ZASTOSOWANIA MATEMATYKI
APPLICATIONES MATHEMATICAE
XIV, 3 (1974)

KRYSTYNA ZIĘTAK (Wrocław)

INVESTIGATION OF THE ALGORITHMS DETERMINING THE OPTIMUM RATIONAL FUNCTION OF THE ADI-METHOD

1. Introduction. In connection with the alternating direction implicit (ADI) method (see, e. g., Varga [3], p. 209) the following problem was formulated:

For given numbers, a natural m and a real k' $(0 < k' < 1)$, find positive parameters $r_{1m}, r_{2m}, \ldots, r_{mm}$ such that

$$L_m(k') \equiv \max_{k' \leqslant x \leqslant 1} \left| \prod_{j=1}^{m} \frac{x - r_{jm}}{x + r_{jm}} \right| \tag{1}$$

has the minimal value.

This problem has a unique solution ([3], p. 223).

Parameters r_{jm} have different values in the interval $(k', 1)$. Let $R_m \equiv R_m(k')$ denote the set of optimal parameters $r_{jm} \equiv r_{jm}(k')$ arranged increasingly. The optimum rational function

$$f_m(x;\ R_m) \equiv \prod_{j=1}^{m} \frac{x - r_{jm}}{x + r_{jm}}$$

in the interval $[k', 1]$ has $m+1$ extremal points $u_{jm} \equiv u_{jm}(k')$ at which it attains absolute maximum values (1) with alternating signs ([3], p. 223) with $k' = u_{0m} < u_{1m} < \ldots < u_{mm} = 1$.

Let $U_m \equiv U_m(k')$ be the set of extremal points u_{jm}. For optimal parameters r_{jm} we have ([4], p. 190)

$$r_{jm}(k') = \operatorname{dn}\left[\left(1 - \frac{2j-1}{2m}\right) K(k);\ k\right], \quad j = 1, 2, \ldots, m, \tag{2}$$

and for extremal points (see [6])

$$u_{jm}(k') = \mathrm{dn}\left(\frac{m-j}{m}K(k);\ k\right), \quad j = 0, 1, \ldots, m, \tag{3}$$

where $k = \sqrt{1-k'^2}$, $K(k)$ is the complete elliptic integral of first kind and $\mathrm{dn}(u;\ k)$ is the Jacobi elliptic function (for fundamental notions of elliptic functions see, e. g., the book [2]).

It is very troublesome to evaluate exactly r_{jm} and u_{jm} immediately from formulas (2) and (3). They are therefore usually replaced by approximate expressions obtained by the expansions of elliptic functions into series with respect to powers of k' ([4], p. 191). Exact algorithms are known only for $m = 2^p$. In the present paper, which is a short version of the second part of paper [6], we compare numerical properties of two such algorithms, namely of the Wachspress-algorithm ([3], p. 225) and of the JR-algorithm [7].

The Wachspress-algorithm for determination of elements of the set $R_m(k')$ for $m = 2^p$ will be called the *WR-algorithm*. In section 2 we recall fundamental properties of parameters r_{jm} and of the algorithms WR and JR. In the following sections we try to explain experimentally observed properties of these algorithms. The error analysis is preceded in section 3 by some numerical examples showing the better behaviour of the JR-algorithm than that of the WR-algorithm. In section 4 we prove that stage I and a single step of stage II of the WR-algorithm and one step of the JR-algorithm are numerically stable. We suppose, therefore, that the worse behaviour of the WR-algorithm must be caused by error propagation from the preceding steps. The propagation of errors and the total error are discussed in section 5 for the WR-algorithm, and in section 6 for the JR-algorithm. Thus we state among others that in the WR-algorithm there appears a distinct trend in the increase of error. In the JR-algorithm we observe simultaneous opposite trends, the expansion and damping of the error, which result in moderate increase of error. In section 7 we compare factors of error propagation in both algorithms. We prove there that the factors of error propagation in the WR-algorithm are always greater than the corresponding factors of the JR-algorithm. This implies that the total error of the WR-algorithm has a greater estimation than the total error of the JR-algorithm. Moreover, it will be stated that for the total error of the element $r_{jm}(k')$, determined by the JR-algorithm, there exists an estimation independent of k', which shows the stability of the whole JR-algorithm.

2. Properties of parameters r_{jm}. WR- and JR-algorithms. We have proved in paper [7] the following

THEOREM 1. *Parameters $r_{jm}(k')$ are decreasing functions of the variable k'. Moreover, for $j = 1, 2, \ldots, m$, we have*

$$\lim_{k' \to 0^+} r_{jm}(k') = 0, \tag{1}$$

$$\lim_{k' \to 1^-} r_{jm}(k') = 1, \tag{2}$$

$$\lim_{k' \to 1^-} \frac{dr_{jm}(k')}{dk'} = \frac{1}{2}\left(1 + \cos\frac{2j-1}{2m}\pi\right), \tag{3}$$

$$r_{m+j,2m} = \sqrt{\frac{r_{jm} + k'^2 + \sqrt{(1-k')(1+k')(r_{jm}-k')(r_{jm}+k')}}{1+r_{jm}}}, \tag{4}$$

$$r_{m-j+1,2m} = \frac{k'}{r_{m+j,2m}}. \tag{5}$$

Properties (1)-(3) will be used in section 7. Properties (4) and (5) immediately imply (see [7]) the following algorithm determining for $m = 2^p$ the elements of the set $R_m(k')$:

JR-algorithm. It is known that $r_{11} = \sqrt{k'}$. In order to obtain the set $R_{2^p}(k')$ we successively appoint from formulas (4) and (5) elements of the sets $R_2(k')$, $R_4(k')$, $R_8(k')$, ..., $R_{2^p}(k')$.

Let

$$r = r_{jn}, \quad z_1 = r_{n+j,2n}, \quad z_2 = r_{n-j+1,2n} \tag{6}$$

$$(j = 1, 2, \ldots, n;\ n = 1, 2, 4, \ldots, 2^{p-1}).$$

It easily follows from formulas (4) and (5) that the elements z_1 and z_2 are square roots of zeros of the trinomial

$$(1+r)z^2 - 2(k'^2 + r)z + k'^2(1+r) = 0. \tag{7}$$

Hence one step of the *JR*-algorithm is reduced to determining the zeros of trinomial (7). This holds also for the *WR*-algorithm. We have ([3], p. 225)

WR-algorithm. Stage I. Sequences $\{a_i\}$ and $\{\beta_i\}$ are obtained from formulas

$$a_0 = k', \quad \beta_0 = 1, \tag{8}$$

$$a_{i+1} = \sqrt{a_i \beta_i}, \quad \beta_{i+1} = \frac{a_i + \beta_i}{2} \quad (i = 0, 1, \ldots, p-1).$$

Stage II. Elements s_{jn} $(j = 1, 2, \ldots, n;\ n = 1, 2, 4, \ldots, 2^p)$ are obtained from formulas

$$s_{11} = \sqrt{\alpha_p \beta_p} = \alpha_{p+1}, \tag{9}$$

$$s_{n+j,2n} = s_{jn} + \sqrt{(s_{jn} - \alpha_i)(s_{jn} + \alpha_i)} \tag{10}$$

$$(j = 1, 2, \ldots, n;\ i = p, p-1, \ldots, 1;\ n = 2^{p-i}),$$

$$s_{n-j+1,2n} = \frac{\alpha_{i-1}\beta_{i-1}}{s_{n+j,2n}}. \tag{11}$$

Elements s_{jm} belong to the set $R_m(k')$ for $m = 2^p$.

