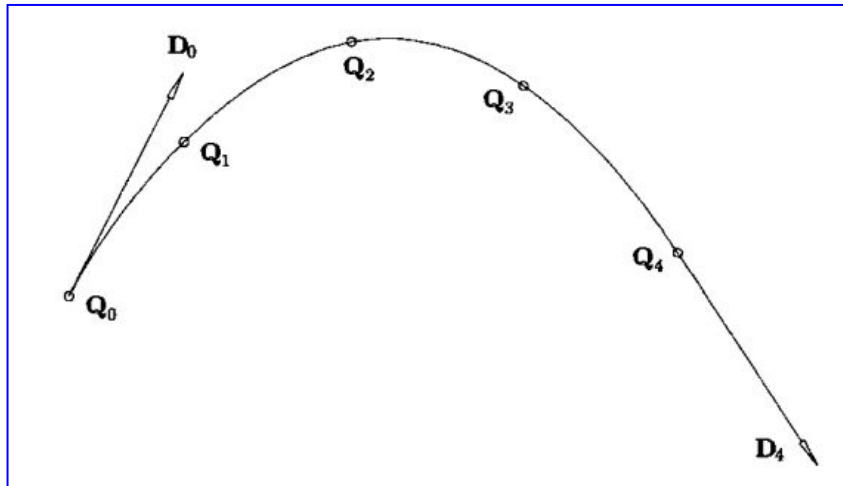


Figure 1.1 A curve interpolating five points and end derivatives(The NURBS Book)

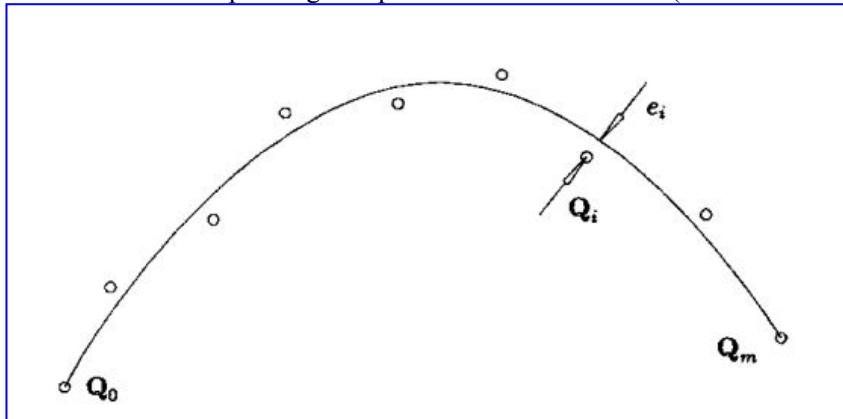


Figure 1.2 A curve approximating points(The NURBS Book)

本文先简要介绍 B 样条插值的原理, 再结合 OPEN CASCADE 源码说明如何对给定点插值 B 样条曲线及 OPEN CASCADE 中插值曲线的一些注意事项。

## 2. Global Interpolation

假设给定一组数据点  $\{Q_k\}$ ,  $k=0,1,\dots,n$ , 我们想要用一条  $p$  次非有理 B 样条曲线插值于这些点。如果我们为每个点  $Q_k$  指定了一个参数值  $u_k$ , 并且选定了合适的节点矢量  $U$ , 我们就可以建立一个系数矩阵为  $(n+1) \times (n+1)$  的线性方程组:

$$Q_k = C(u_k) = \sum_{i=0}^n N_{i,p}(u_k) P_i$$

$n+1$  个控制点  $P_i$  是未知量。剩下的问题是如何确定  $Q_k$  对应的参数值  $u_k$  及节点矢量  $U$ , 这将影响到曲线的形状和参数化。常见的选取  $u_k$  的方法有均匀参数化法、弦长参数化法和向心参数化法。

### 2.1 Equally Spaced 均匀参数法

假设参数限定在  $[0, 1]$  范围内, 那么

$$\bar{u}_0 = 0, \bar{u}_n = 1, \bar{u}_k = \frac{k}{n}, k = 1, 2, \dots, n-1$$

当参数范围为  $[a, b]$  时,

$$\begin{cases} \bar{u}_0 = a \\ \bar{u}_k = a + k \frac{b-a}{n} & 1 \leq k \leq n-1 \\ \bar{u}_n = b \end{cases}$$

