

情報処理演習グループ課題1

役職	学籍番号	名前	備考
リーダー	BQ23107	窪田 大輝	
プログラムリーダー	BQ23008	脇家 優太	
企画係	BQ23071	平川 奨	
連絡係	BQ23060	塚田 水月	
書記 1	BQ23110	信賀 晃	グループワークには参加したものの課題未提出
書記 2	BQ23103	山田 泰我	欠席

ロジスティック写像とは

ロジスティック写像は人口の推移を表す単純なモデルです。 $y = ax(1-x)$ のような二次関数で表されます。ここで、 a は定数を意味し、パラメータと呼ばれます。変数 x の値から y の値を計算し、得られた y の値を新しい x の値として式に代入することで新しい y を計算します。

C言語でのロジスティック写像プログラム

C言語での

すみません。こちらに写真を一枚一枚あげるのは面倒なので、アルバムにまとめてアップロードしました。木村教授の第一回の写像解説動画にて紹介された方法で、値を変更してプロットしてみたモノです。（値しかないものは x のみ、 $x \cdot y$ のみのグラフ作成に使用した数値です。）結果としては、教授の作成したdatファイルと作成方法が違う（visualstudio内でdatファイルに名称変更）ことや、値の表示方法のミスから、アルバム内の写真の通り同様の形のグラフがプロットされてしまいました。もしよろしければお使いください。（長文申し訳ありません）

```
#include <stdio.h>

int main() {

    int n, i;
    double a, x0, x;
    a = 1.5; //ロジスティック写像のパラメータ
    x0 = 0.5; //初期条件  $x_0 \in [0,1]$ 

    printf("0 %f %f\n", x0); //（回数, 値）

    n = 100;

    x = x0;

    for (i = 1; i <= n; i++) {
        x = a * x * (1 - x);
```

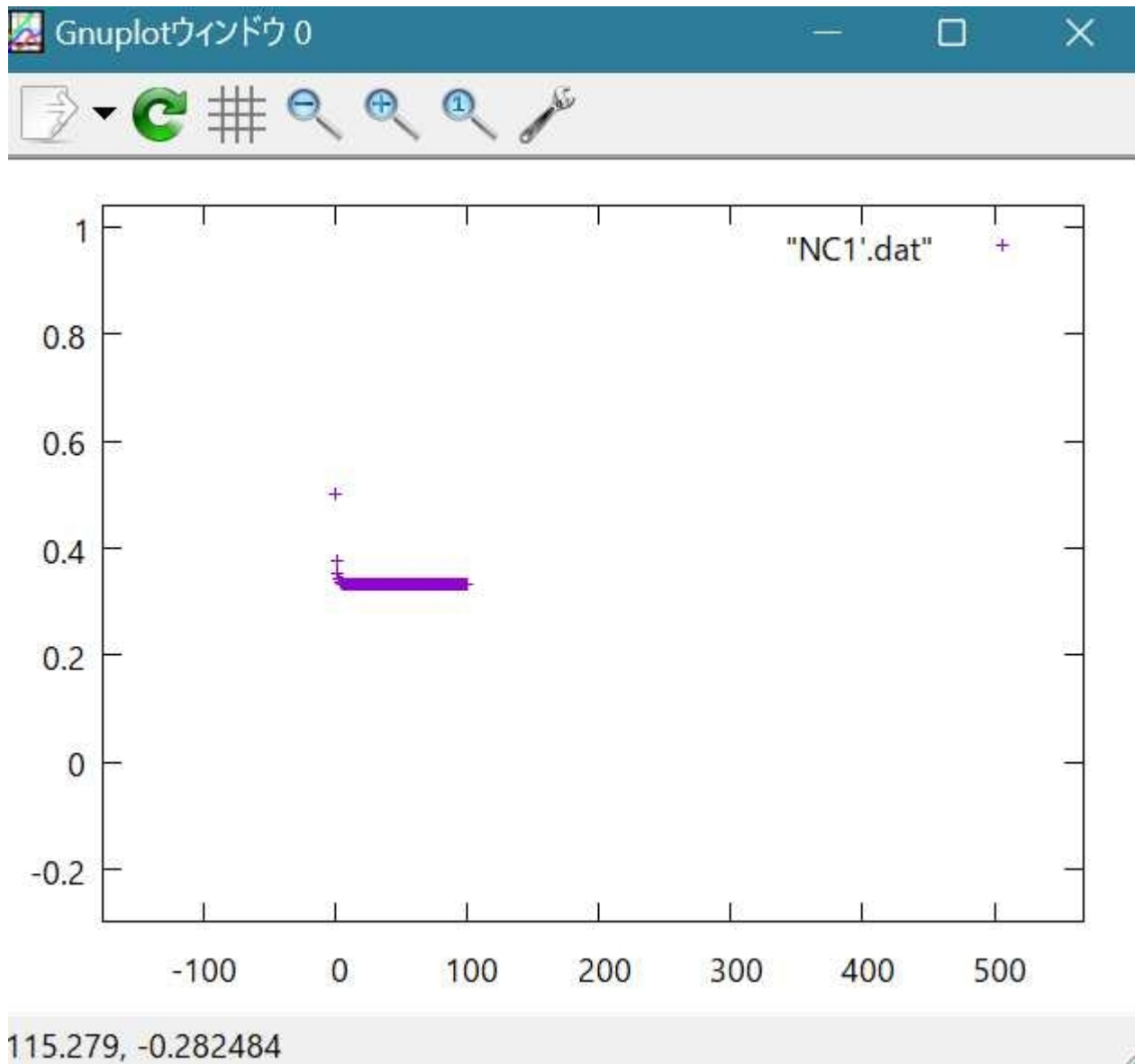
```
        printf("%d %f\n", i, x);  
    }  
  
}
```

