15 导数

Derivative

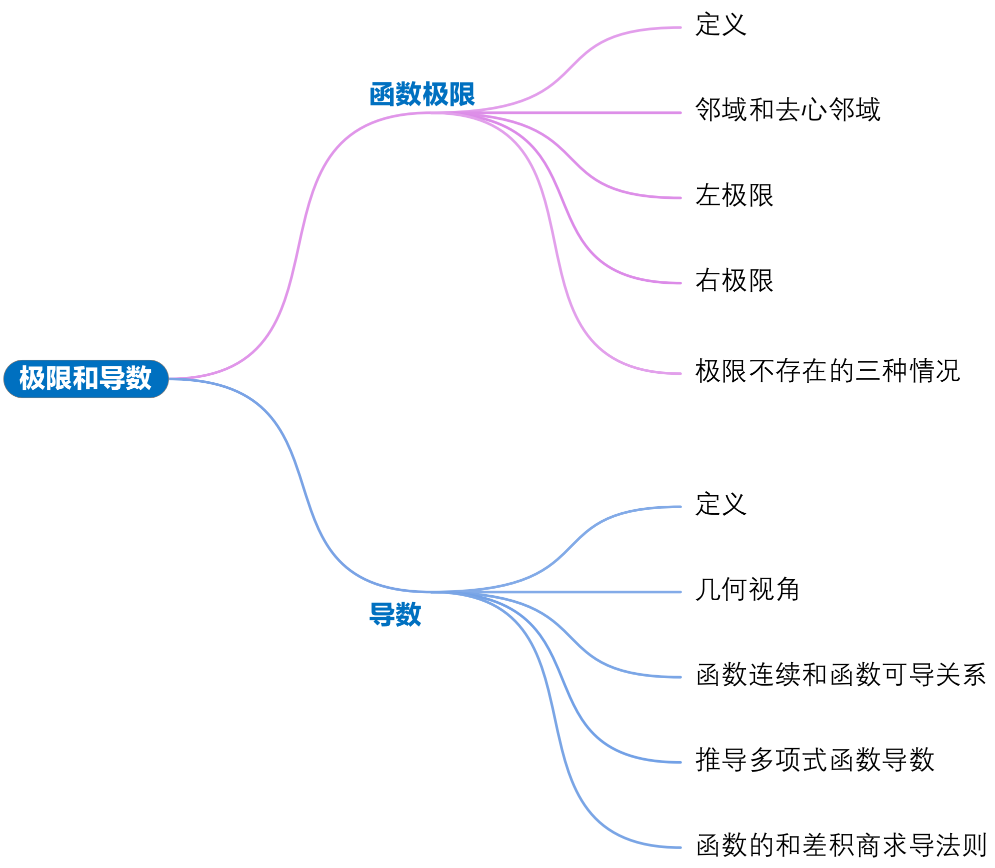
函数切线斜率，即变化率

微积分是现代数学的第一个成就，它的重要性怎么评价都不为过。我认为它比其他任何东西都更明确地定义了现代数学的起源。而作为其逻辑发展的数学分析系统，仍然是精确思维的最大技术进步。

The calculus was the first achievement of modern mathematics and it is difficult to overestimate its importance. I think it defines more unequivocally than anything else the inception of modern mathematics; and the system of mathematical analysis, which is its logical development, still constitutes the greatest technical advance in exact thinking.

—— 约翰·冯·诺伊曼 (John von Neumann) | 美国籍数学家 | 1903 ~ 1957

- `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.limit()` 求解极限
- `sympy.plot_implicit()` 绘制隐函数方程
- `sympy.series()` 求解泰勒展开级数符号式
- `sympy.symbols()` 定义符号变量


极限和导数
函数极限
定义
邻域和去心邻域
左极限
右极限
极限不存在的三种情况
导数
定义
几何视角
函数连续和函数可导关系
推导多项式函数导数
函数的和差积商求导法则

15.1 牛顿小传

“如果说我比别人看得更远，那是因为我站在巨人们的肩上。”

牛顿从伽利略手中接过了智慧火炬；1642 年年底，艾萨克·牛顿 (Sir Isaac Newton) 呱呱坠地，同年年初伽利略驾鹤西征。

艾萨克·牛顿 (Sir Isaac Newton)
英国物理学家、数学家 | 1643年 ~ 1727年
提出万有引力定律、牛顿运动定律，与莱布尼茨共同发明微积分

牛顿坐在果园里，思考物理学。苹果熟了，从树上落下，砸到了牛顿的脑门。一个惊世疑问，苹果为什么会下落？

是的，苹果为什么会下落，而不是飞向更遥远的天际？对这些问题的系统思考让牛顿提出万有引力定律。

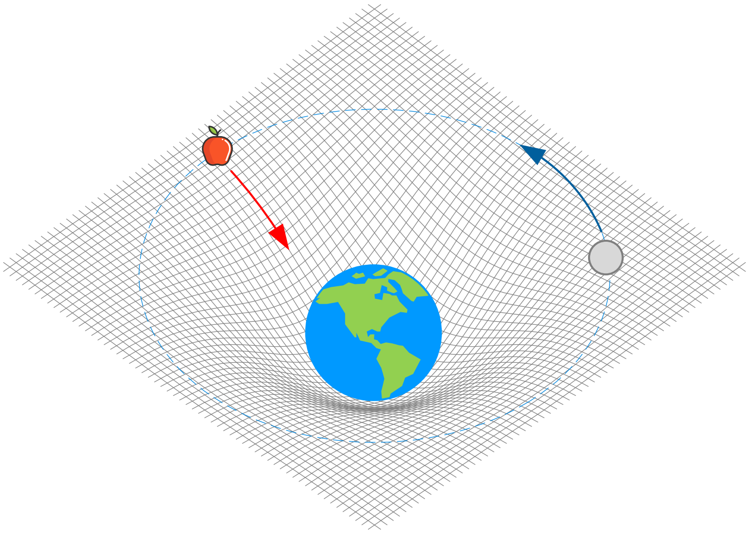


图 1. 地球引力场作用下的月球和苹果

牛顿的成就不止于此。他提出三大运动定律，并出版《自然哲学的数学原理》(*Mathematical Principles of Natural Philosophy*)，他利用三棱镜发现七色光谱，发明反射望远镜，并提出光的微

粒说，他和莱布尼茨分别独立发明微积分等等。任何人有其中任意一个贡献，就可以留名青史；然而，牛顿一个人完成上述科学进步。

自然和自然规律隐藏在黑暗之中。

上帝说：交给牛顿吧，

于是一切豁然开朗。

Nature and Nature's laws lay hid in night:

God said, Let Newton be! and all was light.

