数值分析与算法大作业 2

数值方法求 π^x

班级：自 66

姓名：夏卓凡

学号：2016011496

2018 年 12 月 22 日

目录

1 需求分析

1.1 问题的形式化表述

对本次大作业求解问题可以定义如下符号以进行形式化的表述：给定实数 $x \in [1, 10]$，求函数 $y = \pi^x$ 的数值结果，要求结精确到 6 位小数。具体地，应当分为以下步骤计算最终结果：

- 采用数值方法求 π；
- 采用数值方法求 $\ln \pi$；
- 采用数值方法求 π^x，其中 $x \in [1, 10]$。

由于要求 6 位小数的精度，而 π^{10} 已经有在整数部分有 5 位数，应保证最终结果能有 6 位小数是精确的。

本问题为数学问题，不存在模型误差和观测误差。定义 ϵ 为给定误差上界，ϵ_n 为迭代次数/区间段数为 n 时的总误差，δ_n 为对应的舍入误差，Δ_n 为对应的方法误差。

分析中总是取最差情况进行放缩，实际运行情况会远好于估计情况。

1.2 求 π 的数值方法

1.2.1 使用 Newton 法求平方根

作为使用反正切函数展开式求 π 所用到的引理，应给出足够精度的 $\sqrt{\cdot}$ 计算公式。首先考虑 $f(x) = x^2 - a\ (a > 0)$，令

$$\begin{aligned} \varphi(x) &= x - \frac{f(x)}{f'(x)} \\ &= x - \frac{x^2 - a}{2x} \\ &= x - \frac{1}{2}x + \frac{a}{2x} \\ &= \frac{1}{2}(x + \frac{a}{x}) \end{aligned} \tag{1.1}$$

得到求平方根的递推公式 $x_{n+1} = \varphi(x_n) = \frac{1}{2}(x_n + \frac{a}{x_n})$，可以证明在初始值 x_0 选择得当的时候，可以使得 $\lim\limits_{n\to\infty} x_n = \sqrt{a}$。具体而言，对下面步骤要求的 $a = 3$ 而言，取定初始值为 $x_0 = 1.5 \in [1.5, 2]$，应可以收敛。分析 Newton 法在区间 $[1.5, 2]$ 上的收敛性，其中 $f(x) = x^2 - 3$，$f'(x) = 2x$，$f''(x) = 2 \in \mathcal{C}^{(2)}[1.5, 2]$，考察下面的收敛性定理 [1]

- $f(1.5)f(2) = -0.75 \cdot 1 = -0.75 < 0$；
- $f''(x) = 2$ 在 $[1.5, 2]$ 上恒正；
- $f'(x) \neq 0$ 在 $[1.5, 2]$ 上恒成立；
- $\frac{|f(1.5)|}{2-1.5} = 1.5 < 3 = f'(1.5)$，$\frac{|f(2)|}{2-1.5} = 2 < 4 = f'(2)$。

所以 Newton 法收敛于 $x^* = \sqrt{3}$。

接下来分析渐近收敛速度，首先考察误差 $e_n = x_n - x^*$，以及 $e_{n+1} = x_{n+1} - x^*$，那么

$$\begin{aligned} e_{n+1} &= \varphi(x_n) - \varphi(x^*) \\ &= \varphi^{(1)}(x^*)e_n + \frac{1}{2}\varphi^{(2)}(x^*)e_n^2 + \cdots + \frac{1}{m!}\varphi^{(1)}(x^*)e_n^m + \cdots \end{aligned} \tag{1.2}$$

对 Newton 法 $\varphi'(x^*)=0$，采用带 Lagrange 余项的展开式，有

$$e_{n+1}=\frac{1}{2}\varphi''(\xi^*)e_n^2 \tag{1.3}$$

其中 $\xi\in(x_{n+1},x^*)$，为了估计方便，取 $M=\max\limits_{x\in[1.5,2]}|\varphi''(x)|=\max\limits_{x\in[1.5,2]}|\frac{3}{x^3}|=\frac{8}{9}$，得到递推上界公式

$$|e_{n+1}|\le\frac{M}{2}|e_n|^2 \tag{1.4}$$

递推，并考虑 $|e_0|\le\frac{1}{2}$，我们有

$$\begin{aligned}|e_{n+1}|&\le\frac{2}{M}|\frac{M}{2}e_0|^{2^{n+1}}\\&\le\frac{9}{4}|\frac{4}{9}\times\frac{1}{2}|^{2^{n+1}}\\&=\frac{9}{4}\times(\frac{2}{9})^{2^{n+1}}\end{aligned} \tag{1.5}$$

所以对于给定的误差上界 ϵ，该公式收敛速度 [2] 为 $\mathcal{O}(\log\log\frac{1}{\epsilon})$，受控于 $(\frac{2}{9})^{2^{n+1}}$，速度为“指数的指数”，是超线性收敛（平方收敛），速度较快。Newton 法每一轮迭代有 2 次乘除法运算，1 次加减法，所以算法的时间复杂度为 $\mathcal{O}(n)$。Newton 法每一轮迭代没有额外的存储空间开销，所以算法的空间复杂度为 $\mathcal{O}(1)$。

1.2.2 使用 arctan 1 展开求 π

考虑使用 Maclaurin 级数进行最佳逼近，使用反正切函数的展开式求 π，考虑第一个展开公式

$$\begin{aligned}\frac{\pi}{4}&=\arctan 1\\&=\sum_{k=0}^{+\infty}(-1)^k\frac{1}{2k+1}\end{aligned} \tag{1.6}$$

考虑 Lagrange 余项，设算法取前 n 项求和，那么

$$\begin{aligned}|R_n(\xi)|&=\left|\frac{(\arctan x)^{(n+1)}|_{x=\xi}}{(n+1)!}\right|\ \xi\in(0,1)\\&=\left|\frac{(-1)^n n!}{(n+1)!\sqrt{(1+\xi^2)^{n+1}}}\sin((n+1)\arcsin\frac{1}{\sqrt{1+\xi^2}})\right|\\&\le\frac{1}{n+1}\end{aligned} \tag{1.7}$$

对给定的误差上界 ϵ，该公式收敛速度为 $\mathcal{O}(\frac{1}{\epsilon})$，为次线性收敛，受控于 $\frac{1}{n}$，实际上收敛速度较慢。每一轮循环有 2 次乘除法，2 次加减法运算，时间复杂度为 $\mathcal{O}(n)$；由于不涉及额外的存储空间，空间复杂度为 $\mathcal{O}(1)$。

