静岡理工科大学　情報学部　幸谷担当講義

レポート用紙

講義名　：数値解析１	**年月日：　2024 年　　7 月　　1 日(月)**
学籍番号：99999999	**氏名：幸谷　智紀**

本日の課題　問題 6.1, 6.2, 6.3

感想

2. 課題の難易度はどうでしたか？

詳細

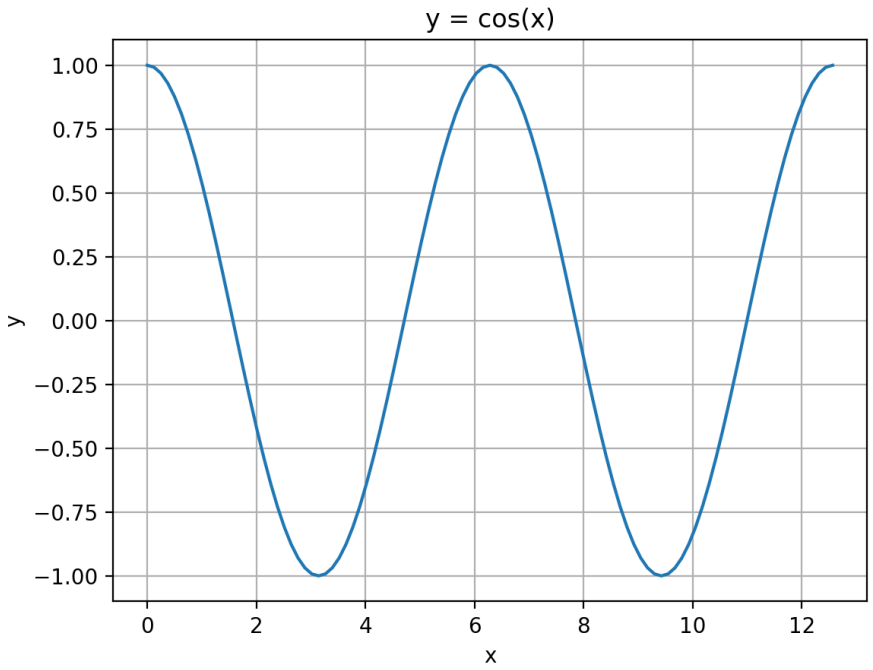
難しかった	9
少し難しかった	7
ちょうど良かった	5
簡単だった	0
とても簡単だった	0

問題 6.1

```
# plot_cosine.py: コサインカーブを描く
import matplotlib.pyplot as plt # Matplotlib
import numpy as np # NumPy

# x の範囲[a, b]を指定
a = 0
b = 4 * np.pi

# x 区間の分割数を指定
#n = 10
n = 100

# x の配列を生成
x_h = (b - a) / n # 小区間幅
x = [ a + i * x_h for i in range(n + 1)]

# x ごとに y の値を計算
y = np.cos(x)

print(x)
```

```
print(y)

# 2 次元グラフを描画
# (1) Figure オブジェクト(fig)と
# その上に Axes オブジェクト(ax)を生成
fig, ax = plt.subplots()

# (2) 折れ線グラフを ax 上に描画
ax.plot(x, y)

# (3) グラフタイトル，x 軸タイトル，y 軸タイトルを指定
ax.set_title('y = cos(x)')
ax.set_xlabel('x')
ax.set_ylabel('y')

# (4) グリッドを描画
ax.grid()

# (5) グラフを表示
plt.show()
```

実行結果:

