《数值分析》之

数值微分

徐岩

中国科学技术大学数学系

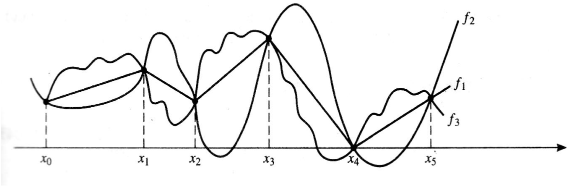
yxu@ustc.edu.cn

https://faculty.ustc.edu.cn/yxu

内容

1. 数值微分：基于Taylor展开、基于多项式插值、Richardson外推
2. 数值积分：通过多项式插值、待定系数法、复化数值积分、Romberg积分、Guass积分

- 只给定函数f在$n+1$个点$x_0,\ldots,x_n$上的值，如何利用这些信息计算导数$f'(c)$和积分$\int_a^b f(x)dx$的值？注意过这些点的函数可以有很多，如下图所示：



基于Taylor展开计算数值微分

- Taylor展开：

$$f(x+h) = f(x) + hf'(x) + \frac{h^2}{2}f''(\xi)$$

其中$\xi \in (x, x+h)$. 为了使等式成立，需要f与f'在闭区间$[x, x+h]$上连续，而且对应的开区间上存在f''

- 重新整理：

$$f'(x) = \frac{1}{h}[f(x+h) - f(x)] - \frac{h}{2}f''(\xi)$$

因此可利用的数值公式与误差项都出现在上述公式中。误差项由两部分组成：h的幂次以及f的高阶导数。因此当$h \to 0$时，误差中的h项使得整体表达式收敛于零

- $-(h/2)f''(\xi)$称为截断误差。在数值计算中，截断误差与舍入误差起着同样重要的作用

例

应用上述公式计算$f(x)=\cos x$在$x=\pi/4$点的导数，这里取$h=0.01$。它的精确度是多少?

- 数值导数值

$$\begin{aligned} f'(x) &\approx \frac{1}{h}[f(x+h)-f(x)] \\ &= \frac{1}{0.01}[0.700000476-0.707106781] \\ &= -0.71063051 \end{aligned}$$

- 精度：

$$\left|\frac{h}{2}f''(\xi)\right| = 0.005|\cos\xi| \leqslant 0.005$$

- 实际上，由于$\xi\in(\pi/4,\pi/4+h)$, 所以$|\cos\xi|<0.707107$, 从而给出误差界为0.0035355.
- 真正的误差为

$$-\sin\frac{\pi}{4}+0.71063051=0.003523729$$

精度分析

- 在前面的数值导数计算公式中，从截断误差$-(h/2)f''(\xi)$的表达式可见：为了精确计算$f'(x)$, 步长h必须很小
- 下面进行一个实验，其中令h通过给定的一个序列收敛到零，分别计算出相应的$f'(x)$的近似值。这里$f(x) = \arctan x$, $x = \sqrt{2}$. 精确结果应当是$f'(x) = 1/(1+x^2)$在$x = \sqrt{2}$点的值$1/3$
    - 运行Mathematica程序“数值微分_Taylor展开.nb”，通过改变其中的m以得到不同的效果

```
In[1]:= f[x_] := ArcTan[x]; m = 8; s = N[Sqrt[2], m]; h = 1; M = 26; F1 = N[f[s], m];
        For[k = 0, k <= M, k++, F2 = N[f[s + h], m]; d = N[F2 - F1, m]; r = N[d / h, m];
         Print[k, "\t", h, "\t", F2, "\t", F1, "\t", d, "\t", r]; h = N[h / 2, m]]
```

