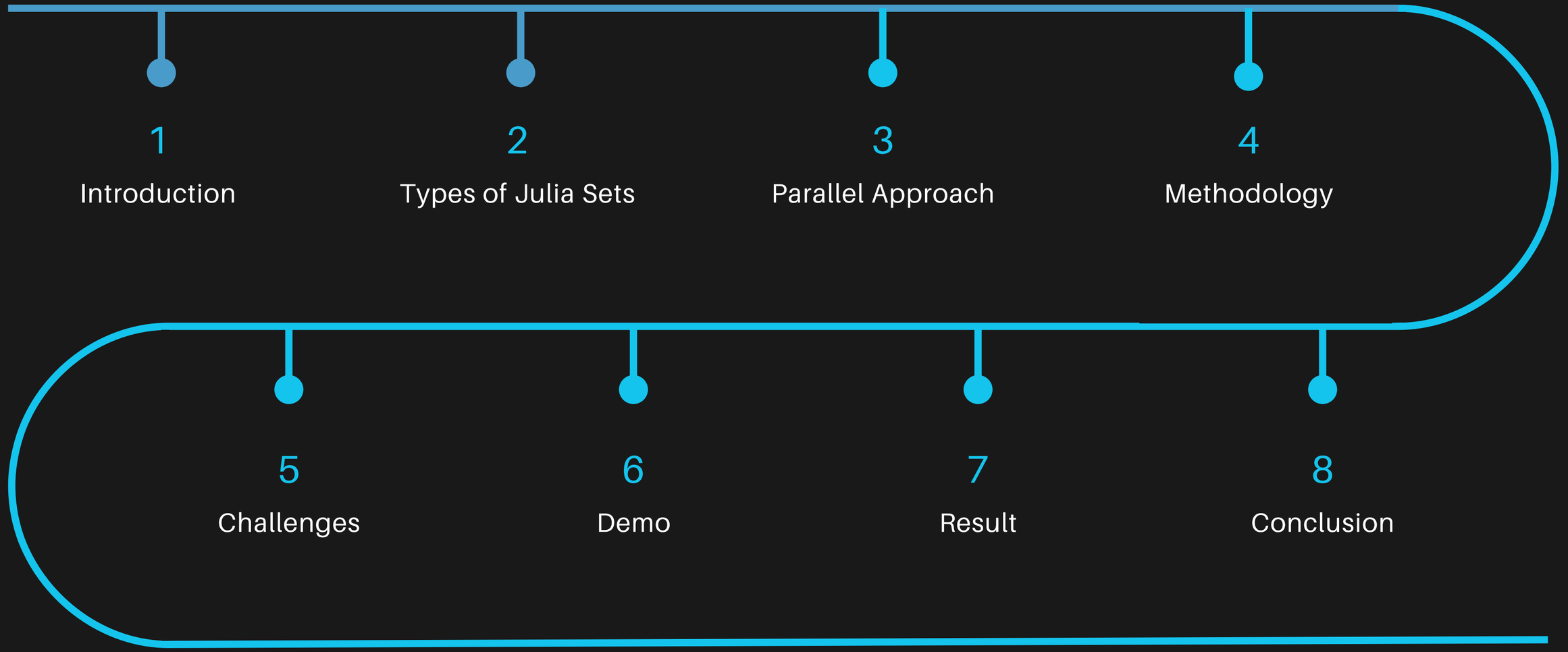
Group 1

Parallel Programming

Julia Sets Generation

WILLIAM MABIA, GRANT WESTERHOLM,
BRANDON PARKER, RHEA SHARMA, TAHA AMIR
RILEY ADAMS, TORIN BORTON-MCCALLUM

Outline


1
Introduction
2
Types of Julia Sets
3
Parallel Approach
4
Methodology
5
Challenges
