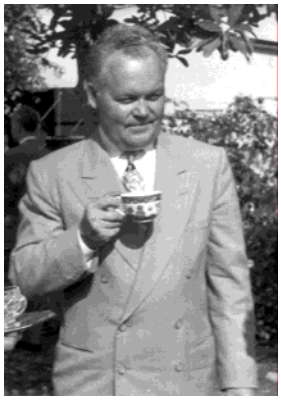
2. METODA GRADIENTULUI CONJUGAT PENTRU REZOLVAREA SISTEMELOR ALGEBRICE LINIARE

Neculai Andrei
Research Institute for Informatics,
Center for Advanced Modeling and Optimization,
8-10, Averescu Avenue, Bucharest 1, Romania,
E-mail: nandrei@ici.ro
Academy of Romanian Scientists
54, Splaiul Independentei, Bucharest 5, Romania

În acest capitol descriem metoda gradientului conjugat pentru rezolvarea sistemelor de ecuaţii algebrice liniare. Această metodă a fost prezentată pentru prima dată de Magnus Hestenes de la Instututul pentru Analiză Numerică a Laboratorului Naţional de Matematică Aplicată – Los Angeles şi de Eduard Stiefel de la Universitatea Tehnică din Zürich.

La început, algoritmii corespunzători acestei metode au fost consideraţi ca tehnici directe de rezolvare a sistemelor liniare, dar repede au fost abandonaţi în urma comparaţiilor numerice cu metodele directe bazate pe eliminarea Gauss. S-a observat că într-o aritmetică exactă, datorită pierderii rapide a ortogonalităţii vectorilor generaţi de algoritm, acesta nu se termină în n paşi, unde n este dimensiunea sistemului. Ca atare, cercetările asupra acestor algoritmi au stagnat aproape două decenii. Situaţia s-a schimbat în anii 1970 când s-a observat că pierderea ortogonalităţii nu eludează convergenţa acestor algoritmi. Această contribuţie foarte importantă a fost adusă de Reid [1971] care a demonstrat foarte multe aspecte esenţiale ale acestor metode arătând potenţialul lor ca metode iterative de rezolvare a sistemelor de ecuaţii algebrice liniare de mari dimensiuni cu matricea coeficienţilor rară. Lucrarea lui Reid a stat la baza a foarte multe dezvoltări a metodelor de gradient conjugat ca metode iterative. În [Andrei, 2000] am prezentat principalele rezultate ale lui Reid şi nu le mai discutăm aici. Contribuţii importante au fost aduse, între alţii, de: Paige [1971, 1972, 1976] care a precizat algoritmi stabili ce nu cer o reortogonalizare completă. Una dintre primele extensii stabile ale algoritmilor de gradient


Magnus Hestenes (1906-1991)

Eduard Stiefel (1909-1978)

conjugat la matrice nedefinite a fost dată de Paige şi Saunders [1975]. Algoritmul lui Paige şi Saunders constituie o realizare majoră, în acest sens nefiind încă depăşit din punctul de vedere al proprietăţilor numerice.

John K. Reid

Imediat Concus şi Golub [1976] au considerat cazul matricelor nesimetrice. O altă extensie majoră a constituit-o introducerea tehnicilor de precondiţionare, care deşi discutate şi analizate încă din anii 1960, doar începând cu anii 1975 au primit o atenţie deosebită mai ales în contextul algoritmilor de gradient conjugat. Lucrarea lui Faber şi Manteuffel [1984], precum şi cea a lui Voevodin [1983] au arătat posibilităţile metodelor de gradient conjugat definite prin formule de recurenţă cu număr mic de termeni. Matematic, algoritmii de gradient conjugat sunt echivalenţi, în sensul că ei realizează acelaşi proces de proiecţie, dar cu implementări diferite.

De fapt, aceasta a fost contribuţia de bază a lui Reid în ceea ce priveşte reabilitarea acestor metode. De-a lungul timpului aceşti algoritmi au fost îndelung rafinaţi, astfel devenind un instrument fundamental pentru rezolvarea unei largi varietăţi de probleme, incluzând aici rezolvarea sistemelor algebrice neliniare şi a problemelor de optimizare cu sau fără restricţii [Andrei, 2008]. În esenţă, metodele de gradient conjugat sunt realizări ale tehnicilor de proiecţie ortogonală pe subspaţii Krylov. Ca atare, aceştia se scufundă în clasa deosebit de bogată a metodelor iterative de rezolvare a sistemelor algebrice liniare [Hackbusch, 1994], [Saad, 1996], [Meurant, 1999], [Barrett *et all*, 1994], [Dongara, Duff, Sorensen şi van der Vorst, 1998].

2.1. METODA PASULUI DESCENDENT

Să considerăm sistemul de ecuaţii algebrice liniare

$$Ax = b, \tag{2.1.1}$$

unde A este o matrice simetrică şi pozitiv definită, precum şi funcţionala

$$\Phi(x) = \frac{1}{2} x^T Ax - x^T b, \tag{2.1.2}$$

care este strict convexă. Atunci, există o soluţie unică x^* a problemei

$$min\ \Phi(x). \tag{2.1.3}$$

Deoarece $\nabla\Phi(x) = Ax - b,$ rezultă că problema de minimizare (2.1.3) este echivalentă cu rezolvarea sistemului de ecuaţii algebrice (2.1.1). Pentru rezolvarea problemei (2.1.3) considerăm următoarea metodă iterativă. Fie x_0 o aproximaţie a soluţiei lui (2.1.3) şi presupunem că *reziduul* $r_k = b - Ax_k \neq 0.$ Alegem o direcţie de căutare $d_k \neq 0$ şi determinăm

$$\min_{\alpha \in R} \Phi(x_k + \alpha d_k). \tag{2.1.4}$$

Cu acestea avem

$$\Psi(\alpha) = \Phi(x_k + \alpha d_k) = \frac{1}{2}\alpha^2 d_k^T A d_k - \alpha d_k^T r_k + \Phi(x_k). \tag{2.1.5}$$

Având în vedere că A este pozitiv definită, atunci $d_k^T A d_k > 0$. Deci, $\Psi(\alpha)$ are un minim unic obținut din condiția necesară de optimalitate

$$\Psi'(\alpha) = \alpha d_k^T A d_k - d_k^T r_k = 0,$$

adică

$$\alpha_k = \frac{d_k^T r_k}{d_k^T A d_k}. \tag{2.1.6}$$

Cu acestea, noua aproximație a punctului de minim x^* al lui (2.1.3) este

$$x_{k+1} = x_k + \alpha_k d_k, \tag{2.1.7}$$

unde α_k este dat de (2.1.6). Desigur, dacă $r_{k+1} = b - Ax_{k+1} \neq 0$, atunci putem repeta acest proces de căutare liniară utilizând noua direcție de căutare d_{k+1}.

Pentru a vedea eficiența procesului de mai sus să comparăm reducerea funcției Φ când facem un pas de la x_k la x_{k+1}. Din (2.1.5) şi (2.1.6) obținem

$$\begin{aligned}\Phi(x_k) - \Phi(x_{k+1}) &= \Phi(x_k) - \Phi(x_k + \alpha_k d_k) = \Phi(x_k) - \Psi(\alpha_k) \\ &= -\frac{1}{2}\alpha_k^2 d_k^T A d_k + \alpha_k d_k^T r_k \\ &= \frac{1}{2}\frac{\left(d_k^T r_k\right)^2}{d_k^T A d_k} > 0.\end{aligned} \tag{2.1.8}$$

Deci funcția Φ îşi reduce valoarea de-a lungul liniei de mai sus dacă şi numai dacă reziduul r_k nu este ortogonal pe direcția de deplasare d_k. Pentru a completa metoda şi deci pentru a avea un algoritm de minimizare a lui (2.1.2) trebuie să precizăm o cale de selecție a direcției d_k la pasul k. După cum ştim, dată aproximarea x_k, atunci cea mai rapidă reducere a valorii funcției Φ este în direcția cu semn schimbat a gradientului funcției Φ in punctul x_k. Aceasta este chiar metoda *pasului descendent* [Cauchy, 1847]. Deci, alegerea optimă locală a direcției de deplasare este

$$d_k = -\nabla\Phi(x_k) = b - Ax_k = r_k. \tag{2.1.9}$$

Pentru funcții pătratice formula iterativă este următoarea:

$$x_{k+1} = x_k + \frac{r_k^T r_k}{r_k^T A r_k} r_k. \tag{2.1.10}$$

Cu acestea avem următorul algoritm de pas descendent.

Algoritmul 2.1.1. (Pasul descendent pentru sisteme liniare)

Pasul 1.	Iniţializare. Selectăm punctul iniţial x_0 şi punem $r_0 = b - Ax_0$ şi $k = 0$.
Pasul 2.	Se testează un criteriu de oprire a iteraţiilor. De exemplu dacă reziduul $\lvert r_k \rvert \le \varepsilon$, unde $\varepsilon > 0$ este o toleranţă suficient de mică, atunci stop; altfel se continuă cu pasul 3.
Pasul 3.	Se calculează: $v_k = Ar_k$, $\alpha_k = (r_k^T r_k)/(r_k^T v_k)$.
Pasul 4.	Se actualizează variabilele şi reziduul: $x_{k+1} = x_k + \alpha_k r_k$, şi $r_{k+1} = r_k - \alpha_k v_k$. Se pune $k = k+1$ şi se continuă cu pasul 2. ♦

Câteva observaţii sunt necesare:

1) La fiecare iteraţie a algoritmului, în pasul 3, sunt necesare următoarele operaţii algebrice: un produs matrice vector şi două produse scalare.
2) În pasul 4 reziduul r_{k+1} se calculează într-o manieră recursivă utilizând reziduul de la pasul k.
3) În criteriul de oprire a iteraţiilor uzual $\varepsilon = 10^{-6}$.

Observăm că pentru soluţia x^* a lui (2.1.1), pentru fiecare $x \in R^n$, putem scrie:

$$(x - x^*)^T A(x - x^*) = 2(\Phi(x) - \Phi(x^*)).$$

Deci, $\tilde{x}$ este un punct de minim al lui Φ dacă şi numai dacă acesta minimizează norma

$$\left\| x^* - \tilde{x} \right\|_A = \sqrt{(x^* - \tilde{x})^T A(x^* - \tilde{x})} \tag{2.1.11}$$

erorii corespunzătoare lui $\tilde{x}$.

Teorema 2.1.1. *Fie A o $n \times n -$matrice simetrică şi pozitiv definită cu valorile proprii $\lambda_1 \le \lambda_2 \le \ldots \le \lambda_n$. Pentru fiecare punct iniţial x_0 şirul x_k generat de algoritmul pasului descendent converge la soluţia x^* a lui (2.1.1) şi*

$$\left\| x_k - x^* \right\|_A \le \left(\frac{\lambda_n - \lambda_1}{\lambda_n + \lambda_1} \right)^k \left\| x_0 - x^* \right\|_A . \tag{2.1.12}$$

Demonstraţie. Considerăm un punct arbitrar $x \in R^n$ şi reziduul corespunzător $r = b - Ax$. Atunci pasul considerat de algoritmul pasului descendent este:

$$\begin{aligned} y = x + \frac{\|r\|_2^2}{r^T Ar}\left(b - Ax\right) &= x + \frac{\|r\|_2^2}{r^T Ar} b - \frac{\|r\|_2^2}{r^T Ar} Ax \\ &= \left(I - t(x)A\right)x + t(x)b, \end{aligned}$$

unde $t(x) = \|r\|_2^2 / (r^T Ar)$. Deci, eroarea este:

$$\begin{aligned} y - x^* &= \left(I - t(x)A\right)x + t(x)b - x^* \\ &= \left(I - t(x)A\right)x + t(x)Ax^* - x^* \\ &= \left(I - t(x)A\right)\left(x - x^*\right). \end{aligned}$$

Fie $z(\sigma) = \left(I - \sigma A\right)x + \sigma b,$ unde $\sigma \in R.$ Atunci, pentru fiecare $\sigma \in R$ avem

$$\left\|y - x^*\right\|_A \le \left\|z(\sigma) - x^*\right\|_A .$$

Demonstrăm că pentru $\sigma = \tilde{\sigma} \equiv 2/(\lambda_n + \lambda_1)$ următoarea inegalitate este satisfăcută

$$\left\|z(\tilde{\sigma}) - x^*\right\|_A \le \frac{\lambda_n - \lambda_1}{\lambda_n + \lambda_1}\left\|x - x^*\right\|_A$$

Într-adevăr, să considerăm erorile $e = x - x^*$ şi $\tilde{e} = z(\tilde{\sigma}) - x^*$, atunci

$$\begin{aligned} \tilde{e} = z(\tilde{\sigma}) - x^* &= \left(I - \tilde{\sigma}A\right)x + \tilde{\sigma}b - x^* \\ &= \left(I - \tilde{\sigma}A\right)x + \tilde{\sigma}Ax^* - x^* = \left(I - \tilde{\sigma}A\right)e. \end{aligned}$$

Observăm că matricea $I - \tilde{\sigma}A$ este simetrică cu valorile proprii $1 - \tilde{\sigma}\lambda_j$. Deci norma ei spectrală este dată de:

$$\left\|I - \tilde{\sigma}A\right\|_2 = max\left\{\left|1 - \frac{2\lambda_1}{\lambda_n + \lambda_1}\right|, \left|1 - \frac{2\lambda_n}{\lambda_n + \lambda_1}\right|\right\} = \frac{\lambda_n - \lambda_1}{\lambda_n + \lambda_1}.$$

Cu acestea obţinem:

$$\begin{aligned} \left\|y - x^*\right\|_A \le \left\|\tilde{e}\right\|_A &= \left\|A^{1/2}\tilde{e}\right\|_2 = \left\|A^{1/2}(I - \tilde{\sigma}A)e\right\|_2 \\ &= \left\|(I - \tilde{\sigma}A)A^{1/2}e\right\|_2 \le \frac{\lambda_n - \lambda_1}{\lambda_n + \lambda_1}\left\|A^{1/2}e\right\|_2 = \frac{\lambda_n - \lambda_1}{\lambda_n + \lambda_1}\left\|e\right\|_A , \end{aligned}$$

de unde, prin inducţie, obţinem (2.1.12). ■

Observaţia 2.1.1. Dacă λ_1 şi λ_n sunt valorile proprii ale lui A ca în teorema 8.1 şi $\kappa(A) = \lambda_n / \lambda_1$ este numărul de condiţionare al lui A definit în raport cu norma Euclidiană, atunci algoritmul pasului descendent este caracterizat de următorul factor de reducere a erorii:

$$\left\|x_k - x^*\right\|_A \le \eta\left\|x_{k-1} - x^*\right\|_A , \tag{2.1.13}$$

unde

$$\eta = \frac{\lambda_n - \lambda_1}{\lambda_n + \lambda_1} = \frac{\kappa(A) - 1}{\kappa(A) + 1}. \tag{2.1.14}$$

Deci, cu cât numărul de condiţionare al lui A este mai mic, cu atât algoritmul pasului descendent este mai rapid. În particular, când $\kappa(A) = 1,$ atunci $\eta = 0$ ceea ce înseamnă că algoritmul găseşte minimul într-un singur pas. ♦

Exemplul 2.1.1. Considerăm

$$\Phi(x) = \frac{1}{2} x^T \begin{bmatrix} 1 & 0 \\ 0 & \gamma \end{bmatrix} x = \frac{1}{2}\left(x_1^2 + \gamma x_2^2\right),$$

unde $\gamma > 0$. Punctul de minim este $x^* = 0$ şi $\Phi(x^*) = 0$. Valorile proprii ale lui A sunt 1 şi γ. Aplicăm algoritmul pasului descendent din punctul inițial $x_0 = (\lambda, 1)$. În acest caz putem obține soluția analitică a iterațiilor ca:

$$x_{1(k)} = \gamma\left(\frac{\gamma-1}{\gamma+1}\right)^k, \qquad x_{2(k)} = \left(-\frac{\gamma-1}{\gamma+1}\right)^k,$$

cu care putem calcula valorile funcției:

$$\Phi(x_k) = \frac{\gamma(\gamma+1)}{2}\left(\frac{\gamma-1}{\gamma+1}\right)^{2k} = \left(\frac{\gamma-1}{\gamma+1}\right)^{2k} \Phi(x_0).$$

Pentru acest exemplu vedem că convergența este liniară, adică la fiecare iterație eroarea evoluează ca o serie geometrică, aceasta fiind redusă cu factorul $(\gamma-1)^2/(\gamma+1)^2$. Pentru $\gamma = 1$ soluția exactă se obține într-o singură iterație. Dacă γ este într-o vecinătate a lui 1, atunci convergența este rapidă. Pe de altă parte, convergența este foarte lentă pentru for $\gamma \gg 1$ sau $\gamma \ll 1$. Tabelul 2.1.1 arată numărul de iterații (#iter) necesare pentru obținerea unei soluții, pentru diverse valori ale lui γ, pentru care $\|r_k\|_2 \le 10^{-9}$.

