Proszę, gdy dodacie jakieś zadanie to w razie niepewności co do jego poprawności wpiszcie “Do sprawdzenia” - na czerwono.

Jeśli ktoś sprawdza innym zadanka, niech zostawia komentarze(te pojawiające się po prawej stronie arkusza.)

Jakby ktoś przez przypadek popsuł coś w pliku, to luzik, jest historia zmian, więc można naprawić.

L15.1. W języku programowania `PWO++` funkcja `cos`(x) oblicza z bardzo dużą dokładnością wartość $\cos(x)$, jednak **tylko wtedy**, gdy $0 \leq x \leq \frac{\pi}{2}$. Wykorzystując funkcję `cos`, zaproponuj `algorytm` wyznaczający wartości funkcji cosinus z dużą dokładnością dla $x \in [-2\pi, 2\pi]$.

Rozwiązanie:

Idea: skorzystać ze wzorów redukcyjnych i monotoniczności funkcji cosinus, aby móc sprowadzić problem obliczania cosinusa z dowolnego przedziału do zadanego w treści zadania.

Rozwiązanie graficzne

L15.2. Jakie znaczenie z punktu widzenia analizy numerycznej ma pojęcie uwarunkowania zadania?

Rozwiązanie:
Pojęcie numerycznego uwarunkowania zadania określa wrażliwość wyniku na zaburzenia danych. Jeśli zadanie jest źle uwarunkowane to nie da sie go rozwiazac dokładnie żadnym algorytmem.
Do sprawdzenia
Zadanie nazywamy źle uwarunkowanym, jeśli mała względna zmiana danych zadania powoduje dużą względną zmianę wyniku.

Skoro z punktu widzenia analizy numerycznej, która zajmuje się przybliżaniem obliczeń, to może chodzi o to, że źle uwarunkowane zadanie uniemożliwia dokonania przybliżenia. Wtedy trzeba albo zdobyć dokładniejsze dane, albo rozpatrzyć inne zadanie.

L15.3. Sprawdź dla jakich wartości x zadanie obliczania wartości funkcji f jest źle uwarunkowane, jeśli: **a)** $f(x) = \ln(x)$, **b)** $f(x) = (x-1)^{10}$.

Rozwiązanie:

Idea rozwiązywania:

Oblicz $\frac{xf'(x)}{f(x)}$ następnie granicę kiedy mianownik dąży do 0

a) f(x) ma miejsce zerowe dla x=1, stąd liczymy granicę wskaźnika w tym punkcie. Dostajemy nieskończoność, więc zadanie jest źle uwarunkowane.
b) Sytuacja identyczna z punktu a. Stąd zadanie jest źle uwarunkowane.

Pełne rozwiązanie:

Aby sprawdzić uwarunkowanie tego zadania, musimy policzyć wskaźnik uwarunkowania: $\left|\frac{xf'(x)}{f(x)}\right|$.

Zadanie jest źle uwarunkowane w punkcie x', wtedy gdy

$$\lim_{x \to x'} \left|\frac{x f'(x)}{f(x)}\right| = \infty$$

a) $f(x) = \ln(x)$ $\qquad f'(x) = \frac{1}{x}$

wskaźnik uwarunkowania: $\left|\frac{1}{\ln(x)}\right|$ $\qquad \ln(1) = 0$

$$\lim_{x \to 1} \left|\frac{1}{\ln(x)}\right| = \infty$$

Zatem zadanie jest źle uwarunkowane w punkcie $x=1$.

b) $f(x) = (x-1)^{10}$ $\qquad f'(x) = 10(x-1)^9$

wskaźnik uwarunkowania: $\left|\frac{10x(x-1)^9}{(x-1)^{10}}\right| = \left|\frac{10x}{x-1}\right|$

$$\lim_{x \to 1} \left|\frac{10x}{x-1}\right| = \infty$$

L15.4. Podaj definicję zadania źle uwarunkowanego, a następnie zbadaj uwarunkowanie zadania obliczania wartości funkcji $f(x) = \cos x$ dla $x \in \mathbb{R}$.

Rozwiązanie:

a) Jeśli niewielkie względne zmiany danych zadania powodują duże względne zmiany jego rozwiązania, to zadanie takie nazywamy źle uwarunkowanym.

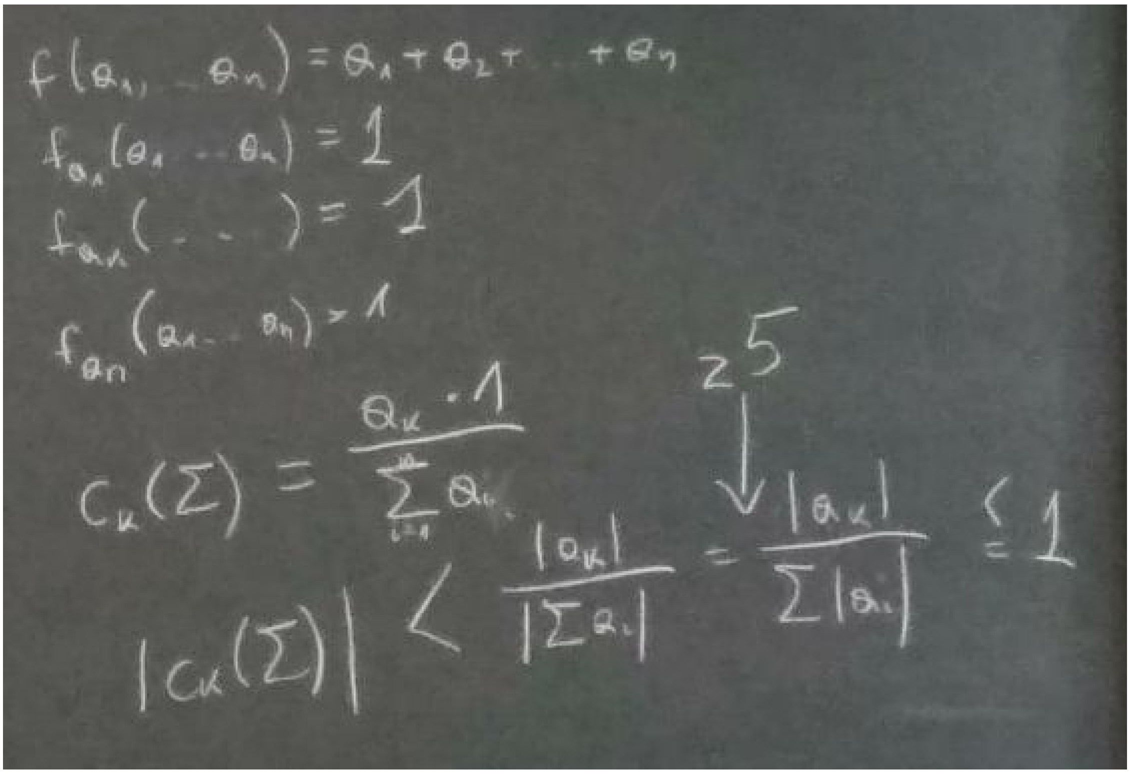
b) Granice wskaźnika uwarunkowania są równe nieskończoności w miejscach zerowych cos(x), stąd zadanie jest źle uwarunkowane w tych punktach.

Do sprawdzenia

L15.5. Załóżmy, że liczby $x_0, x_1, \ldots, x_n$ są tego samego znaku. Uzasadnij, że zadanie obliczania ich sumy jest zadaniem dobrze uwarunkowanym. Jakie znaczenie ma ten fakt w kontekście obliczeń numerycznych?

Rozwiązanie:

Wka to dobrze opisał:



f(a_1, a_2, ..., a_n) to funkcja obliczająca sumę będącą celem zadania

f_a_1, f_a_2, ..., f_a_n to wartości pochodnych cząstkowych po zmiennych a_1, a_2, ..., a_n

następnie używamy C_k(sigma), to jest wskaźnik uwarunkowania dla k-tej zmiennej a_k

A znaczenie jest pewnie takie, że źle uwarunkowanie zadanie wykonywane na liczbach w reprezentacji zmiennopozycyjnej, kumuluje błędy przy wykonywaniu działań i prowadzi do źle oszacowanego wyniku.

