NASA Technical Memorandum 83180

An Implementation of the Programming Structural Synthesis System (PROSSS)

James L. Rogers, Jr.,
Jaroslaw Sobieszczanski-Sobieski,
and Rama B. Bhat

DECEMBER 1981

NASA Technical Memorandum 83180

An Implementation of the Programming Structural Synthesis System (PROSSS)

James L. Rogers, Jr., and Jaroslaw Sobieszczanski-Sobieski
Langley Research Center
Hampton, Virginia

Rama B. Bhat
The George Washington University
Joint Institute for Advancement of Flight Sciences
Langley Research Center
Hampton, Virginia

National Aeronautics
and Space Administration

**Scientific and Technical
Information Branch**

1981

CONTENTS

TABLES

FIGURES

1 INTRODUCTION

Numerous approaches have been documented for combining optimization techniques with an analysis capability (e.g., refs. 1 to 3). The approach documented in this paper is a particular implementation of the method for combining analysis and optimization techniques with applications to structures (ref. 4). This method, called the programming structural synthesis system (PROSSS), combines a large, general purpose, finite element program for structural analysis, SPAR (ref. 5), with a large, general purpose, optimization program, CONMIN (ref. 6) and several, small, problem-dependent FORTRAN programs and subroutines which must be written by the user to interface the analysis and optimization programs. All of the programs are connected by a network of control cards in the standard, Control Data Corporation CYBER Control Language (CCL), documented in reference 7. Familiarity with the theory behind this method (ref. 4) and with the software (documented in refs. 5 to 7) is a prerequisite for understanding the remainder of this document.

This particular implementation of PROSSS is only the first step in a series of implementations. Other implementations are intended to give the user easier access to intermediate results and more control over the flow of the problem, as well as a capability for interactive modeling and data generation (ref. 8). Another implementation includes incorporating PROSSS entirely within the Engineering Analysis Language (EAL, ref. 9) computer program to simplify the maintenance, control, and data management aspects.

This paper describes a particular implementation of PROSSS. First, an overview is given which explains PROSSS in general with respect to this implementation. The second section describes how the input data are prepared. Next, each component of the system is explained in detail. These components include options, procedures, programs and subroutines, and data files used in this implementation. Finally, an example exercise for each option is given to allow the user to anticipate the type of results which might be expected. The appendixes contain annotated listings and flowcharts to clarify the descriptions of the components of the system presented within the body of this paper.

The purpose of this paper is to demonstrate one method for implementing a flexible software system combining large, complex programs with control language and small, user-supplied, problem-dependent programs. It is not intended to be a self-contained user's guide for PROSSS.

Identification of commercial products in this report is used to adequately describe the model. The identification of these commercial products does not constitute official endorsement, expressed or implied, of such products or manufacturers by the National Aeronautics and Space Administration.

2 OVERVIEW OF THIS IMPLEMENTATION OF PROSSS

This implementation of the programming structural synthesis system (PROSSS) combines a general purpose, finite element computer program for structural analysis (SPAR), a state-of-the-art optimization program (CONMIN), and several user-supplied, problem-dependent computer programs. All of the programs are connected by the standard CCL. The results are flexibility of the optimization procedure organization and versatility of the formulation of constraints and design variables.

A flowchart of the analysis-optimization process for this implementation is shown in figure 1. The process results in a minimized objective function, typically

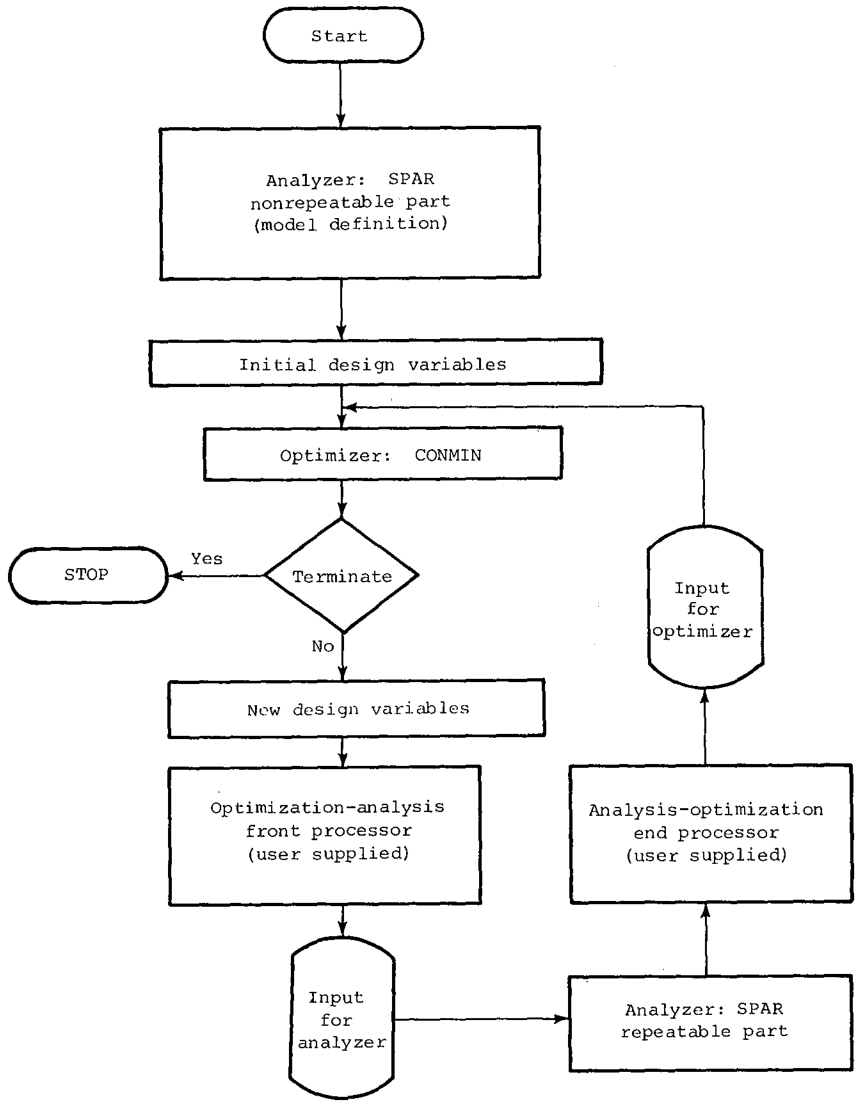


Figure 1.- Basic flow of PROSSS.

the mass defined in terms of a set of design variables, such as cross-sectional dimension of a structural member. This member is subject to a set of constraints such that stress must be less than some allowable value. Notice that the analysis and optimization programs are executed repeatedly by looping through the system until the process is stopped by a user-defined termination criterion. This part of the system is referred to as the repeatable part of PROSSS. However, some of the analysis, such as model definition, need only be done one time and the results saved for future use. This analysis is performed outside of the loop and referred to as the nonrepeatable part of PROSSS. The user must write some small FORTRAN programs (e.g., front processor and end processor) to interface between the analysis and optimization programs. The front processor converts the design variables output from the optimizer into a suitable format for input into the analyzer. The end processor retrieves the behavior variables (i.e., stresses or deflections due to loads) from the analysis program, evaluates the objective function and constraints, and optionally retrieves and evaluates their gradients. These quantities are output in a format suitable for input into the optimizer. These user-supplied programs are problem dependent, because they depend primarily upon the finite elements being used in the model.

Five options for organizing optimization procedures by combining nonlinear or piecewise linear programming methods with analytical or finite difference gradients are shown in table I. Each option is controlled by a complex CCL sequence of commands. These commands are modularized in the form of procedure files and perform the

TABLE I.- OPTIONS FOR OPTIMIZATION

Method	Gradients computed		
	In CONMIN by	External to CONMIN by	
	Finite difference	Finite difference	Analytical
Nonlinear programming	1.1	1.2	1.3
Piecewise linear programming	N/A	2.2	2.3

functions of executing programs in certain sequences, such as if-test branching, looping, and manipulating permanent and temporary files. Each procedure file is written using structured programming techniques to aid in readability.

Although presented in the context of structural analysis, the same system concept could be used for aerodynamic optimization of a wing, if SPAR were replaced by an analyzer with a capability for computational aerodynamics. The system is intended to be used in the following three basic ways:

(1) as a research tool for the development of optimization techniques that will interface with an efficient analysis program

(2) as a research tool for testing new analysis techniques that will interface with an efficient optimization program

(3) as an application tool that can be adapted to a wide range of problem types

3 STEPS FOR IMPLEMENTING PROSSS

The user has access to all of the files containing the components of PROSSS (except SPAR and CONMIN), so that he can modify its programs, procedure files, and flow organization. This capability for component modification makes PROSSS suitable for use as a research test bed. The following step-by-step description explains how to start a new implementation using SPAR as the analyzer and CONMIN as the optimizer. (Details of the files discussed are given in sec 4.)

1. Write a SPAR runstream (see sec 4.4.2) for the entire problem, execute it, and verify the results. Save the SPARLA (see sec 4.4.3) as a permanent file.

2. Divide the SPAR runstream into the nonrepeatable and repeatable parts. The nonrepeatable part contains the TAB (ref. 5) and ELD processors while the repeatable part contains the rest. Execute the nonrepeatable part and save its SPARLA as a permanent file with a different name than the one used in step 1.

3. Write a front processor and its input file CNMNIO. The input file contains the design variables yielding the same structural parameters (behavior variables) as those used as SPAR input in step 1. The front processor reads these design variables and outputs them in a form suitable for input into the SPAR TAB processor. Execute the front processor by using the procedure file PRCFPXX and verify the output file SPFPOUT.

4. Execute the procedure files PRCFPXX and PRCANAL in this order. SPAR output should agree with that obtained in step 1. Save the SPARLA from this execution as a permanent file with name different from those used in steps 1 and 2.

5. Write an end processor and its input data file ENDN. The end processor converts data output by SPAR on SPARLA in step 4 into objective function and constraint data for input into CONMIN. Execute the end processor by using the procedure file PRCEPXX and the SPARLA from step 4 as input. Verify the results on output file CNMNIO.

6. Execute the procedure files PRCFPXX, PRCANAL, PRCEPXX in this order. Verify that the CNMNIO file is the same as that obtained in step 5.

7. If the gradients are to be computed analytically, perform the following additional steps:

(a) Write input files INPT and, if needed, CONS required by the procedure file PRCGRDS, and rerun step 2 by using these files.

(b) Execute the procedure file PRCGRDS using the SPARLA saved in step 4, as input. Save the SPARLD as a permanent file. Verify the SPAR output containing the desired gradient values.

(c) Expand the end processor written in step 5 to handle the gradients available in the SPARLD saved in step 7(b). Execute the end processor and verify its output file CNMNIO to see that its contents are augmented by the gradient values.

(d) Execute the procedure files PRCFPXX, PRCANAL, PRCGRDS, and PRCEPXX in this order and verify that file CNMNIO is the same as that obtained in step 7(c).

8. Write an input file PCNPR containing the CONMIN control parameters. Set ITMAX = 1 (Maximum iterations = 1).

9. Write a file PSTRT containing the starting values of the design variables (same as those used in step 3). Save this file as a permanent file.

10. Select an execution option.

11. In program CONMS1 or CONMS2 (EVALS and FSDS, if applicable), change the dimension statements as required and recompile the program.

12. Write an input file for PROSCRS to convert the general names in the option files (app. A) to specific, problem-dependent names.

13. Execute PROSCRS and verify that the output file CNMNIO is the same as that obtained in step 6 or 7(d).

14. Relax the ITMAX = 1 restriction in the CONMIN control parameter file PCNPR. Proceed with optimization. If option 2.2 or 2.3 (table I) is chosen, allow completion of one linear stage and verify its results before continuing.

15. When the optimization is underway, periodically inspect the CONOUT file for acceptability of the direction the process is taking.

After step 15 has been completed, only the input data files should be changed for a different application to a problem of the same class. In this way, PROSSS is executed like a "black box" because PROSSS is set up and debugged for a specific type of application. All of its components, except the problem-dependent input data files, are protected from unauthorized access. The user thus retains the freedom to change input data as he studies a class of problems, but he cannot change the inner workings of the system. Such restricted use is desirable in production applications, because it facilitates the separation of responsibilities of a support staff specialist from those of the engineer user.

Using PROSSS as a black box is very simple. The user must create an input file PROSSIN for a small program called PROSCRS. (See step 12.) This input file contains the specific names and numbers to replace the general names in the option files (app. A) to solve a particular problem (see sec 4.1). The first column of names, each beginning in column 1, are the general names and the second column of names and numbers, each beginning in column 11, are the specific names and numbers supplied by the user. The first name in the input file must be the option number, thereafter the order is not important. The following are the general names that are to be replaced:

POPT	option number (11, 12, 13, 22, or 23)
NONREPT	1, if nonrepeatable part to be executed; 0, otherwise
NRRS	name of runstream input to SPAR for nonrepeatable part, can be omitted if NONREPT is 0
FUSD	1, if fully stressed design is to be used for optimization program; 0, otherwise
FSDSUB	file containing fully stressed design subroutines to be appended to main program, can be omitted if FUSD is 0
CONMIN	file containing main program for optimization

ENDP1	file containing end processor program
FRNT	file containing front processor program
ENDN	file containing input to end processor
PCNPR	file containing input to optimization program
PSTRT	file containing starting values for design variables
PCONRST	file containing restart values for optimization program, a transfer file
PCNMNIO	file containing data transferred to/from the optimization program
SAVCOUT	file containing cumulative output listing from the optimization program
NSPARLA	SPAR library saved from nonrepeatable part
NGRS or RGS	spar runstream for repeatable part
FLENGTH	field length (octal) required for SPAR execution
BLK	file containing objective function and constraint data, a transfer file
SAVSPLD	SPAR library saved from repeatable part
CNT	file containing testing information for termination criterion
CONS	input file of constants such as the cross-sectional areas of beams for user-supplied subroutines to analytically calculate the gradients and the front processor

The following names can be omitted if options 1.3 or 2.3 (use of analytical gradients) are not chosen:

SUBS	n, the number of subroutines supplied by the user to calculate analytical gradients; one subroutine is needed for each element type containing more than one design variable
BINDEPB	file containing 2n subroutines supplied by the user for calculating analytical gradients; can be omitted if n = 0
INPT	input file of control parameters for calculating analytical gradients

Once the input file, PROSSIN, has been created, it is input into the PROSCRS program. PROSCRS creates three output files: a file of control cards; a file of Text Editor commands (ref. 10) to replace the general names with the specific names; and a file of Text Editor commands (ref. 10) to remove unwanted blanks in the preceding file. The control file is rewound and executed after PROSCRS has completed. The control file performs the following four functions: gets the PROSOPT file (see sec 4.1) and copies the correct option onto file OPTION; edits the file containing edit commands to remove the blanks; edits the OPTION file to change general names to specific names; and starts the option executing.

The program header card for PROSCRS is as follows:

PROGRAM PROSCRS (TAPE8, TAPE9, TAPE10, TAPE11)

where

TAPE8 is the input file of general and specific names

TAPE9 is the output file of control cards

TAPE10 is the output file of edit commands to change from general names to specific names

TAPE11 is the output file of edit commands to remove blanks

Listings of PROSCRS, the input files, the control card file, and the edit file also are presented in appendix A.

4 COMPONENTS OF THE SYSTEM

PROSSS is composed of a system of files consisting of four primary components: option files, procedure files, program files, and data files. Each file is explained in detail with annotated examples listed in the appendixes. For those files which are problem dependent and supplied by the user, the descriptions are given with emphasis on the way the files interface with each other.

4.1 Option Files

There are five option files in PROSSS (table I): options 1.1, 1.2, 1.3, 2.2, and 2.3. Options 1.1, 1.2, and 1.3 use nonlinear mathematical programming, while options 2.2 and 2.3 use piecewise linear programming. Option 1.1 uses gradients calculated inside CONMIN. Options 1.2 and 2.2 use gradients calculated by the finite difference method, while options 1.3 and 2.3 use analytically computed gradients. Five procedure files, one for each of the five options, exist on file PROSOPT. These procedure files consist of a sequence of control cards in CCL (ref. 7). Listings and flowcharts of each option are shown in appendix B. The BEGIN cards in these procedure files control the sequence of execution of the procedure files described in section 4.2. No option procedures have any key words associated with them. Each procedure does, however, have many general names within it. These general names are replaced by specific names (using the Text Editor (ref. 10)) before the option begins executing. These names are described in detail in sections 4.2 to 4.4, and the replacement process was previously explained in section 3. This process was chosen over key word substitution because one BEGIN card (ref. 7) would not hold all the necessary key words and no continuation card is allowed.

4.2 Procedure Files

PROSSS is controlled by nine procedure files, all of which are located on a file named PROSPRC. The specific functions performed by the procedures are (1) nonrepeatable model definition (PRCNRPT), (2) and (3) initialization (PRCINIT and PRCGETF), (4) front processing (PRCFPXX), (5) optimization (PRCOPTM), (6) analysis

(PRCANAL), (7) analytical gradient calculation (PRCGRDS), (8) end processing (PRCEPXX), and (9) printing output (PRCEND). Each procedure consists of a sequence of control cards in CCL (ref. 7). A procedure is called by a BEGIN statement. Commented listings of each of the nine procedure files are shown in appendix C. Each procedure begins with a header card containing a list of key words, if needed. Key word substitution allows the user to substitute key words in the procedure body with parameters specified on the BEGIN statement. Table II provides a quick reference for each procedure file by giving the file name, the purpose, and the calling sequence.

TABLE II.- PROCEDURE FILES IN PROSSS

File name	Purpose	File call command
PRCNRPT	Executes nonrepeatable analysis program	.PROC,PRCNRPT,NROPT,NRRS,FLX,I,NRLA.
PRCINIT	Assembles user supplied programs and subroutines	.PROC,PRCINIT,OP,A,B,NSUB,C,FSD,FSUB.
PRCGETF	Retrieves remainder of files needed for execution	.PROC,PRCGETF,OP,F,E,CN,S,I,C,CT,RS,RGS.
PRCFPXX	Executes the front processor	.PROC,PRCFPXX.
PRCOPTM	Executes the optimization program	.PROC,PRCOPTM,C,D,F.
PRCANAL	Executes the repeatable analysis program	.PROC,PRCANAL,NRLA,FLX,SAUELD.
PRCGRDS	Creates a runstream for input to the analysis program, then executes the analysis program and, optionally, a post processor to find analytical gradients	.PROC,PRCGRDS,NSUB,SAUELD.
PRCEPXX	Executes the end processor	.PROC,PRCEPXX,BLK.
PRCEND	Outputs important files	.PROC,PRCEND.

4.2.1 Nonrepeatable Model Definition: PRCNRPT

This procedure PRCNRPT creates a SPAR library of the joint and element information for the finite element model (app. C). If analytical gradients are required, the derivatives of the stiffness and mass matrices with respect to the design variables are also computed and stored on the library. The procedure header card is as follows:

.PROC,PRCNRPT,NROPT,NRRS,FLX,I,NRLA.

where PRCNRPT is the name of procedure file, with key words:

NROPT option number

NRRS nonrepeatable SPAR runstream

FLX field length (octal)

I input file for analyical gradient calculation (see INPT)

NRLA SPAR library

4.2.2 Initialization: PRCINIT,PRCGETF

Two procedure files, PRCINIT and PRCGETF, are used in the initialization process. The first, PRCINIT, gets the programs required for a particular option (app. C). Some programs must be assembled using various user-supplied main programs and/or subroutines, the names of which are passed through the header card. The procedure header card is as follows:

.PROC,PRCINIT,OP,A,B,BB,NSUB,C,FSD,FSUB.

where PRCINIT is the name of procedure file, with key words:

OP option number

A main program for optimization

B main program for end processor with no gradients

BB main program for end processor with gradients

NSUB number of user-supplied subroutines for analytical gradient calculation

C file containing all user-supplied subroutines for analytical gradient calculation. The first NSUB subroutines are combined with program GNGRDRS. The second NSUB subroutines are combined with program DRVSTRS. Each program is a physical record (not needed if NSUB is zero)

FSD 1, implies fully stressed design is required; 0, implies no fully stressed design is required

FSUB file containing fully stressed design subroutines (not needed if FSD is zero)

The second procedure file PRCGETF, used in the initialization process retrieves the remainder of the files required for executing a particular option (app. C). The procedure header card is as follows:

.PROC,PRCGETF,OP,F,E,CN,S,I,C,CT,RS,RGS.

where PRCGETF is the name of procedure file, with key words:

OP option number

F front processor program (see FPROC)

E input file to end processor (see ENDN)

CN input file to optimization program (see PCNPR,CONPAR)

S input files to optimization program (see PSTRT,STARTX)

I input file for analytical gradient calculation (see INPT)

C input file of constants for analytical gradient calculation (see CONS)

CT input file of constants for optimization program (see CNT)

RS SPAR runstream (no gradients, see NGRS)

RGS SPAR runstream (gradients, see RGS)

4.2.3 Front Processing: PRCFPXX

The procedure file PRCFPXX executes the front processor (app. C). The procedure header card is as follows:

.PROC,PRCFPXX.

where PRCFPXX is the name of procedure file. There are no key words.

4.2.4 Optimization: PROCOPTM

The procedure file PRCOPTM executes the optimization program (app. C). The procedure header card is as follows:

.PROC,PRCOPTM,C,D,F.

where PROCOPTM is the name of procedure file, with key words:

C restart data for optimization program (see PCONRST)

D transfer data to/from optimization program (see PCNMNIO)

F cumulative output from optimization program (see SAVCOUT)

4.2.5 Analysis: PRCANAL

The procedure file PRCANAL merges the output file (SPFPOUT) from the front processor into the SPAR runstream file (SPARRS) and executes the SPAR analysis program. (See app. C.) The initial SPAR input and output are saved for later listing. The procedure header card is as follows:

.PROC,PRCANAL,NRLA,FLX,SAVELD.

where PRCANAL is the name of procedure file, with key words:

NRLA SPAR library from nonrepeatable part

FLX field length (octal)

SAVELD SPAR library for use by end processor

4.2.6 Analytical Gradient Calculations: PRCGRDS

This procedure file, PRCGRDS, creates a SPAR runstream and then uses SPAR and a postprocessor (see sec 4.3.6.3) to SPAR to calculate stress derivatives when forces and moments and derivatives of forces and moments must be converted to stresses and stress derivatives. (See app. C.) The procedure header card is as follows:

```
.PROC,PRCGRDS,NSUB,SAVELD.
```

where PRCGRDS is the name of procedure file, with key words:

NSUB number of user-supplied subroutines for analytical gradient calculations

SAVELD SPAR library for use by end processor

4.2.7 End Processing: PRCEPXX

This procedure file, PRCEPXX, executes the end processor. (See app. C.) The procedure header card is as follows:

```
.PROC,PRCEPXX,BLK.
```

where PRCEPXX is the name of procedure file, with key word:

BLK transfer data file (see BLOCK)

4.2.8 Printing Output: PRCEND

This procedure file, PRCEND, prints CONMIN and SPAR output files and SPAR runstreams. (See app. C.) The procedure header card is as follows:

```
.PROC,PRCEND.
```

where PRCEND is the name of procedure file. There are no key words.

4.3 Program and Subroutine Files

PROSSS uses the following six types of programs during the analysis-optimization process: analysis (SPAR), optimization (CONMIN), front processor (FPROC), end processor (EPROC), control programs, and analytical gradient programs. The analysis program SPAR and the CONMIN subroutine library are not given in the appendixes because they are standard for the system and are documented in references 5 and 6. The user does not have to create either SPAR or the CONMIN subroutine library. Sample listings for all other programs, including the main driver program for the CONMIN subroutine library, are shown in appendix D. Table III provides a quick reference for each of the programs by giving the options and procedures in which the program is used, whether or not the program must be created by the user, and a brief comment about the function of the program.

TABLE III.- PROGRAM AND SUBROUTINE FILES IN PROSSS

Program (P) or subroutine (S) name	User created	Option(s)	Procedure(s)	Comment
SPAR (P)	No	All	PRCNRPT,PRCANAL, PRCGRDS	Finite element structural analysis program and its data management system.
CONMIN (S) subroutines	No	All	PRCOPTM	Problem independent set of optimization subroutines.
CONMS1 (P)	Yes	1.1, 1.2, 1.3	PRCOPTM	Driver program for CONMIN optimization. Has problem dependent dimensioned variables in blank common. (Nonlinear)
CONMS2 (P)	Yes	2.2, 2.3	PRCOPTM	Driver program for CONMIN with linear extrapolation in lieu of full analysis. Has problem dependent dimensioned variables in blank common. (Piecewise linear)
FPROC (P)	Yes	All	PRCFPXX	Creates a file of design variables in a format meaningful for input into SPAR.
EPROC (P)	Yes	All	PRCEPXX	Creates a file of objective function, constraints, and optionally, their gradients in a format meaningful for input into CONMIN.
EVALS (P)	Yes	1.2, 2.2, 2.3	None, called from option file	Computes finite difference gradients. Dimensioned variables are problem dependent.
FSDS (P)	Yes	All	PRCOPTM	Replaces CONMIN. Modifies values of initial design variables by means of fully stressed design techniques. Dimensioned variables are problem dependent. Can be used as an inexpensive means for finding starting design variables.
FSDSUBS (S)	Yes	All	PRCOPTM	Subroutines used in conjunction with FSDS.
SELECTS (P)	Yes	1.2, 1.3	None, called from option file	Determines if constraints are active or violated.
RERITES (P)	No	1.2	None, called from option file	Copies design variables from one file to another.
BLDELDS (P)	No	1.3, 2.3	PRCNRPT	Creates an input file for computing gradients in the analysis program in the nonrepeatble part.
GNGRDRS (P)	No	1.3, 2.3	PRCGRDS	Creates an input file for computing gradients in the analysis program in the repeatable part.
DKDVE21 (S) DKDVE22 DKDVE23 DKDVE43	Yes	1.3, 2.3	PRCGRDS	Creates part of the input to compute gradients in the analysis program in the repeatable part. Computes derivatives of the mass and stiffness matrices with respect to the design variables. These subroutines are used with GNGRDRS.
DRVSTRS (P)	No	1.3, 2.3	PRCGRDS	Converts forces and moments and derivatives of forces and moments to stresses and stress derivatives.
BMSTRS (S) PLTSTRS	Yes	1.3, 2.3	PRCGRDS	Computes stresses and stress derivatives for beam and plate elements. (Subroutines)

4.3.1 Analysis Program: SPAR

4.3.1.1 Overall characteristics.- SPAR is a system of processors which perform linear, finite element, structural analysis (ref. 5). It can compute static deflections, stresses, vibration frequencies and modes, dynamic responses, buckling loads, and mode shapes. Shown in figure 2 is the organization of the SPAR processors

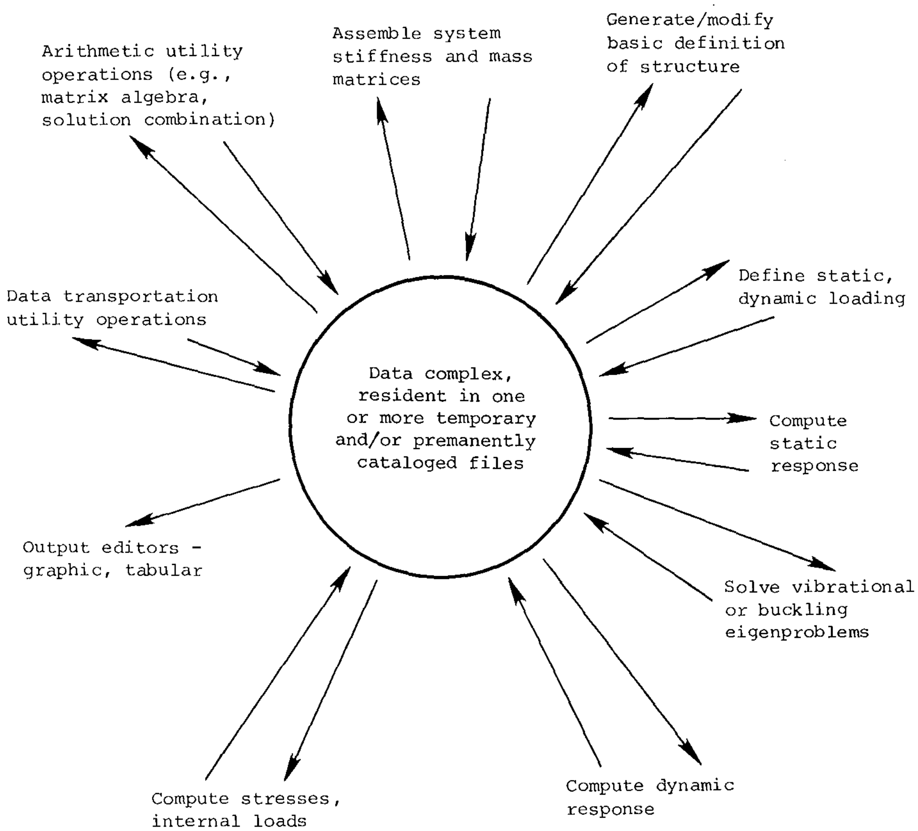


Figure 2.- SPAR system organization.

and data flow. The individual processors communicate with a central body of information known as the data complex. The data complex consists of one or more libraries, which contain the data sets output from the different processors. Each data set has a specific identifying name by which any processor can access it whenever it is required as input for particular computations. A list of the SPAR processors and their functions is given in table IV. The numeral next to the processor names refers to the rows of table V. In table V certain operations performed in SPAR are defined and the operation is broken down according to the type of variable and whether or not the operation occurs in the repeatable or nonrepeatable part of PROSSS. Taken together, tables IV and V show a division of the SPAR processors between nonrepeatable and repeatable parts. Processors without numerals are utilities for functions such as plotting, printing, eigenvalue extraction, etc.

SPAR executes on a processor-by-processor basis. Each processor is executed by a separate explicit command. A runstream consisting of a string of such commands interlaced with the numerical input data is written by the user for a specific problem. Runstreams are described in section 4.4.2. This modularity of SPAR organization and execution makes it well suited for optimization applications.

TABLE IV.- SPAR PROCESSORS

Operation number in table V	Name	Processor Function
1, 2, 3, 6	*TAB	Creates data sets containing tables of joint locations, section properties, material constants, etc.
5, 7	ELD	Defines finite elements making up model
5, 6, 7	E	Generates sets of information for each element, including connected joint numbers, geometrical data, material and section property data
8, 9	EKS	Adds stiffness and stress matrices for each element to set of information produced by E processor
10	TOPO	Analyzes element interconnection topology and creates data sets used to assemble and factor system mass and stiffness matrices
10	K	Assembles unconstrained system stiffness matrix in sparse format
11	M	Assembles unconstrained system mass matrix in sparse format
15	KG	Assembles unconstrained system initial-stress (geometric) stiffness matrix in sparse format
12	INV	Factors assembled system matrices
4	EQNF	Computes equivalent joint loading associated with thermal, dislocational, and pressure loading
13	SSOL	Computes displacements and reactions due to loading applied at joints
14, 16	GSF	Generates element stresses and internal loads
	PSF	Prints information generated by GSF processor
17	EIG	Solves linear vibration and bifurcation buckling eigenproblems
	DR	Performs dynamic response analysis
	SYN	Produces mass and stiffness matrices for systems comprised of interconnected substructures
	STRP	Computes eigenvalues and eigenvectors of substructured systems
4	AUS	Performs array of matrix arithmetic functions and is used in construction, editing, and modification of data sets
	DCU	Performs array of data management functions including display of table of contents, data transfer between libraries, changing data set names, printing data sets, and transferring data between libraries and sequential files
	VPRT	Performs editing and printing of data sets which are in form of vectors on data libraries
	PLTA	Produces data sets containing plot specifications
	PLTB	Generates graphical displays which are specified by PLTA processor

*This processor can operate in an update mode in the repeatable part. (See sec 3.1 in ref. 5.)

TABLE V.- NONREPEATABLE (N) AND REPEATABLE (R) PARTS IN FINITE ELEMENT ANALYSIS BASED ON DISPLACEMENT METHOD

Number	Operation	Type of variable			
		Cross-sectional dimensions	Nodal coordinates	Connectivity	Element type
1	Define material properties	N	N	N	N
2	Define coordinates of nodes	N	R	N	N
3	Define each node's degrees of freedom	N	N	N	N
4	Define loads	N	N	N	N
5	Define types of elements	N	N	N	R
6	Define cross-sectional dimensions	R	N	N	N
7	Define element-node connectivity	N	N	N	R
8	Compute elemental stiffness matrices	R	R	R	R
9	Compute elemental mass matrices	R	R	R	R
10	Assemble structure stiffness matrix	R	R	R	R
11	Assemble structure mass matrix	R	R	R	R
12	Decompose stiffness matrix	R	R	R	R
13	Compute displacements	R	R	R	R
14	Compute loads on elements	R	R	R	R
15	Assemble structure geometrical stiffness matrix	R	R	R	R
16	Compute stresses	R	R	R	R
17	Compute eigenvalues and eigenmodes for vibration and/or buckling	R	R	R	R

4.3.1.2 Analytical calculation of gradients.- Gradients can be calculated in SPAR by using runstreams established specifically for this purpose. The general analysis capability of SPAR is augmented by this runstream to calculate structural response derivatives for static displacements and stresses. The runstream is dependent upon the types of elements used to model the structure. This method of calculating gradients for static analysis is represented in PROSSS by procedure files discussed in section 4.2.6.

4.3.1.3 SPAR data storage and retrieval.- A unique feature of SPAR relevant to PROSSS organization is its set of data libraries. A group of data sets can be assembled to form a named library. Subroutines documented in reference 11 are available to store and retrieve the SPAR library data sets. These subroutines can be executed by FORTRAN CALL statements and, hence, can be used to make the SPAR data storage accessible to non-SPAR FORTRAN programs.

4.3.2 Optimization Program: CONMIN

4.3.2.1 Overall characteristics.- CONMIN is a general purpose, optimization subroutine library capable of solving linear or nonlinear constrained optimization problems. The basic optimization alogrithm is the method of feasible directions. A user's manual describing the program and its execution options (ref. 6) explains all the control parameters used by CONMIN and the CONMIN execution modes.

In CONMIN, the objective function and the constraint functions must be continuous functions of the design variables. The design variables must also be continuous. Therefore, only these two types of optimization problems can be handled by CONMIN and, consequently, by PROSSS.

The CONMIN program organization, shown in figure 3, consists of a main program and the CONMIN subroutine library. The main program reads the initial values of

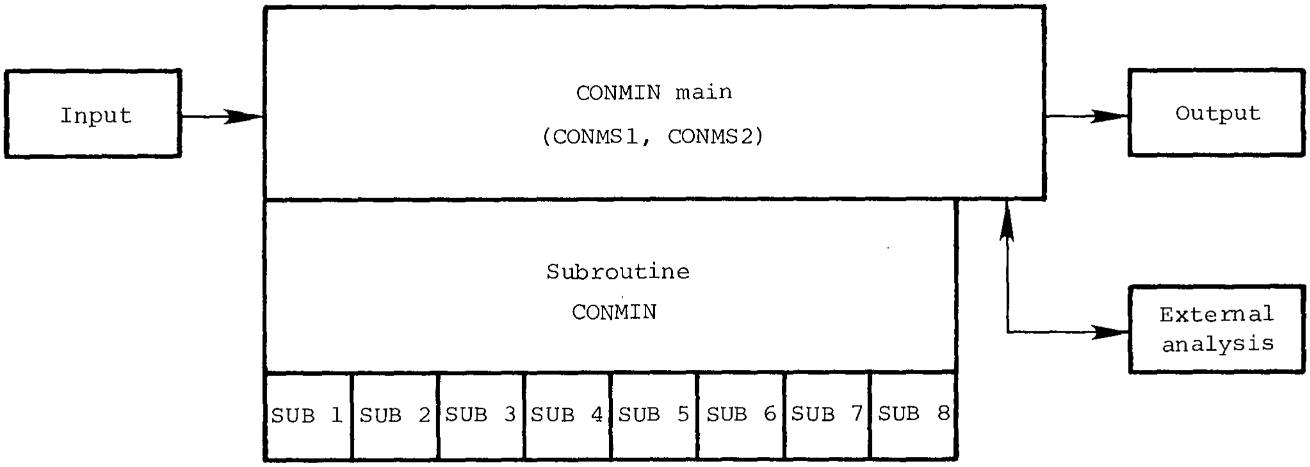


Figure 3.- CONMIN program organization.

design variables and CONMIN control parameters. The computation of the constraints, the objective function, and the gradients of both can be carried out in the main program if the problem is computationally small. In PROSSS, this mode of operation is used for options 2.2 and 2.3 (table I) by taking advantage of the simplicity and speed of the linear extrapolation procedure. If the problem is large, the computation of objective function, constraints, and gradients is executed externally by stopping the main program, performing the external analysis, and restarting the main program. This mode of operation is used in all other options.

Actual optimization is carried out by the CONMIN subroutine library and is controlled by a set of parameters input through the main program. The CONMIN subroutine library also contains a set of the termination criteria. The user can select a criterion from that set, and establish the numerical values associated with that criterion by means of the input control parameters.

In PROSSS, the optimizer CONMIN appears in the form of

1. subroutine CONMIN and a set of associated subroutines

2. two versions of CONMIN main programs
   (a) CONMS1, to be used with options 1.1, 1.2, 1.3
   (b) CONMS2, to be used with options 2.2, 2.3

4.3.2.2 Program: CONMS1.- The program is problem independent except for the blank common statement. (See app. D.) For a new application, the arrays in this statement must be inspected and adjusted according to reference 6, if necessary, to accommodate the problem size. The source CONMS1 must then be recompiled by the user, and the binary code is used in execution.

The program functions are

1. read CONMIN control parameters from PCNPR the first time CONMS1 is executed

2. read analyzer output in the second and subsequent executions for file CNMNIO

3. call subroutine CONMIN

4. output new vector of design variables for the optimizer on file CNMNIO

5. stop itself to permit the external analysis

6. write and save all data needed for subsequent restarts on file PCONRST

7. generate a message on file GONOGO to indicate that the nonlinear programming (NLP) be stopped when subroutine CONMIN detects satisfaction of its termination criteria

The program card is as follows:

```
PROGRAM CONMS1 (INPUT,OUTPUT,TAPE8,TAPE7,TAPE9,TAPE11,TAPE10,TAPE5 = INPUT,
                TAPE6 = OUTPUT)
```

where

INPUT	PCNPR
OUTPUT	SAVCOUT
TAPE7	PCONRST
TAPE8	PSTRT
TAPE9	CNMNIO
TAPE10	PASS
TAPE11	GONOGO

4.3.2.3 Program: CONMS2.- The foregoing description of the program CONMS1 applies to CONMS2 with the following differences. (See app. D.)

The program functions are

1. read CONMIN control parameters from file PCNPR

2. read data for linear extrapolation analysis from file BLOCK

3. call subroutine CONMIN

4. execute the linear extrapolation analysis

5. repeat functions 3 and 4 until two changes in the objective function, obtained by comparing results of three consecutive linear stages, are smaller than a prescribed limit and generate a message to CCL that the piecewise linear programming (PLP) outer loop should be stopped

The program card is as follows:

```
PROGRAM CONMS2 (INPUT,OUTPUT,TAPE7,TAPE8,TAPE9,TAPE10,TAPE11,TAPE5 = INPUT,
               TAPE6 = OUTPUT)
```

where

INPUT	PCNPR	Same as CONMS1
OUTPUT	SAVCOUT	
TAPE7	PSTRT	
TAPE8	BLOCK	
TAPE9	CNT	
TAPE10	PASS	Same as CONMS1
TAPE11	GONOGO	

4.3.3 Front Processor Program: FPROC

This program does not exist in the PROSSS until it is coded by the user for a specific problem. The front processor program must be compiled by the user, and the binary code must be stored to be used in the PROSSS execution. The name of the binary file must be supplied in the input file to PROSCRS. (See app. D for an example of a front processor program.)

The program functions are

1. read the design variables output by CONMIN from file CNMNIO
2. calculate behavior variables using the design variables on file CNMNIO
3. output these behavior variables in a format meaningful for SPAR. (See SPFPOUT in sec 4.4.3.)

The program card is as follows:

```
PROGRAM FPROC (INPUT,OUTPUT,TAPE7,TAPE5 = INPUT,TAPE6 = OUTPUT)
```

where

FPROC	program name
INPUT	CNMNIO
OUTPUT	SPFPOUT
TAPE7	CONS

4.3.4 End Processor Program: EPROC

This program must also be coded by the user. (See app. D.) The end processor program must be compiled by the user, and the binary code must be stored to be used in the PROSSS execution. The name of the binary file must be supplied in the input file to PROSCRS.

The program functions are

1. read user-supplied constants defining limits imposed on the behavior variables from file ENDN; these limits are to be used for the constraint function evaluations

2. retrieve the SPAR output data sets (behavior variables and, optionally, their gradients) from the SPAR libraries and store them in arrays

3. extract from these arrays the design variables to be used for the constraint function evaluations and for computing the objective function

4. evaluate the constraints and the objective function and, optionally, their gradients

5. output these quantities in NAMELIST form on file CNMNIO to be read by CONMS1 or CONMS2 main programs (see the LINKE NAMELIST in app. D)

The SPAR libraries are in a special binary format accessible only by two FORTRAN callable subroutines, DAL and FIN, which are used as standard parts of the end processor code to carry out program function 2. These subroutines are documented in reference 11. Both routines are used in SPAR and must be retrieved from the SPAR relocatable subroutines and loaded with the end processor.

The end processor program card is as follows:

```
PROGRAM EPROC (INPUT,TAPE6,TAPE8,TAPE5 = INPUT)
```

where

EPROC	program name
TAPE5	ENDN
TAPE6	CNMNIO
TAPE8	BLK

4.3.5 Control Programs

There are four problem independent control codes: EVALS, evaluates the finite difference gradients; FSDS, incorporates the fully stressed design technique; SELECTS, selects active or violated constraints; and RERITES, rewrites the design variables to a new file.

4.3.5.1 Program to compute the finite difference gradients: EVALS.- Its functions are

1. read CONMIN control parameters supplied by user from file PCNPR

2. read the design variables from file PSTRT

3. Stop itself to allow execution of the front processor/SPAR/end processor sequence

4. restart - read (from file CNMNIO) and store the objective function and constraint values

5. increment one design variable

6. restart - reset the design variable incremented in step 5 to its original value

7. using the stored objective function and constraint values, compute the finite difference gradients of these functions for the design variable

8. repeat steps 4 to 7 until all variables have been incremented

The program card is as follows:

```
PROGRAM EVALS (INPUT,OUTPUT,TAPE5 = INPUT,TAPE7,TAPE8,TAPE9,TAPE10,TAPE11)
```

where

TAPE5	PCNPR
TAPE7	PASS
TAPE8	PSTRT
TAPE9	CNMNIO
TAPE10	BLOCK
TAPE11	CHECK

See appendix D for a sample listing.

