Sprawozdanie

Laboratorium 1
Metody obliczeniowe w nauce i technice

Wykonanie: Kamil Kurp

Zadanie 3
Podać i zademonstrować własny przykład algorytmu niestabilnego numerycznie

Wybrałem przykład z Analizy numerycznej Davida Kincaida

Aby wyjaśnić to pojęcie, posłużymy się przykładem. Rozważmy ciąg liczb rzeczywistych określony wzorem rekurencyjnym

$$x_0 = 1, \quad x_1 = \frac{1}{3}, \qquad x_{n+1} = \frac{13}{3}x_n - \frac{4}{3}x_{n-1} \quad (n \geqslant 1). \tag{2.3.1}$$

Łatwo sprawdzić, że te związki generują ciąg o elementach

$$x_n = \left(\frac{1}{3}\right)^n. \tag{2.3.2}$$

Istotnie, to wyrażenie jest zgodne z (2.3.1) dla $n = 0$ i $n = 1$, a jeśli jest tak dla $n \leqslant m$, to i dla $n = m + 1$, gdyż

$$x_{m+1} = \frac{13}{3}x_m - \frac{4}{3}x_{m-1} = \frac{13}{3}\left(\frac{1}{3}\right)^m - \frac{4}{3}\left(\frac{1}{3}\right)^{m-1} = \left(\frac{1}{3}\right)^{m-1}\left(\frac{13}{9} - \frac{4}{3}\right) =$$
$$= \left(\frac{1}{3}\right)^{m+1}.$$

W tym przykładzie mamy dane dwa ciągi: jeden dany wzorem ogólnym, drugi wzorem rekurencyjnym, które są sobie równoważne.

Napisałem program, który oblicza kolejne elementy obu wzorów celem ich porównania.

```
Pierwsza kolumna: elementy ciagu x{0} = 1, x{1} = 1/3, x{n+1} = 13/3*x{n} - 3/4*x{n-1}
Druga kolumna: elementy ciagu x{n} = (1/3)^n
Trzecia kolumna: roznica miedzy elementami
Wzor rekurencyjny i ogolny sa rownowazne
1       1       0
0.33333333      0.33333333      0
0.11111111      0.11111111      1.6653345e-16
0.037037037     0.037037037     7.7021722e-16
0.012345679     0.012345679     3.1624009e-15
0.0041152263    0.0041152263    1.267389e-14
0.0013717421    0.0013717421    5.0707051e-14
0.00045724737   0.00045724737   2.0283233e-13
0.00015241579   0.00015241579   8.1133066e-13
5.080526e-05    5.0805263e-05   3.2453232e-12
1.6935075e-05   1.6935088e-05   1.2981293e-11
5.6449773e-06   5.6450293e-06   5.1925172e-11
1.8814687e-06   1.8816764e-06   2.0770069e-10
6.2639467e-07   6.2722547e-07   8.3080275e-10
2.0575195e-07   2.0907516e-07   3.323211e-09
5.6398875e-08   6.9691719e-08   1.3292844e-08
-2.9940803e-08  2.3230573e-08   5.3171376e-08
-2.0494198e-07  7.7435244e-09   2.126855e-07
-8.4816084e-07  2.5811748e-09   8.5074202e-07
-3.4021077e-06  8.603916e-10    3.4029681e-06
-1.3611585e-05  2.867972e-10    1.3611872e-05
-5.4447393e-05  9.5599066e-11   5.4447489e-05
-0.00021778992  3.1866355e-11   0.00021778996
-0.00087115981  1.0622118e-11   0.00087115982
-0.0034846393   3.5407062e-12   0.0034846393
-0.013938557    1.1802354e-12   0.013938557
-0.055754229    3.934118e-13    0.055754229
-0.22301691     1.3113727e-13   0.22301691
-0.89206766     4.3712422e-14   0.89206766
-3.5682706      1.4570807e-14   3.5682706
-14.273083      4.8569357e-15   14.273083
-57.09233       1.6189786e-15   57.09233
-228.36932      5.3965953e-16   228.36932
-913.47728      1.7988651e-16   913.47728
-3653.9091      5.996217e-17    3653.9091
-14615.637      1.998739e-17    14615.637
```

Jak widać na obrazku różnica jest niewielka dla początkowych elementów, ale zaczyna szybko wzrastać, osiągając błąd rzędu 10000 dla 35 elementu.

