数值分析内部课程习题解答

第四章第一部分

SCAU DataHub

2025 年 5 月 3 日

目录

1 The rootfinding problem

Solution 1.1. 代码见 Gitee 仓库.

Solution 1.2. 代码见 Gitee 仓库.

Solution 1.3. 代码见 Gitee 仓库.

Solution 1.4. 代码见 Gitee 仓库.

Solution 1.5. 代码见 Gitee 仓库.

Solution 1.6. 代码见 Gitee 仓库.

Solution 1.7. 根据定义 4.1.7 计算根的重数.

注：定义 4.1.7 来源于多项式，例如 $f(x)=(x-1)^2(x-2)$，显然 $x=1$ 是 2 重根，但是如果写成展开式 $f(x)=x^3-4x^2+5x-2$ 则难以直接看出来，但是 $f(x)$ 展不展开都有个性质：

- $f'(x)=3x^2-8x+5$ 满足 $f(1)=0, f(2)\neq 0$.
- $f''(x)=6x-8$ 满足 $f(1)\neq 0$.

于是我们可以通过 $f(1)=f'(1)=0$ 但 $f''(1)\neq 0$ 来定义 $x=1$ 是 $f(x)$ 的 2 重根. 这个性质进一步推广到一般函数的根的重数的定义.

(a) 根据题设 $f(x)=x^3-2x^2+x-2$, 计算 $f(2)=0$, 但 $f'(x)=3x^2-2x^2+1$ 代入 $x=2$ 得到 $f'(2)=5\neq 0$，因此 $x=2$ 是一个单根. 该根的条件数为 $\kappa=|f'(2)|^{-1}=\frac{1}{5}$.

(b) 4 重根. 过程略.

(c) 单根，条件数为 1. 注意这里需要以极限的形式计算 $f(0$ 和 $f'(x)$. 过程略.

(d) 2 重根. 注意这里 log 没有写底数则默认为 ln. 过程略.

2 Fixed-point iteration

Solution 2.1. 根据不动点的定义验证，并用 4.2.2 式验证收敛. 这里只给出 (a) 的解答.

(a) 将 $x=p=3$ 代入 $g(x)=1+x-\frac{1}{9}x^2$，发现

$$g(p)=1+3-\frac{1}{9}3^2=3=p,$$

故 $x=3$ 是 $g(x)$ 的一个不动点. 接着验证不动点迭代的收敛性，求 $g(x)$ 在 $x=3$ 处的导数值

$$g'(x)|_{x=3}=(1-\frac{2}{9}x)|_{x=3}=\frac{1}{3}<1,$$

因此在该不动点附近的迭代序列收敛.

Solution 2.2. 代码参考 Gitee 仓库.

Solution 2.3. 与第 1 题类似. 解答略.

Solution 2.4. 参考不动点迭代序列分析的部分.

(a) 记 $\epsilon_k = x_k - p$，在 4.2.2 式中给出了误差的近似

$$\epsilon_{k+1} \approx g'(p)\epsilon_k,$$

由此看出，当 $g'(p) > 0$ 时，误差值是恒正的，而当 $g'(p) < 0$ 时，误差值是正负交替的.

计算 $g'(x) = 1 - (2x - 4) = -2x + 5$，当 $x = 2.7$ 附近时 $g'(2.7) < 0$，故误差正负交替，展现出形似螺旋的收敛过程.

(b) $f(x)$ 的根即使得 $f(x) = 0$ 的 x 取值，于是有

$$\begin{aligned} f(x) = x^2 - 4x + 3.5 = 0 \\ \Leftrightarrow \quad x^2 + 3.5 = 4x \\ \Leftrightarrow \quad \underbrace{(x^2 + 3.5)/4}_{\text{记为}\hat{g}(x)} = x, \end{aligned}$$

因此 $\hat{g}(x)$ 的不动点与 $f(x)$ 的根等价.

(c) 代码参考 Gitee 仓库. $x \approx 1.29$ 的不动点序列收敛，$x \approx 2.75$ 的不动点序列发散.