Tabelul 2.1.1. Pasul descendent pentru exemplul 2.1.1.
Numărul de iterații.

γ	#iter	γ	# iter
10	117	10^{-1}	94
10^2	1284	10^{-2}	824
10^3	13989	10^{-3}	7082
10^4	151401	10^{-4}	59298

Performanța algoritmului pasului descendent pentru probleme prost condiționate se explică prin faptul că curbele de nivel constant ale lui Φ sunt elipse foarte deformate. Metoda merge în zigzag de-a lungul unei văi înguste de tipul $\left\{x \in R^2 : \Phi(x) \le \Phi(x_0)\right\}$. Deşi minimul este situat în direcția x_1, de-a lungul iterațiilor algoritmul comută de la direcția în x_1 la direcția în x_2, aceasta reducând puternic viteza de convergență a algoritmului. Pentru probleme prost condiționate, aceasta este comportarea tipică a algoritmului pasului descendent.

2.2. METODA DIRECȚIILOR CONJUGATE

După cum am văzut, pentru evitarea comportării în zigzag a algoritmului pasului descendent trebuie să considerăm o altă metodă de calcul a direcției de deplasare. Notăm cu $ls\{d_1,\ldots,d_k\}$ subspațiul liniar generat de vectorii $d_1,\ldots,d_n$. Cu acestea ne punem problema unei căutări ideale cu următoarea proprietate:

> *„Dacă aproximațiile $x_1,x_2,\ldots,x_k$ sunt determinate prin căutare liniară exactă de-a lungul direcțiilor $d_1,d_2,\ldots,d_k$, atunci punctul de minim x_{k+1} al funcției Φ de-a lungul liniei $x_k+\alpha d_k$, $\alpha\in R$, trebuie să minimizeze Φ pe subspațiul afin $x_0+ls\{d_1,\ldots,d_k\}$ care este generat de toate direcțiile anterioare $d_1,d_2,\ldots,d_k$."*

Deci, dacă suntem capabili să generăm direcții liniar independente care satisfac această proprietate, atunci metoda respectivă nu este numai convergentă, după cum am văzut mai sus, ci este şi finită, deoarece după cel mult n paşi ea găseşte punctul de minim al lui Φ pe R^n, întru-cât $x_0+ls\{d_1,\ldots,d_n\}=R^n$. Zicem că metoda are *proprietatea de terminare finită*.

Definiția 2.2.1. *Fie $A\in R^{n\times n}$ o matrice simetrică şi pozitiv definită. Mulțimea de vectori $d_1,d_2,\ldots,d_k\in R^n$ se numeşte $A-$conjugată dacă $d_j\neq 0$ pentru toți $j=1,\ldots,k$ şi $d_j^T Ad_i\neq 0$ pentru toți $i,j=1,\ldots,k$ şi $i\neq j$.*

Teorema 2.2.1. *Fie $x_0\in R^n$ o aproximație inițială a lui x^* şi $x_1,x_2,\ldots$ iterațiile determinate de o procedură de căutare liniară pentru minimizarea lui (2.1.2), unde direcțiile $d_1,d_2,\ldots$ sunt mutual $A-$conjugate. Atunci x_{k+1} minimizează funcția Φ pe subspațiul afin $x_0+ls\{d_1,\ldots,d_k\}$.*

Demonstrație. Considerăm un punct $x\in x_0+ls\{d_1,\ldots,d_k\}$, atunci x are următoarea reprezentare $x=\tilde{x}+\alpha d_k$, unde $\alpha\in R$ şi

$$\tilde{x}=x_0+\sum_{j=1}^{k}\xi_j d_j.$$

Cu acestea avem:

$$\Phi(x)=\Phi(\tilde{x})+\alpha\sum_{j=1}^{k}\xi_j d_j^T Ad_k-\alpha d_k^T r_0+\frac{1}{2}\alpha^2 d_k^T Ad_k.$$

Dacă $d_j^T Ad_k=0$ pentru toți $j=1,\ldots,k$, atunci

$$\Phi(x) = \Phi(\tilde{x}) + \frac{1}{2}\alpha^2 d_k^T A d_k - \alpha d_k^T r_0$$

şi problema de minimizare a lui Φ pe $x_0 + ls\{d_1, \ldots, d_k\}$ este *decuplată* în două sub-probleme de minimizare. Prima constă în minimizarea funcţiei Φ pe $x_0 + ls\{d_1, \ldots, d_k\}$ şi a doua în minimizarea funcţiei

$$\Psi(\alpha) = \frac{1}{2}\alpha^2 d_k^T A d_k - \alpha d_k^T r_0,$$

care, după cum se vede, este independentă de $\tilde{x}$, dar depinde numai de direcţia de căutare curentă d_k.

Minimul α_k al lui Ψ este

$$\alpha_k = \frac{d_k^T r_0}{d_k^T A d_k}.$$

Dar,

$$\begin{aligned} d_k^T r_k &= d_k^T \left(b - A\left(x_0 + \sum_{j=1}^{k} \alpha_j d_j \right) \right) \\ &= d_k^T \left(b - Ax_0 - \sum_{j=1}^{k} \alpha_j A d_j \right) \\ &= d_k^T r_0 - \sum_{j=1}^{k} \alpha_j d_k^T A d_j = d_k^T r_0. \end{aligned}$$

Deci,

$$\alpha_k = \frac{d_k^T r_k}{d_k^T A d_k}. \ \blacksquare$$

Dacă direcţiile d_j sunt mutual conjugate, adică $d_i^T A d_j = 0$, pentru $i \neq j$, atunci minimul lui Φ pe subspaţiul afin $x_0 + ls\{d_1, \ldots, d_k\}$ poate fi înlocuit prin k căutări liniare consecutive de-a lungul direcţiilor $d_1, d_2, \ldots, d_k$. Problema acum s-a mutat la aceea a determinării unui set de direcţii mutual conjugate. De modul cum se implementează această idee a conjugării depinde tipul de algoritm de gradient conjugat. Deci, pentru a rezolva sistemul algebric liniar $Ax = b$, se poate prezenta următorul algoritm de direcţii conjugate.

Algoritmul 2.2.1. (Direcţii conjugate pentru sisteme liniare)

Pasul 1.	Iniţializare. Selectăm punctul inițial x_0 şi punem $r_0 = b - Ax_0$ şi $k = 0$.
Pasul 2.	Se testează un criteriu de oprire a iteraţiilor. De exemplu dacă

	reziduul $\lvert r_k \rvert \le \varepsilon$, unde $\varepsilon > 0$ este o toleranță suficient de mică, atunci stop; altfel se continuă cu pasul 3.
Pasul 3.	Se alege d_k astfel încât $d_k^T A d_j = 0$, pentru $j = 0, \ldots, k-1$ şi $d_k^T r_k \ne 0$.
Pasul 4.	Se calculează: $\alpha_k = (d_k^T r_k)/(d_k^T A d_k)$.
Pasul 5.	Se actualizează variabilele şi reziduul: $x_{k+1} = x_k + \alpha_k d_k$, $r_{k+1} = r_k - A x_{k+1}$. Se pune $k = k+1$ şi se continuă cu pasul 2. ♦

2.3. METODA GRADIENTULUI CONJUGAT

După cum se cunoaşte, doi vectori $x, y \in R^n$ sunt $A-$conjugați dacă şi numai dacă sunt ortogonali în raport cu produsul scalar $\langle x, y \rangle_A = y^T A x$. Deci, dacă $d_1, d_2, \ldots, d_k$ sunt $A-$conjugați, atunci aceştia sunt liniar independenți. Dată o mulțime de vectori liniar independenți $v_1, \ldots, v_n \in R^n$, atunci se poate construi o mulțime de n vectori $A-$conjugați $v_1, \ldots, v_n$ prin utilizarea algoritmului de ortogonalizare Gram-Schmidt în raport cu produsul scalar $\langle .,. \rangle_A$.

Pentru a implementa algoritmul de direcții conjugate trebuie să precizăm o procedură de selecție a direcțiilor d_k astfel încât $d_k A d_j = 0$, pentru toți $j = 0, \ldots, k-1$ şi $d_k^T r_k \ne 0$. Prima condiție este satisfăcută de complementul $A-$ortogonal a oricărui vector $v \notin ls\{d_1, \ldots, d_k\}$ în raport cu spațiul $ls\{d_1, \ldots, d_k\}$. Pentru a realiza a doua cerință selectăm $v = r_k$, adică

$$d_k = r_k - \sum_{j=1}^{k-1} \frac{d_j^T A r_k}{d_j^T A d_j} d_j. \tag{2.3.1}$$

Deci,

$$d_k^T r_k = r_k^T r_k - \sum_{j=1}^{k-1} \frac{d_j^T A r_k}{d_j^T A d_j} d_j^T r_k = r_k^T r_k > 0,$$

ceea ce înseamnă că d_k este o direcție care satisface a doua cerință de mai sus. Următoarea propoziție arată că suma din (2.3.1) se reduce la ultimul termen.

Propoziția 2.3.1. *Fie $x_0 \in R^n$ dat şi presupunem că $x_1, \ldots, x_k$ sunt generați de algoritmul de direcții conjugate, unde direcțiile sunt date de (2.3.1). Atunci vectorii r_j şi d_j satisfac:*

$$r_k^T r_j = 0, \qquad j = 0, \ldots, k-1, \tag{2.3.2}$$

$$d_j^T A r_k = 0, \quad j = 1, \ldots, k-1. \tag{2.3.3}$$

Demonstrație. Din (2.3.1) obținem

$$r_j = d_j + \sum_{i=1}^{j-1} \frac{d_i^T A r_j}{d_i^T A d_i} d_i.$$

Acum,

$$r_k^T r_j = r_k^T d_j + \sum_{i=1}^{j-1} \frac{d_i^T A r_j}{d_i^T A d_i} r_k^T d_i.$$

Dar $r_k^T d_i = 0$, pentru toți $i = 1, \ldots, j,$ adică (2.3.2) este îndeplinită. Pentru a demonstra (2.3.3), pentru $j = 1, \ldots, k-1,$ putem scrie:

$$d_{j+1} = r_{j+1} - \sum_{i=1}^{j} \frac{d_i^T A r_{j+1}}{d_i^T A d_i} d_i = b - A x_{j+1} - \sum_{i=1}^{j} \frac{d_i^T A r_{j+1}}{d_i^T A d_i} d_i$$

$$= b - A\left(x_j + \alpha_j d_j\right) - \sum_{i=1}^{j} \frac{d_i^T A r_{j+1}}{d_i^T A d_i} d_i.$$

Deoarece $\alpha_j \neq 0,$ rezultă că

$$A d_j = \frac{1}{\alpha_j}\left(r_j - d_{j+1} - \sum_{i=1}^{j} \frac{d_i^T A r_{j+1}}{d_i^T A d_i} d_i \right)$$

şi pentru $j = 1, \ldots, k-1$

$$r_k^T A d_j = \frac{1}{\alpha_j}\left(r_k^T r_j - r_k^T d_{j+1} - \sum_{i=1}^{j} \frac{d_i^T A r_{j+1}}{d_i^T A d_i} r_k^T d_i \right) = 0,$$

ceea ce arată că (2.3.3) are loc. ■

Ca atare, din (2.3.1), utilizând propoziția 2.3.1, rezultă că

$$d_k = r_k - \frac{d_{k-1}^T A r_k}{d_{k-1}^T A d_{k-1}} d_{k-1} \triangleq r_k + \beta_k d_{k-1}. \tag{2.3.4}$$

Concluzia care rezultă din aceste dezvoltări este că în acest proces *nu este necesar să memorăm toate direcțiile de deplasare de-a lungul iterațiilor* pentru a executa $A-$ortogonalizarea, ci numai cea mai recentă direcție.

Observația 2.3.1. Este posibil să îmbunătățim eficiența algoritmului observând că

$$r_{k+1} = b - A x_{k+1} = b - A(x_k + \alpha_k d_k) = r_k - \alpha_k A d_k. \tag{2.3.5}$$

La fiecare iterație aceasta salvează o înmulțire matrice-vector $A x_{k+1}$, deoarece produsul $A d_k$ este deja calculat în pasul 4 al algoritmului de direcții conjugate. ♦

Cu acestea, pentru $k \geq 2$, din (2.3.5) obținem:

$$Ad_{k-1} = \frac{1}{\alpha_{k-1}}(r_{k-1} - r_k).$$

Ortogonalitatea reziduurilor ne conduce la

$$d_{k-1}^T A r_k = \frac{1}{\alpha_{k-1}}(r_{k-1} - r_k)^T r_k = -\frac{1}{\alpha_{k-1}} r_k^T r_k.$$

Pe de altă parte

$$\begin{aligned} d_{k-1}^T A d_{k-1} &= d_{k-1}^T A(r_{k-1} + \beta_{k-1} d_{k-2}) = d_{k-1}^T A r_{k-1} \\ &= \frac{1}{\alpha_{k-1}}(r_{k-1} - r_k)^T r_{k-1} = \frac{1}{\alpha_{k-1}} r_{k-1}^T r_{k-1}. \end{aligned}$$

Deci, din (2.3.4) obținem următoarea expresie a direcției:

$$d_k = r_k + \frac{r_k^T r_k}{r_{k-1}^T r_{k-1}} d_{k-1} \equiv r_k + \beta_k d_{k-1}. \tag{2.3.6}$$

În plus, este posibil să obținem o nouă expresie algebrică a lungimii pasului α_k. Într-adevăr

$$d_k^T r_k = (r_k + \beta_k d_{k-1})^T r_k = r_k^T r_k,$$

care a fost deja calculată în (2.3.6).

Cu aceste manipulări algebrice, în final, obținem următorul algoritm de gradient conjugat, care în esență este metoda Hestenes and Stiefel [1952].

