

## Ramowy program laboratorium z metod numerycznych.

### Skrócone instrukcje do ćwiczeń laboratoryjnych.

Termin	Nr	Tematyka
1		Wprowadzenie, zasady zaliczenia, regulamin, BHP itp.
2	Ćw. 1	Błędy. Liczby zmiennoprzecinkowe IEEE 754. Epsilon maszynowy
3	Ćw. 2	Rozwiązywanie układu równań liniowych metodą eliminacji Gaussa
4	Ćw. 3	Rozwiązywanie układu równań liniowych metodą Gaussa-Seidla
5	Ćw. 4	Rozwiązywanie równań nieliniowych metodą iteracji prostej. Metoda wielokrokowa z korekcją Aitkena.
6	Ćw. 5	Rozwiązywanie równań nieliniowych metodą Newtona i metodą siecznych
7	Ćw. 6	Rozwiązywanie układów równań nieliniowych metoda Newtona-Raphsona
8	Ćw. 7	Interpolacja funkcji dyskretnej
9	Ćw. 8	Aproksymacja funkcji metodą najmniejszych kwadratów I
10	Ćw. 9	Aproksymacja funkcji metodą najmniejszych kwadratów z wykorzystaniem rozkładu macierzy według wartości szczególnych (SVD)
11	Ćw. 10	Aproksymacja metodą najmniejszych kwadratów II do wygładzania oraz estymacji parametrów sygnałów wejściowych
12	Ćw. 11	Algorytmy całkowania numerycznego
13	Ćw. 12	Rozwiązywanie równań różniczkowych – metoda prostokątów i metoda trapezów
14	Ćw. 13	Rozwiązywanie równań różniczkowych – metoda Runge-Kutty IV rzędu
15		Zaliczenie. Termin rezerwowy

Wrocław, 22.02.2016 r.

## Ćwiczenie 1.

### Błędy. Liczby zmiennoprzecinkowe IEEE 754. Epsilon maszynowy.

Program ćwiczenia:

#### Błędy zaokrągleń:

1. Sprawdzić wynik działań:

a = 5;

b = 3;

c = 4;

d = a / b ???

e = a / c ???

Dla zmiennych zadeklarowanych jako liczby zmiennoprzecinkowe wszystko wydaje się być ok. Jak się zmienią wartości d i e jeśli zmienne a, b i c zostaną zadeklarowane jako typu całkowitego ze znakiem?

```
a=int16(5);
```

```
b=int16(3);
```

```
c=int16(4);
```

2. Sprawdzić wynik działania polegającym na 10000-krotnym dodawaniu do pewnej liczby, np. zera, wartości  $a = 1/3$ ; . Dodawanie należy wykonać w pętli (np. typu FOR).

Należy sprawdzić wynik dodawania w przypadku gdy zmienna a jest:

a) typu **double** (typ podwójnej precyzji domyślnie używany przez MATLABa, gdzie wartość przechowywanej liczby jest kodowana zgodnie ze standardem IEEE 754 z wykorzystaniem 64 bitów; liczby rzeczywiste zmiennoprzecinkowe)

b) typu **single** (typ pojedynczej precyzji, odpowiednik typu float znanego w języku C, gdzie wartość przechowywanej liczby zgodnie ze standardem IEEE 754 jest kodowana z wykorzystaniem 32 bitów; liczby rzeczywiste zmiennoprzecinkowe)

Aby zadeklarować taką zmienną w MATLABie należy użyć funkcji `single`:

```
a=single(1/3);
```

Ponadto sprawdzić wynik działania: `suma = a *10000;`

Następnie porównać uzyskane wyniki z wartością spodziewaną sumy. Na którym miejscu po przecinku znajduje się błąd. Skąd się wzięły różnice? Co się będzie działo z błędem gdy liczbę iteracji zwiększymy z 10000 na 100000?

Aby zwiększyć liczbę wyświetlanych na ekranie miejsc wyniku można użyć komendy:

```
format long lub format long g.
```

3. Sprawdzić wynik działania `wynik = 10*(1.1-1)-1`. Działanie wykonać z wykorzystaniem MATLABa oraz aplikacji kalkulatora wbudowanego w system operacyjny Windows, oraz najprostszy kalkulator sprzętowy? Który z tych „kalkulatorów” jest lepszy? (pamiętajmy przy tym o kosztach użytych narzędzi: Matlab to koszt ok. 100tys zł, Windows ok. 400zł, najprostszy kalkulator ok. 4zł).

Sprawdzić wynik operacji:  $1 - 3*(4/3-1) = ?$

4. Porównać ze sobą wartości poniższych wyrażeń (zachować postać wyrażeń):

- a.  $(7.1 - 7) - 0.1$
- b.  $7.1 - (7 + 0.1)$

5. Sprawdzić wartość stałej  $\pi$ , oraz wartość funkcji  $\sin(\pi)$ :

```
pi
wynik = sin(pi)
```

Czy wynik jest równy 0? Jaki jest błąd?

6. Sprawdzić wynik działania:

```
sqrt(1e-16 + 1) - 1 = ?
```

7. Czy 0,1 to prosta do zapamiętania przez komputer liczba rzeczywista? Sprawdzić wynik działania programu:

```
a = 0.0;
for i = 1:10
    a = a + 0.1;
end

if (a==1) disp ('wynik poprawny a=1'); else disp ('a jest różne
od 1, czyli 0.1 to bardzo niewygodna dla komputera liczba!!!!'); end;
```

8. Ile to jest równe?

Zmiennej a przypisać wartość 32767 a następnie dodać do niej 1, tj. wynik = a + 1. Jaki jest wynik? (zmienną a należy zadeklarować jako całkowitą, tj. a=int16(32767).)

