Formal Power Series and Algebraic Combinatorics
Séries Formelles et Combinatoire Algébrique
Vancouver 2004

Utilizing Relationships Among Linear Systems Generated by Zeilberger's Algorithm

S. A. Abramov and H. Q. Le

Abstract. We show that the sequence of first order linear difference equations generated by Zeilberger's algorithm can be described recursively. Each of these difference equations induces a system of linear algebraic equations and the mentioned recurrent relations can be utilized so that the values computed during the investigation of the J-th system can be used to accelerate the investigation of the $(J+1)$-th system. An implementation of this result and an experimental comparison between this implementation and an implementation of the original Zeilberger's algorithm are also done.
Résumé. Nous montrons que la suite des équations linéaires aux différences du premier ordre produites par l'algorithme de Zeilberger peut être décrite de façon récursive. Chacune de ces équations aux différences induit un système d'équations linéaires algébriques et les dites relations de récurrence peuvent être employées de façon que les valeurs calculées pendant l'analyse du J-ème système puisse être utilisées pour accélérer l'analyse du $(J+1)$-ème système. Nous faisons aussi une implantation de ce résultat et une comparaison expérimentale de cette implantation et de l'implantation originale de l'algorithme de Zeilberger.

1. Introduction

Zeilberger's algorithm, named hereafter as $\mathcal{Z}$, has been shown to be a very useful tool in a wide range of applications. These include finding closed forms of definite sums of hypergeometric terms, certifying large classes of identities in combinatorics and in the theory of special functions [**Z91, PWZ**].

For a hypergeometric term (or simply a *term*) $F(n,k)$, $\mathcal{Z}$ tries to find

$$A_0(n), \ldots, A_J(n) \in \mathbb{K}(n), \quad A_J(n) \neq 0, \tag{1.1}$$

and a term $S(n,k)$ such that

$$A_J(n)F(n+J,k) + \cdots + A_0(n)F(n,k) = S(n,k+1) - S(n,k). \tag{1.2}$$

The algorithm uses an item-by-item examination on the values of J. It starts with the value of 0 for J, and keeps on incrementing J until it is successful in finding the $A_0, \ldots, A_J$ and $S(n,k)$ such that (1.2) holds. For a particular value of J under investigation, $\mathcal{Z}$ constructs a system of linear algebraic equations whose coefficients are in $\mathbb{K}(n)$, and its right hand side linearly depends on parameters $A_0, \ldots, A_J$. It then checks for the existence of (1.1) such that the linear system is consistent (see [**Z91, PWZ**] for details). This operation is expensive if the value of J is large.

While the problem of applicability of $\mathcal{Z}$ to a term has been completely solved [**A03**], the issue of efficiency is still an on-going work. For the case where the input term is also a rational function, there is a direct

1991 *Mathematics Subject Classification.* Primary 68W30, 33F10.

Key words and phrases. Zeilberger's algorithm, hypergeometric term, linear systems.

The first author was supported in part by the French-Russian Lyapunov Institute under grant 98-03.

algorithm [**L03**] which avoids the item-by-item examination strategy. For the non-rational hypergeometric case, even though there is an algorithm which computes a non-trivial lower bound J_0 for J [**AL02**], $\mathcal{Z}$ still wastes resource on the fruitless examination at steps $J_0, J_0+1, \ldots, J-1$.

The examination done at each step is independent of that at other steps. However, there are relationships between two consecutive steps, and it would be logical to try to utilize them. It is shown in this paper that after we considered the system corresponding to step J and found that it is not consistent, we can use some intermediate results of this step in order to either reduce the size of the linear system at step $(J+1)$ or simplify this system. In this context, "simplify" means the elimination of the parameters $A_0, \ldots, A_J$ in a number of equations of the $(J+1)$-th system.

