



# 弹簧与微分方程

`In[*]:= NDSolve[{-k x[t] - u x'[t] == m x''[t], x[0] == 3, x'[0] == 0} /. {k -> 10, u -> 2, m -> 5},`  
[数值求解微分方程组](#)  
`{x, x'}, {t, 0, 15}]`

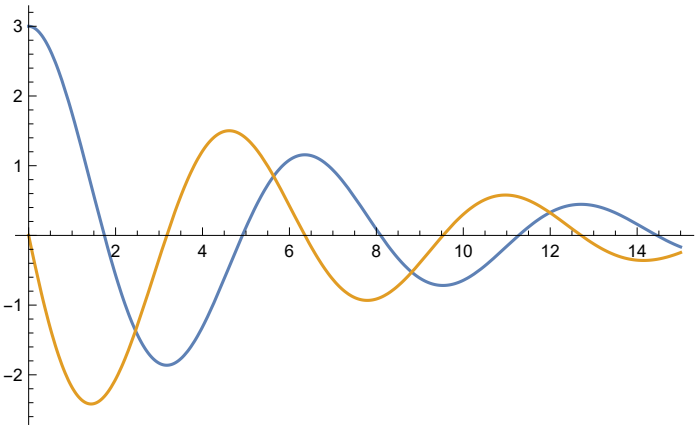
`Out[*]=`

`{ {x -> InterpolatingFunction[`  
 `Domain: {{0., 15.}}`  
`Output: scalar` `],`  
  
`x' -> InterpolatingFunction[`  
 `Domain: {{0., 15.}}`  
`Output: scalar` `]] }`

`In[*]:= {pos, vel} = NDSolveValue[{-k x[t] - u x'[t] == m x''[t], x[0] == 3, x'[0] == 0} /.`  
[数值解的值](#)

`{k -> 10, u -> 3, m -> 10}, {x, x'}, {t, 0, 15}];`  
`Plot[{pos[t], vel[t]}, {t, 0, 15}, PlotRange -> All]`  
[绘图](#) [绘制范围](#) [全部](#)

`Out[*]=`



## 2维，有初速度

```

In[ ]:= {pos, vel} =
  NDSolveValue[{k ({1, 1} - x[t]) - u x'[t] == m x''[t], x[0] == {2, 2}, x'[0] == {-1, 0}} /.
    {k → 10, u → 10, m → 10}, {x, x'}, {t, 0, 15}];
ParametricPlot[{pos[t], vel[t]}, {t, 0, 15}, PlotRange → All]

