Metody obliczeniowe w nauce i technice

SPRAWOZDANIE

Ćwiczenie 6
PIERWIASTKI FUNKCJI

Łukasz Gajewski

I. Cel ćwiczenia

- W GSL metody znajdowania pierwiastkow rownan sa podzielone na dwie grupy (trzy metody z grupy korzystająca ze zmiany znaku oraz trzy z grupy korzystajaca z pochodnej).
- Dla wybranej funkcji jednej zmiennej porownac działanie wszystkich sześciu metod.
- Dla kazdej grupy pokazac przynajmniej jeden przypadek, kiedy metody zawodza.
- Narysować wykres funkcji uzywajac octave: $(x-1)^2(x+1)^3 + \frac{1}{x^4+0.1} - 0.731$
    - Dokonac przyblizenia ewentualnych miejsc zerowych na wykresie (zawężając przedział wykresu).
    - Co obserwujemy ?
- Zbadać działanie metod dostępnych w octave: fsolve oraz fzero dla poszukiwania pierwiastków tej funkcji.
    - Zwrocic uwage na dodatkowe informacje zwracane przez funkcje fsolve (info).
    - Czy wszystkie znalezione pierwiastki sa dobre ?
- Namalowac wstegę Newtona stosujac wybraną implementację metody Newtona dla wybranego wielomianu liczb zespolonych (mozna skorzystac z metod poszukiwania pierwiastkow w wielu wymiarach w GSL.)

II. Porównanie działania metod znajdowania pierwiastków używając biblioteki GSL

a) metody korzystające ze zmiany znaku

1. *gsl_root_fsolver_bisection*

W każdej iteracji obszar jest dzielony na pół i obliczana jest wartość funkcji w połowie przedziału. Znak wyliczonej wartości jest używany do zdeterminowania w której połówce leży pierwiastek. W ten sposób uzyskujemy mniejszy podział, aż do uzyskania istotnie małego przedziału.

2. *gsl_root_fsolver_falsepos*

W każdej iteracji rysujemy linię pomiędzy (a,f(a)) oraz (b,f(b)), gdzie punkt przecięcia linii z osią x jest brany pod uwagę jako „środek przedziału". Następnie analogicznie jak w bisekcji.

3. *gsl_root_ffsolver_brent*

W każdej iteracji aproksymujemy funkcję używając krzywej. Pierwsza iteracja to liniowa interpolacja końcowych punktów, dla kolejnych iteracji algorytm używa dopasowanie odwrotnie kwadratowe do ostatnich trzech punktów, aby zwiększyć dokładność. Gdy przecięcie z osią x mieści się w obecnym przedziale to idziemy dalej, w przeciwnym wypadku bisekcja.

b) metody korzystające z pochodn

1. *gsl_root_fdfsolver_newton*

Algorytm przyjmuje początkowy punkt i za każdą iteracją tworzy styczną do funkcji. Punkt, w którym styczna przecina się z osią x jest kolejnym potencjalnym punktem, w którym może znajdować się pierwiastek funkcji.

$$x_{i+1} = x_i - \frac{f(x_i)}{f'(x_i)}$$

2. *gsl_root_fdfsolver_secant*

Idea pozostaje taka sama jak w metodzie Newtona, jednak liczymy pochodną w punkcie tylko w pierwszej iteracji. W każdej następnej korzystamy ze wzoru:

$$x_1 = x_0 - \frac{f(x_0)}{f'(x_0)} \qquad x_{i+1} = x_i * \frac{f(x_i)}{f'_{est}}, \quad gdzie\ \ f'_{est} = \frac{f(x_i) - f(x_{i-1})}{x_i - x_{i-1}}$$

3. *gsl_root_fdfsolver_steffenson*

Szybszy od powyższych. Korzysta z algorytmu Newtona oraz akceleracji Aitkena „delta-squared”. Produkujemy nową sekwencję Ri:

$$R_i \ = x_i \ - \ (x_{i+1} \ - \ x_i)^2 \ / \ (x_{i+2} - \ 2\, x_{i+1} + \ x_i)$$

c) porównanie działania powyższych metod dla funkcji $\boldsymbol{f(x) = x^2 - 5}$

