3.5数值微分

3.5 数值微分

学习目标:
掌握几个数值微分计算公式。

3.5数值微分

数值微分就是用离散方法即使的近似地求出函数在某点的导数值.按照**Taylor**展开原理可得

$$f'(x)=\frac{f(x+h)-f(x)}{h}+O(h),$$

$$f'(x)=\frac{f(x)-f(x-h)}{h}+O(h),$$

$$f'(x)=\frac{f(x+h)-f(x-h)}{2h}+O(h),$$

其中**h**为一增量。上面几个公式是很实用的，下面我们再讨论一些常用方法。

3.5.1 插值型求导公式

设f（x）是定义在[a，b]上的函数，并给定区间[a，b]上的函数，

并给定区间[a，b]上的n+1个节点 x_k 出的函数值 $f(x_k), k=0,1,\cdots n.$

这样,我们可以建立函数 $f(x)$ 的n次插值多项式 $P_n(x)$

多项式的求导是容易的,称

$$f'(x) \approx P'_n(x) \qquad (3.5.1)$$

为插值型求导公式。

应当指出，即使 $f(x)$ 和 $P_n(x)$ 的值相差不多，导数的近似值 $P_n'(x)$ 与导数的值 $f_x'(x)$ 仍然可能相差很大。因而在使用求导公式（**3.5.1**）时，应注意误差的分析。

依据插值余项定理，求导公式（**3.5.1**）的余项为

$$f'(x)-P_n'(x)=\frac{f^{(n+1)}(\xi)}{(n+1)!}\omega_{n+1}'(x)+\frac{\omega_{n+1}(x)}{(n+1)!}\frac{d}{dx}f^{(n+1)}(\xi),$$

式中 $\omega_{n+1}(x)=(x-x_0)(x-x_1)\cdots(x-x_n)$.

在上述余项公式中，由于 ξ 是**x**的未知函数，我们无法对右端的第二项作出进一步的说明。因此，对于随意给出的点**x**，求导公式的余项是很难估计的。

然而，如果我们限定求节点上的导数值，那么有余项公式

$$f'(x_k)-P_n'(x_k)=\frac{f^{(n+1)}(\xi)}{(n+1)!}\omega_{n+1}'(x_k). \qquad （3.5.2）$$

下面我们考察节点处的导数值。为简化讨论，假定所给的节点是等距的，

h是步长。

1.两点公式

当**n=1**时，由（**3.5.2**）得带余项的两点公式

$$f'(x_0)=\frac{1}{h}[f(x_1)-f(x_0)]-\frac{h}{2}f''(\xi), \qquad （3.5.3）$$

$$f'(x_0)=\frac{1}{h}[f(x_1)-f(x_0)]+\frac{h}{2}f''(\xi). \qquad （3.5.4）$$

2. 三点公式

当**n=2**时，由（**3.5.2**）的带余项的三点公式

$$f'(x_0)=\frac{1}{2h}[-3f(x_0)+4f(x_1)-f(x_2)]+\frac{h^2}{5}f''(\xi), \qquad (3.5.5)$$

$$f'(x_1)=\frac{1}{2h}[-f(x_0)+f(x_2)]-\frac{h^2}{6}f''(\xi), \qquad (3.5.6)$$

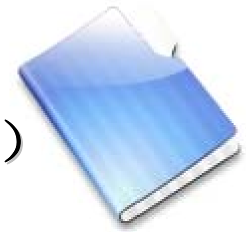
$$f'(x_2)=\frac{1}{2h}[f(x_0)-4f(x_1)+f(x_2)]+\frac{h^2}{3}f''(\xi), \qquad (3.5.7)$$

3. 五点公式

当**n=4**时，由（**3.5.2**）不难导出带余项的五点求导公式。这里给出

其中常用五点公式

$$f'(x_2)=\frac{1}{12h}[f(x_0)-8f(x_1)+f(x_3)-f(x_4)]+\frac{h^2}{30}f''(\xi), \qquad (3.5.8)$$



例 3.9 设 $f(x)=e^{x}$，对h=0.01，计算 $f'(1.8)$的近似值。

解 由（**3.5.5**）式有

$$f'(1.8)\approx\frac{1}{2h}\left[-3f(1.8)+4f(1.81)-f(1.82)\right]=6.0494$$

由（**3.5.6**）有

$$f'(1.8)\approx\frac{1}{2h}\left[f(1.81)-f(1.79)\right]=6.0497$$

由（**3.5.7**）式有

$$f'(1.8)\approx\frac{1}{2h}\left[f(1.78)-4f(1.79)+3f(1.82)\right]=6.0494$$

由（**3.5.8**）式有

$$f'(1.8)\approx\frac{1}{2h}\left[f(1.78)-8f(1.79)+f(1.81)-f(1.82)\right]=6.0496$$