L15.6. Wytłumacz dokładnie kiedy występuje i na czym polega zjawisko utraty cyfr znaczących wyniku. Dla jakich wartości x obliczanie wartości wyrażenia $(\sqrt{x^2+2}+x)^{-1}$ może wiązać się z utratą cyfr znaczących wyniku? Zaproponuj sposób obliczenia wyniku dokładniejszego.

Rozwiązanie:

a) Zjawisko występuje gdy odejmujemy dwie bardzo bliskie liczby(to znaczy ich różnica jest niemal równa zeru).

b) Dla bardzo dużych(co do modułu) liczb ujemnych tzn. spełniających $\sqrt{x^2+2} \approx x$

c) $\frac{\sqrt{x^2+2} - x}{2}$

Gdzie dla liczb nieujemnych używamy bazowego wzoru a tego powyżej dla x < 0.

L15.7. Dla $x \approx 0$ obliczanie wartości wyrażenia $x^{-5}(\sin(3x) - 3x + 9x^3/2)$ może wiązać się z utratą cyfr znaczących wyniku. Zakładając, że $|x| \leq \frac{1}{10}$, zaproponuj taki sposób obliczenia wartości tego wyrażenia, aby mieć pewność, że błąd bezwzględny nie przekracza 10^{-7}.

Idea:

1. korzystamy z rozwinięcia sin(3x) w szereg Taylora - pierwszy i drugi wyraz powinny wyjść 3x oraz 9x^3/2 i skrócić się z tym co mam w wyrażeniu
2. otrzymujemy szereg naprzemienny - korzystamy z jego własności, która mówi, że moduł różnicy sumy szeregu i sumy częściowej do n-tego wyrazu jest mniejszy od wyrazu n+1-szego tzn. $|S - S_n| \leq a_{n+1}$ czyli szukamy wyrazu, który jest mniejszy lub równy zadanej wartości błędu

Rozwiązanie:

1.

Wiemy, że

$$\sin x = x - \frac{x^3}{3!} + \frac{x^5}{5!} - \frac{x^7}{7!} + \ldots$$

Stąd

$$\sin 3x = 3x - \frac{3^3x^3}{3!} + \frac{3^5x^5}{5!} - \frac{3^7x^7}{7!} + \ldots$$

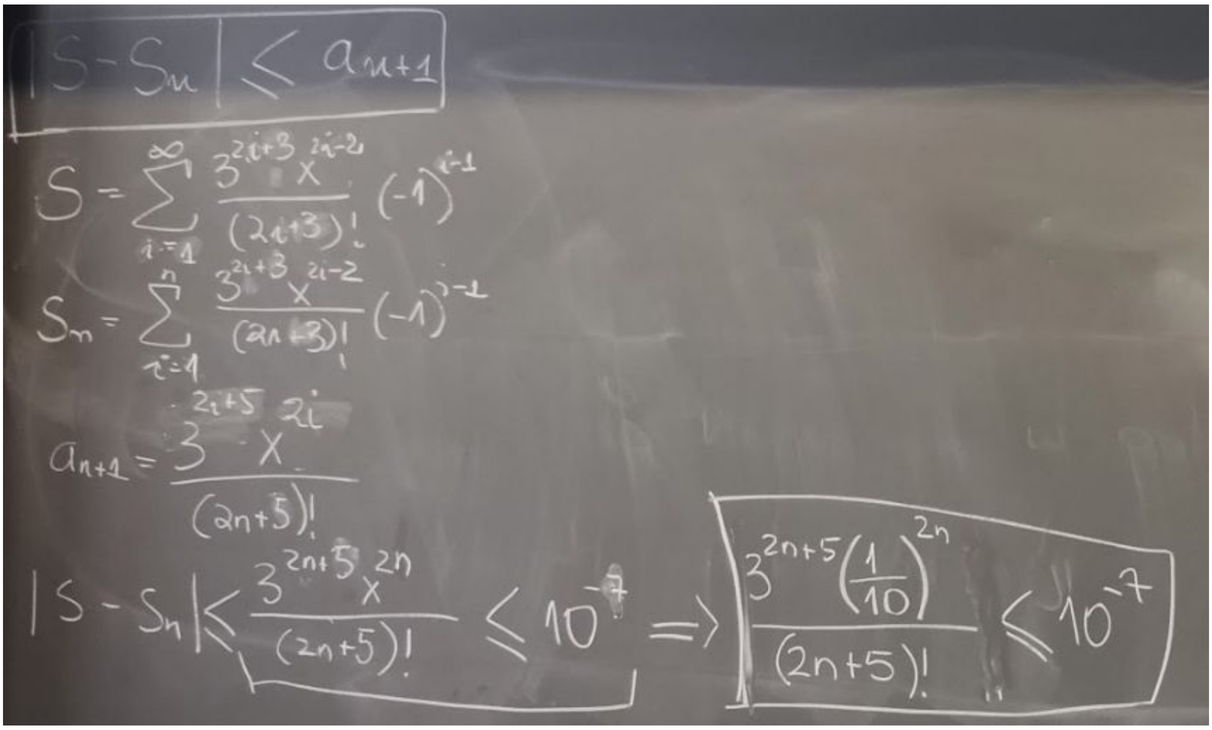
Podstawiamy pod wyrażenie.

Zauważamy, że -3x + (9x^3)/2 oraz dwa pierwsze elementy szeregu zerują się .

Zostaje

$$\frac{\frac{3^5x^5}{5!} - \frac{3^7x^7}{7!} + \ldots}{x^5}$$

2.



L15.8. Sprawdź czy następujący algorytm jest algorytmem numerycznie poprawnym:

```
S:=x[0];

  for i from 1 to 4
    do
      S:=3*S+x[i]
    od;

return(S)
```

https://imgur.com/a/7i6xqT1

15.8

$S_0 := x[0]$

$S_k = 3 \cdot S_{k-1} + x[i]$

Algorytm liczy S_4

$$S_4 = \underbrace{\left(3\cdot\underbrace{\left(3\cdot\underbrace{\left(3\cdot\underbrace{(3\cdot x_0 + x_1)}_{i=1} + x_2\right)}_{i=2} + x_3\right)}_{i=3} + x_4\right)}_{i=4}$$

Dodajmy błędy mnożenia i dodawania $|\alpha|, |\beta| \leq 2^{-t}$

$$S_4 = \Big(\Big(3\Big(\Big(\Big(3\cdot\Big(\Big(\Big(3\cdot\big((3\cdot x_0)(1+\alpha_1)+x_1\big)(1+\beta_1)\Big)\Big)(1+\alpha_2)\Big)+x_2\Big)(1+\beta_2)\Big)\Big)(1+\alpha_3)+$$

$$+x_3\Big)(1+\beta_3)\Big)(1+\alpha_4)\Big)+x_4\Big)(1+\beta_4) =$$

$$= 3\cdot(3\cdot(3\cdot(3\cdot x_0\overbrace{(1+\alpha_1)(1+\alpha_2)(1+\alpha_3)(1+\alpha_4)(1+\beta_1)(1+\beta_2)(1+\beta_3)(1+\beta_4)}^{(1+\varepsilon_0)})+$$

$$+x_1\overbrace{\prod_{k=2}^{4}(1+\alpha_k)\prod_{k=1}^{4}(1+\beta_k)}^{(1+\varepsilon_1)})+$$

$$+x_2\overbrace{\prod_{k=3}^{4}(1+\alpha_k)\prod_{k=2}^{4}(1+\beta_k)}^{(1+\varepsilon_2)})+$$

$$+x_3\overbrace{(1+\alpha_4)(1+\beta_3)(1+\beta_4)}^{(1+\varepsilon_3)})+$$

$$+x_4\overbrace{(1+\beta_4)}^{(1+\varepsilon_4)} =$$

$|\varepsilon_0| \leq 8\cdot 2^{-t}$

$|\varepsilon_1| \leq 7\cdot 2^{-t}$

$|\varepsilon_2| \leq 5\cdot 2^{-t}$

$|\varepsilon_3| \leq 3\cdot 2^{-t}$

$|\varepsilon_4| \leq 1\, 2^{-t}$

$$= 3\cdot(3\cdot(3\cdot(3\cdot x_0(1+\varepsilon_0))+x_1(1+\varepsilon_1))+x_2(1+\varepsilon_2))+x_3(1+\varepsilon_3))+x_4(1+\varepsilon_4)) =$$

$$= 3\cdot(3\cdot(3\cdot(3\cdot\tilde{x}_0)+\tilde{x}_1)+\tilde{x}_2)+\tilde{x}_3)+\tilde{x}_4$$

Dla nieco zaburzonych danych alg zwróci dokładny wynik czyli A.N.P

Do sprawdzenia

L15.9. Niech dany będzie wielomian $w(x) := a_1x/3! - a_3x^3/5! + a_5x^5/7! - a_7x^7/9!$. Rozważmy następujący algorytm obliczania jego wartości w punkcie $x \in \mathbb{R}$:

```
w:=a[7]

  for n from 3 downto 1
    do
      w:=a[2*n-1]-x^2/(2*n+3)/(2*n+2)*w
    od

return(w*x/2/3)
```

Przyjmując, że a_1, a_3, a_5, a_7 oraz x są liczbami maszynowymi, sprawdź czy algorytm ten jest algorytmem numerycznie poprawnym.

