Proceedings of the Steklov Institute of Mathematics, Vol. 218, 1997, pp. 117–128.
Translated from Trudy Matematicheskogo Instituta imeni V. A. Steklova, Vol. 218, 1997, pp. 122–133.

On the Complexity of Integration of Rational Fractions

S. B. Gashkov

Received February, 1997

1. Introduction. The algortithm of integration of rationl fractions is one of the most significant and most often used algorithms of analysis. It is well known that the integral of a rational fraction can be represented as the sum of a rational and a logarithmic part, and that the rational part can be separated only by means of arithmetic operations. The algorithm for this purpose was found by Ostrogradskii in 1844 (see [1]). Twenty years later Hermite proposed a different algorithm for the same purpose (see [3]). In the Russian literature the algorithm for separating the rational part of the integral of a rational fraction is called the Hermite–Ostrogradskii algorithm, but actually the Hermite algorithm is usually presented. In foreign literature the name of Ostrogradskii is usually omitted, although his article [1] was published in St. Petersbourg in French.

After Karatsuba proposed in 1962 a fast algorithm for multiplication of numbers [4] (in his recent article [5] he himself writes, in an interesting form, how it happened) an interest arose to the evaluation of the complexity of algorithms of algebra and analysis and to the consruction of fast modifications of these algorithms. Different fast algorithms have been found for the main arithmetic operations with numbers and polynomials in the works [6–10] and others, as well as fast algorithms for computing the values of polynomials and their interpolations (see, e.g., [11]), fast algorithms for computing determinants and for solving systems of linear equations (after [12] dozens of works were written on this subject), and many other results, among which there are also those which can be considered as polynomial algorithms. It is impossible to cite all of them in a short article (see, e.g., [13–15] and the references therein).

It is shown in [16] how Hermite's algorithm can be modernized so that it could operate with the complexity $O(n \log^2 n)$, where n is the degree of the denominator of the fraction being integrated (which can be assumed to be proper). This algorithm consists of four subalgorithms, namely, a square-free fractorization of the denominator of the fraction, its fractorization into partial fractions, integration of the obtained partial fractions by parts and the summation of the rational parts of the resulting integrals and integrands which are also rational fractions. Each of these algorithms is of interest in itself and has numerous applications, especially the first two and the last one. All these algorithms have the same estimate of the complexity in the order of magnitude.

In the work [16] only the estimates of the complexity of the first and the third algorithm are given in detail since the estimates for the other two algorithms were obtained in [17, 18]. These three works contain a long history of this question, and therefore we omit it here.

However, it is not written there that this history dates back to Ostrogradskii since the algorithm for square-free factorization [16] is given in [1] (the author of [16] evidently does not know this fact). This algorithm is somewhat more complicated for understanding than the standard algorithm (which dates back to the Dutch mathematician of the century before last Johann Hudde), but is, nevertheless, simple enough for its presentation to first-year students, and therefore deserves a wider propagation and application.

2. Estimate of the complexity of the algorithm for rational fractions integration and its subalgorithms. We can propose the estimate of the complexity of integration of regular rational fractions of degree n in the form

$$\lceil \log_2 k \rceil \big(21\,\mathrm{M}(n) + 11n\big) + 24\,\mathrm{M}(n) + 26n + 4\,\mathrm{GD}(n),$$

where k is the number of square-free factors in Ostrogradskii's factorization, $\mathrm{M}(n)$ is the estimate of the complexity of multiplication of two polynomials of degree n satisfying the condition of superadditivity: $\mathrm{M}(x) + \mathrm{M}(y) \le \mathrm{M}(x+y)$, and $\mathrm{GD}(n)$ is a similar estiamte of complexity of computing the GCD of two polynomials. It is known (see, e.g., [13]) that we can take $Cn\log n$ as $\mathrm{M}(n)$ and $C\,\mathrm{M}(n)\log n$ as $\mathrm{GD}(n)$ for a large C (this was proved in [19, 20]).

In this case, we can estimate the complexity of square-free factorization of the polynomial P of degree n as $2\,\mathrm{GD}(n) + 7\,\mathrm{M}(n) + 6n$, the complexity of partial fraction decomposition of a regular fraction with denominator P as

$$\mathrm{GD}(n) + \lceil \log_2 k \rceil \left(17\,\mathrm{M}(n) + \frac{17n}{2}\right) + 8\,\mathrm{M}(n) + 5n,$$

the complxity of integration of the resulting partial fractions as

$$3\,\mathrm{M}(n) + 6\,\mathrm{M}(p) + \mathrm{GD}(n-p) + 15p,$$

where $p = \deg(P, dP/dx)$, and the complexity of summation of the partial fractions resulting from this intergration as

$$\lceil \log_k t \rceil \left(3\,\mathrm{M}(n) + \mathrm{M}(p) + \frac{5p}{2}\right) + \mathrm{M}(n-p) + 3(n-p).$$

3. Ostrogradskii's algorithm for square-free factorization of polynomials. We shall give the algorithm for square-free factorization in the notation of Ostrogradskii and estimate its complexity (i.e., the number of operations it performs). Let $P = P_1$ be a polynomial being factorized. We set

$$P_2 = \left(P_1, \frac{dP_1}{dx}\right), \qquad Q_1 = \frac{P_1}{P_2}, \qquad R_1 = \frac{dP_1/dx}{P_2}, \qquad q_1 = \left(Q_1, R_1 - \frac{dQ_1}{dx}\right),$$
$$Q_2 = \frac{Q_1}{q_1}, \qquad R_2 = \frac{R_1 - dQ_1/dx}{q_1}, \qquad q_2 = \left(Q_2, R_2 - \frac{dQ_2}{dx}\right)$$

and so on, until we get $R_{k+1} = dQ_{k+1}/dx$ (here and in the sequel we denote by (f,g) the GCD of the polynomials f and g). Then $P = q_1 q_2^2 \cdots q_k^k$ is the required decomposition since q_i do not have multiple roots and are coprime. Note that we have incidentally found the decomposition $P = Q_1 Q_2 \cdots Q_k$, where $Q_i = q_i \cdots q_k$, $i = 1, \dots, k$, are also square-free factors. However, they are no longer coprime but successively divide one another.

