

plot 関数群

青木繁伸

2020年3月17日

1 目的

図形描画関数を定義する。

1.1 begin()

プロットの最初に呼ぶ。

```
begin(aspect=None, box=False, xticks=[], yticks=[])
```

<code>aspect</code>	アスペクト比 (デフォルト <code>None</code> の場合はアスペクト比を設定しない)
<code>box</code>	デフォルト <code>False</code> で、軸を描かない
<code>xticks</code>	ティックマークの位置とラベル (デフォルト <code>[]</code> で描かない)
<code>yticks</code>	同上

1.2 end()

プロットの最後に呼ぶ。

```
end()
```

1.3 text()

文字列を描く。

```
text(x, y, label, **arg)
```

```
text2(x.y_l, **arg)
```

`x, y` が全て違う場合には `text2` の方が指定しやすい。

<code>x, y</code>	文字列を描く位置 (ベクトルまたはスカラー)
<code>label</code>	文字列 (ベクトルまたは単一文字列)
<code>x.y_l</code>	<code>x, y, label</code> の 3 要素を持つリスト (のリスト)
<code>arg</code>	以下で確認できる辞書型キーワード <pre>import matplotlib.pyplot help(matplotlib.text.Text)</pre>

```
import numpy as np
```

```

import os
import sys
sys.path.append("statlib")
from plot import begin, end, text, text2

begin(aspect=1)
text(1, 0.5, "test", fontsize=60)
arg = dict(fontsize=20, color="red", verticalalignment="center",
           horizontalalignment="center")
text([0,1,2, 0,1,2, 0,1,2], [0,0,0, 1,1,1, 2,2,2],
     ["bottom-left", "bottom", "bottom-right", "left", "center",
      "right", "top-left", "top", "top-right"], **arg)
text2([[0.1, 0.5, "abcdefg"], [0.5, 0.6, "text2"]], color="blue")
text2([0.2, 0.4, "one element"], color="brown")
end()

```

top-left

top

top-right

left

center

right

bottom-left

bottom

bottom-right

1.4 segment()

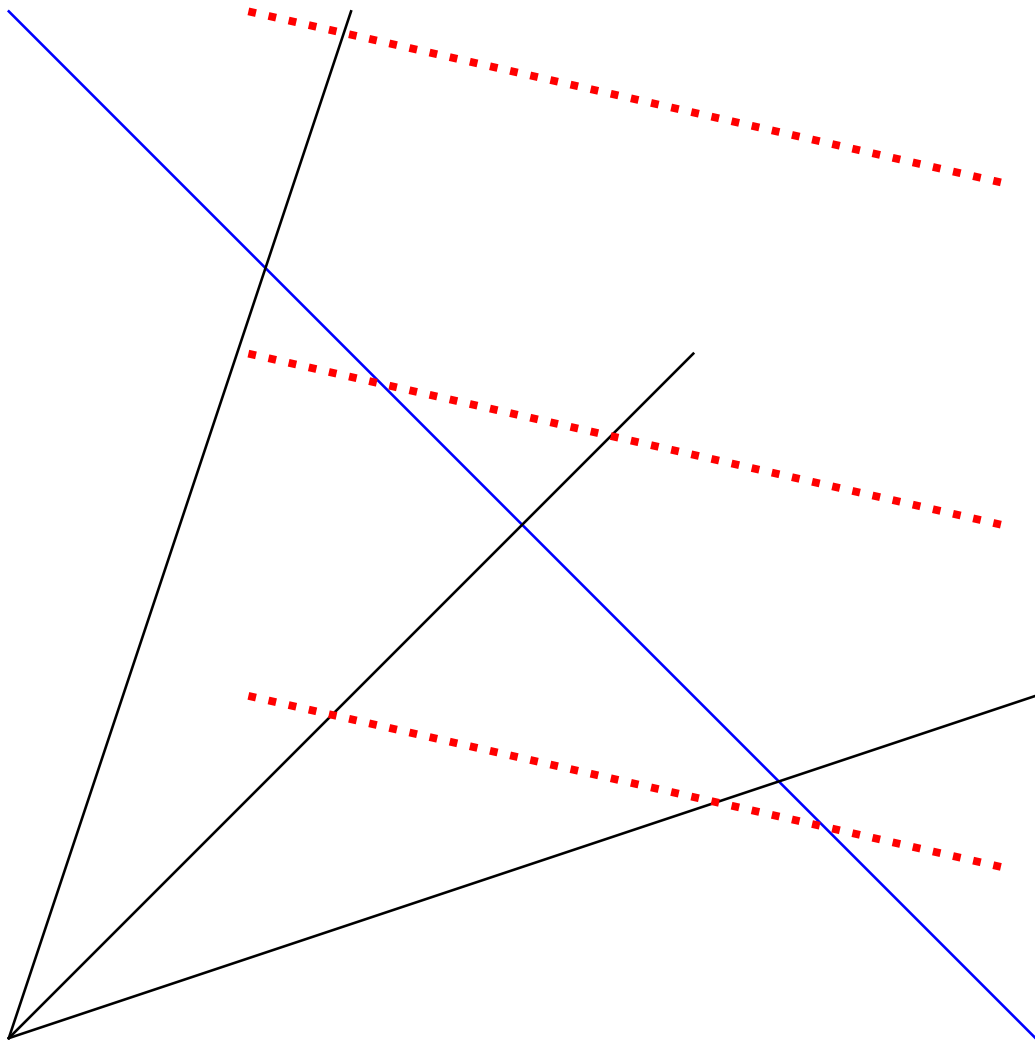
直線を描く。

```
segment(x1, y1, x2, y2, color="black", linewidth=0.5, linestyle="-", **arg)
```

