math

2020 年 10 月 25 日

[1]:
```
from __future__ import print_function, division
import numpy as np
import matplotlib as mpl
import matplotlib.pyplot as plt
%matplotlib inline
#%config InlineBackend.figure_format = 'svg'# 输出图像时格式为 svg 矢量图
#\usepackage{fontspec, xunicode, xltxtra}
#\setmainfont{Microsoft YaHei}
# 加在 tex 文件里支持中文
# 输出选 XeLaTeX
#plt.figure(dpi=300)# 输出大图
```

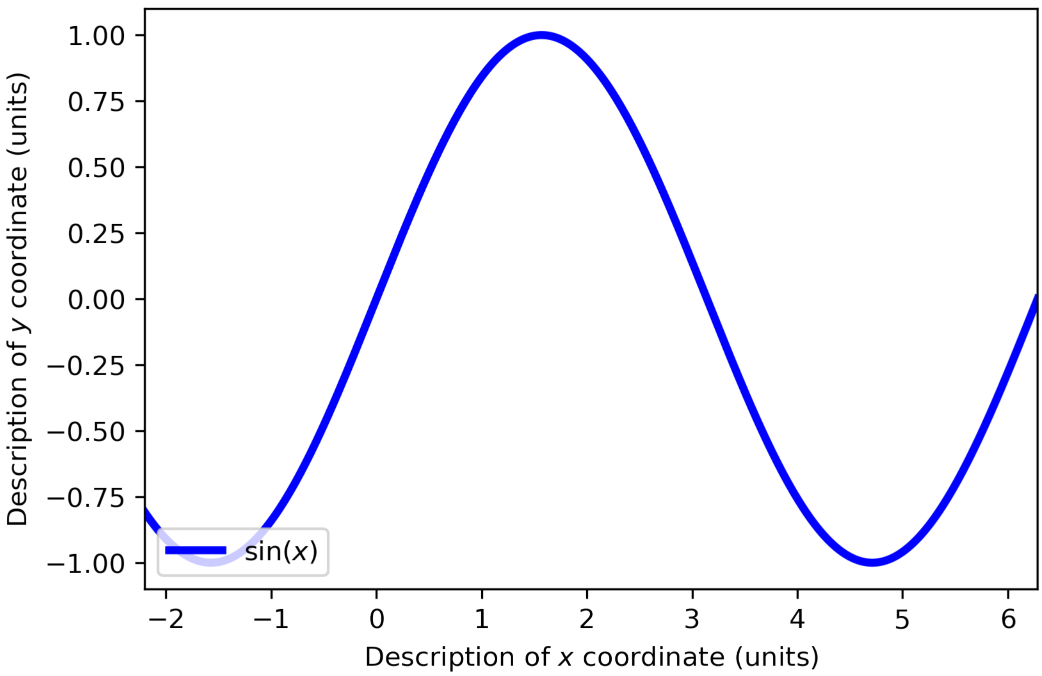
1 函数

1.1 函数的变换

对于函数

$$y = f(x)$$

[25]:
```
# Silly example data
bp_x = np.linspace(-2*np.pi, 2*np.pi, num=500, endpoint=True)
bp_y = np.sin(bp_x)
plt.figure(dpi=300)# 输出大图
# Make the plot
plt.plot(bp_x, bp_y, linewidth=3, linestyle="-",
         color="blue", label=r"$\sin(x)$")
plt.xlabel(r"Description of $x$ coordinate (units)")
plt.ylabel(r"Description of $y$ coordinate (units)")
plt.legend(loc="lower left")
plt.xlim(-0.7*np.pi, 2*np.pi)
plt.ylim(-1.1, 1.1)
plt.show()
```



