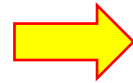


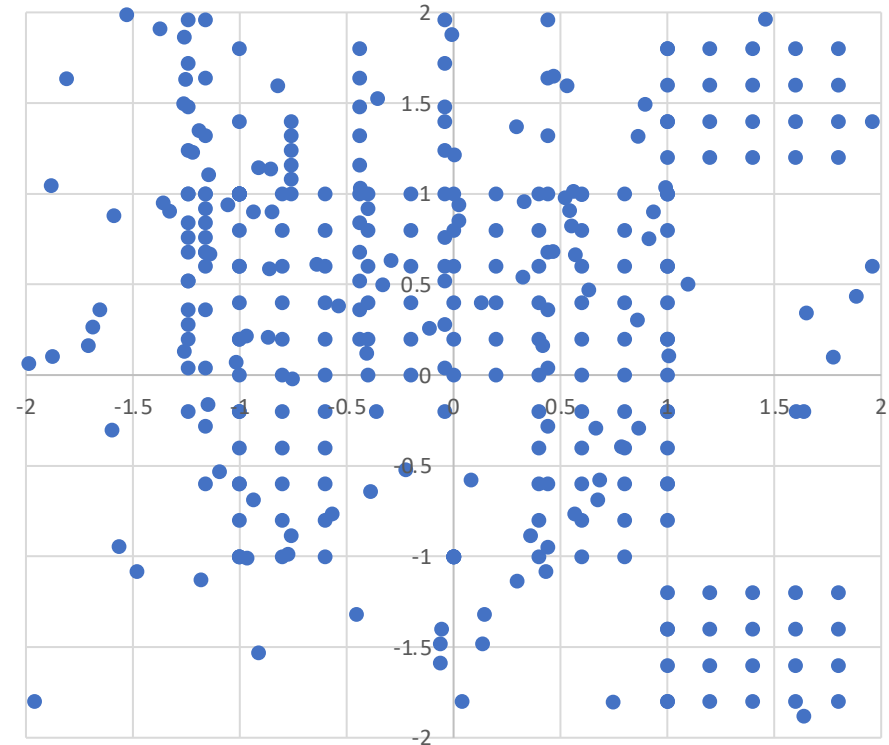
マンデルブロ集合

$$Z_{n+1} = Z_n^2 + C$$

	C		
	0	-1	$1+i$
Z_1	0	0	0
Z_2	0	-1	$1+i$
Z_3	0	0	$1+3i$
Z_4	0	-1	$-7+7i$
\vdots	\vdots	\vdots	$-9407+193i$
\vdots	\vdots	\vdots	\vdots



Excelでプロット



$$Z_2 = Z_1^2 + 0 \\ = 0^2 + 0$$

$$Z_2 = Z_1^2 - 1 \\ = 0^2 - 1 = -1$$

$$Z_3 = Z_2^2 - 1 \\ = (-1)^2 - 1 = 0$$

$$Z_2 = Z_1^2 + (1+i) \\ = 1+i$$

$$Z_3 = Z_2^2 + (1+i) \\ = (1+i)^2 + (1+i) \\ = 1+3i$$

マンデルブロ集合のアルゴリズム

```
size = 4; // 描く領域の一辺の長さ
pixel = 100; // 描く領域の一辺のピクセル数
for (i = 0; pixel > i; i++) { // x (実部) 方向のループ
    x = i * size / pixel - size / 2; // 定数Cの実部
    for (j = 0; pixel > j; j++) { // y (虚部) 方向のループ
        y = j * size / pixel - size / 2; // 定数Cの虚部
        a = 0; // くり返し計算に使う複素数zの実部
        b = 0; // くり返し計算に使う複素数zの虚部
        for (k = 0; 50 > k; k++) { // 上限を50回とするくり返し計算
            _a = a * a - b * b + x; // z^2+Cの計算 (実部)
            _b = 2 * a * b + y; // z^2+Cの計算 (虚部)
            a = _a; // zの値を更新 (実部)
            b = _b; // zの値を更新 (虚部)
            if (a * a + b * b > 4) { // もし絶対値が2を (絶対値の2乗が4を) 超えていたら
                draw(i, j); // (i, j)の位置のピクセルを「マンデルブロ集合でない色」で塗りつぶして
                break; // 次の点の計算へ
            }
        }
    }
}
```

