

ASAP Version 7.5
REFERENCE MANUAL

Breault Research Organization, Inc.
6400 East Grant Road
Tucson, Arizona 85715 USA

Phone: 520-721-0500
FAX: 520-721-9630
Internet: www.breault.com

This document is proprietary with Breault Research Organization, Inc. Recipient agrees not to reproduce this document in whole or in part, nor to redistribute it without the express consent of the Breault Research Organization, Inc.

15 February 2003

CONTENTS

CHAPTER 1 INTRODUCTION

1.1	FEATURES LIST	1-2
-----	-------------------------	-----

CHAPTER 2 GENERAL INPUT STRUCTURE AND TECHNIQUES

2.1	INPUT RECORDS	2-1
2.2	COMMANDS	2-2
2.2.1	Command Description Notation	2-3
2.3	DELIMITED COMMENT STRINGS	2-3
2.4	ARBITRARY COMMENTS	2-4
2.5	ARITHMETIC OPERATIONS	2-4
2.5.1	Registers And Variables For Storage Of Intermediate Arithmetic Results	2-6
2.5.2	Complex-valued Operations	2-7
2.6	REPEATED AND INCREMENTED ENTRIES	2-8
2.7	MACRO COMMANDS	2-8
2.7.1	Creating A Macro Library File	2-9
2.7.2	Calling A Macro	2-10
2.7.3	Defining A Macro From The Input Stream	2-11
2.7.4	Calling A Macro From A Library Other Than The Current	2-12
2.7.5	Optimizing The Speed Of Macro Command Execution	2-13
2.7.6	Internal Predefined Macro Commands	2-13
2.7.6.1	Switching Expression Syntax	2-14
2.7.6.2	Specifying Input Case Sensitivity	2-14
2.7.6.3	Controlling The Echoing Of Input	2-14
2.7.6.4	Controlling Argument Prompting In User Macros	2-14
2.7.6.5	Temporarily Reading Input From A Given File	2-15
2.7.6.6	General I/O Redirection	2-15
2.7.6.7	Copying/moving One File To Another	2-17
2.7.6.8	Executing User Macros From A Menu	2-17
2.7.6.9	Looping Through Single Or Multiple Records .	2-18
2.7.6.10	Premature Exit From A Loop, Macro, Or Input File	2-19
2.7.6.11	Jumping To Other Records Or Skipping Over Input Records	2-20

2.7.6.12	Branching On An Input Error	2-20
2.7.6.13	Conditional Processing Of A Block Of Records	2-21
2.7.6.14	Conditional Block Structures	2-22
2.7.6.15	On-line Help Facility	2-23
2.7.6.16	Immediately Terminating Program Execution	2-23
2.7.6.17	Inserting A Formfeed And/or String Into The Output Stream	2-23
2.7.6.18	Executing A System Command From Within The Program	2-24
2.7.6.19	Displaying The Contents Of The Registers	2-24
2.7.6.20	Setting A Register By Interactive Prompting	2-24
2.7.6.21	Editing/displaying Register Fields On A User-definable Screen	2-25
2.7.6.22	Storing Or Recalling Entire Register Set To Or From A File	2-26
2.7.6.23	Fast Reading Of Numeric Arrays From Files	2-26
2.7.6.24	Debug Listing Of Parser Output	2-27
2.7.6.25	Displaying Elapsed And CPU Time	2-27
2.7.6.26	Defining New In-line Math Functions	2-28
2.7.6.27	Replaying Buffered Output And Input	2-28
2.7.6.28	Grabbing Numbers Or Strings From The Output Buffer	2-29
2.7.6.29	Clearing The Output Buffer	2-29
2.7.6.30	Iterating Through Input Records And A Set Of Registers	2-29
2.7.6.31	Rapidly Evaluating Functions Of A Set Of Registers	2-31
2.7.6.32	Suppressing Any Date And/or Clock Time Outputs	2-32
2.7.6.33	Beeping	2-32
2.7.6.34	On-line Editing Of Files, Macros And Portions Of Input File	2-32
2.7.6.35	On-line Processing Of Graphics	2-33
2.7.6.36	Display Current Command States	2-34
2.7.6.37	Pausing Output	2-34
2.7.6.38	Setting The Default Directory Path	2-35
2.7.6.39	Pausing Program Execution	2-35
2.7.6.40	Local Macro Register/variables	2-35
2.7.6.41	Controlling The GUI Front-end	2-36
2.7.6.42	Controlling Uninitialized Variable Messages	2-36
2.7.6.43	Resetting The Random Operator Seed	2-37
2.7.6.44	Manually Aborting A Calculation	2-37

2.8	ERRORS IN INPUT	2-37
-----	---------------------------	------

CHAPTER 3 INPUT COMMANDS

3.1	INTRODUCTION	3-1
3.2	SUMMARY LISTING OF COMMANDS BY FUNCTION	3-2
3.3	GENERAL INPUT TECHNIQUES	3-11
3.3.1	Radian And Direction Cosine Angle Entries	3-11
3.3.2	Relative And Literal Referencing Of Entities	3-11
3.3.3	Direction Vectors	3-12
3.3.4	Linear Transformations	3-13
3.3.4.1	Translation Operations	3-14
3.3.4.2	Distorting Operations	3-15
3.3.4.3	Rotation Operations	3-16
3.3.4.4	General Operations	3-16
3.4	DEFINING THE GEOMETRY OF THE SYSTEM	3-17
3.4.1	Control Of Entity Updating In Storage	3-18
3.4.2	Curve Or Edge Entities	3-18
3.4.2.1	Edges With No Assumed Symmetry	3-18
3.4.2.1.1	An Equally Divided Straight Line	3-19
3.4.2.1.2	General Spline From Points And Directions	3-20
3.4.2.1.3	Curve Or Surface Net From Users Parametric Function	3-20
3.4.2.2	Polygonal Edges Initially Normal To A Particular Coordinate Axis	3-21
3.4.2.2.1	An Ellipse (or Circle)	3-21
3.4.2.2.2	A Rectangle (or Square)	3-22
3.4.2.2.3	A General Oval	3-22
3.4.2.2.4	A Rectangle With Crudely Rounded Corners	3-23
3.4.2.3	Polycurves Initially Normal To A Particular Coordinate Axis	3-23
3.4.2.3.1	A Rectangle With Precisely Rounded Corners (racetrack)	3-23
3.4.2.3.2	An Edge Patterned After A Text Character	3-24
3.4.2.3.3	A Circular Arc	3-24
3.4.2.3.4	Quadratic Segment From Implicit Coefficients Of Conic	3-25
3.4.2.3.5	Sawtooth Pattern Edge	3-25
3.4.2.3.6	Explicit Polynomial As Same-order Bezier	3-26
3.4.2.3.7	Explicit Spline From Coordinates And Slopes	3-26
3.4.2.4	General Helical Curve	3-27

3.4.2.5	Combining A Set Of Previous Edges Into One .	3-27
3.4.2.6	Distorting Edges	3-28
3.4.2.7	Altering Specific Data On A Set Of Edge Points	3-29
3.4.2.8	Reversing The Parametric "direction" Of A Curve	3-29
3.4.2.9	Smoothing A Piecewise Linear Curve	3-29
3.4.2.10	Coarsening A Curve	3-30
3.4.2.11	Specifying The Optional Sweeping Of A Curve Into A Surface	3-30
3.4.2.12	Imaging A Curve/edge Through A Lens	3-30
3.4.2.13	Rational Bezier Surface Patches	3-31
3.4.2.14	Curves Defined In The Parametric Space Of An Object	3-31
3.4.2.15	Linearly Extend One Or Both Ends Of A Curve	3-32
3.4.3	Surface Or Function Entities	3-32
3.4.3.1	Input For Surfaces With No Preferred Symmetry	3-35
3.4.3.1.1	Arbitrarily Oriented Planes	3-35
3.4.3.1.2	An Ellipsoid (or Sphere)	3-36
3.4.3.1.3	Least Squares Fit Of An Arbitrary Set Of Points To A Surface	3-36
3.4.3.2	The User-programmable Surface Intersection .	3-38
3.4.3.3	The User-programmable Function	3-38
3.4.3.4	Input For Surfaces Initially Normal To A Coordinate Axis	3-40
3.4.3.4.1	Orthogonal Planes	3-40
3.4.3.4.2	Circular Cross Section Horns With Polynomial Profiles	3-41
3.4.3.4.3	General Rectangular/elliptical Cross Section Tubes	3-42
3.4.3.4.4	A Torus (doughnut)	3-43
3.4.3.4.5	A Classical Symmetric Optical Surface . .	3-43
3.4.3.4.6	Surface From Two Distinct Conic Profiles .	3-44
3.4.3.4.7	Surface With A User-defined Radial Or Toric Profile	3-44
3.4.3.4.8	Surfaces Swept Out By A Rotated Curve . .	3-45
3.4.3.4.9	The Perfect Cartesian Ovaloid	3-47
3.4.3.4.10	Explicit Zernike Polynomial Surface . . .	3-47
3.4.3.4.11	Explicit Surface Interpolated From Sampled Data	3-48
3.4.3.4.12	Deformed Cylindrical Surface Interpolated From Sampled Data	3-49

3.4.3.4.13	Circle Swept Along An Explicit Cubic Planar Curve	3-50
3.4.3.4.14	Superconic Asphere Used In Optical Design	3-50
3.4.3.5	Altering Surfaces	3-51
3.4.3.5.1	Nonlinear Distortion Of A "normal" Surface	3-51
3.4.3.5.2	Creating A Multiple Sheet Version Of A Surface	3-52
3.4.3.5.3	Creating Simple Arrays Of Surfaces	3-52
3.4.3.5.4	Parabolic Bending Along A Given Axis	3-53
3.4.3.6	Localizing A Surface With A Simple Bounding Box	3-54
3.4.3.7	Contour Map Of A Cut Of The 3-dimensional Surface Function	3-55
3.4.3.8	Testing For A Particular Branch Of A Surface Function	3-55
3.4.3.9	Renormalizing The Coefficients Of A Surface Function	3-56
3.4.3.10	Bounding Volume Formed From Surface And Its Local Box	3-56
3.4.3.11	Controlling Parameterization For Meshing Surface	3-57
3.4.3.12	Altering Specific Polynomial Coefficients	3-58
3.4.3.13	Converting To Explicit Form	3-58
3.4.4	Lens Entities	3-58
3.4.4.1	General Sequences Of Conicoids	3-59
3.4.4.2	Standard Lens Elements	3-60
3.4.4.3	Perfect Eikonal-based (realistic) Lens	3-63
3.4.4.4	Ideal Lens Element (ABCD And Jones Matrices)	3-63
3.4.4.5	Altering Specific Data On A Set Of Lens Conicoids	3-64
3.4.4.6	Imaging A Global Point Through The Lens	3-65
3.4.4.7	Displaying The Aberrations Of A Centered Imaging System	3-66
3.4.4.8	Locally Or Globally Optimizing A Centered Imaging System	3-67
3.5	DEFINING OPTICAL PROPERTIES	3-71
3.5.1	System Geometry Length Units And Flux Label	3-71
3.5.2	Interpolation Wavelengths	3-72
3.5.3	Refractive Media	3-72
3.5.3.1	Absorption Or Gain Option	3-73
3.5.3.2	Gradient Option	3-73
3.5.3.3	Birefringent Uniaxial Crystal Option	3-74

3.5.3.4	Inhomogeneous Monte-Carlo Scattering Option	3-74
3.5.4	Coatings	3-75
3.5.5	Directional Scatter Models	3-78
3.5.5.1	Standard Options For Scatter Model Input . .	3-78
3.5.5.2	Direct Data Entry Options And Format	3-79
3.5.5.3	Isotropic-surface Scattering	3-80
3.5.5.3.1	Simple Specific Models	3-80
3.5.5.3.2	General Mathematical Models For Isotropic-surface Scattering	3-81
3.5.5.3.2.1	Linear In Free Parameters	3-81
3.5.5.3.2.2	Nonlinear In Free Parameters	3-83
3.5.5.3.3	User Definable Isotropic-surface Scatter Model	3-85
3.5.5.3.4	Models Based On Physical Properties . . .	3-86
3.5.5.3.4.1	Physical Optics Models Of Surface With Random Height Variations	3-86
3.5.5.3.4.2	A Random Distribution Of Spherical Particles	3-87
3.5.5.3.4.3	Rough Reflective Surface Of Random V-cavities	3-88
3.5.5.3.5	Indirectly Interpolating From Entered BSDF Data	3-89
3.5.5.4	Anisotropic-surface Scattering	3-89
3.5.5.4.1	Effective Surface Scattering Of Vane Structure	3-90
3.5.5.4.2	Elliptical Harvey Model	3-91
3.5.5.4.3	Nonlinear In Free Parameters	3-92
3.5.5.4.4	User Definable Anisotropic-surface Scatter Model	3-93
3.5.5.4.5	Directly Interpolating From Entered BSDF Data	3-94
3.5.5.5	Sum Of Other Scattering Models	3-95
3.6	DEFINING OBJECTS	3-95
3.6.0.1	Redefining Some Basic Options	3-96
3.6.0.2	Controlling Subdivision Of Mesh Patches Into Facets	3-97
3.6.0.3	Assigning Surface Roughnesses	3-97
3.6.0.4	Adding Small Deformations To An Object Surface	3-99
3.6.0.5	Bounding (and Slicing) Of Objects	3-99
3.6.0.5.1	Rough Orthogonal Bounds	3-99
3.6.0.5.2	General Boundary Surfaces And Edges . . .	3-101

3.6.0.6	Assigning Optical Properties To Object Interfaces	3-101
3.6.0.6.1	Scatter Characteristics	3-103
3.6.0.6.1.1	Deterministic	3-103
3.6.0.6.1.2	Random	3-104
3.6.0.7	Refraction/reflection Controls	3-105
3.6.0.8	Streamlined Entity And Object Input	3-107
3.6.0.9	Moving Groups Of Objects And Sources As A Whole	3-108
3.6.0.10	Exploding Out Lenses Into Separate Surface Or Curve Based Objects	3-109
3.6.0.11	Specifying A Branch Of The Name Tree For Future Objects	3-109
3.7	INPUT/OUTPUT CONTROL COMMANDS	3-110
3.7.1	Controlling Execution	3-110
3.7.2	Displaying Important Program Information . .	3-111
3.7.3	Saving/retrieving System Data To/from A Disk File	3-113
3.7.4	Exporting A CAD Representation Of The System .	3-113
3.7.5	Limiting The Current Set Of "visible" Objects	3-114
3.8	GRAPHICS CONTROL AND GENERATION	3-114
3.8.1	Sampling	3-115
3.8.2	Geometrical Data Window	3-115
3.8.3	Oblique (nonorthographic) Views	3-116
3.8.4	Perspective Views	3-117
3.8.5	Arrows On Ray And Polarization State Plots .	3-118
3.8.6	Number Of Segments Plotted Per Arc	3-118
3.8.7	Coloring Of Objects By Group And/or Interface Properties	3-118
3.8.8	Plotting Slices Through Objects	3-119
3.8.9	Geometrical Entities That Can Be Directly Plotted	3-119
3.8.10	Replotting Accumulated 3D Graphics Data . .	3-121
3.8.11	Rendering Of The System Geometry	3-123
3.8.12	Mapping Of The System Geometry	3-124
3.8.13	Standard Plot Command Options	3-125
3.8.13.1	Overlayed And Multiple Window Graphics . .	3-125
3.8.13.2	Drawing 2D And/or 3D Graphics Text	3-126
3.8.13.3	Overriding Default Colors With A New Set Of Colors	3-126
3.8.13.4	Temporarily Overriding Number Of Pixels . .	3-127
3.8.13.5	Temporarily Overriding Current Plot Window .	3-127