L15.10. Opisz metodę bisekcji i podaj jej własności.

Rozwiązanie:

Założenia:
Mamy ciągłą funkcję f w przedziale domkniętym [a, b] oraz zachodzi f(a)*f(b) < 0. Wtedy pomiędzy a i b istnieje co najmniej jedno miejsce zerowe funkcji.

Algorytm:

1. Sprawdzenie, czy pierwiastkiem równania jest punkt $x_1 = \frac{a+b}{2}$, czyli czy $f(x_1) = 0$.
   Jeżeli tak jest algorytm kończy działanie, a punkt x_1 jest szukanym miejscem zerowym.
2. W przeciwnym razie, dopóki nie osiągniemy żądanej dokładności, czyli dopóki $|a - b| > \epsilon$:
   1. Zgodnie ze wzorem z punktu pierwszego ponownie wyznaczane jest x_1, dzieląc przedział $[a, b]$ na dwa mniejsze przedziały: $[a, x_1]$ i $[x_1, b]$.
   2. Wybierany jest przedział o znaku przeciwnym niż x_1 i odpowiednio górny albo dolny kraniec przedziału (b albo a) przyjmuje wartość x_1, tj.
      1. Jeżeli $f(x_1)f(a) < 0$, to $b = x_1$,
      2. Jeżeli $f(x_1)f(b) < 0$, to $a = x_1$.
3. Po osiągnięciu żądanej dokładności algorytm kończy działanie, a szukany pierwiastek równania wynosi $\frac{a+b}{2}$.

Własności:
-znajduje tylko jedno miejsce zerowe
-zbieżna liniowo (każdy kolejny przedział zmniejsza się o połowę)
-zbieżność nie zależy od rozpatrywanej funkcji

L15.11. Stosując metodę Newtona, zaproponuj sposób przybliżonego obliczania wartości $\sqrt[5]{a}$ $(a > 0)$. Jak dobrać x_0? Jak powinien wyglądać warunek *stopu*?

Rozwiązanie:

a)

$$f(x) = x^5 - a$$

$$f'(x) = 5x^4$$

$$x_{k+1} = x_k - \frac{x_k^5 - a}{5x_k^4} = \frac{1}{5}\left(5x_k - x_k + \frac{a}{x_k^4}\right) = \frac{1}{5}x_k\left(4 + \frac{a_k}{x_k^5}\right)$$

na podstawie
https://pl.wikipedia.org/wiki/Metoda_Newtona#Przyk%C5%82ad

b) x0 musi być w pobliżu podejrzewanego miejsca zerowego f

c) |f(x)| < epsilon

Jeszcze są dwa warunki stopu (chyba że w rozpatrywanym przypadku wystarczy ten podany wyżej):

- n<= N_max (ograniczamy liczbę iteracji)
- Sprawdzamy, czy iteracje wciąż przybliżają nas do miejsca zerowego, tzn. liczymy: max(| (x_k+1 - x_k) / x_k |, | (x_k+2 - x_k+1) / x_k+1 |, ... , | (x_k+n+1 - x_k+n) / x_k+n |) < Jakaś_dana_mała_liczba

L15.12. Niech α będzie pierwiastkiem pojedynczym funkcji f $(f(\alpha) = 0,\ f'(\alpha) \neq 0)$. Udowodnij, że wówczas rząd zbieżności metody Newtona wynosi $p = 2$.

Przydatne w zadaniu:

Tw. Niech dana będzie metoda jednokrokowa

$$\begin{cases} x_0 - \text{dane} \\ x_{n+1} = F(x_n) \end{cases}$$

znajdowania miejsca zerowego α funkcji f

To znaczy, że $\lim_{n \to \infty} x_n = \alpha$

Rząd tej metody jest liczbą naturalną p $\Leftrightarrow$

$F(\alpha) = \alpha$ tzw. metoda punktu stałego

$F'(\alpha) = 0$

$F''(\alpha) = 0$

$\vdots$

$F^{p-1}(\alpha) = 0$

$F^{p}(\alpha) \neq 0$

Rozwiązanie:

$$f(x) = x$$

$$F(x) = x - \frac{f(x)}{f'(x)}$$

$$F(\alpha) = \alpha$$

$$F'(x) = \frac{f(x) \cdot f''(x)}{[f'(x)]^2}$$

$$F'(\alpha) = 0$$

$$F''(x) = \left(\frac{f(x) \cdot f''(x)}{[f'(x)]^2} \right)' = \frac{(f'(x) \cdot f''(x) + f(x) \cdot f'''(x)) \cdot [f'(x)]^2 - f(x) \cdot f''(x) \cdot [f'(x)]^4}{[f'(x)]^4}$$

$$F''(\alpha) = \frac{f''(\alpha) \cdot [f'(\alpha)]^3}{[f'(\alpha)]^4} \ != 0$$

Stąd rząd jest równy 2

Do sprawdzenia

Rozwiązanie z wikipedii (IMO lepsze): [KLIK TUTAJ]

L15.13. Zaproponuj efektywny algorytm obliczania z dużą dokładnością wartości $\sqrt{a}$ $(a > 0)$ wykorzystując **jedynie** operacje arytmetyczne $(+, -, \cdot, /)$.

Rozwiązanie:

Za pomocą metody Newtona można obliczyć pierwiastek $\sqrt{a}$ dla każdej liczby $a \in \mathbb{R}^+$:

$$\sqrt{a} = x \iff a = x^2 \iff x^2 - a = 0.$$

Funkcja *f*(*x*) ma postać:

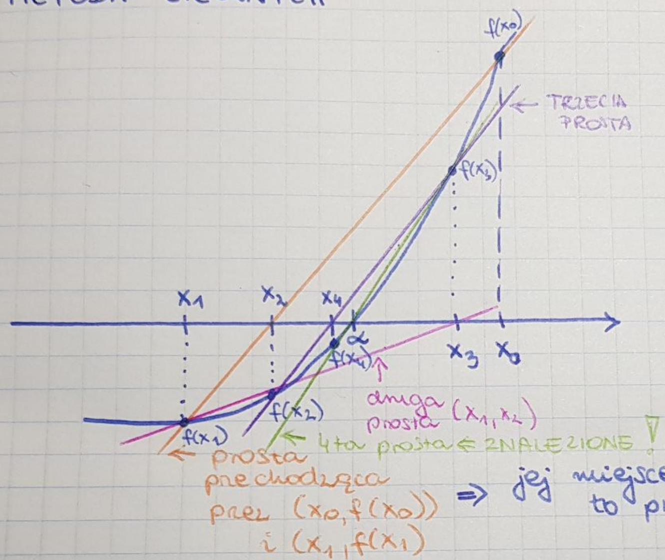
$$f(x) = x^2 - a,$$
$$f'(x) = 2x.$$

Rekurencyjny wzór wynosi:

$$x_{k+1} = x_k - \frac{x_k^2 - a}{2x_k},$$
$$x_{k+1} = \frac{1}{2}\left(x_k + \frac{a}{x_k}\right).$$

L15.14. Sformułuj i podaj interpretację geometryczną metody siecznych. Jak w wypadku tej metody powinien wyglądać *warunek stopu*?

