NASA Contractor Report 4705

User's Manual for Space Debris Surfaces (SD_SURF)

N.C. Elfer

Contract NAS8–38856
Prepared for Marshall Space Flight Center

February 1996

NASA Contractor Report 4705

User's Manual for Space Debris Surfaces (SD_SURF)

N.C. Elfer
Lockheed Martin Marietta Manned Space Systems

National Aeronautics and Space Administration
Marshall Space Flight Center • MSFC, Alabama 35812

Prepared for Marshall Space Flight Center
under Contract NAS8-38856

February 1996

FOREWORD

The SD_SURF computer programs and user's guide were prepared under contract NAS8-38856 from NASA Marshall Space Flight Center. The study contract title was "Structural Damage Prediction and Analysis for Hypervelocity Impacts." The Technical Monitors were Joel Williamsen, Greg Olsen, and Jennifer Robinson. The code and user's manual were created between October, 1990 and September, 1992. Updates are included through October, 1995

ACKNOWLEDGEMENTS

The author wishes to acknowledge the support and assistance of the following people:

From Martin Marietta Manned Space Systems
Robert Meibaum
John Magyari

From NASA Marshall Space Flight Center
Joel Williamsen
Greg Olsen
Jennifer Robinson
Scott Hill

TABLE OF CONTENTS

APPENDICES

FIGURES

1 INTRODUCTION

The SD_SURF code takes a different approach than that used by other codes such as BUMPER [1], BUMPERII [2], or Space Debris Vulnerability (SDV) [3]. All of the codes treat a space vehicle as a faceted geometry. The space debris environment is considered to be a series of threats from different directions. Each direction has a corresponding impact velocity. The other codes calculate the probability of no penetration for each facet based on the exposure area and the penetration resistance (ballistic limit) to each threat's impact velocity and obliquity. This output tells the designer which areas are most vulnerable. However, the output does not give any information to help select the most efficient shield design for a given area. While parametric studies can be designed to cover all possibilities, additional information can help a designer narrow the number of variables to be optimized for a given penetration mechanism.

The new approach, used in the SD_SURF code, first summarizes an exposed area on the spacecraft in a table of velocity and obliquity. The table can be generated from a description of a simple geometry (plane, sphere, or cylinder) or the code can read the GEOM output from BUMPERII. This allows a complex geometry to be processed (including self shielding calculations) and stored as a small file for further trade studies or optimization.

The analyst who is familiar with BUMPERII will feel right at home with the FORTRAN applications on both the Macintosh or the VAX. Installation is described in Section 6. A complete review of this manual is not critical, so the most significant warnings are repeated here.

WARNING: Unlike the VAX, if you save files with the same name on the Macintosh the earlier file will be deleted!

NOTE: Unlike, SHIELD, multiple element ID ranges are lumped together by A_SURF, and multiple PIDs in all of the selected ranges are ignored.

The facets in each range are summed to only one area array regardless of PID. If different PIDs must be analyzed separately, (eg. a window along a module) the analyst must select only those elements with the desired PID. The analyst must know the model!

2 BACKGROUND

Previous approaches will be explained first, since SD_SURF expands on that work. Space debris codes probability codes BUMPERII (and its predecessor BUMPER), and Space Debris Vulnerability (SDV) analyze a space vehicle as a faceted geometry.[1-3] These codes calculate the probability of no penetration for each facet based on the exposure area and the penetration resistance (ballistic limit) to each threat's impact velocity and obliquity, as described in the following sections. This output tells the designer which areas are most vulnerable.

2.1 Environment

The space debris environment is defined in terms of a flux of particles of diameter, *d*, or larger, dependant on the year of interest (due to assumed growth in the environment as well as solar flux) and the spacecraft altitude.[4] Figure 1 shows a flux versus *d* curve for typical parameters of interest.

The space debris environment may be modeled as a series of threats from discrete directions. For low earth orbit (LEO), space debris may be assumed to exist in circular orbits. This assumption fixes the orbital velocity. Debris cannot intercept a spacecraft from more than approximately 10° above or below a plane tangent to the local Earth normal, otherwise the debris would enter the Earth's atmosphere and be removed as a threat. Therefore, the relative impact velocity in LEO is determined by the orbital velocity, V_o, and the intersection angle, Ø, of the two orbits. The impact velocity, V_i, is:

$$V_i = 2\, V_o \cdot \cos\left(\frac{180° - Ø}{2}\right)$$

Figure 2 shows the fraction of the total flux coming from angles relative to the direction of flight. The relative impact velocity for the intersection of 388 km orbits is also shown on the plot.

When the spacecraft attitude is fixed relative to the earth, the orientation of each facet on the surface will determine the most probable impact velocities and obliquities.

2.2 Ballistic Limit Surface

The spectrum of debris sizes, velocities, and obliquities which may impact a shield lead to a variety of penetration mechanisms. These are illustrated in Fig. 3. Figure 4 illustrates a ballistic limit surface for hypervelocity impact on a multi-wall shield. A projectile diameter at a velocity and obliquity above the surface will penetrate the shield. A diameter below the surface will not penetrate the shield. Changes in shield parameters affect each penetration mechanism differently. Therefore, it is important for the designer to know what penetration mechanism has the greatest effect on the overall probability of no penetration.

2.3 Probability Analysis

The probability of no penetration (PNP) from each direction and for each element is based on the Poisson distribution for zero events:

$$PNP_{el} = \exp\left(- \sum_{i=1}^{nthreats} (N_i \cdot A_i) \cdot t \right)$$

where (with consistent units)

N_i = flux which penetrates from each threat direction, *i*.
  = $4 \cdot f_i \cdot N_r(d_i)$

N_r = flux on a randomly tumbling plate of diameter d_i or larger. (As defined in the specifications.)

d_i = diameter to penetrate at the velocity and obliquity of the *i*th threat.

f_i = fraction of flux from threat direction

A_i = projected area of the facet in the flux direction.

t = exposure time.

The total PNP is determined by the product of the PNP for each element.

$$PNP_{total} = \prod_{j=1}^{nelements} PNP_j$$

Figure 5 shows the BUMPERII modules and their input and output as they calculate PNP.

BUMPERII starts with a SuperTab output file finite element model of the spacecraft. Figure 6 shows a model of Space Station Freedom.[6] The GEOMETRY module of BUMPERII calculates the projected area of the elements exposed to each threat direction based. A significant part of this calculation is intercomponent shadowing. This can be a very time consuming process for a large model.

The RESPONSE module creates a ballistic limit surface from a menu of user selected penetration equations. The ballistic limit for each shield of interest is stored in a matrix for every 0.25 km/s and 5° obliquity. This is also stored in binary form in the computer. Another BUMPERII code, RPLOT, reads the binary file and puts out a formatted file with the ballistic limit at 0°, 15°, 30°, 45°, and °60 obliquity for 2D plots.

The SHIELD module calculates the PNP for any range of element numbers requested by the analyst. SHIELD also has an option to create a SuperTab file to plot probability contours on the original geometry model.

3 SD_SURF Analysis Approach

To design the most effective shield, the analyst must know which penetration or damage mechanism is predominant. It is the goal of the Space Debris Surfaces (SD_SURF) computer programs to provide this information.

The flux associated with each point on the ballistic limit surface can be weighted by the probability of an impact at that particular velocity and obliquity.

$$PNP(V,\beta) = \exp[-N(d)\cdot f(V)\cdot A(V,\beta)\cdot t]$$

where

A(V,β) = total projected area of the spacecraft that will be impacted from a debris particle at an obliquity, β, at velocity *V*.

f(V) = the fraction of the total flux at velocity *V*.

N(d) = the flux associated with the diameter *d* that just penetrates at *V* and *β*.

The SD_SURF approach is to store the elements of area in an array in small increments of velocity and obliquity. The total PNP is then given by:

$$PNP_{total} = \exp\left(-t\cdot\sum_{i=1}^{V}\sum_{j=1}^{\beta}(N(d_{i,j})\cdot A(V_i,\beta_j))\right)$$

There is a difference in the PNP calculated for a unit area at a single velocity and obliquity versus distributing the area over two bracketing velocities and two bracketing obliquities. This is due to the non-linear relationship between flux and diameter. On the other hand, the analysis of a curved surface in BUMPERII is more accurate than SD_SURF only if the angle subtended by the facets is smaller than the five degree increments used on the RESPONSE and AREA_SURFACE tables. SD_SURF overall probability calculations may be used as confidently as BUMPERII for models that have coarser increments than 5° facets and 90 threat directions, regardless of the ballistic limit surface. Finer models are prohibitively time consuming and will not necessarily produce a different result.

Like any computer model that treats a continuous process as a discrete or finite element, there is a chance of introducing errors. Of course, BUMPERII and SD_SURF do not require the same level of debugging as a finite element or hydrocode model. One sources of potential error is in the shadowing and area calculation, which BUMPERII does quite well. The partial shadowing option is a good quick way to determine if the discretized environment and geometry affect the effective area. The second potential source of error is how well the ballistic limit surface is interrogated. The old meteoroid method of using the average impact velocity is certainly inappropriate for space debris.

If the ballistic limit surface is smoothly varying there is potentially a small error introduced by lumping all of the exposed area of a curved surface into one flat facet and the debris angular distribution into a discreet number of threats. Each velocity and facet treats all of the exposed area as if it occurs at one velocity and obliquity. This is a relatively small error, the magnitude of which depends on the curvature of the ballistic limit surface.

However, if the velocity and obliquity increments are large, and the ballistic limit surface has deep troughs or sharp peaks, then a larger error is possible. It is possible to miss key areas. In other terms, the ballistic limit surface can be undersampled. What matters to the analyst is whether it affects the result. The shape of the ballistic limit surface has a direct impact on the fidelity of the environment and geometry models needed to sample it. The SD_SURF output provides information to judge whether the cusps in the ballistic limit surface were caught by the model and whether they will influence the PNP.

4 SD_SURF - FORTRAN VERSION

The interrelationship of the FORTRAN modules of SD_SURF is shown in Fig. 7. SD_SURF acts as a post-processor of BUMPERII-RESPONSE and GEOMETRY output. It provides additional information not readily obtainable from BUMPERII.

Only P_SURF and A_SURF are required to perform an analysis. The source codes are in Appendicies A and B respectively.

The A_SURF module reads the BUMPERII-GEOMETRY binary output to create the exposed area matrix as a function of velocity and obliquity. Rather than lump the area of one facet at the nearest velocity and obliquity, A_SURF uses the lever rule to distribute the projected area, for one facet and one threat, over the four nearest velocities and obliquities. The sum of the exposed areas is equal to the area reported by BUMPERII.

The A_SURF module creates both an unformatted file and a formatted file. The unformatted binary file can be read by the P_SURF module. The formatted text file can be used to manually check the output, or it can be read by the EXCEL modules as described in the next section.

The P_SURF module reads in the A_SURF and RESPONSE output files, and uses the same flux routines in BUMPERII-SHIELD to calculate the flux·area·time (NAT) array. A text based contour map is generated which should be compatible with any FORTRAN platform, as well as a text file which may be used for sophisticated graphics packages. Examples of the contour plots will be shown in the examples in the next section of this paper.

The final FORTRAN module is R_PLOT5. The source listing is given in Appendix C. It is used to translate BUMPERII-RESPONSE output files to text formatted files. The text formatted file is set up at 0.5 km/s and 5 degree increments rather than the 0.25 km/s and 5 degree increments used by RESPONSE. Commas are used as delimiters to ease import by EXCEL.

4.1 Platform Selection

SD_SURF and BUMPERII have been compiled on VAX and Macintosh computers. Language Systems FORTRAN version 3.0 was used for compilation on the Macintosh with minimal changes from the original FORTRAN code.[6] Limitations to Language Systems FORTRAN are given in Appendix D.

The Macintosh applications cannot handle finely resolved models or meteoroid analyses due to memory limitations. However, this does not affect the RESPONSE module. Debris and meteoroid analysis with 145 threats in

BUMPERII can be used on models with 2100 elements. If partial shading is used in BUMPERII, additional elements are created so the total number of elements must stay within the allocation.

The Macintosh SD_SURF and BUMPERII applications make it convenient to share data with EXCEL. However, many mainframe computers are networked with Macintosh and IBM compatible PCs. The FORTRAN programs may be run on a mainframe and the text files may be transferred to a personal computer and used in an EXCEL analysis or for import into any available charting package.

Language Systems FORTRAN did not support jumping into IF-THEN or DO loops, as allowed by VAX FORTRAN. However, this requires only 3 minor changes to the original BUMPERII code, and it has been requested that these changes be incorporated in future releases of BUMPERII. Furthermore, Language Systems FORTRAN and Absoft FORTRAN for the Macintosh require that the variable size match between the calling program and the subroutine dummy variables. A REAL*8 variable in the main program must be matched with a REAL*8 dummy variable in the argument list of the called subroutine, otherwise wrong numbers will be transferred. The same holds true for integer variables as well. FORTRAN-LINT by Information Processing Techniques Corpation, (Palo Alto, CA (415)-494-7500) provides a means of checking that this argument mismatch is detected, since it is not identified by either Macintosh compiler.

5 SD_SURF - EXCEL 3.0 VERSION

The the EXCEL version offers an alternative to the FORTRAN version. The final product is not as fast or as "turn key" as a FORTRAN application. However, it has the advantages of a spreadsheet. Customization and error checking is very easy and there is easy access to graphing.

The structure of the EXCEL version is shown in Fig . 8. The backbone of the PNP calculation is the PNP_Template. There are several different areas on the worksheet:

- Ballistic Limit surface, diameter to penetrate in increments of 0.5 km/s and five degrees of obliquity. (It is created on a Ballistic Limit Template or imported from RESPONSE via R_PLOT 5.)
- Environment definition including year, solar flux level (explicit or calculated), and altitude.
- Flux calculation for each diameter in the ballistic limit surface. (This is a function macro that is defined on the function macro worksheet.)
- Area Surface, A(V,β), created using Area_Maker Macro, or imported from A_SURF.
- Flux · Area · Time, N·A·T, for each V and β. (The summation of these cells is used to calculate the PNP.)

Function macros operate as subroutines and are used to calculate ballistic limits or flux for appropiate input values. Command macros provide control of files and the pasting of named arrays from ballistic limit and area templates to the PNP_ Template. Any of the templates may be customized and saved by any name for later use. Hardcoding the names would make it easier for a new user, but the flexibility provided by using general names was deemed to be more important.

The Area Surface maybe created on the Area_Template using the Area_Maker Macro. The analyst selects the geometry desired from a pull-down menu. The standard geometries are shown in Fig. 9. The specific geometry is entered in customized dialog boxes shown in Fig 10. Each facet is analyzed at each velocity increment. This is effectively 64 threats (at equally spaced velocities), compared to the 45 threat default in BUMPERII (at equally spaced angles from the direction of flight).

SD_SURF for EXCEL lacks some of the features of BUMPERII. BUMPERII must be used for shadowing analysis in GEOMETRY, multiyear flux averaging in SHIELD, or the extensive iterations required to run PEN4 in RESPONSE. However, the GEOMETRY and RESPONSE output may be imported via the

FORTRAN A_SURF and R_PLOT5 programs. Multiyear flux calculations can be programmed into the EXCEL macros with a corresponding increase in analysis time.

6 INSTALLATION

6.1 Installation - SD_SURF Macintosh Applications

6.1.1. Select One of the Application Disk Options

The selection of which set of applications to run depends on the machine processor, co-processor, and available RAM.

The MacBUMPERII version 1.3 applications supplied require a either 2.2 megabytes or 6 megabytes (MacBumperIIv13M_Large) of ROM and a math coprocessor. Therefore 8 Meg ROM is recommended for the largest BUMPERII option, and 5 Meg ROM should handle all other options. (Virtual memory was not tested.) The SD_SURF programs are smaller than BUMPERII version 1.3 and will handle the output from any BUMPERII option. CONTOUR will not run. The following limits apply:

MacBumperIIv13M_Large (and SD_SURF v. 1.6)

IELM	=	2100	number of elements
ITH	=	145	number of threats (good for meteoroids)
IPFUNCS	=	31	number of PIDS

MacBumperIIv13M_Small

IELM	=	700
ITH	=	145
IPFUNCS	=	12

If different options for array sizes, no coprocessor, etc. are required, and you do not have a Language Systems FORTRAN compiler, please request a customized application.

If Batch file processing is desired on the Macintosh (useful for RESPONSE surface generation) create a batch.com file without comment lines. Then strip out excess spaces " " using a word processor! This will avoid some errors in file names and whether English or Metric is read properly. Then use option 8 to read in the file.

All of the files should be copied to a new folder on your hard disk. Dragging the floppy on top of the hard disk will put all of the files in a folder on the hard disk.

6.1.2. Install the SuperTab Universal Files

Four SuperTab Universal Files were stuffed and placed on the "SD-Surf EXCEL / Stuffed UNIs" disk. The files were placed in a self-unstuffing archive using StuffIt™ Deluxe by Aladdin Systems, Inc. The files are:

- MB17-ALL.UNI
- MB17-CR1.UNI
- MB6-CR1.UNI
- PLATE.UNI

The first three files were distributed with BUMPERII. The plate edge coordinates in the last file may be edited by hand for a plate of any size and orientation.

Open or "Double click" the application (Stuffed SuperTab.Uni Files.sit) and the files will be unstuffed. They should be placed in the same folder as your BUMPERII application. When the files are unstuffed they require approximately 1,800 K on your hard disk. If PLATE.UNI is already there, it is OK to overwrite it.

6.2 Installation - SD_Surf VAX FORTRAN

The source code is provided on both Macintosh and IBM compatible PC formatted disks. The source code and applications should be transferred to the VAX and compiled. The SD_SURF programs version 1.5 work with any BUMPERII version 1.2a output. The SD_SURF programs version 1.6 work with any BUMPERII version 1.3 output. The BUMPERII versions 1.2aM and 1.3M (Martin Marietta Modified) contains the features necessary to compile on the Macintosh. Version 1.2aM also fixes one error in BUMPERII regarding the memory allocation for the variable IDG. Version 1.3M fixes an error in function PRV (with negligible impact on overall PNP).

Note that on the IBM compatible PC formatted disk, the file named Solar_Flux.Dat was renamed due to PC naming restrictions.

The SD_SURF files should be compiled with large enough variables to open BUMPERII output files (in terms of number of elements, threats, and PID cases). These may be adjusted in the COMMON*.BLK files.

6.3 Installation - SD_SURF EXCEL

The EXCEL files are provided on both Macintosh and IBM compatible PC formatted disks. The files should be copied to a folder or directory on your hard disk.

7 Performing a FORTRAN Analysis

To perform an analysis, the GEOMETRY and RESPONSE modules of BUMPERII should be run as described below. Then A_SURF and P_SURF should be run. R_PLOT5 should be run to plot RESPONSE data or prepare data for EXCEL.

7.1 Running Applications on the Macintosh

The applications are compiled to run in the background under Multifinder or in System 7. This slows down the calculations somewhat, but it allows other work to be performed while the analysis is being performed. The calculation of 30 PIDs using RESPONSE is very time consuming.

> WARNING: **Unlike the VAX, if you save files with the same name on the Macintosh, the earlier file will be deleted!**

To stop an analysis, hold down the "Apple" button and type a period. This is the standard Macintosh command to stop a process.

When the code stops, the results in the window may be printed directly from the application, or they may be saved to review or print with any text editor. The results should be viewed using a uniformly spaced font such as Monaco or Courier.

Double clicking on a text file may not open it directly. Instead start any application that will open text files, and then open the files from that application. If using EXCEL, open using the comma delimited option for TEXT files.

There are two additional features on the Macintosh version of BUMPERII. The FINDER open box may be accessed with a "?" and input from a text file is possible for "batch" processing. These features are described in the following section.

7.1.1. Finder Open Box

On the VAX, when BUMPERII requests a filename from the analyst, he may respond with a "?" for a print out of files in the directory. This feature was retained in SD_SURF modules for the VAX.

On the Macintosh, a "?" response brings up the normal FINDER open box. The analyst may then scroll to the file of choice or type the first letters of the file to jump down the list.

7.1.2. Input from a BATCH.COM File

The analyst may create a file using the existing BATCH.COM option in BUMPERII, and then use that file for input by using the new eighth option in the BUMPERII initial options list: Read from a BATCH.COM file. The initial text responses in BATCH.COM are ignored by BUMPERII and the analysis continues from there. This feature is particularly useful for generating RESPONSE output files. The BATCH.COM file may be edited using a text editor instead of repeating the BATCH.COM process.

7.2 BUMPERII GEOMETRY

The GEOMETRY subroutine is run as normal in BUMPERII.

7.3 BUMPERII RESPONSE

The operation of the RESPONSE subroutine is unaffected. A RESPONSE analysis to support a SHIELD analysis is perfectly acceptable. However, it is not necessary to create a certain number of PIDS as required by SHIELD. Only one shield is required for an analysis of a specific range of elements. Another option is to create a series of shield for parametric analyses (eg. step through bumper and/or rear wall thickness and/or spacing).

7.4 A_SURF

Run A_SURF.

A typical input session is shown in Appendix I. The environment options are identical to BUMPERII and described in the BUMPERII user's manual.

NOTE: Unlike, SHIELD, multiple element ID ranges are lumped together by A_SURF, and multiple PIDs in all of the selected ranges are ignored. The facets in each range are summed to only one area array regardless of PID. If different PIDs must be analyzed separately, (eg. a window along a module) the analyst must select only those elements with the desired PID. The analyst must know the model!

The text file may be reviewed with a text editor, or used by EXCEL or any other charting package.

7.5 P_SURF

Run P_SURF. The A_SURF binary file (.ASB) is used for input, as is a RESPONSE output file (.RSP) .

A typical input session is shown in Appendix I. The environment options are identical to BUMPERII and described in the BUMPERII user's manual.

The text based carpet plot is output to the screen as shown. The analyst may use the A_SURF output with any single PID or all of the PIDs in the RESPONSE output. The latter case is useful for parametric studies.

The output file may be reviewed with a text editor, or used by EXCEL or other charting package. It contains both the text based carpet plot and the calculated values. The description of all PIDs is included in the output due to the structure of RESREAD subroutine used from BUMPERII-SHIELD.

It should be remembered that a group of cells with moderate values of NAT can have a greater influence on the overall PNP than a single cell with maximum NAT.

7.6 R_PLOT5

Any or all of the PIDs in a RESPONSE output file may be converted to text format by R_PLOT5. Run R_PLOT5. Select the Response output file and select one or all of the PIDs for output. The description of all PIDs is included in the output due to the structure of RESREAD subroutine used from BUMPERII-SHIELD.

8 PERFORMING AN EXCEL ANALYSIS

The EXCEL analysis may be performed on either a Macintosh or a PC.

If A_SURF and R_PLOT5 text files are to be used, the PC must import from a VAX or other computer on which the the codes can be compiled. The Macintosh may import from a VAX or use the Macintosh application files.

There are minor differences between the Macintosh and Windows EXCEL spreadsheets and macros. The shorter PC names were adjusted in the macros. The graphics in the AREA_Template on the Macintosh would not convert to the PC. A text description is included.

General guidelines are to open the Macros first (SD_SURF and AREA_MAKER) and let them open the templates (or guide you through the process) so that the names can be recorded to the macro sheets. Use the pull down commands to change sheets or save, so that the current names can be recorded.

Watch the message box at the lower left for instructions.

The Macros are documented in Appendices E, F, and G. Typical output is given in Appendix H.

The analyst should be familiar with EXCEL. These features should be reviewed in the EXCEL manuals:

- Command and Function Macros.

  Command Macros can automatically perform almost any function you would do manually in EXCEL. The macro commands may be accessed by "Run" under "Macro." Most are available installed in pull down menus.

  Function Macros perform like a sub-routine in BASIC or FORTRAN. Variables may be passed to the function macro and one or more variables may be returned. Function macros are used by the worksheet templates to perform calculations. Function macros appear at the end of the list generated by the Paste Function command, and are available for pasting into any worksheet when the macro sheet is open. The arguments for the function may be abbreviated in the name, or may be determined by looking at the macro sheet. They are identified by the ARGUMENTS function in the order they are received.

- Hiding files - The "Hide..." and" UnHide..." commands under "Window" on Excel can be used to keep your work area neat or to let you get to the inner workings respectively. If after you make a change you want the sheet to open up hidden the next time you run it, just change a cell (e.g. add and delete a space), hide the sheet (without saving it), quit, and say yes when it asks if you want to save changes. It will be hidden the next time you open it.

- Changing file links (especially useful for charts or function macros) is under the FILE menu.
- Automatic/Manual recalculation - how to change (Options - Calculation) and avoid on saving or printing (Apple-period stops recalculation on saving).
- Auto_open/close macros. These run automatically at the open or close of a macro sheet. Pull down menus and opening dialog boxes are added (or deleted) with these macros.
- Excel Startup Folder - Files place in this folder (in the System folder) will automatically be opened at the start of an EXCEL session. The SD_Function_Macros may be moved here for general usage. If this is done the SD_SURF Auto_Open Macro should be changed so it will not try to look for the file.
- Open/Save File Options - Review Open comma delimited text files (CSV = comma separated variables) and Save.As Options. See Problems Section.

8.1 SD_SURF Macro / PNP_Template

Open the SD_SURF Macro first. An auto_open macro will try to open the SD_FUNCTION_MACRO sheet. If it fails the analyst is requested to open the sheet. The auto_open macro asks the analyst to open a PNP_Template. (NB: prompts are in the Message window at lower left of the screen) A PNP_Template is provided. If a particular ballistic limit or effective area surface has been included it may be saved by a new name and opened by the analyst the next time.

A pull down menu is installed with these commands:

SD command	Function (This is displayed in the Message Bar.)
Open R_Plot5 Output	Opens R_PLOT5 output to Paste to PNP/Flux Template
-	
Open BL Template	Keeps track of which file to use as Ballistic Limit template using macros.
Ballistic Limit to PNP	Copy Ballistic Limit from Active BL Template to PNP/Flux Template
-	
Open Area Maker Macro	Use Area Maker Macro to open A_Surf output or create new geometries.
-	
Open PNP Template	Keeps track of which file to use as PNP/FLUX template.
Save PNP Template	Keeps track of which file to use as PNP/FLUX template.
Set PNP/Flux Template	Keeps track of which file to use as PNP/FLUX template.
-	
Close SD Surf Macro	Closes Macro and deletes SD menu.

These are the instructions included on the PNP_Template:

QUICK INSTRUCTIONS:

- BALLISTIC LIMIT Calculations
  - Use Ballistic Limit Worksheet & BALLISTIC LIMIT TO PNP (in Pull Down Menu) or...
  - Run RESPONSE and R_PLOT5 and Use OPEN R_PLOT5 OUTPUT (In Pull Down Menu)
- GEOMETRY Analysis
  - Use Area Maker Macro to make a new table or open GEOMETRY/A_SURF output
- ENVIRONMENT Enter in Q15-Q21 this worksheet (or Velocity distribution on Area_Template) or...
  - Change SD Flux Function MACRO (and Quick Flux macro if necessary) or...
  - Change flux formulas D79-V110 (Fill right and down)
- Use the PULL Down Menu on Right end of Menu Bar and watch Message Bar at bottom.

Printing of the results is set up for a Macintosh LaserWriter. Other printers must be formatted by the analyst. The print area contains some pages which are only filled under certain conditions (eg. pasting from the Area Template). These may be deleted from the Print Area if desired.

8.2 Ballistic Limit

The Ballistic Limit Template may be used to create a surface. Only the JSC Whipple Bumper and Multi-shock equations are include in the SD_Function_Macro because the PEN4 routine requires too much computation for an interpreted spreadsheet. The Ballistic Limit Template can be used to "breadboard" new equations or custom modify a RESPONSE / R_PLOT5 text file. Note that BL_Paste copies specific cell ranges rather than named areas if you wish to modify the template or create a new template.

The SD pull down menu can be used to open the template or the R_PLOT5 text file and paste the results to the PNP_TEMPLATE.

8.3 AREA_MAKER MACRO and Area Template

Open the Area Maker Macro first. An auto_ open macro will then request the analyst to open the Area Template.

A pull down menu is installed with these commands:

AreaS Command	Status Bar Text
Clear Arrays	Clears Area_Array & Description_Array on Area Template
Rectangle	Adds a Rectangle to Area_Array
Disk	Adds a Disk to Area_Array
Cone	Adds a Cone to Area_Array
Cylinder	Adds a Cylinder to Area_Array
Sphere	Adds a Sphere to Area_Array
Whole Sphere	Adds a complete sphere to Area_Array. Faster than Sphere!
-	
Open Template	Opens a file to be used as the Area Template
Save Template	Saves Template. Identifies new name as the Area Template
Set Template	Identifies active document as the Area Template
AreaS to PNP	Transfers Effective Area to PNP Template.
-	
Open A_SURF file	Opens A_Surf/ Output and puts in 0.5 km/s increments.
-	
Close AreaS	Closes Area Maker Macro.

The pull down menus are used to create an area array or open an A_SURF text file. Figures 9 and 10 shows the geometries and the Dialog Boxes used to describe each geometry. The analyst is advised to take advantage of symmetry and the Area Multiplier rather than spending extra time calculating symmetric facets. (Eg. a cylinder with its axis along the flight direction can be modelled as one fourth of a cylinder and an area_multiplier of 4. With one axis of rotation, only one half the cylinder need be modelled.) If a different inclination is desired, then the analyst need only put the value on the template and the velocity distribution will re-calculate. The cells refer to a function macro on the AREA_MAKER Macro. The functions, which still need to be normalized, are next to the velocity distribution but have zero cell width. Select adjacent cells and set the width of all of them to standard to see the function cells.

The effective area array may then be copied to the PNP_Template. If the desired PNP_Template is already open it will be replaced by the last saved template. (If it has not been saved, when the "Revert" warning box comes up select "Cancel" to stop the macro.)

9 PROBABILITY STUDIES

9.1 Effective Area

The A_SURF program and the Area_Template calculate the effective exposed area, f(V)·A(V,β), at each velocity and obliquity.

Figure 11 illustrates the analysis of a flat plate that is oriented edge on to the direction of flight. The first part of the analysis is the calculation of the projected area, A(V,β), relative to each impact velocity direction. Figure 11 (b) shows the probability, f(V), associated with each impact velocity. Figure 13(c) shows the final result, f(V)·A(V,β), after multiplying the projected areas by the relative probability.

A_SURF reveals the coarseness, or granularity, in the spacecraft model and debris threat in the GEOMETRY analysis. Solving the first problem (a plate edge on in Fig. 11) using BUMPERII produces Fig. 12. The default of 45 threat directions in BUMPERII gives only 22 velocities due to symmetry. There are now gaps along the velocity axis. The "waves" on the surface are an artifact of the coarseness of the modelling. This does not imply the overall model is in error, but rather it shows how BUMPERII and SD_SURF are querining the ballistic limit surface. If the "wave" spacing is small compared to changes in the ballistic limit surface, then the overall PNP calculation is correct.

Since the distributions are not smooth, the analyst must recognize that adjacent cells with moderately high impact rates can be more significant than a single cell with the maximum impact rate.

The A_SURF output can be used to double check the original SuperTab model. If some elements were entered with normals in the wrong direction there may be unexplained gaps in the model.

The sphere is an easy shape to analyze since it looks the same from any direction. (That is why it is a separate option in the AREA_Maker macro.) The projected area from any direction is shown in Fig. 13. Also shown is what it would look like if modelled using facets that cover 15 degrees of curvature. The granularity, or waviness is obvious.

The sphere is also a good representation of the surface area of any spacecraft which is not Earth oriented. It will appear to be randomly tumbling to the debris flux and average out to the oblique impacts on a sphere with the same surface area.

9.2 Penetration Analysis

Figure 14 shows the P_SURF analysis of the effective area in Fig. 13. This is an example of the text based contour plot. The ballistic limit was the RESPONSE output for a 0.050 inch bumper, 4 inch standoff, MLI, and a 0.125 inch 2219 aluminum rear wall, using the regression equation and default analysis of Wilkinson momentum failure.

Figure 15 is an illustration of the velocities and obliquities for which most penetrating impacts could occur on one early concept for a space station module. (The same RESPONSE ballistic limit surface is used as in the previous example.) It can be noted that BUMPERII analyzed the PNP for one year as 99.88305%, while P_SURF calculated it as 99.88475%. The effective area was identical, but as mentioned previously, partitioning the area to discrete velocities and obliquities will affect the result, just as assuming a curved surface is represented by a flat facet. The probability of penetration (POP = 1 - PNP) was 0.11695% for BUMPERII to 0.11525% for P_SURF. The percent change between the two is 1.5% of the POP. This difference is negligible.

10 Problem Resolution

This section is meant to help trouble shoot any errors encountered in an analysis.

10.1 Macintosh Applications

There are only limited options if a FORTRAN compiler is not available. (Language Systems FORTRAN version 3.0 was used to compile the applications.)

RAM requirements were suggested by the Language Systems FORTRAN linker. If any unusual errors are encountered, the RAM allocation may be increased using the "Get Info" command under "File" in the "FINDER." (The application must be closed for this to work.) If available memory is a problem these techniques may be used to increase available memory:

- Run using the FINDER only. Use System software 6.X, and turn off Multifinder.
- Remove non-essential software from the System folder.
- Use Virtual memory. The software has worked well using VIRTUAL by Connectix to set memory to 8 Megabytes on a machine with 5 Megabytes installed (running System Software 6.07). (However, at the relatively low price of RAM, a hardware upgrade should be considered.) It has not been tested using System 7 virtual memory.

All data files should be kept in the same folder as the applications. Use of the Finder open box by responding with a "?" may be able to use files outside the application folder, but this has not been tested.

10.2 EXCEL Macros and Templates

If a Macro halts you may unhide the macro, using the "Window" command, and see what operation it was attempting to perform. Display values to see cell results. On the Macintosh the "Apple - `" will toggle between formulas and values. (NB: "`" is a single backquote at the upper left of the keyboard under the "~" mark. It is not an apostrophe.)

10.2.1. EXCEL Macros - Error on Open SD_SURF or AREA_MAKER

If an error occurs on opening SD_SURF or AREA_MAKER, the HIDE() command may be at fault. If either document is hidden and any changes are saved when quitting from EXCEL, the document will open as hidden. Then when running the HIDE() command, it will either hide an open sheet, or an error will occur. To avoid this problem:

- Save changes to the macro with it unhidden. The macro will then open normally and hide itself. Do not save changes when quitting EXCEL unless it is unhidden. You should not have to save changes normally. (To avoid being asked if you want to save changes go to the Auto_close macro and replace the "x" in the "x SAVE.AS(,0)" line with an equal sign. This will stop you from being asked if you want to save changes when quitting EXCEL.)
- Disable, clear or delete the HIDE() command in Auto_Open. Save the document as hidden. (Save changes when quitting EXCEL when the macro is hidden. Make sure the "SAVE.AS()" command in Auto_Close is disabled.)
- Work around the error on open by continuing rather than halting.
- Work around by halting, unhiding the macro, and the run Auto_open using the MACRO pull down menu.

10.2.2. EXCEL - Errors on Opening R_PLOT5 or A/P_SURF Output

When opening text files, EXCEL puts everything in the first column until a tab is encountered. The FORTRAN files are set to write commas after every field in the arrays. Do not open directly from the Desktop or Finder. Start EXCEL and use the Open command. Before opening the file use the Text options button to tell EXCEL that it is comma delimited. The macro commands in SD_SURF and AREA_MAKER should do this automatically.

Display of FORTRAN text files will be enhanced with the use of a uniformly spaced font such as Monaco or Courier. Use Styles to redefine Normal, or select the entire sheet and change the font.

Data in one column may be broken into many using EXCEL's Parse command or the Smart Parse in the Flat File Macro. Consult the EXCEL Manuals.

10.2.3. EXCEL - Updating Links

If files are moved from different locations, EXCEL can lose track of where function macros are. It may be necessary to change links to the current function macros. This is explained in the EXCEL Manuals.

10.2.4. EXCEL - Hints for Custom Analyses

The EXCEL user's manual should be consulted for customization hints. Paste_Special and the Table commands are particularly useful. Setting a color monitor to black and white (in the control panel of a Macintosh) can also speed up response time if the screen is updated or redrawn often. (Several macros set ECHO(FALSE) to speed up recalculation.)

Custom ballistic limit surfaces may be easily generated using the Copy and Paste_Special commands. This can be used to add, multiply or replace the values in a selected range of cells. See the EXCEL Manual for details.

The Table command can be used to parametrically vary input parameters (eg. altitude or year) and record output (eg. PNP). A macro is available from the author if input and output are on different sheets, or if the output already depends on a table.

11 REFERENCES

(1) Coronado, A. et al.: "Space Station Integrated Wall Design and Penetration Damage Control," Contract NAS 8-36426, NASA-Marshall Space Flight Center, 1987.

(2) Graves, R.; and Smiley, J.: User's Guide for Design Analysis Code BUMPERII, Report XD683-99402-1 on Contract NAS8-50000 (1991).

(3) Elfer, N.; *et al.* Martin Marietta IR&D M-01S, unpublished research, 1987.

(4) Space Station Program Natural Environment Definition for Design, NASA SSP 30425.

(5) Elfer, N.; and Rajendran, A. M.: "Space Debris Protection," in T. Wierzbicki, N. Jones Eds. *Structural Failure,* John Wiley & Sons, New York, p. 41-78, 1989.

(6) "Language Systems FORTRAN version 3.0," Language Systems Corporation, 441 Carlisle Drive, Herndon, VA, 22070-4802, (703)-478-0181.

