

POLITECHNIKA POZNAŃSKA
WYDZIAŁ BUDOWY MASZYN I ZARZĄDZANIA


STUDIA MAGISTERSKIE
STUDIUM DZIENNE



**Symulacja niestacjonarnego przepływu ciepła
bezsiatkową metodą Kansa w obszarze o
nieregularnym brzegu**

Praca dyplomowa

Prowadzący: dr Tomasz Stręk



2005

[REDACTED]

SPIS TREŚCI

1. Spis treści.....	2
2. Streszczenie.....	3
3. Wprowadzenie.....	4
4. Zjawisko przewodnictwa cieplnego	6
4.1 Równanie ciepła.....	6
5. Bezsiatkowa metoda Kansa dla przypadku stacjonarnego i niestacjonarnego....	8
5.1 Zagadnienie stacjonarne	9
5.2 Zagadnienie niestacjonarne	10
5.3 Operatory różniczkowe dla radialnych funkcji bazowych dla przypadku stacjonarnego.....	11
5.3.1 Funkcja wielomianowa.....	11
5.3.2 Funkcja gaussowska	12
5.3.3 Funkcja poliharmoniczne	14
5.4 Operatory różniczkowe dla radialnych funkcji bazowych dla przypadku niestacjonarnego.....	15
5.4.1 Funkcja wielomianowa.....	15
5.4.2 Funkcja gaussowska	17
5.4.3 Funkcja poliharmoniczne	18
6. Narzędzie programistyczne – Scilab	21
6.1 Przykłady wykorzystania Scilaba do obliczeń i wizualizacji wyników.....	23
7. Wyniki numeryczne.....	27
7.1 Zagadnienie stacjonarne na obszarze regularnym	27
7.2 Zagadnienie niestacjonarne na obszarze regularnym	39
7.3 Zagadnienie stacjonarne na obszarze nieregularnym.....	48
7.4 Zagadnienie niestacjonarne na obszarze nieregularnym.....	51
8. Wnioski	54
9. Literatura	55
10. Dodatki	56

2. Streszczenie

Celem pracy jest symulacja niestacjonarnego przepływu ciepła bezsiatkową metodą Kansa w obszarze o nieregularnym brzegu. Metodę tą zaimplementowano z wykorzystaniem programu Scilab. W pracy badano zagadnienie przepływu ciepła na całym brzegu rozważanego obszaru z warunkiem Dirichleta (temperatura na brzegu była funkcją położenia).

W pierwszej części pracy omówiono podstawy teoretyczne przewodnictwa ciepła w ciałach stałych. W dalszej części pracy szczegółowo opisano metodę Kansa, wyprowadzono operatory różniczkowe dla trzech radialnych funkcji bazowych (RFB): funkcji wielomianowej, funkcji gaussowskiej oraz funkcji poliharmoniczej dotyczących stacjonarnego i niestacjonarnego przepływu na obszarze o regularnym i nieregularnym brzegu. Początkowo jednocześnie wydawałoby się, że równania przedstawione w tym rozdziale są skomplikowane, ale po bardziej wnikliwej analizie okazuje się, że ich wyprowadzenia są proste i powtarzalne. W dalszej części pracy zaprezentowano krótki opis programu Scilab. Wyposażony jest on w setki gotowych funkcji do operacji algebraicznych, rozwiązywania równań liniowych i nieliniowych, grafiki dwu- i trójwymiarowej oraz wiele innych. W końcowej części pracy zaprezentowane zostały tabele z wynikami oraz ich interpretacje graficzne. Ze względu na obszerność zagadnienia przedstawiono tylko wybrane wyniki obliczeń numerycznych. Warto podkreślić zaletę środowiska programu Scilab, które okazało się być efektywnym i efektownym narzędziem do obliczeń numerycznych i inżynierskich.

Dodatkowo do pracy załączono płytę CD, która zawiera program instalacyjny pakietu Scilab oraz kody źródłowe programów wykorzystanych podczas obliczeń numerycznych. Programy mogą być modyfikowane przez użytkownika do własnych celów.

3. Wprowadzenie

Najstarszą, historycznie rzecz biorąc, jest *metoda różnic skończonych* – opisana dość szczegółowo w książkach z lat pięćdziesiątych (np. Collatz, Panow). W tym samym czasie (1946) pojawiła się pewna odmiana tej metody – *metoda bilansów elementarnych*, której podstawy sformułował Wniczew. Metoda bilansów jest bardzo efektywna w obliczeniach cieplnych, a przy tym dzięki swojej oczywistej interpretacji fizycznej zrozumiała i łatwo przyswajalna. Drugą z bardzo rozpowszechnionych w mechanice i termodynamice metod numerycznych jest *metoda elementów skończonych* (MES). Łączy się ją zazwyczaj z nazwiskiem O.C. Zienkiewicza, który w 1967 roku opublikował bardzo obszerną monografię *Finite Element Method*. W latach osiemdziesiątych coraz więcej pisze się o *metodzie elementu brzegowego*. Do najwybitniejszych jej propagatorów należą C.A. Brebbia, J.C.F. Telles, L.C. Wrobel – autorzy znanej monografii *Boundary Element Techniques* (1984) [3].

W ostatniej dekadzie, postęp w stosowaniu radialnych funkcji bazowych (RFB) w bezsiatkowych metodach aproksymacji rozwiązań cząstkowego równania różniczkowego miała przyciągnąć wielu badaczy nauki i inżynierii. Pierwszą dziedziną typu metod bezsiatkowych jest tak zwana metoda Kansa rozwinięta przez Kansa w 1990 roku [6], jest uzyskiwana bezpośrednio z kolokacji RFB (na przykład w szczególności funkcji wielomianowej, *ang. multiquadric*), dla numerycznej aproksymacji rozwiązania. Funkcja wielomianowa została pierwszy raz rozwinięta przez Hardy’ego [8] w 1971 roku, jako wielowymiarowe rozproszenie metody interpolacji do modelowania pola przyciągania ziemskiego. Nieznana przez wielu badaczy akademickich aż do czasu publikacji recenzji Franke’ego [4] opisującej 29 metod interpolacyjnych, gdzie funkcja wielomianowa została zaliczona do najlepszych w oparciu o dokładność, aspekty wizualne, wrażliwości na parametry, nadanie ważności czasu, przechowywanie żądań i łatwość implementacji. Metoda Kansa została ostatnio powiększona do rozwiązania różnych zwykłych i cząstkowych równań różniczkowych włączając dwufazowe i trójfazowe modele mieszanin dla tkanki inżynierskiej, jednowymiarowego nieliniowego równania Burgera [5] z uderzeniem fali, powierzchniowego równanie wody z przypiływem i symulacją prądu [9] oraz problem przepływu ciepła [7].

Tradycyjne radialne funkcje bazowe są definiowane jako funkcje globalne, których wyniki w pełni wynikają z współczynnika macierzy. Przeszkadza to w stosowaniu RFB na dużą skalę z powodu źle uwarunkowanych współczynników macierzy. W wielu przypadkach, dokładność rozwiązania RFB zależy często od wyboru parametru c podstawianego do funkcji wielomianowej czy też gausowskiej. Wybór optymalnej wartości parametru c jest poprzedzony intensywnymi badaniami. Ostatnio, RFB postaci r^3 jest używana zamiennie, aby uniknąć problemu wyboru optymalnego kształtu parametru. Dwie ważne cechy bezsiatkowych metod z wykorzystaniem RFB to fakt że są one „naprawdę” metodami typu „bez siatek” (*ang. „mesh-free”*) i ich implementacja jest bardzo prosta w porównaniu z nurtem dominującym w technikach numerycznych takich jak metoda elementu skończonego, metoda różnic skończonych, metoda elementu brzegowego czy też metoda objętości skończonej.

Celem pracy była symulacja niestacjonarnego przepływu ciepła bezsiatkową metodą Kansa w obszarze o nieregularnym brzegu. Metodę tą zaimplementowano w środowisku programistycznym Scilab. W pracy badano zagadnienie przepływu ciepła z warunkiem Dirichleta na całym brzegu rozważanego obszaru. Temperatura na brzegu była funkcją położenia.

W rozdziale 4 przedstawiono problem przewodnictw ciepła jak również zapisano równanie ciepła. Bardzo ważną część pracy stanowi opis metody Kansa dla przypadku stacjonarnego i niestacjonarnego. W tym rozdziale opisano również operatory różniczkowe

dla poszczególnych przypadków. Na pierwszy rzut oka równania przedstawione w tym rozdziale mogą wydawać się skomplikowane, ale po bardziej wnikliwej analizie okazuje się, że ich wyprowadzenia są proste i powtarzalne. W rozdziale 6 zawarto krótki opis narzędzia programistycznego Scilab, który jest odpowiednikiem pakietu naukowego Matlab. Rozdział 7 zawiera tabele z wynikami oraz interpretacje graficzną dla poszczególnych przypadków RFB. Na końcu w rozdziale 8 zawarto wnioski i przemyślenia własne.

4. Zjawisko przewodnictwa cieplnego

Od kuchennego tosterka do najnowszego wysoko sprawnego mikroprocesora, ciepło jest wszechobecne i ma wielkie znaczenie w inżynierii światowej. Dla optymalizacji sprawności cieplnej i redukcji kosztów, inżynierzy i badacze wykorzystują analizę opartą na metodzie, elementu skończonego. Ponieważ wiele właściwości materiałowych jest zależnych od temperatury, efekt ciepła występuje w wielu innych dziedzinach modelowania multidyscyplinarnego.

Na przykład, zarówno toster jak i mikroukład zawierają elektryczne przewodniki, które wytwarzają energię cieplną podczas przepływu przez nie prądu elektrycznego. Gdy te przewodniki wyzwalają energię cieplną, temperatura układu zwiększa się w tych przewodnikach. Jeśli przewodnictwo elektryczne jest zależne od temperatury, to zmienia się odpowiednio i wpływa na pole elektryczne w przewodniku. Inny przykład z multidyscyplinarnych połączeń, które pociągają za sobą przepływ ciepła są naprężenia cieplne, konwekcja w płynach i nagrzewanie indukcyjne.

Przepływ ciepła jest definiowany jako ruch energii spowodowany różnicą temperatur. Opisywane jest to przez następujące trzy mechanizmy:

- **Przewodzenie** jest przepływem ciepła przez dyfuzję w ośrodku stacjonarnym spowodowane gradientem temperatury. Ośrodkiem może być ciało stałe lub płyn.
- **Konwekcja** jest przepływem ciepła pomiędzy ciepłą powierzchnią a zimnym ruchomym płynem, zimną powierzchnią a ciepłym ruchomym płynem lub ciepłą i zimną powierzchnią płynu. Konwekcja występuje w płynach i gazach.
- **Promieniowanie** jest przepływem ciepła pomiędzy powierzchnią A z temperaturą T_1 i powierzchnią B z temperaturą T_2 przez elektromagnetyczne fale, pod warunkiem, że $T_1 \neq T_2$ i powierzchnia A jest widoczna dla nieskończenie małego obserwatora na powierzchni B.

4.1 Równanie ciepła

Matematyczny model dla przepływu ciepła przez przewodzenie jest *równaniem ciepła*:

$$\rho C \frac{\partial T}{\partial t} - \nabla \cdot (k \nabla T) = Q \quad (4.1)$$

gdzie

- T – temperatura [$^{\circ}\text{C}$ lub K]
- ρ – gęstość [kg/m^3]
- k – przewodnictwo cieplne [$\text{W}/(\text{m}^2 \cdot \text{K})$]
- Q – źródło ciepła lub rozpraszacz [W]
- C – ciepła właściwe [$\text{J}/(\text{kg} \cdot \text{K})$]
 - C_p – ciepło właściwe przy stałym ciśnieniu [$\text{J}/(\text{kg} \cdot \text{K})$]
 - C_v – ciepło właściwe przy stałej objętości [$\text{J}/(\text{kg} \cdot \text{K})$]

Dla stanu ustalonego, temperatura nie zmienia się w czasie, więc pierwszy człon równania (4.1) zawierający pochodną temperatury po czasie zeruje się. Jeśli przewodnictwo cieplne jest anizotropowe \mathbf{k} staje się tensorem przewodnictwa cieplnego w postaci:

$$\mathbf{k} = \begin{bmatrix} k_{xx} & k_{xy} & k_{xz} \\ k_{yx} & k_{yy} & k_{yz} \\ k_{zx} & k_{zy} & k_{zz} \end{bmatrix} \quad (4.2)$$

W modelu przewodnictwa ciepła i konwekcji przez płyn, równanie ciepła zawiera również człon konwekcyjny. Równanie przewodnictwa i konwekcji cieplnej możemy zapisać w postaci:

$$\rho C_p \frac{\partial T}{\partial t} - \nabla \cdot (-k \nabla T + \rho C_p T \mathbf{u}) = Q \quad (4.3)$$

gdzie \mathbf{u} jest wektorem pola prędkości. To pole może być znane w postaci matematycznego wyrażenie zmiennych niezależnych lub obliczone przez rozwiązanie sprzężonego układu równań przewodnictwa oraz bilansu pędu (równanie Navier-Stokesa dla przepływu nieściśliwego). Wektora strumienia ciepła jest definiowany przez wyraz w nawiasie występujący w równaniu (4.3). W przypadku przepływu ciepła z wykorzystaniem konwekcji i przewodnictwa równanie na wektor strumienia ma postać:

$$\mathbf{q} = -k \nabla T + \rho C_p T \mathbf{u} \quad (4.4)$$

Natomiast, jeśli przepływ ciepła jest wyrażona tylko przez przewodnictwo, \mathbf{q} jest określone przez

$$\mathbf{q} = -k \nabla T \quad (4.5)$$

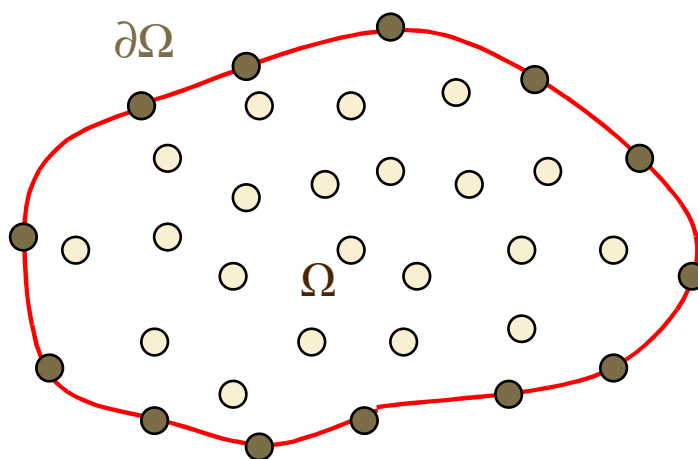
5. Beziatkowa metoda Kansa

Metoda Kansa została wprowadzona w 1990 roku [1] przy rozwiązywaniu cząstkowych równań różniczkowych z kolokacyjnym wykorzystaniem radialnych funkcji bazowych. Ta technika jest bardzo ogólna, prosta i efektywna.

Rozważamy ogólne cząstkowe równanie różniczkowe w obszarze dwuwymiarowym

$$Lu = f(x,y) \text{ w } \Omega \quad Bu = g(x,y) \text{ w } \partial\Omega \quad (5.1)$$

gdzie L – operator różniczkowy, B – operator narzucający warunki graniczne, Ω – obszar wewnętrzny, $\partial\Omega$ – brzegiem obszaru Ω



Rysunek 5.1. Przykładowa graficzna prezentacja obszaru i brzegu.

Wyznaczamy $\{P_i = (x_i, y_i)\}_{i=1}^N$ to będzie N kolokacyjnych punktów w Ω z których $\{(x_i, y_i)\}_{i=1}^{M_1}$ są punktami wewnętrznymi; $\{(x_i, y_i)\}_{i=N_1+1}^N$ są punktami brzegowymi. Na rysunku 5.1 punkty wewnętrzne oznaczone są kolorem jasno-brązowym natomiast punkty brzegowe oznaczone są kolorem brązowym. Kolorem czerwonym oznaczony został obszar nieregularny.

W metodzie Kansa przyjmujemy się, że przybliżone rozwiązanie problemu jest wyrażone przez

$$u(x,y) = \sum_{i=1}^N u_j \varphi_j(x,y) \quad (5.2)$$

gdzie $\{u_j\}_{j=1}^N$ są nieznanymi współczynnikami który będzie wyznaczony, $\varphi_j(x,y) = \varphi(\|P - P_j\|) = \varphi(\sqrt{(x-x_j)^2 + (y-y_j)^2}) = \varphi(r_j)$ jest radialną funkcją bazową. Tutaj $r = \|P - P_j\|$ jest Euklidesową normą pomiędzy punktami $P = (x, y)$ i $P_j = (x_j, y_j)$.

Najczęściej używanymi radialnymi funkcjami bazowymi są:

- Funkcje wielomianowe (ang. the multiquadric) (MQ) $\varphi(r) = (r^2 + c^2)^{\beta/2}$ (β jest parametrem; nieparzystą liczbą całkowitą)
- Funkcje gaussowska (ang. the Gaussians) (GS) $\varphi(r) = e^{-c r^2}$
- Poliharmoniczne funkcje sklejane (ang. the polyharmonic splinem) $\varphi(r) = r^{2n} \log r$ w \mathfrak{R}^2 dla $n \geq 1$ (dla $n=1$ mamy tzw. funkcję cienkich płyt)

Przez podstawienie równania (5.2) do równania (5.1), otrzymujemy

$$\sum_{j=1}^N (L\varphi_j)(x_i, y_i) u_j = f(x_i, y_i), \quad i = 1, 2, \dots, N1 \quad (5.3)$$

$$\sum_{j=1}^N (B\varphi_j)(x_i, y_i) u_j = g(x_i, y_i), \quad i = N1+1, N1+2, \dots, N \quad (5.4)$$

W związku z tym rozwiązujemy następujący układ liniowo - algebraiczny $N \times N$

$$\begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix} \begin{bmatrix} \mathbf{u}^{(1)} \\ \mathbf{u}^{(2)} \end{bmatrix} = \begin{bmatrix} \mathbf{f}^{(1)} \\ \mathbf{f}^{(2)} \end{bmatrix} \quad (5.5)$$

Dla nieznanych $\mathbf{u}^{(1)} = [u_1, u_2, \dots, u_{N1}]^T$, $\mathbf{u}^{(2)} = [u_{N1+1}, u_{N1+2}, \dots, u_N]^T$. Wtedy przybliżone rozwiązanie możemy otrzymać z równania (5.2) w dowolnym punkcie dziedziny Ω . Tutaj otrzymujemy podmacierze:

A_{11} z elementem $(L\varphi_j)(x_i, y_i)$, $i, j = 1, 2, \dots, N1$

A_{12} z elementem $(L\varphi_j)(x_i, y_i)$, $i = 1, 2, \dots, N1$, $j = N1+1, \dots, N$

A_{21} z elementem $(B\varphi_j)(x_i, y_i)$, $i = N1+1, \dots, N$, $j = 1, 2, \dots, N1$

A_{22} z elementem $(B\varphi_j)(x_i, y_i)$, $i, j = N1+1, \dots, N$

I wektor

$$\begin{aligned} \mathbf{f}^{(1)} &= [f(x_1, y_1), f(x_2, y_2), \dots, f(x_{N1}, y_{N1})]^T \\ \mathbf{f}^{(2)} &= [g(x_{N1+1}, y_{N1+1}), g(x_{N1+2}, y_{N1+2}), \dots, g(x_N, y_N)]^T \end{aligned} \quad (5.6)$$

5.1. Zagadnienie stacjonarne

Dla numerycznego sprawdzenia, skupiamy rozwiązanie przy następującym równaniu

$$\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} = f(x, y), \quad (x, y) \in \Omega \subseteq \mathfrak{R}^2 \quad (5.7)$$

$$u|_{\partial\Omega} = g(x,y), \quad (x,y) \in \Omega \quad (5.8)$$

Dla problemu eliptycznego – stacjonarnego, w metodzie Kansa rozwiązanie u jest przybliżane przez

$$u(x,y) = \sum_{j=1}^N u_j \varphi_j(x,y) \quad (5.9)$$

gdzie $\{u_j\}_{j=1}^N$ są nieznanymi współczynnikami który będzie wyznaczony.

Podstawiając równanie (5.9) do równania (5.7) i (5.8), otrzymujemy

$$\sum_{j=1}^N \left(\frac{\partial^2 \varphi_j}{\partial x^2} + \frac{\partial^2 \varphi_j}{\partial y^2} \right) (x_i, y_i) u_j = f(x_i, y_i), \quad i = 1, 2, \dots, N1 \quad (5.10)$$

$$\sum_{j=1}^N \varphi_j(x_i, y_i) u_j = g(x_i, y_i), \quad i = N1+1, \dots, N. \quad (5.11)$$

Równania (5.10)-(5.11) przedstawiają układ równań liniowych (o wymiarze $N \times N$) z nieznanymi współczynnikami $\{u_j\}_{j=1}^N$. Zauważmy, że równanie (5.9) daje globalne przybliżenie rozwiązania równania (5.1) dla każdego punktu w dziedzinie Ω .