Throughout the paper, $\mathbb{K}$ is a field of characteristic zero, $\mathbb{N}$ is the set of nonnegative integers. E_n, E_k denote the shift operators w.r.t. n and k, respectively, defined by $E_n F(n,k) = F(n+1,k)$, $E_k F(n,k) = F(n,k+1)$.

The basic idea of this work was presented in our poster at FPSAC 2003 [**AL03**]. In this paper, this idea is further extended. The derivation of the relationships between two consecutive steps is significantly simplified. A complete Maple implementation and an extensive experimental comparison with an implementation of the original Zeilberger's algorithm are added.

The Maple source code, the help page, and the test results reported in this paper are available, and can be downloaded from

`http://www.scg.uwaterloo.ca/~hqle/code/Linsys/Linsys.html`.

2. Step-by-step examination in $\mathcal{Z}$

2.1. Reduction to a linear algebra problem. For a term $F(n,k)$ and for a particular value of $J \in \mathbb{N}$, set

$$T_J(n,k) = A_J(n)F(n+J,k) + \cdots + A_1(n)F(n+1,k) + A_0(n)F(n,k). \tag{2.1}$$

$\mathcal{Z}$ attempts to compute the A_i's $\in \mathbb{K}(n)$ in (2.1) and a term S such that (1.2) holds. Since F is a term, T_J is also a term [**Z91**]. This allows $\mathcal{Z}$ to use Gosper's algorithm [**G77**] to attain its goal. Given the term T_J in (2.1), Gosper's algorithm determines if there exists a term S_J such that

$$T_J = (E_k - 1)\, S_J, \tag{2.2}$$

and computes S_J if it exists. The algorithm transforms (2.2) into the problem of computing a polynomial solution of a first-order linear recurrence equation with polynomial coefficients and polynomial right hand side (2.4). The process can be summarized as follows.

(1) Compute a PNF$_k$ (also known as Gosper form) of the rational k-certificate $T_J(n,k+1)/T_J(n,k)$. This results in a triple (a_J, b_J, c_J), a_J, b_J, $c_J \in \mathbb{K}(n)[k] \setminus \{0\}$ such that

$$\frac{T_J(n,k+1)}{T_J(n,k)} = \frac{a_J}{b_J} \cdot \frac{E_k c_J}{c_J}, \;\; \gcd(a_J, E_k^h b_J) = 1 \text{ for all } h \in \mathbb{N}. \tag{2.3}$$

See [**PWZ**] for a description of such a construction.

(2) Find a polynomial solution $y(k)$ of the linear recurrence

$$a_J(k)\, y(k+1) - b_J(k-1)\, y(k) = c_J(k) \tag{2.4}$$

provided that such a solution exists.

If it does, then set

$$L_J = A_J(n)\, E_n^J + \cdots + A_1(n)\, E_n + A_0(n), \tag{2.5}$$

$$S_J = \frac{b_J(k-1)\, y(k)}{c_J(k)}\, T_J. \tag{2.6}$$

The computed $\mathcal{Z}$-pair (L_J, S_J) defined in (2.5) and (2.6) is the output from $\mathcal{Z}$. The recurrence operator L_J is called a telescoper for the input term F.

The search for a polynomial solution $y(k)$ of (2.4) can be done using the method of undetermined coefficients. First one computes an upper bound d for the degree of the polynomial $y(k)$. Then one substitutes a generic polynomial of degree d for $y(k)$ into (2.4), equates the coefficients of like powers in k. This results in a system of linear algebraic equations. The problem is reduced to determining if this linear system is consistent. If it is, then compute a solution of the system. Note that this enables one to compute not only a polynomial solution $y(k)$ in (2.4), but also the unknowns A_i's in (2.1).

2.2. Simplificators and the J-increment of a system. The system of linear algebraic equations at step J is of the form

$$M_J\, x_J = u_J \tag{2.7}$$

where M_J is a $\nu \times \kappa$ matrix whose entries are in the field $\mathbb{K}(n)$, and u_J is a column vector where each of its ν entries is in the $\mathbb{K}(n)$-linear space U and of the form

$$R_0 A_0 + \cdots + R_J A_J, \quad R_0, \ldots, R_J \in \mathbb{K}(n). \tag{2.8}$$

We call the system (2.7) a J-parameterized system. If it is consistent, then the system is said to be J-solvable.

