数值分析与算法

大作业二——

Sin(x)的数值求解与误差分析

班级：自 71

姓名：屈晨迪

学号：2017010928

1 需求分析

本次作业要对任意给定的x求取$\sin(x)$的数值解，其中$x \in (-10,10)$，精度为 6 位有效数字，求取结果应精确到小数点后 4 位。

算法要求使用逼近、数值积分、常微分方程三种方法中的任意两种，使用的方法本身应能到达任意精度，并需分析相应的方法误差和存储误差对最终结果的影响，比较不同方法之间的计算代价和收敛速度等性能。

2 方案设计

本次作业使用了两种方法求取$\sin(x)$的数值解，分别是函数逼近与常微分方程，对方案原理和算法设计的详细阐述如下。

2.1 逼近

2.1.1 方案基本原理

泰勒公式是一个用函数某点信息描述其邻域函数值的方法。若函数$f(x)$在包含x_0的某个闭区间$[a,b]$上具有 n 阶导数，在开区间(a,b)上具有$n+1$阶导数，则对于该区间上任意一点x，有下式

$$f(x)=\frac{f(x_0)}{0!}+\frac{f'(x_0)}{1!}(x-x_0)+\frac{f''(x_0)}{2!}(x-x_0)^2+\cdots+\frac{f^{(n)}(x_0)}{n!}(x-x_0)^n+R_n(x)$$

当$x_0=0$时，简化为麦克劳林公式，即

$$f(x)=\frac{f(0)}{0!}+\frac{f'(0)}{1!}x+\frac{f''(0)}{2!}x^2+\cdots+\frac{f^{(n)}(0)}{n!}x^n+R_n(x)$$

令前n项为P_n，即

$$P_n(x)=\frac{f(0)}{0!}+\frac{f'(0)}{1!}x+\frac{f''(0)}{2!}x^2+\cdots+\frac{f^{(n)}(0)}{n!}x^n=\sum_{k=0}^{n}\frac{f^{(k)}(0)}{k!}x^k$$

则 n 阶麦克劳林公式可以写成

$$f(x)=P_n(x)+R_n(x)$$

当用$P_n(x)$近似$f(x)$的值时，$R_n(x)$为误差项，采用$Lagrange$余项表示[1]，即存在ξ严格介于 0 和x之间，使得

$$R_n(x)=\frac{f^{(n+1)}(\xi)}{(n+1)!}x^{n+1}$$

将本次求解的函数$\sin(x)$代入上述公式，有

$$f(x)=P_n(x)+R_n(x)$$

$$P_n(x)=x-\frac{x^3}{3!}+\frac{x^5}{5!}-\frac{x^7}{7!}+\cdots+\frac{x^{2n+1}}{(2n+1)!}=\sum_{k=0}^{n}(-1)^k\frac{x^{2k+1}}{(2k+1)!}$$

[1] 证明详见附录 1

$$R_n(x) = (-1)^{n+1}\frac{\cos(\xi)}{(2n+3)!}x^{2n+3}$$

在本次作业中，要求结果精确到小数点后 4 位，即误差绝对值$|R_n| \le \frac{1}{2} \times 10^{-4}$，考虑到$\cos(x)$为有界函数，函数取值范围为$[-1,1]$，有

$$|R_n(x)| = \left|(-1)^{n+1}\frac{\cos(\xi)}{(2n+3)!}x^{2n+3}\right| \le \left|\frac{x^{2n+3}}{(2n+3)!}\right|$$

观察到此时$|R_n(x)|$即P_n下一项$(-1)^{n+1}\frac{x^{2n+3}}{(2n+3)!}$的绝对值，因此我们可以用下一个累加项的绝对值与给定的误差作比较，以判定累加是否结束。考虑到在本次作业中还存在舍入误差，因此在方法误差中留出一定余量，选择总误差的一半即$\frac{1}{4} \times 10^{-4}$作为方法误差上限，如果有

$$|(-1)^{n+1}\frac{x^{2n+3}}{(2n+3)!}| \le \frac{1}{4} \times 10^{-4}$$

必有

$$|R_n(x)| \le \left|\frac{x^{2n+3}}{(2n+3)!}\right| \le \frac{1}{4} \times 10^{-4}$$