METODA SIECZNYCH



⇒ jej miejsce zerowe to pkt x_2

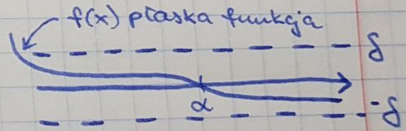
$$\begin{cases} x_0, x_1 - \text{dane} \\ x_{k+1} = x_k - \dfrac{f(x_k)}{\dfrac{f(x_k) - f(x_{k-1})}{x_k - x_{k+1}}} \end{cases} \quad k = 1, 2, \ldots$$

x_0, x_1 - dane, prowadzę prostą przez punkty $(x_0, f(x_0))$ i $(x_1, f(x_1))$, w jej miejscu zerowym mam x_2, później biorę x_1, x_2 i konstruuję prostą …

WARUNEK STOPU: 1. $|f(x_k)| < \delta$ gdzie δ - dana mała liczba

⇓

1 WARUNEK NIE WYSTARCZA !



2. Sprawdzamy czy metoda coś jeszcze wnosi:

$$\max\left(\left| \frac{x_{k+1} - x_k}{x_k} \right|, \left| \frac{x_{k+2} - x_{k+1}}{x_{k+1}} \right|, \ldots, \left| \frac{x_{k+n+1} - x_{k+n}}{x_{k+n}} \right| \right) < \varepsilon$$

(n ustalone, ε - mała liczba)

3. $n \leqslant N_{max}$ ⇐ OGRANICZAM ILOŚĆ ITERACJI (zapętlanie)

L15.15. Podaj efektywny algorytm wyznaczania wartości liczby naturalnej a, której cyframi dziesiętnymi (od najbardziej do najmniej znaczącej) są $a_n, a_{n-1}, \ldots, a_0$, gdzie $a_n \neq 0$.

Hornerem.

```
wynik := a_n
for i = n-1; i --; i >= 0
      wynik:= wynik * 10 + a_i
```

L15.16. Sformułuj i uzasadnij uogólniony schemat Hornera obliczania wartości wielomianu podanego w postaci Newtona.

Rozwiązanie:

Postać Newtona prezentuje się następująco

$$w(x) = a_0 + \sum_{i=1}^{n} a_i \prod_{j=0}^{i-1}(x - x_j) = a_0 + a_1(x - x_0) + a_2(x - x_1)(x - x_0) + \cdots + a_n(x - x_{n-1}) \cdots (x - x_1)(x - x_0)$$

złożoność: n-1 dodawań oraz (n+1 po 2) mnożeń

stąd w postaci uogólnionego schematu hornera:

$$w(x) = \Bigg(\bigg(\Big(\ldots \Big(\big(a_n\,(x - x_{n-1}) + a_{n-1}\big)(x - x_{n-2}) + a_{n-1}\Big) \ldots \Big)(x - x_2) + a_2\bigg)(x - x_1) + a_1\Bigg)(x - x_0) + a_0$$

złożoność: n-1 dodawań n-1 mnożeń

Do sprawdzenia

L15.17. Sformułuj i uzasadnij algorytm Clenshawa obliczania wartości wielomianu podanego w postaci Czebyszewa.

Wiadomo, że: $T_k = 2xT_{k-1}(x) - T_{k-2}(x)$

lepszy sposób na obliczanie:

Algorytm Clenshawa: dla danych $x, c_0, c_1, \ldots, c_n : w(x) \in \Pi_n$ obliczam:

$$\begin{cases} B_{n+2} = B_{n+1} = 0 \\ B_k = 2xB_{k+1} - B_{k+2} + c_k \quad (k = n-1, n-2, \ldots, 0) \end{cases}$$

wtedy $w(x) = \frac{B_0 - B_2}{2}$

$$w(x) = \frac{c_0}{2} \cdot T_0(x) + c_1 \cdot T_1(x) + c_2 \cdot T_2(x) + \ldots + c_n T_n(x)$$

więc $w(x) = \sum_{k=0}^{n}{}' c_k \cdot T_k(x) \quad (1)$

$$B_k = 2xB_{k+1} - B_{k+2} + c_k \Rightarrow (2)\ c_k = B_k - 2xB_{k+1} + B_{k+2}$$

$(2) \rightarrow (1)$

$$w(x) = \sum_{k=0}^{n}{}' (B_k - 2xB_{k+1} + B_{k+2}) \cdot T_k(x) =$$

rozdzielam na pojedyncze sumy:

$$= \underbrace{\sum_{k=0}^{n}{}' B_k \cdot T_k(x)}_{n+1 \text{ wyr.}} - \underbrace{\sum_{k=0}^{n}{}' 2xB_{k+1} \cdot T_k(x)}_{n \text{ wyrazów}} + \underbrace{\sum_{k=0}^{n}{}' B_{k+2} \cdot T_k(x)}_{n-1 \text{ wyrazów}} =$$

przeindeksowane

chcę wrzucić pod jedną sumę i usunąć wyr. zerowe

$$= \frac{1}{2}B_0 \cdot T_0(x) + B_1T_1(x) - xB_1T_0(x) + \sum_{k=2}^{n} B_kT_k(x) - \sum_{k=2}^{n} 2xB_kT_{k-1} + \sum_{k=2}^{n}{}' B_kT_{k-2} =$$

$$= \frac{1}{2}B_0 \cdot T_0(x) + \cancel{B_1x} - \cancel{xB_1} + \sum_{k=2}^{n} B_k\big(\underbrace{T_k(x) - 2xT_{k-1}(x) + T_{k-2}(x)}_{=0}\big) - \frac{1}{2}B_2T_0(x) =$$

$$= \frac{1}{2}B_0 \cdot T_0(x) - \frac{1}{2}B_2T_0(x) =$$

$$= \frac{B_0 - B_2}{2} = w(x)$$

bo $T_k(x) = 2xT_{k-1}(x) - T_{k-2}(x)$

$0 = 2xT_{k-1}(x) - T_k(x) - T_{k-2}(x)$

$0 = T_k(x) - 2xT_{k-1}(x) + T_{k-2}(x)$

L15.18. Podaj postać Newtona wielomianu interpolacyjnego $L_4 \in \Pi_4$ dla danych

$$\begin{array}{c||c|c|c|c|c} x_k & -2 & -1 & 1 & 2 & 3 \\ \hline y_k & 1 & 2 & 10 & 29 & 106 \end{array} .$$

Rozwiązanie:

Skorzystamy z ilorazów różnicowych

$$\begin{cases} f[x_i] = f(x_i) \quad (0 \leqslant i \leqslant N) \\ f[x_k, x_{k+1}, \ldots, x_{k+m}] = \frac{f[x_{k+1}, x_{k+2}, \ldots, x_{k+m}] - f[x_k, x_{k+1}, \ldots, x_{k+m-1}]}{x_{k+m} - x_k} \quad (0 \leq k < k+m \leq n) \end{cases}$$

x	y				
-2	1 = a_1				
-1	2	1 = a_2			
1	10	4	1 = a_3		
2	29	19	5	1 = a_4	
3	106	77	29	6	1 = a_5

Stąd postać Newtona wielomianu:
w(x) = 1 + (x+2) + (x+2)(x+1) + (x+2)(x+1)(x-1) + (x+2)(x+1)(x-1)(x-2)

L15.19. Podaj postać Newtona wielomianu interpolacyjnego dla następujących danych:

a)

x_k	-2	-1	0	1
y_k	2	0	2	-4

,

b)

x_k	1	2	-1	-2	0
y_k	-4	-30	0	2	2

.

