MATHEMATICS OF COMPUTATION
VOLUME 37, NUMBER 156
OCTOBER 1981

Perturbation Theory for Evaluation Algorithms of Arithmetic Expressions

By F. Stummel

Abstract. The paper presents the theoretical foundation of a forward error analysis of numerical algorithms under data perturbations, rounding error in arithmetic floating-point operations, and approximations in 'built-in' functions. The error analysis is based on the linearization method that has been proposed by many authors in various forms. Fundamental tools of the forward error analysis are systems of linear absolute and relative a priori and a posteriori error equations and associated condition numbers constituting optimal bounds of possible accumulated or total errors. Derivations, representations, and properties of these condition numbers are studied in detail. The condition numbers enable simple general, quantitative definitions of numerical stability, backward analysis, well- and ill-conditioning of a problem and an algorithm. The well-known illustration of algorithms and their linear error equations by graphs is extended to a method of deriving condition numbers and associated bounds. For many algorithms the associated condition numbers can be determined analytically a priori and be computed numerically a posteriori. The theoretical results of the paper have been applied to a series of concrete algorithms, including Gaussian elimination, and have proved to be very effective means of both a priori and a posteriori error analysis.

Introduction. Evaluation algorithms are defined by finite sequences $F = (F_0, \ldots, F_n)$ of input operations, evaluations of 'built-in' functions, and arithmetic operations for determining sequences $u = (u_0, \ldots, u_n)$ of input data, intermediate and final results in the form

$$u_t = F_t(u), \qquad t = 0, \ldots, n. \tag{1}$$

It is presupposed in the following that F_0 is a constant function and the function values $F_t(x)$ depend on $x_0, \ldots, x_{t-1}$ but not on $x_t, \ldots, x_n$ for $t = 1, \ldots, n$. Under perturbations, an evaluation algorithm yields approximations v_t of u_t that can be written in the form

$$v_t = (1 + e_t)F_t(v), \qquad t = 0, \ldots, n. \tag{2}$$

The so-called local errors e_t are the relative errors of the data input, function evaluation, or rounding in the arithmetic floating-point operation in step t of the algorithm. We shall assume that the local errors are bounded by $|e_t| \leqslant \gamma_t \eta$ for $t = 0, \ldots, n$, where γ_t are suitable nonnegative weights and η is an accuracy constant.

Received June 12, 1979; revised November 26, 1979 and August 27, 1980.
1980 *Mathematics Subject Classification.* Primary 65G05.
Key words and phrases. Rounding error analysis, evaluation algorithms, a priori and a posteriori error estimates, condition numbers, linear error equations, graphs.

0025-5718/81/0000-0164/$10.75

In vector notation, (1), (2) may be written $u = F(u)$, $v = F(v) + d$, using the residual vector d with the components $d_t = -v_t e'_t$ and $e'_t = -e_t/(1 + e_t)$. By introducing the mapping $A = I - F$, (1), (2) become

$$Au = 0, \qquad Av = d. \tag{3}$$

Thus, the error analysis of an evaluation algorithm is a perturbation theory of the functional equation $Au = 0$ with respect to perturbations of the right-hand side. By Taylor's formula at the point u, the absolute a priori error $\Delta u = v - u$ satisfies the relation $(A'u)\Delta u + R = d$, where R is a remainder term of order $O(\|\Delta u\|^2)$. Neglecting terms of second order in $\|\Delta u\|$ and η gives $(A'u)\Delta u \doteq J_u e$, where $J_u = \operatorname{diag}(u_0, \ldots, u_n)$. In the a posteriori error analysis, one uses Taylor's formula at the point v and obtains the relation $(A'v)\Delta v \doteq J_v e'$ for the error $\Delta v = u - v$. The solutions s of the associated systems of linear error equations

$$(A'u)s = J_u e, \quad \text{or} \quad (A'v)s = J_v e', \tag{4}$$

then yield approximations of the absolute errors Δu or Δv. Correspondingly, approximations r of the relative errors $Pu = J_u^{-1}\Delta u$ or $Pv = J_v^{-1}\Delta v$ are obtained from the systems of linear error equations

$$J_u^{-1}(A'u)J_u r = e, \quad \text{or} \quad J_v^{-1}(A'v)J_v r = e'. \tag{5}$$

The procedure, described above, is analogous to the derivation of difference approximations of a differential equation; the role of truncation errors is played by remainder terms $O(\|\Delta u\|^2)$, $O(\eta^2)$ here. Note that the linear error equations can also be derived simply from the exact error equations in Section 1.1 by neglecting terms of second order in the errors and replacing the absolute and relative errors of u, v by s and r.

The subject of the paper is a study of the structure of the mapping A and its Fréchet-derivative $A'w$, the derivation and proof of error representations of the form

$$\Delta w_t = s_t + O_t(\eta^2), \qquad Pw_t = r_t + O_t(\eta^2), \tag{6}$$

where s, r are the solutions of (4), (5), and associated optimal error estimates

$$|\Delta w_t| \leqslant \sigma_t \eta + O_t(\eta^2), \qquad |Pw_t| \leqslant \rho_t \eta + O_t(\eta^2), \tag{7}$$

a detailed analysis of the remainder terms in (6), (7) and of the properties of the optimal constants σ_t, ρ_t. In (6), (7), one chooses $w = u$ in an a priori and $w = v$ in an a posteriori error analysis. The components s_t, r_t of s, r are linear forms in the local errors $e = (e_0, \ldots, e_n)$, and the bounds $\sigma_t\eta$, $\rho_t\eta$ are associated norms, specified by the error distribution, $|e_t| \leqslant \gamma_t\eta$, of these linear forms. The constants σ_t, ρ_t are the weighted absolute and relative, a priori and a posteriori condition numbers.

Particular attention is paid in the following to the simultaneous treatment of absolute and relative, a priori and a posteriori errors as well as to readily accessible presentations of the coefficients and inhomogeneous terms of the associated error equations, of condition numbers, weights, and so on, in all four cases. For, in applications to concrete examples it is seen that for some algorithms the systems of linear absolute error equations, for others the systems of linear relative error equations, are easier to handle. The general theoretical rounding error analysis of

numerical algorithms uses a priori error representations. In many examples, condition numbers can be computed numerically a posteriori, that is, together with the intermediate and final results of the algorithm.

By a suitable choice of the weights in the local error distribution, the behavior of the algorithm can be studied under data perturbations only, assuming exact 'built-in' functions and arithmetic operations, as well as under rounding errors in the arithmetic operations and 'built-in' functions only, assuming exact data. The associated absolute and relative condition numbers are denoted by σ_t^D, σ_t^R and ρ_t^D, ρ_t^R. The data condition numbers σ_t^D are absolute asymptotic conditions, in the sense of Rice [15], of u_t viewed as a function of the data and thus independent of the algorithm (see Section 2.1). By Wilkinson [25, I.36], a problem is said to be ill-conditioned if small relative errors of the input data can induce great relative errors of the solution u_t of the algorithm. This effect can be formulated quantitatively in our terms by $\rho_t^D \gg 1$. Wilkinson's backward analysis describes the behavior of the algorithm under rounding errors in the arithmetic operations by data perturbations. Our results in Section 2.1 show that $\sigma_t^R/\sigma_t^D = \rho_t^R/\rho_t^D$ is a measure for the magnitude of those data perturbations which are necessary and sufficient in order to represent perturbations by rounding errors in the arithmetic operations. Bauer's concept (see [4], [5]) of a well-conditioned algorithm for the computation of u_t can be specified by the requirement that ρ_t^R/ρ_t^D is not much greater than one.

Also, a well-conditioned algorithm can produce numerical results such that not even the first digit is significant. This happens when the problem is ill-conditioned and the accuracy η is too low. The least number of significant digits in a computed result can be determined by our relative condition numbers ρ_t and the associated notion of numerical stability. Let data perturbations and rounding errors, within a given distribution of local errors, be bounded by the accuracy η_0. Then the result v_t is computed under all these perturbations within the relative error bound η_1 if the stability inequality $\rho_t < \eta_1/\eta_0$ holds. This result follows immediately from (7) for $\eta = \eta_0$. For example, choosing $\eta_1 = \frac{1}{2}$ and $\eta_0 < 1/(2\rho_t)$ guarantees that at least the first digit in the computed result v_t is accurate in the sense of relative errors. It is always assumed that remainder terms of the form $O(\eta^2)$ are negligible against the terms of first order in η.

Evaluation algorithms and their systems of linear error equations may be illustrated by graphs constituting detailed complete flow diagrams of the functional dependences between the data, intermediate and final results, as well as exhibiting the paths of error propagation and the associated error effects along these paths. It seems that this tool is found for the first time in the book of McCracken-Dorn [11] who attribute the idea to H. M. King. In essentially the same form Bauer [5], Brown [7], Linnainmaa [10], and others use graphs in their error analyses. The matrix elements of the solution operators $(A'w)^{-1}$, $J_w^{-1}(A'w)^{-1}J_w$ can be read from the weighted graph. In addition, Section 2.3 provides graph theoretic means for determining condition numbers and associated bounds. For instance, when the graph of the algorithm is a tree, condition numbers are obtained recurrently by simple formulae having the same form as those of the condition numbers of the simplest algorithms in Section 1.1.

Note that the concepts of condition numbers used, for example, by Wilkinson [25], Bauer [5], or the well-known condition numbers of matrices, are defined differently and have other meanings. In the rounding error analysis of a fixed-point arithmetic, Henrici [8, 16.4] states a system of linear equations for the exact errors Δu_t. Also the optimality of error bounds, obtained in this way, is observed there. Linnainmaa [10] studies Taylor expansions of the total errors Δu_t with respect to the local errors $e_0, \ldots, e_n$. In particular, two algorithms for computing the coefficient matrices of the terms of first and second order in the expansions of the total errors are presented. We can interpret the first algorithm as a procedure to compute the rows of the inverse matrix $(A'u)^{-1}$ from the matrix $F'(u)$ (see Section 1.3). This algorithm has been described by Larson-Sameh [9] in a somewhat different context. Bauer [5], Brown [7], and Larson-Sameh [9] use relative or logarithmic derivatives for obtaining representations of the total relative errors Pu_t. This method yields first-order approximations of the errors that coincide with the solutions r_t of the system of linear relative a priori error equations (5).

Another way of deriving a priori error representations and estimates of the kind described above starts from an analysis of the perturbed results v_t viewed as functions of the local errors $e = (e_0, \ldots, e_n)$. Let $v_t = G_t(e)$, then $u_t = G_t(0)$ and, by Taylor expansion,

$$\Delta u_t = G_t'(0)e + O_t(\eta^2). \tag{8}$$

From (4), (6) with $w = u$, and (8) it follows that $s_t = G_t'(0)e$ for all local errors e and

$$G_t'(0) = \mathrm{pr}_t(A'u)^{-1}J_u, \tag{9}$$

where pr_t denotes the orthogonal projection onto the tth coordinate axis. The absolute a priori condition numbers thus permit the representation

$$\sigma_t = \lim_{\eta \to +0} \frac{1}{\eta} \sup_{|e| < \gamma\eta} |G_t(e) - G_t(0)| = \sum_{k=0}^{t} \left| \frac{\partial G_t}{\partial e_k}(0) \right| \gamma_k. \tag{10}$$

Babuška's stability constants Λ in [1], [2] are defined in the form of the first equation in (10). Miller's papers [12], [13] use condition numbers of the kind defined by the second equation in (10). The associated stability constants $1 + \sigma_t^R/\sigma_t^D$ are denoted by ξ in [1], [2]. Miller [12], [13] and Miller-Spooner [14] use the notation ρ and ω_1 for σ_t^R/σ_t^D. When the error analysis is applied to concrete algorithms, the functional dependence G_t of v_t on the local errors is, in general, very complex, so that this approach is limited to small algorithms.

Although the total number of operations, and thus the order of the associated matrices $A'w$, becomes very large for many important algorithms, the matrices $A'w$ are very sparse and highly structured. The directed graph of the functional dependences of an algorithm and the directed graph of an associated linear system of error equations are identical. Thus the structure of the system of linear error equations is, necessarily, related to that of the algorithm. This fact might be the deeper reason why for many important algorithms, including Gaussian elimination of systems of linear algebraic equations, explicit analytical representations of the solutions of the systems of linear error equations and of the associated condition numbers can be determined.

In a series of papers, surveyed in Section 3, the present error analysis has been applied to concrete numerical algorithms. The error estimates have been tested by numerous numerical examples. All these examples have confirmed that the condition numbers clearly and concisely yield crucial information about the numerical behavior of the algorithms. The a posteriori condition numbers have proved to be reliable measures of the magnitude of possible errors.

1. Elements of Error Propagation. The first chapter introduces basic concepts of the perturbation theory for numerical algorithms. Starting points are representations and estimates of the errors in elementary arithmetic operations, $+$, $-$, $\times$, $/$, and 'built-in' functions occurring in the floating-point arithmetic of computers. In particular, condition numbers are defined for the simplest algorithms, consisting of the input of one or two operands followed by an arithmetic operation or function evaluation. It will be seen in Section 2.3 that these condition numbers are also used in determining condition numbers or associated bounds for general algorithms. In Section 1.2 the class of algorithms for evaluating arithmetic expressions is defined and the general form of perturbed algorithms specified in view of typical rounding error analyses. Further, the general error equations for the solutions of the perturbed algorithms are derived, and associated estimates for the remainder terms of Taylor's formula are established. By neglecting remainder terms in the general error equations, linear error equations arise whose solutions approximate the absolute and relative a priori and a posteriori errors being of interest. The algorithm (A) determines uniquely a mapping A in $\mathbf{R}^{n+1}$. It will be shown in Section 1.3 that the linear error equations are defined by the Fréchet-derivative A' of A. The solutions of the linear absolute error equations are obtained by means of the solution operators $L = (A'w)^{-1}$, and of the relative error equations by $L = J_w^{-1}(A'w)^{-1}J_w$. This is in correspondence with the use of so-called relative or logarithmic derivatives (see Bauer [5], Brown [7]).

1.1. *Error Estimates for the Simplest Numerical Algorithms.* Given two numbers a, b and arbitrary approximations a', b' of a, b, the following *absolute* and *relative a priori* and *a posteriori* errors are defined:

(1)

	absolute	relative
a priori	$\Delta a = a' - a$	$Pa = \dfrac{a' - a}{a}$
a posteriori	$\Delta a' = a - a'$	$Pa' = \dfrac{a - a'}{a'}$

In the relative error analysis it will always be assumed that denominators are different from zero. Let us first state representations of the absolute and relative a priori errors

$$(2)\quad \Delta(a \circ b) = a' \circ b' - a \circ b, \quad P(a \circ b) = \frac{a' \circ b' - a \circ b}{a \circ b}, \quad \circ = +, -, \times, /,$$

of sums, differences, products, and quotients under perturbations of the operands a, b. It is well known that

$$
\begin{aligned}
&\Delta(a \pm b) = \Delta a \pm \Delta b, && P(a \pm b) = \frac{a}{c} Pa \pm \frac{b}{c} Pb,\\
&\Delta(ab) = b\Delta a + a\Delta b + \Delta a \Delta b, && P(ab) = Pa + Pb + PaPb,\\
&\Delta(a/b) = \frac{1}{b + \Delta b}\left(\Delta a - \frac{a}{b}\Delta b\right), && P(a/b) = \frac{Pa - Pb}{1 + Pb},
\end{aligned} \tag{3}
$$

where $c = a \pm b$. The numerical computation of $a \circ b$ first requires input or computing of the operands a, b carried out, in general, approximately. The arithmetic floating-point operations are then applied to approximations of a, b. Input of a, b gives, for instance, $a' = \mathrm{fl}(a)$, $b' = \mathrm{fl}(b)$. By fl is meant a function mapping the real numbers into N-digit floating-point numbers. Let g denote the base of the number representation. The floating-point arithmetic thus computes the approximations

$$
v = \mathrm{fl}(a' \circ b') = (1 + e)(a' \circ b'), \qquad |e| \leqslant \eta, \tag{4}
$$

of the exact results $u = a \circ b$ where $\circ = +, -, \times, /$ (see Wilkinson [25]). The floating-point *accuracy constant* η is, for example, $\frac{1}{2}g^{-N+1}$ when fl is symmetric rounding, and g^{-N+1} when fl denotes chopping off the mantissae to N digits. Now

$$
\Delta u = v - u = (1 + e)(a' \circ b' - a \circ b) + e(a \circ b),
$$

so that the absolute and relative a priori errors of v have the representations

$$
\Delta u = (1 + e)\Delta(a \circ b) + ue, \qquad Pu = (1 + e)P(a \circ b) + e. \tag{5}
$$

Inserting the above expressions for $\Delta(a \circ b)$, $P(a \circ b)$ in (3) yields the *absolute* and *relative a priori error equations*.