由上述分析也可知，该方法可以精确到需要的任意精度，只需调整对累加项绝对值上限的限定即可。

2.1.2 算法设计

由方案原理设计算法，使用$P_n = \sum_{k=0}^{n}(-1)^k\frac{x^{2k+1}}{(2k+1)!}$近似$\sin(x)$。输入数值为$num$，初始累加项$add = num$，使用$do-while$循环做累加，每次执行循环体，计算并判断下一个累加项是否有$add > \frac{1}{4} \times 10^{-4}$，若是，则继续循环，若不是，则累加停止。程序框图如图 1 所示。

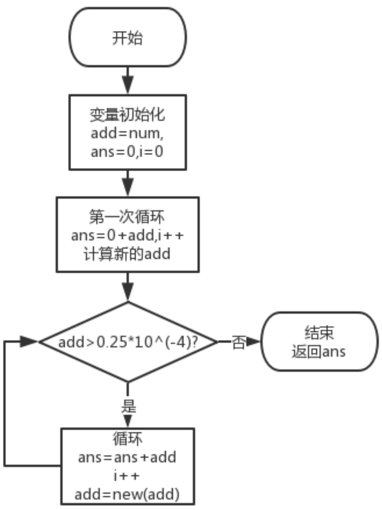

图 1 泰勒展开法流程图

2.2 常微分方程

2.2.1 方案基本原理

考虑到函数$y = \sin(x)$的导数为$y' = \cos(x)$，且有$\sin^2 x + \cos^2 x = 1$，可以列写常微分方程如下

$$\begin{cases} y' = \sqrt{1 - y^2} \\ y(0) = 0 \end{cases}$$

采用变形的欧拉公式进行求解

$$\begin{cases} y_{n+1} = y_n + hK_2 \\ K_1 = f(x_n, y_n) \\ K_2 = f(x_n + \dfrac{h}{2}, y_n + \dfrac{h}{2}K_1) \end{cases}$$

其中$f(x, y) = \sqrt{1 - y^2}$，运用该公式时要将区间$(0, x)$做N等分，h为小区间的长度，有$h = \frac{x}{N}$. 可以证明[2]变形欧拉公式的局部精度为$o(h^3)$，总误差精度为$o(h^2)$，因此可以用h^2与总误差限1×10^{-4}作比较，从而确定 N 和 h，再进行后续迭代。

2.2.2 算法设计

算法主要由两部分构成。首先要确定步长h，将 N 初始值设为 1，不断加 1，判断是否有$\left(\frac{x}{N}\right)^2 > 1 \times 10^{-4}$，若是，则继续增大 N，减小步长，若不是，则终止循环，以当前 N 计算 h 作为后续计算的步长。按照变形欧拉公式进行迭代，取初始值$y_0 = \sin(0) = 0$，$f(x_n, y_n) = \sqrt{1 - y_n^2}$，分别计算$k_1, K_2$和$y_{n+1}$，迭代 N 次，最终获得的$y_{n+1}$即为所求。

注意到常微分方程$y' = \sqrt{1 - y^2} \geq 0$，即$\cos(x) \geq 0$，因此输入的数值$num$有范围限制，程序在获得输入值之后，先判断$num$是否在$[-\frac{\pi}{2}, \frac{\pi}{2}]$的区间里，若不在，根据$\sin(x)$函数的周期性和中心对称性将其转到$[0, \frac{\pi}{2}]$上再进行后续的求解。考虑误差分析中需要用到二阶导数，其二阶导数包含tan(x)部分，当x趋近于π/2时，二阶导趋于无穷，无法进行分析，因此，还需将范围转到$[0, \frac{\pi}{4}]$上，通过函数关系$\sin(x) = \sqrt{1 - \sin\left(\frac{\pi}{2} - x\right)^2}$即可实现转换。完整流程图如图 2 所示。