4.3.5.2 Fully stressed design (FSD) program: FSDS.- The program uses a fully stressed design technique to modify the initial design variables. The program functions are

1. read control parameters supplied by user from file PCNPR

2. read the design variables from file PSTRT

3. stop itself to allow execution of the front processor/SPAR/end processor sequence

4. restart - read values of the stress constraints from file BLOCK

5. change the design variables by FSD technique using the constraint values

6. repeat from 3

The program card is as follows:

```
PROGRAM FSDS (INPUT,OUTPUT,TAPE8,TAPE7,TAPE9,TAPE11,TAPE10,TAPE5 = INPUT,
              TAPE6 = OUTPUT)
```

All files are defined exactly the same as in the program CONMS1. (See app. D for a sample listing.) The program calls a problem independent subroutine FSDSUBS (also in app. D) that carries out FSDS function 5.

4.3.5.3 Program to select the active or violated constraints: SELECTS.- The program functions are

1. read the constraints from file BLOCK

2. determine if the constraint is less than the constraint thickness parameter for linear and side constraints and, thus, active or violated

3. if the constraint is active or violated, the analytical gradient of that constraint is stored and a pointer is set in an array to denote whether or not the constraint is active or violated

The program card is as follows:

```
PROGRAM SELECTS (INPUT,OUTPUT,TAPE5 = INPUT,TAPE6 = OUTPUT)
```

where

TAPE5	BLOCK
TAPE6	CONREST

See appendix D for a sample listing.

4.3.5.4 Program to rewrite the design variables to a different file: RERITES.- The program RERITES copies the design variables from CNMNIO to SO.

The program card is as follows:

```
PROGRAM RERITES (INPUT,TAPE5 = INPUT,TAPE6)
```

where

TAPE5	CNMNIO
TAPE6	SO

See appendix D for a sample listing.

4.3.6 Gradient Programs

There are three problem independent programs, BLDELDS, GNGRDRS, and DRVSTRS, that aid in calculating analytical gradients. The first, BLDELDS, builds a file for input into the analysis program to calculate the derivatives for the mass and stiffness matrices with respect to the design variables in the nonrepeatable part of PROSSS. The second, GNGRDRS, builds a file in the repeatable part for input into the analysis program to compute the derivatives of the stresses and of the forces and moments with respect to the design variables. The third, DRVSTRS, converts the forces and moments and the derivatives of the forces and moments into stresses and derivatives of stresses.

4.3.6.1 Program to create input file for computing gradients in the analysis program in the nonrepeatable part: BLDELDS.- BLDELDS performs the following functions:

1. Read control parameters supplied by the user from file INPT.

2. Read the input model data file for the analysis program from file NRRS.

3. Change all design variables in the file NRRS to unity and place on file RSOUT.

4. Create remainder of file RSOUT to calculate the derivative of the stiffness and mass matrices with respect to the different design variables.

The program card is as follows:

PROGRAM BLDELDS (TAPE5,TAPE23,TAPE20,TAPE21,TAPE22,OUTPUT)

where

TAPE5	INPT
TAPE23	NRRS (after editing)
TAPE20	RSOUT
TAPE21	Scratch file
TAPE22	Scratch file
OUTPUT	Output file for error messages

See appendix D for a sample listing.

4.3.6.2 Program to create input file for computing gradients in the analysis program in the repeatable part: GNGRDRS.- GNGRDRS performs the following functions:

1. Reads control parameters supplied by user from file INPT.

2. Creates the runstream to calculate the derivatives of the stiffness and the mass matrices with respect to the design variables when an element (such as a beam or plate) has more than one contributing factor. The program calls user-supplied subroutine(s) with any of the following names DKDVE21, DKDVE22, DKDVE33, and/or

DKDVE43. These subroutines compute the derivatives of the stiffness matrix with respect to a design variable for a particular element type in SPAR (e.g., E21 or E43). The name(s) used depends upon the elements used in the finite element model. If a subroutine is not used, it remains unsatisfied. Two integer parameters used in naming the created data sets are passed to each subroutine. The first is the counter for the design variable and the second is the number of unconstrained degrees of freedom (fron 1 to 6). A listing of a sample DKDVE21 subroutine is shown in appendix D.

The subroutine card for DKDVE21 is as follows:

SUBROUTINE DKDVE21(NDVJIM,NDF)

where

NDVJIM — the number of the design variables (first, second, third, etc.) for which the derivative of the stiffness and mass matrices with respect to the design variable is being calculated

NDF — number of degrees of freedom per joint squared

3. Create the remainder of the runstream and place on file RSOUT.

The program card is as follows:

PROGRAM GNGRDRS (INPUT,TAPE30,TAPE31,TAPE10,TAPE5 = INPUT,OUTPUT)

where

TAPE5	INPT	
TAPE30	CONS	Used in user-supplied subroutine
TAPE31	CNMNIO	
TAPE10	RSOUT	
OUTPUT	Output file for error messages	

See appendix D for listing of GNGRDRS.

4.3.6.3 Program to convert forces and moments and the derivatives of forces and moments to stresses and stress derivatives: DRVSTRS.- DRVSTRS performs the following functions:

1. Reads control parameters supplied by user from file INPT.

2. Reads forces and moments and the derivatives of forces and moments from file SPARLD, a library created by SPAR.

3. Computes stresses and stress derivatives using a user-supplied subroutine named BMSTRS (for beams) or PLTSTRS (for plates). As before, if a name is not used, it remains unsatisfied. Four integer parameters are passed to BMSTRS. They are (1) a switch and (2) a counter, so certain computations can be skipped if they are not needed; (3) a block counter for accessing an array; and (4) another switch to

determine if the beam is the contributing factor to the stress derivative. The first three parameters are also passed to PLTSTRS. A listing of a sample BMSTRS subroutine is provided in appendix D.

The subroutine card for BMSTRS is as follows:

```
SUBROUTINE BMSTRS (ISW,KCNT,JCNT,IBEAM)
```

where

ISW,KCNT	a switch and a counter used to store certain beam data (e.g., moments of inertia) and make certain computations only on the first time BMSTRS is called from DRVSTRS
JCNT	a counter to find the location in blank common for various data items, depends on the number of beam elements
IBEAM	1, if beam is a contributing factor to stress derivatives; 0, otherwise

4. The stress and stress derivatives are written on SPARLD for processing by the end processor EPROC (sec 4.3.4).

The program card is as follows:

```
PROGRAM DRVSTRS (INPUT,TAPE30,TAPE31,TAPE5 = INPUT,TAPE15,TAPE16,OUTPUT,
                 TAPE6 = OUTPUT)
```

where

TAPE5	INPT	
TAPE30	CONS	Used in user-supplied subroutine
TAPE31	CNMNIO	
TAPE15	Scratch file	
TAPE16	Scratch file	
OUTPUT	Output file for error messages	

See appendix D for a sample listing of DRVSTRS.

4.4 Data Files

There are five types of data files in PROSSS: input, model, transfer, edit, and save. Sample annotated listings of each file are shown in appendix E. Table VI

TABLE VI.- DATA FILES IN PROSSS

File name	Type	Created by	Storage	Problem	Program(s) using file	Comment
PCONPAR, CONPAR	Input	User	Permanent	Dependent	CONMS1,CONMS2, EVALS	Initializes values for optimization program
PSTART STARTX	Input	User	Permanent	Dependent	CONMS1,CONMS2	Initializes design variables for optimization program
PASS	Input	EDPASS1, EDPASS2	Local	Independent	CONMS1,EVALS	Test variable for first pass through system. Created by EDPASS1 file.
INPT	Input	User	Permanent	Dependent	BLDELDS,GNGRDRS, DRVSTRS	Initializes values for gradient calculation
CNT	Input	User	Permanent	Dependent	CONMS1,CONMS2	Objective functions and tolerance for termination test
ENDN	Input	User	Permanent	Dependent	EPROC	Defines limits imposed on design variables in end processor
CONS	Input	User	Permanent	Dependent	GNGRDRS,DRVSTRS	Defines certain constants used in analytical gradient calculation
NRRS	Model	User	Permanent	Dependent	SPAR	Nonrepeatable input runstream to analysis program
SPARRS	Model	User	Permanent	Dependent	SPAR	Repeatable input runstream to analysis program
PCONRST	Transfer	Programs	Local	Dependent	CONMS1,CONMS2, SELECTS	Data saved for subsequent passes through optimization program. Created in optimization program
BLOCK,BLK	Transfer	Programs	Local	Dependent	CONMS1,CONMS2, EVALS,EPROC, SELECTS	Contains objective function and constraint data
PCNMNIO, CNMNIO	Transfer	Programs	Local	Dependent	FPROC,CONMS1, CONMS2,GNGRDRS, DRVSTRS,EPROC, RERITES,EVALS	Transfer design variable, objective function and constraint data between optimization program and front and end processors
CHECK	Transfer	Programs	Local	Dependent	EVALS	Created when analysis has been performed for each design variable combination
GONOGO	Transfer	Programs	Local	Dependent	CONMS1,CONMS2	Created to terminate PROSSS
SPFPOUT	Transfer	Program	Local	Dependent	FPROC,SPAR	Contains updated design variables for input into analysis program
RSOUT	Transfer	Program	Local	Dependent	BLDELDS,GNGRDRS, SPAR	Input runstream to analysis program
SO	Transfer	Program	Local	Dependent	EVALS,RERITES	Design variable information
SPARLA,SPARLB, SPARLC,SPARLD	Transfer	Program	Local	Dependent	SPAR,DRVSTRS, EPROC	Data library created by analysis program
EDPASS1	Edit	External	Permanent	Independent	None	Initializes PASS variable to 1
EDPASS2	Edit	External	Permanent	Independent	None	Reinitializes PASS variable to 1
EDIT1	Edit	External	Permanent	Independent	None	Removes all but element connection data from NRRS file
EDIT2	Edit	External	Permanent	Independent	None	Prepares nonrepeatable RSOUT file for input into analysis program
MERGFP	Edit	External	Permanent	Independent	None	Merges SPFPOUT file into SPARRS file
EDGRDS	Edit	External	Permanent	Independent	None	Prepares repeatable RSOUT file for input into analysis program
SAVCOUT	Save	Program	Permanent	Dependent	CONMS1,CONMS2	Cumulative list of data output from optimization program
NRLA	Save	Program	Permanent	Dependent	SPAR	Data library output from nonrepeatable analysis program
SAVSPLD	Save	Program	Permanent	Dependent	SPAR,DRVSTRS	Data library output from repeatable analysis program

provides a quick reference for each data file by providing the following information about each file:

1. the name of the file

2. the type of file (e.g., input or model)

3. whether the file is created by the user, program, or external to PROSSS; files created external to PROSSS, are created one time, stored on a permanent file, and then accessed by any user independent of the application

4. whether the file is local or permanent

5. whether the file is problem dependent or independent

6. the programs that use this file

7. a brief comment about the file's function

4.4.1 Input Data Files

There are seven input files that primarily provide initialization data to the various programs in PROSSS. (See app. E for sample listings.) These files are as follows:

1. PCNPR (CONPAR) initializes variables used in the optimization program. (Note: PCNPR is the name used in the generalized input files. See app. A.) It is replaced by the specific problem dependent file name before the option is used. CONPAR is the local file name into which the specific file is stored. It is in NAMELIST format with the NAMELIST name CONPAR. A description of the variables can be found in reference 6.

2. PSTRT (STARTX) is also read by the optimization program to initialize the design variables. (See note about PCNPR and CONPAR.) It, too, is in NAMELIST format with the NAMELIST name STARTX.

3. PASS contains one number. In the initialization process, the number is set to 1 using the EDPASS1 file in section 4.4.4, signifying the first pass through the system. After the first pass, the value is changed to 2 and different paths are taken in the programs using PASS as an input variable. Options 1.2 and 2.2 (table I) reset the value to 1 within the system using the EDPASS2 file also described in section 4.4.4. PASS is also in NAMELIST format with the NAMELIST name PASSAGE.

4. INPT defines values used in analytical gradient calculation. INPT has two card types. The first type has the format (6(1X,I4),1X,A4), consists of one card, and contains the following variables:

NOLC	number of load cases
NODV	number of design variables
ISNOLC	new starting number for load cases with derivatives
JOINTS	number of joints in the model

NDF number of degrees of freedom

NOEL number of different element types in the model

VORB a four character word to determine the type of analysis (e.g., STAT,VIBR,BUCK)

The second type has the format (6(1X,A3,1X,I3,1X,I3)), consists of one or more cards, and contains the following variables (one set for each element):

EL(I) element type (e.g., E21,E43)

NSECT(I) largest section property number used for that element type

NODVPE(I) number of design variables per element type

5. CNT is a file that initializes parameters for terminating the optimization process. CNT is in NAMELIST form with the NAMELIST name CNT.

6. ENDN is a file that defines limits imposed on the design variables in the end processor. ENDN is in NAMELIST format with the NAMELIST name EPIN.

7. CONS is a file supplied by the user to define certain constants such as the cross-sectional dimensions of a beam used in analytical gradient calculation. Since the file is only read by user-supplied subroutines, the format is also defined by the user. See section 4.3.6 for more information.

4.4.2 Model Data Files

Model data files contain finite-element-model input data for the analysis program SPAR. Two model data files, called SPAR runstreams, are used in PROSSS. The first, NRRS, defines the model in terms of nodes and elements connecting those nodes. This file is only used in the nonrepeatable part of PROSSS. The second model file, NGRS (no gradients) or RGS (gradients), updates the data for elements used as design variables and contains data for performing the analysis in the repeatable part of PROSSS. Sample listings of NRRS, NGRS, and RGS are contained in appendix E.

4.4.3 Transfer Data Files

Transfer data files pass data among the various programs in PROSSS. These are local files created within the system, and they are returned when the analysis-optimization process is completed. The following transfer files exist in PROSSS:

1. PCONRST is a file in NAMELIST format with the NAMELIST name SAVE that takes initial data from PCNPR and updates these data in each new pass through the optimization program. In two options from table I (2.2 and 2.3), the BLOCK file (described next) is used in place of PCONRST.

2. BLOCK or BLK is a file in NAMELIST format with the NAMELIST name BLK and contains objective function and constraint data.

3. PCNMNIO or TCNMNIO, CNMNIO (see note on PCNPR in section 4.4.1) is a file in NAMELIST format with two NAMELIST names. Name LINKE contains the number of design variables and their values. It is written by the end processor and read by the optimization program. Name LINKF contains objective function and gradient information. It is written by the optimization program and read by the front and end processors. Two options (2.2 and 2.3) replace PCNMNIO with CNT file discussed in section 4.4.1.

4. CHECK is created in options 1.2 and 2.2 when analysis has been performed for each design variable and their combination, if this file exists (i.e., local), then the optimization program is executed.

5. GONOGO is created when the objective function has not changed more than a specified tolerance (see CNT file in section 4.4.1) in three passes. If this file exists (i.e., local), the process is terminated.

6. SPFPOUT is created by the front processor and contains the updated design variables. This file is merged into the NGRS or RGS file using the MERGFP files described in section 4.4.4. The NGRS and RGS files are described in section 4.4.2.

7. RSOUT is an input file (runstream) for the analysis program similar to those found in section 4.4.2. RSOUT is created twice in each of two options (2.2 and 2.3), once in the nonrepeatable part and once in the repeatable part, and is used in calculating the analytical gradients.

8. SO is identical to the PSTRT file discussed in section 4.4.1.

9. SPARLA, SPARLB, SPARLC, SPARLD are data libraries created by the analysis program.

Sample listings of the transfer files are presented in appendix E. Files CHECK, GONOGO, SO, SPARLA, SPARLB, SPARLC, and SPARLD are not listed, either because of their length or because of their binary format.

4.4.4 Edit Data Files

Edit data files contain Text Editor (ref. 10) commands primarily for editing files input to the analysis program. None are created by the user. The following edit data files exist in PROSSS:

1. EDPASS1 creates the PASS file described in section 4.4.1 and initializes the variable to 1.

2. EDPASS2 reinitializes the number in the PASS file to 1.

3. EDIT1 edits the NRRS file described in section 4.4.2 to remove all but the element connection data.

4. EDIT2 edits the RSOUT file in the nonrepeatable part described in section 4.4.3 to prepare it for input to the analysis program.

5. MERGFP merges the SPFPOUT file described in section 4.4.3 into the NGRS or RGS files described in section 4.4.2.

6. EDGRDS edits the RSOUT file in the repeatable part described in section 4.4.3 to prepare it for input to the analysis program.

Sample listings of edit data files are contained in appendix E.

4.4.5 Saved Data Files

Three files are automatically saved for listing (SAVCOUT) and postprocessing or restart purposes (NRLA, SAVSPLD). File SAVCOUT contains a cumulative list of all the information output from the optimization program. File NRLA is a library of data output by the analysis program in the nonrepeatable part of the process. File SAVSPLD is a library of objective function, stress, and/or stress derivative information output by the analysis program in the nonrepeatable part of the process. File SAVSPLD is a library of objective function, stress, and/or stress derivative information output by the analysis program in the repeatable part of the process. These files are either too long or in a binary format and, therefore, are not listed in appendix E.

5 SAMPLE EXECUTIONS OF PROSSS

Each option was executed, using the input files shown in appendix A, to determine the final objective function (min. mass, kg) of the finite element model of the fuselage shown in figure 4. The model is composed of 80 joints, 58 rods,

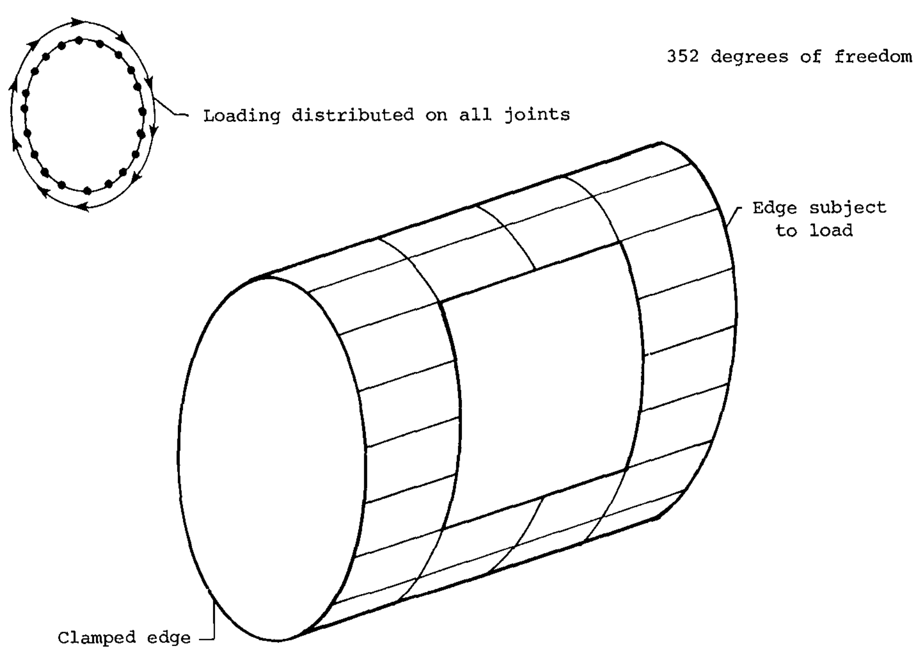


Figure 4.- Fuselage model used for testing.

76 beams, and 56 membranes. There are 352 degrees of freedom. The three design variables are the cross-sectional area of the transverse stringers (beams), the

cross-sectional area of the longitudinal stringers (rods), and the thickness of the panels (membranes). The design variables are handled by the optimizer in reciprocal form to improve the convergence. Their initial recriprocal starting values are 0.05 cm^2, 1.0 cm^2, and 4.0 cm, respectively. The initial objective function is 5460.12 kg. Shown in table VII is a comparison of actual results (not reciprocals)

TABLE VII.- COMPARISON OF RESULTS FROM DIFFERENT PROSSS OPTIONS

Option (table I)	1.1	1.2	1.3	2.2	2.3	FSD 1.1
Objective function, minimized mass, kg	6501.75	6398.91	6574.37	6350.10	6338.20	6362.39
Design variables:						
Cross-sectional area, cm^2	4.7364 1.6333	8.4246 1.5713	12.1567 1.5563	1.2088 1.5706	1.6722 1.6215	1. 1.6284
Thickness, cm	.1	.1	.1218	.1738	.1	.1
Run cost, $	121	127	98	68	31	20

for each option. All final objective functions and design variables are within reasonable limits. The piecewise linear approach is 50 to 75 percent less expensive to execute than the nonlinear approach.

Langley Research Center
National Aeronautics and Space Administration
Hampton, VA 23665
September 11, 1981

APPENDIX A

SAMPLE SETUP OF PROSSS FOR A SPECIFIC OPTION

This section contains listings of the source file PROSCRS, sample input files for each option, and the control card and edit files created by the binary file PROSCRS.

PROSCRS

```
      PROGRAM PROSCRS(TAPE8,TAPE9,TAPE10,TAPE11)
C
C  THIS PROGRAM CREATES A SPECIFIC OPTION PROCEDURE
C  FILE FROM A GENERAL ONE.
C
C
C  THE SPECIFIC NAMES ARE READ IN FROM UNIT 8.
C
      READ(8,10)A1,N1
   10 FORMAT(A7,3X,I2)
C
C  WRITE CONTROL CARDS ON UNIT 9
C
      WRITE(9,20)
   20 FORMAT(*.PROC,PROSSS.*/*GET,PROSOPT/UN=753437N.*)
      IF(N1.EQ.11) GO TO 40
      IF(N1.EQ.12) N=1
      IF(N1.EQ.13) N=2
      IF(N1.EQ.22) N=3
      IF(N1.EQ.23) N=4
      WRITE(9,30) N
   30 FORMAT(*COPYBR,PROSOPT,DUMMY,*,I1,*.*)
   40 WRITE(9,50)
   50 FORMAT(*COPYBR,PROSOPT,OPTION.*/*REWIND,OPTION,EDOPTC,EDOPT.*)
      WRITE(9,55)
   55 FORMAT(*EDIT,EDOPT,,EDOPTC,EDOUT.*/*REWIND,EDOPT.*)
      WRITE(9,60)
   60 FORMAT(*EDIT,OPTION,,EDOPT,EDOUT.*)
      WRITE(9,70)
   70 FORMAT(*RETURN,EDOPTC,PROSOPT,DUMMY,EDOPT,PROSCRB,PROSSIN.*)
      WRITE(9,80) N1
   80 FORMAT(*REWIND,OPTION.*/*BEGIN,OPT*,I2,*,OPTION.*)
C
C  WRITE EDIT COMMANDS ON UNIT 10.
C
      WRITE(10,100) A1,N1
  100 FORMAT(*RS:/*,A7,*/,/*,I2,*/;100*)
  110 READ(8,120) A1,A2
  120 FORMAT(A7,3X,A7)
      IF(EOF(8)) 150,130
  130 WRITE(10,140) A1,A2
  140 FORMAT(*RS:/*,A7,*/,/*,A7,*/;100*)
      GO TO 110
```

```
C
C  REMOVE BLANKS IN EDIT COMMANDS
C
 150   WRITE(11,160)
 160   FORMAT(*RS:/ /;100*/*END*)
       WRITE(10,165)
 165   FORMAT(*END*)
       STOP
       END
```

APPENDIX A

SAMPLE INPUT FILES FOR OPTIMIZATION OPTIONS (TABLE I)

Option 1.1 with nonrepeatable part

General	Specific	Note
POPT	11	
NONREPT	1	Execute nonrepeatable part
NRRS	NRSPARS	Nonrepeatable runstream
FUSD	0	
CONMIN	CONMB1	
ENDP1	EPFUB1	
FRNT	FPFGB1	
ENDN	EPFUIN	
PCNPR	CONP1	
PSTRT	STRP3	
CONS	CONS	
NGRS	RRSNG	
PCONRST	CONREST	
PCNMNIO	CNMNIO	
SAVCOUT	REPCOUT	
NSPARLA	NRLANG	
FLENGTH	100000	
BLK	BLOCK	
SAVSPLD	REPSPLD	

Option 1.1 with fully stressed design

General	Specific	Note
POPT	11	
NONREPT	0	
FUSD	1	Use fully stressed design
FSDSUB	FSDSUBB	
CONMIN	FSDB	
ENDP1	EPFUB1	
FRNT	FPFGB1	
ENDN	EPFUIN	
PCNPR	CONP1	
PSTRT	STRP3	
CONS	CONS	
NGRS	RRSNG	
PCONRST	CONREST	
PCNMNIO	CNMNIO	
SAVCOUT	REPCOUT	
NSPARLA	NRLANG	
FLENGTH	100000	
BLK	BLOCK	
SAVSPLD	REPSPLD	

Option 1.2

General	Specific
POPT	12
NONREPT	0
FUSD	0
CONMIN	CONMB1
ENDP1	EPFUB1
FRNT	FPFGB1
ENDN	EPFUIN
PCNPR	CONP13
PSTRT	STRP3
CONS	CONS
NGRS	RRSNG
PCONRST	CONREST
PCNMNIO	CNMNIO
SAVCOUT	REPCOUT
NSPARLA	NRLANG
FLENGTH	100000
BLK	BLOCK
SAVSPLD	REPSPLD

Option 1.3

General	Specific	Note
POPT	13	
NONREPT	0	
FUSD	0	
CONMIN	CONMB1	
ENDP1	EPFUB1	
ENDP2	EPFGB1	
SUBS	1	For gradients
BINDEPB	BINDEPB	
FRNT	FPFGB1	
ENDN	EPFUIN	
PCNPR	CONP13	
PSTRT	STRP3	
INPT	INPTT	
CONS	CONS	
NGRS	RRSNG	
RGS	RRSG	
PCONRST	CONREST	
PCNMNIO	CNMNIO	
SAVCOUT	REPCOUT	
NSPARLA	NRLAG	
FLENGTH	100000	
BLK	BLOCK	
SAVSPLD	REPSPLD	
BLK	BLOCK	

Option 2.2

General	Specific
POPT	22
NONREPT	1
NRRS	NRSPARS
FUSD	0
CONMIN	CONMB2
ENDP1	EPFUB1
FRNT	FPFGB1
ENDN	EPFUIN
PCNPR	CONP2
PSTRT	STRP4
CONS	CONS
CNT	CNT
NGRS	RRSNG
PCONRST	BLOCK
PCNMNIO	CNT
SAVCOUT	REPCOUT
NSPARLA	NRLANG
FLENGTH	100000
BLK	BLOCK
SAVSPLD	REPSPLD

Option 2.3

General	Specific
POPT	23
NONREPT	0
FUSD	0
CONMIN	CONMB2
ENDP1	EPFGB1
SUBS	1
BINDEPB	BINDEPB
FRNT	FPFGB1
ENDN	EPFUIN
PCNPR	CONP2
PSTRT	STRP4
INPT	INPTT
CONS	CONS
CNT	CNT
NGRS	RRSG
PCONRST	BLOCK
PCNMNIO	CNT
SAVCOUT	REPCOUT
NSPARLA	NRLAG
FLENGTH	100000
SAVSPLD	REPSPLD
BLK	BLOCK

APPENDIX A

CONTROL CARD FILE CREATED BY PROSCRB

```
.PROC,PROSSS.
GET,PROSOPT/UN=753437N.
COPYBR,PROSOPT,OPTION.
REWIND,OPTION,EDOPTC,EDOPT.
EDIT,EDOPT,,EDOPTC,EDOUT.
REWIND,EDOPT.
EDIT,OPTION,,EDOPT,EDOUT.
RETURN,EDOPTC,PROSOPT,DUMMY,EDOPT,PROSCRB,PROSSIN.
REWIND,OPTION.
BEGIN,OPT11,OPTION.
```

EDIT FILE CREATED BY PROSCRB

(After extraneous blanks removed)

```
RS:/POPT    /,/11/;100
RS:/NONREPT/,/1       /;100
RS:/NRRS    /,/NRSPARS/;100
RS:/FUSD    /,/0      /;100
RS:/CONMIN /,/CONMB1 /;100
RS:/ENDP1   /,/EPFUB1 /;100
RS:/FRNT    /,/EPFGB1 /;100
RS:/ENDN    /,/EPFUIN /;100
RS:/PCNPR   /,/CONP1  /;100
RS:/PSTRT   /,/STRP3  /;100
RS:/CONS    /,/CONS   /;100
RS:/NGRS    /,/RRSNG  /;100
RS:/PCONRST/,/CONREST/;100
RS:/PCNMNIO/,/CNMNIO /;100
RS:/SAVCOUT/,/REPCOUT/;100
RS:/NSPARLA/,/NRLANG /;100
RS:/FLENGTH/,/100000 /;100
RS:/BLK     /,/BLOCK  /;100
RS:/SAVSPLD/,/REPSPLD/;100
END
```

APPENDIX B

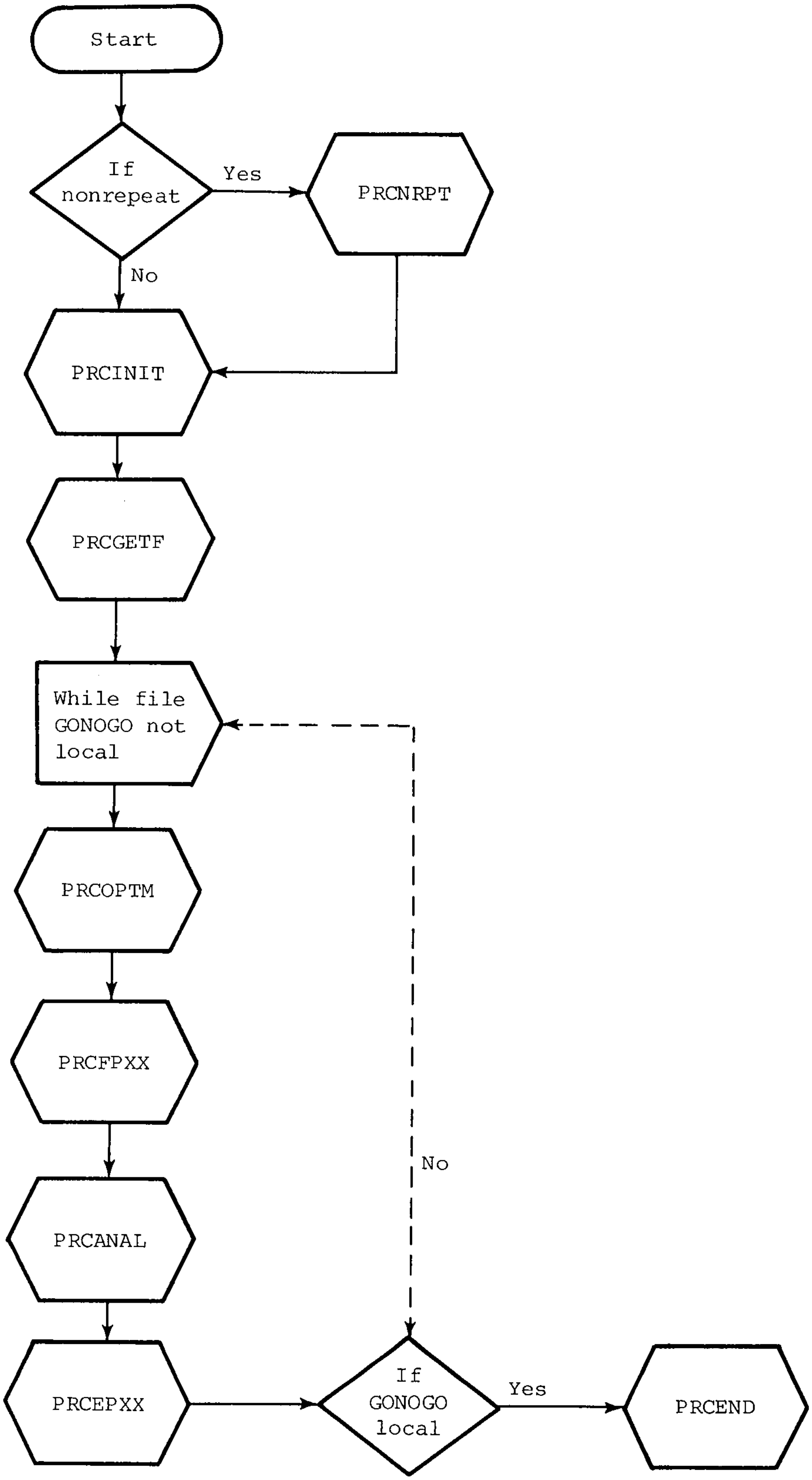
OPTION FILES

The following are listings and flowcharts for each of the optimization options given in table I:

Option 1.1

```
.PROC,OPT11.
IFE,(NONREPT.EQ.1),NONREPEAT.
  BEGIN,PRCNRPT,PROSPRC,POPT,NRRS,FLENGTH,INPT,NSPARLA.
ENDIF,NONREPEAT.
BEGIN,PRCINIT,PROSPRC,POPT,CONMIN,ENDP1,ENDP2,SUBS,BINDEPB,FUSD,FSDSUB.
BEGIN,PRCGETF,PROSPRC,POPT,FRNT,ENDN,PCNPR,PSTRT,INPT,CONS,CNT,NGRS,RGS.
WHILE,(.NOT.(FILE(GONOGO,LO))),OPTION11.
  BEGIN,PRCOPTM,PROSPRC,PCONRST,PCNMNIO,SAVCOUT.
  BEGIN,PRCFPXX,PROSPRC.
  BEGIN,PRCANAL,PROSPRC,NSPARLA,FLENGTH,SAVSPLD.
  BEGIN,PRCEPXX,PROSPRC,BLK.
ENDW,OPTION11.
BEGIN,PRCEND,PROSPRC.
```

APPENDIX B


Start
If nonrepeat
Yes
PRCNRPT
No
PRCINIT
PRCGETF
While file GONOGO not local
PRCOPTM
PRCFPXX
PRCANAL
PRCEPXX
No
If GONOGO local
Yes
PRCEND

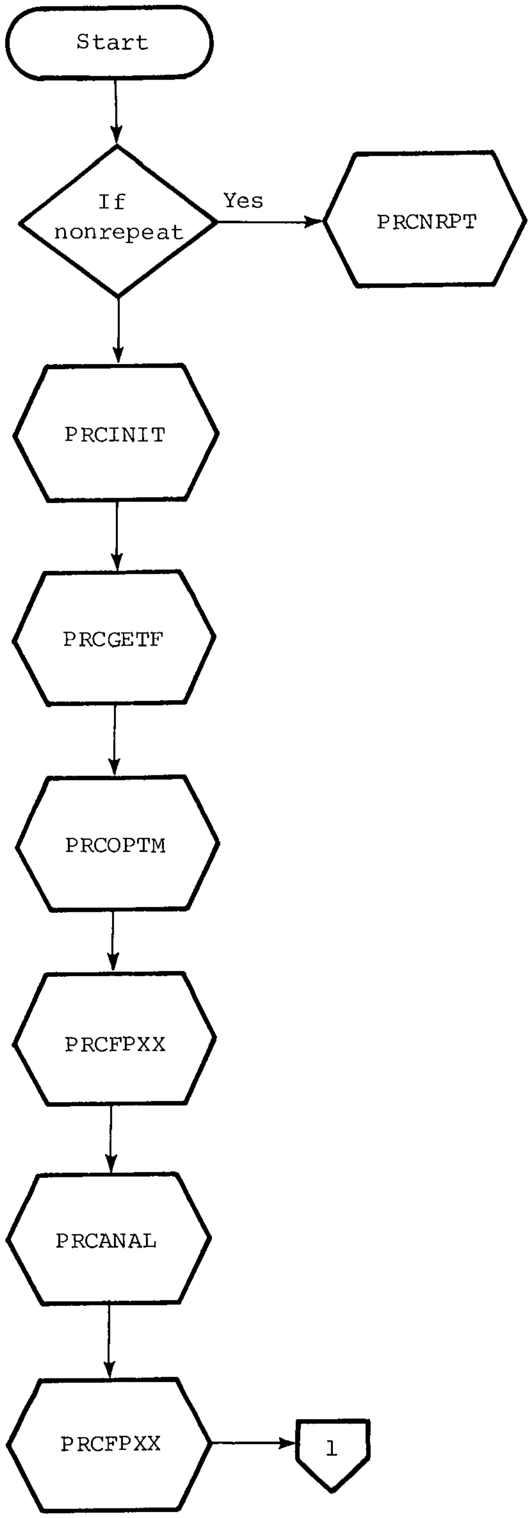
APPENDIX B

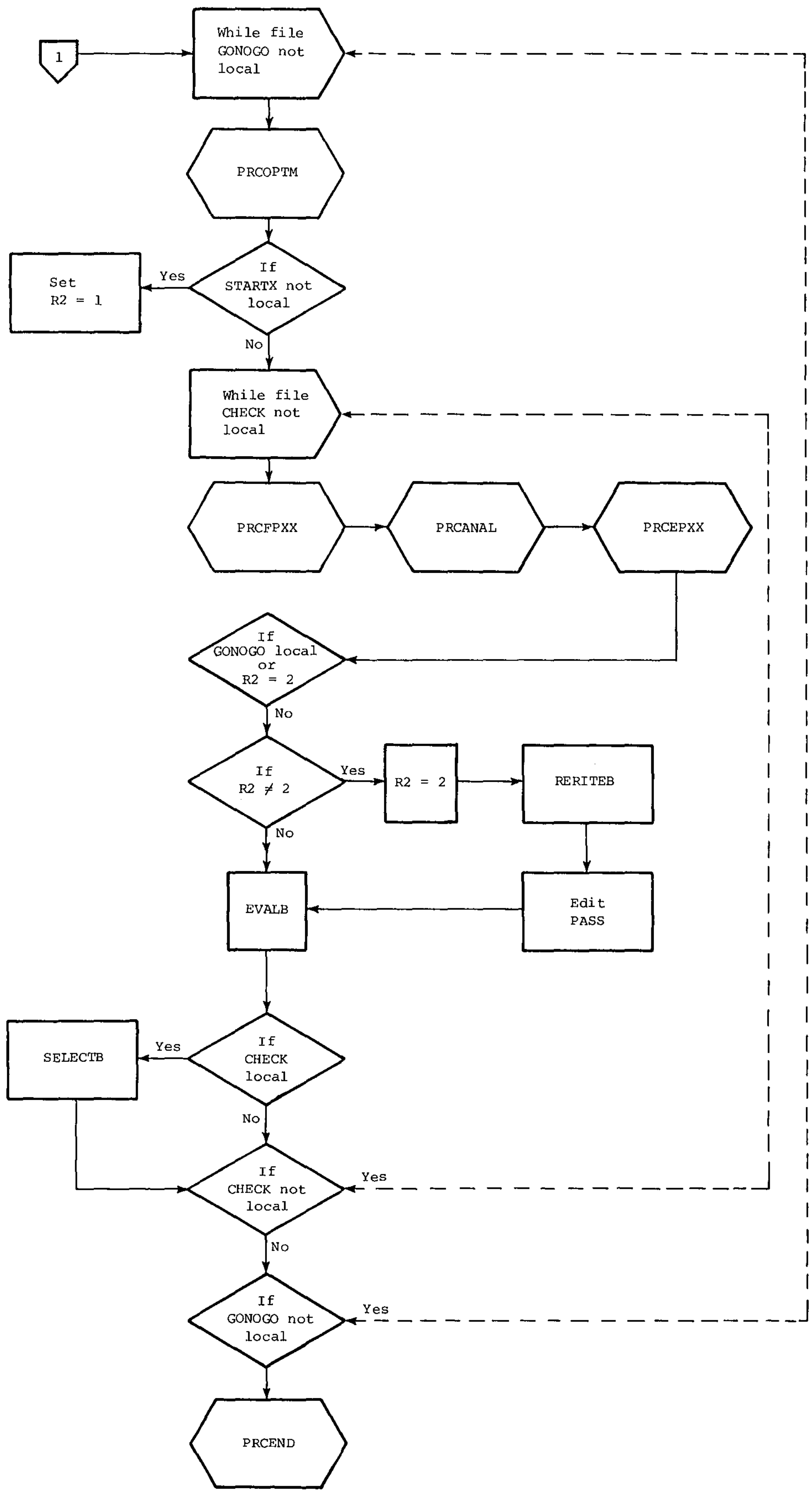
Option 1.2

```
.PROC,OPT12.
IFE,(NONREPT.EQ.1),NONREPEAT.
  BEGIN,PRCNRPT,PROSPRC,POPT,NRRS,FLENGTH,INPT,NSPARLA.
ENDIF,NONREPEAT.
SET(R2=1)
BEGIN,PRCINIT,PROSPRC,POPT,CONMIN,ENDP1,ENDP2,SUBS,BINDEPB,FUSD,FSDSUB.
BEGIN,PRCGETF,PROSPRC,POPT,FRNT,ENDN,PCNPR,PSTRT,INPT,CONS,CNT,NGRS,RGS.
BEGIN,PRCOPTM,PROSPRC,PCONRST,PCNMNIO,SAVCOUT.
BEGIN,PRCFPXX,PROSPRC.
BEGIN,PRCANAL,PROSPRC,NSPARLA,FLENGTH,SAVSPLD.
BEGIN,PRCEPXX,PROSPRC,BLK.
WHILE,(.NOT.(FILE(GONOGO,LO))),OUTERLOOP.
IFE,(.NOT.(FILE(GONOGO,LO))),ENDOUTER.
  RETURN,STARTX.
  BEGIN,PRCOPTM,PROSPRC,PCONRST,PCNMNIO,SAVCOUT.
  SET(R3=1)
  IFE,(.NOT.(FILE(STARTX,LO))),SETR2.
    SET(R2=1)
  ENDIF,SETR2.
  WHILE,(.NOT.(FILE(CHECK,LO))),INNERLOOP.
    IFE,((.NOT.(FILE(STARTX,LO))).OR.(R3.EQ.2)),DONOTSKIP.
      BEGIN,PRCFPXX,PROSPRC.
      BEGIN,PRCANAL,PROSPRC,NSPARLA,FLENGTH,SAVSPLD.
      BEGIN,PRCEPXX,PROSPRC,BLK.
      REWIND,TCNMNIO,CNMNIO.
      COPYBF,CNMNIO,TCNMNIO.
      IFE,((FILE(GONOGO,LO)).OR.(R2.NE.2)),SKIPTOOUT.
        SKIP,ENDOUTER.
      ENDIF,SKIPTOOUT.
    ENDIF,DONOTSKIP.
    IFE,(R2.NE.2),RERITE.
      REWIND,CNMNIO,RERITEB,SO,PASS,EDPASS2.
      SET(R2=2)
LDSET(PRESET=ZERO)
RERITEB,CNMNIO,,SO.
EDIT,PASS,,EDPASS2,EDOUT.
      RETURN,EDOUT.
    ENDIF,RERITE.
    SET(R3=2)
    REWIND,PASS,CONPAR,SO,CNMNIO,EVALB,BLOCK.
LDSET(PRESET=ZERO)
EVALB,CONPAR,,,PASS,SO,CNMNIO,BLOCK,CHECK.
    IFE,(FILE(CHECK,LO)),SELECT.
      RETURN,CHECK.
      RFL,250000.
      REDUCE,-.
      REWIND,SELECTB,BLOCK,CONREST.
SELECTB,BLOCK,,,CONREST.
      RETURN,BLOCK.
```

```
      REWIND,CNMNIO,TCNMNIO.
      COPYBF,TCNMNIO,CNMNIO.
      SKIP,ENDOUTER.
    ENDIF,SELECT.
  ENDW,INNERLOOP.
ENDIF,ENDOUTER.
ENDW,OUTERLOOP.
BEGIN,PRCEND,PROSPRC.
```