—— 亚历山大·蒲柏 (Alexander Pope) | 英国诗人 | 1688 ~ 1744

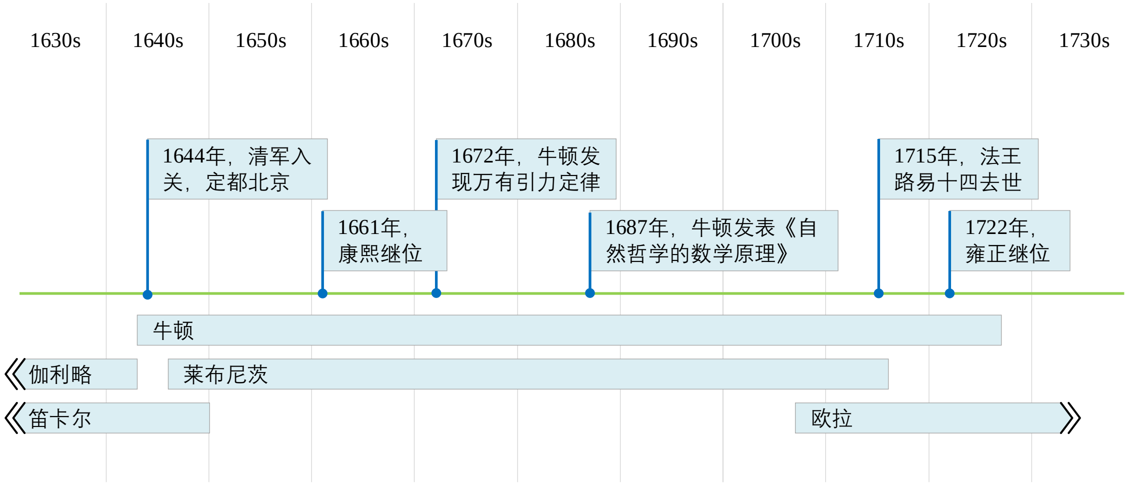


图 2. 牛顿时代时间轴

15.2 极限：研究微积分的重要数学工具

微积分 (calculus) 是研究实数域上函数的微分与积分等性质的学科；而极限是微积分最重要数学工具。**连续** (continuity)、**导数** (derivative) 和**积分** (integral) 这些概念都是通过极限来定义。

上一章简单介绍了数列极限、数列和的极限。本节主要介绍函数极限。

函数极限

首先聊一下函数极限的定义。

设函数 $f(x)$ 在点 a 的某一个去心邻域内有定义，如果存在常数 C，对于任意给定正数 ε，不管它多小，总存在正数 δ，使得 x 满足如下不等式时，

$$0<|x-a|<\delta \tag{1}$$

对应函数值 $f(x)$ 都满足，

$$|f(x)-C|<\varepsilon \tag{2}$$

常数 C 就是函数 $f(x)$ 当 $x \to a$ 时的极限，记做：

$$\lim_{x\to a} f(x)=C \tag{3}$$

举个例子，给定如下函数。

$$f(x)=\left(1+\frac{1}{x}\right)^x \tag{4}$$

如图 3 所示，当 x 趋向正无穷，函数极限为 e。

$$\lim_{x\to+\infty} f(x)=\mathrm{e} \tag{5}$$

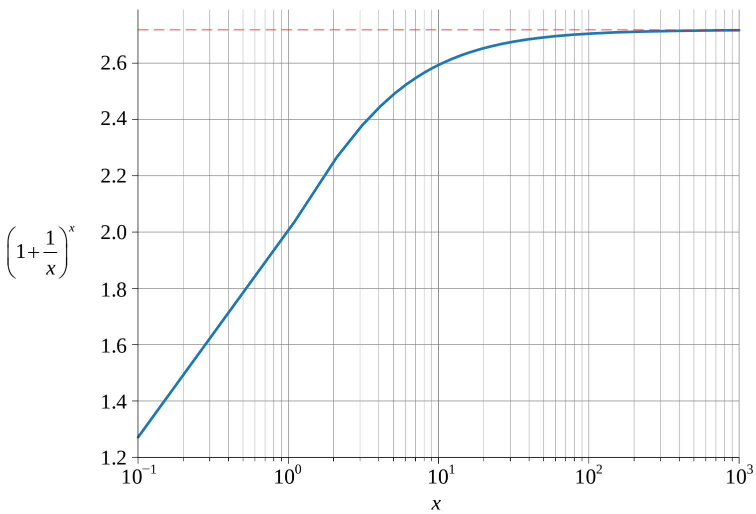


图 3. 当 x 趋向正无穷，函数 $f(x)$ 极限值

邻域

解释一下邻域这个概念；邻域 (neighbourhood) 实际上就是一个特殊的开区间。如图 4 所示，点 a 的 h ($h > 0$) 邻域满足 $a - h < x < a + h$。

a 为邻域的中心，h 为邻域的半径。而去心邻域 (deleted neighborhood 或 punctured neighborhood) 指的是，在 a 的邻域中去掉 a 的集合。

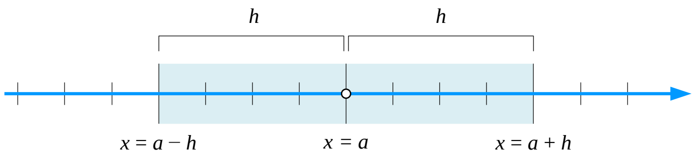


图 4. 邻域

以下代码计算极限并绘制图 3。

```
# Bk3_Ch15_01

from sympy import latex, lambdify, limit, log, oo
from sympy.abc import x
import numpy as np
from matplotlib import pyplot as plt

f_x = (1 + 1/x)**x

x_array = np.linspace(0.1,1000,1000)

f_x_fcn = lambdify(x,f_x)
f_x_array = f_x_fcn(x_array)

f_x_oo_limit = limit(f_x,x,oo)

# visualization

plt.close('all')

fig, ax = plt.subplots()

ax.plot(x_array, f_x_array, linewidth = 1.5)
ax.axhline(y = f_x_oo_limit, color = 'r')

ax.grid(linestyle='--', linewidth=0.25, color=[0.5,0.5,0.5])
ax.set_xlim(x_array.min(),x_array.max())
plt.xscale("log")
ax.set_xlabel('$\it{x}$',fontname = 'Times New Roman')
ax.set_ylabel('$%s$' % latex(f_x), fontname = 'Times New Roman')

plt.grid(True, which="both", ls="-")
```

15.3 左极限、右极限

请注意 (1) 的绝对值符号；如图 5 所示，这代表着 x 从左、右两侧趋向 a。

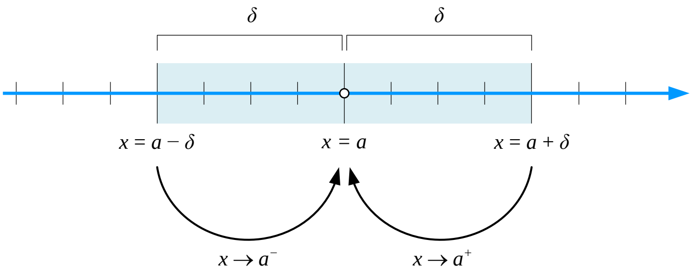


图 5. X 分别从左右两侧趋向 a

右极限

将 (1) 绝对值符号去掉取正得到，

$$0 < x - a < \delta \tag{6}$$

称之为 x 从右侧趋向 a，记做 $x \to a^+$ 。

随之，将 (3) 中极限条件改为 $x \to a^+$ ，C 叫做函数 $f(x)$ 的右极限 (right-hand limit 或 right limit)，记做：

