Laboratorium Techniki Obliczeniowej i Symulacyjnej

Ćwiczenie 11. Interpolacja i aproksymacja funkcji.

Opracował: dr inż. Sebastian Dudzik

1. Cel ćwiczenia

Celem ćwiczenia jest zapoznanie się z podstawowymi metodami numerycznymi w tym: interpolacją, aproksymacją i całkowaniem numerycznym.

2. Wprowadzenie

2.1. Interpolacja

2.1.1. Wprowadzenie

Dana jest funkcja $y = f(x),\ x \in [x_0, x_n]$, dla której znamy tablicę jej wartości $f(x_0) = y_0,\ f(x_1) = y_1,\ f(x_n) = y_n$. Należy wyznaczyć funkcję $W(x)$, taką aby:

$$W(x_0) = y_0,\ W(x_1) = y_1,\ \ldots,\ W(x_n) = y_n. \tag{1}$$

Najczęściej funkcję $W(x)$ dobiera się jako:

$$W(x) = \sum_{i=0}^{n} a_1 \phi_1(x). \tag{2}$$

Zależność (2), można zapisać jako:

$$W(x) = \mathbf{\Phi}(x)\mathbf{A}, \tag{3}$$

gdzie: $\mathbf{\Phi} = [\phi_0(x), \phi_1(x), \ldots, \phi_n(x)]$, $A^T = [a_0, a_1, \ldots, a_n]$.

Macierz $\mathbf{A}$ uzyskuje się rozwiązując układ równań liniowych:

$$\mathbf{XA} = \mathbf{Y}, \tag{4}$$

gdzie:

$$\mathbf{X} = \begin{bmatrix} \phi_0(x_0) & \phi_1(x_0) & \ldots & \phi_n(x_0) \\ \phi_0(x_1) & \phi_1(x_1) & \ldots & \phi_n(x_1) \\ \ldots & \ldots & \ldots & \ldots \\ \phi_0(x_n) & \phi_1(x_n) & \ldots & \phi_n(x_n) \end{bmatrix}, \quad \mathbf{Y} = \begin{bmatrix} y_0 \\ y_1 \\ \vdots \\ y_n \end{bmatrix} \tag{5}$$

2.1.2. Interpolacja Lagrange'a

W wielomianowej interpolacji Lagrange'a, dla $n+1$ węzłów interpolacji przyjmuje się następujące funkcje bazowe:

$$\begin{array}{l} \phi_0(x) = (x-x_1)(x-x_2)(x-x_3)\ldots\ldots\ldots\ldots(x-x_n) \\ \phi_1(x) = (x-x_0)(x-x_2)(x-x_3)\ldots\ldots\ldots\ldots(x-x_n) \\ \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \\ \phi_i(x) = (x-x_0)(x-x_1)\ldots(x-x_{i-1})(x-x_{i+1})\ldots(x-x_n) \\ \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \\ \phi_n(x) = (x-x_0)(x-x_1)(x-x_2)\ldots\ldots\ldots\ldots(x-x_{n-1}) \end{array} \tag{6}$$

Wielomian interpolacyjny wyraża się wzorem:

$$\begin{aligned} W(x) &= a_0(x-x_1)(x-x_2)\ldots(x-x_n) + a_1(x-x_0)(x-x_2)\ldots(x-x_n) + \\ &+ a_n(x-x_0)(x-x_1)\ldots(x-x_{n-1}), \end{aligned} \tag{7}$$

Współczynniki wielomianu można wyznaczyć z układu równań (4), przy czym macierz $\mathbf{X}$ przyjmuje postać:

$$\begin{bmatrix} (x_0-x_1)\ldots(x_0-x_n) & 0 & \ldots & 0 \\ 0 & (x_1-x_0)\ldots(x_1-x_n) & \ldots & 0 \\ \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & \ldots & (x_n-x_0)\ldots(x_n-x_{n-1}) \end{bmatrix} \tag{8}$$

Wielomian interpolacyjny Lagrange'a można zapisać w następującej postaci:

$$W(x) = \sum_{i=0}^{n} y_i \frac{\prod\limits_{j\neq i}(x-x_j)}{\prod\limits_{j\neq i}(x_i-x_j)},\ j=0,1,\ldots,n \tag{9}$$

Poniżej przedstawiono algorytm wyznaczania współczynników wielomianu interpolacyjnego Lagrange'a drugiego rzędu dla zbioru trzech punktów.

