



**U.S. ARMY
RDECOM**

TECHNICAL REPORT RDMR-SS-12-05

LEAST SQUARES POLYNOMIAL REGRESSION WITH CONSTRAINTS: A COMPUTER PROGRAM

Clark D. Mikkelsen

**System Simulation and Development Directorate
Aviation and Missile Research, Development,
and Engineering Center**

July 2012

**Distribution Statement A: Approved for public release;
distribution is unlimited.**

 **AMRDEC**



DESTRUCTION NOTICE

FOR CLASSIFIED DOCUMENTS, FOLLOW THE PROCEDURES IN DoD 5200.22-M, INDUSTRIAL SECURITY MANUAL, SECTION II-19 OR DoD 5200.1-R, INFORMATION SECURITY PROGRAM REGULATION, CHAPTER IX. FOR UNCLASSIFIED, LIMITED DOCUMENTS, DESTROY BY ANY METHOD THAT WILL PREVENT DISCLOSURE OF CONTENTS OR RECONSTRUCTION OF THE DOCUMENT.

DISCLAIMER

THE FINDINGS IN THIS REPORT ARE NOT TO BE CONSTRUED AS AN OFFICIAL DEPARTMENT OF THE ARMY POSITION UNLESS SO DESIGNATED BY OTHER AUTHORIZED DOCUMENTS.

TRADE NAMES

USE OF TRADE NAMES OR MANUFACTURERS IN THIS REPORT DOES NOT CONSTITUTE AN OFFICIAL ENDORSEMENT OR APPROVAL OF THE USE OF SUCH COMMERCIAL HARDWARE OR SOFTWARE.

REPORT DOCUMENTATION PAGE			Form Approved OMB No. 074-0188	
Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing this collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188), Washington, DC 20503				
1. AGENCY USE ONLY		2. REPORT DATE July 2012	3. REPORT TYPE AND DATES COVERED Final	
4. TITLE AND SUBTITLE Least Squares Polynomial Regression with Constraints: A Computer Program			5. FUNDING NUMBERS	
6. AUTHOR(S) Clark D. Mikkelsen				
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Commander, U.S. Army Research, Development, and Engineering Command ATTN: RDMR-SSM-A Redstone Arsenal, AL 35898-5000			8. PERFORMING ORGANIZATION REPORT NUMBER TR-RDMR-SS-12-05	
9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)			10. SPONSORING / MONITORING AGENCY REPORT NUMBER	
11. SUPPLEMENTARY NOTES				
12a. DISTRIBUTION / AVAILABILITY STATEMENT Approved for public release; distribution is unlimited.			12b. DISTRIBUTION CODE A	
13. ABSTRACT (Maximum 200 Words) The least squares polynomial regression constraints program LSPRWC presented in this report is a computer code written to perform a least squares polynomial regression on a given set of data pairs or observations with the option to impose constraints on the regression polynomial. The FORTRAN 77 computer program listing, as well as detailed information on theory, program structure, and limitations, are given along with two sample cases.				
14. SUBJECT TERMS least squares, polynomial regression, constraints, curve fit, FORTRAN 77			15. NUMBER OF PAGES 110	
			16. PRICE CODE	
17. SECURITY CLASSIFICATION OF REPORT UNCLASSIFIED	18. SECURITY CLASSIFICATION OF THIS PAGE UNCLASSIFIED	19. SECURITY CLASSIFICATION OF ABSTRACT UNCLASSIFIED	20. LIMITATION OF ABSTRACT SAR	

NSN 7540-01-280-5500

Standard Form 298 (Rev. 2-89)
Prescribed by ANSI Std. Z39-18
298-102

TABLE OF CONTENTS

	<u>Page</u>
I. INTRODUCTION	1
II. FEATURES.....	2
III. SPECIAL FEATURES.....	3
IV. PROGRAM STRUCTURE	5
V. LIMITATIONS.....	6
VI. ERROR MESSAGES.....	8
REFERENCES	9
APPENDIX A: DERIVATION	A-1
APPENDIX B: FORTRAN 77 CODE	B-1
APPENDIX C: FORTRAN TOOL LIBRARY.....	C-1
APPENDIX D: EXAMPLE 1	D-1
APPENDIX E: EXAMPLE 2.....	E-1

LIST OF ILLUSTRATIONS

<u>Figure</u>	<u>Title</u>	<u>Page</u>
1.	Flow Path for Program LSPRWC	6

I. INTRODUCTION

LSPRWC is a FORTRAN 77 computer program written to perform a least squares polynomial regression on a given set of (x, y) data pairs or observations. In common parlance, LSPRWC, a “curve fit” procedure, finds that n_p order polynomial

$$P(x) = \sum_{j=0}^{n_p} b_j x^j$$

which minimizes the sum of the squared errors, the squared residuals, between the polynomial value $P(x)$ and $y(x)$ over the set of (x, y) data pairs.

In addition, LSPRWC provides for the imposition of constraints on $P(x)$ such that the n_c derivative of $P(x)$ may be specified at a given x location.

LSPRWC was written for LINUX¹/UNIX² users as an analysis tool in science and engineering. The program was intended for users who care about numerical precision, wish to have a well documented source code in hand, and abhor spreadsheet calculations and license fees.

LSPRWC requires command line interaction. No graphical user interfaces are provided with the computer program—although they could certainly be added. The code was not meant for the “point and click” community of computer users.

Least squares polynomial regression is a well known analytical technique dating back to at least the 18th century³. The technique remains a premier method for the functional approximation of presumably repeatable phenomena when observations of that phenomena are known to be imperfect. The technique is also commonly used to quantify those phenomena which either defy description by the known laws of physics or for which the mathematical descriptions are simply too unwieldy or numerically intensive for practical application.

Although the functional form for linear regression need not be based on polynomials, polynomials have proven to be particularly useful for a broad range of applications. Additionally, the closed-form mathematical solution for $P(x)$ is readily derived, as given in Appendix A. Once the b_j coefficients are known, polynomial evaluation reduces to a simple numerical process. Furthermore, it is relatively easy to compute derivatives and other properties, such as arc length and integrals for polynomials.

LSPRWC was specifically written for source code control, source code that is readily tailored for specific applications, and to maintain ease of use with the advent of new computational platforms and operating systems. As such, the code eliminates those inexplicable fee requirements for licensed software which exploits an analytical technique embodied in the public domain for centuries.

The addition of constraints was added to LSPRWC as a particularly useful capability not usually included with commercial least squares regression software packages. Often as not,

some physical constraint—an initial condition such as zero velocity and a second derivative such as gravitational acceleration—are known absolutely and can be included with LSPWRC to better approximate a given phenomena.

LSPWRC, as written for LINUX/UNIX operating systems, requires only a FORTRAN 77⁴ compiler and a terminal window system such as the *X Window System*⁵. The code is known to compile successfully with the Free Software Foundation G77⁶ compiler and to work well on *Apple Mac OS X*⁷ and *Yellow Dog*⁸ LINUX computer operating systems. Although written to FORTRAN 77 standards, the code may compile with FORTRAN 95⁹ compilers and does so with the Intel ifort¹⁰ compiler. LSPWRC memory and disk storage requirements are flexible and readily adjusted.

The LSPWRC computer code is not guaranteed to run successfully or operate without failure on any given computer and/or operating system. Considerable error messaging is included in the FORTRAN 77 code, but this does not assure that the linear regression process operates as intended. Hence, it is advised that the user first execute test cases with known solutions for comparison with the LSPWRC code output.

II. FEATURES

The LSPWRC code was written to provide a user friendly environment while offering extensive analytical capabilities by means of the following features:

- Full directives are written to the terminal window whenever input to the LSPWRC code is requested.
- All operational parameters are input using either NAMELIST or free field reads.
- Input to the LSPWRC code from formatted disk files may be offered as an option. In such cases, the data file format, including free field reads, will be input from the command line.
- Sets of (x, y) data pairs are entered either by NAMELIST or read from a formatted disk file.
- Optional constraints are input either by NAMELIST or read from a formatted disk file. Each constraint is defined by the derivative order $n_c = 0$ through $n_p - 1$, the x location for the constraint, and the value of the n_c derivative of $P(x)$ at the constraint location.
- Computational results from the linear least squares regression are output to the terminal window as the LSPWRC code executes. This output includes the values of the b_j coefficients of the n_p order polynomial $P(x)$ and the standard deviation of the (x, y) data pairs with respect to the regression polynomial.
- As an option, x values may be prescribed for evaluation of the resultant least squares polynomial. These x values may be entered by NAMELIST, input from a

formatted disk file, or alternatively as minimum, maximum, and interval values to produce an array of evaluated $(x, P(x))$ data pairs.

- A plot option is included in the LSPRWC code such that the user may visually compare the evaluated $(x, P(x))$ data with the (x, y) data pairs. No plot package is actually included with the LSPRWC code, but the two data sets, $(x, P(x))$ and (x, y) , are written to formatted data files for ready access to the user's favorite plot package. The plot option subroutine QPS can also easily be modified by the user to include an in-line plot routine for execution with the LSPRWC code.
- Output of results from a LSPRWC run may optionally be written to a formatted disk file as either the entire set of run parameters, input data, and results or as simply the evaluated data. In the latter case, the evaluated data includes the choices of x , the 0 through $n_p - 1$ derivative value of $P(x)$, the radius of curvature, and the arc length as columnated data sets.
- Restart options are included in the LSPRWC code allowing for easy modification of operational parameters, for example polynomial order n_p , without having to re-enter all of the input information.
- Constraints are satisfied using the method of undetermined Lagrange Multipliers¹¹.
- The LSPRWC code includes a matrix inversion subroutine GJEMPS which is based on Gauss-Jordan elimination using a maximum pivot strategy¹². With some minimal effort, the user may substitute their preferred matrix inversion technique should that be desired.
- All of the FORTRAN 77 array sizes are set in PARAMETER statements within the LSPRWC code making it relatively easy to find and change those array size limits should that be required.
- The LSPRWC code maintains 64-bit precision with IMPLICIT REAL*8 statements in each subroutine as required.

III. SPECIAL FEATURES

The following special features were added to the LSPRWC code to enhance the user friendly environment:

- If a simple Yes/No input is requested, then any upper/lower case variation of Yes, No, Y, or N provides a permissible entry as controlled by the FORTRAN subroutine YNOUS listed in Appendix C.

- Disk files to be used for LSPRWC code input may contain alphanumeric headers such as

Thermal Data Set

T(K)	P(Pa)
289.261111111111	0.18305580366000E+04
289.816666666667	0.18967477057200E+04
290.372222222222	0.19650058020000E+04
290.927777777778	0.20346428497200E+04
291.483333333333	0.21070378003200E+04
292.038888888889	0.21815011780800E+04
292.594444444444	0.22587224587200E+04
293.150000000000	0.23373226908000E+04

The disk file data will be read, line by line, and ignored until the prescribed data format is satisfied.

- Command line NAMELIST input can be a nuisance. NAMELIST entries begin in column 2 with either a \$ or & sign followed immediately with the NAMELIST name. The entries continue with the name of the variables to be changed from their current values, an equal sign, and the new value to be assigned to the variable. The NAMELIST entry must then end with either a \$ or & sign. The FORTRAN subroutines NDRUS and NDWUS, as given in Appendix C, were included in the LSPRWC code to simplify this process.

If a NAMELIST input is requested, then all variables are written to the terminal window with default values to facilitate cut and paste input using a three-button mouse.

The starting column for NAMELIST entries is irrelevant.

NAMELIST entries need not begin with the \$ or & sign nor is the NAMELIST name required.

The NAMELIST entry must end with either the \$ or & signs but only if variable values are altered. A simple keyboard return will accomplish a NAMELIST entry with no changes to the variable default values.

- Whenever FORTRAN statements include an option for redirection following an error, e.g.

READ(UNIT=4,NML=PARM,ERR=169),

and the source of the error is trivial, for example a misspelled NAMELIST variable name, then provision is included in the LSPRWC code for correction of the error without the need to restart.

IV. PROGRAM STRUCTURE

The structure of the LSPRWC FORTRAN 77 computer code as listed in Appendix B is reasonably straight forward. The program routines and their purposes are as follows:

MAIN—Executive control for program LSPRWC

XYDIS—Subroutine for input of the (x, y) data set or observations

CDS—Subroutine for input of the constraints

PODS—Subroutine for input of the polynomial order

LSPRWC—Subroutine to perform the least squares polynomial regression with constraints

XDPEDES—Subroutine to provide an x -coordinate set for evaluation of the least squares regression polynomial

PES—Polynomial evaluation subroutine

QPS—Quick plot subroutine

FDFOS—Formatted disk file output subroutine

GJEMPS—Matrix inversion subroutine using Gauss-Jordan elimination with maximum pivot strategy

POWER—Subroutine to calculate integer powers of real numbers

FACTRL—Factorial evaluation subroutine

As can be seen, the actual least squares polynomial regression algorithm resides in subroutine LSPRWC. Subroutines CDS, PODS, XDPEDES, and XYDIS provide or define input information for program LSPRWC. Subroutine PES provides output data for program LSPRWC. Subroutines FDFOS and QPS output information from program LSPRWC. Subroutines FACTRL, GJEMPS, and POWER are computational utility routines. A flow path for program LSPRWC is given in Figure 1.

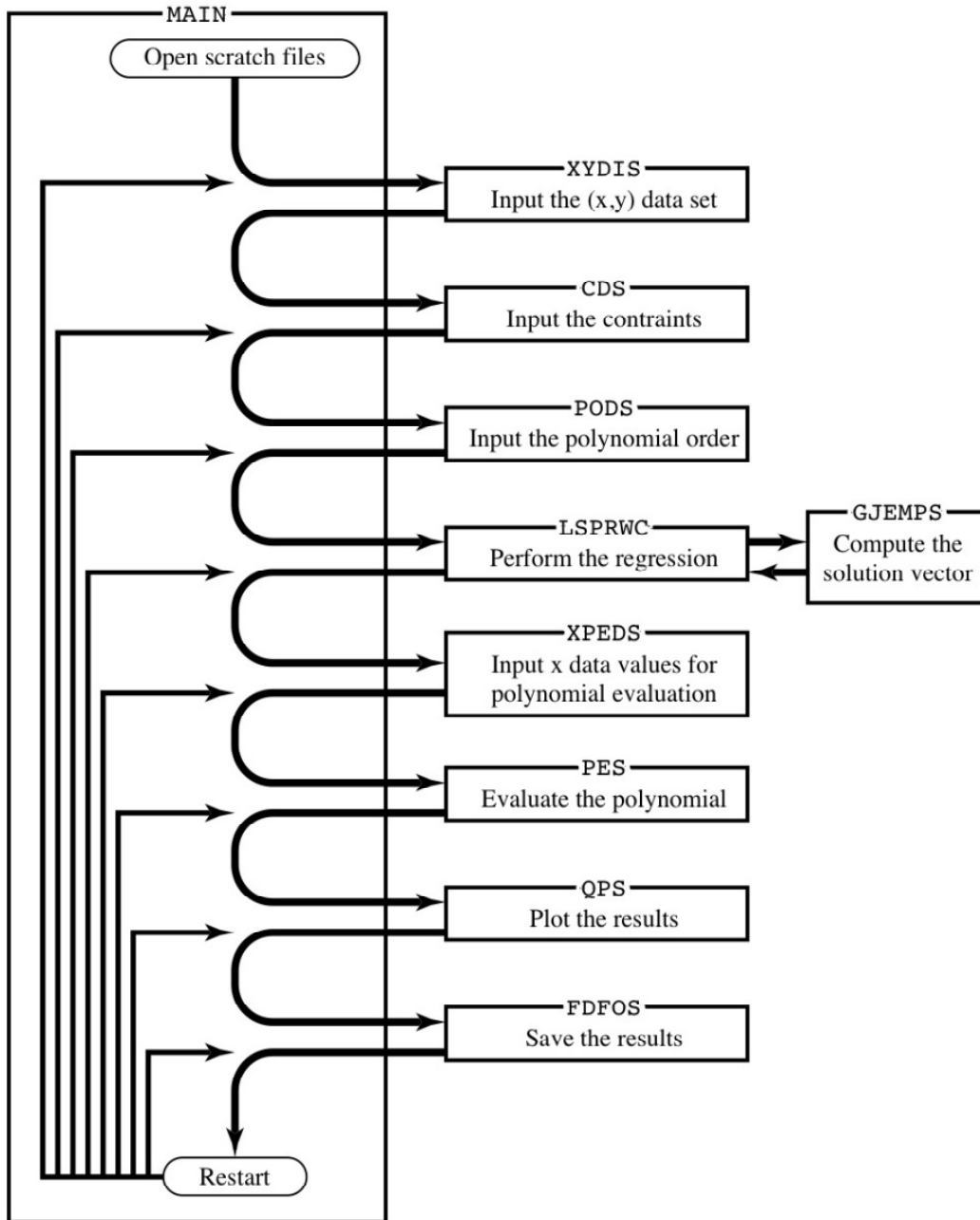


Figure 1. Flow Path for Program LSPRWC

Additional low-level utility routines are listed in the FORTRAN Tool Library in Appendix C.

V. LIMITATIONS

The LSPRWC code has the following requirements or limitations:

- The LSPRWC code was written with the following set of COMMON and PARAMETER statements which set the array sizes consistently in all subroutines.

MXXYDP = Maximum Number of (x, y) Data Pairs

$$= 101$$

MNXCE = Maximum Number of x -Coordinates for polynomial evaluation

$$= 101$$

MXANC = Maximum Number of Allowable Constraints

$$= 20$$

MXNR = Maximum Number of Rows

$$= 30$$

MXNRM1 = Maximum Number of Rows minus 1

$$= 29$$

MXNC = Maximum Number of Columns

$$= 31$$

MXNCM1 = Maximum Number of Columns minus 1

$$= 30$$

The number of rows and columns refers to the size of the augmented matrix for the solution of n linear equations in n unknowns. It follows from the derivation of Appendix A that

$$\text{MXNR} \geq n_p + \xi + 1, \quad (1)$$

$$\text{MXNRM1} = \text{MXNR} - 1, \quad (2)$$

$$\text{MXNC} = \text{MXNR} + 1, \text{ and} \quad (3)$$

$$\text{MXNCM1} = \text{MXNC} - 1. \quad (4)$$

The **PARAMETER** variables **MXANC** and **MXNR** apply only for **NAMelist** input. Any number of (x, y) data pairs or x -coordinates for polynomial evaluation can be read from disk files.

- The set of (x, y) data pairs, as read into program **LSPRWC**, must be monotonically increasing in x . Provision could have been made within program **LSPRWC** to reorder any set of (x, y) data pairs; however, it has been discovered that the failure of a data set to satisfy this requirement is often the indication of unknown errors. Hence, no data set reordering has been implemented.
- The number of (x, y) data pairs or observations, m , must satisfy

$$m \geq n_p + \zeta + 1 . \quad (5)$$

- The polynomial order n_p must satisfy

$$0 \leq n_p \leq 9 . \quad (6)$$

The LSPRWC code could, though not readily, be modified for polynomial orders greater than 9; however, this largely defeats the purpose intended for polynomial regression.

- The constraint order n_c must satisfy

$$0 \leq n_c < n_p . \quad (7)$$

- Polynomials $P(x)$ will not produce an infinite slope. Hence, a transformation—possibly a rotation—may be required for a given set of (x, y) data pairs before a least squares polynomial regression can be completed to eliminate that possibility.

VI. ERROR MESSAGES

The LSPRWC code was written with the following extensive embedded error messaging:

- The error messages cover the standard FORTRAN errors and those associated directly with the operation of program LSPRWC.
- Wherever possible, the LSPRWC code identifies the source of the particular error and the subroutine call structure leading to that error.
- Should readily correctable errors occur, for example a misspelled NAMELIST variable name entry, then program LSPRWC will identify the error and loop, in this case for the NAMELIST entry.
- The standard FORTRAN errors source from the FORTRAN OPEN, CLOSE, READ, and WRITE commands.
- Program LSPRWC will issue an error message should the array limits be exceeded. (See Section V.)
- Program LSPRWC will issue an error message should an illegal value for n_p be entered. (See Section V.)
- Program LSPRWC will issue an error message should an illegal value for n_c be entered. (See Section V.)
- Program LSPRWC will issue an error message should the number of (x, y) data pairs be insufficient for the number of constraints, ζ , and the polynomial order, n_p . (See Section V.)

REFERENCES

1. Bovet, D. and Cesati, M., Understanding the LINUX Kernel, 3rd Edition, ISBN-10: 0-596-00565-2, O'Reilly Media, Inc., 2005.
2. Rosen, K. et al., UNIX: The Complete Reference, 2nd Edition, ISBN-10: 0-07-226336-9, MCGRAW-HILL, 2007.
3. Wikipedia, "Least squares," http://en.wikipedia.org/wiki/Least_squares/
4. FORTRAN 77 Language Reference Manual, Document Number 007-0710-030, Silicon Graphics, Inc., 1990.
5. Young, D.A., The X Window System: Programming and Applications with Xt, OSF/Motif, 2nd Edition, ISBN 0-13-123803-5, Prentice-Hall, 1994.
6. <http://gcc.gnu.org/fortran/>
7. Wikipedia, "Mac OS X," http://en.wikipedia.org/wiki/Mac_OS_X
8. Wikipedia, "Yellow Dog Linux," http://en.wikipedia.org/wiki/Yellow_Dog_Linux
9. Adams, J.C. et al., FORTRAN 95 HANDBOOK, ISBN 0-262-51096-0, The MIT Press, 1997.
10. "Intel Fortran Compiler for Linux," <http://software.intel.com/en-us/articles/intel-fortrancompiler-for-linux-9x-manuals/>
11. Weinstock, R., Calculus of Variations with Applications to Physics and Engineering, New York, Dover Publications, Inc., pp. 6, 1974.
12. Carnahan, B.; Luther, H.A.; and Wilkes, J.L., Applied Numerical Methods, New York, John Wiley and Sons, pp. 281-96, 1969.

APPENDIX A DERIVATION

A set of (x,y) data pairs are to be “curve fit” using the method of least squares; that is, an n_p order polynomial is chosen such that the sum of the squared errors is minimized over the set of (x,y) data pairs. In addition, the n_p order polynomial must satisfy a set of constraints; namely, that the n_c polynomial derivative be specified at a given x location.

Then with the n_p order polynomial given by

$$P(x) = \sum_{j=0}^{n_p} b_j x^j, \quad (1)$$

the squared error to be minimized is

$$S = \sum_{i=1}^m [y_i - P(x_i)]^2, \quad (2)$$

$$= \sum_{i=1}^m (y_i - \sum_{j=0}^{n_p} b_j x_i^j)^2, \quad (3)$$

subject to the constraints

$$G_l(x_l) = P^{k_l}(x_l) = C_l \quad (4)$$

$$= \sum_{j=k_l}^{n_p} \frac{j!}{(j-k_l)!} b_j x_l^{j-k} = C_l, \quad (5)$$

for $l = 1, 2, 3, \dots, \xi$

where

n_p = polynomial order,
 ξ = number of constraints,
 m = number of (x,y) data pairs,
 x_l = l_{th} constraint location,
 k_l = l_{th} constraint order ($0 \leq k_l \leq n_p$),
 C_l = value of the l_{th} constraint.

Using the method of undetermined Lagrange multipliers, the relation to be minimized is

$$S^* = S + \sum_{l=1}^{\xi} \lambda_l G_l \quad (6)$$

by setting

$$\frac{\partial S^*}{\partial b_j} = 0 \quad \text{for } j = 0, 1, \dots, n_p. \quad (7)$$

From equations (3), (5), and (6),

$$S^* = \sum_{i=1}^m (y_i - \sum_{j=0}^{n_p} b_j x_i^j)^2 + \sum_{l=1}^{\xi} \lambda_l \sum_{j=k_l}^{n_p} \frac{j!}{(j-k_l)!} b_j x_l^{j-k_l}. \quad (8)$$

Then using equation (7),

$$\frac{\partial S^*}{\partial b_J} = \sum_{i=1}^m (-2x_i^J)(y_i - \sum_{j=0}^{n_p} b_j x_i^j) + \sum_{l=1}^{\xi} \lambda_l F_{l,J} = 0 \quad (9)$$

where

$$F_{l,J} = \begin{cases} \frac{J!}{(J-k_l)!} x_l^{J-k_l} & , J \geq k_l \\ 0 & , J < k_l \end{cases}. \quad (10)$$

Rewriting equation (9) gives

$$\sum_{i=1}^m \sum_{j=0}^{n_p} b_j x_i^{J-j} + \frac{1}{2} \sum_{l=1}^{\xi} \lambda_l F_{l,J} = \sum_{i=1}^m y_i x_i^J \quad (11)$$

$$\text{for } J = 0, 1, \dots, n_p.$$

The minimization equation (11) provides $n_p + 1$ equations in the $n_p + 1$ unknown coefficients b_j and ξ unknown Lagrange multipliers λ_l . Constraint equation (5) provides the remaining ξ equations in the unknown b_j . Since equations (5) and (11) are linear and algebraic, they may be solved simultaneously using a matrix inversion technique.