Analogous representations hold for the absolute and relative a posteriori errors

$$
\begin{aligned}
&\Delta(a' \circ b') = a \circ b - a' \circ b',\\
&P(a' \circ b') = \frac{a \circ b - a' \circ b'}{a' \circ b'}, \qquad \circ = +, -, \times, /.
\end{aligned} \tag{6}
$$

They are obtained by interchanging a, b and a', b' in (3). Let us further introduce the notation

$$
e' = -\frac{e}{1 + e} \qquad (e \neq -1). \tag{7}
$$

Then, evidently,

$$
e + e' + ee' = 0, \qquad (1 + e)(1 + e') = 1. \tag{8}
$$

Using (4), one has

$$
\Delta v = u - v = a \circ b - a' \circ b' - e(a' \circ b'), \qquad a' \circ b' = \frac{v}{1 + e}.
$$

The absolute and relative a posteriori errors of the approximation v of u can therefore be written in the form

$$
\Delta v = \Delta(a' \circ b') + ve', \qquad Pv = (1 + e')P(a' \circ b') + e'. \tag{9}
$$

From (3), (9) we obtain *absolute* and *relative a posteriori error equations*. Note that the representation of relative a priori errors turns into that of relative a posteriori errors when a, b, e, u are interchanged by a', b', e', v.

Next let a real function f of a single variable be given, twice continuously differentiable in its open domain of definition. Under perturbations of the argument a, the induced approximations $\mathrm{f}(a')$ of $f(a)$ have the absolute and relative a priori errors

$$\begin{aligned} \Delta f(a) &= \left(1 + \frac{f[a, a, a']}{f'(a)}\Delta a\right) f'(a)\Delta a, \\ Pf(a) &= \left(1 + \frac{af[a, a, a']}{f'(a)} Pa\right) \frac{af'(a)}{f(a)} Pa, \end{aligned} \tag{10}$$

using the Hermite generalized divided differences

$$f[a, a, a'] = \int_0^1 \left(\int_0^t f''(a + s(a' - a))\, ds \right) dt. \tag{11}$$

It is presupposed that $f'(a)$ and, in the case of relative errors, a and $f(a)$ are different from zero. The above representations are valid for all pairs a, a' such that the line segment $\overline{aa'}$ belongs to the domain of definition of f.

In the numerical evaluation of real functions, instead of $u = f(a)$ an approximation of f at a neighboring point a' of a is computed. Let v denote this approximation. Then

$$v = (1 + e)f(a'), \qquad e = \frac{v - f(a')}{f(a')} \qquad (f(a') \neq 0). \tag{12}$$

It can be assumed, for example, that elementary 'built-in' functions are computed by

$$v = \mathrm{fl}(f(a')). \tag{13}$$

In this case,

$$e = \frac{\mathrm{fl}(y) - y}{y}, \quad y = f(a'), \quad |e| \leqslant \eta. \tag{14}$$

In line with (5), (9), one finds the error equations

$$\begin{aligned} \Delta u &= (1 + e)\Delta f(a) + ue, \quad & Pu &= (1 + e)Pf(a) + e, \\ \Delta v &= \Delta f(a') + ve', & Pv &= (1 + e')Pf(a') + e', \end{aligned} \tag{15}$$

for the approximations v of $u = f(a)$, where e is specified by (12) and herewith e' by (7). Inserting the representations (10) of $\Delta f(a)$ and $Pf(a)$ into (15) leads to the *absolute* and *relative a priori error equations* of numerical evaluations of f. Interchanging a, a' in (10) and inserting the resulting representations of $\Delta f(a')$, $Pf(a')$ into (15) yields the associated *absolute* and *relative a posteriori error equations*. Again, the relative a priori error equations change to the a posteriori error equations, and vice versa, when a, e, u are interchanged by a', e', u'.

The simplest numerical algorithms consist of the input of a or a, b followed by the computation of $f(a)$ or $a \circ b$ for $\circ = +, -, \times, /$. From the above error equations we can readily deduce associated error estimates. For this purpose, let us assume that the relative errors Pa, Pb, e_+, e_-, ... of the input data and arithmetic floating-point operations $+, -, \ldots$ are bounded by

$$|Pa| \leqslant \rho_a\eta, \quad |Pb| \leqslant \rho_b\eta, \quad |e| \leqslant \gamma\eta. \tag{16}$$

The absolute errors Δa, Δb then satisfy the estimates

$$|\Delta a| \leqslant \sigma_a \eta, \quad |\Delta b| \leqslant \sigma_b \eta, \quad \sigma_a = |a|\rho_a, \quad \sigma_b = |b|\rho_b. \tag{17}$$

The constants ρ_a, ρ_b and σ_a, σ_b are the relative and absolute a priori condition numbers of the data a, b. They permit adjusting the bounds to the magnitude of possible errors of the data approximations a', b'. When a, b are results of a previous numerical computation, in general a', b' have larger errors than input errors of magnitude η. Constants ρ_a, ρ_b, σ_a, σ_b may be zero when input data are exact or unperturbed. The *weight* γ can be set equal to zero if the floating-point operation is performed exactly, for instance, when a loss of significant figures occurs in subtractions. The error equations (3), (5), (10), (15) entail the *absolute* and *relative a priori error estimates*

$$|\Delta u| \leqslant (1 + O(\eta))\sigma\eta, \qquad |Pu| \leqslant (1 + O(\eta))\rho\eta, \tag{18}$$

using the following *absolute* and *relative a priori condition numbers* of arithmetic operations and function evaluations:

$$\begin{aligned}
+,-: &\quad \sigma = \sigma_a + \sigma_b + |u|\gamma, && \rho = \left|\frac{a}{u}\right|\rho_a + \left|\frac{b}{u}\right|\rho_b + \gamma,\\
\times: &\quad \sigma = |b|\sigma_a + |a|\sigma_b + |u|\gamma, && \rho = \rho_a + \rho_b + \gamma,\\
/: &\quad \sigma = \frac{1}{|b|}\sigma_a + \left|\frac{u}{b}\right|\sigma_b + |u|\gamma, && \rho = \rho_a + \rho_b + \gamma,\\
f: &\quad \sigma = |f'(a)|\sigma_a + |u|\gamma, && \rho = \left|\frac{af'(a)}{f(a)}\right|\rho_a + \gamma.
\end{aligned} \tag{19}$$

Note that always $\rho = \sigma/|u|$, $u = a \circ b$ for $\circ = +, -, \times, /$ or $u = f(a)$. The above estimates require for divisions that $\rho_b\eta \leqslant \vartheta < 1$, and for function evaluations that $|f[a, a, a']|/|f'(a)| \leqslant \mu$.

Let us assume that the a posteriori errors Pa', Pb', e' and $\Delta a'$, $\Delta b'$ satisfy estimates corresponding to (16), (17). From (3), (6), (9), (10), (15) the *absolute* and *relative a posteriori error estimates*

$$|\Delta v| \leqslant (1 + O(\eta))\sigma\eta, \qquad |Pv| \leqslant (1 + O(\eta))\rho\eta \tag{20}$$

follow, where σ, ρ denote the *absolute* and *relative a posteriori condition numbers*

$$\begin{aligned}
+,-: &\quad \sigma = \sigma_a + \sigma_b + |v|\gamma, && \rho = \left|\frac{a'}{v}\right|\rho_a + \left|\frac{b'}{v}\right|\rho_b + \gamma,\\
\times: &\quad \sigma = |b'|\sigma_a + |a'|\sigma_b + |v|\gamma, && \rho = \rho_a + \rho_b + \gamma,\\
/: &\quad \sigma = \frac{1}{|b'|}\sigma_a + \left|\frac{v}{b'}\right|\sigma_b + |v|\gamma, && \rho = \rho_a + \rho_b + \gamma,\\
f: &\quad \sigma = |f'(a')|\sigma_a + |v|\gamma, && \rho = \left|\frac{a'f'(a')}{f(a')}\right|\rho_a + \gamma.
\end{aligned} \tag{21}$$

Now $\rho = (1 + O(\eta))\sigma/|v|$, $v = \mathrm{fl}(a' \circ b')$ or $v = \mathrm{fl}(f(a'))$. It is always assumed in the present paper that η and $O(\eta)$ are small compared to 1. The quantities a', b', v are floating-point numbers, so that a posteriori condition numbers can be computed numerically together with the result v.

The above estimates (18) are *sharp* or *optimal* for the class of all perturbations defined by (16). When ρ_a, ρ_b, γ and the data a, b are so chosen that

$$Pa = \operatorname{sgn}\left(\frac{a}{u}\right)\rho_a\eta, \quad Pb = \operatorname{sgn}\left(\frac{b}{u}\right)\rho_b\eta, \quad e = \gamma\eta, \tag{22}$$

the relative a priori error of the sum $u = a + b$ becomes

$$Pu = (1 + \gamma\eta)\left(\left|\frac{a}{u}\right|\rho_a + \left|\frac{b}{u}\right|\rho_b\right)\eta + \gamma\eta, \tag{23}$$

that is, $\rho\eta \leqslant Pu \leqslant (1 + \gamma\eta)\rho\eta$. Analogous statements are true for the other arithmetic operations and function evaluations, as well as for a posteriori error estimates.

1.2. *Evaluation Algorithms and General Representations of Errors.* The evaluation of arithmetic expressions by a computer is carried out in a series of elementary steps: input of numbers, arithmetic operations $+,-,\times,/$ and evaluations of elementary 'built-in' functions $\sqrt{}$, ln, exp, sin, cos, ... In this section a class of finite algorithms for evaluating general arithmetic expressions is defined analogously to typical computer programs. As a rule, data have to be read into the storage first. From these data a sequence of intermediate and final results is computed stepwise. In doing so, further data may be read in. We assume that data, intermediate, and final results are stored in places having the addresses $0, 1, 2, \ldots, n$. In each step of the computation all previously computed results are available for further computations. An evaluation algorithm is thus defined by a constant function F_0, representing the input of a number, and a finite sequence of real functions F_t, having suitable domains of definition def F_t in $\mathbf{R}^t$, in the form

$$u_0 = F_0, \quad u_t = F_t(u_0, \ldots, u_{t-1}), \qquad t = 1, \ldots, n. \tag{A}$$

The functions F_t specify how the next value is computed from the data and intermediate results $u_0, \ldots, u_{t-1}$ in storage places $0, \ldots, t-1$. In view of the further study it is expedient to use vector notation such that

$$u = (u_0, \ldots, u_n), \quad v = (v_0, \ldots, v_n), \quad x = (x_0, \ldots, x_n), \ldots \in \mathbf{R}^{n+1}.$$

The functions F_t are then defined by

$$F_0(x) = F_0, \quad F_t(x) = F_t(x_0, \ldots, x_{t-1}), \qquad t = 1, \ldots, n, \tag{1}$$

for all $x \in \mathbf{R}^{n+1}$ with the property that $(x_0, \ldots, x_{t-1})$ belongs to def F_t.

The class (F) of admissible functions F_t is composed of three subclasses (F0), (F1), (F2) by

$$F_t \in (\mathrm{F}) = (\mathrm{F0}) \cup (\mathrm{F1}) \cup (\mathrm{F2}), \qquad t = 0, \ldots, n.$$

(F0) *Input operations* are represented by constant functions F_t on all of $\mathbf{R}^{n+1}$. In the floating-point arithmetic of a computer input operations are carried out in the form

$$\mathrm{fl}(F_t(x)) = (1 + e_t)F_t(x).$$

The relative errors e_t of these approximations are bounded, for instance, by $|e_t| \leqslant \eta$, where η denotes the floating-point accuracy constant defined in Section 1.1.

(F1) '*Built-in*' *functions*. Functions in this class are defined by

$$F_t(x) = f_t(x_{j_t}),$$

using indices $j_t \in \{0, \ldots, t-1\}$ and elementary, 'built-in', functions

$$f_t \in \{\sqrt{}, \ln, \exp, \sin, \cos, \ldots\}.$$

The domain of definition of F_t consists of all those x in $\mathbf{R}^{n+1}$ such that x_{j_t} belongs to def f_t. In floating-point arithmetic these functions are, at best, evaluated by

$$\mathrm{fl}(F_t(x)) = (1 + e_t)F_t(x), \qquad |e_t| \leqslant \eta.$$

In this class also functions f_t of the form $f_t(z) = z \circ p_t$ are admitted, where $\circ = +, -, \times, /$ or $\uparrow$ and p_t is a constant which can be represented exactly as a floating-point number. A simple example is $f_t(z) = z\uparrow 2$.

(F2) *Arithmetic operations* are represented by functions F_t of the form

$$F_t(x) = (\pm x_{i_t}) \circ (\pm x_{j_t}),$$

where i_t, j_t are distinct indices in $\{0, \ldots, t-1\}$, and $\circ$ are operations in $\{+, -, \times, /\}$. The domain of definition of $F_t = +, -, \times$ is all of $\mathbf{R}^{n+1}$. In the case of divisions, def $F_t = \{x \in \mathbf{R}^{n+1} \mid x_{j_t} \neq 0\}$. Let us assume that the floating-point arithmetic of our computer evaluates the arithmetic operations approximately by

$$\mathrm{fl}(F_t(x)) = (1 + e_t)F_t(x), \qquad |e_t| \leqslant \eta.$$

Note that $-x_{i_t}$ is obtained exactly from x_{i_t} by a change of sign.

The evaluation of functions $F_t \in (\mathrm{F})$ in the floating-point arithmetic of a computer is carried out approximately as described above. Instead of the sequence of solutions $u_0, \ldots, u_n$ of (A), a sequence of approximations $v_0, \ldots, v_n$ is computed such that v_t is an approximation of $y_t = F_t(v)$ for each t. Let e_t denote the relative a priori error of the approximation v_t of y_t and let e_t' be the associated relative a posteriori error,

$$(2) \qquad e_t = \frac{v_t - y_t}{y_t} = Py_t, \quad e_t' = -\frac{e_t}{1 + e_t} = \frac{y_t - v_t}{v_t}, \qquad t = 0, \ldots, n.$$

The sequence $v_0, \ldots, v_n$ thus satisfies the recursion

$$(\tilde{\mathrm{A}}) \qquad v_0 = (1 + e_0)F_0, \quad v_t = (1 + e_t)F_t(v_0, \ldots, v_{t-1}), \qquad t = 1, \ldots, n.$$

In typical applications the errors e_t, e_t' are bounded in modulus by the floating-point accuracy constant η or some multiple of η. We shall call the e_t *local (rounding) errors*.

For the sake of notational simplicity, let us introduce the notation J_w for the diagonal matrix $\mathrm{diag}(w_0, \ldots, w_n)$ and any vector $w = (w_0, \ldots, w_n)$. In this sense,

$$(3) \qquad J_w^{-1} = \mathrm{diag}\left(\frac{1}{w_0}, \ldots, \frac{1}{w_n}\right),$$

provided that $w_t \neq 0$, $t = 0, \ldots, n$. Using this notation, the absolute and relative a priori and a posteriori errors of approximations $v = (v_0, \ldots, v_n)$ of $u = (u_0, \ldots, u_n)$, where $u_t \neq 0$, $v_t \neq 0$ for all t, satisfy the relations

$$(4) \qquad \begin{aligned} &\Delta u = v - u, \quad Pu = J_u^{-1}\Delta u, \quad \Delta u = J_u Pu, \\ &\Delta v = u - v, \quad Pv = J_v^{-1}\Delta v, \quad \Delta v = J_v Pv. \end{aligned}$$

Now the fundamental error equations for the solutions of the perturbed algorithm shall be established. The functions $F_t \in (\mathrm{F})$ are Fréchet-differentiable at each interior point of their domains of definition and satisfy the *Taylor formula*

$$(5) \qquad F_t(x + h) = F_t(x) + F_t'(x)h + R_t(x, h).$$

The *gradients* $F_t'(x)$ of F_t have the form

$$F_t'(x) = \left(\frac{\partial F_t}{\partial x_0}(x), \ldots, \frac{\partial F_t}{\partial x_{t-1}}(x), 0, \ldots, 0 \right), \tag{6}$$

because F_t depends on $x_0, \ldots, x_{t-1}$ only. Input operations $F_t \in$ (F0) are constant functions so that

$$F_t'(x) = 0, \qquad R_t(x; h) = 0. \tag{7}$$

For elementary functions $F_t(x) = f(x_j)$ one obtains, from 1.1.(10) with $a = x_j$, $a' = x_j' = x_j + h_j$, the representation

$$F_t'(x)h = f'(x_j)h_j, \qquad R_t(x; h) = f[x_j, x_j, x_j']h_j^2, \tag{8}$$

where it is assumed that the line segment $\overline{x_j x_j'}$ belongs to def f. Additions and subtractions $F_t(x) = x_i \pm x_j$ give

$$F_t'(x)h = h_i \pm h_j, \qquad R_t(x; h) = 0. \tag{9}$$

For multiplications $F_t(x) = x_i x_j$ it is seen from 1.1(3) that

$$F_t'(x)h = x_j h_i + x_i h_j, \qquad R_t(x; h) = h_i h_j. \tag{10}$$

In the case of divisions one finally has

$$F_t'(x)h = \frac{1}{x_j} h_i - \frac{x_i}{x_j^2} h_j, \qquad R_t(x; h) = \frac{h_j}{x_j'} F_t'(x)h, \tag{11}$$

provided that $x_j \neq 0$, $x_j' \neq 0$.