[2] 证明见 3.2 误差分析

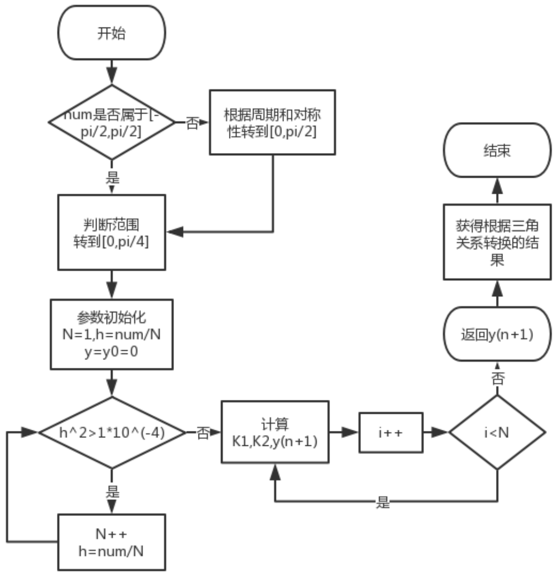

图 2 变形欧拉流程图

3 误差分析

3.1 函数逼近

3.1.1 方法误差

从 2.1.1 的分析可知，使用$taylor$展开逼近$\sin(x)$的方法误差是其展开式的拉格朗日余项，即

$$R_n(x) = (-1)^{n+1}\frac{\cos(\xi)}{(2n+3)!}x^{2n+3}$$

其中ξ严格在 0 和x之间，算法中通过对累加项绝对值的限制规定了方法误差上限，既有

$$|R_n(x)| = |(-1)^{n+1}\frac{\cos(\xi)}{(2n+3)!}x^{2n+3}| \leq |(-1)^{n+1}\frac{x^{2n+3}}{(2n+3)!}| \leq \frac{1}{4}\times 10^{-4}$$

因此方法误差$|R_1| \leq \frac{1}{4}\times 10^{-4}$.

3.1.2 舍入误差

对于本次作业，输入的数值$x \in (-10,10)$，有 6 位有效数字[3]，因此有

$$\begin{cases} |\Delta x| \leq 0.5\times 10^{-6} & x\in(-1,1) \\ |\Delta x| \leq 0.5\times 10^{-5} & other \end{cases}$$

舍入误差随计算不断累积，使用多元函数的误差估计分析舍入误差对结果的影响，在本例中，有

[3] 需要指出，笔者分析泰勒展开误差时，考虑了输入为 6 位有效数字带来的观测误差，但在与助教的交流中发现，此观测误差其实不需考虑，因此后续 3.2 节常微分方程的误差分析中未做考虑

$$P_n^*(x)=x^*-\frac{x^{*3}}{3!}+\frac{x^{*5}}{5!}-\frac{x^{*7}}{7!}+\cdots+\frac{x^{*2n+1}}{(2n+1)!}=f_n(x^*)$$

则

$$|\Delta P_n|=|P_n-P_n^*|=|\left(\frac{\partial f_n}{\partial x}\right)^*\cdot\Delta x|=\left|\left(1-\frac{x^{*2}}{2!}+\frac{x^{*4}}{4!}-\cdots+\frac{x^{*2n}}{2n!}\right)\Delta x\right|$$

可见其总舍入误差与累加次数n相关。

注意到在实际的程序运行中，是用每一步计算出的结果进行计算，再累加迭代，因此还应存在计算机存储误差的累积，上式计算总舍入误差的方法其实是忽略了每次迭代中的存储误差，而本程序中对数值的存储均使用$double$类型，每次运算的截断误差在10^{-16}量级，累积后与10^{-6}相比仍为极小量，因此此种忽略是可以接受的。

