

**ACRITH-XSC
ЯЗЫК ПРОГРАММИРОВАНИЯ
ДЛЯ НАУЧНО-ТЕХНИЧЕСКИХ ВЫЧИСЛЕНИЙ**

Александр Давиденкофф

ACRITH-XSC (High Accuracy Arithmetic – Extended Scientific Computation) представляет собой дальнейшее развитие языка ФОРТРАН 77 и специально предназначен для программирования приложений в научно-технической области. В ACRITH-XSC встроена арифметика высокой точности. Она вместе с интервальной арифметикой, точным скалярным произведением двух векторов и точным вычислением выражений скалярных произведений делает возможной разработку численных алгоритмов, которые дают надежные и автоматически проверенные результаты. ACRITH-XSC содержит удобные для пользователей средства, которые делают программирование проще и надежнее. К этим средствам можно отнести функции и выражения с результатами типа массив, динамические массивы и подмассивы, а также конструкции для определения и переопределения операций. ACRITH-XSC был совместно разработан и реализован Институтом прикладной математики университета г. Карлсруэ и лабораторией новых разработок фирмы IBM в г. Беблинген, ФРГ. С августа 1990 г. ACRITH-XSC – программный продукт фирмы IBM.

**ACRITH-XSC
A PROGRAMMING LANGUAGE FOR
SCIENTIFIC/ENGINEERING COMPUTATION**

Alexander Davidenkoff

ACRITH-XSC (High Accuracy Arithmetic – Extended Scientific Com-

putation) is a completely implemented FORTRAN 77 extension with emphasis on engineering and scientific computation. 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 by 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.

Введение

Проектирование ACRITH-XSC продолжалось с 1976 по 1980 г., в этот же период было определено и реализовано первое расширение языка PASCAL-SC [3,4,8]. Требование перенести имеющиеся в PASCAL-SC концепции для повышения точности также в область программирования на ФОРТРАНе привело сначала к реализации библиотеки подпрограмм ACRITH [5] с 1983 по 1985 г. Арифметика в этой библиотеке удовлетворяла требованиям высокой точности, однако работа с ней являлась слишком сложной. Окончательно это привело к разработке и реализации языка FORTRAN-SC в 1984 – 1989 [1,2,13] годах в виде совместного проекта с лабораторией новых разработок фирмы IBM в г. Бёблинген, ФРГ. FORTRAN-SC соединяет в себе концепции высокоточной арифметики и удобства использования. Начиная с 1987 г. язык FORTRAN-SC применяется во многих университетах и научно-исследовательских институтах. В августе 1990 г. его название изменилось на ACRITH-XSC, который стал программным продуктом фирмы IBM (IBM Program Product) и распространяется теперь во всем мире [6,7].

ACRITH-XSC работает на компьютерах IBM /370 и /390 под операционной системой VM/CMS. Спектр его применений весьма обширен. Почти каждое применение, для которого надежность и достоверность численных результатов играют фундаментальную роль, может быть запрограммировано на ACRITH-XSC. В настоящее время объем вычислительных задач постоянно растет

и этот рост приводит к тому, что вычисления все чаще и чаще производятся на суперкомпьютерах с гигантской производительностью. Ввиду того, что каждая операция с плавающей запятой связана с ошибкой округления, значение надежности численных результатов постоянно возрастает.

Диапазон использования языка ACRITH-XSC широк: электротехническая и электронная промышленность, машиностроение, химическая промышленность, обработка изображений в медицине, авиационная и космическая техника, вычисление и контроль электрических сетей, строительная статика, проектирование и оптимизация регулировочных систем, разведка полезных ископаемых, моделирование. Важная область применения ACRITH-XSC – научно-исследовательская и учебная работа в университетах и в отделах новых разработок на промышленных предприятиях.

Главные концепции ACRITH-XSC:

Интервальная арифметика

Оптимальное преобразование констант и данных

Векторно-матричная арифметика

Динамические массивы

Подмассивы

Определение и переопределение операций

Точное вычисление выражений типа скалярных произведений

Встроенные программы ACRITH для решения стандартных задач

Некоторые концепции языка ФОРТРАН 90, стандарт которого был опубликован в 1991 г., реализованы в языке ACRITH-XSC. Так, например, допустимы функции, выдающие массив в качестве результата, а также определяемые и переопределяемые операции. Очень удобную для пользователя возможность предоставляют динамические массивы и подмассивы. Они позволяют оптимально использовать память и упростить запись программы. Собственные же средства для повышения точности и проверки результатов – это главные особенности ACRITH-XSC. Таким

образом, в распоряжение пользователя предоставлены высоко- точная матричная арифметика с оптимальным скалярным произ- ведением двух векторов, интервальная арифметика и высокоточ- ное вычисление выражений типа скалярных произведений.