APPENDIX B


Start
If nonrepeat
Yes
PRCNRPT
PRCINIT
PRCGETF
PRCOPTM
PRCFPXX
PRCANAL
PRCFPXX
1

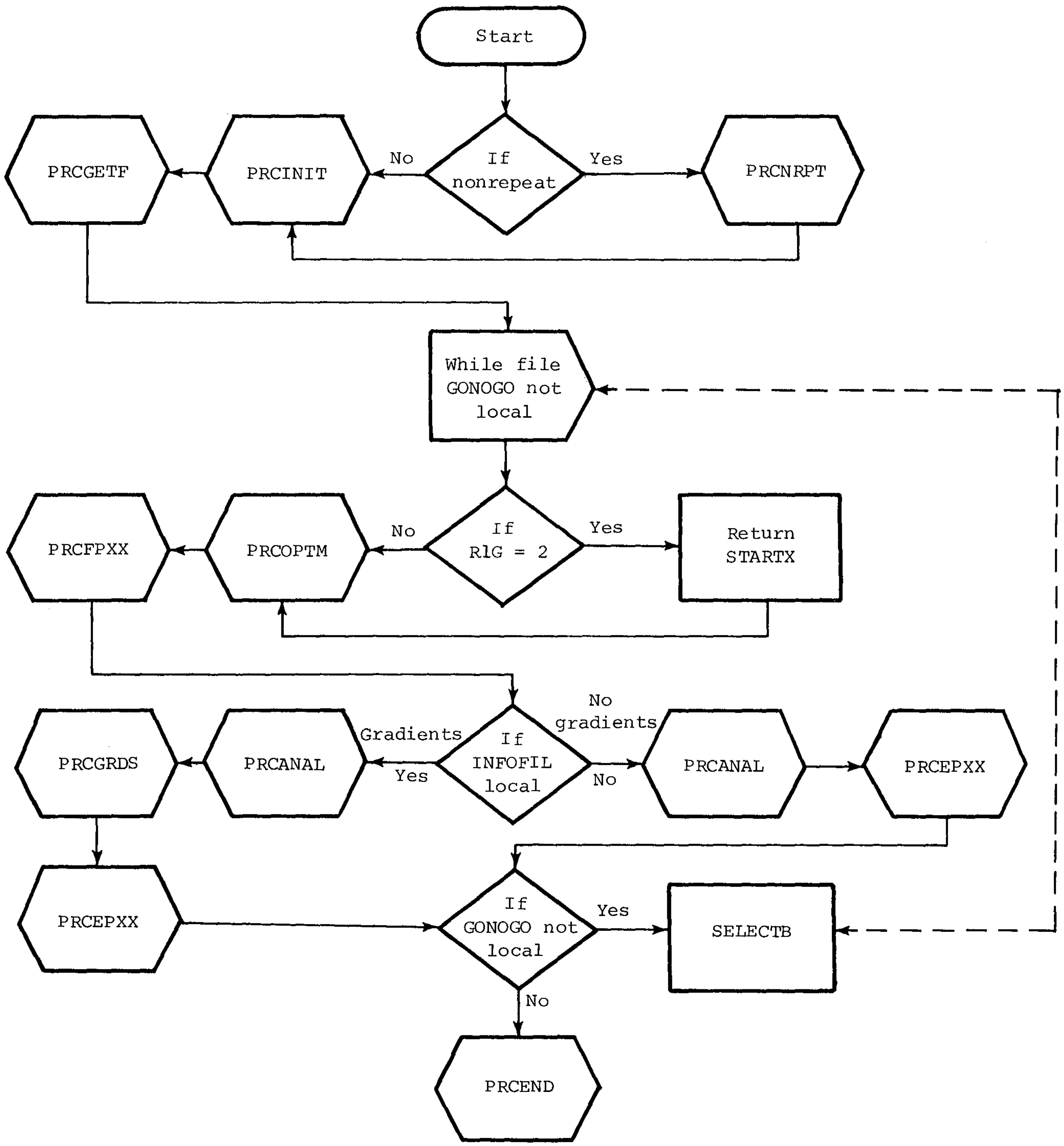

1
While file GONOGO not local
PRCOPTM
If STARTX not local
Yes
Set R2 = 1
No
While file CHECK not local
PRCFPXX
PRCANAL
PRCEPXX
If GONOGO local or R2 = 2
No
If R2 ≠ 2
Yes
R2 = 2
RERITEB
No
EVALB
Edit PASS
If CHECK local
Yes
SELECTB
No
If CHECK not local
Yes
No
If GONOGO not local
Yes
PRCEND

APPENDIX B

Option 1.3

```
.PROC,OPT13.
IFE,(NONREPT.EQ.1),NONREPEAT.
  BEGIN,PRCNRPT,PROSPRC,POPT,NRRS,FLENGTH,INPT,NSPARLA.
ENDIF,NONREPEAT.
BEGIN,PRCINIT,PROSPRC,POPT,CONMIN,ENDP1,ENDP2,SUBS,BINDEPB,FUSD,FSDSUB.
BEGIN,PRCGETF,PROSPRC,POPT,FRNT,ENDN,PCNPR,PSTRT,INPT,CONS,CNT,NGRS,RGS.
WHILE,(.NOT.(FILE(GONOGO,LO))),OPTION13.
  IFE,(R1G.EQ.2),NORETURN.
    RETURN,STARTX.
  ENDIF,NORETURN.
  BEGIN,PRCOPTM,PROSPRC,PCONRST,PCNMNIO,SAVCOUT.
  BEGIN,PRCFPXX,PROSPRC.
  IFE,(FILE(INFOFIL,LO)),GRADS.
    REWIND,TEMPRS,RGRS,ENDG,ENDPROC.
    COPYCF,RGRS,TEMPRS.
    COPYBF,ENDG,ENDPROC.
    BEGIN,PRCANAL,PROSPRC,NSPARLA,FLENGTH,SAVSPLD.
    BEGIN,PRCGRDS,PROSPRC,SUBS,SAVSPLD.
    BEGIN,PRCEPXX,PROSPRC,BLK.
  ENDIF,GRADS.
  IFE,(.NOT.(FILE(INFOFIL,LO))),NOGRADS.
    REWIND,TEMPRS,NGRRS,NGEND,ENDPROC.
    COPYCF,NGRRS,TEMPRS.
    COPYBF,NGEND,ENDPROC.
    BEGIN,PRCANAL,PROSPRC,NSPARLA,FLENGTH,SAVSPLD.
    BEGIN,PRCEPXX,PROSPRC,BLK.
  ENDIF,NOGRADS.
  RETURN,INFOFIL.
  IFE,(.NOT.(FILE(GONOGO,LO))),END13.
    REWIND,BLOCK,CONREST,SELECTB.
    RFL,250000.
    REDUCE,-.
LDSET(PRESET=ZERO)
SELECTB,BLOCK,,,CONREST.
  ENDIF,END13.
ENDW,OPTION13.
BEGIN,PRCEND,PROSPRC.
```

APPENDIX B

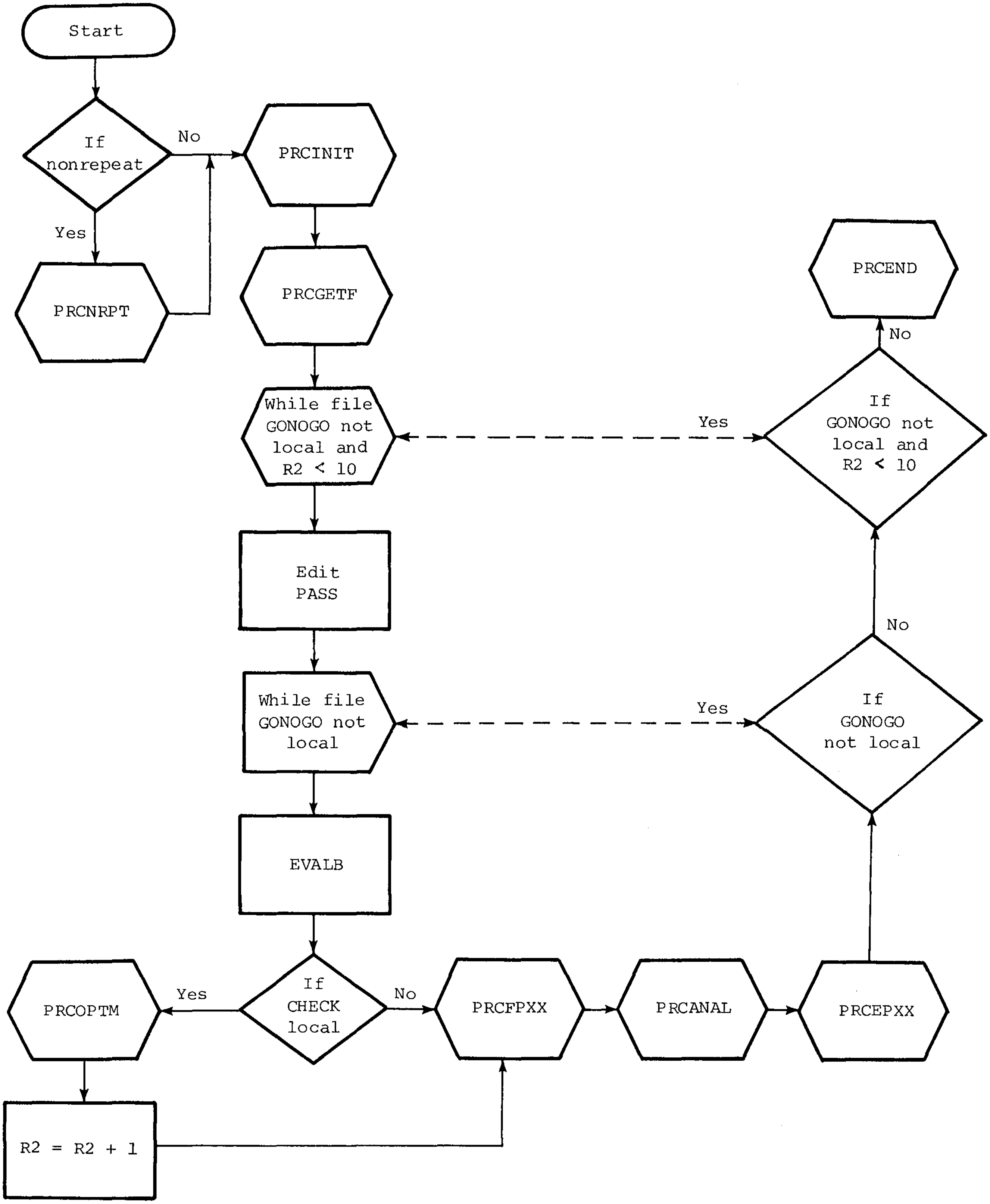

Start
If nonrepeat
No
Yes
PRCINIT
PRCGETF
PRCNRPT
While file GONOGO not local
If R1G = 2
No
Yes
PRCOPTM
PRCFPXX
Return STARTX
If INFOFIL local
Gradients
Yes
No gradients
No
PRCANAL
PRCGRDS
PRCANAL
PRCEPXX
PRCEPXX
If GONOGO not local
Yes
No
SELECTB
PRCEND

APPENDIX B

Option 2.2

```
.PROC,OPT22.
IFE,(NONREPT.EQ.1),NONREPEAT.
  BEGIN,PRCNRPT,PROSPRC,POPT,NRRS,FLENGTH,INPT,NSPARLA.
ENDIF,NONREPEAT.
SET(R2=1)
BEGIN,PRCINIT,PROSPRC,POPT,CONMIN,ENDP1,ENDP2,SUBS,BINDEPB,FUSD,FSDSUB.
BEGIN,PRCGETF,PROSPRC,POPT,FRNT,ENDN,PCNPR,PSTRT,INPT,CONS,CNT,NGRS,RGS.
WHILE,((R2.LT.10).AND.(.NOT.(FILE(GONOGO,LO)))),OPTION22.
  IFE,((R2.LT.10).AND.(.NOT.(FILE(GONOGO,LO)))),END22.
    REWIND,PASS,EDPASS2.
EDIT,PASS,,EDPASS2,EDOUT.
    RETURN,CHECK,EDOUT.
    WHILE,(.NOT.(FILE(GONOGO,LO))),KEEPON.
      REWIND,CONPAR,STARTX,PASS,CNMNIO,BLOCK,EVALB.
LDSET(PRESET=ZERO)
EVALB,CONPAR,,,PASS,STARTX,CNMNIO,BLOCK,CHECK.
      IFE,(FILE(CHECK,LO)),OPTIMIZE.
        BEGIN,PRCOPTM,PROSPRC,PCONRST,PCNMNIO,SAVCOUT.
        SET(R2=R2+1)
        SKIP,END22.
      ENDIF,OPTIMIZE.
      BEGIN,PRCFPXX,PROSPRC.
      BEGIN,PRCANAL,PROSPRC,NSPARLA,FLENGTH,SAVSPLD.
      BEGIN,PRCEPXX,PROSPRC,BLK.
    ENDW,KEEPON.
  ENDIF,END22.
ENDW,OPTION22.
BEGIN,PRCEND,PROSPRC.
```

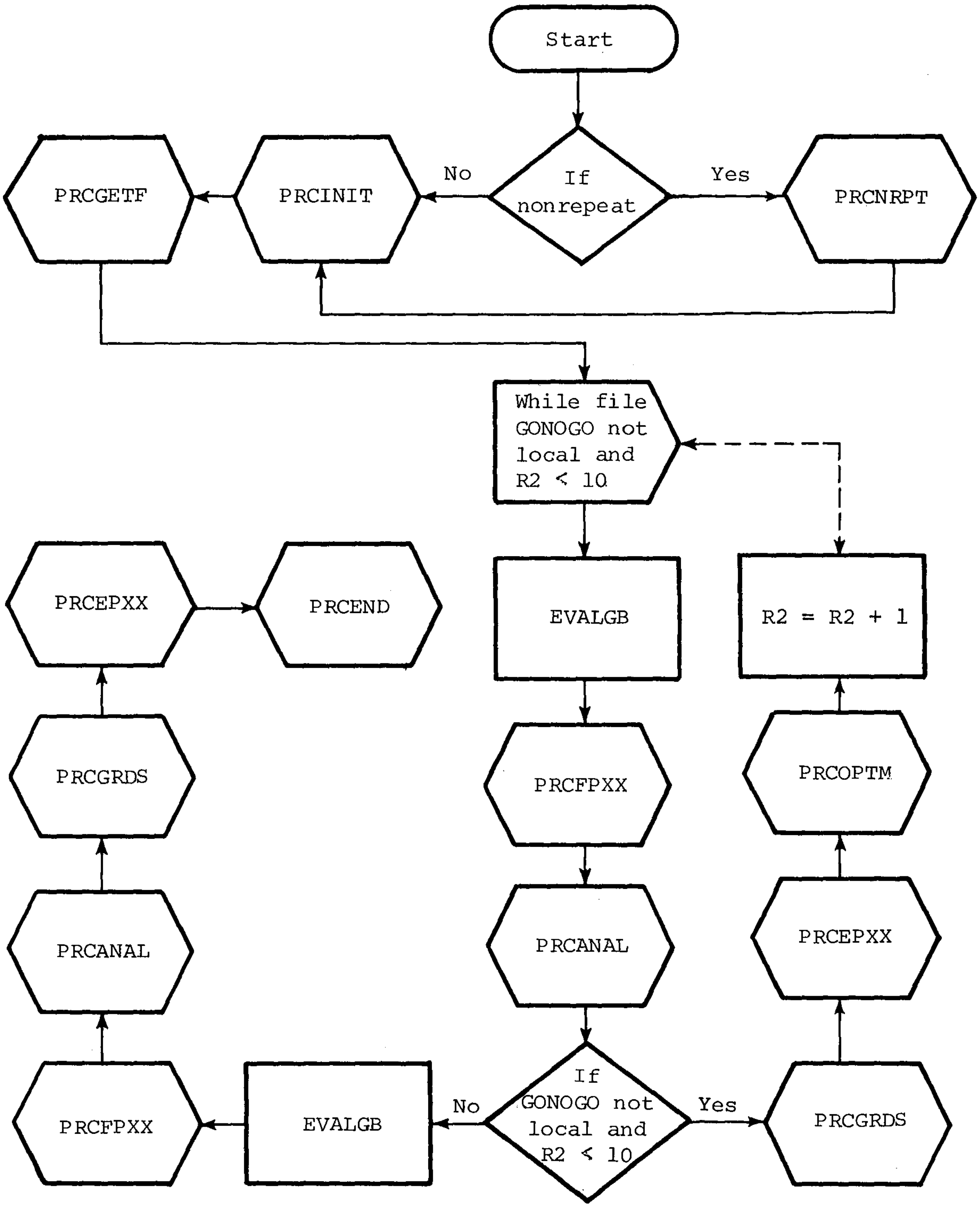
APPENDIX B


Start
If nonrepeat
No
Yes
PRCNRPT
PRCINIT
PRCGETF
PRCEND
No
While file GONOGO not local and R2 < 10
Yes
If GONOGO not local and R2 < 10
Edit PASS
No
While file GONOGO not local
Yes
If GONOGO not local
EVALB
PRCOPTM
Yes
If CHECK local
No
PRCFPXX
PRCANAL
PRCEPXX
R2 = R2 + 1

APPENDIX B

Option 2.3

```
.PROC,OPT23.
IFE,(NONREPT.EQ.1),NONREPEAT.
  BEGIN,PRCNRPT,PROSPRC,POPT,NRRS,FLENGTH,INPT,NSPARLA.
ENDIF,NONREPEAT.
SET(R2=1)
BEGIN,PRCINIT,PROSPRC,POPT,CONMIN,ENDP1,ENDP2,SUBS,BINDEPB,FUSD,FSDSUB.
BEGIN,PRCGETF,PROSPRC,POPT,FRNT,ENDN,PCNPR,PSTRT,INPT,CONS,CNT,NGRS,RGS.
WHILE,((.NOT.(FILE(GONOGO,LO))).AND.(R2.LT.10)),OPTION23.
  REWIND,STARTX,CNMNIO,CONPAR,EVALGB.
LDSET(PRESET=ZERO)
EVALGB,CONPAR,,,STARTX,CNMNIO.
  BEGIN,PRCFPXX,PROSPRC.
  BEGIN,PRCANAL,PROSPRC,NSPARLA,FLENGTH,SAVSPLD.
  IFE,((.NOT.(FILE(GONOGO,LO))).AND.(R2.LT.10)),END23.
    BEGIN,PRCGRDS,PROSPRC,SUBS,SAVSPLD.
    BEGIN,PRCEPXX,PROSPRC,BLK.
    BEGIN,PRCOPTM,PROSPRC,PCONRST,PCNMNIO,SAVCOUT.
    SET(R2=R2+1)
  ENDIF,END23.
ENDW,OPTION23.
  REWIND,STARTX,CNMNIO,CONPAR,EVALGB.
LDSET(PRESET=ZERO)
EVALGB,CONPAR,,,STARTX,CNMNIO.
  BEGIN,PRCFPXX,PROSPRC.
  BEGIN,PRCANAL,PROSPRC,NSPARLA,FLENGTH,SAVSPLD.
  BEGIN,PRCGRDS,PROSPRC,SUBS,SAVSPLD.
  BEGIN,PRCEPXX,PROSPRC,BLK.
BEGIN,PRCEND,PROSPRC.
```


Start
If nonrepeat
No
Yes
PRCINIT
PRCGETF
PRCNRPT
While file GONOGO not local and R2 < 10
EVALGB
R2 = R2 + 1
PRCEPXX
PRCEND
PRCGRDS
PRCFPXX
PRCOPTM
PRCANAL
PRCANAL
PRCEPXX
PRCFPXX
EVALGB
If GONOGO not local and R2 < 10
No
Yes
PRCGRDS

APPENDIX C

PROCEDURE FILES

PRCNRPT

```
.PROC,PRCNRPT,NROPT,NRRS,FLX,I,NRLA.
.*
.*  THE PROCEDURE CREATES A SPAR LIBRARY
.*     FROM THE NON-REPEATABLE PART
.*
GET,SPAR=SPAR14I,DCU=DCU14I/UN=750756N.
GET,NRRS.
RFL,FLX.
REDUCE,-.
SPAR,NRRS,NSPROUT.
REPLACE,SPARLA=NRLA.
.*
.*  TEST TO SEE IF GRADIENTS ARE REQUIRED
.*

IFE,(NROPT.EQ.13.OR.NROPT.EQ.23),GRADIENTS.
GET,INPT=I.
GET,EDIT1,EDIT2,BLDELDB/UN=753437N.
REWIND,NRRS.
.*
.*  EDIT OUT ALL BUT ELD INPUT IN RUNSTREAM
.*
EDIT,NRRS,,EDIT1,EDOUT.
REWIND,NRRS.
.*
.*  CREATE SPAR RUNSTREAM TO FIND DERIVATIVES
.*
BLDELDB,INPT,NRRS,RSOUT.
REWIND,RSOUT.
EDIT,RSOUT,,EDIT2,EDOUT.
REWIND,RSOUT,SPARLA.
.*
.*  EXECUTE SPAR AGAIN TO FIND DERIVATIVES OF
.*     THE STIFFNESS MATRIX WITH RESPECT TO
.*     THE DESIGN VARIABLES
.*
SPAR,RSOUT,NSPROUT.
RETURN,INPT,EDIT1,EDIT2,BLDELDB,EDOUT.
RETURN,RSOUT,TAPE21,TAPE22,NSPROUT.
REPLACE,SPARLB=NRLA.
ENDIF,GRADIENTS.
RETURN,SPAR,DCU,SPARLA,NRRS.
```

APPENDIX C

PRCINIT

```
.PROC,PRCINIT,OP,A,B,BB,NSUB,C,FSD,FSUB.
.*
.*  THIS PROCEDURE FILE CREATES PROGRAMS USED BY THE
.*           DIFFERENT PROSSS OPTIONS.
.*
SET(R1G=1)
MAP,OFF.
.*
.*   CREATE CONMIN
.*
GET,A.
GET,CONMINB/UN=753437N.
.*
.*  TEST FOR FULLY STRESSED DESIGN REQUIREMENT
.*
IFE,(FSD.EQ.1),GETFSDSUB.
RETURN,CONMINB.
GET,CONMINB=FSUB.
ENDIF,GETFSDSUB.
COPYL,A,CONMINB,CONMIN,,RA.
.*
.*  CREATE END PROCESSOR
.*
GET,B.
GET,SCOMBLK,SPARLIB/UN=319925N.
COPYBR,B,ENDPROC.
COPYBR,SCOMBLK,ENDPROC.
COPYBF,B,ENDPROC.
IFE,(OP.EQ.13),GEND.
REWIND,SCOMBLK.
GET,BB.
COPYBR,BB,ENDG.
COPYBR,SCOMBLK,ENDG.
COPYBF,BB,ENDG.
RENAME,NGEND=ENDPROC.
ENDIF,GEND.
.*
.*  TEST TO SEE IF OPTION USING GRADIENTS WAS CHOSEN
.*
IFE,(OP.EQ.13.OR.OP.EQ.23),GRADIENTS.
.*
.*  CREATE PROGRAM FOR GENERATING REPEATABLE SPAR RUNSTREAMS
.*
GET,GNGRDRB/UN=753437N.
COPYBR,GNGRDRB,GREPEAT.
.*
.*  TEST FOR BEAM OR PLATE SUBROUTINES
.*
IFE,(NSUB.NE.0),NOSUBS.
```

```
GET,C.
COPYBR,C,GREPEAT,NSUB.
ENDIF,NOSUBS.
COPYBF,GNGRDRB,GREPEAT.
RETURN,GNGRDRB.
.*
.*  TEST FOR NEED TO CONVERT FORCES AND MOMENTS TO STRESSES
.*
IFE,(NSUB.NE.0),GRADIENTS.
.*
.*  CREATE PROGRAM TO CONVERT FORCES AND MOMENTS TO STRESSES
.*
GET,DRVSTRB/UN=753437N.
REWIND,SCOMBLK.
COPYBR,DRVSTRB,FAM2STR.
COPYBR,C,FAM2STR,NSUB.
COPYBR,SCOMBLK,FAM2STR.
COPYBF,DRVSTRB,FAM2STR.
RETURN,DRVSTRB.
ENDIF,GRADIENTS.
GET,SPAR=SPAR14I,DCU=DCU14I/UN=750756N.
RETURN,SCOMBLK,A,B,C,CONMINB.
```

APPENDIX C

PRCGETF

```
.PROC,PRCGETF,OP,F,E,CN,S,I,C,CT,RS,RGS.
.*
.*  THIS PROCEDURE FILE GETS ALL THE FILES REQUIRED FOR EXECUTING
.*                   A PARTICULAR OPTION
.*
.*
.*   GET FILES USED IN ALL OPTIONS
.*
GET,EDGRDS,MERGFP,EDPASS1/UN=753437N.
GET,FRTPROC=F,ENDIN=E,CONPAR=CN,STARTX=S.
GET,CONS=C,TEMPRS=RS.
.*
.*  CHANGE PASS TO 1 FOR FIRST PASS
.*
EDIT,PASS,,EDPASS1,EDOUT.
RETURN,EDPASS1,EDOUT.
.*
.*  GET ADDITIONAL FILES NEEDED FOR OPTION 1.2
.*
IFE,(OP.EQ.12),OPTION12.
GET,EVALB,RERITEB,SELECTB,EDPASS2/UN=753437N.
ENDIF,OPTION12.
.*
.*  GET ADDITIONAL FILES NEEDED FOR OPTION 1.3
.*
IFE,(OP.EQ.13),OPTION13.
GET,INPT=I,RGRS=RGS.
RENAME,NGRRS=TEMPRS.
GET,SELECTB/UN=753437N.
ENDIF,OPTION13.
.*
.*  GET ADDITIONAL FILES NEEDED FOR OPTION 2.2
.*
IFE,(OP.EQ.22),OPTION22.
GET,EVALB,EDPASS2/UN=753437N.
GET,CNT=CT.
ENDIF,OPTION22.
.*
.*  GET ADDITIONAL FILES NEEDED FOR OPTION 2.3
.*
IFE,(OP.EQ.23),OPTION23.
GET,CNT=CT,INPT=I.
GET,EVALGB/UN=753437N.
ENDIF,OPTION23.
```

APPENDIX C

PRCFPXX

```
.PROC,PRCFPXX.
.*
.*  THIS PROCEDURE FILE EXECUTES THE FRONT PROCESSOR
.*
REWIND,FRTPROC,CNMNIO,CONS,SPFPOUT.
LDSET(PRESET=ZERO)
FRTPROC,CNMNIO,SPFPOUT,CONS.
```

APPENDIX C

PRCOPTM

```
.PROC,PRCOPTM,C,D,F.
.*
.*  THIS PROCEDURE FILE EXECUTES CONMIN
.*
RFL,250000.
REDUCE,-.
REWIND,CONPAR,STARTX,C,D,PASS,GONOGO,CONMIN.
LDSET(PRESET=ZERO)
CONMIN,CONPAR,CONOUT,STARTX,C,D,GONOGO,PASS,INFOFIL.
PACK(CONOUT)
REPLACE,CONOUT=F.
SKIPEI(CONOUT)
```

APPENDIX C

PRCANAL

```
.PROC,PRCANAL,NRLA,FLX,SAVELD.
.*
.*  THIS PROCEDURE FILE EXECUTES SPAR FOR THE ANALYSIS
.*
GET,SPARLA=NRLA.
REWIND,SPARLA,RRS,MERGFP,SPFPOUT,SPAROUT,TEMPRS.
COPYCF,TEMPRS,RRS.
REWIND,RRS.
.*
.*  MERGE OUTPUT FROM THE FRONT PROCESSOR INTO THE
.*          SPAR RUNSTREAM
.*
EDIT,RRS,,MERGFP,EDOUT.
REWIND,RRS.
RFL,FLX.
REDUCE,-.
SPAR,RRS,SPAROUT.
REPLACE,SPARLD=SAVELD.
.*
.*  SAVE INITIAL SPAR INPUT AND OUTPUT FOR LATER LISTING
.*
IFE,(R1G.EQ.1),RENAMES.
RENAME,SPIN1=RRS,SPOUT1=SPAROUT.
SET(R1G=2)
ENDIF,RENAMES.
RETURN,EDOUT,SPFPOUT.
```

PRCGRDS

```
.PROC,PRCGRDS,NSUB,SAVELD.
.*
.*  THIS PROCEDURE FILE CALCULATES GRADIENTS
.*
REWIND,INPT,CONS,CNMNIO,GREPEAT,SPARLA,SPARLD.
.*
.*  CREATE SPAR RUNSTREAM
.*
GREPEAT,INPT,CONS,CNMNIO,RSOUT.
REWIND,RSOUT,EDGRDS.
EDIT,RSOUT,,EDGRDS,EDOUT.
REWIND,RSOUT.
.*
.*  EXECUTE SPAR TO FIND STRESS DERIVATIVES
.*
SPAR,RSOUT,SPAROUT.
REPLACE,SPARLD=SAVELD.
.*
.*  TEST TO SEE IF BEAM OR PLATE FORCES AND MOMENTS
.*          NEED CONVERTING TO STRESSES
.*
IFE,(NSUB.NE.0),FAMS.
REWIND,FAM2STR,SPARLIB,CNMNIO,SPARLD,INPT,CONS.
LDSET(LIB=SPARLIB,PRESET=ZERO)
FAM2STR,INPT,CONS,CNMNIO.
REPLACE,SPARLD=SAVELD.
ENDIF,FAMS.
RETURN,SPAROUT,EDOUT,RSOUT.
```

APPENDIX C

PRCEPXX

```
.PROC,PRCEPXX,BLK.
.*
.*  THIS PROCEDURE FILE EXECUTES THE END PROCESSOR
.*
REWIND,ENDPROC,ENDIN,CNMNIO,BLK,SPARLIB,SPARLD.
LDSET(LIB=SPARLIB,PRESET=ZERO)
ENDPROC,ENDIN,CNMNIO,BLK.
RETURN,SPARLA,SPARLC,SPARLD.
```

APPENDIX C

PRCEND

```
.PROC,PRCEND.
.*
.*  THIS PROCEDURE FILE OUTPUTS IMPORTANT FILES
.*
REWIND,SPIN1,SPOUT1,RRS,SPAROUT,CONOUT,CNMNIO.
COPYSBF,CONOUT,TEMPL.
COPYSBF,CNMNIO,TEMPL.
COPYSBF,SPIN1,TEMPL.
COPYSBF,SPOUT1,TEMPL.
COPYSBF,RRS,TEMPL.
COPYSBF,SPAROUT,TEMPL.
PACK(TEMPL)
REWIND,TEMPL.
REPLACE,TEMPL.
REWIND,TEMPL.
COPYSBF,TEMPL,OUTPUT.
```

APPENDIX D

PROGRAM LISTINGS

CONMS1

This is a main driver program for the CONMIN subroutine library.

```
      PROGRAM CONMS1(INPUT,OUTPUT,TAPE8,TAPE7,TAPE9,TAPE11,
     1 TAPE10,TAPE12,TAPE5=INPUT,TAPE6=OUTPUT)
C     NEW CONMIN SUBROUTINE
      COMMON/CNMN1/DELFUN,DABFUN,FDCH,FDCHM,CT,CTMIN,CTL,CTLMIN,
     1ALPHAX,ABOBJ1,THETA,OBJ,NDV,NCON,NSIDE,IPRINT,NFDG,NSCAL,
     2LINOBJ,ITMAX,ITRM,ICNDIR,IGOTO,NAC,INFO,INFOG,ITER
      COMMON/CNMN2/RDUM(50),IDUM(25)
      COMMON X(20),VLB(20),VUB(20),G(400),SCAL(20),DF(20),
     1A(20,200),S(20),G1(400),G2(400),B(200,200),C(200),ISC(400),
     2IC(200),MS1(400)
      COMMON/CONSAV/RSAV(50),ISAV(25)
      NAMELIST/CONPAR/IPRINT,NDV,ITMAX,NCON,NFDG,NSIDE,ICNDIR,
     1NSCAL,LINOBJ,ITRM,FDCH,FDCHM,CT,CTMIN,CTL,CTLMIN,
     2THETA,PHI,DELFUN,DABFUN,ISC,N1,N2,N3,N4,N5,
     3ALPHAX,ABOBJ1,IGOTO,VLB,VUB
      NAMELIST/STARTX/X
      NAMELIST/SAVE/IPRINT,NDV,ITMAX,NCON,NSIDE,ICNDIR,NSCAL,NFDG,
     1FDCH,FDCHM,CT,CTMIN,CTL,CTLMIN,THETA,PHI,NAC,DELFUN,DABFUN,
     2LINOBJ,ITRM,ITER,INFOG,IGOTO,INFO,OBJ,
     3RDUM,IDUM,
     4X,DF,G,ISC,IC,A,S,G1,G2,C,MS1,B,VLB,VUB,SCAL,RSAV,ISAV,NCOUNT
     5,N1,N2,N3,N4,N5,ALPHAX,ABOBJ1
      NAMELIST/LINKF/NDV,X
      NAMELIST/LINKE/OBJ,G
      NAMELIST/PASSAGE/NPASS
      READ(10,PASSAGE)
C     FIRST PASS,NPASS=1,SUBSEQUENTLY NPASS=2.
      GO TO(100,200),NPASS
  100 CONTINUE
      READ(5,CONPAR)
      READ(8,STARTX)
      REWIND 8
      GO TO 201
  200 CONTINUE
      READ(7,SAVE)
      REWIND 7
      READ(9,LINKE)
      REWIND 9
```

CONMS1 (Conc.)

```
  201 CONTINUE
      NPASS=2
      REWIND 10
      WRITE(10,PASSAGE)
C     SOLVE OPTIMIZATION
      CALL CONMIN(X,VLB,VUB,G,SCAL,DF,A,S,G1,G2,B,C,
     *ISC,IC,MS1,N1,N2,N3,N4,N5)
C     FUNCTION AND CONSTRAINT VALUES
      WRITE(7,SAVE)
      WRITE(9,LINKF)
C     WRITE CONTROL CARD STORED IN PROCFILE GONOGO.
  101 FORMAT(*GOTO,2.*)
      IF(IGOTO.EQ.0)WRITE(11,101)
C     WRITE ON TAPE8 IF GRADIENTS ARE REQUIRED
      IF(INFO.EQ.2) WRITE(12,102)
 102  FORMAT(* INFO = 2*)
      IF(INFO.EQ.2)WRITE(8,STARTX)
      STOP
      END
```

APPENDIX D

CONMS2

This is a main driver program for the CNMIN subroutine library.

```
      PROGRAM CONMS2(INPUT,OUTPUT,TAPE7,TAPE8,TAPE9,TAPE10,
     1 TAPE11,TAPE12,TAPE5=INPUT,TAPE6=OUTPUT)
C     NEW CONMIN SUBROUTINE
      COMMON/CNMN1/DELFUN,DABFUN,FDCH,FDCHM,CT,CTMIN,CTL,CTLMIN,
     1ALPHAX,ABOBJ1,THETA,OBJ,NDV,NCON,NSIDE,IPRINT,NFDG,NSCAL,
     2LINOBJ,ITMAX,ITRM,ICNDIR,IGOTO,NAC,INFO,INFOG,ITER
      COMMON X(20),VLB(20),VUB(20),G(400),SCAL(20),DF(20),
     1A(20,200),S(20),G1(400),G2(400),B(200,200),C(200),ISC(400),
     2IC(200),MS1(400)
      DIMENSION XI(20),GI(400),GRDOBJ(20),GRDG(20,400)
      NAMELIST/CONPAR/IPRINT,NDV,ITMAX,NCON,NFDG,NSIDE,ICNDIR,
     1NSCAL,LINOBJ,ITRM,FDCH,FDCHM,CT,CTMIN,CTL,CTLMIN,
     2THETA,PHI,DELFUN,DABFUN,ISC,N1,N2,N3,N4,N5,
     3ALPHAX,ABOBJ1,IGOTO,VLB,VUB
      NAMELIST/STARTX/X,XINC
      NAMELIST/BLK/OBJ,OBJI,X,XI,G,GI,GRDOBJ,GRDG,ICOUNT
      NAMELIST/CNT/OBJ1,OBJ2,OBJ3,TOL
      BL = .7
      BU = 1.3
      LOOPCT = 50
      READ(5,CONPAR)
      READ(7,STARTX)
      READ(8,BLK)
      READ(9,CNT)
      REWIND 7
      REWIND 8
      REWIND 9
      NSCON = NCON+1
      NCON = NCON+(2*NDV)
      DO 28 NC = NSCON,NCON
      ISC(NC) = 1
   28 CONTINUE
      DO 100 LOOP=1,LOOPCT
C     SOLVE OPTIMIZATION
      CALL CONMIN(X,VLB,VUB,G,SCAL,DF,A,S,G1,G2,B,C,
     *ISC,IC,MS1,N1,N2,N3,N4,N5)
C     FUNCTION AND CONSTRAINT VALUES
      IF(INFO.EQ.2) GO TO 24
      OBJ=OBJI
      DO 9 IJ=1,NCON
    9 G(IJ)=GI(IJ)
      DO 10 I=1,NDV
      DXI=X(I)-XI(I)
      OBJ=OBJ+GRDOBJ(I)*DXI
      DO 20 J=1,NCON
      G(J)=G(J)+GRDG(I,J)*DXI
```

CONMS2 (Conc.)

```
20   CONTINUE
10   CONTINUE
     NSDV = 0
      DO 15 NC = NSCON,NCON,2
     NSDV = NSDV+1
     G(NC) = 1.-X(NSDV)/(BL*XI(NSDV))
     G(NC+1) = X(NSDV)/(BU*XI(NSDV))-1.
15   CONTINUE
     GO TO 70
24   CONTINUE
     DO 23 IDF = 1,NDV
     DF(IDF) = GRDOBJ(IDF)
23   CONTINUE
     NSDV = 0
     DO 25 NC=NSCON,NCON,2
     NSDV = NSDV+1
     GRDG(NSDV,NC) = -1./(BL*XI(NSDV))
     GRDG(NSDV,NC+1) = 1./(BU*XI(NSDV))
25   CONTINUE
     NAC=0
     DO 30 J=1,NCON
     IF(G(J).LT.CTL)GO TO 30
     NAC=NAC+1
     IC(NAC)=J
30   CONTINUE
     DO 40 II=1,NDV
     DO 50 JJ=1,NAC
     J1=IC(JJ)
     A(II,JJ)=GRDG(II,J1)
50   CONTINUE
40   CONTINUE
70   CONTINUE
     IF(IGOTO.EQ.0)GO TO 200
100  CONTINUE
200  CONTINUE
     WRITE(7,STARTX)
     TOL=DELFUN
     OBJ3=OBJ2
     OBJ2=OBJ1
     OBJ1=OBJ
     WRITE(9,CNT)
     DA=ABS((OBJ3-OBJ2)/OBJ2)
     DB=ABS((OBJ2-OBJ1)/OBJ2)
     IF(DA.LE.TOL.AND.DB.LE.TOL)WRITE(10,1)
   1 FORMAT(*TERMINATED.*)
     STOP
     END
```

FPROC

This is an example listing of a front processor program.

```
      PROGRAM FPFGS1(INPUT,OUTPUT,TAPE7,TAPE5=INPUT,TAPE6=OUTPUT)
C     SPAR FRONT PROCESSOR READS DESIGN VARIABLES AND
C     PRINTS THEM IN SPAR SECTION PROPERTY FORMAT
C     FUSELAGE MADE OF ROD BEAM MEMBRANE ELEMENTS
      DIMENSION X(50)
      NAMELIST/LINKF/NDV,X
C     NDV=NUMBER OF DESIGN VARIABLES
C     X(NDV)=DESIGN VARIABLES
C     DV'S ARE X(1)=SECTIONAL AREA OF STRINGER RODS
C              X(2)=NONDIMENSIONAL AREA OF BEAM
C              X(3)=THICKNESS OF MEMBRANE PANEL
      READ(7,10) B10,B20,TO
   10 FORMAT(3F10.3)
      READ(5,LINKF)
      XIN1=1./X(1)
      XIN2=1./X(2)
      XIN3=1./X(3)
C     WRITE E23 ELEMENTS
      PRINT 200
  200 FORMAT(* E23 SECTION PROPERTIES*)
      PRINT 201,XIN1
  201 FORMAT(* 1 *,F8.3)
C     COMPUTE VALUES FOR DSY CARDS
C     WRITE E21 ELEMENTS
      PRINT 202
  202 FORMAT(* E21 SECTION PROPERTIES*)
      AREAO=(2.*B10+B20)*TO
      AREA=AREAO*XIN2
      SCALE=SQRT(AREA/AREAO)
      B1=B10*SCALE
      B2=B20*SCALE
      T=TO*SCALE
      EI1=B1*(B2+2.*T)**3/12.-(B1-T)*B2**3/12.
      ALPHA1=0.
      C=(B2+2.*T)*B1**2/2.-B2*(B1-T)*(B1+T)/2.
      C=C/AREA
      EI2=2.*T*B1**3/12.+2.*T*B1*(B1/2.-C)**2
     1+B2*T**3/12.+B2*T*(C-T/2.)**2
      ALPHA2=0.
      F=(2.*B1+B2)*T**3/3.
      F1=0.
      Z1=((B1*B2)**2)*T/(4.*EI1)+C-T/2.
      Z2=0.
      THETA=0.
      Q1=0.
      Q2=0.
      Q3=0.
      Y11=-(B1-C)
      Y12=.5*B2+T
```

