

POLITECHNIKA BIAŁOSTOCKA

WYDZIAŁ INFORMATYKI

KATEDRA MATEMATYKI

PRACA DYPLOMOWA MAGISTERSKA

TEMAT: KONSTRUKCJA DWUWYMIAROWYCH
KWADRATUR NEWTONA-COTESA I ICH
ZASTOSOWANIE DO OBLICZANIA CAŁKI PODWÓJNEJ

WYKONAWCA: SZYMON DĄBROWSKI

.....
podpis

PROMOTOR: DR JAN POPIOŁEK

.....
podpis

BIAŁYSTOK 2017 r.

Karta dyplomowa

<p>Politechnika Białostocka</p> <p>Wydział Informatyki</p> <p>Katedra Matematyki</p>	<p>Studia stacjonarne</p> <p>II stopnia magisterskie</p>	<p>Numer albumu</p> <p>studenta: 87901</p>
		<p>Rok akademicki 2016/2017</p>
		<p>Kierunek studiów:</p> <p>Informatyka</p> <p>Specjalność: Informatyka i</p> <p>finanse</p>
<p style="text-align: center;">Szymon Dąbrowski</p> <p>Temat pracy dyplomowej: Konstrukcja dwuwymiarowych kwadratur</p> <p>Zakres pracy:</p> <p>Newtona-Cotesa i ich zastosowanie do obliczania całki podwójnej</p> <ol style="list-style-type: none"> 1. Wzrosty Lagrange'a jednej i dwóch zmiennych. 2. Jednowymiarowe kwadratury Newtona-Cotesa. 3. Dwuwymiarowe kwadratury Newtona-Cotesa. 4. Zastosowanie kwadratur do obliczania całek. <div style="display: flex; justify-content: space-between;"> <div> <p>.....</p> <p>Imię i nazwisko promotora - podpis</p> </div> <div> <p>.....</p> <p>Imię i nazwisko kierownika katedry - podpis</p> </div> </div>		
<p>.....</p> <p>Data wydania tematu pracy dyplomowej - podpis promotora</p>	<p>.....</p> <p>Regulaminowy termin złożenia pracy dyplomowej</p>	<p>.....</p> <p>Data złożenia pracy dyplomowej - potwierdzenie dziekanatu</p>
<div style="display: flex; justify-content: space-between;"> <div> <p>.....</p> <p>Ocena promotora</p> </div> <div> <p>.....</p> <p>Podpis promotora</p> </div> </div>		
<p>.....</p> <p>Imię i nazwisko recenzenta</p>	<p>.....</p> <p>Ocena recenzenta</p>	<p>.....</p> <p>Podpis recenzenta</p>

Thesis topic: Construction of twodimensional Newton-Cotes quadratures and their application to calculate of double integral.

SUMMARY

Key words: Newton-Cotes quadratures; Lagrange polynomials;

Plik OswiadczenieOSamodzielności.pdf

Spis treści

Wstęp	6
1 Wielomiany interpolacyjne <i>Lagrange’a</i>	7
1.1 Wielomiany <i>Lagrange’a</i> dla funkcji jednej zmiennej	7
1.2 Wielomiany <i>Lagrange’a</i> dla funkcji dwóch zmiennych	12
2 Kwadratury Newtona-Cotesa dla całki pojedynczej	17
2.1 Wzór trapezów	18
2.2 Wzór Simpsona	25
2.3 Wzór ’prostokątów’	33
3 Kwadratury Newtona-Cotesa dla całek podwójnych	34
3.1 Wzór trapezów	34
3.2 Wzór Simpsona	34
3.3 Wzór ’prostokątów’	34
4 Zastosowanie kwadratur do obliczania całek po obszarach normalnych	35
Podsumowanie	36
Bibliografia	38
Spis procedur w języku Maple	39

Wstęp

Treść wstępu

1. Wielomiany interpolacyjne *Lagrange'a*

W dzisiejszych czasach zaspokojenie głodu czy pragnienia nie wystarcza, by sprostać ludzkim potrzebom. Nieustanna chęć rozwoju naszego gatunku sprawia, że konieczne jest sięganie do wiedzy matematycznej w celu lepszego poznania otaczającego nas świata. Niejednokrotnie posiadamy tylko dyskretne dane pozyskane w trakcie badań i chcąc dokonać analizy matematycznej pewnych zdarzeń czy procesów musimy stworzyć/wyprowadzić regularne funkcje interpolacyjne, które w sposób ciągły i wystarczająco dokładny opisywałyby interesujące nas zjawiska. Jak możemy się spodziewać, często jest to ciężkie do wykonania zadanie.

Funkcje interpolujące w znacznym stopniu upraszczają prowadzenie obliczeń podczas wyznaczania przybliżonych wartości funkcji, których postać analityczna jest bardzo skomplikowana, lub wyliczenie kolejnych jej wartości wymaga wykonania zawiłych obliczeń komputerowych. W tym rozdziale pracy skupimy się zatem na sposobie w jaki możemy wyznaczać tak zwane wielomiany interpolacyjne *Lagrange'a* dla funkcji jednej oraz dwóch zmiennych.

1.1 Wielomiany *Lagrange'a* dla funkcji jednej zmiennej

Jak wspomnieliśmy we wstępie, często w celu uproszczenia obliczeń skłaniamy się do wykorzystania interpolacji. Stosując ją musimy liczyć się z faktem, że pozyskane wyniki są wartościami przybliżonymi, więc z reguły będą różniły się od wyników dokładnych. Przed przejściem do dalszych rozważań należy zdefiniować pojęcie interpolacji.

Definicja 1. Interpolacja

Interpolacja jest metodą numeryczną przybliżania funkcji. Polega ona na konstruowaniu tak zwanych **funkcji interpolujących** (przybliżających) $W(x)$. Wykorzystujemy do tego znane nam wartości **funkcji interpolowanej** (przybliżanej) $f(x)$, dla wybranych argumentów należących do jej dziedziny.

Wielomian $W(x)$ tworzony jest w oparciu o dwa powiązane ze sobą zbiory liczbowe [1]:

$$X = \{x_i : i = 0, 1, \dots, n\}, F = \{f_i : i = 0, 1, \dots, n\}.$$

Zbiory X i F są równoliczne. Elementy $x_i \in X$ definiują współrzędne punktów węzłowych w przestrzeni \mathbb{R}^n , natomiast elementy z F określają wartości funkcji $f(x)$ w węzłach x_i , tzn.

$$f_i = f(x_i), \quad (1.1)$$

przy czym $f_i \in \mathbb{R}$

Wyjaśnijmy również pojęcie węzła, którym przed chwilą operowaliśmy:

Definicja 2. Węzły (Punkty węzłowe)

Węzłami nazywamy punkty w przestrzeni \mathbb{R}^n , będące takimi argumentami funkcji $f(x)$, dla których jesteśmy w stanie wyznaczyć jej wartość.

W węzłach wartości funkcji interpolującej i interpolowanej są równe. Oznacza to, że:

$$W(x_i) = f(x_i) \quad (i = 0, 1, \dots, n) \quad (1.2)$$

Uwaga 1. W tym podrozdziale ograniczać będziemy się jedynie do funkcji jednej zmiennej niezależnej. Funkcje te badane będą na ograniczonym domkniętym przedziale $[a, b]$. W związku z tym, zbiór X zawierał będzie elementy $x_i \in \mathbb{R}$ takie, że $a = x_0 < x_1 < \dots < x_n = b$. Definicja zbioru F nie ulega zmianie.

Na rysunku 1.1 zobrazowaliśmy w sposób symboliczny na czym polega interpolacja, oraz czym są węzły (punkty x_i przecięcia wykresów $W(x)$ i $f(x)$, zachodzi dla nich równość (1.2)).



Rysunek 1.1: [3] Funkcja interpolacyjna $W(x)$ oraz interpolowana $f(x)$

Mając już podstawy do tego, by wiedzieć czym jest interpolacja - przejdźmy o krok dalej. Rozważmy pewien liniowo niezależny układ funkcji, zdefiniowanych na domkniętym przedziale $[a, b]$:

$$\phi_0(x), \phi_1(x), \dots, \phi_n(x) \quad (1.3)$$

oraz zbiór szukanych współczynników

$$\alpha_0, \alpha_1, \dots, \alpha_n \quad (1.4)$$

takich, że ich kombinacja liniowa z (1.3) będzie spełniała poniższy układ równań:

$$\alpha_0 \phi_0(x_i) + \alpha_1 \phi_1(x_i) + \dots + \alpha_n \phi_n(x_i) = f_i \quad (1.5)$$

gdzie $f_i \in F$ oraz $x_i \in X$.

W *interpolacji Lagrange'a* w skład (1.3) wchodzi wielomiany określone w następujący sposób:

$$l_i(x) = \prod_{j=0, j \neq i}^n \frac{x - x_j}{x_i - x_j} \quad (i = 0, 1, \dots, n). \quad (1.6)$$

Są one nazywane funkcjami bazowymi (wielomianami bazowymi) stopnia n . Warto zauważyć, że zachodzi następująca równość [3]:

$$l_i(x_j) = \delta_{ij} = \begin{cases} 1, & \text{gdy } i = j \\ 0, & \text{wp.p.} \end{cases} \quad (1.7)$$

Z powyższego wynika, że tylko w jednym przypadku $l_i(x)$ będzie miała wartość różną od 0 (gdy $x = x_i$), zatem $\sum_{i=0}^n l_i(x) = 1$. Dodatkowo zauważmy, że macierzą charakterystyczną układu (1.5) jest macierz jednostkowa. Skutkuje to tym, iż [2]

$$L_n(x) = \sum_{i=0}^n f_i \cdot l_i(x) = \sum_{i=0}^n f_i \prod_{j=0, j \neq i}^n \frac{x - x_j}{x_i - x_j} \quad (1.8)$$

będzie wielomianem o stopniu nie większym niż n , oraz przyjmie wartości (1.1) w punktach węzłowych. Wielomian (1.8) nazywany jest **wielomianem interpolacyjnym Lagrange'a**. Z jego pomocą w dosyć przystępny sposób możemy interpolować dowolną funkcję.

