

# Epicycles研究笔记 1

假设  $cs = \{c_0, c_1, c_{-1}, c_2, c_{-2}, c_3, \dots, c_K, c_{-K}\}$ , 计算向量 and 的列表

方法一:

```
In[*]:= vectorList[cs_] :=  
  Accumulate@Table[cs[[n]] Exp[I If[EvenQ@n,  $\frac{n}{2}$ ,  $-\frac{n-1}{2}$ ]] I t], {n, Length@cs}]  
  [累加] [表格] [⋯] [⋯] [偶数判定] [虚数单位] [长度]
```

方法二:

```
In[*]:= vectorList2[cs_] :=  
  Table[Sum[cs[[n]] Exp[I If[EvenQ@n,  $\frac{n}{2}$ ,  $-\frac{n-1}{2}$ ]] I t], {n, r}], {r, Length@cs}]  
  [表格] [求和] [⋯] [⋯] [偶数判定] [虚数单位] [长度]
```

测试结果

```
In[*]:= testCS = {2 EI π/3, EI π/2, 0.8 E-I π/6}  
vectorList[testCS]  
vectorList2[testCS]
```

```
Out[*]=  
{2 e $\frac{i \pi}{3}$ , i, 0.69282 - 0.4 i}
```

```
Out[*]=  
{2 e $\frac{i \pi}{3}$ , 2 e $\frac{i \pi}{3}$  + i ei t, 2 e $\frac{i \pi}{3}$  + (0.69282 - 0.4 i) e-i t + i ei t}
```

```
Out[*]=  
{2 e $\frac{i \pi}{3}$ , 2 e $\frac{i \pi}{3}$  + i ei t, 2 e $\frac{i \pi}{3}$  + (0.69282 - 0.4 i) e-i t + i ei t}
```

测试性能

```
In[*]:= testCS2 = RandomComplex[{-3 - 3 I, 3 + 3 I}, 800];  
  [伪随机复数] [虚数单位] [虚数单位]
```

```
In[*]:= Timing@vectorList[testCS2]  
[计算时间]
```

```
Out[*]=
```

```
{0.125, {-2.2485 - 1.66347 i, (-2.2485 - 1.66347 i) - (... 18 ... + ... 1 ... ) ... 1 ... ,  
  (... 796 ... , ... 1 ... , ... 1207 ... + ... 1 ... + (1.82668 + 2.6294 i) e400 i t}}
```

大型输出

显示更少

显示更多

显示全部

设定大小限制...

```
In[ ]:= Timing@vectorList2[testCS2]
```

计算时间

```
Out[ ]:=
```

```
{1.75, {-2.2485 - 1.66347 i, (-2.2485 - 1.66347 i) - (... 18 ... + ... 1 ... ) ... 1 ... ,
... 796 ... , ... 1 ... , ... 1207 ... + ... 1 ... + (1.82668 + 2.6294 i) e^{400 i t}}}
```

大型输出

显示更少

显示更多

显示全部

设定大小限制...

