<table>
<thead>
<tr>
<th>Title or Description</th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Date</td>
<td>10 March 1979</td>
<td></td>
</tr>
<tr>
<td>Dept. No.</td>
<td>20878</td>
<td></td>
</tr>
<tr>
<td>Account</td>
<td>126-6242-41</td>
<td></td>
</tr>
<tr>
<td>Signature</td>
<td>Virginia/Edith Huang</td>
<td></td>
</tr>
<tr>
<td>Department</td>
<td>Computer Science</td>
<td></td>
</tr>
<tr>
<td>Room</td>
<td>51008</td>
<td></td>
</tr>
<tr>
<td>Phone</td>
<td>3402</td>
<td></td>
</tr>
</tbody>
</table>

**Reproduction Requirements**

- **Number of Pages**: 20
- **Number of Copies**: 40

**Type of Paper Stock**

- **Paper Size**: 8½ x 11
- **Paper Colour**: White
- **Printing**: 1 Side
- **Binding/Finishing Operations**: Collating

**SPECIAL INSTRUCTIONS**

- Please enclose in front and back covers
- supplied and staple with three staples down the left-hand side.

by

M.O. Afolabi

Computer Science Dept.
Research Report CS-79-12

University of Waterloo
Waterloo, Ontario, CANADA

March 1979

†Present address: Department of Computer Sciences
University of Lagos
Lagos
NIGERIA
The Design and Implementation of a Package for
Symbolic Series Solution of Ordinary Differential Equations

Abstract

This paper describes the design and implementation of SODEPACK, the Symbolic Ordinary Differential Equation Package, for computing a differential equation (ODE). The approximation is expressed in terms of the indeterminates (if any) present in the ODE or its associated conditions. A very simple user/package interface is provided which enables the user to specify his problem in a natural form. The output from the package is in a clear, readable form.
1. Introduction

This paper describes the structure, design, and implementation of SODEPACK, the Symbolic Ordinary Differential Equation Package, which is designed to compute a polynomial approximation of given degree to a \( v \)-order ODE of the form

\[
G(x,y(x),y'(x),...,y^{(v)}(x)) = 0,
\]

where \( G \) is a polynomial function in \( x, y(x), y'(x),...,y^{(v)}(x) \). The \( v \) conditions associated with the ODE may be quite arbitrary. In general, any linear combination of function and derivative (up to order \( v-1 \)) values at one or more points is allowed. The polynomial approximation is expressed in terms of the indeterminates (if any) present in the ODE or its associated conditions.

Programs for symbolic computation usually involve some fairly complex data and storage structures. A user of these programs would like to specify his problem in a 'natural' form and not be concerned with how the problem is 'massaged' to conform to a format or structure required by a particular programming language and/or computer installation. The output must also be in a 'human-readable' form which reflects the nature of the problem being solved. Even in a package for numerical computation these features are highly desirable, (see, for example, [3,4]).

SODEPACK insulates the user from most system considerations, accepting the ODE to be solved (together with the associated conditions) in a reasonably natural form. The output is also in a clear, readable and adequately documented form. The package provides a variety of
methods for solving ODE's and we offer suggestions for selecting a method for a given problem.

But perhaps the most attractive feature of the package is the simplicity of the user/package interface. The main programs calling the routines in the package are very simple, consisting of less than ten ALTRAN [2] statements, for most problems. The package may be used to solve several problems using a particular method, in one run.

In Section 2 we shall describe the basic structure of SODEPACK. A more detailed description may be found in [1], which also contains examples of mainline programs. In Section 3 we give the listings of the procedures in the package.
2. Basic Structure of the Package

SODEPACK offers a collection of methods for solving general \( n \)-order ODE's of the form given in Section 1. For all the methods in the package, the user and the package interact to solve the ODE problem through the following simple steps:

Step 1: The user calls the module initialization procedure for a particular method, thereby defining the structure of \( G \) and the associated conditions.

Step 2: The user calls the module for the appropriate method (see Figure 1). The package reads in the problem and outputs the polynomial approximation of required degree in both the power and Chebyshev series forms, together with timing statistics and error estimates (where applicable).

The different methods provided in the package correspond to different algorithms for obtaining the polynomial approximation for linear and nonlinear ODE's. The general structure of the methods used in the package may be represented graphically by Figure 1.

On the whole, there are four methods and the user chooses a particular method by calling the appropriate routines in Steps 1 and 2 above. The methods are distinguished by the last two characters of the procedure names as follows (YYYY indicates the function performed by the procedure):
Figure 1  Overall Structure of Methods in SODEPACK

The procedures which make up SODEPACK are divided into six levels as given in Figure 2.
<table>
<thead>
<tr>
<th>LEVEL</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
</tr>
</thead>
<tbody>
<tr>
<td>PINTXi</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>INPCON</td>
<td>INFORM</td>
</tr>
<tr>
<td>SODEXi</td>
<td></td>
<td>PROX</td>
<td>RECXij</td>
<td></td>
<td>INPODE</td>
<td>POWERS</td>
</tr>
<tr>
<td></td>
<td>SOLXi</td>
<td></td>
<td>GSXij</td>
<td>PSXi</td>
<td>PRODK</td>
<td>TPOWS</td>
</tr>
<tr>
<td></td>
<td>EPIX</td>
<td></td>
<td>GSXij</td>
<td></td>
<td>INT</td>
<td>EVAL</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>INIT</td>
<td>TRUNC</td>
</tr>
<tr>
<td></td>
<td>EPIX</td>
<td></td>
<td></td>
<td></td>
<td>CANONP</td>
<td>CHFORM</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>SETP1</td>
<td>TCS</td>
</tr>
<tr>
<td></td>
<td>EPIX</td>
<td></td>
<td></td>
<td></td>
<td>SETP2</td>
<td>ABS</td>
</tr>
<tr>
<td></td>
<td>EPIX</td>
<td></td>
<td></td>
<td></td>
<td>INTEGR</td>
<td>LQABS</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>ERREST</td>
<td>AROUND</td>
</tr>
<tr>
<td></td>
<td>EPIX</td>
<td></td>
<td></td>
<td></td>
<td>etc.</td>
<td>QROUND</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>etc.</td>
<td>etc.</td>
</tr>
</tbody>
</table>

**Figure 2.** Overall Structure of SODEPACK

**Level 1** These procedures form the user/package interface. PINTXi are the package initialization routines for the different methods, while SODEXi are the actual implementations of the methods.

**Level 2** This level consists of the procedures called by SODEXi, namely the prologue, PROX, for inputting and checking data, SOLXi for solving the problem, and EPIX for output requirements.

**Level 3** This level consists of procedures used by SOLXi to generate the solution required. RECXij are the routines for symbolic generation of the relevant recurrence equations. The procedures GSXi compute the required general solution of the recurrence equations.
Level 4 This level consists of the different procedures used by RECXij and GSXij corresponding to the form of the ODE and the solution codes (see [1, Section 5.6]) specified for the problem. The procedures PSXij compute the parametric solutions of the problem and VALXij assign values to the parameters present in these solutions.

Level 5 This level consists of the **package utility** procedures, that is, procedures called by any of the procedures in levels 1-4. The use of these procedures is restricted to SODEPACK. Examples are .INPODE and INPCON which read in a differential equation and its associated conditions, respectively.

Level 6 This level consists of the **general utility** procedures, i.e. procedures called by any of the procedures in levels 1-5 above. These are very basic procedures that could very well form part of a general system function library. Examples are the absolute value routines ABS (real), LQABS (long rational), and Chebyshev series evaluation routine EVAL, for Clenshaw's rule.

Although all the levels are shown in Figure 2, we point out again that only the procedures in Level 1 are visible to the user.

**Restrictions and Assumptions**

1. The package assumes that the solution is required in a finite interval \([a,b]\). However, the indeterminates appearing in the ODE are in the interval \([-1,1]\).

2. The order of the ODE to be solved may not exceed 5.

3. For linear ODE's the degree of approximation may not exceed 20; for nonlinear ODE's the degree may not exceed 15.

4. The degree of the ODE may not exceed 4.
3. **Listings of Procedures**

   In this section we give the listings of the procedures SODEPACK. The procedures are described in more detail in [1].

<table>
<thead>
<tr>
<th>Level</th>
<th>Procedures</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Level 1</td>
<td>Procedures</td>
<td>8</td>
</tr>
<tr>
<td>Level 2</td>
<td>Procedures</td>
<td>21</td>
</tr>
<tr>
<td>Level 3</td>
<td>Procedures</td>
<td>34</td>
</tr>
<tr>
<td>Level 4</td>
<td>Procedures</td>
<td>45</td>
</tr>
<tr>
<td>Level 5</td>
<td>Procedures</td>
<td>60</td>
</tr>
<tr>
<td>Level 6</td>
<td>Procedures</td>
<td>92</td>
</tr>
</tbody>
</table>
3. Listings of Procedures

In this section we give the listings of the procedures in Levels 1, 2, and 3 of SODEPACK (see Figure 5.3 of Section 5.2). The procedures are described in more detail in Section 5.3. The complete listings of all the procedures in the package may be found in [Afo79].

<table>
<thead>
<tr>
<th>Level</th>
<th>Procedures</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Level 1</td>
<td>Procedures</td>
<td>B-2 8</td>
</tr>
<tr>
<td>Level 2</td>
<td>Procedures</td>
<td>B-29</td>
</tr>
<tr>
<td>Level 3</td>
<td>Procedures</td>
<td>B-36</td>
</tr>
<tr>
<td>Level 4</td>
<td>Procedures</td>
<td></td>
</tr>
<tr>
<td>Level 5</td>
<td>Procedures</td>
<td></td>
</tr>
<tr>
<td>Level 6</td>
<td>Procedures</td>
<td></td>
</tr>
<tr>
<td>Procedure</td>
<td>Page</td>
<td></td>
</tr>
<tr>
<td>-----------</td>
<td>------</td>
<td></td>
</tr>
<tr>
<td>PINTC1</td>
<td>9</td>
<td></td>
</tr>
<tr>
<td>PINTT1</td>
<td>10</td>
<td></td>
</tr>
<tr>
<td>PINTT2</td>
<td>11</td>
<td></td>
</tr>
<tr>
<td>PINTN1</td>
<td>12</td>
<td></td>
</tr>
<tr>
<td>SODEC1</td>
<td>13</td>
<td></td>
</tr>
<tr>
<td>SODET1</td>
<td>15</td>
<td></td>
</tr>
<tr>
<td>SODET2</td>
<td>17</td>
<td></td>
</tr>
<tr>
<td>SODEN1</td>
<td>19</td>
<td></td>
</tr>
</tbody>
</table>
List of Level 1 Procedures

<table>
<thead>
<tr>
<th>Procedure</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>PINTC1</td>
<td>B-3</td>
</tr>
<tr>
<td>PINTC2</td>
<td>B-4</td>
</tr>
<tr>
<td>PINTT1</td>
<td>B-5</td>
</tr>
<tr>
<td>PINTT2</td>
<td>B-6</td>
</tr>
<tr>
<td>PINTN1</td>
<td>B-7</td>
</tr>
<tr>
<td>*PKINIT</td>
<td>B-8</td>
</tr>
<tr>
<td>SODEC1</td>
<td>B-10</td>
</tr>
<tr>
<td>SODEC2</td>
<td>B-12</td>
</tr>
<tr>
<td>SODET1</td>
<td>B-14</td>
</tr>
<tr>
<td>SODET2</td>
<td>B-16</td>
</tr>
<tr>
<td>SODEN1</td>
<td>B-18</td>
</tr>
</tbody>
</table>

*Not a Level 1 Procedure, but included here because it performs most of the work required in PINTX1.
PROCEDURE PINTCI
INTEGER ARRAY DEFLT.

(This procedure sets the default cell size for solder module.
Needs to be set at the module level. Manual has detailed
steps on how to call the procedure. Be sure to perform the actual
initialization.

Package procedure required: #KINT.
Declaration in calling procedure.

END # END OF PROCEDURE PINTCI.
PROCEDURE PINIT
    INTEGER ARRAY DEFLT
    \# THIS PROCEDURE SETS THE DEFAULT VALUES FOR SOLVER MODULE
    \# SODET1 FOR THE VARIABLES MAXORD,MAXDEG,MAXVAL,MAXMU,MAXCOD,
    \# AND CALLS PROCEDURE PKINIT TO PERFORM THE ACTUAL
    \# INITIALIZATION.
    \# PACKAGE PROCEDURE REQUIRED: PKINIT.
    \# DECLARATION IN CALLING PROCEDURE: NONE.

    DEFLT = (5,20,5,1,4)
    PKINIT(DEFLT)

END \# END OF PROCEDURE PINIT.
PROCEDURE PINTT2
INTEGER ARRAY DEFLT

This procedure sets the default values for SODET2 module
for the variables MAXORD, MAXDEG, MAXVAL, MAXMU, MAXCOD,
and calls procedure PKINIT to perform the actual
initialization.
Package procedure required: PKINIT.
Declaration in calling procedure: NONE.

DEFLT = (5,20,51,4)
PKINIT(DEFLT)

END  # END OF PROCEDURE PINTT2.
PROCEDURE PINTNL
  INTEGER ARRAY DEFLT
  /* THIS PROCEDURE SETS THE DEFAULT VALUES FOR SOLVER MODULE
  SODENL FOR THE VARIABLES MAXORD,MAXDEG,MAXVAL,MAXMU,MAXCOD,
  AND CALLS PROCEDURE PKINIT TO PERFORM THE ACTUAL
  INITIALIZATION.
  PACKAGE PROCEDURE REQUIRED: PKINIT.
  DECLARATION IN CALLING PROCEDURE: NONE.
  */
  DEFLT = (5,15,5,4,4)
  PKINIT(DEFLT)
END /* END OF PROCEDURE PINTNL.*/
PROCEDURE SODEC1

# MAIN SOLVER MODULE FOR THE CHEBYSHEV SERIES SOLUTION OF
# A LINEAR ODE.
# INPUT.
# THE FOLLOWING INDETERMINATES MAY BE PRESENT IN THE INPUT:
# X - THE VARIABLE IN THE POLYNOMIAL COEFFICIENTS;
# Y - THE UNKNOWN SOLUTION FUNCTION;
# DY - DY(I) REPRESENTS THE I-TH DERIVATIVE OF Y;
# YX - YX(I) REPRESENTS THE VALUE OF Y AT A POINT X(I);
# DXY - DXY(I,J) REPRESENTS THE VALUE OF DY(I) AT A POINT X(J);
# MU - MU(I) REPRESENTS THE UNSPECIFIED INDETERMINATE APPEARING
# IN THE PROBLEM;
# THE INPUT DECK BEGINS WITH
# NUM - THE NUMBER OF PROBLEMS TO BE SOLVED.
# FOR EACH PROBLEM THE INPUT SEQUENCE IS AS FOLLOWS:
# ORDER - THE ORDER OF THE ODE;
# NMAX - THE DEGREE OF POLYNOMIAL APPROXIMATION DESIRED;
# RANGE - THE FINITE INTERVAL (A B) IN WHICH THE SOLUTION IS DESIRED;
# CODE - THE SOLUTION CODE FOR THE PROBLEM;
# DIFFEQ - THE ODE AS A MULTINOMIAL IN THE INDETERMINATES X,Y.
# YX(1),...YX(OFF); (ORDER)
# CONDN - THE ASSOCIATED CONDITIONS; CONDN(1),...CONDN(OFF), WHERE
# AS MULTINOMIALS IN THE INDETERMINATES YX(1),DY(1),
# 1 <= J <= MAXSUB, 1 <= I <= ORDER-1, WHERE MAXSUB IS
# THE NUMBER OF X-VALUES APPEARING IN CONDN;
# XSUB - RATIONAL NUMBERS XSUB(1),...XSUB(MAXSUB) REPRESENTING
# THE X-VALUES ABOVE.

OUTPUT:
# IN ADDITION TO THE INDETERMINATES ABOVE, THE FOLLOWING
# INDETERMINATES MAY APPEAR IN THE OUTPUT FROM THE PACKAGE
# R - THE INDETERMINATE APPEARING AS THE INDEPENDENT VARIABLE
# IN THE RECURRENCE EQUATION;
# CR - CR(J) REPRESENTS THE TERM CR(J) IN THE RECURRENCE EQUATION
# WHERE CR REPRESENTS AN INDETERMINATE;
# T - T(J) REPRESENTS THE J-TH CHEBYSHEV POLYNOMIAL.
# THE FIRST PART OF THE OUTPUT IS AN ECHO OF THE INPUT. IN ADDITION
# TO THE APPROXIMATE SOLUTIONS CHEBYSHEV COEFFICIENTS, AND THE
# PERTURBATION IMPOSED ON THE ODE, THE PACKAGE MAY PRINT OTHER
# INFORMATION DEPENDING ON THE VALUE OF THE MESSAGE LEVEL
# INDICATOR, MSGVL, AS FOLLOWS:
# MSGVL
# SUMMARY OF RESULTS OF COMPUTATION
# MORE DETAILED INFORMATION
# DETAILED INFORMATION PLUS TIMING STATISTICS
# DETAILED DEBUGGING INFORMATION.
# PACKAGE PROCEDURES REQUIRED: POWERS, PROCSCLEC
# SYSTEM PROCEDURE REQUIRED: TIME
# DECLARATION IN CLASSING PROCEDURE: NONE

RATIONAL ARRAY XSUB, RANGE
LOGICAL LINEAR
SET UP GLOBAL LAYOUT AS LONG ALGEBRAIC GLOBL.

LONG ALGEBRAIC (DY(0:MAXORD);YDEG,Y;YDEG,X;XDEG,
MUL:(1:MAXMUL));XDEG,DYX(1:MAXORD,1:MAXVAL);1,
YX(1:MAXVAL);1,TT(0:XDEG);1,PAR(0:MAXPAR);1,
R:XDEG,TR(-MAXDEG:MAXDEG);1,CR(-MAXDEG:MAXDEG);1) GLOBL
EXTERNAL LONG ALGEBRAIC XX=X, XY=Y, RR=R
EXTERNAL LONG ALGEBRAIC ARRAY XT=T,XPAR=PAR
EXTERNAL LONG ALGEBRAIC ARRAY XY=XY,XCR=CR,XTR=TR
EXTERNAL LONG ALGEBRAIC ARRAY XDY=DYX,XMU=MU
EXTERNAL LONG ALGEBRAIC ARRAY XDY=DYX,XMU=MU
EXTERNAL LONG ALGEBRAIC DIFFEQ=GLOBL,YNEW,RHS
EXTERNAL LONG ALGEBRAIC ARRAY CONDN=DIFFEQ,POLY,C
REAL TOLD,TI,TNEW
LONG ALGEBRAIC ALTRAN SOLCI
LONG ALGEBRAIC ARRAY ALTRAN POWERS

XPW = POWERS(XDEG)
READ NUM
DO I = 1,NUM
  PAGE()
  TIME(TOLD)
  TI = TOLD
  PROCN(MAX,ORDER,DIFFEQ,CODE,POLY,RHS,MAXSUB,
  XSUB,CONDN,RANGE)
  TNEW = TIME(TOLD)
  IF(MSGLVL>2) WRITE "TIME USED FOR PROLOGUE WAS", TNEW

# OBTAIN POLYNOMIAL SOLUTION, YNEW.

YNEW = SOLCI(ORDER,POLY,NMAX,RHS,MAXSUB,CONDN,XSUB,CODE,C)
TIME(TOLD)
EPIC(NMAX,YNEW,C,RANGE)
TNEW = TIME(TOLD)
IF(MSGLVL>2) WRITE "TIME USED FOR EPILOGUE WAS", TNEW

SYSERR: # CHECK ERROR STATE.
IERR = ERRNO()
IF (IERR<>0) DO
  IF(INPFLG<>1) DUMP()
  ELSE, DO; SNAP; OK() ; IERR=0; DOEND
DOEND
TNEW = TIME(TI)
IF(MSGLVL>2) WRITE "TOTAL TIME ELAPSED WAS", TNEW
WRITE ",", "-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+", " "
DOEND
RETURN
END # END OF PROCEDURE SODECI.
SET UP GLOBAL LAYOUT AS LONG ALGEBRAIC GLOBL.

LONG ALGEBRAIC (DY(0:MAXORD):YDEG,Y:YDEG,X:XDEG,
   MU(1:MAXI:MAXMU):XDEG,DYX(1:MAXORD,1:MAXVAL):1,
   YX(1:MAXVAL):1,T(0:XDEG):1,PAR(0:MAXPAR):1,
   RXDEG,TR(0:MAXDEG,MXDEG):1,CR(0:MAXDEG,MXDEG):1) GLOBL
EXTERNAL LONG ALGEBRAIC XN=X, XY=Y, RR=R,
EXTERNAL LONG ALGEBRAIC ARRAY XT=T, XPAR=PAR
EXTERNAL LONG ALGEBRAIC ARRAY XYX=XY, XCR=CR, XTR=TR
EXTERNAL LONG ALGEBRAIC ARRAY XDYX=DYX, XMU=MU
EXTERNAL LONG ALGEBRAIC ARRAY XDY=DY, XPWR, TPWR
EXTERNAL LONG ALGEBRAIC DIFSEQ=GLOBL, YNEW, RHS
EXTERNAL LONG ALGEBRAIC ARRAY CONDN=DIFSEQ, POLY, C
REAL TOLD, T1, TNEW
LONG ALGEBRAIC ALTRAN SOLCI
LONG ALGEBRAIC ARRAY ALTRAN POWERS

XPWR = POWERS(XDEG)
READ NUM
DO 1 = 1, NUM
   PAGE()
   TIME(TOLD)
   T1 = TOLD
   PROC(NMAX, ORDER, DIFSEQ, CODE, POLY, RHS, MAXSUB,
        XSUB, CONDN, RANGE)
   TNEW = TIME(TOLD)
   IF(MSG_LVL>2) WRITE " TIME USED FOR PROLOGUE WAS", TNEW

# OBTAIN POLYNOMIAL SOLUTION, YNEW.

YNEW = SOLCI(ORDER, POLY, NMAX, RHS, MAXSUB, CONDN, XSUB, CODE, C)

TIME(TOLD)
EPIC(NMAX, YNEW, C, RANGE)
TNEW = TIME(TOLD)
IF(MSG_LVL>2) WRITE " TIME USED FOR EPILOGUE WAS", TNEW

SYSERR: # CHECK ERROR STATE.
IERR = (IERRNO)
IF (IERR<0) DO
   IF(INPPLG<>0) DUMP()
   ELSE DO, SNAP, OK(), IERR=0; DOEND
DOEND
TNEW = TIME(T1)
IF(MSG_LVL>2) WRITE "TOTAL TIME ELAPSED WAS", TNEW
WRITE ", " +++++++ ++++++++ ++++++++ ++++++++ ++++++++ ++++++++ ++++++++ ++++++++ ++++++++ ++++++++ ++++++++ ++++++++ ++++++++ ++++++++ ++++++++ ++++++++ ++++++++ ++++++++ ++++++++ ++++++++ ++++++++ ++++++++ ++++++++ ++++++++ ++++++++ ++++++++ ++++++++ ++++++++ ++++++++ ++++++++ ++++++++ ++++++++ ++++++++ ++++++++ ++++++++ ++++++++ ++++++++ ++++++++ ++++++++ ++++++++ ++++++++ ++++++++ ++++++++ ++++++++ ++++++++ ++++++++ ++++++++ ++++++++ ++++++++ ++++++++ ++++++++ ++++++++ ++++++++ ++++++++ ++++++++ ++++++++ ++++++++ ++++++++ ++++++++ ++++++++ ++++++++ ++++++++ ++++++++ ++++++++ ++++++++ ++++++++ ++++++++ ++++++++ ++++++++ ++++++++ ++++++++ ++++++++ ++++++++ ++++++++ ++++++++ ++++++++ ++++++++ ++++++++ ++++++++ ++++++++ ++++++++ ++++++++ ++++++++ ++++++++ ++++++++ ++++++++ ++++++++ ++++++++ ++++++++ ++++++++ ++++++++ ++++++++ ++++++++ ++++++++ ++++++++ ++++++++ ++++++++ ++++++++ ++++++++ ++++++++ ++++++++ ++++++++ ++++++++ ++++++++ ++++++++ ++++++++ ++++++++ ++++++++ ++++++++ ++++++++ ++++++++ ++++++++ ++++++++ ++++++++ ++++++++ ++++++++ ++++++++ ++++++++ ++++++++ ++++++++ ++++++++ ++++++++ ++++++++ ++++++++ ++++++++ ++++++++ ++++++++ ++++++++ ++++++++ ++++++++ ++++++++ ++++++++ ++++++++ ++++++++ ++++++++ ++++++++ ++++++++ ++++++++ ++++++++ ++++++++ ++++++++ ++++++++ ++++++++ ++++++++ ++++++++ ++++++++ ++++++++ ++++++++ ++++++++ ++++++++ ++++++++ ++++++++ ++++++++ ++++++++ ++++++++ ++++++++ ++++++++ ++++++++ ++++++++ ++++++++ ++++++++ ++++++++ ++++++++ ++++++++ ++++++++ ++++++++ ++++++++ ++++++++ ++++++++ ++++++++ ++++++++ ++++++++ ++++++++ ++++++++ ++++++++ ++++++++ ++++++++ ++++++++ ++++++++ ++++++++ ++++++++ ++++++++ ++++++++ ++++++++ ++++++++ ++++++++ ++++++++ ++++++++ ++++++++ ++++++++ ++++++++ ++++++++ ++++++++ ++++++++ ++++++++ ++++++++ ++++++++ ++++++++ ++++++++ ++++++++ ++++++++ ++++++++ ++++++++ ++++++++ ++++++++ ++++++++ ++++++++ ++++++++ ++++++++ ++++++++ ++++++++ ++++++++ ++++++++ ++++++++ ++++++++ ++++++++ ++++++++ ++++++++ ++++++++ ++++++++ ++++++++ ++++++++ ++++++++ ++++++++ ++++++++ ++++++++ ++++++++ ++++++++ ++++++++ ++++++++ ++++++++ ++++++++ ++++++++ ++++++++ ++++++++ ++++++++ ++++++++ ++++++++ ++++++++ ++++++++ ++++++++ ++++++++ ++++++++ ++++++++ ++++++++ ++++++++ ++++++++ ++++++++ ++++++++ ++++++++ ++++++++ ++++++++ ++++++++ ++++++++ ++++++++ +++++++
PROCEDURE SODETI

MAIN SOLVER MODULE FOR THE TAU METHOD SOLUTION OF A LINEAR
ODE, BY DIRECT SERIES SUBSTITUTION.