Przykład 1. (Wyznaczanie wielomianu interpolacyjnego Lagrange'a stopnia $n = 2$)

Zbudujemy wielomian interpolacyjny $L_2(x)$ dla funkcji $f(x) = e^{x^2}$ rozpatrywanej na ograniczonym przedziale $[0, 1]$.

W celu wyznaczenia wielomianu Lagrange'a stopnia $n = 2$, potrzebować będziemy $n + 1$ węzłów takich, że $a = x_0 < x_1 < x_2 = b$. Przyjmijmy zatem następujące ich wartości: $x_0 = 0$, $x_1 = \frac{1}{2}$, $x_2 = 1$. Korzystając bezpośrednio z równania (1.8) dostajemy:

$$L_2(x) = f(x_0) \frac{(x - x_1)(x - x_2)}{(x_0 - x_1)(x_0 - x_2)} + f(x_1) \frac{(x - x_0)(x - x_2)}{(x_1 - x_0)(x_1 - x_2)} + f(x_2) \frac{(x - x_0)(x - x_1)}{(x_2 - x_0)(x_2 - x_1)}$$

Podstawiamy kolejno wartości węzłów do równania $L_2(x)$, otrzymując wielomian interpolujący naszą funkcję $f(x) = e^{x^2}$ na $[a, b] = [0, 1]$:

$$L_2(x) = e^0 \frac{(x - \frac{1}{2})(x - 1)}{(0 - \frac{1}{2})(0 - 1)} + e^{(\frac{1}{4})} \frac{(x - 0)(x - 1)}{(\frac{1}{2} - 0)(\frac{1}{2} - 1)} + e^1 \frac{(x - 0)(x - \frac{1}{2})}{(1 - 0)(1 - \frac{1}{2})} = 2,30046x^2 - 0,58218x + 1$$

Wyprowadzona w powyższym przykładzie dla funkcji $f(x)$ i zadanych węzłów postać wielomianu $L_2(x)$ jest jednoznaczna. Mówi o tym następujące twierdzenie:

Twierdzenie 1. (O istnieniu i jednoznaczności wielomianu interpolacyjnego Lagrange'a)

Zadanie interpolacyjne Lagrange'a ma zawsze dokładnie jedno rozwiązanie, które można wyrazić wzorem (1.8). Oznacza to, że dla dowolnej funkcji $f : X \rightarrow \mathbb{R}$ oraz jej wartości $f(x_i)$ w $n + 1$ zadanych węzłach x_i ($i = 0, 1, \dots, n$), istnieje dokładnie jeden wielomian $L_n(x)$ interpolujący $f(x)$, dla którego prawdziwa jest zależność $L_n(x_i) = f(x_i)$.

Dowód.

Dowodzenie rozpoczniemy od wykazania faktu, że rozwiązanie w ogóle istnieje. Skonstruujemy je przy użyciu wielomianów bazowych *Lagrange'a* $l_i(x)$ z (1.6), oraz równania (1.7). Należy zauważyć, że odpowiadająca dla każdego z węzłów x_i , gdzie $(i = 0, 1, \dots, n)$ funkcja $l_i(x)$ jest wielomianem n -tego stopnia. W związku z tym [5]

$$L_n(x) = \sum_{i=0}^n f(x_i) \cdot l_i(x)$$

będzie wielomianem stopnia co najwyżej n . Ponadto odwołując się do (1.7) powyższy wzór możemy przekształcić do następującej postaci:

$$L_n(x_j) = \sum_{i=0}^n f(x_i) \cdot \delta_{ij} = f(x_j).$$

Wynika stąd bezpośrednio fakt, że $L_n(x)$ jest wielomianem interpolacyjnym *Lagrange'a*. Interpoluje on funkcję $f(x)$ przy zadanych węzłach x_i ($i = 0, 1, \dots, n$).

Kolejno pokażmy, że wyznaczany wielomian interpolacyjny *Lagrange'a* $L_n(x)$ jest zawsze jednoznaczny. Jeżeli zapiszemy go w postaci potęgowej (naturalnej) w bazie jednomianów $\{1, x, x^2, \dots, x^n, \dots\}$ tzn.

$$L_n(x) = \sum_{i=0}^n a_i \cdot x^i,$$

to zauważamy, że rozwiązanie naszego problemu sprowadza się do wyznaczenia $n + 1$ współczynników a_i ($i = 0, 1, \dots, n$) takich, że będą one spełniały następujący układ $n + 1$ równań liniowych:

$$\begin{bmatrix} 1 & x_0 & x_0^2 & x_0^3 & \cdots & x_0^n \\ 1 & x_1 & x_1^2 & x_1^3 & \cdots & x_1^n \\ 1 & x_2 & x_2^2 & x_2^3 & \cdots & x_2^n \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & x_n & x_n^2 & x_n^3 & \cdots & x_n^n \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ a_2 \\ \vdots \\ a_n \end{bmatrix} = \begin{bmatrix} f(x_0) \\ f(x_1) \\ f(x_2) \\ \vdots \\ f(x_n) \end{bmatrix} \quad (1.9)$$

Niech symbolicznym zapisem powyższego układu będzie $X \cdot A = F$. Macierz X jest tak zwaną macierzą *Vandermonde'a*. Zatem wiadomo, że jej wyznacznik jest $\neq 0$, przy założeniu,

że $x_0 \neq x_1 \neq \dots \neq x_{n-1} \neq x_n$. W związku z tym układ (1.9) posiada jednoznaczne rozwiązanie, co kończy dowód. \square

Jeżeli obliczylibyśmy wartość wielomianu interpolacyjnego *Lagrange'a* $L_2(x)$ (zbudowanego w przykładzie 1) dla pewnego x należącego do dziedziny $f(x)$ i porównali ją z wartością dokładną $f(x)$ dla tego samego argumentu, to zauważylibyśmy, że wynik jest obarczony pewnym błędem. Nosi on miano *błędu interpolacji*. Dla wielomianów *Lagrange'a* definiujemy go w następujący sposób:

Definicja 3. (Błąd interpolacji wielomianu *Lagrange'a*)

Weźmy

$$M_{n+1} = \sup_{x \in [a, b]} |f^{(n+1)}(x)|$$

$$m_n = \sup_{x \in [a, b]} |\omega_n(x)|,$$

przy czym $\omega_n(x) = (x - x_0) \dots (x - x_n) = \prod_{k=0}^n (x - x_k)$, zaś $f(x)$ to funkcja interpolowana.

Błędem interpolacji wielomianem *Lagrange'a* $L_n(x)$ (stopnia n) nazywamy wówczas takie

$$\delta_L = \frac{M_{n+1} \cdot m_n}{(n+1)!}, \quad (1.10)$$

dla którego zachodzi:

$$|L_n(x) - f(x)| \leq \delta_L, \quad x \in [a, b].$$

1.2 Wielomiany *Lagrange'a* dla funkcji dwóch zmiennych

W poprzednim podrozdziale rozważyliśmy wielomian interpolacyjny *Lagrange'a* funkcji jednej zmiennej niezależnej x . W rzeczywistości z pewnością często zetkniemy się z przypadkiem, gdzie jakieś analizowane przez nas zjawisko opisywane będzie przez więcej niż jedną zmienną. W związku z tym rozpatrzmy przypadek, gdzie wartość funkcji zależna jest od dwóch zmiennych niezależnych x oraz y .

Zdefiniujmy na początku trzy pomocnicze zbiory liczbowe:

$$X = \{x_i : i = 0, 1, \dots, n\} , Y = \{y_j : j = 0, 1, \dots, m\}$$

$$F = \{f_{i,j} = f(x_i, y_j) : x_i \in X , y_j \in Y\}$$

Elementy $x_i \in X$ definiują współrzędne x -owe, zaś $y_j \in Y$ współrzędne y -owe wszystkich węzłów postaci (x_i, y_j) na badanym, ograniczonym obszarze. Zbiór F zawiera natomiast wartości funkcji $f(x, y)$ we wszystkich punktach węzłowych (x_i, y_j) . Ponadto rozpatrujemy tylko takie przypadki, że $f(x, y) \in \mathbb{R}$. Zauważmy dodatkowo, że iloczyn kartezjański zbiorów $X \times Y$ jest równoliczny ze zbiorem F . W związku z tym dla każdego węzła postaci (x_i, y_j) istnieje dokładnie jedna wartość funkcji $f_{i,j} \in F$.

W naszych rozważaniach ograniczymy się do funkcji zdefiniowanych na ograniczonym obszarze prostokątnym $D = \{(x, y) | a \leq x \leq b, c \leq y \leq d\}$. Zatem zbiór X zawierał będzie elementy $x_i \in \mathbb{R}$ takie, że $a = x_0 < x_1 < \dots < x_n = b$, intuicyjnie zbiór Y elementy $y_i \in \mathbb{R}$ takie, że $c = y_0 < y_1 < \dots < y_m = d$. Definicja zbioru F nie zmienia się. Weźmy

$$f : D \rightarrow \mathbb{R}$$

taką, że możemy określić wartości $f(x_i, y_j) \in F$. Warto wspomnieć, że tak jak w przypadku wielomianu interpolacyjnego *Lagrange'a* funkcji jednej zmiennej, wartości funkcji interpolującej i interpolowanej w węzłach również muszą być sobie równe, tzn.:

$$L_{n,m}(x_i, y_j) = f(x_i, y_j), \quad (i = 0, 1, \dots, n ; j = 0, 1, \dots, m) \quad (1.11)$$

(Przez zastosowane oznaczenie $L_{n,m}(x, y)$ będziemy rozumieli wielomian interpolacyjny *Lagrange'a* funkcji dwóch zmiennych, gdzie stopień x wynosi n , zaś stopień y jest równy m .) W skład wielomianu interpolacyjnego $L_{n,m}(x, y)$ wchodziły będą dwie funkcje bazowe (lub inaczej - wielomiany bazowe), po jednej dla każdej zmiennej niezależnej.

