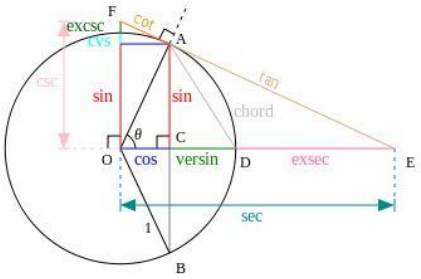
Matplotlib 快速绘图　课后作业

（命题人：极值学院讲师 张阳阳）

作业：使用 Python 编程，完成下列题目，在课程的讨论区提交每题的代码和结果。

1. 三角学传入中国，开始于明崇祯 4 年(1631 年)，这一年，邓玉函、汤若望和徐光启合编《大测》，作为历书的一部份呈献给朝廷，这是我国第一部编译的三角学。在《大测》中，首先将 sine 译为”正半弦”，简称”正弦”，这就成了“正弦”一词的由来。请你在同一个图中画出正矢函数 $ver\sin\theta = 1-\cos\theta$ 和半余矢函数 $ha\cos ver\sin\theta = \frac{1-\sin\theta}{2}$ 的图像。



作图要求如下：

（1）在 0~20 内等间距取 1000 个点作图；

（2）绘图对象的宽度为 8 英寸，高度为 4 英寸；

（3）对于正矢函数，标上图例“versin(x)”，线条为红色，厚度为 2，实线；

（4）对于半余矢函数，标上图例“hacoversin(x)”，线条为蓝色，虚线；

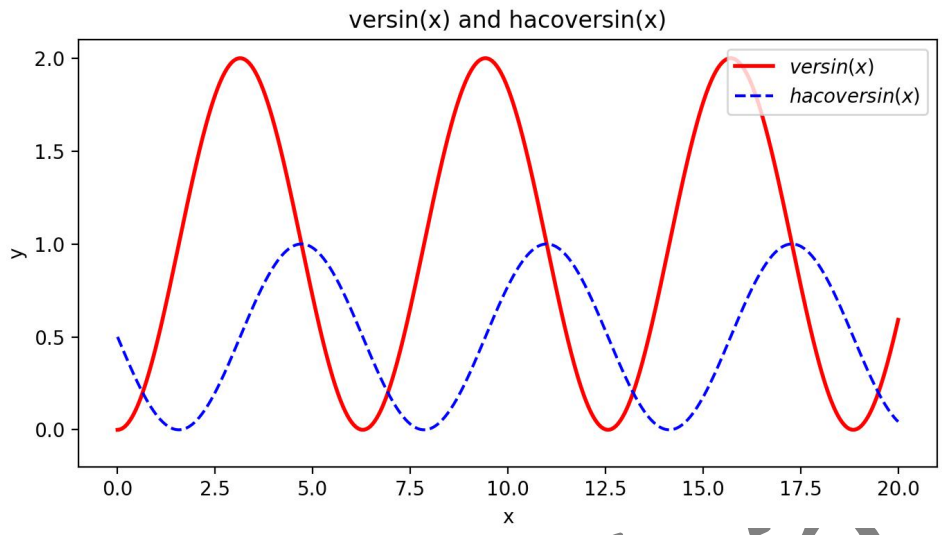
（5）横坐标为“x”，纵坐标为“y”，标题为“versin(x) and hacoversin(x)”；

（6）纵坐标 y 的显示范围为-0.2~2.1；

（7）保存图片名称为“versin(x) and hacoversin(x)”，分辨率为 dpi=200。

```
import numpy as np
import matplotlib.pyplot as plt
x = np.linspace(0, 20, 1000)
y = 1-np.cos(x)
z = (1-np.sin(x))/2
plt.figure(figsize=(8,4))
plt.plot(x,y,label="$versin(x)$",color="red",linewidth=2)
plt.plot(x,z,"b--",label="$hacoversin(x)$")
plt.xlabel("x")
plt.ylabel("y")
plt.title("versin(x) and hacoversin(x)")
plt.ylim(-0.2,2.1)
plt.legend()
```

```
plt.savefig("versin(x) and hacoversin(x)",dpi=200)
plt.show()
```



