Homework 2
Mandelbulb

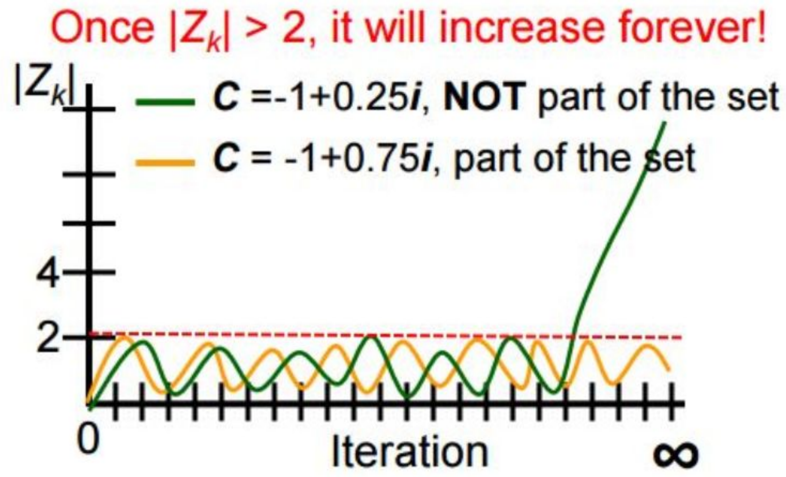
Introduction to Parallel Computing
2022/03/15

https://hackmd.io/@ipc22/hw2

Mandelbrot Set

A set of complex numbers $\mathbb{C}$

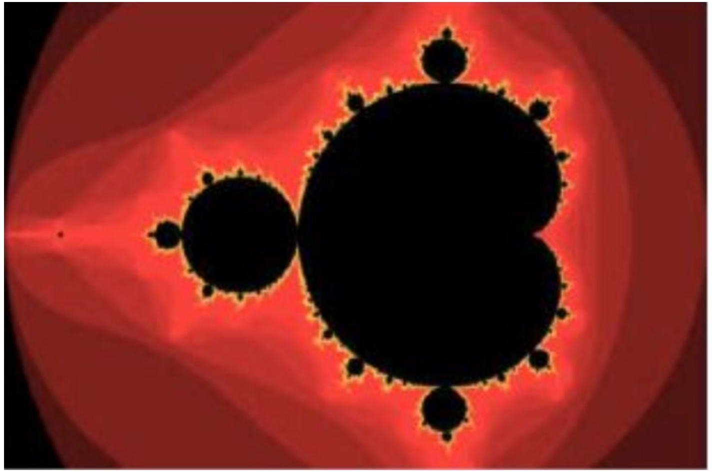
- for every complex number $c \in \mathbb{C}$, under iterations of quadratic map $Z_{k+1} = (Z_k)^2 + c$ remain bounded
  - $Z_0 = c$
  - $Z_{k+1} = (Z_k)^2 + c$
  - $|Z_k| \leq 2$



- if $|Z_k| \leq 2$ for any k, c belongs to the Mandelbrot Set

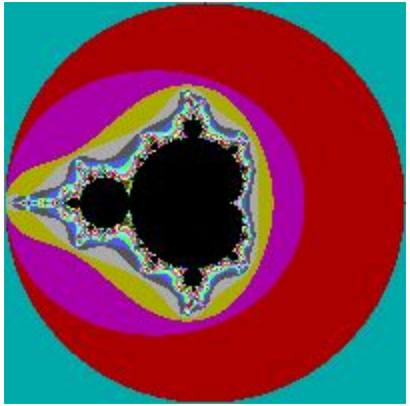
Mandelbrot Set Visualization

- Convert each pixel to the corresponding coordinates on the complex plane
- Plug into the equation repeatedly until $|Z_k| > 2$
- Color the pixel according to the iteration count
- https://www.youtube.com/watch?v=IrYfMfUURYM

