Calcul Numeric

Cursul 6

2020

Anca Ignat

Algoritmul lui Givens

Fie $\boldsymbol{A}$ o matrice reală pătratică de dimensiune $\boldsymbol{n}$. Pp. că avem:

$$A = QR$$

unde $\boldsymbol{Q}$ este o matrice ortogonală iar $\boldsymbol{R}$ este o matrice superior triunghiulară.

$$Ax = b \quad \Leftrightarrow \quad QRx = b \Leftrightarrow Q^T QRx = Q^T b \quad \Leftrightarrow Rx = Q^T b$$

În cazul algoritmului Givens, pentru a aduce sistemul $\boldsymbol{Ax=b}$ la forma $\boldsymbol{Rx = Q^T b}$ se folosesc matricele de rotație. O matrice de rotație $R_{pq}(\theta) = (r_{ij})_{i,j=1,n}$ are următoarea formă :

$$
R_{pq}(\theta) = \begin{array}{cc}
\begin{array}{ccccccc} & & & p & & q & \end{array} & \\
\begin{pmatrix}
1 & 0 & \cdots & 0 & \cdots & 0 & \cdots 0 \\
0 & 1 & \cdots & 0 & \cdots & 0 & \cdots 0 \\
\vdots & & & & & & \\
0 & 0 & \cdots & c & \cdots & s & \cdots 0 \\
\vdots & & & & 1 & & \\
0 & 0 & \cdots & (-s) & \cdots & c & \cdots 0 \\
\vdots & & & & & & \\
0 & 0 & \cdots & 0 & \cdots & 0 & \cdots 1
\end{pmatrix} & \begin{array}{c} \\ \\ \\ p \\ \\ q \\ \\ \\ \end{array}
\end{array}
$$

$$r_{ij} = \begin{cases} \mathbf{1} & \textbf{pentru } \boldsymbol{i = j, \quad i \neq p, i \neq q} \\ \boldsymbol{c} & \textbf{pentru } \boldsymbol{i = j, \quad i = p, i = q} \\ \boldsymbol{s} & \textbf{pentru } \boldsymbol{i = p, j = q} \\ \boldsymbol{-s} & \textbf{pentru } \boldsymbol{i = q, j = p} \\ \mathbf{0} & \text{în rest} \end{cases}$$

unde $\boldsymbol{p, q \in \{1,...,n\}}$ iar $\boldsymbol{c}$ şi $\boldsymbol{s}$ sunt două numere reale care satisfac relația $\boldsymbol{c^2+s^2=1}$. Constantele $\boldsymbol{c}$ şi $\boldsymbol{s}$ pot fi alese astfel încât $\boldsymbol{c=\cos\theta, s=\sin\theta}$. Se arată ușor, folosind relația $\boldsymbol{c^2+s^2=1}$, că matricea $\boldsymbol{R_{pq}(\theta)}$ este ortogonală:

$$\boldsymbol{R_{pq}(\theta)\, R_{pq}^T(\theta) = R_{pq}^T(\theta) R_{pq}(\theta) = I_n}$$

Calculul matricei:

$$B = R_{pq}(\theta)A \,,$$

B se obține din A modificând doar liniile p şi q.
Fie

$$A_i = e_i^T A, \text{ - linia } i \text{ a matricei } A$$

$$B_i = e_i^T B \quad \text{ - linia } i \text{ a matricei } B.$$

Liniile matricei B:

$$B_i = A_i \;, \quad i = 1,\ldots,n\,, \quad i \neq p\,, i \neq q$$

$$B_p = cA_p + sA_q$$

$$B_q = -sA_p + cA_q$$

$$b_{pj} = c\, a_{pj} + s\, a_{qj} \ \ , \ \ j = 1,\ldots,n$$

$$b_{qj} = -s\, a_{pj} + c\, a_{qj} \ \ , \ j = 1,\ldots,n$$

$$b_{ij} = a_{ij} \quad \text{în rest}$$

Calculul matricei :

$$D = A\, R_{pq}^{T}(\theta) \ ,$$

D se obține din ***A*** modificând doar coloanele ***p*** și ***q***.

Notăm $\boldsymbol{Ae_j}$, $\boldsymbol{De_j}$ – coloana j a matricei A şi respectiv D.