Algoritmul 2.3.1. (Gradient conjugat pentru sisteme liniare)

Pasul 1.	Inițializare. Selectăm punctul inițial x_0 şi punem $r_0 = b - Ax_0$, $t_0 = r_0^T r_0$, $d_0 = r_0$ şi $k = 0$.
Pasul 2.	Se testează un criteriu de oprire a iterațiilor. De exemplu dacă reziduul $\lvert r_k \rvert \leq \varepsilon$, unde $\varepsilon > 0$ este o toleranță suficient de mică, atunci stop; altfel se continuă cu pasul 3.
Pasul 3.	Se calculează: $s_k = Ad_k$ şi $\alpha_k = t_k / (d_k^T s_k)$.
Pasul 4.	Se actualizează variabilele şi reziduul: $x_{k+1} = x_k + \alpha_k d_k$, $r_{k+1} = r_k - \alpha_k s_k$.
Pasul 5.	Se calculează: $t_{k+1} = r_{k+1}^T r_{k+1}$, $\beta_{k+1} = t_{k+1} / t_k$ şi direcțiile $d_{k+1} = r_{k+1} + \beta_{k+1} d_k$. Se pune $k = k+1$ şi se continuă cu pasul 2. ♦

Cunoaştem că pentru orice $x \in R^n$, soluția lui (2.1.1) satisface

$$\left\| x - x^* \right\|_A^2 = 2\left(\Phi(x) - \Phi(x^*)\right) \tag{2.3.7}$$

Deci, estimația x_k minimizează eroarea $e = x - x^*$ în raport cu norma $\|.\|_A$ peste spațiul afin $x_0 + ls\{d_0, \ldots, d_k\}$. Reprezentarea lui $ls\{d_0, \ldots, d_k\}$ este dată de următoarea propoziție care este importantă pentru stabilirea unei margini a erorii furnizată de algoritmul de gradient conjugat.

Definiția 2.3.1. *Fie* $A \in R^{n\times n}$ *şi* $b \in R^n$ *date. Atunci spațiul liniar*

$$K_k(b, A) = ls\{b, Ab, \ldots, A^{k-1}b\}$$

se numeşte $k-$ *spațiul Krylov al lui* A *corespunzător lui* b.

Propoziția 2.3.2. *Presupunem că direcțiile* $d_0, \ldots, d_k$ *sunt definite de algoritmul de gradient conjugat. Atunci*

$$ls\{d_0, \ldots, d_k\} = ls\{r_0, Ar_0, \ldots, A^k r_0\}.$$

Demonstrație. Pentru $k = 0$ propoziția este trivială deoarece $d_0 = r_0$. Presupunem că este adevărată pentru un $k < n$. Atunci

$$\begin{aligned} d_{k+1} &= r_{k+1} + \beta_{k+1} d_k \\ &= r_k - \alpha_k A d_k + \beta_{k+1} d_k \\ &= d_k - \beta_k d_{k-1} - \alpha_k A d_k + \beta_{k+1} d_k. \end{aligned} \tag{2.3.8}$$

Dar, $d_k, d_{k-1} \in K_k(r_0, A)$. Deci

$$ls\{d_0, \ldots d_{k+1}\} \subset K_{k+1}(r_0, A).$$

Incluziunea inversă rezultă imediat din (2.3.8) şi $\alpha_k \neq 0$. ■

Pentru a demonstra convergența algoritmului de gradient conjugat trebuie să stabilim o expresie a erorii. Ca mai sus, notăm $e_0 = x^* - x_0$, eroarea lui x_0. Atunci din $r_0 = b - Ax_0 = A(x^* - x_0) = Ae_0$ rezultă că $A^j r_0 = A^{j+1} e_0$, adică elementele spațiului Krylov $z \in K_k(r_0, A)$ se pot exprima sub forma:

$$z = \sum_{j=1}^{k} t_j A^j e_0, \quad t_j \in R.$$

Deci, eroarea $e_k = x^* - x_k$ de la iterația k satisface:

$$\begin{aligned} \|e_k\|_A^2 &= \min_{x \in x_0 + K_k(r_0, A)} \|x^* - x\|_A^2 \\ &= \min_{z \in K_k(r_0, A)} \|x^* - x_0 + z\|_A^2 \\ &= \min_{t_1, \ldots, t_k \in R} \|e_0 + t_1 A e_0 + \ldots + t_k A^k e_0\|_A^2. \end{aligned} \tag{2.3.9}$$

Acum, să considerăm P_k mulțimea polinoamelor p de grad mai mic sau egal cu k, astfel încât $p(0)=1$. Atunci (2.3.9) se poate exprima ca:

$$\|e_k\|_A^2 = \min_{p\in P_k} \|p(A)e_0\|_A^2, \tag{2.3.10}$$

ceea ce arată că eroarea de la iterația k se poate exprima ca produsul unui polinom $p(A)$ şi eroarea inițială e_0.

Pentru a obține o formă mai convenabilă a erorii vom utiliza descompunerea spectrală a matricei A ca:

$$A = \sum_{j=1}^{n} \lambda_j z_j z_j^T, \tag{2.3.11}$$

unde z_j, $j=1,\ldots,n,$ este o mulțime de vectori proprii ortogonali care corespund valorilor proprii $0<\lambda_1 \le \lambda_2 \le \cdots \le \lambda_n$. Observăm că:

$$A^2 = \sum_{i,j=1}^{n} \lambda_i \lambda_j z_i z_i^T z_j z_j^T = \sum_{j=1}^{n} \lambda_j^2 z_j z_j^T,$$

deoarece $z_i^T z_j = \delta_{ij}$. Prin inducție putem demonstra că:

$$A^i = \sum_{j=1}^{n} \lambda_j^i z_j z_j^T.$$

Ca atare, eroarea e_0 se poate exprima sub forma:

$$e_0 = \sum_{j=1}^{n} \gamma_j z_j. \tag{2.3.12}$$

Având în vedere dezvoltările de mai sus, urmează că eroarea de la iterația k se poate scrie sub forma:

$$\|e_k\|_A^2 = \min_{p\in P_k} \left\| \sum_{j=1}^{n} \gamma_j p(\lambda_j) z_j \right\|_A^2. \tag{2.3.13}$$

Dar

$$\begin{aligned} \left\| \sum_{j=1}^{n} \gamma_j p(\lambda_j) z_j \right\|_A^2 &= \left(\sum_{i=1}^{n} \gamma_i p(\lambda_i) z_i \right)^T A \left(\sum_{j=1}^{n} \gamma_j p(\lambda_j) z_j \right) \\ &= \sum_{j=1}^{n} \gamma_j^2 \lambda_j \left(p(\lambda_j) \right)^2. \end{aligned} \tag{2.3.14}$$

Deci, cu acestea obținem o reprezentare foarte importantă a erorii lui x_k ca:

$$\|e_k\|_A^2 = \min_{p\in P_k} \sum_{j=1}^{n} \gamma_j^2 \lambda_j \left(p(\lambda_j) \right)^2. \tag{2.3.15}$$

Utilizând (2.3.15) putem determina o margine superioară a erorii care ne permite să stabilim proprietățile de convergență ale metodei gradientului conjugat. Într-adevăr din

$$\|e_0\|_A^2 = \sum_{j=1}^{n} \lambda_j \gamma_j^2$$

obținem o estimație a valorilor $|p(\lambda_j)|$ prin maximizarea lui $|p|$ pe spectrul lui A:

$$\|e_k\|_A \le \min_{p \in P_k} \max_{\lambda \in \sigma(A)} |p(\lambda)| \|e_0\|_A . \tag{2.3.16}$$

Dar, constanta

$$\min_{p \in P_k} \max_{\lambda \in \sigma(A)} |p(\lambda)|$$

nu este uşor de evaluat. Totuşi, următoarea margine este utilă

$$\max_{\lambda \in \sigma(A)} |p(\lambda)| \le \max_{\lambda_1 \le \lambda \le \lambda_n} |p(\lambda)|$$

pentru a obține următoarea estimație

$$\min_{p \in P_k} \max_{\lambda \in \sigma(A)} |p(\lambda)| \le \max_{\lambda_1 \le \lambda \le \lambda_n} |\overline{p}_k(\lambda)|,$$

unde $\overline{p}_k(x)$ este un polinom de gradul k, pentru care $\overline{p}_k(0) = 1$ şi $\max_{\lambda_1 \le \lambda \le \lambda_n} |\overline{p}_k(\lambda)|$ este uşor de evaluat. Polinoamele care satisfac aceste cerințe sunt tocmai polinoamele Chebyshev de prima speță. Aceste polinoame sunt definite în intervalul $[-1,1]$ şi au următoarea definiție:

$$T_k(x) = \cos\left(k \arccos(x)\right). \tag{2.3.17}$$

Este uşor de observat că $T_k(x)$ este un polinom dacă utilizăm următoarele identități trigonometrice:

$$\cos(\alpha + \beta) = \cos\alpha \cos\beta - \sin\alpha \sin\beta,$$
$$\cos(\alpha + \beta) + \cos(\alpha - \beta) = 2\cos\alpha \cos\beta.$$

Să notăm $\theta = \arccos(x)$. Atunci obținem:

$$T_0(x) = \cos(0\theta) = 1,$$
$$T_1(x) = \cos(1\theta) = x,$$
$$T_2(x) = \cos(2\theta) = \cos^2\theta - \sin^2\theta = 2\cos^2\theta - 1 = 2x^2 - 1,$$
$$T_{k+1}(x) + T_{k-1}(x) = \cos((k+1)\theta) + \cos((k-1)\theta)$$
$$= 2\cos(k\theta)\cos\theta = 2xT_k(x).$$

În general are loc următoarea relație de recurență:

$$T_0(x) = 1, \ T_1(x) = x,$$
$$T_{k+1}(x) = 2xT_k(x) - T_{k-1}(x).$$

Rezolvând ecuația de recurență de mai sus obținem forma explicită a polinoamelor lui Chebyshev:

$$T_k(x) = \frac{1}{2}\left[\left(x+\sqrt{x^2-1}\right)^k + \left(x-\sqrt{x^2-1}\right)^k\right],$$

care este validă în intervalul $(-\infty,\infty)$. Figurile 2.3.1 – 2.3.4 ilustrează câteva polinoame Chebyshev.

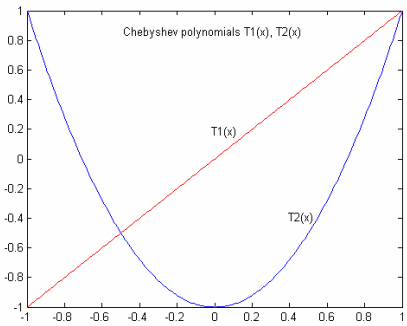


Fig. 2.3.1 Polinoamele Chebyshev $T_1(x), T_2(x)$.

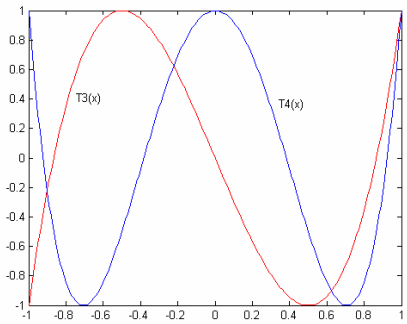


Fig. 2.3.2. Polinoamele Chebyshev $T_3(x), T_4(x)$.

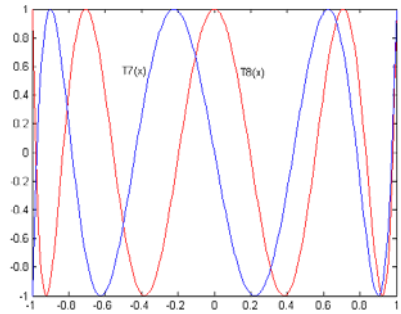


Fig. 2.3.3. Polinoamele Chebyshev $T_7(x), T_8(x)$.

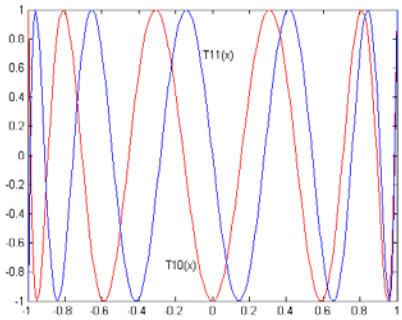


Fig. 2.3.4. Polinoamele Chebyshev $T_{10}(x), T_{11}(x)$.

Polinomul Chebyshev translatat şi scalat se defineşte ca:

$$T_k(x;a,b) = T_k\left(\frac{a+b-2x}{b-a}\right),$$

în care $|T_k(x;a,b)| < 1$ pentru toţi $x \in [a,b]$.

Polinoamele Chebyshev sunt deosebit de utile în studiul convergenţei metodelor iterative. În acest sens, cea mai importantă proprietate a lor este că acestea au cea mai mică normă maximă. După cum am spus, suntem interesaţi în a rezolva următoarea problemă de minimizare:

$$\min_{p\in P_k} \max_{x\in[a,b]} |p(x)|.$$

O soluţie a acestei probleme este dată de polinomul Chebyshev translatat şi scalat:

$$\min_{p\in P_k}\max_{x\in[a,b]}|p(x)| = \max\left|\frac{T_k\left(\dfrac{2x-(a+b)}{b-a}\right)}{T_k\left(\dfrac{a+b}{b-a}\right)}\right| = \frac{1}{\left|T_k\left(\dfrac{a+b}{b-a}\right)\right|}.$$

Dacă $a=-b$, atunci $T_k\left(\dfrac{a+b}{b-a}\right)=1,$ dar în cazul general avem nevoie de o margine superioară a lui $\left|T_k\left(\dfrac{a+b}{b-a}\right)\right|^{-1}$. Pentru a obține o astfel de margine vom utiliza următoarea propoziție:

Propoziția 2.3.3. *Dacă* $0<a<b$, *atunci*

$$\min_{p\in P_k}\max_{x\in[a,b]}|p(x)| \le 2\left(\frac{\sqrt{\dfrac{b}{a}}-1}{\sqrt{\dfrac{b}{a}}+1}\right)^k.$$

Demonstrație. Avem $x=\dfrac{b+a}{b-a}$ aşa încât $x^2-1=\dfrac{2\sqrt{ab}}{b-a}$, $x+\sqrt{x^2-1}=\dfrac{(\sqrt{b}-\sqrt{a})^2}{b-a}=\dfrac{\sqrt{b}+\sqrt{a}}{\sqrt{b}-\sqrt{a}}$ şi $x-\sqrt{x^2-1}=\dfrac{\sqrt{b}-\sqrt{a}}{\sqrt{b}+\sqrt{a}}$. Deci

$$T_k\left(\frac{b+a}{b-a}\right)\ge\frac{1}{2}\left(\frac{\sqrt{b}+\sqrt{a}}{\sqrt{b}-\sqrt{a}}\right)^k. \quad \blacksquare$$

Cu aceasta se poate demonstra următoarea teoremă, care precizează rata de convergență a algoritmului de gradient conjugat.

Teorema 2.3.1. *Fie* $A\in R^{n\times n}$ *o matrice simetrică şi pozitiv definită cu* λ_1 *şi* λ_n *cea mai mică, respectiv cea mai mare valoare proprie. Fie* x_0 *un punct inițial şi* x_k *a* k*-a aproximație a soluției* x^* *a sistemului* $Ax=b$, *obținută de algoritmul de gradient conjugat. Atunci următoarea estimație a erorii*

$$\left\|x_k-x^*\right\|_A \le 2\left(\frac{\sqrt{\kappa}-1}{\sqrt{\kappa}+1}\right)^k\left\|x_0-x^*\right\|_A \tag{2.3.18}$$

are loc, unde $\kappa=\lambda_n/\lambda_1$ *este numărul de condiționare al matricei* A.

