Proiect – Rezolvarea numerică a ecuației de difuzie a căldurii

1. Modelul matematic

Considerăm ecuația de difuzie staționară cu conductivitate variabilă:

$$-\nabla \cdot (k(x,y)\nabla u(x,y)) = f(x,y), \quad (x,y) \in \Omega \subset \mathbb{R}^2 \tag{1}$$

cu condiții la frontieră:

$$u(x,y) = g_D(x,y), \qquad (x,y) \in \Gamma_D \subset \partial\Omega \tag{2}$$

$$k(x,y)\frac{\partial u}{\partial n}(x,y) = g_N(x,y), \qquad (x,y) \in \Gamma_N = \partial\Omega \setminus \Gamma_D \tag{3}$$

Domeniul ales este un dreptunghi $[0,a] \times [0,b]$ cu $a = 5$, $b = 2$. Conductivitatea este:

$$k(x,y) = 1 + 0.5 \cdot \sin(2\pi x) \cdot e^{-y} \tag{4}$$

iar funcția exactă aleasă pentru fluxul de căldură este:

$$u(x,y) = \sin\left(\frac{2\pi x}{3}\right)\cos\left(\frac{\pi y}{2}\right) \tag{5}$$

Din aceasta, calculăm sursa de căldură $f(x,y)$.

2. Discretizarea domeniului

Pentru a discretiza domeniul dreptunghiului, vom discretiza cele două axe Ox și Oy, folosind câte $(N+1)$ noduri pentru fiecare.

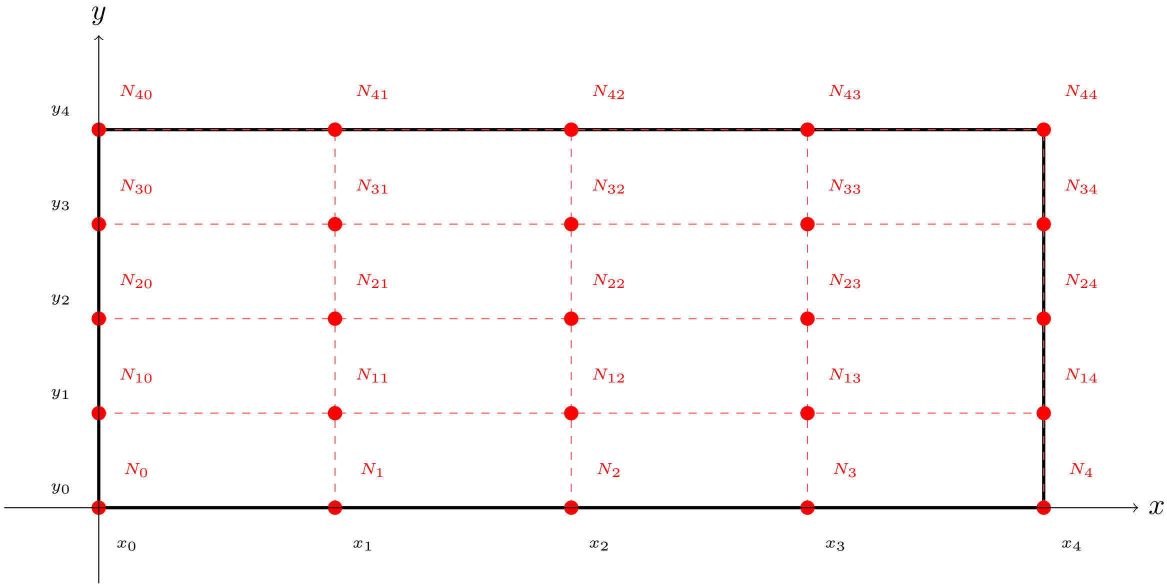


Figure 1: Discretizare a domeniului $[0,5] \times [0,2]$ în 5×5 noduri ($N = 4$)

3. Aproximarea ecuației cu diferențe finite

Ecuația:

$$-\nabla \cdot (k(x,y)\nabla u(x,y)) = f(x,y)$$

este discretizată folosind metoda diferențelor finite de ordin 2, pe un grid uniform de $(N+1) \times (N+1)$ puncte. Notăm cu $h_x = a/N$, $h_y = b/N$ pasul de discretizare pe direcțiile x și y.