FPROC (Conc.)

```
      Y21=C
      Y22=.5*B2+T
      Y31=C
      Y32=-(.5*B2+T)
      Y41=-(B1-C)
      Y42=-(.5*B2+T)
      J=1
      PRINT 103,J,EI1,ALPHA1,EI2,ALPHA2,AREA
      PRINT 1003,F,F1,Z1,Z2,THETA
  103 FORMAT(*DSY*,I2,5E12.4,*%*)
 1003 FORMAT(1X,5E13.5)
      PRINT 104,Q1,Q2,Q3,Y11,Y12,Y21
  104 FORMAT(1X,6E12.4,*%*)
      PRINT 1004,Y22,Y31,Y32,Y41,Y42
 1004 FORMAT(1X,5E12.4)
C     WRITE E41 ELEMENTS
      PRINT 300,XIN3
  300 FORMAT(* SHELL SECTION PROPERTIES*/* 1*,F8.3)
      STOP
      END
```

APPENDIX D

EPROC

This is an example listing of an end processor program.

```
      PROGRAM EPFUS(INPUT,TAPE6,TAPE8,TAPE5=INPUT,OUTPUT)
      DIMENSION A(4500),B(1400)
      DIMENSION A1(400),B1(400),C1(400)
      DIMENSION G(400)
      NAMELIST/EPIN/E23AL,E21AL,E41AL,NSE23,NSE21,NSE41
      NAMELIST/LINKE/OBJ,G
      READ(5,EPIN)
      CALL DAL(4,11,A(1),0,IEA,KADR,IERR,NWDS,NE,LB,ITYPE,
     14HOBJF,3HAUS,1,1)
      OBJ=A(1)
      CALL DAL(4,11,A(1),0,IEA,KADR,IERR,NWDS,NE,LB,ITYPE,
     14HSTRS,3HE23,1,1)
      I1=1
      INL=6*NSE23
      DO 2 IN=6,INL,6
      I=I1
      A1(I1)=A(IN)
      G(I)=ABS(A1(I1))/E23AL-1.
      I1=I1+1
    2 CONTINUE
      CALL DAL(4,11,A(1),0,IEA,KADR,IERR,NWDS,NE,LB,ITYPE,
     14HSTRS,3HE21,1,1)
      J1=1
      KNL1=(NSE21-1)*52 + 6
      DO 4 IM=5,KNL1,52
      B1(J1)=ABS(A(IM))
      A1(J1)=ABS(A(IM+1))
      I=I1+J1-1
      GNUM=B1(J1)
      IF(A1(J1).GT.B1(J1))GNUM=A1(J1)
      G(I)=GNUM/E21AL-1.
      J1=J1+1
4     CONTINUE
      CALL DAL(4,11,A(1),0,IEA,KADR,IERR,NWDS,NE,LB,ITYPE,
     14HSTRS,3HE41,1,1)
      K1=1
      LNL1=(NSE41*23)
      DO 10 NP=21,LNL1,23
      C1(K1)=A(NP)
      B1(K1)=A(NP+1)
      A1(K1)=A(NP+2)
      I=I1+J1+K1-2
      GNUM=SQRT(C1(K1)**2+B1(K1)**2-C1(K1)*B1(K1)+3.*A1(K1)**2)
      G(I)=GNUM/E41AL-1.
      K1=K1+1
10    CONTINUE
      CALL FIN(0,0)
      REWIND 6
      WRITE(6,LINKE)
      STOP
      END
```

APPENDIX D

EVALS

```
      PROGRAM EVALS(INPUT,OUTPUT,TAPE5=INPUT,TAPE7,
     1TAPE8,TAPE9,TAPE10,TAPE11)
      DIMENSION X(20),XI(20),G(400),GI(400),GRDOBJ(20),GRDG(20,400)
      DIMENSION ISC(400),VLB(20),VUB(20)
      NAMELIST/CONPAR/IPRINT,NDV,ITMAX,NCON,NFDG,NSIDE,ICNDIR,
     1NSCAL,LINOBJ,ITRM,FDCH,FDCHM,CT,CTMIN,CTL,CTLMIN,
     2THETA,PHI,DELFUN,DABFUN,ISC,N1,N2,N3,N4,N5,
     3ALPHAX,ABOBJ1,IGOTO,VLB,VUB
      NAMELIST/PASSAGE/NPASS
      NAMELIST/STARTX/X,XINC
      NAMELIST/LINKE/OBJ,G
      NAMELIST/LINKF/NDV,X
      NAMELIST/BLK/OBJ,OBJI,X,XI,G,GI,GRDOBJ,GRDG,ICOUNT
      READ(5,CONPAR)
      REWIND 5
      READ(7,PASSAGE)
      REWIND 7
      READ(8,STARTX)
      REWIND 8
      GO TO(100,200),NPASS
  100 CONTINUE
      ICOUNT=0
      DO 1000 I=1,NDV
      XI(I)=0.0
      G(I)=0.0
      GI(I)=0.0
      GRDOBJ(I)=0.0
      DO 1001 J=1,NCON
      GRDG(I,J)=0.0
 1001 CONTINUE
 1000 CONTINUE
      OBJ=0.0
      OBJI=0.0
      NPASS=2
      WRITE(7,PASSAGE)
      GO TO 201
  200 CONTINUE
      READ(10,BLK)
      REWIND 10
      READ(9,LINKE)
      REWIND 9
      IF(ICOUNT.NE.1)GO TO 300
      OBJI=OBJ
      DO 10 J=1,NDV
   10 XI(J)=X(J)
      DO 20 K=1,NCON
   20 GI(K)=G(K)
      GO TO 400
  300 I=ICOUNT-1
```

EVALS (Conc.)

```
      DELTX=X(I)-XI(I)
      GRDOBJ(I)=(OBJ-OBJI)/DELTX
      DO 30 L=1,NCON
      GRDG(I,L)=(G(L)-GI(L))/DELTX
   30 CONTINUE
      X(I)=XI(I)
  400 X(ICOUNT)=X(ICOUNT)*(1.-XINC)
      LIM=NDV+1
      IF(ICOUNT.EQ.LIM)WRITE(11,401)
      REWIND 11
  401 FORMAT(*GOTO,7.*)
  201 CONTINUE
      ICOUNT=ICOUNT+1
      WRITE(10,BLK)
      REWIND 10
      WRITE(9,LINKF)
      REWIND 9
      STOP
      END
```

FSDS

```
      PROGRAM FSDS(INPUT,OUTPUT,TAPE8,TAPE7,TAPE9,TAPE11,
     1 TAPE10,TAPE5=INPUT,TAPE6=OUTPUT)
C     NEW CONMIN SUBROUTINE
      COMMON/CNMN1/DELFUN,DABFUN,FDCH,FDCHM,CT,CTMIN,CTL,CTLMIN,
     1ALPHAX,ABOBJ1,THETA,OBJ,NDV,NCON,NSIDE,IPRINT,NFDG,NSCAL,
     2LINOBJ,ITMAX,ITRM,ICNDIR,IGOTO,NAC,INFO,INFOG,ITER
      COMMON/CNMN2/RDUM(50),IDUM(25)
      COMMON X(20),VLB(20),VUB(20),G(400),SCAL(20),DF(20),
     1A(20,200),S(20),G1(400),G2(400),B(200,200),C(200),ISC(400),
     2IC(200),MS1(400)
      COMMON/CONSAV/RSAV(50),ISAV(25)
      NAMELIST/CONPAR/IPRINT,NDV,ITMAX,NCON,NFDG,NSIDE,ICNDIR,
     1NSCAL,LINOBJ,ITRM,FDCH,FDCHM,CT,CTMIN,CTL,CTLMIN,
     2THETA,PHI,DELFUN,DABFUN,ISC,N1,N2,N3,N4,N5,
     3ALPHAX,ABOBJ1,IGOTO,VLB,VUB
      NAMELIST/STARTX/X
      NAMELIST/SAVE/IPRINT,NDV,ITMAX,NCON,NSIDE,ICNDIR,NSCAL,NFDG,
     1FDCH,FDCHM,CT,CTMIN,CTL,CTLMIN,THETA,PHI,NAC,DELFUN,DABFUN,
     2LINOBJ,ITRM,ITER,INFOG,IGOTO,INFO,OBJ,
     3RDUM,IDUM,
     4X,DF,G,ISC,IC,A,S,G1,G2,C,MS1,B,VLB,VUB,SCAL,RSAV,ISAV,NCOUNT
     5,N1,N2,N3,N4,N5,ALPHAX,ABOBJ1,OBJ1,OBJ2,OBJ3
      NAMELIST/LINKF/NDV,X
      NAMELIST/LINKE/OBJ,G
      NAMELIST/PASSAGE/NPASS
      READ(10,PASSAGE)
C     FIRST PASS,NPASS=1,SUBSEQUENTLY NPASS=2.
      GO TO(100,200),NPASS
  100 CONTINUE
      READ(5,CONPAR)
      READ(8,STARTX)
      DO 300 K=1,NCON
      G(I)=0.0
  300 CONTINUE
      IGOTO=1
      OBJ1=1.
      OBJ2=1.
      OBJ3=1.
      GO TO 201
  200 CONTINUE
      READ(7,SAVE)
      REWIND 7
      READ(9,LINKE)
      OBJ1=OBJ2
      OBJ2=OBJ3
      OBJ3=OBJ
      OINC1=ABS(1.-OBJ1/OBJ2)
      OINC2=ABS(1.-OBJ2/OBJ3)
      OINC3=ABS(1.-OBJ3/OBJ)
```

FSDS (Conc.)

```
      IF(OINC1.LT.DELFUN.AND.OINC2.LT.DELFUN.AND.
     1OINC3.LT.DELFUN)IGOTO=0
  201 CONTINUE
      NPASS=2
      REWIND 10
      WRITE(10,PASSAGE)
C     SOLVE FSD PROBLEM
      CALL FSDSUB(X,DF,G,ISC,IC,A,S,G1,G2,C,MS1,B,VLB,VUB
     1,SCAL,N1,N2,N3,N4,N5)
C     FUNCTION AND CONSTRAINT VALUES
      WRITE(7,SAVE)
      WRITE(9,LINKF)
C     WRITE CONTROL CARD STORED IN PROCFILE GONOGO.
  101 FORMAT(*GOTO,2.*)
      IF(IGOTO.EQ.0)WRITE(11,101)
      STOP
      END
```

APPENDIX D

FSDSUBS

```
      SUBROUTINE FSDSUB(X,DF,G,ISC,IC,A,S,G1,G2,C,MS1,B,VLB,VUB
     1,SCAL,N1,N2,N3,N4,N5)
      COMMON/CNMN1/DELFUN,DABFUN,FDCH,FDCHM,CT,CTMIN,CTL,CTLMIN,
     1 ALPHAX,ABOBJ1,THETA,OBJ,NDV,NCON,NSIDE,IPRINT,NFDG,NSCAL,
     2 LINOBJ,ITMAX,ITRM,ICNDIR,IGOTO,NAC,INFO,INFOG,ITER
      COMMON/CNMN2/RDUM(50),IDUM(25)
      DIMENSION X(20),VLB(20),VUB(20),G(400),SCAL(20),DF(20),
     1A(20,200),S(20),G1(400),G2(400),B(200,200),C(200),ISC(400),
     2IC(200),MS1(400)
      COMMON/CONSAV/RSAV(50),ISAV(25)
      WRITE(6,5) (X(I),I=1,NDV)
    5 FORMAT(1H1,*DESIGN VARIABLES INTO FSDSUB ARE*,/3(1X,E13.5))
      DO 1 I=1,NCON
      G(I)=G(I)+1.
    1 CONTINUF
      DO 20 I = 1,NDV
      X(I) = 1./X(I)
   20 CONTINUE
      XO=1./VUB(1)
      NSE23=58
      ISTRT=1
      IEND=ISTRT+NSE23
      DO 2 I=ISTRT,IEND
      XNEW=X(1)*G(I)
      IF(XNEW.LE.XO) GO TO 2
      XO=XNEW
      WRITE(6,12) I,G(I),XO
   12 FORMAT(1X,*CONSTRAINT NUMBER *,I5/1X,*CONSTRAINT = *,E13.5/
     1 1X,*NEW DESIGN VARIABLE = *,E13.5//)
    2 CONTINUE
      X(1)=XO
      XO=1./VUB(2)
      NSE21=76
      ISTRT=NSE23+1
      IEND=ISTRT+NSE21
      DO 4 I=ISTRT,IEND
      XNEW=X(2)*G(I)
      IF(XNEW.LE.XO) GO TO 4
      XO=XNEW
      WRITE(6,12) I,G(I),XO
    4 CONTINUE
      X(2)=XO
      XO=1./VUB(3)
      NSE41=56
      ISTRT=NSE23+NSE21+1
      IEND=ISTRT+NSE41
      DO 6 I=ISTRT,IEND
      XNEW=X(3)*G(I)
      IF(XNEW.LE.XO) GO TO 6
```

```
      XO=XNEW
      WRITE(6,12) I,G(I),XO
    6 CONTINUE
      X(3)=XO
      DO 30 I = 1,NDV
      X(I) = 1./X(I)
 30   CONTINUE
      WRITE(6,10) (X(I),I=1,NDV)
 10   FORMAT(////1X,*DESIGN VARIABLES FROM FSDSUB ARE*,/3(1X,E13.5))
      RETURN
      END
```

APPENDIX D

SELECTS

```
      PROGRAM SELECTS(INPUT,OUTPUT,TAPE5=INPUT,TAPE6)
      DIMENSION X(20),VLB(20),VUB(20),G(400),SCAL(20),DF(20),
     1A(20,200),S(20),G1(400),G2(400),B(200,200),C(200),ISC(400),
     2IC(200),MS1(400)
      DIMENSION XI(20),GI(400),GRDOBJ(20),GRDG(20,400)
      DIMENSION RDUM(50),RSAV(50),IDUM(25),ISAV(25)
      NAMELIST/SAVE/IPRINT,NDV,ITMAX,NCON,NSIDE,ICNDIR,NSCAL,NFDG,
     1FDCH,FDCHM,CT,CTMIN,CTL,CTLMIN,THETA,PHI,NAC,DELFUN,DABFUN,
     2LINOBJ,ITRM,ITER,INFOG,IGOTO,INFO,OBJ,
     3RDUM,IDUM,
     4X,DF,G,ISC,IC,A,S,G1,G2,C,MS1,B,VLB,VUB,SCAL,RSAV,ISAV,NCOUNT
     5,N1,N2,N3,N4,N5,ALPHAX,ABOBJ1
      NAMELIST/BLK/OBJ,OBJI,X,XI,G,GI,GRDOBJ,GRDG,ICOUNT
      READ(6,SAVE)
      REWIND 6
      READ(5,BLK)
      OBJ=OBJI
      DO 25 I=1,NDV
      X(I)=XI(I)
      DF(I)=GRDOBJ(I)
   25 CONTINUE
      NAC=0
      DO 30 J=1,NCON
      G(J)=GI(J)
      IF(G(J).LT.CTL)GO TO 30
      NAC=NAC+1
      IC(NAC)=J
   30 CONTINUE
      DO 40 II=1,NDV
      DO 50 JJ=1,NAC
      J1=IC(JJ)
      A(II,JJ)=GRDG(II,J1)
   50 CONTINUE
   40 CONTINUE
      WRITE(6,SAVE)
      STOP
      END
```

APPENDIX D

RERITES

```
PROGRAM RERITES(INPUT,TAPE5=INPUT,TAPE6)
DIMENSION X(20)
NAMELIST/LINKF/NDV,X
NAMELIST/STARTX/X,XINC
DATA XINC/0.1/
READ(5,LINKF)
WRITE(6,STARTX)
STOP
END
```

APPENDIX D

BLDELDS

```
      PROGRAM BLDELDS(TAPE5,TAPE23,TAPE20,TAPE21,TAPE22,OUTPUT)
C
C   THIS PROGRAM CREATES A RUNSTREAM THAT WILL CREATE
C   DMDV AND DKDV FOR PARTICULAR ELEMENTS USED DESIGN VARIABLES
C
      DIMENSION EL(999),NSECT(999),TNAME1(8),TNAME2(8),FOR(9)
      DIMENSION NODVPE(999)
      DATA E21,E22,E23,E41,E43,E44/3HE21,3HE22,3HE23,3HE41,3HE43,3HE44/
      DATA E31,E33/3HE31,3HE33/
      DATA TNAME1/4HDEF ,4HGD  ,4HGTIT,4HDIR ,4HNS  ,3*4HELTS/
      DATA TNAME2/5*4H    ,4HNAME,4HNNOD,4HISCT/
      DATA START,END,XNSECT/4H$STA,4H$END,4HNSEC/
      DATA YNSECT/3HNSE/
C
C   CALL SUBROUTINE TO REMOVE BEGINNING BLANKS
C
      CALL REMOVE
C
C   READ INPUT
C
C      NOEL=NUMBER OF ELEMENTS
C      NODV=NUMBER OF DESIGN VARIABLE ELEMENTS
C      VORB=TYPE OF ANALYSIS (EX. BUCKLING)
C      NDF=NUMBER OF DEGREES OF FREEDOM PER JOINT
C      EL - ELEMENT NAMES CONTAINING DESIGN VARIABLES
C                    (EX. E21)
C      NSECT - LAST SECTION NUMBER USED FOR EACH DESIGN VARIABLE
C      NODVPE - NUMBER OF DESIGN VARIABLES PER ELEMENT
C
      READ(5,5) NOLC,NODV,ISNOLC,JOINTS,NDF,NOEL,VORB
    5 FORMAT(6(1X,I4),1X,A4)
      READ(5,6) (EL(I),NSECT(I),NODVPE(I),I=1,NOEL)
    6 FORMAT(6(1X,A3,1X,I3,1X,I3))
      NDF=NDF*NDF
      WRITE(20,2)
    2 FORMAT(*[XQT TAB*/* UPDATE=1*)
C
C   LOOP ON NUMBER OF ELEMENTS
C
      DO 30 I = 1,NOEL
C
C   CALL SUBROUTINE TO UPDATE TAB BY SETTING DESIGN VARIABLES
C   TO UNITY.
C   KCNT RETURNS THE NUMBER OF POSSIBLE DESIGN VARIABLES FOR
C   A PARTICULAR ELEMENT.
C   KCNT=99 MEANS THE ELEMENT NAME IS BAD
C
      CALL TABNPUT(EL(I),NSECT(I),KCNT)
      IF(KCNT.NE.99) GO TO 30
      PRINT 29,EL(I)
```

```
  29   FORMAT(* ELEMENT NAME *,A3,* DOES NOT EXIST*)
       GO TO 190
  30   CONTINUE
C
C  CONCLUDE UPDATE AND SET UP DISABLES
C
       WRITE(20,31)
  31   FORMAT(* UPDATE=0*)
       WRITE(20,32)
  32   FORMAT(*[XQT DCU*/* COPY 1,2*)
       DO 35 I = 1,8
       IF(I.GT.4) GO TO 33
       DO 37 J = 1,NOEL
       TNAME2(I)=EL(J)
       WRITE(20,34) TNAME1(I),TNAME2(I)
  37   CONTINUE
       GO TO 35
  33   WRITE(20,34) TNAME1(I),TNAME2(I)
  34   FORMAT(* DISABLE 1,*,A4,1X,A4)
  35   CONTINUE
C
C  SET COUNTER FOR TOTAL NUMBER OF DESIGN VARIABLES
C
       IELCNT = 0
       ICNTDV = 1
       ISW=0
       ISAVCNT = 0
C
C  READ IN RUNSTREAM AND CHECK FOR START OF A DESIGN VARIABLE
C
  39   JCNT = 0
  40   READ(21,50) (FOR(J),J=1,9)
  50   FORMAT(A4,A6,7A10)
       IF(EOF(21)) 170,60
  60   IF(FOR(1).EQ.START) GO TO 70
       WRITE(20,50) (FOR(J),J=1,9)
       GO TO 40
  70   IF(ISW.EQ.1) WRITE(20,75)
  75   FORMAT(*[XQT ELD*)
       ISW=1
       READ(21,80) (FOR(J),J=1,9)
  80   FORMAT(A3,A7,7A10)
C
C  SET ICNT = NUMBER OF POSSIBLE DESIGN VARIABLES FOR AN ELEMENT
C
       ICNT=99
C
C  DETERMINE POSSIBLE NUMBER OF DESIGN VARIABLES PER ELEMENT
```

```
C
      IF(FOR(1).EQ.E23.OR.FOR(1).EQ.E41.OR.FOR(1).EQ.E44) ICNT=1
      IF(FOR(1).EQ.E31) ICNT=1
      IF(FOR(1).EQ.E21) ICNT=4
      IF(FOR(1).EQ.E43.OR.FOR(1).EQ.E33) ICNT=12
      IF(FOR(1).EQ.E22) ICNT=21
      TNAME=FOR(1)
      IF(ICNT.EQ.99) TNAME=SAVNAM
      JCNT = JCNT+1
      IF(ICNT.NE.99) IELCNT=IELCNT+1
C
C  SET UNIT = 20 IF ONLY ONE DESIGN VARIABLE PER ELEMENT
C  OTHERWISE SET UNIT = 22 (SCRATCH UNIT)
C
      IUNIT=22
      REWIND 22
      IF(ICNT.EQ.1) IUNIT=20
      IF(ICNT.EQ.99.AND.ISAVCNT.EQ.1) IUNIT=20
C
C CHECK FOR REPEAT OF ELEMENT NAME
C
      IF(ICNT.NE.99) GO TO 86
      WRITE(IUNIT,85) SAVNAM
 85   FORMAT(A3,77X)
      GO TO 860
C
C  READ DATA FROM UNIT 21 AND WRITE DATA ON UNIT 20 OR 22
C  DEPENDING UPON VALUE OF ICNT
C
 86   WRITE(IUNIT,50) (FOR(J),J=1,9)
      SAVNAM=FOR(1)
      ISAVCNT=ICNT
      GO TO 87
860   ICNT = ISAVCNT
      GO TO 870
 87   READ(21,50) (FOR(J),J=1,9)
870   IF(FOR(1).EQ.XNSECT.OR.FOR(1).EQ.YNSECT) GO TO 88
      IF(FOR(1).EQ.END) GO TO 110
      WRITE(IUNIT,50) (FOR(J),J=1,9)
      GO TO 87
 88   IP1=NSECT(IELCNT)+JCNT
      WRITE(IUNIT,90) IP1
 90   FORMAT(*NSECT=*,I3,71X)
100   READ(21,50) (FOR(J),J=1,9)
      IF(FOR(1).EQ.END) GO TO 110
      WRITE(IUNIT,50) (FOR(J),J=1,9)
      GO TO 100
C
C  SKIP THIS IF MORE THAN ONE DESIGN VARIABLE PER ELEMENT
```

```
C
 110   IF(ICNT.NE.1) GO TO 120
C
C  CALL SUBROUTINE TO CREATE REMAINDER OF RUNSTREAM
C
       CALL CRRS(ICNTDV,0,0,TNAME,NDF,NODVPE(IELCNT))
       ICNTDV = ICNTDV+NODVPE(IELCNT)
       GO TO 39
C
C  LOOP ON POSSIBLE NUMBER OF DESIGN VARIABLES PER ELEMENT
C          READ FROM UNIT 22
C
C          WRITE ON UNIT 20
C
 120   DO 160 I = 1,ICNT
       IF(I.NE.1) WRITE(20,75)
       REWIND 22
 130   READ(22,50) (FOR(J),J=1,9)
       IF(EOF(22)) 150,140
 140   IF(FOR(1).EQ.XNSECT.OR.FOR(1).EQ.YNSECT) GO TO 141
       WRITE(20,50) (FOR(J),J=1,9)
       GO TO 130
 141   WRITE(20,90) IP1
       GO TO 130
C
C  CALL SUBROUTINE TO CREATE REMAINDER OF RUNSTREAM
C
 150   CALL CRRS(ICNTDV,ICNT,I,TNAME,NDF,NODVPE(IELCNT))
       IP1=IP1+1
 160   CONTINUE
       ICNTDV = ICNTDV+NODVPE(IELCNT)
       GO TO 39
 170   WRITE(20,180)
 180   FORMAT(* TOC 2*/*[XQT EXIT*)
 190   STOP
       END
       SUBROUTINE TABNPUT(ELNAME,NSCT,ICNT)
C
C  THIS SUBROUTINE CREATES TAB PROCESSOR INPUT FOR A RUNSTREAM
C  DEPENDING UPON ELEMENTS USED.
C  ALL DESIGN VARIABLES ARE SET TO UNITY.
C  ICNT = NUMBER OF DESIGN VARIABLES FOR A PARTICULAR ELEMENT
C  ICNT = 99 MEANS THE ELEMENT NAME IS BAD
C
       DIMENSION ELEMENT(21)
       DATA E21,E22,E23,E41,E43,E44/3HE21,3HE22,3HE23,3HE41,3HE43,3HE44/
       DATA E31,E33/3HE31,3HE33/
       DATA BA,BB,BC,SA,SB/2HBA,2HBB,2HBC,2HSA,2HSB/
       IF(ELNAME.EQ.E21) ELID=BA
```

```
      IF(ELNAME.EQ.E22) ELID=BB
      IF(ELNAME.EQ.E23) ELID=BC
      IF(ELNAME.EQ.E41.OR.ELNAME.EQ.E43.OR.ELNAME.EQ.E31.OR.
     1 ELNAME.EQ.E33) ELID=SA
      IF(ELNAME.EQ.E44) ELID=SB
      WRITE(20,10) ELID
   10 FORMAT(2X,A2)
      ICNT=99
      IF(ELNAME.EQ.E23.OR.ELNAME.EQ.E41.OR.ELNAME.EQ.E44) GO TO 30
      IF(ELNAME.EQ.E31) GO TO 30
      IF(ELNAME.EQ.E21) GO TO 50
      IF(ELNAME.EQ.E43.OR.ELNAME.EQ.E33) GO TO 100
      IF(ELNAME.EQ.E22) GO TO 150
      PRINT 20,ELNAME
   20 FORMAT(* ELEMENT NAME *,A3,* DOES NOT EXIST*)
      GO TO 210
C
C  ELEMENT NAMES E23 , E31 , E41 , E44
C          ICNT = 1
C
   30 K=NSCT+1
      WRITE(20,40) K
   40 FORMAT(1X,I3,* 1.0*)
      ICNT=1
      GO TO 210
C
C  ELEMENT NAME E21
C          ICNT = 4
C
   50 DO 90 I = 1,4
      DO 60 J = 1,10
      ELEMENT(J)=0.0
   60 CONTINUE
      IF(I.EQ.1) ELEMENT(1)=1.0
      IF(I.EQ.2) ELEMENT(3)=1.0
      IF(I.EQ.3) ELEMENT(5)=1.0
      IF(I.EQ.4) ELEMENT(6)=1.0
      K=I+NSCT
      WRITE(20,70) K,(ELEMENT(J),J=1,10)
   70 FORMAT(*  DSY *,I1,10(1X,F3.1))
      WRITE(20,80)
   80 FORMAT(2X,11(*0.0 *))
   90 CONTINUE
      ICNT=4
      GO TO 210
C
C  ELEMENT NAMES E33 , E43
C         ICNT = 12
C
```

```
 100   DO 140 I = 1,12
       DO 110 J = 1,12
       ELEMENT(J)=0.0
 110   CONTINUE
       ELEMENT(1)=1.E-06
       ELEMENT(3)=1.E-06
       ELEMENT(6)=1.E-06
       ELEMENT(7)=1.E-06
       ELEMENT(9)=1.E-06
       ELEMENT(12)=1.E-06
       ELEMENT(I)=1.
       K=I+NSCT
       IF(I.EQ.1) WRITE(20,115)
 115   FORMAT(* FORMAT=UNCOUPLED*)
       WRITE(20,120) K
 120   FORMAT(2X,I2,1X,9(*1.0 *))
       WRITE(20,130) (ELEMENT(J),J=1,6)
 130   FORMAT(1X,6(1X,E10.2))
       WRITE(20,130) (ELEMENT(J),J=7,12)
 140   CONTINUE
       ICNT=12
       GO TO 210
C
C  ELEMENT NAME E22
C         ICNT = 21
C
 150   DO 200 I = 1,21
       DO 160 J = 1,21
       ELEMENT(J)=0.0
 160   CONTINUE
       K=I+NSCT
       ELEMENT(I)=1.
       WRITE(20,170) K,ELEMENT(1)
 170   FORMAT(2X,I2,1X,E10.2)
       KK=2
       DO 190 J = 2,6
       L=KK+J-1
       WRITE(20,180) (ELEMENT(N),N=KK,L)
 180   FORMAT(2X,6(F3.1,1X))
       KK=KK+J
 190   CONTINUE
 200   CONTINUE
       ICNT=21
 210   RETURN
       END
       SUBROUTINE CRRS(ICNTDV,ICNT,J,TNAME,NDF,NODVEL)
C
C  THIS SUBROUTINE CREATES REMAINDER OF RUNSTREAM
C  TO FIND DMDV AND DKDV.
```

```
C         USE NAMES
C            DMDV DIAG 0 I AND
C            DKDV SPAR 25 I
C         WHERE I = 1 TO NUMBER OF POSSIBLE DESIGN VARIABLES
C
      DIMENSION BA(4),BB(21),SAEL(12),TNAME1(9),TNAME2(9)
      DATA DK,DM,CK,CM,SK,SM/2HDK,2HDM,2HCK,2HCM,2HSK,2HSM/
      DATA DV/2HDV/
      DATA BB/2H11,2H21,2H22,2H31,2H32,2H33,2H41,2H42,2H43,
     1 2H44,2H51,2H52,2H53,2H54,2H55,2H61,2H62,2H63,2H64,2H65,
     2 2H66/
      DATA SAEL/2H11,2H12,2H22,2H13,2H23,2H33,2H44,2H45,2H55,
     1 2H46,2H56,2H66/
      DATA DEM/10HDEM DIAG 0/
      DATA KSP/7HK SPAR /
      DATA CHA/9HCHANGE 2,/
      DATA COP/9HCOPY 1,2 /
      DATA BA/2HIX,2HIY,2HDA,2HJO/
      DATA TNAME1/4HDEF ,4HGD  ,4HGTIT,4HELTS,4HNS  ,4HKMAP,
     1 4HAMAP,4HMASK,4HDIR /
      DATA TNAME2/3*4H    ,4HMASK,3*4H    ,4HEFIL,4H    /
      DATA DMDV,DKDV,SPAR/4HDMDV,4HDKDV,6H SPAR /
      DATA DIAG/6H DIAG /
      JCNTDV = ICNTDV
C
C  XQT E , EKS , TOPO , K , DCU
C
      WRITE(20,10)
   10 FORMAT(*[XQT E*/*[XQT EKS*/*[XQT TOPO*/*[XQT K*/*[XQT DCU*)
C
C  DISABLE DATA SETS
C
      DO 30 I = 1,9
      IF(I.EQ.8) TNAME1(8)=TNAME
      IF(I.EQ.1.OR.I.EQ.2.OR.I.EQ.3.OR.I.EQ.9) TNAME2(I)=TNAME
      WRITE(20,20) TNAME1(I),TNAME2(I)
   20 FORMAT(* DISABLE 1,*A4,2X,A4)
   30 CONTINUE
C
C  SET UP ELEMENT NAMES FOR CHANGE AND COPY STATEMENTS
C
      DDV=DV
      DDK=DK
      DDM=DM
      IF(ICNT.NE.4) GO TO 31
      DDV=BA(J)
      GO TO 39
   31 IF(ICNT.NE.12) GO TO 32
      DDK=CK
```

BLDELDS (Conc.)

```
      DDM=CM
      DDV=SAEL(J)
      GO TO 39
   32 IF(ICNT.NE.21) GO TO 39
      DDK=SK
      DDM=SM
      DDV=BB(J)
   39 DO 80 IJK = 1,NODVEL
      WRITE(20,40) COP,DEM
   40 FORMAT(A9,A10,* 0*)
      WRITE(20,50) CHA,DEM,DDM,DDV,DIAG, JCNTDV
   50 FORMAT(A9,A10,* 0,*,2A2,A6,*0 *,I3)
      WRITE(20,60) COP,KSP,NDF
   60 FORMAT(A9,A7,I2,* 0*)
      WRITE(20,70) CHA,KSP,NDF,DDK,DDV,SPAR,NDF,JCNTDV
   70 FORMAT(A9,A7,I2,* 0,*,2A2,A6,I2,I4)
      JCNTDV = JCNTDV+1
   80 CONTINUE
      RETURN
      END
      SUBROUTINE REMOVE
C
C  THIS SUBROUTINE REMOVES THE LEADING BLANKS FROM EACH
C  LINE IN THE RUNSTREAM.
C
      DIMENSION DATIN(80)
      DATA BLANK/1H /
      REWIND 23
    1 READ(23,2) (DATIN(I),I=1,80)
    2 FORMAT(80A1)
      IF(EOF(23)) 7,3
    3 IF(DATIN(1).NE.BLANK) GO TO 50
      DO 6 I = 2,80
      IF(DATIN(I).NE.BLANK) GO TO 60
    6 CONTINUE
   60 L=I-1
      K=80-I
      DO 4 J = 1,K
      DATIN(J) = DATIN(J+L)
    4 CONTINUE
      K=K+1
      DO 5 J = K,80
      DATIN(J) = BLANK
    5 CONTINUE
   50 WRITE(21,2) (DATIN(J),J=1,80)
      GO TO 1
    7 REWIND 21
      RETURN
      END
```

APPENDIX D

GNGRDRS

```
      PROGRAM GNGRDRS(INPUT,TAPE30,TAPE31,TAPE10,TAPE5=INPUT,OUTPUT)
C
C  THIS PROGRAM CREATES A REPEATABLE SPAR RUNSTREAM
C  FOR CALCULATING DERIVATIVES WITH RESPECT TO
C  DESIGN VARIABLES. THE SIZE OF THE RUNSTREAM VARIES
C  WITH THE NUMBER OF LOAD CASES (NOLC) AND THE
C  NUMBER OF DESIGN VARIABLES(NODV).
C
C  THE RUNSTREAM IS OUTPUT ON UNIT 10
C
      DIMENSION EL(999),NSECT(999),NUM(8),MFORM(5),NODVPE(999)
      DATA NUM/1H1,1H2,1H3,1H4,1H5,1H6,1H7,1H8/
      DATA BUCK,VIBR/4HBUCK,4HVIBR/
      DATA E21,E22,E33,E43/3HE21,3HE22,3HE33,3HE43/
      DATA MFORM(1)/10H(*Z11=UNIO/
      DATA MFORM(4)/10H(I1,1X,*Z*/
      DATA MFORM(5)/9H),I1,*)*)/
      DATA MFORM7/9HN(Z10,Z*,/
      DATA MFORM8/5HN(Z*,/
C
C  READ INPUT
C
C    NOEL = NUMBER OF DIFFERENT ELEMENTS
C    NOLC = NUMBER OF LOAD CASES
C    NODV = TOTALNUMBER OF DESIGN VARIABLES
C    ISNOLC = STARTING NUMBER FOR DERIVATIVE LOAD CASES
C    NDF = NUMBER OF DEGREES OF FREEDOM PER JOINT
C    NORB = TYPE OF ANALYSIS (EX. BUCKLING)
C    JOINTS = NUMBER OF JOINTS IN THE MODEL
C    NODVPE = NUMBER OF DESIGN VARIABLES PER ELEMENT
C    EL = NAMES OF ELEMENTS CONTAINING DESIGN VARIABLES
C                  (EX. E21)
C
      READ(5,60) NOLC,NODV,ISNOLC,JOINTS,NDF,NOEL,VORB
  60  FORMAT(6(1X,I4),1X,A4)
      READ(5,61) (EL(I),NSECT(I),NODVPE(I),I=1,NOEL)
  61  FORMAT(6(1X,A3,1X,I3,1X,I3))
      ISNOLC=ISNOLC-1
      NDF=NDF*NDF
C
C  DETERMINE IF THERE IS A E21 , E22 , E33 , OR E43 ELEMENT
C
      ICNTDV = 1
      DO 168 I = 1,NOEL
      JJ = NODVPE(I)
      DO 160 J = 1,JJ
      IF(EL(I).EQ.E21) GO TO 161
      IF(EL(I).EQ.E22) GO TO 163
      IF(EL(I).EQ.E43) GO TO 164
      IF(EL(I).EQ.E33) GO TO 165
      GO TO 166
```

```
C
C  CALL SUBROUTINE TO FIND DK/DV AND DM/DV FOR E21 ELEMENTS
C
 161   CALL DKDVE21(ICNTDV,NDF)
       GO TO 166
C
C  CALL SUBROUTINE TO FIND DK/DV AND DM/DV FOR E22 ELEMENTS
C
 163   CALL DKDVE22(ICNTDV,NDF)
       GO TO 166
C
C  CALL SUBROUTINE TO FIND DK/DV AND DM/DV FOR E43 ELEMENTS
C
 164   CALL DKDVE43(ICNTDV,NDF)
       GO TO 166
C
C  CALL SUBROUTINE TO FIND DK/DV FOR E33 ELEMENTS
C
 165   CALL DKDVE33(ICNTDV,NDF)
 166   ICNTDV = ICNTDV+1
 160   CONTINUE
 168   CONTINUE
       IF(VORB.EQ.VIBR) GO TO 320
       WRITE(10,172)
 172   FORMAT(*[XQT AUS*)
C
C  FIND OBJECTIVE FUNCTIONS
C
 162   WRITE(10,175) JOINTS
 175   FORMAT(* SYSVEC;UNIT VEC*/*  I=1; J=1,*,I8,*; 1.0*/
      1 * DEFINE UN=UNIT VEC*)
       DO 200 I = 1,NODV
       WRITE(10,180) I,I
  180  FORMAT(* DEFINE W*,I3,*=DMDV DIAG 0 *,I3)
  200  CONTINUE
       DO 250 I = 1,NODV
       WRITE(10,220) I,I
  220  FORMAT(* OBJF G*,I3,* 1 1=XTY(UN,W*,I3,*)*)
  250  CONTINUE
       WRITE(10,260)
 260   FORMAT(*[XQT DCU*)
       DO 300 I = 1,NODV
       WRITE(10,270) I
  270  FORMAT(* PRINT 1 OBJF G*,I3)
  300  CONTINUE
C
C  CALL SUBROUTINE TO CREATE RUNSTREAM FOR DERIVATIVE
```

```
C  CALCULATION
C
C
C  FIND APPLIED FORCES AND MOMENTS
C
      WRITE(10,2)
    2 FORMAT(*[XQT AUS*/* OUTLIB=3*)
      DO 4 I = 1,NOLC
      WRITE(10,3) I,I
    3 FORMAT(* DEFINE F*,I2,*=STAT DISP *,I2* 1*)
    4 CONTINUE
      DO 6 I = 1,NODV
      WRITE(10,5) I,NDF,I
    5 FORMAT(* DEFINE L*,I3,*=DKDV SPAR *,I2,I4)
    6 CONTINUE
      DO 10 I = 1,NOLC
      NWNOLC=I+ISNOLC
      WRITE(10,7) NWNOLC
    7 FORMAT(*  ALPHA; CASE TITLE *,I3)
      DO 9 J = 1,NODV
      WRITE(10,8) J,I,J
    8 FORMAT(1X,I3,* "LOAD CASE *,I2,* DERIVATIVE 1 DESIGN VARIABLE *,
     1 I3)
    9 CONTINUE
   10 CONTINUE
      IF(NODV.NE.1) WRITE(10,2003)
 2003 FORMAT(* OUTLIB=4*)
      IF(NODV.EQ.1) WRITE(10,6006)
 6006  FORMAT(* OUTLIB=3 *)
      DO 1010 I = 1,NOLC
      NWNOLC=I+ISNOLC
      DO 1005 J = 1,NODV
      WRITE(10,1003) NWNOLC,J,J,I
 1003 FORMAT(*  APPL FORC *,I3,I3,*= PRODUCT(-1.0 L*,I3,*,1.0 F*,I2,*)*)
 1005 CONTINUE
 1010 CONTINUE
      DO 15 I = 1,NOLC
      NWNOLC=I+ISNOLC
      IF(NODV.EQ.1) GO TO 50
      IF(I.EQ.1) WRITE(10,2009)
 2009 FORMAT(* INLIB=4*/* OUTLIB=3*)
      DO 13 J = 1,NODV,9
      ITOP=9
      ICHK=NODV-J+1
      IF(ICHK.LT.9) ITOP=ICHK
      DO 40 K = 1,ITOP
      L=K+J-1
      WRITE(10,12) K,NWNOLC,L
   12 FORMAT(* DEFINE Z*,I1,*=APPL FORC *,I3,1X,I3)
```

```
   40  CONTINUE
       IF(ITOP.EQ.1) GO TO 41
       MFORM(2)=MFORM7
       IF(J.EQ.1) MFORM(2)=MFORM8
       MFORM(3)=NUM(ITOP-1)
       WRITE(10,MFORM) (K,K=1,ITOP)
   41  IF(ITOP.EQ.1) WRITE(10,42)
   42  FORMAT(*Z11=UNION(Z10 Z1)*)
       WRITE(10,45)
   45  FORMAT(* INLIB=3*/*Z12=UNION(Z11)*/* DEFINE Z10=Z12*/
      1 * INLIB=4*)
   13  CONTINUE
       WRITE(10,46) NWNOLC
   46  FORMAT(* APPL FORC *,I3,* 1=UNION(Z10)*)
   15  CONTINUE
C
C  FIND STRESS AND DISPLACEMENT DERIVATIVES
C
   50  DO 32 I = 1,NOLC
       WRITE(10,16) NODV
   16  FORMAT(*[XQT SSOL*/* RESET L1=1,L2=*,I3)
       NWNOLC=I+ISNOLC
       WRITE(10,17) NWNOLC
   17  FORMAT(* RESET QLIB=3*/* RESET SET=*,I3)
       WRITE(10,18)
   18  FORMAT(*[XQT VPRT*/* LIB=3*)
       WRITE(10,19) NWNOLC
   19  FORMAT(* PRINT APPL  FORC *,I3)
       WRITE(10,20) NWNOLC
   20  FORMAT(* PRINT STAT DISP *,I3)
       IF(VORB.EQ.BUCK) GO TO 400
       WRITE(10,21) NODV
   21  FORMAT(*[XQT GSF*/* RESET L1=1,L2=*,I3/* RESET QLIB=3*)
       WRITE(10,22) NWNOLC
   22  FORMAT(* RESET SET=*,I3)
       WRITE(10,23) NODV
   23  FORMAT(*[XQT PSF*/* RESET L1=1,L2=*,I3/* RESET QLIB=3*)
       WRITE(10,24) NWNOLC
   24  FORMAT(* RESET SET=*,I3)
       GO TO 25
C
C  SET UP RUNSTREAM FOR BUCKLING ANALYSIS
C
  400  IDV = 0
       DO 470 J = 1,NOEL
       NODVEL = NODVPE(J)
       DO 460 K = 1,NODVEL
       IDV = IDV+1
       WRITE(10,425)
```

```
 425   FORMAT(*[XQT GSF*/* RESET EMBED=1*)
       WRITE(10,430) IDV,IDV,NWNOLC
 430   FORMAT(* RESET L1=*,I3,*,L2=*,I3,*,SET=*,I3,*,QLIB=3*)
       WRITE(10,435)
 435   FORMAT(*[XQT PSF*)
       WRITE(10,430) IDV,IDV,NWNOLC
       WRITE(10,440)
 440   FORMAT(*[XQT KG*)
       WRITE(10,445)
 445   FORMAT(*[XQT DCU*)
       WRITE(10,450) NDF,NDF,IDV
 450   FORMAT(* CHANGE 1,KG SPAR *,I3,* 0,DKG SPAR *,I3,I4)
       WRITE(10,455) NDF,IDV
 455   FORMAT(* COPY 1,3 DKG SPAR *,I3,I4)
 460   CONTINUE
 470   CONTINUE
  25   WRITE(10,251)
 251   FORMAT(*[XQT DCU*)
       WRITE(10,26) NWNOLC,NWNOLC
  26   FORMAT(* CHANGE 3,STAT DISP *,I3,* 1,DDIS DISP *,I3,* 1*)
C
C   CHANGE DATA SET NAMES
C
       IDV=0
       DO 31 J=1,NOEL
       NOSECT = NSECT(J)
       NODVEL = NODVPE(J)
       DO 30 K=1,NODVEL
       DO 271 LL = 1,NOSECT
       IDV=IDV+1
       DO 27 KK = 1,NOEL
       IF(EL(KK).EQ.E21) GO TO 29
       IF(EL(KK).EQ.E22) GO TO 29
       IF(EL(KK).EQ.E43) GO TO 29
       IF(EL(KK).EQ.E33) GO TO 29
       WRITE(10,28) EL(KK),NWNOLC,IDV,EL(KK),NWNOLC,IDV
  28   FORMAT(* CHANGE 3,STRS *,A3,2I4,*,DSTR *,A3,2I4)
       WRITE(10,285) EL(KK),NWNOLC,IDV
 285   FORMAT(* COPY 3,4 DSTR *,A3,2I4)
       GO TO 27
C
C   STORE BEAM CROSS DERIVATIVES
C
  29   WRITE(10,290) EL(KK),NWNOLC,IDV,EL(KK),NWNOLC,IDV
  290  FORMAT(* CHANGE 3,STRS *,A3,2I4,*,DFAM *,A3,2I4)
       WRITE(10,292) EL(KK),NWNOLC,IDV
 292   FORMAT(* COPY 3,4 DFAM *,A3,2I4)
  27   CONTINUE
 271   CONTINUE
```

```
 30   CONTINUE
 31   CONTINUE
 32   CONTINUE
320   IF(VORB.EQ.VIBR) WRITE(10,260)
      DO 333 I = 1,NODV
      WRITE(10,332) I
332   FORMAT(* COPY 1,4 OBJF G*,I3)
333   CONTINUE
      WRITE(10,33)
 33   FORMAT(* TOC 3*/* TOC 1*/* TOC 4*/*[XQT EXIT*)
      STOP
      END
```