x1, y1 直線の始点位置 (ベクトルまたはスカラー)
x2, y2 直線の終点位置 (ベクトルまたはスカラー)
color 直線の色 (デフォルトは黒)
linewidth 直線の幅 (デフォルトは 0.5)
linestyle 直線の種類 (デフォルトは実線)
arg 以下で確認できる辞書型キーワード

```
import matplotlib.pyplot  
help(matplotlib.pyplot.Line2D)
```

```
from plot import begin, end, segment  
  
begin(aspect=1)  
segment(0, 3, 3, 0, color="blue")  
segment([0,0,0], [0,0,0], [1, 2, 3], [3, 2, 1])  
segment(0.7, [1,2,3], 2.9, [0.5, 1.5, 2.5], color="red",  
        linewidth=1.5, linestyle=":")  
end()
```



1.5 line()

折線を描く。

```
line(x, y, color="black", linewidth=0.5, linestyle="-", **arg)
```

<code>x, y</code>	座標 (ベクトルまたはスカラー)
<code>color</code>	座標を順に結ぶ直線の色 (デフォルトは黒)
<code>linewidth</code>	直線の幅 (デフォルトは 0.5)
<code>linestyle</code>	直線の種類 (デフォルトは実線)
<code>arg</code>	以下で確認できる辞書型キーワード

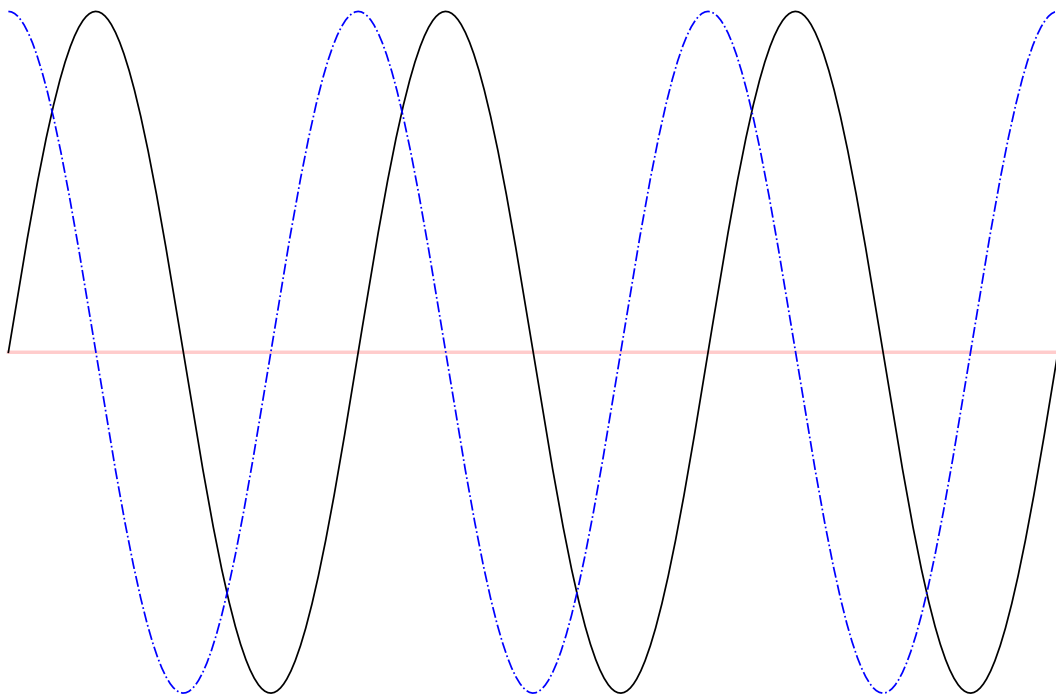
```
import matplotlib.pyplot
help(matplotlib.pyplot.Line2D)
```

```

from plot import begin, end, line

begin()
arg = dict(alpha=0.2)
line([0, 6*np.pi], [0, 0], color="red", linewidth=1, **arg)
x = np.linspace(0, 6*np.pi, 1000)
line(x, np.sin(x))
line(x, np.cos(x), color="blue", linestyle="-.")
end()

```



1.6 point()

