

Difuzia căldurii într-un microprocesor în 1D, 2D și 3D

Soisun Mina-David, Jianu Toma, Grupa 164

1 iunie 2025

1 Introducere

În acest proiect vom studia modul de dispersare a căldurii într-un microprocesor, o problemă din ce în ce mai importantă datorită creșterii consumului de putere electrică a componentelor de calculator moderne. Progresul tehnic al electronicii și al litografiei a mărit semnificativ eficiența chip-urilor care procesează toate datele pe care le folosim zi de zi. Cu toate acestea, extinderea puterii de calcul duce inevitabil la o intensificare a cantității de căldură degajate de acestea.

Vom începe cu un model simplu în 1D, ca mai apoi să scalăm întreg produsul în 2D și, în ultimă instanță, în 3D. Introducerea conductivității termice. $k(x)$, $k(x, y)$, $k(x, y, z)$, discretizarea domeniului $\Omega = (0, L)$ în 1D, $\Omega = (0, L_x) \times (0, L_y)$ în 2D și $\Omega = (0, L_x) \times (0, L_y) \times (0, L_z)$ în 3D ne vor permite rezolvarea ecuațiilor diferențiale propuse la curs și calculul erorii pentru crearea unui grafic logaritmic al acestora în funcție de creșterea numărului de noduri de discretizare.

2 Modelele matematice și tipul de microchip

2.1 Sistemul de ecuații diferențiale

Din ecuația principiului I al termodinamicii deducem faptul că difuzia căldurii poate fi modelată matematic cu ajutorul ecuațiilor diferențiale.

Am notat:

- Ω - Domeniul pe care studiem distribuția temperaturii
- $u(x, y, z)$ — Distribuția temperaturii în domeniul Ω
- $k(x, y, z)$ — Conductivitatea termică a materialului (poate fi constantă sau dependentă de spațiu)
- $f(x, y, z)$ — Sursa internă de căldură (ex. căldura generată de tranzistori)
- $\Gamma_D \subset \partial\Omega$ — Porțiunea frontierei unde se aplică condiții de tip Dirichlet
- $\Gamma_N = \partial\Omega \setminus \Gamma_D$ — Porțiunea frontierei unde se aplică condiții de tip Neumann
- $g_D(x, y, z)$ — Condiția Dirichlet: temperatură impusă pe Γ_D
- $g_N(x, y, z)$ — Condiția Neumann: flux termic impus pe Γ_N

Modelele matematice pe care se bazează această lucrare sunt următoarele:

Pentru modelul 1D se va folosi o variantă rudimentară a modelului difuziei unde am considerat conductivitatea $k(x) = 1, \forall x \in (0, L)$ și o stare de echilibru termic unde variabila timpului nu intervine deloc în difuzia temperaturii $u(x, t) = u(x), \forall t \geq 0$. Astfel, dacă știm temperatura materialului în capăt, putem determina temperatura acestuia în orice punct:

$$\begin{cases} -u''(x) = f(x), & \forall x \in (0, L) \\ u(0) = A, \quad u(L) = B \end{cases}$$

Apoi am trecut la modelarea cu un $k(x)$ neconstant, deci ecuația diferențială se transformă în:

$$\begin{cases} -\frac{d}{dx} \left(k(x) \frac{du}{dx} \right) = f(x), & \forall x \in (0, L) \\ u(0) = A, \quad u(L) = B \end{cases}$$

Pentru 2D vom folosi un model mai sofisticat unde introducem o nouă variabilă de care sa depindă temperatura dar și conductivitate neconstantă:

$$\begin{cases} -\nabla \cdot (k(x, y) \nabla u(x, y)) = f(x, y), & \text{în } \Omega, \\ u(x, y) = g_D(x, y), & \text{pe } \Gamma_D \subset \partial\Omega, \\ k(x, y) \frac{\partial u}{\partial n}(x, y) = g_N(x, y), & \text{pe } \Gamma_N = \partial\Omega \setminus \Gamma_D. \end{cases}$$

unde cu $\frac{\partial u}{\partial n}(x, y)$ s-a notat derivata normală a funcției u la frontieră.

Extinderea la 3D va fi modelată cu ajutorul ecuațiilor:

$$\begin{cases} -\nabla \cdot (k(x, y, z) \nabla u(x, y, z)) = f(x, y, z), & \text{în } \Omega, \\ u(x, y, z) = g_D(x, y, z), & \text{pe } \Gamma_D \subset \partial\Omega, \\ k(x, y, z) \frac{\partial u}{\partial n}(x, y, z) = g_N(x, y, z), & \text{pe } \Gamma_N = \partial\Omega \setminus \Gamma_D. \end{cases}$$

2.2 Tipul de microchip: Intel 8086

Intel 8086 este un microprocesor produs de compania Intel între 1978 și anii 1990. A fost baza primelor calculatoare personale dezvoltat de IBM în anii '80. Procesorul 8086 a stabilit Intel ca un titan în lumea calculului și a chip-urilor de siliciu. Astfel de procesoare au fost concepute cu o soluție de răcire a cărui masă critică să se concentreze pe centrul său, locul de unde se generează majoritatea căldurii. În aceeași manieră se vor dezvolta și modelele noastre, cu sursa căldurii în "inima" domeniului.

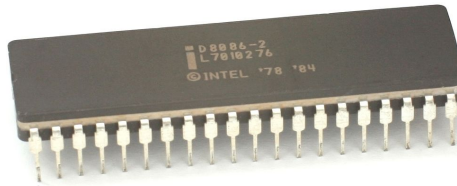


Figura 1: Intel 8086

3 Modelarea pe 1D

3.1 Mențiuni

Pentru a compara modelul dezvoltat de noi cu unul exact am creat grafice atât pentru $u_{exact}(x) = \sin(\pi x)$, deci $f_{exact}(x) = -\pi^2 \sin(\pi x)$, cât și pentru aproximarea numerică. Am ales parametrii $A = 0$ și $B = 0$ și lungimea $L = 1$. De asemenea, am modelat folosind un coeficient $k(x) = x + 1$ pentru a simula o conductivitate termică neconstantă.

