グラフ作成プログラム

中・高・大学生のための

パイソンでグラフを描こう

おとといのジョー　著

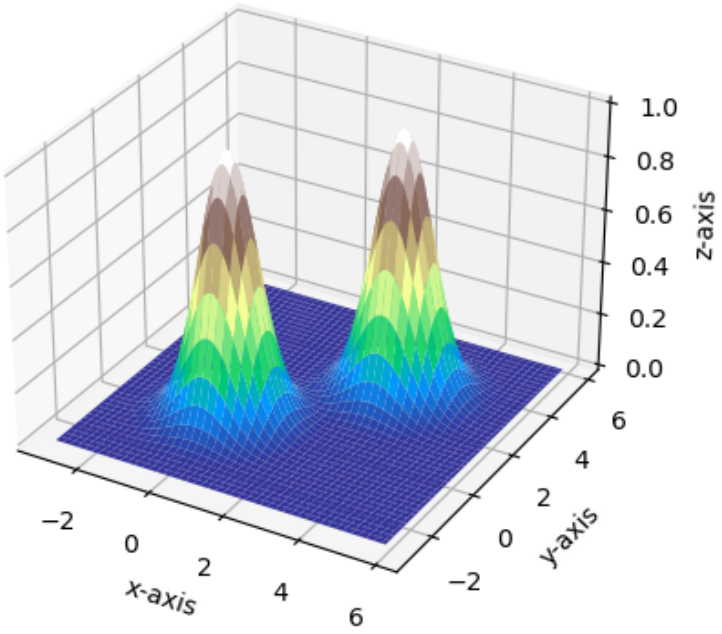
はじめに

中学の数学から，関数のグラフは頻繁に出てくるよね。関数の性質などが分れば，手描きでもかなり正確にグラフを描くことができます。しかし，ちょっと式が複雑だったり，変動がはげしかったり，定義域がよく分からなかったりすると，なかなかきれいに描けませんよね。

プログラム言語パイソン（Python）では，関数のグラフを描くのが，大変簡単です。ここで紹介したプログラムの中の関数を書き換えれば簡単にグラフを描くことができます。宿題やレポート，また論文などにも自分の描いたグラフが使えます。2，3 日練習すれば，誰でも自由に使えるようになると思います。

Let ’ s enjoy programmings !

$$z = e^{-(x^2+y^2)} + e^{-((x-3)^2+(y-3)^2)}$$



二変数関数の曲面

０．　パイソン（Python）について

今年（2021 年）の 4 月 20 日の朝日新聞に，来春から高校で「情報 I」が必修科目になり，プログラミングなどを扱うという報告がありました。教科書会社で扱うプログラミング言語などは，

パイソン，　表計算マクロ，　ジャバスクリプト

が主なものです。特に，パイソンは全ての教科書で扱っていますね。

私は，パイソンやジャバスクリプトは使ったことがないので，早速 Python の参考書：

山田祥寛著；独習 Python，SE（翔泳社） 3000 円（2020 年）
（この参考書では，グラフィックスは扱っていない）

を買って来て，Windows 10 の環境下でインストールし，プログラミングができるようにしました。Python 関連のソフトは無料で提供されています。上の本が示すやり方で，Anaconda と Visual Studio Code （VS コードと呼ぶ）をインストールすれば，直ちに使用可能です。私がインストールした Python のバージョンは ver.3.5.8 です。

Python 関連ソフトのインストールは，　https://www.anaconda.com/products/individual
VSCode のインストールは，　https://code.vidualstudio.com/Download
からできます。本を見ながらインストールするのが安全ですかね。

VS コード（エディターと考えてよい）でプログラムを作成し，実行（Run）したときの画面を下にのっけます。

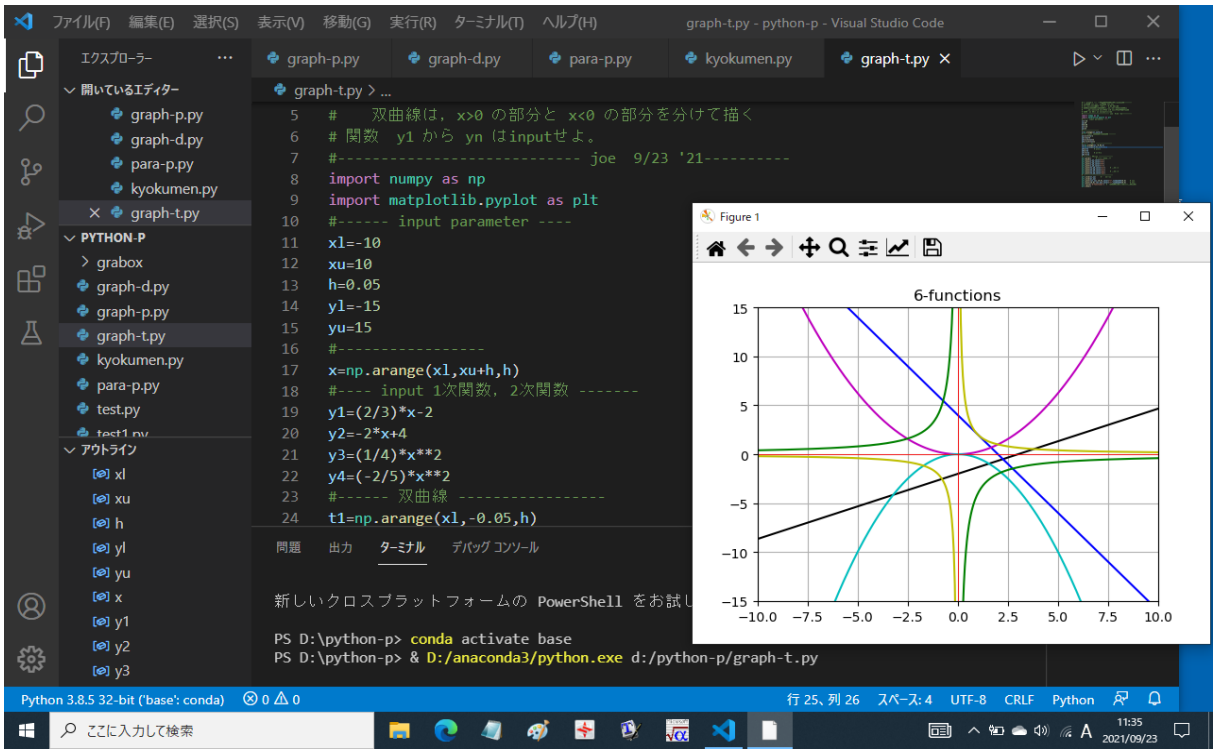


図 A.　パソコン画面のハードコピー

［画面説明］　１．黒の背景の中央にある番号のついた一連の文がプログラムです。＃で始まる行（または文）は，注釈行（プログラムとは無関係な文）です。
２．プログラム（ファイル）名は，やや右上にある graph-t.py です。このプログラムの実行は，画面の左やや中央にあるファイル名 graph-t.py を右クリックし，表示されたコンテキストメニュー

から「ターミナルで Python ファイルを実行」をクリックします。画面右上にある三角印 ▷ をクリックしても同様です。

３．プログラムを実行すると，中央にグラフを描いた白い領域が現れ，目的のグラフを見ることができます。上の図では，6 つの関数のグラフが描かれています。曲線のカラーは指定できます。指定しない場合は，コンピューターが勝手に決めます。グラフの縦・横の縮尺も，コンピューターが勝手に決めます。しかし，図の大きさ（縮尺など）は，できあがってから自由に変えられるので，自分で好みの大きさ・縮尺に操作してください。

４．関数の式を変えて再び実行するときは，図（Figure 1) はキャンセル（右上のバツ印×をクリック）しておいて下さい。画面下の タスクバーに白のメモ用紙のようなアイコンがあると，図はキャンセルされていません。

５．図 A 下のタスクバーにある青いリボンのようなアイコンが，VS コードを立ち上げるためのものです。

１．　1 次関数と頂点が原点の 2 次関数（放物線）

中学校で習う関数は，次のようなものです。

1) 1 次関数： $y = ax + b$, グラフは直線になり，a は傾き，b は切片と呼ばれる。特に，$y = ax$ は比例を表す式と呼ばれている。必ず，原点を通る。

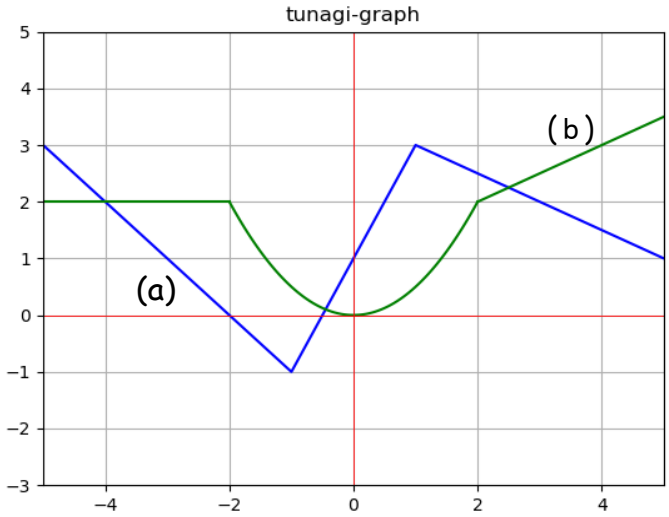
2) 反比例の関数： $y = \dfrac{a}{x}$ $(a \neq 0)$, $x = 0$ で不連続な関数。グラフは原点に関して対称で，双曲線と呼ばれる。

3) 2 次関数： $y = ax^2$ $(a \neq 0)$, $a > 0$ のとき，グラフは上に開いた放物線，$a < 0$ の とき，グラフは下に開いた放物線である。原点は放物線の頂点，y 軸は対称軸と呼ばれる。