(d) 对 $\hat{g}$ 求导，有 $\hat{g}'(x) = x/2$，根据迭代序列收敛的定理可以解释. 这与 Demo 4.2.3 中的两根收敛情况恰好相反，说明通过变换函数，可能解决某些不动点序列不收敛的问题.

Solution 2.5. 根据不动点迭代序列收敛的条件，分类讨论. 解答略.

Solution 2.6. 根据迭代序列收敛的定理可以解释，当一阶导为 0 时，泰勒展开式就不再依赖一阶项，而是依赖更高阶的项，因此会有更快的收敛速度，代码见 Gitee 仓库，解答略.

Solution 2.7. 根据不动点迭代序列收敛的条件，推断 c 的取值范围，解答略.

3 Newton's method

Solution 3.1. 根据式 4.3.5 或 4.3.6 检验二次收敛性. 代码见 Gitee 仓库.

Solution 3.2. 代码见 Gitee 仓库.

Solution 3.3. 代码见 Gitee 仓库.

Solution 3.4. 代码见 Gitee 仓库.

Solution 3.5. 根据牛顿迭代公式验证.

牛顿迭代公式为

$$x_{k+1} = x_k - \frac{f(x_k)}{f'(x_k)},$$

根据题设 $f(x) = x^{-1} - b, (b \neq 0)$，求导 $f'(x) = -x^{-2}$，于是有

$$x_{k+1} = x_k + \frac{x_k^{-1} - b}{x_k^{-2}} = x_k + x_k^2(x_k^{-1} - b) = 2x_k - bx_k^2,$$

故迭代无需除法运算.

Solution 3.6. 根据分段函数的定义分开讨论导数，进而讨论迭代序列.

题设 $f(x) = \operatorname{sign}(x)\sqrt{|x|}$，我们分开讨论正负两部分，有

$$f(x) = \begin{cases} \sqrt{x}, & x > 0 \\ -\sqrt{-x}, & x < 0 \end{cases}$$

进而分别求导，有

$$f'(x) = \begin{cases} \frac{1}{2\sqrt{x}}, & x > 0 \\ \frac{1}{2\sqrt{-x}}. & x < 0 \end{cases}$$

于是牛顿迭代序列为

$$x_{n+1} = \begin{cases} x_n - \frac{\sqrt{x_n}}{\frac{1}{2\sqrt{x_n}}} = -x_n, & x_n > 0 \\ x_n - \frac{-\sqrt{-x_n}}{\frac{1}{2\sqrt{-x_n}}} = -x_n, & x_n < 0 \end{cases}$$

故当 $x_1 \neq 0$ 时，牛顿迭代就会正负抖动，一直不收敛. 原因是 $f(x)$ 的导数不够光滑，尽管这常常不会影响牛顿迭代，但本题给出了一个有影响的例子.

Solution 3.7. 引入条件 $f(r) = f'(r) = 0$ 且 $f'' \neq 0$，并按照 4.3.4 重新推导.

根据教材中收敛性相关内容，定义

$$\epsilon_k = x_k - r, \quad k = 1, 2, \cdots$$

那么可以将 x_k 表示为 $\epsilon_k + r$，于是有

$$\epsilon_{k+1} + r = x_{k+1} = x_k - \frac{f(x_k)}{f'(x_k)} = \epsilon_k + r - \frac{f(\epsilon_k + r)}{f'(\epsilon_k + r)},$$

等式两端消掉 r，并将 $f(x)$ 在 r 附近作泰勒展开，有

$$\epsilon_{k+1} = \epsilon_k - \frac{f(r) + \epsilon_k f'(r) + \frac{1}{2}\epsilon_k^2 f''(r) + O(\epsilon_k^3)}{f'(r) + \epsilon_k f''(r) + O(\epsilon_k^2)},$$