Let

$$s = s_{jn}, \quad y_1 = s_{n+j,2n}, \quad y_2 = s_{n-j+1,2n}, \quad c = \alpha_i \tag{12}$$

$$(j = 1, 2, \ldots, n;\ i = p, p-1, \ldots, 1;\ n = 2^{p-i}).$$

Then it follows from (10) and (11) that elements y_1 and y_2 are zeros of the quadratic trinomial

$$y^2 - 2sy + c^2 = 0. \tag{13}$$

Thus one step of stage II of the WR-algorithm is equivalent to the determination of the zeros of the trinomial (13). This will make the analysis of errors in the next sections easier.

For $m = 2^p$ we have (see [7])

$$U_{2m}(k') = R_m(k') \cup U_m(k') = U_1(k') \cup R_1(k') \cup R_2(k') \cup R_4(k') \cup \ldots \cup R_{2^p}(k')$$

which follows immediately from formulas (1) (1.2) and (1.3). Elements u_{jm} have therefore properties similar to those of the parameters r_{jm}. They can be determined by the algorithms JU and WU (see [6]) which differ from the algorithms JR and WR by the fact that we begin the determination of successive elements from elements k' and 1 for the JU-algorithm (α_p, β_p for WU) instead of the element $\sqrt{k'}$ (α_{p+1} for WR) (details are omitted). Error analysis will be performed only for the algorithms JR and WR. Obviously, it concerns also the algorithms JU and WU.

3. Costs and experimental comparison of accuracy of the algorithms JR and WR. The number of operations in the JR-algorithm is almost two times greater than in the WR-algorithm. This is shown in the following table:

---

(1) Formulas are numbered independently in every section. If, however, the reference is made to a formula from another section, double-numbers are used, the first part indicating the section.

	+	×	/	$\sqrt{\ }$
WR	$3\cdot 2^p + p - 3$	$2^p + 3p$	$2^p - 1$	$2^p + p$
JR	$5\cdot 2^p - 3$	$2^{p+1} - 1$	$2^{p+1} - 2$	$2^{p+1} - 1$

It is, however, to remark here that in the *JR*-algorithm, in the process of determining the set $R_m(k')$ for $m = 2^p$, we obtain all the remaining sets $R_n(k')$, where n is the natural power of 2 smaller than m. This may be used while we choose in the ADI-method the corresponding m which assures a sufficient reduction of the error vector of the solution of the system of linear equations. In the *WR*-algorithm we indirectly obtain the optimal sets of powers of 2 smaller than m for the interval $[\alpha_i, \beta_i]$, whereas it does not hold for the interval $[k', 1]$.

We have compared the Wachspress *WR*-algorithm with the *JR*-algorithm. Calculations, performed on the ODRA 1204 computer (almost 12-digits mantissa), proved that the *JR*-algorithm determines the elements of the set $R_m(k')$ with a greater accuracy than the *WR*-algorithm. As criterion of the accuracy obtained we have assumed the bahaviour of the numerically obtained optimum functions $f_m(x; R_m)$ at points of the numerically determined set $U_m(k')$, i.e. the examination whether this set is the alternans of the function obtained. Since the values of $f_m(u_{jm}; R_m)$ are evaluated with a great accuracy, the criterion is easily verifiable.

Let now RW_m, UW_m and $fW_m(x)$ denote the sets $R_m(k')$, $U_m(k')$ and the function $f_m(x, R_m)$, respectively, obtained numerically from the algorithms *WR* and *WU*, and let RJ_m, UJ_m and $fJ_m(x)$ be the sets $R_m(k')$, $U_m(k')$ and the function $f_m(x; R_m)$, respectively, obtained numerically from the algorithms *JR* and *JU*.

The following table contains examples for $m = 8$ and $k' = .8$ (figures identical with figures of the former numbers are omitted):

j	RJ_m	UJ_m	fJ_m for $u_{jm} \in UJ_m$
0	–	.799 999 999 99	$+.726\ 754\ 096\ 22_{10} - 12$
1	.801 720 353 62	.806 835 859 64	– 3 97
2	.815 209 061 81	.826 608 476 25	+ 3 71
3	.840 705 875 69	.857 074 215 46	– 3 73
4	.875 187 871 88	.894 427 190 99	+ 5 47
5	.914 089 449 47	.933 408 082 48	– 3 63
6	.951 581 311 75	.967 810 061 21	+ 3 85
7	.981 343 360 20	.991 527 570 86	– 4 90
8	.997 854 172 44	1.000 000 000 0	+ 5 36

j	RW_m	UW_m	fW_m for $u_{jm} \in UW_m$	fW_m for $u_{jm} \in UJ_m$
0	–	.799 999 999 99	$.145\,350_{10}-11$	$.145\,350_{10}-11$
1	.806 835 854 41	.806 835 854 41	0	$.349\,112_{10}-23$
2	.806 835 825 43	.826 608 476 25	$.145\,351_{10}-11$	$.145\,351_{10}-11$
3	.857 074 302 70	.857 074 228 53	0	$.340\,072_{10}-23$
4	.857 074 228 53	.894 427 190 99	$.145\,350_{10}-11$	$.145\,350_{10}-11$
5	.933 408 068 25	.933 408 068 25	0	$.339\,883_{10}-23$
6	.933 408 987 47	.967 810 061 19	$.145\,351_{10}-11$	$.145\,351_{10}-11$
7	.991 527 612 89	.991 527 577 27	0	$.348\,039_{10}-23$
8	.991 527 577 27	1.000 000 000 0	$.145\,350_{10}-11$	$.145\,350_{10}-11$

Remark. For both algorithms the elements u_{08}, u_{48} and u_{88} are determined immediately from formulas: $u_{08} = k'$, $u_{48} = \sqrt{k'}$ and $u_{88} = 1$. They are, therefore, charged by a smaller error than other elements. That is why the values of the function $fJ_m(x)$ differ at these points a little from the values at the remaining points. The function $fW_m(x)$ assumes at some points of the set UW_m values equal to 0 since several elements of the set UW_m are identical with elements of the set RW_m.

These examples imply that neither the set UW_m nor the set UJ_m form an alternans of the function $fW_m(x)$. This is above all caused by the errors with which elements of the set RW_m are numerically determined. The quality of the set RW_m depends among others on the parameter k'. If $k' \to 0$, then the sets RW_m are closer to the optimal set $R_m(k')$, obviously only, when m is not too large. The function fW_m assumes maximal absolute values at points of the set UW_m. They are at most two times greater than the maximal absolute values of the function fJ_m. Hence using the function $fW_m(x)$, for example in the ADI-method, does not lead to much worse results than using the function $fJ_m(x)$, the more so that in most applications of the ADI-method the value of k' is small, and the function $fW_m(x)$ is then almost identical with the function $fJ_m(x)$ if p is not too large. The problem of numerical construction of an optimum function is very interesting, independently of applications in the ADI-method. The above-described experiment appears to show that the JR-algorithm gives a better approximation of the optimum function than the Wachspress algorithm.