Coloanele matricei ***D***:

$$D_j = A_j \;\;,\quad j = 1,\ldots,n\,,\quad j \neq p\,,\, j \neq q$$

$$De_p = cAe_p + sAe_q$$

$$De_q = -sAe_p + cAe_q$$

$$d_{ip} = c\, a_{ip} + s\, a_{iq} \;\;,\;\; i = 1,\ldots,n$$

$$d_{iq} = -\, s\, a_{ip} + c\, a_{iq} \;\;,\, i = 1,\ldots,n$$

$$d_{ij} = a_{ij} \quad \text{în rest}$$

Algoritmul lui Givens se desfăşoară în ***(n-1)*** paşi - la pasul ***r*** se transformă coloana ***r*** a matricei ***A*** în formă superior triunghiulară fără a modifica primele ***(r-1)*** coloane.

Pasul 1

Intrare: matricea $\boldsymbol{A}$, vectorul $\boldsymbol{b}$

Ieșire: matricea $\boldsymbol{A}^{(1)}$ (cu prima coloană în formă superior triunghiulară), $\boldsymbol{b}^{(1)}$

Se efectuează următoarele operații de înmulțire cu matrice de rotație:

$$\boldsymbol{R}_{1n}(\theta_{1n})\cdots\boldsymbol{R}_{13}(\theta_{13})\boldsymbol{R}_{12}(\theta_{12})\,\boldsymbol{A}=\boldsymbol{A}^{(1)}$$

$$\boldsymbol{R}_{1n}(\theta_{1n})\cdots\boldsymbol{R}_{13}(\theta_{13})\boldsymbol{R}_{12}(\theta_{12})\,\boldsymbol{b}=\boldsymbol{b}^{(1)}$$

Unghiurile θ_{1i} (constantele c_{1i} și s_{1i}) se aleg astfel ca elementul de pe poziția $(i,1)$ din matricea rezultat să devină 0.

Pasul r

Intrare: matricea $\boldsymbol{A^{(r-1)}}$ (are primele $\boldsymbol{(r-1)}$ coloane în formă superior triunghiulară), $\boldsymbol{b^{(r-1)}}$

Ieșire: matricea $\boldsymbol{A^{(r)}}$ (cu primele $\boldsymbol{r}$ coloane în formă superior triunghiulară), $\boldsymbol{b^{(r)}}$

La acest pas matricea $\boldsymbol{A^{(r-1)}}$ şi vectorul $\boldsymbol{b^{(r-1)}}$ se transformă astfel:

$$\boldsymbol{R_{r\,n}(\theta_{r\,n})\cdots R_{r\,i}(\theta_{r\,i})\cdots R_{r\,r+1}(\theta_{r\,r+1})\,A^{(r-1)} = A^{(r)}}$$

$$\boldsymbol{R_{r\,n}(\theta_{r\,n})\cdots R_{r\,i}(\theta_{r\,i})\cdots R_{r\,r+1}(\theta_{r\,r+1})\,b^{(r-1)} = b^{(r)}}$$

unde elementele $c = c_{ri}$ şi $s = s_{ri}$ din matricele de rotație se aleg astfel ca după înmulțirea cu $R_{ri}(\theta_{ri})\,, i = r+1,\ldots,n$ elementul ***(i,r)*** să devină ***0***.

Considerăm operația $B = R_{ri}(\theta_{ri})A$, unde θ_{ri} se alege astfel ca $b_{ir} = 0$:

$$b_{rj} = c\,a_{rj} + s\,a_{ij}\ ,$$

$$b_{ij} = -s\,a_{rj} + c\,a_{ij}\ ,\ j = 1,\ldots,n$$

$$(b_{kl} = a_{kl} \quad \text{în rest})$$

$$(j = r) \quad b_{ir} = -s\,a_{rr} + c\,a_{ir}$$

Cea mai simplă alegere pentru ***c*** și ***s*** astfel ca să obținem $\boldsymbol{b_{ir}=0}$ este:

$$c = f\, a_{rr} \quad , \quad s = f\, a_{ir} \quad ,$$

$$f \ \ \textbf{ales astfel ca } c^2 + s^2 = 1 \quad \Rightarrow \quad f = \frac{1}{\sqrt{a_{rr}^2 + a_{ir}^2}}$$

$$c = \frac{a_{rr}}{\sqrt{a_{rr}^2 + a_{ir}^2}} \quad , \quad s = \frac{a_{ir}}{\sqrt{a_{rr}^2 + a_{ir}^2}}$$

$$\sqrt{a_{rr}^2 + a_{ir}^2} = 0 \ \Rightarrow a_{rr} = a_{ir} = 0 \ \Rightarrow \ c = 1\,,\ s = 0$$

$$(R_{ir}(\theta) \ = \ I_n)$$

Deci elementul de pe poziția ***(i,r)*** este deja nul. Nu avem ce schimba în matricea ***A*.**