2. 在《Python 机器学习与数据挖掘实践》中级课程和《Python 深度学习与数据挖掘实战》高级课程中，会讲到人工智能中的神经网络算法，Sigmoid 函数是神经网络算法中常用的非线性的激活函数，它的数学形式如下：

$$f(x)=\frac{1}{1+e^{-x}}$$

作图要求如下：

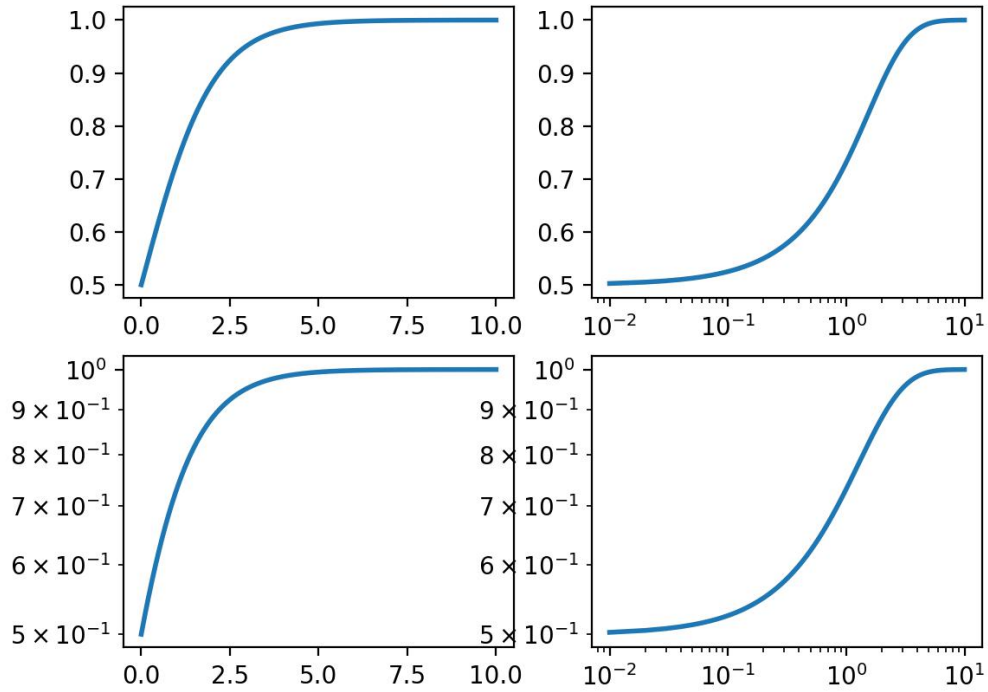
（1）在 0~10 内等间距取 1000 个点作图；

（2）绘制多轴图和对数坐标图：左上图为 plot()绘制的算术坐标系，右上图为 semilogx()绘制的 X 轴对数坐标系，左下图为 semilogy()绘制的 Y 轴对数坐标系，右下图为 loglog()绘制的双对数坐标系；

（3）保存图片名称为“Sigmoid”，分辨率为 dpi=200。

```
import numpy as np
import matplotlib.pyplot as plt
w = np.linspace(0, 10, 1000)
p = 1/(1 + np.exp(-w))
plt.subplot(221)
plt.plot(w, p, linewidth=2)
plt.subplot(222)
plt.semilogx(w, p, linewidth=2)
plt.subplot(223)
plt.semilogy(w, p, linewidth=2)
plt.subplot(224)
plt.loglog(w, p, linewidth=2)
plt.show()
```

```
plt.savefig("Sigmoid",dpi=200)
```



3. 数学家笛卡尔的爱情故事。笛卡尔于 1596 年出生在法国，欧洲大陆爆发黑死病时他流浪到瑞典，认识了瑞典一个小公国 18 岁的公主克里斯汀，后成为她的数学老师，日日相处使他们彼此产生爱慕之心，公主的父亲国王知道了后勃然大怒，下令将笛卡尔处死，后因女儿求情将其流放回法国，克里斯汀公主也被父亲软禁起来。笛卡尔回法国后不久便染上重病，他日日给公主写信，因被国王拦截，克里斯汀一直没收到笛卡尔的信。笛卡尔在给克里斯汀寄出第十三封信后就气绝身亡了，这第十三封信内容只有短短的一个公式：