Sub マンデルブロ集合()

Excelのマクロ

```
Dim a As Double, b As Double
```

```
Dim i As Integer, j As Integer, k As Integer
```

```
Dim amax As Double, amin As Double
```

```
Dim bmax As Double, bmin As Double
```

```
Dim x As Double, y As Double, xdummy As Double, ydummy As Double
```

```
Dim n As Integer, kmax As Integer
```

```
Application.ScreenUpdating = False
```

```
amax = 1.2
```

```
amin = -2
```

```
bmax = 1.2
```

```
bmin = -1.2
```

```
n = 200
```

```
kmax = 100
```

```
For i = 0 To n
```

```
For j = 0 To n
```

```
    a = amin + (amax - amin) * i / n
```

```
    b = bmin + (bmax - bmin) * j / n
```

```
    x = 0
```

```
    y = 0
```

```
    For k = 1 To kmax
```

```
        xdummy = x
```

```
        ydummy = y
```

```
        x = x * x - y * y + a
```

```
        y = 2 * xdummy * ydummy + b
```

```
        If Sqr(x * x + y * y) > 2 Then
```

```
            col = k
```

```
            Exit For
```

```
        End If
```

```
        col = 0
```

```
    Next k
```

```
    Worksheets("Sheet1").Range(Cells(j + 2, i + 2), Cells(j + 2, i + 2)).Interior.Color = RGB((col * 20) Mod 255, (col * 20) Mod 255, (col * 20) Mod 255)
```

```
Next j
```

```
Next i
```

```
Application.ScreenUpdating = True
```

```
End Sub
```

