


Paulina Grabowska 	Metody numeryczne
Kwadratura Gaussa 2d	

1. Wstęp

Całkowanie numeryczne to proces przybliżonego obliczania wartości całki oznaczonej funkcji na określonym przedziale. W odróżnieniu od całkowania analitycznego, które opiera się na dokładnych metodach algebraicznych, całkowanie numeryczne wykorzystuje techniki numeryczne i komputerowe do przybliżenia wyniku.

Ważną kwestią podczas całkowania numerycznego jest także kontrola błędów numerycznych, które mogą wynikać z ograniczeń komputerowych oraz przybliżeń użytych w metodach. Dlatego istotne jest zrozumienie, jakie błędy mogą wystąpić i jak je minimalizować.

Całkowanie numeryczne znajduje zastosowanie w różnych dziedzinach nauki i techniki, szczególnie tam, gdzie nie ma dostępu do rozwiązań analitycznych lub są one zbyt skomplikowane. Dzięki wykorzystaniu metod numerycznych, możemy szybko i efektywnie obliczać wartości całek dla szerokiego zakresu funkcji, co jest kluczowe w wielu aplikacjach inżynierskich, naukowych i matematycznych.

2. Kwadratura Gaussa 2D

Kwadratura Gaussa to technika numerycznej całkowania, która jest używana do przybliżonego obliczania całek oznaczonych, zwłaszcza tych, które są trudne lub niemożliwe do rozwiązania analitycznie. Metoda ta opiera się na znalezieniu wag i miejsc zerowych wielomianów ortogonalnych, co pozwala na dokładniejsze przybliżenie całki niż tradycyjne metody numeryczne, takie jak metoda prostokątów czy trapezów.

Pochodne względem ξ i η są używane do przekształcenia całki dwuwymiarowej w nowym układzie współrzędnych (ξ, η) na całkę dwuwymiarową w standardowym układzie współrzędnych (x, y) . Te pochodne pozwalają na zmianę zmiennych całkowania, co umożliwia zastosowanie standardowych technik całkowania dla nowego układu współrzędnych.

W kontekście kwadratury Gaussa 2D, pochodne te są wykorzystywane do przekształcenia funkcji i obszaru całkowania w taki sposób, aby móc zastosować kwadraturę Gaussa w nowym układzie współrzędnych. Dzięki temu możliwe jest dokładniejsze przybliżenie całki dwuwymiarowej niż w przypadku stosowania prostszych metod całkowania.

$$\begin{pmatrix} \frac{\partial}{\partial \xi} \\ \frac{\partial}{\partial \eta} \end{pmatrix} = \begin{bmatrix} \frac{\partial x}{\partial \xi} & \frac{\partial y}{\partial \xi} \\ \frac{\partial x}{\partial \eta} & \frac{\partial y}{\partial \eta} \end{bmatrix} \begin{pmatrix} \frac{\partial}{\partial x} \\ \frac{\partial}{\partial y} \end{pmatrix} = [J] \begin{pmatrix} \frac{\partial}{\partial x} \\ \frac{\partial}{\partial y} \end{pmatrix} \quad (1)$$

We wzorze (1) przez $[J]$ została oznaczona macierz Jacobiego, natomiast we wzorze (2) $[J_0]$ oznacza Jakobian transformacji układu współrzędnych. Ponadto we wzorze (2) występują też zmienne w_i oraz w_j , które są oznaczeniem wag.

Całkowanie funkcji w układzie ξ i η prowadzone jest metodą Gaussa:

$$\iint f(x,y) dx dy = \int_{-1}^1 \int_{-1}^1 f(\xi, \eta) J_0 d\eta d\xi = \sum_{i=1}^n \sum_{j=1}^n w_i w_j f(\xi_i, \eta_j) J_0 \quad (2)$$

Wartości ξ i η są wybrane tak, aby odwzorować obszar całkowania w standardowym układzie współrzędnych (x, y) na kwadrat o wymiarach 2×2 w nowym układzie współrzędnych (ξ, η) . Te wartości są zazwyczaj wybierane w oparciu o określone zasady, które prowadzą do optymalnego przybliżenia całki.

W przypadku kwadratury Gaussa 2D, węzły i wagi są zwykle wybrane tak, aby zapewnić jak najdokładniejsze przybliżenie całki dla danego stopnia wielomianu. W praktyce, węzły i wagi dla kwadratury Gaussa 2D mogą być obliczane numerycznie lub wykorzystywane z tablic dostępnych w literaturze.

