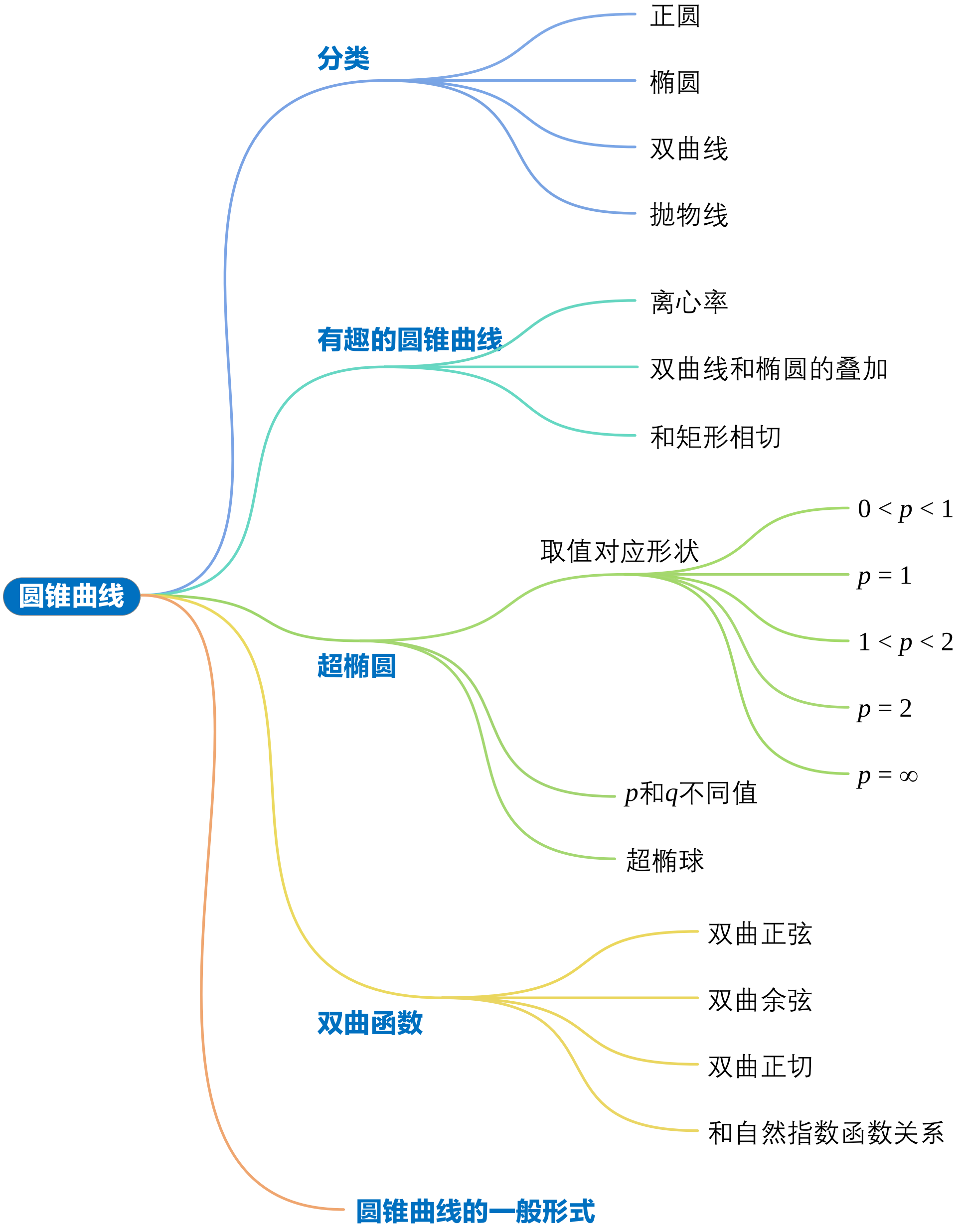
9 Dive into Conic Sections
深入圆锥曲线
探寻和数据科学、机器学习之间联系

> 地球是人类的摇篮，但我们不能永远生活在摇篮里。
>
> ***Earth is the cradle of humanity, but one cannot live in a cradle forever.***
>
> —— 康斯坦丁·齐奥尔科夫斯基 (Konstantin Tsiolkovsky) | 俄罗斯火箭专家 | 1857 ~ 1935

- ◄ matplotlib.patches.Rectangle() 绘制通过定位点，以及设定宽度和高度的矩形
- ◄ matplotlib.pyplot.contour() 绘制等高线图
- ◄ matplotlib.pyplot.contourf() 绘制填充等高线图
- ◄ numpy.cosh() 双曲余弦函数
- ◄ numpy.isinf() 判断是否存在无穷
- ◄ numpy.maximum() 计算最大值
- ◄ numpy.sinh() 双曲正弦函数
- ◄ numpy.tanh() 双曲正切函数
- ◄ sympy.Eq() 定义符号等式
- ◄ sympy.evalf() 将符号解析式中未知量替换为具体数值
- ◄ sympy.plot_implicit()绘制隐函数方程
- ◄ sympy.symbols() 定义符号变量


圆锥曲线
分类
正圆
椭圆
双曲线
抛物线
有趣的圆锥曲线
离心率
双曲线和椭圆的叠加
和矩形相切
超椭圆
取值对应形状
$0 < p < 1$
$p = 1$
$1 < p < 2$
$p = 2$
$p = \infty$
p和q不同值
超椭球
双曲函数
双曲正弦
双曲余弦
双曲正切
和自然指数函数关系
圆锥曲线的一般形式

9.1 圆锥曲线：探索星辰大海

虽然正圆、椭圆、抛物线、双曲线这样的数学概念见诸于中学课本，但是它们在现如今依旧展现着巨大能量；比如，在星辰大海的征途中，圆锥曲线扮演重要的角色。

图 1 所示为四种航天器轨道。当航天器以**第一宇宙速度** (first cosmic velocity) 绕地运行时，运行的轨道为**正圆轨道** (circular orbit)；第一宇宙速度因此被称作**环绕速度** (orbit speed)。提高航天器绕行速度，轨道变为**椭圆轨道** (elliptical orbit)。

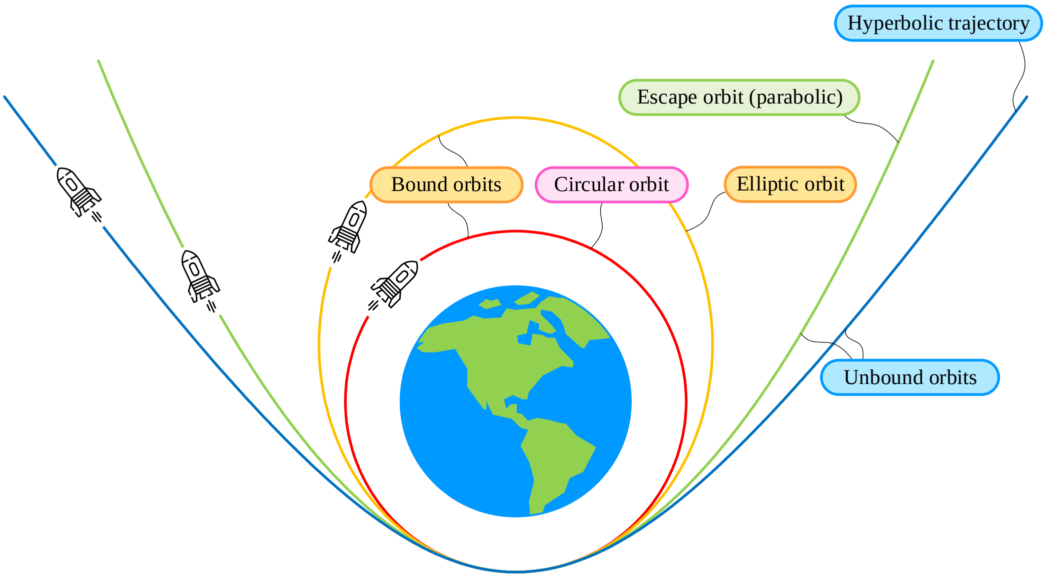


图 1. 航天器的几种轨道

继续提高绕行速度，当航天器速度达到**第二宇宙速度** (second cosmic velocity) 时，航天器便达到逃离地球所需速度；这一速度也叫**逃逸速度** (escape velocity)；这时，航天器运行轨道变为**抛物线轨道** (parabolic trajectory) 或**双曲线轨道** (hyperbolic trajectory)，它可以脱离地球的引力场而成为围绕太阳运行的人造行星。

