数式処理 *J.JSSAC (1998)*
Vol. 6, No. 3, pp. 30 - 46

論文

Univariate Factor Separation and Separation of Multiple/Close Root Factors*

Yu-ichi OZAKI[†)] and Tateaki SASAKI[‡)]
[†)]Master's Program in Science and Engineering
[‡)]Institute of Mathematics & Venture Business Laboratory
University of Tsukuba
Tsukuba-shi, Ibaraki 305, Japan

(RECEIVED 1997/11/4 REVISED 1998/5/8)

Abstract

Given univariate polynomials F, G_0 and H_0 such that $F = G_0H_0 + \mathit{\Delta}_0$, $\|\mathit{\Delta}_0\|/\|F\| = \varepsilon_0 \ll 1$, we consider calculating polynomials G_1 and H_1, such that $F = G_1H_1 + \mathit{\Delta}_1, \|\mathit{\Delta}_1\|/\|F\| = \varepsilon_1 \ll \varepsilon_0$, where $\|P\|$ denotes a norm of a polynomial P. We call this operation univariate *factor separation.* We give a quadratically convergent algorithm to calculate G_1 and H_1. Furthermore, we derive a condition of convergence of the factor separation algorithm and discuss the accuracy of factor separated.

We apply the factor separation to separating multiple/close root factors accurately in two ways. In the first way, we perform the approximate square-free decomposition of F with low accuracy, obtaining multiple/close root factors crudely, then apply the factor separation algorithm. In the second way, we solve the equation $F(x) = 0$ numerically, obtaining approximate roots among which the multiple/close roots are of low accuracies. We combine these multiple/close root factors to a polynomial and use it as an initial factor for the factor separation algorithm.

1 Introduction

Recently, many researchers are getting interested in approximate algebraic computation with floating-point arithmetic [4, 5, 8] where algebraic operations are treated approximately

*Work supported in part by Japanese Ministry of Education, Science and Culture under Grants 06558037, 09308008 and Venture Business Laboratory, Univ. of Tsukuba.

Key words: approximate algebraic computation, factor separation of univariate polynomial, multiple/close root problem

with small "error terms". In approximate algebraic computation, both algebraic and numeric algorithms are combined to each other, and the algorithm to be described in this paper is also of such a kind.

Given a univariate polynomial $F(x)$ and its approximate factors $G_0(x)$ and $H_0(x)$ such that $F(x) = G_0(x)H_0(x) + \Delta_0(x),\ \|\Delta_0\|/\|F\| = \varepsilon_0 \ll 1$, we consider calculating polynomials $G_1(x)$ and $H_1(x)$ such that $F(x) = G_1(x)H_1(x) + \Delta_1(x),\ \|\Delta_1\|/\|F\| = \varepsilon_1 \ll \varepsilon_0$, where $\|P\|$ denotes a norm of a polynomial P. We call this operation *factor separation of accuracy* ε_1. Similar operation is treated in [10], where the authors start from the rational Hermite interpolation and derive an iteration formula for calculating G_1 and H_1. We review the iteration formula in [10]briefly in the text. In this paper, we derive a similar but different iteration formula from the viewpoint of the Hensel construction.

After defining necessary concepts in **2**, we derive a factor separation algorithm in **3**. In **4**, we derive a condition of convergence of the algorithm and show that the algorithm converges quadratically. Furthermore, we show that the accuracies of factors separated become often much worse than the separation accuracy ε. Factor separation is a simple and basic operation, hence it will be applied to many kinds of algebraic and numeric operations. In **5**, we propose two methods of applying the factor separation algorithm to separating multiple/close root factors of a given univariate polynomial accurately. The proposed methods are experimented by many examples in **6**, so as to check the robustness and effectiveness of the factor separation algorithm and reveal its weak points. These experiments show that, although numerical accuracy often decreases largely in the process of factor separation, the algorithm works pretty well so long as we have enough numerical precision.

2 Notations and definitions

Throughout this paper, $F(x)$, $\Delta(x)$, etc. denote univariate polynomials in a variable x with coefficients in $\mathbf{C}$, the complex numbers. (The symbol Δ is used to express small "error terms".) The coefficients are actually represented by floating-point numbers. Let $F(x) = f_\lambda x^\lambda + f_{\lambda-1}x^{\lambda-1} + \cdots + f_0,\ f_\lambda \neq 0$. $\deg(F)$ and $\mathrm{lc}(F)$ denote the degree and the leading coefficient of F, respectively: $\deg(F) = \lambda$ and $\mathrm{lc}(F) = f_\lambda$. A polynomial F is called *monic* if $f_\lambda = 1$. $\mathrm{quo}(F, G)$ and $\mathrm{rem}(F, G)$ denote the quotient and the remainder of F divided by G.

Definition 1
(norm of polynomial) *As a norm of a polynomial F, expressed by $\|F\|$, we adopt the infinity norm, i.e., $\|F\| = \max\{|f_\lambda|, |f_{\lambda-1}|, \cdots, |f_0|\}$. Furthermore, $\|F\|_2$ denotes the square norm of F, i.e., $\|F\|_2 = (|f_\lambda|^2 + |f_{\lambda-1}|^2 + \cdots + |f_0|^2)^{1/2}$.*

Definition 2

(regular polynomial) *A polynomial $F(x)$ is called regular if its roots are on a disc of radius* 1 *placed at the origin.*

Definition 3

(square-free) *A polynomial $F(x)$ is called square-free if it has no multiple root factor.*

Definition 4

(approximate square-free decomposition) *Let ε be a small positive number, $0 < \varepsilon \ll 1$. If a polynomial $F(x)$ is decomposed as*

$$F(x) = Q_1(x)Q_2^2(x)\cdots Q_m^m(x) + \varDelta(x), \quad \|\varDelta\|/\|F\| = \varepsilon, \tag{2.1}$$

where each $Q_i(x)$ is square-free, then we call this decomposition approximate square-free decomposition of accuracy ε.

Remark 1

The norm of a monic regular polynomial F may become pretty large. The most extreme case is that all the roots are at a point of the unit circle, so $\|F\| = {}_{\lambda}C_{\lfloor \lambda/2 \rfloor}$ in this case. However, in most practical cases, $\|F\|$ is not much greater than 1.

Remark 2

Let $\hat{\alpha}$ be a root of $F(x) = 0$, then

$$|\hat{\alpha}| < 1 + \max\{|f_{\lambda-1}|, |f_{\lambda-2}|, \ldots, |f_0|\}/|f_\lambda|. \tag{2.2}$$

Thus, in practical computation, we may adopt the condition $\|F\|/|f_\lambda| \simeq 1$ as an approximate criterion of regularity.