$$\rho = a(1-\sin\theta)$$

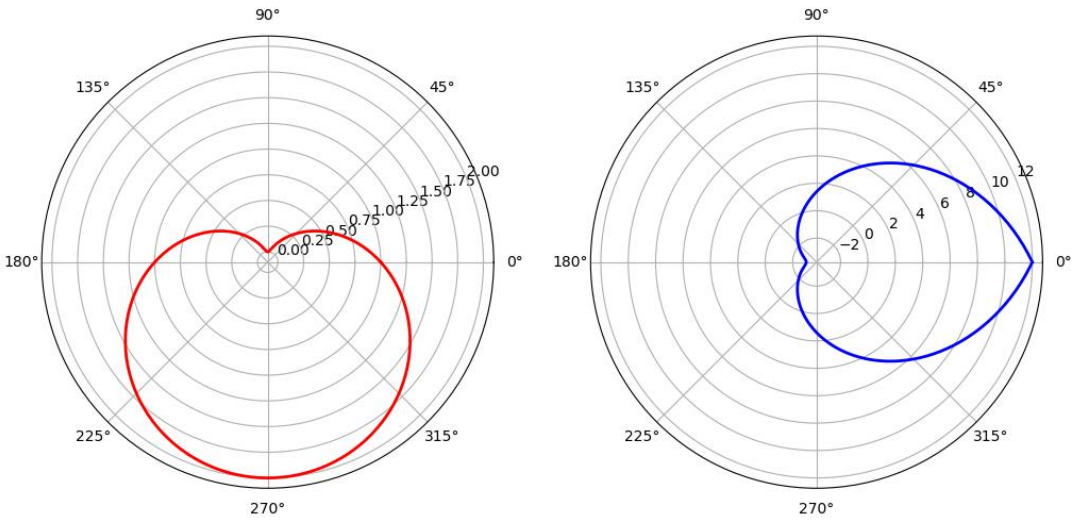
国王看不懂，觉得他们俩之间并不是总是说情话的，大发慈悲就把这封信交给一直闷闷不乐的克里斯汀，公主看到后，立即明了恋人的意图，她马上着手把方程的图形画出来，看到图形，她开心极了，她知道恋人仍然爱着她，原来方程的图形是一颗心的形状。这也就是著名的“心形线”。

国王死后，克里斯汀登基，立即派人在欧洲四处寻找心上人，无奈斯人已故，先她走一步了，徒留她孤零零在人间。据说这封享誉世界的另类情书还保存在欧洲笛卡尔的纪念馆里。

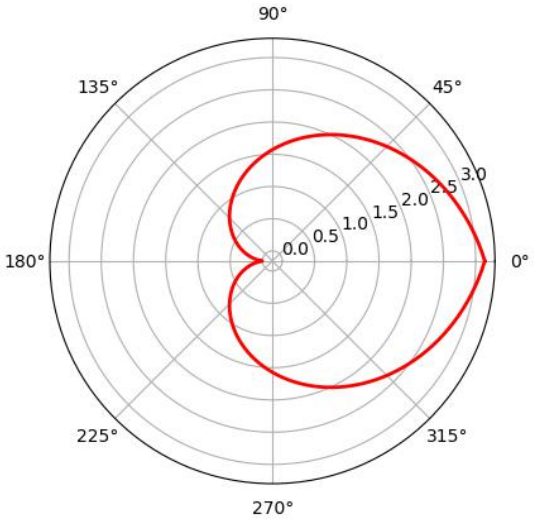
（1）请你在极坐标中左图画出“心形线” $\rho = 1-\sin\theta$ 的函数图像，右图画出“心形线” $\rho = 3(4-5\sin\theta)$ 的函数图像。作图要求如下：左图中线条为红色，厚度为 2，右图中线条为蓝色，厚度为 2，保存图片名称为“笛卡尔心形线 1”，分辨率为 dpi=300。