マーカーを描く。

```
point(x, y, color="black", marker="o", size=1, **arg)
```

<code>x, y</code>	座標 (ベクトルまたはスカラー)
<code>color</code>	座標を描くマーカーの色 (デフォルトは黒)
<code>marker</code>	マーカーの種類 (デフォルトは○)
<code>size</code>	マーカーの半径 (デフォルトは1)
<code>arg</code>	以下で確認できる辞書型キーワード

```

import matplotlib.pyplot
help(matplotlib.pyplot.Line2D)

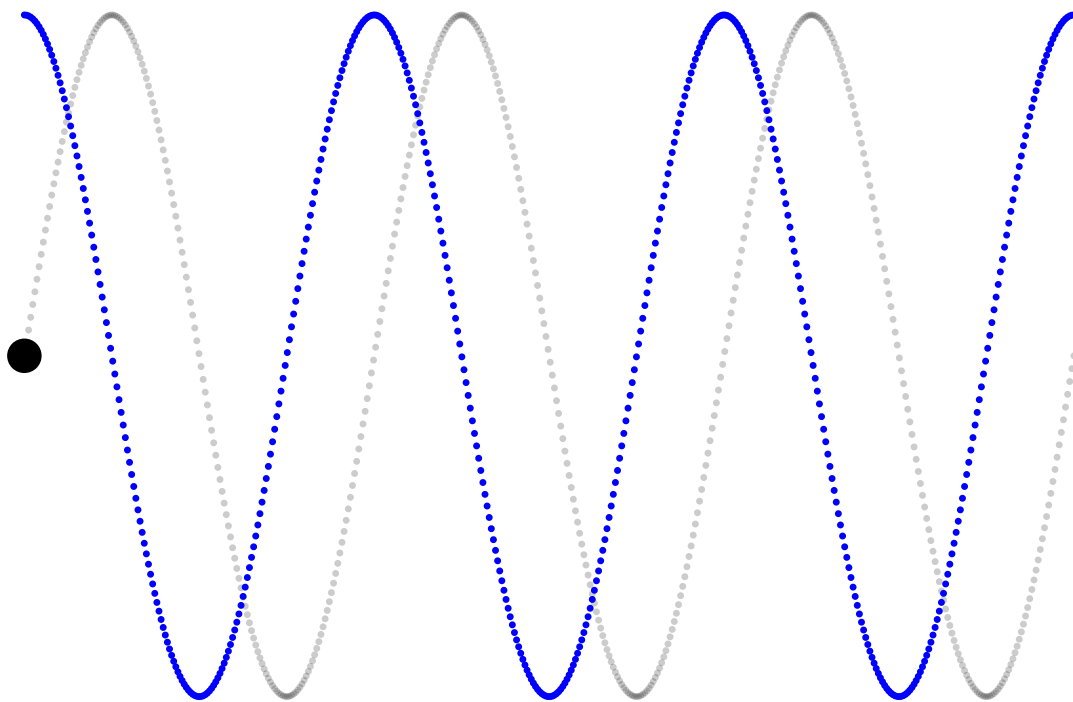
```

```

from plot import begin, end, point

begin()
arg = dict(alpha=0.2)
x = np.linspace(0, 6*np.pi, 500)
point(x, np.sin(x), **arg)
point(x, np.cos(x), color="blue")
point(0, 0, size=3)
end()

```



1.7 box()

長方形を描く。

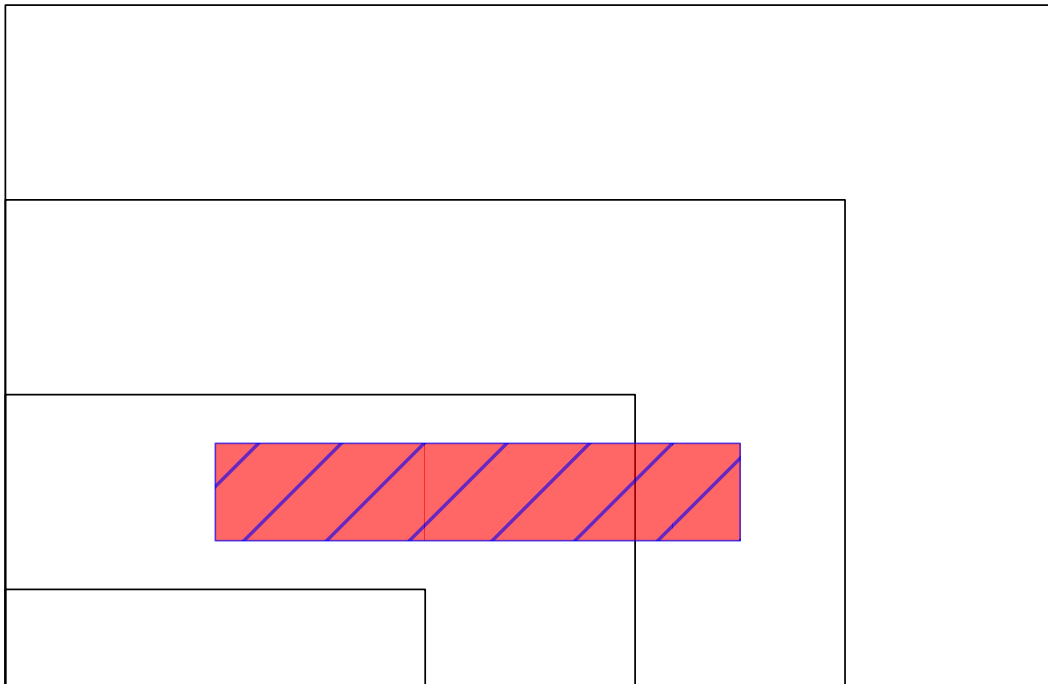
```
box(x1, y1, x2, y2, color="black", fill=False, linewidth=0.5, alpha = 1, hatch="", **arg)
```