Zmiennej a przypisać wartość -32767 a następnie odjąć od niej -10, tj. wynik = a - 10.

Jaki jest wynik? (zmienną a należy zadeklarować jako całkowitą, tj. a=int16(-32767).)

Sprawdzić wartości następujących wyrażeń:

```
intmin('int8')           intmax('int8')
intmin('int16')          intmax('int16')
intmin('int32')          intmax('int32')
intmin('int64')          intmax('int64')
```

```
intmin('uint8')           intmax('uint8')
intmin('uint16')          intmax('uint16')
intmin('uint32')          intmax('uint32')
intmin('uint64')          intmax('uint64')
```

```
help intmin
help intmax
```

```
realmin('single')         realmax('single')
realmin('double')         realmax('double')
```

```
help realmin
help realmax
```

### Dokładność obliczeń:

9. Sprawdzić wynik działania:

$$(2^{53} + 1) - 2^{53} = ?$$

10. Narysować wykres funkcji  $f_1(x) = x^4 - 4x^3 + 6x^2 - 4x + 1$  w przedziale  $x \in \langle 0.9996, 1.0004 \rangle$  i na tej podstawie wyznaczyć jej miejsca zerowe. Skonfrontować uzyskane wyniki z działaniem polecenia `roots`.

Przeprowadzić raz jeszcze rozważania jak wyżej ale tym razem dla funkcji:

$$f_2(x) = (x - 1)^4$$

Porównać i przedyskutować ze sobą funkcje  $f_1(x)$  i  $f_2(x)$  oraz osiągnięte dla nich wyniki.

### Epsilon maszynowy:

**Epsilon maszynowy** jest wartością określającą precyzję obliczeń numerycznych wykonywanych na liczbach zmiennoprzecinkowych.

Często jest definiowana jako najmniejsza liczba nieujemna, której dodanie do jedności daje wynik nierówny 1. Innymi słowy, jest to najmniejszy  $\varepsilon$ , dla którego następujący warunek jest uznawany za niespełniony (przyjmuje wartość *falsz*):  $1 + \varepsilon = 1$ . Taka definicja, choć powszechnie przyjęta jest jednak prawdziwa tylko dla liczby  $1.0$ , gdyż dla liczb większych wartość epsilon jest zwykle większa, a dla mniejszych odpowiednio również się zmniejsza.

Im mniejsza wartość epsilon maszynowego, tym większa jest względna precyzja obliczeń. Wartości tej jednak nie należy mylić ze (zwykle dużo niższą) najmniejszą liczbą uznawaną przez maszynę za różną od zera.

Epsilon maszynowy jest zależny od implementacji. Precyzja obliczeń jest ograniczona od strony sprzętowej przez rozmiar rejestrów, na których są wykonywane obliczenia, a od strony programowej przez typ danych użyty do reprezentowania liczb zmiennoprzecinkowych. Epsilon wyznaczony dla danej liczby można również traktować jako miarę odległości od tej liczby do najbliższej innej liczby, która jest w dokładny sposób reprezentowana w danej jest również

### 11. Zadanie:

Napisać program służący do wyznaczania wartości epsilon maszynowego dla liczb zmiennoprzecinkowych pojedynczej i podwójnej precyzji (**single** i **double**).

W tym celu można skorzystać z następującego algorytmu:

1. przyjąć początkową wartość  $\varepsilon = a$ , gdzie:  $a$  – liczba dla której chcemy wyznaczyć wartość epsilon.
2. sprawdzić czy spełniony jest warunek:  $a + \varepsilon = a$
3. jeśli warunek jest spełniony tzn. że znaleziono wartość epsilon. Jeśli warunek nie jest spełniony to podzielić epsilon przez 2 ( $\varepsilon = \varepsilon / 2$ ) i wróć do punktu 2.

Porównać ze sobą wyniki uzyskane dla liczb:  $1, 10^{-6}, 10^{-14}, 1, 10^6, 10^{14}$  reprezentowanych jako liczby zmiennoprzecinkowe pojedynczej i podwójnej precyzji.

Korzystając z MATLABa i funkcji **eps**, wyznaczyć wartości epsilon maszynowego dla liczb i typów jak wyżej i porównać je z wynikami uzyskanymi samodzielnie.

## Ćwiczenie 2.

### Metoda eliminacji Gaussa rozwiązywania układu równań liniowych

#### Zadanie:

- a) Korzystając z procedury GAUSS.M napisać program do rozwiązywania układów równań liniowych.

*Wskazówka:* napisać własny program, w którym będzie odwołanie do procedury GAUSS:

$$x = \text{gauss}(A, b);$$

gdzie:  $A$  - macierz równania ( $n \times n$ ),  $b$  - wektor prawej strony ( $n \times 1$ ),  $x$  - wektor rozwiązania ( $n \times 1$ ).

Zastosować tę procedurę do rozwiązywania układu równań  $Ax = b$  podanego przez prowadzącego.

- b) Korzystając z trójkątnej macierzy uzyskanej po pierwszym etapie procedury eliminacji Gaussa opracować procedurę do obliczania wyznacznika macierzy. Zastosować tę procedurę do obliczenia wyznacznika macierzy  $A$  (skorzystać z przykładowego programu EL\_GAUSSA.M).