Noduri interioare

Pentru nodurile interioare (i,j):

$$-\frac{k_{i,j}}{h_x^2}(u_{i-1,j} - 2u_{i,j} + u_{i+1,j}) - \frac{k_{i,j}}{h_y^2}(u_{i,j-1} - 2u_{i,j} + u_{i,j+1}) = f_{i,j}$$

Condiții Dirichlet

Pe Γ_D $(y = 0$ sau $y = b)$:

$$u(x_i, y_j) = g_D(x_i, y_j)$$

Condiții Neumann

Pentru $x = 0$ și $x = a$:

$$-k(x,y)\frac{\partial u}{\partial n} \approx \begin{cases} \frac{k_{i+1,j}+k_{i,j}}{2h_x}(u_{i+1,j} - u_{i,j}), & x = 0 \\ \frac{k_{i,j}+k_{i-1,j}}{2h_x}(u_{i,j} - u_{i-1,j}), & x = a \end{cases}$$

Rezolvarea sistemului liniar

Forma matriceală

Problema aproximată prin metoda diferențelor finite duce la un sistem liniar de forma:

$$A \cdot U = b$$

- A este o matrice rară (sparse) de dimensiune $(N+1)^2 \times (N+1)^2$, care encodează relațiile dintre valorile necunoscute în punctele grilei.
- Matricea este construită inițial în format `LIL` (*List of Lists*) pentru a permite inserarea eficientă de elemente. După completare, este convertită în format `CSR` (*Compressed Sparse Row*), optim pentru operații numerice și pentru rezolvarea sistemului liniar.
- Pentru interpolarea spline pătratică unidimensională (folosită ulterior la interpolarea bidimensională), se folosește factorizarea QR, implementată manual. Sistemul rezultat este de forma:
$$A = QR \Rightarrow U = R^{-1}Q^T b$$
- Această rezolvare este realizată prin funcția `rezolva_sistem_QR`, care efectuează pas cu pas ortogonalizarea Gram-Schmidt și înapoi-substituția.
- U este vectorul soluție, cu $(N+1)^2$ elemente, care conține valorile funcției necunoscute (temperatura, de exemplu) în fiecare nod al grilei. Aceste valori corespund punctelor interioare și de pe frontieră, așa cum sunt reprezentate cu N în figura 1.
- Pentru rezolvarea sistemului liniar global (cel rezultat din discretizarea EDP-ului pe domeniu), se utilizează funcția `spsolve` din `scipy.sparse.linalg`. Aceasta folosește factorizare **LU** internă, adaptată pentru matrice sparse în format `CSR`. Este mult mai eficientă computațional decât implementarea manuală QR și este utilizată pentru a obține rapid soluția numerică în cazul grilelor dense (cu N mare).

Construirea aproximării finale folosind o interpolare

După rezolvarea sistemului liniar algebric obținut prin metoda diferențelor finite, soluția numerică este disponibilă doar în nodurile grilei (x_i, y_j). Pentru a obține o aproximare continuă a soluției pe întreg domeniul, s-a utilizat o metodă de interpolare spline pătratică bidimensională.

Această abordare constă în următorii pași:

- **Interpolare pe linii (direcția x):** Pentru fiecare linie orizontală fixă y_j, se construiește o interpolare spline pătratică $s_j(x)$ care aproximează valorile u_{ij} în punctele (x_i, y_j). Acest lucru se face folosind o funcție de interpolare spline pătratică unidimensională bazată pe continuitatea funcției și a derivatei prime.
- **Evaluarea intermediară:** Se evaluează fiecare spline $s_j(x)$ în puncte dense $x \in \{x_k^{\text{dense}}\}$, obținând o matrice intermediară $Z_{\text{row}}[j,k] = s_j(x_k^{\text{dense}})$.
- **Interpolare pe coloane (direcția y):** Pentru fiecare coloană fixă x_k^{dense}, se construiește o spline pătratică $t_k(y)$ pe baza valorilor $Z_{\text{row}}[:,k]$, rezultând astfel o aproximare continuă și în direcția y.
- **Evaluarea finală:** Se evaluează fiecare spline $t_k(y)$ în puncte dense y_l^{dense}, rezultând matricea completă $Z_{\text{dense}}[l,k]$, care aproximează soluția pe un grid dens.
- **Compararea cu soluția exactă:** Se evaluează soluția exactă $u(x,y)$ în aceleași puncte $(x_k^{\text{dense}}, y_l^{\text{dense}})$ și se calculează eroarea absolută punctuală. Aceasta este folosită pentru a evalua acuratețea metodei.