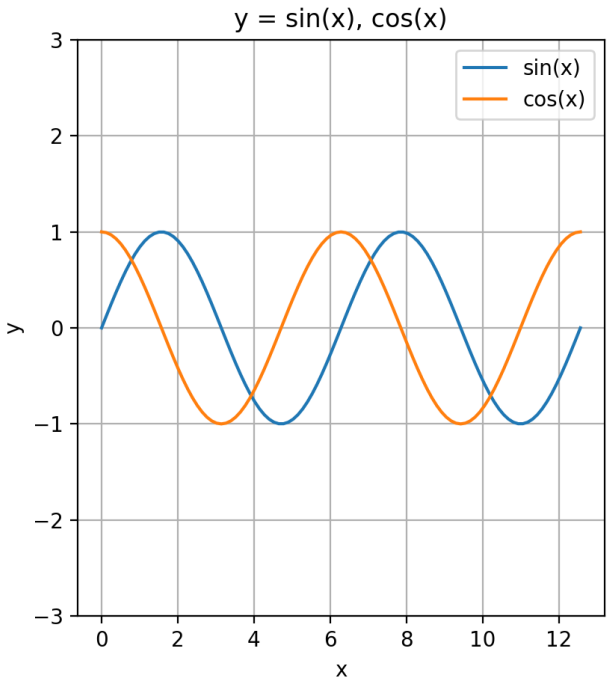


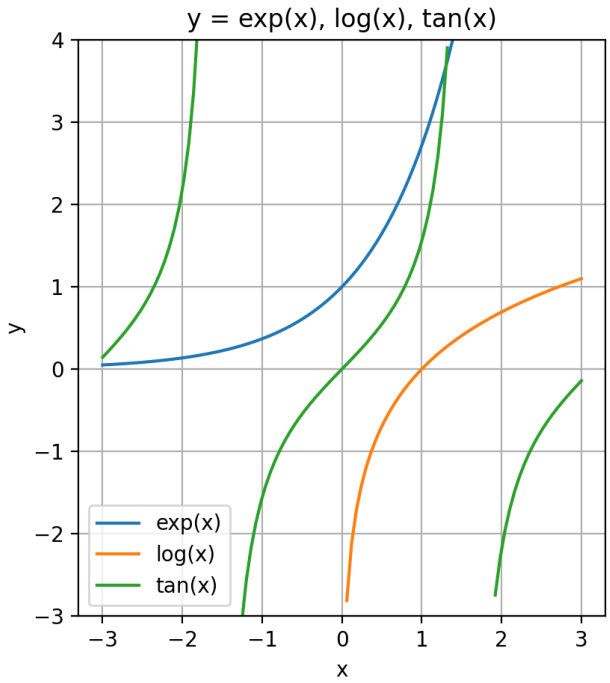
問題 6.2

```
# plot_functions.py: 複数のグラフを並べる+問題 6.2
import matplotlib.pyplot as plt # Matplotlib
import numpy as np # NumPy

# x 区間の分割数を指定
n = 100

# Figure オブジェクト(fig)と
# その上に Axes オブジェクト(ax, ax2)を
# 1 行 2 列に生成し，サイズを 10x5 とする
fig, (ax, ax2) = plt.subplots(1, 2, figsize=(10, 5))

# 三角関数のグラフを描く
# x の範囲[a, b]を指定
a = 0
b = 4 * np.pi
x_h = (b - a) / n # 小区間幅
x = [ a + i * x_h for i in range(n + 1)]

# x ごとに y の値を計算
y_sin = np.sin(x)
y_cos = np.cos(x)
#y_tan = np.tan(x)

# 三角関数グラフを描画
ax.plot(x, y_sin, label='sin(x)')
ax.plot(x, y_cos, label='cos(x)')
#ax.plot(x, y_tan, label='tan(x)')
ax.set_title('y = sin(x), cos(x)')
#ax.set_title('y = sin(x), cos(x), tan(x)')
ax.set_xlabel('x')
ax.set_ylabel('y')
ax.set_ybound(lower=-3, upper=3)
ax.legend()
ax.grid()

# 指数関数を対数関数を fig 上に描画
```

```
# x の配列を生成し，exp(x) と log(x) と tan(x) を計算
a = -3
b = 3
x_h = (b - a) / n # 小区間幅
x = [ a + i * x_h for i in range(n + 1)]
y_exp = np.exp(x)
y_log = np.log(x) # x < の時は NaN を含む
y_tan = np.tan(x) # x=pi/2 * k の時は+-Inf を含む

lower_val, upper_val = -3, 4 # グラフの最小，最大値

# はみ出るところは描画しない
for i in range(len(y_tan)):
    if y_tan[i] < lower_val * 1.1: y_tan[i] = -np.inf
    if y_tan[i] > upper_val * 1.1: y_tan[i] = np.inf

# 折れ線グラフを ax2 上に描画
ax2.plot(x, y_exp, label='exp(x)')
ax2.plot(x, y_log, label='log(x)')
ax2.plot(x, y_tan, label='tan(x)')
ax2.set_title('y = exp(x), log(x), tan(x)')
ax2.set_xlabel('x')
ax2.set_ylabel('y')
ax2.set_ybound(lower=lower_val, upper=upper_val)
ax2.legend()
ax2.grid()

# グラフを表示
plt.show()
```