4) 2 つ以上のの曲線をつないだ関数：　例えば，次のような関数。

$$\text{(a)}\quad y = \begin{cases} -x - 2 & (x \leqq -1 \text{ のとき}) \\ 2x + 1 & (-1 < x \leqq 1 \text{ のとき}) \\ -\dfrac{1}{2}x + \dfrac{7}{2} & (1 < x \text{ のとき}) \end{cases} \qquad \text{(b)}\quad y = \begin{cases} 2 & (x \leqq -2 \text{ のとき}) \\ \dfrac{1}{2}x^2 & (-2 < x \leqq 2 \text{ のとき}) \\ \dfrac{1}{2}x + 1 & (2 < x \text{ のとき}) \end{cases}$$

(a), (b) の 2 つの関数のグラフは，次のようになる。これを描くプログラムは後で示します。



図B　つないだ関数

さて，関数 $y=f(x)$ のグラフを描く基本は x の値を何点かとり（例えば，0, 0.5, 1, 1.5, 2, 2.5, 3 の 7 点；これらを基本の数列と呼びましょう），これらの点における y の値；

$$f(0),\ f(0.5),\ f(1),\ f(1.5),\ f(2),\ f(2.5),\ f(3)$$

を計算して，$xy-$ 平面上の 7 点　$(0,\ f(0)),\ (0.5,\ f(0.5)),\ (1,\ f(1)),\ \cdots,(3,\ f(3))$ を線で結べば，近似の曲線を得ることができます。あくまで，近似の折れ線です。

正確な曲線のグラフを描くためには，x の基本の数列を小さな刻み幅で沢山とればいいのです。例えば，x の範囲が $0 \leqq x \leqq 10$ のとき，これを 100 等分する基本の数列を作りたければ，刻み幅を 0.1 にして；

$$0,\ 0.1,\ 0.2,\ 0.3,\ \cdots,1,\ 1.1,\ 1.2,\ 1.3,\ \cdots,5,\ 5.1,\ \cdots,9.8,\ 9.9,\ 10 \quad \cdots \quad ①$$

とすればいいですね。この基本数列に対して y の値を計算して，$xy-$ 平面の 101 個の点を線で結べば，満足できる曲線を得ることができるでしょう。刻み幅は小さければ小さいほどいいですが，余り小さくしても見た目は変わらないでしょう。刻み幅は 0.1 から 0.01 くらいで十分です。

プログラミングで上の 101 個の x の基本の数列をつくる命令は；

x=np.arange(0, 10.1, 0.1)　$\cdots$　②

です。これが ① と同等の数列です。カッコ内の 10.1（区間の右端の点）は基本の数列には入らないことになっています。この数列に対する，例えば，1 次関数 $y=5x-2$ の値の計算は；

y=5*x-2　$\cdots$　③

と書けばよい。式③が 101 個の y の値の数列です。式中の掛け算は ” * ” で表し，x^2 は ” x**2 ” と書きます。割り算 $\frac{a}{b}$ は ” a/b ” と書きます。

グラフを描く命令は；

plt.plot(x, y, color= ” k ”)　$\cdots$　④

です。すごく簡単ですね。color= ” k ” は ” 黒い線で描け ” という命令です。

ここで，中学校で習う関数のグラフを 7 つ描いて見ました。グラフとプログラムをのせます。

【プログラム C】

```
1   #--- graph-t ------ 中学で習う関数のグラフを描く------------
2   # x の範囲 [xl,xu], y の範囲 [yl,yu]
3   # 直線近似の刻み幅　h=0.1 ~ 0.01 くらい
4   # 関数は 1 次関数，2 次関数，双曲線（反比例の関数），折れ線
5   #     双曲線は，x>0 の部分と x<0 の部分を分けて描く
6   # 関数　y1 から yn は input せよ。
7   #------------------------------- 9/23 '21-- joe --------
8   import numpy as np
9   import matplotlib.pyplot as plt
10  #------ input parameter ----
11  xl=-10
12  xu=10
13  h=0.05
14  yl=-15
15  yu=15
16  #-----------------
17  x=np.arange(xl,xu+h,h)
```

$y = \frac{2}{3}x - 2$ 黒

$y = -2x + 4$ 青

$y = \frac{1}{4}x^2$ マゼンタ

$y = -\frac{2}{5}x^2$ シアン

$y = \frac{2}{x}$ 黄

$y = -\frac{4}{x}$ 緑

折れ線 赤

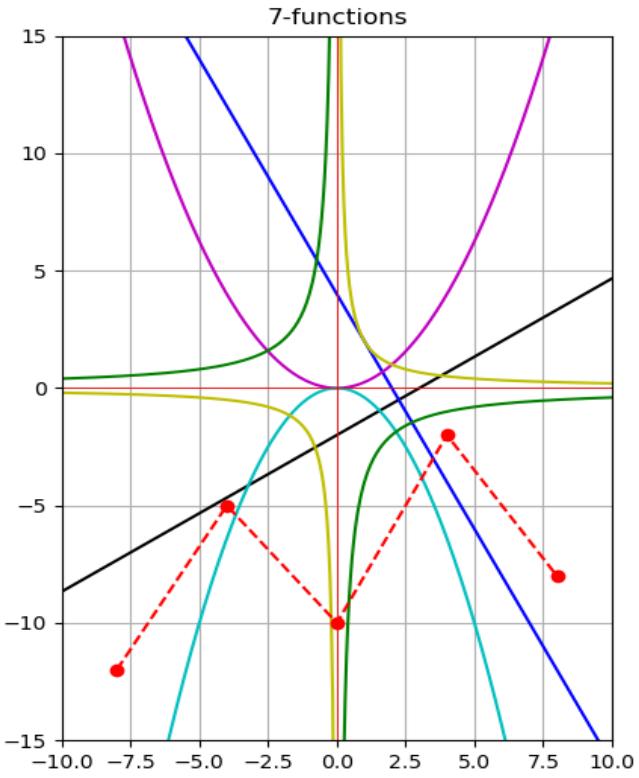


図 C

```
18  #---- input 1次関数, 2次関数 -------
19  y1=(2/3)*x-2
20  y2=-2*x+4
21  y3=(1/4)*x**2
22  y4=(-2/5)*x**2
23  #------ 双曲線 -----------------
24  t1=np.arange(xl,-0.05,h)
25  t2=np.arange(0.05,xu+h,h)
26  y5=2/t1       # y=2/x
27  y6=2/t2
28  y7=-4/t1      # y=-4/x
29  y8=-4/t2
30  # -------- グラフ ---------------
31  plt.grid()      # 補助線を入れる
32  plt.plot(x,y1,color="k")
33  plt.plot(x,y2,color="b")
34  plt.plot(x,y3,color="m")
35  plt.plot(x,y4,color="c")
36  plt.plot(t1,y5,color="y")      # 双曲線1
37  plt.plot(t2,y6,color="y")
38  plt.plot(t1,y7,color="g")      # 双曲線2
39  plt.plot(t2,y8,color="g")
40  # 折れ線グラフ
```

```
41  xo=[-8,-4,0,4,8]
42  yo=[-12,-5,-10,-2,-8]
43  plt.plot(xo,yo,marker="o",color="r",linestyle="--")
44  # ------------------------------
45  plt.xlim(xl,xu)      #  枠の設定
46  plt.ylim(yl,yu)
47  plt.plot([xl,xu],[0,0],color="r",linewidth=0.6)     # x 軸
48  plt.plot([0,0],[yl,yu],color="r",linewidth=0.6)     # y 軸
49  plt.title('7-functions')      # タイトル (日本語使えない)
50  plt.show()
```

プログラムの各文で重要なところを説明しましょう。

8 行； 数値計算用の道具（ライブラリ）を取り込むという意味。例えば arange という命令は，上の① のような数列を作れという意味です。

9 行； グラフを描くのに必要なライブラリを取り込む。例えば plot(x,y) という命令は，複数個の点列 $(x,\ y)$ を直線で結びなさいという意味だよ。

11 行〜15 行； x の範囲と，y の範囲を指定。h は x の基本数列を作るときの刻み幅。

19 行〜22 行； 1 次関数と 2 次関数を 4 つ書いた。

24 行〜29 行； 2 つの双曲線 $y=\dfrac{2}{x},\ y=-\dfrac{4}{x}$ を描くために，負の部分の x の基本数列 (t1) と，正の部分の x の基本数列 (t2) を作り，4 つの y の数列（$y=\dfrac{2}{x}$ の x の正の部分と負の部分および $y=-\dfrac{4}{x}$ の x の正の部分と負の部分）を作った。

31 行； 平面上に補助線を入れる命令。カッコ内に何も書かなければ，コンピューターが勝手に補助線を入れます。

32 行〜39 行； 6 つの関数のグラフを描きます。カラーを指定することができます。曲線の太さはコンピューターがかってに決めています（linewidth=1）。もちろん，太さを指定することもできます。

41 行，42 行； xy-平面上の 5 点 (-8, -12), (-4, -5), (0, -10), (4, -2), (8, -8) を指定します。

43 行； 上の 5 点を赤丸で示し，点を破線で結びます。色は赤を指定しました。

45 行，46 行； グラフを描く xy-平面の大枠を設定。

47 行，48 行； x 軸と y 軸を赤で引きます。線の太さは，関数のグラフより細くしてあります。

49 行； 図の上部に，図のタイトルを入れます。日本語が使えないのが残念ですね。

50 行； 図を表示しなさいという命令。

中学校で習う関数のグラフは，上のプログラムを見ながら，自分で作ることができますよね。やってみて下さい。

新しくプログラムを作らなくても，上の関数の部分を自分の描きたい関数に書き換え，表示する xy-平面の大きさも数値を書き換えて，必要としない文には，行の頭に＃をつければいいですよね。すなわち，ちょっと書き換えてそのまま使うことができます。

Let ' s draw some graphs !!

