Laboratorul de Metode Numerice

Universitatea "POLITEHNICA" Bucureşti, Facultatea de Electrotehnică, http://lmn.pub.ro, lmn@lmn.pub.ro

Laborator - metode numerice în ingineria electrică

L8 - Metode iterative de rezolvare a sistemelor algebrice liniare

Autori:

Conf. Gabriela Ciuprina
Prof. Daniel Ioan
Prep. Marius Piper
As. Marius Rădulescu
As. Mihai Popescu

5 aprilie 2004

Cuprins

Capitolul 8

Metode iterative de rezolvare a sistemelor algebrice liniare

8.1 Caracterizarea metodelor

Metodele iterative sunt metode care permit obţinerea soluţiei numerice a unui sistem de ecuaţii, prin generarea unui şir care tinde către soluţia exactă. Elementele acestui şir de iteraţii se calculează recursiv, iar procesul se opreşte după un număr finit de paşi, la îndeplinirea criteriului de eroare.

Chiar dacă soluţia obţinută prin metode iterative este afectată de erori de trunchiere, erori care nu apar în cazul metodelor directe, este totuşi posibil ca soluţia iterativă să fie mai precisă decât cea obţinută prin metode directe. Pentru o anumită clasă de sisteme, metodele iterative sunt superioare atât din punctul de vedere al erorii cât şi din cel al efortului de cacul.

8.2 Principiul metodei

În lucrare se prezintă cele mai simple metode iterative pentru rezolvarea sistemelor algebrice liniare şi se descrie clasa sistemelor pentru care acestea pot fi aplicate.

Metodele iterative de rezolvare a sistemelor de ecuaţii liniare sunt metodele în care termenul $\mathbf{x}^{(k)}$ al şirului soluţiilor se obţine din termenul anterior, $\mathbf{x}^{(k-1)}$.

Soluţia exactă se obţine teoretic după un număr infinit de iteraţii. În practică, prin efectuarea unui număr finit de iteraţii, se poate ajunge la o aproximare suficient de bună a soluţiei exacte. Dacă şirul iteraţiilor este convergent, cu cât se efectează mai multe iteraţii, cu atât soluţia numerică este mai precis determinată, erorile, atât cele de trunchiere cât şi cele de rotunjire, devenind tot mai mici.

În metodele iterative, se porneşte de la o iniţializare arbitrară pentru vectorul soluţie numerică $\mathbf{x}^0$. Pentru a determina noua valoare a soluţiei numerice, se rescrie ecuaţia sub forma:

$$\mathbf{x} = F(\mathbf{x}), \tag{8.1}$$

în care F se numeşte *aplicaţie cu punct fix*, iar la fiecare pas k al algoritmului, se determină noua soluţie numerică din relaţia:

$$\mathbf{x}^k = F(\mathbf{x}^{k-1}). \tag{8.2}$$

Pentru a aduce sistemul de rezolvat

$$\mathbf{A}\mathbf{x} = \mathbf{b} \tag{8.3}$$

la forma unei aplicaţii cu punct fix, în general se caută o descompunere a matricei $\mathbf{A}$ într-o diferenţă de două matrice:

$$\mathbf{A} = \mathbf{B} - \mathbf{C}, \tag{8.4}$$

sistemul putând fi astfel adus la forma:

$$\mathbf{x} = \mathbf{B}^{-1}(\mathbf{b} + \mathbf{C}\mathbf{x}), \tag{8.5}$$

soluţia la pasul k fiind:

$$\mathbf{x}^k = \mathbf{B}^{-1}(\mathbf{C}\mathbf{x}^{k-1} + \mathbf{b}), \tag{8.6}$$

sau

$$\mathbf{x}^k = \mathbf{M}\mathbf{x}^{k-1} + \mathbf{u}, \tag{8.7}$$

în care $\mathbf{M} = \mathbf{B}^{-1}\mathbf{C}$ se numeşte *matrice de iteraţie*, iar $\mathbf{u} = \mathbf{B}^{-1}\mathbf{b}$.

