第6讲 在 Mathematica 中作图

6－7 图形动画

动画演示提供了交互运行函数和命令的方式，让函数展开、积分运算等数学运算生动起来；

动画演示函数图形简单明了而栩栩如生.

1. Manipulate 交互式演示

`Manipulate[expr, {u, ua, ub}]` 动画演示表达式**expr**在区间 **{ua, ub}** 的所有值

`Manipulate[expr, {u, ua, ub, dstep}]` 控制量**u**在区间 **{ua, ub}** 之间以步长**dstep**变化

`Manipulate[expr, {u, {u1, u2, ...}}]` **u**取离散值**u1, u2, ...**

`Manipulate[expr, {u, ...}, {v, ...}, ]` 设置两个或多个控制量

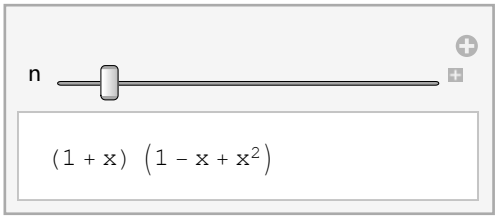
控制量**u**（或称滑杆），**expr**可为函数、图形等多种形式的表达式.

例1：动画演示中的循环变量的步长1常常不能省略.

```
Manipulate[n, {n, 1, 20}]
Manipulate[n, {n, 1, 20, 1}]
```

In[7]:=
```
Manipulate[Factor[1 + x^n], {n, 2, 12, 1}]
```

Out[7]=



例2：选项 **PlotRange** 在动画演示中的作用.

```
Manipulate[Plot[x Sin[a x] / 4, {x, 0, 9}], {a, 0, 2}]
Manipulate[Plot[x Sin[a x] / 4, {x, 0, 9}, PlotRange -> {-3, 3}], {a, 0, 2}]
```

例3：动画演示随初始条件变化的微分方程解函数.

```
Manipulate[
 Plot[Evaluate[y[t] /. First[NDSolve[ {y''[x] == -x y[x], y[0] == a, y'[0] == b},
       y, {x, 0, 4}]]], {t, 0, 4}, PlotRange → 4],
 {{a, 1, TraditionalForm[y[0]]}, -3, 3},
 {{b, 0, TraditionalForm[y'[0]]}, -3, 3}]
```

2. Animate 动画演示

`Animate[expr, {u, umin, umax}]`	在 {umin, umax} 区域内动态演示表达式expr
`Animate[expr, {u, umin, umax, du}]`	按步长du演示表达式expr
`Animate[expr, {u, {u1, u2, ...}}]`	u 取离散值u1, u2
`Animate[expr, {u, ...}, {v, ...}, ...]`	取每一对 {u, v} 的值演示表达式expr

例4：动态演示求导.

```
Animate[D[x^n, x], {n, 2, 10, 1}]
```

```
Animate[D[x^n, x], {n, 2, 10}]
```

例5：双重动画变量.

```
Animate[Plot[Sin[a x] + Sin[b x], {x, 0, 10}, PlotRange → 2], {a, 1, 5}, {b, 1, 5}]
```

例6：演示点绕圆周.

```
Animate[
 Graphics[{Blue, Thick, Circle[], Red, PointSize[0.03], Point[{Cos[t], Sin[t]}]}],
 {t, 0, 2 Pi, Pi/24}]
```

例7：演示一列水波.

```
Animate[Plot3D[Sin[Sqrt[x^2 + y^2] + t * 2 * Pi],
  {x, -8 Pi, 8 Pi}, {y, -8 Pi, 8 Pi}, PlotRange → 10, PlotPoints → 50,
  AspectRatio → 1, Boxed → False], {t, 2, 0}, AnimationRunning → False]
```

例8：直纹面形成单叶双曲面.

```
t = {-1.5, 1.5};
Animate[Graphics3D[Table[Line[{{Cos[t] - Sin[t], Cos[t] + Sin[t], 1},
      {Cos[t] + Sin[t], Sin[t] - Cos[t], -1}}], {t, 0, s, Pi/27}],
  Boxed -> False, PlotRange -> {t, t, t}], {s, 0, 2 Pi, Pi/27}]
```

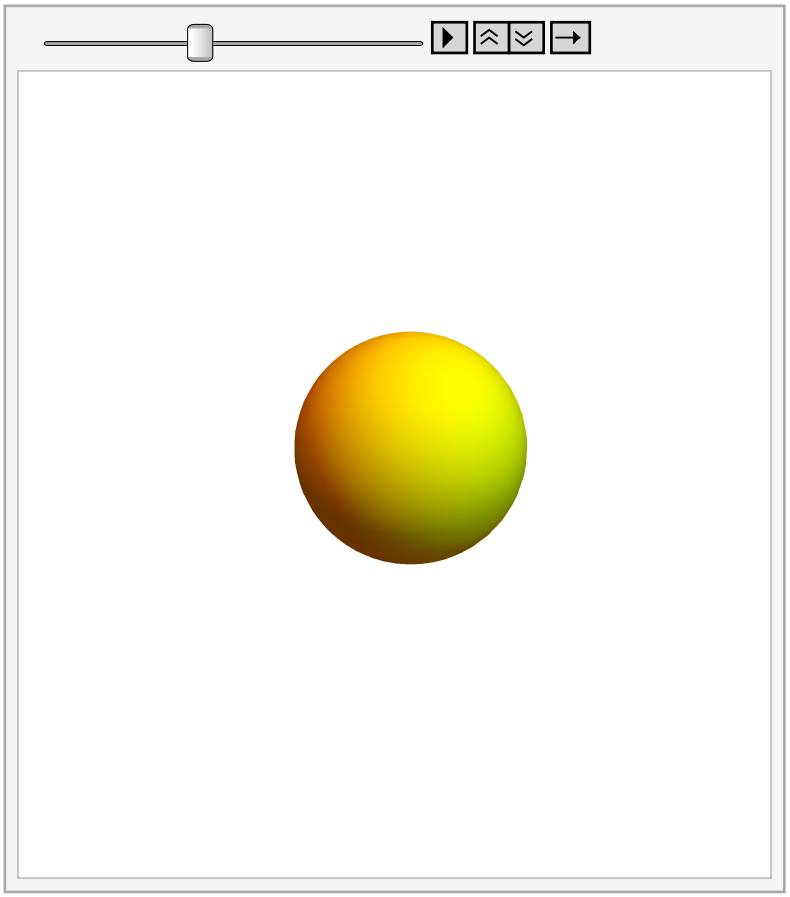
3. ListAnimate

ListAnimate[{expr1, expr2}]　依次运行图形表达式序列 **{expr1, expr2}**, 产生动画效果

例9：变换的球.

In[12]:=
```
A = {Red, Blue, Yellow, Green, Orange, Gray};
ListAnimate[Table[Graphics3D[
    {A[[n]], Sphere[{0, 0, 0}, n]}, PlotRange -> 6, Boxed → False], {n, 6}]]
```

Out[13]=



例10：逐点画出函数图

In[8]:=
```
data = Table[{i, Sin[i]}, {i, 0, 2.1 Pi, 0.075 Pi}];
ListAnimate[Table[ListLinePlot[Take[data, i],
    Mesh -> All, PlotRange -> {{0, 6.5}, {-1, 1}}], {i, Length[data]}]]
```