図 B のプログラムをのせておきます。参考にして下さい。

【プログラム B】

```
1    # ---- tunagi-gra ---- 連続関数をつないだグラフ ---------
2    #  折れ線と 1 次関数と 2 次関数のつなぎ
```

```
3    #-------------------------------- 9/29 '21-- joe --------
4    import numpy as np
5    import matplotlib.pyplot as plt
6    #------ input parameter ----
7    xl=-5
8    xu=5
9    h=0.05
10   yl=-3
11   yu=5
12   #-------------------
13   x=np.arange(-2,2+h,h)
14   #---- 折れ線 --------
15   xo=[-5,-1,1,5]
16   yo=[3,-1,3,1]
17   plt.plot(xo,yo,color="b")
18   x1=[-5,-2]
19   y1=[2,2]
20   plt.plot(x1,y1,color="g")
21   x2=[2,5]
22   y2=[2,3.5]
23   plt.plot(x2,y2,color="g")
24   y3=0.5*x**2                    # 2次関数
25   plt.plot(x,y3,color="g")
26   # -------------------------------
27   plt.grid()
28   plt.xlim(xl,xu)      #  枠の設定
29   plt.ylim(yl,yu)
30   plt.plot([xl,xu],[0,0],color="r",linewidth=0.6)    # x軸
31   plt.plot([0,0],[yl,yu],color="r",linewidth=0.6)    # y軸
32   plt.title('tunagi-graph')     # タイトル(日本語使えない)
33   plt.show()
```

2．　高校で習う関数；三角関数，指数関数などのグラフ

関数が連続の場合，グラフを描くのは簡単です。ただし，対数関数などのように定義域が制限される関数は，独立変数（x のこと）の範囲に注意してプログラムを書いて下さい。

次の 4 つの連続関数のグラフを描きましょう。

$$y = 4\cos x - 2\sin 2x + 1 \qquad \text{（全区間で連続）} \quad \cdots \ ①$$

$$y = 10\,e^{-x^2} - 1 \qquad \text{（　〃　）} \quad \cdots \ ②$$

$$y = -\frac{1}{10}x^3 + \frac{1}{5}x^2 + \frac{29}{10}x - 3 \qquad \text{（　〃　）} \quad \cdots \ ③$$

$$y = 4\log(5+x) \qquad \text{（自然対数，定義域は } -5 < x \text{ ）} \quad \cdots \ ④$$

グラフとプログラムををのせましょう。

$y = 4\cos x - 2\sin 2x + 1$　黒

$y = 10e^{-x^2} - 1$　青

$y = -\frac{1}{10}x^3 + \frac{1}{5}x^2 + \frac{29}{10}x - 3$　紫

$y = 4\log(5+x)$　緑

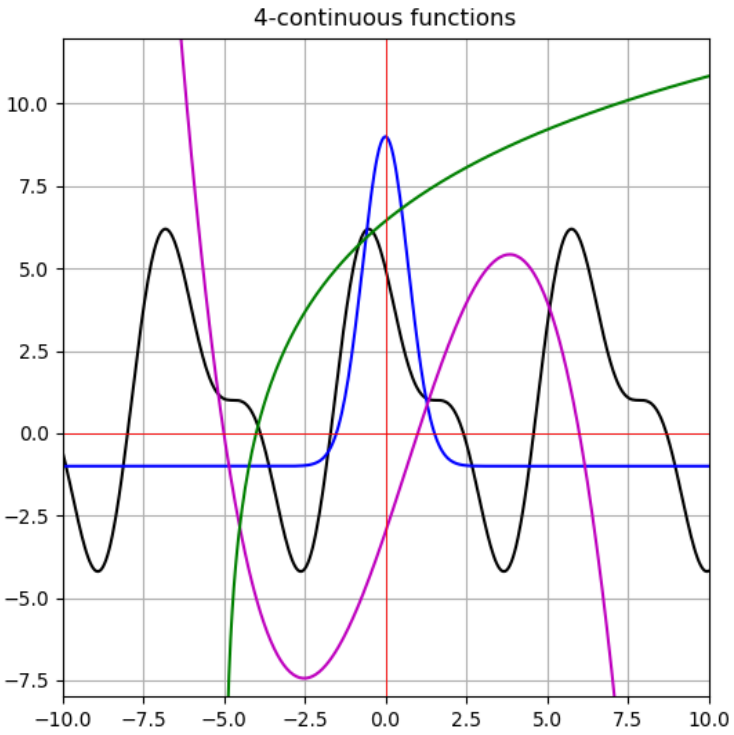


図 D

【プログラム D】

```
#--- graph-c --- 高校で習う関数のグラフを描く-----
# x の範囲 [xl,xu], y の範囲 [yl,yu]
# 直線近似の刻み幅　h=0.1 ~ 0.01 くらい
# 関数は連続関数のみ, 対数関数あり（定義域注意）
# 関数　y1 から yn は input せよ。
#------------------------ joe  9/17 '21----------
import numpy as np
import matplotlib.pyplot as plt
#------ input parameter ----
xl=-10
xu=10
h=0.05
yl=-8
yu=12
#-----------------
x=np.arange(xl,xu+h,h)
#------ input 連続関数 ---------------
y1=4*np.cos(x)-2*np.sin(2*x)+1
y2=10*np.exp(-x*x)-1
y3=(-1/10)*x**3+(1/5)*x**2+(29/10)*x-3
# 自然対数（定義域あり）
t=np.arange(-5+0.01,xu+h,h)
y4=4*np.log(5+t)
#----------- グラフ-------------------
plt.grid()      # 補助線入れる
```

```
plt.plot(x,y1,color="k")
plt.plot(x,y2,color="b")
plt.plot(x,y3,color="m")
plt.plot(t,y4,color="g")
plt.xlim(xl,xu)      #  枠の設定
plt.ylim(yl,yu)
plt.plot([xl,xu],[0,0],color="r",linewidth=0.6)     #x 軸
plt.plot([0,0],[yl,yu],color="r",linewidth=0.6)     #y 軸
plt.title('4-continuous functions')
plt.show()
```

(注意) 定義された x の数列 ”np.arange(xl, xu+h, h) ” に対する三角関数などの値の数列は，
”np.sin(x), np.cos(3*x) ” などと書きます。

ここで，不連続関数のグラフを何も考えずに描いてみましょう。関数は

$$y = \frac{8}{x^2 - 9} - 2 \quad (x = \pm 3 \text{ で不連続}) \quad \cdots \quad ⑤$$

$$y = \tan x \qquad (-\frac{5\pi}{2} < x < \frac{5\pi}{2}\ ) \quad \cdots \quad ⑥$$

$y = \tan x$ は，この範囲では $x = \pm\frac{3\pi}{2},\ \pm\frac{\pi}{2}$ で不連続ですが，何も考えずにグラフを描きましょう。プログラムとグラフは下の通りです。

【プログラム E】

```
#--- furenzoku-test ---- 不連続関数のグラフを描く---
# x の範囲 [xl,xu], y の範囲 [yl,yu]
# 直線近似の刻み幅  h=0.1 から 0.01 の間くらい
# 関数  y1 から yn は input せよ。
#----------------------- joe  10/3 '21------------
import numpy as np
import matplotlib.pyplot as plt
pai=3.141592653589793
#------ input parameter ----
h=0.1
xl=-5*pai/2
xu=-xl
yl=-10
yu=10
#-------------------------
x=np.arange(xl,xu+h,h)
y1=8/(x*x-9)-2
y2=np.tan(x)
# ----------------
plt.grid()
plt.plot(x,y1)
plt.plot(x,y2,color="g")
plt.xlim(xl,xu)
plt.ylim(yl,yu)
```

```
plt.plot([xl,xu],[0,0],color="r",linewidth=0.6)     #x 軸
plt.plot([0,0],[yl,yu],color="r",linewidth=0.6)     #y 軸
plt.title('2-discontinuous functions')
plt.show()
```

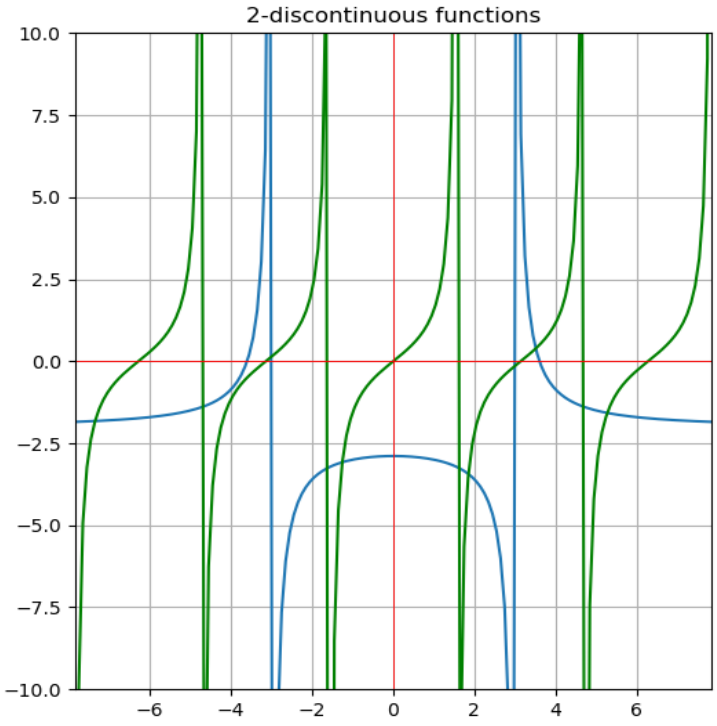
$y = \dfrac{8}{x^2 - 9} - 2$　青

$x = \pm 3$ で不連続

$y = \tan x$　緑

$x = \pm\dfrac{\pi}{2}, \pm\dfrac{3\pi}{2}, \pm\dfrac{5\pi}{2}$

で不連続



図E

さて，上のグラフは正確だと思いますか？ 一見すると,「コンピューターが不連続点を見極めて，そこに縦の漸近線を引いた」と思ってしまいますが，とんでもない間違いです。コンピューターはそんな判断はできません。関数⑤は，$x = -3$ で不連続ですが，この関数を $f(x)$ とかいたとき，コンピューターは不連続点をはさむ次の 2 点；

$(-3-\delta_1,\ f(-3-\delta_1))$,　$(-3+\delta_2,\ f(-3+\delta_2))$　　$(\delta_1,\ \delta_2$ は微小な正の数)

を線で結んだ，にすぎません。したがって，不連続点における縦の漸近線のように見える線は漸近線ではありません。まあ，上のグラフは正確ではないということです。不思議に思う方は，例えば，$h = 0.2$ *or* $h = 0.3$ にしてプログラムを run して見て下さい。