Zadanie 4

Proszę rozpakować zadania zad_gsl.tar.gz (używając poleceń gunzip i tar) i skompilować oraz uruchomić program `dokladnosc.c`. Następnie korzystając z funkcji `gsl_ieee_printf_double` zobaczyć, jak zmienia się mantysa i cecha dla coraz mniejszych liczb. Kiedy mantysa nie jest w postaci znormalizowanej?

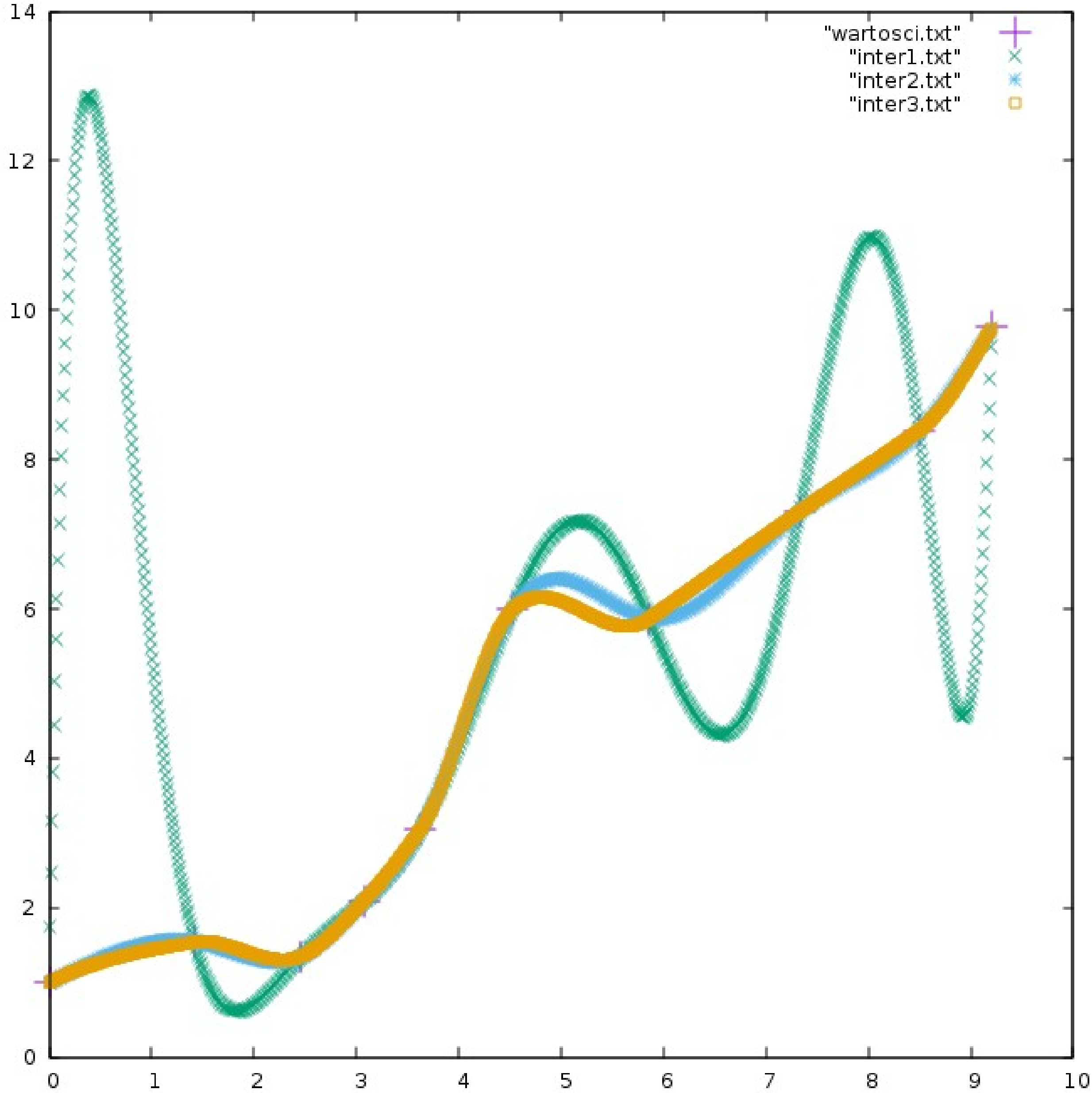
Sprawdziłem jak zmienia się mantysa i cecha poprzez dzielenie zmiennej o wartości początkowej 1/3 przez 2 w pętli.

```
 f= 1.01010101010101010101011*2^-3
fd= 1.0101010101010101010101100000000000000000000000000000*2^-3
 d= 1.0101010101010101010101010101010101010101010101010101*2^-3

 f= 1.01010101010101010101011*2^-4
fd= 1.0101010101010101010101100000000000000000000000000000*2^-4
 d= 1.0101010101010101010101010101010101010101010101010101*2^-4

 f= 1.01010101010101010101011*2^-5
fd= 1.0101010101010101010101100000000000000000000000000000*2^-5
 d= 1.0101010101010101010101010101010101010101010101010101*2^-5

 f= 1.01010101010101010101011*2^-6
fd= 1.0101010101010101010101100000000000000000000000000000*2^-6
 d= 1.0101010101010101010101010101010101010101010101010101*2^-6