其上任意一点

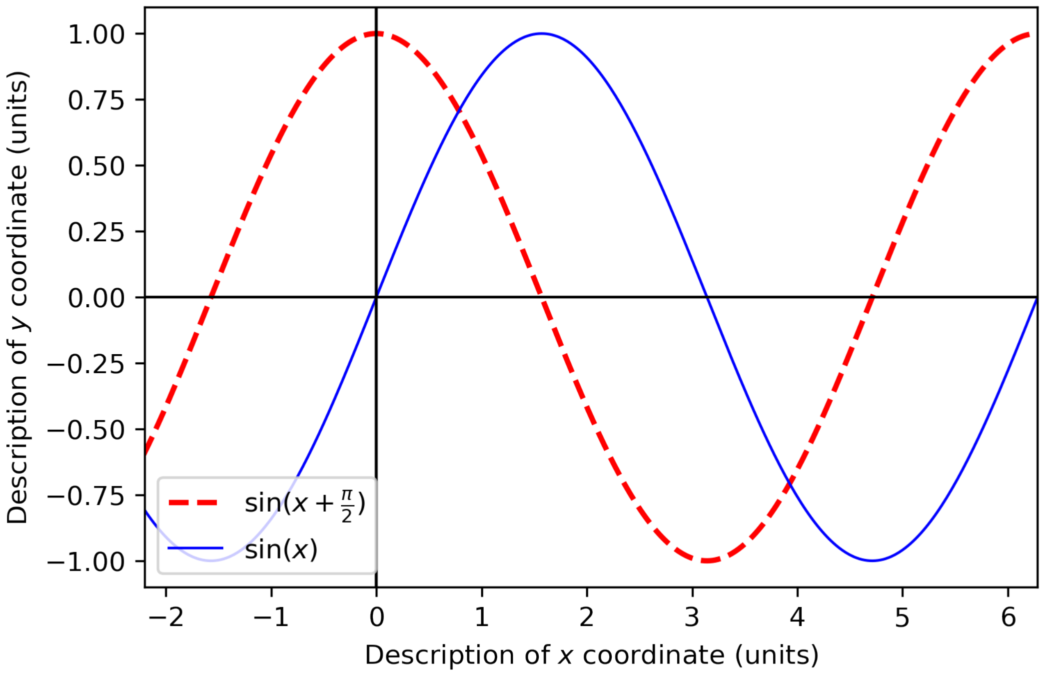
$$(x_0, f(x_0))$$
$$y_0 = f(x_0)$$

首先对其做平移变换，将函数向左平移 $\frac{\pi}{2}$ 个单位得到

$$(x_0 - \frac{\pi}{2}, f(x_0))$$

[40]:
```
# Silly example data
bp_x = np.linspace(-np.pi, 3*np.pi, num=500, endpoint=True)
bp_y = np.sin(bp_x)
plt.figure(dpi=300)# 输出大图
# Make the plot
plt.plot(bp_x-0.5*np.pi, bp_y, linewidth=2, linestyle="--",
         color="red", label=r"$\sin(x+\frac{\pi}{2})$")
plt.plot(bp_x, bp_y, linewidth=1, linestyle="-",
         color="blue", label=r"$\sin(x)$")
plt.plot(bp_x, 0*bp_y, linewidth=1, linestyle="-",
         color="black")
plt.plot(0*bp_x, 2*bp_y, linewidth=1, linestyle="-",
         color="black")
plt.xlabel(r"Description of $x$ coordinate (units)")
plt.ylabel(r"Description of $y$ coordinate (units)")
plt.xlim(-0.7*np.pi, 2*np.pi)
```

```
plt.ylim(-1.1, 1.1)
plt.legend(loc="lower left")
plt.show()
```

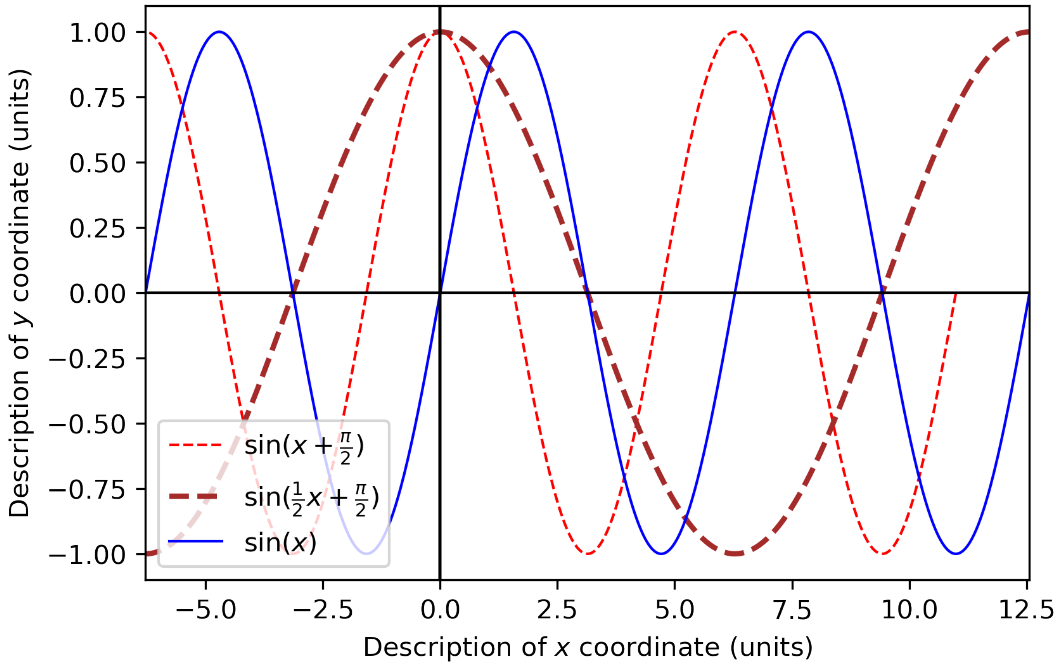


再对其进行伸缩变换，使此刻所有横坐标变为原来的 2 倍得到

$$(2(x_0 - \frac{\pi}{2}), f(x_0))$$

[39]:
```
# Silly example data
bp_x = np.linspace(-2*np.pi, 4*np.pi, num=500, endpoint=True)
bp_y = np.sin(bp_x)
plt.figure(dpi=300)# 输出大图
# Make the plot
plt.plot(bp_x-0.5*np.pi, bp_y, linewidth=1, linestyle="--",
         color="red", label=r"$\sin(x+\frac{\pi}{2})$")
plt.plot(2*bp_x-np.pi, bp_y, linewidth=2, linestyle="--",
         color="brown", label=r"$\sin(\frac{1}{2}x+\frac{\pi}{2})$")
plt.plot(bp_x, bp_y, linewidth=1, linestyle="-",
         color="blue", label=r"$\sin(x)$")
plt.plot(bp_x, 0*bp_y, linewidth=1, linestyle="-",
         color="black")#x-axes
plt.plot(0*bp_x, 2*bp_y, linewidth=1, linestyle="-",
         color="black")#y-axes
```

```python
plt.xlabel(r"Description of $x$ coordinate (units)")
plt.ylabel(r"Description of $y$ coordinate (units)")
plt.xlim(-2*np.pi, 4*np.pi)
plt.ylim(-1.1, 1.1)
plt.legend(loc="lower left")
plt.show()
```