3-4行目: 変数aとbを宣言

5-6行目: 変数iとjを宣言

7-8行目: 変数amaxとaminを宣言し、初期値を設定

9-10行目: 変数bmaxとbminを宣言し、初期値を設定

11行目: 変数nを宣言し、分割数を指定

12行目: 変数kmaxを宣言し、最大反復回数を指定

14行目: For文でiを0からnまで繰り返す

15行目: For文でjを0からnまで繰り返す

16行目: 変数aを計算17行目: 変数bを計算

18-19行目: 変数xとyを初期化

20行目: For文でkを1からkmaxまで繰り返す

21行目: 変数xdummyとydummyに変数xとyの値を代入

22行目: 変数xを計算

23行目: 変数yを計算

24行目: もし $|x+y| > 2$ ならば、変数colに反復回数kを代入して、For文を抜ける

25行目: そうでなければ、変数colに0を代入

26行目: セルの背景色を指定して塗りつぶす

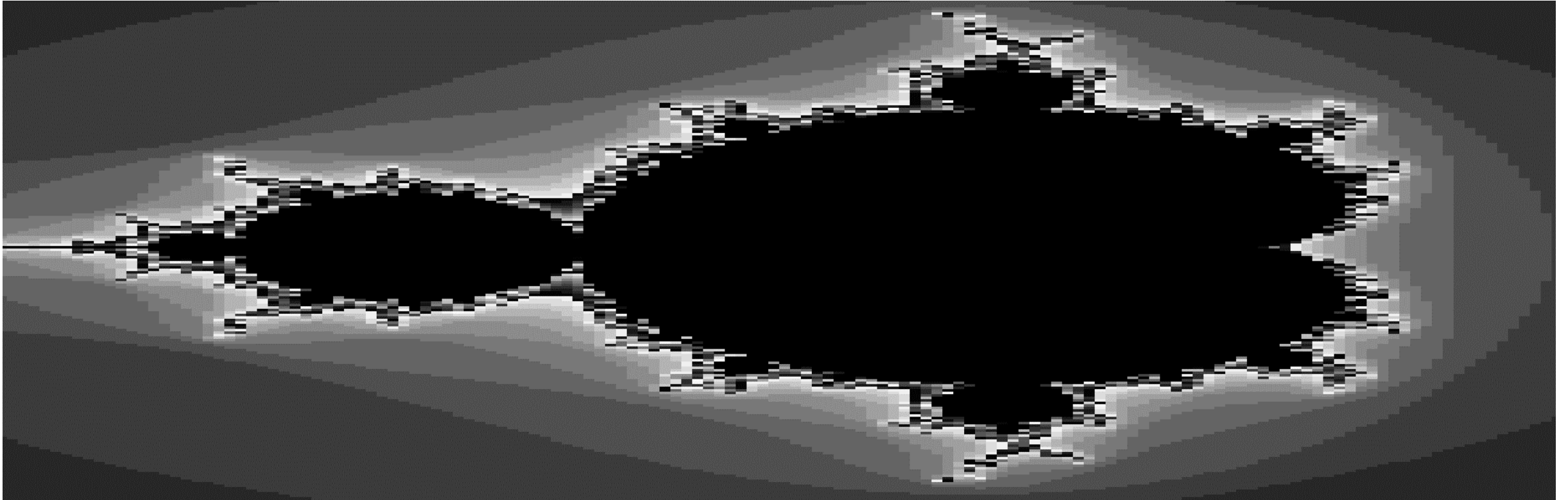
27行目: Next文でkのループを終了

28行目: Next文でjのループを終了

29行目: Next文でiのループを終了

31行目: スクリーン更新を有効化する

Excelのマクロの実行結果



pythonのコード

① ChatGPTにコード問い合わせ

→ 実行



② 以下のエラーのため、ChatGPTに問い合わせ

Image data of dtype object cannot be converted to floatというエラーがでました。なぜ？

↓ 以下の回答

```
import numpy as np
import matplotlib.pyplot as plt
def mandelbrot(c, maxiter):
```

```
    z = c
    for n in range(maxiter):
        if abs(z) > 2:
            return n
        z = z*z + c
    return 0
```

```
xmin, xmax, ymin, ymax = -2.5, 1.5, -2, 2
npoints = 1000
maxiter = 100
```

```
x, y = np.meshgrid(np.linspace(xmin, xmax, npoints), np.linspace(ymin, ymax, npoints))
```

```
c = x + y*1j
```

```
mandel = np.frompyfunc(mandelbrot, 2, 1)(c, maxiter)
```

```
mandel = mandel.astype(float)
```

```
plt.imshow(mandel, extent=[xmin, xmax, ymin, ymax], cmap='viridis')
```

```
plt.axis('off')
```

```
plt.show()
```



Image data of dtype object cannot be converted to float というエラーは、NumPy配列のデータ型が `object` に設定されている場合に発生することがあります。このエラーは、浮動小数点数型に変換できないデータ型を `imshow()` 関数に渡した場合に発生します。このエラーを解決するには、`mandel` 配列のデータ型を浮動小数点数に変換する必要があります。以下のように、`astype()` 関数を使用して、NumPy配列のデータ型を変換することができます：

```
mandel = mandel.astype(float)

plt.imshow(mandel, extent=[xmin, xmax, ymin, ymax], cmap='viridis')
```