Operația $\boldsymbol{B = R_{r\,i}(\theta_{r\,i})A}$, nu afectează forma superior triunghiulară a primelor ***(r-1)*** coloane. În matricea ***B*** aceste coloane vor continua să fie în formă superior triunghiulară.

$$b_{rj} = c\,a_{rj} + s\,a_{ij} = 0\,,\quad b_{ij} = -\,s\,a_{rj} + c\,a_{ij} = 0\;\;,\; j = 1,\ldots,r-1$$

$$\textbf{deoarece}\quad a_{rj} = a_{ij} = 0$$

Înmulțirea $\boldsymbol{B = R_{r\,i}(\theta_{r\,i})A}$ nu schimbă decât liniile ***r*** și ***i*** ale matricei ***B***. În concluzie, operația $\boldsymbol{B = R_{r\,i}(\theta_{r\,i})A}$ nu schimbă elementele nule deja obținute, ci doar face ca elementul de pe poziția ***(i,r)*** să devină ***0***.

Algoritmul lui Givens poate fi descris astfel:

$$R_{n-1n}(\theta_{n-1n})\cdots R_{rn}(\theta_{rn})\cdots R_{r\,r+1}(\theta_{r\,r+1})\cdots R_{1n}(\theta_{1n})\cdots R_{12}(\theta_{12})\,A = R$$

$$R_{n-1n}(\theta_{n-1n})\cdots R_{rn}(\theta_{rn})\cdots R_{r\,r+1}(\theta_{r\,r+1})\cdots R_{1n}(\theta_{1n})\cdots R_{12}(\theta_{12})b = \bar{b} = Q^T b$$

Notăm cu $\widetilde{Q}$ următoarea matrice:

$$\widetilde{Q} = R_{n-1n}(\theta_{n-1n})\cdots R_{rn}(\theta_{rn})\cdots R_{r\,r+1}(\theta_{r\,r+1})\cdots R_{1n}(\theta_{1n})\cdots R_{12}(\theta_{12})$$

Matricea $\widetilde{\boldsymbol{Q}}$ este matrice ortogonală ca produs de matrice ortogonale. Descompunerea $\boldsymbol{QR}$ a matricei $\boldsymbol{A}$ este următoarea:

$$\widetilde{Q}\,A = R \quad (\widetilde{Q}^{-1} = \widetilde{Q}^{T}) \Rightarrow A = \widetilde{Q}^{T} R = QR$$

$$\begin{aligned} Q = \widetilde{Q}^{T} &= \left(R_{n-1n}(\theta_{n-1n})\cdots R_{rn}(\theta_{rn})\cdots R_{r\,r+1}(\theta_{r\,r+1})\cdots R_{1n}(\theta_{1n})\cdots R_{12}(\theta_{12})\right)^{T} \\ &= R_{12}^{T}(\theta_{12})\cdots R_{1n}^{T}(\theta_{1n})\cdots R_{r\,r+1}^{T}(\theta_{r\,r+1})\cdots R_{rn}^{T}(\theta_{rn})\cdots R_{n-1n}^{T}(\theta_{n-1n}) \end{aligned}$$

Pe scurt, ***algoritmul lui Givens*** este următorul:

$$\widetilde{Q} = I_n;$$

$$\textbf{for } r = 1,\cdots,n-1$$

$$\quad \textbf{for } i = r+1,\cdots,n$$

$$\left\{\begin{aligned} A &= R_{ri}(\theta_{ri}) * A; \\ b &= R_{ri}(\theta_{ri}) * b; \\ \widetilde{Q} &= R_{ri}(\theta_{ri}) * \widetilde{Q}; \end{aligned}\right.$$

$\widetilde{Q} = I_n;$

for $r = 1, \ldots, n-1$

for $i = r+1, \ldots, n$

// **construcţia matricii** $R_{ri}(\theta_{ri})$ – **constantele** c **şi** s

- $f = \sqrt{a_{rr}^2 + a_{ir}^2};$
- **if** $(\ f \le \varepsilon\ )\{\ c = 1;\ s = 0;\}$ // $R_{ri}(\theta_{ri}) = I$

  else $\{\ c = a_{rr}\ /\ f\ ;\ s = a_{ir}\ /\ f\ ;\}$

// $A = R_{ri}(\theta_{ri}) * A$

- **for** $j = r+1, \ldots, n$

$$\begin{cases} a_{rj} = c * a_{rj} + s * a_{ij}; \\ a_{ij} = -s * a_{rj}^{vechi} + c * a_{ij}; \end{cases}$$