探索火星约每 26 个月有一个发射窗口；这是因为地球在低轨道绕太阳运行，而火星在高轨道绕行。地球和火星的公转周期不同，两个行星大约每 26 个月"相遇"一次，也就是说地球与火星之间的距离最近。

如图 2 所示，探索火星需要利用**霍曼转移轨道** (Hohmann transfer orbit)；简单来说，霍曼轨道是一条椭圆形的轨道，通过两次加速将航天器从地球所在的低轨道送入火星运动的高轨道。

航天器首先进入绕太阳圆周运动的低轨道。

太空船在低轨道 A 点处上瞬间加速后，进入一个椭圆形的转移轨道；注意，加速瞬间火星位于 B。太空船由此椭圆轨道的近拱点开始，抵达远拱点后再瞬间加速，进入火星所在的目标轨道。反过来，霍曼转移轨道亦可将太空船送往较低的轨道，不过是两次减速而非加速。

拱点 (apsis) 在天文学中是指椭圆轨道上运行天体最接近或最远离它的引力中心，比如太阳，的点。最靠近引力中心的点称为近拱点 (periapsis)；而距离引力中心最远的点就称为远拱点 (apoapsis)。

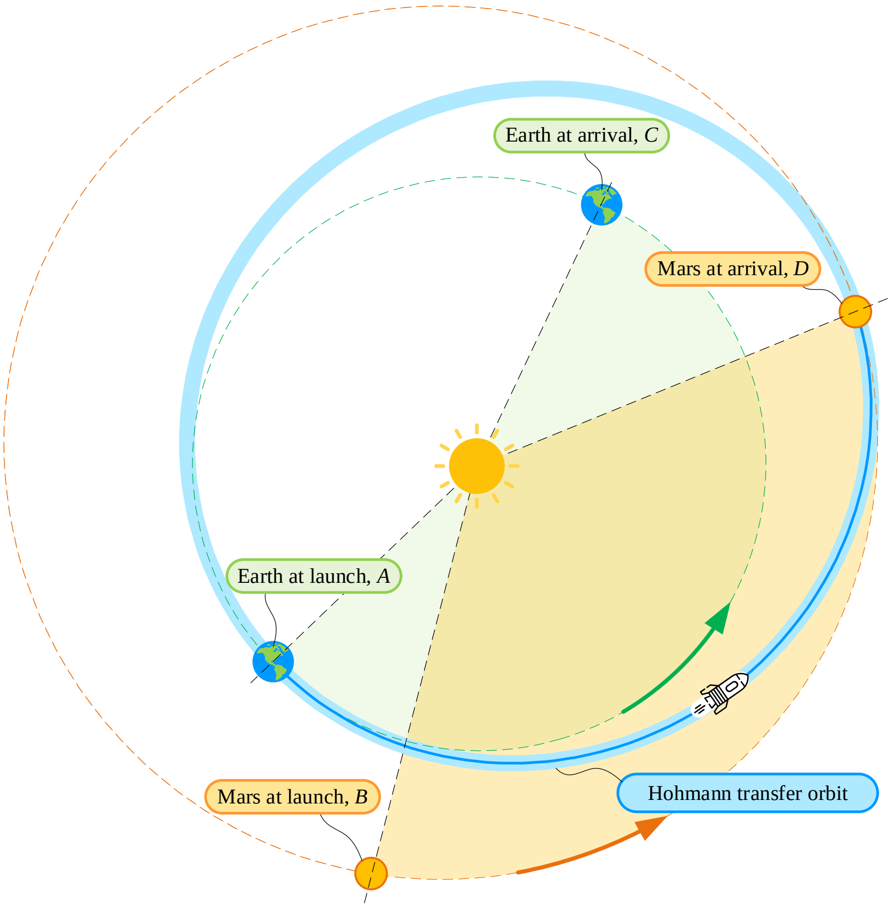


图 2. 探索火星的霍曼轨道

9.2 离心率：联系不同类型圆锥曲线

不同类型圆锥曲线可以通过同**离心率** (eccentricity) e 联系起来。

$$x_2^2 = 2px_1 + \left(e^2 - 1\right)x_1^2, \quad e \ge 0 \tag{1}$$

正圆的离心率 $e = 0$，椭圆的离心率 $0 < e < 1$，抛物线离心率 $e = 1$，双曲线离心率 $e > 1$。(1) 对应的这一组曲线共用 (0, 0) 这个顶点。

当 $p = 1$ 时，离心率 e 取不同数值，可以得到如图 3 所示一组圆锥曲线。

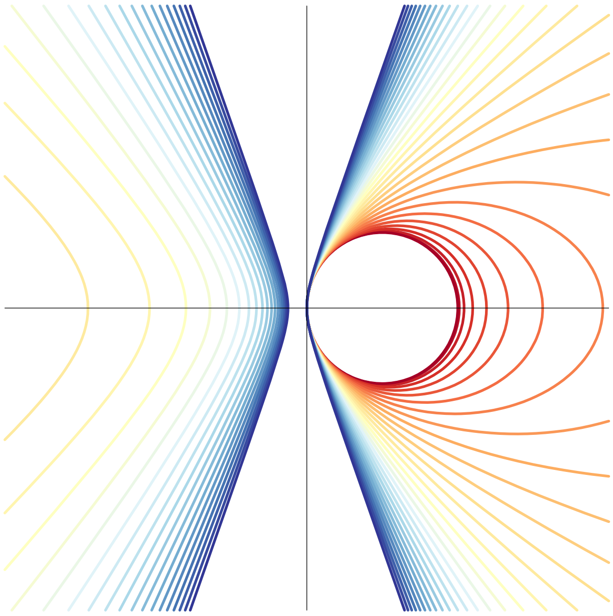


图 3. 离心率连续变化条件下一组圆锥曲线

以下代码绘制图 3；代码采用等高线方式可视化圆锥曲线。

```
# Bk_Ch9_01

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

x = np.linspace(-4,4,num = 201)
y = np.linspace(-4,4,num = 201)
m = 1
n = 1.5

xx,yy = np.meshgrid(x,y);

e_array = np.linspace(0,3,num = 31)

fig, ax = plt.subplots(figsize=(8, 8))

colors = plt.cm.RdYlBu(np.linspace(0,1,len(e_array)))

for i in range(0,len(e_array)):

    e = e_array[i]

    ellipse = yy**2 - (e**2 - 1)*xx**2 - 2*xx;

    color_code = colors[i,:].tolist()

    plt.contour(xx,yy,ellipse,levels = [0], colors = [color_code])
```

```
plt.axvline(x = 0, color = 'k', linestyle = '-')
plt.axhline(y = 0, color = 'k', linestyle = '-')
ax.set_xticks([])
ax.set_yticks([])
ax.set_xlim([-4,4])
ax.set_ylim([-4,4])

ax.spines['top'].set_visible(False)
ax.spines['right'].set_visible(False)
ax.spines['bottom'].set_visible(False)
ax.spines['left'].set_visible(False)
```

9.3 一组有趣的圆锥曲线

本节介绍一组有趣的圆锥曲线，解析式如下。

$$\frac{x_1^2}{m^2}+\frac{x_2^2}{n^2}-2\rho\frac{x_1x_2}{mn}=1 \tag{2}$$

上式可以看做是椭圆和双曲线的“叠加”；$xy = 1$ 实际上是一个旋转双曲线。参数 ρ 可以视作调节双曲线的“影响力”。

点 $(\pm m, 0)$、$(0, \pm n)$ 都满足 (2)，也就是说这四个点都在圆锥曲线上。

如图 4 所示，当 $m = n = 1$ 时且 ρ 非负，圆锥曲线随 ρ 变化。如图 5 所示，当 $m = n = 1$ 时且 ρ 非负，圆锥曲线随 ρ 变化。

