W zadaniu miałam za zadanie zaznaczyć baseny abstrakcji rozwiązań równania $z^3 - 1$=0 na płaszczyźnie (Re z, Im z). Miejsca zerowe równania zespolonego możemu znaleźć metodą Newtona.

Metoda Newtona

Jeśli funkcja

$$g(x) = x - \frac{f(x)}{f'(x)} \quad (1)$$

jest ciągła oraz przeprowadza pewien domknięty przedział w ten sam przedział domknięty, to na mocy twierdzenia Brouwera iteracja ma w tym przedziale punkt stały, będacy rozwiązaniem równania $f(x) = 0$.

Metoda ta polega więc na iterowaniu wielokrotnie (1), dopóki nie znajdziemy się wystarczająco blisko miejsca zerowego. Niestety metoda ta może okazać się rozbieżna lub prowadzić do cykli, co uniemożliwi dotarcie do rozwiązania.

Do wyznaczenia basenów atrakcji posłużyłam się właśnie tą metodą. Basen atrakcji danego miejsca zerowego jest to przedstawienie na płaszczyźnie, z których punktów początkowych trafimy do właśnie tego miejsca zerowego. Granice basenów atrakcji miejsc zerowych na płaszczyźnie zespolonej bardzo często są fraktalne.

Rozwiązaniami równania $z^3 - 1$=0 na płaszczyźnie zespolonej są 3 pierwiastki:
$z_1 = 1$ $z_2 = -\frac{1}{2} + \frac{\sqrt{3}}{2}i$ $z_3 = -\frac{1}{2} - \frac{\sqrt{3}}{2}i$

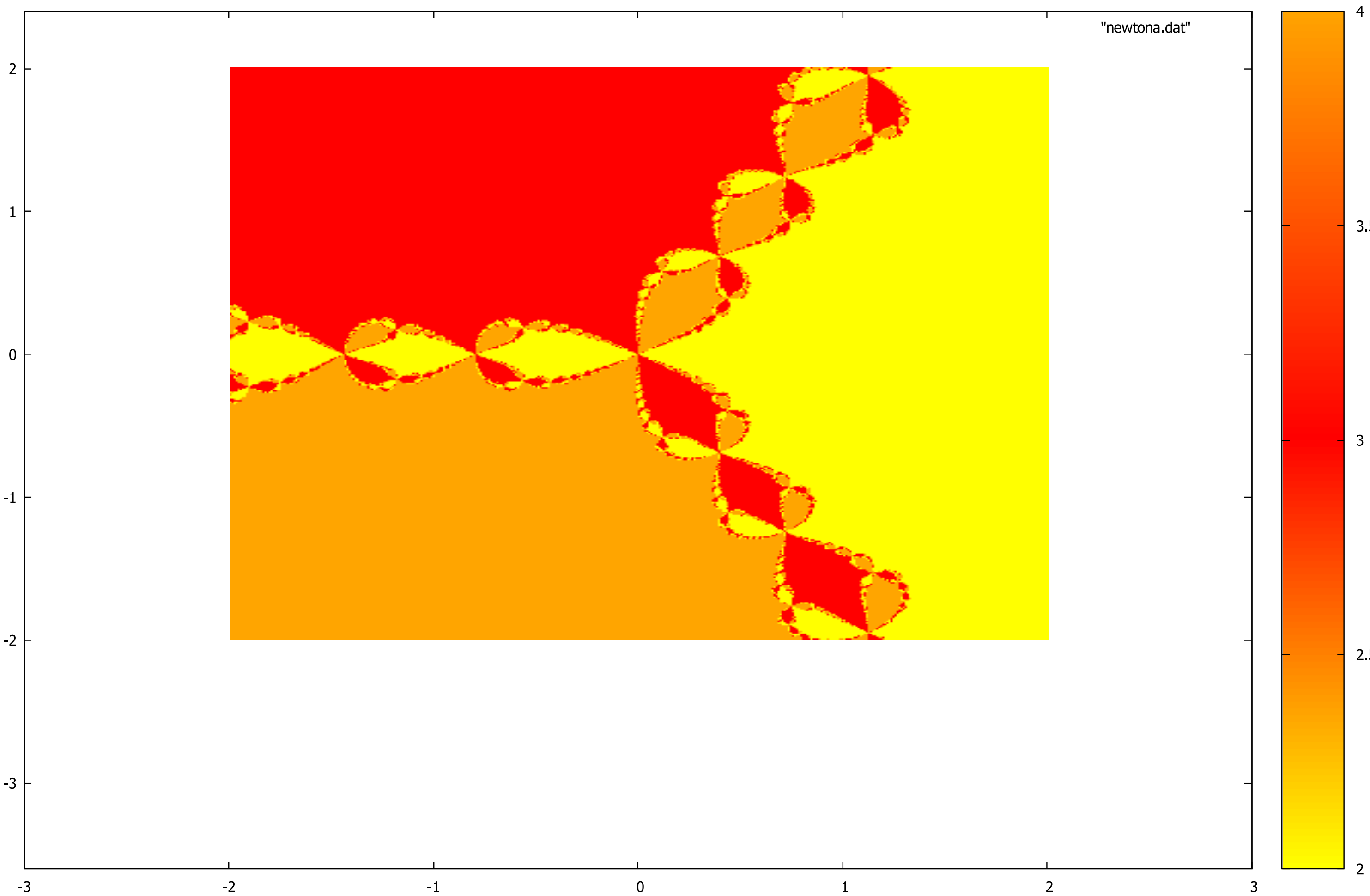
W moim programie, posłużyłam się już znanymi mi wcześniej miejscami zerowymi. Jego działanie przebiega nastepująco:

Rozpoczynam poszukiwanie miejsca zerowego z początkowego punktu na płaszczyźnie (x,y): u mnie było to (-2,-2i). Następnie, gdy uda mi się dotrzeć do któregoś z miejsc zerowych, to zaznaczam, które z miejsc zerowych to było (każde miejsce zerowe ma swój unikalną wartość).Jeśli nie udało się osiągnąć żadnego miejsca zerowego to również taki punkt otrzymuje wartość nadawaną tym punktom, które nie prowadzą do żadnego rozwiązania. Dzięki temu moja płaszczyzna (Re z, Im z) zostaje podzielona na różne baseny atrakcji danych miejsc zerowych, bo przyporządkowuje każdemu punktowi (x,y) jego basen atrakcji.

Wnioski

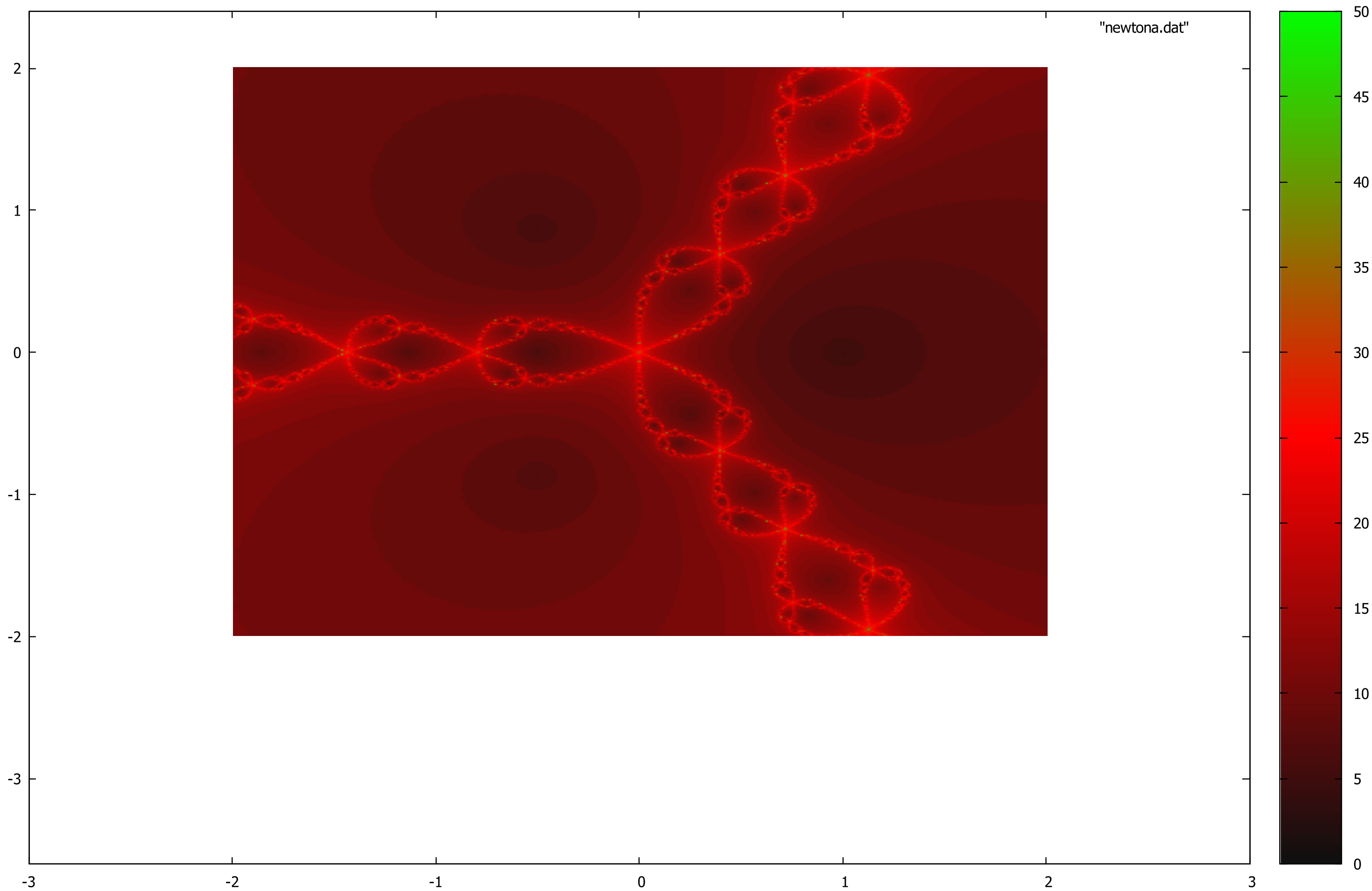
Program wyznacza basen atrakcji poprzez odszukanie miejsc zerowych metodą Newtona. Metoda Newtona jest uruchamiana w programie dla 400 równo oddalonych od siebie punktów. Dla każdego z tych punktów maskymalnie wykonuję 50 iteracji liczenia przybliżenia punktu stałego i w każdej z tych iteracji 3 razy sprawdzam czy to jedno z miejsc zerowych. Dzięki temu otrzymuje 160000 punktów na płaszczyźnie z dokładnością do 10^{-8}.