We wszystkich późniejszych obliczeniach zostaną przyjęte wartości zmiennych (3). Dla czterech punktów całkowania wagi w_0, w_1 wynoszą 1.

$$\begin{aligned} w_0 &= w_1 = 1 \\ \xi_0 &= \eta_0 = 0,57735029 \\ \xi_1 &= \eta_1 = -0,57735029 \end{aligned} \quad (3)$$

Macierz Jacobiego oraz jej wyznacznik, nazywany jakobianem, są używane w transformacjach zmiennych w wielu dziedzinach matematyki, fizyki i inżynierii. W kontekście całkowania numerycznego, zwłaszcza przy użyciu kwadratury Gaussa, macierz Jacobiego jest kluczowym narzędziem do transformacji obszaru całkowania z jednego układu współrzędnych na inny.

Skorzystanie z macierzy Jacobiego umożliwia transformację zmiennych (dla całki dwuwymiarowej, transformacja ta jest niezbędna, gdy chcemy przejść od standardowego układu współrzędnych (x, y) do nowego układu (ξ, η)), wykrywanie zniekształceń oraz skalowanie obszaru całkowania.

$$[J] = \begin{bmatrix} \frac{\partial x}{\partial \xi} & \frac{\partial y}{\partial \xi} \\ \frac{\partial x}{\partial \eta} & \frac{\partial y}{\partial \eta} \end{bmatrix} \quad (4)$$

$$\det|J| = \frac{\partial x}{\partial \xi} \cdot \frac{\partial y}{\partial \eta} - \frac{\partial y}{\partial \xi} \cdot \frac{\partial x}{\partial \eta}$$

Często w analizie numerycznej i metodyce elementów skończonych, N oznacza funkcje kształtu. Te funkcje są używane do przybliżenia zachowania się pola, na przykład temperatury lub odkształcenia, wewnątrz elementów skończonych. Funkcje kształtu N są wykorzystywane do przybliżenia wartości pola wewnątrz elementu skończonego.

$$\begin{aligned} N_1(\xi, \eta) &= 0,25(1 - \xi)(1 - \eta) & N_2(\xi, \eta) &= 0,25(1 + \xi)(1 - \eta) \\ N_3(\xi, \eta) &= 0,25(1 + \xi)(1 + \eta) & N_4(\xi, \eta) &= 0,25(1 - \xi)(1 + \eta) \end{aligned} \quad (5)$$

Sumy pochodnych pomagają nam w tej transformacji, ponieważ pozwalają nam określić, jak zmieniają się współrzędne x i y w zależności od zmian w lokalnych współrzędnych naturalnych ξ i η . Sumy pomagają nam w przeniesieniu analizy z lokalnego, skomplikowanego układu współrzędnych na bardziej zrozumiały globalny układ współrzędnych, co pozwala na bardziej precyzyjną analizę zachowania się pola fizycznego w badanym obszarze.

$$\begin{aligned} \frac{\partial x}{\partial \xi} &= \sum_{i=1}^4 \frac{\partial N_i}{\partial \xi} x_i & \frac{\partial x}{\partial \eta} &= \sum_{i=1}^4 \frac{\partial N_i}{\partial \eta} x_i \\ \frac{\partial y}{\partial \xi} &= \sum_{i=1}^4 \frac{\partial N_i}{\partial \xi} y_i & \frac{\partial y}{\partial \eta} &= \sum_{i=1}^4 \frac{\partial N_i}{\partial \eta} y_i \end{aligned} \quad (6)$$

3. Implementacja numeryczna

Funkcja *gaussQuadArea* (Fragment kodu 1) przyjmuje cztery argumenty:

- *xys* $[[2]]$: Jest to dwuwymiarowa tablica, która zawiera współrzędne punktów w przestrzeni (x, y) węzłów elementu skończonego.
- *points* $[]$: Tablica zawierająca punkty całkowania.
- *w* $[]$: Tablica zawierająca wagi punktów całkowania.
- *numPoints*: Liczba punktów całkowania.