0	1	1.1780972	0.95531662	0.2227806	0.2227806
1	0.50000000	1.0893836	0.95531662	0.1340670	0.2681340
2	0.25000000	1.0297268	0.95531662	0.0744102	0.2976406
3	0.12500000	0.99464439	0.95531662	0.0393278	0.314622
4	0.062500000	0.97555095	0.95531662	0.0202343	0.323749
5	0.031250000	0.96558170	0.95531662	0.0102651	0.328483
6	0.015625000	0.96048682	0.95531662	0.0051702	0.33089
7	0.0078125000	0.95791122	0.95531662	0.0025946	0.33211
8	0.0039062500	0.95661631	0.95531662	0.0012997	0.33272
9	0.0019531250	0.95596706	0.95531662	0.0006504	0.33303
10	0.00097656250	0.95564199	0.95531662	0.0003254	0.3332
11	0.00048828125	0.95547934	0.95531662	0.0001627	0.3333
12	0.00024414063	0.95539799	0.95531662	0.0000814	0.3333
13	0.00012207031	0.95535731	0.95531662	0.0000407	0.333
14	0.000061035156	0.95533696	0.95531662	0.0000203	0.333
15	0.000030517578	0.95532679	0.95531662	0.0000102	0.333
16	0.000015258789	0.95532170	0.95531662	5.1×10^{-6}	0.33
17	7.6293945×10^{-6}	0.95531916	0.95531662	2.5×10^{-6}	0.33
18	3.8146973×10^{-6}	0.95531789	0.95531662	1.3×10^{-6}	0.33
19	1.9073486×10^{-6}	0.95531725	0.95531662	$6.\times10^{-7}$	0.3
20	9.5367432×10^{-7}	0.95531694	0.95531662	$3.\times10^{-7}$	0.3
21	4.7683716×10^{-7}	0.95531678	0.95531662	$2.\times10^{-7}$	0.3
22	2.3841858×10^{-7}	0.95531670	0.95531662	$0.\times10^{-8}$	0.3
23	1.1920929×10^{-7}	0.95531666	0.95531662	$0.\times10^{-8}$	$0.\times10^{-1}$
24	5.9604645×10^{-8}	0.95531664	0.95531662	$0.\times10^{-8}$	$0.\times10^{-1}$
25	2.9802322×10^{-8}	0.95531663	0.95531662	$0.\times10^{-8}$	$0.\times10^{-1}$
26	1.4901161×10^{-8}	0.95531662	0.95531662	$0.\times10^{-8}$	0.

精度的减法相消

- 从实验中可以看到，当h趋向于零时，d的有效数字逐渐减少，直到最后$d = 0$, $r = 0$, 因此并没有使得数值导数的精度越来越高
- 当运算中字长为8位，那么在$k = 11, 12$时得到最佳的结果0.33330000. 此时$d = f(x+h) - f(x)$有四位有效数字，随着k的增加，d中有效数字的个数在减少，而$r = d/h$的有效数字个数不会比d的更多。因此当h很小时，舍入误差使得当h趋向于零时不能得到高的精度
- 当然，如果计算过程中字长变大，那么得到的精度就会更高

高阶公式

- 无论如何，前面给出公式的误差估计只是h的一次方，我们可以通过对Taylor展开进行简单的处理，得到更高阶的数值导数计算公式
- 实际上，由于

$$f(x+h) = f(x) + hf'(x) + \frac{h^2}{2}f''(x) + \frac{h^3}{3!}f'''(\xi_1)$$
$$f(x-h) = f(x) - hf'(x) + \frac{h^2}{2}f''(x) - \frac{h^3}{3!}f'''(\xi_2)$$

所以两式相减得到

$$f'(x) = \frac{1}{2h}\left[f(x+h) - f(x-h)\right] - \frac{h^2}{12}\left[f'''(\xi_1) + f'''(\xi_2)\right]$$

高阶公式的误差估计

- 只要f'''存在，前面公式的误差项是可用的
- 进一步，根据导数的介值定理[1]，存在一点$\xi \in (x-h, x+h)$，$f'''(\xi) = (f'''(\xi_1) + f'''(\xi_2))/2$。因此前页的数值微分公式可以重写为

$$f'(x) = \frac{1}{2h}\left[f(x+h) - f(x-h)\right] - \frac{h^2}{6} f'''(\xi)$$

[1]若f是$[a,b]$上的实值可微函数，再设$f'(a) < \lambda < f'(b)$，那么存在一点$x \in (a,b)$，$f'(x) = \lambda$.