Dla x wielomian bazowy stopnia n ma postać [8] :

$$l_i(x) = \prod_{s=0, s \neq i}^n \frac{x - x_s}{x_i - x_s} \quad (i = 0, 1, \dots, n). \quad (1.12)$$

Natomiast dla zmiennej y wielomianem bazowym stopnia m jest :

$$l_j(y) = \prod_{s=0, s \neq j}^m \frac{y - y_s}{y_j - y_s} \quad (j = 0, 1, \dots, m). \quad (1.13)$$

Powyższe funkcje bazowe (1.12) i (1.13) charakteryzują następujące własności:

$$l_i(x_s) = \delta_{is} = \begin{cases} 1, & \text{gdy } i = s \\ 0, & \text{wp.p.} \end{cases} \quad (1.14)$$

$$l_j(y_s) = \delta_{js} = \begin{cases} 1, & \text{gdy } j = s \\ 0, & \text{wp.p.} \end{cases} \quad (1.15)$$

Wzoruując się na równaniu (1.8) oraz korzystając z (1.12) i (1.13) jesteśmy w stanie wyprowadzić wzór na wielomian interpolacyjny *Lagrange'a* dla funkcji dwóch zmiennych. W związku z tym, jego postać jest następująca:

$$L_{n,m}(x, y) = \sum_{i=0}^n \sum_{j=0}^m f(x_i, y_j) \cdot \left(\prod_{s=0, s \neq i}^n \frac{x - x_s}{x_i - x_s} \cdot \prod_{s=0, s \neq j}^m \frac{y - y_s}{y_j - y_s} \right) \quad (1.16)$$

Z faktu, że pewna znana nam funkcja $f(x, y)$ zostaje przez nas interpolowana przy użyciu innej funkcji, wynika to, że niemal zawsze wartości $f(x_i, y_j)$ i $L_{n,m}(x_i, y_j)$ będą w pewnym stopniu od siebie odbiegać (pomijając punkty węzłowe dla których zachodzi (1.11)). Różnica w tychże wartościach ma miejsce, gdyż jest to spowodowane wystąpieniem błędu interpolacji. Maksymalną wielkość takiego błędu możemy określić z góry - sposób jego wyznaczania precyzuje definicja poniżej.

Definicja 4. (Błąd interpolacji wielomianem *Lagrange'a* dla funkcji dwóch zmiennych)

Niech dane będą następujące równości [9]:

$$M = \sup_{\substack{x \in [a, b] \\ y \in [c, d]}} \left| \frac{\partial^{n+m+2} f(x, y)}{\partial x^{n+1} \partial y^{m+1}} \right|$$

$$m_1 = \sup_{x \in [a, b]} |\omega_n(x)|$$

$$m_2 = \sup_{y \in [c, d]} |\omega_m(y)|$$

Przy czym $\omega_n(x) = \prod_{i=0}^n (x - x_i)$ oraz $\omega_m(y) = \prod_{j=0}^m (y - y_j)$, natomiast $f(x, y)$ to funkcja interpolowana. Błędem interpolacji wielomianem *Lagrange'a* $L_{n,m}(x, y)$ nazywamy takie

$$\delta_{L_{i,j}} = \frac{M \cdot m_1 \cdot m_2}{(n+1)! \cdot (m+1)!}, \quad (1.17)$$

dla którego prawdziwa jest następująca nierówność:

$$|L_{n,m}(x, y) - f(x, y)| \leq \delta_{L_{i,j}} \quad x \in [a, b], \quad y \in [c, d] \quad (1.18)$$

W celu prezentacji metody budowania wielomianu interpolacyjnego *Lagrange'a* dla funkcji dwóch zmiennych rozwiążmy pewien przykład, który przybliży nam to zagadnienie.

Przykład 2. (Wyznaczanie wielomianu interpolacyjnego *Lagrange'a* $L_{1,2}(x, y)$)

Stworzymy wielomian interpolacyjny *Lagrange'a* $L_{1,2}(x, y)$ dla funkcji $f(x, y) = \cos(x^2 + y^2)$ rozważanej na domkniętym obszarze prostokątnym postaci $\Omega = [0, \frac{1}{2}] \times [0, 1]$.

Chcąc wyznaczyć $L_{1,2}(x, y)$ niezbędne jest określenie $n+1 = 1+1 = 2$ wartości $x \in \Omega$ oraz $m+1 = 2+1 = 3$ wartości $y \in \Omega$. Ważne jest, by krańcowe x i y zawierały brzegi obszaru Ω . Zatem niech:

$$x_0 = 0, \quad x_1 = \frac{1}{2}$$

$$y_0 = 0, \quad y_1 = \frac{1}{2}, \quad y_2 = 1.$$

Kolejno korzystając bezpośrednio z (1.16) dostajemy:

$$\begin{aligned} L_{1,2}(x, y) &= \sum_{i=0}^1 \sum_{j=0}^2 f(x_i, y_j) \cdot \left(\prod_{s=0, s \neq i}^1 \frac{x - x_s}{x_i - x_s} \cdot \prod_{s=0, s \neq j}^2 \frac{y - y_s}{y_j - y_s} \right) = \\ &= f(x_0, y_0) \left(\frac{x - x_1}{x_0 - x_1} \cdot \frac{y - y_1}{y_0 - y_1} \cdot \frac{y - y_2}{y_0 - y_2} \right) + f(x_0, y_1) \left(\frac{x - x_1}{x_0 - x_1} \cdot \frac{y - y_0}{y_1 - y_0} \cdot \frac{y - y_2}{y_1 - y_2} \right) + \\ &+ f(x_0, y_2) \left(\frac{x - x_1}{x_0 - x_1} \cdot \frac{y - y_0}{y_2 - y_0} \cdot \frac{y - y_1}{y_2 - y_1} \right) + f(x_1, y_0) \left(\frac{x - x_0}{x_1 - x_0} \cdot \frac{y - y_1}{y_0 - y_1} \cdot \frac{y - y_2}{y_0 - y_2} \right) + \\ &+ f(x_1, y_1) \left(\frac{x - x_0}{x_1 - x_0} \cdot \frac{y - y_0}{y_1 - y_0} \cdot \frac{y - y_2}{y_1 - y_2} \right) + f(x_1, y_2) \left(\frac{x - x_0}{x_1 - x_0} \cdot \frac{y - y_0}{y_2 - y_0} \cdot \frac{y - y_1}{y_2 - y_1} \right) \end{aligned}$$

$$f(x_1, y_1) \left(\frac{x - x_0}{x_1 - x_0} \cdot \frac{y - y_0}{y_1 - y_0} \cdot \frac{y - y_2}{y_1 - y_2} \right) + f(x_1, y_2) \left(\frac{x - x_0}{x_1 - x_0} \cdot \frac{y - y_0}{y_2 - y_0} \cdot \frac{y - y_1}{y_2 - y_1} \right)$$

Wstawiając do powyższego $L_{1,2}(x, y)$, określone na początku wartości x i y otrzymujemy:

$$\begin{aligned} L_{1,2}(x, y) &= \cos(0^2 + 0^2) \left(\frac{x - \frac{1}{2}}{0 - \frac{1}{2}} \cdot \frac{y - \frac{1}{2}}{0 - \frac{1}{2}} \cdot \frac{y - 1}{0 - 1} \right) + \cos(0^2 + \left(\frac{1}{2}\right)^2) \left(\frac{x - \frac{1}{2}}{0 - \frac{1}{2}} \cdot \frac{y - 0}{\frac{1}{2} - 0} \cdot \frac{y - 1}{\frac{1}{2} - 1} \right) + \\ &\quad \cos(0^2 + 1^2) \left(\frac{x - \frac{1}{2}}{0 - \frac{1}{2}} \cdot \frac{y - 0}{1 - 0} \cdot \frac{y - \frac{1}{2}}{1 - \frac{1}{2}} \right) + \cos\left(\left(\frac{1}{2}\right)^2 + 0^2\right) \left(\frac{x - 0}{\frac{1}{2} - 0} \cdot \frac{y - \frac{1}{2}}{0 - \frac{1}{2}} \cdot \frac{y - 1}{0 - 1} \right) + \\ &\quad \cos\left(\left(\frac{1}{2}\right)^2 + \left(\frac{1}{2}\right)^2\right) \left(\frac{x - 0}{\frac{1}{2} - 0} \cdot \frac{y - 0}{\frac{1}{2} - 0} \cdot \frac{y - 1}{\frac{1}{2} - 1} \right) + \cos\left(\left(\frac{1}{2}\right)^2 + 1^2\right) \left(\frac{x - 0}{\frac{1}{2} - 0} \cdot \frac{y - 0}{1 - 0} \cdot \frac{y - \frac{1}{2}}{1 - \frac{1}{2}} \right) = \dots \\ &= 1 - 0,0621752x + 0,335347y - 0,0941494xy - 0,795045y^2 - 0,29363xy^2 \end{aligned}$$

Jak widać na powyższym przykładzie - wszystkie obliczenia niezbędne do otrzymania ostatecznej formy wielomianu interpolacyjnego *Lagrange'a* dla funkcji dwóch zmiennych są niezwykle pracochłonne. Sprawdźmy jeszcze jak bardzo wartości $L_{1,2}(x, y)$ odbiegają od wartości dokładnych $f(x, y) = \cos(x^2 + y^2)$ w $z_0 = (\frac{1}{4}, \frac{1}{2})$ oraz $z_1 = (\frac{1}{3}, \frac{2}{3})$. Więc:

- w punkcie z_0 :

$$\begin{aligned} f\left(\frac{1}{4}, \frac{1}{2}\right) &= \cos\left(\left(\frac{1}{4}\right)^2 + \left(\frac{1}{2}\right)^2\right) = 0,951567948 \\ L_{1,2}\left(\frac{1}{4}, \frac{1}{2}\right) &= 1 - 0,0621752\left(\frac{1}{4}\right) + 0,335347\left(\frac{1}{2}\right) - 0,0941494\left(\frac{1}{4}\right)\left(\frac{1}{2}\right) \\ &\quad - 0,795045\left(\frac{1}{2}\right)^2 - 0,29363\left(\frac{1}{4}\right)\left(\frac{1}{2}\right)^2 = 0,9232479 \end{aligned}$$

- w punkcie z_1 :

$$\begin{aligned} f\left(\frac{1}{3}, \frac{2}{3}\right) &= \cos\left(\left(\frac{1}{3}\right)^2 + \left(\frac{2}{3}\right)^2\right) = 0,849607562 \\ L_{1,2}\left(\frac{1}{3}, \frac{2}{3}\right) &= 1 - 0,0621752\left(\frac{1}{3}\right) + 0,335347\left(\frac{2}{3}\right) - 0,0941494\left(\frac{1}{3}\right)\left(\frac{2}{3}\right) \\ &\quad - 0,795045\left(\frac{2}{3}\right)^2 - 0,29363\left(\frac{1}{3}\right)\left(\frac{2}{3}\right)^2 = 0,785063437 \end{aligned}$$

Na podstawie zaprezentowanych rachunków można zauważyć, że nasze wyniki w pewnym, niezbyt znaczącym stopniu różnią się od siebie. Poprzez zwiększenie liczby węzłów, możliwe jest zmniejszenie błędu interpolacji jakim zostały one obarczone. Jednakże należy pamiętać, że każdy dodatkowy podział obszaru Ω znacząco wydłużyłoby obliczenia, dlatego też nasze rachunki zawężiliśmy tylko do sześciu punktów węzłowych.

2. Kwadratury Newtona-Cotesa dla całki pojedynczej

Niech $f(x)$ będzie funkcją zdefiniowaną na przedziale $[a, b]$ o wartościach rzeczywistych tzn. $f : [a, b] \rightarrow \mathbb{R}$. Rozważmy pewną całkę [4]

$$I(f) = \int_a^b f(x) dx \quad (2.1)$$

Funkcję podcałkową zawsze możemy zastąpić inną funkcją taką, że w miarę możliwości poniższe przybliżenie będzie prawdziwe:

$$\int_a^b f(x) dx \approx \int_a^b g(x) dx \quad (2.2)$$

W praktyce często spotkać możemy się z przypadkiem takim, że do wyznaczenia przybliżonych wartości $I(f)$ stosowane są wzory nazywane kwadraturami. Owe kwadratury opierają się na wartościach $f(x)$ w punktach węzłowych i mogą niezbyt dokładnie przybliżać wynik tzn:

$$\int_a^b f(x) dx \approx \sum_{i=0}^n a_i f(x_i) = S(f), \quad x_i \in [a, b], \quad (2.3)$$

przy czym współczynniki a_i są niezależne od $f(x)$ (nazywamy je współczynnikami wagowymi kwadratury i określają one wielkość udziału $f(x_i)$ w wartości całej sumy $S(f)$), zaś x_i nosi miano węzłów kwadratury.

Naszym celem jest jednak to, by jak najbardziej zminimalizować błąd pojawiający się podczas przybliżania wartości $I(f)$. W związku z tym możemy zastosować zabieg zastąpienia funkcji $f(x)$ w całce $I(f)$ wielomianem interpolującym ją. W tym celu wykorzystamy wielomian interpolacyjny *Lagrange'a* (1.8). Po podstawieniu go do (2.3) otrzymamy [4][7]

$$I(f) = \int_a^b f(x) dx \approx \int_a^b L_n(x) dx = \sum_{i=0}^n a_i f(x_i) = S(f), \quad (2.4)$$

gdzie

$$a_i = \int_a^b l_i(x), \quad (2.5)$$

natomiast

$$l_i(x) = \prod_{j=0, j \neq i}^n \frac{x - x_j}{x_i - x_j} \quad (i = 0, 1, \dots, n). \quad (2.6)$$

W związku z powyższym, całkę (2.1) możemy wyrazić w następujący sposób:

$$\int_a^b f(x) dx \approx \sum_{i=0}^n f(x_i) \int_a^b l_i(x) dx \quad (2.7)$$

Jeżeli w (2.7) rozpatrzmy tylko węzły takie, że $x_0 = a$, $x_n = b$, a każdy węzeł pośredni leżący pomiędzy x_0 a x_n jest postaci $x_i = a + ih$ ($i = 0, 1, \dots, n$), $h = \frac{x_n - x_0}{n}$, to kwadraturę taką nazwiemy kwadraturą **Newtona-Cotesa** [2]. Skupimy się na rozważeniu trzech różnych przypadków, dla których kolejne węzły są równo odległe - będą to: *wzór trapezów*, *wzór Simpsona* oraz dodatkowo *wzór prostokątów*.

2.1 Wzór trapezów

Do wyznaczenia wzoru trapezów będziemy wykorzystywać wielomian interpolacyjny *Lagrange'a* rzędu $n = 1$ ($L_1(x)$) utworzony dla węzłów a i b . Zastosujemy w (2.6) następujące podstawienie: $x = a + hs$. Wartości a, h są pewnymi stałymi [7], natomiast s jest zmienną niezależną.

$$l_i(x) = \prod_{j=0, j \neq i}^n \frac{x - x_j}{x_i - x_j} = \prod_{j=0, j \neq i}^n \frac{a + hs - (a + jh)}{(a + ih) - (a + jh)} = \prod_{j=0, j \neq i}^n \frac{s - j}{i - j}$$

Otrzymujemy zatem:

$$l_i(x) = \prod_{j=0, j \neq i}^n \frac{s - j}{i - j} = \psi_i(s) \quad (2.8)$$

$\psi_i(s)$ to nasza nowa funkcja zmiennej s .

Całkę z równania (2.5) obliczymy metodą całkowania przez podstawienie. Skorzystamy z przedstawionego przed chwilą podstawienia $x = a + hs$ oraz równania (2.8)

$$a_i = \int_a^b l_i(x) dx = \left\{ \begin{array}{l} x = a + hs \\ dx = hds \\ b = a + hs \Rightarrow \frac{b-a}{h} = s \Rightarrow s = n \\ a = a + hs \Rightarrow 0 = hs \Rightarrow s = 0 \end{array} \right\} = h \int_0^n \psi_i(s) ds \quad (2.9)$$

Chcemy wyliczyć teraz wartości współczynników kwadratury a_0 i a_1 . Pamiętamy o tym, że do obliczeń wykorzystujemy postać wielomianu *Lagrange'a* rzędu $n = 1$, zatem będziemy całkowali [7] w granicach $[0, n] = [0, 1]$

$$a_0 = h \int_0^1 \psi_0(s) ds = h \int_0^1 \frac{s-1}{0-1} ds = \int_0^1 (1-s) ds = h \left[s - \frac{s^2}{2} \right]_0^1 = \frac{h}{2}$$

$$a_1 = h \int_0^1 \psi_1(s) ds = h \int_0^1 \frac{s-0}{1-0} ds = \int_0^1 (s) ds = h \left[\frac{s^2}{2} \right]_0^1 = \frac{h}{2}$$

Po wstawieniu wyliczonych współczynników do (2.3) otrzymujemy

$$\begin{aligned} \int_a^b f(x) dx &\approx \sum_{i=0}^1 a_i f(x_i) = a_0 f(x_0) + a_1 f(x_1) = \\ &= \frac{h}{2} f(x_0) + \frac{h}{2} f(x_1) = \frac{h}{2} (f(x_0) + f(x_1)) \end{aligned} \quad (2.10)$$

Wzór ten nazywamy **wzorem trapezów**. Zauważmy, że suma współczynników a_0, a_1 jest równa $h \cdot n$. W późniejszych podrozdziałach również zetkniemy się z taką prawidłowością rozpatrując wyższy rząd wielomianów *Lagrange'a* dla $n > 1$.

Podczas wyznaczania (2.10) przyjęliśmy, że przedział całkowania nie został podzielony, a jedynymi węzłami były jego początek i koniec. W rzeczywistości rozpatrujemy przypadki z wielokrotnym podziałem przedziału. Jeżeli przedział całkowania $[a, b]$ podzielimy na ≥ 2 równe części takie, że $a = x_0 < x_1 < \dots < x_n = b$, to otrzymamy *uogólniony (złożony) wzór trapezów*:

$$\begin{aligned} \int_a^b f(x) dx &= \sum_{i=0}^{n-1} \int_{x_i}^{x_{i+1}} f(x) dx \approx \\ &\approx \left(\frac{h}{2} (f(x_0) + f(x_1)) + \dots + \frac{h}{2} (f(x_{n-1}) + f(x_n)) \right) = \\ &= \frac{h}{2} (f(x_0) + 2f(x_1) + \dots + 2f(x_{n-1}) + f(x_n)) = \\ &= \frac{h}{2} \sum_{i=0}^{n-1} [f(x_i) + f(x_{i+1})] \end{aligned} \quad (2.11)$$

W wyprowadzonym wzorze trapezów (2.11) (mówiąc wzór trapezów będziemy mieli na myśli jego postać złożoną) przybliżamy wartość całki sumując pola trapezów o ustalonej wysokości h (odległość pomiędzy kolejnymi węzłami) i podstawach o długości $f(x_i)$ i $f(x_{i+1})$ dla $i = 0, 1, \dots, n-1$. Rysunek 2.1 z pewnością pomoże nam lepiej zrozumieć to zagadnienie.