Let us try to explain why the error given by the WR-algorithm is in some cases too large. This is connected with the fact that the elements of sequences (2.8) have the common limit

$$(1) \qquad \lim_{p\to\infty} \alpha_p(k') = \lim_{p\to\infty} \beta_p(k') = \frac{\pi}{2}\left(\int_0^{\pi/2} \frac{dt}{[1-(1-k')^2\sin^2 t]^{1/2}}\right)^{-1}.$$

The elements of these sequences lie in the interval $[k', 1]$. If p is sufficiently large, then the elements α_p and β_p have the same number of significant figures in the given arithmetics connected with the computer used. Suppose that p is so large that

$$|\alpha_p - \beta_p| < 2^{-t}\max(\alpha_p, \beta_p), \tag{2}$$

where t is the number of digits of the binary expansion. The numbers α_p and β_p have in this arithmetics a common representation. We get, therefore, the objects

$$s_{11} = s_{12} = s_{22} = \alpha_p.$$

In this pathological case the set RW_m has for $m = 2^p$ at most $m/2$ different elements (this is the case for $m = 8$ and $k' = .79$). The number of different elements of this set is less than $m/2$ since other pairs of elements α_i, β_i have also a common representation (e. g. for $m = 8$ and $k' = .999$ the set RW_m has only two different elements). In this case the set RW_m does not have the properties of an optimal set. Moreover, elements can be charged with such a great error that the ordering of the set will be disturbed. In the given example, for $m = 8$ and $k' = 8$, the successive elements of the set RW_m form no increasing sequence, as it is theoretically assumed. In extreme cases it is at all impossible to determine the set RW_m, since in formula (2.10) we have the root of a negative number (e. g. for $m = 8$ and $k' = .9999$). The same troubles appear when we evaluate elements of the set UW_m.

Let us now observe the behaviour of the difference $\beta_i - \alpha_i$. We give the following examples:

k' \ i	0	1	2	3	4	5	6
.8	.200 000	.005 572 8	.000 004 3	0 (2^{-37})	–	–	–
.0001	.999 900 0	.490 050 0	.184 310 8	.028 579 3	.000 688 7	.000 000 4	0

Hence, in the most frequent applications of the ADI-method ($k' \sim .0001$), inequality (2) is satisfied for $p = 6$. The rapid convergence of sequences (2.8) is the main cause of the bad behaviour of the WR-algorithm. This will be shown in the following sections.

4. Analysis of the error of one step of the algorithms WR and JR. We assume that the calculations are performed on a computer in floating-point arithmetics (see [5]), the fundamental operations of which are in accord with summation rules

$$\mathrm{fl}(a \pm b) = a(1 - \xi_1) \pm b(1 - \xi_2), \quad |\xi_1|, |\xi_2| \leqslant 2^{-t}, \tag{1}$$

and the remaining operations with the following rules:

$$\begin{aligned} &\mathrm{fl}(ab) = ab(1-\varepsilon), \quad |\varepsilon| \leqslant 2^{-t}, \\ &\mathrm{fl}(a/2) = a/2, \quad \mathrm{fl}(\sqrt{a}) = \sqrt{a}(1-\varrho), \quad |\varrho| \leqslant T\cdot 2^{-t}. \end{aligned} \tag{2}$$

The value of T is usually equal to 1.5 or 2. In view of the rounding process applied on the ODRA 1204 computer, we assume $t = 36$, though the mantissa consists of 37 binary digits. The extraction of roots is performed on this computer with double accuracy, and the result has t digits without rounding. Since the Heron-Newton sequence decreases, this cutting-off does not increase the error. We can, therefore, assume that in this case $T = 1/2$. Equality (1) for the ODRA 1204 computer is substituted by the stronger

$$\mathrm{fl}(a \pm b) = (a \pm b)(1-\varepsilon), \quad |\varepsilon| \leqslant 2^{-t}. \tag{3}$$

An arithmetics with such a summation will be called *good*. Any arithmetics for which formula (3) is not satisfied in general, will be called *worse*. It is characterized by formulas

$$\mathrm{fl}(a \pm b) = \begin{cases} (a \pm b)(1-\delta), & |\delta| \leqslant 2^{-t} \text{ if } \operatorname{sign}(a) = \operatorname{sign}(\pm b), \\ a(1-\varepsilon) \pm b, & |\varepsilon| \leqslant 2^{-t} \text{ if } \operatorname{sign}(a) \neq \operatorname{sign}(\pm b) \text{ and } |a| \geqslant |b|, \end{cases} \tag{4}$$

which are easily obtained from (1).

We have proved in [6] that, for both arithmetics, one step of the algorithms WR and JR is stable. Now we will analyze only the "worse" arithmetics.

Let b be the exact result, and $\tilde{b}$ the value obtained in fl-arithmetics, i.e. $\tilde{b} = b(1-\xi)$, where ξ is the relative error of the result.

We now determine relative errors E_i and F_i for numerically evaluated elements of sequences (2.8). Let

$$\tilde{\alpha}_i = \alpha_i(1-E_i), \quad \tilde{\beta}_i = \beta_i(1-F_i), \quad i = 0, 1, \ldots, p.$$

For arithmetics (4) and (2) formulas (2.8) imply

$$\begin{aligned} &\tilde{\alpha}_0 = \alpha_0, \quad \tilde{\beta}_0 = \beta_0, \\ &\tilde{\alpha}_1 = \sqrt{\tilde{\alpha}_0\tilde{\beta}_0(1-\varepsilon_1)}\,(1-\varrho_1) = \alpha_1(1-\varrho_1-\tfrac{1}{2}\tilde{\varepsilon}_1), \\ &\tilde{\beta}_1 = \frac{\tilde{\alpha}_0+\tilde{\beta}_0}{2}(1-\delta_1) = \beta_1(1-\delta_1), \end{aligned}$$

where $|\varepsilon_1|, |\delta_1| \leqslant 2^{-t}$, $|\varrho_1| \leqslant T\cdot 2^{-t}$, and

$$|\tilde{\varepsilon}_1| \leqslant 2^{-t}\left(1 + \frac{2^{-t}}{1-2^{-t}}\right) = 2^{-t} + 2^{-2t} + 2^{-3t} + \ldots$$

We often obtain estimations containing elements of different order which complicates the record giving no essential information about the value of the estimated error. In the sequel we will, therefore, retain only the greatest essential components of estimations. This means that we generally omit components of order $2^{-2t}, 2^{-3t}, \ldots$ in presence of components of order 2^{-t}. We write $|\tilde{\varepsilon}_1| \leqslant 2^{-t}$. Hence

$$E_0 = 0, \quad F_0 = 0, \quad E_1 = \tfrac{1}{2}\tilde{\varepsilon}_1 + \varrho_1, \quad F_1 = \delta_1. \tag{5}$$

It follows from (2.8) that, in general,

$$\tilde{\alpha}_{i+1} = \sqrt{\tilde{\alpha}_i \tilde{\beta}_i (1-\varepsilon_{i+1})}\,(1-\varrho_{i+1}) = \sqrt{\alpha_i \beta_i (1-E_i)(1-F_i)(1-\varepsilon_{i+1})}(1-\varrho_{i+1}),$$

$$\tilde{\beta}_{i+1} = \frac{\tilde{\alpha}_i + \tilde{\beta}_i}{2}(1-\delta_{i+1}) = \frac{\alpha_i + \beta_i}{2}\left(1 - \frac{\alpha_i E_i + \beta_i F_i}{\alpha_i + \beta_i}\right)(1-\delta_{i+1}),$$

where $|\varepsilon_{i+1}|,\ |\delta_{i+1}| \leqslant 2^{-t},\ |\varrho_{i+1}| \leqslant T \cdot 2^{-t}$.

