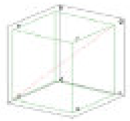


ch02 sec 2.3 导函数

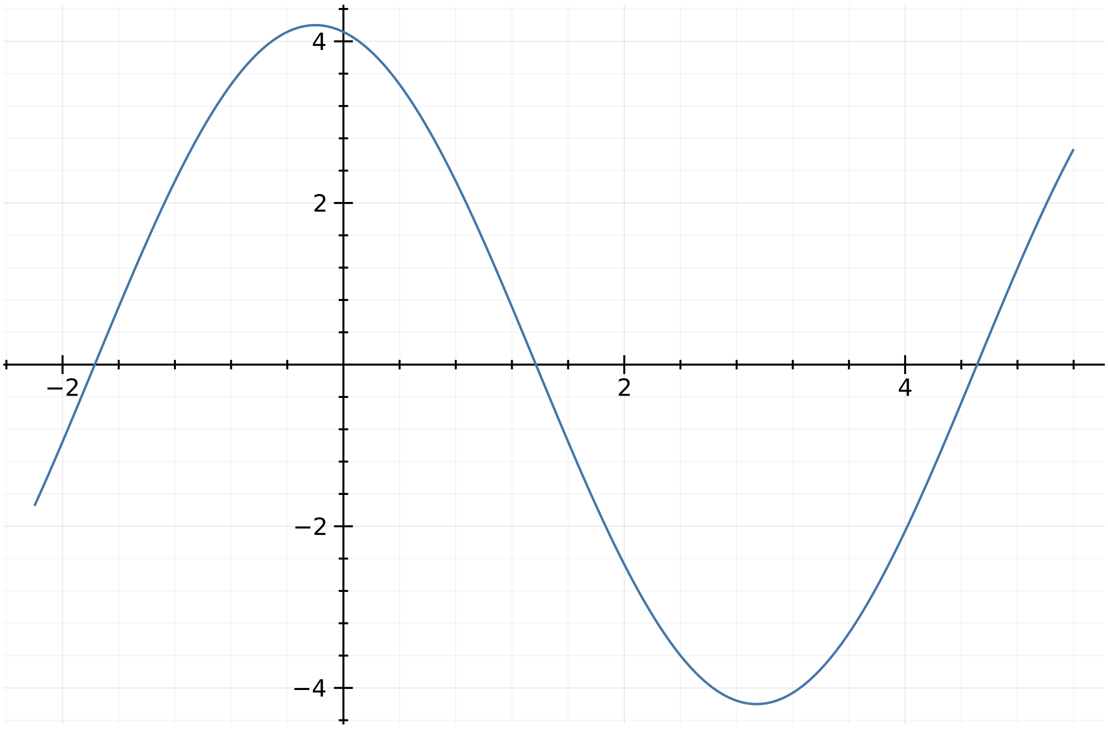
Table of Contents


```julia
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
```

example1 计算并绘制 $f(x)$ 在 −2, −1, 0, 1, 2, 3, 4, 5

这里实际是绘制斜率场, 按照一定间隔绘制出斜率的图形, 会大致直观表示函数的变化特性

```julia
md"""
**example1** 计算并绘制 $f(x)$ 在 -2, -1, 0, 1, 2, 3, 4, 5

这里实际是绘制斜率场, 按照一定间隔绘制出斜率的图形, 会大致直观表示函数的变化特性
"""
```




```
let
    tspan=-2.2:0.02:5.2
    spot=[-2, -1, 0, 1, 2, 3, 4, 5 ]
    f(x)=4.2*cos(x+0.2)  # 并不知道原函数是什么，所以猜测了一个函数表达式
    plot(f,tspan,label=false,frame=:origin)

end
```

	spot	slope
1	-2	4.09064
2	-1	3.01143
3	0	-0.836469
4	1	-3.91532
5	2	-3.39445
6	3	0.247268
7	4	3.66165
8	5	3.70953

```
let
    spot=[-2, -1, 0, 1, 2, 3, 4, 5 ]
    f(x)=4.2*cos(x+0.2)
    h=0.001  #步进一小段距离
    m(h,p=0)=(f(p+h)-f(p))/h
    marr=[m(h,p) for p in spot]
    df=DataFrame(;spot=spot,slope= marr)

end
```

