Aplikace matematiky

Emil Vitásek

Stability of numerical processes

Aplikace matematiky, Vol. 10 (1965), No. 2, 130–145

Persistent URL: http://dml.cz/dmlcz/102942

STABILITY OF NUMERICAL PROCESSES

EMIL VITÁSEK

This report deals with some questions of the stability of numerical processes. Problems of this kind have been brought to the foreground in recent years, namely in connection with work on high speed computers which allow a large number of arithmetic operations to be performed in a very short time. Before giving the precise definition of the concept of a numerical process and its stability, some simple examples will be given, exhibiting the features of numerical practice leading to these problems.

Example 1. Compute the integrals

$$I_n = \frac{1}{e}\int_0^1 x^n e^x \,\mathrm{d}x \tag{1}$$

for $n = 0, 1, \ldots$ by integration by parts. The reccurence relation

$$I_{n+1} = 1 - (n+1)I_n, \quad n = 0, 1, \ldots, \quad I_0 = 1 - \frac{1}{e} \tag{2}$$

is found readily and also the estimates $0 < I_n < I_{n-1}$, $I_n \to \infty$ for $n \to \infty$. Table 1 shows the results obtained on different computers. It is seen that the results are completely unacceptable for $n \geqq 12$.

Example 2. Compute y_n and z_n given by

$$y_n = \frac{z_n}{n}, \quad z_{n+1} = n y_n \quad \text{for} \quad n = 1, 2, \ldots, \quad z_1 = 1\,. \tag{3}$$

Obviously, y_n has the exact value $1/n$. The results obtained on different computers are shown in table 2.

The sequence (3) can be obviously rewritten in the following form:

$$(((((1 : 2) \,.\, 2) : 3) \,.\, 3) : 4) \ldots \tag{4}$$

Rearranging brackets in (4) as follows,

$$(((1 \,.\, \tfrac{1}{2}) \,.\, \tfrac{2}{3}) \,.\, \tfrac{3}{4}) \ldots \tag{5}$$

Table 1.

Evaluation of $I_n = \frac{1}{e}\int_0^1 x^n e^x \, dx$ *by the recurrence formula* (2)

	Floating Decimal Point		Fixed Decimal Point	
Computer	ZUSE 23	D 2	LGP 30[1])	D 1
Computing Centre:[2])	SVUTT Prague	TU Dresden	VUT Brno	TU Dresden
$n =$ 0	0·632120559	0·632120559000	0·63	0·6321205590000
1	0·367879441	0·367879441200	0·37	0·3678794412000
2	0·264241118	0·264241117610	0·25	0·2648411176000
3	0·207276646	0·207276647177	0·00	0·2072766472000
4	0·170893415	0·170893411296	0·99	0·1708934112000
5	0·145532921	0·145532943523	0·00	0·1455329439999
6	0·126802474	0·126802338862	0·99	0·1268023360000
7	0·112382680	0·112383627969	−2·00	0·1123836479974
8	0·100938558	0·100930976248	8·99	0·1009308160205
9	0·0915529728	0·0916212137705	−43·99	0·0916226558151
10	0·0844702721	0·0837878622985	45·00	0·0837734418483
11	0·0708270073	0·0783335147182	−88·99	0·0784921396689
12	0·150075912	0·0599978233837	267·99	0·0580943239729
13	−0·950986862	0·220028296015	−1071·00	0·2447737883521
14	14·3138160	−2·08039614421	5355·99	−2·4268330369304

[1]) This fixed decimal point calculation has been performed intentionally to a small number of decimal places.

[2]) Abbreviations:
SVUTT: State Research Institute for Heating Techniques
VUT: Technical University
TH: Technical University
TU: Technical University
KU: Charles University
UTIA: Institute for the Theory of Information and Automation

we obtain for y_n the reccurence relation

$$y_{n+1} = v_n y_n, \quad v_n = \frac{n}{n+1}, \quad n = 1, 2, \ldots, \quad y_1 = 1. \tag{6}$$

Table 3 shows the results obtained in this case.

Finally consider the following example:

Table 2.

Evaluation of $y_{10000} \cdot 10^4$ *from* (3)

Floating Decimal Point			Comments
Computer	Computing Centre[3])	$y_{10000} \cdot 10^4$	
ZUSE 23	SVUTT Prague	0·999993645	
LGP 30	VUT Brno	0·9992631	Machine language
LGP 30	KU Prague	0·9995524	automatic coding
LGP 30	KU Prague	1·001660	special subroutine
D 2	TU Dresden	1·00000000002	
X 1	TH Braunschweig	1·00000000000	12 dec. plac., aut. coding
X 1	TH Braunschweig	0·99996060	8 dec. plac., aut. coding
SIEMENS 2002	TH Mainz	0·999999761	
ER 56	TH Stuttgart	0·99999972512	

Fixed Decimal Point		
Computer	Computing Centre[3])	$y_{10000} \cdot 10^4$
LGP 30	KU Prague	0·84878
D 1	TU Dresden	0·99999999970654
Ural 1	UTIA Prague	0·99997

[3]) The abbreviations are explained in Table 1.

