PROJECT THEMIS: INFORMATION PROCESSING

AUBURN UNIVERSITY PROJECT THEMIS
TECHNICAL REPORT NUMBER AU-T-6

POLYNOMIAL MANIPULATION SYSTEM -
FORTRAN IV PROGRAM

PREPARED BY
THE COMPUTER CENTER
LELAND H. WILLIAMS, DIRECTOR
AUBURN UNIVERSITY, AUBURN, ALABAMA
NOVEMBER, 1969

CONTRACT DAAH01-68-C-0296
ARMY MISSILE COMMAND
HUNTSVILLE, ALABAMA

DISTRIBUTION OF THIS DOCUMENT IS UNLIMITED.

APPROVED BY:  SUBMITTED BY:

L. H. Williams  L. H. Williams
Director, Computer Center  Director, Computer Center
Associate Professor of  Associate Professor of
Mathematics  Mathematics
THEMIS Project Director

Ben T. Lanham, Jr.
Vice President for Research
Auburn University

AUBURN UNIVERSITY
AUBURN, ALABAMA  36830
FOREWORD

This report is a technical summary reporting the progress of a study conducted in the Mathematics Department and the Computer Center of Auburn University. The study is focused toward fulfillment of Contract No. DAAH01-68-C-0296 granted to Auburn University by the Army Missile Command, Huntsville, Alabama.

ACKNOWLEDGEMENT

The programming and documentation for this report were done by Mr. J. R. Sidbury.
ABSTRACT

A FORTRAN IV program which implements the Polynomial Manipulation System (PMS) is presented and described. PMS uses the Euclidean Algorithm to reduce a system of polynomials in several variables to a resultant system which can be solved sequentially as polynomials in one variable (Kronecker's method). PMS is described briefly and references are given to more complete discussions and to other pertinent literature.
TABLE OF CONTENTS

I. INTRODUCTION ......................................................... 1
II. PROGRAM DESCRIPTION ............................................. 2
III. BASIC FLOW CHARTS .................................................. 3
IV. INPUT/OUTPUT .......................................................... 6
V. EFFICIENCY AND FUTURE WORK ...................................... 6
VI. REFERENCES ............................................................ 7

APPENDICES
A FORTRAN IV Program Listing ....................................... 8
B Example ........................................................................ 16
C Document Control Data - R & D, DD Form 1473 ............... 18
I. INTRODUCTION

The Polynomial Manipulation System (PMS) uses the Euclidean Algorithm for finding the eliminant and the greatest common divisor (g.c.d.) of two multi-variable polynomials. All polynomials involved are represented symbolically; PMS is a computer program whose input is the symbolic representation of two polynomials and whose output is (normally) the symbolic representation of their g.c.d. and eliminant.

The underlying theory and the application of PMS to the problem of solving systems of polynomial equations is discussed in [1], [2], and [3]. The present report describes a FORTRAN IV implementation of PMS developed on the IBM 360 Model 50 at Auburn University.

The program is described in Section II, the basic flow charts are given in Section III, Input/Output is discussed in Section IV, and efficiency of the method is discussed in Section V along with possible future work. The FORTRAN program is reproduced in Appendix A. Appendix B contains a simple example of the use of PMS for reduction of three polynomial equations in three variables to a resultant system which can be solved in sequence as polynomials in one variable.
II. PROGRAM DESCRIPTION

The PMS program is basically a main program with four subroutines, only one of which is significant. The other three subroutines are used for output, format headings on printed output and scaling of coefficients when they become large enough to possibly cause an overflow. In its present form the program is limited in that it is set up to use only 175K of IBM 360 storage. This limitation places constraints on the program which allows storage of only 50,000 polynomial entries (each term has n + 1 entries where n is the number of variables in the polynomial) which are presently set up as follows:

1) The pair of polynomials has, at most, four variables.
2) Each polynomial has at most 3160 terms.
3) The leading coefficient polynomials, to be defined below, can have, at most, 400 terms.