これにより、配列のデータ型が `float` に変換され、エラーが解決されるはずです。

実行結果



描かれたマンデルブロ集合の色を変えたい場合は、pythonコードのどこを変えますか？

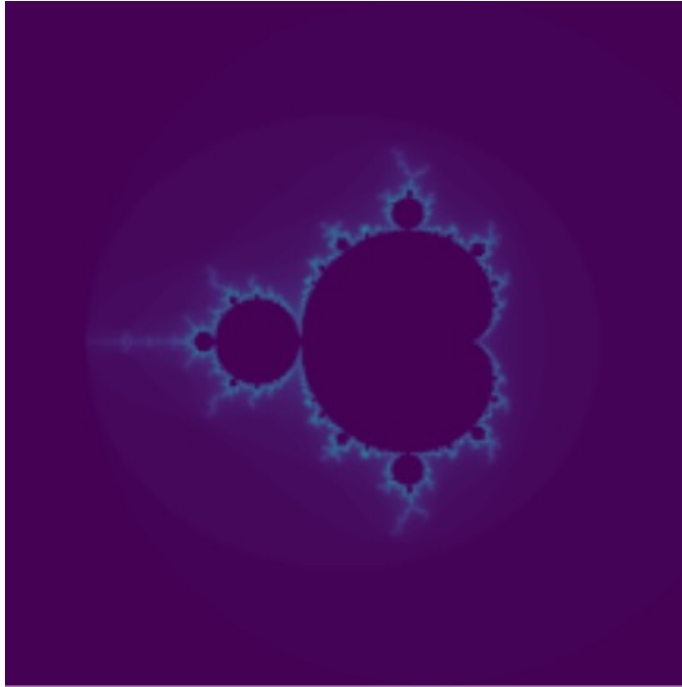
ChatGPTの回答

描かれたマンデルブロ集合の色を変更するには、`imshow()` 関数で使用するカラーマップを変更する必要があります。現在のコードでは、`cmap='viridis'` で定義されている「viridis」というカラーマップが使用されています。他のカラーマップを使用することもできます。例えば、以下のように「hot」カラーマップを使用すると、赤から黄色へのグラデーションになります。

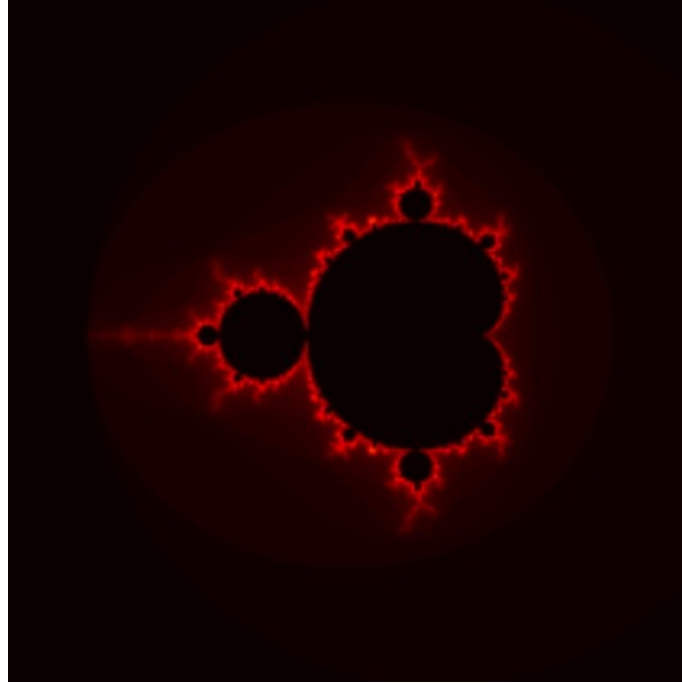
```
import numpy as np
import matplotlib.pyplot as plt
def mandelbrot(c, maxiter):
    z = c
    for n in range(maxiter):
        if abs(z) > 2:
            return n
        z = z*z + c
    return 0
xmin, xmax, ymin, ymax = -2.5, 1.5, -2, 2
npoints = 1000
maxiter = 100
x, y = np.meshgrid(np.linspace(xmin, xmax, npoints),
np.linspace(ymin, ymax, npoints))
c = x + y*1j
mandel = np.frompyfunc(mandelbrot, 2, 1)(c, maxiter)
mandel = mandel.astype(float) # データ型をfloatに変換
plt.imshow(mandel, extent=[xmin, xmax, ymin, ymax],
cmap='hot') # カラーマップを'hot'に変更
plt.axis('off')
plt.show()
```

```
import numpy as np
import matplotlib.pyplot as plt
from matplotlib.colors import LinearSegmentedColormap
# 青色から白色へのカスタムカラーマップを作成
colors = [(0.0, 0.0, 1.0), (1.0, 1.0, 1.0)]
cmap_name = 'blue_white'
cm = LinearSegmentedColormap.from_list(cmap_name, colors, N=256)
def mandelbrot(c, maxiter):
    z = c
    for n in range(maxiter):
        if abs(z) > 2:
            return n
        z = z*z + c
    return 0
xmin, xmax, ymin, ymax = -2.5, 1.5, -2, 2
npoints = 1000
maxiter = 100
x, y = np.meshgrid(np.linspace(xmin, xmax, npoints), np.linspace(ymin, ymax,
npoints))
c = x + y*1j
mandel = np.frompyfunc(mandelbrot, 2, 1)(c, maxiter)
mandel = mandel.astype(float) # データ型をfloatに変換
plt.imshow(mandel, extent=[xmin, xmax, ymin, ymax], cmap=cm) # カスタム
カラーマップを使用
plt.axis('off')
```