- c) Korzystając z  $n$  rozwiązań równania algebraicznego z macierzą ( $n \times n$ ) opracować program do odwracania macierzy kwadratowej dowolnego stopnia. W kolejnych  $i = 1, \dots, n$  rozwiązaniach przyjąć w charakterze wektora prawej strony wektor zerowy z jedną tylko jedynką leżącą na pozycji  $i$ :

$$b_i = \begin{bmatrix} 0 & \dots & 1 & 0 & \dots & 0 \\ 1 & \dots & i & i+1 & \dots & n \end{bmatrix}^T$$

Zastosować tę procedurę do macierzy  $A$ ; wynik porównać z rezultatem uzyskanym na podstawie procedury MATLABa  $\text{inv}(A)$ .

### Ćwiczenie 3.

#### Metoda Gaussa-Seidla iteracyjnego rozwiązywania układów równań liniowych

Rozpatrywany jest następujący układ równań

$$\begin{aligned} a_{11}x_1 + a_{12}x_2 + a_{13}x_3 + \dots + a_{1n}x_n &= b_1 \\ a_{21}x_1 + a_{22}x_2 + a_{23}x_3 + \dots + a_{2n}x_n &= b_2 \\ \dots & \dots \dots \dots \dots \dots \dots \\ a_{n1}x_1 + a_{n2}x_2 + a_{n3}x_3 + \dots + a_{nn}x_n &= b_n \end{aligned} \quad (1)$$

Metoda Gaussa-Seidla iteracyjnego rozwiązywania układu równań (1) przybiera formę następującego algorytmu.

1. Uporządkować wyjściowy układ  $n$  równań tak, aby w macierzy współczynników  $A$  największe co do modułu elementy znalazły się na przekątnej, co jest określone następującym warunkiem

$$|a_{ii}| > |a_{ij}|_{i \neq j}, \quad i=1,2,\dots,n, \quad j=1,2,\dots,n$$

2. Przyjąć warunki początkowe

$$\{x\}_0 = \{x_1^0 \quad x_2^0 \quad \dots \quad x_n^0\}$$

3. Powtarzać proces iteracyjny

$$\begin{aligned} x_1^{k+1} &= -a_{12}x_2^k/a_{11} - a_{13}x_3^k/a_{11} - \dots - a_{1n}x_n^k/a_{11} + b_1/a_{11} \\ x_2^{k+1} &= -a_{21}x_1^{k+1}/a_{22} - a_{23}x_3^k/a_{22} - \dots - a_{2n}x_n^k/a_{22} + b_2/a_{22} \\ \dots & \dots \dots \dots \dots \dots \dots \end{aligned} \quad (2)$$

$$x_n^{k+1} = -a_{n1}x_1^{k+1}/a_{nn} - a_{n2}x_2^{k+1}/a_{nn} - \dots - a_{n,n-1}x_{n-1}^{k+1}/a_{nn} + b_n/a_{nn}$$

co może być zapisane w następującej ogólnej postaci

$$x_i^{k+1} = \frac{b_i}{a_{ii}} - \sum_{j=1}^{i-1} \frac{a_{ij}}{a_{ii}} x_j^{k+1} - \sum_{j=i+1}^n \frac{a_{ij}}{a_{ii}} x_j^k \quad (3)$$

dla kolejnych kroków iteracji  $k=1,2,\dots$  aż spełniony zostanie warunek

$$\max_{i=1,2,\dots,n} |x_i^{k+1} - x_i^k| < \varepsilon$$

gdzie  $\varepsilon$  - założona dokładność obliczeń.

#### Zadanie:

Stosując tę metodę rozwiązać układ równań podany przez prowadzącego.

Uwaga: napisać program, który automatycznie porządkuje macierz współczynników (krok 1) oraz określa rozwiązanie (krok 3) dla ogólnego przypadku dowolnego zadania.

#### Ćwiczenie 4.

#### Metoda prostej iteracji rozwiązywania równań nieliniowych z korekcją Aitkena.

Rozpatrzmy równanie nieliniowe o postaci

$$f(x) = 0 \quad (1)$$

Metoda prostej iteracji poszukiwania wartości  $x$ , dla której spełnione jest równanie (1) polega na przekształceniu go do postaci

$$x = g(x) \quad (2)$$

dla której można sformułować następującą regułę iteracyjną

$$x^{k+1} = g(x^k) \quad (3)$$

z warunkami początkowymi:  $x^0 = x_0$ .

Algorytm (3) prowadzi do rozwiązania, gdy proces iteracyjny jest zbieżny. Zbieżność jest zapewniona, gdy spełniony jest następujący warunek. Dla dowolnie wybranej zmiennej  $\xi$  zachodzi nierówność

$$|g(x) - g(\xi)| \leq K|x - \xi| \quad (4)$$

gdzie  $K < 1$ .