Minor modifications could increase the number of terms or variables or size of leading coefficients at the cost of decreasing the others or by use of a greater amount of machine storage. Still larger polynomials could be processed by use of tape, disc or other storage, but this has not been effected since such increases would only tend to accentuate certain disadvantages of the method to be discussed in Section V.

Consider the pair of polynomials \( U, T : \mathbb{F}^{N-R} \). Let \( x_1, \ldots, x_n \) denote the variables. Each of these polynomial functions can be considered as a polynomial in \( x_1 \) whose coefficients
would then be polynomial functions from $E^{n-1}$ to $R$. Let $U^0$ and $T^0$ denote the polynomials in $x_2, \ldots, x_n$ which are the leading coefficients of $U$ and $T$ respectively considered as polynomials in $x_1$, and let $u$ and $t$ denote the degrees of $U$ and $T$ in $x_1$. We may assume $t > u$. Consider the polynomial $R$ defined by

$$R = U^0T - T^0Ux_1^{t-u}.$$ 

$R$ is a polynomial in $x_1, \ldots, x_n$. Considering $R$ as a polynomial in $x_1$ with polynomial coefficients, it is seen that $\deg(R) < t$. Let $\deg(R) = r$. If $r > u$, let $T = R$ and repeat above procedure. If $r < u$, let $T = U$ and $U = R$ and repeat the above procedure. After a finite number of applications of this algorithm a polynomial $R$ will be found whose degree in $x_1$ is zero. Thus $R$ will be a polynomial in $x_1, x_2, \ldots, x_n$. It is easily seen that at each stage $R$ has any zeros that are common to $U$ and $T$. The $R$ which is free of $x_1$ is called the eliminant of $U$ and $T$.

### III. BASIC FLOW CHARTS

The flowcharts for output and scaling will be omitted as their detail is not significant to the main purpose of the program. The main program flow chart is given on page 4.
MAIN PROGRAM

Read U,T

↓

Compute U⁰, T⁰

↓

Print U,T

↓

3

Determine which of U and T has greatest degree in x₁. If it is T continue, if not interchange U and T, U⁰ and T⁰.

↓

Call RESIDUE
and Form R

↓

Scale if necessary

↓

If R is free of x₁, print R and return to beginning to read two new polynomials. If not, let T = R.
RESIDUE SUBROUTINE

Form $U^0 T$

Form $-T^0 U x_{t-u}^1$

$R = U^0 T - T^0 U x_{t-u}^1$

$T = R$

Calculate new $T^0$

Is $t = 0$?

no

Return to Continue Processing

yes

Return. We Have Eliminant
IV. INPUT/OUTPUT

The polynomials U and T are read in separately. The first card for each polynomial contains the number of variables in the polynomial in columns 31-34 in integer format, right justified. The number of terms in the polynomial appear in columns 35-38 in integer format, right justified. Following this card the terms of the polynomial appear, one term per card. The coefficient appears in columns 1-16 in E format; following this are the exponents of the variables right justified in integer format in columns 17-21, 22-26, 27-31, etc.

The output of this program is available in any medium, although the program is currently set up for printed output only.

V. EFFICIENCY AND FUTURE WORK

The PMS program has not proved useful as a method of reducing polynomial systems of equations to a resultant system for the following reasons:

1) Storage efficiency is low. An inordinate amount of core is needed to process many simple appearing systems of equations.

2) Time efficiency is low. Extreme amounts of time are needed to solve all but the most simple problems. As few as four equations in four variables with small exponents (on the order of ten or less) take many hours of machine time to reach a solution. Simpler problems are solvable in
small amounts of time, but other methods without these disadvantages can be used to solve these systems.

3) Certain types of systems give solutions which have a low order of accuracy. Several articles, [4], [5], [6], have been published discussing this problem as well as the two above.

The basic PMS program will be examined and modified to determine if it is of value in algebraically solving simple systems of differential equations.

