PRÀCTICA 1: MEMÒRIA

Fractals pel Mètode de Newton

Ona Sánchez Núñez: 1601181

1 Problema plantejat

Escriure un programa que respongui a la crida *./dibfr narr xmn xmx nx ymn ymx ny tolcnv maxit* , dibuixant el fractal corresponent.

La motivació del treball és la necessitat d'una empresa de disseny de rutines que generin fractals pel mètode de Newton. L'empresa està interessada en rutines de càlcul i requereix una utilitat que permeti representar fractals des de la línia de comandes Unix, usant el programa gnuplot.

2 Mètode de Newton

2.1 Introducció

El mètode de Newton permet aproximar arrels d'equacions de la forma f(x) = 0. Suposem que f:[a,b] $\to$ R és de classe C^2 i que existeix (a,b) $\ni \alpha$ amb f(α)=0.

Suposem que x és una aproximació de α, i expandim Taylor fins a ordre 1 al voltant de x:

$$0 = \mathrm{f}(\alpha) = \mathrm{f(x)+f'(x)}(\alpha\text{-x})+O((\alpha\text{-x})^2)$$
$$\approx \mathrm{f(x)+f'(x)}(\alpha\text{-x}),$$

d'on

$$\alpha\text{-x} \approx -\frac{f(x)}{f'(x)}$$

Així, x-f(x)/f'(x) $\approx$ x+(α-x) = α i esperem que x-f(x)/f'(x) sigui millor aproximació de α que x.

2.2 Algorisme

Escollim x_0 aproximació inicial de α, triem ϵ tolerància, n_{max} nombre màxim d'iterats que permetrem i fem:

$$\forall \mathrm{i} = 0,1,...,n_{max} - 1,$$
$$x_{i+1} = x_i - \frac{(f(x_i)}{f'(x_i)}$$
si $|x_{i+1} - x_i| \leqslant \epsilon$, tornem x_{i+1}, STOP error (no convergència)

3 Estratègia per programar

- Evitar repetir codi.
- Noms de variables que permetin una lectura entenedora.
- Comentar el codi.
- Validar cada funció un cop realitzada.

4 Manual del software

4.1 Per a les biblioteques

4.1.1 avalp()

Descripció: Rutina que avalua el polinomi p(z) = (z-w_0)(z-w_1)...(z-w_{n-1}), on z = x + iy i w_j = u_j + iv_j. És una funció de tipus void que torna dins *px la part real de p(z) i dins *py la part imaginària.

Arguments que rep:

· x : double, part real del complex z.

· y : double, part imaginària del complex z.

· *px : punter de tipus double, guarda la part real de p(z).

· *py : punter de tipus double, guarda la part imaginària de p(z).

· n : double, llargada dels vectors u i v.

· u[] : vector de tipus double, conté u_j desde j=0 fins j=n-1 (part real de w).

· v[] : vector de tipus double, conté v_j desde j=0 fins j=n-1 (part imaginària de w).

4.1.2 avaldp()

Descripció: Rutina que avalua la derivada de p(z),

$$p'(z) = \sum_{j=0}^{n-1} \prod_{\substack{k=0 \\ k \neq j}}^{n-1} (z-w_k)$$

És una funció de tipus void que torna dins *dpx la part real de p'(z) i dins *dpy la part imaginària.

Arguments que rep:

· x : double, part real del complex z.

· y : double, part imaginària del complex z.

· *dpx : punter de tipus double, guarda la part real de p'(z).

· *dpy : punter de tipus double, guarda la part imaginària de p'(z).

· n : double, llargada dels vectors u i v.

· u[] : vector de tipus double, conté u_j desde j=0 fins j=n-1 (part real de w).

· v[] : vector de tipus double, conté v_j desde j=0 fins j=n-1 (part imaginària de w).

4.1.3 cnvnwt()

Descripció: Rutina que aplica el mètode de Newton a un complex z_0 = x + iy a partir de z_0 fins a un màxim d'iterats. Si un dels iterats s'apropa a l'arrel w_j a distància més petita que la tolerància, la rutina retorna j. Si no, retorna -1. Aquesta rutina utilitza les funcions anteriors avalp() i avaldp() anterios, i s'utilitza per decidir de quin color cal pintar cada part del fractal generat al gnuplot.

