

Pràctica 1: Memòria

Biel González Garriga

30 d'abril de 2021

1 Motivació

Una empresa de disseny ens demana codificar unes rutines per generar fractals del mètode de Newton i una utilitat per representar fractals mitjançant la línia d'ordres Unix amb gnuplot i comparar diferents execucions.

1.1 Objectius

- Desenvolupar la biblioteca amb `avalp`, `avaldp`, `cnvnwt` i el codi client que permeti executar el codi per generar fractals.
- Desenvolupar la utilitat per representar fractals amb gnuplot.

2 Implementació

2.1 Biblioteca

Per a la implementació d'`avalp` i `avaldp` s'ha realitzat un desenvolupament de les fórmules amb l'objectiu de separar la part real de la imaginària dels nombres per així poder fer-los servir en variables diferents i facilitar el treball amb aquests nombres sense la necessitat de programar una aritmètica de nombres complexos. Un cop separades les dues parts s'han realitzat els bucles pertinents a cada funció de la biblioteca per dur a terme els sumatoris i/o productes que siguin necessaris a cada funció.

Finalment, a la funció `cnvnwt` s'ha fet també un desenvolupament de la fórmula del mètode de Newton, per separar part real i complexa, i un càlcul de distàncies per veure si el resultat trobat s'acosta a qualsevol de les arrels donades. Si no és així, torna a iterar fins que trobi que el valor s'acosta suficientment a una de les arrels o es realitza el màxim nombre d'iteracions permeses.

2.2 Codi client

El codi client, situat a `dibfr.c`, és el que s'encarrega de rebre els arguments. A partir dels arguments el codi generarà:

- L'espai on es trobarà el fractal.
- Tots els punts del "canvas" que es pasaran posteriorment a `cnvnwt`.

Després d'això el codi usa `cnwtn` per assignar la conca d'atracció a la qual pertany cada punt i passarà a un arxiu cada un dels punts amb el color corresponent de la conca d'atracció a la que pertany.

3 Manual d'ús

3.1 Biblioteca

`avalp (double x, double y, double *px, double *py, int n, double u[], double v[])`

- `x`: part real del nombre.
- `y`: part complexa del nombre.
- `n`: nombre d'arrels.
- `u[]`: parts reals de les arrels del polinomi.
- `v[]`: parts complexes de les arrels del polinomi.
- Valors generats:
 - `px`: parts reals del polinomi avaluat.
 - `py`: parts complexes del polinomi avaluat.

```

    avaldp (double x, double y, double *dpx, double *dpy, int n, double
u[], double v[])

```

- x: part real del nombre.
- y: part complexa del nombre.
- n: nombre d'arrels.
- u[]: parts reals de les arrels del polinomi.
- v[]: parts complexes de les arrels del polinomi.
- Valors generats:
 - dpx: parts reals de la derivada avaluada.
 - dpy: parts complexes de la derivada avaluada.

```

int cnvnwt (double x, double y, double tolcnv, int maxit, int n,
double u[], double v[])

```

- x: part real del nombre.
- y: part complexa del nombre.
- tolcnv: distància la qual si és més gran que la que hi ha entre el punt i l'arrel s'entén que el punt forma part de la conca d'atracció de l'arrel.
- maxit: nombre màxim d'iterats.
- n: nombre d'arrels.
- u[]: parts reals de les arrels del polinomi.
- v[]: parts complexes de les arrels del polinomi.
- Valors retornats:
 - j: índex de l'arrel de la qual forma part la conca d'atracció.
 - -1: en cas que en maxit iteracions no es trobés una conca d'atracció.

3.2 Codi client

És necessari estar situat al directori on esitigui el arxiu dibfr.c.

```
make dibfr
```

```
./dibfr narr xmn xmx nx ymn ymx ny tolcnv maxit > fractal.txt
```

- narr: nombre d'arrels del polinomi p.
- xmn, xmx, ymn, ymx: definició del rectangle on es situarà la fractal.
- nx, ny: nombre de punts que generarà a cada eix.
- tolcnv, maxit: arguments per cnvntw().
- fractal.txt: arxiu on s'emagatzemaran els resultats de la fractal.

