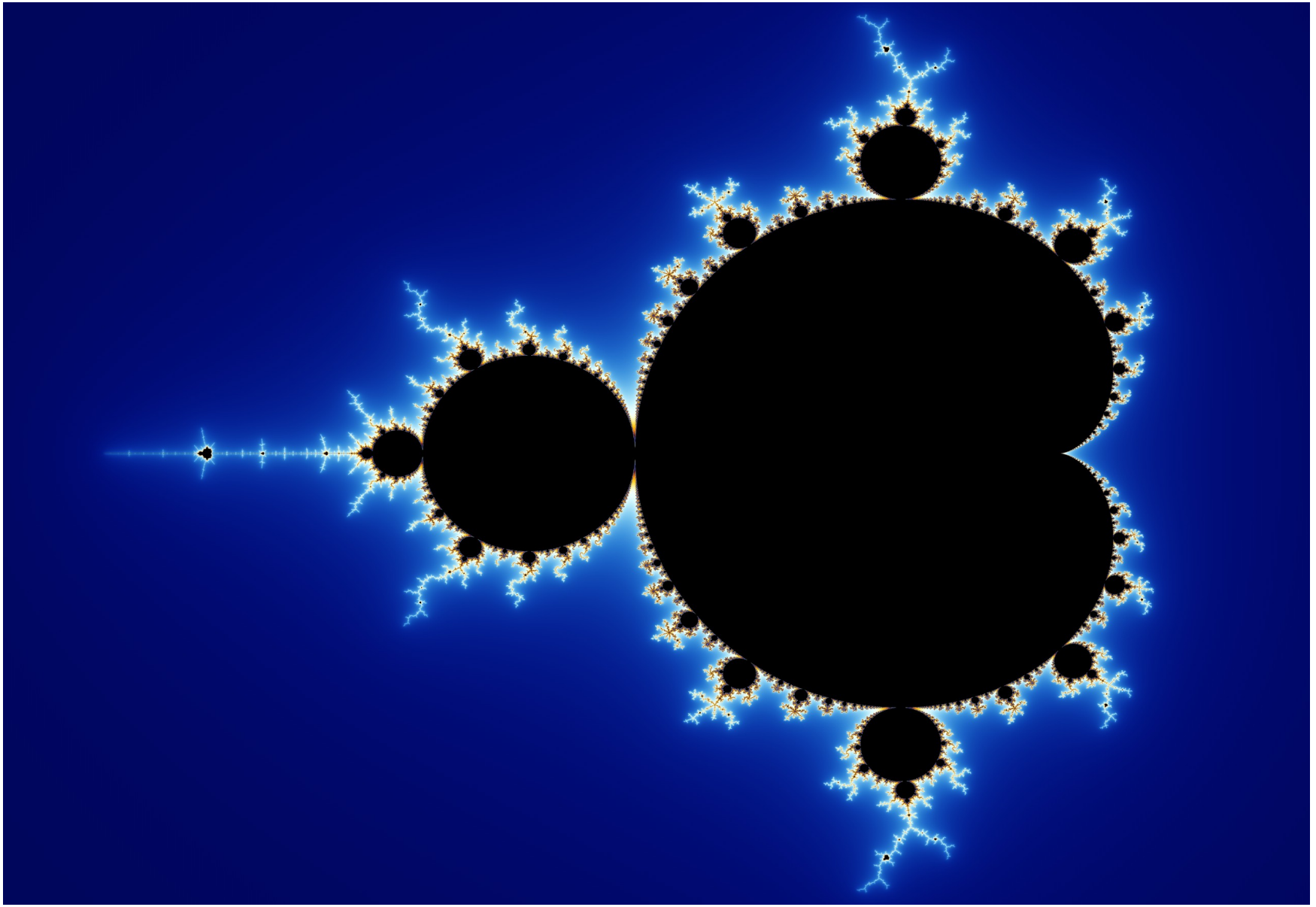
SINF1252

Présentation P2
Fractales

Fractales

- Même modèle à toutes les échelles
- Objet mathématique
- Phénomène naturel (rivières, arbres, …)
- Différents types