Example 3. Compute r_n from the reccurence relation

$$r_{n+1} = \frac{1}{n} r_n + 1\,, \quad n = 1, 2, \ldots, \quad r_1 = 1\,. \tag{7}$$

The results are in table 4.

If we compare the results of these three examples we see that in example 1, a striking phenomenon occurrs which causes all computations to break down. This is, intuitively the phenomenon of numerical instability. In this case, without respect to the type of a computer employed, we obtain fully unacceptable results even for not too great n.

Example 2 still displays a certain loss of decimal places but this phenomenon is definitely less pronouced than in the previous example. It may be seen that the type

Table 3.

Evaluation of $y_{10000} \cdot 10^4$ *from* (6).

Floating Decimal Point			Comments
Computer	Computing Centre	$y_{10000} \cdot 10^4$	
ZUSE 23	SVUTT Prague	0·999991849	
LGP 30	VUT Brno	0·9992850	Machine language
LGP 30	KU Prague	0·9995754	automatic coding
LGP 30	KU Prague	1·001661	special subroutine
D 2	TU Dresden	0·999999999316	
X 1	TH Braunschweig	0·999999999990	12 dec. plac., aut. coding
X 1	TH Braunschweig	0·99988480	8 dec. plac., aut. coding
SIEMENS 2002	TH Mainz	0·9999999745	
ER 56	TH Stuttgart	1·0009004836	

Fixed Decimal Point		
Computer	Computing Centre	$y_{10000} \cdot 10^4$
D 1	TU Dresden	0·9999999998
LGP 30	KU Prague	0·97692
X 1	TH Braunschweig	0·67345081

of computer, the machine program, the fact whether fixed or floating point arithmetic is employed, all exercise a strong influence on the results.

Finally the last example illustrates the type of computation where one obtains very exact results independently of the type of computer and of the machine program. Such behaviour we require intuitively from the numerical process which we call stable.

After showing on simple examples some typical possibilities of behaviour of numerical processes we devote our interest to the exact definition of numerical process and its stability. The concept of numerical stability has been introduced in order to enable us to study the influence of the fact that, in actual cases, it is impossible to operate with exact numbers. Thus, what shall we understand by a numerical process? A numerical process in the proper sense is a succession of elementary arithmetic operations on numbers. However, in order to simplify our study, it will be

convenient to generalize this process and to work with more general objects such as vectors, matrices or, more generally, elements of normed spaces. Hence, we introduce the following definition.

Table 4.

Evaluation of r_{10000} from (7).

Floating Decimal Point		
Computer	Computing Centre	r_{10000}
ZUSE 23	SVUTT Prague	1·00010002
LGP 30	VUT Brno	1·000100
D 2	TU Dresden	1·00010002008

Fixed Decimal Point		
Computer	Computing Centre	r_{10000}
LGP 30	VUT Brno	1·00010002
D 1	TU Dresden	1·0001000201 0378

Definition 1. *Let X_i, $i = 1, 2, \ldots, N \leqq \infty$ be a sequence of normed vector spaces (which in general have different dimensions) and A_i a sequence of continuous operators such that A_i is defined on the cartesian product $X_1 \times X_2 \times \ldots \times X_i$, $i = 1, 2, \ldots N - 1$ and maps this product into X_{i+1}. Further more, let there be given an element $x_1 \in X_1$. Then the sequence of equations*

$$x_{i+1} = A_i(x_1, x_2, \ldots, x_i), \quad i = 1, 2, \ldots, N - 1 \tag{8}$$

will be called a numerical process and the sequence of elements $x_1, x_2, \ldots, x_N$ its solution.

In practical computations each of the spaces X_i is obviously the space of real numbers and A_i is the operator of an elementary arithmetic operation, which will never depend on all previously computed numbers. Furthermore the sequence (8) in Definition 1 is always finite. The number of equations (8) may be given in advance or may be determined on the basis of a criterion imposed on the computation. For instance, in the case of iterations, the process may be terminated when the difference $x_{i+1} - x_i$ is small in some prescribed manner. In such a case it will be convenient

to represent the actual numerical process by an infinite sequence. When employing high speed computers the number N is usually so large that the actual stiuation is better described by an infinite sequence than by a finite one.

In practice, computations cannot be carried out in accordance with our definition since during the evaluation of each of the equations (8) we will commit a certain error and the succeeding computation is then based on this incorrect result. Thus, instead of the elements x_i we compute elements $\tilde{x}_i$ from

$$\tilde{x}_{i+1} = A_i(\tilde{x}_1, \tilde{x}_2, \ldots, \tilde{x}_i) + \delta_i\,, \quad \delta_i \in X_{i+1}\,, \tag{9}$$

where δ_i is, in general, an element with small norm. In an actual numerical process, δ_i depends on several circumstances, and in particular on $\tilde{x}_j$, $j = 1, 2, \ldots, i$, on the method of round-off, on the actual machine program etc. Since our objective is the discovery of the general laws of a process, we will only assume here that the elements δ_i are small. Because we want to estimate the difference $x_i - \tilde{x}_i$, in particular for large values of i, the norm used for the estimates of $x_i - \tilde{x}_i$ and δ_i depends, of course, on the character and purpose of the results.