コンピューターは，関数の不連続点を見つけて，そこを避けてグラフを描くというようなことはしません（ある意味で python ソフトの欠陥）ので，人間（プログラミングをする方）がこのことを考慮しなければなりません。即ち，不連続な関数は，不連続点を避けてグラフを描き，不連続点上にある漸近線は自分で引きましょう（プログラミングするという意味）。不連続点が沢山あると，プログラムを書くのはけっこう大変ですよ。

図 E と同じ関数のグラフを描くのに，不連続点を考慮してプログラミングすると，次のようになります。グラフは図 F です。

【プログラム F】

```
#--- furenzo-d1 ---- 不連続関数のグラフを描く---
# x の範囲 [xl,xu], y の範囲 [yl,yu]
# 直線近似の刻み幅  h=0.1 から 0.01 の間くらい
# 関数は，y=8/(x*x-9)-2,  y=tanx
```

```
#---------------------- joe  10/3 '21------------
import numpy as np
import matplotlib.pyplot as plt
pai=3.141592653589793
#------ input parameter ----
h=0.01
xl=-5*pai/2
xu=-xl
yl=-10
yu=10
#-------分数関数 -------------
x1=np.arange(xl,-3,h)
x2=np.arange(-3+h,3,h)
x3=np.arange(3+h,xu+h,h)
y1=8/(x1*x1-9)-2
y2=8/(x2*x2-9)-2
y3=8/(x3*x3-9)-2
plt.plot(x1,y1,color="b")
plt.plot(x2,y2,color="b")
plt.plot(x3,y3,color="b")
plt.plot([-3,-3],[yl,yu],color="k",linewidth=0.6)  # 漸近線
plt.plot([3,3],[yl,yu],color="k",linewidth=0.6)
# ------ tanx ----------
t1=np.arange(xl+h,-3*pai/2,h)
t2=np.arange(-3*pai/2+h,-pai/2,h)
t3=np.arange(-pai/2+h,pai/2,h)
t4=np.arange(pai/2+h,3*pai/2,h)
t5=np.arange(3*pai/2+h,xu,h)
y4=np.tan(t1)
y5=np.tan(t2)
y6=np.tan(t3)
y7=np.tan(t4)
y8=np.tan(t5)
plt.plot(t1,y4,color="m")
plt.plot(t2,y5,color="m")
plt.plot(t3,y6,color="m")
plt.plot(t4,y7,color="m")
plt.plot(t5,y8,color="m")
plt.plot([-3*pai/2,-3*pai/2],[yl,yu],color="hotpink",linewidth=0.6) # 漸近線
plt.plot([-pai/2,-pai/2],[yl,yu],color="hotpink",linewidth=0.6)
plt.plot([pai/2,pai/2],[yl,yu],color="hotpink",linewidth=0.6)
plt.plot([3*pai/2,3*pai/2],[yl,yu],color="hotpink",linewidth=0.6)
# ------------------------------
plt.grid()
plt.xlim(xl,xu)
plt.ylim(yl,yu)
```

```
plt.plot([xl,xu],[0,0],color="g",linewidth=0.7)     #x 軸
plt.plot([0,0],[yl,yu],color="g",linewidth=0.7)     #y 軸
plt.title('2-discontinuous functions')
plt.show()
# end
```

$y = \dfrac{8}{x^2 - 9} - 2$　　青

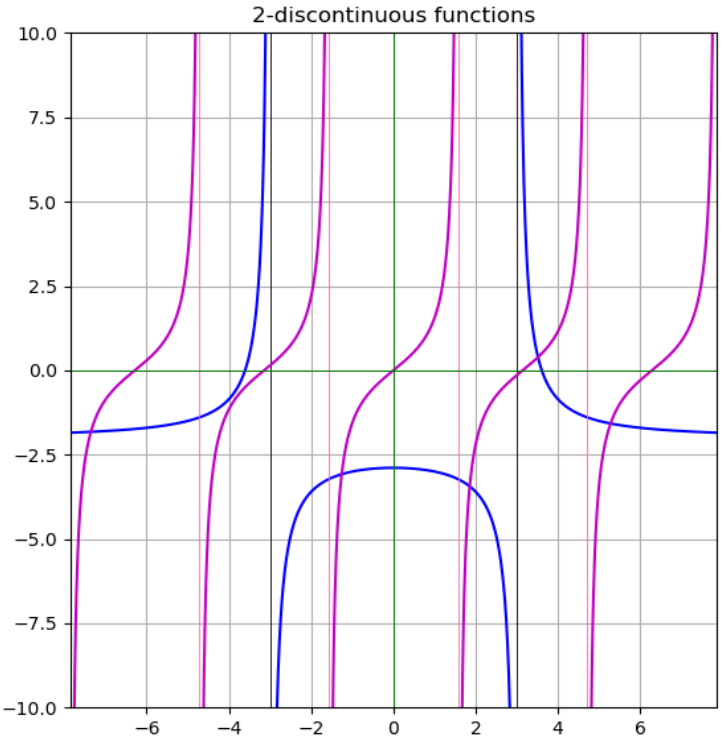
漸近線　$x = \pm 3$　を引く（黒）

$y = \tan x$　　紫

漸近線

$x = \pm\dfrac{\pi}{2}, \pm\dfrac{3\pi}{2}$

を引く（ピンク）



図F

図 F では漸近線は正確に入ってます。また，刻み幅 h を少々大きくしても（0.1〜0.2 位），全体のグラフは大域的にはあまり変化しません。しかし上の［プログラム F］で見られる様に，不連続関数を正確にグラフにするのは面倒ですね。コンピューターソフトの方で，不連続点の処理が自動的にできるようになるといいですね（これは，可能かもしれないが，難しいね！）。

高校の数学 III で扱う，媒介変数表示関数（パラメーター表示関数とも言う）のグラフを考えましょう。パラメーター表示関数とは，

$$\begin{cases} x = f(t) \\ y = g(t) \end{cases} \qquad (\ a \leqq t \leqq b\ ) \quad \cdots \quad ⑦$$

と表される関数で，t を媒介変数またはパラメーターと呼ぶ。関数の表すグラフは $xy-$ 平面上の曲線である。

例 1．(1) 放物線：　$\begin{cases} x = 2t \\ y = t^2 \end{cases}$　$(\ -\infty < t < \infty\ )$　$\Longleftrightarrow$　$y = \dfrac{1}{4}x^2$

(2) 円：　$\begin{cases} x = a\cos t + p \\ y = a\sin t + q \end{cases}$　$(\ 0 \leqq t \leqq 2\pi\ )$　$\Longleftrightarrow$　$(x-p)^2 + (y-q)^2 = a^2$

(3) 楕円：　$\begin{cases} x = a\cos t + p \\ y = b\sin t + q \end{cases}$　$(\ 0 \leqq t \leqq 2\pi\ )$　$\Longleftrightarrow$　$\dfrac{(x-p)^2}{a^2} + \dfrac{(y-q)^2}{b^2} = 1$

(4) 双曲線 1 ： $\begin{cases} x = \dfrac{a}{\cos t} \\ y = b\tan t \end{cases}$ $\left(-\dfrac{\pi}{2} < t < \dfrac{3}{2}\pi \right)$ $\Longleftrightarrow$ $\dfrac{x^2}{a^2} - \dfrac{y^2}{b^2} = 1$

双曲線 2： $\begin{cases} x = a\tan t \\ y = \dfrac{b}{\cos t} \end{cases}$ $\left(-\dfrac{\pi}{2} < t < \dfrac{3}{2}\pi \right)$ $\Longleftrightarrow$ $\dfrac{x^2}{a^2} - \dfrac{y^2}{b^2} = -1$

（注意）双曲線は，$t = \dfrac{\pi}{2}$ で不連続になることに注意せよ.。また，パラメーター表示関数は上の例のように，いつも x, y の式で表されるとは限らない。

ここで，次の 4 つの関数のグラフを描くプログラムと結果をのせます。

(a) $\begin{cases} x = 6\cos t + 2 \\ y = 6\sin t + 2 \end{cases}$ $(0 \leqq t \leqq 2\pi)$ (b) $\begin{cases} x = 6\cos t - 2 \\ y = 3\sin t - 6 \end{cases}$ $(0 \leqq t \leqq 2\pi)$

(c) $\begin{cases} x = 2(2\cos t - \cos 2t) \\ y = 2(2\sin t - \sin 2t) \end{cases}$ $(0 \leqq t \leqq 2\pi)$ (d) $\begin{cases} x = \dfrac{2}{\cos t} \\ y = 3\tan t \end{cases}$ $(-\dfrac{\pi}{2} < t < \dfrac{3}{2}\pi)$

(a) の表す曲線は円，(b) は楕円，(c) はカージオイド（心臓形），(d) は双曲線です。以下はプログラムとグラフです。

【プログラム G】

```
#---- para-p ------ パラメーター表示関数のグラフ ---------------------
#  x=f(t),  y=g(t),      t はパラメーターで   a<=t<=b   とする，ただし
#                  f,g が連続な三角関数のとき   0=<t=<k*pai (k は整数) がよい
#                   f,g が不連続のとき，t の範囲工夫せよ
#                  漸近線などは，描いた方がよい
#    グラフの表示領域は正方形がよい
# ----------------------------------------10.14 '21 --joe-------
import numpy as np
import matplotlib.pyplot as plt
pai=3.141592653589793
# -- input parameter --
tl=0
tu=2*pai
h=0.01
xl=-10
xu=10
yl=xl
yu=xu
# ---------------------
t=np.arange(tl,tu+h,h)   # t の数列作成
# -- input 関数 -----------
x=6*np.cos(t)+2        # 円
y=6*np.sin(t)+2
x1=6*np.cos(t)-2       # 楕円
y1=3*np.sin(t)-6
x2=2*(2*np.cos(t)-np.cos(2*t))     # カージオイド
y2=2*(2*np.sin(t)-np.sin(2*t))
```

```
# ----不連続関数；双曲線---(x/a)**2-(y/b)**2=1 --------
#       x=a/cost, y=btan(t),  -pai/2<t<3pai/2
t1=np.arange(-pai/2+0.1,pai/2-0.1,0.05)
t2=np.arange(pai/2+0.1,3*pai/2-0.1,0.05)
x3=2/np.cos(t1)
y3=3*np.tan(t1)
x4=2/np.cos(t2)
y4=3*np.tan(t2)
# -------------------------------
plt.plot(x,y,color="b",linewidth=0.9)
plt.plot(x1,y1,color="g",linewidth=1.1)
plt.plot(x2,y2,color="k")    # linewidth なしは,linewidth=1
plt.plot(x3,y3,color="m")
plt.plot(x4,y4,color="m")
plt.xlim(xl,xu)       # 表示枠設定
plt.ylim(yl,yu)
# ------ 双曲線の漸近線 y=± (b/a)x ------------
plt.plot([-6.67,6.67],[-10,10],color="y",linewidth=0.6)
plt.plot([-6.67,6.67],[10,-10],color="y",linewidth=0.6)
#  ----------------------------
plt.plot([-10,10],[0,0],color="r",linewidth=0.6)    # x 軸
plt.plot([0,0],[-10,10],color="r",linewidth=0.6)    # y 軸
plt.plot([xl,xu],[yl,yu],color="r",linewidth=0.6)  # y=x
plt.plot([xl,xu],[yu,yl],color="r",linewidth=0.6)  # y=-x
plt.title('4-parametric equations')
plt.show()
```

