Interval Computations
No 4(6), 1992

PASCAL–XSC. A PORTABLE PROGRAMMING SYSTEM FOR SCIENTIFIC COMPUTATIONS

Dmitri Shiriaev

The information on subsequent development of the general purpose programming language PASCAL-XSC, which is oriented to using of interval analysis algorithms, is presented.

PASCAL–XSC. ПЕРЕНОСИМАЯ СИСТЕМА ПРОГРАММИРОВАНИЯ ДЛЯ НАУЧНЫХ ВЫЧИСЛЕНИЙ

Д.Ширяев

Приводится информация о дальнейшем развитии языка программирования общего назначения PASCAL-XSC, ориентированного на использование алгоритмов интервального анализа.

PASCAL–XSC is a general purpose programming language which provides special support for the implementation of sophisticated numerical algorithms with mathematically verified results [4]. PASCAL-XSC simplifies the design of programs in engineering and scientific computation by modular program structure, user-defined operators, overloading of functions, procedures, and operators, dynamic arrays. Arithmetic standard modules for additional numerical data types including operators and standard functions of high accuracy and the exact evaluation of expressions provide the main numerical tools.

The development of PASCAL–XSC programs is supported by the PASCAL–XSC *development system* [1]. The system consists of the PASCAL–XSC *compiler* and the PASCAL–XSC *runtime system* which are both written in ANSI C language. Instead of implementing a large variety of "native code generators" for different processor and operating systems, the PASCAL–XSC system compiles a given PASCAL–XSC source code into ANSI C code which is passed to a C compiler. Finally, the resulting object code and the routines of the PASCAL–XSC runtime system are linked together. The PASCAL–XSC development system has the advantage of being portable across many platforms and is available for personal computers, workstations, mainframes and supercomputers .

The mathematical definition of the arithmetic is an intrinsic part of the language, including optimal arithmetic operations with directed roundings which are directly accessible in the language. Further arithmetic operations for intervals and complex numbers and even vector/matrix operations provided by precompiled arithmetic modules are defined with maximum accuracy according to the rules of semimorphism (see [5]).

From the point of view of mathematics, it is of fundamental importance that results of implemented algorithms are reproducible in spite of different computing facilities. Unfortunately, the arithmetical capabilities of computer systems are quite different concerning the representation of floating-point numbers and the way arithmetic operations are processed. A complete software simulation of the arithmetic defined by the IEEE 754 binary floating-point arithmetic standard [2] is provided. This ensures that PASCAL–XSC programs produce identical results on all platforms. To achieve better performance, the runtime system can be configured in such a way that it adapts to the arithmetic hardware unit of the processor in use.

By using the mathematical modules of PASCAL–XSC, numerical algorithms which deliver highly accurate and automatically verified results can be programmed easily. Programs written in PASCAL–XSC are easily readable since all operations, even those in the higher mathematical spaces, have been realized as operators and can be used in conventional mathematical notation.

In addition, a large number of numerical problem-solving routines with automatic result verification have been developed. The PASCAL–XSC system greatly facilitates the development of such routines.

1. The language PASCAL–XSC

PASCAL–XSC is an eXtension of the programming language PASCAL for Scientific Computation, containing the following features:

- Standard PASCAL
- Universal operator concept (user-defined operators)
- Functions and Operators with arbitrary result type
- Overloading of procedures, functions and operators
- Overloading of assignment operator
- Overloading of the I/O -routines *read and write*
- Module concept
- Dynamic arrays and subarrays
- String concept
- Optimal (exact) scalar product and Controlled rounding
- Standard type *dotprecision* (a fixed-point format covering the whole range of floating-point products)
- Additional arithmetic standard types such as *complex*, *interval* etc.
- Highly accurate arithmetic for all standard types
- Highly accurate standard functions
- Exact evaluation of expressions (#-expressions)

See [4] for details.

2. Different real arithmetics

A special feature of the new compiler is that the real arithmetic is exchangeable to support different applications which may require different properties of the arithmetic (portability, speed, accuracy). See [3] for details.

Supported arithmetics are:

- Software emulation of the IEEE standard arithmetic. A complete floating-point arithmetic for the "double" format of the IEEE binary floating-point standard [2] is simulated in software. All accuracy requirements of the standard are fulfilled inclusive directed roundings, handling of infinity and overflow/underflow detection. No special properties of the hardware or support from the C runtime system is required.

- The hardware arithmetic of the computer in use. The arithmetic operations are supported by the C runtime system. The data format and the accuracy of operations need not necessarily satisfy the IEEE binary floating-point standard. This arithmetic is intended to be used by programs that shall be "fast".
- Multiple precision arithmetic is intended for the programs implementing high-precision numerical algorithms. The arithmetic operations are based on the special multiprecision data type. Variables of this type may hold values with a varying number of mantissa digits during the execution of the program
- A user-defined arithmetic.

