17 Derivatives of Multivariable Functions

多元函数微分

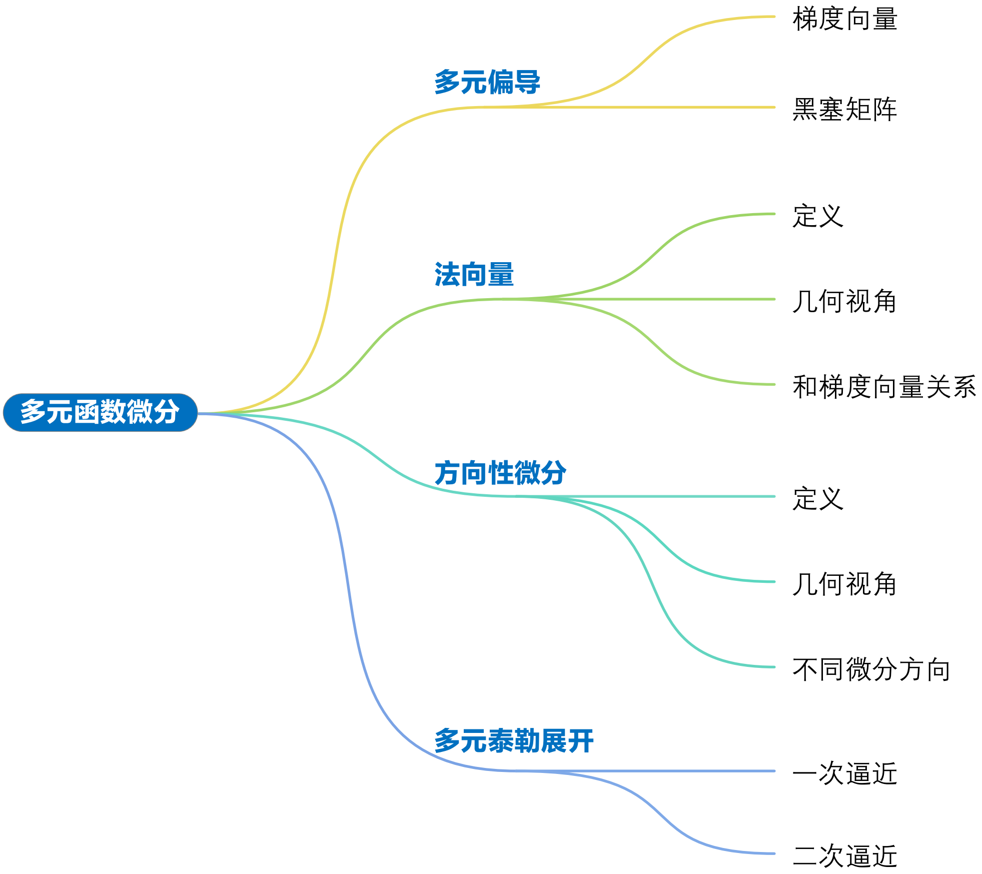
将偏微分拓展到高维和任意方向

数学的终极目标是人类精神的荣誉。

The object of mathematics is the honor of the human spirit.

—— 卡尔·雅可比 (Carl Jacobi) | 普鲁士数学家 | 1804 ~ 1851

- `numpy.meshgrid()` 获得网格数据
- `numpy.multiply()` 向量或矩阵逐项乘积
- `numpy.roots()` 多项式求根
- `numpy.sqrt()` 平方根
- `sympy.abc import x` 定义符号变量 x
- `sympy.diff()` 求解符号导数和偏导解析式
- `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


多元函数微分
多元偏导
梯度向量
黑塞矩阵
法向量
定义
几何视角
和梯度向量关系
方向性微分
定义
几何视角
不同微分方向
多元泰勒展开
一次逼近
二次逼近

17.1 多元偏导：二元到多元

回顾偏导

本系列丛书《数学要素》一册中讲解过**偏导数** (partial derivative) 内容。在数学中，一个多变量的函数的偏导数，就是它关于其中一个变量的导数而保持其他变量恒定。白话说，偏导数关注多元函数某个特定方向上的变化率。

设 $f(x_1, x_2)$ 是定义在平面 $\mathbb{R}^2$ 上的二元函数，$f(x_1, x_2)$ 在点 (a, b) 的某一邻域内有定义。图 1 (a) 网格面为 $f(x_1, x_2)$ 函数曲面，平行 x_1 轴在 $x_2 = b$ 切一刀得到浅蓝色剖面，偏导 $f_{x1}(a,b)$ 就是浅蓝色剖面在 (a, b) 一点的切线。如图 1 (b) 所示，偏导 $f_{x2}(a,b)$ 就是浅蓝色剖面在 (a, b) 一点的切线。该切线平行 x_2y 平面。

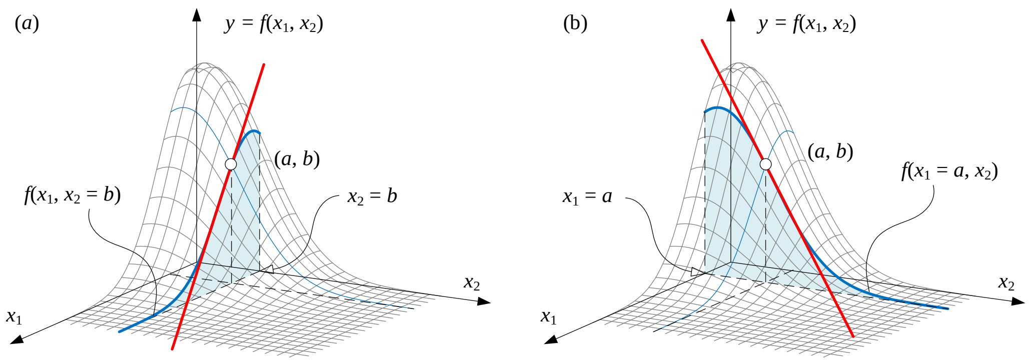


图 1. $f(x_1, x_2)$ 偏导定义

向量形式

为了方便表达和运算，我们可以把上述二元函数在 x_1 和 x_2 方向上的偏导写成列向量的形式：

$$\frac{\partial f(\boldsymbol{x})}{\partial \boldsymbol{x}} = \begin{bmatrix} \dfrac{\partial f(\boldsymbol{x})}{\partial x_1} \\ \dfrac{\partial f(\boldsymbol{x})}{\partial x_2} \end{bmatrix} \tag{1}$$

其中，$\boldsymbol{x}$ 为列向量，$\boldsymbol{x} = [x_1, x_2]^{\mathrm{T}}$。

一次函数

给定如下多元一次函数 $f(\boldsymbol{x})$：

$$f(\boldsymbol{x}) = \boldsymbol{w}^{\mathrm{T}}\boldsymbol{x} + b = \boldsymbol{x}^{\mathrm{T}}\boldsymbol{w} + b \tag{2}$$

其中，$\boldsymbol{x}$ 和 $\boldsymbol{w}$ 均为列向量：

$$\boldsymbol{w} = \begin{bmatrix} w_1 \\ w_2 \\ \vdots \\ w_D \end{bmatrix}, \quad \boldsymbol{x} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_D \end{bmatrix} \tag{3}$$

(2) 展开得到大家熟悉的一次函数形式：

$$f(\boldsymbol{x}) = w_1 x_1 + w_2 x_2 + \cdots w_D x_D + b \tag{4}$$

多元一次函数 $f(\boldsymbol{x})$ 对 $\boldsymbol{x}$ 求一阶偏导，并写成向量形式：

$$\frac{\partial f(\boldsymbol{x})}{\partial \boldsymbol{x}} = \begin{bmatrix} \frac{\partial f(\boldsymbol{x})}{\partial x_1} \\ \frac{\partial f(\boldsymbol{x})}{\partial x_2} \\ \vdots \\ \frac{\partial f(\boldsymbol{x})}{\partial x_D} \end{bmatrix} = \begin{bmatrix} w_1 \\ w_2 \\ \vdots \\ w_D \end{bmatrix} = \boldsymbol{w} \tag{5}$$

本章后文会给上式一个新的名字——梯度向量。另外，请大家注意以下等价：

$$\frac{\partial(\boldsymbol{w}^{\mathrm{T}}\boldsymbol{x})}{\partial \boldsymbol{x}} = \frac{\partial(\boldsymbol{x}^{\mathrm{T}}\boldsymbol{w})}{\partial \boldsymbol{x}} = \frac{\partial(\boldsymbol{w}\cdot\boldsymbol{x})}{\partial \boldsymbol{x}} = \frac{\partial(\boldsymbol{x}\cdot\boldsymbol{w})}{\partial \boldsymbol{x}} = \frac{\partial\langle \boldsymbol{w},\boldsymbol{x}\rangle}{\partial \boldsymbol{x}} = \boldsymbol{w} \tag{6}$$