由于所取 h为很小的距离, 两点之间的斜率可以近似表示该点切线的斜率, 然后可以使用这个斜率绘制出一小段线段,直观显示在该位置的变化趋势.

```
md"""
 由于所取 h为很小的距离, 两点之间的斜率可以近似表示该点切线的斜率, 然后可以使用这个斜率绘制出一小段线段,直观显示在该位置的变化趋势.
"""
```


```
let
    tspan=-3:0.02:5.2
    spot=[-2, -1, 0, 1, 2, 3, 4, 5 ]
    f(x)=50*cos(x+0.2)
    range=0.4
    h=0.001  #步进一小段距离
    m(h,p=0)=(f(p+h)-f(p))/h  #经过某个点的斜率计算, 步进一小段距离
    p1=plot(f,tspan,frame=:zerolines,ls=:dash, lw=:0.5) #函数图
    plotarr=[p1]
    for p in spot  # for 循环生成斜率场图数组
       slope=m(h,p)
       intercept=f(p)
       affline(x)=slope*x+(f(p)-slope*p)
       linesegment=[[p-range affline(p-range)];[p+range affline(p+range)]]
       line=plot!(linesegment[:,1],linesegment[:,2],label=false,lw=2)
       push!(plotarr,line)
    end
    plot!(plotarr...,size=(400,500))


end
```

Notice

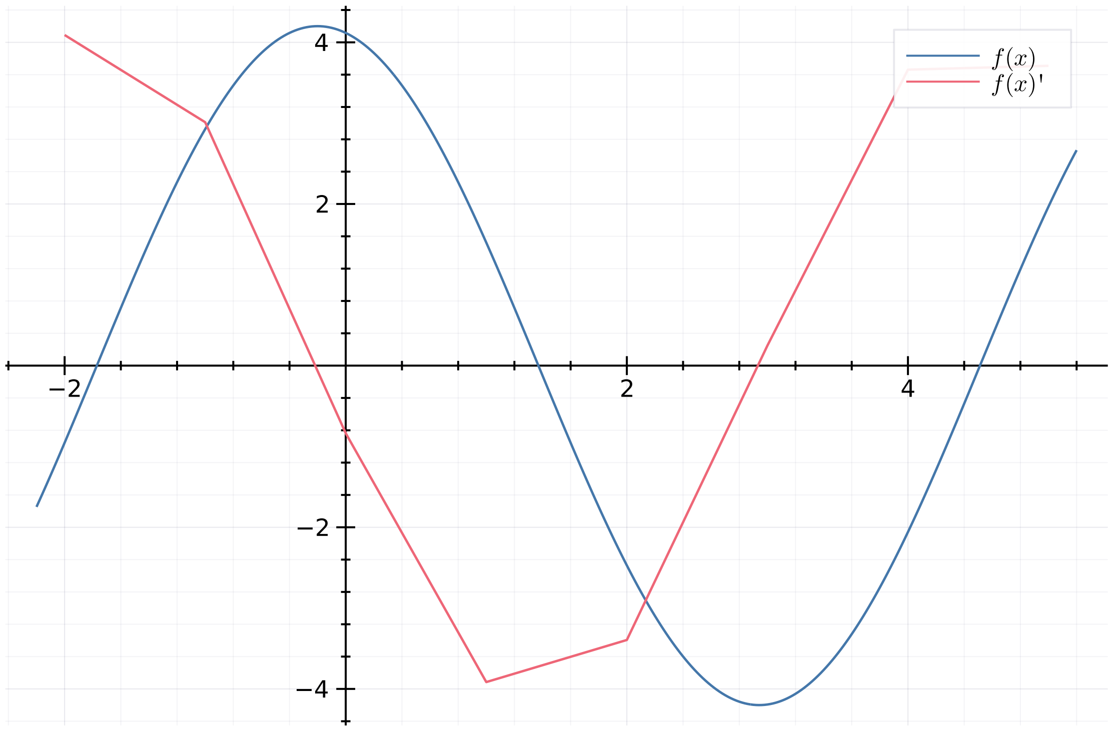
在数值计算部分实际我们已经定义了导函数，给定定义域，定义域就是要求导数的点，值域就是求出的导数，还有求导数的方法