Rozwiązanie analogiczne do zadania L15.18

a)

x	y			
-2	2 = a_0			
-1	0	-2 = a_1		
0	2	2	2 = a_2	
1	-4	-6	-4	-2 = a_3

Stąd postać Newtona wielomianu:

w(x) = 2 - 2 (x+2) + 2(x+2)(x+1) - 2(x+2)(x+1)x

b)

xk	yk				
1	-4 = a_0				
2	-30	-26 = a_1			
-1	0	-10	-8 = a_2		
-2	2	-2	-2	-2 = a_3	
0	2	0	2	-2	0 = a_4

Stąd postać Newtona wielomianu:

w(x) = - 4 - 26 (x-1) - 8 (x-1)(x-2) - 2 (x-1)(x-2)(x+1)

L15.20. Funkcję $f(x) = \cos(x/2)$ interpolujemy wielomianem $L_n \in \Pi_n$ w węzłach będących zerami wielomianu Czebyszewa T_{n+1}. Jak należy dobrać n, aby mieć pewność, że

$$\max_{x\in[-1,1]} |f(x) - L_n(x)| \leq 10^{-8} \ ?$$

Rozwiązanie:

$$\left|\left|f^{(n+1)}(\alpha)\right|\right| = \cos\left(\frac{\alpha}{2}\right) \cdot \left(\frac{1}{2}\right)^{(n+1)} \textit{lub} \ \sin\left(\frac{\alpha}{2}\right) \cdot \left(\frac{1}{2}\right)^{(n+1)}$$

$$p_{n+1} \leq \frac{1}{2^n}$$

cos i sin od jakiejś zmiennej α przyjmuje maksymalną wartość 1 stąd

$$\frac{\left(\frac{1}{2}\right)^{(n+1)}}{(n+1)! \cdot 2^n} \leq 10^{-8}$$

Stąd po wyciągnięciu n z nierówności dostajemy n >= 7

L15.21. Niech $L_n \in \Pi_n$ będzie wielomianem interpolującym funkcję $f(x) = \sin\frac{x}{2}$ w węzłach postaci

$$x_{nk} := \frac{1}{2}\cos\left(\frac{2k+1}{2n+2}\pi\right) + \frac{1}{2} \qquad (k = 0, 1, \ldots, n).$$

Jak należy dobrać n, aby mieć pewność, że

$$\max_{x\in[0,1]} |f(x) - L_n(x)| \leq 10^{-15} ?$$

Rozwiązanie:

Początek zadania identyczny do L15.20

$$\left|\left|f^{(n+1)}(\alpha)\right|\right| = \cos\left(\frac{\alpha}{2}\right)\cdot\left(\frac{1}{2}\right)^{(n+1)} \; lub \; \sin\left(\frac{\alpha}{2}\right)\cdot\left(\frac{1}{2}\right)^{(n+1)}$$

Mamy zakres od zera do 1, popatrzmy na

$$p_{n+1} = (x-x_0)(x-x_1)\cdot\ldots\cdot(x-x_n)$$

Można zauważyć, że max(p_n+1) <= 1, przyjmijmy tę wartość

Ograniczmy $\cos\left(\frac{\alpha}{2}\right)$ oraz $\sin\left(\frac{\alpha}{2}\right)$ wartością 1

Wtedy

$$\frac{\left(\frac{1}{2}\right)^{(n+1)}}{1\cdot(n+1)!} \leq \frac{1}{10^{15}}$$

A z tego wyciągamy już interesujące nas n

Do sprawdzenia

L15.22. Niech dane będą: liczba naturalna n i parami różne liczby rzeczywiste $a_0, a_1, \ldots, a_{n-1}$. Zaproponuj algorytm znajdowania takich liczb $c_0, c_1, \ldots, c_n$, że dla każdego $x \in \mathbb{R}$ zachodzi

$$x^n = c_0 + c_1(x - a_0) + c_2(x - a_0)(x - a_1) + \ldots + c_n(x - a_0)(x - a_1) \cdot \ldots \cdot (x - a_{n-1}).$$

Podaj jego złożoność obliczeniową i pamięciową.

Rozwiązanie:

Po prawej stronie mamy wzór interpolacyjny Newtona, stąd c_0, c_1, ..., c_n są kolejnymi ilorazami różnicowymi.

To znaczy:

$niech\ f(x) = x^n$

$$c_0 = f[a_0]$$

$$c_1 = f[a_0, a_1]$$

...

$$c_n = f[a_0, a_1, ..., a_n]$$

gdzie f[...] to iloczyn różnicowy

Algorytm:

```
def iloraz(tab_x, func, n):
  tab_iloraz = [0 for i in range(0, n+1)]
  for i in range(n+1): # 0, 1, .., n
    for j in range(n, i-1, -1): # n, n-1, .., i
      if i == 0:
        tab_iloraz[j] = func(tab_x[j])
      else:
        tab_iloraz[j] = (tab_iloraz[j] - tab_iloraz[j-1]) / (tab_x[j] - tab_x[j-i])

  return tab_iloraz
```

```
tab_x - tablica z naszymi a_i
func - funkcja x^n
```

```
Złożoność pamięciowa O(n)
Złożoność czasowa O(n^2)
```

Do sprawdzenia

L15.23. (a) Podaj definicję naturalnej funkcji sklejanej trzeciego stopnia.

(b) Znajdź naturalną funkcję sklejaną trzeciego stopnia dla danych

$$\begin{array}{c||c|c|c} x_k & -1 & 0 & 1 \\ \hline y_k & -1 & 2 & -3 \end{array}\,.$$

$$s(x) = \begin{cases} s_1 = Ax^3 + Bx^2 + Cx + D\,(x) \ : \ \{-1 \le x \le 0\} \\ s_2 = Ex^3 + Fx^2 + Gx + H\,(x) \ : \ \{0 \le x \le 1\}. \end{cases} \tag{1}$$

$$s(-1) = s_1(-1) = -1 = A - B - C + D \tag{2}$$

$$s(0) = s_1(0) = s_2(0) = 2 \ = D = H \tag{3}$$

$$s(1) = s_2(1) = -3 = E + F + G + H \tag{4}$$

Ciagłość pierwszej pochodnej

$$s'(x) = \begin{cases} s_1'(x) = 3Ax^2 + 2Bx + C \ : \ \{-1 \le x \le 0\} \\ s_2'(x) = 3Ex^2 + 2Fx + G \ : \ \{0 \le x \le 1\}. \end{cases} \tag{5}$$

$$s' - ciagla => s_1'(0) = s_2'(0) = C = G \tag{6}$$

Ciagłość drugiej pochodnej

$$s'(x) = \begin{cases} s_1''(x) = 6Ax + 2B \ : \ \{-1 \le x \le 0\} \\ s_2''(x) = 6Ex + 2F \ : \ \{0 \le x \le 1\}. \end{cases} \tag{7}$$

$$s'' - ciagla => s_1''(0) = s_2''(0) = B = F \tag{8}$$

Naturalnosć

$$s_1''(-1) = 0 = -6A + 2B \tag{9}$$

$$s_2''(1) = 0 = 6E + 2F \tag{10}$$

Teraz wystarczy stworzyć układ równań i znaleźć nasze funkcje s_1 oraz s_2

L15.24. Niech dane będą wektory $\mathtt{x}{:=}[x_0, x_1, \ldots, x_n]$ ($x_k < x_{k+1}$, $0 \le k \le n-1$), $\mathtt{y}{:=}[y_0, y_1, \ldots, y_n]$ oraz $\mathtt{z}{:=}[z_0, z_1, \ldots, z_m]$. Niech s_n oznacza naturalną funkcję sklejaną trzeciego stopnia (*w skrócie*: NFS3) spełniającą warunki $s_n(x_k) = y_k$ ($0 \le k \le n$). Jak pamiętamy, w języku `PWO++` procedura `NSpline3(x,y,z)` wyznacza wektor $\mathtt{Z} := [s_n(z_0), s_n(z_1), \ldots, s_n(z_m)]$ z tym, że **musi być** $m < 2n$. Załóżmy, że wartości pewnej funkcji ciągłej f znane są **jedynie** w punktach $x_0 < x_1 < \cdots < x_{100}$. Wiadomo, że NFS3 odpowiadająca danym $(x_k, f(x_k))$ ($0 \le k \le 100$) bardzo dobrze przybliża funkcję f. Wywołując procedurę `NSpline3` **tylko raz**, opracuj algorytm numerycznego wyznaczania przybliżonych wartości wszystkich **miejsc zerowych** funkcji f znajdujących się w przedziale $[x_0, x_{100}]$. W swoim rozwiązaniu możesz **użyć wielokrotnie** innej procedury języka `PWO++`, a mianowicie `Solve3(a,b,c,d)` znajdującej z dużą dokładnością wszystkie rzeczywiste miejsca zerowe wielomianu $\mathtt{a}x^3 + \mathtt{b}x^2 + \mathtt{c}x + \mathtt{d}$ albo informującej, że takich miejsc zerowych nie ma.

