Q1. 习题 4 第 4 题

用 integral 求取$\int_{-5\pi}^{10\pi} \mathrm{e}^{-|x|}|\sin x|\mathrm{d}x$的绝对精度为$10^{-9}$的广义积分，并尝试用 trapz 及符号计算此积分。(提示：注意 integral 指令相对误差控制对绝对精度的影响。本题假设符号计算保留 16 位小数为真实值，trapz 步长自定)

答：首先利用符号计算获取一个近似的真实值

```
syms x y(x)
y(x)=exp(-abs(x))*abs(sin(x));
si=vpa(int(y,-5*pi,10*pi),16)
si =
   1.090331328569942
```

In[1]:= $\int_{-5\pi}^{10\pi} \mathrm{e}^{-\mathrm{Abs}[x]}\,\mathrm{Abs}[\mathrm{Sin}[x]]\,\mathrm{d}x$

正弦

Out[1]= $\frac{1}{2}\mathrm{e}^{-10\pi}(1+\mathrm{e}^{\pi})(1+\mathrm{e}^{\pi}+\mathrm{e}^{2\pi}+\mathrm{e}^{3\pi}+\mathrm{e}^{4\pi}+2\mathrm{e}^{5\pi}+2\mathrm{e}^{6\pi}+2\mathrm{e}^{7\pi}+2\mathrm{e}^{8\pi}+2\mathrm{e}^{9\pi})$

然后使用函数 integral 获得较精确的数值积分解

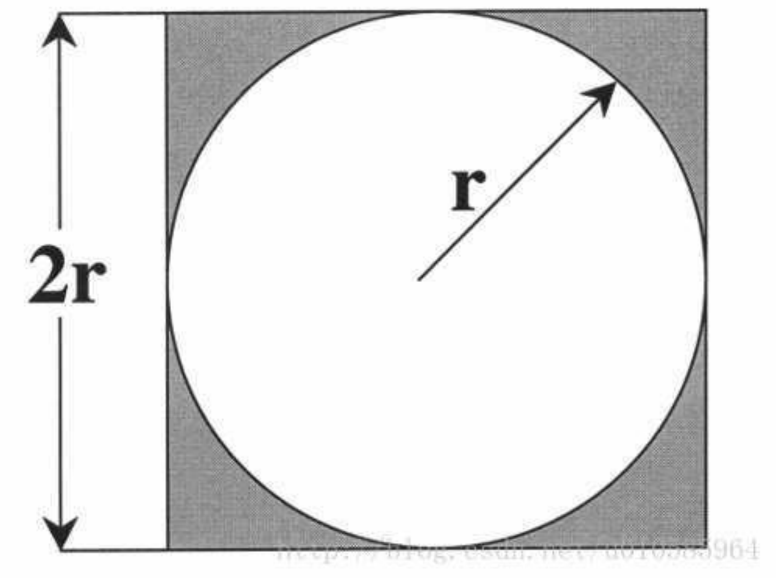
```
fx=@(x)exp(-abs(x)).*abs(sin(x));
format long;
s=integral(fx,-5*pi,10*pi,'Abstol',1e-9)   %如果要提高精度还需减小 Restol
s =
   1.090331540770801
format short e
err1 = double(abs(s - si))
err1 =
   2.1220e-07
```

最后使用粗略的梯形公式数值积分函数 trapz 计算此积分，小步长速度慢，但误差更小

```
x=linspace(-5*pi,10*pi,1e5);
format long;
dx=x(2)-x(1);
st=trapz(exp(-abs(x)).*abs(sin(x)))*dx
st =
   1.090331292961167
format short e
err2 = double(abs(st - si))
err2 =
   3.5609e-08
```

实验结果表明，虽然 integral 在相同步长精度下往往误差更小，在相同误差情况下效率更高，但是如果 trapz 函数设定了更小的间距或更多的分隔步骤，可以获得更高的精度。

Q2.π的估计，如图，若设$r=\frac{1}{2}$,且圆心为$(\frac{1}{2},\frac{1}{2})$可生成$1\times10^6$组符合$(0,1)$均匀分布的横坐标$X$与纵坐标$Y$。易知$(X,Y)$恰好落在正方形区域内。此时统计落在圆形区域$\left(x-\frac{1}{2}\right)^2+\left(y-\frac{1}{2}\right)^2\le\frac{1}{4}$内的点数$N_a$,即可近似估计$\pi\approx4\cdot\frac{N_a}{N}$。用 MATLAB 实现这段代码。



答：本题只需将对应的算法实现即可，注意：无需使用 for 循环否则会降低运行的效率！

```
rng default;    %为了结果可以再现，其实并不必要
x = rand(1,1e6);
y = rand(1,1e6);
r2 = (x-1/2).^2 + (y-1/2).^2;
Na = sum(r2<=1/4);  % ra<=1/4为逻辑矩阵，符合条件的为1，求和即为计数
pi_app = 4*Na/1e6

pi_app =
   3.141648000000000
```

Q3.使用 diff 与 gradient 函数，计算$f(x)=\ln(1+x),0\le x\le2$的近似导数（步长$x=1\times10^{-5}$），然后比较 diff 与 gradient 在$f'(1)$的误差，以分析方法优劣性。

答：显然$f'(x)=\frac{1}{1+x},0\le x\le2$

```
d = 1e-5;
x = 0:d:2;
fx = log(1+x);                    %函数的录入
fp1 = diff(fx)./d;                %diff 求导
fp2 = gradient(fx)./d;            %gradient 求导
fpt = 1./(1+x);                   %导数的真实值
err1 = abs(fp1(1e5+1) - fpt(1e5+1))
%fp1 可以取第 100000 个点的值，不算错
```

```
err1 =
   1.2500e-06
err2 = abs(fp2(1e5+1) - fpt(1e5+1))
err2 =
   8.8267e-12
```