Arguments que rep:

- x : double, part real del complex z.
- y : double, part imaginària del complex z.
- tolcnv : double, tolerància del mètode de Newton.
- maxit : double, màxim nombre d'iteracions.
- n : double, llargada dels vectors u i v.
- u[] : vector de tipus double, conté u_j desde j=0 fins j=n-1 (part real de w).
- v[] : vector de tipus double, conté v_j desde j=0 fins j=n-1 (part imaginària de w).

4.2 Per a les utilitats

4.2.1 dibfr

Descripció: Utilitat que conté el mètode main. Comprova que es rebin tots els arguments necessaris, en cas que no, imprimeix un missatge d'error i retorna -1, en cas contrari, se li han d'introduir els valors u_i v_i r_i g_i b_i,

4.2.2 Com generar un fractal

1. Tenir completes les funcions dels fitxers dibfr.c, nwtfr.c i nwtfr.h
2. Obrir el terminal i situar-se al directori on es tenen guardats els fitxers anteriors.
3. Compilar el programa amb la comanda : gcc dibfr.c nwtfr.c -o dibfr
4. Executar el programa amb la comanda : dibfr narr xmn xmx nx ymn ymx ny tolcnv maxit > fractal.txt
5. Escriure a la terminal per files : u_i v_i r_i g_i b_i, on u = part real arrel, v = part imaginària, v = 1 0 0, g = 0 1 0 i b = 0 0 1. Exemple a Figura 1.
6. Obrir gnuplot i situar-se al directori on es té guardat el fitxer fractal.txt
7. Executar la comanda : unset key
8. Executar la comanda : plot 'fractal.txt' w rgbimage. Exemple a Figura 2.

Apunt tècnic: narr = nombre d'arrels, xmn, xmx, ymn i ymx defineixen el rectangle on es mostra el fractal, nx = nombre de punts horitzontals, ny = nombre de punts verticals, tolcnv = tolerància i maxit = màxim nombre d'iteracions.

```
C:\Users\onasa\Documents\1r 2n semestre\Calc numèric>dibfr 3 -2 2 640 -2 2 480 1e-3 50 > fractal.txt
1 0 1 0 0
-0.5 .8660254037844386 0 1 0
-0.5 -.8660254037844386 0 0 1
```

Figura 1

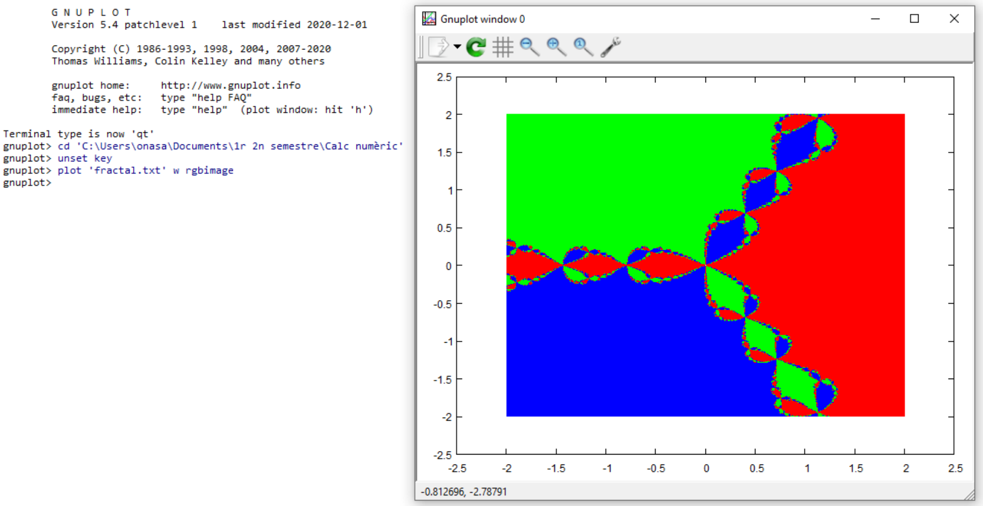


Figura 2

5 Descripció de les proves

5.1 Validació

Per tal de validar les rutines:

- avalp()
- avaldp()
- cnvnwt()

S'ha creat un fitxer test avalp amb un mètode main que les crida una per una i mostra per pantalla els resultats.