Obviously, we must require δ_i to be small in order that $x_i - \tilde{x}_i$ be small for all i. However, even if δ_i are small, this will not ensure that $x_i - \tilde{x}_i$ are small for all i. In fact, it is only for stable processes that $x_i - \tilde{x}_i$ is small more or less independently of i, for small δ_i.

Here attention should be drawn to a very important circumstance, namely whether we can, in practice, realize sufficiently small δ_i. It is obvious fact that a computation does not proceed in the manner that $A_i(\tilde{x}_1, \tilde{x}_2, \ldots, \tilde{x}_i)$ is first evaluated exactly and then the error δ_i added, but the additional error arises from the inaccurrate execution of the operation A_i. Thus, if we desire to make use of the model corresponding to Definition 1, we must consider the problem of its realisation on a computer.

Now we have prepared the intuitive justification for the definition of a stable numerical process.

Definition 2. *Let there be given a numerical process in the sense of Definition* 1, *and let its solution be identically zero. This trivial solution will be said to be numerically stable, if for every* $\varepsilon > 0$ *there exists a* $\delta > 0$ *such that, if* $|\tilde{x}_1|_1 < \delta$, $|\delta_i|_{i+1} < \delta$ *and the elements* $\tilde{x}_i$ *satisfy* (9), *then* $|\tilde{x}_i|_i < \varepsilon$ *for every* $i = 1, 2, \ldots, N$ ($|\ |_i$ *denotes the norm in the space* X_i).

The concept of stability is defined only for a solution and not for the whole numerical process. In fact, a solution is determined by the initial element x_1, so that if we start from a different initial value, the solution may have completely different character. Since the structure of the operators A_i controls this behaviour, we must base our study of stability on a single solution. The restriction to the trivial solution is only formal in order to simplify the formulation.

Till now, we have considered a numerical process and ist behaviour as an independent unity, and disregarded its origin and purpose. Now consider the following example.

Let it be required to solve a boundary value problem for a differential equation by the Ritz method. Then, if we want to increase the accurracy, we must also increase the number of coordinate functions and thus to solve larger and larger systems of simultaneous linear equations. By changing the number of coordinate functions, the original numerical process is transformed into a new similar one, and it would be conveninent for practical computation if the stability of these processes is independent, if possible, of the number of equations. Analogous situations occur in most numerical methods of mathematical analysis when the approximate solution of a continuous problem is approched by approximating the original problem by a finite dimensional problem.

Thus, it is seen to be very convenient to study systems of numerical processes in dependence on an arbitrarily chosen parameter, and to describe the dependence of the stability on this parameter. For this reason we now introduce the following definition:

Definition 3. *Let there be given a sequence of numerical processes*

$$x_{i+1}^{(j)} = A_i^{(j)}(x_1^{(j)}, x_2^{(j)}, \ldots, x_i^{(j)}), \tag{10}$$

$$i = 1, 2, \ldots, N(j) - 1, \quad N(j) \leqq \infty,$$

$$j = 1, 2, \ldots$$

Let each of these processes have a stable trivial solution. The sequence will be said to be an α_k-sequence $(k \geqq 0)$ of numerical processes for the trivial solution, if for every $\varepsilon > 0$ there exists a $\delta > 0$ (independent of j) such that for all $\tilde{x}_i^{(j)}$ with

$$\tilde{x}_{i+1}^{(j)} = A_i^{(j)}(\tilde{x}_1^{(j)}, \tilde{x}_2^{(j)}, \ldots, \tilde{x}_i^{(j)}) + \delta_i^{(j)} \tag{11}$$

there is $|\tilde{x}_i^{(j)}|_i^{(j)} < \varepsilon$ for every i, j, provided that $|\tilde{x}_1^{(j)}|_1^{(j)} < j^{-k}\delta$ and $|\delta_i^{(j)}|_{i+1}^{(j)} < j^{-k}\delta$. If, moreover, there exist $\varepsilon_0 > 0$ and Q which are independent on j such that we can take $\delta = Q \min(\varepsilon, \varepsilon_0)$, then the sequence under consideration will be said to be an α_k - L-sequence of numerical processes.