$$\bullet\ a_{ir} = 0;\ a_{rr} = f\ ;$$

$$//\ b = R_{ri}(\theta_{ri}) * b$$

$$\bullet\ b_r = c * b_r + s * b_i;$$

$$\bullet\ b_i = -s * b_r^{vechi} + c * b_i;$$

$$//\ \widetilde{Q} = R_{ri}(\theta_{ri}) * \widetilde{Q}$$

$$\bullet\ \textbf{for}\ j = 1, \dots, n$$

$$\begin{cases} \tilde{q}_{rj} = c * \tilde{q}_{rj} + s * \tilde{q}_{ij}; \\ \tilde{q}_{ij} = -s * \tilde{q}_{rj}^{vechi} + c * \tilde{q}_{ij}; \end{cases}$$

La sfârșitul acestui algoritm, în matricea $\boldsymbol{A}$ vom avea matricea superior triunghiulară $\boldsymbol{R}$, în vectorul $\boldsymbol{b}$ vom avea $\widetilde{\boldsymbol{Q}}\boldsymbol{b}^{\mathbf{init}}$($\boldsymbol{b}^{\mathbf{init}}$ - vectorul termenilor liberi inițial), iar matricea $\widetilde{\boldsymbol{Q}}$ va conține matricea $\boldsymbol{Q}^T$ din factorizarea $\boldsymbol{QR}$ a matricei $\boldsymbol{A}$.

Numărând operațiile efectuate (exceptând calculul matricei $\widetilde{\boldsymbol{Q}}$) obținem:

$$\frac{n(n-1)}{2} \ \textbf{radicali}$$

$$\frac{n(n-1)(4n+7)}{6}=\frac{2}{3}n^3+O(n^2) - \textbf{adunări/scăderi}$$

$$\frac{2n(n-1)(2n+5)}{3}=\frac{4}{3}n^3+O(n^2) \ \textbf{înmulţiri/împărţiri}$$

QR – algoritmul Gram Schmidt

$$A \in \mathbb{R}^{n\times n} \quad \textbf{cu} \quad \textbf{det A} \neq \mathbf{0}$$

$$A=QR$$

$$\begin{bmatrix} a^1 & a^2 & \cdots & a^n \end{bmatrix} = \begin{bmatrix} q^1 & q^2 & \cdots & q^n \end{bmatrix} \begin{pmatrix} r_{11} & r_{12} & \cdots & r_{1n} \\ 0 & r_{22} & \cdots & r_{2n} \\ \vdots & & & \\ 0 & 0 & \cdots & r_{nn} \end{pmatrix} \quad \textbf{(1)}$$

$a^j = Ae_j$ – coloana j a matricei A

$q^j = Qe_j$ – coloana j a matricei Q

Relația (1) poate fi rescrisă astfel:

$$\begin{cases} r_{11}q^1 = a^1 \\ r_{12}q^1 + r_{22}q^2 = a^2 \\ \vdots \\ r_{1p}q^1 + \cdots + r_{jp}q^j + \cdots + r_{pp}q^p = a^p \\ \vdots \\ r_{1n}q^1 + \cdots + r_{jn}q^j + \cdots + r_{nn}q^n = a^n \end{cases} \tag{2}$$

Algoritmul de calcul al descompunerii ***QR*** cu metoda Gram-Schmidt se desfăşoară în ***n*** paşi, la fiecare pas calculându-se:

- coloana ***p*** din matricea ***R***
- coloana ***p*** din matricea ***Q***

Avem:

$$\mathbf{det}\, A \neq \mathbf{0},\ A = QR,\ Q - \ \textbf{ortogonală} \Rightarrow\ \mathbf{det}\, R \neq \mathbf{0}\ (r_{ii} \neq \mathbf{0}\ \forall i)$$

Pasul 1

Se folosește prima ecuație a sistemului (2)

$$r_{11}q^1 = a^1$$

Se face produsul scalar al acestei ecuații cu vectorii q^1 şi a^1. Se folosește proprietatea coloanelor matricelor ortogonale:

$$\left(q^i, q^j\right)_{\mathbb{R}^n} = \begin{cases} \| q^i \|_2^2 = 1 & \text{pentru} \quad i = j \\ 0 & \text{pentru} \quad i \neq j \end{cases}$$

Din:

$$q^1 = \frac{1}{r_{11}} a^1 \quad \text{şi} \ \left\| q^1 \right\|_2 = 1$$

obținem:

$$r_{11} = \pm \left\| a^1 \right\|_2 \quad , \quad q^1 = \frac{1}{r_{11}} a^1 \ (r_{11} \neq 0 \text{ deoarece } \det A \neq 0)$$