二次函数

给定如下二次函数：

$$f(\boldsymbol{x}) = \boldsymbol{x}^{\mathrm{T}}\boldsymbol{x} = x_1^2 + x_2^2 + \cdots x_D^2 \tag{7}$$

对 (7) 求一阶偏导：

$$\frac{\partial f(\boldsymbol{x})}{\partial \boldsymbol{x}} = \begin{bmatrix} \frac{\partial f(\boldsymbol{x})}{\partial x_1} \\ \frac{\partial f(\boldsymbol{x})}{\partial x_2} \\ \vdots \\ \frac{\partial f(\boldsymbol{x})}{\partial x_D} \end{bmatrix} = \begin{bmatrix} 2x_1 \\ 2x_2 \\ \vdots \\ 2x_D \end{bmatrix} = 2\boldsymbol{x} \tag{8}$$

要是类比的话，$f(\boldsymbol{x}) = \boldsymbol{x}^{\mathrm{T}}\boldsymbol{x}$ 相当于 $f(x) = x^2$。而上式相当于 $f(x)$ 的一阶导数 $f'(x) = 2x$。

(8) 等价于：

$$\frac{\partial\left(\boldsymbol{x}^{\mathrm{T}}\boldsymbol{x}\right)}{\partial\boldsymbol{x}}=\frac{\partial\left(\boldsymbol{x}\cdot\boldsymbol{x}\right)}{\partial\boldsymbol{x}}=\frac{\partial\left\langle\boldsymbol{x},\boldsymbol{x}\right\rangle}{\partial\boldsymbol{x}}=\frac{\partial\left(\left\|\boldsymbol{x}\right\|_2^2\right)}{\partial\boldsymbol{x}}=2\boldsymbol{x} \tag{9}$$

上式中这些运算都相当于 x^2。

二次型

给定如下多元二次型：

$$f\left(\boldsymbol{x}\right)=\boldsymbol{x}^{\mathrm{T}}\boldsymbol{Q}\boldsymbol{x} \tag{10}$$

求 (10) 的一阶偏导：

$$\frac{\partial\left(\boldsymbol{x}^{\mathrm{T}}\boldsymbol{Q}\boldsymbol{x}\right)}{\partial\boldsymbol{x}}=\left(\boldsymbol{Q}+\boldsymbol{Q}^{\mathrm{T}}\right)\boldsymbol{x} \tag{11}$$

注意，$\boldsymbol{Q}$ 为常数矩阵。

如果 $\boldsymbol{Q}$ 为对称矩阵，(10) 的一阶导数可以写成：

$$\frac{\partial\left(\boldsymbol{x}^{\mathrm{T}}\boldsymbol{Q}\boldsymbol{x}\right)}{\partial\boldsymbol{x}}=2\boldsymbol{Q}\boldsymbol{x} \tag{12}$$

假设 $\boldsymbol{Q}$ 为对称矩阵，给定如下二次函数：

$$f\left(\boldsymbol{x}\right)=\frac{1}{2}\boldsymbol{x}^{\mathrm{T}}\boldsymbol{Q}\boldsymbol{x}+\boldsymbol{w}^{\mathrm{T}}\boldsymbol{x}+b \tag{13}$$

对 (13) 求一阶偏导：

$$\frac{\partial f\left(\boldsymbol{x}\right)}{\partial\boldsymbol{x}}=\boldsymbol{Q}\boldsymbol{x}+\boldsymbol{w} \tag{14}$$

举个形似 (13) 的例子：

$$f\left(\boldsymbol{x}\right)=\frac{1}{2}\boldsymbol{x}^{\mathrm{T}}\underbrace{\begin{bmatrix}1 & 2\\ 2 & 3\end{bmatrix}}_{\boldsymbol{Q}}\boldsymbol{x}+\underbrace{\begin{bmatrix}4\\ 5\end{bmatrix}}_{\boldsymbol{w}}^{\mathrm{T}}\boldsymbol{x}+6 \tag{15}$$

对 (15) 求一阶偏导：

$$\frac{\partial f\left(\boldsymbol{x}\right)}{\partial\boldsymbol{x}}=\begin{bmatrix}1 & 2\\ 2 & 3\end{bmatrix}\boldsymbol{x}+\begin{bmatrix}4\\ 5\end{bmatrix} \tag{16}$$

对如下形式函数求一阶偏导：

$$\frac{\partial\left(\left(\boldsymbol{x}-\boldsymbol{c}\right)^{\mathrm{T}}\boldsymbol{Q}\left(\boldsymbol{x}-\boldsymbol{c}\right)\right)}{\partial\boldsymbol{x}}=2\boldsymbol{Q}\left(\boldsymbol{x}-\boldsymbol{c}\right) \tag{17}$$

其中，$\boldsymbol{Q}$ 为对称矩阵。

二阶偏导：黑塞矩阵

黑塞矩阵矩阵 (Hessian matrix) 是一个多元函数的二阶偏导数构成的方阵，描述了函数的局部曲率。黑塞矩阵由德国数学家奥托·黑塞 (Otto Hesse) 引入并以其命名。

假设有一实值函数 $f(\boldsymbol{x})$，如果它的所有二阶偏导数都存在并在定义域内连续，那么 $f(\boldsymbol{x})$ 的黑塞矩阵 $\boldsymbol{H}$ 为：

$$\boldsymbol{H}=\begin{bmatrix} \frac{\partial^2 f}{\partial x_1^2} & \frac{\partial^2 f}{\partial x_1 \partial x_2} & \cdots & \frac{\partial^2 f}{\partial x_1 \partial x_D} \\ \frac{\partial^2 f}{\partial x_2 \partial x_1} & \frac{\partial^2 f}{\partial x_2^2} & \cdots & \frac{\partial^2 f}{\partial x_2 \partial x_D} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial^2 f}{\partial x_D \partial x_1} & \frac{\partial^2 f}{\partial x_D \partial x_2} & \cdots & \frac{\partial^2 f}{\partial x_D^2} \end{bmatrix} \tag{18}$$

(10) 中给定二次函数求二阶偏导，获得黑塞矩阵：

$$\boldsymbol{H}=\frac{\partial\left(\boldsymbol{x}^{\mathrm{T}}\boldsymbol{Q}\boldsymbol{x}\right)}{\partial\boldsymbol{x}\partial\boldsymbol{x}^{\mathrm{T}}}=\boldsymbol{Q}+\boldsymbol{Q}^{\mathrm{T}} \tag{19}$$

如果 $\boldsymbol{Q}$ 为对称，(19) 中黑塞矩阵为：

$$\boldsymbol{H}=\frac{\partial\left(\boldsymbol{x}^{\mathrm{T}}\boldsymbol{Q}\boldsymbol{x}\right)}{\partial\boldsymbol{x}\partial\boldsymbol{x}^{\mathrm{T}}}=2\boldsymbol{Q} \tag{20}$$

以 (15) 为例，多元函数的黑塞矩阵为：

$$\boldsymbol{H}=\begin{bmatrix} \frac{\partial^2 f}{\partial x_1^2} & \underset{x_1\to x_2}{\frac{\partial^2 f}{\partial x_1 \partial x_2}} \\ \underset{x_2\to x_1}{\frac{\partial^2 f}{\partial x_2 \partial x_1}} & \frac{\partial^2 f}{\partial x_2^2} \end{bmatrix}=\begin{bmatrix} 1 & 2 \\ 2 & 3 \end{bmatrix} \tag{21}$$

本书后续会在优化问题中用到黑塞矩阵判断极值点。

本节的内容可能会显得单调。本章后续将依托几何视角帮助读者理解本节内容。

17.2 梯度向量：上山方向

我们给上节讨论的一阶偏导新名字——梯度向量 (gradient vector)。函数 $f(\boldsymbol{x})$ 的梯度向量定义如下：

$$\text{grad}\, f\left(\boldsymbol{x}\right)=\nabla f\left(\boldsymbol{x}\right)=\begin{bmatrix}\dfrac{\partial f}{\partial x_1}\\ \dfrac{\partial f}{\partial x_2}\\ \vdots \\ \dfrac{\partial f}{\partial x_D}\end{bmatrix} \tag{22}$$

梯度向量可以使用 grad() 作为运算符，也使用倒三角微分算子 ∇， ∇ 也叫 Nabla 算子 (Nabla symbol)。

几何视角

从几何视角来看梯度向量，如图 2 所示，在坡面 P 点处放置一个小球，轻轻松开手一瞬间，小球沿着坡面最陡峭方向滚下。瞬间滚动方向在平面上的投影便是梯度下降方向 (direction of gradient descent)，也称“下山”方向。