Let us compare this algorithm with the standard one. In the latter, the sequence of polynomials is recurrently computed from the relation $P_{i+1} = (P_i, dP_i/dx)$ and, with the use of the theorem on the lowering of the multiplicity of the root in the process of differentiation, it is verified that

$$P_i = q_i q_{i+1}^2 \cdots q_k^{k-i+1} = Q_i Q_{i+1} \cdots Q_k,$$

and then it is found by means of division that

$$Q_i = \frac{P_i}{P_{i+1}}, \qquad q_i = \frac{Q_i}{Q_{i+1}}.$$

Ostrogradskii's algorithm is substantiated by his remarkable identities

$$R_{i+1} = \frac{dP_{i+1}/dx}{P_{i+2}} = \frac{R_i - dQ_i/dx}{q_i}, \qquad q_i = \left(Q_i, R_i - \frac{dQ_i}{dx}\right).$$

For the convenience of the reader we shall give their proof, the more so as it is short. Using induction, we find that

$$R_i - \frac{dQ_i}{dx} = \frac{dP_i/dx}{P_{i+1}} - \frac{d(P_i/P_{i+1})}{dx} = \frac{P_i \times (dP_{i+1}/dx)}{P_{i+1}^2} = \frac{Q_i \times (dP_{i+1}/dx)}{P_{i+1}} = \frac{Q_{i+1}q_i \times (dP_{i+1}/dx)}{P_{i+1}},$$

whence we have

$$R_{i+1} = \frac{R_i - dQ_i/dx}{q_i} = \frac{Q_{i+1} \times (dP_{i+1}/dx)}{P_{i+1}} = \frac{dP_{i+1}/dx}{P_{i+2}}.$$

Since

$$P_i = q_i q_{i+1}^2 \cdots q_k^{k-i+1}, \qquad R_i = \frac{dP_i/dx}{P_{i+1}}, \qquad Q_i = q_i \cdots q_k,$$

it follows from the theorem on the lowering of the multiplicity of the root in the process of differentiation that $(R_i, Q_i) = 1$, whence it follows, now without any induction, that

$$\left(Q_i, R_i - \frac{dQ_i}{dx}\right) = (Q_i, R_{i+1}q_i) = (Q_{i+1}q_i, R_{i+1}q_i) = (Q_{i+1}q_i, R_{i+1}q_i) = q_i(Q_{i+1}, R_{i+1}) = q_i.$$

Note that the relations

$$R_i = \frac{dP_i/dx}{P_{i+1}}, \qquad Q_i = \frac{P_i}{P_{i+1}}$$

imply that $\deg R_i = \deg P_i - \deg P_{i+1} - 1 = \deg Q_i - 1$.

The advantage of Ostrogradskii's algorithm as compared to the astandard one is obvious since at every step of iteration it requires the finding of the GCD of the lower-degree polynomials than in the standard algorithm, and one division is used rather than two, and again for polynomials of a lower degree. Morever, it admits of a good evaluation of the complexity when fast algorithms of multiplication, division, and calculation of the GCD are used in it.

4. Estimate of the complexity of Ostrogradskii's algorithm. It can be obtined very simply. We set $\deg Q_i = n_i$, $\deg q_i = m_i$, $\deg P_2 = p = n_2 + \cdots + n_k$, and then $m_i = n_i - n_{i+1}$, $\deg R_i = n_i - 1$. We estimate the complexity of the exact division of a polynomial of degree m by a polynomial of degree n as $3(\mathrm{M}(m-n) + m - n) + \mathrm{M}(m)$ (see, e.g., [21, problems 17.25, 17.32]), and, hence, the complexity of the division of two different polynomials of a degree not higher than m by the same polynomial of degree n can be estimated as $3(\mathrm{M}(m-n) + m - n) + 2M(m)$ (see, e.g., [21, solutions of problems 17.25–17.26]). Then the required estimate has the form

$$\begin{aligned} n &+ \mathrm{GD}(n) + 3\,\mathrm{M}(n-p) + 3n - 3p + 2\,\mathrm{M}(n) + \sum_{i=1}^{k}\bigl(\mathrm{GD}(n_i) + 2\,\mathrm{M}(n_i) + 2n_i\bigr) + \sum_{i=2}^{k}\bigl(3\,\mathrm{M}(n_i) + 3n_i\bigr) \\ &\le n + \mathrm{GD}(n) + 3\,\mathrm{M}(n-p) + 3n - 3p + 2\,\mathrm{M}(n) + \mathrm{GD}(n) + 2\,\mathrm{M}(n) + 2n + 3\,\mathrm{M}(p) + 3p \\ &\le 2\,\mathrm{GD}(n) + 7\,\mathrm{M}(n) + 6n. \end{aligned}$$

The obtained estimate shows that the complexity of Ostrogradskii's algorithm for a square-free factorization of an nth degree polynomial exceeds only twice the complexity of the extended algorithm for finding the GCD of two polynomials of degree n.

5. One more version of Ostrogradskii's algorithm. It is interesting to note that Ostogradskii propossed in his note [2] one more version of the factorization algorithm based on the identity $q_j = \big(Q, R - j \times (dQ/dx)\big)$, where

$$Q = Q_1 = \frac{P_1}{P_2} = \frac{P}{P_2}, \qquad R = R_1 = \frac{dP_1/dx}{P_2}, \qquad P_2 = \left(\frac{dP_1}{dx}, P_1\right).$$

The proof follows from the chain of relations (in which, for brevity, the differentiation is denote by prime)

$$R = \frac{(P_1)'}{P_2} = \frac{(q_1 q_2^2 \cdots q_k^k)'}{q_2 \cdots q_{k-1}^k} = \sum_{i=1}^{k} i q_1 \cdots q_{i-1} (q_i)' q_{i+1} \cdots q_k,$$

$$Q' = \sum_{i=1}^{k} q_1 \cdots q_{i-1} (q_i)' q_{i+1} \cdots q_k, \qquad R - jQ' = \sum_{i=1}^{k} (i-j) q_1 \cdots q_{i-1} (q_i)' q_{i+1} \cdots q_k$$

(justified by the Leibniz differentiation rule) and from the note stating that all terms in the last sum, except for the sth term are divisible by q_s, and the sth term is prime to q_s (since all its factors are prime to it). Consequently, $(R - jQ', q_s) = 1$ for $s \neq j$ and, obviously, the polynomial $R - jQ$ is divisible by the polynomial q_j.