Rozwiązanie Patryka WIlusza

Mamy wektory $x = [x_0, \ldots, x_n]$ $n = 100$
$y = [y_0, \ldots, y_n]$

Oznacza to, że znamy 100 przedziałów $[x_i, x_{i+1}]$

Wiemy, że NSpline3(x, y, z) wyznacza wektor $Z = [s_n(z_0), \ldots, s_n(z_m)]$ $(m < 2n)$

Możemy więc wybrać 200 punktów z.

Wybieramy je więc tak, aby w każdym przedziale $[x_i, x_{i+1}]$ były ustawione równoodległe 2 punkty z, tak, że:

$z_{2i} = x_i + \frac{1}{3}(x_{i+1} - x_i)$

$z_{2i+1} = x_i + \frac{2}{3}(x_{i+1} - x_i)$

Teraz na każdym przedziale $[x_i, x_{i+1}]$ $i = (0, 1, \ldots, n)$ mamy 4 węzły.

Więc na każdym takim odcinku tworzymy funkcję interpolacji Newtona.

Następnie wymnażamy postać wielomianu newtona na wielomian postaci $ax^3 + bx^2 + cx + d$ i współczynniki wysyłamy do funkcji Solve3(a, b, c, d).

Do sprawdzenia

L15.25. Dana jest *postać Béziera* wielomianu $p \in \Pi_n$, tj.

$$p(t) := \sum_{k=0}^{n} a_k B_k^n(t), \qquad \text{gdzie} \qquad B_k^n(t) := \binom{n}{k} t^k (1-t)^{n-k}.$$

Uzasadnij, że

$$p(t) = \sum_{k=0}^{n+1} a_k^{(1)} B_k^{n+1}(t) \qquad \text{dla} \qquad a_k^{(1)} := \frac{n-k+1}{n+1} a_k + \frac{k}{n+1} a_{k-1} \quad (0 \le k \le n+1),$$

gdzie przyjęto $a_{-1} = a_{n+1} := 0$. Jakie zastosowanie może mieć ta zależność?

Idea

Nie jestem pewien ale, chyba trzeba z tego skorzystać

$$B_i^n(u) = \frac{n+1-i}{n+1} B_i^{n+1}(u) + \frac{i+1}{n+1} B_{i+1}^{n+1}(u) \quad (0 \le i \le n).$$

L15.26. Podaj definicję krzywej Béziera P stopnia n o punktach kontrolnych $W_0, W_1, \ldots, W_n \in \mathbb{R}^2$. Uzasadnij, że dla każdego $t \in [0, 1]$, $P(t)$ jest punktem na płaszczyźnie.

Definicja z wykładu.
P(t) jest punktem na płaszczyźnie, ponieważ B(t) sumuje się do 1, więc P(t) jest kombinacją barycentryczną punktów. - DO SPRAWDZENIA

W notatkach miałem jeszcze coś takiego, z własności krzywej Beziera, dla każdego t ∈ [0,1], P(t) ⊂ conv (W0,W1,...,Wn), gdzie conv(W0,W1,...,Wn) to otoczka wypukła dla punktów W0,...,Wn, czyli najmniejszy wielokąt zawierający te punkty. To może w jakiś sposób uzasadniać, że te punkty są punktami na płaszczyźnie. - DO SPRAWDZENIA

L15.27. Niech P będzie krzywą Béziera stopnia n o punktach kontrolnych $W_0, W_1, \ldots, W_n \in \mathbb{R}^2$. Ustalmy $t \in [0, 1]$. Zaproponuj algorytm wyznaczania $P(t)$ w czasie $O(n)$.

Schemat Hornera do opracowania algorytmu obliczania pkt. na krzywej Béziera, który działa liniowo wzgl. liczby jej pkt kontrolnych.

$\boxed{\text{Krzywa Béziera:}}$ $P(t) = \sum_{k=0}^{n} B_k^n(t) W_k$ ($t \in [0,1]$, W_k - punkty)

chcemy obliczyć wartość $P(t)$ (to punkt będący kombinacją wypukłą punktów kontrolnych)

sposób na wielomiany:

$$\frac{\binom{n}{i+1}}{\binom{n}{i}} = \frac{n!}{(i+1)!(n-i-1)!} \cdot \frac{i!(n-i)!}{n!} = \frac{n-i}{i+1}$$

$B_0^n = \binom{n}{0}(1-u)^n$

$B_1^n = \binom{n}{1}u(1-u)^{n-1}$

$B_2^n = \binom{n}{2}u^2(1-u)^{n-2}$

$\vdots$

1 mnożenie   1 mnożenie = liniowo

$$\left.\begin{array}{l} B_i^n = \binom{n}{i}u^i(1-u)^{n-i} \\ B_{i+1}^n = \binom{n}{i+1}u^{i+1}(1-u)^{n-i-1} \end{array}\right) \cdot \frac{u}{1-u} \cdot \frac{\binom{n}{i+1}}{\binom{n}{i}} = \boxed{\frac{u}{1-u} \cdot \frac{n-i}{i+1}}$$

$\vdots$

$B_n^n = \binom{n}{n}u^n$

$P(t) = B_0^n W_0 + B_1^n W_1 + \ldots + B_n^n W_n =$

$= W_0 B_0^n + \frac{u}{1-u}(W_1 B_0^n \cdot n + \frac{u}{1-u}(W_2 B_0^n \cdot n \cdot \frac{n-1}{2} + \ldots \frac{u}{1-u}(W_{n-1} B_{n-1}^n \cdot n \cdot \frac{n-1}{2} \cdot \ldots \cdot \frac{2}{n-1} + \frac{u}{1-u} W_n B_n^n \cdot (*))))$

$(*) = n \cdot \frac{n-1}{2} \cdot \ldots \cdot \frac{2}{n-1} \cdot \frac{1}{n}$

Dane: n -liczba punktów kontrolnych
tab - tablica wartości punktów kontrolnych

Algorytm:

1. Wykonaj podstawienia:
   B = (1 - u)^n # używam tutaj algorytmu szybkiego potęgowania, który działa w czasie O(log n), co nie zaburza ogólnej złożoności O(n), gdyż log n < n
   wynik = B * tab [0]
   C = u / (1 - u)
2. Dla i = 1, 2, ..., n wykonuj:
   B = B * C * ((n - i) / (i + 1))
   wynik = wynik + tab[i] * B
3. Zwróć: wynik

L15.28. Niech p będzie wielomianem zmiennej t stopnia co najwyżej n. W języku `PWO++` procedura `BezierCoeffs(`p, t`)` wyznacza taki wektor `c`$:= [c_0, c_1, \ldots, c_n]$, że

$$p(t) = \sum_{k=0}^{n} c_k B_k^n(t),$$

gdzie $B_0^n, B_1^n, \ldots, B_n^n$ są wielomianami Bernsteina stopnia n. Współczynniki c_k $(0 \leq k \leq n)$ nazywamy *współczynnikami Béziera* wielomianu p. Niestety, procedura ta ma **pewne ograniczenie**, mianowicie: **musi być** $n \leq 50$.

W jaki sposób, używając procedury `BezierCoeffs` co najwyżej **dwa razy**, wyznaczyć współczynniki Béziera wielomianu $w(t) := p(t) \cdot q(t)$, gdzie $p \in \Pi_{50}$, a $q \in \Pi_2$? Jak zmieni się rozwiązanie, jeśli przyjąć, że $q \in \Pi_{50}$?

L15.29. Pomiary (t_k, c_k) $(0 \le k \le N;\ t_k > 0,\ c_k > 1)$ pewnej zależnej od czasu wielkości fizycznej C sugerują, że wyraża się ona wzorem

$$C(t) = 2^{(At^2+2018)^{-1}}.$$

Stosując aproksymację średniokwadratową, wyznacz prawdopodobną wartość parametru A.