数学中，此方向的反方向即梯度方向，也称作梯度上升方向。这个“上山”方向就是 P 点梯度向量的方向。

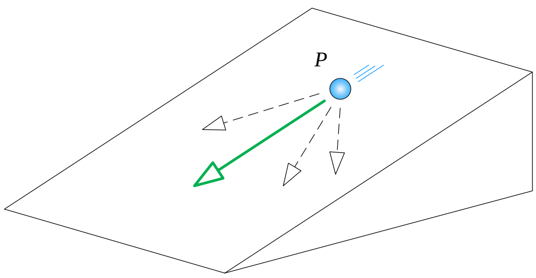


图 2. 梯度方向原理

二元函数

以二元函数为例，$f(x_1, x_2)$ 某一点 P 处梯度向量为：

$$\nabla f\left(\boldsymbol{x}_P\right)=\text{grad}\, f\left(\boldsymbol{x}_P\right)=\begin{bmatrix}\dfrac{\partial f\left(\boldsymbol{x}\right)}{\partial x_1}\\ \dfrac{\partial f\left(\boldsymbol{x}\right)}{\partial x_2}\end{bmatrix}_{\boldsymbol{x}_P} \tag{23}$$

P 处于不同点时，可以得到梯度向量场 (gradient vector field)。图 3 所示为某个函数梯度向量的分布。大家容易发现，梯度向量垂直所在位置等高线。某点梯度向量越长，即向量模越大，这说明该处越陡峭。相反，如果梯度向量模越小，这说明该点越平坦。特殊情况是，梯度向量为 $\boldsymbol{0}$ 向量时，这一点便是驻点，该点切平面平行于水平面。

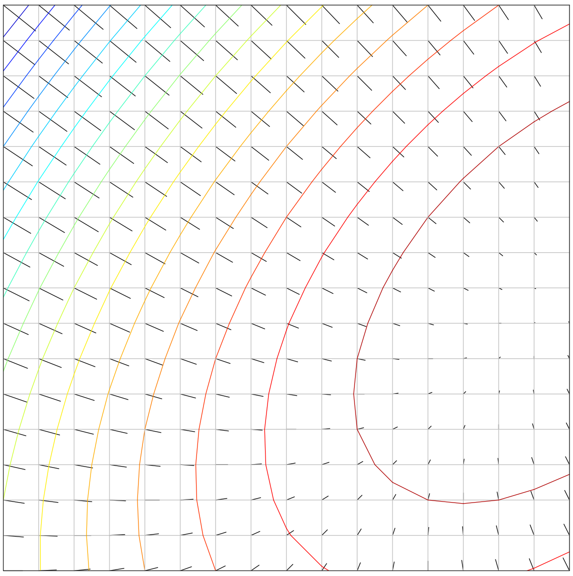


图 3. 梯度向量场

下面我们来看三个例子。

例子：一次函数

给定二元一次 $f(x_1, x_2)$ 函数如下：

$$f(x_1, x_2) = x_1 + x_2 \tag{24}$$

如图 2 (a) 所示，这个函数在三维空间的形状是个平面。

(24) 函数 $f(\boldsymbol{x})$ 的梯度向量定义如下：

$$\nabla f(\boldsymbol{x}) = \begin{bmatrix} \dfrac{\partial f}{\partial x_1} \\ \dfrac{\partial f}{\partial x_2} \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \end{bmatrix} \tag{25}$$

如图 2 (b) 所示，(24) 中给定的二元一次函数的梯度向量的方向和大小不随位置改变。

不存在任何限制条件的话，这个平面不存在任何极值点。本书后续会专门讲解直线、平面和超平面。

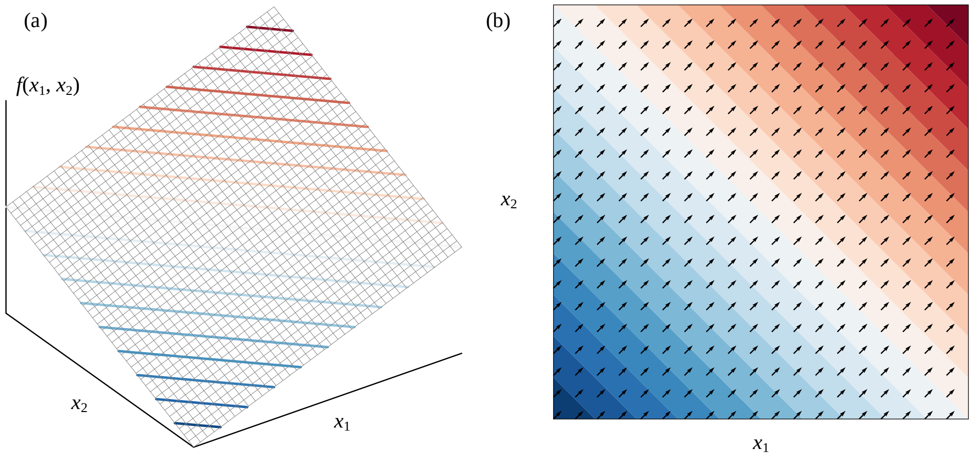


图 4. 平面的梯度向量场

例子：二次函数

给定第二个例子，$f(x_1, x_2)$ 为二元二次函数：

$$f(x_1, x_2) = x_1^2 + x_2^2 \tag{26}$$

图 5 (a) 告诉我们函数的图像是个开口朝上的正圆抛物面，曲面显然存在最小值点，位于 (0, 0)。

(26) 函数 $f(\boldsymbol{x})$ 的梯度向量定义如下：

$$\nabla f(\boldsymbol{x}) = \begin{bmatrix} \frac{\partial f}{\partial x_1} \\ \frac{\partial f}{\partial x_2} \end{bmatrix} = \begin{bmatrix} 2x_1 \\ 2x_2 \end{bmatrix} \tag{27}$$

观察图 5 (b)，容易发现越靠近 (0, 0)，也就是最小值点附近，曲面梯度向量的模越小。在 (0, 0) 处，梯度向量为 $\boldsymbol{0}$。也就是说，该点处 $f(x_1, x_2)$ 对 x_1 和 x_2 偏导数都为 0。

而且，梯度向量垂直于等高线，指向最小值点的反方向，即上山方向。

如果我们现在处于曲面上某一点，沿着梯度下降方向，即下山方向，行走，最终我们会到达最小值点处。这个思路就是基于梯度的优化方法。当然，我们需要制定一个下山的策略，比如下山的步伐怎么确定？路径怎么规划？怎么判定是否到达极值点？不同的基于梯度的优化方法在具体下山策略上有差别。这些内容，我们会在本系列丛书后续分册中讨论。

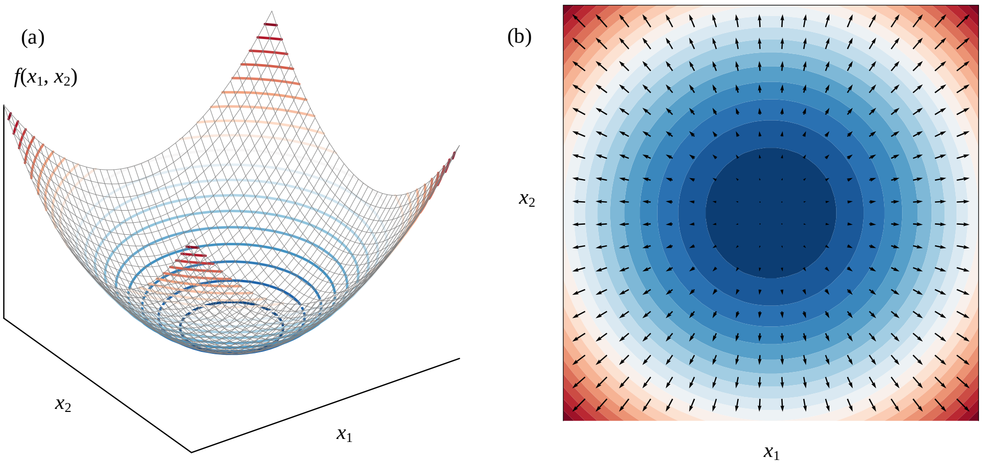


图 5. 正圆抛物面的向量场

例子：高斯函数一阶偏导

给定 $f(x_1, x_2)$ 函数如下：

$$f\left(x_1, x_2\right) = x_1 \exp\left(-\left(x_1^2 + x_2^2\right)\right) \tag{28}$$

图 6 (a) 所示为函数曲面，它存在一个最大值点和一个最小值点。

函数 $f(\boldsymbol{x})$ 的梯度向量定义如下：

$$\nabla f\left(\boldsymbol{x}\right) = \begin{bmatrix} \dfrac{\partial f}{\partial x_1} \\ \dfrac{\partial f}{\partial x_2} \end{bmatrix} = \begin{bmatrix} -2x_1^2 \exp\left(-\left(x_1^2 + x_2^2\right)\right) + \exp\left(-\left(x_1^2 + x_2^2\right)\right) \\ -x_1 x_2 \exp\left(-\left(x_1^2 + x_2^2\right)\right) \end{bmatrix} \tag{29}$$