APPENDIX B
FORTRAN 77 CODE

```

C*****00000001
C*00000002
C*   LEAST SQUARES POLYNOMIAL REGRESSION WITH CONSTRAINTS (LSPRWC)00000003
C*00000004
C*           WRITTEN BY: C.D. MIKKELSEN00000005
C*00000006
C*           2 MAY 198800000007
C*00000008
C*           AERODYNAMICS TECHNOLOGY BRANCH (AMSMI-RD-SS-AT)00000009
C*           SYSTEMS SIMULATION AND DEVELOPMENT DIRECTORATE00000010
C*   US ARMY MISSILE RESEARCH, DEVELOPMENT, AND ENGINEERING CENTER00000011
C*           US ARMY MISSILE COMMAND00000012
C*           REDSTONE ARSENAL, ALABAMA 35898-525200000013
C*00000014
C* REF: CARNAHAN,B., LUTHER, H.A., AND WILKES, J.O.: APPLIED NUMERICAL00000015
C*   METHODS, NEW YORK, JOHN WILEY & SONS, 1969, PP. 571-584.00000016
C*00000017
C*   WEINSTOCK, R.: CALCULUS OF VARIATIONS WITH APPLICATIONS TO00000018
C*   PHYSICS AND ENGINEERING, NEW YORK, DOVER PUBLICATIONS, INC.,00000019
C*   1974, P. 6.00000020
C*00000021
C*           REVISION DATE: 7 JULY 201100000022
C*00000023
C*****00000024
C00000025
C*****00000026
C*00000027
C*   PROGRAM LSPRWC REQUIRES THE FOLLOWING SUBPROGRAMS:00000028
C*00000029
C*   CBUS06 CDS DFCUS FACTRL FDFOS GJEMPS LSPRWC00000030
C*   PES PODS POWER QPS XDPEDS XYDIS YNOUS00000031
C*00000032
C*****00000033
C00000034
C*****00000035
C*00000036
C* PARAMETERS:00000037
C*00000038
C* MXNRM1 = MAXIMUM NUMBER OF ROWS MINUS 1 (MXNRM1 = MXNR - 1)00000039
C*00000040
C* LOGICAL UNIT DEFINITIONS:00000041
C*00000042
C* UNIT FILE00000043
C*00000044
C*   1 INPUT (X,Y) DATA PAIR SCRATCH FILE00000045
C*   2 (XC,NC,CV) CONSTRAINT SCRATCH FILE00000046
C*   3 EVALUATED DATA SCRATCH FILE00000047
C*   4 TEMPORARY USE00000048
C*   5 STANDARD INPUT00000049
C*   6 STANDARD OUTPUT00000050
C*00000051
C*****00000052
C00000053
C00000054
C   PROGRAM MAIN00000055
C00000056
C   IMPLICIT REAL*8(A-H,O-Z)00000057
C00000058
C   CHARACTER CCV*100000059
C00000060

```

INTEGER DUM2	00000061
C	00000062
PARAMETER (MXNRM1=29)	00000063
C	00000064
COMMON/B/B(0:MXNRM1)	00000065
COMMON/NCONST/NCONST	00000066
COMMON/NEVAL/NEVAL	00000067
COMMON/NP/NP	00000068
COMMON/SDEV/SDEV	00000069
C	00000070
C	00000071
C*****	00000072
C*	*00000073
C* OPEN THE SCRATCH FILES	*00000074
C*	*00000075
C*****	00000076
C	00000077
C	00000078
OPEN(UNIT=1,STATUS='SCRATCH',FORM='UNFORMATTED',ERR=116)	00000079
OPEN(UNIT=2,STATUS='SCRATCH',FORM='UNFORMATTED',ERR=117)	00000080
OPEN(UNIT=3,STATUS='SCRATCH',FORM='UNFORMATTED',ERR=118)	00000081
C	00000082
C	00000083
C*****	00000084
C*	*00000085
C* WRITE THE PROGRAM DESCRIPTION	*00000086
C*	*00000087
C*****	00000088
C	00000089
C	00000090
WRITE(UNIT=6,FMT=201)	00000091
WRITE(UNIT=6,FMT=202)	00000092
READ(UNIT=5,FMT=203)CCV	00000093
C	00000094
C	00000095
C*****	00000096
C*	*00000097
C* INPUT THE X-Y DATA POINTS TO BE FITTED	*00000098
C*	*00000099
C*****	00000100
C	00000101
C	00000102
101 CALL XYDIS(*119,*115)	00000103
C	00000104
C	00000105
C*****	00000106
C*	*00000107
C* INPUT THE CONSTRAINTS	*00000108
C*	*00000109
C*****	00000110
C	00000111
C	00000112
102 NCONST=0	00000113
103 WRITE(UNIT=6,FMT=208)	00000114
CALL YNOUS(*104,*105,*103)	00000115
104 CALL CDS(*120,*115)	00000116
C	00000117
C	00000118
C*****	00000119
C*	*00000120

C*	INPUT THE POLYNOMIAL ORDER	*00000121
C*		*00000122
C*****		*00000123
C		00000124
C		00000125
105	CALL PODS(*115)	00000126
C		00000127
C		00000128
C*****		*00000129
C*		*00000130
C*	PERFORM THE LEAST SQUARES POLYNOMIAL REGRESSION WITH CONSTRAINTS	*00000131
C*		*00000132
C*****		*00000133
C		00000134
C		00000135
	CALL LSPRWC(*121)	00000136
	WRITE(UNIT=6,FMT=209)	00000137
	WRITE(UNIT=6,FMT=210)(I,B(I),I=0,NP)	00000138
	WRITE(UNIT=6,FMT=211)NP,SDEV	00000139
	WRITE(UNIT=6,FMT=202)	00000140
	READ(UNIT=5,FMT=203)CCV	00000141
C		00000142
C		00000143
C*****		*00000144
C*		*00000145
C*	INPUT THE X DATA POINTS FOR EVALUATION	*00000146
C*	OF THE LEAST SQUARES POLYNOMIAL	*00000147
C*		*00000148
C*****		*00000149
C		00000150
C		00000151
106	WRITE(UNIT=6,FMT=212)	00000152
	LCV2=0	00000153
	CALL YNOUS(*107,*112,*106)	00000154
107	CALL XDPEDS(LCV2,*122,*115)	00000155
C		00000156
C		00000157
C*****		*00000158
C*		*00000159
C*	EVALUATE THE LEAST SQUARES POLYNOMIAL AT THE PRESCRIBED X VALUES	*00000160
C*		*00000161
C*****		*00000162
C		00000163
C		00000164
108	CALL PES(*123)	00000165
109	WRITE(UNIT=6,FMT=205)	00000166
	CALL YNOUS(*110,*112,*109)	00000167
110	REWIND 3	00000168
	WRITE(UNIT=6,FMT=206)	00000169
	DO 111 I=1,NEVAL	00000170
	READ(UNIT=3,ERR=124)X,Y	00000171
	IF(MOD(I,20).NE.0) GO TO 111	00000172
	WRITE(UNIT=6,FMT=202)	00000173
	READ(UNIT=5,FMT=203)CCV	00000174
	WRITE(UNIT=6,FMT=206)	00000175
111	WRITE(UNIT=6,FMT=207)I,X,Y	00000176
	WRITE(UNIT=6,FMT=202)	00000177
	READ(UNIT=5,FMT=203)CCV	00000178
C		00000179
C		00000180

```

C*****00000181
C*00000182
C*PLOT THE RESULTS00000183
C*00000184
C*****00000185
C00000186
C00000187
112 WRITE(UNIT=6,FMT=213)00000188
CALL YNOUS(*113,*115,*112)00000189
113 CALL QPS(*125)00000190
GO TO 11500000191
C00000192
C00000193
C*****00000194
C*00000195
C*SAVE RESULTS ON FORMATTED DISK FILE00000196
C*00000197
C*****00000198
C00000199
C00000200
114 CALL FDFOS(LCV2,*126)00000201
C00000202
C00000203
C*****00000204
C*00000205
C*RESTART PROCEDURE00000206
C*00000207
C*****00000208
C00000209
C00000210
115 WRITE(UNIT=6,FMT=214)00000211
READ(UNIT=5,FMT=204)LCV100000212
GO TO (101,104,105,107,113,114),LCV100000213
STOP00000214
C00000215
C00000216
C*****00000217
C*00000218
C*ERROR MESSAGES00000219
C*00000220
C*****00000221
C00000222
C00000223
116 WRITE(UNIT=6,FMT=215)00000224
STOP00000225
117 WRITE(UNIT=6,FMT=216)00000226
STOP00000227
118 WRITE(UNIT=6,FMT=217)00000228
STOP00000229
119 WRITE(UNIT=6,FMT=218)00000230
STOP00000231
120 WRITE(UNIT=6,FMT=219)00000232
STOP00000233
121 WRITE(UNIT=6,FMT=220)00000234
GO TO 12800000235
122 WRITE(UNIT=6,FMT=221)00000236
STOP00000237
123 WRITE(UNIT=6,FMT=222)00000238
STOP00000239
124 WRITE(UNIT=6,FMT=223)00000240

```

```

      STOP                                00000241
125  WRITE(UNIT=6,FMT=224)                00000242
      STOP                                00000243
126  WRITE(UNIT=6,FMT=225)                00000244
      STOP                                00000245
127  WRITE(UNIT=6,FMT=226)                00000246
      STOP                                00000247
128  WRITE(UNIT=6,FMT=202)                00000248
      READ(UNIT=5,FMT=203)CCV             00000249
      GO TO 115                           00000250
C                                          00000251
C                                          00000252
C*****00000253
C*                                         00000254
C*                                         00000255
C*                                         00000256
C*****00000257
C                                          00000258
C                                          00000259
201  FORMAT(/,T4,'PROGRAM LSPRWC IS AN INTERACTIVE FORTRAN PROGRAM TO P00000260
>ERFORM A LEAST',/, 'SQUARES POLYNOMIAL REGRESSION WITH CONSTRAINTS; 00000261
>THAT IS, A SET OF X-Y DATA',/, 'POINTS IS CURVE FIT WITH AN NP ORDE00000262
>R POLYNOMIAL OF THE FORM',/,T17,'P(X)=B0+B1*X+B2*X**2+B3*X**3+...00000263
>.+BNP*X**NP',/, 'WITH ANY POLYNOMIAL DERIVATIVES, ZERO THROUGH NP,00000264
> SPECIFIED AT GIVEN X',/, 'LOCATIONS. THE PROCEDURE FOLLOWED IS TH00000265
>E METHOD OF LEAST SQUARES USING',/, 'UNDETERMINED LAGRANGE MULTIPLI00000266
>ERS.',/,T4,'AS AN INTERACTIVE PROGRAM, LSPRWC IS SELF-EXPLANATORY00000267
> AND PROMPTS FOR THE',/, 'NECESSARY INFORMATION. THE X-Y DATA TO B00000268
>E FITTED MAY BE ENTERED BY NAMELIST',/, 'OR READ FROM A FORMATTED D00000269
>ISC FILE. PROGRAM LSPRWC WILL ALSO EVALUATE THE',/, 'RESULTANT LEA00000270
>ST SQUARES POLYNOMIAL AT PRESCRIBED VALUES OF X WHICH, AGAIN',/, '00000271
>MAY BE ENTERED BY NAMELIST OR READ FROM A FORMATTED DISK FILE.') 00000272
202  FORMAT(/,T19,'- ENTER/RETURN TO CONTINUE -') 00000273
203  FORMAT(A1)                                00000274
204  FORMAT(I1)                                00000275
205  FORMAT(/, 'SHOULD THE X-Y DATA BE DISPLAYED FOR VERIFICATION? (Y/N)00000276
>') 00000277
206  FORMAT(/,T10,'NO.',T26,'X',T49,'Y') 00000278
207  FORMAT(T8,I5,1P2D23.13) 00000279
208  FORMAT(/, 'ARE CONSTRAINTS DESIRED? (Y/N)') 00000280
209  FORMAT(T7,'LEAST SQUARES POLYNOMIAL',/,T19,'P(X)=B(0)+B(1)*X+B(2)00000281
>*X**2+....+B(NP)*X**NP',/,T22,'I',T36,'B(I)',/) 00000282
210  FORMAT(' ',T22,I1,1PD25.13) 00000283
211  FORMAT(/,T4,'THE STANDARD DEVIATION FOR THIS POLYNOMIAL OF ORDER 00000284
> ',I1,' IS ',1PD12.5) 00000285
212  FORMAT(/, 'SHOULD DATA POINTS BE PRESCRIBED FOR EVALUATION OF THE L00000286
>EAST SQUARES',/, 'POLYNOMIALS? (Y/N)') 00000287
213  FORMAT(/, 'SHOULD THE RESULTS OF THIS RUN BE QUICK-PLOTTED? (Y/N)')00000288
> 00000289
214  FORMAT(/, 'ENTER:',/, '1, TO RESTART THE PROGRAM',/, '2, TO CHANGE TH00000290
>E CONSTRAINTS',/, '3, TO CHANGE THE POLYNOMIAL ORDER',/, '4, TO CHAN00000291
>GE THE PRESCRIBED X VALUES FOR EVALUATION OF THE POLYNOMIAL',/, '5,00000292
> TO PLOT THE RESULTS',/, '6, TO SAVE THE RESULTS ON A FORMATTED DIS00000293
>C FILE',/, '7, TO STOP') 00000294
215  FORMAT(/, 'ERROR IN PROGRAM LSPRWC: OPEN ERROR ON UNIT 1, STATUS = 00000295
>"SCRATCH"') 00000296
216  FORMAT(/, 'ERROR IN PROGRAM LSPRWC: OPEN ERROR ON UNIT 2, STATUS = 00000297
>"SCRATCH"') 00000298
217  FORMAT(/, 'ERROR IN PROGRAM LSPRWC: OPEN ERROR ON UNIT 3, STATUS = 00000299
>"SCRATCH"') 00000300

```

218	FORMAT('SUBROUTINE XYDIS WAS CALLED FROM PROGRAM LSPRWC')	00000301
219	FORMAT('SUBROUTINE CDS WAS CALLED FROM PROGRAM LSPRWC')	00000302
220	FORMAT('SUBROUTINE LSPRWC WAS CALLED FROM PROGRAM LSPRWC')	00000303
221	FORMAT('SUBROUTINE XDPEDS WAS CALLED FROM PROGRAM LSPRWC')	00000304
222	FORMAT('SUBROUTINE PES WAS CALLED FROM PROGRAM LSPRWC')	00000305
223	FORMAT(/,'ERROR IN PROGRAM LSPRWC: UNFORMATTED READ ERROR ON\$UNIT	00000306
	>3')	00000307
224	FORMAT('SUBROUTINE QPS WAS CALLED FROM PROGRAM LSPRWC')	00000308
225	FORMAT('SUBROUTINE FDFOS WAS CALLED FROM PROGRAM LSPRWC')	00000309
226	FORMAT('SUBROUTINE CBUS06 WAS CALLED FROM PROGRAM LSPRWC')	00000310
	END	00000311

```

SUBROUTINE CDS(*,*)                                00000001
C                                                    00000002
C                                                    00000003
C*****00000004
C*                                                    *00000005
C*          CONSTRAINT DEFINITION SUBROUTINE (CDS)    *00000006
C*                                                    *00000007
C*          REVISION DATE: 1 JULY 2011                *00000008
C*                                                    *00000009
C*****00000010
C                                                    00000011
C*****00000012
C*                                                    *00000013
C*          SUBROUTINE CDS DEFINES CONSTRAINTS FOR PROGRAM LSPRWC. *00000014
C*                                                    *00000015
C* INPUT/OUTPUT VARIABLES:                            *00000016
C*                                                    *00000017
C* CV      = VALUE OF THE NC-TH DERIVATIVE FOR THE CONSTRAINT AT XC *00000018
C* NC      = CONSTRAINT ORDER                          *00000019
C* NCONST = NUMBER OF CONSTRAINTS                      *00000020
C* XC      = X-VALUE FOR THE CONSTRAINT                *00000021
C*                                                    *00000022
C* PARAMETERS:                                         *00000023
C*                                                    *00000024
C* MXNAC   = MAXIMUM NUMBER OF ALLOWABLE CONSTRAINTS *00000025
C*                                                    *00000026
C*****00000027
C                                                    00000028
C                                                    00000029
C          IMPLICIT REAL*8(A-H,O-Z)                   00000030
C                                                    00000031
C          CHARACTER CCV*1,DFmt*3,File*80,Fmt*80      00000032
C                                                    00000033
C          INTEGER DUM2                                00000034
C                                                    00000035
C          PARAMETER (R_NaN=-999.E+00)                00000036
C          PARAMETER (MXNAC=20)                       00000037
C                                                    00000038
C          COMMON/CV/CV(MXNAC)                        00000039
C          COMMON/NC/NC(MXNAC)                        00000040
C          COMMON/NCONST/NCONST                      00000041
C          COMMON/XC/XC(MXNAC)                       00000042
C                                                    00000043
C          DATA DFmt/'(*)'/'                        00000044
C                                                    00000045
C          NAMELIST/PARM/CV,NC,XC                     00000046
C                                                    00000047
C                                                    00000048
C*****00000049
C*                                                    *00000050
C*          INPUT THE CONSTRAINTS                      *00000051
C*                                                    *00000052
C*****00000053
C                                                    00000054
C                                                    00000055
C          DO 101 I=1,MXNAC                            00000056
C          CV(I)=R_NaN                                00000057
C          NC(I)=-1                                    00000058
101  XC(I)=R_NaN                                       00000059
C          WRITE(UNIT=6,FMT=201)                      00000060

```


	READ(UNIT=5,FMT=202)LCV1	00000061
	IF(LCV1.EQ.2) GO TO 104	00000062
C		00000063
102	WRITE(UNIT=6,FMT=203)MXNAC	00000064
	WRITE(UNIT=6,FMT=204)	00000065
	CALL NDRUS(' PARM',4,*102,*121)	00000066
	READ(UNIT=4,NML=PARM,ERR=102)	00000067
	CALL NDRUSE(*102,*121)	00000068
	DO 103 NCONST=MXNAC,1,-1	00000069
	IF(XC(NCONST).NE.R_NaN) GO TO 111	00000070
103	CONTINUE	00000071
	GO TO 111	00000072
C		00000073
104	WRITE(UNIT=6,FMT=205)MXNAC	00000074
	WRITE(UNIT=6,FMT=206)	00000075
	READ(UNIT=5,FMT=207)FILE	00000076
105	WRITE(UNIT=6,FMT=208)	00000077
	WRITE(UNIT=6,FMT=209)	00000078
	CALL DFCUS(Fmt,DFmt,*105)	00000079
	OPEN(UNIT=4,FILE=File,STATUS='OLD',ERR=122)	00000080
	REWIND 4	00000081
	IF(Fmt(1:3).EQ.DFmt) GO TO 107	00000082
	DO 106 NCONST=1,MXNAC	00000083
106	READ(UNIT=4,FMT=Fmt,END=109,ERR=123)XC(NCONST),NC(NCONST),	00000084
	>CV(NCONST)	00000085
	GO TO 110	00000086
107	DO 108 NCONST=1,MXNAC	00000087
108	READ(UNIT=4,FMT=*,END=109,ERR=123)XC(NCONST),NC(NCONST),CV(NCONST)	00000088
	GO TO 110	00000089
109	NCONST=NCONST-1	00000090
110	CLOSE(UNIT=4,STATUS='KEEP',ERR=124)	00000091
C		00000092
C		00000093
C	*****	00000094
C*		*00000095
C*	SORT AND EDIT THE CONSTRAINTS	*00000096
C*		*00000097
C	*****	00000098
C		00000099
C		00000100
111	IF(NCONST.EQ.0) GO TO 120	00000101
	IF(NCONST.EQ.1) GO TO 117	00000102
	DO 112 I=1,NCONST-1	00000103
	DO 112 J=I+1,NCONST	00000104
	IF(XC(I).NE.XC(J)) GO TO 112	00000105
	IF(NC(I).NE.NC(J)) GO TO 112	00000106
	XC(I)=R_NaN	00000107
112	CONTINUE	00000108
113	LCV1=0	00000109
	DO 115 I=2,NCONST	00000110
	IF(XC(I-1).LT.XC(I)) GO TO 115	00000111
	IF(XC(I-1).GT.XC(I)) GO TO 114	00000112
	IF(NC(I-1).LT.NC(I)) GO TO 115	00000113
114	LCV1=1	00000114
	DUM1=XC(I-1)	00000115
	DUM2=NC(I-1)	00000116
	DUM3=CV(I-1)	00000117
	XC(I-1)=XC(I)	00000118
	NC(I-1)=NC(I)	00000119
	CV(I-1)=CV(I)	00000120

	XC(I)=DUM1	00000121
	NC(I)=DUM2	00000122
	CV(I)=DUM3	00000123
115	CONTINUE	00000124
	IF(LCV1.EQ.1) GO TO 113	00000125
	DO 116 NCONST=MXNAC,1,-1	00000126
	IF(XC(NCONST).NE.R_NaN) GO TO 117	00000127
116	CONTINUE	00000128
C		00000129
117	WRITE(UNIT=6,FMT=210)	00000130
	CALL YNOUS(*118,*120,*117)	00000131
118	WRITE(UNIT=6,FMT=211)	00000132
	DO 119 I=1,NCONST	00000133
	IF(MOD(I,20).NE.0) GO TO 119	00000134
	WRITE(UNIT=6,FMT=212)	00000135
	READ(UNIT=5,FMT=213)CCV	00000136
	WRITE(UNIT=6,FMT=211)	00000137
119	WRITE(UNIT=6,FMT=214)I,XC(I),NC(I),CV(I)	00000138
	WRITE(UNIT=6,FMT=212)	00000139
	READ(UNIT=5,FMT=213)CCV	00000140
120	RETURN	00000141
C		00000142
C		00000143
C	*****	00000144
C*		*00000145
C*	ERROR MESSAGES	*00000146
C*		*00000147
C	*****	00000148
C		00000149
C		00000150
121	WRITE(UNIT=6,FMT=215)	00000151
	GO TO 126	00000152
122	WRITE(UNIT=6,FMT=216)	00000153
	CALL CBUS06(6,File,*125)	00000154
	RETURN 1	00000155
123	WRITE(UNIT=6,FMT=217)	00000156
	CALL CBUS06(6,File,*125)	00000157
	RETURN 1	00000158
124	WRITE(UNIT=6,FMT=218)	00000159
	CALL CBUS06(6,File,*125)	00000160
	RETURN 1	00000161
125	WRITE(UNIT=6,FMT=219)	00000162
	RETURN 1	00000163
126	WRITE(UNIT=6,FMT=212)	00000164
	READ(UNIT=5,FMT=213)CCV	00000165
	RETURN 2	00000166
C		00000167
C		00000168
C	*****	00000169
C*		*00000170
C*	FORMAT STATEMENTS	*00000171
C*		*00000172
C	*****	00000173
C		00000174
C		00000175
201	FORMAT(/,'SELECT THE SOURCE OF INPUT FOR THE CONSTRAINTS FROM THE	00000176
	>FOLLOWING LIST:','//','1, FOR KEYBOARD INPUT VIA NAMELIST','/,','2, FOR	00000177
	> FORMATTED DISK FILE INPUT','//','NOTE: THE CONSTRAINTS ARE COMPLETE	00000178
	>LY INDEPENDENT OF THE X-Y DATA POINTS TO BE','/,','FITTED.')	00000179
202	FORMAT(I1)	00000180

```

203  FORMAT(/,'INPUT THE CONSTRAINTS (' ,I2,' MAX) BY NAMELIST WHERE:',/00000181
>/, 'CV      = ARRAY OF POLYNOMIAL DERIVATIVE VALUES',/, 'NC      = ARR00000182
>AY OF POLYNOMIAL DERIVATIVE ORDERS (CONSTRAINT ORDER)',/, 'XC      =00000183
> ARRAY OF POLYNOMIAL DERIVATIVE LOCATIONS (CONSTRAINT X VALUES)',/00000184
>/, 'NOTE: NC VALUES MUST BE IN THE RANGE 0 TO NP.',/,T2,'CURRENT V00000185
>ALUES ARE:') 00000186
204  FORMAT(/,'$PARM CV=____,____,____,NC=____,____,____,XC=____,____,____00000187
>____$',/) 00000188
205  FORMAT(/,'CONSTRAINTS (' ,I2,' MAX) ARE INPUT FROM FORMATTED DISK F00000189
>ILES AS (XC,NC,CV) TRIPLES',/, 'WHERE:',/, 'CV      = POLYNOMIAL DER00000190
>IVATIVE VALUE',/, 'NC      = POLYNOMIAL DERIVATIVE OR CONSTRAINT ORD00000191
>ER',/, 'XC      = POLYNOMIAL DERIVATIVE LOCATION (CONSTRAINT X VALUE00000192
>)',/, 'NOTE: NC VALUES MUST BE IN THE RANGE 0 TO NP.') 00000193
206  FORMAT(/,'INPUT THE FILE NAME OF THE FORMATTED DISK FILE DATA SET.00000194
>') 00000195
207  FORMAT(A80) 00000196
208  FORMAT(/,'INPUT THE DATA FILE FORMAT (INCLUDE PARENTHESES)') 00000197
209  FORMAT('NOTE: AN EXAMPLE FORMAT IS "(2E15.6)"',/,T7,'ENTER"(*)" F000000198
>R A FREE FIELD READ (DEFAULT FORMAT)') 00000199
210  FORMAT(/,'SHOULD THE CONSTRAINTS BE DISPLAYED FOR VERIFICATION? (Y00000200
>/N)') 00000201
211  FORMAT(/,T6,'NO.',T27,'XC(I)',T35,'NC(I)',T58,'CV(I)') 00000202
212  FORMAT(/,T19,'- ENTER/RETURN TO CONTINUE -') 00000203
213  FORMAT(A1) 00000204
214  FORMAT(T8,I3,1PD23.13,I7,D23.13) 00000205
215  FORMAT('SUBROUTINE NDRUS WAS CALLED FROM PROGRAM LSPRWC') 00000206
216  FORMAT(/,'ERROR IN PROGRAM LSPRWC: OPEN ERROR ON UNIT 4, STATUS = 00000207
>"OLD", FILE =',/) 00000208
217  FORMAT(/,'ERROR IN PROGRAM LSPRWC: READ ERROR ON UNIT 4, FILE =',/00000209
>) 00000210
218  FORMAT(/,'ERROR IN PROGRAM LSPRWC: CLOSE ERROR ON UNIT 4, FILE =',00000211
>/) 00000212
219  FORMAT('SUBROUTINE CBUS06 WAS CALLED FROM PROGRAM LSPRWC') 00000213
      END 00000214

```

FUNCTION FACTRL(N)	00000001
C	00000002
C	00000003
C*****	00000004
C*	*00000005
C* FACTORIAL EVALUATION FUNCTION SUBPROGRAM (FACTRL)	*00000006
C*	*00000007
C* REVISION DATE: 15 JUNE 1989	*00000008
C*	*00000009
C*****	00000010
C	00000011
C	00000012
INTEGER FACTRL	00000013
FACTRL=1	00000014
IF(N.LE.1) RETURN	00000015
DO 101 I=0,N-1	00000016
101 FACTRL=FACTRL*(N-I)	00000017
END	00000018

```

SUBROUTINE FDFOS(LCV2,*)                                00000001
C                                                         00000002
C                                                         00000003
C*****00000004
C*                                                         *00000005
C*          FORMATTED DISK FILE OUTPUT SUBROUTINE (FDFOS) *00000006
C*                                                         *00000007
C*          REVISION DATE: 2 OCTOBER 2007                 *00000008
C*                                                         *00000009
C*****00000010
C                                                         00000011
C*****00000012
C*                                                         *00000013
C* PARAMETERS:                                           *00000014
C*                                                         *00000015
C* MXANC  = MAXIMUM NUMBER OF ALLOWABLE CONSTRAINTS      *00000016
C* MXNR   = MAXIMUM NUMBER OF ROWS                       *00000017
C* MXNRM1 = MAXIMUM NUMBER OF ROWS MINUS 1 (MXNRM1 = MXNR - 1) *00000018
C*                                                         *00000019
C*****00000020
C                                                         00000021
C                                                         00000022
C          IMPLICIT REAL*8(A-H,O-Z)                     00000023
C                                                         00000024
C          CHARACTER Alist*1,Blank*1,DFmt*9,File*80,Fmt*80,SList*1 00000025
C                                                         00000026
C          PARAMETER (MXANC=20,MXNRM1=29)                00000027
C                                                         00000028
C          COMMON/B/B(0:MXNRM1)                          00000029
C          COMMON/CV/CV(MXANC)                           00000030
C          COMMON/NC/NC(MXANC)                           00000031
C          COMMON/NCONST/NCONST                          00000032
C          COMMON/NDATA/NDATA                            00000033
C          COMMON/NEVAL/NEVAL                             00000034
C          COMMON/NP/NP                                   00000035
C          COMMON/SDEV/SDEV                               00000036
C          COMMON/XC/XC(MXANC)                           00000037
C                                                         00000038
C          DIMENSION AList(13),NList(13),SList(-1:11),YK(-1:11) 00000039
C                                                         00000040
C          SAVE Blank,DFmt                               00000041
C                                                         00000042
C          DATA Blank/' ',DFmt/'(13E15.6)'/,SList/"X","0","1","2","3","4", 00000043
C          >"5","6","7","8","9","R","S"/                  00000044
C                                                         00000045
C          NAMELIST/PARM/AList                           00000046
C                                                         00000047
C                                                         00000048
C*****00000049
C*                                                         *00000050
C*          SELECT AN OUTPUT OPTION                      *00000051
C*                                                         *00000052
C*****00000053
C                                                         00000054
C                                                         00000055
C          WRITE(UNIT=6,FMT=201)                         00000056
C          READ(UNIT=5,FMT=202)FILE                      00000057
C          WRITE(UNIT=6,FMT=203)                         00000058
C          READ(UNIT=5,FMT=204)LCV1                     00000059
C          IF(LCV1.EQ.2.AND.LCV2.EQ.0) GO TO 113         00000060