FIGURES

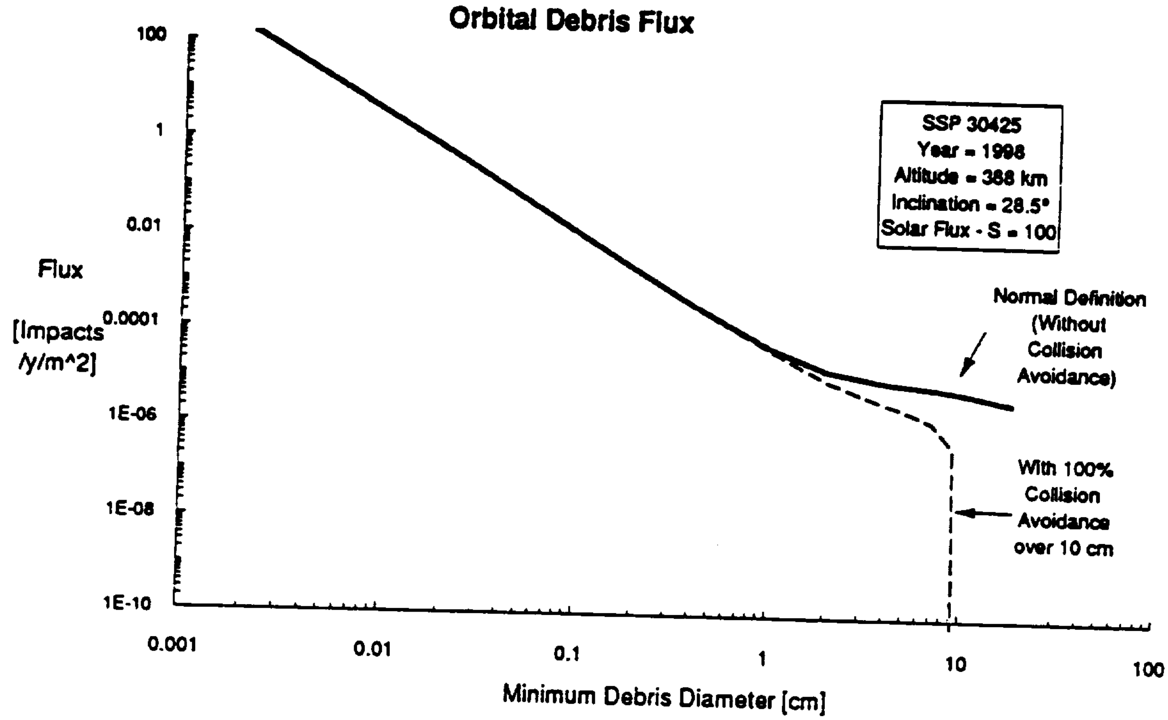


Fig. 1. Impact flux versus space debris diameter

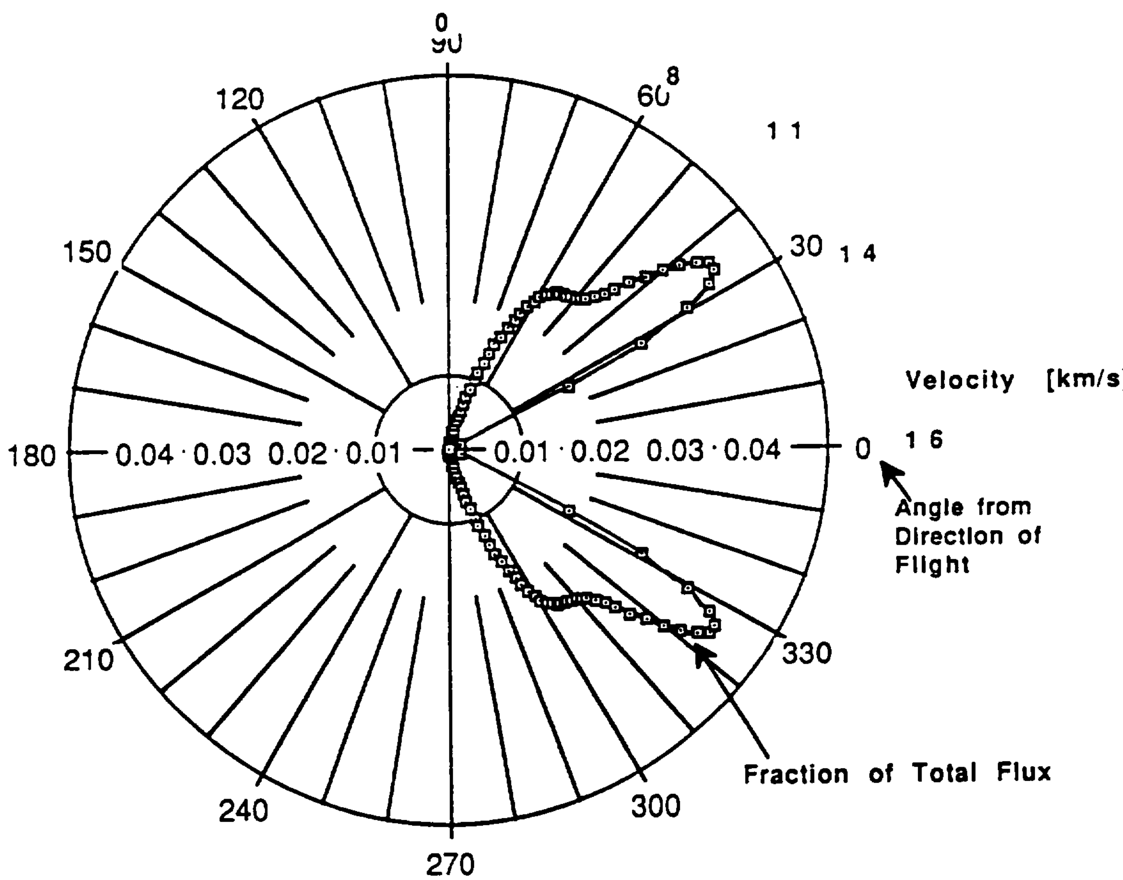


Fig. 2. Angular and velocity distribution of debris flux

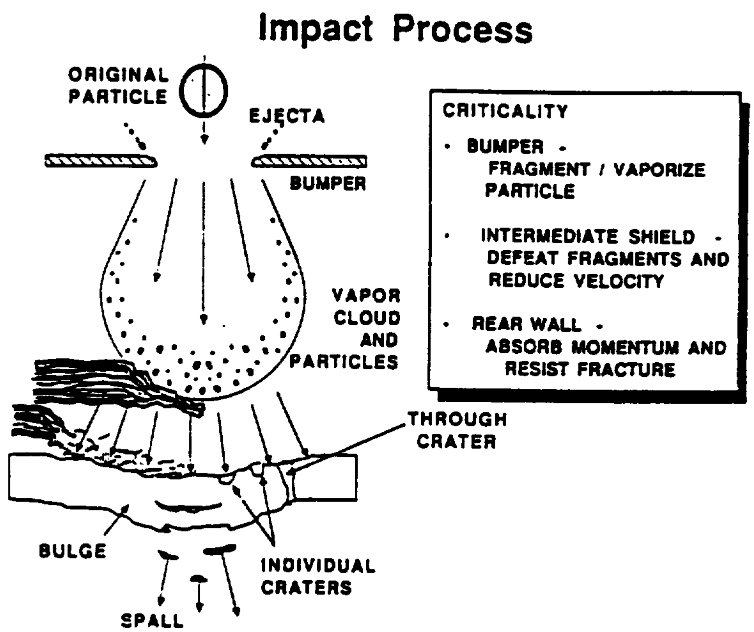


Fig. 3. Penetration mechanisms

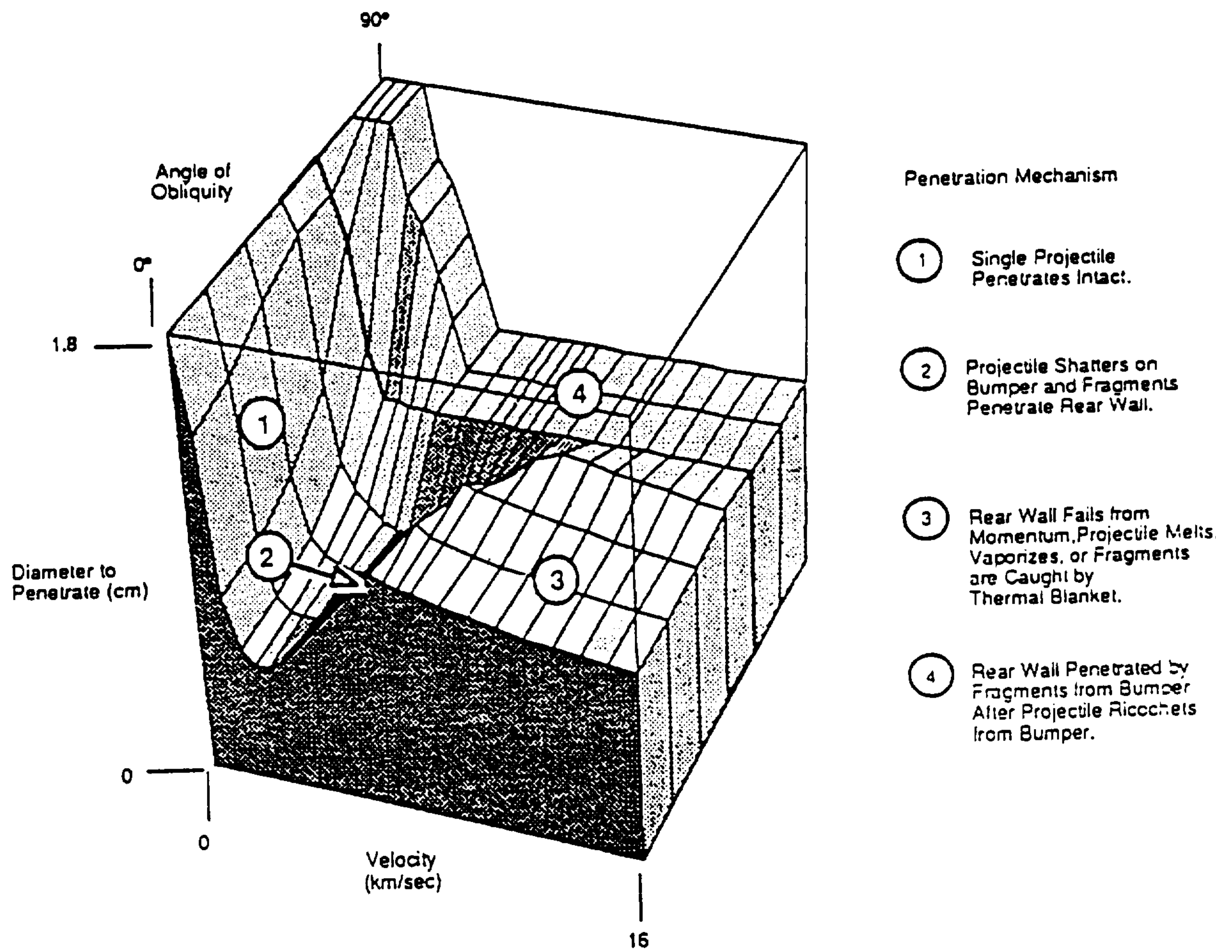


Fig. 4. Ballistic Limit Surface

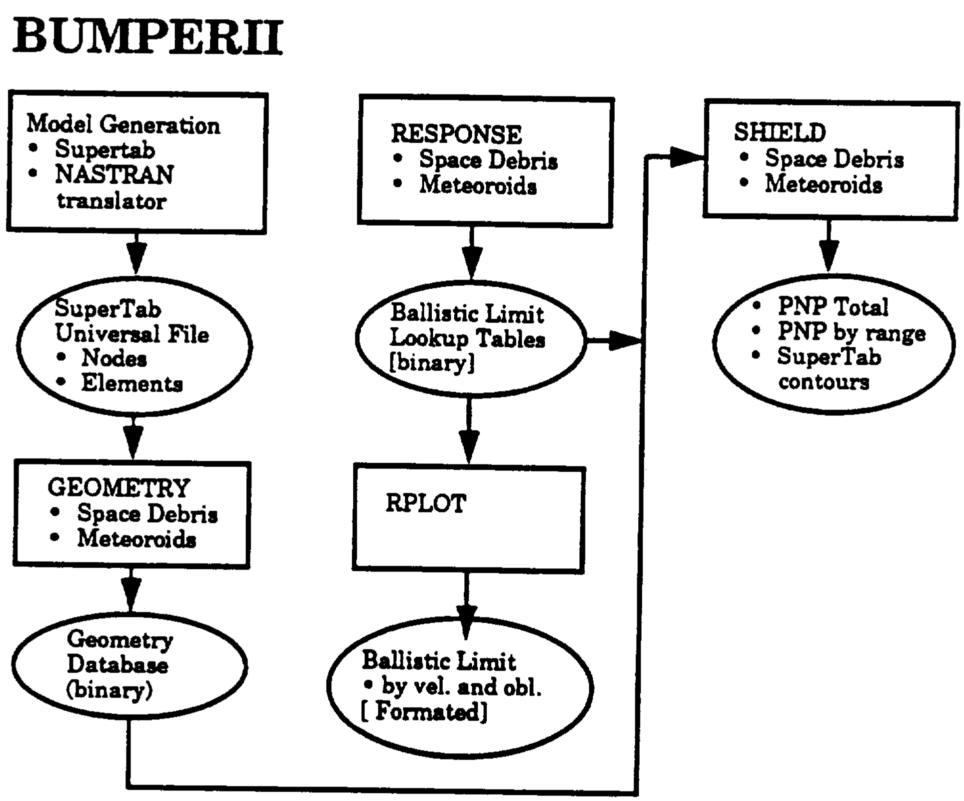


Fig. 5. BUMPERII Modules, Input and Output

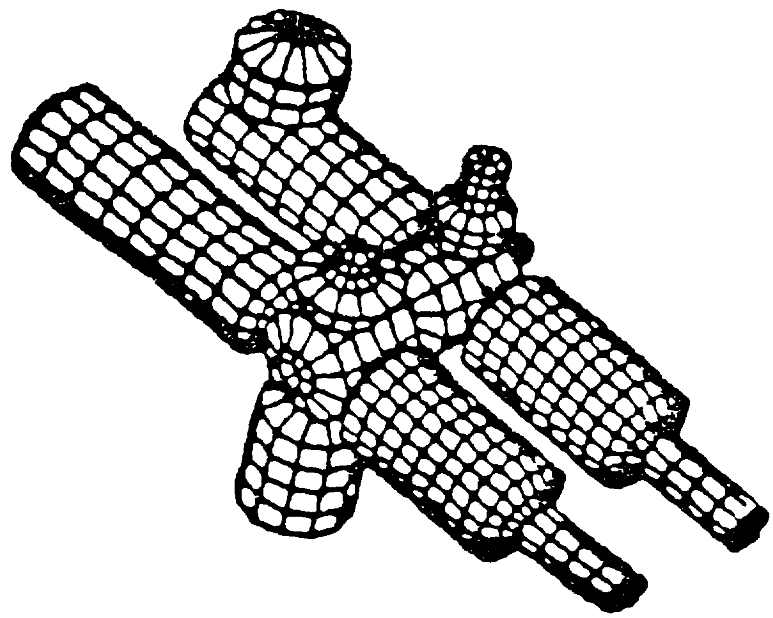


Fig. 6. SSF Model for BUMPERII-GEOMETRY analysis

Space Debris Surfaces - FORTRAN Version

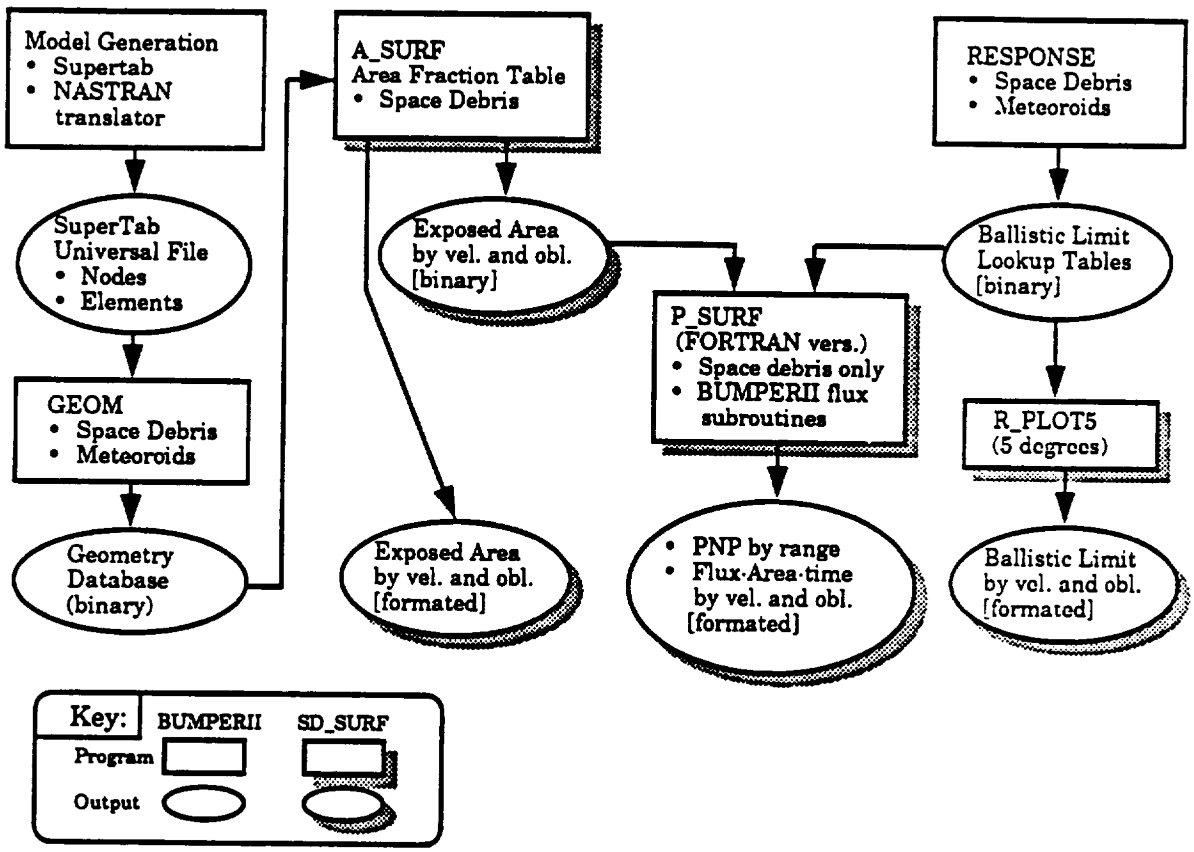


Fig. 7. SD_SURF - FORTRAN and BUMPERII Modules

Space Debris Surfaces - EXCEL Version

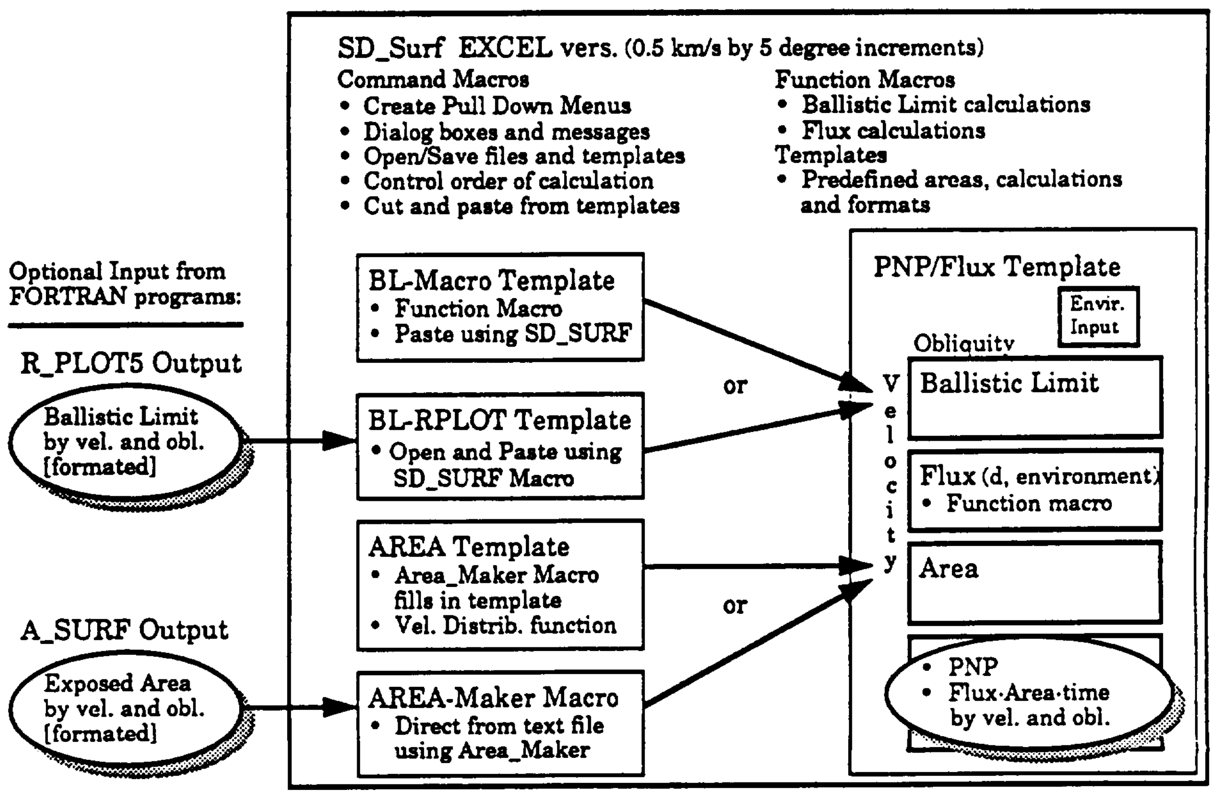
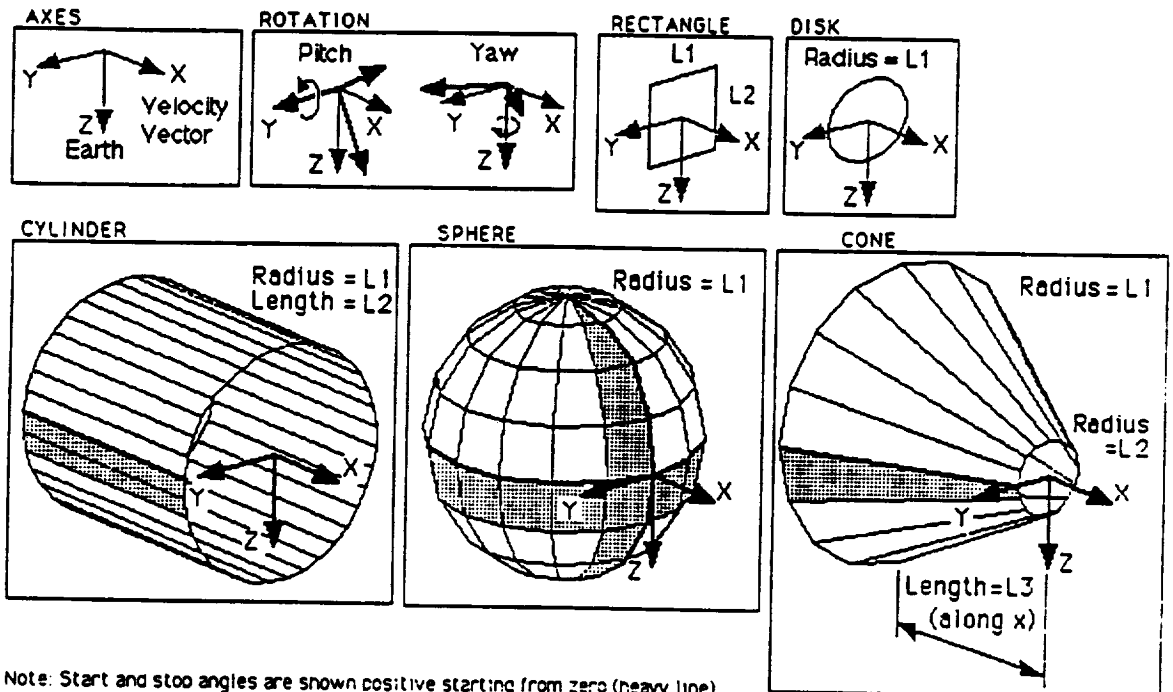


Fig. 8. SD_SURF - EXCEL and SD_SURF - FORTRAN Modules



Note: Start and stop angles are shown positive starting from zero (heavy line).
Negative numbers may be used. (Eg. -90 to 90 for -y side of cylinder.)
Cone and cylinder are not symmetric. 0° to 5° does not also generate 175° to 180°
The difference in the start and stop angles must be evenly divisible by the increment.

Fig. 9. SD-SURF-AREA_MAKER Macro Available Geometries

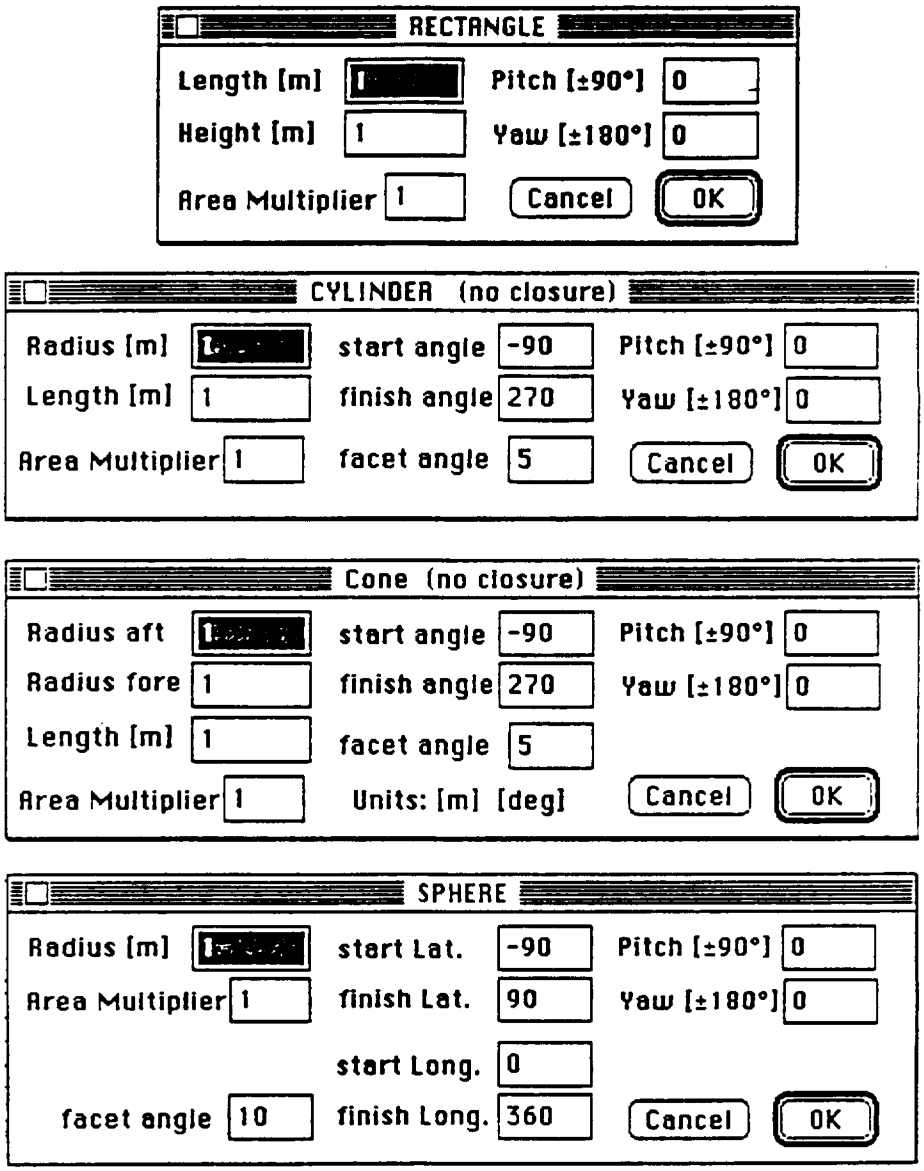


Fig. 10. SD-SURF-AREA_MAKER Macro Dialog Box

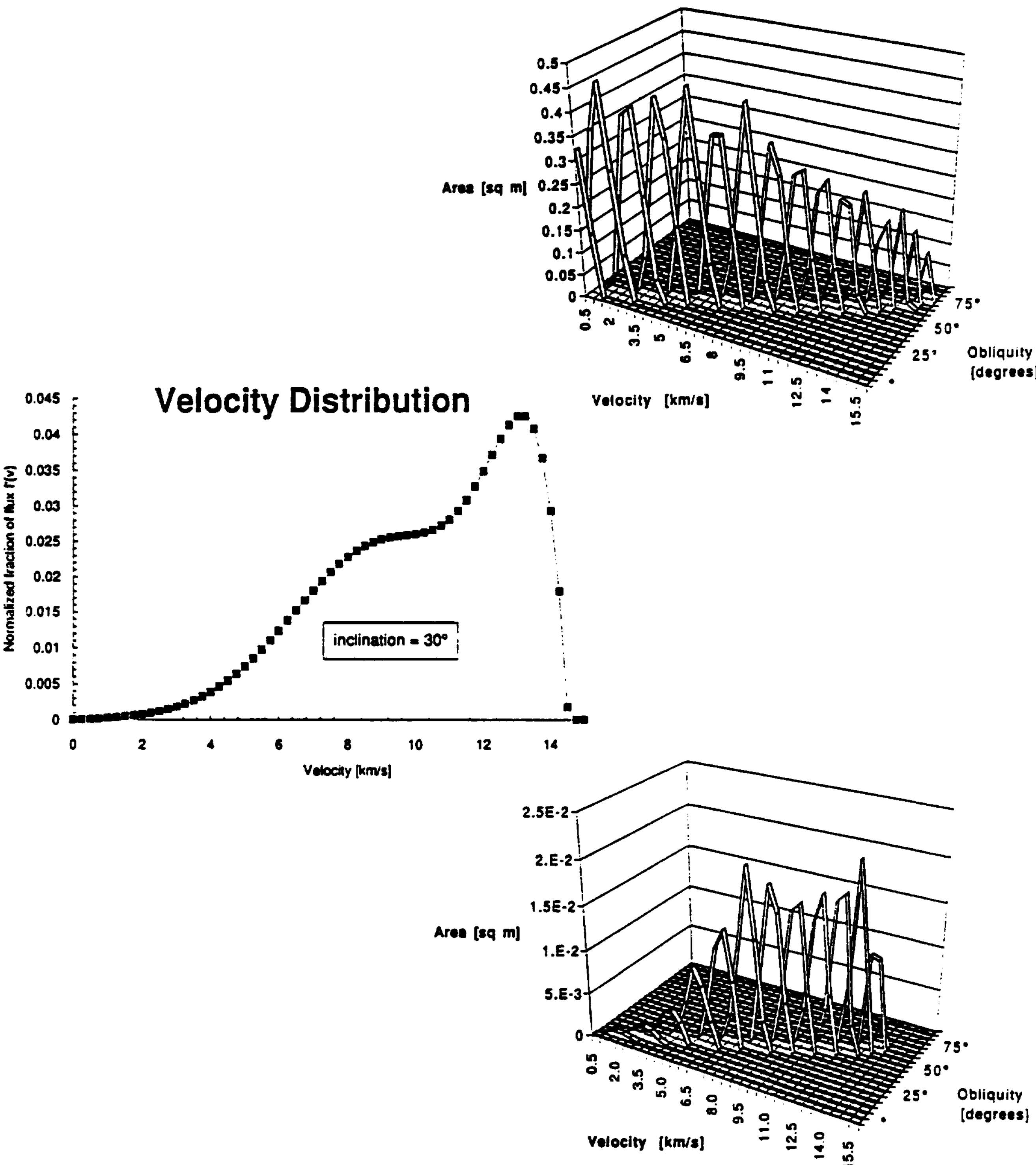


Fig. 11. AREA_MAKER analysis of a plate edge on to x
(The surface normal is in the y axis direction on Fig. 9.)

a) The projected areas in each threat direction.
b) The probability distribution (as in Fig. 2.)
c) The effective area at each velocity and obliquity.

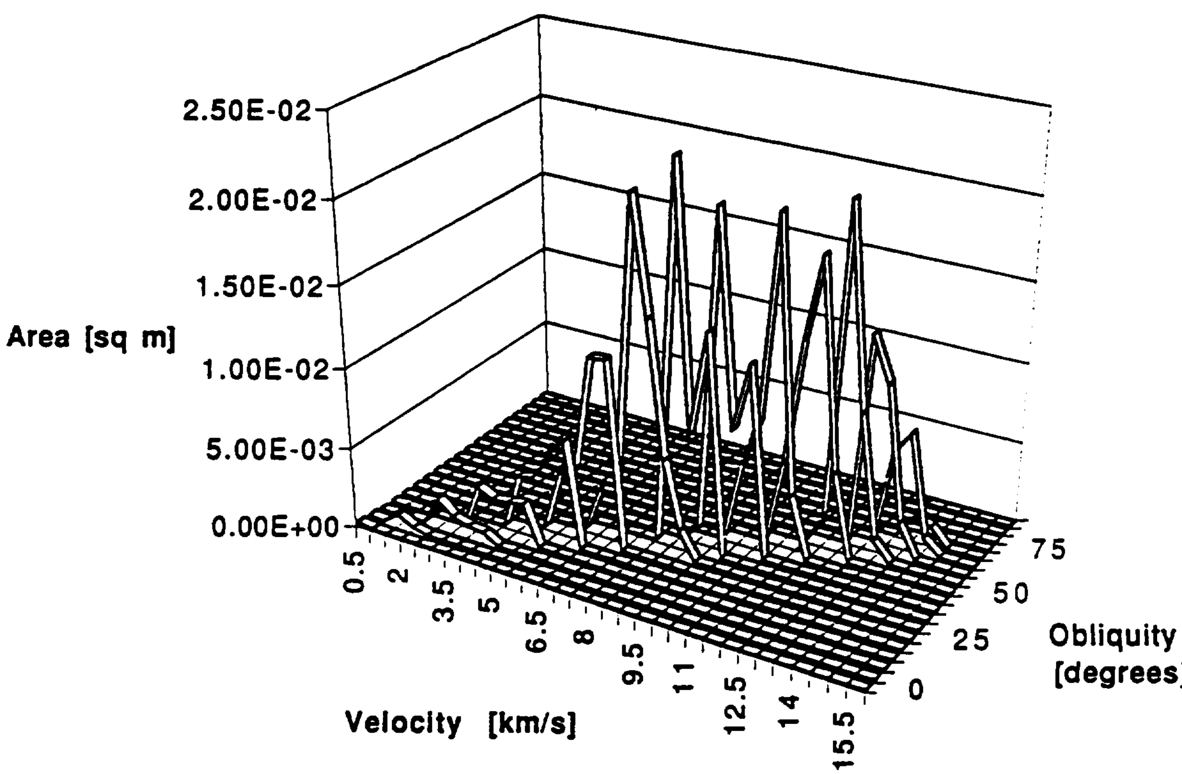


Fig. 12. A_SURF analysis of the same plate in Fig. 11
(45 Threats used in GEOMETRY)

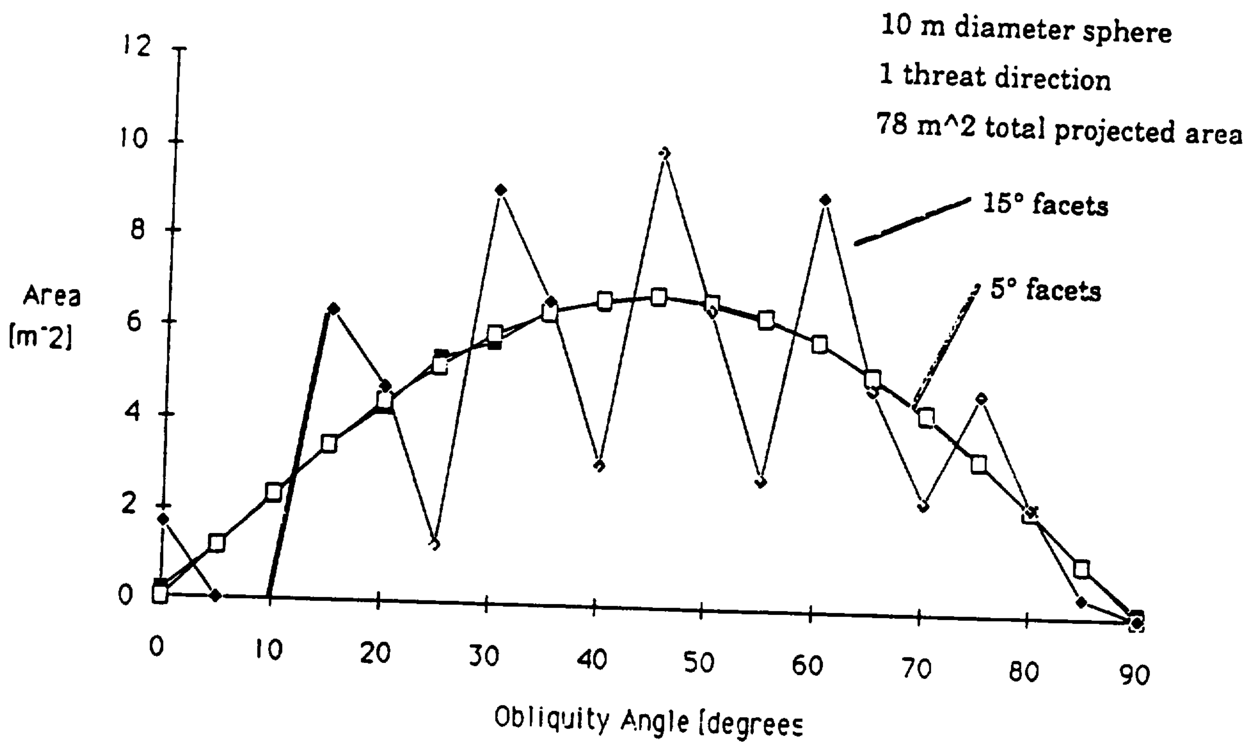


Fig. 13. Area Analysis of a sphere

```
RESPONSE PID:   1     RESPONSE FILE: ONE_RESPONSE.RSP
A_SURF FILE: PLATE_ON_EDGE.ASB
PNP(%)=   99.99709     Total Flux x Area x Time (NAT) = 0.29084E-04
CONTOURS .12345 at equal increments from 0 to max NAT = 0.20709E-05

                             IMPACT VELOCITY  km/s

Obl    1   2   3   4   5   6   7   8   9  10  11  12  13  14  15  16
Deg    I   I   I   I   I   I   I   I   I   I   I   I   I   I   I   I
 90 ................................................................
 85 ................................................................
 80 ................................................................
 75 ................................................................
 70 .......................................................111......
 65 .....................................................11111......
 60 ...................................................2111.........
 55 ...............................................14.11............
 50 ...........................................11.5111..............
 45 ........................................21.31.11................
 40 ....................................14..31......................
 35 .................................32.12..........................
 30 ..........................11.15..11.............................
 25 ......................21..51.11.................................
 20 ..................21..31........................................
 15 .............12...11............................................
 10 .....11..11..11.................................................
  5 .11..11..11.....................................................
  0 .11.............................................................
```

Fig. 14. P_SURF analysis of the flat plate in Fig. 12

```
RESPONSE OUTPUT FILE = 30_050MPIDS.RSP
         1
RESPONSE PID:  11     RESPONSE FILE: 30_050MPIDS.RSP
A_SURF FILE: MB17-ALL.ASB
PNP(%)=   99.88475     Total Flux x Area x Time (NAT) = 0.11532E-02
CONTOURS .12345 at equal increments from 0 to max NAT = 0.26100E-04

                           IMPACT VELOCITY  km/s

Obl    1   2   3   4   5   6   7   8   9  10  11  12  13  14  15  16
Deg    I   I   I   I   I   I   I   I   I   I   I   I   I   I   I   I
 90 ................................................................
 85 ................................................................
 80 ................................................................
 75 ..............................1......1..1.....1..1.1............
 70 .....................................................1.1........
 65 ........................................1..1..2..2.1.212........
 60 ......................1...2...1..1...1.....1..2..2.5132.........
 55 ..........................1.............1..21.3.15.41...........
 50 .................................11..3..2..21.51.3..............
 45 ..........................1...2..1......31.41.1.................
 40 .....................................5..3..........1............
 35 ..............................2..11..1...........1..............
 30 ................................................................
 25 .....................................1...........1.1............
 20 ..................1.....................1.......................
 15 ...........................................1..1..1.1............
 10 ..............................................1..1..............
  5 ...................................................1............
  0 ................................................................
```

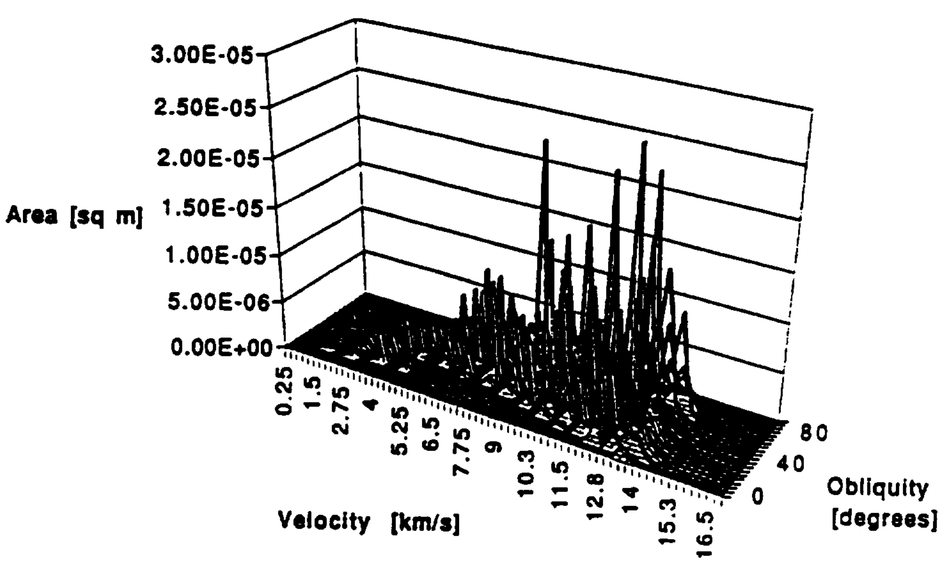


Fig. 15. P_SURF analysis of a SSF module
(1995 exposure environment).

