XVII. COMPUTATION RESEARCH*

Martha M. Pennell
Gail M. Fratar
R. L. Rappaport
Veronica E. McLoud
Kathleen A. Szabo

A. GENERALIZED POLYNOMIAL ROOT-FINDING PROGRAM FOR A TIME-SHARED COMPUTER

The availability of the Project MAC time-sharing system[1] has enabled us to write a computer program, called Quixot Dulcin (see below), which in turn automatically writes, compiles and executes a MAD[2] program to find roots of polynomials whose coefficients may be expressed symbolically as functions of variables called parameters. There may be any number of parameters and the user has the option of declaring one of these to be varying. At execution time the program loops around the value of this parameter thus enabling the user to compute the functional dependence of the roots.

To use this program all one needs to know is how to write his coefficients as MAD arithmetic expressions[3] and how to operate his remote console. Quixot Dulcin leads the user through a series of questions and from his responses creates the necessary MAD code. Only in a time-sharing environment is such an approach feasible. In their studies of dispersion relations, members of the Plasma Electronics Group have made such extensive use of Quixot Dulcin that we no longer have any requests to program such a problem.

The user is first queried for the degree of his polynomial (it must be ≤ 19) and the name he wishes his program to have so that he may refer to it in the future. He then is asked to state the symbols (6 or less alphanumeric characters) he wishes to use for his parameters. Next he is asked for the MAD algebraic expression for the real and imaginary part of each coefficient. These expressions are functions of the symbols he has just typed in and may contain the following: sine, cosine, square root, arctangent, arcosine, absolute value. With this information Quixot Dulcin writes a program in the MAD language and tries to translate it into the equivalent binary code. If, however, the user has typed in an incorrect MAD expression (i.e., an incomplete set of parentheses or a missing operation), Quixot Dulcin enters a correction mode whereby the user may make the necessary changes and a second try to compile the binary code is attempted. The correction mode is entered and re-entered automatically until the binary code is created. Quixot Dulcin then loads this program and transfers control to it. Immediately the user is asked for the numerical values for each of the parameters whose symbols he had just previously typed in. The program then computes the coefficients and roots, and types them both on the user's console. The user is then asked for the next value of the parameter he had declared to be varying. One push on the break button enables him to

*This work was supported in part by the National Science Foundation (Grant GK-524).

```
LOADGO QUIXOT DULCIN
W 2331.7
EXECUTION.
COMPLEX ROOTS OF A COMPLEX POLYNOMIAL, GIVEN ALGEBRAIC COEFFICIENTS
WHAT WOULD YOU LIKE TO NAME YOUR PROGRAM.  (SIX LETTERS OR LESS, PLEASE.)
   ***   ***   ***   ***
*** BE CAREFUL*** FOR IF THERE ARE ANY EXISTING FILES WITH THIS FIRST NAME,
THEY WILL BE DESTROYED
   ***   ***   ***   ***
TEST
TYPE THE DEGREE OF YOUR POLYNOMIAL AS A TWO-DIGIT INTEGER,E.G. 12 OR 05
03
DO NOT USE AS PARAMETER NAMES 'I' AND       'DEG'
TYPE THE NAME OF THE VARYING PARAMETER.
ALPHA
TYPE THE NAMES OF ALL OTHER PARAMETERS, ONE PER LINE, THEN SKIP A LINE.
BETA
GAMMA

COEFFICIENTS--SKIP A LINE AFTER TYPING EACH COEFFICIENT.
DO NOT TYPE PAST POSITION MARKED BY                 *
TYPE REAL ALGEBRAIC COEFFICIENT OF THE  0 POWER.    *
ALPHA+BETA+GAMMA

TYPE IMAG ALGEBRAIC COEFFICIENT OF THE  0 POWER.    *
2.+SQRT.(ALPHA*ALPHA)

TYPE REAL ALGEBRAIC COEFFICIENT OF THE  1 POWER.    *
3.

TYPE IMAG ALGEBRAIC COEFFICIENT OF THE  1 POWER.    *
0.

TYPE REAL ALGEBRAIC COEFFICIENT OF THE  2 POWER.    *
-GAMMA+5.

TYPE IMAG ALGEBRAIC COEFFICIENT OF THE  2 POWER.    *
0.

TYPE REAL ALGEBRAIC COEFFICIENT OF THE  3 POWER.    *
0.

TYPE IMAG ALGEBRAIC COEFFICIENT OF THE  3 POWER.    *
-BETA

DO YOU WANT TO MAKE CORRECTIONS
NO
 FILE   TEST   MAD CREATED
LENGTH 00407.  TV SIZE 00012.  ENTRY 00151
   ***   ***   ***   ***
```

```
IN ØRDER TØ RERUN YØUR PRØGRAM IN THE FUTURE, YØU MUST TYPE THE FØLLØWING SEQUENCE.
LØADGØ   TEST BUTTØN MADINT RØØTS
   ***   ***   ***   ***
EXECUTIØN.
PRESS 'BREAK' BUTTØN TØ CHANGE PARAMETERS.
TYPE  BETA
.1
TYPE  GAMMA
-.2
TYPE  ALPHA
3.


CØEFFICIENTS

 PØWER   REAL            IMAG

    0   .29000E 01    .50000E 01
    1   .30000E 01    .00000E 00
    2   .52000E 01    .00000E 00
    3   .00000E 00  -1.00000E-01

  3 RØØTS
     REAL              IMAG
    .558653E 00     -.519951E 02
   -.819058E 00      .881453E 00
    .260404E 00     -.886375E 00


TYPE  ALPHA
2.


CØEFFICIENTS

 PØWER   REAL            IMAG

    0   .19000E 01    .40000E 01
    1   .30000E 01    .00000E 00
    2   .52000E 01    .00000E 00
    3   .00000E 00  -1.00000E-01

  3 RØØTS
     REAL              IMAG
    .562232E 00     -.519989E 02
   -.793484E 00      .745879E 00
    .231252E 00     -.746986E 00


TYPE  ALPHA
> QUIT,
R  26.816+19.800
```

change the values of all the parameters; two pushes ends the program.

The subroutine which actually computes the roots is Share Distribution No. 692 modified for use on the time-sharing system. It uses Muller's Method.[4] The main control program written in MAD works as follows. MAD statements are created from the MAD expressions that the user has typed in. These statements are inserted between already existing MAD statements which query the user for his numerical values and MAD statements to compute and print out the answers.

Currently work is being done to improve both the accuracy and the speed of the subroutine to compute the roots. In addition a second program is being checked out which will enable the user to type in a complex algebraic expression for his coefficients, i.e., he will not have to perform the algebra to separate his coefficients into their real and imaginary parts.

Martha M. Pennell

References

1. C. Marceau, a former member of the Research Laboratory of Electronics, wrote the original program.

2. B. Arden, B. Galler, and R. Graham, "The Michigan Algorithm Decoder," University of Michigan, April 1965.

3. Ibid, p. 12.

4. D. E. Muller, "A Method for Solving Algebraic Equations Using an Automatic Computer," Mathematical Tables and Other Aids to Computation (National Research Council, Washington, D.C., October 1956), pp. 208-215.