Zmienne

rzeczywiste: a, b, c

macierze: x, y

Podaj $x_1, y_1, x_2, y_2, x_3, y_3$

Oblicz (10)

$$\begin{aligned} a &:= \frac{y_1}{(x_1-x_2)(x_1-x_3)} + \frac{y_2}{(x_2-x_1)(x_2-x_3)} + \frac{y_3}{(x_3-x_1)(x_3-x_2)} \\ b &:= \frac{y_1(x_2+x_3)}{(x_1-x_2)(x_1-x_3)} + \frac{y_2(x_1+x_3)}{(x_2-x_1)(x_2-x_3)} + \frac{y_3(x_1+x_2)}{(x_3-x_1)(x_3-x_2)} \\ c &:= \frac{y_1x_2x_3}{(x_1-x_2)(x_1-x_3)} + \frac{y_2x_1x_3}{(x_2-x_1)(x_2-x_3)} + \frac{y_3x_1x_2}{(x_3-x_1)(x_3-x_2)} \end{aligned}$$

2.1.3. Różnice skończone. Pierwszy wzór interpolacyjny Newtona

Różnicę skończoną rzędu k, funkcji stabelaryzowanej przy stałym kroku $h = x_{i+1} - x_i$ definiuje się w sposób następujący:

$$
\begin{aligned}
&\Delta y_i = y_{i+1} - y_i, \\
&\Delta^2 y_i = \Delta[\Delta y_i] = \Delta y_{i+1} - \Delta y_i = y_{i+2} - 2y_i + y_i, \\
&\vdots \\
&\Delta^k y_i = \Delta[\Delta^{k-1} y_1] = \sum_{j=0}^{k} (-1)^j \binom{k}{j} y_{i+k-1}
\end{aligned}
\tag{11}
$$

Jeżeli $f(x)$ jest wielomianem stopnia n, to różnica skończona rzędu n tej funkcji jest stała a kolejne różnice są równe zero. Jeżeli, dla funkcji stabelaryzowanej przy stałym kroku h utworzy się diagonalną tablicę różnic skończonych, i różnice rzędu k można uznać za stałe, to wielomian stopnia k, przybliża z dużą dokładnością zadany zbiór punktów. Poniżej przedstawiono algorytm obliczający różnice skończone dla funkcji stabelaryzowanej przy stałym kroku.

$$
\begin{array}{l}
\textbf{Zmienne} \\
\qquad\qquad \text{całkowite: } n, i, j \\
\qquad\qquad \text{macierz: } z \\
\qquad \textbf{Podaj } n \\
\qquad \left\{ \begin{array}{l} \textbf{Dla } j := 1, 2, \ldots, n \\ \qquad \left\{ \begin{array}{l} \textbf{Dla } i := 0, 1, \ldots, n-j \\ \qquad \textbf{Oblicz } z_{ij} := z_{i+1,j-1} - z_{i,j-1} \end{array} \right. \end{array} \right.
\end{array}
\tag{12}
$$

W powyższym algorytmie, wartości funkcji stabelaryzowanej w węzłach interpolacji umieszczono w zerowej kolumnie macierzy $\mathbf{Z}$. Wielomian interpolacyjny dla pierwszego wzoru interpolacyjnego Newtona można zapisać jako:

$$
\begin{aligned}
W(x) &= a_0 + a_1 q + a_2 q(q-1) + a_3 q(q-1)(q-2)(q-3) + \ldots + \\
&+ a_n q(q-1)(q-2)\ldots(q-n+1), \ q = \frac{x - x_0}{h}.
\end{aligned}
\tag{13}
$$

Po obliczeniu współczynników wielomianu interpolacyjnego (układ równań (4)), pierwszy wzór interpolacyjny Newtona przyjmuje postać:

$$
W(x) = y_0 + q\Delta y_0 + \frac{q(q-1)}{2!}\Delta^2 y_0 + \ldots \frac{q(q-1)\ldots(q-n+1)}{n!}\Delta^n y_0. \tag{14}
$$

Algorytm tworzący tabelę różnic skończonych do rzędu 2 i obliczający wartość funkcji w dowolnym punkcie przedziału $x \in [x_0, x_n]$, przedstawiono poniżej.