3.9	RAY TRACING COMMANDS	3-127
3.9.1	Commands For Setting Options Before Trace . .	3-127
3.9.1.1	Accuracy Versus Speed Of Ray/surface Intersection Calculations	3-128
3.9.1.2	Controlling The Object Intersection Search .	3-128
3.9.1.3	Setting Conditions On Terminating Trace . .	3-129
3.9.1.4	Refraction/reflection Controls	3-130
3.9.1.5	Trace Output Control	3-132
3.9.1.6	Random Ray Generation	3-133
3.9.1.7	Saving All Future Ray Intersections To A Binary Data File	3-133
3.9.1.8	Setting Up Volume Energy Tracking And Visualization	3-135
3.9.2	Commands For Setting Up Beam Characteristics .	3-136
3.9.2.1	Positional And/or Directional Clipping Of Rays During Creation	3-138
3.9.2.2	Spectral Apodization	3-140
3.9.2.3	User Apodization Of Ray Distributions . . .	3-141
3.9.2.4	Paging Of Extended Ray/beam Memory	3-145
3.9.3	Executing A Single Ray Trace	3-145
3.9.4	Tracing A Group Of Rays Or Beams	3-146
3.9.4.1	Initialization Of Ray And Beam Data	3-146
3.9.4.1.1	Ray Tables	3-147
3.9.4.1.2	Ray Grids	3-148
3.9.4.1.3	Setting Ray Directions According To Source Configuration	3-150
3.9.4.1.4	Setting Ray Directions According To Desired Wavefront Function	3-151
3.9.4.1.5	Creating Rays To Simulate A Random Emitter	3-152
3.9.4.1.5.1	Emitting Surfaces	3-152
3.9.4.1.5.2	Emitting Geometry	3-152
3.9.4.1.5.3	Emitting Volumes	3-153
3.9.4.1.5.4	Emitting Space Curves (wires)	3-154
3.9.4.1.5.5	Emitter Specified By An IES Standard Photometric Data File	3-155
3.9.4.1.5.6	Arbitrary Emitting Ray Set	3-156
3.9.4.1.5.7	Emit According To Distribution In A Binary File	3-156
3.9.4.1.5.8	Composite Single-source Emitter	3-157
3.9.4.1.6	Ray/beam Set For A Coherent Astigmatic Gaussian Mode	3-158

3.9.4.1.7	Decomposing An Existing Arbitrary Field Into A New Gaussian Set	3-158
3.9.4.2	Selecting And Modifying The Current Ray Set	3-160
3.9.4.2.1	Ray Energy Flux	3-162
3.9.4.2.2	Local Beam Shapes	3-162
3.9.4.2.3	Folding Over Ray Coordinates Due To Symmetry	3-163
3.9.4.2.4	Moving Along Straight Ray Paths To A New Location	3-164
3.9.4.2.5	Reversing Ray Directions	3-165
3.9.4.2.6	Imaging Ray Positions Through A Lens . . .	3-165
3.9.4.3	Execution Of Trace	3-166
3.10	ANALYZING THE CURRENT RAY AND BEAM DATA	3-167
3.10.1	Listing Data On A Ray-by-ray Basis	3-167
3.10.2	Listing Data On The Extreme Rays	3-168
3.10.3	Statistics On Current Ray Set	3-168
3.10.4	Statistics On Previous Trace	3-169
3.10.5	Grouping And Listing Current Ray Set By Paths	3-169
3.10.6	Complete Positional Ray Histories	3-170
3.10.7	Best Focus Determination	3-170
3.10.8	Obtaining A Full Field Aberration Display . .	3-171
3.10.9	Accessing Internal Ray Data Directly	3-172
3.10.10	Geometrical Spot Diagrams	3-173
3.10.11	Full-Spherical Far-field Incoherent Radiant Intensity Or Radiance Pattern	3-174
3.10.12	Collection Efficiency As A Function Of Aperture Size And Cone Angle.	3-175
3.10.13	Interpolating OPD Data And Generating A Normalized PSF	3-176
3.10.14	Dumping Currently Selected Rays To A Binary File	3-177
3.10.15	Standard Distribution Clipping Option	3-177
3.10.16	Creating Distributions In The Local Coordinates Of An Entity	3-177
3.11	COHERENT/INCOHERENT FIELD/ENERGY FROM CURRENT BEAMS	3-178
3.11.1	Summing Coherent Beams (possibly Polychromatic And/or Polarized)	3-181
3.11.2	Field Propagation Through Small Geometries And Inhomogeneous Media	3-182
3.11.3	Displaying Data Distributions	3-184
3.11.3.1	Data Processing	3-186

3.11.3.2	Data Plotting	3-190
3.11.3.3	Data File Operations	3-193
3.11.3.4	Distribution File Structure	3-197
3.11.3.5	Inputting An Arbitrary Data Distribution . .	3-198

CHAPTER 4 SEPARATE UTILITY PROGRAMS

4.1	CONVERSION OF 3D VECTOR FILE TO AUTOCAD DXF FORMAT	4-1
4.2	PROGRAMS FOR CONVERTING APART FILES INTO ASAP INPUT	4-1
4.3	CONVERSION OF AUTOCAD DXF FILES TO ASAP INPUT . .	4-2
4.4	SUPPORTED ENTITIES AND RESTRICTIONS ON IGES IMPORT	4-2
4.5	CONVERTING BMP BITMAP IMAGE FILES TO/FROM DISTRIBUTION DATA FILES	4-4

APPENDIX A DEVICE INDEPENDENT GRAPHICS PROTOCOLS

A.1	TWO-DIMENSIONAL PLOT FILES	A-1
A.1.1	Basic Organization Of Graphics Routines	A-1
A.1.2	The Central Plot Routine Protocol	A-2
A.1.3	Directly Accessing The Immediate Graphics Driver	A-3
A.2	THREE-DIMENSIONAL GRAPHICS FILES	A-5

APPENDIX B REVISION HISTORY AND RELEASE NOTES

B.1	VERSION 7.5	B-2
B.2	VERSION 7.1	B-5
B.3	VERSION 7.0	B-7
B.4	VERSION 6.6	B-10
B.5	VERSION 6.5	B-12
B.6	VERSION 6.0	B-15
B.7	VERSION 5.1	B-20
B.8	VERSION 5.0	B-24
B.9	VERSION 4.1	B-26
B.10	VERSION 4.0	B-28
B.11	VERSION 3.3	B-32
B.12	VERSION 3.2	B-33
B.13	VERSION 3.0	B-37
B.14	VERSION 2.9	B-41

B.15	VERSION 2.8	B-43
B.16	VERSION 1.8	B-45
B.17	VERSION 1.0 REFERENCE MANUAL (1983)	B-46
B.17.1	INTRODUCTION	B-46
B.17.2	PROGRAM COMMANDS	B-46
B.17.2.1	Defining The Geometry Of The System	B-47
B.17.2.1.1	Surfaces	B-47
B.17.2.1.1.1	Special Surfaces With No Preferred Symmetry	B-48
B.17.2.1.1.2	Special Surfaces With An Initial Axis Of Symmetry	B-49
B.17.2.1.1.3	Altering Surfaces	B-50
B.17.2.1.1.4	Nonlinear Transformations	B-51
B.17.2.1.1.5	Linear Transformations	B-51
B.17.2.1.2	Media	B-52
B.17.2.1.2.1	Special Case Of Reflecting Substrate	B-53
B.17.2.1.3	Objects	B-53
B.17.2.1.4	Relative Indexing	B-53
B.17.2.2	Input/Output Control Commands	B-54
B.17.2.3	Raytracing Commands	B-55
B.17.2.3.1	Optional Commands	B-55
B.17.2.3.2	Single Ray Trace	B-57
B.17.2.3.3	Tracing A Group Of Rays	B-57
B.17.2.3.3.1	Initialization Of Ray Data	B-58
B.17.2.3.3.2	Execution Of Trace	B-60
B.17.2.3.3.3	Analyzing The Final Ray Data	B-61

CHAPTER 1

INTRODUCTION

BRO's ASAP is an extremely powerful, state-of-the-art software tool that combines advanced 3D surface/structure modeling techniques with a very flexible ray tracing and coherent/incoherent beam propagation algorithm. ASAP has been specifically designed to avoid the inherent limitations in other optical analysis codes by implementing: ray/beam propagation along physically realizable paths; automatic ray/beam splitting into reflected, refracted, diffracted, polarized, and scattered components; wavefront propagation capable of determining coherent/incoherent spread functions at any location, orientation or time; hardware/operating system independent kernel.

ASAP can perform polychromatic geometrical/physical optics analysis of reflection, transmission, diffraction, scattering, polarization, absorption, multiple beam and gradient index effects in virtually any conceivable geometry including high-resolution imaging systems, non-imaging concentrators/illuminators, interferometers, HOE/grating optics, laser resonators, segmented aperture/field systems, and complete sensors with baffle structure. A sophisticated macro command language complete with screen/menu capability permits the user to customize the program for specific problems. Straightforward interfaces to other optical and CAD software are also provided.

1.1 FEATURES LIST

General user features:

- o Easy I/O redirection, e.g. switching between interactive and batch modes.
- o Flexible unformatted command and macro command structure.
- o Arithmetic operations, functions, variables, conditional branches, etc.
- o On-line help by pull-down windows and syntax checking during input.
- o Programmable menus/screens and interactive data prompts.
- o Device independent 2-D and 3-D color graphics screens and/or files.

Performs accurate straightforward modeling of any 3D system:

- o Uses single global coordinate system with no assumed symmetry.
- o Surface entities are general 10-20th order 3D polynomials or user-definable.
- o Models toroids, Cartesian Ovals, axicons, and most exotic surfaces exactly.
- o Curve entities are defined by an arbitrary sequence of lines/arcs in space.
- o Lens entities are arbitrary sequences of reflective or refractive conicoids.
- o Objects are one or more entities bounded by other entities.

Advanced ray/beam propagation capability:

- o Independent ray/beam trace, i.e. transfers to nearest valid object.
- o Splitting into reflected, refracted, diffracted, and scattered ray/beams.
- o Volume absorption/scatter and interface effects included in ray/beam flux.
- o Differential quantities used to calculate ray/beam cross section.
- o Wave propagation by powerful Gaussian beam decomposition technique.
- o Built-in virtual memory management for very large ray/beam sets.

Comprehensive graphics, output, and interface generation:

- o Spot diagrams and incoherent/coherent spread functions at ANY location.
- o Interactive panning and zooming over object and ray/beam graphics.
- o Interactive tracing of rays by "pointing and shooting" on screen.
- o Ray/beam statistics and histories available before/during/after trace.
- o Interfaces to other packages (APART, Lens Design, CAD) are provided.
- o Many other capabilities are user-definable via macro or Fortran routines.

CHAPTER 2

GENERAL INPUT STRUCTURE AND TECHNIQUES

The APART and ASAP software programs distributed by BRO utilize the same input processing routines. They give the programs many advanced capabilities including unformatted alphanumeric entries, inline arithmetic operations, I/O redirection (including automatic switching from disk to keyboard input), on-line help, and a powerful macro command language with programmable menus and data prompts.

2.1 INPUT RECORDS

Input data is read sequentially from records up to 344 characters long (anything past this or a double exclamation point "!!" is ignored). The program first attempts to read input from a file named "defsetup.in?" (the last character in the extension depends on the particular application program). This file should contain any default input settings that would normally be used in every program run. After processing the input from this file, the program starts reading from either the file specified as the first command line argument or if in batch mode, logical unit 1 (BRO001.DAT). The file that this unit has been assigned to (or the system default specification) should have already been created using the system editor. If an end-of-file is reached during the reading of the input disk file and before program execution has been terminated with the proper input command, the program will automatically switch over to prompting the user (with a greater-than sign ">") for data entry

directly from the keyboard. This feature allows the user to execute the program from a file and/or in interactive mode.

Maximum input record length:	344 characters
Premature record terminator:	!!
Startup fetching sequence:	defsetup.in? -> command argument -> Keyboard (BR0001.DAT batch)
Keyboard input prompt:	??>

2.2 COMMANDS

Program commands consist of up to 10,000 unformatted alphanumeric entries separated by one or more of blanks " " or commas ",". Arbitrary string entries can be delimited by double quotes. Otherwise, literal entries consist of an unbroken string of alphanumeric characters (A through Z through 9, and _). They must start with a letter or underscore "_" and can be of any length (but only the first 8 may be significant). Numeric entries can be in integer, floating point or exponential format. Exponential entries, for example "1.34E-5", cannot be longer than 40 characters.

Usually a command begins with a literal and there is only one command in each input record. However, more than one command can be placed in a record by separating them with semicolons ";". Also a command can extend over more than one record. If the last non-blank character in an input record is a comma ",", then the command is continued over to the next record.

Entry delimiters:	blanks and commas
Maximum number of entries per command:	10,000
Literal entries:	begin with a letter
Commands:	usually start with a literal
Maximum exponential numeric entry length:	40 characters
Multiple commands per record separator:	semicolon
Command continuation:	comma last character of record

2.2.1 Command Description Notation

In the chapter sections that describe the program commands, the commands are, for clarity, preceded by three dashes and a space "--- " (just as they are echoed during execution), i.e.

```
--- command ...
```

The format of each command is then shown followed by a detailed explanation of the command's function. A standard notation is used for the quantities shown. Any upper case letters or numbers represent the actual literal form of the entry. Single lower case letters represent numerical entries to be determined by the user. Lower case words represent literal entries that can take on the described set of values. Trailing lower case letters on literals and anything enclosed within square brackets are optional and will not trigger a program error if omitted. Unless otherwise stated, the default value for most optional numeric entries is zero. Alternate forms for an entry will be shown in the same vertical column.

2.3 DELIMITED COMMENT STRINGS

User comment strings may be imbedded in any command without usually affecting execution (they can be meaningful to some commands). Enclose them in an entry delimiter followed by one of the following:

Comment string delimiters: * = / \ < > ' '

If a command begins with any of these characters, then the rest of that command line is ignored. Comment strings delimited in this manner can be of any length (up to 344 characters) and composition (except matching delimiters), and provide the user with a convenient facility for documenting input files.