当$x\in(-1,1)$时，由$|(-1)^{n+1}\frac{x^{2n+3}}{(2n+3)!}|\le\frac{1}{2}\times10^{-4}$可得$n\le3$，说明$taylor$展开在 3 项以内结果即能收敛，此时

$$|\Delta P_n|=\left|\left(1-\frac{x^{*2}}{2!}+\frac{x^{*4}}{4!}-\cdots+\frac{x^{*2n}}{2n!}\right)\Delta x\right|\le1\cdot|\Delta x|\le0.5\times10^{-6}$$

当$x\in[1,10)\cup(-10,-1]$时，可以推算得$n\in[3,16]$，即当前输入情况下$taylor$展开最多在 16 项左右能够收敛，且易知$\left(1-\frac{x^{*2}}{2!}+\frac{x^{*4}}{4!}-\cdots+\frac{x^{*2n}}{2n!}\right)$收敛到$\cos(x^*)$，此时有

$$\begin{aligned}|\Delta P_n|&=\left|\left(1-\frac{x^{*2}}{2!}+\frac{x^{*4}}{4!}-\cdots+\frac{x^{*2n}}{2n!}\right)\Delta x\right|\le\left(|\cos(x^*)|+\left|\frac{x^{2n+2}}{(2n+2)!}\right|\right)\cdot|\Delta x|\\&\le\left(1+\frac{2n+3}{x^*}\times\frac{1}{2}\times10^{-4}\right)\times0.5\times10^{-5}<(1+10^{-3})\times0.5\times10^{-5}\\&=5.005\times10^{-6}\end{aligned}$$

可以看到，此种方法下的总舍入误差$|R_2|\le5.005\times10^{-6}$，约在$10^{-6}$数量级，与方法误差的$\frac{1}{4}\times10^{-4}$相比为较小量，总误差中方法误差占主导。

综上所示，总误差$|R|=|R_1+R_2|\le|R_1|+|R_2|\le\frac{1}{4}\times10^{-4}+5.005\times10^{-6}<\frac{1}{2}\times10^{-4}$满足题目精确到小数点后四位的要求。

3.1.3 计算代价和收敛速度

由收敛速度定义式（图 5）可知，迭代点列$\mathrm{x_k}=\sum_{i=0}^{k}(-1)^i\frac{x^{2i+1}}{(2i+1)!}$，误差项$|\mathrm{x_k}-x^*|=\frac{\cos(\xi)}{(2k+3)!}x^{2k+3}$，$|\mathrm{x_{k+1}}-x^*|=\frac{\cos(\eta)}{(2k+5)!}x^{2k+5}$，则有

$$\lim_{k\to\infty}\frac{|\mathrm{x_{k+1}}-x^*|}{|\mathrm{x_k}-x^*|}=\lim_{k\to\infty}\frac{\frac{\cos(\eta)}{(2k+5)!}x^{2k+5}}{\frac{\cos(\xi)}{(2k+3)!}x^{2k+3}}=\lim_{k\to\infty}\frac{\cos(\eta)\,x^2}{\cos(\xi)\,(2k+5)(2k+4)}$$

当$k\to\infty$时，有$\mathrm{x_k}=x_{k+1}=x^*$，即$\cos(\eta)=\cos(\xi)=\cos(x^*)$，设 x 有实上限r，此时有

$$\lim_{k\to\infty}\frac{|\mathrm{x_{k+1}}-x^*|}{|\mathrm{x_k}-x^*|}=\lim_{k\to\infty}\frac{x^2}{(2k+5)(2k+4)}\le\lim_{k\to\infty}\frac{r^2}{(2k+5)(2k+4)}=0$$

所以可得泰勒展开是一阶超线性收敛的。

收敛速度(已知收敛)：若

$$\lim_{k\to\infty}\frac{||x_{k+1}-x^*||}{||x_k-x^*||}=a$$

当$0<a<1$时，迭代点列$\{x_k\}$的收敛速度是线性的，这时算法称为**线性收敛。**当$a=0$时，$\{x_k\}$的收敛速度是超线性的，称为**超线性收敛。**

二阶收敛：若

$$\lim_{k\to\infty}\frac{||x_{k+1}-x^*||}{||x_k-x^*||^2}=a$$

图 5 点列收敛速度定

本程序中泰勒展开的计算代价与迭代次数成正比，观察到在 x 取 10 时，迭代次数最大为 16；对于一个输入 x，设迭代次数为 n，每次迭代过程中，有加减运算 3 次，乘除运算 7 次，则计算代价为$O(10n)$即$O(n)$.

