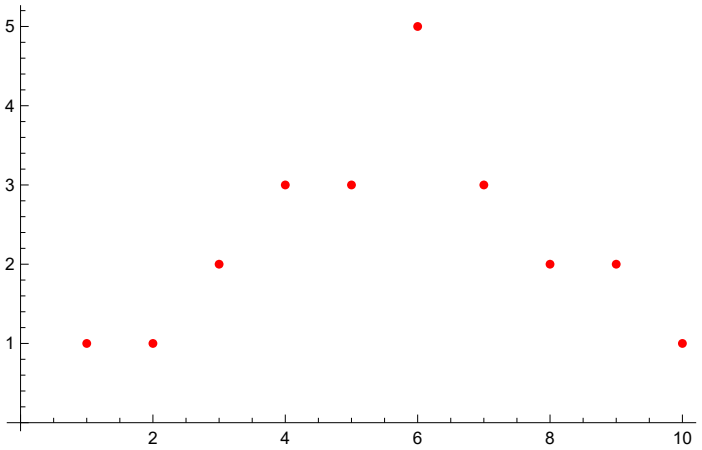
离散傅里叶变换与傅里叶级数

以之前任意构造的数列为例

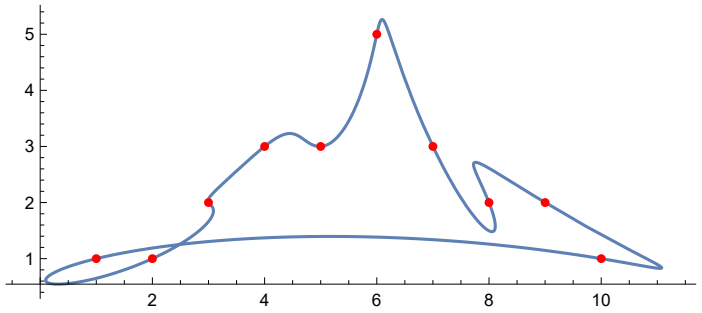
In[ ]:=
```
l = {1, 1, 2, 3, 3, 5, 3, 2, 2, 1};
len = Length[l];
```
长度
```
lp = ListPlot[l, PlotStyle → Red]
```
绘制点集　绘图样式　红色

Out[ ]=



In[ ]:=
```
pts = Table[{r, l[[r]]}, {r, len}];
```
表格
```
f = Fourier[Complex @@@ pts] / √len;
```
傅立叶　复数
```
p = Sum[f[[Mod[n + len, len] + 1]] Exp[n I t], {n, -5, 4}];
```
求和　模余　指…　虚数单位
```
Show[ParametricPlot[ReIm@p, {t, 0, 2 π}], lp, PlotRange → All]
```
显示　绘制参数图　实部虚部列表　绘制范围　全部

Out[ ]=

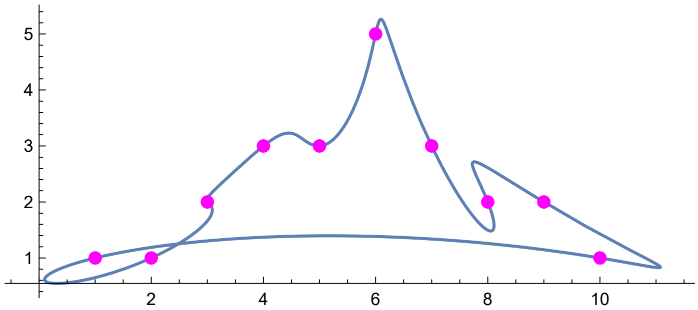


将上面思路包装成函数

In[∘]:=
```
fPlot[l_] := Block[{len, k, f, p}, len = Length@l;
  k = Ceiling[len / 2];
  f = Fourier[Complex @@@ Table[{r, l[[r]]}, {r, len}]] / Sqrt[len];
  p = Sum[f[[Mod[n + len, len] + 1]] Exp[n I t], {n, k - len, k - 1}];
  Show[ParametricPlot[ReIm@p, {t, 0, 2 π}],
   ListPlot[l, PlotStyle → {Magenta, PointSize[Large]}], PlotRange → All]]
```
块 长度 向上取整 傅立叶 复数 表格 求和 模余 指… 虚数单位 显示 绘制参数图 实部虚部列表 绘制点集 绘图样式 品红色 点的大小 大 绘制范围 全部

```
fPlot[l]
```