Appendix A. P_SURF Source Code

P_SURF Listing

Listing from Language Systems FORTRAN (Version 3.0 Tue, Nov 19, 1991)
Sat, Sep 12, 1992   1:27 PM
Options OFF: A BKG=0 CASE CCD CCX CRAY DYN E EXTENDED F77 I2 LINEFEED MC68020 MC68040 MC68881 NOIMPLICIT OV R S SANE SYM T72 TRACE W X Z
Options ON: ANSI C L SAVEALL U VAX

```
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C                                     C
C       P_SURF VER 1.6  8/23/92       C
C                                     C
C          MARTIN MARIETTA            C
C        MANNED SPACE SYSTEMS         C
C                                     C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C
C
C  P_SURF VER 1.6 will compute the Probability of No Penetration (PNP) by
C  space debris for a designated area on a spacecraft.  P_SURF calculates
C  the flux (N) which penetrates the spacecraft multiplied by the exposed
C  area (A) and the exposure time (T) as a function of velocity & obliquity.
C  One data point on the surface represents the sum of all projected
C  areas that can be hit by a particle at a certain velocity and obliquity,
C  multiplied by the fraction of the total flux that will cause a
C  penetration and the exposure time.
C
C  P_SURF VER 1.6 works with BUMPERII Version 1.3
C
C  The code requires two files generated by other code as input.  One
C  output file is from the A_SURF code.  This file contains a selected
C  exposed area of a spacecraft, summarized in a matrix as a function of
C  velocity and obliquity.  The other required file is the output file of
C  the RESPONSE portion of the BUMPERII code.  This file contains the
C  ballistic limit (minimum diameter to penetrate) as a function of velocity
C  and obliquity.  Multiple shield designs may be included in the RESPONSE
C  output, and the PNP calculation may be performed for a specific shield
C  or for each shield in turn.
C
C  The RESREAD and FLUX subroutines are taken directly from BUMPERII
C  version 1.2a except for the COMMONPS.BLK instead of COMMON2.BLK
C  Other modules were modeled after BUMPER for continuity.
C  BUMPER was developed under the NASA contract 'Integrated Wall Design
C  Guide and Penetration Control Plan' by M.A.Wright & A.R.Coronado.
C
C  Note that peaks or waves in the area, flux or probability surfaces
C  may be artifacts produced by granularity in the spacecraft model
C  or threat models used in the GEOMETRY portion of BUMPER. Surface
C  contours reflect the way BUMPER interogates the ballistic limit
C  surface created by RESPONSE.
C
C  P_SURF code was developed under the NASA contract 'Structural Damage
C  Prediction and Analysis for Hypervelocity Impacts Study' under the
C  direction of N. Elfer.
C
```

```
C  Version 1.5 corrects an error in the PNP calculation (found by Ben
C  Hayashida). The FLUX from BUMPERII version 1.2a and 1.3 returns the
C  Debris flux for the old environment, but the flux times the exposure
C  time for the new environment.  The was not recognized in version 1.4.
C
C  Version 1.6 reads BUMPERII ver. 1.3 Response files
C
C  Include module COMMONPS variable list
C
C    alt = operating altitude , km
C    asfile = the output Area Surface filename
C    binc = impact angle (beta) increment , deg
C    conf = text description of wall configuration
C    diam = critical diameter , cm
C    etime = spacecraft exposure time , years
C    flx = number of impacts per projected area per year of diameter D
C           or larger
C    inclin = orbital inclination, degrees
C    idens = debris density, 1- constant density, 2-size function
C    ienv = environment type, 1- JSC 20001&6000, 2- 7/90 memo
C    it = current threat case
C    itype = analysis type , 1- debris, 2-meteoroids
C    nb = number of angles in the response array
C    nc = number of wall configurations in the response array
C    nee = the total number of exposed elements summed
C    nr = number of element ranges to sum over
C    nt = number of threat cases
C    nv = number of velocities in the response array
C    pid = the property id associated with all elements of the ranges
C    psfile = the Probability (Flux Area Time) Surface filename
C    rsfile = the Response Surface filename
C    sflevel = solar flux level
C    units = english or metric
C    vr = impact (relative) velocity , km/sec
C    vinc = impact (relative) velocity increment , km/sec
C
C  Arrays
C
C    area = array containing the value of the surface area for each
C       element, sq-meters
C    areas  = the area surface containing the summed area fractions for
C       each velocity and obliquity for all elements in the
C       specified element id ranges.    (vr,beta)
C    exposed  = list of the number of exposed elements for each threat
C       angle
C    fluxs    = array containing flux corresponding to the diameters
C       in the response surface (for each velocity and obliquity)
C    geometry = array containing the values of the cosine of the impact
C       angle for each exposed element for each threat angle.
C    id = array containing the values of the element and property id
C       for each element
C       1- id
C       2- pid
C    natmax = maximum Flux*Area*Time on one nats surface, [impacts]
```

```
C
C    nats = flux*area*time surface as a function of
C       (velocity,obliquity, pid), [impacts]
C    ner = array containing the range number for each element
C    pids = PID (see scalar) number to process
C    point = array of the element numbers corresponding to the elements
C       in the geometry array.
C    range = array containing the starting and ending elment id for each
C            range to sum over
C            1-starting id
C            2- ending id
C    response = array containing the values of the critical diameter as
C       a function of impact angle and velocity. (vr,beta,pid)
C    standm = shield stand-off, cm
C    shden = shield density, g/cc
C    shthkm = shield thickness, cm
C    tnat = total flux * area * time for each PID and the areas array
C    vwden = vessel wall density, g/cc
C    vwthkm = vessel wall thickness, cm
C
C
C  Main Program Variable List
C
C  Scalers
C
C    answer = user input
C    areae = the area times the threat probability
C    ob = Obliquity for the current threat/element.
C
C
C
      LOGICAL FIRST
C
      CHARACTER*80 ANSWER
C
      INTEGER*2 IC
C
C      REAL*4 PROB
C
      INCLUDE 'COMMONPS.BLK'
C
C  Initialize the Velocity increment and number of velocities
C
      VINC=0.25
      NV=68
C
C  Initialize the Obliquity increment and number of angles.
C
      BINC=5.0
      NB=19
C
C
C
      IBATCOM = 0
```

```
C
C  Write header to screen and read in orbital parameters
C
      CALL PSINPUT
C
C  Read in the A_SURF output file
C
      CALL ASREAD
C
C  Calculate the total effective exposure area.
C
        DO 10 I2=1,NB
          DO 10 I1=1,NV
            taeff = taeff + AREAS(I1,I2)
   10 CONTINUE
C
C
C
C  Read in the Solar flux data
C
      IF ( ISol.EQ.1.OR.ISol.EQ.2 )CALL SOLREAD
C
C
C  Read in the RESPONSE output file.  This is identical to
C  the RESREAD subroutine in BUMPER.
C
      CALL RESREAD
C
C  Verify the Response file has the same increments as the Area_Surface
C
      IF (BINC.NE.5.0 .OR. VINC.NE.0.25 ) THEN
         WRITE (6,*)'RESPONSE FILE HAS DIFFERENT FORMAT THAN AREA_SURF!'
         STOP
      ELSE
         CONTINUE
      ENDIF
C
C  Check array size and set to A_SURF size
C
      IF (NV.LT.68 .OR. NB.LT.19 ) THEN
         WRITE (6,*)'RESPONSE FILE IS SMALLER THAN AREA_SURF!'
         WRITE (6,*) NV,NB
         STOP
      ELSE
         NV=68
         NB=19
      ENDIF
C
C  Determine the RESPONSE PIDs to process.
C
C  If number of cases (NC) is only one then proceed.
C
      IF (NC.EQ.1) THEN
         PIDS(1)=1
```

```
              WRITE ( 6,20 )
        20    FORMAT (/1X,'The one case in the RESPONSE file will be used' )
      C
      C For multiple PIDs select one or all. If only one, NC is set to 1.
      C Write number of PIDs and first PID in A_SURF to screen.
      C
            ELSE
              WRITE ( 6,25 ) NC
        25    FORMAT (/1X,'The Number of PIDs in the RESPONSE file is  ',I4)
              WRITE ( 6,26 ) PID
        26    FORMAT (/1X,'The first PID processed by A_SURF was       ',I4)
      C
              WRITE ( 6,30 )
        30    FORMAT   (//1X,'Enter <CR> to use the A_SURF PID. '
         1             /1X,'Enter the PID number to use a specific PID.',
         2             /1X,'Enter <A> to use all PIDs.')
      C
              READ ( 5,35 ) ANSWER
        35    FORMAT (A)
              IF ( ANSWER(1:1).EQ.' ' ) THEN
                  PIDS(1) = PID
                  NC=1
              ELSE IF (ANSWER(1:1).EQ.'A' .OR. ANSWER(1:1).EQ.'a') THEN
                 DO 40 I1=1,NC
                    PIDS(I1) = I1
        40       CONTINUE
              ELSE
              READ ( ANSWER(1:80),45 )PIDS(1)
        45       FORMAT ( BN,I2 )
                 NC=1
              ENDIF
      C
            ENDIF
      C
      C
      C Calculate the Flux surface using the critical diameters from
      C the Response surface
      C
      C
            DO 120 I3=1,NC
              NATMAX(I3) = 0
              TNAT(I3) = 0
              DO 100 I2=1,NB
                DO 100 I1=1,NV
                  DIAM=RESPONSE(I1,I2,PIDS(I3))
                  CALL FLUX
                  FLUXS(I1,I2,I3) = FLX
      C
      C Calculate the FLUX x AREA x TIME surface
      C (NOTE that FLUX returned FLUX x TIME for the new environment)
      C
                  IF (IEnv.EQ.1) THEN
                   NATS(I1,I2,I3) = FLUXS(I1,I2,I3)*AREAS(I1,I2)*ETIME
                  ELSE
```

```
                  NATS(I1,I2,I3) = FLUXS(I1,I2,I3)*AREAS(I1,I2)
                 END IF
C
C  Calculate the total FLUX x AREA x TIME
C
                 TNAT(I3) = TNAT(I3)+NATS(I1,I2,I3)
C
C  Find the max NAT for the PID number
C
                 IF (NATMAX(I3).LT.NATS(I1,I2,I3) ) THEN
                     NATMAX(I3) =  NATS(I1,I2,I3)
                 ELSE
                     CONTINUE
                 ENDIF
C
C
C
  100   CONTINUE
C
C  Calculate PNP for Ranges in Exposure Surface
C
      PNP(I3) = (DEXP(-TNAT(I3)))*100.D0
C
  120 CONTINUE
C
C  Print out the carpet plot
C
      CALL CARPETPLOT
C
C  Print out the flux x area x time surface in a comma
C  delimited format to be read by spreadsheets.
C
      CALL NATTEXT
C
C
C
C
C  Close summary file
C
       CLOSE ( UNIT=10,STATUS='KEEP' )
       WRITE( 6,60003 ) PSFILE
60003  FORMAT( /' The PNP Surface file is complete.'/
     1        ' filename: ',A )

C
C  Finished
C
      END
C
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
      SUBROUTINE CARPETPLOT
C
```

```
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C  This subroutine takes the NAT surface and writes out a text
C  based carpet plot to the output file
C
C  ARRAY LIST
C
C   CARPET  = LINEAR ARRAY IN VELOCITY OF SYMBOLS FOR CARPET PLOT
C   NATINC  = LINEAR ARRAY OF INCREMENTS TO BREAK CARPET PLOT
C
C
C
C
        CHARACTER*1 CARPET(70)
        REAL*8 NATINC(6)
        INCLUDE 'COMMONPS.BLK'
C
      DO 2500 I3=1,NC
C
C  Calculate the increments in the carpet plot
C
C
        DO 1990 J=1,6
          NATINC(J)=J*NATMAX(I3)/6.D0
 1990   CONTINUE
C
C  Write out header information to screen and to file
C
        WRITE (10,2000) PIDS(I3), RSFILE
        WRITE ( 6,2000) PIDS(I3), RSFILE
 2000   FORMAT(1H1,/,1X,'RESPONSE PID: ',I3,5X,
     1          'RESPONSE FILE: ',A)
        WRITE (10,2005) ASFILE
        WRITE ( 6,2005) ASFILE
 2005   FORMAT(1X,'A_SURF FILE: ',A)
        WRITE (10,2010) PNP(I3),TNAT(I3)
        WRITE ( 6,2010) PNP(I3),TNAT(I3)
 2010   FORMAT(1X,'PNP(%)= ',F10.5,5X,
     1  'Total Flux x Area x Time (NAT) =',E12.5)
        WRITE (10,2020) NATMAX(I3)
        WRITE ( 6,2020) NATMAX(I3)
 2020   FORMAT(1X,'CONTOURS .12345 at equal increments from',
     1              ' 0 to max NAT =',E12.5)
C
        WRITE ( 10,2030 )
        WRITE (  6,2030 )
 2030   FORMAT ( /8X,19X,'IMPACT VELOCITY  km/s',/)
        WRITE ( 10,2040 ) (I,I=1,16,1)
        WRITE (  6,2040 ) (I,I=1,16,1)
 2040   FORMAT ( 1X,'Obl',1X,16I4 )
        WRITE ( 10,2050 )
        WRITE (  6,2050 )
 2050   FORMAT ( 1X,'Deg',1X,16('   I'))
C
```

```
C  Calculate the carpet plot
C
         DO 2500 I2=1,NB
            DO 2300 I1=1,NV
C
C  To plot with max obliquity at the top, I4 is substituted for I2
C
               I4=20-I2
C
               IF (NATS(I1,I4,I3) .LE. NATINC(1)) THEN
                        CARPET(I1)='.'
                    ELSE IF (NATS(I1,I4,I3) .LE. NATINC(2)) THEN
                        CARPET(I1)='1'
                    ELSE IF (NATS(I1,I4,I3) .LE. NATINC(3)) THEN
                        CARPET(I1)='2'
                    ELSE IF (NATS(I1,I4,I3) .LE. NATINC(4)) THEN
                        CARPET(I1)='3'
                    ELSE IF (NATS(I1,I4,I3) .LE. NATINC(5)) THEN
                        CARPET(I1)='4'
                    ELSE IF (NATS(I1,I4,I3) .LE. NATINC(6)) THEN
                        CARPET(I1)='5'
                ENDIF
 2300       CONTINUE
C
C  Write out the contour marks
C
C
        Write ( 6,2400) (I4-1)*5,(CARPET(I1),I1=1,64)
        Write (10,2400) (I4-1)*5,(CARPET(I1),I1=1,64)
 2400       FORMAT ( 1X,I3,1X,64A1 )
C
 2500 CONTINUE
C
      RETURN
C
      END
C
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
      SUBROUTINE NATTEXT
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C  Write NAT array for 5 degree and 0.25 km/sec increments
C
C
      INCLUDE 'COMMONPS.BLK'
      REAL*4 NATK(19)
C
C
C
C
        DO 700 I=1,NC
```

```
            WRITE ( 10,600 ) PIDS(I)
  600       FORMAT ( 1H1,/,1X'RESPONSE PID: ',I3,/ )
            WRITE ( 10,630 ) PIDS(I),(J,J=0,90,5)
  630       FORMAT ( 1X,I6.2,19(',',I12.2) )
C
               DO 690 K=1,NV
                  DO 640 J=1,19
                     NATK(J)=NATS(K,J,I)
  640             CONTINUE
                  VR=K*VINC
                  WRITE( 10,650) VR,(NATK(J),J=1,19)
  650             FORMAT ( F6.2,19(',',E12.4) )
  690          CONTINUE
  700       CONTINUE
C
            RETURN
C
            END
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
          SUBROUTINE RESREAD
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C
C  Resread reads in the output from the RESPONSE code.  This output
C  consists of the critical diameter data as a function of property
C  id, impact angle, and impact velocity.
C
C
C  note: for variables contained in the common block refer to the main
C        listing for definition.
C
C
C  Variable list
C
C    answer = character string represnting user input
C    itf = analysis type for rfile
C    rfile = response output filename
C    ienvr = environment for response file
C
          CHARACTER LENGTH*2
          CHARACTER*80 ANSWER,RFILE,Form
          CHARACTER*46 A46
          CHARACTER*15 B15A, B15B
          CHARACTER*8  C8A, C8B
          CHARACTER*2  D2
          REAL*4       STND
C          INTEGER*4 ITF, ITA, IC, ICT, ICB, IPF
C         !!! CHANGES TO BE COMPATIBLE WITH BUMPERII ver1.3
          INTEGER*2 ITF, ITA, IC, ICT, ICB, IPF, IPFUNC3
          INTEGER*2 IENVR
C
```

```
      INCLUDE 'COMMONPS.BLK'
      IF (IBOTHS.EQ.2) GOTO 60
C
C  Read in the RESPONSE output filename , set default to resp.dat
C
      IF (INDEX(ROOTFILE,'.') .EQ. 0)ROOTFILE='STATION.'
      ANSWER=ROOTFILE(1:INDEX(ROOTFILE,'.'))//'RSP'
      JOT = INDEX( ROOTFILE, '.' )
      WRITE (LENGTH, '(I2)' )JOT+3
      FORM='(/1X,''RESPONSE OUTPUT FILENAME (<CR>=''',A'//LENGTH//
     . ',''') > '',$)'
   10 WRITE ( 6,FORM )ANSWER
      READ ( 5,30 ) RFILE
   30 FORMAT (A)
C
C  !!!!  THIS OPEN FOR THE MAC WILL GIVE THE NORMAL FINDER DIALOG
C        BOX.  THE DIRLIST METHOD IS SKIPPED
      IF (RFILE(1:1).EQ.'?') THEN
          OPEN ( UNIT=23,FILE=*,STATUS='OLD',FORM='UNFORMATTED',ERR=40 )
          INQUIRE( UNIT=23,NAME=RFILE)
          GOTO 60
      END IF
C
C  !!!!  END OF MAC OPEN
C
      IF (RFILE(1:1).EQ.'?') THEN
          CALL DIRLIST
          GOTO 10
      END IF
      IF ( RFILE(1:4).EQ.'    ' ) THEN
         RFILE=ANSWER
      ELSE
         ROOTFILE = RFile(1:INDEX(RFile,'.'))
      ENDIF
C
      IF(IBATCOM.EQ.1) THEN
       WRITE(13,'(A)') RFILE
       RETURN
      END IF
C
C  Open the file
C
      OPEN ( UNIT=23,FILE=RFILE,STATUS='OLD',FORM='UNFORMATTED',ERR=40 )
C

      GO TO 60
C
C  Error control on open
C
   40 WRITE ( 6,50 )
   50 FORMAT ( /1X,'UNABLE TO OPEN FILE' )
      GO TO 10
C
C  Read in the analysis type and the number of property cases.
```

```
C
C   60 READ (23) ITYPEIN,ITF,IDens,NC
C   !!!  NO ERROR CHECKING ON IENVR
   60 READ (23) ITYPEIN,ITF,IENVR,IDens,NC
C       WRITE(6,*)'ITYPEIN,ITF,IDens,NC'
C       WRITE(6,*) ITYPEIN,ITF,IDens,NC
        IF (ITYPEIN.EQ.3.AND.IBOTHS.EQ.1) ITYPE=1
      IF (IDens.EQ.1) THEN
        WRITE (6,63)
   63   FORMAT (/5X,' Constant density threat')
      ELSE IF (IDens.EQ.2) THEN
        WRITE (6,64)
   64   FORMAT (/5X,' Variable density threat')
      END IF
C
C  Check that the response file is the correct analysis type
C
      IF ( ITF.NE.ITYPE ) THEN
         IF ( ITYPE.EQ.1 ) THEN
            WRITE ( 6,70 )
   70       FORMAT ( /1X,'DEBRIS ANALYSIS SPECIFIED IN GEOMETRY FILE ',
     1         'BUT RESPONSE FILE IS FOR METEOROIDS ')
         ELSE
            WRITE ( 6,80 )
   80       FORMAT (/1X,'METEOROID ANALYSIS SPECIFIED IN GEOMETRY FILE',
     1                   ' BUT RESPONSE FILE IS FOR DEBRIS' )
         END IF
C
         WRITE ( 6,90 )
   90    FORMAT ( /1X,'DO YOU WISH TO CONTINUE WITH GEOMETRY OPTION ',
     1               '(<CR>=NO) > ',$)
         READ ( 5,30 ) ANSWER
C
         IF ( ANSWER(1:1).EQ.'Y' .OR. ANSWER(1:1).EQ.'y' ) THEN
            GO TO 10
         ELSE
            STOP
         END IF
C
      END IF
C
C  Read in the impact angle information
C
      READ (23) NB,BINC
C       WRITE(6,*) 'NB,BINC'
C       WRITE(6,*) NB,BINC
C
C  Read in the impact velocity information
C
      READ (23) NV,VINC
C       WRITE(6,*) 'IMPACT VELOCITY, VEL INCR.'
C       WRITE(6,*) NV,VINC
C
C  Initialize RESPONSE to 0.0
```

```
C
      DO 200 I=1,NC
         DO 150 J=1,NB
            DO 100 K=1,NV
               RESPONSE ( K,J,I ) = 0.
  100       CONTINUE
  150    CONTINUE
  200 CONTINUE
C
C     Read in the critical diameter data
C
C        WRITE(6,*) 'NC,NB,NV'
C        WRITE(6,*) NC,NB,NV
C        WRITE(6,*) 'RESPONSE(K,J,I)'
C       Loop thru the property id's
        DO 400 I=1,NC
C
C         Loop thru the impact angles
          DO 300 J=1,NB
C
C           Loop thru the impact velocities
            DO 250 K=1,NV
C
C             Store the critical diameter in response
              READ (23) RESPONSE(K,J,I)
C              WRITE(6,*) RESPONSE(K,J,I)
  250       CONTINUE
  300     CONTINUE
  400   CONTINUE
C
C      IF (INPUTCD.EQ.2) CALL SETDIAMS
C
      READ ( 23,END=440,ERR=440 ) A46
C      WRITE(6,*) 'A46'
C      WRITE ( 6,'( //1X,A)' ) A46
      WRITE ( 10,'( //1X,A)' ) A46
      READ ( 23 ) C8A,ITA,C8B,ICB,UNITS
      WRITE ( 10,'( A,I4)') ' Threat (1 Debris, 2 Meteoroid)  ',ITA
      WRITE ( 10,'( A,I4)') ' Density (1 Constant, 2 Function)  ',IDens
      WRITE ( 10,'( A,I4)') '          Number of PID Cases    ',ICB
      WRITE ( 10,'( 2A)')   '          Units  ',UNITS
C      WRITE(6,*) 'C8A,C8B'
C      WRITE(6,*) C8A,C8B
C      WRITE ( 6,'( A,I4)') ' Threat (1 Debris, 2 Meteoroid)  ',ITA
C      WRITE ( 6,'( A,I4)') ' Density (1 Constant, 2 Function)  ',IDens
C      WRITE ( 6,'( A,I4)') '          Number of PID Cases    ',ICB
C      WRITE ( 6,'( 2A)')   '          Units  ',UNITS
      DO 420 I=1,ICB
        READ ( 23 ) ICT,D2,B15A,B15B,IPF,IPFUNC3
        WRITE ( 10,411) I
 411    FORMAT( /1X,'PID NUMBER ',I4 )
C
        IF (ICT.EQ.2) THEN
          IF ( IPF.EQ.1 ) THEN
```

```
              WRITE ( 10,485)
            ELSE IF ( IPF.EQ.2 ) THEN
              WRITE ( 10,486)
            ELSE IF ( IPF.EQ.3 ) THEN
              WRITE ( 10,487)
            ELSE IF ( IPF.EQ.4 ) THEN
              WRITE ( 10,488)
            ELSE IF ( IPF.EQ.5 ) THEN
              WRITE ( 10,484)
            ELSE IF ( IPF.EQ.6 ) THEN
              WRITE ( 10,489)
            ELSE IF ( IPF.EQ.7 ) THEN
              WRITE ( 10,490)
            ELSE IF ( IPF.EQ.8 ) THEN
              WRITE ( 10,491)
            ELSE IF ( IPF.EQ.9 ) THEN
              WRITE ( 10,492)
            ELSE IF ( IPF.EQ.10 ) THEN
              WRITE ( 10,493)
            ELSE IF ( IPF.EQ.11 ) THEN
              WRITE ( 10,494)
            ELSE IF ( IPF.EQ.12 ) THEN
              WRITE ( 10,495)
            ELSE IF ( IPF.EQ.13 ) THEN
              WRITE ( 10,496)
            ELSE IF ( IPF.EQ.14 ) THEN
              WRITE ( 10,497)
            END IF
          END IF
  485     FORMAT ( /1X,'ORIGINAL PENETRATION FUNCTION')
  486     FORMAT ( /1X,'PEN4 PENETRATION FUNCTION')
  487     FORMAT ( /1X,'REGRESSION PENETRATION FUNCTION')
  488     FORMAT ( /1X,'COUR-PALAIS PENETRATION FUNCTION')
  484     FORMAT ( /1X,'BOEING INTERP PENETRATION FUNCTION')
  489     FORMAT ( /1X,'DEVELOPMENTAL6, USER INPUT')
  490     FORMAT ( /1X,'DEVELOPMENTAL7, USER INPUT')
  491     FORMAT ( /1X,'DEVELOPMENTAL8, USER INPUT')
  492     FORMAT ( /1X,'DEVELOPMENTAL9, USER INPUT')
  493     FORMAT ( /1X,'DEVELOPMENTAL10, USER INPUT')
  494     FORMAT ( /1X,'DEVELOPMENTAL11, USER INPUT')
  495     FORMAT ( /1X,'DEVELOPMENTAL12, USER INPUT')
  496     FORMAT ( /1X,'DEVELOPMENTAL13, USER INPUT')
  497     FORMAT ( /1X,'DEVELOPMENTAL14, USER INPUT')
          WRITE ( 10,'( /A )' ) ' Configuration     Shield       Wall'
C          WRITE ( 6,* ) 'ICT,D2,B12A,B12B'
C          WRITE ( 6,* ) ICT,D2,B12A,B12B
          IF (ICT.EQ.1) CONF = 'Single Plate'
          IF (ICT.EQ.2) CONF = 'Double Plate'
          IF (ICT.EQ.3) CONF = 'Multiwall'
          WRITE ( 10,'( 1X,A,4X,2A )' ) CONF,B12A,B12B
C           WRITE ( 6,'( 1X,A,4X,2A )' ) CONF,B12A,B12B
          READ ( 23 ) ShThk,VWThk,STND,ShDen(I),VWDen(I),ADEN
C          WRITE ( 6,* ) 'ShThk,VWThk,STND,ShDen(I),VWDen(I),ADEN,I'
C          WRITE ( 6,* ) ShThk,VWThk,STND,ShDen(I),VWDen(I),ADEN,I
```

```
             IF (ICT.EQ.3) THEN
                WRITE ( 10,'( A,A,F8.4)') '       Combined Areal Density',
     +                                   ' of All Shields  = ',ADEN
                WRITE ( 10,'( A,F8.4)') '       Total Standoff = '
     +                             ,STND
C                WRITE ( 6,'( A,A,F8.4)') '       Combined Areal Density',
C     +                                   ' of All Shields  = ',ADEN
C                WRITE ( 6,'( A,F8.4)') '       Total Standoff = '
C     +                             ,STND
                GOTO 410
             END IF
C             WRITE ( 6,'( A,F8.4)') '       Shield Thickness = ',ShThk
             IF (SHTHK.NE.0.0)
     +         WRITE ( 10,'( A,F8.4)') '       Shield Thickness = ',ShThk
 410         WRITE ( 10,'( A,F8.4)') '  Vessel Wall Thickness = ',VWThk
C             WRITE ( 6,'( A,F8.4)') '  Vessel Wall Thickness = ',VWThk
             IF (ICT.NE.3) THEN
             IF (SHTHK.NE.0.0.AND.STND.NE.0.0)
     +         WRITE ( 10,'( A,F8.4)') '               Standoff = ',STND
C             WRITE ( 6,'( A,F8.4)') '               Standoff = ',STND
             END IF
             IF ( Units .EQ. ' ENGLISH    ' ) THEN
               ShThkM(I) = ShThk*2.54
               VWThkM(I) = VWThk*2.54
               ADAR(I)=ADEN/.0142233
              ELSE
               ShThkM(I) = ShThk
               VWThkM(I) = VWThk
               ADAR(I)=ADEN
             END IF

C            With or without 30 MLI
             READ ( 23 ) A46
             WRITE ( 10,'( 4X,A)' ) A46
C             WRITE ( 6,'( 4X,A)' ) A46
  420 CONTINUE
      GO TO 450
  440 WRITE ( 10,42 )
   42 FORMAT ( /2X,' No Header following .RSP file ' )
C
  450 IF (IBOTHS.EQ.1) RETURN
C
C     Close the file and return
C
      CLOSE ( UNIT=23,STATUS='KEEP' )
C
C     Write Rfile to summary file
C
      WRITE ( 10,500 )RFILE
  500 FORMAT(1X,'RESPONSE OUTPUT FILE = ',A )
C
C
C  !!!!
      WRITE(RSFILE,'(BN,A)')RFILE
```

```
      RETURN
C
      END
C
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C                              C
      Subroutine ASREAD
C                              C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C  This subroutine opens and reads the table of exposed areas
C  versus velocity and obliquity created by A_SURF.
C
C
       CHARACTER*80 ANSWER
C
C
C
      INCLUDE 'COMMONPS.BLK'
C
C  Read in the ASF filename , set default to DATA.ASB
C
   10 WRITE ( 6,'(/1X,''Area_Surface Binary Output File''
     1          '' <CR=DATA.ASB> :'',$)')
      READ ( 5,'(A)' ) ANSWER
C
C  !!!!  THIS OPEN FOR THE MAC WILL GIVE THE NORMAL FINDER DIALOG
C        BOX.  THE DIRLIST METHOD IS SKIPPED
      IF (ANSWER(1:1).EQ.'?') THEN
          OPEN ( UNIT=2,FILE=*,STATUS='OLD',FORM='UNFORMATTED',
     *           READONLY,ERR=10 )
          INQUIRE( UNIT=2,NAME=ASFILE)
          GOTO 40
      END IF
C
C  !!!!  END OF MAC OPEN
C

      IF ( ANSWER(1:1).EQ.' ' ) ANSWER='DATA.ASB'
C
      WRITE(ASFILE,'(BN,A)')ANSWER
C
C  Open the file
C
      OPEN ( UNIT=2,FILE=ANSWER,STATUS='OLD',FORM='UNFORMATTED'
     *             ,READONLY,ERR=10 )
C
   40 CONTINUE
C
C  Read in the analysis type and the number of ranges
C
      READ (2) ITYPE,NR,PID,AREATOT
      IF( NR.GT.IRNGS ) THEN
```

```
          WRITE( 6,60001 )
60001     FORMAT( /' ---ERROR--- The maximum number of Ranges was'
     *           ,' exceeded.' )
          STOP
        ENDIF
C
C  Read in the ranges
C
        READ (2) ((RANGE(I,J),I=1,2),J=1,NR)
C
C  Read in the impact angle information
C
        READ (2) NB,BINC
C
C  Read in the impact velocity information
C
        READ (2) NV,VINC
C
C  Read the Area Surface array
C
           READ (2) ((AREAS(I,J),I=1,NV),J=1,NB)
C
C  Close the file
C
        CLOSE ( UNIT=2,STATUS='KEEP' )
C
C
C
C  Write A_SURF file to output file
C
        WRITE ( 10,600 ) ASFILE
  600 FORMAT ( 1X,'A_SURF BINARY OUTPUT FILE = ',A )
C
C  Write the number of ranges and the Property ID.
C
        WRITE( 10,621 ) NR,PID,AREATOT
  621 FORMAT( 1X,'RANGES=',I2,'   PID=',I9,
     1   '    EFF. AREA (sq.m) =',F12.5)
C
C      Write the start and end Element ID for each range.
C
        DO 625 I=1,NR
          WRITE ( 10,622 ) I,RANGE(1,I),RANGE(2,I)
  622     FORMAT(1X,'Range ',I2,'  START: ',I12,'END: ',I12)
  625 CONTINUE
C
          RETURN
C
        END
C
C
C
C
C
```

```
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
      SUBROUTINE PSINPUT
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C
C  PSINPUT writes the program header to the screen and reads in the
C  summary ouput filename. It also determines the spacecraft exposure
C  time and operating altitude.
C
C
C
C  note: for variables contained in the common block refer to the main
C        listing for definition
C
C  Variable list
C
C    answer = character string representing user input
C    Psfile = output filenanme
C
C
C
      INCLUDE 'COMMONPS.BLK'
C
      CHARACTER*20 BUMTTM
      CHARACTER*80 ANSWER
C
C

C
C  Write header to screen and summary file
C
      WRITE ( 6,10 )
   10 FORMAT (/1X,'****************************',//1X,3X,
     1            'Space Debris SURFace',
     2     //1X,5X,'Ver. 1.5 8/23/92',/1X,5X,'for BUMPERIIv1.2a',//1X,
     3            '****************************')
C
C  Read in output filename, set default to SDSURF.PS
C
   15 WRITE ( 6,20 )
   20 FORMAT ( /1X,'OUTPUT FILENAME (CR=SDSURF.PS)>',$)
      READ ( 5,30 )PSFILE
   30 FORMAT (A)
C
      IF ( PSFILE(1:1).EQ.' ' ) PSFILE='SDSURF.PS'
C
C  Open psfile
C
C  !!!! PUT CREATOR='XCEL' OR 'MSWD' IN OPEN STATEMENTS ON MAC
C
      OPEN ( UNIT=10,FILE=PSFILE,STATUS='NEW',IOSTAT=IER,
     *       CREATOR='XCEL',ERR=40,RECL=256 )
```

```
                     Δ
### FORTRAN - Warning - This feature is an extension to VAX FORTRAN
    File "macii_p_surf16.f"; Line 912
#-------------------------------------------------------------
C
        GO TO 70
C
C  Error control
C
   40 IF ( IER.EQ.2013 ) THEN
         WRITE ( 6,50 )
   50    FORMAT ( /1X,'FILE ALREADY EXISTS OK TO OVERWRITE (CR=YES,$)>')
         READ ( 5,30 ) ANSWER
C
         IF ( ANSWER(1:1).EQ.'Y' .OR. ANSWER(1:1).EQ.' ' ) THEN
            OPEN ( UNIT=10,FILE=PSFILE,STATUS='UNKNOWN',IOSTAT=IER,
     1             ERR=40)
            REWIND 10
         ELSE
            GO TO 15
         END IF
      ELSE
         WRITE ( 6,60 )
   60    FORMAT (/1X,'UNABLE TO OPEN FILE ' )
         GO TO  15
      END IF
C
   70 CONTINUE
C
C
      WRITE ( 10,75 )
   75 FORMAT (/1X,'*****************************',//1X,3X,
     1            'Space Debris SURFace',
     2   //1X,5X,'Ver. 1.5 8/23/92',/1X,5X,'for BUMPERIIv1.2a',//1X,
     3            '*****************************'//)
C
C
C  Set analysis type to 1 (debris)
C
         ITYPE=1
C
C
C  !!!!!!!!!!FROM BUMPERII Ver1.2.a SHIELD INPUT!!!!!!!!!!
C
C  Determine Environment Definition, set default to 1 (original)
C
   51 WRITE ( 6,52 )
   52 FORMAT (/1X,'ENVIRONMENT ?',/2X,'1-JSC 20001&6000 <CR> ',/2X,
     1        '2- 7/90 MEMO',/1X,'ANSWER 1 OR 2 > ',$)
C
      READ ( 5,53 ) ANSWER
   53 FORMAT (A)
C
      IF ( ANSWER(1:4).EQ.'    ' ) THEN
```

```
         IEnv=1
      ELSE
         READ ( ANSWER(1:80),54 )IEnv
   54    FORMAT ( BN,I1 )
      END IF
C
C  Check that input was correct
C
      IF ( IEnv.EQ.1 .OR. IEnv.EQ.2 ) THEN
         CONTINUE
      ELSE
         WRITE ( 6,956 )
  956    FORMAT ( /1X,'INCORRECT INPUT' )
         GO TO 51
      END IF

  365 CONTINUE
      IF ( ITYPE.EQ.1.AND.IENV.EQ.2 ) THEN
  370    WRITE ( 6,380 )
  380    FORMAT (/1X,'SOLAR FLUX LEVEL ?',/,2X,'1-NOMINAL <CR> ',/,2X,
     1        '2-MINIMUM',/,2X,'3-CONSTANT',/,1X,'ANSWER 1-3 > ',$)
         READ ( 5,30 ) ANSWER
C
         IF ( ANSWER(1:4).EQ.'    ' ) THEN
            ISOL=1
         ELSE
            READ ( ANSWER(1:80),90,ERR=370 ) ISOL
   90       FORMAT (BN,I4)
         END IF
         IF(IBATCOM.EQ.1) WRITE(13,'(A)') ANSWER
         IF ( ISOL.EQ.3 ) THEN
  385       WRITE ( 6,390 )
  390       FORMAT (/1X,'SOLAR FLUX LEVEL (10**4 Jy) (<CR>=70) > ',$)
            READ ( 5,30 ) ANSWER
            IF ( ANSWER(1:4).EQ.'    ' ) ANSWER='70.0'
            READ ( ANSWER(1:80),120,ERR=385 ) SFLEVEL
            IF(IBATCOM.EQ.1) WRITE(13,'(A)') ANSWER
         END IF
C
C  Check that input was correct
C
         IF ( ISOL.LT.1 .OR. ISOL.GT.3 ) THEN
            GO TO 370
         END IF
      END IF
C
C  Determine the spacecraft exposure date, set default to 1995
C
      IF ( ITYPE.EQ.1.AND.IENV.EQ.2 ) THEN
  340    WRITE ( 6,350 )
  350    FORMAT ( /1X,'DATE TO BEGIN EXPOSURE ( 1994-2025 )
     1 (<CR>=1995) > ',$)
         READ ( 5,30 ) ANSWER
C
```

```
            IF ( ANSWER(1:4).EQ.'    ' ) THEN
               DATE=1995.
            ELSE
               ISpot=Index(ANSWER,'.')
               IF (ISPOT.NE.0) GOTO 179
               k=80
               iblank=0
               do while (iblank.eq.0)
                 if (ANSWER(k:k).ne.' ') then
                   iblank=1
                   goto 1110
                 end if
                 k=k-1
 1110           continue
               end do
               ANSWER=ANSWER(1:K)//'.'
  179          READ ( ANSWER(1:80),180,ERR=340 ) DATE
  180          FORMAT ( BN,D20.3 )
            END IF
C
C  Check that date is within range
C
            IF ( DATE.LT.1994 .OR. DATE.GT.2025 ) THEN
               WRITE ( 6,360 )
  360          FORMAT ( 1X,'---ERROR--- Date outside of range' )
               GO TO 340
            END IF
            IF(IBATCOM.EQ.1) WRITE(13,*) DATE
         END IF
         IF (IBOTHS.EQ.2) GOTO 56
C
C  Determine the spacecraft exposure time , set default to 10 years
C
  105 WRITE ( 6,110 )
  110 FORMAT (/1X,'SPACE STATION EXPOSURE TIME (YEARS)
     1 (<CR>=10.0) > ',$)
      READ ( 5,30 ) ANSWER
C
      IF ( ANSWER(1:4).EQ.'    ' ) ANSWER='10.0'
C
      READ ( ANSWER(1:80),120,ERR=105 ) ETIME
  120 FORMAT ( BN,D20.0 )
      IF (ETIME.LT.0.) GOTO 105
      IF(IBATCOM.EQ.1) THEN
        WRITE(13,*) ETIME
        GOTO 151
      END IF
C
   56 IF ( IEnv.EQ.1 ) THEN
        WRITE (10,57)
   57     FORMAT(' JSC-20001 AND JSC-6000 FLUX EQUATIONS')
      ELSE
        WRITE (10,58)
   58     FORMAT(' 7/17/90 MEMO FLUX EQUATIONS')
```

```
         ENDIF
         IF ( ITYPE.EQ.1 ) THEN
            WRITE ( 10,130 )
  130       FORMAT ( /1X,'MAN-MADE ORBITAL DEBRIS ANALYSIS')
            IF ( ISOL.EQ.1 ) WRITE ( 10,400 )
  400       FORMAT ( 1X,'NOMINAL SOLAR FLUX LEVEL' )
            IF ( ISOL.EQ.2 ) WRITE ( 10,410 )
  410       FORMAT ( 1X,'MINIMUM SOLAR FLUX LEVEL' )
            IF ( ISOL.EQ.3 ) WRITE ( 10,420 ) SFLEVEL
  420       FORMAT ( 1X,'SOLAR FLUX LEVEL          = ',F8.3 )
            IF (DATE.NE.0.) WRITE ( 10,430 ) DATE
  430       FORMAT ( 1X,'DATE TO BEGIN EXPOSURE  = ',F8.3 )
         ELSE
            WRITE ( 10,140 )
  140       FORMAT ( 1X,'METEOROID ANALYSIS' )
         END IF
C
         WRITE ( 10,150 )ETIME
  150 FORMAT ( 1X,'SPACECRAFT EXPOSURE TIME (YEARS) =',F8.3 )
C
C  Read in operating altitude , set default to 500 km
C
  151 IF (IBOTHS.EQ.2) GOTO 203
         IF( IEnv.EQ.1) THEN
           AltMin = 350.
           AltMax = 550.
           AltMinnm=350.*0.53995680
           AltMaxnm=550.*0.53995680
         ELSE
           AltMin = 100.
           AltMax = 500.
           AltMinnm=100.*0.53995680
           AltMaxnm=500.*0.53995680
         END IF
C
C        ALT INTERNALLY IS IN KILOMETERS.
C
  160 WRITE ( 6,170 )AltMin,AltMax
  170 FORMAT(/1X,'OPERATING ALTITUDE('F4.0'-',F4.0'km)
     1 (<CR>=388.92) ')
      WRITE(6,205)
 205  FORMAT(' OR ENTER AN "E" OR "e" TO ENTER IN NMILES > ',$)
      READ (5,'(A)')ANSWER
      IF (ANSWER(1:4).EQ.'    ') THEN
        ALT=388.92D0
        ALTNM=210.00D0
        IF (IBATCOM.EQ.1) WRITE ( 13,* ) ALT
        GOTO 189
      END IF
      IF (ANSWER.EQ.'E'.OR.ANSWER.EQ.'e') THEN
         IF(IBATCOM.EQ.1) WRITE(13,'(A)') ANSWER(1:10)
         WRITE(6,171)altminnm,altmaxnm
  171    FORMAT(/1X,'OPERATING ALTITUDE('F4.0'-',F4.0'nmiles)
     1   (<CR>=210.00) > ')
```

```
            READ (5,'(A)')ANSWER
            IF (ANSWER(1:4).EQ.'    ') THEN
              ALTNM=210.00D0
            ELSE
              READ (ANSWER(1:80),215) ALTNM
            END IF
            IF (IBATCOM.EQ.1) WRITE ( 13,* ) ALTNM
            ALT = ALTNM / 0.53995680
         ELSE
            READ(ANSWER(1:80),215) ALT
  215       FORMAT( BN,D20.0 )
            IF (IBATCOM.EQ.1) WRITE ( 13,* ) ALT
         END IF
C        WRITE(6,*) 'ALT',ALT

C
C  Check that altitude is within range
C
  189 IF ( ALT.LT.AltMin .OR. ALT.GT.AltMax ) THEN
            WRITE ( 6,190 )AltMin,AltMax
  190       FORMAT ( 1X,'---ERROR--- Altitude outside of range ',2F8.3 )
            GO TO 160
         END IF
C
C  Write altitude to output file
C
  203 IF (IBATCOM.EQ.1) GOTO 204
         IF ( Units .EQ. ' ENGLISH    ' ) THEN
           IF ( ANSWER.NE.'E'.OR.ANSWER.NE.'e') ALTNM=ALT* 0.53995680
           WRITE ( 10,202 ) ALTNM
  202      FORMAT ( 1X,'OPERATING ALTITUDE (nmiles) = ',F8.3 )
           WRITE ( 10,200 ) ALT
         ELSE
           WRITE ( 10,200 ) ALT
  200      FORMAT ( 1X,'OPERATING ALTITUDE (km) = ',F8.3 )
         END IF

C
C
  204     CONTINUE
C  Finished
C
         RETURN
C
         END
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
         SUBROUTINE FLUX
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C
C  Flux calculates the meteoroid or debris flux for the given critical
```

```
C  diameter based on analysis type.
C
      INCLUDE 'COMMONPS.BLK'
C
      IF (IEnv.EQ.1) THEN
C        This flux definition uses JSC-200001 for debris and JSC-6000 for
C        meteoroids
         CALL Flux20001
      ELSE
C        This flux definition uses the 7/17/90 revision memo to SSP 30425
         CALL Flux790
      END IF

      RETURN
C
      END
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
      SUBROUTINE FLUX20001
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C
C  Flux calculates the meteoroid or debris flux for the given critical
C  diameter based on analysis type.
C
C
C  note: for variables contained in the common block referr to the main
C        listing for definition
C
C  Variable List
C
C   ddiam = diam in double precision , cm
C   ge = gravity focusing factor
C   intercept = intercept of the flux equation
C   mass = critical meteoroid mass, g
C   mden = meteoroid density, g/cc
C   re = earth's radius, km
C   slope = slope of the flux equation
C
C
      INCLUDE 'COMMONPS.BLK'
C
      REAL*8 DDIAM,GE,INTERCEPT,LD,MASS,MDEN,PI,RE,SLOPE
C
      PARAMETER (PI=3.141592653589793238D0)
C
C  Set  mden
C
C
      MDEN=0.50D0
C
C  Calculate the focusing factor, equation
C  is from JSC-30000
```

```
C
      RE=6378.0D0
      GE=0.568D0+0.432D0*(RE/(RE+ALT))
C
C Convert diam to double precision
C
      DDIAM=DIAM
C
C Calculate the flux
C
      IF ( ITYPE.EQ.1 ) THEN
C
C For debris use JSC-20001, use stated equations for diameters
C less then 1 cm , for those greater use third order fit of the
C curve for region up to 5 cm .
C
C The log of the flux varies linearly between 400 and 500 km according
C to D Kesseler of JSC.
C
         LD=DLOG10(DDIAM)
         IF ( DIAM.GT.5.0 )DIAM=5.0
         IF ( DIAM.LE.5.0 ) THEN
            IF ( DIAM.LT.1.0 ) THEN
               SLOPE=-0.0010D0*ALT-2.0200D0
            ELSE
               SLOPE=-0.0022D0*ALT-0.1400D0
            END IF
            INTERCEPT=+0.0036D0*ALT-7.26D0
            FLX=10.0D0**(SLOPE*LD+INTERCEPT)
         ELSE
            WRITE ( 6,100  )
  100       FORMAT ( /1X,'DIAMETER IS GREATER THAN 5 CM LIMIT')
            STOP
         END IF
C
      ELSE
C
C For meteoroids use JSC-3000,  E-06g < mass < 1g
C
         MASS=PI*(DDIAM**3)/6.0D0*MDEN
         FLX=10.0D0**(-14.37D0-1.213D0*DLOG10(MASS))
C
C Account for earth shielding and gravity focusing , also convert to
C number of impacts per sq-m per year
C
         FLX=FLX*GE*3.15576D07
C
      END IF
C
C Correct Flux for difference in Boeing and Nasa definition
C
      FLX=FLX*4.0D0
C
      RETURN
```

```
C
      END

C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
      SUBROUTINE FLUX790
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C
C  Flux calculates the meteoroid or debris flux for the given critical
C  diameter based on analysis type.
C
C
C  note: for variables contained in the common block refer to the main
C        listing for definition
C
C  Variable List
C
C   ddiam = diam in double precision , cm
C   ge = gravity focusing factor
C   intercept = intercept of the flux equation
C   mass = critical meteoroid mass, g
C   mden = meteoroid density, g/cc
C   re = earth's radius, km
C   slope = slope of the flux equation
C
C
      INCLUDE 'COMMONPS.BLK'
C
      REAL*8 DEBFLUX,METFLUX
C
      INTEGER CURYR
C
C  Calculate the flux
C
      IF ( ITYPE.EQ.1 ) THEN
C
C        Sum flux over integral years
C
         FLX = 0.
         CURYR = INT(DATE) - 1
         IF ( ETIME.GE.1. ) THEN
            DO I = 1, ETIME
               CURYR = CURYR + 1
               FLX = FLX + DEBFLUX(CURYR,1)
            END DO
         END IF
C
C        Add fractional year if any
C
         CURYR = CURYR + 1
         FLX = FLX + DEBFLUX(CURYR,1) * (ETIME - INT(ETIME))
```

```
C
      ELSE
         FLX = METFLUX(DIAM,ALT) * ETIME
      END IF
C
      RETURN
C
      END
C
C
C
      DOUBLE PRECISION FUNCTION DEBFLUX(YEAR,MONTH)
C
C     FUNCTION WHICH COMPUTES THE FLUX FOR THE NEW DEBRIS
C     ENVIRONMENT DEFINITION (SEPTEMBER 17, 1990 REVISION
C     MEMO TO SSP 30425)
C
C     DEBFLUX - Flux (impacts per square meter per year)
C     HH      - Flux factor
C     DIAM    - Orbital debris diameter (CM)
C     ALT     - Altitude in kilometers
C     PSI     - Flux enhancement factor
C     YEAR    - YEAR (Year Date i.e. 1994...)
C     SFLUX   - 13 Month smoothed solar radio flux F10.7 units are
C               Expressed in 10**4 Jy; Retarded by 1 year from YEAR
C     AGROWTH - Assumed annual growth rate of mass in orbit
C     FGROWTH - Estimated growth rate of fragment mass
C
      INCLUDE 'COMMONPS.BLK'
C
      REAL HH,PHI1,PHI,F1,F2,P,G1,G2,PSI,TABLE1(25:125)
      INTEGER YEAR,MONTH,HI
C
      REAL*4 SFLUX(12,ISTART:ILAST,2)
C
      COMMON / SOLDAT / SFLUX
C
      DATA TABLE1
     1    /0.900,0.905,0.910,0.912,0.915,0.920,0.922,0.927,0.930,0.935,
     2     0.940,0.945,0.950,0.952,0.957,0.960,0.967,0.972,0.977,0.982,
     3     0.990,0.995,1.000,1.005,1.010,1.020,1.025,1.030,1.040,1.045,
     4     1.050,1.060,1.065,1.075,1.080,1.090,1.100,1.115,1.130,1.140,
     5     1.160,1.180,1.200,1.220,1.240,1.260,1.290,1.310,1.340,1.380,
     6     1.410,1.500,1.630,1.680,1.700,1.710,1.700,1.680,1.610,1.530,
     7     1.490,1.450,1.410,1.390,1.380,1.370,1.380,1.400,1.440,1.500,
     8     1.550,1.640,1.700,1.750,1.770,1.780,1.770,1.750,1.720,1.690,
     9     1.660,1.610,1.560,1.510,1.460,1.410,1.380,1.350,1.320,1.300,
     1     1.280,1.260,1.240,1.220,1.200,1.180,1.165,1.155,1.140,1.125,
     2     1.110/
C
C  Calculate psi
      IF ( INCLIN.LE.25 ) THEN
         PSI = TABLE1(25)
      ELSE
```

```
            LOW = INT(INCLIN)
            HI = LOW + 1
            PSI = (TABLE1(HI)-TABLE1(LOW)) * (INCLIN-LOW) + TABLE1(LOW)
         END IF
C
C Compute flux factor H and PHI
         HH = SQRT( 10**(EXP(-(ALOG10(DIAM)-0.78)**2/0.405769)) )
         IF ( ISOL.LE. 2 ) THEN
            PHI1 = 10.0**( ALT/200. - SFLUX(MONTH,YEAR-1,ISOL)/140. - 1.5 )
         ELSE
            PHI1 = 10.0**( ALT/200. - SFLEVEL/140. - 1.5 )
         END IF
         PHI = PHI1 / ( PHI1 + 1 )
C
         F1 = 1.22E-5 * (DIAM**-2.5)
         F2 = 8.1E+10 * (DIAM + 700.0)**-6
C
         AGROWTH = 0.05
         FGROWTH = 0.02
C Power relationship used with estimated growth rate of fragment mass
C        g1 = (1 + q)^(t - 1988)
         IF ( YEAR.LE.2010 ) THEN
            G1 = (1.0 + FGROWTH)**(YEAR - 1988)
         ELSE
            G1 = (1.0 + FGROWTH)**(2010 - 1988)
            FGROWTH = 0.04
            G1 = G1 * (1.0 + FGROWTH)**(YEAR - 2010)
         END IF
C Linear relationship used with assumed annual growth rate of orbit mass
C        g2 = 1 + p(t - 1988)
         G2 = 1.0 + AGROWTH*(YEAR - 1988)
C Calculate debris flux function based on various factors
C and compute cross sectional flux in lieu of surface area flux
         DEBFLUX = 4.0D0 * HH * PHI * PSI * (F1*G1 + F2*G2)
C
         RETURN
         END
C
C
C
         DOUBLE PRECISION FUNCTION METFLUX(DIAM,ALT)
C
C        FUNCTION WHICH COMPUTES THE FLUX FOR METEOROID
C        ENVIRONMENT DEFINITION (FROM SEPTEMBER 17, 1990
C        REVISION TO SSP 30425)
C
C        METFLUX - FLUX (PARTICLES/SQUARE METER/YEAR)
C        DIAM    - DIAMETER OF METEOROID (CM)
C        MASS     - MASS OF METEOROID (GRAMS)
C
         REAL DIAM
         DOUBLE PRECISION ALT,TEMP,MASS,PI
C
         PARAMETER (PI = 3.141592653589793D0)
```

```
      PARAMETER (C0 = 3.147E+7)
      PARAMETER (C1 = 2.2E+3  )
      PARAMETER (C2 = 15.0    )
      PARAMETER (C3 = 1.3E-9  )
      PARAMETER (C4 = 1.0E+11 )
      PARAMETER (C5 = 1.0E+27 )
      PARAMETER (C6 = 1.3E-16 )
      PARAMETER (C7 = 1.0E+6  )
C
      RE = 6478.0D0
C Determine gravity focusing factor
      GE = 1.0 + RE/(RE + ALT - 100.)
C Determine meteoroid mass based on diameter of particle and dens=.5g/cc
      MASS = PI*(DIAM**3)/6.0*.5
C Compute terms defined in Sept 1990 draft for SSP 30425 update
C to meteoroid environment
      TERM = MASS*MASS
      TEMP = C0*( (C1*MASS**0.306 + C2)**-4.38 + C3*(MASS + C4*TERM +
     1       C5*TERM*TERM)**-0.36 + C6*(MASS + C7*TERM)**-0.85)
C
C Convert to cross sectional flux; use gravity focusing factor GE
C
      METFLUX = 4.0D0*GE*TEMP
C
      RETURN
      END


CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
      SUBROUTINE SOLREAD
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C
C  Solread reads in the output solar flux file.
C
C
C  note: for variables contained in the common block refer to the main
C        listing for definition.
C
C
C  Variable list
C
C    answer = character string represnting user input
C    rfile = response output filename
C
C
      INCLUDE 'COMMONPS.BLK'
C
      CHARACTER*80 ANSWER,RFILE
C
      REAL*4 SFLUX(12,ISTART:ILAST,2)
C
```

```
      COMMON / SOLDAT / SFLUX
C
C  Open the SOLAR_FLUX.DAT file and read it
C
      OPEN ( UNIT=7,FILE='SOLAR_FLUX.DAT',STATUS='OLD',READONLY,ERR=100)
C
      GO TO 200
C
C  Error control for open
C
  100 WRITE ( 6,110 )
  110 FORMAT ( /1X,'DEBRIS VELOCITY DISTRIBUTION FILE SOLAR_FLUX.DAT ',
     1             'WAS NOT FOUND'/' FILENAME ? > ',$)
      READ ( 5,60 ) ANSWER
   60 FORMAT (A)
C
      IF ( ANSWER(1:2).EQ.'  ' ) GO TO 100
C
      OPEN ( UNIT=7,FILE=ANSWER,STATUS='OLD',READONLY,ERR=100 )
C
      IF (IBATCOM.EQ.1) THEN
        WRITE(13,'(A)') ANSWER
        GOTO 251
      END IF
C
C  Search for data while reading through comment lines
C
  200 CONTINUE
      READ ( 7,60,END=250 )ANSWER
      READ ( ANSWER,1000,ERR=200 )IYR,IMON,(SFLUX(IMON,IYR,I),I=1,2)
 1000 FORMAT ( I4,3X,I2,3X,5X,2(3X,F5.1))
C
C  Read the data
C
      DO 225 I=1,PERIOD
         READ ( 7,1000,END=250 )JYR,JMON,(SFLUX(JMON,JYR,J),J=1,2)
  225 CONTINUE
C
C  Fill up gaps in the early, ending years
C
  250 CALL FILL (SFLUX(1,ISTART,1),IYR,IMON,JYR,JMON)
      CALL FILL (SFLUX(1,ISTART,2),IYR,IMON,JYR,JMON)
C
251   CLOSE ( UNIT=7,STATUS='KEEP' )
C
C  Finished
C
      RETURN
C
      END
C
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
```

```
      SUBROUTINE FILL (ARRAY,IYR,IMON,JYR,JMON)
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C
C  Fill fills ARRAY by repeating data through a 132 month cycle.
C
C
C  Variable list
C
C    array = solar data array that has cyclical data
C    iyr   = first year that data was read for array
C    imon  = first month of first year that data was read for array
C    jyr   = last year that data was read for array
C    jmon  = last month of last year that data was read for array
C
C
C
C
      INCLUDE 'COMMONPS.BLK'
C
      REAL ARRAY(*)
C
      INTEGER IYR,IMON,JYR,JMON
C
C  Find the first and last array element with data
C
      IBEG = (IYR-ISTART)*12 + IMON
      IEND  = (JYR-ISTART)*12 + JMON
C
C  Check that at least one whole cycle was read
C
      IF ( IEND-IBEG.LT.131 ) THEN
         WRITE ( 6,100 )
  100    FORMAT ( /1X,'LESS THAN ONE CYCLE IN SOLAR FLUX FILE')
         STOP
      END IF
C
C  Check data for gaps
C
      DO 40 I = IBEG, IEND
         IF ( ARRAY(I).LT..01 ) THEN
            IYEAR = INT((I-1)/12) + ISTART
            IMONTH = MOD((I-1),12) + 1
            WRITE ( 6,110 ) IYEAR, IMONTH
  110       FORMAT (/1X,'NO SOLAR FLUX DATA FOR YEAR ',I4,', MONTH ',I2)
            STOP
         END IF
   40 CONTINUE
C
C  Fill array beginning
C
      DO 50 I = IBEG-1, 1, -1
         ARRAY(I) = ARRAY(I+132)
```

```
   50 CONTINUE
C
C  Fill array ending
C
      DO 60 I = IEND+1, PERIOD
         ARRAY(I) = ARRAY(I-132)
   60 CONTINUE
C
C  Finished , return
C
      RETURN
C
      END
C
C
C
      SUBROUTINE DIRLIST
      CHARACTER*80 LINE
      OPEN(UNIT=17,FILE='DIRECTORY.LIST',STATUS='OLD',ERR=30)
      REWIND 17
5     READ(17,10,ERR=20) LINE
      WRITE(6,11) LINE
      GOTO 5
20    REWIND 17
      CLOSE(UNIT=17,STATUS='KEEP')
10    FORMAT ( A80)
11    FORMAT ( 1X,A80)
30    RETURN
      END

   0 serious errors detected.
   1 warning message generated.
2287 lines compiled.
```

