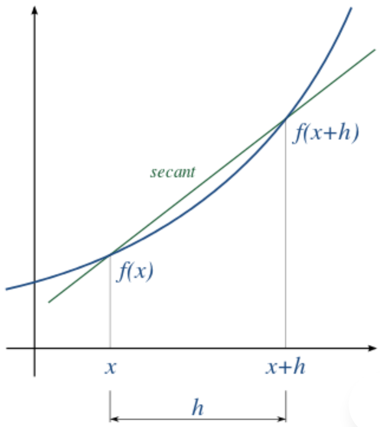
维基百科
自由的百科全书

数值微分

维基百科，自由的百科全书

数值微分是数值方法中的名词，是用函数的值及其他已知资讯来估计一函数导数的算法。

有限差分法



最简单的方式是使用有限差分近似。

简单的二点估计法是计算经过$(x,f(x))$及邻近点$(x+h,f(x+h))$二点形成割线的斜率[1]选择一个小的数值h，表示x的小变化，可以是正值或是负值。其斜率为

$$\frac{f(x+h)-f(x)}{h}.$$

此表示法是牛顿的差商，也称为一阶均差。

割线斜率和切线斜率有些差异，差异大约和h成正比。若h近似于0，则割线斜率近似于切线斜率。因此，函数**f**真正在x处真正的斜率是割线趋近切线时的差商：

$$f'(x)=\lim_{h\to 0}\frac{f(x+h)-f(x)}{h}.$$

若直接将h用0取代会得到除以零的结果，因此计算导数需要一些较不直觉的的方式。

同様的，切线斜率也可以用(x - h)和x二点的割线斜率近似。

另外一种二点估计法是用经过$(x-h,f(x-h))$和$(x+h,f(x+h))$二点的割线，其斜率为

$$\frac{f(x+h)-f(x-h)}{2h}.$$

上述公式称为对称差分，其一次项误差相消，因此割线斜率和切线斜率的差和h^2成正比。对于很小的h而言这个值比单边近似还要准确。特别的是公式虽计算x点的斜率，但不会用到函数在x点的数值。

估计误差为：

$$R=\frac{-f^{(3)}(c)}{6}h^2,$$

其中c为在$x-h$和$x+h$之间的某一点。此误差没有包括因为有限准确度而产生的舍入误差。

很多工程计算机都是用对称差分来计算导数，像德州仪器（TI）的TI-82、TI-83、TI-84及TI-85，其h=0.001[2][3]。

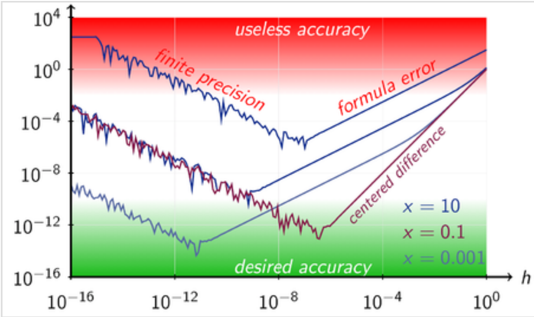
虽然在实务十分常用，但上述二种方式的数值微分常被研究者批评，尤其是被一些鼓励使用自动微分的研究者批评[4]，因为上述的数值微分其精确度不高，若计算器精准度是六位数，用对称差分计算导数只有三位数的精确度，而若是找到一计算斜率的函数，仍可以有几乎六位数的精确度。例如假设$f(x) = x^2$，用2x计算斜率有几乎完整的准确度，而用差分近似就会有上述的问题。

利用浮点数的实际考量

若计算时使用浮点数，就需要考虑h要取到多小。若选的太小，相减之后会有大的舍入误差，事实上整个有限差分的公式都是病态的[5]，若h够小，导数不为零的情形下，在相消后会得到数值微分为零的结果[6]，若h太大，计算割线斜率的结果就会更加准确，但用割线斜率估算切线斜率的误差就更大了。