2.4 ARBITRARY COMMENTS

Letters opposite in case to that set by the \$CASE command can also be used as imbedded comments since the program treats them as blanks. Trailing comments can be entered after an exclamation point "!" since this character signals the program to stop decoding input from the command.

```
Opposite case letters :      treated as blanks
Last entry, stop parsing:  exclamation point !
```

2.5 ARITHMETIC OPERATIONS

The program is capable of performing a variety of arithmetic operations during the input decoding phase. When two numerical entries are separated by (instead of the usual delimiters) one of the following symbols, then the two numerical entries are replaced by the result of the corresponding operation.

Operation	Description	NEW Precedence
		Level
+	ADD the two numerical entries (or concatenate two literals)	4
-	SUBTRACT second from first	4
*	MULTIPLY the two	5
^	RAISE first TO second POWER	6
/	DIVIDE first by second	5
\	REMAINDER after dividing first by second	6
<	Take LESSER of the two	6
>	Take GREATER of the two	6
%	Angle in degrees whose ARCTANGENT is the first divided by second	6
~	Uniformly distributed RANDOM number between first and second	7
'	Form real pair or complex number from REAL and IMAGINARY parts	0
'	Form complex number from MODULUS and PHASE angle (in degrees)	0
([	Store value of expression and operator to the left	1
)]	Recall previous value and operator, evaluate new expression	2

Note that the two pair/complex operators are found under the tilde and double quote, respectively, on most keyboards. When \$EXP is set to OLD, consecutive operations are always evaluated from left to right with no operator precedence

until a delimiter terminates the expression. Nested parentheses or brackets can be used when necessary. In addition, the following set of mathematical functions are also supported:

	Closing right character	
	)	]
-----+-----+-----		
INT	Truncate to integer	Round to nearest integer
EXP	Natural (base e) antilog	Common (base 10) antilog
LOG	Natural (base e) logarithm	Common (base 10) logarithm
ABS	Absolute value	
SGN	Sign (returns -1, 0, or +1)	
SQRT	Square root	
CBRT	Cube root with same sign	
SIN	Sine of angle in RADIANS	Sine of angle in DEGREES
COS	Cosine	Cosine
TAN	Tangent	Tangent
ASIN	Arcsine	Arcsine
ACOS	Arccosine	Arccosine
ATAN	Arctangent	Arctangent
BJ#	#th-order Bessel J function (# from 0 to 9)	
BK#	" Modified Bessel K "	
STEP	0 for X<0, 1 for X>0	
RECT	0 for X >.5, 1 for X <.5	
GAUS	Gaussian EXP(-X^2)	Gaussian EXP(-PI*(X^2))
SINC	SIN(X)/X	SIN(PI*X)/(PI*X)
SOMB	2*BJ1(X)/X	2*BJ1(PI*X)/(PI*X)
FACT	X factorial (actually GAMMA(X +1) with sign of X)	
ERF	Error function	Complementary 1-ERF
FILE	Next available file number starting at X (usually 10)	
	Blackbody integral from 0 to X=microns*degreesK	
FBI	Fractional Energy	Fractional Photons
LPW	Lumens per Watt for X degreeK blackbody	
	Normalized Visual Response at X microns	
EYE	Photopic (bright)	Scotopic (dim)
	Numeric to literal conversion	
LIT	8 or less characters	16 or less characters
	Symmetric zero-mean random numbers of type INT(X) (-15 to 15)	
RAN	Unit RMS	Unit Maximum

RAN Type	Probability Distribution P(Y)	(Max/RMS)^2
-----+-----+-----		
-15	Two delta functions	1
-14	Y ^14	(14+3)/(14+1)
:	:	:
-2	Y ^2	(2+3)/(2+1)
-1	Ramp	2
0	Uniform	3
1	Cosine	5
2	Convolution of 2 uniforms (triangle)	3*2
:	:	:
14	Convolution of 14 uniforms	3*14
15	Gaussian	2*LOG(2^32)

The fractional part of the argument to RAN is the relative amount (total probability) of an additional uniform variate with the same RMS or maximum, e.g. RAN[2.6] has a probability distribution that looks like a house.

The ability to perform arithmetic operations during input decoding is a powerful feature when used with the macro facility.

Examples:

	OLD	NEW
	2.-1./2. ==> .5	2.-1./2. ==> 1.5
	2.-(1./2) ==> 1.5	2.-(1./2) ==> 1.5
SIN[A.]^2+(COS[A.]^2)	==> 1.0	SIN[A]^2+COS[A]^2 ==> 1.0
SQRT((X.^2)+(Y.^2))'(Y%X.)	==> X.'Y.	SQRT(X^2+Y^2)'(Y%X) ==> X'Y
(3.-1.)^4	==> 16.	(3.-1.)^4 ==> 16.

2.5.1 Registers And Variables For Storage Of Intermediate Arithmetic Results

The program has 286 direct registers designated by the letters A through Z by themselves or followed by the numbers 0 through 9. (A special set of registers starting with an underscore "_" instead of a letter are reserved for argument passing). These registers can be used for the storage of both intermediate arithmetic results and literals. In the following examples, the letter R stands for any register. To store the value of any valid arithmetic expression or literal in a register (forming a null entry) or recall a value in a register

as an input number, use the following formats:

	OLD	NEW	Precedence Level
Store numeric in R:	expression=R	R=expression	3
Store literal in R:	literal=R	R="literal"	0
Recall numeric from R:	R.	(R)	
Recall literal from R:	R"	R"	
Recall numeric from register with literal:	literal.	(literal)	

where R is one of 286 registers: A...Z or A0...Z9

The registers are zeroed and blanked out at program start up. The capability to recall the numeric in a register by referencing the literal stored in that register allows the user to assign a register a variable name and then use that name (up to 32 characters) instead of the short fixed register name. An unknown variable is automatically assigned to an unused indirect register location starting at 1768 and working down, i.e. up to 1482 user variables can be created before any conflict with the normal direct register set occurs.

As an example, the following input increments the contents of the register / variable and uses the result as the current input entry:

```
R.+1=R.      !increment register/variable and use result (OLD)
(R=R+1)      !increment register/variable and use result (NEW)
```

If the period had been left off the end of the OLD expression or the parentheses left off the NEW, then the register would still be incremented, but no input would be passed to the program. The current contents of the registers / variables can be displayed using the \$REG predefined macro command discussed in a later section.

2.5.2 Complex-valued Operations

Only one operation is allowed on a complex entry: real / imaginary assignment. The two parts are assigned to two different registers / variables as follows:

name=x

```
name=x'y    ==>
              _name=y
```

2.6 REPEATED AND INCREMENTED ENTRIES

An at-sign "@" imbedded between two entries allows the user to enter redundant data in an efficient manner. For example,

```
3@1.4    ==>  1.4 1.4 1.4
-4@1.4    ==> -1.4 1.4 -1.4 1.4
```

A colon ":" between two integer entries can be used to represent an increasing or decreasing sequence of integers, e.g.

```
3:7      ==>  3 4 5 6 7
4:-1     ==>  4 3 2 1 0 -1
```

If one or both of the entries are less than one in magnitude, then an increment that is also less than one in magnitude is used. For example,

```
.1:.5    ==>  .1 .2 .3 .4 .5
.11:.08  ==>  .11 .10 .09 .08
```

A colon ":" between any two entries that differ by only one character or a set of contiguous integers can be used to generate a simple sequence, e.g.

```
ALITERAL:DLITERAL  ==>  ALITERAL BLITERAL CLITERAL DLITERAL
R11:R0             ==>  R11 R10  R9 R8 R7 R6 R5 R4 R3 R2 R1 R0
```

2.7 MACRO COMMANDS

The macro command capability allows the user to replace often used input sequences with a single command followed by variable alphanumeric arguments.

2.7.1 Creating A Macro Library File

Multiple macro definitions can be stored in a library (logical unit 24 by default). Definitions may be created either directly by an ASCII editor or during the decoding of the input by the program. The latter method will be described in a later section.

The first record of a macro definition contains the name of the macro (up to 32-characters), the number of definition records that follow (or an up to 8-character label that marks the end of the definition records), and possibly the number of argument prompt records after that. This record is read in an unformatted manner with the name starting in column 1 and the numerical entries separated by blanks. Therefore, any characters past the last numerical entry can be used for documentation purposes (e.g. a description of the purpose or arguments of the macro). The macro name must start with a letter and can be the same as a standard program command without causing conflict (for an exception see section on internal predefined macros).

The definition records that follow contain the commands that make up the definition. User macro definitions can reference other user macro commands up to 99 levels deep. The internal predefined macros do not change the level of macro nesting. Up to 99 variables may be used in the definition. These are represented by a pound sign "#" followed by a 1 or 2-digit integer between 1 and 99 inclusive. When the macro is called during input decoding these 2 or 3 characters are replaced by the characters of the corresponding argument of the macro command. Therefore, macro variables may represent literal and numeric quantities of any length. However, an expanded macro record cannot exceed 344 characters.

Default macro library file logical unit:	24
Format of macro initialization record:	name m [n] label {
Macro arguments (string substitution):	#1 through #99
Number of arguments actually present:	#0
Maximum level of macro call nesting:	99

An example of a macro definition is:


```
NAME 2 1                2 definition + 1 argument prompt records
SOURCE .01 #1
H 1.01 C 2.01
Enter collector object number:

ANALYZE 10              next definition with no argument prompts
:
```

Any number of blank lines or comment lines (starting with an exclamation point "!" or asterisk "*") may be inserted before the start of each definition in order to enhance readability of the macro library file.

2.7.2 Calling A Macro

A macro command is unformatted like all other commands. There can be only one user macro command per input record and it should be the last command on the record. The first entry of a macro command is either a dollar sign "\$" or ampersand "&" followed immediately by the macro name. The name can be of any length but only the first 32 characters are significant. The dollar sign "\$" causes the program to print the commands in the macro definition after the variables have been replaced by their corresponding arguments. Only the macro command itself is printed when the ampersand "&" is used.

The remaining entries on the macro command are the optional arguments to be substituted for the variables in the macro definition. If an argument is not present on the macro command that is referenced in the macro definition, then the program prompts the user for the argument in interactive mode or the variable is replaced with a null entry.

Macro name prefixes (echo,no echo of macro records): \$ &

When the previous example is called, it would be expanded as follows:

```
--- :  
--- $NAME 10.02  
-1- SOURCE .01 10.02  
-1- H 1.01 C 2.01  
--- :  
--- &NAME  
Enter collector object number: 5.01  
--- :
```

2.7.3 Defining A Macro From The Input Stream

Instead of defining macros in the library file, one can create individual macro files directly from the input stream. There are two formats, a short or long, respectively:

```
--- name { [ n ] [ description ]  
      :  
      definition records  
      :  
--- }
```

or

```
--- $$name m [ n ] ...  
      $& label  
      &$  
      &&  
--- :  
      m records in macro definition or until "label" literal record reached  
      :  
---[label]
```

Long format macro definitions can be created from the input stream by prefixing the name with two macro prefix characters and entering on successive records the "m" records in the definition. An alternative to specifying the integer number of records "m" in advance is to use a record starting with the given literal (e.g. "_") after the last definition record (and before the "n" argument prompt definitions). The first prefix character determines whether

the macro is defined and executed (\$), or just defined (&). As usual the second character determines whether the individual records in the macro are displayed (\$) or not (&).

Prefix	Action
-----+	-----
\$\$	Define, execute, and echo
\$&	Define and execute
&\$	Define and echo
&&	Just define

For example, to simply define a macro NAME using a meaningful label:

```
--- &&NAME EOM
    :
--- EOM
```

Note, as opposed to \$GO and \$ERR, "label" here must be in the currently acceptable case (see \$CASE). Actually, the above can also be in short format as follows:

```
--- NAME {
    :
--- }
```

2.7.4 Calling A Macro From A Library Other Than The Current

Normally, the program searches for a macro definition in the currently loaded library (see \$IO LIB command). However, this can be overridden on a case-by-case basis using the following syntax:

```
--- library$name ...
    &
```

"library" is the name of the library file (a .LIB extension will be added) and "name" is the desired macro in that file. The program looks for the library file first in the current working directory (see \$PATH), then in the same directory as the currently loaded library (\$IO LIB), and finally in the

directory specified by the FILES switch or environment variable.

2.7.5 Optimizing The Speed Of Macro Command Execution

The following are some suggestions for maximizing the execution speed of user-defined macro commands.

- o Don't put too many definitions in one macro library file. If possible, break up a large library file into a number of smaller files and then use the \$IO LIBRARY command to switch between them.
- o Order your macro definitions within the library file from smallest to largest, or most used to least used.
- o Put several commands on each macro definition record in order to keep the the total number of records in the macro down to a minimum.
- o If possible, use registers to pass variables instead of the macro arguments. A macro that is called with no arguments is executed faster than one with arguments.

2.7.6 Internal Predefined Macro Commands

There are several very useful macros that are internally defined by the program. The user should not define a macro with the same name as any of these internal macros. Also, these macros differ from the user defined ones in that more than one can be placed on a single input record.

2.7.6.1 Switching Expression Syntax -

```
--- $EXP [ OLD ]  
          NEW
```

Instructs the parser to recognize either the OLD left-to-right or the NEW precedence expression syntax. The default in the absence of any \$EXP command is NEW. If no argument is given on the command, the state before the last \$EXP command is restored.

2.7.6.2 Specifying Input Case Sensitivity -

```
--- $CASE [ UPPER ]  
          LOWER  
          BOTH
```

Instructs the parser to recognize only UPPER or LOWER case letters (or BOTH). The default in the absence of any \$CASE command is UPPER. If no argument is given on the command, the state before the last \$CASE command is restored.

2.7.6.3 Controlling The Echoing Of Input -

```
--- $ECHO [ ALL ]  
          NONE
```

Echo (redisplay) ALL input commands or NONE. This overrides the control of echoing specified on the call to a user macro. A \$ECHO command by itself (no argument) restores this default mode.

2.7.6.4 Controlling Argument Prompting In User Macros -

```
--- $ARGS [ ALL ]
          NONE
          USER
          SCR
```

Controls future prompting of unspecified arguments on a user macro. Interactive prompts are supplied for either ALL, NONE or only the ones defined by the USER in the macro definition. The default in the absence of any \$ARGS command is ALL. If no argument is given on the command, the state before the last \$ARGS command is restored.

The SCR option always pops up an editable template for those arguments for which prompts have been defined.

2.7.6.5 Temporarily Reading Input From A Given File -

```
--- $READ name
```

Starting reading future records from the beginning of the given file "name" (with default extension .IN?). If end-of-file is reached, return to calling macro or file automatically. \$READ commands can be directly or indirectly nested.

2.7.6.6 General I/O Redirection -

```
--- $IO iotype n          iotype' ...
          literal
          file [ n ]
--- $IO [ n ]
```

Redirects the ASCII IO specified by one of the seven "iotype"s below to another "file" name and/or unit number "n".