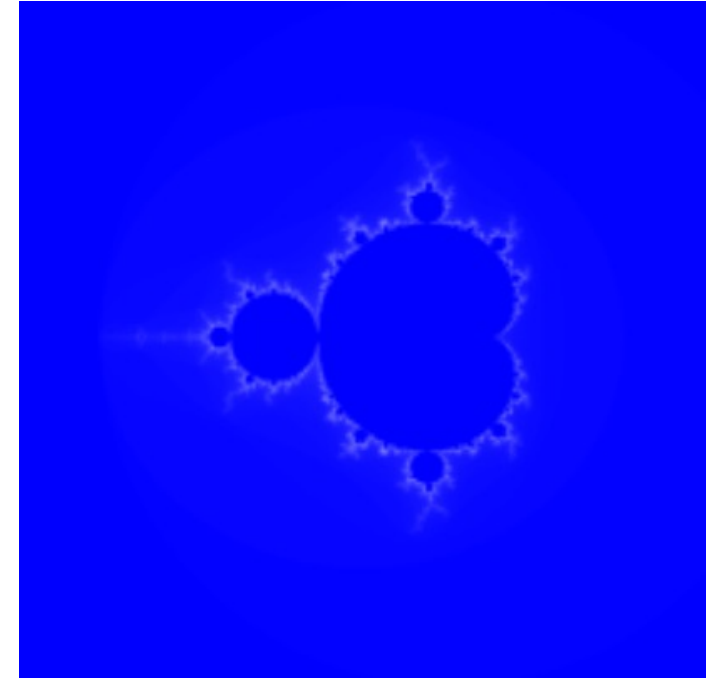
cmap='viridis'



cmap='hot'



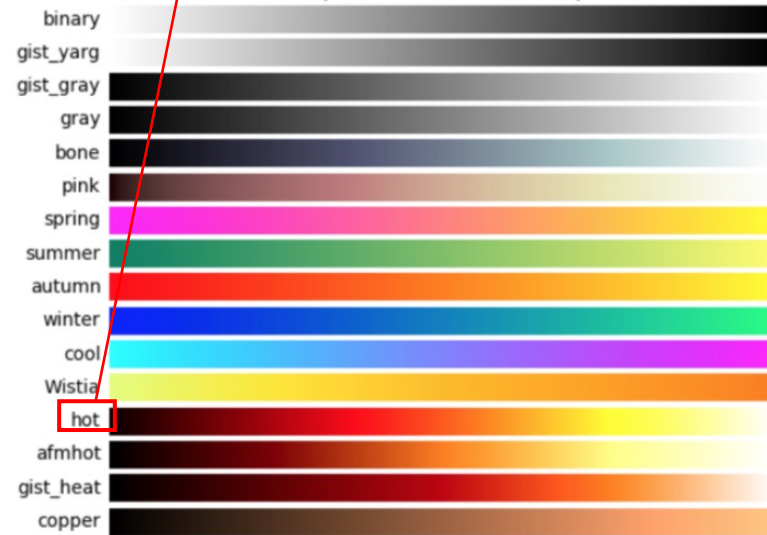
cmap_name = 'blue_white'