The following definition provides important concepts used in this paper.

Definition 2.1. For a J-parameterized system S of the form (2.7), a column vector $y_J \in U^\kappa$ is a *simplificator* of S if the first entry of $u_J - M_J\, y_J$ is zero. The *height* of y_J is the number of all initial entries of $u_J - M_J\, y_J$ each of which equals zero. The *J-increment* of S is the number of all initial entries of u_J which do not depend on $A_0, \ldots, A_{J-1}$.

Suppose the recognition of the J-solvability of system (2.7) is done by an elimination process. During this process we can get an equation of the form

$$0 = \tilde{R}_0 A_0 + \tilde{R}_1 A_1 + \cdots + \tilde{R}_{J-1} A_{J-1} + \tilde{R}_J A_J, \quad \tilde{R}_i \in \mathbb{K}(n). \tag{2.9}$$

Such an equation is called *trivial* if $\tilde{R}_0 = \cdots = \tilde{R}_J = 0$; *irregular* if ($\tilde{R}_0 = \cdots = \tilde{R}_{J-1} = 0$ and $\tilde{R}_J \neq 0$) or if ($\tilde{R}_1 = \cdots = \tilde{R}_J = 0$ and $\tilde{R}_0 \neq 0$); and *regular* otherwise. The existence of an irregular equation implies that the system is not J-solvable.

Although the equations might change their orderings during the elimination process, we assign to each equation a label which is the number of this equation in the original system, and hence are still able to keep track of its position. The process results in two systems W and V: W is a trapezoidal system of regular equations; and the equations of V are those obtained during the elimination process, but not of the form (2.9).

If W is consistent with $A_J \neq 0$, $A_0 \neq 0$, then the original system is J-solvable. Otherwise, it is not J-solvable, and we can construct a simplificator of the system as follows.

(i) Find the maximal N such that equations labeled $1, \ldots, N$ are in V;
(ii) For all $i = 1, \ldots, N$, the unknown x_i was eliminated by an equation with label j, $1 \leq j \leq N$.

This results in a system V', a subsystem of V and consisting of equations labeled $1, \ldots, N$. The vector $(x_1, \ldots, x_N, 0, \ldots, 0)^T$ is evidently a simplificator of height $\geq N$ of the original system.

3. A simplification scheme

3.1. Relationships among J-parameterized systems. Let $F(n, k)$ be the input term. At step J of the item-by-item examination, $\mathcal{Z}$ tries to compute a telescoper L_J of the form (2.5) for F. The k-certificate $(E_k T_J)/T_J$ of the term $T_J(n, k) = L_J\, F$ can be written in the form

$$\frac{v_J(n,k)}{w_J(n,k)} = \frac{\varphi_J(n,k)}{\psi_J(n,k)} \, \frac{p_J(A_0, \ldots, A_J, n, k+1)}{p_J(A_0, \ldots, A_J, n, k)} \tag{3.1}$$

where v_J, $w_J \in \mathbb{K}[n,k]$; $\varphi_J(n,k), \psi_J(n,k) \in \mathbb{K}[n,k]$ and do not depend on $A_0,\ldots,A_J$; p_J is in the $\mathbb{K}(n,k)$-space of linear forms in $A_0,\ldots,A_J$.

