

本篇讨论非线性方程和非线性方程组的数值解.

非线性方程的数值解

二分查找法

对于方程 $f(x) = 0$, 其奇数重根 x_i 有如下的性质: 存在 x_i 的某个邻域 (a, b) , 使得邻域内分别在零点 x_i 左和右的两点 $l \in (a, x_i), r \in (x_i, b)$ 有 $f(l)f(r) < 0$, 所以我们只需保证 $f(l)f(r) < 0$, 每次将 l 或 r 替换为 $\frac{l+r}{2}$, 直到误差达到要求即可.

二分查找法一定收敛, 但收敛速度较慢. 但二分查找法较为依赖初始值的选取, 且无法处理偶数重根.

迭代法

不动点迭代

如果一函数 $g(x)$ 满足在 x^* 处 $g(x^*) = x^*$, 那么我们就可以尝试使用迭代格式

$$x_{n+1} = g(x_n) \quad (7.1)$$

来找到这个 x^* . 但这个迭代格式在某区间 $[a, b]$ 上不一定能够收敛, 于是我们试图寻找使它收敛的条件:

首先, 在这个区间 $[a, b]$ 内方程 $g(x) = x$ 应有根, 即 $\phi(x) = g(x) - x$ 满足: $\phi(a)\phi(b) < 0$

据此可得出两种互斥情况:

1. $g(a) > a; g(b) < b$
2. $g(a) < a; g(b) > b$

其次, 无论 x , 这一迭代法每迭代一次所产生的误差应小于上一次产生的误差, 即:

$$|x_{n+1} - x^*| < |x_n - x^*|$$

或称:

$$|g(x_n) - g(x^*)| < |x_n - x^*|$$

重写为:

$$\frac{|g(x_n) - g(x^*)|}{|x_n - x^*|} < 1$$

据此可得出 $g(a) - g(b) < a - b$, 与上述情况2矛盾.

因此可以得到数值分析版本的**压缩映像原理**(在泛函分析中这一原理有更准确和泛化的定义):

定理8.1 若 $g(x)$ 在区间 $[a, b]$ 上满足条件:

1. $\forall x \in [a, b], a \leq g(x) \leq b$
2. $\exists 0 < L < 1, \forall x, y \in [a, b], s.t. |g(x) - g(y)| \leq L|x - y|$

则初值在 $[a, b]$ 上的迭代格式 $x_{n+1} = g(x_n)$ 收敛于 $x^* = g(x^*)$

在实际中我们很少对于区间内每两点都找到一个 L 来验证上述第2条条件，故压缩映像原理有另一种提法，即对于第2条条件，我们令 $y \rightarrow x$ ，则：

定理8.2 若 $g(x)$ 在区间 $[a, b]$ 上满足条件：

1. $\forall x \in [a, b], a \leq g(x) \leq b$
2. $\forall x \in [a, b], s.t. |g'(x)| < 1$

则初值在 $[a, b]$ 上的迭代格式 $x_{n+1} = g(x_n)$ 收敛于 $x^* = g(x^*)$

然而我们还是觉得麻烦，验证在区间 $[a, b]$ 上处处导数小于1还算是麻烦，故我们又对它换了一种提法：

定理8.3 若 $g(x)$ 在 x^* 的某一邻域内连续可导(注意：连续可导指的是"连续地可导"，指导函数连续，而不是"函数连续且函数可导")，而 $x^* = g(x^*)$ ，若 $g'(x^*) < 1$ ，则一定存在某个 x^* 的邻域，使得 $x_{n+1} = g(x_n)$ 收敛于 x^* 。

迭代的收敛速度

假使一迭代序列 $\{x_n\}$ 收敛于 x^* ，且有如下关系：

$$\lim_{n \rightarrow \infty} \frac{|x_{n+1} - x^*|}{|x_n - x^*|^r} = C \quad (7.2)$$

则称迭代序列 $\{x_n\}$ 是 r 阶收敛的。这一阶数刻画了在 x^* 附近估计误差缩小的速度，故可以刻画序列收敛于 x^* 的速度。

但式7.2的形式看上去就不方便使用，实际工程中更是难以凭借它来估算收敛速度，而这个 r 次方给我们一种提示：是否可以使用级数展开方法，使得这个 r 与导数联系起来？

我们对满足 $g(x^*) = x^*$ 的 $g(x)$ 在 x^* 处执行Taylor展开并代入 x_n ：

$$g(x_n) = g(x^*) + g'(x^*)(x_n - x^*) + \dots + \frac{g^{(r-1)}(x^*)}{(r-1)!}(x_n - x^*)^{r-1} + \frac{g^{(r)}(\xi)}{r!}(x_n - x^*)^r$$

即：

$$x_{n+1} - x^* = g'(x^*)(x_n - x^*) + \dots + \frac{g^{(r-1)}(x^*)}{(r-1)!}(x_n - x^*)^{r-1} + \frac{g^{(r)}(\xi)}{r!}(x_n - x^*)^r = a(x_n - x^*)^r$$