The next theorem constitutes the basis for subsequent error estimates. It shows that the remainder term of the Taylor formula can be estimated locally uniformly in suitable neighborhoods of u.

(12) *Let the solution u of the algorithm* (A) *be an interior point of* def F_t *and* $u_t \neq 0$ *for all t. Then there exist positive constants κ_t, ξ_t such that uniformly for all x, h in*

(i) $$\left| \frac{x_t - u_t}{u_t} \right| \leqslant \xi_t, \quad \left| \frac{h_t}{u_t} \right| \leqslant \xi_t, \qquad t = 0, \ldots, n,$$

the joins $\overline{x, x + h}$ *belong to* def F_t', *the components x_t do not vanish, and the remainder terms satisfy the estimates*

(ii) $$\left| \frac{1}{F_t(x)} R_t(x; h) \right| \leqslant \kappa_t \Big(\max_{j<t} |Px_j| \Big)^2, \qquad t = 0, \ldots, n,$$

where $Px_j = h_j / x_j$.

Proof. (i) As u is an interior point of def F_t for all t, there are positive constants $\xi_t^0 \leqslant 2/3$ such that all x in the neighborhood $|x_t - u_t|/|u_t| \leqslant \xi_t^0$, $t = 0, \ldots, n$, of u belong to def F_t' for all t. Consider first those indices t such that $F_t \in$ (F1), that is, $F_t(x) = f_t(x_{j_t})$. Then u_{j_t} is an interior point of def f_t and there are positive constants $\xi_{j_t}^1 \leqslant \xi_{j_t}^0$ such that all x_{j_t} satisfying $|x_{j_t} - u_{j_t}|/|u_{j_t}| \leqslant \xi_{j_t}^1$ belong to def f_t, where f_t is twice continuously differentiable. Since $f_t(u_{j_t}) = u_t \neq 0$, $\xi_{j_t}^1$ may be chosen so small that also $f_t(x_{j_t}) \neq 0$ for x_{j_t} in this neighborhood. For all indices $k \neq j_t$ and $F_t \in$ (F1) put $\xi_k^1 = \xi_k^0$. Finally let $\xi_t = \frac{1}{2}\xi_t^1$ for $t = 0, \ldots, n$.

(ii) By virtue of the condition $\xi_t \leqslant \frac{1}{2}\xi_t^0$, $t = 0, \ldots, n$, the line segments $\overline{x, x + h}$ belong to def F_t' for all $t = 0, \ldots, n$, whenever x, h is in the neighborhood (12i) of

u, 0. Since $\xi_t < 1$, $x_t \neq 0$ for all t. For input operations, additions and subtractions, (12ii) holds trivially with $\kappa_t = 0$. In view of (10), the remainder term estimate then holds for multiplications with $\kappa_t = 1$. In the case of divisions one has

$$\frac{1}{F_t(x)} R_t(x;h) = -\frac{1}{1 + Px_j}(Px_i - Px_j)Px_j, \qquad i = i_t, j = j_t.$$

Now $\xi_k \leqslant \frac{1}{2}\xi_k^0 \leqslant \frac{1}{3}$ for all k, and consequently

$$\frac{2}{3} \leqslant 1 - \left|\frac{x_k - u_k}{u_k}\right| \leqslant \left|\frac{x_k}{u_k}\right|, \qquad |Px_k| = \left|\frac{h_k}{u_k}\right|\left|\frac{u_k}{x_k}\right| \leqslant \frac{1}{2}.$$

Hence the estimate (12ii) is true for $\kappa_t = 4$. Finally, consider those t such that $F_t \in$ (F1), that is, $F_t(x) = f_t(x_{j_t})$. The estimate (12ii) then follows from (8) and 1.1(11) using the constants

$$\kappa_t = \frac{1}{2} \max \frac{x_j^2}{|f_t(x_j)|} \max|f_t''(z_j)|,$$

the maxima being taken over all x_j, z_j such that

$$\left|\frac{x_j - u_j}{u_j}\right| \leqslant \xi_j, \quad \left|\frac{z_j - u_j}{u_j}\right| \leqslant 2\xi_j, \qquad j = j_t. \quad \square$$

Applying Taylor's formula to the solutions $x = u$, $x + h = v$ of the algorithm (A) and its perturbation ($\tilde{A}$) gives

$$F_t(v) = F_t(u) + F_t'(u)\Delta u + R_t(u;\Delta u), \tag{13}$$

where $h = v - u = \Delta u$, and thus

$$v_t = (1 + e_t)\{u_t + F_t'(u)\Delta u + R_t(u;\Delta u)\}.$$

The *absolute a priori errors* $\Delta u_t = v_t - u_t$ therefore satisfy the system of equations

$$\Delta u_0 = u_0 e_0, \quad \Delta u_t - F_t'(u)\Delta u = u_t e_t + T_t, \qquad t = 1, \ldots, n, \tag{14}$$

and the *relative a priori errors* $Pu_t = (v_t - u_t)/u_t$ the system

$$Pu_0 = e_0, \quad Pu_t - \frac{1}{u_t} F_t'(u) J_u Pu = e_t + \frac{1}{u_t} T_t, \qquad t = 1, \ldots, n, \tag{15}$$

using the remainder terms

$$T_t = e_t F_t'(u)\Delta u + (1 + e_t)R_t(u;\Delta u). \tag{16}$$

Similarly, the Taylor formula can be applied for $x = v$, $x + h = u$ and $h = u - v = \Delta v$. Then

$$F_t(u) = F_t(v) + F_t'(v)\Delta v + R_t(v;\Delta v).$$

From (A), ($\tilde{A}$), and (2) it is seen that $F_t(u) = u_t$ and

$$F_t(v) = (1 + e_t')v_t, \quad \frac{1}{v_t} = \frac{1}{F_t(v)} - \frac{e_t}{v_t}, \quad e_t' = -\frac{e_t}{1 + e_t}.$$

Hence the *absolute a posteriori errors* $\Delta v_t = u_t - v_t$ satisfy the equations

$$\Delta v_0 = v_0 e_0', \quad \Delta v_t - F_t'(v)\Delta v = v_t e_t' + R_t(v;\Delta v), \qquad t = 1, \ldots, n. \tag{17}$$

By (2), e_t' denotes the relative a posteriori error of the approximation v_t of $F_t(v)$. Dividing both sides of (17) by v_t yields the equations

$$Pv_0 = e_0', \quad Pv_t - \frac{1}{v_t} F_t'(v) J_v Pv = e_t' + \frac{1}{v_t} R_t(v;\Delta v), \qquad t = 1, \ldots, n, \tag{18}$$

for the *relative a posteriori errors* $Pv_t = (u_t - v_t)/v_t$. Approximating v_t by $F_t(v)$ yields the corresponding systems

$$\begin{aligned} &\Delta v_0 = v_0 e_0', \quad \Delta v_t - \frac{v_t}{F_t(v)} F_t'(v)\Delta v = v_t e_t' + R_t', \\ &Pv_0 = e_0', \quad Pv_t - \frac{1}{F_t(v)} F_t'(v) J_v Pv = e_t' + \frac{1}{v_t} R_t', \end{aligned} \tag{19}$$

for $t = 1, \ldots, n$, where R_t' denotes the remainder terms

$$R_t' = -e_t F_t'(v)\Delta v + R_t(v; \Delta v). \tag{20}$$

In (17), (18), (19), additionally, e_t' can be approximated by $-e_t$ and the associated remainder $-e_t e_t'$ be added to R_t'.

The above error equations have the general form

$$z_0 = f_0, \quad z_t - \sum_{k=0}^{t-1} b_{tk} z_k = f_t, \qquad t = 1, \ldots, n. \tag{21}$$

The coefficients b_{tk} vanish for $k \geqslant t$ because $\partial F_t/\partial x_k = 0$ due to (6). The coefficients b_{tk}^{abs}, b_{tk}^{rel} of the equations (14), (15), for $z_t = \Delta u_t$, Pu_t read

$$b_{tk}^{\text{abs}} = \frac{\partial F_t}{\partial x_t}(u), \quad b_{tk}^{\text{rel}} = \frac{u_k}{u_t}\frac{\partial F_t}{\partial x_k}(u), \qquad t, k = 0, \ldots, n. \tag{22}$$

For input operations, by (7),

$$b_{tk}^{\text{abs}} = b_{tk}^{\text{rel}} = 0, \tag{23}$$

and for 'built-in' functions

$$b_{tk}^{\text{abs}} = f'(x_j)\delta_{jk}, \quad b_{tk}^{\text{rel}} = \frac{x_j}{f(x_j)} f'(x_j)\delta_{jk}, \qquad j = j_t. \tag{24}$$

The coefficients b_{tk}^{abs} of the arithmetic operations $F_t \in$ (F2) are listed in Table 1.2. Note that the coefficients for additions and subtractions are particularly simple in the absolute a priori error equations, and for multiplications and divisions in the relative a priori error equations. This fact can be used advantageously in the error analysis of specific examples.

TABLE 1.2

Matrix elements in the absolute and relative a priori error equations where $i = i_t, j = j_t$

0	$k = i$	$k = j$	$k = i$	$k = j$	$k \neq i, j$
	b_{tk}^{abs}		b_{tk}^{rel}		
+	1	1	$\frac{u_i}{u_t}$	$\frac{u_j}{u_t}$	0
−	1	−1	$\frac{u_i}{u_t}$	$-\frac{u_j}{u_t}$	0
×	u_j	u_i	1	1	0
/	$\frac{1}{u_j}$	$-\frac{u_t}{u_j}$	1	−1	0

The representation of the matrix elements in Table 1.2 is so chosen that it also yields easily computable coefficients of a posteriori error equations. These are obtained by replacing u_i, u_j, u_t by the solutions v_i, v_j, v_t of the perturbed algorithm ($\tilde{A}$). In this way, the absolute a posteriori error equations for additions, subtractions, and multiplications are given the coefficients

$$b_{tk}^{\mathrm{abs}} = \frac{\partial F_t}{\partial x_k}(v), \qquad k = 0, \ldots, n, \tag{25}$$

and for divisions $F_t(v) = v_i / v_j$,

$$b_{tk}^{\mathrm{abs}} = \frac{\partial F_t}{\partial x_k}(v), \quad k \neq j; \qquad b_{tj}^{\mathrm{abs}} = \frac{v_t}{F_t(v)} \frac{\partial F_t}{\partial x_j}(v). \tag{26}$$

The relative a posteriori error equations for additions and subtractions take on the form (18), that is, (21) has the coefficients

$$b_{tk}^{\mathrm{rel}} = \frac{v_k}{v_t} \frac{\partial F_t}{\partial x_k}(v), \qquad k = 0, \ldots, n, \tag{27}$$

whereas for multiplications and divisions the form (19) is obtained such that the coefficients in (21) become

$$b_{tk}^{\mathrm{rel}} = \frac{v_k}{F_t(v)} \frac{\partial F_t}{\partial x_k}(v), \qquad k = 0, \ldots, n. \tag{28}$$

An error estimate for the solutions of these *modified a posteriori error equations* will be given in Theorem 2.2(15).

1.3. *The Linear Error Equations.* Fundamental notions of the perturbation theory for evaluation algorithms are the associated mappings F and $A = I - F$ in $\mathbf{R}^{n+1}$ that will be introduced now. The functions $F_0, \ldots, F_n$, specifying the algorithm (A), constitute the vector-valued mapping

$$F(x) = (F_0(x), \ldots, F_n(x)) \tag{1}$$

for all $x \in \mathbf{R}^{n+1}$ such that $x \in \operatorname{def} F_t$ for all t. The solution $u = (u_0, \ldots, u_n)$ of the algorithm (A) may thus be viewed as *fixed point* of F,

$$u = F(u). \tag{2}$$

Let $v = (v_0, \ldots, v_n)$ denote the solution of the perturbed algorithm ($\tilde{A}$). By inserting v into (2), the associated *residual* $d = v - F(v)$ is determined. In view of 1.2.($\tilde{A}$), obviously,

$$d_t = e_t F_t(v) = -e_t' v_t, \qquad t = 0, \ldots, n, \tag{3}$$

that is, $d = -J_v e'$.

The algorithm (A) defines uniquely the mapping

$$Ax = x - F(x) = (A_0 x, \ldots, A_n x), \tag{4}$$

having the components

$$A_0 x = x_0 - F_0, \quad A_t x = x_t - F_t(x), \qquad t = 1, \ldots, n,$$

for $x \in \operatorname{def} A = \operatorname{def} F$. The solution u of the unperturbed algorithm (A) is, by (2), a solution of the functional equation

$$Au = 0, \tag{5}$$

and the solution v of the perturbed algorithm ($\tilde{A}$) satisfies the equation

$$Av = d, \tag{6}$$

using the residual vector (3). In this way, the error analysis of the perturbed algorithm becomes a perturbation theory of the mapping A in a neighborhood of u.

From Section 1.2 it follows that the mappings F, A are Fréchet-differentiable at each interior point of their domain of definition. The derivative $F'(x)$ of F is represented by the matrix

$$F'(x) = \left(\frac{\partial F_t}{\partial x_k}(x)\right)_{t,k=0,\ldots,n}, \tag{7}$$

and the associated remainder term by

$$R^F(x;h) = F(x+h) - F(x) - F'(x)h = (R_t(x;h))_{t=0,\ldots,n}, \tag{8}$$

where R_t denotes the remainder terms of the Taylor formula 1.2(5). Hereby also the mapping A is differentiable at all interior points of def A = def F, where

$$A'x = I - F'(x), \qquad x \in \text{def } F', \tag{9}$$

and

$$R^A(x;h) = A(x+h) - Ax - (A'x)h = -R^F(x;h). \tag{10}$$

The derivative $A'u$ is called the *derivative of the algorithm* (A) and correspondingly $A'v$ the derivative of the perturbed algorithm ($\tilde{A}$). From the representation 1.2(6) of the gradients $F_t'(x)$ it is seen that the matrix of $A'x$ is lower triangular and its diagonal elements are equal to 1. Consequently,

(11) *$A'x$ is a bijective linear mapping of $\mathbf{R}^{n+1}$ for each $x \in$ def A' = def F'.*

Using these mappings, the absolute and relative a priori error equations 1.2(14), (15) may be written in the concise form

$$(A'u)\Delta u = J_u e + T, \qquad (A'u)_{\text{rel}}\, Pu = e + J_u^{-1}T. \tag{12}$$

Analogously, the absolute and relative a posteriori error equations 1.2(17), (18) read

$$(A'v)\Delta v = J_v e' + R, \qquad (A'v)_{\text{rel}}\, Pv = e' + J_v^{-1}R, \tag{13}$$

where

$$(A'u)_{\text{rel}} = J_u^{-1}(A'u)J_u, \qquad (A'v)_{\text{rel}} = J_v^{-1}(A'v)J_v. \tag{14}$$

We shall see in Section 2.1 that the remainder terms R, T are of second order in ε as long as the local errors are bounded by $|e_t| \leqslant \varepsilon$, $|e_t'| \leqslant \varepsilon$, and ε is sufficiently small. Hence neglecting the remainder term T in (12) yields the associated *linear a priori error equations*

$$(A'u)s = J_u e, \qquad (A'u)_{\text{rel}}\, r = e. \tag{15}$$

Similarly, by neglecting R in (13), the *linear a posteriori error equations*

$$(A'v)s = J_v e', \qquad (A'v)_{\text{rel}}\, r = e' \tag{16}$$

are found. We shall prove in Chapter 2 that the solutions s, r of (15) approximate the a priori errors Δu, Pu and the solutions s, r of (16) approximate the a posteriori errors Δv, Pv.