Rysunek 2.1: Graficzne przedstawienie idei zastosowania wzoru trapezów do całkowania

Przejdźmy do praktycznego zastosowania (2.11) w celu pokazania jego przydatności. Obliczenia dla zadanej funkcji przeprowadzimy ręcznie - przy relatywnie małej liczbie podziału przedziałów, oraz wykorzystując procedurę stworzoną w języku *Maple* dla dużo większej liczby podziałów.

Przykład 3. (Wyznaczanie przybliżonej wartości całki przy wykorzystaniu wzoru trapezów) Wyznamy przybliżoną wartość całki $\int_0^{\frac{3}{2}} e^{x^2} dx$ przyjmując, że przedział całkowania dzielimy na $n=6$ równych części (7 węzłów).

Do rozwiązania zadania skorzystamy z ogólnego wzoru trapezów. Wyznamy wartości węzłów oraz odległość $h = \frac{b-a}{n}$ pomiędzy kolejnymi węzłami:

$$h = \frac{\frac{3}{2}-0}{6} = \frac{1}{4}, x_0 = 0, x_1 = \frac{1}{4}, x_2 = \frac{2}{4}, x_3 = \frac{3}{4}, x_4 = 1, x_5 = \frac{5}{4}, x_6 = \frac{6}{4}.$$

Wyliczone wartości wstawiamy do (2.11):

$$\begin{aligned} \int_0^{\frac{3}{2}} e^{x^2} dx &= \frac{1}{2} (f(x_0) + 2f(x_1) + 2f(x_2) + 2f(x_3) + 2f(x_4) + 2f(x_5) + f(x_6)) = \\ &= \frac{1}{8} (e^0 + 2e^{(\frac{1}{4})^2} + 2e^{(\frac{2}{4})^2} + 2e^{(\frac{3}{4})^2} + 2e^1 + 2e^{(\frac{5}{4})^2} + e^{(\frac{6}{4})^2}) \approx \\ &\frac{1}{8} (1 + 2 \cdot 1,06449 + 2 \cdot 1,28402 + 2 \cdot 1,75505 + 2 \cdot 2,71828 + 2 \cdot 4,77073 + 9,48773) \approx 4,20911 \end{aligned}$$

Wartość dokładna wynosi w zaokrągleniu 4,06311405862. Jak widać różnica w wyniku jest dosyć znacząca przy ustalonych wartościach węzłów, ponieważ otrzymany przez nas wynik to 4,20911.

Z reguły im więcej podziałów przedziału wykonamy, tym dokładniejsze powinniśmy otrzymać przybliżenie. Można wnioskować w ten sposób również poddając się na analizie wzoru na błąd metody trapezów (im większy n tym mniejsza wartość δ_T):

Definicja 5. (Błąd przybliżenia wzorem trapezów)

Przyjmijmy, że BBT_n oznacza błąd bezwzględny przybliżenia (różnica pomiędzy wartością dokładną i przybliżoną całki). Ponadto niech $f : [a, b] \rightarrow \mathbb{R}$ będzie klasy C^2 na $[a, b]$, oraz n wyznacza ilość podziałów tego przedziału, wtedy spełniona jest następująca nierówność [6]

$$|BBT_n| \leq \delta_T \quad (2.12)$$

Symbol δ_T oznacza błąd metody trapezów i wyraża go następujący wzór:

$$\delta_T = \frac{(b-a)^3}{12n^2} \cdot M_2, \quad (2.13)$$

gdzie

$$M_2 = \sup_{x \in [a, b]} |f''(x)| \quad (2.14)$$

Sprawdzimy jednak słuszność naszego twierdzenia wykorzystując do tego stworzoną na nasze potrzeby procedurę w języku Maple. Posłuży nam ona do zautomatyzowania procesu obliczania przybliżonych wartości całek oznaczonych metodą trapezów:

Procedura 1. (Wyznaczanie przybliżonych wartości całek pojedynczych metodą trapezów)

```
> trapez:=proc(a,b,n)
local h,j,i,x,y,t,tp;
if (((ceil(n)>n) or (floor(n)<n)) or n<=0)
then ERROR('Ostatni parametr musi być liczba naturalna')else
h:=(b-a)/n;
for j from 0 to n do x:= j -> a+j*h od;
for j from 0 to n do y:= j -> f(x(j)) od;
t:= h/2 * sum(y(i)+y(i+1), i=0..n-1);
tp:=evalf(t,12);
fi;
end;
```

Przed przejściem do prezentacji wyników jakie zwróci powyższa procedura, należy wyjaśnić nieco jej zasadę działania i znaczenie parametrów.

Pierwsza linia jest deklaracją procedury *trapez* z następującymi parametrami wejściowymi:

a, b - są odpowiednio dolnym i górnym krańcem przedziału całkowania

n - określa liczbę podprzedziałów na które podzielony zostanie nasz przedział $[a, b]$

Kolejno deklarujemy zmienne pomocnicze z których korzysta procedura (nie pobierane od użytkownika). Jeżeli ostatni parametr $n \in \mathbb{N}$ to zwracany jest stosowny komunikat i procedura kończy swoje działanie, w p.p. wykonuje się główny algorytm tzn. obliczamy wartość kroku h , czyli odległość pomiędzy dwoma kolejnymi węzłami. Następnie wyznaczamy wszystkie węzły x_i na danym przedziale oraz wyliczamy wartości (deklarowanej przed wywołaniem procedury) funkcji f w tychże węzłach. Przedostatni krok polega na zsumowaniu wyliczonych przed chwilą wartości w sposób zgodny ze wzorem (2.11). Algorytm kończymy poprzez przekształcenie sumy do "przyjaznej" postaci i prezentację wyniku w formacie 12 znakowym.

Poniższa tabela prezentuje wyniki procedury 1 (ozn. W_{trapez}) dla coraz to większej liczby podprzedziałów (ozn. n). Uwzględniono w niej również wartości błędów wzoru trapezów (ozn. δ_T) wyliczone zgodnie ze wzorem (2.13) oraz moduł różnicy pomiędzy wartością dokładną całki a otrzymaną w ramach testów procedury (ozn. $|BBT_n|$, zgodnie z (2.12)). Do kalkulacji wykorzystamy funkcję z przykładu 3 oraz podaną w rozwiązaniu informację na temat przybliżonej do 11 miejsc po przecinku wartości dokładnej całki $\int_0^{\frac{3}{2}} e^{x^2} dx \approx 4,06311405862$.

n	W_{trapez}	BBT_n	δ_T
6	4,20911436529	0,146000306	0,815352298
30	4,06904019209	0,005926133	0,032614091
100	4,06364771371	0,000533655	0,002935268
200	4,06324747807	0,000133419	0,000733817
400	4,06314741384	0,000033355	0,000183454
800	4,06312239755	0,000008339	0,000045863
2000	4,06311539304	0,000001334	0,000007338
5000	4,06311427243	0,000000213	0,000001174
7310	4,06311415812	0,000000099	0,000000549
10000	4,06311411177	0,000000053	0,000000293
30000	4,06311406426	0,000000005	0,000000032
50000	4,06311406037	0,000000002	0,000000011
70000	4,06311406006	0,000000001	0,000000005

Tablica 2.1: Wyniki pomiaru dokładności przybliżania wartości całki wzorem trapezów

Na podstawie powyższych pomiarów możemy uznać iż nasze przewidywania odnośnie tego, że w miarę wzrostu wartości n wynik będzie coraz dokładniejszy okazują się być trafne. Procedura generuje prawidłowe wyniki, co można potwierdzić tym, że dla $n = 6$ wartość obliczona "ręcznie" jest taka sama jak otrzymana w trakcie badań. Zauważmy, że już przy $n \geq 100$ wynik zaczyna być dosyć mocno zbliżony do dokładnego - różnice pojawiają się dopiero na 4 miejscu po przecinku. Uwagę należy skupić również na wartościach $|\text{BBT}_n|$ i δ_T z tablicy 2.1. Po ich dokładnej analizie jesteśmy w stanie zauważyć, że $|\text{BBT}_n|$ jest zawsze większe od δ_T . Nie ma znaczenia to, jaki n przyjęliśmy. Nie jest to jednak przypadkowa zależność. Fakt ten w pełni wyjaśnia nam definicja 2.13. Wartość δ_T została wyznaczona przy wykorzystaniu następującego skryptu:

Procedura 2. (Wyznaczanie błędu przybliżenia dla metody trapezów)

```
> trapezBlad:= proc (a, b, n)
local d2ff, simd2ff, M2, deltaT;
if (((ceil(n)>n) or (floor(n)<n)) or n<=0)
then ERROR('Ostatni parametr musi byc liczba naturalna')
else
d2ff:= diff(diff(f(x), x), x);
simd2ff:= simplify(d2ff);
M2:= (Optimization[Maximize])(abs(simd2ff), x = a .. b);
deltaT:= evalf(1/12*M2[1]*(b-a)^3/n^2)
fi;
end;
```

Procedura ta opiera się na wzorach (2.13) oraz (2.14). Wynikiem jej działania jest wartość błędu δ_T dla całki z f (funkcja musi zostać zdefiniowana przed wywołaniem *trapezBlad*) liczonej w granicach $[a, b]$, i dokonanych n podziałów tego przedziału. Parametry wejściowe są definiowane tak samo jak dla procedury 1. Dokładny opis algorytmu dla tego przypadku zostaje pominięty.

Często przeprowadzając obliczenia naukowe musimy zadbać o to, by wynik jaki otrzymamy był jak najdokładniejszy i obciążony co najwyżej pewnym błędem, nie większym

niż dopuszczalny. Również wykorzystując wzór trapezów do całkowania numerycznego, możemy kontrolować wielkość błędu, jaki pojawia się podczas przybliżania wyniku. Sposobem na to jest odpowiedni dobór liczby podziałów przedziału całkowania. Poniższa procedura wskazuje nam, ile co najmniej owych podziałów powinniśmy dokonać, by wielkość błędu nie przekraczała zdefiniowanej przez nas wartości.

