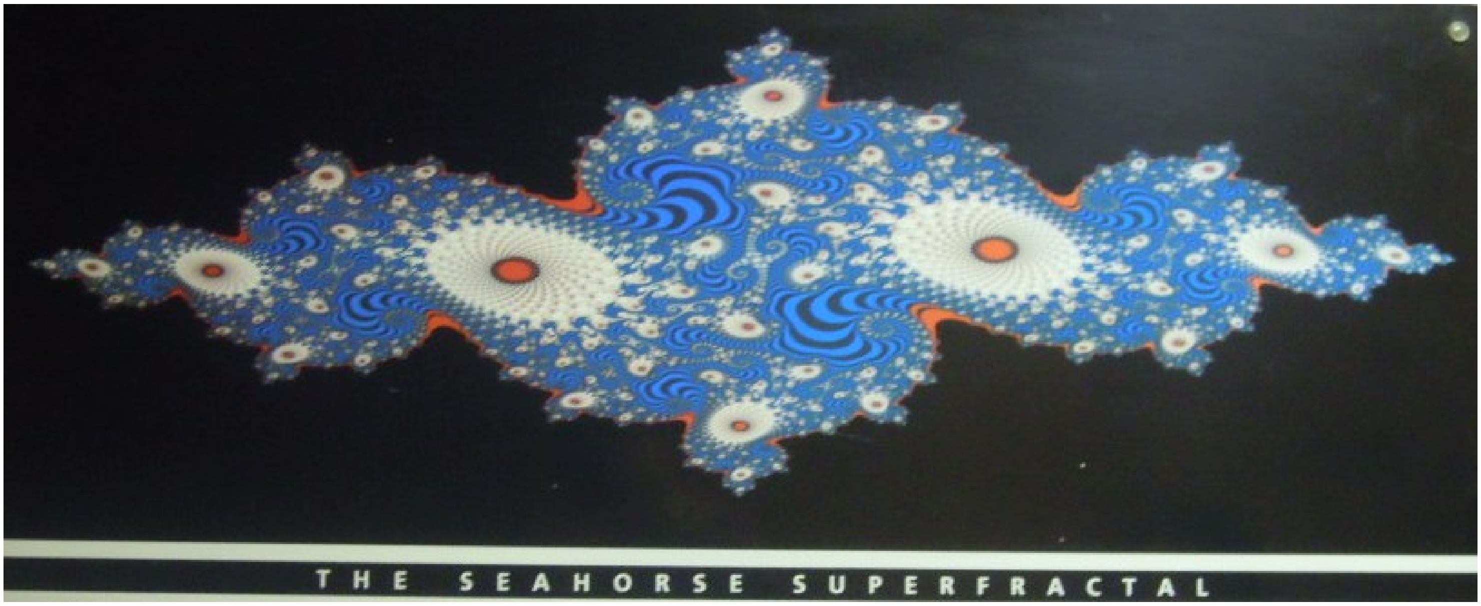
CP400N TERM PROJECT
MPI programming for JULIA SETS

Instructor: I. Kotsireas, e-mail `ikotsire@wlu.ca`



- This Term Project must be completed on a SHARCnet cluster using MPI C programming.
- Strict adherence to the SHARCNET etiquette (as per the course outline) is required.

The purpose of this project is to produce high-resolution pictures of Julia sets using C/MPI programming to compute the corresponding points and OpenGL to render the pictures. Julia sets are fractal sets that are defined via the iteration of functions, such as: $Q_c(z) = z^2 + c$ and $T_c(z) = z^3 + c$, where z, c are complex numbers.

- Familiarize yourself with the background/definition/algorithms for Julia sets, as described on pages 75-112 (chapters 5,6,7) of the book

  ```
  Chaos, fractals, and dynamics.
  Computer experiments in mathematics.
  by Robert L. Devaney. Addison-Wesley, 1990.
  ```

- Write C/MPI routines to compute the points for a Julia set associated with a specific function, on a large square grid. The points should be stored in data files that are readable by OpenGL.
- Pay special attention to the load balancing issues that are associated with the computation of Julia Sets.
- Write an OpenGL routine that renders the points (read from a data file) in high resolution .jpg files. You will have to produce several images, as there is a very large variety of Julia sets possible.

Required Deliverables:

- Term Project Report: code (well-commented), description of problems/issues encountered
- database of Julia set images, in the form of a poster.
- project demonstration: code will be tested with additional examples
- solutions to selected exercises/projects from chapters 6,7 of the aforementioned book.