SUBROUTINE DKDVE21

```
      SUBROUTINE DKDVE21(NDVJIM,NDF)
      DIMENSION X(20)
      NAMELIST/LINKF/NDV,X
C
C   THIS SUBROUTINE CREATES A SPAR RUNSTREAM TO CALCULATE
C
C           DK     DK    DA     DK     DI     DK     DI     DK     DJ
C                                        X             Y             0
C          --- = -- . --- + --- . --- + --- . --- + --- . ---
C           DV     DA    DV     DI     DV     DI     DV     DJ     DV
C             I             I      X      I      Y      I      0      I
C
C
C
C           DM     DM    DA     DM     DI     DM     DI     DM     DJ
C                                        X             Y             0
C          --- = -- . --- + --- . --- + --- . --- + --- . ---
C           DV     DA    DV     DI     DV     DI     DV     DJ     DV
C             I             I      X      I      Y      I      0      I
C
      WRITE(10,1)
    1 FORMAT(*[XQT AUS*)
      WRITE(10,3)NDF,NDVJIM,NDF,NDVJIM
    3 FORMAT(* DEFINE A1=DKDA SPAR *,I2,I4/* DEFINE A2=DKIX*
     1 * SPAR *,I2,I4)
      WRITE(10,4)NDF,NDVJIM,NDF,NDVJIM
    4 FORMAT(* DEFINE A3=DKIY SPAR *,I2,I4/* DEFINE A4=DKJO*
     1 * SPAR *,I2,I4)
      WRITE(10,203)NDVJIM,NDVJIM
  203 FORMAT(* DEFINE B1=DMDA DIAG 0 *,I3/* DEFINE B2=DMIX*
     1 * DIAG 0 *,I3)
      WRITE(10,204)NDVJIM,NDVJIM
  204 FORMAT(* DEFINE B3=DMIY DIAG 0 *,I3/* DEFINE B4=DMJO*
     1 * DIAG 0 *,I3)
C
C  COMPUTE  DA/DV
C
      DADV=1.
C
C   READ IN AND SET UP INITIALIZATION VALUES
C
C   READ IN CONSTANTS FROM UNIT 30
C
      READ(30,5) B10,B20,TO
    5 FORMAT(3F10.3)
C
C   READ IN DESIGN VARIABLES FROM UNIT 31
C
      READ(31,LINKF)
```

```
      AREAO=(2.*B10+B20)*TO
      AREA=AREAO/X(2)
      SCALE=SQRT(AREA/AREAO)
      B1=B10*SCALE
      B2=B20*SCALE
      T=TO*SCALE
      FTR=0.5/SQRT(AREA*AREAO)
      C=(B2+2.*T)*B1**2/2.-B2*(B1-T)*(B1+T)/2.
      C=C/AREA
C
C  COMPUTE FACTORS FOR DI1/DV , DI2/DV , DJO/DV
C
      DB1DV=B10*FTR
      DB2DV=B20*FTR
      DTDV=TO*FTR
      DI1DB1=(B2+2.*T)**3/12.-B2**3/12.
      DI1DB2=(B2+2.*T)**2*B1/4.-B2**2*(B1-T)/4.
      DI1DT=(B2+2.*T)**2*B1/2.+B2**3/12.
      DCDB1=(B1*(B2+2.*T)-B2*B1-2.*C*T)/AREA
      DCDB2=(B1**2/2.-(B1-T)*(B1+T)/2.-C*T)/AREA
      DCDT=(B1**2+B2*T-C*(2.*B1+B2))/AREA
      DI2DB1=(T*B1**2/2.)+(2.*T*(B1/2.-C)**2)+
     1   (4.*T*B1*(B1/2.-C)*(.5-DCDB1))+
     2  (2.*B2*T*(C-T/2.)*DCDB1)
      DI2DB2=(-4.*T*B1*(B1/2.-C)*DCDB2)+(T**3/12.)+
     1  (T*(C-T/2.)**2)+(2.*B2*T*(C-T/2.)*DCDB2)
      DI2DT=(B1**3/6.)+(2.*B1*(B1/2.-C)**2)-
     1  (4.*T*B1*(B1/2.-C)*DCDT)+(B2*T**2/4.)+
     2  (B2*(C-T/2.)**2)+(2.*B2*T*(C-T/2.)*(DCDT-.5))
      DJODB1=2.*T**3/3.
      DJODB2=T**3/3.
      DJODT=(2.*B1+B2)*T**2
C
C  COMPUTE DI1/DV , DI2/DV , DJO/DV
C
      DI1DV=DI1DB1*DB1DV+DI1DB2*DB2DV+DI1DT*DTDV
      DI2DV=DI2DB1*DB1DV+DI2DB2*DB2DV+DI2DT*DTDV
      DJODV=DJODB1*DB1DV+DJODB2*DB2DV+DJODT*DTDV
C
C  CREATE RUNSTREAM TO FIND DK/DV
C
      WRITE(10,106)DADV,DI1DV
      WRITE(10,107)DI2DV,DJODV
  106 FORMAT(8X,*S1=SUM(*,E13.5,1X,*A1,*E13.5,1X,*A2)*)
  107 FORMAT(8X,*S2=SUM(*,E13.5,1X,*A3,*E13.5,1X,*A4)*)
      WRITE(10,108) NDF,NDVJIM
 108  FORMAT(* DKDV SPAR *,I2,I4,*=SUM(S1,S2)*)
C
C  CREATE RUNSTREAM TO FIND DM/DV
```

Subroutine DKDVE21 (Conc.)

```
C
      WRITE(10,206)DADV,DI1DV
      WRITE(10,207)DI2DV,DJODV
  206 FORMAT(8X,*T1=SUM(*,E13.5,1X,*B1,*E13.5,1X,*B2)*)
  207 FORMAT(8X,*T2=SUM(*,E13.5,1X,*B3,*E13.5,1X,*B4)*)
      WRITE(10,208) NDVJIM
 208  FORMAT(* DMDV DIAG 0*,I3,*=SUM(T1,T2)*)
      WRITE(10,209)
 209  FORMAT(*[XQT DCU*/* TOC 1*)
      RETURN
      END
```

APPENDIX D

DRVSTRS

```
      PROGRAM DRVSTRS(INPUT=65,TAPE30,TAPE31,TAPE5=INPUT
     1 ,TAPE15,TAPE16,OUTPUT=65,TAPE6=OUTPUT)
C
C    THIS PROGRAM COMPUTES THE STRESSES AND STRESS DERIVATIVES
C    FOR SPAR BEAM ELEMENTS USING SPAR LIBRARIES AS INPUT.
C
C    THE STRESSES ARE COMPUTED FROM SPARLA USING DATA SET
C                FAMS MASK  I  1 WHERE I = 1 TO NOLC
C    THE STRESSES ARE WRITTEN BACK ONTO SPARLA USING DATA SET
C                STRS MASK  I  1 WHERE I = 1 TO NOLC
C
C    THE STRESS DERIVATIVES ARE COMPUTED FROM SPARLC USING DATA SET
C         DFAM MASK  I  J WHERE I = ISTRTLC TO ISTRTLC+NOLC AND J = 1 TO NODV
C    THE STRESS DERIVATIVES ARE WRITTEN BACK ONTO SPARLC USING DATA SET
C         DSTR MASK  I  J WHERE I = ISTRTLC TO ISTRTLC+NOLC AND J = 1 TO NODV
C
      COMMON KORE,KEVEN,Z(1)
      DIMENSION NAME1(2),NAME2(2),EL(999),NODVPE(999)
      DATA E21,E22,E33,E43/3HE21,3HE22,3HE33,3HE43/
      DATA NAME1/4HFAMS,4HDFAM/
      DATA NAME2/4HSTRS,4HDSTR/
C
C    READ INPUT VALUES
C
C    NOEL IS THE NUMBER OF DIFFERENT ELEMENTS
C    NDF IS THE DEGREES OF FREEDOM PER JOINT
C    VORB IS THE TYPE OF ANALYSIS (EX. BUCKLING)
C    JOINTS IS THE NUMBER OF JOINTS IN THE MODEL
C    NOLC IS THE NUMBER OF LOAD CASES
C    NODV IS THE NUMBER OF DESIGN VRIABLES
C    ISTRTLC IS THE STARTING NUMBER FOR THE DERIVATIVE LOAD CASES
C    EL =NAMES OF ELEMENTS USED AS DESIGN VARIABLES
C              (EX. E21)
C
      READ(5,1) NOLC,NODV,ISNOLC,JOINTS,NDF,NOEL,VORB
    1 FORMAT(6(1X,I4),1X,A4)
      READ(5,2) (EL(I),NSECT,NODVPE(I),I=1,NOEL)
    2 FORMAT(6(1X,A3,1X,I3,1X,I3))
      ISTRTLC=ISNOLC-1
      CALL RSET(IL,0,0)
C
C   LOOP ON THE NUMBER OF LOAD CASES
C
      DO 4 NCNTLC = 1,NOLC
C
C   LOOP ON NUMBER OF ELEMENTS
C
      DO 10 ISTRS = 1,NOEL
C
```

DRVSTRS (Cont.)

```
C   CHECK FOR E21,E22,E33,E43 ELEMENT
C
        IF(EL(ISTRS).EQ.E21.OR.EL(ISTRS).EQ.E43.OR.
      1 EL(ISTRS).EQ.E22.OR.EL(ISTRS).EQ.E33) GO TO 9
        GO TO 10
    9   NODVEL = NODVPE(ISTRS)
C
C   SET NUMBER OF STRESS OR STRESS DERIVATIVES TO BE WRITTEN
C
        IF(EL(ISTRS).EQ.E21.OR.EL(ISTRS).EQ.E22) IWRDCNT = 8
        IF(EL(ISTRS).EQ.E43.OR.EL(ISTRS).EQ.E33) IWRDCNT = 4
C
C   LOOP ON NUMBER OF DESIGN VARIABLES PER ELEMENT
C
        DO 90 JK = 1,NODVEL
        REWIND 15
        REWIND 16
C
C   READ FAMS MASK FROM SPARLA AND COMPUTE STRS MASK
C
        CALL RDATSET(4,NAME1(1),NCNTLC,1,NE1,1,0,EL(ISTRS))
        REWIND 15
        REWIND 16
C
C   CALL SUBROUTINE TO WRITE STRESSES ON SPAR LIBRARY
C
C
C   WRITE STRS MASK ON SPARLA
C
        CALL WRTDATA(4,NAME2(1),NCNTLC,1,NE1,EL(ISTRS),IWRDCNT)
   90   CONTINUE
   10   CONTINUE
        ICNTDV = 1
C
C   LOOP ON NUMBER OF ELEMENTS
C
        DO 3 ISTRS=1,NOEL
        IF(EL(ISTRS).EQ.E21.OR.EL(ISTRS).EQ.E43.OR.
      1 EL(ISTRS).EQ.E22.OR.EL(ISTRS).EQ.E33) GO TO 19
        GO TO 3
   19   DO 30 ICNTDV = 1,NODV
        IBEAM = 0
        NODVEL = NODVPE(ICNTDV)
C
C   SET SWITCH FOR BEAM ELEMENT
C
        IF(EL(ICNTDV).EQ.E21.OR.EL(ICNTDV).EQ.E22) IBEAM=1
        N3=NCNTLC+ISTRTLC
C
```

```
C  SET NUMBER OF STRESSES AND STRESS DERIVATIVES TO BE WRITTEN
C
      IF(EL(ICNTDV).EQ.E21) IWRDCNT = 8
      IF(EL(ICNTDV).EQ.E43) IWRDCNT = 4
C
C  LOOP ON NUMBER OF DESIGN VARIABLES PER ELEMENT
C
      DO 20 JK = 1,NODVEL
C
C  READ DFAM MASK FROM SPARLC AND COMPUTE DSTR MASK
C
      REWIND 15
      CALL RDATSET(4,NAME1(2),N3,ICNTDV,NE1,0,IBEAM,EL(ISTRS))
      REWIND 15
C
C  WRITE DSTR MASK ON SPARLC
C
      CALL WRTDATA(4,NAME2(2),N3,ICNTDV,NE1,EL(ISTRS),IWRDCNT)
   20 CONTINUE
   30 CONTINUE
    3 CONTINUE
    4 CONTINUE
      CALL FIN(0,0)
      STOP
      END
      SUBROUTINE WRTDATA(NU,N1,N3,ISTRS,NE1,X2,IWRDCNT)
C
C   SUBROUTINE TO WRITE STRESS AND STRESS DERIVATIVE
C   DATA SETS BACK INTO SPAR LIBRARIES
C
      COMMON KORE,KEVEN,Z(1)
C
C   SET UP BLOCK SIZES
C   IF NWD3 GT OPEN CORE SIZE , USE MORE THAN ONE BLOCK
C
      NWD3=NE1*IWRDCNT
      LB3=NWD3
      IF(NWD3.LT.KORE) GO TO 61
      LB3=NWD3/2
   61 ISW=0
      IF(NWD3.LT.KORE) ISW=1
      CALL DAL(NU,0,Z(1),KORE,1,KADR3,IERR,NWD3,NE1,LB3,-1,
     1 N1,X2,N3,ISTRS)
   62 I1=1
   63 I2=I1+IWRDCNT-1
      IF(I2.GT.LB3) GO TO 66
C
C   READ STRESSES OR STRESS DERIVATIVES OFF UNIT 15
C
```

```
      READ(15) (Z(I),I=I1,I2)
      IF(EOF(15)) 66,65
   65 I1=I1+IWRDCNT
      GO TO 63
C
C    WRITE STRESSES OR STRESS DERIVATIVES ONTO SPAR LIBRARY
C
   66 CALL RIO(NU,10,2,KADR3,Z(1),LB3)
      IF(ISW.EQ.1) GO TO 67
      ISW=1
      LB3=NWD3-LB3
      GO TO 62
   67 RETURN
      END
      SUBROUTINE RDATSET(NU,N1,N3,ISTRS,NE1,ISW,IBEAM,X2)
C
C    SUBROUTINE TO READ SPAR LIBRARY,STORE DATA, AND
C    COMPUTE STRESSES OR STRESS DERIVATIVES
C
      COMMON KORE,KEVEN,Z(1)
      DATA E21,E22,E33,E43/3HE21,3HE22,3HE33,3HE43/
C
C    SET UP BLOCK SIZE
C
      CALL DAL(NU,10,Z(1),KORE,1,KADR1,IERR,NWD1,NE1,LB1,ITYPE,N1,
     1 X2,N3,ISTRS)
      NI1=LB1/NE1
      KCNT=0
      NBLK1=NWD1/LB1
      IF(NBLK1*LB1.NE.NWD1) NBLK1=NBLK1+1
      DO 6 J = 1,NBLK1
      NLB1=LB1
      IF(J.EQ.NBLK1) NLB1=NWD1-(NBLK1-1)*LB1
C
C    READ DATA FROM SPAR LIBRARY
C
      CALL RIO(NU,20,2,KADR1,Z(1),NLB1)
      DO 30 JCNT=1,NLB1,NI1
      KCNT=KCNT+1
C
C     CALL SUBROUTINE TO COMPUTE STRESSES AND STRESS DERIVATIVES
C
      IF(X2.EQ.E21.OR.X2.EQ.E22) CALL BMSTRS(ISW,KCNT,JCNT,IBEAM)
      IF(X2.EQ.E43.OR.X2.EQ.E33) CALL PLTSTRS(ISW,KCNT,JCNT)
   30 CONTINUE
    6 CONTINUE
      RETURN
      END
```

APPENDIX D

SUBROUTINE BMSTRS

```
      SUBROUTINE BMSTRS(ISW,KCNT,JCNT,IBEAM)
      COMMON KORE,KEVEN,Z(1)
      DIMENSION DY1DV(4),DY2DV(4),Y(4,2),S(8),F3(2),XM1(2),XM2(2)
      DIMENSION X(20)
      NAMELIST/LINKF/NDV,X
      JM1 = JCNT-1
C
C    STORE FORCES AND MOMENTS IF ISTRS EQ 1
C    STORE DERIVATIVES OF FORCES AND MOMENTS IF ISTRS EQ 2
C
      F3(1)=-Z(JM1+13)
      XM1(1)=-Z(JM1+14)
      XM2(1)=-Z(JM1+15)
      F3(2)=Z(JM1+19)
      XM1(2)=Z(JM1+20)
      XM2(2)=Z(JM1+21)
      IF(KCNT.NE.1) GO TO 31
      IF(ISW.NE.1) GO TO 31
C
C    STORE AREA, MOMENTS OF INERTIA, AND Y VALUES IF
C    FIRST TIME THROUGH
C
      XI1=Z(JM1+25)
      XI2=Z(JM1+27)
      ICNT=38
      DO 3 K = 1,4
      DO 2 L = 1,2
      ICNT=ICNT+1
      Y(K,L)=Z(JM1+ICNT)
    2 CONTINUE
    3 CONTINUE
C
C    FIND DERIVATIVE OF Y VALUES IF NEEDED
C
C
C   READ IN CONSTANT NUMBERS FROM UNIT 30
C
      READ(30,10) B10,B20,T0
```

```
   10   FORMAT(3F10.3)
C
C   READ IN DESIGN VARIABLES FROM UNIT 31
C
        READ(31,LINKF)
C
C   CALCULATE DY/DV
C
        A0=(2.*B10+B20)*T0
        A=A0/X(2)
        SCALE=SQRT(A/A0)
        B1=B10*SCALE
        B2=B20*SCALE
        T=T0*SCALE
        FTR=0.5/SQRT(A*A0)
        C=(B2+2.*T)*B1**2/2.-B2*(B1-T)*(B1+T)/2.
        C=C/A
        DB1DV=B10*FTR
        DB2DV=B20*FTR
        DTDV=T0*FTR
        DCDB1=(B1*(B2+2.*T)-B2*B1-2.*C*T)/A
        DCDB2=(B1**2/2.-(B1-T)*(B1+T)/2.-C*T)/A
        DCDT=(B1**2+B2*T-C*(2.*B1+B2))/A
        DCDV=(DCDB1*DB1DV)+(DCDB2*DB2DV)+(DCDT*DTDV)
        DY1DV(1)=-DB1DV+DCDV
        DY1DV(2)=DCDV
        DY1DV(3)=DCDV
        DY1DV(4)=DY1DV(1)
        DY2DV(1)=(.5*DB2DV)+DTDV
        DY2DV(2)=DY2DV(1)
        DY2DV(3)=-DY2DV(1)
        DY2DV(4)=DY2DV(3)
   31   ICNT=0
        DO 200 II = 1,2
        DO 100 I = 1,4
        ICNT=ICNT+1
C
C    COMPUTE STRESSES OR STRESS DERIVATIVES
C
        S(ICNT)=(F3(II)/A)+((XM1(II)/XI1)*Y(I,2))-((XM2(II)/XI2)
      1 *Y(I,1))
C
C    IF AREA OF BEAM IS A CONTRIBUTING FACTOR TO STRESS
C    DERIVATIVES AND ISTRS EQ 1 THEN CALCULATE THE FACTOR
C    AND STORE ON UNIT 16
C
        IF(ISW.NE.1) GO TO 95
        DFAC=((XM1(II)/XI1)*DY2DV(I))-((XM2(II)/XI2)*DY1DV(I))
        WRITE(16) DFAC
```

SUBROUTINE BMSTRS (Conc.)

```
      GO TO 100
C
C    IF AREA OF BEAM IS A CONTRIBUTING FACTOR TO STRESS
C    DERIVATIVES AND ISTRS EQ 2 THEN READ FACTOR OFF UNIT 16
C    AND ADD IT TO STRESS DERIVATIVE
C
   95 IF(IBEAM.EQ.0) GO TO 100
      READ(16) DFAC
      S(ICNT)=S(ICNT)+DFAC
  100 CONTINUE
  200 CONTINUE
C
C  WRITE STRESSES ON UNIT 15
C
      WRITE(15) S
      RETURN
      END
```

APPENDIX E

DATA FILES

The following are sample listings of data files.

INPUT DATA FILES

PCONPAR,CONPAR

```
$CONPAR
IPRINT=5,
NDV=3,
ITMAX=20,
NCON=190,
NFDG=0,
NSIDE=1,
ICNDIR=0,
NSCAL=4,
LINOBJ=0,
ITRM=0,
FDCH=0.,
FDCHM=0.,
CT=0.,
CTMIN=0.025,
CTL=0.,
CTLMIN=0.,
THETA=0.125,
PHI=5.,
DELFUN=0.05,
DABFUN=0.0000000001,
ISC(1)=190*0.0,
N1=20,
N2=400,
N3=200,
N4=200,
N5=400,
ALPHAX=0.0,
ABOBJ1=0.0,
IGOTO=0,
VLB(1)=0.005,
VLB(2)=0.1,
VLB(3)=1.0,
VUB(1)=1.0,
VUB(2)=10.0,
VUB(3)=10.0,
$END
```

(See ref. 6 for a description of the parameters.)

APPENDIX E

PSTART,STARTX

```
$STARTX
X(1)=0.05,
X(2)=1.0,
X(3)=4.0,
$END
```

Starting values for three design variables.

INPT

```
   1     3  100    80     5     3 STAT
E23    1    1 E21    1    1 E41    1    1
```

Card type 1 Number of load cases = 1, Number of design variables = 3,
Starting number for load cases = 100, Number of joints = 80,
Number of nonzero degrees of freedom = 5,
Number of different element types = 3,
Type of analysis = STAT

Card type 2 Element names E23, E21, E41
Last section number for each element 1,1,1
Number of design variables per element 1,1,1

APPENDIX E

CNT

```
$CNT
OBJ1     = 10.0,
OBJ2     = 50.0,
OBJ3     = 10.0,
TOL      = .5E-01,
$END
```

Initial objective functions and tolerance

ENDN

```
$EPIN
E23AL=2000.,
E21AL=2000.,
E41AL=2000.,
NSE23=58,
NSE21=76,
NSE41=56,
$END
```

E23AL, E21AL, E41AL are limits on the design variables
NSE23, NSE21, NSE41 are the number of each type of element

CONS

```
12.          30.          2.
```

Input values to calculate the derivative of the stiffness and mass matrices with respect to the design variable

B1 = 12
B2 = 30
T1 = 2

Cross-sectional dimensions of a beam (See ref. 2 for more detail.)

APPENDIX E

MODEL DATA FILES

NRRS (nonrepeatable SPAR runstream)

```
[XQT TAB
START 80, 5$ ROTATIONS ABOUT  Y EXCLUDED
TITLE" FUSELAGE MODEL,FSPAR1
TEXT
" MEMBRANE-ROD-BEAM FUSELAGE MODEL
"NONREPEATABLE PART
JLOC$ FUSELAGE DIA. 800. CM.,LENGTH=800. CM.
FORMAT=2$CYLINDRICAL COORDINATES
1 400. 0. 0. 400. 337.5 0. 16 1 5
16 400. 0. 800. 400. 337.5 800.
MREF
FORMAT=2
1 -2 0. 0. 10000000.
2 1 0. 0. 10000000.
MATC
1 .72+6 0.3 .0028 22.-6$ AL-ALLOY,METRIC UNITS
E23 SECTION PROPERTIES $ROD ELEMENTS
1 4.168$AREA OF THE RODS
E21 SECTION PROPERTIES$BEAM ELEMENTS
DSY 1 16804. 0. 1262.7 0. 108. 144. 0. 6.0784 0. 0.
0. 0. 0. -8.7778 17. 3.2222 17. 3.2222 -17. -8.7778 -17.
SHELL SECTION PROPERTIES
1 0.1$SKIN THICKNESS
CONSTRAINT CASE 1
ZERO 1,2,3;1,16$ CANTILEVER THE FUSELAGE
ZERO 1,2,3,4,6;36,38
```

Input to TAB processor to tabulate material properties, structural geometry, constraint conditions, etc.

```
[XQT ELD
$START
E23$ROD ELEMENTS
NSECT=1$
NREF=2
1 17 1 4 3 1 $
4 20$
5 21$
6 22$
52 68$
53 69$
54 70$
7 23 1 4 10 1$
$END
$START
E21
NSECT=1$
NREF=1
1 2 2 16 2 16 $
49 50 2 16 2 16$
33 34$
34 35$
39 40$
40 41$
41 42$
42 43$
43 44$
44 45$
45 46$
46 47$
47 48$
48 33$
$END
$START
E41$ MEMBRANE PANELS
NSECT=1$
1 17 18 2 2 16 1 $
49 65 66 50 2 16 1$
17 33 34 18 1 1 2 2 16$
23 39 40 24 1 1 9 2 16$
32 48 33 17$
48 64 49 33$
$END
[XQT DCU
 TOC 1
[XQT EXIT
```

Input to ELD processor to establish element connectivity. (See ref. 5 for more detail.)

APPENDIX E

NGRS (repeatable SPAR runstream)

```
[XQT TAB
 TITLE"FUSELAGE MODEL
TEXT
"MEMBRANE ROD BEAM FUSELAGE MODEL
"REPEATABLE PART WITHOUT GRADIENTS          Update for new design variables
 UPDATE=1
$ MERGE NEW PROPERTIES HERE.
UPDATE=0
[XQT TOPO
[XQT E          Generation of element matrices,
[XQT EKS        assembling stiffness and mass
[XQT K          matrices, decomposing the
[XQT INV        stiffness matrix
[XQT M
 RESET G=981.
[XQT AUS
 SYSVEC;APPLIED FORCES 1
  I=1;J=65;69;77; 10000. -20000. 20000.     Applied loading definition
 SYSVEC;UNIT VEC
  I=1; J=1,80; 1.0
 DEFINE WT=DEM DIAG 0 0
 DEFINE UN=UNIT VEC
 OBJFUN=XTY(UN,WT)
[XQT DCU
 PRINT 1 OBJFUN
[XQT SSOL
[XQT GSF       Static deflection and stress computations
[XQT PSF
[XQT VPRT
 PRINT APPL FORC 1 1
 PRINT STAT DISP 1 1
[XQT DCU
 PRINT 1 STAT DISP 1 1     Print results
 PRINT 1 STRS E21 1 1
 PRINT 1 STRS E23 1 1
 PRINT 1 STRS E41 1 1
 TOC 1
COPY 1,4 OBJF AUS 1 1
COPY 1,4 STRS E21 1 1      Copy data needed for end processor
COPY 1,4 STRS E23 1 1      to library 4 (SPARLD)
COPY 1,4 STRS E41 1 1
TOC 4
[XQT EXIT
```

(See ref. 5 for more details.)

APPENDIX E

TRANSFER DATA FILES

PCONRST

```
1$BLK
OOBJ    = .66844321021006E+04,
OOBJI   = .66844321021006E+04,
OX      = .72164310322613E-01, .6591825639314E+00, .61746618918153E+01, 0.0,     0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
          0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
OXI     = .72164310322613E-01, .6591825639314E+00, .61746618918153E+01, 0.0,     0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
          0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
OG      = -.99571653182067E+00, -.95278668795628E+00, -.92977417601422E+00,      -.96473035096325E+00, -.83931530923177E+00,
          -.88108655172185E+00, -.9560175523689E+00, -.94280239683172E+00,       -.99946668264813E+00, -.96434060366449E+00,
          -.74748216542431E+00, -.90032053868088E+00, -.97225336835133E+00,      -.99427773862803E+00, -.92572788669819E+00,
          -.99283934965684E+00, -.99897134994753E+00, -.99261590906344E+00,      -.83214485071733E+00, -.75653906017337E+00,
          -.89833768332524E+00, -.92285408009627E+00, -.78828474089686E+00,      -.82111839595694E+00, -.96474726678335E+00,
          -.93670072855566E+00, -.79381674253358E+00, -.8808405468923E+00,       -.94761004466038E+00, -.99671110630028E+00,
          -.99425453634863E+00, -.99464221727485E+00, -.97411524169175E+00,      -.97457918813098E+00, -.85051457820339E+00,
          -.91452472423283E+00, -.96152890934078E+00, -.99689707221875E+00,      -.85342643886951E+00, -.91199657662224E+00,
          -.9519194948728E+00, -.99796680302194E+00, -.91180327277382E+00,       -.94280095477239E+00, -.97248129706004E+00,
          -.9922875307293E+00, -.97881077138869E+00, -.99838170122538E+00,       -.99611579076635E+00, -.98543747240342E+00,
          -.84046126583635E+00, -.92015808710181E+00, -.98060156268971E+00,      -.9974137775222E+00, -.83542566600255E+00,
          -.88186594296228E+00, -.95838261295228E+00, -.99940133814644E+00,      -.1E+01, -.1E+01, -.1E+01, -.1E+01,
          -.1E+01, -.1E+01, -.1E+01, -.1E+01, -.1E+01, -.1E+01, -.1E+01,         -.1E+01, -.1E+01, -.1E+01, -.1E+01,
          -.1E+01, -.73968974665649E+00, -.71989799093343E+00,                   -.8511666297607E+00, -.90946623369869E+00,
          -.92579316511445E+00, -.62138791415929E+00, -.65106179155173E+00,      -.66952913185199E+00, -.86838998001038E+00,
          -.92079324083711E+00, -.90775305520754E+00, -.89595924591247E+00,      -.91185785808518E+00, -.90403911307041E+00,
          -.88673784984652E+00, -.89475684350496E+00, -.45413763638525E+00,      -.13258722013038E+00, -.11265722125307E+00,
          -.24574418109944E+00, -.37252680700796E+00, -.1277757083191E+00,       -.14069267080698E+00, -.67131421173064E+00,
          -.66507304183887E+00, -.78599491280269E+00, -.79962619034101E+00,      -.78648540290769E+00, -.84843894794398E+00,
          -.9012625127168E+00, -.68563837392492E+00, -.44630198363183E+00,       .51640634559647E-04, -.32892600834779E+00,
          -.24943260855685E+00, -.23938086529206E+00, -.38544758031867E+00,      -.39376147400844E+00, -.46157025269697E+00,
          -.68328205980549E+00, -.72550634975451E+00, -.71499687253023E+00,      -.69415476505674E+00, -.72640000804193E+00,
          -.81968862721173E+00, -.85820315564034E+00, -.79979623889693E+00,      -.15156346082392E-02, -.80100634622976E+00,
          -.83017223254758E+00, -.74685239264797E+00, -.78238030234334E+00,      -.87646444404299E+00, -.87021102392971E+00,
          -.85536227795987E+00, -.84134608903399E+00, -.8632017810377E+00,       -.870002821107E+00, -.76978718295792E+00,
          -.77922095780907E+00, -.98521723872972E+00, -.98012451403279E+00,      -.98413107002976E+00, -.99541179420632E+00,
          -.99446876229629E+00, -.96210799401357E+00, -.97034013339282E+00,      -.9619855144805E+00, -.98125718230776E+00,
          -.98156422450367E+00, -.97455981195683E+00, -.95360592271773E+00,      -.94710682135374E+00, -.96428256056183E+00,
          -.97383641929083E+00, -.98546675701785E+00, -.97990507070664E+00,      -.97987107140991E+00, -.97916498325269E+00,
          -.97568572267122E+00, -.97597122290727E+00, -.9745631138003E+00,       -.99173926211893E+00, -.98866056518597E+00,
          -.99577916131596E+00, -.99241746260864E+00, -.99323757570048E+00,      -.95338672837885E+00, -.94911390878708E+00,
          -.978056043562E+00, -.97767846314869E+00, -.98205964619667E+00,        -.98533338255271E+00, -.94898290973881E+00,
          -.97195026947835E+00, -.94051829610329E+00, -.91611296668651E+00,      -.97570729368604E+00, -.98070556144624E+00,
          -.98869935028841E+00, -.98263149395566E+00, -.95592082158014E+00,      -.94695508505965E+00, -.96583882604775E+00,
          -.98117265101567E+00, -.91266273291119E+00, -.98976405401755E+00,      -.98060451499322E+00, -.99319984910193E+00,
          -.98755141653778E+00, -.95667596581468E+00, -.94820328914997E+00,      -.96731843183851E+00, -.98253508945565E+00,
          -.98235876947143E+00, -.98601945721173E+00, -.999E+00,                 -.99535212980241E+00, -.99565864518007E+00,
          -.9995E+00, -.9995E+00, -.99591798794941E+00, -.99957561990303E+00,    -.9985E+00, -.9995E+00, -.97110348372347E+00,
          -.97110348372347E+00, -.998E+00, -.99187821009216E+00,                 -.99207662481976E+00, -.99197741745596E+00,
          -.9895E+00, -.99454001647515E+00, -.992676288105E+00,                  -.99360815229007E+00, -.98849999999999E+00,
          -.92095023976184E+00, -.92095023976184E+00, -.92095023976184E+00,      -.996E+00, -.97280786237144E+00, -.98117781068303E+00,
          -.9995E+00, -.9995E+00, -.97841413909813E+00, -.98645886699162E+00,    -.9955E+00, -.9995E+00, -.99837878197616E+00,
          -.99837878197616E+00, -.995E+00, -.98014396447397E+00,                 -.96669896713244E+00, -.99327750132923E+00,
          -.9865E+00, -.98815523639225E+00, -.98261657472901E+00,                -.98538590556063E+00, -.98549999999999E+00,
          -.87083655602638E+00, -.87083655602638E+00, -.87083655602638E+00,      -.99299999999999E+00, -.95958280823957E+00,
          -.98772025745439E+00, -.9995E+00, -.9995E+00, -.99052830345821E+00,    -.99427610641199E+00, -.9925E+00, -.9995E+00,
          -.96460249550426E+00, -.96460249550426E+00, -.992E+00, -.991E+00,      -.96911460483604E+00, -.96911460483604E+00,
          -.9905E+00, -.95444794958863E+00, -.95444794958863E+00,                -.98999999999999E+00, -.97110348372347E+00,
          -.97110348372347E+00, -.9895E+00, -.99187821009216E+00, -.9895E+00,    -.9779066206843E+00, -.973066204476E+00,
          -.9779066206843E+00, -.92095023976184E+00, -.92095023976184E+00,       -.92095023976184E+00, -.90929427774941E+00,
          -.90929427774941E+00, -.90929427774941E+00, -.93422268454974E+00,      -.93422268454974E+00, -.98699999999999E+00,
          -.93489258324666E+00, -.95465946963056E+00, -.98014396447397E+00,      -.96669896713244E+00, -.96669896713244E+00,
          -.96343319687055E+00, -.95760171241116E+00, -.96051745464085E+00,      -.87083655602638E+00, -.87083655602638E+00,
          -.87083655602638E+00, -.84645516794456E+00, -.84645516794456E+00,      -.84645516794456E+00, -.90391088590332E+00,
          -.90391088590332E+00, -.90391088590332E+00, -.95160618946586E+00,      -.95455588846128E+00, -.96460249550426E+00,
          -.95997959731942E+00, -.97635013804871E+00, -.967E+00,                 -.95254819716267E+00, -.95254819716267E+00,
          -.9665E+00, -.95026798790877E+00, -.95026798790877E+00,                -.96599999999999E+00, -.96364007981332E+00,
          -.96364007981332E+00, -.90888641161881E+00, -.90888641161881E+00,      -.90888641161881E+00, -.90781353009334E+00,
          -.90781353009334E+00, -.90781353009334E+00, -.93221102031217E+00,      -.93221102031217E+00, -.93221102031217E+00,
          -.95876147078707E+00, -.95876147078707E+00, -.95876147078707E+00,      -.97507176435795E+00, -.97507176435795E+00,
          -.96299999999999E+00, -.97533419127548E+00, -.9903981902304E+00,       -.9625E+00, -.98356580237213E+00, -.98356580237213E+00,
          -.962E+00, -.94627722499904E+00, -.92107688231724E+00,                 -.94949947037598E+00, -.99251617397503E+00,
          -.95852573485126E+00, -.86072846668075E+00, -.84284953818756E+00,      -.89292910280282E+00, -.95995527565108E+00,
          -.96807400959003E+00, -.92550907309448E+00, -.77319198517722E+00,      -.73302221033734E+00, -.83273927638812E+00,
          -.92685579186241E+00, -.96496205741284E+00, -.91661343224195E+00,      -.91344179675966E+00, -.9346377509811E+00,
          -.84200648976864E+00, -.84018921018152E+00, -.88180515709993E+00,      -.92824400492308E+00, -.95831328884655E+00,
          -.9914014224665E+00, -.97507961387045E+00, -.94178748150775E+00,       -.74575452666014E+00, -.71707517665897E+00,
          -.8423028640259E+00, -.86054835028786E+00, -.91170643000648E+00,       -.97059608204182E+00, -.74493866690373E+00,
          -.92781840774007E+00, -.70712767862938E+00, -.56127962068486E+00,      -.97103920979785E+00, -.92708465969783E+00,
          -.97564802646607E+00, -.92272857126206E+00, -.76508328662255E+00,      -.72011063465216E+00, -.82026460379756E+00,
          -.92051316831144E+00, -.55134478405896E+00, -.98148189029938E+00,      -.90552503766282E+00, -.96786456908414E+00,
          -.92343523599538E+00, -.76120244650771E+00, -.71312757845834E+00,      -.8048184838361E+00, -.90943338823961E+00,
          -.96755483399991E+00, -.94024365540109E+00, 0.0, 0.0, 0.0, 0.0, 0.0,   0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
          0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
OGI       -.99571653182067E+00, -.95278668795628E+00, -.92977417601422E+00,      -.96473035096325E+00, -.83931530923177E+00,
          -.88108655172185E+00, -.9560175523689E+00, -.94280239683172E+00,       -.99946668264813E+00, -.96434060366449E+00,
          -.74748216542431E+00, -.90032053868088E+00, -.97225336835133E+00,      -.99427773862803E+00, -.92572788669819E+00,
          -.99283934965684E+00, -.99897134994753E+00, -.99261590906344E+00,      -.83214485071733E+00, -.75653906017337E+00,
          -.89833768332524E+00, -.92285408009627E+00, -.78828474089686E+00,      -.82111839595694E+00, -.96474726678335E+00,
          -.93670072855566E+00, -.79381674253358E+00, -.8808405468923E+00,       -.94761004466038E+00, -.99671110630028E+00,
          -.99425453634863E+00, -.99464221727485E+00, -.97411524169175E+00,      -.97457918813098E+00, -.85051457820339E+00,
          -.91452472423283E+00, -.96152890934078E+00, -.99689707221875E+00,      -.85342643886951E+00, -.91199657662224E+00,
          -.9519194948728E+00, -.99796680302194E+00, -.91180327277382E+00,       -.94280095477239E+00, -.97248129706004E+00,
          -.9922875307293E+00, -.97881077138869E+00, -.99838170122538E+00,       -.99611579076635E+00, -.98543747240342E+00,
          -.84046126583635E+00, -.92015808710181E+00, -.98060156268971E+00,      -.9974137775222E+00, -.83542566600255E+00,
          -.88186594296228E+00, -.95838261295228E+00, -.99940133814644E+00,      -.1E+01, -.1E+01, -.1E+01, -.1E+01,
          -.1E+01, -.1E+01, -.1E+01, -.1E+01, -.1E+01, -.1E+01, -.1E+01,         -.1E+01, -.1E+01, -.1E+01, -.1E+01,
          -.1E+01, -.73968974665649E+00, -.71989799093343E+00,                   -.8511666297607E+00, -.90946623369869E+00,
          -.92579316511445E+00, -.62138791415929E+00, -.65106179155173E+00,      -.66952913185199E+00, -.86838998001038E+00,
          -.92079324083711E+00, -.90775305520754E+00, -.89595924591247E+00,      -.91185785808518E+00, -.90403911307041E+00,
          -.88673784984652E+00, -.89475684350496E+00, -.45413763638525E+00,      -.13258722013038E+00, -.11265722125307E+00,
```