SD_SURF User's Manual

Appendix B. A_SURF Source Code

Appendix B - A_SURF Listing

```
Listing from Language Systems FORTRAN (Version 3.0 Tue, Nov 19, 1991)
      Tue, Dec 7, 1993   8:04 PM
Options OFF: A C CASE CCD CCX CRAY DYN E EXTENDED F77 I2 LINEFEED MC68040 NOIMPLICIT
OV R S SANE SYM T72 TRACE W X
Options ON:  ANSI BKG=3 L MC68020 MC68881 OPT=1 SAVEALL U VAX Z

CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C                                   C
C       A_SURF VER 1.8  12/7/93     C
C                                   C
C          MARTIN MARIETTA          C
C        MANNED SPACE SYSTEMS       C
C                                   C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C
C
C  A_SURF VER 1.8 will compute the fractional area for a threat and element
C  and stores this in a matrix as a function of velocity and obliquity.
C  This includes the relative probability of each threat occurring.
C  The process is repeated for all elements and threats, summing up the
C  areas.  One data point on the surface represents the sum of all projected
C  areas that can be hit by a particle at a certain velocity and obliquity,
C  times the fraction of the total flux at that velocity.
C
C  The code requires one file generated by other code as input, a GEOMETRY
C  output file from the BUMPER code. This file contains the threat
C  information and the element id, pid, and surface area lists. In
C  addition it includes a list of the exposed elements and their
C  impact angles for each threat case.
C
C  A_SURF code was developed under the NASA contract 'Structural Damage
C  Prediction and Analysis for Hypervelocity Impacts Study' under the
C  direction of N. Elfer. Portions of BUMPERII version 1.2a have been
C  used to maintain compatibility.
C
C  Version 1.6 reads BUMPERII version 1.3 Geometry output.
C
C  Version 1.8 corrects a problem in NEL. It now uses ElemLoc(POINT(J,I))
C  instead of POINT(J,I)
C
C  Include module COMMONAS variable list
C
C  Scalers
C
C    areatot = the total effective area for the ranges
C    binc = impact angle (beta) increment , deg
C    cbeta = cosine of beta the impact obliquity angle
C    it = current threat case
C    itype = analysis type , 1- debris, 2-meteoroids
C    nb = number of angles in the response array
C    nee = the total number of exposed elements summed
C    nel = current element number
C    nelm = total number of elements
C    nr = number of element ranges to sum over
C    nt = number of threat cases
C    nv = number of velocities in the response array
C    pid = the property id associated with all elements of the ranges
```

```
C    vr = impact (relative) velocity , km/sec
C    vinc = impact (relative) velocity increment , km/sec
C
C  Arrays
C
C    area = array containing the value of the surface area for each
C           element, sq-meters
C    areas = the area surface array containing the summed area fractions for
C            each velocity and obliquity for all elements in the specified
C            element id ranges.    (vr,beta)
C    exposed = list of the number of exposed elements for each threat
C              angle.
C    geometry = array containing the values of the cosine of the impact
C               angle for each exposed element for each threat angle.
C    id = array containing the values of the element and property id
C         for each element
C         1- id
C         2- pid
C    ixasc = rotation axis, 1-x, 2-y, 3-z
C    ner = array containing the range number for each element
C    point = array of the element numbers corresponding to the elements
C            in the geometry array.
C    range = array containing the starting and ending elment id for each
C            range to sum over
C            1-starting id
C            2- ending id
C    rotang = rotation angle
C    threat = array containg threat information
C             1-theta angle, rad
C             2-phi angle, rad
C             3-vr, km/sec
C             4-prob
C
C
C  Main Program Variable List
C
C  Scalers
C
C    answer = user input
C    areae = the area times the threat probability
C    asbfile = the binary output Area Surface filename.
C    astfile = the text output Area Surface filename.
C    dela = the delta obliquity for a given threat/element
C    dela00 = distributed area fraction for the upper left quadrant
C    dela01 = distributed area fraction for the upper right quadrant
C    dela10 = distributed area fraction for the lower left quadrant
C    dela11 = distributed area fraction for the lower right quadrant
C    delv = the delta velocity for a given threat/element
C    first = logical first pass flag.
C    ic = current range number
C    ob = Obliquity for the current threat/element.
C    prob = threat probability from threat array
C
C  Arrays
C
C    None.
C
C
```

```
C
      LOGICAL FIRST
C
      CHARACTER*80 ANSWER
C
      INTEGER*2 IC
C
      REAL*4 PROB
C
      INCLUDE 'COMMONAS.BLK'
C
C  Write header to screen and read in output filenames and
C  element id ranges.  Open the output text file for GEOREAD.
C
      CALL INPUT
C
C  Read in the GEOMETRY output file
C
      CALL GEOREAD
C
C  Initialize the Velocity increment and number of velocities for
C  the Area Surface
C
      VINC=0.25
      NV=68
C
C  Initialize the Obliquity increment and number of angles for
C  the Area Surface
C
      BINC=5.0
      NB=19
C
C  Initialize the Area Surface to 0.0.
C
      DO 50 I2=1,NB
         DO 50 I1=1,NV
               AREAS(I1,I2)=0.0
   50 CONTINUE
C
C  Initialize the total effective area
C
      AREATOT = 0
C
C  Initialize the property id for each element range to zero.
C
      PID=0
C
C  Determine the range number of each element and verify all elements
C  in all ranges have the same property id.  The ID array is assumed to be
C  sorted by increasing id number.  And, the ranges are assumed to be in
C  assending order.
C
      FIRST=.TRUE.
      IC=1
      DO 150 I=1,NELM
C
C        is the element id in this range
C
```

```
  110    IF( ID(1,I).LT.RANGE(1,IC) ) THEN
C
C          no
C
           NER(I)=0
         ELSEIF( ID(1,I).LE.RANGE(2,IC) ) THEN
C
C          yes, verify this is not a new property id.
C          all elements processed must have the same property id
C
           NER(I)=IC
           IF( ID(2,I).NE.PID ) THEN
             IF( PID.EQ.0 ) THEN
               PID=ID(2,I)
               WRITE( 6,60000 ) PID
60000          FORMAT( /' Processing Property ID ',I5/ )
             ELSE
               IF( FIRST ) WRITE( 6,60001 ) ID(1,I)
60001          FORMAT( /' ---WARNING--- Multiple Property IDs in '
     *                'Table.  First occurance at element ',I6/)
               FIRST=.FALSE.
             ENDIF
           ENDIF
         ELSE
           IF( IC.LT.NR ) THEN
C
C            next range
C
             IC=IC + 1
             GO TO 110
           ELSE
             NER(I)=0
           ENDIF
         ENDIF
  150 CONTINUE
C
C Process all threat cases.
C
      DO 400 I=1,NT
C
C Set the threat index and get the impact velocity
C
         IT=I
         VR=THREAT(3,IT)
C
C Ignore impact velocities less than VINC
C
         IF( VR.LT.VINC ) THEN
           GO TO 400
         ENDIF
C
C Get the probability of the threat
C
         PROB = THREAT(4,IT)
C
C Evaluate each exposed element
C
         DO 300 J=1,EXPOSED(I)
```

```
C
C  Set the element number
C
C                NEL=POINT(J,I) !!!! Error used in version 1.6
               NEL=ElemLoc(POINT(J,I))
C
C  Get the element range, skip it if not in requested range
C
               IC=NER(NEL)
               IF( IC.EQ.0 ) GO TO 300
C
C  Get the cosine of the impact angle from the Geometry array.
C
               CBETA=GEOMETRY(J,IT)
C
C  Compute the Obliquity
C
               OB=(180./3.14159265) * ACOS(CBETA)
C
C  Compute the area times threat probability
C
C              AREAE=AREA(NEL,IT) * CBETA * PROB
               AREAE=AREA(J,IT) * CBETA * PROB
C
C  Compute the total effective area
C
               AREATOT = AREATOT + AREAE
C
C  Compute delta V and delta obliquity
C
               DELV=AMOD(VR,VINC)
               DELA=AMOD(OB,BINC)
C
C  Compute the Area Surface array indices
C
               IV=INT(VR/VINC)
               IA=INT(OB/BINC) + 1
C
C  Compute the distributed area fractions
C
               DELA00=AREAE * ((VINC - DELV) / VINC) *
     *                       ((BINC - DELA) / BINC)
               DELA10=AREAE * (DELV / VINC) * ((BINC - DELA) / BINC)
               DELA01=AREAE * ((VINC - DELV) / VINC) * (DELA / BINC)
               DELA11=AREAE * (DELV / VINC) * (DELA / BINC)
C
C  Sum to the Area Surface.
C
               AREAS(IV  ,IA)=AREAS(IV  ,IA) + DELA00
               AREAS(IV+1,IA)=AREAS(IV+1,IA) + DELA10
               IF( OB.LT.90.0 ) THEN
                 AREAS(IV  ,IA+1)=AREAS(IV  ,IA+1) + DELA01
                 AREAS(IV+1,IA+1)=AREAS(IV+1,IA+1) + DELA11
               ENDIF
C
C  Count the number of exposed elements distributed in the Area Surface
C
               NEE=NEE + 1
```

```
  300    CONTINUE
  400 CONTINUE
C
C  Output the Area Surface to the output file.
C
C  Open the file
C
      OPEN( UNIT=2,FILE=ASBFILE,STATUS='NEW',FORM='UNFORMATTED' )
C
C     The following output variables are INTEGER*2:
C       ITYPE
C     The following output variables are INTEGER*2:
C       NB, and NV
C     The following output arrays are INTEGER*2:
C       RANGE, PID, AND NR
C     The following output variables are REAL*4:
C       AREATOT, BINC and VINC
C     The following output array is REAL*4:
C       AREAS
C
C
C  output the analysis type, the number of ranges, the property id
C  and the total effective exposure area
C
      WRITE(2) ITYPE,NR,PID,AREATOT
C
C  output the ranges.
C
      WRITE(2) ((RANGE(I,J),I=1,2),J=1,NR)
C
C  output the number of impact angles and the impact angle increment
C
      WRITE(2) NB,BINC
C
C  output the number of velocities and the impact velocity increment
C
      WRITE(2) NV,VINC
C
C  output the Area Surface data
C
      WRITE(2) ((AREAS(I,J) ,I=1,NV) ,J=1,NB)
C
C  close the binary file.
C
      ENDFILE 2
      CLOSE( UNIT=2,STATUS='KEEP' )
C
C
C
C  WRITE TEXT FILE
C
C     Write the number of ranges and the Property ID.
C
      WRITE( 6,621 ) NR,PID,AREATOT
      WRITE( 10,621 ) NR,PID,AREATOT
  621 FORMAT( 1X,'RANGES=',I2,' First PID=',I9,
     1   '   EFF. AREA =',F12.5)
C
```

```
C
      IF( FIRST ) THEN
        CONTINUE
        ELSE
          WRITE( 10,623 )
  623          FORMAT( /' ---WARNING--- Multiple PIDs in Table.'/)
      ENDIF
C
C      Write the start and end Element ID for each range.
C
      DO 625 I=1,NR
        WRITE ( 10,624 ) I,RANGE(1,I),RANGE(2,I)
  624   FORMAT (1X,I2,', START/END,',I12,',',I12)
  625 CONTINUE
C
C
C  Write Area Surface array for 5 degree and 0.25 km/sec increments
C
C
        I=1
      WRITE ( 10,630 ) I,(J,J=0,90,5)
  630   FORMAT ( I6.2,19(',',I12.2) )
C
            DO 700 K=1,NV,1
               VR=K*VINC
               WRITE( 10,650 ) VR,(AREAS(K,J),J=1,19,1)
  650          FORMAT ( F6.2,19(',',E12.4) )
  700       CONTINUE
C
C
C  Close TEXT Output file
C
      CLOSE ( UNIT=10,STATUS='KEEP' )

C
C
C
      WRITE( 6,60005 ) ASBFILE, ASTFILE
60005 FORMAT( /' The Area Surface file is complete.'/
     *  ' binary filename: ',A,/,' text filename: ',A)
C
C  Finished
C
      STOP
C
      END
C
C
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
      SUBROUTINE INPUT
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C
C  Input writes the program header to the screen and reads in the
C  summary ouput filename. It also determines the analysis type, the
```

```
C  spacecraft exposure time, operating altitude and the element id sum
C  ranges.
C
C
C
C  note: for variables contained in the common block refer to the main
C        listing for definition
C
C  Variable list
C
C    answer = character string representing user input
C    ic = counter for the number of element id ranges
C    sfile = summary output filenanme
C
C
C
      INCLUDE 'COMMONAS.BLK'
C
      CHARACTER*80 ANSWER
C
C  Write header to screen
C
      WRITE ( 6,10 )
   10 FORMAT (/1X,'*********************',//1X,3X,'A_SURF VER 1.8',
     1         /1X,'Last Update 12/7/93',
     2         /1X,'for BUMPERII ver1.3',//1X,'*********************')
C
C
C
C
C  Get the BINARY OUTPUT filename from the user
C
      WRITE( 6,60002 )
60002 FORMAT(/' Binary output filename?<CR=DATA.ASB> >',$ )
      READ( 5,50001 ) ASBFILE
50001 FORMAT( A )
C
      IF( ASBFILE.EQ.' ' ) ASBFILE='DATA.ASB'
C
C
C  Get the TEXT OUTPUT filename from the user
C
  600 WRITE( 6,60003 )
60003 FORMAT(/' Text output filename?<CR=DATA.AST> >',$ )
      READ( 5,50002 ) ASTFILE
50002 FORMAT( A )
C
      IF( ASTFILE.EQ.' ' ) ASTFILE='DATA.AST'
C
C
C  Open the text output file for GEOREAD to use.
C  ON VAX USE RECL=256 TO WRITE ENTIRE RESULTS.
C
      OPEN(UNIT=10,FILE=ASTFILE,STATUS='NEW',RECL=256,ERR=600)
C
C
C
C  Read in element ranges to sum over
```

```
C
      IC=0
C
      WRITE ( 6,250 )
  250 FORMAT(/1X,' ONE Area Fraction Table will be created '/
     1     1X,' from ALL of the ranges of element IDs selected.'/
     2     1X,' INPUT THE STARTING AND ENDING ELEMENT ID FOR',
     3     ' EACH RANGE'/1X,' ENTER D <CR> OR <CR> WHEN DONE')
C
  270 IC=IC+1
  275 WRITE ( 6,280 )IC
  280 FORMAT ( /1X,'RANGE',I4,' IN THE TABLE. ')
C
  285 WRITE ( 6,290 )
  290 FORMAT ( 1X,'STARTING ELEMENT ID : ',$)
      READ ( 5,30 ) ANSWER
   30 FORMAT(A)
C
      IF ( ANSWER(1:1).EQ.' ' .OR. ANSWER(1:1).EQ.'D' ) GO TO 500
C
      READ ( ANSWER(1:80),300,ERR=285 ) RANGE(1,IC)
  300 FORMAT ( BN,I12 )
C
  305 WRITE ( 6,310 )
  310 FORMAT ( 1X,'ENDING ELEMENT ID   : ',$)
      READ ( 5,30 ) ANSWER
C
      IF ( ANSWER(1:1).EQ.' ' .OR. ANSWER(1:1).EQ.'D' ) GO TO 305
C
      READ ( ANSWER(1:80),300,ERR=305 ) RANGE(2,IC)
C
C  Check that ending id > starting id
C
      IF ( RANGE(1,IC).GT.RANGE(2,IC) ) THEN
         WRITE ( 6,320 )
  320    FORMAT ( 1X,'---ERROR--- Starting ID greater then Ending ID')
         GO TO 275
      END IF
C
C  Next Range if have not reached max allowed.
C
      IF ( IC.LT.IRNGS ) GO TO 270
         WRITE(6,340) IC
  340    FORMAT(1X,'---WARNING--- A maximum of ',I2,' ranges will '
     *             ,'be processed.')
C
C  Check that values were input
C
  500 CONTINUE
      IF ( RANGE(1,1).EQ.0.0 .AND. RANGE(2,1).EQ.0.0 ) THEN
         WRITE ( 6,330 )
  330    FORMAT ( 1X,'---ERROR--- No Range Values Input' )
         IC=0
         GO TO 270
      END IF
C
C  Set the number of ranges equal to the number read in
C
```

```
      NR=IC-1
C
C  Finished
C
C     RETURN
      END
C
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
      SUBROUTINE GEOREAD
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C
C  Georead reads in the output file from the GEOMETRY code. This file
C  contains the global threat and element data as well as the list of
C  exposed elements and their impact angles for each threat case.
C
C
C
C  note: for variables contained in the common block refer to main
C        listing for definition
C
C
C  Variable List
C
C    answer = character string representing user input
C    gfile = geometry output filename
C    itf = analysis type contained in the
C
C
      INCLUDE 'COMMONAS.BLK'
C
      CHARACTER LENGTH*2
      CHARACTER*1 IX(3)
      CHARACTER*20 BUMDTTM
      CHARACTER*80 ANSWER,GFILE,Form
      CHARACTER*30 AA
      CHARACTER*40 BB
      CHARACTER*12 CC
C      INTEGER*4      ITF
      INTEGER*2      ITF
      REAL*4 AREAMAXSF
C      WRITE(6,*) 'IBOTHS, FIRST LINE IN GEOREAD',IBOTHS
      IF (IBOTHS.EQ.1) GOTO 60
C
C     Read in the GEOMETRY output filename, set the default to station_?.gem
C
      IF (INDEX(ROOTFILE,'.') .EQ. 0) ROOTFILE='STATION.'
      ANSWER=ROOTFILE(1:INDEX(ROOTFILE,'.'))//'GEM'
      JOT = INDEX (ROOTFILE, '.' )
      WRITE (LENGTH, '(I2)' )JOT+3
      FORM = '(/1X,''GEOMETRY OUTPUT FILENAME (<CR>='',A'//LENGTH//','''
     . ) > '',$)'
   10 WRITE (6,FORM)ANSWER
      READ ( 5,30 ) GFILE
   30 FORMAT (A)
```

```
C
C  !!!!  THIS OPEN FOR THE MAC WILL GIVE THE NORMAL FINDER DIALOG
C        BOX.  THE DIRLIST METHOD IS SKIPPED
      IF (GFILE(1:1).EQ.'?') THEN
          OPEN ( UNIT=22,FILE=*,STATUS='OLD',FORM='UNFORMATTED',ERR=40 )
          INQUIRE( UNIT=22,NAME=GFILE)
          IBOTHS=0
          GOTO 60
      END IF
C
C  !!!!  END OF MAC OPEN
C
      IF (GFILE(1:1).EQ.'?') THEN
          CALL DIRLIST
          GOTO 10
      END IF
      IF ( GFILE(1:4).EQ.'    ' ) THEN
        GFILE=ANSWER
      ELSE
        ROOTFILE = GFile(1:INDEX(GFile,'.'))
      ENDIF
C
      IF(IBATCOM.EQ.1) THEN
       WRITE(13,'(A)') GFILE
       RETURN
      END IF
C
C     Open the file
C
      OPEN (UNIT=22,FILE=GFILE,STATUS='OLD',FORM='UNFORMATTED',ERR=40 )
C
      IBOTHS=0
      GO TO  60
C
C     Error control
C
   40 WRITE ( 6,50 )
   50 FORMAT ( /1X,'UNABLE TO OPEN FILE ' )
      GO TO 10
C
C     Read in the analysis type,the number of threat cases, and the
C     number of elements
C
C      WRITE(*,*) 'IBOTHS IN G.READ',IBOTHS
C  60  READ (22) ITYPEIN,IType,IEnv,NT,NELM,Inclin
  60  READ (22) ITYPEIN,IType,IEnv,NT,NELM,Inclin, AREAMAX
C      WRITE(6,*)'ITYPEIN,IType,IEnv,NT,NELM,Inclin'
C      WRITE(6,*) ITYPEIN,IType,IEnv,NT,NELM,Inclin
      IF (ITYPEIN.EQ.3.AND.ITYPE.EQ.1) IBOTHS=1
      IF (ITYPEIN.EQ.3.AND.ITYPE.EQ.2) IBOTHS=2
      IF (IType.EQ.1) THEN
        WRITE (6,62)
   62   FORMAT (/5X,' Debris Analysis ')
        IF (IEnv.EQ.1) THEN
          WRITE (6,64)
        ELSE
          WRITE (6,65)
        END IF
```

```
      ELSE
        WRITE (6,63)
   63   FORMAT (/5X,' Meteoroid Analysis')
        IF (IEnv.EQ.1) THEN
          WRITE (6,64)
   64     FORMAT (/5X,' JSC-20001&6000 Environment')
        ELSE
          WRITE (6,65)
   65     FORMAT (/5X,' JSC-7/90 Memo')
        END IF
      END IF
C
C     Check that the number of threats and the number of elements are less
C     than the maximum allowed
C
      IF ( NT.GT.ITH ) THEN
          WRITE ( 6,100 )NT
  100    FORMAT (/1X,'NUMBER OF THREATS IS GREATER THAN ALLOWED',I9)
         WRITE ( 6,105 )
  105    FORMAT ( 1X,'ARRAY SIZE MUST BE INCREASED & CODE RECOMPILED')
         STOP
      END IF
C
      IF ( NELM.GT.IELM ) THEN
         WRITE ( 6,110 )
  110    FORMAT ( /1X,'NUMBER OF ELEMENTS IS GREATER THAN MAX ALLOWED')
         WRITE ( 6,105 )
         STOP
      END IF
C
C  Initialize the arrays to 0.0
C
      DO 150 I=1,NT
         THREAT(3,I)=0.0
         THREAT(4,I)=0.0
         EXPOSED(I)=0
         DO 140 J=1,NELM
            GEOMETRY(J,I)=0.0
            ID(1,J)=0
            ID(2,J)=0
            POINT(J,I)=0
  140    CONTINUE
  150 CONTINUE
C
C     Read in the Threat data
C
      DO 175 I=1,NT
         READ (22) (THREAT(J,I),J=1,4)
  175 CONTINUE
C
C     Read in the element id, and  property id storing them in the ID
C     array.
C
      DO 180 I=1,100000
        ElemLoc(I) = 0
  180 CONTINUE

      DO 200 I=1,NELM
```

```
             READ (22) (ID(J,I),J=1,6)
             ElemLoc(ID(1,I)) = I
  200 CONTINUE
C
C     Read in the element's surface area storing it in the AREA array.
C
      DO 250 I=1,NELM
         READ (22) AREA(I,0)
C         WRITE(6,*) 'AREA(I,0),I,NELM'
C         WRITE(6,*) AREA(I,0),I,NELM
  250 CONTINUE
C
C  Read in the geometry data for the exposed elements
C
      DO 500 I=1,NT
C
C     Read in the threat case and the number of exposed elements
C
         READ (22) IT,EXPOSED(I)
C         WRITE(6,*) 'IT,EXPOSED(I)',IT,EXPOSED(I)
C
C        Loop thru the exposed elements
C
         DO 400 J=1,EXPOSED(I)
C
C          Read in the element number (storing in the POINT array), and the
C          cosine of the impact angle (storing in the GEOMETRY array).
C
           READ (22) POINT(J,I),GEOMETRY(J,I),Area(J,I)
C
  400    CONTINUE
C
  500    CONTINUE
C
C      CALL LIB$DATE_TIME(BUMDTTM)
C !!!!! TIME ONLY RECORDED FOR MAC VERSION
C      CALL TIME(BUMDTTM)
C
C     Write gfile to summary file
C
C      WRITE (10,575)BUMDTTM
C  575 FORMAT ('1','SHIELD',40X,A,/)
      WRITE ( 10,600 )GFILE
  600 FORMAT ( 1X,'GEOMETRY OUTPUT FILE = ',A )
C
C     To read Header from  .GEM  file            2-8-91
C
      READ ( 22,END=630 ) AA
      WRITE ( 10, '( //A )' ) AA
      DO 610 J = 1,3
        READ (22 ) CC,BB
        WRITE ( 10, '( 2A )' ) CC,BB
  610 CONTINUE
C
      WRITE ( 10,'( /A,I4)') ' Threat (1 Debris, 2 Meteoroid) ',IType
      WRITE ( 10,'( /A,I4)') ' Environment (1 Old, 2 New) ',IEnv
      WRITE ( 10,'( 5X,A,I5)')     '   Number of Threats      ',NT
C
```

```
C     Write inclin to summary file
      IF ( ITYPE.EQ.1.AND.IEnv.EQ.2 ) WRITE ( 10,520 )INCLIN
  520 FORMAT ( 1X,'INCLINATION (DEGREES)   = ',F6.1 )
C
      READ ( 22 ) AA            !  Rotation Axes and Angles
C
      DO 620 J = 1,3             !  Maximum 3 rotations
        READ ( 22 ) IXASC(J),ROTANG(J)
        IF (IXASC(J).NE.0.AND.ROTANG(J).NE.0.AND.J.EQ.1)
     +     WRITE (10,'(/1X,A)') AA
        IF (IXASC(J).NE.0.AND.ROTANG(J).NE.0) THEN
          IF(IXASC(J).EQ.1) IX(J)='X'
          IF(IXASC(J).EQ.2) IX(J)='Y'
          IF(IXASC(J).EQ.3) IX(J)='Z'
          WRITE ( 10,619 ) IX(J),ROTANG(J)
          WRITE ( 6,619 ) IX(J),ROTANG(J)
  619     FORMAT ( 2X,'  Axis=',A3,';   Angle=',F8.2 )
        END IF
  620 CONTINUE
      GO TO 650
  630 WRITE ( 10,640 )
  640 FORMAT ( /2X,'--- No Header following .Gem file ---' )
C
  650 IF (IBOTHS.EQ.1) RETURN
C
C     Close file
C
      CLOSE ( UNIT=22,STATUS='KEEP' )
C
      RETURN
C
      END
C
      SUBROUTINE DIRLIST
      CHARACTER*80 LINE
      OPEN(UNIT=17,FILE='DIRECTORY.LIST',STATUS='OLD',ERR=30)
      REWIND 17
5     READ(17,10,ERR=20) LINE
      WRITE(6,11) LINE
      GOTO 5
20    REWIND 17
      CLOSE(UNIT=17,STATUS='KEEP')
10    FORMAT ( A80)
11    FORMAT ( 1X,A80)
30    RETURN
      END
```

SD_SURF User's Manual

Appendix C. R_PLOT5 Source Code

R_PLOT5 Listing

```
Listing from Language Systems FORTRAN (Version 3.0 Tue, Nov 19, 1991)
      Sat, Sep 12, 1992   11:50 AM
Options OFF: A ANSI BKG=0 CASE CCD CCX CRAY DYN E EXTENDED F77 I2 LINEFEED MC68020
MC68040 MC68881 NOIMPLICIT OV R SANE SAVEALL SYM T72 TRACE U VAX W X Z
Options ON:  C L S

CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C                                  C
C       R_PLOT5 VER 1.6  8/23/92   C
C                                  C
C          MARTIN MARIETTA         C
C        MANNED SPACE SYSTEMS      C
C                                  C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C
C
C  R_PLOT5 VER 1.6 reads the BUMPERII v1.3 - RESPONSE binary output
C  and converts it to formatted output.  The data is put out at
C  5 degree increments rather than the 15 degree increments originally
C  used by RPLOT.  The output velocity increment is set to 0.5 km/s for
C  spreadsheet use. Commas are used to delimit the output to make it
C  more easily read by a spreadsheet program such as Microsoft EXCEL.
C  The output may then be utilized by SD_Surf for EXCEL to perform
C  probability analysis.
C
C  The code requires the output file of the RESPONSE portion of the
C  BUMPERii code.  This file contains the ballistic limit
C  (minimum diameter to penetrate) as a function of velocity
C  and obliquity.
C
C  The RESREAD subroutine is taken directly from BUMPERII v1.2a - Shield
C  version 1.2a except for the COMMONRP.BLK instead of COMMON2.BLK
C  BUMPER was developed under the NASA contract 'Integrated Wall Design
C  Guide and Penetration Control Plan' by M.A.Wright & A.R.Coronado.
C
C  SD_Surf was developed under the NASA contract 'Structural Damage
C  Prediction and Analysis for Hypervelocity Impacts Study' under the
C  direction of N. Elfer.
C
C  Version 1.6 was modified to read BUMPERII version 1.3 file
C
C  Include module COMMONRP variable list
C
C
C    aden = combined areal density of shields
C    conf = text description of wall configuration
C    diam = critical diameter , cm
C    idens = debris density, 1- constant density, 2-size function
C    ienv = environment type, 1- JSC 20001&6000, 2- 7/90 memo
C    it = current threat case
C    itype = analysis type , 1- debris, 2-meteoroids
C    nb = number of angles in the response array
C    nc = number of wall configurations in the response array
```

```
C     nv = number of velocities in the response array
C     pid = the property id associated with all elements of the ranges
C     rootfile = file name
C     rsfile = name of R_Plot5 output file for response surface
C     units = english or metric
C     vr = impact (relative) velocity , km/sec
C     vinc = impact (relative) velocity increment , km/sec
C
C   Arrays
C
C     adar = areal density g/cm**2
C     pids = array containing the current pid number to process
C     response = array containing the values of the critical diameter as
C                a function of impact angle and velocity.
C                (vr,beta,pid)
C     standm = shield stand-off, cm
C     shden = shield density, g/cc
C     shthkm = shield thickness, cm
C     vwden = vessel wall density, g/cc
C     vwthkm = vessel wall thickness, cm
C
C
C   Main Program Variable List
C
C   Scalers
C
C
C     first = logical if only one PID encountered in requested ranges
C     answer = user input
C     ob = Obliquity for the current threat/element.
C
C
C
      LOGICAL FIRST
C
      CHARACTER*80 ANSWER
C
C
      INCLUDE 'COMMONRP.BLK'
C
C   Initialize the Velocity increment and number of velocities
C
      VINC=0.25
      NV=68
C
C   Initialize the Obliquity increment and number of angles.
C
      BINC=5.0
      NB=19
C
C
C
      IBATCOM = 0
C
```

```
C  Write header to screen and read in orbital parameters
C
      CALL RPINPUT
C
C  SPECIFY DEBRIS ANALYSIS FOR RESREAD
C
      ITYPE = 1
C
C  Read in the RESPONSE output file.  This is identical to
C  the RESREAD subroutine in BUMPER.
C
      CALL RESREAD
C
C  Determine the RESPONSE PIDs to process.
C
C  If number of cases (NC) is only one then proceed.
C
      IF (NC.EQ.1) THEN
         PIDS(1)=1
         WRITE ( 6,20 )
   20    FORMAT (/1X,'The one case in the RESPONSE file will be used' )
C
C  For multiple PIDs select one or all. If only one, NC is set to 1.
C  Write number of PIDs and first PID in A_SURF to screen.
C
      ELSE
         WRITE ( 6,25 ) NC
   25    FORMAT (/1X,'The Number of PIDs in the RESPONSE file is  ',I4)
C
         WRITE ( 6,30 )
   30    FORMAT    (//1X,'Enter the PID number to use a specific PID.',
    2              /1X,'Enter <CR> to use all PIDs.')
C
         READ ( 5,35 ) ANSWER
   35    FORMAT (A)
         IF ( ANSWER(1:1).EQ.' ' .OR. ANSWER.EQ.'A') THEN
            DO 40 I1=1,NC
               PIDS(I1) = I1
   40       CONTINUE
         ELSE
         READ ( ANSWER(1:80),45 )PIDS(1)
   45       FORMAT ( BN,I2 )
            NC=1
         ENDIF
C
      ENDIF
C
C
C  Print out the flux x area x time surface in a comma
C  delimited format to be read by spreadsheets.
C
      CALL RP5TEXT
C
C
```

```
C
C
C  Close summary file
C
        CLOSE ( UNIT=10,STATUS='KEEP' )
        WRITE( 6,60003 ) RSFILE
60003   FORMAT( /' The R_PLOT5 file is complete.'/
     1         ' filename: ',A )

C
C  Finished
C
        END
```

Symbol Table for: MACII_R_plot5_16$main

The following symbols were defined but NOT referenced:

Symbol Name	Other Information	Class	Data Type	Size	Offset	RefCnt
ADAR	[Cmn] Dims - 1:36	Array	REAL*4	144	192436	0
ADEN	[Cmn]	Scalar	REAL*4	4	192432	0
BATCOM	[Cmn]	Scalar	REAL*4	4	192624	0
CONF	[Cmn]	Scalar	CHARACTER	12	192632	0
DIAM	[Cmn]	Scalar	REAL*4	4	4	0
FIRST		Scalar	LOGICAL*4	4		0
IBOTHS	[Cmn]	Scalar	INTEGER*2	2	192628	0
IDENS	[Cmn]	Scalar	INTEGER*2	2	192430	0
IENV	[Cmn]	Scalar	INTEGER*2	2	192428	0
IT	[Cmn]	Scalar	INTEGER*2	2	8	0
ITYPEIN	[Cmn]	Scalar	INTEGER*2	2	192630	0
NT	[Cmn]	Scalar	INTEGER*4	4	16	0
PID	[Cmn]	Scalar	INTEGER*2	2	114	0
RANGE	[Cmn]	Scalar	REAL*4	4	30	0
RESPONSE	[Cmn] Dims - 1:70 1:19 1:36	Array	REAL*4	191520	188	0
ROOTFILE	[Cmn]	Scalar	CHARACTER	40	192584	0

```
SHDEN                             Array      REAL*4          144   191852      0
[Cmn ] Dims -  1:36
SHTHK                             Scalar     REAL*4            4   192580      0
[Cmn ]
SHTHKM                            Array      REAL*4          144   191708      0
[Cmn ] Dims -  1:36
STANDM                            Array      REAL*4          144   192284      0
[Cmn ] Dims -  1:36
UNITS                             Scalar     CHARACTER        12   192644      0
[Cmn ]
VR                                Scalar     REAL*4            4       22      0
[Cmn ]
VWDEN                             Array      REAL*4          144   192140      0
[Cmn ] Dims -  1:36
VWTHK                             Scalar     REAL*4            4               0
VWTHKM                            Array      REAL*4          144   191996      0
[Cmn ] Dims -  1:36

Alphabetic List:
Symbol Name                       Class      Data Type     Size     Offset  RefCnt
Other Information
------------------------------- ---------- ------------ -------- -------- ------ ---
----------------------
ADAR                              Array      REAL*4          144   192436      0
[Cmn ] Dims -  1:36
ADEN                              Scalar     REAL*4            4   192432      0
[Cmn ]
ANSWER                            Scalar     CHARACTER        80       38      4
BATCOM                            Scalar     REAL*4            4   192624      0
[Cmn ]
BINC                              Scalar     REAL*4            4        0      1
[Cmn ]
CONF                              Scalar     CHARACTER        12   192632      0
[Cmn ]
DIAM                              Scalar     REAL*4            4        4      0
[Cmn ]
FIRST                             Scalar     LOGICAL*4         4               0
I1                                Scalar     INTEGER*4         4      120      4
IBATCOM                           Scalar     INTEGER*2         2      118      1
IBOTHS                            Scalar     INTEGER*2         2   192628      0
[Cmn ]
IDENS                             Scalar     INTEGER*2         2   192430      0
[Cmn ]
IENV                              Scalar     INTEGER*2         2   192428      0
[Cmn ]
ISHLDS                            Parameter  INTEGER*4                         8 =
36
IT                                Scalar     INTEGER*2         2        8      0
[Cmn ]
ITYPE                             Scalar     INTEGER*2         2       10      1
[Cmn ]
ITYPEIN                           Scalar     INTEGER*2         2   192630      0
[Cmn ]
```

```
NB                                Scalar     INTEGER*2          2        12      1
[Cmn ]
NC                                Scalar     INTEGER*2          2        14      4
[Cmn ]
NT                                Scalar     INTEGER*4          4        16      0
[Cmn ]
NV                                Scalar     INTEGER*2          2        20      1
[Cmn ]
PID                               Scalar     INTEGER*2          2       114      0
[Cmn ]
PIDS                              Array      INTEGER*2         72       116      4
[Cmn ] Dims -  1:36
RANGE                             Scalar     REAL*4             4        30      0
[Cmn ]
RESPONSE                          Array      REAL*4        191520       188      0
[Cmn ] Dims -  1:70 1:19 1:36
ROOTFILE                          Scalar     CHARACTER         40    192584      0
[Cmn ]
RSFILE                            Scalar     CHARACTER         80        34      1
[Cmn ]
SHDEN                             Array      REAL*4           144    191852      0
[Cmn ] Dims -  1:36
SHTHK                             Scalar     REAL*4             4    192580      0
[Cmn ]
SHTHKM                            Array      REAL*4           144    191708      0
[Cmn ] Dims -  1:36
STANDM                            Array      REAL*4           144    192284      0
[Cmn ] Dims -  1:36
UNITS                             Scalar     CHARACTER         12    192644      0
[Cmn ]
VINC                              Scalar     REAL*4             4        26      1
[Cmn ]
VR                                Scalar     REAL*4             4        22      0
[Cmn ]
VWDEN                             Array      REAL*4           144    192140      0
[Cmn ] Dims -  1:36
VWTHK                             Scalar     REAL*4             4                0
VWTHKM                            Array      REAL*4           144    191996      0
[Cmn ] Dims -  1:36
_$CMNBASES                        Cmn Hndls                           124      1

Stack Frame Information:

Temporaries List:
Symbol Name                       Class      Data Type    Size     Offset   RefCnt
Other Information
-------------------------------- ---------- ------------ -------- -------- ------ ---
----------------------
_$TEMP1                           Scalar     INTEGER*4          4         0      2
{Work Area}                                                    24         4
_$TLB0                            Scalar     LOGICAL*1          1        28      4
_$TILAsgn0                        Scalar     INTEGER*4          4        32      2
_$TLB1                            Scalar     LOGICAL*1          1        36      2
_$TLB2                            Scalar     LOGICAL*1          1        37      2
```

```
Variable List:
Symbol Name                     Class      Data Type   Size     Offset   RefCnt
Other Information
-------------------------------- ---------- ----------- -------- -------- ------ ---
----------------------
ANSWER                          Scalar     CHARACTER        80       38      4
IBATCOM                         Scalar     INTEGER*2         2      118      1
I1                              Scalar     INTEGER*4         4      120      4
_$CMNBASES                      Cmn Hndls                           124      1

Variables in Blank Common:             Size:192656

Symbol Name                     Class      Data Type   Size     Offset   RefCnt
Other Information
-------------------------------- ---------- ----------- -------- -------- ------ ---
----------------------
BINC                            Scalar     REAL*4            4        0      1
[Cmn ]
DIAM                            Scalar     REAL*4            4        4      0
[Cmn ]
IT                              Scalar     INTEGER*2         2        8      0
[Cmn ]
ITYPE                           Scalar     INTEGER*2         2       10      1
[Cmn ]
NB                              Scalar     INTEGER*2         2       12      1
[Cmn ]
NC                              Scalar     INTEGER*2         2       14      4
[Cmn ]
NT                              Scalar     INTEGER*4         4       16      0
[Cmn ]
NV                              Scalar     INTEGER*2         2       20      1
[Cmn ]
VR                              Scalar     REAL*4            4       22      0
[Cmn ]
VINC                            Scalar     REAL*4            4       26      1
[Cmn ]
RANGE                           Scalar     REAL*4            4       30      0
[Cmn ]
RSFILE                          Scalar     CHARACTER        80       34      1
[Cmn ]
PID                             Scalar     INTEGER*2         2      114      0
[Cmn ]
PIDS                            Array      INTEGER*2        72      116      4
[Cmn ] Dims -  1:36
RESPONSE                        Array      REAL*4       191520      188      0
[Cmn ] Dims -  1:70 1:19 1:36
SHTHKM                          Array      REAL*4          144   191708      0
[Cmn ] Dims -  1:36
SHDEN                           Array      REAL*4          144   191852      0
[Cmn ] Dims -  1:36
VWTHKM                          Array      REAL*4          144   191996      0
[Cmn ] Dims -  1:36
```

```
VWDEN                                Array     REAL*4        144   192140     0
[Cmn ] Dims -  1:36
STANDM                               Array     REAL*4        144   192284     0
[Cmn ] Dims -  1:36
IENV                                 Scalar    INTEGER*2       2   192428     0
[Cmn ]
IDENS                                Scalar    INTEGER*2       2   192430     0
[Cmn ]
ADEN                                 Scalar    REAL*4          4   192432     0
[Cmn ]
ADAR                                 Array     REAL*4        144   192436     0
[Cmn ] Dims -  1:36
SHTHK                                Scalar    REAL*4          4   192580     0
[Cmn ]
ROOTFILE                             Scalar    CHARACTER      40   192584     0
[Cmn ]
BATCOM                               Scalar    REAL*4          4   192624     0
[Cmn ]
IBOTHS                               Scalar    INTEGER*2       2   192628     0
[Cmn ]
ITYPEIN                              Scalar    INTEGER*2       2   192630     0
[Cmn ]
CONF                                 Scalar    CHARACTER      12   192632     0
[Cmn ]
UNITS                                Scalar    CHARACTER      12   192644     0
[Cmn ]


Local Stackframe size: 128
Local Symbols: 60
```

```
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
      SUBROUTINE RP5TEXT
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C  Write RESPONSE array for 5 degree and 0.5 km/sec increments
C
C  Variable list
C  Temporary array
C
C  RPLINE = Contains diameters to penetrate at the obliquity increments
C           Identical to Response at a particular velocity and PID.
C
      INCLUDE 'COMMONRP.BLK'
      REAL*4 RPLINE(19)
C
C
C
        DO 700 I=1,NC
```

```
                WRITE ( 10,625) PIDS(I)
      625       FORMAT (1H1,/1X,'RESPONSE PID=',',',I9)
              WRITE ( 10,630 ) PIDS(I),(J,J=0,90,5)
      630     FORMAT ( I6.2,19(',',I12.2) )
   C
                  DO 690 K=2,NV,2
                     DO 640 J=1,19
                        RPLINE(J)=RESPONSE(K,J,PIDS(I))
      640            CONTINUE
                     VR=K*VINC
                     WRITE( 10,650) VR,(RPLINE(J),J=1,19)
      650            FORMAT ( F6.2,19(',',F12.8) )
      690         CONTINUE
      700     CONTINUE
   C
              RETURN
   C
              END
```

```
Symbol Table for: RP5TEXT

The following symbols were defined but NOT referenced:

Symbol Name                      Class      Data Type    Size     Offset   RefCnt
Other Information
-------------------------------- ---------- ------------ -------- -------- ------ ---
----------------------
ADAR                             Array       REAL*4           144   192436      0
[Cmn ] Dims -   1:36
ADEN                             Scalar      REAL*4             4   192432      0
[Cmn ]
BATCOM                           Scalar      REAL*4             4   192624      0
[Cmn ]
BINC                             Scalar      REAL*4             4        0      0
[Cmn ]
CONF                             Scalar      CHARACTER         12   192632      0
[Cmn ]
DIAM                             Scalar      REAL*4             4        4      0
[Cmn ]
IBATCOM                          Scalar      INTEGER*2          2                0
IBOTHS                           Scalar      INTEGER*2          2   192628      0
[Cmn ]
IDENS                            Scalar      INTEGER*2          2   192430      0
[Cmn ]
IENV                             Scalar      INTEGER*2          2   192428      0
[Cmn ]
IT                               Scalar      INTEGER*2          2        8      0
[Cmn ]
ITYPE                            Scalar      INTEGER*2          2       10      0
[Cmn ]
ITYPEIN                          Scalar      INTEGER*2          2   192630      0
[Cmn ]
```

```
NB                              Scalar     INTEGER*2        2       12      0
[Cmn ]
NT                              Scalar     INTEGER*4        4       16      0
[Cmn ]
PID                             Scalar     INTEGER*2        2      114      0
[Cmn ]
RANGE                           Scalar     REAL*4           4       30      0
[Cmn ]
ROOTFILE                        Scalar     CHARACTER       40   192584      0
[Cmn ]
RSFILE                          Scalar     CHARACTER       80       34      0
[Cmn ]
SHDEN                           Array      REAL*4         144   191852      0
[Cmn ] Dims -  1:36
SHTHK                           Scalar     REAL*4           4   192580      0
[Cmn ]
SHTHKM                          Array      REAL*4         144   191708      0
[Cmn ] Dims -  1:36
STANDM                          Array      REAL*4         144   192284      0
[Cmn ] Dims -  1:36
UNITS                           Scalar     CHARACTER       12   192644      0
[Cmn ]
VWDEN                           Array      REAL*4         144   192140      0
[Cmn ] Dims -  1:36
VWTHK                           Scalar     REAL*4           4                 0
VWTHKM                          Array      REAL*4         144   191996      0
[Cmn ] Dims -  1:36

Alphabetic List:
Symbol Name                     Class      Data Type    Size     Offset   RefCnt
Other Information
------------------------------- ---------- ----------- -------- -------- ------ ---
----------------------
ADAR                            Array      REAL*4         144   192436      0
[Cmn ] Dims -  1:36
ADEN                            Scalar     REAL*4           4   192432      0
[Cmn ]
BATCOM                          Scalar     REAL*4           4   192624      0
[Cmn ]
BINC                            Scalar     REAL*4           4        0      0
[Cmn ]
CONF                            Scalar     CHARACTER       12   192632      0
[Cmn ]
DIAM                            Scalar     REAL*4           4        4      0
[Cmn ]
I                               Scalar     INTEGER*4        4      144      5
IBATCOM                         Scalar     INTEGER*2        2                 0
IBOTHS                          Scalar     INTEGER*2        2   192628      0
[Cmn ]
IDENS                           Scalar     INTEGER*2        2   192430      0
[Cmn ]
IENV                            Scalar     INTEGER*2        2   192428      0
[Cmn ]
```

```
ISHLDS                          Parameter   INTEGER*4                          8 =
36
IT                              Scalar      INTEGER*2          2          8    0
[Cmn ]
ITYPE                           Scalar      INTEGER*2          2         10    0
[Cmn ]
ITYPEIN                         Scalar      INTEGER*2          2     192630    0
[Cmn ]
J                               Scalar      INTEGER*4          4        148   10
K                               Scalar      INTEGER*4          4        152    4
NB                              Scalar      INTEGER*2          2         12    0
[Cmn ]
NC                              Scalar      INTEGER*2          2         14    1
[Cmn ]
NT                              Scalar      INTEGER*4          4         16    0
[Cmn ]
NV                              Scalar      INTEGER*2          2         20    1
[Cmn ]
PID                             Scalar      INTEGER*2          2        114    0
[Cmn ]
PIDS                            Array       INTEGER*2         72        116    6
[Cmn ] Dims -   1:36
RANGE                           Scalar      REAL*4             4         30    0
[Cmn ]
RESPONSE                        Array       REAL*4        191520        188    2
[Cmn ] Dims -   1:70 1:19 1:36
ROOTFILE                        Scalar      CHARACTER         40     192584    0
[Cmn ]
RPLINE                          Array       REAL*4            76         68    4
Dims -  1:19
RSFILE                          Scalar      CHARACTER         80         34    0
[Cmn ]
SHDEN                           Array       REAL*4           144     191852    0
[Cmn ] Dims -   1:36
SHTHK                           Scalar      REAL*4             4     192580    0
[Cmn ]
SHTHKM                          Array       REAL*4           144     191708    0
[Cmn ] Dims -   1:36
STANDM                          Array       REAL*4           144     192284    0
[Cmn ] Dims -   1:36
UNITS                           Scalar      CHARACTER         12     192644    0
[Cmn ]
VINC                            Scalar      REAL*4             4         26    1
[Cmn ]
VR                              Scalar      REAL*4             4         22    2
[Cmn ]
VWDEN                           Array       REAL*4           144     192140    0
[Cmn ] Dims -   1:36
VWTHK                           Scalar      REAL*4             4               0
VWTHKM                          Array       REAL*4           144     191996    0
[Cmn ] Dims -   1:36
_$CMNBASES                      Cmn Hndls                               156    1

Stack Frame Information:
```

```
Temporaries List:
Symbol Name                      Class      Data Type   Size     Offset   RefCnt
Other Information
-------------------------------- ---------- ----------- -------- -------- ------ ---
----------------------
_$TEMPD4                         Scalar     INTEGER*4          4        0      2
_$TEMP3                          Scalar     INTEGER*4          4        4      2
_$TEMP2                          Scalar     INTEGER*4          4        8      2
_$TEMPD3                         Scalar     INTEGER*4          4       12      2
_$TEMP1                          Scalar     INTEGER*4          4       16      2
{Work Area}                                                   24       20
_$TIL0                           Scalar     INTEGER*4          4       44      8
_$TILAsgn0                       Scalar     INTEGER*4          4       48      2
_$TIL1                           Scalar     INTEGER*4          4       52      2
_$TE0                            Scalar     EXTENDED*12       12       56      2

Variable List:
Symbol Name                      Class      Data Type   Size     Offset   RefCnt
Other Information
-------------------------------- ---------- ----------- -------- -------- ------ ---
----------------------
RPLINE                           Array      REAL*4            76       68      4
Dims -   1:19
I                                Scalar     INTEGER*4          4      144      5
J                                Scalar     INTEGER*4          4      148     10
K                                Scalar     INTEGER*4          4      152      4
_$CMNBASES                       Cmn Hndls                            156      1


Variables in Blank Common:            Size:192656

Symbol Name                      Class      Data Type   Size     Offset   RefCnt
Other Information
-------------------------------- ---------- ----------- -------- -------- ------ ---
----------------------
BINC                             Scalar     REAL*4             4        0      0
[Cmn ]
DIAM                             Scalar     REAL*4             4        4      0
[Cmn ]
IT                               Scalar     INTEGER*2          2        8      0
[Cmn ]
ITYPE                            Scalar     INTEGER*2          2       10      0
[Cmn ]
NB                               Scalar     INTEGER*2          2       12      0
[Cmn ]
NC                               Scalar     INTEGER*2          2       14      1
[Cmn ]
NT                               Scalar     INTEGER*4          4       16      0
[Cmn ]
NV                               Scalar     INTEGER*2          2       20      1
[Cmn ]
VR                               Scalar     REAL*4             4       22      2
[Cmn ]
```

```
VINC                                  Scalar     REAL*4           4      26   1
[Cmn ]
RANGE                                 Scalar     REAL*4           4      30   0
[Cmn ]
RSFILE                                Scalar     CHARACTER       80      34   0
[Cmn ]
PID                                   Scalar     INTEGER*2        2     114   0
[Cmn ]
PIDS                                  Array      INTEGER*2       72     116   6
[Cmn ] Dims -  1:36
RESPONSE                              Array      REAL*4      191520     188   2
[Cmn ] Dims -  1:70 1:19 1:36
SHTHKM                                Array      REAL*4         144  191708   0
[Cmn ] Dims -  1:36
SHDEN                                 Array      REAL*4         144  191852   0
[Cmn ] Dims -  1:36
VWTHKM                                Array      REAL*4         144  191996   0
[Cmn ] Dims -  1:36
VWDEN                                 Array      REAL*4         144  192140   0
[Cmn ] Dims -  1:36
STANDM                                Array      REAL*4         144  192284   0
[Cmn ] Dims -  1:36
IENV                                  Scalar     INTEGER*2        2  192428   0
[Cmn ]
IDENS                                 Scalar     INTEGER*2        2  192430   0
[Cmn ]
ADEN                                  Scalar     REAL*4           4  192432   0
[Cmn ]
ADAR                                  Array      REAL*4         144  192436   0
[Cmn ] Dims -  1:36
SHTHK                                 Scalar     REAL*4           4  192580   0
[Cmn ]
ROOTFILE                              Scalar     CHARACTER       40  192584   0
[Cmn ]
BATCOM                                Scalar     REAL*4           4  192624   0
[Cmn ]
IBOTHS                                Scalar     INTEGER*2        2  192628   0
[Cmn ]
ITYPEIN                               Scalar     INTEGER*2        2  192630   0
[Cmn ]
CONF                                  Scalar     CHARACTER       12  192632   0
[Cmn ]
UNITS                                 Scalar     CHARACTER       12  192644   0
[Cmn ]

Local Stackframe size: 160
Local Symbols: 67
```

```
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
      SUBROUTINE RESREAD
```

```
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C
C  Resread reads in the output from the RESPONSE code.  This output
C  consists of the critical diameter data as a function of property
C  id, impact angle, and impact velocity.
C
C
C  note: for variables contained in the common block refer to the main
C        listing for definition.
C
C
C  Variable list
C
C    answer = character string represnting user input
C    itf = analysis type for rfile
C    rfile = response output filename
C    ienvr = environment for response file
C
      CHARACTER LENGTH*2
      CHARACTER*80 ANSWER,RFILE,Form
      CHARACTER*46 A46
      CHARACTER*12 B12A, B12B
      CHARACTER*8  C8A, C8B
      CHARACTER*2  D2
      REAL*4       STND
C      INTEGER*4 ITF, ITA, IC, ICT, ICB, IPF
C     !!! CHANGES TO BE COMPATIBLE WITH BUMPERII ver1.3
      INTEGER*2 ITF, ITA, IC, ICT, ICB, IPF, IPFUNC3
      INTEGER*2 IENVR
C
      INCLUDE 'COMMONRP.BLK'
      IF (IBOTHS.EQ.2) GOTO 60
C
C  Read in the RESPONSE output filename , set default to resp.dat
C
      IF (INDEX(ROOTFILE,'.') .EQ. 0)ROOTFILE='STATION.'
      ANSWER=ROOTFILE(1:INDEX(ROOTFILE,'.'))//'RSP'
      JOT = INDEX( ROOTFILE, '.' )
      WRITE (LENGTH, '(I2)' )JOT+3
      FORM='(/1X,''RESPONSE OUTPUT FILENAME (<CR>='',A'//LENGTH//
     . ','') > '',$)'
   10 WRITE ( 6,FORM )ANSWER
      READ ( 5,30 ) RFILE
   30 FORMAT (A)
C
C  !!!!  THIS OPEN FOR THE MAC WILL GIVE THE NORMAL FINDER DIALOG
C        BOX.  THE DIRLIST METHOD IS SKIPPED
      IF (RFILE(1:1).EQ.'?') THEN
           OPEN ( UNIT=23,FILE=*,STATUS='OLD',FORM='UNFORMATTED',ERR=40 )
           INQUIRE( UNIT=23,NAME=RFILE)
           GOTO 60
      END IF
```

```
C
C  !!!!  END OF MAC OPEN
C
         IF (RFILE(1:1).EQ.'?') THEN
              CALL DIRLIST
              GOTO 10
         END IF
         IF ( RFILE(1:4).EQ.'    ' ) THEN
            RFILE=ANSWER
         ELSE
            ROOTFILE = RFile(1:INDEX(RFile,'.'))
         ENDIF
C
         IF(IBATCOM.EQ.1) THEN
          WRITE(13,'(A)') RFILE
          RETURN
         END IF
C
C  Open the file
C
         OPEN ( UNIT=23,FILE=RFILE,STATUS='OLD',FORM='UNFORMATTED',ERR=40 )
C


         GO TO 60
C
C  Error control on open
C
      40 WRITE ( 6,50 )
      50 FORMAT ( /1X,'UNABLE TO OPEN FILE' )
         GO TO 10
C
C  Read in the analysis type and the number of property cases.
C
C      60 READ (23) ITYPEIN,ITF,IDens,NC
C      !!!  NO ERROR CHECKING ON IENVR
      60 READ (23) ITYPEIN,ITF,IENVR,IDens,NC
C         WRITE(6,*)'ITYPEIN,ITF,IDens,NC'
C         WRITE(6,*) ITYPEIN,ITF,IDens,NC
          IF (ITYPEIN.EQ.3.AND.IBOTHS.EQ.1) ITYPE=1
         IF (IDens.EQ.1) THEN
           WRITE (6,63)
      63   FORMAT (/5X,' Constant density threat')
         ELSE IF (IDens.EQ.2) THEN
           WRITE (6,64)
      64   FORMAT (/5X,' Variable density threat')
         END IF
C
C  Check that the response file is the correct analysis type
C
         IF ( ITF.NE.ITYPE ) THEN
            IF ( ITYPE.EQ.1 ) THEN
               WRITE ( 6,70 )
      70       FORMAT ( /1X,'DEBRIS ANALYSIS SPECIFIED IN GEOMETRY FILE ',
        1        'BUT RESPONSE FILE IS FOR METEOROIDS ')
```

```
            ELSE
               WRITE ( 6,80 )
     80        FORMAT (/1X,'METEOROID ANALYSIS SPECIFIED IN GEOMETRY FILE',
     1                     ' BUT RESPONSE FILE IS FOR DEBRIS' )
            END IF
C
            WRITE ( 6,90 )
     90     FORMAT ( /1X,'DO YOU WISH TO CONTINUE WITH GEOMETRY OPTION ',
     1               '(<CR>=NO) > ',$)
            READ ( 5,30 ) ANSWER
C
            IF ( ANSWER(1:1).EQ.'Y' .OR. ANSWER(1:1).EQ.'y' ) THEN
               GO TO 10
            ELSE
               STOP
            END IF
C
         END IF
C
C  Read in the impact angle information
C
         READ (23) NB,BINC
C         WRITE(6,*) 'NB,BINC'
C         WRITE(6,*) NB,BINC
C
C  Read in the impact velocity information
C
         READ (23) NV,VINC
C         WRITE(6,*) 'IMPACT VELOCITY, VEL INCR.'
C         WRITE(6,*) NV,VINC
C
C  Initialize RESPONSE to 0.0
C
         DO 200 I=1,NC
            DO 150 J=1,NB
               DO 100 K=1,NV
                  RESPONSE ( K,J,I ) = 0.
    100        CONTINUE
    150     CONTINUE
    200  CONTINUE
C
C     Read in the critical diameter data
C
C           WRITE(6,*) 'NC,NB,NV'
C           WRITE(6,*) NC,NB,NV
C           WRITE(6,*) 'RESPONSE(K,J,I)'
C          Loop thru the property id's
          DO 400 I=1,NC
C
C            Loop thru the impact angles
            DO 300 J=1,NB
C
C              Loop thru the impact velocities
              DO 250 K=1,NV
```

```
C
C             Store the critical diameter in response
              READ (23) RESPONSE(K,J,I)
C              WRITE(6,*) RESPONSE(K,J,I)
  250        CONTINUE
  300      CONTINUE
  400    CONTINUE
C
C  !!!! THE NEXT LINE WAS COMMENTED OUT FOR R_PLOT5
C       IF (INPUTCD.EQ.2) CALL SETDIAMS
C
       READ ( 23,END=440,ERR=440 ) A46
C       WRITE(6,*) 'A46'
C       WRITE ( 6,'( //1X,A)' ) A46
       WRITE ( 10,'( //1X,A)' ) A46
       READ ( 23 ) C8A,ITA,C8B,ICB,UNITS
       WRITE ( 10,'( A,I4)') ' Threat (1 Debris, 2 Meteoroid)  ',ITA
       WRITE ( 10,'( A,I4)') ' Density (1 Constant, 2 Function)  ',IDens
       WRITE ( 10,'( A,I4)') '         Number of PID Cases    ',ICB
       WRITE ( 10,'( 2A)')   '         Units  ',UNITS
C       WRITE(6,*) 'C8A,C8B'
C       WRITE(6,*) C8A,C8B
C       WRITE ( 6,'( A,I4)') ' Threat (1 Debris, 2 Meteoroid)  ',ITA
C       WRITE ( 6,'( A,I4)') ' Density (1 Constant, 2 Function)  ',IDens
C       WRITE ( 6,'( A,I4)') '         Number of PID Cases    ',ICB
C       WRITE ( 6,'( 2A)')   '         Units  ',UNITS
       DO 420 I=1,ICB
         READ ( 23 ) ICT,D2,B12A,B12B,IPF,IPFUNC3
         WRITE ( 10,411) I
 411     FORMAT( /1X,'PID NUMBER ',I4 )
C
         IF (ICT.EQ.2) THEN
           IF ( IPF.EQ.1 ) THEN
             WRITE ( 10,485)
           ELSE IF ( IPF.EQ.2 ) THEN
             WRITE ( 10,486)
           ELSE IF ( IPF.EQ.3 ) THEN
             WRITE ( 10,487)
           ELSE IF ( IPF.EQ.4 ) THEN
             WRITE ( 10,488)
           ELSE IF ( IPF.EQ.5 ) THEN
             WRITE ( 10,484)
           ELSE IF ( IPF.EQ.6 ) THEN
             WRITE ( 10,489)
           ELSE IF ( IPF.EQ.7 ) THEN
             WRITE ( 10,490)
           ELSE IF ( IPF.EQ.8 ) THEN
             WRITE ( 10,491)
           ELSE IF ( IPF.EQ.9 ) THEN
             WRITE ( 10,492)
           ELSE IF ( IPF.EQ.10 ) THEN
             WRITE ( 10,493)
           ELSE IF ( IPF.EQ.11 ) THEN
             WRITE ( 10,494)
```

```
            ELSE IF ( IPF.EQ.12 ) THEN
              WRITE ( 10,495)
            ELSE IF ( IPF.EQ.13 ) THEN
              WRITE ( 10,496)
            ELSE IF ( IPF.EQ.14 ) THEN
              WRITE ( 10,497)
            END IF
          END IF
  485     FORMAT ( /1X,'ORIGINAL PENETRATION FUNCTION')
  486     FORMAT ( /1X,'PEN4 PENETRATION FUNCTION')
  487     FORMAT ( /1X,'REGRESSION PENETRATION FUNCTION')
  488     FORMAT ( /1X,'COUR-PALAIS PENETRATION FUNCTION')
  484     FORMAT ( /1X,'BOEING INTERP PENETRATION FUNCTION')
  489     FORMAT ( /1X,'DEVELOPMENTAL6, USER INPUT')
  490     FORMAT ( /1X,'DEVELOPMENTAL7, USER INPUT')
  491     FORMAT ( /1X,'DEVELOPMENTAL8, USER INPUT')
  492     FORMAT ( /1X,'DEVELOPMENTAL9, USER INPUT')
  493     FORMAT ( /1X,'DEVELOPMENTAL10, USER INPUT')
  494     FORMAT ( /1X,'DEVELOPMENTAL11, USER INPUT')
  495     FORMAT ( /1X,'DEVELOPMENTAL12, USER INPUT')
  496     FORMAT ( /1X,'DEVELOPMENTAL13, USER INPUT')
  497     FORMAT ( /1X,'DEVELOPMENTAL14, USER INPUT')
          WRITE ( 10,'( /A )' ) ' Configuration     Shield       Wall'
C          WRITE ( 6,* ) 'ICT,D2,B12A,B12B'
C          WRITE ( 6,* ) ICT,D2,B12A,B12B
          IF (ICT.EQ.1) CONF = 'Single Plate'
          IF (ICT.EQ.2) CONF = 'Double Plate'
          IF (ICT.EQ.3) CONF = 'Multiwall'
          WRITE ( 10,'( 1X,A,4X,2A )' ) CONF,B12A,B12B
C           WRITE ( 6,'( 1X,A,4X,2A )' ) CONF,B12A,B12B
          READ ( 23 ) ShThk,VWThk,STND,ShDen(I),VWDen(I),ADEN
C          WRITE ( 6,* ) 'ShThk,VWThk,STND,ShDen(I),VWDen(I),ADEN,I'
C          WRITE ( 6,* ) ShThk,VWThk,STND,ShDen(I),VWDen(I),ADEN,I
          IF (ICT.EQ.3) THEN
            WRITE ( 10,'( A,A,F8.4)') '       Combined Areal Density',
     +                              ' of All Shields  = ',ADEN
            WRITE ( 10,'( A,F8.4)') '       Total Standoff = '
     +                        ,STND
C            WRITE ( 6,'( A,A,F8.4)') '       Combined Areal Density',
C    +                              ' of All Shields  = ',ADEN
C            WRITE ( 6,'( A,F8.4)') '       Total Standoff = '
C    +                        ,STND
            GOTO 410
          END IF
C          WRITE ( 6,'( A,F8.4)') '       Shield Thickness = ',ShThk
          IF (SHTHK.NE.0.0)
     +      WRITE ( 10,'( A,F8.4)') '       Shield Thickness = ',ShThk
 410      WRITE ( 10,'( A,F8.4)') '  Vessel Wall Thickness = ',VWThk
C          WRITE ( 6,'( A,F8.4)') '  Vessel Wall Thickness = ',VWThk
          IF (ICT.NE.3) THEN
          IF (SHTHK.NE.0.0.AND.STND.NE.0.0)
     +      WRITE ( 10,'( A,F8.4)') '               Standoff = ',STND
C          WRITE ( 6,'( A,F8.4)') '               Standoff = ',STND
          END IF
```

```
          IF ( Units .EQ. ' ENGLISH    ' ) THEN
            ShThkM(I) = ShThk*2.54
            VWThkM(I) = VWThk*2.54
            ADAR(I)=ADEN/.0142233
           ELSE
            ShThkM(I) = ShThk
            VWThkM(I) = VWThk
            ADAR(I)=ADEN
          END IF

C         With or without 30 MLI
          READ ( 23 ) A46
          WRITE ( 10,'( 4X,A)' ) A46
C          WRITE ( 6,'( 4X,A)' ) A46
  420 CONTINUE
      GO TO 450
  440 WRITE ( 10,42 )
   42 FORMAT ( /2X,' No Header following .RSP file ' )
C
  450 IF (IBOTHS.EQ.1) RETURN
C
C     Close the file and return
C
      CLOSE ( UNIT=23,STATUS='KEEP' )
C
C     Write Rfile to summary file
C
      WRITE ( 10,500 )RFILE
  500 FORMAT(1X,'RESPONSE OUTPUT FILE = ',A )
C
      RETURN
C
      END
```

Symbol Table for: RESREAD

The following symbols were defined but NOT referenced:

Symbol Name Other Information	Class	Data Type	Size	Offset	RefCnt
BATCOM [Cmn]	Scalar	REAL*4	4	192624	0
DIAM [Cmn]	Scalar	REAL*4	4	4	0
IC	Scalar	INTEGER*2	2		0
IENV [Cmn]	Scalar	INTEGER*2	2	192428	0
IT [Cmn]	Scalar	INTEGER*2	2	8	0
NT [Cmn]	Scalar	INTEGER*4	4	16	0

```
PID                              Scalar     INTEGER*2          2      114      0
[Cmn ]
PIDS                             Array      INTEGER*2         72      116      0
[Cmn ] Dims -  1:36
RANGE                            Scalar     REAL*4             4       30      0
[Cmn ]
RSFILE                           Scalar     CHARACTER         80       34      0
[Cmn ]
STANDM                           Array      REAL*4           144   192284      0
[Cmn ] Dims -  1:36
VR                               Scalar     REAL*4             4       22      0
[Cmn ]

Alphabetic List:
Symbol Name                      Class     Data Type    Size      Offset   RefCnt
Other Information
-------------------------------- ---------- ----------- -------- -------- ------ ---
----------------------
A46                              Scalar     CHARACTER         46      338      4
ADAR                             Array      REAL*4           144   192436      4
[Cmn ] Dims -  1:36
ADEN                             Scalar     REAL*4             4   192432      4
[Cmn ]
ANSWER                           Scalar     CHARACTER         80       98      6
B12A                             Scalar     CHARACTER         12      384      2
B12B                             Scalar     CHARACTER         12      396      2
BATCOM                           Scalar     REAL*4             4   192624      0
[Cmn ]
BINC                             Scalar     REAL*4             4        0      1
[Cmn ]
C8A                              Scalar     CHARACTER          8      408      1
C8B                              Scalar     CHARACTER          8      416      1
CONF                             Scalar     CHARACTER         12   192632      4
[Cmn ]
D2                               Scalar     CHARACTER          2      424      1
DIAM                             Scalar     REAL*4             4        4      0
[Cmn ]
FORM                             Scalar     CHARACTER         80      258      2
I                                Scalar     INTEGER*4          4      454     17
IBATCOM                          Scalar     INTEGER*2          2      444      1
IBOTHS                           Scalar     INTEGER*2          2   192628      3
[Cmn ]
IC                               Scalar     INTEGER*2          2               0
ICB                              Scalar     INTEGER*2          2      436      3
ICT                              Scalar     INTEGER*2          2      434      7
IDENS                            Scalar     INTEGER*2          2   192430      4
[Cmn ]
IENV                             Scalar     INTEGER*2          2   192428      0
[Cmn ]
IENVR                            Scalar     INTEGER*2          2      442      1
IPF                              Scalar     INTEGER*2          2      438     15
IPFUNC3                          Scalar     INTEGER*2          2      440      1
ISHLDS                           Parameter  INTEGER*4                          8 =
36
```

IT	Scalar	INTEGER*2	2	8	0
[Cmn]					
ITA	Scalar	INTEGER*2	2	432	2
ITF	Scalar	INTEGER*2	2	430	2
ITYPE	Scalar	INTEGER*2	2	10	3
[Cmn]					
ITYPEIN	Scalar	INTEGER*2	2	192630	2
[Cmn]					
J	Scalar	INTEGER*4	4	458	6
JOT	Scalar	INTEGER*4	4	450	2
K	Scalar	INTEGER*4	4	462	6
LENGTH	Scalar	CHARACTER	2	96	2
NB	Scalar	INTEGER*2	2	12	3
[Cmn]					
NC	Scalar	INTEGER*2	2	14	3
[Cmn]					
NT	Scalar	INTEGER*4	4	16	0
[Cmn]					
NV	Scalar	INTEGER*2	2	20	3
[Cmn]					
PID	Scalar	INTEGER*2	2	114	0
[Cmn]					
PIDS	Array	INTEGER*2	72	116	0
[Cmn] Dims - 1:36					
RANGE	Scalar	REAL*4	4	30	0
[Cmn]					
RESPONSE	Array	REAL*4	191520	188	4
[Cmn] Dims - 1:70 1:19 1:36					
RFILE	Scalar	CHARACTER	80	178	11
ROOTFILE	Scalar	CHARACTER	40	192584	6
[Cmn]					
RSFILE	Scalar	CHARACTER	80	34	0
[Cmn]					
SHDEN	Array	REAL*4	144	191852	2
[Cmn] Dims - 1:36					
SHTHK	Scalar	REAL*4	4	192580	6
[Cmn]					
SHTHKM	Array	REAL*4	144	191708	4
[Cmn] Dims - 1:36					
STANDM	Array	REAL*4	144	192284	0
[Cmn] Dims - 1:36					
STND	Scalar	REAL*4	4	426	4
UNITS	Scalar	CHARACTER	12	192644	3
[Cmn]					
VINC	Scalar	REAL*4	4	26	1
[Cmn]					
VR	Scalar	REAL*4	4	22	0
[Cmn]					
VWDEN	Array	REAL*4	144	192140	2
[Cmn] Dims - 1:36					
VWTHK	Scalar	REAL*4	4	446	4
VWTHKM	Array	REAL*4	144	191996	4
[Cmn] Dims - 1:36					
_$CMNBASES	Cmn Hndls			466	1

Stack Frame Information:

Temporaries List:

Symbol Name / Other Information	Class	Data Type	Size	Offset	RefCnt
_$TEMP10	Scalar	INTEGER*4	4	0	2
_$TEMP9	Scalar	INTEGER*4	4	4	2
_$TEMP8	Scalar	INTEGER*4	4	8	2
_$TEMP7	Scalar	INTEGER*4	4	12	2
_$TEMP6	Scalar	INTEGER*4	4	16	2
_$TEMP5	Scalar	INTEGER*4	4	20	2
_$TEMP4	Scalar	INTEGER*4	4	24	2
_$TEMP3	Scalar	DYNCHAR	4	28	3
_$TEMP2	Scalar	DYNCHAR	4	32	3
_$TEMP1	Scalar	DYNCHAR	4	36	3
{Work Area}			24	40	
_$TLB0	Scalar	LOGICAL*1	1	64	72
_$TIL0	Scalar	INTEGER*4	4	68	14
_$TLB1	Scalar	LOGICAL*1	1	72	6
_$TLB2	Scalar	LOGICAL*1	1	73	6
_$TILAsgn0	Scalar	INTEGER*4	4	76	14
_$TIL1	Scalar	INTEGER*4	4	80	2
_$TE0	Scalar	EXTENDED*12	12	84	6

Variable List:

Symbol Name / Other Information	Class	Data Type	Size	Offset	RefCnt
LENGTH	Scalar	CHARACTER	2	96	2
ANSWER	Scalar	CHARACTER	80	98	6
RFILE	Scalar	CHARACTER	80	178	11
FORM	Scalar	CHARACTER	80	258	2
A46	Scalar	CHARACTER	46	338	4
B12A	Scalar	CHARACTER	12	384	2
B12B	Scalar	CHARACTER	12	396	2
C8A	Scalar	CHARACTER	8	408	1
C8B	Scalar	CHARACTER	8	416	1
D2	Scalar	CHARACTER	2	424	1
STND	Scalar	REAL*4	4	426	4
ITF	Scalar	INTEGER*2	2	430	2
ITA	Scalar	INTEGER*2	2	432	2
ICT	Scalar	INTEGER*2	2	434	7
ICB	Scalar	INTEGER*2	2	436	3
IPF	Scalar	INTEGER*2	2	438	15
IPFUNC3	Scalar	INTEGER*2	2	440	1
IENVR	Scalar	INTEGER*2	2	442	1
IBATCOM	Scalar	INTEGER*2	2	444	1
VWTHK	Scalar	REAL*4	4	446	4
JOT	Scalar	INTEGER*4	4	450	2
I	Scalar	INTEGER*4	4	454	17

```
J                                 Scalar    INTEGER*4        4       458      6
K                                 Scalar    INTEGER*4        4       462      6
_$CMNBASES                        Cmn Hndls                          466      1

Variables in Blank Common:             Size:192656

Symbol Name                       Class      Data Type    Size     Offset   RefCnt
Other Information
--------------------------------- ---------- ----------- -------- -------- ------ ---
----------------------
BINC                              Scalar    REAL*4           4         0      1
[Cmn ]
DIAM                              Scalar    REAL*4           4         4      0
[Cmn ]
IT                                Scalar    INTEGER*2        2         8      0
[Cmn ]
ITYPE                             Scalar    INTEGER*2        2        10      3
[Cmn ]
NB                                Scalar    INTEGER*2        2        12      3
[Cmn ]
NC                                Scalar    INTEGER*2        2        14      3
[Cmn ]
NT                                Scalar    INTEGER*4        4        16      0
[Cmn ]
NV                                Scalar    INTEGER*2        2        20      3
[Cmn ]
VR                                Scalar    REAL*4           4        22      0
[Cmn ]
VINC                              Scalar    REAL*4           4        26      1
[Cmn ]
RANGE                             Scalar    REAL*4           4        30      0
[Cmn ]
RSFILE                            Scalar    CHARACTER       80        34      0
[Cmn ]
PID                               Scalar    INTEGER*2        2       114      0
[Cmn ]
PIDS                              Array     INTEGER*2       72       116      0
[Cmn ] Dims -  1:36
RESPONSE                          Array     REAL*4      191520       188      4
[Cmn ] Dims -  1:70 1:19 1:36
SHTHKM                            Array     REAL*4         144    191708      4
[Cmn ] Dims -  1:36
SHDEN                             Array     REAL*4         144    191852      2
[Cmn ] Dims -  1:36
VWTHKM                            Array     REAL*4         144    191996      4
[Cmn ] Dims -  1:36
VWDEN                             Array     REAL*4         144    192140      2
[Cmn ] Dims -  1:36
STANDM                            Array     REAL*4         144    192284      0
[Cmn ] Dims -  1:36
IENV                              Scalar    INTEGER*2        2    192428      0
[Cmn ]
```

IDENS [Cmn]	Scalar	INTEGER*2	2	192430	4
ADEN [Cmn]	Scalar	REAL*4	4	192432	4
ADAR [Cmn] Dims - 1:36	Array	REAL*4	144	192436	4
SHTHK [Cmn]	Scalar	REAL*4	4	192580	6
ROOTFILE [Cmn]	Scalar	CHARACTER	40	192584	6
BATCOM [Cmn]	Scalar	REAL*4	4	192624	0
IBOTHS [Cmn]	Scalar	INTEGER*2	2	192628	3
ITYPEIN [Cmn]	Scalar	INTEGER*2	2	192630	2
CONF [Cmn]	Scalar	CHARACTER	12	192632	4
UNITS [Cmn]	Scalar	CHARACTER	12	192644	3

Local Stackframe size: 470
Local Symbols: 178

```
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
      SUBROUTINE RPINPUT
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C
C  PSINPUT writes the program header to the screen and reads in the
C  summary ouput filename. It also determines the spacecraft exposure
C  time and operating altitude.
C
C
C
C  note: for variables contained in the common block refer to the main
C        listing for definition
C
C  Variable list
C
C    answer = character string representing user input
C    Psfile = output filenanme
C
C
C
      INCLUDE 'COMMONRP.BLK'
C
      CHARACTER*20 BUMTTM
      CHARACTER*80 ANSWER
```

```
C
C

C
C  Write header to screen
C
      WRITE ( 6,10 )
   10 FORMAT (/1X,'****************************',/1X,3X,
     1     'Space Debris SURFace',/1X,9X,'R_PLOT5'
     2     /1X,5X,'Ver. 1.6 8/23/92',/1X,5X,'FOR BUMPERIIv1.3',/1X,
     3            '****************************')
C
C  Read in output filename, set default to R_PLOT5.RS
C
   15 WRITE ( 6,20 )
   20 FORMAT ( /1X,'OUTPUT FILENAME (CR=R_PLOT5.RS)>',$)
      READ ( 5,30 )RSFILE
   30 FORMAT (A)
C
      IF ( RSFILE(1:1).EQ.' ' ) RSFILE='R_PLOT5.RS'
C
C  Open rsfile
C
C  !!!! PUT CREATOR='XCEL' OR 'MSWD' IN OPEN STATEMENTS ON MAC
C
      OPEN ( UNIT=10,FILE=RSFILE,STATUS='NEW',IOSTAT=IER,
     *       CREATOR='XCEL',ERR=40,RECL=256 )
C
      GO TO 70
C
C  Error control
C
   40 IF ( IER.EQ.2013 ) THEN
         WRITE ( 6,50 )
   50    FORMAT ( /1X,'FILE ALREADY EXISTS OK TO OVERWRITE (CR=YES,$)>')
         READ ( 5,30 ) ANSWER
C
         IF ( ANSWER(1:1).EQ.'Y' .OR. ANSWER(1:1).EQ.' ' ) THEN
            OPEN ( UNIT=10,FILE=RSFILE,STATUS='UNKNOWN',IOSTAT=IER,
     1             ERR=40)
            REWIND 10
         ELSE
            GO TO 15
         END IF
      ELSE
         WRITE ( 6,60 )
   60    FORMAT (/1X,'UNABLE TO OPEN FILE ' )
         GO TO  15
      END IF
C
   70 CONTINUE
C
C
      WRITE ( 10,75 )
```

```
      75 FORMAT (/1X,'****************************',//1X,3X,
     1  '          R_PLOT5',
     2     /1X,5X,'Ver. 1.6 8/23/92',/1X,5X,'FOR BUMPERIIv1.3',/1X,
     3          '****************************'//)
C
C
        RETURN
        END
```

Symbol Table for: RPINPUT

The following symbols were defined but NOT referenced:

```
Symbol Name                     Class      Data Type    Size     Offset   RefCnt
Other Information
------------------------------- ---------- ------------ -------- -------- ------ ---
----------------------
ADAR                            Array       REAL*4           144   192436      0
[Cmn ] Dims -  1:36
ADEN                            Scalar      REAL*4             4   192432      0
[Cmn ]
BATCOM                          Scalar      REAL*4             4   192624      0
[Cmn ]
BINC                            Scalar      REAL*4             4        0      0
[Cmn ]
BUMTTM                          Scalar      CHARACTER         20                 0
CONF                            Scalar      CHARACTER         12   192632      0
[Cmn ]
DIAM                            Scalar      REAL*4             4        4      0
[Cmn ]
IBATCOM                         Scalar      INTEGER*2          2                 0
IBOTHS                          Scalar      INTEGER*2          2   192628      0
[Cmn ]
IDENS                           Scalar      INTEGER*2          2   192430      0
[Cmn ]
IENV                            Scalar      INTEGER*2          2   192428      0
[Cmn ]
IT                              Scalar      INTEGER*2          2        8      0
[Cmn ]
ITYPE                           Scalar      INTEGER*2          2       10      0
[Cmn ]
ITYPEIN                         Scalar      INTEGER*2          2   192630      0
[Cmn ]
NB                              Scalar      INTEGER*2          2       12      0
[Cmn ]
NC                              Scalar      INTEGER*2          2       14      0
[Cmn ]
NT                              Scalar      INTEGER*4          4       16      0
[Cmn ]
NV                              Scalar      INTEGER*2          2       20      0
[Cmn ]
PID                             Scalar      INTEGER*2          2      114      0
[Cmn ]
```

```
PIDS                            Array      INTEGER*2         72      116       0
[Cmn ] Dims -   1:36
RANGE                           Scalar     REAL*4             4       30       0
[Cmn ]
RESPONSE                        Array      REAL*4        191520      188       0
[Cmn ] Dims -   1:70 1:19 1:36
ROOTFILE                        Scalar     CHARACTER         40   192584       0
[Cmn ]
SHDEN                           Array      REAL*4           144   191852       0
[Cmn ] Dims -   1:36
SHTHK                           Scalar     REAL*4             4   192580       0
[Cmn ]
SHTHKM                          Array      REAL*4           144   191708       0
[Cmn ] Dims -   1:36
STANDM                          Array      REAL*4           144   192284       0
[Cmn ] Dims -   1:36
UNITS                           Scalar     CHARACTER         12   192644       0
[Cmn ]
VINC                            Scalar     REAL*4             4       26       0
[Cmn ]
VR                              Scalar     REAL*4             4       22       0
[Cmn ]
VWDEN                           Array      REAL*4           144   192140       0
[Cmn ] Dims -   1:36
VWTHK                           Scalar     REAL*4             4                0
VWTHKM                          Array      REAL*4           144   191996       0
[Cmn ] Dims -   1:36

Alphabetic List:
Symbol Name                     Class     Data Type    Size     Offset   RefCnt
Other Information
------------------------------- ---------- ----------- -------- -------- ------ ---
----------------------
ADAR                            Array      REAL*4           144   192436       0
[Cmn ] Dims -   1:36
ADEN                            Scalar     REAL*4             4   192432       0
[Cmn ]
ANSWER                          Scalar     CHARACTER         80       27       3
BATCOM                          Scalar     REAL*4             4   192624       0
[Cmn ]
BINC                            Scalar     REAL*4             4        0       0
[Cmn ]
BUMTTM                          Scalar     CHARACTER         20                0
CONF                            Scalar     CHARACTER         12   192632       0
[Cmn ]
DIAM                            Scalar     REAL*4             4        4       0
[Cmn ]
IBATCOM                         Scalar     INTEGER*2          2                0
IBOTHS                          Scalar     INTEGER*2          2   192628       0
[Cmn ]
IDENS                           Scalar     INTEGER*2          2   192430       0
[Cmn ]
IENV                            Scalar     INTEGER*2          2   192428       0
[Cmn ]
```

```
IER                                     Scalar      INTEGER*4          4      107      3
ISHLDS                                  Parameter   INTEGER*4                          8 =
36
IT                                      Scalar      INTEGER*2          2        8      0
[Cmn ]
ITYPE                                   Scalar      INTEGER*2          2       10      0
[Cmn ]
ITYPEIN                                 Scalar      INTEGER*2          2   192630      0
[Cmn ]
NB                                      Scalar      INTEGER*2          2       12      0
[Cmn ]
NC                                      Scalar      INTEGER*2          2       14      0
[Cmn ]
NT                                      Scalar      INTEGER*4          4       16      0
[Cmn ]
NV                                      Scalar      INTEGER*2          2       20      0
[Cmn ]
PID                                     Scalar      INTEGER*2          2      114      0
[Cmn ]
PIDS                                    Array       INTEGER*2         72      116      0
[Cmn ] Dims -  1:36
RANGE                                   Scalar      REAL*4             4       30      0
[Cmn ]
RESPONSE                                Array       REAL*4        191520      188      0
[Cmn ] Dims -  1:70 1:19 1:36
ROOTFILE                                Scalar      CHARACTER         40   192584      0
[Cmn ]
RSFILE                                  Scalar      CHARACTER         80       34      5
[Cmn ]
SHDEN                                   Array       REAL*4           144   191852      0
[Cmn ] Dims -  1:36
SHTHK                                   Scalar      REAL*4             4   192580      0
[Cmn ]
SHTHKM                                  Array       REAL*4           144   191708      0
[Cmn ] Dims -  1:36
STANDM                                  Array       REAL*4           144   192284      0
[Cmn ] Dims -  1:36
UNITS                                   Scalar      CHARACTER         12   192644      0
[Cmn ]
VINC                                    Scalar      REAL*4             4       26      0
[Cmn ]
VR                                      Scalar      REAL*4             4       22      0
[Cmn ]
VWDEN                                   Array       REAL*4           144   192140      0
[Cmn ] Dims -  1:36
VWTHK                                   Scalar      REAL*4             4               0
VWTHKM                                  Array       REAL*4           144   191996      0
[Cmn ] Dims -  1:36
_$CMNBASES                              Cmn Hndls                             111      1

Stack Frame Information:

Temporaries List:
```

```
Symbol Name                      Class      Data Type    Size     Offset   RefCnt
Other Information
-------------------------------- ---------- ------------ -------- -------- ------ ---
----------------------
{Work Area}                                                  24        0
_$TLB0                           Scalar      LOGICAL*1        1       24        6
_$TLB1                           Scalar      LOGICAL*1        1       25        2
_$TLB2                           Scalar      LOGICAL*1        1       26        2

Variable List:
Symbol Name                      Class      Data Type    Size     Offset   RefCnt
Other Information
-------------------------------- ---------- ------------ -------- -------- ------ ---
----------------------
ANSWER                           Scalar      CHARACTER       80       27        3
IER                              Scalar      INTEGER*4        4      107        3
_$CMNBASES                       Cmn Hndls                           111        1


Variables in Blank Common:            Size:192656

Symbol Name                      Class      Data Type    Size     Offset   RefCnt
Other Information
-------------------------------- ---------- ------------ -------- -------- ------ ---
----------------------
BINC                             Scalar      REAL*4           4        0        0
[Cmn ]
DIAM                             Scalar      REAL*4           4        4        0
[Cmn ]
IT                               Scalar      INTEGER*2        2        8        0
[Cmn ]
ITYPE                            Scalar      INTEGER*2        2       10        0
[Cmn ]
NB                               Scalar      INTEGER*2        2       12        0
[Cmn ]
NC                               Scalar      INTEGER*2        2       14        0
[Cmn ]
NT                               Scalar      INTEGER*4        4       16        0
[Cmn ]
NV                               Scalar      INTEGER*2        2       20        0
[Cmn ]
VR                               Scalar      REAL*4           4       22        0
[Cmn ]
VINC                             Scalar      REAL*4           4       26        0
[Cmn ]
RANGE                            Scalar      REAL*4           4       30        0
[Cmn ]
RSFILE                           Scalar      CHARACTER       80       34        5
[Cmn ]
PID                              Scalar      INTEGER*2        2      114        0
[Cmn ]
PIDS                             Array       INTEGER*2       72      116        0
[Cmn ] Dims -   1:36
```

```
RESPONSE                        Array     REAL*4      191520     188     0
[Cmn ] Dims -  1:70 1:19 1:36
SHTHKM                          Array     REAL*4         144  191708     0
[Cmn ] Dims -  1:36
SHDEN                           Array     REAL*4         144  191852     0
[Cmn ] Dims -  1:36
VWTHKM                          Array     REAL*4         144  191996     0
[Cmn ] Dims -  1:36
VWDEN                           Array     REAL*4         144  192140     0
[Cmn ] Dims -  1:36
STANDM                          Array     REAL*4         144  192284     0
[Cmn ] Dims -  1:36
IENV                            Scalar    INTEGER*2        2  192428     0
[Cmn ]
IDENS                           Scalar    INTEGER*2        2  192430     0
[Cmn ]
ADEN                            Scalar    REAL*4           4  192432     0
[Cmn ]
ADAR                            Array     REAL*4         144  192436     0
[Cmn ] Dims -  1:36
SHTHK                           Scalar    REAL*4           4  192580     0
[Cmn ]
ROOTFILE                        Scalar    CHARACTER       40  192584     0
[Cmn ]
BATCOM                          Scalar    REAL*4           4  192624     0
[Cmn ]
IBOTHS                          Scalar    INTEGER*2        2  192628     0
[Cmn ]
ITYPEIN                         Scalar    INTEGER*2        2  192630     0
[Cmn ]
CONF                            Scalar    CHARACTER       12  192632     0
[Cmn ]
UNITS                           Scalar    CHARACTER       12  192644     0
[Cmn ]


Local Stackframe size: 116
Local Symbols: 58
```

```
C
C
      SUBROUTINE DIRLIST
      CHARACTER*80 LINE
      OPEN(UNIT=17,FILE='DIRECTORY.LIST',STATUS='OLD',ERR=30)
      REWIND 17
5     READ(17,10,ERR=20) LINE
      WRITE(6,11) LINE
      GOTO 5
20    REWIND 17
      CLOSE(UNIT=17,STATUS='KEEP')
10    FORMAT ( A80)
11    FORMAT ( 1X,A80)
```

```
0621    30     RETURN
0622           END


Symbol Table for: DIRLIST

Alphabetic List:
Symbol Name                      Class      Data Type    Size     Offset   RefCnt
Other Information
-------------------------------- ---------- ------------ -------- -------- ------ ---
----------------------
LINE                             Scalar      CHARACTER         80       24      2

Stack Frame Information:

Temporaries List:
Symbol Name                      Class      Data Type    Size     Offset   RefCnt
Other Information
-------------------------------- ---------- ------------ -------- -------- ------ ---
----------------------
{Work Area}                                                    24        0

Variable List:
Symbol Name                      Class      Data Type    Size     Offset   RefCnt
Other Information
-------------------------------- ---------- ------------ -------- -------- ------ ---
----------------------
LINE                             Scalar      CHARACTER         80       24      2


Local Stackframe size: 104
Local Symbols: 8


Global Symbol Table
Symbol Name                      Class       Result Type Size     Other Information
-------------------------------- ----------- ----------- -------- -------------------
--
Blank Common                     Blank Cmn                 192656
DIRLIST                          Proc Sub
MACII_R_plot5_16$main            Main Prgm
RESREAD                          Proc Sub
RP5TEXT                          Proc Sub
RPINPUT                          Proc Sub

    0 serious errors detected.
    0 warning messages generated.
  754 lines compiled.
```

SD_SURF User's Manual

Appendix D. BUMPERII Modifications for the Macintosh

Limitations

These are the limits in version 3.0 of the Language Systems FORTRAN Compiler:

31	significant characters in a symbolic name (ANSI 77 allows 6)
255	characters in each source code line (ANSI 77 allows 72)
5100	characters in a statement (counting all continuation lines, but not counting comment lines)
409	global symbols (program, subprogram and common block names) in one compile
~5500	local symbols (including all symbolic names, statement labels, subprogram and function references and compiler-generated temporaries)
7	dimensions in a single array
3200	combined array dimensions in a program module
32	levels of nested DO loops and nested implied DO loops
50	nesting depth for block IF statements
32	arguments in a statement function definition
512	actual arguments per CALL reference
20	nested function calls and subscript references
20	nested repeat factors in a format
1500	characters in a packed format
32767	real constants in a program module
32767	complex constants in a program module
32767	character constants in a program module
2147483647	maximum record size for multiple items in an unformatted I/O statement
2147483647	maximum record size for formatted I/O
341	fields in any structure
32767	maximum size of a STRUCTURE element in an array of RECORDS
2147483647	maximum iterations for a DO loop

