Student name: Abao Zhang (张宝)

Student number: 12332459

Homework 7

Question 1

在区间$[x_0, x_5]$上的积分 五阶拉格朗日多项式为：

$$L_i(x) = \prod_{j=0, j\neq i}^{5} \frac{x - x_j}{x_i - x_j}$$

$$P_5(x) = \sum_{i=0}^{5} y_i L_i(x) = \sum_{i=0}^{5} y_i L_i(x)$$

积分结果

$$\int_{x_0}^{x_5} f(x)dx \approx \int_{x_0}^{x_5} P_5(x)dx = \sum_{i=0}^{5} y_i \int_{x_0}^{x_5} L_i(x)dx =$$

令

$$w_k = \int_{x_0}^{x_5} L_i(x)dx$$

于是有

$$\int_{x_0}^{x_5} f(x)dx \approx = \sum_{i=0}^{5} w_i y_i$$

Question 2

(i) 梯形公式

下面给出复合梯形公式计算公式，其中步长$h = \frac{b-a}{M}$

$$\int_a^b f(x)dx \approx \frac{h}{2}\left[f(x_0) + 2\sum_{i=1}^{M-1} f(x_i) + f(x_M)\right]$$

下面给出完整代码，详情见 `question-2.py`

```
def f(x):
    return 2 * np.pi * np.sin(x) * np.sqrt(1 + (np.cos(x)) ** 2)

def trapez(a, b, N, f):
    h = (b - a) / N
    steps = np.linspace(a, b, N + 1, endpoint=True)
    y = np.vectorize(f)(steps)
    return h / 2 * (y[0] + y[-1] + 2 * np.sum(y[1:-1]))

print(trapez(0, np.pi / 4, 10, f))
```

结果为2.4197237019545064

(ii)

下面给出复合辛普森计算公式，其中步长$h = \frac{b-a}{2M}$

$$\int_a^b f(x)dx \approx \frac{h}{3}(f_0 + 4f_1 + 2f_2 + 4f_3 + ... + 2f_{2M-2} + 4f_{2M-1} + f_{2M})$$

下面给出完整代码，详情见 question-2.py，其中w为$[1,4,2,4,...4,2,1]$的数组

```python
def simpson(a, b, M, f):
    h = (b - a) / (2 * M)
    steps = np.linspace(a, b, 2 * M + 1, endpoint=True)
    y = np.vectorize(f)(steps)
    w = np.ones_like(y)
    w[1:-1:2] = 4
    w[2:-2:2] = 2
    return h / 3 * np.sum(w * y)

print(trapez(0, np.pi / 4, 10, f))
```

结果为2.4224337903921787

Question 3

辛普森积分法则误差分析，其中$f(x) = \cos(x)$，积分区间$[-\frac{\pi}{6}, \frac{\pi}{6}]$，$h = \frac{b-a}{2M}$ =根据辛普森误差公式

$$E_{S(f,h)} = \frac{-(b-a)f^{(4)}(c)h^4}{180} = \frac{-\pi f^{(4)}(c)h^4}{540}$$

其中$f^{(4)}(x) = \cos(x)$ 在区间上的取值范围为$\left[\frac{\sqrt{3}}{2}, 1\right]$，根据函数$E_S(f,h)$单调性，取$c = 1$，即$f^{(0)} = 1$，求解不等式$E_S < 5 * 10^{-9}$，下面用代码计算

```python
f = np.cos
a = -np.pi / 6
b = np.pi / 6

f_4 = np.cos
max_f_4 = np.cos(0)
min_f_4 = np.cos(a)

def error(a, b, max_f_4, M):
    h = (b - a) / (2 * M)
    return np.abs((b - a) * max_f_4 * h**4 / 180)

for i in range(1, 20):
    e = error(a, b, max_f_4, i)
    print(f"M={i}, Error={e}")
```

结果如下，可以知道$M = 18$

```
M=14, Error=1.1382520978365635e-08
M=15, Error=8.63745038824483e-09
M=16, Error=6.67222482154685e-09
M=17, Error=5.235460853017735e-09
M=18, Error=4.165437108528561e-09
M=19, Error=3.355337404600136e-09
```

Question 4

首先对五个$g(t)$进行精确积分

$$\int_0^4 g(t)dt = \int_0^4 1dt = 4$$

$$\int_0^4 g(t)dt = \int_0^4 tdt = 8$$

$$\int_0^4 g(t)dt = \int_0^4 t^2dt = \frac{64}{3}$$

$$\int_0^4 g(t)dt = \int_0^4 t^3dt = 64$$

$$\int_0^4 g(t)dt = \int_0^4 t^4dt = \frac{1024}{5}$$

代入积分公式，另积分公式等于精确积分结果，就可以构造出方程组

$$\int_0^4 g(t)dt = w_0 + w_1 + w_2 + w_3 + w_4 = 4$$

$$\int_0^4 g(t)dt = w_1 + 2w_2 + 3w_3 + 4w_4 = 8$$

$$\int_0^4 g(t)dt = w_1 + 4w_2 + 9w_3 + 16w_4 = \frac{64}{3}$$

$$\int_0^4 g(t)dt = w_1 + 8w_2 + 27w_3 + 64w_4 = 64$$

$$\int_0^4 g(t)dt = w_1 + 16w_2 + 81w_3 + 256w_4 = \frac{1024}{5}$$

这里使用作业四 的题目四的代码解方程组，见 `question-4.py`，结果如下

$$\begin{aligned} w_0 &= 0.3111111111111109 \\ w_1 &= 1.4222222222222207 \\ w_2 &= 0.5333333333333368 \\ w_3 &= 1.4222222222222196 \\ w_4 &= 0.3111111111111118 \end{aligned}$$

