数值分析实验报告

WRITE YOUR NAME AND ID HERE

实验一：数值积分与数值微分

第一题：数值积分

0. 数值积分方法

对于积分 $\int_a^b f(x)dx$，可以使用以下三种方法来计算其积分值。

(1). 复合梯形公式

对于给定的区间等分数 n，其在 $[a,b]$ 的节点为 $x_i = a + \frac{b-a}{n}i, i = 0, 1, ..., n$。则此时的复合梯形公式为：

$$\int_a^b f(x)dx \approx T_n = \frac{h}{2}[f(a) + 2\sum_{i=1}^{n-1} f(x_i) + f(b)]$$

其中，$h = \frac{b-a}{n}$。

可以用以下函数实现复合梯形公式：

ADD A CHUNK HERE AND WRITE YOUR CODE

(2). 复合辛普森公式

对于给定的区间等分数 n，其在 $[a,b]$ 的节点为 $x_i = a + \frac{b-a}{n}i, i = 0, 1, ..., n$。则此时的复合辛普森公式为：

$$\int_a^b f(x)dx \approx S_n = \frac{h}{6}[f(a) + 2\sum_{i=1}^{n-1} f(x_i) + 4\sum_{i=0}^{n-1} f(x_{i+\frac{1}{2}}) + f(b)]$$

其中，$h = \frac{b-a}{n}$，$x_{i+\frac{1}{2}} = \frac{1}{2}(x_i + x_{i+1})$。

可以用以下函数实现复合辛普森公式：

ADD A CHUNK HERE AND WRITE YOUR CODE

(3). 龙贝格求积公式

对于将区间 $[a,b]n$ 等分的复合梯形公式，记 $T_n = T(h) = \frac{h}{2}[f(a) + 2\sum_{i=1}^{n-1} f(x_i) + f(b)]$。若对 n 个区间再二等分，即对区间 $2n$ 等分，则有：

$$
\begin{aligned}
T_{2n} = T(\frac{h}{2}) &= \frac{h}{4}[f(a) + 2\sum_{i=1}^{n-1} f(x_i) + 2\sum_{i=0}^{n-1} f(x_{i+\frac{1}{2}}) + f(b)] \\
&= \frac{h}{4}[f(a) + 2\sum_{i=1}^{n-1} f(x_i) + f(b)] + \frac{h}{2}\sum_{i=0}^{n-1} f(x_{i+\frac{1}{2}}) \\
&= \frac{1}{2}T(h) + \frac{h}{2}\sum_{i=0}^{n-1} f(x_{i+\frac{1}{2}})
\end{aligned}
$$

由于 $I - T(h) = -\frac{b-a}{12}h^2 f''(\eta)$，所以 $T(h) = I + O(h^2)$，且 $\lim\limits_{h\to 0} T(h) = I$。因此对 $T(h)$ 按 h 展开有：

$$
\begin{aligned}
T_n = T(h) &= I + \alpha_1 h^2 + \alpha_2 h^4 + ... + \alpha_l h^{2l} + ... \\
T_{2n} = T(\frac{h}{2}) &= I + \alpha_1(\frac{h}{2})^2 + \alpha_2(\frac{h}{2})^4 + ... + \alpha_l(\frac{h}{2})^{2l} + ... \\
\Longrightarrow \frac{4T(\frac{h}{2}) - T(h)}{3} &= I - \frac{\alpha_2}{4}h^4 + ... = I + O(h^4)
\end{aligned}
$$

因此，可以通过加权平均将误差的结束从 $O(h^2)$ 降至 $O(h^4)$。记 $T_0(h) = T(h)$，则可以通过 $T_1(h) = \frac{4T_0(\frac{h}{2}) - T_0(h)}{3} = \frac{4T(\frac{h}{2}) - T(h)}{3}$ 降低误差的精度。同理，可以通过 $T_2(h) = \frac{4T_1(\frac{h}{2}) - T_1(h)}{3}$ 将误差降至 $O(h^6)$。

