

## ALGORITMI EFICIENȚI PENTRU REZOLVAREA SISTEMELOR DE ECUAȚII CE APAR LA DISCRETIZAREA ECUAȚIILOR INTEGRALE SINGULARE

*Maria CAPCELEA, Titu CAPCELEA, Alexandru LAZARI*

*Catedra Matematică Aplicată*

It is marked out the class of iterative algorithms for solving of systems of equations which are obtained when discretizing SIE with discontinuous coefficients. These algorithms permit essential reducing of computational cost for finding an approximate solution, at that not losing in the quality of other numerical characteristics - they are numerically stable and allow us to check precision in the course of iterations without calculation of approximation of a solution.

În ultimii ani prezintă interes sporit problema elaborării algoritmilor eficienți pentru rezolvarea sistemelor de ecuații ce apar la rezolvarea aproximativă a ecuațiilor integrale singulare (EIS). Astfel, la aplicarea metodei de trunchiere la EIS

$$(\Lambda\varphi \equiv) a_0(t)\varphi(t) + \frac{b_0(t)}{\pi i} \int_{\Gamma_0} \frac{\varphi(\tau)}{\tau-t} d\tau + \frac{1}{2\pi i} \int_{\Gamma_0} h(t, \tau)\varphi(\tau) d\tau = f(t), \quad t \in \Gamma_0, \quad (1)$$

unde  $\Gamma_0 = \{t \in \square : |t|=1\}$ ,  $a_0, b_0 \in PC(\Gamma_0)$ ,  $h \in C(\Gamma_0 \times \Gamma_0)$ ,  $f \in L_2(\Gamma_0)$ , apar sisteme de ecuații algebrice liniare (SEAL) cu matrici de tip Töeplitz, pentru a căror soluționare au fost elaborați algoritmi de complexitate  $O(n \log_2^2 n)$  (*a se vedea* [1-4]). Eficacitatea se atinge datorită faptului că algoritmi utilizează generalizări ale formulei Gohberg-Semențul *a se vedea* [1,5], ce exprimă inversa  $A^{-1}$  a matricei sistemului ca sumă de produse de matrici Töeplitz inferior și superior triunghiulare, iar aceasta din urmă permite să se calculeze efectiv soluția  $A^{-1}f$  a sistemului, utilizând algoritmi transformării rapide Fourier (*a se vedea* [6]).

În prezenta lucrare se va evidenția o clasă de algoritmi iterativi efectivi pentru rezolvarea SEAL ce apar la discretizarea EIS (1) conform metodei cuadraturilor mecanice:

$$a(t_j) \sum_{k=0}^n t_j^k \alpha_k + b(t_j) \sum_{k=-n}^{-1} t_j^k \alpha_k + \frac{1}{2n+1} \sum_{k=-n}^n \sum_{s=-n}^n h(t_j, t_s) t_s^{k+1} \alpha_k = f(t_j), \quad j = \overline{-n, n}, \quad (2)$$

unde  $a(t) = a_0(t) + b_0(t)$ ,  $b(t) = a_0(t) - b_0(t)$ ,  $t_j = t_j^{(n)} = \exp(2\pi i j / (2n+1))$ ,  $j = \overline{-n, n}$ .

Relația (2) reprezintă un sistem din  $2n+1$  ecuații algebrice liniare cu  $2n+1$  necunoscute  $\alpha_k$  ( $k = \overline{-n, n}$ ).

În forma vectorială sistemul (2) va fi

$$(A\bar{\alpha} \equiv) (F+H)\bar{\alpha} = \bar{f}, \quad (3)$$

unde  $F = \{F_{jk}\}_{j,k=-n}^n$ ,  $F_{jk} = b(t_j)t_j^k$  pentru  $j = \overline{-n, n}$ ,  $k = \overline{-n, -1}$ , și  $F_{jk} = a(t_j)t_j^k$  pentru  $j = \overline{-n, n}$ ,  $k = \overline{0, n}$ ,

H este matricea  $\{H_{jk}\}_{j,k=-n}^n$ , ale cărei elemente sunt de forma  $H_{jk} = (2n+1)^{-1} \sum_{s=-n}^n h(t_j, t_s) t_s^{k+1}$ , iar

$$\bar{\alpha} = (\alpha_{-n}, \dots, \alpha_{-1}, \alpha_0, \alpha_1, \dots, \alpha_n)^T, \quad \bar{f} = (f(t_{-n}), \dots, f(t_{-1}), f(t_0), f(t_1), \dots, f(t_n))^T.$$