 f= 1.01010101010101010101011*2^-7
fd= 1.0101010101010101010101100000000000000000000000000000*2^-7
 d= 1.0101010101010101010101010101010101010101010101010101*2^-7

 f= 1.01010101010101010101011*2^-8
fd= 1.0101010101010101010101100000000000000000000000000000*2^-8
 d= 1.0101010101010101010101010101010101010101010101010101*2^-8

 f= 1.01010101010101010101011*2^-9
fd= 1.0101010101010101010101100000000000000000000000000000*2^-9
 d= 1.0101010101010101010101010101010101010101010101010101*2^-9

 f= 1.01010101010101010101011*2^-10
fd= 1.0101010101010101010101100000000000000000000000000000*2^-10
 d= 1.0101010101010101010101010101010101010101010101010101*2^-10

 f= 1.01010101010101010101011*2^-11
fd= 1.0101010101010101010101100000000000000000000000000000*2^-11
 d= 1.0101010101010101010101010101010101010101010101010101*2^-11

 f= 1.01010101010101010101011*2^-12
fd= 1.0101010101010101010101100000000000000000000000000000*2^-12
 d= 1.0101010101010101010101010101010101010101010101010101*2^-12

 f= 1.01010101010101010101011*2^-13
fd= 1.0101010101010101010101100000000000000000000000000000*2^-13
 d= 1.0101010101010101010101010101010101010101010101010101*2^-13
```

