What Is Different about AMTRAN?

RICHARD J. PLOCICA

Computation Laboratory, Marshall Space Flight Center, NASA, Hunstville, Alabama
and
Physics Department, University of South Florida, Tampa, Florida

As a programming language, AMTRAN is designed to satisfy two objectives: the reduction of programming cost and effort by at least an order of magnitude, and the provision of a semiautomatic numerical analytical problem solving system. It resembles a blend of FORTRAN and ALGOL but possesses certain additional features. These include the following.

1. Automatic array arithmetic and array manipulation facilities: When performing operations upon arrays, it is usually unnecessary to write DO loops since the DO loops are implied. A set of compact operators for the manipulation of arrays is also present, e.g., REFLECT, TRANSPOSE, the calling of a submatrix, etc. The IF test in AMTRAN also tests and operates automatically upon arrays.
2. Dynamic memory allocation: Dimension statements are unnecessary and arrays are redimensioned as required during execution.
3. "Convenience" operators such as TYPE, SET, SCOPE, RANGE, etc.
4. An expandable instruction set: It is intended that the AMTRAN language be open-ended and that the user be able to add his own specialized, high-level instructions to the system (as seen by him). To this end, the concept has been extended to include binary operations, symbolic operands, flexible formatting arrangements, and other elaborations.
5. List and text processing capabilities: Code and test strings can be stored and manipulated in approximately the same way as data.
6. A built-in instruction manual: Since many users prefer to "learn by doing," the AMTRAN instruction manual has been built into the system.

Specific instructions regarding the use of the various operators may be displayed on the scope upon request.

7. Special conversational-mode facilities: Certain features specific to the interactive mode, such as on-line editing, automatic output formatting, dialog control, and listing and punching of user-defined programs are present in the system.

8. Graphic input and output capabilities are present: This includes a general-purpose graphic display routine which automatically formats displays, subject to a number of user-specified "modifiers" such as HACHURE, POLAR, LOG.X, VECTOR, GRID, etc.

9. Higher-level operators: There are commonly used operators such as SUM, DELTA, MIN, MAX, etc., which facilitate numerical operations. There are also adaptive mathematical operators such as REPRESENT, INTEGRAL, DERIV, SOLVE, ZEROES, MINIMAX, INVERT, LET, etc.

Other features which characterize AMTRAN include the ALGOL IF test with Boolean and array testing features, subscripted subscripts, recursive calling capabilities, card and printer input and output, efficient batch-processing of console-generated programs, "picture-book" formatting, an on-line trace, and nearly all of the facilities of FORTRAN IV.

The principal motivation for developing these rather extensive programming capabilities was to reduce programming cost and effort over FORTRAN IV by factors ranging from 3 : 1 for simple problems to 15 : 1 or 20 : 1 for more complicated mathematical problems. One of these AMTRAN benchmark programs is shown below.

1. X = RANGE A,B, INTERVALS.
2. PRINT $2/\sqrt{\pi}$ ∫ EXP(−X X).
3. NAME.THIS ERF.

These three instructions will cause the computer to compute and to print out the numerical representation of the Gaussian error function over the numerical range from A to B, storing the instruction as a "procedure" under the name ERF. The coding and checkout of this array operator requires about 2 minutes in AMTRAN. The coding, checkout, and timing of a corresponding FORTRAN subroutine is left as an exercise for the reader.

These results are preliminary and should be taken with a grain of salt. Nevertheless, sizable reduction in programming time appears to be occurring.