MAP 2210 – Aplicações de Álgebra Linear

1º Semestre - 2019

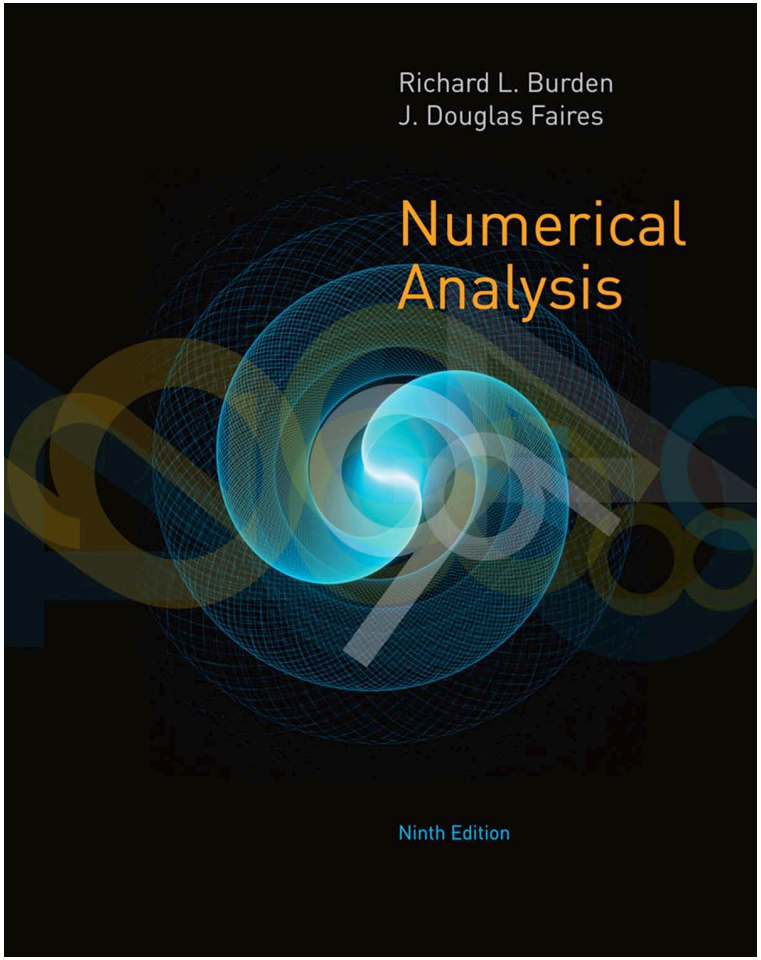
Prof. Dr. Luis Carlos de Castro Santos

lsantos@ime.usp.br

Objetivos

Formação básica de álgebra linear aplicada a problemas numéricos. Resolução de problemas em microcomputadores usando linguagens e/ou software adequados fora do horário de aula.

MAP2210



Numerical Analysis

NINTH EDITION

Richard L. Burden
Youngstown State University

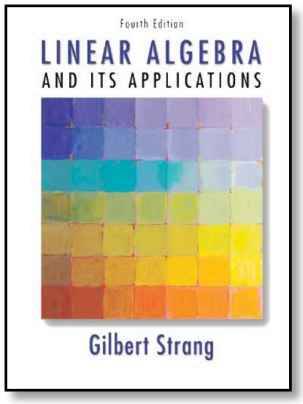
J. Douglas Faires
Youngstown State University

MAP2210

6 Direct Methods for Solving Linear Systems 357

6.1 Linear Systems of Equations 358
6.2 Pivoting Strategies 372
6.3 Linear Algebra and Matrix Inversion 381
6.4 The Determinant of a Matrix 396
6.5 Matrix Factorization 400
6.6 Special Types of Matrices 411
6.7 Survey of Methods and Software 428

+



7 Iterative Techniques in Matrix Algebra 431

7.1 Norms of Vectors and Matrices 432
7.2 Eigenvalues and Eigenvectors 443
7.3 The Jacobi and Gauss-Siedel Iterative Techniques 450
7.4 Relaxation Techniques for Solving Linear Systems 462
7.5 Error Bounds and Iterative Refinement 469
7.6 The Conjugate Gradient Method 479
7.7 Survey of Methods and Software 495

+

9 Approximating Eigenvalues 561

9.1 Linear Algebra and Eigenvalues 562
9.2 Orthogonal Matrices and Similarity Transformations 570
9.3 The Power Method 576
9.4 Householder's Method 593
9.5 The QR Algorithm 601
9.6 Singular Value Decomposition 614
9.7 Survey of Methods and Software 626

EXERCISE SET 7.3

9. The linear system

$$\begin{aligned} 2x_1 - x_2 + x_3 &= -1, \\ 2x_1 + 2x_2 + 2x_3 &= 4, \\ -x_1 - x_2 + 2x_3 &= -5 \end{aligned}$$

has the solution $(1, 2, -1)^t$.

a. Show that $\rho(T_j) = \frac{\sqrt{5}}{2} > 1$.

b. Show that the Jacobi method with $\mathbf{x}^{(0)} = \mathbf{0}$ fails to give a good approximation after 25 iterations.

c. Show that $\rho(T_g) = \frac{1}{2}$.

d. Use the Gauss-Seidel method with $\mathbf{x}^{(0)} = \mathbf{0}$ to approximate the solution to the linear system to within 10^{-5} in the l_∞ norm.


12
1
2
3
4
5
6
7
8
9
10
11

T_j = inv(D)*(L+U)

Input:

$$\begin{pmatrix} 2 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 2 \end{pmatrix}^{-1} \cdot \begin{pmatrix} 0 & 1 & -1 \\ -2 & 0 & -2 \\ 1 & 1 & 0 \end{pmatrix}$$

eigenvalues	$\begin{pmatrix} 0 & \frac{1}{2} & -\frac{1}{2} \\ -1 & 0 & -1 \\ \frac{1}{2} & \frac{1}{2} & 0 \end{pmatrix}$