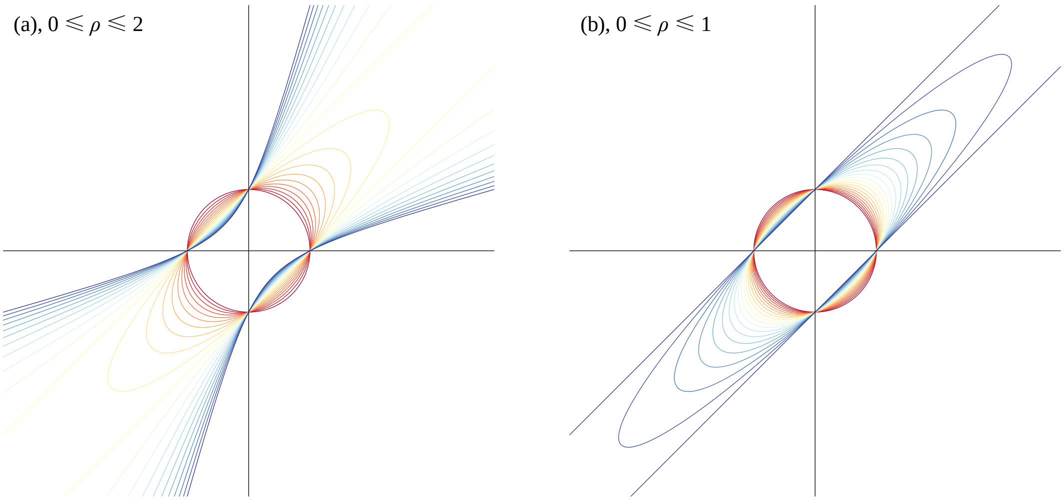


图 4. $m = n = 1$，圆锥曲线随 ρ 变化，ρ 非负

特别地，当 $\rho = 1$ 时，

$$(x-y)^2=1 \tag{3}$$

解析式对应两个直线。

$$x-y=1,\quad x-y=-1 \tag{4}$$

当 $\rho=-1$ 时，解析式也对应两条直线。

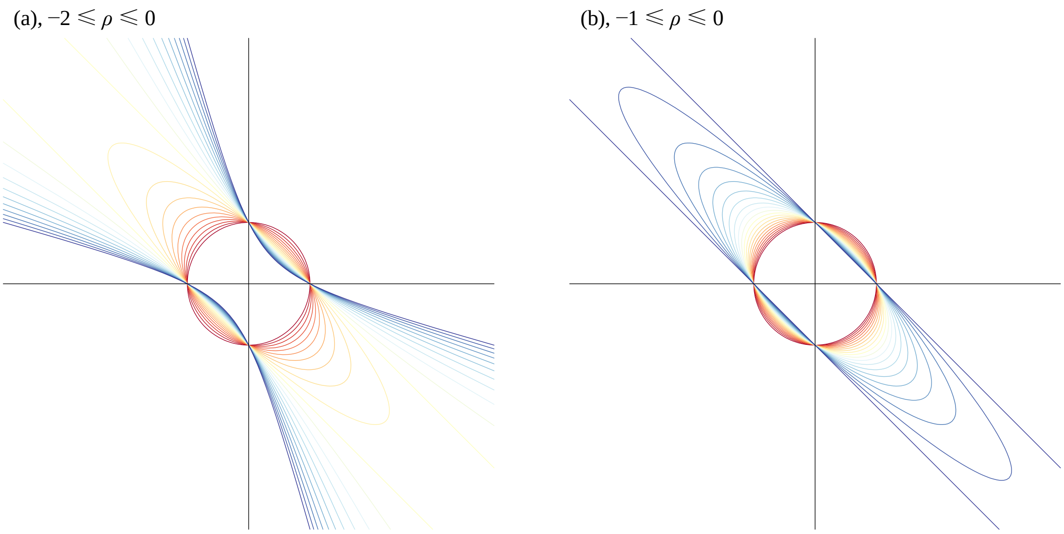


图 5. $m=n=1$，圆锥曲线随 ρ 变化，ρ 非正

图 6 所示为 $m=2$，$n=1$，圆锥曲线随 ρ 变化，ρ 的变化范围为 $[-2, 2]$。

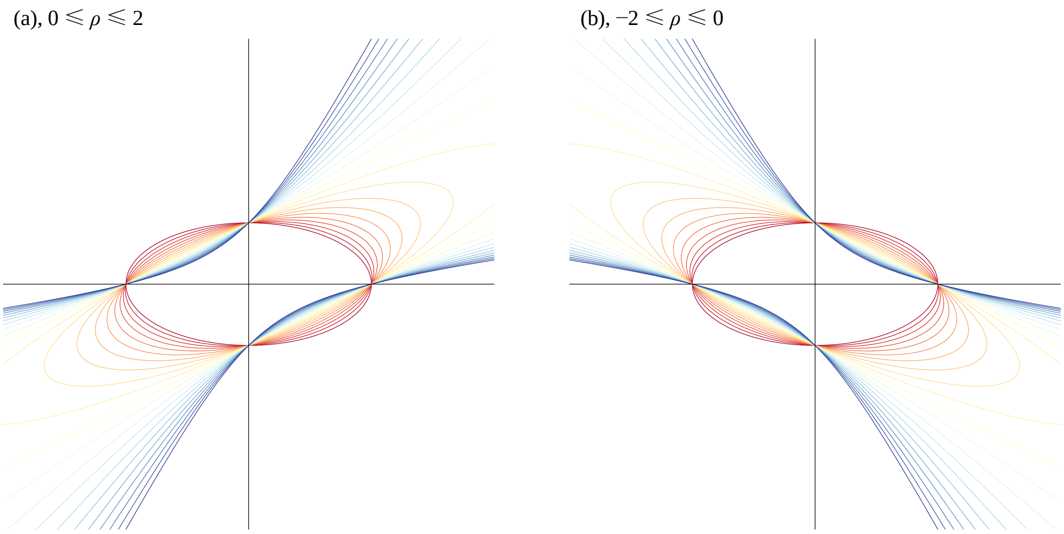


图 6. $m=2$，$n=1$，圆锥曲线随 ρ 变化，ρ 的变化范围为 $[-2, 2]$

以下代码绘制图 4、图 5 和图 6 几幅图像。

```python
# Bk_Ch9_02

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

m = 2
n = 1

x = np.linspace(-4,4,num = 201)
y = np.linspace(-4,4,num = 201)

xx,yy = np.meshgrid(x,y);

rho_array = np.linspace(0,2,num = 21)
# rho_array = np.linspace(0,1,num = 21)

rho_array = np.linspace(0,-2,num = 21)
# rho_array = np.linspace(0,-1,num = 21)

fig, ax = plt.subplots(figsize=(8, 8))

colors = plt.cm.RdYlBu(np.linspace(0,1,len(rho_array)))

for i in range(0,len(rho_array)):

    rho = rho_array[i]

    ellipse = (xx/m)**2 + (yy/n)**2 - 2*rho*(xx/m)*(yy/n);

    color_code = colors[i,:].tolist()

    plt.contour(xx,yy,ellipse,levels = [1], colors = [color_code])

plt.axvline(x = 0, color = 'k', linestyle = '-')
plt.axhline(y = 0, color = 'k', linestyle = '-')
ax.set_xticks([])
ax.set_yticks([])
ax.set_xlim([-4,4])
ax.set_ylim([-4,4])

ax.spines['top'].set_visible(False)
ax.spines['right'].set_visible(False)
ax.spines['bottom'].set_visible(False)
ax.spines['left'].set_visible(False)
```

9.4 特殊椭圆：和给定矩形相切

这一节，我们要在特殊条件约束下绘制椭圆。

给定如图 7 所示的矩形，矩形中心位于原点。绘制和矩形四个边相切的椭圆；椭圆可以是正椭圆，也可以是旋转椭圆。

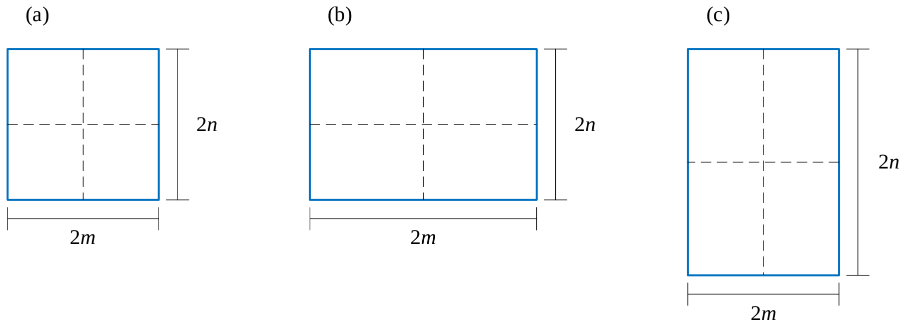


图 7. m、n 大小关系不同的矩形

对上一节 (2) 稍作修改，得到如下解析式。

$$\frac{x_1^2}{m^2}+\frac{x_2^2}{n^2}-\frac{2\rho x_1 x_2}{mn}=1-\rho^2 \tag{5}$$