"iotype"	n	Defaults		Description
		Name	Ext	
INPUT	1	BR0001.DAT	in?	Standard command input
LIBRARY	24	BR0024.DAT	lib	Macro library (new precedence expressions)
MACRO	"	"	"	" (old left-right expressions)
OUTPUT	6	BR0006.DAT	ot?	Standard non-graphical text output
GRAPHICS	"	"	"	Character graphics output (normally off)
USER	7	BR0007.DAT	us?	Archive of interactively entered commands
PLOT	20	BR0020.DAT	pl?	Standard 2D plot instruction output
VECTOR	30	BR0030.DAT	vc?	3D vector graphics instructions
FILE	-	-	-	An arbitrary file (user must specify n)

An "iotype" entry without any trailing arguments, either closes a unit for reading or reinstates writing to a CANCELED unit. The optional short format redirects program INPUT only. A prefix character for each number or a literal determines which of the following operations are performed before the unit is used:

Literal form	Numeric form	Description of Operation
DELETE	0n	Unit n is closed and the associated file deleted
REWIND	-n	Unit n is positioned at the beginning of the file
APPEND	+n	Unit n is positioned at the end of the file
CHANGE	n	The current position of unit n is not changed
CANCEL	0	OUTPUT, GRAPHICS, PLOT or VECTOR is suppressed
CLOSE	none	Unit associated with "iotype" is closed

If no entries are present, this macro toggles the program input between the user's console and the current input unit assignment. This allows one to insert interactive "breakpoints" in the input file.

When directing OUTPUT to a file, it is by default also echoed to the console. If the word ONLY instead of the unit number is placed after the file name, then nothing is sent to the console, i.e.

```
$IO OUTPUT file n !both to file and console
$IO OUTPUT file BOTH !same
$IO OUTPUT file ONLY !just to file
```

2.7.6.7 Copying/moving One File To Another -

```
--- $COPY source destination
    &COPY  s      d
    $MOVE
    &MOVE
```

Do a raw byte-for-byte copy of the source file (name "source" or number "s") to a destination file (name "destination" or number "d"). If the destination file does not exist, it is created. Otherwise, \$COPY overwrites it while &COPY appends to it. The command works with all file types (text and binary), but it usually doesn't make any sense to do an append with one or both files binary. Some examples are:

```
$COPY MODE.DIS 29      !copy mode.dis to bro029.dat
&COPY NEW.LIB UTIL.LIB !add new library to util.lib
$COPY 30 SAVE.VCR      !copy current 3D vectors to save.vcr
```

The MOVE versions behave identically to COPY except that the source file is deleted afterwards.

2.7.6.8 Executing User Macros From A Menu -

```
--- $MENU [ m ] [ 'title' ]
    &MENU
```

Causes the program to display a menu of current macro library names (and any descriptions found on the definition line) with an optional "title" at the top. \$MENU displays the menu in text mode. The user can then select, either by number or abbreviated name, the macro to be executed. &MENU displays the menu in graphics mode. The user can then select the macro to be executed by using the graphics cursor and pressing the space bar or the left mouse button. If the user makes no selection (just a carriage return or right button), the program returns to standard command mode. Another MENU command will be needed to see the menu again. On the other hand, after the selected macro has executed, the program also returns to command mode. However, only a null command entry (just a carriage return) is needed to see the menu again.

The menu entries are displayed horizontally in columns by default. "m" is the

maximum number of columns to use (default 6). For example, entering a 1 would force the menu entries into one vertical column.

2.7.6.9 Looping Through Single Or Multiple Records -

The program has the capability to read a single input record (or multiple records delimited by braces) over and over again as determined by a counter whose updated value is available within the record(s) at each iteration. The command just before the single record to be looped through must have the following format:

```
--- $DO [ i [ j [ k ] ] ]  
    &DO  
--- ... [ ? ] ...  
    |
```

"i" is the starting value of the counter (default 1), "j" the ending value (default 10000), and "k" the increment (default + or - 1) that is added to the counter at each iteration. The counter is treated as an integer and the loop is terminated only when it exactly equals or exceeds the ending value. The sign of "k" is always set to be the same as the sign of "j-i" so that the loop is always executed at least once. For \$DO, a question mark "?" can be used in the record to represent the digits of the current integer counter. For &DO, a vertical bar "|" is used instead. It is not necessary to have a reference to the loop counter in the record. Only one level of looping per macro nesting level is allowed. The loop parameters may be macro arguments themselves or used as arguments for some other macro. An input error or a \$LEAVE in the record will force a premature exit from the loop. For example, the \$DO could be used to perform a scalar product of two register "vectors":

```
S=0 $DO 0 9  
S=S+U?*V?
```

Multiple record do loops must be enclosed in braces, i.e. the next record after the DO and the first record of the block must start with an open brace "{" and the last must end with a closed brace "}".

```
--- $DO i j [ k ]  
--- { ...  
      :  
---      ... }
```

```
--- $NEXT
```

Immediately begins execution of the next \$DO or \$ITER iteration by forcing a branch to the top of the current loop block. For example.

```
$DO 1 10  
{ :  
  This block is executed every iteration  
  :  
  $IF ? LE 5; $NEXT  
  :  
  This block is executed only for iterations 6 through 10  
  : }
```

2.7.6.10 Premature Exit From A Loop, Macro, Or Input File -

```
--- $LEAVE [ n ]
```

Forces an immediate exit from the current DO loop, macro, or input file (in that order). For example, the equivalent to a "while" loop would be:

```
$DO  
{ $IF ... ; $LEAVE  
  :  
  }
```

"n" is the number of construct levels to exit (default 1). If the above loop was also inside a macro and you wanted the test to exit not only the loop but also the macro, then set "n" to 2.

2.7.6.11 Jumping To Other Records Or Skipping Over Input Records -

```
--- $GO m
      +m
      -m
```

If the second entry is an integer "m", then next record fetched will either be macro record "m" or the next general record plus or minus "m". If the target record goes beyond the end of a macro definition, then macro execution is terminated after the current record is processed.

```
--- $GO label
      :
--- label
      :
```

If the second entry is not an integer, then the program skips over (reads but does not process) input records until a record with the alphanumeric string "label" (up to 8 characters) is found starting in the first column (character position 1). Normal input processing resumes on the next record. Jumping back to a previous record in the input file can be accomplished by first rewinding the file, e.g.

```
$IO INPUT REWIND; $GO top !backwards branch example for input from a file
```

However, this input by itself can lead to an infinite loop.

2.7.6.12 Branching On An Input Error -

```
--- $ERR label
      :
--- label
      :
```

Sets a flag such that when an input error occurs, the program skips records until one starting with the alphanumeric string "label" is found. Normal input processing resumes on the next record and the error flag is reset.

2.7.6.13 Conditional Processing Of A Block Of Records -

```
--- $IF a rel b [ log a' rel' b' log' ... ] [ n ]
```

The "a" and "b" entries (either both numerical or both literal) are compared using the specified "rel"ational operator. More complex tests are evaluated from left to right using the "log"ical operators. The allowable operators are:

rel		log	
EQ	Equal	AND	Logical "and"
LT	Less than	OR	Logical "or"
GT	Greater than	EQV	Equal in logical value
NE	Not equal	NEQV	Not equal in logical value
GE	Greater than or equal	XOR	Exclusive "or" (same as NEQV)
LE	Less than or equal		

Normally, the relational operators compare two floating point (real) entries or two full 40 character literals (including trailing blanks). However, if any recognizable alphanumeric character is appended to the operator name, then truncated forms of the two entries are compared, i.e. the integer equivalents of the numerics or the literals truncated to the smaller of the two. Some examples are:

```
1.75 EQ 1.      false
1.75 EQI 1      true
YES EQ Y        false
YES EQS Y       true
```

If the overall \$IF result is true, the next "n" input records are processed (including any commands that follow the \$IF on its record). Otherwise, they are skipped over and processing resumes on the next record. However, if "n" is not specified and more commands follow the \$IF on its record, then the default for "n" is 0 records. Otherwise, the default is 1. This macro combined with others and registers can be used to implement extended DO blocks that includes more than one input record, e.g.

```

:
$ERR end                                !on error, go to "end" of loop
L=0                                     !initialize loop counter
L=L+1                                  !increment loop counter at top of loop
:
:                                     !block of n-3 input records
:
$IF (L) GE 10; $GO end                 !test for end of loop
$GO -n                                !otherwise go back to top of loop
end
:

```

2.7.6.14 Conditional Block Structures -

```

--- $IF a rel b [ 'log a' rel' b' log' ... ] THEN
:
--- [ $ELSEIF ... THEN ]
:
--- [ $ELSE ]
:
--- $ENDIF

```

Use the THEN keyword at the end of a \$IF statement to trigger processing of conditional block structures. These structures can be nested up to 99 deep. However, the control words MUST be uniquely and identically indented at each level of nesting. This mandatory indentation significantly speeds processing and is also a good standard programming practice. The following is an example of an IF structure inside a DO loop block.

```

$DO 1 3
  { $IF ?\2 EQ 0 THEN
    A??
    $ELSE
    B??
  $ENDIF }

```

Note the required vertical alignment of the \$IF,\$ELSE,\$ENDIF commands. Also, using a \$GO to jump into a block may cause unexpected results. Only a forward \$GO to a label can be safely used to jump out of a block.

2.7.6.15 On-line Help Facility -

```
--- $HELP [ command ]  
    &HELP  MACROS
```

Displays information on all or one command or a listing of the MACROS in the current library. If no entry is given, an abbreviated list of all commands is displayed. If only one command (up to two literals) is specified, a complete one or two paragraph explanation of the command is shown. The &HELP form displays only the format of the command with no accompanying text.

2.7.6.16 Immediately Terminating Program Execution -

```
--- $EXIT
```

Immediately terminates program execution and returns control to the operating system. If the program is in interactive mode, the user is asked to verify his/her decision to terminate program execution.

2.7.6.17 Inserting A Formfeed And/or String Into The Output Stream -

```
--- $FF [ text ]  
    &FF  k
```

If the dollar sign "\$" prefix is used, a page advance or formfeed is sent to the current output unit. Any text present on this command (up to 62 characters) will also be printed at the top of the page along with the current date and time. This macro can be used to document or "spruce up" program output. The integer entry form outputs just a formfeed to logical unit "k" instead. If the ampersand "&" prefix is used, just the text (up to 80 characters) preceded and followed by single blank lines is written.

2.7.6.18 Executing A System Command From Within The Program -

```
--- $SYS [ command line ]
```

The operating system "command line" is executed and control is returned to the program. The implementation of this feature is highly dependent on the particular operating system under which the program is executing. However, it can be usually used to copy, rename, list, and even edit files without exiting the program. On most systems, if the "command" is omitted then a secondary process is created so that operating system commands can be entered directly until the proper command (e.g. LOGOFF or EXIT) returns control to the program.

2.7.6.19 Displaying The Contents Of The Registers -

```
--- $REG [ register register' ... ] [ 'text' ]  
    &REG
```

The contents of the register (or registers) are displayed with optional "text". If there are no additional entries on the command, all the modified registers are displayed.

2.7.6.20 Setting A Register By Interactive Prompting -

```
--- &ASK [ register [ register' ... ] ] [ 'prompt' ]  
    $ASK
```

If in interactive mode, the program "prompt"s the user for the numeric value(s) (or literal when only one register given) for the "register"(s) or simply waits for a carriage return after the prompt. If only one register is given and the "\$" prefix is used, the current settings for the "register" are also displayed. For example,

```
--- L=0
--- $ASK L ' Enter total length of system'
    Enter total length of system [L:0] 12
--- $ASK ' Hit CR when ready to go on'
    Hit CR when ready to go on
---
```

Since output and input of the prompt and value, respectively, are always directed to the user's console, this technique works even if normal IO has been redirected or the request is made from within a non-echoing user macro.

2.7.6.21 Editing/displaying Register Fields On A User-definable Screen -

```
--- $SCR [ name ] [ 'title' ]
    m
```

This command can be used to define a screen template from which any group of register values may be displayed or modified by editing their assigned fields. The template may be read from a file called "name.SCR" or from the "m" input records that follow. In the former case and instead of the normal end-of-file, a line with just a closed brace "}" in column one can be used to terminate the template definition and signal the beginning of up to 1000 lines of 79 character-wide help text.

Within this screen template, each register field is delimited by either two back slashes "\" for display-only fields or a backslash and question mark "?" for modifiable fields. The register name immediately follows the first backslash. A colon, period, double quote or nothing at the end of the register name determines the register data type; boolean, floating point, literal or integer, respectively. If floating, a single digit number to the right of the period determines the number of digits to the right of the decimal point that will be displayed. The overall size of the field (and the register value displayed in it) are controlled by the position of the second delimiter. If there are no editable fields in the template (no question marks "?"), then the result is written directly as simple text to the current output device.


```
Screen definition file extension: SCR
Maximum editable screen size:      79x24 characters
Maximum output template size:     344x24 characters
Register field start delimiter:    \
Register field end delimiter:      \ ?
Literal field designator:         "
Integer field designator:
Floating point designator:         .
# digits right of decimal point:  .#
Boolean field designator:         :
```

2.7.6.22 Storing Or Recalling Entire Register Set To Or From A File -

```
--- $RCL [ file ]
    $STO      [ reg reg' ]
    &RCL
    &STO
```

Recalls or stores the range of given registers (or the entire register set) to or from a binary(\$) or text(&) "file" (default LASTVALS) with extension REG. Either form of RCL command can be used to read a binary or text file since, if it does not recognize the requested format, it will try the other.

2.7.6.23 Fast Reading Of Numeric Arrays From Files -

```
--- $FAST n [ m ] [ s ] [ COMPLEX ]
        label
--- ... n or 2n numbers
    :
    m records
--- [ label ]
```

Reads directly from the current input "m" records (default 1 or until "label" reached or until read error) of "n" real or COMPLEX entries each. The program uses the very flexible list-directed input facility of Fortran and therefore:

1. Does not do the extensive parsing it does normally, i.e. only numbers separated by a comma and/or blanks are allowed.
2. This input is never echoed to the output device.
3. If reading directly from a file and not indirectly from a macro or loop, will attempt to read "n" times "m" numbers with line breaks only required after every "n"th number, i.e. additional line breaks can be inserted so that lines are not limited to the normal 344 characters.

"s" is an optional real or complex scaling factor, i.e. each number read will be multiplied by "s" (default 1). Optionally, the program can also assume each successive pair of numbers forms a COMPLEX (real,imaginary) entry. In this case, rule 3 above should be read with "2n" substituted for "n".

2.7.6.24 Debug Listing Of Parser Output -

```
--- $DBG [ ... ]  
--- [ ... ]
```

This macro displays how the parser decoded a line of input into entries. The start, length, type, and value of each entry on the line is listed. Since complicated expressions can decode into a single entry, the results of expressions can be checked in this manner. If the line of input is on the \$DBG command itself, the line is not passed to the application program.

