Interval Computations
No 2, 1991

Advanced Seminar on Programming Languages for Scientific/Engineering Computation,

Leningrad, June 3-6, 1991

– A Summarizing Report –

Alexander Davidenkoff

During the SCAN-90 Conference at Albena/Bulgaria in September 23-28, 1990, a new scientific contact was made between the Institute of Applied Mathematics, University of Karlsruhe, (IAM) and the Special Department of Retraining Personnel, University of Leningrad, (SFPK). The common interests of both institutes led to the organization of the seminar. The intention was that Soviet scientists, researchers, and engineers are taught by scientists of IAM how to use and apply programming languages for scientific/engineering computation. In this area, the IAM has developed the program products ACRITH-XSC (FORTRAN-SC) and PASCAL-XSC which are extensions of FORTRAN and PASCAL with emphasis on getting more reliablitiy in computed numerical results. The seminar with about twenty participants can be seen as the beginning of a new cooperation in this area between German and Soviet scientists. The organization was done by Dr. S. S. Voitenko of SFPK, and the lectures have been given by Dr. D. Cordes, Dipl.-Ing. A. Davidenkoff, and Prof. Dr. U. Kulisch of IAM.

Семинар по алгоритмическим языкам для научных/инженерных вычислений.

Ленинград, 3–6 июня, 1991

– Краткий отчет –

Александр Давиденкофф

На конференции SCAN–90, проходившей 3–6 сентября 1990г. в Албене(Болгария) имели место контакты между Институтом прикладной математики (IAM) Университета Карлсруэ и специальным факультетом переподготовки кадров (СФПК) Ленинградского Университета. Наличие взаимной заинтересованности обоих институтов привело к организации семинара. Основной целью семинара было обучение советских ученых и инженеров специалистами из IAM использованию языков программирования для научных/инженерных расчетов. В этой области в IAM были разработаны оригинальные программные продукты ACRITH–XSC (FORTRAN–SC) и PASCAL–XSC, которые являются расширениями языков Фортран и Паскаль средствами, повышающими надежность при получении численных результатов. Семинар, в котором приняли участие около двадцати человек может рассматриваться как начало сотрудничества в рассматриваемой области между немецкими и советскими специалистами. Семинар был организован д-ром С.С. Войтенко (СФПК). Лекции читали сотрудники IAM д-р Д. Кордес, А. Давиденкофф, проф., д-р У. Кулиш.

1. Introduction

Due to the fast development and progresses in microelectronics, computers are permanently becoming smaller and more powerful. In the area of engineering and scientific computation, larger and larger numerical problems can be solved on a computer in a reasonable time. A natural request is that the results of such problems are reliable and correct in order to avoid much time in manual error analysis or expensive experiments. A fundamental requirement is a computer arithmetic with maximum accuracy. An important tool for numerical reliability is the process of automatic verification allowing to distinguish between errors caused by the mathematical model of a technical problem and those caused by numerical inaccuracies. Only if numerical errors can be excluded the model can be improved. The programming languages PASCAL-XSC and ACRITH-XSC (FORTRAN-SC) have been developed to satisfy the requirements of a highly accurate arithmetic and the need for user-friendly programming concepts. These languages are particularly suited for the development of numerical algorithms delivering highly accurate and automatically verified results. They can be used in many areas of scientific and engineering computation. The main advantages of PASCAL-XSC and ACRITH-XSC are the concepts for improved accuracy, like an interval arithmetic, the optimal dot product of two vectors, the exact evaluation of so-called dot product expressions, and highly accurate standard functions and the concepts for simplifying programming, like dynamic arrays, subarrays, user-defined and overloaded operators. In PASCAL-XSC and ACRITH-XSC, programs can easily be written and read so that the appearance of programming errors is reduced.