The indicated Ostogradskii formulas are more elegant than the previous formulas, but the algorithm described in this section is obviously more complicated than the preceding one since the degrees of the polynomials in the sequence in it, for which we have to find the GCD, do not become lower, and it does not admit of an obvious good upper bound. However, if we have to calculate not all factors q_i, but only any one of them, then this algorithm is obviously the fastest.

6. Estimate of the complexity of the algorithm for factorization of a rational fraction into partial fractions. The main idea of the fast algorithm for factorization into partial fractions consists in the application of Karatsuba's method of division in half.

Let P/Q be an arbitrary proper fraction of degree n, $Q = q_1 \cdots q_s$, $(q_i, q_j) = 1$, $i \neq j$. We set $Q_i = Q/q_i$, and then, as is known, the relation

$$\frac{P}{Q} = \sum_{i=1}^{s} \frac{p_i}{q_i},$$

where

$$p_i = (P \bmod q_i \times Q_i^{-1} \bmod q_i) \bmod q_i.$$

holds true. In order to substantiate this relation, we must verify that $(Q_1, \ldots, Q_s) = 1$, use the linear representation of the GCD $1 = \sum_{i=1}^{s} U_i Q_i$, and note that $U_i \bmod q_i = Q_i^{-1} \bmod q_i$ and

$$\frac{P}{Q} = \sum_{i=1}^{s} \frac{P \times U_i}{q_i} = \sum_{i=1}^{s} \frac{(P \times U_i) \bmod q_i}{q_i}.$$

Assume that we have already calculated $P \bmod q_i$ and $Q_i \bmod q_i$. Then $Q_i^{-1} \bmod q_i$ can be found with complexity $\mathrm{GD}(\deg q_i)$ by the application to the pair $Q_i \bmod q_i, q_i$ of the extended Euclidean algorithm which computes the linear representetion $1 = (Q_i, q_i) = UQ_i + Vq_i$, where $U = Q_i^{-1} \bmod q_i$, and $p_i = (P \bmod q_i)(Q_i^{-1} \bmod q_i)$ can be found with complexity $7\,\mathrm{M}(\deg q_i) + 5\deg q_i$ by means of the multiplication and the subsequent division of the result by q_i (see [21, problems 17.32, 17.34]). With due account of superadditivity, we have the estimate $\mathrm{GD}(n) + 7\,\mathrm{M}(n) + 5n$.

In order to calculate $P \bmod q_i$, we use the method of division in half. We construct a sequence of polynomials

$$Q_{\alpha_1,\dots,\alpha_k}, \qquad \alpha_i = 0,1, \quad k \le l(s) = \lceil \log_2 s \rceil,$$

such that

$$Q = Q_0 \times Q_1, \quad \dots, \quad Q_{\alpha_1,\dots,\alpha_k} = Q_{\alpha_1,\dots,\alpha_k,0} \times Q_{\alpha_1,\dots,\alpha_k,1}, \quad \dots,$$

and such that the number of factors in adjacent polynomials $Q_{\widetilde{\alpha},0}$, $Q_{\widetilde{\alpha},1}$ would differ not more than by 1. Let us calculate the sequence of polynomials $P_{\widetilde{\alpha}}$ recursively:

$$P_{\widetilde{\alpha},\alpha_{k+1}} = P_{\widetilde{\alpha}} \bmod Q_{\widetilde{\alpha},\alpha_{k+1}}.$$

If we set $\deg Q_{\widetilde{\alpha}} = n_{\widetilde{\alpha}}$, then the complexity of calculation of both polynomials $P_{\widetilde{\alpha},0}$, $P_{\widetilde{\alpha},1}$ can be estimated as

$$\begin{aligned} &3\big(\mathrm{M}(n_{\widetilde{\alpha},1}) + n_{\widetilde{\alpha},1}\big) + 2\,\mathrm{M}(n_{\widetilde{\alpha}}) + n_{\widetilde{\alpha},0} + 3\big(\mathrm{M}(n_{\widetilde{\alpha},0}) + n_{\widetilde{\alpha},0}\big) + 2\,\mathrm{M}(n_{\widetilde{\alpha}}) + n_{\widetilde{\alpha},1} \\ &\qquad \le 3\,\mathrm{M}(n_{\widetilde{\alpha}}) + 4n_{\widetilde{\alpha}} + 4\,\mathrm{M}(n_{\widetilde{\alpha}}) = 7\,\mathrm{M}(n_{\widetilde{\alpha}}) + 4n_{\widetilde{\alpha}}), \end{aligned}$$

and the complete complexity can be estimated as

$$\begin{aligned} \sum_{k=1}^{l(s)} \sum_{\alpha_1=0,1;\dots;\alpha_k=0,1} 7\,\mathrm{M}(n_{\widetilde{\alpha}}) + 4n_{\widetilde{\alpha}} &\le l(s)\left(7\,\mathrm{M}\left(\sum_{\alpha_1=0,1;\dots;\alpha_k=0,1} n_{\widetilde{\alpha}}\right) + 4 \sum_{\alpha_1=0,1;\dots;\alpha_k=0,1} n_{\widetilde{\alpha}}\right) \\ &\le l(s)\big(7\,\mathrm{M}(n) + 4n\big). \end{aligned}$$

With the use of Karatsuba's method we evaluate the complexity of calculation of the sequence of polynomials $Q_{\widetilde{\alpha}}$ as $(l(s) - 1)\,\mathrm{M}(n)$.