均匀参数化法是最简单的构造参数的方法, 但是不推荐采用这种方法, 因为当数据点分布步均匀时, 会产生很奇怪的形状, 如打圈自交。

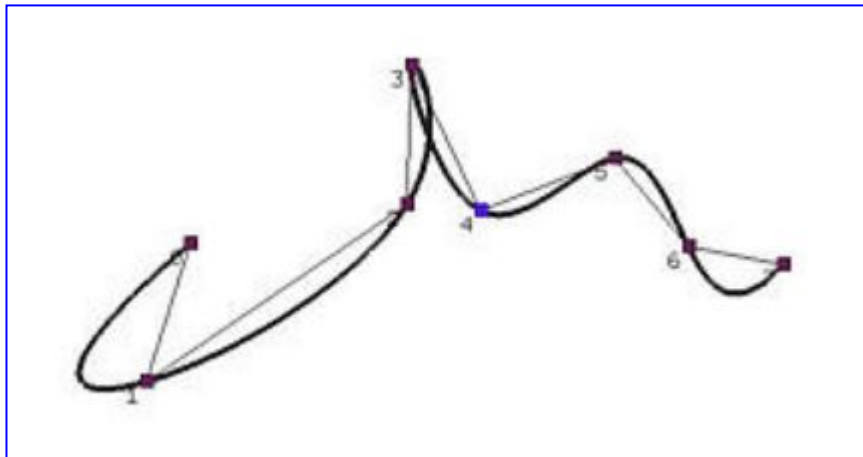


Figure 2.1 B-Spline curve interpolation with the uniformly spaced method<sup>[1]</sup>

### 2.2 Chord Length 弦长参数法

令  $d$  为总弦长, 且把参数范围限定在  $[0, 1]$  之间, 则:

$$d = \sum_{k=1}^n |Q_k - Q_{k-1}|$$

$$\begin{cases} \bar{u}_0 = 0 \\ \bar{u}_k = \bar{u}_{k-1} + \frac{|Q_k - Q_{k-1}|}{d} \\ \bar{u}_n = 1 \end{cases} \quad k = 1, 2, \dots, n-1$$

这是最常用的方法，并且一般用它就足够了，考虑到弦长参数化接近曲线的均匀参数化，在这种意义下，它给出了曲线的一个“好”的参数化。

### 2.3 Centripetal Method 向心参数法

$$d = \sum_{k=1}^n \sqrt{|Q_k - Q_{k-1}|}$$

$$\begin{cases} \bar{u}_0 = 0 \\ \bar{u}_k = \bar{u}_{k-1} + \frac{\sqrt{|Q_k - Q_{k-1}|}}{d} \\ \bar{u}_n = 1 \end{cases} \quad k = 1, 2, \dots, n-1$$

这是一个更新的方法，当数据点急转弯变化时，这个方法能得到比弦长参数化更好的结果。

### 3.BSpline Interpolation in OPEN CASCADE

OPEN CASCADE 对曲线的插值是通过 GeomAPI 包中的 GeomAPI\_Interpolate 实现的。由其代码注释可知，这个类的功能是可以对一系列点进行插值得到 C2 连续的 B 样条曲线，当对插值点处的切矢不作要求时。对点直接插值的构造函数为：

```
GeomAPI_Interpolate (const Handle< TColgp_HArray1OfPnt > &Points, const Standard_Boolean  
PeriodicFlag, const Standard_Real Tolerance)
```

其中参数 Points 为插值点，PeriodicFlag 为是否周期曲线，Tolerance 是对插值点进行检查用的容差。Tolerance 容易产生误解，根据插值曲线的定义，插值曲线是要求通过插值点的，所以不存在插值得到的曲线和插值点之间的容差。

经过查看 OPEN CASCADE 中插值曲线的源码，可以得出对曲线进行插值的步骤如下：

- ❖ 检查是否有重复的插值点 CheckPoints;
- ❖ 生成参数 BuildParameters;
- ❖ 使用 BSplCLib::Interpolate()进行插值;
- ❖ 根据参数及次数生成系数矩阵，再结合插值点，对系数矩阵和插值点组成的方程组进行求解。

