《数值分析》之

函数逼近

徐岩

中国科学技术大学数学系

yxu@ustc.edu.cn

http://staff.ustc.edu.cn/~yxu/

概念

- 实际中，函数$f(x)$多样，复杂，通常只能观测到一些离散数据；或者函数$f(x)$过于复杂而难以运算。这时我们要用近似函数$\varphi(x)$来逼近$f(x)$。
- 函数逼近：在科学计算中用到大量复杂函数，对它们直接进行微分、积分等计算不是很容易，而插值是一种最简单的逼近方法。
- 在计算中函数通常是用离散点表示的，特别是函数来自于微分方程的数值解。插值是复原函数的直接方法。
- 在计算机图形学中需要处理由大量离散点表示的曲线和曲面，这称为曲线和曲面拟合(fitting)，其核心步骤就是函数插值。

发展概况

- 单变量函数插值：从十八世纪开始，单变量多项式插值就得到了系统发展，Lagrange, Hermite, Newton等人都有重要贡献。至今已相当成熟，出现了多项式插值、三角函数插值、样条插值，并且催生了如样条函数理论等在现代大规模科学计算和计算机辅助设计中扮演主要角色的理论体系
- 多变量函数插值：至今还处于发展阶段。比较成熟的是张量积形式的插值理论，而其它形式的插值以及多变量样条理论还不是很完善

插值函数

定义

$f(x)$ 为定义在区间$[a,b]$上的函数，$x_0,x_1,\cdots,x_n$为区间上n+1个互不相同的点，Φ为给定的某一函数类。求Φ上的函数$\varphi(x)$,满足

$$\varphi(x_i)=f(x_i), i=0,1,\cdots,n$$

则称$\varphi(x)$为$f(x)$关于节点$x_0,x_1,\cdots,x_n$在Φ上的插值函数。称$x_0,x_1,\cdots,x_n$为插值节点，称$(x_i,f(x_i))$为插值点。

- 插值函数是否存在唯一?
- 如何构造插值函数?
- 插值函数的误差如何估计?

中国科学技术大学

代数多项式

- 易于运算
- 无穷光滑
- 易于计算积分和导数
- 通常选择代数多项式作为插值基函数

Theorem

若x_i两两不同，则对任意给定的y_i, 存在唯一的次数至多是n次的多项式p_n使得$p_n(x_i) = y_i$, $i = 0, \ldots, n$.

证明：在幂基$\{1, x, \ldots, x^n\}$下待定多项式p的形式为

$$p(x) = a_0 + a_1 x + a_2 x^2 + \cdots + a_n x^n$$

由插值条件$p(x_i) = y_i$, $i = 0, \ldots, n$, 得到如下方程组

$$\begin{pmatrix} 1 & x_0 & x_0^2 & \cdots & x_0^n \\ 1 & x_1 & x_1^2 & \cdots & x_1^n \\ 1 & x_2 & x_2^2 & \cdots & x_2^n \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & x_n & x_n^2 & \cdots & x_n^n \end{pmatrix} \begin{pmatrix} a_0 \\ a_1 \\ a_2 \\ \vdots \\ a_n \end{pmatrix} = \begin{pmatrix} y_0 \\ y_1 \\ y_2 \\ \vdots \\ y_n \end{pmatrix}$$

系数矩阵为Vandermonde矩阵，其行列式非零，因此方程组有唯一解。

- 插值矩阵方法(待定系数法)：不具有实用价值，但具有很高的理论价值
- 注意Vandermonde矩阵是病态的，因此不推荐应用该方法计算插值多项式的形式。
- 对于给定的问题，插值多项式存在唯一。但是可以用不同的方法给出插值多项式的不同表示形式。

不同形式的插值多项式

- Lagrange插值
- Newton插值
- Hermite插值

Lagrange插值

- Lagrange基函数: 由多项式插值定理存在函数$\ell_i(x)$满足$\ell_i(x_j) = \delta_{ij}$. 实际上，

$$\ell_i(x) = \prod_{\substack{j=0\\ j\neq i}}^{n} \frac{x - x_j}{x_i - x_j}$$

- Lagrange插值多项式：

$$L_n(x) = \sum_{k=0}^{n} y_k \ell_k(x)$$

- 如果插值节点相同，给定的数据点$\{y_i\}$有多组，那么采用Lagrange插值是相当有效的。但求值算法不是很直接。
- Lagrange插值的缺点: 无承袭性。增加一个节点，所有的基函数都要重新计算

算法

- 计算Lagrange基函数

$$\ell_i(x) = \prod_{\substack{j=0\\ j\neq i}}^{n} \frac{x - x_j}{x_i - x_j}$$

```
fx=0.0
for(i=0;i<=n;i++) {
    tmp=1.0;
    for(j=0;j<i;j++)
        tmp=tmp*(x-x[j])/(x[i]-x[j]);
    for(j=i+1;j<=n;j++)
        tmp=tmp*(x-x[j])/(x[i]-x[j]);
    fx=fx+tmp*y[i];
}
return fx;
```