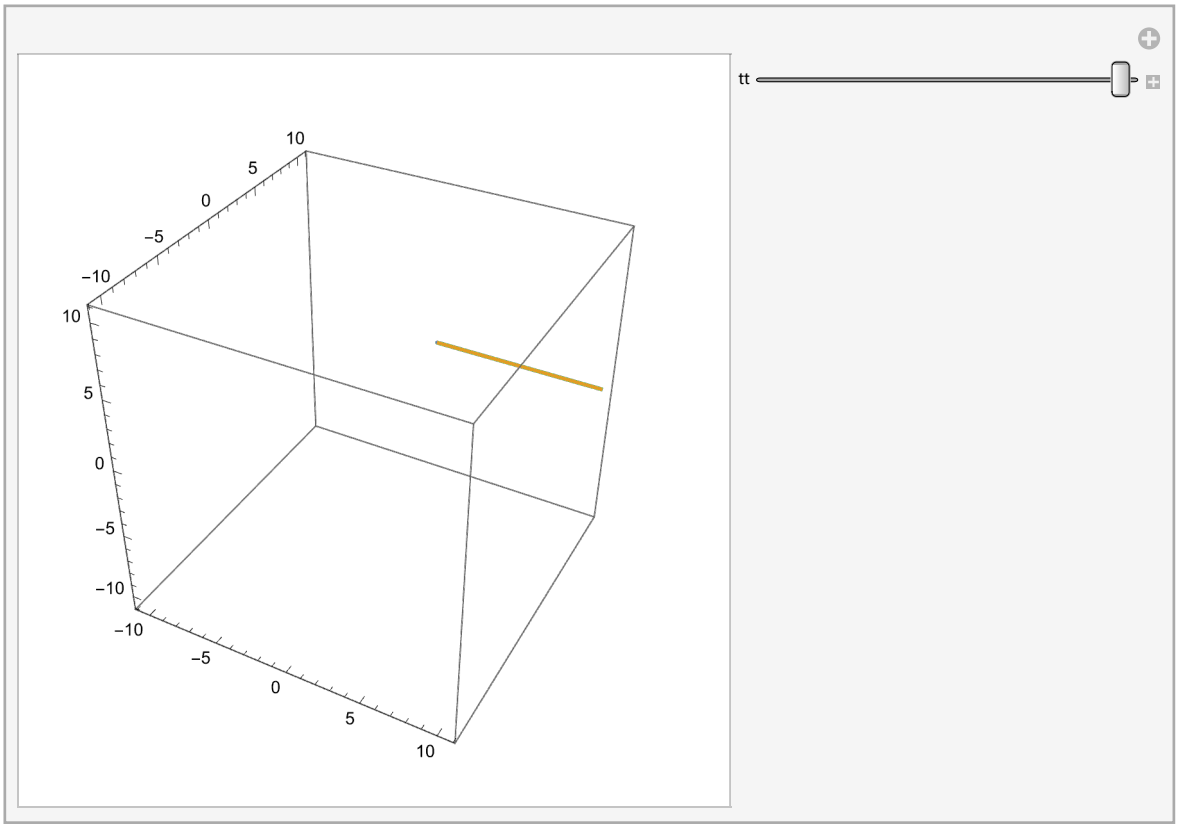
Out[∘]=



In[ ]:=

```
fPlot3d[l_] := DynamicModule[{len, k, f, p}, len = Length@l;
                动态模块                      长度

  k = Ceiling[len / 2];
      向上取整

  f = Fourier[Complex @@@ Table[{r, l[[r]]}, {r, len}]] / Sqrt[len];
      傅立叶   复数          表格

  p = Sum[f[[Mod[n + len, len] + 1]] Exp[n I t], {n, k - len, k - 1}];
      求和   模余                  指…   虚数单位

  Manipulate[Show[ParametricPlot3D[
  交互式操作      显示  绘制三维参数图

      {Prepend[ReIm@p, 0], Prepend[ReIm@p, t]}, {t, 0, tt}], (*ListPlot[l,
       加在前面  实部虚部列表  加在前面  实部虚部列表                  绘制点集

      PlotStyle→{Magenta,PointSize[Large]}],*)PlotRange → 10], {tt, 0.01, 4 π}]]

fPlot3d[l]
```

