

Exam2-2

October 25, 2021

1 Exam 2 Part 2

1.1 Visualizing the Mandelbrot and Julia Sets

```
[1]: import matplotlib.pyplot as plt
import numpy as np

iterations = int(input('iterations per point in the complex plane: '))
img_size = int(input('resolution of image (width & height): '))

real = np.linspace(-1.5,0.5,img_size)
imag = np.linspace(-1,1,img_size)
cplane = [[complex(r,i) for i in imag] for r in real]
mandelbrot = [[None for i in imag] for r in real]

r = 0
while r < img_size:
    i = 0
    while i < img_size:
        C = cplane[r][i]
        Z = complex(0,0)
        n = 0
        while n < iterations:
            Z = Z**2 + C
            if (abs(Z) > 3):
                mandelbrot[r][i] = False
                break
            n += 1
        else:
            mandelbrot[r][i] = True
        i += 1
    r += 1

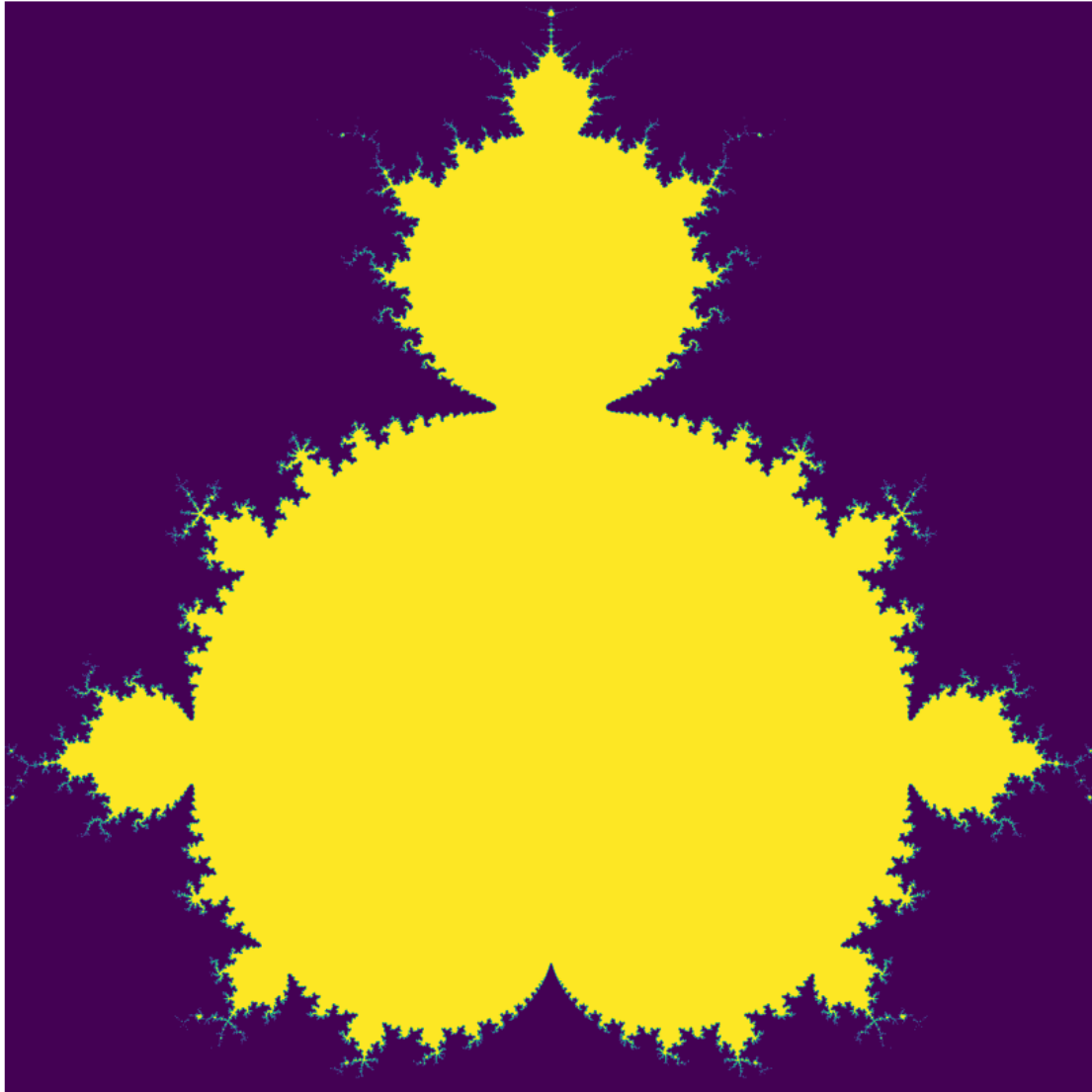
print('done!')
```

iterations per point in the complex plane: 30

resolution of image (width & height): 2048

done!

```
[3]: plt.figure(figsize = (15,15))
plt.axis('off')
plt.imshow(mandelbrot,cmap='viridis')
plt.show()
```



```
[40]: import matplotlib.pyplot as plt
import numpy as np

iterations = int(input('iterations per point in the complex plane: '))
img_size = int(input('resolution of image (width & height): '))
i = float(input('i: '))
j = float(input('j: '))
C = complex(i,j)
```

```

real    = np.linspace(-1.5,1.5,img_size)
imag    = np.linspace(-1.5,1.5,img_size)
cplane  = [[complex(r,i) for i in imag] for r in real]
julia   = [[None for i in imag] for r in real]

r = 0
while r < img_size:
    i = 0
    while i < img_size:
        Z = cplane[r][i]
        n = 0
        while n < iterations:
            Z = Z**2 + C
            if (abs(Z) > 3):
                julia[r][i] = False
                break
            n += 1
        else:
            julia[r][i] = True
        i += 1
    r += 1

print('done!')

```

```

iterations per point in the complex plane: 30
resolution of image (width & height): 2048
i: 0.4
j: 0.2

done!

```

```

[41]: plt.figure(figsize = (15,15))
      plt.axis('off')
      plt.imshow(julia,cmap='magma')
      plt.show()

```