Demonstrație. Din estimația $\|e_k\|_A \le \max\limits_{\lambda_1\le\lambda\le\lambda_n}|p_k(\lambda)|\|e_0\|_A$ alegând

$$p_k(x) = \frac{T_k(x;\lambda_1,\lambda_n)}{T_k(0;\lambda_1,\lambda_n)}$$

şi având în vedere că $\left|T_k(x;\lambda_1,\lambda_n)\right| < 1$ pentru $x \in [\lambda_1,\lambda_n]$, rezultă că:

$$\|e_k\|_A \le \frac{\|e_0\|_A}{T_k(0;\lambda_1,\lambda_n)} = T_k\left(\frac{\lambda_n+\lambda_1}{\lambda_n-\lambda_1}\right)^{-1}\|e_0\|_A .$$

Acum, utilizând propoziția 2.3.3 obținem concluzia teoremei. ■

Deci, $\|e_k\|_A$ este mărginită superior de un şir care converge la zero. Mai mult, descreşterea este monotonă şi aceasta explică de ce metoda gradientului conjugat se poate privi ca o metodă iterativă. Observăm că cu cât κ este mai aproape de unu, cu atât viteza de convergență a metodei este mai mare. După cum se poate vedea, convergența algoritmului depinde de distribuția *tuturor* valorilor proprii ale matricei A.

Teorema 2.3.1 furnizează o margine superioară a vitezei de convergență a algoritmului de gradient conjugat foarte brută. În plus, proprietatea de terminare finită a algoritmului nu este reflectată de concluzia (2.3.18) a teoremei. Deoarece matricea A are cel mult n valori proprii distincte, atunci pentru orice polinom $q \in P_k$, pentru care $q(\lambda_j) = 0,\ j = 1,\ldots,n,$ din (2.3.15) rezultă că:

$$\|e_k\|_A^2 = \min_{p\in P_k}\sum_{j=1}^{n}\gamma_j^2\lambda_j\left(p(\lambda_j)\right)^2 \le \sum_{j=1}^{n}\gamma_j^2\lambda_j\left(q(\lambda_j)\right)^2 = 0. \qquad (2.3.19)$$

Teorema 2.3.2. *Fie $A \in R^{n\times n}$ o matrice simetrică şi pozitiv definită şi x_k o aproximație a soluției x^* a lui (2.1.1) dată de algoritmul de gradient conjugat. Dacă A are $m \le n$ valori proprii distincte, atunci într-o aritmetică exactă, algoritmul de gradient conjugat realizează $x_k = x^*$ după cel mult m iterații.*

Demonstrație. Dacă A are valorile proprii distincte $\lambda_1 < \cdots < \lambda_m$, atunci selectând

$$q(\lambda) = \prod_{j=1}^{m}\left(\lambda - \lambda_j\right)$$

în (2.3.19) obținem $\|e_m\|_A^2 = 0$. ■

Următorul rezultat a fost demonstrat de Axelsson [1976].

Teorema 2.3.3. *Presupunem că $a \le \lambda_1 \le \lambda_2 \le \cdots \le \lambda_{n-m} \le b \le \lambda_{n-m+1} \le \cdots \le \lambda_n$ şi fie $k \ge m$, atunci*

$$\left\|x_k - x^*\right\|_A \le 2\left(\frac{\sqrt{\frac{b}{a}}-1}{\sqrt{\frac{b}{a}}+1}\right)^{(k-m)} \left\|x_0 - x^*\right\|_A .$$

Demonstrație. În estimația $\left\|e_k\right\|_A \le \max\limits_{\lambda_1 \le \lambda \le \lambda_n} \left|p_k(\lambda)\right| \left\|e_0\right\|_A$ selectăm

$$p_k(\lambda) = \prod_{j=1}^{m}\left(1-\frac{\lambda}{\lambda_{n-j+1}}\right)\frac{T_{k-m}\left(\frac{a+b-2\lambda}{b-a}\right)}{T_{k-m}\left(\frac{a+b}{b-a}\right)}.$$

Clar, $p_k(\lambda_j) = 0$ pentru toți λ_j, cu $j = n, n-1, \ldots, n-m+1$. Dar, când $\lambda \in [a,b]$,

$$\left|1-\frac{\lambda}{\lambda_{n-j+1}}\right| < 1, \quad j = 1, \ldots, m.$$

Deci,

$$\max_{a \le \lambda \le b}\left|p_k(\lambda)\right| \le \max_{a \le \lambda \le b}\left|\frac{T_{k-m}\left(\frac{a+b-2\lambda}{b-a}\right)}{T_{k-m}\left(\frac{a+b}{b-a}\right)}\right|,$$

care demonstrează teorema. ■

Acest rezultat este foarte folositor când câteva dintre valorile proprii mari ale matricei A sunt bine separate de restul valorilor proprii. În acest caz, m este mic şi $k-m$ nu este prea mult diferit de k.

Matricea A rareori are un număr mic $m << n$ de valori proprii distincte. Mult mai plauzibil, spectrul acesteia este distribuit într-un număr de m intervale disjuncte I_j, $j = 1, \ldots, m$, de lungime mică. În acest caz, după m iterații, algoritmul de gradient conjugat va produce un reziduu mic. Într-adevăr, selectând un polinom q de grad m cu $q(0) = 1$ şi având valorile proprii distribuite în aceste intervale I_j, atunci marginea superioară a reducerii erorii în cele m iterații, adică $\max\left\{\left|q(\lambda_j)\right|\right\}$, ($\lambda_j$ fiind valorile proprii ale matricei A), va fi mică. Următorul exemplu ilustrează această situație.

Exemplul 2.3.1. Pentru a vedea efectul grupării valorilor proprii asupra vitezei de convergență a algoritmului de gradient conjugat, să considerăm sistemul liniar $Ax = b$, unde A este o matrice diagonală şi b este ales astfel încât întotdeauna

soluția sistemului este $x^* = [1,1,...,1]$. Criteriul de oprire a iteraţiilor este $\|b - Ax_k\| \le 10^{-6}$. Considerăm $n = 1000$ şi patru distribuţii ale valorilor proprii: $A = diag(1,2,...,n)$ cu $\kappa(A) = 1000$; o matrice diagonală cu elementele uniform distribuite în $[0,1)$ cu $\kappa(A) = 994.637$; o matrice diagonală cu elementele distribuite în 10 intervale cu $\kappa(A) = 10.00943$, şi o matrice diagonală cu elementele distribuite în 5 intervale cu $\kappa(A) = 5.00985$.

Figura 2.3.5 ilustrează viteza de convergenţă a algoritmului de gradient conjugat pentru aceste patru cazuri într-o scară semi-logaritmică.

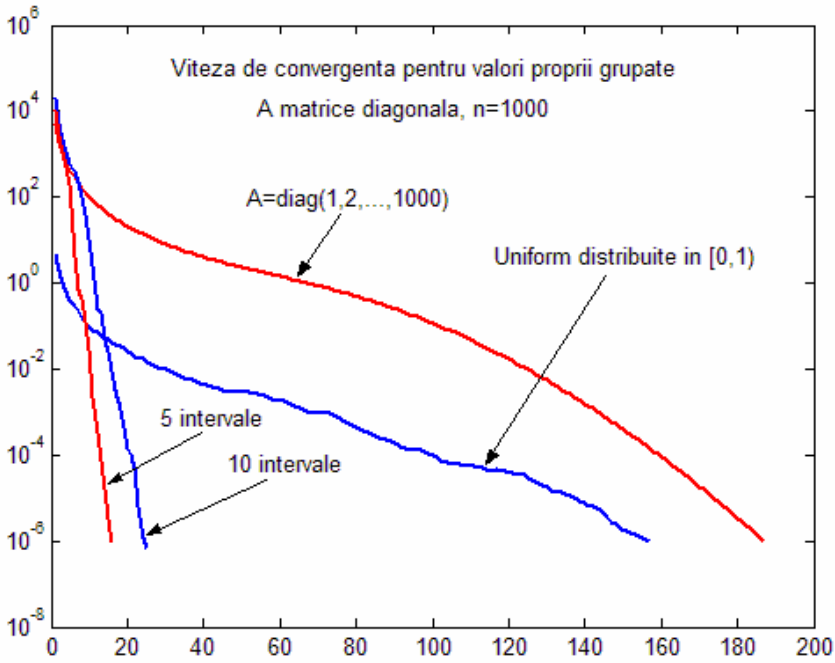


Fig. 2.3.5. Viteza de convergenţă pentru valori proprii grupate.
Evoluția erorii $\|b - Ax_k\|_2$, referitor la numărul de iterații.

Figura 2.3.6 arată situaţia în care matricea A are 999 valori proprii uniform distribuite în $[0,1)$ şi una egală cu 1000. În acest caz numărul de condiţionare al lui A este $\kappa(A) = 996182.727$.

Aceasta este comportarea tipică a algoritmului de gradient conjugat. Observăm că dacă valorile proprii sunt bine grupate într-un număr mic de intervale, atunci reducerea erorii este accelerată, doar câteva iteraţii sunt necesare pentru a obţine o soluţie cu acurateţea impusă. Pe de altă parte, dacă sunt puţine valori proprii mari şi puţine valori proprii mici, atunci algoritmul de gradient conjugat converge mult mai rapid decât predicţia dată de analiza bazată pe numărul de condiţionare a matricei A.

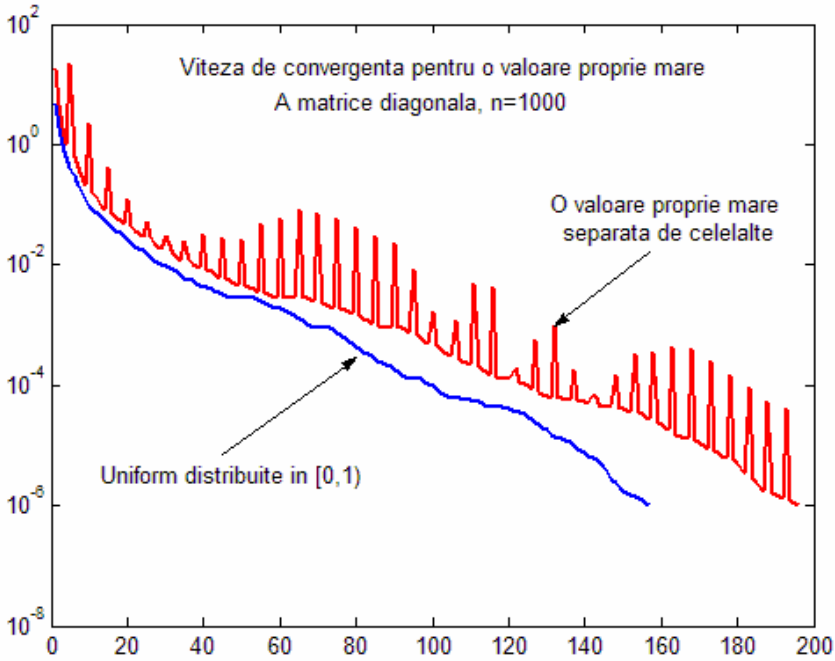


Fig. 2.3.6. Viteza de convergenţă pentru o valoare proprie mare grupată de celelalte. Evoluţia erorii $\|b - Ax_k\|_2$, referitor la numărul de iteraţii.

Să considerăm situaţia în care matricea A este diagonală cu $n = 1000$, cu diferite mulţimi de valori proprii distincte. Proprietăţile de convergenţă sunt date în tabelul 2.3.1.

Tabelul 2.3.1. Numărul de iteraţii în funcţie de numărul de valori proprii distincte

Nr. valori proprii distincte	2	10	20	50	100	500	1000
Nr. iteraţii	3	11	21	43	62	142	188

Vedem că numărul de iteraţii creşte odată cu numărul de valori proprii distincte. Din teorema 2.3.2, în absenţa erorilor, cunoaştem că numărul de iteraţii efectuat de algoritmul de gradient conjugat este egal cu numărul de valori proprii distincte. Totuşi, în virgulă mobilă comportarea algoritmului de gradient conjugat poate fi semnificativ diferită de comportarea sa în aritmetica exactă [Demmel, 1997]. Este important să notăm că numărul de iteraţii este mult inferior dimensiunii sistemului. Aceasta este o caracteristică tipică metodelor de gradient conjugat, faţă de metodele directe bazate pe eliminarea Gauss.

2.4. DISCRETIZAREA ECUAȚIILOR DIFERENȚIALE CU DERIVATE PARȚIALE

O sursă foarte importantă de sisteme de ecuații algebrice liniare o constituie discretizarea ecuațiilor diferențiale cu derivate parțiale. Procedura standard de rezolvare a acestor ecuații este discretizarea, adică aproximarea acestora prin ecuații care implică un număr finit de necunoscute. Matricele sistemelor care provin din discretizare sunt mari şi rare, adică au puține elemente nenule. În această secțiune vom considera două tipuri de ecuații diferențiale cu derivate parțiale: ecuația de difuziune şi ecuația de transport. Pentru început să considerăm cazul unei ecuații diferențiale simple.

Diferențe finite pentru probleme cu o dimensiune
Fie ecuația diferențială cu condițiile inițiale:

$$-u''(x) = f(x)\,,\ x \in (0,1)$$
$$u(0) = u(1) = 1.$$

Pentru integrarea acestei ecuații simple intervalul [0,1] este uniform discretizat prin intermediul a $n+2$ puncte $x_i = ih$, unde $i = 0,1,\ldots,n+1$, iar $h = 1/(n+1)$. Din condițiile inițiale (Dirichlet) cunoaştem $u(x_0)$ şi $u(x_{n+1})$. În toate celelalte puncte se calculează aproximațiile u_i a soluției exacte $u(x_i)$. Dacă se utilizează aproximarea prin diferențe finite centrale

$$\frac{d^2u(x)}{dx^2} = \frac{u(x+h) - 2u(x) + u(x-h)}{h^2} + \frac{h^2}{12}\frac{d^4u(\xi)}{dx^4},\quad x-h \le \xi \le x+h\,,$$

atunci, neglijând termenul de ordinul 4, necunoscutele u_i, u_{i-1} şi u_{i+1} satisfac relația

$$-u_{i-1} + 2u_i - u_{i+1} = h^2 f_i, \tag{2.4.1}$$

în care $f_i \equiv f(x_i)$. Evident că (2.4.1) reprezintă un sistem cu n necunoscute a cărui matrice este tridiagonală.

Exemplul 2.4.1. Să considerăm sistemul $Ax = b$, obținut din discretizarea unei ecuații diferențiale de mai sus, unde:

$$A = \begin{bmatrix} 2 & -1 & & & \\ -1 & 2 & \ddots & & \\ & \ddots & \ddots & \ddots & \\ & & \ddots & 2 & -1 \\ & & & -1 & 2 \end{bmatrix},\quad b = \begin{bmatrix} 1 \\ 0 \\ \vdots \\ 0 \\ 1 \end{bmatrix}.$$

Figura 2.4.1 ilustrează evoluția erorii $\|b - Ax_k\|_2$, referitor la numărul de iterații pentru a obține soluția cu acuratețea mai mică decât sau egală cu 10^{-6}.