PCONRST (Cont.)

```
         -.24574418109944E+00, -.37252680700796E+00, -.1277757083191E+00,      -.14069267080698E+00, -.67131421173064E+00,
         -.66507304183887E+00, -.78599491280269E+00, -.79962619034101E+00,      -.78648540290769E+00, -.84843894794398E+00,
         -.9012625127168E+00, -.68563837392492E+00, -.44630198363183E+00,       .51640634559647E-04, -.32892600834779E+00,
         -.24943260855685E+00, -.23938086529206E+00, -.38544758031867E+00,      -.39376147400844E+00, -.46157025269697E+00,
         -.68328205980549E+00, -.72550634975451E+00, -.71499687253023E+00,      -.69415476505674E+00, -.72640000804193E+00,
         -.81968862721173E+00, -.85820315564034E+00, -.79979623889693E+00,      -.15156346082392E-02, -.80100634622976E+00,
         -.83017223254758E+00, -.74685239264797E+00, -.78238030234334E+00,      -.87646444404299E+00, -.87021102392971E+00,
         -.85536227795987E+00, -.84134608903399E+00, -.8632017810377E+00,       -.870002821107E+00, -.76978718295792E+00,
         -.77922095780907E+00, -.98521723872972E+00, -.98012451403279E+00,      -.98413107002976E+00, -.99541179420632E+00,
         -.99446876229629E+00, -.96210799401357E+00, -.97034013339282E+00,      -.9619855144805E+00, -.98125718230776E+00,
         -.98156422450367E+00, -.97455981195683E+00, -.95360592271773E+00,      -.94710682135374E+00, -.96428256056183E+00,
         -.97383641929083E+00, -.98546675701785E+00, -.97990507070664E+00,      -.97987107140991E+00, -.97916498325269E+00,
         -.97568572267122E+00, -.97597122290727E+00, -.9745631138003E+00,       -.99173926211893E+00, -.98866056518597E+00,
         -.99577916131596E+00, -.99241746260864E+00, -.99323757570048E+00,      -.95338672837885E+00, -.94911390878708E+00,
         -.978056043562E+00, -.97767846314869E+00, -.98205964619667E+00,        -.98533338255271E+00, -.94898290973881E+00,
         -.97195026947835E+00, -.94051829610329E+00, -.91611296668651E+00,      -.97570729368604E+00, -.98070556144624E+00,
         -.98869935028841E+00, -.98263149395566E+00, -.95592082158014E+00,      -.94695508505965E+00, -.96583882604775E+00,
         -.98117265101567E+00, -.91266273291119E+00, -.98976405401755E+00,      -.98060451499322E+00, -.99319984910193E+00,
         -.98755141653778E+00, -.95667596581468E+00, -.94820328914997E+00,      -.96731843183851E+00, -.98253508945565E+00,
         -.98235876947143E+00, -.98601945721173E+00, -.999E+00,                 -.99535212980241E+00, -.99565864518007E+00,
         -.9995E+00, -.9995E+00, -.99591798794941E+00, -.99957561990303E+00,    -.9985E+00, -.9995E+00, -.97110348372347E+00,
         -.97110348372347E+00, -.998E+00, -.99187821009216E+00,                 -.99207662481976E+00, -.99197741745596E+00,
         -.9895E+00, -.99454001647515E+00, -.992676288105E+00,                  -.99360815229007E+00, -.98849999999999E+00,
         -.92095023976184E+00, -.92095023976184E+00, -.92095023976184E+00,      -.996E+00, -.97280786237144E+00, -.98117781068303E+00,
         -.9995E+00, -.9995E+00, -.97841413909813E+00, -.98645886699162E+00,    -.9955E+00, -.9995E+00, -.99837878197616E+00,
         -.99837878197616E+00, -.995E+00, -.98014396447397E+00,                 -.96669896713244E+00, -.99327750132923E+00,
         -.9865E+00, -.98815523639225E+00, -.98261657472901E+00,                -.98538590556063E+00, -.98549999999999E+00,
         -.87083655602638E+00, -.87083655602638E+00, -.87083655602638E+00,      -.99299999999999E+00, -.95958280823957E+00,
         -.98772025745439E+00, -.9995E+00, -.9995E+00, -.99052830345821E+00,    -.99427610641199E+00, -.9925E+00, -.9995E+00,
         -.96460249550426E+00, -.96460249550426E+00, -.992E+00, -.991E+00,      -.96911460483604E+00, -.96911460483604E+00,
         -.9905E+00, -.95444794958863E+00, -.95444794958863E+00,                -.98999999999999E+00, -.97110348372347E+00,
         -.97110348372347E+00, -.9895E+00, -.99187821009216E+00, -.9895E+00,    -.97779066206843E+00, -.973066204476E+00,
         -.97779066206843E+00, -.92095023976184E+00, -.92095023976184E+00,      -.92095023976184E+00, -.90929427774941E+00,
         -.90929427774941E+00, -.90929427774941E+00, -.93422268454974E+00,      -.93422268454974E+00, -.98699999999999E+00,
         -.93489258324666E+00, -.95465946963056E+00, -.98014396447397E+00,      -.96669896713244E+00, -.96669896713244E+00,
         -.96343319687055E+00, -.95760171241116E+00, -.96051745464085E+00,      -.87083655602638E+00, -.87083655602638E+00,
         -.87083655602638E+00, -.84645516794456E+00, -.84645516794456E+00,      -.84645516794456E+00, -.90391088590332E+00,
         -.90391088590332E+00, -.90391088590332E+00, -.95160618946586E+00,      -.95455588846128E+00, -.96460249550426E+00,
         -.95997959731942E+00, -.97635013804871E+00, -.967E+00,                 -.95254819716267E+00, -.95254819716267E+00,
         -.9665E+00, -.95026798790877E+00, -.95026798790877E+00,                -.96599999999999E+00, -.96364007981332E+00,
         -.96364007981332E+00, -.90888641161881E+00, -.90888641161881E+00,      -.90888641161881E+00, -.90781353009334E+00,
         -.90781353009334E+00, -.90781353009334E+00, -.93221102031217E+00,      -.93221102031217E+00, -.93221102031217E+00,
         -.95876147078707E+00, -.95876147078707E+00, -.95876147078707E+00,      -.97507176435795E+00, -.97507176435795E+00,
         -.96299999999999E+00, -.97533419127548E+00, -.9903981902304E+00,       -.9625E+00, -.98356580237213E+00, -.98356580237213E+00,
         -.962E+00, -.94627722499904E+00, -.92107688231724E+00,                 -.94949947037598E+00, -.99251617397503E+00,
         -.95852573485126E+00, -.86072846668075E+00, -.84284953818756E+00,      -.89292910280282E+00, -.95995527565108E+00,
         -.96807400959003E+00, -.92550907309448E+00, -.77319198517722E+00,      -.73302221033734E+00, -.83273927638812E+00,
         -.92685579186241E+00, -.96496205741284E+00, -.91661343224195E+00,      -.91344179675966E+00, -.9346377509811E+00,
         -.84200648976864E+00, -.84018921018152E+00, -.88180515709993E+00,      -.92824400492308E+00, -.95831328884655E+00,
         -.99140142246665E+00, -.97507961387045E+00, -.94178748150775E+00,      -.74575452666014E+00, -.71707517665897E+00,
         -.8423028640259E+00, -.86054835028786E+00, -.91170643000648E+00,       -.97059608204182E+00, -.74493866690373E+00,
         -.92781840774007E+00, -.70712767862938E+00, -.56127962068486E+00,      -.97103920979785E+00, -.92708465969783E+00,
         -.97564802646607E+00, -.92272857126206E+00, -.76508328662255E+00,      -.72011063465216E+00, -.82026460379756E+00,
         -.92051316831144E+00, -.55134478405896E+00, -.98148189029938E+00,      -.90552503766282E+00, -.96786456908414E+00,
         -.92343523599538E+00, -.76120244650771E+00, -.71312757845834E+00,      -.8048184838361E+00, -.90943338823961E+00,
         -.96755483399991E+00, -.94024365540109E+00, 0.0, 0.0, 0.0, 0.0, 0.0,   0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
         0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
OGRDOBJ = -.62369331589676E+04, -.82548591749565E+04, -.12837350593944E+03,     0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
         0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
OGRDG   = .11175061786934E+00, .34600749035647E-02, -.79117958740418E-03, 0.0,  0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
         0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, .17235505730022E+00,      .89177682547543E-01, .84745070683899E-03,
         0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,  0.0, 0.0, 0.0, .40430966511939E+00,
         .29714799662722E-01, .51315734380887E-02, 0.0, 0.0, 0.0, 0.0, 0.0,     0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
         0.0, 0.0, 0.0, 0.0, 0.0, .21620472008486E+00, -.15838774815508E-01,    .39416439478602E-02, 0.0, 0.0, 0.0,
         0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,  .38712494347378E+00, .41671492643 6E+00,
         -.74155340247519E-03, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,     0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
         0.0, .43225111518631E+00, .27832493394894E+00, -.65372018622594E-03,   0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
         0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,                      .27638185337384E+00, .45736845277465E-01,
         .14972987621531E-02, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,      0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
         0.0, .15671714558983E+00, .12417264607074E+00, .71253123140573E-03,    0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
         0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,                      .38678028457312E-03, .20106807852096E-01,
         -.9807320042662E-03, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,      0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
         0.0, .44252195564294E-01, .35948889669762E-01, .3310828279029E-02,     0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
         0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,                      .17011225905968E+01, .28449488889964E+00,
         .58216467098667E-02, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,      0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
         0.0, .32101842783761E+00, .17671642757389E+00, .31355616809258E-02,    0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
         0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,                      .87632893066816E-01, .88815863339087E-01,
         -.12542660247226E-02, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,     0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
         0.0, .19616835418874E-01, .36196258401458E-01, -.12221418094886E-02,   0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
         0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,                      .30179203982636E+00, .83361389678935E-01,
         .40502392474683E-02, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,      0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
         0.0, .55145262286079E-01, -.33504405991778E-01, .24993242123 18E-02,   0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
         0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,                      .47036147813921E-02, -.39649864990256E-03,
         .12199638207825E-03, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,      0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
         0.0, .64596341109715E-01, -.26526777356O3E-01, .20398447545648E-02,    0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
         0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,                      .36098935822495E+00, .34942011638273E+00,
         .43368472888192E-02, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,      0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
         0.0, .16015178867559E+01, .3314067867746E+00, .30378447599813E-02,     0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
         0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,                      .91368130079642E+00, .82299200815036E-01,
         .14082057618438E-02, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,      0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
         0.0, .31369620482007E+00, .12405841456805E+00, .2350035531435E-02,     0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
         0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,                      .35781298034488E+00, .50526148189718E+00,
         .31130741989225E-02, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,      0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
         0.0, .56521633012878E+00, .38521024404618E+00, .17840069496837E-02,    0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
         0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,                      .38033921282695E+00, .32746080120391E-01,
         -.45018766653 9E-03, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
         .25392504904251E+00, .13092356814074E+00, .31350647348645E-03, 0.0,    0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
         0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, .71467132731879E+00,      .28544584180415E+00, .97965538929392E-02,
```

PCONRST (Cont.)

```
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
.13958501261295E+00, .70720798834731E-02, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, .16141272351762E-01, .18144852077654E+00,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
-.38665953807382E-02, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, -.20117460006483E+00, .12512541013345E+00, -.33958625694967E-02,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
-.15218066916739E-02, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, .20870437293292E+00, -.40515271505139E-02, .19415818940305E-02,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
.10112597711589E-02, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
.35801412196865E+00, .27742811949061E+00, .52105001079278E-02, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, .10676363368248E+00,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
.13926638313747E+00, -.77045222265422E-03, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, -.10136846851826E+00, .80128536092242E-01,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
.61546430360825E-02, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, .2373811931183E+00, .15767064957984E+00, .30151059886523E-02,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
.13874906855066E-02, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, -.49112578064658E-01, .24277272569012E-01, -.34761169896338E-03,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
.61627325188233E-02, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, .25303547307784E+00, .30899936575527E-01, .46303397315202E-02,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
.27309263353134E-02, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, .34597334181714E-01, -.43418797191134E-03, .84366439208319E-03,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, .7130537989372E-02,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
-.69722687244906E-01, .27931727435596E-02, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, .87587047150541E-01, -.5318159306574E-01,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
.14041427458492E-02, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, .52595265636248E+00, .23678196736433E+00, .70472428019711E-02,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
.42772622035854E-02, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, -.51061946122292E-01, .61539608252088E-01, .44286640043488E-03,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
.1777638368059E-02, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
.46764091601667E+00, .28269801268097E+00, .60921043750051E-02, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, .39232829401797E+00,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
.39588219635666E-01, .34682647664871E-02, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, -.13044049333427E-01, -.62237516349282E-02,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
.58477603276422E+00, -.18758563887932E-02, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, .22742016476521E+00, .62198432221886E+00,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
-.14506797922134E-02, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, .14244480233015E+00, .19457513109013E-01, .83179539777372E-02,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
.1177958397172E-01, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
.26933875739586E+00, .79876832491923E+00, .32886812080176E-03, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, .7221146831188E-01,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
.78061635219216E+00, -.27500791677151E-02, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, .2930464605462E+00, -.26881913506822E-01,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
.76313120569694E-02, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, .11056219117385E+00, .34210378805843E-01, .7870337944141E-02,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
.81598290622494E-02, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, .24093675032251E+00, .82500798590413E-01, .34026043792563E-02,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
.10085609780922E-01, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, .13049135114181E+00, .24030986038986E+00, -.79236072629342E-03,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
-.37649029253786E-03, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, .23886104533813E+00, .12103528521657E+01, -.53865673235637E-03,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
.67144986483219E-02, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, .15148957888208E+00, .18582524577901E+01, .74850744858743E-02,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
.99267426220445E-02, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, -.65558667353975E-01, .14471495421493E+01, -.40904036940829E-04,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
.77176831531633E-02, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, .14727846230267E+00, .17769840040871E+01, .79235529103828E-02,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
.11626085294917E-02, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, .12046294194177E+00, .7226895447098E+00, .85296295404411E-03,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, .40564165811574E+00,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
-.14522896835556E-02, 0.0, 0.0, 0.0,
-.14127270399955E+00, .11190917091779E+00,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
-.81484115272104E-01, .63169466458283E-01,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
.13809466775123E+00, .27124535258028E-01,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
.18851596036571E+00, .25100137998077E-02,
0.0, 0.0, 0.0, -.36226937919774E-01,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
-.25601165959665E-02, 0.0, 0.0, 0.0,
.47874576560611E+00, .22435243156216E+00,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
.11097665552579E+00, .94855528120199E-01,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
.38359886409389E+00, .68009754364761E-01,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
.12213550144254E+00, .52415043434854E-02,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
.92815567340586E-01, -.16071559966697E-02,
0.0, 0.0, 0.0, .10268689835079E+00,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
.24713279002538E-02, 0.0, 0.0, 0.0,
.10829513739771E+00, -.68779027305205E-02,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
.18544970776681E+00, .12106564115862E+00,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
.84048000397322E-01, -.42953613429226E-01,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
.16027119148499E+00, .59761087332392E-02,
0.0, 0.0, 0.0, .10051988024718E+00,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
.46845356059621E-03, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, .19749204431528E+00,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
-.19350950347007E-02, 0.0, 0.0, 0.0,
.10473706108639E+00, .34586307670773E+00,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
.22236106293466E+00, -.10239485426244E+00,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
.80404186886496E+00, -.14054831485858E-02,
0.0, 0.0, 0.0, 0.0, .96245163702595E-01,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
.1379389636856E-01, 0.0, 0.0, 0.0,
.74951894644574E-01, .19253682047446E-01,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
.25170715079147E+00, .24742621628056E-01,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
.20858369150984E+00, -.18928902012601E-01,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
.10460119701461E+00, .22141247995553E+00,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
.15792606895573E+00, .18138100838745E+01,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
.15366462409591E+00, .15025417741218E+01,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
.14920839197454E+00, .18197180737315E+01,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
.14198647939564E+00, .70171424587551E+00,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
.41000732384468E-01, .27849058317184E+00,
```

PCONRST (Cont.)

```
.10640659178729E-01, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, .24896118761395E+00, .60673598689311E-01, .18173372233225E-01,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
.12796402936399E-01, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, .45081538085297E+00, .82860590795884E-01, .86410647762047E-02,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
.25385629781014E-01, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, .51298845019223E+00, .13111837854774E+00, .25115192963076E-01,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
-.90190401835964E-03, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, .36137419273191E+00, .19741256071347E+01, .10898298933072E-01,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
.24429210957135E-01, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, .22168578516648E+00, .14461251093558E+01, .1170955793495E-01,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
.97293000513285E-02, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, -.17126915146661E+00, .14711033649744E+01, -.10724170877246E-02,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
.13336785079473E-02, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, -.79932526873573E-02, .93486430725812E+00, .15646038224943E-01,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
.17322150202145E-01, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, -.90614115141107E-01, .61761815735158E+00, .16243900419809E-02,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
.23069865580441E-01, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, .43947185444049E+00, .16594365095785E+00, .23239692775255E-01,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
.249443669957E-01, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
.95175419080312E-01, .17268731051117E+00, .1237251228895E-01, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, -.10204997854568E-01,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
-.35994166663685E-01, .19895511709004E-01, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, .35613174811997E+00, .1972210022928E+01,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
-.8604892080408E-02, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, .50226388981947E+00, .33349548313188E+00, -.31347316262561E-02,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, .7974117289016E+00,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
.36990305410963E+00, -.13210842361568E-02, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, -.73194022580029E-02, .24601168914302E+00,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
.14652256462299E-02, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, .1688556251039E+00, .42522840772583E-01, .13223321979383E-01,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
.11383975233041E-01, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, .36191665771971E+00, .11079913670846E+00, .6402934854896E-02,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
.19548682928973E-01, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, .35931592041629E+00, .27024740705131E+00, .92989132565443E-02,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
.91385158994834E-02, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, .44436586119941E-01, .24725214280693E-01, .55414645157803E-03,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
.16092170281996E-02, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, .13692813612373E-02, .24004699604187E-01, .12745435870571E-02,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
-.10474793089762E-03, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, .20611096430546E-01, .84845271191664E-02, .20147356402402E-03,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, .27394054741O5E-01,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
.20320289402327E-01, .40810839826639E-02, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, .72026866685515E-01, .58674550489162E-01,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
.84352206980659E-03, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, -.17365320289 91E-02, .43064305405424E-01, .71163651258482E-03,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
.15098467091172E-02, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, .29865856222391E-01, .26133418595517E-01, .57687377385144E-02,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
.70018790131 36E-02, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
-.962532070582E-02, .32304155596924E-01, .41741154782306E-02, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, .75355894356785E-01,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
.39608913422407E-01, .11349601732948E-03, 0.0, 0.0, 0.0, 0.0, C.0,
0.0, 0.0, 0.0, 0.0, 0.0, .88253635640236E-02, .18875031749024E-01,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
.22622670613247E-02, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, -.57349517331 96E-02, .32764665873179E-01, .16874143016217E-02,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
.41257985273637E-02, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, .92972676694442E-02, -.77395048918233E-02, .41748663293975E-02,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
.30520609666973E-02, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, -.54306068406793E-02, -.91236208057097E-02, .18790024304008E-02,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
.10784351162359E-02, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, -.16706863504595E-01, .18451516875294E-01, -.12058469349251E-03,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
.63895958342839E-03, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, -.18763904988152E-01, -.31379452739383E-02, .14928279219745E-02,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, .4757612033788E-02,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
.13840668861568E-01, .74177489589336E-02, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, -.18617533590406E-01, -.66463961401388E-02,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
.36505542440637E-02, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, .93915558418856E-02, .90790048566953E-02, .23072245213819E-02,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
.42353860329073E+00, .14882798225572E+00,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
.38739789194388E+00, -.33782455126712E+00,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
.22506628944443E+00, .12334331966539E+01,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
.15774615297631E+00, .10415921504408E+01,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
.19164077143239E+00, .15200380398311E+01,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
-.12820384462642E+00, .13894703172631E+01,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
.40127823347146E+00, .30752056459397E+00,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
.40962617652672F+00, .12871763336994E+00,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
.44315919470525E+00, .69324293213204E-01,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
.84478521636109E-01, .12991933224338E-01,
0.0, 0.0, 0.0, .49823948117021E+00,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
.10945749929502E-01, 0.0, 0.0, 0.0,
.12686702778449E+00, .58421260042824E+00,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
.42024507195693E+00, -.10187518370855E-02,
0.0, 0.0, 0.0, .67860839530659F+00,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
.20769702748738E-02, 0.0, 0.0, 0.0,
-.33917349280779E-02, .26913313604 66E+00,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
.33077997478355E+00, .7263463538214E-01,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
.33969384715516E+00, -.14588661561138E+00,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
.33903556298212F+00, .25648916009014E+00,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
.75222689005108E-02, .28470788837539E-01,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
.78136313136254E-02, .14210712309179E-01,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
.44604263944997E-01, .3435195022594E-02,
0.0, 0.0, 0.0, -.31281391566367E-01,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
.21840889148939E-02, 0.0, 0.0, 0.0,
.43362346929269E-01, .31570167464114E-01,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
.56855901259013E-01, .36431833799193E-01,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
-.20806046805337E-02, .29772319956347F-01,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
.37631857026308E-01, .13484665025844E-02,
0.0, 0.0, 0.0, .10797882962838E-01,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
.21412614728244F-02, 0.0, 0.0, 0.0,
-.14321864264527E-01, .21990273959147E-01,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
.53835228358578E-02, -.51278488137006E-02,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
-.47341936648954E-02, .20963658231647E-01,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
.66933115722585E-02, .12674483860434E-01,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
-.79935735006045E-02, .12823248114846E-01,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
.15236815730804E-01, .66673450677587E-02,
0.0, 0.0, 0.0, .59152788064699E-02,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
.41357346077 54E-02, 0.0, 0.0, 0.0,
.76017367126313E-03, -.90303913911679E-03,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
-.12334081934864E-01, .3492130842306E-01,
```

PCONRST (Cont.)

```
.65708061702075E-03, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, -.44332497537557E-02, .33138075816567E-01, .65424978257623E-02,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
.18897109946852E-02, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, .19742783245321E-01, .36128236180278E-01, .74688452599401E-02,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
.11491268388763E-01, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, .73096480743526E-01, .45846142419852E-01, .62992044302357E-03,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
.18942018597352E-02, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, -.55886030128284E-02, .25458577509152E-01, .53809013781 83E-03,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
.15598852733978E-02, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, .17633526025378E-01, .15952033212487E-01, .60772692975856E-02,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
.73408227775358E-02, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, -.3037015980913E-01, .23074202068334E-01, .46556020646545E-02,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
.1742653574767E-02, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
-.33020183107639E-01, .51951289597245E-01, .11754174497623E-01, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, -.1943021637814E-01,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
.18737185124403E-01, .23605067414627E-02, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, -.86351994814501E-02, .79762220481051E-02,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
.17047068318855E-02, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, .10073821575653E-01, .1223539576134E-01, .62397317828833E-02,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
.75240155294456E-02, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, -.27293286284334E-01, .95094403116692E-02, .51068777334354E-02,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
.23468193526341E-02, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, .15238053282446E-01, .34398073715059E-01, .84071573422717E-03,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
.15239450063137E-02, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, -.19202380415542E+00, -.24854917897869E+00,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
-.12190671466 12E-03, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, -.83364346771093E+00, -.10790401761481E+01,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
-.13114255506152E-04, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, -.96011902077712E-01, -.12427458948935E+00,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
-.10706509802111E-03, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, -.81491080626493E-01, -.1054793246748E+00, -.11130858046571E-04,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
-.39342766518457E-04, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, -.96011902077712E-01, -.12427458948935E+00,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,

-.7579125953761E-03, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, -.55488189822576E+01, -.71822053958748E+01, -.7579125953761E-03,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
-.5245702202461E-04, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, -.15595769946543E+01, -.20186642134307E+01,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
-.20781833316946E-03, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, -.1540526819257E+01, -.19940063046093E+01, -.21042039460342E-03,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
-.2753993656292E-03, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, -.10484468070669E+01, -.13570744223376E+01,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
-.1920900580889E-03, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, -.12273869128414E+01, -.15886885004579E+01,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
-.30162787766415E-03, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, -.15179435678504E+02, -.19647753005655E+02,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
-.20733575069264E-02, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, -.15179435678504E+02, -.19647753005655E+02,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
-.10491404404922E-03, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, -.52215377105537E+01, -.67585834822538E+01,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
-.49367799977569E-03, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, -.96011902077712E-01, -.12427458948935E+00,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
-.13114255506152E-04, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, -.41449991263458E+01, -.53651480047066E+01,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
-.35516375622939E-03, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, -.8641071186994E+00, -.11184713054041E+01, -.11802829955537E-03,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
-.13114255506152E-04, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, -.3113124523017E+00, -.40295240876908E+00, -.42522134791443E-04,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
-.42522134791443E-04, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, -.96011902077712E+00, -.12427458948935E+01,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
-.52079424645504E-03, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, -.63945910135324E+01, -.82769443783728E+01,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
-.17632113041636E-03, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, -.25923213560982E+01, -.33554139162123E+01,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
-.31067051272383E-03, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, -.33380314497861E+01, -.43206360789407E+01,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
-.38330593693772E-03, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
```

```
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
-.59234604600411E-02, .50921324914377E-01,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
-.2670769172386E-01, .45035593771627E-01,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
-.10364410327952E-01, .2525662P996171E-01,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
.11968800269736E-01, .20747144730792E-01,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
.33559029945334E-02, .22671454461862E-01,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
.32342285680296E-01, .17366335104058E-01,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
.44198298136115E-01, -.48546754539951E-03,
0.0, 0.0, 0.0, -.18805121298937E-01,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
.77465877754561E-03, 0.0, 0.0, 0.0,
-.10709504179889E-01, .81424457536747E-02,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
.50493519447965E-02, .15035615164244E-01,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
-.83733756371947E-02, .10903621894581E-01,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
.12261787688162E-01, .11310810707713E-01,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
-.26228511012305E-04, 0.0, 0.0, 0.0,
-.89250171656147E+00, -.115522432161E+01,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
-.11386727270279E-03, 0.0, 0.0, 0.0,
-.96011902077712E-01, -.12427458948935E+00,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
-.13114255506152E-04, 0.0, 0.0, 0.0,
-.78384348256147E+00, -.10145807437538E+01,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
-.28803570623313E+00, -.37282376846804E+00,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
-.13114255506152E-04, 0.0, 0.0, 0.0,
-.55488189822576E+01, -.71822053958748E+01,

0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
-.38404760831085E+00, -.49709835795738E+00,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
-.21302245603737E-03, 0.0, 0.0, 0.0,
-.15214766438597E+01, -.1969348395788E+01,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
-.20162499436319E+01, -.26097663792763E+01,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
-.14320723800842E-03, 0.0, 0.0, 0.0,
-.14063270186158E+01, -.18203025785782E+01,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
-.16764864804866E-03, 0.0, 0.0, 0.0,
-.22082737477874E+01, -.28583155582549E+01,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
-.20733575069264E-02, 0.0, 0.0, 0.0,
-.15179435678504E+02, -.19647753005655E+02,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
-.20733575069264E-02, 0.0, 0.0, 0.0,
-.7680952166217E+00, -.99419671591476E+00,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
-.71320928123876E-03, 0.0, 0.0, 0.0,
-.36143083951771E+01, -.46782397013131E+01,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
-.13114255506152E-04, 0.0, 0.0, 0.0,
-.96011902077712E-01, -.12427458948935E+00,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
-.56616499037462E-03, 0.0, 0.0, 0.0,
-.26002198728438E+01, -.33656374916742E+01,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
-.96011902077712E-01, -.12427458948935E+00,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
-.3113124523017E+00, -.40295240876908E+00,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
-.13114255506152E-03, 0.0, 0.0, 0.0,
-.38128314771524E+01, -.4935201327765E+01,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
-.87343650728776E-03, 0.0, 0.0, 0.0,
-.129087976819E+01, -.16708715253039E+01,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
-.35408489866611E-03, 0.0, 0.0, 0.0,
-.22744765672805E+01, -.29440062699012E+01,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
-.45594136115161E-03, 0.0, 0.0, 0.0,
-.28062540085333E+01, -.36323211744421E+01,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
```

PCONRST (Cont.)

```
0.0, -.27843451602536E+01, -.3603963095191E+01, -.38031340967842E-03, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, -.2480245586963E+02, -.32103467953703E+02,
-.33877648126493E-02, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, -.2480245586963E+02, -.32103467953703E+02, -.33877648126493E-02, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, -.2480245586963E+02, -.32103467953703E+02,
-.33877648126493E-02, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, -.1344166629088E+01, -.17398442528508E+01, -.18359957708613E-03, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, -.77610629151172E+01, -.10045659828679E+02,
-.10600827591749E-02, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, -.23580028776566E+01, -.30521199277802E+01, -.32207936258571E-03, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, -.96011902077712E-01, -.12427458948935E+00,
-.13114255506152E-04, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, -.96011902077712E-01, -.12427458948935E+00, -.13114255506152E-04, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, -.18187912017599E+01, -.23541823989967E+01,
-.24842849705149E-03, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, -.10991238213508E+01, -.14226690518616E+01, -.15012920600642E-03, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, -.14401785311657E+01, -.18641188423402E+01,
-.19671383259229E-03, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, -.96011902077712E-01, -.12427458948935E+00, -.13114255506152E-04, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, -.67971634708799E+01, -.87980206803096E+01,
-.92842383647452E-03, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, -.67971634708799E+01, -.87980206803096E+01, -.92842383647452E-03, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, -.15361904332434E+01, -.19883934318295E+01,
-.20982808809844E-03, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, -.17282142373988E+01, -.22369426108082E+01, -.23605659911074E-03, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, -.5930731072227E+01, -.76765396104345E+01,
-.81007792717729E-03, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, -.5930731072227E+01, -.76765396104345E+01, -.81007792717729E-03, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, -.18242261394765E+01, -.23612172002976E+01,
-.24917085461689E-03, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, -.87470780070707E+01, -.11321924730542E+02, -.11947624558477E-02, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, -.87470780070707E+01, -.11321924730542E+02,
-.11947624558477E-02, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, -.19202380415542E+01, -.24854917897869E+01, -.26228511012305E-03, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, -.55488189822576E+01, -.71822053958748E+01,
-.7579125953761E-03, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, -.55488189822576E+01, -.71822053958748E+01, -.7579125953761E-03, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, -.20162499436319E+01, -.26097663792763E+01,
-.2753993656292E-03, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, -.15595769946543E+01, -.20186642134307E+01, -.21302245603737E-03, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, -.20162499436319E+01, -.26097663792763E+01,
-.2753993656292E-03, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, -.42424547428491E+01, -.54912912897816E+01, -.57947644268082E-03, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, -.5171929876863E+01, -.66943727642705E+01,
-.70643335250443E-03, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, -.42424547428491E+01, -.54912912897816E+01, -.57947644268082E-03, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, -.15179435678504E+02, -.19647753005655E+02,
-.20733575069264E-02, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, -.15179435678504E+02, -.19647753005655E+02, -.20733575069264E-02, 0.0, 0.0, 0.0,

0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, -.15179435678504E+02, -.19647753005655E+02,
-.20733575069264E-02, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, -.17417657845222E+02, -.22544832794052E+02, -.23790760349285E-02, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, -.17417657845222E+02, -.22544832794052E+02,
-.23790760349285E-02, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, -.17417657845222E+02, -.22544832794052E+02, -.23790760349285E-02, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, -.1263081033989E+02, -.16348897750584E+02,
-.17252410426469E-02, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, -.1263081033989E+02, -.16348897750584E+02, -.17252410426469E-02, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, -.24963094540205E+01, -.3231139326723E+01,
-.34097064315996E-03, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, -.12502173843707E+02, -.16182394979464E+02, -.17076705972975E-02, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, -.87064611239647E+01, -.11269351597783E+02,
-.11892146000986E-02, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, -.38128314771524E+01, -.4935201327765E+01, -.52079424645504E-03, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, -.63945910135324E+01, -.82769443783728E+01,
-.87343650728776E-03, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, -.63945910135324E+01, -.82769443783728E+01, -.87343650728776E-03, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, -.70216966427199E+01, -.90886488957007E+01,
-.95909279856562E-03, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, -.81414804724839E+01, -.10538059570307E+02, -.11120439529266E-02, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, -.75815885576019E+01, -.98133542330041E+01,
-.10355683757461E-02, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, -.2480245586963E+02, -.32103467953703E+02, -.33877648126493E-02, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, -.2480245586963E+02, -.32103467953703E+02,
-.33877648126493E-02, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, -.2480245586963E+02, -.32103467953703E+02, -.33877648126493E-02, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, -.2948426275969E+02, -.381634419438F+02,
-.40272523184485E-02, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, -.2948426275969E+02, -.381634419438E+02, -.40272523184485E-02, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, -.2948426275969E+02, -.381634419438E+02,
-.40272523184485E-02, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, -.18451397226768E+02, -.23882870417518E+02, -.25202743872473E-02, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, -.18451397226768E+02, -.23882870417518E+02,
-.25202743872473E-02, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, -.18451397226768E+02, -.23882870417518E+02, -.25202743872473E-02, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, -.92927635963407E+01, -.12028241875909E+02,
-.12692975925219E-02, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, -.87263511741278E+01, -.11295096612364E+02, -.11919313799376E-02, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, -.67971634708799E+01, -.87980206803096E+01,
-.92842383647452E-03, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, -.76848699665569E+01, -.99470382286548E+01, -.10496755724245E-02, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, -.45413364596368E+01, -.58781537709519E+01,
-.62030066462883E-03, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, -.6336785537129E+01, -.82021229062968E+01, -.86554086340606E-03, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, -.9111875694857E+01, -.11794106636277E+02,
-.12445901332726E-02, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, -.9111875694857E+01, -.11794106636277E+02, -.12445901332726E-02, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, -.64327974392067E+01, -.83263974957861E+01,
```

APPENDIX E

PCONRST (Cont.)

```
-.87865511891221E-03, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, -.9549730150062E+01, -.12360850774234E+02, -.1304396626799E-02,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
-.1304396626799E-02, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, -.65288093412844E+01, -.84506720852755E+01,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
-.95366656702264E-03, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, -.69819701930323E+01, -.90372283101279E+01,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
-.23897737562265E-02, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, -.17495977851207E+02, -.22646207585946E+02,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
-.23897737562265E-02, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, -.17701996643136E+02, -.22912871408243E+02,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
-.24179138411323E-02, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, -.17701996643136E+02, -.22912871408243E+02,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
-.1778004000255E-02, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, -.13017097759471E+02, -.16848895245212E+02, -.1778004000255E-02,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
-.1778004000255E-02, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, -.79187792572394E+01, -.1024980257816E+02, -.10816252175923E-02,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
-.10816252175923E-02, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, -.79187792572394E+01, -.1024980257816E+02, -.10816252175923E-02,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
-.65383050305476E-03, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, -.47868146388687E+01, -.61958925022181E+01,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
-.97045490745528E-03, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, -.47364224238518E+01, -.61306665073242E+01, -.6469474357584E-03,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
-.2518411732799E-03, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, -.72008926558284E+01, -.93205942117009E+01,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
-.43104453346088E-03, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, -.31557571467449E+01, -.40847063275797E+01,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
-.99668341846758E-03, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, -.1031605162547E+02, -.13352751618946E+02, -.14090683957243E-02,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
-.20700358612677E-02, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, -.96973038102681E+01, -.12551865176039E+02,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
-.19628961331017E-03, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, -.79640461684122E+01, -.10308394551464E+02, -.1087808220181E-02,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
-.36528849453643E-02, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, -.30176629502006E+02, -.39059618259603E+02,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
-.28083102062335E-02, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, -.76895403058353E+01, -.99530833593525E+01,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
-.83737119104673E-03, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, -.14304031159461E+02, -.18514658723728E+02,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
-.59488365144583E-02, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, -.51266090796031E+02, -.663571104262E+02, -.70024298962079E-02,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
-.43869997311803E-02, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, -.14045429098516E+02, -.18179932879644E+02,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
-.91899306299578E-03, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, -.16012205956364E+02, -.20725662954113E+02,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
-.22702927868946E-02, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, -.12551107704764E+02, -.16245733329849E+02,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
-.4143934522976E-02, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, -.30687475806027E+02, -.39720840601319E+02,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
-.31000747386022E-02, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, -.13778859145628E+02, -.17834893663167E+02,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
-.10933803625549E-02, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, -.16511315683071E+01, -.21371693863348E+01,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
-.65362462202977E-03, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, -.1117818925035E+02, -.14468673677531E+02, -.1526827682328E-02,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
-.66684801973231E-02, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, -.54328300867945E+02, -.70320732554104E+02,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
-.41361610675056E-02, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, -.26778036273474E+02, -.34660593043175E+02,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
-.23158088728909E-02, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, -.56462521834028E+01, -.73083196672612E+01,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
-.66898789839286E-02, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, -.13860583935747E+02, -.1794067549358E+02, -.18932156874752E-02,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
-.76816049062684E-02, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, -.84244756196603E+02, -.10904359008E+03, -.1150698230019E-01,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
-.75959840474209E-03, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, -.14001481026108E+02, -.18123047967054E+02,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
-.9549730150062E+01, -.12360850774234E+02,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
-.89176937441836E-03, 0.0, 0.0, 0.0,
-.69819701930323E+01, -.90372283101279E+01,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
-.95366656702264E-03, 0.0, 0.0, 0.0,
-.17495977851207E+02, -.22646207585946E+02,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
-.23897737562265E-02, 0.0, 0.0, 0.0,
-.17495977851207E+02, -.22646207585946E+02,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
-.24179138411323E-02, 0.0, 0.0, 0.0,
-.17701996643136E+02, -.22912871408243E+02,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
-.24179138411323E-02, 0.0, 0.0, 0.0,
-.13017097759471E+02, -.16848895245212E+02,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
-.13017097759471E+02, -.16848895245212E+02,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
-.79187792572394E+01, -.1024980257816E+02,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
-.47868146388687E+01, -.61958925022181E+01,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
-.65383050305476E-03, 0.0, 0.0, 0.0,
-.71048807537507E+01, -.91963196222115E+01,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
-.18437760387345E+01, -.23865219349426E+01,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
-.98356916296143E-03, 0.0, 0.0, 0.0,
-.31557571467449E+01, -.40847063275797E+01,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
-.43104453346088E-03, 0.0, 0.0, 0.0,
-.72969045579061E+01, -.94448688011902E+01,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
-.1515511729325E+02, -.19616276102488E+02,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
-.13245536973709E-02, 0.0, 0.0, 0.0,
-.14370727429526E+01, -.18600988141263E+01,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
-.2674344963852E+02, -.34615825261601E+02,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
-.41218226182363E-02, 0.0, 0.0, 0.0,
-.20560160994136E+02, -.2661238359087E+02,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
-.10503134935704E-02, 0.0, 0.0, 0.0,
-.61305501299527E+01, -.79351787044804E+01,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
-.19537860966582E-02, 0.0, 0.0, 0.0,
-.4355253781921E+02, -.56372945869989E+02,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
-.32118040433742E+02, -.41572515529115E+02,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
-.19184636686231E-02, 0.0, 0.0, 0.0,
-.67281190253654E+01, -.87086518631406E+01,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
-.21871055107202E-02, 0.0, 0.0, 0.0,
-.16621235467068E+02, -.21513970349257E+02,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
-.17143544681813E-02, 0.0, 0.0, 0.0,
-.30338514866495E+02, -.39269157251967E+02,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
-.41915990606391E-02, 0.0, 0.0, 0.0,
-.22696223365225E+02, -.29377231162329E+02,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
-.18820529070738E-02, 0.0, 0.0, 0.0,
-.80048408584129E+01, -.10361197831511E+02,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
-.22552788552743E-03, 0.0, 0.0, 0.0,
-.47853073456181E+01, -.61939415123314E+01,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
-.48821182980016E+02, -.6319250365767E+02,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
-.74206968446546E-02, 0.0, 0.0, 0.0,
-.30281603954161E+02, -.39195493673653E+02,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
-.36576091301588E-02, 0.0, 0.0, 0.0,
-.1695446719262E+02, -.21945294330988E+02,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
-.77122098597091E-03, 0.0, 0.0, 0.0,
-.48977847474099E+02, -.63395284930287E+02,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
-.56238457281415E+02, -.72793175022249E+02,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
-.55611611059635E+01, -.71981806273179E+01,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
-.19124608060812E-02, 0.0, 0.0, 0.0,
```

PCONRST (Conc.)

```
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, -.46761585966768E+01, -.60526630283683E+01,
-.63871600600596E-03, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,     0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, -.14837953698783E+02, -.19205750171324E+02,                       -.20267145195894E-02, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, -.45109600962427E+02, -.58388356238337E+02,
-.61615156037949E-02, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,     0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, -.53745420676744E+02, -.69566271962072E+02, -.73410813012528E-02, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,                      -.34513474520174E+02, -.44673085159525E+02,
-.47141918185965E-02, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,     0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, -.152633638011E+02, -.1975638675581E+02, -.20848212402767E-02,    0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,                      -.8615248131917E+02, -.11151288556665E+03,
-.11767558272037E-01, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,     0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, -.35559178704793E+01, -.46026609623249E+01,                       -.48570244420962E-03, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, -.18141441665425E+02, -.23481674322948E+02,
-.24779375900478E-02, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,     0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, -.61707676926363E+01, -.79872349702622E+01,                       -.84286450366167E-03, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, -.14702257248428E+02, -.19030109232044E+02,
-.20081797558495E-02, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,     0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, -.45854814644597E+02, -.59352935862628E+02,                       -.62633042614839E-02, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, -.55086333691708E+02, -.71301904845808E+02,
-.75242364675319E-02, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,     0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, -.37479497234615E+02, -.48512205594353E+02,                       -.51193205461031E-02, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, -.17390945319697E+02, -.22510256995927E+02,
-.23754273739045E-02, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,     0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, -.62302442017924E+01, -.80642193711508E+01,                       -.85098839372955E-03, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, -.11474640612304E+02, -.14852390388825E+02,
-.15673199423674E-02, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,     0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0,
OICOUNT  = 0,
O$END
```