Let $s_1(n,k)$, $s_2(n,k)$ be relatively prime polynomials such that

$$\frac{F(n,k)}{F(n-1,k)} = \frac{s_1(n,k)}{s_2(n,k)}.$$

Then we can derive the following recurrences:

$$\begin{aligned} p_{J+1}(A_0,\ldots,A_{J+1},n,k) &= p_J(A_0,\ldots,A_J,n,k)\, s_2(n+J+1,k) + \\ & \quad A_{J+1}\prod_{i=1}^{J+1} s_1(n+i,k), \end{aligned} \tag{3.2}$$

$$\varphi_{J+1}(n,k) = \varphi_J(n,k)\, s_2(n+J+1,k), \tag{3.3}$$

$$\psi_{J+1}(n,k) = \psi_J(n,k)\, s_2(n+J+1,k+1). \tag{3.4}$$

(They are similar to (6.3.6)–(6.3.8) in [**PWZ**].) Let

$$\mathrm{PNF}_k\left(\frac{\varphi_J}{\psi_J}\right) = \frac{a_J(k)}{b_J(k)}\,\frac{\xi_J(k+1)}{\xi_J(k)}. \tag{3.5}$$

It follows from (3.3), (3.4) and (3.5) that

$$a_J(k) = a_0(k)\frac{s_2(n+J,k)\cdots s_2(n+1,k)}{s_2(n+J,k+1)\cdots s_2(n+1,k+1)}\,\frac{\xi_0(k+1)}{\xi_0(k)}\,\frac{\xi_J(k)}{\xi_J(k+1)}\,\frac{b_J(k)}{b_0(k)}. \tag{3.6}$$

Let a,b be polynomials in k. Define

$$G_{a(k),b(k)} = a(k)E_k - b(k-1). \tag{3.7}$$

By (3.6) and (3.7), we obtain the following theorem which shows the relationsips between $G_{a_J(k),b_J(k)}$ and $G_{a_{J+1}(k),b_{J+1}(k)}$.

Theorem 3.1. *The operators $G_{a_J(k),b_J(k)}$ and $G_{a_{J+1}(k),b_{J+1}(k)}$ for $J\in\mathbb{N}$ are related by the following recurrence:*

$$G_{a_J(k),b_J(k)} = \frac{\xi_J(k)}{s_2(n+J+1,k)\xi_{J+1}(k)}G_{a_{J+1}(k),b_{J+1}(k)} \circ \frac{s_2(n+J+1,k)\xi_{J+1}(k)b_J(k-1)}{\xi_J(k)b_{J+1}(k-1)}. \tag{3.8}$$

3.2. Polynomial simplification. At step J of the item-by-item examination, it follows from (2.4) and (3.7) that the recurrence

$$G_{a_J(k),b_J(k)}y(k) = c_J(k) \tag{3.9}$$

where $c_J(k) = \xi_J(k)p_J(k)$, $J\in\mathbb{N}$, is considered. By (3.2)

$$c_{J+1}(k) = \frac{\xi_{J+1}(k)}{\xi_J(k)}s_2(n+J+1,k)c_J(k) + \xi_{J+1}(k)A_{J+1}\prod_{i=1}^{J+1} s_1(n+i,k). \tag{3.10}$$

If the right hand side $c_J(k)$ of the J-th recurrence (3.9) is simplified by means of a polynomial $f_J(k)$, then it gets transformed to $c'_J(k)$ where

$$c'_J(k) = c_J(k) - G_{a_J(k),b_J(k)}f_J(k), \quad \deg_k c_J > \deg_k c'_J. \tag{3.11}$$

It follows from (3.2), (3.8) and (3.11) that if we replace $c_J(k)$ by $c'_J(k)$ in the right hand side of (3.10), then the first term $\frac{\xi_{J+1}(k)}{\xi_J(k)}s_2(n+J+1,k)c'_J(k)$ of this right hand side equals

$$\xi_{J+1}(k)s_2(n+J+1,k)p_J(k) - G_{a_{J+1},b_{J+1}}\frac{s_2(n+J+1,k)\xi_{J+1}(k)b_J(k-1)}{\xi_J(k)b_{J+1}(k-1)}f_J(k).$$