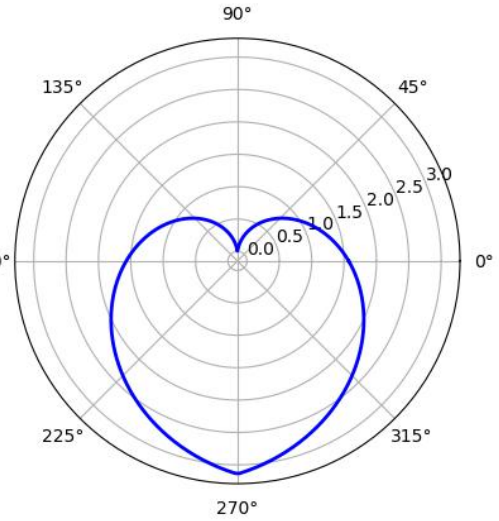
（2）极坐标系下绘制 $\rho=\arccos(\sin\theta)$ ，我们也会得的一个漂亮的心形线。请你在极坐标中左图画出“心形线” $\rho=2\arccos(0.5\sin\theta)$ 的函数图像，右图画出“心形线” $\rho=\arccos(\sin\theta)$ 的函数图像。作图要求如下：左图中线条为红色，厚度为 2，右图中线条为蓝色，厚度为 2，保存图片名称为“笛卡尔心形线 1”，分辨率为 dpi=300。

```
import numpy as np
import matplotlib.pyplot as plt
theta = np.arange(0, 2*np.pi, 0.02)
plt.subplot(121, polar=True)
plt.plot(theta, 1*(1-np.sin(theta)), 'r', linewidth=2)
plt.subplot(122, polar=True)
plt.plot(theta, 3*(4-5*np.sin(0.5*theta)), 'b', linewidth=2)
plt.savefig("笛卡尔心形线 1",dpi=300)
plt.show()
```




```
import numpy as np
import matplotlib.pyplot as plt
theta = np.arange(0, 2*np.pi, 0.02)
plt.subplot(121, polar=True)
plt.plot(theta, 2*np.arccos(np.sin(0.5*theta)), 'r', linewidth=2)
plt.subplot(122, polar=True)
plt.plot(theta, np.arccos(np.sin(theta)), 'b', linewidth=2)
plt.savefig("笛卡尔心形线 2",dpi=300)
plt.show()
```

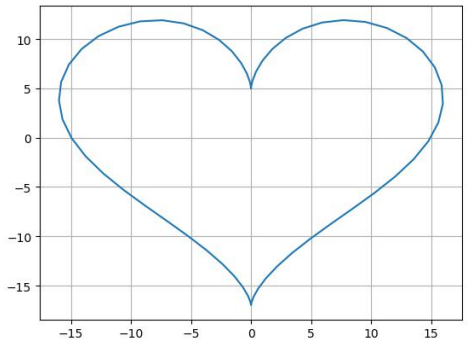




4. 在平面直角坐标系中画出心形线 $\begin{cases} x = 16\sin^3 t \\ y = 13\cos t - 5\cos 2t - 2\cos 3t - \cos 4t \end{cases} (0 \le t \le 2\pi)$ 参数方程表达的图形，并计算出心形线（heart-line）$\rho = a(1+\cos\theta)$ 围成的平面区域的面积。

解：先看 heart-line 的图形，它也是一个对称图形，只需要计算其中的二分之一区域的面积。

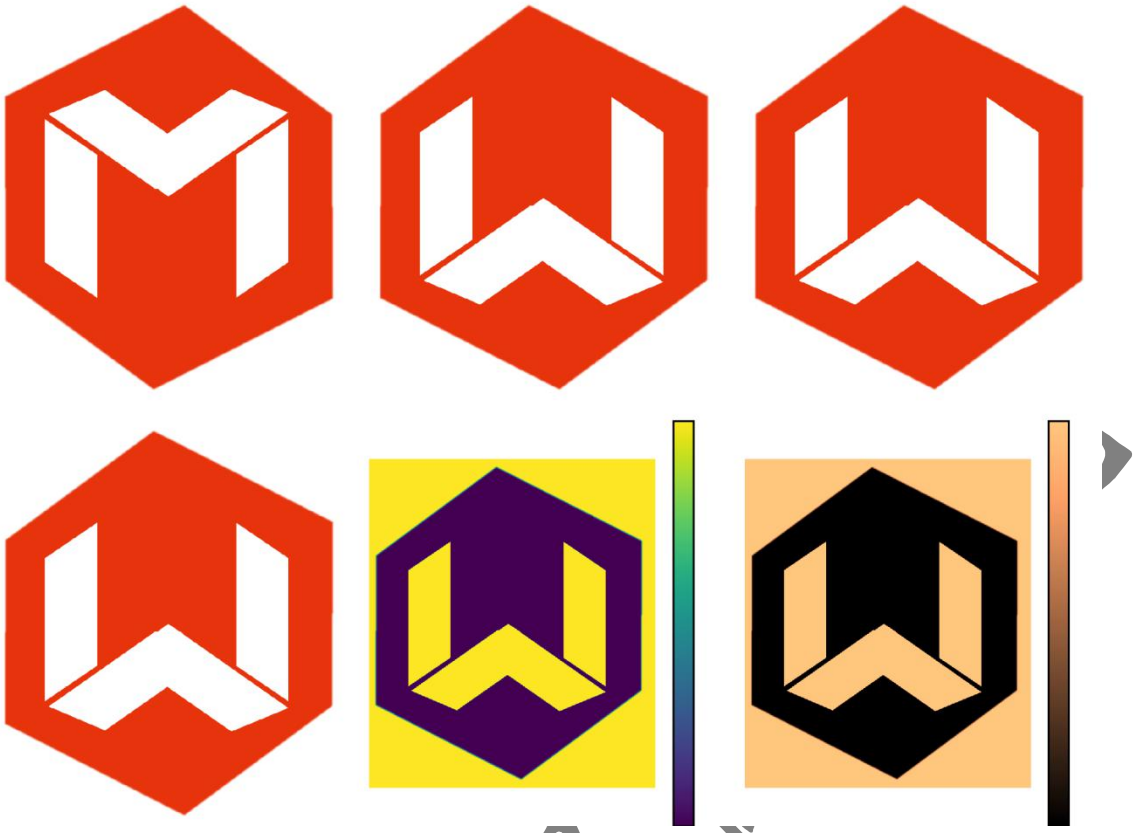
```
import matplotlib.pyplot as plt
import numpy as np
t = np.arange(0,2*np.pi, 0.1)
x = 16*np.sin(t)**3
y = 13*np.cos(t)-5*np.cos(2*t)-2*np.cos(3*t)-np.cos(4*t)
plt.plot(x,y)
plt.grid()
plt.show()
```




```
from sympy import *
t, a = symbols('t a')
rho = a * (1 + cos(t))
y=integrate(1 / 2 * rho ** 2, (t, 0, pi))
print(y)
# 0.75*pi*a**2
```