INPUT:

THE FOLLOWING INDETERMINATES MAY BE PRESENT IN THE INPUT:
X - THE VARIABLE IN THE POLYNOMIAL COEFFICIENTS;
Y - THE UNKNOWN SOLUTION FUNCTION;
DY - DY(I) REPRESENTS THE I-TH DERIVATIVE OF Y;
YX - YX(I) REPRESENTS THE VALUE OF Y AT A POINT X(I);
DYX - DYX(I,J) REPRESENTS THE VALUE OF DY(I) AT A POINT X(J);
MU - MU(I) REPRESENTS THE UNSPECIFIED INDETERMINATE APPEARING
IN THE PROBLEM;

THE INPUT DECK BEGINS WITH
NUM -- THE NUMBER OF PROBLEMS TO BE SOLVED
FOR EACH PROBLEM THE INPUT SEQUENCE IS AS FOLLOWS:
ORDER -- THE ORDER OF THE ODE;
NMAX -- THE DEGREE OF POLYNOMIAL APPROXIMATION DESIRED;
RANGE -- THE FINITE INTERVAL (A,B) IN WHICH THE SOLUTION IS DESIRED;
CODE -- THE SOLUTION CODE FOR THE PROBLEM;
DIFFEQ -- THE ODE AS A MULTINOMIAL IN THE INDETERMINATES X,Y;
DY(I)......DY(ORDER);
CONDN -- THE ASSOCIATED CONDITIONS CONDN(1),...,CONDN(ORDER) WRITTEN
AS MULTINOMIALS IN THE INDETERMINATES YX(I),DYX(I,J);
1 <= J <= MAXSUB, 1 <= I <= ORDER-1, WHERE MAXSUB IS
THE NUMBER OF X-VALUES APPEARING IN CONDN;
XSUB -- RATIONAL NUMBERS XSUB(1),...,XSUB(MAXSUB) REPRESENTING
THE X-VALUES ABOVE.

OUTPUT:

IN ADDITION TO THE INDETERMINATES ABOVE, THE FOLLOWING
INDETERMINATES MAY APPEAR IN THE OUTPUT FROM THE PACKAGE,
R -- THE INDETERMINATE APPEARING AS THE INDEPENDENT VARIABLE IN
THE RECURSION EQUATION;
AR -- AR(I) REPRESENTS THE TERM A(R+J) IN THE RECURSION EQUATION,
WHERE R REPRESENTS AN INDETERMINATE;
T -- T(I) REPRESENTS THE J-TH CHEBYSHEV POLYNOMIAL.

THE FIRST PART OF THE OUTPUT IS AN ECHO OF THE INPUT. IN ADDITION
TO THE APPROXIMATE SOLUTION, ITS CHEBYSHEV COEFFICIENTS, AND THE
PERTURBATION IMPOSED ON THE ODE, THE PACKAGE MAY PRINT OTHER
INFORMATION DEPENDING ON THE VALUE OF THE MESSAGE LEVEL
INDICATOR, MSGlvl, AS FOLLOWS:
MSGlvl
1 SUMMARY OF RESULTS OF COMPUTATION
2 MORE DETAILED INFORMATION
3 DETAILED INFORMATION PLUS TIMING STATISTICS
4 DETAILED DEBUGGING INFORMATION
5 PACKAGE PROCEDURES REQUIRED: POWERS,TPOWS,PROTSOL,TLT,EPIT
6 SYSTEM PROCEDURE REQUIRED: TIML
# DECLARATION IN CALLING PROCEDURE: NONE.

INTEGER ORDER,I,J,K,M,XSUB,NMAX,CODE,NUM
EXTERNAL INTEGER MAXORD,MAXDEG,MAXVAL,MAXMU,XDEG,YDEG
EXTERNAL INTEGER MAXPAR,MSGVL,JERR,INPFLG
RATIONAL ARRAY XSUB,RANGE
LOGICAL LINEAR
SET UP GLOBAL LAYOUT AS LONG ALGEBRAIC GLOBL

LONG ALGEBRAIC (DY, YDEG, XDEG, MU, XM, XMAX, MY, YMAX, XVAL, YVAL, r, xdeg, par, xpar, ypar, xmaxdeg, ymaxdeg) 1
EXTERNAL LONG ALGEBRAIC XX=X, XY=Y, RR=R
EXTERNAL LONG ALGEBRAIC ARRAY XT=1, XAR=AR, XPAR=PAR
EXTERNAL LONG ALGEBRAIC ARRAY XXR=XR, NYX=YY
EXTERNAL LONG ALGEBRAIC ARRAY XDYX=DYX, XMU=MU
EXTERNAL LONG ALGEBRAIC ARRAY XDY=DY, XPOWER, TPOWER
LONG ALGEBRAIC DIFFQ=GLOBL, YNEW, PERT, RHS
LONG ALGEBRAIC ARRAY CONDN=DIFFQ, POLY
REAL TOLD, TI, TNEW
LONG ALGEBRAIC ALTRAN SOLT!
LONG ALGEBRAIC ARRAY ALTRAN POWERS, TPOWS

XPOWER = POWERS(XDEG)
TPOWER = TPOWS(XDEG, XX)

READ NUM
DO I = 1, NUM
  PAGE() TIME(TOLD)
  T1 = TOLD
  PROTNMAX, ORDER, DIFFQ, CODE, POLY, RHS, MAXSUB,
    XSUB, CONDN, RANGE()
  TNEW = TIME(TOLD)
  IF(MSGVL>2) WRITE "TIME USED FOR PROLOGUE WAS", TNEW

# OBTAIN POLYNOMIAL SOLUTION, YNEW.

YNEW = SOLT1(ORDER, POLY, NMAX, RHS, CODE, PERT, MAXSUB, CONDN, XSUB)

TIME(TOLD)
EPIT(NMAX, YNEW, PERT, RANGE)
TNEW = TIME(TOLD)
IF(MSGVL>2) WRITE "TIME USED FOR EPILOGUE WAS", TNEW

SYSERR:  # CHECK ERROR STATE.
  IERR = ERRNO()
  IF(IERR<0) DO
    IF(INPFLG<>1) DUMP()
    ELSE DO; SNAP; OK(); IERR=0; DOEND
  DOEND
  TNEW = TIME(TI)
  IF(MSGVL>2) WRITE "TOTAL TIME ELAPSED WAS", TNEW
  WRITE "", "+++++++++++++++++++++++++++++++", "
  DOEND
RETURN
END  # END OF PROCEDURE SODET1.
PROCEDURE SOLT2

MAIN SOLVER MODULE FOR THE TAU METHOD SOLUTION OF A LINEAR
ODE VIA CANONICAL POLYNOMIALS.

INPUT:

THE FOLLOWING INDETERMINATES MAY BE PRESENT IN THE INPUT:
X - THE VARIABLE IN THE POLYNOMIAL COEFFICIENTS;
Y - THE UNKNOWN SOLUTION FUNCTION;
DY - DY(I) REPRESENTS THE ITH DERIVATIVE OF Y;
YX - YX(I) REPRESENTS THE VALUE OF Y AT A POINT X(I);
DYX - DYX(I) REPRESENTS THE VALUE OF DY(I) AT A POINT X(I);
MU - MU(I) REPRESENTS THE UNSPECIFIED INDETERMINATE APPEARING
IN THE PROBLEM;

THE INPUT DECK BEGINS WITH
NUM - THE NUMBER OF PROBLEMS TO BE SOLVED;
FOR EACH PROBLEM THE INPUT SEQUENCE IS AS FOLLOWS:
ORDER - THE ORDER OF THE ODE;
NMAX - THE DEGREE OF POLYNOMIAL APPROXIMATION DESIRED;
RANGE - THE FINITE INTERVAL (A,B) IN WHICH THE SOLUTION IS DESIRED;
CODE - THE SOLUTION CODE FOR THE PROBLEM;
DIFFLO - THE ODE AS A MULTINOMIAL IN THE INDETERMINATES X,Y,
DY(I),...,DY(ORDER);
CONDN - THE ASSOCIATED CONDITIONS COND(I),...,CONDN(ORDER) WRITTEN
AS MULTINOMIALS IN THE INDETERMINATES YX(I),DYX(I),
1 <= I <= MAXSUB, 1 <= 1 <= ORDER - 1, WHERE MAXSUB IS
THE NUMBER OF X-VALUES APPEARING IN CONDN;
XSUB - RATIONAL NUMBERS XSUB(I),...,XSUB(MAXSUB) REPRESENTING
THE X-VALUES ABOVE.

OUTPUT:

IN ADDITION TO THE INDETERMINATES ABOVE, THE FOLLOWING
INDETERMINATES MAY APPEAR IN THE OUTPUT FROM THE PACKAGE:
R - THE INDETERMINATE APPEARING AS THE INDEPENDENT VARIABLE IN
THE RECURSION EQUATION;
AR - AR(I) REPRESENTS THE TERM A(R+I) IN THE RECURSION EQUATION,
WHERE R REPRESENTS AN INDETERMINATE;
T - T(I) REPRESENTS THE JTH CHEBYSHEV POLYNOMIAL.

THE FIRST PART OF THE OUTPUT IS AN ECHO OF THE INPUT. IN ADDITION
TO THE APPROXIMATE SOLUTIONS CHEBYSHEV COEFFICIENTS, AND THE
PERTURBATION IMPOSED ON THE ODE, THE PACKAGE MAY PRINT OTHER
INFORMATION DEPENDING ON THE VALUE OF THE MESSAGE LEVEL.
INDICATOR, MSGLEVEL, AS FOLLOWS:
MSGLEVEL
1 - SUMMARY OF RESULTS OF COMPUTATION
2 - MORE DETAILED INFORMATION
3 - DETAILED INFORMATION PLUS TIMING STATISTICS
4 - DETAILED DEBUGGING INFORMATION.

PACKAGE PROCEDURES REQUIRED: POWERS,TPOWS,PRO1,SOLD2,EPIT.COM
SYSTEM PROCEDURE REQUIRED: TIME
DECLARATION IN CALLING PROCEDURE: NONE.

INTEGER ORDER, I,J,K, MAXSUB, NMAX, CODE, NUM
EXTERNAL INTEGER IERR, MAXORD, MAXDEG, MAXX, MAXMU, XDEG, YDEG
EXTERNAL INTEGER MAXPAR, MSGLEVEL, IERR, INPFLG
INTEGER MM=-MAXDEG-MAXPAR
RATIONAL ARRAY XSUB,RANGE
LOGICAL LINEAR
SET UP GLOBAL LAYOUT AS LONG ALGEBRAIC GLOBAL.

LONG ALGEBRAIC (DY@MAXORD);YDEG,Y,YDEG,X;XDEG;
MU(1@MAX(1,MAXMU));XDEG;DYX(1@MAXORD,1@MAXVAL);I,
YX(1@MAXVAL,1@MAXORD,1@QR(-MM:0);I,UQ:0:MM);I,
RX;XDEG;PAR(1@MAXPAR);1,XR(1@MAXDEQ;MAXDEG);1) GLOBL
EXTERNAL LONG ALGEBRAIC XX=X, XY=Y, RR=R
EXTERNAL LONG ALGEBRAIC ARRAY XT=T,XQR=QR,XPAR=PAR
EXTERNAL LONG ALGEBRAIC ARRAY XXR=XR,XUQ=UQ,XYX=YX
EXTERNAL LONG ALGEBRAIC ARRAY XDY=DY,XMU=MU
EXTERNAL LONG ALGEBRAIC ARRAY XDY=DY,XPOWER,TPOWER
LONG ALGEBRAIC DIFFEQ=GLOBL,YNEW,PERT,RHS
LONG ALGEBRAIC ARRAY CONDN=DIFFEQ,POLY
REAL TOLD,T,TNEW
LONG ALGEBRAIC ALTRAN SOLT2
LONG ALGEBRAIC ARRAY ALTRAN POWERS,TPOWS

XPOWER = POWERS(XDEG)
TPOWER = TPOWS(XDEG,XX)

READ NUM
DO I = 1, NUM
PAGE()
TIME(TOLD)
T = TOLD
PROT(NMAX,ORDER,DIFFEQ,CODE,POLY,RHS,MAXSUB,
XSUB,CONDN,RANGE)
TNEW = TIME(TOLD)
IF(MSLVL>2) WRITE "TIME USED FOR PROLOGUE WAS", TNEW

# OBTAIN POLYNOMIAL SOLUTION, YNEW.
YNEW = SOLT2(ORDER,POLY,NMAX,RHS,CODE,PERT,MAXSUB,CONDN,XSUB)

TIME(TOLD)
FPIT(NMAX,YNEW,PERT,RANGE)
TNEW = TIME(TOLD)
IF(MSLVL>2) WRITE "TIME USED FOR EPILOGUE WAS", TNEW

SYSERR. CHECK ERROR STATE.
IERR = HERRNO4()
IF (IERR<>0) DO
IF (NPFLG<>1) DUMP()
ELSE DO; SNAP, OK(); IERR=0; DOEND
DOEND
TNEW = TIME(T1)
IF(MSLVL>2) WRITE "TOTAL TIME ELAPSED WAS", TNEW
WRITE "", " +++++++++++++++++++++++++++++++++++++++++++++++++++", " "
DOEND
RETURN
END # END OF PROCEDURE SODET2.
MAIN SOLVER MODULE FOR THE TAU METHOD SOLUTION OF A LINEAR
OR NONLINEAR ODE, BY DIRECT SERIES SUBSTITUTION
A FORM OF NEWTON ITERATION IS USED TO REDUCE THE
TO THAT OF SOLVING SUCCESSIVE LINEAR ODES.

INPUT:
THE FOLLOWING INDETERMINATES MAY BE PRESENT IN THE INPUT:
X ← THE VARIABLE IN THE POLYNOMIAL COEFFICIENTS;
Y ← THE UNKNOWN SOLUTION FUNCTION;
DY ← DY(I) REPRESENTS THE I–TH DERIVATIVE OF Y;
XY ← XY(I) REPRESENTS THE VALUE OF Y AT A POINT X(I);
DXY ← DXY(I,J) REPRESENTS THE VALUE OF DY(I) AT A POINT X(J);
MU ← MU(J) REPRESENTS THE UNSPECIFIED INDETERMINATE APPEARING
IN THE PROBLEM.

THE INPUT DECK BEGINS WITH
NUM ← THE NUMBER OF PROBLEMS TO BE SOLVED.
FOR EACH PROBLEM THE INPUT SEQUENCE IS AS FOLLOWS:
ORDER ← THE ORDER OF THE ODE;
NMAX ← THE DEGREE OF POLYNOMIAL APPROXIMATION DESIRED;
RANGE ← THE FINITE INTERVAL (A,B) IN WHICH THE SOLUTION IS DESIRED;
CODE ← THE SOLUTION CODE FOR THE PROBLEM;
DIFQ← THE ODE AS A MULTINOMIAL IN THE INDETERMINATES X,Y,
DY(I),...,DY(ORDER);
CONDN ← THE ASSOCIATED CONDITIONS CONDNI(1),...,CONDNI(ORDER) WRITTEN
AS MULTINOMIALS IN THE INDETERMINATES YX(I),DYX(I,J),
1 <= J <= MAXSUB, 1 <= I <= ORDER–1, WHERE MAXSUB IS
THE NUMBER OF X–VALUES APPEARING IN CONDN;
XSUB ← RATIONAL NUMBERS XSUB(I),...,XSUB(MAXSUB) REPRESENTING
THE X–VALUES ABOVE.

OUTPUT:
IN ADDITION TO THE INDETERMINATES ABOVE, THE FOLLOWING
INDETERMINATES MAY APPEAR IN THE OUTPUT FROM THE PACKAGE:
R ← THE INDETERMINATE APPEARING AS THE INDEPENDENT VARIABLE IN
THE SUCCESSIVE RECURRENCE EQUATIONS;
AR ← AR(I) REPRESENTS THE TERM AR(I+J) IN THE RECURRENCE EQUATION,
WHERE R REPRESENTS AN INDETERMINATE;
T ← T(I) REPRESENTS THE J–TH CHEBYSHEV POLYNOMIAL.
THE FIRST PART OF THE OUTPUT IS AN ECHO OF THE INPUT. IN ADDITION
TO THE APPROXIMATE SOLUTION, ITS CHEBYSHEV COEFFICIENTS, AND THE
PERTURBATION IMPOSED ON THE ODE, THE PACKAGE MAY PRINT OTHER
INFORMATION DEPENDING ON THE VALUE OF THE MESSAGE LEVEL
INDICATOR, MSGLV. AS FOLLOWS:
MSGLV.
1 SUMMARY OF RESULTS OF COMPUTATION
2 MORE DETAILED INFORMATION
3 DETAILED INFORMATION PLUS TIMING STATISTICS
4 DETAILED DEBUGGING INFORMATION.
WHEN MSGLV > 1, THE RELEVANT INFORMATION IS PRINTED
AT EACH ITERATION.
PACKAGE PROCEDURES REQUIRED: POWERS, TPOWS, PRON, SOLNI, EPIN
SYSTEM PROCEDURE REQUIRED: TIME
DECLARATION IN CALLING PROCEDURE: NONE

INTEGER ORDER, I, J, K, MAXSUB, NMAX, CODE, NUM, N, M, NIER, CURN, NUMIT
EXTERNAL INTEGER MAXORD, MAXDEG, MAXVAL, MAXMU, XDEG, YDEG
EXTERNAL INTEGER MAXPAR, MSGLV, IFRR, INPLG
RATIONAL ARRAY XSUB, RANGE
LOGICAL LINEAR
SET UP GLOBAL LAYOUT AS LONG ALGEBRAIC GLOBL

LONG ALGEBRAIC (DY@:MAXORD), YDEG, Y:VDEG, X:XDEG,
MU(1:MAXM), XDEG, DY(1:MAXORD), Y:MAXVAL), 1,
YX(1:MAXVAL), 1, 10, XDEG), 1, 1, AR(-MAXDEG:MAXDEG), 1,
R,XDEG, PAR(0:MAXPAR), 1, XR(-MAXDEG MAXDEG), 1) GLOBL
EXTERNAL LONG ALGEBRAIC XX=X, XY=Y, RR=R
EXTERNAL LONG ALGEBRAIC ARRAY XT=T, XAR=AR, XPAR=PAR
EXTERNAL LONG ALGEBRAIC ARRAY XXR=XR, XYX=YX
EXTERNAL LONG ALGEBRAIC ARRAY XDYX=DXY, XMU=MU
EXTERNAL LONG ALGEBRAIC ARRAY XDY=Y, XPOW, TPOWER
LONG ALGEBRAIC DIFFEQ=GLOBL, YNEW, PERT, RHS, YINIT
LONG ALGEBRAIC ARRAY COND=DIFFQ, POLY
REAL TOLD, TI, TNEW
LONG ALGEBRAIC ALTRAN SOLN
LONG ALGEBRAIC ARRAY ALTRAN POWER, TPOWS

XPOWER = POWERS(XDEG)
TPOWER = TPOWS(XDEG, XX)

READ NUM
DO 1 = 1, NUM
   PAGE(1)
   TIME(TOLD)
   TI = TOLD
   PRON(CURN, NMAX, NUMIT, ORDER, DIFFEQ, POLY, MAXSUB,
   XSUB, COND, CODE, RHS, YINIT, RANGE)
   TNEW = TIME(TOLD)
   IF(MSGLVL>2) WRITE " TIME USED FOR PROLOGUE WAS", TNEW

# OBTAIN POLYNOMIAL SOLUTION, YNEW.
   YNEW = SOLN(ORDER, POLY, DIFFEQ, RHS, CURN, NMAX, NUMIT,
   NITER, CODE, PERT, MAXSUB, COND, XSUB, YINIT)

TIME(TOLD)
EPIN(NMAX, NITER, YNEW, PERT, RANGE)
TNEW = TIME(TOLD)
IF(MSGLVL>2) WRITE " TIME USED FOR EPILOGUE WAS", TNEW

SYSERR:  # CHECK ERROR STATE.
   IERR = ERRNO()
   IF (IERR<>0) DO
      IF(INPFLG<>0) DUMP()
      ELSE DO, SNAP, OK(); IERR=0; DOEND
   DOEND
   TNEW = TIME(TI)
   IF(MSGLVL>2) WRITE " TOTAL TIME ELAPSED WAS", TNEW
   WRITE "", " +++++++++++++++++++++++++++++++", " "
   DOEND
RETURN
END  # END OF PROCEDURE SODEN1.
<table>
<thead>
<tr>
<th>List of Level 2 Procedures</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>PROC</td>
<td>22</td>
</tr>
<tr>
<td>PROT</td>
<td>23</td>
</tr>
<tr>
<td>PRON</td>
<td>24</td>
</tr>
<tr>
<td>SOLC1</td>
<td>26</td>
</tr>
<tr>
<td>SOLT1</td>
<td>27</td>
</tr>
<tr>
<td>SOLT2</td>
<td>28</td>
</tr>
<tr>
<td>SOLN1</td>
<td>29</td>
</tr>
<tr>
<td>EPIC</td>
<td>31</td>
</tr>
<tr>
<td>EPIT</td>
<td>32</td>
</tr>
<tr>
<td>EPIN</td>
<td>33</td>
</tr>
</tbody>
</table>
List of Level 2 Procedures

<table>
<thead>
<tr>
<th>Procedure</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>PROC</td>
<td>B-21</td>
</tr>
<tr>
<td>PROT</td>
<td>B-22</td>
</tr>
<tr>
<td>PROLCL</td>
<td>B-23</td>
</tr>
<tr>
<td>PRON</td>
<td>B-24</td>
</tr>
<tr>
<td>SOLC1</td>
<td>B-26</td>
</tr>
<tr>
<td>SOLC2</td>
<td>B-28</td>
</tr>
<tr>
<td>SOLT1</td>
<td>B-29</td>
</tr>
<tr>
<td>SOLT2</td>
<td>B-30</td>
</tr>
<tr>
<td>SOLN1</td>
<td>B-31</td>
</tr>
<tr>
<td>EPIC</td>
<td>B-33</td>
</tr>
<tr>
<td>EPIT</td>
<td>B-34</td>
</tr>
<tr>
<td>EPIN</td>
<td>B-35</td>
</tr>
</tbody>
</table>

*Not a Level 2 Procedure, but included here because it performs most of the work required in PROC and PROT.*
PROCEDURE PROC(N,ORDER,DIFEQ,CODE,POLY,
   RHS,MAXSUB,XSUB,CONDN,RANGE),
INTEGER N,ORDER,MAXSUB,CONDN,RANGE,
LOGICAL LINEAR
RATIONAL ARRAY XSUB
RATIONAL ARRAY RANGE
LONG ALGEBRAIC DIFEQ,RHS
LONG ALGEBRAIC ARRAY POLY,CONDN

PROLOGUE ROUTINE FOR CHEBYSHEV SERIES METHODS.
THIS PROCEDURE CALLS PROCEDURE PROGL, TO READ IN THE INPUT
FOR A LINEAR ODE AND ITS ASSOCIATED CONDITIONS.
PACKAGE PROCEDURE REQUIRED: PROGL
DECLARATION IN CALLING PROCEDURE: NONE

PROGL(N,ORDER,DIFEQ,CODE,POLY,RHS,
   MAXSUB,XSUB,CONDN,RANGE)

RETURN
END # END OF PROCEDURE PROC.
PROCEDURE PROT(N, ORDER, DIFEQ, CODE, POLY, RHS, MAXSUB, XSUB, CONDN, RANGE)
INTEGER N, ORDER, MAXSUB, CODE
LOGICAL LINEAR
RATIONAL ARRAY XSUB
RATIONAL ARRAY RANGE
LONG ALGEBRAIC DIFEQ, RHS
LONG ALGEBRAIC ARRAY POLY, CONDN

PROLOGUE ROUTINE FOR TAU METHODS.
THIS PROCEDURE CALLS PROCEDURE PROGLI TO READ IN THE INPUT
FOR A LINEAR ODE AND ITS ASSOCIATED CONDITIONS.
PACKAGE PROCEDURE REQUIRED: PROGLI
DECLARATION IN CALLING PROCEDURE: NONE

PROGLI(N, ORDER, DIFEQ, CODE, POLY, RHS,
MAXSUB, XSUB, CONDN, RANGE)
RETURN
END 4 END OF PROCEDURE PROT.
PROCEDURE PRON(N,NMAX,NUMIT,ORDER,DIFFEQ,POLY,MAXSUB,
   XSUB,CONDN,CODE;RHS,YINIT,RANGE)
   INTEGER N,NMAX,NUMIT,ORDER,MAXSUB,CODE
   RATIONAL ARRAY XSUB,RANGE
   LONG ALGEBRAIC DIFFEQ,RHS,YINIT
   LONG ALGEBRAIC ARRAY POLY,CONDN

THIS PROCEDURE READS IN AN ODE AND ITS ASSOCIATED CONDITIONS AND
CHECKS THE VALIDITY OF THE ORDER, THE DEGREE OF APPROXIMATION
REQUIRED, THE SOLUTION CODE, AND THE RANGE IN WHICH THE SOLUTION
IS REQUIRED.

THE ACTUAL INPUT AND PARAMETER CHECKING ARE PERFORMED BY
A CALL TO PROCEDURE READER.

INPUT PARAMETERS: NONE.

OUTPUT PARAMETERS:
   NMAX - DEGREE OF APPROXIMATION REQUIRED;
   NUMIT - NUMBER OF ITERATIONS REQUIRED;
   N    - DEGREE OF APPROXIMATION IN FIRST ITERATION;
   ORDER - ORDER OF ODE;
   MAXSUB - NUMBER OF X-VALUES IN ASSOCIATED CONDITIONS;
   CODE - SOLUTION CODE;
   XSUB  - AN ARRAY XSUB(1),...,XSUB(MAXSUB) OF X-VALUES APPEARING
           IN THE ASSOCIATED CONDITIONS;
   RANGE - A RATIONAL ARRAY (A,B) INDICATING THE FINITE INTERVAL
           IN WHICH THE SOLUTION IS DESIRED;
   DIFFEQ - THE ODE EXPRESSED AS DIFFEQ=0;
   POLY  - ARRAY POLY(0),...,POLY(ORDER) OF POLYNOMIAL COEFFICIENTS
           IN ODE;
   RHS   - RIGHT-HAND-SIDE POLYNOMIAL OF ODE;
   CONDN - ARRAY CONDN(1),...,CONDN(ORDER) OF ASSOCIATED CONDITIONS;
   YINIT - INITIAL APPROXIMATION SATISFYING ASSOCIATED CONDITIONS.

EXTERNAL VARIABLES REQUIRED: MSGVLX.

PACKAGE PROCEDURES REQUIRED: READER, SETP1,STEP2,INIT.

DECLARATION IN CALLING PROCEDURE: NONE.

INTEGER K
LOGICAL LINEAR
EXTERNAL INTEGER MSGVLX
EXTERNAL LONG ALGEBRAIC XX
LONG ALGEBRAIC ALTRAN INIT
LONG ALGEBRAIC ARRAY ALTRAN SETP1,SETP2
READ IN AND CHECK ODE AND ASSOCIATED CONDITIONS.

READ(RNMAX,ORDER,DIFEQ,CODE,MAXSUB,NSUB,CONDN,RANGE,LINEAR)

DISTINGUISH BETWEEN LINEAR AND NONLINEAR ODE.