ACRITH-XSC содержит новые типы данных для вещественных и комплексных интервалов, а также все типы данных для скаля- ров, векторов и матриц, которые могут быть кодированы в виде простых переменных. Арифметические операции предполагают направленные округления к *ближайшему*, *вниз* и *вверх* и облада- ют наивысшей точностью, т. е. между точным и округленным результатом вычислительной операции не существует предста- вимого машинного числа. Это обозначается также как 1 ulp (*unit in last place* = единица последнего разряда) точности. В слу- чае округления к *ближайшему* точность доходит даже до $\frac{1}{2} \text{ ulp}$. Особая основная операция – это оптимальное скалярное произ- ведение двух векторов. Оно вычисляет простые произведения компонент вектора и их сумму точно без ошибки округления с помощью аккумулятора с фиксированной запятой. Окончатель- ный результат округляется только один раз к *ближайшему* числу с плавающей точкой в желаемом направлении [10,11]. Это слу- жит основой для максимально точного определения умножений вектор-вектор, матрица-вектор и матрица-матрица. Такие опе- рации свободны от эффектов стирания разрядов и представляют собой основу для вычисления проверенных и надежных резуль- татов численных задач [9,12]. Они являются особенно важным вспомогательным средством в алгоритмах, которые используют методы коррекции дефектов.

Число обычных стандартных функций увеличено. Предста- влены в распоряжение функции SQR, SQRT, EXP, LOG, тригономе- трические, гиперболические, их обратные и еще ряд других. До- бавлены типы для вещественных и комплексных интервалов. При этом точность всегда лучше, чем две единицы последнего разря- да (2 ulp), а в большинстве случаев даже меньше 1 ulp . Кроме того, ACRITH-XSC предлагает оптимальный контроль округле- ний при преобразовании констант и входных/выходных данных, так что появляющиеся ошибки преобразования, которые проис- ходят от внешнего десятичного и внутреннего шестнадцатерично- го представления чисел, могут быть минимизированы и оценены.

Интервальная арифметика

С помощью новых типов данных `INTERVAL` и `COMPLEX INTERVAL` в `ACRITH-XSC` реализован комплект операций интервальной арифметики, а именно операции для интервальных скаляров, векторов и матриц. Все основные действия выполняются с максимальной точностью при помощи округлений, направленных *вниз* и *вверх*. Для интервальных констант, а также входных и выходных данных имеется специальная нотация, которая гарантирует правильное округление десятичных данных с наименьшей ошибкой при преобразовании. Пользоваться интервальными операциями просто. Они поддерживаются многочисленными функциями преобразования типа и стандартными функциями, однако последние предусмотрены только для скалярных интервалов. Интервальная арифметика является фундаментом для проверки вычисленного решения посредством включающего алгоритма [12]. Кроме того, интервальная арифметика делает возможным контроль округлений, анализ устойчивости и чувствительности и удобное решение задач, входные данные которых имеют диапазон допусков.

Векторно-матричная арифметика

`ACRITH-XSC` содержит полную векторно-матричную арифметику для точечных и интервальных данных. В распоряжение предоставлены все традиционные математические векторно-матричные операции с максимальной точностью в виде встроенных операций. Сложение и вычитание определены покомпонентно, для интервалов также определены операции *intersection* и *convex hull*, которые дают пересечение и выпуклую оболочку двух интервалов. Встроенными операциями также являются умножение и деление на скаляр, которые выполняются покомпонентно. Произведения *вектор · вектор*, *матрица · вектор* и *матрица · матрица* выполняются с помощью оптимального скалярного произведения, которое всегда гарантирует наивысшую точность. Самой собой разумеется, предоставлены в распоряжение также условные операторы для векторов и матриц, как, например, операторы `.LT.` и `.NE.`, которые вырабатывают значение истина (`true`) тогда и только тогда, когда все сравнения составляющих истинные

(LT), или когда по крайней мере одно сравнение составляющих истинно (NE).

Векторы и матрицы могут быть использованы в программе просто как скалярные переменные; "естественная" математическая нотация может быть таким образом достигнута для выражений, содержащих массивы. Программирование значительно упрощается за счет того, что избегаются длинные последовательности вызовов подпрограмм, сложные вложенные циклы и часто вызывающие ошибки индексные выражения.