Since

$$Q_i \bmod q_i = \frac{Q_i \bmod q_i^2}{q_i},$$

this sequence can also be evaluated by means of dichotomy by the quantity of the same order of magnitude.

7. A more precise definition of multiplicative constants in the estimate of the complexity of the algorithm for factorization into partial fractions. Note that the sequence of remainders modulo $Q_{\widetilde{\alpha}}^2$ that can arise in calculations is represented as $Q_{\widetilde{\alpha}} \times R_{\widetilde{\alpha}}$, where the sequence $R_{\widetilde{\alpha}}$ is computed recursively:

$$R_{\widetilde{\alpha},\alpha_{k+1}} = Q_{\widetilde{\alpha},1-\alpha_{k+1}} \times R_{\widetilde{\alpha}} \bmod Q_{\widetilde{\alpha},\alpha_{k+1}},$$

and the final results of $Q_i \bmod q_i$ have the form $R_{\alpha_1,\dots,\alpha_{l(s)}}$. Since $\deg R_{\widetilde{\alpha}} < n_{\widetilde{\alpha}}$, the complexity of calculating both polynomials $R_{\widetilde{\alpha},0}$, $R_{\widetilde{\alpha},1}$ is estimated (with the aid of problems 17.20, 17.25, 17.32 from [21]) as

$$\begin{aligned} &\mathrm{M}(n_{\widetilde{\alpha}}) + 3\big(\mathrm{M}(2n_{\widetilde{\alpha},0}) + 2n_{\widetilde{\alpha},0}\big) + \mathrm{M}(n_{\widetilde{\alpha}} + n_{\widetilde{\alpha},0}) + n_{\widetilde{\alpha},1} + \mathrm{M}(2n_{\widetilde{\alpha},0}, n_{\widetilde{\alpha},1}) + \mathrm{M}(n_{\widetilde{\alpha}}) \\ &\quad + 3\big(\mathrm{M}(2n_{\widetilde{\alpha},1}) + 2n_{\widetilde{\alpha},1}\big) + \mathrm{M}(n_{\widetilde{\alpha}} + n_{\widetilde{\alpha},1}) + n_{\widetilde{\alpha},0} + \mathrm{M}(2n_{\widetilde{\alpha},1}, n_{\widetilde{\alpha},0}) \\ &\qquad \le 4\,\mathrm{M}(2n_{\widetilde{\alpha}}) + \mathrm{M}(3n_{\widetilde{\alpha}}) + 3\,\mathrm{M}(n_{\widetilde{\alpha}}) + 7n_{\widetilde{\alpha}}, \end{aligned}$$

and the complete complexity as $l(s)(4\,\mathrm{M}(2n)+\mathrm{M}(3n)+3\,\mathrm{M}(n)+7n)$. If we calculate the sequence $R_{\widetilde{\alpha}}$ before we calculate the sequence $P_{\widetilde{\alpha}}$, then, when calculating the latter sequence, we need not "invert" the polynomials $Q_{\widetilde{\alpha}}$ since we have already performed this inversion when we calculated the preceding sequence (this is clear from the solution of problem 17.26 in [21]), and this makes it possible to lower the estimate of the complexity by $3\,\mathrm{M}(n)+3n$, and the total estimate will have the form $l(s)(4\,\mathrm{M}(2n)+\mathrm{M}(3n)+8\,\mathrm{M}(n)+8n)-\mathrm{M}(n)$. We can lower the estimate still more if, before carrying out the multiplication

$$R_{\widetilde{\alpha},\alpha_{k+1}} = Q_{\widetilde{\alpha},1-\alpha_{k+1}} \times R_{\widetilde{\alpha}} \bmod Q_{\widetilde{\alpha},\alpha_{k+1}}$$

we reduce $R_{\widetilde{\alpha}}$ modulo $Q_{\widetilde{\alpha},\alpha_{k+1}}$ and then "invert" the polynomial $Q_{\widetilde{\alpha}}$ only once. Then we shall spend $2\,\mathrm{M}(n)$ operations for multiplication and $3\,\mathrm{M}(n)+3n+4\,\mathrm{M}(n)+2n$ operations for modulo reduction. Then the total estimate will have the form $l(s)(14\,\mathrm{M}(n)+6n)-\mathrm{M}(n)$. We can also note that, when the condition $n_{\widetilde{\alpha},0}=n_{\widetilde{\alpha},1}$ of the equality of the degrees of the adjacent polynomials is fulfilled, this estimate decreases by $2l(s)\,\mathrm{M}(n)$.

8. Estimate of the complexity of the second subalgorihm of the algorithm for decomposition of a rational fraction into partial fractions. Here we speak about the decomposition of a proper fraction of the form P/Q^s into the sum of still more simple fractions

$$\sum_{i=1}^{s} \frac{P_i}{Q^i}.$$

Obviously, this problem is equivalent to the decomposition of the polynomial P in the powers of Q. It can be solved in the same way as the above-given computation of the remainders of a polynomial from the given moduli, only here all q_i are equal to Q and at every step the polynomial $P_{\widetilde{\alpha}}$ is used to construct the polynomials $P_{\widetilde{\alpha},0}$, $P_{\widetilde{\alpha},1}$ such that

$$P_{\widetilde{\alpha}} = P_{\widetilde{\alpha},0}Q_{\widetilde{\alpha},1} + P_{\widetilde{\alpha},1}.$$

If, for every division, we choose the polynomial with the highest degree from two adjacent polynomials $Q_{\widetilde{\alpha}}$, then the complexity of all divisions will be estimated (with the aid of problems 17.20, 17.25, 17.31, 17.32 from [21]) as

$$\sum_{k=1}^{l(s)} \sum_{\alpha_1=0,1;\dots;\alpha_k=0,1} \Big(3\,\mathrm{M}(n_{\widetilde{\alpha},0}) + 3n_{\widetilde{\alpha},0} + \mathrm{M}(n_{\widetilde{\alpha},1}) + n_{\widetilde{\alpha},1} + 2\,\mathrm{M}\Big(\frac{n_{\widetilde{\alpha}}}{2}\Big) + \frac{n_{\widetilde{\alpha}}}{2}\Big)$$
$$\le l(s)\Big(4\,\mathrm{M}\Big(\frac{n}{2}\Big) + \mathrm{M}(n) + \frac{5n}{2}\Big),$$

and the complexity of calculation os all polynomials $Q_{\widetilde{\alpha}}$, which result from a successive squaring or multiplication of adjacent polynomials, is evaluated as $2\,\mathrm{M}(n)$, where $n = s\deg Q$, i.e., the complexity of the whole subalgorithm does not exceed $l(s)(3\,\mathrm{M}(n)+5n/2)+2\,\mathrm{M}(n)$.