$$\lim_{x \to a^+} f(x) = C \tag{7}$$

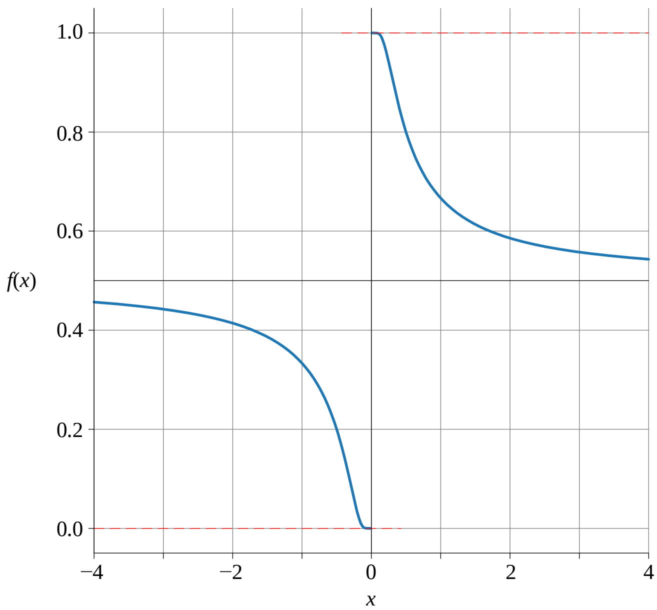


图 6. 函数 $f(x)$ 左右极限不同

左极限

相反，如果将 (1) 绝对值符号去掉并取负，

$$-\delta < x - a < 0 \tag{8}$$

称之为 x 从左侧趋向 a，记做 $x \to a^-$。

将 (3) 中极限条件改为 $x \to a^-$，C 叫做函数 $f(x)$ 的左极限 (left-hand limit 或 left limit)，记做：

$$\lim_{x \to a^-} f(x) = C \tag{9}$$

当 (7) 和 (9) 都成立时，(3) 才成立。也就是，当 $x \to a$ 时函数 $f(x)$ 极限存在的充分必要条件是，左右极限均存在且相等。

极限不存在

请大家格外注意，即便左右极限均存在，如果两者不相等，则极限不存在。

如图 6 所示，函数在 $x = 0$ 的右极限为 1：

$$\lim_{x \to 0^+} \frac{1}{1 + 2^{-1/x}} = 1 \tag{10}$$

而函数在 $x = 0$ 的左极限为 0：

$$\lim_{x \to 0^-} \frac{1}{1 + 2^{-1/x}} = 0 \tag{11}$$

显然函数在 $x = 0$ 处不存在极限；此外，$f(x)$ 在 $x = 0$ 处没有定义。

$\lim_{x \to a} f(x)$ 不存在可能有三种情况：(a) $f(x)$ 在 $x = a$ 处左右极限不一致；(b) $f(x)$在 $x = a$ 处趋向无穷；(c) $f(x)$ 在趋向 $x = a$ 时在两个定值之间震荡。这三种情况分别对应图 7 三幅子图。

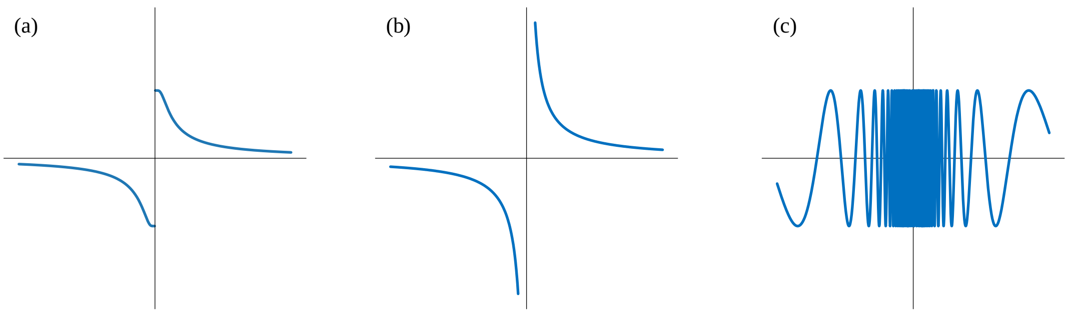


图 7. 极限不存在的三种情况

表 1. 极限的英文表达

数学表达	英文表达
$\lim_{\Delta x \to 0} f(x) = b$	As delta x approaches 0, the limit for f of x equals b.

$\Delta x \to 0$	Delta *x* approaches zero.
$\Delta x \to 0^{+}$	Delta *x* goes to zero from the right. Delta *x* approaches to zero from the right.
$\Delta x \to 0^{-}$	Delta *x* goes to zero from the left. Delta *x* approaches to zero from the left.
$\lim\limits_{\Delta x \to 0}$	The limit as delta *x* approaches zero. The limit as delta *x* tends to zero.
$\lim\limits_{x \to a^{+}}$	The limit as *x* approaches *a* from the right. The limit as *x* approaches *a* from the above.
$\lim\limits_{x \to a^{-}}$	The limit as *x* approaches *a* from the left. The limit as *x* approaches *a* from the below.
$\lim\limits_{x \to c} f(x) = L$	The limit of *f*(*x*) as *x* approaches *c* is *L*.
$\lim\limits_{n \to \infty} a_n = L$	The limit of *a* sub *n* as n approaches infinity equals *L*.
$\lim\limits_{x \to -\infty} f(x) = L_1$	the limit of *f* of *x* as *x* approaches negative infinity is capital *L* sub one.
$\lim\limits_{x \to +\infty} f(x) = L_2$	the limit of f of *x* as *x* approaches positive infinity is capital *L* sub two.

以下代码求函数左右极限，并绘制图 6。

```
# Bk3_Ch15_02

from sympy import latex, lambdify, limit, log, oo
from sympy.abc import x
import numpy as np
from matplotlib import pyplot as plt

f_x = 1/(1 + 2**(-1/x))

f_x_fcn = lambdify(x,f_x)

# right limit
x_array_right = np.linspace(0.01,4,500)
f_x_array_right = f_x_fcn(x_array_right)
f_x_0_limit_right = limit(f_x,x,0,'+')

# left limit
x_array_left = np.linspace(-4,-0.01,500)
f_x_array_left = f_x_fcn(x_array_left)
f_x_0_limit_left = limit(f_x,x,0,'-')

# visualization

plt.close('all')

fig, ax = plt.subplots()

ax.plot(x_array_right, f_x_array_right, linewidth = 1.5, color = 'b')
ax.axhline(y = f_x_0_limit_right, color = 'r')

ax.plot(x_array_left, f_x_array_left, linewidth = 1.5, color = 'b')
ax.axhline(y = f_x_0_limit_left, color = 'r')

ax.axvline(x = 0,   color = 'k')
ax.axhline(y = 0.5, color = 'k')

ax.grid(linestyle='--', linewidth=0.25, color=[0.5,0.5,0.5])
ax.set_xlim(x_array_left.min(),x_array_right.max())

ax.set_xlabel('$\it{x}$',fontname = 'Times New Roman')
ax.set_ylabel('$%s$' % latex(f_x), fontname = 'Times New Roman')
```