2.7.6.25 Displaying Elapsed And CPU Time -

```
--- $TIC [ units ]  
    &TIC
```

\$TIC displays the elapsed and CPU time since the last \$TIC or &TIC command was processed. &TIC displays nothing and just resets the timers. Normally the time "units" are allowed to float but can be fixed for this and all future outputs to MICroseconds, MILLiseconds, SECconds, MINutes, HOurs, DAYs, WEEks or YEArS. Use OFF to go back to the default floating units. The display resolution is always 1/100 of the selected units and is limited to less 10,000 total units.

2.7.6.26 Defining New In-line Math Functions -

```
--- $FCN [ name [ e e' e" ... ] ]
```

Define a new in-line math function with the given up to 32 character "name". The new function can then be used in future arithmetic expressions just like the intrinsic functions SIN, COS, SQRT, etc. The function is defined by the expressions entered after the name. Up to 60 of these functions can be defined at any one time. The result of the last expression is the function value returned during execution. The dummy argument of the function is represented in the defining expressions as the base argument register "_". For example,

```
--- $FCN SECH 2/(EXP(_)+EXP(-_))
```

If no expressions are given, the function "name" is deleted from internal storage. A \$FCN command by itself (no other entries) deletes all current user functions. These user-definable functions are recursive, i.e. one can be in the argument of another. Also, any registers/variables defined in the function before they are used are automatically assumed to be local to that instance of the function, and therefore do not conflict with any external (e.g. global) ones of the same name.

2.7.6.27 Replaying Buffered Output And Input -

```
--- $PLAY [ n ] [ INPUT ] [ 'string' ]  
      ON  
      OFF
```

Replays all current output buffer (up to 2000 lines) or just the output starting from the last occurrence of 'string'. The replayed output will pause for a carriage return key every nth (default 12th) line. To exit the replay, enter an EOF character (CntrlZ or CntrlD) at the pause.

The INPUT option displays only previously echoed input records. The program will then ask for a range of lines to re-execute. Just hit return for none.

Alternatively, the capturing of output in the replay buffer can be turned OFF or ON (the default).

2.7.6.28 Grabbing Numbers Or Strings From The Output Buffer -

```
--- $GRAB 'string' [ i j ] [ reg [ reg' ... ] ]  
    &GRAB
```

\$GRAB numbers or &GRAB words (blank delimited strings) from the output buffer. Scans back through the output buffer for the given search 'string' (delimited like a comment). It then displays or assigns to the given register, the jth number (or word) after the string or the jth number (or word) of the ith relative record. Successive numbers (or words) can also be assigned to any additional registers. The defaults are i=0 (found record) and j=1 (next number or word).

2.7.6.29 Clearing The Output Buffer -

```
--- $CLEAR
```

Clears the output buffer so that the scope of future \$GRAB and / or \$PLAY commands is limited to only the output that follows.

2.7.6.30 Iterating Through Input Records And A Set Of Registers -

```
--- $ITER      vary a b n [ vary' a' b' n' ... ] [ func [ func' ... ] [ 'title' ]  
              [ m ]      d              d'  
--- ...
```

The first format will iterate the next input record (or brace delimited block) while changing the internal register "vary" from "a" to "b" in "n" steps (up to 10000). Actually,

```

if n>0
    vary = a + (i-.5)*(b-a)/n          i=1,n
or if n<0
    vary = a + (i-1)*(b-a)/(|n|-1)    i=1, |n|

```

Up to 40 variables (levels of iteration) are permitted so that the next input record will be executed "n*n'*..." times. If specified, the resulting values in the register "func" (or the sum of the squares if more than one given) at the first 3 levels are continually written to a BRO binary distribution file called ITER.DIS or macro.DIS (with optional "title") for later processing, and one additional iteration will be done with the registers reset to their values at which the discrete value of "func" was minimum.

The second format will iterate the next input record either "m" times while changing the "vary"s randomly or in order to approach the actual minimum of the sum of the squared "func"s (up to 125). If "m" is specified, the "d"s are the probability distribution types, i.e.

$$\text{vary} = a*(1-x) + b*x \quad x=(1+\text{RAN}[d])/2$$

Otherwise if "m" is not specified, the "d"s are fractional derivative increments relative to the ranges "a b" that are used to build a change matrix that will be solved by a Singular Value Decomposition technique. Actually, double-sided derivatives are computed in order to approximate a damping factor from the non-linearity predicted by the homogeneous second derivatives. Therefore, the required number of iterations is $2*(\text{variables}+1)$, i.e.

```

initial state
vary - d*(b-a)/2    d < 1
vary + d*(b-a)/2
      :              d' < 1
      :
predicted min

```

Also, the number of "func"s should be greater than or equal to the number of "vary"s in order for it to find a unique solution. For nonlinear problems, successive \$ITER commands may be needed to reach the precise minimum.

Multiple record iteration loops must be enclosed in braces, i.e. the next record after the ITER and the first record of the block must start with an open brace "{" and the last must end with a closed brace "}".

```

--- $ITER vary a b n [ vary ' ... ] func [ func' ... ]
--- { ...
      :
---   ... }

```

Use &ITER instead of \$ITER to automatically cancel output during loop execution and restore it when completed.

2.7.6.31 Rapidly Evaluating Functions Of A Set Of Registers -

```

--- $EVAL      vary a b n [ vary' a' b' n' ...] func [ func' ...] [ 'title' ]
      [ m ]          d          d'

```

The first format will evaluate the given \$FCN functions while changing the internal register "vary" from "a" to "b" in "n" steps (up to 10000). Actually,

```

if n>0
    vary = a + (i-.5)*(b-a)/n          i=1,n
or if n<0
    vary = a + (i-1)*(b-a)/(|n|-1)    i=1,|n|

```

Up to 40 variables are permitted. The resulting values in the function "func" (or the sum of the squares if more than one given) at the first 3 levels are continually written to a BRO binary distribution file called EVAL.DIS or macro.DIS (with optional "title") for later processing, and one additional evaluation will be done with the registers reset to their values at which the discrete value of "func" was minimum.

The second format will iterate the next input record either "m" times while changing the "vary"s randomly or in order to approach the actual minimum of the sum of the squared "func"s (up to 125). If "m" is specified, the "d"s are the probability distribution types, i.e.

$$\text{vary} = a*(1-x) + b*x \quad x=(1+\text{RAN}[d])/2$$

Otherwise if "m" is not specified, "d"s are fractional derivative increments relative to the ranges "a b" that are used to build a change matrix that will be solved by a Singular Value Decomposition technique. Actually, double-sided derivatives are computed in order to approximate a damping factor from the

non-linearity predicted by the homogeneous second derivatives. Therefore, the required number of evaluations is $2 \times (\text{variables} + 1)$, i.e.

```
initial state
vary - d*(b-a)/2      d < 1
vary + d*(b-a)/2
      :                d' < 1
      :
predicted min
```

Also, the number of "func"s should be greater than or equal to the number of "vary"s in order for it to find a unique solution. For nonlinear problems, successive \$EVAL commands may be needed to reach the precise minimum.

2.7.6.32 Suppressing Any Date And/or Clock Time Outputs -

```
--- $DATIM [ ON  [ ON ] ]
          OFF  OFF
```

Turns ON or OFF any date and /or clock time outputs. When turned OFF, the system date and /or time routines return blank strings.

2.7.6.33 Beeping -

```
--- $BEEP [ n [ s ] ]
```

Causes the terminal or console to beep "n" times (default 1) with "s" seconds (default 0.00) between each beep.

2.7.6.34 On-line Editing Of Files, Macros And Portions Of Input File -

The user can use any editor available on his system to do on-line editing by specifying its command line with the EDITOR switch or environment variable.

```
--- $EDIT filename.ext
```

Edits the given text file specification (which must contain the period "."). If the file is already open, it is first closed, then edited, reopened, and finally positioned at the end of the file if desired.

```
--- $EDIT macroname
```

Edits the given macro (extracting it from the current library file if necessary). After exiting your editor, you will be asked if you want to replace the old macro definition with the new one. In any case, you will also be asked if you want to execute the macro immediately.

```
--- $EDIT [ 'string' ]
```

Edits the current input file (as specified on the last \$IO command) or only the part delimited by blank lines that contains the given 'string'. After exiting your editor, you will be asked if you want to replace the old part of the input file with the new one. In any case, you will also be asked if you want to execute all of the input file, only the edited part, or none of it. Note that obviously no application initialization is automatically done before re-executing the input file. It is up to the user to do this with the appropriate application commands.

2.7.6.35 On-line Processing Of Graphics -

```
--- $PLOT [ OFF ]  
      ASK  
      EACH  
      NORM
```

If no argument given, immediately processes all the plots in the current 2D plot file using the operating system command defined with the PLOTTER switch or environment variable. Otherwise,

OFF Suppresses screen graphics
ASK Will prompt at end of each screen for plotting and/or saving
EACH Will plot each screen without prompting
NORM Restores default (screen graphics and no prompting)

```
--- $VIEW  
&VIEW [ file ]
```

Immediately processes the system geometry in the current 3D vector file using the operating system command defined with the VIEWER switch or environment variable. If a 3D vector "file" name is given, then it is processed by either a new instance of the 3D viewer (\$) or it added to the contents of the current 3D viewer (&).

```
--- $DISP [ ON ]  
        OFF  
        file
```

Immediately displays (via the IMAGER switch or environment variable) the given binary distribution "file" (default BRO009.DAT or "file.DIS"). Otherwise, turns ON or OFF the automatic displaying of such files immediately after their creation.

2.7.6.36 Display Current Command States -

```
--- $SHOW  
&SHOW
```

Display the current states of several internal macro commands. The "&" version will also list the internal stack code (operators / operands) for any \$FCN definitions.

2.7.6.37 Pausing Output -

```
--- $PAGE [ n ]  
        OFF
```

Causes future text output to be paused every "n" lines (default 12), i.e. wait for the user to type a return to continue. Pausing can be turned OFF with this command or by typing an end-of-file (normally a control-Z) at a pause.

2.7.6.38 Setting The Default Directory Path -

```
--- $PATH [ "path" ]
```

Sets the default directory "path" for future file openings. If no path is given, then reverts to the previous one. "path" must include the closing directory character (\, / or] depending on the operating system). The default files (see \$IO command) are closed in the old directory, and then reopened in the new one.

2.7.6.39 Pausing Program Execution -

```
--- $WAIT [ s ] [ 'message' ]
```

Causes the program to wait "s" seconds (default 5) before continuing on. An optional "message" can also be displayed.

2.7.6.40 Local Macro Register/variables -

```
--- $LOC reg [ reg' reg" ... ]
```

Makes the given registers/variables names local to the current macro. In other words, on exit from the macro, the values of these registers will be automatically restored to their values previous to the \$LOC command. For example:

```
MY_MACRO {  
    $LOC A,B,C  
    A=#1 B=#2 C=#3  
    :  
}
```

Therefore, MY_MACRO will not (permanently) change the values of the registers A, B, and C when it is called.

2.7.6.41 Controlling The GUI Front-end -

```
--- $GUI command [ command' ... ]  
    &GUI
```

Sends the given command strings to the GUI front-end. The "\$" form sends the strings exactly as entered. The "&" form parses any expressions and replaces their results with string equivalents.

2.7.6.42 Controlling Uninitialized Variable Messages -

```
--- $UNVAR [ NONE ]  
    WARN  
    ERROR
```

Determines the type of message to be issued for future uninitialized (used before set) variables / registers. The default is WARN, i.e. a message but not a fatal ERROR. If no entry is given, the state before the last \$UNVAR command is restored.

2.7.6.43 Resetting The Random Operator Seed -

```
--- $RAN [ i ] [ DEC ]  
      BEST
```

Resets the random number seed for the "~" operator and RAN function to the integer "i" (default 2000000001). The default uniform random number generator (upon which all others are based) is identical to that used on the old DEC (Digital Equipment Corporation) systems. One can switch to the BEST possible generator based on the same algorithm but using different numerical constants and thus a different seed sequence.

2.7.6.44 Manually Aborting A Calculation -

```
--- $ABORT [ 'message' ]
```

From any macro/loop level, returns to the command prompt as soon as possible and displays the optional "message".

2.8 ERRORS IN INPUT

When an error is encountered during the decoding of a command, the program prints a message with an up arrow "^" pointing to the offending entry, e.g.

```
--- COMMAND entry entry' ....  
***                ^ERROR  
                   message
```

If there is no arrow in the message, then the error is in the number of entries (usually too few). In addition, an explanation of the proper syntax for the erroneous command is displayed. If an error occurs when commands are being read from a disk file, then the program does not execute any following commands but only checks for syntax. An erroneous command entered from the keyboard can be immediately corrected by the user and therefore, does not interfere with normal program execution.

INPUT LANGUAGE COMMANDS

INDEX

\$ABORT, 2-37	\$IF, 2-21
\$ARGS, 2-15	\$IF...THEN, 2-22
Arithmetic, 2-4	\$IO, 2-15
\$ASK, 2-24	\$ITER, 2-29
\$BEEP, 2-32	\$LEAVE, 2-19
	\$LOC, 2-35
\$CASE, 2-14	Math functions, 2-5
\$CLEAR, 2-29	\$MENU, 2-17
\$COPY, 2-17	\$MOVE, 2-17
\$DATIM, 2-32	\$NEXT, 2-19
\$DBG, 2-27	
\$DISP, 2-34	\$PAGE, 2-34
\$DO, 2-18	\$PATH, 2-35
	\$PLAY, 2-28
\$ECHO, 2-14	\$PLOT, 2-33
\$EDIT, 2-33	
\$ELSE, 2-22	\$RAN, 2-37
\$ELSEIF, 2-22	\$READ, 2-15
\$END IF, 2-22	\$REG, 2-24
\$ERR, 2-20	
\$EVAL, 2-31	\$SCR, 2-25
\$EXIT, 2-23	\$SHOW, 2-34
\$EXP, 2-14	\$STO \$RCL, 2-26
	\$SYS, 2-24
\$FAST, 2-26	
\$FCN, 2-28	\$TIC, 2-27
\$FF, 2-23	
\$GO, 2-20	\$UNVAR, 2-36
\$GRAB, 2-29	
\$GUI, 2-36	\$VIEW, 2-34
\$HELP, 2-23	\$WAIT, 2-35

CHAPTER 3

INPUT COMMANDS

3.1 INTRODUCTION

ASAP has been designed to accurately model virtually any optical / mechanical systems. Similar to 3D solids modeling programs, it utilizes a very powerful geometrical approach that permits a nearly limitless variety of systems to be handled in a straightforward manner. As opposed to most other ray tracing algorithms, all surfaces and ray data are referenced by default to a single global Cartesian coordinate system. Smooth continuous object surfaces can be represented by a sequence of simple conicoids or a general 286-term polynomial (taken to 10th or 20th order in the three Cartesian components or their squares) and can be bounded by other surfaces of the same general form. Therefore, anything from a simple plane with a polygonal boundary to an arbitrarily oriented elliptical toroid can be modeled precisely. Even more complex parametric mesh surfaces (NURBS) can be defined by connecting two or more arbitrary curves, each formed by a series of lines and / or conic arcs in space.