6
Demo
7
Result
8
Conclusion

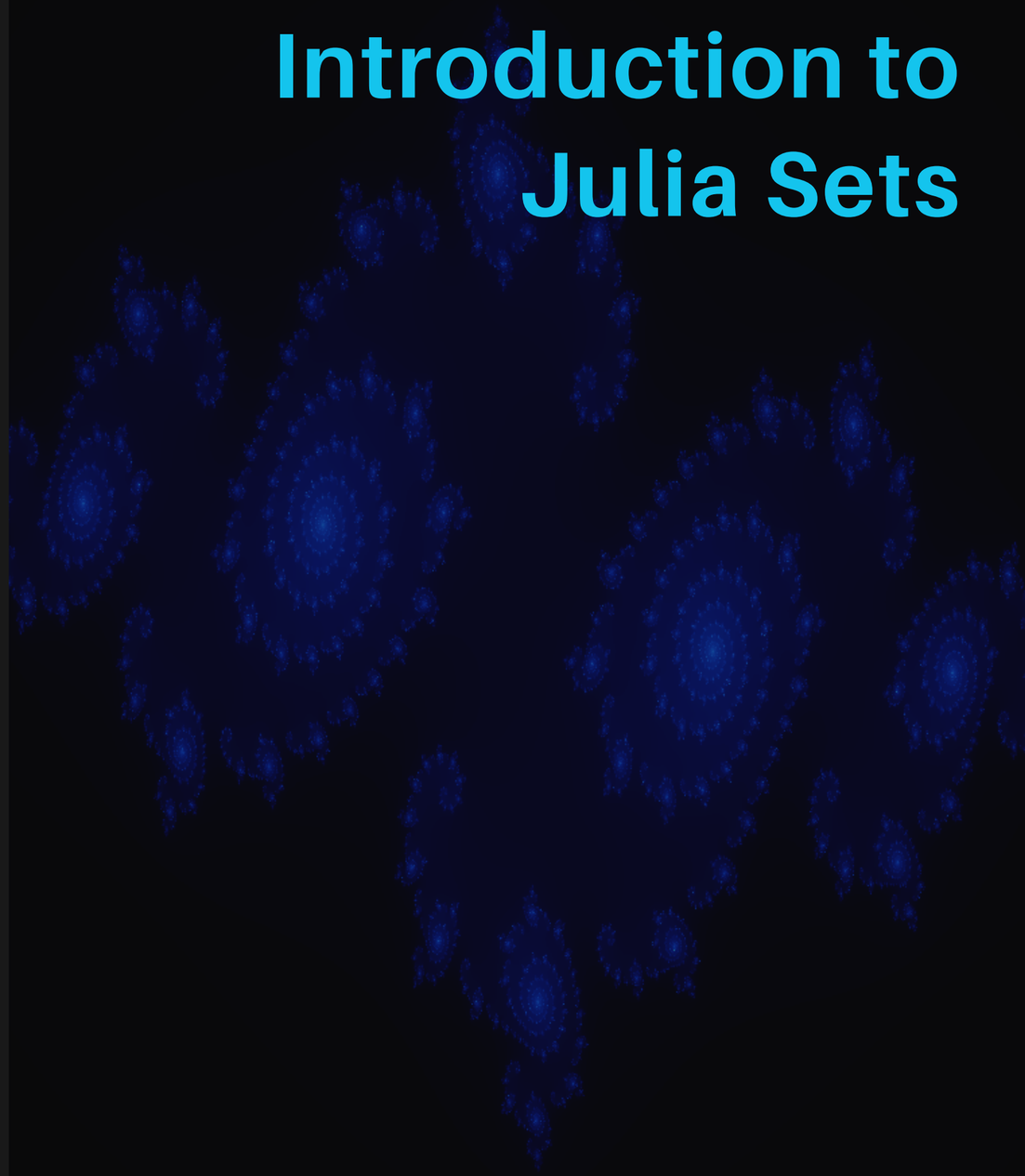
Introduction to Julia Sets

Definition:
Fractals generated by iterating complex numbers

Visualization:
Colour-coded images or animations illustrate intricate patterns

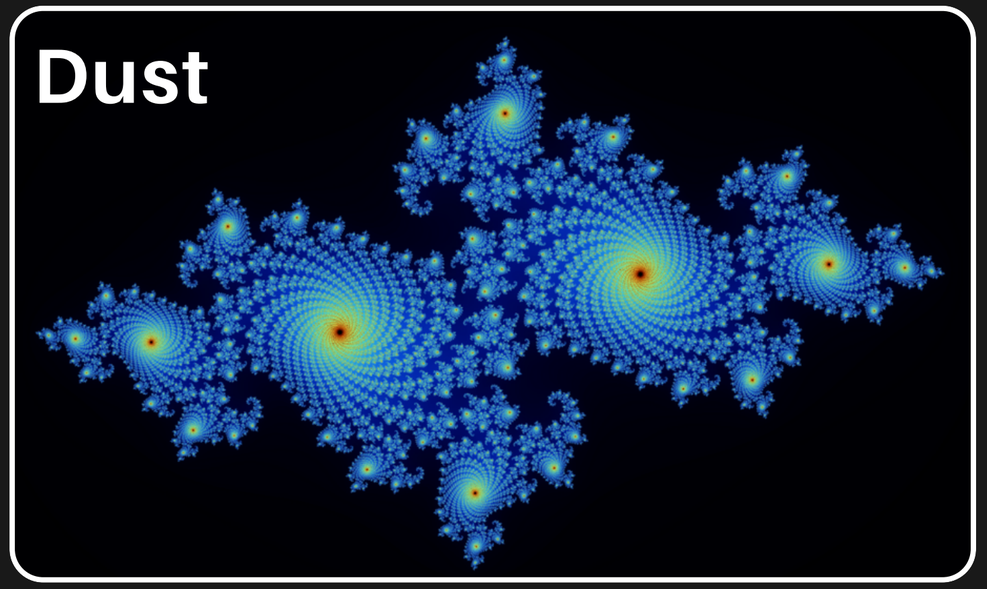
Properties:
Self-similarity, infinite complexity, and non-repeatability

Mandelbrot Set:
Each point in the Mandelbrot set corresponds to a different Julia set



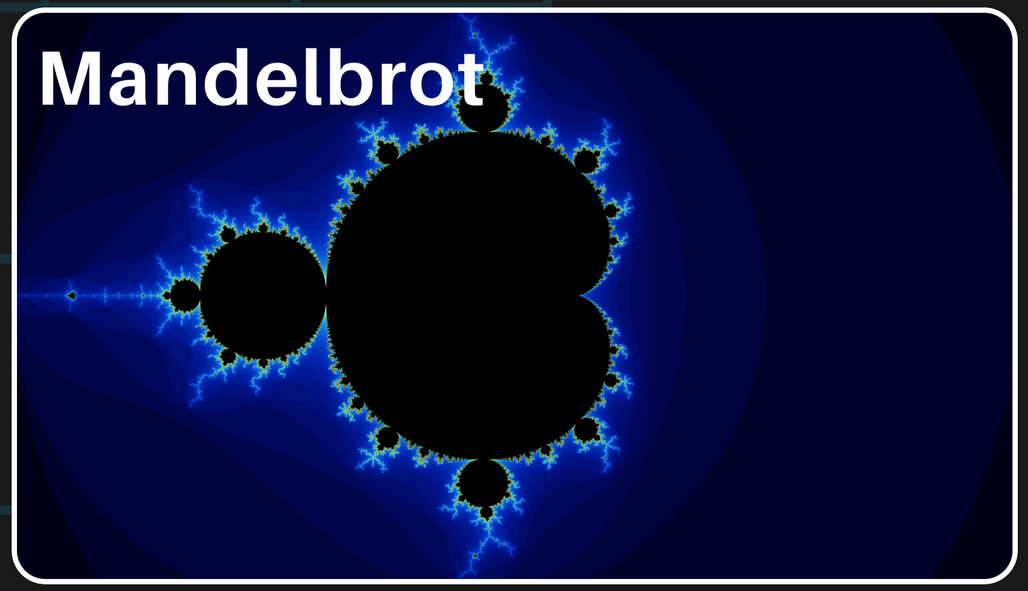
Visualizing Julia sets provides insights into the complex dynamics of mathematical functions and fractals

