

牛顿迭代法学习笔记

Abel

1 从最简单的形式开始

求方程 $kx + b = 0$ 的根。

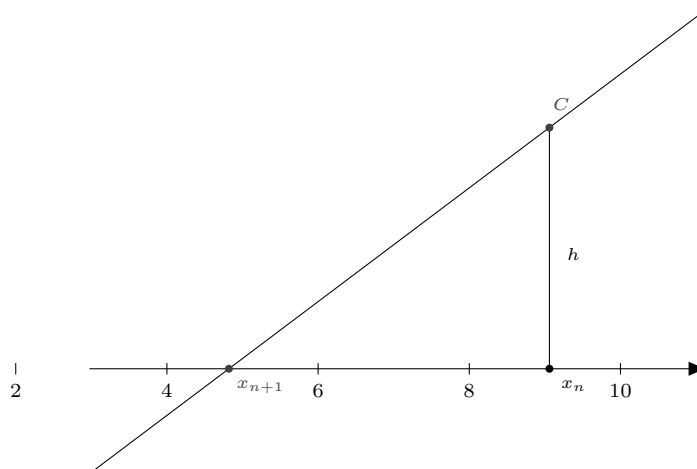
假如已知直线的斜率 k , 且已知 x_n 处的函数值为 h , 如图, 可作直线 $y = kx + b$ 的图像. 同时记

$$f(x) = kx + b$$

函数 $f(x)$ 的导数

$$f'(x) = k$$

就是这条直线的斜率.



如图, 可知

$$\frac{h}{x_n - x_{n+1}} = k$$

$$x_{n+1} = x_n - \frac{h}{k}$$

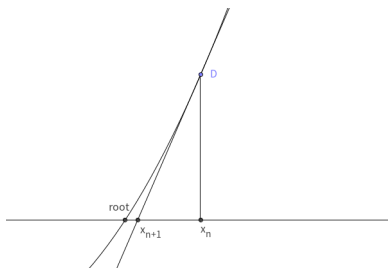
同时, k 就是直线在点 C 处的导数, $k = f'(x_n)$, h 是 x_n 处的函数值那么

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)}$$

对于直线来说, 如果已知方程和一点的函数值, 只需要使用这个公式一次就可以求到方程 $f(x) = 0$ 的根.

2 开方

开方就是求方程 $x^2 = A$ 的根. 可以设 $f(x) = x^2 - A$, 那么依然是求 $f(x) = 0$ 的根. 先任意的猜测一个根 x_n , 然后按照上面的公式来求 x_{n+1}



也就是暂时把抛物线当成直线看待了, 这样可以求到一个 x_{n+1} . 从图像中可以看出虽然它不是方程的根, 但明显比 x_n 更接近方程的根.

于是, 又可以把刚刚得到的 x 值继续迭代下去, 也就是重复

“求函数值, 作切线, 求切线与 x 轴的交点”

这一过程，将得到更加靠近真实根的点. 这一过程使用的公式还是

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)}$$

重复下去，根就会越来越接近方程真实的根.

对于求平方根的过程

$$f(x) = x^2 - A$$

$$f'(x) = 2x$$

$$x_{n+1} = \frac{1}{2} \left(x_n + \frac{A}{x_n} \right)$$

3 优化

这个方法适合于电脑运算. 运算能得到的小数长度取决于电脑的硬件. 因为多迭代几次，可以得到很好的精度. 然而计算圆周率用到的小数长度是很长的，电脑的单精度 float，或者双精度 double float 都无法满足要求. 如果自己写编写高精度的小数运算，长的小数除法是比较难以实现的. 加减法和乘法则相对容易实现. 因此，可以改进成只包含乘法和加法的运算.

3.1 $\frac{1}{\sqrt{A}}$

假如求到倒数的平方根 $B = \frac{1}{\sqrt{A}}$ ，那么再作一次乘法就可以得到 A 的平方根了. $A \times B = \sqrt{A}$. 求倒数的平方根方程和迭代公式如下

$$f(x) = \frac{1}{x^2} - A$$

$$f'(x) = -2x^{-3}$$

$$x_{n+1} = \frac{3}{2}x_n - \frac{1}{2}Ax_n^3$$

可以发现，只需要乘法和减法，不需要除法.

3.2 倒数

倒数也是一种幂函数. 用同样的方法可以求得迭代公式

$$f(x) = \frac{1}{x} - A$$

$$f'(x) = -x^{-2}$$

$$x_{n+1} = 2x_n - Ax_n^2$$

现实中的人们还开发了更高阶的迭代算法，也就是每迭代一次可以获得更多精确的小数. 以后再学习.

拉赞助

如果你觉得本文对你有所启发，可以随意赞助一点