例

设$f(x) = \arctan x$, $x = \sqrt{2}$, 用高阶公式重新计算$f'(x)$的值。正确值为$1/3$

- 通过数值实验，此时在$k = 8, 9, 10$时得到最高的五位精度
- 减法相消现象仍然存在，即当$k > 10$后，舍入误差使得精确位数在减少

```
In[2]:= f[x_] := ArcTan[x]; m = 8; s = N[Sqrt[2], m]; h = 1; M = 26;
For[k = 0, k <= M, k++, F2 = N[f[s + h], m]; F1 = N[f[s - h], m]; d = N[F2 - F1, m];
 r = N[d / 2 / h, m]; Print[k, "\t", h, "\t", F2, "\t", F1, "\t", d, "\t", r]; h = N[h / 2, m]]
```

0	1	1.1780972	0.3926991	0.7853982	0.3926991
1	0.50000000	1.0893836	0.7406126	0.3487710	0.3487710
2	0.25000000	1.0297268	0.86112983	0.1685969	0.3371939
3	0.12500000	0.99464439	0.91106988	0.0835745	0.3342980
4	0.062500000	0.97555095	0.93385414	0.0416968	0.333574
5	0.031250000	0.96558170	0.94474460	0.0208371	0.333394
6	0.015625000	0.96048682	0.95006969	0.0104171	0.333348
7	0.0078125000	0.95791122	0.95270283	0.0052084	0.33334
8	0.0039062500	0.95661631	0.95401213	0.0026042	0.33333
9	0.0019531250	0.95596706	0.95466498	0.0013021	0.33333
10	0.00097656250	0.95564199	0.95499095	0.0006510	0.33333
11	0.00048828125	0.95547934	0.95515382	0.0003255	0.3333
12	0.00024414063	0.95539799	0.95523523	0.0001628	0.3333
13	0.00012207031	0.95535731	0.95527593	0.0000814	0.3333
14	0.000061035156	0.95533696	0.95529627	0.0000407	0.333
15	0.000030517578	0.95532679	0.95530645	0.0000203	0.333
16	0.000015258789	0.95532170	0.95531153	0.0000102	0.333
17	7.6293945×10^{-6}	0.95531916	0.95531407	5.1×10^{-6}	0.33
18	3.8146973×10^{-6}	0.95531789	0.95531535	2.5×10^{-6}	0.33
19	1.9073486×10^{-6}	0.95531725	0.95531598	1.3×10^{-6}	0.33
20	9.5367432×10^{-7}	0.95531694	0.95531630	$6.\times10^{-7}$	0.3
21	4.7683716×10^{-7}	0.95531678	0.95531646	$3.\times10^{-7}$	0.3
22	2.3841858×10^{-7}	0.95531670	0.95531654	$2.\times10^{-7}$	0.3
23	1.1920929×10^{-7}	0.95531666	0.95531658	$0.\times10^{-8}$	0.3
24	5.9604645×10^{-8}	0.95531664	0.95531660	$0.\times10^{-8}$	$0.\times10^{-1}$
25	2.9802322×10^{-8}	0.95531663	0.95531661	$0.\times10^{-8}$	$0.\times10^{-1}$
26	1.4901161×10^{-8}	0.95531662	0.95531661	$0.\times10^{-8}$	$0.\times10^{-1}$

差商

- 向前差商

$$f'(x_0) \approx \frac{f(x_0+h)-f(x_0)}{h}, \quad R(x) = -\frac{h}{2}f''(\xi) = O(h)$$