Procedura 3. (Wyznaczanie najmniejszej liczby n podziałów przedziału całkowania tak, by wynik był obciążony błędem nie większym niż zadany)

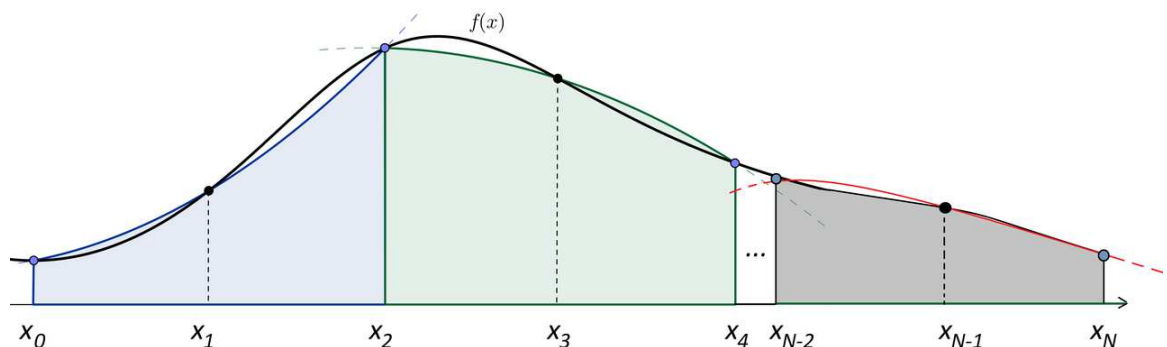
```
> trapeznMin:= proc (a, b, blad)
local d2ff, simd2ff, M2, nMin, p;
d2ff:= diff(diff(f(x), x), x);
simd2ff:= simplify(d2ff);
M2:= (Optimization[Maximize])(abs(simd2ff), x = a .. b);
p:= solve*({1/12*(b-a)^3*M2[1]/n^2 < blad, 0 < n}, {n});
nMin := p[1]
end;
```

Powyższa procedura również korzysta ze wzorów (2.13) i (2.14). Jej parametry wejściowe a i b są tak jak poprzednio granicami całkowania pewnej funkcji f , natomiast $blad$ określa jaki błąd przybliżenia jest przez nas maksymalnie akceptowalny. Kod znajdujący się wewnątrz procedury kolejno oblicza $f''(x)$ i upraszcza postać wyliczonej pochodnej. Następnie szukana jest wartość $M_2 = \sup_{x \in [a,b]} |f''(x)|$. Najważniejszym elementem składowym *trapeznMin* jest jednak wyliczanie wartości dla zmiennej p . By ją wyznaczyć rozwiązujemy nierówność $\frac{(b-a)^3}{12n^2} \cdot M_2 < blad$ ze względu na n . Zakładamy, że interesują nas tylko $n > 0$ (n jest liczbą podziałów i musi być wartością całkowitą dodatnią). Ostatnia linia odpowiada za prezentację obliczonej wartości n . Wynik pojawia się w postaci $N > z$, gdzie z jest liczbą całkowitą - zatem jako wynik traktować będziemy najmniejszą liczbę naturalną większą od z .

2.2 Wzór Simpsona

Często wykorzystywaną przy całkowaniu numerycznym metodą jest *metoda Simpsona*. Zamiennie nazywana jest również *metodą parabol*. Jej nazwa wywodzi się od tego, że funkcję podcałkową przybliżamy na każdym z podprzedziałów utworzonych przez **3** węzły za pomocą paraboli. W związku z tym, że każde 3 kolejne węzły tworzą **2** podprzedziały wnioskujemy, że całkowita liczba podprzedziałów przedziału całkowania jest postaci $N = 2m$ dla $m \in \mathbb{N}$ (we wzorze trapezów nie było wymagane, by liczba podprzedziałów była parzysta).

Przybliżonym wynikiem całki obliczanej metodą *Simpsona* jest suma pól tzw. trapezów "parabolicznych" (jedno z ramion trapezu jest parabolą). Każdy z nich jest ograniczany przez węzły x_{2i} i x_{2i+2} ($i = 0, 1, \dots, m-1$) na osi OX , natomiast na osi OY ogranicza je prosta $y = 0$ oraz parabola przechodząca przez $x_{2i}, x_{2i+1}, x_{2i+2}$ ($i = 0, 1, \dots, m-1$). Należy zauważyć, że istnieje dokładnie jedna parabola zawierająca zadane trzy węzły, więc jej równanie zawsze będzie wyznaczone w sposób jednoznaczny. Metoda całkowania przy wykorzystaniu wzoru *Simpsona* została przedstawiona na rysunku 2.2.



Rysunek 2.2: Graficzne przedstawienie idei zastosowania wzoru parabol do całkowania

Rozpatrzmy początkowo szczególny przypadek, w którym przedział całkowania zostanie podzielony na dokładnie 2 podprzedziały, wydzielone przez 3 równo odległe węzły $a = x_0 < x_1 < x_2 = b$, dla których zachodzi $\frac{x_0+x_2}{2} = x_1$. W celu wyprowadzenia wzoru *Simpsona* wykorzystamy wielomian interpolacyjny *Lagrange'a* $L_2(x)$ (wielomian rzędu $n = 2$) zbudowany zgodnie z (1.8) dla węzłów $a = x_0, x_1, x_2 = b$. By stworzyć $L_2(x)$, który nas

interesuje, potrzebna nam jest wiedza na temat tego, jak wyglądają współczynniki a_0, a_1, a_2 . Ich wartość wyliczymy wykorzystując wzory (2.8) oraz (2.9). Wynika z nich, że każdy ze współczynników a_i ($i = 0, 1, \dots, n$) jest postaci:

$$a_i = h \int_0^n \psi_i(s) ds = h \int_0^n \left(\prod_{j=0, j \neq i}^n \frac{s-j}{i-j} \right) ds. \quad (2.15)$$

W tym przypadku za n przyjmujemy rząd wielomianu interpolacyjnego $L_2(x)$, zatem całkowanie odbędzie się w granicach $[0, n] = [0, 2]$.

$$\begin{aligned} \mathbf{a}_0 &= h \int_0^2 \psi_0(s) ds = h \int_0^2 \left(\prod_{j=0, j \neq 0}^2 \frac{s-j}{0-j} \right) ds = h \int_0^2 \left(\frac{(s-1)}{(0-1)} \cdot \frac{(s-2)}{(0-2)} \right) ds = h \int_0^2 \frac{(s^2 - 2s - s + 2)}{2} ds = \\ &= \frac{h}{2} \int_0^2 (s^2 - 3s + 2) ds = \frac{h}{2} \left[\frac{s^3}{3} - \frac{3s^2}{2} + 2s \right]_0^2 = \frac{h}{2} \left(\left[\frac{8}{3} - \frac{12}{2} + 4 \right] - 0 \right) = \frac{h}{2} \cdot \frac{2}{3} = \frac{\mathbf{h}}{\mathbf{3}} \end{aligned}$$

$$\begin{aligned} \mathbf{a}_1 &= h \int_0^2 \psi_1(s) ds = h \int_0^2 \left(\prod_{j=0, j \neq 1}^2 \frac{s-j}{1-j} \right) ds = h \int_0^2 \left(\frac{(s-0)}{(1-0)} \cdot \frac{(s-2)}{(1-2)} \right) ds = h \int_0^2 \frac{(s^2 - 2s)}{-1} ds = \\ &= h \int_0^2 (2s - s^2) ds = h \left[2\frac{s^2}{2} - \frac{s^3}{3} \right]_0^2 = h \left(\left[4 - \frac{8}{3} \right] - 0 \right) = h \cdot \frac{4}{3} = \frac{\mathbf{4h}}{\mathbf{3}} \end{aligned}$$

$$\begin{aligned} \mathbf{a}_2 &= h \int_0^2 \psi_2(s) ds = h \int_0^2 \left(\prod_{j=0, j \neq 2}^2 \frac{s-j}{2-j} \right) ds = h \int_0^2 \left(\frac{(s-0)}{(2-0)} \cdot \frac{(s-1)}{(2-1)} \right) ds = h \int_0^2 \frac{s(s-1)}{2} ds = \\ &= \frac{h}{2} \int_0^2 (s^2 - s) ds = \frac{h}{2} \left[\frac{s^3}{3} - \frac{s^2}{2} \right]_0^2 = \frac{h}{2} \left(\left[\frac{8}{3} - \frac{4}{2} \right] - 0 \right) = \frac{h}{2} \cdot \frac{2}{3} = \frac{\mathbf{h}}{\mathbf{3}} \end{aligned}$$

Podstawiając obliczone powyżej wartości a_0, a_1, a_2 do (2.3) otrzymujemy szukany przez nas **wzór Simpsona**:

$$\begin{aligned} \int_a^b f(x) dx &\approx \sum_{i=0}^2 a_i f(x_i) = a_0 f(x_0) + a_1 f(x_1) + a_2 f(x_2) = \\ &= \frac{h}{3} f(x_0) + \frac{4h}{3} f(x_1) + \frac{h}{3} f(x_2) = \frac{h}{3} (f(x_0) + 4f(x_1) + f(x_2)). \end{aligned} \quad (2.16)$$

Zauważmy, że dla tego przypadku zachodzi również prawidłowość związana z wartością sumy współczynników, adekwatna do tej ze wzoru trapezów: $a_0 + a_1 + a_2 = \frac{h}{2} + \frac{4h}{3} + \frac{h}{3} = n \cdot h = 2 \cdot h$. Możemy zatem wnioskować, że powyższe obliczenia zostały przeprowadzone prawidłowo.