The eight linear systems (12), (13), (15), (16) have, by (9), (14), the general form

$$z - Bz = f, \tag{17}$$

using the mappings

(18)

B	absolute	relative
a priori	$F'(u)$	$J_u^{-1}F'(u)J_u$
a posteriori	$F'(v)$	$J_v^{-1}F'(v)J_v$

Explicit expressions for elements b_{ik} of the associated matrices of these mappings B have been listed already in Table 1.2. The solutions z in (12), (13) are given by

(19)

z	absolute	relative
a priori	Δu	Pu
a posteriori	Δv	Pv

The right-hand sides of the linear error equations (15), (16) read

(20)

f	absolute	relative
a priori	$J_u e$	e
a posteriori	$J_v e'$	e'

As we have stated above, $A'u$, $A'v$ and thus $(A'u)_{\text{rel}}$, $(A'v)_{\text{rel}}$ are bijective linear mappings. In order to simplify notation in the further study, the associated inverse operators are called *solution operators* and are denoted by L. The solutions of the above eight linear systems are then written

$$z = Lf, \qquad L = (I - B)^{-1}. \tag{21}$$

In the case of the a priori error equations,

$$L^{\text{abs}} = (A'u)^{-1}, \qquad L^{\text{rel}} = (A'u)_{\text{rel}}^{-1}, \tag{22}$$

and, consequently,

$$L^{\text{abs}} = J_u L^{\text{rel}} J_u^{-1}, \qquad L^{\text{rel}} = J_u^{-1} L^{\text{abs}} J_u. \tag{23}$$

Correspondingly, for the associated a posteriori error equations

$$L^{\text{abs}} = (A'v)^{-1}, \qquad L^{\text{rel}} = (A'v)_{\text{rel}}^{-1}, \tag{24}$$

whence

$$L^{\text{abs}} = J_v L^{\text{rel}} J_v^{-1}, \qquad L^{\text{rel}} = J_v^{-1} L^{\text{abs}} J_v. \tag{25}$$

The representation $L = (I - B)^{-1}$ immediately yields the relations

$$L = BL + I = LB + I. \tag{26}$$

Componentwise, the first relation reads

$$L_{tk} = \sum_{l=k}^{t-1} b_{tl} L_{lk} + \delta_{tk}, \qquad k = 0, \ldots, t, \tag{27}$$

because

$$b_{ik} = 0, \qquad i \leqslant k, \qquad L_{ik} = 0, \qquad i < k. \tag{28}$$

A recurrence of this type for computing the entries of the matrix L is found in Henrici [8, (16–19)]. From the second relation in (26) one obtains

$$L_{tt} = 1, \quad L_{tk} = \sum_{l=k+1}^{t} L_{tl} b_{lk}, \qquad k = 0, \ldots, t-1. \tag{29}$$

Using the row vectors

$$L_t = (L_{tk})_{k=0,\ldots,n}, \quad b_t = (b_{tk})_{k=0,\ldots,n}, \quad \delta_t = (\delta_{t0}, \ldots, \delta_{tn}),$$

we can write $L = LB + I$ in the form

$$L_t = \sum_{l=1}^{t} L_{tl} b_l + \delta_t. \tag{30}$$

Now, put

$$L_t^{(j)} = \sum_{l=j+1}^{t} L_{tl} b_l + \delta_t, \qquad j = 0, \ldots, t.$$

From (28), (29) it is seen that

$$L_{tk}^{(j)} = \sum_{l=k+1}^{t} L_{tl} b_{lk} = L_{tk}, \qquad j \leqslant k, k = 0, \ldots, t-1,$$

thus, in particular, $L_{tj} = L_{tj}^{(j)}$. Consequently, the row L_t of the matrix L can be computed recurrently from

$$L_t^{(t)} = \delta_t, \quad L_t^{(j-1)} = L_t^{(j)} + L_{tj}^{(j)} b_j, \qquad j = t, \ldots, 1, \tag{31}$$

and

$$L_t = L_t^{(0)}. \tag{32}$$

This is the algorithm (T) of Linnainmaa [10] applied to the computation of the row L_t of the matrix L from the rows of the matrix B. The same algorithm has been proposed by Larson-Sameh [9] in a somewhat different setting.

2. Condition Numbers, Error Estimates, Graphs. In Sections 2.1, 2.2 it will be shown that the solutions s, r of the linear a priori and a posteriori error equations yield approximations of the a priori errors Δu, Pu and a posteriori errors Δv, Pv. Simultaneously, estimates of these errors will be obtained. A fundamental tool of the error analysis is the notion of condition number. The error approximations s_t, r_t are linear forms in the local errors specifying the perturbations of the algorithm. Our condition numbers are norms of these linear forms with respect to suitably chosen weighted maximum norms over the space of local errors. Therefore condition numbers are optimal bounds of the error approximations s_t, r_t for all local errors in the considered distribution. Consequently, also the associated estimates of the errors Δu, Pu and Δv, Pv are optimal if terms $O_t(\varepsilon)$ are neglected against 1. In addition, Section 2.2 demonstrates that the solutions of the linear a posteriori error equations are approximations of the associated solutions of the a priori error equations and thus the a posteriori condition numbers approximations of the a priori condition numbers. Moreover, it will be shown that, using solutions of the linear a posteriori error equations, approximations of about double precision can be computed.

Each algorithm and its uniquely associated system of linear error equations determines a graph, defined in Section 2.3. Graphs constitute useful means of the error analysis, particularly also for deriving condition numbers of the algorithm. For example, if the graph is a tree, the condition numbers can be determined a priori and be computed a posteriori by simple recursion formulae. In all cases, bounds for condition numbers and hence for error estimates can recursively be obtained and computed a posteriori. For examples we refer to Stummel [17]–[23].

2.1. *A Priori Condition Numbers and Error Estimates.* In the preceding sections the general linear error equations have been established for the class of algorithms considered here. Using the associated solution operators $L = (I - B)^{-1}$, the solutions z of these linear systems have the form $z = Lf$. Due to 1.2(6), 1.3(18), the mappings B are represented by lower-triangular matrices with zeros as diagonal elements. Consequently, the matrices of the solution operators are lower triangular with all diagonal elements equal to 1. By L_{ik} are meant the $(n + 1)^2$ elements of the matrix of L. Then

$$z_t = f_t + \sum_{k=0}^{t-1} L_{tk} f_k, \qquad t = 0, \ldots, n, \tag{1}$$

as

$$L_{tt} = 1, \quad L_{tk} = 0, \qquad t < k.$$

For each t the right-hand side of (1) is a *linear form* L_t over $\mathbf{R}^{n+1}$. Evidently,

$$L_t = \mathrm{pr}_t L, \tag{2}$$

where pr_t is the *projection* onto the tth coordinate axis. The solutions s_t, r_t of the linear a priori error equations 1.3(15) then have the representation

$$s_t = L_t^{\mathrm{abs}} J_u e, \qquad r_t = L_t^{\mathrm{rel}} e. \tag{3}$$

Of fundamental importance in the following are the associated absolute and relative a priori condition numbers

$$\sigma_t^1 = \sum_{k=0}^{t} |L_{tk}^{\mathrm{abs}} u_k|, \quad \rho_t^1 = \sum_{k=0}^{t} |L_{tk}^{\mathrm{rel}}|, \qquad t = 0, \ldots, n. \tag{4}$$

From $L_{tt} = 1$ it follows that

$$\rho_0^1 = 1, \qquad \rho_t^1 \geqslant 1. \tag{5}$$

By virtue of 1.3(23),

$$L^{\mathrm{abs}} J_u = J_u L^{\mathrm{rel}}, \tag{6}$$

thus

$$s_t = u_t r_t, \qquad \sigma_t^1 = |u_t| \rho_t^1. \tag{7}$$

Using the above condition numbers, the solutions s_t, r_t satisfy the inequalities

$$|s_t| \leqslant \sigma_t^1 \varepsilon, \qquad |r_t| \leqslant \rho_t^1 \varepsilon, \tag{8}$$

for all $|e_k| \leqslant \varepsilon$, $k = 0, \ldots, n$. These estimates are *sharp* or *optimal* because the bounds $\sigma_t^1 \varepsilon$, $\rho_t^1 \varepsilon$ are attained for the local error distributions

$$e_k^{\mathrm{abs}} = \varepsilon \operatorname{sgn}(L_{tk}^{\mathrm{abs}} u_k), \quad e_k^{\mathrm{rel}} = \varepsilon \operatorname{sgn}(L_{tk}^{\mathrm{rel}}), \qquad k = 0, \ldots, n. \tag{9}$$

By these means we are now in the position to establish error estimates for the approximations s, r of the a priori errors Δu, Pu. By 1.3(12), (15), (22), obviously,

$$\Delta u_t - s_t = L_t^{\mathrm{abs}} T, \qquad Pu_t - r_t = L_t^{\mathrm{rel}} J_u^{-1} T, \tag{10}$$

whence, using (4),

$$|\Delta u_t - s_t| \leqslant \sigma_t^1 \max_{j<t} \left| \frac{1}{u_j} T_j \right|, \qquad |Pu_t - r_t| \leqslant \rho_t^1 \max_{j<t} \left| \frac{1}{u_j} T_j \right|. \tag{11}$$

We assume throughout the paper that the solution u of (A) is an interior point of def F_t and $u_t \neq 0$ for all t. The first main theorem of the paper then reads:

(12) *For all local errors* $|e_t| \leqslant \varepsilon$, $t = 0, \ldots, n$, *and sufficiently small* ε *the perturbed algorithm* ($\tilde{A}$) *is well defined. The solutions* s, r *of the linear a priori error equations are approximations of the absolute and relative errors* Δu, Pu *with the associated error estimates*

$$\text{(i)} \qquad |\Delta u_t - s_t| \leqslant \sigma_t^1 \omega_t \varepsilon^2, \quad |Pu_t - r_t| \leqslant \rho_t^1 \omega_t \varepsilon^2, \qquad t = 0, \ldots, n.$$

In particular, Δu, Pu *permit the estimates*

$$\text{(ii)} \qquad |\Delta u_t| \leqslant (1 + \omega_t \varepsilon)\sigma_t^1 \varepsilon, \quad |Pu_t| \leqslant (1 + \omega_t \varepsilon)\rho_t^1 \varepsilon, \qquad t = 0, \ldots, n.$$

Proof. (i) First let us define the constants

$$\beta_t' = \max_{j<t} \sum_{k=0}^{j-1} |b_{jk}^{\mathrm{rel}}|, \qquad \kappa_t' = \max_{j<t} \kappa_j,$$

where κ_j denotes the constants in the remainder estimates 1.2(12) at the point u. Next let $\beta = \beta_n'$, $\kappa = \kappa_n'$, and $\rho = \max \rho_t^1$. For arbitrary but fixed $\vartheta > 0$ put

$$\omega = (\beta + (1 + \vartheta)\kappa\tau)\tau, \qquad \tau = (1 + \vartheta)\rho.$$

As u is an interior point of def F, there exists a positive constant ξ such that for each t the statement

$$|x_j - u_j| \leqslant \xi |u_j|, \qquad j = 0, \ldots, t-1 \Rightarrow (x_0, \ldots, x_{t-1}) \in \operatorname{def} F_t$$

is true and the estimates 1.2(12) are applicable for $x = u$ and all $|h_j| \leqslant \xi |u_j|$, $j = 0, \ldots, n$. Finally, let

$$\varepsilon \leqslant \varepsilon_0 = \min\left(\vartheta, \frac{\vartheta}{\omega}, \frac{\xi}{\tau}\right).$$

(ii) The theorem will be proved by finite induction. For $t = 0$, $Pu_0 = r_0 = e_0$ and $\rho_0^1 = 1$, so that the inequalities (12i), (12ii) are valid with $\omega_0 = 0$. For $t > 1$ assume now that the proposition

$$|v_j - u_j| \leqslant \xi |u_j|, \quad |Pu_j| \leqslant (1 + \omega_j \varepsilon)\rho_j^1 \varepsilon, \qquad \omega_j < \omega,$$

is true for $j = 0, \ldots, t-1$. From (8), (11) it then follows that

$$|Pu_t| \leqslant \left(1 + \frac{1}{\varepsilon} \max_{j<t} \left|\frac{1}{u_j} T_j\right|\right)\rho_t^1 \varepsilon,$$

using the remainder terms T_j in 1.2(16), that is,

$$\frac{1}{u_j} T_j = e_j B_j^{\mathrm{rel}} Pu + (1 + e_j)\frac{1}{u_j} R_j(u_j; \Delta u).$$

On setting

$$\text{(13)} \qquad \omega_{t-1}' = \max_{j<t} \omega_j, \quad \rho_{t-1}' = \max_{j<t} \rho_j^1, \quad \tau_t = (1 + \omega_{t-1}' \varepsilon)\rho_{t-1}',$$

the above proposition and Theorem 1.2(12) imply

$$\max_{j<t} |Pu_j| \leqslant \tau_t \varepsilon, \quad \max_{j<t} \left|\frac{1}{u_j} T_j\right| \leqslant \tau_t(\beta_t' + (1 + \varepsilon)\kappa_t' \tau_t)\varepsilon^2.$$

Consequently,

$$\max_{j<t}\left|\frac{1}{u_j}T_j\right| \leqslant \omega_t\varepsilon^2,$$

where $\omega_t = \tau_t(\beta_t' + (1+\varepsilon)\kappa_t'\tau_t)$, $t = 1, \ldots, n$. Finally, the above proposition entails $\omega_{t-1}' \leqslant \omega$, $\tau_t \leqslant (1+\omega\varepsilon)\rho \leqslant \tau$. Thus $\omega_t \leqslant (\beta + (1+\vartheta)\kappa\tau)\tau = \omega$ and

$$\left|\frac{v_t - u_t}{u_t}\right| = |Pu_t| \leqslant (1+\omega_t\varepsilon)\rho_t^1\varepsilon \leqslant \tau\varepsilon \leqslant \xi$$

for all $\varepsilon \leqslant \varepsilon_0$. Hence the above proposition is true also for $j = t$ and consequently for all $j = 0, \ldots, n$. The estimates of the remainder terms T_t/u_t, proved above, immediately yield the error estimates (12i), (12ii). □

The above theorem guarantees that the solutions s, r of the linear error equations are equal to the a priori errors Δu, Pu save for terms of second order in ε,

$$\Delta u_t = s_t + O_t(\varepsilon^2), \qquad Pu_t = r_t + O_t(\varepsilon^2). \tag{14}$$

On this basis, error estimates can be derived regarding specific distributions of the local errors. Let us assume that

$$|e_t| \leqslant \gamma_t\eta, \qquad t = 0, \ldots, n, \tag{15}$$

where $\gamma_0, \ldots, \gamma_n$ are appropriate nonnegative weights. Theorem (12) applies to this specific distribution of local errors in choosing $\varepsilon = \max \gamma_t\eta$. However, the solutions of the linear error equations can be estimated finer by

$$|s_t| = |L_t^{\mathrm{abs}}J_u e| \leqslant \sigma_t\eta, \qquad |r_t| = |L_t^{\mathrm{rel}}e| \leqslant \rho_t\eta, \tag{16}$$

using the *weighted absolute* and *relative a priori condition numbers*

$$\sigma_t = \sum_{k=0}^{t} |L_{tk}^{\mathrm{abs}}u_k|\gamma_k, \quad \rho_t = \sum_{k=0}^{t} |L_{tk}^{\mathrm{rel}}|\gamma_k, \qquad t = 0, \ldots, n. \tag{17}$$

In cases, for example, when F_t is an input operation of a floating-point number represented exactly in the computer, the associated γ_t may be set equal to zero. When a subtraction F_t is performed with loss of significant figures but without rounding error, we may put $\gamma_t = 0$. If a built-in function $F_t = f_t$ is evaluated in lower precision than the arithmetic operations are, the associated γ_t may be chosen suitably greater than 1. In particular, the behavior of the algorithm under data perturbations only and under rounding errors in the 'built-in' functions and arithmetic operations only can be analyzed. For this purpose, to a given sequence of weights (γ_t) of the local error distribution, put

$$\gamma_t^D = \gamma_t, \quad F_t \in (\mathrm{F0}); \quad \gamma_t^R = \gamma_t, \quad F_t \in (\mathrm{F1}) \cup (\mathrm{F2}); \tag{18i}$$

and γ_t^D, $\gamma_t^R = 0$ else for $t = 0, \ldots, n$. The sequences of weights (γ_t^D), (γ_t^R) specify, by (17), *weighted absolute* and *relative data condition numbers* σ_t^D, ρ_t^D *and rounding condition numbers* σ_t^R, ρ_t^R. In this way, the decompositions

$$\gamma_t = \gamma_t^D + \gamma_t^R, \quad \sigma_t = \sigma_t^D + \sigma_t^R, \quad \rho_t = \rho_t^D + \rho_t^R, \tag{18ii}$$

are obtained. The following corollary states error estimates using weighted condition numbers.

