Obliczenia Naukowe Lista 2 Laboratoria

Piotr Szyma

1 listopada 2017

1 Zadanie 1

1.1 Opis problemu

Powtórz zadanie 5 z listy 1, ale usuń ostatnią 9 z x_4 i ostatnią 7 z x_5. Jaki wpływ na wyniki mają niewielkie zmiany danych?

1.2 Rozwiązanie

Uruchomiłem program z pierwszej listy po dokonaniu modyfikacji wejścia. Wyniki zestawiłem w tabeli poniżej. Numeracja algorytmów to odpowiednio 1, 2 - algorytmy sumowania odpowiednio w górę i w dół, 3, 4 - sumy częściowe.

1.3 Wynik

Float64

algorytm	*przed*	*po*
1	$1.025188e-10$	$-4.296343e-03$
2	$-1.564331e-10$	$-4.296343e-03$
3	$0.000000e+00$	$-4.296343e-03$
4	$0.000000e+00$	$-4.296343e-03$

Float32

algorytm	*przed*	*po*
1	$-4.999443e-01$	$-4.999443e-01$
2	$-4.543457e-01$	$-4.543457e-01$
3	$-5.000000e-01$	$-5.000000e-01$
4	$-5.000000e-01$	$-5.000000e-01$

W przypadku arytmetyki Float64 lekka zmiana wektorów przyczyniła się do zmiany ostatecznego wyniku - wciąż znacznie odbiegajacego od realnych wartości. Dla arytmetyki Float32 usunięcie ostatnich cyfr nie miało wpływu na ostateczny wynik, gdyż arytmetyka nie jest wystarczająco dokładna.

ANALIZA WYNIKU

2 Zadanie 2

2.1 Opis problemu

Narysować wykres funkcji $f(x) = e^x \ln(1+e^{-x})$ w co najmniej dwóch dowolnych programach do wizualizacji. Następnie policzyć granicę funkcji $\lim_{x\to\infty} f(x)$. Porównać wykres funkcji z policzoną granicą. Wyjaśnić zjawisko.

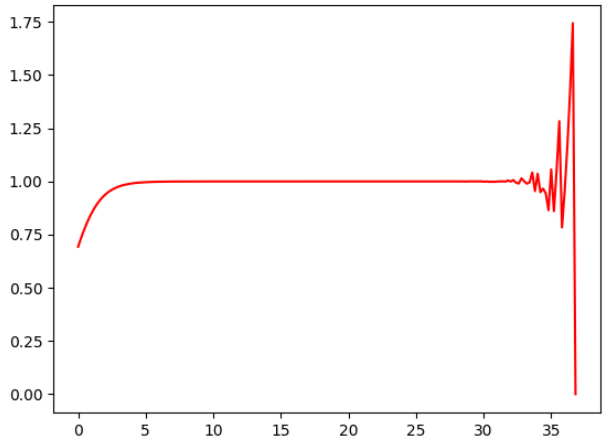
2.2 Rozwiązanie

Oto granica funkcji $f(x)$:

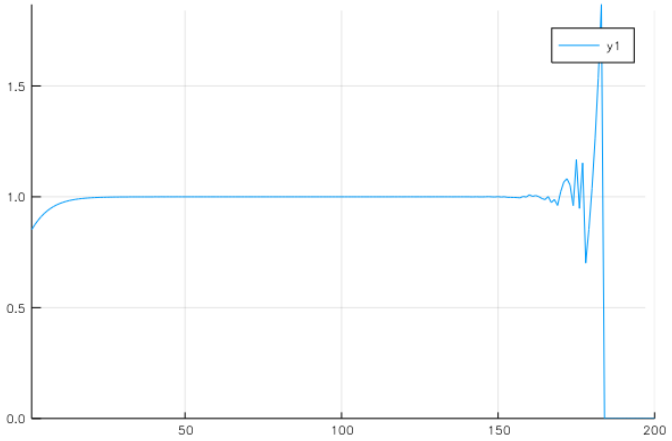
$$\lim_{x\to\infty} e^x \ln(1+e^{-x}) = 1$$

2.3 Wynik

Wykresy wygenerowane za pomocą dwóch programów do wizualzacji załączone są na Rysunkach 1 i 2.



Rysunek 1: Wykres wygenerowany za pomocą biblioteki matplotlib w języku Python



Rysunek 2: Wykres wygenerowany za pomocą biblioteki Plot w języku Julia

ANALIZA WYNIKU

3 Zadanie 3

3.1 Opis problemu

Rozwiązać układy równań podane w specyfikacji zadania za pomocą dwóch algorytmów: eliminacji Gaussa ($\texttt{x} = \texttt{A/b}$) oraz $x = A^{-1}b$ ($\texttt{x} = \texttt{inv(A)} * \texttt{b}$). Eksperymenty wykonać dla macierzy Hilberta $\mathbf{H_n}$ z rosnącym stopniem n > 1 oraz dla macierzy losowej R_n, n = 5, 10, 20 z rosnącym wskaźnikiem uwarunkowania $c = 10, 10^3, 10^7, 10^{12}, 10^{16}$. Porównać obliczony $\tilde{x}$ z rozwiązaniem dokładnym $x = (1, ..., 1)T$, tj. policzyć błędy względne.