Następnie deklarowane są zmienne lokalne funkcji, które będą używane w obliczeniach, korzystając ze wzorów (5):

- *dKSI* $[2][4]$: Tablica przechowująca pochodne cząstkowe po współrzędnych naturalnych ξ dla każdego punktu całkowania i każdej współrzędnej (x, y) .
- *dETA* $[2][4]$: Tablica przechowująca pochodne cząstkowe po współrzędnych naturalnych η dla każdego punktu całkowania i każdej współrzędnej (x, y) .
- *function* $[2][2]$: Macierz Jacobiego, przechowująca wyznacznik macierzy Jacobiego dla każdego punktu całkowania.
- *dxdKSI*, *dydKSI*, *dxdETA*, *dydETA*: Zmienne pomocnicze służące do przechowywania pochodnych cząstkowych.

Zaimplementowano podwójną pętlę *for*, która oblicza pochodne cząstkowe po współrzędnych naturalnych ξ i η dla każdego punktu całkowania, analogicznie jak przedstawiono we wzorze (6). Wykorzystuje ona podane punkty całkowania, aby obliczyć pochodne w czterech punktach dla każdej współrzędnej.

Następnie obliczany jest wyznacznik macierzy Jacobiego dla każdego punktu całkowania. Jest to kluczowy krok w całkowaniu numerycznym, ponieważ wyznacznik Jacobiego jest używany do przekształcenia całki z obszaru lokalnego na obszar naturalny.

Na sam koniec obliczana jest całka z wartości bezwzględnej wyznacznika macierzy Jacobiego pomnożona przez wagi punktów całkowania. Jest to sposób obliczania powierzchni obszaru. Funkcja zwraca obliczoną powierzchnię obszaru.

```

1. double gaussQuadArea (double xys[][2], double points[], double w[], int numPoints) {
2.     double dKSI [2][4];
3.     double dETA [2][4];
4.     double function [2][2];
5.     double dxdKSI, dydKSI, dxdETA, dydETA;
6.     double area = 0.0;
7.
8.     // pochodne cząstkowe
9.     for (int i = 0; i < 2; i++) {
10.         dKSI[i][0] = -0.25 * (1.0 - points[i]);
11.         dKSI[i][1] = 0.25 * (1.0 - points[i]);
12.         dKSI[i][2] = 0.25 * (1.0 + points[i]);
13.         dKSI[i][3] = -0.25 * (1.0 + points[i]);
14.
15.         dETA[i][0] = -0.25 * (1.0 - points[i]);
16.         dETA[i][1] = -0.25 * (1.0 + points[i]);
17.         dETA[i][2] = 0.25 * (1.0 + points[i]);
18.         dETA[i][3] = 0.25 * (1.0 - points[i]);
19.     }
20.
21.     // wyznacznik Jacobianu
22.     for (int i = 0; i < 2; i++) {
23.         for (int j = 0; j < 2; j++) {
24.             dxdKSI = dKSI[i][0] * xys [0][0] + dKSI[i][1] * xys [1][0] + dKSI[i][2] *
xys [2][0] + dKSI[i][3] * xys [3][0];
25.             dydKSI = dKSI[i][0] * xys [0][1] + dKSI[i][1] * xys [1][1] + dKSI[i][2] *
xys [2][1] + dKSI[i][3] * xys [3][1];
26.
27.             dxdETA = dETA[j][0] * xys [0][0] + dETA[j][1] * xys [1][0] + dETA[j][2] *
xys [2][0] + dETA[j][3] * xys [3][0];
28.             dydETA = dETA[j][0] * xys [0][1] + dETA[j][1] * xys [1][1] + dETA[j][2] *
xys [2][1] + dETA[j][3] * xys [3][1];
29.
30.             function[i][j] = dxdKSI * dydETA - dydKSI * dxdETA;
31.         }
32.     }
33.
34.     // powierzchnia czworokąta
35.     for (int i = 0; i < numPoints; i++) {
36.         for (int j = 0; j < numPoints; j++) {
37.             area += abs(function[j][i]) * w[i] * w[j];
38.         }
39.     }
40.
41.     return area;
42. }