<code>x1, y1</code>	長方形の左下隅の座標 (ベクトルまたはスカラー)
<code>x2, y2</code>	長方形の右上隅の座標 (ベクトルまたはスカラー)
<code>color</code>	長方形を描く線の色 (デフォルトは黒)
<code>fill</code>	長方形を塗りつぶすなら <code>True</code> を指定する (デフォルトは <code>False</code>)
<code>linewidth</code>	長方形を描く線の幅 (デフォルトは 0.5)
<code>linestyle</code>	長方形を描く線の種類 (デフォルトは実線)
<code>hatch</code>	長方形のハッチング (デフォルトはハッチングなし)
<code>alpha</code>	アルファチャンネル (デフォルトは 1)
<code>arg</code>	以下で確認できる辞書型キーワード

```
import matplotlib.pyplot
help(matplotlib.pyplot.Line2D)
```

```
from plot import begin, end, box

begin()
box(0, 0, [2,3,4,5], [1,3,5,7])
box(1, 1.5, 3.5, 2.5, color="red", fill=True, hatch="/", alpha=0.6)
end()
```



1.8 circle()

円を描く。

```
circle(ox, oy, r, begin=0, end=360, color="black", linewidth=0.5, linestyle="-")
```

ox, oy	円の中心座標 (ベクトルまたはスカラー)
r	半径 (ベクトルまたはスカラー)
fill	塗りつぶし円を描くときは True にする [デフォルトは False]
begin	描き始め (水平線からの角度。デフォルトは 0)
end	描き始め (水平線からの角度。デフォルトは 0)
color	円を描く線の色, 塗りつぶし円の場合はその色 (デフォルトは黒)
linewidth	円を描く線の幅 (デフォルトは 0.5) 塗りつぶし円の場合は無視される
linestyle	円を描く線の種類 (デフォルトは実線) 塗りつぶし円の場合は無視される

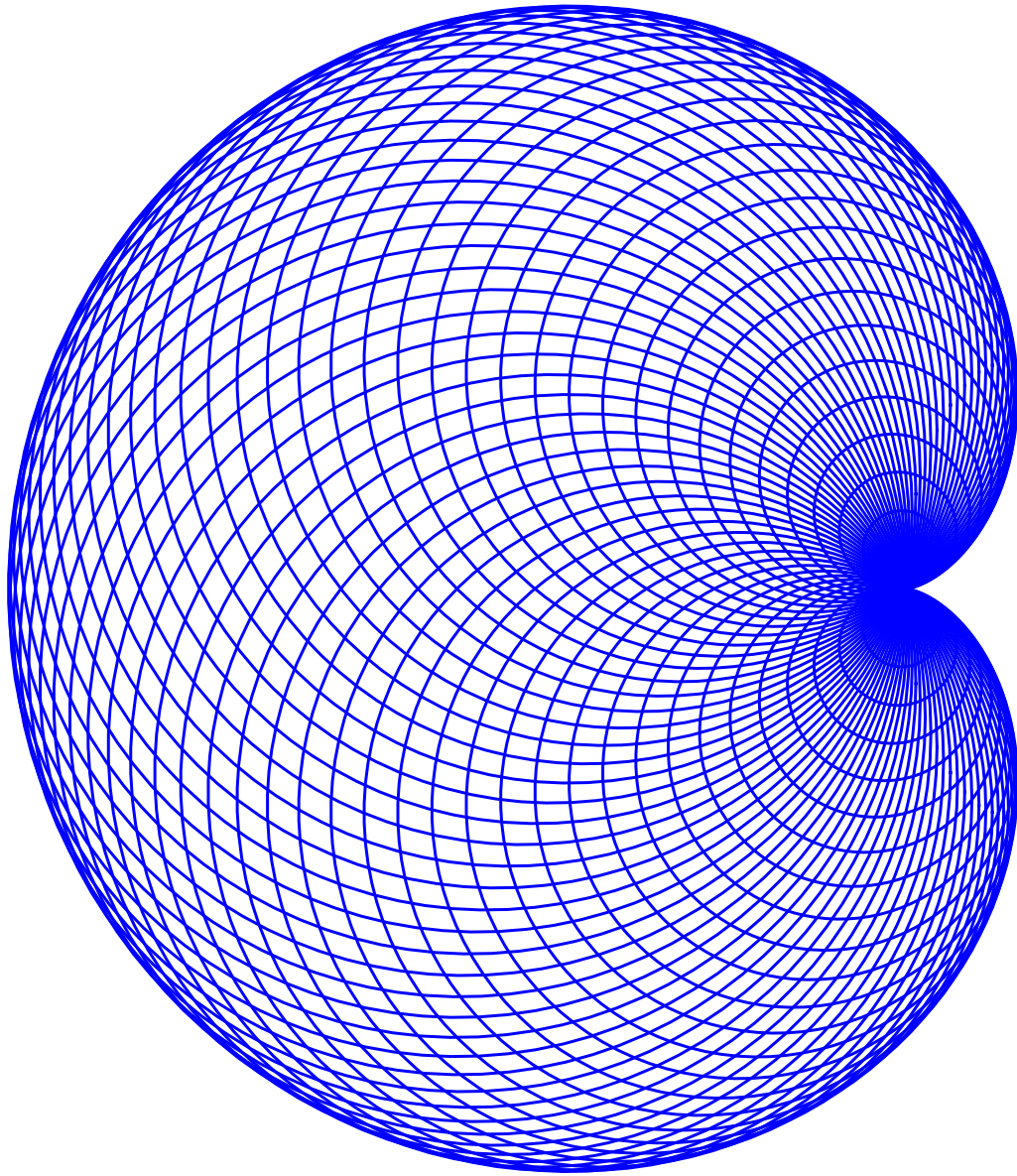
arg 以下で確認できる辞書型キーワード

```
import matplotlib.pyplot
help(matplotlib.pyplot.Line2D)
```

