Faculty of Civil Engineering Belgrade Master Study COMPUTATIONAL ENGINEERING

Faculty of Civil Engineering and Architecture Niš Doctoral Study

LECTURES

LESSON III

3. Linear Systems of Algebraic Equations: Iterative Methods

3.1 Introduction

Consider system of linear equations

$$\begin{aligned} a_{11}x_1 + a_{12}x_2 + \cdots + a_{1n}x_n &= b_1 \\ a_{21}x_1 + a_{22}x_2 + \cdots + a_{2n}x_n &= b_2 \\ &\vdots \\ a_{n1}x_1 + a_{n2}x_2 + \cdots + a_{nn}x_n &= b_n, \end{aligned} \tag{3.1.1}$$

which can be written in matrix form

$$\mathbf{A}\vec{x} = \vec{b}, \tag{3.1.2}$$

where

$$\mathbf{A} = \begin{bmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \vdots \\ a_{n1} & a_{n2} & \dots & a_{nn} \end{bmatrix}, \quad \vec{x} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}, \quad \vec{b} = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{bmatrix}.$$

In this chapter we always suppose that system (3.1.1), i.e. (3.1.2) has an unique solution.

Iterative methods for solving systems (3.1.2) have as goal determination of solution $\vec{x}$ with exactness given in advance. Namely, starting with arbitrary vector $\vec{x}^{(0)}$ $(= [x_1^{(0)} \dots x_n^{(0)}]^T)$ by iterative methods one defines the series $\vec{x}^{(k)}$ $(= [x_1^{(k)} \dots x_n^{(k)}]^T)$ such that

$$\lim_{k \to +\infty} \vec{x}^{(k)} = \vec{x}.$$

3.2 Simple iteration method

One of the most simplest methods for solving system of linear equations is method of simple iteration. For application of this method it is necessary to transform previously system (3.1.2) to the following equivalent form

$$\vec{x} = \mathbf{B}\vec{x} + \vec{\beta}. \tag{3.2.1}$$

Then, the method of simple iteration is given as

$$\vec{x}^{(k)} = \mathbf{B}\vec{x}^{(k-1)} + \vec{\beta} \quad (k = 1, 2, \dots). \tag{3.2.2}$$

Starting from arbitrary vector $\vec{x}^{(0)}$ and using (3.2.2) one generates series $\{\vec{x}^{(k)}\}$, which under some conditions converges to solution of given system.

If

$$\mathbf{B} = \begin{bmatrix} b_{11} & b_{12} & \dots & b_{1n} \\ b_{21} & b_{22} & \dots & b_{2n} \\ \vdots & & & \\ b_{n1} & b_{n2} & \dots & b_{nn} \end{bmatrix}, \quad \text{and} \quad \vec{\beta} = \begin{bmatrix} \beta_1 \\ \beta_2 \\ \vdots \\ \beta_n \end{bmatrix},$$

iterative method (3.2.2) can be written in scalar form

$$\begin{aligned} x_1^{(k)} &= b_{11}x_1^{(k-1)} + \ldots + b_{1n}x_n^{(k-1)} + \beta_1, \\ x_2^{(k)} &= b_{21}x_1^{(k-1)} + \ldots + b_{2n}x_n^{(k-1)} + \beta_2, \\ &\vdots \\ x_n^{(k)} &= b_{n1}x_1^{(k-1)} + \ldots + b_{nn}x_n^{(k-1)} + \beta_n, \end{aligned}$$

where $k = 1, 2, \ldots$.

One can prove (see [1]) that iterative process (3.2.2) converges if all eigenvalues of matrix $\mathbf{B}$ are by modulus less than one. Taking in account that determination of eigenvalues of matrix is rather complicated, in practical applications of method of simple iteration only sufficient convergence conditions are examined. Namely, for matrix $\mathbf{B}$ several norms can be defined, as for example,

$$\begin{aligned} \|\mathbf{B}\|_1 &= (\sum_{ij} b_{ij}^2)^{1/2}, \\ \|\mathbf{B}\|_2 &= \max_i \sum_{j=1}^{n} |b_{ij}|, \\ \|\mathbf{B}\|_3 &= \max_j \sum_{i=1}^{n} |b_{ij}|. \end{aligned} \tag{3.2.3}$$

It is not difficult to prove that iterative process (3.2.2) converges if $\|B\| < 1$, for arbitrary initial vector $\vec{x}^{(0)}$.