图 6 (b) 中，最大值点附近，梯度向量均指向最大值点。最小值点附近，梯度向量均背离最小值点。

在最大值和最小值点处，梯度向量都是 $\boldsymbol{0}$ 向量。

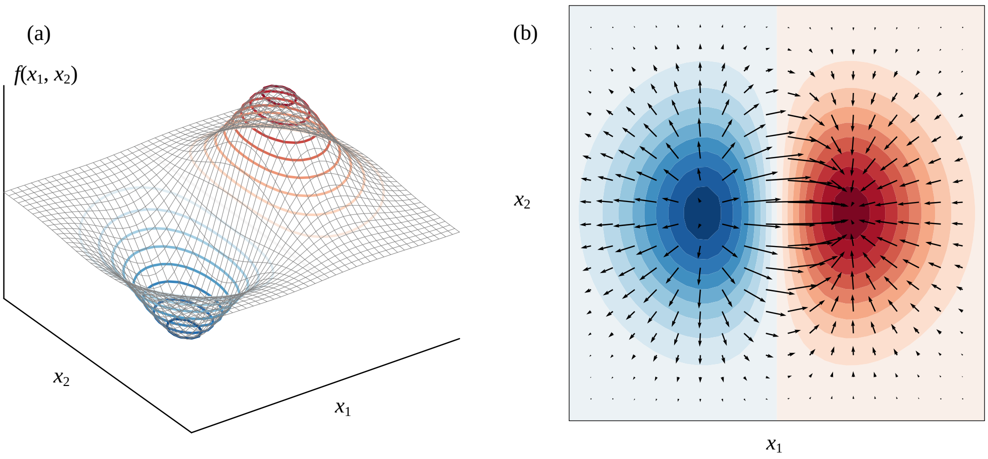


图 6. $x_1 \exp\left(-\left(x_1^2 + x_2^2\right)\right)$ 的梯度向量场

请大家修改 Bk4_Ch17_01.py 并绘制图 4、图 5、图 6。

```
# Bk4_Ch17_01.py

import sympy
import numpy as np
from sympy.functions import exp

#define symbolic vars, function
x1,x2 = sympy.symbols('x1 x2')

f_x = x1*exp(-(x1**2 + x2**2))

print(f_x)

#take the gradient symbolically
grad_f = [sympy.diff(f_x,var) for var in (x1,x2)]
print(grad_f)

f_x_fcn = sympy.lambdify([x1,x2],f_x)

#turn into a bivariate lambda for numpy
grad_fcn = sympy.lambdify([x1,x2],grad_f)

import matplotlib.pyplot as plt

xx1, xx2 = np.meshgrid(np.linspace(-2,2,40),np.linspace(-2,2,40))

# coarse mesh
xx1_, xx2_ = np.meshgrid(np.linspace(-2,2,20),np.linspace(-2,2,20))
V = grad_fcn(xx1_,xx2_)


ff_x = f_x_fcn(xx1,xx2)

color_array = np.sqrt(V[0]**2 + V[1]**2)

# 3D visualization
ax = plt.figure().add_subplot(projection='3d')
ax.plot_wireframe(xx1, xx2, ff_x, rstride=1,
```

```
                  cstride=1, color = [0.5,0.5,0.5],
                  linewidth = 0.2)
ax.contour3D(xx1, xx2, ff_x, 20, cmap = 'RdBu_r')

ax.xaxis.set_ticks([])
ax.yaxis.set_ticks([])
ax.zaxis.set_ticks([])
plt.xlim(-2,2)
plt.ylim(-2,2)
ax.view_init(30, -125)
ax.set_xlabel('$x_1$')
ax.set_ylabel('$x_2$')
ax.set_zlabel('$f(x_1,x_2)$')
plt.tight_layout()


# 2D visualization
fig, ax = plt.subplots()

plt.contourf(xx1, xx2, ff_x,20, cmap = 'RdBu_r')

plt.quiver (xx1_, xx2_, V[0], V[1],
            angles='xy', scale_units='xy',
            edgecolor='none', facecolor= 'k')

plt.show()
ax.set_aspect('equal')
ax.xaxis.set_ticks([])
ax.yaxis.set_ticks([])
ax.set_xlabel('$x_1$')
ax.set_ylabel('$x_2$')
plt.tight_layout()
```

17.3 法向量：垂直于切平面

对于 $y = f(\boldsymbol{x})$ 函数，我们可以把它看做是 $f(\boldsymbol{x}) - y = 0$ 的一个等式。定义 $F(\boldsymbol{x}, y)$ 如下：

$$F(\boldsymbol{x}, y) = f(\boldsymbol{x}) - y \tag{30}$$

函数 $F(\boldsymbol{x}, y)$梯度向量为：

$$\nabla F(\boldsymbol{x}, y) = \begin{bmatrix} \nabla f(\boldsymbol{x}) \\ -1 \end{bmatrix} \tag{31}$$

这个梯度向量就是 $f(\boldsymbol{x})$ 点 $\boldsymbol{x}$ 处的法向量 $\boldsymbol{n}$：

$$\boldsymbol{n} = \begin{bmatrix} \nabla f(\boldsymbol{x}) \\ -1 \end{bmatrix} \tag{32}$$

如图 7 所示，以二元函数 $f(\boldsymbol{x})$为例，$\boldsymbol{n}$ 向水平面投影得到梯度向量 $\nabla f(\boldsymbol{x})$。

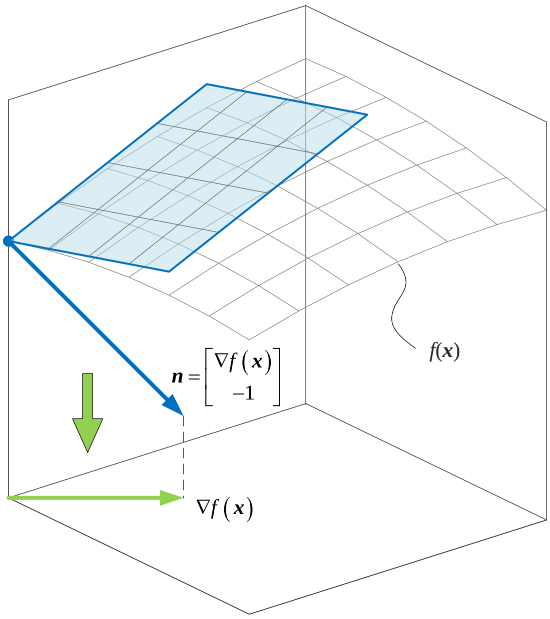


图 7. $\boldsymbol{n}$ 向水平面投影得到梯度向量

图 8 所示为某个二元函数 $f(\boldsymbol{x})$ 曲面上不同点处的法向量，这些法向量向 x_1x_2 平面投影便得到 $f(\boldsymbol{x})$ 的梯度向量。这个视角非常重要，本书后续还会再谈几次。