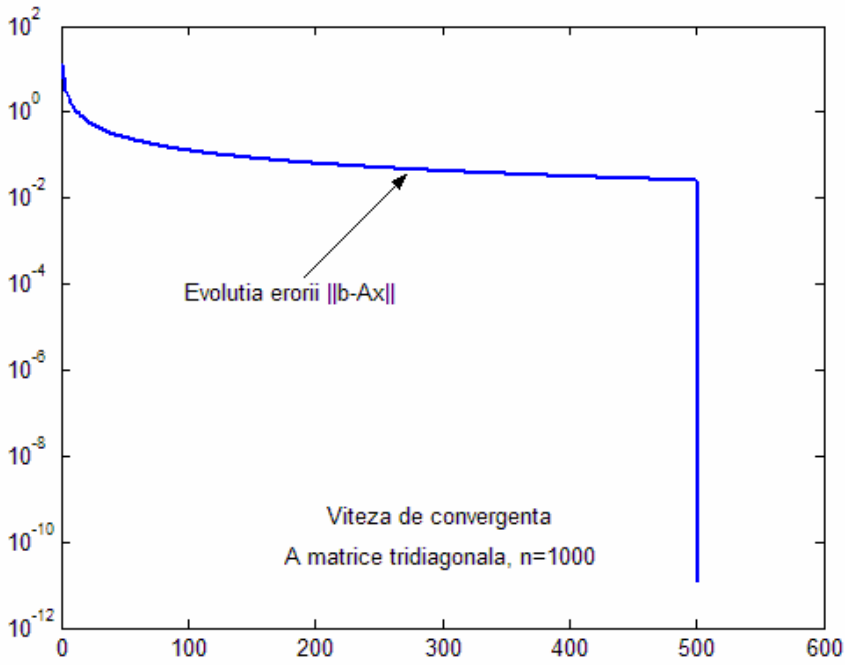


Fig. 2.4.1. Viteza de convergență.
Evoluția erorii $\|b - Ax_k\|_2$, referitor la numărul de iterații.

Valorile proprii ale lui A sunt $\lambda_i = 2\left(1 - \cos\frac{\pi i}{n+1}\right)$. Figura 2.4.2 arată valorile proprii ale lui A pentru $n = 21$. Observăm că valoarea proprie maximă este $\lambda_n = 2\left(1 - \cos\frac{\pi n}{n+1}\right) \approx 4$. Valoarea proprie minimă este $\lambda_1 = 2\left(1 - \cos\frac{\pi}{n+1}\right)$.
Pentru i mic avem

$$\lambda_i = 2\left(1 - \cos\frac{\pi i}{n+1}\right) \approx 2\left(1 - \left(1 - \frac{\pi^2 i^2}{2(n+1)^2}\right)\right) = \left(\frac{\pi i}{n+1}\right)^2.$$

Deci, A este pozitiv definită cu numărul de condiționare

$$\frac{\lambda_n}{\lambda_1} \approx \frac{4(n+1)^2}{\pi^2}, \text{ pentru } n \text{ mare.}$$

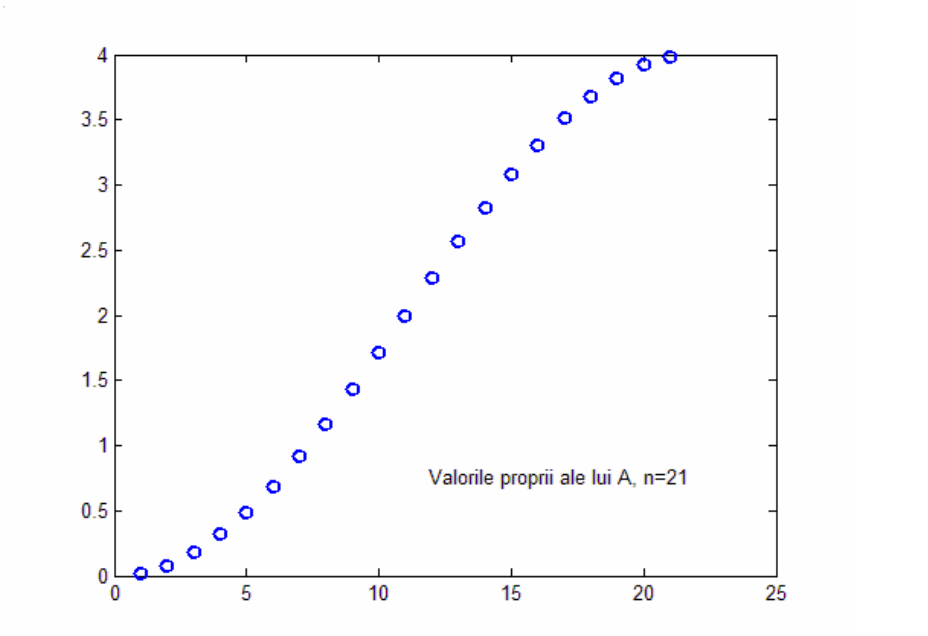


Fig. 2.4.2. Valorile proprii ale matricei A.

În acest caz comportarea algoritmului de gradient conjugat include un „platou" în care pentru un număr mare de iterații eroarea $\|b - Ax_k\|_2$ descreşte foarte puțin. O explicație a acestei comportări a fost dată de Greenbaum şi Strakos [1992] care arată că algoritmul de gradient conjugat aplicat sistemului $Ax = b$ în virgulă mobilă, se comportă exact la fel ca algoritmul de gradient conjugat aplicat sistemului $\tilde{A}\tilde{x} = \tilde{b}$ în aritmetică exactă, unde $\tilde{A}$ este aproape de A în următorul sens: $\tilde{A}$ este de dimensiune mult mai mare decât dimensiunea lui A, dar valorile proprii ale lui $\tilde{A}$ se află grupate în jurul valorilor proprii ale lui A. Vezi de asemenea [Naiman, Babuska şi Elman, 1997].

Ecuația de difuziune

Modelele multor procese fizice se exprimă ca exprimă ca ecuații de difuzie:

$$\frac{\partial u}{\partial t} - \nabla.(a\nabla u) = f\,. \tag{2.4.2}$$

În (2.4.2) u poate reprezenta distribuția de temperatură la momentul de timp t într-un obiect Ω asupra căruia acționează o sursă externă de căldură f. Coeficientul pozitiv $a(x)$ reprezintă conductivitatea termică a materialului din care este făcut obiectul Ω. Pentru a determina temperatura la momentul de timp t este necesar să cunoaştem distribuția inițială de temperatură $u(x,0)$ precum şi anumite condiții pe frontiera domeniului:

$$u(x,t)=0, \quad x\in\partial\Omega, \tag{2.4.3}$$

ceea ce înseamnă că pe frontiera domeniului $\partial\Omega$ temperatura este fixată la o valoare constantă (aici zero).

Ecuația (2.4.2) modelează foarte bine alte fenomene fizice ca, de exemplu, difuzia unei substanțe printr-o regiune permeabilă Ω, caz în care u este concentrația substanței, a este coeficientul de difuziune a materialului, iar f este rata specifică de generare a substanței prin reacții chimice sau anumite surse externe.

Metoda standard de obținere a unei aproximații a soluțiilor ecuației diferențiale cu derivate parțiale (2.4.2) este prin diferențe finite. În acest caz regiunea Ω este discretizată printr-o rețea de discretizare şi în fiecare punct al rețelei din interiorul lui Ω, derivatele din (2.4.2) sunt înlocuite prin rapoartele standard care tind la derivatele funcției când rețeaua de discretizare devine din ce în ce mai fină. Ca exemplu, să considerăm că regiunea Ω este pătratul unitar $[0,1]\times[0,1]$. Atunci o discretizare uniformă $\{x_i, y_j\}$, unde $i=0,1,\ldots n_x+1$ şi $j=0,1,\ldots,n_y+1$, este caracterizată de pasul de discretizare $h_x=1/(n_x+1)$ în direcția x şi $h_y=1/(n_y+1)$ în direcția y. Cu acestea, aproximarea standard prin diferențe finite centrale a derivatelor parțiale din (2.4.2) în direcția x este:

$$\left(\frac{\partial}{\partial x}a\frac{\partial u}{\partial x}\right)(x_i,y_j)\cong\frac{a_{i+1/2,j}(u_{i+1,j}-u_{i,j})-a_{i-1/2,j}(u_{i,j}-u_{i-1,j})}{h_x^2},$$

unde $a_{i\pm1/2,j}=a(x_i\pm h_x/2,y_j)$, iar $u_{i,j}$ reprezintă aproximarea lui $u(x_i,y_j)$. Asemănător se exprimă derivatele parțiale în direcția y:

$$\left(\frac{\partial}{\partial y}a\frac{\partial u}{\partial y}\right)(x_i,y_j)\cong\frac{a_{i,j+1/2}(u_{i,j+1}-u_{i,j})-a_{i,j-1/2}(u_{i,j}-u_{i,j-1})}{h_y^2},$$

unde acum $a_{i,j\pm1/2}=a(x_i,y_j\pm h_y/2)$.

Este important de notat că sunt probleme în care coeficientul $a(x,y)$ este discontinuu pe rețeaua de discretizare. În asemenea situații valorile $a_{i\pm1/2,j}$ şi $a_{i,j\pm1/2}$ se consideră egale cu media valorilor vecine punctelor considerate. De exemplu, dacă $a(x,y)$ este discontinuă de-a lungul liniei $y=y_j$, atunci

$$a_{i\pm1/2,j}\equiv\lim_{\varepsilon\to0^+}\frac{a(x_i\pm1/2h_x,y_j+\varepsilon)+a(x_i\pm1/2h_x,y_j-\varepsilon)}{2}.$$

Dacă suntem interesați de versiunea staționară a problemei (2.4.2)-(2.4.3):

$$-\nabla.(a\nabla u)=f \quad \text{pe } \Omega=(0,1)\times(0,1),$$
$$u(x,0)=u(x,1)=u(0,y)=u(1,y)=0,$$

atunci aproximația prin diferențe finite a acesteia ne conduce la următorul sistem algebric liniar n_xn_y-dimensional:

$$-\left(\frac{a_{i+1/2,j}(u_{i+1,j}-u_{i,j})-a_{i-1/2,j}(u_{i,j}-u_{i-1,j})}{h_x^2}+\right.$$

$$\left.\frac{a_{i,j+1/2}(u_{i,j+1}-u_{i,j})-a_{i,j-1/2}(u_{i,j}-u_{i,j-1})}{h_y^2}\right)=f_{i,j}, \qquad (2.4.4)$$

$$i=1,\ldots,n_x,\quad j=1,\ldots,n_y,$$

în care necunoscutele sunt valorile $u_{i,j}$ din interiorul reţelei de discretizare.

Dacă suntem interesaţi de versiunea nestaţionară (dependentă de timp), atunci deseori derivatele temporale se aproximează prin diferenţe înapoi sau centrale, ceea ce ne conduce la rezolvarea unui sistem de ecuaţii algebrice liniare la fiecare pas de discretizare a intervalului de timp considerat. De exemplu, dacă la momentul de timp t_l soluţia $u_{i,j}^l$ este cunoscută şi dacă se utilizează diferenţele finite înapoi în timp, atunci pentru a calcula aproximaţiile $u_{i,j}^{l+1}$ ale soluţiei ecuaţiei (2.4.2) la momentul de timp $t_{l+1}=t_l+\Delta t$, este necesar să rezolvăm următorul sistem de ecuaţii algebrice liniare:

$$\frac{u_{i,j}^{l+1}-u_{i,j}^l}{\Delta t}-\left(\frac{a_{i+1/2,j}\left(u_{i+1,j}^{l+1}-u_{i,j}^{l+1}\right)-a_{i-1/2,j}\left(u_{i,j}^{l+1}-u_{i-1,j}^{l+1}\right)}{h_x^2}+\right.$$

$$\left.\frac{a_{i,j+1/2}\left(u_{i,j+1}^{l+1}-u_{i,j}^{l+1}\right)-a_{i,j-1/2}\left(u_{i,j}^{l+1}-u_{i,j-1}^{l+1}\right)}{h_y^2}\right)=f_{i,j}^{l+1}, \qquad (2.4.5)$$

$$i=1,\ldots,n_x,\quad j=1,\ldots,n_y.$$

Pentru ca ecuaţiile (2.4.4) sau (2.4.5) să le scriem în formă matriceală, este necesar să precizăm o ordine a ecuaţiile şi necunoscutelor. De obicei se alege ordinea naturală în care punctele de discretizare ale reţelei de discretizare sunt numerotate de la stânga la dreapta şi de jos în sus. Cu această ordonare a ecuaţiilor şi necunoscutelor sistemul (2.4.4) se poate scrie ca:

$$Au=f, \qquad (2.4.6)$$

unde A este o matrice bloc tridiagonală cu n_y blocuri diagonale, fiecare de dimensiune $n_x\times n_x$, u este un n_xn_y- vector al valorilor funcţiei necunoscută cu $u_{i,j}$ memorate în poziţia $(j-1)n_x+i$ şi f este n_xn_y- vectorul membrului drept cu $f_{i,j}$ în poziţia $(j-1)n_x+i$.

Definim:

$$d_{i,j}\equiv\frac{a_{i+1/2,j}+a_{i-1/2,j}}{h_x^2}+\frac{a_{i,j+1/2}+a_{i,j-1/2}}{h_y^2},$$

$$b_{i+1/2,j} \equiv \frac{-a_{i+1/2,j}}{h_x^2},$$

$$c_{i,j+1/2} \equiv \frac{-a_{i,j+1/2}}{h_y^2}.$$

Atunci matricea A se poate scrie sub forma:

$$A = \begin{bmatrix} S_1 & T_{3/2} & & \\ T_{3/2} & \ddots & \ddots & \\ & \ddots & \ddots & T_{n_y-1/2} \\ & & T_{n_y-1/2} & S_{n_y} \end{bmatrix}, \tag{2.4.7}$$

unde

$$S_j = \begin{bmatrix} d_{1,j} & b_{3/2,j} & & \\ b_{3/2,j} & \ddots & \ddots & \\ & \ddots & \ddots & b_{n_x-1/2,j} \\ & & b_{n_x-1/2,j} & d_{n_x,j} \end{bmatrix}, \tag{2.4.8}$$

$$T_{j+1/2} = \begin{bmatrix} c_{1,j+1/2} & & & \\ & \ddots & & \\ & & \ddots & \\ & & & c_{n_x,j+1/2} \end{bmatrix}. \tag{2.4.9}$$

Pentru problemele dependente de timp (nestaționare) sistemul (2.4.6) rămâne acelaşi, dar elementele diagonale ale matricei A sunt înmulțite cu $1/\Delta t$ şi la membrul drept se adună termenii $u_{i,j}^l / \Delta t$.