由于此方法收敛过于缓慢，不在正式程序中使用，也不再后面分析其误差。

1.2.3 使用 $\arctan\frac{\sqrt{3}}{3}$ 展开求 π

由于 $x=1$ 的展开式收敛速度过于缓慢，考虑使用更快收敛的方法，第二个展开公式

$$
\begin{aligned}
\frac{\pi}{6} &= \arctan\frac{\sqrt{3}}{3} \\
&= \sum_{k=0}^{+\infty}\frac{(-1)^k\left(\frac{\sqrt{3}}{3}\right)^{2k+1}}{2k+1} \\
&= \frac{\sqrt{3}}{3}\sum_{k=0}^{+\infty}(-1)^k\frac{1}{(2k+1)\cdot 3^k}
\end{aligned}
\tag{1.8}
$$

设取前 n 项进行计算，分析余项

$$
\begin{aligned}
|R_n| &= \left|\frac{\sqrt{3}}{3}\sum_{k=n+1}^{+\infty}(-1)^k\frac{1}{(2k+1)\cdot 3^k}\right| \\
&\le \frac{\sqrt{3}}{3}\sum_{k=n+1}^{+\infty}\left|(-1)^k\frac{1}{(2k+1)\cdot 3^k}\right| \\
&\le \frac{\sqrt{3}}{3}\sum_{k=n+1}^{+\infty}\left(\frac{1}{3}\right)^k \\
&= \frac{\sqrt{3}}{2\cdot 3^{n+1}}
\end{aligned}
\tag{1.9}
$$

对给定的误差上界 ϵ，该公式收敛速度为 $\mathcal{O}(\log\frac{1}{\epsilon})$，为线性收敛，受控于 $\frac{1}{3^n}$，比 (1.6) 的收敛速度有大幅提升。每一轮循环进行 3 次乘除法，2 次加减法，时间复杂度为 $\mathcal{O}(n)$；由于不涉及额外的存储空间，空间复杂度为 $\mathcal{O}(1)$。

1.2.4 使用 BBP 公式计求 π

BBP 公式 [3] 以 Bailey-Borwein-Plouffe 命名，通过十六进制的位抽取算法展开来计算 π 的任意小数位。展开式如下

$$
\pi = \sum_{k=0}^{+\infty}\frac{1}{16^k}\left(\frac{4}{8k+1}-\frac{2}{8k+4}-\frac{1}{8k+5}-\frac{1}{8k+6}\right) \tag{1.10}
$$

$$
= \sum_{k=0}^{+\infty}\frac{1}{16^k}\left(\frac{120k^2+151k+47}{512k^4+1024k^3+712k^2+194k+15}\right) \tag{1.11}
$$

设取前 n 项进行计算，分析余项

$$
\begin{aligned}
|R_n| &= \left|\sum_{k=n+1}^{+\infty}\frac{1}{16^k}\left(\frac{4}{8k+1}-\frac{2}{8k+4}-\frac{1}{8k+5}-\frac{1}{8k+6}\right)\right| \\
&\le \sum_{k=n+1}^{+\infty}\frac{1}{16^k} \\
&= \frac{1}{15\cdot 16^{n+1}}
\end{aligned}
\tag{1.12}
$$

对给定的误差上界 ϵ，该公式收敛速度为 $\mathcal{O}(\log\frac{1}{\epsilon})$，受控于 $\frac{1}{16^n}$，由于 16>3，在常数方面更占优势，其收敛速度实际上比 (1.8) 更快。每一轮循环进行 16 次乘除法，6 次加减法，时间复杂度为 $\mathcal{O}(n)$；由于不涉及额外的存储空间，空间复杂度为 $\mathcal{O}(1)$。

1.3 求 $\ln\pi$ 的数值方法

1.3.1 使用复化 Simpson 公式求 $\ln\pi$

使用数值积分方法计算积分

$$\ln\pi = \int_1^\pi \frac{1}{x}\,\mathrm{d}x \tag{1.13}$$

考虑复化 Simpson 公式，即在积分区间 $[1,\pi]$ 上 n 等分，记 $h=\frac{\pi-1}{n}$，$x_k=1+kh$，那么

$$S(f)=\frac{h}{6}\sum_{k=0}^{n-1}\left[f\left(x_k\right)+4f\left(x_k+\frac{h}{2}\right)+f\left(x_k+h\right)\right] \tag{1.14}$$

对本问题，$f(x)=\frac{1}{x}$，$f^{(4)}(x)=\frac{24}{x^5}$ 分析余项表达式

$$\begin{aligned} |R_S(f,n)| &= \left|-\frac{(\pi-1)h^4}{2880}f^{(4)}(\eta)\right| \quad \eta\in[1,\pi] \\ &= \left|\frac{(\pi-1)^5}{2880n^4}\right|\left|f^{(4)}(\eta)\right| \\ &\le \frac{(\pi-1)^5}{2880n^4}\max_{x\in[1,\pi]}\left|f^{(4)}(x)\right| \\ &\le \frac{(\pi-1)^5}{120n^4} \end{aligned} \tag{1.15}$$

对给定的误差上界 ϵ，该公式收敛速度为 $\mathcal{O}(\frac{1}{\epsilon^{\frac{1}{4}}})$，为次线性收敛，受控于 $\frac{1}{n^4}$。每一轮循环中，进行 5 次乘除法，6 次加减法，时间复杂度为 $\mathcal{O}(n)$；由于不涉及额外的存储空间，空间复杂度为 $\mathcal{O}(1)$。

1.3.2 使用复化 Cotes 公式求 $\ln\pi$

考虑更高阶的复化 Cotes 公式，积分区间仍为 $[1,\pi]$，记 $h=\frac{\pi-1}{n}$，$x_k=1+kh$，那么

$$C(f)=\frac{h}{90}\sum_{k=0}^{n-1}\left[7f\left(x_k\right)+12f\left(x_k+\frac{h}{4}\right)+32f\left(x_k+\frac{h}{2}\right)+12f\left(x_k+\frac{3h}{4}\right)+7f\left(x_k+h\right)\right] \tag{1.16}$$

首先求 Cotes 公式 $C(f)$ 在区间 $[a,b]$ 上的余项，由于 Cotes 公式具有 5 阶代数精度，可构造 $H(x)$ 满足

- $H(a)=f(a)$；
- $H(\frac{3a+b}{4})=f(\frac{3a+b}{4})$；
- $H(\frac{a+b}{2})=f(\frac{a+b}{2})$；
- $H(\frac{a+3b}{4})=f(\frac{a+3b}{4})$；
- $H(b)=f(b)$；

- $H'(\frac{a+b}{2}) = f'(\frac{a+b}{2})$。

这样，$\deg H \le 5$，偏差为

$$f(x) - H(x) = \frac{f^{(6)}(\eta)}{6!}(x-a)\left(x - \frac{3a+b}{4}\right)\left(x - \frac{a+b}{2}\right)^2\left(x - \frac{a+3b}{4}\right)(x-b) \tag{1.17}$$