5.2. Zagadnienie niestacjonarne

Niestacjonarny przepływ ciepła opisany jest następującym równaniem różniczkowym cząstkowym typu parabolicznego

$$\frac{\partial u}{\partial t} - \alpha \left(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} \right) = f(x,y,t,u,u_x,u_y), \quad (x,y) \in \Omega \subseteq \mathcal{R}^2, \quad 0 \leq t \leq T \quad (5.12)$$

z warunkiem brzegowymi

$$u|_{\partial\Omega_1} = g_1(x,y,t), \quad (x,y) \in \partial\Omega_1 \quad (5.13)$$

$$\frac{\partial u}{\partial n} \Big|_{\partial\Omega_2} = g_2(x,y,t), \quad (x,y) \in \partial\Omega_2 \quad (5.14)$$

gdzie n jest wektorem normalnym do rozważanego brzegu a $\partial\Omega = \partial\Omega_1 + \partial\Omega_2$.

Zagadnienie rozważane jest z następującym warunkiem początkowym

$$u|_{t=0} = h(x,y), \quad (x,y) \in \Omega \quad (5.15)$$

W problemie opisanym równaniem (5.12) możemy pochodną czasową zastąpić (przybliżyć) jawnym schematem różnicowym,

$$\frac{u^{n+1} - u^n}{\delta t} - \alpha \left(\frac{\partial^2 u^{n+1}}{\partial x^2} + \frac{\partial^2 u^{n+1}}{\partial y^2} \right) = f(x, y, t_n, u^n, u_x^n, u_y^n) \quad (5.16)$$

gdzie δt jest krokiem czasowym i u^n, u^{n+1} oznacza odpowiednio rozwiązanie w czasie $t_n = n\delta t$ i $t_{n+1} = (n+1)\delta t$. Przybliżone rozwiązanie dla problemu parabolicznego – niestacjonarnego (5.12), (5.13) i (5.15), będzie wyrażane wtedy jako

$$u(x, y, t_{n+1}) = \sum_{j=1}^N u_j^{n+1} \varphi_j(x, y) \quad (5.17)$$

gdzie $\{u_j^{n+1}\}_{j=1}^N$ są nieznanymi współczynnikami, które będą wyznaczone w każdym kroku czasowym $t = t_{n+1}$. W pracy rozwiązywane będą zagadnienia z warunkiem brzegowym (5.13) dla $\partial\Omega = \partial\Omega_1$ oraz $g(x, y, t) = g_1(x, y, t)$.

Podstawiając równanie (5.17) do równania (5.12) i (5.13), otrzymujemy

$$\sum_{j=1}^N \left(\frac{1}{\delta t} \varphi_j - \alpha \left(\frac{\partial^2 \varphi_j}{\partial x^2} + \frac{\partial^2 \varphi_j}{\partial y^2} \right) \right) (x_i, y_i) u_j^{n+1} = \frac{1}{\delta t} u^n(x_i, y_i) + f(x_i, y_i, t_n, u^n(x_i, y_i), u_x^n(x_i, y_i), u_y^n(x_i, y_i)), \quad i = 1, 2, \dots, N1 \quad (5.18)$$

$$\sum_{j=1}^N \varphi_j(x_i, y_i) u_j^{n+1} = g(x_i, y_i, t_{n+1}), \quad i = N1+1, \dots, N. \quad (5.19)$$

Równania (5.18) i (5.19) reprezentują liniowy układ równań z nieznanymi wartościami współczynników $\{u_j^{n+1}\}_{j=1}^N$. Po rozwiązaniu układu (5.18)–(5.19) i wykorzystaniu wyrażenia (5.17) otrzymujemy przybliżone rozwiązanie w każdym punkcie obszaru Ω .

5.3 Operatory różniczkowe dla radialnych funkcji bazowych dla przypadku stacjonarnego

5.3.1. Funkcja wielomianowa

Dla funkcji wielomianowej (ang. Multiquadric (MQ)) określonej równaniem

$$\varphi_j = \varphi(r_j) = \sqrt{r_j^2 + c^2} \quad (5.20)$$

gdzie

$$r_j = r_j(x, y) = \sqrt{(x - x_j)^2 + (y - y_j)^2} \quad (5.21)$$

otrzymujemy następujące wyrażenia na pierwsze i drugie pochodne funkcji wielomianowej (5.20)

$$\begin{aligned}\frac{\partial \varphi_j}{\partial x} &= \frac{x - x_j}{\sqrt{r_j + c^2}}, & \frac{\partial \varphi_j}{\partial y} &= \frac{y - y_j}{\sqrt{r_j + c^2}} \\ \frac{\partial^2 \varphi_j}{\partial x^2} &= \frac{(y - y_j)^2 + c^2}{(r_j + c^2)^{3/2}}, & \frac{\partial^2 \varphi_j}{\partial y^2} &= \frac{(x - x_j)^2 + c^2}{(r_j + c^2)^{3/2}}\end{aligned}\quad (5.22)$$

Wykorzystując równania (5.22) w operatorze $L\varphi_j$ równania (5.3) otrzymujemy

$$\begin{aligned}rs_{ij} &= r_j^2 = (x_i - x_j)^2 + (y_i - y_j)^2 \\ \sum_{j=1}^N \left(\frac{rs_{ij} + 2c^2}{(r_j + c^2)^{3/2}} \right) u_j &= f(x_i, y_i), \quad i = 1, 2, \dots, N1\end{aligned}\quad (5.23)$$

Natomiast wstawiając wyrażenia (5.20) do wyrażenia (5.4) w miejsce operatora różniczkowego otrzymujemy

$$\sum_{j=1}^N (\sqrt{rs_{ij} + c^2}) u_j = g(x_i, y_i), \quad i = N1+1, N1+2, \dots, N \quad (5.24)$$

Należy tutaj zauważyć że wyrażenie (5.23) odnosi się do punktów wewnętrznych, natomiast wyrażenie (5.24) do punktów znajdujących się na brzegu. Wykorzystując równania (5.23) oraz (5.24) możemy stworzyć układ (5.5). Kolejne elementy podmacierzy macierzy A (występującej po lewej stronie układu) przyjmą następującą postać:

$$\begin{aligned}A_{11(i,j)} &= L\varphi_j(x_i, y_i) = \left(\frac{rs_{ij} + 2c^2}{(r_{ij} + c^2)^{3/2}} \right), \\ i &= 1, 2, \dots, N1 \\ A_{12(i,j)} &= L\varphi_j(x_i, y_i) = \left(\frac{rs_{ij} + 2c^2}{(r_{ij} + c^2)^{3/2}} \right), \\ i &= 1, 2, \dots, N1; j = N1+1, \dots, N \\ A_{21(i,j)} &= B\varphi_j(x_i, y_i) = (\sqrt{rs_{ij} + c^2}), \\ i &= N1+1, \dots, N; j = 1, 2, \dots, N1 \\ A_{22(i,j)} &= B\varphi_j(x_i, y_i) = (\sqrt{rs_{ij} + c^2}), \\ i &= N1+1, \dots, N; j = N1+1, \dots, N\end{aligned}\quad (5.25)$$

Natomiast wektor wyrazów wolnych f (występujący po prawej stronie układu) możemy zapisać w następującej postaci:

$$\begin{aligned}\mathbf{f}^{(1)} &= [f(x_1, y_1), f(x_2, y_2), \dots, f(x_{N1}, y_{N1})]^T \\ \mathbf{f}^{(2)} &= [g(x_{N1+1}, y_{N1+1}), g(x_{N1+2}, y_{N1+2}), \dots, g(x_N, y_N)]^T\end{aligned}\quad (5.26)$$

5.3.2 Funkcja gaussowska

Dla funkcji gaussowskiej (ang. the Gaussians (GS)) określonej równaniem

$$\varphi_j = \varphi(r_j) = e^{-c r_j^2} \quad (5.27)$$

gdzie

$$r_j = r_j(x, y) = \sqrt{(x - x_j)^2 + (y - y_j)^2} \quad (5.28)$$

otrzymujemy następujące wyrażenia na pierwsze i drugie pochodne funkcji gaussowskiej (5.27)

$$\begin{aligned} r_{s_{i,j}} &= r_j^2 = (x_i - x_j)^2 + (y_i - y_j)^2 \\ \frac{\partial \varphi_j}{\partial x} &= -2ce^{-c(rs_{i,j})}(x_i - x_j), \\ \frac{\partial \varphi_j}{\partial y} &= -2ce^{-c(rs_{i,j})}(y_i - y_j), \\ \frac{\partial^2 \varphi_j}{\partial x^2} &= -2ce^{-c(rs_{i,j})} + 4c^2 e^{-c(rs_{i,j})}(x_i - x_j), \\ \frac{\partial^2 \varphi_j}{\partial y^2} &= -2ce^{-c(rs_{i,j})} + 4c^2 e^{-c(rs_{i,j})}(y_i - y_j), \end{aligned} \quad (5.29)$$

Wykorzystując równania (5.29) w operatorze $L\varphi_j$ równania (5.3) otrzymujemy

$$\sum_{j=1}^N \left(4ce^{-c(rs_{i,j})} (c(rs_{i,j}) - 1) \right) u_j = f(x_i, y_i), \quad i = 1, 2, \dots, N1 \quad (5.30)$$

Natomiast wstawiając wyrażenia (5.27) do wyrażenia (5.4) w miejsce operatora różniczkowego otrzymujemy

$$\sum_{j=1}^N \left(e^{-c(rs_{i,j})} \right) u_j = g(x_i, y_i), \quad i = N1+1, N1+2, \dots, N \quad (5.31)$$

Należy tutaj zauważyć że wyrażenie (5.30) odnosi się do punktów wewnętrznych, natomiast wyrażenie (5.31) do punktów znajdujących się na brzegu. Wykorzystując równania (5.30) oraz (5.31) możemy stworzyć układ (5.5). Kolejne elementy podmacierzy macierzy A (występującej po lewej stronie układu) przyjmą następującą postać:

$$\begin{aligned}
rs_{ij} &= r_j^2 = (x_i - x_j)^2 + (y_i - y_j)^2 \\
A_{11(i,j)} &= L\varphi_j(x_i, y_i) = \left(4ce^{-c(rs_{ij})}(c(rs_{ij}) - 1)\right), \\
i &= 1, 2, \dots, N1 \\
A_{12(i,j)} &= L\varphi_j(x_i, y_i) = \left(4ce^{-c(rs_{ij})}(c(rs_{ij}) - 1)\right), \\
i &= 1, 2, \dots, N1; j = N1 + 1, \dots, N \\
A_{21(i,j)} &= B\varphi_j(x_i, y_i) = \left(e^{-c(rs_{ij})}\right), \\
i &= N1 + 1, \dots, N; j = 1, 2, \dots, N1 \\
A_{22(i,j)} &= B\varphi_j(x_i, y_i) = \left(e^{-c(rs_{ij})}\right), \\
i &= N1 + 1, \dots, N; j = N1 + 1, \dots, N
\end{aligned} \tag{5.32}$$

Natomiast wektor wyrazów wolnych f (występujący po prawej stronie układu) możemy zapisać w następującej postaci:

$$\begin{aligned}
\mathbf{f}^{(1)} &= [f(x_1, y_1), f(x_2, y_2), \dots, f(x_{N1}, y_{N1})]^T \\
\mathbf{f}^{(2)} &= [g(x_{N1+1}, y_{N1+1}), g(x_{N1+2}, y_{N1+2}), \dots, g(x_N, y_N)]^T
\end{aligned} \tag{5.33}$$

5.3.3 Funkcja poliharmoniczna

Dla poliharmonicznych funkcji sklejaných (ang. the polyharmonic splines) określonych równaniem

$$\varphi(r) = r^{2n} \log r \tag{5.34}$$

gdzie

$$r_j = r_j(x, y) = \sqrt{(x - x_j)^2 + (y - y_j)^2} \tag{5.35}$$

otrzymujemy następujące wyrażenia na pierwsze i drugie pochodne funkcji poliharmonicznej (5.34)

$$\begin{aligned}
rs_{ij} &= r_j^2 = (x_i - x_j)^2 + (y_i - y_j)^2 \\
\frac{\partial \varphi_j}{\partial x} &= (x_i - x_j)(rs_{ij})^{n-1}(1 + n \log rs_{ij}), \\
\frac{\partial \varphi_j}{\partial y} &= (rs_{ij})^{n-1}(y_i - y_j)(1 + n \log rs_{ij}), \\
\frac{\partial^2 \varphi_j}{\partial x^2} &= 4n(x_i - x_j)^2(rs_{ij})^{n-2} + \left(-\frac{2(x_i - x_j)^2}{(rs_{ij})^2} + \frac{1}{rs_{ij}}\right)(rs_{ij})^n + \\
&\quad (4(n-1)n(x_i - x_j)^2(rs_{ij})^{n-2} + 2n(rs_{ij})^{n-1})\log(\sqrt{rs_{ij}}), \\
\frac{\partial^2 \varphi_j}{\partial y^2} &= (rs_{ij})^n \left(\frac{1}{rs_{ij}} - \frac{2(y_i - y_j)^2}{(rs_{ij})^2}\right) + 4n(y_i - y_j)^2(rs_{ij})^{n-2} + \\
&\quad (2n(rs_{ij})^{n-1} + 4(n-1)n(y_i - y_j)^2(rs_{ij})^{n-2})\log(\sqrt{rs_{ij}}),
\end{aligned} \tag{5.36}$$

Wykorzystując równania (5.36) w operatorze $L\varphi_j$ równania (5.3) otrzymujemy

$$\sum_{j=1}^N \left(2n(rs_{i,j})^{(n-1)}(2 + n \log rs_{i,j}) \right) u_j = f(x_i, y_i), \quad i = 1, 2, \dots, N1 \quad (5.37)$$

Natomiast wstawiając wyrażenia (5.34) do wyrażenia (5.4) w miejsce operatora różniczkowego otrzymujemy

$$\sum_{j=1}^N rs_{i,j} \log(\sqrt{rs_{i,j}}) u_j = g(x_i, y_i), \quad i = N1+1, N1+2, \dots, N \quad (5.38)$$

Należy tutaj zauważyć że wyrażenie (5.37) odnosi się do punktów wewnętrznych, natomiast wyrażenie (5.38) do punktów znajdujących się na brzegu. Wykorzystując równania (5.37) oraz (5.38) możemy stworzyć układ (5.5). Kolejne elementy podmacierzy macierzy A (występującej po lewej stronie układu) przyjmą następującą postać:

$$\begin{aligned} A_{11(i,j)} &= L\varphi_j(x_i, y_i) = \left(2n(rs_{i,j})^{(n-1)}(2 + n \log rs_{i,j}) \right), \\ i &= 1, 2, \dots, N1 \\ A_{12(i,j)} &= L\varphi_j(x_i, y_i) = \left(2n(rs_{i,j})^{(n-1)}(2 + n \log rs_{i,j}) \right), \\ i &= 1, 2, \dots, N1; j = N1+1, \dots, N \\ A_{21(i,j)} &= B\varphi_j(x_i, y_i) = (rs_{i,j}) \log(\sqrt{rs_{i,j}}), \\ i &= N1+1, \dots, N; j = 1, 2, \dots, N1 \\ A_{22(i,j)} &= B\varphi_j(x_i, y_i) = (rs_{i,j}) \log(\sqrt{rs_{i,j}}), \\ i &= N1+1, \dots, N; j = N1+1, \dots, N \end{aligned} \quad (5.39)$$

Natomiast wektor wyrazów wolnych f (występujący po prawej stronie układu) możemy zapisać w następującej postaci:

$$\begin{aligned} \mathbf{f}^{(1)} &= [f(x_1, y_1), f(x_2, y_2), \dots, f(x_{N1}, y_{N1})]^T \\ \mathbf{f}^{(2)} &= [g(x_{N1+1}, y_{N1+1}), g(x_{N1+2}, y_{N1+2}), \dots, g(x_N, y_N)]^T \end{aligned} \quad (5.40)$$

5.4 Operatory różniczkowe dla radialnych funkcji bazowych dla przypadku niestacjonarnego

5.4.1 Funkcja wielomianowa

Dla funkcji wielomianowej (ang. Multiquadric (MQ)) określonej równaniem

$$\varphi_j = \varphi(r_j) = \sqrt{r_j^2 + c^2} \quad (5.41)$$

gdzie

$$r_j = r_j(x,y) = \sqrt{(x - x_j)^2 + (y - y_j)^2} \quad (5.42)$$

otrzymujemy następujące wyrażenia na pierwsze i drugie pochodne funkcji wielomianowej (5.43)

$$\begin{aligned} \frac{\partial \varphi_j}{\partial x} &= \frac{x - x_j}{\sqrt{r_j + c^2}}, & \frac{\partial \varphi_j}{\partial y} &= \frac{y - y_j}{\sqrt{r_j + c^2}} \\ \frac{\partial^2 \varphi_j}{\partial x^2} &= \frac{(y - y_j)^2 + c^2}{(r_j + c^2)^{3/2}}, & \frac{\partial^2 \varphi_j}{\partial y^2} &= \frac{(x - x_j)^2 + c^2}{(r_j + c^2)^{3/2}} \end{aligned} \quad (5.43)$$

W równaniu (5.18) w miejsce drugich pochodnych podstawiamy wyrażenia (5.43) oraz w miejsce φ_j wstawiamy równanie (5.41) i otrzymujemy

$$\begin{aligned} rs_{ij} &= r_j^2 = (x_i - x_j)^2 + (y_i - y_j)^2 \\ \sum_{j=1}^N \left(\frac{1}{\delta t} \sqrt{rs_{ij} + c^2} - \alpha \left(\frac{rs_{ij} + 2c^2}{(r_j + c^2)^{3/2}} \right) \right) u_j^{n+1} &= \frac{1}{\delta t} u^n(x_i, y_i) + f(x_i, y_i, t_n), \\ i &= 1, 2, \dots, N1 \end{aligned} \quad (5.44)$$

Natomiast wstawiając wyrażenia (5.41) do wyrażenia (5.19) w miejsce operatora różniczkowego otrzymujemy

$$\sum_{j=1}^N (\sqrt{rs_{ij} + c^2}) u_j^{n+1} = g(x_i, y_i, t_{n+1}), \quad i = N1 + 1, \dots, N \quad (5.45)$$

Należy tutaj zauważyć że wyrażenie (5.44) odnosi się do punktów wewnętrznych, natomiast wyrażenie (5.45) do punktów znajdujących się na brzegu. Wykorzystując równania (5.44) oraz (5.45) możemy stworzyć układ (5.5). Kolejne elementy podmacierzy macierzy A (występującej po lewej stronie układu) rzyjmą następującą postać:

$$\begin{aligned} A_{11(i,j)} &= \left(\frac{1}{\delta t} \sqrt{rs_{ij} + c^2} - \alpha \left(\frac{rs_{ij} + 2c^2}{(r_j + c^2)^{3/2}} \right) \right), \\ i &= 1, 2, \dots, N1 \\ A_{12(i,j)} &= \left(\frac{1}{\delta t} \sqrt{rs_{ij} + c^2} - \alpha \left(\frac{rs_{ij} + 2c^2}{(r_j + c^2)^{3/2}} \right) \right), \\ i &= 1, 2, \dots, N1; j = N1 + 1, \dots, N \\ A_{21(i,j)} &= \left(\sqrt{rs_{ij} + c^2} \right), \\ i &= N1 + 1, \dots, N; j = 1, 2, \dots, N1 \\ A_{22(i,j)} &= \left(\sqrt{rs_{ij} + c^2} \right), \\ i &= N1 + 1, \dots, N; j = N1 + 1, \dots, N \end{aligned} \quad (5.46)$$

Natomiast wektor wyrazów wolnych f (występujący po prawej stronie układu) możemy zapisać w następującej postaci:

$$\begin{aligned}\mathbf{f}^{(1)} &= [f(x_1, y_1), f(x_2, y_2), \dots, f(x_{N_1}, y_{N_1})]^T \\ \mathbf{f}^{(2)} &= [g(x_{N_1+1}, y_{N_1+1}), g(x_{N_1+2}, y_{N_1+2}), \dots, g(x_N, y_N)]^T\end{aligned}\quad (5.47)$$

5.4.2 Funkcja gaussowska

Dla funkcji gaussowskiej (ang. the Gaussians (GS)) określonej równaniem

$$\varphi_j = \varphi(r_j) = e^{-c r_j^2} \quad (5.48)$$

gdzie

$$r_j = r_j(x, y) = \sqrt{(x - x_j)^2 + (y - y_j)^2} \quad (5.49)$$

otrzymujemy następujące wyrażenia na pierwsze i drugie pochodne funkcji gaussowskiej (5.50)

$$\begin{aligned}rs_{i,j} &= r_j^2 = (x_i - x_j)^2 + (y_i - y_j)^2 \\ \frac{\partial \varphi_j}{\partial x} &= -2ce^{-c(rs_{i,j})}(x_i - x_j), \\ \frac{\partial \varphi_j}{\partial y} &= -2ce^{-c(rs_{i,j})}(y_i - y_j), \\ \frac{\partial^2 \varphi_j}{\partial x^2} &= -2ce^{-c(rs_{i,j})} + 4c^2 e^{-c(rs_{i,j})}(x_i - x_j)^2, \\ \frac{\partial^2 \varphi_j}{\partial y^2} &= -2ce^{-c(rs_{i,j})} + 4c^2 e^{-c(rs_{i,j})}(y_i - y_j)^2,\end{aligned}\quad (5.50)$$