Results:

$$\lambda_1 = \frac{i\sqrt{5}}{2}$$

$$\lambda_2 = -\frac{i\sqrt{5}}{2}$$

$$\lambda_3 = 0$$

T_g = inv(D – L)*U

Input:

$$\begin{pmatrix} 2 & 0 & 0 \\ 2 & 2 & 0 \\ -1 & -1 & 2 \end{pmatrix}^{-1} \cdot \begin{pmatrix} 0 & 1 & -1 \\ 0 & 0 & -2 \\ 0 & 0 & 0 \end{pmatrix}$$

eigenvalues	$\begin{pmatrix} 0 & \frac{1}{2} & -\frac{1}{2} \\ 0 & -\frac{1}{2} & -\frac{1}{2} \\ 0 & 0 & -\frac{1}{2} \end{pmatrix}$

Results:

$$\lambda_1 = -\frac{1}{2}$$

$$\lambda_2 = 0$$

Jacobi					**Gauss-Seidel**			
it	x1	x2	x3		it	x1	x2	x3
0	0,00	0,00	0,00		0	0,00	0,00	0,00
1	-0,50	2,00	-2,50		1	-0,50	2,50	-1,50
2	1,75	5,00	-1,75		2	1,50	2,00	-0,75
3	2,88	2,00	0,88		3	0,88	1,88	-1,13
4	0,06	-1,75	-0,06		4	1,00	2,13	-0,94
5	-1,34	2,00	-3,34		5	1,03	1,91	-1,03
6	2,17	6,69	-2,17		6	0,97	2,06	-0,98
7	3,93	2,00	1,93		7	1,02	1,96	-1,01
8	-0,46	-3,86	0,46		8	0,98	2,02	-1,00
9	-2,66	2,00	-4,66		9	1,01	1,99	-1,00
10	2,83	9,32	-2,83		10	0,99	2,01	-1,00
11	5,58	2,00	3,58		11	1,00	2,00	-1,00
12	-1,29	-7,16	1,29		12	1,00	2,00	-1,00
13	-4,72	2,00	-6,72		13	1,00	2,00	-1,00
14	3,86	13,44	-3,86		14	1,00	2,00	-1,00
15	8,15	2,00	6,15		15	1,00	2,00	-1,00
16	-2,58	-12,31	2,58		16	1,00	2,00	-1,00
17	-7,94	2,00	-9,94		17	1,00	2,00	-1,00
18	5,47	19,88	-5,47		18	1,00	2,00	-1,00
19	12,18	2,00	10,18		19	1,00	2,00	-1,00
20	-4,59	-20,35	4,59		20	1,00	2,00	-1,00
21	-12,97	2,00	-14,97		21	1,00	2,00	-1,00
22	7,98	29,94	-7,98		22	1,00	2,00	-1,00
23	18,46	2,00	16,46		23	1,00	2,00	-1,00
24	-7,73	-32,92	7,73		24	1,00	2,00	-1,00
25	-20,83	2,00	-22,83		25	1,00	2,00	-1,00

7.4 Relaxation Techniques for Solving Linear Systems

We saw in Section 7.3 that the rate of convergence of an iterative technique depends on the spectral radius of the matrix associated with the method. One way to select a procedure to accelerate convergence is to choose a method whose associated matrix has minimal spectral radius. Before describing a procedure for selecting such a method, we need to introduce a new means of measuring the amount by which an approximation to the solution to a linear system differs from the true solution to the system. The method makes use of the vector described in the following definition.

Definition 7.23

Suppose $\tilde{\mathbf{x}} \in \mathbb{R}^n$ is an approximation to the solution of the linear system defined by $A\mathbf{x} = \mathbf{b}$. The **residual vector** for $\tilde{\mathbf{x}}$ with respect to this system is $\mathbf{r} = \mathbf{b} - A\tilde{\mathbf{x}}$. ■

In procedures such as the Jacobi or Gauss-Seidel methods, a residual vector is associated with each calculation of an approximate component to the solution vector. The true objective is to generate a sequence of approximations that will cause the residual vectors to converge rapidly to zero. Suppose we let

$$\mathbf{r}_i^{(k)} = (r_{1i}^{(k)}, r_{2i}^{(k)}, \ldots, r_{ni}^{(k)})^t$$

denote the residual vector for the Gauss-Seidel method corresponding to the approximate solution vector $\mathbf{x}_i^{(k)}$ defined by

$$\mathbf{x}_i^{(k)} = (x_1^{(k)}, x_2^{(k)}, \ldots, x_{i-1}^{(k)}, x_i^{(k-1)}, \ldots, x_n^{(k-1)})^t.$$

The mth component of $\mathbf{r}_i^{(k)}$ is

$$r_{mi}^{(k)} = b_m - \sum_{j=1}^{i-1} a_{mj} x_j^{(k)} - \sum_{j=i}^{n} a_{mj} x_j^{(k-1)}, \tag{7.13}$$

or, equivalently,

$$r_{mi}^{(k)} = b_m - \sum_{j=1}^{i-1} a_{mj} x_j^{(k)} - \sum_{j=i+1}^{n} a_{mj} x_j^{(k-1)} - a_{mi} x_i^{(k-1)},$$

for each $m = 1, 2, \ldots, n$.

In particular, the ith component of $\mathbf{r}_i^{(k)}$ is

$$r_{ii}^{(k)} = b_i - \sum_{j=1}^{i-1} a_{ij}x_j^{(k)} - \sum_{j=i+1}^{n} a_{ij}x_j^{(k-1)} - a_{ii}x_i^{(k-1)},$$

so

$$a_{ii}x_i^{(k-1)} + r_{ii}^{(k)} = b_i - \sum_{j=1}^{i-1} a_{ij}x_j^{(k)} - \sum_{j=i+1}^{n} a_{ij}x_j^{(k-1)}. \tag{7.14}$$

Recall, however, that in the Gauss-Seidel method, $x_i^{(k)}$ is chosen to be

$$x_i^{(k)} = \frac{1}{a_{ii}}\left[b_i - \sum_{j=1}^{i-1} a_{ij}x_j^{(k)} - \sum_{j=i+1}^{n} a_{ij}x_j^{(k-1)}\right], \tag{7.15}$$

so Eq. (7.14) can be rewritten as

$$a_{ii}x_i^{(k-1)} + r_{ii}^{(k)} = a_{ii}x_i^{(k)}.$$

Consequently, the Gauss-Seidel method can be characterized as choosing $x_i^{(k)}$ to satisfy

$$x_i^{(k)} = x_i^{(k-1)} + \frac{r_{ii}^{(k)}}{a_{ii}}. \tag{7.16}$$

We can derive another connection between the residual vectors and the Gauss-Seidel technique. Consider the residual vector $\mathbf{r}_{i+1}^{(k)}$, associated with the vector $\mathbf{x}_{i+1}^{(k)} = (x_1^{(k)}, \ldots, x_i^{(k)}, x_{i+1}^{(k-1)}, \ldots, x_n^{(k-1)})^t$. By Eq. (7.13) the ith component of $\mathbf{r}_{i+1}^{(k)}$ is

$$r_{i,i+1}^{(k)} = b_i - \sum_{j=1}^{i} a_{ij} x_j^{(k)} - \sum_{j=i+1}^{n} a_{ij} x_j^{(k-1)}$$

$$= b_i - \sum_{j=1}^{i-1} a_{ij} x_j^{(k)} - \sum_{j=i+1}^{n} a_{ij} x_j^{(k-1)} - a_{ii} x_i^{(k)}.$$