Types of Julia Sets



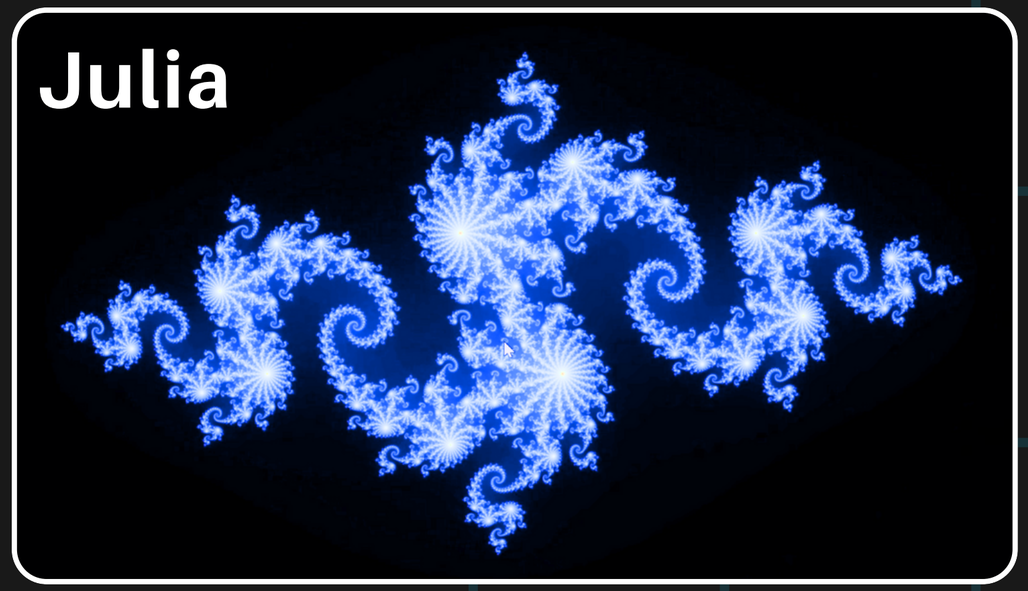
Mandelbrot vs Julia Set

$$z = z^2 + c$$



z = 0

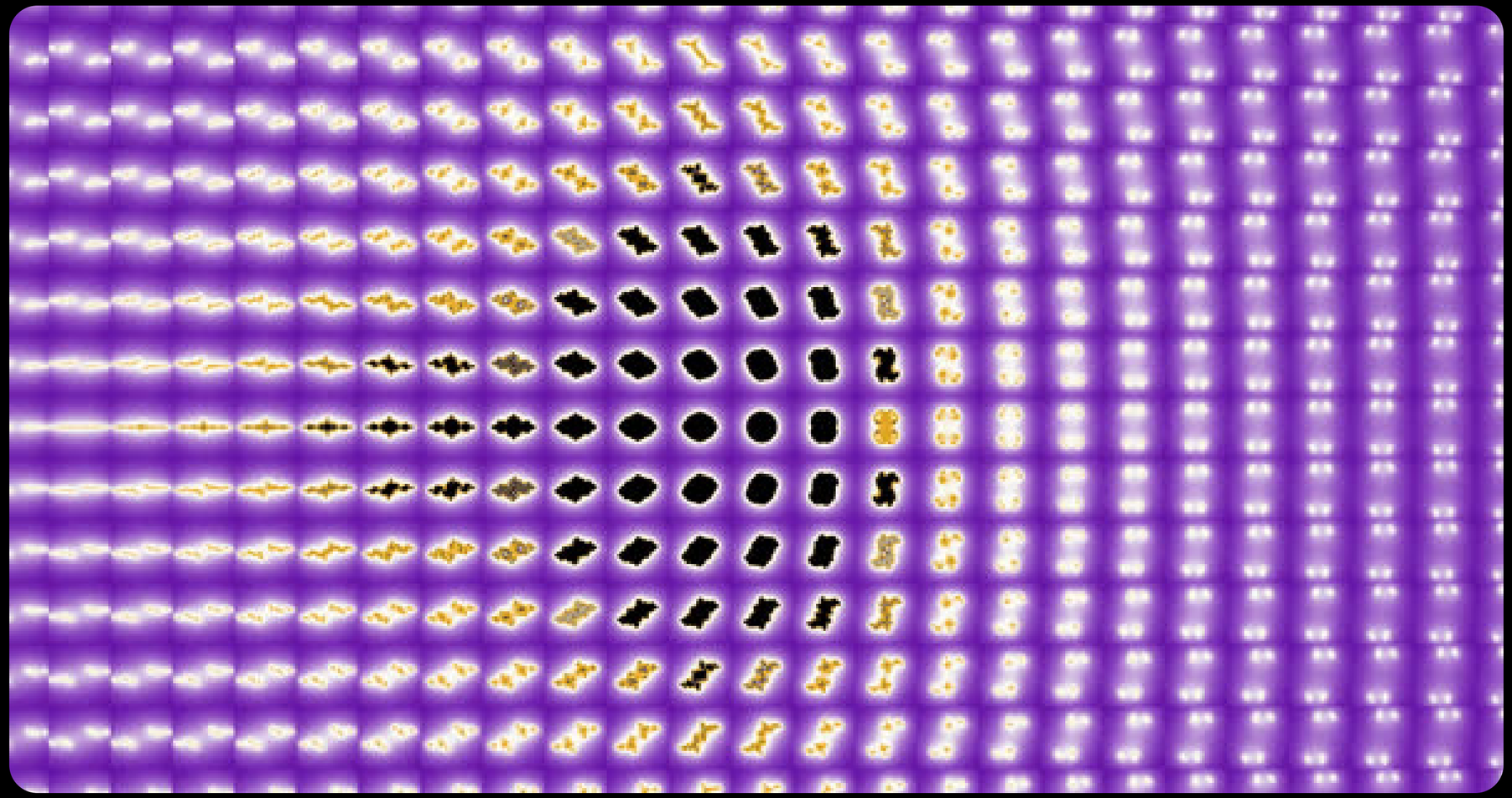
c = pixel location



z = pixel location

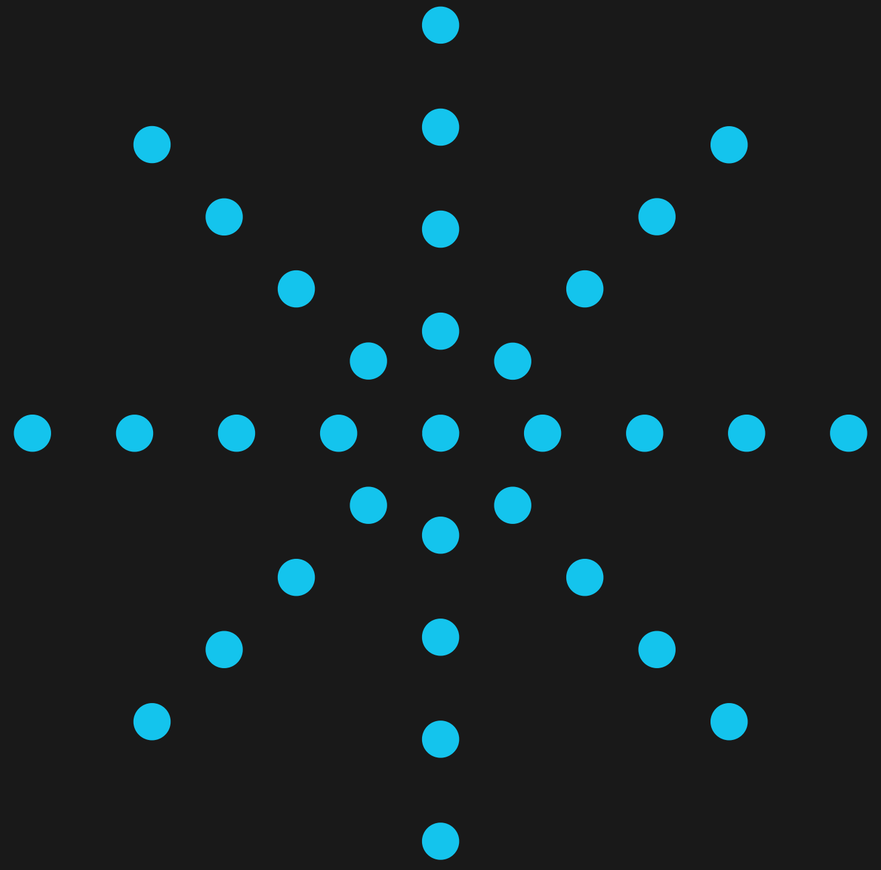
c = constant

Mandelbrot and Julia Set



Parallel MPI Approach to Julia Set Rendering

Our goal is to use the capabilities of parallel programming with MPI to efficiently generate high-resolution images of Julia sets.



