ch02 sec 2.2 在某个点的导数

```
md"# ch02 sec 2.2  在某个点的导数"
```

Table of Contents

ch02 sec 2.2 在某个点的导数
瞬时变化率: 导数
导数的可视化
代数方法计算导数

```
begin
    using PlutoUI  , Plots  ,DataFrames  ,HypertextLiteral  ,LaTeXStrings
    ,Symbolics
    gr()
    theme(:bright)
    @htl("""<script src="https://cdn.bootcdn.net/ajax/libs/mathjax/3.2.0/es5/tex-svg-full.min.js"></script>""")
     PlutoUI.TableOfContents()

end
```

在上一小节,我们已经见过了瞬时速度的定义:

$$\lim_{h \to 0} \frac{s(a+h)-s(a)}{h}$$

瞬时速度的分子是函数的变化值, 分母是定义域区间的一小段间隔, 商就是函数值变化除以变量变化的幅度.

```
md"""
在上一小节,我们已经见过了瞬时速度的定义:

$$\lim_{h \to 0} \frac{s(a+h)-s(a)}{h}$$

瞬时速度的分子是函数的变化值, 分母是定义域区间的一小段间隔,  商就是函数值变化除以变量变化的幅度.

"""
```

瞬时变化率: 导数

函数$f(x)$在定义域内 a 点的导数定义为:

$$f(a)'=\lim_{h \to 0} \frac{f(a+h)-f(a)}{h}$$

如果a点的导数存在, 就可以说函数$f(x)$ 在 a 点可微分.

```
md"""
## 瞬时变化率: 导数

函数$f(x)$在定义域内 a 点的导数定义为:

$$f{(a)}'=\lim_{h \to 0} \frac{f(a+h)-f(a)}{h}$$

如果a点的导数存在, 就可以说函数$f(x)$ 在 a 点可微分.
"""
```

导数的可视化

下面看一下交互式导数可视化的图形.

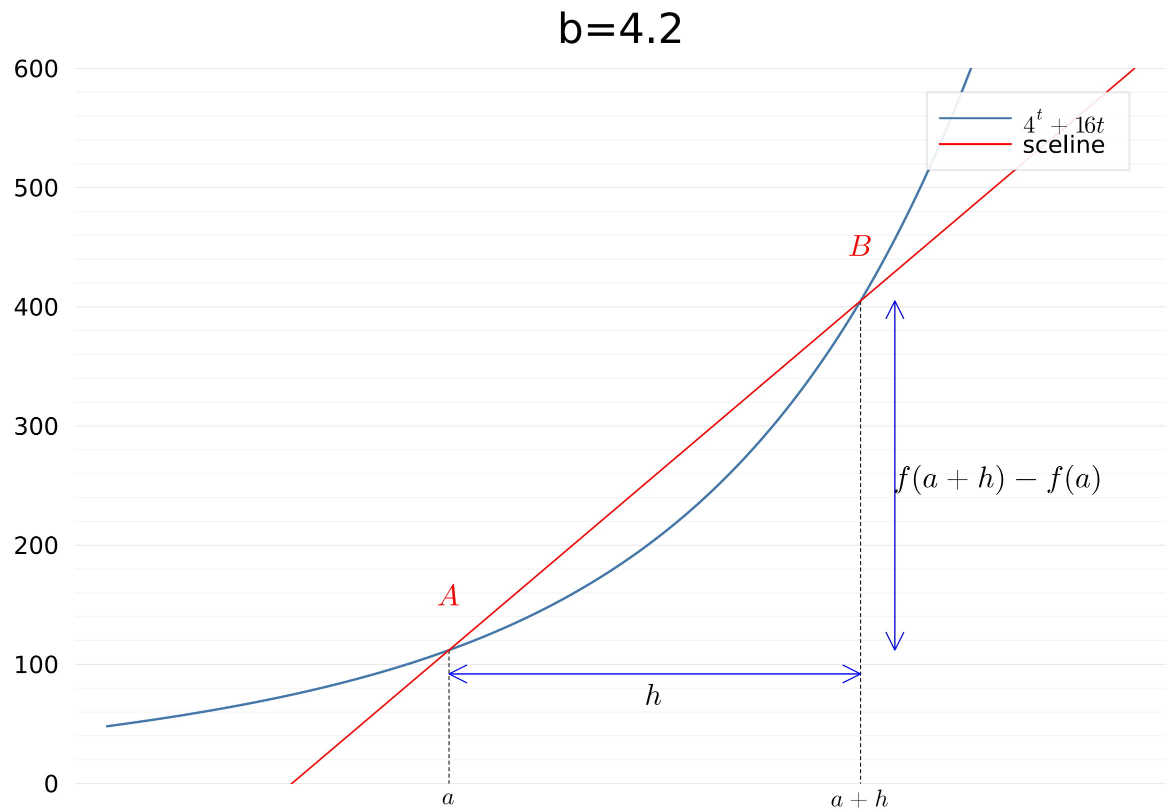
```
md"""
## 导数的可视化
下面看一下交互式导数可视化的图形.

"""