15.4 几何视角看导数：切线斜率

导数 (derivative) 描述函数在某一点处的变化率。几何角度来看，导数可以视作曲线切线斜率。

切线斜率

举个中学物理中的例子，加速度 a 是速度 V 的变化率，速度 V 是距离 S 的变化率。

如图 8 所示，匀速直线运动中，距离函数 $S(t)$ 对于时间 t 是一个一次函数；从图像角度，$S(t)$ 是一条斜线。

$S(t)$ 图像的切线斜率不随时间变化，也就是说匀速直线运动的速度函数 $V(t)$ 的图像为常数函数。

而 $V(t)$ 的切线斜率为 0，说明加速度 $a(t)$ 图像为取值为 0 的常数函数。

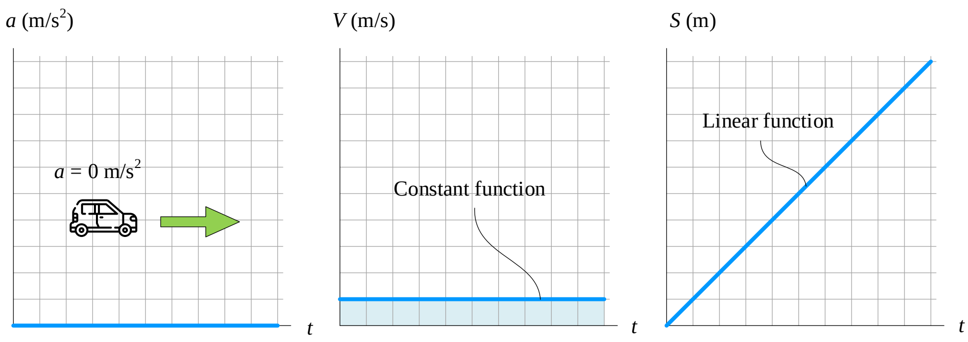


图 8. 匀速直线运动：加速度、速度、距离图像

再看个例子；如图 9 所示，对于匀加速直线运动，距离函数 $S(t)$ 对于时间 t 是一个二次函数；从图像上看，$S(t)$ 在不同时间 t 位置，切线斜率不同。

完成本章学习后，大家会知道二次函数的导数是一次函数，也就是说速度函数 $V(t)$ 的图像为一次函数。

显然，速度函数 $V(t)$ 的切线斜率不随时间变化；因此，$a(t)$ 图像为常数函数，即加速度为定值。

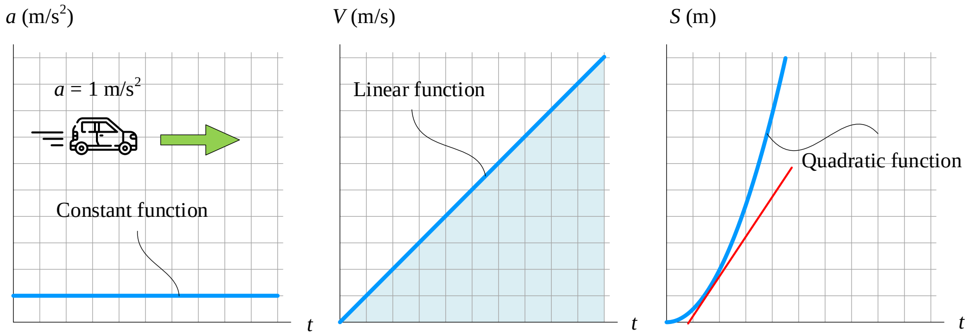


图 9. 匀加速直线运动：加速度、速度、距离

函数导数定义

下面看一下函数导数的确切定义。

对于函数 $y = f(x)$ 自变量 x 在 a 点处一个微小增量 Δx，会导致函数值增量 $\Delta y = f(a + \Delta x) - f(a)$。

当 Δx 趋向于 0 时，函数值增量 Δy 和自变量增量 Δx 比值的极限存在，则称 $y = f(x)$ 在 a 处可导 (function f of x is differentiable at a)；这个极限值便是函数 $f(x)$ 在 a 点处一阶导数值。

$$f'(a) = f'(x)\big|_{x=a} = \frac{\mathrm{d}\, f(x)}{\mathrm{d}\, x}\Big|_{x=a} = \lim_{\Delta x \to 0} \frac{f(a+\Delta x) - f(a)}{\Delta x} \tag{12}$$

如图 10 所示，从几何角度看，随着 Δx 不断减小，割线不断接近切线。

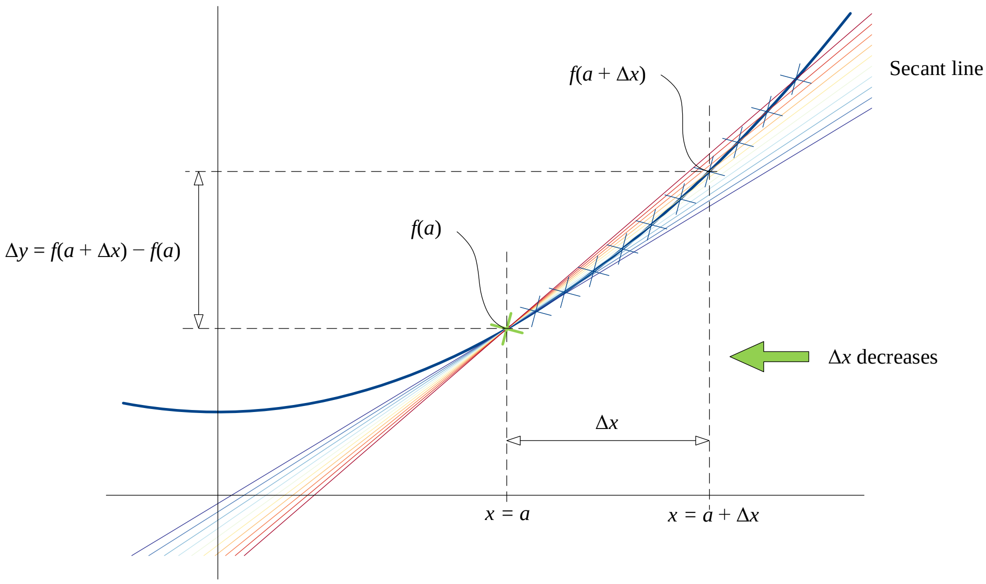


图 10. 导数就是变化率

Δ 和 d 都是“差” (difference) 的含义。但是，代表 Δ 近似值，比如 $\Delta x \to 0$；而 d 是精确值，比如 dx。也就是说，dx 是 Δx 趋向于 0 的精确值。

如果函数 $y = f(x)$ 在 $x = a$ 可导 (differentiable) 则函数在该点处连续 (continuous)；但是，函数在某一点处连续并不意味着函数可导，如图 11 所示两种情况。

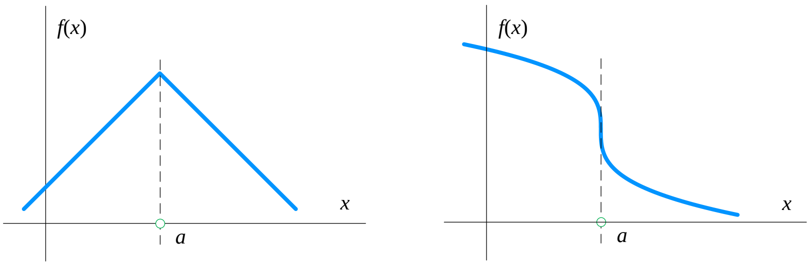


图 11. 函数在 $x = a$ 不存在导数的两种情况

再次注意，本书用 x_1、x_2、x_3 等等表达变量，而不是变量 x 取值。如果有必要对自变量取值进行编号，本书会使用上标记法 $x^{(1)}$、$x^{(2)}$、$x^{(3)}$ 等等。

