下载与安装

- MMA14 版本下载 BitTorrent种子
- 在线激活：针对MMA14

基本应用

1. 计算器

```
Print[2+2]
```

2. 素因子分解 $2^{2^n}+1$

```
Print[FactorInteger[Table[2^2^n + 1, {n, 6}]]]
```

3. 验证欧拉公式

```
Print[E^(Pi I)+1==0]
```

4. 计算π的前200位

```
Print[N[Pi, 200]]
```

5. 计算展开式

```
Print[Expand[(a + b)^3]]
```

6. 因子分解

```
Print[Factor[x^3 + y^3 + z^3 - 3 x y z]]
```

7. 求解一元三次方程的根

```
Print[Solve[x^3 - 2 x - 1 == 0, x]]
```

8. 计算极限

```
Print[Limit[(Tan[x] - x)/(x - Sin[x]), x -> 0]]
```

9. 计算导数

```
Print[D[x^x, x]]
```

10. 计算不定积分

```
Print[Integrate[x^2 Cos[x], x]]
```

11. 计算无穷函数项级数的和

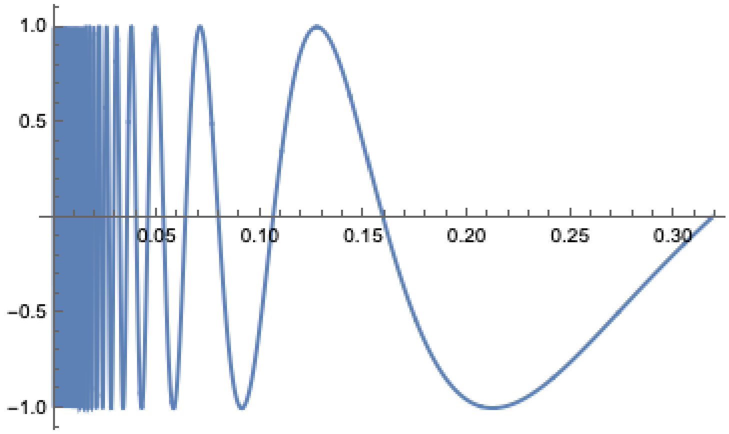
```
Print[Sum[x^(2 n)/(n^2 Binomial[2 n,
n]), {n, Infinity}]]
```

12. 求解常微分方程的通解

```
Print[FullSimplify[DSolve[y''[x] +
y[x] == 8 x Sin[x], y[x], x]]]
```

13. 画图$y=\sin(1/x)$

```
f=Plot[Sin[1/x], {x, 0, 1/Pi},
PlotPoints -> 1000];
Export["sin1x.png", f];
```