Remark 3

The polynomial $Q_i^i(x)$ in (2.1) is the product of all the i-multiple/close root factors of $F(x)$, of mutual distance $\le \delta$, where $\delta \simeq \sqrt{\varepsilon}$ so long as $F(x)$ is regular and there is no special relation among its coefficients, see [7].

3 Iteration formula for factor separation

Let $F(x)$ be a given monic regular polynomial which is factored as

$$F(x) = G(x)H(x), \quad \deg(G) = \mu,\ \deg(H) = \nu, \tag{3.1}$$

where G and H are also monic. Although the true factors G and H are unknown, suppose that we have approximate factors G_i and H_i such that

$$\begin{cases} F(x) = G_i(x)H_i(x) + \varDelta_i(x), & \|\varDelta_i\|/\|F\| \ll 1, \\ \mathrm{lc}(G_i) = \mathrm{lc}(H_i) = 1. \end{cases} \tag{3.2}$$

We assume that G_i and H_i have no mutually close root of mutual distance $< \delta,\ 0 < \delta \ll 1$.

Now, we derive an iteration formula. Putting

$$G = G_i + \varDelta_{G_i}, \quad H = H_i + \varDelta_{H_i}, \tag{3.3}$$

and substituting these for G and H in (3.1), we obtain

$$\begin{aligned} \varDelta_i = F - G_iH_i &= (G_i + \varDelta_{G_i})(H_i + \varDelta_{H_i}) - G_iH_i \\ &= \varDelta_{H_i}G_i + \varDelta_{G_i}H_i + \varDelta_{G_i}\varDelta_{H_i}. \end{aligned} \tag{3.4}$$

Assuming that $\max\{\|\varDelta_{G_i}\|/\|G_i\|,\ \|\varDelta_{H_i}\|/\|H_i\|\} = \varepsilon_i \ll 1$, we neglect the last term in (3.4) and determine polynomials u_i and v_i to satisfy

$$\varDelta_i = u_iG_i + v_iH_i, \quad \deg(u_i) < \deg(H_i), \quad \deg(v_i) < \deg(G_i). \tag{3.5}$$

Thus, we determine better approximations G_{i+1} and H_{i+1} as follows.

$$G_{i+1} = G_i + v_i, \quad H_{i+1} = H_i + u_i. \tag{3.6}$$

Note that, because of the degree conditions on u_i and v_i, we have

$$\mathrm{lc}(G_{i+1}) = \mathrm{lc}(G_i) = 1, \quad \mathrm{lc}(H_{i+1}) = \mathrm{lc}(H_i) = 1.$$

The corrections u_i and v_i are calculated as follows. Since G_i and H_i are relatively prime by the assumption, the extended Euclidean algorithm allows us to calculate polynomials A and B satisfying

$$\begin{cases} AG_i + BH_i = 1, \\ \deg(A) < \deg(H_i), \quad \deg(B) < \deg(G_i). \end{cases} \tag{3.7}$$

Dividing $\varDelta_iA$ and $\varDelta_iB$ by H_i and G_i, respectively, we calculate u_i and v_i as

$$\begin{cases} \varDelta_iA = Q_AH_i + u_i, \quad u_i = \mathrm{rem}(\varDelta_iA,\ H_i), \\ \varDelta_iB = Q_BG_i + v_i, \quad v_i = \mathrm{rem}(\varDelta_iB,\ G_i). \end{cases} \tag{3.8}$$

We can show that u_i and v_i in (3.8) satisfy (3.5) as follows. Multiplying $\varDelta_i$ to the first equality in (3.7) and using decompositions in (3.8), we have

$$\begin{aligned} \varDelta_i &= \varDelta_i\cdot(AG_i + BH_i) \\ &= u_iG_i + v_iH_i + (Q_A + Q_B)G_iH_i. \end{aligned}$$

Since $\deg(\varDelta_i) < \deg(G_iH_i)$, we see $Q_A + Q_B = 0$ and we obtain (3.5).

The factor separation is done as follows. We first determine initial approximate factors $G_0(x)$ and $H_0(x)$ suitably and set the required separation accuracy ε_c. Then, we apply iteration formula (3.6) with (3.7) and (3.8) to $G_0(x)$ and $H_0(x)$. If we have $G_i(x)$ and

$H_i(x)$ such that $\|\varDelta_i\|/\|F\| = \varepsilon < \varepsilon_c$ then we stop the procedure. We distinguish ε_c from ε, by calling it as *cutoff separation accuracy.*

Now, we briefly mention on [10]in which the authors want to determine a factor X of a polynomial F by giving an approximate factor X_0 such that $F = Q_0X_0 + \varDelta_0$, $\|\varDelta_0\|/\|F\| = \varepsilon_0 \ll 1$. They first determine Q_0 and $\varDelta_0$ as $Q_0 = \mathrm{quo}(F, X_0)$ and $\varDelta_0 = \mathrm{rem}(F, X_0)$. Then, they calculate polynomials Q_1 and X_1 satisfying $X_1Q_0 + Q_1X_0 = \varDelta_0$. (They propose to use the Chinese remainder algorithm, instead of the extended Euclidean algorithm, to calculate Q_1 and X_1.) With Q_1 and X_1 as initial polynomials, they iteratively calculate Q_i and X_i $(i = 2, 3, \ldots)$, satisfying $X_i(Q_0 + Q_{i-1}) + Q_iX_0 = \varDelta_0$. Then, X_i converges to X quadratically, so long as $\varepsilon_0 \ll 1$. Note that, in this formula, $\deg(X_i) > \deg(\varDelta_0)$ and X_0 and $\varDelta_0$ are not updated.