精确值 $e^{1.8}=6.0496$ 。计算结果显然与它们的余项相一致，由（**3.5.8**）式计算所得的结果最精确。

用插值多项式 $P_n(x)$ 作为 $f(x)$ 的近似函数，还可以建立高阶数值微分公式

$$f^{k}(x) \approx P_n^{k}(x), k = 1,2\cdots .$$

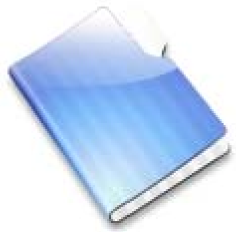
然而，对于用插值法建立的数值求导公式通常导数值的精确度比用插值公式求得的函数值的精确度差，高阶导数值的精度比低阶导数值的精度差。所以，不宜用次方法建立高阶数值求导公式。

3.5.2 三次样条求导

我们知道，三次样条函数**S(x)**作为**f（x）**的近似函数，不但彼此的函数值很接近，导数值也很接近。因此用样条函数建立数值微分公式是很自然的。

设在区间**[a，b]**上，给定一种划分 $a=x_0<x_1\cdots<x_n=b,$ 及相应的函数值 $h_k=x_{k+1}-x_k$ 再给定适当的边界条件，按三次样条函数的算法，建立关于节点上的一阶导数 m_k 或二节导数 M_k 的样条方程组。求得 m_k 或 $M_k, k=0,1,\cdots,n$ 从而得到三次样条插值函数**S(x)**的表达式。这样，可得数值微分的公式

$$f^{(i)}(x)\approx S^{(i)}(x), i=0,1,2$$



与前面插值型数值微分公式不同，样条数值微分公式（**3.5.9**）可以用来计算插值范围内任何一点（不仅是节点）上的导数值。误差估计由（**2.3.21**）给出。

对节点上的导数值，若求得的是 $m_k, k=0,1,\cdots,n$ 则由**S(x)**的表达式有

$$f'(x_k) \approx m_k,$$

$$f''(x_k) \approx S''(x_k) = -\frac{2}{h_k}(2m_k + m_{k+1}) + \frac{6}{h_k} f[x_k, x_{k+1}].$$

若求得的是 $M_k, k=0,1,\cdots,n$ 则由**S(x)**的表达式有

$$f'(x_k) \approx S'(x_k) = -\frac{h_k}{6}(2M_k + M_{k+1}) + f[x_k, x_{k+1}].$$

$$f''(x_k) \approx M_k,$$

3.5.3 数值微分的外推算法

先看一个简单的例子。求 $f(x)=-\cot x$ 在**x=0.004**出的一阶导数值。采用中点微分公式（**3.5.6**），即 $f'(x)\approx\dfrac{f(x+h)-f(x-h)}{2h}$

取**h=0.0016**，那么得 $f'(0.004)\approx 626.3350438$ 而 $f'(0.004)=625.33344002$ 由此看见，仅有两位有效数字。利用**Richardson**外推法可以提高计算精度。

对于中心差商，记 $f'(x)\approx G(h)=\dfrac{1}{2h}[f(x+h)-f(x-h)].$

由**Taylor**级数展开有

$$f'(x)-G(h)=f'(x_n)+\frac{h^2}{6}f'''(x_n)+\frac{h^4}{120}f^{(5)}(x_5)+\cdots$$

利用**Richardson**外推公式，取$q=2, P_k=2k,$ 则有

$$\begin{cases} G(h)=G_1(h), \\ G_{m+1}(h)\dfrac{4^m G_m(h/2)-G_m(h)}{4^m-1}, m=1,2,\cdots. \end{cases}$$

外推公式（**3.5.9**）的终止标准是$\left|G_{m+1}(h)-G_m(h/2)\right|<\varepsilon, \varepsilon$ 是预先给定的误差小量。

例 3.10 设 $f(x)=x^2e^{-x}$ 设h分别取0.1,0.05,0.025时求出x=0.5出的一阶导数的中心差商，进行外推，并与精确值进行比较。

解 先分别取**h=0.1,0.05,0.025,**求出节点**x=0.5**处的中心差商值，见表**3-6**，再按（**3.5.9**）式进行外推，外推两次，结果列于表**3-6**中。从表**3-6**可见，**h=0.025**时的中心差商值只有**3**位有效数字，外推一次达到**5**位有效数字，外推两次达到**9**位有效数字。

表3-6

h	$G_1(h)$	$G_2(h)$	$G_3(h)$	$G_4(h)$
0.1	0.4516049081	0.4548999231	0.454897994	0.454897994
0.05	0.4540761693	0.4548981152		
0.025	0.4546926288			