IF (LINEAR) DO  # LINEAR ODE.
  N = 0
  WRITE * CODE IS LINEAR; ONLY ITERATION NEEDED
  CODE = 1
  WRITE * CODE SET TO 1
  N = NMAX
  YINIT = 0
  POLY = SETP(ORDER,DIFEQ,RHS)
  END
ELSE DO  # NONLINEAR ODE.
  N = 1
  K = 1
  AGAIN: N = N + 1, K = K + 1
  IF (K > NMAX) GO TO AGAIN
  N = 1
  YINIT = INT(ORDER,NMAX,MAXSUB,NSUB,CONDN,XX)
  POLY = SETP(ORDER,N,DIFEQ,YINIT,RHS)
  END
IF (MSGLV. > 0) DO
  WRITE * THE INITIAL APPROXIMATION IS*, YINIT
  WRITE * THE POLYNOMIAL COEFFICIENTS OF THE ODE ARE*, POLY
  WRITE * THE RIGHT HAND SIDE POLYNOMIAL IS*, RHS
  END

RETURN
END # END OF PROCEDURE PRON.
PROCEDURE SOLCI(ORDER,POLY,N,RHS,MASUB,CONDN,XSUB,COEF,CODE,GEF)
  INTEGER VALUE N,ORDER,MASUB,COEF
  RATIONAL ARRAY VALUE XSUB
  LONG ALGEBRAIC VALUE RHS
  LONG ALGEBRAIC ARRAY COEF
  LONG ALGEBRAIC ARRAY VALUE POLY,CONDN

# THIS PROCEDURE COMPUTES A (FINITE) CHEBYSHEV SERIES SOLUTION
# OF A LINEAR ODE, BY DIRECT SERIES SUBSTITUTION.
# INPUT PARAMETERS:
#  N — DEGREE OF APPROXIMATION DESIRED;
#  ORDER — ORDER OF ODE;
#  CODE — THE METHOD OF SOLUTION DESIRED IS INDICATED
#  BY THE VALUE OF CODE AS FOLLOWS:
#  CODE = 1 — SOLUTION BY BACKWARD RECURRENCE
#  CODE = 2 — SOLUTION BY GAUSSIAN ELIMINATION;
#  DIFFEQ,POLY,RHS — INFORMATION ABOUT THE ODE;
#  MASUB,XSUB,CONDN — INFORMATION ABOUT THE ASSOCIATED
#  CONDITIONS.
# OUTPUT PARAMETERS:
#  COEF — THE REQUIRED CHEBYSHEV COEFFICIENTS;
#  YSTAR — THE POLYNOMIAL APPROXIMATION (RETURNED);
# EXTERNAL VARIABLES REQUIRED: MSGVLI,XX.
# PACKAGE PROCEDURES REQUIRED: GSCI1,GSCI2,RECCI,CHFORM.
# SYSTEM PROCEDURES REQUIRED: TIME, DEG,PINTN.
# DECLARATION IN CALLING PROCEDURE: LONG ALGEBRAIC ALTRAN SOLCI.

INTEGER HALFN,DEGRHS
REAL TOLD,TNEW
LONG ALGEBRAIC EQN,YSTAR
EXTERNAL INTEGER MSGVLI
EXTERNAL LONG ALGEBRAIC XX
LONG ALGEBRAIC ALTRAN RECCI,CHFORM,PINTN
LONG ALGEBRAIC ALTRAN GSCI1,GSCI2

# OBTAIN RECURRENCE EQUATION.

TIME(TOLD)
EQN = RECCI(ORDER,POLY,HALFN)
IF(MSGVLI>1) WRITE " THE RECURRENCE EQUATION IS" , EQN
TNEW = TIME(TOLD)
IF(MSGVLI>2)
  WRITE "TIME TO GENERATE RECURRENCE EQUATION WAS", TNEW

# OBTAIN THE CHEBYSHEV COEFFICIENTS AND POLYNOMIAL APPROXIMATION.

IF (CODE==1) DO
  YSTAR = GSCI1(N,ORDER,EQN,HALFN,MASUB,CONDN, XSUB,RHS,COEF)
END
ELSE IF (CODE==2) DO
  YSTAR = GSCI2(N,ORDER,EQN,HALFN,MASUB,CONDN, XSUB,RHS,COEF)
END

RETURN(YSTAR)
END # END OF PROCEDURE SOLCI.
PROCEDURE SOLTH(ORDER,POLY,N,RHS,CODE,PERT,MAXSUB,CONDN,XSUB)
    INTEGER VALUE ORDER,N,MAXSUB,CODE
RATIONAL ARRAY VALUE XSUB
LONG ALGEBRAIC VALUE RHS
LONG ALGEBRAIC ARRAY VALUE CONDN
LONG ALGEBRAIC PERT
LONG ALGEBRAIC ARRAY POLY

THIS PROCEDURE COMPUTES THE EXACT POLYNOMIAL SOLUTION OF
A PERTURBED LINEAR ODE, BY DIRECT SERIES SUBSTITUTION.
INPUT PARAMETERS:
N - DEGREE OF APPROXIMATION DESIRED;
ORDER - ORDER OF ODE;
POLY - POLYNOMIAL COEFFICIENTS OF ODE;
RHS - RIGHT-HAND-SIDE POLYNOMIAL OF ODE;
MAXSUB,XSUB,CONDN - INFORMATION ABOUT ASSOCIATED CONDITIONS;
CODE - THE FORM IN WHICH THE ODE IS TO BE SOLVED
IS INDICATED BY THE VALUE OF CODE AS FOLLOWS:
CODE = 1 INTEGRATED FORM WITH ORIGINAL CONDITIONS
CODE = 2 INTEGRATED FORM WITH PERTURBED CONDITIONS
CODE = 3 DIFFERENTIATED FORM WITH ORIGINAL CONDITIONS
CODE = 4 DIFFERENTIATED FORM WITH PERTURBED CONDITIONS.
OUTPUT PARAMETERS:
YSTAR - THE REQUIRED POLYNOMIAL APPROXIMATION(RETURNED);
PERT - PERTURBATION ON ODE.
EXTERNAL VARIABLES REQUIRED: MSGVL.
PACKAGE PROCEDURES REQUIRED: GST12, RECT12, RECT11.
SYSTEM PROCEDURES REQUIRED: TIME.
DECLARATION IN CALLING PROCEDURE: LONG ALGEBRAIC ALTRAN SOLT1.

INTEGER LH1,LH2
REAL TOLD,FNEW
LONG ALGEBRAIC YSTAR,EQN
EXTERNAL INTEGER MSGVL
LONG ALGEBRAIC ALTRAN GST12,RECT12,RECT11

OBTAIN RECURRANCE EQUATION.

TIME(TOLD)
EQN = RECT12(POLY,ORDER,H1,H2,CODE)
TNEW = TIME(TOLD)
IF(MSGVL>1) WRITE " THE RECURRANCE EQUATION IS", EQN
IF (MSGVL>2) WRITE "TIME TO GENERATE RECURRANCE EQUATION WAS", TNEW

OBTAIN POLYNOMIAL SOLUTION AND PERTURBATION.

YSTAR = GST12(EQN,N,H1,H2,CODE,PERT,MAXSUB,CONDN,XSUB,RHS,ORDER)

RETURN (YSTAR)
END # END OF PROCEDURE SOLTH.
PROCEDURE SOLT2(N, ORDER, POLY, CODE, PERT, MAXSUB, CONDN, XSUB,
   INTEGER N, ORDER, MAXSUB, CODE
   RATIONAL ARRAY POLY, CODE
   LONG ALGEBRAIC VALUE CODE
   LONG ALGEBRAIC ARRAY POLY
   INTEGER ORDER, N, MAXSUB, CODE
   RATIONAL ARRAY POLY, CODE
   LONG ALGEBRAIC VALUE CODE
   LONG ALGEBRAIC ARRAY CODE
   LONG ALGEBRAIC ARRAY POLY

// THIS PROCEDURE COMPUTES THE EXACT POLYNOMIAL SOLUTION OF
// A PERURBED LINEAR ODE, USING THE CANONICAL POLYNOMIALS
// ASSOCIATED WITH THE LINEAR OPERATOR DEFINED BY THE ODE.
// INPUT PARAMETERS:
// N        — DEGREE OF APPROXIMATION DESIRED;
// ORDER    — ORDER OF ODE;
// POLY     — POLYNOMIAL COEFFICIENTS OF ODE;
// RHS      — RIGHT-HAND-SIDE POLYNOMIAL OF ODE;
// MAXSUB, XSUB, CONDN — INFORMATION ABOUT ASSOCIATED CONDITIONS;
// CODE     — THE FORM IN WHICH THE ODE IS TO BE SOLVED
// IS INDICATED BY THE VALUE OF CODE AS FOLLOWS:
// CODE = 1 INTEGRATED FORM WITH ORIGINAL CONDITIONS
// CODE = 2 INTEGRATED FORM WITH PERURBED CONDITIONS
// CODE = 3 DIFFERENTIATED FORM WITH ORIGINAL CONDITIONS
// CODE = 4 DIFFERENTIATED FORM WITH PERURBED CONDITIONS
// OUTPUT PARAMETERS:
// YSTAR    — THE REQUIRED POLYNOMIAL APPROXIMATION(RETURNED);
// PERT     — PERTURBATION ON ODE.
// EXTERNAL VARIABLES REQUIRED: MSGGLVL.
// PACKAGE PROCEDURES REQUIRED: GST21, RECT21.
// SYSTEM PROCEDURES REQUIRED: TIME.
// DECLARATION IN CALLING PROCEDURE: LONG ALGEBRAIC ALTRAN SOLT2.

INTEGER I, H1, H2
REAL TOLD, TNEW
LONG ALGEBRAIC YSTAR, EQN, GENPOL
EXTERNAL INTEGER MSGGLVL
LONG ALGEBRAIC ALTRAN GST21, RECT21

// OBTAIN THE RECURRENCE EQUATION DEFINING THE CANONICAL POLYNOMIALS.
TIME(TOLD)
EQN = RECT21(POLY, ORDER, H1, H2, CODE, GENPOL)
IF (MSGGLVL > 1) WRITE "THE RECURRENCE EQUATION IS", EQN
TNEW = TIME(TOLD)
IF (MSGGLVL > 2)
   WRITE "TIME TO GENERATE RECURRENCE EQUATION WAS", TNEW

// OBTAIN POLYNOMIAL SOLUTION AND PERTURBATION.
YSTAR = GST21(N, ORDER, POLY, CODE, H1, H2, EQN, GENPOL,
   RHS, MAXSUB, CONDN, XSUB, PERT)

RETURN(YSTAR)
END # END OF PROCEDURE SOLT2.
PROCEDURE SOLNORDER,POLY,DIFSEQ,RHS,CURN,NMAX,NUMIT,
NITER,CODE,PERT,MAXSUB,CONDN,XSUB,YINIT
INTEGER NITER,CURN
INTEGER VALUE ORDER,NMAX,NUMIT,MAXSUB,.Code
RATIONAL ARRAY VALUE XSUB
LONG ALGEBRAIC VALUE DIFSEQ,RHS,YINIT
LONG ALGEBRAIC ARRAY VALUE POLY,CONDN
LONG ALGEBRAIC PERT

/* THIS PROCEDURE COMPUTES A POLYNOMIAL APPROXIMATION TO
A GIVEN LINEAR OR NONLINEAR ODE BY A FORM OF NEWTON
ITERATION. */

/* INPUT PARAMETERS: */
ORDER — ORDER OF THE ODE;
NMAX — DEGREE OF APPROXIMATION REQUIRED;
CURN — THE DEGREE OF APPROX IN FIRST ITERATION;
NUMIT — NUMBER OF ITERATIONS REQUIRED;
CODE — SOLUTION CODE INDICATING MODIFICATIONS
TO BE PERFORMED ON SUCCESSIVE ITERATES AS
FOLLOWS:
CODE = 1 NO ROUNDDING OR SHORTENING
CODE = 2 ROUNDDING ONLY
CODE = 3 SHORTENING ONLY
CODE = 4 ROUNDDING AND SHORTENING;
DIFSEQ — THE ODE REPRESENTED AS DIFSEQ=0;
POLY — POLYNOMIAL COEFFICIENTS OF INITIAL LINEAR ODE;
RHS — RIGHT-HAND-SIDE POLYNOMIAL OF INITIAL LINEAR ODE;
CONDN,XSUB,MAXSUB — CONDITIONS ASSOCIATED WITH ODE;
YINIT — THE INITIAL APPROXIMATION TO THE SOLUTION FUNCTION.

/* OUTPUT PARAMETERS: */
CURN — DEGREE OF LAST APPROXIMATION OBTAINED. IF THE
PROCEDURE TERMINATES NORMALLY, CURN = NMAX;
NITER — NUMBER OF ITERATIONS PERFORMED. IF THE PROCEDURE
TERMINATES NORMALLY, NITER = NUMIT;
YNEW — THE LAST POLYNOMIAL APPROXIMATION COMPUTED;
PERT — PERTURBATION. IF THE PROCEDURE TERMINATES NORMALLY,
PERT IS THE PERTURBATION ON THE ORIGINAL ODE;
ELSE PERT IS THE PERTURBATION ON THE LAST LINEAR
SOLVED.
EXTERNAL VARIABLES REQUIRED: MAXMU,MSGVL,XMU.
PACKAGE PROCEDURES REQUIRED: SOLT, NEXTT.
SYSTEM PROCEDURES REQUIRED: ANUM, CTPO.
DECLARATION IN CALLING PROCEDURE: LONG ALGEBRAIC ALTRAN SOLN.

INTEGER CURIT,SC=1
LONG ALGEBRAIC YNEW=YINIT,NUM,DEN,YCORR
LONG INTEGER ALTRAN CTPO
LONG ALGEBRAIC ALTRAN SOLT
EXTERNAL INTEGER MAXMU,MSGVL
EXTERNAL LONG ALGEBRAIC ARRAY XMU
WRITE "CONDNS FOR THE CORRECTION TERM ARE:"
DO I = 1, ORDER
   CONDNI = CONDNI1(XMUL = 1, MAXUL, MAXVUL, 0)
   NUM = ANUMI(CONDNI, 0, DEN)
   CONDNG = (NUM - CTPO(1, EXPAND(NUM))/DEN)
   WRITE CONDNI1
ENDDO

DO CURIT = 1, NNUM
   NITER = CURIT
   IF (MSG > 1) DO
      PAGE()
      WRITE " ", " 
      WRITE " ", " 
      WRITE " ", " 
      WRITE " ", " 
      WRITE " ", " 
      WRITE " ", " 
      WRITE " ", " 
      WRITE " ", " 
      WRITE " ", " 
   ENDDO

# OBTAIN CORRECTION TERM.
YCORR = SOLU(ORDER, POLY, CURN, RHE, SC, PERT, MAXSUB, CONDNI, XSUB)

# PREPARE FOR THE NEXT ITERATION.
NEXTIT(CURN, NMAX, ORDER, YNEW, YCORR, POLY, PERT, CODE, DIP, PEQ, RHS)
ENDDO

RETURN(YNEW)
END # END OF PROCEDURE SOLNI.
PROCEDURE EPIC(NMAX,YNEW, RANGE)

INTEGER VALUE NMAX
RATIONAL ARRAY VALUE RANGE
LONG ALGEBRAIC VALUE: YNEW
LONG ALGEBRAIC ARRAY VALUE C

# THIS PROCEDURE OUTPUTS THE RESULTS OF COBYSHEV SERIES
# SOLUTION OF A LINEAR ODE.
# INPUT PARAMETERS:
# NMAX — DEGREE OF APPROXIMATION;
# YNEW — THE POLYNOMIAL APPROXIMATION;
# C — ARRAY C[0],...,C[NMAX] OF REQUIRED COBYSHEV
# COEFFICIENTS;
# RANGE — THE FINITE INTERVAL (A,B) IN WHICH THE SOLUTION IS
# REQUIRED.
# OUTPUT PARAMETERS: NONE
# EXTERNAL VARIABLES REQUIRED: XX.
# PACKAGE PROCEDURES REQUIRED: CWRITE,TRANS2.
# DECLARATION IN CALLING PROCEDURE: NONE.

INTEGER DEGREE
LONG ALGEBRAIC YTRANS
EXTERNAL LONG ALGEBRAIC XX

# OUTPUT RESULTS OF COMPUTATION.
WRITE " THE DEGREE OF APPROXIMATION IS", NMAX
WRITE " THE POLYNOMIAL SOLUTION IS", YNEW
CWRITE(C,NMAX)

# TRANSFORM RESULTS BACK TO RANGE (A,B), IF NECESSARY.
IF (XX.(RANGE(I)=(-1,1))) THEN
    WRITE " SOLUTION TRANSFORMED BACK TO GIVEN RANGE (A,B)"
    TRANS2(RANGE,YNEW,YTRANS)
    WRITE " THE TRANSFORMED POLYNOMIAL SOLUTION IS", YTRANS
END IF

RETURN
END # END OF PROCEDURE EPIC.
PROCEDURE EPIT(NMAX,YNEW,PERT,RANGE)
INTEGER VALUE NMAX
RATIONAL ARRAY VALUE RANGE
LONG ALGEBRAIC VALUE PERT,YNEW

# THIS PROCEDURE OUTPUTS THE RESULTS OF SOLUTION BY THE
# TAU METHOD OF A LINEAR ODE.
# INPUT PARAMETERS:
#    NMAX    --  DEGREE OF APPROXIMATION.
#    YNEW    --  THE POLYNOMIAL APPROXIMATION.
#    PERT    --  THE PERTURBATION IMPOSED ON THE ORIGINAL ODE.
#    RANGE   --  THE FINITE INTERVAL (A,B) IN WHICH THE SOLUTION IS
#                REQUIRED.
# OUTPUT PARAMETERS:  NONE.
# EXTERNAL VARIABLES REQUIRED: XX.
# PACKAGE PROCEDURES REQUIRED: TCS,CHFORM,CWRIT,TRANS2.
# DECLARATION IN CALLING PROCEDURE: NONE.

INTEGER DEGREE,
LONG ALGEBRAIC YTRANS,PTRANS,PI
LONG ALGEBRAIC ARRAY C
EXTERNAL LONG ALGEBRAIC XX
LONG ALGEBRAIC ALTRAN CHFORM
LONG ALGEBRAIC ARRAY ALTRAN TCS

# OUTPUT RESULTS OF COMPUTATION.

PI = PERT
C = TCS(CHFORM(YNEW,NMAX,XX),NMAX)
DEGREE = DEG(PERT,XX)
PERT = CHFORM(PERT,DEGREE,XX)
WRITE ' THE DEGREE OF APPROXIMATION IS', NMAX
WRITE ' THE POLYNOMIAL SOLUTION IS', YNEW
WRITE ' THE PERTURBATION IS', PERT
CWRIT(C,NMAX)

# TRANSFORM RESULTS BACK TO RANGE (A,B), IF NECESSARY.

IF (.NOT.(RANGE == (-1,1))) DO
WRITE ' SOLUTION TRANSFORMED BACK TO GIVEN RANGE (A,B)
TRANS2(RANGE,YNEW,YTRANS)
WRITE ' THE TRANSFORMED POLYNOMIAL SOLUTION IS', YTRANS
TRANS2(RANGE,PI,PTRANS)
WRITE ' THE TRANSFORMED PERTURBATION IS', PTRANS
END

RETURN
END    # END OF PROCEDURE EPIT.
PROCEDURE EPIN(NMAX,NITER,YNEW,PERT,RANGE)
  INTEGER VALUE NMAX,NITER
  RATIONAL ARRAY VALUE RANGE
  LONG ALGEBRAIC VALUE PERT,YNEW

# THIS PROCEDURE OUTPUTS THE RESULTS OF ITERATIVE SOLUTION BY THE
# TAU METHOD OF A LINEAR OR NONLINEAR ODE.
# INPUT PARAMETERS:
#  NMAX  -- DEGREE OF APPROXIMATION:
#  NITER -- NUMBER OF ITERATIONS PERFORMED:
#  YNEW  -- THE POLYNOMIAL APPROXIMATION;
#  PERT  -- THE PERTURBATION IMPOSED ON THE ORIGINAL ODE:
#  RANGE -- THE FINITE INTERVAL (A,B) IN WHICH THE SOLUTION IS
#           REQUIRED.
# OUTPUT PARAMETERS: NONE.
# EXTERNAL VARIABLES REQUIRED: XX.
# PACKAGE PROCEDURES REQUIRED: TCS,CHFORM,CWRIT,TRANS2.
# DECLARATION IN CALLING PROCEDURE: NONE.

INTEGER DEGREE
LONG ALGEBRAIC YTRANS,PTRANS,P1
LONG ALGEBRAIC ARRAY C
EXTERNAL LONG ALGEBRAIC XX
LONG ALGEBRAIC ALTRAN CHFORM
LONG ALGEBRAIC ARRAY ALTRAN TCS

# OUTPUT RESULTS OF COMPUTATION.

PAGE()
WRITE " %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
WRITE " THE DEGREE OF APPROXIMATION IS", NMAX
WRITE " THE NUMBER OF ITERATIONS PERFORMED WAS", NITER

P1 = PERT
C = TCSCHFORM(YNEW,NMAX,XX,NMAX)
DEGREE = DEG(PERT,XX)
PERT = CHFORM(PERT,DEGREE,XX)
WRITE " THE POLYNOMIAL SOLUTION IS", YNEW
WRITE " THE PERTURBATION IS", PERT
CWRIT(C,NMAX)

# TRANSFORM RESULTS BACK TO RANGE (A,B), IF NECESSARY.

IF ( NOT (RANGE = (-1,1)) ) DO
  WRITE " SOLUTION TRANSFORMED BACK TO GIVEN RANGE (A,B)"
  TRANS2(RANGE,YNEW,YTRANS)
  WRITE " THE TRANSFORMED POLYNOMIAL SOLUTION IS", YTRANS
  TRANS2(RANGE,P1,PTRANS)
  WRITE " THE TRANSFORMED PERTURBATION IS", PTRANS
DOEND

RETURN
END  # END OF PROCEDURE EPIN.
## List of Level 3 Procedures

<table>
<thead>
<tr>
<th>Procedure</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>RECC11</td>
<td>35</td>
</tr>
<tr>
<td>RECT11</td>
<td>36</td>
</tr>
<tr>
<td>RECT12</td>
<td>37</td>
</tr>
<tr>
<td>RECT21</td>
<td>39</td>
</tr>
<tr>
<td>GSC11</td>
<td>40</td>
</tr>
<tr>
<td>GSC12</td>
<td>41</td>
</tr>
<tr>
<td>GST12</td>
<td>42</td>
</tr>
<tr>
<td>GST21</td>
<td>44</td>
</tr>
<tr>
<td>Procedure</td>
<td>Page</td>
</tr>
<tr>
<td>-----------</td>
<td>------</td>
</tr>
<tr>
<td>RECC11</td>
<td>B-37</td>
</tr>
<tr>
<td>RECC21</td>
<td>B-38</td>
</tr>
<tr>
<td>RECT11</td>
<td>B-39</td>
</tr>
<tr>
<td>RECT12</td>
<td>B-40</td>
</tr>
<tr>
<td>RECT21</td>
<td>B-42</td>
</tr>
<tr>
<td>GSC11</td>
<td>B-43</td>
</tr>
<tr>
<td>GSC12</td>
<td>B-44</td>
</tr>
<tr>
<td>GSC21</td>
<td>B-45</td>
</tr>
<tr>
<td>GSC22</td>
<td>B-46</td>
</tr>
<tr>
<td>GST12</td>
<td>B-47</td>
</tr>
<tr>
<td>GST21</td>
<td>B-49</td>
</tr>
</tbody>
</table>
PROCEDURE RECC11(ORDER,POLY,HALFN)
  INTEGER VALUE ORDER
  LONG ALGEBRAIC ARRAY VALUE POLY
  INTEGER HALFN

# THIS PROCEDURE RETURNS THE RECURRENCE EQUATION DEFINING THE
# CHUBYSHEV SERIES COEFFICIENTS OF A LINEAR ODE.
# INPUT PARAMETERS:
#  ORDER — ORDER OF ODE;
#  POLY — POLYNOMIAL COEFFICIENTS OF ODE.
# OUTPUT PARAMETERS:
#  HALFN — "HALF" THE LENGTH OF THE RECURRENCE EQUATION;
#  EQN — THE RECURRENCE EQUATION RETURNED IN THE FORM
#        V(0)*XCR(-H) + ------ + V(2H)*XCR(H)
#        WHERE H=HALFN, R IS AN INDETERMINATE REPRESENTED
#        BY RR. XCR(J) REPRESENTS C(R+J) AND V(1) ARE
#        RATIONAL EXPRESSIONS IN R.
# EXTERNAL VARIABLES REQUIRED: MSGLV,RR,XTR,XCR.
# PACKAGE PROCEDURE REQUIRED: GENCHE.
# SYSTEM PROCEDURES REQUIRED: GETBLK.
# DECLARATION IN CALLING PROCEDURE: LONG ALGEBRAIC ALTRAN RECC11.

INTEGER 1)
LONG ALGEBRAIC EQN,GENPOL,COEF
EXTERNAL INTEGER MSGLV,
EXTERNAL LONG ALGEBRAIC RR
EXTERNAL LONG ALGEBRAIC ARRAY XTR,XCR
LONG ALGEBRAIC ALTRAN GENCHE

IF (MSGLV>3) WRITE "ENTER RECC11"

# OBTAIN THE GENERATING POLYNOMIAL GENPOL.

GENPOL = GENCHE(POLY,ORDER,HALFN)

# OBTAIN THE RECURRENCE EQUATION.

  EQN = 0
 DO J = -HALFN,HALFN
    COEF = GETBLK(GENPOL,XTR(0,J))
    EQN = EQN + COEF*RR=RR-J*XCR(-J)
 END

IF (MSGLV>3) WRITE GENPOL,EQN,HALFN, "EXIT RECC11"

RETURN(EQN)

END # END OF PROCEDURE RECC11.
PROCEDURE RECTIL(POLY,ORDER,MAXK,MINK,CODE)
INTEGER VALUE ORDER
INTEGER ARRAY VALUE CODE
INTEGER MINK,MAXK
LONG ALGEBRAIC ARRAY POLY

// THIS PROCEDURE RETURNS THE RECURRERECE EQUATION
// DEFING THE POWER SERIES COEFFICIENTS OF A LINEAR
// ODE, IN EITHER THE DIFFERENTIATED OR INTEGRATED FORM.
// INPUT PARAMETER:
// ORDER - ORDER OF ODE;
// POLY - POLYNOMIAL COEFFICIENTS OF ODE;
// CODE - SOLUTION CODE.
// OUTPUT PARAMETERS:
// MAXK,MINK - INTEGER VALUES DEFINED BELOW;
// EQN - THE REQUIRED RECURRERECE EQUATION RETURNED
// IN THE FORM
// \( Y(x) = \sum_{i=0}^{H} x^i \) \( Y(x) \) \( \sum_{i=0}^{H} x^i \)
// WHERE \( H = -MINK, H = -MAXK \) IS AN
// INDETERMINATE REPRESENTED AS RR,
// \( XAR(j) \) REPRESENTS \( A_{R+1} \), AND \( V(i) \) ARE
// RATIONAL FUNCTIONS IN R.
// EXTERNAL VARIABLES REQUIRED: MSGVL,RR,XXR,XAR.
// PACKAGE PROCURURES REQUIRED: GENINT, GENDIFF.
// SYSTEM PROCURURES REQUIRED: GETBLK.
// DECLARATION IN CALLING PROCEDURE: LONG ALGEBRAIC ALTRAN RECTIL.