Warunek ten daleko nie zawsze jest spełniony, co dyskwalifikuje całą metodę, jako sposób znajdowania miejsc zerowych funkcji (1). Aby rozszerzyć obszar zbieżności i przyspieszyć zbieżność procesu iteracyjnego można stosować jego korekcję według metody Aitkena. Korekcja jest dokonywana na podstawie trzech kolejnych wartości  $x(k)$ ,  $x(k+1)$ ,  $x(k+2)$  przybliżenia, uzyskanych według metody prostej iteracji według następującej reguły

$$x_p^{k+1} = x^k - \frac{(x^{k+1} - x^k)^2}{x^{k+2} - 2x^{k+1} + x^k} \quad (5)$$

Wynik tej korekcji przyjmuje się w charakterze kolejnego przybliżenia rozwiązania:  $x^{k+1} = x_p^{k+1}$ , po czym następują znów dwa kroki procedury (3) do kolejnej korekcji (5). W ten sposób uzyskuje się algorytm o następującej postaci.

4. Przyjąć warunki początkowe  
 $x^0 = x_0$ ,  $k = 0$  - numer kroku iteracji
5. Wykonać dwa kroki prostej iteracji  
 $y^k = g(x^k)$   
 $z^k = g(y^k)$
6. Skorygować wynik:  
 $\Delta^k = \frac{(y^k - x^k)^2}{z^k - 2y^k + x^k}$   
 $x^{k+1} = x^k - \Delta^k$
4. Jeśli  $abs(\Delta^k) > eps$ ,  $k = k + 1$ , przejdź do 2

#### Zadanie:

Stosując tę metodę znaleźć miejsca zerowe funkcji podanej przez prowadzącego. Przedstawić graficznie przebieg funkcji w pobliżu rozwiązania (rozwiązań).

## Ćwiczenie 5.

### Rozwiązywanie równań nieliniowych metodą Newtona i metodą siecznych

#### Zadanie:

Znaleźć wszystkie miejsca zerowe funkcji jak w zadaniu 4. (metoda iteracji prostej iteracji z korekcją Aitkena) na podstawie:

- algorytmu Newtona,
- metody siecznych.

Przedstawić graficznie przebieg funkcji w pobliżu rozwiązania (rozwiązań).

Przeprowadzić analizę porównawczą wszystkich (tzn. 4) metod badanych w zad. 3, 4 i 5 pod kątem:

- poprawności znajdowania rozwiązań,
- wpływu warunków początkowych na wynik rozwiązania,
- nakładów obliczeniowych (czas obliczeń – ilość iteracji),
- wpływu przyjętych różnych wartości parametru  $eps$ , na ww.
- inne.



## **Ćwiczenie 6.**

### **Metoda Newtona-Raphsona rozwiązywania układów równań**

#### Zadanie:

Stosując metodę Newtona-Raphsona wyznaczyć rozwiązanie podanego przez prowadzącego układu równań nieliniowych.

Wynik zilustrować graficznie.

Przeanalizować wpływ warunków początkowych na wynik rozwiązania.

## Ćwiczenie 7. Interpolacja funkcji dyskretnej

Interpolacja polega na przedstawieniu funkcji w postaci ciągłej, gdy znana jest ona w postaci dyskretnej. Jest to zatem zdanie odwrotne do dyskretyzacji lub próbkowania wielkości ciągłej.

Załóżmy, że dla danego zbioru zmiennych niezależnych z przedziału  $\langle a; b \rangle$ :  $x_1, x_2, \dots, x_{n+1}$  znane są przyporządkowane im wartości funkcji:  $y_1, y_2, \dots, y_{n+1}$ . Zadaniem interpolacji jest wyznaczenie przybliżonych wartości funkcji dla wartości zmiennych niezależnych z przedziału  $\langle a; b \rangle$ , lecz nie będących punktami ze zbioru  $x_1, x_2, \dots, x_{n+1}$  (punkty te tworzą tzw. węzły interpolacji).

Typowymi przykładami zastosowania interpolacji jest obliczanie całek oraz pochodnych funkcji dyskretnych w czasie lub określenie wartości funkcji danej w postaci tabelarycznej.

W charakterze funkcji interpolacyjnej najczęściej jest stosowany wielomian. Załóżmy, że dana jest funkcja  $f(x)$  w postaci tablicy, w której punktom  $x_1, x_2, \dots, x_n$ , zwanym węzłami interpolacji przyporządkowane są wartości  $f(x_1), f(x_2), \dots, f(x_n)$ . Zakłada się, że  $x_i \neq x_j$  dla  $i \neq j$ . Oznaczmy  $i$ -tą różnicę

$$h_i = x_{i+1} - x_i \quad (1)$$

$$\text{Wyrażenia } \Delta_i = \frac{f(x_{i+1}) - f(x_i)}{h_i} = \frac{f(x_{i+1}) - f(x_i)}{x_{i+1} - x_i} \quad (2)$$

nazywa się ilorazami różnicowymi pierwszego rzędu. Odpowiednio także:

$$\Delta_i^{(2)} = \frac{\Delta_{i+1} - \Delta_i}{h_{i+1} + h_i} = \frac{(x_{i+1} - x_i)(f(x_{i+2}) - f(x_{i+1})) - (x_{i+2} - x_{i+1})(f(x_{i+1}) - f(x_i))}{(x_{i+2} - x_{i+1})(x_{i+1} - x_i)(x_{i+2} - x_i)} \quad (3)$$

$$\Delta_i^{(3)} = \frac{\Delta_{i+1}^{(2)} - \Delta_i^{(2)}}{h_{i+2} + h_{i+1} + h_i} = \frac{\Delta_{i+1}^{(2)} - \Delta_i^{(2)}}{x_{i+3} - x_i} \quad \text{- iloraz trzeciego rzędu} \quad (4)$$

$$\text{i ogólnie: } \Delta_i^{(k)} = \frac{\Delta_{i+1}^{(k-1)} - \Delta_i^{(k-1)}}{x_{i+k} - x_i} \quad \text{dla } k = 1, 2, \dots, n-1, i = 1, 2, \dots, n-k \quad (5)$$

Wielomian interpolacyjny Newtona ma następującą postać:

$$P(x) = f(x_1) + \Delta_1(x - x_1) + \Delta_1^{(2)}(x - x_1)(x - x_2) + \dots + \Delta_1^{(n-1)}(x - x_1)(x - x_2)\dots(x - x_{n-1}) \quad (6)$$

Można sprawdzić, że:  $P(x_i) = f(x_i)$  dla  $i = 1, 2, \dots, n$ .