- 向后差商

$$f'(x_0) \approx \frac{f(x_0)-f(x_0-h)}{h}, \quad R(x) = \frac{h}{2}f''(\xi) = O(h)$$

- 中心差商

$$f'(x_0) \approx \frac{f(x_0+h)-f(x_0-h)}{2h}, \quad R(x) = -\frac{h^2}{6}f'''(\xi) = O(h^2)$$

数据误差

- 数值微分对于函数值的计算误差或测量误差非常敏感，因为此时函数值的误差被乘以$1/(2h)$
- 因此当计算由有误差数据确定的导数时，需要进行数据光滑化处理（去噪）
  - 即尽可能多得用当前点周围点信息把数据噪音去掉，因此在实际应用中，为了计算数据导数，通常是根据当前点以及周围点进行数据拟合，然后根据拟合出来的表达式计算当前点的导数
- 数值积分公式对数据误差不是很敏感

设定最佳步长（事后估计法）

- 设$D(h), D(h/2)$分别为步长为$h, h/2$的差商公式
- $$\begin{aligned} & f'(x)-D(h)=O(h), \quad f'(x)-D(h/2)=O(h/2) \\ \Rightarrow & \frac{f'(x)-D(h)}{f'(x)-D(h/2)}=\frac{O(h)}{O(h/2)}\approx 2 \\ \Rightarrow & f'(x)-D(h)=2f'(x)-2D(h/2) \\ \Rightarrow & f'(x)-D(h/2)=D(h/2)-D(h) \end{aligned}$$
- 当
$$|D(h)-D(h/2)|<\varepsilon$$
时的步长$h/2$就是合适的步长

高阶导数公式

- 根据高阶的Taylor展开，我们可以得到高阶导数的计算公式。
- 如

$$f(x+h)=f(x)+hf'(x)+\frac{h^2}{2}f''(x)+\frac{h^3}{3!}f'''(x)+\frac{h^4}{4!}f^{(4)}(\xi_1)$$
$$f(x-h)=f(x)-hf'(x)+\frac{h^2}{2}f''(x)-\frac{h^3}{3!}f'''(x)+\frac{h^4}{4!}f^{(4)}(\xi_2)$$

  两式相加，得到

$$f''(x)=\frac{1}{h^2}\left[f(x+h)-2f(x)+f(x-h)\right]-\frac{h^2}{12}f^{(4)}(\xi)$$

  其中$\xi\in(x-h,x+h)$
- 这个公式常用于二阶微分方程的数值求解中

通过多项式插值计算数值导数

- 假设函数f在$x_0, x_1, \ldots, x_n$上的值已知，那么f在这些结点上存在唯一的插值多项式，从而有

$$f(x) = \sum_{k=0}^{n} f(x_i)\ell_i(x) + \frac{1}{(n+1)!} f^{(n+1)}(\xi_x) w(x)$$

其中$\ell_i(x)$为Lagrange插值基函数，$w(x) = \prod_{i=0}^{n}(x - x_i)$

- 对其求导：

$$f'(x) = \sum_{i=0}^{n} f(x_i)\ell_i'(x) + \frac{1}{(n+1)!} f^{(n+1)}(\xi_x) w'(x)$$
$$+ \frac{1}{(n+1)!} w(x) \frac{d}{dx} f^{(n+1)}(\xi_x)$$

- 如果我们是在结点处计算数值导数，即不妨设$x = x_\alpha$，由于$w(x_\alpha) = 0$，则结果得到简化：

$$f'(x_\alpha) = \sum_{i=0}^{n} f(x_i)\ell_i'(x_\alpha) + \frac{1}{(n+1)!} f^{(n+1)}(\xi_{x_\alpha}) w'(x_\alpha)$$