W równaniu (5.18) w miejsce drugich pochodnych podstawiamy wyrażenia (5.50) oraz w miejsce φ_j wstawiamy równanie (5.48) i otrzymujemy

$$\begin{aligned}\sum_{j=1}^N \left(\frac{1}{\delta t} (e^{-c(rs_{i,j})}) - \alpha (4ce^{-c(rs_{i,j})}(c(rs_{i,j}) - 1)) \right) u_j^{n+1} = \\ \frac{1}{\delta t} u^n(x_i, y_i) + f(x_i, y_i, t_n), \quad i = 1, 2, \dots, N1\end{aligned}\quad (5.51)$$

Natomiast wstawiając wyrażenia (5.48) do wyrażenia (5.19) w miejsce operatora różniczkowego otrzymujemy

$$\sum_{j=1}^N (e^{-c(rs_{i,j})}) u_j^{n+1} = g(x_i, y_i, t_{n+1}), \quad i = N1+1, \dots, N \quad (5.52)$$

Należy tutaj zauważyć że wyrażenie (5.51) odnosi się do punktów wewnętrznych, natomiast wyrażenie (5.52) do punktów znajdujących się na brzegu. Wykorzystując

równania (5.51) oraz (5.52) możemy stworzyć układ (5.5). Kolejne elementy podmacierzy macierzy A (występującej po lewej stronie układu) przyjmą następującą postać:

$$\begin{aligned}
 rs_{ij} &= r_j^2 = (x_i - x_j)^2 + (y_i - y_j)^2 \\
 A_{11(i,j)} &= \left(\frac{1}{\delta t} \left(e^{-c(rs_{ij})} \right) - \alpha \left(4ce^{-c(rs_{ij})} (c(rs_{ij} - 1)) \right) \right), \\
 i &= 1, 2, \dots, N1 \\
 A_{12(i,j)} &= \left(\frac{1}{\delta t} \left(e^{-c(rs_{ij})} \right) - \alpha \left(4ce^{-c(rs_{ij})} (c(rs_{ij} - 1)) \right) \right), \\
 i &= 1, 2, \dots, N1; j = N1 + 1, \dots, N \\
 A_{21(i,j)} &= \left(e^{-c(rs_{ij})} \right), \\
 i &= N1 + 1, \dots, N; j = 1, 2, \dots, N1 \\
 A_{22(i,j)} &= \left(e^{-c(rs_{ij})} \right), \\
 i &= N1 + 1, \dots, N; j = N1 + 1, \dots, N
 \end{aligned} \tag{5.53}$$

Natomiast wektor wyrazów wolnych \mathbf{f} (występujący po prawej stronie układu) możemy zapisać w następującej postaci:

$$\begin{aligned}
 \mathbf{f}^{(1)} &= [f(x_1, y_1), f(x_2, y_2), \dots, f(x_{N1}, y_{N1})]^T \\
 \mathbf{f}^{(2)} &= [g(x_{N1+1}, y_{N1+1}), g(x_{N1+2}, y_{N1+2}), \dots, g(x_N, y_N)]^T
 \end{aligned} \tag{5.54}$$

5.4.3 Funkcja poliharmoniczne

Dla poliharmonicznych funkcji sklejaných (ang. the polyharmonic splinem) określonych równaniem

$$\varphi(r) = r^{2n} \log r \tag{5.55}$$

gdzie

$$r_j = r_j(x, y) = \sqrt{(x - x_j)^2 + (y - y_j)^2} \tag{5.56}$$

otrzymujemy następujące wyrażenia na pierwsze i drugie pochodne funkcji poliharmonicznej (5.57)

$$\begin{aligned}
rs_{ij} &= r_j^2 = (x_i - x_j)^2 + (y_i - y_j)^2 \\
\frac{\partial \varphi_j}{\partial x} &= (x_i - x_j)(rs_{ij})^{n-1}(1 + n \log rs_{ij}), \\
\frac{\partial \varphi_j}{\partial y} &= (rs_{ij})^{n-1}(y_i - y_j)(1 + n \log rs_{ij}), \\
\frac{\partial^2 \varphi_j}{\partial x^2} &= 4n(x_i - x_j)^2(rs_{ij})^{n-2} + \left(-\frac{2(x_i - x_j)^2}{(rs_{ij})^2} + \frac{1}{rs_{ij}} \right) (rs_{ij})^n + \\
&\quad (4(n-1)n(x_i - x_j)^2(rs_{ij})^{n-2} + 2n(rs_{ij})^{n-1}) \log(\sqrt{rs_{ij}}), \\
\frac{\partial^2 \varphi_j}{\partial y^2} &= (rs_{ij})^n \left(\frac{1}{rs_{ij}} - \frac{2(y_i - y_j)^2}{(rs_{ij})^2} \right) + 4n(y_i - y_j)^2(rs_{ij})^{n-2} + \\
&\quad (2n(rs_{ij})^{n-1} + 4(n-1)n(y_i - y_j)^2(rs_{ij})^{n-2}) \log(\sqrt{rs_{ij}}),
\end{aligned} \tag{5.57}$$

W równaniu (5.18) w miejsce drugich pochodnych podstawiamy wyrażenia (5.57) oraz w miejsce φ_j wstawiamy równanie (5.55) i otrzymujemy

$$\begin{aligned}
\sum_{j=1}^N \left(\frac{1}{\delta t} (rs_{ij} \log(\sqrt{rs_{ij}})) - \alpha (2n(rs_{ij})^{n-1}(2 + n \log rs_{ij})) \right) u_j^{n+1} = \\
\frac{1}{\delta t} u^n(x_i, y_i) + f(x_i, y_i, t_n), \quad i = 1, 2, \dots, N1
\end{aligned} \tag{5.58}$$

Natomiast wstawiając wyrażenia (5.55) do wyrażenia (5.19) w miejsce operatora różniczkowego otrzymujemy

$$\sum_{j=1}^N (rs_{ij} \log(\sqrt{rs_{ij}})) u_j^{n+1} = g(x_i, y_i, t_{n+1}), \quad i = N1 + 1, \dots, N \tag{5.59}$$

Należy tutaj zauważyć że wyrażenie (5.58) odnosi się do punktów wewnętrznych, natomiast wyrażenie (5.59) do punktów znajdujących się na brzegu. Wykorzystując równania (5.58) oraz (5.59) możemy stworzyć układ (5.5). Kolejne elementy podmacierzy macierzy A (występującej po lewej stronie układu) przyjmą następującą postać:

$$\begin{aligned}
A_{11(i,j)} &= \left(\frac{1}{\delta t} (rs_{ij} \log(\sqrt{rs_{ij}})) - \alpha (2n(rs_{ij})^{n-1}) (2 + n \log rs_{ij}) \right), \\
i &= 1, 2, \dots, N1 \\
A_{12(i,j)} &= \left(\frac{1}{\delta t} (rs_{ij} \log(\sqrt{rs_{ij}})) - \alpha (2n(rs_{ij})^{n-1}) (2 + n \log rs_{ij}) \right), \\
i &= 1, 2, \dots, N1; j = N1 + 1, \dots, N \\
A_{21(i,j)} &= (rs_{ij}) \log(\sqrt{rs_{ij}}), \\
i &= N1 + 1, \dots, N; j = 1, 2, \dots, N1 \\
A_{22(i,j)} &= (rs_{ij}) \log(\sqrt{rs_{ij}}), \\
i &= N1 + 1, \dots, N; j = N1 + 1, \dots, N
\end{aligned} \tag{5.60}$$

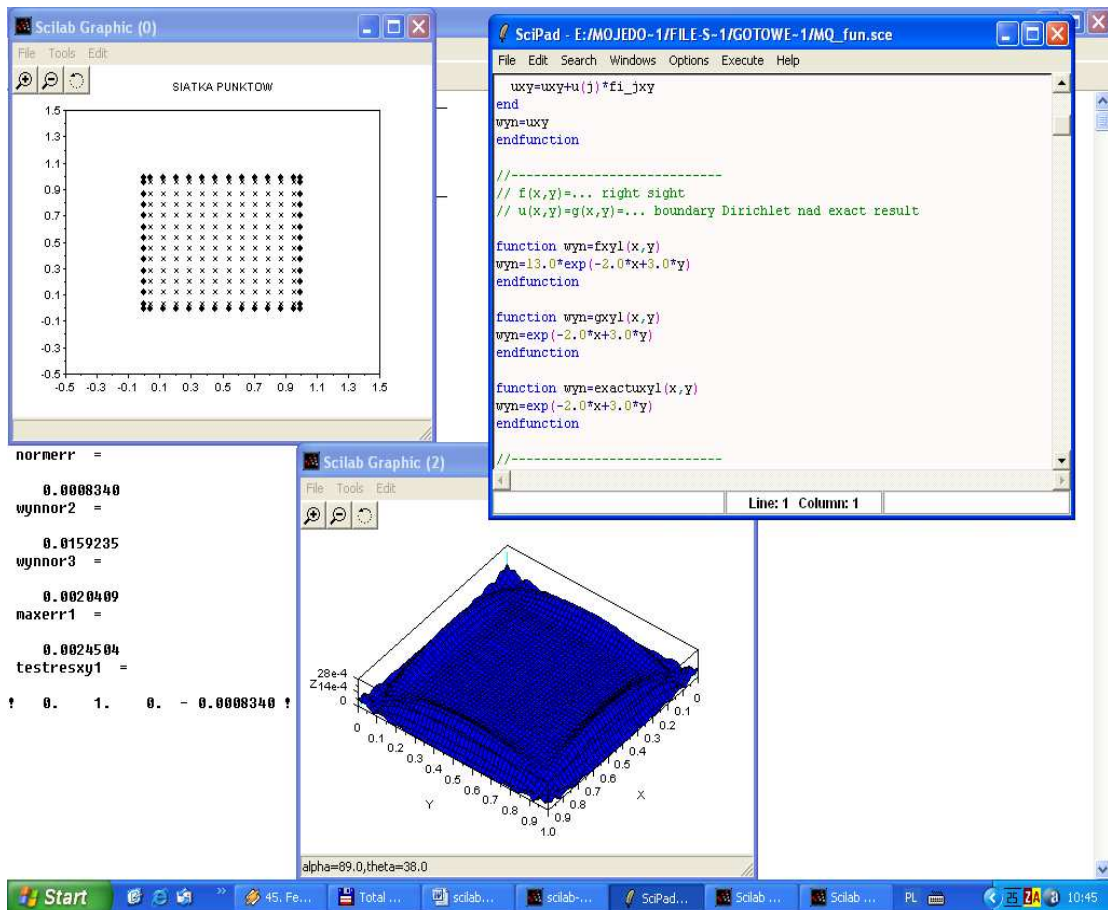
Natomiast wektor wyrazów wolnych \mathbf{f} (występujący po prawej stronie układu) możemy zapisać w następującej postaci:

$$\begin{aligned}
\mathbf{f}^{(1)} &= [f(x_1, y_1), f(x_2, y_2), \dots, f(x_{N1}, y_{N1})]^T \\
\mathbf{f}^{(2)} &= [g(x_{N1+1}, y_{N1+1}), g(x_{N1+2}, y_{N1+2}), \dots, g(x_N, y_N)]^T
\end{aligned} \tag{5.61}$$

Jak można zauważyć powyżej, implementacja metody Kansa jest nieskomplikowana. To są główne zalety ze ta technika staje się popularna i została zastosowana w wielu dziedzinach nauki, w których zjawiska są opisywane cząstkowymi równaniami różniczkowymi. Jakkolwiek dowód rozwiązywalności dla liniowego systemu wynikającego z metody jest nadal niepotwierdzony, nawet dla eliptycznego problemu.

6. Narzędzie programistyczne - Scilab

Scilab [10] jest odpowiednikiem pakietu naukowego Matlab. Wyposażony jest w setki gotowych funkcji do operacji algebraicznych, rozwiązywania równań liniowych i nieliniowych, grafiki dwu- i trójwymiarowej oraz wiele innych. Został opracowany w INRIA (Institut National de Recherche en Informatique et en Automatique) we Francji, dostępny jest od 1994 roku. Należy do programów rodziny CAS (Computer Algebra Systems), wspomagających obliczenia naukowe. Choć jest opatrzony licencją GPL, Scilab w niczym nie ustępuje - a często przewyższa - oprogramowanie komercyjne [10]. Szczegółowy opis środowiska dostępny jest w bogatej literaturze dostępnej w internecie, jak choćby na stronie producenta Scilaba [11].



Rys1. Interface programu SciLab w czasie wykonywanego programu.

Podstawowym obiektem języka jest macierz. Może to sprawić kłopot początkującemu użytkownikowi, ale dzięki temu przy odpowiednim zaplanowaniu struktury skryptu unikamy stosowania wielu zmiennych skalarnych i zapewniany zwięzłość zapisu. Skalar leksykalnie można traktować jako macierz posiadającą jeden wiersz i jedną kolumnę. W porównaniu z typowymi językami programowania stosowanymi w obliczeniach (np. Fortran, C), nawet słaba znajomość składni wystarcza do rozwiązywania praktycznych problemów. Należy pamiętać, że wielkość liter ma znaczenie. a i A to dwie różne zmienne.

Komentarze w programie mają składnię zapożyczoną z języka programowania C++, czyli każda linijka poprzedzona znakami `//` jest ignorowana. Stałą od zmiennej najłatwiej odróżnić po procencie poprzedzającym nazwę, np. `%pi`, `%e` - stała Eulera, podstawa logarytmu naturalnego; `%i` - jednostka urojona, czyli pierwiastek kwadratowy z -1 ; `%t`, `%f` - prawda i fałsz logiczny; `%inf` to oznaczenie nieskończoności; `%nan` -

wyrażenie nieoznaczone. Istnieje też stała `%eps` określająca precyzję obliczeń; jest to największa liczba, która spełnia równanie $1 + \%eps = 1$. Zmienne w programie nie muszą być deklarowane przed użyciem, jak ma to miejsce w innych językach programowania. Dla przykładu $A1 = [1, 2, 3, 4]$ lub $A2 = [1 \ 2 \ 3 \ 4]$ definiuje nam czteroelementowy wektor wierszowy, a $A3 = [1; 2 \ 3; 4]$ definiuje nam czteroelementowy wektor kolumnowy.

Transpozycja (zamiana wierszy na kolumny) takiego wektora jest prosta - wystarczy apostrof po nazwie zmiennej: $A1'$. Macierz tworzymy w podobny sposób, np. $B = [1, -2; 1, 3]$ (gdzie „średnik” oddziela poszczególne wiersze macierzy a „przecinek” poszczególne wyrazy w wierszu). „Średnik” ma także inne zadanie - oddziela poszczególne instrukcje. Odwrócenie macierzy B możemy uzyskać dzięki procedurze funkcyjnej `inv`. Macierz C będzie macierzą odwrotną macierzy B po wykonaniu polecenia $C = \text{inv}(B)$. Wyznacznik takiej macierzy też możemy otrzymać od razu: `det(C)`. Wymiar zmiennej rozszerza się automatycznie, polecenie `D(3,3)=2`; powoduje powstanie macierzy o 3 wierszach i 3 kolumnach z określonym elementem `D(3,3)` równym 2. Pozostałe elementy tej macierzy są zerowe. Średnik na końcu instrukcji zapobiega wyświetleniu wartości zmiennej. Rozszerzenia wymiarów struktury można dokonać dzięki kolejnemu podstawieniu nieistniejącego elementu `D(4,4)=3`. Po kilku takich podstawieniach trudno zapamiętać, jakie są aktualne wymiary macierzy; do akcji wkracza wtedy funkcja `size`, np. `[wiersz, kolumna] = size(D)`. Zwraca ona dwie liczby całkowite; dla zmiennej skalarnej kolumny i wiersze są równe jedności. Przy większych wymiarach macierzy trudno wprowadzić wartości wszystkich elementów w sposób jawny. Macierz jednostkową (tylko elementy diagonalne równe jedności, reszta zerowa) o wymiarach 100×100 tworzy zapis `DUZA_MACIERZ = eye(100,100)`; macierz złożoną za samych zer `zeroes(100,100)`; wszystkie elementy równe jedności `ones(100,100)`. W bardziej skomplikowanych przypadkach możemy wykorzystać podstawienie w miejsce elementu macierzy innej macierzy, a nie liczby. Funkcja `matrix` modyfikuje macierz podaną jako pierwszy argument i tworzy z niej macierz o wymiarach podanych przez następne dwa argumenty.

Oprócz działań na macierzach, możemy wykonywać instrukcje na ich elementach. Używamy wówczas operatorów poprzedzonych kropką $F = E.^2$ podniesie wszystkie elementy macierzy do kwadratu, zaś zapis $F = E^2$ lub $F = E * E$ spowoduje wykonanie mnożenia macierzowego według reguł algebraicznych. Jeżeli argumentem funkcji jest macierz, to rezultatem będzie zmienna macierzowa np. $F = \cos(E)$. Operator `^` to oczywiście potęgowanie. Funkcja `inv(A)` oblicza macierz odwrotną do macierzy x . Scilab jest w stanie poradzić sobie z wymiarami macierzy rzędu tysięcy. Po ostatniej instrukcji brak przecinka, dzięki czemu program wyświetli wartość zmiennej.

Możemy również każde rozwiązanie przedstawić za pomocą wykresu dwu- lub trójwymiarowego. Służą do tego polecenia `plot2d` i `plot3d`. Każdy wykres pojawia się w oddzielnym okienku, co pozwala obserwować rozwiązania dla wielu punktów i funkcji. Każde okienko można wzbogacić o opisy osi współrzędnych, komentarzy własnych jak i wiele innych rzeczy.

Podczas tworzenia kolejnych zmiennych może zabraknąć pamięci. Są dwa sposoby rozwiązania takiego problemu: zwiększenie dostępnej przestrzeni instrukcją `stacksize` lub usunięcie niepotrzebnych zmiennych. Ilość dostępnej pamięci zależy od wolnej pamięci systemowej i przestrzeni wymiany. Wszystkie (niechronione) zmienne możemy usunąć poleceniem `clear`. Jeśli chcemy usunąć jedynie zmienną a , wystarczy zapis `clear a`. Istnieje też możliwość ustalenia swoich predefiniowanych zmiennych w pliku `.scilab`. Zostaną one automatycznie dołączone do środowiska po ponownym uruchomieniu programu. Jeśli wykonanie procedury obliczeniowej okaże się niemożliwe, to powstaje sytuacja wyjątkowa; na szczęście funkcja `ieee(tryb)` reguluje zachowanie środowiska w takich warunkach.

Istnieje możliwość kompilowania własnych funkcji. Służą do tego polecenie `comp(funkcja)`. Kompilowane i interpretowane funkcje niczym się nie różnią, ale zwykle

te pierwsze działają szybciej. Funkcje dostarczone ze standardowymi bibliotekami Scilaba są skompilowane. Polecenie **getf('funkcje')** ładuje wszystkie zdefiniowane tam funkcje, a następnie je kompiluje. Porównanie szybkości dwóch funkcji - interpretowanej i kompilowanej - można przeprowadzić za pomocą funkcji **timer()**, która wyświetla czas, jaki minął od poprzedniego wywołania.

Dłuższe programy powinny być zapisywane w plikach zewnętrznych, co pozwala na ich rozwijanie i testowanie. Polecenie **exec('test.sci',tryb)** dołącza do środowiska programu instrukcje zawarte w pliku test.sci; jeśli tryb=0, to wyświetlone zostaną tylko wyniki działania (domyślnie), a jeśli tryb=1, to wyświetlone zostaną również wykonywane instrukcje. Standardowym rozszerzeniem takich plików jest sci. Dla zapominalskich twórcy pozostawili jeszcze jedną furtkę - możliwość zapisu sesji interaktywnej: **diary('plik')** rozpoczyna zapis wszystkich wpisanych poleceń i ich wyniki. Jeśli chcemy zakończyć sesję, wystarczy wpisać **diary(0)**. Powstały w ten sposób plik tekstowy możemy analizować w dowolnym edytorze.

Dobrze skonstruowany system pomocy w tak obszernym programie jest niezbędny. Tutaj dokumentacja w formacie Adobe Acrobat liczy ponad 600 stron [2]. Twórcy pakietu nie zapomnieli o użytkownikach. Jeśli mamy problem ze znalezieniem składni danej funkcji, wystarczy wpisać w wierszu poleceń *help nazwa_funkcji*; jeśli interesują nas informacje dotyczące np. operacji wielomianowych, wystarczy **apropos poly**. W łatwy sposób można dowiedzieć się, jakie zmienne zostały dotychczas stworzone - pomocą służy **who**. Dodatkową otrzymaną informacją jest ilość dostępnej i zajętej pamięci. Listę podstawowych instrukcji języka otrzymamy za pomocą polecenia **what**.