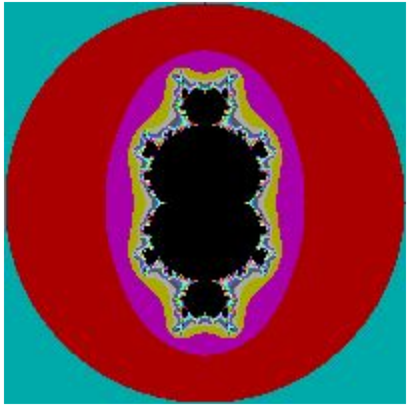


Powers of Mandelbrot Set

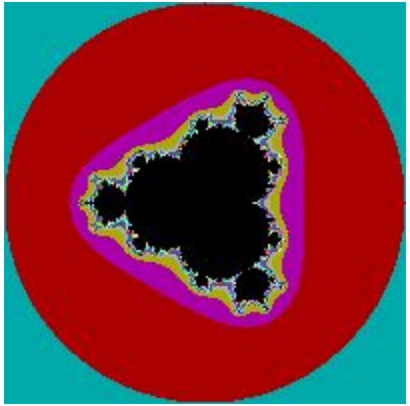
$Z_{k+1} = (Z_k)^2 + c$



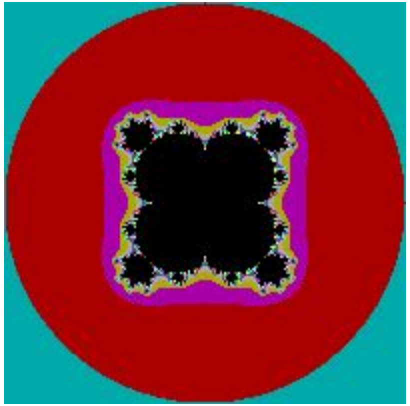
$Z_{k+1} = (Z_k)^3 + c$



$Z_{k+1} = (Z_k)^4 + c$



$Z_{k+1} = (Z_k)^5 + c$



Mandelbulb

- 3D fractal using spherical coordinates.
- In this assignment, we refer to power-8 mandelbulb

$$v_{k+1} = v_k^8 + C$$

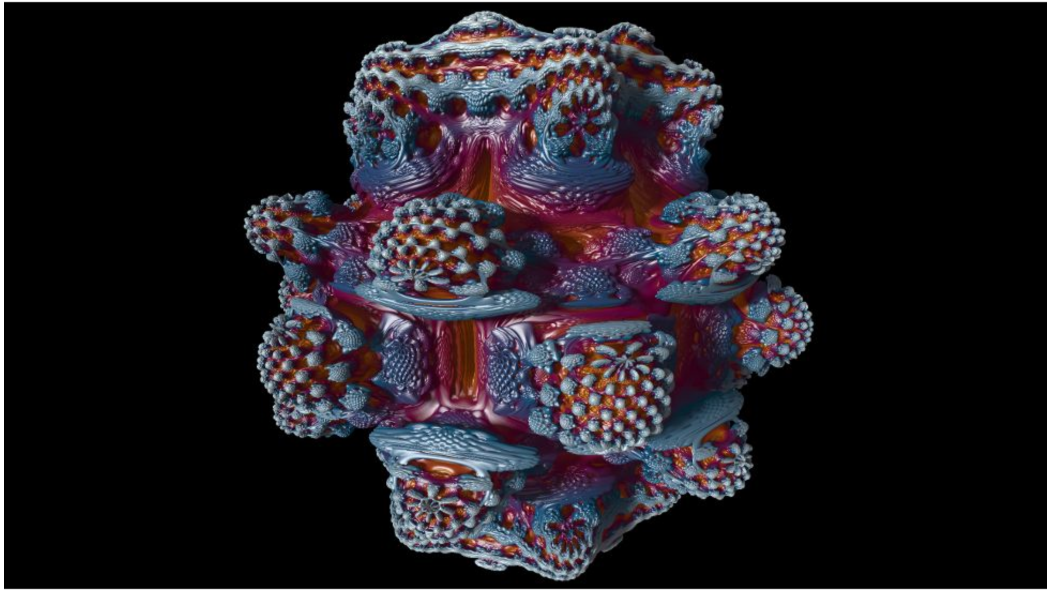
$$v = \langle x, y, z \rangle \text{ in } \mathbb{R}^3,\; v^n := r^n \langle \cos(n\theta)\cos(n\phi), \cos(n\phi)\sin(n\theta), -\sin(\phi) \rangle$$