In view of this

$$\begin{aligned} E_{i+1} &= \tfrac{1}{2}(E_i + F_i + \varepsilon_{i+1}) + \varrho_{i+1} \\ F_{i+1} &= \frac{\alpha_i E_i + \beta_i F_i}{\alpha_i + \beta_i} + \delta_{i+1} \end{aligned} \qquad (i = 0, 1, \ldots, p-1). \tag{6}$$

Auxiliary values e_i and f_i are defined by

$$\begin{aligned} e_0 &= 0, & f_0 &= 0, \\ e_{i+1} &= \tfrac{1}{2}(e_i + f_i + 1) + T, & f_{i+1} &= \frac{\alpha_i e_i + \beta_i f_i}{\alpha_i + \beta_i} + 1. \end{aligned} \tag{7}$$

It is easy to verify, comparing (5) and (6), that $|E_i| \leqslant e_i \cdot 2^{-t}$ and $|F_i| \leqslant f_i \cdot 2^{-t}$.

Since $\alpha_i, \beta_i > 0$, we have

$$\frac{\alpha_i e_i + \beta_i f_i}{\alpha_i + \beta_i} \leqslant \max(e_i, f_i). \tag{8}$$

Hence it is possible to modify sequences (7) by substituting the expression on the left-hand side of (8) by $\max(e_i, f_i) = e_i$. It is, however, empirically stated that we obtain a better consistency with elements of sequences (7) if sequences (7) are substituted by sequences

$$\begin{aligned} e_0^* &= 0, & f_0^* &= 0, \\ e_{i+1}^* &= \tfrac{1}{2}(e_i^* + f_i^* + 1) + T, & f_{i+1}^* &= \tfrac{1}{2}(e_i^* + f_i^*) + 1, \end{aligned} \tag{9}$$

since the left-hand side of (8) tends asymptotically to the arithmetic mean $\frac{1}{2}(e_i + f_i)$ for $p \to \infty$. It is easily seen that

$$\begin{aligned} e_i^* &= (-.25 + .5T) + (.75 + .5T)i \\ f_i^* &= (.25 - .5T) + (.75 + .5T)i \end{aligned} \qquad (i = 1, 2, \ldots, p).$$

This implies the following estimation of relative errors of the elements $\tilde{\alpha}_i$ and $\tilde{\beta}_i$ for the ODRA 1204 computer:

$$|E_i| \leqslant i \cdot 2^{-36}, \qquad |F_i| \leqslant i \cdot 2^{-36}.$$

We have thus shown that sequences $\{\tilde{\alpha}_i\}$ and $\{\tilde{\beta}_i\}$ differ from sequences $\{\alpha_i\}$ and $\{\beta_i\}$ on the error level of their representation.

We will now examine the relative error given by one step of stage II of the *WR*-algorithm. For numerically stable solution of the quadratic equation (2.13) the zero obtained is the rounded true zero of the quadratic trinomial which is seemingly disturbed and has the coefficients disturbed on the error level of their representation. For "good" arithmetics such a result was obtained by Kahan [1] for any quadratic trinomial. The discriminant of the trinomial (2.13) can be presented in the form of a product of two factors which shows that, using the good arithmetics, we can get a much stronger result: the zero evaluated is the several times rounded exact zero of the trinomial (2.13) (see [6]). If we use the "worse" arithmetics, we obtain stability of the same type as for the "good" arithmetics and as for the quadratic trinomial, the discriminant of which cannot be presented as the product of factors. Namely, according to (2.12), zeros of the trinomial (2.13) are obtained from the formulas

$$y_1 = s + \sqrt{(s-c)(s+c)}, \tag{10}$$

$$y_2 = c^2/y_1. \tag{11}$$

It follows from (2) and (4) that

$$\begin{aligned} \tilde{y}_1 &= \mathrm{fl}\left(s + \sqrt{(s-c)(s+c)}\right) \\ &= s(1-\delta) + \sqrt{[s(1-\sigma)-c](s+c)(1-\varepsilon)(1-\eta)}\,(1-\varrho), \end{aligned} \tag{12}$$

where $|\delta|, |\sigma|, |\varepsilon|, |\eta| \leqslant 2^{-t}$ and $|\varrho| \leqslant T \cdot 2^{-t}$.

Let $\bar{s} = s(1-\sigma)$. It follows from (12) that

$$\begin{aligned} \tilde{y}_1 &= \bar{s} + s(\sigma-\delta) + \sqrt{(\bar{s}-c)(\bar{s}+c)\left(1+\frac{\sigma s}{\bar{s}+c}\right)}\left(1-\frac{1}{2}\varepsilon-\frac{1}{2}\eta-\varrho\right) \\ &= \left[\bar{s} + \sqrt{(\bar{s}-c)(\bar{s}+c)}\right]\left(1+\frac{s(\sigma-\delta)+\xi\sqrt{(\bar{s}-c)(\bar{s}+c)}}{\bar{s}+\sqrt{(\bar{s}-c)(\bar{s}+c)}}\right) \\ &= \bar{y}_1(1-\chi_1), \end{aligned} \tag{13}$$

where $|\chi_1|, |\xi| \leqslant (T+3/2)2^{-t}$, and the element $\bar{y}_1 = \bar{s} + \sqrt{(\bar{s}-c)(\bar{s}+c)}$ is the exact zero of the trinomial (2.13) with one coefficient disturbed on the error level of its representation. A similar result is obtained for y_2,

$$\tilde{y}_2 = \mathrm{fl}\left(\frac{c^2}{y_1}\right) = \frac{c^2}{\bar{y}_1}(1-\chi_1-\delta) = \bar{y}_2(1-\chi_2), \tag{14}$$

where $|\delta| \leqslant 2 \cdot 2^{-t}$, $|\chi_2| \leqslant (T+7/2)2^{-t}$, and $\bar{y}_2$ is the exact zero of the disturbed trinomial (2.13). Hence, for the "worse" arithmetic, the error made in one step of stage II of the *WR*-algorithm can be represented by two components: by at most several times rounding of the result and by propagation of one rounding of the "old" element s. Thus one step of the *WR*-algorithm, in which we apply formulas (10) and (11), is stable.

It is easy to verify that if we substitute (11) by the equivalent formula

$$y_2 = s - \sqrt{(s-c)(s+c)}\,,$$

then one step of the *WR*-algorithm would not be stable. For details see [6].