Teorema 2.4.1. *Presupunem că* $a(x,y) \geq \alpha > 0$ *în* $(0,1)\times(0,1)$. *Atunci matricea* A *a sistemului (2.4.6), definită de (2.4.7)-(2.4.9) este simetrică şi pozitiv definită.*

Demonstrație. Simetria este evidentă. Matricea este slab diagonal dominantă, ceea ce înseamnă că din teorema lui Gerschgorin toate valorile proprii sunt mai mari sau egale cu zero. Presupunem că există un vector nenul v astfel încât $Av = 0$ şi că elementul vectorului v corespunzător punctului (i,j) din rețeaua de discretizare are cea mai mare valoare absolută. Evident că se poate alege semnul lui v astfel încât acest element să fie pozitiv. Din definiția matricei A şi presupunerea $a(x,y) > 0$, rezultă că $v_{i,j}$ se poate exprima ca o medie ponderată a vecinilor lui din v:

$$v_{i,j} = w_{i-1,j}v_{i-1,j} + w_{i+1,j}v_{i+1,j} + w_{i,j-1}v_{i,j-1} + w_{i,j+1}v_{i,j+1},$$

$$w_{i\pm1,j} \equiv \frac{1}{d_{i,j}} \frac{a_{i\pm1/2,j}}{h_x^2}, \qquad w_{i,j\pm1} \equiv \frac{1}{d_{i,j}} \frac{a_{i,j\pm1/2}}{h_y^2},$$

unde termenii corespunzători nodurilor de pe frontieră sunt zero. Ponderile $w_{i\pm1,j}$ şi $w_{i,j\pm1}$ sunt pozitive şi suma lor este egală cu 1. Rezultă deci că dacă toţi vecinii lui $v_{i,j}$ sunt puncte interioare, atunci toate trebuie să aibă aceeaşi valoare maximă, deoarece nici unul nu poate fi mai mare decât $v_{i,j}$. Utilizând acest argument în punctele vecine în cele din urmă găsim un punct cu aceeaşi valoare maximă care are cel puţin un punct vecin pe frontiera domeniului. Dar valoarea lui v în acest punct este o sumă ponderată a valorilor punctelor interioare, unde acu suma ponderilor este mai mică decât 1. Deci valoarea lui v în unul dintre aceste puncte interioare trebuie să fie mai mare decât $v_{i,j}$ dacă $v_{i,j} > 0$, ceea ce este o contradicţie. Deci singurul vector v pentru care $Av = 0$ este vectorul nul, ceea ce arată că A este pozitiv definită. ■

Este clar că matricea A corespunzătoare problemelor dependente de timp (nestaţionare) (2.4.5) este de asemenea pozitiv definită, deoarece ea este strict diagonal dominantă.

Să observăm că argumentul utilizat în demonstraţia teoremei 2.4.1 este de tipul *principiul maximului discret*. Într-adevăr, în demonstraţie nu se face uz de valorile elementelor matricei A, ci numai de faptul că aceasta are elemente diagonale pozitive şi elemente ne-diagonale nepozitive (aşa încât ponderile din media ponderată sunt pozitive); că A este slab diagonal dominantă cu dominanţă diagonală tare în cel puţin o linie şi că plecând dintr-un punct (nod) oarecare (i, j) al reţelei, se poate ajunge la oricare alt punct prin intermediul unui drum care uneşte cele mai apropiate noduri.

Matricea A definită de (2.4.7)-(2.4.9) se numeşte aproximarea în 5 puncte, deoarece derivatele de ordinul doi în punctul (i, j) al reţelei de discretizare sunt aproximate în termenii valorilor funcţiei în punctul (i, j) şi alte patru puncte vecine care se află sus, jos şi dreapta, stânga. Evident că o aproximaţie mai fină se obţine cu o aproximare în 9 puncte în care se utilizează valoarea funcţiei în punctul (i, j), precum şi în alte opt puncte vecine.

Un alt mod de abordare pentru obţinerea unor aproximaţii ale soluţiei ecuaţiilor cu derivate parţiale este *metoda elementului finit*. Ideea acestei metode este de a aproxima soluţia prin funcţii polinomiale pe porţiuni.

După cum se vede din dezvoltările de mai sus o sursă foarte importantă de sisteme algebrice liniare cu matricea coeficienţilor simetrică şi pozitiv definită este dată de discretizarea ecuaţiei de difuziune. Metodele de gradient conjugat sunt foarte bune pentru rezolvarea acestor sisteme. În cele ce urmează să considerăm un caz particular important în care coeficientul $a(x, y)$ este o constantă. Aceasta ne conduce la ecuaţia Poisson pentru care putem preciza anumite proprietăţi asupra valorilor proprii ale matricei asociate prin discretizare [Greenbaum, 1997].

Ecuația Poisson

În cazul special când coeficientul $a(x,y)$ este o constantă, să zicem $a(x,y)=1$, atunci matricea sistemului $Au=f$ pentru problema staționară (cunoscută ca ecuația Poisson) are următoarea formă specială:

$$A=\begin{bmatrix} S & T & & \\ T & \ddots & \ddots & \\ & \ddots & \ddots & T \\ & & T & S \end{bmatrix}, \tag{2.4.10}$$

unde

$$T=(-1/h_y^2)I$$

şi

$$S=\begin{bmatrix} d & b & & \\ b & \ddots & \ddots & \\ & \ddots & \ddots & b \\ & & b & d \end{bmatrix}, \quad d=\frac{2}{h_x^2}+\frac{2}{h_y^2}, \quad b=\frac{-1}{h_x^2}. \tag{2.4.11}$$

Matricea S se numeşte *matrice TST – Toeplitz, simetrică, tridiagonală*. Matricea A este o matrice *TST bloc*, deoarece este o matrice simetrică, bloc tridiagonală şi matricele de pe diagonala principală sunt de asemenea TST. Următoarea teoremă precizează valorile proprii şi vectorii proprii unei matrice TST.

Propoziția 2.4.1. *Fie G o $m\times m-$matrice TST cu elementele diagonale egale cu α şi elementele nediagonale β. Atunci valorile proprii ale lui G sunt:*

$$\lambda_k=\alpha+2\beta\cos\left(\frac{k\pi}{m+1}\right), \quad k=1,\ldots,m, \tag{2.4.12}$$

şi vectorii proprii (ortonormali) sunt:

$$q_l^{(k)}=\sqrt{\frac{2}{m+1}}\sin\left(\frac{lk\pi}{m+1}\right), \quad l,k=1,\ldots,m. \tag{2.4.13}$$

Demonstrație. Cel mai simplu este să verificăm că într-adevăr valorile proprii şi vectorii proprii de mai sus (2.4.12) şi (2.4.13) verifică relațiile fundamentale de definiție a acestor elemente pentru matricea TST dată. Totuşi, în cele ce urmează vom prezenta modul cum aceste perechi proprii sunt obținute.

Presupunem că $\beta\neq 0$, deoarece altfel G este un multiplu al matricei identitate şi propoziția este trivială. Fie λ o valoare proprie a lui G şi q vectorul propriu corespunzător, deci (λ,q) formează o pereche proprie a lui G. Dacă considerăm $q_0=q_{m+1}=0$, atunci $Aq=\lambda q$ se poate scrie ca:

$$\beta q_{l-1}+(\alpha-\lambda)q_l+\beta q_{l+1}=0, \quad l=1,\ldots,m. \tag{2.4.14}$$

Vedem că (2.4.14) este o ecuaţie cu diferenţe finite care se poate rezolva prin considerarea ecuaţiei liniare diferenţiale corespunzătoare. Pentru (2.4.14) ecuaţia caracteristică este:

$$\chi(z) = \beta + (\alpha - \lambda)z + \beta z^2.$$

Dacă notăm cu z_+ şi cu z_- rădăcinile acestei ecuaţii, atunci soluţia generală a ecuaţiei cu diferenţe (2.4.14) este:

$$q_l = c_1 z_+^l + c_2 z_-^l,$$

unde c_1, c_2 sunt constante determinate din condiţiile $q_0 = q_{m+1} = 0.$

Rădăcinile ecuaţiei $\chi(z) = 0$ sunt;

$$z_\pm = \frac{(\lambda - \alpha) \pm \sqrt{(\lambda - \alpha)^2 - 4\beta^2}}{2\beta}. \tag{2.4.15}$$

Condiţia $q_0 = 0$ implică $c_1 + c_2 = 0.$ Pe de altă parte, condiţia $q_{m+1} = 0$ implică $z_+^{m+1} = z_-^{m+1}.$ Această ecuaţie are $m+1$ soluţii:

$$z_+ = z_- \exp\left(\frac{2\pi k\imath}{m+1}\right), \quad k = 0, 1, \ldots, m, \quad \imath = \sqrt{-1}. \tag{2.4.16}$$

Cazul $k = 0$ se poate elimina deoarece corespunde situaţiei $z_+ = z_-$, pentru care $q_l = 0.$ Acum înmulţind (2.4.16) cu $\exp(-\pi k\imath/(m+1))$ şi introducând valorile lui $z_\pm$ din (2.4.15) obţinem:

$$\left((\lambda - \alpha) + \sqrt{(\lambda - \alpha)^2 - 4\beta^2}\right)\exp\left(\frac{-\pi k\imath}{m+1}\right) =$$

$$\left((\lambda - \alpha) + \sqrt{(\lambda - \alpha)^2 - 4\beta^2}\right)\exp\left(\frac{\pi k\imath}{m+1}\right).$$

O mică prelucrare algebrică ne conduce la:

$$\sqrt{(\lambda - \alpha)^2 - 4\beta^2}\cos\left(\frac{k\pi}{m+1}\right) = (\lambda - \alpha)\imath\sin\left(\frac{k\pi}{m+1}\right).$$

Ridicând la pătrat şi rezolvând în privinţa lui λ obţinem:

$$\lambda = \alpha \pm 2\beta\cos\left(\frac{k\pi}{m+1}\right).$$

Observăm că dacă în relaţia de mai sus luăm semnul +, atunci obţinem (2.4.12), în timp ce semnul – repetă aceleaşi valori, care se pot elimina.

Acum introducând această valoare a lui λ în (2.4.15) obţinem:

$$z_\pm = \cos\left(\frac{k\pi}{m+1}\right) \pm \imath\sin\left(\frac{k\pi}{m+1}\right),$$

De unde rezultă că

$$q_l^{(k)} = c_1(z_+^l - z_-^l) = 2c_1\imath \sin\left(\frac{k\pi l}{m+1}\right), \quad k,l = 1,\ldots,m.$$

Dacă considerăm $c_1 = -(\imath/2)\sqrt{2/(m+1)}$, ca în (2.4.13), atunci se vede imediat că fiecare vector $q^{(k)}$ are norma egală cu 1. Adică vectorii proprii sunt ortogonali deoarece matricea este simetrică. ■

Corolarul 2.4.1. *Toate matricele TST, $m \times m -$ dimensionale comută unele cu altele.*

Demonstrație. Din (2.4.13) vedem că toate matricele TST au aceiaşi vectori ortonormali. ($q_l^{(k)}$ nu depind de α şi β.) Dacă $G_1 = Q\Lambda_1 Q^T$ şi $G_2 = Q\Lambda_2 Q^T$, atunci $G_1G_2 = Q\Lambda_1\Lambda_2 Q^T = Q\Lambda_2\Lambda_1 Q^T = G_2G_1$. ■

Teorema 2.4.2. *Valorile proprii ale matricei A definită de (2.4.10)-(2.4.11) sunt:*

$$\lambda_{j,k} = \frac{4}{h_x^2}\sin^2\left(\frac{j\pi}{2(n_x+1)}\right) + \frac{4}{h_y^2}\sin^2\left(\frac{k\pi}{2(n_y+1)}\right), \tag{2.4.17}$$

$$j = 1,\ldots,n_x, \quad k = 1,\ldots,n_y.$$

Vectorii proprii ai matricei A sunt:

$$u_{m,l}^{(j,k)} = \frac{2}{\sqrt{(n_x+1)(n_y+1)}}\sin\left(\frac{mj\pi}{n_x+1}\right)\sin\left(\frac{lk\pi}{n_y+1}\right), \tag{2.4.18}$$

$$m,j = 1,\ldots,n_x, \quad l,k = 1,\ldots,n_y,$$

unde $u_{m,l}^{(j,k)}$ sunt componentele vectorului propriu $\lambda_{j,k}$ corespunzătoare punctului (m,l) din rețeaua de discretizare.

Demonstrație. Fie λ o valoare proprie a matricei A cu vectorul propriu u care se poate partiționa în forma:

$$u = \begin{bmatrix} u_1 \\ \vdots \\ u_{n_y} \end{bmatrix}, \quad u_l = \begin{bmatrix} u_{1,l} \\ \vdots \\ u_{n_x,l} \end{bmatrix}, \quad l = 1,\ldots,n_y.$$

Ecuația $Au = \lambda u$ se poate scrie sub forma:

$$Tu_{l-1} + (S - \lambda I)u_l + Tu_{l+1} = 0, \quad l = 1,\ldots,n_y, \tag{2.4.19}$$

unde aici pentru uniformitatea scrierii am introdus $u_0 = u_{n_y+1} = 0$. Utilizând propoziția 2.4.1 putem scrie $S = Q\Lambda_S Q^T$ şi $T = Q\Lambda_T Q^T$, unde Λ_S şi Λ_T sunt matrice diagonale cu al $j-$ lea element diagonal:

$$\lambda_{S,j} = \frac{2}{h_x^2} + \frac{2}{h_y^2} - \frac{2}{h_x^2}\cos\left(\frac{j\pi}{n_x+1}\right), \quad \lambda_{T,j} = \frac{-1}{h_y^2}.$$

Al $m-$lea element al coloanei j al matricei Q este

$$q_m^{(j)} = \sqrt{\frac{2}{n_x+1}}\sin\left(\frac{mj\pi}{n_x+1}\right), \quad m, j = 1,\ldots,n_x.$$

Înmulțind la stânga (2.4.19) cu Q^T obținem:

$$\Lambda_T y_{l-1} + (\Lambda_S - \lambda I)y_l + \Lambda_T y_{l+1} = 0,$$

unde $y_l = Q^T u_l$, $l = 1,\ldots,n_y$. Deoarece matricele sunt diagonale, ecuațiile corespunzătoare liniilor verticale din rețeaua de discretizare se decuplează ca:

$$\lambda_{T,j} y_{j,l+1} + \lambda_{S,j} y_{j,l} + \lambda_{T,j} y_{j,l-1} = \lambda y_{j,l}, \quad j = 1,\ldots,n_x. \tag{2.4.20}$$

Dacă pentru o valoare fixată a lui j vectorul $(y_{j,1},\ldots,y_{j,n_y})^T$ este un vector propriu al matricei

$$\begin{bmatrix} \lambda_{S,j} & \lambda_{T,j} & & \\ \lambda_{T,j} & \ddots & \ddots & \\ & \ddots & \ddots & \lambda_{T,j} \\ & & \lambda_{T,j} & \lambda_{S,j} \end{bmatrix}$$

cu valoarea proprie λ. Dacă celelalte componente ale vectorului y sunt nule, atunci ecuațiile (2.4.20) vor fi satisfăcute. Din propoziția 2.4.1 rezultă că valorile proprii ale acestei matrice sunt:

$$\begin{aligned} \lambda_{j,k} &= \lambda_{S,j} + 2\lambda_{T,j}\cos\left(\frac{k\pi}{n_y+1}\right) \\ &= \frac{2}{h_x^2} + \frac{2}{h_y^2} - \frac{2}{h_x^2}\cos\left(\frac{j\pi}{n_x+1}\right) - \frac{2}{h_y^2}\cos\left(\frac{k\pi}{n_y+1}\right) \\ &= \frac{4}{h_x^2}\sin^2\left(\frac{j\pi}{2(n_x+1)}\right) + \frac{4}{h_y^2}\sin^2\left(\frac{k\pi}{2(n_y+1)}\right). \end{aligned}$$

Vectorii proprii corespunzători sunt:

$$y_{j,l}^{(j,k)} = \sqrt{\frac{2}{n_y+1}}\sin\left(\frac{lk\pi}{n_y+1}\right).$$

Deoarece al $l-$lea bloc al lui $u^{(j,k)}$ este egal cu Q înmulțit cu al $l-$lea bloc al lui y şi deoarece numai al $j-$lea element al blocului l al lui y este nenul, atunci

$$u_{m,l}^{(j,k)} = q_m^{(j)} y_{j,l}^{(j,k)} = \frac{2}{\sqrt{(n_x+1)(n_y+1)}}\sin\left(\frac{mj\pi}{n_x+1}\right)\sin\left(\frac{lk\pi}{n_y+1}\right).$$

Pentru $j = 1, \ldots, n_x$ ca mai sus se pot obține toate cele $n_x n_y$ perechi proprii $(\lambda_{j,k}, u^{(j,k)})$. ■

Corolarul 2.4.2. *Presupunem că $h_x = h_y = h$. Atunci cea mai mică şi respectiv cea mai mare valoare proprie a matricei A din (2.4.10)-(2.4.11) se comportă ca*

$$2\pi^2 + O(h^2) \quad \text{şi} \quad 8h^{-2} + O(1) \tag{2.4.21}$$

când $h \to 0$, deci numărul de condiționare al matricei A este $(4/\pi^2)h^{-2} + O(1)$.