Zmienne
całkowite: n, i, j, k
rzeczywiste: x_0, h, x, q, q_0
macierze: z
Podaj n, x, x_0, h
Dla $j := 1, 2$
  Dla $i := 0, 1, \ldots, n-j$
    Oblicz $z_{ij} := z_{i+1,j-1} - z_{i,j-1}$
Oblicz $q_0 := \dfrac{x - x_0}{h}$ (15)
Oblicz $k := int(q_0)$
Oblicz $q := q_0 - k$
Jeżeli $k \geq n-1$ **to Drukuj** "brakuje danych"
  w przeciwnym razie
  Oblicz $wart := z_{k0} + qz_{k1} + \dfrac{q(q-1)}{2} z_{k2}$
  Drukuj $wart$

2.1.4. Wielomiany Czebyszewa

W metodzie opartej na wielomianach interpolacyjnych Czebyszewa, zakłada się, że argumenty interpolowanej funkcji mieszczą się w przedziale $[-1, 1]$. Jeżeli tak nie jest należy dokonać normalizacji:

$$x* = \frac{a+b}{2} + \frac{b-a}{2} x, \tag{16}$$

gdzie: $x^* \in [a, b]$, x — wartość argumentu dla przedziału znormalizowanego $[-1, 1]$.

Bazę Czebyszewa $T_0(x), T_1(x), \ldots, T_n(x)$ stanowi zbiór wielomianów określonych zależnością:

$$T_{k+1}(x) = 2xT_k(x) - T_{k-1}(x),\ T_0(x) = 1,\ T_1(x) = x. \tag{17}$$

Współczynniki wielomianu interpolacyjnego oblicza się ze wzoru 4, przy czym jako funkcje bazowe przyjmuje się wielomiany Czebyszewa. Przy dowolnym doborze węzłów, wyznaczanie macierzy $\mathbf{X}^{-1}$ jest o wiele dokładniejsze. Przy odpowiednim doborze węzłów i modyfikacji bazy, macierz odwrotną można obliczyć na podstawie wzoru:

$$\mathbf{X}^{-1} = \frac{2}{n+1} \mathbf{X}^T. \tag{18}$$

W takim przypadku węzły interpolacji muszą spełniać równanie:

$$x_i = \cos \frac{(2i+1)\pi}{2n+2},\ i = 0, 1, \ldots, n, \tag{19}$$

a wektor funkcji bazowych przyjmuje postać:

$$\Phi = \begin{bmatrix} \frac{1}{\sqrt{2}} & x & 2x^2 - 1 & 4x^3 - 3x & \ldots \end{bmatrix} \tag{20}$$

Poniżej przedstawiono algorytm wyznaczania macierzy $\mathbf{X}^{-1}$ dla bazy wielomianów Czebyszewa ($n = 3$), przy doborze węzłów interpolacyjnych zgodnie z zależnością (19).

Zmienne

całkowite: i, j

rzeczywiste: wsp

macierze: x, z

Oblicz $wsp := \frac{1}{2}$

$$\left\{\begin{array}{l}\textbf{Dla } i := 0, 1, \ \ldots, 3 \\ \qquad \textbf{Oblicz } x_i := \cos\left(\dfrac{2i+1}{8}\pi\right)\end{array}\right.$$

Zdefiniuj $T0(y) := \dfrac{1}{\sqrt{2}}$

Zdefiniuj $T1(y) := y$

Zdefiniuj $T2(y) := 2y^2 - 1$ (21)

Zdefiniuj $T3(y) := 4y^3 - 3y$

$$\left\{\begin{array}{l}\textbf{Dla } j := 0, 1, \ \ldots, 3 \\ \qquad \textbf{Oblicz } z_{0j} := wspT0(x_j) \\ \qquad \textbf{Oblicz } z_{1j} := wspT1(x_j) \\ \qquad \textbf{Oblicz } z_{2j} := wspT2(x_j) \\ \qquad \textbf{Oblicz } z_{3j} := wspT3(x_j)\end{array}\right.$$

$$\left\{\begin{array}{l}\textbf{Dla } i := 0, 1, \ \ldots, 3 \\ \qquad \textbf{Dla } j := 0, 1, \ \ldots, 3 \\ \qquad\qquad \textbf{Drukuj } z_{ij}\end{array}\right.$$