The analysis of the error made in one step of the *JR*-algorithm is similar to that of the error made in one step of the *WR*-algorithm since the discriminant of the trinomial (2.7) can also be presented as a product. It can be easily proved that for the "worse" arithmetics, with the additional assumption (2) that the binary expansion of the number k' has zeros only on positions lower than 2^{-t}, i.e.

$$\mathrm{fl}(1-k') = 1-k', \tag{15}$$

evaluated from formulas

$$z_1 = \left(\frac{r+k'^2+\sqrt{(1-k')(1+k')(r-k')(r+k')}}{1+r}\right)^{1/2},$$

$$z_2 = k'/z_1;$$

the values $\tilde{z}_1$ and $\tilde{z}_2$ (see (2.6)) are equal to

$$\tilde{z}_1 = \bar{z}_1(1-\psi_1), \quad |\psi_1| \leqslant (\tfrac{7}{2}+\tfrac{3}{2}T)2^{-t}, \tag{16}$$

$$\tilde{z}_2 = \bar{z}_2(1-\psi_2), \quad |\psi_2| \leqslant (\tfrac{9}{2}+\tfrac{3}{2}T)2^{-t}; \tag{17}$$

$\bar{z}_1$ and $\bar{z}_2$ are the exact square roots of zeros of the trinomial (2.7) in which the element r was disturbed on the error level of its representation ($\bar{r} = r(1-\delta)$, $|\delta| \leqslant 2^{-t}$). If we would use the "good" arithmetics, then the values $\tilde{z}_1$ and $\tilde{z}_2$ will be equal to several times rounded exact values z_1 and z_2 (see [6]). Consequently, we have the following

COROLLARY 1. *In both algorithms, one step is numerically stable in the following sense: evaluated "new" values are several times rounded true values corresponding to the — perhaps a little disturbed — "old" values.*

Since the values numerically obtained show a good behaviour of the *JR*-algorithm and a worse one of the *WR*-algorithm, it is to expect that the influence of the error of the "old" element on the error of the "new" element is for the *JR*-algorithm considerably smaller than for the *WR*-algorithm, which will be proved in the following sections.

(2) This additional assumption causes a certain loss of generalization, but considerably simplifies the transformations.

5. Propagation of the error in the WR-algorithm. Total error. Let us now examine the influence of errors of the elements s_{jn} and a_i on the accuracy of the element $s_{j,2n}$. In section 4 we have examined the error made in one step of the WR-algorithm. Now we deal with the propagation of the error in one step.

Let s, c, y_1 and y_2 have the same meaning as in sections 2 and 4 (see (2.12)). We assume that the element s is evaluated with the relative error $\Delta s/s$, and the element c with the error $\Delta c/c$. The propagation error of the trinomial (2.13) will then be expressed by the formula

$$\frac{\delta y}{y} \cong \left(\frac{\partial y}{\partial s}\frac{s}{y}\right)\frac{\Delta s}{s} + \left(\frac{\partial y}{\partial c}\frac{c}{y}\right)\frac{\Delta c}{c} = \frac{s}{y-s}\frac{\Delta s}{s} + \left(1 - \frac{s}{y-s}\right)\frac{\Delta c}{c}. \tag{1}$$

This implies that

$$\frac{\delta y_1}{y_1} \cong g(s;\ c)\frac{\Delta s}{s} + \big(1 - g(s;\ c)\big)\frac{\Delta c}{c}, \tag{2}$$

$$\frac{\delta y_2}{y_2} \cong -\frac{\delta y_1}{y_1} + 2\frac{\Delta c}{c} \cong -g(s;\ c)\frac{\Delta s}{s} + \big(1 + g(s;\ c)\big)\frac{\Delta c}{c}, \tag{3}$$

where

$$g(s;\ c) \overset{\text{df}}{=} s/\sqrt{s^2 - c^2}, \qquad s > c.$$

The total error of one step of the WR-algorithm is determined by the error made and propagated. It was shown in section 4 that in case of the "worse" arithmetics the errors of elements y_1 and y_2, determined by formulas (4.10) and (4.11), consist of several times rounding of the result and of the propagation of one rounding of the "old" element s. To obtain expressions for total errors of the elements y_1 and y_2 it is therefore necessary to modify formulas (2) and (3). Now, those errors are equal to

$$\frac{\Delta y_1}{y_1} \approx g(s;\ c)\left(\frac{\Delta s}{s} + \xi_1\right) + \big(1 - g(s;\ c)\big)\frac{\Delta c}{c} + \eta_1, \tag{4}$$

$$\frac{\Delta y_2}{y_2} \approx -g(s;\ c)\left(\frac{\Delta s}{s} + \xi_2\right) + \big(1 + g(s;\ c)\big)\frac{\Delta c}{c} + \eta_2, \tag{5}$$

and formulas (4.13) and (4.14) imply that

$$|\xi_1|, |\xi_2| \leqslant 2^{-t}, \qquad |\eta_1| \leqslant \left(\frac{3}{2} + T\right)2^{-t}, \quad |\eta_2| \leqslant \left(\frac{7}{2} + T\right)2^{-t}.$$

The error propagation factor $g(s;\ c)$ is always greater than 1. It may cause a distinct "expansion" of the error $\Delta s/s$, particularly for s close to c. The greatest loss of accuracy takes place in the first step of the WR-algorithm, since we have then $s = a_{p+1}$, $c = a_p$ (see (3.1)). For sufficiently great p, the factor $g(a_{p+1};\ a_p)$ assumes arbitrarily large values.

This is not to understand that in this case errors (4) and (5) are arbitrarily large, since formula (4) gives the relationship between Δy_1, Δs and Δc only then, when the linearisation conditions are satisfied. The linearisation condition will not be satisfied if in the total differential (1) the derivatives are unbounded. For example, for $m = 8$ and $k' = .8$ the ODRA 1204 computer gave no values of $g(a_{p+1}; a_p)$ since a floating-point overflow was signaled (elements a_{p+1} and a_p had a common representation). It was, however, the set RW_m (see section 3) numerically obtained, the elements of which are charged with a great error — only one digit is significant — but yet it was not so great as it would follow from the linear estimation of the error (1). This last fact does not reduce the importance of expression (1) for the analysis of sufficiently little disturbances

For the given task and sufficiently strong arithmetics, the first differential in expression (1) describes sufficiently true the dependence of δy on Δs and Δc.

Expressions (4) and (5) serve above all to the comparison of error estimations in algorithms WR and JR and, therefore, the assumption of the strong arithmetics is not an essential restriction.

Let us now investigate the total error of the element $s_{jm} \in RW_m$. Let

$$w_{0jm}, w_{1jm}, \ldots, w_{p-1,jm} \tag{6}$$

be the successive elements which are used in stage II of the WR-algorithm to evaluate the element $w_{pjm} = s_{jm}$. Hence w_{ijm} is one of the elements s_{ln} for $n = 2^i$. We write

$$g_{ijm} = g(w_{ijm};\ a_{p-i}), \tag{7}$$

$$i = 0, 1, \ldots, p-1;\ m = 2^p;\ j = 1, 2, \ldots, m.$$

From (4) and (5) we have

$$\frac{\Delta w_{i+1,jm}}{w_{i+1,jm}} \cong \varepsilon_{ijm} g_{ijm}\left(\frac{\Delta w_{ijm}}{w_{ijm}} + \xi_{ijm}\right) + (1 - \varepsilon_{ijm} g_{ijm}) \frac{\Delta a_{p-i}}{a_{p-i}} + \eta_{i+1,jm}, \tag{8}$$

where $|\xi_{ijm}| \leqslant 2^{-t}$, $|\eta_{ijm}| \leqslant (7/2 + T)2^{-t}$, and ε_{ijm} is equal to $+1$ or -1 depending on the use of formula (4.10) or formula (4.11), respectively, to the evaluation of the element $w_{i+1,jm}$. To simplify the notation we will use instead of w_{ijm}, g_{ijm}, ε_{ijm}, ξ_{ijm} and η_{ijm} the symbols w_i, g_i, ε_i, ξ_i and η_i, respectively. The application of formula (8) to successive elements of the set (6) leads to the following expression for the total error of the element $w_{pjm} = s_{jm} = r_{jm} \in RW_m$:

$$\frac{\Delta r_{jm}}{r_{jm}} \cong G_{0jm}\xi_0 + \sum_{i=1}^{p-1} (\xi_i + \eta_i) G_{ijm} + \eta_p + G_{0jm} \frac{\Delta a_{p+1}}{a_{p+1}} + \tag{9}$$

$$+ \sum_{i=0}^{p-1} (G_{i+1,jm} - G_{ijm}) \frac{\Delta a_{p-i}}{a_{p-i}},$$

where

$$G_{ijm} \equiv G_{ijm}(k') = \prod_{l=i}^{p-1} \varepsilon_{ljm} g_{ljm}, \qquad i = 0, 1, \ldots, p-1, \tag{10}$$

$$G_{pjm} \equiv 1,$$

$$|\xi_i| \leqslant 2^{-t}, \qquad |\eta_i| \leqslant (3.5+T)2^{-t}.$$

It is, therefore, seen that the total error of the element r_{jm} depends above all on the magnitudes of the factors G_{ijm} and on the relative errors of elements a_i. From (7) and (10) it follows that $|G_{0jm}| > |G_{1jm}| > \ldots > |G_{pjm}| = 1$. Therefore, the estimation of the error (9) can be very rough. Successive steps of stage II of the WR-algorithm do not smooth out the loss of accuracy in the first step, for all g_k are greater than 1.

The above considerations lead to the following

COROLLARY 1. *Though one step of the WR-algorithm is stable (and for a "good" arithmetics — very exact), nevertheless, the values of propagation factors of the error G_{ijm} being large, there may occur a significant increase of the error. Elements r_{jm} will then be evaluated with a large relative error.*

6. Error propagation in the JR-algorithm. Total error. The mechanism of error propagation of the preceding step acts in the JR-algorithm similarly as in the WR-algorithm. Let r, z_1, z_2 have the same meaning as in sections 2 and 4 (see (2.6)). Let the element r be determined with an error $\Delta r/r$, and let the parameter k' be charged with an error $\Delta k'/k'$; the case $|\Delta k'| \ll k'$ is particularly interesting. For elements z_1 and z_2, similarly as for elements y_1 and y_2 (see (5.1), (5.4) and (5.5)), the propagated error is expressed by

$$\frac{\delta z_1}{z_1} \approx d(r;\ k')\frac{\Delta r}{r} + \frac{1}{2}\big(1 - c(r;\ k')\big)\frac{\Delta k'}{k'},$$

$$\frac{\delta z_2}{z_2} \approx -d(r;\ k')\frac{\Delta r}{r} + \frac{1}{2}\big(1 + c(r;\ k')\big)\frac{\Delta k'}{k'}, \tag{1}$$

where

$$d(r;\ k') = \frac{1}{2}\frac{r}{1+r}\sqrt{\frac{1-k'^2}{r^2-k'^2}},$$

$$c(r;\ k') = \frac{r-k'^2}{1-k'^2}\sqrt{\frac{1-k'^2}{r^2-k'^2}}. \tag{2}$$

It is easy to prove that

$$d(r;\ k') > \tfrac{1}{2}\big(c(r;\ k') - 1\big) > 0, \qquad k' < r < 1. \tag{3}$$

Therefore, the leading role in expression (1) is played by the term $d(r;\ k')(\Delta r/r)$. In order to obtain the total error of the elements z_1 and z_2 it is necessary to consider at the same time the errors made and propagated. Estimates (4.16) and (4.17) imply for the total errors the expressions

$$(4)\qquad \begin{aligned} \frac{\Delta z_1}{z_1} &\cong d(r;\ k')\left(\frac{\Delta r}{r}+\xi_1\right)+\frac{1}{2}\left(1-c(r;\ k')\right)\frac{\Delta k'}{k'}+\eta_1,\\ \frac{\Delta z_2}{z_2} &\cong -d(r;\ k')\left(\frac{\Delta r}{r}+\xi_2\right)+\frac{1}{2}\left(1+c(r;\ k')\right)\frac{\Delta k'}{k'}+\eta_2, \end{aligned}$$

where

$$|\xi_1|,\ |\xi_2|\leqslant 2^{-t},\qquad |\eta_1|\leqslant (3.5+1.5T)2^{-t},\qquad |\eta_2|\leqslant (4.5+1.5T)2^{-t}.$$

In view of (3), factors of error propagation $d(r;\ k')$ determine the magnitude of the estimates of errors (4). The function $d(r;\ k')$ of variable r decreases for fixed k'. It is easy, therefore, to verify that

$$(5)\qquad \frac{1}{4}<d(r;\ k')<\frac{1}{2}\frac{\sqrt{1+k'}}{1+\sqrt{k'}}<\frac{1}{2}\qquad \text{for } r>\sqrt{k'}.$$

This may cause in some steps of the JR-algorithm the effect of "damping" of the error.

We now deal with the total error of the element r_{jm}. Let

$$(6)\qquad v_{0jm},\ v_{1jm},\ \ldots,\ v_{p-1,jm}\qquad (j=1,2,\ldots,m;\ m=2^p)$$

be the corresponding elements of the sets $R_1(k'),\ R_2(k'),\ \ldots,\ R_{2^{p-1}}(k')$ by virtue of which the JR-algorithm finds the element $v_{pjm}=r_{jm}\in RJ_m$, where $v_{ijm}\in R_n(k')$ for $n=2^i$.

Similarly as in the case of WR-algorithm (see (5.7)-(5.10)), we get for the total error of the element $r_{jm}\in RJ_m$ the expression

$$(7)\qquad \frac{\Delta r_{jm}}{r_{jm}}\cong (\varrho+\xi_{0jm})D_{0jm}+\sum_{i=1}^{p-1}(\xi_{ijm}+\eta_{ijm})D_{ijm}+\eta_{pjm}+E_{jm}\frac{\Delta k'}{k'},$$

where

$$(8)\qquad D_{ijm}\equiv D_{ijm}(k')=\prod_{l=i}^{p-1}\varepsilon_{ljm}d_{ljm}\qquad (i=0,1,\ldots,p-1),$$

$$D_{pjm}\equiv 1,$$

$$(9)\qquad d_{ijm}=d(v_{ijm};\ k'),\qquad c_{ijm}=c(v_{ijm};\ k'),$$

$$E_{jm}\equiv E_{jm}(k')=\frac{1}{2}\left[D_{0jm}+\sum_{i=0}^{p-1}(1-\varepsilon_{ijm}c_{ijm})D_{i+1,jm}\right],$$

$$|\varrho|\leqslant T\cdot 2^{-t},\qquad |\xi_{ijm}|\leqslant 2^{-t},\qquad |\eta_{ijm}|\leqslant (4.5+3.5T)2^{-t}.$$

The term ε_{ijm} is equal to $+1$ or -1 according to the use of (2.4) or (2.5), respectively, for the determination of the element $v_{i+1,jm}$.