円

$$(x-2)^2+(y-2)^2=6^2$$

楕円

$$\frac{(x+2)^2}{3^2}+\frac{(y+6)^2}{2^2}=1$$

双曲線

$$\frac{x^2}{2^2}-\frac{y^2}{3^2}=1$$

カージオイド

$$x=2(2\cos t-\cos 2t)$$
$$y=2(2\sin t-\sin 2t)$$

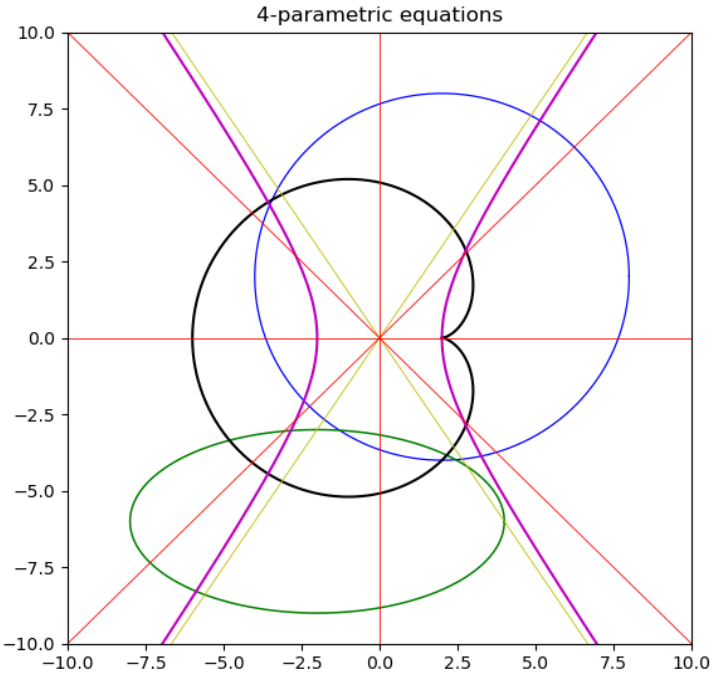


図 G

３．　大学で習う 逆三角関数，双曲線関数，二変数関数などのグラフ

逆関数の定義を復習しておきましょう。連続な単調関数（単調増加または単調減少） $y=f(x)$ に対して，逆関数は定義されます。

$$y=f(x) \quad (a \leqq x \leqq b,\ or\ -\infty<x<\infty \text{ で定義されている}) \quad \cdots \quad ①$$

の x と y を交換した式 $x=f(y)$ を改めて y について解いた関数

$$y=f^{-1}(x) \quad (\text{定義域は①の値域}) \quad \cdots \quad ②$$

を $y=f(x)$ の逆関数という。関数①, ② の 2 つのグラフは，幾何学的には，直線 $y=x$ に関して対称である。

例２．(1) $y=2x-4$ の逆関数は　$y=\dfrac{1}{2}x+2$. これら 2 つの関数の定義域と値域は実数全体。

(2) 指数関数　$y=2^x-1$　（定義域は実数全体，値域は $-1<y$ ）　の逆関数は

$$y=\log_2(1+x) \quad (\text{定義域は } -1<x, \text{値域は実数全体}).$$

$\because)$ $x=2^y-1 \quad \Leftrightarrow \quad 2^y=1+x \quad \Leftrightarrow \quad y=\log_2(1+x)$.

(3) 三角関数の逆関数（グラフは図 H）

(a) $y=\sin x$ の逆関数は，これが単調増加である区間 $-\dfrac{\pi}{2} \leqq x \leqq \dfrac{\pi}{2}$ で考えます。 $x=\sin y$ は y について解けないので，これを

$$y=\sin^{-1}x \quad (\text{アークサインエックスと読む}) \quad \cdots \quad ③$$

と書いたのが逆関数です。逆関数③の定義域は　$-1 \leqq x \leqq 1$, 値域は　$-\dfrac{\pi}{2} \leqq y \leqq \dfrac{\pi}{2}$.

(b) $y=\cos x$ の逆関数は $0 \leqq x \leqq \pi$ で考えて，逆関数は

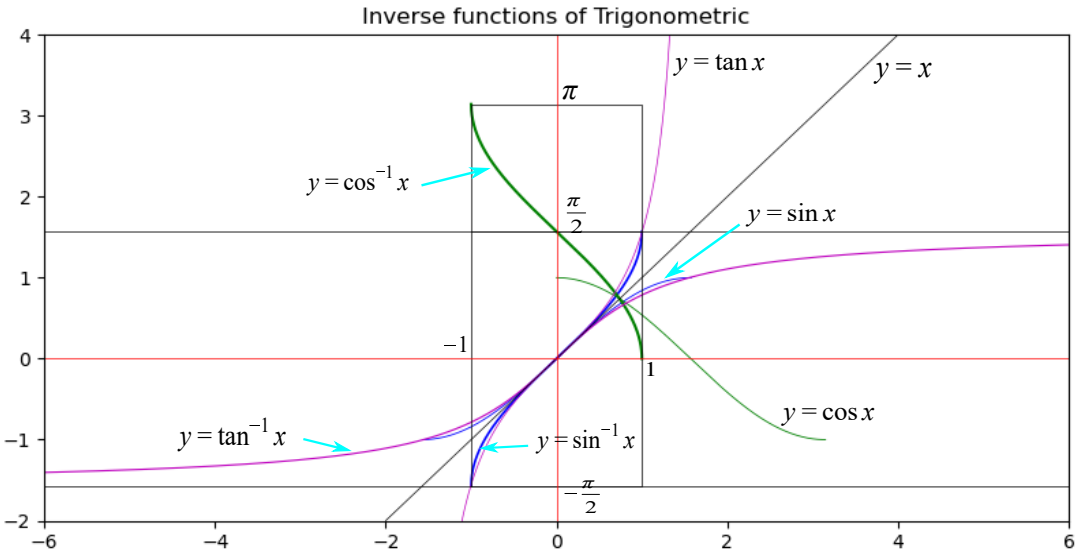
$$y=\cos^{-1}x \quad (\text{アークコサインエックスと読む}) \quad \cdots \quad ④$$

式④の定義域は　$-1 \leqq x \leqq 1$, 値域は $0 \leqq y \leqq \pi$.

(c) $y=\tan x$ の逆関数は　$-\dfrac{\pi}{2}<x<\dfrac{\pi}{2}$ で考えて（原点を通る 1 本の枝），逆関数は

$$y=\tan^{-1}x \quad (\text{アークタンジェントエックスと読む}) \quad \cdots \quad ⑤$$

式⑤の定義域は実数全体，値域は　$-\dfrac{\pi}{2}<y<\dfrac{\pi}{2}$.



図H

図 H では，原点近傍に曲線が集まっていて，見栄えは良くないが $y = \sin^{-1} x$ は太いブルー，$y = \sin x$ は細いブルーで描かれている。同様に

$y = \cos^{-1} x$ は太いグリーン，　$y = \cos x$ は細いグリーン

$y = \tan^{-1} x$ は太いマゼンタ，　$y = \tan x$ は細いマゼンタ

で描かれている。図 H のプログラムを以下にのせる。

【プログラム H】

```
# --- gyakusan ---------  逆三角関数 3 つ--------
#   y=arcsin(x),  y=arccos(x), y=arctan(x)
#      元の三角関数と共に 3 つのグラフを表示
#       必要な補助線も入れる
# -------------------------10/19 '21 ---joe-----
import numpy as np
import matplotlib.pyplot as plt
pai=3.141592653589793
# ---- parameters -----
h=0.01
xl=-6
xu=6
yl=-2
yu=4
# ----- 計算 (arcsin, arccos) --------
x1=np.arange(-pai/2,pai/2+h,h)
x2=np.arange(-1,1+0.0005,0.0005)
x3=np.arange(0,pai+h,h)
x4=np.arange(-pai/2+h,pai/2,h)
x5=np.arange(xl,xu+h,h)
y1=np.sin(x1)
y2=np.arcsin(x2)
y3=np.cos(x3)
y4=np.arccos(x2)
y5=np.tan(x4)
y6=np.arctan(x5)
#----------------
plt.plot(x1,y1,color="b",linewidth=0.6)
plt.plot(x2,y2,color="b",linewidth=1.2)
plt.plot(x3,y3,color="g",linewidth=0.6)
plt.plot(x2,y4,color="g")
plt.plot(x4,y5,color="m",linewidth=0.5)
plt.plot(x5,y6,color="m",linewidth=0.8)
# ---------  逆三角の枠，漸近線，設定  ------
plt.plot([-2,4],[-2,4],color="k",linewidth=0.5)
plt.plot([1,1,-1,-1],[pai/2,pai,pai,pai/2],color="k",linewidth=0.5)
plt.plot([-1,1,1,-1,-1],[-pai/2,-pai/2,pai/2,pai/2,-pai/2],color="k",linewidth=0.5)
plt.plot([xl,xu],[pai/2,pai/2],color="k",linewidth=0.5)
plt.plot([xl,xu],[-pai/2,-pai/2],color="k",linewidth=0.5)
```

```
# ----------------------------
plt.xlim(xl,xu)      #  枠の設定
plt.ylim(yl,yu)
plt.plot([xl,xu],[0,0],color="r",linewidth=0.5)     # x 軸
plt.plot([0,0],[yl,yu],color="r",linewidth=0.5)     # y 軸
plt.title('Inverse functions of Trigonometric')
plt.show()
# end
```