```

Fragment kodu 1: funkcja *gaussQuadArea*

Funkcja *main* jest punktem wejścia programu (*Fragment kodu 2*). Zmienna *xy*s zawiera współrzędne czterech węzłów elementu skończonego. Każdy węzeł jest zdefiniowany przez dwie współrzędne (x, y). Tablica *points* zawiera punkty całkowania. W tym przypadku używane są dwa punkty całkowania. Tablica *w* zawiera wagi punktów całkowania. Wartości wag są równe 1.0 dla każdego punktu całkowania. Wywołanie funkcji *gaussQuadArea* z przekazanymi argumentami *xy*s, *points*, *w* i liczbą punktów całkowania równą 2. Funkcja ta oblicza powierzchnię obszaru na podstawie podanych danych. Następuje wyświetlenie obliczonej powierzchni obszaru na standardowym wyjściu.

```

1. int main () {
2.     double xys [4][2] = {{1, 0}, {0, 1}, {-1, 0}, {0, -1}};
3.     double points [2] = {0.577350269, -0.577350269};
4.     double w [2] = {1.0, 1.0};
5.
6.     double area = gaussQuadArea (xys, points, w, 2);
7.     cout << "area of quadrangle: " << area << endl;
8.
9.     return 0;
10.}
11.

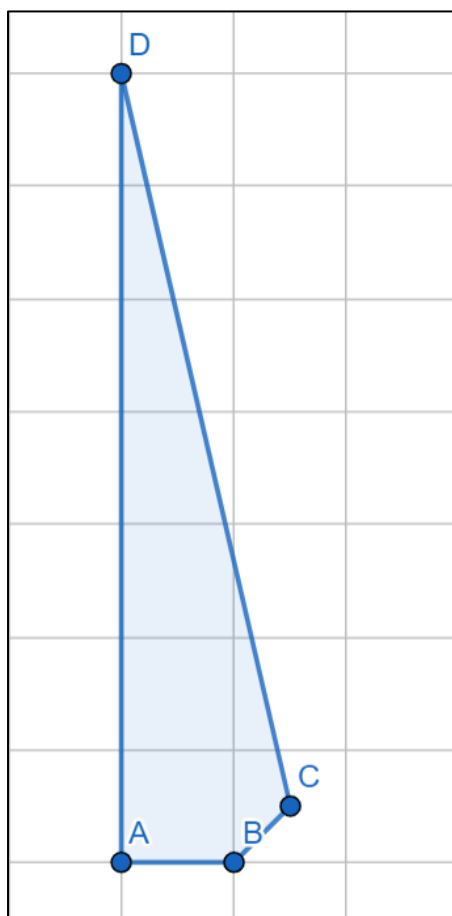
```

Fragment kodu 2: funkcja *main*

Sprawdzono poprawność działania funkcji dla czworoboku o współrzędnych (0, 0), (2, 0), (3, 1), (0, 14). Spodziewanym polem jest pole równe 22. Taki też wynik otrzymano (*Zrzut ekranu 1*), co pozwala przypuszczać, że funkcja została zaimplementowana poprawnie. Figur zwizualizowano w programie GeoGebra (*Zrzut ekranu 2*).

area of quadrangle: 22

Zrzut ekranu 1 Zrzut ekranu konsoli po kompilacji dla danych testowych



Zrzut ekranu 2 Zrzut ekranu wizualizacji danych testowych

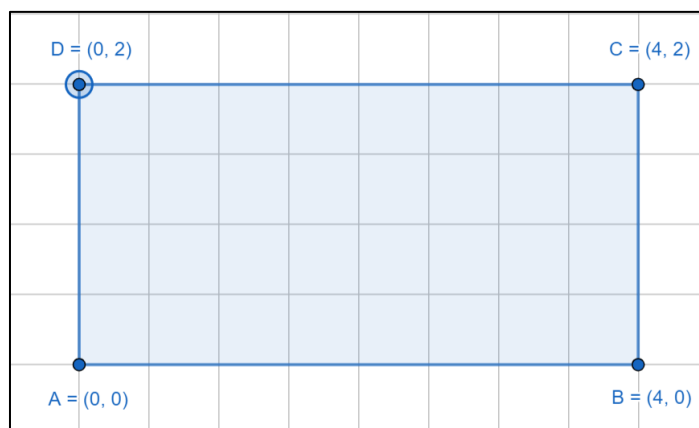
4. Testy jednostkowe

4.1 Test dla prostokąta

Wybrano cztery współrzędne mające odpowiadać czterem wierzchołkom prostokąta w kartezjańskim układzie współrzędnych (*Tabela 1*), gdzie zestawiono również wartość otrzymaną po kompilacji programu oraz wartość oczekiwaną otrzymaną w programie Math Open Reference. Czworokąt zwizualizowano w programie GeoGebra (*Rys. 1*).