(19) *Let any nontrivial sequence of weights* $\gamma_0, \ldots, \gamma_n$ *be given. For every sequence of local errors in the distribution* (15) *and sufficiently small* $\varepsilon = \gamma\eta$ *the solutions of the*

perturbed algorithm ($\tilde{A}$) *then satisfy the error estimates*

(i) $$|\Delta u_t| \leqslant \sigma_t\eta + \sigma_t^1\omega_t\gamma^2\eta^2, \quad |Pu_t| \leqslant \rho_t\eta + \rho_t^1\omega_t\gamma^2\eta^2, \qquad t = 0, \ldots, n.$$

These estimates are optimal in the sense that for each j *there exists a sequence* (e_t) *in* (15) *and an associated sequence* (v_t) *of solutions of* ($\tilde{A}$) *such that*

(ii) $$\pm\Delta u_j = \sigma_j\eta + O_j(\eta^2), \qquad Pu_j = \rho_j\eta + O_j(\eta^2).$$

Proof. (i) The error estimates (12i) and the inequalities (16) immediately entail the estimate (19i). For any j, in particular, the local rounding error distribution

$$e_k = \eta\gamma_k \operatorname{sgn}\left(L_{jk}^{\mathrm{rel}}\right), \qquad k = 0, \ldots, n,$$

may be chosen. Thus the solutions of the linear error equations become

$$s_j = u_j r_j = \pm\sigma_j\eta, \qquad r_j = \sum_{k=0}^{t} L_{jk}^{\mathrm{rel}} e_k = \rho_j\eta.$$

For sufficiently small η the so perturbed algorithm ($\tilde{A}$) is well defined. That is, there exists an associated solution $v = (v_0, \ldots, v_n)$ of ($\tilde{A}$) satisfying the error estimates (12i). This yields

$$|\pm\Delta u_j - \sigma_j\eta| \leqslant \sigma_j^1\omega_j\gamma^2\eta^2, \qquad |Pu_j - \rho_j\eta| \leqslant \rho_j^1\omega_j\gamma^2\eta^2,$$

whence (19ii) follows. □

Before we continue our study let us first define the notion of *sensitivity* of a solution z of a given *problem* with respect to data perturbations. Let the solution z be a function Z of m parameters in a neighborhood U of a data vector $c = (c_1, \ldots, c_m)$. Let Z be Fréchet-differentiable at the point c and let $c_i \neq 0$ for $i = 1, \ldots, m$. Then, for sufficiently small η, all data vectors $c' = (c_1', \ldots, c_m')$ having the property

(20) $$|Pc_i| \leqslant \gamma_i^c\eta, \qquad i = 1, \ldots, m,$$

belong to the neighborhood U. The constants $\gamma_1^c, \ldots, \gamma_m^c$ are positive weights. Obviously, the error

(21) $$\Delta z = Z(c') - Z(c) = Z'(c)J_c Pc + o(\eta)$$

is bounded, optimally, by

$$|\Delta z| \leqslant \sum_{i=1}^{m} \left|\frac{\partial Z}{\partial c_i}(c)c_i\right|\gamma_i^c\eta + o(\eta)$$

for all c' in the neighborhood (20) of c. The factor of η on the right-hand side of the last inequality is an *asymptotic absolute condition* $\|Z'(c)\|$ *of the function* Z in the sense of Rice [15] if U is equipped with the norm

$$\|h\| = \max_{i=1,\ldots,m} \frac{1}{\gamma_i^c}\left|\frac{h_i}{c_i}\right|, \qquad h = (h_1, \ldots, h_m).$$

Now choose any algorithm of the kind that is considered in this paper for computing the solution of the given problem such that under data perturbations only, assuming exact 'built-in' functions and arithmetic operations,

$$u_{t_i} = c_i, \quad v_{t_i} = c_i', \quad \gamma_{t_i}^D = \gamma_i^c, \qquad i = 1, \ldots, m,$$

$\gamma_t^D = 0$ for $t \notin \{t_1, \ldots, t_m\}$, and $Z(c) = u_t$, $Z(c') = v_t$ for all c' in the neighborhood (20) of c. Thus, on the one hand (21) holds and on the other hand, by (3), (14),

$$\Delta u_t = L_t^{\text{abs}} J_u e^D + O_t(\eta^2),$$

where $e_{t_i}^D = Pc_i$, $e_t^D = 0$ for $t \notin \{t_1, \ldots, t_m\}$. Consequently,

$$L_t^{\text{abs}} J_u e^D = Z'(c) J_c Pc + o(\eta)$$

for all $Pc_i = e_{t_i}^D$ in (20). From this relation one readily concludes that the coefficients of the two linear forms coincide and, consequently,

$$\sigma_t^D = \sum_{k=0}^{t} |L_{tk}^{\text{abs}} u_k| \gamma_k^D = \sum_{i=1}^{m} \left| \frac{\partial Z}{\partial c_i}(c) c_i \right| \gamma_i^c. \tag{22}$$

This result shows that the value of σ_t^D is independent of the special evaluation algorithm and a measure of the data sensitivity of the solution of the given problem.

The well-known *backward analysis* represents perturbations of the algorithm under rounding errors by means of data perturbations. This procedure may now be specified quantitatively as follows. Error distributions (15) with the weights (γ_t^R) and accuracy constant $\eta = \eta_R$ induce perturbations of function evaluations and arithmetic operations only, the data remain unperturbed. By Theorem (12) and (16), (17), the absolute errors $(\Delta u_t)^R = v_t^R - u_t$ of the sequence (v_t^R) of solutions of the so perturbed algorithm satisfy the relations

$$(\Delta u_t)^R = s_t^R + O_t(\eta_R^2), \qquad |s_t^R| \leqslant \sigma_t^R \eta_R.$$

For any index j such that $\sigma_j^D \neq 0$ choose the data accuracy

$$\eta_D = \frac{\sigma_j^R}{\sigma_j^D} \eta_R = \frac{\rho_j^R}{\rho_j^D} \eta_R. \tag{23}$$

Under data perturbations only, the associated linear forms $L_j^{\text{abs}} J_u e^D$ then have the closed intervals

$$\left[-\sigma_j^D \eta_D, +\sigma_j^D \eta_D\right] = \left[-\sigma_j^R \eta_R, +\sigma_j^R \eta_R\right]$$

as their range. Hence, s_j^R belongs to this range and there exists a sequence e^D of local errors in $|e_t^D| \leqslant \gamma_t^D \eta_D$ such that $s_j^R = L_j^{\text{abs}} J_u e^D = s_j^D$. By virtue of Theorem (12), using this sequence e^D of local errors and the accuracy constant η_D, there exists a sequence (v_t^D) of solutions of the algorithm under data perturbations only such that $(\Delta u_t)^D = s_t^D + O_t(\eta_D^2)$. It follows from the above that η_D is the least constant such that the absolute error $(\Delta u_j)^D = v_j^D - u_j$ gives the representation $(\Delta u_j)^R = (\Delta u_j)^D + O_j(\eta_R^2)$ for arbitrary local rounding errors e_t^R in $|e_t^R| \leqslant \gamma_t^R \eta_R$. Consequently, the constant $\sigma_j^R / \sigma_j^D = \rho_j^R / \rho_j^D$ measures the stability of the algorithm for computing u_j in the sense of Wilkinson's backward error analysis [25, I–39].

An algorithm is called by Bauer [4], [5] well-conditioned (German: gutartig) if the influence of rounding errors in the arithmetic operations is at most of the same order of magnitude as the influence of data perturbations whose magnitude

corresponds to rounding errors. Thus the computation of u_t by the algorithm (A) is *well-conditioned* if the estimates

$$\sigma_t^R \leqslant \beta\sigma_t^D, \qquad \rho_t^R < \beta\rho_t^D \tag{24}$$

hold where the constant β is not much greater than 1.

Example 1. The *product algorithm*

$$u_0 = b_0, \quad u_t = b_t u_{t-1}, \qquad t = 1, \ldots, n, \tag{25}$$

for the computation of $u_n = \prod_{t=0}^{n} b_t$ possesses the condition numbers

$$\rho_t^D = t + 1, \qquad \rho_t^R = t. \tag{26}$$

Hence $\rho_t^R \leqslant \rho_t^D$, so that this algorithm is well-conditioned. □

Example 2. The *summation algorithm*

$$u_0 = c_0, \quad u_t = u_{t-1} + c_t, \qquad t = 1, \ldots, n, \tag{27}$$

for the computation of the sequence of partial sums $u_t = \sum_{k=0}^{t} c_k$, $t = 0, \ldots, n$, has the condition numbers

$$\sigma_t^D = \sum_{j=0}^{t} |c_j|, \qquad \sigma_t^R = \sum_{j=1}^{t} |u_j|. \tag{28}$$

In the case of nonnegative terms it follows that

$$\rho_t^D = 1, \qquad \rho_t^R = \frac{1}{u_t} \sum_{j=1}^{t} u_j \leqslant t. \tag{29}$$

Consequently, the summation algorithm for the computation of u_n is well-conditioned if and only if ρ_n^R is not much greater than 1, so that the relative error Pu_n is bounded by a low multiple of the floating-point accuracy constant η. This condition is not true in many cases. A well-known concrete example is the summation of $c_j = h$ for $j = 0, \ldots, n$. In this case

$$u_t = (t + 1)h, \quad \rho_t^R = \frac{t + 3}{t + 1} \cdot \frac{t}{2}, \qquad t = 1, \ldots, n. \tag{30}$$

For

$$h = .555, \quad n = 200, \quad u_n = 111.555, \quad \rho_n^R \doteq 101,$$

in 3-digit decimal floating point $v_n = 133$, $Pu_n \doteq .192$, is computed. Here $\eta = 5.-03$, $\rho_n^R \eta = .505$, so that this bound overestimates the error only by a factor of 2.63. □

2.2. *A Posteriori Condition Numbers and Error Estimates.* In many examples the condition numbers of the algorithm or appropriate bounds can be computed from the data and intermediate results arising during the computation. For this purpose the a posteriori condition numbers are needed, being specified by the sequence of solutions $v_0, \ldots, v_n$ of the perturbed algorithm ($\tilde{A}$). The linear a posteriori error equations 1.3(16) yield, together with the solution operators 1.3(24), the approximations

$$s_t = L_t^{\text{abs}} J_v e', \qquad r_t = L_t^{\text{rel}} e', \tag{1}$$

of the a posteriori errors Δv_t, Pv_t. The associated *weighted a posteriori condition numbers*

$$\sigma_t = \sum_{k=0}^{t} |L_{tk}^{\mathrm{abs}} v_k| \gamma_k, \qquad \rho_t = \sum_{k=0}^{t} |L_{tk}^{\mathrm{rel}}| \gamma_k \tag{2}$$

are optimal bounds in the estimates

$$|s_t| \leqslant \sigma_t \eta', \qquad |r_t| \leqslant \rho_t \eta' \tag{3}$$

for all local relative a posteriori errors $|e_k'| \leqslant \gamma_k \eta'$, $k = 0, \ldots, n$. In view of 1.3(25),

$$s_t = v_t r_t, \qquad \sigma_t = |v_t| \rho_t. \tag{4}$$

The aim of the following investigation is to compare the solutions of the linear a priori and a posteriori error equations as well as the a priori and a posteriori condition numbers. For distinctness the upper index i indicates a priori terms from Section 2.1 and the upper index 0 a posteriori terms defined above. In addition, the nonweighted a priori condition numbers

$$\sigma_t^1 = \sum_{k=0}^{t} |L_{tk}^i u_k|, \quad \rho_t^1 = \frac{\sigma_t^1}{|u_t|}, \qquad t = 0, \ldots, n, \tag{5}$$

are used in error estimates, L^i denoting the absolute a priori solution operator $(A'u)^{-1}$.

The first lemma is basic for the following. It shows that the derivatives F', $A' = I - F'$ are Lipschitz continuous at u.

(6) *There exist constants ζ_t such that for all local errors $|e_k| \leqslant \varepsilon$, $k = 0, \ldots, n$, and sufficiently small ε the following inequalities are valid*:

$$\text{(i)} \qquad \sum_{k=0}^{t-1} \left| \frac{u_k}{u_t} \left(\frac{\partial F_t}{\partial x_k}(v) - \frac{\partial F_t}{\partial x_k}(u) \right) \right| \leqslant \zeta_t \max_{k<t} |Pu_k|, \qquad t = 0, \ldots, n.$$

Proof. (i) For those t for which F_t is an input operation, $\partial F_t / \partial x_k = 0$, so that the above inequality holds trivially with $\zeta_t = 0$. For additions and subtractions F_t, the partial derivatives $\partial F_t / \partial x_k$ are constant and equal to 0, +1, −1 for each k. Hence for these t the left side in (6i) vanishes, and the inequality is true too with $\zeta_t = 0$.

(ii) Next consider the indices t such that F_t is a multiplication. As is readily seen, now

$$\sum_{k=0}^{t-1} \left| \frac{u_k}{u_t} \left(\frac{\partial F_t}{\partial x_k}(v) - \frac{\partial F_t}{\partial x_k}(u) \right) \right| = |Pu_{i_t}| + |Pu_{j_t}|,$$

whence (6i) follows using the constant $\zeta_t = 2$.

(iii) By virtue of Theorem 2.1(12), there exists an ε_1 such that for all $\varepsilon \leqslant \varepsilon_1$ the solutions v_t of the perturbed algorithm $(\tilde{A})$ are in the neighborhoods defined by 1.2(12i). In particular, then $|Pu_t| \leqslant \frac{1}{3}$ for all t. One easily verifies the estimate

$$\sum_{k=0}^{t-1} \left| \frac{u_k}{u_t} \left(\frac{\partial F_t}{\partial x_k}(v) - \frac{\partial F_t}{\partial x_k}(u) \right) \right| \leqslant \frac{2|Pu_j|}{|1 + Pu_j|} + \frac{|Pu_j - Pu_i|}{|1 + Pu_j|^2}$$

for divisions $F_t(x) = x_i / x_j$. As $|Pu_j| \leqslant \frac{1}{3}$, the denominators are bounded from below by $\frac{2}{3}$ and $\frac{4}{9}$. Thus the inequality (6i) holds in this case with $\zeta_t = 7.5$.