This induces the change of $c_{J+1}(k)$ by $\tilde{c}_{J+1}(k)$ where

$$\tilde{c}_{J+1}(k) = c_{J+1}(k) - G_{a_{J+1},b_{J+1}} \frac{s_2(n+J+1,k)\xi_{J+1}(k)b_J(k-1)}{\xi_J(k)b_{J+1}(k-1)} f_J(k). \tag{3.12}$$

Once a polynomial $g_{J+1}(k)$ is found such that for

$$c'_{J+1}(k) = \tilde{c}_{J+1}(k) - G_{a_{J+1},b_{J+1}} g_{J+1}(k),$$

we have $\deg_k c'_{J+1} < \deg_k c_{J+1}$. Then the right hand side $c_{J+1}(k)$ of the $(J+1)$-th recurrence $G_{a_{J+1}(k),b_{J+1}(k)} y(k) = c_{J+1}(k)$ will be simplified by means of the polynomial $f_{J+1}(k)$ where

$$f_{J+1}(k) = \frac{s_2(n+J+1,k)\xi_{J+1}(k)b_J(k-1)}{\xi_J(k)b_{J+1}(k-1)} f_J(k) + g_{J+1}(k).$$

Let $\deg_k c_J - \deg_k c'_J = H_J > 0$. Let the two terms in the right hand side of (3.10) be R and S, i.e.,

$$R = \frac{\xi_{J+1}(k)}{\xi_J(k)} s_2(n+J+1,k)c_J(k), \;\; S = \xi_{J+1}(k)A_{J+1}\prod_{i=1}^{J+1} s_1(n+i,k).$$

Note that S is independent of $A_0,\ldots,A_J$. By comparing the degrees of R and S in (3.10), we obtain the following theorem which reflects changes to the $(J+1)$-system because of the replacement of c_J by c'_J.

THEOREM 3.2. *Suppose it is recognized that the J-system of the form (2.7) is not J-solvable, and that a simplificator $y_J(k)$ of height $H_J > 0$ for this system is computed.*

(1) $\deg_k S > \deg_k R$: *let σ_J, σ_{J+1} be the J-increment of the J-system, and the $(J+1)$-increment of the $(J+1)$-system, respectively. Then*

$$\sigma_{J+1} = \deg_k S - \deg_k R + \max\{H_J, \sigma_J\},$$

i.e., if $H_J > \sigma_J$ then the $(J+1)$-increment of the $(J+1)$-system is increased, and we have a simpler $(J+1)$-system;

(2) $\deg_k S \leq deg_k R$: *the degree of c_{J+1} w.r.t. k is decreased by $\min\{H_J, \deg_k R - \deg_k S\}$. This leads to a system of linear algebraic equations of smaller size to be solved.*

4. Implementation

We implemented the result of this paper in the computer algebra system Maple [**M**], and performed experiments of our program (called M) on four different sets of data. A comparison between this implementation and the one of the original $\mathcal{Z}$ (called Z) in Maple 9 (the function `Zeilberger` in the `SumTools:-Hypergeometric` module) was also done. Note that the development of M is based on Z.

The result shows that it is worthwhile incorporating the simplification scheme presented in this paper into $\mathcal{Z}$.

EXPERIMENT 1. The first set of input consists of seven hypergeometric terms

$$T_i(n,k) = \binom{2\,n}{2\,k}^i, \;\; 2 \leq i \leq 8.$$

Table 1 shows the time and space requirements[1]. $\operatorname{ord} L_i$ indicates the order of the computed minimal telescoper L_i of the input term T_i.

Each input term in the following three sets of data is an r-term [**A03**]. Since every hypergeometric term is conjugate to an r-term, i.e., they share the same rational certificates, and since $\mathcal{Z}$ in principal only works with the certificates of the input term, the sets of data we use can be considered to cover all possible forms of input hypergeometric terms.

[1]All the reported timings were obtained on a 400Mhz SUN SPARC SOLARIS with 1Gb RAM.