3.2 Definirea Pasului de Discretizare

Domeniul $[0, L]$ este împărțit în $N + 1$ subintervale egale:

$$h = \frac{L}{N + 1}$$

unde:

- h - pasul de discretizare
- N - numărul de noduri interioare

$h = L / (N + 1)$ # Pasul de discretizare
 $x = \text{np.linspace}(0, L, N+2)$ # Nodurile

3.3 Generarea Nodurilor

Se definesc $N + 2$ noduri echidistante:

$$x_i = i \cdot h, \quad i = 0, 1, 2, \dots, N + 1$$

Exemplu pentru $L = 1, N = 4$:

$$h = 0.2 \Rightarrow \text{Nodurile: } \{0.0, 0.2, 0.4, 0.6, 0.8, 1.0\}$$

3.4 Clasificarea Nodurilor

- **Noduri de graniță** (valori cunoscute):

- x_0 cu $u_0 = A$
- x_{N+1} cu $u_{N+1} = B$

- **Noduri interioare** (necunoscute):

$$x_1, x_2, \dots, x_N$$

- **Exemplu de Numerotare** ($N = 3$)

Index (i)	x_i	Tip nod	Valoare
0	0.0	Graniță	$u_0 = A$ (cunoscut)
1	0.25	Interior	u_1 (necunoscut)
2	0.5	Interior	u_2 (necunoscut)
3	0.75	Interior	u_3 (necunoscut)
4	1.0	Graniță	$u_4 = B$ (cunoscut)

3.5 Aproximarea cu diferențe finite

Ecuția este aproximată folosind:

$$-\frac{k_{i+\frac{1}{2}}(u_{i+1} - u_i) - k_{i-\frac{1}{2}}(u_i - u_{i-1})}{h^2} = f_i \quad (1)$$

Implementarea în Python:

```
# Evaluarea lui k in punctele medii
x_mijloc = 0.5 * (x[:-1] + x[1:])
k_mijloc = k(x_mijloc)

# Constructia matricei
diagonala_principala = (k_mijloc[:-1] + k_mijloc[1:]) / h**2
diagonala_secundara = -k_mijloc[1:-1] / h**2
```

3.6 Sistemul liniar

Se obține un sistem tridiagonal:

$$\mathbf{A}\mathbf{u} = \mathbf{F} \quad (2)$$

Rezolvat cu:

```
u_interior = np.linalg.solve(A_mat, F)
```

3.7 Soluția finală

Soluția este asamblată din:

```
u = np.zeros(N+2)
u[0], u[-1] = A, B # Conditii la limita
u[1:-1] = u_interior # Solutia in interior
```

3.8 Analiza erorii

Eroarea maximă este calculată ca:

$$e = \max_{0 \leq i \leq N+1} |u_i - u_{\text{exact}}(x_i)| \quad (3)$$

```
h_values = [L/(N+1) for N in N_values]
erori = [np.max(np.abs(u - u_ex)) for u, u_ex in solutions]
```

3.9 Rezultate numerice

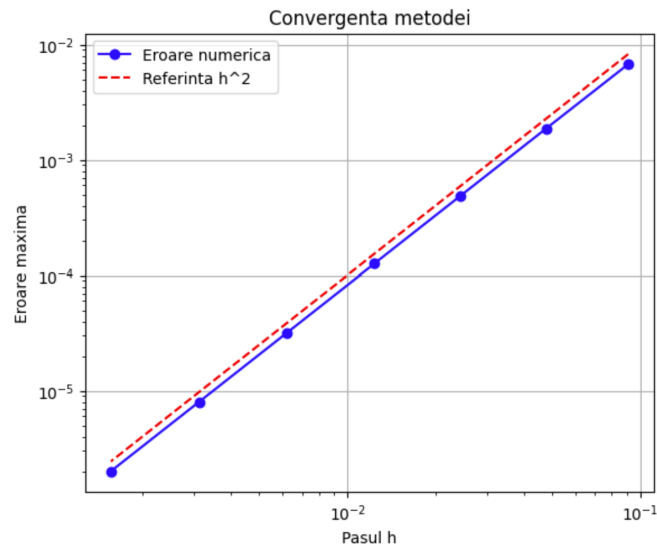


Figura 2: Convergența metodei

După cum se poate observa din graficul creat în scala logaritmică din figura 2, metoda folosită se apropie foarte mult de eroarea $O(h^2)$

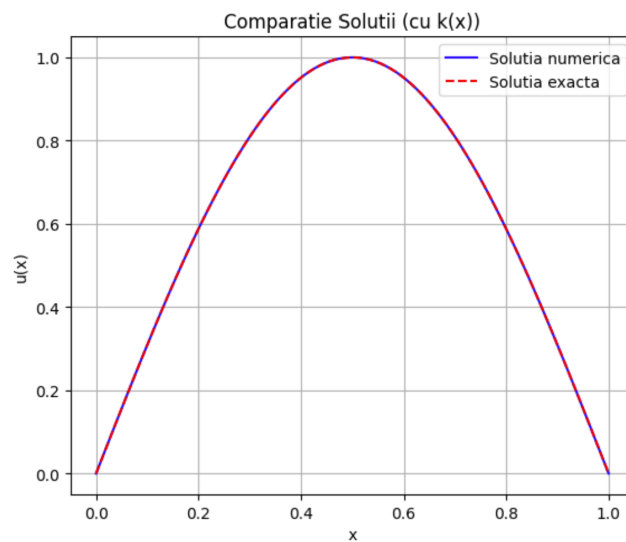


Figura 3: Grafice soluții

Eficiența metodei și corectitudinea ei se pot observa și din figura 3 care scoate în evidență faptul că graficele celor 2 funcții se suprapun aproape perfect.