Sistemul complex  $A\bar{\alpha} = \bar{f}$  de ordinul  $N = 2n+1$  este echivalent unui sistem de ordinul  $2N$  cu componente reale. Într-adevăr, dacă  $A = A_{re} + iA_{im}$  ( $A_{re}, A_{im} \in M_{N,N}(\square)$ ) și  $\bar{\alpha} = \bar{\alpha}_{re} + i\bar{\alpha}_{im}$ ,  $\bar{f} = \bar{f}_{re} + i\bar{f}_{im}$ ,  $\bar{\alpha}_{re}, \bar{\alpha}_{im}, \bar{f}_{re}, \bar{f}_{im} \in \square^N$ , atunci sistemul (3) este echivalent sistemului  $(A_{re} + iA_{im}) \cdot (\bar{\alpha}_{re} + i\bar{\alpha}_{im}) = \bar{f}_{re} + i\bar{f}_{im}$ . Prin egalarea părților reale și imaginare din ambii membri se obține sistemul

$$\begin{bmatrix} A_{re} & -A_{im} \\ A_{im} & A_{re} \end{bmatrix} \cdot \begin{bmatrix} \bar{\alpha}_{re} \\ \bar{\alpha}_{im} \end{bmatrix} = \begin{bmatrix} \bar{f}_{re} \\ \bar{f}_{im} \end{bmatrix}. \quad (4)$$

Acesta este un sistem cu componente reale de ordinul  $2N$  echivalent sistemului  $A\bar{\alpha} = \bar{f}$ .

Cu toate că matricele sistemelor menționate nu sunt rare, este totuși posibil să se construiască algoritmi efectivi pentru rezolvarea lor ce au complexitatea  $O(mn \log_2 n)$  ( $m \leq n$ ), exploatând proprietățile structurale ale părții caracteristice a operatorului integral singular ce intră în componența ecuației inițiale.

În cazul în care este necesar să se obțină aproximația soluției EIS cu o precizie înaltă, sistemele obținute la aplicarea metodelor direct-aproximative (MDA) pot avea dimensiuni mari. Și atunci aplicarea metodelor directe (Gauss, Jordan, rotațiilor etc.) este inutilă, deoarece procesul de obținere a soluției SEAL implică  $O(n^3)$  operații aritmetice ( $n$  – dimensiunea sistemului) și un volum de  $O(n^2)$  unități de memorie, iar acestea pot fi numere inadmisibil de mari. Aplicarea celor mai simple metode iterative implică probleme adăugătoare ce țin de studiul convergenței lor, deoarece sistemele menționate, în general, nu sunt simetrice sau pozitiv definite și nu posedă proprietăți speciale, cum ar fi, de exemplu, proprietatea de predominanță diagonală.

Ideea de bază ce permite în mod practic să se reducă esențial resursele de calcul utilizate în calculul soluției SEAL, fără a înrăutăți alte caracteristici numerice, ține de utilizarea algoritmilor iterativi în baza metodelor proiecționale și, în special, a subclasei lor de metode ce țin de proiectarea pe subspațiile Krylov.

Să considerăm SEAL

$$Ax = b, \quad (5)$$

unde  $A \in \mathbb{R}^{n \times n}$ ,  $\det A \neq 0$ ,  $b \in \mathbb{R}^n$ ,  $x \in \mathbb{R}^n$ . Deoarece matricea  $A$  este nedegenerată, sistemul (5) are soluție unică. Fie că se cunoaște careva aproximație inițială  $x^{(0)} \in \mathbb{R}^n$  a soluției sistemului. Fie că dispunem de un cortegiu de perechi de subspații  $\langle K_m, L_m \rangle_{m=1}^n$ ,  $K_m, L_m \subseteq \mathbb{R}^n$ ,  $m = \overline{1, n}$ , astfel încât  $\dim K_m = \dim L_m = m$ , iar  $x^{(0)} + K_m$  ( $m = \overline{1, n}$ ) sunt varietățile liniare generate de vectorul  $x^{(0)}$  și subspațiile  $K_m$ .

Metodă proiecțională pentru rezolvarea SEAL (5) este procesul iterativ, conform căruia la pasul  $m$  ( $1 \leq m \leq n$ ) se găsește aproximația  $x^{(m)} \in x^{(0)} + K_m$ , impunând condiția Petrov-Galerkin  $r^{(m)} := b - Ax^{(m)} \perp L_m$ .

Pentru  $x^{(m)} \in x^{(0)} + K_m$  putem scrie  $x^{(m)} = x^{(0)} + \delta^{(m)}$ ,  $\delta^{(m)} \in K_m$  și, dacă  $r^{(0)} = b - Ax^{(0)}$  este vectorul residuu inițial, atunci avem  $r^{(m)} = r^{(0)} - A\delta^{(m)} \in r^{(0)} + AK_m$  și  $r^{(0)} - A\delta^{(m)} \perp L_m$ .

Fie că  $\{v^{(j)}\}_{j=1}^m$  și  $\{w^{(j)}\}_{j=1}^m$  sunt baze ale subspațiilor  $K_m$  și  $L_m$ , respectiv. Ușor observăm că ultima relație de ortogonalitate are loc atunci și numai atunci, când

$$(r^{(0)} - A\delta^{(m)}, w^{(j)}) = 0, \quad j = \overline{1, m}, \quad 1 \leq m \leq n. \quad (6)$$