2.2. Aproksymacja

2.2.1. Wprowadzenie

Dana jest funkcja jednej zmiennej $y = f(x)$, $x \in [a, b]$, zadana w postaci zbioru punktów: $f(x_1) = y_1$, $f(x_2) = y_2, \ldots, f(x_n) = y_n$. Zadanie aproksymacji polega na takim doborze funkcji $F(x, p_0, \ldots, p_k)$, $x \in [a, b]$, aby w sensie przyjętego kryterium funkcja ta odtwarzała $f(s)$ z jak największą dokładnością. Aproksymacja punktowa ma miejsce, gdy $f(x)$ dana jest w postaci dyskretnej a aproksymacja integralna, jeśli $f(x)$ jest zadana w postaci wzoru analitycznego. Kryteria aproksymacji punktowej dla funkcji jednej zmiennej są tak dobrane, aby różnice pomiędzy wartościami funkcji w punktach (x_i, y_i), były jak najmniejsze. Ogólna postać funkcji $F(x_i, p_0, \ldots, p_k)$ jest zadana z góry, natomiast optymalizuje się wartości parametrów $p_0, \ldots, p_k$.

2.2.2. Metoda najmniejszych kwadratów

W metodzie najmniejszych kwadratów, wartości parametrów funkcji F dobiera się z uwzględnieniem następującego kryterium:

$$\sum_{i=1}^{n}[F(x_i, p_0, \ldots, p_k) - y_i]^2 = \min. \tag{22}$$

Zazwyczaj zagadnienie aproksymacji rozpatruje się w klasie wielomianów uogólnionych, t.j.:

$$F(x, p_0, \ldots, p_k) = p_0\phi_0(x) + \ldots + p_k\phi_k(x). \tag{23}$$

W przypadku aproksymacji liniowej funkcji jednej zmiennej, rozpatrywać będziemy zbiór punktów: (x_1, y_1), (x_2, y_2), $\ldots (x_n, y_n)$. Postać funkcji aproksymującej jest następująca:

$$y = p_0 + p_1 x, \tag{24}$$

a kryterium najmniejszych kwadratów przyjmuje postać:

$$S(p_0, p_1) = \sum_{i=1}^{n}(p_0 + p_1 x_i - y_i)^2 = \min. \tag{25}$$

Minimalizując funkcję S (warunek konieczny i wystarczający), otrzymuje się układ równań:

$$\begin{array}{l} \dfrac{\partial S}{\partial p_0} = 2\sum\limits_{i=1}^{n}(p_0 + p_1 x_i - y_i) = 0 \\ \dfrac{\partial S}{\partial p_1} = 2\sum\limits_{i=1}^{n}(p_0 + p_1 x_i - y_i)x_i = 0 \end{array}, \tag{26}$$

lub, po przekształceniach:

$$\begin{array}{l} p_0 n + p_1 \sum\limits_i x_i = \sum\limits_i y_i \\ p_0 \sum\limits_i x_i + p_1 \sum\limits_i x_i^2 = \sum x_i y_i \end{array}. \tag{27}$$

Wprowadzając macierz danych wejściowych:

$$\mathbf{X} = \begin{bmatrix} 1 & x_1 \\ 1 & x_2 \\ \vdots & \vdots \\ 1 & x_n \end{bmatrix}, \tag{28}$$

powyższy układ równań można zapisać w formie macierzowej:

$$\mathbf{X}^T\mathbf{X}\mathbf{P} = \mathbf{X}^T\mathbf{Y}, \tag{29}$$

gdzie:

$$\mathbf{P} = \begin{bmatrix} p_0 \\ p_1 \end{bmatrix}. \tag{30}$$

Algorytm aproksymujący zbiór punktów $P_i(x_i, y_i)$, $i = 1, 2, \ldots, n$ wzorem w postaci $y = p_0 + p_1 x$ przedstawiono poniżej.