W języku Julia za pomocą funkcji `cond(A)` można sprawdzić jaki jest wskaźnik uwarunkowania wygenerowanej macierzy. Natomiast za pomocą funkcji `rank(A)` można sprawdzić jaki jest rząd macierzy.

3.2 Rozwiązanie

W celu wykonania podanych w treści zadania obliczeń, stworzyłem w języku Julia odpowiednie funkcje. Błędy względne obliczeń wyliczyłem przy pomocy normy wektora wg. wzoru:

$$\delta = \frac{||\widetilde{x} - x||}{||x||}$$

Poniżej przedstawiłem tabelę - zestawienie wyników.

3.3 Wynik

Macierz Hilberta

n	$gauss$	$random$
1	$0.0000000000e+00$	$0.0000000000e+00$
2	$5.6610488670e-16$	$1.4043333874e-15$
3	$8.0225937723e-15$	$0.0000000000e+00$
4	$4.4515459602e-13$	$4.0996530222e-13$
5	$1.6828426299e-12$	$3.2543043466e-12$
6	$2.6189133023e-10$	$1.2883031376e-10$
7	$1.1328142969e-08$	$4.0328055817e-09$
8	$3.5398446689e-07$	$3.8628224721e-07$
9	$6.0153150454e-06$	$1.1373234205e-05$
10	$2.5639289838e-04$	$4.7875286566e-04$
11	$9.2062284978e-03$	$1.3773749165e-02$
12	$2.1237698492e-01$	$8.1189912224e-02$
13	$5.6147916245e+00$	$3.7001925765e+00$
14	$5.0203088003e+01$	$5.3099680319e+01$
15	$2.1930762488e+00$	$5.9233167411e+00$
16	$9.3939371128e+00$	$7.3739406019e+00$
17	$1.8900539672e+00$	$7.7696770018e+00$
18	$1.7078652201e+01$	$3.3606846658e+01$
19	$1.9533296871e+01$	$1.1440959715e+01$
20	$2.2728620198e+01$	$3.3526120880e+01$

Losowa macierz o zadanym współczynniku *cond*

n	$cond$	$gauss$	$random$
5	1	$1.7199501140e-16$	$7.0216669372e-17$
5	3	$1.9860273226e-16$	$7.0216669372e-17$
5	7	$0.0000000000e+00$	$9.9301366130e-17$
5	12	$4.9650683065e-17$	$7.0216669372e-17$
5	16	$2.2204460493e-16$	$4.0943002132e-16$
10	1	$2.8737410464e-16$	$2.1925147984e-16$
10	3	$2.9582808635e-16$	$2.4575834280e-16$
10	7	$3.3674731643e-16$	$3.0807438657e-16$
10	12	$3.4577699598e-16$	$2.7644330376e-16$
10	16	$3.1791949214e-16$	$3.7975470272e-16$
20	1	$5.8904629596e-16$	$4.9836525323e-16$
20	3	$4.0716587481e-16$	$3.1499827100e-16$
20	7	$3.4488466676e-16$	$3.7731252496e-16$
20	12	$3.6316343785e-16$	$4.5165750266e-16$
20	16	$7.7992673607e-16$	$8.1319573640e-16$

ANALIZA WYNIKU

4 Zadanie 4

4.1 Opis problemu

„Złośliwy wielomian" Wilkinsona. Zainstalować pakiet `Polynomials`.

(a) Użyć funkcji `roots` z pakietu `Polynomials` do obliczenia 20 zer wielomianu P zadanego w treści zadania. Sprawdzić obliczone pierwiastki $z_k, 1 \leq k \leq 20$, obliczając $|P(z_k)|$, $|p(z_k)|$ i $|z_k - k|$. Wyjaśnić rozbieżność.
Zapoznać się z funkcjami `Poly`, `poly`, `polyval` z pakietu `Polynomials`.

(b) Powtórzyć eksperyment Wilkinsona, tj. zmienić współczynnik -210 na $-210 - 2^{-23}$. Wyjaśnić zjawisko.

4.2 Rozwiązanie

Funkcja `Poly(x)` generuje wielomian w zależności od współczynników podanych jako parametr. Funkcja `poly(x)` tworzy wielomian na podstawie pierwiastków wielomianu, natomiast funkcja `polyval(p, x)` pozwala na wyliczenie wartości wielomianu `p` dla zadanego argumentu `x`.

