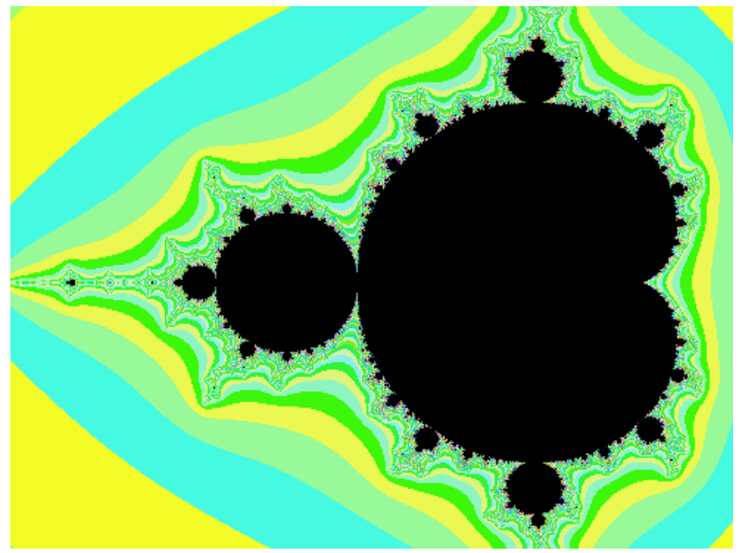
Lab 1: Load Balancing

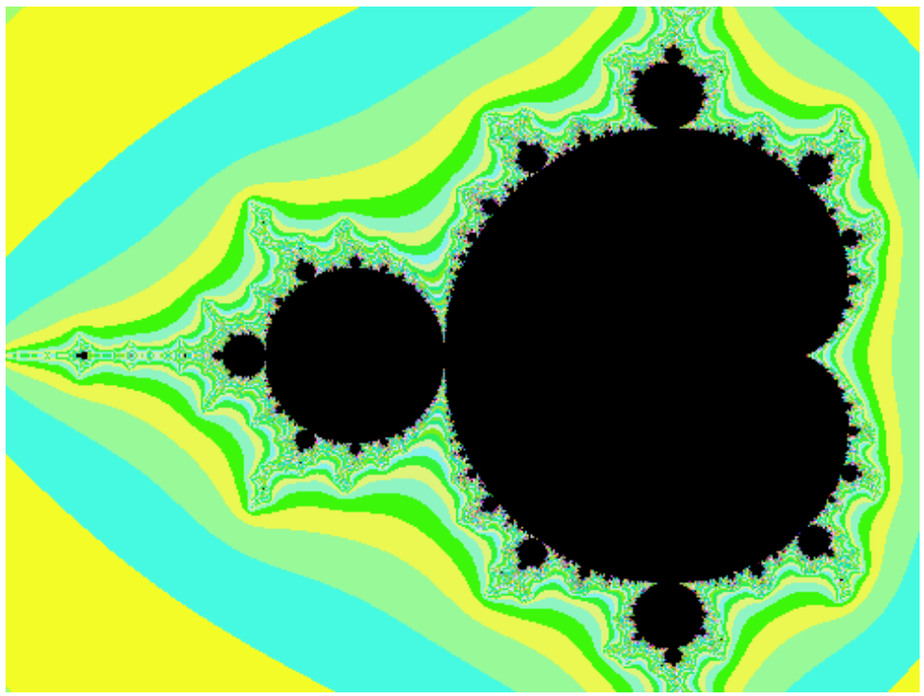


- Working with threads (Pthreads) on multicore CPU
- Mandelbrot fractal image generation
- Test if a complex number is in the Mandelbrot set
- For those interested in the maths, check out:
  - https://en.wikipedia.org/wiki/Mandelbrot_set
  - https://www.youtube.com/watch?v=NGMRB4O922I

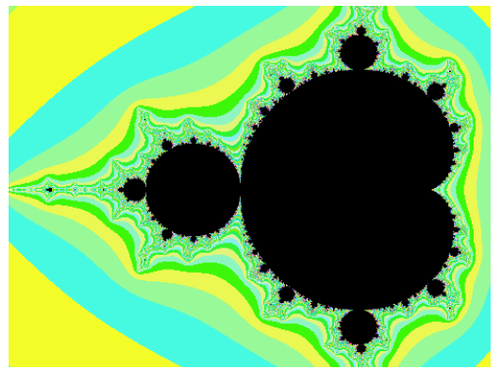
Mandelbrot Algorithm

```
int is_in_Mandelbrot(float Cre, float Cim)
{
   int iter;
   float x=0.0, y=0.0, xto2=0.0, yto2=0.0, dist2;

   for (iter = 0; iter <= MAXITER; iter++)
   {
      y = x * y;
      y = y + y + Cim;
      x = xto2 - yto2 + Cre;
      xto2 = x * x;
      yto2 = y * y;
      dist2 = xto2 + yto2;
      if ((int)dist2 >= MAXDIV)
         break; // diverges
  }
   return iter;
}
```

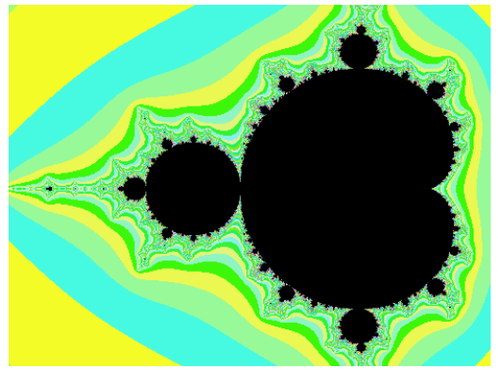


Load Balancing



- Each image pixel is an **independent** unit of work
  - => embarrassingly parallel!
- However, all pixels are not **equal amount** of work!
  - Load balancing becomes a problem.

Lab 1



- Goal for the lab:
  - Implement a solution with near-equal load
  - Try different approaches
  - Utilize properties of the domain
    - How well will your solution work in a general case?