```

4.2

```
@bind deltah Slider(3.2:0.2:4.2, default=4.0 ,show_value=true)
```




```
let
    a=3        # 值域中的 a
    b= deltah     # 值域中的 b 由上面Slider 的 deltah 控制
    h=b-a      # 差值
    f(t)=4^t+16t    # 函数
    tspan=2:0.02:5
    m=(f(b)-f(a))/h   #仿射割线的斜率
    fl(x)=m*x-m*a+f(a)   #经过 AB两点的仿射直线方程
    http://ambrnet.com/MathCalc/Line/Line_.htm
    yoffset=50    #用于注释的 y轴偏移距离
    pA=[a f(a)]    #曲线上 A 点的坐标
    pB=[b f(b)]    #曲线上 B 点的坐标
    pa=[a 0]       #  x轴上的投影坐标
    pb=[b 0]      #  y轴上的投影坐标
    ph=[a+0.5h f(a)-20]  # 用于标注 h的中间点坐标
    pf=[b+0.1 f(a)+(f(b)-f(a))/2]
    lAa=[pA;pa]    # 从 A到 a 的线段的 矩阵 下同
    lBb=[pB;pb]
    ldeltah=[pA;[a f(a)]]
    arrowhA=[ph;[a f(a)-20]]
    arrowhB=[ph;[b f(a)-20]]
    arrowfA=[pf;[b+0.1 f(a)]]
    arrowfB=[pf;[b+0.1  f(b)]]
    ann=[(a,f(a)+yoffset,text(L"A",color=:red,pointsize=10)),
          (b,f(b)+yoffset,text(L"B",color=:red,pointsize=10)),
          (a,-10,text(L"a",pointsize=8)),
           (b,-10,text(L"a+h",pointsize=8)),
           (ph[:,1],ph[:,2],text(L"h",pointsize=10,valign=:top)),
           (pf[:,1],pf[:,2],text(L"f(a+h)-f(a)",pointsize=10, halign=:left))

    ]
```

```julia
    plot(f,tspan,label=L"4^t+16t",ylims=
    (0,600),ann=ann,xticks=:none,title="b=$(deltah)")
    plot!(lAa[:,1],lAa[:,2], ls=:dash,color=:black,lw=0.5,label=false)
    plot!(lBb[:,1],lBb[:,2], ls=:dash,color=:black,lw=0.5,label=false)
    plot!(ldeltah[:,1],ldeltah[:,2], ls=:dash,color=:black,lw=0.5,label=false)
    plot!(arrowhA[:,1],arrowhA[:,2],arrow=0.1,color=:blue,lw=0.6,label=false)
    plot!(arrowhB[:,1],arrowhB[:,2],arrow=0.1,color=:blue,lw=0.6,label=false)
    plot!(arrowfA[:,1],arrowfA[:,2],arrow=0.1,color=:blue,lw=0.6,label=false)
    plot!(arrowfB[:,1],arrowfB[:,2],arrow=0.1,color=:blue,lw=0.6,label=false)
    plot!(fl,tspan,color=:red,label="sceline",lw=0.8)


end
```