1.8.1 カージオイド

```
import numpy as np
from plot import begin, end, circle

begin(aspect=1)
t = np.arange(0, 360, 5)
l = 100
ox = l*np.cos(np.radians(t))
oy = l*np.sin(np.radians(t))
circle(ox+320, oy+250, np.sqrt((ox-ox[0])**2+(oy-oy[0])**2),
       color="blue")
end()
```

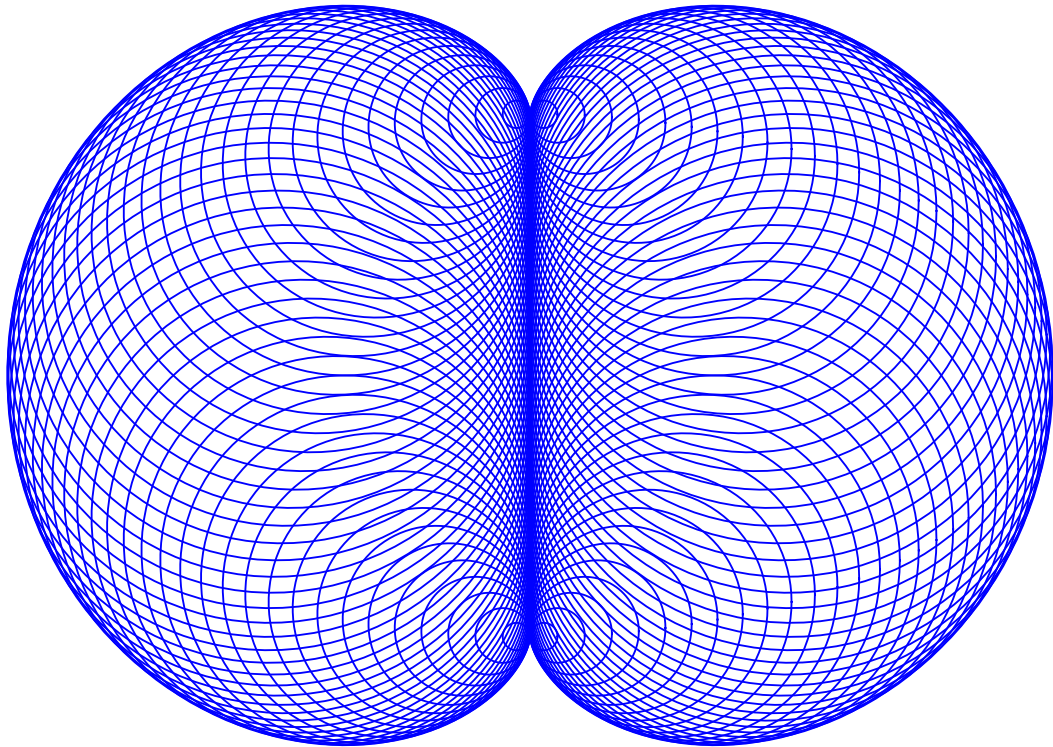



1.8.2 ネフロイド

```
import numpy as np
from plot import begin, end, circle

begin(aspect=1)
t = np.arange(0, 360, 3)
l = 130
ox = l*np.cos(np.radians(t))
```

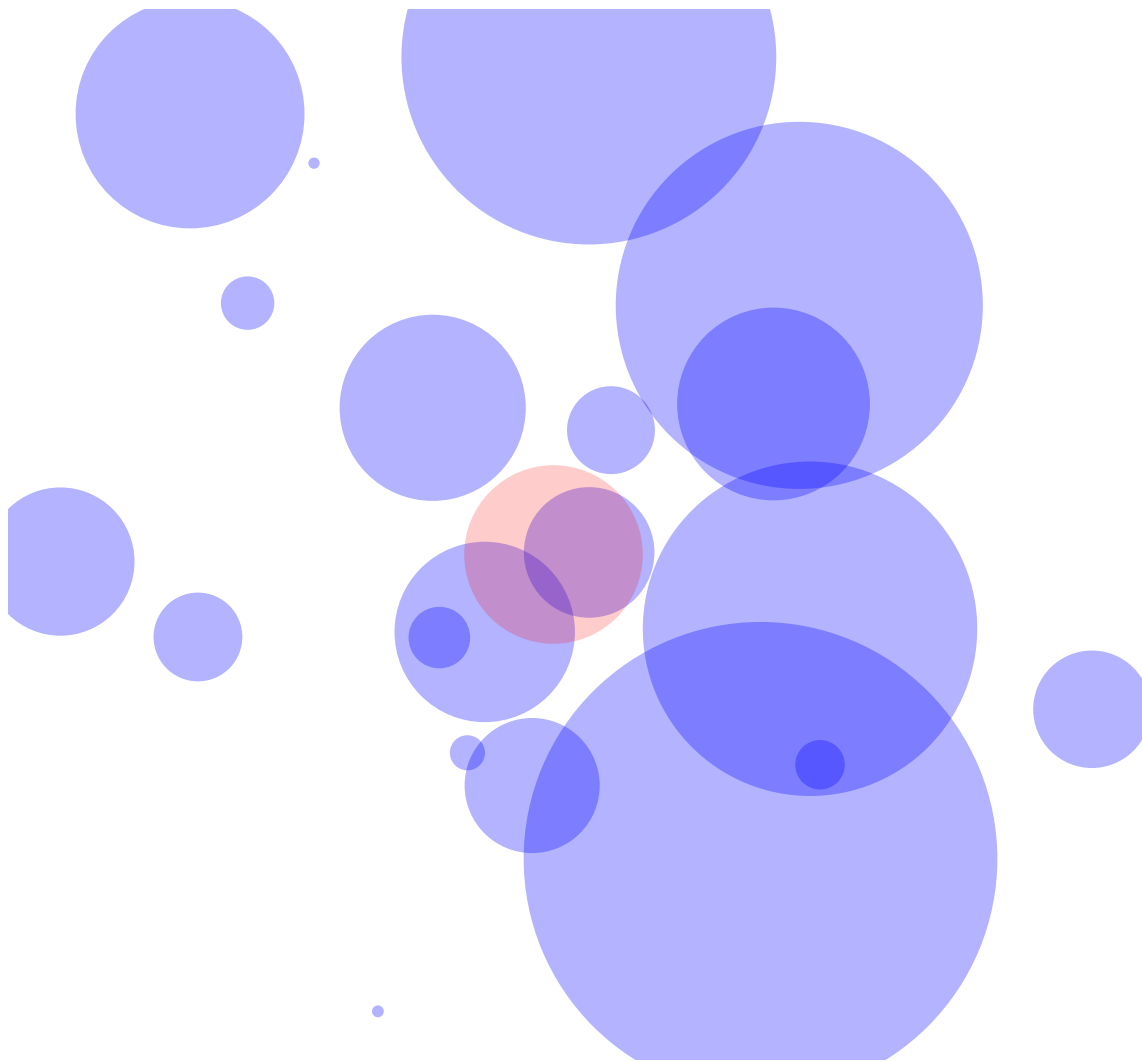
```
oy = l*np.sin(np.radians(t))
circle(ox+250, oy+250, abs(ox), color="blue")
end()
```