La introducerea pentru baze a notațiilor matriceale  $V_m = [v^{(1)} | v^{(2)} | \dots | v^{(m)}]$  și  $W_m = [w^{(1)} | w^{(2)} | \dots | w^{(m)}]$  avem  $\delta^{(m)} = V_m y^{(m)}$ ,  $y^{(m)} \in \mathbb{R}^m$ . Atunci, relația (6) poate fi scrisă sub forma  $W_m^T (r^{(0)} - AV_m y^{(m)}) = 0$ , de unde obținem  $y^{(m)} = (W_m^T AV_m)^{-1} W_m^T r^{(0)}$ , și atunci aproximația la pasul  $m$  se va preciza conform formulei

$$x^{(m)} = x^{(0)} + V_m (W_m^T AV_m)^{-1} W_m^T r^{(0)}. \quad (7)$$

Soluția aproximativă este definită doar când matricea  $W_m^T AV_m$  este nedegenerată. Evident, la construcția metodelor proiecționale subspațiile  $K_m$  și  $L_m$  și bazele lor trebuie să se aleagă astfel, încât matricea  $W_m^T AV_m$  sau să fie de dimensiuni mici, sau să aibă structură simplă, comodă pentru inversare.

Metodă a subspațiilor Krylov (*a se vedea* [7]) este o metodă proiecțională în care la pasul  $m$  în calitate de subspațiu  $K_m$  se alege subspațiul Krylov de dimensiune  $m$  definit de matricea  $A$  și vectorul residuu inițial  $r^{(0)}$ :  $K_m = K_m(A, r^{(0)}) := \text{span}\{r^{(0)}, Ar^{(0)}, \dots, A^{m-1}r^{(0)}\}$ . Aproximațiile ce se obțin conform metodei subspațiilor Krylov sunt de forma  $x^{(m)} = x^{(0)} + q_{m-1}(A)r^{(0)}$ , unde  $q_{m-1}$  este polinom de grad ce nu întrece  $m-1$ . Diferiți algoritmi ai metodei subspațiilor Krylov sunt generați în funcție de modalitatea de alegere a subspațiilor  $L_m$ , precum și de modalitatea de construire a bazelor subspațiilor. Ținând cont de proprietatea  $K_m \subseteq K_{m+1}$  a

subspațiilor Krylov și de faptul că în algoritmi considerați se construiește bază, devine evident, că aceștia converg (dacă calculele se fac exact) în cel mult  $n$  iterații, unde  $n$  este dimensiunea sistemului.

Orice algoritm al metodei subspațiilor Krylov include două etape:

1. Construirea bazei ortonormate a subspațiului Krylov  $K_m$ ;
2. Calculul corecției  $y^{(m)}$  și al aproximației curente  $x^{(m)}$ .

Baza evidentă  $r^{(0)}, Ar^{(0)}, \dots, A^{m-1}r^{(0)}$  a subspațiului  $K_m(A; r^{(0)})$  nu este atractivă din punct de vedere numeric, deoarece vectorii  $A^k r^{(0)}$ ,  $k = 0, 1, 2, \dots$  tind către vectorul propriu dominant (conform metodei puterilor), și, prin urmare, vectorii bazei devin liniar dependenți în aritmetică cu precizie finită. De aceea, în locul bazei standard, în algoritmi metodei subspațiilor Krylov se formează o bază ortonormată, utilizând în acest scop metoda Arnoldi. La baza metodei Arnoldi stă procedura de ortogonalizare Gram-Schmidt modificată (a se vedea [7]), ce se aplică vectorilor  $v^{(1)}, Av^{(1)}, \dots, A^{m-1}v^{(1)}$ , unde  $v^{(1)} = r^{(0)} / \|r^{(0)}\|_2$ . Presupunem că a fost construită baza ortonormată  $V_m = [v^{(1)} | v^{(2)} | \dots | v^{(m)}]$  ( $m \geq 1$ ) a spațiului  $K_m(A; r^{(0)})$ . Se calculează produsul  $AV^{(m)}$ ; se ortogonalizează vectorul rezultat în raport cu fiecare dintre vectorii  $v^{(1)}, \dots, v^{(m)}$  deja calculați și apoi se normalizează. Vectorul obținut  $v^{(m+1)}$  (posibil, este vector nul) semnifică în reprezentare matriceală extinderea bazei  $V_m$  cu un vector-coloană adăugător. Ușor se verifică că  $V_{m+1} = [v^{(1)} | \dots | v^{(m)} | v^{(m+1)}]$  formează o bază ortonormată a spațiului  $K_{m+1}(A; r^{(0)})$ . Procesul de ortogonalizare poate fi descris algebric prin relația

$$h_{k+1,k} v^{(k+1)} = Av^{(k)} - \sum_{j=1}^k h_{jk} v^{(j)}. \quad (8)$$

Coefficienții ortogonalizării  $h_{jk}$  pot fi reuniți sub forma unei matrice, completând în ea pozițiile libere cu zerouri. Coeficientul  $h_{m+1,m}$  corespunzător vectorului  $v^{(m+1)}$  semnifică extinderea matricei cu o linie adăugătoare (posibil, să fie linie cu toate elementele zerouri). Fie  $\bar{H}_m$  este matricea coeficienților  $h_{jk}$ , completată în ultima linie cu ajutorul lui  $h_{m+1,m}$ , iar  $H_m$  – aceeași matrice fără ultima linie și care are dimensiunea  $m \times m$ . Atunci, conform metodei Arnoldi și din (8) rezultă că matricea  $H_m$  este o matrice în forma Hessenberg superioară și pentru ea sunt adevărate relațiile