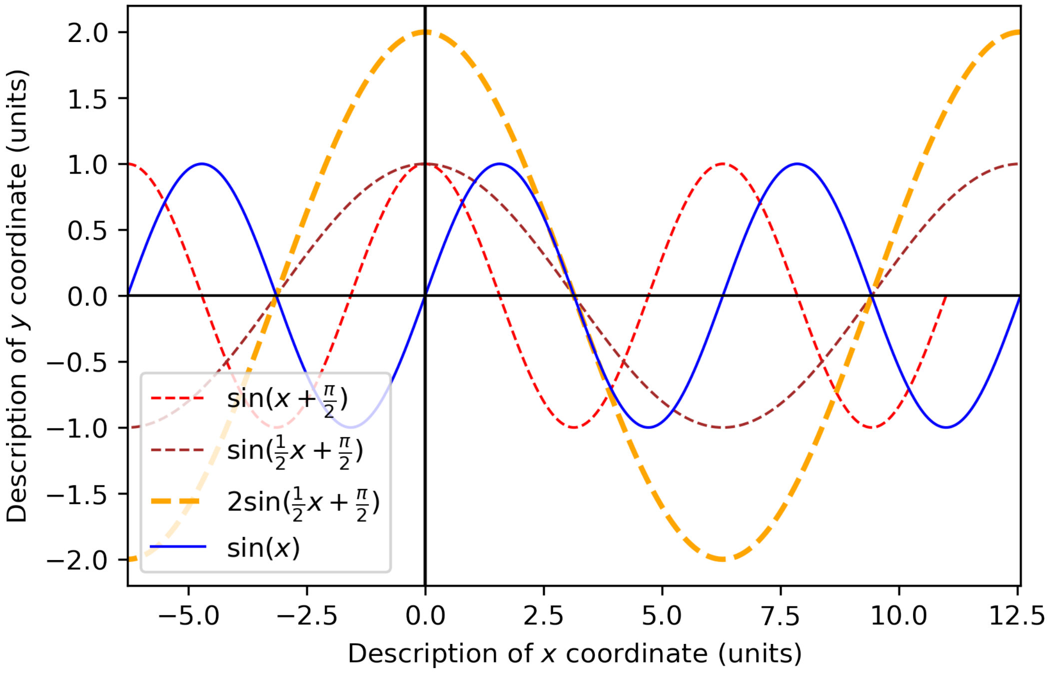


再在竖直方向进行拉伸两倍得到

$$(2(x_0 - \frac{\pi}{2}), 2f(x_0))$$

[48]:
```python
# Silly example data
bp_x = np.linspace(-2*np.pi, 4*np.pi, num=500, endpoint=True)
bp_y = np.sin(bp_x)
plt.figure(dpi=300)# 输出大图
# Make the plot
plt.plot(bp_x-0.5*np.pi, bp_y, linewidth=1, linestyle="--",
         color="red", label=r"$\sin(x+\frac{\pi}{2})$")
plt.plot(2*bp_x-np.pi, bp_y, linewidth=1, linestyle="--",
         color="brown", label=r"$\sin(\frac{1}{2}x+\frac{\pi}{2})$")
plt.plot(2*bp_x-np.pi, 2*bp_y, linewidth=2, linestyle="--",
         color="orange", label=r"$2\sin(\frac{1}{2}x+\frac{\pi}{2})$")
plt.plot(bp_x, bp_y, linewidth=1, linestyle="-",
         color="blue", label=r"$\sin(x)$")
```

```
plt.plot(bp_x, 0*bp_y, linewidth=1, linestyle="-",
         color="black")#x-axes
plt.plot(0*bp_x, 3*bp_y, linewidth=1, linestyle="-",
         color="black")#y-axes
plt.xlabel(r"Description of $x$ coordinate (units)")
plt.ylabel(r"Description of $y$ coordinate (units)")
plt.xlim(-2*np.pi, 4*np.pi)
plt.ylim(-2.2, 2.2)
plt.legend(loc="lower left")
plt.show()
```



最后得到关系式

$$x' = 2x_0 - \pi$$
$$y' = 2y_0$$

其中 $y_0 = f(x_0)$

变换可得

$$x_0 = \tfrac{1}{2}x' + \tfrac{\pi}{2}$$
$$y_0 = \tfrac{1}{2}y'$$

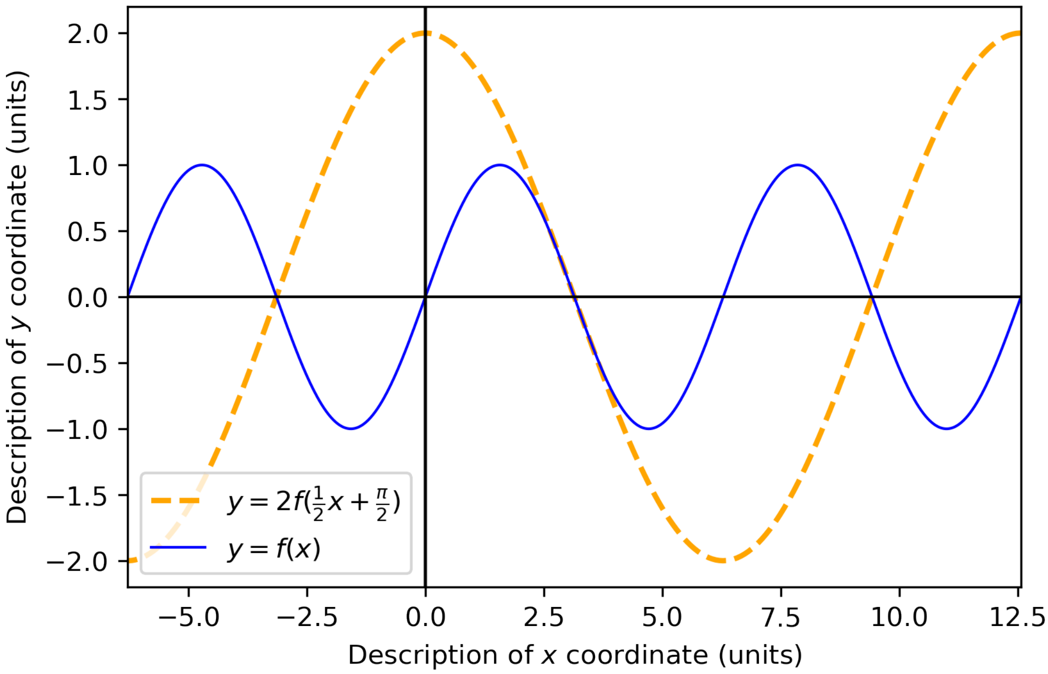
带入可得

$$\frac{1}{2}y' = f(\frac{1}{2}x' + \frac{\pi}{2})$$

容易知道对于任意的

$$(x', y')$$
$$y = 2f(\frac{1}{2}x + \frac{\pi}{2})$$

[53]:
```
# Silly example data
bp_x = np.linspace(-2*np.pi, 4*np.pi, num=500, endpoint=True)
bp_y = np.sin(bp_x)
plt.figure(dpi=300)# 输出大图
# Make the plot
plt.plot(2*bp_x-np.pi, 2*bp_y, linewidth=2, linestyle="--",
         color="orange", label=r"$y=2f(\frac{1}{2}x+\frac{\pi}{2})$")
plt.plot(bp_x, bp_y, linewidth=1, linestyle="-",
         color="blue", label=r"$y = f(x)$")
plt.plot(bp_x, 0*bp_y, linewidth=1, linestyle="-",
         color="black")#x-axes
plt.plot(0*bp_x, 3*bp_y, linewidth=1, linestyle="-",
         color="black")#y-axes
plt.xlabel(r"Description of $x$ coordinate (units)")
plt.ylabel(r"Description of $y$ coordinate (units)")
plt.xlim(-2*np.pi, 4*np.pi)
plt.ylim(-2.2, 2.2)
plt.legend(loc="lower left")
plt.show()
```