参数 ρ 影响椭圆的倾斜程度，ρ 取值范围在−1 和 1 之间。

(5) 可以进一步写成。

$$\frac{1}{1-\rho^2}\left(\frac{x_1^2}{m^2}+\frac{x_2^2}{n^2}-\frac{2\rho x_1 x_2}{mn}\right)=1 \tag{6}$$

以矩形的中心为原点构造平面直角坐标系，矩形和椭圆相切的切点 A、B、C、D 的坐标为。

$$A(m,\rho n),\quad B(\rho m,n),\quad C(-m,-\rho n),\quad D(-\rho m,-n) \tag{7}$$

请大家格外注意，AC 连线，我们将在本系列丛书的条件概率和线性回归话题中谈到这条直线。

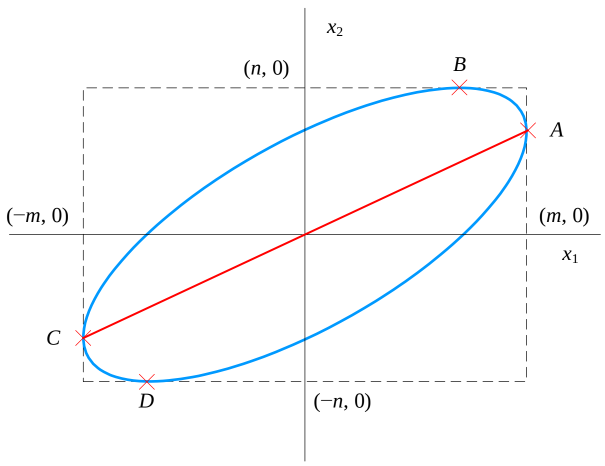


图 8. 四个切点的位置

当 $\rho = 0$ 时，椭圆为正椭圆，即。

$$\frac{x_1^2}{m^2} + \frac{x_2^2}{n^2} = 1 \tag{8}$$

如图 9 所示，椭圆和矩形相切的四个切点 A、B、C、D 的坐标为。

$$A(m,0),\quad B(0,n),\quad C(-m,0),\quad D(0,-n) \tag{9}$$

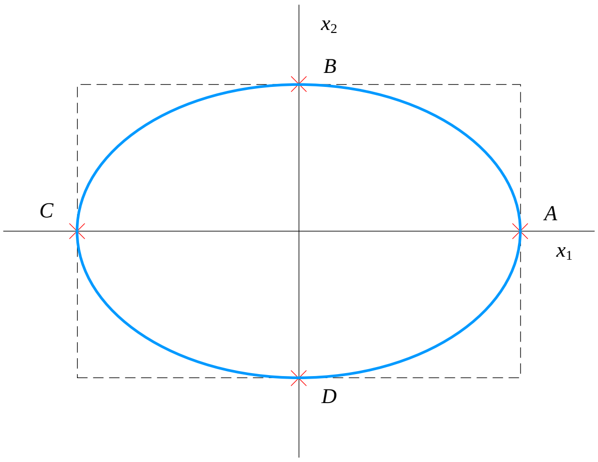


图 9. 当 $\rho = 0$ 时，四个切点的位置

当 $\rho = 1$ 时，椭圆蜕化为一条线段，对应解析式为。

$$\frac{x_1}{m} - \frac{x_2}{n} = 0 \tag{10}$$

当 $\rho = -1$ 时，椭圆也是一条线段。

$$\frac{x_1}{m} + \frac{x_2}{n} = 0 \tag{11}$$

两种情况对应的图像为图 10。

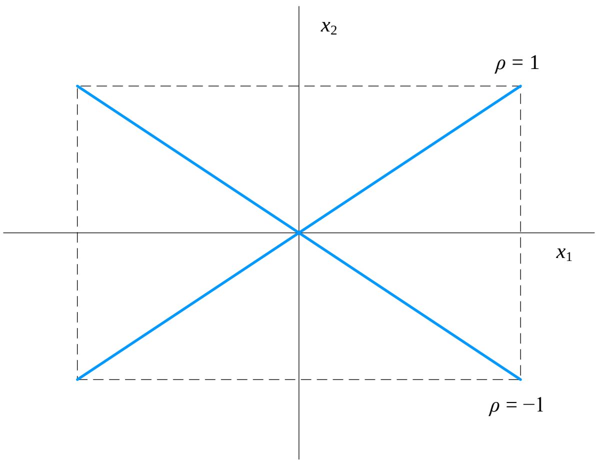


图 10. 当 $\rho = \pm 1$ 时，椭圆蜕化成线段

图 11 所示为，当 $m = n$，ρ 取值不同时和给定正方形相切椭圆；当 ρ 靠近 0 时，椭圆形状越接近正圆。此外，请大家注意切点位置移动。

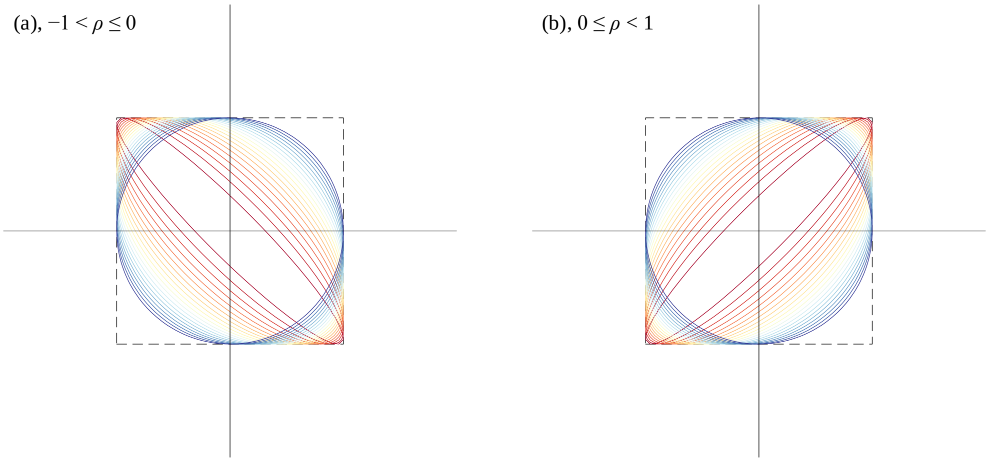


图 11. $m = n$ 时，和给定正方形相切椭圆

图 12 和图 13 所示为，$m > n$ 和 $m < n$ 两种情况条件下，椭圆形状随 ρ 变化。

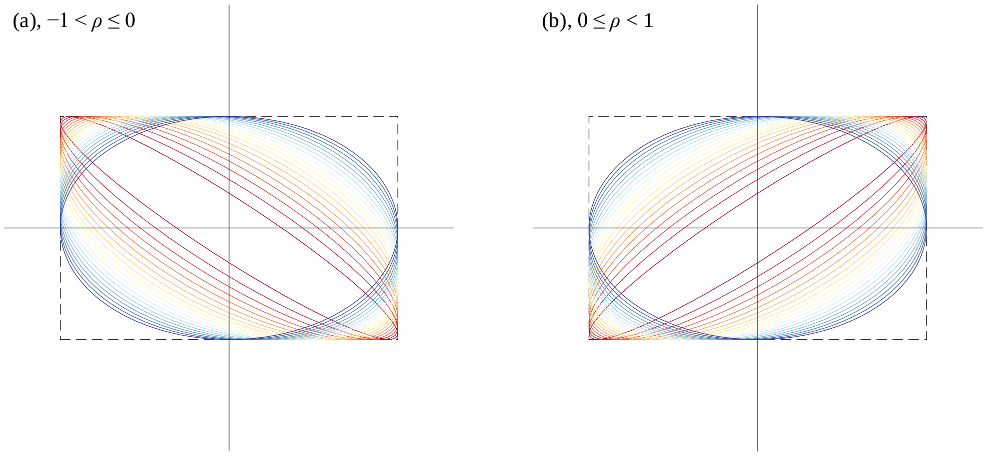


图 12. $m > n$ 时，和给定矩形相切椭圆

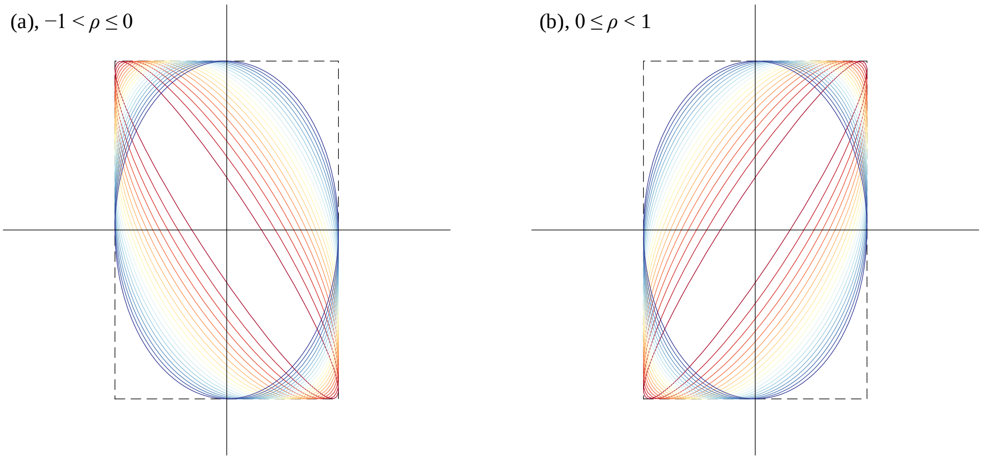


图 13. $m < n$ 时，和给定矩形相切椭圆

我们之所以讨论这种特殊形态的椭圆，是因为它和二元高斯分布的概率密度函数直接相关。二元高斯分布的概率密度函数 $f_{X,Y}(x,y)$ 解析式如下。

$$f_{X,Y}(x,y) = \frac{1}{2\pi\sigma_X\sigma_Y\sqrt{1-\rho_{X,Y}^2}} \times \exp\left(\frac{-1}{2}\frac{1}{\left(1-\rho_{X,Y}^2\right)}\underbrace{\left(\left(\frac{x-\mu_X}{\sigma_X}\right)^2 - 2\rho_{X,Y}\left(\frac{x-\mu_X}{\sigma_X}\right)\left(\frac{y-\mu_Y}{\sigma_Y}\right)+\left(\frac{y-\mu_Y}{\sigma_Y}\right)^2\right)}_{\text{Ellipse}}\right) \quad (12)$$

其中，μ_X 和 μ_Y 分别为随机变量 X、Y 的期望值；σ_X 和 σ_Y 分别为随机变量 X、Y 的均方差；$\rho_{X,Y}$ 为 X 和 Y 线性相关系数，取值区间为 $(-1, 1)$。

相信大家已经在 (12) 看到了 (6)。

以下代码绘制图 11、图 12、图 13。

```
# Bk_Ch9_03

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

x = np.linspace(-4,4,num = 201)
y = np.linspace(-4,4,num = 201)
m = 1
n = 1.5

xx,yy = np.meshgrid(x,y);

rho_array = np.linspace(-0.95,0,num = 20)

fig, ax = plt.subplots(figsize=(8, 8))

# Create a Rectangle patch
rect = patches.Rectangle((-m, -n), 2*m, 2*n,
                         linewidth = 0.25, edgecolor='k',
                         linestyle = '--',
                         facecolor = 'none')
```

```python
# Add the patch to the Axes
ax.add_patch(rect)

colors = plt.cm.RdYlBu(np.linspace(0,1,len(rho_array)))

for i in range(0,len(rho_array)):

    rho = rho_array[i]

    ellipse = ((xx/m)**2 - 2*rho*(xx/m)*(yy/n) + (yy/n)**2)/(1 - rho**2);

    color_code = colors[i,:].tolist()

    plt.contour(xx,yy,ellipse,levels = [1], colors = [color_code])

plt.axvline(x = 0, color = 'k', linestyle = '-')
plt.axhline(y = 0, color = 'k', linestyle = '-')
ax.set_xticks([])
ax.set_yticks([])
ax.set_xlim([-2,2])
ax.set_ylim([-2,2])

ax.spines['top'].set_visible(False)
ax.spines['right'].set_visible(False)
ax.spines['bottom'].set_visible(False)
ax.spines['left'].set_visible(False)
```

