An Efficient Algorithm for Computing the Reliability of Consecutive-k-out-of-n:F Systems

Thomas Cluzeau, Jörg Keller, *Member, IEEE Computer Society,* and Winfrid Schneeweiss, *Senior Life Member, IEEE*

***Abstract*—Many algorithms for computing the reliability of linear or circular consecutive-k-out-of-n:F systems appeared in this *Transactions*. The best complexity estimate obtained for solving this problem is $\mathcal{O}(k^3\,\log(n/k))$ operations in the case of i.i.d. components. Using fast algorithms for computing a selected term of a linear recurrence with constant coefficients, we provide an algorithm having arithmetic complexity $\mathcal{O}(k\,\log(k)\,\log(\log(k))\,\log(n)+k^{\omega})$ where $2<\omega<3$ is the exponent of linear algebra. This algorithm holds generally for linear, and circular consecutive-k-out-of-n:F systems with independent but not necessarily identical components.**

***Index Terms*—consecutive-k-out-of-n:F systems.**

NOTATION

$\mathcal{O}(\cdot)$	asymptotic runtime, up to constant factor
$\mathrm{M}(k)$	time to multiply to polynomials of $degree \leq k$
$\mathrm{Prob}\{\cdot\}$	probability of an event
$E(\cdot)$	expected value of a random variable
$U_L(n)$, $U_C(n)$	unreliabilities of linear and circular systems with n components

I. INTRODUCTION

THE study of consecutive-k-out-of-n:F systems has caught the attention of researchers since the early 1980s (see [1]–[7], and references therein). These systems consist of n components arranged in an open line (linear case), or a circle (circular case); they fail if at least k consecutive components fail. Classical examples are sequences of microwave stations, pump stations for transporting oil by pipes [4], or rows of street lanterns.

More precisely, people have been interested in computing efficiently the reliability of such systems in terms of the failure probabilities of each component (see [8]–[14], and references therein). The arithmetic complexities of existing algorithms for different kinds of arrangements (linear or circular), and independent components (identical or non-identical) are summarized in Table I.

While comparing these complexity estimates, the reader must keep in mind that the parameter k is always assumed to be lower or equal to n. Normally, n is much larger than k; as in most applications, k typically represents a constant, and n represents a variable. Furthermore, as it is remarked in [15], these algorithms do not allow one to compute the desired reliability within a reasonable execution time when k and n become large, even when the components are i.i.d. with the same failure probability.

In this paper, we use computer algebra efficient techniques to derive an algorithm having complexity $\mathcal{O}(k\,\log(k)\,\log(\log(k))\,\log(n)\,+\,k^{\omega})$ for computing the reliability of linear or circular consecutive-k-out-of-n:F systems with independent components. The unreliability of such systems can be expressed depending on the unreliability of systems with $n-1,\ldots,n-k$ components (see [8, Equation 1a.] for example); this is straightforward by using probability arguments, but we show here that the corresponding linear recurrence equation of order k can be recovered using Shannon decomposition [16]. As a consequence, computing the unreliability of a system with n components boils down to computing the nth term of a linearly recurrent sequence of order k having constant coefficients. The latter problem has been well studied in computer algebra (see [17], [18], and references therein), and it has been shown that this task can be performed using only $\mathcal{O}(\mathrm{M}(k)\,\log(n)+k^{\omega})$[1] operations in the base field where $\mathrm{M}(k)$ is the number of operations needed to multiply two polynomials of degree at most k, and $2<\omega<3$ is the exponent of linear algebra or matrix multiplication exponent defined such that the number of operations needed to multiply two square matrices of size k is $\mathcal{O}(k^{\omega})$ [19, 12.1]. Using the Fast Fourier Transform, we further have $\mathrm{M}(k)=\mathcal{O}(k\,\log(k)\,\log(\log(k)))$ [19, 8.2-8.3] so that we obtain the complexity estimate announced. For systems having non-identical components, we replace a polynomial dependency in n by a logarithmic one, but the exponent of the polynomial dependency in k is greater. Consequently, our algorithm supersedes the existing ones when $k\ll n$, which is the case in most applications. In the case of identical components, compared to the existing algorithm in $\mathcal{O}(k^3\,\log(n/k))$, our algorithm has the same logarithmic dependency in n, but the exponent of the polynomial dependency in k is reduced.

The paper is organized as follows. In the next section,

Manuscript received xx; revised xx.