Una din problemele metodelor iterative este *convergenţa şirului de iteraţii*. Se demonstrează că o condiţie suficientă pentru ca metoda să fie convergentă este ca valorile proprii ale matricei de iteraţie $\mathbf{M} = \mathbf{B}^{-1}\mathbf{C}$ să fie toate, în modul, mai mici decât 1. Definind *raza de convergenţa* a matricei $\mathbf{M}$, $ro(\mathbf{M})$, ca fiind modulul celei mai mari valori proprii, condiţia de convergenţă se scrie:

$$ro(\mathbf{M}) < 1 \tag{8.8}$$

Această condiţie de convergenţă este corelată cu norma matricei de iteraţie $\mathbf{M}$. Se demonstrează că, pentru orice matrice, există următoarea relaţie între norma şi raza sa de convergenţă:

$$ro(\mathbf{M}) \leq \|\mathbf{M}\| \tag{8.9}$$

Prin urmare, dacă matricea $\mathbf{M}$ are norma subunitară ($\|\mathbf{M}\| < 1$), raza sa de convergenţă va fi şi ea mai mică decât 1, iar metoda iterativă va fi în acest caz convergentă.

Metodele iterative cele mai cunoscute sunt:

- metoda deplasărilor simultane (Jacobi),
- metoda deplasărilor succesive (Gauss-Seidel),
- metoda suprarelaxărilor succesive(Frankel-Young),
- metoda direcţiilor alternante (Peaceman-Rachford),
- metoda iteraţiilor bloc,
- metoda factorizării incomplete,
- metoda Southwell.

Metoda Jacobi a deplasărilor simultane constă în alegerea partitiţiei matricei $\mathbf{A}$ astfel: $\mathbf{B}$ este matricea alcătuită din elementele diagonale ale lui $\mathbf{A}$

$$\mathbf{B} = \mathbf{D}, \tag{8.10}$$

iar matricea $\mathbf{C}$ conţine restul elementelor din matricea $\mathbf{A}$, luate cu semn schimbat:

$$\mathbf{C} = -(\mathbf{L} + \mathbf{U}), \tag{8.11}$$

în care s-au notat cu $\mathbf{L}$ şi $\mathbf{U}$ triunghiul inferior, respectiv cel superior din $\mathbf{A}$. Cu această descompunere, vectorul soluţie la fiecare pas k va avea expresia:

$$\mathbf{x}^k = \mathbf{D}^{-1}(\mathbf{b} - (\mathbf{L} + \mathbf{U})\mathbf{x}^{k-1}), \tag{8.12}$$

iar matricea de iteraţie $\mathbf{M}$ va fi:

$$\mathbf{M} = -\mathbf{D}^{-1}(\mathbf{L} + \mathbf{U}). \tag{8.13}$$

Se consideră linia i a sistemului de ecuaţii care trebuie rezolvat:

$$a_{i1}x_1 + a_{i2}x_2 + \ldots + a_{ii}x_i + \ldots + a_{in}x_n = b_i \tag{8.14}$$

Partiţionarea aleasă pentru matricea $\mathbf{A}$ revine la a determina pe x_i la pasul curent k din ecuaţia i, cu relaţia:

$$x_i^k = \frac{b_i - \sum_{j=1, j\neq i}^{n} a_{ij}x_j^{k-1}}{a_{ii}}, \tag{8.15}$$

în care x_i din membrul stâng reprezintă componenta i a noii soluţii, iar x_j din membrul drept sunt valorile obţinute la precedentul pas al iteraţiei. Se observă că, pentru determinarea noii soluţii, trebuie cunoscute, pe tot parcursul iteraţiei k, valorile soluţiei de la pasul anterior $k-1$.