而基 $\{1, x_n - x^*, (x_n - x^*)^2, \dots\}$ 线性无关，因此可以得出：

$$g'(x^*) = g''(x^*) = \dots = g^{(r-1)}(x^*) = 0 \neq g^{(r)}(x^*) \quad (7.3)$$

式子(7.2)与(7.3)完全等价，因此可以使用式子(7.3)来计算收敛阶数：**第一个在解处非零的导数是几阶，收敛阶数就是几阶。**

函数值迭代

简单迭代法

故基于压缩映像原理之扩展(定理8.3), 我们可以将方程 $f(x) = 0$ 经过同解转化, 化为能够收敛的 $g(x) = x$ 的形式来迭代求解. 这种转化不仅仅是两边同时加一个 x , 可能还需要基于其导函数对其调整, 使其根周围的导数小于1, 故在此之前, 可以先使用二分查找法缩小根的范围.

简单迭代法收敛较慢, 且大部分情况下是一阶收敛(线性收敛), 不能满意.

Steffensen加速

在 n 充分大时, 有关系:

$$\frac{x_n - x^*}{x_{n+1} - x^*} \approx \frac{x_{n+1} - x^*}{x_{n+2} - x^*}$$

解出

$$x^* = x_n - \frac{(x_{n+1} - x_n)^2}{x_{n+2} - 2x_{n+1} + x_n}$$

我们令 $x_{n+3} = x^*$, 即

$$x_{n+3} = x_n - \frac{(x_{n+1} - x_n)^2}{x_{n+2} - 2x_{n+1} + x_n} \quad (7.4)$$

即得Steffensen加速法原理. 它至少是2阶收敛的.

在Steffensen加速的运算过程中, 我们首先使用一般迭代法求出 x_1, x_2, x_3 , 后直接持续使用式子7.4直到进入误差允许范围.

插值迭代

插值迭代的主要思想是用线性方程或简单的多项式方程近似非线性方程, 逐步逼近其解.

Newton迭代法

Newton迭代法使用 $f(x)$ 在 x_n 处的切线与 x 轴的交点作 x_{n+1} . 即:

$$x_{n+1} = g(x_n) = x_n - \frac{f(x_n)}{f'(x_n)} \quad (7.5)$$

Newton法同样满足 $g(x^*) = x^*$

**收敛速度: **易求得 $g'(x^*) = 0$, 故Newton法至少2阶收敛.

但Newton法对重根是线性收敛的, 因此有如下改进手段:

1. $x_{n+1} = x_n - r \frac{f(x_n)}{f'(x_n)}$, r 为根的重数

2. 使用同解方程 $\phi(x) = \frac{f(x)}{f'(x)} = 0$ 的根代替 $f(x)$ 的根

弦截法

弦截法基于Newton法，使用差商 $\Delta f(x)/\Delta x$ 来代替式(7.5)中的导数：

$$x_{n+1} = x_n - \frac{f(x_n)(x_n - x_{n-1})}{f(x_n) - f(x_{n-1})} \tag{7.6}$$

它需要两个初值 x_0, x_1 ，一般取确定有解区间的两端点。

muller法(抛物线法)

抛物线法是三点的弦截法。使用过 x_n, x_{n-1}, x_{n-2} 三点的抛物线 $\phi(x)$ 的，在 $x_{n-2} \rightarrow x_n$ 方向上的零点作 x_{n+1} 。

非线性方程组的数值解

对于非线性方程组：

$$\begin{cases} f_1(x_1, x_2, x_3, \dots, x_n) = 0 \\ f_2(x_1, x_2, x_3, \dots, x_n) = 0 \\ f_3(x_1, x_2, x_3, \dots, x_n) = 0 \\ \dots \\ f_n(x_1, x_2, x_3, \dots, x_n) = 0 \end{cases} \tag{7.7}$$

有两种解法：其一是在某一点线性展开为超平面，以这个切平面来近似这个方程；其二是将其转化为同解的、具有最小值0的非线性函数，再通过梯度下降求出其最值。

Newton-Raphson法，线性化求解

不难得出，对于 $f_i(x_1, x_2, x_3, \dots, x_n) = 0$ ，以其在点 $p = (x_{p1}, x_{p2}, x_{p3}, \dots, x_{pn})$ 处的切平面代替之可有：

$$f(x_1, \dots, x_n) = f_i(p) + \left(\frac{\partial f_i}{\partial x_1}\right)_p (x_1 - x_{p1}) + \left(\frac{\partial f_i}{\partial x_2}\right)_p (x_2 - x_{p2}) + \dots + \left(\frac{\partial f_i}{\partial x_n}\right)_p (x_n - x_{pn})$$