Algorytm interpolacyjny Newtona sprowadza się zatem do obliczenia ilorazów różnicowych oraz określenia wartości wielomianu dla konkretnej wartości zmiennej  $x$ . Ważne znaczenie ma przypadek, gdy wszystkie punkty stałe (węzły) są jednakowo od siebie oddalone. Wówczas mamy:

$h = h_i = x_{i+1} - x_i = \text{const}$  oraz

$$\Delta_i = \frac{f(x_{i+1}) - f(x_i)}{h}, \quad \Delta_i^{(2)} = \frac{\Delta_{i+1} - \Delta_i}{2h} = \frac{f(x_{i+2}) - 2f(x_{i+1}) + f(x_i)}{2h^2},$$

$$\Delta_i^{(k)} = \frac{\Delta_{i+1}^{(k-1)} - \Delta_i^{(k-1)}}{kh} \quad \text{dla } k = 1, 2, \dots, n-1, i = 1, 2, \dots, n-k$$

dzięki czemu upraszcza się reprezentacja funkcji interpolującej (6), gdyż:

$$x_2 = x_1 + h, \quad x_k = x_1 + (k-1)h, \quad k = 1, 2, \dots, n \quad (7)$$

Wprowadzając nową zmienną:  $q = x - x_1$ ,

a zatem:  $q - h = x - x_2 = x - x_1 - h$ ,  $q - (n-2)h = x - x_{n-1} = x - x_1 - (n-2)h$

otrzymamy z (6)

$$P(q) = f(x_1) + q\Delta_1 + q(q-h)\Delta_1^{(2)} + \dots + q(q-h)\dots(q-(n-2)h)\Delta_1^{(n-1)} \quad (8)$$

**Przykład.** Określić ogólną postać wielomianu interpolacyjnego Newtona dla funkcji dyskretnej reprezentowanej przez trzy kolejne równooddalone punkty.

W tym przypadku wielomian interpolacyjny (8) jest ograniczony do trzech wyrazów:

$P(q) = f(x_1) + q\Delta_1 + q(q-h)\Delta_1^{(2)}$ , co po podstawieniu odpowiednich wartości przyjmuje następującą formę ( $q$  jest odległością od początku przedziału):

$$P(q) = f(x_1) + q \frac{f(x_2) - f(x_1)}{h} + q(q-h) \frac{f(x_3) - 2f(x_2) + f(x_1)}{2h^2}$$

Podstawiając  $d = q/h$  ( $d$  jest w ten sposób względną odległością od początku przedziału) otrzymamy:

$$P(d) = f(x_1) + d(f(x_2) - f(x_1)) + d(d-1) \frac{f(x_3) - 2f(x_2) + f(x_1)}{2}$$

Po uporządkowaniu otrzymamy:

$$P(d) = \frac{1}{2} (2f(x_1) - d(3f(x_1) - 4f(x_2) + f(x_3)) + d^2(f(x_1) - 2f(x_2) + f(x_3)))$$

W ten sposób, na przykład, wartość funkcji w środku drugiego ( $d=1,5$ ) odcinka można oszacować następująco:

$$P(1,5) = \frac{1}{2} \left( 2f(x_1) - \frac{3}{2}(3f(x_1) - 4f(x_2) + f(x_3)) + \frac{9}{4}(f(x_1) - 2f(x_2) + f(x_3)) \right) = \frac{1}{8} (3f(x_3) + 6f(x_2) - f(x_1))$$

Funkcję interpolującą można wykorzystać do określenia algorytmu numerycznego różniczkowania. Odpowiednią formułę uzyskuje się przez różniczkowanie funkcji interpolującej:

$$D(x) = \frac{d}{dx} P(x)$$

Na przykład, dla aproksymacji 3-punktowej (jak w Przykładzie) otrzymamy:

$$P'(q) = \frac{d}{dq} P(q) = \frac{1}{h} \left( f(x_2) - f(x_1) - \frac{f(x_3) - 2f(x_2) + f(x_1)}{2} + 2q \frac{f(x_3) - 2f(x_2) + f(x_1)}{2h} \right)$$

$$\text{lub } P'(d) = \frac{1}{2h} [-(3f(x_1) - 4f(x_2) + f(x_3)) + 2d(f(x_1) - 2f(x_2) + f(x_3))].$$

$$\text{Dla różniczkowania na końcu przedziału } (d=2) \text{ otrzymamy: } D(2) = \frac{3f(x_3) - 4f(x_2) + f(x_1)}{2h}.$$

$$\text{Podobnie, w środku przedziału: } D(1) = \frac{f(x_3) - f(x_1)}{2h}.$$

### Zadanie:

Pomiar wielkości  $y(k)$  odbywa się ze stałym okresem próbkowania  $T$ . Wynik pomiaru jest umieszczony w poniższej Tabelicy.

k	0	1	2	3	4	5	6	7	8	9
$y(k)$										

Stosując 3-punktową interpolację określić wartości tej funkcji oraz jej pochodnej z  $m$ -krotnie większą częstotliwością próbkowania (z okresem  $T_1$ ).