În metoda *Gauss-Seidel*, a deplasărilor succesive, partiţionarea se alege astfel:

$$\mathbf{B} = \mathbf{D} + \mathbf{L}, \tag{8.16}$$

matricea $\mathbf{B}$ conţinând astfel partea inferior triunghiulară a matricei $\mathbf{A}$ inclusiv diagonala, iar

$$\mathbf{C} = -\mathbf{U} \tag{8.17}$$

matricea $\mathbf{C}$ conţinând partea superior triunghiulară a matricei. Matricea de iteraţie va avea forma

$$\mathbf{M} = -(\mathbf{D} + \mathbf{L})^{-1}\mathbf{U}. \tag{8.18}$$

Această partiţionare presupune că în membrul stâng al ecuaţiei i din sistem rămân termenii care conţin $x_j, j \leq i$, iar în membrul drept trec toţi ceilalţi termeni:

$$a_{i1}x_1 + a_{i2}x_2 + \ldots + a_{ii}x_i = b_i - a_{i,i+1}x_{i+1} - \ldots - a_{in}x_n. \tag{8.19}$$

Ca şi până acum, componentele vectorului soluţie aflate în membrul stâng reprezintă valori "noi", calculate la pasul curent k, pe când componentele din membrul drept sunt cele calculate la pasul anterior. Din această relaţie rezultă valoarea componentei x_i la pasul curent:

$$x_i^k = \frac{b_i - \sum\limits_{j=1}^{i-1} a_{ij}x_j^{(k)} - \sum\limits_{j=i+1}^{n} a_{ij}x_j^{(k-1)}}{a_{ii}}. \tag{8.20}$$

Se observă că o componentă x_i a soluţiei la pasul k este calculată în funcţie de componentele precedente $1, \ldots, i-1$, deja calculate la pasul curent şi de următoarele componente $i+1, \ldots, n$, calculate la pasul precedent. Algoritmul nu necesită păstrarea vechii componente i, după ce cea nouă a fost calculată, de aceea x_i nou se poate plasa în memorie în aceeaşi locaţie ca şi vechea valoare. Astfel, algoritmul Gauss-Seidel nu necesită spaţiu pentru memorarea decât a unui vector soluţie, spre deosebire de algoritmul Jacobi, unde trebuiau memoraţi atât x nou cât şi x vechi. În metoda Gauss-Seidel, imediat ce o componentă a fost determinată, ea este folosită în calculele următoare, înlocuind valoarea veche care se pierde, idee cunoscută sub numele de *principiul lui Seidel*.

Una din problemele care apar la rezolvarea sistemelor de ecuaţii liniare prin metode iterative este *alegerea criteriului de oprire a procesului iterativ*. O metodă de a rezolva această problemă a criteriului de oprire constă în evaluarea, după fiecare iteraţie, a erorii Cauchy

$$e = \|\mathbf{x}^{nou} - \mathbf{x}^{vechi}\| \tag{8.21}$$

şi întreruperea calculelor atunci când această valoare devine mai mică decât eroarea impusă, ε.

În ceea ce priveşte *convergenţa* metodelor, se demonstrează că la metodele Jacobi şi Gauss-Seidel, o condiţie suficientă ca metodele să fie convergente este ca matricea $\mathbf{A}$ a sistemului să fie diagonal dominantă, adică

$$|a_{ii}| > \sum_{j \neq i} |a_{ij}| \quad \forall i. \tag{8.22}$$

Desigur, aşa cum s-a arătat, condiţia de mai sus este echivalentă cu impunerea condiţiei ca norma matricei de iteraţie să fie subunitară, procesul iterativ fiind cu atât mai rapid convergent cu cât norma matricei de iteraţie este mai mică. În cazul matricelor simetrice şi pozitiv definite, metoda Gauss-Seidel este de aproximativ 2 ori mai rapidă decât metoda Jacobi. Acest avantaj, corelat şi cu necesitatea memorării unui singur vector soluţie, face ca metoda Gauss-Seidel să fie preferabilă metodei Jacobi din toate punctele de vedere.