4 Analysis of convergence property

Let $(P_3, P_4, \cdots, P_{k_i})$ be a polynomial remainder sequence generated by $P_1 = G_i$ and $P_2 = H_i$. Since G_i and H_i are relatively prime, we assume that $P_{k_i} = \text{constant}$ $(\neq 0)$. The extended Euclidean algorithm calculates the sequences $(A_3, A_4, \cdots, A_{k_i})$ and $(B_3, B_4, \cdots, B_{k_i})$ such that

$$\begin{cases} A_jG_i + B_jH_i = P_j, \quad j = 3, 4, \cdots, k_i, \\ \deg(A_j) < \deg(H_i) - \deg(P_j), \quad \deg(B_j) < \deg(G_i) - \deg(P_j). \end{cases} \tag{4.1}$$

We assume that each P_j $(3 \le j \le k_i)$ is normalized as

$$\max\{\|A_j\|, \|B_j\|\} = 1. \tag{4.2}$$

Remark 4

According to the analysis in [7, 9],$|P_{k_i}|$ becomes very small if G_i and H_i have mutually close roots; if they have m mutually close roots of mutual distance $\le \delta$ then $|P_{k_i}|$ becomes less than about δ^{2m-1}. Even if G_i and H_i have no mutually close root, $|P_{k_i}|$ may become considerably small if $\deg(F)$ is large; $|P_{k_i}|$ is proportional to the resultant of G_i and H_i and we have $\mathrm{resultant}(G_i, H_i) = \prod_m \prod_n (\alpha_m - \beta_n)$, where $G_i = \prod_m (x - \alpha_m)$ and $H_i = \prod_n (x - \beta_n)$.

Let $(\varDelta_{G_i}, \varDelta_{H_i})$ and (u_i, v_i) be defined as in (3.3) and (3.5), respectively. Equalities in (3.4) and (3.5) give us

$$\varDelta_{H_i}G_i + \varDelta_{G_i}H_i + \varDelta_{G_i}\varDelta_{H_i} = u_iG_i + v_iH_i.$$

Using (3.3) and G_{i+1} and H_{i+1} in (3.6), we can rewrite this equality as follows.

$$-\varDelta_{G_i}\varDelta_{H_i} \quad = \quad (\varDelta_{H_i} - u_i)G_i + (\varDelta_{G_i} - v_i)H_i$$

$$\begin{aligned} &= (H - H_{i+1})G_i + (G - G_{i+1})H_i \\ &= \varDelta_{H_{i+1}} G_i + \varDelta_{G_{i+1}} H_i. \end{aligned} \tag{4.3}$$

On the other hand, the first relation in (4.1) with $j = k_i$ gives

$$1 = (A_{k_i} G_i + B_{k_i} H_i)/P_{k_i}.$$

Multiplying $-\varDelta_{G_i}\varDelta_{H_i}$ to this equality, we obtain

$$-\varDelta_{G_i}\varDelta_{H_i} = -(\varDelta_{G_i}\varDelta_{H_i} A_{k_i}/P_{k_i})G_i - (\varDelta_{G_i}\varDelta_{H_i} B_{k_i}/P_{k_i})H_i. \tag{4.4}$$

We note that $\varDelta_{G_{i+1}}$ and $\varDelta_{H_{i+1}}$ satisfying (4.3) exist and are unique if we impose conditions $\deg(\varDelta_{H_{i+1}}) < \deg(H_i)$ and $\deg(\varDelta_{G_{i+1}}) < \deg(G_i)$. Hence, from (4.3) and (4.4), we find

$$\begin{cases} \varDelta_{G_{i+1}} = -\mathrm{rem}(\varDelta_{G_i}\varDelta_{H_i} B_{k_i}, G_i)/P_{k_i}, \\ \varDelta_{H_{i+1}} = -\mathrm{rem}(\varDelta_{G_i}\varDelta_{H_i} A_{k_i}, H_i)/P_{k_i}. \end{cases} \tag{4.5}$$

4.1 Condition of convergence

Equations in (4.5) allow us to bound $\|\varDelta_{G_{i+1}}\|$ and $\|\varDelta_{H_{i+1}}\|$ in terms of the product of $\|\varDelta_{G_i}\|$, $\|\varDelta_{H_i}\|$ and $C/|P_{k_i}|$, where C is a number which bounds the increasing ratio of the coefficients in the products and remainder computations in (4.5). That is, we put

$$\begin{gathered} \max\{\|\varDelta_{G_{i+1}}\|,\ \|\varDelta_{H_{i+1}}\|\} \le C\,\|\varDelta_{G_i}\|\cdot\|\varDelta_{H_i}\|/|P_{k_i}|, \\ \text{(an upper-bound of } C \text{ is determined below).} \end{gathered} \tag{4.6}$$

¿From (4.6), we have $\|\varDelta_{G_{i+1}}\|/\|\varDelta_{G_i}\| \le C\|\varDelta_{H_i}\|/|P_{k_i}|$ and $\|\varDelta_{H_{i+1}}\|/\|\varDelta_{H_i}\| \le C\|\varDelta_{G_i}\|/|P_{k_i}|$. The iteration will converge only when we have $\|\varDelta_{G_{i+1}}\|/\|\varDelta_{G_i}\| < 1$ and $\|\varDelta_{H_{i+1}}\|/\|\varDelta_{H_i}\| < 1$. Therefore, the convergence condition for our iteration procedure is that the following inequalities are satisfied for each i.

$$\|\varDelta_{G_i}\| < |P_{k_i}|/C \quad \text{and} \quad \|\varDelta_{H_i}\| < |P_{k_i}|/C\,. \tag{4.7}$$

Now, we determine an upper-bound of C.

Lemma 5
Let univariate polynomials $P(x)$ and $Q(x)$ be represented as

$$\begin{cases} P(x) = p_m x^m + p_{m-1}x^{m-1} + \cdots + p_0, \\ Q(x) = q_n x^n + q_{n-1}x^{n-1} + \cdots + q_0, \quad n \le m. \end{cases} \tag{4.8}$$

Then, the norm of the remainder $\mathrm{rem}(P, Q)$ *is bounded as follows.*

$$\|\mathrm{rem}(P,Q)\| \le \left|D/q_n^{m-n+1}\right|, \tag{4.9}$$

$$D^2 = \max_{i=1}^{n}\left\{(p_m^2 + \cdots + p_n^2 + p_{n-i}^2)\cdot\textstyle\prod_{j=0}^{m-n}(q_n^2 + \cdots + q_{n-j}^2 + q_{n-j-i}^2)\right\}. \tag{4.10}$$

Proof The remainder can be expressed by determinants as follows, see [1].(The dashed line in the matrix below is to show that the determinant is an augmented one.)

$$q_n^{m-n+1}\mathrm{rem}(P,Q) = \left| \begin{array}{ccc|ccc} q_n & \cdots & q_{2n-m} & x^{n-1}q_{2n-m-1} & \cdots & \\ & \ddots & \vdots & \vdots & \ddots & \\ & & q_n & x^{n-1}q_{n-1} & \cdots & x^0 q_0 \\ p_m & \cdots & p_n & x^{n-1}p_{n-1} & \cdots & x^0 p_0 \end{array} \right| . \tag{4.11}$$