TABLE 1. First experiment: time and space requirements of Z and M

i	ord L_i	Timing (seconds)		Memory (kilobytes)	
		Z	M	Z	M
2	2	0.54	0.54	2,977	2,959
3	3	3.57	2.81	18,168	15,335
4	4	31.24	23.35	179,221	132,859
5	5	199.40	142.13	1,021,542	762,488
6	6	1,523.86	1,242.03	6,429,441	4,902,558
7	7	8,563.81	6,205.83	28,326,178	18,530,142
8	8	42,122.52	36,917.47	92,161,603	66,414,167
Total time		52,444.94	44,534.16		

EXPERIMENT 2. The second set of tests consists of twenty randomly-generated hypergeometric terms each of which is of the form

$$T_i(n,k) = \frac{1}{(a_i\, n + b_i\, k + c_i)!}, \ -15 \le a_i, b_i, c_i \le 15, \ |b_i| \ge 6.$$

Table 2 shows the time and space requirements.

TABLE 2. Second experiment: time and space requirements of Z and M

i	ord L_i	Timing (seconds)		Memory (kilobytes)	
		Z	M	Z	M
1	3	1.05	0.72	5,722	4,176
2	5	19.43	19.58	98,068	105,292
3	6	116.98	91.04	561,628	552,879
4	8	196.16	118.29	787,282	675,806
5	3	7.58	7.02	54,561	53,492
6	6	15.23	14.45	79,553	70,003
7	8	34.68	16.30	173,941	105,203
8	3	1.99	1.46	10,592	9,270
9	10	3,163.60	1,369.30	7,799,715	5,418,995
10	6	79.05	65.70	342,584	287,447
11	15	14,558.05	4,568.70	23,774,518	14,933,116
12	13	4,503.45	3,226.63	10,566,736	10,922,477
13	7	29.58	31.93	170,337	177,498
14	5	166.36	155.62	693,555	761,711
15	7	5.90	5.38	29,689	27,090
16	11	2,456.17	1,402.33	6,576,152	5,966,455
17	3	17.33	15.53	92,278	101,312
18	3	3.04	4.16	30,133	32,015
19	13	133.53	95.87	640,949	536,200
20	3	10.79	10.66	61,675	56,596
Total time		25,519.95	11,220.67		

EXPERIMENT 3. The third set of tests consists of twenty randomly-generated hypergeometric terms each of which is of the form

$$T_i(n,k) = \frac{(a_{i1}n + b_{i1}k + c_{i1})}{(a_{i3}n + b_{i3}k + c_{i3})} \frac{(a_{i2}n + b_{i2}k + c_{i2})!}{(a_{i4}n + b_{i4}k + c_{i4})!}$$

where $-5 \le a_{ij}, b_{ij}, c_{ij} \le 5$, $1 \le j \le 4$. Table 3 shows the time and space requirements.

TABLE 3. Third experiment: time and space requirements of Z and M

i	ord L_i	Timing (seconds)		Memory (kilobytes)	
		Z	M	Z	M
1	7	251.94	213.26	1,149,488	1,105,805
2	6	362.69	259.91	1,744,877	1,258,468
3	2	0.89	0.78	4,529	4,184
4	6	41.57	31.33	214,770	180,945
5	4	21.26	13.89	106,702	72,569
6	8	261.54	185.26	1,426,237	902,509
7	6	87.39	49.76	449,139	285,798
8	5	98.97	63.83	527,146	308,765
9	9	740.47	708.66	3,580,681	3,488,491
10	5	5.36	4.39	24,382	22,782
11	2	0.70	0.58	3,661	3,380
12	5	61.74	48.81	301,470	251,228
13	4	14.08	11.54	76,225	67,605
14	8	1,191.93	1,098.12	5,615,755	5,450,072
15	8	2,424.06	2,157.03	9,813,850	9,280,051
16	8	1,470.97	1,185.56	7,071,945	5,827,483
17	7	1.60	1.51	7,987	7,864
18	1	0.71	0.52	3,966	3,093
19	6	180.37	145.82	778,584	673,263
20	5	7.88	7.74	43,221	39,400
Total time		7,226.12	6,188.30		