8.3 Pseudocodul algoritmilor

Următoarea procedură permite rezolvarea sistemelor de ecuatii liniare prin metoda *Jacobi*.

```
procedura Jacobi (n,a,b,x,nrit,eps)
    tablou real a(n,n),b(n),x(n)
    întreg nrit
    real eps
    tablou real xn(N)                 ; x nou
    ; inițializări
    k = 0                             ; contor iterații
    pentru i=1,n
        x_i = 0                       ; inițializarea soluției
    ; iterații
    repetă                            ; parcurge iterațiile
        err= 0;                       ; eroarea la pasul curent
        pentru i= 1,n                 ; parcurge ecuațiile
            s = b(i)
```

```
                    pentru j=1,n                  ; parcurge ecuaţia i
                            s = s - a(i,j)x(j)
                    s = s + a(i,i)x(i)
                    xn(i) = s/a(i,i)              ; x nou
                    s = |xn(i) - x(i)|
                    dacă err < s atunci err = s
            pentru i= 1,n
                    x(i) = xn(i)                  ; înlocuieşte x vechi cu x nou
            k = k + 1                             ; incrementează contor iteraţii
        până când (err < eps) sau (k > nrit)
retur
```

Următoarea procedură permite rezolvarea sistemelor de ecuaţii liniare prin metoda *Gauss-Seidel*

```
procedura Gauss-Seidel(n,a,b,x,nrit,eps)
        tablou real a(n,n), b(n), x(n)
        întreg nrit
        real eps
        iniţializări
        k = 0                                     ; contor iteraţii
        pentru i=1,n
                x(i) = 0                          ; iniţializarea soluţiei
                ; iteraţii
                repetă                            ; parcurge iteraţiile
                err = 0                           ; eroarea la pasul curent
                pentru i=1,n                      ; parcurge ecuaţiile
                        s = b(i)
                        pentru j=1,n              ; parcurge ecuaţia i
                                s = s - a(i,j)x(j)
                        s = (s + a(i,i)x(i))/a(i,i)
                        err = err + (s - x(i))
                        x(i) = s                  ; x nou
                k = k + 1                         ; incrementează contor iteraţii
                err = sqrt(err)
        până când (err < eps) sau (k > nrit)
retur
```

Procedurile Jacobi şi Gauss-Seidel au parametrii:

- de intrare
  - `n` = dimensiunea sistemului;
  - `a(n,n)`= matricea sistemului;
  - `b(n)` = vectorul termenilor liberi;
  - `nrit` = numărul maxim de iteraţii;
  - `eps` = eroarea admisă;
- de ieşire `x(n)` = vectorul soluţie.

Pentru a demonstra modul diferit în care se poate evalua eroarea, în algoritmul Jacobi s-a folosit norma Cebîşev a erorii, iar în algoritmul Gauss-Seidel norma Euclidiană.

8.4 Analiza algoritmilor

Efort de calcul

Pentru o iteraţie, ordinul de complexitate al metodelor Jacobi şi Gauss-Seidel este $\mathrm{O}(n(n+2))$. Efortul de calcul pentru rezolvarea întregului sistem de ecuaţii liniare prin metode iterative este de ordinul $\mathrm{O}(mn^2)$, în care numărul total de iteraţii m care vor fi efectuate nu este în general cunoscut dinainte. Efortul de calcul depinde de norma matricei de iteraţie, fiind cu atât mai mic cu cât norma este mai mică.

Necesar de memorie

Pentru memorarea matricei sistemului, a vectorilor termen liber şi soluţie sunt necesare n^2+2n locaţii de memorie. În plus, la algoritmul Jacobi mai sunt necesare n locaţii pentru memorarea soluţiei obţinute la pasul curent.