Perceptually Uniform Sequential colormaps



Sequential (2) colormaps



マンデルブロ集合の範囲指定して拡大したい。

マンデルブロ集合の範囲を指定して拡大するには、描画する範囲を変更する必要があります。具体的には、`imshow()` 関数に渡す `extent` パラメータを変更します。

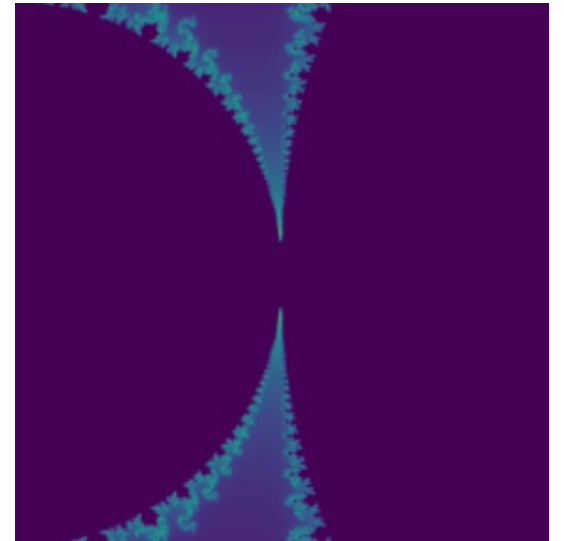
以下は、範囲を指定して拡大した例です。ここでは、最初に描画する範囲を `xmin=-2.5, xmax=1.5, ymin=-2, ymax=2` とし、その中心点を `(cx, cy)` に指定しています。次に、幅と高さがそれぞれ `wx, wy` である領域を描画するための新しい範囲を計算し、それを `extent` パラメータとして `imshow()` 関数に渡しています。これにより、指定された範囲内にマンデルブロ集合が描画されます。

ChatGPTの回答

```
import numpy as np
import matplotlib.pyplot as plt
def mandelbrot(c, maxiter):
    z = c
    for n in range(maxiter):
        if abs(z) > 2:
            return n
        z = z*z + c
    return 0
# 描画する範囲を指定
xmin, xmax, ymin, ymax = -2.5, 1.5, -2, 2
cx, cy = -0.75, 0
wx, wy = 0.5, 0.5 # 幅と高さ
npoints = 1000
maxiter = 100
# 拡大した領域の範囲を計算
x0, y0 = cx - wx/2, cy - wy/2
x1, y1 = cx + wx/2, cy + wy/2
x, y = np.meshgrid(np.linspace(x0, x1, npoints), np.linspace(y0, y1, npoints))
c = x + y*1j
mandel = np.frompyfunc(mandelbrot, 2, 1)(c, maxiter)
mandel = mandel.astype(float)
plt.imshow(mandel, extent=[x0, x1, y0, y1], cmap='viridis')
plt.axis('off')
plt.show()
```

この例では、中心点 (cx, cy) を中心に、幅 wx 、高さ wy の領域を描画します。 x_0, y_0, x_1, y_1 を計算し、`imshow()` 関数に渡すことで、指定された範囲が描画されます。

$cx, cy = -0.75, 0$



$cx, cy = -0.75, 0.4$