一种可以产生够小的h，但又不会产生舍入误差的方式是$\sqrt{\varepsilon}x$（不过x不能为0），其中最小浮点数ε大约是2.2×10^{-16}数量级。[7]。以下是一个一个可以平衡舍入误差和公式误差，有最佳精确度的h为

$$h = 2\sqrt{\varepsilon\left|\frac{f(x)}{f''(x)}\right|}$$ [8]（不过f''(x) = 0时不成立），而且需要有关函数的资讯。



在浮点数运算下，不同的h造成的舍入误差及公式误差，只有在特定值下误差才是最小值

上述的最小浮点数是针对双精度（64-bit）变量，单精度变量在这类计算几乎不太实用。其计算结果在二进制中不太可能是“整数”。虽然x是可以用浮点数表示的数字，但$x + h$几乎不会也是可用浮点数表示（而且和x不同）的数字，因此$x + h$需调整为机器可读的数字，因此会出现$(x + h) - x$不等于h的情形，因此用二个函数计算值计算微分时，二个位置的差不会是h。几乎所有的十进制分数在二进制下都会是循环小数（都像1/3在十进制中的情形一样），例如$h = 0.1$在二进制下会是循环小数，是 0.000110011001100...。因此在浮点数下一个可能计算的方式是：

```
h:=sqrt(eps)*x;
xph:=x + h;
dx:=xph - x;
slope:=(F(xph) - F(x))/dx;
```

先计算$(x + h) - x$的值，再用这个值作为微分算式的分母，不过若是用电脑计算，编译器优化的机能可能会认为dx和h相同，因此让上述的方式失效。若是用C或其他类似的编程语言，可以让xph宣告成Volatile变量，以避免此一问题。

高阶方法

也有用更高阶估计导数的方法，或是估计高阶导数的方法。

以下就是一阶导数的五点法（一维下的五点模版）[9]

$$f'(x) = \frac{-f(x+2h)+8f(x+h)-8f(x-h)+f(x-2h)}{12h} + \frac{h^4}{30}f^{(5)}(c)$$

其中$c \in [x-2h, x+2h]$.

微分求积

微分求积（Differential quadrature）是用函数在特定位置数值的加权和来近似导数[10][11]，其名称类似数值积分中用的求积（quadrature），也就是像梯形法或是辛普森法中用的加权和，有许多方式可找出加权的系数，在求解偏微分方程时会用到微分求积。

复变的方法

传统用有限差分近似数值微分的方式是病态的，不过若f是全纯函数，在实轴上的值都是实数，可以用复平面中靠近x的位置来求值，此方式为数值稳定的方式，例如[6]一阶导数可以用以下的复数导数公式计算[12]:

$$f'(x) \approx \Im(f(x+ih))/h.$$

上述公式只在计一阶导数时有效，若要拓展到任意阶导数，需要用到多重复数，结果也会是多重复数的导数。[13]

而任意阶的导数可以用柯西积分公式计算：

$$f^{(n)}(a) = \frac{n!}{2\pi i}\oint_\gamma \frac{f(z)}{(z-a)^{n+1}}\,\mathrm{d}z,$$