对其加上绝对值再求积分为

$$\begin{aligned}
|R_C(f)| &= \left|\int_a^b \frac{f^{(6)}(\eta)}{6!}(x-a)\left(x - \frac{3a+b}{4}\right)\left(x - \frac{a+b}{2}\right)^2\left(x - \frac{a+3b}{4}\right)(x-b)\,\mathrm{d}x\right| \\
&\le \left|\frac{f^{(6)}(\eta)}{6!}\right|\int_a^b \left|(x-a)\left(x - \frac{3a+b}{4}\right)\left(x - \frac{a+b}{2}\right)^2\left(x - \frac{a+3b}{4}\right)(x-b)\right|\mathrm{d}x \\
&= \left|\frac{f^{(6)}(\eta)}{6!}\right|\left(\frac{37(b-a)^7}{172032} \times 2 + \frac{5(b-a)^7}{86016}\right) = \frac{|f^{(6)}(\eta)|}{720 \times 2048}(b-a)^7
\end{aligned} \tag{1.18}$$

接下来分析复化 Cotes 公式的余项

$$\begin{aligned}
|R_C(f,n)| &= \left|\sum_{k=0}^{n-1} R_C(f_k)\right| \\
&\le \frac{h^7}{6! \cdot 2048}|f^{(6)}(\eta_k)|\ \eta_k \in [x_k, x_{k+1}] \\
&= \frac{(\pi-1)^7}{6! \cdot 2048 n^6}|f^{(6)}(\eta)|\ \eta \in [1,\pi] \\
&\le \frac{(\pi-1)^7}{6! \cdot 2048 n^6}\max_{x\in[1,\pi]}|f^{(6)}(x)|
\end{aligned} \tag{1.19}$$

对 $f(x) = \frac{1}{x}$，$f^{(6)}(x) = \frac{6!}{x^7}$，所以

$$|R_C(f,n)| \le \frac{(\pi-1)^7}{2048 n^6} \tag{1.20}$$

对给定的误差上界 ϵ，该公式收敛速度为 $\mathcal{O}(\frac{1}{\epsilon^{\frac{1}{6}}})$，为次线性收敛，受控于 $\frac{1}{n^6}$。每一轮循环中，进行 13 次乘除法，10 次加减法，时间复杂度为 $\mathcal{O}(n)$；由于不涉及到额外的存储空间，空间复杂度为 $\mathcal{O}(1)$。

1.4 求 π^x 的数值方法

1.4.1 使用 Taylor 展开式求 $e^{\xi \ln \pi}$

在后续步骤之中使用 x 的整数、小数部分分别计算的方法可以适当提高精度，考虑仅使用指数函数 $\exp(x)$ 计算 π^x 中 x 的小数部分 ξ 对应的值，可做加强假设 $|\xi| \le 0.5$，又知道 $\ln \pi < 1.5$，所以 $|\xi \ln \pi| < \frac{3}{4}$，这加速了 Taylor 级数的收敛。

首先考虑 e^x 的 Taylor 级数

$$\pi^{\xi} = \exp(\xi \ln \pi) = \sum_{k=0}^{+\infty} \frac{(\xi \ln \pi)^k}{k!} \tag{1.21}$$

其余项为

$$\begin{aligned}
|R_n| &= \frac{e^{\eta}}{(n+1)!} \\
&< \frac{e^{\frac{3}{4}}}{(n+1)!}
\end{aligned} \tag{1.22}$$

分析前后两项比的极限，有

$$\lim_{n\to\infty}\frac{e_{n+1}}{e_n}=\frac{1}{n+1}\to 0 \tag{1.23}$$

$$\lim_{n\to\infty}\frac{e_{n+1}}{e_n^2}=\frac{n!}{(n+1)e^{\frac{3}{4}}}\to +\infty \tag{1.24}$$

可以判定该公式的收敛速度为超线性收敛和次平方收敛，受控于 $\frac{1}{n!}$。每轮循环中，进行 3 次乘除法，2 次加减法，时间复杂度为 $\mathcal{O}(n)$；由于不涉及额外的存储空间，空间复杂度为 $\mathcal{O}(1)$。

1.4.2 使用 Runge-Kutta4 阶公式求 $e^{\xi\ln\pi}$

求指数函数 $\exp(x)$ 的值还可以构造微分方程来求，令

$$\begin{cases} y'=f(x,y)=y \\ y(0)=1 \end{cases} \tag{1.25}$$

使用 Runge-Kutta 法的 4 阶公式求解此方程，迭代步骤如下

$$K_1=f(x_n,y_n) \tag{1.26}$$

$$K_2=f\left(x_n+\frac{h}{2},y_n+\frac{h}{2}K_1\right) \tag{1.27}$$

$$K_3=f\left(x_n+\frac{h}{2},y_n+\frac{h}{2}K_2\right) \tag{1.28}$$

$$K_4=f(x_n+h,y_n+hK_3) \tag{1.29}$$

$$y_{n+1}=y_n+\frac{h}{6}\left(K_1+2K_2+2K_3+K_4\right) \tag{1.30}$$

由于初值点为 $x=0$，为了只利用 $\exp(0)=1$，需要控制所求终值点距离 $x=0$ 不能太远，否则误差将持续增大。所以采用了仅将 x 的小数部分引入公式计算, 且 $|\xi\ln\pi|<0.75$，这样做可以有效避免这个问题。

接下来分析 Runge-Kutta4 阶公式的局部方法误差，由于隐式分析上面方程过于复杂，显示代入 $f(x,y)=y$，并且 $y^{(m)}(x_n)=y(x_n)=y_n$ 有

$$y_{n+1}=\left(1+h+\frac{1}{2}h^2+\frac{1}{6}h^3+\frac{1}{24}h^4\right)y_n \tag{1.31}$$

同理，假设 $y_n=y(x_n)$，那么 $y(x_{n+1})$ 在 $x=x_n$ 处的 Taylor 级数为

$$\begin{aligned} y(x_{n+1}) &= y(x_n)+y^{(1)}(x_n)h+\frac{1}{2}y^{(2)}(x_n)h^2+\frac{1}{6}y^{(3)}(x_n)h^3 \\ &\quad +\frac{1}{24}y^{(4)}(x_n)h^4+\frac{1}{120}y^{(5)}(\eta_n)h^5 \quad \eta_n\in[x_n,x_{n+1}] \\ &= \left(1+h+\frac{1}{2}h^2+\frac{1}{6}h^3+\frac{1}{24}h^4\right)y(x_n)+\frac{1}{120}y^{(5)}(\eta_n)h^5 \end{aligned} \tag{1.32}$$