代入题设条件 $f(r)=f'(r)=0$ 且 $f''\neq 0$，一步步化简

$$\begin{aligned}
\epsilon_{k+1} =&\epsilon_k - \frac{\frac{1}{2}\epsilon_k^2 f''(r)+O(\epsilon_k^3)}{\epsilon_k f''(r)+O(\epsilon_k^2)}\\
=&\epsilon_k - \frac{\frac{1}{2}\epsilon_k(\epsilon_k f''(r))+O(\epsilon_k^2)}{\epsilon_k f''(r)+O(\epsilon_k^2)}\\
\approx&\epsilon_k-\frac{1}{2}\epsilon_k\\
=&\frac{1}{2}\epsilon_k,
\end{aligned}$$

因此，当增加题设条件，即 f 的根的重数为 2 时，牛顿法收敛阶数退化为一阶.

Solution 3.8. 反例为 $f(x)=\ln x$，过程略. 这里也告诉我们，$f(x)$ 的导数趋于 0 不能推出收敛.

4 Interpolation-based methods

Solution 4.1. 根据式 4.4.5 或 4.4.6 检验二次收敛性. 代码见 Gitee 仓库.

Solution 4.2. 代码见 Gitee 仓库.

Solution 4.3. 代码见 Gitee 仓库.

Solution 4.4. 代码见 Gitee 仓库.

Solution 4.5. 根据割线法牛顿迭代公式及线性函数的定义可证，过程略.

割线方法就是取两个点近似切线，以获取牛顿迭代公式中需要的斜率值，对线性函数而言，任意取两点计算的斜率就是其本身的斜率，经过一次迭代后找到零点就是线性函数的零点.

Solution 4.6. 根据割线法牛顿迭代公式，代入题设 $f(x)$ 可推得.

一般来说，当 $x_k=x_{k-1}$ 时，割线法牛顿迭代是失效的，会导致 $x_{k+1}=x_k$ 而无法迭代寻根.

但是对于二次多项式而言，我们可以对公式在形式上做一些变化，利用变化后的公式使其正常运行. 具体地，根据题设 $f(x)=ax^2+b(x)+c$，得到

$$\begin{aligned}
x_{k+1} &= x_k - \frac{f(x_k)(x_k-x_{k-1})}{f(x_k)-f(x_{k-1})}\\
&= x_k - \frac{f(x_k)(x_k-x_{k-1})}{a(x_k+x_{k-1})(x_k-x_{k-1})+b(x_k-x_{k-1})},
\end{aligned}$$

当 $x_k\neq x_{k-1}$ 时，分式上下的 (x_k-x_{k-1}) 可以约掉，于是有

$$x_{k+1}=x_k-\frac{f(x_k)}{a(x_k+x_{k-1})+b},$$

这里我们不太严谨地再套上 $x_k=x_{k-1}$ 的条件，得到迭代公式

$$x_{k+1}=x_k-\frac{f(x_k)}{2ax_k+b}.$$

实际上，上述推导的逻辑是有误的，但是最终的迭代公式却是有效的，因为，如果我们考虑原始的牛顿迭代公式，那么

$$x_{k+1} = x_k - \frac{f(x_k)}{f'(x_k)} = x_k - \frac{f(x_k)}{2ax_k + b},$$

这和前面有割线法牛顿迭代公式推导的一样，成功避开了 $x_k = x_{k-1}$ 导致失效的情况.

Solution 4.7. 根据割线法牛顿迭代公式以及题设可证.

由割线法牛顿迭代公式，代入 $f(x) = x^2$ 可以推导

$$\begin{aligned}
x_{k+1} &= x_k - \frac{f(x_k)(x_k - x_{k-1})}{f(x_k) - f(x_{k-1})} \\
&= x_k - \frac{x_k^2(x_k - x_{k-1})}{x_k^2 - x_{k-1}^2} \\
&= x_k - \frac{x_k^2(x_k - x_{k-1})}{(x_k + x_{k-1})(x_k - x_{k-1})} \\
&= x_k - \frac{x_k^2}{x_k + x_{k-1}} \\
&= \frac{x_k x_{k-1}}{x_k + x_{k-1}},
\end{aligned}$$

等式两边取倒数，得到

$$\frac{1}{x_{k+1}} = \frac{1}{x_{k-1}} + \frac{1}{x_k},$$

因为题设 $1/x_1, 1/x_2$ 是正整数，因此 $1/x_1, 1/x_2, \cdots$ 是一个 Fibonacci 数列.

