Faculty of Civil Engineering — Faculty of Civil Engineering and Architecture
Belgrade — Niš
Master Study — Doctoral Study
COMPUTATIONAL ENGINEERING

LECTURES

LESSON IV

4. Linear Systems of Algebraic Equations: Orthogonalization, Gradient Methods, Relaxation Methods

4.1. Orthogonalization method

Consider the system of linear algebraic equations with regular matrix.

$$
\begin{aligned}
a_{11}x_1 + a_{12}x_2 + \cdots + a_{1n}x_n &= b_1\\
a_{21}x_1 + a_{22}x_2 + \cdots + a_{2n}x_n &= b_2\\
&\vdots\\
a_{n1}x_1 + a_{n2}x_2 + \cdots + a_{nn}x_n &= b_n,
\end{aligned}
\tag{4.1.1}
$$

or, in matrix form

$$\mathbf{A}\vec{x} = \vec{b}. \tag{4.1.2}$$

If we define $(n+1)$-dimensional vectors of form

$$\vec{y} = \begin{bmatrix} x_1\\ x_2\\ \vdots\\ x_n\\ 1\end{bmatrix}, \quad \vec{a}_i = \begin{bmatrix} a_{i1}\\ a_{i2}\\ \vdots\\ a_{in}\\ -b_i\end{bmatrix}, \qquad (i = 1, \dots, n),$$

then this system can be written in the form

$$(\vec{a}_i, \vec{y}) = \vec{y}^T\vec{a}_i = 0 \qquad (i = 1, \dots, n). \tag{4.1.3}$$

The equations (4.1.3) point out to the possibility of solving system (4.1.1) by using orthogonality of vector $\vec{y}$ to vectors $\vec{a}_i$ $(i = 1, \dots, n)$. The mentioned orthogonality is equivalent to orthogonality of vector $\vec{y}$ to linear subspace H_n, which is generated by vectors $\vec{a}_i$ $(i = 1, \dots, n)$, i.e. $H_n = L(\vec{a}_1, \dots, \vec{a}_n)$. Staring from basis $\{\vec{a}_1, \dots, \vec{a}_n\}$, by means of Gram-Schmidt procedure, we construct orthonormed basis $B_n = \{\vec{v}_1, \dots, \vec{v}_n\}$ of semi-space H_n. Vector $\vec{y}$ is obviously orthogonal to all vectors of orthonormed basis B_n.

Vector $\vec{a}_{n+1} = [0\ \ 0\ \ \dots\ \ 0\ \ 1]^T$, being linearly independent to vectors of basis B_n, shell be orthogonalized too. Thus,

$$\vec{u} = \vec{a}_{n+1} - \sum_{i=1}^{n} (\vec{a}_{n+1}, \vec{v}_i)\vec{v}_i.$$

Let the coordinates of vector $\vec{u}$ be, in turn, $u_1, \ldots, u_n, u_{n+1}$, i.e. $\vec{u} = [u_1 \;\; u_2 \;\; \ldots \;\; u_n \;\; u_{n+1}]^T$. Because vector $\vec{u}$ satisfies orthogonality conditions $(\vec{u}, \vec{v}_i) = 0 \quad (i = 1, \ldots, n)$, we conclude that also $(\vec{u}, \vec{a}_i) = 0 \quad (i = 1, \ldots, n)$, i.e.

$$\begin{aligned}
(\vec{u}, \vec{a}_1) &= a_{11}u_1 + a_{12}u_2 + \cdots + a_{1n}u_n - b_1 u_{n+1} = 0\\
(\vec{u}, \vec{a}_2) &= a_{21}u_1 + a_{22}u_2 + \cdots + a_{2n}u_n - b_2 u_{n+1} = 0\\
&\vdots\\
(\vec{u}, \vec{a}_n) &= a_{n1}u_1 + a_{n2}u_2 + \cdots + a_{nn}u_n - b_n u_{n+1} = 0.
\end{aligned}$$

Note that $u_{n+1} \neq 0$. Namely, if would be $u_{n+1} = 0$, then $n-$tuple $(u_1, u_2, \ldots, u_n)$ would be solution of homogeneous system of equations with matrix $\mathbf{A} = [a_{ij}]$. Nevertheless, because homogeneous system ($\det \mathbf{A} \neq 0$) has only trivial solutions, we would have $u_i = 0 \quad (i = 1, \ldots, n)$, and vector $\vec{a}_{n+1}$ would be linear combination of vectors of basis B_n, what is in contradiction to choice of this vector.