Thomas Cluzeau is with École Nationale Supérieure d'Ingénieurs de Limoges (ENSIL), Parc Ester Technopole, 16 rue d'Atlantis, 87068 Limoges, France.
Jörg Keller and Winfrid Schneeweiss are with FernUniversität in Hagen, School of Mathematics and Computer Science, 58084 Hagen, Germany.

[1]The original result in [17] holds for the case of homogeneous recurrences, and the exponent of k in the second summand is 2 instead of ω. The result for the non-homogeneous case is easily deduced in Section I.B below.

TABLE I
ARITHMETIC COMPLEXITIES OF DIFFERENT ARRANGEMENTS OF COMPONENTS

	identical	non-identical
linear	$\mathcal{O}(k^3\ \log(n/k))$ [14]	$\mathcal{O}(n)$ [8]
circular	$\mathcal{O}(n)$ [10]/$\mathcal{O}(k^3\ \log(n/k))$ [14]	$\mathcal{O}(n\,k)$ [9], [11]

we recall, and adapt needed facts, and results concerning the computation of a selected term of a linear recurrence. In Section 3, we first construct the linear recurrence satisfied by the unreliability of a linear consecutive-k-out-of-n:F system with independent components using Shannon decomposition. Then, we apply the results of Section 2 to yield an algorithm computing this unreliability, and we extend this to the circular case. Finally, Section 4 recalls the main contribution of the paper, and points out that fast algorithms in computer algebra may be used to improve a lot of computational methods.

II. LINEAR RECURRENCES WITH CONSTANT COEFFICIENTS

Let $\mathbb{K}$ be a field. We consider a linear recurrence

$$a_0\,u(n) + a_1\,u(n-1) + \cdots + a_r\,u(n-r) = b, \qquad (1)$$

with constant coefficients $a_0, \ldots, a_r, b$ in $\mathbb{K}$. We assume further that $a_0 \neq 0$, and $a_r \neq 0$ so that r is *the order* of the recurrence.

We aim at solving the following problem: given the linear recurrence (1), the "initial conditions" $u(0), \ldots, u(r-1)$, and a positive integer N, compute efficiently[2] $u(N)$. A classical fact consists in rewriting the recurrence (1) in matrix form; noting that $U_n = (u(n), \ldots, u(n+r-1))^T$, and $B = (0, \ldots, 0, \frac{b}{a_0})^T$ where v^T denotes the transpose of the vector v, we get

$$U_{n+1} = \mathcal{R}\,U_n + B, \qquad (2)$$

with

$$\mathcal{R} = \begin{pmatrix} 0 & 1 & 0 & 0 \\ \vdots & \ddots & \ddots & 0 \\ 0 & \ldots & 0 & 1 \\ -\frac{a_r}{a_0} & \ldots & \ldots & -\frac{a_1}{a_0} \end{pmatrix}.$$

The first vector U_n involving $u(N)$ is U_{N-r+1} so that computing $u(N)$ reduces to computing U_{N-r+1}. Moreover, applying (2) recursively, and using the fact that the entries of the $r \times r$ matrix $\mathcal{R}$ do not depend on n, we obtain

$$U_{N-r+1} = \mathcal{R}^{N-r+1}\,U_0 + \left(\sum_{i=0}^{N-r} \mathcal{R}^i\right) B, \qquad (3)$$

where $U_0 = (u(0), \ldots, u(r-1))^T$ is the known vector of initial conditions.

[2]Note that we are interested in the case where the value of N is large; in particular, we assume $r < N$.

A. Homogeneous case

In the homogeneous case, where b, and thus B, are zero, the problem boils down to computing the $(N-r+1)$th power of the companion matrix $\mathcal{R}$, and the product matrix-vector $\mathcal{R}^{N-r+1}\,U_0$. Note that the latter product can be computed using only $\mathcal{O}(r^2)$ operations [19]. This method is developed in [17], and the result is the following:

Lemma 1 ([17], Proposition 2.4, and 3.2): The nth power of a companion matrix of size k can be computed using $\mathcal{O}(\mathrm{M}(k)\ \log(n))$ operations where $\mathrm{M}(k)$ is the number of operations needed to multiply two polynomials of degree at most k. Consequently, when $b = 0$, the Nth term of the homogeneous linear recurrence (1) can be computed using $\mathcal{O}(\mathrm{M}(r)\ \log(N) + r^2)$ operations.

B. Non-homogeneous case

We now consider the case where b, and thus B, are non-zero. As a consequence of Lemma 1, we obtain the following proposition.