Această interpolare bidimensională prin spline pătratice asigură o aproximare netedă și continuă a soluției pe întreg domeniul, cu un cost computațional redus, datorită structurii simple a spline-urilor pătratice și a implementării optimizate prin metode directe (ex. factorizare QR).

În codul Python, funcția `spline_bi2d` construiește această interpolare bidimensională, iar evaluarea pe un grid dens se face secvențial: mai întâi pe rânduri, apoi pe coloane, pentru a reduce redundanța și costul de calcul. Această abordare modulară este eficientă și extensibilă pentru diverse aplicații numerice în 2D.

4. Convergența erorii

Pentru a evalua acuratețea metodei numerice implementate, a fost analizată eroarea absolută maximă dintre soluția numerică și soluția exactă $u(x, y)$, pe o grilă de test. Eroarea a fost evaluată pe o grilă mai densă, folosind o interpolare spline pătratică bidimensională aplicată soluției numerice definite pe grila discretizată.

Pentru a estima rata de convergență, am construit graficul $\log_{10}(\text{eroare})$ în funcție de $\log_{10}(h)$, unde $h = h_x = h_y$ este pasul de discretizare. Relația liniară între aceste două mărimi indică o convergență de tip:

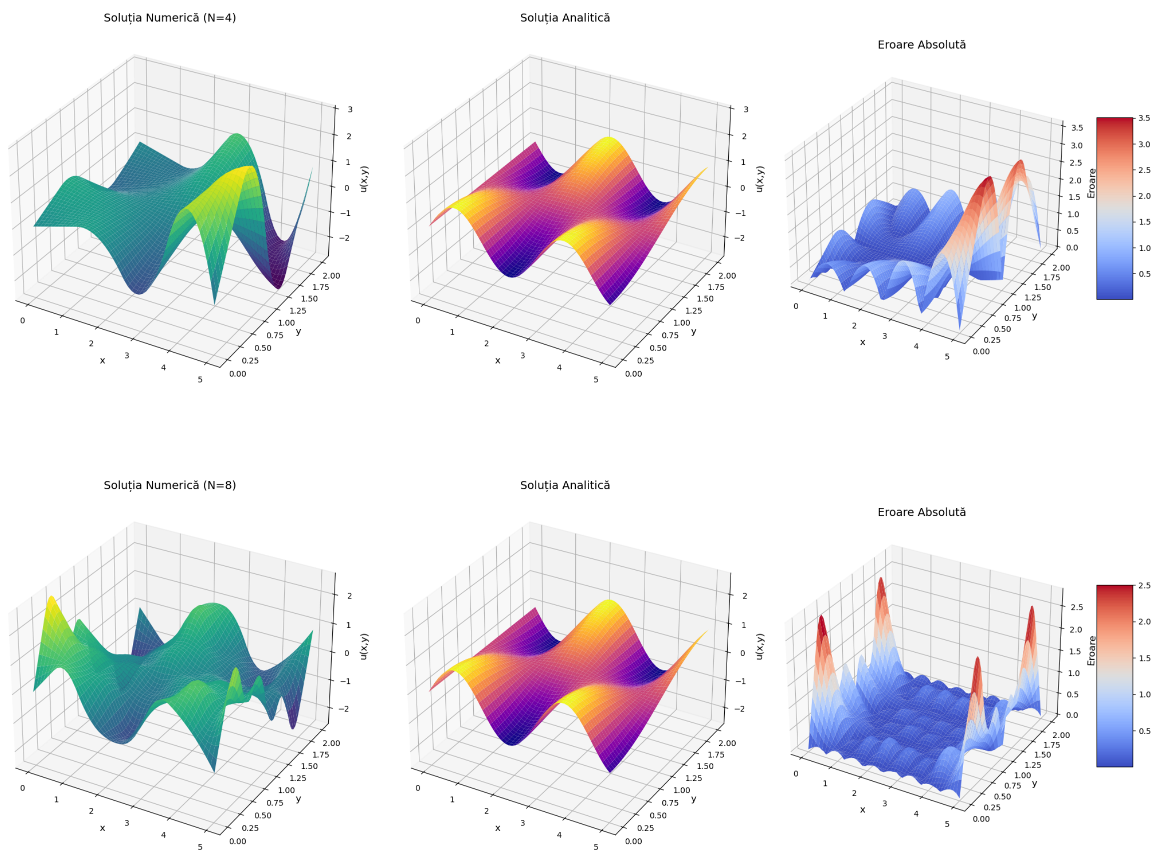
$$\max_{x,y} |u(x, y) - \tilde{u}_h(x, y)| \approx C \cdot h^p \tag{6}$$