The programme of the seminar consists of the following topics:

- Contemporary Computer Arithmetic
- Fundamentals and Principles of Automatic Verification in Numerical Analysis
- Introduction in PASCAL-XSC
- Introduction in ACRITH-XSC
- Introduction in FORTRAN 90
- Self-Verifying Numerical Algorithms in PASCAL-XSC and ACRITH-XSC
- PASCAL-XSC Computer Practicum on PC's

2. Computer Arithmetic and Principles of Verification

Nowadays, a very important requirement of a contemporary floating-point computer arithmetic is maximum accuracy including the directed roundings *to nearest*, *downwards*, *upwards*, and *to zero*. Maximum accuracy means that there is no other floating-point number between the rounded result and the exact result. An arithmetic of this quality is the basis for developing numerical algorithms delivering correct and even verified results. Considering the new generation of vector and parallel computers, which are able to solve gigantic numerical problems based on floating-point arithmetic, different methods of vectorization and parallelization may influence the sequence of operations and, therefore, the effects of rounding. Thus, differing results can be calculated depending on the computing method. In these cases, much time and effort must be spent for an error analysis in order to find out the correctness of the results of a large numerical problem. In order to avoid this, a highly accurate arithmetic is needed. The ANSI/IEEE Floating-Point Standard 754 for binary numbers [IEEE85] provides the four basic arithmetic operations $+, -, *, /$ with maximum accuracy including the directed roundings *to Zero*, *to Nearest*, *to Minus Infinity*, and *to Plus Infinity*. Many floating-point processors available on the market are equipped with an arithmetic according to this standard, especially in the class of microcomputers.

In the Kulisch-Arithmetic [Kul81,Kul86,Boh90], a further basic operation is added, namely the optimal scalar product, also called *dot product*, of two vectors evaluated with maximum accuracy. These basic dot product operations are fundamental for higher arithmetic operations like *multiply and add* or *multiply and accumulate* [IGR89] and for vector and matrix operations, all computed with maximum accuracy. The optimal dot product is also a basic operation for the verified and reliable solution of numerical problems (e.g., solving a system of linear equations) and a very important tool for reducing the round-off effects in case of cancellation of leading digits [Kul83,Kul89]. The directed roundings enable the implementation of a maximally accurate interval arithmetic which is the basis for the realization of self-verifying numerical algorithms. Furthermore, the Kulisch-Arithmetic is the theoretical basis for the programming languages ACRITH-XSC and PASCAL-XSC.

One of the main goals of the XSC-languages is to provide the tools for developing and implementing numerical algorithms delivering verified results. As mentioned above, the prerequisite is an arithmetic with maximum accuracy combined with an interval arithmetic. One of the principle techniques of result verification, the *interval defect correction method*, can be summarized as follows [Kul88,Kul83]:
For a given problem first an approximation is computed by a conventional method. Then, this approximate value is improved by an interval iteration, in which this value is included in an interval as narrow as possible. The optimal inclusion is attained if the lower bound and the upper bound of the resulting interval differ by at most 1 *ulp* (unit in last place). One of the most important operations in this method is the computation of the *defect*, also called *residuum* which is used for the improvement of the approximation (see also the following example). Since this defect is often very small cancellation effects must be avoided which is possible if the optimal dot product defined in the Kulisch Arithmetic is used.

The mathematical proof, that the sketched method really finds an enclosure of the exact result, is based on fixed-point theorems like that of Brouwer which has been illustrated in the seminar (cf.[Kau84]). Hence, if the given problem can be expressed as a fixed-point problem this theorem can be applied and the existence of a verified solution can be proved. Typical examples which have been presented are the verified solution of a system of linear equations and the Interval Newton method finding the root of a function.

For illustration of this method, the program structure for the verified solving of a system of linear equations is shown:

With a non-singular $n \times n$ matrix A, a vector of unknowns x, and a right-hand side vector b the algorithm for the verified solution of the problem $A \cdot x = b$ has the following structure :

(1) *Find an approximation inverse $R \approx A^{-1}$ applying a conventional fast method.*

(2) *Compute an inclusion vector $\tilde{z}$ of the approximate solution $\tilde{z} := \#\#(R \cdot b)$;*
The notation '##', used in PASCAL-XSC and ACRITH-XSC [PXSC,AXSC], *indicates that the expression enclosed in parantheses is evaluated exactly (right #) and that after completion each exact result element is rounded to its smallest enclosing interval (left #).*

$$x^{(0)} := \tilde{z};$$

(3) *Compute an inclusion E with maximum accuracy of the residuum matrix*

$$E := \#\#(I - R \cdot A)$$

where I is the Idendity matrix. This means that the residuum is evaluated without intermediate round-off errors. The maximally accurate evaluation of the interval matrix E avoids cancellation effects and is essential for the following iteration steps.