此方法称为理查森外推，若连续使用此方法，则可得到龙贝格求积公式，其思路如下：

1. 计算第 k 次迭代时的复合梯形求积公式 $T_0^{(k)} = T_{2^k} = T(\frac{h}{2^k})$。
2. 不断对当前所得出的积分值进行理查森加速得到 $T_m^{(k)}, m = 1, 2, ..., k$，其中 $T_m^{(k)} = \frac{4^m T_{m-1}^{(k+1)} - T_{m-1}^{(k)}}{4^m - 1}$。
3. 若 $|T_k^{(k)} - T_{k-1}^{(k-1)}| < \epsilon$，则输出 $T_k^{(k)}$。
4. 回到第 1 步进行迭代。

可以用以下函数实现龙贝格求积公式：

ADD A CHUNK HERE AND WRITE YOUR CODE

1.$\int_{-1.96}^{1.96} \frac{1}{\sqrt{2\pi}} e^{-\frac{x^2}{2}} dx$

(1). 复合梯形公式

复合梯形公式的求积结果如下：

ADD A CHUNK HERE AND WRITE YOUR CODE

(2). 复合辛普森公式

复合辛普森公式的求积结果如下：

ADD A CHUNK HERE AND WRITE YOUR CODE

(3). 龙贝格求积公式

龙贝格公式的求积结果如下：

ADD A CHUNK HERE AND WRITE YOUR CODE

(4). 复合梯形公式和复合辛普森公式的比较

已知积分 $\int_{-1.96}^{1.96} \frac{1}{\sqrt{2\pi}} e^{-\frac{x^2}{2}} dx = 0.95000$，则使用复合梯形公式和复合辛普森公式的误差如下表所示：

n	复合梯形公式的误差	复合辛普森公式的误差

2. $\int_0^1 \sqrt{x} \ln x dx$

(1). 复合梯形公式

复合梯形公式的求积结果如下：

ADD A CHUNK HERE AND WRITE YOUR CODE

(2). 复合辛普森公式

复合辛普森公式的求积结果如下：

ADD A CHUNK HERE AND WRITE YOUR CODE

(3). 龙贝格求积公式

龙贝格公式的求积结果如下：

ADD A CHUNK HERE AND WRITE YOUR CODE

(4). 复合梯形公式和复合辛普森公式的比较

已知积分 $\int_0^1 \sqrt{x} \ln x dx = -0.44444$，则使用复合梯形公式和复合辛普森公式的误差如下表所示：

n	复合梯形公式的误差	复合辛普森公式的误差

n	复合梯形公式的误差	复合辛普森公式的误差

第二题：数值微分

0. 数值微分方法

(1). 两点公式

对于 $f(x)$ 在 a, b 上建立两点插值函数 $P_1(x) = \frac{x-b}{a-b}f(a) + \frac{x-a}{b-a}f(b)$。使用 $P_1(x)$ 在 a, b 处的导数来近似 $f(x)$ 在 a, b 处的导数。

$f'(a) = f'(b) \approx P'(a) = P'(b) = \frac{f(b)-f(a)}{b-a}$

可以使用如下函数实现数值微分的两点公式：

ADD A CHUNK HERE AND WRITE YOUR CODE

(2). 外推法

外推法的思想与龙贝格求积公式类似，使用外推降低误差的阶数。定义 $G(h) = \frac{f(a+h)-f(a-h)}{2h}$。由于 $\lim_{h\to 0} G(h) = f'(a)$ 且 $G(h) = f'(a) + O(h^2)$。所以有：

$$
\begin{aligned}
G(h) &= f'(a) + \alpha_1 h^2 + \alpha_2 h^4 + ... \\
G(\frac{h}{2}) &= f'(a) + \alpha_1 (\frac{h}{2})^2 + \alpha_2 (\frac{h}{2})^4 + ... \\
\Longrightarrow \frac{4G(\frac{h}{2}) - G(h)}{3} &= f'(a) + O(h^4)
\end{aligned}
$$