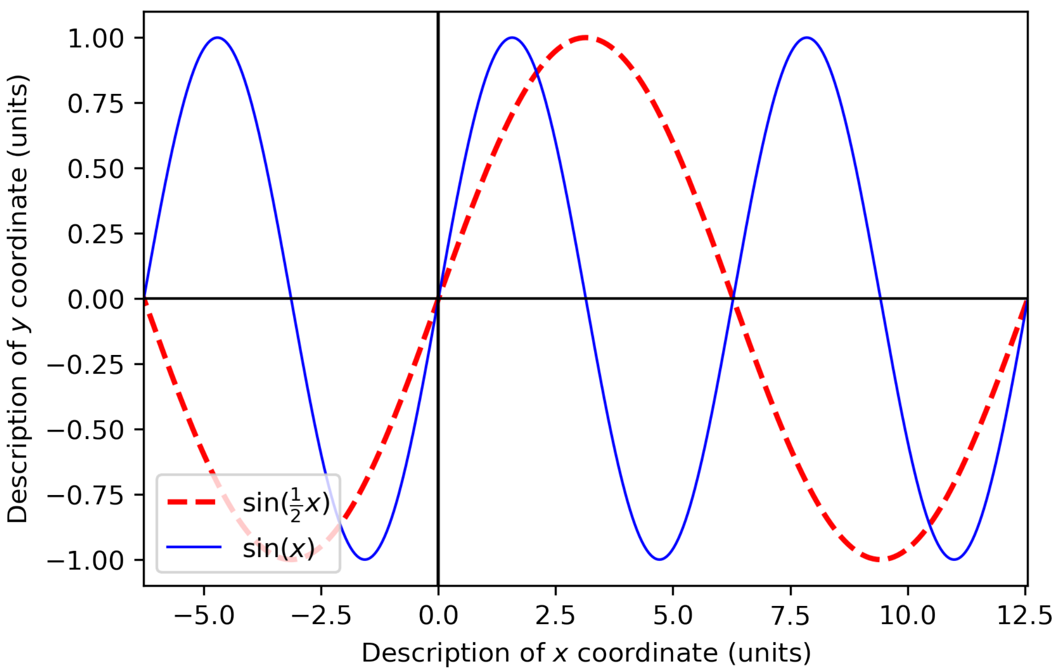
同理，对于

$$y = f(x)$$

先把横坐标拉伸 2 倍

$$(2x_0, y_0)$$

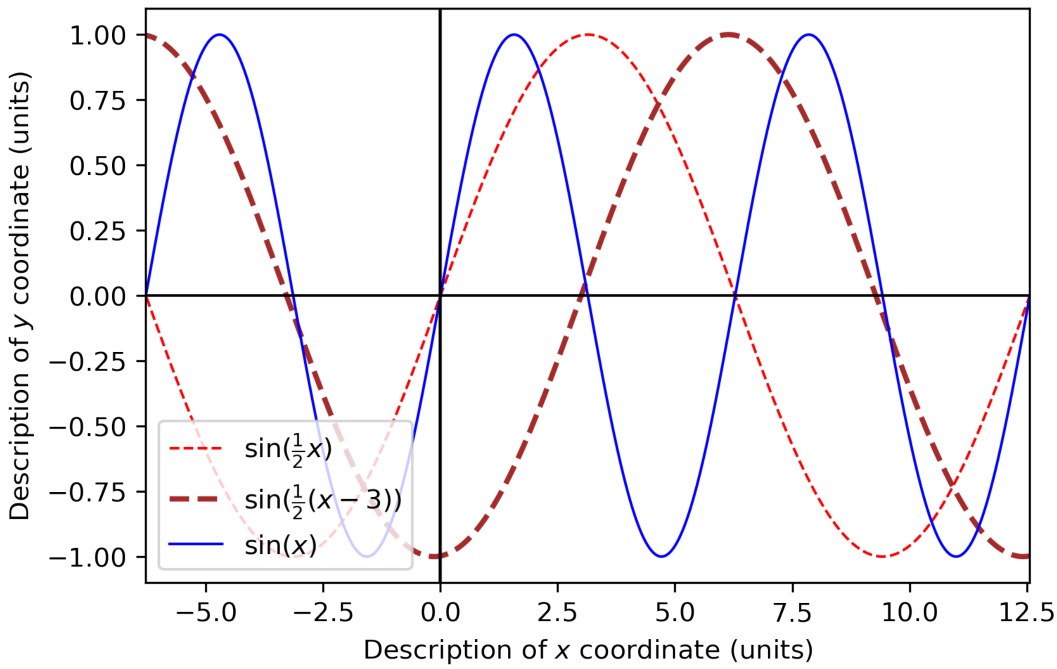
[58]:
```
# Silly example data
bp_x = np.linspace(-2*np.pi, 4*np.pi, num=500, endpoint=True)
bp_y = np.sin(bp_x)
plt.figure(dpi=300)# 输出大图
# Make the plot
plt.plot(2*bp_x, bp_y, linewidth=2, linestyle="--",
         color="red", label=r"$\sin(\frac{1}{2}x)$")
plt.plot(bp_x, bp_y, linewidth=1, linestyle="-",
         color="blue", label=r"$\sin(x)$")
plt.plot(bp_x, 0*bp_y, linewidth=1, linestyle="-",
         color="black")#x-axes
plt.plot(0*bp_x, 3*bp_y, linewidth=1, linestyle="-",
         color="black")#y-axes
plt.xlabel(r"Description of $x$ coordinate (units)")
plt.ylabel(r"Description of $y$ coordinate (units)")
plt.xlim(-2*np.pi, 4*np.pi)
plt.ylim(-1.1, 1.1)
plt.legend(loc="lower left")
plt.show()
```