Badania naukowe są często niedofinansowane, a jednocześnie rośnie zastosowanie metod obliczeniowych w chemii, fizyce, biologii, naukach technicznych czy informatyce. Wszędzie tam Scilab - w połączeniu z Linuksem - może się okazać wręcz nieodzowny. I to nie tylko ze względu na darmowy charakter, ale także na wielką siłę drzemącą w tym programie i jego wysoką elastyczność.

6.1 Przykłady wykorzystania Scilaba do obliczeń i wizualizacji wyników

Przykład 1

Najprostszym zastosowaniem operacji macierzowych jest rozwiązywanie układu równań liniowych, np. $2x+y+3z=9$ $x-2y+z=-2$ $3x+2y+2z=7$.

W Scilabie zadanie to realizuje następujący fragment programu (kodu)

```
-->A=[2 1 3; 1 -2 1; 3 2 2];
-->B=[9 -2 7]';
-->x=A\B;
-->x=inv(A)*B
x =
! - 1. !
! 2. !
! 3. !
```

W tabelce umieszczono polecenia jak i otrzymane wyniki programu.

Przykład 2

Rozwiązanie macierzy i różne sposoby jej zapisania

```
-->A=[1 2;3 4];
-->B=[5 6;7 8];
-->C=[9 10;11 12];
-->D=[A,B,C];
```



```

-->E=matrix(D,3,4)
E =
! 1.  4.  6.  11.!
! 3.  5.  8.  10.!
! 2.  7.  9.  12.!
-->F=eye(E)
F =
! 1.  0.  0.  0.!
! 0.  1.  0.  0.!
! 0.  0.  1.  0.!
-->G=eye(4,3)
G =
! 1.  0.  0.!
! 0.  1.  0.!
! 0.  0.  1.!
! 0.  0.  0.!
-->H=ones(4,4)
H =
! 1.  1.  1.  1.!
! 1.  1.  1.  1.!
! 1.  1.  1.  1.!
! 1.  1.  1.  1.!
-->I=zeros(4,4)
I =
! 0.  0.  0.  0.!
! 0.  0.  0.  0.!
! 0.  0.  0.  0.!
! 0.  0.  0.  0.!

```

Przykład 3

Zastosowanie wizualizacji w Scilabie

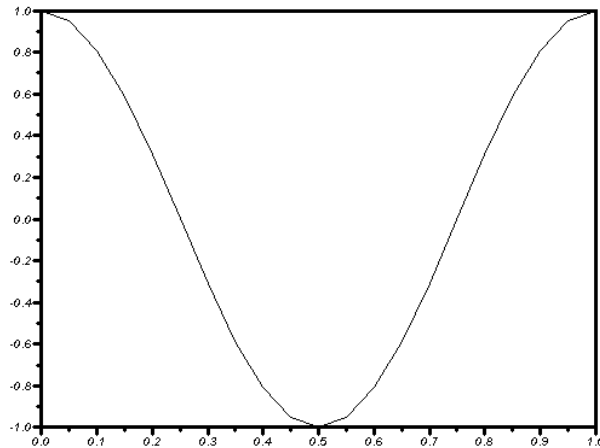
Plot2d

Rozwiązanie funkcji $ct=\cos(2*\pi*t)$ z krokiem czasowym $t=0.05$ w przedziale $(0,1)$. W Scilabie zadanie to realizuje następujący fragment kodu:

```

-->t=(0:0.05:1);
-->ct=cos(2*%pi*t);
-->plot(t,ct);
-->xset("font",5,4);xset("thickness",3);
-->plot(t,ct,'Time','Cosine','Simple Plot')

```

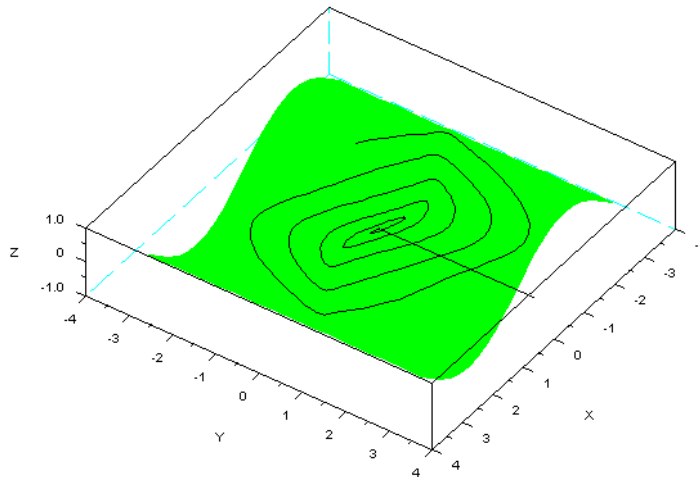


Połączenie plot2d i plot3d

```

-->r=(%pi):-0.01:0;x=r.*cos(10*r);y=r.*sin(10*r);
--> function z=surf(x,y),
--> z=sin(x)*cos(y);
--> endfunction
-->t=%pi*(-10:10)/10;
-->plot3d(t,t,surf,theta=35,alpha=45,leg="X@Y@Z",flag=[-3,2,3]);
-->z=sin(x).*cos(y);
-->[x1,y1]=geom3d(x,y,z);
-->xpoly(x1,y1,"lines");
-->[x1,y1]=geom3d([0,0],[0,0],[5,0]);
-->xsegs(x1,y1);
-->xstring(x1(1),y1(1));

```



Przykład 4

Rozwiązanie liniowego układu równań $A*x+b=0$

```

-->A=rand(5,3)*rand(3,8);
-->b=A*ones(8,1);[x,kerA]=linsolve(A,b);A*x+b //compatible b
ans =
  1.0D-14 *
! - 0.1776357 !
! - 0.0444089 !
! - 0.1776357 !
! - 0.0888178 !

```

```

! - 0.3552714 !
-->b=ones(5,1);[x,kerA]=linsolve(A,b);A*x+b //uncompatible b
ans =
! 1. !
! 1. !
! 1. !
! 1. !
! 1. !
-->A=rand(5,5);[x,kerA]=linsolve(A,b), -inv(A)*b //x is uniquey
x =
! - 2.4905817 !
! 2.0896496 !
! - 3.2911329 !
! 1.7180914 !
! - 0.3848339 !
ans =
! - 2.4905817 !
! 2.0896496 !
! - 3.2911329 !
! 1.7180914 !
! - 0.3848339 !

```

7. Wyniki numeryczne

W rozdziale tym przedstawione są wyniki symulacji dla stacjonarnego i niestacjonarnego przepływu ciepła bezsiatkową metodą Kansa w obszarach o regularnym lub nieregularnym brzegu. Metodę tą zaimplementowano w środowisku programistycznym Scilab. W pracy badano zagadnienie przepływu ciepła z warunkiem Dirichleta na całym brzegu rozważanego obszaru.

7.1 Zagadnienie stacjonarne na obszarze regularnym

Przy rozwiązywaniu numerycznym wykorzystaliśmy następującą funkcję

$$f(x, y) = 8x(x-1) + 8y(y-1) \quad (7.1)$$

oraz rozwiązanie na brzegu i rozwiązanie dokładne

$$u(x, y) = g(x, y) = 4xy(1-x)(1-y) \quad (7.2)$$

Całkowita liczba punktów $N = N_2 * N_2 + 4(mc+1)$

gdzie

N_2 – liczba punktów w jednym kierunku

mc – liczba punktów na jednym brzegu siatki

c, m – parametry RFB (ustalane przez użytkownika)

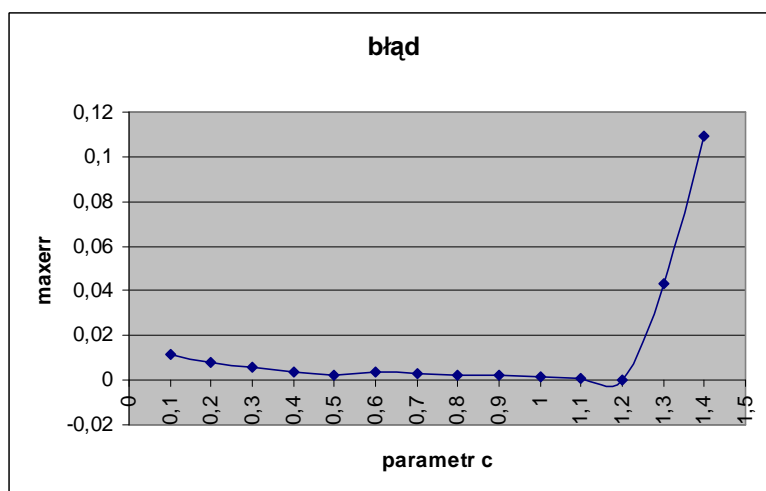
$normerr$ – największy błąd rozwiązania

$maxerr$ – różnica rozwiązania przybliżonego i rozwiązania dokładnego

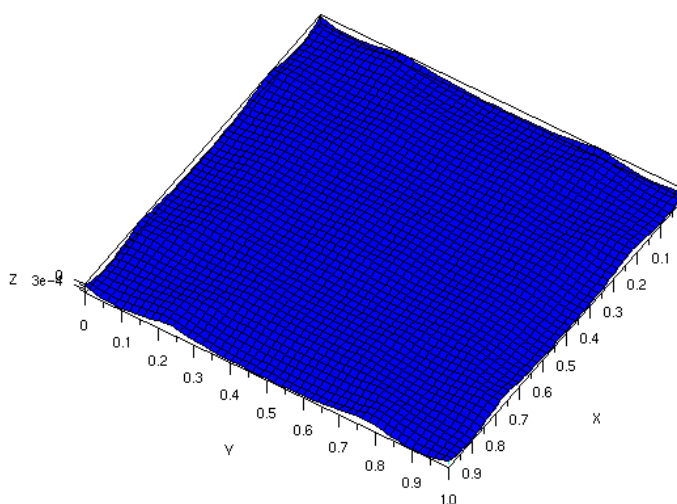
W tej części przedstawione są wyniki rozwiązania zagadnienie (5.7) z warunkami (5.8) z funkcjami (7.1) i (7.2) z wykorzystaniem funkcji wielomianowej opisanej równaniem (5.20). Tabela 7.1, 7.2, 7.3 i 7.4 przedstawiają zależność błędu rozwiązania od parametru c RFB oraz liczby punktów. Zagadnienie rozwiązano z całkowitą liczbą punktów: $N=64$ (wyniki Tabela 7.1), $N=92$ (wyniki Tabela 7.2), $N=144$ (wyniki Tabela 7.3), $N=196$ (wyniki Tabela 7.4). Rysunek 7.1, 7.3, 7.5, 7.7 przedstawia zależności błędu $maxerr$ od parametru c dla poszczególnych przypadków. Rysunek 7.2, 7.4, 7.6, 7.8 przedstawia rozkład błędu $maxerr$ na powierzchni siatki.

Tabela 7.1. Zależność błędu rozwiązania od parametru c RFB i liczby punktów dla $N_2=mc=6$

N_2	Mc	C	$normerr$	$maxerr$
6	6	0,1	1.954D-14	0.0113854
6	6	0,2	6.128D-14	0.0078524
6	6	0,3	2.003D-13	0.0056913
6	6	0,4	0.0009811	0.0033898
6	6	0,5	0.0006404	0.0022682
6	6	0,6	0.0033712	0.0033712
6	6	0,7	0.0028549	0.0028549
6	6	0,8	0.0024096	0.0024096
6	6	0,9	0.0020307	0.0020307
6	6	1	0.0399939	0.0013934
6	6	1,1	0.0282471	0.0007394
6	6	1,2	0.0200686	0.0004183
6	6	1,3	0.0630776	0.0434110
6	6	1,4	0.1722337	0.1090923



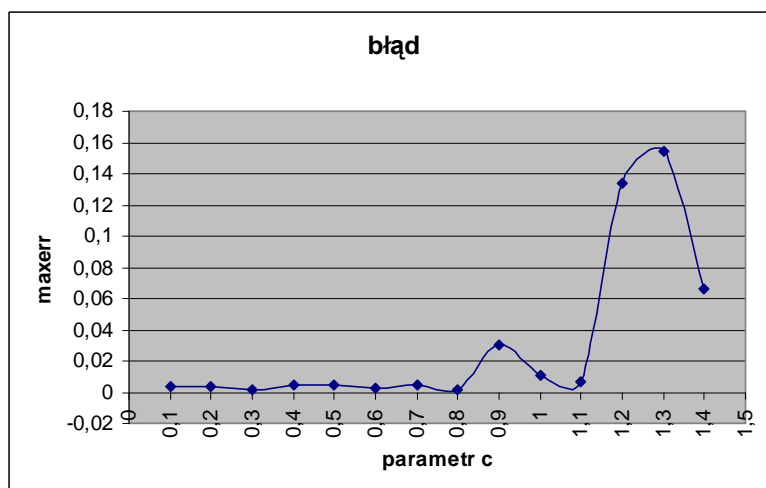
Rysunek 7.1. Wykres zależności błędu maxerr od parametru c



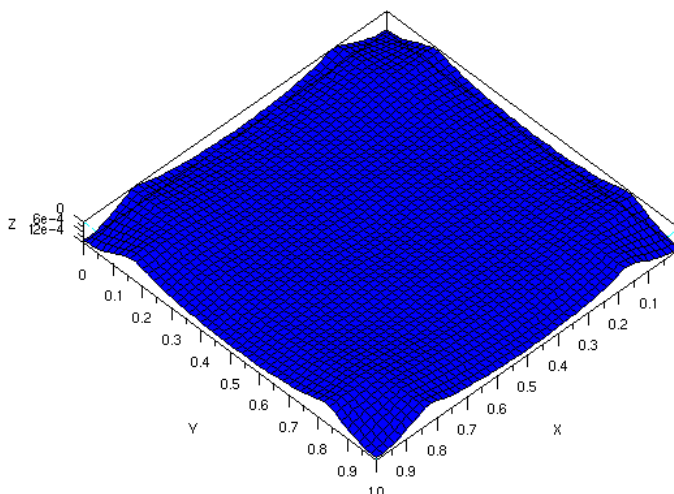
Rysunek 7.2. Rozkład błędów na powierzchni siatki dla c=1.1

Tabela 7.2. Zależność błędów rozwiązania od parametru c RBF i liczby punktów dla $N_2=8$ i $m_c=6$.

N2	Mc	c	normerr	maxerr
8	6	0,1	2.309D-14	0.0031653
8	6	0,2	1.013D-13	0.0036745
8	6	0,3	0.0008561	0.0017085
8	6	0,4	0.0044267	0.0045356
8	6	0,5	0.0042235	0.0044080
8	6	0,6	0.0109015	0.0030065
8	6	0,7	0.0402305	0.0050632
8	6	0,8	0.0278450	0.0010585
8	6	0,9	0.1071260	0.0298263
8	6	1	0.0752494	0.0108244
8	6	1,1	0.0560154	0.0071234
8	6	1,2	0.1342690	0.1342690
8	6	1,3	0.1544249	0.1544249
8	6	1,4	0.0835125	0.0660098



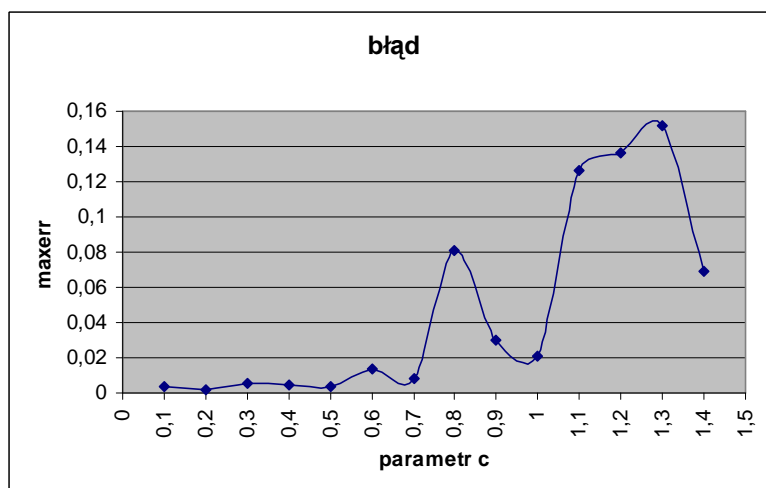
Rysunek 7.3. Wykres zależności błędu maxerr od parametru c



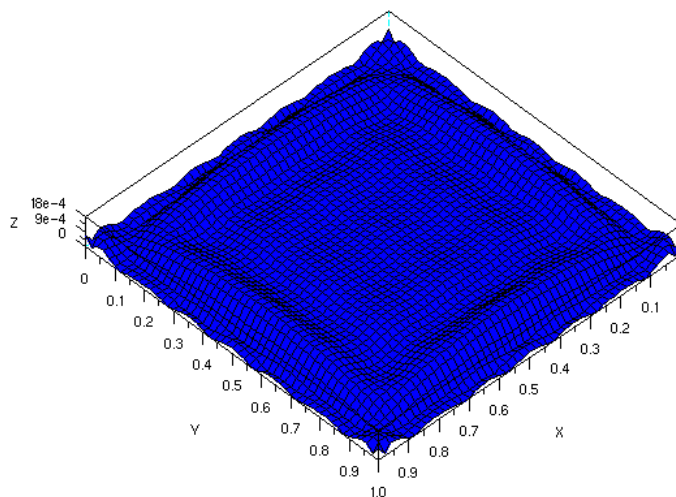
Rysunek 7.4. Rozkład błędów na powierzchni siatki dla c=0,8

Tabela 7.3. Zależność błędów rozwiązania od parametru c RBF i liczby punktów dla N2=8 i mc=6.

N2	Mc	c	normerr	maxerr
10	10	0,1	7.505D-14	0.0032010
10	10	0,2	0.0006849	0.0016444
10	10	0,3	0.0054937	0.0054937
10	10	0,4	0.0047114	0.0047114
10	10	0,5	0.0477500	0.0032516
10	10	0,6	0.1072362	0.0137734
10	10	0,7	0.0771653	0.0077284
10	10	0,8	0.1959317	0.0809233
10	10	0,9	0.1370268	0.0299398
10	10	1	0.1071163	0.0212022
10	10	1,1	0.1261928	0.1261928
10	10	1,2	0.1359728	0.1359728
10	10	1,3	0.1517363	0.1517363
10	10	1,4	0.0947993	0.0694404



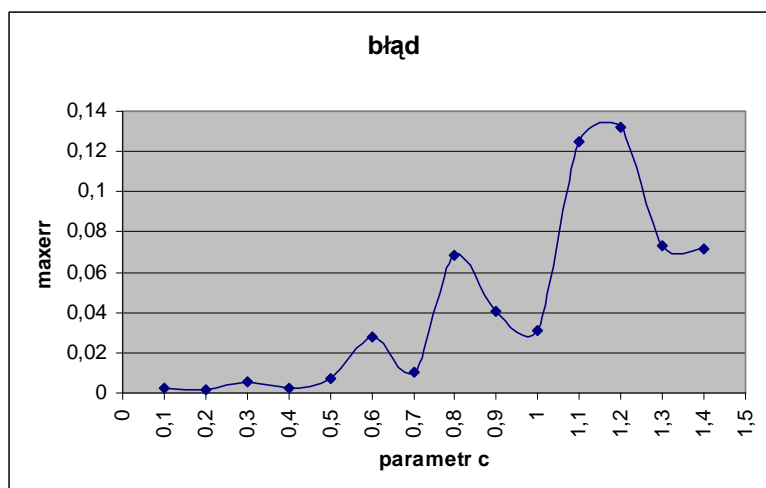
Rysunek 7.5. Wykres zależności błędu maxerr od parametru c



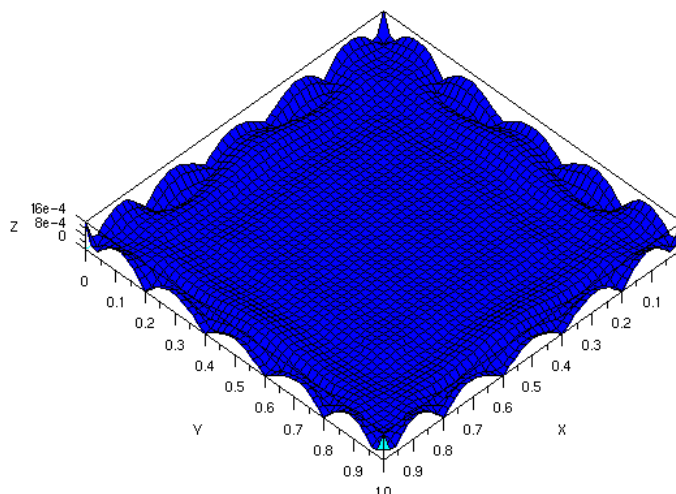
Rysunek 7.6. Rozkład błędów na powierzchni siatki dla c=0,2

Tabela 7.4. Zależność błędów rozwiązania od parametru c RBF i liczby punktów dla $N_2=mc=12$.