Dacă matricea sistemului este o matrice rară, atunci metodele iterative se dovedesc extrem de eficiente din punctul de vedere al memoriei, ele negenerând umpleri.

Analiza erorilor

Spre deosebire de metodele directe, la care singurele erori care apar sunt cele de rotunjire, la metodele iterative apar şi erori de trunchiere prin reţinerea din şirul convergent către soluţia exactă, a unui număr finit de termeni. Datorită convergenţei lor, metodele iterative au proprietatea remarcabilă de a corecta erorile de rotunjire apărute pe parcurs.

Eroarea absolută la iterţia k este de cel puţin $\|M\|$ ori mai mică decât eroarea de la pasul anterior:

$$e_k = \|\mathbf{x}^k - \mathbf{x}\| \leq \|\mathbf{M}\|\|\mathbf{x}^{k-1} - \mathbf{x}\| = \|\mathbf{M}\|e_{k-1} \leq \|\mathbf{M}\|^k e_0. \tag{8.23}$$

Se constată că eroarea finală depinde de eroarea iniţială, de numărul de iteraţii efectuate şi de norma matricei de iteraţie, care determină viteza de convergenţă. Aceeaşi relaţie este valabilă şi pentru reziduul $\|\mathbf{A}\mathbf{x}^k - \mathbf{b}\|$.

8.5 Chestiuni de studiat

- Rezolvarea unor sisteme de ecuaţii liniare prin metodele Jacobi şi Gauss-Seidel;
- Analiza experimentală a erorilor şi a efortului de calcul, la metodele Jacobi şi Gauss-Seidel;
- Implementarea algoritmilor;
- Căutare de informaţii pe Internet.

8.6 Mod de lucru

Pentru desfăşurarea lucrării se selectează lucrarea *Metode iterative de rezolvare a sistemelor algebrice liniare* din meniul general de lucrări.

Aceasta are ca efect afişarea meniului:

1. Rezolvare de sisteme cu metodele Jacobi/Gauss-Seidel
2. Analiza algoritmilor

8.6.1 Rezolvarea unor sisteme de ecuaţii liniare prin metodele Jacobi/Gauss-Seidel

Programul lansat permite rezolvarea unor sisteme liniare prin metode iterative.

Se vor introduce:

- numărul de ecuaţii;

- elementele $a(i,j)$ ale matricei sistemului;
- termenii liberi $b(i)$;
- eroarea admisă;
- numărul maxim de iteraţii admis.

Programul afişează în consola Scilab informaţii despre procesul iterativ: dacă a fost convergent sau nu, norma matricei de iteraţie, în cazul în care procesul a fost convergent, câte iteraţii au fost necesare şi care este soluţia.

Se vor introduce parametrii necesari pentru rezolvarea următoarelor sisteme:

$$\left\{ \begin{array}{rcrcr} x_1 & + & x_2 & = & 5 \\ 2x_1 & + & 3x_2 & = & 13 \end{array} \right. \quad : R \quad \begin{array}{l} x_1 = 2 \\ x_2 = 3 \end{array}$$

Se va inversa ordinea ecuaţiilor şi se va comenta efectul asupra soluţiei.

$$\left\{ \begin{array}{rcrcrcr} 8x_1 & + & 2x_2 & + & x_3 & = & 15 \\ 10x_1 & + & 4x_2 & + & x_3 & = & 21 \\ 50x_1 & + & 25x_2 & + & 8x_3 & = & 124 \end{array} \right. \quad R: \begin{array}{l} x_1 = 1 \\ x_2 = 2 \\ x_3 = 3 \end{array}$$

$$\left\{ \begin{array}{rcrcrcr} 2x_1 & + & x_2 & + & x_3 & = & 7 \\ x_1 & + & 2x_2 & + & x_3 & = & 8 \\ x_1 & + & x_2 & + & x_3 & = & 6 \end{array} \right. \quad R: \begin{array}{l} x_1 = 1 \\ x_2 = 2 \\ x_3 = 3 \end{array}$$

$$\left\{ \begin{array}{rcrcrcr} 3x_1 & + & & & x_3 & = & 10 \\ x_1 & + & 2x_2 & + & x_3 & = & 12 \\ x_1 & + & x_2 & + & 2x_3 & = & -3 \end{array} \right. \quad R: \begin{array}{l} x_1 = 2 \\ x_2 = 3 \\ x_3 = 4 \end{array}$$

Se va inversa ordinea a două ecuaţii din sistem şi se va comenta efectul asupra soluţiei.