Tabela 1. Zestawienie współrzędnych oraz wartości oczekiwanych i otrzymanych dla testu jednostkowego dla prostokąta

Współrzędna x	Współrzędna y	Wartość otrzymana	Wartość spodziewana
0	0	8	8
4	0		
4	2		
0	2		



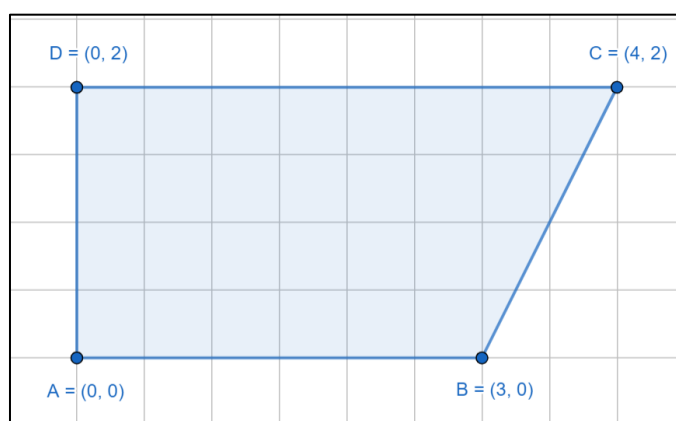
Rys.1 Wizualizacja czworokąta w programie GeoGebra
w teście jednostkowym dla prostokąta

4.2 Test dla trapezu

Wybrano cztery współrzędne mające odpowiadać czterem wierzchołkom trapezu w kartezjańskim układzie współrzędnych (*Tabela 2*), gdzie zestawiono również wartość otrzymaną po kompilacji programu oraz wartość oczekiwaną otrzymaną w programie Math Open Reference. Czworokąt zwizualizowano w programie GeoGebra (*Rys. 2*).

Tabela 2. Zestawienie współrzędnych oraz wartości oczekiwanych i otrzymanych dla testu jednostkowego dla trapezu

Współrzędna x	Współrzędna y	Wartość otrzymana	Wartość spodziewana
0	0	7	7
3	0		
4	2		
0	2		



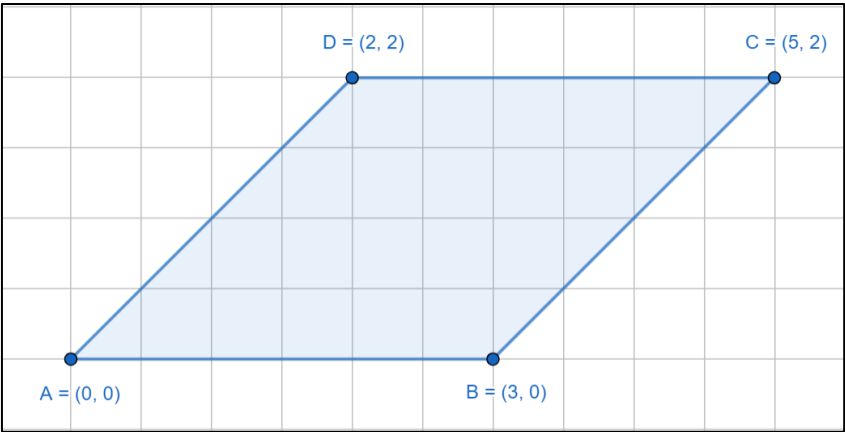
Rys.2 Wizualizacja czworokąta w programie GeoGebra w teście jednostkowym dla trapezu

4.3 Test dla równoległoboku

Wybrano cztery współrzędne mające odpowiadać czterem wierzchołkom równoległoboku w kartezjańskim układzie współrzędnych (*Tabela 3*), gdzie zestawiono również wartość otrzymaną po kompilacji programu oraz wartość oczekiwaną otrzymaną w programie Math Open Reference. Czworokąt zwizualizowano w programie GeoGebra (*Rys. 3*).

Tabela 3. Zestawienie współrzędnych oraz wartości oczekiwanych i otrzymanych dla testu jednostkowego dla równoległoboku

Współrzędna x	Współrzędna y	Wartość otrzymana	Wartość spodziewana
0	0	6	6
3	0		
5	2		
2	2		