INTEGER JJ,MLK
LONG ALGEBRAIC EQN,P,GENPOL,AJ
EXTERNAL INTEGER MSGVL
EXTERNAL LONG ALGEBRAIC RR
EXTERNAL LONG ALGEBRAIC ARRAY XXR,XAR
LONG ALGEBRAIC ALTRAN GENINT,GENDIFF

IF (MSGVL>3) WRITE "ENTER RECTIL"

// OBTAIN THE GENERATING POLYNOMIAL GENPOL.

IF (CODE(3) < 4) DO  # INTEGRATED FORM.
  GENPOL = GENINT(POLY,ORDER,MAXK)
  MINK = 0
END DO
ELSE DO  # DIFFERENTIATED FORM.
  GENPOL = GENDIFF(POLY,ORDER,MAXK,MINK)
END DO

// OBTAIN THE RECURRERECE EQUATION.

EQN = 0
DO J = MINK,MAXK
  AJ = GETBLK(GENPOL,XXR(J),J)
  MJ = MINK-J
  EQN = EQN + AJ(RR=RR+MJ)*XAR(MJ)
END DO
MAXK = MAXK - MINK

IF (MSGVL>3) WRITE GENPOL,EQN,MINK,MAXK,"EXIT RECTIL" RETURN(EQN)

END END OF PROCEDURE RECTIL
PROCEDURE RECT12(POLY,ORDER,H1,H2,CODE)
INTEGER VALUE ORDER,CODE
INTEGER H2,H1
LONG ALGEBRAIC ARRAY POLY

# THIS PROCEDURE RETURNS THE RECURRENCE EQUATION
# DEFINING THE POWER SERIES COEFFICIENTS OF A LINEAR
# ODE, IN EITHER THE DIFFERENTIATED OR INTEGRATED FORM.
# THE RECURRENCE EQUATION FOR THE INTEGRATED FORM IS
# WITHOUT EXPLICITLY CONVERTING THE ODE TO INTEGRATED FORM.
# INPUT PARAMETER:
# ORDER - ORDER OF ODE;
# POLY - POLYNOMIAL COEFFICIENTS OF ODE;
# CODE - SOLUTION CODE.
# OUTPUT PARAMETERS:
# MAXK,MINK - INTEGER VALUES DEFINED BELOW;
# EQN - THE REQUIRED RECURRENCE EQUATION RETURNED
# IN THE FORM
# V(-H2)*XAR(-H2) + ... + V(-H1)*XAR(-H1)
# WHERE H1=-MINK,H2=-MAXK, R IS AN
# INDETERMINATE REPRESENTED AS RR.
# XAR(J) REPRESENTS A(K+J), AND V(I) ARE
# RATIONAL FUNCTIONS IN R.
# EXTERNAL VARIABLES REQUIRED: MSGVVL,RR,XXR,XAR.
# PACKAGE PROCEDURES REQUIRED: GENDIF.
# SYSTEM PROCEDURES REQUIRED: GETBLK.
# DECLARATION IN CALLING PROCEDURE: LONG ALGEBRAIC ALTRAN RECT12.

INTEGER I,K,J
LONG ALGEBRAIC EQN,P,GENPOL,AJ
EXTERNAL INTEGER MSGVVL
EXTERNAL LONG ALGEBRAIC RR
EXTERNAL LONG ALGEBRAIC ARRAY XXR,XAR
LONG ALGEBRAIC ALTRAN GENDIF

IF (MSGVVL>3) WRITE "ENTER RECT12"

# OBTAIN THE GENERATING POLYNOMIAL GENPOL.

GENPOL = GENDIF(POLY,ORDER,H1,H2)
# Obtain the recurrence equation.

IF (CODE > 2) DO  # Differentiated form.
    EQN = 0
    DO J = H2, H1
       A1 = GETBLK(GENPOL, XXR(J), I)
       EQN = EQN + A1*(RR = RR - J)*XAR(-J)
       GENPOL = GENPOL - A1*XAR(J)
    DOEND
    H2 = -H2
    DOEND

ELSE DO  # Integrated form.
    EQN = 0
    DO J = H2, H1
       MJ = H2 - J
       A1 = GETBLK(GENPOL, XXR(J), I)
       EQN = EQN + A1*(RR = RR + MJ)*XAR(MJ)
       GENPOL = GENPOL - A1*XXR(J)
    DOEND
    H1 = H1 - H2
    H2 = 0
    P = 1
    DO J = 0, ORDER - 1
       P = P*R(RR - I)
    DOEND
    K = ORDER - H1
    IF (K > 0) P = P*R(RR = RR + K)
    EQN = EQN/P
    DOEND

IF (MSGLVL > 3) WRITE GENPOL, EQN, H2, H1, "EXIT RECT12"
RETURN(EQN)

END # End of procedure RECT12.
PROCEDURE RECT21(POLY, ORDER, MINK, MAXK, CODE, GENPOL)
  INTEGER VALUE ORDER, CODE
  INTEGER MINK, MAXK
  LONG ALGEBRAIC GENPOL
  LONG ALGEBRAIC ARRAY POLY

  # THIS PROCEDURE RETURNS THE RECURRENCE EQUATION DEFINING
  # THE CANONICAL POLYNOMIALS ASSOCIATED WITH A LINEAR ODE.
  # IN EITHER THE DIFFERENTIATED OR INTEGRATED FORM.
  # INPUT PARAMETERS:
  #    ORDER  = ORDER OF ODE;
  #    POLY   = POLYNOMIAL COEFFICIENTS OF ODE;
  #    CODE   = SOLUTION CODE.
  # OUTPUT PARAMETERS:
  #    GENPOL  = THE GENERATING POLYNOMIAL RETURNED IN THE FORM
  #              U(H1)*XXR(H1) + ... + U(H3)*XXR(H3)
  #    WHERE H1=MINK, H2=MAXK, R IS AN INDETERMINATED
  #    REPRESENTED AS RR, X*R(R+1) IS REPRESENTED BY XXR(R),
  #    AND U(I) ARE RATIONAL EXPRESSIONS IN R;
  #    MINK, MAXK = DEFINED AS ABOVE;
  #    EQN  = THE REQUIRED RECURRENCE EQUATION (RETURNED) IN THE
  #           FORM V(H1)*XQR(H1) + ... + V(-1)*XQR(-1) + V(0)*XQR(-1),
  #    WHERE H = H1-H2*Q(R+1) IS REPRESENTED BY XQR(R), AND
  #    V(I) ARE RATIONAL EXPRESSIONS IN R.
  # EXTERNAL VARIABLES REQUIRED: MSGVL1, RR, XX, XQR.
  # PACKAGE PROCEDURES REQUIRED: GENINT, GENDIF.
  # SYSTEM PROCEDURES REQUIRED: GETBLK.
  # DECLARATION IN CALLING PROCEDURE: LONG ALGEBRAIC ALTRAN RECT21.

  INTEGER I,J,MK, MINK
  LONG ALGEBRAIC EQN, P, A1
  EXTERNAL INTEGER MSGVL1
  EXTERNAL LONG ALGEBRAIC RR, XX
  EXTERNAL LONG ALGEBRAIC ARRAY XQR, XQR2
  LONG ALGEBRAIC ALTRAN GENINT, GENDIF

  IF (MSGVL1>3) WRITE "ENTER RECT21"
  # OBTAIN THE GENERATING POLYNOMIAL, GENPOL.

  IF (CODE<3) DO  # INTEGRATED FORM.
    GENPOL = GENINT(POLY, ORDER, MAXK)
    MINK = 0
  DOEND

  ELSE DO  # DIFFERENTIATED FORM.
    GENPOL = GENDIF(POLY, ORDER, MAXK, MINK)
  DOEND

  # OBTAIN THE RECURRENCE EQUATION.

  P = 0
  DO I = MINK, MAXK-1
    A1 = GETBLK(GENPOL, XXR(I, 1))
    P = P + A1*XQR(I-MAXK)
  DOEND

  A1 = GETBLK(GENPOL, XXR(MAXK, 1))
  EQN = (XXR(-MAXK) - P*A1)*RR = RR - MAXK

  IF (MSGVL1>3) WRITE GENPOL, EQN, MINK, MAXK, "EXIT RECT21"
  RETURN(EQN)
END
PROCEDURE GSCHI(NMAX,ORDER,EQN,HALFN,MAXB,CONDN,XSUB,RHS,C)
   INTEGER VALUE NMAX,ORDER,MAXB,HALFN
   RATIONAL ARRAY VALUE XSUB
   LONG ALGEBRAIC ARRAY VALUE CONDN
   LONG ALGEBRAIC ARRAY C
   LONG ALGEBRAIC VALUE EQN,RHS

   # THIS PROCEDURE SOLVES THE RECURRENCE EQUATION DEFINING
   # THE CHEBYSHEV COEFFICIENTS OF THE INTEGRATED
   # FORM OF A LINEAR ODE, FOR A FINITE POLYNOMIAL APPROXIMATION.
   # THE SOLUTION IS OBTAINED BY BACKWARD RECURRENCE.
   # INPUT PARAMETERS:
   #    NMAX   - DEGREE OF APPROXIMATION DESIRED;
   #    ORDER  - ORDER OF ODE;
   #    EQN,HALFN -- INFORMATION ABOUT RECURRENCE EQUATION;
   #    MAXB,XSUB,CONDN -- INFORMATION ABOUT ASSOCIATED CONDITIONS;
   #    RHS    -- RIGHT-HAND-SIDE POLYNOMIAL OF ODE.
   OUTPUT PARAMETERS:
   #    C       - THE REQUIRED CHEBYSHEV COEFFICIENTS;
   #    YSTAR   - THE POLYNOMIAL APPROXIMATION (RETURNED);
   # EXTERNAL VARIABLES REQUIRED: MSLVL,AX,XPAR,RR,XX.
   # PACKAGE PROCEDURES REQUIRED: CHFORM,VALCII,PSCH1,TCS.
   # SYSTEM PROCEDURES REQUIRED: TNEW,GETBLK.
   # DECLARATION IN CALLING PROCEDURE: LONG ALGEBRAIC ALTRAN GSCHI.

   INTEGER L,L,M,NMIN,DEGREE,DEGRH,N,
   REAL TOLD,TNEW
   LONG ALGEBRAIC 1,COFF,LCVAL,LNEW,EQN,YSTAR
   LONG ALGEBRAIC ARRAY C,PARLIS,PARVAL
   EXTERNAL INTEGER MSLVL.
   EXTERNAL LONG ALGEBRAIC RR,XX
   EXTERNAL LONG ALGEBRAIC ARRAY XPAR,XXPAR
   LONG ALGEBRAIC ALTRAN CHFORM,PSCH1
   LONG ALGEBRAIC ARRAY ALTRAN TCS

   IF (MSLVL>3) WRITE "ENTER GSCHI"
   TIME(TOLD)

   # SOLVE THE RECURRENCE EQUATION FOR PARAMETRIC SOLUTION.
   YSTAR = PSCH1(NMAX,NMIN,ORDER,RHS,EQN,HALFN,C)
   TNEW = TIME(TOLD)
   IF(MSLVL>3)...
   WRITE " TIME TO COMPUTE PARAMETRIC SOLUTION WAS",TNEW
   IF (MSLVL>3) WRITE YSTAR,RHS

   # DETERMINE THE VALUES OF THE PARAMETERS.
   VALCII(C,NMAX,ORDER,YSTAR,EQN,HALFN,HALFN+NMIN,
        MAXB,CONDN,XSUB,RHS,PARLIS,PARVAL)
   YSTAR = YSTAR(PARLIS=PARVAL)
   C = TCS(CHFORM(YSTAR NMAX,XX),NMAX)
   TNEW = TIME(TOLD)
   IF (MSLVL>3)...
   WRITE " TIME TO ASSIGN VALUES TO PARAMETERS WAS",TNEW
   IF (MSLVL>3) WRITE YSTAR,C,PARLIS,PARVAL,"EXIT GSCHI"

   RETURN(YSTAR)
END # END OF PROCEDURE GSCHI.
PROCEDURE GSCI(NMAX,ORDER,EQN,HALFN,MAXSUB,XSUB,CONDN,XC,CHFRS)
INTEGER VALUE NMAX,ORDER,MAXSUB,HALFN
RATIONAL ARRAY VALUE XSUB
LONG ALGEBRAIC VALUE EQN,RHS
LONG ALGEBRAIC ARRAY C
LONG ALGEBRAIC ARRAY VALUE CONDN

/* THIS PROCEDURE SOLVES THE RECURRANCE EQUATION DEFINING 
THE CHEBYSHEV COEFFICIENTS OF THE INTEGRATED 
FORM OF A LINEAR ODE, FOR A FINITE POLYNOMIAL APPROXIMATION. 
THE SOLUTION IS OBTAINED BY GAUSSIAN ELIMINATION. 
INPUT PARAMETERS: 
NMAX -- DEGREE OF APPROXIMATION DESIRED; 
ORDER -- ORDER OF ODE; 
EQN,HALFN -- INFORMATION ABOUT RECURRANCE EQUATION; 
MAXSUB,XSUB,CONDN -- INFORMATION ABOUT ASOCIATED CONDITIONS; 
RHS -- RIGHT-HAND-SIDE POLYNOMIAL OF ODE. 
OUTPUT PARAMETERS: 
C -- THE REQUIRED CHEBYSHEV COEFFICIENTS; 
YSTAR -- THE POLYNOMIAL APPROXIMATION (RETURNED); 
EXTERNAL VARIABLES REQUIRED: MSGLVL,XX,XPAR. 
PACKAGE PROCEDURES REQUIRED: EVAL,VALC12,CHFORM. 
SYSTEM PROCEDURES REQUIRED: TIME,DEG,PINTN. 
DECLARATION IN CALLING PROCEDURE: LONG ALGEBRAIC ALTRAN GSCI2. */

INTEGER DEGRHS, 1 REAL TOLD,TNEW
LONG ALGEBRAIC YSTAR
LONG ALGEBRAIC ARRAY PARLIS,PARVAL
EXTERNAL INTEGER MSGLVL
EXTERNAL LONG ALGEBRAIC XX
EXTERNAL LONG ALGEBRAIC ARRAY XPAR
LONG ALGEBRAIC ALTRAN EVAL,CHFORM,PINTN

IF (MSGLVL>3) WRITE " ENTER GSCI2" TIME(TOLD);

/* OBTAIN PARAMETRIC SOLUTION. */
YSTAR = EVAL(XPAR,NMAX,XX)
IF (RHS<>0) DO
   DEGRHS = DEG(RHS,XX) + ORDER
   RHS = CHFORM(PINTN(RHS,ORDERSXX),DEGRHS,XX)
END
TNEW = TIME(TOLD);
IF (MSGLVL>2) WRITE " TIME TO COMPUTE PARAMETRIC SOLUTION WAS",TNEW
IF (MSGLVL>3) WRITE YSTAR,RHS

/* ASSIGN VALUES TO PARAMETERS. */
VALC12(NMAX,ORDER,YSTAR,EQN,HALFN, 
       MAXSUB,CONDN,XSUB,RHS,PARLIS,PARVAL)
YSTAR = YSTAR(PARLIS=PARVAL)
C = PARVAL
TNEW = TIME(TOLD)
IF (MSGLVL>2) WRITE " TIME TO ASSIGN VALUES TO PARAMETERS WAS", TNEW
IF (MSGLVL>3) WRITE C,YSTAR,PARLIS,PARVAL, "EXIT GSCI2"

RETURN(YSTAR)
END OF PROCEDURE GSCI2
PROCEDURE GST12(EQN,N,H1,H2,CODE,PERT,
MAXSUB,CONDN,XSUB,RPOLY,ORDER)
INTEGER VALUE N,ORDER,MAXSUB,H1,H2,CODE
LONG ALGEBRAIC VALUE EQN,RPOLY
LONG ALGEBRAIC PERT
LONG ALGEBRAIC ARRAY VALUE CONDN
RATIONAL ARRAY VALUE XSUB

# THIS PROCEDURE SOLVES THE RECURRENCE EQUATION DEFINING
# THE POWER SERIES COEFFICIENTS OF THE SOLUTION FUNCTION
# OF A PERTURBED LINEAR ODE IN EITHER INTEGRATED
# OR DIFFERENTIATED FORM.
# INPUT PARAMETERS:
#   N – DEGREE OF APPROXIMATION DESIRED;
#   ORDER – ORDER OF ODE;
#   CODE – SOLUTION CODE;
#   EQN,HI,HI2 – INFORMATION ABOUT RECURRENCE EQUATION;
#   MAXSUB,XX,CONDN – INFORMATION ABOUT ASSOCIATED
#   CONDITIONS;
#   RPOLY – RIGHT-HAND-SIDE POLYNOMIAL OF ODE.
# OUTPUT PARAMETER:
#   YSTAR – THE REQUIRED SOLUTION FUNCTION (RETURNED).
# PERT – THE PERTURBATION ON THE ODE.
# EXTERNAL VARIABLES REQUIRED: MSG_LVL, XX, XAR.
# PACKAGE PROCEDURES REQUIRED: PST12, VAL12.
# SYSTEM PROCEDURES REQUIRED: GETBLK, TIME.
# DECLARATION IN CALLING PROCEDURE: LONG ALGEBRAIC ALTRAN GST12.

INTEGER IJK,HNPERT,NP
INTEGER ARRAY HIST
REAL TOLD,TNEW
LONG ALGEBRAIC YSTAR
LONG ALGEBRAIC ARRAY (-H1,H2) COEF
LONG ALGEBRAIC ARRAY (0N) B
LONG ALGEBRAIC ARRAY PAR1,PARVAL,RCOEF
EXTERNAL INTEGER MSG_LVL
EXTERNAL LONG ALGEBRAIC XX
EXTERNAL LONG ALGEBRAIC ARRAY XAR

IF (MSG_LVL>3) WRITE "ENTER GST12"
TIME(TOLD)

SET UP ARRAY OF COEFFICIENTS IN RECURRENCE EQN.

DO I = -H1,H2
   COEF(I) = GETBLK(EQN,XAR(I),)
DOEND
# Obtain parametric approximation, ystar, and perturbation, pert.

PST12(N, ORDER, H, H1, H2, NP, NP, RCOEF, RCOEF, B, YSTAR, PERT, CODE)

TNEW = TIME(TOLD)
IF (MSGLVL > 3) WRITE "TIME TO COMPUTE PARAMETRIC SOLUTION WAS", TNEW
IF (MSGLVL > 3) WRITE H, NP, I1IST, NP, RCOEF, RCOEF, YSTAR, PERT

# Compute the values of the parameters, if any.

IF (H + NP > 0) DO
VAL12(YSTAR, PERT, B, N, ORDER, CODE, H, NP, COEF, RCOEF,
MAXSUB, CONDN, XSUB, I1IST, H1, H2, PARLIS, PARVAL)
YSTAR = YSTAR(PARLIS = PARVAL)
PRT = PRT(PARLIS = PARVAL)
ENDDO
TNEW = TIME(TOLD)
IF (MSGLVL > 3) WRITE "TIME TO ASSIGN VALUES TO PARAMETERS WAS", TNEW
IF (MSGLVL > 3) WRITE YSTAR, PERT, "EXIT GST12"
RETURN(YSTAR)
END # end of procedure GST12.
PROCEDURE GST21(N, ORDER, POLY, CODE, H1, H2, EQN, GENPOL, RPOLY, MAXSUB, CONDN, XSUB, PERT)
INTEGER VALUE N, CODE, ORDER, H1, H2, MAXSUB
RATIONAL ARRAY VALUE XSUB
LONG ALGEBRAIC PERT
LONG ALGEBRAIC VALUE RPOLY, EQN, GENPOL
LONG ALGEBRAIC ARRAY VALUE CONDN, POLY

# THIS PROCEDURE OBTAINS THE SOLUTION FUNCTION OF A PERTURBED
# LINEAR ODE IN EITHER INTEGRATED OR DIFFERENTIATED FORM.
# THE SOLUTION IS OBTAINED USING THE CANONICAL POLYNOMIALS
# ASSOCIATED WITH THE LINEAR OPERATOR DEFINED BY THE ODE.
# INPUT PARAMETER:
# N - DEGREE OF APPROXIMATION REQUIRED;
ORDER - ORDER OF ODE;
POLY - POLYNOMIAL COEFFICIENTS OF ODE;
RPOLY - RIGHT-HAND-SIDE POLYNOMIAL OF ODE;
EQN, GENPOL, H1, H2 - INFORMATION ABOUT THE RECURRENCE
   EQUATION;
MAXSUB, XSUB, CONDN - INFORMATION ABOUT THE ASSOCIATED
   CONDITIONS
# OUTPUT PARAMETERS:
YSTAR - THE REQUIRED SOLUTION FUNCTION (RETURNED);
PERT - PERTURBATION ON ODE.
EXTERNAL VARIABLES REQUIRED: MSLVL, XX, XPAR.
PACKAGE PROCEDURES REQUIRED: GST21, VALT21, CANONP.
SYSTEM PROCEDURES REQUIRED: TIME, PINT.
DECLARATION IN CALLING PROCEDURE: LONG ALGEBRAIC ALTRAN GST21.

INTEGER L,M,NMM,DEGREE,R,S1,V1,H
INTEGER ARRAY S
REAL TOLD,TNEW
LONG ALGEBRAIC YSTAR
LONG ALGEBRAIC ARRAY PARS,PARVAL,PARV
EXTERNAL INTEGER MSLVL
EXTERNAL LONG ALGEBRAIC XX
EXTERNAL LONG ALGEBRAIC ARRAY XPAR

IF (MSLVL<=1) WRITE " ENTER GST21"
TIME(TOLD)

# OBTAIN THE CANONICAL POLYNOMIALS.
CANONP(N, POLY, ORDER, CODE, H1, H2, EQN, GENPOL, Q, H, S1, V, VI)

# OBTAIN RIGHT-HAND-SIDE POLYNOMIAL.
R = S1-V1 + ORDER
IF (CODE<3) DO  # INTEGRATED FORM
   R = S1-V1
   DO 1 = ORDER
      RPOLY = PINT(RPOLY, XX) + XPAR(S1+1)
   BOUND
   DOEND

<table>
<thead>
<tr>
<th>Procedure</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>GENCHE</td>
<td>46</td>
</tr>
<tr>
<td>GENDIF</td>
<td>47</td>
</tr>
<tr>
<td>GENINT</td>
<td>48</td>
</tr>
<tr>
<td>PSC11</td>
<td>49</td>
</tr>
<tr>
<td>PST12</td>
<td>50</td>
</tr>
<tr>
<td>PST21</td>
<td>52</td>
</tr>
<tr>
<td>VALC11</td>
<td>53</td>
</tr>
<tr>
<td>VALC12</td>
<td>55</td>
</tr>
<tr>
<td>VALT12</td>
<td>56</td>
</tr>
<tr>
<td>VALT21</td>
<td>58</td>
</tr>
</tbody>
</table>
## List of Level 4 Procedures

<table>
<thead>
<tr>
<th>Procedure</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>GENCHE</td>
<td>46</td>
</tr>
<tr>
<td>GENDIF</td>
<td>47</td>
</tr>
<tr>
<td>GENINT</td>
<td>48</td>
</tr>
<tr>
<td>PSC11</td>
<td>49</td>
</tr>
<tr>
<td>PST12</td>
<td>50</td>
</tr>
<tr>
<td>PST21</td>
<td>52</td>
</tr>
<tr>
<td>VALC11</td>
<td>53</td>
</tr>
<tr>
<td>VALC12</td>
<td>55</td>
</tr>
<tr>
<td>VALT12</td>
<td>56</td>
</tr>
<tr>
<td>VALT21</td>
<td>58</td>
</tr>
</tbody>
</table>
PROCEDURE GENCHE(POLY,ORDER,HALFN)
  INTEGER VALUE ORDER
  INTEGER HALFN
  LONG ALGEBRAIC ARRAY POLY

  # THIS PROCEDURE COMPUTES THE GENERATING POLYNOMIAL FOR
  # THE CHEBYSHEV SERIES SOLUTION OF THE INTEGRATED FORM
  # OF A LINEAR OF GIVEN ORDER. THE POLYNOMIAL COEFFICIENTS
  # ARE GIVEN IN ARRAY POLY AND THE GENERATING POLYNOMIAL IS
  # RETURNED IN THE FORM
  #  V(MINK)*XTR(MINK) + ..... + V(MAXK)*XTR(MAXK)
  # WHERE R IS AN INDETERMINATE REPRESENTED AS RR,
  # XTR(I) REPRESENTS THE (R+J)TH CHEBYSHEV POLYNOMIAL,
  # AND V(I) ARE RATIONAL EXPRESSIONS IN R.
  #EXTERNAL VARIABLES REQUIRED: XX.
  #PACKAGE PROCEDURES REQUIRED: INTEGR,CHFORM,PRODTR,INFORM.
  #SYSTEM PROCEDURES REQUIRED: DEG,IMAX.
  #DECLARATION IN CALLING PROCEDURE: LONG ALGEBRAIC ALTRAN GENCHE.

  INTEGER DEGREE, I,J,M,HALF
  LONG ALGEBRAIC P,GENPOL,TERM
  EXTERNAL LONG ALGEBRAIC XX
  LONG ALGEBRAIC ALTRAN PRODTR,INTEGR,CHFORM
  LONG ALGEBRAIC ARRAY ALTRAN INFORM

  POLY = INFORM(ORDER,POLY,XX)
  HALFN = 0
  GENPOL = 0

  DO M = 0,ORDER
    P = POLY(M)
    DEGREE = DEG(P,XX)
    P = CHFORM(P,DEGREE,XX)
    TERM = PRODTR(P,DEGREE)
    HALF = DEGREE

    DO I = 1,M
      TERM = INTEGR(TERM,HALF)
      HALF = HALF + 1
    DOEND

    GENPOL = GENPOL + TERM
    HALFN = IMAX(HALFN,HALF)
  DOEND

  RETURN(GENPOL)

END # END OF PROCEDURE GENCHE.
PROCEDURE GENDIF(POLY, ORDER, MAXK, MINK)
    INTEGER VALUE ORDER
    LONG ALGEBRAIC ARRAY VALUE POLY
    INTEGER MAXK, MINK

    # THIS PROCEDURE COMPUTES THE GENERATING POLYNOMIAL FOR
    # THE POWER SERIES SOLUTION OF THE DIFFERENTIATED FORM
    # OF A LINEAR OF GIVEN ORDER. THE POLYNOMIAL COEFFICIENTS
    # ARE GIVEN IN ARRAY POLY AND THE GENERATING POLYNOMIAL IS
    # RETURNED IN THE FORM
    #   V(MINK)*XXR(MINK) + ... + V(MAXK)*XXR(MAXK)
    # WHERE R IS AN INDETERMINATE REPRESENTED AS RR,
    # XXR(J) REPRESENTS X**(R+J) AND V(I) ARE RATIONAL EXPRESSIONS
    # IN R.
    # EXTERNAL VARIABLES REQUIRED: XX, RR, XXR.
    # PACKAGE PROCEDURES REQUIRED: NONE.
    # SYSTEM PROCEDURES REQUIRED: DEG, GETBLK, IMAX, IMIN.
    # DECLARATION IN CALLING PROCEDURE: LONG ALGEBRAIC ALTRAN GENDIF.