Appendix D

VAX Extensions That Are Accepted But Not Executed:

```
CLOSE options
      DISPOSE/DISP= 'SUBMIT'
                    'SUBMIT/DELETE'
DEFINEFILE
DELETE
DICTIONARY
FIND
INQUIRE options
      DEFAULTFILE
      KEYED
      ORGANIZATION=  'SEQUENTIAL'
                     'RELATIVE'
                     'INDEXED'

      RECORDTYPE=    'FIXED'
                     'VARIABLE'
                     'STREAM_CR'
OPEN options
      ACCESS=  'KEYED'
      ASSOCIATEVARIABLE
      BLOCKSIZE
      BUFFERCOUNT
      DEFAULTFILE
      DISPOSE/DISP=  'SUBMIT'
                     'SUBMIT/DELETE'
      EXTENDSIZE
      INITIALSIZE
      KEY
      NOSPANBLOCKS
      ORGANIZATION=  'SEQUENTIAL'
                     'RELATIVE'
                     'INDEXED'
      RECORDTYPE=    'FIXED'
                     'VARIABLE'
                     'STREAM_CR'
                     'STREAM_LF'
      SHARED
      USEROPEN
READ/WRITE
      KEYID
REWRITE
UNLOCK
```

VAX VMS FORTRAN Features Not Supported:

PARAMETER statements of the form: PARAMETER p=c, [p=c]

Octal constant notation: "77 or "77" (many other types of octal notation are available)

Extended Range DO loops

Indexed files

Radix-50 constants

External BLOCK DATA subprograms

EXTERNAL *v [,*v]

TYPE FUNCTION *NAME** V*

CALL a function

SD_SURF User's Manual

Appendix E. SD_SURF Macro

Appendix E - SD_SURF_MACRO

	A	B	C
1		**Summary Information**	
2	*Title:*	Space Debris Surfaces Macro	
3	*Contract:*	NAS8-38856	
4	*Version:*	v1.1	
5	*Programmer:*	Norman Elfer, Ph.D. (504)-257-3162	
6	*Corporation:*	Martin Marietta Manned Space Systems	
7	*Creation Date:*	Ver 1.1 - Feb. 14, 1992	
8			
9	*Notice*	This series of EXCEL Macros were written in	
10		support of contract NAS8-38856 from	
11		NASA-Marshall Space Flight Center.	
12			
13			
14	***SUBROUTINES***		
15	***NAME***	PURPOSE	
16	*Auto_open*	Calls Opening Dialog Box.	
17		Opens Function Macro Sheets.	
18	*Auto_close*	Deletes menu.	
19	*A_surf_0.5_km/s*	Changes A_surf output to 0.5 km/s increments	
20	*Close_Macro*	Close SD Surf from menu bar. Continue EXCEL	
21	*Open/Save/Set...*	Function described and sets variable name.	
22	*BL_PASTE*	Pastes named arrays from Ballistic Limit to PNP	
23	*RPLOT_Open*	Opens FORTRAN text file output and pastes to PNP	
24			
25	***Auto_open***	**Auto_open**	
26	***SD_Surf.name***	=GET.DOCUMENT(1)	
27		=MESSAGE(1,"SD Pull Down Menu added. Unhide macro to modify.")	
28		=CALCULATION(3)	
29		=SHORT.MENUS(FALSE)	
30		=ADD.MENU(1,SD)	
31		=DIALOG.BOX(Intro_Dialog_box)	
32		=ALERT("Automatic Recalculation was turned off. See Options. If you don't want to recalculate when saving use Apple-period to stop re-calc.",2)	
33		=MESSAGE(0,)	
34		=HIDE()	
35		=ERROR(2,OPEN_SD_FUNCTION MACRO)	
36		=OPEN?("SD_Function_Macros",,TRUE)	*Open Function Macro Sheets*
37		=HIDE()	
38		=ERROR(1)	
39		=Open_PNP_Template()	*Calls a subroutine*
40		=RETURN()	
41			
42	*auto_close*	**close_Macro**	
43		=ACTIVATE(SD_Surf.name)	
44		=CLOSE()	
45		=RETURN()	
46			
47	*auto_close*	**auto_close**	
48		=DELETE.MENU(1,"SD")	
49		=RETURN()	
50			
51	*command*	**Open SD Function Macro**	

	A	B	C
52		=MESSAGE(1,"Please open the SD Funtion Macro Sheet. It will be hidden.")	*General open box used*
53		=OPEN?("SD_Function_Macros",,TRUE)	
54		=HIDE()	
55		=MESSAGE(0,)	
56		=Open_PNP_Template()	*Calls a subroutine*
57		=RETURN()	
58			
59			
60	*command*	**Open BL Template**	
61		=MESSAGE(1,"Please open a Ballistic Limit Macro TEMPLATE. Default is READ ONLY.")	*General open box used if*
62	*open_template*	=OPEN?("BL Template",,TRUE)	*user wants to select*
63		=MESSAGE(0,)	*a previously modified*
64		= IF(open_BLTemplate =FALSE,HALT(),)	*template.*
65	*BL_Template.Name*	=GET.DOCUMENT(1)	
66		=FULL(TRUE)	
67		=FORMULA.GOTO(!$A$1,TRUE)	
68		=RETURN()	
69			
70	*command*	**Save BL Template**	
71		=MESSAGE(1,"Save with new or old name. Apple-. to stop recalculation.")	
72		=SAVE.AS?()	
73		=IF(B73 =FALSE,HALT(),)	
74		=MESSAGE(0,)	
75	*command*	**Set BL Template**	
76		=SET.VALUE(BL_Template.Name,GET.DOCUMENT(1))	
77		=RETURN()	
78			
79	*command*	**Open PNP Template**	
80		=MESSAGE(1,"Please open a PNP TEMPLATE. Default is READ ONLY.")	*General open box used if*
81		=OPEN?("PNP/FLUX Template",,TRUE)	*user wants to select*
82		= IF(B81 =FALSE,HALT(),)	*a previously modified*
83		=MESSAGE(0,)	*template.*
84	*PNP_Template.Nam*	=GET.DOCUMENT(1)	
85		=FULL(TRUE)	
86		=FORMULA.GOTO(!$A$1,TRUE)	
87		=RETURN()	
88			
89	*command*	**Save PNP Template**	
90		=MESSAGE(1,"Save with new or old name. Apple-. to stop recalculation.")	
91		=SAVE.AS?()	
92		= IF(B92 =FALSE,HALT(),)	
93		=MESSAGE(0,)	
94	*command*	**Set PNP Template**	
95		=SET.VALUE(PNP_Template.Name,GET.DOCUMENT(1))	
96		=RETURN()	
97			
98	*command*	**Open Area Maker**	
99		=MESSAGE(1,"Please open Area Maker Macro sheet.")	*General open box used if*
100		=OPEN?("AREA_MAKER_MACRO",,TRUE)	*user wants to select*
101		= IF(B100 =FALSE,HALT(),)	*a previously modified*

	A	B	C
102		=MESSAGE(0,)	*template.*
103		=RUN(!Auto_Open,FALSE)	
104		=RETURN()	
105			
106	*A_surf_0.5_km/s*	**A surf 0.5 km s**	
107		• Use with A_SURF output.	
108		• Cut from 0.25 km/s and add	
109		to 0.5 km/s multiples	
110		• Start on first row (0.25 km/s)	
111	*A_Alert*	=ALERT("This will delete every other 0.25km/s A_Surf entry. You must have selected the 1st row to delete.",1)	*Alert - You can change*
112		=IF(A_Alert=FALSE(),HALT(),)	*your mind here.*
113		=ECHO(FALSE)	*Speeds up Macro*
114			
115		=SELECT("R[+0]C2:R[+0]C20")	*Adds first row to second and deletes first row*
116		=COPY()	
117		=SELECT("R[+1]C2")	
118		=PASTE.SPECIAL(3,2,FALSE,FALSE)	
119		=SELECT("R[-1]")	
120		=EDIT.DELETE(2)	
121	*counter*	=FOR("counter",1,33,1)	*Counter Loop*
122		= SELECT("R[+1]")	*Sets up to divide by2*
123		= INSERT(2)	
124		= SELECT("RC2:RC20")	
125		= FORMULA("0.5")	
126		= FILL.RIGHT()	
127		= SELECT("R[+1]C2:R[+1]C20")	*Divides odd cells by 2*
128		= COPY()	
129		= SELECT("R[-1]C2:R[-1]C20")	
130		= PASTE.SPECIAL(3,4,FALSE,FALSE)	
131		= COPY()	*Adds to cells above and below*
132		= SELECT("R[-1]C2")	
133		= PASTE.SPECIAL(3,2,FALSE,FALSE)	
134		= SELECT("R[+3]C2")	
135		= PASTE.SPECIAL(3,2,FALSE,FALSE)	
136		= SELECT("R[-2]")	
137		= EDIT.DELETE(2)	*Deletes cells*
138		= EDIT.DELETE(2)	
139		=NEXT()	*End Loop*
140		=ECHO(TRUE)	
141		=RETURN()	
142			
143	*BL_PASTE*	**BL PASTE**	
144	*BLP_Alert*	=ALERT("This will paste the ballistic limit surface on the PNP/FLUX WS Template. The ballistic limit template must be active.",1)	*Alert - You can change*
145		=IF(BLP_Alert =FALSE(),HALT(),)	*your mind here.*
146		=ECHO(FALSE)	*Speeds up Macro*
147		=CALCULATION(3,,,,,,)	
148		=SET.NAME("BL.name",GET.DOCUMENT(1))	
149		=ACTIVATE(BL.name)	*Copy and paste Header*
150		=SELECT("r6c5:r16c6")	
151		=COPY()	
152		=ACTIVATE(PNP_Template.Name)	
153		=SELECT("R16C7")	

	A	B	C
154		=PASTE.SPECIAL(3,1,FALSE,FALSE)	
155		=ACTIVATE(BL.name)	*Copy and paste Sample calc*
156		=SELECT("R6C10:R10C11")	
157		=COPY()	
158		=ACTIVATE(PNP_Template.Name)	
159		=SELECT("R17C21")	
160		=PASTE.SPECIAL(3,1,FALSE,FALSE)	
161		=SELECT("R15C21")	
162		=FORMULA("Worksheet calculation")	
163		=SELECT("R16C21")	
164		=FORMULA(BL.name)	
165		=ACTIVATE(BL.name)	*Copy and paste BL data*
166		=SELECT("R21C4:R52C22")	
167		=COPY()	
168		=ACTIVATE(PNP_Template.Name)	
169		=SELECT("R44C4")	
170		=PASTE.SPECIAL(3,1,FALSE,FALSE)	
171		=CALCULATION(1)	
172		=ECHO(TRUE)	
173		=RETURN()	
174			
175	*RPLOT_Open*	RPLOT_Open	
176		=WORKSPACE(,,,,TRUE,,,,,)	*Sets up message box.*
177		=MESSAGE(TRUE,"Directions will be listed HERE.")	
178		=ALERT("See MESSAGE box at lower left.",2)	
179		=MESSAGE(TRUE,"Open the RPLOT Data File")	*Open Response/R-Plot*
180		=OPEN?(,,,2)	
181		=IF(B182 =FALSE(),HALT(),)	
182		=SET.NAME("RPLOT.name",GET.DOCUMENT(1))	
183		x MESSAGE(TRUE,"Open the RPLOT SUMMARY File")	*x Open Response Summary*
184		x OPEN?(,,,2)	*The summary file is not*
185		x IF(B178 =FALSE(),HALT(),)	*used due to current*
186		x SET.NAME("RSUM.name",GET.DOCUMENT(1))	*output structure.*
187		=MESSAGE(TRUE,"Open the R-PLOT Ballistic Limit Template (BL-RPLOT)")	*Open BL Template*
188		=OPEN?("BL-RPLOT",,TRUE,,)	
189		=IF(B190 =FALSE(),HALT(),)	
190		=SET.NAME("BL.name",GET.DOCUMENT(1))	
191		=ACTIVATE(RPLOT.name)	*Tansfer Response Data*
192		=SELECT(INPUT("Select first diameter on Response Table. (v =0.25 & obl = 0.)",8,"Response Table",,,))	
193		=IF(B194 =FALSE,HALT(),)	
194		=SELECT("RC:R[35]C[20]")	
195		=COPY()	
196		=ACTIVATE(BL.name)	
197		=SELECT("R17C3")	
198		=PASTE.SPECIAL(3,1,FALSE,FALSE)	
199		x ACTIVATE(RSUM.name)	*Transfer Response*
200		=ACTIVATE(RPLOT.name)	*Summary*
201		=SELECT(INPUT("Select up to 12 lines of description to be paste to the Template.",8,"Response Description","R1C13:RC22",,))	
202		=IF(B203 =FALSE,HALT(),)	
203		=COPY()	
204		=ACTIVATE(BL.name)	

	A	B	C
205		=SELECT("R2C5")	
206		=PASTE.SPECIAL(3,1,FALSE,FALSE)	
207		=SELECT("R4C11")	*Record File Names*
208		=FORMULA("RESPONSE OUTPUT FILES")	
209		x SELECT("R5C11")	
210		x FORMULA(RSUM.name)	
211		=SELECT("R6C11")	
212		=FORMULA(RPLOT.name)	
213		=ACTIVATE(RPLOT.name)	*CLOSE TEXT FILES*
214		=CLOSE(FALSE)	
215		x ACTIVATE(RSUM.name)	
216		x CLOSE(FALSE)	
217		=ECHO(TRUE)	*SAVE BALLISTIC LIMIT WS*
218		=MESSAGE(TRUE,"Enter Name to Save Ballistic Limit in EXCEL Format.")	
219		=SAVE.AS?(RPLOT.name,1,"",FALSE)	
220		=MESSAGE(FALSE)	
221		=IF(B221 =FALSE(),HALT(),)	
222		=SET.NAME("BL.name",GET.DOCUMENT(1))	
223		=ALERT("OK to do PNP Calculation? This will take a few minutes!",1)	*CONTINUE TO PNP?*
224		=IF(B225 =FALSE(),HALT(),)	
225		=MESSAGE(TRUE,"Open the PNP/FLUX WS TEMPLATE. Cancel to use current Template.")	*Open PNP Template*
226		=OPEN?("PNP/FLUX WS TEMPLATE",,TRUE,,)	
227		=IF(B228 =FALSE,ACTIVATE(PNP_Template.Name),)	
228		=MESSAGE(TRUE,"Transfering data and calculating flux and PNP.")	
229		=SET.NAME("PNP.name",GET.DOCUMENT(1))	
230		=ECHO(FALSE)	*Speed up macro*
231		=CALCULATION(3,,,,,,)	
232		=ACTIVATE(BL.name)	*Copy & paste BL_Header_1*
233		=SELECT("r2c5:r14c6")	
234		=COPY()	
235		=ACTIVATE(PNP.name)	
236		=SELECT("R15C4")	
237		=PASTE.SPECIAL(3,1,FALSE,FALSE)	
238		=ACTIVATE(BL.name)	*Copy & paste BL_Header_2*
239		=SELECT("R4C11:R10C11")	
240		=COPY()	
241		=ACTIVATE(PNP.name)	
242		=SELECT("R15C19")	
243		=PASTE.SPECIAL(3,1,FALSE,FALSE)	
244		=ACTIVATE(BL.name)	*Copy and paste BL data*
245		=SELECT("R19C4:R50C22")	
246		=COPY()	
247		=ACTIVATE(PNP.name)	
248		=SELECT("R44C4")	
249		=PASTE.SPECIAL(3,1,FALSE,FALSE)	
250		=CALCULATION(1)	*Perform Calculations*
251		=ECHO(TRUE)	
252		=MESSAGE(FALSE)	
253		=RETURN()	
254			
255			

	E	F	G	H	I	J	K	L	M
1		DIALOGS							
2		type	x	y	widt	high	text	init/result	*names*
3					337	374	SD Surf for EXCEL 3.0		
4	*OK Button*	1	242	330	64		OK		
5	*Text*	5	26	8			Space Debris Surfaces - SD SURF MACRO		
6	*Text*	5	26	31			Ver 1.1 - Feb. 14, 1992		
7	*Text*	5	26	50			------------------------------------		
8	*Text*	5	26	69			STRUCTURAL DAMAGE PREDICTION AND		
9	*Text*	5	26	92			ANALYSIS FOR HYPERVELOCITY IMPACTS		
10	*Text*	5	26	115			Contract NAS8-38856		
11	*Text*	5	26	134			------------------------------------		
12	*Text*	5	26	153			NASA - Marshall Space Flight Center		
13	*Text*	5	26	176			Technical Monitors:		
14	*Text*	5	56	199			Greg Olsen		
15	*Text*	5	56	222			Jennifer Robinson		
16	*Text*	5	56	245			Joel Williamsen		
17	*Text*	5	26	272			Martin Marietta Manned Space Systems		
18	*Text*	5	26	295			Program Manager:		
19	*Text*	5	56	318			Norman Elfer		
20	*Text*	5	56	341			(504)-257-3162		

	O	P	Q	R	S	T
1	MENUS					
2	Nam	Command	Macro	K	Status Bar Text	Help
3		SD				
4		Open R_Plot Output	RPLOT_Open		Opens R_PLOT output and Pastes to PNP/Flux Template	
5		-				
6		Open BL Template	Open_BL_Template		Keeps track of file to use as Ballistic Limit template.	
7		Ballistic Limit to PNP_T	BL_Paste		Copy Ballistic Limit from Active Template to PNP_Templat	
8		-				
9		Open Area Maker Macro	Open_Area_Maker		Area Maker opens A_SURF output & creates new geometries	
10		-				
11		Open PNP/Flux Template	Open_PNP_Template		Keeps track of which file to use as PNP/FLUX template.	
12		Save PNP/Flux Template	Save_PNP_Template		Keeps track of which file to use as PNP/FLUX template.	
13		Set PNP/Flux Template	Set_PNP_Template		Keeps track of which file to use as PNP/FLUX template.	
14		-				
15		Close SD Surf Macro	close_macro		Closes Macro and deletes SD menu.	

	V	W	X
2	VARIABLES	REFERENCE	TYPE
3			
4	Auto_Close	=$B$48:$B$49	0
5	Auto_Open	=$B$25:$B$38	2
6	A_Alert	=$A$111:$B$111	0
7	A_surf_0.5_km_s	=$B$106:$B$141	2
8	BL.name	="R_PLOT5.RS xl"	0
9	BLP_Alert	=$A$144:$B$144	0
10	BL_PASTE	=$B$144:$B$173	2
11	BL_Template.Name	=$B$65	0
12	close_Macro	=$B$43:$B$45	0
13	counter	=34	0
14	Intro_Dialog_box	=$F$3:$L$20	0
15	*Open_Area_Maker*	*=$B$99:$B$104*	*2*
16	*Open_BL_Template*	*=$B$61:$B$68*	*2*
17	*Open_PNP_Template*	*=$B$80:$B$87*	*2*
18	*OPEN_SD_FUNCTION_MACRO*	*=$B$51:$B$57*	*2*
19	*PNP.name*	*="PNP TEMPLATE"*	*0*
20	PNP_Template.Name	=$B$84	0
21	Print_Area	=$A$1:$C$255,$E$1:$M$20,$O	0
22	RPLOT.name	="R_PLOT5.RS"	0
23	RPLOT_Open	=$B$175:$B$253	2
24	RSUM.name	="R063125M.sum"	0
25	Save_BL_Template	=$B$71:$B$77	2
26	Save_PNP_Template	=$B$90:$B$96	2
27	SD	=$P$3:$T$15	0
28	SD_Surf.name	=$B$26	0
29	Set_BL_Template	=$B$76:$B$77	2
30	Set_PNP_Template	=$B$95:$B$96	2

SD_SURF User's Manual

Appendix F. AREA_MAKER Macro

	A	B	C
1		**Summary Information**	
2	*Title:*	Space Debris Surfaces-Area Maker Macro	
3	*Contract:*	NAS8-38856	
4	*Version:*	v1.1	
5	*Programmer:*	Norman Elfer, Ph.D. (504)-257-3162	
6	*Corporation:*	Martin Marietta Manned Space Systems	
7	*Creation Date:*	February 14, 1992	
8			
9	*Notice*	This series of EXCEL Macros were written in	
10		support of contract NAS8-38856 from	
11		NASA-Marshall Space Flight Center.	
12			
13			
14	*COMMAND MACROS*		
15	*NAME*	PURPOSE	
16	*Auto_open*	Calls Opening Dialog Box and adds Pull Down Menus.	
17		Opens Area Template Worksheet	
18		Opens Function Macro Sheets.	
19	*Auto_close*	Removes Menus	
20	*Set_Template*	Identifies active document as the Area Template	
21	*Template_Open*	Opens the Area Template	
22	*Template_Save*	Saves Template and Identifies it as the Area Template	
23	*Clear_Area_Array*	Clears Area Array & Descriptions on Template	
24	*Rectangle*	Creates Area Array and descriptions on Area	
25	*Disk*	Template. This is done by opening dialog boxes	
26	*Cylinder*	for user input and creating facets which are sent	
27	*Disk*	to Rotate and then to Area_Matrix.	
28	*Cone*		
29	*Sphere*		
30	*AreaS_to_PNP*	Copies Area_array and Description_Array to	
31		PNP Template.	
32	*A_Plot_Manipulatio*	Open A_Plot text file and compresses for EXCEL.	
33	*Close_AreaS*	Closes macro which will start auto_close.	
34			
35	*FUNCTION MACROS*		*Input /output*
36	*Area_Matrix*	Adds facets to area array on Area Template	*Area,Phi,Theta*
37			*/Total Projected Area*
38	*Rotate*	Rotates facet orientation	*Phi, Theta, Pitch,*
39			*Yaw / Phi, Theta*
40	*Velocity_Dist*	Calculates probability distribution, f(v). Needs to be	*velocity, orbital inclination*
41		normalized.	*/ f(v)*
42			
43			
44	*Auto_open*	Auto_open	*Open Area Template*
45	*A_Maker_name*	=GET.DOCUMENT(1)	
46		=HIDE()	*Add menu*
47		=ADD.MENU(1,AreaS)	
48		=Reset_initial_values()	*Reset Dialog inital values*
49		=CALCULATION(3)	*Turn Calculation Off*
50		=SHORT.MENUS(FALSE)	
51		=DIALOG.BOX(INTRO_DIALOG_BOX)	
52		=SET.VALUE(A_Template,"Area Template")	

	A	B	C
53		=ERROR(2,TEMPLATE_OPEN)	*Make sure Area Template*
54		=ACTIVATE(A_Template)	*is open*
55		=ERROR(1)	
56		=FULL(TRUE)	
57		=FORMULA.GOTO(!$A$1,TRUE)	
58		=RETURN()	
59			
60	*command*	**TEMPLATE_OPEN**	*Called by Auto_open Error*
61		=MESSAGE(1,"Please open any AREA TEMPLATE. Default is READ ONLY.")	*General open box used if*
62		=ERROR(2)	*user wants to select*
63	*open_template*	=OPEN?("Area Template",,TRUE)	*a previously modified*
64		=MESSAGE(0,)	*template.*
65		=ERROR(1)	
66		= IF(open_template =FALSE,Alert1(),)	
67	*A_TEMPLATE*	=GET.DOCUMENT(1)	
68		=FULL(TRUE)	
69		=FORMULA.GOTO(!$A$1,TRUE)	
70		=RETURN()	
71			
72	*command*	**Template Save**	
73		=MESSAGE(1,"Save with new or old name. Apple-. to stop recalculation.")	
74		=SAVE.AS?()	
75		= IF(B74 =FALSE,HALT(),)	
76		=MESSAGE(0,)	
77		=SET.VALUE(A_Template,GET.DOCUMENT(1))	
78		=RETURN()	
79			
80	*command*	**Set Template**	
81		=SET.VALUE(A_Template,GET.DOCUMENT(1))	
82		=RETURN()	
83			
84	*command*	**Close_macro**	
85		=ACTIVATE(A_Maker_name)	
86		=CLOSE()	*Does not save changes.*
87		=RETURN()	
88			
89			
90	*command*	**auto_close**	
91		=DELETE.MENU(1,"AreaS")	
92		X SAVE.AS(,0)	*Does not save changes.*
93		=RETURN()	
94			
95	*command*	**Reset initial values**	
96			
97		=ACTIVATE(A_Maker_name)	
98		=SELECT(!Dialog boxes default values)	

	A	B	C
99		=COPY()	
100		=SELECT(!Dialog_boxes_initial_values)	
101		=PASTE.SPECIAL(3,1,FALSE,FALSE)	
102		=RETURN()	
103			
104	*command*	**Clear Area Array**	*Clears Area array and descriptions*
105		=ERROR(2,Alert2)	
106		=ACTIVATE(A_Template)	
107		=ERROR(1)	
108		=ECHO(FALSE)	
109		=SELECT(!Area_array)	
110		=FORMULA.FILL(0)	
111		=SELECT(!Area_Descriptions)	
112		=CLEAR(3)	
113		=ECHO(TRUE)	
114		=RETURN()	
115			
116		**Rectangle**	
117		=ERROR(2,Alert2)	
118		= ACTIVATE(A_Template)	
119		=ERROR(1)	
120		=FORMULA.GOTO(!Axes,TRUE)	
121		=DIALOG.BOX(Rectangle_Dialog_box)	
122		=IF(B121 =FALSE,HALT(),)	
123		=ECHO(FALSE)	
124	*Area.Rec*	=Area_Multiplier.rec*Length.rec*Height.rec	*Calculations*
125	*Phi.Rec*	=Rotate(90,0,Pitch.rec,Yaw.rec)	
126	*Theta.Rec*	=Rotate(90,0,Pitch.rec,Yaw.rec)	
127		=Area_Matrix(Area.Rec,Phi.Rec,Theta.Rec)	*Call Area_Matrix*
128		=COUNT(OFFSET(!Area_Descriptions,0,0,,1))	*Get current no. of geom.*
129	*Number.Rec*	=B128+1	*in area array.*
130	*Description.Rec*		
131	No.	=Number.Rec	
132	Geom	Rectangle	
133	L1	=Length.rec	
134	L2	=Height.rec	
135	L3		
136	Multiplier	=Area_Multiplier.rec	
137	Pitch	=Pitch.rec	
138	Yaw	=Yaw.rec	
139	Lat. Start		
140	Lat. Finish		
141	Incr.		
142	Long. Start		
143	Long. Finish		
144	Surf Area [m^2]	=Area.Rec	
145		=ACTIVATE(A_Maker_name)	*Description Paste*
146		=SELECT(Description.Rec)	
147		=COPY()	
148		=ACTIVATE(A_Template)	
149		=SELECT(OFFSET((!Area_Descriptions),Number.Rec-1,0,1,1))	
150		=PASTE.SPECIAL(3,1,FALSE,TRUE)	
151		=ECHO(TRUE)	
152		=RETURN()	
153			

	A	B	C
154		Disk	
155		=ERROR(2,Alert2)	
156		= ACTIVATE(A_Template)	
157		=ERROR(1)	
158		=FORMULA.GOTO(!Axes,TRUE)	
159		=DIALOG.BOX(Disk_Dialog_box)	
160		=IF(B159 =FALSE,HALT(),)	
161		=ECHO(FALSE)	
162	*Area.dsk*	=Area_Multiplier.dsk*PI()*Radius.dsk^2	*Calculations*
163	*Phi.dsk*	=Rotate(90,0,Pitch.dsk,Yaw.dsk)	
164	*Theta.dsk*	=Rotate(90,0,Pitch.dsk,Yaw.dsk)	
165		=Area_Matrix(Area.dsk,Phi.dsk,Theta.dsk)	*Call Area_Matrix*
166		=COUNT(OFFSET(!Area_Descriptions,0,0,,1))	*Get current no. of geom.*
167	*Number.dsk*	=B166+1	*in area array.*
168	*Description.dsk*		
169	No.	=Number.dsk	
170	Geom	Disk	
171	L1	=Radius.dsk	
172	L2		
173	L3		
174	Multiplier	=Area_Multiplier.dsk	
175	Pitch	=Pitch.dsk	
176	Yaw	=Yaw.dsk	
177	Lat. Start		
178	Lat. Finish		
179	Incr.		
180	Long. Start		
181	Long. Finish		
182	Surf Area [m^2]	=Area.dsk	
183		=ACTIVATE(A_Maker_name)	*Description Paste*
184		=SELECT(Description.dsk)	
185		=COPY()	
186		=ACTIVATE(A_Template)	
187		=SELECT(OFFSET((!Area_Descriptions),Number.dsk-1,0,1,1))	
188		=PASTE.SPECIAL(3,1,FALSE,TRUE)	
189		=ECHO(TRUE)	
190		=RETURN()	
191			
192	*Command*	Cylinder	
193		=ERROR(2,Alert2)	
194		= ACTIVATE(A_Template)	
195		=ERROR(1)	
196		=FORMULA.GOTO(!Axes,TRUE)	
197		=DIALOG.BOX(Cylinder_Dialog_Box)	
198		=IF(B197 =FALSE,HALT(),)	
199		=ECHO(FALSE)	
200	*Area.cyl*	=Area_Multiplier.cyl*PI()*radius.cyl*Length.cyl*(finish_angle.cyl-start_angle.cyl)/180	*Calculations*
201	*num.of.facets.cyl*	=(finish_angle.cyl-start_angle.cyl)/facet_angle.cyl	
202		=IF(OR(B201<>INT(B201),B201<0))	*Error check on angles*
203		= ALERT("Finish_angle must be greater than Start_angle, and the difference evenly divisible by	
204		= Cylinder()	
205		=END.IF()	
206	*Facet.Area.cyl*	=Area.cyl/num.of.facets.cyl	
207		=FOR("n",1,num.of.facets.cyl,1)	*Loop thru each facet*

	A	B	C
208		= MOD(90+start_angle.cyl+(n-0.5)*facet_angle.cyl,360)	
209	*Initial.Phi.cyl*	= IF(B208>180,360-B208,(IF(B208<-180,360+B208,B208)))	
210	*Initial.Theta.cyl*	= IF(OR(AND(B208<0,B208>-180),B208>180),-90,90)	
211	*Phi.cyl*	= Rotate(Initial.Phi.cyl,Initial.Theta.cyl,Pitch.cyl,Yaw.cyl)	
212	*Theta.cyl*	= Rotate(Initial.Phi.cyl,Initial.Theta.cyl,Pitch.cyl,Yaw.cyl)	
213		= Area_Matrix(Facet.Area.cyl,Phi.cyl,Theta.cyl)	*Call Area_Matrix*
214		=NEXT()	
215		=COUNT(OFFSET(!Area_Descriptions,0,0,,1))	*Get current no. of geom.*
216	*Number.cyl*	=B215+1	*in area array.*
217	*Description.cyl*	*Description.cyl*	
218	No.	=Number.cyl	
219	Geom	Cylinder	
220	L1	=radius.cyl	
221	L2	=Length.cyl	
222	L3		
223	Multiplier	=Area_Multiplier.cyl	
224	Pitch	=Pitch.cyl	
225	Yaw	=Yaw.cyl	
226	Lat. Start	=start_angle.cyl	
227	Lat. Finish	=finish_angle.cyl	
228	Incr.	=facet_angle.cyl	
229	Long. Start		
230	Long. Finish		
231	Surf Area [m^2]	=Area.cyl	
232		=ACTIVATE(A_Maker_name)	*Description Paste*
233		=SELECT(Description.cyl)	
234		=COPY()	
235		=ACTIVATE(A_Template)	
236		=SELECT(OFFSET((!Area_Descriptions),Number.cyl-1,0,1,1))	
237		=PASTE.SPECIAL(3,1,FALSE,TRUE)	
238		=ECHO(TRUE)	
239		=RETURN()	
240			
241	*command*	Cone	
242		=ERROR(2,Alert2)	
243		= ACTIVATE(A_Template)	
244		=ERROR(1)	
245		=FORMULA.GOTO(!Axes,TRUE)	
246		=DIALOG.BOX(Cone_dialog_box)	
247		=IF(B246 =FALSE,HALT(),)	
248		=ECHO(FALSE)	
249	*Cone.Angle.rad*	=ATAN((Radius_aft.cone-Radius_for.cone)/Length.cone)	*Calculations*
250	*Cone.Angle.deg*	=Cone.Angle.rad*PI()/180	
251		=SQRT(Length.cone^2+((Radius_aft.cone-Radius_for.cone)^2))	
252	*Area.cone*	=Area_Multiplier.cone*PI()*(Radius_aft.cone+Radius_for.cone)*B251*(Finish_angle.cone-Start_angle.cone)/180	
253	*num.of.facets.cone*	=(Finish_angle.cone-	
254		=IF(OR(B253<>INT(B253),B253<0))	*Error check on angles*
255		= ALERT("Finish_angle must be greater than Start_angle, and the difference evenly divisible by	
256		= Cone()	
257		=END.IF()	

	A	B	C
258	*Facet.Area.cone*	=Area.cone/num.of.facets.cone	
259		=FOR("n",1,num.of.facets.cone,1)	*Loop thru each facet*
260		= MOD(90+Start_angle.cone+(n-0.5)*facet_angle.cone,360)	*Cone axis is initially in*
261	*Initial.Phi.cone*	= IF(B260>180,360-B260,(IF(B260<-180,360+B260,B260)))	*neg. z direction and rotated*
262	*Initial.Theta.cone*	= IF(OR(AND(B260<0,B260>-180),B260>180),-90,90)	*to +x direction.*
263	*Phi.cone*	= Rotate(Initial.Phi.cone,Initial.Theta.cone,Pitch.cone,Yaw.cone)	
264	*Theta.cone*	= Rotate(Initial.Phi.cone,Initial.Theta.cone,Pitch.cone,Yaw.cone)	
265		= Area_Matrix(Facet.Area.cone,Phi.cone,Theta.cone)	*Call Area_Matrix*
266		=NEXT()	
267		=COUNT(OFFSET(!Area_Descriptions,0,0,,1))	*Get current no. of geom.*
268	*Number.cone*	=B267+1	*in area array.*
269	*Description.cone*	*Description.cone*	
270	No.	=Number.cone	
271	Geom	Cone	
272	L1	=Radius_for.cone	
273	L2	=Radius_aft.cone	
274	L3	=Length.cone	
275	Multiplier	=Area_Multiplier.cone	
276	Pitch	=Pitch.cone	
277	Yaw	=Yaw.cone	
278	Lat. Start	=Start_angle.cone	
279	Lat. Finish	=Finish_angle.cone	
280	Incr.	=facet_angle.cone	
281	Long. Start		
282	Long. Finish		
283	Surf Area [m^2]	=Area.cone	
284		=ACTIVATE(A_Maker_name)	*Description Paste*
285		=SELECT(Description.cone)	
286		=COPY()	
287		=ACTIVATE(A_Template)	
288		=SELECT(OFFSET((!Area_Descriptions),Number.cone-1,0,1,1))	
289		=PASTE.SPECIAL(3,1,FALSE,TRUE)	
290		=ECHO(TRUE)	
291		=RETURN()	
292			
293		**Whole_Sphere**	
294	*Area.WS*	=INPUT("Enter the area of the sphere [m^2] or a formula starting with an equal sign.",1,"Sphere"," =4*PI()*(Radius)^2",,)	
295		=IF(B294 =FALSE,HALT(),)	
296	*Radius.WS*	=SQRT(Area.WS/PI()/4)	
297	*Area_Multiplier.WS*	=INPUT("Enter an area multiplier",1,"multiplier",1,,)	
298		=IF(B297 =FALSE,HALT(),)	
299		=ECHO(FALSE)	
300		=SET.VALUE(Area.WS,Area.WS*Area_Multiplier.WS)	
301		=FORMULA.ARRAY(" =R_one_Sphere_areaS*Area.WS/4", Whole_Sphere_areaS)	
302		=ACTIVATE(A_Maker_name)	
303		=SELECT(!Whole_Sphere_areaS)	
304		=COPY()	
305		=ERROR(2,Alert2)	

	A	B	C
306		=ACTIVATE(A_Template)	
307		=ERROR(1)	
308		=SELECT(!Area_array)	
309		=PASTE.SPECIAL(3,2,FALSE,FALSE)	
310		=COUNT(OFFSET(!Area_Descriptions,0,0,,1))	*Get current no. of geom.*
311	*Number.WS*	=B310+1	*in area array.*
312	*Description.WS*	*Description.WS*	
313	No.	=Number.WS	
314	Geom	Whole Sphere	
315	L1	=Radius.WS	
316	L2		
317	L3		
318	Multiplier	=Area_Multiplier.WS	
319	Pitch		
320	Yaw		
321	Lat. Start		
322	Lat. Finish		
323	Incr.		
324	Long. Start		
325	Long. Finish		
326	Surf Area [m^2]	=Area.WS	
327		=ACTIVATE(A_Maker_name)	*Description Paste*
328		=SELECT(Description.WS)	
329		=COPY()	
330		=ACTIVATE(A_Template)	
331		=SELECT(OFFSET((!Area_Descriptions),Number.WS-1,0,1,1))	
332		=PASTE.SPECIAL(3,1,FALSE,TRUE)	
333		=ECHO(TRUE)	
334		=RETURN()	
335			
336	*command*	Sphere	
337		=ERROR(2,Alert2)	
338		= ACTIVATE(A_Template)	
339		=ERROR(1)	
340		=FORMULA.GOTO(!Axes,TRUE)	
341		=DIALOG.BOX(Sphere_Dialog_Box)	
342		=IF(B341 =FALSE,HALT(),)	
343		=ECHO(FALSE)	
344	*num.lat.facets.Sph*	=(finish_lat.sph-start_Lat.sph)/facet_angle.sph	*Calculations*
345	*num.long.facets.Sph*	=(finish_Long.sph-start_Long.sph)/facet_angle.sph	
346		=IF(OR(start_Lat.sph<-90,finish_lat.sph>90,))	*Error check on latitudes*
347		= ALERT("Latitudes must be equal or between -90 and +90 degrees.",2)	
348		= Sphere()	
349		=END.IF()	
350		=IF(OR(B345<>INT(B345),B344<>INT(B344),B345<0,B344<0))	*Error check on start*
351		= ALERT("Finish_angle must be greater than Start_angle, and the difference evenly divisible by	*and finish angles.*
352		= Sphere()	
353		=END.IF()	
354	*Total.facets.Sph*	=B344*B345	
355		=ALERT("There are "&B354&" facets. Do you wish to continue?",1)	*Too long to continue?*
356		=IF(B355 =FALSE,HALT(),"ok")	
357	*Total.Area.Sph*	=4*PI()*(Radius.sph)^2	

	A	B	C
358	*Latitude.Area.sph*	=4*PI()*(Radius.sph)^2*(COS((90- finish_lat.sph)*PI()/180)-COS((90- start_Lat.sph)*PI()/180))	
359	*Area.Sph*	=Latitude.Area.sph*(finish_Long.sph-	
360	*Facet.Area.Sph*	=Area.Sph/num.long.facets.Sph/num.lat.facets.Sph	
361		=FOR("i",1,num.lat.facets.Sph,1)	*Loop thru latitude facets.*
362		= FOR("j",1,num.long.facets.Sph,1)	*Loop thru long. facets.*
363		= (i-1)*num.long.facets.Sph+j	
364		= MESSAGE(1,"Working on "&B363&" facet of "&Total.facets.Sph)	
365	*Initial.Phi.Sph*	= start_Lat.sph+(i-0.5)*facet_angle.sph+90	
366	*Initial.Theta.Sph*	= -(start_Long.sph+(j-0.5)*facet_angle.sph)	
367	*Phi.Sph*	= Rotate(Initial.Phi.Sph,Initial.Theta.Sph,Pitch.sph,Yaw.sph	
368	*Theta.Sph*	= Rotate(Initial.Phi.Sph,Initial.Theta.Sph,Pitch.sph,Yaw.sph	
369		= Area_Matrix(Facet.Area.Sph,Phi.Sph,Theta.Sph)	*Call Area_Matrix*
370		= NEXT()	
371		=NEXT()	
372		=MESSAGE(0,)	
373		=COUNT(OFFSET(!Area_Descriptions,0,0,,1))	*Get current no. of geom.*
374	*Number.Sph*	=B373+1	*in area array.*
375	*Description.Sph*	*Description.Sph*	
376	No.	=Number.Sph	
377	Geom	Sphere	
378	L1	=Radius.sph	
379	L2		
380	L3		
381	Multiplier	=Area_Multiplier.sph	
382	Pitch	=Pitch.sph	
383	Yaw	=Yaw.sph	
384	Lat. Start	=start_Lat.sph	
385	Lat. Finish	=finish_lat.sph	
386	Incr.	=facet_angle.sph	
387	Long. Start	=start_Long.sph	
388	Long. Finish	=finish_Long.sph	
389	Surf Area [m^2]	=Area.Sph	
390		=ACTIVATE(A_Maker_name)	*Description Paste*
391		=SELECT(Description.Sph)	
392		=COPY()	
393		=ACTIVATE(A_Template)	
394		=SELECT(OFFSET((!Area_Descriptions),Number.Sph- 1,0,1,1))	
395		=PASTE.SPECIAL(3,1,FALSE,TRUE)	
396		=ECHO(TRUE)	
397		=RETURN()	
398			
399	*command*	**A Plot Manipulation**	
400		=ALERT("Compresses A_SURF/A_PLOT output Area Array from 0.25 to 0.5 km/s increments.",1)	*Use with A_SURF output.*
401		=IF(B400 =FALSE,HALT(),)	*Cut from .25 km/s and*
402		=MESSAGE(1,"Open the A_SURF/A_PLOT output file.")	*add to 0.5 km/s bins*
403		=OPEN?(,,TRUE,2,,,,)	*Start on Row with Obliquity*
404		=MESSAGE(0)	
405		=REFTEXT(INPUT("Click on the row with the obliquities (above the first line of the array.",8,"Area Array - 0.25 to 0.5 km/s increments",,,))	
406		=IF(B405 =FALSE,HALT(),)	

	A	B	C
407		=ECHO(FALSE)	
408		=SELECT(TEXTREF(B405))	
409		=SELECT("R[+1]C2:R[+1]C20")	*First row to second*
410		=COPY()	
411		=SELECT("R[+1]C2")	
412		=PASTE.SPECIAL(3,2,FALSE,FALSE)	
413		=SELECT("R[-1]")	
414		=EDIT.DELETE(2)	
415		=FOR("n",1,32)	*counter*
416		= SELECT("R[+1]")	*Insert blanks*
417		= INSERT(2)	
418		= SELECT("RC2:RC20")	*multiply by 1/2*
419		= FORMULA("0.5")	
420		= FILL.RIGHT()	
421		= SELECT("R[+1]C2:R[+1]C20")	
422		= COPY()	
423		= SELECT("R[-1]C2:R[-1]C20")	
424		= PASTE.SPECIAL(3,4,FALSE,FALSE)	
425		= COPY()	
426		= SELECT("R[-1]C2")	
427		= PASTE.SPECIAL(3,2,FALSE,FALSE)	*Add above*
428		= SELECT("R[+3]C2")	
429		= PASTE.SPECIAL(3,2,FALSE,FALSE)	*Add below*
430		= SELECT("R[-2]")	
431		= EDIT.DELETE(2)	*delete extra rows*
432		= EDIT.DELETE(2)	
433		=NEXT()	*finished*
434		=SELECT(TEXTREF(B405))	
435		=SELECT("R[1]C2:R[32]C20")	
436		=ALERT("Please select area array from 0.5 to 16 km/s and paste to PNP Template. Copy any model or range information desired.",2)	
437		=ECHO(TRUE)	
438		=RETURN()	
439			
440			
441	*Command*	**AreaS to PNP**	
442		=ACTIVATE(A_Template)	
443		=CALCULATE.DOCUMENT()	
444		=SELECT(!Eff_Area_Array)	
445		=COPY()	
446		=MESSAGE(1,"Please, open the PNP Template.")	
447		=OPEN?()	
448	*PNP.name.AtoP*	=GET.DOCUMENT(1)	
449		=ACTIVATE(PNP.name.AtoP)	Paste
450		=SELECT(!AREA_TABLE)	Effective Area Array
451		=PASTE.SPECIAL(3,1,FALSE,FALSE)	
452		=ACTIVATE(A_Template)	Copy and paste
453		=SELECT(!Areas_and_Labels)	Descriptions with headers
454		=COPY()	
455		=ACTIVATE(PNP.name.AtoP)	
456		=SELECT(OFFSET(!AREA_TABLE,0,20,1,1))	
457		=PASTE.SPECIAL(1,1,FALSE,FALSE)	
458		=FORMULA.GOTO("Geometry_Info")	Geometry information
459		=SELECT("RC[1]:R[3]C[3]")	at top of PNP Template
460		=CLEAR(3)	
461		=SELECT("R[4]C[-2]:R[15]C[2]")	
462		=CLEAR(3)	

	A	B	C
463		=SELECT("R[-4]C[2]")	
464		=FORMULA("Created by AreaS Macro")	
465		=SELECT("R[1]C")	
466		=FORMULA("Details next to Area Table")	
467		=RETURN()	