Zmienne

całkowite: i, j

rzeczywiste: wsp

macierze: x, z

Podaj n

Podstaw $s1 := 0$

Podstaw $s2 := 0$

Podstaw $s3 := 0$

Podstaw $s4 := 0$

$$\left\{\begin{array}{l} \textbf{Dla } i := 1, 2, \ \ldots, n \\ \quad \textbf{Oblicz } s1 := s1 + x_i \\ \quad \textbf{Oblicz } s2 := s2 + y_i \\ \quad \textbf{Oblicz } s3 := s3 + x_i^2 \\ \quad \textbf{Oblicz } s4 := s4 + x_i y_i \end{array}\right. \tag{31}$$

Oblicz $w := ns3 - s1^2$

Oblicz $w1 := s2s3 - s_4 s_1$

Oblicz $w2 := ns4 - s1s2$

Oblicz $p0 := \dfrac{w1}{w}$

Oblicz $p1 := \dfrac{w2}{w}$

Drukuj $p0, p1$

Rozważania dotyczące aproksymacji zbioru punktów funkcją liniową można uogólnić na klasę funkcji, liniowych względem większej liczby parametrów. W takim przypadku, kryterium najmniejszych kwadratów można zapisać jako:

$$S(p_0, \ldots, p_k) = \sum_{i=1}^{n} [p_0\phi_0(x_i) + p_1\phi_1(x_i) + \ldots + p_k\phi_k(x_i) - y_i]^2 = \min. \tag{32}$$

Wykorzystując warunki istnienia ekstremum funkcji wielu zmiennych i wprowadzając uogólnioną macierz danych $\mathbf{X}$ oraz wektor wartości funkcji $\mathbf{Y}$ w postaci:

$$\mathbf{X} = \begin{bmatrix} \phi_0(x_1) & \phi_1(x_1) & \phi_2(x_1) & \dots & \phi_k(x_1) \\ \phi_0(x_2) & \phi_1(x_2) & \phi_2(x_2) & \dots & \phi_k(x_2) \\ . & . & . & \dots & . \\ \phi_0(x_n) & \phi_1(x_n) & \phi_2(x_n) & \dots & \phi_k(x_n) \end{bmatrix}, \quad \mathbf{Y} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix}, \tag{33}$$

wartości nieznanych parametrów, można obliczyć rozwiązując układ równań:

$$\mathbf{P} = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{Y}, \tag{34}$$

gdzie:

$$\mathbf{P} = \begin{bmatrix} p_0 \\ p_1 \\ \dots \\ p_k \end{bmatrix} \tag{35}$$

W praktyce inżynierskiej, często występuje potrzeba aproksymacji liniowej funkcji dwóch zmiennych. Dla przykładu rozpatrzmy przypadek funkcji liniowej zmiennych x i y, zawierającej trzy parametry p_0, p_1, p_2:

$$z = F(x, y, p_0, p_1, p_2) = p_0 + p_1 x + p_2 y. \tag{36}$$

Uwzględniając, że aproksymacja dotyczy zbioru punktów: $(x_1, y_1, z_1),\ x_2, y_2, z_2,\ \dots,\ (x_n, y_n, z_n)$, kryterium najmniejszych kwadratów formułuje się następująco:

$$S(p_0, p_1, p_2) = \sum_{i=1}^{n} (p_0 + p_1 x_i + p_2 y_i - z_i)^2 = \min. \tag{37}$$

Wykorzystując warunek konieczny istnienia ekstremum otrzymuje się:

$$\begin{cases} \dfrac{\partial S}{\partial p_0} = 2\sum\limits_i (p + 0 + p_1 x_i + p_2 y_i) = 0 \\ \dfrac{\partial S}{\partial p_1} = 2\sum\limits_i (p + 0 + p_1 x_i + p_2 y_i) x_i = 0 \\ \dfrac{\partial S}{\partial p_2} = 2\sum\limits_i (p + 0 + p_1 x_i + p_2 y_i) y_i = 0 \end{cases} \tag{38}$$

Jeżeli macierz danych wejściowych zapiszemy jako:

$$\mathbf{X} = \begin{bmatrix} 1 & x_1 & y_1 \\ 1 & x_2 & y_2 \\ \vdots & \vdots & \vdots \\ 1 & x_n & y_n \end{bmatrix}, \tag{39}$$

to współczynniki funkcji aproksymującej można obliczyć rozwiązując układ równań:

$$\mathbf{P} = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{Y}, \tag{40}$$

gdzie: $\mathbf{P} = [\ p_0 \quad p_1 \quad p_2\ ]^T$. Algorytm doboru współczynników wzoru aproksymującego postaci $z = p_0 + p_1 x + p_2 y$, dla zbioru punktów $P_i(x_i, y_i, z_i),\ i = 1, 2, \ldots, n$ przedstawiono poniżej.