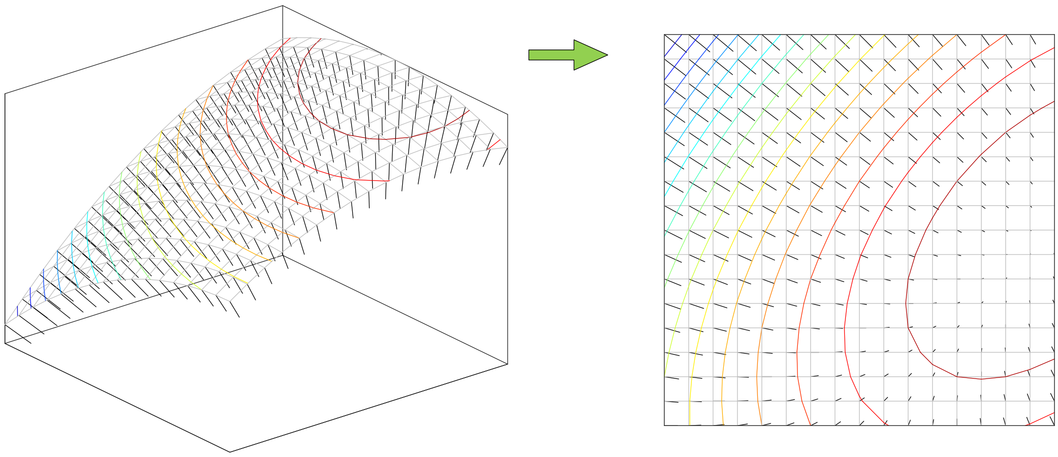


图 8. 曲面法向量场投影得到梯度向量场

图 9 给出的是 (28) 中函数在不同点处的法向量场，这些向量向水平面投影便得到图 6 (b)。

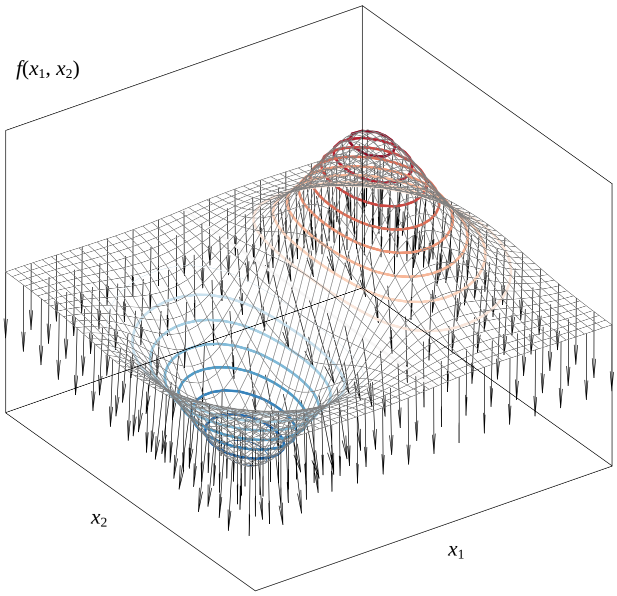


图 9. $x_1 \exp\left(-\left(x_1^2 + x_2^2\right)\right)$ 的法向量场

Bk4_Ch17_02.py 绘制图 9。

```
# Bk4_Ch17_02.py

import sympy
import numpy as np
from sympy.functions import exp

#define symbolic vars, function
x1,x2 = sympy.symbols('x1 x2')

f_x = x1*exp(-(x1**2 + x2**2))

print(f_x)

#take the gradient symbolically
grad_f = [sympy.diff(f_x,var) for var in (x1,x2)]
print(grad_f)

f_x_fcn = sympy.lambdify([x1,x2],f_x)

#turn into a bivariate lambda for numpy
grad_fcn = sympy.lambdify([x1,x2],grad_f)

import matplotlib.pyplot as plt

xx1, xx2 = np.meshgrid(np.linspace(-2,2,40),np.linspace(-2,2,40))

# coarse mesh
xx1_, xx2_ = np.meshgrid(np.linspace(-2,2,15),np.linspace(-2,2,15))
V = grad_fcn(xx1_,xx2_)


ff_x = f_x_fcn(xx1,xx2)
ff_x_ = f_x_fcn(xx1_,xx2_)
```

```
color_array = np.sqrt(V[0]**2 + V[1]**2)

# 3D visualization + vectors
ax = plt.figure().add_subplot(projection='3d')
ax.plot_wireframe(xx1, xx2, ff_x, rstride=1,
                  cstride=1, color = [0.5,0.5,0.5],
                  linewidth = 0.2)
ax.contour3D(xx1, xx2, ff_x, 20, cmap = 'RdBu_r')

lengths = np.sqrt(V[0]**2+V[1]**2+1**2)

for x1,y1,z1,u1,v1,w1,l in zip(xx1_.flatten(),
                               xx2_.flatten(),
                               ff_x_.flatten(),
                               V[0].flatten(),
                               V[1].flatten(),
                               V[0].flatten()*0 - 1,
                               lengths.flatten()):

    ax.quiver(x1, y1, z1, u1, v1, w1, length=l*0.2,
              color = 'k')

ax.xaxis.set_ticks([]); ax.yaxis.set_ticks([]); ax.zaxis.set_ticks([])
plt.xlim(-2,2); plt.ylim(-2,2)
ax.view_init(30, -125)
ax.set_xlabel('$x_1$'); ax.set_ylabel('$x_2$'); ax.set_zlabel('$f(x_1,x_2)$')
plt.tight_layout()
```

17.4 方向性微分：函数任意方向的变化

《数学要素》一册提到过，光滑曲面 $f(x_1, x_2)$ 某点的切线有无数条，如图 10 所示。而偏导数仅仅分析了其中两条切线的变化率，即沿着 x_1 和 x_2 轴方向。本节将介绍一个全新的数学工具——方向性微分 (directional derivative)，它可以分析光滑曲面某点处不同方向切线的变化率。

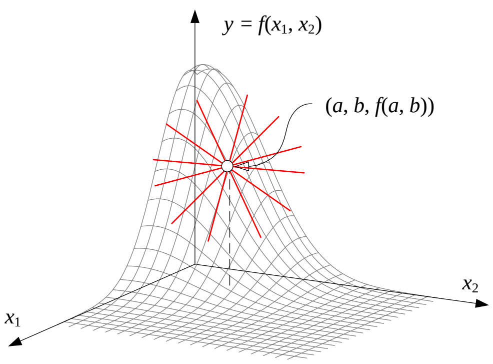


图 10. 光滑曲面 $f(x_1, x_2)$ 某点的切线有无数条

以二元函数为例

二元函数 $f(x_1, x_2)$ 写作 $f(\boldsymbol{x})$：

$$f(\boldsymbol{x}) = f(x_1, x_2) \tag{33}$$

x_1x_2 平面上，$P(x_1, x_2)$ 点处，任意偏离 P 点微小移动 $(\Delta x_1, \Delta x_2)$ 都导致 $f(\boldsymbol{x})$ 大小发生变化，函数值高度变化为：

$$\Delta f = f(\boldsymbol{x} + \Delta\boldsymbol{x}) - f(\boldsymbol{x}) = f(x_1 + \Delta x_1, x_2 + \Delta x_2) - f(x_1, x_2) \tag{34}$$

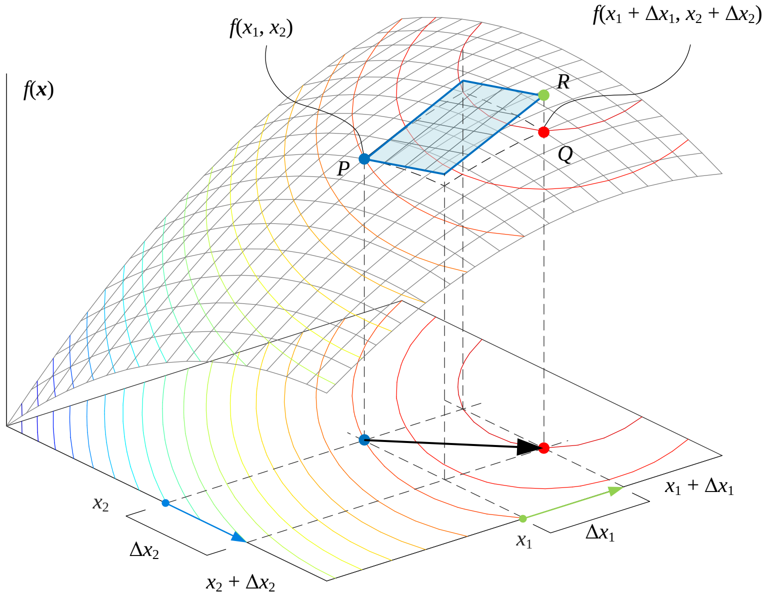


图 11. 曲面从 P 点线性移动到 R 点对应位置变化

用一阶偏微分近似求解 Δf：

$$\begin{aligned}\Delta f &= f(\boldsymbol{x} + \Delta\boldsymbol{x}) - f(\boldsymbol{x}) \\ &= f(x_1 + \Delta x_1, x_2 + \Delta x_2) - f(x_1, x_2) \approx \frac{\partial f(\boldsymbol{x})}{\partial x_1}\Delta x_1 + \frac{\partial f(\boldsymbol{x})}{\partial x_2}\Delta x_2\end{aligned} \tag{35}$$

上式便是从书之前讲过的二元函数泰勒一阶展开。如图 11 所示，上式相当于用二元一次函数斜面近似函数曲面，即：

$$f(x_1 + \Delta x_1, x_2 + \Delta x_2) \approx f(x_1, x_2) + \frac{\partial f(\boldsymbol{x})}{\partial x_1}\Delta x_1 + \frac{\partial f(\boldsymbol{x})}{\partial x_2}\Delta x_2 \tag{36}$$

几何视角

投影位移量 $(\Delta x_1, \Delta x_2)$ 一致情况下，沿着曲面，从 $P(x_1, x_2)$ 点运动到 $Q(x_1 + \Delta x_1, x_2 + \Delta x_2)$点。而沿着 P 点切面，移动到了 R 点。

R 点和 Q 点高度差便是估算误差。

图 12 为图 11 局部放大图，这张图更清晰地展示估算过程。

在 (x_1, x_2) 点处，二元函数曲面的高度为 $f(x_1, x_2)$。沿着蓝色斜面运动，在 x_1 方向上移动 Δx_1 带来的高度变化为 $\left.\frac{\partial f(\boldsymbol{x})}{\partial x_1}\right|_P \Delta x_1$。同样沿着蓝色斜面运动，在 x_2 方向上移动 Δx_2 带来的高度变化为 $\left.\frac{\partial f(\boldsymbol{x})}{\partial x_2}\right|_P \Delta x_2$。两个高度变化之和便是对的 Δf 一阶逼近。