- Julia sets are **fractals** produced from complex quadratic polynomials
- Fractal creation is resource-intensive, making it a perfect candidate for parallelization

Methodology

- Project executed on SHARCnet high-performance computing cluster with strict adherence to computational etiquette
- Programmed in C for MPI which enabled the distribution of computations across multiple processes
- PNG library used for rendering computed Julia sets into high-resolution images that show intricate details of fractals

Methodology

ALGORITHM OVERVIEW

We used a Message Passing Interface (MPI) to distribute the computation of the Julia and Mandelbrot sets across multiple processes.

PARALLELIZATION STRATEGY

We divided the computational task among various MPI processes, assigning each a portion of the grid representing the complex plane.

VISUALIZATION TECHNIQUE

We mapped iteration counts or divergence tests from the computation to the colours in the fractals. This lets us generate detailed images that reveal the intricate patterns of both the Julia and Mandelbrot sets.

HOW WE DID IT

Challenges

MEMORY MANAGEMENT

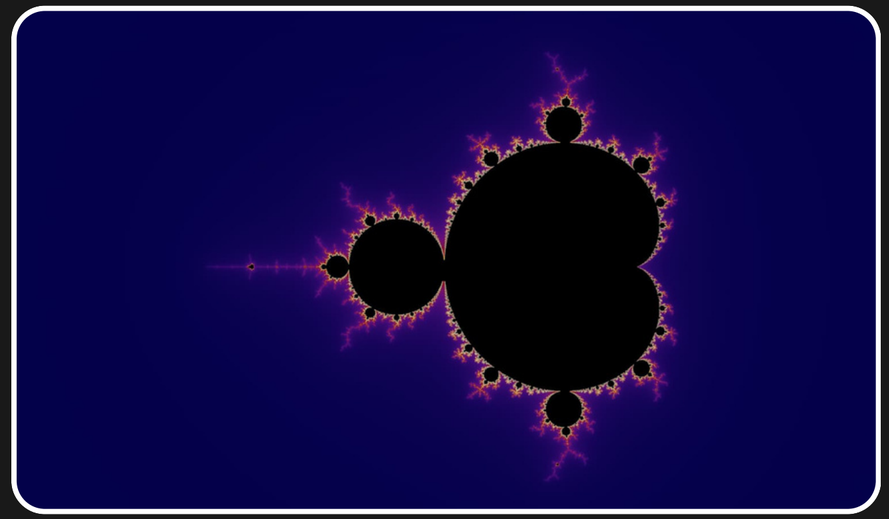
A chunk-based approach deals with high memory demands