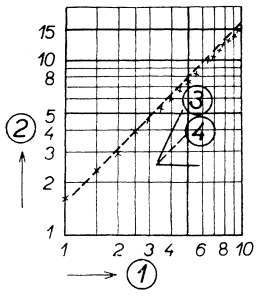

Fig. 1. Error trend for the process (3) in fixed point
1 ... $n/1000$
2 ... $\delta z_n \times 10^2$, computation on LGP 30
3 ... slope of the theoretical α_2-L-sequence
4 ... actual slope corresponding to α_1-L-sequence

In practice, almost all numerical processes are of the α_k - L-type. In essence, Definition 3 expresses the fact that as we increase the value of j we must also increase the accurracy of the computations proportionally to j^k in order to obtain the result

with the same accuracy. Since, in general, the length of a machine word is given, it will be rather difficult to increase the accuracy in actual computations. However, in the case of an α_k - L-sequence of numerical processes and small errors, we can interpret the above definition by saying that for constant error δ in the elementary operation, the loss of accurracy in the result is at most proportional to j^k.

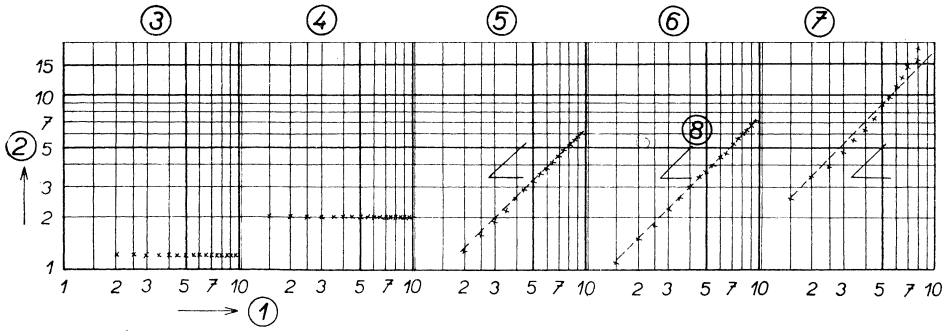

Fig. 2. Error trend for the process (3) in floating point

1 ... $n/1000$
2 ... error δz_n
3 ... $\delta z_n \times 10^6$, computation on SIRIUS
4 ... $\delta z_n \times 10^4$, computation on D 2
5 ... $\delta z_n \times 10^6$, computation on ZUSE 23
6 ... $\delta z_n \times 10^4$, computation on LGP 30
7 ... $\delta z_n \times 10^8$, computation on ER 56
8 ... slope of theoretical α_1-L-sequence

Before pointing out some possibilities of utilisation of the practical significance of the concept of α_k-sequences, some remarks will be made concerning the stability, in our sense, of the examples dealt with above.

First investigate the integrals (1). Because we are interested in the character of the loss of accurracy in dependence on the length of the process, we will study the reccurence relation (2) as a sequence of numerical processes:

$$I_{i+1}^{(j)} = 1 - (i+1) I_i^{(j)}, \quad i = 1, 2, \ldots, j-1, \quad j = 1, 2, \ldots \tag{12}$$

It is seen easily that the sequence (12) is not an α_k-sequence for any value of k. This conclusion agrees very well with the experiments because we have seen that the reccurence relation (12) is completely worthless for practical computation.

Let us study in similar manner the stability of the numerical processes in the other examples. It can be shown that the computation of the sequence (4) is an α_2-sequence when computing in fixed point arithmetic, and an α_1-sequence when using floating point airthmetic; and the computation of the sequence (5) is an α_1-sequence in fixed point arithmetic and an α_0-sequence in floating point arithmetic.

(The difference between fixed and floating point arithmetic can be expressed by convenient selection of norms in Definition 3.)

Let us compare these theoretical results with experiments. The figures 1, 2, 3, 4 show the absolute value of errors in logaritmic scales, so that the errors should lie on a straight line with slope k. While in the three last figures there is very good accordance with theoretical conclusions, from the first figure we see that the actual computation forms an α_1-sequence opposite to theoretical α_2-sequence. By explaining this difference we meet one circumstance which is of great importance in the interpretation of our concept of numerical stability, namely the concept of computation in fixed or floating point arithmetic. Our theoretical results were obtained under the classical concept of fixed point computation, where it is assumed that all numbers have the same number of digits to the right of the decimal (or binary) point. To-day the term fixed point arithmetic is, in connection with digital computers in particular, used when during the computation different quantities are evaluated in different scales which do not change

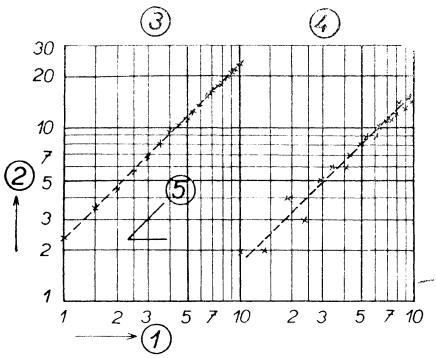

Fig. 3. Error trend for the process (6) in fixed point

1 ... $n/1000$
2 ... $\delta y_n \times 10^7$, computation on LGP 30
3 ... slope of theoretical α_1-L-sequence