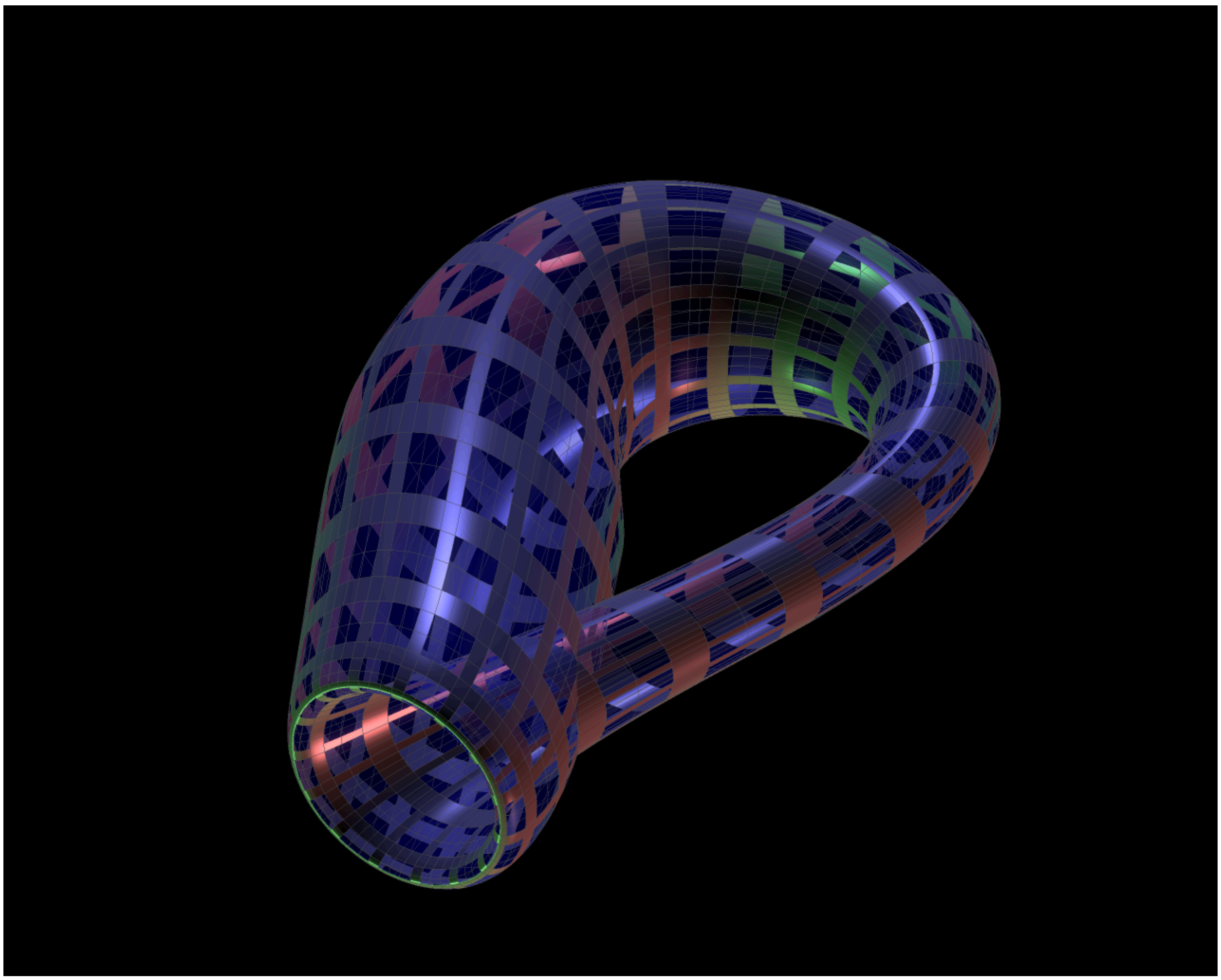
14. 可视化克莱因瓶

```
(* A Stylized Klein Bottle. Created
by Jeff Bryant  *)

bx = 6*Cos[u]*(1 + Sin[u]); by =
16*Sin[u]; rad = 4*(1 - Cos[u]/2);
X = If[Inequality[Pi, Less, u,
LessEqual, 2*Pi], bx + rad*Cos[v +
Pi],
    bx + rad*Cos[u]*Cos[v]];
Y = If[Inequality[Pi, Less, u,
LessEqual, 2*Pi], by,
   by + rad*Sin[u]*Cos[v]];
Z = rad*Sin[v];
o = 0.2; col1 = Blue; col2 = Gray;
darklights = {{"Directional",
RGBColor[0.5, 0.5, 1],
    ImageScaled[{0, 1, 0}]},
        {"Directional", RGBColor[1,
0.5, 0.5],
    ImageScaled[{1, -1, 0}]},
{"Directional", RGBColor[0.5, 1,
0.5],
    ImageScaled[{-1, -1, 0}]}};
gr = ParametricPlot3D[{X, Y, Z}, {u,
0, 2*Pi}, {v, 0, 2*Pi},
```

```
  PlotPoints -> {48, 12}, Axes ->
False, Boxed -> False, Mesh -> 59,
  MeshShading -> {{{col1,
Opacity[o], Specularity[White,
128]}, {col1,
     Opacity[o],
Specularity[White, 128]}, {col2,
     Specularity[White, 128]}},
{{col1, Opacity[o],
     Specularity[White, 128]},
{col1, Opacity[o],
     Specularity[White, 128]},
{col2,
     Specularity[White, 128]}},
{{col1, Opacity[o],
     Specularity[White, 128]},
{col1, Opacity[o],
     Specularity[White, 128]},
{col2,
     Specularity[White, 128]}},
{{col2,
     Specularity[White, 128]},
{col2,
     Specularity[White, 128]},
{col2, Specularity[White, 128]}}},
  MeshStyle -> GrayLevel[.3],
ImageSize -> {1280, 1024},
```

```
    MeshFunctions -> {#4 &, #5 &},
  Background -> Black,
    Lighting -> darklights,
  SphericalRegion -> True,
    ViewAngle -> \[Pi]/12];
    Export["KleinBottle.png", gr];
```