5. 使用 Python 编程。读取附件中所给的“极值学院.PNG”图像数据，然后显示出来，创建六幅子图，第一幅图只读取图像并展示，第二幅图反转图像数组的第 0 轴，第三幅图让图表的原点在左下角，第四幅图变成取值范围为 0.0 到 255.0 的浮点数组并展示，第五幅图显示图像中的红色通道，将最小值映射为蓝色、将最大值映射为红色.，使用 colorbar()将颜色映射表在图表中显示出来，第六幅图使用名为 copper 的颜色映射表显示图像的红色通道，使其有老照片的味道，最后保存图片名称为“极值学院 logo”，分辨率为 dpi=300。

```
import matplotlib.pyplot as plt
import matplotlib.cm as cm
plt.subplots_adjust(0,0,1,1,0.05,0.05)
plt.subplot(231)
img = plt.imread("极值学院.png")
plt.imshow(img)
plt.subplot(232)
plt.imshow(img[::-1])
plt.subplot(233)
plt.imshow(img, origin="lower")
img = img[::-1]
plt.subplot(234)
plt.imshow(img*1.0)
plt.subplot(235)
plt.imshow(img[:,:,0])
plt.colorbar()
plt.subplot(236)
plt.imshow(img[:,:,0], cmap=cm.copper)
plt.colorbar()
for ax in plt.gcf().axes:
    ax.set_axis_off()
    ax.set_axis_off()
plt.savefig("极值学院 logo",dpi=300)
plt.show()
```



6. 薛定谔方程（Schrödinger equation）又称薛定谔波动方程（Schrödinger wave equation），是由奥地利物理学家薛定谔提出的量子力学中的一个基本方程，也是量子力学的一个基本假设。下面是三维运动自由粒子的含时薛定谔方程：

$$i\hbar\frac{\partial\Psi}{\partial t}=\left[-\frac{\hbar^2}{2m}\nabla^2+U(\vec{r},t)\right]\Psi$$

求解薛定谔方程，已知粒子在一维无限深势阱中运动，其基态波函数为二元函数：

$$\Psi(x,t)=\sqrt{\frac{2}{a}}\sin\left(\frac{\pi}{a}x\right)e^{-i\frac{\pi^2\hbar}{2ma^2}t}$$