再右移 3 个单位

$$(2x_0+3, y_0)$$

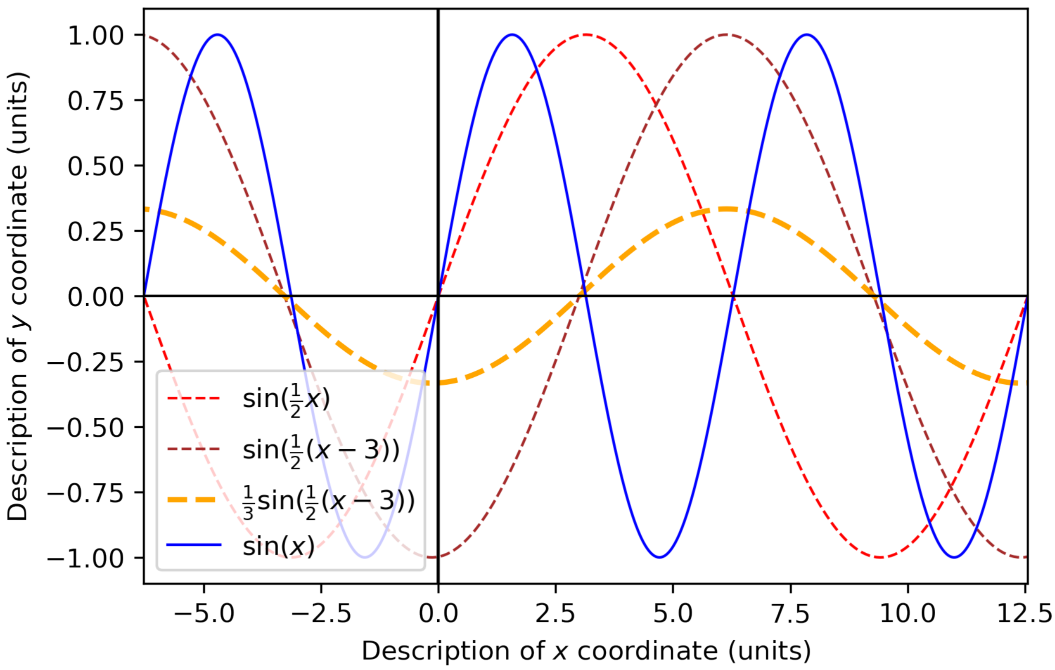
[60]:
```
# Silly example data
bp_x = np.linspace(-2*np.pi, 4*np.pi, num=500, endpoint=True)
bp_y = np.sin(bp_x)
plt.figure(dpi=300)# 输出大图
# Make the plot
plt.plot(2*bp_x, bp_y, linewidth=1, linestyle="--",
         color="red", label=r"$\sin(\frac{1}{2}x)$")
plt.plot(2*bp_x+3, bp_y, linewidth=2, linestyle="--",
         color="brown", label=r"$\sin(\frac{1}{2}(x-3))$")
plt.plot(bp_x, bp_y, linewidth=1, linestyle="-",
         color="blue", label=r"$\sin(x)$")
plt.plot(bp_x, 0*bp_y, linewidth=1, linestyle="-",
         color="black")#x-axes
plt.plot(0*bp_x, 3*bp_y, linewidth=1, linestyle="-",
         color="black")#y-axes
plt.xlabel(r"Description of $x$ coordinate (units)")
plt.ylabel(r"Description of $y$ coordinate (units)")
plt.xlim(-2*np.pi, 4*np.pi)
plt.ylim(-1.1, 1.1)
plt.legend(loc="lower left")
plt.show()
```



再竖着拉伸 $\frac{1}{3}$ 倍

$$(2x_0+3, \frac{1}{3}y_0)$$

[62]:
```
# Silly example data
bp_x = np.linspace(-2*np.pi, 4*np.pi, num=500, endpoint=True)
bp_y = np.sin(bp_x)
plt.figure(dpi=300)# 输出大图
# Make the plot
plt.plot(2*bp_x, bp_y, linewidth=1, linestyle="--",
         color="red", label=r"$\sin(\frac{1}{2}x)$")
plt.plot(2*bp_x+3, bp_y, linewidth=1, linestyle="--",
         color="brown", label=r"$\sin(\frac{1}{2}(x-3))$")
plt.plot(2*bp_x+3, bp_y/3, linewidth=2, linestyle="--",
         color="orange", label=r"$\frac{1}{3}\sin(\frac{1}{2}(x-3))$")
plt.plot(bp_x, bp_y, linewidth=1, linestyle="-",
         color="blue", label=r"$\sin(x)$")
plt.plot(bp_x, 0*bp_y, linewidth=1, linestyle="-",
         color="black")#x-axes
plt.plot(0*bp_x, 3*bp_y, linewidth=1, linestyle="-",
         color="black")#y-axes
plt.xlabel(r"Description of $x$ coordinate (units)")
plt.ylabel(r"Description of $y$ coordinate (units)")
plt.xlim(-2*np.pi, 4*np.pi)
plt.ylim(-1.1, 1.1)
plt.legend(loc="lower left")
plt.show()
```