EXPERIMENT 4. The fourth set of tests consists of twenty randomly-generated hypergeometric terms each of which is of the form

$$T_i(n,k) = \frac{(a_{i1}n + b_{i1}k + c_{i1})!\,(a_{i2}n + b_{i2}k + c_{i2})!}{(a_{i3}n + b_{i3}k + c_{i3})!\,(a_{i4}n + b_{i4}k + c_{i4})!}$$

where $-6 \le a_{ij}, b_{ij}, c_{ij} \le 6$, $1 \le j \le 4$. Table 4 shows the time and space requirements.

References

[A03] S.A. Abramov. When does Zeilberger's algorithm succeed? *Advances in Applied Mathematics* **30** (2003) 424–441.

[AL03] S.A. Abramov, H.Q. Le. The sequence of linear algebraic systems generated by Zeilberger's algorithm. *Proceedings of the 2003 Formal Power Series and Algebraic Combinatorics*, on CD (Poster session).

[AL02] S.A. Abramov, H.Q. Le. A lower bound for the order of telescopers for a hypergeometric term. *Proceedings of the 2002 Formal Power Series and Algebraic Combinatorics*, on CD.

[G77] R.W. Gosper, Jr. Decision procedure for indefinite hypergeometric summation. *Proc. Natl. Acad. Sci. USA* **75** (1977) 40–42.

[L03] H.Q. Le. A direct algorithm to construct the minimal Z-pairs for rational functions. *Advances in Applied Mathematics* **30** (2003) 137–159.

TABLE 4. Fourth experiment: time and space requirements of Z and M

i	ord L_i	Timing (seconds)		Memory (kilobytes)	
		Z	M	Z	M
1	9	17.19	10.80	89,010	66,383
2	8	529.95	433.59	2,434,037	1,954,207
3	7	74.33	46.76	453,495	301,637
4	8	323.57	258.28	1,354,762	1,200,625
5	6	184.65	135.11	996,090	786,864
6	9	2,934.08	1,221.24	14,901,781	5,977,105
7	7	223.33	190.57	1,081,266	910,766
8	9	9,338.72	7,982.59	31,056,490	28,264,207
9	6	52.94	37.09	239,317	164,542
10	7	308.47	236.07	1,506,582	1,171,261
11	7	2,070.33	709.82	10,329,829	3,364,322
12	4	14.13	11.44	79,847	62,823
13	9	1,865.57	1,712.06	9,582,506	8,528,627
14	7	50.60	40.76	269,926	216,128
15	7	171.14	138.65	823,582	674,723
16	6	39.51	30.28	211,825	175,698
17	8	943.22	690.12	5,363,613	3,628,208
18	5	89.86	59.02	446,246	307,635
19	11	17,514.39	16,398.59	63,496,067	58,960,912
20	6	133.81	88.30	643,233	473,633
Total time		36,879.79	30,431.14		

[M] M.B. Monagan, K.O. Geddes, K.M. Heal, G. Labahn, S.M. Vorkoetter, J. McCarron, P. DeMarco. *Maple 9 Introductory Programming Guide*. Toronto: Maplesoft, a division of Waterloo Maple Inc., 2003.

[PWZ] M. Petkovšek, H. Wilf, D. Zeilberger. *A=B*. A.K. Peters, Wellesley, Massachusetts, 1996.

[Z91] D. Zeilberger. The method of creative telescoping. *Journal of Symbolic Computation* **11** (1991) 195–204.

Dorodnicyn Computing Centre, Russian Academy of Sciences, Vavilova 40, 119991, Moscow, GSP-1, Russia
E-mail address: abramov@ccas.ru

Algorithms Project, INRIA Rocquencourt, 78153 Le Chesnay Cedex, France
E-mail address: ha.le@inria.fr