以下代码绘制图 10。

```
# Bk3_Ch15_03

import numpy as np
import matplotlib.pyplot as plt
from matplotlib import cm # Colormaps
from sympy import latex, lambdify

def plot_secant(x0, y0, x1, y1, color):

    k = (y1 - y0)/(x1 - x0)

    x = np.linspace(-1,4,100)

    secant_y_x = k*(x - x0) + y0

    plt.plot(x, secant_y_x, color = color,
             linewidth = 0.25)

delta_Xs = np.linspace(0.1, 1, 10)

from sympy.abc import x

f_x = x**2

x_array = np.linspace(-1,4,100)

f_x_fcn = lambdify(x,f_x)
y_array = f_x_fcn(x_array)

x0 = 1
y0 = f_x_fcn(x0)

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

```
plt.plot(x_array, y_array, color = '#00448A',
         linewidth = 1.25)
plt.plot(x0, y0, color = '#92D050', marker = 'x', markersize = 12)

colors = plt.cm.RdYlBu_r(np.linspace(0,1,len(delta_Xs)))

for i in np.linspace(1,len(delta_Xs),len(delta_Xs)):

    x1 = x0 + delta_Xs[int(i)-1]
    y1 = f_x_fcn(x1)
    plt.plot(x1, y1, color = '#00448A',
             marker = 'x', markersize = 12)

    plot_secant(x0, y0, x1, y1, colors[int(i)-1])

plt.xlabel('X'); plt.ylabel('$y = f(x)$')
ax.set_title('$f(x) = %s$' % latex(f_x))
ax.set_xlim(0, 2); ax.set_ylim(-1, 4)

fig, ax = plt.subplots()

plt.plot(x0, y0, color = '#92D050', marker = 'x', markersize = 12)

colors = plt.cm.RdYlBu_r(np.linspace(0,1,len(delta_Xs)))

for i in np.linspace(1,len(delta_Xs),len(delta_Xs)):

    x1 = x0 + delta_Xs[int(i)-1]
    y1 = f_x_fcn(x1)

    k = (y1 - y0)/(x1 - x0)

    plt.plot(delta_Xs[int(i)-1], k, color = colors[int(i)-1],
             marker = 'x', markersize = 12)

