

用谱方法解微分方程

December 3, 2009

这是一个关于用谱方法解微分方程的介绍，有许多细节问题没有介绍，例如算法收敛性，稳定性等定理没有涉及，一方面我还没有看懂，一方面那些内容涉及更深入些的数学知识，而那些数学知识不能三言两语解释清楚。我在最后列出了参考书，有兴趣的同学可以找来看看。

1 Chebyshev 多项式

Chebyshev 多项式是微分方程

$$\left(\sqrt{1-x^2}y'\right)' + \frac{n}{\sqrt{1-x^2}}y = 0 \quad (1)$$

的解。记 n 阶Chebyshev 多项式为 T_n 。Chebyshev 多项式关于测度(权函数) $1/\sqrt{1-x^2}$ 正交

$$(T_n, T_m) = \int_{-1}^1 \frac{T_n T_m}{\sqrt{1-x^2}} dx = \frac{2}{\pi} (1 + \delta_{0n}) \delta_{mn} = \begin{cases} 0 & k \neq j \\ \frac{2}{\pi} & k = j \neq 0 \\ \pi & k = j = 0 \end{cases} \quad (2)$$

Chebyshev 多项式的递归公式

$$T_{n+1}(x) = 2xT_n(x) - T_{n-1}(x) \quad (3)$$

前5个Chebyshev 多项式为

$$\begin{aligned} T_0 &= 1 \\ T_1 &= x \\ T_2 &= 2x^2 - 1 \\ T_3 &= 4x^3 - 3x \\ T_4 &= 8x^4 - 8x^2 + 1 \\ T_5 &= 16x^5 - 20x^3 + 5x \end{aligned}$$

他们的图：
可以看出

1. T_n 的奇偶性与 n 相同。
2. T_n 是 n 次多项式。

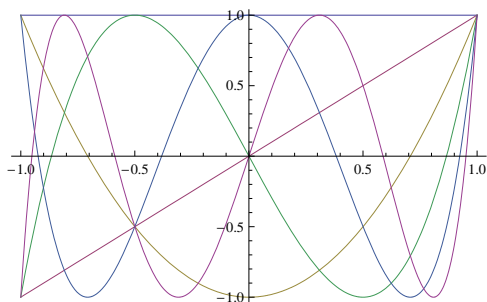


Figure 1: Chebyshev 多项式

3. $T_n(\pm 1) = (\pm 1)^n$ 。

4. T_n 在区间 $[-1, 1]$ 上有 n 个零点。

Chebyshev 多项式的求导公式

$$T'_{2N}(x) = \sum_{k=0}^N 4NT_{2k-1}(x) \quad (4)$$

$$T'_{2N+1}(x) = (2N+1)T_0 + (4N+2) \sum_{k=1}^N T_{2k} \quad (5)$$

2 Lagrange 插值多项式

如果有一个未知函数，仅仅知道这个函数在某些点的函数值。能否用较为简单的函数来代替未知函数。或者如果有一个较为复杂的函数，仅仅能计算出少数点的函数值，可不可以用较为简单的函数近似代替原函数，计算出其他点的函数值。这就是函数插值需要解决的问题。插值方法包括Lagrange 插值，Newton 插值，Hermite 插值，样条插值等。这几种插值主要是具体应用中要求不同而灵活采用。在此简要介绍一下Lagrange 插值。

定义 1 分布于区间 $[-1, 1]$ 上的 $N+1$ 个点 $x_i \in [-1, 1]$, $0 \leq i \leq N$ 称为 X 上的格子。 x_i 称为格子的节点。