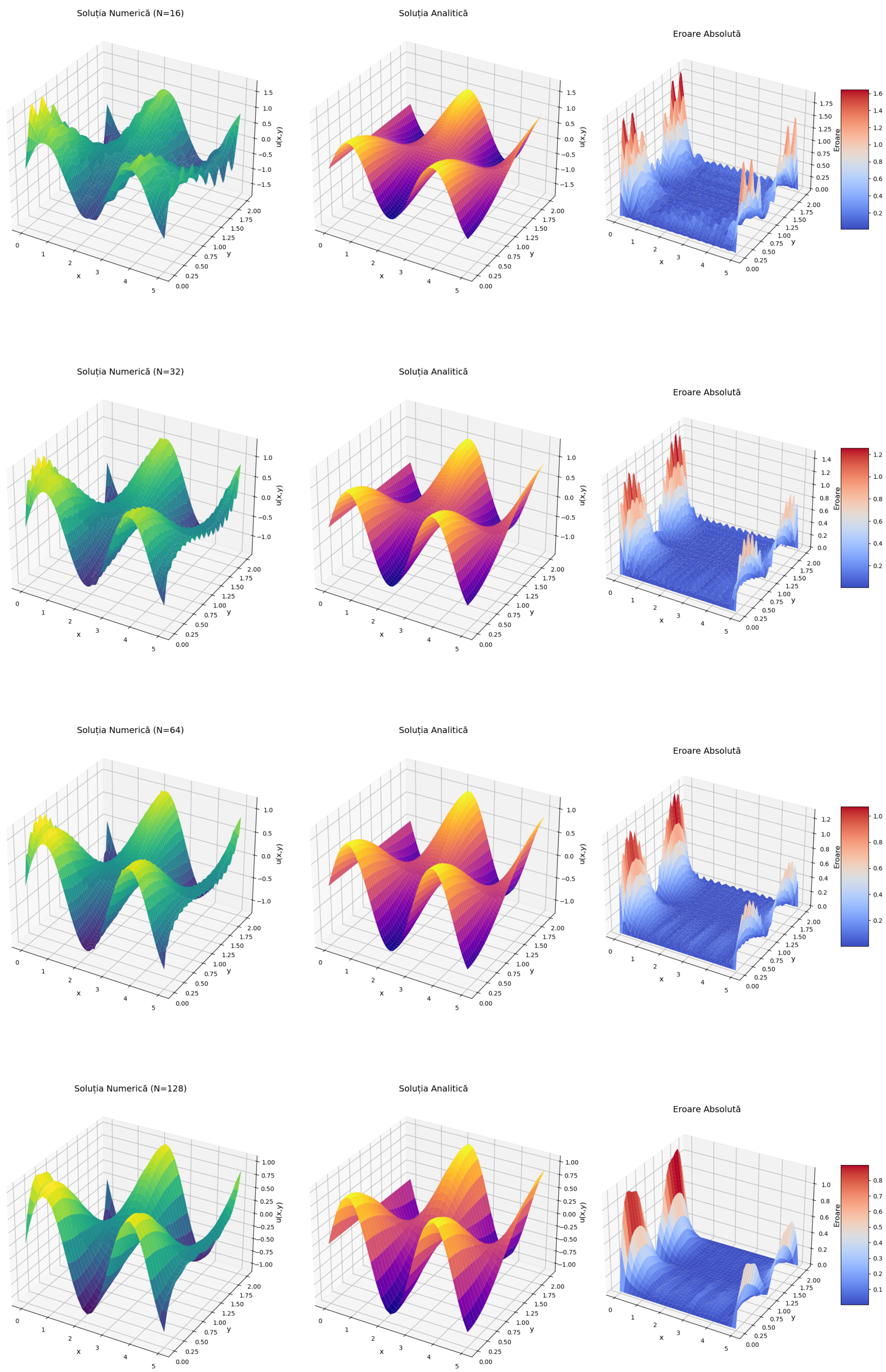
unde p este ordinea de convergență. Estimăm p numeric prin regresie liniară aplicată pe datele logaritmice:

$$\log_{10}(\text{eroare}) = p \cdot \log_{10}(h) + \log_{10}(C) \tag{7}$$

De asemenea, ratele locale de convergență între grile succesive (N_i, N_{i+1}) au fost calculate ca