3. Exception handling

To offer a comprehensive and convenient reaction for different erroneous situations, exception handling with trace-back is provided. Exception handling is independent from the hardware or the operating system in use. Since PASCAL–XSC offers the overloading of functions and operators, module concept and recursion, Trace-back is a necessary tool for the location of the position where an exceptional condition occurred. For the more detailed information see [3].

The handling of the following exceptions is supported:

- IEEE 754 exceptions
  - division by zero
  - inexact result
  - invalid operation
  - exponent overflow
  - exponent underflow
- Input/output error
- Range error
- Invalid argument to standard function
- Allocation error

4. Problem-solving routines

Routines for solving common numerical problems have been implemented in PASCAL–XSC . The applied methods compute a highly accurate enclosure of the true solution of the problem and, at the same time,

prove the existence and the uniqueness of the solution in the computed interval. The advantages of these new routines are :

- The solution is computed with maximum or high, but always controlled accuracy, even in many ill-conditioned cases.
- The correctness of the result is automatically verified, i.e. an enclosing set is computed, which guarantees existence and often also uniqueness of the true solution contained in this set.
- If no solution exists or if the problem is extremely ill-conditioned, an error message is issued.

Among others, PASCAL–XSC routines cover the following subjects:

- linear systems of equations
  - full systems (*real*, *complex*, *interval*, *cinterval*)
  - matrix inversion (*real*, *complex*, *interval*, *cinterval*)
  - least squares problems (*real*, *complex*, *interval*, *cinterval*)
  - computation of pseudo inverses (*real*, *complex*, *interval*, *cinterval*)
  - band matrices (*real*)
  - sparse matrices (*real*)
- polynomial evaluation
  - in one variable (*real*, *complex*, *interval*, *cinterval*)
  - in several variables (*real*)
- zeros of polynomials (*real*, *complex*, *interval*, *cinterval*)
- eigenvalues and eigenvectors
  - symmetric matrices (*real*)
  - arbitrary matrices (*real*, *complex*, *interval*, *cinterval*)
- initial and boundary value problems of ordinary differential equations
  - linear
  - nonlinear
- evaluation of arithmetic expressions
- nonlinear systems of equations
- numerical quadrature
- integral equations
- automatic differentiation
- optimization

5. The availability of the PASCAL–XSC system

Until now the PASCAL–XSC system has been successfully installed and thoroughly tested on many computers (see the table). On some systems hardware arithmetic is supported, making the generated programs faster, for other ones the hardware arithmetic support is in work, but is not yet available. The system is widely spread and used by many students for educational purposes and for software development.

Computer	Operating System	C compiler	Hardware support
IBM PC	DOS	Microsoft C 6.0	Available
IBM PC	DOS	Turbo C 2.0	Available
IBM PC	DOS	Turbo C++ 1.0	Available
IBM PC	DOS	Borland C++ 2.0	Available
IBM PC	DOS	Zortech C 3.0 v2	Available
Sun SPARC Station	SunOS 4.1	System	Available
Micro VAX Station	ULTRIX	System	
DEC Station 3100	ULTRIX	System	
IBM PS/2	AIX	System	
IBM RS 6000	AIX	System	
HP 9000/300 Series	UNIX	System	
HP 9000/800 Series	UNIX	System	
HP 9000/700 Series	UNIX	System	
NeXT	UNIX	System	
CONVEX	UNIX	System	
VAX	VMS	System	
Transputer T800	HELIOS	System	
Data General	ULTRIX	System	

The IBM PC version for the Zortech C compiler supports 32 bit code generation and has a built-in DOS extender which allows to overcome the 640K DOS memory limit. The version for the Sun SPARC Station provides direct access to the hardware-supported quadruple arithmetic. The version of PASCAL–XSC on a Transputer System under the operating system HELIOS combines the concepts of PASCAL–XSC for reliable numerical computing and the advantages of parallel computing.

References

1. Allendörfer, U. and Shiriaev, D. *PASCAL-XSC. A portable development system.* In: "Brezinsky, C. and Kulisch, U. (eds) Computational and applied mathematics I – algorithms and theory. Proceedings of the 13th IMACS World Congress, Dublin, Ireland", Elsevier Science Publishers B.V. To be published.

2. *IEEE standard for binary floating-point arithmetic.* American National Standards Institute / Institute of Electrical and Electronics Engineers, ANSI/IEEE Std. 754-1985, New York, 1985.

3. Cordes, D. *Runtime system for a PASCAL-XSC compiler.* In: "Kaucher, E., Markov, S. M. and Mayer, G. (eds.) Computer arithmetic, scientific computation and mathematical modelling. IMACS annals on computing and applied mathematics **12**", J.C. Baltzer, Basel, 1991, pp. 151–160.

4. Klatte, R., Kulisch, U., Neaga, M., Ratz, D. and Ullrich, Ch. *PASCAL-XSC language reference with examples.* Springer, Heidelberg, 1992.

5. Kulisch, U. and Miranker, W. L. *Computer arithmetic in theory and practice.* Academic Press, New York, 1981.