If we divide all equations of last system with u_{n+1}, it is easy to conclude that vector $\vec{y}$, with $x_i = \dfrac{u_i}{u_{n+1}} \quad (i = 1, \ldots, n)$, is solution of system of equation (4.1.3). Then $n-$tuple $(x_1, \ldots, x_n)$ is solution of system (4.1.1).

Finally, note that

(1) Given method of orthogonalization demands greater number of multiplications and dividing than Gauss method of elimination;

(2) Orthogonalization can be done by solving of system of linear algebraic equation.

4.2. Relaxation methods

Let $\vec{x}_p$ be approximative solution of system of equations

$$\mathbf{A}\vec{x} = \vec{b}. \tag{4.2.1}$$

Because exact solution $\vec{x} = \mathbf{A}^{-1}\vec{b}$ is unknown, it is a question in which extent $\vec{x}_p$ satisfies given system of equations. As a measure of this, it is usual used norm of vector of residuum

$$\vec{r} = \mathbf{A}\vec{x}_p - \vec{b}. \tag{4.2.2}$$

If $\vec{r}$ is zero-vector, it is easy to see that $\vec{x}_p$ is exact solution of system (4.2.1). Based on previous, one can conclude that system (4.2.1) is almost satisfied if the components of vector r are close to zero. Nevertheless, last statement is not always valid. Namely, at weak conditioned systems, norm of vector $\vec{x}_p - \vec{\mathbf{A}}^{-1}\vec{b}$ can be large, in the case when norm of residuum vector small, regarding to $\vec{x}_p - \mathbf{A}^{-1}\vec{b} = \mathbf{A}^{-1}\vec{r}$.

In spite of this deficiency, vector $\vec{r}$ is of great importance in broad class of iterative methods, so known relaxation methods.

Relaxation method is every one at which next approximation of solution is obtained on the base of previous approximation and residuum vector (In general case it depends on several previous approximations), which is used as indicator for magnitude of correction.

First ideas on relaxation methods were given by Gauss. Nevertheless, systematic research in this field is done in second half of last century. Being very convenient for systems with large numbers of equations, they are applied increasingly for solving partial differential equations in their applications to technology problems, especially in Finite Elements Method (FEM). There exists reach bibliography on relaxation methods (see, for example, [5], [6], [7], [8], [9], [10]).

Successive-Over-Relaxation (SOR) is a method which generalizes one variant of Gauss-Seidel method for solving system $\mathbf{A}\vec{x} = \vec{b}$.

Let the matrix $\mathbf{D} = diag(\mathbf{A})$ be regular. Decompose matrix $\mathbf{A}$ to form

$$\mathbf{A} = \mathbf{C}_1 + \mathbf{D} + \mathbf{C}_2,$$

where

$$\mathbf{C}_1 = \begin{bmatrix} 0 & 0 & \dots & 0 & 0 \\ a_{21} & 0 & & 0 & 0 \\ \vdots & & & & \\ a_{n1} & a_{n2} & & a_{n,n-1} & 0 \end{bmatrix} \text{ and } \mathbf{C}_2 = \begin{bmatrix} 0 & a_{12} & \dots & a_{1,n} \\ 0 & 0 & & a_{2,n} \\ \vdots & & & \\ 0 & 0 & & 0 \end{bmatrix}.$$

If we take for computation of residuum vector $\vec{r}$ for x_i the last obtained value (like at Gauss-Seidel method), i.e.

$$\vec{r} = \mathbf{C}_1\vec{x}^{(k)} + (\mathbf{D} + \mathbf{C}_2)\vec{x}^{k-1} - \vec{b}$$

and if we put $\vec{\delta}^{(k)} = -\omega\vec{x}$, we can form iterative process of form

$$\mathbf{B}\vec{\delta}^{(k)} = -\omega\vec{x} \qquad (k = 1, 2, \ldots),$$

i.e.