因此，可不断使用外推法提高数值微分的精度。记 $G_0(h) = G(h), G_m(h) = \frac{4^m G_{m-1}(\frac{h}{2}) - G_{m-1}(h)}{4^m - 1}$，则有 $G_m(h) = f'(a) + O(h^{2m+2})$。类似地，可以通过判断 $|G_m(h) - G_{m-1}(h)| < \epsilon$ 来输出数值微分的结果。不过由于数值微分会受到舍入误差的影响，此处 m 不宜过大。

ADD A CHUNK HERE AND WRITE YOUR CODE

下面使用数值微分方法对函数 $f(x) = e^{-\frac{1}{x}}$ 求区间 $[1, 2]$ 上的导数。

1. 两点法求导数

ADD A CHUNK HERE AND WRITE YOUR CODE

2. 外推法求导数

ADD A CHUNK HERE AND WRITE YOUR CODE

3. 一阶导数图像

ADD A CHUNK HERE AND WRITE YOUR CODE

4. 二阶导数

ADD A CHUNK HERE AND WRITE YOUR CODE

实验二：解非线性方程

题目： 求方程 $x^3 - \cos(x) - 5x - 1 = 0$ 的根。

记 $f(x) = x^3 - \cos x - 5x - 1$，$f(x)$ 的图像如图1所示。从图1中可以看出：方程 $f(x) = 0$ 有 3 个根，第一个位于 $[-3, -1]$ 内，第二个位于 $[-1, 1]$ 内，第三个位于 $[1, 3]$ 内。

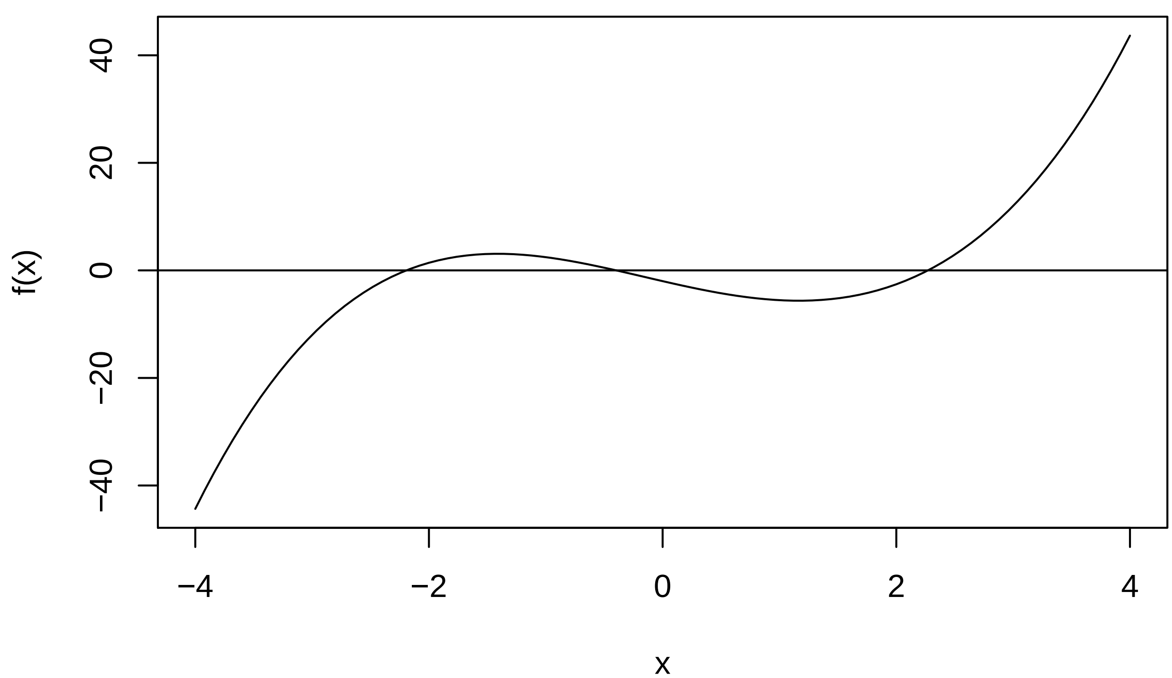

图 1: 函数图像

1. 二分法求根

二分法求根的思路是：对于连续函数 $f(x)$，若其在区间 $[a, b]$ 上满足 $f(a) \cdot f(b) < 0$，则其在 (a, b) 内存在零点 $\bar{x}$。且我们可以通过二分法求出 $\bar{x}$，其思路是：