Un cop iniciat el programa, s'han de tornar a donar uns altres arguments, aquests son les arrels del polinomi i el color R/G/B assignat a l'arrel (1=color, 0=no color). De la següent manera:

```
partReal partComplexa R G B
```

3.3 Gnuplot

Les comandes de gnuplot s'han d'executar després d'executar la comanda `make dibfr` i hi ha dues opcions:

- Visualització normal:

```
gnuplot  
unset key  
plot 'fractal.txt' w rgbimage
```
- Visualització interactiva:

```
gnuplot  
load 'dibprimer.gnu'
```

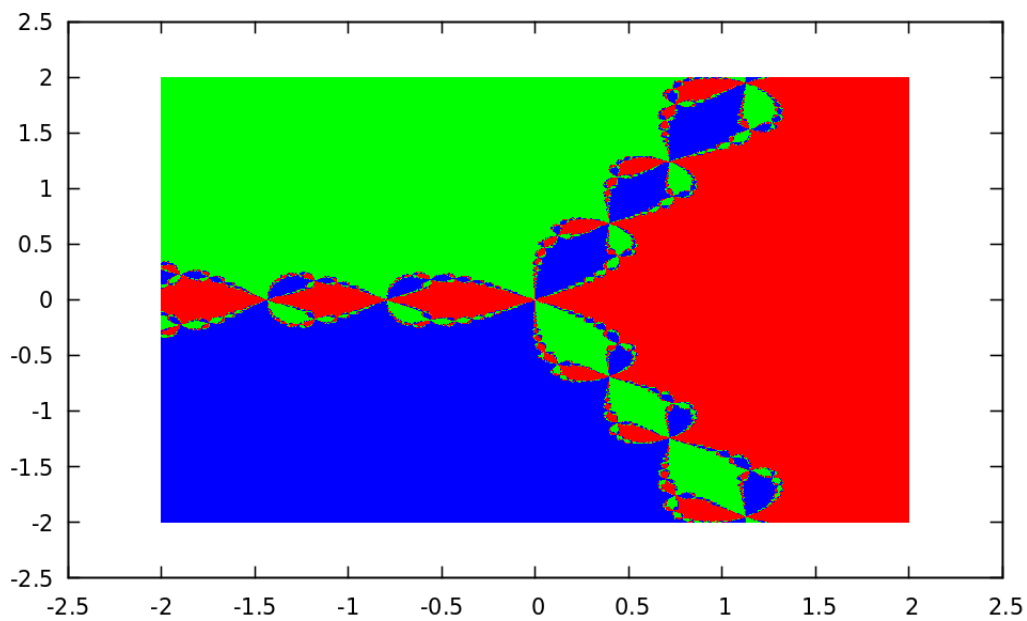
En aquest cas s'han d'omplir els arxius "dibprimer.gnu" i "recalczoom.gnu" amb les dades que et demana, ja venen explicades a l'arxiu.

Aquesta visualització permet que al fer zoom des de gnuplot si prems la tecla "z" recalculi la fractal, per així veure bé la fractal per molt que t'acostis.

3.4 Exemples d'ús

```
make dibfr
./dibfr 3=n -2=xmn 2=xmx 640=nx -2=ymn 2=ymx 480=ny 1e-3=tolcnv 50=maxit
> fractal.txt
1 0 1 0 0
-0.5 .8660254037844386 0 1 0
-0.5 -.8660254037844386 0 0 1
gnuplot
unset key
plot 'fractal.txt' w rgbimage
```

El resultat es:

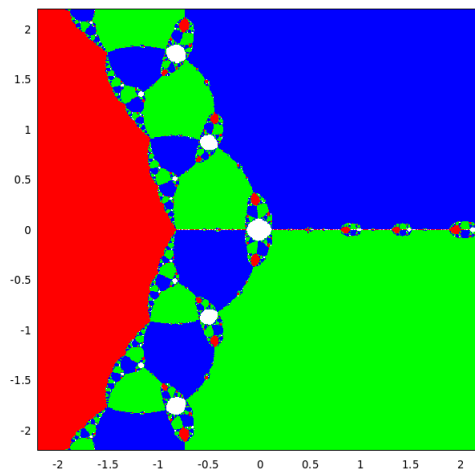


4 Proves realitzades

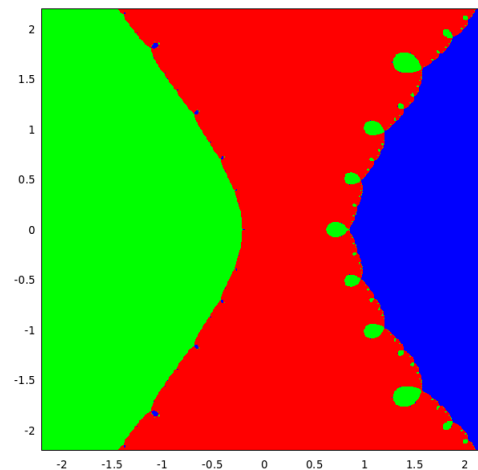
S'han fet una sèrie de proves amb el programa finalitzat per experimentar i veure diferents fractals. Per trobar les arrels s'ha fet servir Wolfram¹.