Proposition 1: The Nth term of the linear recurrence (1) can be computed in terms of the initial conditions $u(0), \ldots, u(r-1)$ using $\mathcal{O}(\mathrm{M}(r)\ \log(N) + r^{\omega})$ operations, where $\mathrm{M}(k)$ is the number of operations needed to multiply two polynomials of degree at most k, and ω is the exponent of linear algebra defined such that the number of operations needed to multiply two square matrices of size k is $\mathcal{O}(k^{\omega})$.

Proof: From (3), the problem reduces to computing $\mathcal{R}^{N-r+1}$; the sum $\sum_{i=0}^{N-r} \mathcal{R}^i$; and the two products matrix-vector $\mathcal{R}^{N-r+1}\,U_0$, and $\left(\sum_{i=0}^{N-r} \mathcal{R}^i\right) B$. From Lemma 1, $\mathcal{R}^{N-r+1}$ can be computed in $\mathcal{O}(\mathrm{M}(r)\ \log(N) + r^2)$ operations. Now, $\sum_{i=0}^{N-r} \mathcal{R}^i$ is a sum of terms in a geometric progression, and thus we have

$$\sum_{i=0}^{N-r} \mathcal{R}^i = (I_r - \mathcal{R})^{-1}\ \left(I_r - \mathcal{R}^{N-r+1}\right),$$

where I_r is the identity matrix of size r. Consequently, the sum can be deduced from $\mathcal{R}^{N-r+1}$; the inverse of $I_r - \mathcal{R}$; and the product of the two matrices $(I_r - \mathcal{R})^{-1}$, and $\left(I_r - \mathcal{R}^{N-r+1}\right)$. The latter inverse, and product can both be computed using $\mathcal{O}(r^{\omega})$ operations (see [20], [21], or [18, 2.3.1] for the complexity of computing the inverse) so that the complexity for computing $\sum_{i=0}^{N-r} \mathcal{R}^i$ is $\mathcal{O}(\mathrm{M}(r)\ \log(N) + r^{\omega})$. The result then follows because a product matrix-vector of size r can be computed in $\mathcal{O}(r^2)$ operations, and $2 < \omega < 3$ [19]. ∎

Note that, as we compute the whole vector U_{N-r+1}, we get the same complexity estimate for computing a slice of r consecutive elements $u(N-r+1), \ldots, u(N)$. This remark will be useful for the circular case in Section III.B (see the proof of Corollary 1).

III. APPLICATION TO CONSECUTIVE-k-OUT-OF-n:F SYSTEMS

In this section, we consider a consecutive-k-out-of-n:F system with (stochastically) independent components. We denote each component i by an indicator variable X_i, where $X_i = 0$ if the component is functional, and $X_i = 1$ otherwise.

A. *Case of a linear arrangement of the components*

The Boolean function of the fault-tree for the linear consecutive-k-out-of-n:F system is

$$X_L(n) = (X_1 \wedge X_2 \wedge \cdots \wedge X_k) \vee \cdots \vee (X_{n-k+1} \wedge \cdots \wedge X_n). \quad (4)$$

The goal in this section is to compute the unreliability of the system, *i.e.*, the probability of fault

$$U_L(n) = \mathrm{Prob}\{X_L(n) = 1\} = E(X_L(n)),$$

in terms of the failure probability $q_i = \mathrm{Prob}\{X_i = 1\}$ of each component i in $\{1, \ldots, n\}$. Note that the case of identical components corresponds to $q_i = q$ for all i in $\{1, \ldots, n\}$.

To achieve this goal, we are going to construct a linear recurrence equation of order k satisfied by $U_L(n)$. This linear recurrence, given in Lemma 2 below, can be obtained directly using probability arguments (see [8, Equation 1a.]); we propose here another approach using Shannon decomposition (see [16] for example). For the sake of completeness, and accessibility, some details on this method are given below.