By the manner in which $x_i^{(k)}$ is defined in Eq. (7.15) we see that $r_{i,i+1}^{(k)} = 0$. In a sense, then, the Gauss-Seidel technique is characterized by choosing each $x_{i+1}^{(k)}$ in such a way that the ith component of $\mathbf{r}_{i+1}^{(k)}$ is zero.

Choosing $x_{i+1}^{(k)}$ so that one coordinate of the residual vector is zero, however, is not necessarily the most efficient way to reduce the norm of the vector $\mathbf{r}_{i+1}^{(k)}$. If we modify the Gauss-Seidel procedure, as given by Eq. (7.16), to

$$x_i^{(k)} = x_i^{(k-1)} + \omega \frac{r_{ii}^{(k)}}{a_{ii}}, \tag{7.17}$$

then for certain choices of positive ω we can reduce the norm of the residual vector and obtain significantly faster convergence.

Methods involving Eq. (7.17) are called **relaxation methods**. For choices of ω with $0 < \omega < 1$, the procedures are called **under-relaxation methods**. We will be interested in choices of ω with $1 < \omega$, and these are called **over-relaxation methods**. They are used to accelerate the convergence for systems that are convergent by the Gauss-Seidel technique. The methods are abbreviated **SOR**, for **Successive Over-Relaxation**, and are particularly useful for solving the linear systems that occur in the numerical solution of certain partial-differential equations.

Before illustrating the advantages of the SOR method, we note that by using Eq. (7.14), we can reformulate Eq. (7.17) for calculation purposes as

$$x_i^{(k)} = (1 - \omega)x_i^{(k-1)} + \frac{\omega}{a_{ii}}\left[b_i - \sum_{j=1}^{i-1} a_{ij}x_j^{(k)} - \sum_{j=i+1}^{n} a_{ij}x_j^{(k-1)}\right].$$

To determine the matrix form of the SOR method, we rewrite this as

$$a_{ii}x_i^{(k)} + \omega\sum_{j=1}^{i-1} a_{ij}x_j^{(k)} = (1 - \omega)a_{ii}x_i^{(k-1)} - \omega\sum_{j=i+1}^{n} a_{ij}x_j^{(k-1)} + \omega b_i,$$

so that in vector form, we have

$$(D - \omega L)\mathbf{x}^{(k)} = [(1 - \omega)D + \omega U]\mathbf{x}^{(k-1)} + \omega\mathbf{b}.$$

That is,

$$\mathbf{x}^{(k)} = (D - \omega L)^{-1}[(1 - \omega)D + \omega U]\mathbf{x}^{(k-1)} + \omega(D - \omega L)^{-1}\mathbf{b}. \tag{7.18}$$

Letting $T_\omega = (D - \omega L)^{-1}[(1 - \omega)D + \omega U]$ and $\mathbf{c}_\omega = \omega(D - \omega L)^{-1}\mathbf{b}$, gives the SOR technique the form

$$\mathbf{x}^{(k)} = T_\omega\mathbf{x}^{(k-1)} + \mathbf{c}_\omega. \tag{7.19}$$

Example 1

The linear system $A\mathbf{x} = \mathbf{b}$ given by

$$\begin{aligned} 4x_1 + 3x_2 &= 24, \\ 3x_1 + 4x_2 - x_3 &= 30, \\ -x_2 + 4x_3 &= -24, \end{aligned}$$

has the solution $(3, 4, -5)^t$. Compare the iterations from the Gauss-Seidel method and the SOR method with $\omega = 1.25$ using $\mathbf{x}^{(0)} = (1, 1, 1)^t$ for both methods.


12
1
2
3
4
5
6
7
8
9
10
11

T_g = inv(D – L)*U

Input:

$$\begin{pmatrix} 4 & 0 & 0 \\ 3 & 4 & 0 \\ 0 & -1 & 4 \end{pmatrix}^{-1} \cdot \begin{pmatrix} 0 & -3 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{pmatrix}$$

eigenvalues	$\begin{pmatrix} 0 & \frac{3}{4} & 0 \\ 0 & \frac{9}{16} & \frac{1}{4} \\ 0 & \frac{9}{64} & \frac{1}{16} \end{pmatrix}$

Results:

$$\lambda_1 = \frac{5}{8}$$

$$\lambda_2 = 0$$

T_w = inv(D – wL)*[(1-w)D+wU]

Input:

$$\begin{pmatrix} 4 & 0 & 0 \\ 3 \times 1.25 & 4 & 0 \\ 0 & -1.25 & 4 \end{pmatrix}^{-1} \cdot \begin{pmatrix} -0.25 \times 4 & -3 \times 1.25 & 0 \\ 0 & -0.25 \times 4 & 1 \times 1.25 \\ 0 & 0 & -0.25 \times 4 \end{pmatrix}$$

eigenvalues	$\begin{pmatrix} -0.25 & -0.9375 & 0 \\ 0.234375 & 0.628906 & 0.3125 \\ 0.0732422 & 0.196533 & -0.152344 \end{pmatrix}$

Results:

$$\lambda_1 \approx -0.25$$

$$\lambda_2 \approx 0.238281 + 0.0756454\,i$$

$$\lambda_3 \approx 0.238281 - 0.0756454\,i$$

Gauss-Seidel, w = 1

Gauss-Seidel			
w	1	(1-w)	0
SOR			
it	x1	x2	x3
0	1,00	1,00	1,00
1	5,25	3,81	-5,05
2	3,14	3,88	-5,03
3	3,09	3,93	-5,02
4	3,05	3,95	-5,01
5	3,03	3,97	-5,01
6	3,02	3,98	-5,00
7	3,01	3,99	-5,00
8	3,01	3,99	-5,00
9	3,01	4,00	-5,00
10	3,00	4,00	-5,00