3.2 常微分方程

3.2.1 方法误差

分析变形欧拉公式的方法误差。变形的欧拉公式为

$$\begin{cases} y_{n+1}=y_n+hK_2 \\ K_1=f(x_n,y_n) \\ K_2=f(x_n+\dfrac{h}{2},y_n+\dfrac{h}{2}K_1) \end{cases}$$

假设$y_n=y(x_n)$，计算其局部方法误差表示为

$$\Delta_n=y(x_{n+1})-y_{n+1}=y(x_{n+1})-y_n-hK_2$$

注意到$y'(x_n)=f_n$，将其中各个部分在x_n处泰勒展开，有

$$y(x_{n+1})\approx y(x_n)+hy'(x_n)+\frac{h^2}{2}y^{(2)}(x_n)+\frac{h^3}{3!}y^{(3)}(x_n)$$

$$K_2\approx y'\left(x_n+\frac{h}{2}\right)=y'(x_n)+\frac{h}{2}y^{(2)}(x_n)+\frac{h^2}{2\cdot 2!}y^{(3)}(x_n)$$

代入局部误差表达式，近似有

$$\begin{aligned}\Delta_n=&\ y(x_n)+hy'(x_n)+\frac{h^2}{2}y^{(2)}(x_n)+\frac{h^3}{3!}y^{(3)}(x_n)-y_n\\ &-h\left(y'(x_n)+\frac{h}{2}y^{(2)}(x_n)+\frac{h^2}{2\cdot 2!}y^{(3)}(x_n)\right)\\ =&\ \frac{h^3}{3!}y^{(3)}(x_n)-\frac{h^3}{8}y^{(3)}(x_n)=\frac{h^3}{24}y^{(3)}(x_n)=o(h^3)\end{aligned}$$

可以看到，变形的欧拉公式局部截断误差为$o(h^3)$.

下面计算总的方法误差，将变形的欧拉公式改写为预测和校正形式，即

$$\begin{cases} \bar{y}_{n+\frac{1}{2}}=y_n+\dfrac{h}{2}f(x_n,y_n) \\ y_{n+1}=y_n+hf\left(x_{n+\frac{1}{2}},\bar{y}_{n+\frac{1}{2}}\right) \end{cases}$$

则其累计误差

$$\begin{cases} \bar{\Delta}_{n+\frac{1}{2}} \le \Delta_n\left(1+\frac{h}{2}\cdot\frac{\partial f}{\partial y}(x_n,y_n)\right)+\frac{h^2}{8}y^{(2)}(\xi) \\ \Delta_{n+1} \le \Delta_n + h\cdot\frac{\partial f}{\partial y}(x_n,y_n)\cdot\bar{\Delta}_{n+\frac{1}{2}}+\frac{h^3}{24}y^{(3)}(\eta) \end{cases}$$

其中ξ严格介于x_n和$x_{n+\frac{1}{2}}$之间，η介于x_n和x_{n+1}之间。

可以设

$$\left|\frac{\partial f}{\partial y}(x_n,y_n)\right| \le M \quad \left|y^{(2)}(\xi)\right| \le L \quad \left|y^{(3)}(\eta)\right| \le T$$

则将累积误差公式简写为

$$\begin{cases} \bar{\Delta}_{n+\frac{1}{2}} \le \Delta_n\left(1+\frac{h}{2}M\right)+\frac{h^2}{8}L \\ \Delta_{n+1} \le \Delta_n + h\cdot M\cdot\bar{\Delta}_{n+\frac{1}{2}}+\frac{h^3}{24}T \end{cases}$$

