Symbolic Computation
(An Editorial)

The short description given below of the scope, goal, and structure of the *Journal of Symbolic Computation* is the result of intensive discussions among the persons involved in initiating this journal. The editors will use it as a guide for the development of the journal and potential authors can use it to decide whether to make this journal a forum for their research results.

Scope of the Journal

The scope of the *Journal of Symbolic Computation* is

—the algorithmic solution of problems dealing with symbolic objects.

In a very broad sense, all mathematical objects (and their representations in computers) could be considered as symbolic objects. However, in accordance with two major research traditions that have developed in the last two decades, namely

—computational algebra and computational logic,

only certain classes of symbolic objects and specific algorithmic problems will be considered in this journal. Still, the scope is broad enough to integrate computational geometry, as a companion to computational algebra, and automatic programming (computer-assisted program verification, transformation and synthesis), as a field that is intimately connected with computational logic.

In these areas, all three aspects of the algorithmic solution of problems will be within the scope of the journal, namely

—mathematical foundations, correctness and complexity of new (sequential and parallel) algorithms
—implementation and integration of these algorithms in software systems
—applications of these systems for advanced problem solving in the natural and technical sciences.

Various names have been used in the past for the different sub-areas covered in the journal. Thus, for example, computational algebra is also called computer algebra, formula manipulation, symbolic and algebraic computation, computation in finite terms, computer analysis, analytic computation, or symbolic mathematics. Similarly, different names are in use for computational logic, computational geometry and automatic programming. We do not want to give priority to the usage of particular names. Rather, we chose "symbolic computation" as a short name for embracing the above areas without wanting to restrict the usage of other forms. Moreover, a standardised name or an exact definition of the boundaries of the field or the subfields could prevent further evolution of the field.

0747-7171/85/010001+06 $03.00/0

More specifically, the following topics are certainly within the scope of this journal:

—symbolic integration, symbolic summation, symbolic solution of differential equations and of other problems in analysis
—term simplification
—arithmetic in basic and higher algebraic domains
—polynomial factorisation
—symbolic solution of equations and systems of equations
—combinatorial group theory
—permutation and matrix groups
—computation of characters and representations
—computations in Lie groups
—computational number theory
—computational problems in non-associative and other algebras
—algorithmic combinatorics
—computational geometry
—computational aspects of algebraic geometry
—algorithmic analysis of real manifolds
—algorithmic problems in differential geometry
—algebraic algorithms in coding theory and cryptography
—interface between symbolic and numerical algorithms

—universal automated theorem proving
—unification
—automated theorem proving in special theories (for example, in the theory of real closed fields)
—automated proof checking
—algorithmic proof theory
—algorithmic problems in combinatorial logic and lambda calculus
—algorithmic logic
—automatic program synthesis
—automatic program transformation
—automatic program verification
—symbolic execution of programs
—algorithmic treatment of abstract data type specifications
—interpreters for high-level programs (functional programs, rewrite rule programs, logic programs)

—intrinsic complexity of problems in symbolic computation
—complexity analysis of algorithms in symbolic computation
—heuristic tuning of algorithms in symbolic computation
—design issues of software systems for symbolic computation
—parallel and other special hardware for symbolic computation
—programming languages for symbolic computation
—descriptions of available symbolic software systems
—descriptions of typical symbolic systems applications
—impact of symbolic computation on mathematical education

For a pragmatic further specification of the scope of the journal it may also help to mention some areas that, though algorithmic, will definitely be excluded from the journal.

Such areas are, for example, numerical analysis (if not combined with symbolic methods), algorithm theory (in the sense of general recursive function theory), compiler construction (if not in connection with symbolic software systems), algorithmic graph theory (if not relevant for one of the areas included). These areas will be excluded either because they consider objects (for example, floating point numbers) that are not symbolic objects in the sense of the research tradition of computational algebra and computational logic or because we feel that these areas are already covered sufficiently by existing journals.