Solution 4.8. 参考上一节的收敛性推导.

首先，回顾教材的定义，$\epsilon_k = x_k - r$，进而 x_k 可以表示为 $r + \epsilon_k$ (根加上微小误差). 对于割线法牛顿迭代公式

$$x_{k+1} = x_k - \frac{f(x_k)(x_k - x_{k-1})}{f(x_k) - f(x_{k-1})},$$

可以转换为

$$\epsilon_{k+1} = \epsilon_k - \frac{f(r + \epsilon_k)(\epsilon_k - \epsilon_{k-1})}{f(r + \epsilon_k) - f(r + \epsilon_{k-1})}.$$

利用二阶的泰勒展开公式，结合 $f(r) = 0$，有

$$f(r + \epsilon) = \epsilon_k f'(r) + \frac{1}{2}\epsilon_k^2 f''(r) + O(\epsilon_k^3),$$

$$f(r + \epsilon) = \epsilon_{k-1} f'(r) + \frac{1}{2}\epsilon_{k-1}^2 f''(r) + O(\epsilon_{k-1}^3),$$

于是

$$f(r + \epsilon_k) - f(r + \epsilon_{k-1}) = f'(r)(\epsilon_k - \epsilon_{k-1}) + \frac{1}{2}f''(\epsilon_k + \epsilon_{k-1})(\epsilon_k - \epsilon_{k-1}) + O(\epsilon_{k-1}^3)$$

代入牛顿迭代公式

$$
\begin{aligned}
\epsilon_{k+1} &= \epsilon_k - \frac{f(r+\epsilon_k)(\epsilon_k-\epsilon_{k-1})}{f'(r)(\epsilon_k-\epsilon_{k-1})+\frac{1}{2}f''(r)(\epsilon_k+\epsilon_{k-1})(\epsilon_k-\epsilon_{k-1})+O(\epsilon_{k-1}^3)} \\
&= \epsilon_k - \frac{f(r+\epsilon_k)}{f'(r)+\frac{1}{2}f''(r)(\epsilon_k+\epsilon_{k-1})+O(\epsilon_{k-1}^3)} \\
&= \epsilon_k - \frac{\epsilon_k f'(r)+\frac{1}{2}\epsilon_k^2 f''(r)+O(\epsilon_k^3)}{f'(r)+\frac{1}{2}f''(r)(\epsilon_k+\epsilon_{k-1})+O(\epsilon_{k-1}^3)} \\
&= \epsilon_k - \epsilon_k\left[1+\frac{1}{2}\frac{f''(r)}{f'(r)}\epsilon_k+O(\epsilon_k^3)\right]\left[1+\frac{1}{2}\frac{f''(r)}{f'(r)}(\epsilon_k+\epsilon_{k-1})+O(\epsilon_{k-1}^3)\right]^{-1},
\end{aligned}
$$

最后结合几何级数近似 $(1+z)^{-1}\approx 1-z$，上式近似为

$$
\epsilon_{k+1} \approx \epsilon_k - \epsilon_k\left[1+\frac{1}{2}\frac{f''(r)}{f'(r)}\epsilon_k\right]\left[1-\frac{1}{2}\frac{f''(r)}{f'(r)}(\epsilon_k+\epsilon_{k-1})\right],
$$

仅展开到二阶项 $O(\epsilon_{k-1}^2)$，即

$$
\epsilon_{k+1} \approx \epsilon_k - \epsilon_k\left[1-\frac{1}{2}\frac{f''(r)}{f'(r)}\epsilon_{k-1}\right] = \frac{1}{2}\frac{f''(r)}{f'(r)}\epsilon_k\epsilon_{k-1}.
$$

Solution 4.9. 代码见 Gitee 仓库.