将$\bar{\Delta}_{n+\frac{1}{2}}$代入，即有

$$\Delta_{n+1} \le \Delta_n + h\cdot M\left(\Delta_n\left(1+\frac{h}{2}M\right)+\frac{h^2}{8}L\right)+\frac{h^3}{24}T \le \left(1+hM+\frac{h^2}{2}M^2\right)\Delta_n+\left(\frac{ML}{8}+\frac{T}{24}\right)h^3$$

将上述不等式配凑为等比数列，即

$$\Delta_{\mathrm{n+1}}+\frac{3ML+T}{24M+12hM^2}\mathrm{h}^2 \le \left(1+hM+\frac{h^2}{2}M^2\right)\left(\Delta_{\mathrm{n}}+\frac{3ML+T}{24M+12hM^2}\mathrm{h}^2\right)$$

可有

$$\Delta_{\mathrm{n+1}}+\frac{3ML+T}{24M+12hM^2}\mathrm{h}^2 \le \left(1+hM+\frac{h^2}{2}M^2\right)^{n+1}\cdot\frac{3ML+T}{24M+12hM^2}\mathrm{h}^2$$

当$\mathrm{h}\to 0$时，h^2可视为小量忽略，又$\mathrm{n}=\frac{1}{\mathrm{h}}$，$\mathrm{n}\to\infty$，此时有

$$\lim_{h\to 0}\left(1+hM+\frac{h^2}{2}M^2\right)^{n+1} = \lim_{h\to 0}((1+\mathrm{hM})^{\frac{1}{hM}+1})^M = e^M$$

趋于常数，则方法累积误差可以写为

$$\Delta_{\mathrm{n+1}} \le (\mathrm{e}^{\mathrm{M}}-1)\cdot\frac{3ML+T}{24M+12hM^2}\mathrm{h}^2$$

在本例中，$\mathrm{f_n}=y'(x_n)=\sqrt{1-y^2}$，即有

$$\frac{\partial f}{\partial y}(x_n,y_n) = -y(1-y^2)^{-\frac{1}{2}}$$

$$\mathrm{y}^{(2)}(x_n) = f_n' = \frac{\partial f}{\partial y}(x_n,y_n)y_n' = -y$$

$$\mathrm{y}^{(3)}(x_n) = -\sqrt{1-y^2}$$

在$\mathrm{x}\in[0,\frac{\pi}{4}]$上，y递增，有$\mathrm{y}\in[0,\frac{\sqrt{2}}{2}]$，则有

$$\left|\frac{\partial f}{\partial y}(x_n,y_n)\right| \le \frac{\sqrt{2}}{2}\left(1-\left(\frac{\sqrt{2}}{2}\right)^2\right)^{-\frac{1}{2}} = 1$$

$$\left|y^{(2)}(\xi)\right| = y \le \frac{\sqrt{2}}{2}$$

$$\left|y^{(3)}(\eta)\right| = \sqrt{1-y^2} \le 1$$

因此可以取

$$\mathrm{M}=1 \quad \mathrm{L}=\frac{\sqrt{2}}{2} \quad T=1$$

则方法累计误差为

$$\Delta_{\mathrm{n+1}} \le (\mathrm{e}^{\mathrm{M}}-1)\cdot\frac{3ML+T}{24M+12hM^2}\mathrm{h}^2 \le (e-1)\cdot\frac{\frac{3\sqrt{2}}{2}+1}{24+12h}h^2 \le \frac{5.3634h^2}{24+12h}$$

程序中确定 N 和 h 的值时，判定条件为

$$\mathrm{h}^2 \le 1\times10^{-4}$$

即$\mathrm{h} \le 1\times10^{-2}$，由分析可知，$\Delta_{\mathrm{n+1}}$随 h 的增大而增大，因此

$$\Delta_{\mathrm{n+1}} \le \frac{5.3634\times10^{-4}}{24+12\times10^{-2}} \approx 2.2236\times10^{-5}$$

小于给定的误差限$\frac{1}{2}\times10^{-4}$.