Se vor comenta rezultatele obţinute, convergenţa metodelor pentru fiecare sistem rezolvat.

Document disponibil la `www.lmn.pub.ro/Education/methods/methods.html`

8.6.2 Analiza experimentală a algoritmilor

Se selectează opţiunea *Analiza algoritmilor.* Aceasta are ca efect afişarea unui meniu cu opţiunile:

- Eroarea în funcţie de numărul de iteraţii;
- Numărul de iteraţii în funcţie de norma matricei de iteraţie.

La selectarea opţiunii *Eroarea în funcţie de numărul de iteraţii*, utilizatorul introduce: norma matricei de iteraţie Jacobi şi numărul de ecuaţii (valoare iniţială, valoare finală, pas). Astfel se apelează un program care rezolvă, pe baza celor două metode iterative, sisteme de ecuaţii liniare generate aleator. Programul afişează rezultatele numerice în consola Scilab şi reprezintă grafic variaţia erorii Cauchy în funcţie de iteraţie. Valorile recomandate sunt: norma matricei de iteraţie Jacobi 0.5, dimensiunile sistemelor 20, 40, 60. Datele se pot nota într-un tabel de tipul:

iteraţii		1	5	9	13	17
n = 20	eroare J					
	eroare GS					
n = 40	eroare J					
	eroare GS					
n = 60	eroare J					
	eroare GS					

Se vor reprezenta pe acelaşi grafic erorile în funcţie de numărul de iteraţii. Se vor comenta rezultatele obţinute.

La selectarea opţiunii *Numărul de iteraţii în funcţie de norma matricei de iteraţie* utilizatorul alege dimensiunea n a sistemului şi eroarea de oprire. Programul afişează numărul de iteraţii necesar celor două metode pentru sisteme generate aleator, de dimensiune n, având norma matricei de iteraţie 0.1, 0.2, ..., 0.9. Datele se pot nota într-un tabel de tipul:

norma matricei Jacobi	0.1	0.2	...	0.9
nr. iteraţii Jacobi				
nr. iteraţii Gauss-Seidel				

Se vor reprezenta pe acelaşi grafic numărul de iteraţii în funcţie de norma matricei. Se vor comenta rezultatele obţinute.

8.6.3 Implementarea algoritmilor

Se va implementa pe calculator, în limbajul C, o procedură proprie de rezolvare a unui sistem de ecuaţii liniare, prin una din cele două metode iterative. Se va compila şi testa procedura, eliminându-se eventualele erori.

Se va scrie pseudocodul şi se va implementa pe calculator un program principal, care apelează procedura anterioară şi rezolvă sistemul de ecuaţii:

$$\left\{\begin{array}{rcrcrcr} 2x & + & & & z & = & 5 \\ x & + & y & + & z & = & 6 \\ & & y & + & 3z & = & 11 \end{array}\right. \qquad \begin{array}{rl} R: & x=1 \\ & y=2 \\ & z=3 \end{array}$$

Acest program va permite introducerea datelor de la consolă şi afişarea pe ecran a soluţiei. Se vor nota şi comenta rezultatele obţinute şi eventualele dificultăţi apărute pe parcursul lucrului.