4 Modelarea în 2D

4.1 Mențiuni

Vom considera vederea de sus în jos a procesorului Intel 8086 pentru această modelare în 2 dimensiuni. Astfel, $\Omega = [0, 1] \times [0, 1]$. Reducerea domeniului la un pătrat de latură de lungime arbitrară 1 ne permite să discretizăm în mod egal fiecare axă și astfel obținem un model omogen care să se apropie de realitate

4.2 Matricea conductivității și nodurile Dirichlet și Neumann

- $k(x, y) = \begin{bmatrix} 1+x^2 & 0 \\ 0 & 1+y^2 \end{bmatrix}$ este coeficientul de conductivitate;
- $u_{exact}(x, y) = e^{\alpha((x-1/2)^2+(y-1/2)^2)}$ este o funcție sursă definită astfel încât soluția exactă să fie cunoscută.
- $\alpha = 30$ este un coeficient ales în așa fel încât sursa de caldură să fie concentrată în mijlocul domeniului

Condiții la frontieră:

- **Dirichlet** pe marginile $x = 0$ și $x = 1$: $u = u_{exact}$;
- **Neumann omogene** (izolație): $\frac{\partial u}{\partial y} = 0$ pentru $y = 0$ și $y = 1$.

4.3 Discretizarea domeniului și numerotarea nodurilor

Domeniul pătrat este discretizat cu pas spațial $h = \frac{1}{n-1}$, folosind o grilă uniformă cu $n \times n$ puncte. Numerotarea se face implicit bidimensional, iar funcțiile sunt stocate ca matrice în Python: $u[i, j] \approx u(x_i, y_j)$.

4.4 Aproximarea ecuației cu diferențe finite

Discretizăm operatorul divergență:

$$-\nabla \cdot (k \nabla u) \approx - \left(\frac{\partial}{\partial x} \left(k_x \frac{\partial u}{\partial x} \right) + \frac{\partial}{\partial y} \left(k_y \frac{\partial u}{\partial y} \right) \right)$$

Aplicăm diferențe finite centralizate în punctul (i, j) :

$$\begin{aligned} (k_x u_x)_x &\approx \frac{1}{h^2} (k_{i+1,j} u_{i+1,j} - (k_{i+1,j} + k_{i-1,j}) u_{i,j} + k_{i-1,j} u_{i-1,j}) \\ (k_y u_y)_y &\approx \frac{1}{h^2} (k_{i,j+1} u_{i,j+1} - (k_{i,j+1} + k_{i,j-1}) u_{i,j} + k_{i,j-1} u_{i,j-1}) \end{aligned}$$

4.5 Sistemul liniar și metoda de rezolvare

Obținem un sistem de tip:

$$A\mathbf{u} = \mathbf{f},$$

unde A este o matrice unde majoritatea elementelor sunt egale cu 0, iar \mathbf{u} și \mathbf{f} sunt vectori cu valorile necunoscute, respectiv sursa.

Rezolvarea se face iterativ, prin determinarea elementelor $u_{i,j}$ astfel:

$$u_{i,j} = \frac{1}{\text{coef}} (\text{suma vecini ponderați} + h^2 f_{i,j})$$

Condițiile de tip Neumann $\frac{\partial u}{\partial y} = 0$ sunt implementate astfel:

$$\begin{aligned} \left. \frac{\partial u}{\partial y} \right|_{j=0} &\approx \frac{u_{i,1} - u_{i,0}}{h} = 0 \Rightarrow u_{i,0} = u_{i,1}, \\ \left. \frac{\partial u}{\partial y} \right|_{j=n-1} &\approx \frac{u_{i,n-1} - u_{i,n-2}}{h} = 0 \Rightarrow u_{i,n-1} = u_{i,n-2}. \end{aligned}$$

4.6 Compararea soluțiilor

Pentru a vizualiza soluția numerică și cea exactă, se utilizează funcții de contur (`contourf`) pentru:

- Soluția numerică
- Soluția exactă (analitică)
- Harta erorii absolute.
- Convergența la eroarea de $O(h^2)$

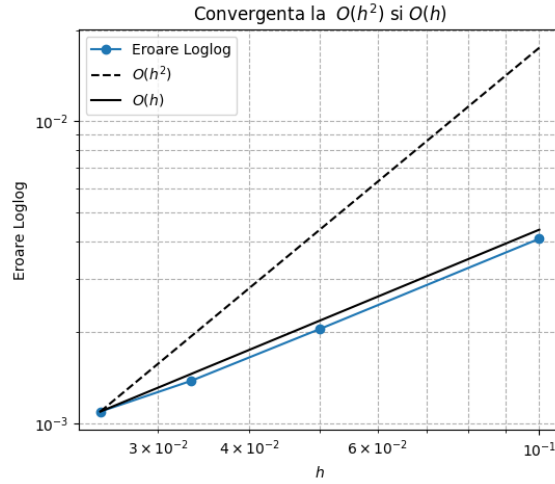


Figura 4: Convergența la $O(h^2)$

Din cauza integrării Nodurilor Neumann care au o eroare de $O(h)$ rezultată de integrarea metodei diferențelor finite cu formula de mai sus, eroarea se depărtează de dreapta corespunzătoare lui $O(h^2)$, dar este foarte apropiată de $O(h)$.