Pasul p

Se folosește ecuația ***p*** sistemului (2):

$$r_{1p} q^1 + \cdots + r_{jp} q^j + \cdots + r_{pp} q^p = a^p$$

La acest pas se cunosc deja coloanele $q^1, q^2, \ldots, q^{p-1}$. Se face, pe rând, produsul scalar al acestei ecuații cu vectorii:

$$q^1, q^2, \ldots, q^{p-1}$$

$$q^p \text{ şi } a^p.$$

$$\left(\sum_{k=1}^{p} r_{kp} q^k, q^j\right)_{\mathbb{R}^n} = \left(a^p, q^j\right)_{\mathbb{R}^n} \quad j = 1, \dots, p-1$$

$$\left(\sum_{k=1}^{p} r_{kp} q^k, q^j\right)_{\mathbb{R}^n} = \sum_{\substack{k=1 \\ k \neq j}}^{p} r_{kp} \left(q^k, q^j\right)_{\mathbb{R}^n} + r_{jp} \left(q^j, q^j\right)_{\mathbb{R}^n} = r_{jp}$$

$$r_{jp} = \left(a^p, q^j\right)_{\mathbb{R}^n} \quad j = 1, \dots, p-1$$

Avem:

$$q^p = \frac{1}{r_{pp}} \left(a^p - r_{1p} q^1 - \cdots - r_{p-1\,p} q^{p-1}\right)$$

$$\| q^p \|_2^2 = 1 = \frac{1}{r_{pp}^2} \| \left(a^p - r_{1p} q^1 - \cdots - r_{p-1\,p} q^{p-1} \right) \|_2^2$$

Obținem:

$$r_{pp} = \pm \left\| a^p - r_{1p} q^1 - \cdots - r_{p-1\,p} q^{p-1} \right\|_2$$

$$q^p = \frac{1}{r_{pp}} \left(a^p - r_{1p} q^1 - \cdots - r_{p-1\,p} q^{p-1} \right) = \frac{1}{r_{pp}} \left(a^p - \sum_{j=1}^{p-1} r_{jp} q^j \right)$$

Algoritmul Gram Schmidt modificat

for $\boldsymbol{i = 1,n}$

$\qquad \boldsymbol{v^i = a^i;}$

for $\boldsymbol{i = 1, n}$

$\qquad \boldsymbol{r_{ii} = \|v^i\|_2;}$

$\qquad \boldsymbol{q^i = (1/ r_{ii}) v^i;}$

$\qquad$ *for* $\boldsymbol{j = (i + 1) , n}$

$\qquad\qquad \boldsymbol{r_{ij} = (q^i , v^j);}$

$\qquad\qquad \boldsymbol{v^j = v^j - r_{ij} q^i;}$

$$\mathbf{M:}\ \frac{(n^3 + 3n^2)}{2} \qquad\qquad \mathbf{A:}\ \frac{(n^3 + n^2 - 2)}{2}$$

Exemple descompuneri ***QR***

http://profs.info.uaic.ro/~ancai/CN/curs/exempleQR/

Metode iterative pentru rezolvarea sistemelor de ecuaţii liniare

$$\boldsymbol{Ax = b}\,, \quad \boldsymbol{A} \in \mathbb{R}^{n \times n}\,, \quad \boldsymbol{b} \in \mathbb{R}^{n} \qquad (1)$$

- se presupune cunoscut că $\boldsymbol{A}$ este nesingulară, **det** $\boldsymbol{A \neq 0}$;
- soluţia exactă a sistemului (1) se notează cu $\boldsymbol{x}^*$:

$$\boldsymbol{x^* = A^{-1}b} \qquad (2)$$

- $\boldsymbol{n}$ - dimensiunea sistemului este "*mare*";
- $\boldsymbol{A}$ este matrice rară - cu "*puţine*" elemente $\boldsymbol{a_{ij} \neq 0}$;

- pentru a aproxima soluţia $\boldsymbol{x}^*$ matricea $\boldsymbol{A}$ nu se schimbă (transformă) ci doar se folosesc elementele nenule ale matricei ;
- se construieşte un şir de vectori $\{\boldsymbol{x}^{(k)}\} \subseteq \mathbb{R}^n$, şir care în anumite cazuri, converge la $\boldsymbol{x}^*$:

$$\boldsymbol{x}^{(k)} \to \boldsymbol{x}^* \quad \textbf{pentru } \boldsymbol{k} \to \infty$$