First we replace the Boolean operators $\vee$, and $\wedge$ by additions, and multiplications defined as

$$X_i \vee X_j = X_j + \overline{X_j}\, X_i, \quad X_i \wedge X_j = X_i\, X_j,$$

where the Boolean negation is replaced by applying the identity $\overline{X} = 1 - X$. Note that, in the sequel, to apply the rules

$$E(T_1\, T_2) = E(T_1)\, E(T_2), \quad E(T_1 + T_2) = E(T_1) + E(T_2),$$

we have to ensure that the terms concerned, *i.e.*, T_1, and T_2 in the formula above, are stochastically independent.
To achieve the orthogonalization, we apply recursively Shannon decomposition (see [16] for example). We first apply it with X_n to derive the Boolean fault-tree function φ:

$$\begin{aligned} \varphi(X_1, \ldots, X_n) \quad = \quad & X_n\, \varphi(X_1, \ldots, X_{n-1}, 1) \\ & + \overline{X_n}\, \varphi(X_1, \ldots, X_{n-1}, 0). \end{aligned} \quad (5)$$

From (4), we get

$$X_L(n) = X_n\, (X_L(n-2) \vee X_{n-k+1} \cdots X_{n-1}) + \overline{X_n}\, X_L(n-1),$$

because the term $X_{n-k} \wedge \cdots \wedge X_{n-1}$ in (4) is absorbed by the last term for $X_n = 1$. Noting $T_1 = X_L(n-2) \vee (X_{n-k+1} \cdots X_{n-1})$, a further Shannon decomposition step shortens T_1 to T_2:

$$T_1 = X_{n-1}\, (X_L(n-3) \vee X_{n-k+1} \cdots X_{n-2}) + \overline{X_{n-1}}\, X_L(n-2),$$

and $T_2 = X_L(n-3) \vee (X_{n-k+1} \cdots X_{n-2})$. Continuing similarly, we finally get

$$T_{k-1} = X_L(n-k) \vee X_{n-k+1} = X_{n-k+1} + \overline{X_{n-k+1}}\, X_L(n-k).$$

Hence, $X_L(n)$ depends linearly on $X_L(n-1), \ldots, X_L(n-k)$. Computing the expected value for stochastically independent components, we get [8, Equation 1a.] the following lemma.

Lemma 2: The unreliability $U_L(n)$ of a linear consecutive-k-out-of-n:F system satisfies the linear recurrence of order k with constant coefficients given by

$$\begin{aligned} U_L(n) - p_n\, U_L(n-1) - p_{n-1}\, q_n\, U_L(n-2) - \cdots \\ \cdots - p_{n-k+1} \textstyle\prod_{i=n-k+2}^{n} q_i\, U_L(n-k) = \prod_{i=n-k+1}^{n} q_i, \end{aligned} \quad (6)$$

where, for i in $\{1, \ldots, n\}$, q_i is the (constant) failure probability of the component i, and $p_i = 1 - q_i$.

To apply Proposition 1 to the recurrence (6), its coefficients must be constants, *i.e.*, must not depend on n. This is ensured when each q_i is a given constant. Now, as a direct consequence of Lemma 2 and Proposition 1, we obtain the following theorem which provides the main result of the paper:

Theorem 1: The unreliability $U_L(n)$ of a linear consecutive-k-out-of-n:F system with (stochastically) independent components can be computed using $\mathcal{O}(\mathrm{M}(k) \log(n) + k^\omega)$ operations, where $\mathrm{M}(k)$ is the number of operations needed to multiply two polynomials of degree at most k, and ω is the exponent of linear algebra defined such that the number of operations needed to multiply two square matrices of size k is $\mathcal{O}(k^\omega)$. In particular, using the Fast Fourier Transform, we get an algorithm in $\mathcal{O}(k \log(k) \log(\log(k)) \log(n) + k^\omega)$ operations.

For the last assertion of the theorem, we use the fact that the Fast Fourier Transform corresponds to $\mathrm{M}(k) = k \log(k) \log(\log(k))$ [19, 8.2-8.3].

B. *Case of a circular arrangement of the components*

If one now considers a circular consecutive-k-out-of-n:F system, then the Boolean function of the fault tree becomes

$$X_C(n) = X_L(n) \vee (X_{n-k+2} \wedge \cdots \wedge X_1) \vee \cdots \vee (X_n \wedge \cdots \wedge X_{k-1}).$$

It is well known (see [4], [9], [11], [12], or [1, Theorem 2] for the case of identical components) that the unreliability $U_C(n)$ of the circular consecutive-k-out-of-n:F system can be expressed as a linear combination of the $U_L(m)$ for $n - k \leq m \leq n - 1$. Once again, this relation can be recovered using Shannon decomposition techniques [16], and following exactly the same strategy as in the linear case; details are left to the reader. As a consequence, we get the following corollary.