W gnuplot prezentuje się to nastepująco:



Powyżej zaznaczone są 3 baseny atrakcji, dla 3 różnych miejsc zerowych. Widać tutaj jak niewielka zmiana punktu z którego rozpoczynamy przybliżanie punktu stałego wpływa na to, do którego miejsca zerowego metoda będzie zbiegać.

Ponizej prezentuje również wykres, w którym modyfikuję numer basenu atrakcji na liczbę zależną od liczby iteracji - im mniej iteracji jest potrzebnych tym ciemniejszy jest kolor - widać na nim również gdzie znajdują się miejsca zerowe.



Aby uruchomić program należy wykonać komende : make all, a następnie make run.

Kod tego rozwiązania prezentuje się nastepująco:

```c
#include <stdio.h>
#include <math.h>
#include <complex.h>
#include <stdlib.h>

double complex funkcja(double complex z){
return z*z*z-1;
}

double complex pochodnaFunkcji(double complex z){
return 3*z*z;
}


int main(){
FILE *plik;
if((plik=fopen("newtona.dat","w"))==NULL) return -1;
int n=0;
int N=400;
double newtona(double complex z){
int i=0,j=0,counter=0;
double epsilon=1e-8;
double roznicaX=0.0;
double roznicaY=0.0;
double complex pierwiastki[3]={1,-0.5+I*sqrt(3)/2.0,-0.5-I*sqrt(3)/2.0};
double nrBasenu[3]={1,2,3};
double temp=0;
for(i=0;i<50;i++){
z-=(funkcja(z)/pochodnaFunkcji(z));
counter++;
for(j=0;j<3;j++){
roznicaX=creal(z)-creal(pierwiastki[j]);
roznicaY=cimag(z)-cimag(pierwiastki[j]);
if(fabs(roznicaX)<epsilon && fabs(roznicaY)<epsilon ) {
temp =((double) counter);
counter=0;
return nrBasenu[j]+temp;
}
}
}
return 4;
}
double pixelX[N];
double pixelY[N];
double complex liczba=-2.0-I*2.0;
double pixelZ[N][N];
pixelY[0]=-2.0; pixelX[0]=-2.0; pixelZ[0][0]=newtona(liczba);
double complex liczba2=0;
int i =1,j=1;
```

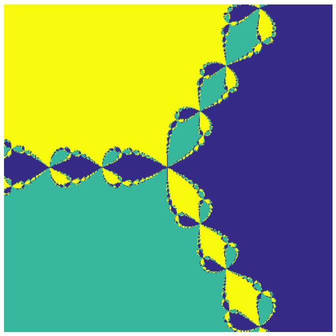
```
for(i=1;i<N;i++){
pixelX[i]=pixelX[i-1]+4/399.;
pixelY[i]=pixelY[i-1]+4/399.;
}

for(i=1;i<N;i++){
for(j=1;j<N;j++){
liczba2=pixelX[i]+I*pixelY[j];
pixelZ[i][j]=newtona(liczba2);
// printf("numer %d %d \n",i, j);
fprintf(plik,"%.8f %.8f %.8f\n",pixelX[i],pixelY[j],pixelZ[i][j]);

}


}
fclose(plik);
return 0;
}
```

Na koniec dodam,że użyłam też programu Matlab aby zaznaczyć te basenu atrakcji. Wygląda to tak:



A kod jest o wiele krótszy:

```
clear
[x,y]=meshgrid([-2:0.01:2]);
z=x+1i*y;
for n =1 :1000
imagesc(ceil(mod(angle(z)+pi/3,2*pi)/(2*pi/3)));
axis equal;
axis off;
drawnow;
z=z-(z.^3-1)./(3*z.^2);
end
```