The above enumerative characterisation of the scope was simply by inclusion and exclusion. However, it may also make evident some important common features of the areas covered by the *Journal of Symbolic Computation*. First, the algorithmic emphasis of the journal is on practical algorithms, i.e. algorithms that perform in reasonable computing time or at least aim at showing a direction for the solution of a symbolic problem on real computers. Second, the objects treated by the algorithms should either be algebraic in nature or expressions in formal languages. Typically, algebraic objects admit exact computation as opposed to the approximative character of numerical computations. Examples of algebraic objects are basic objects as the integers of arbitrary precision and the rationals, polynomials with coefficients in (basic) algebraic domains, matrices of (basic) algebraic objects, (representatives of) residue classes modulo congruence relations, elements of (finite) groups etc. Typically, computations on expressions in formal languages (terms, formulae, programs) have the flavour of operating on a metalevel of numerical computations. For example, numerically, a fixed finite sum can be evaluated for variable input values; a symbolic algorithm can take a whole spectrum of finite sum expressions as inputs and can transform them into a simpler form, whose simplicity then allows more efficient numerical evaluation.

Aspects of Symbolic Computation

The researchers working in the development of new algorithms for symbolic computation, the implementers of symbolic computation software systems, and the users applying these systems to problem solving in science and engineering naturally tend to lose contact with each other. This unfortunate situation has often been deplored in the past.

In fact, the feedback obtainable from system users would be a crucial stimulus for algorithm and system designers in the further improvement of the systems. Similarly, the interaction between algorithm designers and system designers is of vital importance. Finally, for improving the quality and scope of applications, clear and concise information about existing systems and the power of available algorithms is mandatory for potential users of symbolic software systems in science and engineering.

This multiple flow of information between algorithm designers, system designers and users, however, has proved to be difficult to achieve in the past. First, it is often nearly impossible for algorithm and systems designers to understand the details of an application paper that presupposes a huge amount of special knowledge about an application area, such as general relativity, high energy physics or industrial mechanics. Furthermore, these papers normally include only occasional remarks on the successful use of symbolic software systems. Similarly, it is often difficult for the system designers and potential users to extract the algorithmic details from basic research papers on symbolic algorithms.

Seemingly the above necessities and constraints contradict each other. Finding a balanced solution that hopefully satisfies the needs of the algorithm designers, the system implementers and the users as well was not an easy task for the editorial board of the *Journal of Symbolic Computation.* After careful consideration, we finally came up with a structure for the issues of the journal that should serve the needs of and make the journal attractive to all three groups of researchers involved in symbolic computation.

Goal of the Journal

It is the explicit goal of the *Journal of Symbolic Computation* to promote the growth and interaction of the research areas enumerated above and of contiguous areas by establishing one common publication forum. This endeavour seems to be overdue for several reasons.

First, in spite of a large increase in research activity over the last two decades none of these research areas has yet found a specific publication forum. Rather, the pertinent research results have been scattered up to now in many different journals or published in conference proceedings only. This is particularly true of computer algebra.

Furthermore, from a mathematical point of view, it is becoming increasingly clear that on the basis of common mathematical insights these areas share many advanced algorithmic ideas as, for example, lifting and completion. Also, many particular algorithms, e.g. unification algorithms or polynomial factorisation algorithms, occur as important subalgorithms in the solution of problems in several of these areas. For some of these research areas, e.g. the area of decision procedures for real closed fields or certain geometrical decision procedures, it would actually be hard to determine whether they belong to computational algebra, computational geometry, or computational logic. Some of the areas are prerequisites to each other. For example, algorithmic proof theory provides a basis for advanced approaches to program synthesis. Thus, there is strong evidence that mathematically the areas enumerated in the description of the scope of this journal are developing into a coherent field.

Finally, from the practical point of view, it seems that users are starting to require more than just independent software systems for numerical analysis, computer algebra, automated theorem proving, computational geometry, and automatic programming. Rather, the construction of software systems for "scientific computation", i.e. software systems integrating numeric, algebraic, geometric and logic computation, embedded in an automatic programming and knowledge engineering environment is a major challenge for the next decade. A common research publication integrating the component areas is of vital importance to move towards the successful development of these "mathematical expert systems".