VI. REFERENCES


53 IUMAX(K-1, JU) = IR(K, J)
55 CONTINUE
DO 10 J = 1, NTERM1
10 U(J) = R(J)
DO100 = 1, MAX
99 U(J) = IR(I, J)
DO 20 J = 1, NTERM2.
20 R(J) = T(J)
DO91 = 1, NVAR2
91 TR(I, J) = IT(I, J)
CALL PRINT(2, NTERM2, 0, 0, NVAR2)
IF(NVAR2 .GE. NVAR1) GO TO 1075
MAX = NVAR1
NVAR2 = NVAR2 + 1
DO 104 J = 1, NTERM2
104 IR(J, K) = 0
NVAR? = MAX
104 IR(J, K) = 0
NVAR? = MAX
1075 MAX = NVAR2
JPWRT = 0
DO 56 J = 1, NTERM2
56 IF(JPWRT .GE. 56, 57) 56, 57, 56
57 JT = JT + 1
TMAX(JT) = R(J)
IF(ABS(TMAX(JT)) .LT. 1.E6) LM = 1
DO 58 K = 1, MAX
58 ITMAX(K-1, JT) = IR(K, J)
56 CONTINUE
DO 80 J = 1, NTERM2
T(J) = R(J)
DO81 = 1, MAX
81 IT(I, J) = IR(I, J)
107 IF(JPWRT .GT. JPWRU) 70, 71, 71
70 NN = JPWRT
JPWRT = JPWRU
JPWRU = NN
MAXT = NTERM1
IF(NTERM2 .GT. NTERM1) MAXT = NTERM2
DO80 I = 1, MAX
80 TEMP = U(I)
U(I) = T(I)
T(I) = TEMP
DO80 J = 1, MAX
TEMP = U(J, I)
U(J, I) = IT(J, I)
80 IT(J, I) = TEMP
NN = JJ
JJ = JT
JT = NN
NN = NTERM1
NTERM1 = NTERM2
NTERM2 = NN
NN = NVAR1
NVAR1 = NVAR2
NVAR2 = NN
JJ = JU
   IF(JT.GT.JJ) JJ = JT
   MA = MAX - 1
   DO 73 J = 1, JJ
   TW = UMAX(J)
   UMAX(J) = TMAX(J)
   TMAX(J) = TW
   DO 73 K = 1, MA
   IUT = UMAX(K, J)
   UMAX(K, J) = IUT
  73 IUT = UMAX(K, J)
   CONTINUE
C
   IF(LM.EQ.1) CALL SCALE2(INTERM1, INTERM2, MAX)
   LM = 0
   101 CALL RESIJU(INTERM1, INTERM2, MAX, JPRURU, JPRURT, JU, JT)
   IF(LM.EQ.1) CALL SCALE2(INTERM1, INTERM2, MAX)
   IF(LM.EQ.1) GOTO 3001
   IF(LS2.EQ.1) CALL SCALE2(INTERM1, INTERM2, MAX)
   LS2 = 0
   3001 LM = 0
   300 IF(LS1.EQ.0) GOTO 107
   LS1 = 0
   I = 0
   301 CALL PRINT(3, JU, INTERM1, INTERM2, MAX)
   WRITE(6, 10000) ITIMES
   10000 FORMAT(1H, 5HSCALE, 13)
   GO TO 100
   1 FORMAT(30X, 214)
   4 FORMAT(F16.7, 1015)
   END
SUBROUTINE FOR PRINTING THE TWO POLYNOMIALS AND THE RESIDUE

SUBROUTINE PRINT (L, JU, NTERM1, NTERM2, N)

IMPLICIT INTEGER*2(IN)

COMMON /MAXO(400), IUMAX(3,400), I1MAX(3,400), TM1AX(400), R(3160),
NIR(4,3160), ITRATE, ITMAX, IU(3160), IV(4,3160), T(3160), TT(4,3160),
NL87, LS1, L1

C******************************************************************************
C******************************************************************************