于是局部方法误差为

$$|y_{n+1}-y(x_{n+1})|=\frac{1}{120}y^{(5)}(\eta_n)h^5\sim O(h^5) \tag{1.33}$$

可知其累积方法误差为 $O(h^4)$，下面对此给出一个估计，记 Δ_n 为每步的累积方法误差（带绝对值），依据 (1.31) 可得

$$\Delta_{n+1}\le\left(1+h+\frac{1}{2}h^2+\frac{1}{6}h^3+\frac{1}{24}h^4\right)\Delta_n+\frac{M_5}{120}h^5 \tag{1.34}$$

其中 M_5 为 $y^{(5)}(x)$ 在 $[0,\xi\ln\pi]$ 上的最大值，也就是 $e^{\xi\ln\pi}$，对上式递推，有

$$\begin{aligned}\left(\Delta_{n+1}+\frac{M_5h^4}{120\left(1+\frac{1}{2}h+\frac{1}{6}h^2+\frac{1}{24}h^3\right)}\right) &\le \left(1+h+\frac{1}{2}h^2+\frac{1}{6}h^3+\frac{1}{24}h^4\right)\\ &\quad\cdot\left(\Delta_n+\frac{M_5h^4}{120\left(1+\frac{1}{2}h+\frac{1}{6}h^2+\frac{1}{24}h^3\right)}\right)\\ &\le\cdots\\ &\le\left(1+h+\frac{1}{2}h^2+\frac{1}{6}h^3+\frac{1}{24}h^4\right)^{n+1}\\ &\quad\cdot\frac{M_5h^4}{120\left(1+\frac{1}{2}h+\frac{1}{6}h^2+\frac{1}{24}h^3\right)}\end{aligned} \tag{1.35}$$

其中认为 $\Delta_0=0$，所以

$$\Delta_n\le\left[\left(1+h+\frac{1}{2}h^2+\frac{1}{6}h^3+\frac{1}{24}h^4\right)^n-1\right]\frac{M_5h^4}{120\left(1+\frac{1}{2}h+\frac{1}{6}h^2+\frac{1}{24}h^3\right)} \tag{1.36}$$

当 n 很大时，考虑 $nh=\xi\ln\pi$，第一项趋于常数 $(e^{\xi\ln\pi}-1)$，第二项分母中 h 及其高阶项相比于 1 可以忽略，于是

$$\begin{aligned}\Delta_n &\le \frac{(e^{\xi\ln\pi}-1)M_5}{120}h^4\\ &=\frac{(e^{\xi\ln\pi}-1)e^{\xi\ln\pi}(\xi\ln\pi)^4}{120n^4}\\ &\le\frac{1.2245\times10^{-3}}{n^4}\end{aligned} \tag{1.37}$$

所以给定误差上界 ϵ，该公式收敛速度为 $\mathcal{O}(\frac{1}{\epsilon^{\frac{1}{4}}})$ 受控于 $\frac{1}{n^4}$，为次线性收敛。每轮循环中进行 11 次乘除法，11 次加减法，时间复杂度为 $\mathcal{O}(n)$；由于不涉及额外的存储空间，空间复杂度为 $\mathcal{O}(1)$。

2 方案设计

2.1 程序环境

使用 C++ 语言，使用 Visual Studio 2017 进行开发。未使用任何第三方库。在 Windows10 下进行构建与测试。

2.2 依赖项说明

在第一方的 STL 库中，选用 `std::function` 配合 lambda 表达式进行函数对象的构建与传递，增强代码可读性；选用 `std::chrono` 进行运行计时。除此之外，未使用任何数学类程序库。

2.3 整体计算过程设计

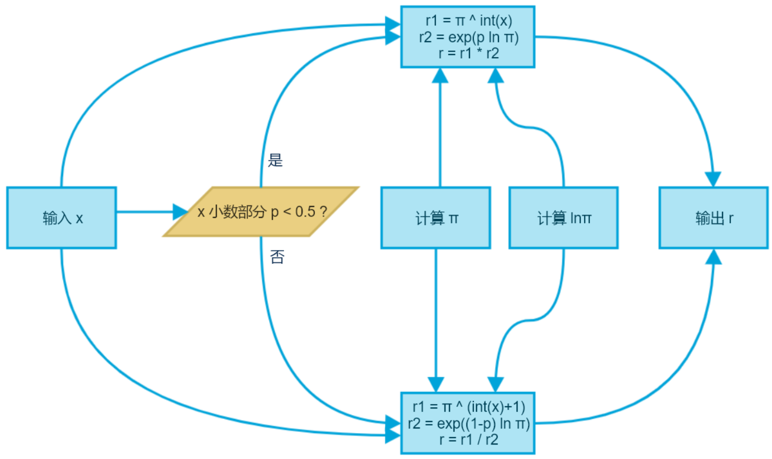

图 1: 整体程序框图

首先使用各个任务中所列的方法计算 π 和 $\ln\pi$，对给定的输入 $x \in [1, 10]$，首先将其分解成整数部分和小数部分

$$x = x_I + \xi \tag{2.1}$$

其中 x_I 为整数部分，ξ 为小数部分，则最终的结果可以写成

$$\begin{aligned}\pi^x &= e^{x\ln\pi} \\ &= e^{(x_I+\xi)\ln\pi} \\ &= \underbrace{\pi\cdots\pi}_{x_I\text{个}}\times e^{\xi\ln\pi} \\ &= \underbrace{\pi\cdots\pi}_{x_I+1\text{个}}\div e^{(1-\xi)\ln\pi}\end{aligned} \tag{2.2}$$

如果 $\xi < 0.5$，使用第一个表示，否则使用第二个表示，这样做能将指数函数的输入项的绝对值控制在 $0.5\ln\pi$ 之内。但应注意，输入的 x 应比较正常，否则由于 ξ 的误差过大也会使得整体误差变得不可控制。

2.4 实验结果分析

以 $\pi^{9.9}$ 的计算结果说明各个方法的效果，精确值来自 Wolfram Alpha 计算引擎。

首先给出各个方法在 Debug 模式下输出 14 位小数的结果，并进行简要分析。从下图可以看出，对于 π 的 14 位小数精确值 3.14159265358979 而言，$\arctan\frac{\sqrt{3}}{3}$ 和 BBP 两种方法均达到了该精度，具有 15 位有效数字。计算速度上，BBP 大概是 $\arctan\frac{\sqrt{3}}{3}$ 的 10 倍左右，可见常数上的优势还是很明显的；使用 BBP 方法的计算结果作为 π 的计算值。

对于 $\ln\pi$ 的 14 位小数精确值 1.14472988584940 而言，复化 Cotes 方法达到了该精度，而复化 Simpson 方法只有 13 位小数精度。计算速度上，复化 Cotes 方法大概是复化 Simpson 的 10 倍左右，这是方法误差中的收敛速度决定的。使用复化 Cotes 方法的计算结果作为 $\ln\pi$ 的计算值。