- 而

$$w'(x) = \sum_{i=0}^{n} \prod_{\substack{j=0 \\ j\neq i}}^{n} (x - x_j) \Longrightarrow w'(x_\alpha) = \prod_{\substack{j=0 \\ j\neq \alpha}}^{n} (x_\alpha - x_j)$$

所以带有误差项的数值微分公式为

$$f'(x_\alpha) = \sum_{i=0}^{n} f(x_i)\ell_i'(x_\alpha) + \frac{1}{(n+1)!} f^{(n+1)}(\xi_{x_\alpha}) \prod_{\substack{j=0 \\ j\neq \alpha}}^{n} (x_\alpha - x_j)$$

- 此公式特别适用于非等距结点

例

给出当$n=2, \alpha=1$时上述公式的显式表达

- 此时，三个Lagrange插值基函数为

$$\ell_0(x)=\frac{(x-x_1)(x-x_2)}{(x_0-x_1)(x_0-x_2)}$$
$$\ell_1(x)=\frac{(x-x_0)(x-x_2)}{(x_1-x_0)(x_1-x_2)}$$
$$\ell_2(x)=\frac{(x-x_0)(x-x_1)}{(x_2-x_0)(x_2-x_1)}$$

- 它们的导数分别是

$$\ell_0'(x)=\frac{2x-x_1-x_2}{(x_0-x_1)(x_0-x_2)}$$
$$\ell_1'(x)=\frac{2x-x_0-x_2}{(x_1-x_0)(x_1-x_2)}$$
$$\ell_2'(x)=\frac{2x-x_0-x_1}{(x_2-x_0)(x_2-x_1)}$$

- 计算在$x = x_1$点的值，我们有

$$
\begin{aligned}
\ell_0'(x_1) &= \frac{x_1 - x_2}{(x_0 - x_1)(x_0 - x_2)} \\
\ell_1'(x_1) &= \frac{2x_1 - x_0 - x_2}{(x_1 - x_0)(x_1 - x_2)} \\
\ell_2'(x_1) &= \frac{x_1 - x_0}{(x_2 - x_0)(x_2 - x_1)}
\end{aligned}
$$

- 因而带有误差项的数值微分公式是

$$
\begin{aligned}
f'(x_1) = & f(x_0)\frac{x_1 - x_2}{(x_0 - x_1)(x_0 - x_2)} \\
& + f(x_1)\frac{2x_1 - x_0 - x_2}{(x_1 - x_0)(x_1 - x_2)} \\
& + f(x_2)\frac{x_1 - x_0}{(x_2 - x_0)(x_2 - x_1)} \\
& + \frac{1}{6}f'''(\xi_{x_1})(x_1 - x_0)(x_1 - x_2)
\end{aligned}
$$

例：等距情形

- 在$n = 2, \alpha = 1$时，前面的数值微分公式简化为

$$f'(x) = f(x-h)\frac{-1}{2h} + f(x+h)\frac{1}{2h} - \frac{1}{6}f'''(\xi_x)h^2$$

这就是前面给出的高阶公式

- Richardson外推(extrapolation)技术是通过巧妙应用Taylor级数改进数值微分公式的精度
- 函数$f(x)$的Taylor级数为

$$f(x+h)=\sum_{k=0}^{\infty}\frac{1}{k!}h^k f^{(k)}(x)$$
$$f(x-h)=\sum_{k=0}^{\infty}\frac{1}{k!}(-1)^k h^k f^{(k)}(x)$$

两式相减，消去了所有的偶次项：

$$f(x+h)-f(x-h)=2hf'(x)+\frac{2}{3!}h^3 f'''(x)+\frac{2}{5!}h^5 f^{(5)}(x)+\cdots$$

- 重新整理得：

$$L=\varphi(h)+a_2h^2+a_4h^4+a_6h^6+\cdots$$

其中$L=f'(x)$, $\varphi(h)=\frac{1}{2h}[f(x+h)-f(x-h)]$,
$a_k=-\frac{2}{(k-1)!}f^{(k-1)}(x)$