(iv) In function evaluations $F_t(x) = f_t(x_i)$, $i = i_t$, finally,

$$\sum_{k=0}^{t-1} \left| \frac{u_k}{u_t} \left(\frac{\partial F_t}{\partial x_k}(v) - \frac{\partial F_t}{\partial x_k}(u) \right) \right| = \left| \frac{u_i^2}{f_t(u_i)} f_t''(w_i) P u_i \right|,$$

w_i being some intermediate point of the line segment $\overline{u_i v_i}$. By Theorem 1.2(12), the join $\overline{uv}$ belongs to the domain of definition of F'. The right-hand side of the last equation is bounded by $2\kappa_t |Pu_i|$, using the constants κ_t of Theorem 1.2(12). Consequently, (6i) is true for $\zeta_t = 2\kappa_t$. □

By the above lemma, we can now estimate the distance between the a priori and a posteriori solution operators

$$L^i = (A'u)^{-1}, \qquad L^0 = (A'v)^{-1}. \tag{7}$$

(8) *There exists a constant μ such that for all local errors $|e_k| \leqslant \varepsilon$, $k = 0, \ldots, n$, and sufficiently small ε the associated solution operators suffice the following estimates*

$$\text{(i)} \qquad \sum_{k=0}^{t} |L_{tk}^0 - L_{tk}^i| \, |u_k| \leqslant \sigma_t^1 \mu \varepsilon, \qquad t = 0, \ldots, n.$$

Proof. (i) As is readily verified, the difference of the solution operators has the representation

$$L^0 - L^i = -L^i (I + C)^{-1} C,$$

where $C = (A'v - A'u)L^i = (F'(u) - F'(v))L^i$. This implies

$$(L^0 - L^i) J_u = -L^i J_u (I + D)^{-1} D, \tag{9}$$

using the notation

$$D = J_u^{-1} C J_u = J_u^{-1} (F'(u) - F'(v)) L^i J_u.$$

Let us further denote by J_σ the diagonal matrix $\operatorname{diag}(\sigma_0^1, \ldots, \sigma_n^1)$. In the maximum absolute row sum norm then

$$\|J_\sigma^{-1} L^i J_u\| = \max_t \frac{1}{\sigma_t^1} \sum_{k=0}^{t} |L_{tk}^i u_k| = 1. \tag{10}$$

The matrix D is bounded in this norm by

$$\|D\| \leqslant \|J_u^{-1}(F'(u) - F'(v))J_\sigma\| = \max_t \sum_{k=0}^{t-1} \left| \frac{u_k}{u_t} \left(\frac{\partial F_t}{\partial x_k}(v) - \frac{\partial F_t}{\partial x_k}(u) \right) \right| \rho_k^1.$$

From Lemma (6) we infer the further estimate

$$\|D\| \leqslant \zeta \rho \max_t |Pu_t|, \quad \zeta = \max_t \zeta_t, \quad \rho = \max_t \rho_t^1. \tag{11}$$

Now choose any constant ϑ in $0 < \vartheta < 1$ and put

$$\tau = (1 + \vartheta)\rho, \quad \varepsilon \leqslant \min\left(\varepsilon_1, \frac{\vartheta}{\omega}, \frac{\vartheta}{\zeta \rho \tau} \right), \quad \omega = \max_t \omega_t.$$

Theorem 2.1(12) then guarantees that

$$\max_t |Pu_t| \leqslant (1 + \omega\varepsilon)\rho\varepsilon \leqslant \tau\varepsilon.$$

The above estimate (11) hereby yields $\|D\| \leqslant \zeta\rho\tau\varepsilon \leqslant \vartheta < 1$. Finally (9), (10), (11) entail the estimate

$$\|J_\sigma^{-1}(L^0 - L^i)J_u\| \leqslant \frac{\|D\|}{1 - \|D\|} \leqslant \mu\varepsilon,$$

using the constant $\mu = \zeta\rho\tau/(1 - \vartheta)$, and thence the asserted inequality (8i). □

Having made these preparations, we are in the position to prove the *comparison theorem*. By s^i, r^i are meant the solutions of the linear a priori error equations 1.3(15), and by s^0, r^0 the solutions of the linear a posteriori error equations 1.3(16). Further, σ_t^i, ρ_t^i denote the weighted a priori condition numbers 2.1(17) and σ_t^0, ρ_t^0 the a posteriori condition numbers (2).

(12) *For all local errors* $|e_k| \leqslant \gamma_k\eta$, $k = 0, \ldots, n$, *and sufficiently small* η *the approximations of the absolute and relative a priori and a posteriori errors satisfy the relations*

(i) $$s_t^0 = -s_t^i + O_t(\eta^2), \qquad r_t^0 = -r_t^i + O_t(\eta^2),$$

and the associated weighted condition numbers the relations

(ii) $$\sigma_t^0 = \sigma_t^i + O_t(\eta), \qquad \rho_t^0 = \rho_t^i + O_t(\eta).$$

Proof. (i) The solutions of the linear absolute a priori and a posteriori error equations have the representations $s^i = L^iJ_ue$, $s^0 = L^0J_ve'$. Thus,

$$s_t^i + s_t^0 = L_t^i(J_u - J_v)e + L_t^iJ_v(e + e') + (L_t^0 - L_t^i)J_ve'.$$

The first term on the right side is bounded by

$$|L_t^i(J_u - J_v)e| \leqslant \sum_{k=0}^{t} |L_{tk}^i u_k Pu_k e_k| \leqslant \sigma_t^i\|Pu\|\eta,$$

using the maximum norm for Pu. By virtue of 1.1(8), $e_k + e_k' = -e_ke_k'$. For $|e_k| \leqslant \gamma_k\eta$ it follows that

$$|e_k'| = \left|\frac{e_k}{1 + e_k}\right| \leqslant \frac{\gamma_k\eta}{1 - \gamma\eta} \leqslant \gamma'\eta, \quad \gamma = \max \gamma_k, \quad \gamma' = \frac{\gamma}{1 - \gamma\eta}.$$

Therefore, the second term on the right side suffices

$$|L_t^iJ_v(e + e')| \leqslant \sum_{k=0}^{t} \left|L_{tk}^i u_k \frac{v_k}{u_k} e_ke_k'\right| \leqslant \sigma_t^i(1 + \|Pu\|)\gamma'\eta^2.$$

By virtue of Lemma (8) for $\varepsilon = \gamma\eta$, the third term can finally be estimated by

$$|(L_t^0 - L_t^i)J_ve'| \leqslant \sum_{k=0}^{t} \left|(L_{tk}^0 - L_{tk}^i)u_k \frac{v_k}{u_k} e'\right| \leqslant \sigma_t^1\mu\gamma(1 + \|Pu\|)\gamma'\eta^2.$$

On choosing ε as small as in the proof of Lemma (8), we have $\|Pu\| \leqslant \tau\varepsilon$ and $\|Pu\| \leqslant \frac{1}{3}$. The above appraisals then entail $|s_t^i + s_t^0| \leqslant (\sigma_t^i\phi + \sigma_t^1\psi)\eta^2$, using the constants

$$\phi = \tau\gamma + \tfrac{4}{3}\gamma', \qquad \psi = \tfrac{4}{3}\mu\gamma\gamma'.$$

(ii) The solutions r^i, r^0 of the linear relative a priori and a posteriori error equations permit the representation

$$r_t^i + r_t^0 = \frac{1}{1 + Pu_t}\left\{\frac{s_t^i + s_t^0}{u_t} + r_t^iPu_t\right\}.$$

For $\gamma\eta = \varepsilon \leqslant \varepsilon_1$, again $|Pu_t| \leqslant \tau\varepsilon$ and $|Pu_t| \leqslant \frac{1}{3}$. Using the above estimates for $s_t^i + s_t^0$ and 2.1(16), thus

$$\tfrac{2}{3}|r_t^i + r_t^0| \leqslant (\rho_t^i\phi + \rho_t^1\psi)\eta^2 + \rho_t^i\tau\gamma\eta^2.$$

(iii) The weighted absolute condition numbers satisfy the relations

$$\frac{1}{\gamma}|\sigma_t^i - \sigma_t^0| \leqslant \sum_{k=0}^{t} |L_{tk}^i u_k - L_{tk}^0 v_k|$$

$$\leqslant \sum_{k=0}^{t} |L_{tk}^i u_k Pu_k| + \sum_{k=0}^{t} |(L_{tk}^0 - L_{tk}^i)v_k|.$$

The first term on the right side can be estimated by

$$\sum_{k=0}^{t} |L_{tk}^i u_k Pu_k| \leqslant \sigma_t^1 \|Pu\| \leqslant \sigma_t^1\tau\gamma\eta.$$

Using Lemma (8), the second term possesses the upper bound

$$\sum_{k=0}^{t} |(L_{tk}^0 - L_{tk}^i)u_k(1 + Pu_k)| \leqslant \sigma_t^1\mu\varepsilon(1 + \|Pu\|).$$

Hence, for $\varepsilon = \gamma\eta \leqslant \varepsilon_1$, the above yields

$$|\sigma_t^i - \sigma_t^0| \leqslant \sigma_t^1\chi\eta, \qquad \chi = \left(\tau + \tfrac{4}{3}\mu\right)\gamma.$$

(iv) Finally, the difference of the weighted relative a priori and a posteriori condition numbers is bounded by

$$|\rho_t^i - \rho_t^0| \leqslant \frac{1}{|1 + Pu_t|}\left\{\frac{|\sigma_t^i - \sigma_t^0|}{|u_t|} + \rho_t^i|Pu_t|\right\}.$$

Together with the estimate for $|\sigma_t^i - \sigma_t^0|$ this entails

$$\tfrac{2}{3}|\rho_t^i - \rho_t^0| \leqslant \rho_t^1\chi\eta + \rho_t^i\tau\gamma\eta. \quad \square$$

In computing a posteriori condition numbers or solutions of linear a posteriori error equations, Table 1.2 yields modified coefficients. The modified relative a posteriori error equations $\tilde{r} - \tilde{B}\tilde{r} = e'$ for approximations $\tilde{r}$ of Pv are defined by matrix elements $\tilde{b}_{tk}$ of $\tilde{B}$ of the form 1.2(27), (28),

$$(13) \quad F_t = +,-: \quad \tilde{b}_{tk} = \frac{v_k}{v_t}\frac{\partial F_t}{\partial x_k}(v); \qquad F_t = \times, /: \quad \tilde{b}_{tk} = \frac{v_k}{F_t(v)}\frac{\partial F_t}{\partial x_k}(v),$$

whereas the elements b_{tk} of the unmodified relative a posteriori error equations $r - Br = e'$, by 1.3(18), read

$$(14) \qquad b_{tk} = \frac{v_k}{v_t}\frac{\partial F_t}{\partial x_k}(v).$$

We limit the further study to the four arithmetic operations $+, -, \times, /$ and establish the theorem

(15) *The approximations r_t, $\tilde{r}_t$ of the relative a posteriori errors Pv_t satisfy, for sufficiently small $\varepsilon = \gamma\eta$, the relations*

(i) $$\tilde{r}_t = r_t + O_t(\eta^2),$$

and the associated weighted relative a posteriori condition numbers the relations

(ii) $$\tilde{\rho}_t = (1 + O_t(\eta))\rho_t, \qquad t = 1, \ldots, n.$$

Proof. (i) The two equations for r, $\tilde{r}$ immediately imply the representation

$$\tilde{r} - r = L\Delta B\tilde{r}, \quad L = (A'v)^{-1}_{\text{rel}}, \quad \Delta B = \tilde{B} - B.$$

When $F_k = +, -$, $\Delta b_{kl} = 0$, whereas for $F_k = \times, /$ we obtain

$$\Delta b_{kl} = \tilde{b}_{kl} - b_{kl} = -e'_k \tilde{b}_{kl}$$

because $F_k(v)/v_k = 1 + e'_k$. The above leads to the estimate

$$|\tilde{r}_t - r_t| \leqslant \eta' \sum_{\substack{k=0 \\ F_k = \times,/}}^{t} |L^{\text{rel}}_{tk}| \gamma_k \sum_{l=0}^{k-1} |\tilde{b}_{kl}\tilde{r}_l| \leqslant 2\eta' \rho_t \max_{j<t} |\tilde{r}_j|$$

because $|\tilde{b}_{kl}| = 1$ exactly twice and zero else for $F_k = \times, /$. Further, the last inequality yields

$$\max_{j<t} |\tilde{r}_j| \leqslant \rho'_t \eta' + 2\eta' \rho'_t \max_{j<t} |\tilde{r}_j|,$$

using the constants $\rho'_t = \max_{j<t} \rho_j$, $\eta' = \eta/(1 - \gamma\eta)$. Hence

$$\max_{j<t} |\tilde{r}_j| \leqslant \frac{\rho'_t}{1 - 2\rho'_t \eta} \eta',$$

as far as $2\rho'_t\eta' < 1$. Inserting this expression into the above inequality for $\tilde{r}_t - r_t$, we obtain

$$|\tilde{r}_t - r_t| \leqslant \frac{2\rho'_t \rho_t}{1 - 2\rho'_t \eta'} \eta'^2,$$

and thus the assertion (15i) is proved.

(ii) Next, using appropriate constants ψ_t,

$$|\tilde{r}_t| \leqslant |r_t| + \rho_t \psi_t \eta^2, \qquad |r_t| \leqslant |\tilde{r}_t| + \rho_t \psi_t \eta^2.$$

The associated weighted relative a posteriori condition numbers ρ_t, $\tilde{\rho}_t$, permit the estimate $|\tilde{r}_t| \leqslant \tilde{\rho}_t \eta'$, $|r_t| \leqslant \rho_t \eta'$, for all local errors $|e_k| \leqslant \gamma_k \eta$ where $\gamma = \max \gamma_k$ and η' as above. By means of the local errors

$$\tilde{e}_k = -\gamma_k \eta \operatorname{sgn}(\tilde{L}^{\text{rel}}_{tk}), \qquad e_k = -\gamma_k \eta \operatorname{sgn}(L^{\text{rel}}_{tk}),$$

we finally conclude

$$\frac{\tilde{\rho}_t}{1 + \gamma\eta} \leqslant \frac{\rho_t}{1 - \gamma\eta} + \rho_t \psi_t \eta, \qquad \frac{\rho_t}{1 + \gamma\eta} \leqslant \frac{\tilde{\rho}_t}{1 - \gamma\eta} + \rho_t \psi_t \eta,$$

hereby proving (15ii). □

By Theorems (12) and (15), we have established the a posteriori error representation

$$\frac{u_t - v_t}{v_t} = \tilde{r}_t + O_t(\eta^2), \tag{16}$$

where $\tilde{r}$ is the solution of the modified linear relative a posteriori error equations. Solving this relation for u_t yields

$$u_t = v_t(1 + \tilde{r}_t + O_t(\eta^2)).$$

Hence

$$v'_t = v_t(1 + \tilde{r}_t) \tag{17}$$

is an approximation of u_t having the relative error $(u_t - v_t')/v_t' = O_t(\eta^2)$. Consequently, v_t' approximates u_t within about double precision. The sequence of solutions $u_0, \ldots, u_n$ of the algorithm (A) may thus be computed in about double precision from the approximations v_t and the solutions $\tilde{r}_t$ of the modified linear a posteriori error equations, both computed in single precision. We call this procedure *extrapolation to double precision*. The a posteriori coefficients b_{tk} are readily computed from the data and intermediate results in performing the algorithm. If the local errors e_t are available numerically, the solutions $\tilde{r}_t$ of the linear a posteriori error equations and the extrapolated approximations v_t' can be computed together with v_t.

TABLE 2.2

Extrapolation to double precision in Cramer's rule

t	u_t	v_t	e_t	$-\tilde{r}_t \times 10^3$	
0	a	0.455	1.00–03	1.00	
1	b	0.111	−1.00–03	−1.00	
2	c	0.273	1.00–03	1.00	
3	d	0.778	2.86–04	0.29	
4	f	0.404	−1.00–04	−0.10	
5	g	0.566	6.07–04	0.61	
6	ad	0.354	2.82–05	1.31	
7	bc	0.0303	−9.90–05	−0.10	
8	df	0.314	−9.93–04	−0.81	
9	bg	0.0628	−4.14–04	−0.81	
10	ag	0.258	1.83–03	3.44	
11	cf	0.110	−2.65–03	−1.75	
12	D_0	0.324	9.27–04	2.36	
13	D_1	0.251	−7.96–04	−1.60	
14	D_2	0.148	0	7.29	
15	x	0.775	3.98–04	−3.56	$\frac{x' - x}{x} = 3.57\text{–}03,$
16	y	0.457	4.59–04	5.39	$\frac{y' - y}{y} = 5.40\text{–}03,$
	$x = 0.\bar{7}\ldots,$	$x' = 0.775,$	$x'' = 0.777\,769,$		$\frac{x'' - x}{x} = -1.13\text{–}05,$
	$y = 0.\overline{45}\ldots,$	$y' = 0.457,$	$y'' = 0.454\,550,$		$\frac{y'' - y}{y} = 1.00\text{–}05.$

Example. Two linear equations in two unknowns,

$$ax + by = f, \qquad cx + dy = g, \tag{18}$$

have, by Cramer's rule, the solution

$$x = \frac{D_1}{D_0}, \quad y = \frac{D_2}{D_0}, \qquad (D_0 \neq 0),$$

using the determinants

$$D_0 = \begin{vmatrix} a & b \\ c & d \end{vmatrix}, \quad D_1 = \begin{vmatrix} f & b \\ g & d \end{vmatrix}, \quad D_2 = \begin{vmatrix} a & f \\ c & g \end{vmatrix}.$$

Consider the numerical example

$$\frac{5}{11}x + \frac{1}{9}y = \frac{40}{99}, \qquad \frac{3}{11}x + \frac{7}{9}y = \frac{56}{99}, \tag{19}$$

having the solution $x = 7/9$, $y = 5/11$. Table 2.2 lists the approximations v_t in solving Cramer's rule using 3-digit decimal floating-point arithmetic, the associated local rounding errors e_t of input and arithmetic operations, and the approximations $\tilde{r}_t$ of the relative a posteriori errors described above. The numerical results show that the extrapolated results x'', y'' approximate x, y essentially within double precision.