15. 作曲

```
s = Sound[
SoundNote[##, "Piano"] & @@@
Transpose[{{"B", "B", "C5", "D5",
"D5", "C5", "B", "A", "G", "G",
   "A", "B", "B", "A", "A", "B",
"B", "C5", "D5", "D5", "C5", "B",
    "A", "G", "G", "A", "B", "A",
"G", "G", "A", "A", "B", "G",
   "A", "B", "C5", "B", "G", "A",
"B", "C5", "B", "A", "G", "A",
   "D", "B", "B", "B", "C5", "D5",
"D5", "C5", "B", "A", "G", "G",
    "A", "B", "A", "G", "G"}, {0.5,
0.5, 0.5, 0.5, 0.5, 0.5, 0.5,
   0.5, 0.5, 0.5, 0.5, 0.5, 0.75,
0.25, 1, 0.5, 0.5, 0.5, 0.5,
   0.5, 0.5, 0.5, 0.5, 0.5, 0.5,
0.5, 0.5, 0.75, 0.25, 1, 0.5,
   0.5, 0.5, 0.5, 0.5, 0.25, 0.25,
0.5, 0.5, 0.5, 0.25, 0.25, 0.5,
    0.5, 0.5, 0.5, 0.5, 0.5, 0.5,
0.5, 0.5, 0.5, 0.5, 0.5, 0.5,
   0.5, 0.5, 0.5, 0.5, 0.5, 0.75,
0.25, 1}}]];
```

```
s // EmitSound;
Export["music.mp3", s]
```

0:00 / 0:32

16. 哈勃望远镜的太空图

    - 图片来源: https://science.nasa.gov/wp-content/uploads/2023/04/m74-xlarge_web-jpg.webp

```
M74 = Import["m74-xlarge_web-jpg.webp.jpeg"];
(*转换为油画*)
img=ImageEffect[M74, {"OilPainting", 6}];
Export["M74.png", img];
```

17. 寻找

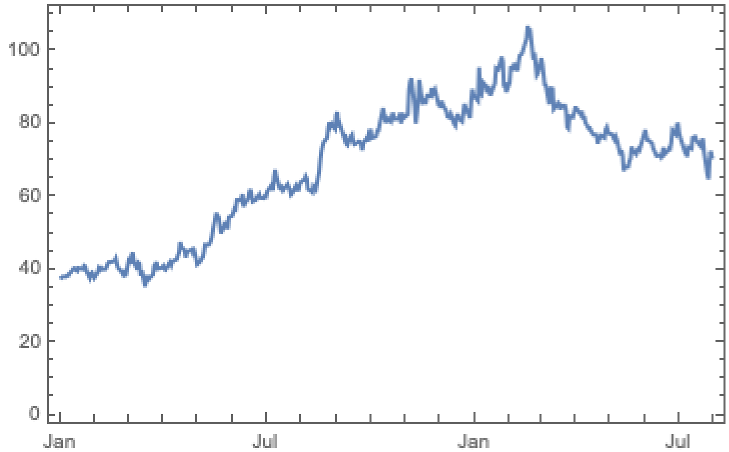
    - 图片来源: http://farm1.staticflickr.com/35/103000621_bcaee4a234.jpg

```
Grothendieck =
Import["http://farm1.staticflickr.com/35/103000621_bcaee4a234.jpg"]
face=HighlightImage[Grothendieck,
FindFaces[Grothendieck][[2]]]
Export["Grothendieck.png", face];
```

```
HighlightImage[$Failed, FindFaces[$Failed][2]]
```