Równanie (2.16) opisuje przypadek, w którym całka $\int_a^b f(x)dx$ jest przybliżana polem znajdującym się tylko pod jedną parabolą na przedziale $[a, b]$ (podzielono go na dwa równe podprzedziały $[a = x_0, x_1]$ i $[x_1, x_2 = b]$). Wykres omawianej paraboli przechodzi przez punkty $(a = x_0, f(x_0))$, $(\frac{b+a}{2} = x_1, f(x_1))$, $(b = x_2, f(x_2))$.

Dla przypadku ogólnego, w którym przedział całkowania $[a, b]$ zostanie podzielony na $N = 2m$ ($m = 1, 2, \dots$) podprzedziałów równej długości $h = \frac{b-a}{N} = \frac{b-a}{2m}$, przy czym każdy węzeł jest postaci $x_i = a + ih$ ($i = 0, 1, 2, \dots, N = 2m$), wzór (2.16) przyjmuje następującą formę:

$$\begin{aligned} \int_a^b f(x)dx &= \sum_{i=0}^{m-1} \int_{x_{2i}}^{x_{2i+2}} f(x)dx \approx \\ &\approx \left(\frac{h}{3} (f(x_0) + 4f(x_1) + f(x_2)) + \dots + \frac{h}{3} (f(x_{2m-2}) + 4f(x_{2m-1}) + f(x_{2m})) \right) = \\ &= \frac{h}{3} (f(x_0) + 4(f(x_1) + f(x_3) + \dots + f(x_{2m-1}))) + 2(f(x_2) + f(x_4) + \dots + f(x_{2m-2})) + f(x_{2m})) = \\ &= \frac{h}{3} \sum_{i=0}^{m-1} (f(x_{2i}) + 4f(x_{2i+1}) + f(x_{2i+2})). \end{aligned} \quad (2.17)$$

Równanie (2.17) nosi miano *uogólnionego wzoru Simpsona* (mówiąc wzór *Simpsona* będziemy mieli na myśli tę postać uogólnioną). Nie trudno się domyśleć, że w związku z tym, iż wzór ten jedynie przybliży dokładną wartość całki, to wiąże się z tym fakt obciążenia wyniku pewnym błędem. Oszacowanie owego błędu przybliżenia wzorem *Simpsona* wyjaśnia poniższa definicja:

Definicja 6. (Błąd przybliżenia wzorem *Simpsona*)

Przyjmijmy, że BBS_N oznacza błąd bezwzględny przybliżenia (tzn. różnica pomiędzy wartością dokładną i przybliżoną całki). Ponadto niech $f : [a, b] \rightarrow \mathbb{R}$ będzie klasy C^4 na $[a, b]$, oraz $N = 2m$ ($m \in \mathbb{N}$) wyznacza ilość podziałów przedziału $[a, b]$, wtedy spełniona jest następująca nierówność [6]:

$$|BBS_N| \leq \delta_S \quad (2.18)$$

Przez δ_S rozumiemy błąd metody *Simpsona*, który jest opisany następującym wzorem:

$$\delta_S = \frac{(b-a)^5}{180 \cdot N^4} \cdot M_4 \stackrel{N=2m}{=} \frac{(b-a)^5}{2880 \cdot m^4} \cdot M_4, \quad (2.19)$$

przy czym

$$M_4 = \sup_{x \in [a, b]} |f^{(4)}(x)|. \quad (2.20)$$

Zbadajmy dokładność, z jaką wzór (2.17) przybliża wartości całek oznaczonych. W tym celu przedstawmy kompletne wyliczenia dla przypadku, gdzie liczba podziałów przedziału całkowania będzie względnie mała, natomiast sama całka określona tak jak w przykładzie 3.

Przykład 4. (Wyznaczanie przybliżonej wartości całki metodą *Simpsona*)

Obliczmy przybliżoną wartość całki $\int_0^{\frac{3}{2}} e^{x^2} dx$ dzieląc przedział całkowania na $N = 2m = 6$ podprzedziałów równej długości (7 węzłów).

Do wykonania rachunków niezbędna jest nam długość podprzedziału $h = \frac{b-a}{N}$ ($[a, b] = [0, \frac{3}{2}]$) oraz wartości węzłów $x_i = a + ih$ ($i = 0, 1, \dots, 6$).

Zatem: $h = \frac{\frac{3}{2}-0}{6} = \frac{1}{4}$, $x_0 = 0$, $x_1 = \frac{1}{4}$, $x_2 = \frac{2}{4}$, $x_3 = \frac{3}{4}$, $x_4 = 1$, $x_5 = \frac{5}{4}$, $x_6 = \frac{6}{4}$.

Kolejno wstawiamy powyższe dane do (2.17):

$$\begin{aligned} \int_0^{\frac{3}{2}} e^{x^2} dx &= \frac{1}{3} \sum_{i=0}^2 (f(x_{2i}) + 4f(x_{2i+1}) + f(x_{2i+2})) = \\ &= \frac{1}{12} ((f(x_0) + 4f(x_1) + f(x_2)) + (f(x_2) + 4f(x_3) + f(x_4)) + (f(x_4) + 4f(x_5) + f(x_6))) = \\ &= \frac{1}{12} \left((e^{0^2} + 4e^{(\frac{1}{4})^2} + e^{(\frac{2}{4})^2}) + (e^{(\frac{2}{4})^2} + 4e^{(\frac{3}{4})^2} + e^{1^2}) + (e^{1^2} + 4e^{(\frac{5}{4})^2} + e^{(\frac{6}{4})^2}) \right) = \\ &= \frac{1}{12} \left(e^0 + 4 \left(e^{(\frac{1}{16})} + e^{(\frac{9}{16})} + e^{(\frac{25}{16})} \right) + 2 \left(e^{(\frac{4}{16})} + e \right) + e^{(\frac{36}{16})} \right) = \\ &= \frac{1}{12} (1 + 4(1,06449 + 1,75505 + 4,77073) + 2(1,28402 + 2,71828) + 9,48773) = 4,071123 \end{aligned}$$

Wynik, który otrzymaliśmy w przeprowadzonych obliczeniach jest bardzo zbliżony do wartości dokładnej $\int_0^{\frac{3}{2}} e^{x^2} dx = 4,06311405862\dots$. Chcemy jednak sprawdzić, przy jak dużej liczbie podziałów przedziału całkowania przybliżenie zacznie się stabilizować tak, byśmy mogli uznać je za wystarczająco rzetelne (przyjmijmy, że błąd bezwzględny przybliżenia ma być nie większy niż 0,0000001). Podobnie jak dla wzoru trapezów, wykorzystajmy stworzone na nasze potrzeby procedury automatyzujące przeprowadzanie obliczeń.

Procedura 4. (Wznaczenie przybliżonych wartości całek pojedynczych metodą *Simpsona*)

```
>simpson :=proc(a,b,N)
local h,i,j,m,x,y,s,sp;
if (((ceil(N)>N) or (floor(N)<N)) or (mod(N,2)=1 and N>0) or N<=0)
then ERROR('Ostatni parametr musi byc liczba naturalna postaci \2m')
else
m:=N/2;
h:=(b-a)/(2*m);
for j from 0 to 2*m do x:= j -> a+j*h od;
for j from 0 to 2*m do y:= j -> f(x(j)) od;
s:= h/3 * sum(y(2*i)+4*y(2*i+1)+y(2*i+2), i=0..m-1);
sp:=evalf(s,12);
fi;
end;
```

Powyższy skrypt tworzy procedurę w języku *Maple*, realizującą algorytm wyznaczania przybliżonej wartości całki pojedynczej metodą *Simpsona*. Jej parametrami wejściowymi są:

a - wartość dolnej granicy całkowania

b - wartość górnej granicy całkowania

N - opisuje ilość podziałów przedziału całkowania funkcji $f(x)$ (konieczne jest zdefiniowanie $f(x)$ przed wywołaniem każdej prezentowanej w tym podrozdziale procedury). Wartość parametru N musi być liczbą naturalną i podzielną przez 2, gdyż zastosowanie metody parabol wymaga co najmniej 2 podprzedziałów wydzielonych przez 3 równo odległe węzły.

Kod znajdujący się po nagłówku procedury to definicja zmiennych lokalnych, wykorzystywanych wewnątrz owej procedury. Kolejno sprawdzana jest poprawność wprowadzonego parametru N . Jeżeli N posiada niezerową część dziesiętną, bądź jest liczbą naturalną nieparzystą lub całkowitą ≤ 0 wtedy algorytm kończy swoje działanie zwracając błąd o treści 'Ostatni parametr musi by liczba naturalna postaci $2m$ '. W przeciwnym wypadku następuje przejście do głównego algorytmu, który rozpoczyna się od obliczenia ilości par podprzedziałów oraz odległości h pomiędzy kolejnymi punktami węzłowymi. Następnie wyznaczane są wszystkie węzły x_i na danym przedziale $[a, b]$ oraz obliczane wartości funkcji $f(x)$ w tych węzłach.

Przedostatni krok polega na zsumowaniu wyliczonych przed chwilą wartości $f(x_i)$ w sposób zgodny ze wzorem (2.17). Ostatecznie procedura upraszcza postać sumy i zwraca ją w formie 12 znakowego wyniku (nie wliczając w to przecinka).

Zanim zaprezentujemy wyniki zwracane przez procedurę 4, to zechcemy pochylić się jeszcze przez moment nad kodem dodatkowej procedury umożliwiającej oszacowanie błędu δ_S z równania (2.19). Informacja na temat jego wielkości będzie nam pomocna podczas kontroli, czy obliczenia są przeprowadzane poprawnie, tzn. musi zachodzić $|BBS_N| \leq \delta_S$ (2.18) dla coraz to większych ilości podziałów przedziału całkowania.