```

```

OPEN(UNIT=4,FILE=File,FORM='FORMATTED',STATUS='UNKNOWN',ERR=114) 00000061
REWIND 4 00000062
IF(LCV1.EQ.2) GO TO 104 00000063
C 00000064
C 00000065
C*****00000066
C* 00000067
C* OUTPUT ALL INPUT DATA AND RESULTS 00000068
C* 00000069
C*****00000070
C 00000071
C 00000072
WRITE(UNIT=4,FMT=205,ERR=115) 00000073
WRITE(UNIT=4,FMT=206,ERR=115) 00000074
REWIND 1 00000075
DO 101 I=1,NDATA 00000076
READ(UNIT=1,ERR=116)X,Y 00000077
101 WRITE(UNIT=4,FMT=207,ERR=115)I,X,Y 00000078
IF(NCONST.EQ.0) GO TO 102 00000079
WRITE(UNIT=4,FMT=208,ERR=115) 00000080
WRITE(UNIT=4,FMT=209,ERR=115)(I,XC(I),NC(I),CV(I),I=1,NCONST) 00000081
102 WRITE(UNIT=4,FMT=210,ERR=115) 00000082
WRITE(UNIT=4,FMT=211,ERR=115)(I,B(I),I=0,NP) 00000083
WRITE(UNIT=4,FMT=212,ERR=115)NP,SDEV 00000084
IF(LCV2.EQ.0) GO TO 112 00000085
WRITE(UNIT=4,FMT=213,ERR=115) 00000086
REWIND 3 00000087
DO 103 I=1,NEVAL 00000088
READ(UNIT=3,ERR=117)X,Y 00000089
103 WRITE(UNIT=4,FMT=207,ERR=115)I,X,Y 00000090
GO TO 112 00000091
C 00000092
C 00000093
C*****00000094
C* 00000095
C* OUTPUT ONLY THE EVALUATED DATA 00000096
C* 00000097
C*****00000098
C 00000099
C 00000100
104 DO 105 K=3,13 00000101
AList(K)=Blank 00000102
105 NList(K)=-999 00000103
AList(1)='X' 00000104
AList(2)='0' 00000105
WRITE(UNIT=6,FMT=214)SList(-1) 00000106
WRITE(UNIT=6,FMT=215)(SList(K),K,K=0,NP) 00000107
WRITE(UNIT=6,FMT=216)(SList(K),K=10,11) 00000108
WRITE(UNIT=6,FMT=217) 00000109
CALL NDRUS('PARM',9,*104,*118) 00000110
READ(UNIT=9,NML=PARM,ERR=104) 00000111
CALL NDRUSE(*104,*118) 00000112
DO 106 NL=1,13 00000113
IF(AList(NL).EQ.Blank) GO TO 107 00000114
106 CONTINUE 00000115
107 NL=NL-1 00000116
IF(NL.EQ.0) GO TO 119 00000117
DO 110 I=1,NL 00000118
DO 108 K=-1,11 00000119
IF(AList(I).EQ.SList(K)) GO TO 109 00000120

```

108	CONTINUE	00000121
	GO TO 120	00000122
109	NList(I)=K	00000123
	IF(K.EQ.10) NList(I)=NP+1	00000124
	IF(K.EQ.11) NList(I)=NP+2	00000125
	IF(NP.LT.K.AND.K.LT.10) GO TO 120	00000126
110	CONTINUE	00000127
C		00000128
	WRITE(UNIT=6,FMT=218)	00000129
	CALL CBUS06(6,DFmt,*122)	00000130
	CALL DFCUS(Fmt,DFmt,*104)	00000131
	REWIND 3	00000132
	DO 111 I=1,NEVAL	00000133
	READ(UNIT=3,ERR=117)(YK(K),K=-1,NP+2)	00000134
111	WRITE(UNIT=4,FMT=Fmt,ERR=115)(YK(NList(K)),K=1,NL)	00000135
112	CLOSE(UNIT=4,STATUS='KEEP',ERR=121)	00000136
	RETURN	00000137
C		00000138
C		00000139
C	*****	00000140
C*		*00000141
C*	ERROR MESSAGES	*00000142
C*		*00000143
C	*****	00000144
C		00000145
C		00000146
113	WRITE(UNIT=6,FMT=219)	00000147
	CALL CBUS06(6,File,*122)	00000148
	WRITE(UNIT=6,FMT=220)	00000149
	RETURN	00000150
114	WRITE(UNIT=6,FMT=221)	00000151
	CALL CBUS06(6,File,*122)	00000152
	RETURN 1	00000153
115	WRITE(UNIT=6,FMT=222)	00000154
	CALL CBUS06(6,File,*122)	00000155
	RETURN 1	00000156
116	WRITE(UNIT=6,FMT=223)	00000157
	RETURN 1	00000158
117	WRITE(UNIT=6,FMT=224)	00000159
	RETURN 1	00000160
118	WRITE(UNIT=6,FMT=225)	00000161
	RETURN 1	00000162
119	WRITE(UNIT=6,FMT=226)	00000163
	GO TO 104	00000164
120	WRITE(UNIT=6,FMT=227)I,AList(I)	00000165
	GO TO 104	00000166
121	WRITE(UNIT=6,FMT=228)	00000167
	CALL CBUS06(6,File,*122)	00000168
	RETURN 1	00000169
122	WRITE(UNIT=6,FMT=229)	00000170
	RETURN 1	00000171
C		00000172
C		00000173
C	*****	00000174
C*		*00000175
C*	FORMAT STATEMENTS	*00000176
C*		*00000177
C	*****	00000178
C		00000179
C		00000180

```

201  FORMAT(/,'INPUT THE FILE NAME OF THE FORMATTED DISK FILE.')          00000181
202  FORMAT(A80)                                                            00000182
203  FORMAT(/,'SELECT THE TYPE OF OUTPUT FOR THE FORMATTED DISK FILE FR00000183
>OM THE FOLLOWING LIST:',/, '1, FOR ALL INPUT DATA AND RESULTS',/, '200000184
>, FOR ONLY THE EVALUATED DATA')                                         00000185
204  FORMAT(I1)                                                            00000186
205  FORMAT(T6,'LEAST SQUARES POLYNOMIAL REGRESSION WITH CONSTRAINTS (L00000187
>SPRWC)',/,T22,'WRITTEN BY: C.D. MIKKELSEN',/,T31,'2 MAY 1988',/,00000188
>,T13,'AERODYNAMICS TECHNOLOGY BRANCH (AMSMI-RD-SS-AT)',/,T13,'SYST00000189
>EMS SIMULATION AND DEVELOPMENT DIRECTORATE',/,T6,'US ARMY MISSILE 00000190
>RESEARCH, ENGINEERING, AND DEVELOPMENT CENTER',/,T25,'US ARMY MISS00000191
>ILE COMMAND',/,T18,'REDSTONE ARSENAL, ALABAMA 35898-5252')              00000192
206  FORMAT(/,T20,'X-Y DATA POINTS TO BE FITTED:',/,T12,'NO.',T28,'X',00000193
>T51,'Y')                                                                  00000194
207  FORMAT(T10,I5,1PD23.13)                                              00000195
208  FORMAT(/,T25,'POLYNOMIAL CONSTRAINTS',/,T8,'NO.',T29,'XC(I)',T37,00000196
>'NC(I)',T60,'CV(I)')                                                    00000197
209  FORMAT(T8,I3,1PD23.13,I7,D23.13)                                     00000198
210  FORMAT(/,T28,'LEAST SQUARES POLYNOMIAL',/,T15,'P(X)=B(0)+B(1)*X+B00000199
>(2)*X**2+....+B(NP)*X**NP',/,T23,'I',T37,'B(I)',/)                    00000200
211  FORMAT(T23,I1,1PD25.13)                                              00000201
212  FORMAT(/,T1,'THE STANDARD DEVIATION FOR THIS POLYNOMIAL OF ORDER '00000202
>,I1,' IS ',1PD12.5)                                                    00000203
213  FORMAT(/,T22,'X-Y EVALUATED DATA POINTS:',/,T12,'NO.',T28,'X',T51,00000204
>'Y')                                                                      00000205
214  FORMAT(/,'INPUT THE EVALUATED DATA VARIABLE LIST FOR OUTPUT BY NAM00000206
>ELIST WHERE:',/, ' ',A1,' ' = X VALUES')                              00000207
215  FORMAT(' ',A1,' ' = ',I1,'th DERIVATIVE OF Y AT X')                00000208
216  FORMAT(' ',A1,' ' = RADIUS OF CURVATURE AT X',/, ' ',A1,' ' = A00000209
>RC LENGTH',/, 'NOTE: VARIABLES WILL BE LISTED IN THE ORDER OF INPU00000210
>T',/, 'CURRENT VALUES ARE:')                                           00000211
217  FORMAT(/,'$PARM AList= "X","0"$',/)                                  00000212
218  FORMAT(/,'INPUT THE DATA FILE FORMAT (INCLUDE PARENTHESES)',/, 'NO00000213
>TE: AN EXAMPLE FORMAT IS "(F15.7,E15.6)"',/,T7,'THE DEFAULT FORMAT00000214
> IS:')                                                                    00000215
219  FORMAT(/,'WARNING IN SUBROUTINE FDFOS: NO DATA POINTS HAVE BEEN PR00000216
>ESCRIBED FOR EVALUATION',/, 'OF THE LEAST SQUARES POLYNOMIAL. HENCE00000217
>, FILE',/)                                                              00000218
220  FORMAT(/,'WILL NOT BE OPENED.')                                       00000219
221  FORMAT(/,'ERROR IN SUBROUTINE FDFOS: OPEN ERROR ON UNIT 4, STATUS 00000220
>= "UNKNOWN", FILE =',/)                                                00000221
222  FORMAT(/,'ERROR IN SUBROUTINE FDFOS: WRITE ERROR ON UNIT 4, FILE =00000222
>',/)                                                                    00000223
223  FORMAT(/,'ERROR IN SUBROUTINE FDFOS: UNFORMATTED READ ERROR ON UNI00000224
>T 1')                                                                    00000225
224  FORMAT(/,'ERROR IN SUBROUTINE FDFOS: UNFORMATTED READ ERROR ON UNI00000226
>T 3')                                                                    00000227
225  FORMAT('SUBROUTINE NDRUS WAS CALLED FROM SUBROUTINE FDFOS')          00000228
226  FORMAT(/,'ERROR IN SUBROUTINE FDFOS: THE OUTPUT LIST MUST NOT BE E00000229
>MPTY')                                                                    00000230
227  FORMAT(/,'ERROR IN SUBROUTINE FDFOS: ILLEGAL VALUE FOR',/, 'AList('00000231
>,I2,') = ',A1)                                                         00000232
228  FORMAT(/,'ERROR IN SUBROUTINE FDFOS: CLOSE ERROR ON UNIT 4, FILE =00000233
>',/)                                                                    00000234
229  FORMAT('SUBROUTINE CBUS06 WAS CALLED FROM SUBROUTINE FDFOS')        00000235
END                                                                        00000236

```



```

SUBROUTINE GJEMPS(EPS,INDIC,N,DETER,*)
C
C
C*****
C*
C*      GAUSS-JORDAN ELIMINATION USING MAXIMUM PIVOT STRATEGY (GJEMPS)
C*
C*      REVISION DATE: 21 APRIL 2000
C*
C*****
C
C*****
C*
C* SUBROUTINE GJEMPS COMPUTES THE INVERSE OF AN N BY N MATRIX AND/OR
C* THE N SOLUTIONS OF A SET OF N LINEAR EQUATIONS IN N UNKNOWNNS USING
C* GAUSS-JORDAN ELIMINATION WITH MAXIMUM PIVOT STRATEGY.
C*
C* INPUT VARIABLES:
C*
C* EPS      = MINIMUM ALLOWABLE MAGNITUDE FOR A PIVOT ELEMENT (SUGGESTED
C*              VALUE 1.0E-10)
C* INDIC    = CONTROL VARIABLE SUCH THAT:
C*              = NEGATIVE, TO COMPUTE THE INVERSE OF N BY N MATRIX A IN
C*              PLACE
C*              = ZERO      , TO COMPUTE THE N SOLUTIONS X(1)...X(N)
C*              CORRESPONDING TO THE SET OF LINEAR EQUATIONS WITH
C*              AUGMENTED MATRIX OF COEFFICIENTS IN THE N BY N+1 ARRAY A
C*              AND IN ADDITION COMPUTE THE INVERSE OF THE COEFFICIENT
C*              MATRIX IN PLACE
C*              = POSITIVE, TO COMPUTE THE N SOLUTIONS X(1)...X(N)
C*              CORRESPONDING TO THE SET OF LINEAR EQUATIONS WITH
C*              AUGMENTED MATRIX OF COEFFICIENTS IN THE N BY N+1 ARRAY A
C* N        = NUMBER OF ROWS IN MATRIX A
C*
C* OUTPUT VARIABLES:
C*
C* DETER    = DETERMINATE OF THE ORIGINAL COEFFICIENT MATRIX FORMED BY
C*              THE FIRST N COLUMNS OF ARRAY A
C* X        = VECTOR OF N SOLUTIONS
C*
C* INPUT/OUTPUT VARIABLES:
C*
C* A        = AUGMENTED COEFFICIENT MATRIX
C*
C* PARAMETERS:
C*
C* MXNC     = MAXIMUM NUMBER OF COLUMNS (MXNC = MXNR + 1)
C* MXNCM1   = MAXIMUM NUMBER OF COLUMNS MINUS 1 (MXNCM1 = MXNC - 1)
C* MXNR     = MAXIMUM NUMBER OF ROWS
C* MXNRM1   = MAXIMUM NUMBER OF ROWS MINUS 1 (MXNRM1 = MXNR - 1)
C*
C* NOTE: SHOULD NO ACCEPTABLE PIVOT ELEMENT BE FOUND, COMPUTATIONS ARE
C*       TERMINATED AND THE DETERMINATE IS RETURNED WITH A TRUE ZERO
C*       VALUE.
C*
C* REF: CARNAHAN, B., LUTHER, H.A., AND WILKES, J.O.: APPLIED
C*       NUMERICAL METHODS, NEW YORK, 1969, JOHN WILEY & SONS, INC.
C*
C*****
C

```

C		00000061
	IMPLICIT REAL*8(A-H,O-Z)	00000062
C		00000063
	PARAMETER (MXNC=31,MXNCM1=30,MXNR=30,MXNRM1=29)	00000064
C		00000065
	DIMENSION A(MXNR,MXNC), IROW(MXNR), JCOL(MXNR), JORD(MXNR), X(MXNR),	00000066
	>Y(MXNR)	00000067
C		00000068
	COMMON/B/B(0:MXNRM1)	00000069
	COMMON/C/C(0:MXNRM1,0:MXNCM1)	00000070
C		00000071
	EQUIVALENCE (A(1,1),C(0,0)), (X(1),B(0))	00000072
C		00000073
C		00000074
C	*****	00000075
C*		*00000076
C*	BEGIN ELIMINATION PROCEDURE	*00000077
C*		*00000078
C	*****	00000079
C		00000080
C		00000081
	IF(N.GT.MXNR) GO TO 114	00000082
	MAX=N	00000083
	IF(INDIC.GE.0) MAX=N+1	00000084
	DETER=1.0D+00	00000085
	DO 107 K=1,N	00000086
	KM1=K-1	00000087
C		00000088
C		00000089
C	*****	00000090
C*		*00000091
C*	SEARCH FOR THE PIVOT ELEMENT	*00000092
C*		*00000093
C	*****	00000094
C		00000095
C		00000096
	PIVOT=0.0D+00	00000097
	DO 103 I=1,N	00000098
	DO 103 J=1,N	00000099
C		00000100
C		00000101
C	*****	00000102
C*		*00000103
C*	SCAN IROW AND JCOL ARRAYS FOR INVALID PIVOT SUBSCRIPTS	*00000104
C*		*00000105
C	*****	00000106
C		00000107
C		00000108
	IF(K.EQ.1) GO TO 102	00000109
	DO 101 ISCAN=1,KM1	00000110
	DO 101 JSCAN=1,KM1	00000111
	IF(I.EQ.IROW(ISCAN)) GO TO 103	00000112
101	IF(J.EQ.JCOL(JSCAN)) GO TO 103	00000113
102	IF(DABS(A(I,J)).LE.DABS(PIVOT)) GO TO 103	00000114
	PIVOT=A(I,J)	00000115
	IROW(K)=I	00000116
	JCOL(K)=J	00000117
103	CONTINUE	00000118
C		00000119
C		00000120

```

C*****00000121
C*00000122
C*INSURE THE SELECTED PIVOT IS LARGER THAN EPS00000123
C*00000124
C*****00000125
C00000126
C00000127
IF(DABS(PIVOT).GT.EPS) GO TO 10400000128
DETER=0.0D+000000129
RETURN00000130
C00000131
C00000132
C*****00000133
C*00000134
C*UPDATE THE DETERMINATE VALUE AND NORMALIZE PIVOT ROW ELEMENTS00000135
C*00000136
C*****00000137
C00000138
C00000139
104DETER=DETER*PIVOT00000140
DO 105 J=1,MAX00000141
105A(IROW(K),J)=A(IROW(K),J)/PIVOT00000142
C00000143
C00000144
C*****00000145
C*00000146
C*00000147
C*00000148
C*****00000149
C00000150
C00000151
A(IROW(K),JCOL(K))=1.0D+00/PIVOT00000152
DO 107 I=1,N00000153
AIJCK=A(I,JCOL(K))00000154
IF(I.EQ.IROW(K)) GO TO 10700000155
A(I,JCOL(K))=-AIJCK/PIVOT00000156
DO 106 J=1,MAX00000157
106IF(J.NE.JCOL(K)) A(I,J)=A(I,J)-AIJCK*A(IROW(K),J)00000158
107CONTINUE00000159
C00000160
C00000161
C*****00000162
C*00000163
C*ORDER SOLUTION VALUES (IF ANY) AND CREATE JORD ARRAY00000164
C*00000165
C*****00000166
C00000167
C00000168
DO 108 I=1,N00000169
JORD(IROW(I))=JCOL(I)00000170
108IF(INDIC.GE.0) X(JCOL(I))=A(IROW(I),MAX)00000171
C00000172
C00000173
C*****00000174
C*00000175
C*00000176
C*00000177
C*****00000178
C00000179
C00000180

```

```

        INTCH=0                                00000181
        DO 109 I=1,N-1                          00000182
        DO 109 J=I+1,N                          00000183
        IF(JORD(J).GE.JORD(I)) GO TO 109        00000184
        JTEMP=JORD(J)                          00000185
        JORD(J)=JORD(I)                        00000186
        JORD(I)=JTEMP                          00000187
        INTCH=INTCH+1                          00000188
109     CONTINUE                               00000189
        IF(INTCH/2*2.NE.INTCH) DETER=-DETER    00000190
C                                              00000191
C                                              00000192
C*****00000193
C*                                              *00000194
C*                                UNSCRAMBLE THE INVERSE *00000195
C*                                              *00000196
C*****00000197
C                                              00000198
C                                              00000199
C                                IF(INDIC.GT.0) RETURN 00000200
        DO 111 J=1,N                          00000201
        DO 110 I=1,N                          00000202
110     Y(JCOL(I))=A(IROW(I),J)                00000203
        DO 111 I=1,N                          00000204
111     A(I,J)=Y(I)                            00000205
        DO 113 I=1,N                          00000206
        DO 112 J=1,N                          00000207
112     Y(IROW(J))=A(I,JCOL(J))                00000208
        DO 113 J=1,N                          00000209
113     A(I,J)=Y(J)                            00000210
        RETURN                                00000211
C                                              00000212
C                                              00000213
C*****00000214
C*                                              *00000215
C*                                ERROR MESSAGES *00000216
C*                                              *00000217
C*****00000218
C                                              00000219
C                                              00000220
114     WRITE(UNIT=6,FMT=201)N,MXNC            00000221
        RETURN 1                             00000222
C                                              00000223
C                                              00000224
C*****00000225
C*                                              *00000226
C*                                FORMAT STATEMENTS *00000227
C*                                              *00000228
C*****00000229
C                                              00000230
C                                              00000231
201     FORMAT(/,'ERROR IN SUBROUTINE GJEMPS: THE REQUESTED NUMBER OF ROWS00000232
>,N',/, 'EXCEEDS THE DIMENSIONED MAXIMUM NUMBER OF ROWS, MXNR FOR',/00000233
>,'N =',I5,' MXNR =',I5,/, 'REDIMENSION PARAMETERS:',/, 'MXNR  = M00000234
>AXIMUM NUMBER OF ROWS',/, 'MXNC  = MAXIMUM NUMBER OF COLUMNS',/, 'M00000235
>XNRM1 = MAXIMUM NUMBER OF ROWS MINUS 1',/, 'MXNCM1 = MAXIMUM NUMBER00000236
> OF COLUMNS MINUS 1',/, 'AND RECOMPILE.')    00000237
        END                                  00000238

```

```

SUBROUTINE LSPRWC(*)                                00000001
C                                                    00000002
C                                                    00000003
C*****00000004
C*                                                    00000005
C*   LEAST SQUARES POLYNOMIAL REGRESSION WITH CONSTRAINTS (LSPRWC) 00000006
C*                                                    00000007
C*   REVISION DATE: 6 JULY 2011                        00000008
C*                                                    00000009
C*****00000010
C                                                    00000011
C*****00000012
C*                                                    00000013
C*   SUBROUTINE LSPRWC PERFORMS A LEAST SQUARES POLYNOMIAL REGRESSION 00000014
C*   WITH CONSTRAINTS; THAT IS, A SET OF X-Y DATA POINTS IS CURVE FIT 00000015
C*   WITH AN NP ORDER POLYNOMIAL OF THE FORM             00000016
C*                                                    00000017
C*            $P(X)=B0+B1*X+B2*X**2+B3*X**3+...+BNP*X**NP$  00000018
C*                                                    00000019
C*   WITH ANY POLYNOMIAL DERIVATIVES, ZERO THROUGH NP, SPECIFIED AT 00000020
C*   GIVEN X LOCATIONS.  THE PROCEDURE FOLLOWED IS THE METHOD OF LEAST 00000021
C*   SQUARES USING UNDETERMINED LAGRANGE MULTIPLIERS.    00000022
C*                                                    00000023
C*   INPUT VARIABLES:                                   00000024
C*                                                    00000025
C*   CV      = ARRAY OF POLYNOMIAL DERIVATIVE VALUES    00000026
C*   NC      = ARRAY OF POLYNOMIAL DERIVATIVE ORDERS (CONSTRAINT ORDER) 00000027
C*   NCONST  = NUMBER OF CONSTRAINTS OR (XC,NC,CV) DATA TRIPLES 00000028
C*   NDATA   = NUMBER OF X-Y DATA PAIRS                 00000029
C*   NP      = POLYNOMIAL ORDER                         00000030
C*   X       = DATA POINT X VALUE                     00000031
C*   XC      = ARRAY OF POLYNOMIAL DERIVATIVE LOCATIONS (CONSTRAINT X 00000032
C*             VALUES)                                00000033
C*   Y       = DATA POINT Y VALUE                     00000034
C*                                                    00000035
C*   OUTPUT VARIABLES:                                  00000036
C*                                                    00000037
C*   B       = ARRAY OF REGRESSION COEFFICIENTS         00000038
C*   SDEV    = STANDARD DEVIATION OF THE X-Y DATA POINTS ABOUT THE 00000039
C*             REGRESSION LINE                          00000040
C*                                                    00000041
C*   PARAMETERS:                                        00000042
C*                                                    00000043
C*   MXANC   = MAXIMUM NUMBER OF ALLOWABLE CONSTRAINTS 00000044
C*   MXNC    = MAXIMUM NUMBER OF COLUMNS (MXNC = MXNR + 1) 00000045
C*   MXNCM1  = MAXIMUM NUMBER OF COLUMNS MINUS 1 (MXNCM1 = MXNC - 1) 00000046
C*   MXNR    = MAXIMUM NUMBER OF ROWS                  00000047
C*   MXNRM1  = MAXIMUM NUMBER OF ROWS MINUS 1 (MXNRM1 = MXNR - 1) 00000048
C*                                                    00000049
C*   REF: CARNAHAN,B., LUTHER, H.A., AND WILKES, J.O.: APPLIED NUMERICAL 00000050
C*           METHODS, NEW YORK, JOHN WILEY & SONS, 1969, PP. 571-584. 00000051
C*                                                    00000052
C*           WEINSTOCK, R.: CALCULUS OF VARIATIONS WITH APPLICATIONS TO 00000053
C*           PHYSICS AND ENGINEERING, NEW YORK, DOVER PUBLICATIONS, INC., 00000054
C*           1974, P. 6.                                00000055
C*                                                    00000056
C*****00000057
C                                                    00000058
C                                                    00000059
C   IMPLICIT REAL*8(A-H,O-Z)                        00000060

```

C		00000061
	INTEGER COL, COLMAX, FACTRL, ROW, ROWMAX	00000062
C		00000063
	PARAMETER (MXANC=20, MXNCM1=30, MXNRM1=29)	00000064
C		00000065
	COMMON/B/B(0:MXNRM1)	00000066
	COMMON/C/C(0:MXNRM1, 0:MXNCM1)	00000067
	COMMON/CV/CV(MXANC)	00000068
	COMMON/NC/NC(MXANC)	00000069
	COMMON/NCONST/NCONST	00000070
	COMMON/NDATA/NDATA	00000071
	COMMON/NP/NP	00000072
	COMMON/SDEV/SDEV	00000073
	COMMON/XC/XC(MXANC)	00000074
C		00000075
C		00000076
C	*****	00000077
C*		*00000078
C*	ZERO THE COEFFICIENT MATRIX AND SOLUTION VECTOR	*00000079
C*		*00000080
C	*****	00000081
C		00000082
C		00000083
	ROWMAX=NP+NCONST	00000084
	COLMAX=ROWMAX+1	00000085
	DO 101 ROW=0, ROWMAX	00000086
	DO 101 COL=0, COLMAX	00000087
101	C(ROW, COL)=0.0D+00	00000088
C		00000089
C		00000090
C	*****	00000091
C*		*00000092
C*	CALCULATE THE COEFFICIENT MATRIX	*00000093
C*		*00000094
C	*****	00000095
C		00000096
C		00000097
	REWIND 1	00000098
C		00000099
C		00000100
C	*****	00000101
C*		*00000102
C*	MINIMIZATION EQUATIONS - COEFFICIENTS OF THE B(J)	*00000103
C*		*00000104
C	*****	00000105
C		00000106
C		00000107
102	READ(UNIT=1, END=105, ERR=113) X, Y	00000108
	DO 104 JJ=0, NP	00000109
	ROW=JJ	00000110
	DO 103 J=0, NP	00000111
	COL=J	00000112
103	C(ROW, COL)=C(ROW, COL)+POWER(X, J+JJ)	00000113
C		00000114
C		00000115
C	*****	00000116
C*		*00000117
C*	VECTOR OF CONSTANTS	*00000118
C*		*00000119
C	*****	00000120

C		00000121
C		00000122
104	C (ROW, COLMAX) = C (ROW, COLMAX) + Y * POWER (X, JJ)	00000123
	GO TO 102	00000124
C		00000125
C		00000126
C	*****	00000127
C*		*00000128
C*	COEFFICIENTS OF THE LAMBDA(L)	*00000129
C*		*00000130
C	*****	00000131
C		00000132
C		00000133
105	IF (NCONST.EQ.0) GO TO 108	00000134
	DO 106 JJ=0, NP	00000135
	ROW=JJ	00000136
	F1=FACTRL (JJ)	00000137
	DO 106 L=1, NCONST	00000138
	IF (JJ.LT.NC(L)) GO TO 106	00000139
	COL=NP+L	00000140
	F2=F1/FACTRL (JJ-NC(L)) / 2.0D+00	00000141
	C (ROW, COL) = F2 * POWER (XC(L), JJ-NC(L))	00000142
106	CONTINUE	00000143
C		00000144
C		00000145
C	*****	00000146
C*		*00000147
C*	CONSTRAINT EQUATIONS - COEFFICIENTS OF THE B(J)	*00000148
C*		*00000149
C	*****	00000150
C		00000151
C		00000152
	DO 107 L=1, NCONST	00000153
	ROW=NP+L	00000154
	C (ROW, COLMAX) = CV (L)	00000155
	DO 107 J=NC(L), NP	00000156
	COL=J	00000157
	F1=FACTRL (J) / FACTRL (J-NC(L))	00000158
107	C (ROW, COL) = F1 * POWER (XC(L), J-NC(L))	00000159
C		00000160
C		00000161
C	*****	00000162
C*		*00000163
C*	COMPUTE THE SOLUTION VECTOR	*00000164
C*		*00000165
C	*****	00000166
C		00000167
C		00000168
108	CALL GJEMPS (1.0D-20, +1, ROWMAX+1, DETER, *112)	00000169
	IF (DETER.EQ.0.0D+00) GO TO 114	00000170
C		00000171
C		00000172
C	*****	00000173
C*		*00000174
C*	COMPUTE THE STANDARD DEVIATION	*00000175
C*		*00000176
C	*****	00000177
C		00000178
C		00000179
	SDEV=0.0D+00	00000180