考虑一个连续函数 f ，给定一组节点 $\{x_i\}$, $0 \leq i \leq N$ ，则存在唯一的 N 次多项式 $I_N^X f$

$$I_N^X f = \sum_{i=0}^N f(x_i) l_i^X(x) \quad (6)$$

其中

$$l_i^X(x) = \prod_{j=0, j \neq i}^N \frac{x - x_j}{x_i - x_j} \quad (7)$$

是一个 N 次多项式。在任意节点处

$$I_N^X f(x_i) = f(x_i) \quad 0 \leq i \leq N \quad (8)$$

对于一个插值多项式与原函数的的差别可以用下面定理判断

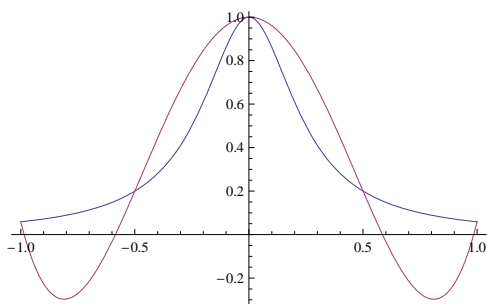


Figure 2: N=4的等距插值

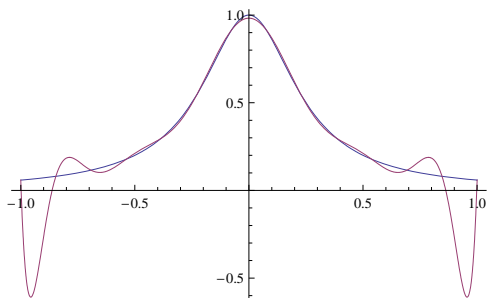


Figure 3: N=13的等距插值

定理 1 (误差估计) 设 $f(x)$ 在 $[a, b]$ 内具有 n 阶连续导数, 在 $[a, b]$ 内具有 $n+1$ 阶导数, $\phi(x)$ 是满足插值条件(8)的次数不超过 n 的插值多项式, 则对任意 $x \in [a, b]$, 存在 $\xi = \xi(x)$ 使得

$$R_n(x) = f(x) - \phi(x) = \frac{f^{(n+1)}(\xi)}{(n+1)!} \omega_{n+1}(x) \quad (9)$$

成立, 其中 $\omega(x) = \prod_{i=0}^n (x - x_i)$. 若当 $x \in [a, b]$ 时, $|f^{(n+1)}(x)| \leq M_{n+1}$, 则有

$$R_n(x) \leq \frac{M_{n+1}}{(n+1)!} |\omega_{n+1}(x)| \quad (10)$$

以下两图是 $1/(1+16x^2)$ 的等距插值。

从图2以及图3可以看出随着插值点的增加, 插值效果没有提高。这种现象叫做Runge 现象。

有没有办法找到一个代数多项式, 令其在区间中可以比较“均匀”的代替原函数呢? 这首先要明确均匀的意义, 它的定义一般有两种。记 $f(x)$ 为原函数, $\varphi(x)$ 为替代函数。一种要求 $f(x)$ 与 $\varphi(x)$ 的差在要求区间上的最大值小于某一标准, 即以 $\|f - \varphi\| \equiv \max_{x \in [a, b]} |f(x) - \varphi(x)|$ 作为标准, 这称为在一致意义下逼近。一种要求 $f(x)$ 与 $\varphi(x)$ 的差的平方的积分值小于某一标准, 即以 $\|f - \varphi\|_2 \equiv \int_a^b [f(x) - \varphi(x)]^2 dx$ 为标准, 这称为在平方意义下逼近。可以看出如果达到一致逼近的标准可以有效的避免Runge 现象。剩下的问题就是如何找到在一致意义下逼近原函数的代数多项式了。

3 最佳一致逼近

首先是这样的多项式存在吗? Weierstrass 在1885年证明了一个定理, 保证了这样

多项式的存在

定理 2 (Weierstrass 定理) 若函数 $f(x)$ 在区间 $[a, b]$ 上连续, 则对任意正数 ε , 存在多项式 $\phi(x)$, 使得对一切 $x \in [a, b]$, 都有 $|f(x) - \phi(x)| < \varepsilon$ 成立。

这个定理从理论上肯定了闭区间上连续函数可以用多项式以任意精度来一致逼近。但没有给出具体的多项式。Bernstein 在给出这个定理的一个构造性证明, 在此过程中他给出了一个可以逼近闭区间连续函数的多项式, 现在称为 Bernstein 多项式。

$$B_n(f, x) = \sum_{k=0}^n f\left(\frac{k}{n}\right) \binom{n}{k} x^k (1-x)^{n-k}, \quad n = 1, 2, \dots$$