Here, the augmented determinant denotes the sum of n determinants each of which is constructed by attaching one of the right n columns to the left $m-n+1$ columns. Applying Hadamard's inequality to the above determinants, we obtain the lemma. ∎

Proposition 6
The number C in (4.6) *is bounded as follows.*

$$C < \sqrt{\lambda}\, \min\{\mu,\nu\}\, \max\{\mu\|G_i\|_2^{\lambda-2},\, \nu\|H_i\|_2^{\lambda-2}\}. \tag{4.12}$$

Proof Noting that $\deg(\Delta_{G_i}) < \deg(G_i) = \mu$, $\deg(\Delta_{H_i}) < \deg(H_i) = \nu$, $\mu+\nu=\lambda$, and normalization of A_{k_i} and B_{k_i}, we see

$$\begin{array}{ll} \deg(\Delta_{G_i}\Delta_{H_i}B_{k_i}) \le \lambda+\mu-3, & \|\Delta_{G_i}\Delta_{H_i}B_{k_i}\| \le \|\Delta_{G_i}\|\,\|\Delta_{H_i}\|\,\min\{\mu,\nu\}\mu, \\ \deg(\Delta_{G_i}\Delta_{H_i}A_{k_i}) \le \lambda+\nu-3, & \|\Delta_{G_i}\Delta_{H_i}A_{k_i}\| \le \|\Delta_{G_i}\|\,\|\Delta_{H_i}\|\,\min\{\mu,\nu\}\nu. \end{array}$$

Substituting $\Delta_{G_i}\Delta_{H_i}B_{k_i}$ and G_i, and $\Delta_{G_i}\Delta_{H_i}A_{k_i}$ and H_i for P and Q in Lemma 1, and bounding the p-factor and q-factor in (3.10) as

$$\begin{array}{rcl} (p_m^2+\cdots+p_n^2+p_{n-i}^2) & \le & (m-n+2)\,\|P\|^2, \\ (q_n^2+\cdots+q_{n-j}^2+q_{n-j-i}^2) & \le & (q_n^2+\cdots+q_{n-j}^2+\cdots+q_0^2), \end{array}$$

we obtain the upper-bound in (4.12). ∎

Remark 5
The above upper-bound of C is too large an over-estimation. However, if $\max\{\|G_i\|_2, \|H_i\|_2\} \gg 1$ *then C becomes inevitably large, although the dependence of C on $\|G_i\|_2$ and $\|H_i\|_2$ is not so strong actually.*

When $\max\{\|G_i\|, \|H_i\|\} \gg 1$, *we can apply a transformation $x \to cx, c > 1$, making G_i and H_i satisfy* $\max\{\|G_i\|, \|H_i\|\} = 1$. *Then, C will not depend on G_i and H_i so strongly. However, in this case, $|P_{k_i}|$ is made small by this transformation.*

4.2 Order of convergence

Next, let us consider the convergence order. As usual, we consider only the local convergence and not the global convergence, which means that G_i and H_i are assumed to be enough close to G and H, respectively.

Similarly to (4.1) and (4.2), we have polynomials A', B' and P'_k (=constant) such that

$$\begin{cases} A'G + B'H = P'_k, \quad \max\{\|A'\|, \|B'\|\} = 1, \\ \deg(A') < \deg(H), \quad \deg(B') < \deg(G). \end{cases} \tag{4.13}$$

Since we are discussing the local convergence, we can assume that $|P_{k_i}| \simeq |P'_k|$. Then, (4.6) gives us the following inequality.

$$\begin{aligned} \frac{\max\{\|\Delta_{G_{i+1}}\|, \|\Delta_{H_{i+1}}\|\}}{\max\{\|\Delta_{G_i}\|, \|\Delta_{H_i}\|\}^2} &\le \frac{C\|\Delta_{G_i}\|\cdot\|\Delta_{H_i}\|/|P_{k_i}|}{\max\{\|\Delta_{G_i}\|, \|\Delta_{H_i}\|\}^2} \\ &\le \frac{C}{|P_{k_i}|} \simeq \frac{C}{|P'_k|}\,. \end{aligned} \tag{4.14}$$

Therefore, the above iteration converges quadratically.

4.3 Accuracy of factor separated

According to (3.3), the accuracies of G_i and H_i are given by $\|\Delta_{G_i}\|/\|G_i\|$ and $\|\Delta_{H_i}\|/\|H_i\|$, respectively, which are nearly equal to $\|v_i\|/\|G_i\|$ and $\|u_i\|/\|H_i\|$.

It should be emphasized that the accuracy of a separated factor G_i is not the same as the separation accuracy $\|\Delta_i\|/\|F\|$ but the former is often much larger than the latter. Comparing (3.7) and (4.1), and noting the normalization in (4.2), we see

$$A = A_{k_i}/|P_{k_i}|, \quad B = B_{k_i}/|P_{k_i}|. \tag{4.15}$$

Substituting the right-hand side expressions for A and B in (3.8), we find that

$$\begin{cases} u_i = \mathrm{rem}(\Delta_i A_{k_i}, H_i)/|P_{k_i}|, \\ v_i = \mathrm{rem}(\Delta_i B_{k_i}, G_i)/|P_{k_i}|. \end{cases} \tag{4.16}$$

Proposition 7
The norms $\|u_i\|$ and $\|v_i\|$ are bounded as

$$\max\{\|u_i\|, \|v_i\|\} < \sqrt{\lambda}\,\max\{\mu\|G_i\|_2^{\lambda-1}, \nu\|H_i\|_2^{\lambda-1}\}\,\frac{\|\Delta_i\|}{|P_{k_i}|}. \tag{4.17}$$

Proof Applying Lemma 1 to the remainders in (4.16), and performing a similar evaluation as in Prop. 1, we obtain the proposition. ▮

Although the above upper-bound in Prop. 2 is too large an over-estimation, the equations in (4.16) show clearly that $\|u_i\|$ and $\|v_i\|$ become larger than $\|\Delta_i\|$ by at least about a factor of $1/|P_{k_i}|$. Hence, if G_i and H_i have mutually close roots then accuracies of G_i and H_i are worse than the separation accuracy $\|\Delta_i\|$ by a factor of about $|P_{k_i}|$. We will convince ourselves of this by examples below.