	IDENOM=NDATA-(NP+1)-NCONST-1	00000181
	IF (IDENOM.LE.0) RETURN	00000182
	DENOM=IDENOM	00000183
	REWIND 1	00000184
109	READ (UNIT=1,END=111,ERR=113) X,Y	00000185
	PX=0.0D+00	00000186
	DO 110 J=NP,1,-1	00000187
110	PX=(PX+B(J))*X	00000188
	PX=PX+B(0)	00000189
	SDEV=SDEV+(Y-PX)**2	00000190
	GO TO 109	00000191
111	SDEV=DSQRT(SDEV/DENOM)	00000192
	RETURN	00000193
	C	00000194
	C	00000195
	C*****	00000196
	C*	*00000197
	C* ERROR MESSAGES	*00000198
	C*	*00000199
	C*****	00000200
	C	00000201
	C	00000202
112	WRITE (UNIT=6,FMT=201)	00000203
	RETURN 1	00000204
113	WRITE (UNIT=6,FMT=202)	00000205
	RETURN 1	00000206
114	WRITE (UNIT=6,FMT=203)	00000207
	RETURN 1	00000208
	C	00000209
	C	00000210
	C*****	00000211
	C*	*00000212
	C* FORMAT STATEMENTS	*00000213
	C*	*00000214
	C*****	00000215
	C	00000216
	C	00000217
201	FORMAT('SUBROUTINE GJEMPS WAS CALLED FROM SUBROUTINE LSPRWC')	00000218
202	FORMAT(/,'ERROR IN SUBROUTINE LSPRWC: UNFORMATTED READ ERROR ON UN	00000219
	>IT = 1')	00000220
203	FORMAT('SUBROUTINE GJEMPS WAS CALLED FROM SUBROUTINE LSPRWC')	00000221
	END	00000222


```

SUBROUTINE PES(*)                                00000001
C                                                    00000002
C                                                    00000003
C*****00000004
C*                                                    00000005
C*          POLYNOMIAL EVALUATION SUBROUTINE (PES)    00000006
C*                                                    00000007
C*          REVISION DATE: 3 MAY 2002                00000008
C*                                                    00000009
C*****00000010
C                                                    00000011
C*****00000012
C*                                                    00000013
C* GIVEN X AND THE COEFFICIENTS B(0),B(1),...,B(NP), SUBROUTINE PES 00000014
C* EVALUATES THE K-TH DERIVATIVE OF POLYNOMIALS OF THE FORM:    00000015
C*                                                    00000016
C*           $P(X)=B(0)+B(1)*X+B(2)*X**2+...+B(NP)*X**NP$  00000017
C*                                                    00000018
C* INPUT VARIABLES:                                         00000019
C*                                                    00000020
C* B      = ARRAY OF POLYNOMIAL COEFFICIENTS             00000021
C* NP     = POLYNOMIAL ORDER                             00000022
C* X      = X VALUE FOR EVALUATION OF THE POLYNOMIAL      00000023
C*                                                    00000024
C* OUTPUT VARIABLES:                                       00000025
C*                                                    00000026
C* R      = VALUE AT X OF THE RADIUS OF CURVATURE OF THE POLYNOMIAL 00000027
C* S      = VALUE AT X OF THE APPROXIMATE ARC LENGTH ALONG THE 00000028
C*          POLYNOMIAL                                     00000029
C* YK     = VALUE AT X OF THE K-TH DERIVATIVE OF THE POLYNOMIAL 00000030
C*                                                    00000031
C* PARAMETERS:                                             00000032
C*                                                    00000033
C* MXNR   = MAXIMUM NUMBER OF ROWS                       00000034
C* MXNRM1 = MAXIMUM NUMBER OF ROWS MINUS 1 (MXNRM1 = MXNR - 1) 00000035
C*                                                    00000036
C*****00000037
C                                                    00000038
C                                                    00000039
C          IMPLICIT REAL*8(A-H,O-Z)                     00000040
C                                                    00000041
C          INTEGER FACTRL                                00000042
C                                                    00000043
C          PARAMETER (MXNRM1=29)                         00000044
C                                                    00000045
C          COMMON/B/B(0:MXNRM1)                          00000046
C          COMMON/NP/NP                                    00000047
C                                                    00000048
C          DIMENSION YK(0:9)                             00000049
C                                                    00000050
C                                                    00000051
C          REWIND 2                                       00000052
C          REWIND 3                                       00000053
C          S=0.0E+00                                       00000054
C          N=0                                             00000055
101 READ(UNIT=2,END=106,ERR=107)X                       00000056
C          N=N+1                                          00000057
C                                                    00000058
C                                                    00000059
C*****00000060

```

C*		*00000061
C*	EVALUATE THE NK-TH DERIVATIVE OF THE POLYNOMIAL (0 TO NP)	*00000062
C*		*00000063
C*****		*00000064
C		00000065
C		00000066
	DO 103 NK=0,NP	00000067
	YK(NK)=0.0D+00	00000068
	DO 102 J=NP,NK+1,-1	00000069
102	YK(NK)=(YK(NK)+FACTRL(J)*B(J)/FACTRL(J-NK))*X	00000070
103	YK(NK)=YK(NK)+FACTRL(NK)*B(NK)	00000071
C		00000072
C		00000073
C*****		*00000074
C*		*00000075
C*	EVALUATE THE RADIUS OF CURVATURE OF THE POLYNOMIAL	*00000076
C*		*00000077
C*****		*00000078
C		00000079
C		00000080
	R=1.0E+32	00000081
	IF(NP.LT.2) GO TO 104	00000082
	IF(YK(2).EQ.0.0E+00) GO TO 104	00000083
	R=((1.0E+00+YK(1)*YK(1))**1.5)/DABS(YK(2))	00000084
104	CONTINUE	00000085
C		00000086
C		00000087
C*****		*00000088
C*		*00000089
C*	CALCULATE THE ARC LENGTH	*00000090
C*		*00000091
C*****		*00000092
C		00000093
C		00000094
C		00000095
	IF(N.EQ.1) GO TO 105	00000096
	DeltaX=X-Xsave	00000097
	DeltaY=YK(0)-Ysave	00000098
	DeltaS=DSQRT(DeltaX*DeltaX+DeltaY*DeltaY)	00000099
	S=S+DeltaS	00000100
105	Xsave=X	00000101
	Ysave=YK(0)	00000102
	WRITE(UNIT=3,ERR=108)X,(YK(K),K=0,NP),R,S	00000103
	GO TO 101	00000104
C		00000105
C		00000106
106	END FILE 3	00000107
	RETURN	00000108
C		00000109
C		00000110
C*****		*00000111
C*		*00000112
C*	ERROR MESSAGES	*00000113
C*		*00000114
C*****		*00000115
C		00000116
C		00000117
107	WRITE(UNIT=6,FMT=201)	00000118
	RETURN 1	00000119
108	WRITE(UNIT=6,FMT=202)	00000120

RETURN 1	00000121
C	00000122
C	00000123
C*****	00000124
C*	*00000125
C* FORMAT STATEMENTS	*00000126
C*	*00000127
C*****	00000128
C	00000129
C	00000130
201 FORMAT(/,'ERROR IN SUBROUTINE PES: UNFORMATTED READ ERROR ON UNIT	00000131
>= 2')	00000132
202 FORMAT(/,'ERROR IN SUBROUTINE PES: UNFORMATTED WRITE ERROR ON UNIT	00000133
> = 3')	00000134
END	00000135

```

SUBROUTINE PODS(*)                                00000001
C                                                    00000002
C                                                    00000003
C*****00000004
C*                                                    00000005
C*          POLYNOMIAL ORDER DEFINITION SUBROUTINE (PODS) 00000006
C*                                                    00000007
C*          REVISION DATE: 11 SEPTEMBER 2007          00000008
C*                                                    00000009
C*****00000010
C                                                    00000011
C*****00000012
C*                                                    00000013
C* SUBROUTINE PODS DEFINES THE POLYNOMIAL ORDER FOR PROGRAM LSPRWC. 00000014
C*                                                    00000015
C* INPUT VARIABLES:                                     00000016
C*                                                    00000017
C* NDATA = NUMBER OF X,Y DATA POINTS TO BE FIT      00000018
C* NCONST = NUMBER OF CONSTRAINTS                     00000019
C*                                                    00000020
C* INPUT/OUTPUT VARIABLES:                             00000021
C*                                                    00000022
C* NP      = POLYNOMIAL ORDER                         00000023
C*                                                    00000024
C*****00000025
C                                                    00000026
C*****00000027
C*                                                    00000028
C* PARAMETERS:                                         00000029
C*                                                    00000030
C* MXNAC = MAXIMUM NUMBER OF ALLOWABLE CONSTRAINTS    00000031
C*                                                    00000032
C*****00000033
C                                                    00000034
C                                                    00000035
C          IMPLICIT REAL*8 (A-H,O-Z)                  00000036
C                                                    00000037
C          PARAMETER (MXNAC=20,MXNR=30,MXNRM1=29)      00000038
C                                                    00000039
C          COMMON/NC/NC(MXNAC)                         00000040
C          COMMON/NCONST/NCONST                       00000041
C          COMMON/NDATA/NDATA                         00000042
C          COMMON/NP/NP                               00000043
C                                                    00000044
C          DATA NP/3/                                 00000045
C                                                    00000046
C          NAMELIST/PARM/NP                           00000047
C                                                    00000048
C                                                    00000049
C*****00000050
C*                                                    00000051
C*          INPUT THE POLYNOMIAL ORDER                 00000052
C*                                                    00000053
C*****00000054
C                                                    00000055
C                                                    00000056
C          NPMAX=MIN0(9,NDATA-NCONST-1)               00000057
101 WRITE(UNIT=6,FMT=201)NPMAX                       00000058
CALL NDWUS('PARM',6,*103)                            00000059
CALL NDWUSI('NP',1,1,1,NP,*103)                     00000060

```

	CALL NDWUSE(*103)	00000061
	CALL NDRUS('PARM',4,*101,*104)	00000062
	READ(UNIT=4,NML=PARM,ERR=101)	00000063
	CALL NDRUSE(*101,*104)	00000064
	IF(NP.LT.0.OR.NP.GT.9) GO TO 105	00000065
	IF(NP.GT.NPMAX) GO TO 106	00000066
	DO 102 I=1,NCONST	00000067
	IF(NC(I).LT.0.OR.NC(I).GT.NP) GO TO 107	00000068
102	CONTINUE	00000069
	RETURN	00000070
C		00000071
C		00000072
C	*****	00000073
C*		*00000074
C*	ERROR MESSAGES	*00000075
C*		*00000076
C	*****	00000077
C		00000078
C		00000079
103	WRITE(UNIT=6,FMT=204)	00000080
	GO TO 108	00000081
104	WRITE(UNIT=6,FMT=205)	00000082
	GO TO 108	00000083
105	WRITE(UNIT=6,FMT=206)NP	00000084
	GO TO 108	00000085
106	WRITE(UNIT=6,FMT=207)NP,NCONST,NDATA,NPMAX	00000086
	GO TO 108	00000087
107	WRITE(UNIT=6,FMT=208)NC(I),I,NP	00000088
108	WRITE(UNIT=6,FMT=202)	00000089
	READ(UNIT=5,FMT=203)CCV	00000090
	RETURN 1	00000091
C		00000092
C		00000093
C	*****	00000094
C*		*00000095
C*	FORMAT STATEMENTS	*00000096
C*		*00000097
C	*****	00000098
C		00000099
C		00000100
201	FORMAT(/,'INPUT THE POLYNOMIAL ORDER BY NAMELIST WHERE:',//,'NP	00000101
	> = THE POLYNOMIAL ORDER',//,'NOTE: NP MUST BE IN THE RANGE OF 0 TO	00000102
	> 9.',//,T7,'THERE MUST BE AT LEAST NP+NCONST+1 X-Y DATA POINTS.',//,	00000103
	>T7,'THE MAXIMUM VALUE FOR NP IS ',I5,'.',//,'CURRENT VALUES ARE:')	00000104
202	FORMAT(/,T19,'- ENTER/RETURN TO CONTINUE -')	00000105
203	FORMAT(A1)	00000106
204	FORMAT('SUBROUTINE NDWUS WAS CALLED FROM SUBROUTINE PODS')	00000107
205	FORMAT('SUBROUTINE NDRUS WAS CALLED FROM SUBROUTINE PODS')	00000108
206	FORMAT(/,'ERROR IN SUBROUTINE PODS: NP = ',I5,' IS NOT ALLOWED.',/	00000109
	>,'THE POLYNOMIAL ORDER MUST BE GREATER THAN OR EQUAL TO 0 AND LESS	00000110
	> THAN OR EQUAL',//,'TO 9.')	00000111
207	FORMAT(/,'ERROR IN SUBROUTINE PODS: NP = ',I5,' IS NOT ALLOWED FOR	00000112
	> NCONST = ',I5,' AND',//,'NDATA = ',I5,'. THERE MUST BE AT LEAST N	00000113
	>P+2*NCONST X-Y DATA POINTS. THE',//,'MAXIMUM VALUE FOR NP IS ',I5,	00000114
	>'.'')	00000115
208	FORMAT(/,'ERROR IN SUBROUTINE PODS: NC(',I3,') = ',I5,' IS NOT ALL	00000116
	>OWED FOR NP = ',I5,'.',//,'THE CONSTRAINT ORDER MUST BE GREATER THA	00000117
	>N OR EQUAL TO 0 AND LESS THAN OR EQUAL TO NP.')	00000118
	END	00000119

FUNCTION POWER(X,N)	00000001
C	00000002
C	00000003
C*****	00000004
C*	*00000005
C* INTEGER POWERS OF REAL NUMBERS FUNCTION SUBPROGRAM (POWER)	*00000006
C*	*00000007
C* REVISION DATE: 15 JUNE 1989	*00000008
C*	*00000009
C*****	00000010
C	00000011
C	00000012
REAL*8 POWER,X	00000013
POWER=1.0D+00	00000014
IF(N.EQ.0) RETURN	00000015
DO 101 I=1,N	00000016
101 POWER=POWER*X	00000017
END	00000018

```

SUBROUTINE QPS(*)                                00000001
C                                                    00000002
C                                                    00000003
C*****00000004
C*                                                    *00000005
C*                QUICK PLOT SUBROUTINE (QPS)          *00000006
C*                                                    *00000007
C*                REVISION DATE: 1 JULY 2011          *00000008
C*                                                    *00000009
C*****00000010
C                                                    00000011
C*****00000012
C*                                                    *00000013
C*    SUBROUTINE QPS WRITES DATA FILES TO CONSTRUCT A QUICK PLOT OF *00000014
C*    RESULTS FOR PROGRAM LSPRWC USING THE PostScript CARTESIAN *00000015
C*    COORDINATE PLOT PROGRAM PSSCPP.                *00000016
C*                                                    *00000017
C* INPUT VARIABLES:                                *00000018
C*                                                    *00000019
C* NCONST = NUMBER OF CONSTRAINTS                    *00000020
C* X      = X VALUE FOR EVALUATION OF THE POLYNOMIAL *00000021
C* XC     = ARRAY OF CONSTRAINT X VALUES            *00000022
C* Y      = Y VALUE FROM EVALUATION OF THE POLYNOMIAL AT X *00000023
C*                                                    *00000024
C* PARAMETERS:                                    *00000025
C*                                                    *00000026
C* MXNAC  = MAXIMUM NUMBER OF ALLOWABLE CONSTRAINTS *00000027
C*                                                    *00000028
C*****00000029
C                                                    00000030
C                                                    00000031
C    IMPLICIT REAL*8(A-H,O-Z)                        00000032
C                                                    00000033
C    CHARACTER*10 File                               00000034
C                                                    00000035
C    PARAMETER (MXNAC=20,One=1.0E+00,Zero=0.0E+00)  00000036
C                                                    00000037
C    COMMON/NCONST/NCONST                            00000038
C    COMMON/XC/XC(MXNAC)                            00000039
C                                                    00000040
C    DIMENSION File(3)                               00000041
C                                                    00000042
C    DATA File/'@lsprwc.cl','@lsprwc.ed','@lsprwc.dp'/,Norm/0/ 00000043
C                                                    00000044
C    NAMELIST/PAARM/Norm,Xmax,Xmin,Ymax,Ymin         00000045
C                                                    00000046
C    FNX(Q)=(Q-Xmin)/(Xmax-Xmin)                     00000047
C    FNY(Q)=(Q-Ymin)/(Ymax-Ymin)                     00000048
C                                                    00000049
C                                                    00000050
C*****00000051
C*                                                    *00000052
C*                FIND THE RANGE OF THE DATA          *00000053
C*                                                    *00000054
C*****00000055
C                                                    00000056
C                                                    00000057
C    Xmin=+1.0D+75                                    00000058
C    Xmax=-1.0D+75                                    00000059
C    Ymin=+1.0D+75                                    00000060

```

	Ymax=-1.0D+75	00000061
	REWIND 1	00000062
101	READ(UNIT=1,END=102,ERR=116)X,Y	00000063
	Xmin=DMIN1(X,Xmin)	00000064
	Xmax=DMAX1(X,Xmax)	00000065
	Ymin=DMIN1(Y,Ymin)	00000066
	Ymax=DMAX1(Y,Ymax)	00000067
	GO TO 101	00000068
102	REWIND 3	00000069
103	READ(UNIT=3,END=104,ERR=117)X,Y	00000070
	Xmin=DMIN1(X,Xmin)	00000071
	Xmax=DMAX1(X,Xmax)	00000072
	Ymin=DMIN1(Y,Ymin)	00000073
	Ymax=DMAX1(Y,Ymax)	00000074
	GO TO 103	00000075
104	IF(NCONST.EQ.0) GO TO 106	00000076
	DO 105 I=1,NCONST	00000077
	Xmin=DMIN1(XC(I),Xmin)	00000078
105	Xmax=DMAX1(XC(I),Xmax)	00000079
106	IF(Xmin.GE.Xmax) RETURN	00000080
	IF(Ymin.GE.Ymax) RETURN	00000081
107	WRITE(UNIT=6,FMT=201)	00000082
	CALL NDWUS('PARM',6,*118)	00000083
	CALL NDWUSI('Norm',1,1,1,Norm,*118)	00000084
	CALL NDWUSD('Xmax',1,1,1,Xmax,*118)	00000085
	CALL NDWUSD('Xmin',1,1,1,Xmin,*118)	00000086
	CALL NDWUSD('Ymax',1,1,1,Ymax,*118)	00000087
	CALL NDWUSD('Ymin',1,1,1,Ymin,*118)	00000088
	CALL NDWUSE(*118)	00000089
	CALL NDRUS('PARM',4,*107,*119)	00000090
	READ(UNIT=4,NML=PARM,ERR=107)	00000091
	CALL NDRUSE(*107,*119)	00000092
C		00000093
C		00000094
C*****		00000095
C*		*00000096
C*	MARK THE CONSTRAINT LOCATIONS	*00000097
C*		*00000098
C*****		00000099
C		00000100
C		00000101
	N=1	00000102
	OPEN(UNIT=4,FILE=File(N),STATUS='UNKNOWN',FORM='FORMATTED',	00000103
	>ERR=120)	00000104
	IF(NCONST.EQ.0) GO TO 109	00000105
	DO 108 I=1,NCONST	00000106
	IF(Norm.EQ.0) WRITE(UNIT=4,FMT=202,ERR=121)XC(I),Ymin,INT(Zero)	00000107
	IF(Norm.EQ.1) WRITE(UNIT=4,FMT=202,ERR=121)FNX(XC(I)),Zero,	00000108
	>INT(Zero)	00000109
	IF(Norm.EQ.0) WRITE(UNIT=4,FMT=202,ERR=121)XC(I),Ymax,INT(One)	00000110
	IF(Norm.EQ.1) WRITE(UNIT=4,FMT=202,ERR=121)FNX(XC(I)),One,	00000111
	>INT(One)	00000112
108	CONTINUE	00000113
	GO TO 110	00000114
109	WRITE(UNIT=4,FMT=202,ERR=121)Zero,Zero,INT(Zero)	00000115
110	CLOSE(UNIT=4,STATUS='KEEP',ERR=122)	00000116
C		00000117
C		00000118
C*****		00000119
C*		*00000120


```

C*                               WRITE THE EVALUATED DATA                               *00000121
C*                               *00000122
C*****00000123
C                                00000124
C                                00000125
C                                00000126
      N=2
      OPEN(UNIT=4,FILE=File(N),STATUS='UNKNOWN',FORM='FORMATTED',
>ERR=120)                                00000127
111  REWIND 3                            00000128
112  READ(UNIT=3,END=113,ERR=117)X,Y      00000129
      IF(Norm.EQ.0) WRITE(UNIT=4,FMT=202,ERR=121)X,Y      00000130
      IF(Norm.EQ.1) WRITE(UNIT=4,FMT=202,ERR=121)FNX(X),FNY(Y) 00000131
      GO TO 112                                00000132
113  CLOSE(UNIT=4,STATUS='KEEP',ERR=122)  00000133
C                                           00000134
C                                           00000135
C*****00000136
C*                               *00000137
C*                               *00000138
C*                               *00000139
C*                               *00000140
C*****00000141
C                                00000142
C                                00000143
C                                00000144
      N=3
      OPEN(UNIT=4,FILE=File(N),STATUS='UNKNOWN',FORM='FORMATTED',
>ERR=120)                                00000145
      REWIND 1                                00000146
114  READ(UNIT=1,END=115,ERR=116)X,Y      00000147
      IF(Norm.EQ.0) WRITE(UNIT=4,FMT=202,ERR=121)X,Y      00000148
      IF(Norm.EQ.1) WRITE(UNIT=4,FMT=202,ERR=121)FNX(X),FNY(Y) 00000149
      GO TO 114                                00000150
115  CLOSE(UNIT=4,STATUS='KEEP',ERR=122)  00000151
C                                           00000152
C                                           00000153
C                                           00000154
C*****00000155
C*                               *00000156
C*                               *00000157
C*                               *00000158
C*****00000159
C                                00000160
C                                00000161
      RETURN                                00000162
C                                           00000163
C                                           00000164
C*****00000165
C*                               *00000166
C*                               *00000167
C*                               *00000168
C*****00000169
C                                00000170
C                                00000171
116  WRITE(UNIT=6,FMT=205)                00000172
      RETURN 1                                00000173
117  WRITE(UNIT=6,FMT=206)                00000174
      RETURN 1                                00000175
118  WRITE(UNIT=6,FMT=207)                00000176
      RETURN 1                                00000177
119  WRITE(UNIT=6,FMT=208)                00000178
      RETURN 1                                00000179
120  WRITE(UNIT=6,FMT=209)File(N)         00000180

```

	RETURN 1	00000181
121	WRITE(UNIT=6,FMT=210)File(N)	00000182
	RETURN 1	00000183
122	WRITE(UNIT=6,FMT=211)File(N)	00000184
	RETURN 1	00000185
	C	00000186
	C	00000187
	C*****	00000188
	C*	*00000189
	C* FORMAT STATEMENTS	*00000190
	C*	*00000191
	C*****	00000192
	C	00000193
	C	00000194
201	FORMAT(/,'INPUT THE PLOT VARIABLES BY NAMELIST WHERE:',//,'Norm	00000195
	>= RENormILIZATION CONTROL',/,T8,'= 0, TO NOT NormalIZE THE PLOT DA	00000196
	>TA',/,T8,'= 1, TO NormalIZE THE PLOT DATA',/, 'Xmax = MAXIMUM X-V	00000197
	>ALUE',/, 'Xmin = MINIMUM X-VALUE',/, 'Ymax = MAXIMUM Y-VALUE',/,	00000198
	>'Ymin = MINIMUM Y-VALUE',//,'CURRENT VALUES ARE:')	00000199
202	FORMAT(2E15.6,I15)	00000200
203	FORMAT('lsprwc.l')	00000201
204	FORMAT('lsprwc.ps')	00000202
205	FORMAT(/,'ERROR IN SUBROUTINE QPS: UNFORMATTED READ ERROR ON UNIT1	00000203
	>')	00000204
206	FORMAT(/,'ERROR IN SUBROUTINE QPS: UNFORMATTED READ ERROR ON UNIT3	00000205
	>')	00000206
207	FORMAT('SUBROUTINE NDWUS WAS CALLED FROM SUBROUTINE QPS')	00000207
208	FORMAT('SUBROUTINE NDRUS WAS CALLED FROM SUBROUTINE QPS')	00000208
209	FORMAT(/,'ERROR IN SUBROUTINE QPS: OPEN ERROR ON UNIT 4, STATUS =	00000209
	>"UNKNOWN", FILE =',A10)	00000210
210	FORMAT(/,'ERROR IN SUBROUTINE QPS: WRITE ERROR ON UNIT 4, FILE = '	00000211
	>,A10)	00000212
211	FORMAT(/,'ERROR IN SUBROUTINE QPS: CLOSE ERROR ON UNIT 4, FILE =',	00000213
	>A10)	00000214
	END	00000215

```

SUBROUTINE XDPEDS(LCV2,*,*)                                00000001
C                                                            00000002
C                                                            00000003
C*****00000004
C*00000005
C*    X DATA POINTS FOR EVALUATION DEFINITION SUBROUTINE (XDPEDS) 00000006
C*00000007
C*    REVISION DATE: 1 JULY 2011                                00000008
C*00000009
C*****00000010
C00000011
C*****00000012
C*00000013
C*    SUBROUTINE XDPEDS DEFINES THE X COORDINATE SET FOR EVALUATION OF 00000014
C*    THE RESULTANT LEAST SQUARES POLYNOMIAL SET.                00000015
C*00000016
C*    INPUT/OUTPUT VARIABLES:                                     00000017
C*00000018
C*    X      = ARRAY OF X COORDINATES FOR EVALUATION OF THE LEAST SQUARES 00000019
C*    POLYNOMIAL SET                                             00000020
C*00000021
C*    OUTPUT VARIABLES:                                          00000022
C*00000023
C*    NEVAL   = NUMBER OF X COORDINATES FOR EVALUATION          00000024
C*00000025
C*    PARAMETERS:                                                00000026
C*00000027
C*    MNXCE   = MAXIMUM NUMBER OF X COORDINATES FOR POLYNOMIAL EVALUATION 00000028
C*00000029
C*****00000030
C00000031
C00000032
C    IMPLICIT REAL*8(A-H,O-Z)                                00000033
C00000034
C    CHARACTER CCV*1,DFmt*3,File*80,Fmt*80                    00000035
C00000036
C    PARAMETER (R_NaN=-999.E+00)                                00000037
C    PARAMETER (MNXCE=101)                                       00000038
C00000039
C    COMMON/NEVAL/NEVAL                                          00000040
C    COMMON/XINT/XINT                                            00000041
C    COMMON/XMAX/XMAX                                            00000042
C    COMMON/XMIN/XMIN                                            00000043
C00000044
C    DIMENSION X(MNXCE)                                          00000045
C00000046
C    DATA DFmt/'(')'/,NEVAL/0/                                  00000047
C00000048
C    NAMELIST/PARM/X,XINT,XMAX,XMIN                             00000049
C00000050
C00000051
C*****00000052
C*00000053
C*    INPUT THE X DATA POINTS FOR EVALUATION                    00000054
C*    OF THE LEAST SQUARES POLYNOMIAL                            00000055
C*00000056
C*****00000057
C00000058
C00000059
C    IF(NEVAL.EQ.0) WRITE(UNIT=6,FMT=201)                      00000060