$$AV_m = V_{m+1} \bar{H}_m = V_m H_m + h_{m+1,m} v^{(m+1)} e_m^T, \quad (9)$$

$$V_m^T AV_m = H_m. \quad (10)$$

În afară de aceasta, drept urmare a faptului că baza  $\{v^{(j)}\}$  este ortonormată, are loc egalitatea

$$V_m^T v^{(k)} = e_k, \quad k = \overline{1, m}. \quad (11)$$

Cei mai cunoscuți algoritmi ai metodei subspațiilor Krylov pentru matrici dense, fără structură specială, sunt metodele ortogonalizării complete (MOC), residuurilor minimale generalizate (MGRM) și metoda gradientilor conjugați pentru ecuația normală (MGCEN).

Acești algoritmi posedă următoarele proprietăți importante:

- ✓ Se aplică sistemelor cu matrici dense și structură arbitrară;
- ✓ Sunt stabili numeric datorită tehnicii de ortogonalizare utilizate;
- ✓ Permit să se controleze exactitatea pe parcursul iterațiilor (fără a se calcula aproximația soluției).

### Metoda ortogonalizării complete (MOC)

La fiecare pas  $m$  se alege  $K_m = L_m = K_m(A, v^{(1)})$ , unde  $v^{(1)} = r^{(0)} / \beta$ ,  $\beta = \|r^{(0)}\|_2$ . În acest caz, pentru baze avem  $V_m = W_m$ , și, ținând cont de relațiile (10), obținem  $W_m^T AV_m = H_m$ . SEAL cu matricea coeficienților  $H_m$  se reduce cu ajutorul transformărilor de eliminare Gauss la un sistem cu matrice superior triunghiulară,

iar ultimul se rezolvă prin metoda parcursului invers. Deoarece  $r^{(0)} = \beta v^{(1)}$ , în virtutea relației (11) avem  $W_m^T r^{(0)} = \beta e_1$ . Astfel, formula (7) se reduce la

$$x^{(m)} = x^{(0)} + V_m y^{(m)}, \quad y^{(m)} = H_m^{-1}(\beta e_1). \quad (12)$$

Condiția de stopare a calculelor este  $\|r^{(m)}\|_2 < \varepsilon \|r^{(0)}\|_2$ . Datorită relației  $\|r^{(m)}\|_2 = h_{m+1,m} |y_m^{(m)}|$ , la fiecare iterație a metodei MOC se poate evalua descreșterea normei residuului  $\|r^{(m)}\|_2 = \|b - Ax^{(m)}\|_2$  fără a găsi în mod explicit soluția aproximativă. Vom prezenta algoritmul MOC care, în baza relației (12) și a metodei Arnoldi, necesară pentru construirea bazei  $V_m$ , descrie procesul de obținere a soluției SEAL conform metodei descrise:

### Algoritmul MOC

Se alege aproximația inițială  $x^{(0)} \in \mathbb{R}^n$

$$r^{(0)} := b - Ax^{(0)}; \beta := \|r^{(0)}\|_2; v^{(1)} := r^{(0)}/\beta$$

Se alocă memorie pentru vectorul  $f^{(0)} \in \mathbb{R}^2$ ; Se pune  $f^{(0)} = \bar{0}$ ;  $f_1^{(0)} := \beta$ ;  $m := 0$

do

$$m := m + 1$$

Se alocă memorie pentru vectorul  $h^{(m)} \in \mathbb{R}^{m+1}$ ; Se pune  $h^{(m)} = \bar{0}$ ;

Se alocă memorie pentru vectorul  $f^{(m)} \in \mathbb{R}^{m+1}$ ; Se pune  $f^{(m)} = \bar{0}$ ;

For  $i = \overline{1, m}$  do

$$\left\{ \begin{array}{l} f_i^{(m)} := f_i^{(m-1)} \end{array} \right\}$$

Se eliberează memoria alocată pentru  $f^{(m-1)}$

$$\omega^{(m)} := Av^{(m)}$$

for  $i = \overline{1, m}$  do

$$\left\{ \begin{array}{l} h_{i,m} := (v^{(i)})^T \omega^{(m)} \\ \omega^{(m)} := \omega^{(m)} - h_{i,m} v^{(i)} \end{array} \right\}$$

$$h_{m+1,m} := \|\omega^{(m)}\|_2; c := h_{m+1,m}/\beta$$

$$v^{(m+1)} := \omega^{(m)}/h_{m+1,m}$$

if  $h_{m,m} \neq 0$  then

$$\left\{ \begin{array}{l} t_m := h_{m+1,m}/h_{m,m} \\ h_{m+1,m} := 0 \\ f_{m+1} := f_{m+1} - t_m f_m \end{array} \right\}$$

for  $i = \overline{2, m}$

$$\left\{ \begin{array}{l} h_{i,m} := h_{i,m} - t_{i-1} h_{i-1,m} \end{array} \right\}$$

else

$$\left\{ \right.$$

Se afișează mesajul „metoda Gauss nu poate fi aplicată la triunghiularizarea sistemului”.

```

Ieșire din program
    }
     $y_m^{(m)} := f_m / h_{m,m}$ 
     $\rho := c |y_m^{(m)}|$ 
while  $\rho \geq \varepsilon$ 
for  $k = \overline{m-1, 1}$ 
    {
     $y_k^{(m)} := \left( f_k - \sum_{i=k+1}^m h_{k,i} y_i^{(m)} \right) / h_{k,k}$ 
    }
 $x^{(m)} := x^{(0)} + \sum_{i=1}^m y_i^{(m)} v^{(i)}$ 