Corollary 1: The unreliability $U_C(n)$ of a circular consecutive-k-out-of-n:F system with (stochastically) independent components can be computed using $\mathcal{O}(\mathrm{M}(k) \log(n) + k^\omega)$ operations, where $\mathrm{M}(k)$ is the number of operations needed to multiply two polynomials of degree at most k, and ω is the exponent of linear algebra defined such that the number of operations needed to multiply two matrices of dimension k is $\mathcal{O}(k^\omega)$. In particular, using the Fast Fourier Transform, we get an algorithm in

$\mathcal{O}(k\,\log(k)\,\log(\log(k))\,\log(n) + k^{\omega})$ operations.

Proof: From the explanations above, to get $U_C(n)$, one needs $U_L(n-k),\ldots,U_L(n-1)$, *i.e.*, a slice of k consecutive elements of a solution of (6), which is of order k. The first assertion follows then from Theorem 1, and the last sentence of Section II.B. The second assertion of the theorem is clear as the Fast Fourier Transform corresponds to $\mathrm{M}(k) = k\,\log(k)\,\log(\log(k))$ [19, 8.2-8.3]. ■

IV. Conclusion

We have applied computer algebra techniques to handle the problem of computing the unreliability of consecutive-k-out-of-n:F systems. More precisely, the use of fast algorithms for computing a selected term of a linear recurrence with constant coefficients provides an algorithm having a complexity lower than that of all previous known algorithms for this problem, when $k \ll n$; which is the case in most applications where k typically represents a constant, and n represents a variable. Note that the same improvement is obtained for computing the (un)reliability of consecutive-k-out-of-n:G systems [22, Lemma 1].

We point out that the application of fast algorithms for manipulating linear recurrences, and computing powers of matrices is not restricted to the computation of the unreliability of consecutive-k-out-of-n:F systems. In the case of ladder-type redundancy structures (the so called *reliability block diagrams*), we find coupled linear recurrences. It is classical that these recurrences can be written in matrix form $U_n = \mathcal{R}\,U_{n-1} + U_0$, where U_n is the unknown vector, and U_0 contains the initial conditions. However, comparing to the one defined in Section II, the square matrix $\mathcal{R}$, say of size k, is no longer a companion matrix, so that, classically, we would use binary powering to compute its nth power in $\mathcal{O}(k^{\omega}\,\log(n))$ operations (see [17], or [18, 2.3.4] for example). This procedure would yield an algorithm in $\mathcal{O}(k^{\omega}\,\log(n))$ operations for computing U_n. Now, when $k \ll n$, which is the case in most applications, we can provide a little improvement to this algorithm. Indeed, there exists a Las Vegas type probabilistic algorithm for computing the nth power of a square matrix of size k in only $\mathcal{O}((k+\log(n))\,\mathrm{M}(k) + k^{\omega}\,\log(k))$ operations (see [23, Corollary 6.4], or [18, 2.3.4]), so that by using it, we manage to provide an algorithm in $\mathcal{O}((k+\log(n))\,\mathrm{M}(k) + k^{\omega}\,\log(k))$ for computing U_n.

In general, using fast algorithms in computer algebra for handling problems that can be written in terms of linear recurrence (or differential) equations may improve many computational methods.

Acknowledgment

The authors would like to thank the anonymous referees for their interesting comments which have improved the quality of the paper.

References

[1] C. Derman, G. J. Lieberman, and S. M. Ross, "On the consecutive-k-out-of-n:F system," *IEEE Trans. Rel.*, vol. R-31, no. 1, pp. 57–63, Apr. 1982.

[2] R. C. Bollinger and A. A. Salvia, "Consecutive-k-out-of-n:F networks," *IEEE Trans. Rel.*, vol. R-31, no. 1, pp. 53–56, Apr. 1982.

[3] M. Lambiris and S. Papastavridis, "Exact reliability formulas for linear and circular consecutive-k-out-of-n:F systems," *IEEE Trans. Rel.*, vol. 34, no. 2, pp. 124–126, Jun. 1985.

[4] M. T. Chao, J. C. Fu, and M. V. Koutras, "Survey of reliability studies of consecutive-k-out-of-n:F and related systems," *IEEE Trans. Rel.*, vol. 44, no. 1, pp. 120–127, Mar. 1995.

[5] E. Peköz and S. M. Ross, "A simple derivation of exact reliability formulas for linear an circular consecutive-k-out-of-n:F systems," *J. Appl. Prob.*, vol. 32, pp. 554–557, 1995.

[6] G. J. Chang, L. Cui, and F. K. Hwang, *Reliabilities of consecutive-k systems*. Dordrecht: Kluwer Academic Publishers, 2000.

