# 用 NURBS 曲面逼近一般光滑曲面的技术路线

许中兴
RISE 团队
智能软件研究中心
中国科学院软件研究所
xuzhongxing@iscas.ac.cn

## 1 引言

在《OpenCascade 中的 Thin Plate Spline 插值》中，我们讨论了如何用 Thin plate spline 函数来近似生成包含任意约束的光滑曲面。然而在 CAD 系统中，曲线曲面的通用表示是 NURBS 方式。所以本文讨论从一般光滑曲面到 NURBS 曲面的变换过程。

一种直接的方式是，直接在光滑曲面上进行网格状采样，然后对网格状采样点直接进行 NURBS 插值。但是在实际的工程实现中不采用这样的方式。因为通常情况下，使用单个 NURBS 曲面对一个光滑曲面进行整体插值得到的逼近曲面往往会和原始曲面有较大出入，贴合性不好。故而需要用迭代的方式对原始曲面进行分片逼近。先尝试对原始曲面进行整体逼近，然后通过采样计算出逼近面和原始面的误差，如果误差过大，则对原始曲面进行一次分片，再重复逼近过程，直至得到较好的逼近分片。在这个过程中可以看到，要进行很多次曲面逼近，而不是一次。而用 NURBS 对采样点进行插值，归结到最后需要求解一个线性方程组，这个计算代价较高，不适合进行多次计算。

所以，在这个过程中不采用直接插值的变换方式，而是借助正交多项式，先得到对原始曲面的多项式逼近，再从多项式逼近得到 NURBS 逼近。

光滑函数的正交多项式逼近是将函数向一系列的正交多项式 $P_n$ 进行投影。本文中所使用的正交多项式是 Legendre 多项式。投影的过程是将光滑

函数与 Legendre 函数在 $[-1,1]$ 区间上做积分。用高斯积分公式，这个过程只需要计算函数值求和，速度很快。

从 Legendre 多项式向 B 样条变换，属于多项式之间的变换，无法贴合的现象会减轻许多（尽管也有可能需要对逼近过程进行迭代，实际中如果 knots 取的足够多，一次逼近就够了），效率会高很多。

本文涉及众多数学理论。本文的目的不是对这些数学理论进行详尽的描述，而是针对要解决的具体问题，从浩瀚的文献和程序代码中整理出一条完整可行的技术路线。对涉及的理论尽量给出最原始最合适的参考文献。

# 2 总体过程

这篇文章主要是讨论一元函数的逼近，因为曲面的逼近过程中需要进行多次曲线的逼近。同时，从一元函数的逼近向二元函数逼近的推广也非常直接，把基函数 $\phi_i(x)$ 变成张量基 $\phi_i(x)\phi_j(y)$，把一重积分变成二重积分即可。

假设我们要用 B 样条逼近一个光滑函数 $f(x)$, $f(x)$ 的形式有可能是已知的，但比较复杂，比如包含了指数对数函数，也有可能是未知的，但是给定一个参数 $x_0$ 可以通过一个过程来计算函数的值 $f(x_0)$。例如，在扫略面中，截面线通过各种方式变换到扫略引导线的各个位置上，得到一条新的截面线，这个新的截面线的形式往往不能通过有理多项式的形式表示出来，但是它在每个参数的值是可以计算的。

整个逼近过程分成两个阶段，第一阶段用 Legendre 多项式逼近原始函数 $f(x)$，得到 $L(f)$。第二阶段用 B 样条逼近 $L(f)$，得到最终需要的 B 样条曲线。

# 3 Legendre 多项式

Legendre 多项式 $P_n(x)$ 是更一般的正交多项式 Jacobi 多项式的一个特例。有关正交多项式的理论参见 [1]。本节中的公式来自于文献 [2]。在这里说点题外话，这本书的名字是“现代分析”，但它的内容却是地地道道的古典分析，描述的内容都是具体的函数，所使用的方法全部是经典的。现代数学在进入二十世纪之后逐渐开始将关注点放在抽象结构上，不再关心具体的数学对象。但是在计算机发明之后，数学里的古典内容在计算的加持之下