4.3 Wynik

Wyniki

k	$\|P(z_k)\|$	$\|p(z_k)\|$	$\|z_k - k\|$
1	3744.9845589677943	634368.0	$3.4638958368304884e-14$
2	0.286276201356511	$2.000828411463103e18$	1.5000000000121532
3	0.008870981937623812	$7.081833742276362e17$	2.666666671480031
4	0.001292543281409042	$6.975238303156173e16$	3.749999292770549
5	0.0002468752348492309	$5.532738743218158e17$	4.800030116128743
6	$6.359660288768332e-5$	$8.711755039228508e17$	5.832765543928463
7	$1.5599974479103758e-5$	$1.1467631332545471e18$	6.860997629758284
8	$1.5599974479103758e-5$	$1.1467631332545471e18$	7.860997079449626
9	$3.294359732426998e-6$	$1.4150728707341862e18$	8.887894252414426
10	$3.294359732426998e-6$	$1.4150728707341862e18$	9.887892124924674
11	$6.147094764624633e-7$	$1.635099126269708e18$	10.910961341089823
12	$6.147094764624633e-7$	$1.635099126269708e18$	11.91095953363977
13	$1.1060473806865536e-7$	$1.7942764158130138e18$	12.928459786425634
14	$1.1060473806865536e-7$	$1.7942764158130138e18$	13.928458857167234
15	$2.1833150474298537e-8$	$1.8986230967648804e18$	14.940250578500368
16	$2.1833150474298537e-8$	$1.8986230967648804e18$	15.94025025231326
17	$5.2738380570532775e-9$	$1.9603045451252319e18$	16.94726438591253
18	$5.2738380570532775e-9$	$1.9603045451252319e18$	17.94726431805519
19	$1.810441792686032e-9$	$1.9903054634328896e18$	18.950637706265248
20	$1.810441792686032e-9$	$1.9903054634328896e18$	19.950637702768113

ANALIZA WYNIKU

5 Zadanie 5

5.1 Opis problemu

Równanie rekurencyjne (model logistyczny, model wzrostu populacji)

$$p_{n+1} := p_n + rp_n(1 - p_n)\ dla\ n = 0, 1, 2, \ldots$$

Zadanie polegało na przeprowadzeniu następującego eksperymentu:

(a) Dla danych $p_0 = 0.01$ oraz $r = 3$ wykonać 40 iteracji w/w wyrażenia (1), następnie 40 iteracji z modyfikacją - po 10 iteracji obciąć wynik (2), odrzucając cyfry po 3 miejscu po przecinku. Obliczenia wykonać w arytmetyce `Float32`. Porównać wyniki.

(b) Porównać wyliczanie 40 iteracji dla arytmetyk `Float32` oraz `Float64`.

5.2 Rozwiązanie

W celu wyliczenia zadanych rekurencji stworzyłem program w Julii. Wyniki obliczeń przedstawiłem w tabeli poniżej.

5.3 Wynik

Wyniki (a)

n	(1)	(2)
1	0.039700001478195	0.039700001478195
2	0.154071733355522	0.154071733355522
3	0.545072615146637	0.545072615146637
4	1.288978099822998	1.288978099822998
5	0.171518802642822	0.171518802642822
10	0.722930610179901	0.722000002861023
15	1.270483732223511	1.257216930389404
20	0.579903602600098	1.309691071510315
25	1.007080554962158	1.092910766601563
30	0.752920925617218	1.319182157516479
35	1.021098971366882	0.034241437911987
40	0.258605480194092	1.093567967414856

Wyniki (b)

n	$Float32$	$Float64$
1	0.039700001478195	0.039700000000000
2	0.154071733355522	0.154071730000000
3	0.545072615146637	0.545072626044421
4	1.288978099822998	1.288978001188801
5	0.171518802642822	0.171519142109176
10	0.722930610179901	0.722914301179573
15	1.270483732223511	1.270261773935077
20	0.579903602600098	0.596529312494691
25	1.007080554962158	1.315588346001072
30	0.752920925617218	0.374146489639287
35	1.021098971366882	0.925382128557105
40	0.258605480194092	0.011611238029749

ANALIZA WYNIKU

6 Zadanie 6

6.1 Opis problemu

Dla zadanego równania rekurencyjnego przeprowadź serię eksperymentów.

$$x_{n+1} = x_{n,2} + c \; dla \; n = 0, 1, 2, \ldots$$

Wykonać w arytmetyce `Float64` 40 iteracji wyrażenia dla 7 zestawów danych podanych w treści zadania.

6.2 Rozwiązanie

W celu wyliczenia zadanych rekurencji stworzyłem program w Julii. Wyniki obliczeń przedstawiłem w tabeli poniżej.

6.3 Wynik

ANALIZA WYNIKU