$$定义域\ x\in[-2,-1,0,1,2,3,4,5]$$

$$值域\ y\in[4.09064,3.01143,-0.836469,-3.91532,-3.39445,0.247268,3.66165,3.70953]$$

$$映射规则:{f(x)}'=\frac{f(x+h)-f(x)}{h}$$

```
md"""
!!! notice
	在数值计算部分实际我们已经定义了导函数，给定定义域，定义域就是要求导数的点,值域就是求出的导数，还有求导数的方法
$$定义域 x\in [-2, -1, 0, 1, 2, 3, 4, 5 ]$$

$$值域 y \in [4.09064,3.01143,-0.836469,-3.91532,-3.39445,0.247268,3.66165,3.70953]$$

$$映射规则:{f(x) }'=\frac{f(x+h)-f(x)}{h}$$

"""
```




```
let
    tspan=-2.2:0.02:5.2
    spot=[-2, -1, 0, 1, 2, 3, 4, 5 ]
    h=0.001  #步进一小段距离
    f(x)=4.2*cos(x+0.2)  # 并不知道原函数是什么，所以猜测了一个函数表达式
    m(h,p=0)=(f(p+h)-f(p))/h
    marr=[m(h,p) for p in spot]
    df=DataFrame(;spot=spot,slope= marr)
    plot(f,tspan,frame=:origin,label=L"f(x)")
    plot!(df.spot,df.slope,label=L"{f(x)}'")




end
```

导函数的图形反映了函数的变化的特性, 注意几个要点:

1. 与x轴的交点(y=0)的点
2. 正负符号交换的区间,这里是我们寻找函数最大值或者最小值的区域
3. 同符号折返的点, 例如下降变上升或者上升变下降(称为反射点)
4. 导函数是把定义域的值映射为该点的切线的斜率, 度量的是在某个x处的变化情况,这与原来函数的意义不同

后面的学习要反复理解这几条, 函数能够表现出和生物一样的各种特性和脾气, 一旦你理解了,你就能和它融洽相处

```
md"""
 导函数的图形反映了函数的变化的特性, 注意几个要点:
 1.  与x轴的交点(y=$0$)的点
 2.  正负符号交换的区间,这里是我们寻找函数最大值或者最小值的区域
 3.  同符号折返的点, 例如下降变上升或者上升变下降(称为反射点)
 4.  导函数是把定义域的值映射为该点的切线的斜率, 度量的是在某个$x$处的变化情况,这与原来函数的意义不同

**后面的学习要反复理解这几条,  函数能够表现出和生物一样的各种特性和脾气, 一旦你理解了,你就能和它融洽相处**
"""
```

导函数的图像告诉了我们内容?

Notice

如果在区间内 ${f(x)}'>0$,说明$f(x)$ 在区间内递增.