Динамические массивы и подмассивы

Динамические массивы и подмассивы значительно упрощают использование векторов и матриц, а также массивов более высокой размерности. При исполнении программы динамические массивы занимают только необходимое место в памяти и предотвращают перекомпиляцию задач с различными размерами массивов. Память может быть распределена и явно, и неявно, таким образом, работа пользователя облегчается. Дальнейшее преимущество заключается в возможности изменения размеров и индексных пределов массива в ходе работы. Само собой разумеется, динамические массивы совместимы со статическими массивами. Для повышения надежности при вызовах подпрограмм в ACRITH-XSC реализована проверка диапазона индексов и типов аргументов. На основании этих концепций отпадает необходимость работы с часто вызывающими ошибки COMMON-блоками или с определенными пользователем рабочими массивами. Динамические массивы делают возможным реализацию функций с результатами типа массив и программирование произвольных векторных выражений.

Динамические массивы определяются словом DYNAMIC в разделе определений, при этом определяются тип данных и количество индексов, но не размер массива и диапазон индексов. Операторы ALLOCATE и FREE резервируют и, соответственно, освобождают память. Рассмотрим следующий пример:

```
DYNAMIC / COMPLEX INTERVAL (:,:) / Dynmatrix
```

```
DYNAMIC / REAL (:) / X; Y
```

```

READ (*,*) n
ALLOCATE Dynmatrix (n, -n:n), X (0:2*n)
READ (*,*) Dynmatrix, X
Y = Dynmatrix * X
FREE Dynmatrix, X

```

Здесь определяются комплексная интервальная матрица *Dynmatrix* и вещественные векторы *X*, *Y* как динамические массивы. Прежде всего вводится целое положительное число *n*, а затем отводится место в памяти для $(n, 2n + 1)$ -матрицы и вектора с $2n + 1$ составляющими. Присваивание, которое следует за описанием *Dynmatrix* и *X*, осуществляет неявное (автоматическое) размещение вектора *X* с *n* составляющими. В последней строчке освобождается место, занятое в памяти *Dynmatrix* и *X*.

Подмассивы позволяют непосредственно обращаться к любым прямоугольным частям массивов. Они могут быть использованы как операнды в любых видах векторных операций. Строки и столбцы матриц могут быть обработаны в виде простых векторов. Индексация подмассивов очень проста, например, вектор пятой строки $(10, 10)$ -матрицы *A* изображается посредством $A(5, :)$, вектор седьмого столбца — посредством $A(:, 7)$, подматрица, которая располагается от третьей строки к пятой строке и от четвертого столбца к восьмому столбцу, обозначается посредством $A(3:5, 4:8)$. Подмассивы полностью совместимы с целыми массивами и сокращают количество трудоемких индексных циклов.

Определение и переопределение операций

ACRITH-XSC разрешает определение новых и переопределение предопределенных операторов. При этом операторы с одним или двумя операндами могут быть определены для любых функций с одним или двумя аргументами. Операторная концепция распространяется на скалярные и векторные операнды. Из-за того, что в ACRITH-XSC реализованы функции с результатом типа массив, возможно определение матричных функций, как, например, функции нахождения обратной матрицы или матричной степен-

ной функции. Их использование происходит таким же образом, как и использование функций в скалярном выражении. Возможность пользователя определять свои операции открывает новые области применения. Пользователь может определять свои собственные операции для своей специальной задачи таким образом, что программа будет в значительной мере близка к математической формулировке задачи. Следствием является простое кодирование сложных выражений и сокращение вероятности ошибок при программировании.

Операторы определяются в разделе определений словом OPERATOR, за ним следует операторный символ и функция, которая реализует оператор вместе с определением аргументов и результата. Посредством

OPERATOR // = MODULO (INTEGER, INTEGER) INTEGER

OPERATOR .INV. = INVERT (INTERVAL(:, :)) INTERVAL(:, :)

стандартный оператор // может быть переопределен для определения вычета при целочисленном делении, а унарный оператор .INV. — для обращения интервальной матрицы. Сами операции определяются в пользовательских функциях MODULO или INVERT.

Точная обработка СП-выражений

Единственная в своем роде возможность в ACRITH-XSC — это точная обработка выражений типа скалярных произведений (СП-выражений), которые состоят из сумм вещественных и комплексных констант, переменных, векторов, матриц или их простых произведений. Они обрабатываются всегда с помощью оптимального скалярного произведения без ошибок округления и могут либо присваиваться переменным нового типа данных DOT PRECISION или COMPLEX DOT PRECISION, либо округляться с наивысшей точностью к ближайшему числу с плавающей точкой. СП-выражения, которые предотвращают типичные эффекты стирания разрядов, представляют собой очень важное вспомогательное средство для методов коррекции дефектов [9,12]. Вместе с интервальной арифметикой они разрешают проверку вычисленного решения с помощью включающего алгоритма.

