应用数学和力学，第6卷第6期(1985年6月)
Applied Mathematics and Mechanics

应用数学和力学编委会编
重庆出版社出版

一个构造三次样条函数的简便方法*

谭福期

（长沙工程兵学校，1982年5月19日收到）

摘　要

本文给出一个构造三次样条函数的简便方法，并讨论了与之有关的问题，最后提出了针对不同情况的处理方法.

采用样条函数作为插值函数的优越性已为大家所熟知。然而由于构造样条函数比较麻烦，也限制了它的应用的广泛性。大家知道，为了构造样条函数，需要将各节点的函数值以及边界条件同时引入一方程组，然后求解这一方程组得到构造样条函数所必需的待定参数。例如对三次样条函数来说就是求解出各节点的二阶导数（即样条的弯矩），才能得到样条函数。这样做，不仅是引起数学处理上的麻烦和浪费电子计算机的内存，而且对于某些情况下的应用简直是不可能的。例如，当我们只有简单的计算工具（包括袖珍电子计算机）或者因我们计算的问题简单不值得编排程序上电子计算机的时候，按照过去构造样条函数的方法，应用样条函数作插值计算是不可能的。可不可以寻求一个构造常用的三次样条函数的简便方法呢？为此，作者经过研究，觉得对于常用的三次样条函数可以采取逐段递推构造法，效果是相当好的。下面就介绍逐段递推构造法。

我们仅考虑相邻的三个结点x_{k-1}，x_k，x_{k+1}，若已知其函数值分别为f_{k-1}，f_k，f_{k+1}，以及x_{k-1}点的二阶导数为f''_{k-1}。那么，我们可以假设这一段的样条函数为

$$s(x)=A(x-x_{k-1})(x-x_k)(x-x_{k+1})+B(x-x_{k-1})(x-x_k)+C(x-x_{k-1})+D \qquad (1)$$

将x_{k-1}，x_k，x_{k+1}点的函数值代入（1）式，可以得到含待定常数A，B，C，D的三个方程。再对$s(x)$求二阶导数，然后以x_{k-1}点的已知二阶导数f''_{k-1}代入，又可以获得一方程。此四个方程联立求解可唯一确定A，B，C，D的值。求得的结果如下：

$$\left.\begin{aligned}
&D=f_{k-1},\quad C=\frac{f_k-f_{k-1}}{x_k-x_{k-1}}\\
&B=\left(\frac{f_{k+1}-f_{k-1}}{x_{k+1}-x_{k-1}}-\frac{f_k-f_{k-1}}{x_k-x_{k-1}}\right)\Big/(x_{k+1}-x_k)\\
&A=\left(\frac{f''_{k-1}}{2}-B\right)\Big/(2x_{k-1}-x_k-x_{k+1})
\end{aligned}\right\} \qquad (2)$$

* 钱伟长推荐.

按（2）式求得的A，B，C，D代入（1）式，即得到x_{k-1}至x_{k+1}这一段的样条函数$s(x)$。若对求得的$s(x)$求二阶导数，再以$x=x_{k+1}$代入，便可求得x_{k+1}点处的二阶导数，我们以f''_{k+1}表示之，则

$$f''_{k+1}=2A(2x_{k+1}-x_{k-1}-x_k)+2B \tag{3}$$

其中A与B即为（2）式中求出的A，B值。再将（3）式求得的f''_{k+1}的值作为构造下一段样条函数的初值，则可求得下一段（即x_{k+1}至x_{k+3}一段）的样条函数。（方法同前）如此递推下去，就可以求得各段样条函数。

这样一来，我们只要知道各结点的函数值以及初始的二阶导数，即可以按上面介绍的逐段递推法求得三次样条函数。

是不是任何被插值的函数都可以按此法构造三次样条函数呢？或者说对于不同的实际问题应该怎样处理呢？下面我们就来讨论这个问题。

必须指出，严格地说这样构造的函数并不是精确的样条函数，而只是样条函数的近似解，因为在各分段结点上一阶导数是不连续的。对于有的情况分段结点上一阶导数的左边值与右边值相差甚微，这时我们可以把它看作样条函数有效的近似解；而对于另一些情况，可能分段结点上一阶导数的左边值与右边值相差很大，这时仍把它当作样条函数的近似解就不妥当了。那么究竟什么样的情况才是可行的呢？它是受哪些条件控制的呢？下面我们先从物理意义上作些说明，然后再进行数学分析，最后对于不同的情况提出相应的处理方法。