plt.xlabel('$\Delta$X'); plt.ylabel('$k$')
ax.set_xlim(0, 1); ax.set_ylim(2, 3)
```

15.5 导数也是函数

导数也常被称作导数函数或导函数，因为导数也是函数。

图 12 所示函数曲线在不同点处切线斜率随着自变量 x 变化。函数 $f(x)$ 对自变量 x 的一阶导数 $f'(x)$ 也是一个函数，它的自变量也是 x。$f'(x)$可以读作 (f prime of x)。

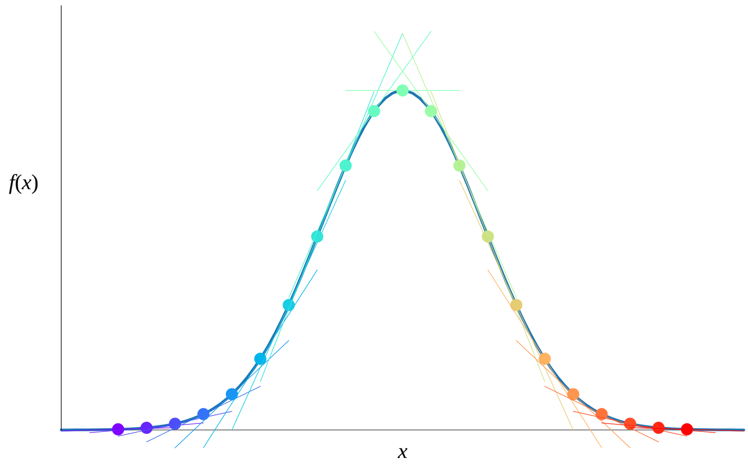


图 12. 函数不同点处切线斜率随着自变量 x 变化

一阶导数

给定二次函数，$f(x) = x^2$；下面利用 (12) 推导它的一阶导数。

$$
\begin{aligned}
f'(x) &= \lim_{\Delta x \to 0} \frac{f(x+\Delta x) - f(x)}{\Delta x} = \lim_{\Delta x \to 0} \frac{(x+\Delta x)^2 - x^2}{\Delta x} \\
&= \lim_{\Delta x \to 0} \frac{2x\Delta x + \Delta x^2}{\Delta x} \\
&= \lim_{\Delta x \to 0} 2x + \underbrace{\Delta x}_{\to 0} = 2x
\end{aligned}
\tag{13}
$$

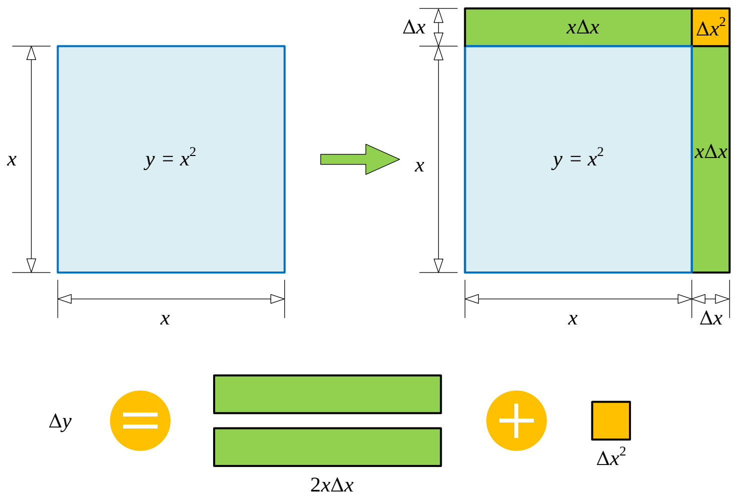


图 13. 几何角度推导 $f(x) = x^2$ 的一阶导数

几何角度来看，$f(x) = x^2$ 相当于边长为 x 的正方形面积。图 13 所示为当 x 增加到 $x + \Delta x$ 时，函数值变化对应正方形面积变化。x 到 $x + \Delta x$，正方形面积增加 $2x\Delta x + \Delta x^2$。

根据导数定义，函数导数为比值 $(2x\Delta x + \Delta x^2)/\Delta x = 2x + \Delta x$；当 $\Delta x \to 0$ 时，可以消去 Δx 一项。

举个例子，图 14 (a) 所示函数如下。

$$f(x) = x^2 + 2 \tag{14}$$

根据前文推导，它的一阶导数解析式如下。

$$f'(x) = 2x \tag{15}$$

如图 14 (b) 所示，(14) 这个二次函数的一阶导数图像为一条斜线。

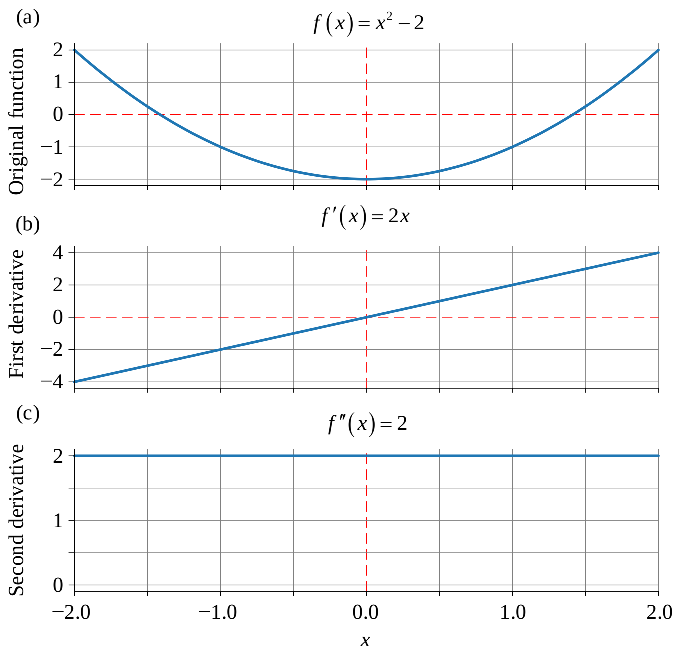


图 14. 二次函数、一阶导数、二阶导数

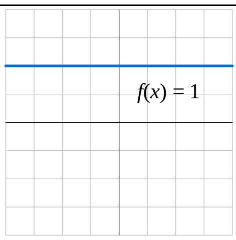
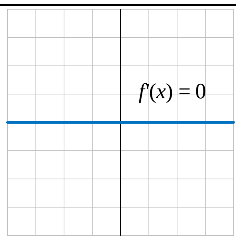
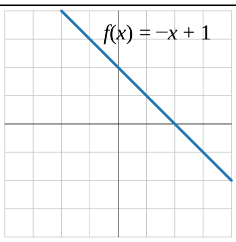
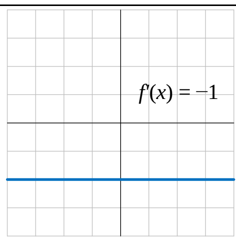
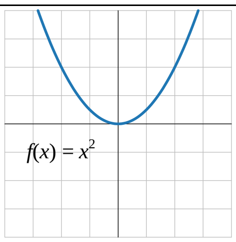
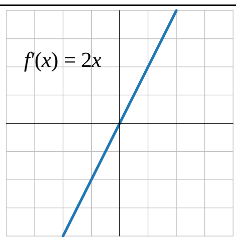
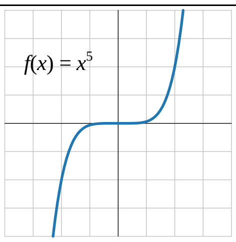
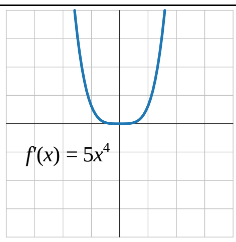
$x < 0$ 时，x 增大 $f(x)$ 减小，此时函数导数为负；$x > 0$，x 增大 $f(x)$ 增大，函数导数为正。值得注意的是 $x = 0$，$f(x)$ 取得**最小值** (minimum)，此处函数 $f(x)$ 导数值为 0。

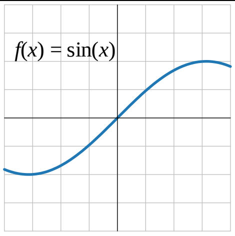
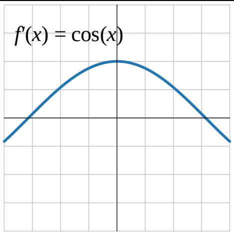
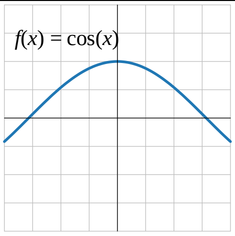
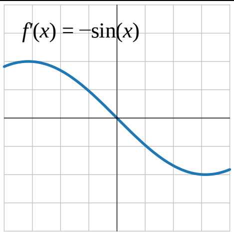
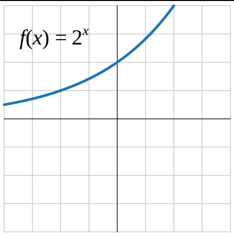
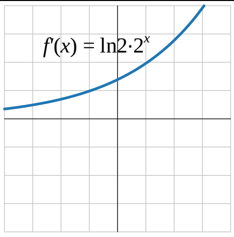
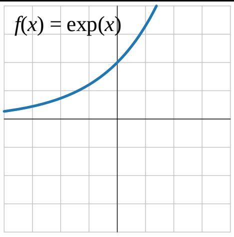
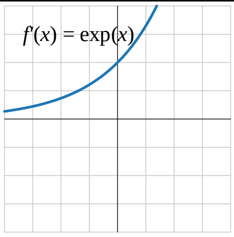
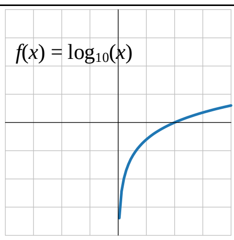
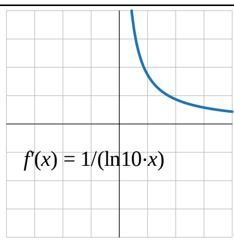
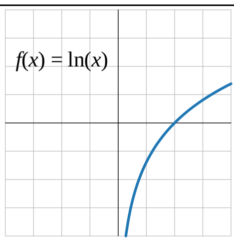
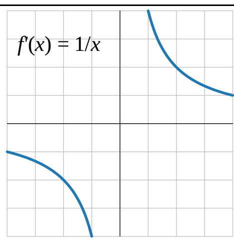
类似地，推导 $f(x) = x^n$ 的导数，n 为大于 1 的正整数。

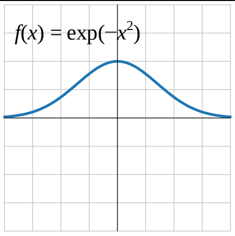
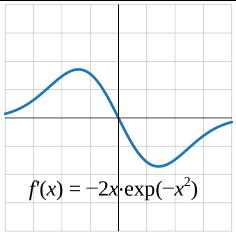
$$
\begin{aligned}
f'(x) &= \lim_{\Delta x \to 0} \frac{f(x+\Delta x) - f(x)}{\Delta x} = \lim_{\Delta x \to 0} \frac{(x+\Delta x)^n - x^n}{\Delta x} \\
&= \lim_{\Delta x \to 0} \frac{x^n + n x^{n-1}\Delta x + \dfrac{n(n-1)}{2} x^{n-2}\Delta x^2 + \cdots \Delta x^n - x^n}{\Delta x} \\
&= \lim_{\Delta x \to 0} n x^{n-1} + \underbrace{\frac{n(n-1)}{2} x^{n-2}\Delta x + \cdots \Delta x^{n-1}}_{\to 0} = n x^{n-1}
\end{aligned} \tag{16}
$$

表 2 总结了常用函数导数及图像，请大家自行绘制这些图像。

表 2. 常用函数导数及图像

函数	函数图像举例	一阶导数	一阶导数图像举例
常数函数 $f(x)=C$		$f'(x)=0$	
一次函数 $f(x)=ax$		$f'(x)=a$	
二次函数 $f(x)=ax^2+bx+c$		$f'(x)=2ax+b$	
幂函数 $f(x)=x^p$		$f'(x)=px^{p-1}$	

正弦函数 $f(x)=\sin x$		$f'(x)=\cos x$	
余弦函数 $f(x)=\cos x$		$f'(x)=-\sin x$	
指数函数 $f(x)=b^x$ $(b>0,b\neq 1)$		$f'(x)=\ln b\cdot b^x$	
自然指数函数 $f(x)=\mathrm{e}^x=\exp(x)$		$f'(x)=\mathrm{e}^x=\exp(x)$	
对数函数 $f(x)=\log_b x$ $(x>0,b>0,b\neq 1)$		$f'(x)=\dfrac{1}{\ln b\cdot x}$	
自然对数函数 $f(x)=\ln x$ $(x>0)$		$f'(x)=\dfrac{1}{x}$	

高斯函数 $f(x)=\exp\left(-\gamma x^2\right)$		$f'(x)=-2\gamma x\exp\left(-\gamma x^2\right)$	

二阶导数

(14) 这个二次函数的二阶导数是其一阶导数的一阶导数，

$$f''(x)=2 \tag{17}$$