```

Din punct de vedere practic, la aplicarea metodei MOC procesul de obținere a soluției SEAL (5) necesită  $O(mn^2 + m^2n)$  operații aritmetice. Volumul de memorie utilizat este de ordinul  $O(n^2 + (m+3)n + m^2/2)$ , iar aceasta pentru numere  $n$  mari limitează cea mai mare valoare a lui  $m$  ce poate fi utilizată. Un remediu constă în a restarta algoritmul periodic (abordare ce se dovedește a fi mai simplă și din punctul de vedere al programării). Se alege din start careva dimensiune  $m$  a subspațiului, relativ mică în raport cu ordinul sistemului. După ce aproximația  $x^{(m)}$  este calculată, se verifică dacă ea este suficient de exactă. Dacă exactitatea cerută încă nu este atinsă, atunci întregul proces se repetă cu vectorul  $x^{(m)}$  în calitate de aproximație inițială.

### Metoda generalizată a residuurilor minimale (MGRM)

La fiecare pas se alege  $K_m = K_m(A, v^{(1)})$ ,  $L_m = AK_m(A, v^{(1)})$ , unde  $v^{(1)} = r^{(0)} / \beta$ ,  $\beta = \|r^{(0)}\|_2$ . Atunci, vectorul  $\tilde{x}$  este rezultatul proiecției soluției SEAL (5) pe subspațiul  $K_m$  ortogonal subspațiului  $L_m$  dacă și numai dacă  $\|b - A\tilde{x}\|_2 = \min_{x \in x^{(0)} + K_m} \|b - Ax\|_2$  (a se vedea [7, p.127]). Ținând cont de aceasta, în metoda MGRM, în locul problemei de proiectare (7) se utilizează problema echivalentă de minimizare a funcționalei  $\|b - Ax\|_2$  pe subspațiul  $x^{(0)} + K_m$ .

Pentru  $x \in x^{(0)} + K_m$  are loc reprezentarea  $x = x^{(0)} + V_m y$ ,  $y \in \square^m$ . Definim funcționala  $J(y) = \|b - Ax\|_2 = \|b - A(x^{(0)} + V_m y)\|_2$ . Atunci, utilizând relațiile (9), (11), obținem  $r = r^{(0)} - AV_m y = \beta v^{(1)} - V_{m+1} \bar{H}_m y = V_{m+1} (\beta e_1 - \bar{H}_m y)$ . Deoarece matricea  $V_{m+1}$  este compusă din vectori (coloane) ortonormați, are loc  $J(y) = \|\beta e_1 - \bar{H}_m y\|_2$ . Astfel, conform metodei MGRM, aproximația soluției este vectorul  $x^{(m)} = x^{(0)} + V_m y^{(m)}$ , unde  $y^{(m)}$  minimizează funcționala  $J(y) = \|\beta e_1 - \bar{H}_m y\|_2$ , adică

$$y^{(m)} = \arg \min_y \|\beta e_1 - \bar{H}_m y\|_2. \quad (13)$$

Pentru a găsi coeficienții  $y_i^{(m)}$  ( $i = \overline{1, m}$ ), este necesar să se rezolve SEAL

$$\bar{H}_m y = \beta e_1. \quad (14)$$

Acest sistem este supradefinit (deoarece matricea  $\bar{H}_m$  are dimensiunea  $(m+1) \times m$ ), de aceea, după cum se vede din relația (13), se va soluționa în sensul celor mai mici pătrate. Deoarece matricea sistemului (14) este în forma Hessenberg superioară, aceasta se rezolvă cu ajutorul reducerii la forma superior triunghiulară. Atunci ultima linie a matricei superior triunghiulare va conține doar elemente nule. Dar, spre deosebire de metoda MOC, în care sistemul asemănător sistemului (14) are matrice pătrată, în cazul metodei MGRM pentru

a reduce la forma triunghiulară se utilizează transformări ortogonale (transformările ortogonale garantează faptul că norma residuului nu va crește), și anume: ținând cont de structura matricei, se utilizează transformările de rotație Givens. Transformarea de rotație, definită prin matricea  $G_k$ , se construiește astfel, încât la aplicarea ei vectorului  $(h_{k,k}, h_{k+1,k})^T$  să se anuleze componenta a doua a acestuia. În total se aplică consecutiv  $m$  transformări de rotație  $G_1, G_2, \dots, G_m$ . Matricea  $Q_m = G_m G_{m-1} \dots G_1$  este matrice ortogonală ca produs de matrici ortogonale. La aplicarea matricei  $Q_m$  sistemului (14), obținem SEAL  $\bar{R}_m y = \bar{g}_m$ , în care  $\bar{R}_m = Q_m \bar{H}_m$ ,  $\bar{g}_m = Q_m (\beta e_1) = (\gamma_1, \dots, \gamma_{m+1})^T$ . Vom nota prin  $R_m$  matricea ce se obține din  $\bar{R}_m$  prin excluderea ultimei linii, iar prin  $g_m$  – vectorul  $m$ -dimensional ce se obține din  $\bar{g}_m$ , excluzând ultimul coeficient.