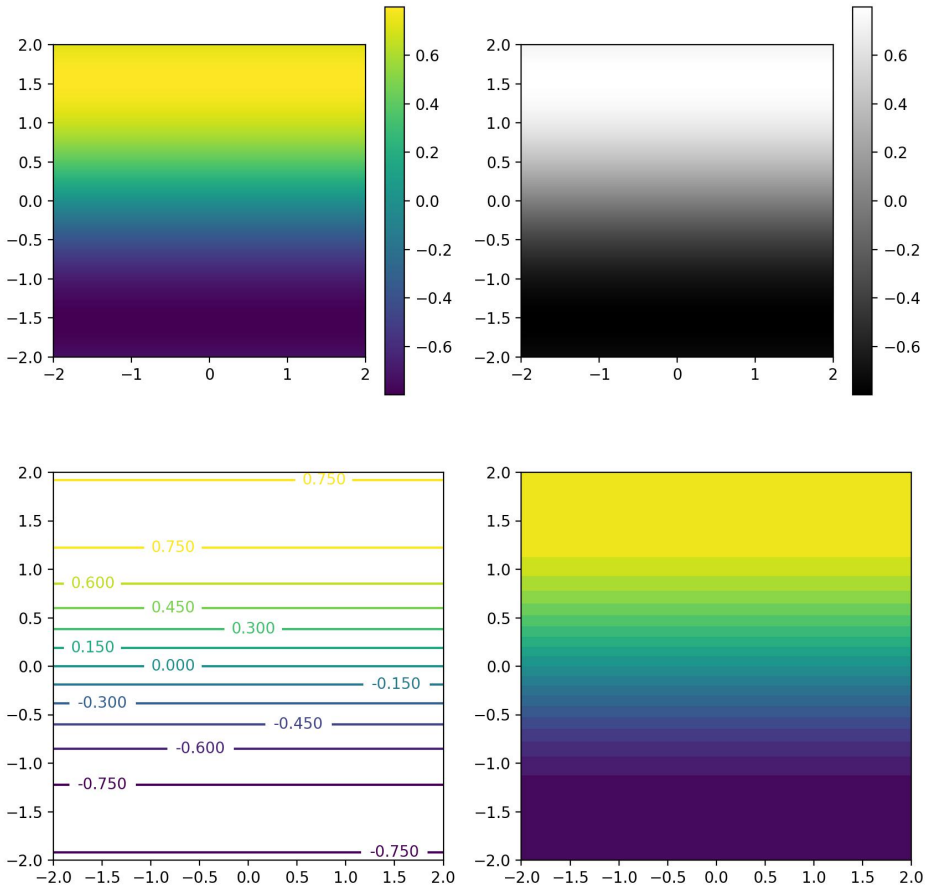
令 $a=\pi$，$m=1$，$\hbar$ 为普朗克常数，在作图时忽略上述二元函数中的虚数 i，使用 imshow()可视化上述基态波函数二元函数。创建四幅子图，首先通过数组的广播功能计算出表示函数值的二维数组 Z，注意它的第 0 轴表示 Y 轴、 第 1 轴表示 X 轴。然后将 X、Y 轴的取值范围保存到 extent 列表中。将 extent 列表传递给 imshow()的 extent 参数，这样一来，图表的 X、Y 轴的刻度标签将使用 extent 列表所指定的范围。

作图要求如下：

第一幅图中画出二元函数图像，横纵坐标的范围均为-2.0~2.0；第二幅图中画出灰度图；第三幅图中调用 contour()，将整个函数的取值范围等分为 10 个区间，即显示的等值线图中

将有 9 条等值线；第四幅图中调用 contourf()，将取值范围等分为 20 份，绘制带填充效果的等值线图，最后保存图片名称为“Schrödinger”，分辨率为 dpi=200。

```
import numpy as np
from math import pi
from scipy import constants as const
import matplotlib.pyplot as plt
import matplotlib.cm as cm
x, t = np.ogrid[-2:2:200j, -2:2:200j]
a = pi
z = np.sqrt(2/a)*np.sin(pi/a*x)*np.exp(-1*pi**2*const.h*t/2/a**2)
extent = [np.min(x), np.max(x), np.min(t), np.max(t)]
plt.figure(figsize=(10,10))
plt.subplot(221)
plt.imshow(z, extent=extent, origin="lower")
plt.colorbar()
plt.subplot(222)
plt.imshow(z, extent=extent, cmap=cm.gray, origin="lower")
plt.colorbar()
plt.subplot(223)
cs = plt.contour(z, 10, extent=extent)
plt.clabel(cs)
plt.subplot(224)
plt.contourf(x.reshape(-1), t.reshape(-1), z, 20)
plt.savefig("Schrödinger",dpi=200)
plt.show()
```