$$\mathbf{D}\vec{x}^{(k)} = \mathbf{D}\vec{x}^{(k-1)} + \omega[\vec{b} - \mathbf{C}_1\vec{x}^{(k)} - (\mathbf{D} + \mathbf{C}_2)\vec{x}^{(k-1)}], \tag{4.2.3}$$

where $k = 1, 2, \ldots$ and ω real parameter. Parameter ω is called relaxation factor and can be, in general case, changed during computation.

Iterative process (4.2.3) can be presented in scalar form, as follows.

$$a_{ii}\overset{*}{x}_i^{(k)} = -\sum_{j<i} a_{ij}x_j^{(k)} - \sum_{j>i} a_{ij}x_j^{(k-1)} + b_i,$$
$$x_i^{(k)} = x_i^{(k-1)} + \omega(\overset{*}{x}_i^{(k)} - x_i^{(k-1)})$$

where $i = 1, \ldots, n$ and $k = 1, 2 \ldots$.

Note that for $\omega = 1$ (4.2.3) reduces to Gauss-Seidel process.

On the basis of (4.2.3) we have

$$(\mathbf{D} + \omega\mathbf{C}_1)\vec{x}^{(k)} = [(1-\omega)\mathbf{D} - \omega\xi]\vec{x}^{(k-1)} + \omega\vec{b} \quad (k = 1, 2, \ldots), \tag{4.2.4}$$

i.e.

$$\vec{x}^{(k)} = K(\omega)\vec{x}^{(k-1)} + \vec{f}(\omega) \qquad (k = 1, 2, \ldots),$$

where

$$K(\omega) = (\mathbf{D} + \omega\mathbf{C}_1)^{-1}[(1-\omega)\mathbf{D} - \omega\xi] \text{ and } \vec{f}(\omega) = \omega(\mathbf{D} + \omega\mathbf{C}_1)^{-1}\vec{b}.$$

Chebyshev semi-iterative method is a method connected to SOR method in a way described in this sub-section.

Given a system of equations

$$\vec{x} = \mathbf{B}\vec{x} + \vec{\beta} \tag{4.2.5}$$

with Hermitian matrix $\mathbf{B}$ of order n. If $\rho(\mathbf{B}) < 1$, the iterative process

$$\vec{x}^k = \mathbf{B}\vec{x}^{k-1} + \vec{\beta} \qquad (k = 1, 2, \ldots) \tag{4.2.6}$$

converges to solution $\vec{x}$ of system (4.2.5). In order to accelerate the convergence of process (4.2.6), consider linear combination of vector $\vec{x}^k$ (this procedure is often called linear acceleration), i.e.

$$\vec{y}^k = \sum_{i=0}^{k} c_{ki}\vec{x}^i \qquad (k = 0, 1, \ldots) \tag{4.2.7}$$

under condition $\sum_{i=0}^{k} c_{ki} = 1$.

If we put $\vec{\varepsilon}^{(k)} = \vec{x}^k - \vec{x}$ and $\vec{\xi}(k) = \vec{y}^k - \vec{x} \quad (k = 0, 1, \ldots)$, we have

$$\vec{\xi}(k) = \sum_{i=0}^{k} c_{ki}\vec{\varepsilon}^i = (\sum_{i=0}^{k} c_{ki}\mathbf{B}^i)\vec{\varepsilon}^{(0)},$$

i.e.

$$\vec{\xi}^k = Q_k(\mathbf{B})\vec{\varepsilon}^{(0)}, \tag{4.2.8}$$

where

$$Q_k(t) = \sum_{i=0}^{k} c_{ki}\vec{t}^i \qquad \text{and} \qquad Q_k(1) = 1. \tag{4.2.9}$$