```

IF (NEVAL.NE.0) WRITE (UNIT=6,FMT=202)	00000061
READ (UNIT=5,FMT=203)LCV1	00000062
REWIND 2	00000063
LCV2=1	00000064
GO TO (101,107,112,116),LCV1	00000065
C	00000066
C	00000067
C*****	00000068
C*	*00000069
C* NAMELIST INPUT	*00000070
C*	*00000071
C*****	00000072
C	00000073
C	00000074
101 DO 102 I=1,MNXCE	00000075
102 X(I)=R_NaN	00000076
103 WRITE (UNIT=6,FMT=204)MNXCE	00000077
WRITE (UNIT=6,FMT=205)	00000078
CALL NDRUS (' PARM' ,4,*103,*117)	00000079
READ (UNIT=4,NML=PARM,ERR=103)	00000080
CALL NDRUSE (*103,*117)	00000081
DO 104 NEVAL=MNXCE,1,-1	00000082
IF (X (NEVAL) .NE.R_NaN) GO TO 105	00000083
104 CONTINUE	00000084
105 DO 106 I=1,NEVAL	00000085
106 WRITE (UNIT=2,ERR=122)X(I)	00000086
GO TO 115	00000087
C	00000088
C	00000089
C*****	00000090
C*	*00000091
C* FORMATTED DISK FILE INPUT	*00000092
C*	*00000093
C*****	00000094
C	00000095
C	00000096
107 WRITE (UNIT=6,FMT=206)	00000097
WRITE (UNIT=6,FMT=207)	00000098
READ (UNIT=5,FMT=208)File	00000099
108 WRITE (UNIT=6,FMT=209)	00000100
WRITE (UNIT=6,FMT=210)	00000101
CALL DFCUS (FMT,DFmt,*108)	00000102
OPEN (UNIT=4,FILE=File,STATUS=' OLD' ,ERR=118)	00000103
REWIND 4	00000104
NEVAL=0	00000105
IF (Fmt(1:3) .EQ.DFmt) GO TO 110	00000106
109 READ (UNIT=4,FMT=Fmt,END=111,ERR=119)X(1)	00000107
WRITE (UNIT=2,ERR=122)X(1)	00000108
NEVAL=NEVAL+1	00000109
GO TO 109	00000110
110 READ (UNIT=4,FMT=*,END=111,ERR=119)X(1)	00000111
WRITE (UNIT=2,ERR=122)X(1)	00000112
NEVAL=NEVAL+1	00000113
GO TO 110	00000114
111 CLOSE (UNIT=4,STATUS=' KEEP' ,ERR=120)	00000115
GO TO 115	00000116
C	00000117
C	00000118
C*****	00000119
C*	*00000120

C*	MINIMUM, MAXIMUM, AND INTERVAL VALUES	*00000121
C*		*00000122
C*****		*00000123
C		00000124
C		00000125
112	WRITE(UNIT=6,FMT=211)	00000126
	CALL NDWUS('PARM',6,*121)	00000127
	CALL NDWUSD('XINT',1,1,1,XINT,*121)	00000128
	CALL NDWUSD('XMAX',1,1,1,XMAX,*121)	00000129
	CALL NDWUSD('XMIN',1,1,1,XMIN,*121)	00000130
	CALL NDWUSE(*121)	00000131
	CALL NDRUS('PARM',4,*112,*117)	00000132
	READ(UNIT=4,NML=PARM,ERR=112)	00000133
	CALL NDRUSE(*112,*117)	00000134
	NEVAL=0	00000135
	X(1)=XMIN	00000136
	DUM1=XMAX+XINT/2	00000137
113	IF(X(1).GT.DUM1) GO TO 114	00000138
	WRITE(UNIT=2,ERR=122)X(1)	00000139
	NEVAL=NEVAL+1	00000140
	DUM3=X(1)	00000141
	X(1)=XMIN+NEVAL*XINT	00000142
	GO TO 113	00000143
114	IF(DUM3.EQ.XMAX) GO TO 115	00000144
	WRITE(UNIT=2,ERR=122)XMAX	00000145
	NEVAL=NEVAL+1	00000146
115	END FILE 2	00000147
116	RETURN	00000148
C		00000149
C		00000150
C*****		*00000151
C*		*00000152
C*	ERROR MESSAGES	*00000153
C*		*00000154
C*****		*00000155
C		00000156
C		00000157
117	WRITE(UNIT=6,FMT=212)	00000158
	GO TO 124	00000159
118	WRITE(UNIT=6,FMT=213)	00000160
	CALL CBUS06(6,File,*123)	00000161
	RETURN 1	00000162
119	WRITE(UNIT=6,FMT=214)	00000163
	CALL CBUS06(6,File,*123)	00000164
	RETURN 1	00000165
120	WRITE(UNIT=6,FMT=215)	00000166
	CALL CBUS06(6,File,*123)	00000167
	RETURN 1	00000168
121	WRITE(UNIT=6,FMT=216)	00000169
	GO TO 124	00000170
122	WRITE(UNIT=6,FMT=217)	00000171
	RETURN 1	00000172
123	WRITE(UNIT=6,FMT=218)	00000173
	RETURN 1	00000174
124	WRITE(UNIT=6,FMT=219)	00000175
	READ(UNIT=5,FMT=220)CCV	00000176
	RETURN 2	00000177
C		00000178
C		00000179
C*****		*00000180

C*		*00000181
C*	FORMAT STATEMENTS	*00000182
C*		*00000183
C*****		*00000184
C		00000185
C		00000186
201	FORMAT(/,'SELECT THE SOURCE OF INPUT FOR THE X DATA TO BE EVALUATE	00000187
	>D FROM THE FOLLOWING',/,'LIST:',//,'1, FOR KEYBOARD INPUT VIA NAME	00000188
	>LIST',/,'2, FOR FORMATTED DISK FILE INPUT',/,'3, FOR INPUT OF MINI	00000189
	>MUM, MAXIMUM, AND INTERVAL VALUES')	00000190
202	FORMAT(/,'SELECT THE SOURCE OF INPUT FOR THE X DATA TO BE EVALUATE	00000191
	>D FROM THE FOLLOWING',/,'LIST:',//,'1, FOR KEYBOARD INPUT VIA NAME	00000192
	>LIST',/,'2, FOR FORMATTED DISK FILE INPUT',/,'3, FOR INPUT OF MINI	00000193
	>MUM, MAXIMUM, AND INTERVAL VALUES',/,'4, TO USE THE PREVIOUSLY PRE	00000194
	>SCRIBED X VALUES')	00000195
203	FORMAT(I1)	00000196
204	FORMAT(/,'INPUT THE DATA SET X VALUES (' ,I3,' MAX) BY NAMELIST WHE	00000197
	>RE:',//,'X = ARRAY OF X VALUES',//,'CURRENT VALUES ARE:')	00000198
205	FORMAT(/,'\$PARM X=_____,_____,_____\$',/)	00000199
206	FORMAT(/,'DATA SETS ARE INPUT FROM FORMATTED DISK FILES AS X VALUE	00000200
	>S WHERE:',//,'X = X VALUE')	00000201
207	FORMAT(/,'INPUT THE FILE NAME OF THE FORMATTED DISK FILE DATA SET.	00000202
	>')	00000203
208	FORMAT(A80)	00000204
209	FORMAT(/,'INPUT THE DATA FILE FORMAT (INCLUDE PARENTHESES)')	00000205
210	FORMAT('NOTE: AN EXAMPLE FORMAT IS "(E15.6)"',/,'T7,'ENTER "(*)" FO	00000206
	>R A FREE FIELD READ (DEFAULT FORMAT)')	00000207
211	FORMAT(/,'INPUT THE X DATA MAXIMUM, MINIMUM, AND INTERVAL VALUES B	00000208
	>Y NAMELIST WHERE:',//,'XINT = X DATA POINT INTERVAL',/,'XMAX =	00000209
	> MAXIMUM X DATA VALUE',/,'XMIN = MINIMUM X DATA VALUE',//,'CURRE	00000210
	>NT VALUES ARE:')	00000211
212	FORMAT('SUBROUTINE NDRUS WAS CALLED FROM PROGRAM LSPRC')	00000212
213	FORMAT(/,'ERROR IN PROGRAM LSPRC: OPEN ERROR ON UNIT 4, STATUS =	00000213
	>"OLD", FILE =',/)	00000214
214	FORMAT(/,'ERROR IN PROGRAM LSPRC: READ ERROR ON UNIT 4, FILE =',/)	00000215
	>)	00000216
215	FORMAT(/,'ERROR IN PROGRAM LSPRC: CLOSE ERROR ON UNIT 4, FILE =',/)	00000217
	>)	00000218
216	FORMAT('SUBROUTINE NDWUS WAS CALLED FROM PROGRAM LSPRC')	00000219
217	FORMAT(/,'ERROR IN PROGRAM LSPRC: UNFORMATTED WRITE ERROR ON UNIT	00000220
	> 2')	00000221
218	FORMAT('SUBROUTINE CBUS06 WAS CALLED FROM PROGRAM LSPRC')	00000222
219	FORMAT(/,'T19,'- ENTER/RETURN TO CONTINUE -')	00000223
220	FORMAT(A1)	00000224
	END	00000225

```

SUBROUTINE XYDIS(*,*)                                00000001
C                                                       00000002
C                                                       00000003
C*****00000004
C*                                                       *00000005
C*               X-Y DATA INPUT SUBROUTINE (XYDIS)      *00000006
C*                                                       *00000007
C*               REVISION DATE: 1 JULY 2011              *00000008
C*                                                       *00000009
C*****00000010
C                                                       00000011
C*****00000012
C*                                                       *00000013
C* SUBROUTINE XYDIS DEFINES THE X,Y DATA POINTS TO BE FIT FOR PROGRAM *00000014
C* LSPRWC.                                                *00000015
C*                                                       *00000016
C* INPUT/OUTPUT VARIABLES:                               *00000017
C*                                                       *00000018
C* X      = ARRAY OF X COORDINATES                       *00000019
C* Y      = ARRAY OF Y COORDINATES                       *00000020
C*                                                       *00000021
C* OUTPUT VARIABLES:                                    *00000022
C*                                                       *00000023
C* NDATA  = NUMBER OF X,Y DATA POINTS TO BE FIT        *00000024
C*                                                       *00000025
C* PARAMETERS:                                           *00000026
C*                                                       *00000027
C* MNXYDP = MAXIMUM NUMBER OF (X,Y) DATA PAIRS FOR NAMELIST INPUT *00000028
C*                                                       *00000029
C*****00000030
C                                                       00000031
C                                                       00000032
C      IMPLICIT REAL*8(A-H,O-Z)                          00000033
C                                                       00000034
C      CHARACTER CCV*1,Crdimg*80,DFmt*3,File*80,Fmt*80,Fmt_s*80 00000035
C                                                       00000036
C      PARAMETER (R_NaN=-999.E+00)                       00000037
C      PARAMETER (MNXYDP=101)                             00000038
C                                                       00000039
C      COMMON/NDATA/NDATA                                 00000040
C      COMMON/XINT/XINT                                   00000041
C      COMMON/XMAX/XMAX                                   00000042
C      COMMON/XMIN/XMIN                                   00000043
C                                                       00000044
C      DIMENSION X(MNXYDP),Y(MNXYDP)                     00000045
C                                                       00000046
C      DATA DFmt/'(*)'/'                                  00000047
C                                                       00000048
C      NAMELIST/PARM/X,Y                                  00000049
C                                                       00000050
C                                                       00000051
C*****00000052
C*                                                       *00000053
C*               INPUT THE X-Y DATA POINTS TO BE FITTED *00000054
C*                                                       *00000055
C*****00000056
C                                                       00000057
C                                                       00000058
C      WRITE(UNIT=6,FMT=203)                              00000059
C      READ(UNIT=5,FMT=204)LCV1                           00000060

```

REWIND 1	00000061
IF(LCV1.EQ.2) GO TO 106	00000062
C	00000063
C	00000064
C*****	00000065
C*	*00000066
C* NAMELIST INPUT	*00000067
C*	*00000068
C*****	00000069
C	00000070
C	00000071
DO 101 I=1,MNXYDP	00000072
X(I)=R_NaN	00000073
101 Y(I)=0.0	00000074
102 WRITE(UNIT=6,FMT=205)MNXYDP	00000075
WRITE(UNIT=6,FMT=206)	00000076
CALL NDRUS('PARM',4,*102,*119)	00000077
READ(UNIT=4,NML=PARM,ERR=102)	00000078
CALL NDRUSE(*102,*119)	00000079
DO 103 NDATA=MNXYDP,1,-1	00000080
IF(X(NDATA).NE.R_NaN) GO TO 104	00000081
103 CONTINUE	00000082
104 DO 105 I=1,NDATA	00000083
105 WRITE(UNIT=1,ERR=120)X(I),Y(I)	00000084
GO TO 112	00000085
C	00000086
C	00000087
C*****	00000088
C*	*00000089
C* FORMATTED DISK FILE INPUT	*00000090
C*	*00000091
C*****	00000092
C	00000093
C	00000094
106 WRITE(UNIT=6,FMT=207)	00000095
WRITE(UNIT=6,FMT=208)	00000096
READ(UNIT=5,FMT=209)File	00000097
107 WRITE(UNIT=6,FMT=210)	00000098
WRITE(UNIT=6,FMT=211)	00000099
CALL DFCUS(Fmt,DFmt,*107)	00000100
C	00000101
OPEN(UNIT=4,FILE=File,STATUS='OLD',ERR=121)	00000102
REWIND 4	00000103
NDATA=0	00000104
C	00000105
108 READ(UNIT=4,FMT=209,END=122,ERR=123)Crdimg	00000106
CALL CBUS01(Crdimg,L)	00000107
IF(L.EQ.0) GO TO 108	00000108
IF(Fmt(1:3).NE.DFmt) READ(Crdimg,FMT=Fmt,ERR=108)X(1),Y(1)	00000109
IF(Fmt(1:3).EQ.DFmt) READ(Crdimg,*,ERR=108)X(1),Y(1)	00000110
BACKSPACE 4	00000111
C	00000112
IF(Fmt(1:3).EQ.DFmt) GO TO 110	00000113
109 READ(UNIT=4,FMT=Fmt,END=111,ERR=123)X(1),Y(1)	00000114
WRITE(UNIT=1,ERR=120)X(1),Y(1)	00000115
NDATA=NDATA+1	00000116
GO TO 109	00000117
110 READ(UNIT=4,FMT=*,END=111,ERR=123)X(1),Y(1)	00000118
WRITE(UNIT=1,ERR=120)X(1),Y(1)	00000119
NDATA=NDATA+1	00000120

	GO TO 110	00000121
111	CLOSE(UNIT=4,STATUS='KEEP',ERR=125)	00000122
112	END FILE 1	00000123
C		00000124
C		00000125
C	*****	00000126
C*		*00000127
C*	SEARCH FOR THE MINIMUM AND MAXIMUM X VALUES WITHIN THE DATA SET	*00000128
C*		*00000129
C	*****	00000130
C		00000131
C		00000132
	REWIND 1	00000133
	XMAX=-1.0D+75	00000134
	XMIN=+1.0D+75	00000135
	XSAV=-1.0D+75	00000136
113	READ(UNIT=1,END=114,ERR=124)X(1),Y(1)	00000137
	IF(X(1).LT.XSAV) GO TO 126	00000138
	XMAX=DMAX1(XMAX,X(1))	00000139
	XMIN=DMIN1(XMIN,X(1))	00000140
	XSAV=X(1)	00000141
	GO TO 113	00000142
114	IF(XMAX.EQ.XMIN) GO TO 127	00000143
	XINT=(XMAX-XMIN)/1.0D+02	00000144
C		00000145
C		00000146
C	*****	00000147
C*		*00000148
C*	DISPLAY THE X-Y DATA SET	*00000149
C*		*00000150
C	*****	00000151
C		00000152
C		00000153
115	WRITE(UNIT=6,FMT=212)	00000154
	CALL YNOUS(*116,*118,*115)	00000155
116	REWIND 1	00000156
	WRITE(UNIT=6,FMT=213)	00000157
	DO 117 I=1,NDATA	00000158
	READ(UNIT=1,ERR=124)X(1),Y(1)	00000159
	IF(MOD(I,20).NE.0) GO TO 117	00000160
	WRITE(UNIT=6,FMT=201)	00000161
	READ(UNIT=5,FMT=202)CCV	00000162
	WRITE(UNIT=6,FMT=213)	00000163
117	WRITE(UNIT=6,FMT=214)I,X(1),Y(1)	00000164
	WRITE(UNIT=6,FMT=201)	00000165
	READ(UNIT=5,FMT=202)CCV	00000166
118	RETURN	00000167
C		00000168
C		00000169
C	*****	00000170
C*		*00000171
C*	ERROR MESSAGES	*00000172
C*		*00000173
C	*****	00000174
C		00000175
C		00000176
119	WRITE(UNIT=6,FMT=215)	00000177
	GO TO 129	00000178
120	WRITE(UNIT=6,FMT=216)	00000179
	RETURN 1	00000180

```

121  WRITE(UNIT=6,FMT=217)                                00000181
      CALL CBUS06(6,File,*128)                            00000182
      RETURN 1                                             00000183
122  WRITE(UNIT=6,FMT=218)                                00000184
      CALL CBUS06(6,File,*128)                            00000185
      RETURN 1                                             00000186
123  WRITE(UNIT=6,FMT=219)                                00000187
      CALL CBUS06(6,File,*128)                            00000188
      RETURN 1                                             00000189
124  WRITE(UNIT=6,FMT=220)                                00000190
      RETURN 1                                             00000191
125  WRITE(UNIT=6,FMT=221)                                00000192
      CALL CBUS06(6,File,*128)                            00000193
      RETURN 1                                             00000194
126  WRITE(UNIT=6,FMT=222)                                00000195
      GO TO 129                                           00000196
127  WRITE(UNIT=6,FMT=223)                                00000197
      GO TO 129                                           00000198
128  WRITE(UNIT=6,FMT=224)                                00000199
      RETURN 1                                             00000200
129  WRITE(UNIT=6,FMT=201)                                00000201
      READ(UNIT=5,FMT=202)CCV                             00000202
      RETURN 2                                             00000203
C                                                         00000204
C                                                         00000205
C*****00000206
C*                                                         *00000207
C*                                                         *00000208
C*          FORMAT STATEMENTS                             *00000209
C*****00000210
C                                                         00000211
C                                                         00000212
201  FORMAT(/,T19,'- ENTER/RETURN TO CONTINUE -')        00000213
202  FORMAT(A1)                                           00000214
203  FORMAT(/,'SELECT THE SOURCE OF INPUT FOR THE X-Y DATA TO BE FITTED00000215
> FROM',/, 'THE FOLLOWING LIST:',/, '1, FOR KEYBOARD INPUT VIA NAMEL00000216
>IST',/, '2, FOR FORMATTED DISK FILE INPUT',/, 'NOTE: THE X-Y DATA M00000217
>UST BE MONOTONICALLY INCREASING IN X.')                00000218
204  FORMAT(I1)                                           00000219
205  FORMAT(/,'INPUT THE DATA SET (X,Y) PAIRS (',I3,' MAX) BY NAMELIST 00000220
>WHERE:',/, 'X      = ARRAY OF X VALUES',/, 'Y      = ARRAY OF Y VAL00000221
>UES',/, 'CURRENT VALUES ARE:')                        00000222
206  FORMAT(/,'$PARM X=____,____,____,Y=____,____,____$ ',/) 00000223
207  FORMAT(/,'DATA SETS ARE INPUT FROM FORMATTED DISK FILES AS (X,Y) P00000224
>AIRS WHERE:',/, 'X      = X VALUE',/, 'Y      = Y VALUE') 00000225
208  FORMAT(/,'INPUT THE FILE NAME OF THE FORMATTED DISK FILE DATA SET.00000226
>')                                                       00000227
209  FORMAT(A80)                                           00000228
210  FORMAT(/,'INPUT THE DATA FILE FORMAT (INCLUDE PARENTHESES)') 00000229
211  FORMAT('NOTE: AN EXAMPLE FORMAT IS "(2E15.6)"',/,T7,'ENTER"(*)" FO00000230
>R A FREE FIELD READ (DEFAULT FORMAT)')                00000231
212  FORMAT(/,'SHOULD THE X-Y DATA BE DISPLAYED FOR VERIFICATION? (Y/N)00000232
>')                                                       00000233
213  FORMAT(/,T10,'NO.',T26,'X',T49,'Y')                00000234
214  FORMAT(T8,I5,1P2D23.13)                             00000235
215  FORMAT('SUBROUTINE NDRUS WAS CALLED FROM SUBROUTINE XYDIS') 00000236
216  FORMAT(/,'ERROR IN SUBROUTINE XYDIS: UNFORMATTED WRITE ERROR ON UN00000237
>IT 1')                                                  00000238
217  FORMAT(/,'ERROR IN SUBROUTINE XYDIS: OPEN ERROR ON UNIT 4, STATUS 00000239
>= "OLD", FILE =',/)                                    00000240

```

```

218  FORMAT(/,'ERROR IN SUBROUTINE XYDIS:  UNANTICIPATED EOF ON UNIT 4, 00000241
>FILE =',/), 00000242
219  FORMAT(/,'ERROR IN SUBROUTINE XYDIS: READ ERROR ON UNIT 4, FILE ='00000243
>,/, 00000244
220  FORMAT(/,'ERROR IN SUBROUTINE XYDIS: UNFORMATTED READ ERROR ON UNI00000245
>T 1'), 00000246
221  FORMAT(/,'ERROR IN PROGRAM LSPRWC: CLOSE ERROR ON UNIT 4, FILE ='00000247
>/), 00000248
222  FORMAT(/,'ERROR IN SUBROUTINE XYDIS: ALL X-Y DATA POINTS MUST BE',00000249
>/, 'MONOTONICALLY INCREASING IN X. ') 00000250
223  FORMAT(/,'ERROR IN SUBROUTINE XYDIS: ALL X-Y DATA POINTS HAVE THE 00000251
>SAME X VALUE LEADING TO',/, 'A SINGULAR MATRIX. ') 00000252
224  FORMAT('SUBROUTINE CBUS06 WAS CALLED FROM SUBROUTINE XYDIS') 00000253
      END 00000254

```

APPENDIX C
FORTRAN TOOL LIBRARY

SUBROUTINE CBUS01(Buf,L)	00000001
C	00000002
C	00000003
C*****	00000004
C*	*00000005
C* CHARACTER BUFFER UTILITY SUBROUTINE (CBUS01)	*00000006
C*	*00000007
C* REVISION DATE: 3 FEBRUARY 1997	*00000008
C*	*00000009
C*****	00000010
C	00000011
C*****	*00000012
C*	*00000013
C* SUBROUTINE CBUS01 FINDS THE NON-TRAILING-BLANK LENGTH OF CHARACTER	*00000014
C* BUFFERS.	*00000015
C*	*00000016
C* INPUT/OUTPUT VARIABLES:	*00000017
C*	*00000018
C* Buf = CHARACTER BUFFER	*00000019
C*	*00000020
C* OUTPUT VARIABLES:	*00000021
C*	*00000022
C* L = NON-TRAILING-BLANK LENGTH OF Buf	*00000023
C*	*00000024
C*****	00000025
C	00000026
C	00000027
C CHARACTER Blank*1,Buf*(*)	00000028
C	00000029
C SAVE Blank	00000030
C	00000031
C DATA Blank/' '/	00000032
C	00000033
C DO 101 L=LEN(Buf),1,-1	00000034
C IF(Buf(L:L).NE.Blank) RETURN	00000035
101 CONTINUE	00000036
C L=0	00000037
C END	00000038

```

SUBROUTINE CBUS06(LUnit,Buf,*)                                00000001
C                                                            00000002
C                                                            00000003
C*****00000004
C*                                                            *00000005
C*      CHARACTER BUFFER UTILITY SUBROUTINE (CBUS06)          *00000006
C*                                                            *00000007
C*      REVISION DATE: 11 MARCH 1999                          *00000008
C*                                                            *00000009
C*****00000010
C                                                            00000011
C*****00000012
C*                                                            *00000013
C*      SUBROUTINE CBUS06 WRITES CHARACTER BUFFERS, LESS THE TRAILING *00000014
C*      BLANKS, TO UNIT LUnit USING DYNAMIC FORMAT CONSTRUCTION. *00000015
C*                                                            *00000016
C* INPUT/OUTPUT VARIABLES:                                     *00000017
C*                                                            *00000018
C* LUnit  = LOGICAL UNIT NUMBER                                *00000019
C* Buf    = CHARACTER BUFFER                                   *00000020
C*                                                            *00000021
C*****00000022
C                                                            00000023
C                                                            00000024
C      CHARACTER Buf*(*),Fmt*15                               00000025
C                                                            00000026
C      INTEGER LUnit                                           00000027
C                                                            00000028
C      CALL CBUS01(Buf,L)                                       00000029
C      IF(L.EQ.0) L=1                                           00000030
C      WRITE(Fmt,FMT=201,ERR=101)L                               00000031
C      WRITE(UNIT=LUnit,FMT=Fmt,ERR=102)(Buf(1:L))              00000032
C      RETURN                                                    00000033
C                                                            00000034
C                                                            00000035
C*****00000036
C*                                                            *00000037
C*      ERROR MESSAGES                                          *00000038
C*                                                            *00000039
C*****00000040
C                                                            00000041
C                                                            00000042
101  WRITE(UNIT=6,FMT=202)                                       00000043
      RETURN 1                                                  00000044
102  WRITE(UNIT=6,FMT=203)LUnit                                  00000045
      RETURN 1                                                  00000046
C                                                            00000047
C                                                            00000048
C*****00000049
C*                                                            *00000050
C*      FORMAT STATEMENTS                                       *00000051
C*                                                            *00000052
C*****00000053
C                                                            00000054
C                                                            00000055
201  FORMAT(2H(A,I5,1H))                                         00000056
202  FORMAT(/,'ERROR IN SUBROUTINE CBUS06: WRITE ERROR',/, 'WHILE ATTEMPT00000057
>TING TO WRITE VARIABLE L TO Fmt')                             00000058
203  FORMAT(/,'ERROR IN SUBROUTINE CBUS06: WRITE ERROR',/, 'WHILE ATTEMPT00000059
>TING TO WRITE VARIABLE Buf TO UNIT = ',I2)                    00000060

```

END

00000061

```

SUBROUTINE DFCUS(Fmt,DFmt,*)                                00000001
C                                                            00000002
C                                                            00000003
C*****00000004
C*                                                            *00000005
C*      DYNAMIC FORMAT CONSTRUCTION UTILITY SUBROUTINE (DFCUS) *00000006
C*                                                            *00000007
C*      REVISION DATE: 20 NOVEMBER 2000                      *00000008
C*                                                            *00000009
C*****00000010
C                                                            00000011
C*****00000012
C*                                                            *00000013
C*      SUBROUTINE DFCUS READS A FORTRAN FORMAT FROM UNIT 5, STORES THE *00000014
C*      FORMAT IN A CHARACTER BUFFER, AND CHECKS THE FORMAT FOR ERRORS. *00000015
C*      Blank FORMATS ARE REPLACED WITH A DEFAULT FORMAT.    *00000016
C*                                                            *00000017
C* INPUT VARIABLES:                                          *00000018
C*                                                            *00000019
C* DFmt   = DEFAULT FORMAT CHARACTER BUFFER                 *00000020
C*                                                            *00000021
C* OUTPUT VARIABLES:                                        *00000022
C*                                                            *00000023
C* Fmt    = FORMAT CHARACTER BUFFER                         *00000024
C*                                                            *00000025
C*****00000026
C                                                            00000027
C                                                            00000028
C      CHARACTER CCV*1,DFmt*(*),Fmt*(*),LParn*1,RParn*1    00000029
C                                                            00000030
C      SAVE LParn,RParn                                     00000031
C                                                            00000032
C      DATA LParn/' ('/,RParn/')'/'                       00000033
C                                                            00000034
C      READ(UNIT=5,FMT=201)Fmt                             00000035
C      ENTRY DFCUSE(Fmt,DFmt,*)                             00000036
C      CALL CBUS03(Fmt)                                      00000037
C      CALL CBUS01(Fmt,LFmt)                                00000038
C      IF(LFmt.GT.0) GO TO 101                               00000039
C      CALL CBUS01(DFmt,LDFmt)                              00000040
C      LFmt=LEN(Fmt)                                         00000041
C      IF(LDFmt.GT.LFmt) GO TO 102                           00000042
C      Fmt(1:LDFmt)=DFmt(1:LDFmt)                           00000043
C      RETURN                                                00000044
101 IF(Fmt(1:1).EQ.LParn.AND.Fmt(LFmt:LFmt).EQ.RParn) RETURN 00000045
C                                                            00000046
C                                                            00000047
C*****00000048
C*                                                            *00000049
C*      ERROR MESSAGES                                       *00000050
C*                                                            *00000051
C*****00000052
C                                                            00000053
C                                                            00000054
C      WRITE(UNIT=6,FMT=202)                                00000055
C      CALL CBUS06(6,Fmt,*103)                              00000056
C      WRITE(UNIT=6,FMT=203)                                00000057
C      READ(UNIT=5,FMT=201)CCV                              00000058
C      RETURN 1                                              00000059
102 WRITE(UNIT=6,FMT=204)LDFmt,LFmt                        00000060