    INTEGER I, J, M, MJ, DEGREE, M1, M2
    LONG ALGEBRAIC P, GENPOL, TERM, UR, AJ
    EXTERNAL LONG ALGEBRAIC XX, RR
    EXTERNAL LONG ALGEBRAIC ARRAY XXR

    UR = 1
    GENPOL = 0

    DO M = 0, ORDER
        P = POLY(M)
        IF (P<>0) DO
            DEGREE = DEG(P, XX)
            TERM = 0

            DO J = DEGREE, 0, -1
                AJ = GETBLK(P, XX, J)
                IF (AJ <> 0) DO
                    I = J
                    TERM = TERM + AJ*XXR(-M+J)
                END DO
            END DO

            GENPOL = GENPOL + UR*TERM
            M1 = DEGREE - M
            M2 = I - M
            IF (NULL(MAXK)) MAXK = M1
            ELSE MAXK = IMA(MAXK, M1)
            IF (NULL(MINK)) MINK = M2
            ELSE MINK = IMIN(MINK, M2)
        END IF
        UR = UR *(RR - M)
    END DO

    RETURN(GENPOL)

END # END OF PROCEDURE GENDIF.
PROCEDURE GENINT(POLY,ORDER,MAXK)
INTEGER VALUE ORDER
INTEGER MAXK
LONG ALGEBRAIC ARRAY POLY

# THIS PROCEDURE COMPUTES THE GENERATING POLYNOMIAL FOR
# THE POWER SERIES SOLUTION OF THE INTEGRATED FORM
# OF A LINEAR OF GIVEN ORDER. THE POLYNOMIAL COEFFICIENTS
# ARE GIVEN IN ARRAY POLY AND THE GENERATING POLYNOMIAL IS
# RETURNED IN THE FORM
# V(MINK)*XXR(MINK) + .... + V(MAXK)*XXR(MAXK)
# WHERE R IS AN INDETERMINATE REPRESENTED AS RR,
# XXR(J) REPRESENTS X**(R+J) AND V(I) ARE RATIONAL EXPRESSIONS
# IN R.
# EXTERNAL VARIABLES REQUIRED: XX.
# PACKAGE PROCEDURES REQUIRED: INT,PRODXR,INFORM.
# SYSTEM PROCEDURES REQUIRED: DEG,IMAX.
# DECLARATION IN CALLING PROCEDURE: LONG ALGEBRAIC ALTRAN GENINT.

INTEGER DEGREE,I,J,M
LONG ALGEBRAIC P,GENPOL,TERM
EXTERNAL LONG ALGEBRAIC XX
LONG ALGEBRAIC ALTRAN PRODXR,INT
LONG ALGEBRAIC ARRAY ALTRAN INFORM

POLY = INFORM(ORDER,POLY,XX)
MAXK = 0
GENPOL = 0

DO M = 0,ORDER
   # ADD APPROPRIATE TERM INTO GENPOL.
   P = POLY(M)
   DEGREE = DEG(P,XX)
   TERM = PRODXR(P,DEGREE,XX)
   DO I = 1,M
      TERM = INT(TERM,DEGREE)
      DEGREE = DEGREE + 1
   DOEND
   GENPOL = GENPOL + TERM
MAXK = IMAX(MAXK,DEGREE)
DOEND

RETURN(GENPOL)
END # END OF PROCEDURE GENINT.
PROCEDURE PSC11(NMAX,NMIN,ORDER,RHS,EQN,HALF, C
    INTEGER VALUE NMAX,ORDER,HALF
    INTEGER NMIN
    LONG ALGEBRAIC ARRAY C
    LONG ALGEBRAIC VALUE RHS,EQN

    # THIS PROCEDURE RETURNS THE PARAMETRIC CHEBYSHEV SERIES SOLUTION
    # (OF DEGREE N) OF A LINEAR ODE OF GIVEN ORDER, USING
    # BACKWARD RECURRENCE.
    # EXTERNAL VARIABLES REQUIRED: RR,XX,XPAR,XCR,XT.
    # PACKAGE PROCEDURES REQUIRED: CHFORM, EVAL, BAKREC.
    # DECLARATION IN CALLING PROCEDURE: LONG ALGEBRAIC ALTRAN PSC11.

    INTEGER DEGRHS,I,J,N
    LONG ALGEBRAIC NEWEQN,YSTAR,LCVAL,LCOEF
    LONG ALGEBRAIC ARRAY INDET,RLIST
    EXTERNAL LONG ALGEBRAIC RR,XX
    EXTERNAL LONG ALGEBRAIC ARRAY XPAR, XCR, XT
    LONG ALGEBRAIC ALTRAN CHFORM, PINTN, EVAL
    LONG ALGEBRAIC ARRAY ALTRAN BAKREC

    # COMPUTE NMIN.

    LCOEF = GETBLK(EQN,XCR(-HALF),1)
    NMIN = 0
    DO J = 0,NMAX
        LCVAL = LCOEF(RR=J+HALF)
        IF (LCVAL==0) NMIN = J+1
    DOEND
    WRITE NMIN

    # NOW DETERMINE THE LIST OF RIGHT-HAND-SIDES INVOLVED IN
    # SOLVING THE RECURRENCE EQUATION FOR C(NMIN),....,C(NMAX).

    DEGRHS = DEG(RHS,XX)
    IF (RHS<>0) DEGRHS = DEGRHS+ORDER
    RHS = CHFORM(PINTN(RHS,ORDERSXX),DEGRHS,XX)
    RLIST = 0$(1)
    DO I = NMIN+HALF,DEGRHS
        RLIST = (RLIST,GETBLK(RHS,XT(I),1))
    DOEND

    N = 2*HALF
    NEWEQN = EQN(RR = RR+HALF)
    INDET = IS(XCR(-HALF))
    DO I = 1,N
        INDET = (INDET,XCR(-HALF+1))
    DOEND

    C = BAKREC(NEWEQN,N,ISINDET,RR,RLIST,NMIN,NMAX,XPAR,HALF)

    # INTRODUCE FURTHER PARAMETERS FOR UNDEFINED COEFFICIENTS.

    DO I = 0,NMIN-1
        C(I) = XPAR(HALF+I+1)
    DOEND

    YSTAR = EVAL(C,NMAX,XX)
PROCEDURE PST12(N,ORDER,H,H1,H2,NPERT,COEF,RPOLY,MAXSUB,
               CONDN,XSUB,ILIST,NP,RCOEF,B,YSTAR,PERT,CODE)
INTEGER VALUE N,H1,H2,ORDER,MAXSUB,CODE
INTEGER H,NP,NPERT
INTEGER ARRAY ILIST
RATIONAL ARRAY VALUE XSUB
LONG ALGEBRAIC VALUE RPOLY
LONG ALGEBRAIC ARRAY VALUE COEF,CONDN
LONG ALGEBRAIC YSTAR,PERT
LONG ALGEBRAIC ARRAY(0:N) B
LONG ALGEBRAIC ARRAY RCOEF

# THIS PROCEDURE RETURNS THE PARAMETRIC TAU METHOD SOLUTION
# AND PERTURBATION OF A LINEAR ODE, USING DIRECT SERIES
# SUBSTITUTION.
# EXTERNAL VARIABLES REQUIRED: RR,XX,TPOWER,XPAR.
# SYSTEM PROCEDURES REQUIRED: PINTN,IMAX,TPS.
# DECLARATION IN CALLING PROCEDURE: NONE.

INTEGER I,J,IEND,DEGREE,IBEG,IEND
LONG ALGEBRAIC P,SUM,LCOEF,Y1
EXTERNAL LONG ALGEBRAIC XX,RR
EXTERNAL LONG ALGEBRAIC ARRAY TPOWER,XPAR
LONG ALGEBRAIC ALTRAN PINTN
LONG ALGEBRAIC ARRAY ALTRAN TPS

# DISTINGUISH BETWEEN DIFFERENTIATED AND INTEGRATED FORMS.

IF ( CODE<3 ) DO  # INTEGRATED FORM.
   RPOLY = PINTN(RPOLY,ORDER,XX)
   NPERT = N
   H = H1
   IBEG = ORDER
END
ELSE DO  # DIFFERENTIATED FORM.
   NPERT = N - ORDER
   H = ORDER + H1
   IBEG = 0
END

# INITIALIZE SET OF "SPECIAL" VALUES OF I.

IEND = H1 - I
ILIST = 0$0
DO I = IBEG,IEND
   ILIST = (ILIST,I)
END

# OBTAIN PARAMETRIC FORM OF PERTURBATION.

PERT = 0
DO I = 1,H
   PERT = PERT + XPAR(I)*TPOWER(NPERT+I)
END
RCOEF = TPS(RPOLY + PERT,XX,NPERT+H)
SET UP AND SOLVE UPPER TRIANGULAR SYSTEM DEFINING THE COEFFICIENTS OF THE SOLUTION.

NP = 0
LCOEF = COEF(-H1)
IBEG = H1
IEND = N + H1
DEGREE = N
YSTAR = 0

DO I = IEND, IBEG, -1
   P = LCOEF(RR=I)
   IF (P==0) DO       # SINGULARITY IN TRIANGULAR SYSTEM.
      NP = NP + 1
      ILIST = (ILIST, I)
      B(DEGREE) = XPAR(H+NP)
   END

ELSE DO
   SUM = 0
   JBEG = -H1 + 1
   JEND = IMIN(H2, N-1)
   DO J = JBEG, JEND
      IF ( COEF(J)<0 ) SUM = SUM + COEF(J)(RR=I)*(B(I+J))
   END
   B(DEGREE) = (RCOEF(I) - SUM)/P
   END

YSTAR = YSTAR*XX + B(DEGREE)
DEGREE = DEGREE - 1
END   # END OF PROCEDURE PST12.
PROCEDURE PST21(N,RPOLY,Q,H,V,V1,S1,R,YSTAR,PERT)
    INTEGER VALUE N,H,V1,S1,R
    LONG ALGEBRAIC VALUE RPOLY
    LONG ALGEBRAIC ARRAY VALUE Q,V
    LONG ALGEBRAIC YSTAR,PERT

    # THIS PROCEDURE RETURNS THE PARAMETRIC TAU METHOD SOLUTION
    # AND PERTURBATION OF A LINEAR ODE, USING CANONICAL
    # POLYNOMIALS.
    # EXTERNAL VARIABLES REQUIRED: XX, XPAR, TPOWER.
    # SYSTEM PROCEDURES REQUIRED: TPS.
    # DECLARATION IN CALLING PROCEDURE: NONE.

    INTEGER I,J,L,K
    LONG ALGEBRAIC ARRAY B
    EXTERNAL LONG ALGEBRAIC XX
    EXTERNAL LONG ALGEBRAIC ARRAY TPOWER, XPAR
    LONG ALGEBRAIC ARRAY ALTRAN TPS

    # COMPUTE THE PERTURBATION.
    K = N+H
    L = K -(R-V1)
    PERT = 0
    DO I = 1,R
       PERT = PERT + XPAR(I)*TPOWER(L+I)
    DOEND

    # OBTAIN PARAMETRIC SOLUTION.
    RPOLY = RPOLY + PERT
    B = TPS(RPOLY,XX,K)
    YSTAR = 0
    DO I = 0,K
       YSTAR = YSTAR + B(I)*Q(I)
    DOEND

    # ADD TO YSTAR THE TERMS, IF ANY, CONTRIBUTED BY POLYNOMIAL SOLNS.
    DO I = 1,V1
       YSTAR = YSTAR + V(I)* XPAR(R+I)
    DOEND

    RETURN
END  # END OF PROCEDURE PST21.
PROCEDURE VALCI1(C,N,ORDER,YSTAR,EQN,HALFN,NEQNS,
   MAXSUB,CONDN,XSUB,RHS,PARLIS,PARVAL)
INTEGER VALUE N,ORDER,MACSUB,HALFN,NEQNS
RATIONAL ARRAY VALUE XSUB
LONG ALGEBRAIC VALUE EQN,RHS,YSTAR
LONG ALGEBRAIC ARRAY VALUE CONDN,C

# THIS PROCEDURE ASSIGNS VALUES TO THE PARAMETERS IN THE
# PARAMETRIC CHEBYSHEV SERIES SOLUTION OF A GENERAL
# LINEAR ODE, USING BACKWARD RECURRENCE.
# EXTERNAL VARIABLES REQUIRED: RR,XX,XPAR,XCR,XT.
# PACKAGE PROCEDURES REQUIRED: IABS,LISTS.
# SYSTEM PROCEDURES REQUIRED: ASOLVE, GETBLK.
# DECLARATION IN CALLING PROCEDURE: NONE.

INTEGER I,J,M,II,NMI,L=IMAX(1,NEQNS)
LONG ALGEBRAIC YSTAR,T,P,TEMP
LONG ALGEBRAIC ARRAY(-HALFN:HALFN) COEF
LONG ALGEBRAIC ARRAY(1:L) B=0,ROW=0
LONG ALGEBRAIC ARRAY(1:L,1:L) MAT=0
LONG ALGEBRAIC ARRAY PARLIS,PARVAL
EXTERNAL LONG ALGEBRAIC XX,RR
EXTERNAL LONG ALGEBRAIC ARRAY XPAR,XCR,XT
INTEGER ALTRAN IABS
LONG ALGEBRAIC ARRAY ALTRAN ASOLVE

# SET UP SUBSTITUTION LISTS FOR ALL Y AND DERIVATIVE VALUES
# APPEARING IN THE ASSOCIATED CONDITIONS.

LISTS(YSTAR,ORDER,CONDN,MACSUB,XSUB,PARLIS,PARVAL,XX)

# SET UP THE FIRST "ORDER" ROWS OF THE LINEAR SYSTEM
# DEFINING THE CHEBYSHEV COEFFICIENTS.

DO I = 1,ORDER
   ROW(I) = CONDN(I)(PARLIS=PARVAL)
DOEND

# SET UP THE REMAINING ROWS OF THE LINEAR SYSTEM.

DO I = -HALFN,HALFN
   COEF(I) = GETBLK(EQN,XCR(I),1)
DOEND

DO I = ORDER,NEQNS-1
   T = 0
   DO J = -HALFN,HALFN
      JJ = IABS(I+J)
      IF (JJ<=N) T = T + COEF(J)(RR=1)*C(JJ)
   DOEND
   P = GETBLK(RHS,XT(I),1)
   IF (I==0) P = P*2
   ROW(I+1) = T - P
DOEND
# SOLVE THE SYSTEM.

PARLIS = 0$0
DO I = 1,NEQNS
   PARLIS = (PARLIS,XPAR(I))
   P = ROW(I)
   DO M = 1,NEQNS
      T = GETBLK(P,XPAR(M),I)
      MAT(I,M) = T
      P = P - T*XPAR(M)
   DOEND
   B(I) = - P
DOEND

PARVAL = ASOLVE(MAT,B,SING)
RETURN

SING:
IF (MAT==0 .AND. B ==0) DO
   PARVAL = NEQNS$I$
   RETURN
DOEND
ELSE DO
   WRITE " *** ERROR 5031 *** "
   WRITE " UNABLE TO ASSIGN VALUES TO PARAMETERS IN SOLUTION"
   FRETURN
DOEND

END # END OF PROCEDURE VALC11.
PROCEDURE VALC12(N,ORDER,YSTAR,EQN,HALF,N,MAXSUB,CONDN,XSUB,RHS,PRLIS,PARVAL)
INTEGER VALUE N,ORDER,MAXSUB,HALF
RATIONAL ARRAY VALUE XSUB
LONG ALGEBRAIC VALUE EQN,RHS,YSTAR
LONG ALGEBRAIC ARRAY PRLIS,PARVAL
LONG ALGEBRAIC ARRAY VALUE CONDN

# THIS PROCEDURE ASSIGN VALUES TO THE PARAMETERS IN THE
# PARAMETRIC CHEBYSHEV SERIES SOLUTION OF A GENERAL
# LINEAR ODE, USING GAUSSIAN ELIMINATION.
# EXTERNAL VARIABLES REQUIRED: RR,XX, XPAR,XCR,XT.
# PACKAGE PROCEDURES REQUIRED: IABS,LISTS.
# SYSTEM PROCEDURES REQUIRED: ASOLVE, GETBLK.
# DECLARATION IN CALLING PROCEDURE: NONE.

INTEGER I,J,M,IJ,NMI
LONG ALGEBRAIC YSTAR,T,P,TEMP
LONG ALGEBRAIC ARRAY(-HALF:HALF) COEF
LONG ALGEBRAIC ARRAY(0:N) B=0
LONG ALGEBRAIC ARRAY(0:N,0:N) MAT=0
LONG ALGEBRAIC ARRAY PRLIS,PARVAL
EXTERNAL LONG ALGEBRAIC XX,RR
EXTERNAL LONG ALGEBRAIC ARRAY XPAR,XCR,XT
INTEGER IABS
LONG ALGEBRAIC ARRAY ALTRAN ASOLVE

# SET UP THE FIRST "ORDER" ROWS OF THE LINEAR SYSTEM
# DEFINING THE CHEBYSHEV COEFFICIENTS.

DO I = 1,ORDER
LISTS(YSTAR,ORDER,CONDN,MAXSUB,XSUB,PRLIS,PARVAL,XX)
   P = CONDN(I)(PRLIS=PARVAL)
   DO M = 0,N
      T = GETBLK(P,XPAR(M),1)
      MAT(I-1,M) = T
      P = P - T*XPAR(M)
   DOEND
   B(I-1) = -P
DOEND

# SET UP THE REMAINING ROWS OF THE LINEAR SYSTEM.

DO I = -HALF,HALF
   COEF(I) = GETBLK(EQN,XCR(I),1)
DOEND

PARLIS = 0$0
DO I = 0,N
   PARLIS = (PARLIS,XPAR(I))
DOEND
DO I = ORDER,N
  DO J = -HALFN, HALFN
    JJ = JABS(I+J)
    IF (JJ <= N) DO
      P = COEF(J)(RR=I)
      MAT(I, JJ) = MAT(I, JJ) + P
    ENDDO
  ENDDO
  P = GETBLK(RHS, XT(I), 1)
  IF (I == 0) P = P*2
  B(I) = P
ENDDO

# SOLVE THE SYSTEM.

PARVAL = ASOLVE(MAT, B, SING)
RETURN

SING:
  IF (MAT == 0 .AND. B == 0) DO
    PARVAL = NS(1)
    RETURN
  ENDDO
  ELSE DO
    WRITE "*** ERROR 5031 ***"
    WRITE "UNABLE TO ASSIGN VALUES TO PARAMETERS IN SOLUTION"
    FRETURN
  ENDDO

END # END OF PROCEDURE VALC12.
PROCEDURE VALT12(YSTAR,PERT,B,N,ORDER,CODE,H,NP,COEF,RCOEF,
MAXSUB,CONDN,XSUB,ILIST,H1,H2,PARLIS,PARVAL)
INTEGER VALUE H,H1,H2,ORDER,MAXSUB,N,NP
INTEGER ARRAY VALUE ILIST,CODE.
RATIONAL ARRAY VALUE XSUB
LONG ALGEBRAIC ARRAY VALUE COEF,CONDN,B,RCOEF
LONG ALGEBRAIC ARRAY PARLIS,PARVAL

# THIS PROCEDURE ASSIGNS VALUES TO THE PARAMETERS IN
# THE PARAMETRIC TAU METHOD SOLUTION OF A GENERAL LINEAR
# ODE, USING SERIES SUBSTITUTION.
# EXTERNAL VARIABLES REQUIRED: RR,XXX, XPAR.
# PACKAGE PROCEDURE REQUIRED: LISTS.
# SYSTEM PROCEDURE REQUIRED: GETBLK, ASOLVE, IMIN.
# DECLARATION IN CALLING PROCEDURE: NONE.

INTEGER I,J,K,L,N,M,NM1,MH=1MAX(1,H+NP),JBEG,JEND
LONG ALGEBRAIC SUM,T,TEMP
EXTERNAL LONG ALGEBRAIC RR,XX
EXTERNAL LONG ALGEBRAIC ARRAY XPAR
LONG ALGEBRAIC ARRAY (1:MH) C,ROW
LONG ALGEBRAIC ARRAY (1:MH,1:MH) MAT = 0
LONG ALGEBRAIC ARRAY ALTRAN ASOLVE

# SET UP SUBSTITUTION LISTS FOR ALL Y AND DERIVATIVE VALUES
# APPEARING IN THE ASSOCIATED CONDITIONS.

IF (CODE == 2 .OR. CODE==4) DO  # PERTURBED CONDITIONS.
    YSTAR = YSTAR - PERT
DOEND
LISTS(YSTAR,ORDER,CONDN,MAXSUB,XSUB,PARLIS,PARVAL,XX)

# SET UP THE FIRST "ORDER" ROWS OF THE
# LINEAR SYSTEM DEFINING THE PARAMETERS,
# USING THE ASSOCIATED CONDITIONS.

DO I = 1,ORDER
    ROW(I) = CONDN(I)(PARLIS=PARVAL)
DOEND

# SET UP THE REMAINING (IF ANY) ROWS OF THE LINEAR SYSTEM USING
# THE SPECIAL CASES OF THE RECURRENCE EQUATION.

L = H + NP - ORDER
DO M = 1,L
    I = ILIST(M)
    SUM = 0
    JBEG = -IMIN(H1,1)
    JEND = IMIN(H2,N-1)
    DO J = JBEG,JEND
        IF(COEF(J)<0) SUM = SUM + B(I+J)*COEF(J)(RR=1)
    DOEND
    ROW(ORDER+M) = SUM - RCOEF(I)
DOEND
# SET UP AND SOLVE COEFFICIENT MATRIX AND RIGHT HAND SIDE.

PARLIS = 0$0$
DO K = 1,H+NP
   PARLIS = (PARLIS,XPAR(K))
   SUM = ROW(K)
   DO J = 1,H+NP
      T = GETBLK(SUM,XPAR(J),1)
      MAT(K,J) = T
      SUM = SUM - T*XPAR(J)
   DOEND
   C(K) = -SUM
DOEND

PARVAL = ASOLVE(MAT,C,SING)

RETURN

# CHECK FOR SINGULARITY IN MAT.

SING:
IF (MAT == 0 .AND. C == 0) DO
   PARVAL = (H+NP)$1$
   RETURN
DOEND
ELSE DO
   WRITE " *** ERROR 5031 *** "
   WRITE " UNABLE TO ASSIGN VALUES TO PARAMETERS IN SOLUTION"
   RETURN
DOEND

END # END OF PROCEDURE VALT12.
PROCEDURE VALT21(YSTAR,PERT,CODE,S,S1,ORDER,MAXSUB,
    CONDN,XSUB,N,PARLIS,PARVAL)
INTEGER VALUE N,ORDER,MAXSUB,S1,CODE
RATIONAL ARRAY VALUE XSUB
INTEGER ARRAY VALUE S
LONG ALGEBRAIC VALUE YSTAR,PERT
LONG ALGEBRAIC ARRAY CONDN
LONG ALGEBRAIC ARRAY PARLIS,PARVAL

# THIS PROCEDURE ASSIGNs VALUES TO THE PARAMETERS IN
# THE PARAMETRIC TAU METHOD SOLUTION OF A GENERAL LINEAR
# ODE, USING CANONICAL POLYNOMIALS.
# EXTERNAL VARIABLES REQUIRED: XUQ,XXX,XPAR.
# PACKAGE PROCEDURE REQUIRED: LISTS.
# SYSTEM PROCEDURE REQUIRED: GETBLK,ASOLVE.
# DECLARATION IN CALLING PROCEDURE: NONE.

INTEGER I,J,K,NPAR=S1+ORDER,M2 = IMAX(1,NPAR)
LONG ALGEBRAIC SUM,T,TEMP,G
LONG ALGEBRAIC ARRAY(I:M2) C,ROW
LONG ALGEBRAIC ARRAY(I:M2,1:M2) MAT
EXTERNAL LONG ALGEBRAIC XX
EXTERNAL LONG ALGEBRAIC ARRAY XUQ,XPAR
LONG ALGEBRAIC ARRAY ALTRAN ASOLVE

IF (NPAR<1) RETURN

# SET UP S1 ROWS OF THE LINEAR SYSTEM DEFINING THE PARAMETERS
# USING THE COMPONENTS THE UNDEFINED CANONICAL POLYNOMIALS.

DO I =1,S1
    T = XUQ(S(I))
    G = GETBLK(YSTAR,T,1)
    YSTAR = YSTAR - T*G
    ROW(I) = G
END

# SET UP THE REMAINING "ORDER" ROWS OF THE LINEAR SYSTEM
# USING THE ASSOCIATED CONDITIONS.

IF (CODE==2 .OR. CODE==4) YSTAR = YSTAR - PERT
LISTS(YSTAR,ORDER,CONDN,MAXSUB,XSUB,PARLIS,PARVAL,XX)

DO I = 1,ORDER
    ROW(S1+I) = CONDN(I)(PARLIS=PARVAL)
END
# ASSIGN COEFFICIENT MATRIX OF LINEAR SYSTEM TO MAT AND THE 
# RIGHT-HAND SIDE TO C. SOLVE THE LINEAR SYSTEM.