bisekcja

```
Selected 1 method
using bisection method
 iter [    lower,     upper]      root        err  err(est)
    1 [0.0000000, 2.5000000] 1.2500000 -0.9860680 2.5000000
    2 [1.2500000, 2.5000000] 1.8750000 -0.3610680 1.2500000
    3 [1.8750000, 2.5000000] 2.1875000 -0.0485680 0.6250000
    4 [2.1875000, 2.5000000] 2.3437500 +0.1076820 0.3125000
    5 [2.1875000, 2.3437500] 2.2656250 +0.0295570 0.1562500
    6 [2.1875000, 2.2656250] 2.2265625 -0.0095055 0.0781250
    7 [2.2265625, 2.2656250] 2.2460938 +0.0100258 0.0390625
    8 [2.2265625, 2.2460938] 2.2363281 +0.0002601 0.0195312
    9 [2.2265625, 2.2363281] 2.2314453 -0.0046227 0.0097656
   10 [2.2314453, 2.2363281] 2.2338867 -0.0021813 0.0048828
   11 [2.2338867, 2.2363281] 2.2351074 -0.0009606 0.0024414
Converged:
   12 [2.2351074, 2.2363281] 2.2357178 -0.0003502 0.0012207
```

newton

```
Selected 4 method
using newton method
iter        root        err   err(est)
    1  3.0000000 +0.7639320 -2.0000000
    2  2.3333333 +0.0972654 -0.6666667
    3  2.2380952 +0.0020273 -0.0952381
Converged:
    4  2.2360689 +0.0000009 -0.0020263
[lgajewski@localhost Lab6]$
```

falsepos

```
Selected 2 method
using falsepos method
 iter [    lower,     upper]      root        err  err(est)
    1 [1.0000000, 2.5000000] 1.0000000 -1.2360680 1.5000000
    2 [2.1428571, 2.5000000] 2.1428571 -0.0932108 0.3571429
    3 [2.2307692, 2.3214286] 2.2307692 -0.0052987 0.0906593
    4 [2.2359686, 2.2760989] 2.2359686 -0.0000994 0.0401303
    5 [2.2360671, 2.2560338] 2.2360671 -0.0000009 0.0199667
    6 [2.2360680, 2.2460504] 2.2360680 -0.0000000 0.0099825
    7 [2.2360680, 2.2410592] 2.2360680 -0.0000000 0.0049912
    8 [2.2360680, 2.2385636] 2.2360680 -0.0000000 0.0024956
Converged:
    9 [2.2360680, 2.2360680] 2.2360680 +0.0000000 0.0000000
[lgajewski@localhost Lab6]$
```

secant

```
Selected 5 method
using secant method
iter        root        err   err(est)
    1  3.0000000 +0.7639320 -2.0000000
    2  2.5000000 +0.2639320 -0.5000000
    3  2.2727273 +0.0366593 -0.2272727
    4  2.2380952 +0.0020273 -0.0346320
Converged:
    5  2.2360845 +0.0000165 -0.0020108
[lgajewski@localhost Lab6]$
```

brent

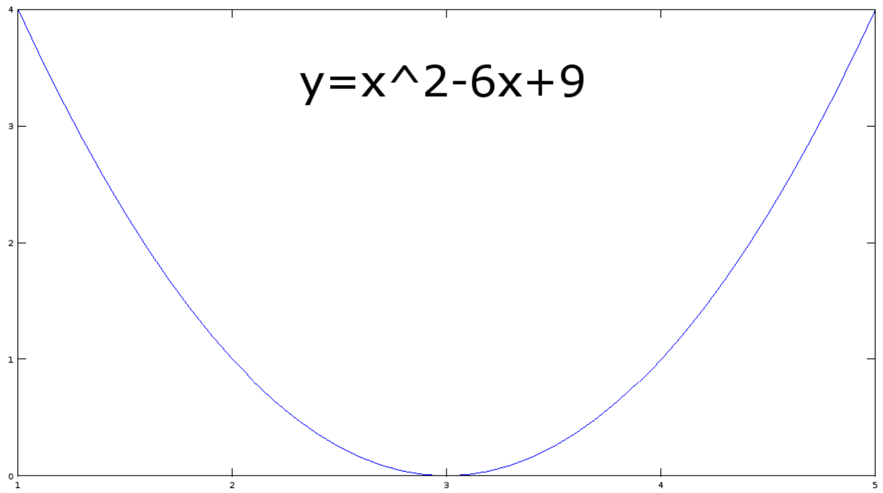
```
Enter digit: 3

Selected 3 method
using brent method
 iter [    lower,     upper]      root        err  err(est)
    1 [1.0000000, 5.0000000] 1.0000000 -1.2360680 4.0000000
    2 [1.0000000, 3.0000000] 3.0000000 +0.7639320 2.0000000
    3 [2.0000000, 3.0000000] 2.0000000 -0.2360680 1.0000000
    4 [2.2000000, 3.0000000] 2.2000000 -0.0360680 0.8000000
    5 [2.2000000, 2.2366300] 2.2366300 +0.0005621 0.0366300
Converged:
    6 [2.2360634, 2.2366300] 2.2360634 -0.0000046 0.0005666
[lgajewski@localhost Lab6]$
```