从实验结果可见，中心差分的误差明显小于向前差分，而相邻两点的向前差分平均事实上就等于中心差分的结果。

Q4.（选做）用理论与数值算法计算下列二重积分（高数难度）

$$\iint_D \frac{x+y}{\sqrt{(1+x^2+y^2)^3}}\mathrm{d}x\mathrm{d}y,\ \{D: 0\le x\le 1, 0\le y\le 1\}$$

（1）利用已学数学分析的知识尝试解出这个问题

答：为了节省运算时间，本题可以使用轮换对称性，再化为二次积分即可，即

$$\iint_D \frac{x+y}{\sqrt{(1+x^2+y^2)^3}}\mathrm{d}x\mathrm{d}y = 2\iint_D \frac{y}{\sqrt{(1+x^2+y^2)^3}}\mathrm{d}x\mathrm{d}y$$

$$= 2\int_0^1 \mathrm{d}x\int_0^1 \frac{y}{\sqrt{(1+x^2+y^2)^3}}\mathrm{d}y = 2\int_0^1 \left.\frac{-1}{\sqrt{1+x^2+y^2}}\right|_{y=0}^{y=1}\mathrm{d}x$$

$$= 2\int_0^1 \frac{1}{\sqrt{1+x^2}} - \frac{1}{\sqrt{2+x^2}}\mathrm{d}x = 2\left[\ln\left|x+\sqrt{1+x^2}\right| - \ln\left|x+\sqrt{2+x^2}\right|\right]\Big|_{x=0}^{x=1} \text{(二类换元或查表)}$$

$$= 2\left[\ln(1+\sqrt{2}) - \ln 1 - \ln(1+\sqrt{3}) + \ln\sqrt{2}\right] = 2\ln\frac{2+\sqrt{2}}{1+\sqrt{3}} = 2\ln\left(\frac{\sqrt{2}}{2}-\sqrt{3}-\frac{\sqrt{6}}{2}+1\right)$$

附 MATLAB 符号运算方法：

```
syms x y f(x,y)
f(x,y) = (x+y)/sqrt(1+x^2+y^2)^3;
gt = int(int(f(x,y),y,0,1),x,0,1)
gt =
  log((2^(1/2)/2 - 3^(1/2) - 6^(1/2)/2 + 1)^2)
```

（2）数值运算部分需要分别使用二维中矩形方法（即正方形剖分法，将积分正方形区域按 2000×2000 的倍数均匀分成四百万个正方形，对每个正方形取中点函数值即可），二维复合辛普森公式（仍为 2000×2000 个网格，网格内套用辛普森公式，具体公式设置可查阅课外资料）进行计算，观察结果与（1）所得答案的误差，分析算法优劣性。

答：中矩形算法的原理较简单，只需将每个网格中心的函数值，乘上网格面积，求和即可

```
%% 中矩形算法
f = @(x,y) ((x+y)./sqrt((1+x.^2+y.^2).^3));
%匿名函数既可以用于符号运算，也可以用于数值运算
d = 1/2000;
x = d/2:d:1-d/2;y = x;
intmr = sum(sum(f(x',y)))*d*d;
err1 = abs(double(gt) - intmr)
```

```
err1 =
    2.009240007705415e-08
```

二维复合辛普森公式与一维的情况类似，因为本质上其实就是对x与y方向分别进行一次累计。所以将一维复合辛普森公式进行推广，即可得到，对于每一个正方形网格块，四角的权值为$\frac{1}{6}\times\frac{1}{6}=\frac{1}{36}$，四条边中点的权值为$\frac{1}{6}\times\frac{2}{3}=\frac{1}{9}$，而正方形中点的权值为$\frac{2}{3}\times\frac{2}{3}=\frac{4}{9}$，按照这种思路，可以完成下面的代码：

```
%% 复合辛普森公式
f = @(x,y) ((x+y)./sqrt((1+x.^2+y.^2).^3));
d = 1/2000;
x = d/2:d:1-d/2;y = x;
intmr = sum(sum(f(x',y)))*d*d*4/9;                  % 中心点权值4/9
intmr = intmr + sum(sum(f((x-d/2)',y)))*d*d*1/9;  % 左边中点权值1/9
intmr = intmr + sum(sum(f((x+d/2)',y)))*d*d*1/9;  % 右边中点权值1/9
intmr = intmr + sum(sum(f(x',(y-d/2))))*d*d*1/9;  % 下边中点权值1/9
intmr = intmr + sum(sum(f(x',(y+d/2))))*d*d*1/9;  % 上边中点权值1/9
intmr = intmr + sum(sum(f((x-d/2)',(y-d/2))))*d*d*1/36;
intmr = intmr + sum(sum(f((x-d/2)',(y+d/2))))*d*d*1/36;
intmr = intmr + sum(sum(f((x+d/2)',(y-d/2))))*d*d*1/36;
intmr = intmr + sum(sum(f((x+d/2)',(y+d/2))))*d*d*1/36;
% 四个角的权值都是1/36
err2 = abs(double(gt) - intmr)
err2 =
    6.106226635438361e-16
```

从结果观察，辛普森公式的误差几乎达到了机器误差 eps 级别。这是因为中矩形公式的代数精度为 1，而辛普森公式的代数精度为 3，因此，再之后的数值积分计算中，普通的重积分（非瑕积分）基本可以使用辛普森公式达到很好的代数精度。

本题不仅考察方法的再现能力，同时也间接考察数组化编程，在代码的实现中，滥用 for 循环可能会被扣 0.2~0.5 分。