```


	STOP	00000061
103	WRITE(UNIT=6,FMT=205)	00000062
	RETURN 1	00000063
C		00000064
C		00000065
C	*****	00000066
C*		*00000067
C*	FORMAT STATEMENTS	*00000068
C*		*00000069
C	*****	00000070
C		00000071
C		00000072
201	FORMAT(A80)	00000073
202	FORMAT(/,'ERROR IN SUBROUTINE DFCUS: ILLEGAL VALUE FOR Fmt =')	00000074
203	FORMAT('FORMATS MUST BEGIN WITH "(" AND END WITH ")"',//,T20,'- EN	00000075
	>TER/RETURN TO CONTINUE -')	00000076
204	FORMAT(/,'FATAL ERROR IN SUBROUTINE DFCUS:',//,'THE NON-TRAILING-B	00000077
	>LANK LENGTH OF THE DEFAULT FORMAT CHARACTER',//,'BUFFER, LDFmt =',I	00000078
	>3,' IS GREATER THAN THE LENGTH OF THE FORMAT',//,'CHARACTER BUFFER,	00000079
	> LFmt =',I3,//,'EDIT AND RECOMPILE THE FORTRAN CODE.')	00000080
205	FORMAT('SUBROUTINE CBUS06 WAS CALLED FROM SUBROUTINE DFCUS')	00000081
	END	00000082

```

SUBROUTINE NDRUS(NBuf,KUnit,*,*)                                00000001
C                                                                00000002
C                                                                00000003
C*****00000004
C*                                                                *00000005
C*          NAMELIST-DIRECTED READ UTILITY SUBROUTINE (NDRUS)    *00000006
C*                                                                *00000007
C*          REVISION DATE: 23 MARCH 1999                        *00000008
C*                                                                *00000009
C*****00000010
C                                                                00000011
C*****00000012
C*                                                                *00000013
C* SUBROUTINE NDRUS EXECUTES AN INTELLIGENT NAMELIST-DIRECTED READ *00000014
C* PROCESS FOR PROGRAM PSCPP WITH INPUT VIA THE TEMPORARY SCRATCH *00000015
C* FILE ON LOGICAL UNIT NUMBER Unit.                            *00000016
C*                                                                *00000017
C* PARAMETERS:                                                  *00000018
C*                                                                *00000019
C* IBS      = INPUT CHARACTER BUFFER SIZE                       *00000020
C*                                                                *00000021
C* VARIABLES:                                                  *00000022
C*                                                                *00000023
C* NBuf     = NAMELIST GROUP-NAME CHARACTER BUFFER             *00000024
C* IBuf     = INPUT CHARACTER BUFFER                            *00000025
C* LUnit    = INPUT TEMPORARY SCRATCH FILE LOGICAL UNIT NUMBER *00000026
C*                                                                *00000027
C*****00000028
C                                                                00000029
C                                                                00000030
C          CHARACTER AEnd*4,Amsand*1,Blank*1,Comma*1,DEnd*4,Dollar*1,IBuf*80,00000031
>NBuf*(*),RBuf*80,Switch*1                                00000032
C                                                                00000033
C          INTEGER EPontr,SPontr,TPontr                        00000034
C                                                                00000035
C          LOGICAL OP                                          00000036
C                                                                00000037
C          PARAMETER (IBS=80)                                00000038
C                                                                00000039
C          EQUIVALENCE (Amsand,AEnd(1:1)),(Dollar,DEnd(1:1)) 00000040
C                                                                00000041
C          SAVE AEnd,Blank,Comma,DEnd,EPontr,LUnit,SPontr,Switch,TPontr 00000042
C                                                                00000043
C          DATA AEnd/'&END'/,Blank/' '/,Comma/' ','/,DEnd/'$END'/,Switch/'U'/ 00000044
C                                                                00000045
C                                                                00000046
C*****00000047
C*                                                                *00000048
C*          CHECK FOR A VALID NAMELIST GROUP-NAME CHARACTER BUFFER *00000049
C*                                                                *00000050
C*****00000051
C                                                                00000052
C                                                                00000053
C          LUnit=KUnit                                         00000054
          CALL CBUS04(NBuf,Switch,*110)                      00000055
          CALL CBUS03(NBuf)                                    00000056
          CALL CBUS01(NBuf,L)                                  00000057
          IF(L.EQ.0) GO TO 111                                  00000058
          IF(L+2.GT.IBS) GO TO 112                             00000059
C                                                                00000060

```

C	00000061
C*****	*00000062
C*	*00000063
C* OPEN THE TEMPORARY SCRATCH FILE ON LOGICAL UNIT NUMBER LUnit	*00000064
C*	*00000065
C*****	*00000066
C	00000067
C	00000068
INQUIRE(UNIT=LUnit,ERR=113,OPENED=OP)	00000069
IF(OP.EQV..TRUE.) CLOSE(UNIT=LUnit,STATUS='DELETE',ERR=114)	00000070
OPEN(UNIT=LUnit,STATUS='SCRATCH',FORM='FORMATTED',ERR=115)	00000071
C	00000072
C	00000073
C*****	*00000074
C*	*00000075
C* SUPPLY THE NAMELIST INITIALIZATION AND NAMELIST GROUP-NAME	*00000076
C*	*00000077
C*****	*00000078
C	00000079
C	00000080
CALL CBUS07(RBuf)	00000081
RBuf(2:2)=Dollar	00000082
RBuf(3:L+3)=NBuf(1:L)	00000083
WRITE(UNIT=LUnit,FMT=201,ERR=116)RBuf	00000084
C	00000085
C	00000086
C*****	*00000087
C*	*00000088
C* READ THE NAMELIST INPUT BUFFER	*00000089
C*	*00000090
C*****	*00000091
C	00000092
C	00000093
101 READ(UNIT=5,FMT=201,ERR=101)IBuf	00000094
C	00000095
C	00000096
C*****	*00000097
C*	*00000098
C* CHECK FOR A BLANK INPUT BUFFER	*00000099
C*	*00000100
C*****	*00000101
C	00000102
C	00000103
CALL CBUS01(IBuf,J)	00000104
IF(J.NE.0) GO TO 102	00000105
C	00000106
C	00000107
C*****	*00000108
C*	*00000109
C* SUPPLY THE DEFAULT CONDITION	*00000110
C*	*00000111
C*****	*00000112
C	00000113
C	00000114
CALL CBUS07(IBuf)	00000115
IBuf(2:2)=Dollar	00000116
WRITE(UNIT=LUnit,FMT=201,ERR=116)IBuf	00000117
GO TO 109	00000118
C	00000119
C	00000120

```

C*****00000121
C*00000122
C*      SEARCH FOR THE NAMELIST GROUP-NAME IN THE INPUT BUFFER00000123
C*00000124
C*****00000125
C00000126
C00000127
102  SPontr=100000128
      CALL CBUS04(IBuf,Switch,*110)00000129
      CALL CBUS01(IBuf,EPontr)00000130
      TPontr=INDEX(IBuf(SPontr:EPontr),NBuf(1:L))00000131
      IF(TPontr.EQ.0) GO TO 10800000132
C00000133
C00000134
C*****00000135
C*00000136
C*      CHECK FOR A VALID NAMELIST INPUT00000137
C*00000138
C*****00000139
C00000140
C00000141
      SPontr=TPontr-100000142
      IF(SPontr.LT.1) GO TO 10400000143
      DO 103 I=1,SPontr00000144
        IF(IBuf(I:I).EQ.Blank) GO TO 10300000145
        IF(IBuf(I:I).EQ.Dollar) GO TO 10300000146
        IF(IBuf(I:I).EQ.Amsand) GO TO 10300000147
        RETURN 100000148
103  CONTINUE00000149
C00000150
104  SPontr=TPontr+L00000151
      IF(SPontr.GT.EPontr) GO TO 10500000152
      IF(IBuf(SPontr:SPontr).EQ.Blank) GO TO 10500000153
      IF(IBuf(SPontr:SPontr).EQ.Comma) GO TO 10500000154
      IF(IBuf(SPontr:SPontr).EQ.Dollar) GO TO 10500000155
      IF(IBuf(SPontr:SPontr).EQ.Amsand) GO TO 10500000156
      RETURN 100000157
C00000158
105  SPontr=SPontr-100000159
      DO 106 I=1,SPontr00000160
106  IBuf(I:I)=Blank00000161
      GO TO 10800000162
C00000163
C00000164
C*****00000165
C*00000166
C*      CONTINUE TO READ AND FLUSH THE INPUT BUFFER00000167
C*00000168
C*****00000169
C00000170
C00000171
107  READ(UNIT=5,FMT=201,ERR=107)IBuf00000172
      CALL CBUS04(IBuf,Switch,*110)00000173
      CALL CBUS01(IBuf,EPontr)00000174
108  WRITE(UNIT=LUnit,FMT=201,ERR=116)IBuf00000175
C00000176
C00000177
C*****00000178
C*00000179
C*      CHECK FOR TERMINATION IN THE INPUT BUFFER00000180

```

```

C*                                                    *00000181
C*****                                                    *00000182
C                                                    00000183
C                                                    00000184
C      SPontr=EPontr-3                                00000185
C      IF(IBuf(EPontr:EPontr).EQ.Dollar) GO TO 109    00000186
C      IF(IBuf(EPontr:EPontr).EQ.Amsand) GO TO 109    00000187
C      IF(IBuf(SPontr:EPontr).EQ.DEnd) GO TO 109      00000188
C      IF(IBuf(SPontr:EPontr).EQ.AEnd) GO TO 109      00000189
C      GO TO 107                                       00000190
109  REWIND LUnit                                     00000191
C      RETURN                                          00000192
C                                                    00000193
C                                                    00000194
C*****                                                    *00000195
C*                                                    *00000196
C*      CLOSE THE TEMPORARY SCRATCH FILE ON LOGICAL UNIT NUMBER UNIT *00000197
C*                                                    *00000198
C*****                                                    *00000199
C                                                    00000200
C                                                    00000201
C      ENTRY NDRUSE(*,*)                             00000202
C      CLOSE(UNIT=LUnit,STATUS='DELETE',ERR=114)      00000203
C      RETURN                                          00000204
C                                                    00000205
C                                                    00000206
C*****                                                    *00000207
C*                                                    *00000208
C*      ERROR MESSAGES                               *00000209
C*                                                    *00000210
C*****                                                    *00000211
C                                                    00000212
C                                                    00000213
110  WRITE(UNIT=6,FMT=202)                            00000214
C      RETURN 2                                       00000215
111  WRITE(UNIT=6,FMT=203)                            00000216
C      RETURN 2                                       00000217
112  WRITE(UNIT=6,FMT=204)                            00000218
C      CALL CBUS06(6,NBuf,*117)                     00000219
C      WRITE(UNIT=6,FMT=205)IBS                       00000220
C      RETURN 2                                       00000221
113  WRITE(UNIT=6,FMT=206)LUnit                        00000222
C      RETURN 2                                       00000223
114  WRITE(UNIT=6,FMT=207)LUnit                        00000224
C      RETURN 2                                       00000225
115  WRITE(UNIT=6,FMT=208)LUnit                        00000226
C      RETURN 2                                       00000227
116  WRITE(UNIT=6,FMT=209)LUnit                        00000228
C      RETURN 2                                       00000229
117  WRITE(UNIT=6,FMT=210)                            00000230
C      RETURN 2                                       00000231
C                                                    00000232
C                                                    00000233
C*****                                                    *00000234
C*                                                    *00000235
C*      FORMAT STATEMENTS                           *00000236
C*                                                    *00000237
C*****                                                    *00000238
C                                                    00000239
C                                                    00000240

```

```

201  FORMAT(A80)                                00000241
202  FORMAT('SUBROUTINE CBUS04 WAS CALLED FROM SUBROUTINE NDRUS') 00000242
203  FORMAT(/,'ERROR IN SUBROUTINE NDRUS: EMPTY NAMELIST GROUP-NAME CHA00000243
>RACTER',/, 'BUFFER, NBuf')                    00000244
204  FORMAT(/,'ERROR IN SUBROUTINE NDRUS: THE LENGTH OF THE NAMELIST GR00000245
>OUP-NAME NBuf =')                              00000246
205  FORMAT('PLUS 2 EXCEEDS THE INPUT BUFFER SIZE, IBS =',I3)    00000247
206  FORMAT(/,'ERROR IN SUBROUTINE NDRUS: INQUIRE ERROR ON UNIT ',I1) 00000248
207  FORMAT(/,'ERROR IN SUBROUTINE NDRUS: CLOSE ERROR ON UNIT ',I1,/, 'S00000249
>TATUS = "DELETE"')                             00000250
208  FORMAT(/,'ERROR IN SUBROUTINE NDRUS: OPEN ERROR ON UNIT ',I1,', ST00000251
>ATUS = "SCRATCH"')                             00000252
209  FORMAT(/,'ERROR IN SUBROUTINE NDRUS: WRITE ERROR ON UNIT ',I1) 00000253
210  FORMAT('SUBROUTINE CBUS06 WAS CALLED FROM SUBROUTINE NDRUS') 00000254
      END                                         00000255

```

```

SUBROUTINE NDWUS(NBuf,KUnit,*)                                00000001
C                                                                00000002
C                                                                00000003
C*****00000004
C*                                                                *00000005
C*      NAMELIST-DIRECTED WRITE UTILITY SUBROUTINE (NDWUS)      *00000006
C*                                                                *00000007
C*      REVISION DATE: 23 MARCH 1999                             *00000008
C*                                                                *00000009
C*****00000010
C                                                                00000011
C*****00000012
C*                                                                *00000013
C* SUBROUTINE NDWUS EXECUTES AN INTELLIGENT NAMELIST-DIRECTED WRITE *00000014
C* PROCESS FOR PROGRAM PSCPP WITH OUTPUT TO THE HIGH LEVEL PLOT *00000015
C* COMMAND FILE ON LOGICAL UNIT NUMBER LUnit.                   00000016
C*                                                                *00000017
C* PARAMETERS:                                                    *00000018
C*                                                                *00000019
C* FieldL = FIELD LENGTH                                         *00000020
C* OBS     = OUTPUT CHARACTER BUFFER SIZE                         *00000021
C*                                                                *00000022
C* VARIABLES:                                                     *00000023
C*                                                                *00000024
C* AFL      = AVAILABLE FIELD LENGTH                             *00000025
C* CBuf      = CHARACTER BUFFER                                  *00000026
C* DBuf      = DOUBLE PRECISION ARRAY BUFFER                     *00000027
C* IBuf      = INTEGER ARRAY BUFFER                              *00000028
C* N         = NUMBER OF ELEMENTS IN RBuf OR IBuf                *00000029
C* N1        = STARTING ELEMENT IN RBuf OR IBuf                  *00000030
C* N2        = ENDING ELEMENT IN RBuf OR IBuf                    *00000031
C* NBuf      = NAMELIST GROUP-NAME CHARACTER BUFFER              *00000032
C* NNFL      = NAMELIST NAME FIELD LENGTH                        *00000033
C* OBuf      = OUTPUT CHARACTER BUFFER                           *00000034
C* RBuf      = REAL ARRAY BUFFER                                 *00000035
C* RFL      = REQUIRED FIELD LENGTH                               *00000036
C* LUnit     = OUTPUT FILE LOGICAL UNIT NUMBER                   *00000037
C* VBuf      = VARIABLE NAME CHARACTER BUFFER                    *00000038
C* VNFL      = VARIABLE NAME FIELD LENGTH                        *00000039
C*                                                                *00000040
C*****00000041
C                                                                00000042
C                                                                00000043
C      CHARACTER Apos*2,CBuf*(*),Dollar*1,Equal*1,NBuf*(*),OBuf*75, 00000044
>VBuf*(*)                                                         00000045
C                                                                00000046
C      REAL*8 DBuf                                                 00000047
C                                                                00000048
C      INTEGER AFL,EPontr,FieldL,OBS,RFL,SPontr,TPontr,VNFL        00000049
C                                                                00000050
C      PARAMETER (FieldL=15,OBS=75)                                00000051
C                                                                00000052
C      DIMENSION DBuf(N),IBuf(N),RBuf(N)                           00000053
C                                                                00000054
C      EQUIVALENCE (AFL,NNFL,VNFL)                                 00000055
C                                                                00000056
C      SAVE AFL,Apos,Dollar,EPontr,Equal,ICV,LUnit,OBuf,RFL,SPontr, 00000057
>TPontr,VNFL                                                       00000058
C                                                                00000059
C      DATA Apos/' ','"/,Dollar/'$'/,Equal/'='/'                00000060

```

C		00000061
C		00000062
C	*****	00000063
C*		*00000064
C*	OPEN THE NAMELIST OUTPUT BUFFER	*00000065
C*		*00000066
C	*****	00000067
C		00000068
C		00000069
	LUnit=KUnit	00000070
	CALL CBUS03(NBuf)	00000071
	CALL CBUS01(NBuf,L)	00000072
	NNFL=FieldL	00000073
	IF((L+2).GT.NNFL) GO TO 124	00000074
	CALL CBUS07(Obuf)	00000075
	SPontr=2	00000076
	EPontr=2	00000077
	Obuf(SPontr:EPontr)=Dollar	00000078
	SPontr=EPontr+1	00000079
	EPontr=EPontr+L	00000080
	Obuf(SPontr:EPontr)=NBuf(1:L)	00000081
	TPontr=FieldL	00000082
	CALL CBUS01(Obuf,L)	00000083
	VNFL=TPontr-L-1	00000084
	RETURN	00000085
C		00000086
C		00000087
C	*****	00000088
C*		*00000089
C*	INSERT A CHARACTER VARIABLE INTO THE OUTPUT BUFFER	*00000090
C*		*00000091
C	*****	00000092
C		00000093
C		00000094
	ENTRY NDWUSC(VBuf,CBuf,*)	00000095
	CALL CBUS03(CBuf)	00000096
	CALL CBUS01(CBuf,L)	00000097
	RFL=L+4	00000098
	ICV=1	00000099
	GO TO 118	00000100
101	CALL CBUS01(CBuf,L)	00000101
	AFL=0	00000102
102	TPontr=TPontr+FieldL	00000103
	AFL=AFL+FieldL	00000104
	IF(RFL.GT.AFL) GO TO 102	00000105
	IF(TPontr.GT.OBS) TPontr=OBS	00000106
	SPontr=TPontr-L-2	00000107
	EPontr=SPontr	00000108
	Obuf(SPontr:EPontr)=Apos(1:1)	00000109
	SPontr=EPontr+1	00000110
	EPontr=TPontr-2	00000111
	Obuf(SPontr:EPontr)=CBuf(1:L)	00000112
	SPontr=TPontr-1	00000113
	EPontr=TPontr	00000114
	Obuf(SPontr:EPontr)=Apos	00000115
	VNFL=0	00000116
	RETURN	00000117
C		00000118
C		00000119
C	*****	00000120

C*		*00000121
C*	INSERT AN INTEGER ARRAY INTO THE OUTPUT BUFFER	*00000122
C*		*00000123
C*****		*00000124
C		00000125
C		00000126
	ENTRY NDWUSI(VBuf,N,N1,N2,IBuf,*)	00000127
	RFL=FieldL	00000128
	ICV=2	00000129
	GO TO 118	00000130
103	DO 107 I=N1,N2	00000131
104	TPontr=TPontr+FieldL	00000132
	IF(TPontr.LE.OBS) GO TO 106	00000133
	JCV=1	00000134
	GO TO 123	00000135
105	GO TO 104	00000136
106	SPontr=TPontr-FieldL+1	00000137
	EPontr=TPontr	00000138
107	WRITE(Obuf(SPontr:EPontr),FMT=201)IBuf(I)	00000139
	VNFL=0	00000140
	RETURN	00000141
C		00000142
C		00000143
C*****		*00000144
C*		*00000145
C*	INSERT A REAL ARRAY INTO THE OUTPUT BUFFER	*00000146
C*		*00000147
C*****		*00000148
C		00000149
C		00000150
	ENTRY NDWUSR(VBuf,N,N1,N2,RBuf,*)	00000151
	RFL=FieldL	00000152
	ICV=3	00000153
	GO TO 118	00000154
108	DO 112 I=N1,N2	00000155
109	TPontr=TPontr+FieldL	00000156
	IF(TPontr.LE.OBS) GO TO 111	00000157
	JVC=2	00000158
	GO TO 123	00000159
110	GO TO 109	00000160
111	SPontr=TPontr-FieldL+1	00000161
	EPontr=TPontr	00000162
112	WRITE(Obuf(SPontr:EPontr),FMT=202)RBuf(I)	00000163
	VNFL=0	00000164
	RETURN	00000165
C		00000166
C		00000167
C*****		*00000168
C*		*00000169
C*	INSERT A DOUBLE PRECISION ARRAY INTO THE OUTPUT BUFFER	*00000170
C*		*00000171
C*****		*00000172
C		00000173
C		00000174
	ENTRY NDWUSD(VBuf,N,N1,N2,DBuf,*)	00000175
	RFL=FieldL	00000176
	ICV=4	00000177
	GO TO 118	00000178
113	DO 117 I=N1,N2	00000179
114	TPontr=TPontr+FieldL	00000180

	IF(TPontr.LE.OBS) GO TO 116	00000181
	JVC=3	00000182
	GO TO 123	00000183
115	GO TO 114	00000184
116	SPontr=TPontr-FieldL+1	00000185
	EPontr=TPontr	00000186
117	WRITE(Obuf(SPontr:EPontr),FMT=202)DBuf(I)	00000187
	VNFL=0	00000188
	RETURN	00000189
C		00000190
C		00000191
C	*****	00000192
C*		*00000193
C*	INSERT A VARIABLE NAME INTO THE OUTPUT BUFFER	*00000194
C*		*00000195
C	*****	00000196
C		00000197
C		00000198
118	CALL CBUS03(VBuf)	00000199
	CALL CBUS01(VBuf,L)	00000200
	IF((L+2)+RFL.GT.OBS) RETURN 1	00000201
119	IF((L+2).LE.VNFL) GO TO 120	00000202
	TPontr=TPontr+FieldL	00000203
	VNFL=VNFL+FieldL	00000204
	GO TO 119	00000205
120	IF(TPontr+RFL.LE.OBS) GO TO 122	00000206
	JVC=4	00000207
	GO TO 123	00000208
121	GO TO 119	00000209
122	SPontr=TPontr-L-1	00000210
	EPontr=TPontr-2	00000211
	Obuf(SPontr:EPontr)=VBuf(1:L)	00000212
	SPontr=EPontr+2	00000213
	EPontr=TPontr	00000214
	Obuf(SPontr:EPontr)=Equal	00000215
	GO TO (101,103,108,113),ICV	00000216
C		00000217
C		00000218
C	*****	00000219
C*		*00000220
C*	FLUSH THE OUTPUT BUFFER	*00000221
C*		*00000222
C	*****	00000223
C		00000224
C		00000225
123	WRITE(UNIT=LUnit,Fmt=203,ERR=129)Obuf	00000226
	CALL CBUS07(Obuf)	00000227
	TPontr=0	00000228
	VNFL=0	00000229
	GO TO (105,110,115,121),JVC	00000230
C		00000231
C		00000232
C	*****	00000233
C*		*00000234
C*	CLOSE THE NAMELIST OUTPUT BUFFER	*00000235
C*		*00000236
C	*****	00000237
C		00000238
C		00000239
	ENTRY NDWUSE(*)	00000240

CALL CBUS01(Obuf,L)	00000241
SPontr=L	00000242
EPontr=L	00000243
Obuf(SPontr:EPontr)=Dollar	00000244
WRITE(UNIT=LUnit,Fmt=203,ERR=129)Obuf	00000245
IF(LUnit.EQ.6) WRITE(UNIT=LUnit,Fmt=204,ERR=129)	00000246
RETURN	00000247
C	00000248
C	00000249
C*****	00000250
C*	*00000251
C* ERROR MESSAGES	*00000252
C*	*00000253
C*****	00000254
C	00000255
C	00000256
124 WRITE(UNIT=6,FMT=205)	00000257
CALL CBUS06(6,NBuf,*130)	00000258
WRITE(UNIT=6,FMT=206)NNFL	00000259
RETURN 1	00000260
125 WRITE(UNIT=6,FMT=207)	00000261
CALL CBUS06(6,VBuf,*130)	00000262
WRITE(UNIT=6,FMT=208)	00000263
CALL CBUS06(6,CBuf,*130)	00000264
WRITE(UNIT=6,FMT=209)OBS	00000265
RETURN 1	00000266
126 WRITE(UNIT=6,FMT=207)	00000267
CALL CBUS06(6,VBuf,*130)	00000268
WRITE(UNIT=6,FMT=210)OBS	00000269
RETURN 1	00000270
127 WRITE(UNIT=6,FMT=207)	00000271
CALL CBUS06(6,VBuf,*130)	00000272
WRITE(UNIT=6,FMT=211)OBS	00000273
RETURN 1	00000274
128 WRITE(UNIT=6,FMT=207)	00000275
CALL CBUS06(6,VBuf,*130)	00000276
WRITE(UNIT=6,FMT=212)OBS	00000277
RETURN 1	00000278
129 WRITE(UNIT=6,FMT=213)LUnit	00000279
RETURN 1	00000280
130 WRITE(UNIT=6,FMT=214)	00000281
131 RETURN 1	00000282
C	00000283
C	00000284
C*****	00000285
C*	*00000286
C* FORMAT STATEMENTS	*00000287
C*	*00000288
C*****	00000289
C	00000290
C	00000291
201 FORMAT(I14,',')	00000292
202 FORMAT(E14.6,',')	00000293
203 FORMAT(A75)	00000294
204 FORMAT()	00000295
205 FORMAT(/,'ERROR IN SUBROUTINE NDWUS: THE LENGTH OF THE NAMELIST GR00000296	
>OUP-NAME NBuf =')	00000297
206 FORMAT('EXCEEDS THE NAMELIST NAME FIELD LENGTH =',I3)	00000298
207 FORMAT(/,'ERROR IN SUBROUTINE NDWUS: THE LENGTH OF THE VARIABLE NA00000299	
>ME VBuf =')	00000300

208	FORMAT('PLUS THE LENGTH OF THE CHARACTER VARIABLE CBuf =')	00000301
209	FORMAT('EXCEEDS THE OUTPUT BUFFER SIZE, OBS =',I3)	00000302
210	FORMAT('PLUS THE FIELD LENGTH FOR AN INTEGER VARIABLE EXCEEDS THE	00000303
	>OUTPUT BUFFER',/, 'SIZE, OBS =',I3)	00000304
211	FORMAT('PLUS THE FIELD LENGTH FOR A REAL VARIABLE EXCEEDS THE OUTP	00000305
	>UT BUFFER',/, 'SIZE, OBS =',I3)	00000306
212	FORMAT('PLUS THE FIELD LENGTH FOR A DOUBLE PRECISION VARIABLE EXCE	00000307
	>EDS THE OUTPUT BUFFER',/, 'SIZE, OBS =',I3)	00000308
213	FORMAT(/, 'ERROR IN SUBROUTINE NDWUS: WRITE ERROR ON UNIT ',I1)	00000309
214	FORMAT('SUBROUTINE CBUS06 WAS CALLED FROM SUBROUTINE NDWUS')	00000310
	END	00000311

SUBROUTINE YNOUS(*,*,*)	00000001
C	00000002
C	00000003
C*****	00000004
C*	*00000005
C* YES/NO/OTHER UTILITY SUBROUTINE (YNOUS)	*00000006
C*	*00000007
C* REVISION DATE: 29 SEPTEMBER 2000	*00000008
C*	*00000009
C*****	00000010
C	00000011
C*****	00000012
C*	*00000013
C* SUBROUTINE YNOUS READS ONE BYTE FROM UNIT 5 ANTICIPATING A 'Y' OR	*00000014
C* 'Y' YES RESPONSE OR ALTERNATIVELY A 'N' OR 'n' NO RESPONSE. A YES	*00000015
C* RESPONSE RESULTS IN A RETURN 1; A NO RESPONSE RESULTS IN A	*00000016
C* RETURN 2. ALL OTHER RESPONSES RESULT IN A RETURN 3.	*00000017
C*	*00000018
C*****	00000019
C	00000020
C	00000021
CHARACTER CCV*1,No*1,Switch*1,Yes*1	00000022
C	00000023
SAVE No,Yes	00000024
C	00000025
DATA No/'N'/,Switch/'U'/,Yes/'Y'/'	00000026
C	00000027
C	00000028
C*****	00000029
C*	*00000030
C* READ ONE BYTE AND COMPARE WITH Yes/No	*00000031
C*	*00000032
C*****	00000033
C	00000034
C	00000035
READ(UNIT=5,FMT=201)CCV	00000036
CALL CBUS04(CCV,Switch,*101)	00000037
IF(CCV.EQ.Yes) RETURN 1	00000038
IF(CCV.EQ.No) RETURN 2	00000039
RETURN 3	00000040
C	00000041
C	00000042
C*****	00000043
C*	*00000044
C* ERROR MESSAGES	*00000045
C*	*00000046
C*****	00000047
C	00000048
C	00000049
101 WRITE(UNIT=6,FMT=202)	00000050
RETURN 3	00000051
C	00000052
C*****	00000053
C*	*00000054
C* FORMAT STATEMENTS	*00000055
C*	*00000056
C*****	00000057
C	00000058
C	00000059
201 FORMAT(A1)	00000060

202 FORMAT('SUBROUTINE CBUS04 WAS CALLED FROM SUBROUTINE YNOUS')
 END

00000061
00000062

APPENDIX D
EXAMPLE 1

In this example, the LSPRWC code was applied to a set of observations taken for an object accelerating from rest at a constant 9.81 m/s^2 , the acceleration of gravity. The true behavior for the object, in terms of displacement as a function of time, is given by

$$\frac{d^2x}{dt^2} = \text{constant} \quad (1)$$

$$= a, \quad (2)$$

and integrating equation (1) twice gives

$$\frac{dx}{dt} = v_0 + at, \quad (3)$$

$$= v(t), \quad (4)$$

and

$$x(t) = x_0 + v_0t + \frac{1}{2}at^2 \quad (5)$$

where

a = acceleration,

t = time,

v = velocity,

v_0 = initial velocity,

x = displacement,

x_0 = initial position.