2D fractal



3D fractal

Ensemble de Julia

$$z_{n+1} = z_n^2 + c$$

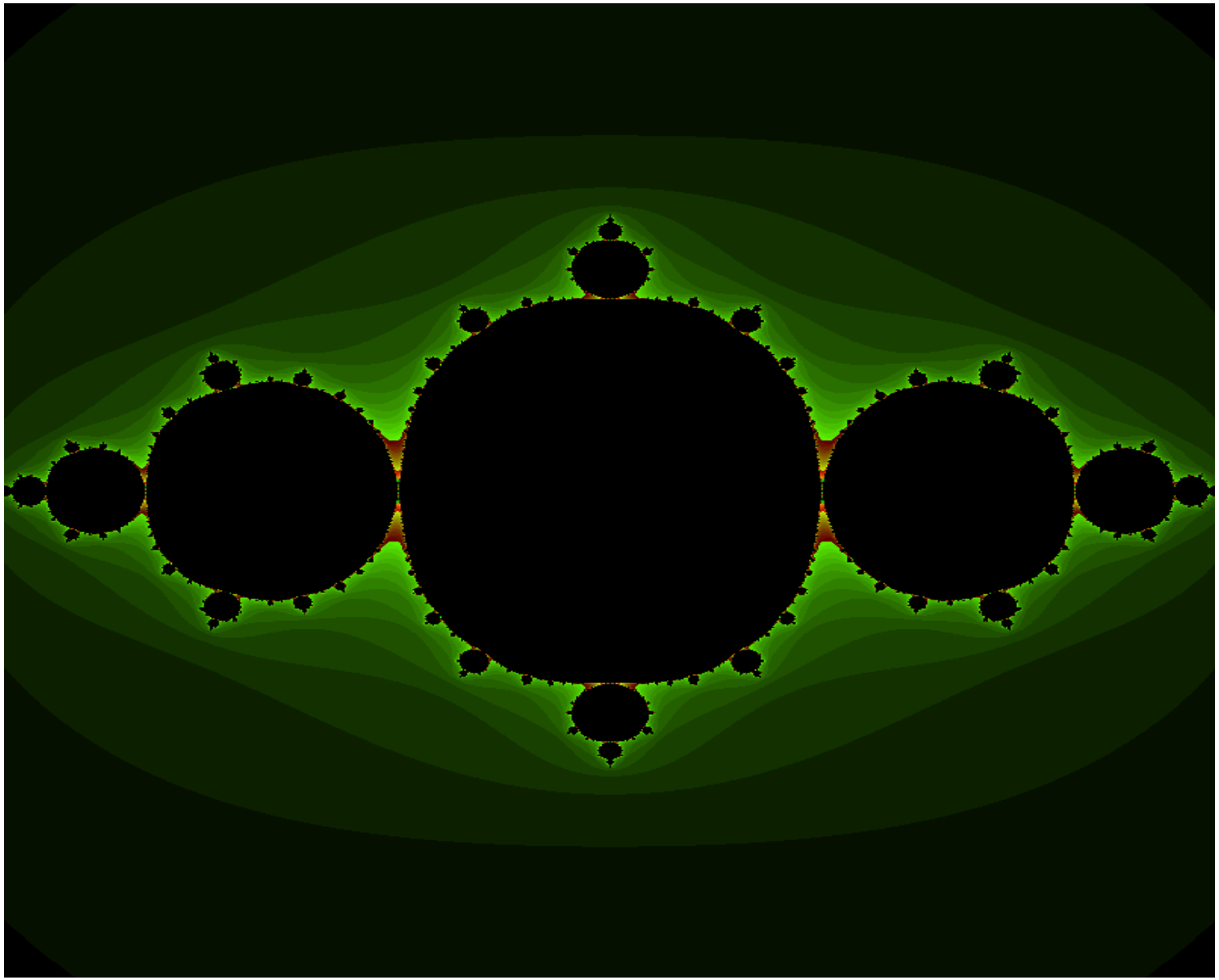
$$z\ ,\ c \in \mathbb{C}$$

- Relation de récurrence
- Défini sur un plan complexe [-1;1] x [-1;1]
- Pour chaque c = a + bi, il existe une fractale