7. 可以使用等值线绘制隐函数曲线，显然，无法像绘制一般函数那样，先创建一个等差数组表示变量的取值点，然后计算出数组中每个 x 所对应的 y 值。可以使用等值线解决这个问题，显然隐函数的曲线就是值等于 0 的那条等值线。请你绘制函数：

$$f(x,y)=\left(x^2+y^2\right)^4-\left(x^2-y^2\right)^3$$

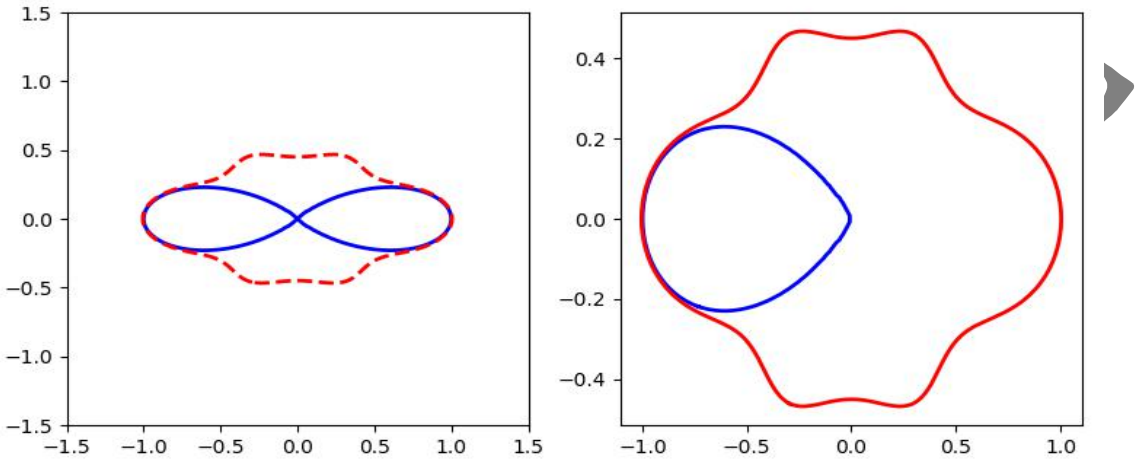
在 $f(x,y)=0$ 和 $f(x,y)=0.01$ 时的曲线。

```
import numpy as np
import matplotlib.pyplot as plt

y, x = np.ogrid[-1.5:1.5:200j, -1.5:1.5:200j]
f = (x**2 + y**2)**4 - (x**2 - y**2)**3
plt.figure(figsize=(9, 4))
plt.subplot(121)
extent = [np.min(x), np.max(x), np.min(y), np.max(y)]
cs = plt.contour(f, extent=extent, levels=[0, 0.01],
```

```
                        colors=["b", "r"], linestyles=["solid", "dashed"], linewidths=[2, 2])

    plt.subplot(122)
    for c in cs.collections:
        data = c.get_paths()[0].vertices
        plt.plot(data[:, 0], data[:, 1],
                 color=c.get_color()[0], linewidth=c.get_linewidth()[0])
    plt.show()
```



8. mpl_toolkits.mplot3d 模块在 matplotlib 基础上提供了三维绘图的功能。请你使用 matplotlib 绘制下列函数的三维曲面：

$$f(x,y)=-x^2+xy-y^2$$

作图要求如下：

（1）x 和 y 的取值范围为-3~3，分别取 200 个点，三维曲面用红色绘制；

（2）横坐标为“X”，纵坐标为“Y”，z 坐标为“Z”；

（3）最后保存图片名称为“3d”，分辨率为 dpi=200。

```
    import numpy as np
    import mpl_toolkits.mplot3d
    import matplotlib.pyplot as plt
    x, y = np.mgrid[-3:3:200j, -3:3:200j]
    z = x * np.exp( - x**2 + x*y - y**2)
    ax = plt.subplot(111, projection='3d')
    ax.plot_surface(x, y, z, rstride=2, cstride=1, cmap = plt.cm.Reds_r)
    ax.set_xlabel("X")
    ax.set_ylabel("Y")
    ax.set_zlabel("Z")
    plt.savefig("3d",dpi=200)
    plt.show()
```