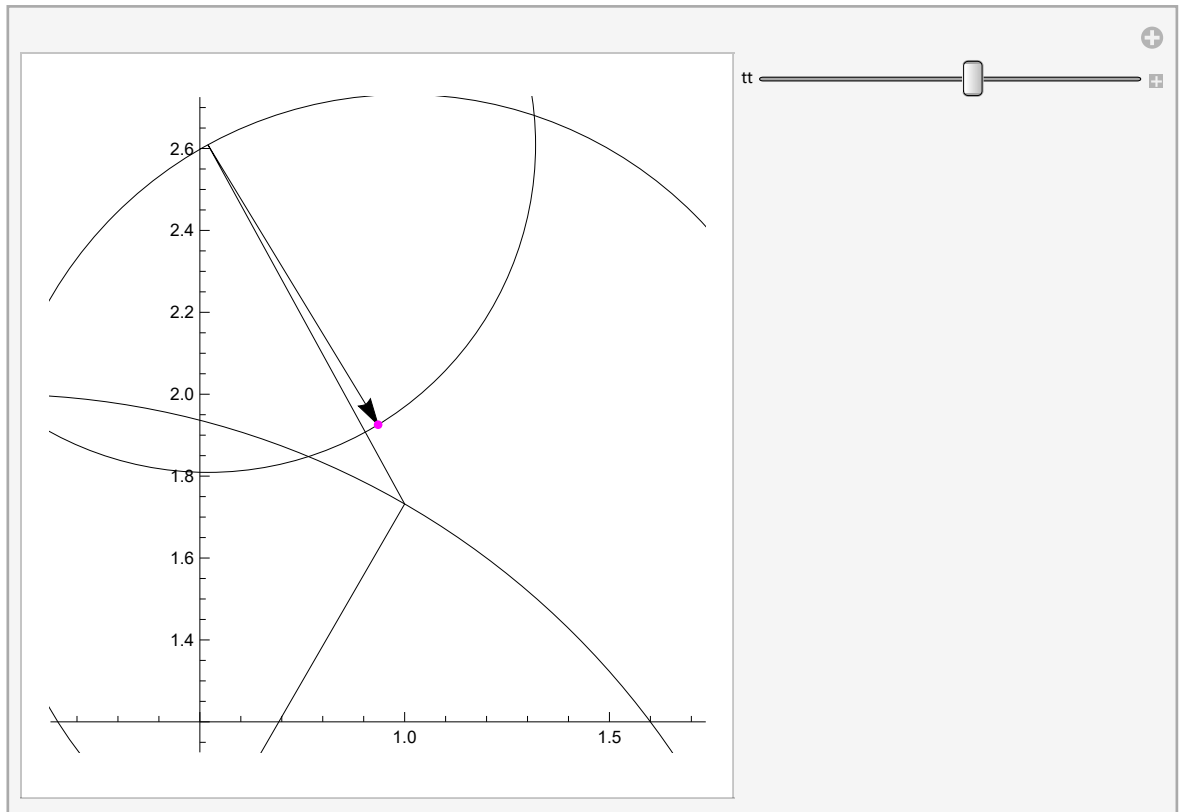
从结果可以看出，方法二性能不如方法一

因为随着输入列表长度的增加，方法二会出现越来越多的重复计算，性能差距拉大。

计算所有点，线，圆数据，即在上面计算的基础上，将复指数转为平面坐标，并计算对应圆的半径

```
In[ ]:= epicycles[cs_] := DynamicModule[{len, ps, path, circles, line}, len = Length@cs;
                                         |动态模块                                     |长度
    ps = ReIm@vectorList[cs];
                                         |实部虚部列表
    circles =
      Table[{ps[[r]], Norm@cs[[r + 1]]}, {r, len - 1}] ~Prepend~ {{0, 0}, Norm@cs[[1]]};
                                         |表格 |模 |加在前面 |模
    line = ps ~Prepend~ {0, 0};
                                         |加在前面
    Manipulate[path = Table[Last@ps, {t, 0, 2 π, 0.01}];
               |交互式操作 |表格 |最后一个
    Graphics[{{Thin, Circle@@@ (circles /. t -> tt)}, {Arrow@(line /. t -> tt)},
             |图形 |细 |圆 |箭头
             {Thick, Pink, Line@Take[path, Min[tt / 0.01 + 1, Length@path]]},
             |粗 |粉色 |线段 |选取 |最小值 |长度
             {PointSize[Medium], Magenta, Point@(Last@ps /. t -> tt), Point@{0, 0}}},
             |点的大小 |中 |品红色 |点 |最后一个 |点
    PlotRange -> (MinMax[path[[All, #]], Norm@Last@cs] & /@ {1, 2}),
    |绘制范围 |最小最大值列表 |全部 |模 |最后一个
    Axes -> True], {tt, 0, 4 π, 0.01}]]
    |坐标轴 |真
```

In[ ]:= **epicycles[testCS]**  
 Out[ ]:=



In[ ]:= **epicycles**[**RandomComplex**[{-3 - 3 I, 3 + 3 I}, 5] ~ Join ~ **RandomComplex**[{-1 - I, 1 + I}, 5]]

    [伪随机复数]      [虚数单位] [虚数单位] [连接]    [伪随机复数]      [虚…]    [虚数单位]

Out[ ]:=