5 Separation of multiple/close root factors

If we solve a univariate algebraic equation numerically, we can calculate single roots to accuracy about ε_M, with ε_M the machine epsilon, while m multiple roots can be calculated only to accuracy about $\sqrt[m]{\varepsilon_M}$ and accuracy of close roots also decreases largely, which we call the *multiple/close root problem.* This problem is inevitable so long as we employ floating-point arithmetic. The reason is as follows. Let $F(x)$ have m multiple roots at $x = r$, hence $F(x) = (x-r)^m Q(x)$. Then, for a number δ s.t. $|\delta| \le \sqrt[m]{\varepsilon_M}$, we have $F(r+\delta) = \delta^m Q(r+\delta)$, which means that all the numerical accuracy is lost during the evaluation of $F(r+\delta)$. Since the evaluation of $F(x)$ at $x \simeq r$ is necessary for the root computation, m multiple roots cannot be calculated more accurately than about $\sqrt[m]{\varepsilon_M}$. Next, let $F(x)$ have m close roots around $x = r$, hence $F(x) = (x-r-\delta_1)\cdots(x-r-\delta_m)Q(x)$, with $|\delta_i| \le \bar{\delta}$ $(i = 1,\ldots,m)$. Then, for a number δ s.t. $|\delta| \le \bar{\delta}$, we have $|F(r+\delta)| = |(\delta-\delta_1)\cdots(\delta-\delta_m)Q(r+\delta)| \le 2^m\bar{\delta}^m|Q(r+\delta)|$. This means that, in the $\bar{\delta}$-neighborhood of $x = r$, the numerical accuracy is lost to about $\bar{\delta}^m$ in the evaluation of $F(r+\delta)$, hence the close roots of mutual distance about $\bar{\delta}$ can be calculated only to accuracy about $\varepsilon_M/\bar{\delta}^m$.

One practical method to attack the multiple/close root problem is to separate a factor $G(x)$ accurately which contains all the multiple/close roots around $x = r$ and no other root, then we may solve $G(x) = 0$ by a suitable method (for example, by moving the origin to $x \simeq r$ and scaling up the x-axis). We apply the factor separation algorithm to determining $G(x)$ accurately. Only one problem in this application is how to determine the initial factors $G_0(x)$ and $H_0(x)$, and we give two methods to determine them below.

5.1 Using approximate square-free decomposition

Applying the approximate square-free decomposition algorithm to $F(x)$, we can calculate polynomials $Q_1(x), Q_2(x), \ldots, Q_l(x)$ such that

$$F(x) = Q_1(x)Q_2^2(x)\cdots Q_l^l(x) + \varDelta(x), \quad \|\varDelta\|/\|F\| = \varepsilon \ll 1. \tag{5.1}$$

Suppose $F(x)$ contains close roots of mutual distance δ, $0 < \delta \ll 1$. Since $F(x)$ is regular, all the close roots of mutual distance less than about $\sqrt{\varepsilon}$ are treated as approximate multiple roots in (5.1), see [7].Hence, each $Q_i(x)$ has no close root of mutual distance less than about $\sqrt{\varepsilon}$. Let $\deg(Q_i) = m_i$ and $r_{i,j}$ $(j = 1,\ldots,m_i)$ be the roots of $Q_i(x)$. Then, we choose the initial factors $G_0(x)$ and $H_0(x)$ as follows.

$$G_0(x) = (x - r_{i,j})^i, \quad H_0(x) = \mathrm{quo}(F(x), G_0(x)). \tag{5.2}$$

$G_0(x)$ corresponds to all the multiple/close roots around $x = r_{i,j}$. Furthermore, $G_0(x)$ and $H_0(x)$ have no mutually close root of mutual distance less than about $\sqrt{\varepsilon}$.

numerical roots	error bounds
$r_1 = 1.0 + 6.9892417174226i \times 10^{-26}$	$4.5687730195968 \times 10^{-17}$
$r_2 = 0.49999999943258 - 0.000000002469049354895 5i$	0.000000068625370507708
$r_3 = 0.49999999768121 + 0.0000000013138319584329i$	0.00000005881663943024
$r_4 = 0.2 + 1.8772219940818i \times 10^{-24}$	$1.36161454799 \times 10^{-16}$
$r_5 = 0.09999909626683 - 0.0000004200606074535 7i$	0.0000079238384679933
$r_6 = 0.100000812092505 - 0.0000005773786571105 5i$	0.0000088086156664535
$r_7 = 0.100000111696375 + 0.0000010009911173602 1i$	0.0000069488669111646
$r_8 = -0.1 - 3.4117107627346i \times 10^{-25}$	$4.9513242866064 \times 10^{-17}$
$r_9 = -0.3 + 1.2721150119997i \times 10^{-24}$	$3.6927706438334 \times 10^{-16}$
$r_{10} = -0.6 + 2.6459097352384i \times 10^{-23}$	$1.3442407502748 \times 10^{-14}$
$r_{11} = -0.7 + 1.2489317923373i \times 10^{-23}$	$7.2517218753707 \times 10^{-15}$
$r_{12} = -1.0 - 1.260411171906i \times 10^{-24}$	$1.00631954131092 \times 10^{-15}$

Table I: Numerical roots and their error bounds for the polynomial in Example 1.

5.2 Using Smith's error bound

As for the errors of approximate solutions of a univariate polynomial equation, Smith's error bound [6]is well-known and quite useful. Let r_i be an approximate solution to the i-th root of $F(x) = 0$ and δ_{r_i} be the correction to r_i in the Durand-Kerner quadratic method [2, 3],or D-K method in short, then, speaking roughly, the error bound for r_i is $\deg(F) \times |\delta_{r_i}|$. Therefore, when the D-K method converges, we can decide definitely whether or not a given approximate solution corresponds to a multiple/close root, by observing the value of Smith's error bound.

Example 1
Computation of roots of the following $F(x)$ by the D-K method.

$$F(x) = (x-1)(x-0.5)^2(x-0.2)(x-0.1)^3(x+0.1)(x+0.3)(x+0.6)(x+0.7)(x+1)$$

The numbers in Table I are approximate roots computed with double-precision floating-point arithmetic and corresponding Smith's error bounds. Noting that $\varepsilon_M \simeq 2.2 \times 10^{-16}$, $\sqrt{\varepsilon_M} \simeq 1.5 \times 10^{-8}$ and $\sqrt[3]{\varepsilon_M} \simeq 6 \times 10^{-6}$, we see from Table I definitely that r_2 and r_3 are double roots, r_5, r_6 and r_7 are triple roots, and others are single roots. Furthermore, we see that r_{10} and r_{11} are mutually close roots of mutual distance $\simeq 1/\sqrt{7.25 \times 10^{-15}/2.2 \times 10^{-16}} \simeq 0.17$, because the accuracies of two close roots of mutual distance $\bar{\delta}$ are about $\varepsilon_M/\bar{\delta}^2$.