N_2	M_c	c	normerr	maxerr
12	12	0,1	0.0008587	0.0023637
12	12	0,2	0.0015625	0.0015625
12	12	0,3	0.0057057	0.0057057
12	12	0,4	0.0312496	0.0025819
12	12	0,5	0.0733099	0.0075379
12	12	0,6	0.1173891	0.0279423
12	12	0,7	0.1205092	0.0100331
12	12	0,8	0.1354752	0.0684905
12	12	0,9	0.1457487	0.0404504
12	12	1	0.1170806	0.0306827
12	12	1,1	0.1247097	0.1247097
12	12	1,2	0.1320063	0.1320063
12	12	1,3	0.1234728	0.0733341
12	12	1,4	0.1017681	0.0716369



Rysunek 7.7. Wykres zależności błędów maxerr od parametru c



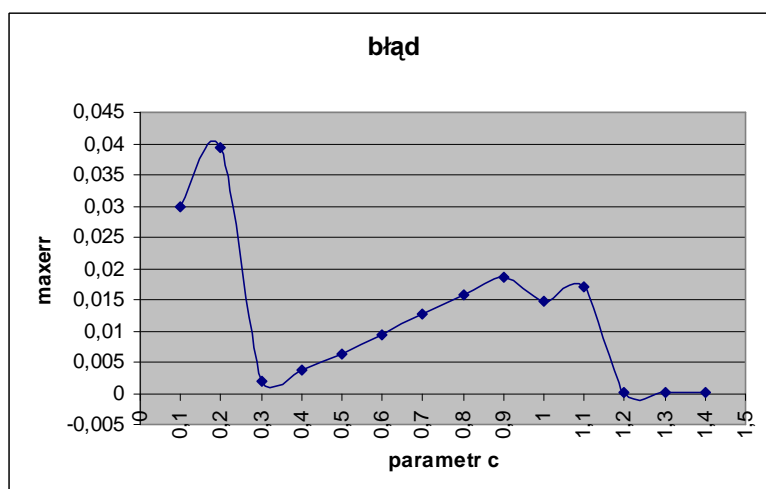
Rysunek 7.8. Rozkład błędów na powierzchni siatki dla c=0.2

W tej części przedstawione są wyniki rozwiązania zagadnienia (5.7) z warunkami (5.8) z funkcjami (7.1) i (7.2) z wykorzystaniem funkcji gaussowskiej opisanej równaniem (5.27). Tabela 7.5, 7.6, 7.7, 7.8 przedstawiają zależność błędów rozwiązania od parametru c RFB oraz liczby punktów. Zagadnienie rozwiązano z całkowitą liczbą punktów: N=64 (wyniki Tabela 7.5), N=92 (wyniki Tabela 7.6), N=144 (wyniki Tabela 7.7), N=196 (wyniki Tabela 7.8). Rysunek 7.9, 7.11, 7.13, 7.15 przedstawia zależności błędów maxerr od parametru c dla poszczególnych. Rysunek 7.10, 7.12, 7.14, 7.16 przedstawia rozkład błędów maxerr na powierzchni siatki.

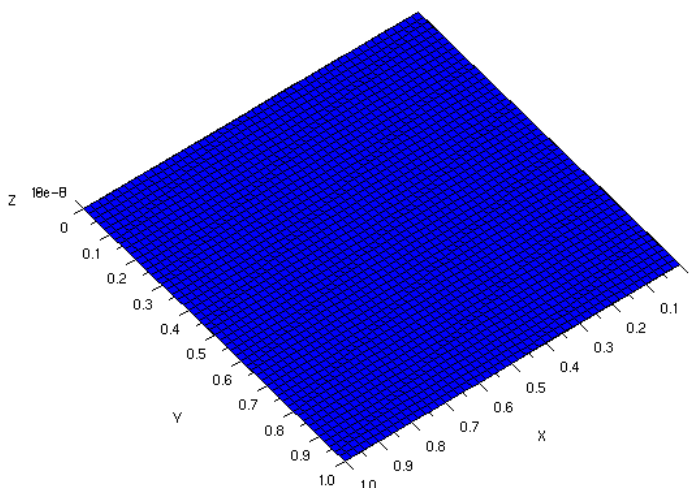
Tabela 7.5. Zależność błędów rozwiązania od parametru c RFB i liczby punktów dla N2=mc=6.

N2	Mc	Rysunek	normerr	maxerr
6	6	0,1	0.0624322	0.0299667
6	6	0,2	0.1097560	0.0394721
6	6	0,3	0.0073266	0.0017957
6	6	0,4	0.0124937	0.0037399
6	6	0,5	0.0186152	0.0063310
6	6	0,6	0.0255010	0.0093753
6	6	0,7	0.0330511	0.0126337
6	6	0,8	0.0412764	0.0158445
6	6	0,9	0.0503270	0.0187168

6	6	1	0.0460284	0.0147812
6	6	1,1	0.0556963	0.0171457
6	6	1,2	0.0033715	0.0001615
6	6	1,3	0.0041670	0.0001705
6	6	1,4	0.0050601	0.0001784



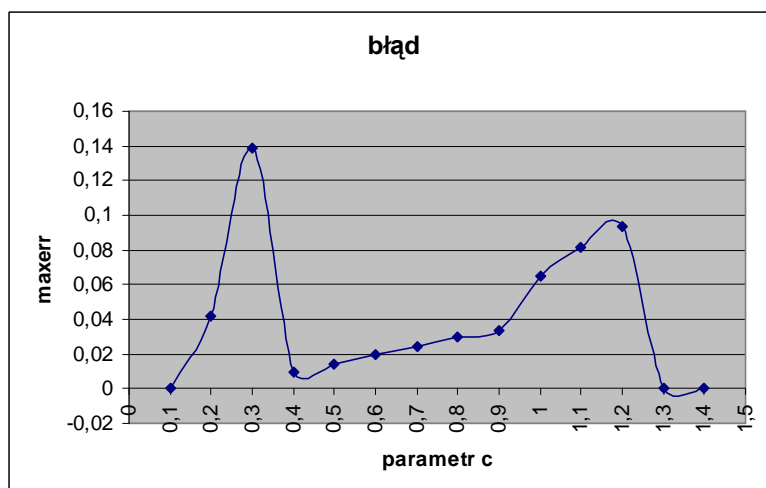
Rysunek 7.9. Wykres zależności błędów maxerr od parametru c



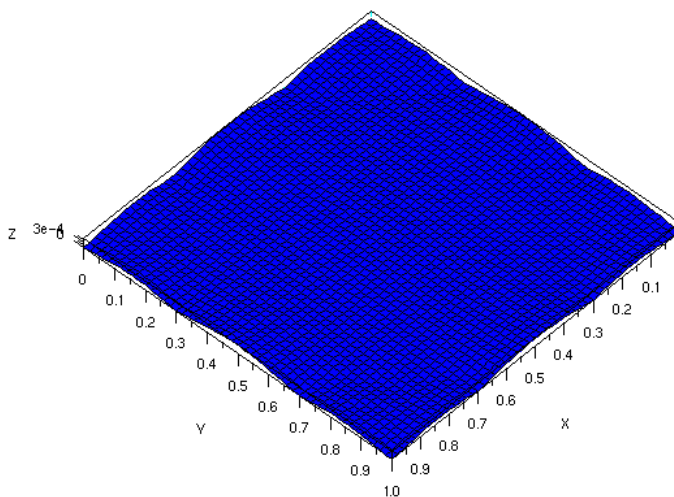
Rysunek 7.10. Rozkład błędów na powierzchni siatki dla c=1,2

Tabela 7.6. Zależność błędów rozwiązania od parametru c RBF i liczby punktów dla N2=8 i mc=6.

N2	Mc	c	normerr	maxerr
8	6	0,1	0.0885048	0.0331027
8	6	0,2	0.1554137	0.0419937
8	6	0,3	0.1390229	0.1390229
8	6	0,4	0.0171146	0.0095818
8	6	0,5	0.0240087	0.0145530
8	6	0,6	0.0312953	0.0197019
8	6	0,7	0.0389409	0.0247280
8	6	0,8	0.0470244	0.0294032
8	6	0,9	0.0557661	0.0335018
8	6	1	0.1299819	0.0649190
8	6	1,1	0.1727019	0.0811124
8	6	1,2	0.2202545	0.0931086
8	6	1,3	0.0096565	0.0004508
8	6	1,4	0.0117903	0.0005303
8	6	2,0	0.0301723	0.0007251



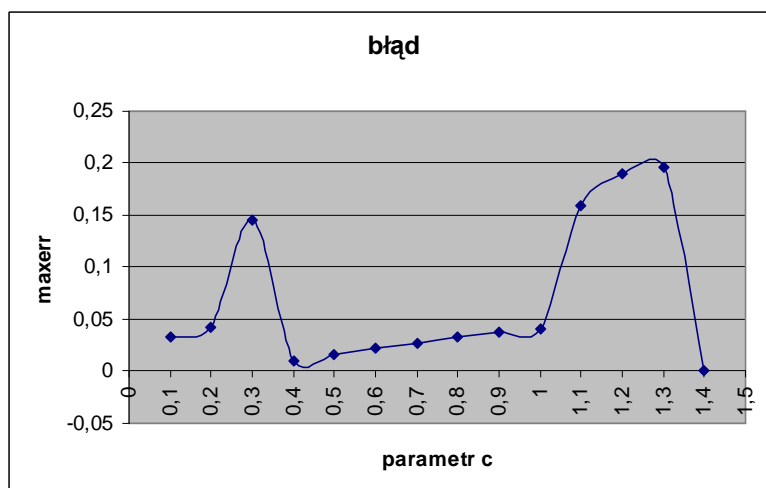
Rysunek 7.11. Wykres zależności błędu maxerr od parametru c



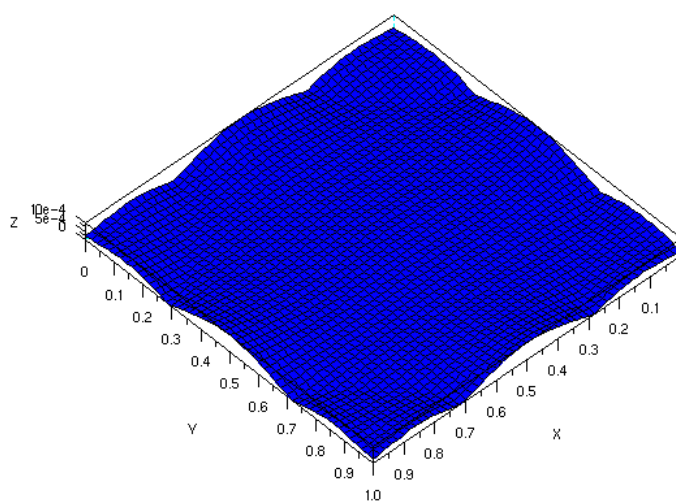
Rysunek 7.12. Rozkład błędów na powierzchni siatki dla $c=1.3$

Tabela 7.7. Zależność błędów rozwiązania od parametru c RBF i liczby punktów dla $N_2=mc=10$.

N2	Mc	c	normerr	maxerr
10	10	0,1	1.097D-01	0.0333188
10	10	0,2	0.1923644	0.0428996
10	10	0,3	0.1450778	0.1450778
10	10	0,4	0.0182454	0.0106498
10	10	0,5	0.0257229	0.0161441
10	10	0,6	0.0336331	0.0217797
10	10	0,7	0.0419321	0.0272646
10	10	0,8	0.0506814	0.0323865
10	10	0,9	0.0600859	0.0369325
10	10	1	0.0706080	0.0406094
10	10	1,1	0.3543684	0.1587807
10	10	1,2	0.4713807	0.1901072
10	10	1,3	0.5686181	0.1956861
10	10	1,4	0.0142684	0.0009720
10	10	1,5	0.0173137	0.0011406
10	10	1,6	0.0207284	0.0013036



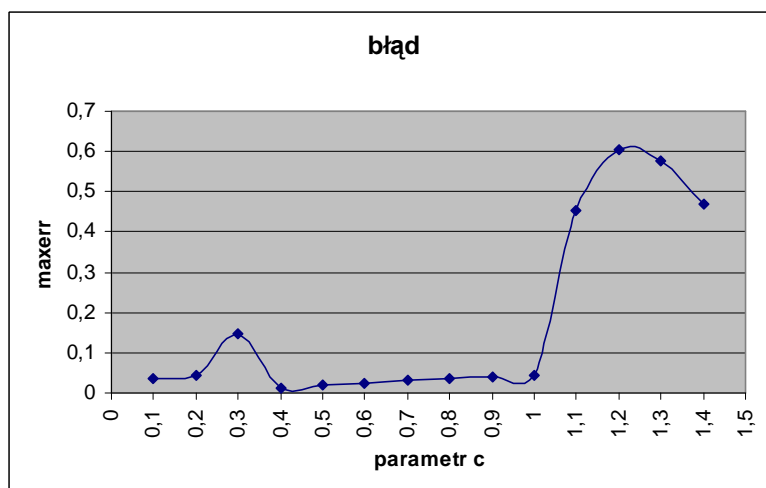
Rysunek 7.13. Wykres zależności błędu maxerr od parametru c



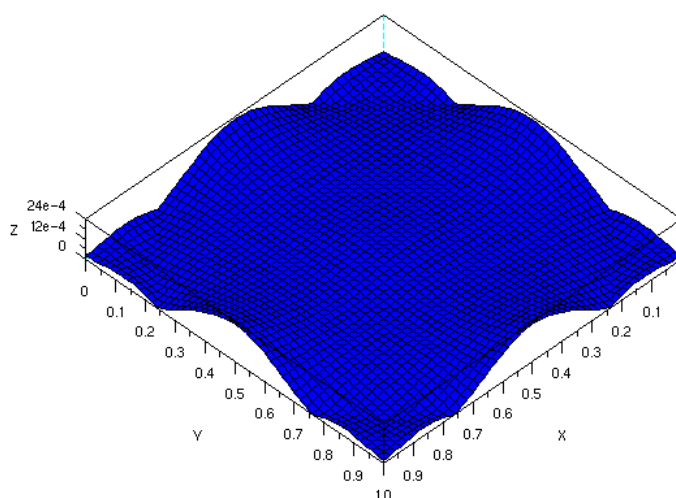
Rysunek 7.14. Rozkład błędu na powierzchni siatki dla c=1.5

Tabela 7.8. Zależność błędu rozwiązania od parametru c RBF i liczby punktów dla $N_2=mc=12$.

N2	Mc	c	normerr	maxerr
12	12	0,1	0.1239573	0.0348731
12	12	0,2	0.2171956	0.0444146
12	12	0,3	0.1455295	0.1455295
12	12	0,4	0.0185781	0.0130766
12	12	0,5	0.0258792	0.0191534
12	12	0,6	0.0335727	0.0251949
12	12	0,7	0.0416201	0.0309773
12	12	0,8	0.0500576	0.0363442
12	12	0,9	0.0590462	0.0411223
12	12	1	0.0689913	0.0450460
12	12	1,1	0.9015030	0.4537807
12	12	1,2	1.3307635	0.6036837
12	12	1,3	1.4894106	0.5776975
12	12	1,4	1.5230091	0.4693389
12	12	2.0	0.0470259	0.0022409



Rysunek 7.15. Wykres zależności błędu maxerr od parametru c



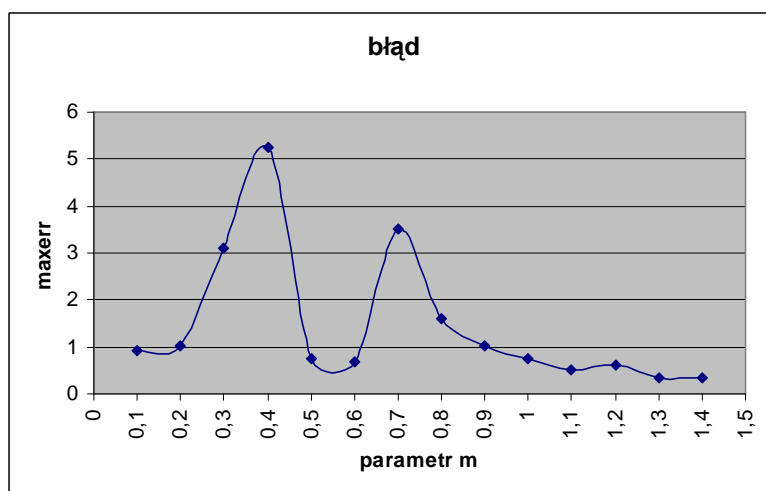
Rysunek 7.16. Rozkład błędu na powierzchni siatki dla c=2.0

W tej części przedstawione są wyniki rozwiązania zagadnienie (5.7) z warunkami (5.8) z funkcjami (7.1) i (7.2) z wykorzystaniem funkcji poliharmoniczne opisanej równaniem (5.34). Tabela 7.9, 7.10, 7.11, 7.12 przedstawiają zależność błędu rozwiązania od parametru c RFB oraz liczby punktów. Zagadnienie rozwiązano z całkowitą liczbą punktów: N=64 (wyniki Tabela 7.9), N=92 (wyniki Tabela 7.10), N=144 (wyniki Tabela 7.11), N=196 (wyniki Tabela 7.12). Rysunek 7.17, 7.19, 7.21, 7.23 przedstawia zależności błędu maxerr od parametru c dla poszczególnych. Rysunek 7.18, 7.20, 7.22, 7.24 przedstawia rozkład błędu maxerr na powierzchni siatki.

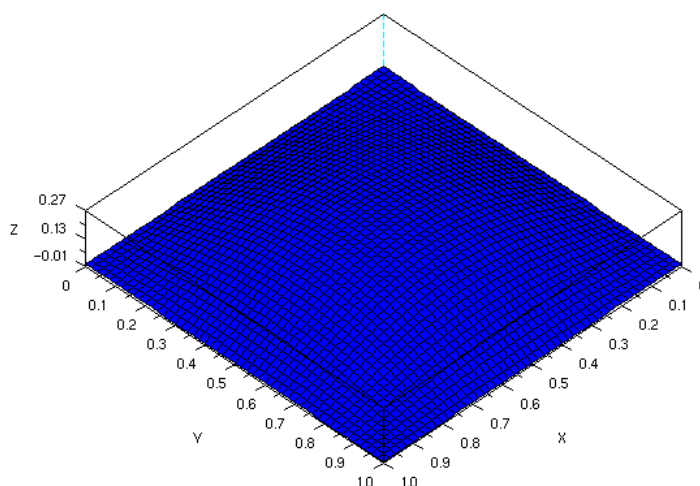
Tabela 7.9. Zależność błędu rozwiązania od parametru m RBF i liczby punktów dla N2=mc=6.

N2	Mc	m	normerr	maxerr
6	6	0,1	3.508D-14	0.9130592
6	6	0,2	2.442D-14	1.0056973
6	6	0,3	2.842D-14	3.106756
6	6	0,4	3.073D-13	5.2541275
6	6	0,5	2.576D-14	0.7425793
6	6	0,6	6.617D-14	0.6961774
6	6	0,7	5.809D-13	3.4970123
6	6	0,8	2.802D-13	1.5891566
6	6	0,9	6.617D-14	1.0292578

6	6	1	6.306D-14	0.7407603
6	6	1,1	2.753D-14	0.5174409
6	6	1,2	9.059D-14	0.6244887
6	6	1,3	7.150D-14	0.3516181
6	6	1,4	3.686D-14	0.3247153
6	6	1.8	0.0006767	0.2479517



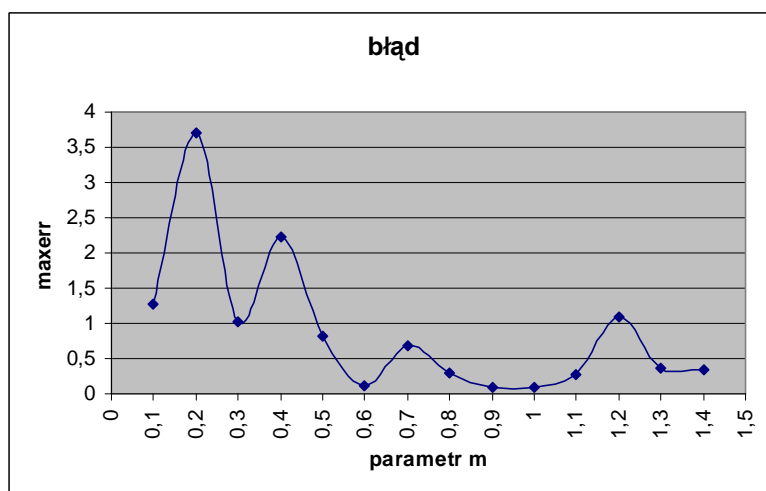
Rysunek 7.17. Wykres zależności błędu maxerr od parametru m



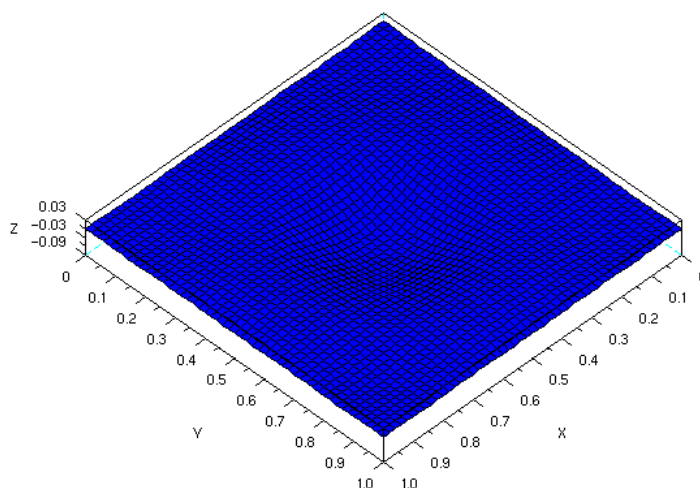
Rysunek 7.18. Rozkład błędu na powierzchni siatki dla $c=1.8$

Tabela 7.10. Zależność błędu rozwiązania od parametru c RBF i liczby punktów dla $N_2=8$ i $m_c=6$.