多项式插值误差定理

Theorem

设$f \in C^{n+1}[a,b]$，多项式p是f在不同结点$x_0, x_1, \ldots, x_n$上的插值多项式，$\deg p \leqslant n$。则对$[a,b]$中每个x，都有$\xi_x \in (a,b)$使得

$$f(x) - p(x) = \frac{1}{(n+1)!} f^{(n+1)}(\xi_x) \prod_{i=0}^{n} (x - x_i)$$

证明：当x与某个结点重合时结论成立。对其它情形，固定x，令

$$\phi(t) = f(t) - p(t) - \frac{f(x) - p(x)}{w(x)} w(t), w(t) = \prod_{i=0}^{n} (t - x_i)$$

则$\phi(t)$在$[a,b]$内有$n+2$个零点，从而存在$\xi_x \in (a,b)$使得$\phi^{(n+1)}(\xi_x) = 0$，即

$$0 = f^{(n+1)}(\xi_x) - (n+1)! \frac{f(x) - p(x)}{w(x)}.$$

UNIVERSITY OF SCIENCE & TECHNOLOGY OF CHINA

中国科学技术大学

- 可以应用于基于多项式构造的各种近似算法(如数值积分)的误差分析
- 建立插商与导数之间的关系：在结点确定的区间内存在一点ξ使得

$$f[x_0,\ldots,x_n]=\frac{f^{(n)}(\xi)}{n!}$$

Lagrange插值多项式性质

当$f(x) = x^k, k = 0, 1, \cdots, n$关于节点$x_0, x_1, \ldots, x_n$上的Lagrange插值多项式就是其本身，因此Lagrange基函数$\ell_i(x)$满足

$$L_n(x) = \sum_{i=0}^{n} \ell_i(x) x_i^k = x^k, k = 0, 1, \cdots, n$$

令$k = 0$,得到

$$\sum_{i=0}^{n} \ell_i(x) = 1$$

多项式插值事后误差估计

- 给定$x_0, x_1, \cdots, x_{n+1}$
- 取$x_0, x_1, \cdots, x_n$，构造$L_n(x)$
- 取$x_1, x_2, \cdots, x_{n+1}$，构造$\widetilde{L}_n(x)$

$$f(x) - L_n(x) = \frac{f^{(n+1)}(\xi_1)}{(n+1)!}(x-x_0)(x-x_1)\cdots(x-x_n)$$

$$f(x) - \widetilde{L}_n(x) = \frac{f^{(n+1)}(\xi_2)}{(n+1)!}(x-x_1)(x-x_2)\cdots(x-x_{n+1})$$

- 近似$f^{(n+1)}(\xi_1) \approx f^{(n+1)}(\xi_2)$,有

$$\frac{f(x) - L_n(x)}{f(x) - \widetilde{L}_n(x)} \approx \frac{x-x_0}{x-x_{n+1}}$$

$$\Longrightarrow f(x) - L_n(x) \approx \frac{x-x_0}{x-x_{n+1}}\left(L_n(x) - \widetilde{L}_n(x)\right)$$

- 插值误差可以用2组插值函数的差来估计

上机作业1

对函数

$$f(x) = \frac{1}{1+x^2}, x \in [-5, 5]$$

构造Lagrange插值多项式$p_L(x)$，插值节点取为:

1. $x_i = 5 - \frac{10}{N} i, \;\; i = 0, 1, \cdots, N$
2. $x_i = -5\cos(\frac{2i+1}{2N+2}\pi), \;\; i = 0, 1, \cdots, N \quad$ (Chebyshev point)

并计算如下误差

$$\max_i \{|f(y_i) - p(y_i)|, \; y_i = \frac{i}{10} - 5, \; i = 0, 1, \cdots, 100\}$$

对$N = 5, 10, 20, 40$比较以上两组节点的结果。

输出形式如下：

N=5

Max Error of grid (1)：XXXXXXXXXXXXXXXXXX

Max Error of grid (2)：XXXXXXXXXXXXXXXXXX

算法

- 计算Lagrange基函数

$$\ell_i(x) = \prod_{\substack{j=0 \\ j \neq i}}^{n} \frac{x - x_j}{x_i - x_j}$$

```
fx=0.0
for(i=0;i<=n;i++) {
    tmp=1.0;
    for(j=0;j<i;j++)
        tmp=tmp*(x-x[j])/(x[i]-x[j]);
    for(j=i+1;j<=n;j++)
        tmp=tmp*(x-x[j])/(x[i]-x[j]);
    fx=fx+tmp*y[i];
}
return fx;
```