We see that Karatsuba's method was actually used here as well.

The final estimate of the complexity of the algorithm for decomposition into partial fractions has the form

$$\lceil \log_2 k \rceil \Big(17\,\mathrm{M}(n) + \frac{17n}{2}\Big) + 8\,\mathrm{M}(n) + 5n + \mathrm{GD}(n),$$

where k is the length of the decomposition of the denominator into squre-free factors, $q_1q_2^2\cdots q_k^k$, and $n=\deg Q$.

9. Estimate of the complexity of the algorithm proper for integration of rational fractions. We shall first estimate the complexity of integration of the fraction

$$\frac{P}{Q^s} = \sum_{i=1}^{s} \frac{P_i}{Q^i},$$

where $\deg P_i < \deg Q$, $s \deg Q = n$, $s > 1$. For this purpose, we use the identity

$$\int \frac{P_i}{Q^i}\, dx = \frac{-P_i B/(i-1)}{Q^{i-1}} + \int \frac{P_i A + d(P_i B/(i-1))/dx}{Q^{i-1}}\, dx,$$

where $AQ + B \times (dQ/dx) = 1$, $\deg A, \deg B < \deg Q$, which is obtained by integration by parts. We evaluate the complexity of computing the polynomials A and B as $\mathrm{GD}(\deg Q)$ and then find the numerator of the fraction under the sign of the second integral with complexity $2\,\mathrm{M}(\deg Q) + 6 \deg Q$, and divide it by Q with a remainder, representing it as $CQ + D$, where $\deg C, \deg D < \deg Q$, with complexity $6\,\mathrm{M}(\deg Q) + 5 \deg Q$, and then sum up C with P_{i-2} and D with P_{i-1} with complexity $2 \deg Q$. As a result, the problem reduces to a similar problem but i decreases by 1. This means that the integration of the fraction

$$\frac{P}{Q^s} = \sum_{i=1}^{s} \frac{P_i}{Q^i}$$

is carried out with complexity

$$(s-1)\big(5\,\mathrm{M}(\deg Q) + 10 \deg Q\big) + 3\,\mathrm{M}(\deg Q) + \deg Q + \mathrm{GD}(\deg Q)$$

since we must compute A, B and "invert" the polynomial Q only once and need not divide by $i-1$ when $i = 2$.

We must also have in mind the complexity of reduction of the rational part of the integral to the form

$$S_0 + \sum_{i=1}^{s-1} \frac{S_i}{Q^i},$$

where $\deg S_i < \deg Q$. It should be noted that $-P_i B/(i-1)$ is computed with complexity $\mathrm{M}(\deg Q) + 2 \deg Q$ (for $i = 2$ the last term vanishes), the division of the result by Q and, hence, its representation in the form $C_i Q + D_i$, where $\deg C_i, \deg D_i < \deg Q$, is carried out with complexity $3\,\mathrm{M}(\deg Q) + 2 \deg Q$ (since "inversion" of the polynomial Q was performed earlier), and the summation (for $i > 2$) of the polynomials C_i and D_{i-1} is performed with complexity $\deg Q$. The final result is the estimate

$$(s-1)\big(4\,\mathrm{M}(\deg Q) + 5 \deg Q\big) - 3 \deg Q.$$

Adding this estimate to that obtained earlier, we finally have the estimate

$$(s-1)\big(9\,\mathrm{M}(\deg Q) + 15 \deg Q\big) + 3\,\mathrm{M}(\deg Q) - 2 \deg Q + \mathrm{GD}(\deg Q).$$

Using the superadditivity of the function $\mathrm{M}(n)$, we reduce this estimate to the form

$$6\,\mathrm{M}(n - \deg Q) + 15n + 3\,\mathrm{M}(n) - 17 \deg Q + \mathrm{GD}(\deg Q).$$

Applying this estiamte to each of the fractions resulting from the decomposition of the initial fraction N/P, where $P = q_1 q_2^2 \cdots q_k^k$, and $n = \deg P$, and using the superadditivity of the functions $\mathrm{M}(n)$ and $\mathrm{GD}(n)$, we get the estimate of the complexity of the integration proper in the form

$$3\,\mathrm{M}(n - \deg q_1) + 6\,\mathrm{M}(p) + 15p - 2 \deg Q + 2 \deg q_1 + \mathrm{GD}(\deg Q) < 3\,\mathrm{M}(n) + 6\,\mathrm{M}(p) + \mathrm{GD}(n-p) + 15p,$$

but this time we understand Q as the polynomial $q_1 q_2 \cdots q_k$, and $p = \deg P_1$, where $P_1 = P/Q$.