N_2	M_c	M	normerr	maxerr
8	6	0,1	6.595D-14	1.275659
8	6	0,2	7.905D-14	3.6991527
8	6	0,3	1.599D-14	1.0212842
8	6	0,4	1.359D-13	2.2323971
8	6	0,5	1.785D-13	0.8157260
8	6	0,6	9.059D-14	0.1106109
8	6	0,7	4.210D-13	0.6877060
8	6	0,8	5.080D-13	0.3040188
8	6	0,9	6.972D-14	0.0989413
8	6	1	5.596D-14	0.0805305
8	6	1,1	3.268D-13	0.2654541
8	6	1,2	1.439D-13	1.0938176
8	6	1,3	6.040D-14	0.3671871
8	6	1,4	6.661D-14	0.3357572
8	6	2.0	0.0004727	0.2867172



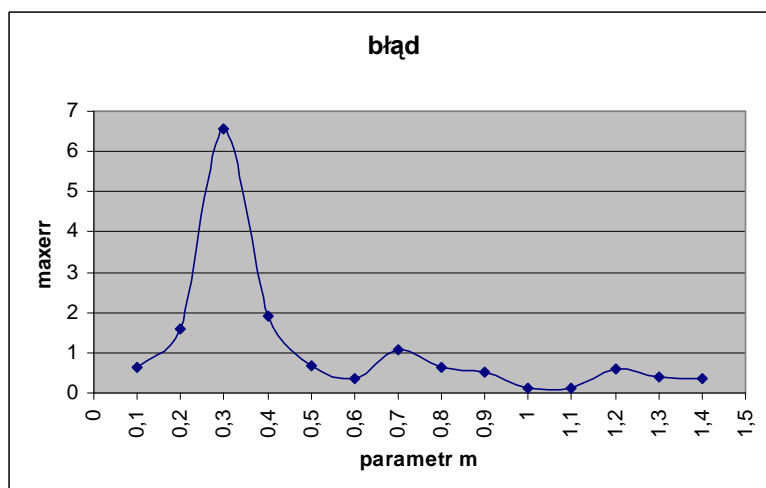
Rysunek 7.19. Wykres zależności błędu maxerr od parametru m



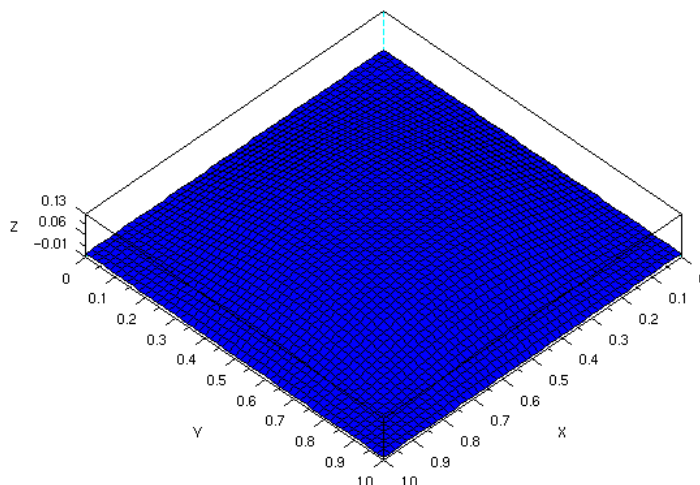
Rysunek 7.20. Rozkład błędów na powierzchni siatki dla $c=1.0$

Tabela 7.11. Zależność błędów rozwiązania od parametru c RBF i liczby punktów dla $N_2=mc=10$.

N2	Mc	m	normerr	maxerr
10	10	0,1	2.998D-13	0.6375128
10	10	0,2	1.372D-13	1.5919246
10	10	0,3	1.132D-12	6.5708422
10	10	0,4	6.062D-14	1.9074994
10	10	0,5	1.958D-13	0.6758597
10	10	0,6	0.0001688	0.3511017
10	10	0,7	2.706D-12	1.0779286
10	10	0,8	1.298D-12	0.6315494
10	10	0,9	0.0033381	0.5102281
10	10	1	0.0027852	0.1219150
10	10	1,1	0.0010388	0.1116492
10	10	1,2	0.0067753	0.5972305
10	10	1,3	0.0010995	0.4100541
10	10	1,4	0.0006917	0.3421126
10	10	2,0	0.0009788	0.2916652



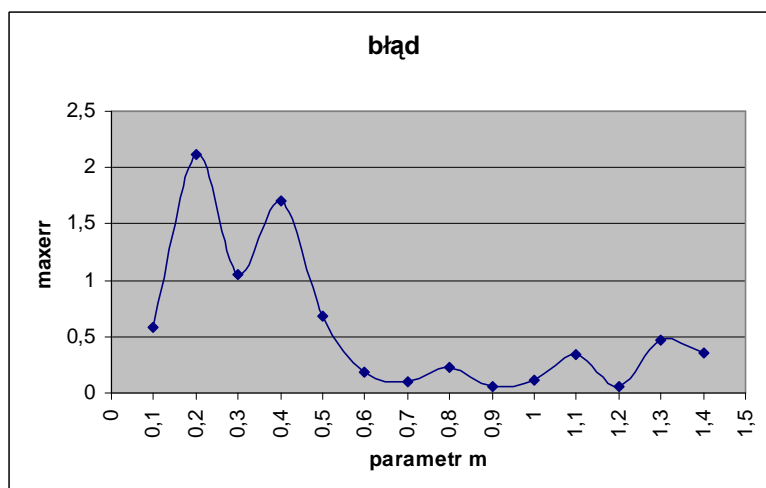
Rysunek 7.21. Wykres zależności błędów maxerr od parametru m



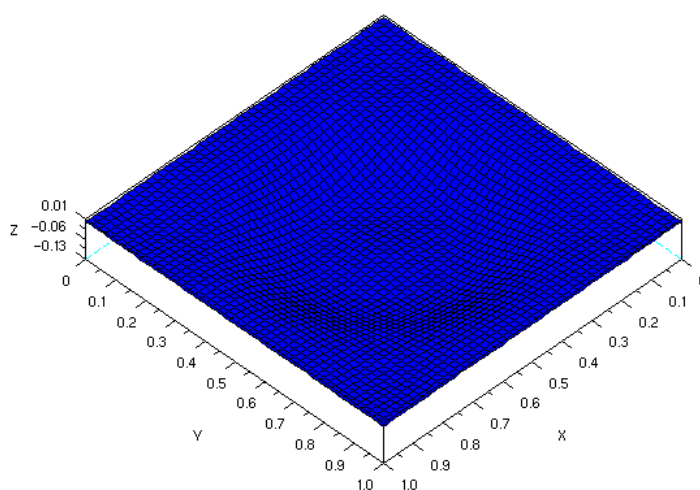
Rysunek 7.22. Rozkład błędów na powierzchni siatki dla $c=1.1$

Tabela 7.12. Zależność błędów rozwiązania od parametru c RBF i liczby punktów dla $N_2=mc=12$.

N_2	Mc	m	normerr	maxerr
12	12	0,1	9.592D-14	0.5845045
12	12	0,2	3.297D-13	2.1122169
12	12	0,3	2.596D-13	1.0465579
12	12	0,4	1.601D-13	1.7042643
12	12	0,5	6.157D-13	0.6794285
12	12	0,6	0.0023757	0.1815579
12	12	0,7	0.0087255	0.1039941
12	12	0,8	0.0011648	0.2239790
12	12	0,9	0.0063447	0.0502756
12	12	1	0.0010257	0.1177367
12	12	1,1	0.0007713	0.3348944
12	12	1,2	0.0036220	0.0560490
12	12	1,3	0.0027220	0.4644245
12	12	1,4	0.0005519	0.3525119
12	12	2.0	0.0013834	0.2953421



Rysunek 7.23. Wykres zależności błędu maxerr od parametru m



Rysunek 7.24. Rozkład błędności na powierzchni siatki dla c=0.9

7.2 Zagadnienie niestacjonarne na obszarze regularnym

Przy rozwiązywaniu numerycznym wykorzystaliśmy następującą funkcję zależną od czasu

$$f(x, y, t) = x(x-1) + y(y-1) - 4t \quad (7.3)$$

z warunkiem brzegowym

$$g(x, y, t) = x(x-1) + y(y-1) - 4t \quad (7.4)$$

oraz warunkiem początkowym

$$u(x, y, 0) = g(x, y, 0) \quad (7.5)$$

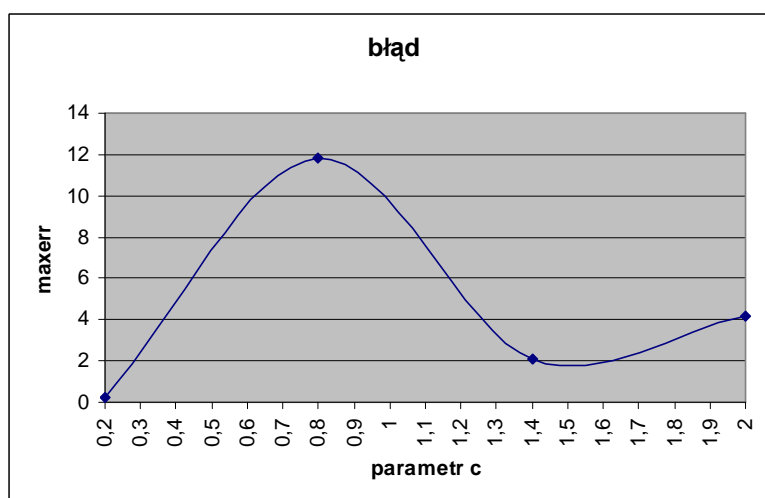
Całkowita liczba punktów $N = N_2 * N_2 + 4(mc + 1)$
gdzie

N_2 – liczba punktów w jednym kierunku
 mc – liczba punktów na jednym brzegu siatki
 c, m – parametry RFB (ustalane przez użytkownika)
 dt – krok czasowy
 $nstep$ – ilość kroków
 $maxerr$ – różnica rozwiązania przybliżonego i rozwiązania dokładnego

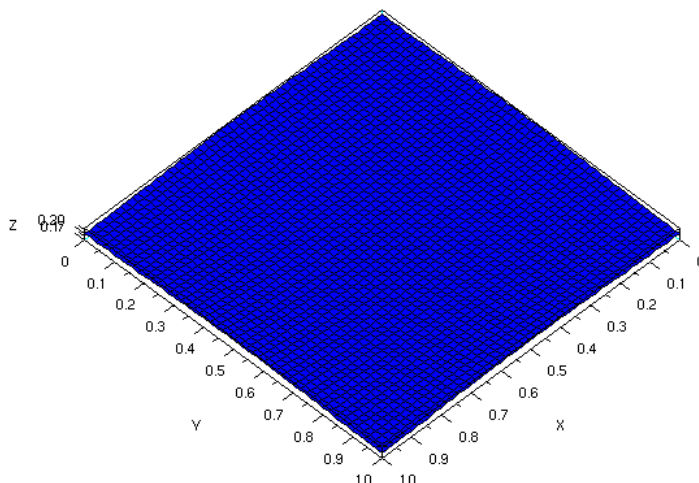
W tej części przedstawione są wyniki rozwiązania zagadnienie (5.12) z warunkami (5.13) i (5.15) z funkcjami (7.3), (7.4) i (7.5) z wykorzystaniem funkcji gaussowskiej opisanej równaniem (5.27). Tabela 7.13 do 7.20 przedstawiają zależność błędu rozwiązania od parametru c RFB, kroku czasowego, ilości kroków oraz ilości punktów. Zagadnienie rozwiązano z całkowitą liczbą punktów: $N=64$ (wyniki Tabela 7.13-7.20), różnią się tylko liczbą kroków oraz krokiem czasowym. Rysunek 7.24 – 7.38 przedstawia zależności błędu $maxerr$ od parametru c dla poszczególnych przypadków. Rysunek 7.25 - 7.39 przedstawia rozkład błędu $maxerr$ na powierzchni siatki. Na rysunkach przedstawiono tylko najlepsze rozwiązania z przedziału czasowego $t=(0,1)$.

Tabela 7.13. . Zależność błędu rozwiązania od parametru c , kroku czasowego $dt=0.01$, ilości kroków $nstep$ RFB i liczby punktów dla $N_2=mc=6$.

N_2	mc	c	dt	$nstep$	t	$maxerr$
6	6	0.2	0.01	25	0.25	0.3029780
6	6	0.8	0.01	25	0.25	1.2806833
6	6	1.4	0.01	25	0.25	0.3231402
6	6	2.0	0.01	25	0.25	1.8751755
6	6	0.2	0.01	50	0.5	0.3922071
6	6	0.8	0.01	50	0.5	8.9861341
6	6	1.4	0.01	50	0.5	0.3997528
6	6	2.0	0.01	50	0.5	10.586481
6	6	0.2	0.01	75	0.75	0.2294075
6	6	0.8	0.01	75	0.75	4.2869453
6	6	1.4	0.01	75	0.75	0.4945893
6	6	2.0	0.01	75	0.75	9.2617965
6	6	0.2	0.01	100	1	0.2132147
6	6	0.8	0.01	100	1	11.852697
6	6	1.4	0.01	100	1	2.0647655
6	6	2.0	0.01	100	1	4.1658733



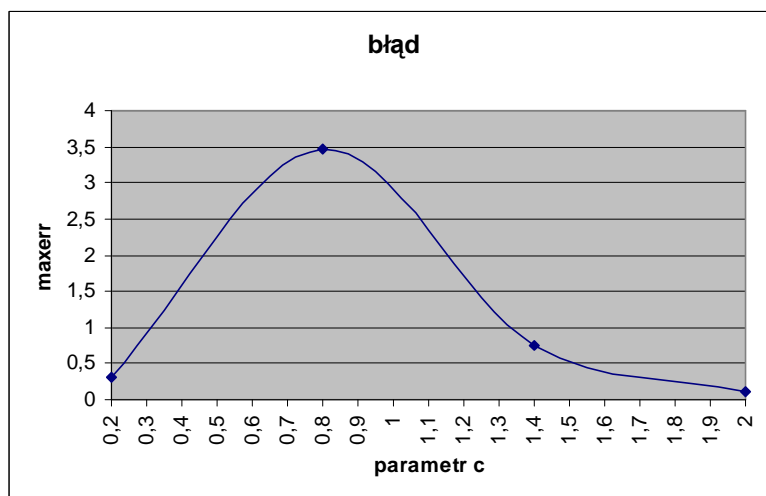
Rysunek 7.24. Wykres zależności błędności $maxerr$ od parametru c



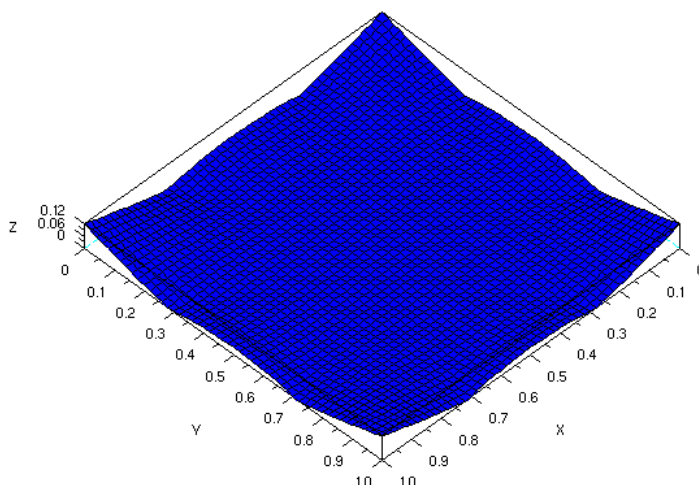
Rysunek 7.25. Rozkład błędu na powierzchni siatki dla $t=1$, $c=0.2$, $dt=0.01$ oraz $nstep=100$

Tabela 7.14. . Zależność błędu rozwiązania od parametru c , kroku czasowego $dt=0.02$, ilości kroków $nstep$ RBF i liczby punktów dla $N2=mc=6$.

$N2$	mc	c	dt	$nstep$	t	$maxerr$
6	6	0.2	0.02	12	0.24	0.2645802
6	6	0.8	0.02	12	0.24	4.1811468
6	6	1.4	0.02	12	0.24	0.1559103
6	6	2.0	0.02	12	0.24	0.0954451
6	6	0.2	0.02	25	0.5	0.3006686
6	6	0.8	0.02	25	0.5	3.4761856
6	6	1.4	0.02	25	0.5	0.7570325
6	6	2.0	0.02	25	0.5	0.1172551
6	6	0.2	0.02	37	0.74	0.1707809
6	6	0.8	0.02	37	0.74	3.1430269
6	6	1.4	0.02	37	0.74	54.666293
6	6	2.0	0.02	37	0.74	0.1221578
6	6	0.2	0.02	50	1	0.2880906
6	6	0.8	0.02	50	1	20.410524
6	6	1.4	0.02	50	1	1303.1228
6	6	2.0	0.02	50	1	1.5751447



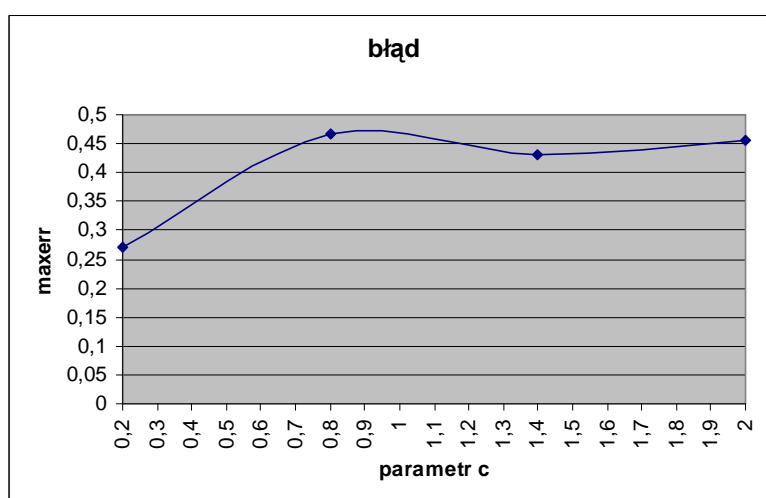
Rysunek 7.26. Wykres zależności błędu $maxerr$ od parametru c



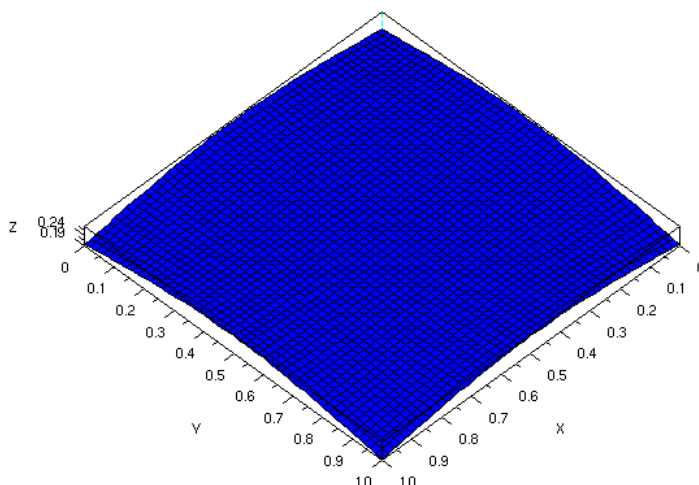
Rysunek 7.27. Rozkład błędu na powierzchni siatki dla $t=0.5$, $c=2.0$, $dt=0.02$ oraz $nstep=25$

Tabela 7.15. . Zależność błędu rozwiązania od parametru c , kroku czasowego $dt=0.1$, ilości kroków $nstep$ RBF i liczby punktów dla $N2=mc=6$.