3.3 Gauss-Seidel method

Gauss-Seidel method is constructed by modification of simple iterative method. As we have seen, at simple iteration method, the value of i-th component $x_i^{(k)}$ of vector $\vec{x}^{(k)}$ is obtained from values $x_1^{(k-1)}, \ldots, x_n^{(k-1)}$, i.e.

$$x_i^{(k)} = \sum_{j=1}^{n} b_{ij}x_j^{(k-1)} + \beta_i \quad (i = 1, \ldots, n;\ k = 1, 2, \ldots).$$

This method can be modified in this way so that for computation of $x_i^{(k)}$ are used all previously computed values $x_1^{(k)}, \ldots, x_{i-1}^{(k)}, x_i^{(k-1)}, \ldots, x_n^{(k-1)}$ and the rest will be part of vector , obtained in previous iteration, i.e

$$x_i^{(k)} = \sum_{j=1}^{i-1} b_{ij}x_j^{(k)} + \sum_{j=1}^{n} b_{ij}x_j^{(k-1)} + \vec{\beta}_i \quad (i = 1, \ldots, n;\ k = 1, 2, \ldots). \tag{3.3.1}$$

Noted modification of simple iterative method is known as Gauss-Seidel method.

The iterative process (3.3.1) can be written in matrix form too. Namely, let

$$\mathbf{B} = \mathbf{B}_1 + \mathbf{B}_2,$$

where

$$\mathbf{B}_1 = \begin{bmatrix} 0 & 0 & \dots & 0 & 0 \\ b_{21} & 0 & & 0 & 0 \\ \vdots & & & & \\ b_{n1} & b_{n2} & & b_{n,n-1} & 0 \end{bmatrix}, \quad \mathbf{B}_2 = \begin{bmatrix} b_{11} & b_{12} & \dots & b_{1n} \\ 0 & b_{22} & & b_{2n} \\ \vdots & & & \\ 0 & 0 & & b_{nn} \end{bmatrix}.$$

Then (3.3.1) becomes

(3.3.2) $$\vec{x}^{(k)} = \mathbf{B}_1\vec{x}^{(k)} + \mathbf{B}_2\vec{x}^{(k-1)} + \vec{\beta} \quad (k = 1, 2, \dots).$$

Theorem 3.3.1. *For arbitrary vector* $\vec{x}^{(0)}$ *, iterative process* (3.3.2) *converges then and only then if all roots of equation*

$$\det[\mathbf{B}_2 - (\mathbf{I} - \mathbf{B}_1)\lambda] = \begin{bmatrix} b_{11} - \lambda & b_{12} & \dots & b_{1n} \\ b_{21}\lambda & b_{22} - \lambda & & b_{2n} \\ \vdots & & & \\ b_{n1}\lambda & b_{n2}\lambda & & b_{nn} - \lambda \end{bmatrix} = 0$$

are by modulus less than one.

3.4 Program realization

Program 3.4.1. Let's write a program for solving a system of linear equations of form $\vec{x} = \mathbf{B}\vec{x} + \vec{\beta}$, by simple iteration method. Because this method converges when norm of matrix $\mathbf{B}$ is less than one, for examination of this condition we will write a subroutine `NORMA`, in which, depending on k, are computed norms $(k = 1, 2, 3)$ in accordance with formula (3.2.3). Parameters in list of parameters are of following meaning:

`A` - matrix stored as vector, which norm is to be calculated;
`N` - order of matrix;
`K` - number which defines norm `(K=1,2,3)`;
`ANOR` - corresponding norm of matrix `A`.

```
      SUBROUTINE NORMA(A,N,K,ANOR)
      DIMENSION A(1)
      NU=N*N
      ANOR=0
      GO TO (10, 20,40),K
   10 DO 15 I=1,NU
   15 ANOR=ANOR+A(I)**2
      ANOR=SQRT(ANOR)
      RETURN
   20 DO 25 I=1,N
      L=-N
      S=0.
      DO 30 J=1,N
      L=L+N
      IA=L+I
   30 S=S+ABS(A(IA))
      IF(ANOR-S) 35,25,25
   35 ANOR=S
   25 CONTINUE
      RETURN
   40 L=-N
      DO 50 J=1,N
      S=0.
      L=L+N
      DO 45 I=1,N
      LI=L+I
   45 S=S+ABS(A(LI))
      IF(ANOR-S) 55,50,50
```

```
55   ANOR=S
50   CONTINUE
     RETURN
     END
```

Main program is organized in this way that before iteration process begins, the convergence is examined. Namely, if at least one norm satisfies $\|\mathbf{B}\|_k < 1$ $(k = 1, 2, 3)$, iterative process proceeds, and if not, the message that convergence conditions are not fulfilled is printed and program terminates.

For multiplying matrix **B** by vector $\vec{x}^{(k+1)}$ we use subroutine **MMAT**, which is given in 2.2.5.2. As initial vector we take $\vec{x}^{(0)}$.

As criteria for termination of iterative process we adopted

$$|x_i^{(k)} - x_i^{(k-1)}| \le \varepsilon \quad (i = 1, \ldots, n).$$

On output we print the last iteration which fulfills above given criteria.

```
      DIMENSION B(100), BETA(10), X(10), X1(10)
      OPEN(8,FILE='ITER.IN')
      OPEN(5,FILE='ITER.OUT')
      READ(8,5) N, EPS
    5 FORMAT(I2,E5.0)
      NN=N*N
      READ(8,10)(B(I),I=1,NN),(BETA(I),I=1,N)
   10 FORMAT(16F5.1)
      WRITE(5,13)
   13 FORMAT(1H1,5X,'MATRICA B', 24X,'VEKTOR BETA')
      DO 15 I=1,N
   15 WRITE(5,20) (B(J),J=I,NN,N),BETA(I)
   20 FORMAT(/2X,4F8.1,5X,F8.1)
      DO 30 K=1,3
      CALL NORMA(B,N,K,ANOR)
      IF(ANOR-1.) 25,30,30
   30 CONTINUE
      WRITE(5,35)
   35 FORMAT(5X,'USLOVI ZA KONVERGENCIJU'
     1' NISU ZADOVOLJENI')
      GO TO 75
   25 ITER=0
      DO 40 I=1,N
   40 X(I)=BETA(I)
   62 ITER=ITER+1
      CALL MMAT(B,X,X1,N,N,1)
      DO 45 I=1,N
   45 X1(I)=X1(I)+BETA(I)
      DO 55 I=1,N
      IF(ABS(X1(I)-X(I))-EPS)55,55,60
   55 CONTINUE
      WRITE(5,42)ITER
   42 FORMAT(/3X,I3,'.ITERACIJA'/)
      WRITE(5,50)(I,X1(I),I=1,N)
   50 FORMAT(3X,4(1X,'X(',I2,')=',F9.5))
      GO TO 75
   60 DO 65 I=1,N
   65 X(I)=X1(I)
      GO TO 62
   75 CLOSE(8)
      CLOSE(5)
      STOP
      END
```

Taking accuracy $\varepsilon = 10^{-5}$, for one concrete system of equation of fourth degree (see output listing) we get the solution in fourteenth iteration.

```
  MATRICA B                                VEKTOR BETA
  -.1      .4      .1      .1                .7
   .4     -.1      .1      .1                .7
   .1      .1     -.2      .2               1.2
   .1      .1      .2     -.2              -1.6
 14.ITERACIJA
X( 1)=  1.00000 X( 2)=  1.00000 X( 3)=  1.00000
X( 4)= -1.00000
```

Program 3.4.2. Write a code for obtaining a matrix $\mathbf{S}, = e^{\mathbf{A}}$ where $\mathbf{A}$ is given square matrix of order n, by using formula

$$e^{\mathbf{A}} = \sum_{k=0}^{+\infty} \frac{1}{k!} \mathbf{A}^k. \tag{3.4.2.1}$$