## Ćwiczenie 8. Aproksymacja metodą najmniejszych kwadratów I

Zadanie:

Obserwowany proces jest określony następującą funkcją:

$$f(t) = A \sin(\omega t + \varphi) + \delta(t) \quad (1)$$

gdzie:  $\omega$ ,  $\varphi$ ,  $A$ , model zakłóceń:  $\delta(t)$  podaje prowadzący.

Pomiar sygnału  $f(t)$  odbywa się metodą cyfrową z częstotliwością próbkowania  $f_T$ .

Pomiar jest aproksymowany za pomocą dwóch modeli: wielomianowego i trygonometrycznego (dokładną postać modeli ustala prowadzący). Określić współczynniki obu modeli na podstawie pomiaru sygnału w pierwszym okresie składowej podstawowej. Narysować przebiegi sygnału oryginalnego i obu aproksymacji.

## Ćwiczenie 9.

### Metoda Najmniejszych Kwadratów z wykorzystaniem rozkładu macierzy według wartości szczególnych – SVD

Dowolna macierz  $\mathbf{A}$  ( $m \times n$ ) może być przedstawiona w następującej postaci:

$$\mathbf{A} = \mathbf{U}\mathbf{W}\mathbf{V}^T \quad (1)$$

gdzie:

$\mathbf{U}$  - macierz ( $m \times m$ ), której kolumny spełniają następującą zależność:

$$\sum_{i=1}^m u_{ik}u_{ij} = 1, \quad 1 \leq k \leq n, \quad 1 \leq j \leq n, \quad \text{to znaczy, są wzajemnie ortogonalne;}$$

$\mathbf{W} = \begin{bmatrix} \Sigma & 0 \\ 0 & 0 \end{bmatrix}$  - macierz ( $m \times n$ ) wartości szczególnych, przy czym:

$$\Sigma = \begin{bmatrix} \sigma_1 & & & \\ & \sigma_2 & & \\ & & \dots & \\ & & & \sigma_r \end{bmatrix}, \quad \sigma_1 \geq \sigma_2 \geq \dots \geq \sigma_r \geq 0, \quad r \leq n - \text{rzęd macierzy } \mathbf{A}^* ; \quad (2)$$

$\mathbf{V}$  - macierz kwadratowa ( $n \times n$ ), której kolumny spełniają następującą zależność:

$$\sum_{i=1}^n v_{ik}v_{ij} = 1, \quad 1 \leq k \leq n, \quad 1 \leq j \leq n, \quad \text{a więc są również wzajemnie ortogonalne.}$$

Z powyższego opisu wynika, że:

$$\mathbf{U}^T \mathbf{U} = \mathbf{V}^T \mathbf{V} = \mathbf{V}\mathbf{V}^T = \mathbf{1} \quad (3)$$

Kolumny macierzy  $\mathbf{U}$  są wektorami własnymi macierzy kwadratowej  $\mathbf{A}\mathbf{A}^T$ , natomiast kolumny macierzy  $\mathbf{V}$  są wektorami własnymi macierzy  $\mathbf{A}^T \mathbf{A}$ . Wartości szczególne rozkładu (1) – a więc elementy diagonalne macierzy  $\Sigma$  – są natomiast pierwiastkami kwadratowymi wartości własnych macierzy  $\mathbf{A}\mathbf{A}^T$  lub  $\mathbf{A}^T \mathbf{A}$ . Zależność (1) jest nazywana rozkładem macierzy  $\mathbf{A}$  według wartości szczególnych (ang. SVD – *Singular Value Decomposition*). Wynika stąd sposób obliczania macierzy rozkładu (1).

Wektor własny  $\mathbf{x}$  macierzy  $\mathbf{H}$  spełnia następujące równanie:

$$\mathbf{H}\mathbf{x} = \lambda\mathbf{x} \quad (4)$$

przy czym  $\lambda$  jest wartością własną macierzy  $\mathbf{H}$ . W takim przypadku mówi się, że wektor  $\mathbf{x}$  jest skojarzony w wartość własną  $\lambda$ .

W celu określenia wartości własnych  $\lambda$  oraz odpowiadających im wektorów własnych  $\mathbf{x}$  należy rozwiązać równanie (4), co jest równoważne następującej zależności:

$$(\mathbf{H} - \lambda\mathbf{1})\mathbf{x} = 0 \quad (5)$$

Jednoznaczne rozwiązanie (5) uzyskuje się wówczas, gdy spełniony jest warunek:

$$\det(\mathbf{H} - \lambda\mathbf{1}) = \begin{vmatrix} h_{11} - \lambda & h_{12} & \dots & h_{1m} \\ h_{21} & h_{22} - \lambda & \dots & h_{2m} \\ \vdots & \vdots & \dots & \vdots \\ h_{m1} & h_{m2} & \dots & h_{mm} - \lambda \end{vmatrix} = 0 \quad (6)$$

\* Rząd macierzy  $r$  jest największą liczbą niezależnych wierszy (lub kolumn) macierzy.