Following the above arguments, we choose $G_0(x)$ to be the product of factors $(x - r_i)$'s, where r_i ranges over all the roots for which Smith's error bounds are much greater than

ε_M.

6 Experiments

We have mainly the following three worries about the methods proposed above. 1) Robustness: do the methods work for not only polynomials whose roots are well clustered but also polynomials whose roots are not well clustered? 2) Effectiveness: are the separated factors enough accurate? 3) How much cancellation errors occur during the computation? In order to check the robustness and effectiveness of the factor separation algorithm and reveal its weak points, we made the following two kinds of experiments.

In the first experiment, we generate 1000 polynomials of degree 15 by distributing their roots randomly in the real interval $[-1, 1]$, and apply the factor separation algorithm with initial factors determined by the approximate square-free decomposition with appropriate decomposition accuracy, as described in **5.1**. Since the average distance between two neighboring roots is $\bar{d} = 2/(\deg(F) + 1)$, we set the cutoff decomposition accuracy ε_c as $\varepsilon_c = (0.5\bar{d})^2 \simeq 0.0039$ so that all the roots of mutual distance less than $0.5\bar{d}$ are treated as close roots. Since the roots are distributed randomly, it is not always easy to distinguish some roots as close roots from others, hence this experiment is suited for checking the robustness of the algorithm. Furthermore, as we will see below, we can also check the effectiveness and reveal a weak point of the algorithm. We also generate 100 polynomials of degree 30 by distributing their roots randomly inside a unit disc, and perform a similar experiment.

In the second experiment, we give 20 polynomials of degree 12 generated artificially so that they have multiple and/or close roots, and apply the factor separation algorithm with initial factors determined by using Smith's error bounds of the roots computed numerically, as described in **5.2**. The purpose of this experiment is to show the effectiveness of the factor separation algorithm for the multiple/close root problem.

We explain two typical examples for the first kind of experiment.

Example 2
Factor separation using approximate square-free decomposition.
Let $F(x) = (x - r_1)(x - r_2)\cdots(x - r_{15})$, *where*

$$\begin{array}{lll} r_1 = 0.906978, & r_6 = 0.075609, & r_{11} = -0.517318, \\ r_2 = 0.738607, & r_7 = -0.091147, & r_{12} = -0.552766, \\ r_3 = 0.640075, & r_8 = -0.332034, & r_{13} = -0.784881, \\ r_4 = 0.506494, & r_9 = -0.335729, & r_{14} = -0.92664, \\ r_5 = 0.232769, & r_{10} = -0.346839, & r_{15} = -0.97263. \end{array} \tag{6.1}$$

Approximate square-free decomposition of $F(x)$, *with cutoff decomposition accuracy* $\varepsilon_c \simeq$

0.0039, *gives us* $F(x) \simeq Q_1(x)Q_3^3(x) + \Delta(x)$, $\|\Delta\| = 0.000180\cdots$, *where*

$$\begin{aligned} Q_3 &= x + 0.3433117570824, \\ Q_1 &= x^{12} + 0.72951672875281x^{11} - 2.2768896449443x^{10} - 1.4745190326655x^9 \\ &\quad + 1.9526688106761x^8 + 1.0520800448672x^7 - 0.78593426196072x^6 \\ &\quad - 0.31528372577108x^5 + 0.14746174585131x^4 + 0.03444041818503x^3 \\ &\quad - 0.0104776477942184x^2 - 0.00030686537793658x + 0.000050067094332961. \end{aligned}$$

Since Q_3 *is of degree 1, we need not calculate its root, and we put* $G_0(x)$ *and* $H_0(x)$ *as follows.*

$$G_0 = {Q_3}^3, \quad H_0 = \mathrm{quo}(F, G_0).$$

With these initial factors, the factor separation algorithm of cutoff separation accuracy 10^{-13} *converges at the fifth iteration, and we obtain*

$$\begin{aligned} G_5 &= x^3 + 1.014602x^2 + 0.34307969394299x + 0.038663337422453, \\ H_5 &= x^{12} + 0.74485000000001x^{11} - 2.270831553243x^{10} + \cdots. \end{aligned}$$

Since $G_0(x)$ *corresponds to roots* r_8, r_9 *and* r_{10}, *we check the accuracy of* $G_5(x)$ *as follows.*

$$\|G_5 - (x - r_8)(x - r_9)(x - r_{10})\| = 9.992\cdots \times 10^{-15}.$$

We see that the factor corresponding to these close roots is separated accurately.

Example 3

The case in which a separated factor is not so accurate.
Let $F(x) = (x - r_1)(x - r_2)\cdots(x - r_{15})$, *where*

$$\begin{array}{lll} r_1 = 0.580397, & r_6 = 0.090934, & r_{11} = -0.68825, \\ r_2 = 0.514122, & r_7 = -0.163329, & r_{12} = -0.703934, \\ r_3 = 0.496965, & r_8 = -0.190935, & r_{13} = -0.733996, \\ r_4 = 0.399967, & r_9 = -0.451888, & r_{14} = -0.74046, \\ r_5 = 0.226436, & r_{10} = -0.655015, & r_{15} = -0.766936. \end{array} \tag{6.2}$$

As in the above Example 2, we find

$$\begin{aligned} Q_3 &= x + 0.71581397514413, \\ Q_1 &= x^{12} + 0.63848007456761x^{11} - 1.231418428669x^{10} + \cdots. \end{aligned}$$

We put the initial factors $G_0(x)$ *and* $H_0(x)$ *as follows.*

$$G_0 = {Q_3}^3, \quad H_0 = \mathrm{quo}(F, G_0).$$

Then, the factor separation algorithm of cutoff separation accuracy 10^{-13} *converges at the eighth iteration, and we obtain*

$$G_8 = x^3 + 2.1783899989548x^2 + 1.5814143865241x + 0.38258438220795.$$

polynomial	acc. of G_i	number	average $\log_{10}\lvert P_{k_i}\rvert$
approx. square-free		284	
contains close roots	$< 1.0 \times 10^{-13}$	483	-6.59
	$> 1.0 \times 10^{-13}$	223	-7.62
	separation fails	10	-12.2 $(= \log_{10}\lvert P_{k_0}\rvert)$

Table II: Results of the first experiment (15 real roots).