有对应关系

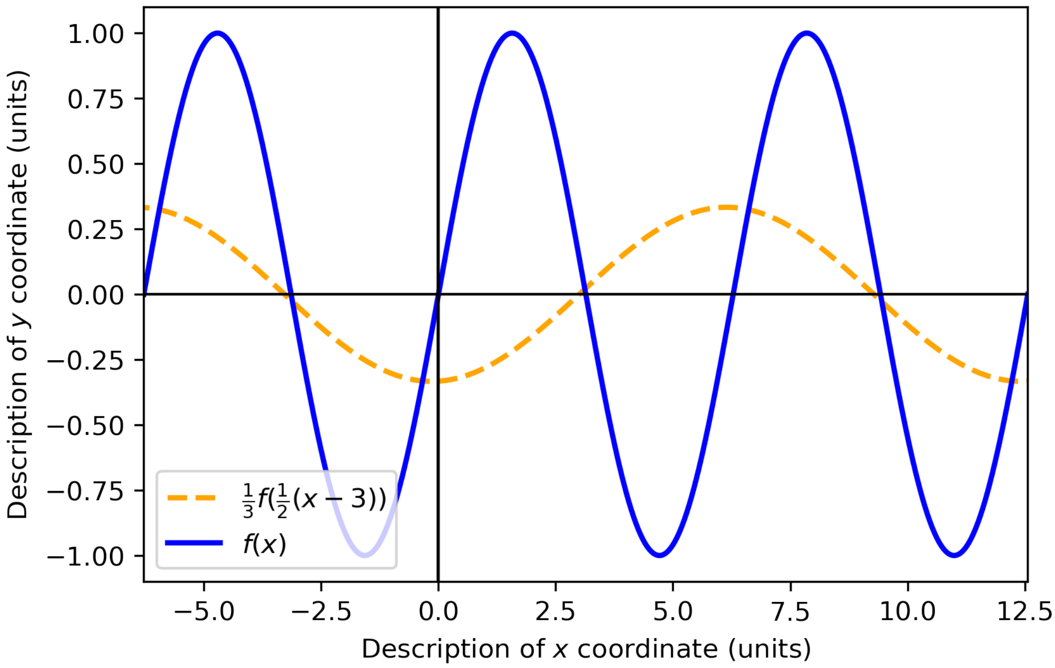
$$x' = 2x_0 + 3$$
$$y' = \frac{1}{3}y_0$$

代入有

$$y' = \frac{1}{3}f(\frac{1}{2}(x'-3))$$

[3]:
```
# Silly example data
bp_x = np.linspace(-2*np.pi, 4*np.pi, num=500, endpoint=True)
bp_y = np.sin(bp_x)
plt.figure(dpi=300)# 输出大图
# Make the plot
plt.plot(2*bp_x+3, bp_y/3, linewidth=2, linestyle="--",
         color="orange", label=r"$\frac{1}{3}f(\frac{1}{2}(x-3))$")
plt.plot(bp_x, bp_y, linewidth=2, linestyle="-",
         color="blue", label=r"$f(x)$")
plt.plot(bp_x, 0*bp_y, linewidth=1, linestyle="-",
         color="black")#x-axes
plt.plot(0*bp_x, 3*bp_y, linewidth=1, linestyle="-",
         color="black")#y-axes
plt.xlabel(r"Description of $x$ coordinate (units)")
plt.ylabel(r"Description of $y$ coordinate (units)")
plt.xlim(-2*np.pi, 4*np.pi)
plt.ylim(-1.1, 1.1)
```

```
plt.legend(loc="lower left")
plt.show()
```