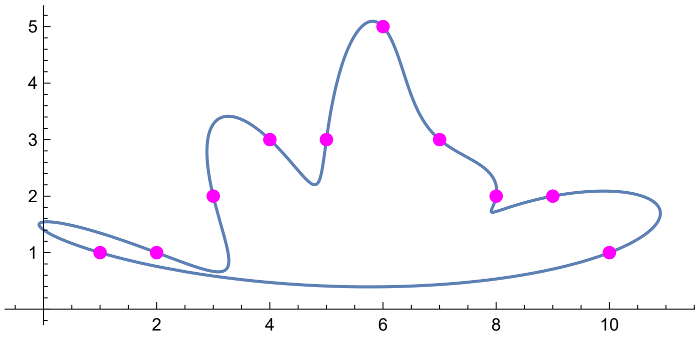
Out[ ]=



In[ ]:=

```
fPlot22[l_] := Block[{len, k, f, p}, len = Length@l;
块 长度
  k = Floor[len / 2];
  向下取整
  f = Fourier[Complex @@@ Table[{r, l⟦r⟧}, {r, len}]] / √len;
  傅立叶 复数 表格
  p = Sum[f⟦Mod[n + len, len] + 1⟧ Exp[n I t], {n, k - len + 1, k}];
  求和 模余 指… 虚数单位
  Show[ParametricPlot[ReIm@p, {t, 0, 2 π}],
  显示 绘制参数图 实部虚部列表
   ListPlot[l, PlotStyle → {Magenta, PointSize[Large]}], PlotRange → All]]
   绘制点集 绘图样式 品红色 点的大小 大 绘制范围 全部

fPlot22[l]
```

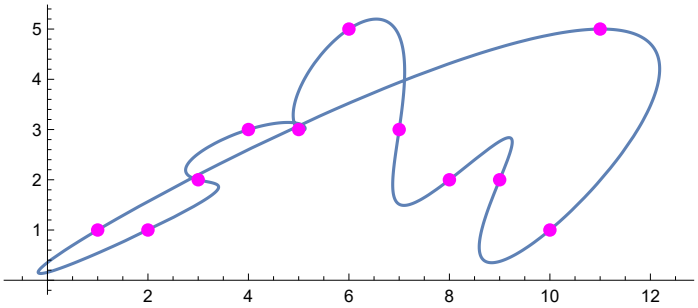
Out[ ]=



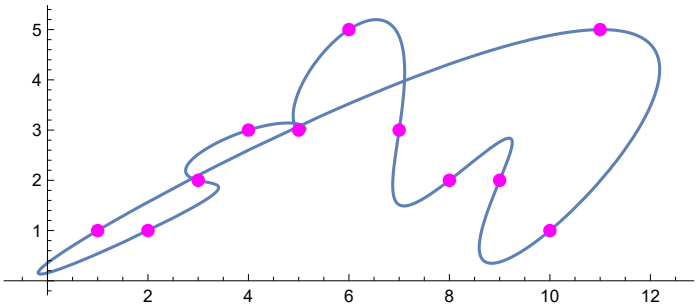
对比fPlot和fPlot22，差别在于求和范围{-5,4}和{-4,5}，换句话说当采样点的个数为偶数时，最中间的频率的相位是0或π。奇数点不存在这种差异，如下所示