1.8.3 塗りつぶし円

```
from plot import begin, end, circle

np.random.seed(123)
x = np.random.rand(20)
y = np.random.rand(20)
r = np.random.rand(20)*10
begin(aspect=1)
circle(x, y, r, fill=True, color="blue", alpha=0.3)
circle(0.5, 0.5, 6, fill=True, color="red", alpha=0.2)
end
```



1.9 ellipse()

楕円を描く。

```
ellipse(ox, oy, ra, rb, phi=0, begin=0, end=360, color="black",
        linewidth=0.5, linestyle="-")
```

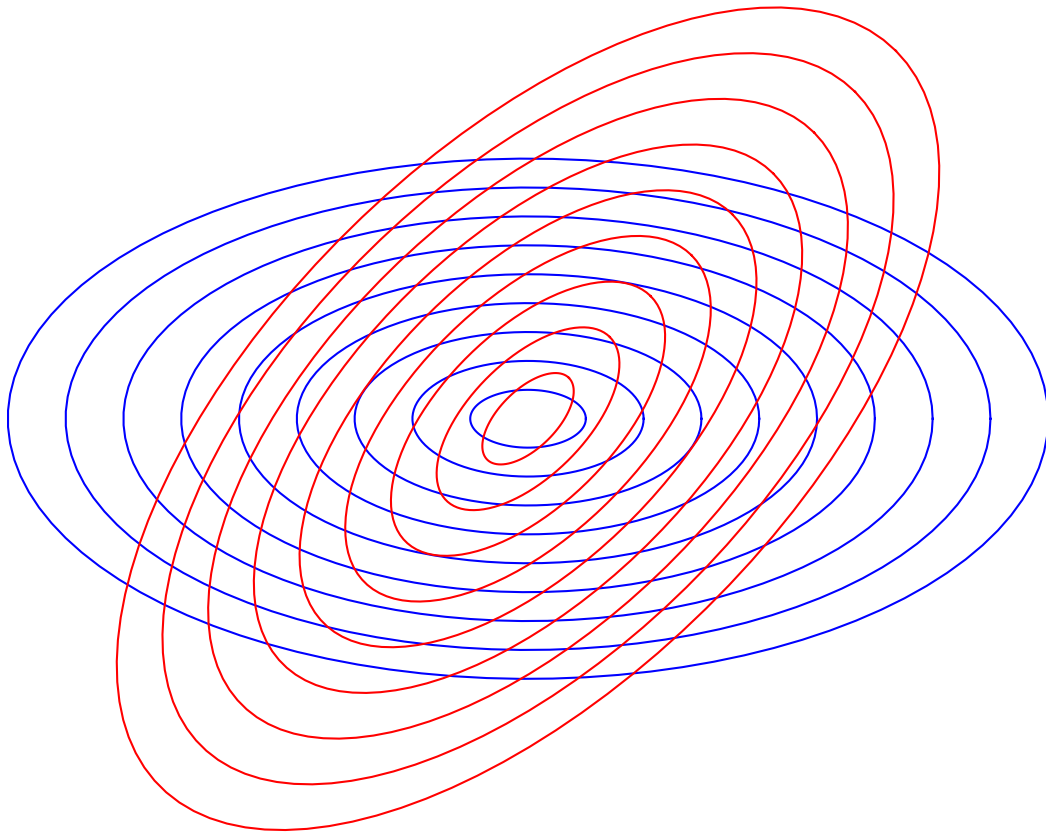
<code>ox, oy</code>	楕円の中心座標 (ベクトルまたはスカラー)
<code>ra, rb</code>	長径と短径 (ベクトルまたはスカラー)
<code>phi</code>	長径と水平線の角度 (デフォルトは 0)
<code>begin</code>	描き始め (水平線からの角度。デフォルトは 0)
<code>end</code>	楕円を描く線の色 (デフォルトは黒)
<code>linewidth</code>	楕円を描く線の幅 (デフォルトは 0.5)
<code>linestyle</code>	楕円を描く線の種類 (デフォルトは実線)
<code>arg</code>	以下で確認できる辞書型キーワード

```
import matplotlib.pyplot
```

```
help(matplotlib.pyplot.Line2D)
```

```
from plot import begin, end, ellipse

begin(aspect=1)
l = np.linspace(0, 100, 10)
ellipse(250, 250, l, l/2, color="blue")
ellipse(250, 250, l, l/2, phi=45, color="red")
end()
```