Assume that $\{\vec{e}_i\}$ is orthonormed system of eigenvectors of Hermitian matrix $\mathbf{B}$, where $\mathbf{B}\vec{e}_i = \lambda_i\vec{e}_i \quad (\lambda_i \equiv \lambda_i(\mathbf{B})) \quad (i = 1, \ldots, n)$. If we develop vector $\vec{\varepsilon}^{(0)}$ by eigenvectors $\vec{e}_i$, i.e. $\vec{\varepsilon}^{(0)} = p_1\vec{e}_1 + \ldots + p_n\vec{e}_n$, the equality (4.2.8) becomes

$$\vec{\xi}(k) = \sum_{i=1}^{n} p_i Q_k(\lambda_i)\vec{e}_i.$$

Denote with $\mathcal{P}_k$ set of all polynomials $\mathbf{Q}_k$ of form (4.2.9), i.e. set of all polynomials of degree k with normalization $Q_k(1) = 1$. It can be shown that general three-term recurrence formula for generating of polynomials $Q_k \in \mathcal{P}_k$ is of form

$$Q_{k+1}(t) = (\alpha_k t + 1 - \alpha_k - \beta_k)Q_k(t) + \beta_k Q_{k-1}(t) \qquad (k = 0, 1, \ldots) \tag{4.2.10}$$

where $Q_0(t) = 1,\ \beta_0 = 0,\ \alpha_k$ and β_k are real numbers.

Because of

$$||\vec{\xi}^{(k)}||_E = \big(\sum_{i=1}^{n} p_i^2 Q_k(\lambda_i)^2\big)^{1/2} \le \max_i |Q_k(\lambda_i)| \big(\sum_{i=1}^{n} p_i^2\big)^{1/2},$$

i.e.

$$||\vec{\xi}^{(k)}||_E \le ||\vec{\varepsilon}^{\,(0)}||_E \max_i |Q_k(\lambda_i)|,$$

for choice of polynomials $(\tilde{Q}_k)$ we adopt a criteria

$$\min_{Q_k \in \mathcal{P}_k} \big(\max_{-\rho \le t \le \rho} |Q_k(t)|\big) = \max_{-\rho \le t \le \rho} |\tilde{Q}_k(t)|,$$

where $\rho \equiv \rho(\mathbf{B})$. Then we get

$$\tilde{Q}_k(t) = \frac{T_k(t/\rho)}{T_k(1/\rho)} \qquad (k = 0, 1, \ldots), \tag{4.2.11}$$

where T_k is Chebyshev polynomial of degree k. It is easy to show that in this case recurrence relation (4.2.10) reduces to

$$\tilde{Q}_{k+1}(t) = (\alpha_k t\tilde{Q}_k(t) + (1 - \alpha_k)\tilde{Q}_{k-1}(t) \qquad (k = 0, 1, \ldots) \tag{4.2.12}$$

where

(4.2.13). $$\alpha_0 = 1,\quad \beta_0 = 0,\ \alpha_k = \frac{2}{\rho}\cdot\frac{T_k(1/\rho)}{T_{k+1}(1/\rho)},\ \beta_k = 1-\alpha_k \qquad (k = 1,2,\ldots)$$

Thus, on basis (4.2.12) is generated set of polynomials $\{\tilde{Q}_k\}$, and then, based on (4.2.7) series $\{\vec{y}^{\,(k)}\}$. This procedure, or algorithm, is known as Chebyshev semi-iterative method.

The given method can be represented in explicit form (see [1], pp. 281, and [7]) as

(4.2.14). $$\vec{y}^{\,(k+1)} = \vec{y}^{\,(k-1)} + \alpha_k\big(\mathbf{B}\vec{y}^{\,(k)} + \vec{\beta} - \vec{y}^{\,(k-1)}\big) \qquad (k = 1,2,\ldots),$$

where series $\{\alpha_k\}$ is defined using (4.2.13) and

$$\vec{y}^{\,(0)} = \vec{x}^{\,(0)} \ \text{ and } \ \vec{y}^{\,(1)} = \mathbf{B}\vec{x}^{\,(0)} + \vec{\beta}.$$

On the basis of previous, members of series $\{\alpha_k\}$ are

$$\alpha_0 = 1,\ \alpha_1 = \frac{2}{2-\rho^2},\ \alpha_k = \frac{1}{1-\frac{1}{4}\rho^2\alpha_{k-1}} \qquad (k = 2,3,\ldots),$$

whereby

$$2 > \alpha_1 > \alpha_2 > \ldots \leq \lim_{k\to\infty}\alpha_k = \omega_{opt} = \frac{2}{1+\sqrt{1-\rho^2}} > 1.$$

Thus, α_k tends to optimal relaxation factor obtained by using SOR method.