其中 $f(x)$ 在 $[0, 1]$ 上连续

定理 3 设函数 $f(x)$ 在闭区间 $[0, 1]$ 连续, 则多项式序列 $B_n(f, x)$ 在这闭区间上一致收敛于 $f(x)$ 。

但是 Bernstein 多项式收敛速度很慢, 需要取高次的 Bernstein 多项式才能达到较高精度。计算机是有限精度的计算工具, 计算高次多项式会带来很大的误差, 这在计算方面是十分不利的。所以 Bernstein 多项式在理论上是重要的, 但在实际计算中很少使用。既然计算高次多项式会造成数值不稳定, 接下来的问题就是如何找到尽可能低次的多项式, 而达到尽可能高的逼近精度。这个问题就是最佳逼近问题。这个问题的严格数学表述是: 对于给定的函数 $f(x) \in C[a, b]$, 在次数不超过 n 的多项式集合 $P_n = \text{span}\{1, x, \dots, x^n\}$ 中去寻找一个多项式 $p_n^*(x)$ 使得

$$\|f - p_n^*\|_\infty \leq \min_{p_n \in P_n} \|f - p_n\|_\infty$$

定理 4 (Borel 存在性定理) 对任意给定在 $[a, b]$ 上连续的函数 $f(x)$, 总存在 $p_n(x) \in P_n$ 使得

$$\|f - p_n^*\|_\infty \leq \min_{p_n \in P_n} \|f - p_n\|_\infty \quad (11)$$

这个定理肯定了最佳逼近多项式的存在性

定理 5 (Chebyshev 定理) 对任意 $f \in C_{[a,b]}$, $f \notin P_n$, 是 f 的最佳一致逼近多项式的充要条件是 $f - p$ 在 $[a, b]$ 上存在至少有 $n + 2$ 个点组成的交错点组。

Chebyshev 定理给出了最佳逼近多项式的刻画。由 Chebyshev 定理还可以得到一个推论, 推论说最佳逼近多项式的唯一的

定理 6 (唯一性) 若函数 $f(x)$ 在 $[a, b]$ 上是连续函数, 则 $f(x)$ 的最佳逼近多项式 $p_n(x) \in P_n$ 是唯一的。

这个定理就肯定了, 如果插值点选得合适, 那么插值多项式就是最佳一致逼近多项式。现在回过头考虑前面提到的 Runge 现象。

定义 2

$$\Lambda_N(X) = \max_{x \in [-1, 1]} \sum_{i=0}^N |l_i^X(x)|$$

称为格点的 Lebesgue 常数

对于任何一个插值而言有

$$\|f - I_N^X f\|_\infty \leq (1 + \Lambda_N(X)) \|f - p_N^*\|_\infty \quad (12)$$

关于 $\Lambda_N(X)$ 有如下定理

定理 7 (Eröds) 对于任意选定的一组格点 X , 存在常数 $C > 0$ 使得

$$\Lambda_N(X) > \frac{2}{\pi} \ln(N+1) - C$$

这个定理表明当 $N \rightarrow \infty$ 时 $\Lambda_N(X) \rightarrow \infty$ 。这表明, 如果插值点取得不好, 并不是插值点取得越多越好, 如果插值点取得不合适, Runge 现象会随着插值点的增加而越来越严重的。这个定理与Faber 1914 年得到的一个结果有关, Faber 的定理表明对于任何一组格点, 至少存在一个连续函数 f , 使得它的插值多项式不能一致收敛到 f 。

虽然知道了插值点是什么样的点, 但是一般来说, 如何选择插值点还是一件困难的事, 只在一些特殊的问题中可以计算出插值点的位置, 所以通常情况下用的是一种近似的最佳逼近式。式(10) 表明, 若要减小插值的误差, 就要选取合适的插值点使得 $\omega_{N+1}^X(x) = \prod_{i=0}^N (x - x_i)$ 最小。 $\omega_{N+1}^X(x)$ 是一个首项是系数是一的 $N+1$ 次多项式, 而误差最小为0, 于是问题转化为寻找一个首项系数为1的与0的误差最小的多项式的问题, 即最小零偏差问题。