Se poate arăta (a se vedea [7, p.162]), că vectorul  $y^{(m)} \in \square^m$  ce minimizează funcționala  $J(y) = \|\beta e_1 - \bar{H}_m y\|_2$  se determină din relația  $y^{(m)} = R_m^{-1} g_m$ . Pentru norma residuului  $r^{(m)}$  la pasul  $m$  avem  $\|b - Ax^{(m)}\|_2 = |\gamma_{m+1}|$ , ceea ce permite să se evalueze descreșterea normei residuului fără a găsi în mod explicit soluția aproximativă.

Algoritmul MGRM ce descrie procesul de implementare a metodei poate fi organizat în modul următor.

#### Algoritmul MGRM

Se alege aproximația inițială  $x^{(0)} \in \square^n$   
 $x := x^{(0)}$ ;  $r := b - Ax$ ;  $\beta := \|r\|_2$ ;  $v^{(1)} := r/\beta$ ;  $m := 0$   
do

$m := m + 1$

Se alocă memorie pentru vectorii  $r^{(m)}, h^{(m)} \in \square^{m+1}$ ; Se pune  $r^{(m)} = \bar{0}$ ;  $h^{(m)} = \bar{0}$ ;

Se alocă memorie pentru vectorul  $g^{(m)} \in \square^{m+1}$ ; Se pune  $g^{(m)} = \bar{0}$ ;

if  $m = 1$  then  $g_1^{(m)} := \beta$

else

{  
for  $k = \overline{1, m}$  do

{  
 $g_i^{(m)} := g_i^{(m-1)}$   
}

Se eliberează memoria alocată pentru  $g^{(m-1)}$

}  
 $w^{(m)} := Av^{(m)}$

for  $k = \overline{1, m}$

{  
 $h_k^{(m)} := (v^{(k)})^T w^{(m)}$   
 $w^{(m)} := w^{(m)} - h_k^{(m)} v^{(k)}$   
}  
 $h_{m+1}^{(m)} := \|w^{(m)}\|_2$ ,  $v^{(m+1)} := w^{(m)} / h_{m+1}^{(m)}$

$r_1^{(m)} := h_1^{(m)}$

for  $k = \overline{2, m}$

{  
 $\gamma := c_{k-1} r_{k-1}^{(m)} + s_{k-1} h_k^{(m)}$   
 $r_k^{(m)} := -s_{k-1} r_{k-1}^{(m)} + c_{k-1} h_k^{(m)}$   
 $r_{k-1}^{(m)} := \gamma$   
}

$$\begin{aligned} \delta &:= \sqrt{r_m^{(m)^2} + (h_{m+1}^{(m)})^2}, c_m := r_m^{(m)} / \delta, s_m := h_{m+1}^{(m)} / \delta \\ r_m^{(m)} &:= c_m r_m^{(m)} + s_m h_{m+1}^{(m)} \\ g_m^{(m)} &:= c_m g_m^{(m)}, g_{m+1}^{(m)} := -s_m g_m^{(m)} \\ \rho &:= |g_{m+1}^{(m)}| \end{aligned}$$

while  $\rho \geq \varepsilon$

$$y_m := g_m^{(m)} / r_m^{(m)}$$

for  $k = \overline{m-1, 1}$

$$\left\{ \begin{aligned} y_k &:= \left( g_k^{(m)} - \sum_{i=k+1}^m r_k^{(i)} y_i \right) / r_k^{(k)} \end{aligned} \right\}$$

$$x := x + \sum_{i=1}^m y_i v^{(i)}$$

La aplicarea algoritmului procesul de obținere a soluției SEAL (5) necesită  $O(mn^2 + m^2n)$  operații aritmetice și un volum de memorie de ordinul  $O(n^2 + (m+4)n + m^2/2)$ . Pentru valori mari ale lui  $m$  matricea Arnoldi  $V_m$  poate deveni inacceptabil de mare pentru a fi memorată și atunci, pentru a depăși această problemă, ciclul DO-WHILE se va stopa după  $m$  iterații (numărul  $m$  se alege înainte de a începe calculele) și se va restarta cu soluția aproximativă obținută, în calitate de aproximație inițială.

În unele cazuri este mai simplu de aplicat următoarea metodă.

#### Metoda gradientilor conjugați pentru ecuația normală (MGCEN)

Deoarece sistemul (5) este echivalent SEAL  $A^T A x = A^T b$  cu matrice simetrică și pozitiv definită, la rezolvarea acestuia se poate aplica metoda gradientilor conjugați (*a se vedea* [7, p.236]).