对于 $\pi^{9.9}$ 的 14 位小数精确值 83518.70267659344770 而言，Taylor 展开方法有着 11 位的更高精度，而 Runge-Kutta4 阶公式的结果只有 10 位精度；而且 Taylor 展开的速度也远高于 Runge-Kutta4 阶公式，这是由于阶乘的收敛速度远高于多项式，而且速度比到达了 562 倍，可见 Taylor 展开更适用于这个问题。

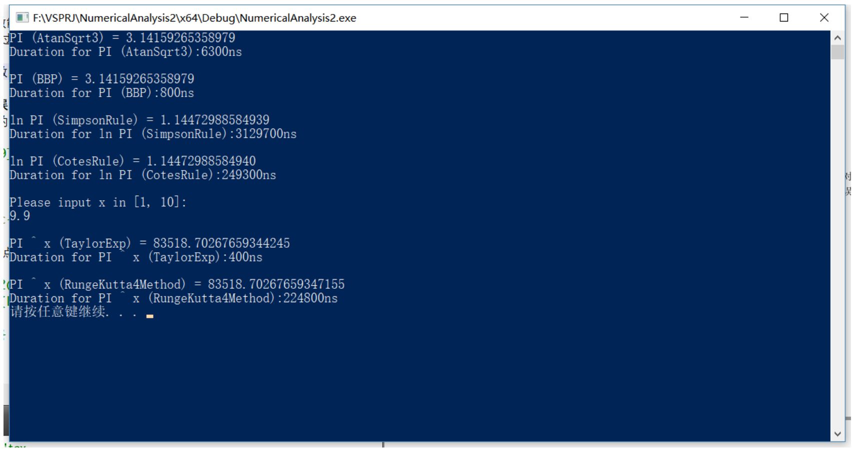

图 2: $\pi^{9.9}$ 在 Debug 模式下输出 14 位小数的结果

接下来给出 Release 模式下的结果，可以看到 6 位小数精度的要求下各个方法的结果都满足要求。同时注意到由于开启了 O2 优化，各种方法的计算速度之间的差异没有之前那么大。

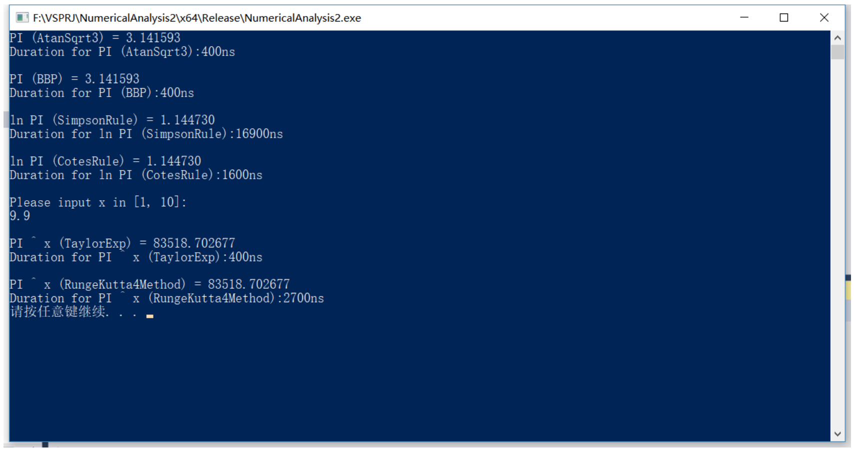

图 3: $\pi^{9.9}$ 在 Release 模式下输出 6 位小数的结果

3 误差分析

3.1 使用 Newton 法求 $\sqrt{3}$ 的误差分析

分析舍入误差，首先计算 (1.1) 误差的传递关系，令

$$L = \frac{M}{2}|e_0| \le \frac{2}{9} \tag{3.1}$$

所以

$$\delta_{n+1} \le L\delta_n + \frac{1}{2} \times 10^{-15} \tag{3.2}$$

其中 $\frac{1}{2} \times 10^{-15}$ 为双精度浮点数单步存储误差上限（十进制表示）。递推上式可以得到

$$\begin{aligned}\left(\delta_{n+1} + \frac{10^{-15}}{2(L-1)}\right) &\le L\left(\delta_n + \frac{10^{-15}}{2(L-1)}\right) \\ &\le \cdots \\ &\le L^{n+1}\frac{10^{-15}}{2(L-1)}\end{aligned} \tag{3.3}$$

整理可得

$$\delta_n \le (1-L^n)\frac{10^{-15}}{2(1-L)} \tag{3.4}$$

综合方法误差 (1.5) 可知，当 n 不是很大时就可以达到预设精度。由于 $\sqrt{3}$ 是后续计算的基础，取 14 位有效数字，总误差 $\epsilon < \frac{1}{2}\times 10^{-14}$。又知道方法误差的收敛速度极高，令方法误差 Δ_n 小于其 1%，即

$$\Delta_n \le \frac{9}{4}\times\left(\frac{2}{9}\right)^{2^n} < \frac{1}{2}\times 10^{-16} \tag{3.5}$$

解得 $n > 4.672$，取 $n = 5$，计算舍入误差

$$\delta_5 \le \left(1-\left(\frac{2}{9}\right)^5\right)\frac{10^{-15}}{2\times(1-\frac{2}{9})} = 6.425\times 10^{-16} \tag{3.6}$$

计算方法误差和总误差并验证

$$\Delta_5 \le 2.814\times 10^{-21} \tag{3.7}$$

$$\epsilon_5 \le \Delta_5 + \delta_5 = 6.425\times 10^{-16} < 0.6\times 10^{-15} \tag{3.8}$$

可以看出满足要求，且所求 $\sqrt{3}$ 的误差为 $\epsilon(\sqrt{3}) \le 0.6\times 10^{-15}$。

3.2 使用 $\arctan\frac{\sqrt{3}}{3}$ 求 π 的误差分析

由于此方法非递推，只需要将循环中每次的存储误差以及 $\sqrt{3}$ 的观测误差纳入考虑即可。首先，计算公式为

$$\pi = 2\sqrt{3}\sum_{k=0}^{+\infty}(-1)^k\frac{1}{(2k+1)\cdot 3^k} \tag{3.9}$$

$$\frac{\partial(\pi)}{\partial(\sqrt{3})} = 2\sum_{k=0}^{+\infty}(-1)^k\frac{1}{(2k+1)\cdot 3^k} = \frac{\pi}{\sqrt{3}} \le 2 \tag{3.10}$$