Let S_k be k-th partial sum of series (3.4.2.1), and U_k its general member. Then the equalities

$$U_k = \frac{1}{k} U_{k-1} \mathbf{A}, \quad S_k = S_{k-1} + U_k \quad (k = 1, 2, \ldots). \tag{3.4.2.2}$$

hold, whereby $\mathbf{U}_0 = \mathbf{S}_0 = \mathbf{I}$ (unity matrix of order n). By using equality (3.4.2.2) one can write a program for summation of series (3.4.2.1), where we usually take as criteria for termination of summation the case when norm of matrix is lesser than in advance given small positive number ε. In our case we will take norm $\|.\|_2$ (see formula (3.2.3)) and $\varepsilon = 10^{-5}$.

By using subroutine MMAT for matrices multiplication and subroutine NORMA for calculation of matrix norm, we have written the following program for obtaining the matrix $e^{\mathbf{A}}$

```
C==================================================
C   ODREDJIVANJE MATRICE EXP(A)
C==================================================
      DIMENSION A(100), S(100), U(100), P(100)
      OPEN(8,FILE='EXPA.IN')
      OPEN(5,FILE='EXPA.OUT')
      READ(8,10) N,EPS
   10 FORMAT(I2,E5.0)
      NN=N*N
      READ(8,15)(A(I),I=1,NN)
   15 FORMAT(16F5.0)
C  FORMIRANJE JEDINICNE MATRICE
      DO 20 I=1,NN
      S(I)=0.
   20 U(I)=0.
      N1=N+1
      DO 25 I=1,NN,N1
      S(I)=1.
  25  U(I)=1.
C SUMIRANJE MATRICNOG REDA
      K=0
  30  K=K+1
      CALL MMAT(U,A,P,N,N,N)
      B=1./K
      DO 35 I=1,NN
      U(I)=B*P(I)
   35 S(I)=S(I)+U(I)
C ISPITIVANJE USLOVA ZA PREKID SUMIRANJA
      CALL NORMA(U,N,2,ANOR)
      IF(ANOR.GT.EPS)GO TO 30
      WRITE(5,40) ((A(I),I=J,NN,N),J=1,N)
   40 FORMAT(2X,<5*N-9>X,'M A T R I C A      A'
     1 //(<N>F10.5))
      WRITE(5,45)((S(I),I=J,NN,N),J=1,N)
```

```
   45 FORMAT(//<5*n-9>X,'M A T R I C A   EXP(A)'
     1 //(<N>F10.5))
      CLOSE(8)
      CLOSE(5)
      END
```

This program has been tested on the example

$$\mathbf{A} = \begin{bmatrix} 4 & 3 & -3 \\ 2 & 3 & -2 \\ 4 & 4 & -3 \end{bmatrix},$$

for which can be obtained analytically

$$\mathbf{A} = \begin{bmatrix} 3e-2 & 3e-3 & -3e+3 \\ 2e-2 & 2e-1 & -2e+2 \\ 4e-4 & 4e-4 & -4e+5 \end{bmatrix}. \tag{3.4.2.3}$$

Output listing is of form

```
          M A T R I C A      A
     4.00000
     3.00000
    -3.00000
     2.00000
     3.00000
    -2.00000
     4.00000
     4.00000
    -3.00000
        M A T R I C A   EXP(A)
    16.73060
    14.01232
   -14.01232
     9.34155
    12.05983
    -9.34155
    18.68310
    18.68310
   -15.96482
```

By using (3.4.2.3) it is not hard to prove that all figures in obtained results are exact.

It is suggested to readers to write a code for previous problem using program `Mathematica`.

Bibliography

[1] Milovanović, G.V., *Numerical Analysis I*, Naučna knjiga, Beograd, 1988 (Serbian).

[2] Milovanović, G.V. and Djordjević, Dj.R., *Programiranje numeričkih metoda na FORTRAN jeziku.* Institut za dokumentaciju zaštite na radu "Edvard Kardelj", Niš, 1981 (Serbian).

(The full list of references and further reading is given on the end of Chapter 4.)