1.10 polygon()

正多角形を描く。

```
polygon(x, y, l, k, color="black", linewidth=0.5, linestyle="-", **arg)
```

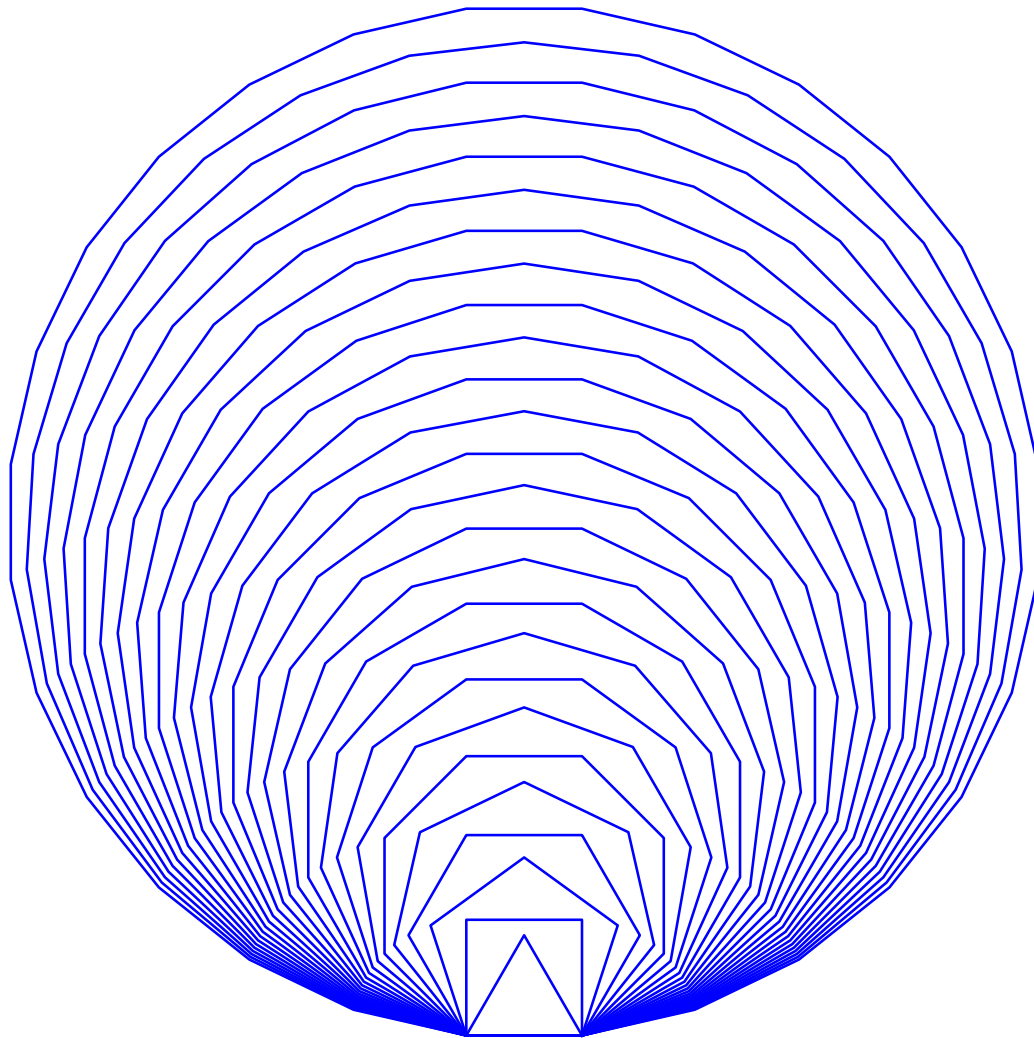
x, y	正多角形の描き始め座標 (ベクトルまたはスカラー)
l	正多角形の辺の長さ (ベクトルまたはスカラー)
k	正 k 角形の指定
color	正多角形を描く線の色 (デフォルトは黒)
linewidth	正多角形を描く線の幅 (デフォルトは 0.5)

`linestyle` 正多角形を描く線の種類 (デフォルトは実線)

`arg` 以下で確認できる辞書型キーワード

```
import matplotlib.pyplot  
help(matplotlib.pyplot.Line2D)
```

```
from plot import begin, end, polygon  
  
begin(aspect=1)  
polygon(0, 0, 1, np.arange(3, 30), color="blue")  
end()
```



1.11 polygon2()