9.5 超椭圆：和范数有关

超椭圆 (superellipse) 是对椭圆的拓展，最常见的超椭圆的解析式为。

$$\left|\frac{x_1}{a}\right|^p + \left|\frac{x_2}{b}\right|^p = 1 \tag{13}$$

一般情况，p 为大于 0 的数值。

特别地，当 $p = 2$，(13) 所示为椭圆解析式。

还有两个特殊的例子，当 $p = 1$ 时，超椭圆图形为菱形。

$$\left|\frac{x_1}{a}\right| + \left|\frac{x_2}{b}\right| = 1 \tag{14}$$

当 $p = +\infty$时，超椭圆图形为长方形。

举个例子，当 $a = 2$，$b = 1$ 时，超椭圆的解析式为。

$$\left|\frac{x_1}{2}\right|^p + \left|\frac{x_2}{1}\right|^p = 1 \tag{15}$$

图 14 所示为 p 取不同值时，超椭圆的形状。

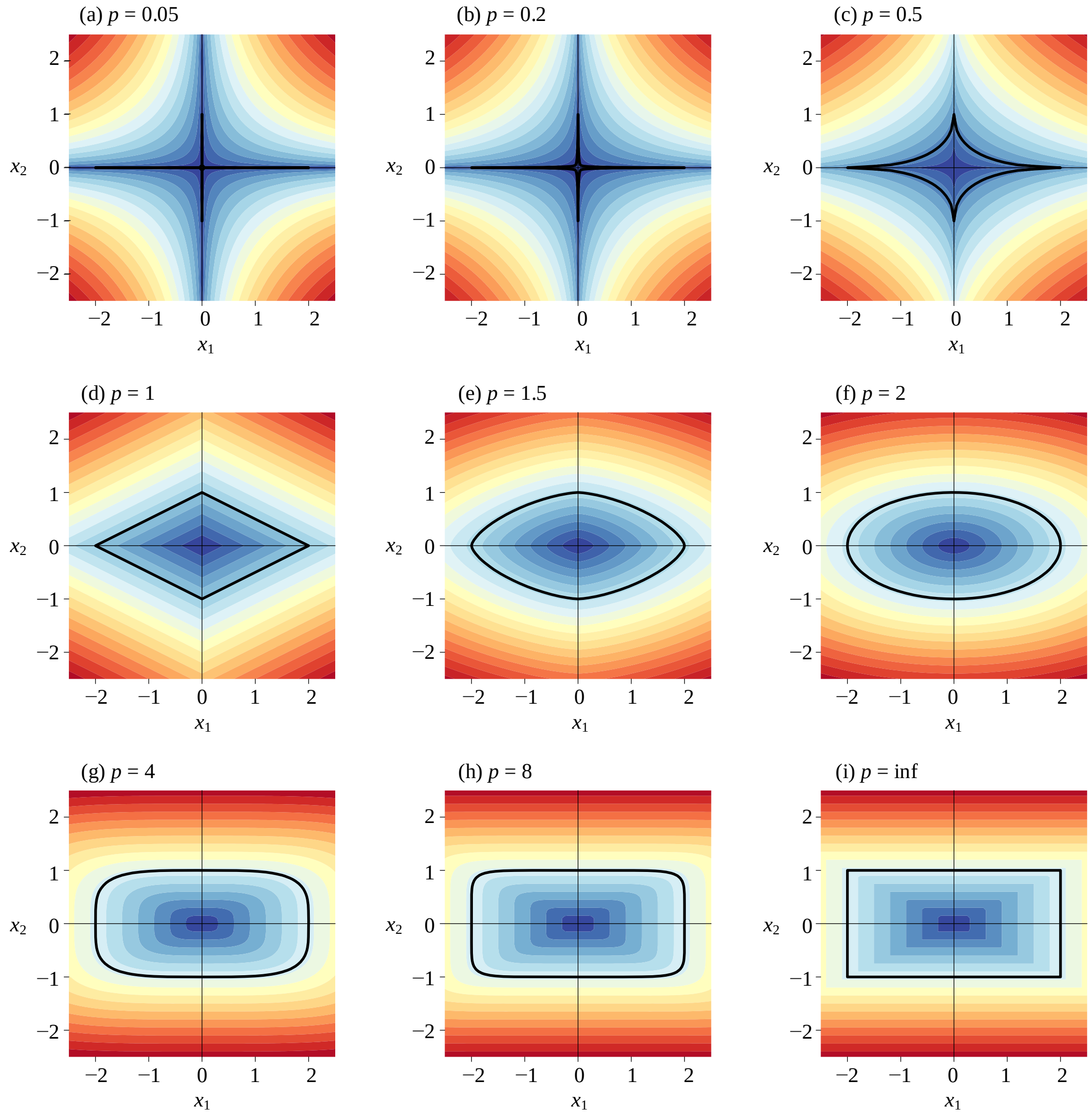


图 14. 超椭圆 p 取不同值时，超椭圆的形状，$a = 2$，$b = 1$

再举个例子，当 $a = 1$，$b = 1$ 时，超椭圆的解析式为。

$$\left|x_1\right|^p + \left|x_2\right|^p = 1 \tag{16}$$