次の例題で，大学で習うであろう関数のグラフを描く練習をしましょう。

例３． 次の関数の定義域，値域などを調べグラフを描け。

(1) $y = 2\log(1+x^2) - 4$　　(2) $y = 2\cos^{-1}\left(\dfrac{x-2}{4}\right)$

(3) $y = \sinh x = \dfrac{e^x - e^{-x}}{2}$　　（ハイパボリックサインエックスと読む）

(4) $y = \sinh^{-1} x$　（ (3) の逆関数）

解説）(1) 定義域は実数全体，値域は $-4 \leqq y$. 微分は

$y' = \dfrac{4x}{1+x^2}$, $y'' = \dfrac{4(1-x)(1+x)}{(1+x^2)^2}$　だから，点 $(0,-4)$ が極小値かつ最小値。

$x<0$ で単調減少，$x>0$ で単調増加。変曲点は ± 1. 従って，$x<-1$ で上に凸，$-1<x<1$ で下に凸，$1<x$ で上に凸。グラフを手描きするときは，増減表を書くのが基本ですが，ここでは増減表は書きません。

(2) 定義域は $-2 \leqq x \leqq 6$, 値域は $0 \leqq y \leqq 2\pi$. 微分は

$$y' = -\frac{2}{\sqrt{12+4x-x^2}} < 0, \quad y'' = (4-2x)(12+4x-x^2)^{-\frac{3}{2}}.$$

だから，グラフは単調減少。変曲点は $x=2$ で，$x<2$ で下に凸，$2<x$ で上に凸。

(3) これは双曲線関数の 1 つ。定義域・値域，共に実数全体。微分は

$$y' = \frac{e^x + e^{-x}}{2} > 0, \quad y'' = \frac{e^x - e^{-x}}{2}$$

だから，単調増加関数で，原点が変曲点，$x<0$ で上に凸，$0<x$ で下に凸。

(4) $y = \sinh x$ の逆関数を求めよう。$x = \dfrac{e^y - e^{-y}}{2} \quad \Leftrightarrow \quad 2x = e^y - \dfrac{1}{e^y}$

より，$Y = e^y$ とおくと（$Y>0$ に注意），$Y^2 - 2xY - 1 = 0 \quad \therefore \quad Y = x + \sqrt{x^2+1}$.

よって，逆関数は

$$y = \sinh^{-1} x = \log(x + \sqrt{x^2+1}). \quad \cdots \quad ⑥$$

（注意）双曲線関数は上の (3),(4) 以外に，次の 4 つがある。

$y = \cosh x = \dfrac{e^x + e^{-x}}{2}$　$(0 \leqq x)$,　とその逆関数　$y = \cosh^{-1} x$,　および

$y = \tanh x = \dfrac{e^x - e^{-x}}{e^x + e^{-x}}$　$(-\infty < x < \infty)$,　とその逆関数　$y = \tanh^{-1} x$.

問１． 逆関数 $y = \cosh^{-1} x$, $y = \tanh^{-1} x$ を求め，それらの定義域・値域を示せ。

以下に，例 3 のグラフ 4 つと，そのプログラムをのせる。

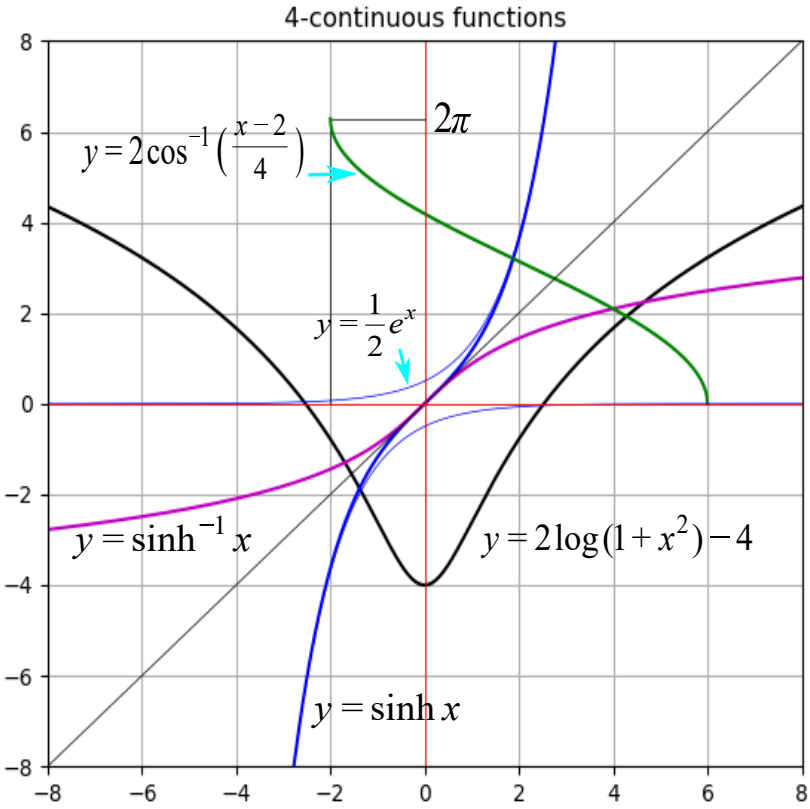


図 I

図 I には，$y=\sinh x$ の漸近曲線としての $y=\frac{1}{2}e^{x}$ と $y=-\frac{1}{2}e^{-x}$ が描かれていますが，原点から離れた所では $y=\sinh x$ とほとんど重なって見えてしまいますね。描かない方が良かったですかね。

【プログラム I】

```
#--- graph-dai --- 大学で習うだろう関数のグラフ-----
# x の範囲 [xl,xu], y の範囲 [yl,yu]
# 直線近似の刻み幅  h=0.1 ~ 0.01 くらい
# 逆関数，対数関数などは定義域，値域に注意
# 関数  y1 から yn は input せよ。
#------------------------ joe  9/17 '21----------
import numpy as np
import matplotlib.pyplot as plt
pai=3.141592653589793
#------ input parameter ----
xl=-8
xu=8
h=0.01
yl=xl
yu=xu
#-----------------
x=np.arange(xl,xu+h,h)
t=np.arange(-2,6+h/2,h/2)
#------ input 関数 ---------------
```

```
y1=2*np.log(1+x*x)-4
y2=np.sinh(x)
y3=np.log(x+np.sqrt(x*x+1))
y4=2*np.arccos((t-2)/4)
#----漸近線など --------
y5=x
y6=0.5*np.exp(x)
y7=-0.5*np.exp(-x)
plt.plot([-2,-2],[0,2*pai],color="k",linewidth=0.5) # 補助線
plt.plot([-2,0],[2*pai,2*pai],color="k",linewidth=0.5)
#----------- グラフ-------------------
plt.plot(x,y1,color="k")
plt.plot(x,y2,color="b")
plt.plot(x,y3,color="m")
plt.plot(t,y4,color="g")
plt.plot(x,y5,color="k",linewidth=0.5)  # 補助曲線
plt.plot(x,y6,color="b",linewidth=0.5)
plt.plot(x,y7,color="b",linewidth=0.5)
# ---- 枠の設定など  ------
plt.grid()
plt.xlim(xl,xu)
plt.ylim(yl,yu)
plt.plot([xl,xu],[0,0],color="r",linewidth=0.6)    #x 軸
plt.plot([0,0],[yl,yu],color="r",linewidth=0.6)    #y 軸
plt.title('4-continuous functions')
plt.show()
# end
```

さて最後に，次の形の二変数関数；

$$z = f(x, y), \quad \text{定義域：}\ a \leqq x \leqq b, \quad c \leqq y \leqq d \quad \cdots \quad ⑦$$

の 3 次元グラフの描き方を解説しましょう。と，言っても，私も使ってみるのは初めてです。

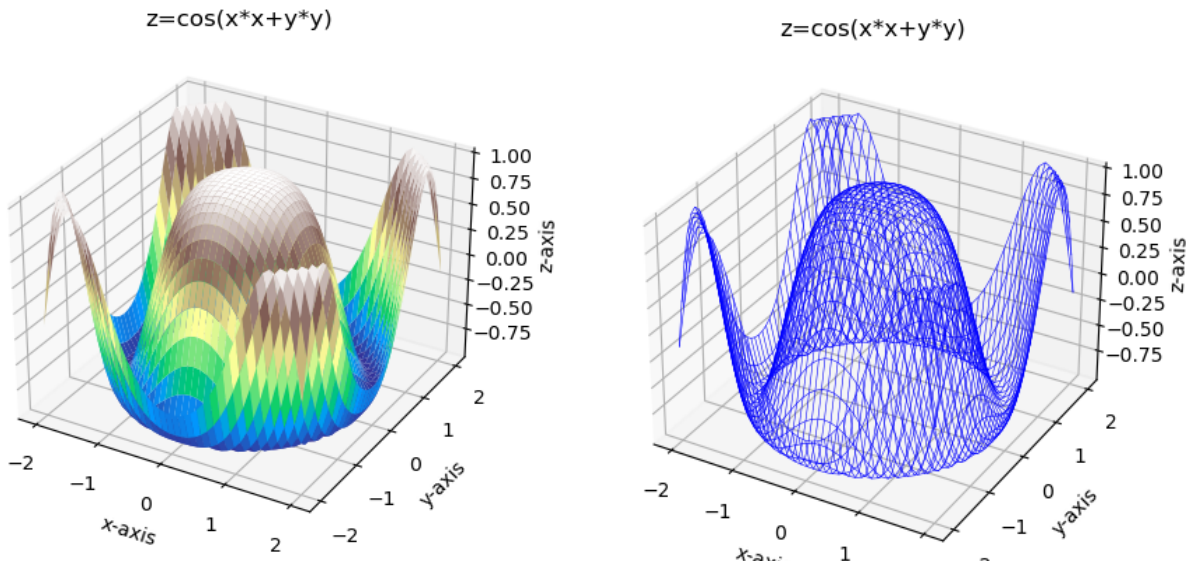
二変数関数⑦の定義域は $xy-$ 平面上の四角形であり，点 (x, y) がこの四角形を埋めつくす様に動いたとき，z の値は一般に連続した曲面（平面を含む）になります。このような曲面を，平面上に表現したのが，3 次元グラフです。中学校で習うみとり図のことです。空間内の立体を平面上で表現するので，難しいよね。立体は，見る位置によってそのみとり図は違ってくるよね。そんなこんなで，グラフを描くのは易しくないです。