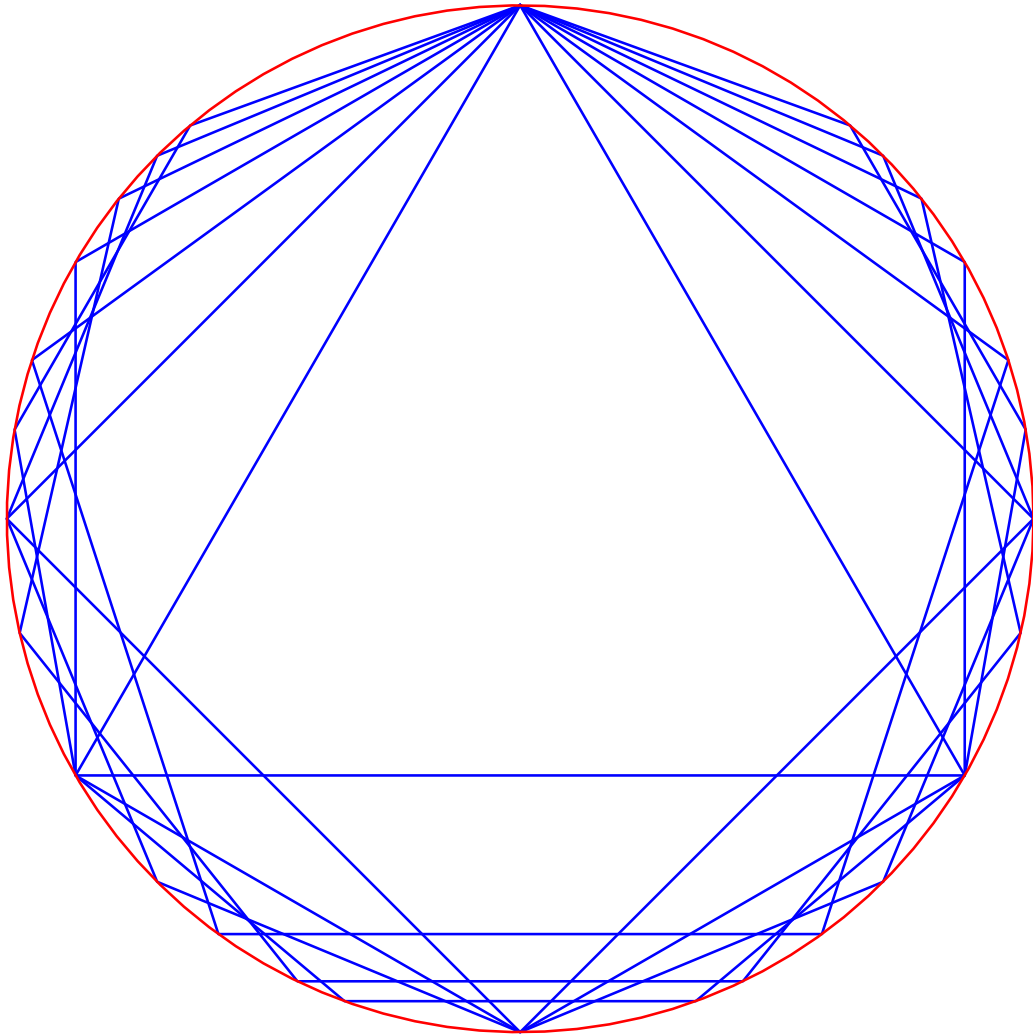
円に内接する正多角形を描く。

```
polygon2(ox, oy, r, k, phi=90, color="black", linewidth=0.5, linestyle="-", **arg)
```

<code>ox, oy</code>	円の中心座標 (ベクトルまたはスカラー)
<code>r</code>	円の半径 (ベクトルまたはスカラー)
<code>k</code>	正 k 角形の指定
<code>phi</code>	書き始めの角度指定 (デフォルトは 90)
<code>color</code>	正多角形を描く線の色 (デフォルトは黒)
<code>linewidth</code>	正多角形を描く線の幅 (デフォルトは 0.5)
<code>linestyle</code>	正多角形を描く線の種類 (デフォルトは実線)
<code>arg</code>	以下で確認できる辞書型キーワード

```
import matplotlib.pyplot  
help(matplotlib.pyplot.Line2D)
```

```
begin(aspect=1)  
from plot import begin, end, polygon2  
  
polygon2(0, 0, 1, np.arange(3, 10), color="blue")  
circle(0, 0, 1, color="red")  
end()
```



1.12 grid()

格子 (グリッド) を描く。

```
grid(x1, y1, x2, y2, wx, wy=None, color="black", linewidth=0.5, linestyle="-", **arg)
```

<code>x1, y1</code>	格子の左下隅の座標 (スカラー)
<code>x2, y2</code>	格子の右上隅の座標 (スカラー)
<code>wx</code>	横方向の格子の間隔
<code>wy</code>	縦方向の格子の間隔 (None ならば <code>wx</code> と同じ)
<code>color</code>	格子を描く線の色 (デフォルトは黒)
<code>linewidth</code>	格子を描く線の幅 (デフォルトは 0.5)
<code>linestyle</code>	格子を描く線の種類 (デフォルトは実線)
<code>arg</code>	以下で確認できる辞書型キーワード

```
import matplotlib.pyplot
help(matplotlib.pyplot.Line2D)
```

```
from plot import begin, end, grid

begin(aspect=1)
grid(0, 0, 5, 3, 0.25, color="green")
end()
```