example3 局部放大的正弦函数

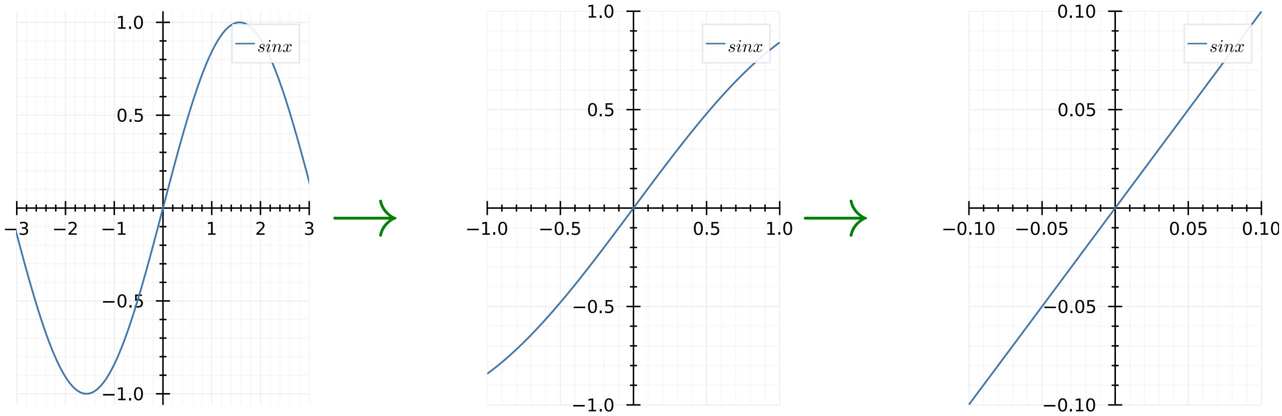
在 $x=0$ 附近的位置, 当我们缩小绘图的区间(放大图像局部)

当我们在极小的区间内研究非线性函数的行为时,可以用线性的直线来代替. 物理学家费曼说过:我们用线性近似解决非线性问题,是因为我们只能理解直线

```julia
md"""
example3 局部放大的正弦函数

在 $x=0$ 附近的位置, 当我们缩小绘图的区间(放大图像局部)

当我们在极小的区间内研究非线性函数的行为时,可以用线性的直线来代替. 物理学家费曼说过:我们用线性近似解决非线性问题,是因为我们只能理解直线
"""
```



```
let
    tspan1=-pi:0.02:pi
    tspan2=-1:0.02:1
    tspan3= -0.1:0.002:0.1
    f(x)=sin(x)
    function ann(x,y,str)
        return (x,y,text(str, pointsize=32, color=:green, halign=:center,
        valign=:center, rotation=0,))
    end
    l = @layout [
    a{0.3w} b{0.05w} c{0.3w} d{0.05w} e{0.3w} ]
    p1=plot(f,tspan1,label=L"sinx",xlims=(-3,3),frame=:origin)
    ann1=[ann(0.5,0.5,L"\rightarrow")]
    p2=plot(ann=ann1,grid=false,ticks=false)
    p3=plot(f,tspan2,label=L"sinx",xlims=(-1,1),frame=:origin,ylims=(-1,1))
    p4=plot(ann=ann1,grid=false,ticks=false)
    p5=plot(f,tspan3,label=L"sinx",xlims=(-0.1,0.1),ylims=(-0.1,0.1),frame=:origin)
    plot!(p1,p2,p3,p4,p5,layout=l,size=(900,300))

end
```