Conform metodei MGCEN, la pasul  $m$  în calitate de aproximație  $x^{(m)}$  se alege elementul  $x$  al subspațiului Krylov  $x^{(0)} + K_m(A^T A, A^T r^{(0)}) = x^{(0)} + \text{span}\{A^T r^{(0)}, A^T A A^T r^{(0)}, \dots, (A^T A)^{m-1} A^T r^{(0)}\}$  ce minimizează norma residuului  $\|b - Ax\|_2$ . Diferența în comparație cu metoda MGRM constă în subspațiul în raport cu care se minimizează norma residuului.

Metoda nu se aplică în cazul când matricea  $A$  a sistemului este slab condiționată, deoarece pentru numărul de condiționare al matricei  $A^T A$  are loc  $k(A^T A) = k^2(A)$  ( $k(A) = \|A\|_2 \|A^{-1}\|_2$ ,  $\|\cdot\|_2$  – norma spectrală a matricei).

Algoritmul de calcul (algoritmul MGCEN) poate fi descris în modul următor:

#### Algoritmul MGCEN

Se alege aproximația inițială  $x^{(0)} \in \mathbb{R}^n$

$$r^{(0)} := b - Ax^{(0)}; \beta := \|r^{(0)}\|_2; z^{(0)} := A^T r^{(0)}; p^{(0)} := z^{(0)}$$

$$m := 0$$

do

$$\omega^{(m)} := A p^{(m)}$$

$$\alpha_m := \|z^{(m)}\|_2^2 / \|\omega^{(m)}\|_2^2$$

$$x^{(m+1)} := x^{(m)} + \alpha_m p^{(m)}$$

$$r^{(m+1)} := r^{(m)} - \alpha_m \omega^{(m)}$$

$$\rho := \|r^{(m+1)}\|_2$$

$$\begin{aligned} z^{(m+1)} &:= A^T r^{(m+1)} \\ \beta_m &:= \|z^{(m+1)}\|_2^2 / \|z^{(m)}\|_2^2 \\ p^{(m+1)} &:= z^{(m+1)} + \beta_m p^{(m)} \\ m &:= m+1 \end{aligned}$$

while  $\rho \geq \varepsilon\beta$

Pentru  $m$  pași ai metodei MGCEN sunt necesare  $O(mn^2)$  operații aritmetice și un volum de memorie de ordinul  $O(n^2)$ .

### Exemplul 1

În EIS (1) se consideră  $a_0(t) = \frac{1}{2}(t^{1/4} + 1)$ ,  $b_0(t) = \frac{1}{2}(t^{1/4} - 1)$ ,  $f(t) = t^{-50} + t^{-2} - 2t^{53/4} + 7t^{205/4}$ ,  $h(t, \tau) = 2t^2\tau^2$ .

Soluția exactă este  $\varphi_*(t) = t^{-50} + t^{-2} - 2t^{13} + 7t^{51}$ . Ușor se verifică că au loc condițiile ce asigură convergența metodei quadraturilor mecanice (*a se vedea* [8]). În Tabelul de mai jos se ilustrează eficacitatea algoritmilor iterativi pentru rezolvarea SEAL (cu componente reale) ce apar la discretizarea acestei EIS cu coeficienți continui pe porțiuni conform metodei quadraturilor mecanice.

Tabelul 1

N	Gauss	MGCEN		MGRM		MGRM restartat		MOC	
	TCS	TCS	Nr_iter	TCS	Nr_iter	TCS	Nr_iter	TCS	Nr_iter
420	0,02	0,0005	17	0,02	420	0,006	375	0,01	417
840	0,15	0,0015	18	0,10	837	0,04	573	0,08	833
1680	1,55	0,006	18	1,39	1670	0,27	1029	0,63	1656
3360	6,57	0,025	19	5,50	3338	2,00	1892	5,07	3328
6720	59,07	0,106	20	53,41	6673	15,96	3589	40,78	6671
13440	342,49	2,1	21	-	-	-	-	-	-

Aici avem:  $N$ -ordinul SEAL cu componente reale; TCS – timpul efectiv de calcul al soluției SEAL (în minute); Nr\_iter – numărul de iterații efectuate pentru a atinge exactitatea  $\varepsilon$ . Exactitatea cu care se precizează soluția SEAL este  $\varepsilon = 10^{-16}$ . În calitate de aproximație inițială în metodele iterative se ia vectorul cu toate componentele nule. Calculele s-au realizat la un calculator AMD Athlon(tm) 64 Processor 3200+ 1.99 GHz și 1GB RAM.

S-a observat că, în cazul SEAL ce apar la discretizarea EIS (1), cea mai efectivă dintre metodele ce se compară este MGCEN.

Vom menționa că algoritmi studiați pot fi modificați neesențial, astfel încât ei să poată fi aplicați și la rezolvarea SEAL  $Ax = b$  cu componente complexe  $A \in \mathbb{C}^{n \times n}$ ,  $b \in \mathbb{C}^n$ . Unicele modificări ce apar țin de realizarea produselor scalare complexe.

În fiecare din acești algoritmi se utilizează procedura de înmulțire a matricei sistemului la vector. Pentru a genera elementele matricei  $A$  în mod direct, sunt necesare cel puțin  $O(n^3)$  operații aritmetice, fapt care deranjează, deoarece se dorește implementarea unor algoritmi de complexitate  $O(mn^2)$  la rezolvarea SEAL (3).