图 15 所示为 p 取不同值时，超椭圆的形状。

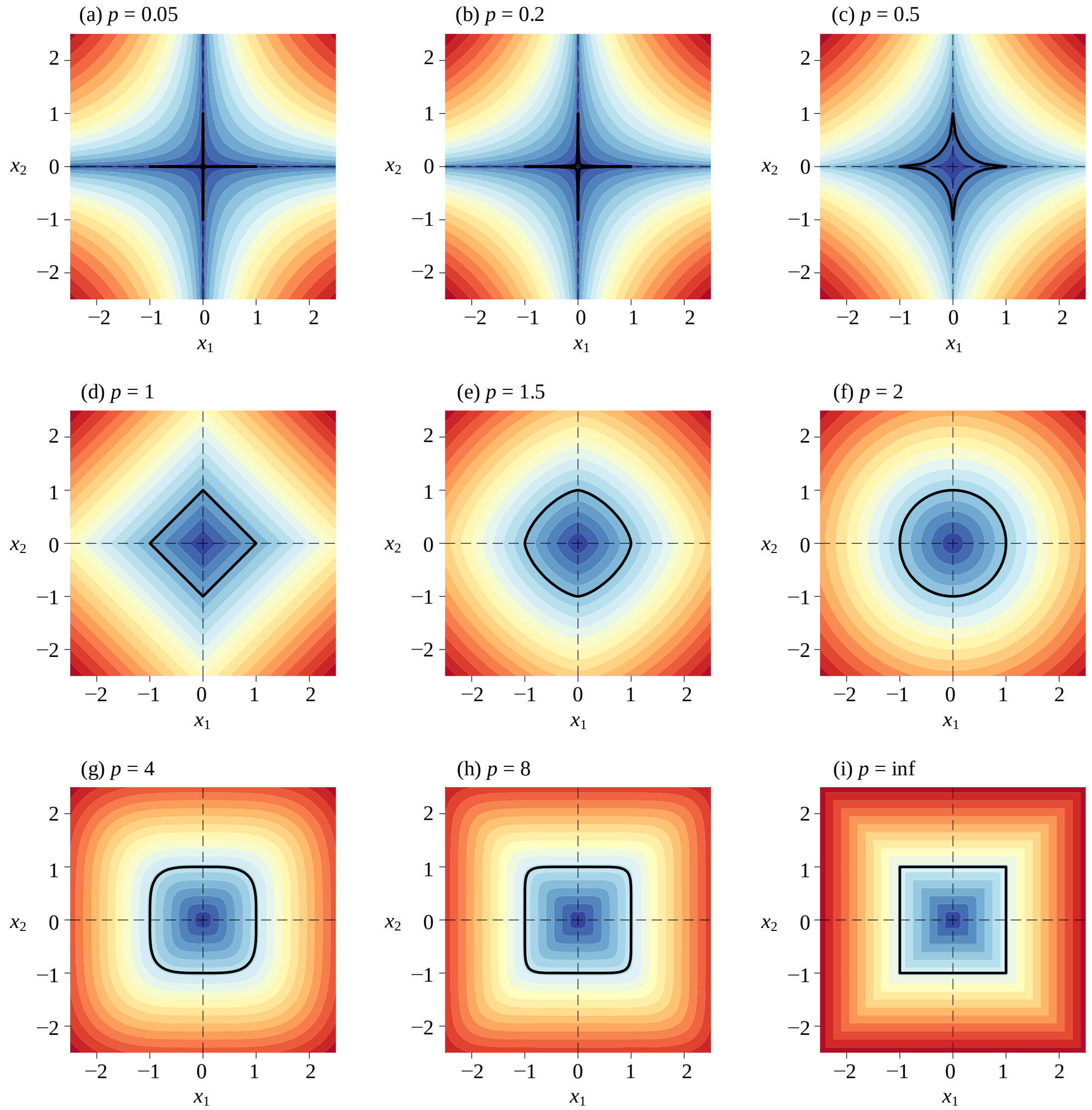


图 15. 超椭圆 p 取不同值时，超椭圆的形状，$a = 1$，$b = 1$

将 (13) 解析式进一步推广，得到如下二维平面的超椭圆解析式。

$$\left|\frac{x_1}{a}\right|^p+\left|\frac{x_2}{b}\right|^q=1 \tag{17}$$

其中，p 和 q 为正数。

举个例子，当 $a = 1$，$b = 1$ 时，(17) 对应的超椭圆的解析式为。

$$|x_1|^p+|x_2|^q=1 \tag{18}$$

图 16 所示为 p 和 q 取不同值时，(18) 对应超椭圆的形状。

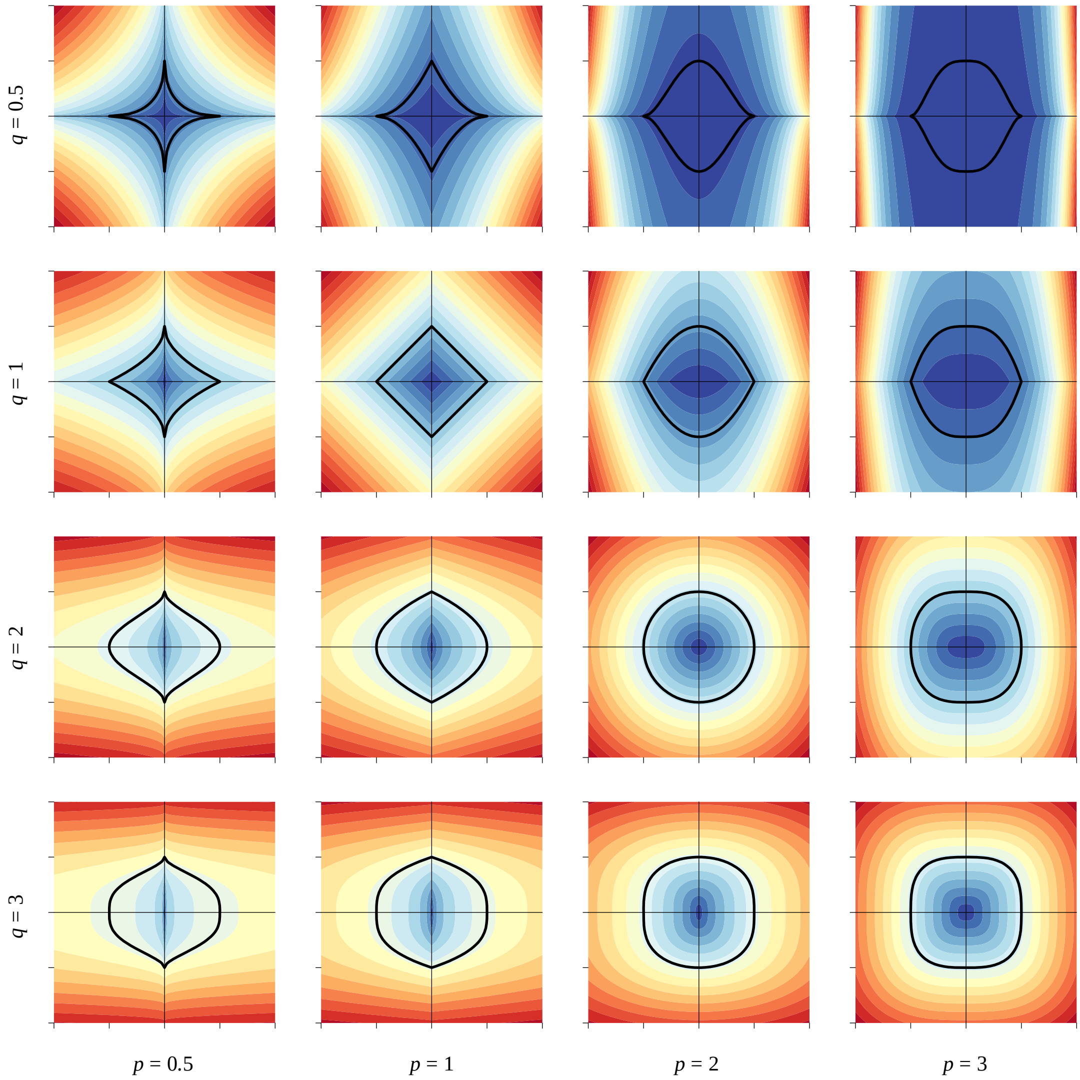


图 16. p 和 q 取不同值时，超椭圆的形状，$a = 1$，$b = 1$

从二维到三维，可以得到**超椭球** (superellipsoid) 的解析式。

$$\left(\left|\frac{x_1}{a}\right|^r + \left|\frac{x_2}{b}\right|^r\right)^{\frac{t}{r}} + \left|\frac{x_3}{c}\right|^t = 1 \tag{19}$$