1. 取 $x_0 = (a + b)/2$。若 $f(x_0) = 0$，则输出 x_0。否则转到下一步。
2. 若 $f(a) \cdot f(x_0) > 0$，则 $a_1 = x_0, b_1 = b$，否则 $a_1 = a, b_1 = x_0$。
3. 重复第二步，构造 $[a_1, b_1], [a_2, b_2], \ldots$，并计算 $x_k = (a_k + b_k)/2$。直至 $f(x_k) = 0$ 或 $|f(x_k)| < \epsilon$ 停止迭代并输出 x_k。

可以使用以下函数实现二分法：

ADD A CHUNK HERE AND WRITE YOUR CODE

分别在 $[-3,-1],[-1,1],[1,3]$ 上使用二分法，所得结果如下所示：

ADD A CHUNK HERE AND WRITE YOUR CODE

2. 不动点迭代法

对于 $f(x)=0$，可以将其变换为 $x=\phi(x)$。对于初值 x_0，构造序列 $x_1=\phi(x_0),x_2=\phi(x_1),...,x_k=\phi(x_{k-1}),...$。如果序列 $\{x_i\}_{i=0}^{\infty}$ 收敛，则其收敛值即为方程 $f(x)=0$ 的根。其思路是：

1. 对于 x_k，计算 $x_{k+1}=\phi(x_k)$。
2. 若 $|x_{k+1}-x_k|<\epsilon$，则输出 x_{k+1}。
3. 若迭代次数超过一定次数，停止算法，输出算法不收敛。
4. 回到第 1 步进行迭代。

可以使用以下函数实现不动点法：

ADD A CHUNK HERE AND WRITE YOUR CODE

下面对于两个不同的迭代函数进行不动点迭代法。其中 $\phi_1(x)=\frac{x^3-\cos(x)-1}{5},\phi_2(x)=\sqrt[3]{\cos(x)+5x+1}$。初值取分别设置为-3,-2,-1,0,1,2,3。

(1) $\phi_1(x)=\frac{x^3-\cos(x)-1}{5}$

对 $\phi_1(x)$ 使用不动点迭代法可得到如下结果：

ADD A CHUNK HERE AND WRITE YOUR CODE

(2) $\phi_2(x)=\phi_2(x)=\sqrt[3]{\cos(x)+5x+1}$

对 $\phi_2(x)$ 使用不动点迭代法可得到如下结果：

ADD A CHUNK HERE AND WRITE YOUR CODE

3. 埃特金加速法

对于不动点迭代法中的迭代函数 $\phi(x)$，不动点迭代法使用迭代公式 $x_{k+1}=\phi(x_k)$。埃特金加速法使用如下的迭代公式：$y_k=\phi(x_k),z_k=\phi(y_k),x_{k+1}=x_k-\frac{(y_k-x_k)^2}{z_k-2y_k+x_k}$。

更一般的，我们将使用埃特金加速法的不动点迭代法称为斯蒂芬森迭代法。其思路如下：

1. 对于 x_k，计算 $x_{k+1}=x_k-\frac{(\phi(x_k)-x_k)^2}{\phi(\phi(x_k))-2\phi(x_k)+x_k}$。

2. 若 $|x_{k+1}-x_k|<\epsilon$，则输出 x_{k+1}。
3. 若迭代次数超过一定次数，停止算法，输出算法不收敛。
4. 回到第 1 步进行迭代。

可以使用如下函数实现斯蒂芬森迭代法：

ADD A CHUNK HERE AND WRITE YOUR CODE

斯蒂芬森迭代法在迭代函数 $\phi_1(x)=\frac{x^3-\cos(x)-1}{6}, \phi_2(x)=\sqrt[3]{\cos(x)+6x+1}$ 和初值-3,-2,-1,0,1,2,3 下的结果为：