如图 14 (c) 所示，(14) 这个二次函数的二阶导数图像为一条水平线，即常数函数。

图 15 所示为，高斯函数以及其一阶导数和二阶导数函数图像。容易发现，函数 $f(x)$ 在 $x = 0$ 处取得最大值，对应的一阶导数为 0，二阶导数为负。这一点对于求解一元函数的极值非常重要，本书后续将介绍。

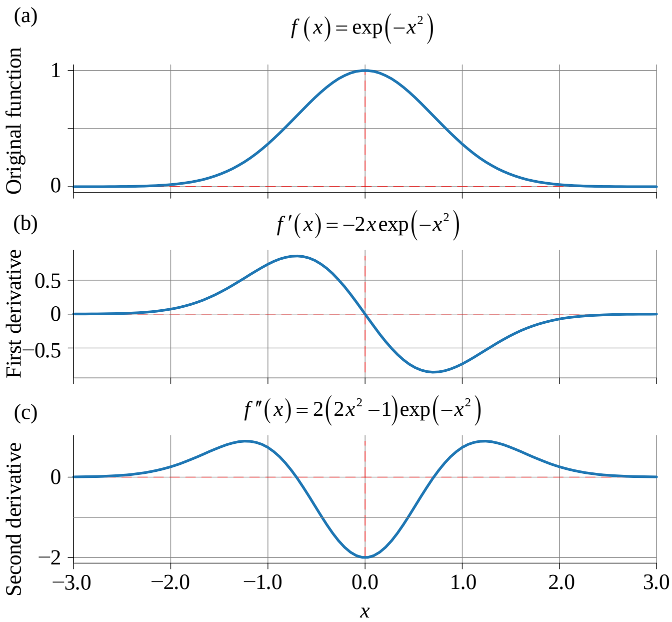


图 15. 高斯函数、一阶导数、二阶导数

图 16 所示为，三次函数图像，以及其一阶导数和二阶导数函数图像。容易发现，$x = 0$ 处函数一阶导数为 0；但是，$x = 0$ 既对应函数的最大值，也不是最小值。

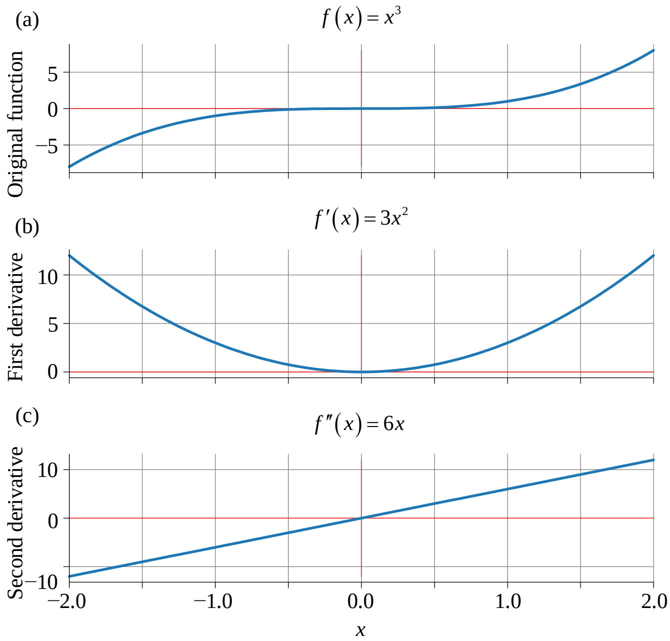


图 16. 三次函数、一阶导数、二阶导数

驻点

有了以上分析，我们可以聊一聊驻点这个概念。

对于一元函数 $f(x)$，驻点 (stationary point) 是函数一阶导数为 0 的点；从图像上来看，一元函数 $f(x)$ 在驻点处的切线平行于 x 轴。

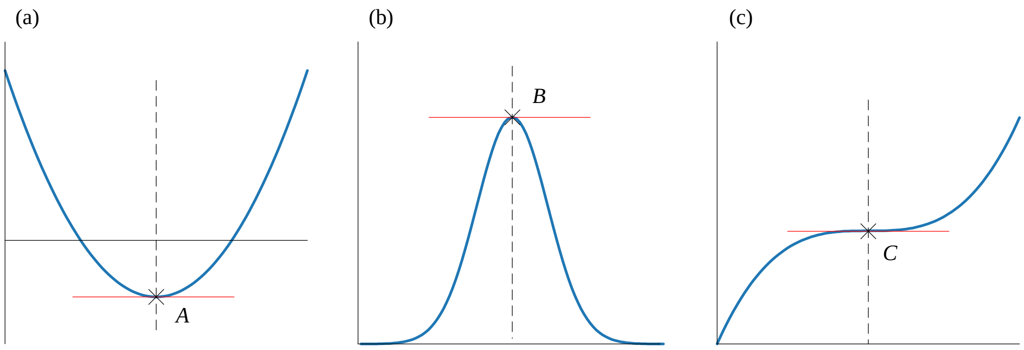


图 17. 驻点可能是极小值、极大值或鞍点

如图 17 所示，驻点可能是一元函数的极小值、极大值，或者是鞍点。注意，这里我们没有用最大值和最小值，这是因为函数可能存在不止一个“山峰”或“山谷”。本书后续将在讲解优化问题时深入探讨这些概念。

表 3. 常用导数法则

和	$\left(f(x)+g(x)\right)' = f'(x)+g'(x)$
差	$\left(f(x)-g(x)\right)' = f'(x)-g'(x)$
积	$\left(f(x)\cdot g(x)\right)' = f'(x)\cdot g(x)+f(x)\cdot g'(x)$
商	$\left(\frac{f(x)}{g(x)}\right)' = \frac{f'(x)\cdot g(x)-f(x)\cdot g'(x)}{g^2(x)}$
倒数	$\left(\frac{1}{f(x)}\right)' = \frac{-f'(x)}{f^2(x)}$

表 4. 导数相关的英文表达

数学表达	英文表达
$\mathrm{d}y$	d *y* differential of *y*
$\frac{\mathrm{d}y}{\mathrm{d}x}$	the derivative of *y* with respect to *x* the derivative with respect to *x* of *y* d *y* by d *x* d *y* over d *x*
$\mathrm{d}f(x)$	The derivative of *f* of *x*
$\frac{\mathrm{d}f(x)}{\mathrm{d}x}$	The derivative of *f* of *x* with respect to *x*
$\frac{\mathrm{d}f(a)}{\mathrm{d}x}$	the derivative of *f* with respect to *x* at *a* d *y* by d *x* at *a* d *y* over d *x* at *a*
$\frac{\mathrm{d}x^3}{\mathrm{d}x}=3x^2$	The derivative of *x* cubed with respect to *x* equals three *x* squared.
$\frac{\mathrm{d}^2y}{\mathrm{d}x^2}$	d two *y* by d *x* squared the second derivative of *y* with respect to *x*
$\frac{\mathrm{d}^2x^3}{\mathrm{d}x^2}=6x$	The second derivative of *x* cubed with respect to *x* equals to six *x*.
$\frac{\mathrm{d}^ny}{\mathrm{d}x^n}$	*n*th derivative of *y* with respect to *x*
$f'(x)$	*f* dash *x* *f* prime of *x* the derivative of *f* of *x* with respect to *x* the first-order derivative of *f* with respect to *x*
$f'(a)$	*f* prime of *a*
$f''(x)$	*f* double-dash *x* *f* double prime of *x* the second derivative of *f* with respect to *x* the second-order derivative of *f* with respect to *x*
$f'''(x)$	*f* triple prime of *x* *f* triple-dash *x* *f* treble-dash *x* the third derivative of *f* with respect to *x* the third-order derivative of *f* with respect to *x*
$f^{(4)}(x)$	the fourth derivative of *f* with respect to *x* the fourth-order derivative of *f* with respect to *x*
$f^{(n)}(x)$	the *n*th derivative of *f* with respect to *x* the *n*th-order derivative of *f* with respect to *x* *f* to the *n*th prime of *x*
$f'\left(g(x)\right)$	*f* prime of *g* of *x* *f* prime at *g* of *x*

$f'(g(x))g'(x)$	the product of *f* prime of *g* of *x* and g prime of *x*
$(f(x)g(x))'$	the quantity of *f* of *x* times *g* of *x*, that quantity prime
$f'(x)g(x)+f(x)g'(x)$	*f* prime of *x* times *g* of *x*, that product plus *f* of *x* times *g* prime of *x*
$\left(\frac{f(x)}{g(x)}\right)'$	the quantity *f* of *x* over *g* of *x*, that quantity prime
$\frac{f'(x)g(x)-f(x)g'(x)}{g^2(x)}$	the fraction, the numerator is *f* prime of *x* times *g* of *x*, that product minus *f* of *x* times *g* prime of *x*, the denominator is *g* squared of *x*

以下代码绘制图 14；请读者修改代码绘制本节其他图像。本节代码采用 sympy.abc import x 定义符号变量，然后利用 sympy.diff() 计算一阶导数函数符号式；利用 sympy.lambdify() 将符号式转换成函数。