现在暂时考虑如下问题, 如何用 P_{N-1} 中的多项式来逼近 x^N 。

$$\|x^N - \sum_{k=0}^{N-1} a_k x^k\|_\infty = \min \quad (13)$$

这个问题和最小零偏差问题是同一问题。根据Chebyshev 定理对最佳逼近多项式的刻画, 这个 $N+1$ 次多项式在 $[-1, 1]$ 上有 $N+1$ 个点符号交错, 并轮流到达最大值最小值。这和三角函数 $\cos nx$ 是一样的, 并且 $\cos nx$ 可以展开成 $\cos x$ 的乘幂的多项式。于是得到

$$T_n(x) = C \cos(n \arccos x) \quad (14)$$

选取适当的 $C = \frac{1}{2^{n-1}}$, 得到 $\tilde{T}_n = \frac{1}{2^{n-1}} \cos(n \arccos x)$ 是首项为1的与0偏差最小的多项式。于是在选取插值点时, 可以选取 T_n 的零点作为插值点, 这样 R 取极值。

4 Tau 方法

Tau 方法的思路是选定一组基, 将微分算子化为函数空间中的矩阵, 将分方程近似化为代数方程

$$Lu(x) = S(x) \quad x \in [-1, 1] \quad (15)$$

被转化为

$$\sum_{j=0}^N L_{ij} \tilde{u}_j \quad (16)$$

令 $R = Lu - S$ 则在选定的基底下要求 $(R, \xi_k) = 0, \forall k \leq N$

从Chebyshev 多项式的求导公式可以很容易的得到一次导数, 二次导数, 以及乘以 x 的矩阵表示。

对于乘以 x , $b_n = \frac{1}{2} [(1 + \delta_{0n-1}) a_{n-1} + a_{n+1}]$

对于一次导数 $b_n = \frac{2}{1+\delta_{0n}} \sum_{p=n+1, p+n \text{ odd}}^N p a_p$

对于二次导数 $b_n = \frac{1}{1+\delta_{0n}} \sum_{p=n+2, p+n \text{ even}} p(p^2 - n^2) a_p$

考虑一个具体的微分方程 $\frac{d^2 u}{dx^2} - 4 \frac{du}{dx} + 4u = \exp(x) - \frac{4e}{1+x^2}$, 边界条件 $u(x = -1) = 0$ 和 $u(x = 1) = 0$

$$D = \begin{pmatrix} 0 & 1 & 0 & 3 & 0 & 5 & 0 & 7 & \dots \\ 0 & 0 & 4 & 0 & 8 & 0 & 12 & 0 & \dots \\ 0 & 0 & 0 & 6 & 0 & 10 & 0 & 14 & \dots \\ 0 & 0 & 0 & 0 & 8 & 0 & 12 & 0 & \dots \\ 0 & 0 & 0 & 0 & 0 & 10 & 0 & 14 & \dots \\ 0 & 0 & 0 & 0 & 0 & 0 & 12 & 0 & \dots \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 14 & \dots \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \dots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots \end{pmatrix}$$

$$D^2 = \begin{pmatrix} 0 & 0 & 4 & 0 & 32 & 0 & 108 & 0 & \dots \\ 0 & 0 & 0 & 24 & 0 & 120 & 0 & 336 & \dots \\ 0 & 0 & 0 & 0 & 48 & 0 & 192 & 0 & \dots \\ 0 & 0 & 0 & 0 & 0 & 80 & 0 & 280 & \dots \\ 0 & 0 & 0 & 0 & 0 & 0 & 120 & 0 & \dots \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 168 & \dots \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \dots \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \dots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots \end{pmatrix}$$

$$I = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \dots \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & \dots \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & \dots \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & \dots \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & \dots \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & \dots \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & \dots \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & \dots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots \end{pmatrix}$$