IF(1.E0.4) WRITE(6,310)
IF(1.E0.7) WRITE(6,311)
IF(1.E0.1) WRITE(6,312)
50 CALL CRITN(H3)
   WRITE(6,*315) JU(1:K), (I(K(I,K),I=1,N))
315 CONTINUE
   K = N-1
503 CONTINUE
   RETURN
43 FORMAT(1X,E15.7,2X,10(15,E1))
311 FORMAT(1X,'THE POLYNOMIAL IS:')
312 FORMAT(1X,'THE POLYNOMIAL IS:')
310 FORMAT(1X,'THE ELIMINANT IS:')
540 FORMAT(1X,214)
541 FORMAT(1X,E15.7,1015)
END
SUBROUTINE FOR PRINTING HEADINGS

SUBROUTINE PRINT(k)

THIS SUBROUTINE MERELY PRINTS COLUMN HEADINGS FOR THE
VARIABLES DEPENDING ON THE NUMBER OF VARIABLES,
THAT IS ITS ONLY PURPOSE. IT WILL HANDLE UP TO 10
VARIABLES.

GO TO (21, 22, 22, 23, 24, 25, 26, 27, 28, 29, 30), K

WRITE(6,31)
RETURN

WRITE(6,32)
RETURN

WRITE(6,33)
RETURN

WRITE(6,34)
RETURN

WRITE(6,35)
RETURN

WRITE(6,36)
RETURN

WRITE(6,37)
RETURN

WRITE(6,38)
RETURN

WRITE(6,39)
RETURN

WRITE(6,40)
RETURN

31 FORMAT(1HO, 11HCOEFFICIENT, 10X, 4HX(1))
32 FORMAT(1HO, 11HCOEFFICIENT, 10X, 4HX(1), 5X, 4HX(2))
33 FORMAT(1HO, 11HCOEFFICIENT, 10X, 4HX(1), 5X, 4HX(2), 5X, 4HX(3))
34 FORMAT(1HO, 11HCOEFFICIENT, 10X, 4HX(1), 5X, 4HX(2), 5X, 4HX(3), 5X, 4HX(4))
35 FORMAT(1HO, 11HCOEFFICIENT, 10X, 4HX(1), 5X, 4HX(2), 5X, 4HX(3), 5X, 4HX(4),
1,5X,4HX(5))
36 FORMAT(1HO, 11HCOEFFICIENT, 10X, 4HX(1), 5X, 4HX(2), 5X, 4HX(3), 5X, 4HX(4),
1,5X,4HX(5), 5X, 4HX(6))
37 FORMAT(1HO, 11HCOEFFICIENT, 10X, 4HX(1), 5X, 4HX(2), 5X, 4HX(3), 5X, 4HX(4),
1,5X,4HX(5), 5X, 4HX(6), 5X, 4HX(7))
38 FORMAT(1HO, 11HCOEFFICIENT, 10X, 4HX(1), 5X, 4HX(2), 5X, 4HX(3), 5X, 4HX(4),
1,5X,4HX(5), 5X, 4HX(6), 5X, 4HX(7), 5X, 4HX(8))
39 FORMAT(1HO, 11HCOEFFICIENT, 10X, 4HX(1), 5X, 4HX(2), 5X, 4HX(3), 5X, 4HX(4),
1,5X,4HX(5), 5X, 4HX(6), 5X, 4HX(7), 5X, 4HX(8), 5X, 4HX(9))
40 FORMAT(1HO, 11HCOEFFICIENT, 10X, 4HX(1), 5X, 4HX(2), 5X, 4HX(3), 5X, 4HX(4),
1,5X,4HX(5), 5X, 4HX(6), 5X, 4HX(7), 5X, 4HX(8), 5X, 4HX(9), 5X, 4HX(10))
END
C RESIDUE SUBROUTINE FORS RESIDUE AND SORTS TERMS

SUBROUTINE RESIDUE(TU, NTT, NV, JPU, JPT, JU, JU)