example4

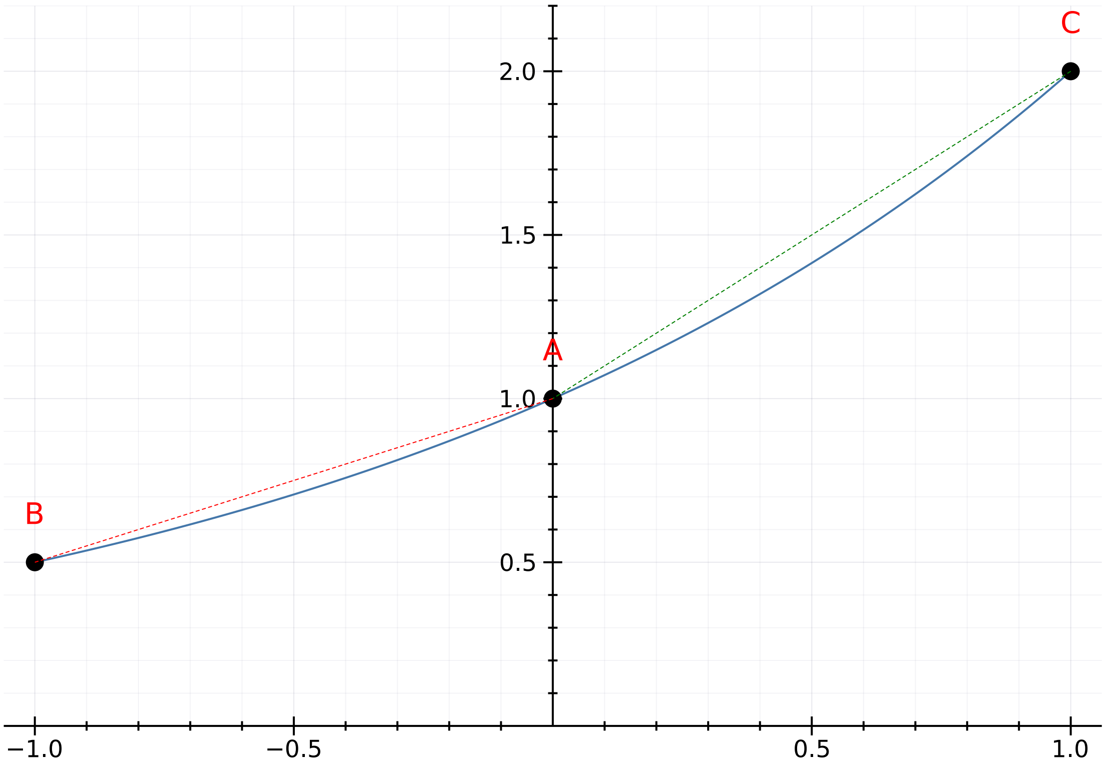
用数值和图形方法估计 函数 $f(x)=2^x$ 在 $x=0$ 处的导数

```
md"""
    **example4**

    用数值和图形方法估计 函数 $f(x)=2^x$ 在 $x=0$ 处的导数
"""
```

几何方法

```
md"""
  几何方法
"""
```




```
let
    tspan=-1.0:0.02:1.0
    f(x)=2^x
    yoffset=0.15
    pb=[-1.0 f(-1.0)]
    pa=[0.0 f(0.0)]
    pc=[1.0 f(1.0)]
    lineab=[pa;pb]
    lineca=[pc;pa]

    ann=[(-1.0,f(-1.0)+yoffset,text("B",color=:red,pointsize=10)),
          (0.0,f(0.0)+yoffset,text("A",color=:red,pointsize=10)),
          (1.0,f(1.0)+yoffset,text("C",color=:red,pointsize=10)),
    ]
    plot(f,tspan,frame=:origin, lw=1,label=false)

    scatter!([-1,0, 1],[f(-1.0) ,f(0.0), f(1.0)],label=false,ann=ann,ylims=
    (0,2.2),color=:black,ms=5)
    plot!(lineab[:,1],lineab[:,2],label=false,color=:red,ls=:dash,lw=0.5)
    plot!(lineca[:,1],lineca[:,2],label=false,color=:green,ls=:dash,lw=0.5)
end
```