$$
\begin{array}{l}
\textbf{Zmienne} \\
\qquad \text{całkowite: } n, i \\
\qquad \text{rzeczywiste: } s1, s2, s3, s4, s5, s6, s7, s8 \\
\qquad \text{macierze: } x, y, z, a, b, p \\
\quad \textbf{Podaj } n \\
\quad \textbf{Podstaw } s1 := 0 \\
\quad \textbf{Podstaw } s2 := 0 \\
\quad \textbf{Podstaw } s3 := 0 \\
\quad \textbf{Podstaw } s4 := 0 \\
\quad \textbf{Podstaw } s5 := 0 \\
\quad \textbf{Podstaw } s6 := 0 \\
\quad \textbf{Podstaw } s7 := 0 \\
\quad \textbf{Podstaw } s8 := 0 \\
\quad \left\{ \begin{array}{l}
\textbf{Dla } i := 1, 2,\ \ldots,\ n \\
\quad \textbf{Oblicz } s1 := s1 + x_i \\
\quad \textbf{Oblicz } s2 := s2 + y_i \\
\quad \textbf{Oblicz } s3 := s3 + x_i^2 \\
\quad \textbf{Oblicz } s4 := s4 + y_i^2 \\
\quad \textbf{Oblicz } s5 := s5 + x_i y_i \\
\quad \textbf{Oblicz } s6 := s6 + z_i \\
\quad \textbf{Oblicz } s7 := s7 + x_i z_i \\
\quad \textbf{Oblicz } s8 := s8 + y_i z_i
\end{array} \right. \\
\quad \textbf{Podstaw } a11 := n \\
\quad \textbf{Podstaw } a12 := s1 \\
\quad \textbf{Podstaw } a13 := s2 \\
\quad \textbf{Podstaw } a21 := s1 \\
\quad \textbf{Podstaw } a22 := s3 \\
\quad \textbf{Podstaw } a23 := s5 \\
\quad \textbf{Podstaw } a31 := s2 \\
\quad \textbf{Podstaw } a32 := s5 \\
\quad \textbf{Podstaw } a33 := s2 \\
\quad \textbf{Podstaw } b1 := s6 \\
\quad \textbf{Podstaw } b2 := s7 \\
\quad \textbf{Podstaw } b3 := s8 \\
\quad \textbf{Oblicz } \mathbf{P} := \mathbf{A}^{-1}\mathbf{B} \\
\quad \textbf{Drukuj } p_1, p_2, p_3
\end{array}
\tag{41}
$$

3. Program ćwiczenia

1. Uruchomienie programu MATLAB.
   W ćwiczeniu wykorzystano program MATLAB w wersji 5.3 (R11.1). Uruchomienie programu następuje poprzez skrót na pulpicie (Matlab5.3) lub bezpośrednio z katalogu *C:\MatlabR11\ bin*.

2. Uruchomienie programu Wordpad.exe.
   Program można uruchomić poprzez wywołanie: *Start\Programy\Akcesoria\ Wordpad* lub poprzez skrót na pulpicie.

3. Przejście do katalogu roboczego dla grupy laboratoryjnej.
   Domyślnym katalogiem startowym (roboczym) programu MATLAB jest *C:\Matlab R11\ work*. Zadanie polega na przejściu do podkatalogu katalogu *work*. Podkatalog (utworzony na pierwszych zajęciach laboratoryjnych) nazwany jest wybranymi 2 nazwiskami studentów, wchodzących w skład grupy laboratoryjnej.

   (a) Wprowadzić:
   `»pwd`
   W programie MATLAB każde wprowadzone polecenie zatwierdza się klawiszem `<ENTER>`. Zwrócić uwagę na ścieżkę dostępu do katalogu bieżącego.

   (b) Wprowadzić:
   `»cd` *`nazwa_podkatalogu`*
   Parametr *nazwa_ pod-katalogu* powinien składać się z nazwisk 2 wybranych studentów grupy laboratoryjnej (np. `»cd KowalskiNowak`).

