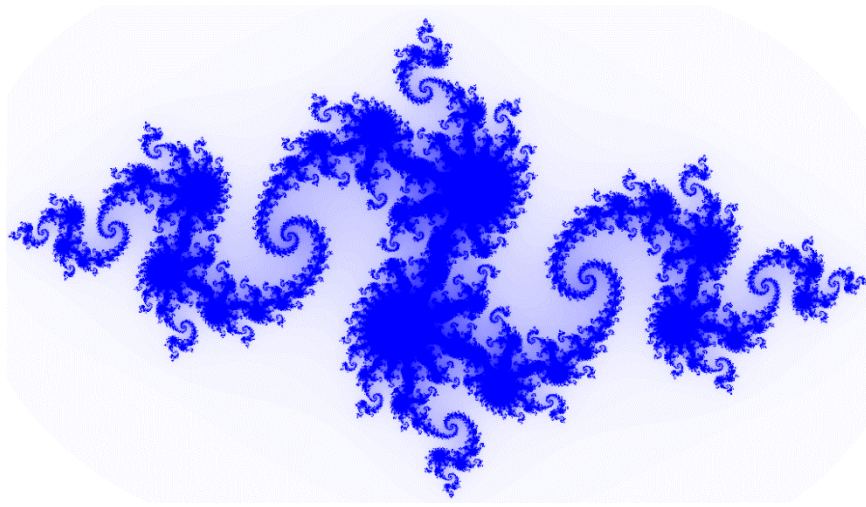


GPU Programming

TP1 – Julia sets



Install GLUT and GLEW.

Compile and run the provided base code. Change some features (colors, controls, etc.) to get familiar with it.

If the program performs only at 60 frames/s, you can deactivate vertical sync, a display option found in the Nvidia Control Panel which synchronizes the frame rate of the program with the monitor refresh rate (60Hz).

Julia sets are beautiful escape-time fractals. Browse the internet and make yourself knowledgeable about the Julia set algorithm. This is the pseudo-code for the Julia color of a single complex point (x,y) , depending on the constant seed (sx, sy) and the computing precision p :

```
function juliaColor (x, y, sx, sy: float, p: int): color in 0..1 gray scale
{
    a = complex (x,y)
    seed = complex (sx, sy)
    for (i=0; i<p; i++) {
        a = a * a + seed;
        if squaredMagnitude(a) > 4 then return 1-i/(float)p
    }
    return 0
}
```

Exercices (1 point per question)

- 1) Write a CPU version of the Julia set fractals. The user can change the seed by dragging the mouse, and increment/decrement the precision by hitting special keys.
- 2) Write an equivalent GPU version of the Julia set fractals. Use 2D thread and 2D block indexing. The user can toggle between the CPU mode and the GPU mode.
- 3) GPU version 2: use pinned host memory for faster data transfer.