SOR, w = 1.25

SOR			
w	1,25	(1-w)	-0,25
SOR			
it	x1	x2	x3
0	1,00	1,00	1,00
1	6,31	3,52	-6,65
2	2,62	3,96	-4,60
3	3,13	4,01	-5,10
4	2,96	4,01	-4,97
5	3,00	4,00	-5,01
6	3,00	4,00	-5,00
7	3,00	4,00	-5,00
8	3,00	4,00	-5,00
9	3,00	4,00	-5,00
10	3,00	4,00	-5,00

Solution For each $k = 1, 2, \ldots$, the equations for the Gauss-Seidel method are

$$x_1^{(k)} = -0.75x_2^{(k-1)} + 6,$$
$$x_2^{(k)} = -0.75x_1^{(k)} + 0.25x_3^{(k-1)} + 7.5,$$
$$x_3^{(k)} = 0.25x_2^{(k)} - 6,$$

and the equations for the SOR method with $\omega = 1.25$ are

$$x_1^{(k)} = -0.25x_1^{(k-1)} - 0.9375x_2^{(k-1)} + 7.5,$$
$$x_2^{(k)} = -0.9375x_1^{(k)} - 0.25x_2^{(k-1)} + 0.3125x_3^{(k-1)} + 9.375,$$
$$x_3^{(k)} = 0.3125x_2^{(k)} - 0.25x_3^{(k-1)} - 7.5.$$

The first seven iterates for each method are listed in Tables 7.3 and 7.4. For the iterates to be accurate to seven decimal places, the Gauss-Seidel method requires 34 iterations, as opposed to 14 iterations for the SOR method with $\omega = 1.25$. ■

Gauss-Seidel

Table 7.3

k	0	1	2	3	4	5	6	7
$x_1^{(k)}$	1	5.250000	3.1406250	3.0878906	3.0549316	3.0343323	3.0214577	3.0134110
$x_2^{(k)}$	1	3.812500	3.8828125	3.9267578	3.9542236	3.9713898	3.9821186	3.9888241
$x_3^{(k)}$	1	−5.046875	−5.0292969	−5.0183105	−5.0114441	−5.0071526	−5.0044703	−5.0027940

SOR method with $\omega = 1.25$.

Table 7.4

k	0	1	2	3	4	5	6	7
$x_1^{(k)}$	1	6.312500	2.6223145	3.1333027	2.9570512	3.0037211	2.9963276	3.0000498
$x_2^{(k)}$	1	3.5195313	3.9585266	4.0102646	4.0074838	4.0029250	4.0009262	4.0002586
$x_3^{(k)}$	1	−6.6501465	−4.6004238	−5.0966863	−4.9734897	−5.0057135	−4.9982822	−5.0003486

An obvious question to ask is how the appropriate value of ω is chosen when the SOR method is used. Although no complete answer to this question is known for the general $n \times n$ linear system, the following results can be used in certain important situations.

Theorem 7.24

(Kahan)

If $a_{ii} \neq 0$, for each $i = 1, 2, \ldots, n$, then $\rho(T_\omega) \geq |\omega - 1|$. This implies that the SOR method can converge only if $0 < \omega < 2$. ■

Theorem 7.25

(Ostrowski-Reich)

If A is a positive definite matrix and $0 < \omega < 2$, then the SOR method converges for any choice of initial approximate vector $\mathbf{x}^{(0)}$. ■

Theorem 7.26

If A is positive definite and tridiagonal, then $\rho(T_g) = [\rho(T_j)]^2 < 1$, and the optimal choice of ω for the SOR method is

$$\omega = \frac{2}{1 + \sqrt{1 - [\rho(T_j)]^2}}.$$

With this choice of ω, we have $\rho(T_\omega) = \omega - 1$. ■

Example 2 Find the optimal choice of ω for the SOR method for the matrix

$$A = \begin{bmatrix} 4 & 3 & 0 \\ 3 & 4 & -1 \\ 0 & -1 & 4 \end{bmatrix}.$$

Solution This matrix is clearly tridiagonal, so we can apply the result in Theorem 7.26 if we can also who that it is positive definite. Because the matrix is symmetric, Theorem 6.24 on page 416 states that it is positive definite if and only if all its leading principle submatrices has a positive determinant. This is easily seen to be the case because

$$\det(A) = 24, \quad \det\left(\begin{bmatrix} 4 & 3 \\ 3 & 4 \end{bmatrix}\right) = 7, \quad \text{and} \quad \det([4]) = 4.$$

Because

$$T_j = D^{-1}(L+U) = \begin{bmatrix} \frac{1}{4} & 0 & 0 \\ 0 & \frac{1}{4} & 0 \\ 0 & 0 & \frac{1}{4} \end{bmatrix} \begin{bmatrix} 0 & -3 & 0 \\ -3 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix} = \begin{bmatrix} 0 & -0.75 & 0 \\ -0.75 & 0 & 0.25 \\ 0 & 0.25 & 0 \end{bmatrix},$$

we have

$$T_j - \lambda I = \begin{bmatrix} -\lambda & -0.75 & 0 \\ -0.75 & -\lambda & 0.25 \\ 0 & 0.25 & -\lambda \end{bmatrix},$$

so

$$\det(T_j - \lambda I) = -\lambda(\lambda^2 - 0.625).$$

Thus

$$\rho(T_j) = \sqrt{0.625}$$

and

$$\omega = \frac{2}{1 + \sqrt{1 - [\rho(T_j)]^2}} = \frac{2}{1 + \sqrt{1 - 0.625}} \approx 1.24.$$

This explains the rapid convergence obtained in Example 1 when using $\omega = 1.25$. ■

ALGORITHM 7.3

SOR

To solve $A\mathbf{x} = \mathbf{b}$ given the parameter ω and an initial approximation $\mathbf{x}^{(0)}$:

INPUT the number of equations and unknowns n; the entries a_{ij}, $1 \leq i, j \leq n$, of the matrix A; the entries b_i, $1 \leq i \leq n$, of $\mathbf{b}$; the entries XO_i, $1 \leq i \leq n$, of $\mathbf{XO} = \mathbf{x}^{(0)}$; the parameter ω; tolerance TOL; maximum number of iterations N.

