ACRITH : HIGH-ACCURACY ARITHMETIC

AN ADVANCED TOOL FOR NUMERICAL COMPUTATION

J.H. Bleher / A.E. Roeder / S.M. Rump
IBM Development and Research
7030 Boeblingen
Germany

ABSTRACT

The High-Accuracy Arithmetic Subroutine Library (ACRITH) is a program product for engineering / scientific application. It consists of a subroutine library for solving problems in numerical computation. All results obtained have algorithmically verified accuracy.

INTRODUCTION

With existing processors and conventional techniques, it is often difficult, or in some cases not even possible to solve problems in numerical computation to the required degree of accuracy. To illustrate this in more detail Figure 1 shows the procedure used in the engineering / scientific environment when solving real-world problems.

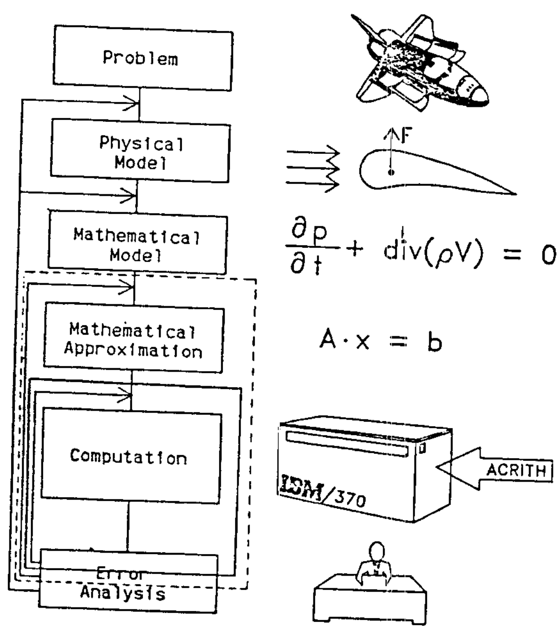

Figure 1: Solution of real-world problems, with ACRITH

The first task is the description of the problem followed by the description of the physical model. Knowledge and experience leads from the physical level to the description of the problem by mathematical equations. Unfortunately those equations normally are not adequate for the computational phase. Their complexity again requires hard brain work to get equations that can be handled by standard routines available on our computers. We call this part the mathematical approximation.

In our example the mathematical approximation results in a linear equation described by the equation $A \cdot x = b$ where A is a matrix, x the vector of the unknowns and the vector b is called the right hand side.

After computation a difficult and often time consuming task is the error analysis. Inaccuracies during computation may occur due to:

- loss of digits in small differences through cancellation
- rounding during computation
- conversion of input numbers to machine representation (e.g. from decimal to hexadecimal).

If the result is not correct different "loops" for a correction are possible. Along the inner loop the computation is executed again with a change in the precision of the floating-point computation for example from short to long or even to extended precision. If, after a repeated error analysis the result is not satisfactory, the professional has a different algorithm ready to deal with this "ill-conditioned problem". He is changing his program using the new algorithm and runs the problem again.

Now, let us assume that this time he concludes his error analysis with the strong feeling that he can trust his results. Otherwise the two outer loops would have asked him to think over his physical or mathematical model and start all the work again. The bold box in Figure 1 indicates where ACRITH can help the user to complete his task. ACRITH directly influences the computa-

CH2146-9/85/0000/0318$01.00 © 1985 IEEE

tional phase. For the given input parameters the solution is determined with verified accuracy, in this example the solution of the system of linear equations. Therewith the numerical part of the error analysis is no longer necessary.

In addition to that the knowledge of the capabilities of ACRITH influences the mathematical approximation. This is depicted by the dashed box. Some approximation methods which deliver good results have not been used in the past due to problems during numerical computation. Having in mind that ACRITH delivers results of verified accuracy some methods can be successfully used again.