(4) $i := 0$;

repeat

Enlarge slighty the interval vector $x^{(i)}$;

Compute $x^{(i+1)} := \tilde{z} + E \cdot x^{(i)}$;

$i := i + 1$;

until $x^{(i+1)} < x^{(i)}$ *or* $i \geq i_{max}$.

3. PASCAL-XSC, ACRITH-XSC and FORTRAN 90

PASCAL–XSC [Kla91] is a universal PASCAL extension with extensive, standard modules for scientific computation. It is available for personal computers, workstations, mainframes, and supercomputers by means of an implementation in C.

By using the mathematical modules of PASCAL–XSC, numerical algorithms which deliver highly accurate and automatically verified results can be programmed in an easy manner. PASCAL–XSC simplifies the design of programs in engineering/scientific computation by modular program structure, user-defined operators, overloading of functions, procedures, and operators, functions and operators with arbitrary result type, dynamic arrays, arithmetic standard modules for additional numerical data types with operators of highest accuracy, standard functions of high accuracy and exact evaluation of expressions.

The most important advantage of the new language is that programs written in PASCAL–XSC can be understood without difficulties by reducing unnecessary overhead. This is due to the fact that all operations, even those in the higher mathematical spaces, have been realized as operators and can be used in conventional mathematical notation.

In addition to PASCAL–XSC, numerous numerical problem-solving routines with automatic result verification are available. The language supports the development of such routines.

In the seminar, the concepts of PASCAL-XSC have been illustrated by numerous basic numerical problems like the matrix inversion, the verified solution of systems of linear and nonlinear equations as well as the verified solution of an initial value problem for systems of linear and non-linear ordinary differential equations. The use of the PASCAL-XSC system and the examples have been demonstrated on a laptop computer connected with an LCD-display so that the participants have got a practical impression of this language and its use. Furthermore, results of solved problems have been sketched, such as the possibility to apply an enclosure method for the restricted Three Body problem by Multiple Shooting, the computation of solutions of the nonlinear Pendulum equation, and the verified proof of stability for a gear drive with four degrees of freedom.

ACRITH-XSC (High Accuracy Arithmetic - Extended Scientific Computation) [IBM90a] is a compiler comprising the complete FORTRAN 77 language and extensions with emphasis on engineering and scientific computation. The goals and the concepts of ACRITH-XSC are the same as those of PASCAL-XSC. In ACRITH-XSC, a highly accurate arithmetic has been integrated enabling the development of numerical algorithms, which deliver highly accurate and automatically verified results, by means of an interval arithmetic, the optimal dot product of two vectors and the exact evaluation of dot product expressions. ACRITH-XSC contains concepts making programming easier and less erroneous; these are functions and expressions with array results, dynamic arrays and subarrays, and an operator concept. ACRITH-XSC has been developed and implemented in the Institute for Applied Mathematics at the University of Karlsruhe in collaboration with the IBM Research and Development Laboratory Böblingen in Germany. Since August 1990, ACRITH-XSC is an IBM Program Product.

The historical development of ACRITH-XSC has been started in the years 1976 through 1980, in which the first version of the PASCAL extension PASCAL-SC has been defined and implemented for micro computers. The manifest demand to transfer the concepts, provided in PASCAL-SC for the improvement of accuracy, to the FORTRAN world on mainframes led to the implementation of the IBM Subroutine Library ACRITH [IBM86] in 1983 through 1985. This subroutine library provides the arithmetical tools, but is difficult to use. Finally, in 1984 through 1989 the language FORTRAN-SC [Ble87,Met89] has been specified and implemented combining both concepts of an accurate arithmetic and user-friendly programming. FORTRAN-SC has been in use at various universities and research institutions since 1987. In August 1990 the name FORTRAN-SC has been changed to ACRITH-XSC [IBM90a,IBM90b] and became an IBM Program Product which is now distributed worldwide.