```

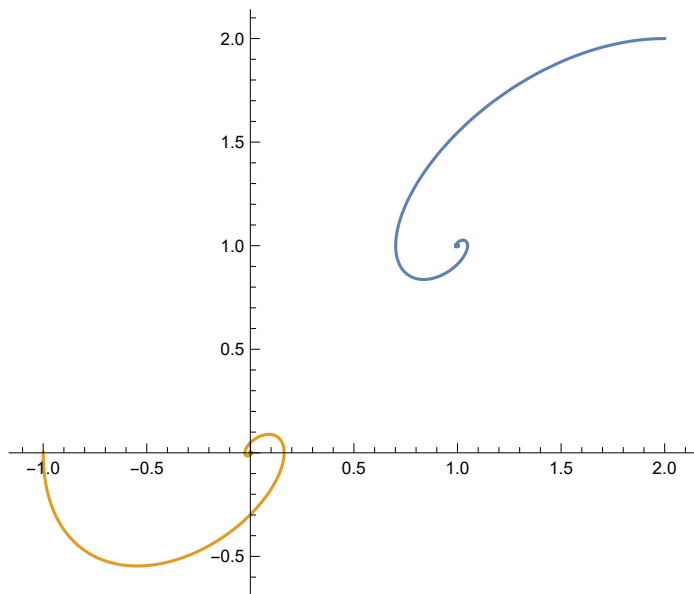
数值解的值

绘制参数图

绘制范围

全部

Out[ ]:=



## 上面其实更像万有引力

水平面上一个长度为1、劲度系数为k的弹簧，一端固定在原点，另一端连接一个质量为m的小球，设置小球的初始位置和速度，计算其位置和速度随时间的变化

只考虑随速度变化的阻力

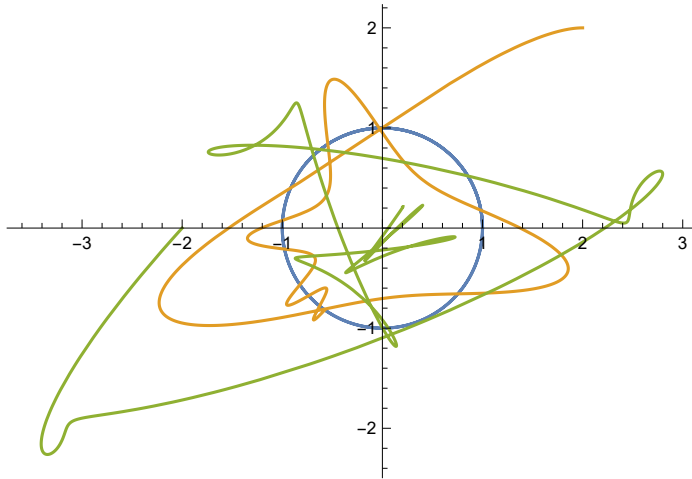
```

In[ ]:= {pos, vel} = NDSolveValue[
    {k (Normalize[x[t]] - x[t]) - u x'[t] == m x''[t], x[0] == {2, 2}, x'[0] == {-2, 0}} /.
    {k -> 50, u -> 3, m -> 10}, {x, x'}, {t, 0, 15}];
ParametricPlot[{{Cos[t], Sin[t]}, pos[t], vel[t]}, {t, 0, 15}, PlotRange -> All]

```

数值解的值  
正规化  
余弦  
正弦  
绘制参数图  
绘制范围  
全部

Out[ ]:=



In[ ]:= pos

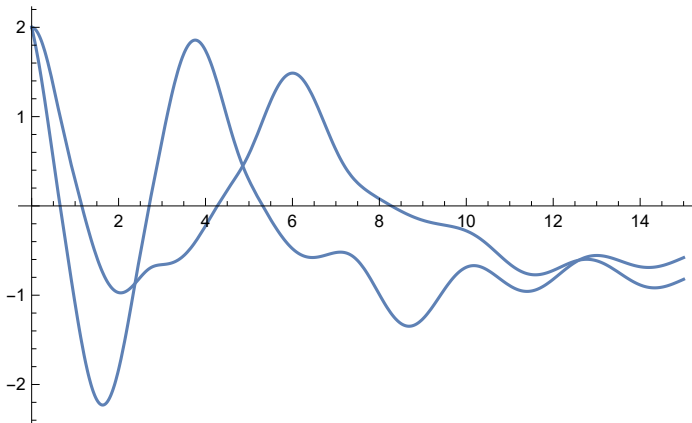
Out[ ]:=

InterpolatingFunction[  
 Domain: {{0., 15.}}  
 Output dimensions: {2}

In[ ]:= Plot[%28[x], {x, 0., 15.}]

绘图

Out[ ]:=



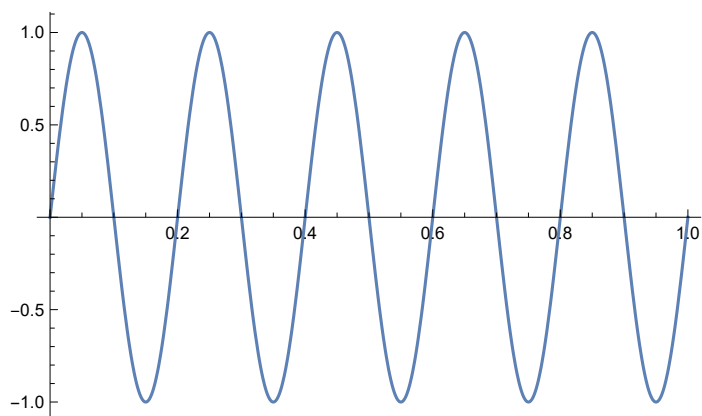
## 可视化

画一个弹簧

In[ ]:= **Plot**[**Sin**[ $2\pi 5 x$ ], {x, 0, 1}]

[绘图](#) [正弦](#)