Bundles of rays can be traced through the system such that after every reflection or refraction each ray always transfers to the nearest object. Also, rays are allowed to interact with an object any number of times, i.e. multiple bounces. Therefore, the program easily handles the "funneling" effect of non-imaging radiation collectors as well as imaging systems. In addition, each ray may be assigned an arbitrary total flux that is reduced by volume and surface absorption (or increased in a gain medium) as the ray propagates through the system. The standard Fresnel equations are used to not only calculate (as a function of incidence angle and polarization) transmission

losses at interfaces between two dielectric media, but also reflection losses at any dielectric/conductor interfaces. The program is also capable of splitting any ray into reflected, transmitted, diffractive, near specular, diffuse, and back scatter components.

Each ray may also be treated as a coherent/incoherent scalar/vector beam (normally Gaussian). Groups of these beams can be combined to simulate the optical or electromagnetic field incident on a system. Each beam and thus the entire "wavefront" is then rigorously propagated. The resulting field may then be calculated and displayed at any location within the system (including on the surfaces of grazing incidence optics) and not just near focus. Therefore, ASAP cannot only perform geometrical optics modeling, but also accurate physical optics modeling of any system.

3.2 SUMMARY LISTING OF COMMANDS BY FUNCTION

Below is a concise listing of all the commands, grouped by function. The grouping is in the approximate order that one would use in a typical analysis. In addition, at the end of this chapter is an index listing each command alphabetically and the page number of its detailed explanation. Indented commands indicate that they must follow the appropriate unindented command. A slash between entries indicates only one of those entries at a time is allowed. Commas mean any combination of the entries. And a semi-colon indicates that these are actually separate commands that must be grouped together.

===== Define/modify entities or single entity objects=====

UPDATE	Control entity updating in storage
ENTITIES ,OBJECTS	Start streamlined entity/object input
:	

===== Define/modify curvedge entities =====

CURVES/EDGES	Begin defining curvedge(s)
POINTS	Enter line/arc/control points directly
LINE ,DASHED	Equally divided straight line
ELLIPSE	Elliptical (circular) polygon
RECTANGLE	Rectangular (square)

OVAL	Something between ellipse and rectangle
ROUNDED	Rectangle with crudely rounded corners
RACETRACK	Rectangle with precisely rounded corners
CHARACTER	Edge patterned after a character
ARC	Portion or all of a circle
CONIC	Quadratic segment given conic coefficients
SAWTOOTH	Shape and number of teeth
HELIX	General helical (coiled) curve
BEZIER	Explicit polynomial as same order Bezier
SPLINE	Curvature (G2) continuous cubic segments
USERCURVE	Parametric curve or surface using \$FCN
COMPOSITE	Combine set of previous edges into one
REPEAT	Repeat a previous edge definition
SAG	Sag the edge
ASCALE	Non-linear asymmetric scaling
ALTER X,Y,Z,Q	Alter specific edge point(s) data
SMOOTH	Smooth a piecewise linear curve
COARSEN	Opposite of SMOOTH
INVERT	Reverse curve's parametric direction
SWEEP POS/DIR/AXIS/OFF	Sweep curve into a surface
IMAGE	Image curve through a specified lens
PATCHES	Points represent Bezier surface patches
UVSPACE	Curve in parametric space of an object
EXTEND	Linearly extend one or both ends
MATRIX;ROTATE;SHIFT;SCALE;SKEW; PLACE;ALIGN;XEQ	Linear transformation of edge

===== Define/modify surfunc entities =====

SURFACES/FUNCTIONS	Begin defining surface function(s)
HORN	Horn given profiles
TUBE	Tube-like surface (cone,cylinder)
OPTICAL	Classic rotationally symmetric optic
BICONIC	Surface with two distinct conic profiles
TORUS	Torus (doughnut)
REVOLUTION ,FIT	Rotated 2D curve
AXICONIC	Rotated conic curve with foci
CARTOVAL	Cartesian Oval
CONDUIT	Circle swept along planar explicit cubic
SUPERCONIC	Special asphere used in optical design
ASYM	Distort an axially symmetric surface
ZERNIKE	Explicit Zernike polynomial surface

SAMPLED	Explicit surface interpolated from samples
GENERAL/COEFFICIENTS ,EXPLICIT	Enter surface coefficients directly
PLANE	Planar surface
ELLIPSOID	Orthogonal ellipsoid (sphere)
FITTED	Least squares fit to a set of points
USERSURF	User-programmable surface intersection
USERFUNC	User-programmable function
USERSAG	User-programmable radial or toric profile
CORNER	Axis-aligned corner of a cube.
REPEAT	Repeat a previous surface definition
FCN	Wrap macro function around surfunc
FMAP	Output a surface function map
MULTIPLE	Expand surface into multiple sheets
ARRAY EXPONENT, BOUNDS/SEARCH	Replicate surface into a spatial array
TEST OFF/POINT/DIRECTION/AXIS	Set test for particular branch
BEND	Bend surface in given direction
LOCAL	Localize surface in a "box"
RENORM	Renormalize surface coefficients
SOLID	Bounding volume formed with local box
PARAMETERIZE	Set local axis for meshing surface
ALTER	Alter specific polynomial coefficients
EXPLICIT	Convert to explicit form
MATRIX;ROTATE;SHIFT;SCALE;SKEW;	Linear transformation of surface
PLACE;ALIGN;XEQ	

===== Define/modify lens entities =====

LENSES	Begin defining lens entity(s)
SEQUENCE ,CURV/RADI	Arbitrary set of conicoid surfaces
MIRROR	Single reflecting surface
SINGLET FL/CV/RD,APLANAT	Two refractive surface lens
MANGIN FL/CV/RD	Second surface mirror
DOME	Refractive with one side a hemisphere
DOUBLET	Cemented (achromatic) doublet
TELESCOPE	One or two mirror telescope and corrector
RIGHT	Right-angle prism
PENTA	Penta-prism
WEDGE	Wedge of glass
AFOCAL	Afocal beam expander
COMPOSITE	Combine set of previous lenses into one
PERFECT	For object at infinity but realistic
IDEAL	Specify matrices of an ideal lens

REPEAT	Duplicate a previous lens
ALTER X,Y,Z,U,V,W,H,C/R,K,O,M	Alter specific lens conicoid(s) data
IMAGE	Image a global point through lens
ABERRATIONS ,LIST,PLOT	Display aberrations of centered lens
VARIABLES ,TH,CV,CC,BN,GL	Declare variables for optimizing lens
MINIMIZE ,DIST,TLEN,GLTH	Minimize RMS spot subject to constraints
STORE	Store in ASAP,CodeV,OSLO or ZEMAX formats
MATRIX;ROTATE;SHIFT;SCALE;SKEW; PLACE;ALIGN;XEQ	Linear transformation of lens

===== Create/modify media,coatings, scatter models =====

UNITS	Set system geometry length units
WAVELENGTH ,UNITS	Set MEDIA and COATING wavelengths
MEDIA	Begin defining media (glasses)
,ABSORB,GRIN,CRYSTAL,SCATTER/USER	Indices,absorption,gradient,birefringence
MODELS ,PLOT	Begin defining scatter models
LAMBERTIAN	Simple constant
HARVEY	Polished (smooth) surface
POLYNOMIAL/TRINOM/BINOM ,FIT	General polynomials with linear data fit
NONLINEAR ,FIT	Nonlinear in free parameters
VANES ,EDGES,DIFFRACT	Vane structure
USERBSDF	User-programmable
PARTICLES/VOLUME ,MIE	Random scattering centers or small spheres
VCAVITY	Rough (random v-cavities) surface
BSDFDATA/RAWDATA	Interpolates from entered values
RMS	Estimates scatter from surface statistics
PHYSICAL	Comprehensive physical reflective scatter
SUM	Sum of other models
COATINGS PROPERTIES/LAYERS/MODELS	Begin defining optical coatings

===== Create/modify objects =====

BRANCH	Set position in object name hierarchy
OBJECTS	Begin defining objects
REDEFINE SURF/NORM,THICK,COLOR	Redefine basic options
DEFORM ,AXIS,FCN	Add small user-programmable deformation
BOUNDS/RBOUNDS ,MULTIPLE,POINT	Complex bounding surfaces,curves,volumes
LIMITS AXIS/REPEAT/STATS/EXPAND	Simple orthogonal limiting box
FACETS	Set subdivision of patches into facets
MATRIX;ROTATE;SHIFT;SCALE;SKEW; PLACE;ALIGN;XEQ	Linear transformation of entire object

INTERFACE COATING ,DIFFRACT	Assign optical properties to object
ROUGHNESS ,RANDOM/MODEL	Affect specular by surface roughness
SCATTER MODELS/RMS/BSDF,RANDOM TOWARDS	Assign scattering characteristics Preferential random scattering
SPLIT	Specifies specular ray/beam splitting
LEVEL	Specifies scattered ray/beam level
FRESNEL	Flux variation with incidence angle
HALT	Sets conditions for halting trace
GROUP	Temporary collection of objects
MATRIX;ROTATE;SHIFT;SCALE;SKEW; PLACE;ALIGN;XEQ	Linear transformation of entire group
EXPLODE	Create separate objects from lens parts

===== Setup plots and verify system =====

TITLE	Specify user ID and/or default title
PIXELS ,ON/OFF,FILL	Sampling number and aspect
WINDOW	Define 2D graphics window on 3D data
OBLIQUE	Oblique (nonorthographic) views
VIEW ,CENTER,EYE,DOLLY,ZOOM,ORBIT	Perspective views
PRINT SURF/EDGE/LENS/COAT/MED/OBJ	Full display of system information
SPRINT SURF/EDGE/LENS/COAT/MED/OBJ	Shorter display of system information
QPRINT SURF/EDGE/LENS/COAT/MED/OBJ	Shortest display of system information
DIMENSIONS	Show dimensions of main arrays
NUMBERS ,NAMES/SUMMARY	List currently used entity numbers
PROFILES ,NOOPTIM,QUICK	Draw system profiles (slices)
PLOT	Plot geometry and/or ray data
RELOT ,OPTIM,NORAYS	Replot all 3D vector graphics
DRAWING ,DIMENSIONS,NORAYS	Draw four views of 3D vector graphics
VUFACETS ,LIST	Facets and views current objects
CONSIDER	Limit current set of objects
ARROWS	Set rescaling of arrows on plots
SEGMENTS	Number of segments per arc
SHOW ,ALL	Display current command settings
COLORS	Set colors by object's interface
LIGHTS	Specify light sources for render
RENDER ,DEPTH,RAYS,MODEL	Render current object surfaces
MAP ,DEPTH,SLOPES	Map current object surfaces
TREE ,ENTITIES	Display object name hierarchy

===== Standard plot options =====

...OVERLAY	Overlay next plot on top of this one
...COLORS	Override normal colors
...TEXT	Annotate plot with 2D or 3D text
...PIXELS	Temporarily override number of PIXELS
...XY/YZ/ZX/YX/ZY/XZ	Temporarily override current WINDOW

===== Setup beam creation =====

PARABASAL ,CLIP	Controls parabasal ray calculations
WIDTHS ,EDGE	Sets relative beam width factors
WAVELENGTH	Set operating wavelength
CLIP POS/DIR	Clip rays during creation and for POS:
BOUNDS/RBOUNDS ,MULTIPLE,POINT	Complex bounding surfaces,curves,volumes
USERAPOD POS/DIR/ANG/BOTH/OFF	User-programmable apodization
POLARIZ	Set polarization direction and values
XMEMORY FULL/NORM/MIN	Controls extended ray/beam paging
BEAMS INCOH/COHER,GEOM/DIFF,SHAPE	Sets future beam characteristics
SPECTRUM VIS/SCO/THERM/PHOT/FCN/OFF	Sets spectral weighting for future beams
IMMERSE	Sets starting medium for future beams

===== Create rays/beams =====

RAYSET	Begin defining a table of rays/beams
GRID REC/POL/ELL/OBJ/HEX/WIN/DAT	Define a spatial grid of rays/beams
SOURCE POSIT/DIREC/FOCUS/LINE/WAVE	Specify source(s) for the rays/beams
EMITTING	Composite of random emitters
EMIT DISK/RECT	Simulate a random emitting surface
EMIT CONE/PYRAMID/BOX/SPHEROID	Simulate a random emitting volume
EMIT FILAMENT/HELIX	Simulate a random emitting curve (wire)
EMIT RAYS	Arbitrary collection of rays
EMIT IES	Emitter specified by IES file data
EMIT ENTITY/OBJECT	Random emission from defined surface(s)
EMIT DATA	Emit according to distribution data file
GAUSSIAN	Rays for coherent astigmatic Gaussian mode
DECOMPOSE POSITION/DIRECTION	Decompose an existing coherent field

===== Modify ray/beam data =====

SELECT SOURCE/OBJECT	Select ray group for future operations
SUBSET ,RESET	Create subset of current rays/beams
FLUX	Assign total flux (power) in rays/beams
SHAPE	Specify shape and coherence of beams

REVERSE	Reverse current ray/beam directions
MOVE BY/TO, POINT/PLANE/SPHERE/FOCI	Move to new positions on rays/beams
BILATERAL	Flip ray/beam data about a plane
IMAGE	Image ray points through a specified lens
APODIZE POS/DIR/ANG/BOTH	User-programmable apodization
MATRIX; ROTATE; SHIFT; SCALE; SKEW; PLACE; ALIGN; XEQ	Linear transformation of ray data

===== Setup trace =====

WARNINGS	Controls warning messages
AXIS	Local and/or cylindrical coordinates
SEED , QUASI	Initializes random number seed
SEARCH ALL/SEQUENTIAL/LIST	Object intersection search pattern
ALLOWED ALL/SEQUENTIAL/LIST	Object intersection halt pattern
HALT	Sets conditions for halting trace
CUTOFF	Set absolute flux and number threshold
SAVE , file#	Saves all intersection data to file
SPLIT	Specifies specular ray/beam splitting
LEVEL	Specifies scattered ray/beam level
FRESNEL	Flux variation with incidence angle
MISSED ARROWS/LINE/OFF	Controls missed ray plotting
ACCURACY HIGH/MEDIUM/LOW	Accuracy of ray/surface intersections
VOXELS FL/AB/X/Y/Z/OFF	Setup volume energy tracking during trace

===== Trace ray/beams =====

RAY , PLOT, DIR, SEARCH, GALOP, LENS	Trace a single ray
TRACE , PLOT/GRA, LIST/DIR, STATS, STEP	Trace current set of rays/beams

===== Analyze ray/beam data =====

LIST POS/DIR/RAY/SOU/INT/ELL	List specific data for all or some rays
EXTREMES POS/DIR/FLUX/LEN	List data on extreme rays
STATS POSITION/DIRECTION	Statistics of ray/beam data by object
PATHS AVE/PEAK/TOT/OBJ	Grouping and listing by ray path
HISTORY , PLOT/SPOTS/LIST	Display histories of rays in SAVE file
SPOTS POSITION/DIRECTION	Distribution of current ray data
FOCUS , MODE, MOVE	Find (and move to) best focus
PLOT BEAMS, RAYS, POLARIZATION	Plot beams, rays or polarization
FFAD , SPOTS, REFERENCE	Full Field Aberration (Spot) Display
OPDMAP file	Interpolate OPDs into file for DISPLAY