4.3. Gradient methods

In the class of methods used at minimization of functionals the special role have gradient methods. In this section we will consider two gradient methods for solving system of linear equations

(4.3.1) $$\mathbf{A}\vec{x} = \vec{b},$$

where $\mathbf{A}$ is normal matrix. First method is Steepest Descent Method, and second is Method of Conjugate Gradients. Note that method of conjugate gradients by virtue is nit iterative method. Both methods are based on minimization of functional $F : R^n \to R$ defined by

(4.3.2) $$F(\vec{x}) = (\mathbf{A}\vec{x},\vec{x}) - 2(\vec{b},\vec{x}).$$

Because $\mathbf{A}$ is normal matrix, we have

$$F(\vec{x}) - F(\mathbf{A}^{-1}\vec{b}) = (\mathbf{A}(\vec{x}-\mathbf{A}^{-1}\vec{b}),\vec{x}-\mathbf{A}^{-1}\vec{b}) \geq 0,$$

we conclude that functional F reaches minimum for $\vec{x} = \mathbf{A}^{-1}\vec{b}$, what is solution of system (4.3.1).

Steepest Descent (SD) Method. Every method of form

(4.3.3) $$\vec{x}^{\,(k)} = \vec{x}^{\,(k-1)} - \alpha_{k-1}\, grad\, F\vec{x}^{\,(k-1)} \qquad (k = 1,2,\ldots)$$

is called Steepest Descent (SD) method.

If parameter α_p is determined from condition that

(4.3.4) $$F(\vec{x}^{\,(p)} - \alpha_p\, grad\, F(\vec{x}^{\,(p)})) \qquad (\forall p \in N_0)$$

is minimal, method (4.3.3) is called Steepest Descent method.

From (4.3.2) it follows $grad\, F(x) = 2(\mathbf{A}\vec{x} - \vec{b})$. Then

$$\vec{x}^{(k)} = \vec{x}^{(k-1)} - 2\alpha_{k-1}(\mathbf{A}\vec{x}^{(k-1)} - \vec{b}) \quad (k = 1, 2, \ldots). \tag{4.3.5}$$

If we in equation

$$F(\vec{x} + t\vec{r}) = F(\vec{x}) + 2t(\mathbf{A}\vec{x}^{(k-1)} - \vec{b}) \quad (k = 1, 2, \ldots).$$

proceed substitution

$$\begin{pmatrix} \vec{x} & \vec{r} & t \\ \vec{x}^{(p)} & \mathbf{A}\vec{x}^{(p)} - \vec{b} & -2\alpha_p \end{pmatrix} \quad (p \in N_0),$$

we obtain (4.3.4) in form

$$F(\vec{x}^{(p+1)}) = F(\vec{x}^{(p)}) - 4\alpha_p(\vec{r}^{(p)}, \vec{r}^{(p)}) + 4\alpha_p^2(\mathbf{A}\vec{r}^{(p)}, \vec{r}^{(p)}),$$

where $\vec{r}^{(p)} = \mathbf{A}\vec{x}^{(p)} - \vec{b}$.