PARLIS = 050
DO I = 1,NPAR
  PARLIS = (PARLIS,XPAR(I))
  SUM = ROW(I)
  DO J = 1,NPAR
    T = GETBLK(SUM,XPAR(J),1)
    MAT(I,J) = T
    SUM = SUM - T*XPAR(J)
  DOEND
  C(I) = -SUM
DOEND
PARVAL = ASOLVE(MAT,C,SING)
RETURN

# CHECK FOR SINGULARITY IN MAT.

SING:
IF (MAT==0 .AND. C==0) DO
  PARVAL = NPAR$(I)
  RETURN
DOEND
ELSE DO
  WRITE " *** ERROR 5031 *** "
  WRITE " UNABLE TO ASSIGN VALUES TO PARAMETERS IN SOLUTION"
  FRETURN
DOEND
END # END OF PROCEDURE VALT21.
## List of Level 5 Procedures

<table>
<thead>
<tr>
<th>Procedure</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>AROUND</td>
<td>61</td>
</tr>
<tr>
<td>ASHRND</td>
<td>62</td>
</tr>
<tr>
<td>BAKREC</td>
<td>63</td>
</tr>
<tr>
<td>CANONP</td>
<td>66</td>
</tr>
<tr>
<td>ERREST</td>
<td>67</td>
</tr>
<tr>
<td>EVALG</td>
<td>68</td>
</tr>
<tr>
<td>GENTQ</td>
<td>69</td>
</tr>
<tr>
<td>INIT</td>
<td>71</td>
</tr>
<tr>
<td>INPCON</td>
<td>73</td>
</tr>
<tr>
<td>INPODE</td>
<td>74</td>
</tr>
<tr>
<td>INT</td>
<td>75</td>
</tr>
<tr>
<td>INTEGR</td>
<td>76</td>
</tr>
<tr>
<td>LISTS</td>
<td>77</td>
</tr>
<tr>
<td>NEXTIT</td>
<td>78</td>
</tr>
<tr>
<td>OUTIT</td>
<td>79</td>
</tr>
<tr>
<td>PRODTR</td>
<td>80</td>
</tr>
<tr>
<td>PRODXR</td>
<td>81</td>
</tr>
<tr>
<td>PKINIT</td>
<td>82</td>
</tr>
<tr>
<td>PROGL</td>
<td>84</td>
</tr>
<tr>
<td>QGROUND</td>
<td>85</td>
</tr>
<tr>
<td>READER</td>
<td>86</td>
</tr>
<tr>
<td>REDUZS</td>
<td>87</td>
</tr>
<tr>
<td>SETP1</td>
<td>89</td>
</tr>
<tr>
<td>SETP2</td>
<td>90</td>
</tr>
<tr>
<td>Procedure</td>
<td>Page</td>
</tr>
<tr>
<td>-----------</td>
<td>------</td>
</tr>
<tr>
<td>AROUND</td>
<td>61</td>
</tr>
<tr>
<td>ASHRND</td>
<td>62</td>
</tr>
<tr>
<td>BAKREC</td>
<td>63</td>
</tr>
<tr>
<td>CANNONP</td>
<td>66</td>
</tr>
<tr>
<td>CERREST</td>
<td>67</td>
</tr>
<tr>
<td>EUHLC</td>
<td>68</td>
</tr>
<tr>
<td>GENTO</td>
<td>69</td>
</tr>
<tr>
<td>JN11</td>
<td>71</td>
</tr>
<tr>
<td>JNPCON</td>
<td>72</td>
</tr>
<tr>
<td>JNPORD</td>
<td>73</td>
</tr>
<tr>
<td>JNT1</td>
<td>74</td>
</tr>
<tr>
<td>INTEGR</td>
<td>75</td>
</tr>
<tr>
<td>LISTS</td>
<td>76</td>
</tr>
<tr>
<td>NEXTIT</td>
<td>77</td>
</tr>
<tr>
<td>OUT11</td>
<td>78</td>
</tr>
<tr>
<td>PRODTR</td>
<td>79</td>
</tr>
<tr>
<td>PRODXR</td>
<td>80</td>
</tr>
<tr>
<td>PK2011</td>
<td>81</td>
</tr>
<tr>
<td>PROGL</td>
<td>82</td>
</tr>
<tr>
<td>GROUND</td>
<td>83</td>
</tr>
<tr>
<td>LEADER</td>
<td>84</td>
</tr>
<tr>
<td>REJUES</td>
<td>85</td>
</tr>
<tr>
<td>SETHX</td>
<td>86</td>
</tr>
<tr>
<td>SEIUH</td>
<td>87</td>
</tr>
<tr>
<td>SHF11</td>
<td>88</td>
</tr>
</tbody>
</table>
PROCEDURE AROUND(P,N,M,X)
INTEGER VALUE N,M
LONG ALGEBRAIC VALUE P,X

# THIS PROCEDURE RETURNS THE ROUNDED FORM OF A POLYNOMIAL P
# OF GIVEN DEGREE N IN THE INDEPENDENT VARIABLE X.
# THE ROUNDED FORM CONTAINS INTEGERS AT MOST M DIGITS LONG.
# THE PROCEDURE ASSUMES THAT THE COEFFICIENTS OF THE POWERS
# OF X IN P ARE RATIONAL NUMBERS.
# PACKAGE PROCEDURES REQUIRED: QROUND.
# SYSTEM PROCEDURES REQUIRED: TPS,TPSEVL,ANUM.
# DECLARATION IN CALLING PROCEDURE: LONG ALGEBRAIC ALTRAN AROUND.

LONG RATIONAL DEN
LONG ALGEBRAIC NEWP,NUM
LONG RATIONAL ARRAY Q
LONG RATIONAL ARRAY ALTRAN QROUND
LONG ALGEBRAIC ALTRAN TPSEVL
LONG ALGEBRAIC ARRAY ALTRAN TPS

NUM = ANUM(P,DEN)
IF (DEN <= 10**M) RETURN(P)

Q = TPS(P,X,N)
Q = QROUND(Q,N,M)
NEWP = TPSEVL(Q,X,N)

RETURN(NEWP)
END  # END OF PROCEDURE AROUND.
PROCEDURE ASHRND(P,N,CODE,X,MAXERR,XIND)
INTEGER VALUE N, CODE.
REAL VALUE MAXERR
LONG ALGEBRAIC VALUE P,X,XIND

# THIS PROCEDURE RETURNS THE SHORTENED AND/OR ROUNDED FORM
# (DEPENDING ON THE VALUE OF CODE) OF A POLYNOMIAL P OF
# GIVEN DEGREE N IN THE INDEPENDENT VARIABLE X,
# AND CONTAINING ONE INDETERMINATE XIND. THE ROUNING ERROR
# NOTE THAT XIND MAY BE OMITTED IF NO SHORTENING IS REQUIRED.
# PACKAGE PROCEDURES REQUIRED: AROUND, TRUNC, PREC.
# SYSTEM PROCEDURES REQUIRED: TPS, TPSEVL, NTRM, DEG, ANUM.
# DECLARATION IN CALLING PROCEDURE: LONG ALGEBRAIC ALTRAN ASHRND.

INTEGER I, NTERMS, M, L, DEGN, DEGD
LOGICAL ROUND=.FALSE., SHORT=.FALSE.
LONG ALGEBRAIC NUM, DEN, RECDEN, NEWP
LONG ALGEBRAIC ARRAY ANEW
INTEGER ALTRAN PREC
LONG ALGEBRAIC ALTRAN TRUNC, AROUND, TPSEVL
LONG ALGEBRAIC ARRAY ALTRAN TPS

# ANALYZE CODE.
IF (CODE==1) RETURN(P)
IF (CODE==2 .OR. CODE==4) ROUND=.TRUE.
IF (CODE==3 .OR. CODE==4) SHORT=.TRUE.

# SET UP FOR SHORTENING AND/OR ROUNING.
NUM = ANUM(P,DEN)
DEGN = DEG(NUM,DEN)
DEGD = DEG(DEN,XIND)
NEWP = P

# PERFORM SHORTENING AND/OR ROUNING.

IF (SHORT) DO  # SHORTEN.
   IF (DEGD==0) RECDEN = 1/DEN
   ELSE RECDEN = TPSEVL(TPS(1/DEN,XIND,DEGD),XIND,DEGD)
   NEWP = TRUNC(NUM*RECDEN,N,XIND)
   DEGD = 0
END DO

IF (ROUND) DO  # ROUND.
   IF (DEGD<>0) DO
      WRITE " *** ERROR 5032 ***"
      WRITE " WRONG SOLUTION CODE SPECIFIED FOR PROBLEM"
      RETURN
   END DO
   NTERMS = NTRM(NEWP)
   M = PREC(MAXERR/NTERMS)
   IF ( (DEGN+DEGD) == 0 ) NEWP = AROUND(NEWP,N,M,X)
   ELSE DO
      ANEW = TPS(NEWP,X,N)
      DO I = 0, N
         ANEW(I) = AROUND(ANEW(I),N,M,XIND)
      END DO
      NEWP = TPSEVL(ANEW,X,N)
   END DO
END DO

RETURN(NEWP)
PROCEDURE BAKREC(EQN, N, ZK, K, RLIST, KMIN, KMAX, PAR, NPAR)
INTEGER    VALUE N, KMIN, KMAX, NPAR
LONG ALGEBRAIC VALUE EQN, K
LONG ALGEBRAIC ARRAY VALUE RLIST, PAR
LONG ALGEBRAIC ARRAY (0:N) VALUE ZK

# PROCEDURE TO SOLVE A RECURRENCE EQUATION FOR AN NPAR-
# PARAMETER SOLUTION.

# INPUT PARAMETERS:
# EQN -- THE LEFT SIDE OF THE RECURRENCE EQUATION;
# N -- INDICATES THAT EQN CONTAINS N+1 TERMS;
# ZK -- ARRAY OF THE N+1 INDETERMINATES APPEARING IN EQN
#    REPRESENTING THE DEPENDENT VARIABLE Z -- I.E. EQN IS
#    OF THE FORM
#        U0 * ZK(0) + U1 * ZK(1) + ... + UN * ZK(N)
#    WHERE THE UI ARE RATIONAL EXPRESSIONS IN THE INDE-
#    TERMINATE K AND ZK(I) REPRESENTS Z(K+1);
# K -- THE NAME OF THE INDETERMINATE IN EQN;
# RLIST -- LIST OF RIGHT-HAND SIDES OF THE RECURRENCE EQU-
#   ATION CORRESPONDING TO THE CASES K = KMIN, ..., KMAX
#    OF EQN( THE RIGHT-HAND SIDE IS ZERO BEYOND THE NUMBER
#    OF ELEMENTS IN RLIST);
# KMIN, KMAX -- INDICATE THE RANGE OF SUBSCRIPTS FOR WHICH
#    THE SOLUTION IS TO BE COMPUTED -- I.E. THE SOLUTION
#    Z(KMIN), ..., Z(KMAX) IS DESIRED;
# PAR -- NAMES OF THE PARAMETERS TO BE USED;
# NPAR -- NUMBER OF PARAMETERS DESIRED IN THE SOLUTION.

# OUTPUT:
# THE VALUE RETURNED IS AN ARRAY DIMENSIONED FROM 0 TO
# KMAX CONTAINING THE DESIRED NPAR-PARAMETER SOLUTION. IF
# KMIN > 0 THEN THE FIRST KMIN ELEMENTS OF THE ARRAY ARE
# ARBITRARILY SET TO ZERO.

# ASSUMPTIONS:
# IT IS ASSUMED THAT KMIN >= 0, KMAX >= KMIN+NPAR, AND
# THAT KMAX >= KMIN+LENGTH-1 WHERE LENGTH IS THE LENGTH OF
# THE ARRAY RLIST.

# THE FOLLOWING DECLARATION MUST APPEAR IN THE CALLING
# PROCEDURE:
# LONG ALGEBRAIC ARRAY ALTRAN BAKREC .

INTEGER I, J, NZEROS, KVAL, KLAST, LENGTH, L
LONG ALGEBRAIC U0, ZK0
LONG ALGEBRAIC ARRAY UNLIST
LONG ALGEBRAIC ZKVAL
LONG ALGEBRAIC ARRAY GENSOLO, SOLN, VALIST
LONG ALGEBRAIC ARRAY (0:KMAX) Z
# SOLVE RECURRENCE FOR Z(K).

U0 = GETBLK(EQN, ZK(0), 1)
ZK0 = ZK(0) - EQN / U0

# 'UNLIST' IS THE LIST OF UNKNOWNS IN ZK0.

UNLIST = 1S( ZK(1) )
DO L = 2, N
   UNLIST = (UNLIST, ZK(L))
DOEND

# COMPUTE NPAR INDEPENDENT SOLUTIONS AND FORM THEIR LINEAR
# COMBINATION INTO GENSOL.

GENSOL = 0
DO I = 1, NPAR
   SOLN = 1S(I)
   NZEROS = N

   DO KVAL = KMAX-1, KMIN, -1
      NZEROS = NZEROS - 1
      IF (NZEROS >= 0) VALIST = (SOLN, NZEROS$0)
      ELSE DO
         VALIST = 1S( SOLN(I) )
         DO J = 2, N
            VALIST = (VALIST, SOLN(J))
         DOEND
      DOEND # END OF ELSE-CLAUSE.

   ZKVAL = ZK0(K = KVAL)(UNLIST = VALIST)
   SOLN = (ZKVAL, SOLN)

DOEND

GENSOL = GENSOL + PAR(I) * (SOLN, (I-1)$0)
DOEND
# COMPUTE A PARTICULAR SOLUTION AND ADD IT INTO GEN_SOL.

LENGTH = DBINFO(RLIST)(0,1)

IF (LENGTH > 0) DO

   KLAST = KMIN + LENGTH - 1
   SOLN = IS( RLIST(LENGTH) / U0(K=KLAST) )
   NZEROS = N: L = LENGTH

   DO KVAL = KLAST-1, KMIN, -1

      NZEROS = NZEROS - 1
      IF (NZEROS >= 0) VALIST = (SOLN, NZEROS$0)
      ELSE DO
         VALIST = IS( SOLN(1) )
         DO J = 2, N
            VALIST = (VALIST, SOLN(J))
         ENDDO
      ENDDO  # END OF ELSE-CLAUSE.

   ZKVAL = ZK0(K = KVAL)(UNLIST = VALIST)
   L = L - 1
   ZKVAL = ZKVAL + RLIST(L) / U0(K=KVAL)
   SOLN = (ZKVAL, SOLN)

   ENDDO

   GEN_SOL = GEN_SOL + (SOLN, (KMAX-KLAST)$0)

   ENDDO  # END OF IF-STATEMENT.

# RETURN 'GEN_SOL' IN POSITIONS KMIN TO KMAX OF ARRAY Z,
# WHICH IS DIMENSIONED FROM 0 TO KMAX.

Z = IS(KMIN$0, GEN_SOL)

RETURN( Z )

END  # END OF PROCEDURE BAKREC.
PROCEDURE CANONP(N,POLY,ORDER,CODE,H1,H2,EQN,GENPOL,Q,H,S,S1,V,V1)
INTEGER VALUE N,ORDER,CODE,H1,H2
LONG ALGEBRAIC VALUE EQN,GENPOL
INTEGER H=H2,S1,V1
INTEGER ARRAY S
LONG ALGEBRAIC ARRAY POLY,Q,V

# THIS PROCEDURE COMPUTES THE CANONICAL POLYNOMIALS ASSOCIATED WITH
# A LINEAR OPERATOR WITH POLYNOMIAL COEFFICIENTS.
# INPUT PARAMETERS:
# N  -- MAXIMUM DEGREE (IN THE INDEPENDENT VARIABLE X) OF
# THE CANONICAL POLYNOMIALS;
# ORDER -- ORDER OF LINEAR OPERATOR;
# POLY  -- ARRAY POLY(0),...,POLY(ORDER) OF POLYNOMIAL COEFFICIENTS
# OF LINEAR OPERATOR;
# CODE  -- SOLUTION CODE INDICATING THE FORM (INTEGRATED OR
# DIFFERENTIATED) IN WHICH THE OPERATOR IS APPLIED;
# EQN  -- THE RECURSION EQUATION DEFINING THE CANONICAL
# POLYNOMIALS;
# GENPOL -- THE GENERATING POLYNOMIAL ASSOCIATED WITH THE
# LINEAR OPERATOR, IN THE FORM
# V(H1)*XXR(H1) + ... + V(H2)*XXR(H2),
# WHERE R IS AN INDETERMINATE REPRESENTED AS RR, XXR(I)
# REPRESENTS X**(R+J), AND V(I) ARE RATIONAL EXPRESSIONS IN R.
# OUTPUT PARAMETERS:
# S  -- MINIMAL SET OF INDICES OF UNDEFINED CANONICAL POLYNOMIALS;
# V  -- SET OF POLYNOMIAL SOLUTIONS OF THE HOMOGENEOUS FORM
# OF THE LINEAR OPERATOR;
# S1,V1 -- CARDINALITY OF S AND V RESPECTIVELY;
# Q  -- ARRAY Q(0),...,Q(N+H) OF CANONICAL POLYNOMIALS;
# H  -- THE "HEIGHT" OF THE LINEAR OPERATOR.
# EXTERNAL VARIABLES REQUIRED: MSGVL,XX,RR,XXR.
# PACKAGE PROCEDURES REQUIRED: RECT21,GENT,EVALG,REDUZS.
# DECLARATION IN CALLING PROCEDURE: NONE.

INTEGER I,J,MM,Z1,G1
INTEGER ARRAY Z,G
LONG ALGEBRAIC XP
EXTERNAL INTEGER MSGVL
EXTERNAL LONG ALGEBRAIC XX,RR
EXTERNAL LONG ALGEBRAIC ARRAY XXR
LONG ALGEBRAIC ALTRAN RECT21
LONG ALGEBRAIC ARRAY ALTRAN GENTQ

IF(MSGVL>3) WRITE "ENTER CANONP"

# COMPUTE (TENTATIVE) Q AND THE SET Z OF THE INDICES OF
# CANONICAL POLYNOMIALS THAT CANNOT BE GENERATED FROM THE
# RECURSION EQUATION.
# REDUCE THE SET Z TO S, THE MINIMAL SET OF INDICES OF UNDEFINED
# CANONICAL POLYNOMIALS. OBTAIN POLYNOMIAL SOLUTIONS, IF ANY.

Q = GENTQ(N,H1,H2,SEQN,Z,Z1)
EVALG(N,H,GENPOL,G,G1)
REDUZS(Q,Z,Z1,G,G1,H,N,GENPOL,V,V1,S,S1)
IF(MSGVL>3) WRITE Q,Z,S,G,V,H,GENPOL,"EXIT CANONP"
RETURN
END # END OF PROCEDURE CANONP.
PROCEDURE ERREST(ORDER,PERT,POLY,CODE,MAXERR)
    INTEGER VALUE ORDER, CODE
    REAL MAXERR
    LONG ALGEBRAIC VALUE PERT
    LONG ALGEBRAIC ARRAY VALUE POLY

    # THIS PROCEDURE RETURNS AN ESTIMATE ERR, (TOGETHER
    # WITH ITS MAXIMUM VALUE, MAXERR) OF THE ERROR IN A TAU METHOD
    # APPROXIMATE SOLUTION OF A LINEAR ODE, USING THE PERTURBATION,
    # PERT, IMPOSED ON THE LINEAR ODE.
    # EXTERNAL VARIABLES REQUIRED: XX, MAXMU, XMU, MSG_LVL.
    # PACKAGE PROCEDURES REQUIRED: AMAX, TRUNC.
    # SYSTEM PROCEDURES REQUIRED: DEG, SR.
    # DECLARATION IN CALLING PROCEDURE: LONG ALGEBRAIC ALTRAN ERREST.

    INTEGER I, DEGREE, MM
    LONG ALGEBRAIC ERR, ERRSOL, P, Q
    EXTERNAL INTEGER MAXMU, MSG_LVL
    EXTERNAL LONG ALGEBRAIC ARRAY XMU
    EXTERNAL LONG ALGEBRAIC XX
    LONG ALGEBRAIC ALTRAN AMAX
    LONG ALGEBRAIC ALTRAN TRUNC

    IF (MSG_LVL > 3) WRITE " ENTER ERREST"

    DEGREE = DEG(PERT, XX)
    P = PERT
    Q = POLY(ORDER)
    ERR = P/Q
    MM = IMAX(1, MAXMU)
    P = P(XMU = MM$())
    IF (DEG(Q, XX) > 0) Q = TRUNC(Q, 0, XX)
    P = P/Q(XMU = MMS(0))
    ERRSOL = AMAX(P, DEGREE, XX)
    MAXERR = SR(ERRSOL)

    IF (MSG_LVL > 3) WRITE ERR, MAXERR, " EXIT ERREST"
    RETURN(ERR)

END # END OF PROCEDURE ERREST.
PROCEDURE EVALG(N,H,GENPOL,G,GI)

INTEGER VALUE N,H
LONG ALGEBRAIC VALUE GENPOL
INTEGER GI
INTEGER ARRAY G

# THIS PROCEDURE EVALUATES THE SET OF INADEQUATE INDICES G,
# ASSOCIATED WITH THE GENERATING POLYNOMIAL OF A LINEAR ODE,
# WITH HEIGHT H. GI IS THE CARDINALITY OF G.
# EXTERNAL VARIABLES REQUIRED: XX,RR,XXR.
# SYSTEM PROCEDURES REQUIRED: DEG.
# DECLARATION IN CALLING PROCEDURE: NONE.

INTEGER NPH,I,J
LONG ALGEBRAIC F,XI
LONG ALGEBRAIC ARRAY XLIST,VALIST
EXTERNAL LONG ALGEBRAIC XX,RR
EXTERNAL LONG ALGEBRAIC ARRAY XXR

NPH = N+H
DO I = 0,H
   XLIST = (XLIST,XXR(I))
DOEND

G = 0$0
GI = 0
DO I = 0,NPH
   XI = XX**I
   VALIST = 0$0
   DO J=0,H
      VALIST = (VALIST,XI)
      XI = XI*XX
   DOEND
   F = GENPOL(RR=I)(XLIST = VALIST)

   IF (DEG(F,XX) < I+H) DO
      G = (G,I)
      GI = GI + 1
   DOEND
DOEND

END # END OF PROCEDURE EVALG.
PROCEDURE GENTQ(N,H1,H2,EQN,Z,Z1)
   INTEGER VALUE N,H1,H2
   INTEGER Z1
   INTEGER ARRAY Z
   LONG ALGEBRAIC VALUE EQN

   # THIS PROCEDURE GENERATES THE TENTATIVE CANONICAL POLYNOMIALS, Q,
   # ASSOCIATED WITH A LINEAR OPERATOR, GIVEN THE RECURRENCE
   # EQUATION DEFINING THE CANONICAL POLYNOMIALS. THE PROCEDURE ALSO
   # THE SET Z OF INDICES OF THE CANONICAL POLYNOMIALS
   # THAT CANNOT BE GENERATED FROM THE RECURRENCE EQUATION.
   # Z1 IS THE CARDINALITY OF Z.
   # EXTERNAL VARIABLES REQUIRED: XX,RR,XXR,XQR,XUQ.
   # SYSTEM PROCEDURES REQUIRED: ANUM,GETBLK,IMAX.
   # DECLARATION IN CALLING PROCEDURE: LONG ALGEBRAIC ARRAY ALTRAN GENTQ.

   INTEGER I,HH=H1-H2,NH=N+H2,M1,J
   LONG ALGEBRAIC XP,DEN,D1,P
   LONG ALGEBRAIC ARRAY(0:NH) Q
   LONG ALGEBRAIC ARRAY(HH:0) COEF
   EXTERNAL LONG ALGEBRAIC XX,RR
   EXTERNAL LONG ALGEBRAIC ARRAY XQR,XUQ,XXR

   # SET UP THE COEFFICIENTS IN RECURRENCE EQUATION.

   EQN = ANUM(EQN,DEN)
   DO I = HH,0
      COEF(I) = GETBLK(EQN,XQR(I),1)
   DOEND
   XP = GETBLK(EQN,XXR(-H2),1)

   # INITIALIZE SET Z.

   Z=0$0
   DO I = 0,H2-1
      Q(I) = XUQ(I)
      Z = (Z,1)
   DOEND
   Z1 = H2
# COMPUTE THE REQUIRED CANONICAL POLYNOMIALS.

DO I = H2,NH
   DI = DEN(RR=I)
   IF (DI <> 0) DO # Q(I) DEFINED.
      Q(I) = XP(RR=I)
      M1 = IMAX(HH,-I)
      DO J = M1,-1
         P = COEF(J)(RR=I)
         Q(I) = Q(I) + P*Q(I+J)
      DOEND
      Q(I) = Q(I)/DI
   DOEND

ELSE DO # Q(I) UNDEFINED.
   Q(I) = XUQ(I)
   Z = (Z,I)
   Z! = Z! + 1
   DOEND
   XP = XP*XX
   DOEND

RETURN(Q)
END # END OF PROCEDURE GENTQ.
PROCEDURE INIT(ORDER,N,MAXSUB,XSUB,CONDN,X)
INTEGER VALUE ORDER,N,MAXSUB
RATIONAL ARRAY VALUE XSUB
LONG ALGEBRAIC VALUE X
LONG ALGEBRAIC ARRAY VALUE CONDN

# THIS PROCEDURE Computes an initial polynomial approximation,
# YINIT, satisfying the Aassociated conditions of an ODE.
# An error is reported if no initial can be computed with
# degree not exceeding N.
# external variables required: MSGlvl,XPAR.
# system procedures required: ASOLVE,TPSEVL,GETBLK,LISTS.
# declaration in calling procedure: long algebraic altran INIT.

INTEGER I,J,K,DEGP,ORDM1=ORDER-1
LONG ALGEBRAIC T,TEMP,XP,P,YINIT=0,P0,XI,XIM1
LONG ALGEBRAIC ARRAY (0:ORDM1) C=0
LONG ALGEBRAIC ARRAY (0:ORDM1,0:ORDM1) MAT=0
LONG ALGEBRAIC ARRAY PARLIS,UNLIST,VALIST
LONG ALGEBRAIC ALTRAN TPSEVL
LONG ALGEBRAIC ARRAY ALTRAN ASOLVE
EXTERNAL INTEGER MSGlvl
EXTERNAL LONG ALGEBRAIC ARRAY XPAR

IF(MSGlvl>3) WRITE "ENTER INIT"
P0 = TPSEVL(XPAR,X,ORDM1)
PARLIS = P0
DO I = 0,ORDM1
   PARLIS = (PARLIS,XPAR(I))
DOEND

P = P0
DEGP = ORDM1
XI = X
XIM1 = 1

AGAIN: LISTS(P,ORDER,CONDN,MAXSUB,XSUB,UNLIST,VALIST,X)
IF(MSGlvl>3) WRITE P,UNLIST,VALIST

# set up coefficient matrix of linear system defining
# the initial approximation, YINIT.

DO I = 1,ORDER
   XP = CONDN(I)(UNLIST=VALIST)
   TEMP = 0
   DO J = 0,ORDM1
      T = GETBLK(XP,XPAR(J),1)
      MAT(I-1,J) = T
      TEMP = TEMP + T*XPAR(J)
   DOEND
   C(I-1) = TEMP - XP
DOEND
IF(MSGlvl>3) WRITE MAT,C

# SOLVE LINEAR SYSTEM , IF POSSIBLE, FOR YINIT.

C = ASOLVE(MAT,C,SING)
IF (.NOT. (C==0) .OR. DEGP==N) DO
   YINIT = P(PARLIS = C)
   IF (MSGVL>3) WRITE YINIT, "EXIT INIT"
   RETURN(YINIT)
END

# TRY A NEW APPROXIMATION.

NEXTP:
P = P0*XI + XIM1
XIM1 = XIM1 + XI
XI = XI*X
GO TO AGAIN

# RESOLVE SINGULARITY HERE.

SING:
IF(MSGVL>3) WRITE "SINGULAR"
IF (DEGP > N) DO
   WRITE " ** ERROR 5031  *****"
   WRITE " UNABLE TO COMPUTE INITIAL APPROXIMATI0N "
   FRETURN
END

END # END OF PROCEDURE INIT.
PROCEDURE INPCON(ORDER, MAXSUB, XSUB, CONDN)
INTEGER VALUE ORDER
INTEGER MAXSUB
RATIONAL ARRAY XSUB
EXTERNAL INTEGER MAXORD, MAXMU, MAXVAL
LONG ALGEBRAIC(DY(0:MAXORD),Y,X,MU(1:IMAX(1, MAXMU)),
DYX(1:MAXORD, 1:MAXVAL), YX(1:MAXVAL)) CONDN, P
ARRAY CONDN

# THIS PROCEDURE READS IN THE CONDITIONS ASSOCIATED WITH
# AN ODE OF GIVEN ORDER. THE CONDITIONS ARE RETURNED IN
# CONDN(1),...,CONDN(ORDER), AND THE X-VALUES IN
# XSUB(1),...,XSUB(MAXSUB), WHERE MAXSUB IS THE NUMBER
# OF X-VALUES APPEARING IN THE CONDITIONS.
# THE PROCEDURE ASSUMES THAT ALL X-VALUES APPEARING IN THE
# CONDITIONS ARE ASSIGNED VALUES.
# EXTERNAL VARIABLES REQUIRED: MAXORD, MAXMU, MAXVAL.
# SYSTEM PROCEDURES REQUIRED: DEG.
# DECLARATION IN CALLING PROCEDURE: NONE.