СП-выражение заключается в скобки, перед которыми ставится специальный символ '#' и, если желательно округление точного результата, еще один символ, который определяет вид округления. Внутри такого выражения может быть для математической Σ -нотации использовано слово Sum. Так, например,

```
DOUBLE REAL U(1000), V(1000), X, Y, Z
```

```
.....
```

```
Z = #*(U*V - X*Y + SUM (U(i) * U(1001-i), i=1,1000))
```

присваивает Z наилучшее возможное приближение с плавающей точкой значения выражения, причем следующий за '#' символ '*' специфицирует округление к *ближайшему* числу.

Встроенные в ACRITH программы для решения стандартных задач

Наряду с упомянутыми средствами в ACRITH-XSC встроена библиотека подпрограмм IBM ACRITH [5]. Вместе с основными арифметическими операциями и стандартными математическими функциями, которые обрабатываются с наивысшей точностью, подпрограммная библиотека содержит стандартные программы для решения стандартных задач из численной математики. Это, например, *обращение матриц, поиски нулей у полиномов, разреженные линейные системы уравнений, вычисление собственных значений* и еще некоторые другие. Решения всегда верифицированы и находятся в узких интервалах. Если какая-либо задача не имеет решения, то выдается соответствующее сообщение. Дружественный интерфейс обеспечивает удобный для пользователя вызов подпрограмм.

Литература

1. 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 [12].

2. Davidenkoff, A.: *ACRITH-XSC – A programming language for scientific/engineering computation. (A survey)*. Talk at GAMM 91 Conference, Cracow, Poland, April 1-5, 1991. In: *Zeitschrift für angewandte Mathematik und Mechanik (ZAMM)*, 72 (1992) 6, T 465 – T 467, 1992.
3. Geörg, S., Hammer, R., Neaga, M., Ratz, D., Shiryaev, D.: *PASCAL-XSC: a portable development system for scientific computation and numerical data processing*. Institute for Applied Mathematics, University of Karlsruhe, D-7500 Karlsruhe, Germany, 1992.
4. Hammer, R., Neaga, M., Ratz, D., Shiryaev, D.: *PASCAL-XSC: A new language for scientific computing*, *Interval Computations* 2, pp. 47–81, 1991. (in Russian).
5. IBM *high accuracy arithmetic subroutine library (ACRITH)*. *Program description and user's guide*. SC33-6164-02, 3rd edition, IBM Corporation, 1986.
6. IBM *high accuracy arithmetic – extended scientific computation. General information*, Version 1, Release 1, SC33-6461-01, IBM Corporation, 1990.
7. IBM: *high accuracy arithmetic – extended scientific computation. ACRITH-XSC language reference, Version 1, Release 1*, SC33-6462-00, IBM Corporation, 1990.
8. Klatte, R., Kulisch, U., Neaga, M., Ratz, D., Ullrich, Ch.: *PASCAL-XSC, language reference with examples*. Springer-Verlag: Berlin, Heidelberg, 1992. Russian version in preparation.
9. Kulisch, U. (ed.): *Wissenschaftliches Rechnen mit Ergebnisverifikation, Eine Einführung*. Vieweg-Verlag: Akademie-Verlag, Berlin, 1989.
10. Kulisch, U., Miranker, W. L.: *Computer arithmetic in theory and practice*. Academic Press, New York, 1981.
11. Kulisch, U., Miranker, W. L.: *The arithmetic of the digital computer: a new approach*. *SIAM Review* 28, no. 1, 1986.
12. Kulisch, U., Stetter, H. J. (eds.): *Scientific computation with automatic result verification*. Springer-Verlag: Wien, New-York, 1988.
13. Metzger, M., Walter, W.: *FORTTRAN-SC: A programming language for engineering scientific computation*. In: 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
 e-mail: AE12@DKAUNI2.BITNET or
 AE12@IBM3090.rz.uni-karlsruhe.de (Internet)

и ста-
 гление
 ет вид
 атема-
 ер,

ающей
 символ

строена
 лжны-
 матиче-
 точно-
 ле про-
 емати-
 номов,
 бствен-
 ифици-
 бо зада-
 бщение.
 льзова-

Walter, W.:
Scientific com-
 37. A copy