POSITION AND ALIGNMENT

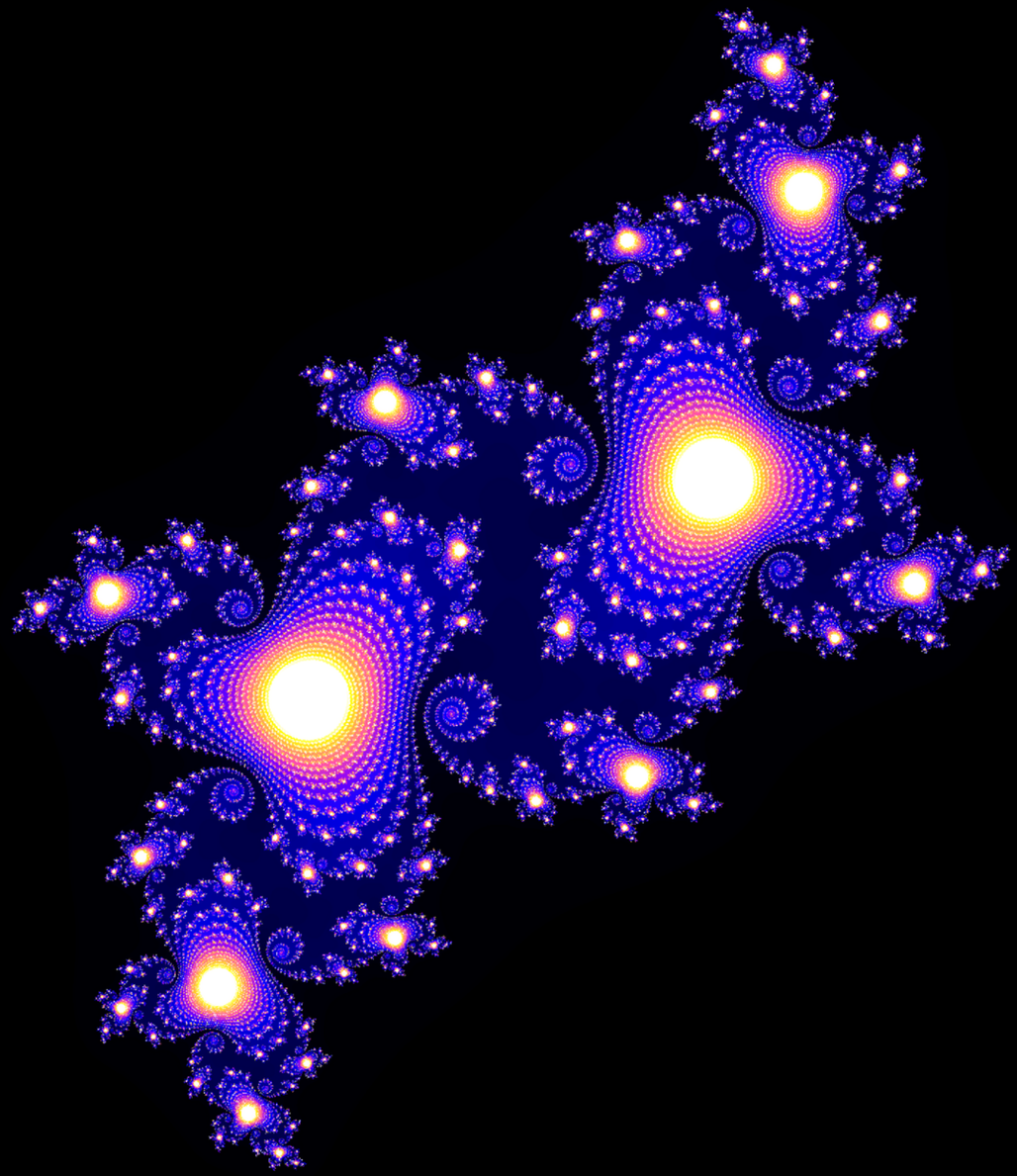
Adjusted scale and translation parameters allows for precise alignment

COLOUR MAPPING

Iterative testing helps identify colour schemes that balance aesthetics and clarity



Demo



Results

PERFORMANCE IMPROVEMENT

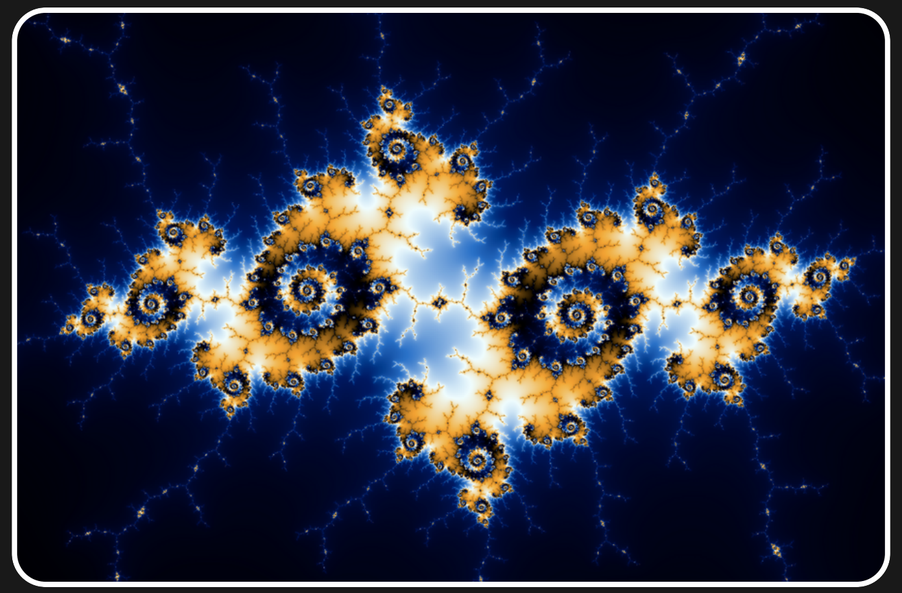
Faster results thanks to parallelization