注意到在变形欧拉公式的迭代过程及从$[0,\frac{\pi}{2}]$转到$[0,\frac{\pi}{4}]$的过程中，还会有开根号函数mysqrt()产生的方法误差，mysqrt()函数通过二分法实现对输入值x的开根运算，二分法停止的条件是$|\mathrm{res}-\frac{\mathrm{x}}{\mathrm{res}}| \le 1\times10^{-15}$，有

$$|\mathrm{res}^2-\mathrm{x}| \le |\mathrm{res}|\times10^{-15}$$
$$|\mathrm{res}-\mathrm{x}| \le |\frac{\mathrm{res}}{res+x}|\times10^{-15} \le 1\times10^{-15}$$

在10^{-15}量级左右，与每步$\mathrm{o}(\mathrm{h}^3)$约10^{-6}的误差相比为小量，在方法误差分析中忽略不计。

3.2.2 舍入误差

（1）数值范围转换

由于常微分方程中出现根号，有自变量范围的限制，且在分析误差时发现当x接近π/2时会出现误差限无穷大的情况，因此需要先将x转换到$[0,\frac{\pi}{2}]$的范围上，再通过三角等式将其转到$[0,\frac{\pi}{4}]$上。

首先分析转换到$[0,\frac{\pi}{2}]$的过程中出现的初始舍入误差。考虑到计算机中存储的π本身即有误差，为避免多步误差的累积，范围转换采用如下方法，先将输入弧度制转为角度，再判断其与90°的关系，利用正弦函数的对称性和周期性，将其转换到[0,90°]上，再转变为弧度，作为后续输入。按上述方法计算，共有两步弧度制和角度制的转换产生舍入误差，使用的数据类型为 double，因此初始误差在1×10^{-15}左右。

对于$[\frac{\pi}{4},\frac{\pi}{2}]$上的数值计算结果，需要通过$\sin(\mathrm{x})=\sqrt{1-\sin^2(\frac{\pi}{2}-x)}$进行转换得到，此步计算中开根号函数mysqrt()会产生一定的舍入误差，认为大致在10^{-15}左右。

（2）变形欧拉公式迭代中的累积

在变形欧拉公式的计算中，需要不断迭代，每步迭代均会产生舍入误差，在本次程序中，此误差还包含了开根号函数mysqrt()中每一步二分法产生的舍入误差。程序中y_n, K_1, K_2均使用double类型存储，每次计算产生舍入误差为1×10^{-15}左右。其中K_1, K_2的计算会存在开根号的步骤，此处分析扩大误差容限，认为每步舍入误差为5×10^{-15}，即$\frac{1}{2}\times10^{-14}$.

由 3.2.1 方法误差中的分析易得，舍入误差可以表示为

$$\begin{cases}\bar{\delta}_{n+\frac{1}{2}} \le \delta_n\left(1+\dfrac{h}{2}M\right)+\dfrac{1}{2}\times10^{-m}\\ \delta_{n+1}\le\delta_n+h\cdot M\cdot\bar{\delta}_{n+\frac{1}{2}}+\dfrac{1}{2}\times10^{-m}\end{cases}$$

$$\begin{aligned}\delta_{n+1}&\le\delta_n+h\cdot M\left(\delta_n\left(1+\frac{h}{2}M\right)+\frac{1}{2}\times10^{-m}\right)+\frac{1}{2}\times10^{-m}\\&\le\left(1+hM+\frac{h^2}{2}M^2\right)\delta_n+(1+hM)\frac{1}{2}\times10^{-m}\end{aligned}$$

凑等比数列，有

$$\delta_{n+1}+\frac{(1+\mathrm{hM})\times10^{-\mathrm{m}}}{2hM+h^2M^2}\le\left(1+hM+\frac{h^2}{2}M^2\right)\left(\delta_n+\frac{(1+\mathrm{hM})\times10^{-\mathrm{m}}}{2hM+h^2M^2}\right)$$