18. 京东股票数据

```
(* 实时股价 *)
Print[FinancialData["JD"]]
(* 2020.01.01-2021.01.31 *)
Export["JD2020-2021.png",
DateListPlot[FinancialData["JD",
{"Jan. 1, 2020", "Jul. 31, 2021"}]]]
```



19. 假设$f(x) = 1 + x + x^2 + x^3 + x^4 + x^5$, 试求$f(x)$在$x = -1$处的Taylor展开式。

```
f[x_] := Sum[x^k, {k, 0, 5}]
Print[f[x]]
g=Series[f[x], {x, -1, 6}];
Print[Normal[g]]
```

基础知识

原子(atom)

以下三类对象被称为原子（Atom）：

- **符号（Symbol）**：由字母和数字组成的有限序列，其中数字不能作为起始字符。符号可以理解为变量的名称。例如，`x1` 是一个符号（变量名）。

- **数字（Number）**：包括整数、有理数、实数和复数，通常作为变量的值。例如，`42`、`3/7`、`π` 和 `2 + 3I`。
- **字符串（String）**：由双引号 `"` 括起来的任意字符序列，也可以作为变量的值。例如，`"Hello, world!"`。

```
x1=2+3*I;
Print[Sin[x1]//N];
```

内建符号

Mathematica系统内建符号的特点：

- **Camel命名法**：符号通常由第一个字母大写的单词组成，例如：True、False、FactorInteger、SetAttributes。
- **判断函数命名**：用于判断的函数名末尾通常带有“Q”，如：EvenQ、PrimeQ、MatchQ。
- **人名命名**：某些符号以人名为基础，格式为人名加符号名，例如：EulerGamma、BesselJ、DiracDelta。

因此，在命名自定义符号时，建议避免与内建符号冲突。一种有效的方法是使用小写字母开头的camel命名法。

条件与循环

1. 条件判断

```
rollDie[] := Module[{result},
result = RandomInteger[{1, 6}];
If[result == 1,
    Print["You rolled a 1, what a
bad start!"],
    If[result == 6,
        Print["You rolled a 6, what
a lucky start!"],
        Print["You rolled a ",
result, ", keep going!"]
    ]
]
]
rollDie[]  (* Call the function to
simulate rolling a die *)
```

2. 多重条件判断

```
numberGuessingGame[] :=
Module[{target = RandomInteger[{1,
100}], guess, difference},
  Print["Welcome to the Number
Guessing Game! Guess a number
between 1 and 100."];

  While[True,
    guess = Input["Enter your guess:
"];
    difference = Abs[target -
guess];

    Which[
      difference == 0,
        Print["Congratulations!
You've guessed the right number: ",
target, "!"];
        (* Exit the loop when
guessed correctly *)
        Break[],
      difference <= 5,
        Print["Very close! You're
within 5."],
      difference <= 15,
```

```
          Print["Close! You're within
15."],
        difference <= 30,
          Print["Not too far! You're
within 30."],
        True,
          Print["Way off! Try again!"]
      ];
    ]
  ]
(* Start the game *)
numberGuessingGame[]
```

`Which`是多个条件，而如果只有一个表达式，需要根据该表达式匹配不同的情形，则使用`Switch`:

```
animal = "Dog";
description = Switch[
    animal,
    "Dog", "A dog is a loyal
companion.",
    "Cat", "A cat is independent and
curious.",
    "Bird", "A bird can sing and
fly.",
    _, "Unknown animal."
]
```