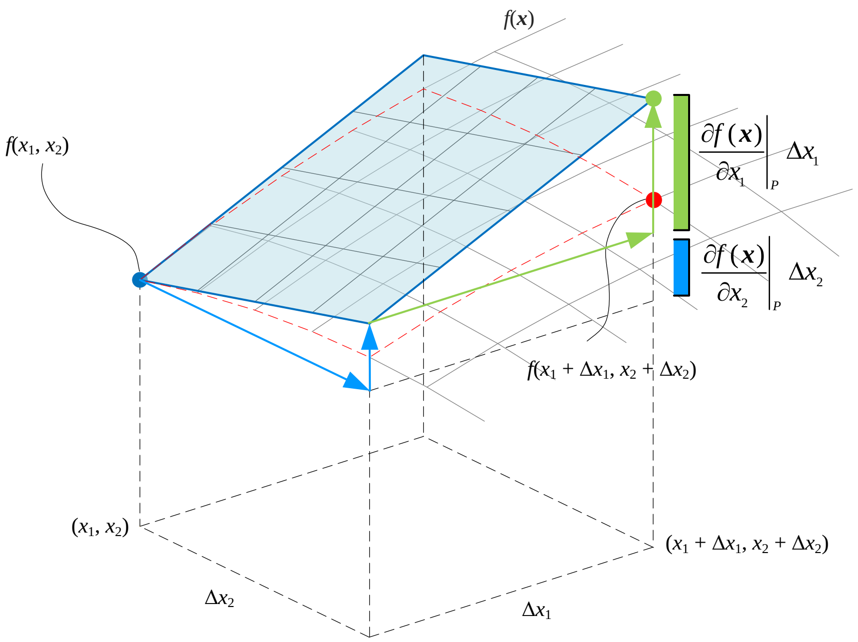


图 12. 二元函数一阶泰勒展开估算

(35) 可以写成两个向量内积关系：

$$\Delta f \approx \begin{bmatrix} \Delta x_1 \\ \Delta x_2 \end{bmatrix} \cdot \begin{bmatrix} \frac{\partial f(\boldsymbol{x})}{\partial x_1} \\ \frac{\partial f(\boldsymbol{x})}{\partial x_2} \end{bmatrix} = \begin{bmatrix} \Delta x_1 \\ \Delta x_2 \end{bmatrix}^{\mathrm{T}} \begin{bmatrix} \frac{\partial f(\boldsymbol{x})}{\partial x_1} \\ \frac{\partial f(\boldsymbol{x})}{\partial x_2} \end{bmatrix} \tag{37}$$

换个角度，向量 $(\Delta x_1, \Delta x_2)$ 决定了 P 点方向微分方向，如图 13 所示。

也就是说，有了向量 $(\Delta x_1, \Delta x_2)$，我们可以量化二元函数 $f(x_1, x_2)$ 在任意方向的函数变化，以及变化率。

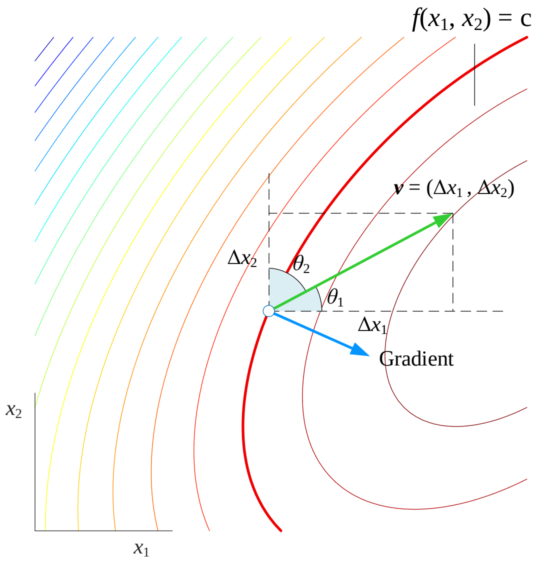


图 13. x_1x_2 平面上方向微分

单位向量

x_1x_2 平面上，给定一个方向，用单位向量 $\boldsymbol{v}$ 表示：

$$\boldsymbol{v} = \begin{bmatrix} v_1 \\ v_2 \end{bmatrix} \tag{38}$$

令单位向量 $\boldsymbol{v}$ 为：

$$\boldsymbol{v} = \begin{bmatrix} \cos\theta_1 \\ \cos\theta_2 \end{bmatrix} \tag{39}$$

图 13 给出 θ_1 和 θ_2 角度定义。

对于上述二元函数，定义方向性微分为：

$$\nabla_v f(\boldsymbol{x}) = \boldsymbol{v} \cdot \nabla f(\boldsymbol{x}) = \boldsymbol{v}^{\mathrm{T}} \nabla f(\boldsymbol{x}) = \langle \boldsymbol{v}, \nabla f(\boldsymbol{x}) \rangle \tag{40}$$

展开得到方向导数和偏导之间关系为：

$$\nabla_v f(\boldsymbol{x}) = \frac{\partial f(\boldsymbol{x})}{\partial x_1} \cos\theta_1 + \frac{\partial f(\boldsymbol{x})}{\partial x_2} \cos\theta_2 = \begin{bmatrix} \cos\theta_1 \\ \cos\theta_2 \end{bmatrix}^{\mathrm{T}} \begin{bmatrix} \dfrac{\partial f(\boldsymbol{x})}{\partial x_1} \\ \dfrac{\partial f(\boldsymbol{x})}{\partial x_2} \end{bmatrix} \tag{41}$$

(40) 也适用于多元函数。

不同方向

根据向量内积法则，(40) 可以写成：

$$\begin{aligned}\nabla_v f(\boldsymbol{x}) &= \nabla f(\boldsymbol{x}) \cdot \boldsymbol{v} \\ &= \|\nabla f(\boldsymbol{x})\| \cdot \|\boldsymbol{v}\| \cos(\theta) \\ &= \|\nabla f(\boldsymbol{x})\| \cos(\theta)\end{aligned} \tag{42}$$