APPENDIX E

PCNMNIO,TCNMNIO,CNMNIO
(LINKE)

```
1$LINKE
0OBJ    = .65191554849502E+04,
0G      = -.99612540953328E+00, -.94935039139788E+00, -.91326307998275E+00,
          -.87930132866256E+00, -.94943039392475E+00, -.94116590506151E+00,
          -.71020040190442E+00, -.89261372792225E+00, -.97487936989847E+00,
          -.98692171568133E+00, -.99870395183731E+00, -.98738794677924E+00,
          -.87958719907644E+00, -.91531243530964E+00, -.78631650827427E+00,
          -.93397310233129E+00, -.77047471646326E+00, -.86407397671257E+00,
          -.99403678281254E+00, -.99996910131648E+00, -.96744642732971E+00,
          -.91200828420965E+00, -.96650889933874E+00, -.99483668345919E+00,
          -.94940311795159E+00, -.99989399485171E+00, -.89573828310831E+00,
          -.99018839937236E+00, -.98376330035505E+00, -.99045073900280E+00,
          -.82434636287098E+00, -.9120574721001E+00, -.98214871107437E+00,
          -.86756342073507E+00, -.95093275316163E+00, -.99832006306207E+00,
          -.1E+01, -.1E+01, -.1E+01, -.1E+01, -.1E+01, -.1E+01, -.1E+01,
          -.1E+01, -.74909438806767E+00, -.72954813618485E+00,
          -.90282052842991E+00, -.62905417515892E+00, -.66527827273449E+00,
          -.90770470027496E+00, -.89395654824174E+00, -.8793369517628E+00,
          -.8901757062883E+00, -.8976449385336E+00, -.46854238283735E+00,
          -.24711377863709E+00, -.39518502256687E+00, -.13919476158796E+00,
          -.67468261637019E+00, -.77234036094163E+00, -.76618221228883E+00,
          -.84766656409377E+00, -.63793231317853E+00, -.46193976913077E+00,
          -.24502352169532E+00, -.23959949195004E+00, -.40988970873151E+00,
          -.6526814201401E+00, -.72292659609883E+00, -.67389465554642E+00,
          -.8011859435867E+00, -.83853632642516E+00, -.75776726442792E+00,
          -.83172622535049E+00, -.74143756616352E+00, -.77867218617681E+00,
          -.83176825974654E+00, -.81898052850954E+00, -.84961350322596E+00,
          -.76297373173829E+00, -.98736357652989E+00, -.98282139695363E+00,
          -.99487442911615E+00, -.9664135369308E+00, -.97234897927056E+00,
          -.98590798888861E+00, -.97796488204858E+00, -.95647748754038E+00,
          -.97738100702009E+00, -.98893806384817E+00, -.98188428119686E+00,
          -.97535305052494E+00, -.97532192221135E+00, -.97649886243115E+00,
          -.99744886203042E+00, -.9937573797467E+00, -.99260175650848E+00,
          -.97719796730478E+00, -.97761026985015E+00, -.98304752633251E+00,
          -.97698605918478E+00, -.94411065994063E+00, -.92072310217804E+00,
          -.99124622805876E+00, -.98435713337221E+00, -.95769095630818E+00,
          -.98276211395677E+00, -.9179431990493E+00, -.99407172721408E+00,
          -.98806697674274E+00, -.95800680012779E+00, -.9499041619961E+00,
          -.9857114846051E+00, -.9872529852147E+00, -.999E+00,
          -.9995E+00, -.9995E+00, -.99993087506942E+00, -.99683580852458E+00,
          -.96900739553173E+00, -.998E+00, -.98643172913928E+00,
          -.9895E+00, -.99396611076974E+00, -.9920672516409E+00,
          -.90987779519357E+00, -.90987779519357E+00, -.90987779519357E+00,
          -.9995E+00, -.9995E+00, -.98297436387062E+00, -.99030969623985E+00,
          -.98287264686675E+00, -.995E+00, -.97707504851133E+00,
          -.9865E+00, -.99070813374448E+00, -.98518823156656E+00,
          -.8499385102383E+00, -.8499385102383E+00, -.8499385102383E+00,
          -.98067631297244E+00, -.9995E+00, -.9995E+00, -.99310698754 91E+00,
          -.94213492554406E+00, -.94213492554406E+00, -.992E+00, -.991E+00,
          -.97881283586869E+00, -.94723339796774E+00, -.94723339796774E+00,
          -.96900739553173E+00, -.98643172913928E+00, -.98643172913928E+00,
          -.9665706839099E+00, -.9665706839099E+00, -.90987779519357E+00,
          -.8936828031804E+00, -.8936828031804E+00, -.8936828031804E+00,
          -.98699999999999E+00, -.94472481566735E+00, -.95848355393768E+00,
          -.97572805951452E+00, -.97260602241225E+00, -.94404672883729E+00,
          -.8499385102383E+00, -.8499385102383E+00, -.82438284204789E+00,
          -.88200180719071E+00, -.88200180719071E+00, -.88200180719071E+00,
          -.94213492554406E+00, -.96271864755008E+00, -.96271864755008E+00,
          -.9665E+00, -.94444916156149E+00, -.94444916156149E+00,
          -.96137907098406E+00, -.88484191976059E+00, -.88484191976059E+00,
          -.88034284141698E+00, -.88034284141698E+00, -.91538255235338E+00,
          -.94629421083819E+00, -.94629421083819E+00, -.94629421083819E+00,
          -.96299999999999E+00, -.97013328686123E+00, -.98541888584109E+00,
          -.962E+00, -.92039912916808E+00, -.90807469844997E+00,
          -.93945745009628E+00, -.84215731744931E+00, -.81565767492049E+00,
          -.99669101193845E+00, -.89938393696357E+00, -.73790229779899E+00,
          -.89432901557804E+00, -.93156502723075E+00, -.91020188142891E+00,
          -.80037304446194E+00, -.79262896248144E+00, -.85281515040045E+00,
          -.98629331802906E+00, -.98288362915108E+00, -.91017033386876E+00,
          -.78200267755801E+00, -.81924111775387E+00, -.89637741792666E+00,
          -.94558078766161E+00, -.64689703335836E+00, -.47707932595215E+00,
          -.98927365439634E+00, -.89151667521111E+00, -.71708545712041E+00,
          -.89426451038258E+00, -.46775877720043E+00, -.94372557709015E+00,
          -.8940282512407E+00, -.7058197755502E+00, -.6499277366643E+00,
          -.97755141582649E+00, -.93867167473174E+00, 0.0, 0.0, 0.0, 0.0, 0.0,
          0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
          -.95356513252776E+00, -.84214423402685E+00,
          -.99979629148632E+00, -.96075886305849E+00,
          -.99720836488841E+00, -.91485150041137E+00,
          -.82433643186822E+00, -.7249595045366E+00,
          -.81520651288817E+00, -.95893891032271E+00,
          -.95393604621982E+00, -.99161167681861E+00,
          -.97156065370306E+00, -.84098757771036E+00,
          -.83906068026602E+00, -.90549132514388E+00,
          -.93087470450384E+00, -.96557861963069E+00,
          -.98925733799774E+00, -.98097479877064E+00,
          -.99166439025051E+00, -.82191679433026E+00,
          -.1E+01, -.1E+01, -.1E+01, -.1E+01,
          -.1E+01, -.1E+01, -.1E+01, -.1E+01,
          -.8555794437228E+00, -.89572289200123E+00,
          -.68484870563006E+00, -.84153935056572E+00,
          -.90437128673897E+00, -.88433350905692E+00,
          -.14518157664209E+00, -.12502803443248E+00,
          -.15104666686232E+00, -.67967170154829E+00,
          -.76155441953953E+00, -.82912833439247E+00,
          -.21266704426779E-02, -.3016749995367E+00,
          -.41244546781569E+00, -.44396189413034E+00,
          -.65338441640601E+00, -.6791717013882E+00,
          -.37165588282875E-02, -.82038559977319E+00,
          -.87732442402454E+00, -.8484942996734E+00,
          -.83076303087437E+00, -.7531514893643E+00,
          -.98638214625238E+00, -.99681860151884E+00,
          -.96756705162441E+00, -.98432872881608E+00,
          -.95040587164743E+00, -.96732995861147E+00,
          -.98201669039594E+00, -.98228870367083E+00,
          -.99094469781908E+00, -.98984434804296E+00,
          -.9550939102102E+00, -.95071340162716E+00,
          -.98880354578167E+00, -.95226436327479E+00,
          -.9801032152709E+00, -.98327246641335E+00,
          -.94949777189421E+00, -.96801583592123E+00,
          -.98259498667683E+00, -.9939622954354E+00,
          -.96822612862022E+00, -.98356914269604E+00,
          -.98629380414674E+00, -.97824701591081E+00,
          -.9985E+00, -.9995E+00, -.96900739553173E+00,
          -.98770719158992E+00, -.9870694603646E+00,
          -.99301668120532E+00, -.98849999999999E+00,
          -.996E+00, -.976080108363E+00, -.99526263701584E+00,
          -.9955E+00, -.9995E+00, -.98287264686675E+00,
          -.97572805951452E+00, -.99932650550159E+00,
          -.98794818265552E+00, -.98549999999999E+00,
          -.99299999999999E+00, -.95946213788356E+00,
          -.99688035588484E+00, -.9925E+00, -.9995E+00,
          -.95501464317643E+00, -.95501464317643E+00,
          -.98999999999999E+00, -.96900739553173E+00,
          -.9870694603646E+00, -.9665706839099E+00,
          -.90987779519357E+00, -.90987779519357E+00,
          -.92864353373375E+00, -.92864353373375E+00,
          -.97707504851133E+00, -.97572805951452E+00,
          -.94404672883729E+00, -.8499385102383E+00,
          -.82438284204789E+00, -.82438284204789E+00,
          -.94213492554406E+00, -.94213492554406E+00,
          -.967E+00, -.94879761792019E+00, -.94879761792019E+00,
          -.96599999999999E+00, -.96137907098406E+00,
          -.88484191976059E+00, -.88034284141698E+00,
          -.91538255235338E+00, -.91538255235338E+00,
          -.97037324267011E+00, -.97037324267011E+00,
          -.9625E+00, -.97941385648939E+00, -.97941385648939E+00,
          -.94593232206489E+00, -.98530249587403E+00,
          -.87325004674732E+00, -.95296421184628E+00,
          -.69466145746591E+00, -.79542273026274E+00,
          -.90340609329802E+00, -.93101150259791E+00,
          -.90664199692837E+00, -.95088621799345E+00,
          -.69178532176906E+00, -.65400161038015E+00,
          -.98740163198104E+00, -.69409373565937E+00,
          -.96080428791377E+00, -.92969494436063E+00,
          -.66962572402796E+00, -.77867190326332E+00,
          -.89993733344525E+00, -.96618125361199E+00,
          -.75386996271018E+00, -.88903804907803E+00,
          0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0$END
```

(LINKF)

```
1$LINKF
0NDV    = 3,
0X      = .93832401644896E-01, .64498804411676E+00, .80270604593599E+01, 0.0,
          0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
          0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0$END
```

APPENDIX E

BLOCK,BLK

```
1$BLK
OOBJ     = .65191554849502E+04,
OOBJI    = .65191554849502E+04,
OX       = .93832401644896E-01, .64498804411676E+00, .80270604593599E+01, 0.0,   0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
           0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
OXI      = .93832401644896E-01, .64498804411676E+00, .80270604593599E+01, 0.0,   0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
           0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
OG       = -.99612540953328E+00, -.94935039139788E+00, -.91326307998275E+00,   -.95356513252776E+00, -.84214423402685E+00,
           -.87930132866256E+00, -.94943039392475E+00, -.94116590506151E+00,   -.99979629148632E+00, -.96075886305849E+00,
           -.71020040190442E+00, -.89261372792225E+00, -.97487936989847E+00,   -.9972083648841E+00, -.91485150041137E+00,
           -.98692171568133E+00, -.99870395183731E+00, -.98738794677924E+00,   -.82433643186822E+00, -.7249595045366E+00,
           -.87958719907644E+00, -.91531243530964E+00, -.78631650827427E+00,   -.81520651288817E+00, -.95893891032271E+00,
           -.93397310233129E+00, -.77047471646326E+00, -.86407397671257E+00,   -.95393604621982E+00, -.99161167681861E+00,
           -.99403678281254E+00, -.99996910131648E+00, -.96744642732971E+00,   -.97156065370306E+00, -.84098757771036E+00,
           -.91200828420965E+00, -.96650889933874E+00, -.99483668345919E+00,   -.83906068026602E+00, -.90549132514388E+00,
           -.94940311795159E+00, -.99989399485171E+00, -.89573828310831E+00,   -.93087470450384E+00, -.96557861963069E+00,
           -.99018839937236E+00, -.98376330035505E+00, -.9904507390028E+00,   -.98925733799774E+00, -.98097479877064E+00,
           -.82434636287098E+00, -.9120574721001E+00, -.98214871107437E+00,   -.99166439025051E+00, -.82191679433026E+00,
           -.86756342073507E+00, -.95093275316163E+00, -.99832006306207E+00,   -.1E+01, -.1E+01, -.1E+01, -.1E+01,
           -.1E+01, -.1E+01, -.1E+01, -.1E+01, -.1E+01, -.1E+01, -.1E+01,   -.1E+01, -.1E+01, -.1E+01, -.1E+01,
           -.1E+01, -.74909438806767E+00, -.72954813618485E+00,   -.8555794437228E+00, -.89572289200123E+00,
           -.90282052842991E+00, -.62905417515892E+00, -.66527827273449E+00,   -.68484870563006E+00, -.84153935056572E+00,
           -.90770470027496E+00, -.89395654824174E+00, -.8793369517628E+00,   -.90437128673897E+00, -.88433350905692E+00,
           -.8901757062883E+00, -.8976449385336E+00, -.46854238283735E+00,   -.14518157664209E+00, -.12502803443248E+00,
           -.24711377863709E+00, -.39518502256687E+00, -.13919476158796E+00,   -.15104666686232E+00, -.67967170154829E+00,
           -.67468261637019E+00, -.77234036094163E+00, -.76618221228883E+00,   -.76155441953953E+00, -.82912833439247E+00,
           -.84766656409377E+00, -.63793231317853E+00, -.46193976913077E+00,   -.21266704426779E-02, -.3016749995367E+00,
           -.24502352169532E+00, -.23959949195004E+00, -.40988970873151E+00,   -.41244546781569E+00, -.44396189413034E+00,
           -.6526814201401E+00, -.72292659609883E+00, -.67389465554642E+00,   -.65338441640601E+00, -.6791717013882E+00,
           -.8011859435867E+00, -.83853632642516E+00, -.75776726442792E+00,   -.37165588282875E-02, -.82038559977319E+00,
           -.83172622535049E+00, -.74143756616352E+00, -.77867218617681E+00,   -.87732442402454E+00, -.8484942996734E+00,
           -.83176825974654E+00, -.81898052850954E+00, -.84961350322596E+00,   -.83076303087437E+00, -.7531514893643E+00,
           -.76297373173829E+00, -.98736357652989E+00, -.98282139695363E+00,   -.98638214625238E+00, -.99681860151884E+00,
           -.99487442911615E+00, -.9664135369308E+00, -.97234897927056E+00,   -.96756705162441E+00, -.98432872881608E+00,
           -.9859079888861E+00, -.97796488204858E+00, -.95647748754038E+00,   -.95040587164743E+00, -.96732995861147E+00,
           -.97738100702009E+00, -.98893806384817E+00, -.98188428119686E+00,   -.98201669039594E+00, -.98228870367083E+00,
           -.97535305052494E+00, -.97532192221135E+00, -.97649886243115E+00,   -.99094469781908E+00, -.98984434804296E+00,
           -.99744886203042E+00, -.9937573797467E+00, -.99260175650848E+00,   -.9550939102102E+00, -.95071340162716E+00,
           -.97719796730478E+00, -.97761026985015E+00, -.98304752633251E+00,   -.98880354578167E+00, -.95226436327479E+00,
           -.97698605918478E+00, -.94411065994063E+00, -.92072310217804E+00,   -.9801032152709E+00, -.98327246641335E+00,
           -.99124622805876E+00, -.98435713337221E+00, -.95769095630818E+00,   -.94949777189421E+00, -.96801583592123E+00,
           -.98276211395677E+00, -.9179431990493E+00, -.99407172721408E+00,   -.98259498667683E+00, -.9939622954354E+00,
           -.98806697674274E+00, -.95800680012779E+00, -.9499041619961E+00,   -.96822612862022E+00, -.98356914269604E+00,
           -.9857114846051E+00, -.9872529852147E+00, -.999E+00,   -.98629380414674E+00, -.97824701591081E+00,
           -.9995E+00, -.9995E+00, -.99993087506942E+00, -.99683580852458E+00,   -.9985E+00, -.9995E+00, -.96900739553173E+00,
           -.96900739553173E+00, -.998E+00, -.98643172913928E+00,   -.98770719158992E+00, -.9870694603646E+00,
           -.9895E+00, -.99396611076974E+00, -.9920672516409E+00,   -.99301668120532E+00, -.98849999999999E+00,
           -.90987779519357E+00, -.90987779519357E+00, -.90987779519357E+00,   -.996E+00, -.976080108363E+00, -.99526263701584E+00,
           -.9995E+00, -.9995E+00, -.98297436387062E+00, -.99030969623985E+00,   -.9955E+00, -.9995E+00, -.98287264686675E+00,
           -.98287264686675E+00, -.995E+00, -.97707504851133E+00,   -.97572805951452E+00, -.99932650550159E+00,
           -.9865E+00, -.99070813374448E+00, -.98518823156656E+00,   -.98794818265552E+00, -.98549999999999E+00,
           -.8499385102383E+00, -.8499385102383E+00, -.8499385102383E+00,   -.99299999999999E+00, -.95946213788356E+00,
           -.98067631297244E+00, -.9995E+00, -.9995E+00, -.9931069875491E+00,   -.99688035588484E+00, -.9925E+00, -.9995E+00,
           -.94213492554406E+00, -.94213492554406E+00, -.992E+00, -.991E+00,   -.95501464317643E+00, -.95501464317643E+00,
           -.97881283586869E+00, -.94723339796774E+00, -.94723339796774E+00,   -.98999999999999E+00, -.96900739553173E+00,
           -.96900739553173E+00, -.98643172913928E+00, -.98643172913928E+00,   -.9870694603646E+00, -.9665706839099E+00,
           -.9665706839099E+00, -.9665706839099E+00, -.90987779519357E+00,   -.90987779519357E+00, -.90987779519357E+00,
           -.8936828031804E+00, -.8936828031804E+00, -.8936828031804E+00,   -.9286435337375E+00, -.9286435337375E+00,
           -.98699999999999E+00, -.94472481566735E+00, -.95848355393768E+00,   -.97707504851133E+00, -.97572805951452E+00,
           -.97572805951452E+00, -.97260602241225E+00, -.94404672883729E+00,   -.94404672883729E+00, -.8499385102383E+00,
           -.8499385102383E+00, -.8499385102383E+00, -.82438284204789E+00,   -.82438284204789E+00, -.82438284204789E+00,
           -.88200180719071E+00, -.88200180719071E+00, -.88200180719071E+00,   -.94213492554406E+00, -.94213492554406E+00,
           -.94213492554406E+00, -.96271864755008E+00, -.96271864755008E+00,   -.967E+00, -.94879761792019E+00, -.94879761792019E+00,
           -.9665E+00, -.94444916156149E+00, -.94444916156149E+00,   -.96599999999999E+00, -.96137907098406E+00,
           -.96137907098406E+00, -.88484191976059E+00, -.88484191976059E+00,   -.88484191976059E+00, -.88034284141698E+00,
           -.88034284141698E+00, -.88034284141698E+00, -.91538255235338E+00,   -.91538255235338E+00, -.91538255235338E+00,
           -.94629421083819E+00, -.94629421083819E+00, -.94629421083819E+00,   -.97037324267011E+00, -.97037324267011E+00,
           -.96299999999999E+00, -.97013328686123E+00, -.98541888584109E+00,   -.9625E+00, -.97941385648939E+00, -.97941385648939E+00,
           -.962E+00, -.92039912916808E+00, -.90807469844997E+00,   -.94593232206489E+00, -.98530249587403E+00,
           -.93945745009628E+00, -.84215731744931E+00, -.81565767492049E+00,   -.87325004674732E+00, -.95296421184628E+00,
           -.99669101193845E+00, -.89938393696357E+00, -.73790229779899E+00,   -.69466145746591E+00, -.79542273026274E+00,
           -.89432901557804E+00, -.93156502723075E+00, -.91020188142891E+00,   -.90340609329802E+00, -.93101150259791E+00,
           -.80037304446194E+00, -.79262896248144E+00, -.85281515040045E+00,   -.90664199692837E+00, -.95088621799345E+00,
           -.98629331802906E+00, -.98288362915108E+00, -.91017033386876E+00,   -.69178532176906E+00, -.65400161038015E+00,
           -.7820026755801E+00, -.81924111775387E+00, -.89637741792666E+00,   -.98740163198104E+00, -.69409373565937E+00,
           -.94558079766161E+00, -.64689703335836E+00, -.47707932595215E+00,   -.96080428791377E+00, -.92969494436063E+00,
           -.98927365439634E+00, -.89151667521111E+00, -.71708545712041E+00,   -.66962572402796E+00, -.77867190326332E+00,
           -.89426451038258E+00, -.4677587720043E+00, -.94372557709015E+00,   -.89993733344525E+00, -.96618125361199E+00,
           -.8940282512407E+00, -.7058197755502E+00, -.6499277366643E+00,   -.75386996271018E+00, -.88903804907803E+00,
           -.97755141582649E+00, -.93867167473174E+00, 0.0, 0.0, 0.0, 0.0, 0.0,   0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
           0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
OGI      = -.99612540953328E+00, -.94935039139788E+00, -.91326307998275E+00,   -.95356513252776E+00, -.84214423402685E+00,
           -.87930132866256E+00, -.94943039392475E+00, -.94116590506151E+00,   -.99979629148632E+00, -.96075886305849E+00,
           -.71020040190442E+00, -.89261372792225E+00, -.97487936989847E+00,   -.9972083648841E+00, -.91485150041137E+00,
           -.98692171568133E+00, -.99870395183731E+00, -.98738794677924E+00,   -.82433643186822E+00, -.7249595045366E+00,
           -.87958719907644E+00, -.91531243530964E+00, -.78631650827427E+00,   -.81520651288817E+00, -.95893891032271E+00,
           -.93397310233129E+00, -.77047471646326E+00, -.86407397671257E+00,   -.95393604621982E+00, -.99161167681861E+00,
           -.99403678281254E+00, -.99996910131648E+00, -.96744642732971E+00,   -.97156065370306E+00, -.84098757771036E+00,
           -.91200828420965E+00, -.96650889933874E+00, -.99483668345919E+00,   -.83906068026602E+00, -.90549132514388E+00,
           -.94940311795159E+00, -.99989399485171E+00, -.89573828310831E+00,   -.93087470450384E+00, -.96557861963069E+00,
           -.99018839937236E+00, -.98376330035505E+00, -.9904507390028E+00,   -.98925733799774E+00, -.98097479877064E+00,
           -.82434636287098E+00, -.9120574721001E+00, -.98214871107437E+00,   -.99166439025051E+00, -.82191679433026E+00,
           -.86756342073507E+00, -.95093275316163E+00, -.99832006306207E+00,   -.1E+01, -.1E+01, -.1E+01, -.1E+01,
           -.1E+01, -.1E+01, -.1E+01, -.1E+01, -.1E+01, -.1E+01, -.1E+01,   -.1E+01, -.1E+01, -.1E+01, -.1E+01,
           -.1E+01, -.74909438806767E+00, -.72954813618485E+00,   -.8555794437228E+00, -.89572289200123E+00,
           -.90282052842991E+00, -.62905417515892E+00, -.66527827273449E+00,   -.68484870563006E+00, -.84153935056572E+00,
           -.90770470027496E+00, -.89395654824174E+00, -.8793369517628E+00,   -.90437128673897E+00, -.88433350905692E+00,
           -.8901757062883E+00, -.8976449385336E+00, -.46854238283735E+00,   -.14518157664209E+00, -.12502803443248E+00,
```

BLOCK,BLK (Cont.)

```
          -.24711377863709E+00, -.39518502256687E+00, -.13919476158796E+00,        -.15104666686232E+00, -.67967170154829E+00,
          -.67468261637019E+00, -.77234036094163E+00, -.76618221228883E+00,        -.76155441953953E+00, -.82912833439247E+00,
          -.84766656409377E+00, -.63793231317853E+00, -.46193976913077E+00,        -.21266704426779E-02, -.3016749995367E+00,
          -.24502352169532E+00, -.23959949195004E+00, -.40988970873151E+00,        -.41244546781569E+00, -.44396189413034E+00,
          -.6526814201401E+00, -.72292659609883E+00, -.67389465554642E+00,         -.65338441640601E+00, -.6791717013882E+00,
          -.8011859435867E+00, -.83853632642516E+00, -.75776726442792E+00,         -.37165588282875E-02, -.82038559977319E+00,
          -.83172622535049E+00, -.74143756616352E+00, -.77867218617681E+00,        -.87732442402454E+00, -.8484942996734E+00,
          -.83176825974654E+00, -.81898052850954E+00, -.84961350322596E+00,        -.83076303087437E+00, -.7531514893643E+00,
          -.76297373173829E+00, -.98736357652989E+00, -.98282139695363E+00,        -.98638214625238E+00, -.99681860151884E+00,
          -.99487442911615E+00, -.9664135369308E+00, -.97234897927056E+00,         -.96756705162441E+00, -.98432872881608E+00,
          -.9859079888861E+00, -.97796488204858E+00, -.95647748754038E+00,         -.95040587164743E+00, -.96732995861147E+00,
          -.97738100702009E+00, -.98893806384817E+00, -.98188428119686E+00,        -.98201669039594E+00, -.98228870367083E+00,
          -.97535305052494E+00, -.97532192221135E+00, -.97649886243115E+00,        -.99094469781908E+00, -.98984434804296E+00,
          -.99744886203042E+00, -.9937573797467E+00, -.99260175650848E+00,         -.9550939102102E+00, -.95071340162716E+00,
          -.97719796730478E+00, -.97761026985015E+00, -.98304752633251E+00,        -.98880354578167E+00, -.95226436327479E+00,
          -.97698605918478E+00, -.94411065994063E+00, -.92072310217804E+00,        -.9801032152709E+00, -.98327246641335E+00,
          -.99124622805876E+00, -.98435713337221E+00, -.95769095630818E+00,        -.94949777189421E+00, -.96801583592123E+00,
          -.98276211395677E+00, -.9179431990493E+00, -.99407172721408E+00,         -.98259498667683E+00, -.9939622954354E+00,
          -.98806697674274E+00, -.95800680012779E+00, -.9499041619961E+00,         -.96822612862022E+00, -.98356914269604E+00,
          -.9857114846051E+00, -.9872529852147E+00, -.999E+00,                     -.98629380414674E+00, -.97824701591081E+00,
          -.9995E+00, -.9995E+00, -.99993087506942E+00, -.99683580852458E+00,      -.9985E+00, -.9995E+00, -.96900739553173E+00,
          -.96900739553173E+00, -.998E+00, -.98643172913928E+00,                   -.98770719158992E+00, -.9870694603646E+00,
          -.9895E+00, -.99396611076974E+00, -.9920672516409E+00,                   -.99301668120532E+00, -.98849999999999E+00,
          -.90987779519357E+00, -.90987779519357E+00, -.90987779519357E+00,        -.996E+00, -.976080108363E+00, -.99526263701584E+00,
          -.9995E+00, -.9995E+00, -.98297436387062E+00, -.99030969623985E+00,      -.9955E+00, -.9995E+00, -.98287264686675E+00,
          -.98287264686675E+00, -.995E+00, -.97707504851133E+00,                   -.97572805951452E+00, -.99932650550159E+00,
          -.9865E+00, -.99070813374448E+00, -.98518823156656E+00,                  -.98794818265552E+00, -.98549999999999E+00,
          -.8499385102383E+00, -.8499385102383E+00, -.8499385102383E+00,           -.99299999999999E+00, -.95946213788356E+00,
          -.98067631297244E+00, -.9995E+00, -.9995E+00, -.9931069875491E+00,       -.99688035588484E+00, -.9925E+00, -.9995E+00,
          -.94213492554406E+00, -.94213492554406E+00, -.992E+00, -.991E+00,        -.95501464317643E+00, -.95501464317643E+00,
          -.97881283586869E+00, -.94723339796774E+00, -.94723339796774E+00,        -.98999999999999E+00, -.96900739553173E+00,
          -.96900739553173E+00, -.98643172913928E+00, -.98643172913928E+00,        -.9870694603646E+00, -.9665706839099E+00,
          -.9665706839099E+00, -.9665706839099E+00, -.90987779519357E+00,          -.90987779519357E+00, -.90987779519357E+00,
          -.8936828031804E+00, -.8936828031804E+00, -.8936828031804E+00,           -.9286435337375E+00, -.9286435337375E+00,
          -.98699999999999E+00, -.94472481566735E+00, -.95848355393768E+00,        -.97707504851133E+00, -.97572805951452E+00,
          -.97572805951452E+00, -.97260602241225E+00, -.94404672883729E+00,        -.94404672883729E+00, -.8499385102383E+00,
          -.8499385102383E+00, -.8499385102383E+00, -.82438284204789E+00,          -.82438284204789E+00, -.82438284204789E+00,
          -.88200180719071E+00, -.88200180719071E+00, -.88200180719071E+00,        -.94213492554406E+00, -.94213492554406E+00,
          -.94213492554406E+00, -.96271864755008E+00, -.96271864755008E+00,        -.967E+00, -.94879761792019E+00, -.94879761792019E+00,
          -.9665E+00, -.94444916156149E+00, -.94444916156149E+00,                  -.96599999999999E+00, -.96137907098406E+00,
          -.96137907098406E+00, -.88484191976059E+00, -.88484191976059E+00,        -.88484191976059E+00, -.88034284141698E+00,
          -.88034284141698E+00, -.88034284141698E+00, -.91538255235338E+00,        -.91538255235338E+00, -.91538255235338E+00,
          -.94629421083819E+00, -.94629421083819E+00, -.94629421083819E+00,        -.97037324267011E+00, -.97037324267011E+00,
          -.96299999999999E+00, -.97013328686123E+00, -.98541888584109E+00,        -.9625E+00, -.97941385648939E+00, -.97941385648939E+00,
          -.962E+00, -.92039912916808E+00, -.90807469844997E+00,                   -.94593232206489E+00, -.98530249587403E+00,
          -.93945745009628E+00, -.84215731744931E+00, -.81565767492049E+00,        -.87325004674732E+00, -.95296421184628E+00,
          -.99669101193845E+00, -.89938393696357E+00, -.73790229779899E+00,        -.69466145746591E+00, -.79542273026274E+00,
          -.89432901557804E+00, -.93156502723075E+00, -.91020188142891E+00,        -.90340609329802E+00, -.93101150259791E+00,
          -.80037304446194E+00, -.79262896248144E+00, -.85281515040045E+00,        -.90664199692837E+00, -.95088621799345E+00,
          -.98629331802906E+00, -.98288362915108E+00, -.91017033386876E+00,        -.69178532176906E+00, -.65400161038015E+00,
          -.7820026755801E+00, -.81924111775387E+00, -.89637741792666E+00,         -.98740163198104E+00, -.69409373565937E+00,
          -.94558079766161E+00, -.64689703335836E+00, -.47707932595215E+00,        -.96080428791377E+00, -.92969494436063E+00,
          -.98927365439634E+00, -.89151667521111E+00, -.71708545712041E+00,        -.66962572402796E+00, -.77867190326332E+00,
          -.89426451038258E+00, -.4677587720043E+00, -.94372557709015E+00,         -.89993733344525E+00, -.96618125361199E+00,
          -.8940282512407E+00, -.7058197755502E+00, -.6499277366643E+00,           -.75386996271018E+00, -.88903804907803E+00,
          -.97755141582649E+00, -.93867167473174E+00, 0.0, 0.0, 0.0, 0.0, 0.0,     0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
          0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
OGRDOBJ = -.36890144472753E+04, -.86221934504831E+04, -.75960654402034E+02,        0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
          0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
OGRDG   = .78926539876721E-01, .28390412395783E-01, -.15738492110516E-02, 0.0,     0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
          0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, .12269690290882E+00,        .98887082417648E-01, .88188998650221E-03,
          0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,    0.0, 0.0, 0.0, .36426593146111E+00,
          .53583318742154E-01, .44350914790497E-02, 0.0, 0.0, 0.0, 0.0, 0.0,       0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
          0.0, 0.0, 0.0, 0.0, 0.0, .18892946185546E+00, .39202536405585E-02,       .33431401003272E-02, 0.0, 0.0, 0.0,
          0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,    .24675571897595E+00, .48197870716377E+00,
          -.25729274305052E-02, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,       0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
          0.0, .32027204666577E+00, .31834913939118E+00, -.14914748289047E-02,     0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
          0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,                        .19982463503243E+00, .72870399802941E-01,
          .10632379912837E-02, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,        0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
          0.0, .10135902593217E+00, .14518169559407E+00, .23271213969678E-03,      0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
          0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,                        -.24723245869879E-01, .59514936390792E-02,
          .69133298906272E-04, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,        0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
          0.0, -.25071053588403E-01, .91788919665193E-01, .14739948851079E-02,     0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
          0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, .1403336045341E+01,    .39633038016161E+00, .3761926439405E-02,
          0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,    0.0, 0.0, 0.0, .22284188334738E+00,
          .21934662630485E+00, .21005198924919E-02, 0.0, 0.0, 0.0, 0.0, 0.0,       0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
          0.0, 0.0, 0.0, 0.0, 0.0, .46512145522568E-01, .10401924624308E+00,       -.15761294712521E-02, 0.0, 0.0, 0.0,
          0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,    -.84471682175374E-02, .46750154328815E-01,
          -.14177697982001E-02, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,       0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
          0.0, .24092252933934E+00, .11762798099008E+00, .30564028336268E-02,      0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
          0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,                        .61920854157778E-01, -.25737316426616E-01,
          .20789841216751E-02, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,        0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
          0.0, .34975339595409E-02, .83116370561983E-03, .76770715017189E-04,      0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
          0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,                        .69962149051024E-01, -.20067024870172E-01,
          .16858722210823E-02, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,        0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
          0.0, .2610252900614E+00, .39790206238286E+00, .28563869884409E-02,       0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
          0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,                        .13607110652603E+01, .40067600673199E+00,
          .2268912846587E-02, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,    0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
          .77744244178567E+00, .12699791713365E+00, .81895582369301E-03, 0.0,      0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
          0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, .23426608947063E+00,        .15508020142841E+00, .16927451035223E-02,
          0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,    0.0, 0.0, 0.0, .23724928530189E+00,
          .57563700973023E+00, .70165854250785E-03, 0.0, 0.0, 0.0, 0.0, 0.0,       0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
          0.0, 0.0, 0.0, 0.0, 0.0, .39522523299978E+00, .45524794280089E+00,       .10240302158338E-03, 0.0, 0.0, 0.0,
          0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,    .29908587458665E+00, .53745756804416E-01,
          -.52002771855775E-03, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,       0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
          0.0, .1815906244513E+00, .15560895536147E+00, -.14525848663396E-03,      0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
          0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,                        .54710621610612E+00, .39456502036822E+00,
          .6324136150149E-02, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,    0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
```

BLOCK,BLK (Cont.)

```
.345839237333585E+00, .19409645784753E+00, .50591371757303E-02, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, -.48223050522651E-01,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
-.10711244403795E+00, .35570739304558E-02, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, .18888125186885E+00, -.10902562610131E+00,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
.15465386865193E-02, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, .16600958605975E+00, .24644345166096E-01, .10950029903294E-02,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
.23860418715761E-03, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, .25929253061642E+00, .34379971332891E+00, .29522976362113E-02,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
.15144922584506E-02, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, -.63904428835811E-01, .14972824919785E+00, -.10813134438011E-02,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
.2477910323997E-02, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
.36568572180097E+00, .29659536421146E+00, .38410027583693E-02, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, .17104643334213E+00,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
.11295338239925E+00, .87288022082958E-03, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, .50028617961948E-01, -.25505615774562E-01,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
.46978745942851E-02, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, .20759141930337E+00, .61268584767827E-01, .36956592716755E-02,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
.23282078728027E-02, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, .2897376512494E-01, .38427372412322E-02, .71142811840551E-03,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
-.20639516360097E-02, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, .10101903724239E+00, -.59143558185581E-01, .23822068288878E-02,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
.21419493824203E-02, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, .91076229535821E-01, .41552156933788E-02, .10953353585502E-02,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
.42732092847206E-02, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, .13598001668768E+00, .15643024126114E+00, .30618561440707E-02,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
.26717977761713E-03, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, .9167569821944E-01, -.42032818194575E-01, .16813497356581E-02,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
.4174651136579E-02, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
.31176541883768E+00, .21174027446915E+00, .43278087489787E-02, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, .92105472033354E-01,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
-.13637062666039E-01, .67992481378305E-03, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
-.26786148143023E-02, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, .14061438783291E+00, .65758839542781E+00, -.2836609402581E-02,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
-.54434642228305E-03, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, .91626419879446E-01, .89912465928372E-01, .50915538822803E-02,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
.8029785791129E-02, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
.15882995464188E+00, .84016036737538E+00, -.82005182922512E-03, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, .19739519666738E-01,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
.7994067082097E+00, -.3136207582019E-02, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, .20945245858784E+00, .53190371472942E-01,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
.57827044036118E-02, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, .79912573177753E-01, .73609087796254E-01, .58688670373984E-02,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
.59035424093312E-02, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, .17067638597305E+00, .12723381101003E+00, .1752743744502E-02,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
.73907042994823E-02, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, .79483055029343E-01, .26346675835114E+00, -.13756614482521E-02,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
-.10020502673861E-02, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, .16588656614686E+00, .12213742083595E+01, -.93070987190833E-03,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
.5184555153218E-02, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
.70326494936094E-01, .18859713667252E+01, .56768719066748E-02, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, .22065123096564E+00,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
.14462691206813E+01, -.78688416559545E-03, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, .72754885600766E-01, .1846987862366E+01,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
.60969105958447E-02, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, .8432907786175E-01, .75874620507823E+00, -.12654379055407E-02,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
-.15906084002066E-02, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, -.42467297296758E-02, .31637225502197E+00, .86022886734123E-02,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
.20369540060941E+00, -.19202088449185E-02,
0.0, 0.0, 0.0, .15703038694912E+00,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
.29076039064656E-02, 0.0, 0.0, 0.0,
.72917568101705E-01, -.60277544445565E-01,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
.93845482170129E-01, .53958630036139E-01,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
.64397707344027E-01, .21604719346294E+00,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
.11888038743704E+00, -.81230006637633E-01,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
.18994673879294E+00, .21085716606909E-02,
0.0, 0.0, 0.0, .73962869964255E-01,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
.41435765447448E-03, 0.0, 0.0, 0.0,
.3094337536228E+00, .11591212903451E+00,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
.10265122513568E+00, .18479512823294E-01,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
-.12253611473572E-01, .10541197572862E+00,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
.86928734492241E-01, -.44934009928297E-01,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
.39531365319327F+00, .32291629631969E+00,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
-.60689312343349E-01, .66274447195374F-01,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
.34295123528193E+00, .3481518735965E+00,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
.48673548741028E-01, .30930071573164E-02,
0.0, 0.0, 0.0, -.24388169146443E-03,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
.12143106849978E+00, .61649584727708F+00,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
.89433412383154E-01, .32662945501621E+00,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
.16663624513147E+00, -.24323529494574F-01,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
.83028781463533F+00, -.23158733610362E-02,
0.0, 0.0, 0.0, .38030710288452E-01,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
.10059529771906E-01, 0.0, 0.0, 0.0,
.53177170775163E-01, .53906350158263E-01,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
.17493290392634E+00, .78804243846096E-01,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
.15737040141055E+00, .36670959194181E-01,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
.62198338106425E-01, .24284129583417E+00,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
.83457020169097E-01, .18352355510542E+01,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.Q, 0.0,
.14594984188102E+01, .92409794333991E-02,
0.0, 0.0, 0.0, -.53167122787094E-01,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
.58753891979396E-02, 0.0, 0.0, 0.0,
.77194473768355E-01, .18028496709041E+01,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
.70965922817646E-01, .77909843849226E+00,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
```

BLOCK,BLK (Cont.)

```
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
.14945399069611E-01, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, .2828541475263E+00, .231771223486E+00, .94711282983042E-02, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, .30884521094681E+00,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
-.25002233688704E+00, .20686966409076E-01, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, .37088417319344E+00, .23985951502979E+00,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
-.12949481078369E-02, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, .3131848498431E+00, .19774525587961E+01, .93644768450129E-02,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, .1032729022134E+00,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
.14425970059508E+01, .10972267927051E-01, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, .128536442236E+00, .15042818221301E+01,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
.3489446667605E-03, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
-.10085311525503E+00, .13509436222302E+01, .21032527540741E-02, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, .26583653577062E+00,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
.38368740544557E+00, .1373290766639E-01, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, .97664188729101E-01, .52092961015906E+00,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
.176604969227E-01, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
.24772181124072E+00, .30170031984762E+00, .17426987280658E-01, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, .30794136702124E+00,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
.22768625863E+00, .95642630984787E-02, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, -.21033346596439E-01, .12180313757275E+00,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
.16047337152651E-01, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, .30991304616567E+00, .19759377270757E+01, .93695683066404E-02,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
-.61711335809877E-02, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, .37057840201186E+00, .36381847996048E+00, -.32795998533899E-02,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
-.22051699150136E-02, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, .50447080591616E+00, .42178227917339E+00, -.22065435548166E-02,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
.12726413699367E-02, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, .10220073458039E+00, .73021775807188E-01, .99862926787257E-02,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
.10310978489117E-01, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, .22159881647118E+00, .14322219118764E+00, .84562350144488E-02,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
.38447427372408E-02, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, .26144675501775E+00, -.54470528293759E-01, .14875915395192E-01,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
.64958922835764E-02, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, .24608758735156E+00, .31864933465458E+00, .64280626942186E-02,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
.55734842643271E-03, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, .44679698622918E-03, .25896363093004E-01, .10890323311076E-02,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
.81654728597109E-03, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, .32855078186802E-02, .14240133774486E-01, -.21292520906353E-03,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
.44968046446735E-03, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, .13692622479839E-01, .42606345583374E-01, .23046188846264E-02,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
.28367997771736E-02, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, .36978064150658E-01, .54400174273616E-01, .14190452110609E-02,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
.333203252421E-03, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
-.66548652150093E-02, .44754983535092E-01, .37420109701178E-04, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, .32071447211026E-01,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
.31921959366144E-01, .39391987877141E-02, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, -.4540259080553E-02, .36814759018869E-01,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
.30102568188383E-02, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, .420732393986E-01, .33925713741553E-01, .96059509114236E-03,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
.58577465460713E-04, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, .35293209219343E-02, .21067688067768E-01, .13641852188609E-02,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
.14929202920649E-02, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, -.48116423838411E-02, .30100789028812E-01, .10477047172852E-02,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
.30902555524153E-02, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, .6277298191956E-02, -.57338025786589E-02, .32109631840998E-02,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
.22463204406527E-02, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, -.33035975388791E-02, -.72296562502574E-02, .14465523378887E-02,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
.74521671564677E-03, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, -.11515937162072E-01, .14793359654818E-01, -.15137049428975E-03,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
.25048498865467E-03, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, -.81572382453851E-02, -.9216109225098E-02, .13893080932866E-02,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
.47828147142831E-02, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, .17123428542131E-02, .18048179094193E-01, .53680216527008E-02,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
.3382210082441E-02, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
-.18170862835454E-03, -.50315336832825E-03, .28026343566681E-02, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, .57106343172232E-02,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
.35378576552811E-01, .6577002828683E-04, 0.0, 0.0, 0.0, 0.0, 0.0,
```

```
.17244588629705E+00, .12028391610359E+00,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
.16246605533156E+00, .57594698610244E-02,
0.0, 0.0, 0.0, .29518216800296E+00,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
.1976881686928E-01, 0.0, 0.0, 0.0,
.15756560496435E+00, .12440214890676E+01,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
.11190748279254E+01, .19345648461275E-01,
0.0, 0.0, 0.0, .16039393742102E+00,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
.9654277032022E-02, 0.0, 0.0, 0.0,
-.14076593423092E+00, .14203331764872E+01,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
.85129335565062E+00, .14708693178389E-01,
0.0, 0.0, 0.0, .27756052645958E+00,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
.39360657344484E-02, 0.0, 0.0, 0.0,
.23855206089964E+00, .25082021927259E+00,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
.17085002691958E+00, .1994450881686E-01,
0.0, 0.0, 0.0, .40551597731269E-01,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
.10541139820356E-01, 0.0, 0.0, 0.0,
.37378540439128E+00, .44258567196151E-01,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
.69558790568419E-01, .54945387249829E+00,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
.59500812528526E+00, .48298847854761E+00,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
-.12128953499997E-01, .25888461258018E+00,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
.12262784879487E+00, .96722811426976E-01,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
.24773554864409E+00, .1804966697371E+00,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
.26301989177529E+00, .33488204478325E+00,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
.23077816081843E-01, .18527253228607E-01,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
.11099979523464E-02, .21523546873199E-01,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
.15669672675302E-01, .10512504232278E-03,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
-.1849533892574E-01, .20216181467688E-01,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
.33040161789987E-01, .30661845030233E-01,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
.32309520897019E-01, .10678549362208E-02,
0.0, 0.0, 0.0, .16008590469009E-01,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
.47377257815548E-02, 0.0, 0.0, 0.0,
-.10495427354865E-01, .29216735820522E-01,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
.10180159480441E-01, .29882105531271E-01,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
-.10285493510057E-01, .21625311098429E-01,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
.33495632824748E-02, -.19880815204399E-02,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
-.41792385127946E-02, .17986053306238E-01,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
.52759917474498E-02, .11304566124855E-01,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
-.42044024996076E-02, .14318900228102E-01,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
.23729477388694E-02, .18914530964705E-01,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
-.11201594172393E-01, -.1032675051565E-01,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
.10833157404599E-01, .16041372025725E-02,
0.0, 0.0, 0.0, -.7832443125071E-02,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
```

