Newtonov fraktal (3. laboratorijska vaja)

Besedilo naloge

Uporabite program iz druge laboratorijske vaje za rešitev enačbe $f(z) = z^3 - 1 = 0$. (V tem primeru je enačba za Newtonovo metodo $z_{n+1} = z_n - (z_n^3 - 1) / 3z_n^2$).

Tri rešitve enačbe $f(z) = z^3 - 1 = 0$ so $r_1 = 1$, $r_2 = -1/2 + \sqrt{3}/2i$ in $r_3 = -1/2 - \sqrt{3}/2i$.

Pri tem je zanimivo ugotoviti, katere začetne točke z_0 v kompleksni ravnini konvergirajo proti kateri od treh rešitev. Napišite program, ki to preizkusi za vse točke z_0 s celoštevilskim realnim in imaginarnim delom med –500 in 499. Za vsako od točk preverite, proti kateri od rešitev konvergira (uporabite isto natančnost kot v drugi laboratorijski vaji, tj. 10^{-5}), in ugotovitev vpišite na ustrezno mesto v dvorazsežnostno tabelo velikosti 1000 × 1000. Če postopek konvergira proti prvi rešitvi, v tabelo vpišite vrednost 80, če konvergira proti drugi rešitvi, v tabelo vpišite vrednost 160, v primeru tretje rešitve pa vpišite 240. Če postopek ne konvergira proti nobeni rešitvi (npr. po 50 iteracijah), na ustrezno mesto v tabeli vpišite ničlo. Dobljeno tabelo potem pretvorite v bitno sliko.

Opomba: Z vpisovanjem treh različnih vrednosti boste dobili sliko fraktala iz treh barv. Če vrednostim dodate informacijo o številu iteracij, ki so potrebne, da dosežete določeno rešitev, boste dobili veliko bolj dinamično sliko.

Uporaba kompleksnih števil

C pozna standardno knjižnico `<complex.h>`, ki je relativno enostavna za uporabo. Naslednji primer je dovolj zgovoren:

```
#include <stdio.h>
#include <complex.h>

int main() {
  float complex z;
  float complex z1 = 1.0 + 3.0 * I;
  float complex z2 = 1.0 - 4.0 * I;
  z = z1 * z2;
  printf("Z = %.2f %+.2fi\n", creal(z), cimag(z));
  z = z1 + z2;
  printf("Z = %.2f %+.2fi\n", creal(z), cimag(z));
  z = z1 - z2;
  printf("Z = %.2f %+.2fi\n", creal(z), cimag(z));
  z = z1 / z2;
  printf("Z = %.2f %+.2fi\n", creal(z), cimag(z));
  printf("abs(Z) = %.2f\n", cabs(z));
  return 0;
}
```

Pretvorba tabele v bitno sliko

Za pomoč imate na voljo funkcijo `shraniBMP`, ki iz tabele celih števil ustvari bitno sliko. Funkcija kot prvi parameter sprejme dvorazsežnostno tabelo tipa `unsigned char`

(vrednosti med vključno 0 in 255), kjer vsaka od vrednosti predstavlja točko določene barve. Drugi in tretji parameter funkcije predstavljata širino in višino podane tabele, četrti parameter pa ime datoteke, kamor želimo, da se slika shrani. Na primer:

```
#define DIM 1000
unsigned char slika[DIM][DIM];
//...
shraniBMP(slika, DIM, DIM, "slikca.bmp");
```