$$P_N D = \begin{pmatrix} 0 & 1 & 0 & 3 & 0 & 5 & 0 & 7 \\ 0 & 0 & 4 & 0 & 8 & 0 & 12 & 0 \\ 0 & 0 & 0 & 6 & 0 & 10 & 0 & 14 \\ 0 & 0 & 0 & 0 & 8 & 0 & 12 & 0 \\ 0 & 0 & 0 & 0 & 0 & 10 & 0 & 14 \\ 0 & 0 & 0 & 0 & 0 & 0 & 12 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

$$P_N D^2 = \begin{pmatrix} 0 & 0 & 4 & 0 & 32 & 0 & 108 & 0 \\ 0 & 0 & 0 & 24 & 0 & 120 & 0 & 336 \\ 0 & 0 & 0 & 0 & 48 & 0 & 192 & 0 \\ 0 & 0 & 0 & 0 & 0 & 80 & 0 & 280 \\ 0 & 0 & 0 & 0 & 0 & 0 & 120 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 168 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

$$P_N I = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

$L_N = P_N D^2 - 4P_N D + 4P_N I$ 就是在 N 维自空间中微分算子的表示。可以看出这个微分算子不满秩，并且秩为 $N - 2$ ，代表微分方程有 2 个基础解。当将方程右边的非其次项也用 Chebyshev 多项式展开并阶段 N 项后可以求得这个微分方程的通解。加上边界条件就可以得到微分方程的特解了。微分方程的边界条件放在 L_N 矩阵的最后两行

$$\begin{pmatrix} 4 & -4 & 4 & -12 & 32 & -20 & 108 & -28 \\ 0 & 4 & -16 & 24 & -32 & 120 & -48 & 336 \\ 0 & 0 & 4 & -24 & 48 & -40 & 192 & -56 \\ 0 & 0 & 0 & 4 & -32 & 80 & -48 & 280 \\ 0 & 0 & 0 & 0 & 4 & -40 & 120 & -56 \\ 0 & 0 & 0 & 0 & 0 & 4 & -48 & 168 \\ T_0(-1) & T_1(-1) & T_2(-1) & T_3(-1) & T_4(-1) & T_5(-1) & T_6(-1) & T_7(-1) \\ T_0(1) & T_1(1) & T_2(1) & T_3(1) & T_4(1) & T_5(1) & T_6(1) & T_7(1) \end{pmatrix} \begin{pmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \\ a_4 \\ a_5 \\ a_6 \\ a_7 \end{pmatrix} = \begin{pmatrix} -2\sqrt{2}e + J_0(1) \\ J_1(1) \\ (-8 + 6\sqrt{2})e + J_2(1) \\ J_3(1) \\ (48 - 34\sqrt{2})e + 25J_0(1) - 56J_1(1) \\ 49J_1(1) - 204J_2(1) \\ 0 \\ 0 \end{pmatrix}$$

经过计算可以得到解，画出图形为

5 配置法(拟谱法, 伪谱法)

配置法的思路是在区域内选择一些点，将非其次项表示出来，并选定一种基函

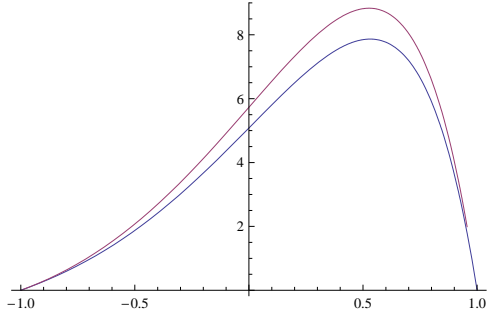


Figure 4: Tau 方法的图形

数，最后所求是解在这组基函数上的展开系数。

$$Lu(x_n) = S(x_n), \forall n \leq N \quad (17)$$

而可以选定一种基函数将方程改写为

$$\sum_{i=0}^N \sum_{j=0}^N L_{ij} \tilde{u}_j T_i(x_n) = S(x_n), \forall n \leq N \quad (18)$$