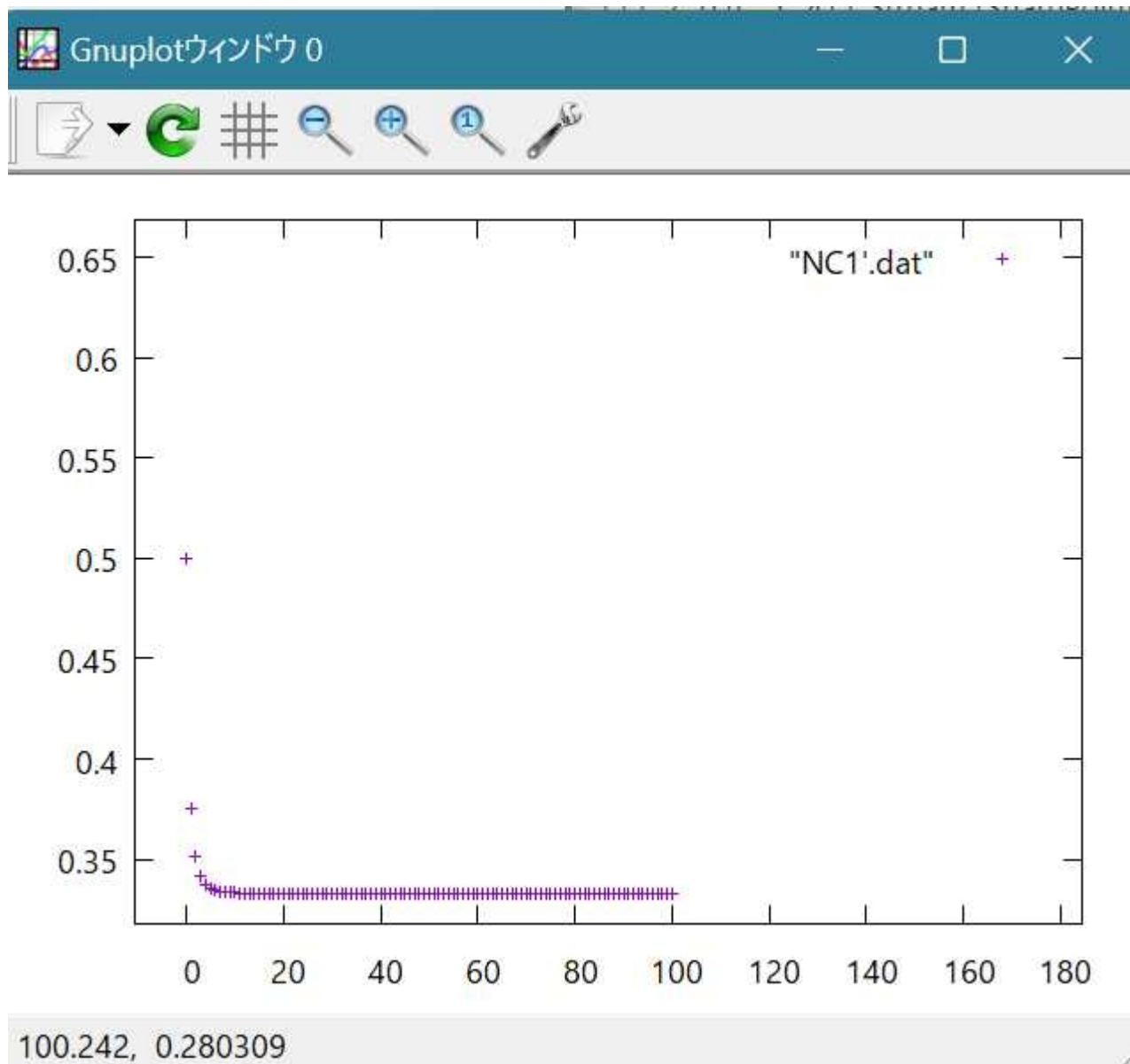
Fig.1 題名 : NC1.c

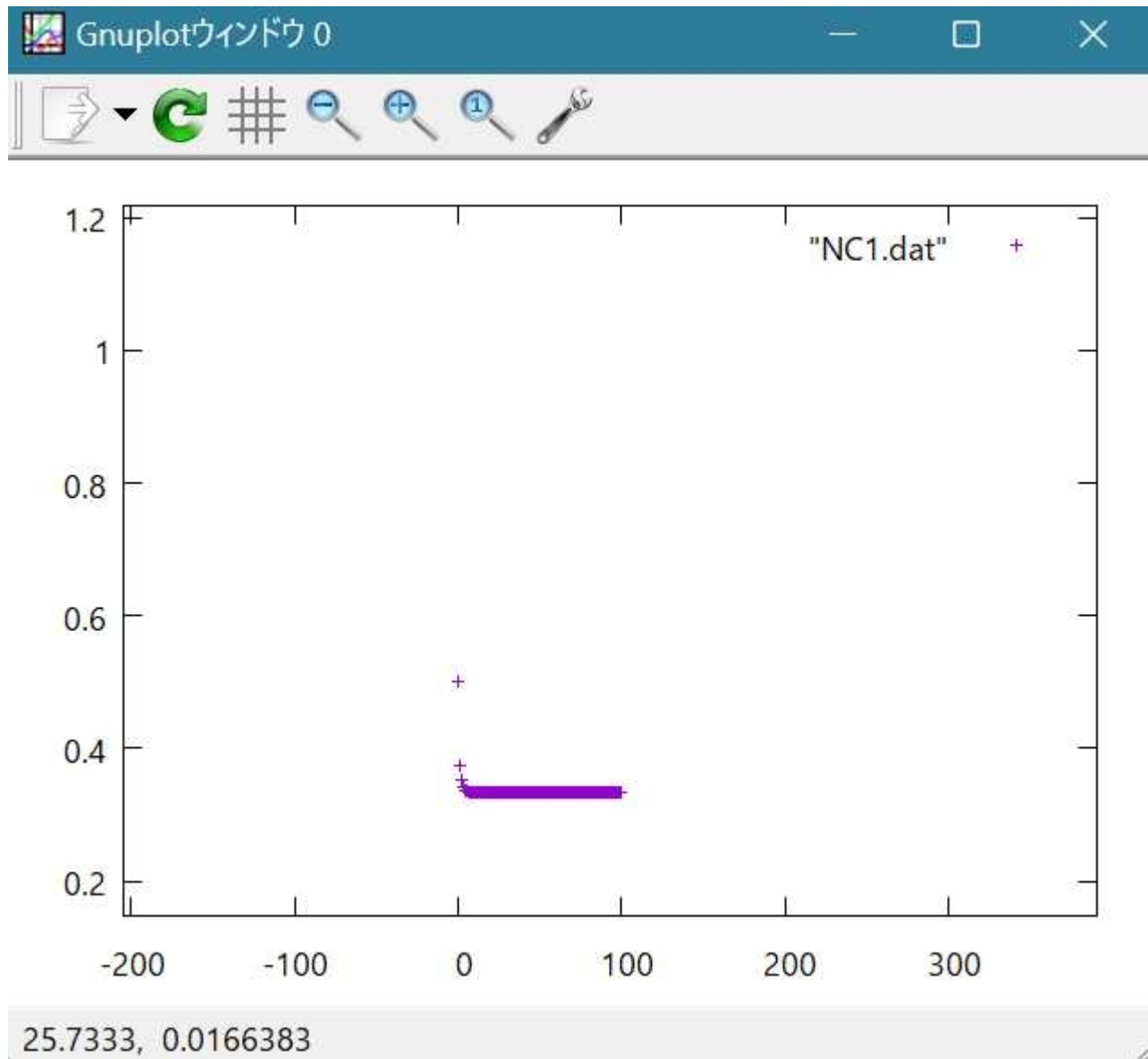
センタリングしたい文章

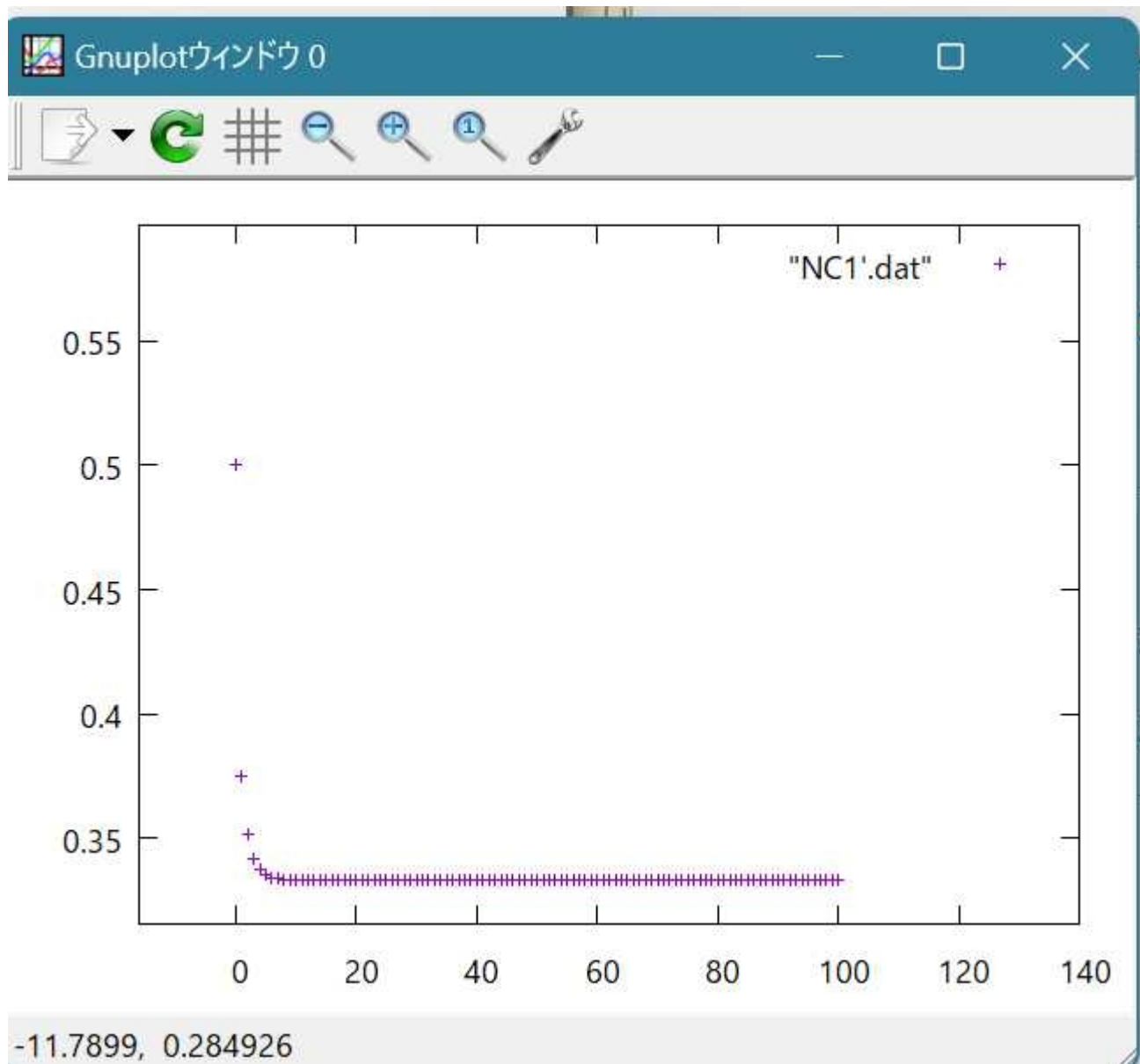
```
#include <stdio.h>  
  
int main() {  
  
    int n, i;  
    double a, x0, x,y0,y,e;  
    a = 1.5; //ロジスティック写像のパラメータ  
    x0 = 0.5; //初期条件 x0 ∈ [0,1]  
    e = 0.0001; //初期条件の差  
    y0 = x0 + e; //もう一つの初期条件;  
  
    printf("0 %f %f\n", x0, y0); // (回数, 値)  
  
    n = 100;  
  
    x = x0;  
    y = y0;  
  
    for (i = 1; i <= n; i++) {  
        x = a * x * (1 - x);  
        y = a * y * (1 - y);  
        printf("%d %f %f\n", i, x,y);  
    }  
  
}
```