2.3. *Associated Graphs.* In typical applications of the perturbation theory, the number n of steps and thus of equations in the systems 1.3(15), (16), (17) often becomes very large. However, the matrices (b_{tk}) of the linear error equations are *sparse*; for input operations, F_t is constant and $b_{tk} = 0$ for all k; 'built-in' functions F_t give $b_{tk} = 0$ for all $k \neq j_t$; for arithmetic operations $F_t \in$ (F2), $b_{tk} = 0$ for all $k \neq i_t, j_t$. Hence the linear error equations have the form of inhomogeneous linear recursions of zeroth, first and second order with variable coefficients. In general, however, these equations have a complex structure because z_t is not obtained from z_{t-1}, z_{t-2} but from z_{i_t}, z_{j_t}. This structure can be illustrated by the associated graph (see McCracken-Dorn [11], Bauer [5]): each t is assigned a point in $\mathbf{R}^2$, called *node*, and nodes k are connected to the node t by means of a *path* $\overrightarrow{kt}$ if the coefficient or *weight* b_{tk}, that is, $\partial F_t(u)/\partial x_k$ or $\partial F_t(v)/\partial x_k$ can be nonzero. In addition, every path $\overrightarrow{kt}$ is labelled by its associated weight b_{tk}. By convention, labels 1 may be omitted. By these means linear error equations can simply be read from the graph of the algorithm. Figure 2.3 shows the building blocks of the graphs.

The graphs in Figure 2.3 may as well be viewed as graphs of the simplest numerical algorithms analyzed in Section 1.1. For instance, let

$$u_0 = a, \quad u_1 = b, \quad u_2 = u_1 \circ u_0, \tag{1}$$

using an arithmetic operation $\circ \in \{+, -, \times, /\}$. In floating-point arithmetic, this algorithm is realized by

$$v_0 = \mathrm{fl}(a), \quad v_1 = \mathrm{fl}(b), \quad v_2 = \mathrm{fl}(v_1 \circ v_0). \tag{2}$$

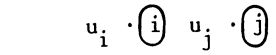
The associated linear absolute and relative a priori and a posteriori error equations and their graphs then have the form shown in Figure 2.3, where $i = 0, j = 1, t = 2$, and $r_0 = e_0 = Pa$, $r_1 = e_1 = Pb$.

Graphs constitute important means in the study of solutions of linear systems

$$z - Bz = f \Leftrightarrow z_t - \sum_{k=0}^{t-1} b_{tk} z_k = f_t, \qquad t = 0, \ldots, n, \tag{3}$$

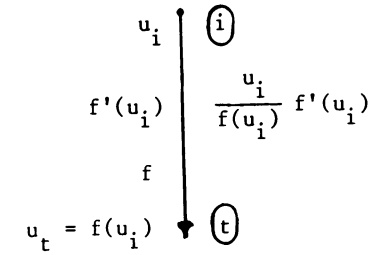
with sparse matrices, as we will explain now. The matrix B is lower triangular and its diagonal elements are zero. Thus B is nilpotent and $B^{n+1} = 0$. The solutions z of the linear system may then be represented by the finite Neumann series

$$z = (I - B)^{-1} f = \sum_{l=0}^{n} B^l f. \tag{4}$$



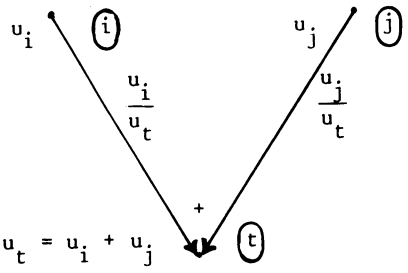
$$s_i = u_i e_i, \quad s_j = u_j e_j$$

$$r_i = e_i, \quad r_j = e_j$$



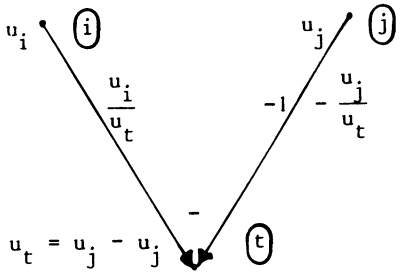
$$s_t = f'(u_i)s_i + u_t e_t$$

$$r_t = \frac{u_i}{f(u_i)} f'(u_i) r_i + e_t$$



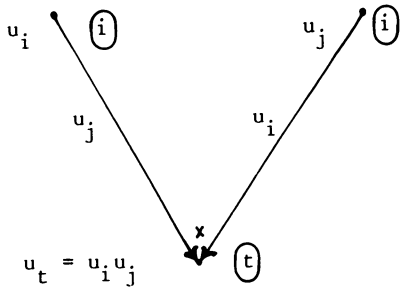
$$s_t = s_i + s_j + u_t e_t$$

$$r_t = \frac{u_i}{u_t} r_i + \frac{u_j}{u_t} r_j + e_t$$



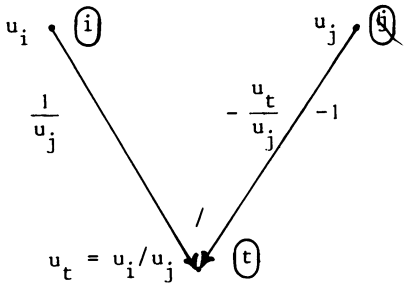
$$s_t = s_i - s_j + u_t e_t$$

$$r_t = \frac{u_i}{u_t} r_i - \frac{u_j}{u_t} r_j + e_t$$



$$s_t = u_j s_i + u_i s_j + u_t e_t$$

$$r_t = r_i + r_j + e_t$$



$$s_t = \frac{1}{u_j} s_i - \frac{u_t}{u_j} s_j + u_t e_t$$

$$r_t = r_i - r_j + e_t$$

FIGURE 2.3

Building blocks of the graphs of linear absolute (s_t, weights b_{tk}^{abs} left of the path) and relative (r_t, weights b_{tk}^{rel} right of the path) a priori error equations. The associated a posteriori error equations are obtained by replacing u_i, u_j, u_t, e_t by v_i, v_j, v_t, e'_t.

The powers of B have the elements

$$B^0 = (\delta_{ik}), \quad B^1 = (b_{ik}),$$
$$B^l = (b_{ik}^{(l)}), \quad b_{ik}^{(l)} = \sum_{k_l=0}^{n} \cdots \sum_{k_2=0}^{n} b_{ik_l} \cdots b_{k_2 k}, \qquad i, k = 0, \ldots, n, \tag{5}$$

for $l = 2, \ldots, n$. As $b_{ik} = 0$ for $i \leqslant k$, matrix elements $b_{ik}^{(l)}$ can be different from zero only if the condition

$$i > k_l > k_{l-1} > \cdots > k_2 > k_1 \geqslant 0 \tag{6}$$

holds, and therefore

$$b_{ik}^{(l)} = 0 \quad \text{for } i < k + l \tag{7}$$

and all $l = 1, 2, \ldots$. Correspondingly, the components z_t of the solution vector z of (3) possess the representation

$$z_t = L_t f = f_t + \sum_{l=1}^{t} \sum_{k=0}^{t-1} b_{ik}^{(l)} f_k, \qquad t = 0, \ldots, n. \tag{8}$$

In this way, explicit representations of the components L_t of the solution operator L have been obtained. Interchanging the order of summation gives

$$L_0 f = f_0, \quad L_t f = \sum_{k=0}^{t} L_{tk} f_k = f_t + \sum_{k=0}^{t-1} \left(\sum_{l=1}^{t} b_{ik}^{(l)} \right) f_k, \qquad t = 1, \ldots, n. \tag{9}$$

Consequently, the matrix elements of the solution operators $L = (I - B)^{-1}$ have the representation

$$L_{tt} = 1, \quad L_{tk} = \sum_{l=1}^{t} b_{ik}^{(l)}, \qquad k = 0, \ldots, t-1;$$
$$L_{tk} = 0, \qquad k = t+1, \ldots, n. \tag{10}$$

The basis of our further investigations is the explicit representation

$$z_t = L_t f = f_t + \sum_{k_1=0}^{t-1} b_{tk_1} f_{k_1} + \sum_{k_2=0}^{t-1} \sum_{k_1=0}^{t-1} b_{tk_2} b_{k_2 k_1} f_{k_1} + \cdots$$
$$+ \sum_{k_t=0}^{t-1} \cdots \sum_{k_1=0}^{t-1} b_{tk_t} \cdots b_{k_2 k_1} f_{k_1}, \tag{11}$$

ensuing immediately from (5), (8). A sequence of paths $\vec{k_1 k_2}, \ldots, \vec{k_l t}$ of the graph is a *path of length* l from the node k_1 to the node t. We call the associated product $b_{tk_l} \cdots b_{k_2 k_1}$ the *error effect* and the term $b_{tk_l} \cdots b_{k_2 k_1} f_{k_1}$ the *error contribution* along this path at the node t. The error effect is simply the product of all weights along the path. On the right side of (11), evidently, only those terms of the sum can be nonzero which belong to a path of the graph, all other terms vanish. In view of (6), necessarily $t > k_l > \cdots > k_2 > k_1$ for every path in the graph. The l-fold sum in (11) is the sum of the error contributions at the node t along all paths of length l. The node t itself is assigned the path or *loop* of length zero and the error effect 1. The representation (11) may thus be read as follows

(12) *The solution $z_t = L_t f$ of the linear error equations is the sum of the error contributions along all paths in the graph ending at the node t.*

On setting especially

$$f^{(k)} = (\delta_{0k}, \ldots, \delta_{nk}), \qquad k = 0, \ldots, n,$$

as inhomogeneous terms, the associated solutions $z_t = L_t f^{(k)}$ become the matrix elements L_{tk} of the solution operators $L = (I - B)^{-1}$. In particular,

$$(13) \qquad L_{tk} = \sum_{l=1}^{t} b_{tk}^{(l)} = b_{tk} + \sum_{k_2=0}^{t-1} b_{tk_2} b_{k_2 k} + \cdots + \sum_{k_t=0}^{t-1} \cdots \sum_{k_2=0}^{t-1} b_{tk_t} \cdots b_{k_2 k}$$

is the *total error effect* associated to all paths from the node k to the node t (see Bauer [5, p. 88]).

Now let us consider our condition numbers in this context. The weighted absolute and relative a priori and a posteriori condition numbers 2.1(17), 2.2(2) have the general form

$$(14) \qquad \lambda_t = \sum_{k=0}^{t} |L_{tk}| \alpha_k,$$

where

(15)

λ_t	absolute	relative
a priori	σ_t^i	ρ_t^i
a posteriori	σ_t^0	ρ_t^0

α_k	absolute	relative
a priori	$\lvert u_k \rvert \gamma_k$	γ_k
a posteriori	$\lvert v_k \rvert \gamma_k$	γ_k

and L_{tk} denotes the matrix elements of the associated solution operators 1.3(22), (24). The condition numbers are thus weighted sums of the absolute values of the total error effects. It is readily seen from (14) that

$$(16) \qquad \lambda_t = \max_{\substack{|f_k| \le \alpha_k \\ k=0,\ldots,n}} |L_t f|, \qquad t = 0, \ldots, n.$$

The maximum is attained for $f_k = \alpha_k \operatorname{sgn} L_{tk}$, $k = 0, \ldots, n$.

The condition numbers can be computed easily if, for instance, all matrix elements L_{tk} are nonnegative. This is the case, in particular, if the elements b_{tk} are nonnegative. Then $I - B$ is a so-called M-matrix. Under the assumption $L_{tk} \geqslant 0$,

$$(17) \qquad \lambda_t = \sum_{k=0}^{t} L_{tk} \alpha_k = L_t \alpha, \qquad t = 0, \ldots, n,$$

where $\alpha = (\alpha_0, \ldots, \alpha_n)$. That is, the vector $\lambda = (\lambda_0, \ldots, \lambda_n)$ of condition numbers is a solution of the linear system $\lambda - B\lambda = \alpha$ and, consequently, can be obtained *recursively* from

$$(18) \qquad \lambda_0 = \alpha_0, \quad \lambda_t = \sum_{k=0}^{t-1} b_{tk} \lambda_k + \alpha_t, \qquad t = 1, \ldots, n.$$

In the general case, where the weights b_{tk} have arbitrary signs, at least bounds can be determined for the solutions $z_t = L_t f$ and thus for the corresponding condition numbers λ_t. The above results immediately yield the theorem:

(19) *The recursion*

$$(\mathrm{i}) \qquad \mu_0 = \alpha_0, \quad \mu_t = \sum_{k=0}^{t-1} |b_{tk}| \mu_k + \alpha_t, \qquad t = 1, \ldots, n,$$

generates a sequence (μ_t) *of bounds having the property*

(ii) $$|z_t| = |L_t f| \leqslant \mu_t, \qquad \lambda_t \leqslant \mu_t,$$

for all $f = (f_0, \ldots, f_n)$ *such that* $|f_k| \leqslant \alpha_k$, $k = 0, \ldots, n$.

When an algorithm is built up of input operations and additions only, the coefficients b_{tk} of the linear absolute error equations consist, according to Table 1.2, of zeros and ones, that is, B is a *binary matrix*. The same is true for the linear relative error equations when the algorithm consists of input operations and multiplications only. For a binary matrix B, one readily sees from (13) that

(20) $$L_{tk} = \text{Number of paths from } k \text{ to } t.$$

The explicit representations of solution operators and associated condition numbers become very simple when the graph of the algorithm is a *tree*. In these graphs there exists to each $k < t$ exactly one path $\overrightarrow{k_1 k_2}, \ldots, \overrightarrow{k_s t}$ from $k = k_1$ to t, where the length s of the path depends on the pair k, t. Now the total error effects are simply

(21) $$L_{tt} = 1, \qquad L_{tk} = \sum_{l=1}^{t} b_{tk}^{(l)} = b_{tk_s} \cdots b_{k_2 k},$$

and the associated condition numbers have the form

(22) $$\lambda_0 = \alpha_0, \qquad \lambda_t = \sum_{k=0}^{t-1} |b_{tk_s}| \cdots |b_{k_2 k}| \alpha_k + \alpha_t.$$

In this case, the sequence (λ_t) is the solution of the *recursion formulae*

(23) $$\lambda_0 = \alpha_0, \quad \lambda_t = \sum_{k=0}^{t-1} |b_{tk}| \lambda_k + \alpha_k, \qquad t = 1, \ldots, n.$$

For, this system has the form of the linear error equations with coefficients $|b_{tk}|$ and inhomogeneous terms α_t instead of b_{tk}, f_t. According to (11), (21), the solution of the linear system (23) has just the representation (22) because the associated graph is identical with the graph of the linear error equations of the algorithm and thus also a tree.

Note the interesting fact that the condition numbers (18), (23) are obtained in the same way as those of the elementary operations in Section 1.1. For example, on setting in 1.1(19)

$$a = u_i, \quad b = u_j, \quad u = u_t, \quad \gamma = \gamma_t,$$

and using b_{tk}^{abs}, b_{tk}^{rel} from Table 1.2, the condition numbers $\lambda_t = \sigma_t^i$, ρ_t^i in (18), (23) are specified as σ, ρ in 1.1(19). The a posteriori condition numbers $\lambda_t = \sigma_t^0$, ρ_t^0 are obtained as σ, ρ in 1.1(21). Also the bounds (19) are computed in this way.

When the graph of the algorithm is a tree and, additionally, B a binary matrix, there exists for each node $k < t$ exactly one path to the node t and the associated error effects are equal to 1. From (21) we then infer

(24) $$L_{tk} = 1, \qquad k \leqslant t,$$

and from (22) the condition numbers

(25) $$\lambda_t = \sum_{k=0}^{t} \alpha_k, \qquad t = 0, \ldots, n.$$

In view of the above, the numerical stability of evaluation algorithms for the condition numbers (18), (23) and the bounds (19) becomes of interest. The following, last theorem serves to answer this question. Note that i_t, j_t denote the indices of the two operands in the arithmetic operations $F_t(u) = u_{i_t} \circ u_{j_t}$.