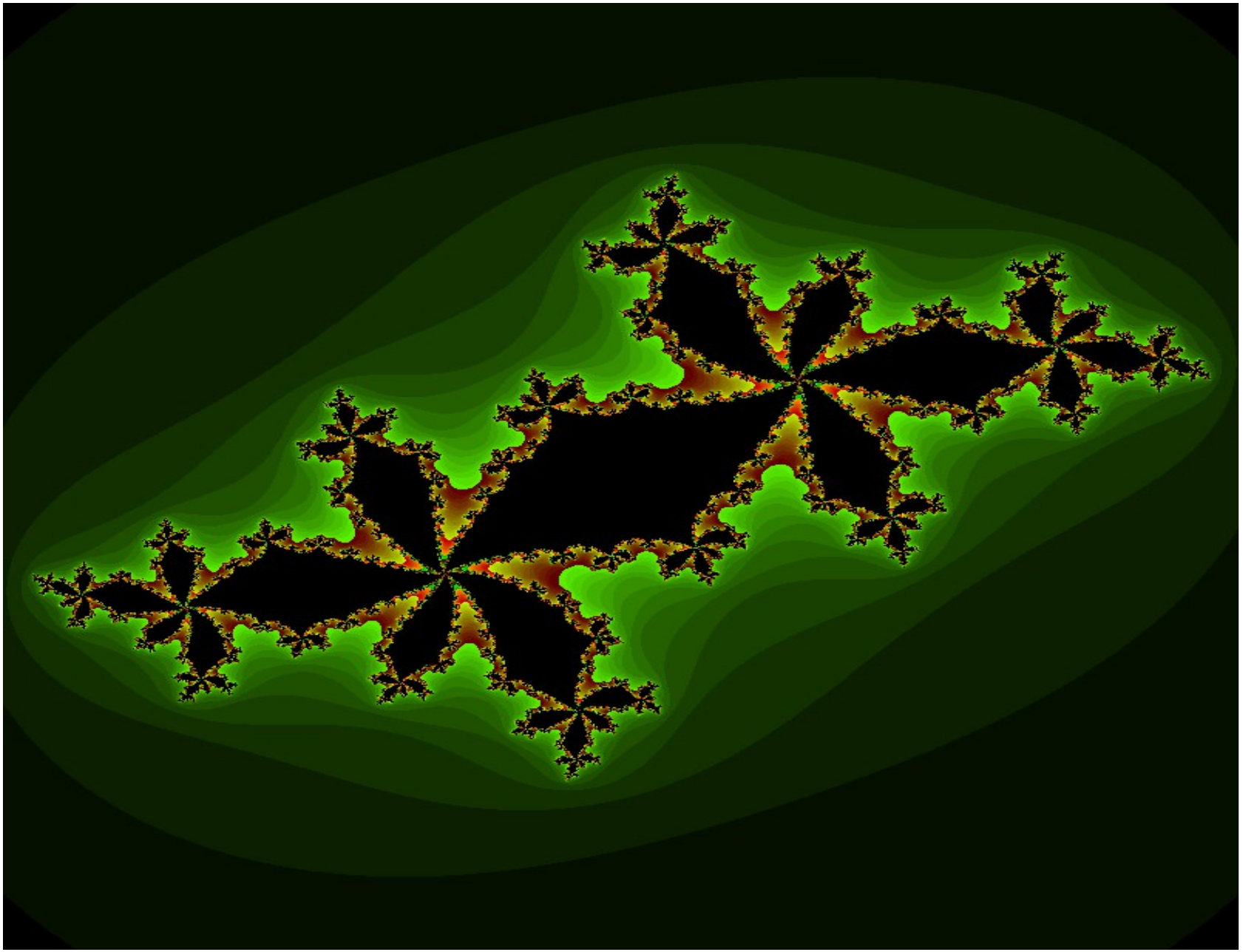
Image

- Représenter une fractale dans une image de NxM pixels
- Associer chaque pixel (x,y) à un point € [-1;1]x[-1;1] (nombre z)
- Calculer la relation de récurrence pour chaque z, pour c = a + bi donné
- Valeur du pixel = nombre d'itérations
  - jusqu'à une condition d'arrêt
- Transformation #itérations => couleur

Exemple (1) c = -0.8 + 0.0i



Exemple (2) c = -0.52 + 0.57i



Projet

- Input: un ou plusieurs fichiers, ou stdin
- 1 description de fractale/ligne (nom, coordonnées a + bi, taille de l'image à générer)
- 10, 100, 1000, … fractales
- Output: fichier bitmap (BMP) représentant la fractale dont la moyenne arithmétique des valeurs des pixels est la plus haute

$$max\left(\frac{1}{MN}\sum_{x=0}^{M-1}\sum_{y=0}^{N-1}V(x,y)\right)$$

Code: calcul d'une fractale

- ```
  int w, h, x, y;
  w = fractal_get_width(f);
  h = fractal_get_height(f);
  for (x = 0; x < w; x++) {
      for (y = 0; y < h; y++) {
          fractal_compute_value(f, x, y);
      }
  }
  ```
  ```

Architecture

- Calcul d'une fractale: coûteux, prop. taille image
- Parallélisable
- Multi-threading (librairie pthreads)
- Phase d'architecture, PAS de code

Code

- Template, API
- fractal_compute_value donné
  - Applique rel de récurrence sur un pixel donné
- write_bitmap_sdl donné
  - Transforme une fractale en fichier BMP
- Unit tests !!!

Étapes

- S8: Interviews d'architecture
- S11: permanence
- S12: remise du projet