10. Estimate of the complexity of addition of the obtained rational fractions. Ostrogradskii himself represented the result of integration as

$$\int \frac{N}{P}\,dx = \frac{X}{P} + \int \frac{Y}{Q}\,dx.$$

In order to find the polynomials X and Y, we must sum up the obtained fractions (and the polynomials in the rational parts of the integrals). Applying the method of division in half and noting that the addition of two fractions of degree d requires not more than $3\,\mathrm{M}(d)+2d$ operations, we can compute the sum of proper fractions

$$\sum_{i=1}^{s} \frac{P_i}{Q^i}$$

(solving the inverse of the problem considered at the end of Section 8) in

$$l(s)\Big(\mathrm{M}(s\deg Q) + \frac{1}{2}s\deg Q\Big)$$

operations. Polynomials of a degree lower than $\deg q_i$ are added with complexity $\deg Q$ and then their sum is added to the proper fraction obtained earlier with complexity $2\deg Q + \mathrm{M}(\deg Q)$. Thus, for the complexity of addition of fractions we get the estimate

$$\begin{aligned} &3\deg Q + \mathrm{M}(\deg Q) + \lceil \log_2 k\rceil \Big(3\,\mathrm{M}(p) + 2p + \mathrm{M}(p) + \frac{p}{2} + 3\,\mathrm{M}(\deg Q) + 2\deg Q\Big) \\ &\quad \le \lceil \log_k t\rceil \Big(3\,\mathrm{M}(n) + \mathrm{M}(p) + \frac{5p}{2}\Big) + \mathrm{M}(n-p) + 3(n-p). \end{aligned}$$

The resulting estimate of the complexity of the algorithm for integration of a rational fraction has the form

$$\begin{aligned} &\lceil \log_2 k\rceil \Big(3\,\mathrm{M}(n) + \mathrm{M}(p) + \frac{5p}{2}) + \mathrm{M}(n-p) + 3(n-p) + 3\,\mathrm{M}(n) + 6\,\mathrm{M}(p) + \mathrm{GD}(n-p) + 15p \\ &\quad + \lceil \log_2 k\rceil \Big(17\,\mathrm{M}(n) + \frac{17n}{2}\Big) + 8\,\mathrm{M}(n) + 5n + \mathrm{GD}(n) + 2\,\mathrm{GD}(n) + 7\,\mathrm{M}(n) + 6n \\ &\qquad \le \lceil \log_k \Big(20\,\mathrm{M}(n) + \mathrm{M}(p) + \frac{17n}{2} + \frac{5p}{2}\Big) \\ &\qquad\quad + 3\,\mathrm{GD}(n) + \mathrm{GD}(n-p) + 19\,\mathrm{M}(n) + 5\,\mathrm{M}(p) + 14n + 12p \\ &\qquad \le \lceil \log_2 n\rceil (21\,\mathrm{M}(n) + 11n) + 24\,\mathrm{M}(n) + 26n + 4\,\mathrm{GD}(n). \end{aligned}$$

11. Estimate of the complexity of finding the logarithmic part of an integral. The logarithmic part of the integral

$$\int \frac{N}{P}\,dx,$$

namely,

$$\int \frac{Y}{Q}\,dx,$$

cannot be found by purely arithmetic means (this problem is discussed, e.g., in [23]). However, if we know the roots of the polynomial Q, then it is, obviously, sufficient to decompose the fraction

Y/Q into partial fractions in order to find the algorithmic part. Indeed, if $Q = q_1 \cdots q_m$, where $q_i(x) = x - c_i$, $c_i \in \mathbb{C}$, $(q_i, q_j) = 1$, $i \neq j$, then we set $Q_i = Q/q_i$ and, as is well known, obtain the decomposition

$$\frac{Y}{Q} = \sum_{i=1}^{s} \frac{p_i}{q_i},$$

where

$$p_i = Y \times Q_i^{-1} \bmod q_i = Y \bmod q_i Q_i^{-1} \bmod q_i = \frac{Y(c_i)}{Q'(c_i)} = b_i \in \mathbb{C}.$$

Since the numbers $Y(c_i)$ and $Q'(c_i)$ are the remainders of the division of the plynomials $Y(x)$ and $Q'(x) = dQ/dx$ by the binomials $q_i(x) = x - c_i$, it follows, as was already noted in a more general situation, that they can be computed with complexity $l(m)(15\,\mathrm{M}(m) + 8m) + 2m - \mathrm{M}(m)$ (we have taken into account here that it suffices to calculate only once the products of the binomials q_i which are necessary for the computation of both numerators and denominators of the fractions b_i). Then we carry out the pairwise addition of the complex conjugate fractions with complexity not higher than $5m/2$ and integrate all fractions in a simple way with complexity not exceeding $4m$.

Note that if we are given the decomposition of the denominator P of the fraction being integrated into factors over the field $\mathbb{R}$, then there is no need to begin with calculating the polynomial Q by Ostsgradskii's method since the problem reduces at once to the decomposition of the fraction N/P into the sum of fractions of the form Y/Q^s, where Q is either a linear binomial or a quadratic trinomial. In the first case, the decomposition of such a fraction into the sum of partial fractions reduces to finding the Taylor expansion of the polynomial P relative to the point a, where $Q(x) = x - a$, and, as is shown in [22], this can be done quicker than in the process described above, namely, with complexity $O(s \log s)$. The second case, as was noted in [17], reduces to the first one if we isolate a perfect square in Q, make the requisite linear change of variables in the polynomial Y (and this is the same as the Taylor expansion just mentioned), and then present it in the form $Y_0(x^2) + xY_1(x^2)$. After the substitution $z = x^2$ the problem again reduces to the first case, and therefore, in the second case, the complexity of finding the decomposition of the fraction under consideration into the sum of partial fractions remains the same in the order of magnitude.

12. Estimate of the complexity of the original Ostrogradskii algorithm for integration of rational fractions. The algorithm of square-free factorization of polynomials presented above is only the beginning of Ostrogradskii's algorithm for integration of rational fractions [1]. This algorithm is too large to be given here (the work [1] is 50 pages long). Using the techniques described above, we can evaluate its complexity as

$$2\,\mathrm{GD}(n) + \mathrm{GD}(p) + O\left(\frac{nk\,\mathrm{M}(n_2)}{n_2}\right) + O\big(\log k \times \mathrm{M}(n)\big),$$

where $p = n - n_1$ and $n_i = \deg Q_i$, and the sequence of polynomials Q_i is the same as that given in Section 3. Generally speaking, this estimate is worse than the estimate of the complexity of Hermite's algorithm, but for $p/n \to 0$ and $(k \log n_2) = o(\log^2 n)$ it is asymptotically twice as good.