```
# Bk3_Ch15_04

from sympy import latex, lambdify, diff, sin
from sympy.abc import x
import numpy as np
from matplotlib import pyplot as plt

# function
f_x = x**2 - 2

x_array = np.linspace(-2,2,100)

f_x_fcn = lambdify(x,f_x)
f_x_array = f_x_fcn(x_array)

# first order derivative

f_x_1_diff = diff(f_x,x)
print(f_x_1_diff)
f_x_1_diff_fcn = lambdify(x,f_x_1_diff)
f_x_1_diff_array = f_x_1_diff_fcn(x_array)

# second order derivative

f_x_2_diff = diff(f_x,x,2)
print(f_x_2_diff)
f_x_2_diff_fcn = lambdify(x,f_x_2_diff)
f_x_2_diff_array = f_x_2_diff_fcn(x_array)
f_x_2_diff_array = f_x_2_diff_array + x_array*0

#%% plot first-, second-order derivatives as functions

fig, ax = plt.subplots(3,1)

# original function

ax[0].plot(x_array, f_x_array, linewidth = 1.5)
ax[0].hlines(y=0, xmin = x_array.min(), xmax = x_array.max(),
             color='r', linestyle='--')
ax[0].vlines(x=0, ymin = f_x_array.min(), ymax = f_x_array.max(),
             color='r', linestyle='--')
ax[0].set_title('$f(x) = %s$' % latex(f_x))
ax[0].set_ylabel('$f(x)$')
ax[0].set_xlim((x_array.min(),x_array.max()))
ax[0].spines['right'].set_visible(False)
```

```python
ax[0].spines['top'].set_visible(False)
ax[0].set_xticklabels([])
ax[0].grid(linestyle='--', linewidth=0.25, color=[0.5,0.5,0.5])

# first-order derivative

ax[1].plot(x_array, f_x_1_diff_array, linewidth = 1.5)
ax[1].hlines(y=0, xmin = x_array.min(), xmax = x_array.max(),
             color='r', linestyle='--')
ax[1].vlines(x=0,
             ymin = f_x_1_diff_array.min(),
             ymax = f_x_1_diff_array.max(),
             color='r', linestyle='--')

ax[1].set_xlabel("$\it{x}$")
ax[1].set_title('$f\'(x) = %s$' % latex(f_x_1_diff))
ax[1].set_ylabel('$f\'(x)$')
ax[1].grid(linestyle='--', linewidth=0.25, color=[0.5,0.5,0.5])
ax[1].set_xlim((x_array.min(),x_array.max()))
ax[1].spines['right'].set_visible(False)
ax[1].spines['top'].set_visible(False)

# second-order derivative

ax[2].plot(x_array, f_x_2_diff_array, linewidth = 1.5)
ax[2].hlines(y=0, xmin = x_array.min(), xmax = x_array.max(),
             color='r', linestyle='--')
ax[2].vlines(x=0,
             ymin = f_x_2_diff_array.min(),
             ymax = f_x_2_diff_array.max(),
             color='r', linestyle='--')

ax[2].set_xlabel("$\it{x}$")
ax[2].set_title('$f^{(2)}(x) = %s$' % latex(f_x_2_diff))
ax[2].set_ylabel('$f\'(x)$')
ax[2].grid(linestyle='--', linewidth=0.25, color=[0.5,0.5,0.5])
ax[2].set_xlim((x_array.min(),x_array.max()))
ax[2].spines['right'].set_visible(False)
ax[2].spines['top'].set_visible(False)
```

每个天才的诞生都需要时代、社会、思想的土壤。牛顿之所以成为牛顿，是一代代巨匠层层累土的结果。

牛顿开创经典牛顿力学体系，让当时人类思想界的面貌天翻地覆，它是人类文明的划时代的里程碑。但是必须认识到牛顿的力学体系是基于哥白尼、开普勒、伽利略等人知识之上的继承和发展。在牛顿所处的时代，哥白尼的日心说已经深入人心，开普勒提出行星运动三定律，伽利略发现惯性定律和自由落体定律。牛顿之所以能发明微积分，离不开笛卡尔创立的解析几何。

人类知识体系是由一代代学者不断继承发展而丰富壮大的。每一个发现、每一条定理，都是知识体系重要的一环，它们既受深受前辈学者影响，又启迪后世学者。