其中，θ 为 $\nabla f(\boldsymbol{x})$ 和 $\boldsymbol{v}$ 之间夹角。

图 14 所示为 x_1x_2 平面上六种不同方向微分情况。

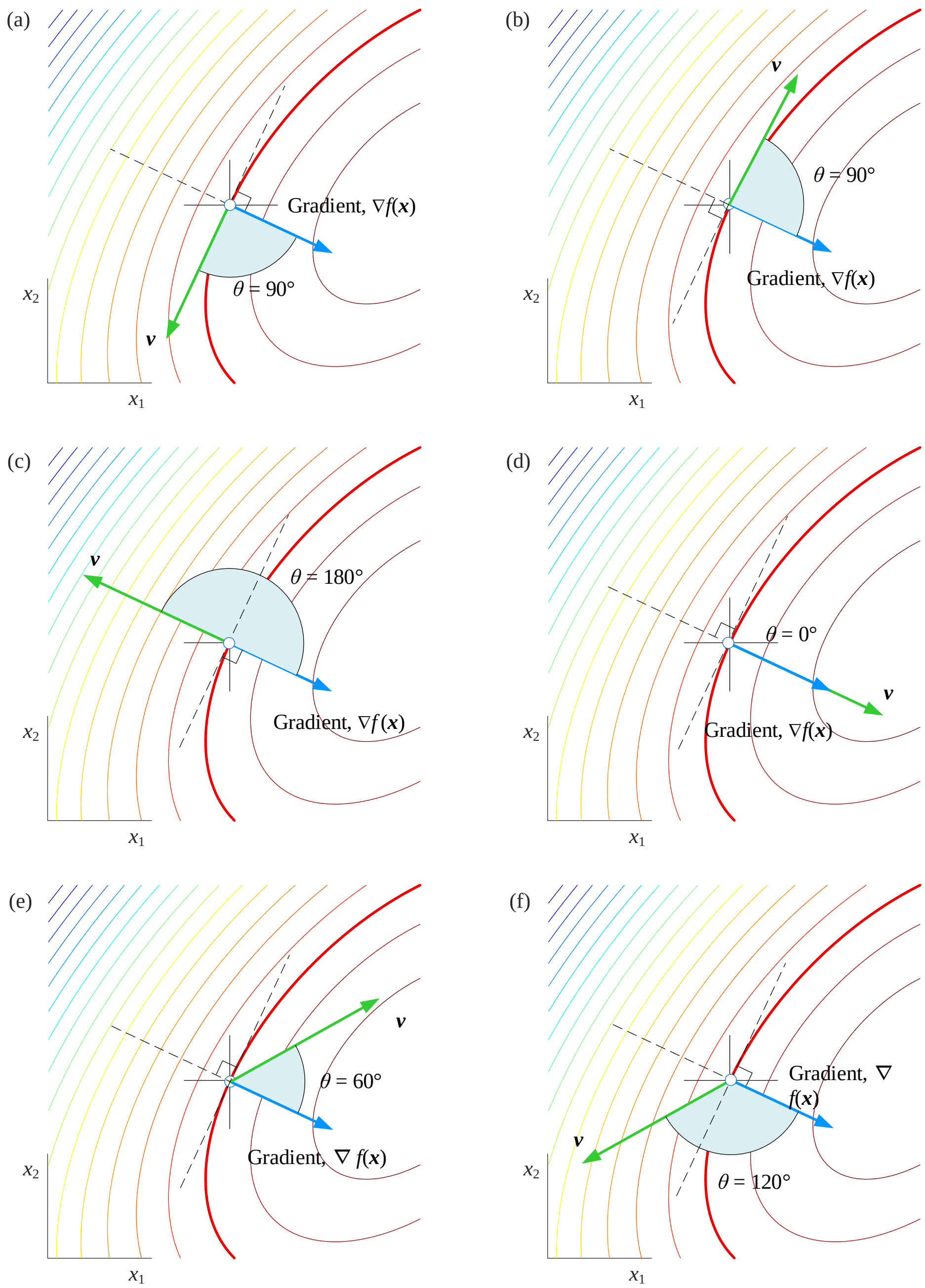


图 14. x_1x_2 平面上六种方向微分情况

若 $\theta = 90°$，则说明方向导数沿着等高线切向方向，函数值不会有任何变化，如图 14 (a) 和 (b) 所示。

如图 14 (c)，若 $\theta = 180°$，方向导数沿着梯度相反方向，即梯度下降方向。这是函数值下降最快方向。

如图 14 (d)，$\theta = 0°$，方向导数和梯度同向，这是函数值上升最快方向。用 $\nabla f\left(\boldsymbol{x}\right)$ 表达 $\boldsymbol{v}$：

$$\boldsymbol{v} = \eta \nabla f\left(\boldsymbol{x}\right) \quad \eta > 0 \tag{43}$$

因此，

$$\begin{aligned} f(\boldsymbol{x}+\Delta \boldsymbol{x}) - f(\boldsymbol{x}) &= f(x_1+\Delta x_1, x_2+\Delta x_2) \\ &\approx \nabla f(\boldsymbol{x}) \cdot \boldsymbol{v} \\ &= \nabla f(\boldsymbol{x}) \cdot \eta \nabla f(\boldsymbol{x}) \\ &= \eta \left\| \nabla f(\boldsymbol{x}) \right\|^2 > 0 \end{aligned} \tag{44}$$

当 θ 为锐角，函数变化大于 0，函数值上升，如图 14 (e)。

当 θ 为钝角，函数变化小于 0，函数值下降，如图 14 (f)。另外，$\nabla f\left(\boldsymbol{x}\right)$ 和向量 $\boldsymbol{v}$ 关系，和本书前文介绍的投影 (projection) 几乎完全一致。

$\boldsymbol{v} = [1, 0]^{\mathrm{T}}$ 对应 $f(x_1, x_2)$ 对 x_1 偏导。$\boldsymbol{v} = [0, 1]^{\mathrm{T}}$ 对应 $f(x_1, x_2)$ 对 x_2 偏导。可见，方向性微分比偏导更灵活。

方向导数可以用来研究多元函数在某一特定方向的函数变化率，机器学习和深度学习很多算法在求解优化问题时都会用到方向导数这个重要的数学工具。

17.5 泰勒展开：一元到多元

从书《数学要素》中介绍泰勒展开 (Taylor series expansion)。本节将一元泰勒展开扩展到多元函数应用。

一元泰勒展开

一元函数 $f(x)$ 泰勒展开形式为：

$$\begin{aligned} f\left(x\right) &= \sum_{n=0}^{\infty} \frac{f^{(n)}\left(a\right)}{n!}\left(x-a\right)^n \\ &= f\left(a\right) + \frac{f'\left(a\right)}{1!}\left(x-a\right) + \frac{f''\left(a\right)}{2!}\left(x-a\right)^2 + \frac{f'''\left(a\right)}{3!}\left(x-a\right)^3 + \cdots \end{aligned} \tag{45}$$

上式保留一阶导数部分就是线性逼近：

$$f\left(x\right) \approx f\left(a\right) + f'\left(a\right)\left(x-a\right) \tag{46}$$

线性逼近

更一般情况，对于多元函数 $f(\boldsymbol{x})$，当 $\boldsymbol{x}$ 足够靠近 $\boldsymbol{x}_P$ 时，$f(\boldsymbol{x})$ 函数值用泰勒一阶展开逼近，如下式：

$$\begin{aligned} f(\boldsymbol{x}) &\approx f(\boldsymbol{x}_P) + \nabla f(\boldsymbol{x}_P)^{\mathrm{T}}(\boldsymbol{x} - \boldsymbol{x}_P) \\ &= f(\boldsymbol{x}_P) + \nabla f(\boldsymbol{x}_P)^{\mathrm{T}} \Delta \boldsymbol{x} \end{aligned} \tag{47}$$

$\boldsymbol{x}_P$ 为泰勒级数展开点 (expansion point of Taylor series)，$\nabla f(\boldsymbol{x}_P)$ 为多元函数 $f(\boldsymbol{x})$ 在 $\boldsymbol{x}_P$ 处梯度列向量。

图 15 所示一元和二元线性逼近的相似处。一元线性逼近是用切线逼近曲线，二元线性逼近是用切面逼近曲面。

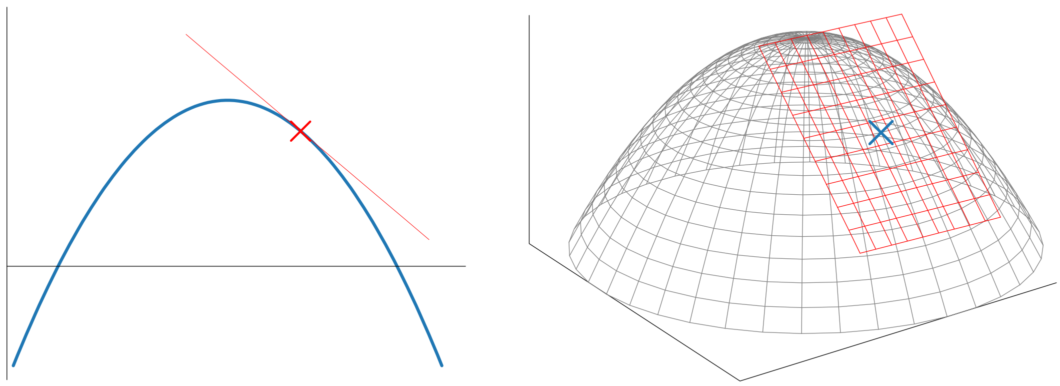


图 15. 一元到二元线性逼近

二次逼近

多元函数 $f(\boldsymbol{x})$ 泰勒二阶级数展开式矩阵运算如下：

$$\begin{aligned} f(\boldsymbol{x}) &\approx f(\boldsymbol{x}_P) + \nabla f(\boldsymbol{x}_P)^{\mathrm{T}}(\boldsymbol{x} - \boldsymbol{x}_P) + \frac{1}{2}(\boldsymbol{x} - \boldsymbol{x}_P)^{\mathrm{T}} \nabla^2 f(\boldsymbol{x}_P)(\boldsymbol{x} - \boldsymbol{x}_P) \\ &= f(\boldsymbol{x}_P) + \nabla f(\boldsymbol{x}_P)^{\mathrm{T}} \Delta \boldsymbol{x} + \frac{1}{2} \Delta \boldsymbol{x}^{\mathrm{T}} \nabla^2 f(\boldsymbol{x}_P) \Delta \boldsymbol{x} \\ &= f(\boldsymbol{x}_P) + \nabla f(\boldsymbol{x}_P)^{\mathrm{T}} \Delta \boldsymbol{x} + \frac{1}{2} \Delta \boldsymbol{x}^{\mathrm{T}} \boldsymbol{H} \Delta \boldsymbol{x} \end{aligned} \tag{48}$$

上式就是二次逼近。

二次曲面

本章最后讨论二次曲面在某点切面。采用圆锥曲线一般式，令 $y = f(x_1, x_2)$：

$$y = f(x_1, x_2) = Ax_1^2 + Bx_1x_2 + Cx_2^2 + Dx_1 + Ex_2 + F \tag{49}$$

$y = f(x_1, x_2)$ 写成矩阵运算：

$$y = f(x_1, x_2) = \frac{1}{2}\begin{bmatrix} x_1 \\ x_2 \end{bmatrix}^{\mathrm{T}} \begin{bmatrix} 2A & B \\ B & 2C \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} D \\ E \end{bmatrix}^{\mathrm{T}} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + F \tag{50}$$

构造函数 $F(x_1, x_2, y)$：

$$F(x_1, x_2, y) = Ax_1^2 + Bx_1x_2 + Cx_2^2 + Dx_1 + Ex_2 + F - y \tag{51}$$

在三维空间中一点 $P\ (p_1, p_2, p_y)$，$F(x_1, x_2, y)$ 曲面法向量 $\boldsymbol{n}_P$ 通过下式得到：

$$\boldsymbol{n}_P = \begin{bmatrix} \dfrac{\partial F}{\partial x_1} \\ \dfrac{\partial F}{\partial x_2} \\ \dfrac{\partial F}{\partial y} \end{bmatrix}\Bigg|_{(p_1, p_2, p_y)} = \begin{bmatrix} 2Ap_1 + Bp_2 + D \\ Bp_1 + 2Cp_2 + E \\ -1 \end{bmatrix} \tag{52}$$