Demonstrație. Din (2.4.17) vedem că cea mai mică valoare proprie a lui A corespunde lui $j = k = 1$. Cea mai mare este cea corespunzătoare lui $j = k = n_x = n_y$. Adică

$$\lambda_{\min} = \frac{8}{h^2}\sin^2\left(\frac{\pi h}{2}\right), \quad \lambda_{\max} = \frac{8}{h^2}\sin^2\left(\frac{\pi}{2} - \frac{\pi h}{2}\right).$$

Acum, dezvoltând în serie Taylor $\sin(x)$ şi $\sin(\pi/2 - x)$ şi introducând aceste dezvoltări în relațiile de mai sus obținem 92.4.21). Împărțind $\lambda_{\max}$ prin $\lambda_{\min}$ obținem numărul de condiționare al matricei A. ■

Deoarece cunoaştem valorile şi vectorii proprii pentru matricea sistemului care provine din discretizarea în 5 puncte a ecuației Poisson definită pe un pătrat, rezultă că deseori matricele de precondiționare sunt analizate şi chiar testate pe această problemă particulară, cunoscută ca o problemă model. Avantajul utilizării metodelor de gradient conjugat pentru rezolvarea sistemelor liniare algebrice care provin din discretizarea sistemelor de ecuații diferențiale cu derivate parțiale este că acestea se pot aplica unor probleme mai generale decât problema model, ca: ecuația de difuziune cu coeficient de difuziune neconstant, ecuația Poisson pe un domeniu neregulat, ecuația Poisson cu o rețea de discretizare neuniforma etc. În continuare să prezentăm un exemplu care provine din discretizarea ecuației Poisson [Greenbaum, 1997].

Exemplul 2.4.2. Să considerăm acum sistemul $Ax = b$, în care:

$$A = \begin{bmatrix} B & -I & & & \\ -I & B & -I & & \\ & \ddots & \ddots & \ddots & \\ & & -I & B & -I \\ & & & -I & B \end{bmatrix}, \text{ unde } B = \begin{bmatrix} 4 & -1 & & & \\ -1 & 4 & -1 & & \\ & \ddots & \ddots & \ddots & \\ & & -1 & 4 & -1 \\ & & & -1 & 4 \end{bmatrix}$$

şi $B \in R^{n_1 \times n_1}$, matricea A având n_2 blocuri pe diagonala principală. Deci, $A \in R^{n \times n}$, unde $n = n_1 n_2$. Membrul drept b este selectat astfel încât soluția sistemului este $x = [1,1,...,1]^T$. Considerând $n = 5000$, viteza de convergență pentru diferite valori ale lui n_1 şi n_2 este ilustrată în figura 2.4.3.

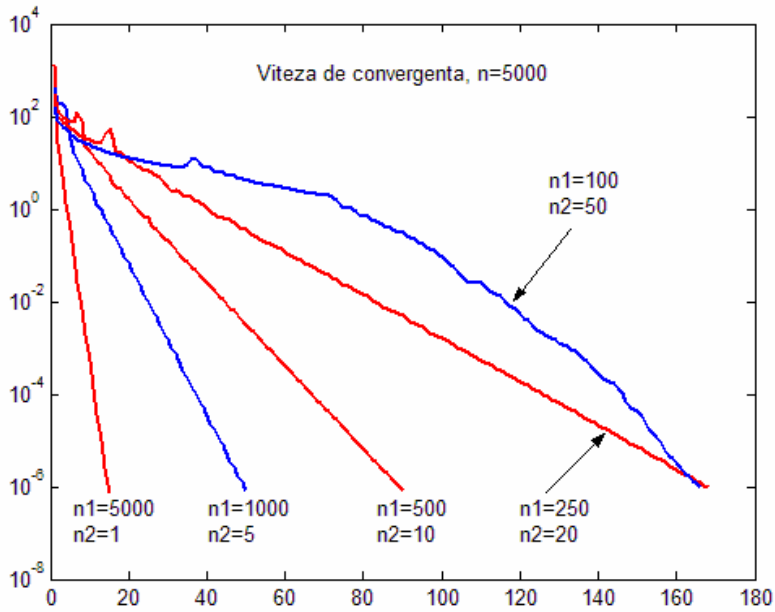


Fig. 2.4.3. Viteza de convergență.
Evoluția erorii $\|b - Ax_k\|_2$, referitor la numărul de iterații.

Din teorema 2.4.2 (vezi (2.4.17)), pentru acest caz particular în care $h_x = h_y = 1$, valorile proprii ale matricei A sunt:

$$\lambda_{ij} = 4\sin^2\left(\frac{i\pi}{2(n_1+1)}\right) + 4\sin^2\left(\frac{j\pi}{2(n_2+1)}\right).$$

Figura 2.4.4 conține o reprezentare grafică a valorilor proprii ale matricei pentru $n_1 = 21$ şi $n_2 = 11$. Din corolarul 2.4.2 observăm că cea mai mare valoare proprie este $\lambda_{\max} = 8$ şi cea mai mică este $\lambda_{\min} = 8\sin^2(\pi/2)$. Deci, A este pozitiv definită cu numărul de condiționare

$$\frac{\lambda_{\max}}{\lambda_{\min}} \approx \frac{4}{\pi^2}.$$

Din acest exemplu vedem că cu cât sunt mai puține blocuri pe diagonala principală, cu atât viteza de convergența este mai mare, în sensul că eroarea $\|b - Ax_k\|_2$ se reduce cel mai rapid.

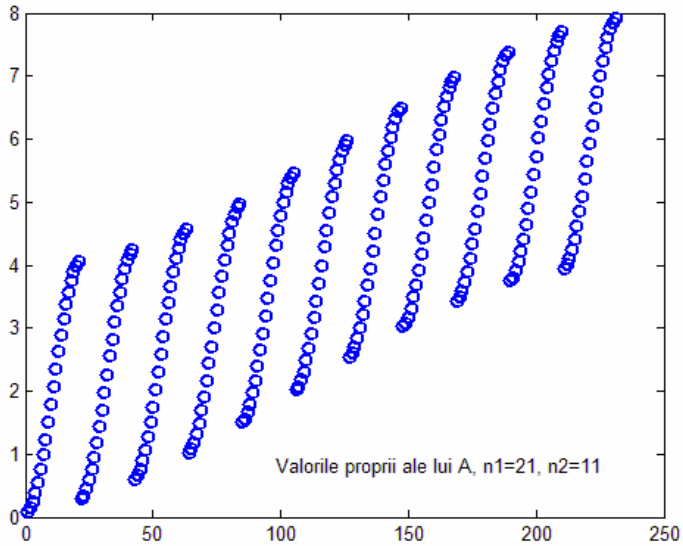


Fig. 2.4.4. Valorile proprii ale lui A.

În continuare să prezentăm o aplicație a dezvoltărilor teoretice de mai sus în ceea ce privește ecuația căldurii și a calculului distribuției de temperatură într-o placă dreptunghiulară cu mai multe surse de căldură.

Calculul temperaturii staționare într-o placă dreptunghiulară

Pentru început să prezentăm modul cum se obține ecuația de difuziune a căldurii, după care vom prezenta problema calculului temperaturii staționare într-o placă dreptunghiulară cu mai multe surse de căldură.

Ecuațiile de transfer a căldurii. Prima lege a termodinamicii are forma:

$$Q = W + \frac{\mathrm{d}U}{\mathrm{d}t}, \tag{2.4.22}$$

unde Q rata de transfer a căldurii, W este lucrul mecanic şi U este energia termică internă a sistemului, aşa după cum se vede în figura 1. $\mathrm{d}U / \mathrm{d}t$ este rata de schimbare în timp a energiei termice interne. În general, analiza procesului de transfer a căldurii se poate face fără a ne referi la lucrul mecanic făcut de sistem. Totuşi, în analiza sistemelor reale, transferul de căldură trebuie combinat cu lucrul mecanic.

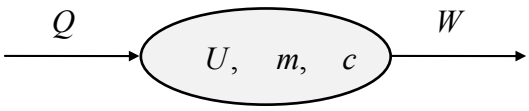


Fig. 2.4.5. Prima lege a termodinamicii pentru un sistem închis.

Când sistemul nu transferă lucru mecanic, atunci prima lege a termodinamicii este:

$$Q = \frac{\mathrm{d}U}{\mathrm{d}t} = mc\frac{\mathrm{d}T}{\mathrm{d}t}, \tag{2.4.23}$$

unde m este masa sistemului, c este căldura specifică şi T este temperatura sistemului. Ecuaţia de transfer a căldurii se scrie conform *legii lui Fourier*. Joseph Fourier a publicat lucrarea sa remarcabilă: *Théorie Analytique de la Chaleur* în 1822, în care a formulat o expunere foarte completă a teoriei conducţiei căldurii. Această lege empirică este următoarea:

Fluxul de căldură q rezultat din conducţia termică este proporţional cu mărimea gradientului temperaturii cu semn schimbat:

$$q = -V\nabla T,$$

unde V este conductivitatea termică.

Pe componente, legea lui Fourier este:

$$q_x = -V\frac{\mathrm{d}T}{\mathrm{d}x}, \quad q_y = -V\frac{\mathrm{d}T}{\mathrm{d}y}, \quad q_z = -V\frac{\mathrm{d}T}{\mathrm{d}z}.$$

În general, conductivitatea termică V este o funcţie de punct şi temperatură. Totuşi, majoritatea materialelor sunt omogene şi deci putem considera $V = V(T)$.

Cu acestea, să considerăm un volum R mărginit de o suprafaţă S şi un mic element diferenţial $\mathrm{d}S$, centrat într-un punct de pe suprafaţa S. Referindu-ne la elementul diferenţial $\mathrm{d}S$, se pot asocia doi vectori: primul este vectorul n normal la suprafaţa S, şi al doilea $q = -V\nabla T$ fluxul de căldură prin suprafaţa S. În acelaşi timp putem considera că în volumul R este distribuită o sursă (volumetrică) de căldură f. Aceasta poate rezulta, de exemplu, din anumite reacţii chimice, nucleare sau electrice. Ca atare, căldura care *iese* prin elementul de suprafaţă $\mathrm{d}S$ este:

$$(-V\nabla T).(n\mathrm{d}S). \tag{2.4.24}$$

Considerând acum şi căldura generată (sau consumată) în volumul R, atunci căldura totală care curge *prin* S în volumul R este:

$$Q = -\int_S (-V\nabla T).(ndS) + \int_R fdR. \tag{2.4.25}$$

Dar, rata de creştere a energiei interne în volumul R este:

$$\frac{\mathrm{d}U}{\mathrm{d}t} = \int_R \left(\rho c\frac{\partial T}{\partial t}\right)\mathrm{d}R, \tag{2.4.26}$$

unde ρ este densitatea substanţei din volumul R. Cu acestea, din (2.4.23) obţinem imediat:

$$\int_S V\nabla T.n\mathrm{d}S = \int_R \left[\rho c\frac{\partial T}{\partial t} - f\right]\mathrm{d}R. \tag{2.4.27}$$

Aplicând formula Gauss-Ostrogradski, obţinem:

$$\int_R \left(\nabla . V \nabla T - \rho c \frac{\partial T}{\partial t} + f \right) dR = 0. \qquad (2.4.28)$$

Deoarece volumul R este arbitrar, rezultă:

$$\rho c \frac{\partial T}{\partial t} = \nabla . V \nabla T + f, \qquad (2.4.29)$$

cunoscută ca *ecuaţia căldurii*.

Având în vedere discuţiile de mai sus, această ecuaţie descrie difuzia căldurii în medii incompresibile (nu am inclus lucrul mecanic efectuat de sistem) şi fără convecţie (mediul nu execută nici o mişcare relativă).

Pentru rezolvarea acestei ecuaţii, adică pentru determinarea difuziei de căldură în medii trebuie precizate condiţiile în care se execută aceste transport. Aceasta revine la a preciza informaţii asupra conductivităţii materialului prin care se face difuzia de căldură, a densităţii şi căldurii specifice a cestuia, precum şi a condiţiilor pe frontiera domeniului în care se înregistrează difuzia de căldură. Cu acestea rezolvarea ecuaţiei căldurii se face prin discretizarea domeniului şi deci calculul temperaturii în punctele reţelei de discretizare. Se pot considera mai multe situaţii pe care le vom considera în cele ce urmează.

Calculul temperaturii staţionare într-o placă dreptunghiulară. În cazul staţionar caracteristicile problemei nu se schimbă în timp. Vom considera o placă dreptunghiulară pentru care conductivitatea termină V [$Wm^{-1}K^{-1}$] este eterogenă, simetric distribuită de-a lungul plăcii, ca în figura 2.4.6. Atunci ecuaţia de transport a căldurii devine:

$$\nabla . V \nabla T + f = 0. \qquad (2.4.30)$$

Pentru rezolvarea ecuaţiei (2.4.30) vom aplica operatorul diferenţial ∇ atât conductivităţii termice cât şi gradientului temperaturii obţinând următoarea expresie a ecuaţiei căldurii:

$$V \nabla . \nabla T + \nabla T . \nabla V + f = 0. \qquad (2.4.31)$$

Utilizând aproximaţiile prin diferenţe finite, în două dimensiuni, obţinem sistemul:

$$V_{i,j} \frac{(T_{i+1,j} - 2T_{i,j} + T_{i-1,j})}{(\Delta x)^2} + V_{i,j} \frac{(T_{i,j+1} - 2T_{i,j} + T_{i,j-1})}{(\Delta y)^2}$$
$$+ \frac{(V_{i+1,j} - V_{i-1,j})}{2(\Delta x)} \frac{(T_{i+1,j} - T_{i-1,j})}{2(\Delta x)} + \frac{(V_{i,j+1} - V_{i,j-1})}{2(\Delta y)} \frac{(T_{i,j+1} - T_{i,j-1})}{2(\Delta y)} + f_{i,j} = 0,$$

pentru $i = 1, \ldots, n_x$ şi $j = 1, \ldots, n_y$, unde n_x şi respectiv n_y sunt numărul punctelor de discretizare de-a lungul laturilor plăcii.

Pentru determinarea distribuţiei temperaturii în nodurile acestei reţele (v. fig.2.4.6) trebuie să rezolvăm acest sistem de ecuaţii algebrice la care se adaugă condiţiile pe frontiera plăcii.