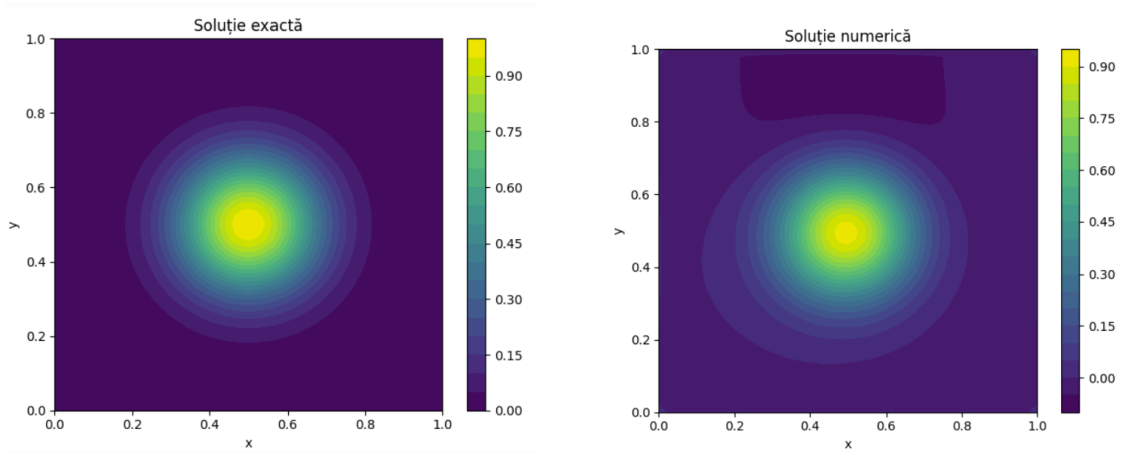


Figura 5: Graficul soluțiilor obținute

Acesta este rezultatul final al modelului, unde sursa de caldură are concentrația maximă de caldură iar dispersia se face în întreg planul determinat de domeniul Ω , imitând astfel comportamentul real al unui microprocesor.

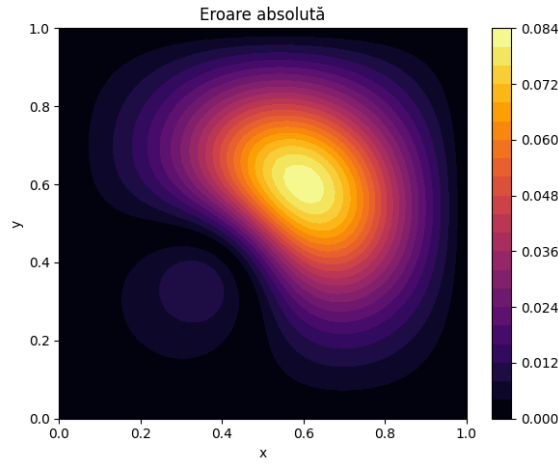


Figura 6: Graficul erorii absolute

Din acest grafic reiese faptul ca modelul numeric se apropie cu diferențe de maximum 8×10^{-2} în modul față de modelul exact

4.7 Eficientizarea modelului numeric

Din motive de eficiență, am demarat procesul de studiu și de integrare a unor metode pentru a scade eroarea dintre modelul numeric dezvoltat de noi și modelul exact.

În urma consultării materialelor, am decis sa integrăm metoda diferențelor centrale și a nodurilor fictive pentru reducerea erorii si apropierea de $O(h^2)$,

Metoda Nodurilor Fictive Pentru condițiile Neumann:

$$\left. \frac{\partial u}{\partial x} \right|_{x=0} = g_0(y), \quad \left. \frac{\partial u}{\partial x} \right|_{x=1} = g_1(y)$$

Se introduc noduri fictive $u_{-1,j}$ și $u_{n,j}$:

$$\begin{cases} \frac{u_{1,j} - u_{-1,j}}{2h} = g_0(y_j) & (\text{diferență centrală}) \\ \frac{u_{n,j} - u_{n-2,j}}{2h} = g_1(y_j) & (\text{diferență centrală}) \end{cases}$$

Versiunea nouă introduce medierea la interfețe:

$$\begin{cases} k_{x,i+1/2,j} = \frac{1}{2}(k_{x,i,j} + k_{x,i+1,j}) \\ k_{x,i-1/2,j} = \frac{1}{2}(k_{x,i,j} + k_{x,i-1,j}) \\ k_{y,i,j+1/2} = \frac{1}{2}(k_{y,i,j} + k_{y,i,j+1}) \\ k_{y,i,j-1/2} = \frac{1}{2}(k_{y,i,j} + k_{y,i,j-1}) \end{cases}$$

Eliminând nodurile fictive, obținem:

$$\begin{aligned} u_{0,j} &= u_{1,j} - hg_0(y_j) + \frac{h^2}{2} \left. \frac{\partial^2 u}{\partial x^2} \right|_{x=0} + O(h^3) \\ u_{n-1,j} &= u_{n-2,j} + hg_1(y_j) + \frac{h^2}{2} \left. \frac{\partial^2 u}{\partial x^2} \right|_{x=1} + O(h^3) \end{aligned}$$

Analiză Comparativă

Caracteristică	Versiunea veche	Versiunea nouă
Aproximare coeficienți	Valori discrete	Mediere la interfețe
Convergență	$O(h^2)$	$O(h^3)$
Stabilitate numerică	Bună	Îmbunătățită
Tratare condiții Neumann	Diferențe unilaterale	Diferențe centrate

Tabela 1: Comparatie între versiuni

Implementarea Numerică [1] Calculează $u_{\text{fictiv_stânga}} = u_{1,:} - 2h \cdot g_0(y)$ Calculează $u_{\text{fictiv_dreapta}} = u_{n-2,:} + 2h \cdot g_1(y)$ Actualizează granițele: $u_{0,1:-1} = \frac{1}{2}(u_{\text{fictiv_stânga}} + u_{1,1:-1})$ $u_{n-1,1:-1} = \frac{1}{2}(u_{\text{fictiv_dreapta}} + u_{n-2,1:-1})$