OUTPUT the approximate solution $x_1, \ldots, x_n$ or a message that the number of iterations was exceeded.

Step 1 Set $k = 1$.

Step 2 While $(k \leq N)$ do Steps 3–6.

Step 3 For $i = 1, \ldots, n$

$$\text{set } x_i = (1 - \omega)XO_i + \frac{1}{a_{ii}}\left[\omega\left(-\sum_{j=1}^{i-1} a_{ij}x_j - \sum_{j=i+1}^{n} a_{ij}XO_j + b_i\right)\right].$$

Step 4 If $\|\mathbf{x} - \mathbf{XO}\| < TOL$ then OUTPUT $(x_1, \ldots, x_n)$;
(The procedure was successful.)
STOP.

Step 5 Set $k = k + 1$.

Step 6 For $i = 1, \ldots, n$ set $XO_i = x_i$.

Step 7 OUTPUT ('Maximum number of iterations exceeded');
(The procedure was successful.)
STOP.

■

EXERCISE SET 7.4

1. Find the first two iterations of the SOR method with $\omega = 1.1$ for the following linear systems, using $\mathbf{x}^{(0)} = \mathbf{0}$:

a.
$$\begin{aligned} 3x_1 - x_2 + x_3 &= 1, \\ 3x_1 + 6x_2 + 2x_3 &= 0, \\ 3x_1 + 3x_2 + 7x_3 &= 4. \end{aligned}$$

b.
$$\begin{aligned} 10x_1 - x_2 &= 9, \\ -x_1 + 10x_2 - 2x_3 &= 7, \\ -2x_2 + 10x_3 &= 6. \end{aligned}$$

c.
$$\begin{aligned} 10x_1 + 5x_2 &= 6, \\ 5x_1 + 10x_2 - 4x_3 &= 25, \\ -4x_2 + 8x_3 - x_4 &= -11, \\ -x_3 + 5x_4 &= -11. \end{aligned}$$

d.
$$\begin{aligned} 4x_1 + x_2 + x_3 + x_5 &= 6, \\ -x_1 - 3x_2 + x_3 + x_4 &= 6, \\ 2x_1 + x_2 + 5x_3 - x_4 - x_5 &= 6, \\ -x_1 - x_2 - x_3 + 4x_4 &= 6, \\ 2x_2 - x_3 + x_4 + 4x_5 &= 6. \end{aligned}$$

3. Repeat Exercise 1 using $\omega = 1.3$.


12
1
2
3
4
5
6
7
8
9
10
11

Ex.7.4 1.b

SOR			
w	1.1	(1-w)	-0.1
it	x1	x2	x3
0	0.00000	0.00000	0.00000
1	0.99000	0.66110	0.80544
2	0.96372	0.77508	0.74997
3	0.97889	0.74981	0.74996
4	0.97459	0.75281	0.75062
5	0.97535	0.75257	0.75050
6	0.97525	0.75258	0.75052
7	0.97526	0.75258	0.75052
8	0.97526	0.75258	0.75052
9	0.97526	0.75258	0.75052
10	0.97526	0.75258	0.75052

Ex.7.4 3.b

SOR			
w	1.3	(1-w)	-0.3
it	x1	x2	x3
0	0.00000	0.00000	0.00000
1	1.17000	0.75790	0.97705
2	0.91753	0.81739	0.69940
3	1.00100	0.71650	0.75647
4	0.96284	0.76656	0.75237
5	0.98080	0.74814	0.74881
6	0.97302	0.75375	0.75133
7	0.97608	0.75233	0.75021
8	0.97498	0.75261	0.75062
9	0.97535	0.75258	0.75049
10	0.97523	0.75257	0.75052

7.5 Error Bounds and Iterative Refinement

p/Métodos Diretos

It seems intuitively reasonable that if $\tilde{\mathbf{x}}$ is an approximation to the solution $\mathbf{x}$ of $A\mathbf{x} = \mathbf{b}$ and the residual vector $\mathbf{r} = \mathbf{b} - A\tilde{\mathbf{x}}$ has the property that $\|\mathbf{r}\|$ is small, then $\|\mathbf{x} - \tilde{\mathbf{x}}\|$ would be small as well. This is often the case, but certain systems, which occur frequently in practice, fail to have this property.

Example 1

The linear system $A\mathbf{x} = \mathbf{b}$ given by

$$\begin{bmatrix} 1 & 2 \\ 1.0001 & 2 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 3 \\ 3.0001 \end{bmatrix}$$

has the unique solution $\mathbf{x} = (1, 1)^t$. Determine the residual vector for the poor approximation $\tilde{\mathbf{x}} = (3, -0.0001)^t$.

Solution We have

$$\mathbf{r} = \mathbf{b} - A\tilde{\mathbf{x}} = \begin{bmatrix} 3 \\ 3.0001 \end{bmatrix} - \begin{bmatrix} 1 & 2 \\ 1.0001 & 2 \end{bmatrix} \begin{bmatrix} 3 \\ -0.0001 \end{bmatrix} = \begin{bmatrix} 0.0002 \\ 0 \end{bmatrix},$$

so $\|\mathbf{r}\|_\infty = 0.0002$. Although the norm of the residual vector is small, the approximation $\tilde{\mathbf{x}} = (3, -0.0001)^t$ is obviously quite poor; in fact, $\|\mathbf{x} - \tilde{\mathbf{x}}\|_\infty = 2$. ■

Theorem 7.27

Suppose that $\tilde{\mathbf{x}}$ is an approximation to the solution of $A\mathbf{x} = \mathbf{b}$, A is a nonsingular matrix, and $\mathbf{r}$ is the residual vector for $\tilde{\mathbf{x}}$. Then for any natural norm,

$$\|\mathbf{x} - \tilde{\mathbf{x}}\| \leq \|\mathbf{r}\| \cdot \|A^{-1}\|$$

and if $\mathbf{x} \neq \mathbf{0}$ and $\mathbf{b} \neq \mathbf{0}$,

$$\frac{\|\mathbf{x} - \tilde{\mathbf{x}}\|}{\|\mathbf{x}\|} \leq \|A\| \cdot \|A^{-1}\| \frac{\|\mathbf{r}\|}{\|\mathbf{b}\|}. \tag{7.20}$$