Square trinomial $\alpha_p \to \Phi(\alpha_p) = F(\vec{x}^{(p+1)})$ has minimal value if

$$2\alpha_p = \frac{(\vec{r}^{(p)}, \vec{r}^{(p)})}{(\mathbf{A}\vec{r}^{(p)}, \vec{r}^{(p)})}. \tag{4.3.6}$$

On the basis of (4.3.5) and (4.3.6), we obtain SD method

$$\vec{x}^{(k)} = \vec{x}^{(k-1)} - \frac{||\mathbf{A}\vec{x}^{(k-1)} - \vec{b}||_E^2}{(\mathbf{A}(\mathbf{A}\vec{x}^{(k-1)} - \vec{b})(\mathbf{A}\vec{x}^{(k-1)} - \vec{b})}(\mathbf{A}\vec{x}^{(k-1)} - \vec{b}), \tag{4.3.7}$$

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

Theorem 4.3.1. *Let $\vec{x}$ be exact solution of system* (4.3.1), *which matrix* $\mathbf{A}$ *is normal. For steepest descent method the following inequalities hold.*

$$F(\vec{x}^{(k)}) - F(\vec{x}) \leq \Big(\frac{M - m}{M + m}\Big)^2 (F(\vec{x}^{(k-1)}) - F(\vec{x})), \tag{4.3.8}$$

$$||\vec{x}^{(k)} - \vec{x}|| \leq \Big(\frac{M - m}{M + m}\Big)^k \sqrt{\frac{M}{m}} ||\vec{x}^{(0)} - \vec{x}||_E, \tag{4.3.9}$$

where $\vec{x}^{(0)}$ is arbitrary initial vector, and $0 < m \leq \lambda_i(\mathbf{A}) \leq M$.

The proof of given theorem can be found in ([1], pp. 284-286).

<u>Method of Conjugate Gradients (CG)</u>. Theoretically considering (with no taking in account round-off error), CG method converges to exact solution in at most n iterations (n is matrix order). This method consists in construction of series $\{\vec{x}^{(k)}\}$ by using

$$\vec{x}^{(k)} = \vec{x}^{(0)} + \sum_{i=0}^{k-1} \alpha_i \mathbf{A}^i \vec{r}^{(0)} \qquad (k = 1, \ldots, n),$$

starting with arbitrary initial vector $\vec{x}^{(0)}$, whereby $\vec{r}^{(0)} = \mathbf{A}\vec{x}^{(0)} - \vec{b}$, whereas coefficients α_i are determined from condition that function $(\alpha_0, \ldots, \alpha_{k-1}) \to F(\vec{x}^{(k)})$ reaches minimum. For practical reason, in application of this method, it is efficient to construct series of vectors $\{\vec{p}^{(k)}\}$, being mutual conjugate in sense

$$(\vec{p}^{(i)}, \mathbf{A}\vec{p}^{(j)}) = 0 \qquad (i \neq j).$$

Then the induced method can be represented recursively using the following formulas:

$$\begin{aligned}
\vec{x}^{\,(k+1)} &= \vec{x}^{\,(k)} - c_k \vec{p}^{\,(k)} && (k = 0, 1, \ldots, n-1),\\
\vec{r}^{\,(k)} &= \mathbf{A}\vec{x}^{\,(k)} - \vec{b} && (k = 0, 1, \ldots, n),\\
\vec{p}^{\,(0)} &= -\vec{r}^{\,(0)},\ \vec{p}^{\,(k)} = -\vec{r}^{\,(k)} + q_k \vec{p}^{\,(k-1)} && (k = 1, \ldots, n-1),\\
c_k &= \frac{(\vec{p}^{\,(k)}, \vec{r}^{\,(k)})}{(\vec{p}^{\,(k)}, \mathbf{A}\vec{p}^{\,(k)})},\ q_k = \frac{(\vec{r}^{\,(k)}, \mathbf{A}\vec{p}^{\,(k-1)})}{(\vec{p}^{\,(k-1)}, \mathbf{A}\vec{p}^{\,(k-1)})}, && (k = 0, 1, \ldots, n-1).
\end{aligned}$$

As previously mentioned, for some $m \le n$ will be theoretically $\vec{r}^{\,(m)} = 0$, meaning that the exact solution of system (4.3.1) is determined.

Writing code in algorithmic language (e.g. `Fortran`), object-oriented language (e.g. `C++`), and sophisticated system (e.g. `Mathematica`) for all given methods is recommended for reader.