題名 : NC1'.c









Pythonでのロジスティック写像プログラム

ここではPythonでのロジスティック写像プログラムを示す。

```
import matplotlib.pyplot as plt #グラフを描画するためにMatplotlibのpyplotモジュールを
pltとしてインポート

def logistic_map(a, x): #ロジスティクス写像のパラメータをa、現在の状態をxとおく
    return a * x * (1 - x)

def generate_logistic_map_sequence(a, x0, num_steps): #ロジスティック写像グラフを生
成する関数を定義。x0は初期条件、num_stepsは繰り返し回数
    result = [] #ロジスティクス写像の結果を格納するためのリスト
    current_x = x0 #current_xを初期条件x0に設定
    for _ in range(num_steps):
        result.append(current_x) #現在のcurrent_xの値を結果リストに追加
        current_x = logistic_map(a, current_x) #current_xを更新
    return result

def plot_logistic_map(a, x0, num_steps): #ロジスティック写像のシーケンスをプロットする
```

関数を定義

```
sequence = generate_logistic_map_sequence(a, x0, num_steps) #ロジスティック写像のシーケンスを生成
```

```
plt.plot(sequence, 'o-', markersize=2) #Matplotlibを使用してシーケンスをプロット, 'o-'は点と線でつながるスタイルを指定。
```

```
plt.title(f'Logistic Map (a={a}, x0={x0})')
```

```
plt.xlabel('Steps')
```

```
plt.ylabel('x')
```

```
plt.show()
```

#パラメータの設定

```
a_value=3.8 #ロジスティック写像のパラメータ
```

```
x0_value=0.5 #初期値
```

```
num_steps_value=100 #ステップ数
```

#ロジスティック写像のプロット

```
plot_logistic_map(a_value, x0_value, num_steps_value)
```