Remark. Formula (7) is obtained from (4) if we assume that $|\Delta k'| \ll k'$, since in this case for $v_{0jm} = \sqrt{k'}$ we have

$$\frac{\Delta v_{0jm}}{v_{0jm}} \approx \frac{1}{2}\frac{\Delta k'}{k'} + \varrho.$$

The factor E_{jm} defines the influence of the error $\Delta k'$ on the final error of the element r_{jm}. If we would not take into consideration the errors produced in single steps of the algorithm, then the total error of r_{jm} will be equal to the propagated error of initial data, i.e. to the parameter k'. It is empirically stated that (see [6])

$$0 \leqslant E_{jm}(k') \leqslant 1.$$

Hence, factors of error propagation D_{ijm} determine above all the estimate of error (7). Taking into account that the function $d(r;\ k')$ is monotonic (see [3]), we see that D_{ijm} have moderate estimates. Now, if for a fixed k' the element v_{ijm} is equal to r_{1n} $(n = 2^i)$, then d_{ijm} assumes the greatest value among the possible values of factors d_{ilm} $(l \neq j)$. But $d_{i+1,jm}$ is in this case always smaller than .5. This implies the following

COROLLARY 1. *Tendencies to the expanding and damping of errors of single steps of the algorithm JR are in considerable degree equivalent. This points to the possibility of the existence of a moderate estimate of the total error of the element* r_{jm}.

In next sections we deal with close analysis of the behaviour of factors d_{ijm} and D_{ijm} and, moreover, they will be compared with analogous factors for the WR-algorithm.

7. Comparison of factors of error propagation in the algorithms WR and JR. Stability of the JR-algorithm. We assume that applying the algorithms WR and JR gives the same element of the set $R_m(k')$, i.e. $w_{pjm} = v_{pjm} = r_{jm}(k')$. We will now compare factors of the error propagation in a single step of both algorithms, i.e. the values of g_{ijm} and d_{ijm}.

THEOREM 1. *For fixed* k' $(0 < k' < 1)$ *the elements* w_{ijm} *and* v_{ijm} *(see* (5.6) *and* (6.6)) *satisfy, for* $i = 0, 1, \ldots, p-1$; $m = 2^p$ *and* $j = 1, 2, \ldots, m$, *the inequalities*

$$g_{ijm} > 2\,\frac{1+v_{ijm}}{\sqrt{1-k'^2}}\,d_{ijm}, \tag{1}$$

$$g_{ijm} > \frac{v_{ijm}}{v_{ijm}-k'^2}\sqrt{1-k'^2}\,c_{ijm}, \tag{2}$$

where g_{ijm}, c_{ijm} *and* d_{ijm} *are defined by* (5.7) *and* (6.9).

Proof. The way taken in the algorithms WR and JR is the same, i.e. if $w_{ijm} = s_{ln}$, then $v_{ijm} = r_{ln}$ $(n = 2^i)$. It is well known that

$$k' < r_{1n} < r_{2n} < \ldots < r_{nn} < 1,$$

$$\alpha_{p-i} < s_{1n} < s_{2n} < \ldots < s_{nn} < \beta_{p-i}.$$

Elements s_{ln} form the set of optimal parameters for the interval $[\alpha_{p-i}, \beta_{p-i}]$, and elements $r_{ln}(k')$ that for the interval $[k', 1]$ (let us remark that in fact elements s_{ln} are also functions of the parameter k'). Let

$$k'_{p-i} = \frac{\alpha_{p-i}}{\beta_{p-i}}.$$

Transforming the interval $[\alpha_{p-i}, \beta_{p-i}]$ into the interval $[k'_{p-i}, 1]$ we therefore have

$$r_{ln}(k'_{p-i}) = \frac{s_{ln}}{\beta_{p-i}}. \tag{3}$$

From (2.5) it follows that

$$r \equiv \frac{k'_{p-i}}{r_{ln}(k'_{p-i})} = r_{n-l+1,n}(k'_{p-i}). \tag{4}$$

Using (3) and (4) we get

$$g_{ijm} = \frac{s_{ln}}{\sqrt{s_{ln}^2 - \alpha_{p-i}^2}} = \frac{1}{\sqrt{1-r^2}}. \tag{5}$$

It easily follows from (2.8) that

$$1 > k'_p > k'_{p-1} > \ldots > k'_0 = k'. \tag{6}$$

The element $r_{ln}(k')$ is an increasing function of k' (see Theorem 2.1), which implies, in view of (4) and (6), that $r > r_{n-l+1,n}(k')$. Hence, from (5), we have

$$g_{ijm} = g(s_{ln};\ \alpha_{p-i}) > \frac{1}{\sqrt{1-r_{n-l+1,n}^2(k')}} = g\big(r_{ln}(k');\ k'\big).$$

From the definition of g_{ijm} and d_{ijm} we infer that

$$g_{ijm} > g\big(r_{ln}(k');\ k'\big) = 2\,\frac{1+v_{ijm}}{\sqrt{1-k'^2}}\,d_{ijm},$$

i.e. we have proved inequality (1). In the same way inequality (2) can be obtained, which completes the proof.

Let us now estimate the factors appearing at d_{ijm} and c_{ijm} in inequalities (1) and (2). Now, Theorem 2.1 (see (2.1) and (2.2)) implies

$$\lim_{k'\to 0+} \frac{1+v_{ijm}(k')}{\sqrt{1-k'^2}} = 1, \qquad \lim_{k'\to 1-} \frac{1+v_{ijm}(k')}{\sqrt{1-k'^2}} = \infty \tag{7}$$

and, moreover, we see that

$$\frac{1+v_{ijm}(k')}{\sqrt{1-k'^2}}$$

is an increasing function of k'. In view of (7) and Theorem 1, this implies that

The factor g_{ijm} increases for $k' \to 1^-$ to infinity. If p is sufficiently large, then g_{ijm} is large also for other values of k', for then (see (6)) k'_{p-i} is close to 1.

From this, in view of Theorem 1, we infer the following

COROLLARY 1. *The factor of error propagation in a single step of the WR-algorithm is at least two times greater than the analogous factor for the JR-algorithm, and for $k' \sim 1$ or for sufficiently large p it is incomparably greater. The factors d_{ijm} and c_{ijm} are, however, finite.*

LEMMA 1. *If $v_{ijm} = r_{ln}(k')$, $n = 2^i$, then*

$$\lim_{k' \to 0^+} d_{ijm} = \tfrac{1}{2}, \tag{8}$$

$$\lim_{k' \to 0^+} c_{ijm} = 1, \tag{9}$$

$$\lim_{k' \to 1^-} d_{ijm} = \left(4 \sin \frac{2l-1}{2n} \frac{\pi}{2}\right)^{-1}, \tag{10}$$

$$\lim_{k' \to 1^-} c_{ijm} = \frac{1}{2}\left(\frac{1}{\sin \dfrac{2l-1}{2n} \dfrac{\pi}{2}} + \sin \frac{2l-1}{2n} \frac{\pi}{2}\right). \tag{11}$$

Proof. The limit (8) follows immediately from (2.1) and (2.5). Indeed,

$$\lim_{k' \to 0^+} d_{ijm} = \frac{1}{2} \lim_{k' \to 0^+} \frac{1}{1+r_{ln}} \sqrt{\frac{1-k'^2}{1-(k'/r_{ln})^2}} = \frac{1}{2}.$$

In the same way we prove (9).

To obtain (10) we apply (2.2) and (2.3):

$$\lim_{k' \to 1^-} \frac{r_{ln}(k')-k'}{1-k'} = 1 - \lim_{k' \to 1^-} \frac{d}{dk'} r_{ln}(k') = \sin^2 \frac{2l-1}{2n} \frac{\pi}{2}. \tag{12}$$

In view of the definition of d_{ijm} this implies immediately the limit (10). Proceeding analogously we obtain (11). Indeed,

$$\lim_{k' \to 1^-} \frac{r_{ln}(k')-k'^2}{1-k'^2} = 1 - \frac{1}{4}\left(1+\cos \frac{2l-1}{2n} \pi\right).$$

In view of formula (12) this implies (11), which completes the proof.