Sample algorithm to compute a Julia set for $Q_c(z) = z^2 + c$

Notation: the orbit $Orb(z_0)$ of a point z_0 under the function $Q_c(z)$ is the sequence of points:

$$z_0, Q_c(z_0), \quad Q_c(Q_c(z_0)), \quad Q_c(Q_c(Q_c(z_0))), \quad Q_c^4(z_0), \quad Q_c^5(z_0), \ldots$$

i.e. the list of successive iterates of the point z_0 under the function $Q_c(z)$.

Algorithm:

- Write $c = c_1 + ic_2$, where c_1, c_2 are real numbers, with the property that $|c| \leq 2$, i.e. $c_1^2 + c_2^2 \leq 4$.
- Select a 1000×1000 grid in the plane (one million points).
- For each point z_0 on this grid:
  - transform screen coordinates to xy coordinates.
  - compute the first 50 points on the orbit $Orb(z_0)$.
- If any point on $Orb(z_0)$ lies outside the circle of radius 2, then stop computing points on this orbit and color-code the point z_0 white.
- If all 50 points on $Orb(z_0)$ lie inside the circle of radius 2, then color-code the point z_0 black.

Test/benchmark cases for debugging: (page 91 of the book)

- $c = 0$, should produce a picture of a disk.
- $c = -1$, $c = 0.3 - 0.4i$, $c = 0.360284 + 0.100376i$, $c = -0.1 + 0.8i$, see next page and page 91 of the book.

Color images

In order to better capture the dynamics of (filled) Julia sets, you can assign different colors to different **escape times**, depending upon the time of the escape. More specifically:

1. points that escape between iterations 1 and 5 $\rightsquigarrow$ color 1.
2. points that escape between iteration 6 and 10 $\rightsquigarrow$ color 2.
3. etc.

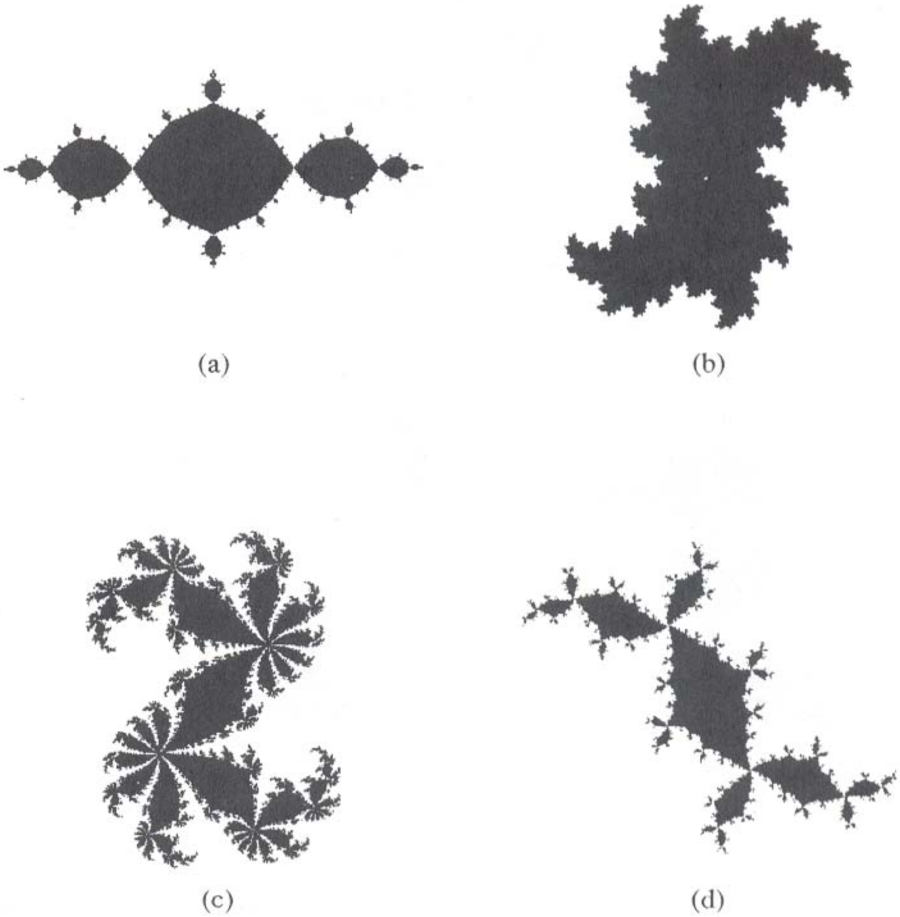


Figure 6.2 Julia sets for Q_c (a) $c = -1$, (b) $c = .3 - .4i$, (c) $c = .360284 + .100376i$, (d) $c = -.1 + .8i$.