为了方便求解，我们将所有的方程都在同一点 p 处展开，故我们可以使用方程组：

$$\begin{cases} f_1(p) + \left(\frac{\partial f_1}{\partial x_1}\right)_p(x_1 - x_{p1}) + \left(\frac{\partial f_1}{\partial x_2}\right)_p(x_2 - x_{p2}) + \dots + \left(\frac{\partial f_1}{\partial x_n}\right)_p(x_n - x_{pn}) = 0 \\ f_2(p) + \left(\frac{\partial f_2}{\partial x_1}\right)_p(x_1 - x_{p1}) + \left(\frac{\partial f_2}{\partial x_2}\right)_p(x_2 - x_{p2}) + \dots + \left(\frac{\partial f_2}{\partial x_n}\right)_p(x_n - x_{pn}) = 0 \\ f_3(p) + \left(\frac{\partial f_3}{\partial x_1}\right)_p(x_1 - x_{p1}) + \left(\frac{\partial f_3}{\partial x_2}\right)_p(x_2 - x_{p2}) + \dots + \left(\frac{\partial f_3}{\partial x_n}\right)_p(x_n - x_{pn}) = 0 \\ \dots \\ f_n(p) + \left(\frac{\partial f_n}{\partial x_1}\right)_p(x_1 - x_{p1}) + \left(\frac{\partial f_n}{\partial x_2}\right)_p(x_2 - x_{p2}) + \dots + \left(\frac{\partial f_n}{\partial x_n}\right)_p(x_n - x_{pn}) = 0 \end{cases}$$

来代替上述的方程组. 用矩阵的形式表示则为:

$$\begin{pmatrix} \frac{\partial f_1}{\partial x_1} & \frac{\partial f_1}{\partial x_2} & \frac{\partial f_1}{\partial x_3} & \dots & \frac{\partial f_1}{\partial x_n} \\ \frac{\partial f_2}{\partial x_1} & \frac{\partial f_2}{\partial x_2} & \frac{\partial f_2}{\partial x_3} & \dots & \frac{\partial f_2}{\partial x_n} \\ \frac{\partial f_3}{\partial x_1} & \frac{\partial f_3}{\partial x_2} & \frac{\partial f_3}{\partial x_3} & \dots & \frac{\partial f_3}{\partial x_n} \\ \vdots & \vdots & \vdots & & \vdots \\ \frac{\partial f_n}{\partial x_1} & \frac{\partial f_n}{\partial x_2} & \frac{\partial f_n}{\partial x_3} & \dots & \frac{\partial f_n}{\partial x_n} \end{pmatrix} \begin{pmatrix} x_1 - x_{p1} \\ x_2 - x_{p2} \\ x_3 - x_{p3} \\ \vdots \\ x_n - x_{pn} \end{pmatrix} = \begin{pmatrix} f_1(x_0, x_1, \dots, x_n) \\ f_2(x_0, x_1, \dots, x_n) \\ f_3(x_0, x_1, \dots, x_n) \\ \vdots \\ f_n(x_0, x_1, \dots, x_n) \end{pmatrix} \quad (7.8)$$

第一项偏导数矩阵为数学分析学中Jacobi矩阵.

将上述式子变形:

$$\begin{pmatrix} \frac{\partial f_1}{\partial x_1} & \frac{\partial f_1}{\partial x_2} & \frac{\partial f_1}{\partial x_3} & \dots & \frac{\partial f_1}{\partial x_n} \\ \frac{\partial f_2}{\partial x_1} & \frac{\partial f_2}{\partial x_2} & \frac{\partial f_2}{\partial x_3} & \dots & \frac{\partial f_2}{\partial x_n} \\ \frac{\partial f_3}{\partial x_1} & \frac{\partial f_3}{\partial x_2} & \frac{\partial f_3}{\partial x_3} & \dots & \frac{\partial f_3}{\partial x_n} \\ \vdots & \vdots & \vdots & & \vdots \\ \frac{\partial f_n}{\partial x_1} & \frac{\partial f_n}{\partial x_2} & \frac{\partial f_n}{\partial x_3} & \dots & \frac{\partial f_n}{\partial x_n} \end{pmatrix} \left(\begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ \vdots \\ x_n \end{pmatrix} - \begin{pmatrix} x_{p1} \\ x_{p2} \\ x_{p3} \\ \vdots \\ x_{pn} \end{pmatrix} \right) = \begin{pmatrix} f_1(x_0, x_1, \dots, x_n) \\ f_2(x_0, x_1, \dots, x_n) \\ f_3(x_0, x_1, \dots, x_n) \\ \vdots \\ f_n(x_0, x_1, \dots, x_n) \end{pmatrix} \quad (7.8')$$

简记为 $J(x - x_0) = F$, 则可以直接构建迭代格式

$$x_n = x_{n-1} J^{-1} F \quad (7.9)$$

梯度下降法求解

易知, 上述的非线性方程组(7.7)与方程:

$$\sum_{i=1}^n f^2(x_1, x_2, x_3, \dots, x_n) = 0$$

同解(平方和为零当且仅当各项均为零)

故正如我们在炼丹炉中所作的工作那样, 我们可以设置损失函数 $E = \sum_{i=1}^n f^2(x_1, x_2, x_3, \dots, x_n) = 0$, 对其执行梯度下降算法求解.

不会考. 因为人类很难求这种东西的梯度.