我们知道，三次样条函数的物理意义是一条过各插值点的弹性样条的挠曲线方程，它的走向取决于各插值点的位置和边界条件。通常构造样条函数需要两个边界条件，使样条函数唯一确定。而我们这种逐段递推构造法只需一个边界条件，另一个边界条件是从属的，只要初始边界条件给定，终端的边界条件便对应确定，它们之间有确定的对应关系。这样从属的终端值有两种可能：一是接近实际的终端边界条件，因而误差较小，是有效的；二是与实际的终端边界条件完全不相符合，因而误差很大，是不允许的。毫无疑问，为了使得我们构造的函数符合要求，有必要附加一定的条件。下面我们就来分析这个问题。

从样条函数的物理意义我们知道，作为一条弹性挠曲线，其弹性势能必须取极小值。也就是说，必须

$$\int_{x_0}^{x_N}[s''(x)]^2dx=\min \tag{4}$$

下面我们对x_{k-1}至x_{k+1}一段求这个积分。

由（1）可知　$s''(x)=2[3Ax+B-A(x_{k-1}+x_k+x_{k+1})]$

将$s''(x)$代入$\int_{x_{k-1}}^{x_{k+1}}[s''(x)]^2dx$中进行积分，并注意到

$$A=\left(\frac{f''_{k-1}}{2}-B\right)\Big/(2x_{k-1}-x_k-x_{k+1})$$

于是化简整理后得

$$\int_{x_{k-1}}^{x_{k+1}}[s''(x)]^2dx=12A^2(x_{k+1}-x_{k-1})^3+6A(x_{k+1}-x_{k-1})^2f''_{k-1}+(x_{k+1}-x_{k-1})(f''_{k-1})^2 \tag{5}$$

由（5）式可知，$\int_{x_{k-1}}^{x_{k+1}}[s''(x)]^2dx$是关于$A$的二次函数。根据二次函数的性质，由于$A^2$项的系数为正，且判别式$b^2-4ac<0$，故它恒为正，当$A=-b/2a$时有极小值。在这里

$$a=12(x_{k+1}-x_{k-1})^3,\quad b=6(x_{k+1}-x_{k-1})^2 f''_{k-1},\quad c=(x_{k+1}-x_{k-1})(f''_{k-1})^2$$

为使$\int_{x_{k-1}}^{x_{k+1}}[s''(x)]^2dx$取极小值，必须

$$A=-\frac{b}{2a}=\frac{-f''_{k-1}}{4(x_{k+1}-x_{k-1})} \tag{6}$$

由（2）式可知

$$A=\left[\frac{f''_{k-1}}{2}-(m_1-m_2)\frac{1}{(x_{k+1}-x_k)}\right]\cdot\frac{1}{2x_{k-1}-x_k-x_{k+1}}$$

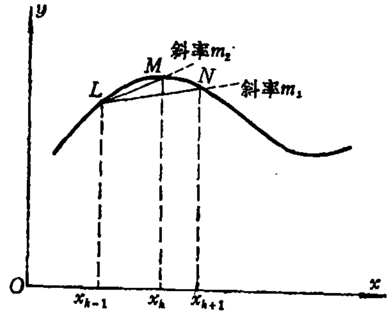

图 1

其中 $m_1=(f_{k+1}-f_{k-1})/(x_{k+1}-x_{k-1})$,

$m_2=(f_k-f_{k-1})/(x_k-x_{k-1})$分别表示线段$LN$与$LM$的斜率(见图1)。将$A$代入（6）式得

$$\left[\frac{f''_{k-1}}{2}-(m_1-m_2)\frac{1}{x_{k+1}-x_k}\right]\frac{1}{2x_{k-1}-x_k-x_{k+1}}=\frac{-f''_{k-1}}{4(x_{k+1}-x_{k-1})}$$