Question 5

(a)

由中点公式

$$M(f,h) = h\sum_{k=1}^{N} f(a+(k-0.5)h)$$

将$[a,b]$分为2^J个区间，则有间隔$h_J = \frac{b-a}{2^J}$，代入中点公式， 于是有

$$M(J) = M(f,h_J) = h_J \sum_{k=1}^{2^J} f(a+(k-0.5)h_J)$$

当$J=0$ 时，$h_0 = (b-a)$，代入，即可得

$$M(0) = (b-a)f\left(\frac{a-b}{2}\right)$$

(b)

在 Romberg 积分中，使用顺序中点法（Sequential Midpoint Rule）代替顺序梯形法（Sequential Trapezoidal Rule）可以通过如下步骤实现：

- 初始值

$$M(h_0) = (b-a)f\left(a+\frac{b-a}{2}\right)$$

- 递推关系

$$R(J,K) = R(J,K-1) + \frac{R(J,K-1) - R(J-1,K-1)}{4^K - 1}$$

Question 6

根据给定代码，使用如下 MATLAB 代码进行调用并计算，其中f_1, f_2分别表示两个定积分的被积部分

```
function y=f1(x)
    y = sqrt(4*x-x*x);
end

function y=f2(x)
    y=4/(1+x*x);
end

for i = 1:22
    [R1, quad1, err1, h1]=romber(@f1, 0, 2, i, 1e-10);
    [R2, quad2, err2, h2]=romber(@f2, 0, 1, i, 1e-10);
    fprintf('n=%d,  quad1=%d,  err1=%d, quad2=%d, err2=%d\n', i, quad1,err1, quad2,
err2);
end
```

输出结果如下：

```
n=1,  quad1=3.135506e+00,  err1=1.128802e-02, quad2=3.141593e+00, err2=6.881516e-06
n=2,  quad1=3.135506e+00,  err1=1.128802e-02, quad2=3.141593e+00, err2=6.881516e-06
n=3,  quad1=3.135506e+00,  err1=1.128802e-02, quad2=3.141593e+00, err2=6.881516e-06
n=4,  quad1=3.135506e+00,  err1=1.128802e-02, quad2=3.141593e+00, err2=6.881516e-06
n=5,  quad1=3.139447e+00,  err1=3.940566e-03, quad2=3.141593e+00, err2=1.163947e-08
n=6,  quad1=3.140835e+00,  err1=1.387929e-03, quad2=3.141593e+00, err2=4.852119e-11
n=7,  quad1=3.141325e+00,  err1=4.900872e-04, quad2=3.141593e+00, err2=4.852119e-11
n=8,  quad1=3.141498e+00,  err1=1.731897e-04, quad2=3.141593e+00, err2=4.852119e-11
n=9,  quad1=3.141559e+00,  err1=6.121967e-05, quad2=3.141593e+00, err2=4.852119e-11
n=10,  quad1=3.141581e+00,  err1=2.164249e-05, quad2=3.141593e+00, err2=4.852119e-11
n=11,  quad1=3.141588e+00,  err1=7.651452e-06, quad2=3.141593e+00, err2=4.852119e-11
n=12,  quad1=3.141591e+00,  err1=2.705141e-06, quad2=3.141593e+00, err2=4.852119e-11
n=13,  quad1=3.141592e+00,  err1=9.564022e-07, quad2=3.141593e+00, err2=4.852119e-11
n=14,  quad1=3.141592e+00,  err1=3.381375e-07, quad2=3.141593e+00, err2=4.852119e-11
n=15,  quad1=3.141593e+00,  err1=1.195494e-07, quad2=3.141593e+00, err2=4.852119e-11
n=16,  quad1=3.141593e+00,  err1=4.226703e-08, quad2=3.141593e+00, err2=4.852119e-11
n=17,  quad1=3.141593e+00,  err1=1.494365e-08, quad2=3.141593e+00, err2=4.852119e-11
n=18,  quad1=3.141593e+00,  err1=5.283361e-09, quad2=3.141593e+00, err2=4.852119e-11
n=19,  quad1=3.141593e+00,  err1=1.867987e-09, quad2=3.141593e+00, err2=4.852119e-11
n=20,  quad1=3.141593e+00,  err1=6.604162e-10, quad2=3.141593e+00, err2=4.852119e-11
n=21,  quad1=3.141593e+00,  err1=2.335150e-10, quad2=3.141593e+00, err2=4.852119e-11
n=22,  quad1=3.141593e+00,  err1=8.257572e-11, quad2=3.141593e+00, err2=4.852119e-11
```

要求精度达到小数点后 10 位，即 err 小于$1e-10$，由上面结果可知：

- $n = 22$时, $f_1(x) \approx \mathrm{quad1} = 3.141593$，误差为$\mathrm{err1} = 8.257572e-11$
- $n = 6$时, $f_2(x) \approx \mathrm{quad2} = 3.141593$，误差为$\mathrm{err1} = 4.852119e-11$

比较收敛速率，可以发现第二个定积分收敛度更快。