N2	mc	c	dt	nstep	t	maxerr
6	6	0.2	0.1	2	0.2	0.3107348
6	6	0.8	0.1	2	0.2	0.4053606
6	6	1.4	0.1	2	0.2	0.4242126
6	6	2.0	0.1	2	0.2	0.4328425
6	6	0.2	0.1	5	0.5	0.2710862
6	6	0.8	0.1	5	0.5	0.4664788
6	6	1.4	0.1	5	0.5	0.4321893
6	6	2.0	0.1	5	0.5	0.4565939
6	6	0.2	0.1	7	0.7	0.4119239
6	6	0.8	0.1	7	0.7	0.4812909
6	6	1.4	0.1	7	0.7	0.4380746
6	6	2.0	0.1	7	0.7	0.4747372
6	6	0.2	0.1	10	1	0.4533962
6	6	0.8	0.1	10	1	0.5000563
6	6	1.4	0.1	10	1	0.4467244
6	6	2.0	0.1	10	1	0.4959059



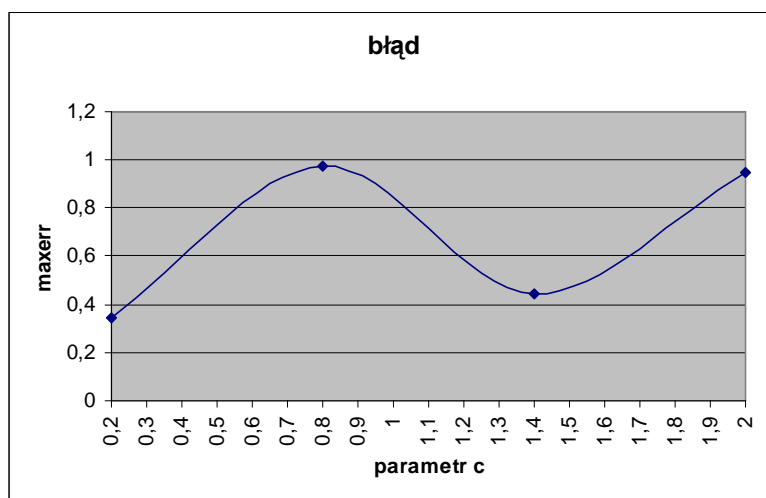
Rysunek 7.28. Wykres zależności błędu maxerr od parametru c



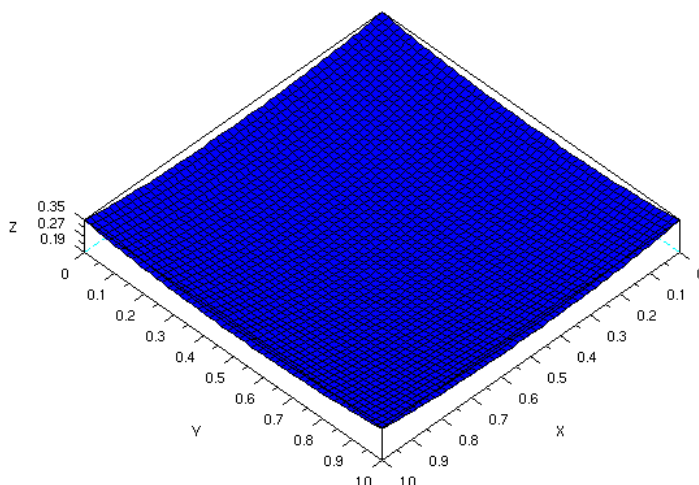
Rysunek 7.29. Rozkład błęd na powierzchni siatki dla $t=0.5$, $c=0.2$, $dt=0.1$ oraz $nstep=5$

Tabela 7.16. . Zależność błędu rozwiązania od parametru c , kroku czasowego $dt=0.01$, ilości kroków $nstep=10$ RBF i liczby punktów dla $N2=mc=10$.

N2	mc	c	dt	nstep	t	maxerr
10	10	0.2	0.01	25	0.25	0.3459502
10	10	0.8	0.01	25	0.25	0.9715474
10	10	1.4	0.01	25	0.25	0.4412888
10	10	2.0	0.01	25	0.25	0.9501994
10	10	0.2	0.01	50	0.5	0.5418663
10	10	0.8	0.01	50	0.5	3.1509075
10	10	1.4	0.01	50	0.5	1.2605725
10	10	2.0	0.01	50	0.5	5.539489
10	10	0.2	0.01	75	0.75	0.4452801
10	10	0.8	0.01	75	0.75	6.2791507
10	10	1.4	0.01	75	0.75	16.291362
10	10	2.0	0.01	75	0.75	9.4054181
10	10	0.2	0.01	100	1	2.5779367
10	10	0.8	0.01	100	1	11.14934
10	10	1.4	0.01	100	1	46.168739
10	10	2.0	0.01	100	1	8.7977085



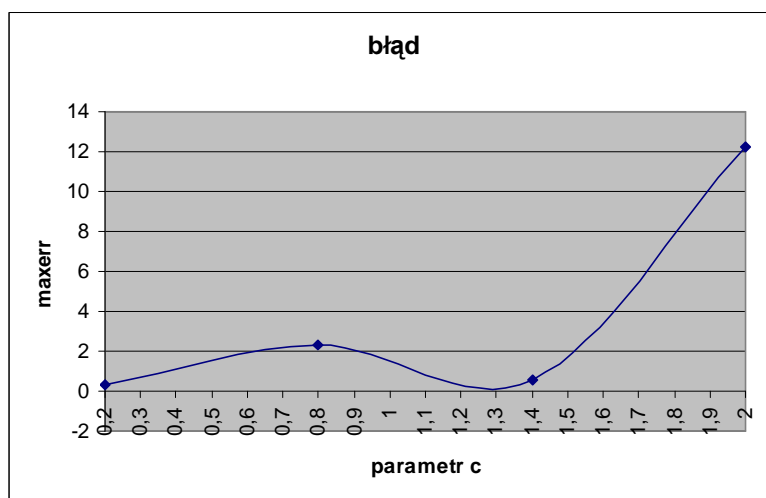
Rysunek 7.30. Wykres zależności błędu maxerr od parametru c



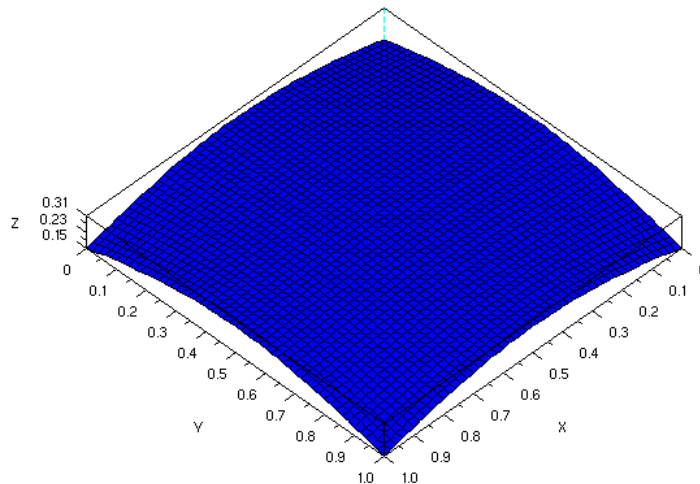
Rysunek 7.31. Rozkład błędu na powierzchni siatki dla $t=0.25$, $c=0.2$, $dt=0.01$ oraz $nstep=25$

Tabela 7.17. . Zależność błędu rozwiązania od parametru c , kroku czasowego $dt=0.02$, ilości kroków $nstep=10$ RBF i liczby punktów dla $N2=mc=10$.

N2	mc	c	dt	nstep	t	maxerr
10	10	0.2	0.02	12	0.24	0.3106977
10	10	0.8	0.02	12	0.24	0.4083652
10	10	1.4	0.02	12	0.24	0.3807367
10	10	2.0	0.02	12	0.24	38.738643
10	10	0.2	0.02	25	0.5	0.4379106
10	10	0.8	0.02	25	0.5	4.7365132
10	10	1.4	0.02	25	0.5	0.4844002
10	10	2.0	0.02	25	0.5	112.02064
10	10	0.2	0.02	37	0.74	0.3037182
10	10	0.8	0.02	37	0.74	2.29893
10	10	1.4	0.02	37	0.74	0.5883958
10	10	2.0	0.02	37	0.74	12.23086
10	10	0.2	0.02	50	1	1.3447407
10	10	0.8	0.02	50	1	6.8995245
10	10	1.4	0.02	50	1	0.7001236
10	10	2.0	0.02	50	1	76.43203



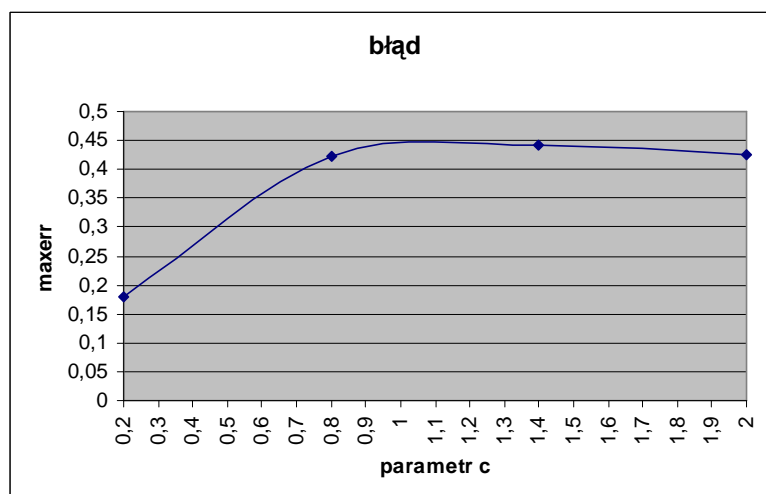
Rysunek 7.32. Wykres zależności błędu maxerr od parametru c



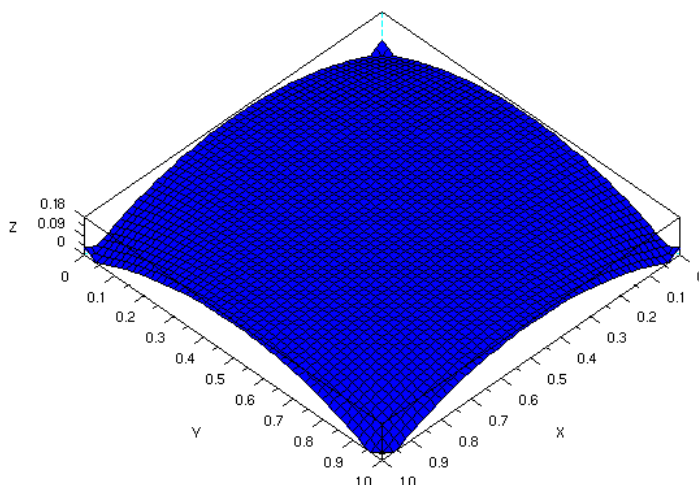
Rysunek 7.33. Rozkład błędu na powierzchni siatki dla $t=0.74$, $c=0.2$, $dt=0.02$ oraz $nstep=37$

Tabela 7.18. . Zależność błędu rozwiązania od parametru c , kroku czasowego $dt=0.1$, ilości kroków $nstep=10$ RBF i liczby punktów dla $N2=mc=10$.

N2	mc	c	dt	nstep	t	Maxerr
10	10	0.2	0.1	2	0.2	0.1789611
10	10	0.8	0.1	2	0.2	0.4224860
10	10	1.4	0.1	2	0.2	0.4423895
10	10	2.0	0.1	2	0.2	0.4243520
10	10	0.2	0.1	5	0.5	1.2683184
10	10	0.8	0.1	5	0.5	0.4668062
10	10	1.4	0.1	5	0.5	0.4491393
10	10	2.0	0.1	5	0.5	0.4762319
10	10	0.2	0.1	7	0.7	2.4197583
10	10	0.8	0.1	7	0.7	0.4782001
10	10	1.4	0.1	7	0.7	0.5108651
10	10	2.0	0.1	7	0.7	0.4945273
10	10	0.2	0.1	10	1	0.3203393
10	10	0.8	0.1	10	1	0.5067506
10	10	1.4	0.1	10	1	0.6559977
10	10	2.0	0.1	10	1	0.5210085



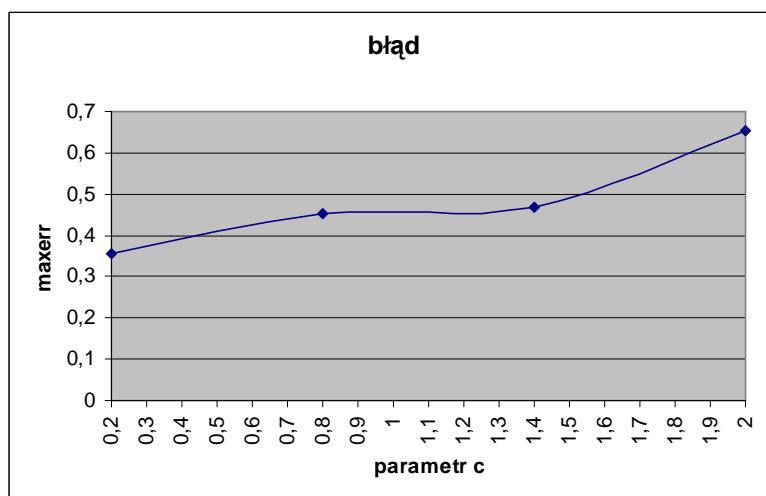
Rysunek 7.34. Wykres zależności błędu maxerr od parametru c



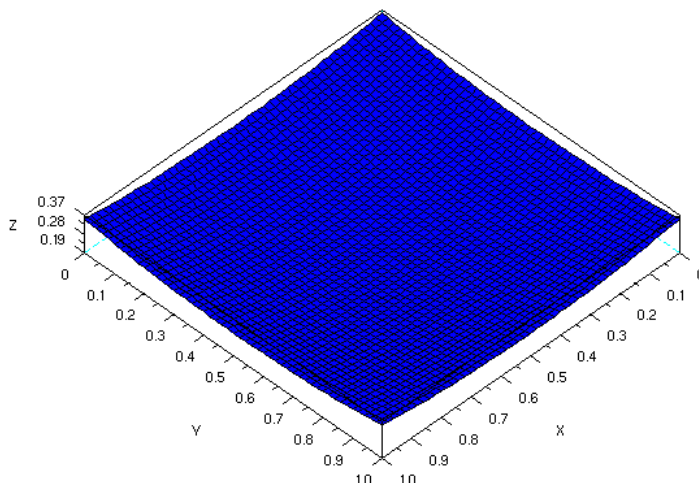
Rysunek 7.33. Rozkład błęd na powierzchni siatki dla $t=0.2$, $c=0.2$, $dt=0.1$ oraz $nstep=2$

Tabela 7.19. . Zależność błędu rozwiązania od parametru c , kroku czasowego $dt=0.01$, ilości kroków $nstep=10$ RBF i liczby punktów dla $N2=mc=12$.

N2	mc	c	dt	nstep	t	maxerr
12	12	0.2	0.01	25	0.25	0.3560341
12	12	0.8	0.01	25	0.25	0.4537643
12	12	1.4	0.01	25	0.25	0.4681896
12	12	2.0	0.01	25	0.25	0.6518993
12	12	0.2	0.01	50	0.5	0.5825366
12	12	0.8	0.01	50	0.5	0.5451727
12	12	1.4	0.01	50	0.5	1.7751378
12	12	2.0	0.01	50	0.5	4.013431
12	12	0.2	0.01	75	0.75	1.0900692
12	12	0.8	0.01	75	0.75	0.6383298
12	12	1.4	0.01	75	0.75	5.6114022
12	12	2.0	0.01	75	0.75	0.814938
12	12	0.2	0.01	100	1	2.9138923
12	12	0.8	0.01	100	1	0.7415501
12	12	1.4	0.01	100	1	8.9186998
12	12	2.0	0.01	100	1	9.8362355



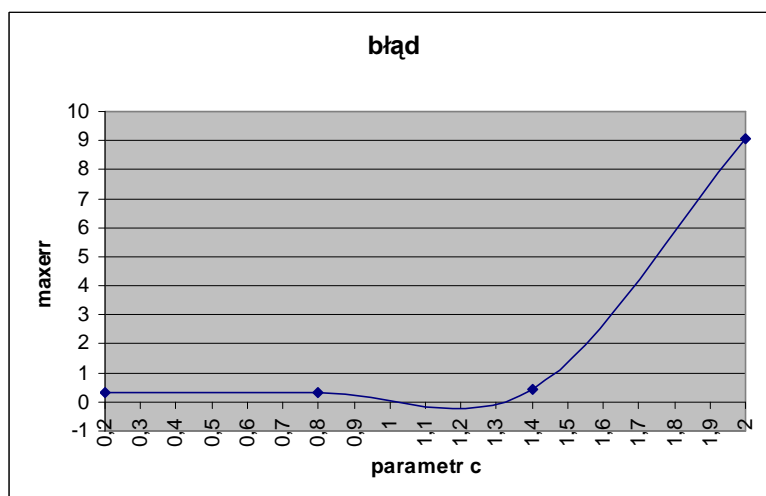
Rysunek 7.36. Wykres zależności błędu maxerr od parametru c



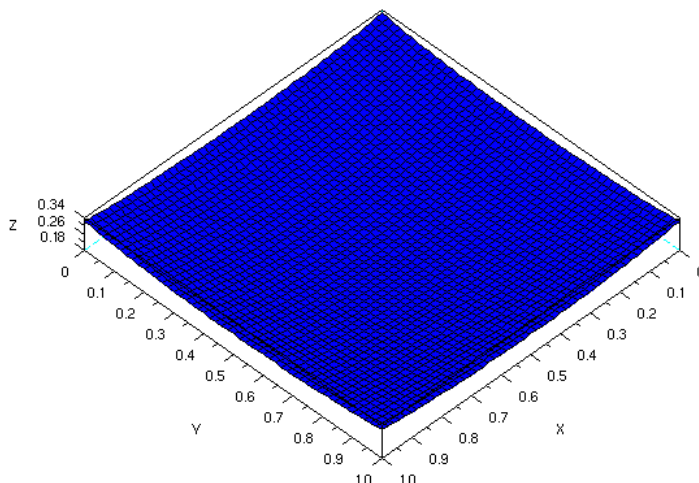
Rysunek 7.37. Rozkład błędu na powierzchni siatki dla $t=0.25$, $c=0.2$, $dt=0.01$ oraz $nstep=25$

Tabela 7.20. Zależność błędu rozwiązania od parametru c , kroku czasowego $dt=0.02$, ilości kroków $nstep=10$ RBF i liczby punktów dla $N2=mc=12$.

N2	mc	c	dt	nstep	t	maxerr
12	12	0.2	0.02	12	0.24	0.3223064
12	12	0.8	0.02	12	0.24	0.333493
12	12	1.4	0.02	12	0.24	0.4054508
12	12	2.0	0.02	12	0.24	9.049469
12	12	0.2	0.02	25	0.5	0.4802625
12	12	0.8	0.02	25	0.5	2.0495544
12	12	1.4	0.02	25	0.5	0.5271618
12	12	2.0	0.02	25	0.5	95.669205
12	12	0.2	0.02	37	0.74	0.8913009
12	12	0.8	0.02	37	0.74	4.6677115
12	12	1.4	0.02	37	0.74	0.6416059
12	12	2.0	0.02	37	0.74	62.856737
12	12	0.2	0.02	50	1	3.2254802
12	12	0.8	0.02	50	1	9.0623952
12	12	1.4	0.02	50	1	1.7531784
12	12	2.0	0.02	50	1	72.389193



Rysunek 7.38. Wykres zależności błędu maxerr od parametru c



Rysunek 7.39. Rozkład błędu na powierzchni siatki dla $t=0.24$, $c=0.2$, $dt=0.02$ oraz $nstep=12$

7.3 Zagadnienie stacjonarne na obszarze nieregularnym

Przy rozwiązywaniu numerycznym wykorzystaliśmy następującą funkcję

$$f(x, y) = 8x(x-1) + 8y(y-1) \quad (7.6)$$

oraz rozwiązanie na brzegu i rozwiązanie dokładne

$$u(x, y) = g(x, y) = 4xy(1-x)(1-y) \quad (7.7)$$

Całkowita liczba punktów $N = ((N2 * N2) + 2) + ((4 * mc) - 17)$

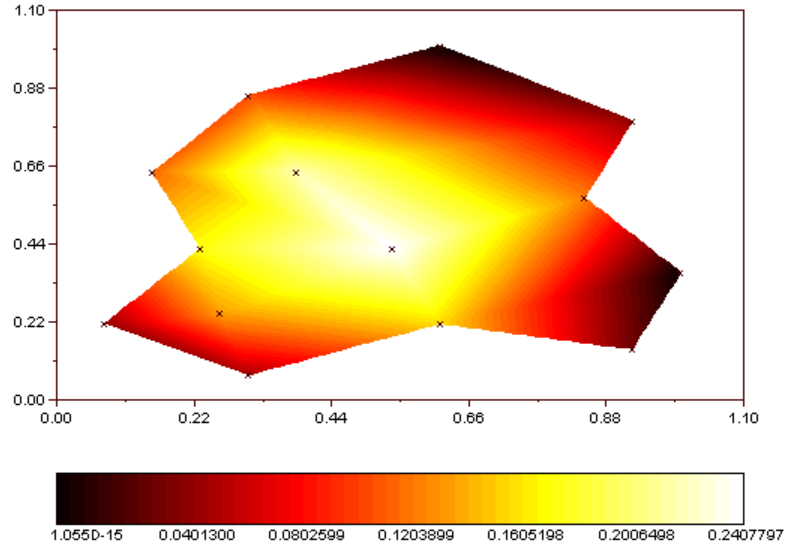
gdzie

$N2$ – liczba punktów w jednym kierunku

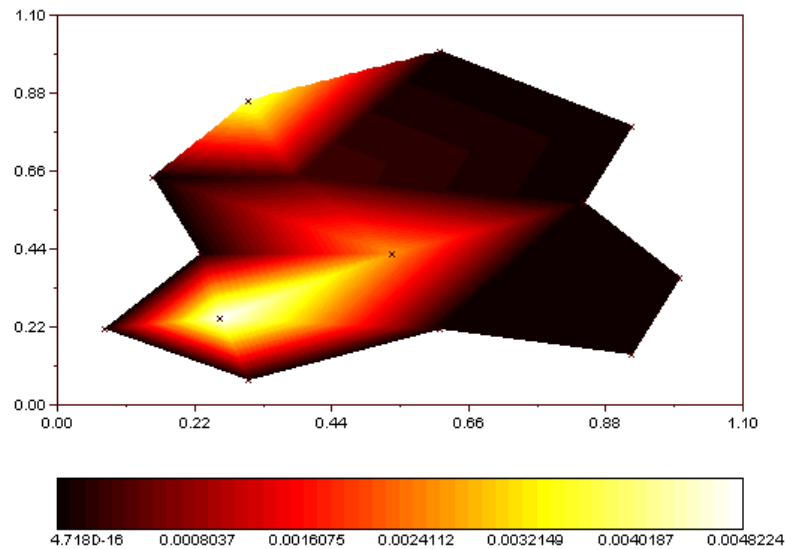
mc – liczba punktów na jednym brzegu siatki

c, m – parametry RFB (ustalane przez użytkownika)

W tej części przedstawione są wyniki rozwiązania zagadnienie (5.7) z warunkami (5.8) z funkcjami (7.6) i (7.7) z wykorzystaniem funkcji wielomianowej opisanej równaniem (5.20). Przedstawiono tutaj rozwiązanie dla najlepszego parametru $c=0.1$. Zagadnienie rozwiązano z całkowitą liczbą punktów: $N=38$. Rysunek 7.40 przedstawia rozwiązanie numeryczne na powierzchni obszaru nieregularnego. Rysunek 7.41 przedstawia rozkład błędu na powierzchni obszaru nieregularnego. Kolor biały na rysunku przedstawia rozwiązanie lub błąd maksymalny, natomiast kolor czarny oznacza rozwiązanie lub błąd minimalny.