■

Proof Since $\mathbf{r} = \mathbf{b} - A\tilde{\mathbf{x}} = A\mathbf{x} - A\tilde{\mathbf{x}}$ and A is nonsingular, we have $\mathbf{x} - \tilde{\mathbf{x}} = A^{-1}\mathbf{r}$. Theorem 7.11 on page 440 implies that

$$\|\mathbf{x} - \tilde{\mathbf{x}}\| = \|A^{-1}\mathbf{r}\| \leq \|A^{-1}\| \cdot \|\mathbf{r}\|.$$

Moreover, since $\mathbf{b} = A\mathbf{x}$, we have $\|\mathbf{b}\| \leq \|A\| \cdot \|\mathbf{x}\|$. So $1/\|\mathbf{x}\| \leq \|A\|/\|\mathbf{b}\|$ and

$$\frac{\|\mathbf{x} - \tilde{\mathbf{x}}\|}{\|\mathbf{x}\|} \leq \frac{\|A\| \cdot \|A^{-1}\|}{\|\mathbf{b}\|} \|\mathbf{r}\|.$$

■ ■ ■

Condition Numbers

The inequalities in Theorem 7.27 imply that $\|A^{-1}\|$ and $\|A\| \cdot \|A^{-1}\|$ provide an indication of the connection between the residual vector and the accuracy of the approximation. In general, the relative error $\|\mathbf{x}-\tilde{\mathbf{x}}\|/\|\mathbf{x}\|$ is of most interest, and, by Inequality (7.20), this error is bounded by the product of $\|A\| \cdot \|A^{-1}\|$ with the relative residual for this approximation, $\|\mathbf{r}\|/\|\mathbf{b}\|$. Any convenient norm can be used for this approximation; the only requirement is that it be used consistently throughout.

Definition 7.28

The **condition number** of the nonsingular matrix A relative to a norm $\|\cdot\|$ is

$$K(A) = \|A\| \cdot \|A^{-1}\|.$$

■

With this notation, the inequalities in Theorem 7.27 become

$$\|\mathbf{x} - \tilde{\mathbf{x}}\| \leq K(A)\frac{\|\mathbf{r}\|}{\|A\|}$$

and

$$\frac{\|\mathbf{x} - \tilde{\mathbf{x}}\|}{\|\mathbf{x}\|} \leq K(A)\frac{\|\mathbf{r}\|}{\|\mathbf{b}\|}.$$

For any nonsingular matrix A and natural norm $\|\cdot\|$,

$$1 = \|I\| = \|A \cdot A^{-1}\| \leq \|A\| \cdot \|A^{-1}\| = K(A).$$

A matrix A is **well-conditioned** if $K(A)$ is close to 1, and is **ill-conditioned** when $K(A)$ is significantly greater than 1. Conditioning in this context refers to the relative security that a small residual vector implies a correspondingly accurate approximate solution.

Example 2

Determine the condition number for the matrix

$$A = \begin{bmatrix} 1 & 2 \\ 1.0001 & 2 \end{bmatrix}.$$

Solution We saw in Example 1 that the very poor approximation $(3, -0.0001)^t$ to the exact solution $(1, 1)^t$ had a residual vector with small norm, so we should expect the condition number of A to be large. We have $\|A\|_\infty = \max\{|1| + |2|, |1.001| + |2|\} = 3.0001$, which would not be considered large. However,

$$A^{-1} = \begin{bmatrix} -10000 & 10000 \\ 5000.5 & -5000 \end{bmatrix}, \quad \text{so} \quad \|A^{-1}\|_\infty = 20000,$$

and for the infinity norm, $K(A) = (20000)(3.0001) = 60002$. The size of the condition number for this example should certainly keep us from making hasty accuracy decisions based on the residual of an approximation. ■

Como encontrar o no. de condição sem pagar o preço da inversa ?

If we assume that the approximate solution to the linear system $A\mathbf{x} = \mathbf{b}$ is being determined using t-digit arithmetic and Gaussian elimination, it can be shown (see [FM], pp. 45–47) that the residual vector $\mathbf{r}$ for the approximation $\tilde{\mathbf{x}}$ has

$$\|\mathbf{r}\| \approx 10^{-t}\|A\| \cdot \|\tilde{\mathbf{x}}\|. \tag{7.21}$$

From this approximation, an estimate for the effective condition number in t-digit arithmetic can be obtained without the need to invert the matrix A. In actuality, this approximation assumes that all the arithmetic operations in the Gaussian elimination technique are performed using t-digit arithmetic but that the operations needed to determine the residual are done in double-precision (that is, $2t$-digit) arithmetic. This technique does not add significantly to the computational effort and eliminates much of the loss of accuracy involved with the subtraction of the nearly equal numbers that occur in the calculation of the residual.

The approximation for the t-digit condition number $K(A)$ comes from consideration of the linear system

$$A\mathbf{y} = \mathbf{r}.$$

The solution to this system can be readily approximated because the multipliers for the Gaussian elimination method have already been calculated. So A can be factored in the form P^tLU as described in Section 5 of Chapter 6. In fact $\tilde{\mathbf{y}}$, the approximate solution of $A\mathbf{y} = \mathbf{r}$, satisfies

$$\tilde{\mathbf{y}} \approx A^{-1}\mathbf{r} = A^{-1}(\mathbf{b} - A\tilde{\mathbf{x}}) = A^{-1}\mathbf{b} - A^{-1}A\tilde{\mathbf{x}} = \mathbf{x} - \tilde{\mathbf{x}}; \tag{7.22}$$

and

$$\mathbf{x} \approx \tilde{\mathbf{x}} + \tilde{\mathbf{y}}.$$

So $\tilde{\mathbf{y}}$ is an estimate of the error produced when $\tilde{\mathbf{x}}$ approximates the solution $\mathbf{x}$ to the original system. Equations (7.21) and (7.22) imply that

$$\|\tilde{\mathbf{y}}\| \approx \|\mathbf{x} - \tilde{\mathbf{x}}\| = \|A^{-1}\mathbf{r}\| \leq \|A^{-1}\| \cdot \|\mathbf{r}\| \approx \|A^{-1}\| \left(10^{-t}\|A\| \cdot \|\tilde{\mathbf{x}}\|\right) = 10^{-t}\|\tilde{\mathbf{x}}\|K(A).$$

This gives an approximation for the condition number involved with solving the system $A\mathbf{x} = \mathbf{b}$ using Gaussian elimination and the t-digit type of arithmetic just described:

$$K(A) \approx \frac{\|\tilde{\mathbf{y}}\|}{\|\tilde{\mathbf{x}}\|}10^t. \tag{7.23}$$

Illustration

The linear system given by

$$\begin{bmatrix} 3.3330 & 15920 & -10.333 \\ 2.2220 & 16.710 & 9.6120 \\ 1.5611 & 5.1791 & 1.6852 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 15913 \\ 28.544 \\ 8.4254 \end{bmatrix}$$

has the exact solution $\mathbf{x} = (1, 1, 1)^t$.