于是原问题又化为了一个代数方程组求解问题。依旧解上面的例子。

$$D = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 3 & 0 & 5 & 0 & 7 \\ 0 & 0 & 4 & 0 & 8 & 0 & 12 & 0 \\ 0 & 0 & 0 & 6 & 0 & 10 & 0 & 14 \\ 0 & 0 & 0 & 0 & 8 & 0 & 12 & 0 \\ 0 & 0 & 0 & 0 & 0 & 10 & 0 & 14 \\ 0 & 0 & 0 & 0 & 0 & 0 & 12 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

$$D^2 = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 4 & 0 & 32 & 0 & 108 & 0 \\ 0 & 0 & 0 & 24 & 0 & 120 & 0 & 336 \\ 0 & 0 & 0 & 0 & 48 & 0 & 192 & 0 \\ 0 & 0 & 0 & 0 & 0 & 80 & 0 & 280 \\ 0 & 0 & 0 & 0 & 0 & 0 & 120 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 168 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

$$I = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

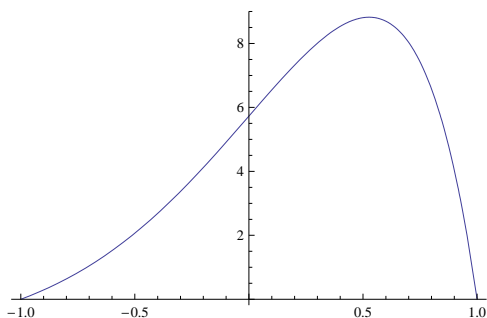


Figure 5: 配置法给出的解

$$L = D^2 - 4D + 4I = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 4 & -4 & 4 & -12 & 32 & -20 & 108 & -28 \\ 0 & 4 & -16 & 24 & -32 & 120 & -48 & 336 \\ 0 & 0 & 4 & -24 & 48 & -40 & 192 & -56 \\ 0 & 0 & 0 & 4 & -32 & 80 & -48 & 280 \\ 0 & 0 & 0 & 0 & 4 & -40 & 120 & -56 \\ 0 & 0 & 0 & 0 & 0 & 4 & -48 & 168 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

选取 $T_6(x)$ 的零点 $x_i = \frac{2i+1}{2N+1}\pi$ 其中 $i = 0, 1, 2 \dots 5$

$$T = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & T_0(x_0) & T_1(x_0) & T_2(x_0) & T_3(x_0) & T_4(x_0) & T_5(x_0) & 0 \\ 0 & T_0(x_1) & T_1(x_1) & T_2(x_1) & T_3(x_1) & T_4(x_1) & T_5(x_1) & 0 \\ 0 & T_0(x_2) & T_1(x_2) & T_2(x_2) & T_3(x_2) & T_4(x_2) & T_5(x_2) & 0 \\ 0 & T_0(x_3) & T_1(x_3) & T_2(x_3) & T_3(x_3) & T_4(x_3) & T_5(x_2) & 0 \\ 0 & T_0(x_4) & T_1(x_4) & T_2(x_4) & T_3(x_4) & T_4(x_4) & T_5(x_2) & 0 \\ 0 & T_0(x_5) & T_1(x_5) & T_2(x_5) & T_3(x_5) & T_4(x_5) & T_5(x_2) & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

记 $\tilde{u} = (\tilde{u}_0, \tilde{u}_1 \dots, \tilde{u}_7)^T$

边界条件要单独给出，而不是微分方程的外推，在配置法下，就是将 x_0, x_N 这两点有关的条件用边界条件来给出。

最后就可以解得方程的解，图像为：

6 Galerkin 方法

Galerkin 方法选取的每个基函数要求各自满足边界条件，于是最后就不需要再将边界条件作为额外的方程附加入求解的方程组了。当然，这样做有损失那就是基函数有可能不正交了。

7 参考书

1. 数值逼近. 蒋尔雄, 赵风光编著. 复旦大学出版社.1996.
2. 现代数值分析. 蔺小林, 蒋耀林编著. 国防工业出版社. 2004.

3. 数学分析新讲(第三册). 张筑生. 北京大学出版社.1991.