O schemă generală de deducere a unei metode iterative

Fie descompunerea:

$$\boldsymbol{A} = \boldsymbol{B} - \boldsymbol{C}\ , \quad \boldsymbol{B}, \boldsymbol{C} \in \boldsymbol{R}^{n \times n}\ , \quad \boldsymbol{B} \text{ “}\textit{uşor}\text{” inversabilă} \qquad (3)$$

Ce înseamnă ***B*** "*uşor*" inversabilă ? Sistemul liniar, având ca matrice a sistemului matricea ***B***:

$$\boldsymbol{Bx} = \boldsymbol{f}$$

se rezolvă ‘uşor’ (adică repede) – ca în cazul sistemelor cu matrici diagonale sau triunghiulare, de exemplu.

$$Ax^* = b \leftrightarrow Bx^* - Cx^* = b \leftrightarrow$$
$$Bx^* = Cx^* + b \leftrightarrow x^* = B^{-1}Cx^* + B^{-1}b = Mx^* + d$$

unde

$$M := B^{-1}C \in \mathbb{R}^{n\times n}\,, \quad d := B^{-1}b \in \mathbb{R}^n \qquad (4)$$

Şirul $\{x^{(k)}\}$ se construieşte astfel:

$$x^{(k+1)} := Mx^{(k)} + d,\, k = 0,1,2,\dots\ x^{(0)} \in \mathbb{R}^n \text{ ales arbitrar} \quad (5)$$

Vectorul $x^{(k+1)}$ poate fi privit şi ca soluţia sistemului liniar:

$$Bx = f \;\text{ cu }\; f := Cx^{(k)} + b \qquad (6)$$

Cunoscând vectorul $\boldsymbol{x}^{(k)}$, următorul element din şir, $\boldsymbol{x}^{(k+1)}$, se poate construi fie utilizând relaţia (5) (dacă putem construi matricea ***M*** explicit), fie rezolvând sistemul liniar (6).

Matricea ***M*** poartă numele de ***matricea iteraţiei*** iar vectorul $\boldsymbol{x}^{(0)} \in \mathbb{R}^{n}$ se numeşte ***iteraţia iniţială***.

Ne punem problema convergenţei şirului $\boldsymbol{x}^{(k)}$:

$$\boldsymbol{x}^{(k)} \to \boldsymbol{x}^{*} \quad , \quad \boldsymbol{k} \to \infty$$

Se ştie că această convergenţă nu are loc pentru orice matrice ***B***. Avem următorul rezultat general de convergenţă.

Teorema de convergență

Fie $\boldsymbol{A} \in \mathbb{R}^{n\times n}$ o matrice nesingulară şi $\boldsymbol{B},\boldsymbol{C} \in \mathbb{R}^{n\times n}$, $\det \boldsymbol{B} \neq 0$, astfel ca $\boldsymbol{A}=\boldsymbol{B}-\boldsymbol{C}$. Fie $\boldsymbol{x}^{(0)} \in \mathbb{R}^{n}$ un vector oarecare şi $\{\boldsymbol{x}^{(k)}\}$ şirul de vectori dat de relaţia (5) cu $\boldsymbol{M}$ şi $\boldsymbol{d}$ daţi de (4). Atunci:

$$\boldsymbol{x}^{(k)} \to \boldsymbol{x}^{*}, \boldsymbol{k} \to \infty, \ \forall \boldsymbol{x}^{(0)} \Leftrightarrow \rho(\boldsymbol{M}) < 1 \qquad (7)$$

unde $\rho(\boldsymbol{M}) = \max\{|\lambda|; \lambda - \textbf{valoare proprie a matricii } \boldsymbol{M}\}$ este raza spectrală a matricei $\boldsymbol{M}$. Dacă există o normă matricială naturală astfel ca $\|\boldsymbol{M}\| < \boldsymbol{1}$ atunci șirul $\{\boldsymbol{x}^{(k)}\}$ converge la soluția $\boldsymbol{x}^{*}$ a sistemului (1).

$$\|\boldsymbol{M}\| < 1 \Rightarrow \quad \boldsymbol{x}^{(k)} \to \boldsymbol{x}^{*}, \boldsymbol{k} \to \infty, \forall \boldsymbol{x}^{(0)}. \qquad (8)$$