	E	F	G
6	*command*	**Alert1**	*Called by Template_open*
7	*Alert1_answer*	=ALERT("An AREA TEMPLATE needs to be opened. OK to repeat, or CANCEL to continue without template.",1)	
8		=IF(Alert1_answer =TRUE(),TEMPLATE_OPEN(),)	
9		=RETURN()	
10			
11			
12	*command*	**Alert2**	*Called by error on Activate(A_template)*
13		=ALERT("AREA TEMPLATE cannot be found. Please use OPEN-, SAVE- or SET TEMPLATE and repeat command.",3)	
14		=HALT()	
15		=RETURN()	
16			
17			
18	*function*	**Area_Matrix**	*This macro accepts an area and its normal to fill in the area matrix on Area Template*
19	*Area.AM*	=ARGUMENT("Area.AM",1,F23)	
20	*Phi.AM*	=ARGUMENT("Phi.AM",1,F24)	
21	*Theta.AM*	=ARGUMENT("Theta.AM",1,F25)	
22		Input	
23	*Area.AM*	1	*[meter^2]*
24	*Phi.AM*	90	*[degrees]*
25	*Theta.AM*	-45.00000002	*[degrees]*
26	*Phi.rad.AM*	=Phi.AM*PI()/180	*Convert to radians*
27		=IF(F26<0,ALERT("Negative angle phi from zenith in Area_Matrix Macro",2),)	
28		=IF(F26<0,HALT(),)	
29		=IF(ABS(F25)>180,ALERT("Theta > 180 in Area_Matrix Macro",2),)	
30		=IF(ABS(F25)>180,HALT(),)	
31	*Theta.rad.AM*	=Theta.AM*PI()/180	
32	*Phi.deg.AM*	=Phi.AM	
33	*Theta.deg.AM*	=Theta.AM	
34		**Parameters**	
35	*VINC.AM*	0.5	*[km/s]*
36	*AINC.AM*	5	*[deg.] A for Angle as in*
37	*Orb.Vel.AM*	8	*[km/s]*
38		= ACTIVATE(A_Template)	
39		=FOR("Velocity.AM",0.5,2*Orb.Vel.AM,VINC.AM)	*Start loop through*
40	*Velocity.AM*	= Velocity.AM	*the threat velocities*
41		= FOR("threat.quad.AM",-1,1,2)	*Loop once for Port*
42	*Threat.quad.AM*	= threat.quad.AM	*& once Starboard*
43	*Threat.Ang.rad.AM*	= threat.quad.AM*ACOS(Velocity.AM/2/Orb.Vel.AM)	
44		= IF(ABS(Theta.AM-Threat.Ang.rad.AM*180/PI())<90)	*Facing Threat?*
45	*Obliquity.rad.AM*	= ACOS(COS(Theta.rad.AM-Threat.Ang.rad.AM)*SIN(Phi.rad.AM))	
46	*Obliquity.deg.AM*	= Obliquity.rad.AM*180/PI()	
47	*Proj.Area.AM*	= Area.AM*COS(Obliquity.rad.AM)	
48	*Sum.Proj.Area.AM*	= Sum.Proj.Area.AM+F47	
49	*Vindex.AM*	= INT(Velocity.AM/VINC.AM)	*Index numbers on the*
50	*Aindex.AM*	= INT(Obliquity.deg.AM/AINC.AM)+1	*area_array template*
51	*Del.Obl.AM*	= MOD(Obliquity.deg.AM,AINC.AM)	
52	*Area.Obl.minus.AM*	= Proj.Area.AM*(AINC.AM-Del.Obl.AM)/AINC.AM/2	*Fractions at each obl.*
53	*Area.Obl.plus.AM*	= Proj.Area.AM*Del.Obl.AM/AINC.AM/2	*/2 for left & right*
54	*Array.Obl.minus.AM*	= INDEX(!Area_array,Vindex.AM,Aindex.AM)	*Starting values*
55	*Array.Obl.plus.AM*	= INDEX(!Area_array,Vindex.AM,Aindex.AM+1)	
56	*New.A.Obl.minus.AM*	= Array.Obl.minus.AM+Area.Obl.minus.AM	*Final Values*
57	*New.A.Obl.plus.AM*	= Array.Obl.plus.AM+Area.Obl.plus.AM	

	E	F	G
58		= FORMULA(New.A.Obl.minus.AM,OFFSET((!Area_array), Vindex.AM-1,Aindex.AM-1,1,1))	*array with new values.*
59		x FORMULA(New.A.Obl.minus.AM)	
60		= FORMULA(New.A.Obl.plus.AM,OFFSET((!Area_array), Vindex.AM-1,Aindex.AM,1,1))	
61		x FORMULA(New.A.Obl.minus.AM)	
62		= END.IF()	
63		= NEXT()	*next starboard threats*
64		=NEXT()	*Next threat velocity*
65		=RETURN(Sum.Proj.Area.AM)	
66			
67	*function*	**Rotate**	*Rotates initial orientaion.*
68	*Phi.Rot*	=ARGUMENT("Phi.Rot",1,F72)	*Angles in degrees.*
69	*Theta.Rot*	=ARGUMENT("Theta.Rot",1,F73)	*Note that Phi is*
70	*Pitch.Rot*	=ARGUMENT("Pitch.Rot",1,F74)	*from -z pole and Theta*
71	*Yaw.Rot*	=ARGUMENT("Yaw.Rot",1,F75)	*is from x to -y when*
72	*Phi.Rot*	90	*z points to earth.*
73	*Theta.Rot*	0	
74	*Pitch.Rot*	0	
75	*Yaw.Rot*	45	
76		=IF(AND(Pitch.Rot =0,Yaw.Rot =0))	*Bypass calculations*
77		= SET.VALUE(finalphi.rot,Phi.Rot)	*if no rotation.*
78		= SET.VALUE(finaltheta.rot,Theta.Rot)	
79		=ELSE()	
80	*y.rot*	= COS(Phi.Rot*PI()/180)	*Pitch rotation*
81	*x.rot*	= -COS(Theta.Rot*PI()/180)*SIN(Phi.Rot*PI()/180)	
82	*sign.rot*	= IF(OR(Theta.Rot<0,Theta.Rot>180),-1,1)	
83	*newx.rot*	= TRUNC(x.rot*COS(Pitch.Rot*PI()/180)+y.rot*SIN(Pitch.Rot*PI()/180),8)	
84	*newy.rot*	= TRUNC(y.rot*COS(Pitch.Rot*PI()/180)-x.rot*SIN(Pitch.Rot*PI()/180),8)	
85	*newPhi.rot*	= 180/PI()*ACOS(newy.rot)	
86	*newTheta.rot*	= IF(newPhi.rot =0,0,sign.rot*180/PI()*ACOS(-newx.rot/SIN(newPhi.rot*PI()/180)))	
87	*x2.rot*	= -SIN(Phi.Rot*PI()/180)*SIN(Theta.Rot*PI()/180)	*yaw rotation*
88	*y2.rot*	= -newx.rot	
89	*newx2.rot*	= TRUNC(x2.rot*COS(Yaw.Rot*PI()/180)+y2.rot*SIN(Yaw.Rot*PI()/180),8)	
90	*newy2.rot*	= TRUNC(y2.rot*COS(Yaw.Rot*PI()/180)-x2.rot*SIN(Yaw.Rot*PI()/180),9)	
91	*sign2.rot*	= IF(newx2.rot<0,1,-1)	
92	*finalPhi.rot*	= newPhi.rot	
93	*finalTheta.rot*	= IF(finalphi.rot =0, 0,sign2.rot*180/PI() * ACOS(newy2.rot/SIN(finalphi.rot*PI()/180)))	
94		=END.IF()	
95		=RESULT(64)	
96		=RETURN(F92:F93)	
97			
98		**Velocity_Dist**	
99	*vel.VD*	=ARGUMENT("vel.VD",1,F101)	
100	*inc.VD*	=ARGUMENT("inc.VD",1,F102)	
101	*vel.VD*	0.5	
102	*inc.VD*	28.5	
103			

	E	F	G
104	*A.VD*	2.5	
105	*B.VD*	=IF(inc.VD<60,0.5,IF(inc.VD<80,0.5-0.01*(inc.VD-60),0.3))	
106	*C.VD*	=IF(inc.VD<100,0.0125,0.0125+0.00125*(inc.VD-100))	
107	*D.VD*	=1.3-(0.01*(inc.VD-30))	
108	*E.VD*	=0.55+(0.005*(inc.VD-30))	
109	*F.VD*	=IF(inc.VD<50,0.3+(0.0008*((inc.VD-50)^2)),IF(inc.VD<80,0.3-0.01*(inc.VD-50),0))	
110	*G.VD*	=IF(inc.VD<60,18.7,IF(inc.VD<80,18.7+0.289*((inc.VD-60)^3),250))	
111	*H.VD*	=1-(0.0000757*((inc.VD-60)^2))	
112	*vo.VD*	=IF(inc.VD<60,7.25+(0.015*(inc.VD-30)),7.7)	
113	*fv1.VD*	=2*vel.VD*vo.VD-(vel.VD^2)	
114	*fv2.VD*	=G.VD*EXP(-(((vel.VD-A.VD*vo.VD)/(B.VD*vo.VD))^2))	
115	*fv3.VD*	=F.VD*EXP(-(((vel.VD-D.VD*vo.VD)/(E.VD*vo.VD))^2))	
116	*fv4.VD*	=H.VD*C.VD*(4*vel.VD*vo.VD-(vel.VD^2))	
117	*fv.VD*	=MAX(0,fv1.VD*(fv2.VD+fv3.VD)+fv4.VD)	
118		=RETURN(F117)	

	I	J	K	L	M	N	O	P	Q
1		DIALOGS							
2	Type (Text)	type	x	y	wide	high	text	init/re	*Default valu*
3									
4									
5									
6									
7									
8		Rectangle Dialog box							
9	Dialog Box		62	150	334	105	RECTANGLE		
10	OK button	1	264	71	48		OK		
11	cancel button	2	184	71	64		Cancel		
12	Text	5	8	10			Length [m]		
13		5	8	74			Area Multiplier		
14		5	9	38			Height [m]		
15		5	173	10			Pitch [±90°]		
16		5	174	39			Yaw [±180°]		
17	Edit box - nu	8	97	7	63		Length.rec	1	1
18		8	97	35	64		Height.rec	1	1
19		8	118	69	43		Area Multiplier.rec	1	1
20		8	264	7	50		Pitch.rec	0	0
21		8	265	36	50		Yaw.rec	0	0
22									
23									
24		Disk Dialog box							
25	Dialog Box		62	150	334	105	DISK		
26	OK button	1	264	71	48		OK		
27	cancel button	2	184	71	64		Cancel		
28	Text	5	6	40			Area Multiplier		
29		5	8	10			Radius [m]		
30		5	173	10			Pitch [±90°]		
31		5	174	39			Yaw [±180°]		
32	Edit box - nu	8	97	7	63		Radius.dsk	1	1
33		8	116	35	43		Area Multiplier.dsk	1	1
34		8	264	7	50		Pitch.dsk	0	0
35		8	265	36	50		Yaw.dsk	0	0
36									
37		Sphere Dialog Box							
38	Dialog Box		26	125	476	135	SPHERE		
39	OK button	1	405	101	48		OK		
40	cancel button	2	325	101	64		Cancel		
41	Text	5	7	39			Area Multiplier		
42		5	8	10			Radius [m]		
43		5	24	102			facet angle		
44		5	170	74			start Long.		
45		5	170	101			finish Long.		
46		5	171	11			start Lat.		
47		5	171	38			finish Lat.		
48		5	318	10			Pitch [±90°]		
49		5	319	39			Yaw [±180°]		
50	Edit box - nu	8	95	7	63		Radius.sph	1	1
51		8	114	98	45		facet angle.sph	10	10
52		8	115	35	43		Area Multiplier.sph	1	1

	I	J	K	L	M	N	O	P	Q
53		8	256	7	50		start Lat.sph	-90	-90
54		8	256	35	50		finish lat.sph	90	90
55		8	256	70	50		start Long.sph	0	0
56		8	256	98	50		finish Long.sph	360	360
57		8	405	.7	50		Pitch.sph	0	0
58		8	406	36	50		Yaw.sph	0	0
59									
60		Cone dialog box							
61			25	125	476	128	Cone (no closure)		
62	OK button	1	404	96	48		OK		
63	cancel button	2	324	96	64		Cancel		
64	Text	5	4	100			Area Multiplier		
65		5	8	10			Radius aft		
66		5	8	37			Radius fore		
67		5	8	66			Length [m]		
68		5	171	11			start angle		
69		5	171	38			finish angle		
70		5	171	71			facet angle		
71		5	178	100			Units: [m] & [deg]		
72		5	318	10			Pitch [±90°]		
73		5	319	39			Yaw [±180°]		
74	Edit box - nu	8	95	7	63		Radius aft.cone	1	1
75		8	95	35	63		Radius for.cone	1	1
76		8	95	64	63		Length.cone	1	1
77		8	112	96	43		Area Multiplier.cone	1	1
78		8	256	7	50		Start angle.cone	-90	-90
79		8	256	35	50		Finish angle.cone	270	270
80		8	261	67	45		facet angle.cone	5	5
81		8	405	7	50		Pitch.cone	0	0
82		8	406	36	50		Yaw.cone	0	0
83									
84		Cylinder Dialog Box							
85			25	125	476	110	CYLINDER (no closure)		
86	OK button	1	405	71	48		OK		
87	cancel button	2	325	71	64		Cancel		
88	Text	5	4	72			Area Multiplier		
89		5	8	10			Radius [m]		
90		5	8	37			Length [m]		
91		5	171	11			start angle		
92		5	171	38			finish angle		
93		5	171	71			facet angle		
94		5	318	10			Pitch [±90°]		
95		5	319	39			Yaw [±180°]		
96	Edit box - nu	8	95	7	63		radius.cyl	1	1
97		8	95	35	63		Length.cyl	1	1
98		8	112	68	43		Area Multiplier.cyl	1	1

	I	J	K	L	M	N	O	P	Q
99		8	256	7	50		start angle.cyl	-90	-90
100		8	256	35	50		finish angle.cyl	270	270
101		8	261	67	45		facet angle.cyl	5	5
102		8	405	7	50		Pitch.cyl	0	0
103		8	406	36	50		Yaw.cyl	0	0
104									
105									
106					337	392	SD Surf for EXCEL 3.0		
107	*OK Button*	1	242	342	64		OK		
108	*Text*	5	26	10			Space Debris Surfaces - AREA MAKER		
109	*Text*	5	26	33			Ver 1.1 - Feb. 14, 1992		
110	*Text*	5	26	56			------------------------------------		
111	*Text*	5	26	79			STRUCTURAL DAMAGE PREDICTION AND		
112	*Text*	5	26	102			ANALYSIS FOR HYPERVELOCITY IMPACTS		
113	*Text*	5	26	125			Contract NAS8-38856		
114	*Text*	5	26	148			------------------------------------		
115		5	26	171			NASA - Marshall Space Flight Center		
116		5	26	194			Technical Monitors:		
117		5	56	217			Greg Olsen		
118		5	56	240			Jennifer Robinson		
119		5	56	263			Joel Williamsen		
120	*Text*	5	26	290			Martin Marietta Manned Space Systems		
121	*Text*	5	26	313			Program Manager:		
122	*Text*	5	56	336			Norman Elfer		
123	*Text*	5	56	359			(504)-257-3162		

	S	T	U	V	W	X
1	MENUS					
2	Name	Command	Macro	Key	Status Bar Text	Help
3	AreaS	AreaS				
4		Clear Arrays	Clear_Area_Array		Clears Area_Array & Description_ on Area Template	
5		Rectangle	Rectangle		Adds a Rectangle to Area_Array	
6		Disk	Disk		Adds a Disk to Area_Array	
7		Cone	Cone		Adds a Cone to Area_Array	
8		Cylinder	Cylinder		Adds a Cylinder to Area_Array	
9		Sphere	Sphere		Adds a Sphere to Area_Array	
10		Whole Sphere	Whole_Sphere		Adds whole sphere to Area_Array. Faster than Sphere.	
11		-				
12		Open Template	TEMPLATE_OPEN		Opens a file to be used as the Area Template	
13		Save Template	Template_Save		Saves & identifies new name as the Area Template	
14		Set Template	Set_Template		Identifies active document as the Area Template	
15		AreaS to PNP	AreaS_to_PNP		Transfers Effective Area to PNP Template.	
16		-				
17		Open A_Plot fil	A_PLOT_Manipula		Opens A_Surf Output. Use 0.5 km/s increments.	
18		-				
19		Close AreaS	Close_macro		Closes Area Maker Macro.	

	Z	AA	AB	AC	AD	AE	AF	AG	AH	AI	AJ	AK	AL	AM	AN	AO	AP	AQ	AR	AS
4				Formula = SIN((AJ5+2.5)*PI()/180)^2-(SIN((AJ5-2.5)*PI()/180)^2)																
5	Obliquity deg	0	5	10	15	20	25	30	35	40	45	50	55	60	65	70	75	80	85	90
6	Whole Sphere areas	0.002	0.015	0.03	0.044	0.056	0.067	0.075	0.082	0.086	0.087	0.086	0.082	0.075	0.067	0.056	0.044	0.03	0.015	0.002
7	Whole Sphere	=R_	=R	=R	=R	=R	=R	=R	=R	=R	=R	=R	=R	=R	=R	=R	=R	=R	=R	=R
8				=R_one_Sphere_areaS*Area.WS/4																
9																				

	AX	AY	AZ
2			
3	VARIALBLE LISTING	REFERENCE	TYPE
4			
5	A.VD	=$F$104	0
6	AINC.AM	=$F$36	0
7	Aindex.AM	=$F$50	0
8	Alert1	=$F$6:$F$9	2
9	Alert1_answer	=$F$7	0
10	Alert2	=$F$13:$F$15	0
11	Area.AM	=$F$23	0
12	Area.cone	=$B$252	0
13	Area.cyl	=$B$200	0
14	Area.dsk	=$B$162	0
15	Area.Obl.minus.AM	=$F$52	0
16	Area.Obl.plus.AM	=$F$53	0
17	Area.Rec	=$B$124	0
18	Area.Sph	=$B$359	0
19	Area.WS	=$B$294	0
20	AreaS	=$T$3:$X$19	0
21	AreaS_to_PNP	=$B$442:$B$467	2
22	Area_Matrix	=$F$18:$F$65	1
23	Area_Multiplier.cone	=$P$77	0
24	Area_Multiplier.cyl	=$P$98	0
25	Area_Multiplier.dsk	=$P$33	0
26	Area_Multiplier.rec	=$P$19	0
27	Area_Multiplier.sph	=$P$52	0
28	Area_Multiplier.WS	=$B$297	0
29	Array.Obl.minus.AM	=$F$54	0
30	Array.Obl.plus.AM	=$F$55	0
31	Auto_Close	=$B$90:$B$93	2
32	Auto_Open	=$B$44:$B$58	2
33	A_Maker_name	=$B$45	0
34	A_PLOT_Manipulation	=$B$400:$B$438	2
35	A_Template	=$B$67	0
36	B.VD	=$F$105	0
37	C.VD	=$F$106	0
38	Clear_Area_Array	=$B$104:$B$114	2
39	Cone	=$B$241:$B$291	2
40	Cone.Angle.rad	=$B$249	0
41	Cone_dialog_box	=$J$61:$P$82	0
42	Cylinder	=$B$192:$B$239	2
43	Cylinder_Dialog_Box	=$J$85:$P$103	0
44	D.VD	=$F$107	0
45	Del.Obl.AM	=$F$51	0
46	Description.cone	=$B$270:$B$283	0
47	Description.cyl	=$B$218:$B$231	0
48	Description.dsk	=$B$169:$B$182	0
49	Description.Rec	=$B$131:$B$144	0
50	Description.Sph	=$B$376:$B$389	0
51	Description.WS	=$B$313:$B$326	0
52	Dialog_boxes_default_values	=$Q$17:$Q$103	0
53	Dialog_boxes_initial_values	=$P$17:$P$103	0

	AX	AY	AZ
54	Disk	=$B$154:$B$190	2
55	Disk_Dialog_box	=$J$25:$P$35	0
56	E.VD	=$F$108	0
57	F.VD	=$F$109	0
58	Facet.Area.cone	=$B$258	0
59	Facet.Area.cyl	=$B$206	0
60	Facet.Area.Sph	=$B$360	0
61	facet_angle.cone	=$P$80	0
62	facet_angle.cyl	=$P$101	0
63	facet_angle.sph	=$P$51	0
64	finalphi.rot	=$F$92	0
65	finaltheta.rot	=$F$93	0
66	Finish_angle.cone	=$P$79	0
67	finish_angle.cyl	=$P$100	0
68	finish_lat.sph	=$P$54	0
69	finish_Long.sph	=$P$56	0
70	fv.VD	=$F$117	0
71	fv1.VD	=$F$113	0
72	fv2.VD	=$F$114	0
73	fv3.VD	=$F$115	0
74	fv4.VD	=$F$116	0
75	G.VD	=$F$110	0
76	H.VD	=$F$111	0
77	Height.rec	=$P$18	0
78	i	=3	0
79	inc.VD	=$F$102	0
80	Initial.Phi.cone	=$B$261	0
81	Initial.Phi.cyl	=$B$209	0
82	Initial.Phi.Sph	=$B$365	0
83	Initial.Theta.cone	=$B$262	0
84	Initial.Theta.cyl	=$B$210	0
85	Initial.Theta.Sph	=$B$366	0
86	INTRO_DIALOG_BOX	=$J$106:$P$123	0
87	j	=3	0
88	Latitude.Area.sph	=$B$358	0
89	Length.cone	=$P$76	0
90	Length.cyl	=$P$97	0
91	Length.rec	=$P$17	0
92	n	=2	0
93	New.A.Obl.minus.AM	=$F$56	0
94	New.A.Obl.plus.AM	=$F$57	0
95	newPhi.rot	=$F$85	0
96	newTheta.rot	=$F$86	0
97	newx.rot	=$F$83	0
98	newx2.rot	=$F$89	0
99	newy.rot	=$F$84	0

	AX	AY	AZ
100	newy2.rot	=$F$90	0
101	num.lat.facets.Sph	=$B$344	0
102	num.long.facets.Sph	=$B$345	0
103	num.of.facets.cone	=$B$253	0
104	num.of.facets.cyl	=$B$201	0
105	Number.cone	=$B$268	0
106	Number.cyl	=$B$216	0
107	Number.dsk	=$B$167	0
108	Number.Rec	=$B$129	0
109	Number.Sph	=$B$374	0
110	Number.WS	=$B$311	0
111	Obliquity.deg.AM	=$F$46	0
112	Obliquity.rad.AM	=$F$45	0
113	open_template	=$B$63	0
114	Orb.Vel.AM	=$F$37	0
115	Orbital_Velocity_kms	=#REF!	0
116	Phi.AM	=$F$24	0
117	Phi.cone	=$B$263	0
118	Phi.cyl	=$B$211	0
119	Phi.deg.AM	=$F$32	0
120	Phi.dsk	=$B$163	0
121	Phi.rad.AM	=$F$26	0
122	Phi.Rec	=$B$125	0
123	Phi.Rot	=$F$72	0
124	Phi.Sph	=$B$367	0
125	Pitch.cone	=$P$81	0
126	Pitch.cyl	=$P$102	0
127	Pitch.dsk	=$P$34	0
128	Pitch.rec	=$P$20	0
129	Pitch.Rot	=$F$74	0
130	Pitch.sph	=$P$57	0
131	PNP.name.AtoP	=$B$448	0
132	Print_Area	=$A$1:$C$467,$E$6:$	0
133	Proj.Area.AM	=$F$47	0
134	radius.cyl	=$P$96	0
135	Radius.dsk	=$P$32	0
136	Radius.sph	=$P$50	0
137	Radius.WS	=$B$296	0
138	Radius_aft.cone	=$P$74	0
139	Radius_for.cone	=$P$75	0
140	Record1	=$B$458	2
141	Recorder	=$B$458:$B$16384	0
142	Rectangle	=$B$116:$B$152	2
143	Rectangle_Dialog_box	=$J$9:$Q$21	0
144	Reset_initial_values	=$B$95:$B$102	2
145	Rotate	=$F$67:$F$96	1
146	R_one_Sphere_areaS	=$AA$6:$AS$6	0
147	Set_Template	=$B$81:$B$82	2
148	sign.rot	=$F$82	0
149	sign2.rot	=$F$91	0
150	Sphere	=$B$337:$B$397	0
151	Sphere_Dialog_Box	=$J$38:$P$58	0
152	Start_angle.cone	=$P$78	0
153	start_angle.cyl	=$P$99	0
154	start_Lat.sph	=$P$53	0

	AX	AY	AZ
155	start_Long.sph	=$P$55	0
156	Sum.Proj.Area.AM	=$F$48	0
157	TEMPLATE_OPEN	=$B$60:$B$70	2
158	Template_Save	=$B$73:$B$78	2
159	Theta.AM	=$F$25	0
160	Theta.cone	=$B$264	0
161	Theta.cyl	=$B$212	0
162	Theta.deg.AM	=$F$33	0
163	Theta.dsk	=$B$164	0
164	Theta.rad.AM	=$F$31	0
165	Theta.Rec	=$B$126	0
166	Theta.Rot	=$F$73	0
167	Theta.Sph	=$B$368	0
168	Threat.Ang.rad.AM	=$F$43	0
169	threat.quad.AM	=3	0
170	Total.facets.Sph	=$B$354	0
171	vel.VD	=$F$101	0
172	Velocity.AM	=16.5	0
173	Velocity_Dist	=$F$99:$F$118	1
174	VINC.AM	=$F$35	0
175	Vindex.AM	=$F$49	0
176	vo.VD	=$F$112	0
177	Whole_Sphere	=$B$293:$B$334	2
178	Whole_Sphere_areaS	=$AA$7:$AS$7	0
179	x.rot	=$F$81	0
180	x2.rot	=$F$87	0
181	y.rot	=$F$80	0
182	y2.rot	=$F$88	0
183	Yaw.cone	=$P$82	0
184	Yaw.cyl	=$P$103	0
185	Yaw.dsk	=$P$35	0
186	Yaw.rec	=$P$21	0
187	Yaw.Rot	=$F$75	0
188	Yaw.sph	=$P$58	0

SD_SURF User's Manual

Appendix G. SD_FUNCTION Macro

	A	B	C
1		**Summary Information**	
2	*Title:*	Space Debris Function Macros	
3	*Contract:*	NAS8-38856	
4	*Version:*	v1.1	
5	*Programmer:*	Norman Elfer, Ph.D. (504)-257-3162	
6	*Corporation:*	Martin Marietta Manned Space Systems	
7	*Creation Date:*	Feb. 14, 1992	
8			
9	*Notice*	This series of EXCEL Macros were	
10		written in support of a contract with	
11		NASA-Marshall Space Flight Center.	
12			
13			
14			
15	*SUBROUTINES*		
16	*NAME*	INPUT	OUTPUT
17	*Meteoroid_Flux*	Diameter	*Flux [impacts/year/m^2]*
18	*Debris_Flux*	diameter, year, inclination, altitude,	*Flux [impacts/year/m^2]*
19		growth rate, flux factor, Solar radio flux	
20	*Quick_Flux*	Diameter, A Full Debris Flux Calc.)	*Flux [impacts/year/m^2]*
21	*PNP*	Flux, time [days], Area [m^2]	*PNP*
22	*Velocity_Dist*	velocity, inclination	*Flux fraction*
23			
24			
25			
26	*function*	**Meteoroid Flux**	
27		=ARGUMENT("Met_diam_cm.MFlx",1,B28)	
28	*Met_diam_cm.MFlx*	1	
29	*Meteoroid Flux*	=10^(-6.2-3.66*LOG10(Met_diam_cm.MFlx))	
30		=RETURN(B29)	*Returns Meteoroid Flux*
31			
32			
33		**Debris Flux**	
34		=ARGUMENT("diam.SDF",1,B41)	*From MSFC Memo ES44-(193-90)* *Dr. B. J. Anderson*
35		=ARGUMENT("Year.SDF",1,B42)	*Proposed Revisions to SSP30425*
36		=ARGUMENT("incl.SDF",1,B43)	*9/19/90*
37		=ARGUMENT("h.alt.SDF",1,B44)	
38		=ARGUMENT("p.growth.SDF",1,B45)	
39		=ARGUMENT("k.Flux.Factor.SDF",1,B46)	
40		=ARGUMENT("S.SDF",1,B47)	
41	*diam.SDF*	1	*Diameter (cm)*
42	*Year.SDF*	1995	*Year*
43	*incl.SDF*	28.5	*Inclination*

	A	B	C
44	h.alt.SDF	388	altitude in km ≤ 2000 km
45	p.growth.SDF	0.05	annual growth rate
46	k.Flux.Factor.SDF	4	flux factor after computer analysis
47	S.SDF	0	
48	q.SDF	=IF(Year.SDF< =2010,0.02,0.04)	As in Bumperil, year-1 in solar flux calculation
49	Solar_Flux_Year	=IF(Year.SDF<2008.6,Year.SDF-1,1996+MOD(Year.SDF-1997,11))	11 year cycle beyond 2008
50	S.calc.SDF	=IF(S.SDF>1,S.SDF,VLOOKUP(Solar_Flux_Year,SOLAR_FLUX.DAT,3))	Solar radio flux 10^4 Jy
51			Column 3 in array is Nominal
52	Xi.SDF	=VLOOKUP(Incl.SDF,Phi_table,2)	Inclination factor for 30°
53			
54	Hd.SDF	=SQRT(10^(EXP(-((LOG10(diam.SDF)-0.78)^2/0.637^2))))	
55	phi_one.SDF (h,S)	=10^(h.alt.SDF/200-S.calc.SDF/140-1.5)	
56	phi.SDF (h,S)	=phi_one.SDF/(phi_one.SDF+1)	
57			
58	F1(d)	=1.22*10^(-5)*diam.SDF^(-2.5)	
59	F2(d)	=8.1*10^10*(diam.SDF+700)^(-6)	
60			
61	g1(t)	=(1+q.SDF)^(Year.SDF-1988)	
62	g2(t)	=1+p.growth.SDF*(Year.SDF-1988)	
63			
64	Flux	=k.Flux.Factor.SDF*Hd.SDF*phi.SDF*Xi.SDF*(F_one*g_one+F_two*g_two)	
65		=RETURN(B64)	
66			
67			
68	function	QUICK_FLUX	This only recalculates the
69		=ARGUMENT("diam.QDF",1,B71)	diameter dependent portions
70		=ARGUMENT("Any_Complete_Flux_Calc.QDF",1,C76)	of the debris flux equation.
71	diam.QDF	0.70618021	Use of the second term
72	Hd.QDF	=SQRT(10^(EXP(-((LOG10(diam.QDF)-0.78)^2/0.637^2))))	from a complete flux calculation
73	F_one.QDF	=1.22*10^(-5)*diam.QDF^(-2.5)	forces a recalc if anything
74	F_two.QDF	=8.1*10^10*(diam.QDF+700)^(-6)	changes.
75	Flux.QDF	=k.Flux.Factor.SDF*Hd.SDF*phi.SDF*Xi.SDF*(F_one.QDF*g_one+F_two.QDF*g_two)	
76		=RETURN(Flux.QDF)	
77			
78			
79		PNP	
80		=ARGUMENT("Flux.PNP",1,B83)	
81		=ARGUMENT("time_days.PNP",1,B84)	
82		=ARGUMENT("Area.PNP",1,B85)	
83	Flux.PNP	2.05552E-05	
84	time_days.PNP	365	
85	Area.PNP	1	
86			
87	PNP	=EXP(-Area.PNP*time_days.PNP/365*Flux.PNP)	
88		=RETURN(B87)	
89			
90			
91			
92		Velocity_Dist	
93		=ARGUMENT("vel.VD",1,B95)	
94		=ARGUMENT("inc.VD",1,B96)	
95	vel.VD	0.25	
96	inc.VD	30	
97			
98	A.VD	2.5	
99	B.VD	=IF(inc.VD<60,0.5,IF(inc.VD<80,0.5-0.01*(inc.VD-60),0.3))	
100	C.VD	=IF(inc.VD<100,0.0125,0.0125+0.00125*(inc.VD-100))	

	A	B	C
101	*D.VD*	=1.3-(0.01*(inc.VD-30))	
102	*E.VD*	=0.55+(0.005*(inc.VD-30))	
103	*F.VD*	=IF(inc.VD<50,0.3+(0.0008*((inc.VD-50)^2)),IF(inc.VD<80,0.3-0.01*(inc.VD-50),0))	
104	*G.VD*	=IF(inc.VD<60,18.7,IF(inc.VD<80,18.7+0.289*((inc.VD-60)^3),250))	
105	*H.VD*	=1-(0.0000757*((inc.VD-60)^2))	
106	*vo.VD*	=IF(inc.VD<60,7.25+(0.015*(inc.VD-30)),7.7)	
107	*fv1.VD*	=2*vel.VD*vo.VD-(vel.VD^2)	
108	*fv2.VD*	=G.VD*EXP(-(((vel.VD-A.VD*vo.VD)/(B.VD*vo.VD))^2))	
109	*fv3.VD*	=F.VD*EXP(-(((vel.VD-D.VD*vo.VD)/(E.VD*vo.VD))^2))	
110	*fv4.VD*	=H.VD*C.VD*(4*vel.VD*vo.VD-(vel.VD^2))	
111	*fv.VD*	=MAX(0,fv1.VD*(fv2.VD+fv3.VD)+fv4.VD)	
112		=RETURN(B111)	

	E	F
1	Space Debris	
2	Phi Lookup Table	
3	Inclination °	PHI(I)
4	25	0.9
5	26	0.905
6	27	0.91
7	28	0.912
8	28.5	0.9135
9	29	0.915
10	30	0.92
11	31	0.922
12	32	0.927
13	33	0.93
14	34	0.935
15	35	0.94
16	36	0.945
17	37	0.95
18	38	0.952
19	39	0.957
20	40	0.96
21	41	0.967
22	42	0.972
23	43	0.977
24	44	0.982
25	45	0.99
26	46	0.995
27	47	1
28	48	1.005
29	49	1.01
30	50	1.02
31	51	1.025
32	52	1.03
33	53	1.04
34	54	1.045
35	55	1.05
36	56	1.06
37	57	1.065
38	58	1.075
39	59	1.08
40	60	1.09
41	61	1.1
42	62	1.115
43	63	1.13

	E	F
44	64	1.14
45	65	1.16
46	66	1.18
47	67	1.2
48	68	1.22
49	69	1.24
50	70	1.26
51	71	1.29
52	72	1.31
53	73	1.34
54	74	1.38
55	75	1.41
56	76	1.5
57	77	1.63
58	78	1.68
59	79	1.7
60	80	1.71
61	81	1.7
62	82	1.68
63	83	1.61
64	84	1.53
65	85	1.49
66	86	1.45
67	87	1.41
68	88	1.39
69	89	1.38
70	90	1.37
71	91	1.38
72	92	1.4
73	93	1.44
74	94	1.5
75	95	1.55
76	96	1.64
77	97	1.7
78	98	1.75
79	99	1.77
80	100	1.78
81	101	1.77
82	102	1.75
83	103	1.72
84	104	1.69
85	105	1.66
86	106	1.61
87	107	1.56
88	108	1.51
89	109	1.46
90	110	1.41
91	111	1.38
92	112	1.35
93	113	1.32
94	114	1.3
95	115	1.28
96	116	1.26
97	117	1.24
98	118	1.22
99	119	1.2
100	120	1.18

	E	F
101	121	1.165
102	122	1.155
103	123	1.14
104	124	1.125
105	125	1.11

	H	I	J
3	*function*	**JSC_WHIPPLE**	
4		=ARGUMENT("t_bumper.cm.WPL",1,I9)	
5		=ARGUMENT("t_rear_wall.cm.WPL",1,I10)	
6		=ARGUMENT("Spacing.cm.WPL",1,I11)	
7		=ARGUMENT("Velocity.kmps.WPL",1,I12)	
8		=ARGUMENT("Obliquity.deg.WPL",1,I13)	
9	*t_bumper.cm.WPL*	0.127	*ECHO INPUT*
10	*t_rear_wall.cm.WPL*	0.3175	
11	*Spacing.cm.WPL*	10.16	
12	*Velocity.kmps.WPL*	6	
13	*Obliquity.deg.WPL*	0	
14	*Density_Proj.WPL [g/cc].*	2.7	*MATERIAL PROPERTIES*
15	*Density_RW.WPL [g/cc].*	2.7	
16	*Density_BUMP.WPL [g/cc].*	2.7	
17	*Material_Strength.ksi.WPL [ksi]*	55	
18	*V_normal.WPL*	=Velocity.kmps.WPL*COS(Obliquity.deg.WPL*PI()/180)	CALCULATED VALUES
19	*Crit. diam [cm]*	=3.918*t_rear_wall.cm.WPL^(2/3)*Density_Proj.WPL^(-1/3)*Density_BUMP.WPL^(-1/9)*V_normal.WPL^(-2/3)*Spacing.cm.WPL^(1/3)*(Material_Strength.ksi.WPL/70)^(1/3)	V ≥ 7
20		=(((t_rear_wall.cm.WPL*(Material_Strength.ksi.WPL/40)^0.5+t_bumper.cm.WPL)/(1.248*Density_Proj.WPL^0.5*COS(Obliquity.deg.WPL*PI()/180)))^(18/19)*(1.75-V_normal.WPL/4))	3< V < 7
21		=((1.071*t_rear_wall.cm.WPL^(2/3)*Density_Proj.WPL^(-1/3)*Density_BUMP.WPL^(-1/9)*Spacing.cm.WPL^(1/3)*(Material_Strength.ksi.WPL/70)^(1/3))*(V_normal.WPL/4-0.75))	
22	*Crit. diam [cm]*	=I21+I20	
23	*Crit. diam [cm]*	=((t_rear_wall.cm.WPL*(Material_Strength.ksi.WPL/40)^0.5+t_bumper.cm.WPL)/(0.6*(COS(Obliquity.deg.WPL*PI()/180))^(5/3)*Density_Proj.WPL^0.5*Velocity.kmps.WPL^(2/3)))^(18/19)	V ≤ 3
24	*Critical Diam. [cm]*	=IF(V_normal.WPL< =3,I23,IF(V_normal.WPL<7,I22,I19))	
25		=RETURN(I24)	
26			
27	*Function*	**MULTI-SHOCK**	
28		=ARGUMENT("Areal_Dens_Bump.g_per_sqcm.MS",1,I33)	
29		=ARGUMENT("t_rear_wall.cm.MS",1,I34)	
30		=ARGUMENT("Spacing.cm.MS",1,I35)	
31		=ARGUMENT("Velocity.MS",1,I36)	
32		=ARGUMENT("Obliquity.deg.MS",1,I37)	
33	*Areal_Dens_Bump.gpsqcm.MS*	1.3716	*Echo Input*
34	*t_rear_wall.cm.MS*	0.125	
35	*Spacing.cm.MS*	12	
36	*Velocity.MS*	6	
37	*Obliquity.deg.MS*	0	
38	*Density_Proj.MS [g/cc].*	2.7	
39	*Density_RW.MS [g/cc].*	2.7	
40	*Material_Strength.ksi.MS [ksi]*	55	
41	*V_normal.MS*	=Velocity.MS*COS(Obliquity.deg.MS*PI()/180)	*CALCULATED VALUES*
42	*Crit. diam [cm]*	=0.354*I34^(1/3)*I38^(-1/3)*I39^(1/3)*I41^(-1/3)*I35^(2/3)*(I40/40)^(1/6)	*V ≥ 7*
43		=(((I34*(I40/40)^0.5+0.37*I33)/(0.624*I38^0.5*COS(I37*PI()/180)))^(18/19)*(2-I41/3))	*3< V < 7*
44		=((0.1948*I34^(1/3)*I38^(-1/3)*I39^(1/3)*I35^(2/3)*(I40/40)^(1/6))*(I41/3-1))	
45	*Crit. diam [cm]*	=I44+I43	

	H	I	J
46	*Crit. diam [cm]*	=((I34*(I40/40)^0.5+0.37*I33)/(0.3*(COS(I37*PI()/180))^(5/3)*I38^0.5*I36^(2/3)))^(18/19)	*V ≤ 3*
47	*Critical Diam. [cm]*	=IF(I41< =3,I46,IF(I41<6,I45,I42))	
48		=RETURN(I47)	

	L	M	N	O	P
1	The solar flux data was taken from				
2	BUMPERII SOLAR_FLUX.DAT				
3	! This data comes from the March 1990 change req				
4	! to SSP 30425 table 4-1. The copy used was mis				
5	! the page that covered 7/95-12/97. Most of the				
6	! missing data was made up by using the values list				
7	! for eleven years later. 9//97-12/97 was estimated				
8	! by interpolation.				
9	Year	Max	Nomina	Min	
10	1993.000	178.6	121.5	87.8	
11	1993.083	176.3	120.5	86.5	
12	1993.167	174.9	119.5	85.9	
13	1993.250	171.1	117.9	85	
14	1993.333	164.5	116.3	83.6	
15	1993.417	158.1	114.6	82.3	
16	1993.500	154.4	112.9	81.6	
17	1993.583	152.7	111.1	81.5	
18	1993.667	150.8	109.5	81.9	
19	1993.750	148.1	108	81.6	
20	1993.833	145	106.4	81.4	
21	1993.917	141.1	104.9	80.2	
22	1994.000	137	103.4	80.3	
23	1994.083	132.4	101.9	80	
24	1994.167	125.4	100.3	78.9	
25	1994.250	119.5	98.9	77.6	
26	1994.333	118.4	97.7	76.6	
27	1994.417	118.7	96.6	74.8	
28	1994.500	119.4	95.6	74	
29	1994.583	119.8	94.8	73.4	
30	1994.667	119	93.9	73.2	
31	1994.750	117.7	92.8	73.1	
32	1994.833	116.4	91.7	72.7	
33	1994.917	114.6	90.6	71.7	
34	1995.000	110.8	89.6	71.1	
35	1995.083	105.4	88.4	70.6	
36	1995.167	103.2	87.3	70.1	
37	1995.250	102	86.5	69.9	
38	1995.333	100.4	85.7	70	
39	1995.417	98.2	84.8	69.9	
40	1995.500	96.6	83.6	69.7	
41	1995.583	94.6	82.5	69.5	
42	1995.667	93.8	81.8	69.4	
43	1995.750	92.7	81.1	69.3	

	L	M	N	O	P
44	1995.833	92	80.3	69	
45	1995.917	91.8	79.6	68.8	
46	1996.000	91.4	78.9	68.5	
47	1996.083	90.8	78.2	68.2	
48	1996.167	90.1	77.5	68.2	
49	1996.250	89.1	76.9	68.2	
50	1996.333	88.2	76.4	68.2	
51	1996.417	87	75.9	68.3	
52	1996.500	85.4	75.3	68.3	
53	1996.583	83.2	74.8	68.3	
54	1996.667	80.5	74.2	68.3	
55	1996.750	78.5	73.5	67.9	
56	1996.833	77.6	72.9	67.6	
57	1996.917	77.1	72.3	67.4	
58	1997.000	76.9	72	67.4	
59	1997.083	76.7	71.6	67.2	
60	1997.167	76.5	71.3	67.1	
61	1997.250	76.2	70.9	67	
62	1997.333	75.2	70.6	67	
63	1997.417	74.2	70.3	67	
64	1997.500	74	70.1	67	
65	1997.583	73.5	69.9	67	
66	1997.667	73.8	70.1	67	
67	1997.750	74.1	70.2	67	
68	1997.833	74.3	70.4	67	
69	1997.917	74.6	70.5	67	
70	1998.000	74.9	70.7	67	
71	1998.083	76.2	71.1	67.1	
72	1998.167	78.4	71.6	67.2	
73	1998.250	79.8	72.2	67.3	
74	1998.333	81.5	72.8	67.4	
75	1998.417	84.1	73.6	67.5	
76	1998.500	87.7	74.5	67.7	
77	1998.583	93.4	75.7	67.9	
78	1998.667	97.9	77	68	
79	1998.750	101.7	78.4	68	
80	1998.833	107.7	80.1	68	
81	1998.917	114.5	82	68	
82	1999.000	121.1	84	68.1	
83	1999.083	129.1	86.2	68.4	
84	1999.167	137.6	88.5	68.5	
85	1999.250	143.4	91	68.6	
86	1999.333	147.6	93.7	68.8	
87	1999.417	151.7	96.3	68.7	
88	1999.500	155.7	98.9	68.8	
89	1999.583	160.1	101.6	69.2	
90	1999.667	164.8	104.4	69.7	
91	1999.750	169.1	107.2	70.1	
92	1999.833	173	110.2	70.6	
93	1999.917	177.1	113.2	70.7	
94	2000.000	186.1	116.2	71.3	
95	2000.083	191.5	119.3	72.2	
96	2000.167	194.3	122	72.6	
97	2000.250	196.9	124.3	73.3	
98	2000.333	199.6	126.5	73.9	
99	2000.417	204.2	128.6	74.1	
100	2000.500	210.6	131	74.4	

	L	M	N	O	P
101	2000.583	214.8	133.3	74.5	
102	2000.667	217.2	135.6	74.6	
103	2000.750	221.6	137.6	74.5	
104	2000.833	226.9	139.6	74.1	
105	2000.917	229.9	141.4	73.6	
106	2001.000	231.7	143.2	73.5	
107	2001.083	233.7	144.6	73.6	
108	2001.167	235.6	145.6	74	
109	2001.250	238.8	146.7	75.1	
110	2001.333	242.8	147.2	75.8	
111	2001.417	245.2	147.7	76.5	
112	2001.500	224.5	148.1	78.1	
113	2001.583	243.3	148.4	80.1	
114	2001.667	244.7	148.7	82.5	
115	2001.750	245.7	148.2	84	
116	2001.833	243.3	146.8	85.5	
117	2001.917	239.4	145.7	87.9	
118	2002.000	235	145.1	89.5	
119	2002.083	232.9	144.9	92.2	
120	2002.167	233.3	144.9	93.8	
121	2002.250	233.1	144.7	94.9	
122	2002.333	231.2	144.2	95	
123	2002.417	229.1	143.5	94.7	
124	2002.500	228.1	142.7	94.9	
125	2002.583	227.6	142.3	96.5	
126	2002.667	226.7	142.1	97.3	
127	2002.750	225.6	141.3	96.8	
128	2002.833	223	140.1	96	
129	2002.917	218.6	138.4	96	
130	2003.000	215.2	136.8	96.6	
131	2003.083	212	135.5	96.7	
132	2003.167	206.9	134.3	95.1	
133	2003.250	204	133	95	
134	2003.333	203.6	131.6	96.3	
135	2003.417	200.4	129.8	96.5	
136	2003.500	196.8	128.3	94.7	
137	2003.583	195.7	127.3	93.6	
138	2003.667	194.8	126.5	93.5	
139	2003.750	191.5	125.1	91.9	
140	2003.833	187.4	123.5	88.7	
141	2003.917	182.9	122.3	86.6	
142	2004.000	178.6	121.5	87.8	
143	2004.083	176.3	120.5	86.5	
144	2004.167	174.9	119.5	85.9	
145	2004.250	171.1	117.9	85	
146	2004.333	164.5	116.3	83.6	
147	2004.417	158.1	114.6	82.3	
148	2004.500	154.4	112.9	81.6	
149	2004.583	152.7	111.1	81.5	
150	2004.667	150.8	109.5	81.9	
151	2004.750	148.1	108	81.6	
152	2004.833	145	106.4	81.4	
153	2004.917	141.1	104.9	80.2	
154	2005.000	137	103.4	80.3	
155	2005.083	132.4	101.9	80	
156	2005.167	125.4	100.3	78.9	
157	2005.250	119.5	98.9	77.6	
158	2005.333	118.4	97.7	76.6	
159	2005.417	118.7	96.6	74.8	
160	2005.500	119.4	95.6	74	
161	2005.583	119.8	94.8	73.4	
162	2005.667	119	93.9	73.2	