Using Gaussian elimination and five-digit rounding arithmetic leads successively to the augmented matrices

$$\begin{bmatrix} 3.3330 & 15920 & -10.333 & 15913 \\ 0 & -10596 & 16.501 & 10580 \\ 0 & -7451.4 & 6.5250 & -7444.9 \end{bmatrix}$$

and

$$\begin{bmatrix} 3.3330 & 15920 & -10.333 & 15913 \\ 0 & -10596 & 16.501 & -10580 \\ 0 & 0 & -5.0790 & -4.7000 \end{bmatrix}.$$

The approximate solution to this system is

$$\tilde{\mathbf{x}} = (1.2001, 0.99991, 0.92538)^t.$$

The residual vector corresponding to $\tilde{\mathbf{x}}$ is computed in double precision to be

$$\begin{aligned}\mathbf{r} &= \mathbf{b} - A\tilde{\mathbf{x}} \\ &= \begin{bmatrix} 15913 \\ 28.544 \\ 8.4254 \end{bmatrix} - \begin{bmatrix} 3.3330 & 15920 & -10.333 \\ 2.2220 & 16.710 & 9.6120 \\ 1.5611 & 5.1791 & 1.6852 \end{bmatrix} \begin{bmatrix} 1.2001 \\ 0.99991 \\ 0.92538 \end{bmatrix} \\ &= \begin{bmatrix} 15913 \\ 28.544 \\ 8.4254 \end{bmatrix} - \begin{bmatrix} 15913.00518 \\ 28.26987086 \\ 8.611560367 \end{bmatrix} = \begin{bmatrix} -0.00518 \\ 0.27412914 \\ -0.186160367 \end{bmatrix},\end{aligned}$$

so

$$\|\mathbf{r}\|_\infty = 0.27413.$$

The estimate for the condition number given in the preceding discussion is obtained by first solving the system $A\mathbf{y} = \mathbf{r}$ for $\tilde{\mathbf{y}}$:

$$\begin{bmatrix} 3.3330 & 15920 & -10.333 \\ 2.2220 & 16.710 & 9.6120 \\ 1.5611 & 5.1791 & 1.6852 \end{bmatrix} \begin{bmatrix} y_1 \\ y_2 \\ y_3 \end{bmatrix} = \begin{bmatrix} -0.00518 \\ 0.27413 \\ -0.18616 \end{bmatrix}.$$

This implies that $\tilde{\mathbf{y}} = (-0.20008, 8.9987 \times 10^{-5}, 0.074607)^t$. Using the estimate in Eq. (7.23) gives

$$K(A) \approx \frac{\|\tilde{\mathbf{y}}\|_\infty}{\|\tilde{\mathbf{x}}\|_\infty} 10^5 = \frac{0.20008}{1.2001} 10^5 = 16672. \tag{7.24}$$

To determine the *exact* condition number of A, we first must find A^{-1}. Using five-digit rounding arithmetic for the calculations gives the approximation:

$$A^{-1} \approx \begin{bmatrix} -1.1701 \times 10^{-4} & -1.4983 \times 10^{-1} & 8.5416 \times 10^{-1} \\ 6.2782 \times 10^{-5} & 1.2124 \times 10^{-4} & -3.0662 \times 10^{-4} \\ -8.6631 \times 10^{-5} & 1.3846 \times 10^{-1} & -1.9689 \times 10^{-1} \end{bmatrix}.$$

Theorem 7.11 on page 440 implies that $\|A^{-1}\|_\infty = 1.0041$ and $\|A\|_\infty = 15934$.

As a consequence, the ill-conditioned matrix A has

$$K(A) = (1.0041)(15934) = 15999.$$

The estimate in (7.24) is quite close to $K(A)$ and requires considerably less computational effort.

Since the actual solution $\mathbf{x} = (1, 1, 1)^t$ is known for this system, we can calculate both

$$\|\mathbf{x} - \tilde{\mathbf{x}}\|_\infty = 0.2001 \quad \text{and} \quad \frac{\|\mathbf{x} - \tilde{\mathbf{x}}\|_\infty}{\|\mathbf{x}\|_\infty} = \frac{0.2001}{1} = 0.2001.$$

The error bounds given in Theorem 7.27 for these values are

$$\|\mathbf{x} - \tilde{\mathbf{x}}\|_\infty \le K(A)\frac{\|\mathbf{r}\|_\infty}{\|A\|_\infty} = \frac{(15999)(0.27413)}{15934} = 0.27525$$

and

$$\frac{\|\mathbf{x} - \tilde{\mathbf{x}}\|_\infty}{\|\mathbf{x}\|_\infty} \le K(A)\frac{\|\mathbf{r}\|_\infty}{\|\mathbf{b}\|_\infty} = \frac{(15999)(0.27413)}{15913} = 0.27561. \qquad \square$$

Iterative Refinement

In Eq. (7.22), we used the estimate $\tilde{\mathbf{y}} \approx \mathbf{x} - \tilde{\mathbf{x}}$, where $\tilde{\mathbf{y}}$ is the approximate solution to the system $A\mathbf{y} = \mathbf{r}$. In general, $\tilde{\mathbf{x}}+\tilde{\mathbf{y}}$ is a more accurate approximation to the solution of the linear system $A\mathbf{x} = \mathbf{b}$ than the original approximation $\tilde{\mathbf{x}}$. The method using this assumption is called **iterative refinement**, or *iterative improvement*, and consists of performing iterations on the system whose right-hand side is the residual vector for successive approximations until satisfactory accuracy results.