- $r = \sqrt{x^2 + y^2 + z^2},\; \theta = \arctan\left(\frac{y}{x}\right),\; \phi = \arctan(\frac{z}{r})$

$$x = r\sin(\phi)\cos(\theta), y = r\sin(\phi)\sin(\theta), z = r\cos(\phi)$$

Mandelbulb Visualization

- Generate 3D images by ray tracing
- We use ray marching algorithm



Ray Tracing

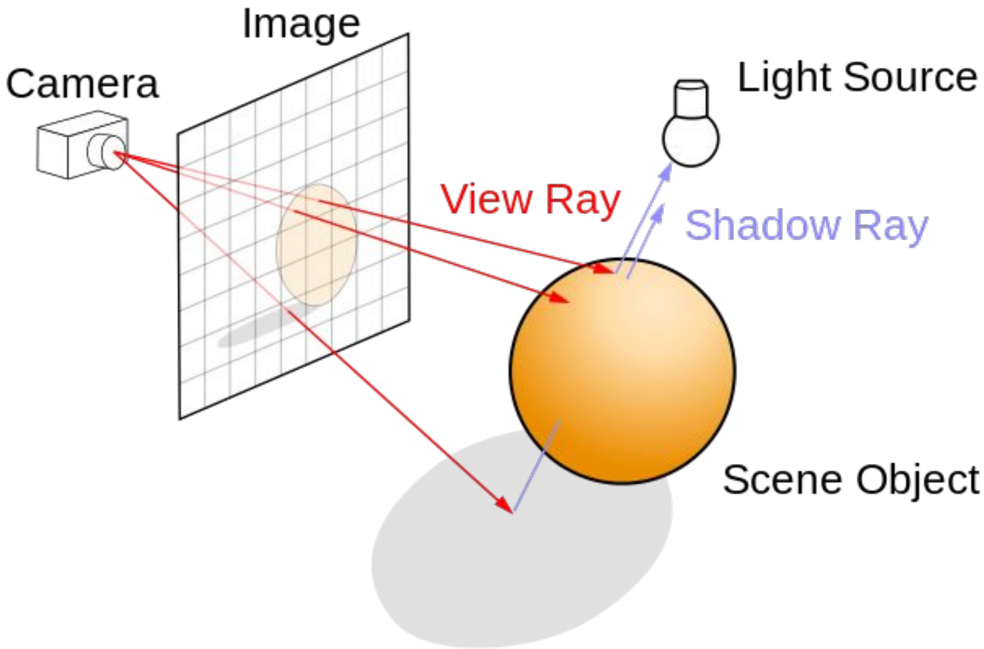
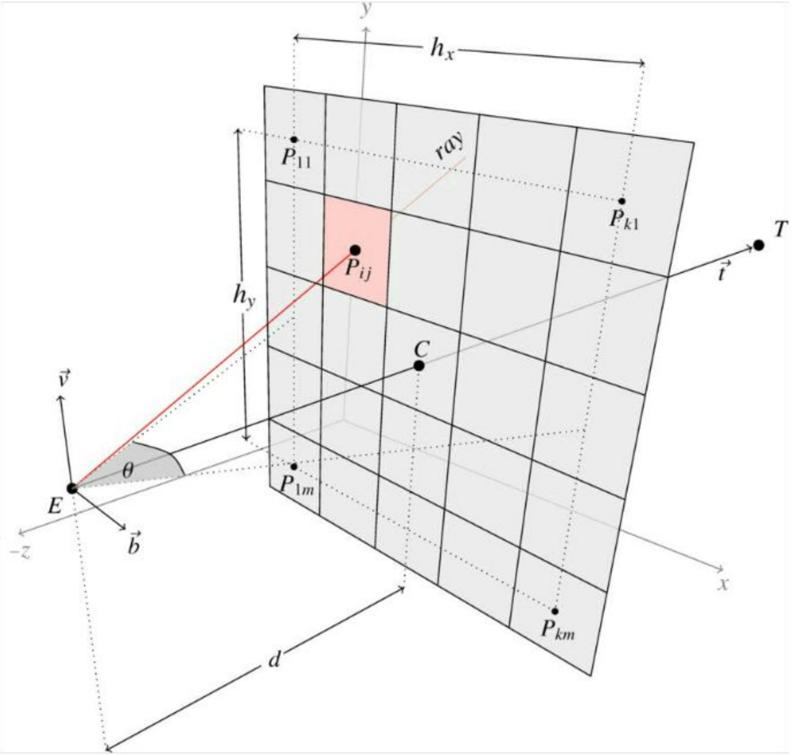