(1) $\phi_1(x)=\frac{x^3-\cos(x)-1}{5}$

对 $\phi_1(x)$ 使用斯蒂芬森迭代法可得到如下结果：

ADD A CHUNK HERE AND WRITE YOUR CODE

(2) $\phi_2(x)=\phi_2(x)=\sqrt[3]{\cos(x)+5x+1}$

对 $\phi_2(x)$ 使用斯蒂芬森迭代法可得到如下结果：

ADD A CHUNK HERE AND WRITE YOUR CODE

4. 牛顿法

牛顿法对函数 $f(x)$ 在 x_k 处做泰勒展开 $f(x^*)\approx f(x_k)+f'(x_k)(x^*-x_k)$，其中 x^* 是方程 $f(x)=0$ 的根。因此可以得到递推式 $x_{k+1}=x_k-\frac{f(x_k)}{f'(x_k)}$。牛顿法的思路如下：

1. 对于 x_k，计算 $x_{k+1}=x_k-\frac{f(x_k)}{f'(x_k)}$。
2. 若 $|x_{k+1}-x_k|<\epsilon$，则输出 x_{k+1}。
3. 若迭代次数超过一定次数，停止算法，输出算法不收敛。
4. 回到第 1 步进行迭代。

牛顿法主要有两个缺点：第一、牛顿法需要计算导数 $f'(x)$，计算量较大。第二、牛顿法不能保证算法的收敛性。针对这两个问题，可以分别使用简化的牛顿法和牛顿下山法对牛顿法进行改进。

简化的牛顿法：使用 $f'(x_0)$ 代替 $f'(x_k)$ 减少计算量。其递推式为 $x_{k+1}=x_k-\frac{f(x_k)}{f'(x_0)}$。

牛顿下山法：引入压缩因子 λ，将递推公式变为 $x_{k+1}=x_k-\lambda\frac{f(x_k)}{f'(x_k)}$，使得 $|f(x_{k+1})|<|f(x_k)|$，从而保证其收敛性。其中，λ 的取值开始时为 1，在每次递推中不断折半，直至满足递减的条件。牛顿下山法的思路如下：

1. 对于 x_k，设定 $\lambda=1$。

- 1.1. 计算 $x_{k+1}=x_k-\lambda\frac{f(x_k)}{f'(x_k)}$。
- 1.2. 判断 $|f(x_{k+1})|<|f(x_k)|$。若满足则调到第 2 步。

- 1.3.$\lambda = \lambda/2$，并回到 1.1 步。

2. 若 $|x_{k+1} - x_k| < \epsilon$，则输出 x_{k+1}。
3. 若迭代次数超过一定次数，停止算法，输出算法不收敛。
4. 回到第 1 步进行迭代。

可以使用以下函数实现牛顿法及其改进方法：

ADD A CHUNK HERE AND WRITE YOUR CODE

普通牛顿法的结果如下所示：

ADD A CHUNK HERE AND WRITE YOUR CODE

简化牛顿法的结果如下所示：

ADD A CHUNK HERE AND WRITE YOUR CODE

牛顿下山法的结果如下所示：

ADD A CHUNK HERE AND WRITE YOUR CODE

5. 埃特金加速法的收敛速度

埃特金加速法的收敛速度和不动点法的收敛速度可如下表所示。

对于迭代函数 $\phi_1(x) = \frac{x^3 - \cos(x) - 1}{5}$，其结果为

初值	不动点法的根	不动点法的迭代次数	斯蒂芬森法的根	斯蒂芬森法的迭代次数

对于迭代函数 $\phi_2(x) = \sqrt[3]{\cos(x) + 5x + 1}$，其结果为

初值	不动点法的根	不动点法的迭代次数	斯蒂芬森法的根	斯蒂芬森法的迭代次数

初值	不动点法的根	不动点法的迭代次数	斯蒂芬森法的根	斯蒂芬森法的迭代次数

6. 牛顿法的收敛速度

牛顿法的收敛速度可如下表所示。

初值	普通牛顿法 (根/迭代次数)	简化牛顿法 (根/迭代次数)	牛顿下山法 (根/迭代次数)