RADIANT ,MAP/AREA	Radiant pattern in spherical coordinates
DUMP	Currently selected rays to binary file
COLLECTION	Efficiency vs. aperture and cone angle

===== Modify or use internal ray/beam data as input =====

GET	Load ray data into input registers
PUT	Move values in registers back

===== Calculate diffraction/propagation effects =====

IRRADIANCE	Set irradiance definition direction
VIOLATION	Controls paraxial/positivity/stability
SPREAD DIR/POS/APROX/NOR ,DOWN,CLIP	Specular/scattered energy distributions
FIELDSUM ,DELT,ADD,MULT,COUPLE,CONT	Coherent beam summation
FIELDBPM ,DELT,MULT,COUPLE,CONT	Finite-difference beam propagation

===== Display/modify data distributions =====

DISPLAY	Read distribution data, modify/display
NORMALIZE	Normalize by constant or maximum
FORM	Set to power or logarithmic form
FFT ,SIZE	Fourier Transform data
ABEL ,INVERSE	Abel transform data
AVERAGE ,WEIGHT	Average data over several pixels
RADIAL ,FUNC/INTEG/BOTH	Radially average about a point
TRANSPOSE	Transpose distribution array
MODIFY	Modify regions of distribution
COMBINE	Multiply or add another data file
REDUCE	Reduce to a smaller subset of data
WRITE	Write current data to ASAP file(s)
DMAP	Print a map of data
RANGE	Override min/max for data plotting
THRESHOLD	Reset floor and/or ceiling data values
PLOT3D	Plot in 3D of data plus major profiles
ISOMETRIC	Isometric plot
GRAPH ,APPEND	Plot 1D profiles of data
CONTOUR ,LOW/HIGH,TICS,VECTOR	Contour or "color" map plot
DIRECTIONAL ,UNWRAP,RADIANCE	Polar plot of angular distribution
MESH	Write distribution to 3D system file
ENCLOSED	Plot percent enclosed function
SECTION	Display or transfer a section of data

TABLE	Display a table of numeric values
ANGLES ,RADIANCE	Convert direction cosines to angles
VALUES	List value(s) at actual coordinates
OFFSET	Shifts coordinate origin
FOLD ,FIRST/SECOND/BOTH	Average data about one or both centers
PICTURE	Produce grey-scale picture
IESFILE	Writes IES file of angular distribution
TEXTFILE	User-definable text file of distribution
DATA	Write actual data in specified format
EOF	End of text file
HEADER	Redefine header (e.g. labeling)
REPLICATE	Replicate distribution one or more times
APPEND	Append current data to given file
HISTOGRAM	Plot of data value distribution

===== Save or recover system data and control execution =====

SYSTEM NEW/TO/FROM file	Read/write system data to binary file
RESET	Reinitialize all settings
DOMACROS FIRST/LAST/NEVER	Transparent macro execution control
LSQFIT ,NORM/OFF,LIST	Controls SVD fitting algorithm
FTSIZE	Sets Fourier Transform size
CADEXPORT IGS/DXF/VCR	Export object surfaces to CAD file
RETURN	Return to previous command level
END ,OFF	End execution or ignore future ENDS

In addition to these primary commands, the user can define his or her own commands via the macro input facility described in the last chapter. By default (see DOMACROS command), ASAP first checks the input keyword against the primary command list. If it does not find a match, it then searches the current macros and library file (if it exists). If a match is found, the macro is expanded in the normal fashion except that the "\$" or "&" prefix on the macro name is not needed. The program does not echo the internal macro lines and therefore, these new commands are indistinguishable from normal program commands. This capability does not alter the normal macro expansion when the macro name prefix is present so that a macro can still be executed (and echoed) even if its name conflicts with a primary command.

3.3 GENERAL INPUT TECHNIQUES

3.3.1 Radian And Direction Cosine Angle Entries

Angle entries that by default are in radians (i.e. the PARABASAL divergence, ROUGHNESS RANDOM slope, and INTERFACE RMS/BSDF back cone angle) can also be entered in degrees by appending a "D" to the end of the number. For example, the following two entries are equivalent:

```
2.5D <==> 2.5/57.29578 radians
```

If the entry is a direction cosine coordinate (CLIP DIR or SPOTS/SPREAD DIR window), then the result is the sine of the angle, e.g.

```
2.5D <==> SIN(2.5/57.29578)
```

3.3.2 Relative And Literal Referencing Of Entities

Alternate schemes are available when referencing surface/edge/lens, media, object, ray or source numbers within commands. Instead of using the actual absolute number, one can specify a number relative to the largest number defined by using a decimal entry of the form ".i" where i is an integer between 1 and 9999 inclusive. ".i" is equivalent the 1 plus largest number defined, minus i.

REFERENCING METHOD	FORMAT	DESCRIPTION
Absolute	n	actual entity number
Relative	.i	max+1-i
Literal	name	exact or abbreviation

Some examples of relative indexing are:

```
.2 ==> next to the largest number defined so far  
-.1 ==> negative of the largest number defined
```


In addition, for most commands that require the specification of a particular object, media or coating, the name can be used in place of directly specifying the number or using the relative indexing described above. The program first attempts an exact match (ignoring blanks). Otherwise, the characters in an abbreviation must be present in the same order as in original object, media or coating name but not necessarily consecutive, i.e. any number of original characters can be skipped in order to make the abbreviation small enough but unique. An underscore "_" in a literal media reference always requires an exact match and is also used to separate the catalog name (file with extension CAT) from the glass name, e.g. "SCHOTT_BK7".

3.3.3 Direction Vectors

```
--- ... a,b,c ...
```

Some commands (e.g. PLANE NORMAL, ALIGN, RAY, SOURCE DIRECTION) require the specification of a direction vector. The following formats can be used for these three input entries (shown as simply "a,b,c" in the command descriptions).

THE 3 ENTRIES	DESCRIPTION
-----+-----+-----	-----+-----+-----
a b c	Actual relative components of direction vector
-----+-----+-----	-----+-----+-----
+/- b c	Two actual direction cosines, sign of third
a +/- c	
a b +/-	
-----+-----+-----	-----+-----+-----
X a a'	Spherical angles (in degrees) from and around given axis
Y	
Z	
-----+-----+-----	-----+-----+-----
DIR RAY n	Current direction of given ray number
SOU	Average direction of rays from given source
-----+-----+-----	-----+-----+-----
OBJ n n'	Direction of vector between positions of the two entities
SUR	
EDG	
LEN	
RAY	
-----+-----+-----	-----+-----+-----
NOR SUR n	Direction of normal to given entity at its reference point
LEN	
EDG	
OBJ	
LIM	at LIMITS box center
-----+-----+-----	-----+-----+-----
LOC X n	Direction of given local coordinate axis of SURFACE n
Y	
Z	
-----+-----+-----	-----+-----+-----
TAN n u	Tangent to CURVE/EDGE n at parametric value u

The resulting vector is automatically normalized to unit length by the program.

3.3.4 Linear Transformations

Linear transformations change the scaling or orientation of a geometrical

entity (surface, edge, lens, object, group, rays) by applying a general 4-by-4 linear transformation matrix to it. Any number of the following elementary operations can be applied after any entity definition to build up the final matrix. The order in which these operations are entered into the input stream is exactly the order in which they will be applied to the entity. These commands must be grouped together following an entity definition with no other commands between them (except for a comment). The LIST option causes the resulting 4-by-4 transformation matrix to be printed and decoded into simple operations if possible. A general transformation for a position vector (X,Y,Z) has the form:

$$\begin{array}{|c|c|c|c|c|c|} \hline 1 & & & & & \\ \hline X' & & D_x & A_{xx} & A_{xy} & A_{xz} \\ \hline Y' & = & D_y & A_{yx} & A_{yy} & A_{yz} \\ \hline Z' & & D_z & A_{zx} & A_{zy} & A_{zz} \\ \hline \end{array} \begin{array}{|c|} \hline 1 \\ \hline X \\ \hline Y \\ \hline Z \\ \hline \end{array}$$

The A's are the rotation submatrix while the D's are the translation vector. The first row is a dummy used to make the matrix square (and thus invertible). For the transformation of a direction vector the dummy row contains all zeros.

3.3.4.1 Translation Operations -

```
--- SHIFT [ x y z ] [ LIST ]
      X x
      Y y
      Z z
--- SHIFT d ALONG a,b,c [ LIST ]
```

Translates the entity BY the given distances (x,y,z). The default distances move the entity's reference point to the global origin. To shift a distance "d" along a given direction "a,b,c", use the last format.

```
--- PLACE [ x y z ] [ LIST ]
      X x
      Y y
      Z z
```

Translates the entity TO the point (x,y,z). The following form determines this point from the fractional distance "d" (default 0) from a given entity "k" to

another point or entity "k".

```

--- PLACE AT k [ d k' ] [ LIST ]
      SUR      x y z
      EDG
      LEN
      OBJ
      RAY

```

The AT refers to the same entity type as the entity being transformed.

```

--- PLACE CUR k u

```

Places the entity at the global position that corresponds to parametric coordinate "u" of CURve / edge "k".

3.3.4.2 Distorting Operations -

```

--- SCALE a [ b c ] [ LIST ]
      X a [ x ]
      Y b   y
      Z c   z

```

Scales the entity by the uniform factor "a" or nonuniformly by separate coordinate factors (a,b,c) about the optional point (x,y,z). This point is defaulted to the entity's reference point.

```

--- SCALE FROM units [ LIST ]

```

Scales from the given length "units" to the current system UNITS (see this command for allowable "units" strings). If the system UNITS have not be set, they are set to "units" and no scaling takes place.

```

--- SKEW X a Y [ c ] [ LIST ]
      Z
      Y a Z
      X
      Z a X
      Y

```

Skews the entity in the first direction by an angle "a" measured in degrees from the second direction. The corresponding coordinate of the skew center is optionally given by "c". The skew center is defaulted to the entity's reference coordinate.

3.3.4.3 Rotation Operations -

```
--- ROTATE X d [ y z ] [ LIST ]
           Y d   z x
           Z d   x y
--- ROTATE d ABOUT a,b,c [ x y z ] [ LIST ]
```

Rotates the entity through an angle "d" (in degrees) about an axis displaced by the fourth and fifth entries from the coordinate axis given by the second entry. To rotate about an arbitrarily oriented axis direction "a,b,c", use the last form. The sign of the angle is determined by the right-hand rule where a positive value results when the thumb points along the positive direction of the rotation axis. The center of rotation is defaulted to the entity's reference point.

```
--- ALIGN a,b,c a',b',c' [ x,y,z ] [ LIST ]
```

Performs the most direct coordinate rotation necessary to align the given direction vector with another general direction. The rotation axis passes through the given point, i.e. this point remains fixed during the transformation. This point is defaulted to the entity's reference point.

3.3.4.4 General Operations -

```
--- MATRIX
--- x  a  b  c
--- y  a' b' c'
--- z  a" b" c" [ LIST ]
```

Allows the user to specify a 3X4 (first dummy row excluded) transformation

matrix directly. Another form of the matrix command recalls either the PREVIOUSly used matrix, its INVERSE or the matrix of SURFACE entity "k" (use zero for inverse of the SURFACE entity currently being transformed).

```
--- MATRIX [ PREVIOUS ] [ LIST ]  
          INVERSE  
          k
```

```
--- XEQ [ LIST ]
```

Forces the program to immediately apply the current transformation matrix to the entity instead of waiting for a non-transformation command to signal the end to the transformation command set. The transformation matrix is then reinitialized to the identity matrix and the program begins looking for a new set of transformation commands.

3.4 DEFINING THE GEOMETRY OF THE SYSTEM

The program uses a single global Cartesian coordinate system with components x,y, and z. "Edges", "lenses", and "surfaces" are the basic geometrical entities of the program. Edges (or curves) are made up of a series of connected points in space. Lenses are arbitrary sequences of circular conicoids. Surfaces are represented mathematically by a single equation $f(x,y,z)=0$ and are either infinite in extent or self-enclosed (e.g. a plane or sphere, respectively). The volumes of space between surfaces can be assigned refractive properties and are called "media". An "object" is a lens, several surfaces or many connected curves bounded by other surfaces and/or edges, and represents something that can absorb, refract, reflect, diffract, polarize, or scatter radiation.

```
*****  
* Unless otherwise stated, ALL COORDINATES ARE GLOBAL. *  
*****
```

3.4.1 Control Of Entity Updating In Storage

```
--- UPDATE [ OFF ]
```

Controls how basic entity data (i.e. SURFACE coefficients, LENS conicoids or EDGE points) is updated in storage. Normally, when an already defined entity is updated at a later time, then the new data overwrites the old. This works fine as long as the amount of new data is less than or equal to the old. If this is not the case, then the data for the next entity can be corrupted. To prevent this, use an UPDATE command before making entities "larger". This tells the program to place all future entity data at the end of storage. The old data is not deleted but is never used again. To return to the default overwrite mode, use the OFF option.

3.4.2 Curve Or Edge Entities

```
--- EDGES    [ i ]
    CURVES
    ENTITIES
```

Begin defining EDGES starting with number "i" (defaults to one plus highest). EDGE, LENS, and SURFACE data currently reside in the same internal storage locations. Therefore, an EDGE number cannot be the same as an already defined LENS or SURFACE number.

3.4.2.1 Edges With No Assumed Symmetry -

```
--- POINTS x y z q [ x' y' z' q'... ]
                ABS
                REL
--- [ x" y" z" q" ... ]
--- :
```

Enter the coordinates of the edge directly. The "q" parameters control how

each edge point is connected to the next, i.e.

q=0 Not connected to the next point (open)
q=1 Connected by straight line to the next point
q=2 Connected by a quadratic rational Bezier (conic) curve to the point after next. The next point is the intermediate control vertex. The next q is the intermediate weighting factor and is always positive:
 0 straight line
 <1 ellipse
 1 parabola
 >1 hyperbola
q=-2 Connected by an elliptical arc to the point after next. The next point is the center of the parent ellipse. The next q is the angular subtense (in degrees) of the arc and must be less than 180.
q=n Connected by an nth (up to 20th) degree rational Bezier curve. The intermediate n-1 points are control vertices with given (positive) weight factors.

A non-zero "q" for the last point tells the program to connect this point with the first, i.e. a closed curve. The first point entered always becomes the edge's reference point and is normally also the first point on the actual edge. However, if the first "q" entry is a literal, then the second point entered is the first point on the edge, and it (and all other points) are measured either ABSolutely or RELatively from the reference point.

```
--- REPEAT [ i ]
```

Repeat the "i"th or last edge description so that it can be modified to form a different edge.