1.2 实例详解

把

$$y = f(\omega x + \phi)$$

向左平移 m 个单位

对于点

$$(x_0, y_0)$$

有映射

$$f : \omega x_0 + \phi \longrightarrow y_0$$

平移后的点为

$$(x_0 - m, y_0)$$

记 $x_0 - m = x_1$ 则有映射

$$f : \omega(x_1 + m) + \phi \longrightarrow y_0$$

即对于变换后的点而言任意给出的 x 与 y 有关系式

$$y = f(\omega(x + m) + \phi)$$

直接有**推论**

$$y=f(\omega x+\phi)=f(\omega(x+\frac{\phi}{\omega}))$$

可以由

$$y=f(\omega x)$$

向左平移 $\frac{\phi}{\omega}$ 个单位得到

同理，把

$$y=f(\omega x+\phi)$$

在水平方向拉伸 k 倍

拉伸后的点为

$$(kx_0,y_0)$$

记 $x_2=kx_0$ 有

$$f:\omega\frac{x_2}{k}+\phi\longrightarrow y_0$$

平移后的函数即为

$$y=f(\omega\frac{x_2}{k}+\phi)$$

1.2.1 推论

在水平方向的伸缩不改变 ϕ

水平方向的平移作用在 ω 的括号里

如何快速

$$y=f(\omega_1x+\phi_1)\longrightarrow y=f(\omega_2x+\phi_2)$$

根据推论首先改 ω 不改变 ϕ

即在水平方向伸缩 $\frac{\omega_1}{\omega_2}$ 倍得到

$$\omega_1x_0=\omega_2x_1$$
$$x_1=\tfrac{\omega_1}{\omega_2}x_0$$

然后再水平平移，我不写了，你懂得

1.3 奇怪的题

锐角三角形 ABC 中，欲证

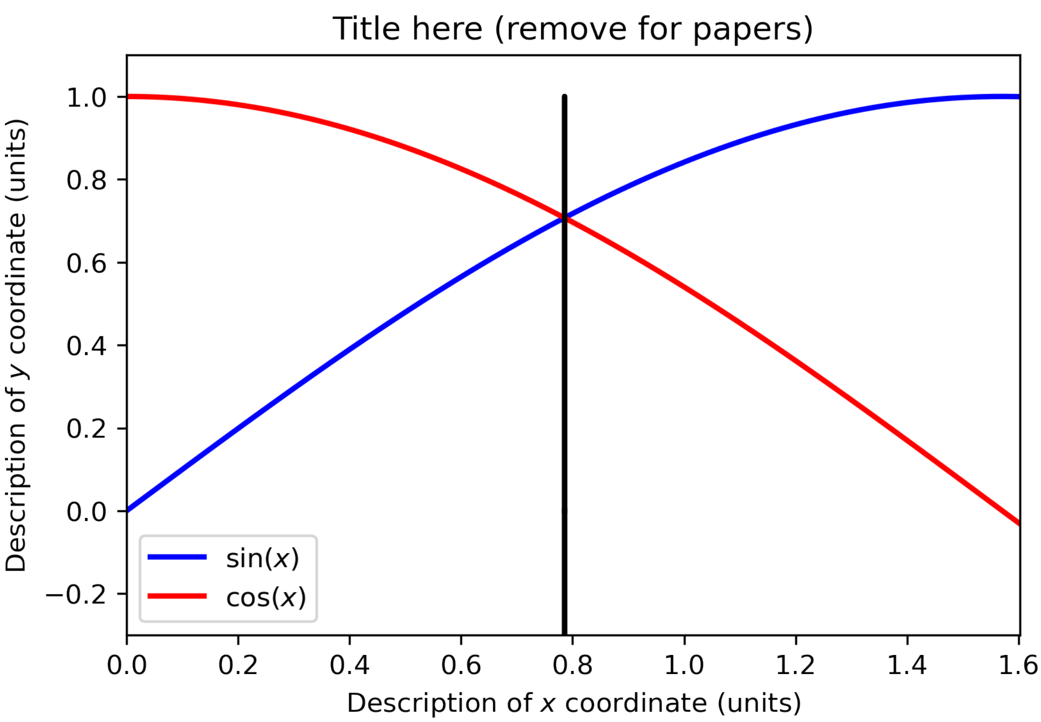
$$sinA+sinB+sinC>cosA+cosB+cosC$$

注意到

$$A+B>\frac{\pi}{2}$$

[17]:
```python
# Silly example data
bp_x = np.linspace(0, 2*np.pi, num=1000, endpoint=True)
bp_y = np.sin(bp_x)
plt.figure(dpi=300)
# Make the plot
plt.plot(bp_x, bp_y, linewidth=2, linestyle="-",
         color="blue", label=r"$\sin(x)$")
plt.plot(bp_x-0.5*np.pi, bp_y, linewidth=2, linestyle="-",
         color="red", label=r"$\cos(x)$")
plt.plot(bp_x*0+0.25*np.pi, bp_y, linewidth=2, linestyle="-",
         color="black")
plt.plot(0.6,np.sin(0.6),linewidth=2, linestyle="-",
         color="red")
plt.xlabel(r"Description of $x$ coordinate (units)")
plt.ylabel(r"Description of $y$ coordinate (units)")
plt.title(r"Title here (remove for papers)")
plt.xlim(0, 0.51*np.pi)
plt.ylim(-0.3,1.1)
plt.legend(loc="lower left")

plt.show()
```



注意到

$$\frac{\pi}{4} - A < B - \frac{\pi}{4}$$

所以有

$$sinA > cosB$$

得证

2 易错点

2.1 注意平凡情况

比如斜率为 0 啊，二次函数首项系数为 0 啊啥的，这一写，不就有分了？先写上再说

导数切线题注意看好，是**在**某点的切线，还是**过**某点的切线

概率题，把事件写上，要不然扣分！

2.2 咱们来看看一个简单的矩阵

$$\begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & \cdots & a_{mn} \end{bmatrix}$$

条件告诉你

$$a_{ik} \neq a_{jk}$$

说的是啥意思呢

说的是同一列中的第 i 个和第 j 个不等！

2.3 有关向量的角度

注意到向量的角度从 0 取到 π，其中，0 不是锐角，π 不是钝角

3 圆锥曲线

3.1 二次曲线系

简单回顾

对于任意二次曲线

$$A_1x^2 + B_1y^2 + C_1xy + D_1x + E_1y + F_1 = 0$$
$$A_2x^2 + B_2y^2 + C_2xy + D_2x + E_2y + F_2 = 0$$

过它们交点的二次曲线可以写成

$$\mu(A_1x^2 + B_1y^2 + C_1xy + D_1x + E_1y + F_1) + \lambda(A_2x^2 + B_2y^2 + C_2xy + D_2x + E_2y + F_2) = 0$$

这时候，只要配配系数就可以解决问题了

比如圆的方程必没有 xy 项，且 x^2 和 y^2 系数相等

3.1.1 实例详解

椭圆

$$Ax^2 + By^2 + F = 0$$

椭圆外一点 E 在定直线 x=C 上运动，与左顶点 A 与右顶点 B 连直线交椭圆于四个点 ACBD

证明 AB 与 CD 交点为定点

只需设出 l_{EA} 和 l_{EB} 的方程并乘在一块再与椭圆方程构建二次曲线系

就可以得到一定过 ABCD 四个点的二次曲线

随后，把 l_{AB} 和 l_{CD} 的直线根据 CD 过定点设出来，调参数即可

3.2 中点弦

中点弦是指，这有一个点，过这个点的直线与二次曲线恰好交于两点，这个点是中点，那么这条直线就能确定了

对于任意二次曲线

$$Ax^2 + By^2 + Cxy + Dx + Ey + F = 0$$

设

$$(x_0, y_0)$$

为那个中点

则有另外两个被分成两段的点

$$(x_1, y_1), (2x_0 - x_1, 2y_0 - y_1)$$

立刻有方程

$$Ax_1^2 + By_1^2 + Cx_1y_1 + Dx_1 + Ey_1 + F = 0 \qquad (1)$$
$$A(2x_0 - x_1)^2 + B(2y_0 - y_1)^2 + C(2x_0 - x_1)(2y_0 - y_1) + D(2x_0 - x_1) + E(2y_0 - y_1) + F = 0 \qquad (2)$$

(2)-(1) 得到

$$A(4x_0^2 - 4x_0x_1) + B(4y_0^2 - 4y_0y_1) + C(4x_0y_0 - 2(x_0y_1 + y_0x_1)) + D(2x_0 - 2x_1) + E(2y_0 - 2y_1) = 0$$