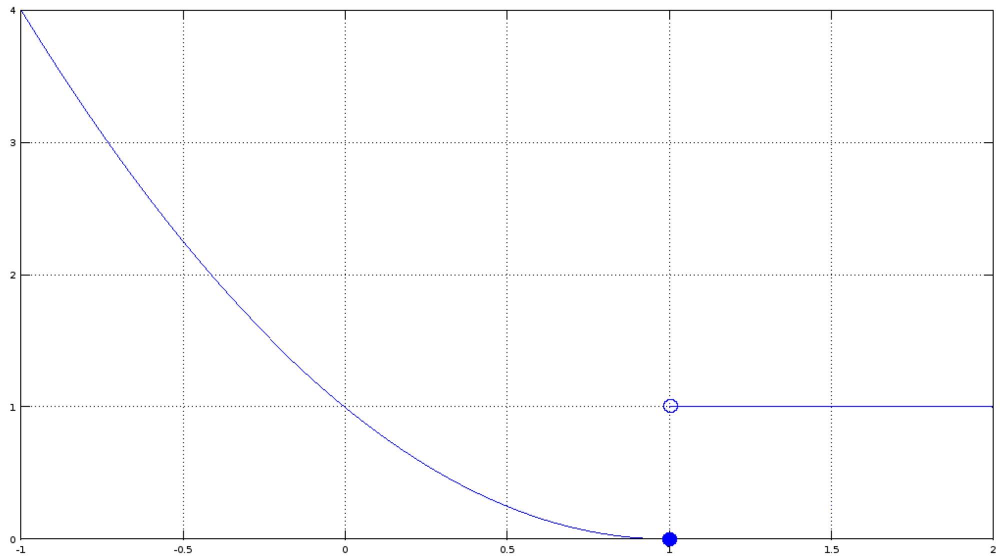
steffenson

```
Selected 6 method
using steffenson method
iter        root        err   err(est)
    1  3.0000000 +0.7639320 -2.0000000
    2  2.3333333 +0.0972654 -0.6666667
    3  2.2222222 -0.0138458 -0.1111111
    4  2.2360248 -0.0000431  0.0138026
Converged:
    5  2.2360680 -0.0000000  0.0000431
[lgajewski@localhost Lab6]$
```