Rozwiązanie równania (6) jest równoważne znalezieniu pierwiastków wielomianu  $m$ -tego stopnia względem  $\lambda$ . W ogólnym przypadku jest zatem  $m$  wartości własnych:  $\lambda_1, \lambda_2, \dots, \lambda_m$ . Podstawiając kolejno te wartości własne do (5) otrzymuje się  $m$  równań:

$$\begin{bmatrix} h_{11} - \lambda_i & h_{12} & \cdots & h_{1m} \\ h_{21} & h_{22} - \lambda_i & \cdots & h_{2m} \\ \vdots & \vdots & \cdots & \vdots \\ h_{m1} & h_{m2} & \cdots & h_{mm} - \lambda_i \end{bmatrix} \begin{bmatrix} x_{1i} \\ x_{2i} \\ \vdots \\ x_{mi} \end{bmatrix} = 0, \quad i = 1, 2, \dots, m, \quad (7)$$

po rozwiązaniu których uzyskuje się  $m$  wektorów własnych:  $\mathbf{x}_i = [x_{1i} \ x_{2i} \ \cdots \ x_{mi}]^T$ ,  $i = 1, 2, \dots, m$ .

W rozpatrywanym przypadku macierz  $\mathbf{H} = \mathbf{A}\mathbf{A}^T$  (jeśli obliczana jest macierz  $\mathbf{U}$ ) lub  $\mathbf{H} = \mathbf{A}^T\mathbf{A}$  (jeśli obliczana jest macierz  $\mathbf{V}$ ). Macierz  $\mathbf{\Sigma}$  (2) powstaje przez uporządkowanie pierwiastków z wartości własnych macierzy  $\mathbf{H}$ . W standardowych programach do rozkładu macierzy według wartości szczególnych stosowane są zazwyczaj inne, bardziej efektywne algorytmy w stosunku do przedstawionego powyżej.

Właściwości rozkładu SVD:

$$\mathbf{A} = \mathbf{U}\mathbf{W}\mathbf{V}^T, \quad \mathbf{A}^T = \mathbf{V}\mathbf{W}^T\mathbf{U}^T \quad (8a)$$

$$\mathbf{A}^T\mathbf{A} = \mathbf{V}\mathbf{W}^T\mathbf{U}^T\mathbf{U}\mathbf{W}\mathbf{V}^T = \mathbf{V}\mathbf{W}^T\mathbf{W}\mathbf{V}^T, \quad \mathbf{A}\mathbf{A}^T = \mathbf{U}\mathbf{W}\mathbf{W}^T\mathbf{U}^T \quad (8b)$$

$$\mathbf{A}^\# = (\mathbf{A}^T\mathbf{A})^{-1}\mathbf{A}^T = \mathbf{V}\mathbf{W}^{-1}\mathbf{U}^T - \text{macierz pseudo-odwrotna} \quad (8c)$$

gdzie  $\mathbf{W}^{-1} = \mathbf{W}^\# = \begin{bmatrix} \Sigma^{-1} & 0 \\ 0 & 0 \end{bmatrix}_{(n \times m)}$ ; elementy diagonalne macierzy  $\Sigma^{-1}$  można obliczyć jako

odwrotności odpowiednich elementów diagonalnych macierzy  $\Sigma$ .

Ostatnią zależność można bezpośrednio wykorzystać do rozwiązywania zadań MNK:

$$\mathbf{x} = \mathbf{H}^\#\mathbf{y} = \mathbf{V}\mathbf{W}^\#\mathbf{U}^T\mathbf{y} \quad (9)$$

gdzie:  $\mathbf{y}$  - wektor pomiarów,  $\mathbf{x}$  - wektor poszukiwanych parametrów funkcji aproksymującej.

Zastosowanie z użyciem programu MATLAB:

$[\mathbf{U}, \mathbf{W}, \mathbf{V}] = \text{svd}(\mathbf{A});$  rozkład macierzy  $\mathbf{A}$  według wartości szczególnych,

$[\mathbf{C}, \mathbf{D}] = \text{eig}(\mathbf{A}^*\mathbf{A}');$  wartości własne macierzy  $\mathbf{A}^*\mathbf{A}'$ .

Zadanie:

1. Określić rozkład macierzy  $\mathbf{A}$  (podaje prowadzący) według wartości szczególnych. Obliczyć wartości własne i wektory własne macierzy  $\mathbf{A}^T\mathbf{A}$  oraz  $\mathbf{A}\mathbf{A}^T$ . Porównać oba wyniki.

2. Sprawdzić związek (8c):

$$(\mathbf{A}^T\mathbf{A})^{-1}\mathbf{A}^T = \mathbf{V}\mathbf{W}^{-1}\mathbf{U}^T$$

3. Rozwiązać problem MNK z Zadania 8 za pomocą metody SVD.

## Ćwiczenie 10. Aproksymacja metodą najmniejszych kwadratów II

Zadanie:

Obserwowany proces jest określony następującą funkcją:

$$f(t) = A_1 \sin(\omega t + \varphi_1) + A_2 \sin(2\omega t + \varphi_2) + A_5 \sin(5\omega t + \varphi_5) + \delta(t) \quad (1)$$

gdzie:  $\omega$ ,  $\varphi_1, A_1, \varphi_2, A_2, \varphi_5, A_5$  oraz model zakłóceń  $\delta(t)$  podaje prowadzący.