检查插值点代码如下：

```
static Standard_Boolean CheckPoints(const TColgp_Array1OfPnt& PointArray,  
                                   const Standard_Real Tolerance)  
{  
    Standard_Integer ii ;  
    Standard_Real tolerance_squared = Tolerance * Tolerance,  
    distance_squared ;  
    Standard_Boolean result = Standard_True ;  
    for (ii = PointArray.Lower() ; result && ii < PointArray.Upper() ; ii++) {  
        distance_squared =  
            PointArray.Value(ii).SquareDistance(PointArray.Value(ii+1)) ;  
        result = (distance_squared >= tolerance_squared) ;  
    }  
    return result ;  
}
```

由上述代码可知，Tolerance 主要是用于检查插值点是否在容差范围内有重合现象。生成参数代码如下：

```
static void BuildParameters(const Standard_Boolean PeriodicFlag,  
                           const TColgp_Array1OfPnt& PointsArray,  
                           Handle(TColStd_HArray1OfReal)& ParametersPtr)  
{  
    Standard_Integer ii,  
    index ;  
    Standard_Real distance ;  
    Standard_Integer  
        num_parameters = PointsArray.Length() ;  
    if (PeriodicFlag) {  
        num_parameters += 1 ;  
    }  
    ParametersPtr =
```

```

new TColStd_HArray1OfReal(1,
                          num_parameters) ;
ParametersPtr->SetValue(1, 0. 0e0) ;
index = 2 ;
for (ii = PointsArray.Lower() ; ii < PointsArray.Upper() ; ii++) {
    distance =
        PointsArray.Value(ii).Distance(PointsArray.Value(ii+1)) ;
    ParametersPtr->SetValue(index,
                          ParametersPtr->Value(ii) + distance) ;
    index += 1 ;
}
if (PeriodicFlag) {
    distance =
        PointsArray.Value(PointsArray.Upper()).
        Distance(PointsArray.Value(PointsArray.Lower())) ;
    ParametersPtr->SetValue(index,
                          ParametersPtr->Value(ii) + distance) ;
}
}

```

由上述代码可知，OPEN CASCADE 插值生成参数的方法如下：

$$\begin{aligned} \bar{u}_0 &= 0 \\ \bar{u}_k &= \bar{u}_{k-1} + |Q_k - Q_{k-1}| \quad k = 1, 2, \dots, n \end{aligned}$$

不是上述三种常用方法的之一，和弦长参数化法类似，但是没有去除以总弦长。生成节点矢量之前为了得到曲线的次数，做了如下处理：

```

if (num_poles == 2 && !myTangentRequest) {
    degree = 1 ;
}
else if (num_poles == 3 && !myTangentRequest) {
    degree = 2 ;
    num_distinct_knots = 2 ;
}
else {
    degree = 3 ;
    num_poles += 2 ;
    if (myTangentRequest)
        for (ii = myTangentFlags->Lower() + 1 ;
            ii < myTangentFlags->Upper() ; ii++) {
            if (myTangentFlags->Value(ii)) {
                num_poles += 1 ;
            }
        }
}
}

```

由上述代码可知，插值要求至少有两个插值点。当只有两个插值点时，插值曲线次数为 1，即为直线；当有三个插值点且没有切矢的要求时，插值曲线次数为 2 次；当插值点数多于 3 个时，