化简后为

$$m_1-m_2=\frac{(x_{k+1}-x_k)(3x_{k+1}+x_k-4x_{k-1})}{4(x_{k+1}-x_{k-1})}f''_{k-1}$$

或

$$f''_{k-1}=\frac{4(m_1-m_2)(x_{k+1}-x_{k-1})}{(x_{k+1}-x_k)(3x_{k+1}+x_k-4x_{k-1})} \tag{7}$$

由于

$$\frac{(x_{k+1}-x_k)(3x_{k+1}+x_k-4x_{k-1})}{4(x_{k+1}-x_{k-1})}>0$$

故m_1-m_2与f''_{k-1}同号，即 $(m_1-m_2)f''_{k-1}>0$ （8）

很显然，当m_1-m_2与f''_{k-1}满足（7）式时，$\int_{x_{k-1}}^{x_{k+1}}[s''(x)]^2dx$取极小值。我们知道，对样条函数来说，只要满足(4)式，即由x_0至x_N整个积分取极小值，而不可能要求每一段的$\int_{x_{k-1}}^{x_{k+1}}[s''(x)]^2dx$都取极小值。逐段递推法构造的三次样条函数只是样条函数的近似解，更不可能使每一段的$\int_{x_{k-1}}^{x_{k+1}}[s''(x)]^2dx$取极小值。实际上对于第一段我们可以在给定初始的二阶导数f''_0时使其满足这一条件，但对于第二段就不可能了。因为第二段的初始二阶导数是由第一段求出的，它是已确定的，而第二段的m_1-m_2是由被插值函数确定的，也是已知的。显然不可能恰好满足（7）式。以后各段也是如此。作者经过大量的分析研究认为，对于逐段递推法，只要第一个初始值给得合理，以后各段又不违背（8）式，则按此法构造的三次样条函数就有很好的精度，是完全可以应用的。所谓第一个初始值给得合理，很大程度上取决于各人的经验。如无经验可鉴，则可按（7）式求出。若知其精确的初始二阶导数时，以此精确值作初始条件最好。于是我们可以把处理的方法归纳如下：

1. 对于第一段，已知精确的初始二阶导数时，即以此值作为初始二阶导数。若不知精确值则可按（7）式解出，即以

$$f''_0=\frac{4(m_1-m_2)(x_2-x_0)}{(x_2-x_1)(3x_2+x_1-4x_0)}$$

作为初始条件，或者按经验选取适当的f''_0。

2. 从第二段起，如果前一段求出的末一结点的二阶导数与相邻的下一段的 m_1-m_2 同号，即满足（8）式，则可按本文介绍的逐段递推法做下去。相反，如果遇到前一段求出的末一结点的二阶导数与相邻的下一段的m_1-m_2反号时，即不满足（8）式，则应从这一段起重新给定初始的二阶导数后再按本文介绍的逐段递推法做下去。例如在 x_{k-1} 至 x_{k+1} 一段求出的 f''_{k+1}为正，而下一段（即 x_{k+1} 至 x_{k+3} 一段）的 m_1-m_2 的值为负（即$[(f_{k+3}-f_{k+1})/(x_{k+3}-x_{k+1})]-[(f_{k+2}-f_{k+1})/(x_{k+2}-x_{k+1})]$ 的值为负），那么就应该按前一条所说的办法重新给定 f''_{k+1}，当使其满足（7）式时，即为

$$f''_{k+1}=\frac{4(m_1-m_2)(x_{k+3}-x_{k+1})}{(x_{k+3}-x_{k+2})(3x_{k+3}+x_{k+2}-4x_{k+1})}$$

其中 $m_1=(f_{k+3}-f_{k+1})/(x_{k+3}-x_{k+1})$，$m_2=(f_{k+2}-f_{k+1})/(x_{k+2}-x_{k+1})$，然后再按逐段递推法做下去。显然，在这一个分段结点上二阶导数是不连续的。若后面再遇到此种情形，照此办理。