GENERATED IMAGES

Generate sets that display clear images and are intuitive to use and explore

INSIGHTS

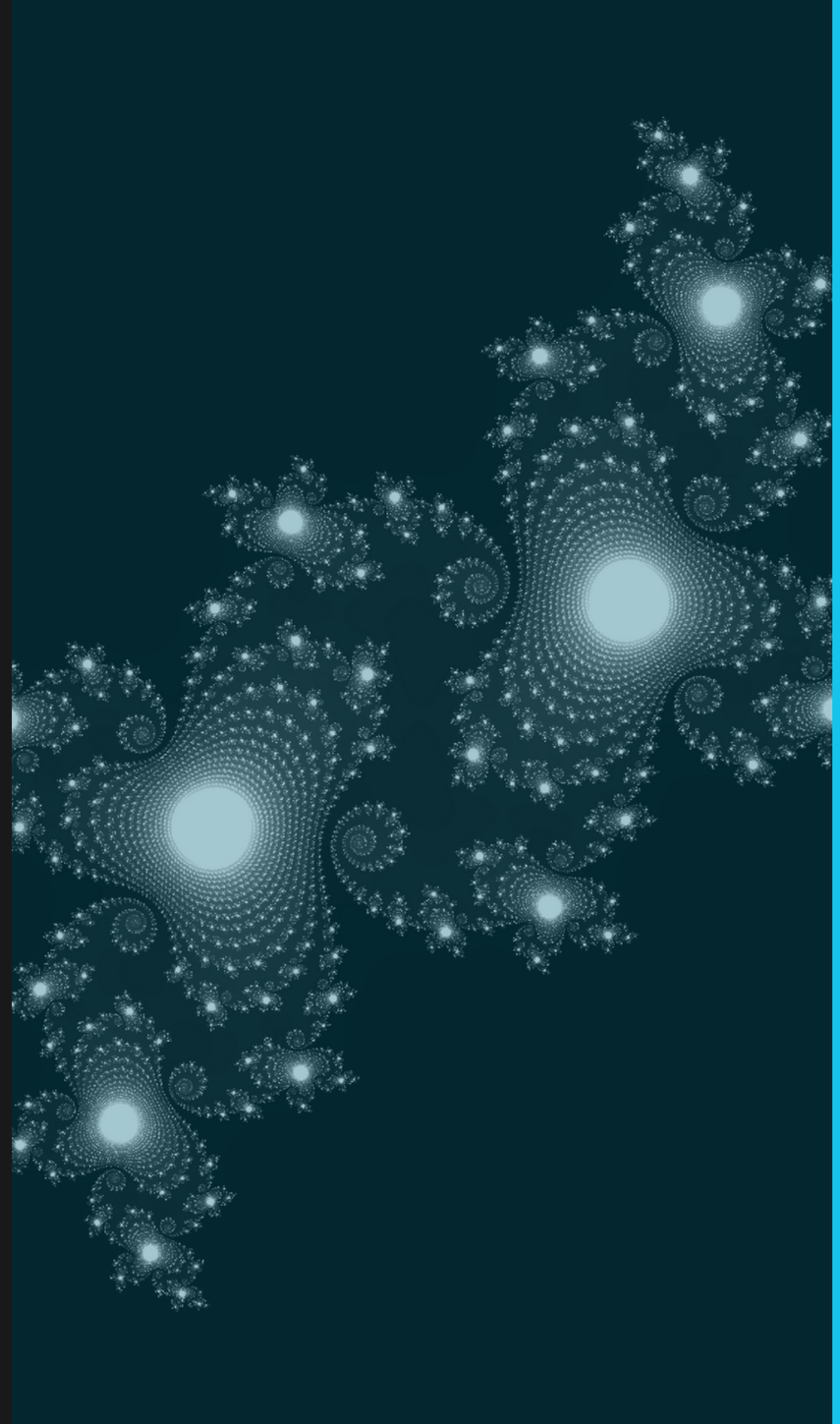
Deeper understanding of how fractal geometry works and the self-similarity principle of fractals



THE POWER OF VISUALIZATIONS

Conclusion

Thank you! Any questions?



GROUP 1