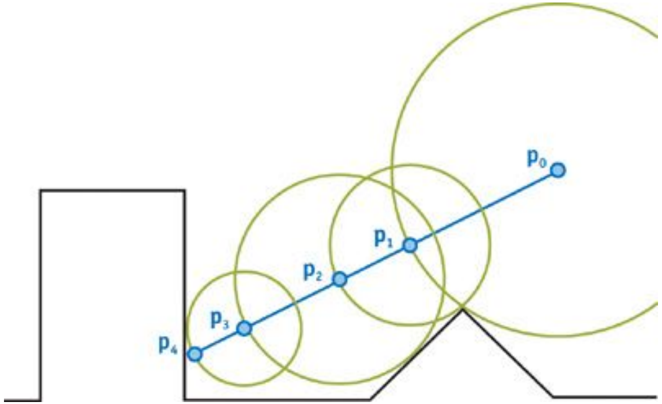
Image
Camera
Light Source
View Ray
Shadow Ray
Scene Object


y
h_x
ray
P_{11}
P_{k1}
P_{ij}
h_y
T
$\vec{t}$
C
$\vec{v}$
θ
E
$\vec{b}$
$-z$
P_{1m}
x
P_{km}
d

Ray Marching

[Often used for 3D fractal rendering]

1. Start at the “beginning” of the ray
2. Evaluate the distance function to estimate how close is to the object
3. Keep moving forward, the step should be short enough to not tunnel through the surface



Distance Function for Ray Marching

The approximate distance function of the mandelbulb is:

$$DE = \frac{0.5r\ln(r)}{dr}$$

Where $r = |v_k|$ and $dr = |v_k'|$.

We can get dr by scalar derivative $dr_{k+1} = n|v_k|^{n-1}dr_k + 1$ and $dr_0 = 1$

Goal

- We provide a sequential version of sample code called `hw2.cc`
- You are asked to parallelize it by MPI and OpenMP (or pthread)
- Understand the importance of Load Balancing

Input

```
./executable $c $x1 $y1 $z1 $x2 $y2 $z2 $width $height $filename
```

- $c int Number of thread per process
- $x1 double camera position x
- $y1 double camera position y
- $z1 double camera position z
- $x2 double camera target position x
- $y2 double camera target position y
- $z2 double camera target position z
- $width unsigned int width of the image
- $height unsigned int height of the image
- $filename string file name of the output PNG image

Output

- Save the result to `$filename`
- The output image should be a 32bit PNG image with RGBA channels.

Resources

- `/home/ipc22/share/hw2/`
  - `hw2.cc`
  - `Makefile`
  - `samples/`

Execute

- Check `samples/xx.txt`
- `01.txt:`
  - `N         = 2`
  - `n         = 3`
  - `c         = 4`
  - `pos       = -0.522 2.874 1.340`
  - `tarpos    = 0 0 0`
  - `width     = 64`
  - `height    = 64`
  - `timelimit = 5`

```
srun -N 2 -n 3 -c 4 \
./hw2 4 -0.522 2.874 1.340 0 0 0 64 64
1.png
```

Launch 3 processes on 2 nodes

Each process has 4 CPUs

Judge

- `hw2-judge`
- Scoreboard: https://apollo.cs.nthu.edu.tw/ipc22/scoreboard/hw2/

Report

- Explain your implementation, especially in the following aspects
  - How do you implement your program, what scheduling algorithm did you use: static, dynamic, guided, etc.?
  - How do you partition the task?
  - What techniques do you use to reduce execution time?
  - Other efforts you make in your program.
- Analysis
  - Design your own plots to show the load balance of your algorithm between threads/processes.
  - If you have modified the default parameter settings, please also compare the results of the default settings and your settings
- Conclusion

Submission

- Due: Tue, 2022/3/29 23:59
- Submit the following files to EEClass:
  - hw2.cc
  - report.pdf
  - Makefile (optional)

Q & A

Feel free to ask if you have any questions.