Then for

$$x_0 = 0, \quad (6)$$

$$v_0 = 0, \quad (7)$$

equations (3), (4), and (5) reduce to

$$v(t) = at \quad (8)$$

and

$$x(t) = \frac{1}{2}at^2. \quad (9)$$

The observations for this example are given in Table D-1 along with the true values for displacement, velocity, and acceleration. Figures D-1 through 3 present *X Window System* screen shots from LSPRWC code execution using this data set.

Table D-1. Trajectory Data

Time (s)	True Position (m)	True Velocity (m/s)	True Acceleration (m/s**2)	Measured Position (m)
0.000000E+00	0.000000E+00	0.000000E+00	0.981000E+01	0.000000E+00
0.100000E+00	0.490500E-01	0.981000E+00	0.981000E+01	0.841678E+00
0.200000E+00	0.196200E+00	0.196200E+01	0.981000E+01	0.262622E+00
0.300000E+00	0.441450E+00	0.294300E+01	0.981000E+01	-0.293670E+00
0.400000E+00	0.784800E+00	0.392400E+01	0.981000E+01	0.177628E+01
0.500000E+00	0.122625E+01	0.490500E+01	0.981000E+01	0.179120E+01
0.600000E+00	0.176580E+01	0.588600E+01	0.981000E+01	0.223221E+01
0.700000E+00	0.240345E+01	0.686700E+01	0.981000E+01	0.211730E+01
0.800000E+00	0.313920E+01	0.784800E+01	0.981000E+01	0.322466E+01
0.900000E+00	0.397305E+01	0.882900E+01	0.981000E+01	0.306116E+01
0.100000E+01	0.490500E+01	0.981000E+01	0.981000E+01	0.466451E+01
0.110000E+01	0.593505E+01	0.107910E+02	0.981000E+01	0.669913E+01
0.120000E+01	0.706320E+01	0.117720E+02	0.981000E+01	0.781203E+01
0.130000E+01	0.828945E+01	0.127530E+02	0.981000E+01	0.790575E+01
0.140000E+01	0.961380E+01	0.137340E+02	0.981000E+01	0.105721E+02
0.142784E+01	0.100000E+02	0.140071E+02	0.981000E+01	0.920335E+01

PROGRAM LSPRWC IS AN INTERACTIVE FORTRAN PROGRAM TO PERFORM A LEAST SQUARES POLYNOMIAL REGRESSION WITH CONSTRAINTS; THAT IS, A SET OF X-Y DATA POINTS IS CURVE FIT WITH AN NP ORDER POLYNOMIAL OF THE FORM

$$P(X)=B_0+B_1X+B_2X^{**2}+B_3X^{**3}+....+B_{NP}X^{**NP}$$

WITH ANY POLYNOMIAL DERIVATIVES, ZERO THROUGH NP, SPECIFIED AT GIVEN X LOCATIONS. THE PROCEDURE FOLLOWED IS THE METHOD OF LEAST SQUARES USING UNDETERMINED LAGRANGE MULTIPLIERS.

AS AN INTERACTIVE PROGRAM, LSPRWC IS SELF-EXPLANATORY AND PROMPTS FOR THE NECESSARY INFORMATION. THE X-Y DATA TO BE FITTED MAY BE ENTERED BY NAMELIST OR READ FROM A FORMATTED DISK FILE. PROGRAM LSPRWC WILL ALSO EVALUATE THE RESULTANT LEAST SQUARES POLYNOMIAL AT PRESCRIBED VALUES OF X WHICH, AGAIN, MAY BE ENTERED BY NAMELIST OR READ FROM A FORMATTED DISK FILE.

- ENTER/RETURN TO CONTINUE -

SELECT THE SOURCE OF INPUT FOR THE X-Y DATA TO BE FITTED FROM THE FOLLOWING LIST:

- 1, FOR KEYBOARD INPUT VIA NAMELIST
- 2, FOR FORMATTED DISK FILE INPUT

NOTE: THE X-Y DATA MUST BE MONOTONICALLY INCREASING IN X.
2

DATA SETS ARE INPUT FROM FORMATTED DISK FILES AS (X,Y) PAIRS WHERE:

X = X VALUE
Y = Y VALUE

INPUT THE FILE NAME OF THE FORMATTED DISK FILE DATA SET.
trajectory.dat

INPUT THE DATA FILE FORMAT (INCLUDE PARENTHESES)

NOTE: AN EXAMPLE FORMAT IS "(2E15,6)"

ENTER"(*)" FOR A FREE FIELD READ (DEFAULT FORMAT)
(E15,6,T61,E15,6)

SHOULD THE X-Y DATA BE DISPLAYED FOR VERIFICATION? (Y/N)

y

NO.	X	Y
1	0.0000000000000E+00	0.0000000000000E+00
2	1.0000000000000E-01	8.4167800000000E-01
3	2.0000000000000E-01	2.6262200000000E-01
4	3.0000000000000E-01	-2.9367000000000E-01
5	4.0000000000000E-01	1.7762800000000E+00
6	5.0000000000000E-01	1.7912000000000E+00
7	6.0000000000000E-01	2.2322100000000E+00
8	7.0000000000000E-01	2.1173000000000E+00
9	8.0000000000000E-01	3.2246600000000E+00
10	9.0000000000000E-01	3.0611600000000E+00
11	1.0000000000000E+00	4.6645100000000E+00
12	1.1000000000000E+00	6.6991300000000E+00
13	1.2000000000000E+00	7.8120300000000E+00
14	1.3000000000000E+00	7.9057500000000E+00
15	1.4000000000000E+00	1.0572100000000E+01
16	1.4278400000000E+00	9.2033500000000E+00

- ENTER/RETURN TO CONTINUE -

Figure D-1 LSPRWC Screen Shot—Part 1

```

ARE CONSTRAINTS DESIRED? (Y/N)
N

INPUT THE POLYNOMIAL ORDER BY NAMELIST WHERE:

NP      = THE POLYNOMIAL ORDER

NOTE: NP MUST BE IN THE RANGE OF 0 TO 9.
      THERE MUST BE AT LEAST NP+NCONST+1 X-Y DATA POINTS.
      THE MAXIMUM VALUE FOR NP IS      9.

CURRENT VALUES ARE:
$PARM      NP =              3$

Np=2$
      LEAST SQUARES POLYNOMIAL

      
$$P(X)=B(0)+B(1)*X+B(2)*X**2+....+B(NP)*X**NP$$


      I          B(I)

      0          2.7806885044562E-01
      1          -3.2364562759090E-01
      2          5.0277277787627E+00

      THE STANDARD DEVIATION FOR THIS POLYNOMIAL OF ORDER  2 IS  7.17016E-01

      - ENTER/RETURN TO CONTINUE -

SHOULD DATA POINTS BE PRESCRIBED FOR EVALUATION OF THE LEAST SQUARES
POLYNOMIALS? (Y/N)
Y

SELECT THE SOURCE OF INPUT FOR THE X DATA TO BE EVALUATED FROM THE FOLLOWING
LIST:

1, FOR KEYBOARD INPUT VIA NAMELIST
2, FOR FORMATTED DISK FILE INPUT
3, FOR INPUT OF MINIMUM, MAXIMUM, AND INTERVAL VALUES
3

INPUT THE X DATA MAXIMUM, MINIMUM, AND INTERVAL VALUES BY NAMELIST WHERE:

XINT      = X DATA POINT INTERVAL
XMAX      = MAXIMUM X DATA VALUE
XMIN      = MINIMUM X DATA VALUE

CURRENT VALUES ARE:
$PARM      XINT =  0.142784E-01,      XMAX =  0.142784E+01,
           XMIN =  0.000000E+00$

SHOULD THE X-Y DATA BE DISPLAYED FOR VERIFICATION? (Y/N)
n

SHOULD THE RESULTS OF THIS RUN BE QUICK-PLOTTED? (Y/N)
N

```

Figure D-2. LSPRWC Screen Shot—Part 2

```

ENTER:
1, TO RESTART THE PROGRAM
2, TO CHANGE THE CONSTRAINTS
3, TO CHANGE THE POLYNOMIAL ORDER
4, TO CHANGE THE PRESCRIBED X VALUES FOR EVALUATION OF THE POLYNOMIAL
5, TO PLOT THE RESULTS
6, TO SAVE THE RESULTS ON A FORMATTED DISC FILE
7, TO STOP
6

INPUT THE FILE NAME OF THE FORMATTED DISK FILE.
np2.dat

SELECT THE TYPE OF OUTPUT FOR THE FORMATTED DISK FILE FROM THE FOLLOWING LIST:
1, FOR ALL INPUT DATA AND RESULTS
2, FOR ONLY THE EVALUATED DATA
2

INPUT THE EVALUATED DATA VARIABLE LIST FOR OUTPUT BY NAMELIST WHERE:

"X"   = X VALUES
"0"   = 0th DERIVATIVE OF Y AT X
"1"   = 1th DERIVATIVE OF Y AT X
"2"   = 2th DERIVATIVE OF Y AT X
"R"   = RADIUS OF CURVATURE AT X
"S"   = ARC LENGTH

NOTE: VARIABLES WILL BE LISTED IN THE ORDER OF INPUT

CURRENT VALUES ARE:

$PARM AList= "X","0"$

AList="X","0","1","2"$

INPUT THE DATA FILE FORMAT (INCLUDE PARENTHESES)

NOTE: AN EXAMPLE FORMAT IS "(F15.7,E15.6)"
      THE DEFAULT FORMAT IS:
(13E15.6)

ENTER:
1, TO RESTART THE PROGRAM
2, TO CHANGE THE CONSTRAINTS
3, TO CHANGE THE POLYNOMIAL ORDER
4, TO CHANGE THE PRESCRIBED X VALUES FOR EVALUATION OF THE POLYNOMIAL
5, TO PLOT THE RESULTS
6, TO SAVE THE RESULTS ON A FORMATTED DISC FILE
7, TO STOP
7

```

Figure D-3. LSPRWC Screen Shot—Part 3

Following along through Figure D-1, the user was presented with an introduction to LSPRWC and then asked for the data source. Here the source was selected as a formatted disk file followed by the file name and read format. The (x, y) data pairs—time and measured position—were displayed for verification and match the values given in Table D-1.

Moving on through Figure D-2, the code asked if problem constraints were desired and none were selected. The user was then asked to enter the polynomial order which was set to $n_p = 2$ by means of NAMELIST. With the problem thus fully defined, program LSPRWC

returned values for the coefficients of the least squares regression polynomial, b_0 , b_1 , and b_2 , and the standard deviation for the fit to the (x, y) data pairs as input. This 2nd order regression polynomial was then evaluated at specific values of x , in this case time, using the minimum, maximum, and interval option.

Following through Figure D-3, menu options were chosen to output the evaluated data to a formatted disk file writing, in order, each value of x , y , and the first and second derivatives, $\frac{dy}{dx}$, and $\frac{d^2y}{dx^2}$. With that, the LSPRWC code menu option was selected for program termination.

The results from this particular LSPRWC run are shown in Figure D-4 as plots of displacement and velocity as a function of time along with the true values and the measurements of Table D-1. The 2nd order least squares regression polynomial fits quite well in this case, as might be expected considering that the true solution for displacement, given by equation (9), is 2nd order in time.

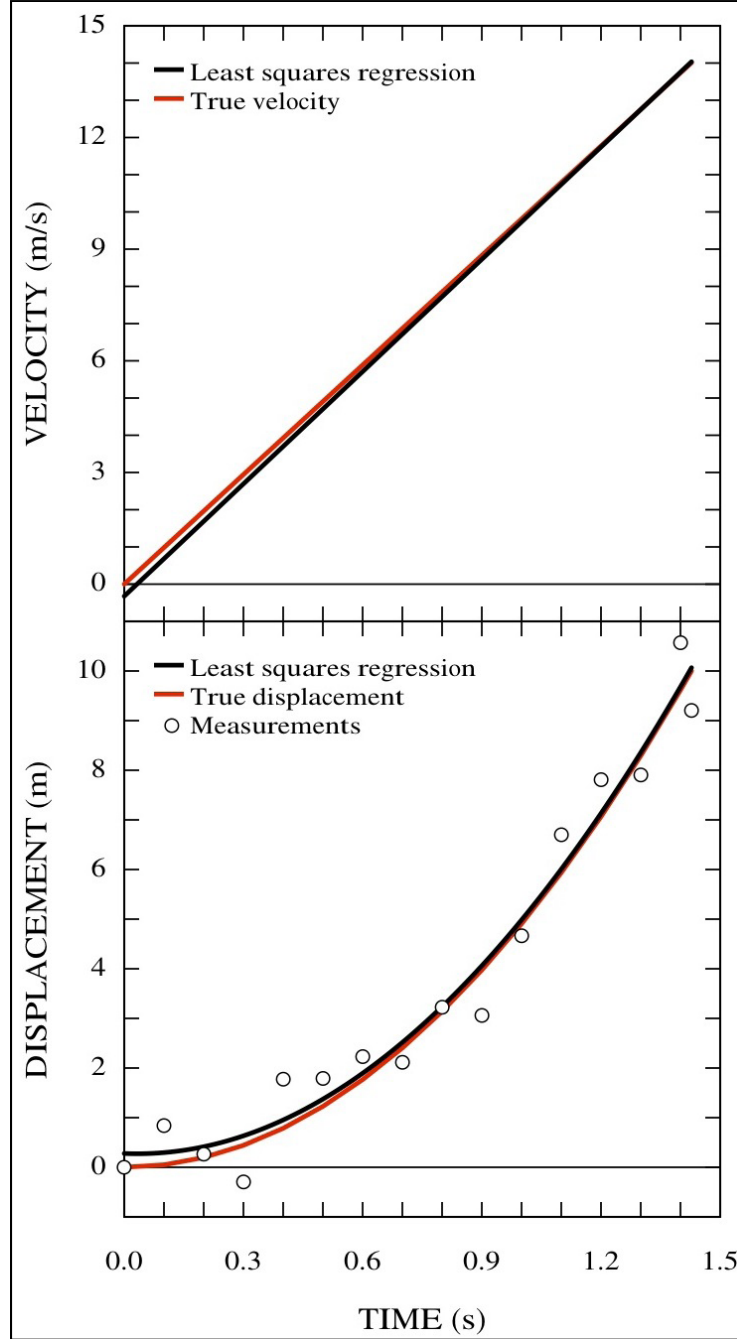


Figure D-4. Least Squares Polynomial Regression for $n_p = 2$

The LSPRWC code could have been run with any choice of polynomial order from 0 through 9 as was done to produce the results of Figure D-5. This plot of displacement as a function of time shows a reasonable fit for the regression polynomials for $n_p = 2$ and 3. Clearly, the representation begins to degrade with polynomial orders of 4 and greater.

Section V. LIMITATIONS notes that the LSPRWC code limits the polynomial order such that $0 \leq n_p \leq 9$. This example serves as excellent justification for this limitation since the

9th order least squares regression polynomial gives the best fit to the measured data in Figure D-5 yet the polynomial representation is quite unlike the true solution.

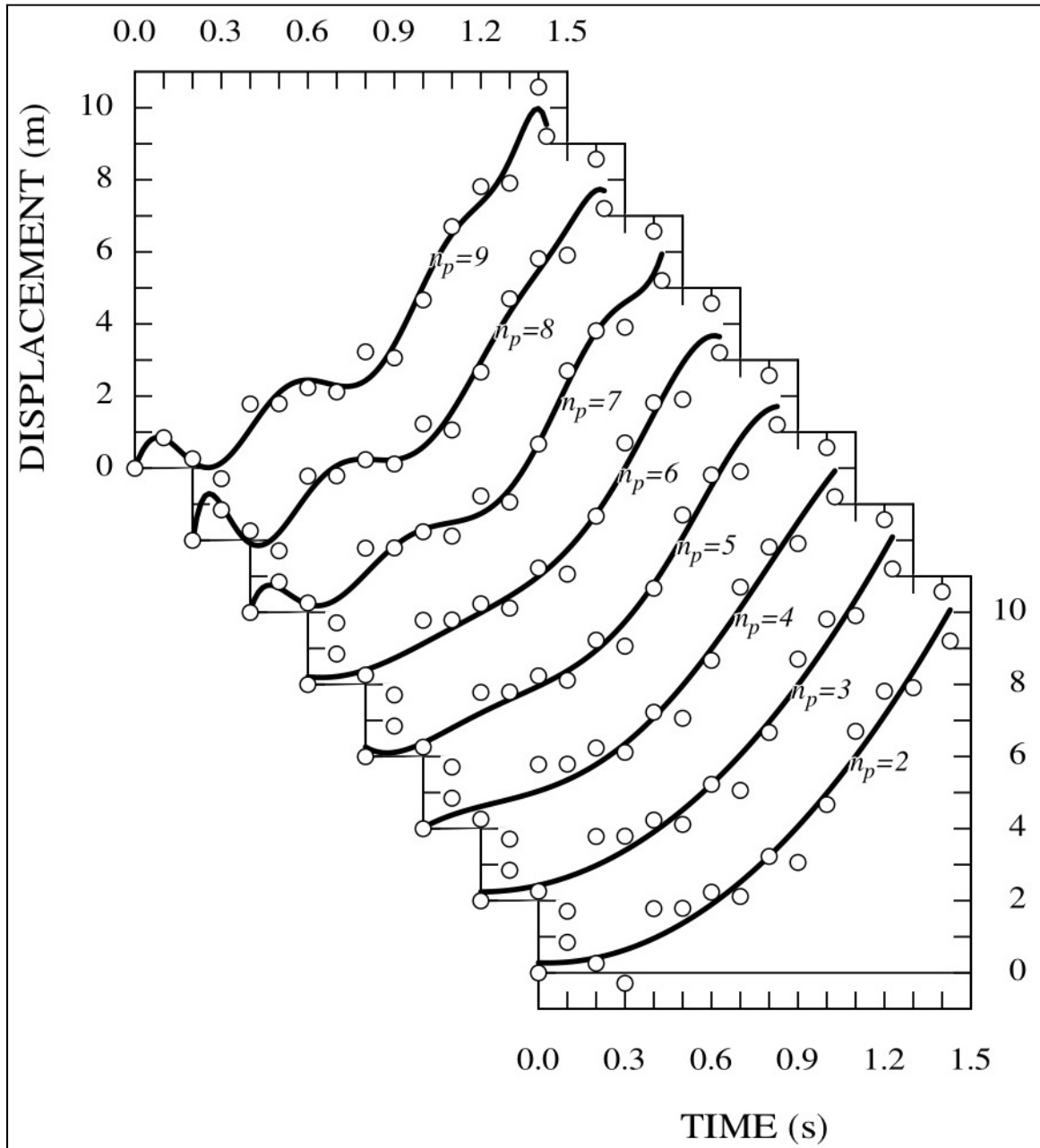


Figure D-5. Least Squares Polynomial Regressions for $n_p = 2, 3, 4, 5, 6, 7, 8, 9$

Examining once again the plots of displacement and velocity as a function of time in Figure D-4, it is obvious that this 2nd order regression polynomial does not satisfy either equation (6) or (7); that is to say, the displacement and velocity are non-zero at time equal zero. Furthermore, the polynomial coefficients b_0 and b_1 are non-zero as shown in Figure D-2. Still, the acceleration $a = 2 \times b_2 = 10.055 \text{ m/s}^2$ is reasonably close to the true value of 9.81 m/s^2 .

Since the initial displacement and velocity are known, these could be used as constraints with the LSPRWC code to improve the regression. Figures D-6 through 9 present *X Window System* screen shots from LSPRWC code execution using again the observations of Table D-1 but with these constraints.

```
AS AN INTERACTIVE PROGRAM, LSPRWC IS SELF-EXPLANATORY AND PROMPTS FOR THE
NECESSARY INFORMATION. THE X-Y DATA TO BE FITTED MAY BE ENTERED BY NAMELIST
OR READ FROM A FORMATTED DISK FILE. PROGRAM LSPRWC WILL ALSO EVALUATE THE
RESULTANT LEAST SQUARES POLYNOMIAL AT PRESCRIBED VALUES OF X WHICH, AGAIN,
MAY BE ENTERED BY NAMELIST OR READ FROM A FORMATTED DISK FILE.
```

```
- ENTER/RETURN TO CONTINUE -
```

```
SELECT THE SOURCE OF INPUT FOR THE X-Y DATA TO BE FITTED FROM
THE FOLLOWING LIST:
```

- 1, FOR KEYBOARD INPUT VIA NAMELIST
- 2, FOR FORMATTED DISK FILE INPUT

```
NOTE: THE X-Y DATA MUST BE MONATONICALLY INCREASING IN X.
2
```

```
DATA SETS ARE INPUT FROM FORMATTED DISK FILES AS (X,Y) PAIRS WHERE:
```

```
X      = X VALUE
Y      = Y VALUE
```

```
INPUT THE FILE NAME OF THE FORMATTED DISK FILE DATA SET.
trajectory.dat
```

```
INPUT THE DATA FILE FORMAT (INCLUDE PARENTHESES)
```

```
NOTE: AN EXAMPLE FORMAT IS "(2E15,6)"
```

```
ENTER"(*)" FOR A FREE FIELD READ (DEFAULT FORMAT)
(E15,6,T61,E15,6)
```

```
SHOULD THE X-Y DATA BE DISPLAYED FOR VERIFICATION? (Y/N)
n
```

Figure D-6. LSPRWC Screen Shot—Part 1


```

ARE CONSTRAINTS DESIRED? (Y/N)
y
SELECT THE SOURCE OF INPUT FOR THE CONSTRAINTS FROM THE FOLLOWING LIST:
1, FOR KEYBOARD INPUT VIA NAMEDLIST
2, FOR FORMATTED DISK FILE INPUT

NOTE: THE CONSTRAINTS ARE COMPLETELY INDEPENDENT OF THE X-Y DATA POINTS TO BE
FITTED.
1
INPUT THE CONSTRAINTS (20 MAX) BY NAMEDLIST WHERE:

CV      = ARRAY OF POLYNOMIAL DERIVATIVE VALUES
NC      = ARRAY OF POLYNOMIAL DERIVATIVE ORDERS (CONSTRAINT ORDER)
XC      = ARRAY OF POLYNOMIAL DERIVATIVE LOCATIONS (CONSTRAINT X VALUES)

NOTE: NC VALUES MUST BE IN THE RANGE 0 TO NP.

CURRENT VALUES ARE:

$PARM CV=____,____,____,NC=____,____,____,XC=____,____,____$

XC=2*0.0,NC=0,1,CV=2*0.0$

SHOULD THE CONSTRAINTS BE DISPLAYED FOR VERIFICATION? (Y/N)
Y

      NO.           XC(I)   NC(I)           CV(I)
      1      0.0000000000000D+00      0      0.0000000000000D+00
      2      0.0000000000000D+00      1      0.0000000000000D+00

      - ENTER/RETURN TO CONTINUE -

INPUT THE POLYNOMIAL ORDER BY NAMEDLIST WHERE:

NP = THE POLYNOMIAL ORDER

NOTE: NP MUST BE IN THE RANGE OF 0 TO 9.
      THERE MUST BE AT LEAST NP+NCONST+1 X-Y DATA POINTS.
      THE MAXIMUM VALUE FOR NP IS      9.

CURRENT VALUES ARE:
$PARM      NP =              3$

NP=2$
      LEAST SQUARES POLYNOMIAL

      P(X)=B(0)+B(1)*X+B(2)*X**2+....+B(NP)*X**NP

      I           B(I)
      0      0.0000000000000D+00
      1      0.0000000000000D+00
      2      4.9614940868317D+00

      THE STANDARD DEVIATION FOR THIS POLYNOMIAL OF ORDER 2 IS  8.03870D-01

      - ENTER/RETURN TO CONTINUE -

```

Figure D-7. LSPRWC Screen Shot—Part 2

```

SHOULD DATA POINTS BE PRESCRIBED FOR EVALUATION OF THE LEAST SQUARES
POLYNOMIALS? (Y/N)
y

SELECT THE SOURCE OF INPUT FOR THE X DATA TO BE EVALUATED FROM THE FOLLOWING
LIST:

1, FOR KEYBOARD INPUT VIA NAMELIST
2, FOR FORMATTED DISK FILE INPUT
3, FOR INPUT OF MINIMUM, MAXIMUM, AND INTERVAL VALUES
3

INPUT THE X DATA MAXIMUM, MINIMUM, AND INTERVAL VALUES BY NAMELIST WHERE:

XINT  = X DATA POINT INTERVAL
XMAX  = MAXIMUM X DATA VALUE
XMIN  = MINIMUM X DATA VALUE

CURRENT VALUES ARE:
$PARM  XINT = 0.142784E+01,      XMAX = 0.142784E+01,
        XMIN = 0.000000E+00$

SHOULD THE X-Y DATA BE DISPLAYED FOR VERIFICATION? (Y/N)
n

SHOULD THE RESULTS OF THIS RUN BE QUICK-PLOTTED? (Y/N)
n

ENTER:
1, TO RESTART THE PROGRAM
2, TO CHANGE THE CONSTRAINTS
3, TO CHANGE THE POLYNOMIAL ORDER
4, TO CHANGE THE PRESCRIBED X VALUES FOR EVALUATION OF THE POLYNOMIAL
5, TO PLOT THE RESULTS
6, TO SAVE THE RESULTS ON A FORMATTED DISC FILE
7, TO STOP
6

INPUT THE FILE NAME OF THE FORMATTED DISK FILE.
np2_c.dat

SELECT THE TYPE OF OUTPUT FOR THE FORMATTED DISK FILE FROM THE FOLLOWING LIST:
1, FOR ALL INPUT DATA AND RESULTS
2, FOR ONLY THE EVALUATED DATA
2

INPUT THE EVALUATED DATA VARIABLE LIST FOR OUTPUT BY NAMELIST WHERE:

"X"   = X VALUES
"0"   = 0th DERIVATIVE OF Y AT X
"1"   = 1th DERIVATIVE OF Y AT X
"2"   = 2th DERIVATIVE OF Y AT X
"R"   = RADIUS OF CURVATURE AT X
"S"   = ARC LENGTH

NOTE: VARIABLES WILL BE LISTED IN THE ORDER OF INPUT

CURRENT VALUES ARE:

$PARM AList= "X","0"$
AList="X","0","1","2"$

```

Figure D-8. LSPRWC Screen Shot—Part 3

```

INPUT THE DATA FILE FORMAT (INCLUDE PARENTHESES)

NOTE: AN EXAMPLE FORMAT IS "(F15.7,E15.6)"
      THE DEFAULT FORMAT IS:
(13E15.6)

ENTER:
1, TO RESTART THE PROGRAM
2, TO CHANGE THE CONSTRAINTS
3, TO CHANGE THE POLYNOMIAL ORDER
4, TO CHANGE THE PRESCRIBED X VALUES FOR EVALUATION OF THE POLYNOMIAL
5, TO PLOT THE RESULTS
6, TO SAVE THE RESULTS ON A FORMATTED DISC FILE
7, TO STOP
7

```

Figure D-9. LSPRWC Screen Shot—Part 4

Following along through Figure D-6, the user was presented with the introduction to LSPRWC, the data source was selected as a formatted disk file, and the file name and read format were entered.