Demonstrație: Scăzând relațiile (5) şi $x^* = Mx^* + d$ obţinem:

$$x^{(k+1)} - x^* = M(x^{(k)} - x^*),\, k = 0,1,2,\ldots$$

Avem:

$$x^{(p)} - x^* = M(x^{(p-1)} - x^*) = M^2(x^{(p-2)} - x^*) = \cdots = M^p(x^{(0)} - x^*)$$

$$x^{(p)} - x^* = M^p(x^{(0)} - x^*),\, \forall p$$

Prin urmare:

$$x^{(p)} \to x^*,\, p \to \infty \;\Leftrightarrow\; M^p \to 0,\, p \to \infty$$

$$M^p \to 0,\, p \to \infty \;\Leftrightarrow\; \rho(M) < 1$$

Dacă:

$$\| M \| < 1 \;\Rightarrow\; M^p \to 0,\, p \to \infty \;\Rightarrow\; x^{(p)} \to x^*,\, p \to \infty \quad \forall x^{(0)}$$

Evaluarea erorii absolute $\| x^{(k)} - x^{*} \|$

Presupunem $\|M\|<1$ (şirul $\{x^{(k)}\}$ converge la x^{*}).
Avem din (5):

$$x^{(l+1)} = Mx^{(l)} + d$$

$$x^{(l)} = Mx^{(l-1)} + d$$

$$x^{(l+1)} - x^{(l)} = M(x^{(l)} - x^{(l-1)}) \quad \forall l$$

Pentru orice k, j, folosind relaţiile de mai sus, avem:

$$x^{(k+j+1)} - x^{(k+j)} = M(x^{(k+j)} - x^{(k+j-1)}) = \cdots = M^{j}(x^{(k+1)} - x^{(k)}) \, \forall k, j$$

Aplicând succesiv relaţia precedentă obţinem:

$$x^{(k+p)} - x^{(k)} = x^{(k+p)} - x^{(k+p-1)} + x^{(k+p-1)} - x^{(k+p-2)} + \cdots +$$

$$+ x^{(k+2)} - x^{(k+1)} + x^{(k+1)} - x^{(k)} =$$

$$= \sum_{j=0}^{p-1} (x^{(k+j+1)} - x^{(k+j)})$$

$$x^{(k+p)} - x^{(k)} = \sum_{j=0}^{p-1} (x^{(k+j+1)} - x^{(k+j)}) = (\sum_{j=0}^{p-1} M^{j})(x^{(k+1)} - x^{(k)})$$

Făcând $p \to \infty$ obţinem:

$$x^* - x^{(k)} = \left(\sum_{j=0}^{\infty} M^j\right) M \left(x^{(k)} - x^{(k-1)}\right)$$

$$\| M \| < 1 \Rightarrow \sum_{j=0}^{\infty} M^j = (I_n - M)^{-1}$$

Mai avem şi evaluarea:

$$\| M \| < 1 \quad \Rightarrow \quad \frac{1}{1 + \| M \|} \leq \| (I_n - M)^{-1} \| \leq \frac{1}{1 - \| M \|}$$

Prin urmare:

$$\| x^* - x^{(k)} \| \leq \frac{\| M \|}{1 - \| M \|} \| x^{(k)} - x^{(k-1)} \|$$

Această relaţie ne spune că din punct de vedere practic putem opri algoritmul atunci când diferenţa dintre două iteraţii succesive devine suficient de mică, acest lucru asigurând apropierea de soluţie.

În continuare vedea cum se memorează matricele rare și vom particulariza matricea $\boldsymbol{B}$.

Memorarea matricelor rare

- se memorează doar valorile nenule şi suficiente informaţii despre indici astfel ca să se poată reconstitui complet matricea

Pp. că matricea ***A*** are ***NN*** elemente nenule.

Memorare comprimată pe linii

Se folosesc 3 vectori:

valori – vector de numere reale de dimensiune ***NN***
ind_col – vector de indici de dimensiune ***NN***
inceput_linii – vector de întregi de dimensiune ***n+1***

În vectorul ***valori*** se memorează elementele nenule ale matricii ***A*** în ordinea liniilor iar în vectorul ***ind_col*** se memorează indicii de coloană ai elementelor din ***valori***. În vectorul ***inceput_linii*** se stochează indicele/poziţia în vectorul ***valori / ind_col*** al/a primului element de pe linia ***i*** memorat în vectorii ***valori / ind_col***.