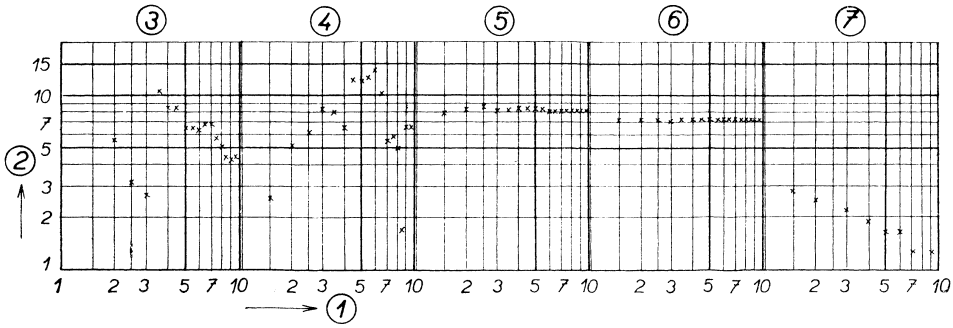

Fig. 4. Error trend for the process (6) in floating point

1 ... $n/1000$
2 ... error δy_n
3 ... $\delta y_n \times 10^{10}$, computation on SIRIUS
4 ... $\delta y_n \times 10^{14}$, computation on D 2
5 ... $\delta y_n \times 10^{10}$, computation on ZUSE 23
6 ... $\delta y_n \times 10^8$, computation on LGP 30
7 ... $\delta y_n \times 10^7$, computation on ER 56

during the computation. This non-classical concept of fixed point computation is, in some sense at least, intermediate between fixed point computation in the classical sense and floating point computation. When computing the sequence (4) the non-classical concept of fixed point arithmetic was used. If the machine program for computation of this sequence is modified, using the same scales for all quantities in the computation, the results exhibited on fig. 5 are obtained. Thus it may be seen that the

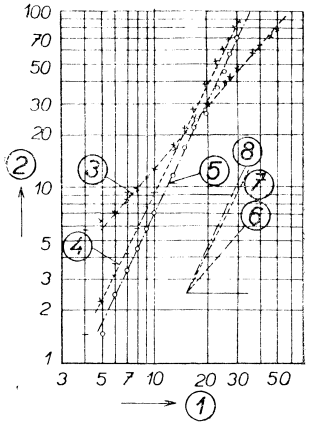

Fig. 5. Error trend for the process (3) in fixed point

1 ... $n/10$
2 ... $\delta z_n \times 10^4$ for 3
$\delta z_n \times 10^3$ for 4
$\delta z_n \times 10^2$ for 5
3 ... computation of y_n to 32 bits
4 ... computation of y_n to 18 bits
5 ... computation of y_n to 15 bits
6 ... actual slope of α_k-sequence 3: $k = 1{,}09$
7 ... actual slope of α_k-sequence 4: $k = 1{,}96$
8 ... actual slope of α_k-sequence 5: $k = 2{,}03$

use of different scales changes the actual character of the fixed point computation and that the actual process behaves as a floating point computation. However, in our experience, this phenomenon is rather rare. In more complicated calculations these distinctions disappear, and the characterisation by means of α_k-sequences is almost independent of the choice of scales.

Now we shall notice some practical aspects of the theory of α_k-sequences. The assessment of numerical stability by means of α_k-sequences has an asymptotic essentially qualitative character. This qualitative assessment can be exploited in various ways which will be illustrated by examples.

A comparison of different methods of solution of a given problem will be most conveniently based not only on computer time and required memory capacity but also on their numerical stability. In should be borne in mind, however, that the assessment by means of α_k-sequences is only relative with respect to the parameter which depends on the sequence of numerical processes in which our problem has been embedded. This fact is very important especially in problems of mathematical analysis where the numerical processes used are often finite - dimensional approximations of continuous problems. In such cases it is natural to introduce the sequence of numerical processes in such a manner that its limit is the continuous problem. Here when comparing two processes it is necessary to choose the index of the process in the sequence in both cases in such a manner that the truncation errors be of the same order with respect to this index.

Another application of our concept of α_k-sequences is in a situation when the solution of a problem involves a combination of several methods. For example let it be required to solve a large system of linear algebraic equations (arising e.g. from application of the finite difference method). It may happen that it is most convenient to apply, first, some direct method such as elimination, and then, to increase accuracy, another method, e.g., iteration. Naturally there arises the question whether we are really able to increace accuracy significantly in this manner. In fact if we know that the method of elimination forms an α_i-sequence while that of iteration forms an α_j-sequence of numerical processes, iteration will improve the final solution only if $j < i$ in the best possible way.