Vom considera două situaţii. În prima este vorba de cazul realist în care pe frontiera plăcii se aplică două surse calde pe două laturi ale plăcii. În al doilea,

suplimentar se aplică o sursă caldă într-un anumit punct al plăcii. În figura 2.4.6 se prezintă ambele aceste situații. Vedem că placa are o conductivitate termină distribuită simetric în placă cu valorile 0,5 $Wm^{-1}K^{-1}$, respectiv 1,5 $Wm^{-1}K^{-1}$. Pentru ilustrarea conductivității termice a materialului, precum şi a distribuției temperaturii în placă am considerat o discretizare foarte grosieră. In situațiile concrete de obicei se consideră discretizări foarte fine.

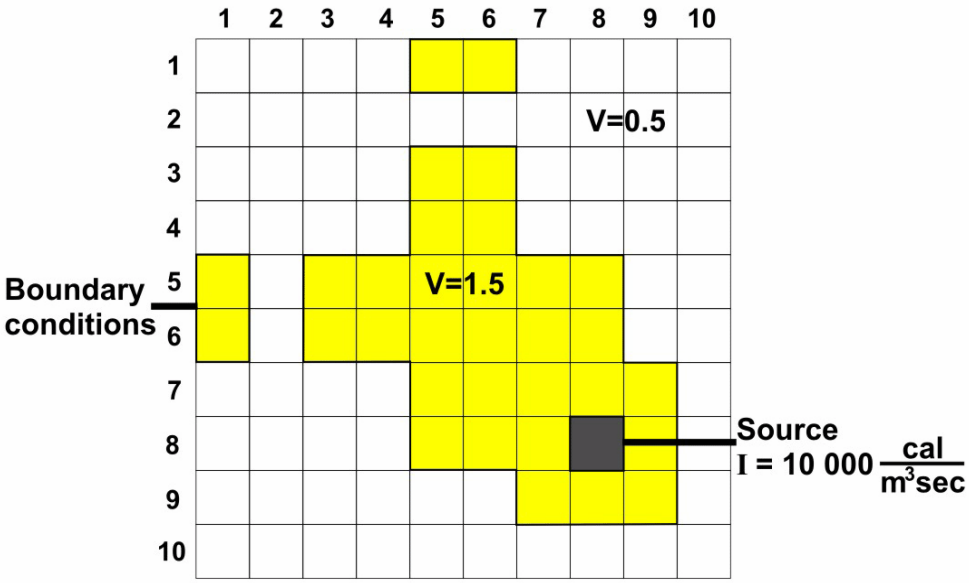


Fig. 2.4.6. Placa şi discretizarea ei pentru calculul transferului căldurii.
Sursele de căldură sunt plasate pe frontiera plăcii în primul caz.
În al doilea caz, suplimentar, apare o sursă de căldură.

Considerăm cazul în care pe frontiera plăcii se menține o temperatură constantă nulă cu excepția celulelor (I1,J5), (I1,J6), (I5,J1) şi (I6,J1) în care temperatura are 100 grade Celsius. Dacă $n_x = 10$ şi $n_y = 10$, ca în figura 2.4.6, atunci soluția sistemului discretizat de mai sus este:

I1	0.00	0.00	0.00	0.00	100.00	100.00	0.00	0.00	0.00	0.00
I2	0.00	3.95	7.91	14.63	32.47	32.02	13.52	6.17	2.52	0.00
I3	0.00	7.91	13.03	18.16	22.14	21.40	15.88	8.65	3.91	0.00
I4	0.00	14.63	18.16	18.30	18.34	17.01	13.58	8.64	4.45	0.00
I5	100.00	32.47	22.14	18.34	16.14	13.93	11.02	8.14	5.27	0.00
I6	100.00	32.02	21.40	17.01	13.93	11.56	9.18	6.94	4.41	0.00
I7	0.00	13.52	15.88	13.58	11.02	9.18	7.20	5.23	3.07	0.00
I8	0.00	6.17	8.65	8.64	8.14	6.94	5.23	3.69	2.16	0.00
I9	0.00	2.52	3.91	4.45	5.27	4.41	3.07	2.16	1.26	0.00
I10	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00

În figura 2.4.7 se arată suprafața temperaturii care se realizează în placă. Figura 2.4.8 arată curbele de temperatură constantă din placă.

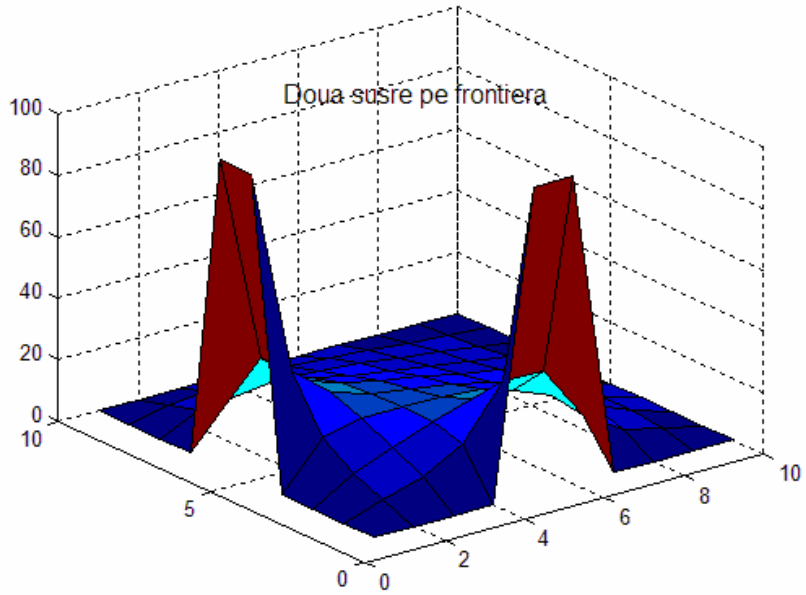


Fig. 2.4.7. Soluția ecuației căldurii.
Suprafața temperaturii pe o placă cu două surse de căldură pe frontieră.

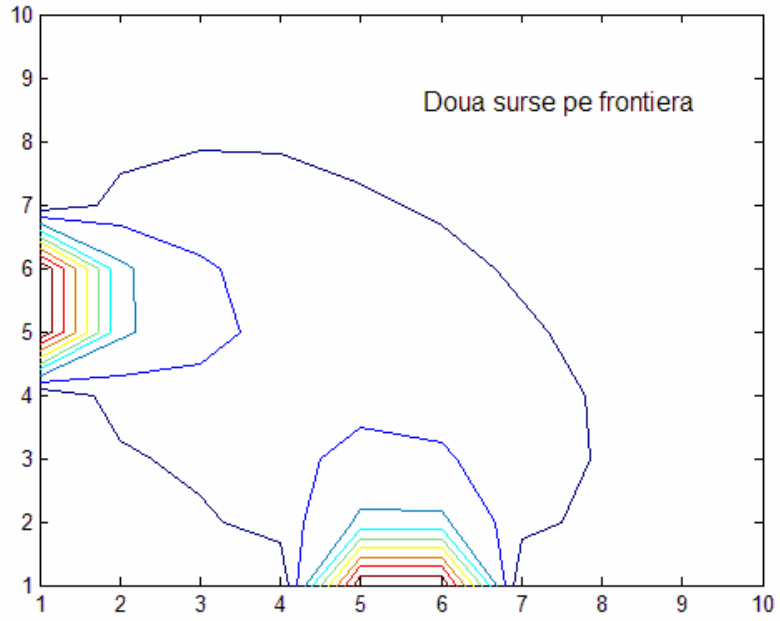


Fig. 2.4.8. Soluția ecuației căldurii.
Curbele de temperatură constantă pe o placă cu două surse de căldură pe frontieră.

Considerând acum situația în care asupra plăcii se aplică suplimentar o sursă de căldură plasată în centrul celulei (I3,J3), atunci distribuția temperaturii de-a lungul plăcii se realizează ca în figura 2.4.9. În acest caz curbele de temperatură constantă sunt cele din figura 2.4.10.

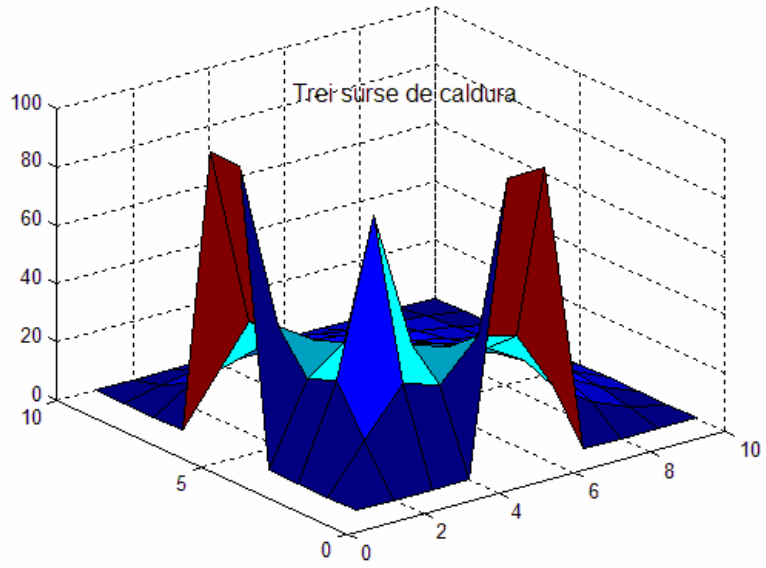


Fig. 2.4.9. Soluția ecuației căldurii.
Suprafața temperaturii pe o placă cu trei surse de căldură.

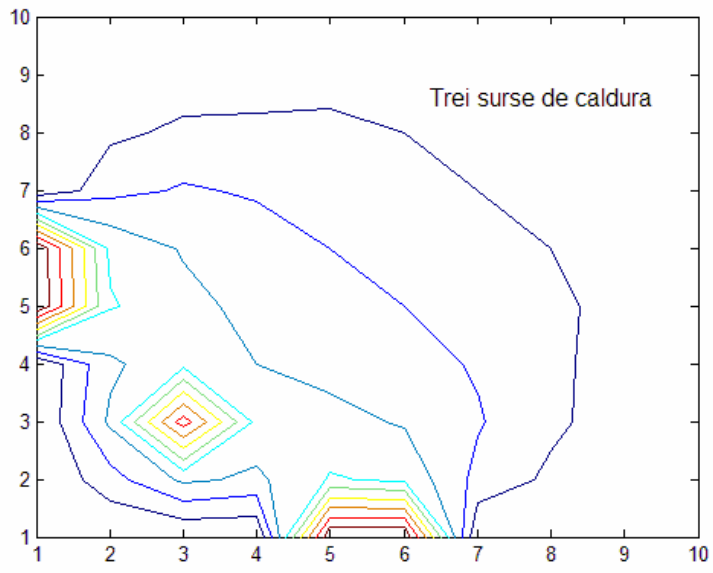


Fig. 2.4.10. Soluția ecuației căldurii.
Curbele de temperatură constantă pe o placă cu trei surse de căldură.

Se observă că simetria distribuţiei conductivităţii termice şi a condiţiilor pe frontieră determină simetria soluţiei ecuaţiei căldurii.

Referinţe

Andrei, N., (1999a) *Programarea Matematică Avansată. Teorie, metode Computaţionale, Aplicaţii.* Editura Tehnică, Bucureşti, 1999.

Andrei, N., (1999b) *Programarea Matematică.* Editura Tehnică, Bucureşti, 1999.

Andrei, N., (2000) *Optimizare fără Restricţii. Metode de Direcţii Conjugate*. Editura MatrixRom, Bucureşti, 2000.

Andrei, N., (2008) *Critica Raţiunii Algoritmilor de Optimizare fără Restricţii*. Editura Academiei Române, Bucureşti. (sub tipar)

Axelsson, O., (1976) *A class of iterative methods for finite element equations*. Com. Meth. Appl. Mech. Eng., vol.9, pp.123-137. 1976

Barrett, R., Berry, M., Chan, T., Demmel, J., Donato, J., Dongarra, J., Eijkhout, V., Pozo, R., Romine, C., van der Vost, H., (1994) *Templates for the Solution of Linear Systems: Building Blocks for Iterative Methods.* SIAM, Philadelphia, PA, 1994.

Cauchy, A., (1847) *Méthodes générales pour la résolution des systémes déquations simultanées.* C.R. Acad. Sci. Par., vol. 25, (1848), pp. 536-538.

Concus, P., Golub, G.H., (1976) *A generalized conjugate gradient method for nonsymmetric systems of linear equations.* Technical report STAN-CS-76-535, Stanford University, Stanford, CA, 1976.

Demmel, J.W., (1997) *Applied Numerical Linear Algebra.* SIAM, 1997.

Dongarra, J., Duff, I.S., Sorensen, D.C., van der Vost, H., (1998) *Numerical Linear Algebra for High-Performance Computers.* SIAM, Philadelphia, 1998.

Faber, V., Manteuffel, T., (1984) *Necessary and sufficient conditions for the existence of a conjugate gradient method*. SIAM J. Numer. Anal., 21 (1984), pp.352-362.

Greenbaum, A., (1997) *Iterative Methods for Solving Linear Systems.* SIAM – Society for Industrial and Applied Mathematics, Frontiers in Applied Mathematics, Philadelphia.

Greenbaum, A., Strakos, Z., (1992) *Predicting the behavior of finite precision Lanczos and conjugate gradient computations.* SIAM J. Matrix Anal. Appl., 13: 121-137, 1992.

Hackbusch, W., (1994) *Iterative solution of large linear systems of equations.* Springer Verlag, NY, 1994.

Hestenes, M.R., Stiefel, E., (1952) *Methods of conjugate gradients for solving linear systems*, J. Research Nat. Bur. Standards Sec. B. 48 (1952) 409-436.

Meurant, G., (1999) *Computer Solution of Large Linear Systems.* Elsevier, Amsterdam, 1999.

Naiman, A.E., Babuska, I.M. şi H.C. Elman, (1997) *A note on conjugate gradient convergence*. Numerischke Mathematik, 76: 209-230, 1997.

Paige, C.C., (1971) *The computation of eigenvalues and egienvectors of very large sparse matrices.* PhD Thesis, University of London, London, England, 1971.

Paige, C.C., (1972) *Computational variants of the Lanczos method for the eigenproblem.* J. Inst. Maths. Applics., 10 (1972), pp.373-381.

Paige, C.C., (1976) *Error analysis of the Lanczos algorithm for tridiagonalizing a symmetric matrix.* J. Inst. Maths. Applics., 18 (1976), pp.341-349.

Paige, C.C., Saunders, M.A., (1975) *Solution of sparse indefinite systems of linear equations.* SIAM J. Numerical Analysis, 12 (1975), pp.617-629.

Reid, J.K., (1971) *On the method of conjugate gradients for the solution of large sparse systems of linear equations.* In J.K. Reid (Ed.) „Large Sparse Sets of Linear Equations", pages 231-254, Academic Press, 1971.

Reid, J.K., (1972) *The use of conjugate gradients for systems of linear equations possessing "Property A".* SIAM J. Numer. Anal., 9 (1972), pp.325-332.

Saad, Y., (1996) *Iterative Methods for Sparse Linear Systems*. PWS Publishing Company. 1996.

Voevodin, V.V., (1983) *The problem of non-selfadjoint generalization of the conjugate gradient method has been closed.* USSR Comput. Math. Math. Phys., 23 (1983), pp 143-144.

March 31, 2008