Rozwiązanie:

$$C(t) = 2^{\frac{1}{At^2+2018}}$$

$$\log_2 C(t) = \frac{1}{At^2+2018}$$

$$\left[\log_2 C(t)\right]^{-1} = At^2 + 2018$$

$$\left[\log_2 C(t)\right]^{-1} - 2018 = At^2$$

Stąd bazą jest t^2

Dostajemy

$$\left[< t^2, t^2 > \right][A] = \left[< t^2, \left[\log_2 C(t)\right]^{-1} - 2018 \right]$$

Teraz wystarczy policzyć iloczyny skalarne i dostaniemy wartość A

L15.30. Wyznacz funkcję postaci $y(x) = \frac{ax^2 - 3}{x^2 + 1}$ najlepiej dopasowaną w sensie aproksymacji średniokwadratowej do danych

$$\begin{array}{c||c|c|c|c} x_k & x_0 & x_1 & \dots & x_n \\ \hline y_k & y_0 & y_1 & \dots & y_n \end{array},$$

przy założeniu, że $s_2 = 10$, $s_4 = -3$, gdzie $s_m := \sum_{k=0}^{n} \frac{x_k^m}{(x_k^2+1)^2}$ $(m = 2, 4)$.

Rozwiązanie:

$$Szukamy\ w^* \in F : ||f - w^*||_2 = \min_{a \in R} ||f - a||_2 = \min_{a \in R} \sqrt{\sum_{k=0}^{n} (y(x_k) - a)^2}$$

Weźmy funkcję błędu a

$$E(a) = \sum_{k=0}^{n} \left(y(x_k) - \frac{ax_k^2 - 3}{x_k^2 + 1} \right)^2$$

Obliczmy jej pochodną:

$$E'(a) = -2 \sum_{k=0}^{n} \left(y(x_k) - \frac{ax_k^2 - 3}{x_k^2 + 1} \right) \frac{x_k^2}{x_k^2 + 1} = 0$$

Podzielmy obustronnie przez -2 oraz rozdzielmy sumy:

$$E'(a) = \sum_{k=0}^{n} \frac{y_k x_k^2}{x_k^2 + 1} - a \sum_{k=0}^{n} \frac{ax_k^4}{(x_k^2 + 1)^2} + \sum_{k=0}^{n} \frac{3x_k^2}{(x_k^2 + 1)^2} = 0$$

Niech

$$A = y_k \sum_{k=0}^{n} \frac{x_k^2}{x_k^2 + 1}, B = \sum_{k=0}^{n} \frac{x_k^4}{(x_k^2 + 1)^2}, C = 3 \sum_{k=0}^{n} \frac{x_k^2}{(x_k^2 + 1)^2}$$

Możemy zauważyć, że

$B = s_4, C = 3s_2$

Wtedy

$$a = \frac{A - 30}{-3} = 10 - \frac{A}{3}$$

Do sprawdzenia

L15.31. (a) Znajdź wielomiany P_0, P_1, P_2 ortogonalne względem iloczynu skalarnego

$$(f,g) := f(-2)g(-2) + f(-1)g(-1) + f(0)g(0) + f(1)g(1) + f(2)g(2).$$

(b) Wykorzystując wynik otrzymany w punkcie (a), wyznacz wielomian $w_2^* \in \Pi_2$ najlepiej dopasowany w sensie aproksymacji średniokwadratowej do danych

x_k	-2	-1	0	1	2
y_k	4	1	1	1	4

Rozwiązanie:

a)

Mamy

x_k	-2	-1	0	1	2

Wielomiany P_0, P_1, P_2 konstruujemy następującą zależnością rekurencyjną

$$P_0(x) = 1$$

$$P_1(x) = x - \frac{<xP_0, P_0>}{<P_0, P_0>}$$

$$P_n(x) = \left(x - \frac{<xP_{n-1}, P_{n-1}>}{<P_{n-1}, P_{n-1}>}\right)P_{n-1}(x) - \frac{<P_{n-1}, P_{n-1}>}{<P_{n-2}, P_{n-2}>}P_{n-2}(x) \qquad dla\ n > 1$$

Stąd

$$P_0(x) = 1$$

$$P_1(x) = x$$

$$P_2(x) = x^2 - 2$$

b) Tutaj wystarczy podstawić wartości z zadania pod wzórki

$$w_2^* = \sum_{k=0}^{n} a_k P_k(x)$$

$$a_k = \frac{<f, P_k>}{<P_k, P_k>}$$

a0 = 11/5

a1 = 0

a2 = 6/7

wm = 11/5 + 6/7(x^2-2)

Do sprawdzenia

L15.32. Niech $P_0, P_1, \ldots, P_N$ będą wielomianami ortogonalnymi względem iloczynu skalarnego postaci

$$(f, g)_N := \sum_{k=0}^{N} f(x_k) g(x_k),$$

gdzie $x_k := -a + \frac{2ak}{N}$ $(k = 0, 1, \ldots, N;\ a > 0)$. Udowodnij, że jeśli α jest miejscem zerowym wielomianu P_k $(0 \leq k \leq N)$, to także $-\alpha$ jest miejscem zerowym tego wielomianu.

Wskazówka: x_0 + x_1 + x_2 + ... + x_N = 0

Do sprawdzenia.

L15.33. Podaj definicję ciągu wielomianów ortogonalnych względem dyskretnego iloczynu skalarnego $(\cdot, \cdot)_N$. Jak efektywnie wyznaczać takie wielomiany? Jakie jest ich zastosowanie w aproksymacji średniokwadratowej na zbiorze dyskretnym?

L15.34. Podaj definicję rzędu kwadratury liniowej $Q_n(f) := \sum_{k=0}^{n} A_k^{(n)} f(x_k^{(n)})$. Udowodnij, że jeśli rząd kwadratury Q_n wynosi przynajmniej $n+1$, to jest to kwadratura liniowa.

L15.35. Jaki maksymalnie rząd może mieć kwadratura liniowa? Odpowiedź uzasadnij.

L15.36. Opisz ideę kwadratur złożonych. Wyprowadź złożony wzór Simpsona.

http://wazniak.mimuw.edu.pl/index.php?title=MN14#Kwadratury_z.C5.82o.C5.BCone

L15.37. Opisz metodę Romberga obliczania przybliżonej wartości całki $\int_{-2}^{3} f(x)\,\mathrm{d}x$.

Rozwiązanie z wiki:

Niech dany będzie zbiór $a = x_0, x_1, \cdots, x_{2^i} = b$ dzielących przedział (a, b) na 2^i równych części taki, że znane są wartości funkcji $f(x_i) = y_i$

Niech $h_i = \dfrac{b-a}{2^i}$, oznacza długość kroku.

Metodę Romberga można opisać rekurencyjnie:

$$\begin{cases} R_{0,i} & : R_{2^i} = h_i \cdot \sum_{k=0}^{2^i-1} \left(\frac{f(x_k)+f(x_{k+1})}{2}\right) \\ R_{m,i} & : \frac{4^m \cdot R_{m-1,i+1} - R_{m-1,i}}{4^m - 1} \end{cases}$$

Nie wiem co więcej można o tym powiedzieć, ponieważ mamy podane jedynie $a = -2$ oraz $b = 3$

Do sprawdzenia

L15.38. Opisz kwadratury złożone. Jaką mają one przewagę nad kwadraturami Newtona-Cotesa? Czy są one związane z metodą Romberga? Jeśli tak, to w jaki sposób?