If the process is applied using t-digit arithmetic and if $K_\infty(A) \approx 10^q$, then after k iterations of iterative refinement the solution has approximately the smaller of t and $k(t-q)$ correct digits. If the system is well-conditioned, one or two iterations will indicate that the solution is accurate. There is the possibility of significant improvement on ill-conditioned systems unless the matrix A is so ill-conditioned that $K_\infty(A) > 10^t$. In that situation, increased precision should be used for the calculations.

ALGORITHM 7.4

Iterative Refinement

To approximate the solution to the linear system $A\mathbf{x} = \mathbf{b}$:

INPUT the number of equations and unknowns n; the entries a_{ij}, $1 \le i, j \le n$ of the matrix A; the entries b_i, $1 \le i \le n$ of $\mathbf{b}$; the maximum number of iterations N; tolerance TOL; number of digits of precision t.

OUTPUT the approximation $\mathbf{xx} = (xx_1, \ldots, xx_n)^t$ or a message that the number of iterations was exceeded, and an approximation $COND$ to $K_\infty(A)$.

Step 0 Solve the system $A\mathbf{x} = \mathbf{b}$ for $x_1, \ldots, x_n$ by Gaussian elimination saving the multipliers m_{ji}, $j = i+1, i+2, \ldots, n$, $i = 1, 2, \ldots, n-1$ and noting row interchanges.

Step 1 Set $k = 1$.

Step 2 While $(k \le N)$ do Steps 3–9.

Step 3 For $i = 1, 2, \ldots, n$ (*Calculate* $\mathbf{r}$.)

$$\text{set } r_i = b_i - \sum_{j=1}^{n} a_{ij}x_j.$$

(*Perform the computations in double-precision arithmetic.*)

Step 4 Solve the linear system $A\mathbf{y} = \mathbf{r}$ by using Gaussian elimination in the same order as in Step 0.

Step 5 For $i = 1, \ldots, n$ set $xx_i = x_i + y_i$.

Step 6 If $k = 1$ then set $COND = \dfrac{\|\mathbf{y}\|_\infty}{\|\mathbf{xx}\|_\infty} 10^t$.

Step 7 If $\|\mathbf{x} - \mathbf{xx}\|_\infty < TOL$ then OUTPUT ($\mathbf{xx}$);
OUTPUT ($COND$);
(*The procedure was successful.*)
STOP.

Step 8 Set $k = k + 1$.

Step 9 For $i = 1, \ldots, n$ set $x_i = xx_i$.

Step 10 OUTPUT ('Maximum number of iterations exceeded');
OUTPUT ($COND$);
(*The procedure was unsuccessful.*)
STOP.

If t-digit arithmetic is used, a recommended stopping procedure in Step 7 is to iterate until $|y_i^{(k)}| \le 10^{-t}$, for each $i = 1, 2, \ldots, n$.

Illustration

In our earlier illustration we found the approximation to the linear system

$$\begin{bmatrix} 3.3330 & 15920 & -10.333 \\ 2.2220 & 16.710 & 9.6120 \\ 1.5611 & 5.1791 & 1.6852 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 15913 \\ 28.544 \\ 8.4254 \end{bmatrix}$$

using five-digit arithmetic and Gaussian elimination, to be

$$\tilde{\mathbf{x}}^{(1)} = (1.2001, 0.99991, 0.92538)^t$$

and the solution to $A\mathbf{y} = \mathbf{r}^{(1)}$ to be

$$\tilde{\mathbf{y}}^{(1)} = (-0.20008, 8.9987 \times 10^{-5}, 0.074607)^t.$$

By Step 5 in this algorithm,

$$\tilde{\mathbf{x}}^{(2)} = \tilde{\mathbf{x}}^{(1)} + \tilde{\mathbf{y}}^{(1)} = (1.0000, 1.0000, 0.99999)^t,$$

and the actual error in this approximation is

$$\|\mathbf{x} - \tilde{\mathbf{x}}^{(2)}\|_\infty = 1 \times 10^{-5}.$$

Using the suggested stopping technique for the algorithm, we compute $\mathbf{r}^{(2)} = \mathbf{b} - A\tilde{\mathbf{x}}^{(2)}$ and solve the system $A\mathbf{y}^{(2)} = \mathbf{r}^{(2)}$, which gives

$$\tilde{\mathbf{y}}^{(2)} = (1.5002 \times 10^{-9}, 2.0951 \times 10^{-10}, 1.0000 \times 10^{-5})^t.$$

Since $\|\tilde{\mathbf{y}}^{(2)}\|_\infty \leq 10^{-5}$, we conclude that

$$\tilde{\mathbf{x}}^{(3)} = \tilde{\mathbf{x}}^{(2)} + \tilde{\mathbf{y}}^{(2)} = (1.0000, 1.0000, 1.0000)^t$$

is sufficiently accurate, which is certainly correct. ■

Theorem 7.29

Suppose A is nonsingular and

$$\|\delta A\| < \frac{1}{\|A^{-1}\|}.$$

The solution $\tilde{\mathbf{x}}$ to $(A+\delta A)\tilde{\mathbf{x}} = \mathbf{b}+\delta\mathbf{b}$ approximates the solution $\mathbf{x}$ of $A\mathbf{x} = \mathbf{b}$ with the error estimate

$$\frac{\|\mathbf{x}-\tilde{\mathbf{x}}\|}{\|\mathbf{x}\|} \le \frac{K(A)\|A\|}{\|A\| - K(A)\|\delta A\|}\left(\frac{\|\delta\mathbf{b}\|}{\|\mathbf{b}\|} + \frac{\|\delta A\|}{\|A\|}\right). \tag{7.25}$$

■

The theorem is independent of the particular numerical procedure used to solve $A\mathbf{x} = \mathbf{b}$. It can be shown, by means of a backward error analysis (see [Wil1] or [Wil2]), that if Gaussian elimination with pivoting is used to solve $A\mathbf{x} = \mathbf{b}$ in t-digit arithmetic, the numerical solution $\tilde{\mathbf{x}}$ is the actual solution of a linear system:

$$(A+\delta A)\tilde{\mathbf{x}} = \mathbf{b}, \quad \text{where } \|\delta A\|_\infty \le f(n)10^{1-t}\max_{i,j,k}|a_{ij}^{(k)}|.$$

for some function $f(n)$. Wilkinson found that in practice $f(n) \approx n$ and, at worst, $f(n) \le 1.01(n^3+3n^2)$.

AULA 10