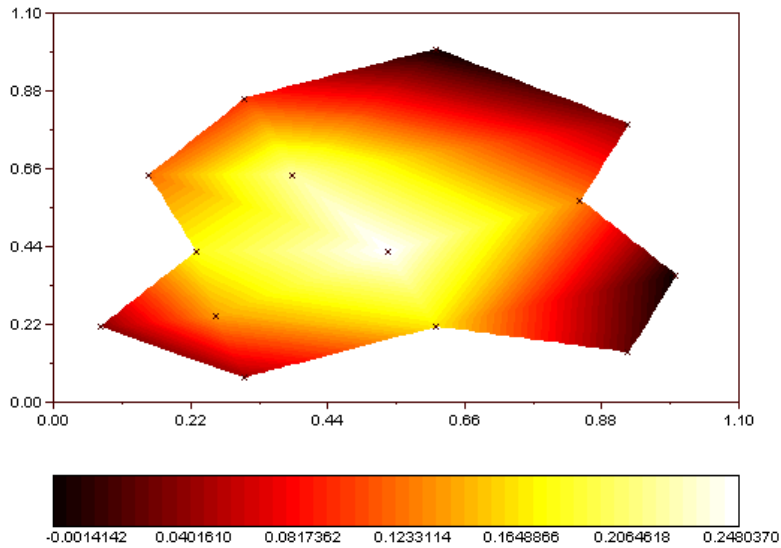


Rysunek 7.40 Rozwiązanie numeryczne na obszarze nieregularnym

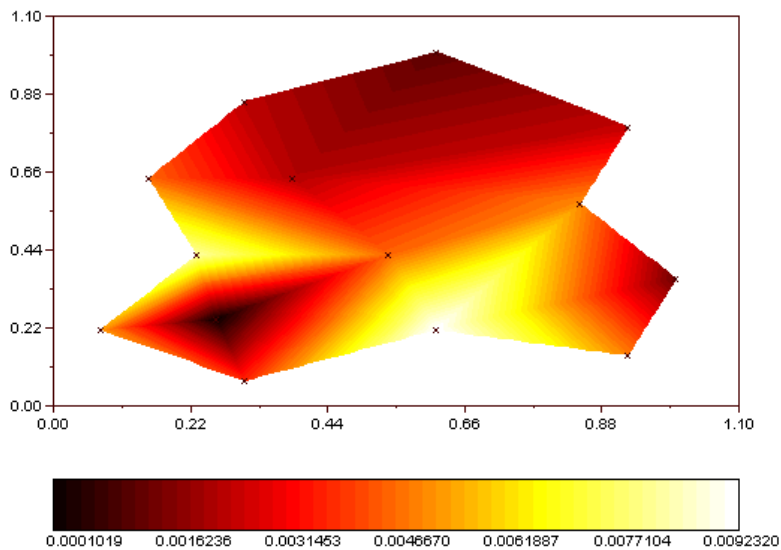


Rysunek 7.41 Rozkład błędów na powierzchni obszaru nieregularnego

W tej części przedstawione są wyniki rozwiązania zagadnienia (5.7) z warunkami (5.8) z funkcjami (7.6) i (7.7) z wykorzystaniem funkcji gaussowskiej opisanej równaniem (5.27). Przedstawiono tutaj rozwiązanie dla najlepszego parametru $c=3.0$. Zagadnienie rozwiązano z całkowitą liczbą punktów: $N=38$. Rysunek 7.42 przedstawia rozwiązanie numeryczne na powierzchni obszaru nieregularnego. Rysunek 7.43 przedstawia rozkład błędów na powierzchni obszaru nieregularnego.

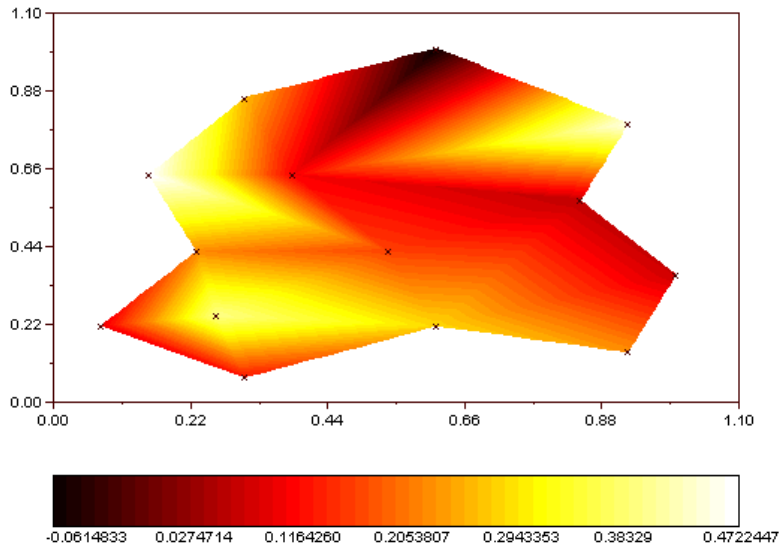


Rysunek 7.42 Rozwiązanie numeryczne na obszarze nieregularnym

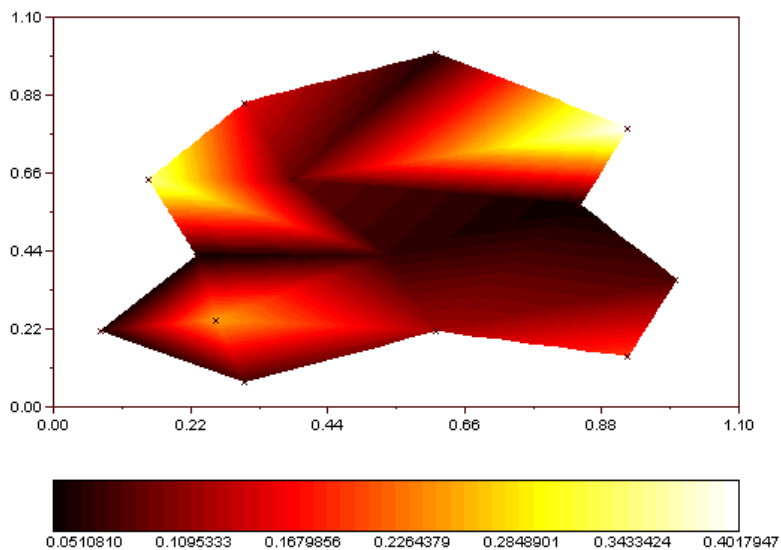


Rysunek 7.43 Rozkład błędów na powierzchni obszaru nieregularnego

W tej części przedstawione są wyniki rozwiązania zagadnienia (5.7) z warunkami (5.8) z funkcjami (7.6) i (7.7) z wykorzystaniem funkcji poliharmoniczne opisanej równaniem (5.34). Przedstawiono tutaj rozwiązanie dla najlepszego parametru $m=0.1$. Zagadnienie rozwiązano z całkowitą liczbą punktów: $N=38$. Rysunek 7.44 przedstawia rozwiązanie numeryczne na powierzchni obszaru nieregularnego. Rysunek 7.45 przedstawia rozkład błędów na powierzchni obszaru nieregularnego.



Rysunek 7.44 Rozwiązanie numeryczne na obszarze nieregularnym



Rysunek 7.45 Rozkład błędów na powierzchni obszaru nieregularnego

7.4 Zagadnienie niestacjonarne na obszarze nieregularnym

Przy rozwiązywaniu numerycznym wykorzystaliśmy następującą funkcję zależną od czasu

$$f(x, y, t) = x(x-1) + y(y-1) - 4t \quad (7.8)$$

z warunkiem brzegowym

$$g(x, y, t) = x(x-1) + y(y-1) - 4t \quad (7.9)$$

oraz warunkiem początkowym

$$u(x, y, 0) = g(x, y, 0) \quad (7.10)$$

Całkowita liczba punktów $N = ((N2 * N2) + 2) + ((4 * mc) - 17)$

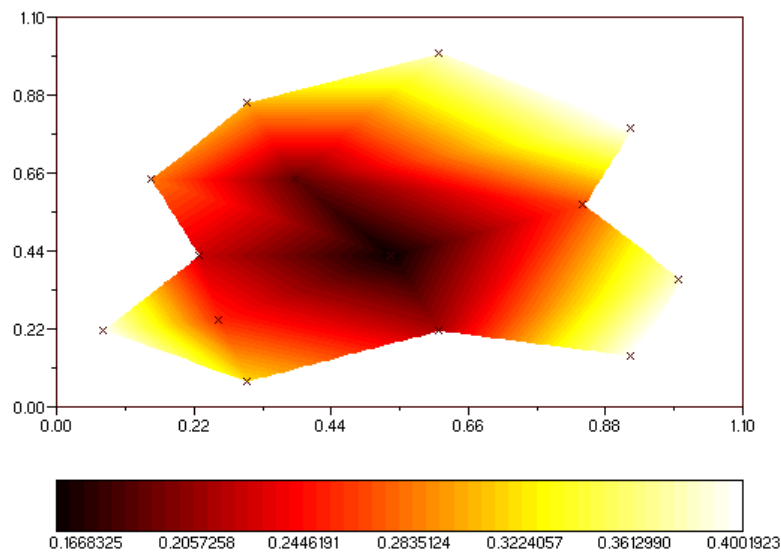
gdzie

$N2$ – liczba punktów w jednym kierunku

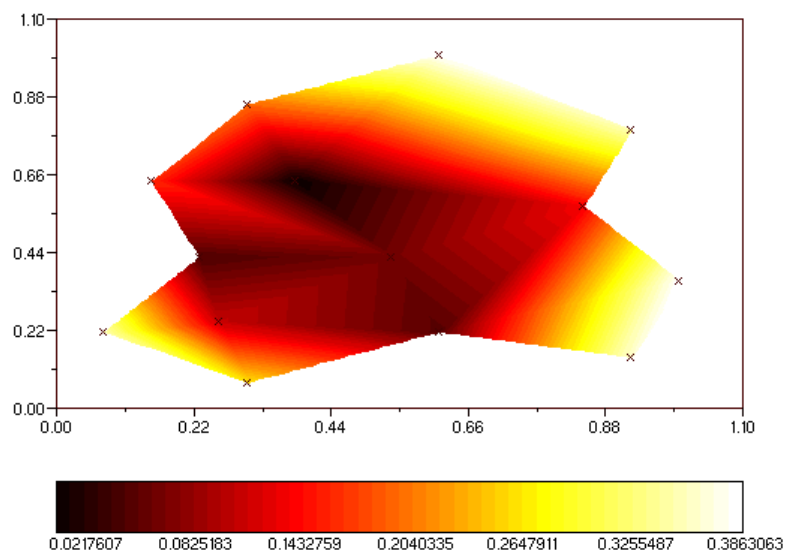
mc – liczba punktów na jednym brzegu siatki

c – parametr RFB (ustalany przez użytkownika)

W tej części przedstawione są wyniki rozwiązania zagadnienie (5.12) z warunkami (5.13) i (5.15) z funkcjami (7.8), (7.9) i (7.10) z wykorzystaniem funkcji wielomianowej opisanej równaniem (5.20). Przedstawiono tutaj rozwiązanie dla parametru $c=1.5$, kroku czasowego $dt=0.01$ oraz liczby kroków $nstep=20$. Zagadnienie rozwiązano z całkowitą liczbą punktów: $N=38$. Rysunek 7.46 przedstawia rozwiązanie numeryczne na powierzchni obszaru nieregularnego. Rysunek 7.47 przedstawia rozkład błędów na powierzchni obszaru nieregularnego. Kolor biały na rysunku przedstawia rozwiązanie lub błąd maksymalny, natomiast kolor czarny oznacza rozwiązanie lub błąd minimalny.



Rysunek 7.46 Rozwiązanie numeryczne na obszarze nieregularnym



Rysunek 7.47 Rozkład błędów na powierzchni obszaru nieregularnego

8. Wnioski

Dla stacjonarnego przepływu wykorzystane zostały wszystkie funkcje RFB zaproponowane w rozdziale 5. Wszystkie dostarczyły satysfakcjonujących wyników. Porównanie błędów rozwiązań przedstawiono w tabelach i na wykresach w rozdziale 7. Dla funkcji RFB wielomianowej najlepsze wyniki zostały zaobserwowane dla siatki $N=64$. Wyniki dla tego przypadku przedstawiono graficznie w rozdziale 7.1 na rysunkach 7.1-7.3 oraz tabeli 7.1. Dla funkcji RFB gaussowskiej najlepsze wyniki zostały otrzymane dla siatki $N=64$. Wyniki dla tego przypadku przedstawiono graficznie w rozdziale 7.1 na rysunkach 7.9-7.10 oraz tabeli 7.5. Dla funkcji RFB poliharmonicznej wyniki były najmniej zadowalające, najlepsze wyniki wychodzą dla siatki $N=196$. Wyniki dla tego przypadku przedstawiono graficznie w rozdziale 7.1 na rysunkach 7.23-7.24 oraz tabeli 7.12.

W zagadnieniach niestacjonarnych dla funkcji RFB wielomianowej i funkcji poliharmonicznej wyniki wychodzą niesatysfakcjonujące. Spowodowane jest to złym uwarunkowaniem a tym samym rozwiązaniem „generowanego” przez metodę Kansa liniowego układu równań. Dla funkcji RBF gaussowskiej najlepsze rezultaty wychodzą dla siatki $N=64$, a wyniki dla tego przypadku przedstawiono graficznie w rozdziale 7.2 na rysunkach 7.26-7.27 oraz tabeli 7.14.

W przypadku zagadnienia stacjonarnego na obszarze nieregularnym uzyskane wyniki były tak jak w przypadku obszaru regularnego satysfakcjonujące. Dla funkcji RFB wielomianowej najlepsze wyniki zostały zaobserwowane dla siatki $N=38$ i parametru $c=0.1$. Wyniki dla tego przypadku przedstawiono graficznie w rozdziale 7.3 na rysunkach 7.40 i 7.41. Dla funkcji RFB gaussowskiej najlepsze wyniki zostały otrzymane dla siatki $N=38$ i parametru $c=3.0$. Wyniki dla tego przypadku przedstawiono graficznie w rozdziale 7.3 na rysunkach 7.42 i 7.43. Dla funkcji RFB poliharmonicznej najlepsze wyniki wychodzą dla siatki $N=38$ i parametru $c=0.1$. Wyniki dla tego przypadku przedstawiono graficznie w rozdziale 7.3 na rysunkach 7.44 i 7.45.

W przypadku zagadnienia niestacjonarnego dla obszaru nieregularnego dla funkcji RFB gaussowskiej i funkcji poliharmonicznej wyniki wychodzą niesatysfakcjonujące. Dla funkcji RBF wielomianowej najlepsze rezultaty wychodzą dla siatki $N=38$ i parametru $c=1.5$, wyniki dla tego przypadku przedstawiono graficznie w rozdziale 7.4 na rysunkach 7.46 i 7.47.

Jak można zauważyć powyżej, implementacja metody Kansa jest nieskomplikowana. To są główne zalety, że ta technika staje się popularna i została zastosowana w wielu dziedzinach nauki, w których zjawiska są opisywane cząstkowymi równaniami różniczkowymi. Jakkolwiek dowód rozwiązywalności dla liniowego systemu wynikającego z metody Kansa jest jeszcze niepotwierdzony, nawet dla eliptycznego problemu [1]. Uogólnienie tej metody dla innych problemów jest aktualnie badane. Ze względu na obszerność zagadnienia autor przedstawił tylko wybrane wyniki obliczeń numerycznych. Warto podkreślić zaletę środowiska programistycznego Scilab, które okazało się być efektywnym i efektywnym narzędziem do obliczeń numerycznych i inżynierskich.

9. Literatura

- [1] Jichun Li, *Mathematical justification for RBF – MFS*. Engineering Analysis with Boundary Elements. 2001; 25:897 – 901
- [2] Jichun Li, Hon YC, Chen CS *Numerical comparisons of two meshless methods using radial basis functions*. Engineering Analysis with Boundary Elements. 2002; 26:205 – 225
- [3] Bohdan Mochnecki, Józef S. Suchy, *Modelowanie i symulacja krzepnięcia odlewów*, Wydawnictwo naukowe PWN, 1993
- [4] Franke R. *Scattered data interpolation: tests of some of methods*. Math Comput 1982; 48:181 – 200
- [5] Hon YC, Lu MW, Xue WM, Zhou X. *A new formulation and computation of the triphasic model for mechano-electrochemical mixtures*. Comput Mech 1999; 24:155 – 165
- [6] Kansa EJ. *Multiquadric – a scattered data approximation scheme with applications to computational fluid dynamics II*. Comput Math Appl 1990; 19(8/9):147 – 161
- [7] Zerroukat M, Power H, Chen CS. *A numerical method for heat transfer problem using collocation and radial basis functions*. Int J Numer Meth Engng 1998; 42:1263 – 1278
- [8] Hardy RL, *Multiquadric equations of topography and other irregular surface*. J Geophys Res 1971; 76:1905 – 1915
- [9] Hon YC, Cheung KF, Mao XZ, Kansa EJ. *Multiquadric solution for shallow water equations*. ASCE J Hydraul Engng 1999; 125(5):524 – 533
- [10] <http://www.linux.com.pl/artykuly/2000/12/scilab.php>
- [11] <http://scilabsoft.inria.fr/>

10. Dodatki

Płyta CD zawiera:

1. Program instalacyjny pakietu Scilab
2. Prace magisterską w rozdziałach (pliki *.doc)
3. Kody źródłowe programów wykorzystanych do obliczeń:
 - s_mq_re.sce – program do obliczania rozwiązania na obszarze o regularnym brzegu z wykorzystaniem funkcji wielomianowej dla przepływu stacjonarnego
 - s_mq_nr.sce – program do obliczania rozwiązania na obszarze o nieregularnym brzegu z wykorzystaniem funkcji wielomianowej dla przepływu stacjonarnego
 - ns_mq_nr.sce – program do obliczania rozwiązania na obszarze o nieregularnym brzegu z wykorzystaniem funkcji wielomianowej dla przepływu niestacjonarnego
 - s_gs_re.sce – program do obliczania rozwiązania na obszarze o regularnym brzegu z wykorzystaniem funkcji gaussowskiej dla przepływu stacjonarnego
 - s_gs_nr.sce – program do obliczania rozwiązania na obszarze o nieregularnym brzegu z wykorzystaniem funkcji gaussowskiej dla przepływu stacjonarnego
 - ns_gs_re.sce – program do obliczania rozwiązania na obszarze o regularnym brzegu z wykorzystaniem funkcji gaussowskiej dla przepływu niestacjonarnego
 - s_ph_re.sce – program do obliczania rozwiązania na obszarze o regularnym brzegu z wykorzystaniem funkcji poliharmonicznej dla przepływu stacjonarnego
 - s_ph_nr.sce – program do obliczania rozwiązania na obszarze o nieregularnym brzegu z wykorzystaniem funkcji poliharmonicznej dla przepływu stacjonarnego