[7] W. Kuo and M. Zuo, *Optimal reliability modeling: principles and applications*. New York: John Wiley, 2003.

[8] F. K. Hwang, "Fast solutions for consecutive-k-out-of-n:F system," *IEEE Trans. Rel.*, vol. R-31, pp. 447–448, Dec. 1982.

[9] I. Antonopoulou and S. Papastavridis, "Fast recursive algorithms to evaluate the reliability of a circular consecutive-k-out-of-n:F system," *IEEE Trans. Rel.*, vol. R-36, pp. 83–84, Apr. 1987.

[10] D. Z. Du and F. K. Hwang, "A direct algorithm for computing reliability of a consecutive-k cycle," *IEEE Trans. Rel.*, vol. R-37, pp. 70–72, Apr. 1988.

[11] F. K. Hwang, "An $\mathcal{O}(k\,n)$-time algorithm for computing the reliability of a circular consecutive-k-out-of-n:F system," *IEEE Trans. Rel.*, no. 1, pp. 161–162, Mar. 1993.

[12] J.-S. Wu and R.-J. Chen, "Efficient algorithm for reliability of a circular consecutive-k-out-of-n:F system," *IEEE Trans. Rel.*, vol. 42, no. 1, pp. 163–164, Mar. 1993.

[13] ——, "An $\mathcal{O}(k\,n)$ recursive algorithm for circular consecutive-k-out-of-n:F systems," *IEEE Trans. Rel.*, vol. 41, pp. 303–305, 1992.

[14] F. K. Hwang and P. E. Wright, "An $\mathcal{O}(k^3\,\log(n/k))$ algorithm for the consecutive-k-out-of-n:F system," *IEEE Trans. Rel.*, vol. 44, no. 1, pp. 128–131, Mar. 1995.

[15] M. Muselli, "New improved bounds for reliability of consecutive-k-out-of-n:F systems," *J. Appl. Prob.*, vol. 37, pp. 1164–1170, 2000.

[16] W. Schneeweiss, *The fault-tree method*. Hagen: LiLoLe-Verlag, 1999.

[17] C. M. Fiduccia, "An efficient formula for linear recurrences," *SIAM J. Comput.*, no. 1, pp. 106–112, Feb. 1985.

[18] A. Bostan, "Algorithmique efficace pour les opérations de base en calcul formel," Ph.D. dissertation, École Polytechnique, 2003.

[19] J. von zur Gathen and J.Gerhard, *Modern computer algebra*. New York: Cambridge University Press, 1999.

[20] V. Strassen, "Gaussian elimination is not optimal," *Numerische Mathematik*, vol. 13, pp. 354–356, 1969.

[21] J. Bunch and J. Hopcroft, "Triangular factorization and inversion by fast matrix multiplication," *Mathematics of Computation*, vol. 28, pp. 231–236, 1974.

[22] W. Kuo, W. Zhang, and M. Zuo, "A consecutive-k-out-of-n:G system: The mirror image of a consecutive-k-out-of-n:F system," *IEEE Trans. Rel.*, vol. 39, no. 2, pp. 244–253, 1990.

[23] M. Giesbrecht, "Nearly optimal algorithms for canonical matrix form," *SIAM J. Comput.*, no. 5, pp. 948–969, 1995.

Thomas Cluzeau received a MS in 2000, and a PhD in 2003 both in Mathematics, and Applications from the University of Limoges (France). He is a member of the CANSO project of the DMI of XLIM (UMR CNRS 6172) from the university of Limoges. His research interests are in computer algebra, and applications.

Jörg Keller is an associate professor of computer science and engineering at FernUniversitaet, the German distance teaching university, since 1996. He received M.Sc. and Ph.D. degrees in computer science in 1989, and 1992 from Saarland University, Saarbrücken, Germany, and has been an assistant professor there. His research interests are parallel processing, IT-security, and fault tolerance.

Winfrid G. Schneeweiss is a professor emeritus of computer engineering at FernUniversitaet, the German university for distance studies in Hagen. He is now the manager of LiLoLe-Verlag (Publ. Co.), serving life-long learning especially in the field of reliability/safety modelling. See www.lilole-verlag.de for details of his books, and those of several colleagues. He earned a Diplom-Physicist degree from Frankfurt University in 1957, a Ph.D. degree from Munich Technical University in 1968, and his Habilitation for computer science from Karlsruhe University in 1973. He has published widely with emphasis on fault tree analysis, and Petri net design. He is a Senior Member of IEEE, and a member of several German engineering/scientific societies.