パイソンの 3 次元グラフは，定義域の四角形の右下隅に立った位置から，中央に向かって立体を見おろした時に見えるような図形になっていますね。けっこう上手に描かれていると思います。

1 つグラフを描いて見ましょう。関数は

$$z = \cos\ (x^2 + y^2) \quad \cdots \quad ⑧$$

です。曲面は，水面上に石を落としたときにできるような減衰しない無限の波のような形です。原点を通る平面（$xy-$ 平面と垂直なもの）で切ったとき，切り口の図形はコサインカーブです。2 種類のグラフとプログラムをのせます。



図J

【プログラム J】

```
1   # --- kyokumen2 ----------2 変数関数のグラフ------------
2   #  関数  z=f(X,Y)  は input する.
3   #               ラージ X  ラージ Y  に注意
4   #  terrain は地形を意味する: 地図のような配色
5   # -------------------------------10.28 '21 --joe-------
6   import matplotlib.pyplot as plt
7   import numpy as np
8   fig=plt.figure()                  # 描画領域の設定
9   ax=fig.add_subplot(projection='3d')  # 3 次元座標軸の設定
10  # ---- parameters ----
11  xl=-2
12  xu=2
13  yl=xl
14  yu=xu
15  h=0.1
16  # ------ 格子点はラージ X, ラージ Y -----
17  x=np.arange(xl,xu+h,h)
18  y=np.arange(yl,yu+h,h)
19  X,Y=np.meshgrid(x,y)
20  # 関数の input; ラージ X, ラージ Y を使う -------
21  z=np.cos(X*X+Y*Y)
22  # z=X**3+2*(X+1)*Y**2-4*X
23  # ---------------- どちらかを選べ ----------------------
24  # ax.plot_surface(X,Y,z,cmap='terrain')
25  ax.plot_wireframe(X,Y,z,linewidth=0.4,color="b")
26  # ---------------------------------------------------
27  ax.set_xlabel("x-axis")
28  ax.set_ylabel("y-axis")
29  ax.set_zlabel("z-axis")
```

```
30  ti="z=cos(x*x+y*y)"     # 図のタイトル（式など書け）
31  plt.title(ti)
32  plt.show()
33  # end
```

8 行と 9 行の命令で，3 次元グラフを描く準備は完了しているようです（私も詳しいことは分りません)。19 行のメッシュグリッドというのが，碁盤の目のような格子点で，この格子点上の z の値を求め，それらを線で結び曲面を形作るという方法ですね。

25 行の命令が，グラフを点と線のみで表したワイヤフレームというグラフです（図 J の右図)。これの代わりに 24 行を使ったときは，曲面に色を付けたタレインとよばれるグラフです（図 J の左図)。市販の地図を思わせるような色付けになってます。まあまあ見やすいですが，どっちが分かりやすい図でしょうか？　両方必要かもしれませんね。

図 J のようなみとり図では曲面の大体の形は分りますが，曲面に上下の変動があり極値をいくつかもつ様なときは，極値があるのかないのかさえ判断できませんよね。正確には，偏微分などの計算を用いて極値の判定をしなければなりませんね。しかしながら，詳細な等高線図があれば，極値があるのか無いのかの判定はできそうです。パイソンでは等高線図も作れますので挑戦してみましょう（私も初トライ)。

2 つの二変数関数の等高線図を描いてみましょう：

$$z = 2\sin\left(\frac{x^2}{2} + y^2\right), \qquad \cdots \quad (\dagger)$$

$$z = y^2 - 4x^2(1 - x^2). \qquad \cdots \quad (\ddagger)$$

式 (†) の等高線図は，原点中心の楕円形であることは明らかですよね。式 (‡) の等高線は，予想できますか？　等高線図を描いてみました。グラフとプログラムをのせます。

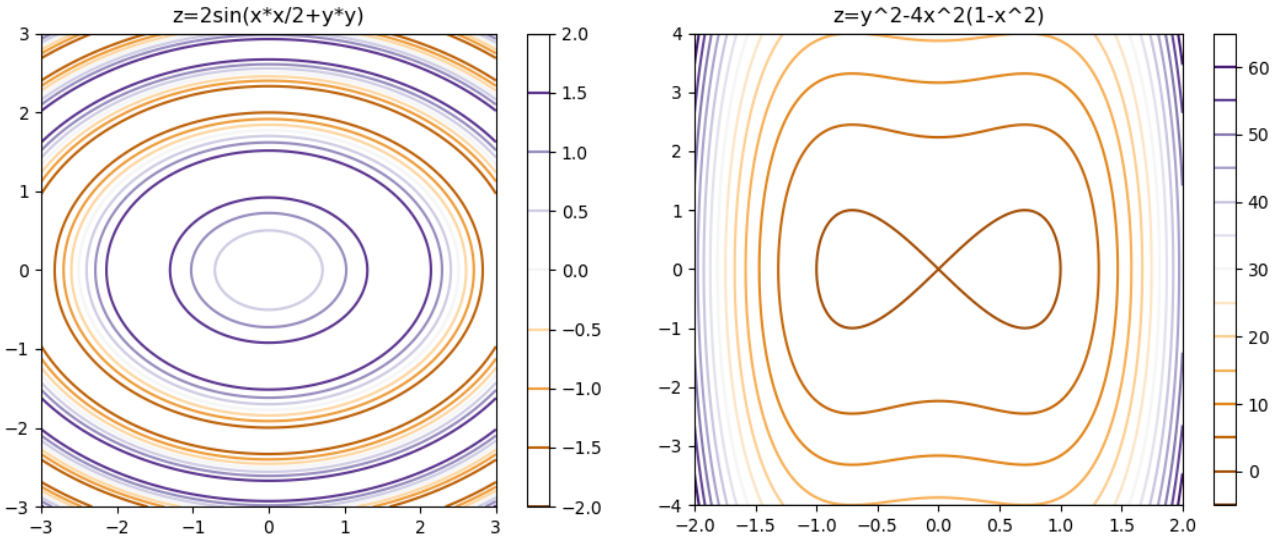


図 K

【プログラム K】

```
1  # --- toukousen ----- z=f(x,y) の等高線を描く --------
2  #    領域は   [xl,xu] × [yl,yu]（正方形がいいかな）
3  #    等高線のレベルは k（標準は 10，最大 16 まで）
```

```
 4  # --------------------------- 10.31, '21--joe------
 5  import matplotlib.pyplot as plt
 6  import numpy as np
 7  # ---parameters ----
 8  xl=-3
 9  xu=3
10  yl=-3
11  yu=3
12  h=0.01
13  k=8
14  # ----- 関数定義 (書き換えること) --------
15  def f(x,y):
16  #    return Y**2-4*X*X*(1-X*X)
17      return 2*np.sin(X*X/2+Y*Y)
18  # ---------------------------------------
19  x=np.arange(xl,xu+h,h)
20  y=np.arange(yl,yu+h,h)
21  X,Y=np.meshgrid(x,y)
22  Z=f(X,Y)
23  # -----------------
24  fig,ax=plt.subplots()
25  contk=ax.contour(X,Y,Z,k,cmap='PuOr')
26  ax.set_aspect('equal','box')          # これが無いと正方形出力
27  plt.colorbar(contk)
28  #----- 図のタイトル (式を変えたとき書き直す) -----
29  plt.title('z=2sin(x*x/2+y*y)')
30  plt.savefig("cur-fig.png",bbox_inches='tight',pad_inches=0.1)
31  plt.show()
32  # end
```

【プログラムの説明】

13 行： k は，等高線で分割される z の領域の個数である。等高線の高さ（z の値）は，切りのいい数に自動的に設定するようなので，プログラマーの指定した数（k）と一致しないこともある。

15 行，17 行： ここで，関数を定義する。

21 行： 計算する領域の碁盤の目を設定する。

24 行： グラフ表示の位置を決定する。2 つ以上のグラフを並べるとき，カッコの中にその指定を書く。

25 行： 等高線を描くための条件を指示する（等高線の数やカラー）。Pu はパープルで，まん中より上の等高線を紫で描くという意味である。値が大きくなるほど色が濃くなる。Or はオレンジを意味する。まん中より下の等高線はオレンジで，下に行くほど色が濃くなる。等高線の高さは，図の右側に表示される。