Graficul log-log arată convergența erorii care indică apropierea de $O(h^3)$ datorită adăugării a încă unui nivel de precizie, iar cel al erorii absolute arată diferențe între cele 2 metode de maximum 5×10^{-3}

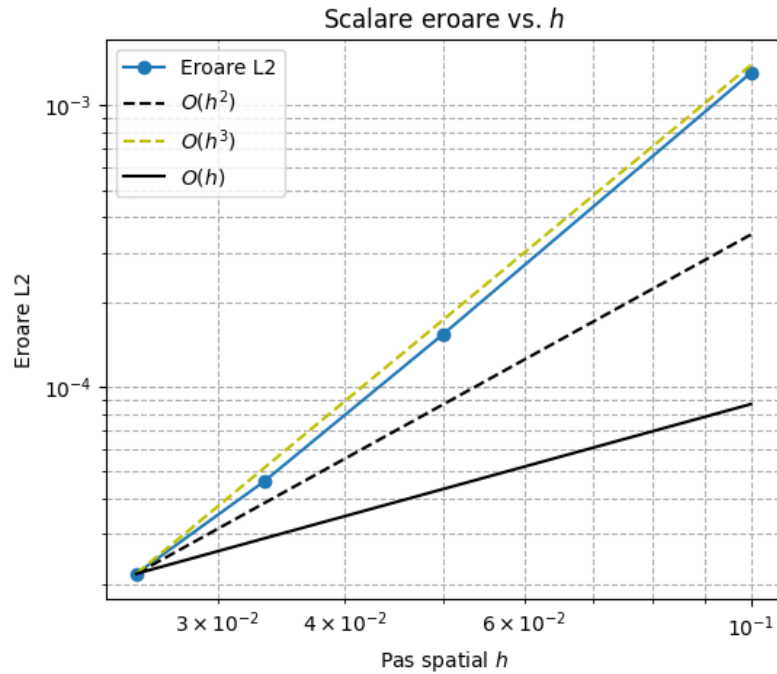


Figura 7: Grafic loglog

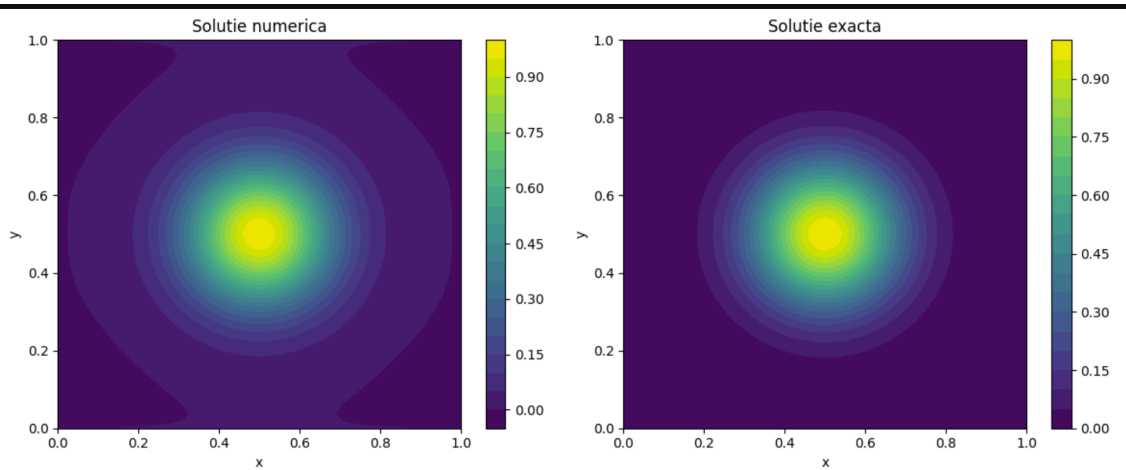


Figura 8: Soluții 2D îmbunătățite

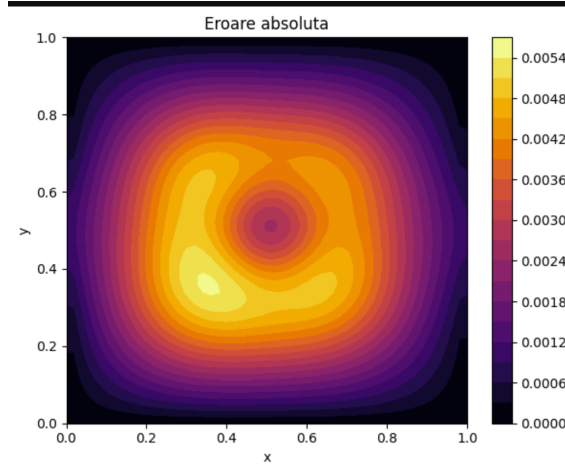


Figura 9: Eroare absoluta îmbunătățită

5 Modelarea în 3D

5.1 Mențiuni

Extinderea modelului pentru a include a treia dimensiune s-a bazat pe modelul 2D. Analiza și interpretarea rezultatelor se va face prin 2 tipuri de grafice: un "heat map" 2D la jumătatea înălțimii domeniului $\Omega = [0, 1] \times [0, 1] \times [0, 1]$ și un grafic efectiv al cubului realizat cu funcția slice pentru a împărți domeniul în eșantioane din 3 în 3 elemente ale discretizării care au fost plotate cu scatter și colorate în funcție de temperatura lor caracteristică din calcule.

În lumea reală, procesoarele moderne nu mai au miezul de Siliciu plasat în mijlocul geometric al lor, ci l-au plasat cât mai sus pentru a asigura un transfer al căldurii cât mai eficient. Fiind foarte fragil, acest miez de Siliciu este acoperit de un strat de cupru, metal cunoscut pentru proprietățile foarte bune de a conduce căldura. Pentru a atinge eficiența maximă în procesul de răcire al procesoarelor lor, majoritatea producătorilor de laptop își montează soluțiile de răcire direct pe partea de siliciu, astfel asigurând un transfer cât mai bun între cele 2 medii conductoare.