13. Estimate of the complexity of the algorithm of the Chinese remainder theorem and of the problem of interpolation of polynomials. The Chinese remainder theorem and the related algorithms can boast of numerous applications in different divisions of mathematics and in many specific works (say, the algorithm for fast multiplicaton [8] uses modular arithmetic which is really based on the Chinese remainder theorem). It is interesting to note that the fast version of the Chinese algorithm is closely connected with the fast algorithm of decomposition into partial fractions, and both of them are actually based on the idea which was used for the first time by Karatsuba in [4].

Suppose that we are given polynomials $q_1, \dots, q_k$, $(q_i, q_j) = 1$, $i \neq j$, and polynomials $p_1, \dots, p_k$, $\deg p_i < n_i = \deg q_i$, $n = n_1 + \dots + n_k$. The Chinese remainder theorem asserts the existence and uniqueness of a polynomial P modulo Q such that $P \bmod q_i = p_i$, $1 \leq i \leq k$. We set

$$Q = q_1 \cdots q_k, \qquad Q_i = \frac{Q}{q_i}, \qquad R_i = Q_i^{-1} \bmod q_i,$$

and then, as is known, the relations

$$P = \sum_{i=1}^{s} (p_i Q_i R_i) \bmod Q = Q \sum_{i=1}^{s} \frac{(p_i R_i) \bmod q_i}{q_i}$$

holds true. Therefore, in order to compute the polynomial P, it suffices to find the products $r_i = p_i R_i \bmod q_i$, and this can be done in the same way as in Sections 6, 7 with complexity $\mathrm{GD}(n) + 7\,\mathrm{M}(n) + 5n + l(k)(10\,\mathrm{M}(n) + 5n) - \mathrm{M}(n)$, and then compute the sum of proper fractions of degree n_i

$$\frac{P}{Q} = \sum_{i=1}^{k} \frac{r_i}{q_i}$$

just as in Section 10 with complexity $l(k)(3\,\mathrm{M}(n) + n)$. The final estimate has the form

$$\mathrm{GD}(n) + l(k)(13\,\mathrm{M}(n) + 6n) + 6\,\mathrm{M}(n) + 5n.$$

When $q_i(x) = x - c_i$ and c_i and p_i belong to the field of coefficients, the polynomial P, which is constructed with the aid of the inverse Chinese algorithm, is known to be the solution of the interpolation problem $P(c_i) = p_i$. In this case, the computation of $R_i = Q_i^{-1} \bmod q_i$ from the given $Q_i \bmod q_i$ instead of the application of the extended algorithm for finding the GCD of polynomials requires only n division operations. It follows from what was said above that the complexity of its solution can be estimated as

$$l(n)(13\,\mathrm{M}(n) + 6n) + 6\,\mathrm{M}(n) + 6n.$$

If n is the power of 2, then, with due account of the remark made in Section 6, we can lower slightly this estimate.

The direct Chinese algorithm uses the polynomial P to compute its remainders from the given moduli q_i and, in particular, its values at the given points c_i. The estimate of its complexity was actually obtained in Section 6.

Note that all the results given above can be generalized to the numerical version of the Chinese remainder theorem and to the numerical version of the problem of the decomposition of a fraction into the sum of partial fractions. It should also be pointed out that in [13] the question concerning the fast Chinese algorithm (in the version without preliminary information processing) is presented not very clearly and the problem of the decomposition of a fraction into partial fractions is not considered at all (articles [17, 18] appeared later).

14. Estimate of the complexity of multiplication of polynomials and the algorithm of the Chinese remainder theorem. In order to compute the polynomial r, which is the product of the polynomials p and q of a degree smaller than $n/2$, it is natural to perform their multiplication modulo $x^n - 1$, for which purpose we must first calculate the remainders of the division of these polynomials by linear pairwise coprime binomials $x - \varepsilon^k$ resulting from the factorization of the polynomial $x^n - 1$ (say, over the field $\mathbb{C}$), i.e., simply calculate the values $p(\varepsilon^k)$ and $q(\varepsilon^k)$, $k = 0, \dots, n-1$, and then, multiplying them pairwise, obtain the values $r(\varepsilon^k) = p(\varepsilon^k)q(\varepsilon^k)$ and, using the "inverse" Chinese algorithm, restore the polynomial r. It is convenient to assume n to be equal to the power of 2 (adding, when necessary, the zero lowest coefficients to the polynomials p and q).

Applying the technique presented in Section 6 to compute the values $p(\varepsilon^k)$ and $q(\varepsilon^k)$ for $k = 0, \dots, n-1$, we first find

$$p \bmod x^{n/2} - 1, \quad p \bmod x^{n/2} + 1, \qquad q \bmod x^{n/2} - 1, \quad q \bmod x^{n/2} + 1,$$

then calculate the remainders moduli $x^{n/4} + 1$, $x^{n/4} - 1$, $x^{n/4} + i$, $x^{n/4} - i$, and so on, until we find remainders moduli $x^2 - \varepsilon^{2k}$, $k = 0, \dots, n/2$, $x - \varepsilon^k$, $k = 0, \dots, n$. Since the division of a polynomial of a degree lower than m by a binomial of degree $m/2$ is carried out by a "school" algorithm with complexity m, the complexity of the whole algorithm for computing the values $p(\varepsilon^k)$ and $q(\varepsilon^k)$, $k = 0, \dots, n-1$, is estimated as $2n \log_2 n$ (multiplications and divisions carried out in the field $\mathbb{C}$).