Besides promoting the growth and interaction of the various subareas of symbolic computation, it is also a goal of the *Journal of Symbolic Computation* to provide a home for the three groups of researchers in symbolic computation, namely the algorithm designers, the system implementers, and the users, and also for their interaction with one another.

The editorial policy, reflected in the structure of the issues of the *Journal of Symbolic Computation,* aims to guarantee that these goals will be achieved.

Structure of the Issues

Typically, each issue of the journal will have the following structure:

Tutorial Section
Research Contributions Section
Systems Descriptions Section
Applications Letters Section
Bibliography Section.

Tutorial Section. Each tutorial section will contain one or two tutorials on subfields of symbolic computation. These contributions will be invited. Normally, they will give an overview of the mathematical foundations and basic ideas of the algorithmic methods used in that particular area and/or a survey on available systems and/or successful applications. This section should bridge the gap between algorithm designers on the one hand and system designers and users on the other but also stimulate the interaction between researchers working in different subfields of symbolic computation.

Research Contributions Section. Original papers are presented in this section, which, of course, forms the core of the journal. The research aspect can either be on symbolic algorithms (mathematical foundation, correctness, complexity), or on new software techniques for the implementation of symbolic software systems. This section contains up-to-date information for specialists actively involved in research on symbolic computation. However, authors of such papers will be encouraged to provide introductions that clarify the relevance of their paper for contiguous areas of symbolic computation in order to make the papers interesting for as wide a readership as possible.

Systems Descriptions Section. This section will contain one or two (invited) survey papers describing running and well established symbolic (hardware and) software systems that have found acceptance with users. Typically, these papers will provide descriptions of symbolic systems as they appear to the user and should motivate the application of these systems in problem solving. Of course, such descriptions will also be interesting for other system designers. However, these surveys will not normally concentrate on details of the specific software techniques used. (Papers on new software techniques belong in the research contributions section.)

Applications Letters Section. In this section short letters (1 to 4 pages each) will be included describing successful applications of symbolic software systems in various application fields. Since we feel that this section will be of particular importance for bridging the gap between users of symbolic software systems and algorithm and system designers, the editorial policy for the applications letters section is described in more detail here.

Typically, application letters will have the following structure:

—short statement of the problem solved
—short characterisation of the mathematical method, in particular symbolic method, used
—symbolic software system used
—results achieved
—suggestions for other possible applications, generalisations, and improvements
—details of how and where to obtain program and output listings and reference to a detailed publication.

The extent of the details of an application letter should allow another specialist in the application area to test similar problems using the same symbolic program or system. It should also allow the developers of the symbolic software system to understand the significance of the application model, the scope of its impact, and the possible improvements for the system. Finally, it should challenge the algorithm designer to determine which algorithmic problems need (better) solutions. Application letters will be refereed like all the other papers in the journal. They will typically be short versions (extended abstracts) of detailed publications in other refereed journals or proceedings, for example in a journal of the particular application field. However, many details pertinent to the particular application science can normally be left out and more information on the symbolic computation aspect should be added. Hence, a clear reference to the detailed publication must be made in the application letter and a copy of the detailed publication should be enclosed when submitting the application letter. In case the material covered in the application letter has not yet been published elsewhere, a program listing and tape for verification and a detailed technical report or equivalent material must be enclosed for verification when submitting the application letter.

Bibliography Section. In some issues we hope to include at least one short (annotated) bibliography on a particular subfield of symbolic computation, for which it might be difficult to obtain bibliographic data. In some issues we will have complete bibliographies on major and well-established subfields of symbolic computation. From time to time, updates to previous bibliographies will be published.

It is our strong desire that by a vigorous cooperation with the authors the goals set forth in this editorial will be achieved.

The Editors