ACRITH-XSC is running on IBM /370 and /390 mainframes under the operating system VM/CMS. The scope of applications of ACRITH-XSC is wide-spread. Nearly every application in which the reliability and correctness of numerical results play a fundamental role is suited for being programmed in ACRITH-XSC. Today, the numerical problems are permanently increasing, and, therefore, they are solved on supercomputers providing very high performance rates. Hence, the significance of reliability of numerical results is becoming more and more important. ACRITH-XSC can be used for solving numerical problems in electrical, mechanical, civil and chemical engineering, in fluid dynamics, in aviation and astronautics, in power net distribution and failure prevention, in design and optimization of control systems, even in geology for exploration of mineral deposits and in simulation. A very important application of ACRITH-XSC is in education and research at universities and in development departments of the industry.

Some concepts of the forthcoming new standard FORTRAN 90 [FTN90] which will be published in 1991 are contained in ACRITH-XSC in a very similar manner, such as functions with array result, operator definition and overloading, dynamic arrays and subarrays. However, the tools for result verification and improvement of accuracy are inherent to ACRITH-XSC. These features, which are explained in the following, are interval arithmetic, optimal conversion of constants and data, vector/matrix operations including the optimal dot product, and the evaluation of dot product expressions.

As for PASCAL-XSC, many illustrative examples and numerical applications have been presented at the seminar. Beside basic numerical problems, applications have been demonstrated like the automatic differentation and the computation of a damped electrical oscillator, for which the whole process from the technical-mathematical model to its realization in ACRITH-XSC has been discussed. Furthermore, the general problem to develop an inclusion method for a given problem and to implement it has been sketched for the example of the highly accurate evaluation of a polynomial. The determination of an enclosure of the unique solution of an ordinary differential equation as initial value problem has been sketched based on the Picard-Lindeloef iteration and using interval polynomials for the inclusion. Finally, the application of computing the critical rotation speeds of a shaft design was shown which demonstrated that the conventional computation issued unusable results for high rotation speeds and a realization by ACRITH-XSC produced reliable and expected results also for high speeds. Some of these examples are described in [Kul88,Kul89].

Unfortunately, the use of the ACRITH-XSC system could only be presented theoretically since an IBM mainframe computer was not available for a demonstration.

4. PASCAL-XSC Computer Practicum

The last section of the seminar was a computer practicum in PASCAL-XSC. The practicum took place in a pool with eight AT compatible personal computer near the place of the seminar. The PASCAL-XSC compiler has been installed under the MS-DOS 3.30 operating system. The Borland Turbo C++ 1.0 compiler was used to compile the generated C code. For the participants many various exercises, containing mainly simple basic numerical problems for getting highly accurate results, have been prepared so that they could select according to their interests. For an easy use of the PASCAL-XSC system a simple dialog manager menu has been provided enabling the participant an easy way to edit, compile, recompile in case of errors, and to run programs. By this practicum the participants have got the base knowledge of programming in PASCAL-XSC.

5. Conclusion

The purpose of the seminar was to teach scientists, researchers and engineers how to use PASCAL-XSC and ACRITH-XSC for solving numerical problems with high accuracy. The participants have learned the principles of automatic verification of simple numerical algorithms. Many illustrative examples and practical exercises in programming with PASCAL-XSC have allowed an easy entrance to thus very important field of scientific and engineering computation. The participants have got an insight into the importance of reliable computing and a guide to apply the taught methods for their own problems.

The seminar was the first activity in the intended cooperation between IAM and SFPK. SFPK is planning to organize educational courses for XSC-langugages in the Soviet Union; furthermore, the publication of literature about the XSC-languages in Russian language and the development of program modules for specific application fields are planned. Also, the exchange of specialists and lecturers is intended in order to intensify and coordinate the common work.