4. Interpolacja Lagrange'a.

   (a) Napisać w języku MATLAB i uruchomić program implementujący algorytm (10) dla następującego zbioru węzłów: $f(0) = 4), f(1) = 2, f(2.5) = 2.75$. Zawartość okna poleceń i okna edytora skopiować do programu Wordpad.

   (b) Napisać w języku MATLAB program wykreślający wielomian interpolacyjny Lagrange'a, oraz węzły interpolacji dla współczynników obliczonych przez program napisany w punkcie poprzednim. Wykres i zawartość okna edytora skopiować do programu Wordpad.

   (c) Napisać w języku MATLAB program obliczający wartość wielomianu interpolacyjnego zadanego współczynnikami wyliczonymi w poprzednich podpunktach, dla dwóch wybranych punktów rozmieszczonych pomiędzy węzłami interpolacji (zagęszczanie siatki).

   (d) Napisać program przeprowadzający interpolację zboru wyżej wymienionych punktów z wykorzystaniem funkcji `interp1` języka MATLAB. Program powinien przeprowadzać interpolację następującymi metodami:

      i. najbliższego sąsiada (`nearest`)
      ii. liniowa (`linear`)

iii. sześcienne funkcje sklejane (splajny) (`spline`)

iv. wielomian 3-go stopnia (`cubic`)

(e) Napisać w języku MATLAB program wykreślający wyniki interpolacji przeprowadzonej w poprzednim podpunkcie w formie graficznej. Wykres i zawartość okna edytora skopiować do programu Wordpad.

5. Różnice skończone. Pierwszy wzór Newtona.

(a) Na podstawie algorytmu (11), napisać w języku MATLAB program obliczający różnice skończone, zadanego rzędu dla funkcji stabelaryzowanej. Zawartość okna edytora skopiować do programu Wordpad.

(b) Wykonać program dla funkcji zadanej w postaci tabeli 1, do rzędu 5-go.

Nr	0	1	2	3	4	5
x	1.2	1.3	1.4	1.5	1.6	1.7
y	1.728	2.197	2.744	3.375	4.096	4.913

Tab. 1. Funkcja stabelaryzowana

(c) Dana jest stabelaryzowana funkcja ciśnienia pary nasyconej helu w zależności od temperatury: $T \log p = f(T)$:

T	0.8	1.0	1.2	1.4	1.6	1.8
$T \log p$	0.3566	0.9383	1.5598	2.2169	2.9059	3.6229

Tab. 2. Funkcja stabelaryzowana

Na podstawie algorytmu (15), napisać program tworzący tablicę różnic skończonych do rzędu II i obliczający wartość funkcji dla dowolnego punktu, leżącego pomiędzy węzłami interpolacji. Zawartość okna edytora skopiować do programu Wordpad.

(d) Napisać w języku MATLAB program wykreślający wielomian interpolacyjny wynikający z I wzoru Newtona oraz węzły interpolacji dla współczynników obliczonych przez program napisany w punkcie poprzednim. Wykres skopiować do programu Wordpad.

(e) Napisać program przeprowadzający interpolację zboru punktów z tabel 1 oraz 2 z wykorzystaniem funkcji `interp1` języka MATLAB. Program powinien przeprowadzać interpolację następującymi metodami:

i. najbliższego sąsiada (`nearest`)

ii. liniowa (`linear`)

iii. sześcienne funkcje sklejane (splajny) (`spline`)

iv. wielomian 3-go stopnia (`cubic`)

(f) Napisać w języku MATLAB program wykreślający wyniki interpolacji przeprowadzonej w poprzednim podpunkcie w formie graficznej. Wykres i zawartość okna edytora skopiować do programu Wordpad.

6. Wielomiany Czebyszewa.

(a) Napisać w języku MATLAB program normalizujący zbiór węzłów $x_0 = 4,\ x_1 = 6,\ x_2 = 7,\ x_3 = 10$. Przetestować program dla zbioru węzłów podanego przez prowadzącego. Zawartość okien edytora i linii poleceń skopiować do programu Wordpad.

(b) Napisać w języku MATLAB program obliczający współczynniki zadanej liczby funkcji bazowych Czebyszewa. Przetestować program dla liczby funkcji podanej przez prowadzącego. Zawartość okien edytora i linii poleceń skopiować do programu Wordpad.