III. Przykłady, dla których funkcje z biblioteki GSL zawodzą

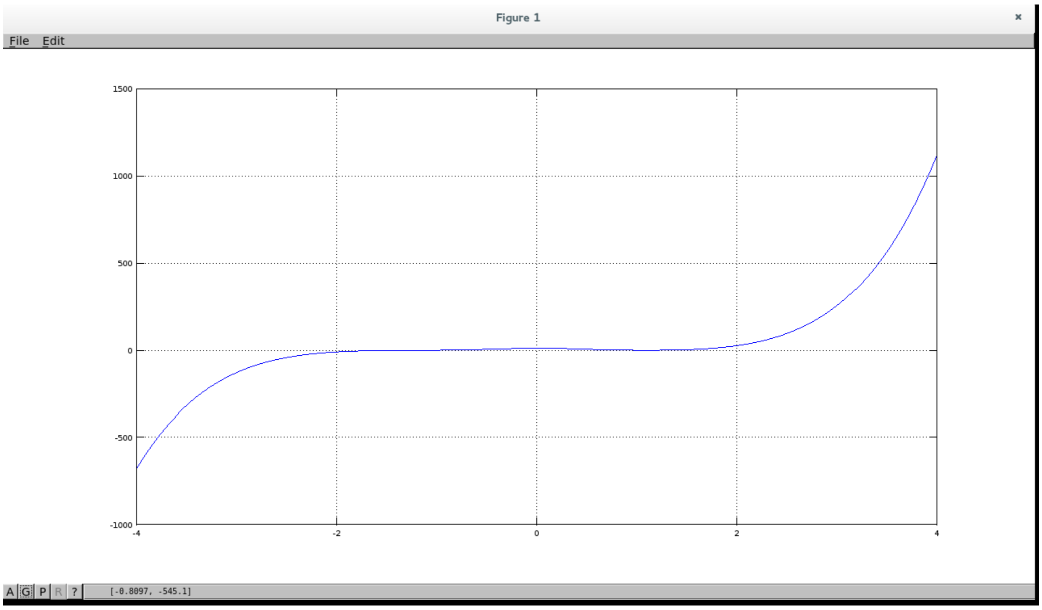


Funkcje bazujące na zmianie znaku nie radzą sobie z podwójnymi pierwiastkami.

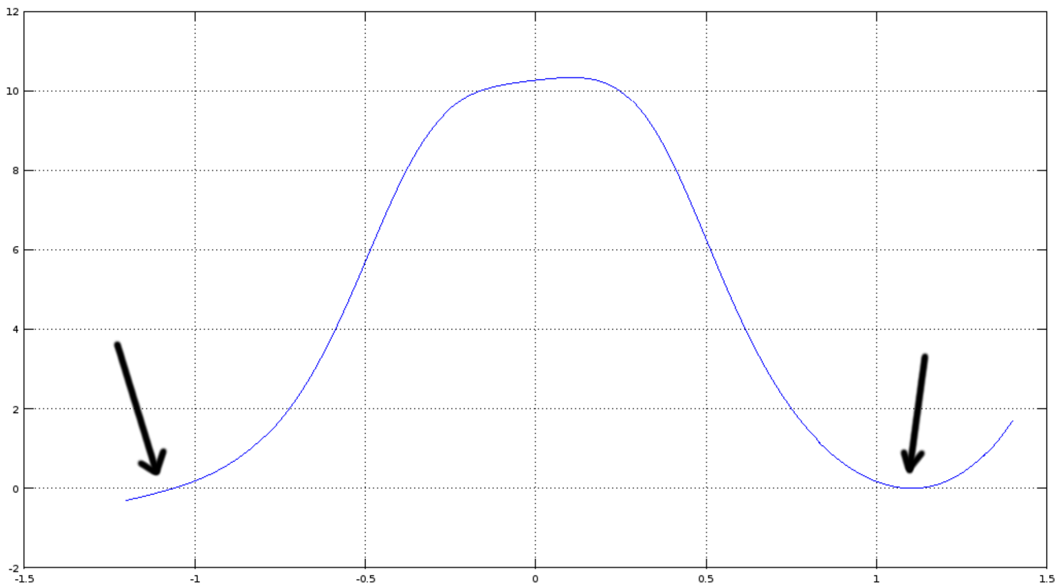


Funkcje bazujące na pochodnej nie radzą sobie z funkcjami, gdzie pochodna jest równa 0.

IV. Wykres funkcji - OCTAVE



$$(x-1)^2(x+1)^3+\frac{1}{x^4+0.1}-0.731$$



Wykres pokazuje możliwe miejsca zerowe funkcji (obserwujemy zmianę znaku i funkcję bliską zeru)

V. Wyznaczanie miejsc zerowych przy pomocy narzędzia octave

a) *Function File: fsolve (fcn, x0, options)*
Function File: [x, fvec, info, output, fjac] = fsolve (fcn, …)

$$x_0 - punkt\ początkowy$$
$$info - informacja\ na\ temat\ znalezionego\ pierwiastka$$
$$fvec - wartość\ w\ punkcie$$

Możliwe wartości info:

1 Converged to a solution point. Relative residual error is less than specified by TolFun.
2 Last relative step size was less that TolX.
3 Last relative decrease in residual was less than TolF.
0 Iteration limit exceeded.
-3 The trust region radius became excessively small.

b) Function File: fzero *(fun, x0)*
Function File: fzero *(fun, x0, options)*
Function File: *[x, fval, info, output]* = fzero *(…)*

Możliwe wartości info:

1 The algorithm converged to a solution.
0 Maximum number of iterations or function evaluations has been reached.
-1 The algorithm has been terminated from user output function.
-5 The algorithm may have converged to a singular point.

```
@(x) (((x - 1) .^ 2) .* ((x + 1) .^ 3) + (1 ./ ((x .^ 4) + 0.1)) - 0.731)

octave:36> [x, fval, info, output] = fzero (f, 0)
```

```
octave:32> [x, fvec, info, output, fjac] = fsolve (f, [-2;2])
```

fsolve	fzero

fsolve

```
x =

  -1.0608
   1.1058

fvec =

   4.5984e-10
   3.9670e-04

info =  3
output =

  scalar structure containing the fields:

    iterations =  33
    successful =  21
    funcCount =  77

fjac =

   2.6060e+00  -7.3509e-17
   7.0280e-15   4.9558e-06
```

fzero

```
x = -1.0608
fval =    3.6484e-08
info =  1
output =

  scalar structure containing the fields:

    iterations =  18
    funcCount =  28
    bracketx =

      -1.0608  -1.0608

    brackety =

      -4.4409e-16   3.6484e-08
```

VI. Wstęga Newtona przy pomocy metod poszukiwania pierwiastków w wielu wymiarach w GSL.

$$f(x) = z^3 - 1, \quad z = x + yi$$

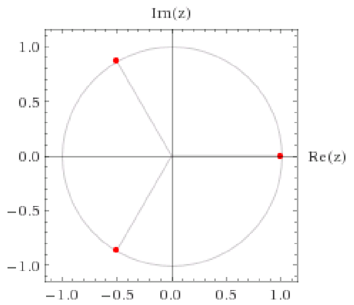
– *<u>wyrażenia pomocnicze</u>*

Complex solutions:

$z \approx -0.50000 - 0.86603\,i$

$z \approx -0.50000 + 0.86603\,i$

Roots in the complex plane:



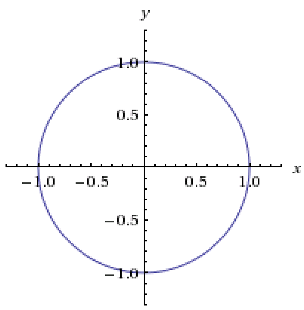
Input:

$\sqrt{x^2 + y^2} = 1$

Geometric figure:

circle

Implicit plot:

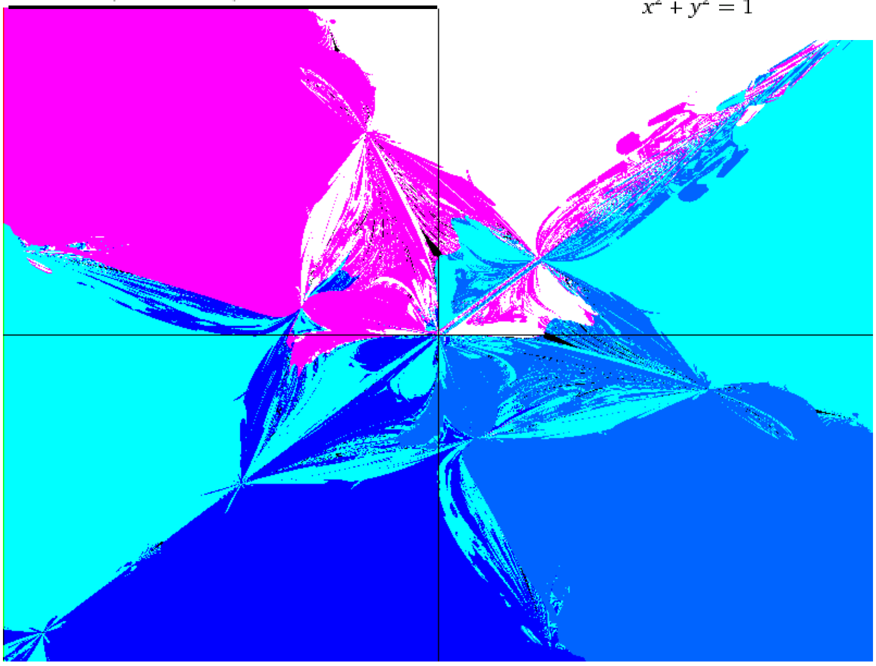


Alternate form assuming x and y are real:

$x^3 + i\,(3\,x^2\,y - y^3) - 3\,x\,y^2 - 1 = 0$

Alternate form:

$x^2 + y^2 = 1$



Wstęga Newtona, fraktal Newtona dla $z^3 - 1 = 0$

Łukasz Gajewski