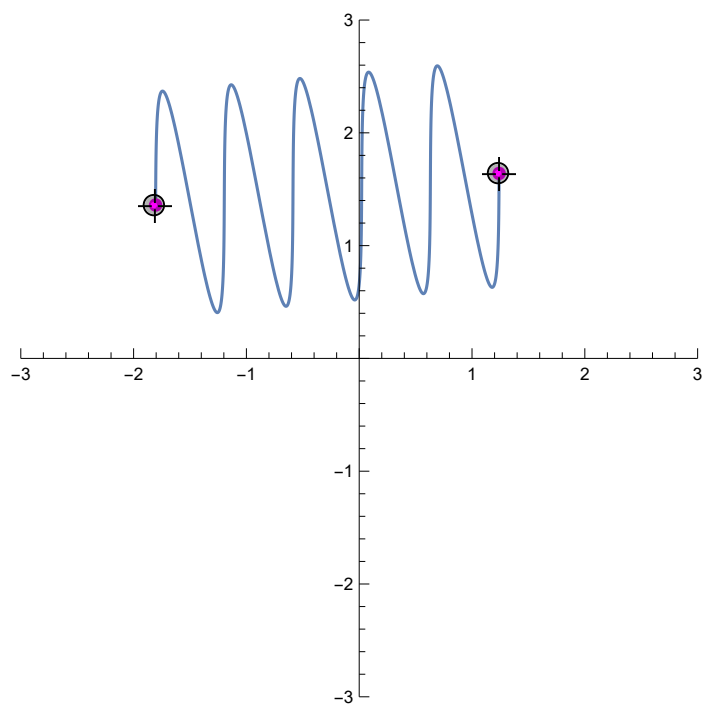
Out[ ]:=



将弹簧两端固定到任意两个点

In[ ]:= **DynamicModule**[{pts = **RandomPoint**[**Disk**[{0, 0}, 3], 2], v},  
[动态模块](#) [伪随机点](#) [圆盘](#)  
**LocatorPane**[**Dynamic**[pts], **Dynamic@Show**[v = **Complex@@**(pts[[2]] - pts[[1]]);  
[定位器窗格](#) [动态](#) [动态](#) [显示](#) [复数](#)  
**ParametricPlot**[**RotationMatrix**[**Arg**[v]] . {**Abs**[v] x, **Sin**[ $2\pi 5 x$ ]} + pts[[1]], {x, 0, 1},  
[绘制参数图](#) [旋转矩阵](#) [辐角](#) [绝对值](#) [正弦](#)  
**PlotRange** → 3], **Graphics**[{**Magenta**, **PointSize**[**Large**], **Point**[pts]}]]]]]

Out[ ]:=



```

In[ ]:= DynamicModule[{pt1 = {0, 0}, pt2 = {2, 2}, v}, LocatorPane[Dynamic[pt2,
  动态模块 定位器窗格 动态
    {pt2 = # &, {pos, vel} = NDSolveValue[{k (Normalize[x[t]] - x[t]) - u x'[t] == m x''[t],
      数值解的值 正规化
        x[0] == #, x'[0] == {-2, 0}} /. {k → 50, u → 3, m → 10},
        {x, x'}, {t, 0, 15}] &}], Dynamic[v = Complex@@ (pt2 - pt1);
      动态 复数
    Show[ParametricPlot[{Cos[t], Sin[t]}, pos[t], vel[t]], {t, 0, 15}, PlotRange → All],
      显示 绘制参数图 余弦 正弦 绘制范围 全部
    ParametricPlot[RotationMatrix[Arg[v]].{Abs[v] x, Sin[2 π 5 x]} + pt1,
      绘制参数图 旋转矩阵 辐角 绝对值 正弦
        {x, 0, 1}, PlotRange → 3],
      绘制范围
    Graphics[{Magenta, PointSize[Large], Point[pt2]}], ImageSize → Large]]]
  图形 品红色 点的大小 大 点 图像尺寸 大

```

Set: 列表 {pos, vel} 和 NDSolveValue[{k Plus[<<2>>] - Times[<<2>>] == m x''[t], x[0] == #1, x'[0] == {-2, 0}} /. {k → 50, u → 3, m → 10}, {x, x'}, {t, 0, 15}] & 形状不同.

Out[ ]:=