(26) *Let the algorithm* (A) *possess the following properties*: (a) *all input data, intermediate, and final results* $u_0, \ldots, u_n$ *are positive numbers*; (b) *the algorithm consists only of input operations, additions, multiplications, and divisions. Then the recursion*

$$\text{(i)} \qquad \begin{aligned} \tau_t &= \gamma_t, && F_t \in (\mathrm{F0}), \\ \tau_0 &= \gamma_0, \quad \tau_t = \max(\tau_{i_t}, \tau_{j_t}) + \gamma_t, && F_t = +, \\ \tau_t &= \tau_{i_t} + \tau_{j_t} + \gamma_t, && F_t = \times, /, \end{aligned}$$

determines a sequence of bounds for the relative a priori condition numbers of the algorithm, that is,

$$\text{(ii)} \qquad \rho_t \leqslant \tau_t, \qquad t = 0, \ldots, n.$$

Proof. The proposition is proved by finite induction with respect to t. For $t = 0$, $\rho_0 = \gamma_0 = \tau_0$. Now assume that $\rho_k \leqslant \tau_k$ for $k = 0, \ldots, t-1$. When F_t is an input operation, $\rho_t = \gamma_t = \tau_t$. When F_t is an addition, the solution of the linear relative a priori error equations satisfies the estimate

$$|r_t| \leqslant \left|\frac{u_i}{u_t}\right||r_i| + \left|\frac{u_j}{u_t}\right||r_j| + \gamma_t\eta, \qquad i = i_t, j = j_t.$$

By assumption, $|r_i| \leqslant \tau_i\eta$, $|r_j| \leqslant \tau_j\eta$. As u_i, u_j, u_t are positive, $|u_i| + |u_j| = |u_t|$, so that

$$\frac{1}{\eta}|r_t| \leqslant \left|\frac{u_i}{u_t}\right|\tau_i + \left|\frac{u_j}{u_t}\right|\tau_j + \gamma_t \leqslant \max(\tau_i, \tau_j) + \gamma_t = \tau_t.$$

Finally, $F_t = \times, /$ leads to the estimate

$$\frac{1}{\eta}|r_t| \leqslant |r_i| + |r_j| + \gamma_t \leqslant \tau_i + \tau_j + \gamma_t = \tau_t.$$

In this way, $|r_t| \leqslant \tau_t\eta$ for all $|e_k| \leqslant \gamma_k\eta$, $k = 0, \ldots, n$, and by induction for all $t = 0, \ldots, n$. In addition, from (16) for $f_t = e_t/\eta$, $L_t^{\mathrm{rel}} f = r_t/\eta$ one concludes $\lambda_t = \rho_t \leqslant \tau_t$. □

Having determined the above sequence of bounds (τ_t), to each addition $F_t = +$ an index $m_t \in \{i_t, j_t\}$ can be assigned such that

$$\text{(27)} \qquad \tau_{m_t} = \max(\tau_{i_t}, \tau_{j_t}).$$

Let us further introduce the matrix (β_{tk}) with the elements

$$\text{(28)} \qquad \begin{aligned} \beta_{tk} &= 0, && F_t \in (\mathrm{F0}), \\ \beta_{tk} &= \delta_{km_t}, && F_t = +, \\ \beta_{tk} &= \delta_{ki_t} + \delta_{kj_t}, && F_t = \times, /, \end{aligned}$$

for $k, t = 0, \ldots, n$. Obviously, (β_{tk}) is a binary matrix and the sequence of bounds (τ_t) is the solution of the linear system

$$\text{(29)} \qquad \tau_t - \sum_{k=0}^{t-1} \beta_{tk}\tau_k = \gamma_t, \qquad t = 0, \ldots, n.$$

The graph of this system is obtained from the graph of the associated algorithm (A), named in Theorem (26), if to each node t such that $F_t = +$ of the two paths $\vec{i_t}t$, $\vec{j_t}t$ only $\vec{m_t}t$ is kept and the other deleted. Denote by N_{tk} the number of paths in this *reduced graph* from k to t. Then, by (20), the following estimate and representation is established

$$\rho_t \leqslant \gamma_t + \sum_{k=0}^{t-1} N_{tk}\gamma_k = \tau_t, \qquad t = 0, \ldots, n. \tag{30}$$

3. A Survey of Examples and Applications. This section briefly surveys some typical examples and applications of the above error analysis.

3.1. The paper [18], in particular, studies the class of elementary one-step algorithms

$$u_0 = a, \quad u_t = b_t o_t u_{t-1} + c_t, \qquad t = 1, \ldots, n, \tag{1}$$

where $o_t \in \{Nop, \times, /, \backslash\}$. Special cases are the well-known algorithms for computing partial products ($o_t = \times$, $a = b_0$, $c_t = 0$), partial sums ($o_t = Nop$, $a = c_0$, $b_t = 1$), solutions of inhomogeneous bidiagonal systems of linear equations ($o_t = \times$), Horner's scheme for the evaluation of Taylor polynomials ($o_t = \times$, $a = c_0$, $b_t = z$), Newton polynomials ($o_t = \times$, $a = c_0$, $b_t = z - z_{n-t}$) and finite continued fractions ($o_t = /$). The graph of the algorithm (1) is a tree so that by virtue of 2.3(23) the condition numbers can be determined recursively. For instance, the relative a priori condition numbers ρ_t of computing u_t from (1) satisfy the recursion

$$\rho_0 = \gamma_0^a, \quad \rho_t = \left|\frac{b_t o_t u_{t-1}}{u_t}\right|\left(\rho_{t-1} + \gamma_t^b + \gamma_t^0\right) + \left|\frac{c_t}{u_t}\right|\gamma_t^c + \gamma_t^+ \tag{2}$$

for $t = 1, \ldots, n$. In addition, the paper [18] analyzes some further elementary algorithms and contains many numerical examples illustrating the error analysis.

3.2. The papers [16, Sections 4.3, 4.4], [19] deal with difference schemes

$$u_k^0 = u_k \quad (i = 0), \qquad u_k^i = u_k^{i-1} - u_{k-1}^{i-1}, \tag{1}$$

and the Neville-Aitken algorithm

$$u_k^0 = u_k \quad (i = 0), \qquad u_k^i = \frac{x - x_{k-i}}{x_k - x_{k-i}} u_k^{i-1} - \frac{x - x_k}{x_k - x_{k-i}} u_{k-1}^{i-1}, \tag{2}$$

$k = i, \ldots, m$, $i = 1, \ldots, m$. The graphs of these algorithms are no longer tree-like but constitute an important tool in deriving the linear error equations and associated condition numbers. It is shown that also for the above algorithms the condition numbers can be obtained from simple recursion formulae. For difference schemes of a smooth function on equidistant meshes, using the condition numbers, a lower bound for the step-widths is obtained, called "critical step-width", which guarantees that at least the leading digit of the mth order difference is significant. It is proved that Romberg extrapolations ($x_k = x_0/4^k$) are strongly stable. Further the condition numbers of the extrapolation algorithm (2) are determined for the systems of nodal points $x_k = x_0/(k+1)^2$, $x_k = x_0/(k+1)$ and applied to numerical examples.

3.3. The paper [17] establishes the error analysis of numerically solving two linear equations in two unknowns,

$$ax + by = f, \qquad cx + dy = g. \tag{1}$$

The relative data and rounding condition numbers of computing the solutions x, y by Cramer's rule and Gaussian elimination are determined. Two important results of the paper read: for a nonsingular linear system Cramer's rule is always well-conditioned or backward stable,

$$\frac{\rho_x^R}{\rho_x^D} \leqslant 2.5, \qquad \frac{\rho_y^R}{\rho_y^D} \leqslant 2.5; \tag{2}$$

Gaussian elimination is backward stable, and

$$\frac{\rho_x^R}{\rho_x^D} \leqslant 2.75, \qquad \frac{\rho_y^R}{\rho_y^D} \leqslant 2, \tag{3}$$

provided that the system is properly pivoted such that $|bc| \leqslant |ad|$. The algorithms are analyzed further with respect to the behavior of the residuals of the computed solutions. It is shown that Gaussian elimination is, additionally, well-conditioned in this sense whereas Cramer's rule is not. It is proved that the relative condition numbers, the stability constants (2), (3), and the above pivotal strategy are invariant under scaling of the linear system.

3.4. In [20], [21], [22] the forward error analysis of Gaussian elimination and two-sided elimination of tridiagonal linear systems is presented. Both explicit representations and recursions of the absolute a priori data and rounding condition numbers σ_i^D, σ_i^R of the solutions x_i, $i = 1, \ldots, n$, are derived. In addition, residual condition numbers τ_j^D, τ_j^R, $j = 1, \ldots, n$, are determined. When the tridiagonal coefficient matrix is an M-matrix or positive definite, Theorem [21, 2.3(21)] ensures both the backward stability and the residual stability of Gaussian elimination without pivoting for computing the solutions x_i and proves the stability estimates

$$\frac{\sigma_i^R}{\sigma_i^D} \leqslant 4, \qquad \frac{\tau_j^R}{\tau_j^D} \leqslant 4, \qquad i, j = 1, \ldots, n. \tag{1}$$

The paper [22] contains the corresponding forward error analysis of two-sided elimination. For every two-sided elimination-regular tridiagonal linear system, the computation of the solutions x_i by two-sided elimination is well-conditioned or backward stable and

$$\frac{\sigma_i^R}{\sigma_i^D} \leqslant 5.5, \qquad i = 1, \ldots, n. \tag{2}$$

Babuška has proved the estimate $\sigma_i^R/\sigma_i^D < 9$ in [2], using a backward error analysis. For the case $n = 3$, Miller [12] has shown numerically that σ_2^R/σ_2^D is about 5.5 so that our estimate (2) seems to be sharp for $n \geqslant 3$. The general results of the forward error analysis are tested and illustrated by numerical examples in [20], [21], [22].

3.4. A forthcoming paper develops the error analysis of Gaussian elimination

$$(1) \qquad a_{ik}^1 = a_{ik}, \qquad a_{ik}^{t+1} = a_{ik}^t - \frac{a_{it}^t a_{tk}^t}{a_{tt}^t},$$

$i = t+1, \ldots, m$, $k = t+1, \ldots, n$, $t = 1, \ldots, m-1$, for arbitrary rectangular matrices $A = (a_{ik})_{i=1,\ldots,m;\, k=1,\ldots,n}$ such that $a_{tt}^t \neq 0$, $t = 1, \ldots, m-1$. The associated system of linear absolute a priori error equations reads

$$(2) \qquad s_{ik}^1 = f_{ik}^1, \qquad s_{ik}^{t+1} = s_{ik}^t + p_i^{t+1} s_{tk}^t + q_k^{t+1} s_{it}^t + p_i^{t+1} q_k^{t+1} s_{tt}^t + f_{ik}^{t+1},$$

where

$$p_i^{t+1} = -\frac{a_{it}^t}{a_{tt}^t}, \qquad q_k^{t+1} = -\frac{a_{tk}^t}{a_{tt}^t},$$

f_{ik}^1 are the absolute errors of the coefficients a_{ik}, and

$$f_{ik}^t = a_{ik}^t e_{tik}^{-} - p_i^t q_k^t a_{t-1,t-1}^{t-1}(e_{tik}^{\times} + e_{tik}^{\cdot/\cdot}), \qquad t = 2, \ldots, m.$$

An explicit representation of the solution s_{ik}^t in terms of the f_{ik}^t is obtained that immediately yields associated condition numbers σ_{ik}^t.

The error analysis of this important algorithm is applied to forward elimination of a system of m linear equations with $n - m$ right-hand sides, back substitution or solving a triangular system of linear equations, solving an inhomogeneous linear system, and computing the inverse of a nonsingular square matrix.

This forward error analysis differs significantly from Wilkinson's backward error analysis. Forward error analysis compares the numerical results obtained under data perturbations or rounding errors of a floating-point arithmetic directly with the exact results and establishes optimal bounds of the possible errors, whereas backward error analysis primarily estimates the residuals of approximate solutions and subsequently obtains error estimates of solution vectors in suitable norms by means of condition numbers of the coefficient matrix. This procedure may overestimate the actual error considerably. Moreover, the forward error analysis uses a 'finer topology': the error estimates bound the components of the error vectors and do not use norms; the condition numbers and stability constants depend on the solutions and thus yield pointwise, not uniform, estimates. Finally, our stability constants and relative data and rounding condition numbers are invariant with respect to scaling of the linear system.

Acknowledgement. The author acknowledges valuable comments of the referees.

Fachbereich Mathematik
University of Frankfurt
Robert-Mayer Strasse 10
D-6000 Frankfurt, West Germany

1. I. Babuška, *Numerical Stability in Numerical Analysis*, Proc. IFIP-Congress 1968, Amsterdam, North-Holland, Amsterdam, 1969, pp. 11–23.

2. I. Babuška, "Numerical stability in problems of linear algebra," *SIAM J. Numer. Anal.*, v. 9, 1972, pp. 53–77.

3. I. Babuška, M. Prager & E. Vitasek, *Numerical Processes in Differential Equations*, Wiley, New York, 1966.

4. F. L. Bauer, "Genauigkeitsfragen bei der Lösung linearer Gleichungssysteme," *Z. Angew. Math. Mech.*, v. 46, 1966, pp. 409–421.

5. F. L. BAUER, "Computational graphs and rounding error," *SIAM J. Numer. Anal.*, v. 11, 1974, pp. 87–96.

6. F. L. BAUER ET AL., *Moderne Rechenanlagen*, Chap. 3, Teubner, Stuttgart, 1964.

7. W. S. BROWN, *A Realistic Model of Floating-Point Computation*, Proc. Sympos. Mathematical Software III, Madison, 1977 (J. R. Rice, Ed.), Academic Press, New York, 1977, pp. 343–360.

8. P. HENRICI, *Elements of Numerical Analysis*, Chap. 16, Wiley, New York, 1964.

9. J. LARSON & A. SAMEH, "Efficient calculation of the effects of roundoff errors," *ACM Trans. Math. Software*, v. 4, 1978, pp. 228–236.

10. S. LINNAINMAA, "Taylor expansion of the accumulated rounding error," *BIT*, v. 16, 1976, pp. 146–160.

11. D. D. MCCRACKEN & W. S. DORN, *Numerical Methods and Fortran Programming*, Wiley, New York, 1964.

12. W. MILLER, "Software for roundoff analysis," *ACM Trans. Math. Software*, v. 1, 1975, pp. 108–128.

13. W. MILLER, "Computer search for numerical instability," *J. Assoc. Comput. Mach.*, v. 22, 1975, pp. 512–521.

14. W. MILLER & D. SPOONER, "Software for roundoff analysis, II," *ACM Trans. Math. Software*, v. 4, 1978, pp. 369–387.

15. J. R. RICE, "A theory of condition," *SIAM J. Numer. Anal.*, v. 3, 1966, pp. 287–310.

16. F. STUMMEL, *Fehleranalyse numerischer Algorithmen*, Lecture Notes, Univ. of Frankfurt, 1978.

17. F. STUMMEL, "Rounding errors in numerical solutions of two linear equations in two unknowns," Preprint, 1980.

18. F. STUMMEL, *Rounding Error Analysis of Elementary Numerical Algorithms*, Proc. Conf. Fundamentals of Numerical Computation, Berlin 1979 (G. Alefeld & R. D. Grigorieff, Eds.), *Computing*, Suppl. 2, 1980, pp. 169–195.

19. F. STUMMEL, "Rounding error anlaysis of difference and extrapolation schemes." (To appear.)

20. F. STUMMEL, *Rounding Error in Gaussian Elimination of Tridiagonal Linear Systems, Survey of Results*, Proc. Internat. Sympos. 'Interval Mathematics 1980', Freiburg (K. Nickel, Ed.), Academic Press, New York, 1980, pp. 223–245.

21. F. STUMMEL, "Rounding error in Gaussian elimination of tridiagonal linear systems. I," *SIAM J. Numer. Anal.* (Submitted.)

22. F. STUMMEL, "Rounding error in Gaussian elimination of tridiagonal linear systems. II," *Linear Algebra Appl.* (Submitted.)

23. F. STUMMEL, "Forward error analysis of Gaussian elimination." (To appear.)

24. M. TIENARI, "On some topological properties of numerical algorithms," *BIT*, v. 12, 1972, pp. 409–433.

25. J. H. WILKINSON, *Rounding Errors in Algebraic Processes*, Prentice-Hall, Englewood Cliffs, N.J., 1963.