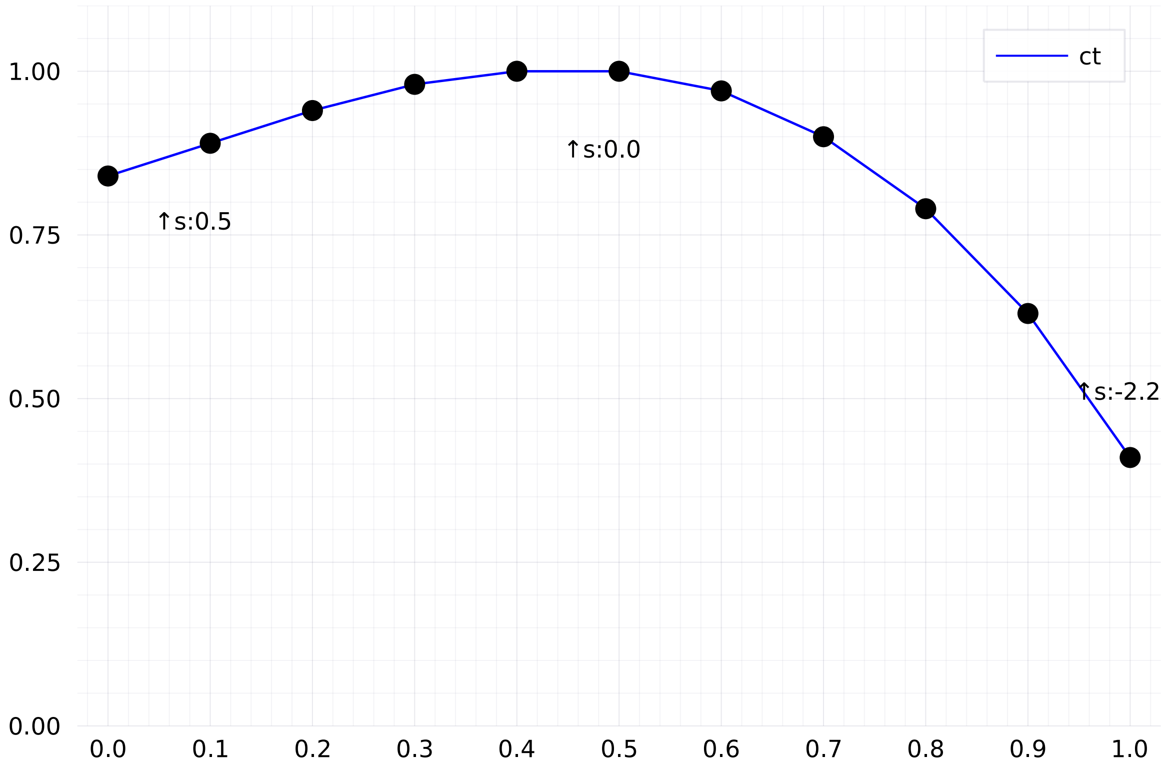
如果在区间内 ${f(x)}'<0$,说明$f(x)$ 在区间内递减.

```
md"""
## 导函数的图像告诉了我们内容?

!!! notice
	如果在区间内 ${f(x)}'>0$,说明$f(x)$ 在区间内递增.

	如果在区间内 ${f(x)}'<0$,说明$f(x)$ 在区间内递减.

"""

```

数值角度看导函数

example3 药物浓度在血液中的变化情况

```
md"""
## 数值角度看导函数
**example3  药物浓度在血液中的变化情况**
"""
```