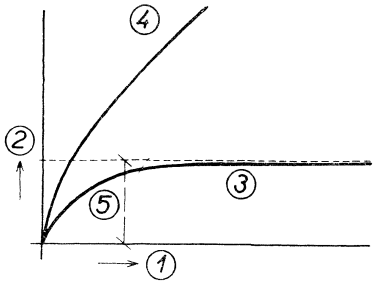

Fig. 6. Local and global stability

1 ... error ε
2 ... erorr δ
3 ... local stability
4 ... global stability
5 ... critical error

Occasionally we may use the conclusions concerning α_k-sequences even in a quantitative manner. For example let it be required to obtain estimates of the round-off errors when solving the differential equation

$$y'' - g(x)\, y = f(x)$$

with boundary conditions $y(0) = y(1) = 0$ by the simplest finite difference method with the mesh-size h. When solving the difference equation by Gauss' elimination we obtain a sequence of numerical processes which forms an α_2-sequence with respect to $1/h$ independently of g. Thus we may solve the difference equation for $g = 0$, and since in this case the exact solution of the system of linear equations can be found, we can also determine the round-off errors. Then we can expect that using the same computer and the same program the errors will be the same in the cases $g = 0$ and $g \neq 0$.

We have given here an outline of several simple ways in which the theory of α_k-sequences may be employed. In conclusion of this theoretical part of the present paper dealing with general questions of numerical stability, I should like to mention another question which can be very important in practical computations. We have introduced a certain concept of stability. The basic idea of this concept is that, in the case of stability, any required accuracy in the result can be achieved by sufficient accuracy in the individual operations. The graph representing the dependence of the error in the individual operations on the error in the result, in the case of stable processes, passes through the origin (see fig. 6). However, the following two cases may occure which are qualitatively distinct.

The case which we will call *global stability*, when the function $\delta = f(\varepsilon)$ is not bounded (see fig. 6). In this case to finite errors in individual operations there corresponds finite errors of result regardless of their magnitude.

Another case, which we will call *local stability*, occurs when the function $\delta = f(\varepsilon)$ is bounded. Then, to a finite error in the individual operations there corresponds a finite error in the result only if the former is smaller than the critical error. The numerical process becomes "unstable" if this critical error is exceeded.

Practical computations frequently involve locally stable processes. The critical accuracy often varies with different arrangements of operations, details of the machine program and modifications which lead to improving convergence.

We have introduced the concept of a numerical process and its stability, and investigated some its aspects. It is obvious that, from a purely deductive point of view, the stability in the sense defined above cannot ensure a correct result without exact estimates and detailed analysis of the overall performance of the computer. However, in complicated cases it is in general not feasible to derive detailed estimates. But in our experience, processes which are stable in our sense are most likely to be realizable without difficulties.

After having explained the general theory of α_k - sequences, let us consider some concrete numerical methods of mathematical analysis.

First take difference methods for solving initial value problems for ordinary differential equations. Let there be given a differential equation

$$y' = f(x, y) \tag{13}$$

with initial condition $y(0) = y_0$. Divide the interval $\langle 0, a \rangle$ into N parts of length h. The approximate solution of (13) at the points $x_n = nh$, $n = 0, 1, \ldots, N$, is to be found from the difference equation

$$\sum_{\nu=0}^{k} \alpha_\nu y_{n+\nu} = h \sum_{\nu=0}^{k} \beta_\nu f(x_{n+\nu}, y_{n+\nu}), \quad n = 0, 1, \ldots, N - k. \tag{14}$$

We shall say that the degree of the difference equation (14) is p if, for every sufficiently smooth solution $y(x)$ of the differential equation (13), the equation

$$\sum_{\nu=0}^{k} \alpha_\nu y(x_{n+\nu}) - h \sum_{\nu=0}^{k} \beta_\nu y'(x_{n+\nu}) = 0(h^{p+1}) \tag{15}$$

holds. The convergence theory for the formulae (14) was studied very thoroughly by Dahlquist. The basic theorem of Dahlquist's theory can be expressed in the following form:

Theorem 1. *Let $y(x)$ be a (sufficiently smooth) solution of the differential equation (13), let be given a difference formula of degree $p \geq 1$; let all roots of the characteristic polynomial*

$$p(\zeta) = \sum_{\nu=0}^{k} \alpha_\nu \zeta^\nu \tag{16}$$

have absolute values at most 1 and let all roots lying on the unit circle be simple.

Then the following estimate holds:

$$|y(x_n) - \tilde{y}_n| \leqq K_1 \vartheta + K_2 \left(\frac{\delta}{h} + K_3 h^p \right), \tag{17}$$

where $y(x_n)$ is the exact solution at the point x_n, $\tilde{y}_n$ is the solution of the equation

$$\sum_{\nu=0}^{k} \alpha_\nu \tilde{y}_{n+\nu} = h \sum_{\nu=0}^{k} \beta_\nu f(x_{n+\nu}, \tilde{y}_{n+\nu}) + \delta_n, \quad n = 0, 1, \ldots, N - k, \tag{18}$$

$\vartheta = \max\limits_{\nu=0,1,\ldots,k-1} |y(x_\nu) - \tilde{y}_\nu|$, $\delta = \max\limits_{n=0,1,\ldots,N-1} |\delta_n|$ *and* K_1, K_2, K_3 *are constants which can depend on the equation* (13) *and on the definition interval of its solution, but are independent of* h.