9. 调用 Axes3D 对象的 plot_surface()绘制三维曲面。其中：数组 x 和 y 构成了 X-Y 平面上的网格，而数组 z 则是网格上各点在曲面上的取值。通过 cmap 参数来指定值和颜色之间的映射，即曲面上各点的高度值与其颜色的对应关系。请你画出下列多元函数的三维曲面图：

$$z(x,y)=\sin(\sqrt{|x^2-y^2|})$$

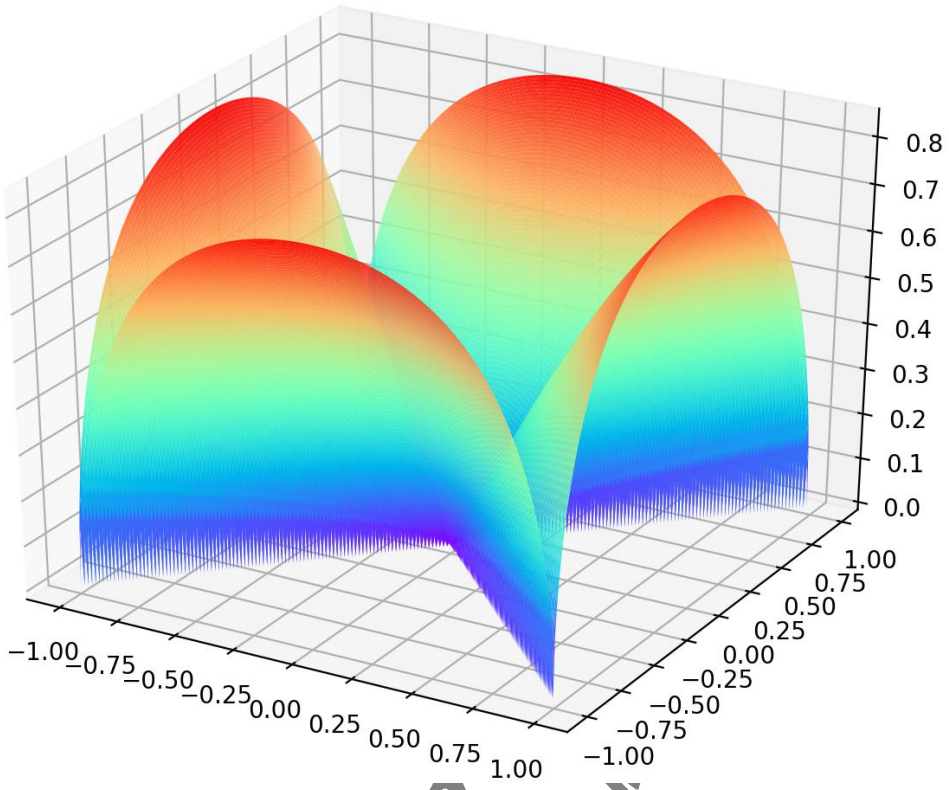
作图要求如下：

（1）x 和 y 的取值范围为-1.0~1.0，每隔 0.01 取一个数；

（2）cmap 参数选择彩虹图“rainbow”；

（3）最后保存图片名称为“rainbow”，分辨率为 dpi=200。

```
from mpl_toolkits.mplot3d import Axes3D
import matplotlib.pyplot as plt
import numpy as np
fig = plt.figure()
ax = Axes3D(fig)
X = np.arange(-1, 1, 0.01)
Y = np.arange(-1, 1, 0.01)
X, Y = np.meshgrid(X, Y)
R = np.sqrt(abs(X**2 - Y**2))
Z = np.sin(R)
ax.plot_surface(X, Y, Z, rstride=1, cstride=1, cmap='rainbow')
plt.savefig("rainbow",dpi=200)
plt.show()
```



10. 调用 Axes3D 对象的 plot_surface()绘制三维曲面。其中：数组 x 和 y 构成了 X-Y 平面上的网格，而数组 z 则是网格上各点在曲面上的取值。通过 cmap 参数来指定值和颜色之间的映射，即曲面上各点的高度值与其颜色的对应关系。请你画出下列多元函数的三维曲面图：

$$z(x,y)=\sin(\pi\sqrt{x^2+y^2})$$

作图要求如下：

（1）x 和 y 的取值范围为-5.0~5.0，每隔 0.1 取一个数；

（2）cmap 参数选择热图“hot”；

（3）最后保存图片名称为“hot map”，分辨率为 dpi=300。

```
from mpl_toolkits.mplot3d import Axes3D
import matplotlib.pyplot as plt
import numpy as np
fig = plt.figure()
ax = Axes3D(fig)
X = np.arange(-5, 5, 0.1)
Y = np.arange(-5, 5, 0.1)
X, Y = np.meshgrid(X, Y)
R = np.sqrt(X**2 + Y**2)
Z = np.sin(np.pi*R)
ax.plot_surface(X, Y, Z, rstride=1, cstride=1, cmap='hot')
```

```
plt.savefig("hot map",dpi=300)
plt.show()
```