Figura 10: Procesor Intel i9-12900K separat de stratul de cupru protector

Pentru a ilustra aceste avansuri în lumea tehnicii de calcul, vom muta sursa de căldură pe axa Z și vom face o comparație calitativă între plasarea ei în mijlocul modelului la coordonatele $x = 0.5, y = 0.5, z = 0.5$, apoi în punctul de coordonate $x = 0.5, y = 0.5, z = 0.75$ și în final, în punctul $x = 0.5, y = 0.5, z = 0.9$ pentru a demonstra cum această deplasare a sursei de căldură afectează transferul către marginile modelului

În plus, pentru analiza diferențelor între modelele cu Noduri Neumann și Dirichlet și cele care includ doar noduri Dirichlet, vom vizualiza schimbările generate de adăugarea fluxului de căldură în aproximările numerice.

5.2 Modelul matematic

- Tensorul de conductivitate: $k(x, y, z) = \begin{cases} k_x = 5(1 + x^2) \\ k_y = 5(1 + y^2) \\ k_z = 5(1 + z^2) \end{cases}$

- Termenul sursă $f(x, y, z)$ calculat analitic din soluția exactă
- Soluția exactă: $u_{\text{exacta}}(x, y, z) = e^{-\alpha[(x-0.5)^2+(y-0.5)^2+(z-0.5)^2]}$ cu $\alpha = 30$

5.3 Discretizarea Domeniului

Domeniul cubic $[0, 1]^3$ este discretizat uniform:

- Număr de noduri: $n \times n \times n$
- Pasul de discretizare: $h = \frac{L}{n-1}$ cu $L = 1.0$
- Nodurile sunt indexate cu tripleți (i, j, k) pentru direcțiile x, y, z

5.4 Aproximarea cu Diferențe Finite

Schema de discretizare utilizată:

$$\frac{k_x^{i+\frac{1}{2}}(u_{i+1,j,k} - u_{i,j,k}) - k_x^{i-\frac{1}{2}}(u_{i,j,k} - u_{i-1,j,k})}{h^2} + (\text{termeni similari pentru } y, z) = f_{i,j,k} \quad (4)$$

unde coeficienții la interfețe se calculează ca medie armonică:

$$k_x^{i+\frac{1}{2}} = \frac{2k_x^i k_x^{i+1}}{k_x^i + k_x^{i+1}} \quad (5)$$

5.5 Sistemul Liniar și Rezolvarea

- Sistemul liniar rezultat este rezolvat iterativ cu metoda descompunerii LU
- 200 de iterații sunt suficiente pentru convergență
- Condiții la limită:
 - Dirichlet pe fețele $y = 1, z = 0, z = 1$
 - Neumann pe fețele $x = 0, x = 1, y = 0$ (pentru modelul cu ambele tipuri de noduri)
 - Dirichlet pe toate cele 6 fețe (pentru celălalt model)

5.6 Vizualizarea soluțiilor

Așa cum am menționat la începutul capitolului, vom compara soluțiile Neumann + Dirichlet și cea strict Dirichlet

5.7 Doar Dirichlet

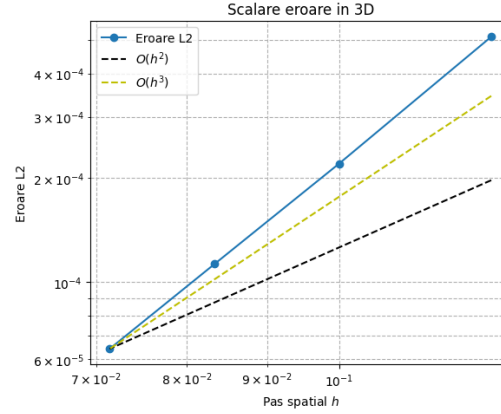


Figura 11: Convergența în cazul Dirichlet

Folosirea exclusivă a condițiilor Dirichlet oferă o stabilitate foarte bună, observabilă prin comportamentul rectiliniu al graficului log-log.

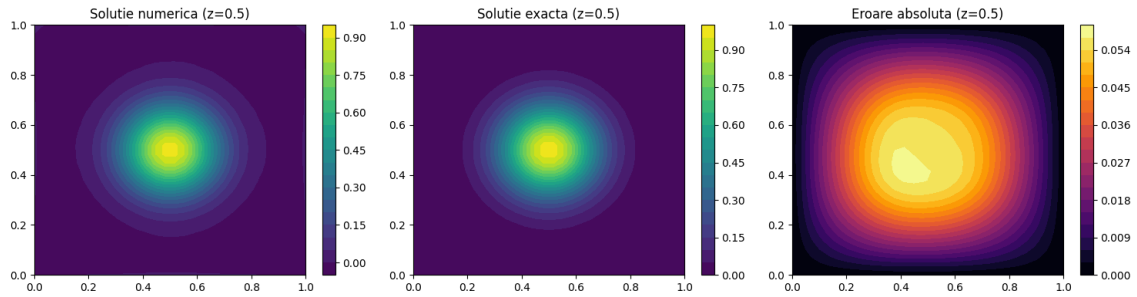
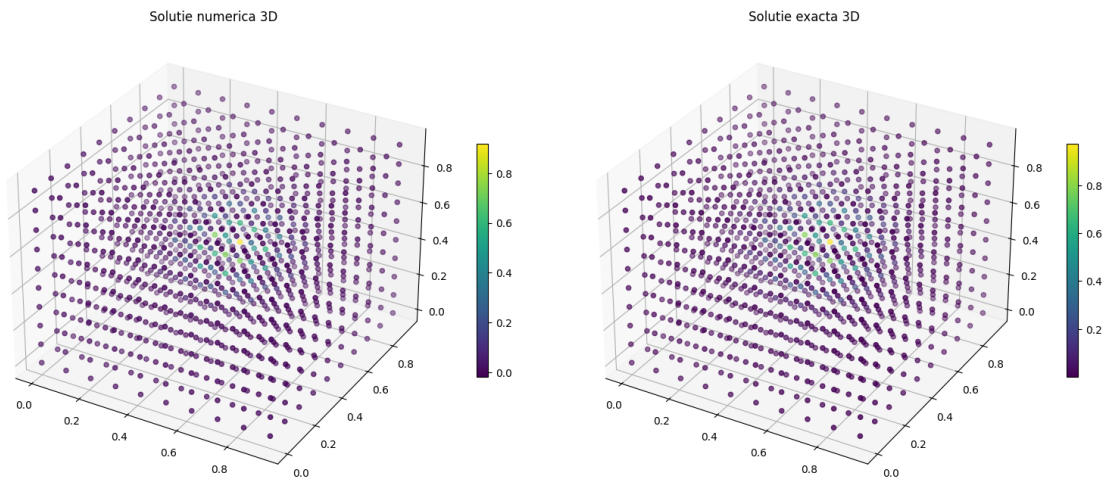