图 17 所示为 $a = 1$ 和 $b = 1$，t 和 r 取不同值时，超椭球的形状。

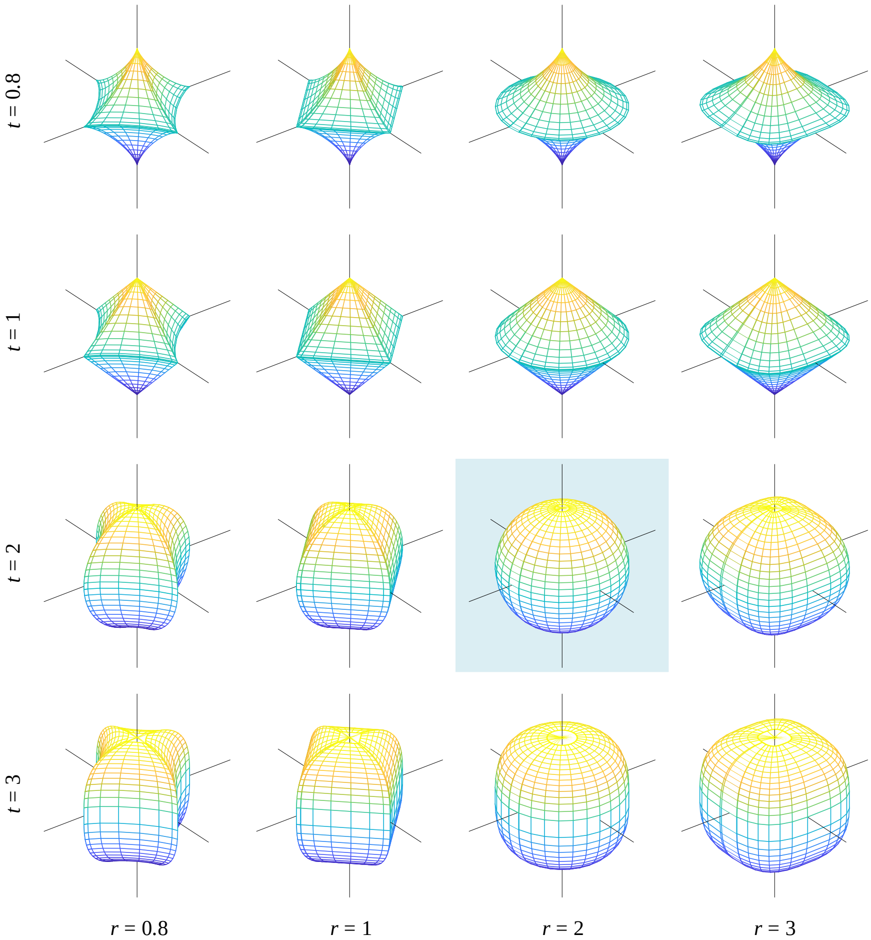


图 17. t 和 r 取不同值时，超椭球的形状，a = 1，b = 1

本节介绍的超椭圆和 L^p 范数紧密联系。L^p 范数的定义如下。

$$\|\boldsymbol{x}\|_p = \left(|x_1|^p + |x_2|^p + \cdots + |x_D|^p\right)^{1/p} = \left(\sum_{i=1}^{D}|x_i|^p\right)^{1/p} \tag{20}$$

其中，

$$\boldsymbol{x} = \begin{bmatrix} x_1 & x_2 & \cdots & x_D \end{bmatrix}^{\mathrm{T}} \tag{21}$$

图 18 所示为随着 p 增大，L^p 范数等距线一层层包裹。在数据科学和机器学习中，L^p 范数常用来度量距离。当 $p = 2$，就是 L^2 范数，这便是前文介绍的欧氏距离。

本系列丛书将在《矩阵力量》一册系统讲解范数。

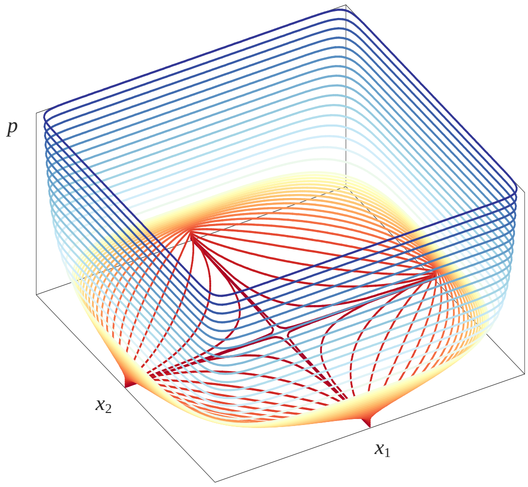


图 18. 随着 p 增大，L^p 范数等距线一层层包裹

以下代码绘制图 14、图 15、图 16。

```
# Bk_Ch9_04

import matplotlib.pyplot as plt
import numpy as np

a = 1;
b = 1;

p = [0.5,1,2,3]
q = p

pp,qq = np.meshgrid(p,q)
pp = pp.flatten()
qq = qq.flatten()

x1 = np.linspace(-2, 2, num=101);
x2 = x1;

xx1, xx2 = np.meshgrid(x1,x2)

fig, axes = plt.subplots(ncols=4,nrows=4,
                         figsize=(12, 12))

for p, q, ax in zip(pp, qq, axes.flat):

    if np.isinf(p):
        zz = np.maximum(np.abs(xx1/a),np.abs(xx2/b))
    else:
        zz = ((np.abs((xx1/a))**p) + (np.abs((xx2/b))**q))**(1./q)

    # plot contour of Lp
    ax.contourf(xx1, xx2, zz, 20, cmap='RdYlBu_r')

    # plot contour of Lp = 1
    ax.contour (xx1, xx2, zz, [1], colors='k', linewidths = 2)

    # decorations
```

```
    ax.axhline(y=0, color='k', linewidth = 0.25)
    ax.axvline(x=0, color='k', linewidth = 0.25)
    ax.set_xlim(-2, 2)
    ax.set_ylim(-2, 2)
    ax.spines['top'].set_visible(False)
    ax.spines['right'].set_visible(False)
    ax.spines['bottom'].set_visible(False)
    ax.spines['left'].set_visible(False)
    ax.set_xlabel('$x_1$')
    ax.set_ylabel('$x_2$')
    ax.set_title('p = ' + str(p) + 'q = ' + str(q))
    ax.set_aspect('equal', adjustable='box')

plt.show()
```