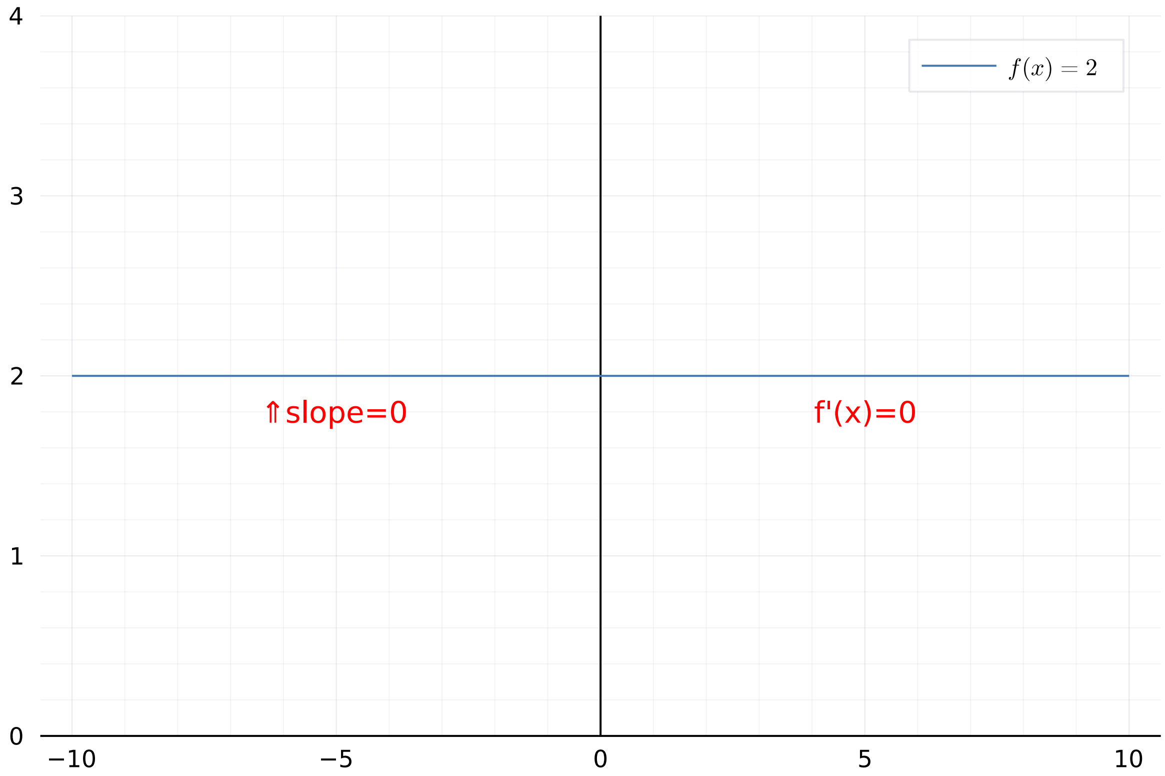

```
let
    yoffset=0.1
    h=0.1
    tspan=0:h:1
    tspanp=0:h:0.9
    ct=[0.84,0.89,0.94,0.98,1.00,1.00,0.97,0.90,0.79,0.63,0.41]
    @show df=DataFrame(;time=tspan,ct=ct)
    ctp=[]   # ct'
    size=length(ct)-1
    for i in 1:size
        prime=(ct[i+1]-ct[i])/h
        push!(ctp,round(prime,digits=2))
    end
    function midpointval(i)
         a=ct[i]
         b=ct[i+1]
        return a>=b ? (a) : (b)

    end
    ann=[
       (0.05,midpointval(1)-yoffset,text("↑s:$(ctp[1])",halign=:left,valign=:top,pointsize=8))
       (0.45,midpointval(5)-yoffset,text("↑s:$(ctp[5])",halign=:left,valign=:top,pointsize=8))
       (0.95,midpointval(10)-yoffset,text("↑s:$(ctp[10])",halign=:left,valign=:top,pointsize=8))
    ]

    plot(tspan,ct,label="ct",color=:blue,xticks=0:0.1:1,ann=ann)
    scatter!(tspan,ct,label=false,ylims=(0,1.1),mc=:black)


end
```

```
df = DataFrame(; time = tspan, ct = ct) = 11×2 DataFrame
 Row │ time     ct
     │ Float64  Float64
─────┼──────────────────
   1 │     0.0     0.84
   2 │     0.1     0.89
   3 │     0.2     0.94
   4 │     0.3     0.98
   5 │     0.4     1.0
   6 │     0.5     1.0
   7 │     0.6     0.97
   8 │     0.7     0.9
   9 │     0.8     0.79
  10 │     0.9     0.63
  11 │     1.0     0.41
```

从函数表达式获取导函数

平行于 x 轴的直线斜率为 0, 导数也为 0

```
md"""
## 从函数表达式获取导函数

### 平行于 x 轴的直线斜率为 0, 导数也为 0

"""
```




```
let
    xspan=-10:10
    y(x)=2
    ann=[(5,1.8,text("f'(x)=0",pointsize=10,color=:red)),
          (-5,1.8,text("⇑slope=0",pointsize=10,color=:red))
    ]
    plot(y,xspan, frame=:zerolines,label=L"f(x)=2",lw=1,ylims=(0,4),ann=ann)
end
```

线性函数的导数

如:

$$f(x) = mx + b$$

$$f(x)' = m$$

```
md"""
 ### 线性函数的导数
如:

$$f(x)=mx+b$$

$${f(x)}'=m$$
"""
```

幂函数的导函数

这部分请参考教材 page 127

结论是:

$$\text{如果有 } f(x)=x^n, \text{则 } {f(x)}'=nx^{n-1}$$

```
md"""
### 幂函数的导函数
这部分请参考教材 page 127

结论是:

$$如果有f(x)=x^n, 则 {f(x)}'=nx^{n-1}$$
"""
```

```
Enter cell code...
```