INTEGER I,J,K,M
RATIONAL R
LONG ALGEBRAIC ARRAY(1:IMAX(1, ORDER)) CD=P

MAXSUB = 0
IF (ORDER<1) RETURN

# READ IN ASSOCIATED CONDITIONS.
DO M = 1, ORDER; READ CD(M); DOEND
CONDN = CD
DO M = 1, ORDER; WRITE CONDN(M); DOEND

# SET UP TO READ IN XVALUES APPEARING IN ASSOC. CONDITIONS.
DO J = MAXVAL, 1, -1
DO M = 1, ORDER
   IF (DEG(CONDN(M), YX(J))>0) GO TO DEF
   DO I = 1, ORDER-1
      IF (DEG(CONDN(M), DYX(I, J)) > 0) GO TO DEF
   DOEND
DEFEND
DOEND
DEF: MAXSUB = J

XSUB = 0$0
DO J = 1, MAXSUB
   READ R
   XSUB = (XSUB, R)
   WRITE XSUB(J)
DOEND

RETURN
END # END OF PROCEDURE INPCON.
PROCEDURE INPODE(ORDER,DIFFEQ,LINENAR)
INTEGER VALUE ORDER
LOGICAL LINEAR
EXTERNAL INTEGER MAXMU,MAXORD
LONG ALGEBRAIC(DY(0:MAXORD),Y,X,MU(1:IMAX(1,MAXMU))) DIFFEQ

# THIS PROCEDURE READS IN AN ODE OF GIVEN ORDER.
# THE ODE IS REPRESENTED AS DIFEQ=0.
# THE LOGICAL VARIABLE LINEAR IS SET TO .TRUE. IF
# THE ODE IS LINEAR AND TO .FALSE. IF OTHERWISE.
# EXTERNAL VARIABLES REQUIRED: MAXMU,MAXORD
# SYSTEM PROCEDURES REQUIRED: GETBLK,DEG.
# DECLARATION IN CALLING PROCEDURE: NONE.

INTEGER I,J,K,L,M
LONG ALGEBRAIC P,FCHECK

# READ IN ODE.

READ DIFEQ
WRITE DIFEQ

# CHECK IF ODE IS LINEAR.

LINEAR = .FALSE.
FCHECK = DIFEQ(Y=DY(0))
DO I = 0,ORDER
   L = DEG(FCHECK,DY(I))
   IF (L==0) GO TO ON1
   IF (L > 1) GO TO ON2
   P = GETBLK(FCHECK,DY(I),1)
   DO J = I+1,ORDER
      IF (DEG(P,DY(I)) <> 0) GO TO ON2
   ENDDO
ON1: DOEND
   LINEAR = .TRUE.
ON2: RETURN
END # END OF PROCEDURE INPODE.
PROCEDURE INT(P, DEGREE)
INTEGER VALUE DEGREE
LONG ALGEBRAIC VALUE P

# THIS PROCEDURE INTEGRATES A POLYNOMIAL P WHICH IS ASSUMED
# TO BE IN POWER SERIES FORM
# V(0)*XXR(0) + ..... + V(H)*XXR(H)
# WHERE H=DEGREE, R IS AN INDETERMINATE REPRESENTED AS
# RR, XXR(J) REPRESENTS X**(R+J), AND V(I) ARE RATIONAL
# EXPRESSIONS IN R.
# EXTERNAL VARIABLES REQUIRED: RR, XXR.
# SYSTEM PROCEDURES REQUIRED: GETBLK.
# DECLARATION IN CALLING PROCEDURE: LONG ALGEBRAIC ALTRAN INT.

INTEGER J
LONG ALGEBRAIC NEWP
EXTERNAL LONG ALGEBRAIC RR
EXTERNAL LONG ALGEBRAIC ARRAY XXR

NEWP = 0
DO J = 0, DEGREE
    NEWP = NEWP + GETBLK(P, XXR(J), 1)*XXR(J+1)/(RR+J+1)
END DO

RETURN(NEWP)

END # END OF PROCEDURE INT.
PROCEDURE INTEGR(P,HALFN)
    INTEGER VALUE HALFN
    LONG ALGEBRAIC VALUE P

# THIS PROCEDURE INTEGRATES A POLYNOMIAL P WHICH IS ASSUMED
# TO BE IN THE CHEBYSHEV FORM
#   V(0)*XTR(-H) + .... + V(H)*XTR(H)
# WHERE H=HALFN, R IS AN INDETERMINATE REPRESENTED AS RR,
# XTR(J) REPRESENTS THE (R+J)-TH CHEBYSHEV POLYNOMIAL,
# AND V(I) ARE RATIONAL EXPRESSIONS IN R.
# EXTERNAL VARIABLES REQUIRED: RR,XTR.
# SYSTEM PROCEDURES REQUIRED: GETBLK.
# DECLARATION IN CALLING PROCEDURE: LONG ALGEBRAIC ALTRAN INTEGR.

INTEGER J
LONG ALGEBRAIC NEWP
EXTERNAL LONG ALGEBRAIC RR
EXTERNAL LONG ALGEBRAIC ARRAY XTR

NEWP = 0
DO J = -HALFN,HALFN
    NEWP = NEWP + GETBLK(P,XTR(J),1)* (XTR(J+1)/
                       (RR+J+1) - XTR(J-1)/(RR+J-1))/2
END DO

RETURN(NEWP)

END # END OF PROCEDURE INTEGR.
PROCEDURE NEXTIT(N,NMAX,ORDER,YNEW,YCORR,POLY,PERT,CODE,
   DIFFEQ,RHS)
   INTEGER VALUE ORDER, CODE, NMAX
   INTEGER N
   LONG ALGEBRAIC RHS, YNEW, PERT
   LONG ALGEBRAIC ARRAY POLY
   LONG ALGEBRAIC VALUE YCORR, DIFFEQ

   # THIS PROCEDURE SETS UP FOR THE NEXT STAGE IN THE
   # (NEWTON) ITERATIVE SOLUTION OF AN ODE.
   # EXTERNAL VARIABLES REQUIRED: XX, XY, MASGLVL, XMU, XDY.
   # PACKAGE PROCEDURES REQUIRED: ERREST, ASHRND, SETP2.
   # SYSTEM PROCEDURES REQUIRED: IMIN, DIFF.
   # DECLARATION IN CALLING PROCEDURE: NONE.

   INTEGER I, M, M1, M2, MAXN, DEGREE
   REAL MAXERR
   LONG ALGEBRAIC ERR, P
   LONG ALGEBRAIC ARRAY UNLIST, VALIST
   EXTERNAL INTEGER MASGLVL
   EXTERNAL LONG ALGEBRAIC XX, XY
   EXTERNAL LONG ALGEBRAIC ARRAY XMU, XDY
   LONG ALGEBRAIC ALTRAN ERREST, ASHRND
   LONG ALGEBRAIC ARRAY ALTRAN SETP2

   YNEW = YNEW + YCORR
   IF (CODE > 1 .AND. YCORR<>0) DO
      IF (CODE==2 .OR. CODE==4) DO
         ERR = ERREST(ORDER, PERT, POLY, CODE, MAXERR)
      DOEND
      YNEW = ASHRND(YNEW, N, CODE, XX, MAXERR, XMU(I))
   DOEND

   IF (MASGLVL>1) OUTIT(N, POLY, RHS, YNEW, YCORR, PERT)

   IF (N < NMAX) DO      # SET UP FOR NEXT ITERATION.
      MAXN = N
      N = IMIN(N*2, NMAX)
      POLY = SETP2(N, ORDER, MAXN, DIFFEQ, YNEW, RHS)
   DOEND

   ELSE DO      # THIS WAS THE LAST ITERATION.

      # OBTAIN PERTURBATION ON ORIGINAL SYSTEM.

      UNLIST = I$(XY)
      VALIST = I$(YNEW)
      P = YNEW
      DO 1 = 1, ORDER
         UNLIST = (UNLIST, XDY(I))
         P = DIFF(P, XX)
         VALIST = (VALIST, P)
      DOEND
      PERT = DIFFEQ(UNLIST=VALIST)
   DOEND

   RETURN
END # END OF PROCEDURE NEXTIT.
PROCEDURE OUTIT(N,POLY,RHS,YNEW,YCORR,PERT)
   INTEGER VALUE N
   LONG ALGEBRAIC VALUE RHS,YNEW,YCORR,PERT
   LONG ALGEBRAIC ARRAY VALUE POLY

   # THIS PROCEDURE OUTPUTS THE RESULTS OF THE CURRENT ITERATION
   # IN THE SOLUTION OF AN ODE.
   # EXTERNAL VARIABLES REQUIRED: XX.
   # PACKAGE PROCEDURES REQUIRED: CHFORM, TCS, CWRIT.
   # SYSTEM PROCEDURES REQUIRED: DEG.
   # DECLARATION IN CALLING PROCEDURE: NONE.

   INTEGER DEGREE
   LONG ALGEBRAIC ARRAY C
   EXTERNAL LONG ALGEBRAIC XX
   LONG ALGEBRAIC ALTRAN CHFORM
   LONG ALGEBRAIC ARRAY ALTRAN TCS

   WRITE " THE COEFFICIENTS OF THE ODE ARE", POLY
   WRITE " THE RIGHT HAND SIDE OF ODE IS", RHS
   WRITE " THE DEGREE OF APPROXIMATION IS", N
   WRITE " THE CORRECTION TERM IS", YCORR
   WRITE "THE POLYNOMIAL SOLUTION IS", YNEW
   C = TCS(CHFORM(YNEW,N,XX),N)
   CWRIT(C,N)

   DEGREE = DEG(PERT,XX)
   PERT = CHFORM(PERT,DEGREE,XX)
   WRITE "THE PERTUBATION IS", PERT

   RETURN
END   # END OF PROCEDURE OUTIT.
PROCEDURE PRODTR(P,DEGREE)
   INTEGER VALUE DEGREE
   LONG ALGEBRAIC VALUE P

   # THIS PROCEDURE MULTIPLIES A POLYNOMIAL P OF GIVEN DEGREE
   # ASSUMED TO BE IN CHEBYSHEV FORM, BY THE CHEBYSHEV
   # POLYNOMIAL OF DEGREE R WHERE R IS AN INDETERMINATE.
   # EXTERNAL VARIABLES REQUIRED: XTR,XT.
   # SYSTEM PROCEDURES REQUIRED: GETBLK.
   # DECLARATION IN CALLING PROCEDURE: LONG ALGEBRAIC ALTRAN PRODTR.

   INTEGER J
   LONG ALGEBRAIC COEF,NEWP
   EXTERNAL LONG ALGEBRAIC ARRAY XTR,XT

   NEWP = 0
   DO J = 0,DEGREE
      COEF = GETBLK(P,XT(J),1)
      NEWP = NEWP + COEF*(XTR(J)+XTR(-J))/2
   DOEND

   RETURN(NEWP)

END  # END OF PROCEDURE PRODTR.
PROCEDURE PRODXR(P, DEGREE, X)
INTEGER VALUE DEGREE
LONG ALGEBRAIC VALUE P, X

# THIS PROCEDURE MULTIPLIES A POLYNOMIAL P OF GIVEN
# DEGREE IN X BY X**R WHERE R IS AN INDETERMINATE.
# EXTERNAL VARIABLES REQUIRED: XXR.
# SYSTEM PROCEDURES REQUIRED: GETBLK.
# DECLARATION IN CALLING PROCEDURE: LONG ALGEBRAIC ALTRAN PRODXR.

INTEGER J
LONG ALGEBRAIC NEWP, COEF
EXTERNAL LONG ALGEBRAIC ARRAY XXR

NEWP = 0
DO J = 0, DEGREE
   COEF = GETBLK(P, X, J)
   NEWP = NEWP + COEF * XXR(J)
END
RETURN(NEWP)

END # END OF PROCEDURE PRODXR.
PROCEDURE PKINIT(DEFLT)
INTEGER ARRAY DEFLT

# THIS PROCEDURE Initializes THE PACKAGE FOR THE VARIOUS SOLVER
# SOLVER MODULES. THE VALUES OF THE PACKAGE PARAMETERS THAT MAY
# BE SET BY THE USER ARE CHECKED AGAINST THE DEFAULT (MAXIMUM) VALUES
# AND ARE RESET WHERE NECESSARY.
# INPUT PARAMETER:
# DEFLT – AN ARRAY OF DEFAULT (MAXIMUM) VALUES OF PARAMETERS
# INTERPRETED AS FOLLOWS:
# DEFLT(1) – MAXORD, THE MAXIMUM ORDER OF ODE ALLOWED;
# DEFLT(2) – MAXDEG, THE MAXIMUM DEGREE OF APPROXIMATION ALLOWED;
# DEFLT(3) – MAXMU, THE MAXIMUM NUMBER OF INDETERMINATES ALLOWED
# IN A PROBLEM;
# DEFLT(4) – YDEG, THE MAXIMUM DEGREE OF ODE ALLOWED;
# DEFLT(5) – MAXCOD, THE MAXIMUM VALUE OF THE SOLUTION CODE.
# OTHER PACKAGE PARAMETERS:
# MAXVAL – MAXIMUM NUMBER OF X-VALUES ALLOWED IN THE ASSOCIATED
# CONDITIONS;
# XDEG – MAXIMUM EXPONENT OF THE INDEPENDENT VARIABLE X;
# MAXPAR – MAXIMUM NUMBER OF SYMBOLIC PARAMETERS INTRODUCED
# DURING THE COMPUTATION;
# MSG_LVL – THE MESSAGE LEVEL INDICATOR, CONTROLLING THE
# AMOUNT OF INFORMATION PRINTED BY THE PACKAGE.
# EXTERNAL VARIABLES: ALL PACKAGE PARAMETERS ARE EXTERNAL VARIABLES.
# SYSTEM PROCEDURES REQUIRED: IMIN,IMAX,NULL.
# DECLARATION IN CALLING PROCEDURE: NONE.

EXTERNAL INTEGER MAXORD,MAXDEG,MAXVAL,MAXCOD,MAXMU
EXTERNAL INTEGER MAXPAR,XDEG,YDEG,MSG_LVL,IERR,INPFLG

# CHECK AND SET (INTERFACE) PACKAGE PARAMETERS.
IF (NULL(MAXORD)) MAXORD = DEFLT(1)  # MAXIMUM ORDER.
ELSE IF (MAXORD<0 .OR. MAXORD>5) DO
  WRITE "$ *** ERROR 5011 *** – INVALID MAXIMUM ORDER",MAXORD
  MAXORD = DEFLT(1)
  WRITE "$ MAXIMUM ORDER RESET TO ", MAXORD
DOEND
ELSE MAXORD = IMAX(MAXORD,1)

IF (NULL(MAXDEG)) MAXDEG = DEFLT(2)  # MAXIMUM DEGREE.
ELSE IF (MAXDEG<0 .OR. MAXDEG>15) DO
  WRITE "$ *** ERROR 5012 *** – INVALID MAXIMUM DEGREE",MAXDEG
  MAXDEG = DEFLT(2)
  WRITE "$ MAXDEG RESET TO ", MAXDEG
DOEND
IF (NULL(MAXMU)) MAXMU = DEFLT(3)  # NUMBER OF INDETERMINATES
ELSE IF (MAXMU<0 .OR. MAXMU>DEFLT(3)) DO
  WRITE " ERROR 5013 ****
  WRITE " INVALID MAXIMUM NUMBER OF INDETERMINATES"
  MAXMU = DEFLT(3)
  WRITE " MAXMU RESET TO ", MAXMU
DOEND

IF (NULL(MAXVAL)) MAXVAL = 5  # NUMBER OF X-VALUES.
ELSE IF (MAXVAL<0 .OR. MAXVAL>5) DO
  WRITE " *** ERROR 5014 ****
  WRITE " INVALID MAXIMUM NUMBER OF X-VALUES"
  MAXVAL = 5
  WRITE " MAXVAL RESET TO ", MAXVAL
DOEND
ELSE MAXVAL = IMAX(MAXVAL,1)

IF (NULL(MSGLVL)) MSGLVL = 1  # MESSAGE INDICATOR.
ELSE IF (MSGLVL <1 .OR. MSGLVL >4) DO
  WRITE " *** ERROR 5015 ****
  WRITE " INVALID MESSAGE INDICATOR"
  MSGLVL = 1
  WRITE " MSGLVL RESET TO", MSGLVL
DOEND
WRITE MAXORD,MAXDEG,MAXMU,MAXVAL,MSGLVL

# SET OTHER PACKAGE PARAMETERS.

YDEG = DEFLT(4)
MAXPAR = IMIN(15,MAXDEG+10)
XDEG = IMIN(MAXDEG*YDEG + MAXPAR,40)
MAXCOD = DEFLT(5)

WRITE " 
WRITE " ******** PACKAGE Initialization COMPLETED ******** "
WRITE " 

RETURN
END # END OF PROCEDURE PKINIT.
PROCEDURE PROLGL(N, ORDER, DiffEq, CODE, POLY, 
RHS, MaxSub, XSub, CondN, Range) 
INTEGER N, ORDER, MaxSub, CODE 
RATIONAL ARRAY xSub, Range 
LONG ALGEBRAIC DiffEq, RHS 
LONG ALGEBRAIC ARRAY POLY, CondN 

# THIS PROCEDURE READS IN AN ODE AND ITS ASSOCIATED CONDITIONS AND 
# CHECKS THE VALIDITY OF THE ORDER, THE DEGREE OF APPROXIMATION 
# REQUIRED, THE SOLUTION CODE, AND THE RANGE IN WHICH THE SOLUTION 
# IS REQUIRED. A FATAL ERROR OCCURS IF THE ODE IS NONLINEAR. 
# THE ACTUAL INPUT AND PARAMETER CHECKING ARE PERFORMED BY 
# A CALL TO PROCEDURE READER. 
# INPUT PARAMETERS: NONE. 
# OUTPUT PARAMETERS: 
N — DEGREE OF APPROXIMATION REQUIRED; 
ORDER — ORDER OF ODE; 
MaxSub — NUMBER OF X-VALUES IN ASSOCIATED CONDITIONS; 
CODE — SOLUTION CODE; 
xSub — AN ARRAY XSub(1),...,XSub(MaxSub) OF X-VALUES APPEARING 
in the associated conditions; 
Range — A RATIONAL ARRAY (A,B) INDICATING THE FINITE INTERVAL 
in which the solution is desired; 
DiffEq — THE ODE EXPRESSED AS DiffEq=0; 
POLY — ARRAY POLY(0),...,POLY(Order) OF POLYNOMIAL COEFFICIENTS 
in ODE; 
RHS — RIGHT-HAND-SIDE POLYNOMIAL OF ODE; 
CondN — ARRAY CondN(1),...,CondN(Order) OF ASSOCIATED CONDITIONS. 
EXTERNAL VARIABLES REQUIRED: MSGlvl. 
PACKAGE PROCEDURES REQUIRED: READER, SETP1. 
DECLARATION IN CALLING PROCEDURE: NONE. 

LOGICAL LINEAR 
EXTERNAL INTEGER MSGlvl 
LONG ALGEBRAIC ARRAY ALTRAN SETP1 

# READ IN AND CHECK ODE AND ASSOCIATED CONDITIONS. 
READER(N, ORDER, DiffEq, CODE, MaxSub, XSub, CondN, Range, LINEAR) 

# CHECK THAT ODE IS LINEAR. 
IF (.NOT. LINEAR) DO 
WRITE "] "*** ERROR 5024 ***" 
WRITE "] INVALID ODE — ODE IS NOT LINEAR" 
FRETURN 
DOEND 

# SET UP POLYNOMIAL COEFFICIENTS IN ODE. 
POLY = SETP1(Order, DiffEq, RHS) 
IF (MSGlvl > 1) DO 
WRITE "] THE POLYNOMIAL COEFFICIENTS OF THE ODE ARE:",POLY 
WRITE "] THE RIGHT HAND SIDE POLYNOMIAL IS:", RHS 
DOEND 

RETURN 
END # END OF PROCEDURE PROLGL.
PROCEDURE QROUND(Q,K,M)
INTEGER VALUE K,M
LONG RATIONAL ARRAY VALUE Q

# THIS PROCEDURE ROUNDS THE ELEMENTS OF THE RATIONAL ARRAY
# Q(0), ...., Q(K) TO M DECIMAL DIGITS.
# PACKAGE PROCEDURES REQUIRED: LQABS.
# SYSTEM PROCEDURES REQUIRED: INUM,IFLR.
# DECLARATION IN CALLING PROCEDURE: LONG RATIONAL ARRAY ALTRAN QROUND.

INTEGER I
LONG INTEGER A,B,C,DEN,REM
LONG RATIONAL R
LONG INTEGER ALTRAN IFLR=S9IFLR
LONG RATIONAL ALTRAN LQABS

DEN = 10**M
DO I = 0,K
   R = Q(I)
   A = INUM(R,B)

   C = IFLR(A*DEN,B,REM)
   IF ( (2*REM) > B ) C = C + 1
   Q(I) = C/DEN
END DO

RETURN(Q)

END # END OF PROCEDURE QROUND.
INTEGER NMAX,ORDER,CODE
LOGICAL LINEAR
RATIONAL ARRAY XSUB,RANGE
LONG ALGEBRAIC DIFFEQ
LONG ALGEBRAIC ARRAY CONDN

# THIS PROCEDURE READS IN AND CHECKS AN ODE PROBLEM, AND THE
# ASSOCIATED PROBLEM PARAMETERS.
# EXTERNAL VARIABLES REQUIRED: INPFLG,MAXORD,MAXDEG,MAXCOD.
# PACKAGE PROCEDURES REQUIRED: INPODE,INPCON,TRANSI.
# DECLARATION IN CALLING PROCEDURE: NONE.

RATIONAL ARRAY(2) RG
EXTERNAL INTEGER INPFLG,MAXORD,MAXDEG,MAXCOD

# READ IN ODE AND ASSOCIATED CONDITIONS.

INPFLG = 0
READ ORDER,NMAX,RG,CODE
RANGE = RG
WRITE ORDER,NMAX,RANGE,CODE
INPODE(ORDER,DIFFEQ,LINAER)
INPCON(ORDER,MAXSUB,XSUB,CONDN)
INPFLG = 1

# CHECK PROBLEM PARAMETERS.

IF (ORDER<0 .OR. ORDER>MAXORD) DO
   # ORDER.
   WRITE " *** ERROR 5021 ***"
   WRITE " ORDER OUT OF RANGE"
   RETURN
END DO

IF (NMAX<0 .OR. NMAX>MAXDEG) DO
   # NMAX.
   WRITE " *** ERROR 5022 ***"
   WRITE " NMAX OUT OF RANGE"
   RETURN
END DO

IF (CODE<1 .OR. CODE>MAXCOD) DO
   # CODE.
   WRITE " *** ERROR 5023 ***"
   WRITE " SOLUTION CODE OUT OF RANGE"
   RETURN
END DO

IF (RANGE(1) >= RANGE(2)) DO
   # RANGE.
   WRITE " *** ERROR 5027 ***"
   WRITE " INVALID RANGE"
   RETURN
END DO

TRANSFORM PROBLEM TO (-1,1) IF NECESSARY.

IF (.NOT. RANGE==(-1,1)) DO
   TRANSI(ORDER,RANGE,DIFFEQ,CONDN,XSUB,MAXSUB)
   WRITE " PROBLEM TRANSFORMED TO RANGE (-1,1)"
   WRITE " THE TRANSFORMED PROBLEM IS GIVEN BY"
   WRITE DIFFEQ,CONDN,XSUB
END DO

RETURN
PROCEDURE REDUZS(Q,Z,Z1,G,G1,H,N,GENPOL,V,V1,S,S1)
   INTEGER VALUE G1,H,N,Z1
   INTEGER V1=0,S1=Z1
   INTEGER ARRAY VALUE G,Z
   INTEGER ARRAY S=Z
   LONG ALGEBRAIC VALUE GENPOL
   LONG ALGEBRAIC ARRAY (0:N+H) Q
   LONG ALGEBRAIC ARRAY V

# THIS PROCEDURE OBTAINS THE MINIMAL SET S OF INDICES OF
# UNDEFINED CANONICAL POLYNOMIALS ASSOCIATED WITH A
# LINEAR OPERATOR DEFINED BY THE GENERATING POLYNOMIAL
# GENPOL.
# EXTERNAL VARIABLES REQUIRED: XX,XXR,XUQ.
# PACKAGE PROCEDURES REQUIRED: SUBS.
# SYSTEM PROCEDURES REQUIRED: TPS,GETBLK,DEG.
# DECLARATION IN CALLING PROCEDURE: NONE.

INTEGER I,J,K,L,M,RP
LONG ALGEBRAIC ARRAY (0:H) COEF
LONG ALGEBRAIC B,C,J,P,T,SUM
LONG ALGEBRAIC ARRAY XLIST,VALIST,L1,L2,A
EXTERNAL LONG ALGEBRAIC XX
EXTERNAL LONG ALGEBRAIC ARRAY XXR,XUQ
LONG ALGEBRAIC ALTRAN SUBS
LONG ALGEBRAIC ARRAY ALTRAN TPS

# COMMENCE REDUCTION IF POSSIBLE.

IF (G1 < 1) RETURN  # NO REDUCTION POSSIBLE.

DO I = 0,H
   COEF(I) = GETBLK(GENPOL,XXR(I),1)
DOEND
XLIST=0S0; VALIST=0S0
L1 = 0S0; L2 =Z1S0
DO I = 1,Z1
   L1 = (L1,XUQ(Z(I)))
DOEND
V1 = 0; L=1; M=0

BACK: IF ( L > G1) GO TO FINI
K = G(L)
C = XX**K
B = SUBS(C,COEF,H,XX)
L = L+1
CHECKB:
IF (B=0) DO  # IS C A POLYNOMIAL SOLUTION?
   V = (V,C)
   V1 = V1 + 1
   GO TO BACK
DOEND
ELSE DO
   RP = DEG(B,XX)
   A = TPS(B,XX,RP)
   JP = A(RP)
   IF ( Q(RP)<=XUQ(RP) ) DO
      T = Q(RP)(L1=L2)
      B = B - JP*SUBS(T,COEF,H,XX)
      C = C - JP*T
      GO TO CHECKB
   DOEND
ELSE DO  # DEFINE Q(RP).
   SUM = C
   DO I = 0,RP-1
      SUM = SUM - Q(I)*A(I)
   DOEND
   Q(RP) = SUM/JP
   M = M+1
   XLIST = (XLIST,XUQ(RP))
   VALIST = (VALIST,Q(RP))
   GO TO BACK
DOEND
DOEND
FINI:  # SUBSTITUTE FOR ANY NEWLY DEFINED CAN. POLYS.
IF (M>0) DO
   S1 = 0; S=0$0
   DO I =0,N+H
      IF (Q(I)==XUQ(I)) DO
         S1 = S1 + 1
         S = (S,I)
      DOEND
   ELSE Q(I) = Q(I)(XLIST=VALIST)
   DOEND
DOEND
END  # END OF PROCEDURE REDUZS
PROCEDURE SETPI(ORDER,DIFFEQ,RHS)
INTEGER VALUE ORDER
LONG ALGEBRAIC VALUE DIFFEQ
LONG ALGEBRAIC RHS

# THIS PROCEDURE RETURNS THE POLYNOMIAL COEFFICIENTS OF
# A LINEAR ODE OF GIVEN ORDER, THE RIGHT-HAND-SIDE
# POLYNOMIALS ARE RETURNED IN RHS.
# EXTERNAL VARIABLES REQUIRED: MSGVL,J,XX,XY,XYD.
# SYSTEM PROCEDURES REQUIRED: GETBLK.
# DECLARATION IN CALLING PROCEDURE: LONG ALGEBRAIC ARRAY ALTRAN SETPI.