Figura 12: Secțiune plană prin soluția 3D (Dirichlet)



(a) Soluția numerică

(b) Soluția exactă

Figura 13: Comparație între soluția numerică și cea exactă (Dirichlet)

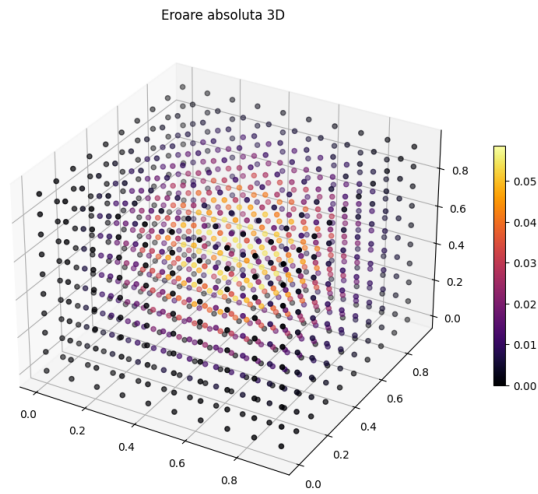


Figura 14: Eroarea absolută pentru Dirichlet

5.8 Neumann+Dirichlet

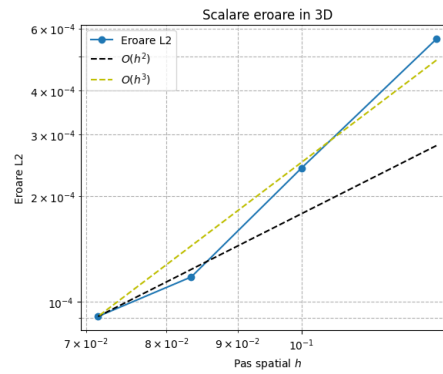


Figura 15: Grafic log-log al convergenței (Neumann + Dirichlet)

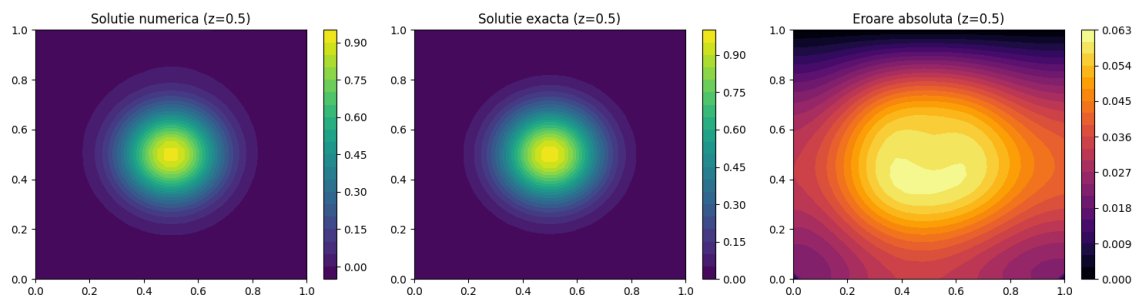


Figura 16: Secțiune plană din soluția 3D (Neumann + Dirichlet)

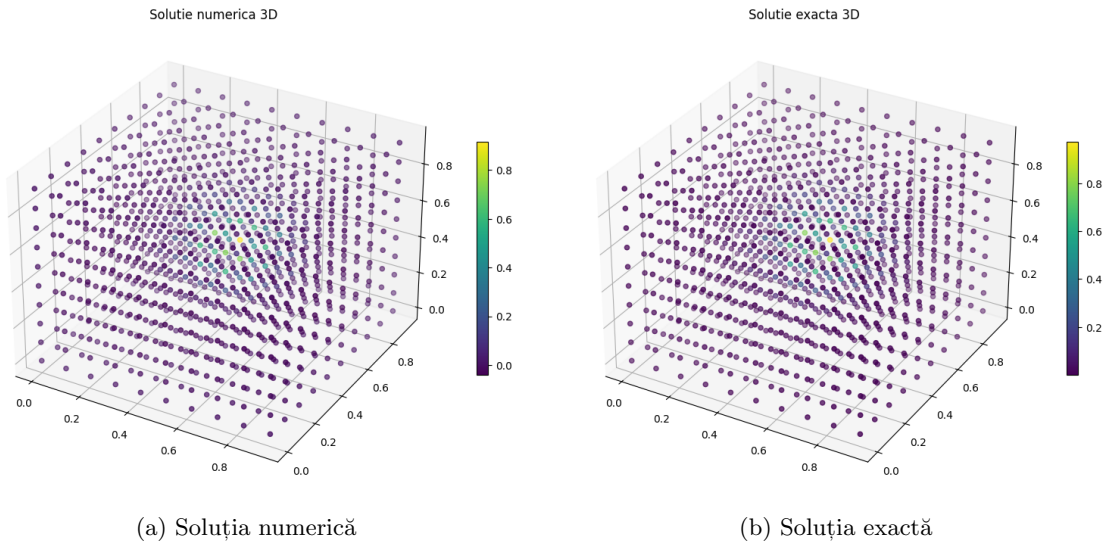


Figura 17: Comparație între soluția numerică și cea exactă (Neumann + Dirichlet)