In fact, we have

$$\|F - G_8H_8\| = 4.662\cdots \times 10^{-14}.$$

However, noting that three roots closest to the roots of $Q_3(x)$ *are* r_{12}, r_{13} *and* r_{14}*, we have*

$$\|G_8 - (x - r_{12})(x - r_{13})(x - r_{14})\| = 1.539\cdots \times 10^{-9}.$$

That is, G_8 *is about* 3.3×10^4 $(\simeq 1.54 \times 10^{-9}/4.66 \times 10^{-14})$ *times less accurate than the separation accuracy. This low accuracy is due to the smallness of* $|P_{k_8}|$ *described in* **4.3**; *in fact, we have* $1/|P_{k_8}| \simeq 9.5 \times 10^4$ *in this case. The smallness of* $|P_{k_8}|$ *is due to that* $G_i(x)$ *and* $H_i(x)$ *have mutually close roots; among the roots* $r_{10} \sim r_{15}$ *which are rather close one another, three roots* $r_{12} \sim r_{14}$ *are contained in* $G_0(x)$ *while others in* $H_0(x)$*.*

The results of 1000 similar computations with double-precision floating-point numbers are summarized in Table II. Among 1000 polynomials generated, 284 polynomials are decided to be approximately square-free at accuracy 0.0039. For the rest polynomials, factor separation succeeds for 706 ones but fails for 10 ones. Among the polynomials for which factor separation succeeds, the accuracy of factor $G_i(x)$ obtained is greater than 10^{-13} for 223 ones. Even for other 483 polynomials for which factor separation is quite successful, the average value of $|P_{k_i}|$ is rather small: average$(|P_{k_i}|) \simeq 2.57 \times 10^{-7}$. This means that a large amount of accuracy usually decreases during the execution of factor separation, which is common to methods based on the Euclidean algorithm, as analyzed in [9].This is obviously a weak point of the factor separation algorithm.

We have investigated the reason of failure for the 10 "separation fails" samples in Table II. For 8 samples, $\|G_i(x)\|$ and $\|H_i(x)\|$ grew very rapidly as i increased and the iteration diverged. This divergence is due to that initial factors $G_0(x)$ and $H_0(x)$ have mutually very close roots, indicating that the determination of $H_0(x)$ as $H_0 = \mathrm{quo}(F, G_0)$ is not always adequate. For the last 2 samples, the reason is precision lack: all the significant digits were lost during the computation of the remainder sequence.

To be fair, we also test polynomials whose roots are distributed randomly inside the unit disc in the complex plane. In this case, polynomials of degree 15 have a very small possibility of having close roots, so we generat polynomials of degree 30. The results of

polynomial	acc. of G_i	number	average $\log_{10}\lvert P_{k_i}\rvert$
approx. square-free		17	
contains close roots	$< 1.0 \times 10^{-13}$	51	-11.4
	$> 1.0 \times 10^{-13}$	22	-12.6
	separation fails	10	-13.0 ($= \log_{10}\lvert P_{k_0}\rvert$)

Table III: Results of the first experiment (30 complex roots).

experiment on 100 such samples with $\varepsilon_c = 0.0039$ are shown in Table III. Note that, in this experiment, we encounter much greater accuracy decreasing than that in Table II, and 9 "separation fails" samples in Table III are due to precision lack.

In the experiment, we have found an interesting phenomenon which may be called "recoupling of close roots": suppose the initial factors are set by the approximate square-free decomposition algorithm as $G_0 \simeq (x - r_1)(x - r_2)$ and $H_0 \simeq (x - r_3)(x - r_4)\cdots$, where r_1, r_2, r_3, r_4 are mutually close roots, then the factor separation algorithm gives us factors as $G_i \simeq (x - r_1)(x - r_3)$ and $H_i \simeq (x - r_2)(x - r_4)\cdots$. We note that the "recoupling" samples are included in "$> 1.0 \times 10^{-13}$" cases in Tables II and III.

Before giving the results of the second experiment, we explain two examples.

Example 4

Factor separation using Smith's error bound.

Let $F(x)$ be the same polynomial as used in Example 1. The D-K method gives us approximate roots of $F(x)$ and their error bounds as in Table I. We will separate the factor $(x - 0.1)^3$.

Using the approximate roots around $x = 0.1$, we put

$$\begin{aligned} G_0 &= (x - (0.09999909626683 - 0.00000042006060745357i)) \\ &\quad \times (x - (0.100000812092505 - 0.00000057737865711055i)) \\ &\quad \times (x - (0.100000111696375 + 0.00000100099111736021i)), \\ H_0 &= (x - (1.0 + 6.989\cdots i \times 10^{-26})) \\ &\quad \times \cdots \times (x - (-1.0 - 1.260\cdots i \times 10^{-24})). \end{aligned}$$

We see that $G_0(x)$ and $H_0(x)$ are not so accurate:

$$\|F - G_0 H_0\| = 0.0000000306\cdots.$$

With $G_0(x)$ and $H_0(x)$ as initial factors, the factor separation algorithm with cutoff separation accuracy 10^{-13} converges at the second iteration and we obtain

$$\begin{aligned} G_2 &= x^3 - (0.3 - 8.355\cdots i \times 10^{-17})x^2 + (0.03 - 1.670\cdots i \times 10^{-17})x \\ &\quad -0.001 - 8.354\cdots i \times 10^{-19}. \end{aligned}$$

numerical roots	error bounds
$r_1 = 1.0 + 1.2822039276924i \times 10^{-25}$	$5.4479465855004 \times 10^{-16}$
$r_2 = 0.50000000086145 - 0.000000003695338123761 8i$	0.000000067287807094736
$r_3 = 0.49999999836455 + 0.000000004885088817023 3i$	0.000000050828316994146
$r_4 = 0.2 - 4.3889644103006i \times 10^{-25}$	$5.4464568271483 \times 10^{-16}$
$r_5 = 0.100010000483665 + 1.494067905993i \times 10^{-14}$	0.000000013533803099881
$r_6 = 0.099994999805561 + 0.0000086605905057124i$	0.0000000104225543952628
$r_7 = 0.099995000055215 + -0.0000086601579160862i$	0.00000001913690749834
$r_8 = -0.1 - 4.0060633975116i \times 10^{-27}$	$7.4269864935165 \times 10^{-17}$
$r_9 = -0.3 + 4.7768925741988i \times 10^{-27}$	$5.6811855508313 \times 10^{-17}$
$r_{10} = -0.6 + 6.4180265608037i \times 10^{-26}$	$8.5861120967374 \times 10^{-16}$
$r_{11} = -0.7 - 2.071838872006i \times 10^{-26}$	$2.9347290930232 \times 10^{-16}$
$r_{12} = -1.0 + 1.1436211361243i \times 10^{-25}$	$1.9171995324719 \times 10^{-15}$

Table IV: Numerical roots and their error bounds for the polynomial in Example 5.