実行結果：





考察：そのまま tan(x)のグラフを描画すると不連続点も描画してしまう。解決策として，グラフに入りきらない所に±∞を代入するようにした。

問題 6.3

```
# plot_diff_ploy.py: 数値微分と相対誤差の描画 問題 6.3
import numpy as np # NumPy
import matplotlib.pyplot as plt # Matplotlib

# 元の関数
def func(x):
    # return np.exp(np.cos(x)) - x ** 3
    return np.sin(np.exp(x))

# 真の導関数
def true_dfunc(x):
    #return -np.exp(np.cos(x)) * np.sin(x) - 3 * x ** 2
    return np.exp(x) * np.cos(np.exp(x))

# 前進差分商: (f(x + h) - f(x)) / h
def forward_diff(x, h, f):
    return (f(x + h) - f(x)) / h

# 中心差分商: (f(x + h) - f(x - h)) / 2h
```

```
def central_diff(x, h, f):
    return (f(x + h) - f(x - h)) / (2.0 * h)

# 後退差分商: (f(x) - f(x - h)) / h
def backward_diff(x, h, f):
    return (f(x) - f(x - h)) / h

# x = [-5, 5]
x = np.linspace(-5, 5, 100)
h = 10.0**(-5)

# 前進差分商，中心差分商，後退差分商
fdiff = forward_diff(x, h, func)
cdiff = central_diff(x, h, func)
bdiff = backward_diff(x, h, func)

# 相対誤差チェック
reldiff_f = np.abs((fdiff - true_dfunc(x)) / true_dfunc(x))
print('Forward diff relerr = ', reldiff_f)
reldiff_c = np.abs((cdiff - true_dfunc(x)) / true_dfunc(x))
print('Central diff relerr = ', reldiff_c)
reldiff_b = np.abs((bdiff - true_dfunc(x)) / true_dfunc(x))
print('Backword diff relerr= ', reldiff_b)

# 導関数のグラフ描画
true_dfunc_y = true_dfunc(x)
print('x                =', x)
print('True dfunc         =', true_dfunc_y)

fig, ax1 = plt.subplots()

# 関数グラフの y 軸（左）を設定
ax1_color = 'red' # 赤
ax1.set_title('f¥'(x) and Relerr of Numerical diff')
ax1.set_xlabel('x')
ax1.set_ylabel('f¥'(x)', color=ax1_color)
ax1.plot(x, true_dfunc_y, '--', label='f¥'(x)', color=ax1_color) # 赤の
点線
```

```
ax1.tick_params(axis='y', labelcolor=ax1_color) # y 軸に目盛を入れる

# 2 番目の y 軸（右）を設定
ax2 = ax1.twinx() # x 軸は共通
ax2.set_ylabel('Relative Error')
ax2.set_yscale('log') # log スケールに変更
ax2.plot(x, reldiff_f, 'o', label='Forward') # ○プロット
ax2.plot(x, reldiff_c, label='Central')
ax2.plot(x, reldiff_b, label='Backward')
ax2.tick_params(axis='y')

# 凡例
fig.legend(loc='upper left') #凡例を左上に配置

# グラフ描画
plt.show()
```

実行結果：