	L	M	N	O	P
163	2005.750	117.7	92.8	73.1	
164	2005.833	116.4	91.7	72.7	
165	2005.917	114.6	90.6	71.7	
166	2006.000	110.8	89.6	71.1	
167	2006.083	105.4	88.4	70.6	
168	2006.167	103.2	87.3	70.1	
169	2006.250	102	86.5	69.9	
170	2006.333	100.4	85.7	70	
171	2006.417	98.2	84.8	69.9	
172	2006.500	96.6	83.6	69.7	
173	2006.583	94.6	82.5	69.5	
174	2006.667	93.8	81.8	69.4	
175	2006.750	92.7	81.1	69.3	
176	2006.833	92	80.3	69	
177	2006.917	91.8	79.6	68.8	
178	2007.000	91.4	78.9	68.5	
179	2007.083	90.8	78.2	68.2	
180	2007.167	90.1	77.5	68.2	
181	2007.250	89.1	76.9	68.2	
182	2007.333	88.2	76.4	68.2	
183	2007.417	87	75.9	68.3	
184	2007.500	85.4	75.3	68.3	
185	2007.583	83.2	74.8	68.3	
186	2007.667	80.5	74.2	68.3	
187	2007.750	78.5	73.5	67.9	
188	2007.833	77.6	72.9	67.6	
189	2007.917	77.1	72.3	67.4	
190	2008.000	76.9	72	67.4	
191	2008.083	76.7	71.6	67.2	
192	2008.167	76.5	71.3	67.1	
193	2008.250	76.2	70.9	67	
194	2008.333	75.2	70.6	67	
195	2008.417	74.2	70.3	67	
196	2008.500	74	70.1	67	
197	2008.583	73.5	69.9	67	

	S	T	U
2	VARIABLE	REFERENCE	TYPE
3			
4	A.VD	=$B$98	0
5	Any_Complete_Flux_Calc.QDF	=$C$76	0
6	Area.PNP	=$B$85	0
7	Areal_Dens_Bump.gpsqcm.MS	=$I$33	0
8	Areal_Dens_Bump.g_per_sqcm.MS	=$I$33	0
9	B.VD	=$B$99	0
10	C.VD	=$B$100	0
11	D.VD	=$B$101	0
12	Debris_Flux	=$B$34:$B$65	1
13	Density_BUMP.WPL	=$I$16	0
14	Density_Proj.MS	=$I$38	0
15	Density_Proj.WPL	=$I$14	0
16	Density_RW.MS	=$I$39	0
17	Density_RW.WPL	=$I$15	0
18	diam.QDF	=$B$71	0
19	diam.SDF	=$B$41	0
20	E.VD	=$B$102	0
21	F.VD	=$B$103	0
22	Flux.PNP	=$B$83	0
23	Flux.QDF	=$B$75	0
24	fv1.VD	=$B$107	0
25	fv2.VD	=$B$108	0
26	fv3.VD	=$B$109	0
27	fv4.VD	=$B$110	0
28	F_one	=$B$58	0
29	F_one.QDF	=$B$73	0
30	F_two	=$B$59	0
31	F_two.QDF	=$B$74	0
32	G.VD	=$B$104	0
33	g_one	=$B$61	0
34	g_two	=$B$62	0
35	h.alt.SDF	=$B$44	0
36	H.VD	=$B$105	0
37	Hd.QDF	=$B$72	0
38	Hd.SDF	=$B$54	0
39	inc.VD	=$B$96	0
40	incl.SDF	=$B$43	0
41	JSC_WHIPPLE	=$I$4:$I$25	1
42	k.Flux.Factor.SDF	=$B$46	0
43	Material_Strength.ksi.MS	=$I$40	0
44	Material_Strength.ksi.WPL	=$I$17	0

	S	T	U
45	Met_diam_cm.MFlx	=$B$28	0
46	MFlux_Dcm	=$B$26:$B$30	1
47	MULTI_SHOCK	=$I$28:$I$48	1
48	Obliquity.deg.MS	=$I$37	0
49	Obliquity.deg.WPL	=$I$13	0
50	p.growth.SDF	=$B$45	0
51	phi.SDF	=$B$56	0
52	phi_one.SDF	=$B$55	0
53	Phi_table	=$E$4:$F$105	0
54	PNP_FDaysArea	=$B$79:$B$88	1
55	Print_Area	=$A$1:$C$112,$E$1:	0
56	q.SDF	=$B$48	0
57	QUICK_FLUX	=$B$69:$B$76	1
58	S.calc.SDF	=$B$50	0
59	S.SDF	=$B$47	0
60	SOLAR_FLUX.DAT	=$L$10:$O$197	0
61	Solar_Flux_Year	=$B$49	0
62	Spacing.cm.MS	=$I$35	0
63	Spacing.cm.WPL	=$I$11	0
64	time_days.PNP	=$B$84	0
65	t_bumper.cm.WPL	=$I$9	0
66	t_rear_wall.cm.MS	=$I$34	0
67	t_rear_wall.cm.WPL	=$I$10	0
68	vel.VD	=$B$95	0
69	Velocity.kmps.WPL	=$I$12	0
70	Velocity.MS	=$I$36	0
71	Velocity_Dist	=$B$92:$B$112	1
72	vo.VD	=$B$106	0
73	V_normal.MS	=$I$41	0
74	V_normal.WPL	=$I$18	0
75	Xi.SDF	=$B$52	0
76	Year.SDF	=$B$42	0

SD_SURF User's Manual

Appendix H. Typical EXCEL Output

PNP	SHIELD INFO	GEOMETRY INFO	ENVIRONMENT INFO	SHIELD CALCULATION
0.999975	PID NUMBER 2	Model: Created by AreaS Macro	year 1995	RESPONSE OUTPUT FILES
		Element: Details next to Area Table	inclination 28.5°	
	REGRESSION PENETRATION FUNCTION	Range:	altitude 388	35pids.rs
		Σ Area [m^2]:	mass growth rate 5%	
1-PNP	Configuration Shield Wall	0.3685	flux factor 4	
2.50E-05	Double Plate 6061-T6 2219-T87		Solar Flux (autocalc if 0) 0	
	Shield Thickness = 0.0500		Exposure Time [yr] 1	
	Vessel Wall Thickness = 0.1250			FLUX CALCULATION
	Standoff = 4.0000			diameter [cm] 1
	With 30 Layers of MLI against vessel wall			
				F [impacts/m^2/yr 2E-05

File: PNP_Template Date: 2/17/92

Ballistic Limit Surface [cm]

km/s \ °	0°	5°	10°	15°	20°	25°	30°	35°	40°	45°	50°	55°	60°	65°	70°	75°	80°	85°	90°
0.5	1.33	1.33	1.34	1.37	1.40	1.44	1.50	1.58	1.67	1.79	1.94	2.15	2.42	2.42	2.42	2.42	2.42	2.42	2.42
1.0	0.78	0.79	0.79	0.81	0.83	0.85	0.89	0.93	0.99	1.06	1.15	1.27	1.43	1.43	1.43	1.43	1.43	1.43	1.43
1.5	0.52	0.52	0.52	0.53	0.53	0.54	0.55	0.57	0.58	0.60	0.65	0.72	0.81	0.81	0.81	0.81	0.81	0.81	0.81
2.0	0.42	0.43	0.44	0.44	0.45	0.46	0.49	0.53	0.54	0.56	0.59	0.62	0.67	0.67	0.67	0.67	0.67	0.67	0.67
2.5	0.44	0.45	0.46	0.47	0.47	0.48	0.48	0.49	0.52	0.54	0.56	0.59	0.62	0.62	0.62	0.62	0.62	0.62	0.62
3.0	0.47	0.48	0.49	0.49	0.50	0.50	0.51	0.51	0.51	0.52	0.54	0.57	0.60	0.60	0.60	0.60	0.60	0.60	0.60
3.5	0.50	0.51	0.51	0.52	0.52	0.53	0.53	0.53	0.53	0.54	0.55	0.57	0.60	0.60	0.60	0.60	0.60	0.60	0.60
4.0	0.52	0.53	0.54	0.55	0.55	0.55	0.55	0.55	0.55	0.55	0.56	0.58	0.61	0.61	0.61	0.61	0.61	0.61	0.61
4.5	0.55	0.57	0.57	0.58	0.58	0.58	0.57	0.57	0.57	0.57	0.57	0.59	0.61	0.61	0.61	0.61	0.61	0.61	0.61
5.0	0.59	0.60	0.60	0.61	0.61	0.60	0.60	0.59	0.59	0.58	0.59	0.60	0.62	0.62	0.62	0.62	0.62	0.62	0.62
5.5	0.62	0.63	0.64	0.64	0.64	0.63	0.62	0.61	0.61	0.60	0.60	0.61	0.63	0.63	0.63	0.63	0.63	0.63	0.63
6.0	0.65	0.67	0.67	0.67	0.67	0.66	0.65	0.64	0.63	0.62	0.61	0.62	0.64	0.64	0.64	0.64	0.64	0.64	0.64
6.5	0.66	0.67	0.67	0.67	0.67	0.68	0.68	0.66	0.65	0.64	0.63	0.63	0.65	0.65	0.65	0.65	0.65	0.65	0.65
7.0	0.65	0.65	0.66	0.66	0.66	0.67	0.68	0.69	0.67	0.65	0.64	0.64	0.66	0.66	0.66	0.66	0.66	0.66	0.66
7.5	0.64	0.64	0.65	0.65	0.65	0.66	0.67	0.68	0.69	0.67	0.66	0.65	0.67	0.67	0.67	0.67	0.67	0.67	0.67
8.0	0.63	0.63	0.64	0.64	0.64	0.65	0.66	0.67	0.68	0.69	0.67	0.67	0.68	0.68	0.68	0.68	0.68	0.68	0.68
8.5	0.62	0.63	0.63	0.63	0.63	0.64	0.65	0.66	0.67	0.68	0.69	0.68	0.68	0.68	0.68	0.68	0.68	0.68	0.68
9.0	0.62	0.62	0.62	0.62	0.63	0.63	0.64	0.65	0.66	0.67	0.69	0.69	0.69	0.69	0.69	0.69	0.69	0.69	0.69
9.5	0.61	0.61	0.61	0.62	0.62	0.63	0.63	0.64	0.65	0.67	0.68	0.70	0.70	0.70	0.70	0.70	0.70	0.70	0.70
10.0	0.60	0.60	0.61	0.61	0.61	0.62	0.63	0.63	0.65	0.66	0.67	0.69	0.71	0.71	0.71	0.71	0.71	0.71	0.71
10.5	0.60	0.60	0.60	0.60	0.61	0.61	0.62	0.63	0.64	0.65	0.67	0.69	0.71	0.71	0.71	0.71	0.71	0.71	0.71
11.0	0.59	0.59	0.59	0.59	0.60	0.60	0.61	0.62	0.63	0.64	0.66	0.68	0.70	0.70	0.70	0.70	0.70	0.70	0.70
11.5	0.58	0.58	0.59	0.59	0.59	0.60	0.60	0.61	0.62	0.64	0.65	0.67	0.69	0.69	0.69	0.69	0.69	0.69	0.69
12.0	0.58	0.58	0.58	0.58	0.59	0.59	0.60	0.61	0.62	0.63	0.64	0.66	0.69	0.69	0.69	0.69	0.69	0.69	0.69
12.5	0.57	0.57	0.57	0.58	0.58	0.59	0.59	0.60	0.61	0.62	0.64	0.66	0.68	0.68	0.68	0.68	0.68	0.68	0.68
13.0	0.57	0.57	0.57	0.57	0.57	0.58	0.59	0.59	0.60	0.62	0.63	0.65	0.67	0.67	0.67	0.67	0.67	0.67	0.67
13.5	0.56	0.56	0.56	0.57	0.57	0.57	0.58	0.59	0.60	0.61	0.63	0.64	0.67	0.67	0.67	0.67	0.67	0.67	0.67
14.0	0.56	0.56	0.56	0.56	0.56	0.57	0.58	0.58	0.59	0.61	0.62	0.64	0.66	0.66	0.66	0.66	0.66	0.66	0.66
14.5	0.55	0.55	0.55	0.55	0.56	0.56	0.57	0.58	0.59	0.60	0.61	0.63	0.65	0.65	0.65	0.65	0.65	0.65	0.65
15.0	0.55	0.55	0.55	0.55	0.55	0.56	0.57	0.57	0.58	0.59	0.61	0.63	0.65	0.65	0.65	0.65	0.65	0.65	0.65
15.5	0.54	0.54	0.54	0.55	0.55	0.55	0.56	0.57	0.58	0.59	0.60	0.62	0.64	0.64	0.64	0.64	0.64	0.64	0.64
16.0	0.54	0.54	0.54	0.54	0.55	0.55	0.56	0.56	0.57	0.59	0.60	0.62	0.64	0.64	0.64	0.64	0.64	0.64	0.64

File: PNP_Template Date: 2/17/92

FLUX [impacts per sq meter per year

km/s\deg	0°	5°	10°	15°	20°	25°	30°	35°	40°	45°	50°	55°	60°	65°	70°	75°	80°	85°	90°
0.5	1.24E-5	1.23E-5	1.2E-5	1.16E-5	1.1E-5	1.03E-5	9.47E-6	8.55E-6	7.59E-6	6.61E-6	5.65E-6	4.72E-6	3.88E-6	3.88E-6	3.88E-6	3.88E-6	3.88E-6	3.88E-6	3.88E-6
1.0	4.23E-5	4.19E-5	4.09E-5	3.93E-5	3.72E-5	3.43E-5	3.13E-5	2.78E-5	2.43E-5	2.07E-5	1.7E-5	1.36E-5	1.05E-5	1.05E-5	1.05E-5	1.05E-5	1.05E-5	1.05E-5	1.05E-5
1.5	1.15E-4	1.14E-4	1.13E-4	1.11E-4	1.08E-4	1.04E-4	9.9E-5	9.37E-5	8.72E-5	8.E-5	6.61E-5	5.19E-5	3.87E-5	3.87E-5	3.87E-5	3.87E-5	3.87E-5	3.87E-5	3.87E-5
2.0	1.94E-4	1.87E-4	1.78E-4	1.71E-4	1.64E-4	1.56E-4	1.32E-4	1.1E-4	1.03E-4	9.47E-5	8.55E-5	7.57E-5	6.25E-5	6.25E-5	6.25E-5	6.25E-5	6.25E-5	6.25E-5	6.25E-5
2.5	1.72E-4	1.63E-4	1.55E-4	1.49E-4	1.45E-4	1.41E-4	1.37E-4	1.33E-4	1.15E-4	1.06E-4	9.64E-5	8.55E-5	7.36E-5	7.36E-5	7.36E-5	7.36E-5	7.36E-5	7.36E-5	7.36E-5
3.0	1.49E-4	1.42E-4	1.35E-4	1.31E-4	1.28E-4	1.26E-4	1.23E-4	1.21E-4	1.18E-4	1.14E-4	1.05E-4	9.35E-5	8.06E-5	8.06E-5	8.06E-5	8.06E-5	8.06E-5	8.06E-5	8.06E-5
3.5	1.3E-4	1.23E-4	1.18E-4	1.15E-4	1.13E-4	1.12E-4	1.11E-4	1.1E-4	1.09E-4	1.06E-4	1.01E-4	9.31E-5	8.18E-5	8.18E-5	8.18E-5	8.18E-5	8.18E-5	8.18E-5	8.18E-5
4.0	1.13E-4	1.07E-4	1.03E-4	1.01E-4	1.E-4	9.99E-5	1.E-4	1.01E-4	1.E-4	9.89E-5	9.54E-5	8.9E-5	7.9E-5	7.9E-5	7.9E-5	7.9E-5	7.9E-5	7.9E-5	7.9E-5
4.5	9.83E-5	9.36E-5	9.07E-5	8.9E-5	8.87E-5	8.93E-5	9.04E-5	9.17E-5	9.25E-5	9.24E-5	9.02E-5	8.52E-5	7.65E-5	7.65E-5	7.65E-5	7.65E-5	7.65E-5	7.65E-5	7.65E-5
5.0	8.59E-5	8.17E-5	7.94E-5	7.84E-5	7.87E-5	7.99E-5	8.17E-5	8.37E-5	8.55E-5	8.62E-5	8.52E-5	8.14E-5	7.39E-5	7.39E-5	7.39E-5	7.39E-5	7.39E-5	7.39E-5	7.39E-5
5.5	7.5E-5	7.15E-5	6.96E-5	6.92E-5	6.98E-5	7.15E-5	7.37E-5	7.64E-5	7.89E-5	8.05E-5	8.05E-5	7.79E-5	7.14E-5	7.14E-5	7.14E-5	7.14E-5	7.14E-5	7.14E-5	7.14E-5
6.0	6.56E-5	6.26E-5	6.12E-5	6.11E-5	6.22E-5	6.41E-5	6.67E-5	6.98E-5	7.28E-5	7.53E-5	7.63E-5	7.45E-5	6.91E-5	6.91E-5	6.91E-5	6.91E-5	6.91E-5	6.91E-5	6.91E-5
6.5	6.3E-5	6.29E-5	6.24E-5	6.17E-5	6.07E-5	5.93E-5	6.04E-5	6.38E-5	6.74E-5	7.04E-5	7.21E-5	7.13E-5	6.68E-5	6.68E-5	6.68E-5	6.68E-5	6.68E-5	6.68E-5	6.68E-5
7.0	6.58E-5	6.56E-5	6.52E-5	6.44E-5	6.33E-5	6.19E-5	6.02E-5	5.84E-5	6.23E-5	6.58E-5	6.82E-5	6.82E-5	6.46E-5	6.46E-5	6.46E-5	6.46E-5	6.46E-5	6.46E-5	6.46E-5
7.5	6.84E-5	6.82E-5	6.78E-5	6.7E-5	6.58E-5	6.44E-5	6.26E-5	6.05E-5	5.81E-5	6.16E-5	6.45E-5	6.53E-5	6.25E-5	6.25E-5	6.25E-5	6.25E-5	6.25E-5	6.25E-5	6.25E-5
8.0	7.09E-5	7.07E-5	7.02E-5	6.94E-5	6.83E-5	6.68E-5	6.49E-5	6.28E-5	6.02E-5	5.77E-5	6.11E-5	6.25E-5	6.04E-5	6.04E-5	6.04E-5	6.04E-5	6.04E-5	6.04E-5	6.04E-5
8.5	7.33E-5	7.31E-5	7.26E-5	7.17E-5	7.05E-5	6.9E-5	6.71E-5	6.49E-5	6.23E-5	5.93E-5	5.78E-5	5.98E-5	5.85E-5	5.85E-5	5.85E-5	5.85E-5	5.85E-5	5.85E-5	5.85E-5
9.0	7.55E-5	7.54E-5	7.48E-5	7.4E-5	7.27E-5	7.11E-5	6.92E-5	6.69E-5	6.42E-5	6.11E-5	5.77E-5	5.73E-5	5.66E-5	5.66E-5	5.66E-5	5.66E-5	5.66E-5	5.66E-5	5.66E-5
9.5	7.77E-5	7.75E-5	7.69E-5	7.6E-5	7.48E-5	7.31E-5	7.11E-5	6.87E-5	6.6E-5	6.28E-5	5.93E-5	5.53E-5	5.48E-5	5.48E-5	5.48E-5	5.48E-5	5.48E-5	5.48E-5	5.48E-5
10.0	7.97E-5	7.95E-5	7.89E-5	7.8E-5	7.67E-5	7.5E-5	7.3E-5	7.05E-5	6.77E-5	6.44E-5	6.08E-5	5.67E-5	5.3E-5	5.3E-5	5.3E-5	5.3E-5	5.3E-5	5.3E-5	5.3E-5
10.5	8.21E-5	8.19E-5	8.13E-5	8.04E-5	7.9E-5	7.73E-5	7.52E-5	7.26E-5	6.97E-5	6.64E-5	6.26E-5	5.84E-5	5.37E-5	5.37E-5	5.37E-5	5.37E-5	5.37E-5	5.37E-5	5.37E-5
11.0	8.45E-5	8.43E-5	8.37E-5	8.27E-5	8.13E-5	7.95E-5	7.73E-5	7.47E-5	7.17E-5	6.83E-5	6.44E-5	6.01E-5	5.53E-5	5.53E-5	5.53E-5	5.53E-5	5.53E-5	5.53E-5	5.53E-5
11.5	8.68E-5	8.66E-5	8.6E-5	8.5E-5	8.36E-5	8.17E-5	7.95E-5	7.68E-5	7.37E-5	7.02E-5	6.62E-5	6.18E-5	5.68E-5	5.68E-5	5.68E-5	5.68E-5	5.68E-5	5.68E-5	5.68E-5
12.0	8.91E-5	8.89E-5	8.83E-5	8.72E-5	8.58E-5	8.39E-5	8.16E-5	7.88E-5	7.57E-5	7.2E-5	6.8E-5	6.34E-5	5.83E-5	5.83E-5	5.83E-5	5.83E-5	5.83E-5	5.83E-5	5.83E-5
12.5	9.14E-5	9.12E-5	9.05E-5	8.95E-5	8.8E-5	8.6E-5	8.37E-5	8.08E-5	7.76E-5	7.39E-5	6.97E-5	6.5E-5	5.98E-5	5.98E-5	5.98E-5	5.98E-5	5.98E-5	5.98E-5	5.98E-5
13.0	9.36E-5	9.34E-5	9.27E-5	9.16E-5	9.01E-5	8.81E-5	8.57E-5	8.28E-5	7.95E-5	7.57E-5	7.14E-5	6.66E-5	6.12E-5	6.12E-5	6.12E-5	6.12E-5	6.12E-5	6.12E-5	6.12E-5
13.5	9.58E-5	9.56E-5	9.49E-5	9.38E-5	9.22E-5	9.02E-5	8.77E-5	8.48E-5	8.13E-5	7.74E-5	7.3E-5	6.81E-5	6.26E-5	6.26E-5	6.26E-5	6.26E-5	6.26E-5	6.26E-5	6.26E-5
14.0	9.8E-5	9.78E-5	9.71E-5	9.59E-5	9.43E-5	9.22E-5	8.97E-5	8.67E-5	8.32E-5	7.92E-5	7.47E-5	6.97E-5	6.4E-5	6.4E-5	6.4E-5	6.4E-5	6.4E-5	6.4E-5	6.4E-5
14.5	1.E-4	9.99E-5	9.92E-5	9.8E-5	9.64E-5	9.42E-5	9.16E-5	8.86E-5	8.5E-5	8.09E-5	7.63E-5	7.12E-5	6.54E-5	6.54E-5	6.54E-5	6.54E-5	6.54E-5	6.54E-5	6.54E-5
15.0	1.02E-4	1.02E-4	1.01E-4	1.E-4	9.84E-5	9.62E-5	9.36E-5	9.04E-5	8.68E-5	8.26E-5	7.79E-5	7.27E-5	6.68E-5	6.68E-5	6.68E-5	6.68E-5	6.68E-5	6.68E-5	6.68E-5
15.5	1.04E-4	1.04E-4	1.03E-4	1.02E-4	1.E-4	9.82E-5	9.55E-5	9.23E-5	8.85E-5	8.43E-5	7.95E-5	7.41E-5	6.82E-5	6.82E-5	6.82E-5	6.82E-5	6.82E-5	6.82E-5	6.82E-5
16.0	1.06E-4	1.06E-4	1.05E-4	1.04E-4	1.02E-4	1.E-4	9.74E-5	9.41E-5	9.03E-5	8.6E-5	8.11E-5	7.56E-5	6.95E-5	6.95E-5	6.95E-5	6.95E-5	6.95E-5	6.95E-5	6.95E-5

File: PNP_Template Date: 2/17/92

GEOMETRY MODEL
Model: Created by AreaS Macro
Element: Details next to Area Table
Range:
Σ Area [m^2]:

AREA

km/s \ °	0°	5°	10°	15°	20°	25°	30°	35°	40°	45°	50°	55°	60°	65°	70°	75°	80°	85°	90°
0.5	8.8E-5	4.91E-5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1.0	8.33E-5	2.11E-4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1.5	0	4.54E-4	3.73E-5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2.0	0	4.28E-4	3.31E-4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2.5	0	2.31E-4	9.11E-4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
3.0	0	0	1.42E-3	2.74E-4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
3.5	0	0	1.17E-3	1.31E-3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
4.0	0	0	3.73E-4	3.19E-3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
4.5	0	0	0	3.67E-3	1.34E-3	0	0	0	0	0	0	0	0	0	0	0	0	0	0
5.0	0	0	0	2.44E-3	4.38E-3	0	0	0	0	0	0	0	0	0	0	0	0	0	0
5.5	0	0	0	0	8.79E-3	1.89E-4	0	0	0	0	0	0	0	0	0	0	0	0	0
6.0	0	0	0	0	6.79E-3	4.62E-3	0	0	0	0	0	0	0	0	0	0	0	0	0
6.5	0	0	0	0	2.88E-3	1.11E-2	0	0	0	0	0	0	0	0	0	0	0	0	0
7.0	0	0	0	0	0	1.33E-2	3.11E-3	0	0	0	0	0	0	0	0	0	0	0	0
7.5	0	0	0	0	0	7.63E-3	1.1E-2	0	0	0	0	0	0	0	0	0	0	0	0
8.0	0	0	0	0	0	0	2.03E-2	1.44E-17	0	0	0	0	0	0	0	0	0	0	0
8.5	0	0	0	0	0	0	1.24E-2	8.93E-3	0	0	0	0	0	0	0	0	0	0	0
9.0	0	0	0	0	0	0	3.36E-3	1.84E-2	0	0	0	0	0	0	0	0	0	0	0
9.5	0	0	0	0	0	0	0	1.55E-2	6.15E-3	0	0	0	0	0	0	0	0	0	0
10.0	0	0	0	0	0	0	0	5.59E-3	1.56E-2	0	0	0	0	0	0	0	0	0	0
10.5	0	0	0	0	0	0	0	0	1.66E-2	4.24E-3	0	0	0	0	0	0	0	0	0
11.0	0	0	0	0	0	0	0	0	6.6E-3	1.45E-2	0	0	0	0	0	0	0	0	0
11.5	0	0	0	0	0	0	0	0	0	1.77E-2	4.17E-3	0	0	0	0	0	0	0	0
12.0	0	0	0	0	0	0	0	0	0	6.57E-3	1.67E-2	0	0	0	0	0	0	0	0
12.5	0	0	0	0	0	0	0	0	0	0	1.78E-2	6.76E-3	0	0	0	0	0	0	0
13.0	0	0	0	0	0	0	0	0	0	0	3.22E-3	2.12E-2	0	0	0	0	0	0	0
13.5	0	0	0	0	0	0	0	0	0	0	0	1.04E-2	1.08E-2	0	0	0	0	0	0
14.0	0	0	0	0	0	0	0	0	0	0	0	0	1.04E-2	2.73E-3	0	0	0	0	0
14.5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
15.0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
15.5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
16.0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

File: PNP_Template Date: 2/17/92

Exp. Time 1 years

PNP = 0.999975
1-PNP = 2.5E-5

FLUX * AREA * TIME

km/s \ °	0°	5°	10°	15°	20°	25°	30°	35°	40°	45°	50°	55°	60°	65°	70°	75°	80°	85°	90°
0.5	1.1E-09	6E-10	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1.0	3.5E-09	8.8E-09	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1.5	0	5.2E-08	4.2E-09	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2.0	0	8E-08	5.9E-08	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2.5	0	3.8E-08	1.4E-07	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
3.0	0	0	1.9E-07	3.6E-08	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
3.5	0	0	1.4E-07	1.5E-07	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
4.0	0	0	3.9E-08	3.2E-07	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
4.5	0	0	0	3.3E-07	1.2E-07	0	0	0	0	0	0	0	0	0	0	0	0	0	0
5.0	0	0	0	1.9E-07	3.4E-07	0	0	0	0	0	0	0	0	0	0	0	0	0	0
5.5	0	0	0	0	6.1E-07	1.4E-08	0	0	0	0	0	0	0	0	0	0	0	0	0
6.0	0	0	0	0	4.2E-07	3E-07	0	0	0	0	0	0	0	0	0	0	0	0	0
6.5	0	0	0	0	1.7E-07	6.6E-07	0	0	0	0	0	0	0	0	0	0	0	0	0
7.0	0	0	0	0	0	8.3E-07	1.9E-07	0	0	0	0	0	0	0	0	0	0	0	0
7.5	0	0	0	0	0	4.9E-07	6.9E-07	0	0	0	0	0	0	0	0	0	0	0	0
8.0	0	0	0	0	0	0	1.3E-06	9.1E-22	0	0	0	0	0	0	0	0	0	0	0
8.5	0	0	0	0	0	0	8.3E-07	5.8E-07	0	0	0	0	0	0	0	0	0	0	0
9.0	0	0	0	0	0	0	2.3E-07	1.2E-06	0	0	0	0	0	0	0	0	0	0	0
9.5	0	0	0	0	0	0	0	1.1E-06	4.1E-07	0	0	0	0	0	0	0	0	0	0
10.0	0	0	0	0	0	0	0	3.9E-07	1.1E-06	0	0	0	0	0	0	0	0	0	0
10.5	0	0	0	0	0	0	0	0	1.2E-06	2.8E-07	0	0	0	0	0	0	0	0	0
11.0	0	0	0	0	0	0	0	0	4.7E-07	9.9E-07	0	0	0	0	0	0	0	0	0
11.5	0	0	0	0	0	0	0	0	0	1.2E-06	2.8E-07	0	0	0	0	0	0	0	0
12.0	0	0	0	0	0	0	0	0	0	4.7E-07	1.1E-06	0	0	0	0	0	0	0	0
12.5	0	0	0	0	0	0	0	0	0	0	1.2E-06	4.4E-07	0	0	0	0	0	0	0
13.0	0	0	0	0	0	0	0	0	0	0	2.3E-07	1.4E-06	0	0	0	0	0	0	0
13.5	0	0	0	0	0	0	0	0	0	0	0	7.1E-07	6.7E-07	0	0	0	0	0	0
14.0	0	0	0	0	0	0	0	0	0	0	0	0	6.6E-07	1.8E-07	0	0	0	0	0
14.5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
15.0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
15.5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
16.0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

File: PNP_Template Date: 2/17/92

Area Template - Used with Area Maker Macro

No.	Geom	Dimensions [meters]			Area	Rotation [deg]		Angle (Lat.)[deg]		Incr.	Longitude(Sph)		Surface
		L1	L2	L3	Multiplie	Pitch	Yaw	Start	Finish	[deg]	Start	Finish	Area [m^2]
1	Rectang	1	1		1	•	90°						1
												Total	1

Inclination= 00.0°

Effective Area: 0

Options	L1	L2	L3
Rectang	Length	Height	
Disk	Radius		
Cylinde	Radius	Length	
Cone	Radius1	Radius2	Length
Sphere	Radius		

File: PNP_Template Date: 2/17/92

Appendix I. Typical FORTRAN Input and Output.

```
**********************

   A_SURF VER 1.3

Last Update 2/17/92

**********************

Binary output filename?<CR=DATA.ASB> >PLATE_ON_EDGE.ASB

Text output filename?<CR=DATA.AST> >PLATE_ON_EDGE.AST

 ONE Area Fraction Table will be created
 from ALL of the ranges of element IDs selected.
 INPUT THE STARTING AND ENDING ELEMENT ID FOR EACH RANGE
 ENTER D <CR> OR <CR> WHEN DONE

RANGE    1 IN THE TABLE.
STARTING ELEMENT ID : 1
ENDING ELEMENT ID   : 2

RANGE    2 IN THE TABLE.
STARTING ELEMENT ID :

GEOMETRY OUTPUT FILENAME (<CR>=STATION.GEM  ) > PLATE.GEM

     Debris Analysis

     JSC-7/90 Memo

Processing Property ID     4

RANGES= 1 First PID=        4    EFF. AREA =     0.35370

The Area Surface file is complete.
binary filename: PLATE_ON_EDGE.ASB
text filename: PLATE_ON_EDGE.AST

STOP
```

```
Output from MacII_P_surf                                    Mon, Feb 17, 1992

****************************

   Space Debris SURFace

     Ver. 1.4 1/31/92

****************************

OUTPUT FILENAME (CR=SDSURF.PS)>PLATE_ON_EDGE.PSURF

ENVIRONMENT ?
 1-JSC 20001&6000 <CR>
 2- 7/90 MEMO
ANSWER 1 OR 2 > 2

SOLAR FLUX LEVEL ?
 1-NOMINAL <CR>
 2-MINIMUM
 3-CONSTANT
ANSWER 1-3 >

DATE TO BEGIN EXPOSURE ( 1994-2025 )  (<CR>=1995) >

SPACE STATION EXPOSURE TIME (YEARS) (<CR>=10.0) > 1

OPERATING ALTITUDE(100.-500.km)  (<CR>=388.92)
OR ENTER AN "E" OR "e" TO ENTER IN NMILES >

Area_Surface Binary Output File' <CR=DATA.ASB> :PLATE_ON_EDGE.ASB

RESPONSE OUTPUT FILENAME (<CR>=STATION.RSP) > ONE_RESPONSE.RSP

     Constant density threat

The one case in the RESPONSE file will be used


RESPONSE PID:   1     RESPONSE FILE: ONE_RESPONSE.RSP
A_SURF FILE: PLATE_ON_EDGE.ASB
PNP(%)=   99.99709     Total Flux x Area x Time (NAT) = 0.29084E-04
CONTOURS .12345 at equal increments from 0 to max NAT = 0.20709E-05

                              IMPACT VELOCITY  km/s

Obl    1   2   3   4   5   6   7   8   9  10  11  12  13  14  15  16
Deg    I   I   I   I   I   I   I   I   I   I   I   I   I   I   I   I
 90 .................................................................
 85 .................................................................
 80 .................................................................
 75 .................................................................
 70 .......................................................111.......
 65 .....................................................11111.......
 60 ...................................................2111..........
 55 ................................................14.11............
 50 ...........................................11.5111...............
 45 ........................................21.31.11.................
 40 ....................................14..31.......................
 35 .................................32.12...........................
 30 ..........................11.15..11..............................
 25 ......................21..51.11..................................
 20 ..................21..31.........................................
 15 .............12...11.............................................
 10 .....11..11..11..................................................
```

```
  5 .11..11..11.....................................................
  0 .11.............................................................
```

The PNP Surface file is complete.
filename: PLATE_ON_EDGE.PSURF

PLATE_ON_EDGE.AST.xl

GEOMETRY OUTPUT FILE = PLATE.GEM

GEOMETRY PARAMETERS :
Input = PLATE.UNI
Output = PLATE.GEM
Summary = GEOMETRY.SUM

Threat (1 2 Meteoroid) 1

Environme 2 New) 2
Number of Threats 45
INCLINATION (DEGREES) = 28.5
RANGES= 1 First PID= 4 EFF. AREA = 0.35370
1 START/END 1 2

1	0	5	10	15	20	25	30	35	40	45	50	55	60	65	70	75	80	85	90
0.25	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
0.5	1.66E-04	1.11E-04	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
0.75	2.92E-05	1.95E-05	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
1	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
1.25	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
1.5	0.00E+00	5.32E-04	1.33E-04	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
1.75	0.00E+00	4.16E-04	1.04E-04	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
2	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
2.25	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
2.5	0.00E+00	2.13E-09	8.60E-04	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
2.75	0.00E+00	4.90E-09	1.98E-03	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
3	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
3.25	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
3.5	0.00E+00	0.00E+00	1.21E-04	4.83E-04	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
3.75	0.00E+00	0.00E+00	1.12E-03	4.47E-03	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
4	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
4.25	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
4.5	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
4.75	0.00E+00	0.00E+00	0.00E+00	4.66E-03	6.99E-03	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
5	0.00E+00	0.00E+00	0.00E+00	1.71E-04	2.57E-04	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
5.25	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
5.5	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
5.75	0.00E+00	0.00E+00	0.00E+00	0.00E+00	1.13E-02	7.50E-03	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
6	0.00E+00	0.00E+00	0.00E+00	0.00E+00	9.22E-04	6.15E-04	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
6.25	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
6.5	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
6.75	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	2.31E-02	5.77E-03	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
7	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	8.49E-05	2.12E-05	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
7.25	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
7.5	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	8.08E-09	7.06E-03	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
7.75	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	3.23E-08	2.83E-02	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
8	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
8.25	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
8.5	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	4.14E-03	1.66E-02	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
8.75	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	3.34E-03	1.33E-02	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
9	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00

PLATE_ON_EDGE.AST.xl

9.25	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	1.07E-03	1.61E-03	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
9.5	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	1.32E-02	1.98E-02	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
9.75	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
10	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
10.25	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	1.52E-02	1.01E-02	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
10.5	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	4.25E-03	2.83E-03	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
10.75	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
11	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	1.67E-02	4.18E-03	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
11.25	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	7.55E-03	1.89E-03	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
11.5	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
11.75	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	1.84E-08	2.41E-02	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
12	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	4.27E-09	5.60E-03	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
12.25	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	9.46E-04	3.79E-03	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
12.5	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	4.81E-03	1.92E-02	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
12.75	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
13	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	7.62E-03	1.14E-02	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
13.25	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	2.40E-03	3.61E-03	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
13.5	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	6.51E-03	4.34E-03	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
13.75	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	4.16E-03	2.77E-03	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
14	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	4.94E-03	1.23E-03	0.00E+00	0.00E+00	0.00E+00	0.00E+00
14.25	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	1.87E-03	5.80E-04	0.00E+00	0.00E+00	0.00E+00	0.00E+00
14.5	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	1.33E-09	8.71E-04	0.00E+00	0.00E+00	0.00E+00	0.00E+00
14.75	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
15	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
15.25	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
15.5	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
15.75	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
16	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
16.25	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
16.5	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
16.75	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
17	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00

```
****************************

   Space Debris SURFace

     Ver. 1.4 1/31/92

****************************


7/17/90 MEMO FLUX EQUATIONS

MAN-MADE ORBITAL DEBRIS ANALYSIS
NOMINAL SOLAR FLUX LEVEL
DATE TO BEGIN EXPOSURE  = 1995.000
SPACECRAFT EXPOSURE TIME (YEARS) =   1.000
OPERATING ALTITUDE (km) =  388.920
A_SURF BINARY OUTPUT FILE = PLATE_ON_EDGE.ASB
RANGES= 1   PID=        4   EFF. AREA (sq.m) =      0.35370
Range  1  START:            1END:             2


RESPONSE PARAMETERS :
Threat (1 D  2 Meteoroid)     1
Density (1 · 2 Function)     1
         Number of PID Cases       1
         Units   ENGLISH

PID NUMBER    1

REGRESSION PENETRATION FUNCTION

Configuration     Shield      Wall
Double Plate    6061-T6     2219-T87
      Shield Thickness =   0.0300
 Vessel Wall Thickness =   0.1250
              Standoff =   4.0000
    With 30 Layers of MLI against vessel wall
RESPONSE OUTPUT FILE = ONE_RESPONSE.RSP
          1
RESPONSE PID:   1     RESPONSE FILE: ONE_RESPONSE.RSP
A_SURF FILE: PLATE_ON_EDGE.ASB
PNP(%)=   99.99709     Total Flux x Area x Time (NAT) = 0.29084E-04
CONTOURS .12345 at equal increments from 0 to max NAT = 0.20709E-05

                          IMPACT VELOCITY  km/s
```

PLATE_ON_EDGE.PSURF

```
Obl   1   2   3   4   5   6   7   8   9  10  11  12  13  14  15  16
Deg   I   I   I   I   I   I   I   I   I   I   I   I   I   I   I   I
 90 ................................................................
 85 ................................................................
 80 ................................................................
 75 ................................................................
 70 .......................................................111......
 65 .....................................................11111......
 60 ...................................................2111.........
 55 ...............................................14.11............
 50 ...........................................11.5111..............
 45 ........................................21.31.11................
 40 ....................................14..31......................
 35 ..................................32.12.........................
 30 ............................11.15..11...........................
 25 ........................21..51.11...............................
 20 ....................21..31......................................
 15 ............12...11.............................................
 10 .....11..11..11.................................................
  5 .11..11..11.....................................................
  0 .11.............................................................
        1
RESPONSE PID:   1
```

1	0	5	10	15	20	25	30	35	40	45	50	55	60
0.25	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
0.5	2.64E-09	1.75E-09	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
0.75	8.45E-10	5.59E-10	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
1	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
1.25	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
1.5	0.00E+00	8.71E-08	2.09E-08	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
1.75	0.00E+00	9.17E-08	2.18E-08	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
2	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
2.25	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
2.5	0.00E+00	5.14E-13	1.98E-07	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
2.75	0.00E+00	1.10E-12	4.25E-07	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
3	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
3.25	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
3.5	0.00E+00	0.00E+00	2.13E-08	8.30E-08	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
3.75	0.00E+00	0.00E+00	1.84E-07	7.19E-07	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
4	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
4.25	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
4.5	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
4.75	0.00E+00	0.00E+00	0.00E+00	5.83E-07	8.74E-07	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
5	0.00E+00	0.00E+00	0.00E+00	2.01E-08	3.03E-08	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
5.25	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
5.5	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00

PLATE_ON_EDGE.PSURF

5.75	0.00E+00	0.00E+00	0.00E+00	0.00E+00	1.12E-06	7.63E-07	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
6	0.00E+00	0.00E+00	0.00E+00	0.00E+00	8.63E-08	5.92E-08	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
6.25	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
6.5	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
6.75	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	1.90E-06	5.00E-07	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
7	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	6.63E-09	1.75E-09	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
7.25	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
7.5	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	5.98E-13	5.31E-07	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
7.75	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	2.43E-12	2.07E-06	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
8	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
8.25	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
8.5	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	3.18E-07	1.23E-06	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
8.75	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	2.60E-07	1.01E-06	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
9	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
9.25	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	8.31E-08	1.20E-07	0.00E+00	0.00E+00	0.00E+00	0.00E+00
9.5	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	1.03E-06	1.49E-06	0.00E+00	0.00E+00	0.00E+00	0.00E+00
9.75	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
10	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
10.25	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	1.19E-06	7.61E-07	0.00E+00	0.00E+00	0.00E+00
10.5	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	3.38E-07	2.16E-07	0.00E+00	0.00E+00	0.00E+00
10.75	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
11	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	1.31E-06	3.09E-07	0.00E+00	0.00E+00
11.25	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	5.98E-07	1.42E-07	0.00E+00	0.00E+00
11.5	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
11.75	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	1.49E-12	1.85E-06	0.00E+00	0.00E+00
12	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	3.50E-13	4.35E-07	0.00E+00	0.00E+00
12.25	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	7.44E-08	2.79E-07	0.00E+00
12.5	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	3.82E-07	1.43E-06	0.00E+00
12.75	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
13	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	5.81E-07	8.07E-07
13.25	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	1.85E-07	2.57E-07
13.5	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	4.70E-07
13.75	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	3.03E-07
14	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
14.25	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
14.5	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
14.75	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
15	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
15.25	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
15.5	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
15.75	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
16	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
16.25	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
16.5	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
16.75	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
17	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00

PLATE_ON_EDGE.PSURF

65	70	75	80	85	90
0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00

PLATE_ON_EDGE.PSURF

0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
3.13E-07	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
2.02E-07	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
3.63E-07	9.08E-08	0.00E+00	0.00E+00	0.00E+00	0.00E+00
1.39E-07	4.31E-08	0.00E+00	0.00E+00	0.00E+00	0.00E+00
9.97E-14	6.54E-08	0.00E+00	0.00E+00	0.00E+00	0.00E+00
0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00

☆U.S. GOVERNMENT PRINTING OFFICE–1996–733-109/40006

REPORT DOCUMENTATION PAGE	Form Approved OMB No. 0704-0188

1. AGENCY USE ONLY (Leave Blank)	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
	February 1996	Contractor Report (Final)

4. TITLE AND SUBTITLE
User's Manual for Space Debris Surfaces (SD_SURF)

5. FUNDING NUMBERS
NAS8-38856

6. AUTHOR(S)

N. C. Elfer

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

Lockheed Martin Marietta Manned Space Systems
P.O. Box 29304
New Orleans, LA 70189

8. PERFORMING ORGANIZATON REPORT NUMBERS
M-799

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

George C. Marshall Space Flight Center
Marshall Space Flight Center, Alabama 35812

10. SPONSORING/MONITORING AGENCY REPORT NUMBER
NASA CR-4705

11. SUPPLEMENTARY NOTES
Technical Monitor: Joel Williamsen, Structures and Dynamics Laboratory, Science and Engineering Directorate

12a. DISTRIBUTION/AVAILABILITY STATEMENT

Unclassified - Unlimited

Subject Category 18

12b. DISTRIBUTION CODE

13. ABSTRACT ***(Maximum 200 words)***

A unique collection of computer codes, Space Debris Surfaces (SD_SURF), have been developed to assist in the design and analysis of space debris protection systems. SD_SURF calculates and summarizes a vehicle's vulnerability to space debris as a function of impact velocity and obliquity. An SD_SURF analysis will show which velocities and obliquities are the most probable to cause a penetration. This determination can help the analyst select a shield design which is best suited to the predominant penetration mechanism. The analysis also indicates the most suitable parameters for development or verification testing. The SD_SURF programs offer the option of either FORTRAN programs and Microsoft EXCEL spreadsheets and macros. The FORTRAN programs work with BUMPERII version 1.2a or 1.3 (COSMIC released). The EXCEL spreadsheets and macros can be used independently or with selected output from the SD_SURF FORTRAN programs.

14. SUBJECT TERMS
bumpers, debris shields, hypervelocity impacts, impact, meteoroids, orbital debris, probability of no penetration, space debris

15. NUMBER OF PAGES
208

16. PRICE CODE
A10

17. SECURITY CLASSIFICATION	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
Unclassified	Unclassified	Unclassified	Unlimited

National Aeronautics and
Space Administration
Code JTT
Washington, DC
20546–0001