切面上任意一点 (x_1, x_2, y) 和切点 P 构成向量 $\boldsymbol{p}$：

$$\boldsymbol{p} = \begin{bmatrix} x_1 - p_1 \\ x_2 - p_2 \\ y - p_y \end{bmatrix} \tag{53}$$

$\boldsymbol{p}$ 垂直于 $\boldsymbol{n}_P$，因此两者向量内积为 0，得到如下等式：

$$(2Ap_1 + Bp_2 + D)(x_1 - p_1) + (Bp_1 + 2Cp_2 + E)(x_2 - p_2) - y + p_y = 0 \tag{54}$$

整理得到切面 $t(x_1, x_2)$ 解析式：

$$t(x_1, x_2) = (2Ap_1 + Bp_2 + D)(x_1 - p_1) + (Bp_1 + 2Cp_2 + E)(x_2 - p_2) + p_y \tag{55}$$

另外，以上切面解析式就是 P 点泰勒一次逼近：

$$t(x_1, x_2) = f(p_1, p_2) + \nabla f(p_1, p_2)^{\mathrm{T}} \begin{bmatrix} x_1 - p_1 \\ x_2 - p_2 \end{bmatrix} \tag{56}$$

$y = f(x_1, x_2)$ 在 P 点梯度向量：

$$\nabla f(p_1, p_2) = \begin{bmatrix} 2A & B \\ B & 2C \end{bmatrix} \begin{bmatrix} p_1 \\ p_2 \end{bmatrix} + \begin{bmatrix} D \\ E \end{bmatrix} = \begin{bmatrix} 2Ap_1 + Bp_2 + D \\ Bp_1 + 2Cp_2 + E \end{bmatrix} \tag{57}$$

将 (57) 代入 (56)，同样可以得到 (55) 结果。

举个例子

以下二次曲面为例讲解求解 $y = f(x_1, x_2)$ 曲面某点处切面过程。给定二元函数 $y = f(x_1, x_2)$，

$$y = f(x_1, x_2) = -4x_1^2 - 4x_2^2 \tag{58}$$

构造一个 $F(x_1, x_2, y)$ 函数如下：

$$F(x_1, x_2, y) = -4x_1^2 - 4x_2^2 - y \tag{59}$$

在 $F(x_1, x_2, y)$ 曲面上任意一点 (x_0, y_0, z_0) 法向量 $\boldsymbol{n}$ 通过下式得到：

$$\boldsymbol{n} = \begin{bmatrix} -8x_1 \\ -8x_2 \\ -1 \end{bmatrix} \tag{60}$$

图 16 (a) 所示为二次曲面和曲面上 A 点 $(0, -1.5, -9)$ 切面。

将 A 点坐标带入上式，得到 A 点处曲面切面解析式，如下：

$$t(x_1, x_2) = 12x_2 + 9 \tag{61}$$

图 16 (b) 所示为 B 点 $(-1.5, 0, -9)$ 曲面切面。根据上述思路，请读者自行求解切面解析式。

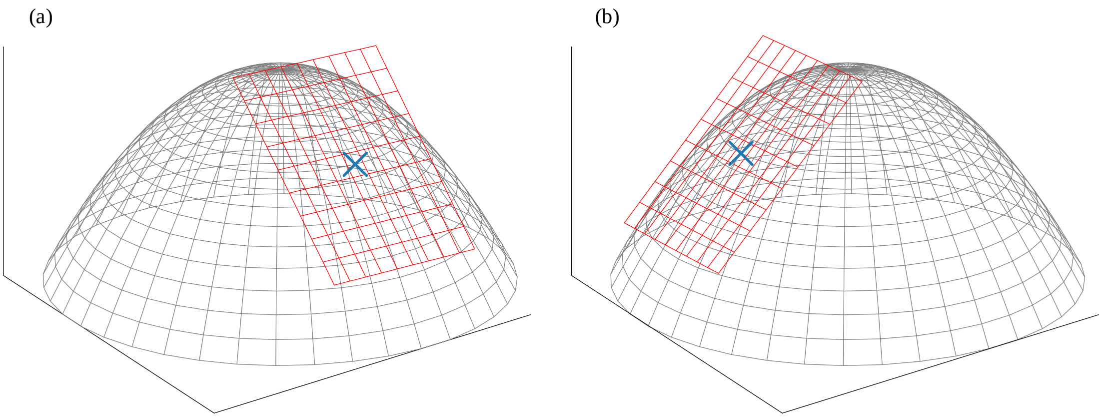


图 16. 二次凹曲面 A 点处切面

Bk4_Ch17_03.py 绘制图 16。

```
# Bk4_Ch17_03.py

import sympy
from sympy import Matrix, Transpose
import numpy as np
from sympy.functions import exp
import matplotlib.pyplot as plt
```

```python
def mesh_circ(c1, c2, r, num):

    theta = np.arange(0,2*np.pi+np.pi/num,np.pi/num)
    r     = np.arange(0,r,r/num)
    theta,r = np.meshgrid(theta,r)
    xx1 = np.cos(theta)*r + c1
    xx2 = np.sin(theta)*r + c2

    return xx1, xx2


#define symbolic vars, function
x1,x2,p1,p2 = sympy.symbols('x1 x2 p1 p2')

f_x = -4*x1**2 -4*x2**2
f_p = -4*p1**2 -4*p2**2

print(f_x)

#take the gradient symbolically
grad_f = [sympy.diff(f_p,var) for var in (p1,p2)]
print(grad_f)

f_x_fcn = sympy.lambdify([x1,x2],f_x)

t_x = Matrix(grad_f).T*Matrix([[x1 - p1], [x2 - p2]]) + Matrix([f_p])
print(t_x)

t_x_fcn = sympy.lambdify([x1,x2,p1,p2],t_x)

#turn into a bivariate lambda for numpy
grad_fcn = sympy.lambdify([x1,x2],grad_f)

xx1, xx2 = mesh_circ(0, 0, 3, 20)

# expansion point
p1 = -1.5
p2 = 0
py = f_x_fcn(p1,p2)

# coarse mesh
xx1_, xx2_ = np.meshgrid(np.linspace(p1-1, p1+1, 10),
                         np.linspace(p2-1, p2+1, 10))

# quadratic surface
ff_x = f_x_fcn(xx1,xx2)

# tangent plane
tt_x = t_x_fcn(xx1_, xx2_, p1, p2)

# 3D visualization
fig = plt.figure()
ax = fig.gca(projection='3d')

ax.plot_wireframe(xx1, xx2, ff_x, rstride=1,
                  cstride=1, color = [0.5,0.5,0.5],
                  linewidth = 0.2)

ax.plot_wireframe(xx1_, xx2_, np.squeeze(tt_x), rstride=1,
                  cstride=1, color = [1,0,0],
                  linewidth = 0.2)
ax.plot(p1,p2,py,'x')

ax.xaxis.set_ticks([])
ax.yaxis.set_ticks([])
ax.zaxis.set_ticks([])
plt.xlim(-3,3)
plt.ylim(-3,3)
ax.view_init(30, -125)
ax.set_xlabel('$x_1$')
ax.set_ylabel('$x_2$')
```

```
ax.set_zlabel('$f(x_1,x_2)$')
```

本章将一元函数微分推广到多元函数，并介绍了几个重要数学工具——梯度向量、黑塞矩阵、法向量、方向导数、一次泰勒逼近、二次泰勒逼近。本书后续将利用这些数学工具分析各种数学问题。

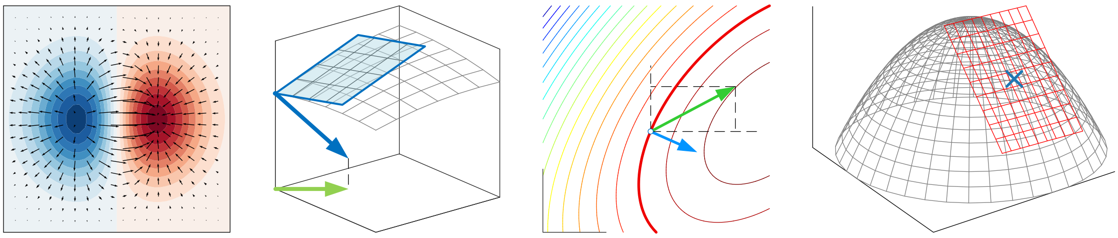


图 17. 总结本章重要内容的四副图

本章仅仅讨论了本书后续将会用到的矩阵微分法则。大家如果对这个话题感兴趣的话，推荐大家参考 *The Matrix Cookbook*。下载地址为：

https://www.math.uwaterloo.ca/~hwolkowi/matrixcookbook.pdf