In[ ]:=
```
fPlot[l~Append~5]
```
追加
```
fPlot22[l~Append~5]
```
追加

Out[ ]=



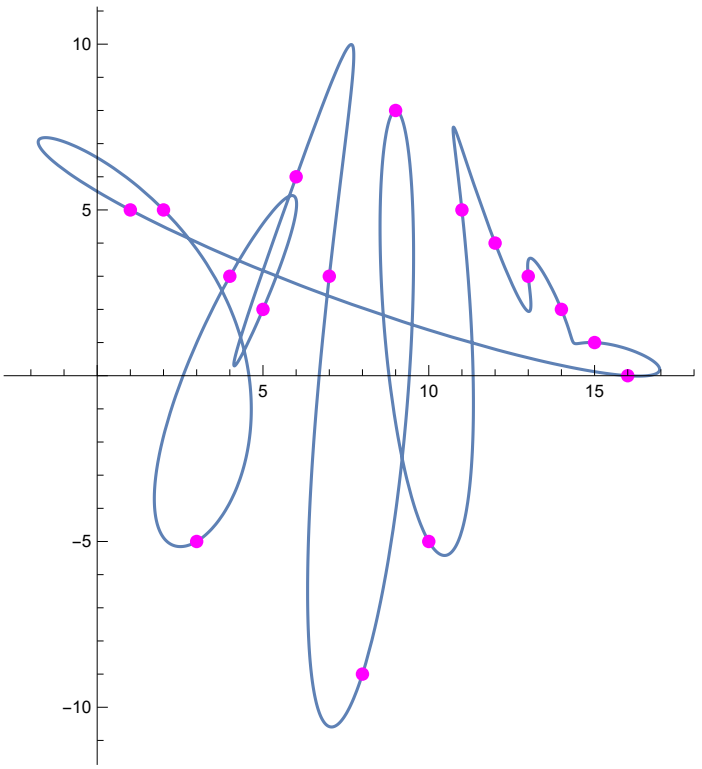
Out[ ]=



下面随便玩玩，感觉可以做个签名游戏

In[ ]:=
```
fPlot[{5, 5, -5, 3, 2, 6, 3, -9, 8, -5, 5, 4, 3, 2, 1, 0}]
```

Out[ ]=

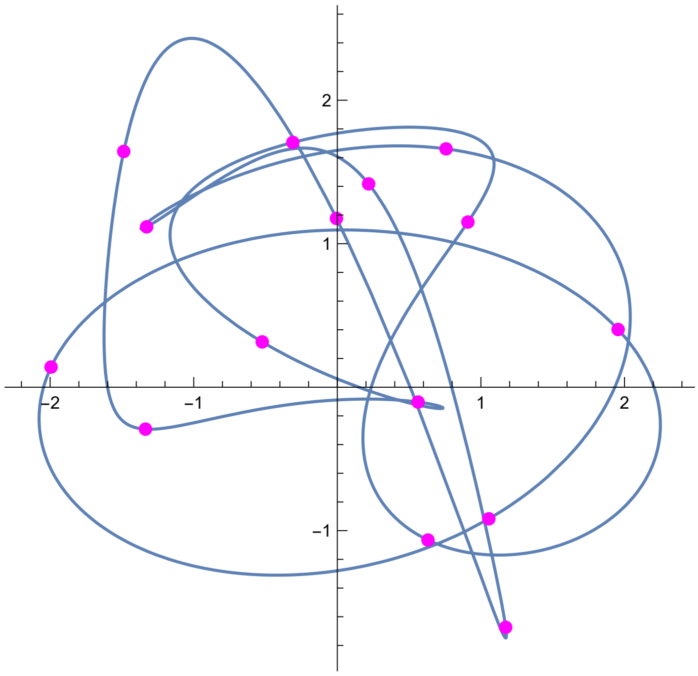


再扩展到更一般的情况，接受以复数表示的点的坐标列表，

In[∘]:=

```
ePlot[cs_] := Block[{len, k, f, p}, len = Length@cs;
               块                         长度
  k = Ceiling[len / 2];
      向上取整
  f = Fourier[cs] / √len ;
      傅立叶
  p = Sum[f[[Mod[n + len, len] + 1]] Exp[n I t], {n, k - len, k - 1}];
      求和  模余                      指…  虚数单位
  Show[ParametricPlot[ReIm@p, {t, 0, 2 π}],
  显示  绘制参数图      实部虚部列表
   ListPlot[ReIm@cs, PlotStyle → {Magenta, PointSize[Large]}], PlotRange → All]]
   绘制点集  实部虚…   绘图样式      品红色   点的大小   大       绘制范围    全部
ePlot[RandomComplex[{-2 - 2 I, 2 + 2 I}, 15]]
      伪随机复数          虚数单位 虚数单位
```

Out[∘]=



心里蹦出两个字“完美”。