En el cas de avalp(), es mostra la part real i la imaginària de p(z).
En el cas de avaldp(), es mostra la part real i la imaginària de p'(z).
En el cas de cnvnwt(), es mostra j.

Els resultats s'han validat contra la taula 1 del PDF *pr01*.

La rutina:

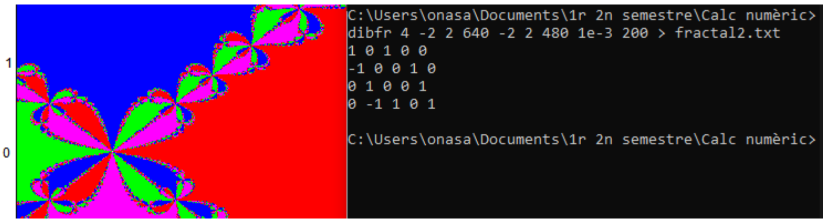
· dibfr

S'ha validat generant el fractal de la Figura 2 i comparant-lo amb el fractal del PDF *pr01*.

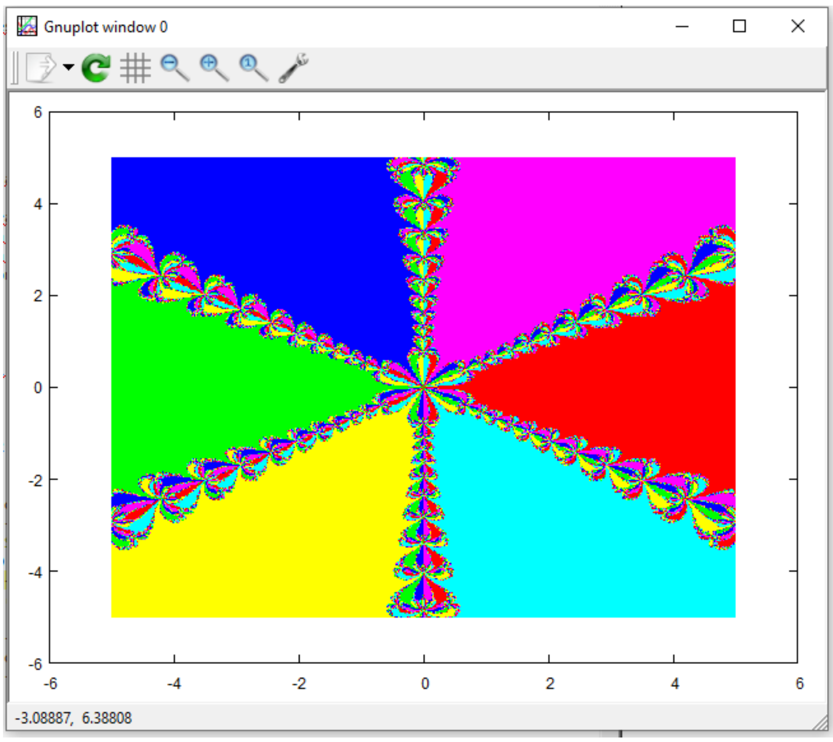
5.2 Experiments

El fractal generat a la Figura 2 correspon al polinomi z^3 - 1.
Altres fractals:

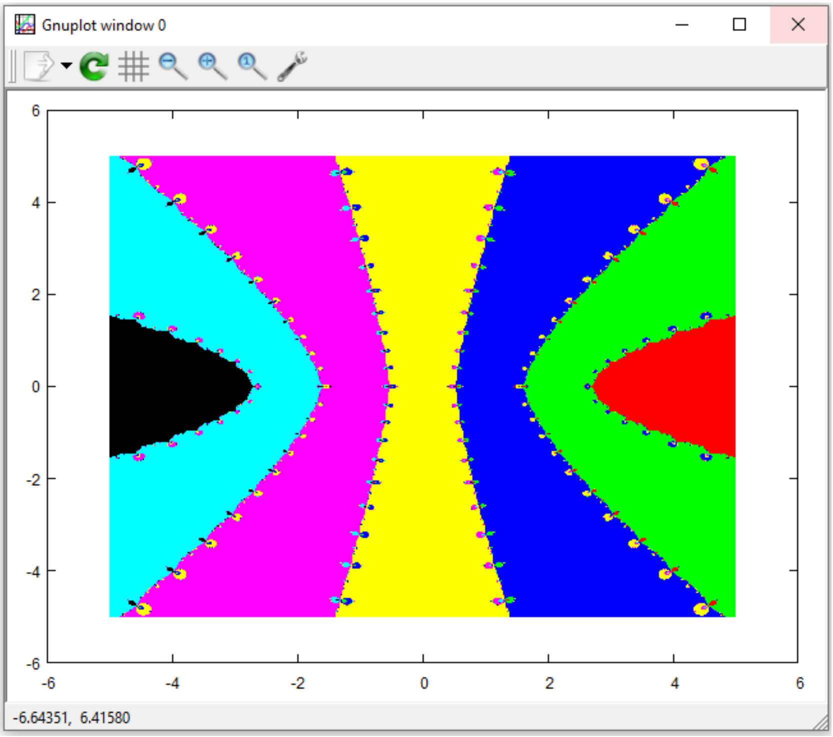
1. Prova amb el polinomi de grau 4 z^4 - 1. Les arrels són: 1, -1, i, -i.



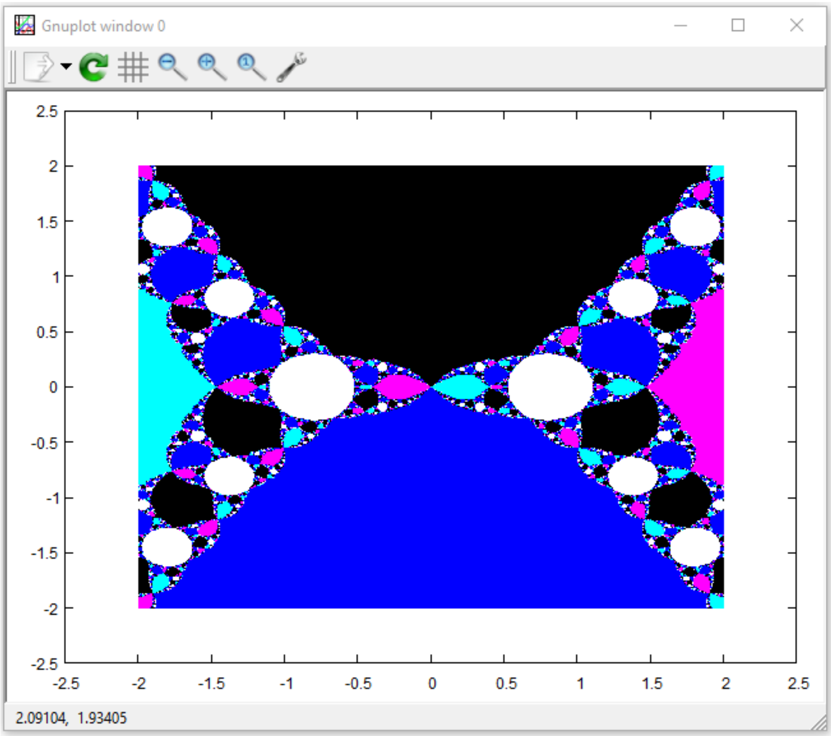
2. Prova amb el polinomi z^6 -1i. Les arrels són: 1, -1, -0.5-0.866i, -0.5+0.866i, 0.5-0.866i i 0.5+0.866i.



3. Prova amb el polinomi (z-3)(z-2)(z-1)z(z+1)(z+2)(z+3). Les arrels són: 3, 2, 1, 0, -1, -2, -3.



4. Prova amb el polinomi d'arrels i, -i, 2.3 i -2.3:



6 Bibliografia

pr01.pdf: Pràctica 1, Fractals generats pel mètode de Newton.

apunts-v001.pdf: Apunts de Càlcul Numèric.

https://es.symbolab.com/solver/roots-calculator: Calculadora d'arrels.

https://www.chiark.greenend.org.uk/ sgtatham/newton/: Fractals derived from Newton-Raphson iteration.