ACRITH DESCRIPTION

The High-Accuracy Arithmetic Subroutine Library is an IBM-First. It consists of routines for solving problems in numerical computation. All results obtained have an algorithmical verification of the correctness and the accuracy.

Together with the ACRITH Subroutine Library a so-called Online Training Component (OTC) is provided. The OTC has been designed to give the user a valuable tool for familiarization with the capabilities of ACRITH. In addition to that it allows solving of numerical problems interactively.

ACRITH runs on all System /370 processors under VM/SP, MVS/370, MVS/XA (24 bit-addressing mode), VSE/SP, and on the PC XT/370. It is callable from VS Fortran and Assembler programs. ACRITH arithmetic, which bases on a sound theory by Kulisch and Miranker [1] makes use of the High-Accuracy Arithmetic Facility. This architecture RPQ provides 20 new instructions, with rounding. The microcode implementation on all 4361 processors results in a remarkable performance improvement of the arithmetic and the subroutines when running on one of those processors.
With the ACRITH package a software simulation written in Assembler for the new instructions is provided enabling the subroutines to run on all System /370 processors.

SUBROUTINE LIBRARY

The ACRITH subroutine library offers a large variety of routines for solving problems in numerical computation. They deal with arithmetic expressions, polynomial evaluation, zeros of polynomials, linear equations, matrix inversion, linear programming, eigenvalues and eigenvectors, and standard functions. The theoretical basis for the algorithms is given in [2].

ENVIRONMENT

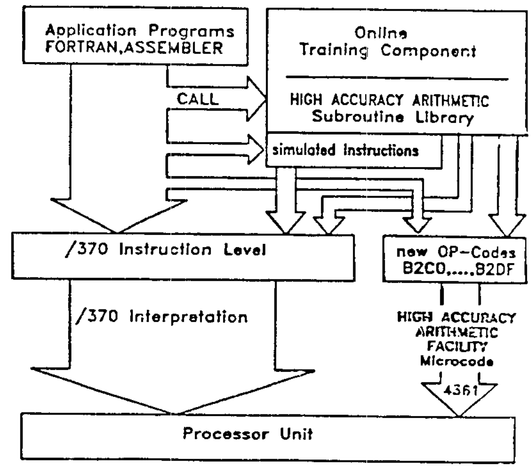

Figure 2: ACRITH application environment

The ACRITH application environment is depicted in Figure 2. In the upper left corner the application programs written in FORTRAN and/or Assembler are shown. After compilation /370 instructions are executed to obtain results.
To take advantage of the capabilities of ACRITH parts of the old programs or calls to other libraries have to be substituted by a "CALL" to the ACRITH subroutines. The subroutines execute on standard /370 level and use in addition the 20 new instructions, either directly on /370 machine level on all 4361 processors, or, via the simulated instructions on the standard /370 level.

Advanced users may directly use the 20 instructions to write their own programs. This possibility is also shown in Figure 2 by the small horizontal arrows.

At this point it should be mentioned that an application program using ACRITH runs on all /370 processors. If the microcoded instructions of the ACRITH Facility are available (4361 processor), the subroutines automatically use them, without the necessity of recompilation or relink, i.e. ACRITH programs always choose the fastest mode.

ACRITH FACILITY

The new and outstanding capabilities provided with the ACRITH package base on the architectural definition of 20 instructions.

These instructions can be divided in two classes, namely the basic arithmetic instructions and the accumulator instructions. The basic arithmetic instructions consist of add, subtract, multiply, divide, and load. Each of them can be executed with one of the four possible roundings upwards (towards + infinity), downwards (towards - infinity), to nearest floating-point number, and to zero.