The factor separation is done quite well; in fact, we have

$$\|F - G_2H_2\| = 2.246\cdots \times 10^{-16}.$$

Example 5
The case in which a given polynomial has close roots.
Let $F(x)$ be the following polynomial.

$$\begin{aligned} F(x) &= (x-1)(x-0.5)^2(x-0.2)((x-0.1)^3-10^{-15}) \\ &\quad (x+0.1)(x+0.3)(x+0.6)(x+0.7)(x+1) \end{aligned}$$

$F(x)$ has three close roots of mutual distance about 10^{-5} around $x = 0.1$. Numerical roots and their error bounds determined by the D-K method are given in Table IV.

We see that the error bounds corresponding to roots r_5, r_6 and r_7 are a little smaller than those of multiple roots, but much greater than those of single roots. Moreover, we see that r_5, r_6 and r_7 are not multiple roots even though they have large errors. Therefore, we put the initial factor G_0 so as to contain these five inaccurate roots.

$$\begin{aligned} G_0 &= (x-r_2)(x-r_3)(x-r_5)(x-r_6)(x-r_7), \\ H_0 &= (x-r_1)(x-r_4)(x-r_8)(x-r_9)(x-r_{10})(x-r_{11})(x-r_{12}). \end{aligned}$$

The accuracies of these initial factors are as follows.

$$\begin{aligned} \|G_0 - (x-0.5)^2((x-0.1)^3-10^{-15})\| &= 0.000000001678\cdots, \\ \|H_0 - (x-1)(x-0.2)\cdots(x+0.7)(x+1)\| &= 2.220\cdots \times 10^{-16}. \end{aligned}$$

We see that H_0 is quite accurate, while G_0 is not. The factor separation algorithm with cutoff separation accuracy 10^{-13} converges at the second iteration, giving

$$\begin{aligned}
G_2 &= x^5 - (1.3 + 1.29\cdots i \times 10^{-21})x^4 + (0.58 + 2.86\cdots i \times 10^{-22})x^3 \\
&\quad + (-0.106000000000001 - 8.58\cdots i \times 10^{-24})x^2 \\
&\quad + (0.008500000000001 - 1.69\cdots i \times 10^{-24})x \\
&\quad + (-0.00025000000000025 + 9.87\cdots i \times 10^{-26}), \\
H_2 &= x^7 + (1.5 + 1.63\cdots i \times 10^{-18})x^6 + (-0.37 + 7.68\cdots i \times 10^{-19})x^5 \\
&\quad + (-1.487 - 1.34\cdots i \times 10^{-18})x^4 + (-0.6588 - 9.24\cdots i \times 10^{-19})x^3 \\
&\quad + (-0.01552 - 5.96\cdots i \times 10^{-20})x^2 + (0.0288 + 3.92\cdots i \times 10^{-20})x \\
&\quad + (0.00252 + 3.73\cdots i \times 10^{-21}).
\end{aligned}$$

The separation accuracy and accuracies of separated factors are

$$\begin{aligned}
\|F - G_2H_2\| &= 2.552\cdots \times 10^{-16}, \\
\|G_2 - (x-0.5)^2((x-0.1)^3 - 10^{-15})\| &= 2.220\cdots \times 10^{-16}, \\
\|H_2 - (x-1)(x-0.2)\cdots(x+0.7)(x+1)\| &= 5.551\cdots \times 10^{-16}.
\end{aligned}$$

In addition to the above two examples, we have tested such polynomials that 1) having several different multiple roots, 2) having multiple roots and close roots of mutual distance about 10^{-3}, 3) having close roots of mutual distances about 10^{-3} and 10^{-2}, and so on. For all the polynomials we have tested, the factor separation algorithm using Smith's error bound worked quite well and gave quite accurate results.

7 Concluding remarks

The factor separation algorithm is so simple and powerful that it will be applied to many practical problems. As we have shown above, the algorithm works quite well and gives accurate results so long as the close roots form clusters each of which is separated well from others and initial factors $G_0(x)$ and $H_0(x)$ have no mutually very close root. Even if we cannot clearly separate some roots as close roots from others, the factor separation algorithm works considerably well. However, we must overcome the following difficulties.

1. If the initial factors $G_0(x)$ and $H_0(x)$ are not adequately determined then the iteration may diverge. Determination of the initial factors depends on the problem to which we apply the factor separation algorithm, and we must be careful for the determination.

2. For polynomials of high degrees, the accuracy will be lost largely during the execution of the factor separation algorithm, which is the most serious weak point practically. We need a high precision floating-point number system.

3. For irregular polynomials, Proposition 1 indicates that the condition of convergence

of the iteration procedure becomes sever because $C \gg 1$, and Proposition 2 suggests that the accuracies of separated factors are much worse than the separation accuracy.

Acknowledgements

The authors thank one of the referees for valuable comments for revising the manuscript.

References

[1] G. E. Collins, *Subresultants and reduced polynomial remainder sequences*, J. ACM, **14** (1967), 128-142.

[2] E. Durand, *Solutions numériques des équations algébriques*, Tome I. Masson et Cie, Paris, 1960.

[3] I. O. Kerner, *Ein Gesamtschrittverfahren zur Berechnung der Nullstellen von Polynomen*, Numer. Math., **8** (1966), 290–294.

[4] T. Sasaki, *Approximate algebraic computation* (*in Japanese*), Collection of Research Reports, RIMS, Kyoto Univ., No. 676 (1988), 307-319.

[5] K. Shirayanagi, An algorithm to compute floating point Gröbner bases, *Mathematical Computation with Maple V: Ideas and Applications* (Ed. T. Lee), Birkhäuser, pp. 95-106, 1993.

[6] B. T. Smith, *Error bounds for zeros of a polynomial based upon Gerschgorin's theorems*, J. ACM, **17** (1970), 661-674.

[7] T. Sasaki and M-T. Noda, *Approximate square-free decomposition and root-finding of ill-conditioned algebraic equations*, J. Inform. Process., **12** (1989), 159-168.

[8] See *Booklet of Workshop on Symbolic-Numeric Algebra for Polynomials*, INRIA, Sophia-Antipolis, France, July 1996.

[9] T. Sasaki and M. Sasaki, *Analysis of accuracy decreasing in polynomial remainder sequence with floating-point number coefficients*, J. Inform. Process., **12** (1989), 394-403.

[10] T. Sakurai, H. Sugiura and T. Torii, *Numerical factorization of a polynomial by rational Hermite interpolation*, Numerical Algorithms, **3** (1992), 411-418.