又开始放出光芒。

由于正交多项式乘上任意常数仍然是正交的,我们按照惯例规定 $P_n(1) = 1$。Legendre 多项式的闭形式如下。

$$P_n(x) = \sum_{r=0}^{m} (-1)^r \frac{(2n-2r)!}{2^n r!(n-r)!(n-2r)!} x^{n-2r}$$

其中 $m = \frac{1}{2}n$ 取下整数。

注意这个形式的 Legendre 多项式是没有归一的, 即它的范数不是 1. 它的归一化常数为:

$$\gamma_n = \int_{-1}^{1} P_n(x)P_n(x)dx = \frac{2}{2n+1}$$

在工程实现中，我们不使用这个闭形式来计算，而是使用递推公式来计算 $P_n(x)$ 的值。

$$(n+1)P_{n+1}(x) = (2n+1)xP_n(x) - nP_{n-1}(x)$$

导数递推公式:

$$(1-x^2)P'_n(x) = -nxP_n(x) + nP_{n-1}(x)$$

# 4 Legendre 逼近

这部分的内容是标准的，参见 [3] 第 7 章。

注意这里因为 Legendre 多项式没有归一，积分得到的投影系数需要除以归一化常数 $\gamma_n$ 才是逼近级数里的系数。

# 5 Legendre-Gauss 积分公式

这部分的内容是标准的，参见 [3] 第 8 章。

假设用 $n$ 次 Legendre 多项式的根 $x_1, \ldots, x_n$ 构造积分公式，积分权重为:

$$\lambda_i = -\frac{2}{n+1} \frac{1}{P_{n+1}(x_i)P'_n(x_i)}$$

# 6 多项式求根

针对多项式的任意精度的求根有很好的开源库实现，如 mpmath[4]. 在实际工程实现中，不会每次都现计算 Legendre 多项式的根，而是会事先把要用到的所有的根（以及对应的高斯积分系数）都计算出来，放在数据表中供其他程序使用。

所使用的算法是 Durand-Kerner 方法 [5]。能够把根求到指定的精度的理论依据，即所求得的根的误差估计来自于文献 [6] 以及它的参考文献 [7]。

# 7 Legendre 多项式的 B 样条逼近

方法参见文献 [8].

# 8 对边界条件的处理

我们往往需要最终的 NURBS 曲面符合边界条件的约束，即让边界位置，一阶导数，甚至二阶导数符合指定的值。一方面我们可以在从 Legendre 多项式向 NURBS 转换的时候指定边界条件 [8]。另一方面，我们在对光滑函数进行 Legendre 逼近的时候，可以采用一个技巧，就是先对光滑函数 $f$ 进行一次 Hermite 插值，将原始函数的边界条件提取出来，得到 $H(f)$, 然后再对 $f - H(f)$ 进行 Legendre 逼近，得到 $L(f)$, 最后的逼近结果是 $H(f) + L(f)$.

这样做的理论依据是，当被逼近的原始函数的值越接近 0 的时候，同样的逼近精度能得到的误差就越小，参见 [9] 第 6 章第 3 节。

# 参考文献

[1] Gabor Szego. *Orthogonal Polynomials.* American Mathematical Society, 1975.

[2] E. T. Whittaker and G. N. Watson. *A Course of Modern Analysis.* Cambridge University Press, 1950.

[3] Francis B. Hildebrand. *Introduction to Numerical Analysis.* Dover Publications, 1987.

[4] mpmath - python library for arbitrary-precision floating-point arithmetic. `https://mpmath.org/`.

[5] Durand–kerner method. `https://en.wikipedia.org/wiki/Durand%E2%80%93Kerner_method`.

[6] M. S. Petkovic, C. Carstensen, and M. Trajkovic. Weierstrass formula and zero-finding methods. *Numerische Mathematik*, 69:353–372, 1995.

[7] D. Braess and K. P. Hadeler. Simultaneous inclusion of the zeros of a polynomial. *Numerische Mathematik*, 21:161–165, 1973.

[8] Franz-Erich Wolter and Seamus T. Tuohy. Approximation of high-degree and procedural curves. *Engineering with Computers*, 8:61–80, 1992.

[9] John R. Rice. *The Approximation of Functions Volume 1.* Addison-Wesley, 1967.