之后估计舍入误差上界，由于乘 3^k 和取倒数会各引入 1 个双精度存储误差

$$\delta_n \le n\times 10^{-15} + 2\epsilon(\sqrt{3}) \tag{3.11}$$

令总误差 $\epsilon < \frac{1}{2}\times 10^{-13}$，即取 13 位有效数字，令方法误差小于其 1%，即

$$\Delta_n \le \frac{\sqrt{3}}{2\cdot 3^{n+1}} < \frac{1}{2}\times 10^{-15} \tag{3.12}$$

解得 $n > 30.939$，取 $n = 31$，计算各个误差

$$\delta_{31} \le 3.220\times 10^{-14} \tag{3.13}$$

$$\Delta_{31} \le 4.674\times 10^{-16} \tag{3.14}$$

$$\epsilon_{31} \le \Delta_{31} + \delta_{31} \le 3.236674\times 10^{-14} < \frac{1}{2}\times 10^{-13} \tag{3.15}$$

可以看出满足要求，且此方法计算的 π 其误差满足 $\epsilon(\pi) \le 0.35\times 10^{-13}$。

3.3 使用 BBP 公式求 π 的误差分析

BBP 公式与 Taylor 级数类似，是非递推方法。由于 16^k 的倒数仅为移位运算，不引入误差；而上下整数相除仅有 1 项误差，所以估计舍入误差上界为

$$\delta_n \le \frac{n}{2} \times 10^{-15} \tag{3.16}$$

仍取 13 位有效数字，令总误差 $\epsilon < \frac{1}{2} \times 10^{-13}$，并让方法误差为其 1%，有

$$\Delta_n \le \frac{1}{15 \cdot 16^{n+1}} < \frac{1}{2} \times 10^{-15} \tag{3.17}$$

解得 $n > 10.73$，取 $n = 11$，计算各个误差

$$\Delta_{11} \le 2.368 \times 10^{-16} \tag{3.18}$$

$$\delta_{11} \le 5.5 \times 10^{-15} \tag{3.19}$$

$$\epsilon_{11} \le \Delta_{11} + \delta_{11} \le 5.7368 \times 10^{-15} \tag{3.20}$$

可以看出其满足要求，计算 π 的误差满足 $\epsilon(\pi) \le 0.6 \times 10^{-14}$。

3.4 使用复化 Simpson 公式求 $\ln \pi$ 的误差分析

考虑 π 的误差 $\epsilon(\pi) \le \frac{1}{2} \times 10^{-13}$，首先计算 h 的误差

$$\delta h \le \frac{1}{n}\delta\pi + \frac{1}{2} \times 10^{-15} \approx \frac{1}{2} \times 10^{-15} \tag{3.21}$$

再将复化 Simpson 公式写成

$$S(f) = \frac{h}{6}\left[f(1) + 2\sum_{k=1}^{n-1} f(x_k) + 4\sum_{k=0}^{n-1} f(x_{k+0.5}) + f(\pi)\right] \tag{3.22}$$

其中 $\delta x_k = k\delta h + \frac{1}{2} \times 10^{-15}$，给出舍入误差的传递关系

$$\delta_n \le \left|\frac{\partial S(f)}{\partial h}\right| |\delta h| + \sum_{k=1}^{n-1}\left|\frac{\partial S(f)}{\partial f(x_k)}\right| |\delta f(x_k)| + \sum_{k=0}^{n-1}\left|\frac{\partial S(f)}{\partial f(x_{k+0.5})}\right| |\delta f(x_{k+0.5})| + \left|\frac{\partial S(f)}{\partial f(\pi)}\right| |\delta f(\pi)| + \frac{1}{2} \times 10^{-15} \tag{3.23}$$

对其中各项误差分别分析，首先

$$\begin{aligned}\left|\frac{\partial S(f)}{\partial h}\right| |\delta h| &= \frac{\ln \pi}{h}|\delta h| \\ &\le \frac{\ln \pi}{\pi - 1} n \times \frac{1}{2} \times 10^{-15}\end{aligned} \tag{3.24}$$

第二项中 $|\delta f(x_k)| \le |f'(x_k)||\delta x_k| + \frac{1}{2} \times 10^{-15}$，$|f'(x_k)| = \frac{1}{x_k^2} < 1$，$\delta x_k = \frac{k}{n}\delta\pi + \frac{1}{2} \times 10^{-15}$，所以

$$\begin{aligned}\sum_{k=1}^{n-1}\left|\frac{\partial S(f)}{\partial f(x_k)}\right| |\delta f(x_k)| &\le \sum_{k=1}^{n-1} \frac{h}{3}\left(\frac{k}{n}\delta\pi + 1 \times 10^{-15}\right) \\ &= \frac{(n-1)(\pi - 1)}{6n}\left(\delta\pi + 2 \times 10^{-15}\right)\end{aligned} \tag{3.25}$$

第三项同理，有

$$\sum_{k=0}^{n-1}\left|\frac{\partial S(f)}{\partial f(x_{k+0.5})}\right||\delta f(x_{k+0.5})| \leq \frac{\pi-1}{3}\left(\delta\pi + 2\times 10^{-15}\right) \tag{3.26}$$

第四项可以直接分析为

$$\left|\frac{\partial S(f)}{\partial f(\pi)}\right||\delta f(\pi)| \leq \frac{\pi-1}{6n\pi^2}\delta\pi + \frac{1}{2}\times 10^{-15} \tag{3.27}$$

将这些项求和,但注意 n 非常大时,忽略 $\frac{1}{n}$ 的项,并将趋于常数的项视作常数,以及设 $\delta\pi \approx 35\times 10^{-15}$，可以得到舍入误差估计

$$\delta_n \leq \left(\frac{n\ln\pi}{2(\pi-1)} + \frac{37(\pi-1)}{2} + 1\right)\times 10^{-15} \tag{3.28}$$

希望 $\ln\pi$ 有 12 位有效数字，$\epsilon < \frac{1}{2}\times 10^{-12}$，令方法误差为 1%

$$\Delta_n \leq \frac{(\pi-1)^5}{120n^4} < \frac{1}{2}\times 10^{-14} \tag{3.29}$$

解得 $n > 2943.630$，取 $n = 2944$，计算各误差

$$\Delta_{2944} \leq 4.997\times 10^{-15} \tag{3.30}$$

$$\delta_{2944} \leq 8.274\times 10^{-13} \tag{3.31}$$

$$\epsilon_{2944} \leq \Delta_{2944} + \delta_{2944} \leq 8.32397\times 10^{-13} \tag{3.32}$$