其中积分会用数值积分计算。

Lyness和Moler在1967年提出用复变量来计算数值微分[14]。Abate和Dubner提出一种用复数拉普拉斯变换的数值反演为基础的算法[15]。

参考资料

1. Richard L. Burden, J. Douglas Faires (2000), *Numerical Analysis*, (7th Ed), Brooks/Cole. ISBN 0-534-38216-9
2. Katherine Klippert Merseth. Windows on Teaching Math: Cases of Middle and Secondary Classrooms. Teachers College Press. 2003: 34. ISBN 978-0-8077-4279-2.
3. Tamara Lefcourt Ruby; James Sellers; Lisa Korf; Jeremy Van Horn; Mike Munn. Kaplan AP Calculus AB & BC 2015. Kaplan Publishing. 2014: 299. ISBN 978-1-61865-686-5.
4. Andreas Griewank; Andrea Walther. Evaluating Derivatives: Principles and Techniques of Algorithmic Differentiation, Second Edition. SIAM. 2008: 2– [2016-07-03]. ISBN 978-0-89871-659-7. （原始内容存档于2016-07-29）.
5. Numerical Differentiation of Analytic Functions, B Fornberg - ACM Transactions on Mathematical Software (TOMS), 1981
6. Using Complex Variables to Estimate Derivatives of Real Functions, W Squire, G Trapp - SIAM REVIEW, 1998
7. Following *Numerical Recipes in C*, Chapter 5.7 (http://www.nrbook.com/a/bookcpdf/c5-7.pdf) （页面存档备份 (https://web.archive.org/web/20190924145956/http://www.nrbook.com/a/bookcpdf/c5-7.pdf)，存于互联网档案馆）
8. p. 263 [1] (http://www.uio.no/studier/emner/matnat/math/MAT-INF1100/h10/kompendiet/kap11.pdf) （页面存档备份 (https://web.archive.org/web/20191029023557/http://www.uio.no/studier/emner/matnat/math/MAT-INF1100/h10/kompendiet/kap11.pdf)，存于互联网档案馆）
9. Abramowitz & Stegun, Table 25.2
10. Differential Quadrature and Its Application in Engineering: Engineering Applications, Chang Shu, Springer, 2000, ISBN 978-1-85233-209-9
11. Advanced Differential Quadrature Methods, Yingyan Zhang, CRC Press, 2009, ISBN 978-1-4200-8248-7
12. Martins, JRRA; Sturdza, P; Alonso, JJ. The Complex-Step Derivative Approximation. ACM Transactions on Mathematical Software. 2003, **29** (3): 245–262. doi:10.1145/838250.838251. CiteSeerX: 10.1.1.141.8002.
13. 存档副本 (PDF). [2012-11-24]. （原始内容 (PDF)存档于2014-01-09）.
14. Lyness, J. N.; Moler, C. B. Numerical differentiation of analytic functions. SIAM J.Numer. Anal. 1967, **4**: 202–210. doi:10.1137/0704019.
15. Abate, J; Dubner, H. A New Method for Generating Power Series Expansions of Functions. SIAM J. Numer. Anal. March 1968, **5** (1): 102–112. doi:10.1137/0705008.

相关条目

- 自动微分
- 差分
- 五点模板
- 数值积分
- 数值常微分方程
- Savitzky–Golay滤波器
- 数值分析软件列表

外部链接

- http://mathworld.wolfram.com/NumericalDifferentiation.html （页面存档备份 (https://web.archive.org/web/20200728093922/http://mathworld.wolfram.com/NumericalDifferentiation.html)，存于互联网档案馆）
- https://web.archive.org/web/20130820223117/http://math.fullerton.edu/mathews/n2003/NumericalDiffMod.html

- Numerical Differentiation Resources: Textbook notes, PPT, Worksheets, Audiovisual YouTube Lectures (http://numericalmethods.eng.usf.edu/topics/continuous_02dif.html) at Numerical Methods for STEM Undergraduate (https://web.archive.org/web/20060906070428/http://numericalmethods.eng.usf.edu/)
- ftp://math.nist.gov/pub/repository/diff/src/DIFF Fortran code for the numerical differentiation of a function using Neville's process to extrapolate from a sequence of simple polynomial approximations.
- NAG Library numerical differentiation routines (http://www.nag.co.uk/numeric/fl/nagdoc_fl24/html/D04/d04conts.html) （页面存档备份 (https://web.archive.org/web/20191202165033/http://www.nag.co.uk/numeric/fl/nagdoc_fl24/html/D04/d04conts.html)，存于互联网档案馆）
- http://graphulator.com （页面存档备份 (https://web.archive.org/web/20170302024948/http://graphulator.com/)，存于互联网档案馆） Online numerical graphing calculator with calculus function. (http://graphulator.com) （页面存档备份 (https://web.archive.org/web/20170302024948/http://graphulator.com/)，存于互联网档案馆）