Pentru matricea  $F$  are loc reprezentarea  $F = D_c F_n + D_d F'_n$ , unde  $D_c = \text{diag}(c(t_{-n}), \dots, c(t_{-1}), c(t_0), c(t_1), \dots, c(t_n))$ ,  $D_d = \text{diag}(d(t_{-n}), \dots, d(t_{-1}), d(t_0), d(t_1), \dots, d(t_n))$ , iar  $F_n = \{F_{jk}\}_{j,k=-n}^n$ ,  $F_{jk} = t_j^k$ ,  $F'_n = \{F'_{jk}\}_{j,k=-n}^n$ ,  $F'_{jk} = -t_j^k$  pentru  $j = -n, n$ ,  $k = -n, -1$ , și  $F'_{jk} = t_j^k$  pentru  $j = -n, n$ ,  $k = 0, n$ . La fel, pentru matricea  $H$  are loc relația  $H = G F_n^{-1}$ , unde  $G = \{G_{jk}\}_{j,k=-n}^n$ ,  $G_{jk} = k_\rho(t_j, t_{-k}) t_{-k}$ , iar  $F_n^{-1} = (2n+1)^{-1} \{(t_j)^{-k}\}_{j,k=-n}^n$ .

Ținând cont de acestea, ușor observăm că procedura de înmulțire a matricei  $A = F + H$  la vectorul  $v \in \mathbb{C}^{2n+1}$



se poate realiza în  $O(n^2)$  operații, fără a avea necesitatea de a genera elementele matricelor  $F$  și  $H$ . În cazul când nucleul  $h(t, \tau)$  este nul, procedura menționată necesită  $O(n \log_2 n)$  operații (se utilizează algoritmi transformării rapide Fourier). Atunci procesul de obținere a soluției SEAL (3) conform algoritmilor MOC, MGRM și MGCEN necesită  $O(mn \log_2 n + m^2 n)$  operații aritmetice și un volum de memorie de ordinul  $O(mn)$ .

**Exemplul 2**

Să considerăm EIS (1) în care  $a_0(t) = 0.5(t^{1/4} + 1)$ ,  $b_0(t) = 0.5(t^{1/4} - 1)$ ,  $f(t) = t^{5/4} - t^{-1}$ ,  $h(t, \tau) = 0$ . Soluția exactă este  $\varphi_*(t) = t - t^{-1}$ . Ușor se verifică că au loc condițiile ce asigură convergența metodei quadraturilor mecanice. În Tabelul 2 sunt aduse date ce permit să se compare algoritmi MGRM optimizați (la înmulțirea matricii SEAL la vectori se utilizează algoritmi transformării rapide Fourier) cu cei standard pentru rezolvarea SEAL (cu componente complexe) ce apar la discretizarea EIS conform metodei quadraturilor mecanice.

**Tabelul 2**

N	Algoritmul optimizat				Algoritmul standard			
	MGRM		MGRM restartat		MGRM		MGRM restartat	
	TCS	Nr. Iter.	TCS	Nr. Iter.	TCS	Nr. Iter.	TCS	Nr. Iter.
105	0,02	211	0,007	21	0,03	211	0,009	21
210	0,17	420	0,05	25	0,21	420	0,07	25
420	1,50	839	0,48	29	1,61	839	0,54	29
840	9,35	1676	2,50	34	12,92	1676	4,21	33
1680	100,64	3351	28,50	37	180,95	3351	44,57	38

**Referințe:**

1. Heinig G., Rost K. Algebraic methods for Toeplitz-like matrices and operators. - Berlin: Akademie-Verlag, 1984.
2. Воеводин В.В., Тыртышников Е.Е. Вычислительные процессы с трёхдиагональными матрицами. - Москва: Наука, 1987. - 320 с.
3. Ammar G.S., Gragg W.B. The generalized Schur algorithm for the superfast solution of Toeplitz systems, Rational approximation and its applications in mathematics and physics, Lecture notes in mathematics 1237. - Berlin: Springer-Verlag, 1987, p.315-330.
4. VanBarel M., Heinig G., Kravanja P., A stabilized superfast solver for nonsymmetric Toeplitz systems // SIAM Journal on Matrix Analysis and Applications, 23 (2), 2001, p.494-510.
5. Гохберг И., Семенцул А. Об обращении конечных трёхдиагональных матриц и их непрерывных аналогов. Математические исследования. - Кишинев: Штиинца, 1972, т.7, вып.2, с.201-224.
6. Нуссбаумер Г. Быстрое преобразование Фурье и алгоритмы вычисления свертки. - Москва: Радио и Связь, 1985. -248 с.
7. Saad Y. Iterative methods for solving sparse linear systems. - Boston: PWS Publishing Company, 1996. - 463 p.
8. Capcelea T., Collocation and quadrature methods for solving singular integral equations with piecewise continuous coefficients // Buletinul Academiei de Științe a Republicii Moldova. Matematica. - 2006. - Vol.3(52). - P.27-44.

Prezentat la 05.06.2008