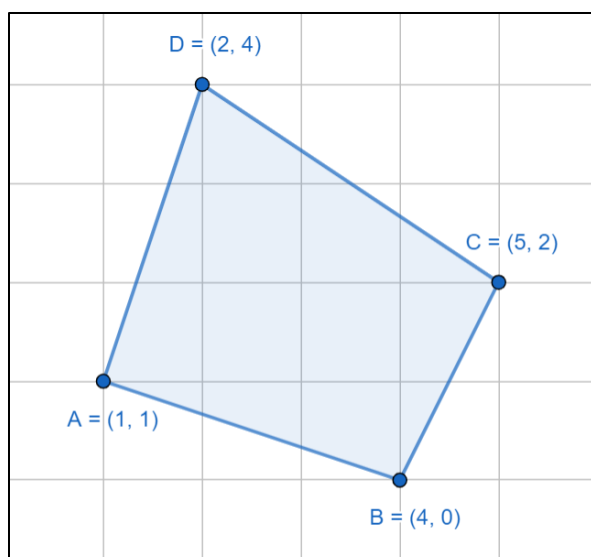
Rys. 3 Wizualizacja czworokąta w programie GeoGebra
w teście jednostkowym dla równoległoboku

4.4 Test dla czworoboku nieregularnego

Wybrano cztery współrzędne mające odpowiadać czterem wierzchołkom czworoboku nieregularnego w kartezjańskim układzie współrzędnych (*Tabela 4*), gdzie zestawiono również wartość otrzymaną po kompilacji programu oraz wartość oczekiwaną otrzymaną w programie Math Open Reference. Czworokąt zwizualizowano w programie GeoGebra (*Rys. 4*).

Tabela 4. Zestawienie współrzędnych oraz wartości oczekiwanych i otrzymanych dla testu jednostkowego dla czworoboku nieregularnego

Współrzędna x	Współrzędna y	Wartość otrzymana	Wartość spodziewana
1	1	9	9
4	0		
5	2		
2	4		



Rys. 4 Wizualizacja czworokąta w programie GeoGebra w teście jednostkowym dla czworokąta nieregularnego

5. Opracowanie wyników

Zebrane wyniki zestawiono w Tabeli 5. W testach jednostkowych sprawdzano różne czworokąty. Wszystkie otrzymane wyniki pokrywają się z wynikami spodziewanymi, otrzymanymi w programie Math Open Reference, co pozwala na stwierdzenie że funkcja *gaussQuadArea* licząca kwadraturę Gaussa 2D została poprawnie zaimplementowana i zwraca poprawne wyniki.

Tabela 5. Zestawienie testów oraz wartości oczekiwanych i otrzymanych dla testów jednostkowych

Nr testu	Figura	Wartość otrzymana	Wartość spodziewana
1	Prostokąt	8	8
2	Trapez	7	7
3	Równoległobok	6	6
4	Czworokąt nieregularny	9	9

6. Wnioski

Kwadratura Gaussa jest skuteczną metodą numeryczną do obliczania całek oznaczonych, szczególnie w przypadku funkcji o nieregularnych kształtach i złożonych obszarach. Wyniki testów potwierdzają poprawność działania algorytmu dla różnych rodzajów figur.

Jest to metoda zapewniająca wysoką dokładność obliczeń, nawet dla niewielkiej liczby punktów całkowania. Wartości otrzymane są zgodne z wartościami oczekiwanymi, co sugeruje, że metoda ta dostarcza wiarygodne wyniki.

Metoda ta może być stosowana do różnych kształtów obszarów, takich jak prostokąty, trapezy, równoległoboki oraz czworoboki nieregularne. Jest to wszechstronna technika, która może być używana w różnych dziedzinach nauki i inżynierii.

Kwadratura Gaussa może być bardziej efektywna niż inne metody całkowania numerycznego, szczególnie dla funkcji wymagających dużej liczby punktów całkowania. Dzięki zastosowaniu odpowiednio dobranych punktów i wag, możliwe jest uzyskanie dokładnych wyników przy minimalnym nakładzie obliczeniowym.

Kwadratura Gaussa stanowi potężne narzędzie do numerycznego rozwiązywania problemów całkowania, zapewniając wysoką dokładność i efektywność nawet dla skomplikowanych funkcji i obszarów. Jej wszechstronność i skuteczność sprawiają, że jest powszechnie stosowana w różnych dziedzinach nauki i technologii.

7. Źródła

Wykłady z Metod Numerycznych autorstwa dr hab. Danuty Szeligi

Prezentacja Metody Numeryczne. Kwadratury Gaussa 2D autorstwa dr. Hab. Inż. Marcina Hojnego.

Wikipedia – artykuły o całkowaniu numerycznym, kwadraturze Gaussa

Calculator Math Open Reference: <https://www.mathopenref.com/coordpolygonareacalc.html>

Program Geogebra: <https://www.geogebra.org/calculator>