Korzystając z metody najmniejszych kwadratów określić współczynniki filtrów ortogonalnych do estymacji  $m$ -tej harmonicznej sygnału. Filtry powinny realizować przetwarzanie zgodnie z następującymi zależnościami:

$$\begin{aligned} X_{mc}(k) &= p \sum_{i=0}^{N-1} h_c(i) f(k+i-N+1), \\ X_{ms}(k) &= p \sum_{i=0}^{N-1} h_s(i) f(k+i-N+1) \end{aligned} \quad (2)$$

gdzie:  $h_c(i)$ ,  $h_s(i)$  - współczynniki filtrów,  $N$  - liczba próbek sygnału w okresie składowej podstawowej.

Pomiar sygnału  $f(t)$  odbywa się metodą cyfrową z częstotliwością próbkowania  $f_T$ .

Do określenia współczynników filtru przyjąć model sygnału, który podaje prowadzący.

Dokonać przetwarzania sygnału zgodnie z (2) i określić amplitudę wskazanej harmonicznej.

## Ćwiczenie 11. Całkowanie funkcji

### Zadanie:

Obliczyć wartość całki dla podanej przez prowadzącego funkcji jednej zmiennej korzystając z metody:

- prostokątów,
- trapezów,
- Simpsona.

przy zadanej długości kroku całkowania  $T$ .

Porównać otrzymane wyniki z wartością uzyskaną analitycznie.



## Ćwiczenie 12.

**Rozwiązywanie równań różniczkowych – metoda prostokątów i metoda trapezów.**

Zadanie:

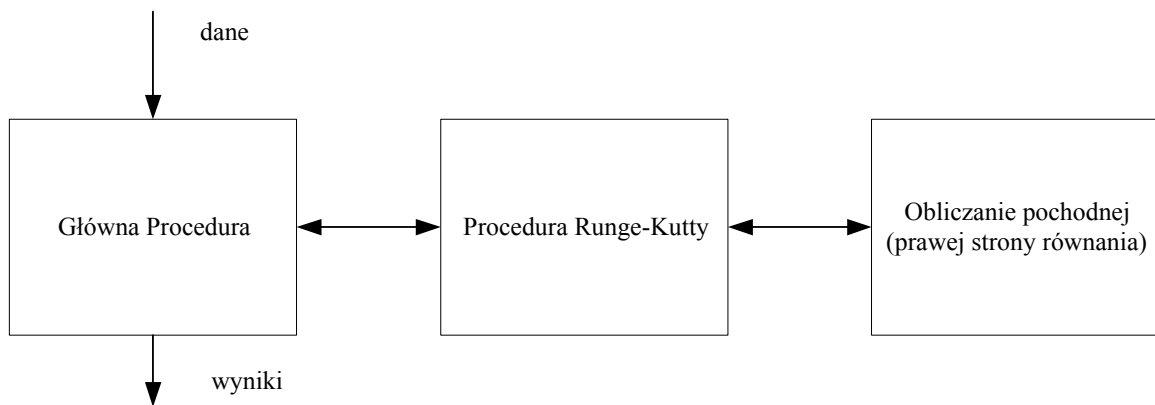
Dla określonego przez prowadzącego równania różniczkowego, i określonych warunków początkowych, przedstawić graficznie rozwiązanie według metody trapezów oraz niejawniej metody Eulera. Przyjąć krok całkowania równy  $h$ .

### Ćwiczenie 13. Rozwiązywanie równań różniczkowych

#### Zadanie:

Dla danego przez prowadzącego układu równań różniczkowych, z warunkami początkowymi, stosując metodę Runge-Kutty 4-rzędu określić przebieg rozwiązania w przedziale  $t = 0 \dots t_{max}$ , przyjmując stały krok całkowania  $h$ . Wynik przedstawić graficznie.  
Porównać rozwiązania dla różnych wartości  $h$ .

Uwaga: opracować program o następującej strukturze:



## Ćwiczenie 14 (program rozszerzony).

### Wykorzystanie rozkładu LU do rozwiązywania układu równań liniowych

#### Zadanie:

1. Dokonać rozkładu LU macierzy współczynników  $\mathbf{A}$  układu równań liniowych  $\mathbf{Ax} = \mathbf{b}$  z zadania 2 (metoda eliminacji Gaussa rozwiązywania układu równań liniowych) na trójkątne macierze  $\mathbf{L}$  oraz  $\mathbf{U}$ .
2. Znaleźć wektor rozwiązania  $\mathbf{x}$  w dwóch etapach, rozwiązując kolejno następujące równania:  
$$\mathbf{Lz} = \mathbf{b}$$
a następnie  
$$\mathbf{Ux} = \mathbf{z},$$
pamiętając, że  $\mathbf{LUx} = \mathbf{b}$ , gdzie  $\mathbf{LU} = \mathbf{A}$ .
3. Wyznaczyć wartość wyznacznika macierzy  $\mathbf{A}$  na podstawie wartości wyznaczników macierzy  $\mathbf{L}$  i  $\mathbf{U}$ .
4. Porównać uzyskane rezultaty z wynikami działania funkcji LU programu MATLAB (przykładowe użycie: a)  $[\mathbf{L}, \mathbf{U}] = \text{lu}(\mathbf{A})$ ; b)  $[\mathbf{L}, \mathbf{U}, \mathbf{P}] = \text{lu}(\mathbf{A})$ ; - z porządkowaniem macierzy  $\mathbf{A}$  i macierzą permutacji)
5. Porównać uzyskane rezultaty z wynikami otrzymanymi w zadaniu 2.