Continuing through Figure D-7, the code asked if problem constraints were desired and a positive response was entered with the constraints defined by means of NAMELIST. The constraints were then displayed for verification to confirm that the 0th and 1st derivatives were set to zero at time zero. The polynomial order was again set to $n_p = 2$ and with the problem thus fully defined, program LSPRWC returned values for the coefficients of the least squares regression polynomial.

Following now through Figure D-8, this second-order regression polynomial was evaluated at specific values of x again using the minimum, maximum, and interval option. Menu options were then chosen to output the evaluated data to a formatted disk file writing, in order, each value of x , y , and the 1st and 2nd derivatives.

Finally, in Figure 9, the LSPRWC code menu option was selected for program termination.

The results from this LSPRWC run are shown in Figure D-10 as plots of displacement and velocity as a function of time along with the true values and the measurements of Table D-1. This 2nd order least squares regression polynomial agrees exceptionally well with the true relations for displacement and velocity having constraints imposed to satisfy equations (6) and (7).

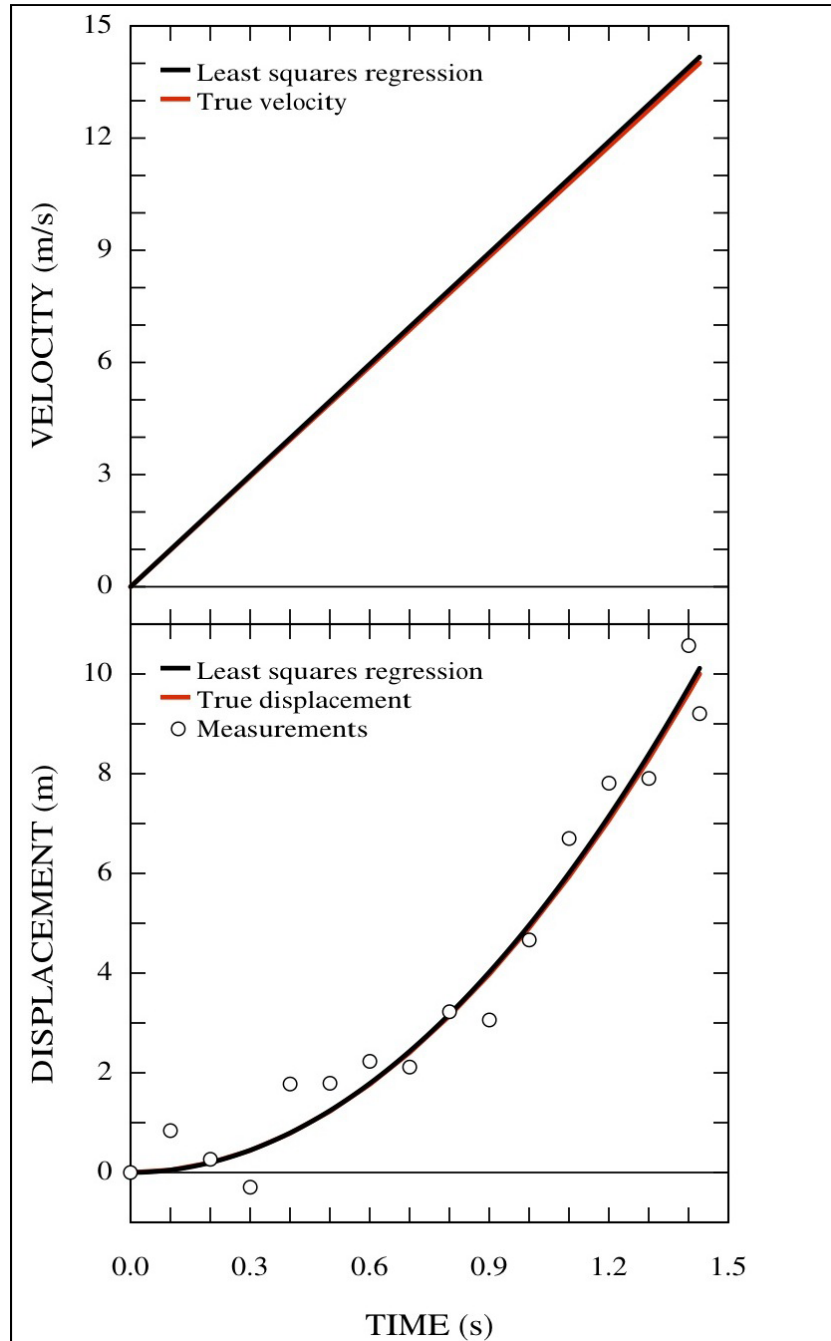


Figure D-10. Least Squares Polynomial Regression for $n_p = 2$ with Constraints

Indeed, the polynomial coefficients b_0 and b_1 are now zero valued, as shown in Figure D-7. Furthermore, the acceleration, $a = 2 \times b_2 = 9.92 \text{ m/s}^2$, is much closer to the true value of 9.81 m/s^2 demonstrating the value of added constraints to the least squares polynomial regression procedure when they are known.

APPENDIX E
EXAMPLE 2

For this example, the LSPRWC code was applied to the representation of a set of observations artificially generated through the application of random dispersions to a hyperbolic spiral. The hyperbolic spiral in polar coordinates is given by

$$R = a/\theta \quad (1)$$

or in Cartesian coordinates

$$x = a/\cos(\theta), \quad (2)$$

$$y = a/\sin(\theta), \quad (3)$$

where

a = constant,

R = radius,

θ = angle,

x = ordinate,

y = abscissa.

A partial list of the observations for this example are given in Table E-1, along with the true values for the x and y coordinates. Note that these observations were generated using equation (1) for equal 1° increments of θ from 90° through 720° with a dispersion applied to R within a uniformly distributed random resolution. These polar coordinates were then converted to Cartesian coordinates to produce the values given in Table E-1.

Table E-1. Observations, Cartesian Coordinates

True X	True Y	Measured X	Measured Y
0.000000E+00	0.636620E+00	0.000000E+00	0.637909E+00
-0.109885E-01	0.629528E+00	-0.106533E-01	0.610330E+00
-0.217347E-01	0.622401E+00	-0.207991E-01	0.595609E+00
-0.322433E-01	0.615239E+00	-0.328693E-01	0.627184E+00
-0.425186E-01	0.608045E+00	-0.434283E-01	0.621054E+00
-0.525648E-01	0.600818E+00	-0.500020E-01	0.571525E+00
-0.623858E-01	0.593562E+00	-0.630342E-01	0.599730E+00
-0.719856E-01	0.586275E+00	-0.722125E-01	0.588124E+00
-0.813677E-01	0.578961E+00	-0.751539E-01	0.534748E+00
-0.905357E-01	0.571620E+00	-0.929371E-01	0.586782E+00
-0.994931E-01	0.564253E+00	-0.984405E-01	0.558284E+00
-0.108243E+00	0.556862E+00	-0.113103E+00	0.581865E+00
-0.116789E+00	0.549448E+00	-0.118323E+00	0.556665E+00
-0.125133E+00	0.542013E+00	-0.128515E+00	0.556658E+00
-0.133280E+00	0.534556E+00	-0.139035E+00	0.557639E+00
-0.141231E+00	0.527081E+00	-0.143160E+00	0.534281E+00
-0.148989E+00	0.519587E+00	-0.157430E+00	0.549024E+00
-0.156558E+00	0.512077E+00	-0.161426E+00	0.528002E+00
-0.163939E+00	0.504551E+00	-0.171451E+00	0.527671E+00
-0.171135E+00	0.497011E+00	-0.169825E+00	0.493208E+00
-0.178148E+00	0.489458E+00	-0.163469E+00	0.449128E+00

0.772813E-01	-0.266101E-01	0.486087E-01	-0.167373E-01
0.776233E-01	-0.252213E-01	0.119837E+00	-0.389375E-01
0.779406E-01	-0.238288E-01	0.115467E+00	-0.353019E-01
0.782333E-01	-0.224330E-01	0.112810E+00	-0.323478E-01
0.785014E-01	-0.210344E-01	0.424245E-01	-0.113676E-01
0.787448E-01	-0.196333E-01	0.323353E-01	-0.806209E-02
0.789636E-01	-0.182302E-01	0.126867E+00	-0.292896E-01
0.791578E-01	-0.168255E-01	0.104055E+00	-0.221176E-01
0.793274E-01	-0.154197E-01	0.893899E-01	-0.173756E-01
0.794723E-01	-0.140131E-01	0.111402E+00	-0.196431E-01
0.795926E-01	-0.126062E-01	0.818384E-01	-0.129619E-01
0.796885E-01	-0.111995E-01	0.101321E+00	-0.142397E-01
0.797598E-01	-0.979327E-02	0.902876E-01	-0.110859E-01
0.798066E-01	-0.838801E-02	0.470222E-01	-0.494223E-02
0.798290E-01	-0.698413E-02	0.865510E-01	-0.757223E-02
0.798271E-01	-0.558206E-02	0.116908E+00	-0.817499E-02
0.798009E-01	-0.418219E-02	0.421504E-01	-0.220901E-02
0.797505E-01	-0.278495E-02	0.118973E+00	-0.415463E-02
0.796760E-01	-0.139075E-02	0.544780E-01	-0.950917E-03
0.795775E-01	0.000000E+00	0.751481E-01	0.000000E+00
0.795775E-01	0.000000E+00	0.798177E-01	0.000000E+00

The full set of observations from Table E-1 are presented in the plot of Figure E-1 and show immediately that this set of (x, y) data pairs or observations will not be suitable for a least squares polynomial regression. The true position data depicting the hyperbolic spiral, as indicated in red, exhibits locations of infinite slope which cannot be represented with polynomials. Furthermore, the (x, y) observations may well be multivalued for a given x location, as is certainly the case for the hyperbolic spiral. Although artificially generated specifically for this example, similar Cartesian coordinate data sets do occur and pose similar problems for regression.

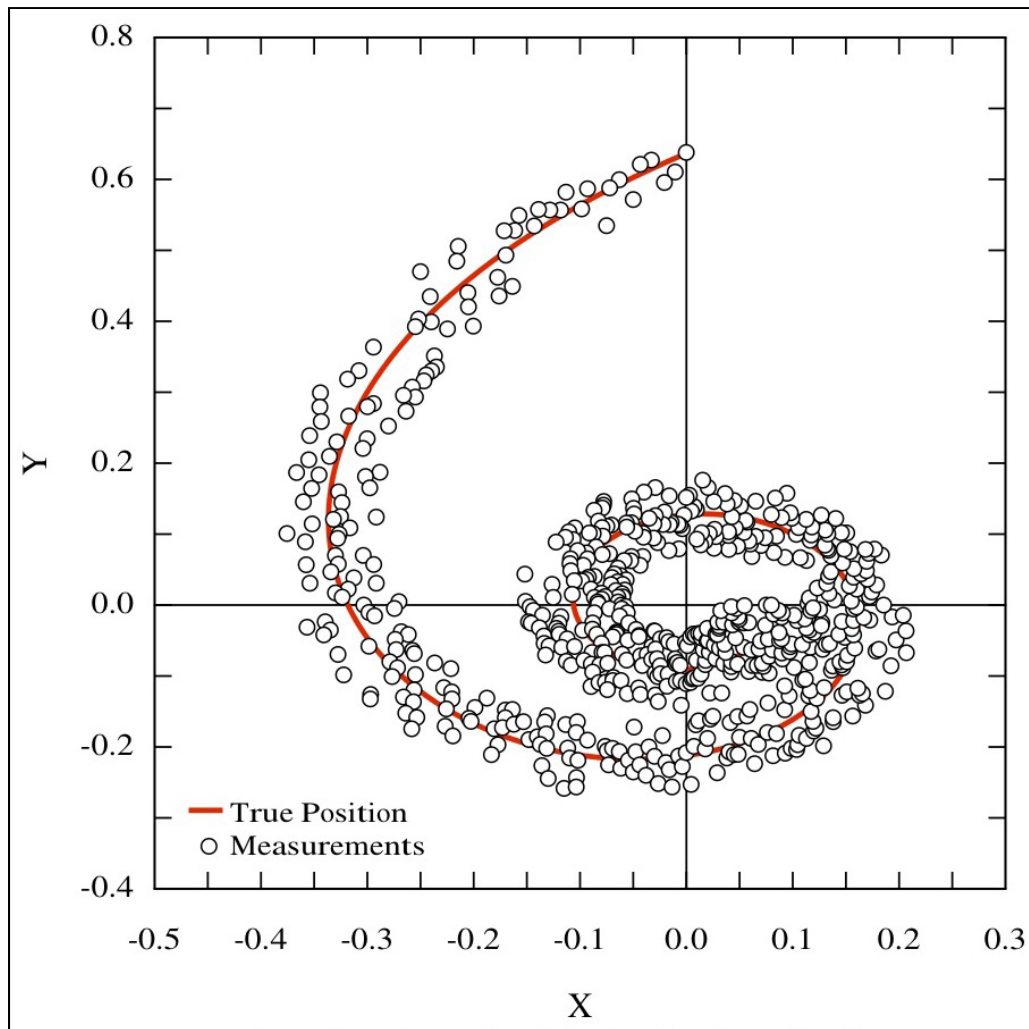


Figure E-1. Measured and True Position in (x, y) Coordinates

Something must be done to fit this (x, y) data set of Table E-1 with a least squares polynomial regression and this is accomplished, at least for this example, through a coordinate transformation. Figure E-2 shows the data in polar coordinates, data which is clearly well behaved and amenable to treatment with the method of least squares polynomial regression.

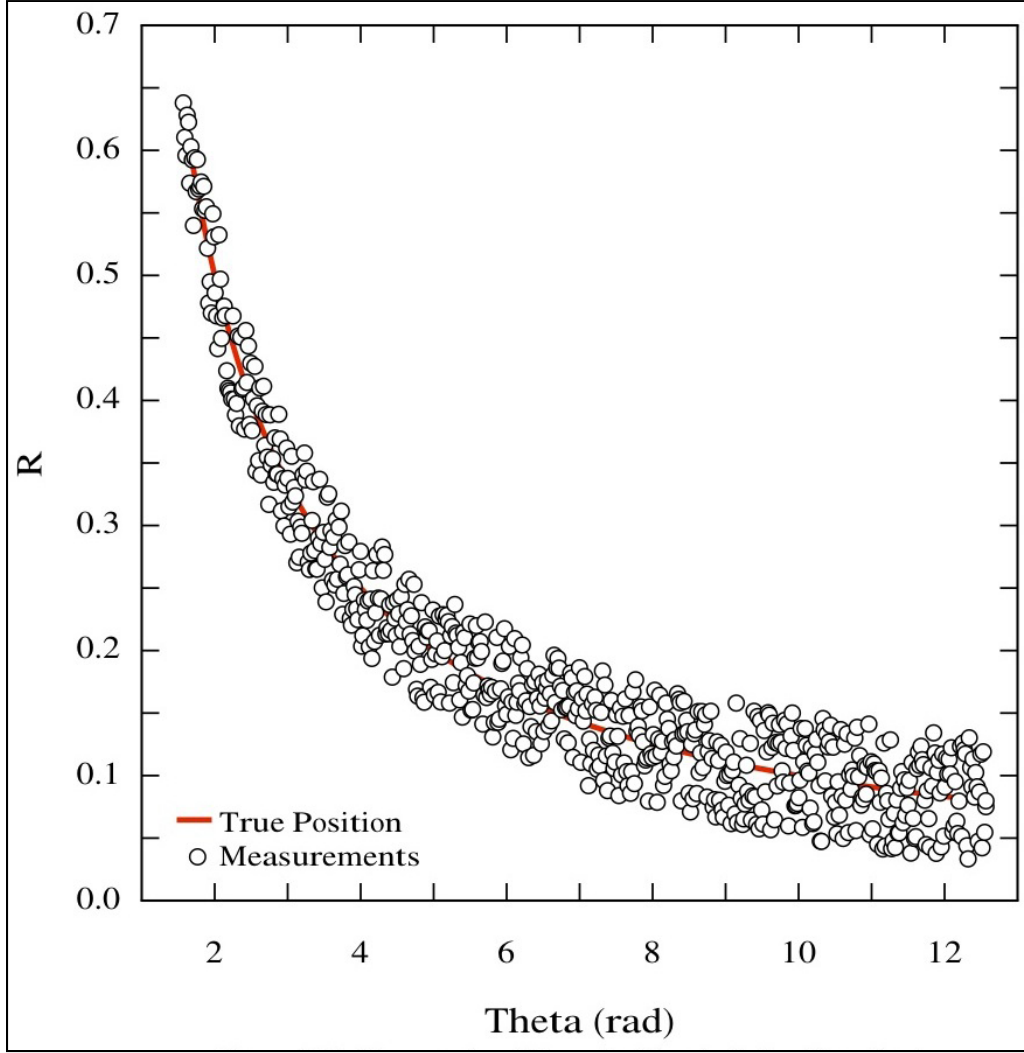


Figure E-2. Measured and True Position in Polar Coordinates

The LSPRWC code was then run with the observations in the polar coordinates of Figure E-2 for polynomial orders from 2 through 5 to produce the results of Figure E-3. This plot of R versus θ shows a reasonable fit for the regression polynomials, although none truly duplicate the hyperbolic spiral a/θ as shown in the figure. Perhaps this should not be unanticipated since R is a function of θ to the -1 power, while the LSPRWC code limits the n_p order regression polynomial, in this case, to

$$R(\theta) = \sum_{j=0}^{n_p} b_j \theta^j$$

for $0 \leq n_p \leq 9$.

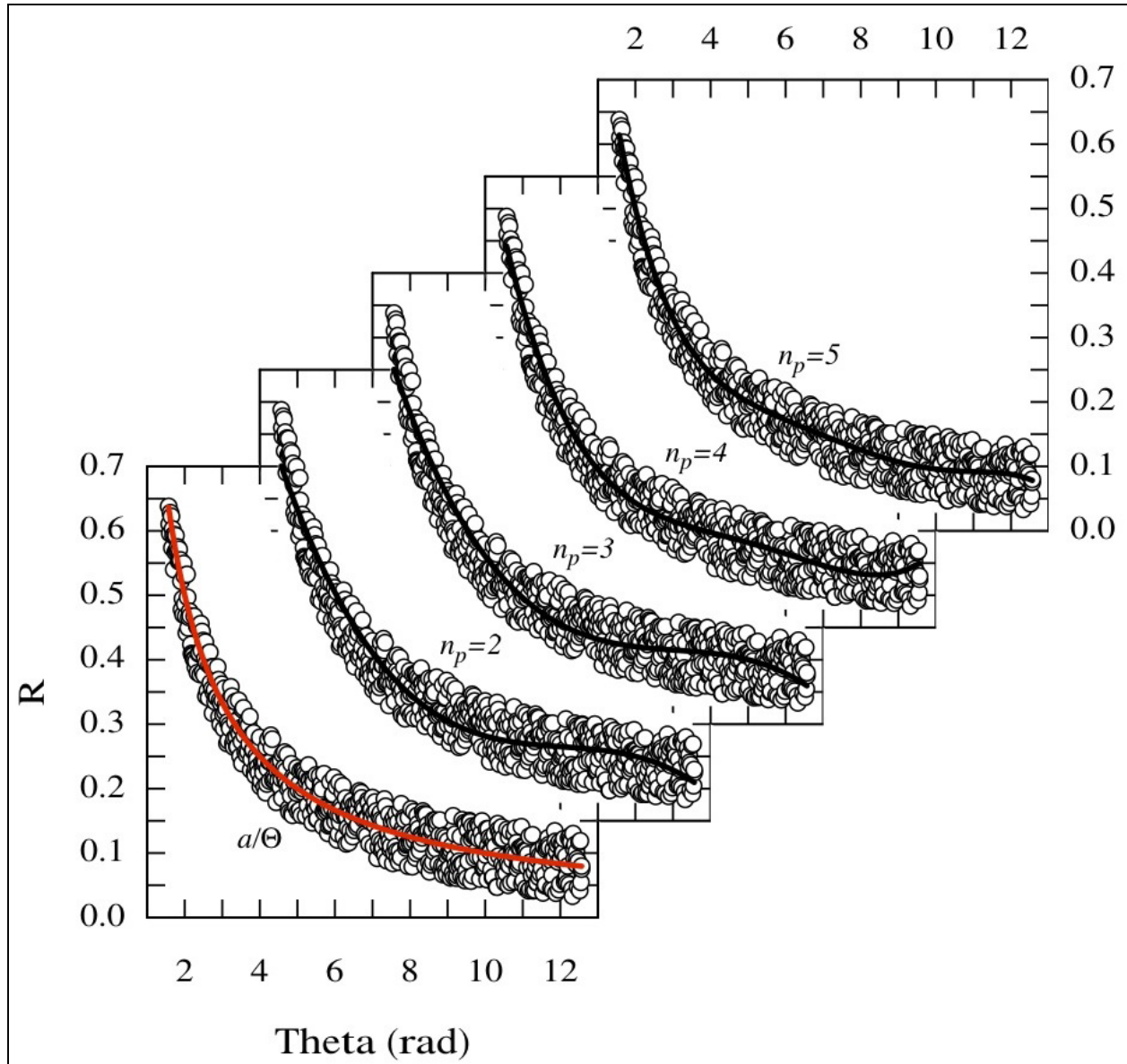


Figure E-3. Least Squares Polynomial Regression for $n_p = 2, 3, 4, 5$

Assuming the 5th order regression polynomial to give the best fit, a set of evaluations from that polynomial over the range $90^\circ \leq \theta \leq 720^\circ$ were converted to Cartesian coordinates and plotted in Figure E-4 along with the observations and true values from Table E-1. Surprisingly, the fit is quite good with the largest departure from the theoretical occurring for values of θ near 90° rather than 720° , as might be surmised from Figure E-2.

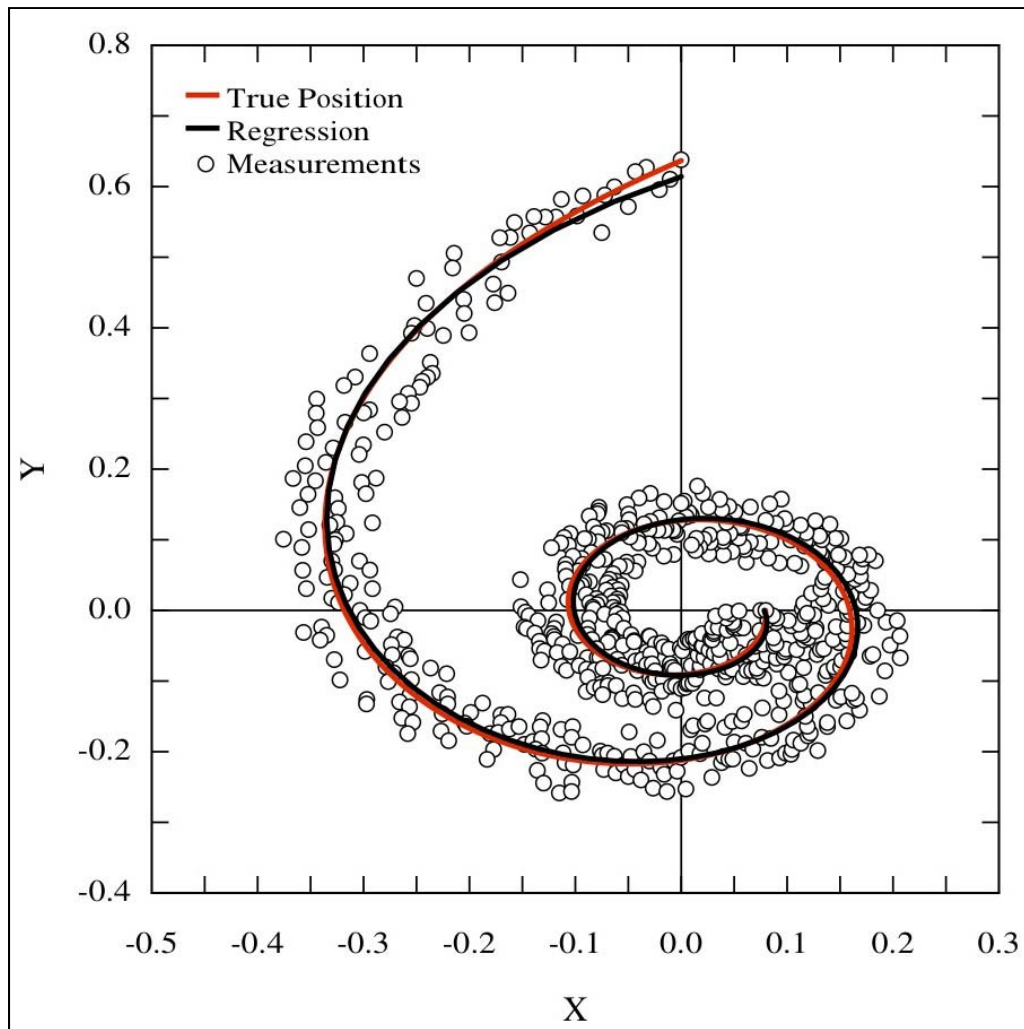


Figure E-4. Least Squares Polynomial Regression for $n_p = 5$

INITIAL DISTRIBUTION LIST

		<u>Copies</u>
Weapon Systems Technology Information Analysis Center Alion Science and Technology 201 Mill Street Rome, NY 13440	Ms. Gina Nash gnash@alionscience.com	Electronic
Defense Technical Information Center 8725 John J. Kingman Rd., Suite 0944 Fort Belvoir, VA 22060-6218	Mr. Jack L. Rike jrike@dtic.mil	Electronic
AMSAM-L	Ms. Anne C. Lanteigne anne.lanteigne@us.army.mil	Electronic
	Mr. Michael K. Gray michael.k.gray@us.army.mil	Electronic
RDMR		Electronic
RDMR-CSI		Electronic
RDMR-SS	Mr. Gregg Tackett Gregory.tackett@us.army.mil	Electronic
	Mr. Mike Eisen mike.eisen@us.army.mil	Electronic
RDMR-SSM-A	Mr. Lamar Auman lamar.auman@us.army.mil	Electronic
	Dr. Clark Mikkelsen clark.mikkelsen@us.army.mil	Electronic/Hardcopy