- 在上述公式中，我们需要$h>0$. 在$h=0$时我们得不到任何信息。对每个$h>0$, $a_2h^2+a_4h^4+\cdots$给出了误差
- 实际上，通过取不同的h，我们可以进一步消去误差项中的低次项。如根据在h和$h/2$的表达式，

$$\begin{aligned} L&=\varphi(h)+a_2h^2+a_4h^4+a_6h^6+\cdots \\ 4L&=4\varphi(h/2)+a_2h^2+a_4h^4/4+a_6h^6/16+\cdots \\ \hline 3L&=4\varphi(h/2)-\varphi(h)-3a_4h^4/4-15a_6h^6/16-\cdots \end{aligned}$$

得到

$$L=\frac{4}{3}\varphi(h/2)-\frac{1}{3}\varphi(h)-a_4h^4/4-5a_6h^6/16-\cdots$$

- 采用公式

$$
\begin{aligned}
L &= \frac{4}{3}\varphi(h/2) - \frac{1}{3}\varphi(h) \\
&= \frac{2}{3h}[f(x+h/2) - f(x-h/2)] - \frac{1}{6h}[f(x+h) - f(x-h)]
\end{aligned}
$$

计算前面例子中的导数

- 当$k = 4, 5, 6, 7$时得到六位数的精度

In[1]:=
```
f[x_] := ArcTan[x]; m = 8; s = N[Sqrt[2], m]; h = 1; M = 30; d = Table[0, {M}]; F1 = N[f[s], m];
For[k = 0, k ≤ M, k++, d[[k]] = N[(f[s + h] - f[s - h]) / 2 / h, m]; h = N[h / 2, m]]; For[k = 1,
 k <= M, k++, r = N[d[[k]] + (d[[k]] - d[[k - 1]]) / 3, m]; Print[k, "\t", d[[k]], "\t", r];]
```

1	0.3487710	0.3341283
2	0.3371939	0.3333348
3	0.3342980	0.3333327
4	0.3335745	0.333333
5	0.333394	0.333333
6	0.333348	0.333333
7	0.333337	0.333333
8	0.33333	0.33333
9	0.33333	0.33333
10	0.33333	0.33333
11	0.33333	0.3333
12	0.3333	0.3333
13	0.3333	0.3333
14	0.3333	0.333
15	0.333	0.333
16	0.333	0.333
17	0.333	0.333
18	0.33	0.33
19	0.33	0.33
20	0.33	0.33
21	0.33	0.3
22	0.3	0.3
23	0.3	0.3
24	0.3	$0.\times 10^{-1}$
25	$0.\times 10^{-1}$	$0.\times 10^{-1}$
26	$0.\times 10^{-1}$	$0.\times 10^{-1}$

进一步外推

- 令
$$\psi(h)=\frac{4}{3}\varphi(h/2)-\frac{1}{3}\varphi(h)$$
那么可以应用$\psi(h)$在h和$h/2$的取值进一步消去低阶项
- 实际上，
$$\begin{aligned} L &= \psi(h)+b_4h^4+b_6h^6+\cdots \\ 16L &= 16\psi(h/2)+b_4h^4+b_6h^6/4+\cdots \\ \hline 15L &= 16\psi(h/2)-\psi(h)-3b_6h^6/4-\cdots \end{aligned}$$
- 从而令
$$\theta(h)=\frac{16}{15}\psi(h/2)-\frac{1}{15}\psi(h)$$
得到
$$L=\theta(h)+c_6h^6+c_8h^8+\cdots$$
- 类似地，
$$L=\frac{64}{63}\theta(h/2)-\frac{1}{63}\theta(h)-\frac{3}{252}c_8h^8-\cdots$$