Przy zmniejszaniu liczby poprzez dzielenie przez dwa można zauważyć dekrementację cechy tej liczby.

Mantysa nie jest w postaci znormalizowanej wtedy, kiedy nie zawiera się w przedziale [1, B), gdzie B – podstawa systemu liczbowego.

Zadanie 5
Proszę uruchomić program `interpolacja.c`. Korzystając ze skryptu gnuplotowego `wynik_plot` narysować wykres. Narysować na jednym wykresie krzywe otrzymane różnymi metodami interpolacji (w przykładzie ustawione jest `gsl_interp_polynomial`).

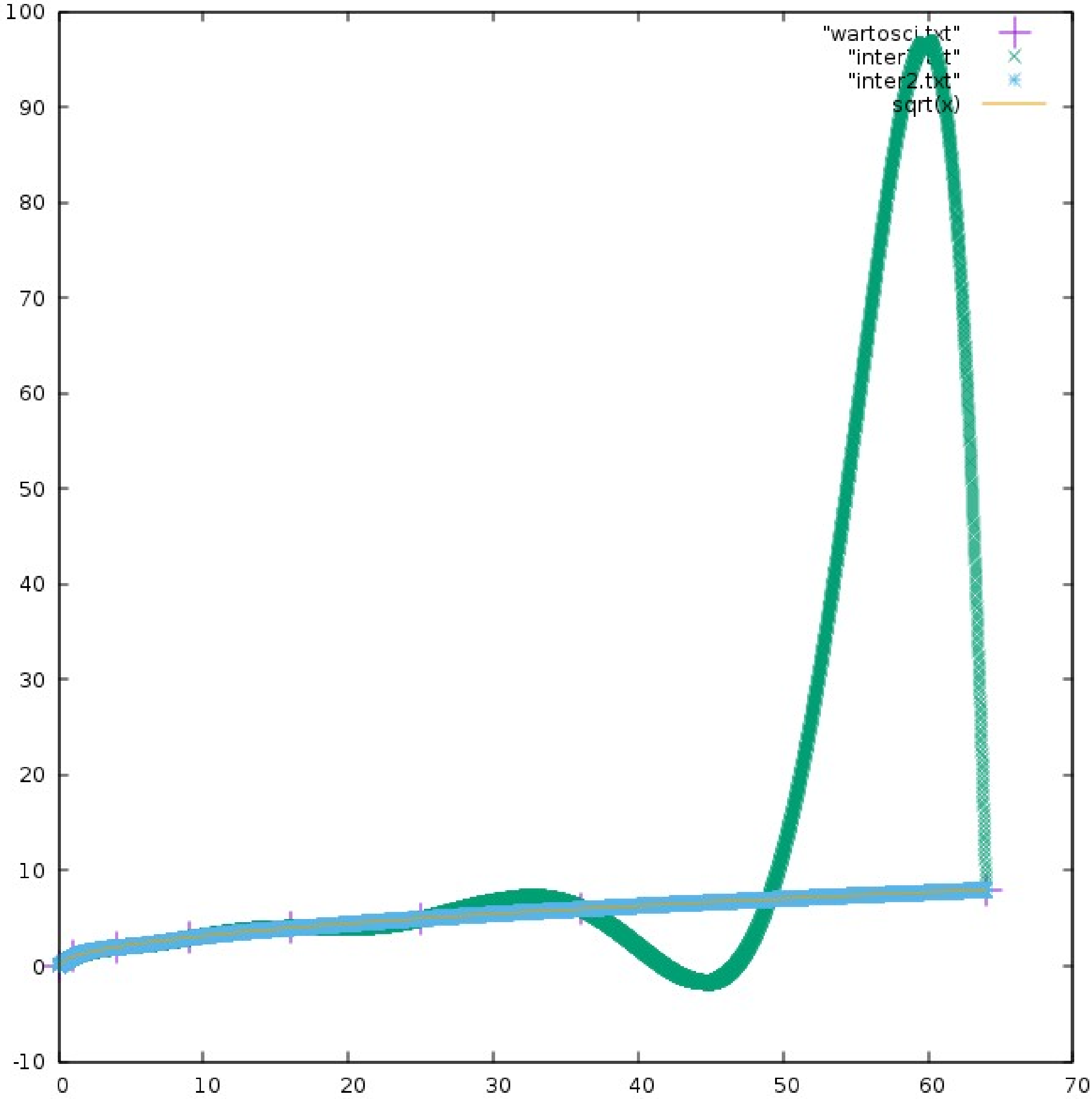


Wykorzystałem poniższe metody interpolacji:
1) interpolacja wielomianowa
2) interpolacja funkcjami sklejanymi
3) interpolacja funkcji sklejanej Akima (Akima spline)

Zadanie 6
Proszę wykonać interpolację dla punktów: (0,0), (1,1), (4,2), (9,3), (16,4), (25,5), (36,6), (49,7), (64,8) reprezentujących pierwiastek kwadratowy.

1. Obliczyć interpolację wielomianową stopnia 8. Na wykresie porównać z funkcją `sqrt`.
2. Obliczyć interpolację funkcjami sklejanymi. Na wykresie porównać z funkcją `sqrt`.
3. Która interpolacja jest dokładniejsza na całej dziedzinie [0,64]?
4. Która jest dokładniejsza na przedziale od 0 do 1?

1, 2)



1) interpolacja wielomianowa
2) interpolacja funkcjami sklejanymi

3, 4)

```
Wielomiany:
dla przedzialu od 0 do 1 suma roznic wartosci wynosi 11.658517
dla przedzialu od 0 do 64 suma roznic wartosci wynosi 82280.957686
Funkcje sklejane:
dla przedzialu od 0 do 1 suma roznic wartosci wynosi 14.475949
dla przedzialu od 0 do 64 suma roznic wartosci wynosi 120.558010
```

Licząc sumę różnic między funkcją interpolowaną a rzeczywistą wartością w danym punkcie dla wielu punktów przedziału (różnica stała między punktami 0.01) można dojść do wniosku, że na przedziale od 0 do 1 interpolacja wielomianowa daje dokładniejsze wartości, ale dla całej dziedziny znacznie lepsze wyniki daje interpolacja funkcjami sklejanymi.