26 行： 縦横の比率を，実際の長さに指定する。これがないと，どんな領域も正方形になる。

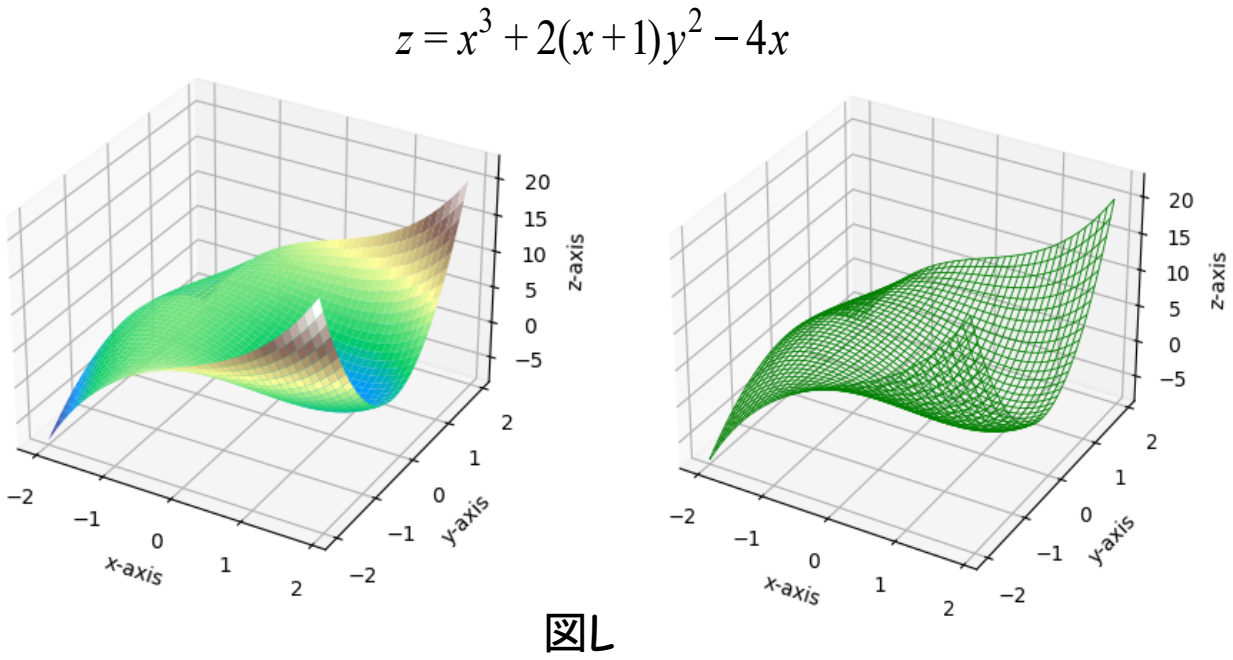
27 行： 等高線のプロット。

30 行： 作ったグラフをカレントドライブにセーブする。このファイルはグラフを保存しておく部屋（ディレクトリー）にコピーしておくのがよい。

例４．グラフから極値の判定ができるか否か，次の例で見てみましょう。

$$z = x^3 + 2(x+1)y^2 - 4x \qquad \cdots \quad (\bigstar)$$

みとり図は次のようになります。２つのタイプで描いてみました。



図L

グラフからは，極小値が１つありそうだというのがわかります（左図の右の方の青い部分）。極大値があるかないかは，グラフからは分りませんね。グラフを回転させて見ることができれば，曲面の凹凸の具合はもっと正確に認識できるかもしれません。このような立体図形の回転は，パイソンでもできるようなので，読者は挑戦してみて下さい。

関数 $(\bigstar)$ は，偏微分などの計算から，解析的に極値が分ります：

$$\text{点}\ \left(-\frac{2}{\sqrt{3}},\ 0\right) \fallingdotseq (-1.155,\ 0) \quad \text{で極大値} \quad \frac{16}{9}\sqrt{3} \fallingdotseq 3.079,$$

$$\text{点}\ \left(\frac{2}{\sqrt{3}},\ 0\right) \quad \text{で極小値} \quad -\frac{16}{9}\sqrt{3} \fallingdotseq -3.079 \quad \text{をとる。}$$

【参考】極値判定法

点 (a,b) は，$f_x(a,b)=0,\ \ f_y(a,b)=0$ を満たす点とする。また，

$$\Delta(x,y) = f_{xx}(x,y)f_{yy}(x,y) - \{f_{xy}(x,y)\}^2 \ \text{とおく。}$$

(1) $\Delta(a,b) > 0$ のとき，

$f_{xx}(a,b) > 0$ ならば，$f(x,y)$ は (a,b) で極小値をとる。

$f_{xx}(a,b) < 0$ ならば，$f(x,y)$ は (a,b) で極大値をとる。

(2) $\Delta(a,b) < 0$ のとき，

$f(x,y)$ は，(a,b) で極値をとらない。

問２．関数 $(\bigstar)$ が極大値と極小値をもつことを示せ。

次ページに，等高線図を２つ示します。

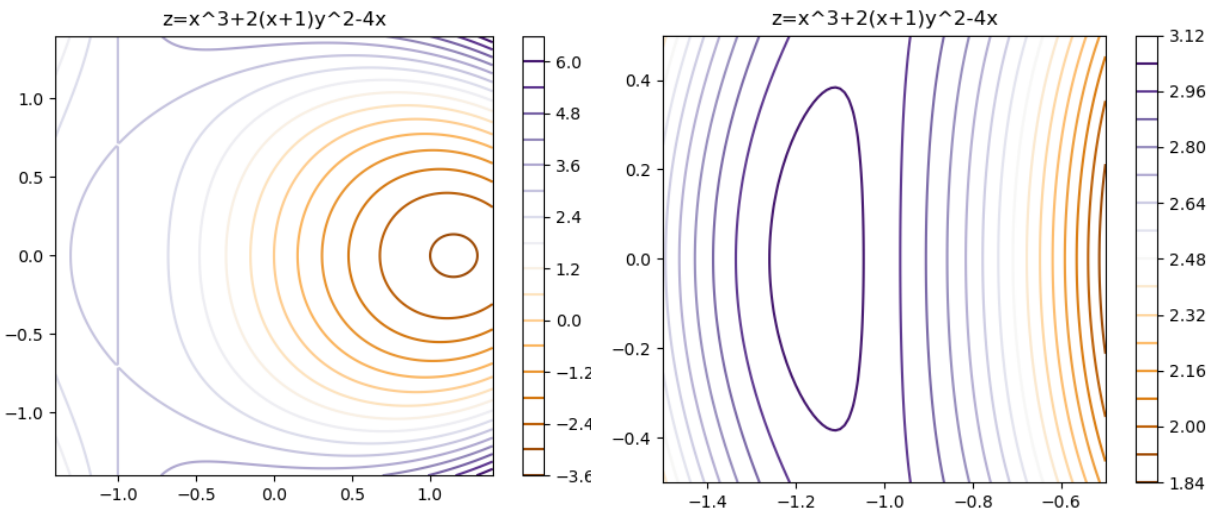


図 M

左の図では，右の方に等高線の同心円（正確には，同心閉曲線）があるので，この中に極小値をとる点があることが分かる。しかし，図の左の方に極大値があるか否かを判断するのは難しそうですね。という訳で，領域を狭くして描いた等高線が右のものです。この図では同心閉曲線（たて長のもの）がはっきり分るので，極大値の存在が分ります。等高線図もけっこう役に立つよね。（例 4 おわり）

【問の答え】

問 1． $y=\cosh x \quad (x \geqq 0)$ の逆関数は $x=\dfrac{e^y+e^{-y}}{2}$ より，$Y=e^y$ とおくと

$$2x=Y+\frac{1}{Y} \quad \Leftrightarrow \quad Y^2-2xY+1=0 \quad \Leftrightarrow \quad Y=x\pm\sqrt{x^2-1}.$$

$y \geqq 0$ より $y=\cosh^{-1}x=\log\left(x+\sqrt{x^2-1}\,\right)$ を得る。$x \geqq 1$ である。

$y=\tanh x \quad (-\infty<x<\infty)$ の逆関数は $x=\dfrac{e^y-e^{-y}}{e^y+e^{-y}}$ より，$Y=e^y$ とおくと

$$x\left(Y+\frac{1}{Y}\right)=Y-\frac{1}{Y} \quad \Leftrightarrow \quad (1-x)Y^2=1+x \quad \Leftrightarrow \quad Y=\sqrt{\frac{1+x}{1-x}}.$$

従って，$y=\tanh^{-1}x=\dfrac{1}{2}\log\left(\dfrac{1+x}{1-x}\right)$ を得る。$|x|<1$ である。

問 2． 連立方程式 $f_x=3x^2+2y^2-4=0,\ f_y=4y(x+1)=0$ より，

解 $\left(\pm\dfrac{2}{\sqrt{3}},\ 0\right),\ \left(-1,\ \pm\dfrac{1}{\sqrt{2}}\right)$ を得る。

$\Delta\left(\dfrac{2}{\sqrt{3}},\ 0\right)=16(2+\sqrt{3})>0,\ f_{xx}\left(\dfrac{2}{\sqrt{3}},\ 0\right)=4\sqrt{3}>0$ だから，

点 $\left(\dfrac{2}{\sqrt{3}},\ 0\right)$ で極小値をとる。

同様に，$\Delta\left(-\dfrac{2}{\sqrt{3}},\ 0\right)=16(2-\sqrt{3})>0,\ f_{xx}\left(-\dfrac{2}{\sqrt{3}},\ 0\right)=-4\sqrt{3}<0$ だから

点 $\left(-\dfrac{2}{\sqrt{3}},\ 0\right)$ で極大値をとる。

$\Delta\left(-1,\ \pm\dfrac{1}{\sqrt{2}}\right)=-8<0$ だから，点 $\left(-1,\ \pm\dfrac{1}{\sqrt{2}}\right)$ では極値をとらない。

参考図書

［1］金城俊哉；Python プログラミング　逆引き大全，秀和システム，2021 年，3000 円.

この本では，最後の方でグラフィックスを扱っているが，十分なものではない。日本語の本では，グラフィックスに関しての良い参考書は見当たりません。私は，いくつかのホームページから，グラフィックス関連のプログラミングを見つけて，参考にしながらこの小論を書かせてもらいました。グラフィックス関連の日本語のいい参考書が出るといいですね。

［2］一昨日冗；十進 BASIC で 方程式を遊ぶ，東京図書出版，2014 年，3500 円

ガラパゴス的言語 BASIC のプログラムを使いたい方は，この本をどうぞ。3 次元グラフは扱っていませんが，平面上のグラフ，微分方程式・差分方程式などの解曲線を描くことができます。また，複素関数によるフラクタル図形も扱ってるよ。

［3］一昨日冗；半期で学ぶ 微積分，東京図書出版，2013 年，1524 円

大学初年度で学ぶ「微積分学」のやさしい教科書。独学にも最適。

（2021 年 11 月 8 日完，一昨日 冗）