Thus factors d_{ijm} and c_{ijm} are small for $k' \sim 0$. For $k' \sim 1$, however, some of those factors are great, in particular for

$$\frac{2l-1}{2n}\frac{\pi}{2} \sim 0.$$

Nevertheless, D_{ijm} do not increase infinitely because of the special mechanism of determining successive elements v_{ijm} (see section 6).

The total error of WR-algorithm is expressed by (5.9), and of the JR-algorithm by (6.7). The form of these expressions is similar. The estimates of ξ_i and η_i are of the same order (for the JR-algorithm they are at least one and a half times greater than for the WR-algorithm). The errors $\Delta a_{p-i}/a_{p-i}$ have estimates not smaller than the error $\Delta k'/k'$. Therefore, using Theorem 1, we obtain

COROLLARY 2. *The total error of the WR-algorithm has an estimate greater than the total error of the JR-algorithm (often incomparably greater).*

In this way the numerical predominance of the JR-algorithm over the WR-algorithm has been shown. The above considerations imply that the large values of G_{ijm} decide in the first place upon the negative properties of the WR-algorithm. Even then, however, when the values of G_{ijm} are moderate — which is the case for small k' and p — the estimate of the error of the WR-algorithm is greater than that of the error of the JR-algorithm. To empirically observed cases of failure of the process of determination of elements of the RW_m — for $k' \sim 1$ or for large p — large values of G_{ijm} correspond, i.e. a large estimation of the error. The error analysis performed has fully explained the empirically stated properties pointing to the unstability of the WR-algorithm. The analysis for the JR-algorithm points to its stability:

THEOREM 2. *For every p there exist constants t_0 and K such that for every $k' \in (0, 1)$, j ($j = 1, 2, \ldots, 2^p$) and $t > t_0$ the relative error of the element $r_{jm}(k')$, $m = 2^p$, is estimated by the expression*

$$\left|\frac{\Delta r_{jm}(k')}{r_{jm}(k')}\right| \leqslant K \cdot 2^{-t}. \tag{13}$$

Remark. Thus the JR-algorithm is stable for sufficiently strong floating-point arithmetics.

Proof. Factors d_{ijm} and c_{ijm} are continuous functions of k'. It follows from Lemma 2 that at the ends of the interval $[0, 1]$ they assume finite values. Hence, for $k' \in (0, 1)$, factors d_{ijm} and c_{ijm} are bounded. Let

$$|d_{ijm}| < K_{ijm}^{(1)} < \infty, \qquad |c_{ijm}| < K_{ijm}^{(2)} < \infty. \tag{14}$$

As we have already mentioned (see section 5), for t sufficiently large the right-hand sides of (5.4) and (5.5) give a good approximation of relative

errors. This obviously holds also for the JR-algorithm, or for formulas (6.4). Therefore, if t_0 is sufficiently large for a given p, then, for $t > t_0$, from (6.7) and (14) for $|\Delta k'/k'| < K_3 \cdot 2^{-t}$ we obtain the following estimate of the relative error for $T = 1$:

$$\left|\frac{\Delta r_{jm}}{r_{jm}}\right| < \left\{2K_1^p + 9 + 9\sum_{i=0}^{p-1} K_1^{p-i} + \frac{K_3}{2}\left[K_1^p + \sum_{i=0}^{p-1}(1+K_2)\prod_{l=i+1}^{p-1} K_1\right]\right\}2^{-t},$$

where

$$K_1 = \max_{i,j} K_{ijm}^{(1)}, \qquad K_2 = \max_{i,j} K_{ijm}^{(2)}.$$

The constants K_1, K_2 and K_3 depend only on p. Thus we have proved that for any p there exist constants K and t_0 such that in the t-digit arithmetics ($t > t_0$) the relative error of the element r_{jm} has the estimate (13) independent of k' and j, which completes the proof.

Theorem 2 does not imply the values of the constant K. In view of (10) we may suppose that the estimate (13) will be large. We have empirically examined the behaviour of D_{ijm}, and examples are given in the following table:

	$k' = .8$				$k' = .1$			
i	D_{i18}	D_{i28}	D_{i38}	D_{i48}	D_{i18}	D_{i28}	D_{i38}	D_{i48}
0	.0244	.0696	.1040	.1226	.0312	.0823	.117	.1236
1	.0691	.1965	.2938	.3463	.0783	.2066	.2803	.3104
2	.2550	.3012	.4502	1.278	.2630	.3447	.4676	1.042

We have observed that the greatest values of $|D_{ijm}|$ are assumed for $i = p-1$ and $j = 2^i$, i.e. for this element r_{jm} which is the closest to $\sqrt{k'}$. Just for this element we come close in the last but one step of the algorithm to the left end of the interval $[k', 1]$ (see (10)). Most of the factors D_{ijm} assume moderate, often very small values. For example, for $p = 10$ ($m = 1024$) and $j = 512$ for $k' = .9999$ we have obtained the following values of $|D_{ijm}|$ ($i = 0, 1, \ldots, p-1$): .001, .0028, .102, .04, .1592, .6391, 2.544, 10.17, 40.72, 162.8. It is therefore to expect that the constant K is not too large for moderate values of p. The JR-algorithm determines, therefore, with great accuracy the elements r_{jm}.

Acknowledgement. We give sincere thanks to Dr. A. Kiełbasiński for valuable advices and hints without which this paper would not appear.

References

[1] W. Kahan, *A survey of error analysis*, IFIP Congress 1971, Ljubljana.
[2] M. Oberhettinger, *Anwendung der elliptischen Fuktionen in Physik und Technik*, Berlin 1949.
[3] R. S. Varga, *Matrix iterative analysis*, Englewood Cliffs 1962.
[4] E. L. Wachspress, *Iterative solution of elliptic systems*, Englewood Cliffs 1966.
[5] J. H. Wilkinson, *Rounding errors in algebraic processes*, London 1963.
[6] K. Ziętak, *Funkcje wymierne z metody naprzemiennych kierunków*, Doctoral thesis, University of Wrocław, 1972.
[7] — *Construction and features of the optimum rational function used in the ADI-method*, Zastosow. Matem. 14 (1974), p. 277-311.

MATHEMATICAL INSTITUTE
UNIVERSITY OF WROCŁAW
50-384 WROCŁAW

Received on 2. 5. 1973

KRYSTYNA ZIĘTAK (Wrocław)

BADANIE ALGORYTMÓW WYZNACZANIA OPTYMALNEJ FUNKCJI WYMIERNEJ Z METODY NAPRZEMIENNYCH KIERUNKÓW

STRESZCZENIE

W pracy podane są wyniki porównania efektywności algorytmu Wachspressa i algorytmu *JR* (patrz [7]). Teoretyczna analiza błędów zaokrągleń wytworzonych i przenoszonych w kolejnych krokach dla obu algorytmów poparta jest przykładami numerycznymi obliczeń wykonanych na m.c. ODRA 1204. Okazuje się, że błąd całkowity algorytmu Wachspressa ma większe oszacowanie niż błąd całkowity algorytmu *JR*.