在这里我们会用到中值定律和斜率计算方法, 由于$f(x)=2^x$ 是连续连续函数,适用中值定律斜率计算方法-如果在坐标系中有两点$a(x_1,y_1),b(x_2,y_2)$ 斜率为:

$$m_{ba}=\frac{y_2-y_1}{x_2-x_1}$$

1. 直线AB 的斜率

$$\frac{2^0-2^{-1}}{0-(-1)} =\frac{1}{2}$$

2. 直线CA 的斜率

$$\frac{2^1-2^{0}}{1-0} =1$$

3. 由于A点位于 B和C 中间, A点处的导数一定在$\frac{1}{2}$ 和 1 之间. 这就是中值定律告诉我们结论

```
md"""
 在这里我们会用到中值定律和斜率计算方法, 由于$f(x)=2^x$ 是连续连续函数,适用中值定律
   斜率计算方法-如果在坐标系中有两点$a(x_1,y_1),b(x_2,y_2)$
   斜率为:

$$m_{ba}=\frac{y_2-y_1}{x_2-x_1}$$



1. 直线AB 的斜率

$$\frac{2^0-2^{-1}}{0-(-1)} =\frac{1}{2}$$

2. 直线CA 的斜率

$$\frac{2^1-2^{0}}{1-0} =1$$

3. 由于A点位于 B和C 中间, A点处的导数一定在$\frac{1}{2}$ 和 $1$ 之间. 这就是中值定律告诉我们结论

"""
```

example4 的数值计算 我们在$x=0$ 的左右两侧, 分别取一小段距离, 看看函数值的变化情况有什么规律,计算方法仍是斜率计算

$$\frac{f(0+h)-f(0)}{h-0} =\frac{2^h-2^0}{h}=\frac{2^h-1}{h}$$

```
md"""
 example4 的数值计算
 我们在$x=0$ 的左右两侧, 分别取一小段距离, 看看函数值的变化情况有什么规律,计算方法仍是斜率计算

$$\frac{f(0+h)-f(0)}{h-0} =\frac{2^h-2^0}{h}=\frac{2^h-1}{h}$$
"""
```

	h	hvalue	diffq
1	-0.0003	0.999792	0.693075
2	-0.0002	0.999861	0.693099
3	-0.0001	0.999931	0.693123
4	0.0001	1.00007	0.693171
5	0.0002	1.00014	0.693195
6	0.0003	1.00021	0.693219

```
let
    hspan=[-0.0003,-0.0002,-0.0001,0.0001,0.0002,0.0003]
    f(x)=2^x

    diffQ(h,p=0)=(f(h)-f(p))/h  # p=0 为初始点,默认为0 点
    value=[f(h) for h in hspan]
    diff=[diffQ(h,0) for h in hspan]
    df=DataFrame(;h=hspan,hvalue=value,diffq=diff)

end
```

从数值计算可以看到在极小的一步下, 斜率趋近于 0.693

所以

$$f'(0) \approx 0.693$$

```
md"""
 **从数值计算可以看到在极小的一步下, 斜率趋近于 0.693**

 所以

$${f}'(0)\approx 0.693$$

"""
```

example5 : 找到$f(x)=2^x$ 在 $x=0$ 处的近似切线方程

在 example4 中已经用 数值方法近似获得了斜率, 由于在$x=0$ 的截距为 1(截距定义就是$x=0$处 y的取值)

所以$x=0$ 处的切线方程为:

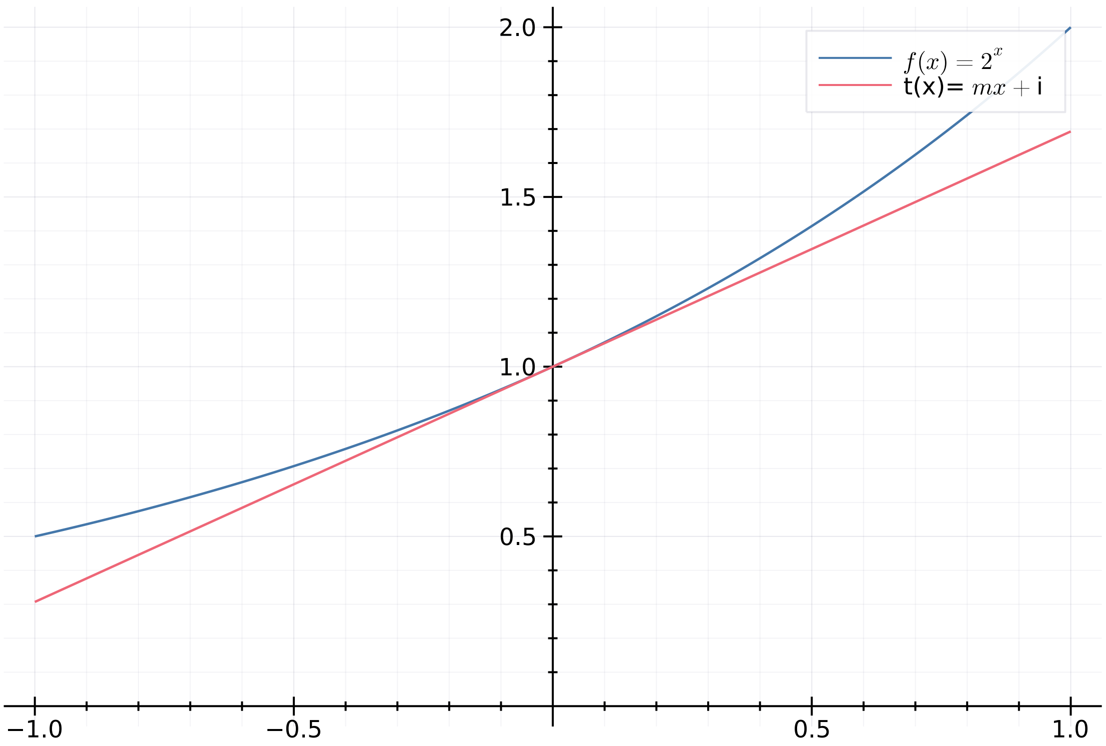
$$y=0.693x+1$$

```
md"""
**example5** : 找到$f(x)=2^x$ 在 $x=0$ 处的近似切线方程

在 example4 中已经用 数值方法近似获得了斜率, 由于在$x=0$ 的截距为 1(截距定义就是$x=0$处 y的取
值)

所以$x=0$ 处的切线方程为:

$$y=0.693x+1$$
"""
```




```
let
    tspan=-1.0:0.02:1.0
    f(x)=2^x
    m=0.693  # 斜率作为参数传入方程,截距一样处理
    intercept=i=1   # 截距参数
    tagentline(x)= m*x+i
    plot([f,tagentline],tspan,frame=:origin,label=[L"f(x)=2^x" L"t(x)= $mx+$i"])
end

```

代数方法计算导数

请参见 书的 example6

在用代数方法计算导数时,有个问题要注意,就是遇到分母为 0 的情况, 当我们在计算 $x=0$ 处的导数时, 我们计算的 h 是离 0 很近的一段距离,并不是 0 本身. 这就是数系的威力所在, 如果定义在实数系, 我们可以无限靠近 0, 分母为 0 不行, 不破坏这条规则, 利用实数系里的数字可以无限靠近 0, 从而解决问题.

数系的演化都是为了解决出现的问题, 前提是不破坏已有的规则.

```
md"""
## 代数方法计算导数

请参见 书的 example6

在用代数方法计算导数时,有个问题要注意,就是遇到分母为 $0$ 的情况, 当我们在计算 $x=0$ 处的导数时, 我们计算的 $h$ 是离 $0$ 很近的一段距离,并不是 $0$ 本身. 这就是数系的威力所在, 如果定义在实数系, 我们可以无限靠近 $0$, 分母为 $0$ 不行, 不破坏这条规则, 利用实数系里的数字可以无限靠近 $0$, 从而解决问题.

数系的演化都是为了解决出现的问题, 前提是不破坏已有的规则.
"""
```