很遗憾在这个估计下无法达到 12 位有效数字，只有 11 位有效数字，$\epsilon(\ln\pi) < 0.84\times 10^{-12}$。

3.5 使用复化 Cotes 公式求 $\ln\pi$ 的误差分析

仿照复化 Simpson 公式，写出复化 Cotes 公式的误差传递表达式

$$\begin{aligned}\delta_n \leq & \left|\frac{\partial C(f)}{\partial h}\right||\delta h| + \sum_{k=0}^{n-1}\left|\frac{\partial C(f)}{\partial f(x_k)}\right||\delta f(x_k)| + \sum_{k=0}^{n-1}\left|\frac{\partial C(f)}{\partial f(x_{k+\frac{1}{4}})}\right||\delta f(x_{k+\frac{1}{4}})| + \sum_{k=0}^{n-1}\left|\frac{\partial C(f)}{\partial f(x_{k+\frac{1}{2}})}\right||\delta f(x_{k+\frac{1}{2}})| \\ & + \sum_{k=0}^{n-1}\left|\frac{\partial C(f)}{\partial f(x_{k+\frac{3}{4}})}\right||\delta f(x_{k+\frac{3}{4}})| + \sum_{k=0}^{n-1}\left|\frac{\partial C(f)}{\partial f(x_{k+1})}\right||\delta f(x_{k+1})| + \frac{1}{2}\times 10^{-15}\end{aligned} \tag{3.33}$$

逐项分析，首先

$$\left|\frac{\partial C(f)}{\partial h}\right||\delta h| \leq \frac{n\ln\pi}{\pi-1}\cdot\frac{1}{2}\times 10^{-15} \tag{3.34}$$

再分析一个求和项的误差

$$\sum_{k=0}^{n-1}\left|\frac{\partial C(f)}{\partial f(x_k)}\right||\delta f(x_k)| \leq \sum_{k=0}^{n-1}\frac{7h}{90}\left(\frac{k}{n}\delta\pi + 1\times 10^{-15}\right) = \frac{7(\pi-1)}{90}\left(\frac{n-1}{2n}\delta\pi + 1\times 10^{-15}\right) \tag{3.35}$$

其中 $|f'(x_k)| = \frac{1}{x_k^2} < 1$，$\delta x_k = \frac{k}{n}\delta\pi + \frac{1}{2}\times 10^{-15}$，$|\delta f(x_k)| \le |f'(x_k)||\delta x_k| + \frac{1}{2}\times 10^{-15}$。以此类推，可以得到下面的求和项误差

$$\sum_{k=0}^{n-1}\left|\frac{\partial C(f)}{\partial f(x_{k+\frac{1}{4}})}\right||\delta f(x_{k+\frac{1}{4}})| \le \frac{32(\pi-1)}{90}\left(\frac{n-\frac{1}{2}}{2n}\delta\pi + 1\times 10^{-15}\right) \tag{3.36}$$

$$\sum_{k=0}^{n-1}\left|\frac{\partial C(f)}{\partial f(x_{k+\frac{1}{2}})}\right||\delta f(x_{k+\frac{1}{2}})| \le \frac{12(\pi-1)}{90}\left(\frac{1}{2}\delta\pi + 1\times 10^{-15}\right) \tag{3.37}$$

$$\sum_{k=0}^{n-1}\left|\frac{\partial C(f)}{\partial f(x_{k+\frac{3}{4}})}\right||\delta f(x_{k+\frac{3}{4}})| \le \frac{32(\pi-1)}{90}\left(\frac{n+\frac{1}{2}}{2n}\delta\pi + 1\times 10^{-15}\right) \tag{3.38}$$

$$\sum_{k=0}^{n-1}\left|\frac{\partial C(f)}{\partial f(x_{k+1})}\right||\delta f(x_{k+1})| \le \frac{7(\pi-1)}{90}\left(\frac{n+1}{2n}\delta\pi + 1\times 10^{-15}\right) \tag{3.39}$$

将这些项求和,但注意 n 非常大时,忽略 $\frac{1}{n}$ 的项,并将趋于常数的项视作常数,以及设 $\delta\pi \approx 35\times 10^{-15}$,可以得到舍入误差估计

$$\delta_n \le \left(\frac{n\ln\pi}{2(\pi-1)} + \frac{37(\pi-1)}{2} + 1\right)\times 10^{-15} \tag{3.40}$$

希望 $\ln\pi$ 有 12 位有效数字，$\epsilon < \frac{1}{2}\times 10^{-12}$，令方法误差为 1%

$$\Delta_n \le \frac{(\pi-1)^7}{2048n^6} < \frac{1}{2}\times 10^{-14} \tag{3.41}$$

解得 $n > 164.997$，取 $n = 165$，验算误差

$$\Delta_{165} \le 4.999\times 10^{-15} \tag{3.42}$$

$$\delta_{165} \le 84.717\times 10^{-15} \tag{3.43}$$

$$\epsilon_{165} \le 89.717\times 10^{-15} < 0.90\times 10^{-13} \tag{3.44}$$

可以看出满足要求，此方法计算的 $\ln\pi$ 误差满足 $\epsilon(\ln\pi) < 0.90\times 10^{-13}$。

3.6 使用 Taylor 展开式求 $e^{\xi\ln\pi}$ 的误差分析

分析舍入误差，由于此方法非递推，每一步包含 4 步舍入，并纳入 $\ln\pi$ 的误差

$$\delta_n \le 2n\times 10^{-15} + \xi e^{\xi\ln\pi}\delta\ln\pi \tag{3.45}$$

这里估计 $\delta\ln\pi < 0.84\times 10^{-12}$，且 $\xi < 0.5$，$e^{\xi\ln\pi} < 1.8$，那么

$$\delta_n \le 2n\times 10^{-15} + 0.756\times 10^{-12} \tag{3.46}$$

令其有 11 位有效数字，总误差 $\epsilon < \frac{1}{2}\times 10^{-11}$，令方法误差小于其 1%，那么

$$\Delta_n \le \frac{e^{\frac{3}{4}}}{(n+1)!} < \frac{1}{2}\times 10^{-13} \tag{3.47}$$

取 $n = 16$，验算误差

$$\Delta_{16} \le 5.952\times 10^{-15} \tag{3.48}$$

$$\delta_{16} \le 0.756\times 10^{-12} \tag{3.49}$$

$$\epsilon_{16} \le \Delta_{16} + \delta_{16} \le 0.756\times 10^{-12} < \frac{1}{2}\times 10^{-11} \tag{3.50}$$

可见其满足要求，计算出的 π^x 的小数部分单独能保证有 11 位有效数字。

3.7 使用 Runge-Kutta4 阶公式求 $e^{\xi \ln \pi}$ 的误差分析

首先分析递推的舍入误差

$$\delta_{n+1} \le \left(1+h+\frac{h^2}{2}+\frac{h^3}{6}+\frac{h^4}{24}\right)\delta_n + \left(1+h+\frac{h^2}{2}+\frac{h^3}{6}\right)\delta h + \frac{1}{2}\times 10^{-15} \tag{3.51}$$