For the restoration (interpolation) of the polynomial r from its known values $r(\varepsilon^k)$, $k = 0, \dots, n-1$, it is natural to use the Lagrange formula

$$r = f(x) \sum_{k=0}^{n-1} \frac{r(\varepsilon^k)/f'(\varepsilon^k)}{x - \varepsilon^k} = f(x) \sum_{k=0}^{n-1} \frac{r(\varepsilon^k)\varepsilon^k}{n(x - \varepsilon^k)},$$

where $f(x) = x^n - 1$, and perform the summation of the fractions in the same way as we did in Section 10. Since, when the order of summation is properly chosen, all the resulting fractons have binomial denominators and the multiplication of a polynomial of a degree lower than m by the "school" method is of complexity m, the addition of two fractions of this kind has the complexity $4m$, and the whole interpolation algorithm has the complexity $2n + 2n \log_2 n$. As a result, we get the well-known relation $\mathrm{M}(n) = O(n \log n)$.

We see that the fast multiplication of polynomials is actually carried out with the aid of the Chinese remainder theorem and the method of division in half which was first applied to this problem by Karatsuba [4]. As Karatsuba recounts in his articale [5], Kolmogorov spoke of the possibility of application of the Chinese remainder theorem to multiplication of numbers and initiated the research in this field.

The descrete Fourier transformation was not mentioned in the given proof, although, actually, the caculation of the values of a polynomial in nth degree roots of unity, which was carried out above, coincides with the discrete Fourier transformation, and the technique, which can be used for this purpose, coincides with one of the variety of algorithms of Fourier fast transformations known today (the monograph [24] is devoted to these algorithms). A similar algorithm for Fourier fast transformation is described in [13]. The interpolation of a polynomial from the values in nth-degree roots of unity coincides with Fourier inverse transformation, and therefore, the algorithm indicated above could not be used for its performance since the estimate of the complexity of the inverse Fourier transformation coincides, as is well known, with the estimate of the copmplexity of the direct Fourier transformation. The application of the Chinese theorem in this situation is equivalent to the application of the so-called theorem on a cyclic convolution.

ACKNOWLEDGMENTS

This work was supported by the Russian Foundation for Fundamental Research (project no. 97-01-00721).

REFERENCES

1. Ostrogradskii, M. V., On the Integration of Rational Fractions, in: *Selected Works*, Moscow: Akad. Nauk SSSR, 1958, pp. 56–105.
2. Ostrogradskii, M. V., Note on the Equal Factors of Integral Polynomials, in: *Selected Works*, Moscow: Akad. Nauk SSSR, 1958, pp. 106–107.
3. Hermite, C., *Quevres de Charles Hermite*, vol. 3 (Picard, E., Ed.), Paris: Gautnier-Villars, Imprimeur-Librae, 1912.
4. Karatsuba, A. A. and Ofman, Yu. P., Multiplication of Multiplace Numbers with the Use of Automata, *Dokl. Akad. Nauk SSSR*, 1962, vol. 145, no. 2, pp. 293–294.
5. Karatsuba, A. A., Complexity of Computations, *Trudy Mat. Inst. Rus. Akad. Nauk*, 1995, vol. 211, pp. 186–202.
6. Toom, A. L., On the Complexity of the Scheme of Functional Elements Which Realizes the Multiplication of Integers, *Dokl. Akad. Nauk SSSR*, 1963, vol. 150, no. 2, pp. 496–498.
7. Schönhage, A. and Strassen, V., Schnelle Multiplikation grossen Zahlen, *Computing*, 1971, vol. 7, no. 3–4, pp. 281–292.
8. Schönhage, A., Schnelle Multiplikation grossen Zahlen, *Computing*, 1966, vol. 1, pp. 182–196.
9. Cook, S. A., *On the Minimum Computation Time of Functions*, Thesis Harvard Univ., 1966, pp. 51–77.
10. Benderskii, Yu. V., Fast Computations, *Dokl. Akad. Nauk SSSR*, 1975, vol. 223, no. 5, pp. 1041–1043.
11. Kung, H. T., *Fast Evaluation and Interpolation*, Pittsburgh: Dept. Comput. Sci., Carnegie-Mellon Univ., 1973.
12. Strassen, V., Gaussian Elimination is not Optimal, *Numer. Math.*, 1969, vol. 13, no. 4, pp. 354–356.
13. Aho, A., Hopcroft, J., and Ullman, J., *The Design and Analysis of Computer Algorithms*, USAM: Addison-Wesley, 1975.
14. Knuth, D., *The Art of Computer Programming*, vol. 2, Reading: Addison-Wesley, 1969.
15. Schönhage, A., Grotefeld, A. F. W., and Vetter, E., *Fast Algorithms*, B. 1, Manheim, Leipzig, Wien, Zürich: Wissenschaftsverlag, 1994.
16. Yun, D. Y. Y., Fast Algorithm for Rational Function Integration, in: *Proc. IFIP Congr.*, Amsterdam: North-Holland, 1977, vol. 7, pp. 493–498.
17. Kung, H. T. and Tong, D. M., Fast Algorithms for Fraction Decompositon, *SIAM J. Comput.*, 1977, vol. 6, pp. 582–593.
18. Chin, P. I., Complexity of Partial Fraction Expansion, *SIAM J. Comput.*, 1977, vol. 6, pp. 554–562.
19. Moenk, R., Fast Algorithm of GCD's, in: *Proc. 5th Ann. ACM Sympos. on Theory of Computing*, 1973, pp. 142–151.
20. Schönhage, A., Schnelle berechnung von kettenbruchentwicklungen, *Acta Inform.*, 1971, vol. 1, pp. 139–144.
21. Gashkov, C. B. and Chubarikov, V. N., *Arifmetika, Algoritmy. Slozhnost Vychislenii* (Arithmetic Algorithms. Complexity of Computations), Moscow: Nauka, 1996.
22. Aho, A. V., Steiglitz, K., and Ullman, J. D., Evaluation of Polynomials at Fixed Set of Points, *SIAM J. Comput.*, 1975, vol. 6, pp. 533–539.
23. Davenport, J., *On the Integration of Algebraic Functions*, Heidelberg: Springer, 1981.
24. Blahut, R., *Bystrye algoritmy tsifrovoi obrabotki signalov* (Fast Algorithms of Digital Signal Proc.), Moscow: Mir, 1989.

Translated by I. Aleksandrova