(c) Napisać w języku MATLAB program wyznaczający siatkę węzłów zgonie z zależnością (19). Przetestować program. Zawartość okien edytora i linii poleceń skopiować do programu Wordpad.

(d) Napisać program wyznaczający macierz $\mathbf{X}^{-1}$ dla bazy utworzonej z wielomianów Czebyszewa, na podstawie algorytmu (21). Przetestować program dla następujących danych:

x	0.924	0.383	-0.383	-0.924
y	0.3566	0.9383	1.5598	2.2169

Tab. 3. Funkcja stabelaryzowana

7. Aproksymacja liniowa funkcji jednej zmiennej.

(a) W pewnym zakładzie przemysłowym dokonano pomiarów zużycia energii elektrycznej L [MWh] potrzebnej do wyprodukowania N [tys. sztuk] wyrobu. Wyniki badań zebrano w tabeli. Napisać program obliczający parametry p_0, p_1

N	1	2	3	4	5	6	7	8	9	10
L	10	18	22	27	36	49	56	64	70	78

Tab. 4. Funkcja stabelaryzowana

wzoru aproksymującego w postaci: $L = p_0 + p_1 N$ na podstawie algorytmu (30). Program powinien wykreślać punkty aproksymowane oraz prostą aproksymującą. Zawartość okien edytora i linii poleceń oraz wykres skopiować do programu Wordpad

8. Aproksymacja wielomianowa

(a) Wygenerować wektor wartości funkcji $erf(c)$. Wprowadzić:

```
x = (0: 0.1: 2.5)';
y = erf(x);
```

(b) Przeprowadzić aproksymację wygenerowanej funkcji wielomianami stopnia 1, 3 i 6. Wprowadzić:

```
p1 = polyfit(x,y,1)
p3 = polyfit(x,y,3)
p6 = polyfit(x,y,6)
x = (0: 0.1: 5)';
y = erf(x);
f1 = polyval(p1,x);
f3 = polyval(p3,x);
f6 = polyval(p6,x);
plot(x,y,'o',x,f1,'-',x,f3,'--',x,f6,'.' )
axis([0  5  0  2])
grid on
```

Zawartość okien edytora i linii poleceń oraz wykres skopiować do programu Wordpad.

9. Aproksymacja liniowa z czynnikami nie będącymi wielomianami.

(a) Rozważana będzie następująca postać funkcji aproksymującej: $y = a_0 + a_1 e^{-t} + a_2 t e^{-t}$. Jest to przypadek liniowy względem parametrów a_0, a_1, a_2, natomiast nieliniowy względem t. Należy obliczyć wartości współczynników funkcji aproksymującej. Wprowadzić:

```
t = [0 0.3 0.8 1.1 1.6 2.3]';
y = [0.6 0.67 1.01 1.35 1.47 1.25]';
X = [ones(size(t))  exp(-t)  t.*exp(-t)];
a = X\y
T = (0:0.1:2.5)';
Y = [ones(size(T))  exp(-T)  T.*exp(-T)]*a;
plot(T,Y,'-',t,y,'o'), grid on
```

10. Aproksymacja liniowa funkcji dwóch zmiennych

(a) Zbiór punktów zebrano w tabeli. Napisać program obliczający współczynniki wzoru w postaci $z = p_0 + p_1 x + p_2 y$, aproksymującego zadany zbiór punktów. Wykorzystać algorytm (41).

4. Opracowanie sprawozdania

W sprawozdaniu należy umieścić polecenia oraz wyniki ich działania skopiowane w trakcie ćwiczenia z okna środowiska MATLAB. Do każdej linii kodu oraz do każdego wyniku,

x	0.2	0.2	0.2	0.4	0.4	0.6
y	0.2	0.4	0.6	0.2	0.4	0.6
z	3.25	2.58	2.42	2.78	2.40	1.56

Tab. 5. Funkcja stabelaryzowana

należy dodać komentarz objaśniający.

Przykład.
... `2+round(6/9+3*2)/2-3` — obliczenie wartości wyrażenia. Funkcja `round(6/9+3*2)` zaokrągla wynik działania `6/9+3*2` do najbliższej liczby całkowitej...