考虑初始范围转换的舍入误差10^{-15}，即有总舍入误差

$$\delta_{n+1}\le\left(1+hM+\frac{h^2}{2}M^2\right)^{n+1}\left(\frac{(1+\mathrm{hM})\times10^{-\mathrm{m}}}{2hM+h^2M^2}+10^{-15}\right)-\frac{(1+\mathrm{hM})\times10^{-\mathrm{m}}}{2hM+h^2M^2}$$

当$h\to0$时，分析同方法误差部分，即

$$\lim_{h\to0}\left(1+hM+\frac{h^2}{2}M^2\right)^{n+1}=e^M$$

已知此时各参数的取值

$$\mathrm{M}=1\quad \mathrm{L}=\frac{\sqrt{2}}{2}\quad T=1\quad \mathrm{m}=14$$

即有

$$\delta_{n+1}\le(e-1)\frac{(1+\mathrm{h})\times10^{-14}}{2h+h^2}+e\times10^{-15}$$

可以看到当 h 越小时，δ_{n+1}越大，而方法误差关于 h 的表达式为$\Delta_{n+1}\le\frac{5.3634h^2}{24+12h}$，随 h 的减小而缩小，比较$\delta_{n+1}$和$\Delta_{n+1}$两者的速度，注意到$\delta_{n+1}$的系数中包含$10^{-14}$的项，在 h 减小时，$\delta_{n+1}$增加的量远小于$\Delta_{n+1}$减小的量，因此两者的和，即总误差与 h 正相关。

3.2.3 计算代价和收敛速度

已知方法误差表达式$\Delta_{n+1}\le\frac{5.3634h^2}{24+12h}$，即$o(h^2)$，该算法的收敛速度为$\frac{1}{N^2}$；给定要求的精度，如精确到小数点后d位，迭代次数 N 应有$\left(\frac{x}{N}\right)^2\approx10^{-d}$，即$N\approx x\cdot10^{d/2}$，本例中$x\in[0,\frac{\pi}{4}]$，因此计算复杂度为$O(10^{\frac{d}{2}})$.

4 方法比较

	逼近（泰勒展开）	常微分方程（变形欧拉）
方法误差	$\frac{\cos(\xi)}{(2n+3)!}x^{2n+3}$	$\frac{5.3634h^2}{24+12h}$
是否有自变量范围限制	无（但有收敛范围）	有$[0,\frac{\pi}{2}]$，本程序中转至$[0,\frac{\pi}{4}]$
收敛速度	较快	较慢
计算代价	较小	较大

4 项目总体设计

4.1 编译环境及运行

本程序为在 VS2017 平台上使用 C#语言编写的 Windows 窗体应用程序，编译和运行于.NET Framework 4.6.1 框架。

4.2GUI 设计

本次大作业的界面设计如图 3 所示，用户可以在界面上方的文本框中输入想要求解的弧度数值[4]，点击“求解”按钮，下方会显示使用泰勒展开法和变形欧拉公式法求得的结果、误差和迭代次数，如图 4 所示。

图 3 初始界面

[4] 本次大作业规定为$x \in (-10,10)$，有 6 位有效数字

图 4 求解界面

例如输入$x = 4.12416$，两种方法得到的误差均小于$\frac{1}{2} \times 10^{-4}$，满足误差限要求，泰勒展开的迭代次数为 9，变形欧拉为 99，可以看到变形欧拉的收敛速度较慢，需要迭代较多次，这与先前的分析相符。

5 实验总结和心得体会

本次大作业相对第一次大作业的编程难度有所下降，但误差分析的要求大大提升。通过本次作业，我重新复习了函数逼近、常微分方程等数值求解的方法，加深了对泰勒展开、欧拉法、改进欧拉、龙格-库塔等方法误差分析的掌握和理解，再次回顾了几类主要的误差来源，认识到在计算机编程运算中舍入误差会随迭代累积，并用多元函数误差分析进行计算，了解了各种方法收敛速度和复杂度的计算方法；同时，我也复习了 c#编程相关的知识，整体而言收获颇丰。最后感谢老师和助教对我本次大作业的帮助！