INTEGER I,J,K,NEWORD
LONG ALGEBRAIC P,LHS
LONG ALGEBRAIC ARRAY (0:ORDER) POLY=0
EXTERNAL INTEGER MSGVL
EXTERNAL LONG ALGEBRAIC XX,XY
EXTERNAL LONG ALGEBRAIC ARRAY XDY

IF (MSGVL>3) WRITE "ENTER SETPI"
# PICK OFF THE POLYNOMIAL COEFFICIENTS.

LHS = 0
NEWORD = -1
DIFFEQ = DIFFEQ(XY=XYD(0))

DO I = 0,ORDER
   P = GETBLK(DIFFEQ,XYD(I),1)
   IF (P<>0) DO
      POLY(I) = P
      NEWORD = I
      LHS = LHS + P*XYD(I)
   ENDDO
ENDDO
RHS = LHS - DIFFEQ

IF (NEWORD <> ORDER) DO
   WRITE " *** ERROR 5025 *** 
   WRITE " WRONG ORDER SPECIFIED FOR ODE", ORDER
   WRITE " ODE IS ACTUALLY OF ORDER", NEWORD
   FRETURN
ENDDO

IF (MSGVL>3) WRITE POLY,RHS, "EXIT SETPI"
RETURN(POLY)

END   # END OF PROCEDURE SETPI.
PROCEDURE SETP2(N,ORDER,MAXN,DIFFEQ,YINIT,RHS)
INTEGER VALUE N,ORDER,MAXN
LONG ALGEBRAIC VALUE DIFFEQ,YINIT
LONG ALGEBRAIC RHS

# THIS PROCEDURE RETURNS THE POLYNOMIAL COEFFICIENTS,POLY,
# OF THE LINEAR ODE OBTAINED FROM A NONLINEAR ODE (OF GIVEN
# ORDER), DIFFEQ, BY NEWTON ITERATION. THE
# RIGHT-HAND-SIDE POLYNOMIAL OF THE LINEAR ODE IS
# RETURNED IN RHS.
# YINIT IS THE CURRENT ESTIMATE FOR THE SOLUTION FUNCTION Y(X).
# THE POLYNOMIAL COEFFICIENTS ARE TRUNCATED TO DEGREE MAXN IN X.
# EXTERNAL VARIABLES REQUIRED: MSGVL,XX,XY,XY.
# PACKAGE PROCEDURES REQUIRED: TRUNC.
# SYSTEM PROCEDURES REQUIRED: DIFF.
# DECLARATION IN CALLING PROCEDURE: LONG ALGEBRAIC ARRAY ALTRAN SETP2.

INTEGER I,J,K,NEWORD
LONG ALGEBRAIC P,LHS
LONG ALGEBRAIC ARRAY YLIST,YVAL
LONG ALGEBRAIC ARRAY(0:ORDER) POLY=0
EXTERNAL INTEGER MSGVL
EXTERNAL LONG ALGEBRAIC XX,XY
EXTERNAL LONG ALGEBRAIC ARRAY XDY
LONG ALGEBRAIC ALTRAN TRUNC

IF (MSGVL>3) WRITE "ENTER SETP2"
LHS = 0
NEWORD = -1
DIFFEQ = DIFFEQ(XY=XY(0))

SET UP LISTS OF Y AND DERIVATIVE VALUES.

YLIST = XDY(0)
YVAL = YINIT
P = YINIT
DO 1 = 1,ORDER
   P = DIFF(P,XX)
   YLIST = (YLIST,XDY(I))
   YVAL = (YVAL,P)
1)
# NOW GET THE COEFFICIENTS.

RHS = DIFEQ(YLIST = YVAL)
RHS = - TRUNC(RHS,N,XX)
DO I =0,ORDER
   P = DIFF(DIFEQ.XDY(I))(YLIST=YVAL)
   IF (P<>0) DO
      P = TRUNC(P,MAXN,XX)
      NEWORD = I
      POLY(I) = P
   DOEND
DOEND

# CHECK THE ORDER OF ODE.

IF (NEWORD <> ORDER) DO
   WRITE " *** ERROR 5025 ***"
   WRITE " WRONG ORDER SPECIFIED FOR ODE",ORDER
   WRITE " ODE IS ACTUALLY OF ORDER",NEWORD
   FRETURN
DOEND

IF (MSGlvl>3) WRITE POLY,RHS,"EXIT SETP!"
RETURN(POLY)
END # END OF PROCEDURE SETP2.
<table>
<thead>
<tr>
<th>Procedure</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>ABS</td>
<td>93</td>
</tr>
<tr>
<td>AMAX</td>
<td>94</td>
</tr>
<tr>
<td>CHROM</td>
<td>95</td>
</tr>
<tr>
<td>COM</td>
<td>96</td>
</tr>
<tr>
<td>CWRT</td>
<td>97</td>
</tr>
<tr>
<td>EVAL</td>
<td>98</td>
</tr>
<tr>
<td>IABS</td>
<td>99</td>
</tr>
<tr>
<td>INFORM</td>
<td>100</td>
</tr>
<tr>
<td>LQABS</td>
<td>101</td>
</tr>
<tr>
<td>POWERS</td>
<td>102</td>
</tr>
<tr>
<td>PREC</td>
<td>103</td>
</tr>
<tr>
<td>SUBS</td>
<td>104</td>
</tr>
<tr>
<td>TCS</td>
<td>105</td>
</tr>
<tr>
<td>TPOWS</td>
<td>106</td>
</tr>
<tr>
<td>TRANS1</td>
<td>107</td>
</tr>
<tr>
<td>TRANS2</td>
<td>108</td>
</tr>
<tr>
<td>TRUNC</td>
<td>109</td>
</tr>
<tr>
<td>Procedure</td>
<td>Page</td>
</tr>
<tr>
<td>-----------</td>
<td>------</td>
</tr>
<tr>
<td>ABS</td>
<td>93</td>
</tr>
<tr>
<td>AMAX</td>
<td>94</td>
</tr>
<tr>
<td>CHFORM</td>
<td>95</td>
</tr>
<tr>
<td>COM</td>
<td>96</td>
</tr>
<tr>
<td>CWRITE</td>
<td>97</td>
</tr>
<tr>
<td>EVAL</td>
<td>98</td>
</tr>
<tr>
<td>IABS</td>
<td>99</td>
</tr>
<tr>
<td>INFORM</td>
<td>100</td>
</tr>
<tr>
<td>LQABS</td>
<td>101</td>
</tr>
<tr>
<td>POW1NS</td>
<td>102</td>
</tr>
<tr>
<td>PRED</td>
<td>103</td>
</tr>
<tr>
<td>SUBS</td>
<td>104</td>
</tr>
<tr>
<td>TCS</td>
<td>105</td>
</tr>
<tr>
<td>TPROC5</td>
<td>106</td>
</tr>
<tr>
<td>TRANS</td>
<td>107</td>
</tr>
<tr>
<td>TRANS2</td>
<td>108</td>
</tr>
<tr>
<td>TRIGG</td>
<td>109</td>
</tr>
</tbody>
</table>
PROCEDURE ABS(X)
  LONG REAL X

# THIS PROCEDURE RETURNS THE ABSOLUTE VALUE OF
# THE REAL NUMBER X.
# DECLARATION IN CALLING PROCEDURE: REAL ALTRAN ABS.

  IF (X < 0.0) RETURN(-X)
  ELSE RETURN(X)

END # END OF PROCEDURE ABS.
PROCEDURE AMAX(P,DEGREE,X)
   INTEGER VALUE DEGREE
   LONG ALGEBRAIC VALUE P,X

# THIS PROCEDURE RETURNS THE MAXIMUM ABSOLUTE VALUE
# OF A POLYNOMIAL P, OF GIVEN DEGREE IN X, IN THE
# INTERVAL -1<=X<=1. THE COEFFICIENTS OF P ARE
# ASSUMED TO BE RATIONAL NUMBERS.
# PACKAGE PROCEDURES REQUIRED: LQABS,CHFORM,TCS
# DECLARATION IN CALLING PROCEDURE: LONG ALGEBRAIC ALTRAN AMAX.

INTEGER I
LONG ALGEBRAIC MAXP, PMAX
LONG ALGEBRAIC ARRAY A
LONG RATIONAL ALTRAN LQABS
LONG ALGEBRAIC ALTRAN CHFORM
LONG ALGEBRAIC ARRAY ALTRAN TCS

PMAX = CHFORM(P,DEGREE,X)
A = TCS(PMAX,DEGREE)

MAXP = LQABS(A(0))/2
DO I = 1,DEGREE
   MAXP = MAXP + LQABS(A(I))
END

RETURN(MAXP)

END # END OF PROCEDURE AMAX.
PROCEDURE CHFORM(P, DEGREE, X)
   INTEGER VALUE DEGREE
   LONG ALGEBRAIC VALUE P, X

   PROCEDURE TO COMPUTE THE CHEBYSHEV FORM OF A POLYNOMIAL P
   OF GIVEN DEGREE IN THE INDETERMINATE X. IT IS ASSUMED
   THAT THE EXTERNAL ARRAY XPOWER HAS BEEN INITIALIZED TO
   CONTAIN THE CHEBYSHEV FORM OF THE POWER X**K AS ITS
   ELEMENT XPOWER(K), FOR 0 <= K <= DEGREE.
   EXTERNAL VARIABLES REQUIRED: XPOWER.
   SYSTEM PROCEDURES REQUIRED: GETBLK.
   DECLARATION IN CALLING PROCEDURE: LONG ALGEBRAIC ALTRAN CHFORM.

   INTEGER K
   LONG ALGEBRAIC COEF, NEWP
   EXTERNAL LONG ALGEBRAIC ARRAY XPOWER

   NEWP = 0
   DO K = 0, DEGREE
      COEF = GETBLK(P, X, K)
      NEWP = NEWP + COEF*XPOWER(K)
   DOEND

   RETURN(NEWP)

END  # END OF PROCEDURE CHFORM.
PROCEDURE COM(K)
    INTEGER VALUE K
    INTEGER I,J
    INTEGER ARRAY (1:K,1:K) COMB

    # THIS PROCEDURE RETURNS A K*K ARRAY OF BINOMIAL COEFFICIENTS.
    # DECLARATION IN CALLING PROCEDURE: INTEGER ARRAY ALTRAN COM.

    DO I = 1,K
        COMB(I,1) = 1
        DO J = 2,I-1
            COMB(I,J) = COMB(I-1,J-1) + COMB(I-1,J)
        DOEND
        COMB(I,I) = 1
    DOEND

    RETURN(COMB)
END # END OF PROCEDURE COM.
PROCEDURE CWRT(C,N)
   INTEGER VALUE N
   LONG ALGEBRAIC ARRAY VALUE C

# THIS PROCEDURE PRINTS THE N+1 COEFFICIENTS CHEBYSHEV COEFFICIENTS
# AS REALS IF POSSIBLE (I.E. IF THE COEFFICIENTS CONTAIN NO
# INDETERMINATES) AND OTHERWISE IN SYMBOLIC FORM.
# DECLARATION IN CALLING PROCEDURE: NONE.

INTEGER I, J
LONG REAL ARRAY(0:N) CI REAL
RATIONAL ALTRAN LQT

WRITE "THE CHEBYSHEV COEFFICIENTS ARE:"

DO J = 0, N
   C(J) = LQT(C(J), ASIS)
   CI REAL(J) = LR(C(J))
   WRITE CI REAL(J)
   GO TO ON
ASIS: WRITE C(J)
ON: DOEND

RETURN

END # END OF PROCEDURE CWRT.
PROCEDURE EVAL(C, DEGREE, X)
   INTEGER VALUE DEGREE
   LONG ALGEBRAIC VALUE X
   LONG ALGEBRAIC ARRAY VALUE C

   # THIS PROCEDURE EVALUATES A TRUNCATED CHEBYSHEV SERIES C
   # OF GIVEN DEGREE AT THE GIVEN ALGEBRAIC POINT X.
   # DECLARATION IN CALLING PROCEDURE: LONG ALGEBRAIC ALTRAN EVAL.

   INTEGER I
   LONG ALGEBRAIC BOLD, TEMP, TWOX, B

   TWOX = 2*X
   BOLD = 0; TEMP = 0
   B = C(DEGREE)

   DO I = DEGREE-1, 0, -1
      BOLD = TEMP
      TEMP = B
      B = TWOX*B - BOLD + C(I)
   DOEND

   RETURN( (B - BOLD)/2)

END # END OF PROCEDURE EVAL.
PROCEDURE IABS(K)
    INTEGER VALUE K

    # THIS PROCEDURE RETURNS THE ABSOLUTE VALUE OF THE SHORT
    # INTEGER K.
    # DECLARATION IN CALLING PROCEDURE: INTEGER ALTRAN IABS.

    IF (K<0) RETURN(-K)
    ELSE RETURN(K)

END  # END OF PROCEDURE IABS.
PROCEDURE INFORM(ORDER, POLY, X)
INTEGER VALUE ORDER
LONG ALGEBRAIC VALUE X
LONG ALGEBRAIC ARRAY VALUE POLY

# THIS PROCEDURE CONVERTS A LINEAR ODE WITH POLYNOMIAL
# COEFFICIENTS TO INTEGRATED FORM.
# THE COEFFICIENTS OF THE INTEGRATED FORM ARE RETURNED
# IN Q(0), ..., Q(K), WHERE K IS THE ORDER OF THE ODE.
# PACKAGE PROCEDURES REQUIRED: COM
# SYSTEM PROCEDURES REQUIRED: DIFFN.
# DECLARATION IN CALLING PROCEDURE: LONG ALGEBRAIC ARRAY ALTRAN INFORM.

INTEGER M, I, SIGN
INTEGER ARRAY COMB
LONG ALGEBRAIC P
LONG ALGEBRAIC ARRAY(0:ORDER) Q
LONG ALGEBRAIC ARRAY XLIST
INTEGER ARRAY ALTRAN COM
LONG ALGEBRAIC ALTRAN DIFFN

COMB = COM(ORDER)
Q(0) = POLY(ORDER)
DO M = 1, ORDER
  # DETERMINE THE M-TH POLYNOMIAL IN THE INTEGRATED FORM.
  P = POLY(ORDER - M) - (ORDER - M + 1) * DIFFN(POLY(ORDER - M + 1), X)
  SIGN = -1
  XLIST = ISX
  DO I = M - 2, 0, -1
    SIGN = -SIGN
    XLIST = (XLIST, X)
    P = P + SIGN * COMB(ORDER - I, M - I) * DIFFN(POLY(ORDER - I), XLIST)
  ENDDO
  Q(M) = P
ENDDO

RETURN(Q)
END # END OF PROCEDURE INFORM.
PROCEDURE LQABS(K)
    LONG RATIONAL VALUE K

    # THIS PROCEDURE RETURNS THE ABSOLUTE VALUE OF THE LONG
    # RATIONAL NUMBER K.
    # DECLARATION IN CALLING PROCEDURE: LONG RATIONAL ALTRAN LQABS.

    IF (K>=0) RETURN(K)
    ELSE RETURN(-K)

END    # END OF PROCEDURE LQABS.
PROCEDURE POWERS(MAXDEG)
   INTEGER VALUE MAXDEG

   # PROCEDURE TO COMPUTE THE CHEBYSHEV FORM OF THE POWERS X**K
   # FOR 0 <= K <= MAXDEG, THE K-TH CHEBYSHEV POLYNOMIAL IS
   # REPRESENTED BY THE INDETERMINATE XT(K), WHERE XT IS AN
   # EXTERNAL ALGEBRAIC ARRAY OF INDETERMINATES. THE VALUE
   # RETURNED IS AN ARRAY DIMENSIONED 0 TO MAXDEG CONTAINING
   # THE APPROPRIATE CHEBYSHEV FORMS.
   # EXTERNAL VARIABLES REQUIRED: XT.
   # SYSTEM PROCEDURES REQUIRED: IMOD.
   # DECLARATION IN CALLING PROCEDURE: LONG ALGEBRAIC ARRAY ALTRAN POWERS.

INTEGER K, HALFK, KMOD2, J
LONG INTEGER ARRAY KCHOOS, TEMP
LONG ALGEBRAIC ARRAY (0:MAXDEG) A
EXTERNAL LONG ALGEBRAIC ARRAY XT

A(0) = XT(0)
IF (MAXDEG > 0) A(1) = XT(1)
IF (MAXDEG > 1) A(2) = (XT(2)+XT(0))/2
KCHOOS = 1$(1)

DO K = 3, MAXDEG

   KMOD2 = IMOD(K, 2, HALFK)
   # UPDATE ARRAY KCHOOS SO THAT KCHOOS(J) = "K CHOOSE J",
   # FOR J = 1, ..., K/2 .
   IF (KMOD2 == 1) TEMP = KCHOOS
   ELSE TEMP = (KCHOOS, KCHOOS(HALFK−1))
   KCHOOS = HALFK$0) # CREATES ARRAY OF DESIRED LENGTH.
   KCHOOS(1) = K
   DO J = 2, HALFK
      KCHOOS(J) = TEMP(J) + TEMP(J−1)
   DOEND

   # COMPUTE CHEBYSHEV FORM OF X**K INTO ARRAY ELEMENT A(K).
   A(K) = XT(K)
   DO J = 1, HALFK−1
      A(K) = A(K) + KCHOOS(J) * XT(K−2*J)
   DOEND
   IF (KMOD2==1) A(K) = (A(K) + KCHOOS(HALFK)*XT(1))/2**(K−1)
   ELSE A(K) = (A(K) + 1/2*KCHOOS(HALFK)*XT(0))/2**(K−1)
   DOEND

RETURN(A)

END OF PROCEDURE POWERS.
PROCEDURE PREC(X)
REAL VALUE X

# THIS PROCEDURE COMPUTES THE PRECISION OF THE REAL NUMBER X.
# THAT IS, IT RETURNS THE INTEGER M SUCH THAT 10**(-M) <= X.
# DECLARATION IN CALLING PROCEDURE: INTEGER ALTRAN PREC.

INTEGER M
REAL Y,T

M = 0
Y = 1.0
T = 0.1
BACK:
IF (Y > X) DO
  M = M + 1
  Y = Y*T
  GO TO BACK
DOEND

RETURN(M)

END # END OF PROCEDURE PREC.
PROCEDURE SUBS(P,COEF,H,X)
  INTEGER VALUE H
  LONG ALGEBRAIC VALUE P,X
  LONG ALGEBRAIC ARRAY VALUE COEF

  # THIS PROCEDURE SUBSTITUTES A POLYNOMIAL P INTO
  # A GENERATING POLYNOMIAL (OF HEIGHT H), WHOSE COEFFICIENTS
  # ARE GIVEN IN ARRAY COEF.
  # EXTERNAL VARIABLES REQUIRED: RR.
  # SYSTEM PROCEDURES REQUIRED: TPS, DEG.
  # DECLARATION IN CALLING PROCEDURE: LONG ALGEBRAIC ALTRAN SUBS.

  INTEGER I,J,DEGREE
  LONG ALGEBRAIC SUM,T,Q
  LONG ALGEBRAIC ARRAY B
  EXTERNAL LONG ALGEBRAIC RR
  LONG ALGEBRAIC ARRAY ALTRAN TPS

  DEGREE = DEG(P,X)
  B = TPS(P,X,DEGREE)
  Q = 0
  DO I = 0,DEGREE
    SUM = 0
    DO J = 0,H
      T = COEF(J)(RR=I)
      IF (T<> 0) SUM = SUM + T*X**(I+J)
    DOEND
    Q = Q + B(I)*SUM
  DOEND

  RETURN(Q)
END # END OF PROCEDURE SUBS.
PROCEDURE TCS(P, DEGREE)
    INTEGER VALUE DEGREE
    LONG ALGEBRAIC VALUE P

    # THIS PROCEDURE EXTRACTS THE CHEBYSHEV COEFFICIENTS, C,
    # OF A POLYNOMIAL P WHICH IS ASSUMED TO BE IN THE FORM
    #    C(0)*XT(0)/2 + C(1)*XT(1) + ... + C(N)*XT(N),
    # WHERE N = DEGREE, AND XT(I) REPRESENTS THE I-TH CHEBYSHEV
    # POLYNOMIAL.
    # EXTERNAL VARIABLES REQUIRED: XT.
    # SYSTEM PROCEDURES REQUIRED: GETBLK.
    # DECLARATION IN CALLING PROCEDURE: LONG ALGEBRAIC ARRAY ALTRAN TCS.

    INTEGER K
    LONG ALGEBRAIC ARRAY(0:DEGREE) C
    EXTERNAL LONG ALGEBRAIC ARRAY XT

    DO K = 0, DEGREE
        C(K) = GETBLK(P, XT(K), 1)
        P = P - C(K)*XT(K)
    DOEND
    C(0) = C(0)*2

    RETURN(C)
END # END OF PROCEDURE TCS.
PROCEDURE TPOWS(MAXDEG,X)
INTEGER VALUE MAXDEG
LONG ALGEBRAIC VALUE X

# THIS PROCEDURE RETURNS THE CHEBYSHEV POLYNOMIALS OF DEGREE
# NOT EXCEEDING MAXDEG. THE POLYNOMIALS ARE IN THE
# INDEPENDENT VARIABLE X.
# DECLARATION IN CALLING PROCEDURE: LONG ALGEBRAIC ARRAY ALTRAN TPOWS.

LONG ALGEBRAIC ARRAY (0:MAXDEG) A
LONG ALGEBRAIC R = 1, S,Y, T
INTEGER I

Y = X
S = Y
A(0) = R
IF (MAXDEG == 0) RETURN(A)
A(I) = S
IF (MAXDEG == 1) RETURN(A)

DO I = 2,MAXDEG
   T = 2*Y*S - R
   A(I) = T
   R = S
   S = T
ENDDO

RETURN(A)
END # END OF PROCEDURE TPOWS.
PROCEDURE TRANSI(ORDER,RANGE,DIFFEQ,CONDN,XSUB,MAXSUB)
    INTEGER VALUE ORDER,MAXSUB
    RATIONAL ARRAY RANGE,XSUB
    LONG ALGEBRAIC DIFFEQ
    LONG ALGEBRAIC ARRAY CONDN

    # THIS PROCEDURE TRANSFORMS AN ODE (AND ASSOCIATED
    # CONDITIONS) FROM AN ARBITRARY FINITE RANGE TO THE
    # INTERVAL (-1,1).
    # EXTERNAL VARIABLES REQUIRED: XX,XDY,XDYX.
    # DECLARATION IN CALLING PROCEDURE: NONE.

    INTEGER I,J
    RATIONAL A,B
    LONG ALGEBRAIC P,Q
    LONG ALGEBRAIC ARRAY UNLIST,VALIST
    EXTERNAL LONG ALGEBRAIC XX
    EXTERNAL LONG ALGEBRAIC ARRAY XDY,XDYX

    A = RANGE(1)
    B = RANGE(2)
    P = 2/(B-A)

    # PERFORM TRANSFORMATION ON DIFFEQ.
    UNLIST = (XX)
    VALIST = (((B-A)*XX + (B+A))/2)
    Q = P
    DO I = 1,ORDER
        UNLIST = (UNLIST,XDY(I))
        VALIST = (VALIST,Q*XY(I))
        Q = Q*P
    DOEND
    DIFFEQ = DIFFEQ(UNLIST=VALIST)

    # PERFORM TRANSFORMATION ON ASSOCIATED CONDITIONS.
    UNLIST = 0$0; VALIST = 0$0
    Q = P
    DO I =1,ORDER-1
        DO J = 1,MAXSUB
            UNLIST = (UNLIST,XDYX(I,J))
            VALIST = (VALIST,Q*XYX(I,J))
        DOEND
        Q = Q*P
    DOEND

    DO I = 1,ORDER
        CONDN(I) = CONDN(I)(UNLIST = VALIST)
    DOEND
    Q = (2*XX - (B+A))/(B-A)
    DO I = 1,MAXSUB
        XSUB(I) = Q(XX=XSUB(I))
    DOEND
    RETURN
PROCEDURE TRANS2(RANGE,P,NEWP)
   RATIONAL ARRAY VALUE RANGE
   LONG ALGEBRAIC VALUE P
   LONG ALGEBRAIC NEWP

# THIS PROCEDURE TRANSFORMS THE POLYNOMIAL P FROM
# INTERVAL (-1,1) TO AN ARBITRARY (GIVEN) FINITE
# RANGE, GIVING NEWP.
# EXTERNAL VARIABLES REQUIRED: XX
# DECLARATION IN CALLING PROCEDURE: NONE.

RATIONAL A,B
LONG ALGEBRAIC Z
EXTERNAL LONG ALGEBRAIC XX

A = RANGE(1)
B = RANGE(2)
Z = (2*XX - (B+A))/(B-A)

NEWP = P(XX=Z)

RETURN
END # END OF PROCEDURE TRANS2.
PROCEDURE TRUNC(P, DEGREE, X)
   VALUE DEGREE
   LONG ALGEBRAIC VALUE P, X

PROCEDURE TRUNCATES A POLYNOMIAL P TO A GIVEN
DEGREE IN X.

PACKAGE PROCEDURES REQUIRED: TCS, CHFORM, EVAL.
DECLARATION IN CALLING PROCEDURE: LONG ALGEBRAIC ALTRAN TRUNC.

INTEGER DEGP
LONG ALGEBRAIC ARRAY B
LONG ALGEBRAIC ALTRAN EVAL, CHFORM
LONG ALGEBRAIC ARRAY ALTRAN TCS

P = DEG(P, X)
DEGP <= DEGREE) RETURN(P)
   = TCS(CHFORM(P, DEGP, X), DEGREE)
   EVAL(B, DEGREE, X)
RETURN(P).
END OF PROCEDURE TRUNC.

PROCEDURE TPOWS(MAXDEG, X)
   VALUE MAXDEG.
   LONG ALGEBRAIC VALUE X

THIS PROCEDURE RETURNS THE CHEBYSHEV POLYNOMIALS OF DEGREE
NOT EXCEEDING MAXDEG. THE POLYNOMIALS ARE IN THE
INDEPENDENT VARIABLE X.
DECLARATION IN CALLING PROCEDURE: LONG ALGEBRAIC ARRAY ALTRAN TPOWS.

LONG ALGEBRAIC ARRAY (0: MAXDEG) A
LONG ALGEBRAIC R = 1, S, Y, T
INTEGER I

X = Y
S = Y
A(0) = R
IF (MAXDEG == 0) RETURN(A)
A(1) = S
IF (MAXDEG == 1) RETURN(A)
I = 2, MAXDEG
T = 2*Y*S - R
A(I) = T
A(I) = S
S = T
END
RETURN(A)
END OF PROCEDURE TPOWS.
4. References