递推上式，即可得到

$$\delta_n \le \left[\left(1+h+\frac{h^2}{2}+\frac{h^3}{6}+\frac{h^4}{24}\right)^n - 1\right] \frac{\left(1+h+\frac{h^2}{2}+\frac{h^3}{6}\right)\delta h + \frac{1}{2}\times 10^{-15}}{h+\frac{h^2}{2}+\frac{h^3}{6}+\frac{h^4}{24}} \tag{3.52}$$

考虑到 n 很大，h 很小，$\delta h = \frac{\xi}{n}\delta \ln \pi$，对上式进一步化简

$$\begin{aligned} \delta_n &\le \left(e^{\xi \ln \pi} - 1\right)\frac{n}{\xi \ln \pi}\left(\frac{\xi}{n}\delta \ln \pi + \frac{1}{2}\times 10^{-15}\right) \\ &\le \frac{0.8\delta \ln \pi}{\ln \pi} + \frac{0.4n}{\xi \ln \pi}\times 10^{-15} \end{aligned} \tag{3.53}$$

在适当的输入下，令其有 11 位有效数字，即 $\epsilon < \frac{1}{2}\times 10^{-11}$，方法误差为其 1%，那么

$$\Delta_n \le \frac{1.2245\times 10^{-3}}{n^4} \le \frac{1}{4}\times 10^{-13} \tag{3.54}$$

解得 $n > 332.651$，取 $n = 333$，那么计算各个误差

$$\Delta_{333} \le 9.958\times 10^{-14} \tag{3.55}$$

$$\delta_{333} \le 1.628\times 10^{-12} \tag{3.56}$$

$$\epsilon_{333} \le \Delta_{333} + \delta_{333} \le 1.728\times 10^{-12} < \frac{1}{2}\times 10^{-11} \tag{3.57}$$

可见其能满足小数部分 11 位精度要求，可以使得最终的 π^x 具有 6 位小数的精度。

3.8 最终结果误差分析

由于计算过程是整数乘或除以小数部分，考虑最大 10 次算满的情况，由于

$$\delta\pi^x \le 10\pi^9\delta\pi + \frac{1}{2}\times 10^{-11} < 1.044\times 10^{-8} \tag{3.58}$$

所以这可以保证在 5 位整数的情况下有 7 位小数的精度，达到精度要求。

4 总结与讨论

4.1 double 双精度浮点型与 IEEE754 标准

本次大作业中我仔细学习了 IEEE754 浮点数标准，双精度浮点数在机器中表示为 1 位符号位 s，11 位偏移位 e 和 52 位尾数位 f。并发现实际上前两个问题中值不大于 4，对于十进制的 15 位有效数字，对应于二进制的 $\frac{1}{2}\times 2^{-50}$。事实上这两个值有一定差别，在一定的放缩后似乎会造成比较大的影响，但对于误差上界的估计只是放大了，所以在上面误差分析得到的误差上界比实际的误差上界要更大一些，n 的选取也更加保守。

4.2 Gauss-Legendre 迭代法求 π

事实上，在寻找求 π 的方法的过程中，我发现了 Gauss-Legengre 迭代法 [4]。它的速度更快，与 Newton 法的收敛阶相当，但因为难于误差分析，也不太理解理论背景，没有在正式的程序中使用它，现作简单整理。

$$a_0 = 1 \quad b_0 = \frac{1}{\sqrt{2}} \quad t_0 = \frac{1}{4} \quad p_0 = 1 \tag{4.1}$$

设定以上初值后进行以下迭代

$$a_{n+1} = \frac{a_n + b_n}{2} \tag{4.2}$$

$$b_{n+1} = \sqrt{a_n b_n} \tag{4.3}$$

$$t_{n+1} = t_n - p_n(a_n - a_{n-1})^2 \tag{4.4}$$

$$p_{n+1} = 2p_n \tag{4.5}$$

最终结果为

$$\pi = \frac{(a_{n+1} + b_{n+1})^2}{4t_{n+1}} \tag{4.6}$$

4.3 收敛阶的定义

本次作业中的收敛速度取决于收敛阶的定义，我们采用 [2] 的商收敛阶定义。定义

$$Q_p = \limsup_{k\to\infty} \frac{||x_{k+1} - x^*||_2}{||x_k - x^*||_2^p}, p \in [1, +\infty] \tag{4.7}$$

若 $Q_1 = 0$，则称超线性收敛；若 $0 < Q_1 < 1$，称线性收敛；若 $Q_1 \geq 1$，称次线性收敛。若 $Q_1 = 0$，则称超平方收敛；若 $0 < Q_1 < +\infty$，称平方收敛；若 $Q_1 = +\infty$，称次平方收敛。

4.4 总结与反思

本次大作业可以说是将上课讲到的各种常见的数值积分，方程求根，微分方程求解等计算方法都实际使用了一遍。但应对于具体的问题，一些看上去比较高大上的方法并不好用，而朴素的 Taylor 展开方法确总能在关键时刻解决问题。除此之外，我在本次大作业中对误差分析的过程有着更深的体会，方法误差和舍入误差的意义也理解得更透彻，特别是 n 这一迭代次数/区间个数的选取更是体现了理论与实际之间的某种妥协（trade-off）。以及我在各种误差之中也增强了“选取主项，忽略小项”的工程思维。总的来说，本次大作业处理了一个实际的计算问题，运用了课内和课外的众多方法，锻炼了我的公式推导，误差分析，估计参数的能力，对“数值分析”的本质有了更深刻的认识。

参考文献

[1] 李庆扬. 数值分析. 清华大学出版社有限公司, 2001.

[2] 维基百科. 收敛速度 — 维基百科, 自由的百科全书. https://zh.wikipedia.org/w/index.php?title=%E6%94%B6%E6%96%82%E9%80%9F%E5%BA%A6&oldid=51694613, 2018.

[3] 维基百科. 贝利-波尔温-普劳夫公式 — 维基百科，自由的百科全书. https://zh.wikipedia.org/w/index.php?title=%E8%B4%9D%E5%88%A9-%E6%B3%A2%E5%B0%94%E6%B8%A9-%E6%99%AE%E5%8A%B3%E5%A4%AB%E5%85%AC%E5%BC%8F&oldid=49289437, 2018.

[4] 维基百科. 高斯-勒让德算法 — 维基百科, 自由的百科全书. https://zh.wikipedia.org/w/index.php?title=%E9%AB%98%E6%96%AF-%E5%8B%92%E8%AE%A9%E5%BE%B7%E7%AE%97%E6%B3%95&oldid=33126260, 2014.