References

Ble87. Bleher, J. H., Rump, S. M., Kulisch, U., Metzger, M., Ullrich, Ch., Walter, W.: *FORTRAN-SC: A Study of a FORTRAN Extension for Engineering/Scientific Computation with Access to ACRITH.* Computing 39, 93–110, Springer, 1987. A copy in [Kul88].

Boh90. Bohlender, G.: *What Do We Need Beyond IEEE Arithmetic?*In: Ullrich (ed.): Computer Arithmetic and Self-Validating Numerical Methods. Academic Press, 1990.

FTN90. International Standard Organization ISO: *Programming Language Fortran.* ISO/IEC 1539:1991, 1991.

IBM86. IBM: *High Accuracy Arithmetic Subroutine Library (ACRITH).* Program Description and User's Guide. SC33-6164-02, 3rd edition, IBM Corporation, 1986.

IBM90a. IBM: *High Accuracy Arithmetic – Extended Scientific Computation.* ACRITH-XSC Language Reference, Version 1, Release 1, SC33-6462-00, IBM Corporation, 1990.

IBM90b. IBM: *High Accuracy Arithmetic – Extended Scientific Computation.* General Information, Version 1, Release 1, SC33-6461-01, IBM Corporation, 1990.

IEEE85. American National Standards Institute / Institute of Electrical and Electronics Engineers: *IEEE Standard for Binary Floating-Point Arithmetic.* ANSI/IEEE Std 754–1985, New York, 1985.

IGR89. IMACS, GAMM: *Resolution on Computer Arithmetic.* In: Mathematics and Computers in Simulation 31, S.297–298, 1989.

Kla91. Klatte, R., Kulisch, U., Neaga, M., Ratz, D., Ullrich, Ch.: *PASCAL–XSC, Sprachbeschreibung mit Beispielen.* Springer Verlag, 1991. English Version to appear 1991.

Kau84. Kaucher, E., Miranker, W.L.: *Self-Validating Numerics for Function Space Problems.* Academic Press, 1984.

Kul81. Kulisch, U., Miranker, W. L.: *Computer Arithmetic in Theory and Practice.* Academic Press, New York, 1981.

Kul82. Kulisch, U., Ullrich, Ch. (eds.): *Wissenschaftliches Rechnen und Programmiersprachen.* Berichte des German Chapter of the ACM 10, B. G. Teubner, Stuttgart, 1982.

Kul83. Kulisch, U., Miranker, W. L. (eds.): *A New Approach to Scientific Computation.* Notes and Reports in Computer Science and Applied Mathematics, Academic Press, Orlando, 1983.

Kul86. Kulisch, U., Miranker, W. L.: *The Arithmetic of the Digital Computer: A New Approach.* SIAM Review 28, no. 1, 1986.

Kul88. Kulisch, U., Stetter, H. J. (eds.): *Scientific Computation with Automatic Result Verification.* Springer-Verlag Wien New York, 1988.

Kul89. Kulisch, U. (ed.): *Wissenschaftliches Rechnen mit Ergebnisverifikation, Eine Einführung.* Vieweg-Verlag; Akademie-Verlag, Berlin, 1989.

Met89. Metzger, M., Walter, W.: *FORTRAN-SC: A Programming Language for Engineering Scientific Computation.* In [Ull90].

Ull90. Ullrich, Ch. (ed.): *Computer Arithmetic and Self-Validating Numerical Methods.* Proceedings of SCAN-89, Basel (Oct. 2-6, 1989), Academic Press, 1990.

Institut für Angewandte Mathematik
Universität Karlsruhe
Kaiserstr. 12
W-7500 Karlsruhe
Germany

Tel. : +49 (721) 377936
FAX : +49 (721) 695283
e-mail : AE12@DKAUNI2.BITNET or
: AE12@ibm3090.rz.uni-karlsruhe.de (Internet)