3. 循环

```
countdownGame[start_Integer] :=
Module[{event, i},
  Print["Starting countdown from ",
start, "!"];

  For[i = start, i > 0, i--,
    event = RandomChoice[{"Nothing
happens.", "You find a treasure!",
"A monster appears!", "You gain a
bonus point!"}];
    Print[i, ": ", event];
    Pause[1];  (* Pause for a second
for dramatic effect *)
  ];

  Print["Blast off!"];
]

countdownGame[10]  (* Start
countdown from 10 *)
```

参考资料

- 清华刘思齐: [链接]
- Wolfram U: [链接] 可以获得证书，有你名字

用MMA模拟三体运行

假设有三个天体, 其质量为:

```
m = 14982844643 {1, 1, 1};
```

初始位置处于正三角形:

```
p0 = {{-0.5, 0, 0}, {0.5, 0, 0}, {0, 0,
2 Sqrt[3]/2}};
```

初始速度为单位速度，分别为:

```
v0 = Normalize /@ {{0, -1, 0}, {0, 1,
0}, {1, 1, 1}};
```

根据牛顿运动定律，我们知道三体的位置满足如下常微分方程组:

```
D[p[k],{t,2}] = \sum_{i\neq k}G m[i]
\frac{p[i]-p[k]}{\lvert p[i]-p[k]
\rvert^3}.
```

利用MMA的数值计算，容易解得如下运行图，点击运行：三体。

完整代码如下：

```
Manipulate[Module[{v0, p0, m, gravConst,
p, eq, res, pres},
  (*Initial velocities*)
  v0 = Normalize /@ {{0, -1, 0}, {0, 1,
0}, {1, 1, 1}};
  (*Initial positions*)
  p0 = {{-0.5, 0, 0}, {0.5, 0, 0}, {0,
0, 2 Sqrt[3]/2}};
  (*Masses*)
  m = 14982844643 {1, 1, 1};
  (*Gravitational constant*)
  gravConst = 6.67430*10^(-11);
  (*Positions as functions of time*)
  p = {{x1[t], y1[t], z1[t]}, {x2[t],
y2[t], z2[t]}, {x3[t], y3[t], z3[t]}};
  (*Equations of motion*)
  eq[i_, j_, k_, t_] :=
   gravConst*(m[[i]]*(p[[i]] -
p[[k]])/Norm[p[[i]] - p[[k]]]^3 +
      m[[j]]*(p[[j]] -
p[[k]])/Norm[p[[j]] - p[[k]]]^3);
  (*Solving the differential equations*)
  res = NDSolve[Flatten[{
      Thread[D[p[[1]], {t, 2}] == eq[3,
2, 1, t]],
```

```
       Thread[D[p[[2]], {t, 2}] == eq[1,
3, 2, t]],
       Thread[D[p[[3]], {t, 2}] == eq[1,
2, 3, t]],
       Thread[(p /. t :> 0) == p0],
       Thread[(D[p, t] /. t :> 0) ==
v0]}], p, {t, 0, tmax}][[1]];
  (*Extracting positions at final time*)
  pres = p /. res;
  (*Creating points for each mass*)
  points =
   Graphics3D[{PointSize[Large],
     Table[{ColorData[1, i],
Point[pres[[i]] /. t -> tmax],
       Point[pres[[i]] /. t -> 0]}, {i,
3}]}];
  (*Creating trajectories*)
  trajectories =
   ParametricPlot3D[pres, {t, 0, tmax},
    PlotStyle -> Table[ColorData[1, i],
{i, 3}],
    PlotRange -> {{-1, 4}, {-2, 4}, {0,
4}},
    AxesLabel -> {"X", "Y", "Z"},
PlotLegends -> {"1", "2", "3"}];
  (*Displaying trajectories and points*)
  Show[trajectories, points, AxesLabel -
```

```
> {"X", "Y", "Z"}]],
 (*Manipulate control*)
 {{tmax, 0.01}, 0, 14, AnimationRate ->
0.5, Appearance -> "Open"}
]
```

R语言

[R教程]