9.6 双曲函数：基于单位双曲线

当 $a = 1$ 和 $b = 1$ 时，双曲线为**单位双曲线** (unit hyperbola)。

$$x_1^2 - x_2^2 = 1 \qquad a,b > 0 \tag{22}$$

单位双曲线可以用来定义**双曲函数** (hyperbolic function)。

如图 19 所示，最基本的双曲函数是双曲正弦函数 sinh 和双曲余弦函数 cosh。

双曲正切，可以通过如下比例计算得到。

$$\tanh\theta = \frac{\sinh\theta}{\cosh\theta} \tag{23}$$

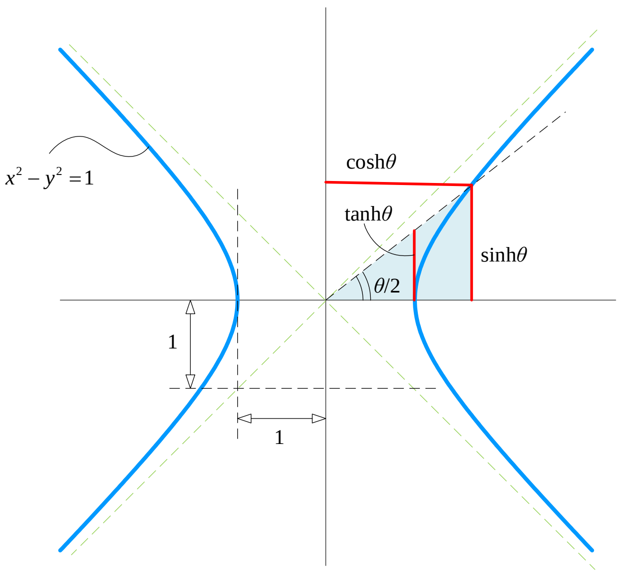


图 19. 单位双曲线和双曲函数的关系

图 20 所示为 $\sinh\theta$、$\cosh\theta$ 和 $\tanh\theta$ 三个函数的关系。

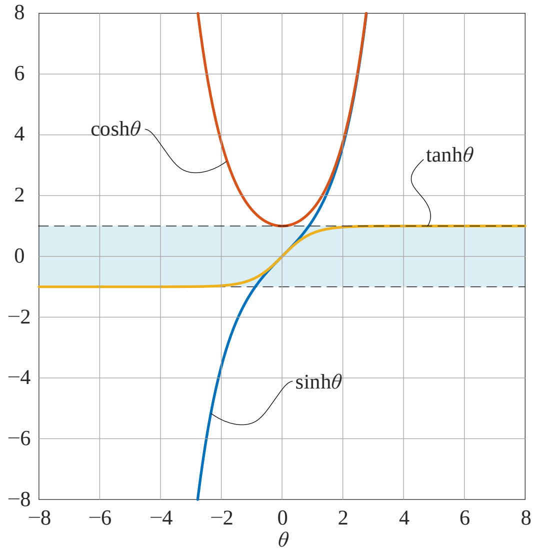


图 20. $\sinh\theta$、$\cosh\theta$ 和 $\tanh\theta$ 三者关系

表 1. 用英文表达双曲函数

数学表达	英文表达	中文表达
$\sinh\theta$	hyperbolic sine theta sinh /sɪntʃ/ theta	双曲正弦
$\cosh\theta$	hyperbolic co sine theta cosh /kɒʃ/ theta	双曲余弦
$\tanh\theta$	hyperbolic tangent theta tanh /tæntʃ/ theta	双曲正切

此外，$\sinh\theta$、$\cosh\theta$ 和 $\tanh\theta$ 三个函数和指数函数 $\exp(\theta)$ 存在以下关系。

$$
\begin{aligned}
\sinh\theta &= \frac{\exp(\theta)-\exp(-\theta)}{2} \\
\cosh\theta &= \frac{\exp(\theta)+\exp(-\theta)}{2} \\
\tanh\theta &= \frac{\sinh\theta}{\cosh\theta} = \frac{\exp(\theta)-\exp(-\theta)}{\exp(\theta)+\exp(-\theta)}
\end{aligned}
\tag{24}
$$

图 21 所示为 $\sinh\theta$ 和 $\cosh\theta$ 与指数函数关系。

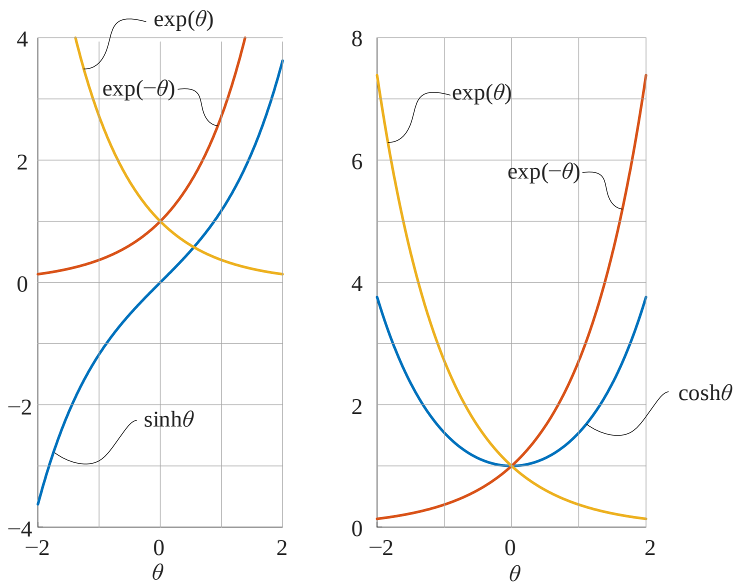


图 21. sinhθ 和 coshθ 与指数函数关系

9.7 圆锥曲线一般式

圆锥曲线的一般形式解析式如下。

$$Ax_1^2 + Bx_1x_2 + Cx_2^2 + Dx_1 + Ex_2 + F = 0 \tag{25}$$

注意当 $B = 0$ 时，圆锥曲线没有旋转。

满足下列条件，圆锥曲线为正圆。

$$Ax_1^2 + Cx_2^2 + Dx_1 + Ex_2 + F = 0, \quad A = C \tag{26}$$

满足下列条件，圆锥曲线为正椭圆，即没有旋转。

$$Ax_1^2 + Cx_2^2 + Dx_1 + Ex_2 + F = 0, \quad A \neq C, \quad AC > 0 \tag{27}$$

满足下列条件，圆锥曲线为正双曲线。

$$Ax_1^2 + Cx_2^2 + Dx_1 + Ex_2 + F = 0, \quad AC < 0 \tag{28}$$

满足下列条件，圆锥曲线为正抛物线。

$$\begin{cases} Ax_1^2 + Dx_1 + Ex_2 + F = 0 \\ Cx_2^2 + Dx_1 + Ex_2 + F = 0 \end{cases} \tag{29}$$

注意当 $B \neq 0$ 时，圆锥曲线存在旋转，需要通过 $B^2 - 4AC$ 来判断圆锥曲线类型。

$B^2-4AC<0$ 时，圆锥曲线为椭圆；$B^2-4AC=0$，圆锥曲线为抛物线；$B^2-4AC>0$ 时，圆锥曲线为双曲线。

大家可能会问，为何要采用 B^2-4AC 来判断圆锥曲线类型？我们将在《矩阵力量》回答这个问题。

把 (25) 写成如下矩阵运算式。

$$\frac{1}{2}\begin{bmatrix} x_1 \\ x_2 \end{bmatrix}^{\mathrm{T}}\begin{bmatrix} 2A & B \\ B & 2C \end{bmatrix}\begin{bmatrix} x_1 \\ x_2 \end{bmatrix}+\begin{bmatrix} D \\ E \end{bmatrix}^{\mathrm{T}}\begin{bmatrix} x_1 \\ x_2 \end{bmatrix}+F=0 \tag{30}$$

进一步写作

$$\frac{1}{2}\boldsymbol{x}^{\mathrm{T}}\boldsymbol{Q}\boldsymbol{x}+\boldsymbol{w}^{\mathrm{T}}\boldsymbol{x}+F=0 \tag{31}$$

其中，

$$\boldsymbol{Q}=\begin{bmatrix} 2A & B \\ B & 2C \end{bmatrix},\quad \boldsymbol{w}=\begin{bmatrix} D \\ E \end{bmatrix} \tag{32}$$

目前，不需要大家掌握 (30) 这个矩阵运算式；我们也将在《矩阵力量》一册深入分析这个等式。

正如牛顿所言，“我不知道世人看我的眼光。依我看来，我不过是一个在海边玩耍的孩子，不时找到几个光滑卵石、漂亮贝壳，而惊喜万分；而展现在我面前的是，真理的浩瀚海洋，静候探索。”

人类何尝不是在宇宙某个角落玩耍的一群孩子，手握的知识不过沧海一粟，却雄心万丈一心要去探索星辰大海。

但也正是这群孩子将无数的不可能变成了可能，现在他们已经在地月系、甚至太阳系的边缘跃跃欲试。

今人不见古时月，今月曾经照古人。宇宙的星辰大海一直都在人类眼前，它从未走远。路漫漫其修远兮，吾将上下而求索。

地球不过是人类的摇篮，我们的征途是星辰大海。这句话含蓄而浪漫，刘慈欣《三体》中则说的更为露骨而冷酷“我们都是阴沟里的虫子，但总还是得有人仰望星空。”