插值曲线次数为 3。即对于多于三个点进行插值时，最高只能得到 3 次曲线，也即 C2 连续的曲线。进行 B 样条插值的代码如下：

```

void BSplCLib::Interpolate(const Standard_Integer      Degree,
                           const TColStd_Array1OfReal& FlatKnots,
                           const TColStd_Array1OfReal& Parameters,
                           const TColStd_Array1OfInteger& ContactOrderArray,
                           const Standard_Integer      ArrayDimension,
                           Standard_Real&             Poles,
                           Standard_Integer&          InversionProblem)
{
    Standard_Integer ErrorCode,
        UpperBandWidth,
        LowerBandWidth ;
    // Standard_Real *PolesArray = &Poles ;
    math_Matrix InterpolationMatrix(1, Parameters.Length(),
                                    1, 2 * Degree + 1) ;
    ErrorCode =
    BSplCLib::BuildBSpMatrix(Parameters,
                             ContactOrderArray,
                             FlatKnots,
                             Degree,
                             InterpolationMatrix,
                             UpperBandWidth,
                             LowerBandWidth) ;

    if(ErrorCode)
        Standard_OutOfRange::Raise("BSplCLib::Interpolate");

    ErrorCode =
    BSplCLib::FactorBandedMatrix(InterpolationMatrix,
                                 UpperBandWidth,
                                 LowerBandWidth,
                                 InversionProblem) ;

    if(ErrorCode)
        Standard_OutOfRange::Raise("BSplCLib::Interpolate");

    ErrorCode =
    BSplCLib::SolveBandedSystem(InterpolationMatrix,
                                UpperBandWidth,
                                LowerBandWidth,
                                ArrayDimension,
                                Poles) ;

    if(ErrorCode)
        Standard_OutOfRange::Raise("BSplCLib::Interpolate");
}

```

先是根据参数及插值曲线次数生成系数矩阵，再对系数矩阵和插值点构成的方程组进行求解，计算出 B 样条曲线的控制顶点 Poles。有了节点矢量，次数及控制顶点，B 样条就确定下来了：

```

myCurve =
    new Geom_BSplineCurve(poles,
        myParameters->Array1(),
        mults,
        degree) ;
myIsDone = Standard_True ;

```

OPEN CASCADE 提供的插值接口使用还是很简单，如对已知点进行插值，其用法如下：

```

int main(int argc, char* argv[])
{
    Handle_TColgp_HArray1ofPnt aPoints = new TColgp_HArray1ofPnt aPoints(1, 3);
    Handle_Geom_BSplineCurve aBSplineCurve;

    aPoints.SetValue(1, gp_Pnt(0.0, 0.0, 0.0));
    aPoints.SetValue(2, gp_Pnt(1.0, 1.0, 0.0));
    aPoints.SetValue(3, gp_Pnt(2.0, 6.0, 3.0));

    GeomAPI_Interpolate          aInterpolator(aPoints,          Standard_False,
Precision::Approximation());

    if (aInterpolator.IsDone())
    {
        aBSplineCurve = aInterpolator.Curve();

        GeomTools::Dump(aBSplineCurve, std::cout);
    }
}

```

## 4. Conclusion

综上所述，对给定点进行 B 样条插值时，需要考虑参数值及节点矢量的选择。参数值和节点矢量确定后，剩下就是利用 B 样条基函数对给定点的参数计算得到的系数组成的线性方程进行求解。

在使用 OPEN CASCADE 的曲线插值类 `GeomAPI_Interpolate` 时，需要注意公差 `Tolerance` 是用来对插值点进行检查的，且插值得到的曲线最高只能是三次曲线。

## 5. Acknowledgments

首先，感谢 `cnblog` 提供了一个表现自己的舞台 <http://www.cppblog.com/eryar/>，能在网上和世界连通，知道不是一个人在战斗。

其次，感谢 OPEN CASCADE 的开源分享，才得以学到几何造型相关的知识，比起直接啃国内教材来，学习效率不可同日而语。正如“Talk is cheap, show me the code”所说，将代码和书本结合起来学习时，收获更大。

最后，感谢国内外友人对我的肯定和鼓励，他们自强不息，激情创业的精神总是让人兴奋。生活的理想就是为了理想的生活 *The ideal of life is to live for ideals!* 人生充满了起起落落，关键在于在顶端时尽情享受，在低谷时不失勇气。

## 6. References

1. Hongxin Zhang, Jieqing Feng. B-Spline Interpolation and Approximation. Zhejiang University. 2006-12-18. <http://www.cad.zju.edu.cn/home/zhx/GM/009/00-bsia.pdf>
2. Les Piegl, Wayne Tiller. The NURBS Book. Springer-Verlag. 1995
3. 赵罡, 穆国旺, 王拉柱译. 非均匀有理 B 样条. 清华大学出版社. 1995
4. 易大义, 陈道琦. 数值分析引论. 浙江大学出版社. 1998