Richardson外推算法

- 上述过程可以执行任意多步，得到不断增加精度的公式
- 执行M步的Richardson外推算法为
  1. 选取h的一个初值，如$h=1$，并且计算$M+1$个数
  $$D(n,0)=\varphi(h/2^n), \quad n=0,1,\dots,M$$
  2. 执行下列公式的计算
  $$D(n,k)=\frac{4^k}{4^k-1}D(n,k-1)-\frac{1}{4^k-1}D(n-1,k-1)$$
  这里$k=1,2,\dots,M$，$n=k,k+1,\dots,M$
  3. $D(M,M)$就是所需要的结果
- 在上述算法中，$D(0,0)=\varphi(h)$，$D(1,0)=\varphi(h/2)$，$D(1,1)=\psi(h)$

结果的精度

- 根据Richardson外推算法的计算过程,

$$\begin{aligned} D(n,0) &= L + \mathcal{O}(h^2) \\ D(n,1) &= L + \mathcal{O}(h^4) \\ D(n,2) &= L + \mathcal{O}(h^6) \\ D(n,3) &= L + \mathcal{O}(h^8) \end{aligned}$$

- 我们将证明

$$D(n,k-1) = L + \mathcal{O}(h^{2k}), \qquad \text{当 } h \to 0$$

Richardson外推定理

Theorem

在算法中定义的$D(n,k)$满足下列形式的等式

$$D(n,k-1)=L+\sum_{j=k}^{\infty}A_{j,k}(h/2^n)^{2j}$$

证明：当$k=1$时，由$D(n,0)$的定义以及

$$L=\varphi(h)+a_2h^2+a_4h^4+a_6h^6+\cdots$$

可知定理成立：

$$D(n,0)=\varphi(h/2^n)=L-\sum_{j=1}^{\infty}a_{2j}(h/2^n)^{2j}$$

因此可设$A_{j,1}=-a_{2j}$

现在对k进行归纳证明。假设$k-1$时定理成立，那么根据算法中$D(n,k)$的定义以及归纳假设，

$$
\begin{aligned}
D(n,k) &= \frac{4^k}{4^k-1}\left[L+\sum_{j=k}^{\infty} A_{j,k}\left(\frac{h}{2^n}\right)^{2j}\right] \\
&\quad - \frac{1}{4^k-1}\left[L+\sum_{j=k}^{\infty} A_{j,k}\left(\frac{h}{2^{n-1}}\right)^{2j}\right] \\
&= L+\sum_{j=k}^{\infty} A_{j,k}\frac{4^k-4^j}{4^k-1}\left(\frac{h}{2^n}\right)^{2j}
\end{aligned}
$$

从而我们可以定义

$$
A_{j,k+1} = A_{j,k}\frac{4^k-4^j}{4^k-1}
$$

显然$A_{k,k+1}=0$，定理所需要的形式成立。 □

- 在算法中的$D(n,k)$形成如下的三角阵列

$$\begin{array}{ccccc}
D(0,0) & & & & \\
D(1,0) & D(1,1) & & & \\
D(2,0) & D(2,1) & D(2,2) & & \\
\vdots & \vdots & \vdots & \ddots & \\
D(M,0) & D(M,1) & D(M,2) & \cdots & D(M,M)
\end{array}$$

H.W.

编程实现用Richardson外推计算$f'(x)$的值，$h=1$。函数$f(x)$分别取

- $\ln x,\ x=3,\ M=3$
- $\tan x,\ x=\sin^{-1}(0.8),\ M=4$
- $\sin(x^2+\frac{1}{3}x),\ x=0,\ M=5.$

输出相应的三角阵列

$$\begin{array}{ccccc}
D(0,0) & & & & \\
D(1,0) & D(1,1) & & & \\
D(2,0) & D(2,1) & D(2,2) & & \\
\vdots & \vdots & \vdots & \ddots & \\
D(M,0) & D(M,1) & D(M,2) & \cdots & D(M,M)
\end{array}$$