4.1 Proves amb tres arrels

- $z^3 - 2z + 2$



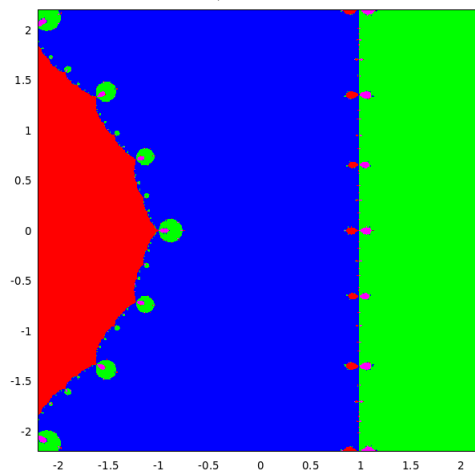
- $-2z^3 + 2z^2 + z$:



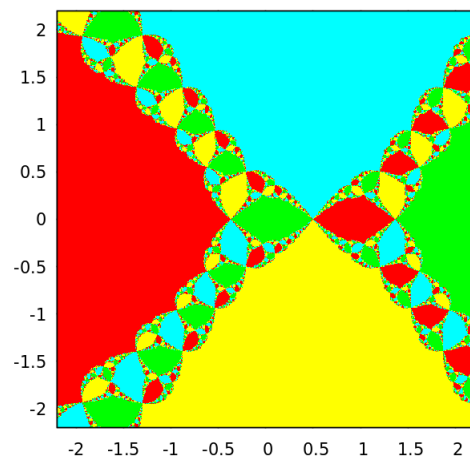
¹<https://www.wolframalpha.com/calculators/equation-solver-calculator>

4.2 Proves amb quatre arrels

- $z^4 - 4z^3 - 2z^2 + 12z - 3$:

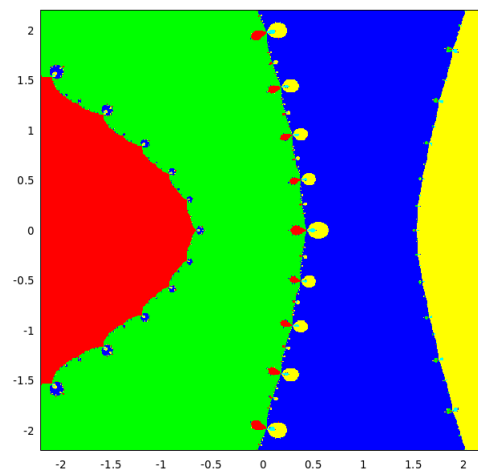


- $z^4 - 2z^3 + z - 2$:

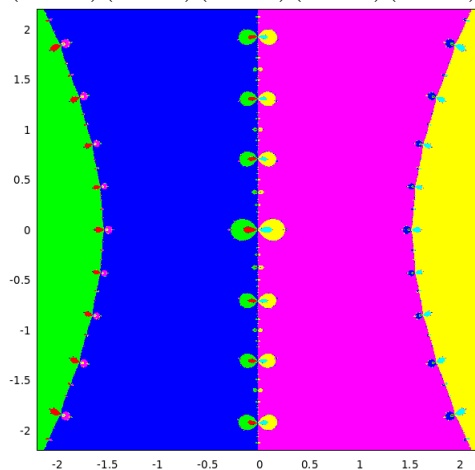


4.3 Proves amb més de quatre arrels

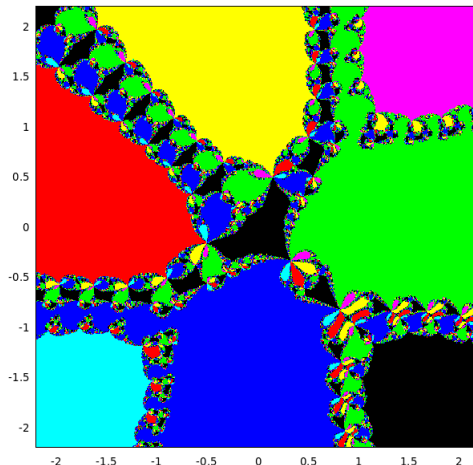
- $z^5 - 5z^4 + 5z^3 + 5z^2 - 6z - 1$ (5 arrels):



- $(z + 3)(z + 2)(z + 1)(z - 3)(z - 2)(z - 1)$ (6 arrels):



- $z^8 + 15x^4 - 16$ (7 arrels):



4.4 Conclusions de les proves

El programa actual representa correctament fins a set arrels, amb vuit o més s'hauria de canviar el sistema de colors per donar opció a més possibles combinacions de R, G, B. Com podem observar a les proves realitzades, hi ha fractals amb un centre com podria ser el de $z^4 - 2z^3 + z - 2$ i d'altres que consisteixen en unes franges més o menys verticals com es el cas de $(z + 3)(z + 2)(z + 1)(z - 3)(z - 2)(z - 1)$.

Com a curiositat, a la fractal de $z^3 - 2z + 2$ les zones blanques són degudes al fet que els punts no pertanyen a cap conca d'atracció, i per molt que s'augmenti el nombre d'iterats sempre queden punts blancs tot i que cada cop es fan una mica més petits.