All instructions are defined and architected for System /370 short and long floating-point format. The results obtained with these instructions are all of maximum accuracy, which means that, within the given floating-point format used, no floating-point number lies between the computed result and the result obtained with infinite precision. A key point of the Kulisch/Miranker theory is the introduction of an additional instruction: the scalar product with maximum accuracy. This is the step from the maximum accuracy single operation to maximum accuracy composed operations. The instructions defined in conjunction with the scalar product are called accumulator instructions:

- add/subtract operand to/from accumulator
- multiply and accumulate (scalar product)
- round from accumulator
- add/subtract accumulator to/from accumulator
- clear accumulator

The long accumulator occupies a 168 byte storage area. The layout of an accumulator in storage is depicted in Figure 3.

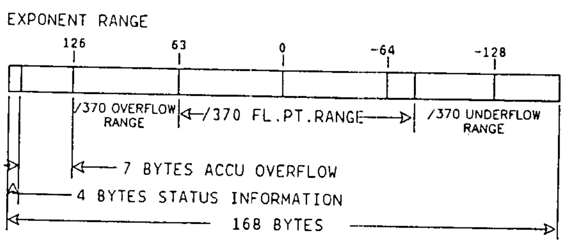

Figure 3: ACRITH accumulator layout

The accumulator consists of a four byte status area on the left, followed by a 164 byte numeric area. When a floating-point number is added to the accumulator, the fraction is positioned in the numeric area that corresponds to the exponent. The fraction is added at that point, and any carries are propagated to the left as far as necessary.

The exponent range covered in the accumulator is 16 ** 126 through 16 ** (-128), which is twice the standard /370 exponent range. Together with 14 digits for accumulator overflow no exponent overflow can occur, because the numeric area is large enough to allow any reasonable number of scalar products of the largest representable floating-point numbers to be accumulated.

EXAMPLE

In order to demonstrate the capabilities of ACRITH we have selected a set of linear equations. The solution of the linear system.

37639840 X - 46099201 Y = 0
29180479 X - 35738642 Y = -1

is obtained with ACRITH as

X = 46099201 and Y = 37639840.

This result is verified to be correct as well as the matrix is automatically verified to be non-singular.
With a conventional approach, using the 'Gaussian' elimination method one would have obtained three different results depending on the precision:

single: no result (divide exception)
double: X = 41095618.5 Y = 33554432
extended: X = 46099201 Y = 37639840.

In case of single precision the divide exception alerts the user that something is wrong with his problem. The results obtained with double and extended precision differ already in the second digit. It is up to the user, to verify by time consuming error analysis that the extended result is correct.

ACRITH, however delivers the result including the verification step, which means that the user can trust his result.

HISTORICAL BACKGROUND

In the historical background we want to give an overview on methods used in the past to obtain accurate or exact results. The methods have been derived after users realized the problems with roundoff errors and cancellation and the resulting errors in numerical computation.

The different approaches are:

- Symbolic computation
- Algebraic computation
- Naive interval arithmetic

The three methods were invented in the early sixties. The first two are strongly connected and they are often together referred to as "Symbolic and Algebraic Manipulation." The research and development in this area is in steady progress.
The third method, the naive interval arithmetic, was proposed as a global solution to numerical problems. It turned out that this is not the case. The naive interval arithmetic has been finished after few years whereas the sophisticated interval mathematics has been settled as an individual part of numerical analysis.

The standard floating-point algorithms do not deliver verified results (as has been demonstrated by examples) or provide error bounds for the results. The delivered results are often of high accuracy, but sometimes vastly wrong.

The new methods ACRITH is based on, deliver always results which are verified to be correct, i.e. no wrong results are possible. Moreover, the results of the lower level algorithms are always of maximum accuracy. The results of the higher level algorithms are almost always of maximum accuracy. The property of maximum accuracy means, that between the computed result and the infinite precise result there is no other floating-point number.

REFERENCE

[1] Kulisch, U., Miranker, W.L.: Computer Arithmetic in Theory and Practice. Academic Press, New York (1981).

[2] Rump, S.M.: Solving Algebraic Problems with High Accuracy, in "A New Approach to Scientific Computation", Eds. U.W. Kulisch and W.L. Miranker, Academic Press (1983).