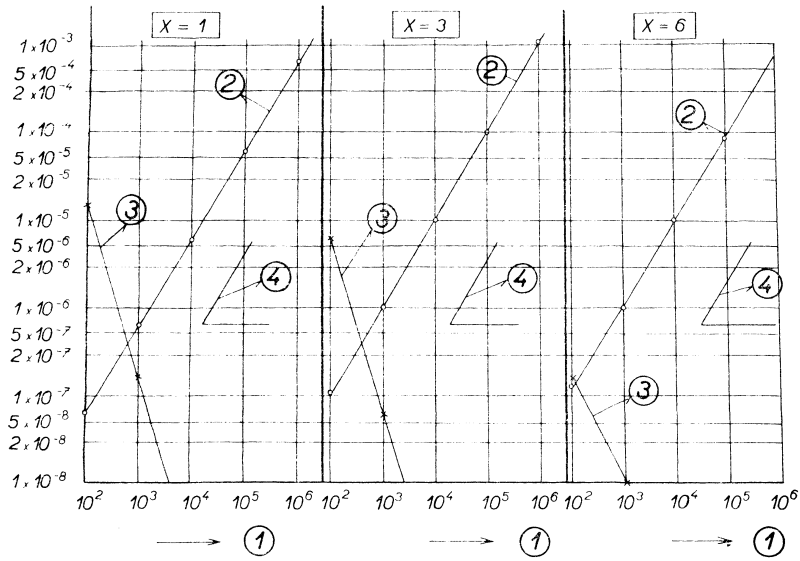

Fig. 7. Dependence of the error of the numerical solution of the differential equation $y' = 1 - y$, $y(0) = 2$ by formula (19) on the step length h

1 ... $1/h$
2 ... round-off error
3 ... error of the method
4 ... α_1-L-sequence

From this theorem it follows immediately that, under the same assumptions, the following theorem holds:

Theorem 2. *The difference formula* (14) *defines a sequence of numerical processes which is an* α_1 *- L-sequence with respect to the parametr* $1/h$.

It is essential for the assertion of this theorem as well as for the Dahlquist's convergence theorem that the interval in which the solution is sought is finite. Thus we cannot eliminate cases in which the errors increase by extending the definition interval. Therefore, if we solve our differential equation on a very long interval, it will be convenient to have a condition which makes it possible to eliminate formulae with above mentioned property. It can be shown that the assertion of Theorem 2 also holds on infinite intervals (certainly only for such differential equations for which it is possible to speak about solutions on infinite intervals) if we eliminate those formulae which have characteristic polynomials with more than one root on the unit circle.

Let us illustrate these theorems by a simple example. Solve the differential equation $y' = 1 - y$ with the initial condition $y(0) = 2$ by Adams-Bashforth's formula

$$y_{n+2} = y_{n+1} + h\left(\tfrac{3}{2}y'_{n+1} - \tfrac{1}{2}y'_n\right) \tag{19}$$

for $h = 10^{-2}, 10^{-3}, \ldots, 10^{-6}$. The computation was carried out on the Czechoslovak computer Epos 1. Fig. 7 shows the truncation and round-off errors for several points of the domain of the solution of our differential equation, in dependence on $1/h$. We have the possibility to exhibit separately the truncation error (i.e. the error between the exact solution of the given differential equation and the exact solution of (19)) and the round-off error because in our case the difference equation (19) is linear with constant coefficients so that its solution is available in explicit form. From the figure we see that the actual sequence of numerical processes indeed forms an α_1-sequence, since the characteristic polynomial of the formula (19) has only one root on the unit circle, independently of the length of the interval.

As another example of application of the theory of α_k-sequences, we will study the difference method for solving linear partial differential equation of parabolic type. In order to simplify the formulations we will restrict our considerations to the first boundary value problem for the heat-conduction equation

$$\frac{\partial u}{\partial t} = \frac{\partial^2 u}{\partial x^2}\,; \tag{20}$$

however all our conclusions hold even for other boundary conditions and for more general equation

$$c(x, t)\frac{\partial u}{\partial t} = \frac{\partial}{\partial x}\left(p(x, t)\frac{\partial u}{\partial x}\right) - q(x, t)\,u + f(x, t)\,. \tag{21}$$

Let us cover the x, t-plane by a network of mesh-sizes h, τ respectively, and approximate the equation (20) by the difference equation

$$\frac{u_n^{(l)} - u_n^{(l-1)}}{\tau} = \alpha\frac{1}{h^2}\Delta u_n^{(l)} + (1 - \alpha)\frac{1}{h^2}\Delta u_n^{(l-1)} \tag{22}$$

or

$$\left(\frac{h^2}{\tau} E - \alpha \Delta\right) u_n^{(l)} = \left(\frac{h^2}{\tau} E + (1 - \alpha) \Delta\right) u_n^{(l-1)}. \tag{23}$$