作者以正弦函数$\sin 10x$为例做了一个计算，其结果列于表1中。函数$\sin 10x$在0至$\pi/10$之间是一个半波，将它分成18等分，每个分点的函数值作为插值点的已知值f_0，$f_1,\cdots,f_{18}$，即表1中的第二行。由于在计算中未遇到前一段的末一结点的的二阶导数与相邻下一段的m_1-m_2反号的情况，故中间没有二阶导数间断点。表1中的f''栏中的计算值第一个是给定的初始值，其余的是根据本文提出的方法计算出来的，与精确值非常接近。（初始值给的是精确值，即 $f''_0=0$）f' 的计算值是根据本文提出的方法计算出来的。（对于各分点，f' 的左边值与右边值略有差异，但差得甚微，都与精确值接近。表1中列的是左边值）f 一列求的是两结点之间的插值，可以看出是相当精确的。

显而易见，逐段递推法比较适用于那些变化缓慢的函数插值。对于那些变化急剧的函数（尤其是二阶导数经常反向）逐段递推法误差较大。因为它将不断遇到前一段求出的末一结点的二阶导数与相邻下一段的m_1-m_2反号的情形，从而老是要重新给定初始二阶导数，使二阶导数不连续的点增多，整体的光顺性就差了。也可以每段的初始二阶导数都按（7）式给定，这时分段结点上的一阶导数和二阶导数都不连续，此时就是分段三次样条了。

表 1 函数 sin 10x 与按逐段递推法构造的三次样条函数值对照

x	f_k	f'' 精确值	f'' 计算值	f' 精确值	f' 计算值	f 精确值	f 计算值
0.000000	0.000000	0.000000	0.000000	10.000000	10.050077		
0.008727						0.087156	0.087154
0.017453	0.173648	−17.364818	−17.320810	9.848078	9.848540		
0.026180						0.258819	0.258823
0.034907	0.342020	−34.202014	−34.641619	9.396926	9.395082		
0.043633						0.422618	0.422619
0.052360	0.500000	−50.000000	−49.873218	8.660254	8.660660		
0.061087						0.573576	0.573582
0.069813	0.642788	−64.278761	−65.104818	7.660444	7.657288		
0.078540						0.707107	0.707110
0.087266	0.766044	−76.604444	−76.410178	6.427876	6.428178		
0.095993						0.819152	0.819160
0.104720	0.866025	−86.602540	−87.715539	5.000000	4.995911		
0.113446						0.906308	0.906314
0.122173	0.939693	−93.969262	−93.730988	3.420201	3.420365		
0.130900						0.965926	0.965934
0.139626	0.984808	−98.480775	−99.746437	1.736482	1.731956		
0.148353						0.996195	0.996148
0.157080	1.000000	−100.000000	−99.179116	0.000000	−0.001655		
0.165806						0.996195	0.996126
0.174533	0.984808	−98.480775	−98.611795	−1.736482	−1.727706		
0.183260						0.965926	0.965912
0.191986	0.939693	−93.969262	−93.730988	−3.420201	−3.417061		
0.200713						0.906308	0.906335
0.209440	0.866025	−86.602540	−88.350181	−5.000000	−5.010383		
0.218166						0.819152	0.819181
0.226893	0.766044	−76.604444	−76.410178	−6.427876	−6.431478		
0.235619						0.707107	0.707089
0.244346	0.642788	−64.278761	−63.970176	−7.660444	−7.656528		
0.253073						0.573576	0.573561
0.261799	0.500000	−50.000000	−49.873218	−8.660254	−8.657360		
0.270526						0.422618	0.422641
0.279252	0.342020	−34.202014	−35.776260	−9.396926	−9.404793		
0.287979						0.258819	0.258845
0.296706	0.173648	−17.364818	−17.320810	−9.848078	−9.851840		
0.305433						0.087156	0.087132
0.314159	0.000000	0.000000	1.134641	−10.000000	−9.993091		

参 考 文 献

[1] Anlbery, J. H. et al, *The Theory of Spline and Their Application*, Acad. Press, N. Y. (1967).

[2] 谭福期, Fourier 积分的数值计算法, 应用数学和力学, **2**, 5 (1981), 581—584.

A Simplified Method for the Construction of the Cubic Spline Function

Tan Fu-qi
(*The P. L. A. Engineer School, Changsha*)

Abstract

This paper provides a simplified method for the construction of the cubic spline function, discusses problems related to the method, and gives the methods of treatment for different cases.