- ***inceput_linii(n+1) = NN+1***
- ***inceput_linii(i+1) – inceput_linii(i) =***
  numărul de elemente nenule de pe linia ***i, i=1,n***

$$A = \begin{pmatrix} 102.5 & 0.0 & 2.5 & 0.0 & 0.0 \\ 3.5 & 104.88 & 1.05 & 0.0 & 0.33 \\ 0.0 & 0.0 & 100.0 & 0.0 & 0.0 \\ 0.0 & 1.3 & 0.0 & 101.3 & 0.0 \\ 0.73 & 0.0 & 0.0 & 1.5 & 102.23 \end{pmatrix}$$

n=5 , NN=12

valori = (102.5, 2.5, 0.33, 1.05, 104.88, 3.5, 100.0, 101.3, 1.3, 1.5, 0.73, 102.23)

ind_col = (1, 3, 5, 3, 2, 1, 3, 4, 2, 4, 1, 5)

inceput_linii = (1, 3, 7, 8, 10, 13)

Dacă se ştie că matricea are maxim ***n_max*** elemente nenule pe fiecare linie se pot folosi 2 matrici pentru memorarea rară:

valori – matrice de numere reale de dimensiune ***n x n_max***
ind_col – matrice de indici de dimensiune ***n x n_max***

În matricea ***valori*** se memorează pe linia ***i*** elementele nenule de pe linia ***i*** a matricei ***A*** iar în matricea ***ind_col*** se memorează indicii de coloană ai elementelor corespunzătoare din matricea ***valori***.

$$A = \begin{pmatrix} 102.5 & 0.0 & 2.5 & 0.0 & 0.0 \\ 0.0 & 104.88 & 1.05 & 0.0 & 0.33 \\ 0.0 & 0.0 & 100.0 & 0.0 & 0.0 \\ 0.0 & 1.3 & 0.0 & 101.3 & 0.0 \\ 0.73 & 0.0 & 0.0 & 1.5 & 102.23 \end{pmatrix}$$

$$valori = \begin{pmatrix} 102.5 & 2.5 & 0 \\ 104.88 & 1.05 & 0.33 \\ 100.0 & 0 & 0 \\ 101.3 & 1.3 & 0 \\ 102.23 & 1.5 & 0.73 \end{pmatrix} \quad ind\_col = \begin{pmatrix} 1 & 3 & 0 \\ 2 & 3 & 5 \\ 3 & 0 & 0 \\ 4 & 2 & 0 \\ 5 & 4 & 1 \end{pmatrix}$$

Diagonalele matricei ***A***:

$$
\begin{aligned}
&\boldsymbol{d_0} : \left(\boldsymbol{a_{11}}, \boldsymbol{a_{22}}, \ldots, \boldsymbol{a_{nn}}\right)\\
&\boldsymbol{d_1} : \left(\boldsymbol{a_{12}}, \boldsymbol{a_{23}}, \ldots, \boldsymbol{a_{n-1n}}\right)\\
&\boldsymbol{d_{-1}} : \left(\boldsymbol{a_{21}}, \boldsymbol{a_{32}}, \ldots, \boldsymbol{a_{nn-1}}\right)\\
&\boldsymbol{d_2} : \left(\boldsymbol{a_{13}}, \boldsymbol{a_{24}}, \ldots, \boldsymbol{a_{n-2n}}\right)\\
&\boldsymbol{d_{-2}} : \left(\boldsymbol{a_{31}}, \boldsymbol{a_{42}}, \ldots, \boldsymbol{a_{nn-2}}\right)\\
&\vdots
\end{aligned}
$$

Pentru matricele care au elementele nenule plasate pe câteva din diagonalele matricei ***A*** (***n_d*** diagonale cu elemente nenule) se pot folosi pentru memorare o matrice şi un vector:

diag – matrice cu numere reale de dimensiune ***n x n_d***
diag_no – vector de întregi de dimensiune ***n_d***

În matricea ***diag*** se memorează pe coloane diagonalele cu elemente nenule iar în ***diag_no*** este specificat numărul diagonalei care e memorat în coloana ***j*** a matricei ***diag***.

$$\boldsymbol{diag(i,j) = a_{i\ i+diag\_no(j)}}$$

$$A = \begin{pmatrix} 20.5 & 2.0 & 0.0 & 0.0 & 0.0 \\ 0.0 & 40.5 & 3.0 & 0.0 & 0.0 \\ 1.0 & 0.0 & 100.0 & 0.0 & 0.0 \\ 0.0 & 2.3 & 0.0 & 101.5 & 4.0 \\ 0.0 & 0.0 & 3.0 & 0.0 & 102.5 \end{pmatrix}$$

$$diag = \begin{pmatrix} * & 20.5 & 2.0 \\ * & 40.5 & 3.0 \\ 1.0 & 100.0 & 0.0 \\ 2.3 & 101.5 & 4.0 \\ 3.0 & 102.5 & * \end{pmatrix} \quad diag\_no = (-2, 0, 1)$$

Alte tipuri de memorări rare:
http://netlib.org/linalg/html_templates/node90.html