Rozwiązanie:

a)Opisz kwadratury złożone

Idea kwadratur złożonych polega na podzieleniu przedziału całkowania na równoodległe podprzedziały. Następnie w każdym z podprzedziałów wykonujemy kwadraturę prostą tzn. np. Simpsona lub wzór Trapezów. Ostatnim krokiem jest zsumowanie wyników otrzymanych z 'całkowania' każdego z podprzedziałów .

b)Szczerze nie wiem, ale pewnie mają o wiele wyższy rząd niż Q^{NC}

c)Tak, Metoda Romberga opiera się na złożonym wzorze trapezów

d)

$$R_{0,i} = T_{2^i} = h_i \sum_{k=0}^{2^i-1} \frac{f(x_k) + f(x_{k+1})}{2}$$

Do sprawdzenia

L15.39. Znajdź rozkład LU macierzy $A := \begin{bmatrix} 1 & 2 & -1 & 2 \\ -2 & -5 & 3 & -4 \\ 4 & 12 & -10 & 9 \\ -8 & -24 & 32 & -16 \end{bmatrix}$. Następnie wykorzystaj otrzymany rozkład do rozwiązania układu równań $Ax = b$, gdzie $b := [17, -33, 70, -112]^T$

Rozwiązanie Mateusza Kacały:

L15.39

Znajdź rozkład LU macierzy A. Otrzymany rozkład wykorzystaj do rozwiązania układu równań $Ax=b$.

$$A = \begin{bmatrix} 1 & 2 & -1 & 2 \\ -2 & -5 & 3 & -4 \\ 4 & 12 & -10 & 9 \\ -8 & -24 & 32 & -16 \end{bmatrix} \qquad b = \begin{bmatrix} 17 \\ -33 \\ 70 \\ -112 \end{bmatrix}$$

Zadanie to można rozwiązać metodą faktoryzacji.

1) Najpierw znajdujemy rozkład LU macierzy.

$$U = \begin{bmatrix} G & H & I & J \\ 0 & K & L & M \\ 0 & 0 & N & O \\ 0 & 0 & 0 & P \end{bmatrix}$$

Rozkład LU znajdujemy rozwiązując układy równań wynikające z mnożenia macierzy LU.

Niewiadome odnajdujemy w kolejności:

1) wiersz macierzy U (od lewej)
2) kolumna macierzy L (od góry)
3) wróć do 1)

$$L = \begin{bmatrix} 1 & 0 & 0 & 0 \\ A & 1 & 0 & 0 \\ B & C & 1 & 0 \\ D & E & F & 1 \end{bmatrix} \quad \begin{bmatrix} 1 & 2 & -1 & 2 \\ -2 & -5 & 3 & -4 \\ 4 & 12 & -10 & 9 \\ -8 & -24 & 32 & -16 \end{bmatrix} = A$$

$G \cdot 1 = 1 \Rightarrow G = 1$
$H \cdot 1 = 2 \Rightarrow H = 2$
$I \cdot 1 = -1 \Rightarrow I = -1$
$J \cdot 1 = 2 \Rightarrow J = 2$

$A \cdot G = -2 \Rightarrow A = -2$
$B \cdot G = 4 \Rightarrow B = 4$
$D \cdot G = -8 \Rightarrow D = -8$

$\overset{(2)}{H} \cdot \overset{(-2)}{A} + K = -5 \Rightarrow K = -1$
$\overset{(-1)}{I} \cdot \overset{(-2)}{A} + L = 3 \Rightarrow L = 0$
$\overset{(2)}{J} \cdot \overset{(-2)}{A} + M = -4 \Rightarrow M = 0$

$\overset{(4)}{B} \cdot \overset{(2)}{H} + C \cdot \overset{(-1)}{K} = 12 \Rightarrow C = -4$
$\overset{(-8)}{D} \cdot \overset{(2)}{H} + E \cdot \overset{(-1)}{K} = -24 \Rightarrow E = 8$

$\overset{(4)}{B} \cdot \overset{(-1)}{I} + C \cdot \overset{(0)}{L} + N = -10 \Rightarrow N = -6$
$\overset{(4)}{B} \cdot \overset{(2)}{J} + C \cdot \overset{(0)}{M} + O = 9 \Rightarrow O = 1$

$\overset{(-8)(-1)}{D \cdot I} + E \cdot \overset{(0)}{L} + F \cdot \overset{(-6)}{N} = 32 \Rightarrow F = -4$
$\overset{(-8)(2)}{D \cdot J} + E \cdot \overset{(0)}{M} + \overset{(-4)(1)}{F \cdot O} + P = -16 \Rightarrow P = 4$

Stąd:
$$L = \begin{bmatrix} 1 & 0 & 0 & 0 \\ -2 & 1 & 0 & 0 \\ 4 & -4 & 1 & 0 \\ -8 & 8 & -4 & 1 \end{bmatrix}, \quad U = \begin{bmatrix} 1 & 2 & -1 & 2 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & -6 & 1 \\ 0 & 0 & 0 & 4 \end{bmatrix}$$

2) Gdy mamy już rozkład LU musimy rozwiązać dwa prostsze równania $Lz=b$ oraz $Ux=z$. Wektor x będzie tutaj rozwiązaniem naszego pierwotnego układu $Ax=b$

Mateusz Kucała

L15.39 Ciąg dalszy zadania

$L \cdot z = b$

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ -2 & 1 & 0 & 0 \\ 4 & -4 & 1 & 0 \\ -8 & 8 & -4 & 1 \end{bmatrix} \cdot \begin{bmatrix} z_1 \\ z_2 \\ z_3 \\ z_4 \end{bmatrix} = \begin{bmatrix} 17 \\ -33 \\ 70 \\ -112 \end{bmatrix}$$

$z_1 = 17$

$-2 \cdot 17 + z_2 = -33 \Rightarrow z_2 = 1$

$4 \cdot 17 + (-4) \cdot 1 + z_3 = 70 \Rightarrow z_3 = 6$

$-8 \cdot 17 + 8 \cdot 1 + (-4) \cdot 6 + z_4 = -112 \Rightarrow z_4 = 40$

$$z = \begin{bmatrix} 17 \\ 1 \\ 6 \\ 40 \end{bmatrix}$$

$U \cdot x = z$

$$\begin{bmatrix} 1 & 2 & -1 & 2 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & -6 & 1 \\ 0 & 0 & 0 & 4 \end{bmatrix} \cdot \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} = \begin{bmatrix} 17 \\ 1 \\ 6 \\ 40 \end{bmatrix}$$

$x_4 = 10$

$10 + (-6) \cdot x_3 = 6 \Rightarrow x_3 = \frac{4}{6}$

$x_2 = -1$

$x_1 + \overset{-2}{2 \cdot (-1)} + \overset{-\frac{4}{6}}{(-1) \cdot \left(\frac{4}{6}\right)} + \overset{20}{2 \cdot 10} = 17 \Rightarrow x_1 = -\frac{1}{3}$

Zatem ostateczny wynik to: $x = \begin{bmatrix} -1/3 \\ -1 \\ 4/6 \\ 10 \end{bmatrix}$ ■

Podobne zadania z rozkładem LU:

- Szukanie macierzy odwrotnej

  korzystamy z $A^{-1} = L^{-1} \cdot U^{-1}$, a odwrotności L oraz U można szybko wyliczyć metodą Gaussa-Jordana

- Szukanie wyznacznika macierzy

  $\det(A) = \det(L \cdot U) = \det(L) \cdot \det(U)$

L15.40. Niech dana będzie macierz $A \in \mathbb{R}^{n\times m}$. Przypomnijmy, że *rzędem* macierzy nazywamy maksymalną liczbę jej liniowo niezależnych kolumn. Opracuj algorytm numerycznego wyznaczania rzędu macierzy A. Podaj jego złożoność czasową i pamięciową.

L15.41. Niech dana będzie macierz nieosobliwa $A \in \mathbb{R}^{n \times n}$. Zaproponuj efektywny algorytm wyznaczania macierzy odwrotnej A^{-1} i podaj jego złożoność.

L15.42. Niech dane będą macierze $A, B \in \mathbb{R}^{n\times n}$. Opracuj oszczędny algorytm wyznaczania takiej macierzy $X \in \mathbb{R}^{n\times n}$, aby zachodziła równość $AX = B$. Podaj jego złożoność czasową i pamięciową.

L15.43. Opracuj metodę wyznaczania rozkładu LU macierzy $A_n \in \mathbb{R}^{n\times n}$ postaci

$$A_n := \begin{bmatrix} a_1 & & & & & c_1 \\ & a_2 & & & & c_2 \\ & & a_3 & & & c_3 \\ & & & \ddots & & \vdots \\ & & & & a_{n-1} & c_{n-1} \\ b_1 & b_2 & b_3 & \cdots & b_{n-1} & a_n \end{bmatrix},$$

gdzie zaznaczono jedynie niezerowe elementy. Podaj jej złożoność.