4.4. Packages for systems of linear algebraic equations

For many computer-early years ago (late sixties and early seventies of previous century) the most popular program (at least at Niš University), linear equations solver, was `SIMQ` from `SSP` (Scientific Subroutine Package) by IBM corporation.

Nowadays, in many cases you will have no alternative but to use sophisticated black-box program packages. Several good ones are available. `LINPACK` was developed at Argonne National Laboratories and deserves particular mention because it is published, documented, and available for free use. A successor to `LINPACK`, `LAPACK`, is becoming available. Packages available commercially include those in the `IMSL` and `NAG` libraries. One should keep in mind that the sophisticated packages are designed with very large linear systems in mind. They therefore go to great effort to minimize not only the number of operations, but also the required storage. Routines for the various tasks are usually provided in several versions, corresponding to several possible simplifications in the form of the input coefficient matrix: symmetric, triangular, banded, positive definite, etc. If one has a large matrix in one of these forms, he should certainly take advantage of the increased efficiency provided by these different routines, and not just use the form provided for general matrices. There is also a great watershed dividing routines that are direct (i.e., execute in a predictable number of operations) from routines that are iterative (i.e., attempt to converge to the desired answer in however many steps are necessary). Iterative methods become preferable when the battle against loss of significance is in danger of being lost, either due to large n or because the problem is close to singular. Very interesting techniques are those the borderline between direct and iterative methods, namely the iterative improvement of a solution that has been obtained by direct methods.

Many commercial software packages contain solvers for systems of linear algebraic equations. Some of the more prominent packages are `Matlab` and `Mathcad`. The spreadsheet `Excel` can also be used to solve systems of equations. More sophisticated packages, such as `Mathematica`, `Macsyma`, and `Maple` also contain linear equations solvers.

The book *Numerical Recipes* [4] contains several subroutines for solving systems of linear algebraic equations. Some algorithms, from which some are codded, are given in book *Numerical Methods for Engineers and Scientists* [3] (see Chapter 1).

Some general guidelines for selecting a method for solving systems of linear algebraic equations are given as follows.

- Direct elimination methods are preferred for small systems ($n \le 50$ to 100) and systems with few zeros (nonsparse systems). Gauss elimination is the method of choice.

- For tridiagonal systems, the Thomas algorithm is the method of choice ([3], Chapter 1).
- LU factorization methods are the methods of choice when more than one vectors $\vec{b}$ must be considered.
- For large systems that are not diagonally dominant, the round-off errors can be large.
- Iterative methods are preferred for large, sparse matrices that are diagonally dominant. The SOR (Successive-Over-Relaxation) method is recommended. Numerical experimentation to find the optimum over-relaxation factor ω is usually worthwhile if the system of equations is to be solved for many vectors $\vec{b}$.

Bibliography (Cited references and further reading)

[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).

[3] Hoffman, J.D., *Numerical Methods for Engineers and Scientists.* Taylor & Francis, Boca Raton-London-New York-Singapore, 2001.

[4] Press, W.H., Flannery, B.P., Teukolsky, S.A., and Vetterling, W.T., *Numerical Recepies - The Art of Scientific Computing.* Cambridge University Press, 1989.

[5] Albrecht, J., *Fehlerabschätzungen bei Relaxationsverfahren zur numerischen Auflösung linearer Gleichungsysteme.* Numer. Math. 3(1961), 188-201.

[6] Apostolatos, N. und Kulisch, U., *Über die Konvergenz des Relaxionsvrfahrens bei nicht-negativen und diagonaldominanten Matrizen.* Computing 2(1967), 17-24.

[7] Golub, G.H. & Varga, R.S., *Chebyshev semi-iterative methods, successive overrelaxation iterative methods and second order Richardson iterative methods.* Numer. Math. 3(1961), 147-168.

[8] Soutwell, R.V., *Relaxation Methods in Theoretical Physics.* 2 vols. Oxford Univ. Press. Fair Lawn., New Jersey, 1956.

[9] Varga, R.S., *Matrix Iterative Analysis.* Prentice Hall, Englewood Cliffs, New Jersey, 1962.

[10] Young, D.M., *Iterative Solution of Large Systems.* Academic Press, New York, 1971.