C**********************************************************************
C**********************************************************************
C
C IMPLICIT INTEGER*2(I-N)
COMMON WMAX(400), IUMAX(3, 400), ITMAX(3, 400), TMAX(400), R(3160),
NIR(4, 3160), ITA*T, ITTIMES, U(3160), T(3160), IT(4, 3160),
NLS?, LS1, LS
DIMENSION I3(5)
MAXR = 3160
MAXP = 430
INTEGER SWITCH
C**********************************************************************
C**********************************************************************
C
I = 1
DO 20 J = 1, NTT
A = T(J)
99999 J = J + 1
5210 IF(K4 = KJ, NV)
IF(ABS(A) .GT. 1.0) 46, 46, 46
46 LS2 = 1
RETURN
45 DO 10 L = 1, JU
K(J) = IUMAX(J)*A
IR(J, J) = P(J)
10 DO 20 KK = 2, NV
10 IR(KK, J) = IUMAX(KK-1, L) + IR(KK)
46 = I - 1
IF(MM .LE. 9) GOTO 10
DO 30 KK = 1, MM
30 IF(IR(KK, KK) .GE. IR(1, 1)) GOTO 10
DO 700 KK = 2, NV
700 CONTINUE
R(KK) = R(KK) + R(1)
IF(AABS(KK)) .GT. 1. E-9) GOTO 702
MM = MM - 1
DO 800 ILK = KK, MM
800 ILP = ILK + 1
800 CONTINUE
800 R(ILK) = IR(NPC, ILK + 1)
I = I + 1
700 = I - 1
GOTO 701
701 IF( , LT. , MAX*) GOTO 19
11 IF(I, ILT. , MAX*) GOTO 19
12 FORMAT( 44, 25HTPS MANY TERMS IN RESIDUE) STOP
14 I = I + 1
20 CONTINUE
DO 40 J = 1, JU
A = U(J)
40 CONTINUE
END
IF ( e^lJ* F JPt ) OT040
IF' A ~ S - Fh) 47,48,403
4g LS?=I
RF TI
P, N
47 Or 41 L = 1, J1
8(1) = -TMAX(L) * A
IF (1,1) = IR(1) + JPT - JPU
DO 30 K = 2,NV
30 IR(K,1) = TMAX(K-1,L) + IR(K)
M4E=1-1
DO 30 KKK=1,NM
IF (IR(1,KK),M4,IR(1,11)) GOTO360
DO 30 KKK=2,NV
IF (IR(KK,KK),M4,IR(KK,KK)) GOTO389
CONTINUE
3709
*(KK) = *(KK)+R(I)
IF (A55(I,KK),GT,1,1f-9)) GOTO360
**(M4)=**(M4)+1
**(KK) = **(KK) +NM
**(KK) = **(KK)+1
3529 20 (IV,1K) = IR(ND0,1L+1)
I=1-1
2702 1=1-1
3703 GOTO3701
3711 CONTINUE
3711 IF (1,1,MAXF) GOTO41
3711 T=1+1
40 CONTINUE
T = T-1
29 J9XR=T=0
109 DO 105 M = 1,1
15 105 (I,2,J) = J9XR M T = 105,105,130
109 J9XR T = 15 (1,4)
109 CONTINUE
JPT = J9XR M T
IF (J9XR M T) 135,142,130
142: LSI = 1
U = 1
37 CONTINUE
19 JT = 0
3711 =1,400
TMAXF=0,0
904.AJ=1,J
7 ITMAXF(J,J,11)=0
25 H065,J=1,1
E: (J9XR M T = IR(I,11)) 56,57,56
52 JT = JT + 1
TMAX(JT) = T(J)
IF (TMAX(JT),56,1,16) LM=1
MIE I = 2, NV
ITMAX(K-1,JT) = IR(K,J)
50 CONTINUE
86 CONTINUE
DOJ=1,N
T(J)=R(J)
DO1 L=1,NV
1 IT(L,J)=IR(L,J)
376 NTT=1
RETURN
END