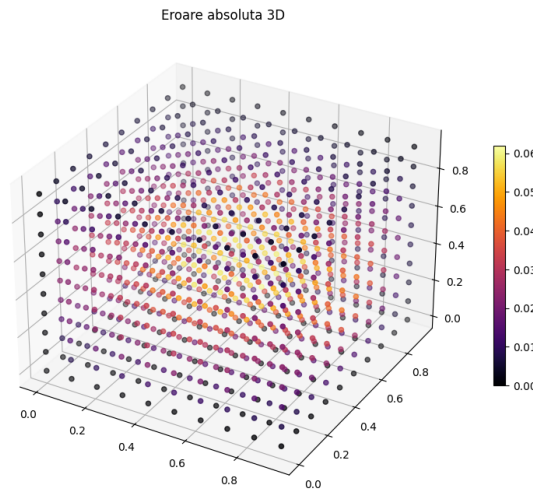


Figura 18: Eroarea absolută (Neumann + Dirichlet)

5.9 Comparația dintre cele doua metode

În graficul convergenței erorii generat de algoritmul unde am folosit doar noduri Dirichlet se observă un comportament predictibil și constant. În timp ce convergența erorii algoritmului Neumann + Dirichlet nu au același grad de siguranță a scalării la modele mai mari.

5.10 Mutarea sursei de căldură

Pentru a muta în sistemul de coordonate carteziane, va trebui să modific parametrii lui u_{sursa} și ai lui dz din funcția f_{sursa}

5.10.1 Mutarea la $z = 0.7$

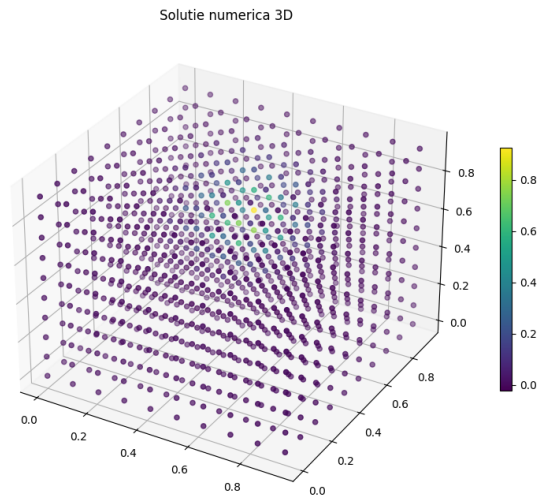


Figura 19: Sursa la $z = 0.7$

Se observă cu ușurință această mutare, mai ales cand ne îndreptăm atenția către nodurile ce formeaza planul de sus al cubului.

5.10.2 Mutarea z la 0.9

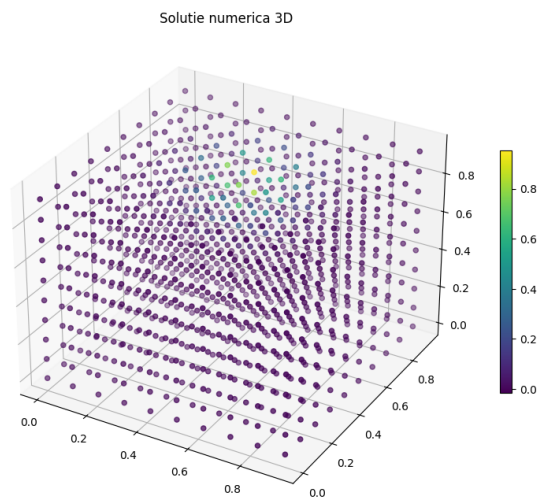


Figura 20: Sursa la $z = 0.9$

Efectul este amplificat cu atât mai mult când mutăm această sursă de căldură la aceste coordonate.

6 Concluzii

Acest proiect a urmărit modelarea procesului de difuzie a căldurii într-un microprocesor folosind ecuații diferențiale parțiale și metode numerice de diferențe finite. Începând de la un model simplificat unidimensional, am avansat treptat către modele bidimensionale și tridimensionale, crescând astfel fidelitatea simulărilor față de realitatea fizică a unui microprocesor modern.

Am demonstrat că metoda diferențelor finite oferă o aproximare stabilă și convergentă, confirmată de graficele log-log ale erorii, care au evidențiat o convergență de ordin $O(h^2)$ sau chiar $O(h^3)$ în versiunea optimizată. Introducerea nodurilor fictive și a medierii coeficienților la interfețe a permis o creștere semnificativă a preciziei numerice, în special în tratarea condițiilor Neumann.

Extinderea la 3D ne-a permis o analiză calitativă a impactului poziționării sursei de căldură în interiorul cipului, o observație esențială pentru ingineria termică a microprocesoarelor moderne. Compararea soluțiilor cu condiții Dirichlet față de cele mixte a evidențiat diferențe semnificative în comportamentul erorii și stabilitatea numerică.

În ansamblu, acest studiu confirmă potențialul metodelor numerice în analiza și optimizarea distribuției termice în microprocesoare, cu aplicații directe în proiectarea sistemelor de răcire și creșterea durabilității componentelor electronice.

7 Bibliografie

https://en.wikipedia.org/wiki/Intel_8086

<https://www.youtube.com/watch?v=Expdq4yX2OI>

https://matplotlib.org/stable/plot_types/index

<https://numpy.org/>