8.6.4 Căutare de informaţii pe Internet

Se vor căuta pe Internet informaţii (coduri) legate de rezolvarea iterativă a sistemelor algebrice de ecuaţii prin metode iterative. Cuvinte cheie recomandate: *Iterative methods for algebraic systems*, *Jacobi*, *Gauss-Seidel.*

8.7 Întrebări şi probleme

1. Daţi o explicaţie calitativă pentru convergenţa mai rapidă a metodei Gauss-Seidel faţa de metoda Jacobi.

2. Există sisteme de ecuaţii, nedominant diagonale, care se pot rezolva totuşi prin metoda Gauss-Seidel, deşi metoda Jacobi nu este convergentă?

3. Care este explicaţia că inversând ordinea a 2 ecuaţii în sistemele propuse pentru rezolvare cu primul program demonstrativ, metodele iterative nu conduc la obţinerea soluţiei?

4. Cum scade eroarea în funcţie de numărul de iteraţii, la metodele Jacobi şi Gauss-Seidel? Cum este corelată scăderea erorii cu norma matricei de iteraţie?

5. Scrieţi pseudocodul unei proceduri care să genereze, cu ajutorul unui generator de numere aleatoare, o matrice a cărei normă este dată.

Document disponibil la `www.lmn.pub.ro/Education/methods/methods.html`

6. Care sunt matricele $\mathbf{B}$ şi $\mathbf{C}$ în care s-a partiţionat matricea sistemului la o metodă iterativă care ar folosi următoarea expresie pentru determinare a componentei i a soluţiei :
$$x_i^k = \frac{b_i - \sum_{j=i+1}^{n} a_{ij} x_j^k - \sum_{j=1}^{i-1} a_{ij} x_j^{k-1}}{a_{ii}}.$$

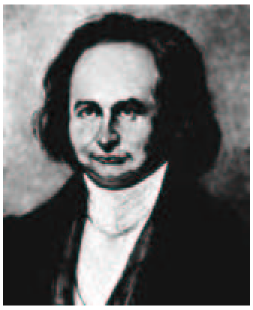
7. Scrieţi pseudocodul unui polialgoritm de rezolvare a unui sistem de ecuaţii liniare prin metode directe a cărui soluţie este rafinată ulterior prin metode iterative, în vederea eliminării erorilor de rotunjire.

8. Analizaţi teoretic modul în care depinde numărul de iteraţii m, necesare atingerii unei precizii dorite, în funcţie de norma matricei $\mathbf{M}$.

9. Modificaţi pseudocodul algoritmului Gauss-Seidel prin adoptarea unui criteriu de eroare relativ la norma reziduului $\|\mathbf{Ax} - \mathbf{b}\|$.

10. Scrieţi un algoritm de rezolvare iterativă a unui sistem cu matrice tridiagonală.

11. Scrieţi un algoritm de rezolvare iterativă a sistemului $\mathbf{Ax} = \mathbf{b}$ bazat pe partiţia:
$$\mathbf{A} = \mathbf{B} - \mathbf{C} = \mathbf{L} + \mathbf{D} - (-\mathbf{U}),$$
cu $\mathbf{D}$ matrice bloc diagonală.



Carl Gustav Jacob Jacobi

Born: 10 Dec 1804 in Potsdam, Prussia (now Germany)
Died: 18 Feb 1851 in Berlin, Germany
http://www-gap.dcs.st-and.ac.uk/~history/Mathematicians/Jacobi.html
http://scienceworld.wolfram.com/biography/Jacobi.html

Philipp Ludwig von Seidel

Born: 24 Oct 1821 in Zweibrcken, Germany
Died: 13 Aug 1896 in Munich, Germany
http://www-gap.dcs.st-and.ac.uk/~history/Mathematicians/Seidel.html

Augustin Louis Cauchy

Born: 21 Aug 1789 in Paris, France
Died: 23 May 1857 in Sceaux (near Paris), France
http://www-gap.dcs.st-and.ac.uk/~history/Mathematicians/Cauchy.html
http://scienceworld.wolfram.com/biography/Cauchy.html