BLOCK,BLK (Cont.)

```
0.0, 0.0, 0.0, 0.0, 0.0, -.42889480752246E-02, .33179337435938E-01,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
.84358469975271E-03, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, .12421898464328E-01, .36491183519891E-01, .53310549039986E-02,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
.81041618024183F-02, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, .44380734387683E-01, .43238334945249E-01, .22104893188436E-03,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
.10784152389429E-02, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, -.72367846574043E-02, .26048061577378E-01, .1289876730604E-03,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
.13312091002014E-02, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, .1032784639081E-01, .19853511912262E-01, .43364976853738E-02,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
.51260107451377E-02, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, -.22680722978243E-01, .20825823855078E-01, .34022563031437E-02,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
.13407325555864E-02, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, -.22083584442151E-01, .54815943938525E-01, .82502952776033E-02,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
-.86901876235455E-03, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, -.15315864039782E-01, .21512988875294E-01, .14783272085817E-02,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
.5063940767865E-03, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
-.78683242837697E-02, .23142199586804E-02, .14800338692614E-02, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, .64353330045557E-02,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
.19280649489573E-01, .54329756035217E-02, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, -.19381774684125E-01, .89252804731508E-02,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
.1706440924499E-02, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
.87422291977067E-02, .33363738291085E-01, .33520887508938E-03, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, .84463463152382E-02,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
-.25960940794902E+00, -.15519829001364E-04, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, -.15567227376803E+01, -.35582573906976E+01,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
-.33760259333354E-03, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, -.5678901550609E-01, -.12980470397451E+00, -.77599145006819E-05,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
-.77599145006819E-05, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, -.78510735087398E-02, -.17945482301499E-01,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
-.4910771062605E-04, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, -.17036704651827E+00, -.38941411192354E+00,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
-.77599145006819E-05, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, -.35200789914449E+01, -.80459716968046E+01,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
-.48099992165438E-03, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, -.22715606202436E+00, -.51921881589805E+00,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
-.21057724360242E-03, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, -.13961929748266E+01, -.3191328713371E+01, -.19078228447093E-03,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
-.20067976403667E-03, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, -.11925693256279E+01, -.27258987834647E+01,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
-.93644929066736E-04, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, -.900985939142E+00, -.20594161049153E+01, -.12311489804412E-03,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
-.10837991355543E-03, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, -.13061473566401E+01, -.29855081914138E+01,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
-.13986812078216E-02, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, -.10235902572391E+02, -.23396572232858E+02,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
-.13986812078216E-02, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, -.45431212404872E+00, -.10384376317961E+01,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
-.37123262793735E-03, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, -.53806035993009E+00, -.12298639995556E+01,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
-.77599145006819E-05, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, -.5678901550609E-01, -.12980470397451E+00, -.77599145006819E-05,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
-.26423496136727E-03, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, -.11006056209864E+01, -.25156940220157E+01,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
-.69839230506137E-04, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, -.5678901550609E-01, -.12980470397451E+00, -.77599145006819F-05,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
-.26581359187396E-03, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, -.19452910453244E+01, -.4446422006656F+01, -.26581359187396E-03,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
-.77599145006818E-04, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, -.26037708511326E+01, -.59515330832331E+01,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
-.37669636586596E-03, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, -.76494179026625E-01, -.17484550798842E+00,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
-.20951769151841E-03, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, -.10553518737301E+01, -.24122558973361E+01,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
-.22987611329471E-03, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, -.1368821684104E+01, -.31287651655091E+01, -.18704214434193F-03,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
-.22503752051977E-03, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
```

```
.46475057237199E-02, 0.0, 0.0, 0.0,
-.20113690442286E-02, .50822637478472E-01,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
-.16753982784108E-01, .48307681490781E-01,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
-.59483003041077E-02, .26631393725611E-01,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
.44265242252536E-02, .13914152060607E-01,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
-.27528284007648E-03, .28673978388793E-01,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
.17898958792504E-01, .14772248782544E-01,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
-.89874982265769F-02, .42520322010657E-01,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
-.54331270132041E-02, .76275935443108F-02,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
.14625741707872E-01, .45506211944412E-02,
0.0, 0.0, 0.0, .12249169690353E-02,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
.38161596820777E-02, 0.0, 0.0, 0.0,
-.57328548161158E-02, .1005577119499E-01,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
.13899797783822E-01, .93211496017418E-03,
0.0, 0.0, 0.0, -.11357803101218E+00,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
-.21271781590176F-03, 0.0, 0.0, 0.0,
-.24706611014889E+01, -.56472793205179E+01,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
-.5678901550609E-01, -.12980470397451E+00,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
-.10728071022794E-05, 0.0, 0.0, 0.0,
-.35938263752343E+00, -.82145387557044E+00,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
-.23279743502046E-04, 0.0, 0.0, 0.0,
-.5678901550609E-01, -.12980470397451E+00,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
-.48099992165438E-03, 0.0, 0.0, 0.0,
-.35200789914449E+01, -.80459716968046E+01,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
-.31039658002727E-04, 0.0, 0.0, 0.0,
-.15410574885995E+01, -.35224507650412E+01,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
-.1468625231713E+01, -.33568897392061E+01,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
-.16295820451432E-03, 0.0, 0.0, 0.0,
-.68531725811803E+00, -.15664544106966F+01,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
-.79315159863001E+00, -.18129352578059E+01,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
-.17847803351568E-03, 0.0, 0.0, 0.0,
-.10235902572391E+02, -.23396572232858E+02,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
-.13986812078216E-02, 0.0, 0.0, 0.0,
-.10235902572391E+02, -.23396572232858E+02,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
-.62079316005455E-04, 0.0, 0.0, 0.0,
-.27167741941549E+01, -.62098289060857E+01,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
-.73523063431445E-04, 0.0, 0.0, 0.0,
-.5678901550609E-01, -.12980470397451E+00,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
-.19337382283037E+01, -.44200153155014E+01,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
-.15039185732862E-03, 0.0, 0.0, 0.0,
-.51110113955481E+00, -.11682423357706E+01,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
-.19452910453244F+01, -.4446422006656E+01,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
-.5678901550609E+00, -.12980470397451E+01,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
-.35579132696868E-03, 0.0, 0.0, 0.0,
-.27567592091859E+01, -.63012240992099E+01,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
-.10452519448639F-04, 0.0, 0.0, 0.0,
-.15333034186644E+01, -.35047270073118E+01,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
-.14420817538915E-03, 0.0, 0.0, 0.0,
-.1682291494478E+01, -.38452744336822F+01,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
-.16468814496766E+01, -.37643364152608E+01,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
```

BLOCK,BLK (Cont.)

```
0.0, -.17043688537888E+02, -.38957374512983E+02,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
-.23289286607915E-02, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, -.17043688537888E+02, -.38957374512983E+02,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
-.10863880300955E-03, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, -.46042105606629E+01, -.10524010383569E+02, -.62914068812807E-03,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
-.29990031834347E-03, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, -.56789015506609E-01, -.12980470397451E+00, -.77599145006819E-05,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
-.77599145006819E-05, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, -.78289478191508E+00, -.17894908813621E+01,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
-.48416343212275E-04, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, -.85183523259136E+00, -.19470705596177E+01,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
-.77599145006819E-05, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, -.65722012210782E+01, -.15022317720431E+02,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
-.89805606070727E-03, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, -.90862424809745E+00, -.20768752635922E+01,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
-.13967846101227E-03, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, -.51093482524006E+01, -.11678621851341E+02,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
-.69816504546702E-03, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, -.2406396384765E+01, -.55003871362457E+01, -.32882116434161E-03,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
-.81892864052374E-03, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, -.59931267620278E+01, -.13698706313078E+02,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
-.15519829001364E-03, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, -.35200789914449E+01, -.80459716968046E+01,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
-.48099992165438E-03, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, -.15410574885995E+01, -.35224507650412E+01,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
-.21057724360242E-03, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, -.1468625231713E+01, -.33568897392061E+01, -.20067976403667E-03,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
-.51881726935086E-03, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, -.37968358995972E+01, -.86785649582911E+01,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
-.51881726935086E-03, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, -.10235902572391E+02, -.23396572232858E+02,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
-.13986812078216E-02, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, -.10235902572391E+02, -.23396572232858E+02.
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
-.16500247145445E-02, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, -.12075297877505E+02, -.27600944521137E+02,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
-.16500247145445E-02, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, -.81045269380818E+01, -.18524809959742E+02,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
-.11074401545356E-02, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, -.14765144031583E+01, -.33749223033373E+01,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
-.85786140886159E-03, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, -.47153561983807E+01, -.10778059982384E+02,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
-.35579132696868E-03, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, -.27567592091859E+01, -.63012240992099E+01,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
-.37669636586596E-03, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, -.31113540360082E+01, -.71117343029 24E+01, -.42514984782903E-03,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
-.86838520051218E-03, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, -.63550623673511E+01, -.14525995599362E+02,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
-.23289286607915E-02, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0; -.17043688537888E+02, -.38957374512983E+02,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
-.23289286607915E-02, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, -.19946251012156E+02, -.45591866401638E+02,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
-.27255482611222E-02, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, -.19946251012156E+02, -.45591866401638E+02,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
-.18313117748701E-02, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, -.13402002402274E+02, -.30633440974273E+02,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
-.18313117748701E-02, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, -.65722012210782E+01, -.15022317720431E+02,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
-.89805606070727E-03, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, -.65722012210782E+01, -.15022317720431E+02,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
-.57860021496237E-03, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, -.42343426047334E+01, -.96785898370633E+01,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
-.512154357045E-03, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
-.5815465739758E+01, -.13292620097319E+02, -.79465221434112E-03, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, -.5815465739758E+01,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
-.86969151662923E+01, -.51991427154568E-03, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, -.63093548509207E+01, -.14421520278091E+02,
-.23289286607915E-02, 0.0, 0.0, 0.0,
-.17043688537888E+02, -.38957374512983E+02,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
-.23289286607915E-02, 0.0, 0.0, 0.0,
-.79504621708526E+00, -.18172658556432E+01,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
-.21947463244849E+01, -.50166109486148E+01,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
-.56789015506609E-01, -.12980470397451E+00,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
-.10697837454216E-03, 0.0, 0.0, 0.0,
-.35432303605774E+00, -.80988896174636E+00,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
-.11639871751023E-03, 0.0, 0.0, 0.0,
-.56789015506609E-01, -.12980470397451E+00,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
-.89805606070727E-03, 0.0, 0.0, 0.0,
-.65722012210782E+01, -.15022317720431E+02,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
-.12415863201091E-03, 0.0, 0.0, 0.0,
-.10222022791096E+01, -.23364846715412E+01,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
-.69816504546702E-03, 0.0, 0.0, 0.0,
-.51093482524006E+01, -.11678621851341E+02,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
-.59931267620278E+01, -.13698706313078E+02,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
-.81892864052374E-03, 0.0, 0.0, 0.0,
-.11357803101218E+01, -.25960940794902E+01,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
-.48099992165438E-03, 0.0, 0.0, 0.0,
-.35200789914449E+01, -.80459716968046E+01,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
-.21057724360242E-03, 0.0, 0.0, 0.0,
-.15410574885995E+01, -.35224507650412E+01,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
-.37968358995972E+01, -.86785649582911E+01,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
-.51881726935086E-03, 0.0, 0.0, 0.0,
-.37968358995972E+01, -.86785649582911E+01,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
-.13986812078216E-02, 0.0, 0.0, 0.0,
-.10235902572391E+02, -.23396572232858E+02,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
-.13986812078216E-02, 0.0, 0.0, 0.0.
-.12075297877505E+02, -.27600944521137E+02,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
-.16500247145445E-02, 0.0, 0.0, 0.0,
-.12075297877505E+02, -.27600944521137E+02,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
-.11074401545356E-02, 0.0, 0.0, 0.0,
-.81045269380818E+01, -.18524809959742E+02,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
-.20175777701773E-03, 0.0, 0.0, 0.0,
-.62780466003378E+01, -.14349957878872E+02,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
-.64432814363141E-03, 0.0, 0.0, 0.0,
-.26037708511326E+01, -.59515330832331E+01,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
-.37669636586596E-03, 0.0, 0.0, 0.0,
-.27567592091859E+01, -.63012240992099E+01,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
-.63550623673511E+01, -.14525995599362E+02,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
-.86838520051218E-03, 0.0, 0.0, 0.0,
-.17043688537888E+02, -.38957374512983E+02,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
-.23289286607915E-02, 0.0, 0.0, 0.0,
-.17043688537888E+02, -.38957374512983E+02,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
-.27255482611222E-02, 0.0, 0.0, 0.0,
-.19946251012156E+02, -.45591866401638E+02,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
-.27255482611222E-02, 0.0, 0.0, 0.0,
-.13402002402274E+02, -.30633440974273E+02,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
-.18313117748701E-02, 0.0, 0.0, 0.0,
-.13402002402274E+02, -.30633440974273E+02,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
-.89805606070727E-03, 0.0, 0.0, 0.0,
-.65722012210782E+01, -.15022317720431E+02,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
-.89805606070727E-03, 0.0, 0.0, 0.0,
-.42343426047334E+01, -.96785898370633E+01,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
-.57860021496237E-03, 0.0, 0.0, 0.0,
-.3748075023402E+01, -.85671104623178E+01,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
-.13292620097319E+02, -.79465221434112E-03,
0.0, 0.0, 0.0, -.38048640389081E+01,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
-.86213951344792E-03, 0.0, 0.0, 0.0,
```

BLOCK,BLK (Cont.)

```
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
-.86213951344792E-03, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, -.38616530544141E+01, -.88267198702667E+01,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
-.59939021420113E-03, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, -.43864890734912E+01, -.10026356516268E+02,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
-.1787233713441E-02, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, -.13079428008735E+02, -.29896121031499E+02, -.1787233713441E-02,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
-.1787233713441E-02, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, -.13590424468372E+02, -.310641240966E+02, -.18570586399975E-02,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
-.18570586399975E-02, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, -.13590424468372E+02, -.310641240966E+02, -.18570586399975E-02,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
-.13132483180074E-02, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, -.96106830929793E+01, -.21967485485696E+02,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
-.13132483180074E-02, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, -.6099797786953E+01, -.13942528127731E+02, -.83350466417449E-03,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
-.83350466417449E-03, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, -.6099797786953E+01, -.13942528127731E+02, -.83350466417449E-03,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
-.45980220762463E-03, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, -.33649487628034E+01, -.76913849298594E+01,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
-.57423367305046E-03, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, -.33922024711071E+01, -.77536797153394E+01,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
-.22629639839566E-03, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, -.42591761629568E+01, -.97353527980884E+01,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
-.31949342708223E-03, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, -.23381336460695E+01, -.53443565287435E+01,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
-.58975350205182E-03, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, -.90409101759437E+01, -.206651349489E+02, -.12353919036709E-02,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
-.14266649609551E-02, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, -.61409004012701E+01, -.14036477857912E+02,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
-.22810275078178E-03, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, -.68763036105201E+01, -.15717415536226E+02,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
-.24496914423032E-02, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, -.20937238314738E+02, -.47857001873838E+02,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
-.19671376004124E-02, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, -.53422322056049E+01, -.12210933115001E+02,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
-.51354928882787E-04, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, -.11427774327876E+02, -.2612087655505E+02, -.15615440931159E-02,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
-.40677115198101E-02, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, -.34679750453151E+02, -.79268758251293E+02,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
-.31750042438881E-02, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, -.12001902345765E+02, -.27433181703174E+02,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
-.10620990750918E-02, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, -.10199093495903E+02, -.23312436397177E+02,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
-.14991209145883E-02, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, -.78355776974178E+01, -.17910062965848E+02,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
-.30981762140136E-02, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, -.2355279413031E+02, -.53835472275966E+02, -.32183630421233E-02,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
-.22842836973763E-02, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, -.10603418168104E+02, -.24236615904728E+02,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
-.76223749835193E-03, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, -.1556777949969E+01, -.35583835914202E+01, -.212725360365E-03,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
-.26564314869906E-03, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, -.10202676605667E+02, -.23320626440587E+02,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
-.47834391018545E-02, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, -.39297815826408E+02, -.89824437080526E+02,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
-.33832811977516E-02, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, -.20530237933478E+02, -.46926706401445E+02,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
-.1608204754458E-02, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, -.14308978335601E+01, -.32706548625257E+01,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
-.47476129130125E-02, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, -.61808258508478E+01, -.14127736900127E+02,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
-.54800976621525E-02, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, -.59392300533917E+02, -.13575512659386E+03,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
-.60831074916481E-03, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, -.79851097897204E+01, -.18251853870357E+02,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
-.16647104957838E-03, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
-.63093548509207E+01, -.14421520278091E+02,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
-.52767418604636E-03, 0.0, 0.0, 0.0,
-.43864890734912E+01, -.10026356516268E+02,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
-.59939021420113E-03, 0.0, 0.0, 0.0,
-.13079428008735E+02, -.29896121031499E+02,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
-.13079428008735E+02, -.29896121031499E+02,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
-.13590424468372E+02, -.310641240966E+02,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
-.96106830929793E+01, -.21967485485696E+02,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
-.13132483180074E-02, 0.0, 0.0, 0.0,
-.96106830929793E+01, -.21967485485696E+02,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
-.6099797786953E+01, -.13942528127731E+02,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
-.33649487628034E+01, -.76913849298594E+01,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
-.45980220762463E-03, 0.0, 0.0, 0.0,
-.42023871474507E+01, -.96055480941139E+01,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
-.46352628074649E-03, 0.0, 0.0, 0.0,
-.16560942361329E+01, -.37853944140319E+01,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
-.58199358755114E-03, 0.0, 0.0, 0.0,
-.23381336460695E+01, -.53443565287435E+01,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
-.31949342708223E-03, 0.0, 0.0, 0.0,
-.43159651784628E+01, -.98651575020629E+01,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
-.10440694750252E+02, -.23864673110936E+02,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
-.83912111605366E-03, 0.0, 0.0, 0.0,
-.16693135794203E+01, -.38156103444696E+01,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
-.93961002181211E-03, 0.0, 0.0, 0.0,
-.17927461093788E+02, -.40977445366069E+02,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
-.28609613629478E-02, 0.0, 0.0, 0.0,
-.14396010121325E+02, -.32905480321494E+02,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
-.72998738909001E-03, 0.0, 0.0, 0.0,
-.37582834867349E+00, -.859044431569E+00,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
-.29768540948808E+02, -.68043029293204E+02,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
-.47388019676547E-02, 0.0, 0.0, 0.0,
-.23235483486606E+02, -.53110183876318E+02,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
-.16399956086345E-02, 0.0, 0.0, 0.0,
-.77727094595038E+01, -.17766362763633E+02,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
-.13936514448675E-02, 0.0, 0.0, 0.0,
-.10970945730984E+02, -.25076686930382E+02,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
-.10706896827414E-02, 0.0, 0.0, 0.0,
-.22673236546969E+02, -.51825035737902E+02,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
-.1671696541234E+02, -.38210571663603E+02,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
-.14489002435804E-02, 0.0, 0.0, 0.0,
-.55782466558655E+01, -.12750399868858E+02,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
-.19440436990939E+01, -.44435709023223E+01,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
-.13941410576063E-02, 0.0, 0.0, 0.0,
-.35006416282523E+02, -.80015430136732E+02,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
-.53698358416473E-02, 0.0, 0.0, 0.0,
-.24759706873535E+02, -.56594156327l2E+02,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
-.28053469429376E-02, 0.0, 0.0, 0.0,
-.11769248840288E+0?, -.26901397182209E+02,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
-.19552451735048E-03, 0.0, 0.0, 0.0,
-.347442311781E+02, -.79416144173368E+02,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
-.84457671468242E-03, 0.0, 0.0, 0.0,
-.40104739695717E+02, -.9166885211488E+02,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
-.81156394425004E-02, 0.0, 0.0, 0.0,
-.44517718028732E+01, -.10175575608844E+02,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
-.10911224414542E-02, 0.0, 0.0, 0.0,
-.12182772136202E+01, -.27846602316236E+01,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
```

BLOCK,BLK (Conc.)

```
      0.0, -.12321322427176E+02, -.28163291720783E+02,                          -.16836426502228E-02, 0.0, 0.0, 0.0,
      0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, -.32132876724974E+02, -.73447276977137E+02,
      -.43907853274902E-02, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,      0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
      0.0, -.37523259761979E+02, -.85768270186688E+02,                          -.51273522695354E-02, 0.0, 0.0, 0.0,
      0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, -.25138009435025E+02, -.57458856156291E+02,
      -.34349742145504E-02, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,      0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
      0.0, -.12009228718856E+02, -.27449927858779E+02,                          -.16409967182379E-02, 0.0, 0.0, 0.0,
      0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, -.60450910699258E+02, -.13817483008603E+03,
      -.82602928459693E-02, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,      0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
      0.0, -.63915381504472E+01, -.14609369614298E+02,                          -.87336942071125E-03, 0.0, 0.0, 0.0,
      0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, -.11364920645116E+02, -.25977209622077E+02,
      -.15529554743501E-02, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,      0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
      0.0, -.38410666260495E+01, -.87796647273672E+01,                          -.52486116098227E-03, 0.0, 0.0, 0.0,
      0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, -.12036062566999E+02, -.27511262954724E+02,
      -.16446634197198E-02, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,      0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
      0.0, -.33412410655729E+02, -.76371953899722E+02,                          -.45656267790436E-02, 0.0, 0.0, 0.0,
      0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, -.39760518381646E+02, -.90882053023955E+02,
      -.54330616650904E-02, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,      0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
      0.0, -.27954965008333E+02, -.63897673259282E+02,                          -.38198960908373E-02, 0.0, 0.0, 0.0,
      0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, -.12602839902987E+02, -.2880676638372E+02,
      -.17221105039666E-02, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,      0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
      0.0, -.25496659894377E+01, -.5827863646577E+01, -.34839818769549E-03, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
      0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,                          -.69655504292425E+01, -.15921410213396E+02,
      -.95180512110327E-03, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,      0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
      0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
      0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
      0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
      0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
      0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
      0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
      0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
      0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
      0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
      0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
      0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
      0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
      0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
      0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
      0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
      0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
      0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
      0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
      0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
      0.0, 0.0,
 OICOUNT  = 0,
 O$END
```

APPENDIX E

SPFPOUT

Rod data

```
E23 SECTION PROPERTIES
1   10.657
```

Cross-sectional area, cm^2 (inverse)

Beam data

```
 E21 SECTION PROPERTIES
DSY 1  40393.3008      0.0000    3035.1865      0.0000     167.4450%
     346.14588      0.00000      7.56862      0.00000      0.00000
       0.0000       0.0000       0.0000     -10.9297     21.1677      4.0122%
      21.1677       4.0122     -21.1677     -10.9297    -21.1677
```

Membrane data

```
SHELL SECTION PROPERTIES
1     .125
```

Thickness, cm (inverse)

(See ref. 5 for more details.)

APPENDIX E

RSOUT (nonrepeatable part)

```
[XQT TAB
 UPDATE=1
 BC
 2 1.0
 BA
 DSY 2 1.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
 DSY 3 0.0 0.0 1.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
 DSY 4 0.0 0.0 0.0 0.0 1.0 0.0 0.0 0.0 0.0 0.0
 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
 DSY 5 0.0 0.0 0.0 0.0 0.0 1.0 0.0 0.0 0.0 0.0
 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
 SA
 2 1.0
 UPDATE=0
[XQT DCU
 COPY 1,2
 DISABLE 1,DEF E23
 DISABLE 1,DEF E21
 DISABLE 1,DEF E41
 DISABLE 1,GD E23
 DISABLE 1,GD E21
 DISABLE 1,GD E41
 DISABLE 1,GTIT E23
 DISABLE 1,GTIT E21
 DISABLE 1,GTIT E41
 DISABLE 1,DIR E23
 DISABLE 1,DIR E21
 DISABLE 1,DIR E41
 DISABLE 1,NS
 DISABLE 1,ELTS NAME
 DISABLE 1,ELTS NNOD
 DISABLE 1,ELTS ISCT
[XQT ELD
E23 $ROD EEMENTS
NSECT= 2
NREF=2
1 17 1 4 3 1 $
4 20$
5 21$
6 22$
52 68$
53 69$
54 70$
7 23 1 4 10 1$
[XQT E
[XQT EKS
[XQT TOPO
```

RSOUT (nonrepeatable part cont.)

```
RESET MAXSUB=2500
[XQT K
[XQT DCU
 DISABLE 1,DEF E23
 DISABLE 1,GD E23
 DISABLE 1,GTIT E23
 DISABLE 1,ELTS MASK
 DISABLE 1,NS
 DISABLE 1,KMAP
 DISABLE 1,AMAP
 DISABLE 1,E23 EFIL
 DISABLE 1,DIR E23
COPY 1,2 DEM DIAG 0 0
CHANGE 2,DEM DIAG 0 0,DMDV DIAG 0 1
COPY 1,2 K SPAR 25 0
CHANGE 2,K SPAR 25 0,DKDV SPAR 25 1
[XQT ELD
E21
NSECT= 2
NREF=1
1 2 2 16 2 16 $
49 50 2 16 2 16$
33 34$
34 35$
39 40$
40 41$
41 42$
42 43$
43 44$
44 45$
45 46$
46 47$
47 48$
48 33$
[XQT E
[XQT EKS
[XQT TOPO
RESET MAXSUB=2500
[XQT K
[XQT DCU
 DISABLE 1,DEF E21
 DISABLE 1,GD E21
 DISABLE 1,GTIT E21
 DISABLE 1,ELTS MASK
 DISABLE 1,NS
 DISABLE 1,KMAP
 DISABLE 1,AMAP
 DISABLE 1,E21 EFIL
 DISABLE 1,DIR E21
```

RSOUT (nonrepeatable part cont.)

```
COPY 1,2 DEM DIAG 0 0
CHANGE 2,DEM DIAG 0 0,DMIX DIAG 0 2
COPY 1,2 K SPAR 25 0
CHANGE 2,K SPAR 25 0,DKIX SPAR 25 2
[XQT ELD
E21
NSECT= 3
NREF=1
1 2 2 16 2 16 $
49 50 2 16 2 16$
33 34$
34 35$
39 40$
40 41$
41 42$
42 43$
43 44$
44 45$
45 46$
46 47$
47 48$
48 33$
[XQT E
[XQT EKS
[XQT TOPO
RESET MAXSUB=2500
[XQT K
[XQT DCU
 DISABLE 1,DEF E21
 DISABLE 1,GD E21
 DISABLE 1,GTIT E21
 DISABLE 1,ELTS MASK
 DISABLF 1,NS
 DISABLE 1,KMAP
 DISABLE 1,AMAP
 DISABLE 1,E21 EFIL
 DISABLE 1,DIR E21
COPY 1,2 DEM DIAG 0 0
CHANGE 2,DEM DIAG 0 0,DMIY DIAG 0 2
COPY 1,2 K SPAR 25 0
CHANGE 2,K SPAR 25 0,DKIY SPAR 25 2
[XQT ELD
E21
NSECT= 4
NREF=1
1 2 2 16 2 16 $
49 50 2 16 2 16$
33 34$
34 35$
```

RSOUT (nonrepeatable part cont.)

```
39 40$
40 41$
41 42$
42 43$
43 44$
44 45$
45 46$
46 47$
47 48$
48 33$
[XQT E
[XQT EKS
[XQT TOPO
RESET MAXSUB=2500
[XQT K
[XQT DCU
 DISABLE 1,DEF E21
 DISABLE 1,GD E21
 DISABLE 1,GTIT E21
 DISABLE 1,ELTS MASK
 DISABLE 1,NS
 DISABLE 1,KMAP
 DISABLE 1,AMAP
 DISABLE 1,E21 EFIL
 DISABLE 1,DIR E21
COPY 1,2 DEM DIAG 0 0
CHANGE 2,DEM DIAG 0 0,DMDA DIAG 0 2
COPY 1,2 K SPAR 25 0
CHANGE 2,K SPAR 25 0,DKDA SPAR 25 2
[XQT ELD
E21
NSECT= 5
NREF=1
1 2 2 16 2 16 $
49 50 2 16 2 16$
33 34$
34 35$
39 40$
40 41$
41 42$
42 43$
43 44$
44 45$
45 46$
46 47$
47 48$
48 33$
[XQT E
[XQT EKS
```

RSOUT (nonrepeatable part conc.)

```
[XQT TOPO
RESET MAXSUB=2500
[XQT K
[XQT DCU
 DISABLE 1,DEF E21
 DISABLE 1,GD E21
 DISABLE 1,GTIT E21
 DISABLE 1,ELTS MASK
 DISABLE 1,NS
 DISABLE 1,KMAP
 DISABLE 1,AMAP
 DISABLE 1,E21 EFIL
 DISABLE 1,DIR E21
COPY 1,2 DEM DIAG 0 0
CHANGE 2,DEM DIAG 0 0,DMJO DIAG 0 2
COPY 1,2 K SPAR 25 0
CHANGE 2,K SPAR 25 0,DKJO SPAR 25 2
[XQT ELD
E41 $ MEMBANE PANELS
NSECT= 2
1 17 18 2 2 16 1 $
49 65 66 50 2 16 1$
17 33 34 18 1 1 2 2 16$
23 39 40 24 1 1 9 2 16$
32 48 33 17$
48 64 49 33$
[XQT E
[XQT EKS
[XQT TOPO
RESET MAXSUB=2500
[XQT K
[XQT DCU
 DISABLE 1,DEF E41
 DISABLE 1,GD E41
 DISABLE 1,GTIT E41
 DISABLE 1,ELTS MASK
 DISABLE 1,NS
 DISABLE 1,KMAP
 DISABLE 1,AMAP
 DISABLE 1,E41 EFIL
 DISABLE 1,DIR E41
COPY 1,2 DEM DIAG 0 0
CHANGE 2,DEM DIAG 0 0,DMDV DIAG 0 3
COPY 1,2 K SPAR 25 0
CHANGE 2,K SPAR 25 0,DKDV SPAR 25 3
 TOC 2
[XQT EXIT
```

APPENDIX E

RSOUT (repeatable part)

```
[XQT AUS
 DEFINE A1=DKDA SPAR 25 2
 DEFINE A2=DKIX SPAR 25 2
 DEFINE A3=DKIY SPAR 25 2
 DEFINE A4=DKJO SPAR 25 2
 DEFINE B1=DMDA DIAG 0 2
 DEFINE B2=DMIX DIAG 0 2
 DEFINE B3=DMIY DIAG 0 2
 DEFINE B4=DMJO DIAG 0 2
  S1=SUM( .10000+01 A1, .31119+03 A2)
  S2=SUM( .23383+02 A3, .26667+01 A4)
 DKDV SPAR 25 2=SUM(S1,S2)
  T1=SUM( .10000+01 B1, .31119+03 B2)
  T2=SUM( .23383+02 B3, .26667+01 B4)
 DMDV DIAG 0 2=SUM(T1,T2)
[XQT DCU
 TOC 1
[XQT AUS
 SYSVEC;UNIT VEC
 I=1; J=1, 80; 1.0
 DEFINE UN=UNIT VEC
 DEFINE W1=DMDV DIAG 0 1
 DEFINE W2=DMDV DIAG 0 2
 DEFINE W3=DMDV DIAG 0 3
 OBJF G1 1 1=XTY(UN,W1)
 OBJF G2 1 1=XTY(UN,W2)
 OBJF G3 1 1=XTY(UN,W3)
[XQT DCU
 PRINT 1 OBJF G1
 PRINT 1 OBJF G2
 PRINT 1 OBJF G3
[XQT AUS
 OUTLIB=3
 DEFINE F1=STAT DISP 1 1
 DEFINE L1=DKDV SPAR 25 1
 DEFINE L2=DKDV SPAR 25 2
 DEFINE L3=DKDV SPAR 25 3
 ALPHA; CASE TITLE 100
 1 "LOAD CASE 1 DERIVATIVE 1 DESIGN VARIABLE 1
 2 "LOAD CASE 1 DERIVATIVE 1 DESIGN VARIABLE 2
 3 "LOAD CASE 1 DERIVATIVE 1 DESIGN VARIABLE 3
 OUTLIB=4
 APPL FORC 100 1= PRODUCT(-1.0 L1,1.0 F1)
 APPL FORC 100 2= PRODUCT(-1.0 L2,1.0 F1)
 APPL FORC 100 3= PRODUCT(-1.0 L3,1.0 F1)
 INLIB=4
 OUTLIB=3
 DEFINE Z1=APPL FORC 100 1
 DEFINE Z2=APPL FORC 100 2
```

RSOUT (repeatable part conc.)

```
 DEFINE Z3=APPL FORC 100 3
Z11=UNION(Z1 Z2 Z3)
 INLIB=3
Z12=UNION(Z11)
 DEFINE Z10=Z12
 INLIB=4
 APPL FORC 100 1=UNION(Z10)
[XQT SSOL
 RESET L1=1,L2= 3
 RESET QLIB=3
 RESET SET=100
[XQT VPRT
 LIB=3
 PRINT APPL FORC 100
 PRINT STAT DISP 100
[XQT GSF
 RESET L1=1,L2= 3
 RESET QLIB=3
 RESET SET=100
[XQT PSF
 RESET L1=1,L2= 3
 RESET QLIB=3
 RESET SET=100
[XQT DCU
 CHANGE 3,STAT DISP 100 1,DDIS DISP 100 1
 CHANGE 3,STRS E23 100 1,DSTR E23 100 1
 COPY 3,4 DSTR E23 100 1
 CHANGE 3,STRS E21 100 1,DFAM E21 100 1
 COPY 3,4 DFAM E21 100 1
 CHANGE 3,STRS E41 100 1,DSTR E41 100 1
 COPY 3,4 DSTR E41 100 1
 CHANGE 3,STRS E23 100 2,DSTR E23 100 2
 COPY 3,4 DSTR E23 100 2
 CHANGE 3,STRS E21 100 2,DFAM E21 100 2
 COPY 3,4 DFAM E21 100 2
 CHANGE 3,STRS E41 100 2,DSTR E41 100 2
 COPY 3,4 DSTR E41 100 2
 CHANGE 3,STRS E23 100 3,DSTR E23 100 3
 COPY 3,4 DSTR E23 100 3
 CHANGE 3,STRS E21 100 3,DFAM E21 100 3
 COPY 3,4 DFAM E21 100 3
 CHANGE 3,STRS E41 100 3,DSTR E41 100 3
 COPY 3,4 DSTR E41 100 3
 COPY 1,4 OBJF G1
 COPY 1,4 OBJF G2
 COPY 1,4 OBJF G3
 TOC 3
 TOC 1
 TOC 4
[XQT EXIT
```

APPENDIX E

EDIT DATA FILES

EDPASS1

```
A
> $PASSAGE
 NPASS=1,
 $END>
```

Creates PASS file, setting NPASS to 1 for first pass

EDPASS2

```
RS:/2/,/1/
END
```

Changes NPASS from 2 to 1 when restarting

EDIT1

```
F:/[XQT ELD/
F:/[XQT/;2
D;*
R
F:/[XQT ELD/
E;*
R
D;*
R
ADD
$
END
```

Edits out all information from NRRS file except ELD input in nonrepeatable part

EDIT2

```
RS:/    /,/ /;*
RS:/  /,/ /;*
A:/TOPO/;*
>RESET MAXSUB=2500>
END
```

Removes blanks from RSOUT file in nonrepeatable part

APPENDIX E

MERGFP

```
MERGE:/SPFPOUT/,/$ MERGE NEW/;1
R
RS:/E+/,/+/;*
RS:/E-/,/-/;*
END
```

SPFPOUT file is merged into SPARS. E+ and E- are changed to + and - because SPAR does not accept E's in input

EDGRDS

```
RS:/E+/,/+/;*
RS:/.            ./,/. ./;*
RS:/.           0/,/. 0/;*
RS:/.           -/,/. -/;*
RS:/   /,/ /;*
RS:/E-/,/-/;*
RS:/   /,/  /;*
RS:/  /,/ /;*
RS:/ W /,/ W/;*
RS:/ G /,/ G/;*
RS:/,W /,/,W/;*
RS:/ F /,/ F/;*
RS:/ L /,/ L/;*
END
```

Remove blanks from RSOUT in repeatable part

(See ref. 10 for more detail on edit commands.)

REFERENCES

1. Haftka, R. T.; and Prasad, B.: Programs for Analysis and Resizing of Complex Structures. Trends in Computerized Structural Analysis and Synthesis, Ahmed K. Noor and Harvey G. McComb, Jr., eds., Pergamon Press Ltd., c.1978, pp. 323-330. (Also available in Comput. & Struct., vol. 10, no. 1/2, Apr. 1979, pp. 323-330.)

2. The NASTRAN Theoretical Manual (Level 16.0). NASA SP-221(03), 1976.

3. Schmit, L. A.; and Miura, H.: A New Structural Analysis/Synthesis Capability - ACCESS 1. AIAA J., vol. 14, no. 5, May 1976, pp. 661-671.

4. Sobieszcanski-Sobieski, J.; and Bhat, R. B.: Adaptable Structural Synthesis Using Advanced Analysis and Optimization Coupled by a Computer Operating System. A Collection of Technical Papers on Structures - AIAA/ASME/ASCE/AHS 20th Structures, Structural Dynamics, and Materials Conference, Apr. 1979, pp. 60-71. (Available as AIAA Paper 79-0723.)

5. Whetstone, W. D.: SPAR Structural Analysis System Reference Manual - System Level 13A. Volume I: Program Execution. NASA CR-158970-1, 1978.

6. Vanderplaats, Garret N.: CONMIN - A Fortran Program for Constrained Function Minimization. User's Manual. NASA TM X-62,282, 1973.

7. NOS Version 1 Reference Manual - Volume 1. Publ. No. 60435400, Control Data Corp., Sept. 1979.

8. Rogers, James L., Jr.; Dovi, Augustine R.; and Riley, Kathleen M.: Distributing Structural Optimization Software Between a Mainframe and a Minicomputer. Engineering Software II, R. A. Adey, ed., CML Publ. Ltd., 1981, pp. 400-415.

9. Whetstone, W. D.: EISI-EAL: Engineering Analysis Language. Proceedings of the Second Conference on Computing in Civil Engineering, American Soc. Civil Eng., 1980, pp. 276-285.

10. NOS Version 1 Text Editor Reference Manual. Publ. No. 60436100, Control Data Corp., Sept. 1979.

11. Giles, Gary L.; and Haftka, Raphael T.: SPAR Data Handling Utilities. NASA TM-78701, 1978.

1. Report No. NASA TM-83180	2. Government Accession No.	3. Recipient's Catalog No.
4. Title and Subtitle AN IMPLEMENTATION OF THE PROGRAMMING STRUCTURAL SYNTHESIS SYSTEM (PROSSS)		5. Report Date December 1981
		6. Performing Organization Code 505-33-63-02
7. Author(s) James L. Rogers, Jr., Jaroslaw Sobieszczanski-Sobieski, and Rama B. Bhat		8. Performing Organization Report No. L-14246
9. Performing Organization Name and Address NASA Langley Research Center Hampton, VA 23665		10. Work Unit No.
		11. Contract or Grant No.
12. Sponsoring Agency Name and Address National Aeronautics and Space Administration Washington, DC 20546		13. Type of Report and Period Covered Technical Memorandum
		14. Sponsoring Agency Code

15. Supplementary Notes

Rama B. Bhat: Joint Institute for Advancement of Flight Sciences, The George Washington University, Hampton, Virginia, on leave from Concordia University, Montreal, Canada.

16. Abstract

This paper describes a particular implementation of the programming structural synthesis system (PROSSS). This software system combines a state-of-the-art optimization program, a production-level structural analysis program, and user-supplied, problem-dependent interface programs. These programs are combined using standard command language features existing in modern computer operating systems. PROSSS is explained in general with respect to this implementation along with the steps for the preparation of the programs and input data. Each component of the system is described in detail with annotated listings for clarification. The components include options, procedures, programs and subroutines, and data files as they pertain to this implementation. The paper concludes with an example exercising each option in this implementation to allow the user to anticipate the type of results that might be expected.

17. Key Words (Suggested by Author(s))	18. Distribution Statement
Optimization Finite element analysis system Software system Gradients	FEDD Distribution Subject Category 61

19. Security Classif. (of this report)	20. Security Classif. (of this page)	21. No. of Pages	22. Price
Unclassified	Unclassified	134	

Available: NASA's Industrial Applications Centers

NASA-Langley, 1981