Procedura 5. (Wyznaczanie błędu przybliżenia dla metody *Simpsona*)

```
> simpsonBlad:= proc(a, b, N)
local d4ff, simd4ff, M4, deltaS;
if (((ceil(N)>N) or (floor(N)<N)) or (mod(N,2)=1 and N>0) or N<=0)
then ERROR('Ostatni parametr musi byc liczba naturalna postaci \2m')
else
d4ff:= diff(diff(diff(diff(f(x),x),x),x),x);
simd4ff:= simplify(d4ff);
M4:= (Optimization[Maximize])(abs(simd4ff), x = a .. b);
deltaS:= evalf(1/180*M4[1]*(b-a)^5/N^4);
fi;
end;
```

Powyższa procedura ma za zadanie wyznaczyć wartość δ_S , tak by prawdziwe było oszacowanie (2.18). Parametry a , b i N definiowane są w sposób identyczny, jak w procedurze 4. Aby wykonał się algorytm główny (znajdujący się wewnątrz procedury w bloku *else*), ponownie musi zostać spełniony warunek mówiący o tym, że N jest liczą naturalną i parzystą. Jeżeli taka prawidłowość zachodzi, to obliczamy wartość $f^{(4)}(x)$ oraz upraszczamy jej postać. Zgodnie z (2.20) wyliczamy maksimum $f^{(4)}(x)$ na przedziale $[a, b]$ (jest to najbardziej znaczący krok) i podstawiamy wszystkie znane już nam argumenty do (2.19). W ten sposób dochodzimy do pożądanej przez nas formy δ_S .

Posiadając już odpowiednie narzędzia (procedura 4 i 5) oraz wiedzę z zakresu metodyki ich działania, wykonajmy pomiary, które zobrazują nam w sposób praktyczny dokładność metody *Simpsona*. Niech N stanowi liczbę podziałów przedziały całkowania, W_{simpson} opisuje wartości zwracane przez procedurę 4. Ponadto przez $|BBS_N|$ oznaczmy błąd bezwzględny przybliżenia wartości całki metodą *Simpsona*, natomiast przez symbol δ_S rozumiemy będziemy wynik wywołania procedury 5. Do obliczeń ponownie wykorzystajmy całkę z przykładu 4 (zastosujemy również jej wartość dokładną, która jest podana w jego rozwiązaniu):

N	W_{simpson}	$ BBS_N $	δ_S
4	4,09788104673	0,03476698811	0,31426966780
6	4,07112329317	0,00800923455	0,06207795906
10	4,06425014407	0,00113608545	0,00804530349
20	4,06318804961	0,00007399099	0,00050283147
40	4,06311873247	0,00000467386	0,00003142697
60	4,06311498371	0,00000092509	0,00000620779
80	4,06311435154	0,00000029292	0,00000196419
100	4,06311417861	0,00000011999	0,00000080453
104	4,06311416121	0,00000010259	0,00000068771
106	4,06311415370	0,00000009508	0,00000063726
150	4,06311408231	0,00000002369	0,00000015892
200	4,06311406613	0,00000000751	0,00000005028
400	4,06311405910	0,00000000048	0,00000000314
1000	4,06311405847	0,00000000015	0,00000000008

Tablica 2.2: Wyniki pomiaru dokładności przybliżania wartości całki pojedynczej wzorem *Simpsona*

Analizując wyniki z powyższej tabeli można zauważyć, że metoda *parabol* jest bardziej optymalna niż metoda *trapezów*, dla której to podobne zestawienie sporządzono w tablicy 2.1. Błąd bezwzględny przybliżenia spadł poniżej 0,0000001 już dla przypadku, w którym dokonaliśmy $N = 106$ podziałów przedziału całkowania, dla porównania w metodzie trapezów podziałów tych musieliśmy wykonać aż $n = 7310$. Wynik zwracany przez procedurę *simpson* zaczyna być bardzo mocno zbliżony do dokładnego (4,06311405862) dla zaledwie $N \geq 20$ podziałów (różnica zauważalna jest dopiero na 5 miejscu po przecinku). Ponadto mamy podstawy do tego, by uznać algorytm za poprawny, ponieważ wartość zwracana przez procedurę

simpson dla $N = 6$ jest taka sama, jak ta otrzymana w przykładzie 4. Co więcej dla każdego z prezentowanych w tabeli przypadków spełniona jest nierówność (2.18) z definicji 6.

W poprzednim podrozdziale wspominaliśmy, że niejednokrotnie interesuje nas minimalizacja błędu dla otrzymanego wyniku, tak by jego wielkość nie przekraczała pewnej akceptowalnej wartości granicznej. Wykonując drobne zmiany w procedurze 5 jesteśmy w stanie otrzymać nową procedurę, wyznaczającą ilość podziałów N przedziału całkowania $[a, b]$, tak by błąd przybliżenia nie przekraczał określonej przez nas wartości:

Procedura 6. (Wyznaczanie najmniejszej liczby N podziałów przedziału całkowania tak, by wynik był obciążony błędem nie większym niż zadany)

```
> simpsonMinN:= proc (a, b, blad)
local d4ff, simd4ff, M4, Nmin, p;
d4ff:= diff(diff(diff(diff(f(x), x), x), x), x);
simd4ff:= simplify(d4ff);
M4:= (Optimization[Maximize])(abs(simd4ff), x = a .. b);
p:= solve*({1/180*(b-a)^5*M4[1]/N^4 < blad, N>0}, {N});
Nmin := p[1]
end;
```

Zmiana polega na tym, że ostatnim parametrem wejściowym jest teraz nie ilość podziałów przedziału, lecz akceptowalna maksymalna wielkość błędu. Również zamiast obliczać δ_S , przekształcamy równanie (2.19) do postaci następującej nierówności:

$$\frac{(b-a)^5}{180 \cdot N^4} \cdot M_4 < blad.$$

W linii trzeciej od końca jest ona rozwiązywana ze względu na zmienną N , przy założeniu, że interesują nas tylko takie wartości, gdzie $N > 0$. Finalnie procedura wypisuje wynik w postaci $N > z$, przy czym $z \in \mathbb{Z}_+$. Nam jednak najbardziej zależy na najbliższej, większej od z liczbie naturalnej, podzielnej przez 2. Właśnie ta liczba jest szukanym przez nas N , dla którego błąd przybliżenia będzie mniejszy niż ten podany jako trzeci parametr wywołania *simpsonMinN*.

2.3 Wzór 'prostokątów'

3. Kwadratury Newtona-Cotesa dla całek podwójnych

3.1 Wzór trapezów

3.2 Wzór Simpsona

3.3 Wzór 'prostokątów'

4. Zastosowanie kwadratur do obliczania całek po obszarach normalnych

Podsumowanie

Celem przyświecającym pisaniu pracy było...

Bibliografia

- [1] Olszowski B., *Wybrane metody numeryczne : podręcznik dla studentów wyższych szkół technicznych*, Wydawnictwo Politechniki Krakowskiej, Kraków 2007, s. 27-37
- [2] Kosma Z., *Metody numeryczne dla zastosowań inżynierskich*, Wydawnictwo Politechniki Radomskiej, Radom 2007, s. 155-160, 262-283
- [3] Cichoń Cz., Politechnika Świętokrzyska, Wydział Zarządzania i Modelowania Komputerowego, <http://www.tu.kielce.pl/~sk/epi/epi-metody-obliczeniowe.pdf>, stan z 09.03.2017r.
- [4] Małecka A., Wydział Matematyki i Informatyki UAM, Zakład Teorii Funkcji Rzeczywistych, <http://www.staff.amu.edu.pl/~awoj/OLAT/kwadraturyNC.pdf>, stan z 19.03.2017r.
- [5] Moszyński K., Wydział Matematyki, Informatyki i Mechaniki Uniwersytetu Warszawskiego, <https://www.mimuw.edu.pl/~kmoszyns/cc.pdf>, stan z 27.03.2017r.
- [6] Ferenstein E., Osłowski B., Winnicki A., Polsko-Japońska Akademia Technik Komputerowych, kurs *Analiza Matematyczna*, <http://edu.pjwstk.edu.pl/wyklady/am/scb/main79.html>, stan z 15.03.2017r.
- [7] Orchel M., Akademia Górniczo-Hutnicza im. Stanisława Staszica w Krakowie, <http://home.agh.edu.pl/~morchel/files/mownit/zajecia4/zajecia4-mownit.pdf>, stan z 11.03.2017r.
- [8] Bozorgmanesh A. R., Int. J. Industrial Mathematics Vol. 1, No. 2 (2009) str. 175-181, http://ijim.srbiau.ac.ir/article_2021_227340ff28516b33b11f8a4436633fc2.pdf, stan z 17.06.2017r.

- [9] Tănase D., BULETINUL Universității Petrol – Gaze din Ploiești, Vol. LIX No. 1/2007, Seria Matematică - Informatică - Fizică, str. 7-12 , http://bulletin-mif.unde.ro/docs/20071/Buletin_UPG_MIF_Nr1_2007-02.pdf, stan z 17.06.2017r.

Spis procedur w języku Maple

- [1] Wznaczanie przybliżonych wartości całek pojedynczych metodą trapezów (str. 21)
- [2] Wyznaczanie błędu przybliżenia dla metody trapezów (całka pojedyncza, str. 23)
- [3] Wyznaczanie najmniejszej liczby n podziałów przedziału całkowania tak, by wynik był obciążony błędem nie większym niż zadany (Metoda trapezów, całka pojedyncza, str. 24)
- [4] Wznaczanie przybliżonych wartości całek pojedynczych metodą *Simpsona* (str. 29)
- [5] Wyznaczanie błędu przybliżenia dla metody *Simpsona* (całka pojedyncza, str. 30)
- [6] Wyznaczanie najmniejszej liczby N podziałów przedziału całkowania tak, by wynik był obciążony błędem nie większym niż zadany (Metoda *Simpsona*, całka pojedyncza, str. 32)