[11] Ralston,A., *A First Course in Numerical Analysis.* McGraw-Hill, New York, 1965.

[12] Hildebrand, F.B., *Introduction to Numerical Analysis.* Mc.Graw-Hill, New York, 1974.

[13] Acton, F.S., *Numerical Methods That Work* (corrected edition). Mathematical Association of America, Washington, D.C., 1990.

[14] Abramowitz, M., and Stegun, I.A., *Handbook of Mathematical Functions.* National Bureau of Standards, Applied Mathematics Series, Washington, 1964 (reprinted 1968 by Dover Publications, New York).

[15] Rice, J.R., *Numerical Methods, Software, and Analysis.* McGraw-Hill, New York, 1983.

[16] Forsythe, G.E., Malcolm, M.A., and Moler, C.B., *Computer Methods for Mathematical Computations.* Englewood Cliffs, Prentice-Hall, NJ, 1977.

[17] Forsythe, G.E., *Solving linear algebraic equations can be interesting.* Bull. Amer. Math. Soc. 59(1953), 299-329.

[18] Forsythe & Moler, C.B., *Computer Solution of Linear Algebraic Systems.* Prentice-Hall, Englewood Cliffs, NJ, 1967.

[19] Kahaner, D., Moler, C., and Nash, S., 1989, *Numerical Methods and Software.* Englewood Cliffs, Prentice Hall, NJ, 1989.

[20] Hamming, R.W., *Numerical Methods for Engineers and Scientists.* Dover, New York, 1962 (reprinted 1986).

[21] Ferziger, J.H., *Numerical Methods for Engineering Applications.* Stanford University, John Willey & Sons, Inc., New York, 1998

[22] Pearson, C.E., *Numerical Methods in Engineering and Science.* University of Washington, Van Nostrand Reinhold Company, New York, 1986.

[23] Stephenson, G. and Radmore, P.M., *Advanced Mathematical Methods for Engineering and Science Students.* Imperial College London, University College, London Cambridge Univ. Press, 1999

[24] Milovanović, G.V. and Kovačević, M.A., *Zbirka rešenih zadataka iz numeričke analize.* Naučna knjiga, Beograd, 1985. (Serbian).

[25] *IMSL Math/Library Users Manual* , IMSL Inc., 2500 City West Boulevard, Houston TX 77042

[26] *NAG Fortran Library*, Numerical Algorithms Group, 256 Banbury Road, Oxford OX27DE, U.K.

[27] Dongarra, J.J., et al., *LINPACK Users Guide.* S.I.A.M., Philadelphia, 1979.

[28] Golub, G.H., and Van Loan,C.F., *Matrix Computations, 2nd ed.* (Baltimore: Johns Hopkins University Press), 1989.

[29] Gill, P.E., Murray, W., and Wright, M.H., *Numerical Linear Algebra and Optimization, vol. 1.* Addison-Wesley, Redwood City, CA, 1991.

[30] Stoer, J., and Bulirsch, R., *Introduction to Numerical Analysis.* Springer-Verlag, New York, 1980.

[31] Coleman, T.F.,and Van Loan,C., *Handbook for Matrix Computations.* S.I.A.M., Philadelphia, 1988.

[32] Wilkinson, J.H., and Reinsch, C. , *Linear Algebra, vol. II of Handbook for Automatic Computation.* Springer-Verlag, New York, 1971.

[33] Westlake, J.R. *A Handbook of Numerical Matrix Inversion and Solution of Linear Equations.* Wiley, New York, 1968.

[34] Johnson, L.W., and Riess, R.D., *Numerical Analysis, 2nd ed.* Addison- Wesley, Reading, MA, 1982.

[35] Ralston, A., and Rabinowitz, P., *A First Course in Numerical Analysis, 2nd ed.* McGraw-Hill, New York, 1978.

[36] Isaacson, E., and Keller, H.B., *Analysis of Numerical Methods.* Wiley, New York, 1966.

[37] Horn,R.A., and Johnson, C.R., *Matrix Analysis.* Cambridge University Press, Cambridge, 1985.