Here E denotes the identity operator, Δ is the operator defined by $\Delta u_n^{(l)} = u_{n-1}^{(l)} - 2u_n^{(l)} + u_{n+1}^{(l)}$, $u_n^{(l)}$ is the approximation of the solution at the point $(nh, l\tau)$ and α is a parameter, $0 \leqq \alpha \leqq 1$. For $\alpha = 0$ we obtain the so-called explicit formula, in other cases implicit formulae. It is well known that using these formulae, we obtain approximate solution with truncation error $O(\tau + h^2)$ in the case $\alpha \neq \frac{1}{2}$ and with truncation error $O(\tau^2 + h^2)$ in the case $\alpha = \frac{1}{2}$.

First investigate the explicit formula. In this case the condition

$$\tau \leqq \tfrac{1}{2} h^2 \tag{24}$$

guarantees the convergence theorem. The stability of the explicit difference formula for solving equation (20) is described in the following theorem:

Theorem 3. *The sequence of numerical processes* (23) *forms, if* $\alpha = 0$ *and the assumption* (24) *holds, an* α_2*-sequence of numerical processes with respect to* $1/h$.

The big advantage of the explicit formula consists in its simplicity. But this advantage is weakened by that fact the inequality (24) must be satisfied. This inequality considerably restricts the choice of the magnitude of the time meshsize so that, esspecially when increasing the accuracy, the dimensions of the network increase rapidly.

This disadvantage is obviated on using implicit formulae for $\frac{1}{2} \leqq \alpha \leqq 1$. The so-called Crank-Nicolson formula, obtained by putting $\alpha = \frac{1}{2}$, is evidently the most convenient among implicit formulae because it allows to take $\tau = O(h)$ without reducing the truncation error. In this case the numerical process by means of which we achieve the approximate solution of our problem is not described by equation (23) alone. We must also consider the numerical process by means of which the system of linear equations on each time row is solved. The stability of the Crank-Nicolson formula is then described in the following general theorem (which follows from the results of Samarski):

Theorem 4. *The sequence of numerical processes which arises when solving the equation* (20) *by means of the Crank-Nicolson formula is an* $\alpha_{2+\varrho}$*-sequence of numerical processes with respect to* $1/h$ *if* $\tau = O(h)$ *and if the numerical process for solving the above mentioned system of linear equations leads to errors in residues of order* $1/h^{\varrho}$.

This theorem describes the stability of the Crank-Nicolson formula under the assumption that the behaviour of the chosen method for solving the corresponding system of linear equations with respect to round-off errors is known. The most convenient

method for solving system of linear equations with a three diagonal matrix is Gauss' elimination, as it needs only $O(n)$ operations for solving n equations. In our special case, for elimination $\varrho = 0$ and consequently the Crank-Nicolson formula together with Gaussian elimination forms an α_2-sequence of numerical processes.

This result shows that the explicit formula as well as Crank-Nicolson formula behave in the same manner with respect to round-off errors. But a big advantage of the latter consists in the number of necessary operations. The explicit formula requires $O(1/h^3)$ operations but the Crank-Nicolson formula only $O(1/h^2)$ operations for reaching the solution in the same time.

Questions of stability of the difference analogue of parabolic equation are frequently discussed in literature. But almost all authors understand under the stability of a difference equation essentially only its correctness, i.e. the continuous dependence on boundary conditions and on right-hand sides. This concept is obviously of basic importance especially when studying convergence, but as a characterisation of the whole numerical process it may be, as we have seen, incomplete.

In this report I have tried to describe and illustrate on some examples the model of a numerical process and its stability. This model is based on experience which have been obtained at the Institute of Mathematics of the Czechoslovak Academy of Sciences.

At the end of this report I should like to try to describe the basic idea of classification of numerical processes by means of α_k-sequences. This classification is an effort to extend the amount of information available concerning the given numerical method. It is intended to yield a mathematically founded basis for heuristic considerations which are carried out before the beginning of computation by anyone working in numerical methods.

References

[1] *G. Dahlquist:* Convergence and Stability in the Numerical Integration of Ordinary Differential Equations, Mathem. Scand., 4 (1956), 33 – 53.

[2] *P. Henrici:* Discrete Variable Methods in Ordinary Differential Equations, J. Wiley & Sons Inc., New York, London, 1962.

[3] *I. Babuška, M. Práger, E. Vitásek:* Numerické řešení diferenciálních rovnic, SNTL Praha, 1964.

[4] *I. Babuška, M. Práger, E. Vitásek:* Numerical Solution of Differential Equations, J. Wiley & Sons Inc., New York, London, 1965.

[5] *А. А. Самарский:* Однородные разностные схемы для нелинейных уравнений параболического типа, Ж. вычисл. мат. и мат. физ., 2 (1962), 25 – 56.

Emil Vitásek, C. Sc., Matematický ústav ČSAV, Praha 1, Žitná 25, ČSSR.