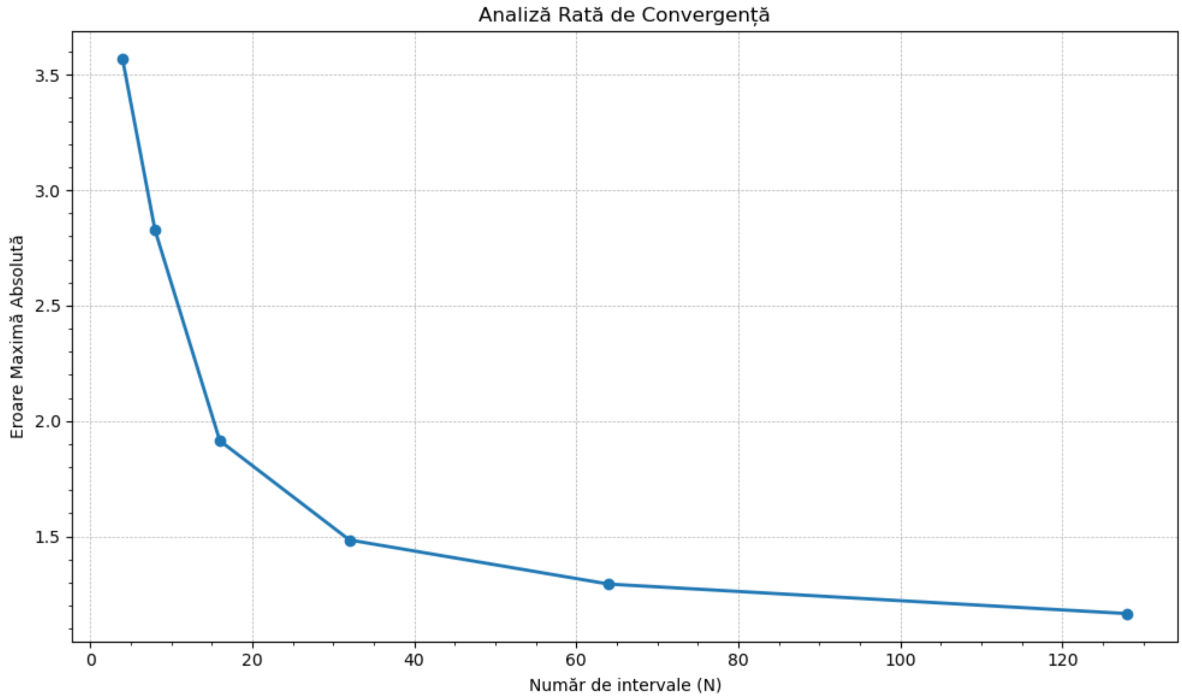
$$p_i = \frac{\log(\text{eroare}_{i+1}) - \log(\text{eroare}_i)}{\log(h_{i+1}) - \log(h_i)} \tag{8}$$




Soluția Numerică (N=16)
Soluția Analitică
Eroare Absolută
Soluția Numerică (N=32)
Soluția Analitică
Eroare Absolută
Soluția Numerică (N=64)
Soluția Analitică
Eroare Absolută
Soluția Numerică (N=128)
Soluția Analitică
Eroare Absolută
x
y
u(x,y)
Eroare

Rate de convergență

-Rata globală de convergență: **0.38**



5. Concluzii

Lucrarea a prezentat o metodă numerică completă pentru rezolvarea ecuației staționare a difuziei căldurii cu coeficient variabil, utilizând diferențe finite de ordin doi pe un grid bidimensional. Sistemul algebric rezultat a fost tratat eficient folosind structuri de date sparse și metode numerice pentru a ajusta rezolvarea normal analitică la una rezolvabila de calculator: factorizare QR (pentru spline-uri unidimensionale) și factorizare LU (prin `spsolve`) pentru sistemul global.

Pentru a obține o soluție continuă pe întreg domeniul, s-a construit o interpolare spline pătratică bidimensională, care a oferit o reprezentare precisă a soluției numerice. Testele numerice au validat metoda ce creste pe măsură ce mărim numărul de noduri N.

Datorită organizării clare a codului și a metodelor utilizate pentru interpolare și vizualizare, modelul obținut este ușor de adaptat și extins pentru alte probleme similare de difuzie în 2D.