化简得到

$$(2Ax_0 + Cy_0 + D)(x_1 - x_0) + (2By_0 + Cx_0 + E)(y_1 - y_0) = 0$$

看得到

$$(2Ax_0 + Cy_0 + D, 2By_0 + Cx_0 + E)$$

是该直线的法向量

所以中点弦即为

$$(2Ax_0+Cy_0+D)(x-x_0)+(2By_0+Cx_0+E)(y-y_0)=0$$

或者这个形式

$$(2Ax_0+Cy_0+D)x+(2By_0+Cx_0+E)y=(2Ax_0+Cy_0+D)x_0+(2By_0+Cx_0+E)y_0$$

注意到 (x_0, y_0) 关于该曲线的极线为

$$Ax_0x+By_0y+C\frac{x_0y+y_0x}{2}+D\frac{x_0+x}{2}+E\frac{y_0+y}{2}+F=0$$

或者换一种写法

$$(2Ax_0+Cy_0+D)x+(2By_0+Cx_0+E)y=-2F$$

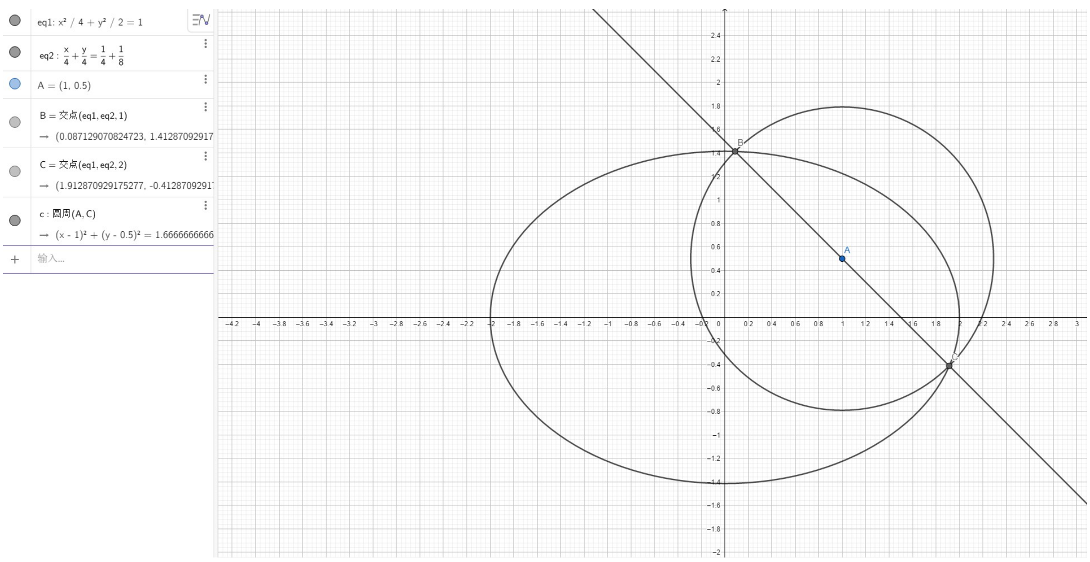
形式相近，方便记忆

3.2.1 结论

求中点弦只要把这个点的极线写出来，左边只放 x 和 y，在右边放 x 和 y 对应前面的系数和 x_0, y_0

举例，椭圆 $Ax^2+By^2+F=0$ 的中点弦是

$$Ax_0x+By_0y=Ax_0^2+By_0^2$$



4 集合

简单的定义

$$\forall a\in A, \exists b=a\in B\Rightarrow A\subseteq B$$

简单的应用

$$if \quad \forall x_1 \in D_1, f(x_1) \in A, \exists x_2 \in D_2, g(x_2) \in B \Rightarrow A \subseteq B$$

其中，A 和 B 分别为 $x \in D_1$ 时 $f(x_1)$ 的值域和 $x \in D_2$ 时 $g(x_2)$ 的值域

5 小技巧

5.1 代数

见到形如

$$xy + \frac{1}{x} + \frac{1}{y} = 1$$

的式子该怎么处理呢？

可以两边乘以 xy 然后当作 x 的二次函数来解

可以利用

$$(a+1)(b+1) = ab + a + b + 1$$

这个结论

换元

$$\begin{cases} xy = a \\ \frac{1}{x} = b \\ \frac{1}{y} = c \end{cases}$$

有

$$\begin{cases} a + b + c = 1 \\ abc = 1 \end{cases}$$

5.2 数论

对于偶数来说，可以表示为 2^n 以及 $2^m \cdot c$ 其中

$$c = 2k + 1 \quad k \in \mathbb{N}^+$$

对于

$$(i+j)(i-j) \quad (i, j \in \mathbb{N}^+)$$

可以表示除了 1，2 之外的所有自然数

而对于

$$(i+j)(i-j+1) \quad (i, j \in \mathbb{N}^+)$$

来说，可以表示除了 2 的幂之外的所有偶数

给定一个不是 2 的幂的偶数，必然能写成

$$2^m \cdot c \quad \text{where} \quad c = 2k + 1 \quad k \in \mathbb{N}^+$$

则只需令

$$
\begin{aligned}
i + j &= 2^m \\
i - j + 1 &= c = 2k + 1
\end{aligned}
$$

就能解出一组 (i,j)

5.3 导数题

对于任意一道导数题，先观察一下函数是否过某定点，与参数无关的，可以尝试验证 ±1,±2,$\pm e$，很可能后面要用到。该技巧适用于一切函数题。

5.4 裂项

对于 $n(n+1)(n+2)(n+3)(n+4)$ 这样的整式，可以

$$n(n+1)(n+2)(n+3)(n+4) = \frac{1}{6}[n(n+1)(n+2)(n+3)(n+4)(n+5) - (n-1)n(n+1)(n+2)(n+3)(n+4)]$$

5.5 最后一题

- 要抽象思考，比如说，你要求的东西只跟集合的基数有关（cardinality），那就没必要关注具体的细节，直接计算 card 就行
- 注意**奇偶分析**会是极其有用的手段。有的时候可以直接用奇数和偶数来代替你要写的具体的项 (mod2)，举个例子 $|a_{n+1} - a_n| = 1$，且 $a_1 = 0$，你发现，写出每一项是不好写的，但是写奇偶是没问题的，奇数项为 1，偶数项为 0，这样的手段往往可以帮你快速判断或证明一个命题为假
- 有时候一定要想想，我正在求的东西我真的在乎吗？我真的在乎里面具体每一个数吗？我大概是不在乎的，抽象点！
- 一个 n 元集合 $A = \{a_1.....a_n\}$ 的所有子集可以用一个 n 位的 2 进制数表示，或者，n 维向量进行表示，用 1 和 0 来代表选或没选该元素，好处是**可以用 *0* 来湮灭我所不需要的东西**。比如我告诉你 A 的不同的两个子集的交集的 card 是个偶数，就可以变成，

  $$\vec{B_i} \cdot \vec{B_j} = 2n \quad n \in \mathbb{N}^+$$

  ，这样我就又可以通过点乘来奇偶分析了！