SUBROUTINE SCALF2(NTFM1, NTERM2, NV)
IMPLICIT *INTEGER*2(I-N)
COMMON UM(X(400), IUMAX(3,400), IMAX(3,400), TMAX(400), R(3160),
NIR(4,3160), ITRAF, ITIMES, U(3160), IU(4,3160), T(3160), IT(4,3160),
NLS2, LSL1, L3)
EQUIVALENCE(NVAR1, MAX)
MAX = NV
NVAR2 = NVAR1
DO 1 I = 1,400
U = UM(I) / 1000.
1 T = TM(I) / 1000.
DO 7 J = 1, NTERM1
7 U(J) = U(J) / 1000.
DO 9 J = 1, NTERM2
9 T(J) = T(J) / 1000.
ITIMES = ITIMES + 1
RETURN
END
APPENDIX B

This appendix presents an example problem. The following system of three polynomials in three variables is reduced to a single polynomial in one variable:

\[(1) \quad x_1 + x_2 + x_3 = 0\]
\[(2) \quad x_1 + x_2^2 = 0\]
\[(3) \quad x_1 + x_3^2 = 0\]

First, \(x_1\) is eliminated between (1) and (2) producing the following printout which has been labeled for expository convenience:

<table>
<thead>
<tr>
<th>COEFFICIENT</th>
<th>(x(1))</th>
<th>(x(2))</th>
<th>(x(3))</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.1000000E 01</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0.1000000E 01</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>0.1000000E 01</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
</tbody>
</table>

Second, \(x_1\) is eliminated between (3) and (2) producing the following printout:

<table>
<thead>
<tr>
<th>COEFFICIENT</th>
<th>(x(1))</th>
<th>(x(2))</th>
<th>(x(3))</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.1000000E 01</td>
<td>0</td>
<td>2</td>
<td>0</td>
</tr>
<tr>
<td>-0.1000000E 01</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>-0.1000000E 01</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
</tbody>
</table>

\(15\)
Finally $(E_1)$ and $(E_2)$ are treated as a pair of polynomials in two variables $x_1$ and $x_2$. Then $x_1$ ($x_2$ in our first system) is eliminated, producing the following printout:

<table>
<thead>
<tr>
<th>COEFFICIENT</th>
<th>x(1)</th>
<th>x(2)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.1000000E01</td>
<td>2</td>
<td>0</td>
</tr>
<tr>
<td>-0.1000000E01</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>-0.1000000E01</td>
<td>0</td>
<td>1</td>
</tr>
</tbody>
</table>

$(E_1)$

<table>
<thead>
<tr>
<th>COEFFICIENT</th>
<th>x(1)</th>
<th>x(2)</th>
</tr>
</thead>
<tbody>
<tr>
<td>-0.1000000E01</td>
<td>2</td>
<td>0</td>
</tr>
<tr>
<td>0.1000000E01</td>
<td>0</td>
<td>2</td>
</tr>
</tbody>
</table>

$(E_2)$

<table>
<thead>
<tr>
<th>COEFFICIENT</th>
<th>x(1)</th>
<th>x(2)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.1000000E01</td>
<td>0</td>
<td>4</td>
</tr>
<tr>
<td>-0.2000000E01</td>
<td>0</td>
<td>3</td>
</tr>
</tbody>
</table>

$(E_3)$ is our eliminant free of $x_1$ and $x_2$, so it can be solved. Using its solution $(E_1)$ can then be solved. Using this $(E_2)$ can be solved and the solutions to the resultant system $(2), (E_2), (E_3)$ are the solutions to the system $(1), (2), (3)$. Three other equations from these six could have been taken to form the resultant system.
POLYNOMIAL MANIPULATION SYSTEM-FORTRAN IV PROGRAM

A FORTRAN IV program which implements the Polynomial Manipulation System (PMS) is presented and described. PMS uses the Euclidean Algorithm to reduce a system of polynomials in several variables to a resultant system which can be solved sequentially as polynomials in one variable (Kronecker's method). PMS is described briefly and references are given to more complete discussions and to other pertinent literature.
<table>
<thead>
<tr>
<th>KEY WORDS</th>
<th>LINE A</th>
<th>LINE B</th>
<th>LINE C</th>
</tr>
</thead>
<tbody>
<tr>
<td>Polynomials</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Resultant</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Eliminant</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Euclid's Algorithm</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>FORTRAN</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Program</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>