3.4.2.1.1 An Equally Divided Straight Line -

```
--- LINE x y z x' y' z' [ n ] [ DASHED ]
```

Creates a straight line from the first point to the second that is equally divided into "n" connected segments. The default is 1 or the value specified on a previous LINE command (use "-n" if you do want it to become the default for future LINE commands). If the DASHED option is present, then every other segment of the line is open (not connected).

3.4.2.1.2 General Spline From Points And Directions -

```

--- SPLINE [ x y z ] [ CURV ]
---   x  y  z  [ a, b, c ] [ k ]
---   x' y' z' [ a',b',c' ] [ k' ]
--- [ x" y" z" [ a",b" c" ] [ k" ] ]
---           :
```

General 3D, curvature (G2) continuous, piecewise cubic, spline curve with given reference point (default origin) and defined in terms of subsequent relative points and tangent directions (and optional inverse radii of CURVatures). If a tangent direction is not provided for any point, one is calculated by finite differences using the adjacent points. Therefore, this command can also be used as an (often better) alternative to POINTS followed by SMOOTH 3 (although it does create a curve with 3 times as many Bezier points).

3.4.2.1.3 Curve Or Surface Net From Users Parametric Function -

```

--- USERCURVE fcn u u' m [ v v' n ]
                        SPLINE
                        CURV
```

Normally, creates a piecewise linear curve (or bilinear surface) from a \$FCN function "fcn". The last three expressions of this \$FCN definition are the X, Y, and Z coordinates of a point as function of the parameter register "_" or "_1" (and optionally "_2" for a surface). For a curve, the parameter will range from "u" to "u'" in "m" segments. For a surface, the second parameter will range from "v" to "v'" in "n" segments. For example, a wiggly "circle" definition would be:

```

PI=ACOS(-1)
$FCN SQUIGGLE A=2*PI*_ R=1+.1*SIN(10*A),
      x R*COS(A) y R*SIN(A) z 0
EDGE
  USERCURVE SQUIGGLE 0 1 100
  SMOOTH

```

or a bilinear approximation to a "sphere" surface:

```

$FCN SPHERE R=1 P=_1*PI/180 A=_2*PI/180,
      x R*COS(A)*SIN(P) y R*SIN(A)*SIN(P) z R*COS(P)
EDGE
  USERCURVE SPHERE 0 180 6 0 360 6

```

Optionally, the derivatives of the \$FCN function can be used to smooth the curve into a curvature (G2) continuous, piecewise cubic, non-rational SPLINE or rational CURVe.

3.4.2.2 Polygonal Edges Initially Normal To A Particular Coordinate Axis -

These edges are made up of coplanar straight line segments, i.e. convex polygons whose vertices lie on a particular curve.

3.4.2.2.1 An Ellipse (or Circle) -

```

--- ELLIPSE X x y z [ n a a' ]
      Y y z x
      Z z x y

```

Defines an elliptical polygon normal to the given axis (second entry) at the specified location (third entry). The fourth and fifth entries are the semi-major widths of the ellipse. "n" is the total number of points or segments on the ellipse. The default is 16 or the value specified on a previous ELLIPSE command (use "-n" if you do want it to become the default for future ELLIPSE commands). "a a'" set the effective angular range (in degrees from first semi-major axis) over which the ellipse is defined (default is 0 to

360 degrees). They are only actual angles when they are multiples of 90 degrees or the aspect ratio of the figure is unity (i.e. the ellipse becomes a circle). In the Z axis case, the effective and actual angles are related by the following formula:

$$x \tan(\text{actual}) = y \tan(\text{effective})$$

Once the last three optional entries in brackets are specified, they become the default settings for most future EDGE commands.

3.4.2.2.2 A Rectangle (or Square) -

```
--- RECTANGLE X x y z [ n a a' ]  
               Y y z x  
               Z z x y
```

Defines a rectangular polygon. See ELLIPSE for description of parameters. When defining a closed rectangle (0-360 degrees), "n" should be a multiple of 4 so that there are edge points at each corner. The default is 4 or the value specified on a previous RECTANGLE command (use "-n" if you do want it to become the default for future RECTANGLE commands).

3.4.2.2.3 A General Oval -

```
--- OVAL X x y z q [ n a a' ]  
          Y y z x  
          Z z x y
```

Defines a polygonal edge that can continuously vary between an ellipse (q=0) and a rectangle (q=1). See ELLIPSE for description of other parameters.

3.4.2.2.4 A Rectangle With Crudely Rounded Corners -

```

---  ROUNDED X x y z r [ n a a' ]
          Y y z x
          Z z x y

```

Defines a rectangular polygon with "rounded" corners of radius "r". See ELLIPSE for description of other parameters.

3.4.2.3 Polycurves Initially Normal To A Particular Coordinate Axis -

These edges consist of coplanar straight line and / or higher-order curved segments. Initially, the reference point is the specified axial location and therefore not necessarily on the edge.

```

---  POINTS X x   y z   q [ y' z' q' ... ]
          Y y   z x       z' x'
          Z z   x y       x' y'
---    :

```

A general edge in the plane defined by the third and fourth entries. See the full 3d POINTS command for further details (excluding reference point).

3.4.2.3.1 A Rectangle With Precisely Rounded Corners (racetrack) -

```

---  RACETRACK X x y z y' [ z' ] [ SPLIT ]
          Y y z x z'   x'
          Z z x y x'   y'

```

In the plane designated by the second and third entries, a racetrack edge (of overall semi-major widths given by the fourth and fifth entries) is created with straight sides and elliptical corners of corresponding semi-major widths given by the sixth and seventh entries. Therefore, if the overall semi-widths are equal to the corner semi-widths, the racetrack has no straightaways, i.e. it is a pure ellipse. Conversely, if the corner semi-widths are zero, the racetrack has no elliptical turns, i.e. it is a pure rectangle. For

compatibility with older versions, the straightaways can be SPLIT into two equal line segments.

3.4.2.3.2 An Edge Patterned After A Text Character -

```
--- CHARACTER X x y [ z ] string
              Y y z   x   'string'
              Z z x   y
```

Creates an edge patterned after the character "string" in the plane given by the second and third entries. The fourth entry is the width of the character and the fifth is the height (defaulted to the width).

3.4.2.3.3 A Circular Arc -

```
--- ARC X x y [ z y' z' [ a ] ]
        Y y z   x z' x'
        Z z x   y x' y'
```

Defines a circular arc in the plane given by the second and third entries, starting at the unprimed point and rotating about the primed center point (default 0,0) through an angle "a" (default 360 degrees). For example, using the defaults allows one to define a complete circle of radius "r" centered on the axis with the following shortened versions.

```
--- ARC X x r
        Y y
        Z z
```

Arcs up to 135 degrees are represented by a single quadratic rational Bezier segment. Beyond that, the arc is subdivided into more of these segments (up to 4 for a full 360 degrees).

3.4.2.3.4 Quadratic Segment From Implicit Coefficients Of Conic -

```

--- CONIC X x  c y z y2 yz z2  y' z' y" z"
                                     Y y' y" [ p ]
                                     Z z' z"
      y  c z x z2 zx x2  z' x' z" x"
                                     Z z' z" [ p ]
                                     X x' x"
      z  c x y x2 xy y2  x' y' x" y"
                                     X x' x" [ p ]
                                     Y y' y"

```

Creates a single quadratic rational Bezier segment in the plane given by the 2nd and 3rd entries. Entries 4 through 9 are the implicit coefficients of this conic curve. Entries 10 and 11 are the coordinates in this plane of the start point of the segment, 12 and 13 the stop point. Instead of giving a pair of coordinates for each point, only one coordinate value for each point can be given and the program will solve for the others (smallest root). In this case, "p" is an optional vertex radius of curvature of a parabola to subtract from the curve.

For example, to define the equivalent of a classical optical conicoid:

```

CURVE
R=1 !vertex radius of curvature
K=-1 !conic constant
H=1 !outer aperture height
O=0 !inner obscuration height
CONIC X 0, 0 0 -2*R 1 0 K+1, Y (0) (H)
SWEEP AXIS 360 Z
OBJECT
INTERFACE ...

```

3.4.2.3.5 Sawtooth Pattern Edge -

```

--- SAWTOOTH X x y z y' z' n [ w ]
      Y y z x z' x'
      Z z x y x' y'

```

Creates a sawtooth pattern edge in the plane give by second and third entries. The edge starts at the origin in that plane and the first tip (or control point) is located at the unprimed fourth and fifth entries. The first tooth ends at the primed sixth and seventh entries. There will be "n" teeth altogether. Instead of sharp pointed teeth, a positive Bezier weight factor "w" can be used to get rounded conic teeth (see POINTS command for more information).

3.4.2.3.6 Explicit Polynomial As Same-order Bezier -

```
--- BEZIER X x y y' c [ c' [ c" ... ] ]
          Y y z z'
          Z z x x'
```

Creates a single Bezier segment in the plane given by the second and third entries. The range of the segment in that plane is given by the fourth and fifth entries. The order of the Bezier segment is the same as the order of the explicit polynomial given by the remaining coefficients and must not exceed 20. For a segment in the Z=z plane, the exact equation of the curve is:

$$Y = c + c'X + c''X^2 + \dots \quad x < X < x'$$

3.4.2.3.7 Explicit Spline From Coordinates And Slopes -

```
--- SPLINE X [ x ]
          Y y
          Z z
--- h s d [ c ]
--- h' s' [ d' ]
--- [ h" s" [ d" ] ]
--- :
```

Explicit 2D, curvature (G2) continuous, piecewise cubic, spline curve in the given plane (default 0). Defined in terms of subsequent heights (next coordinate), slopes, optional distances (remaining coordinate), and starting

curvature "c" (inverse radius).

3.4.2.4 General Helical Curve -

```
--- HELIX X x y z x' y' z' t [ n ]
          Y y z x y' z' x'
          Z z x y z' x' y'
```

Creates a general helical curve starting at the plane whose axial location is given by the third entry to the plane given by the sixth entry. If these two axial locations are the same, then the helix becomes a simple planar spiral. The elliptical semi-heights at the first plane are entries four and five, seven and eight for the second plane. The helix will consist of "t" turns (not necessarily integer). By default, the helix is perfectly smooth. Optionally, it can consist of "n" linear segments.

3.4.2.5 Combining A Set Of Previous Edges Into One -

```
--- COMPOSITE [ -n ] [ GAPS q [ q' ] ]
              i [ i' ]
```

Combine the last "n" edges, edge "i" through "i'" or all defined since EDGE command into one edge. GAPS between the endpoints of one edge and the start of the next can be handled as follows:

q	action
---	-----+-----
-1	merge gap points into one
0	leave gaps open (default)
1	linearly connect gaps

Specify "q'" for the last-to-first point gap.

3.4.2.6 Distorting Edges -

```

--- SAG [ X ] i [ ABS ]
      Y      REL
      Z [ CV c [ c' ] ] [ CC k [ k' ] ] [ AD d [ d' ] ] [ VP p [ p' ] ]
      RD r  r'
      SPHERE r [ q q' [ p p' ] ]

```

Deforms the last EDGE specification by sagging the points on the edge in the direction given by the second entry or the original normal axis of the edge. The actual sag can be specified in three different ways.

First, the edge can be sagged to the previously-defined "i"th SURFACE. This can be done ABSolutely (the default) for all points or RELatively for any interior control points. The latter will retain the shapes of individual segments of the edge.

Second, the sag can be specified independently via a standard optical sag formula. "c", "r", "d", "k", and "p" specify the vertex curvature, vertex radius, fourth order deformation coefficient, conic constant, and vertex offset respectively of the first semi-major axis component. The primed quantities refer to the second axis and are defaulted to the same values as the first.

And third, all the edge points can be exactly sagged to a SPHERE. The center of the sphere is located a distance "r" from the edge's reference point (or the given vertex point "p p'") along the given coordinate axis. The mapping of the other two coordinate directions to the surface of the sphere is controlled by the "q" and "q'" parameters. A value of zero performs a "latitude" mapping and one a "longitude". The parameters can be varied continuously between these cases.

```

--- ASCALE s s'

```

Scales the previous edge along the axis that follows the normal axis of the edge, e.g. an ELLIPSE Z will be scaled in the x direction. The scaling varies linearly from "s" at the bottom to "s'" at the top of the edge (as measured along the third coordinate direction).

For Z axis example: $x' = (ay+b) x$

3.4.2.7 Altering Specific Data On A Set Of Edge Points -

```
--- ALTER n [ n' [ n" ] ] data a [ b [ j ] ] data' a' [ b' [ j' ] ] ...
```

Alters the given data on points "n" through "n'" (default n) for the current edge, i.e. for i = n to n' by n" (default 1):

$$\begin{aligned} \text{data}(i) &= a + b \text{ data}(j) && [\text{defaults: } b=0, j=i] \\ \text{data}'(i) &= a' + b' \text{data}'(j') \\ &: \end{aligned}$$

where

"data"	Description
-----+-----	
X	x coordinate of edge point
Y	y
Z	z
Q W	Connection or weighting factor

3.4.2.8 Reversing The Parametric "direction" Of A Curve -

```
--- INVERT
```

Reverses the parametric direction of a curve, i.e. spatially it remains the same curve but the last point becomes the first and vice-versa. Useful when connecting curves into a parametric mesh OBJECT.

3.4.2.9 Smoothing A Piecewise Linear Curve -

```
--- SMOOTH [ 2 ]
           3
```

Quadratically (2, the default) or cubically (3) smooths the current curve assuming it is piecewise linear (e.g. one created by ELLIPSE, ROUNDED, or OVAL).

3.4.2.10 Coarsening A Curve -

```
--- COARSEN [ m ]
```

Does the opposite of the SMOOTH command by converting the current edge to a piecewise linear curve using every "m"th (default 1) point.

3.4.2.11 Specifying The Optional Sweeping Of A Curve Into A Surface -

```
--- SWEEP DIR d [ a,b,c ]  
          TO   x y z  
          POS f [ x y z ]  
          AXIS t   a,b,c [ x y z ]
```

If POSition, sweep edge a fractional distance "f" towards point "x y z" (default is centroid of edge points). If DIREction, sweep an actual distance "d" along direction "a,b,c" (default is normal to best fit plane through edge points) or sweep first point on edge TO "x y z". If AXIS, sweep the edge an angle "t" in degrees around the axis specified by the given direction and point. This information is used to define the surface of a single edge object or a BOUNDS edge. The sweep can be turned OFF or modified with the following shortened version:

```
--- SWEEP [ OFF ]  
          d  
          f  
          t
```

3.4.2.12 Imaging A Curve/edge Through A Lens -

```
--- IMAGE l k k'
```

Images the points of the current curve / edge from space "k" to space "k'" of LENS entity "l". The spaces are numbered from 1 (before the first conicoid) to N+1 (after the last conicoid N). It uses the auxiliary axis technique that

steps the image through the centers of curvature of the conicoids (the aperture semi-diameters, conic constants and obscuration ratios of the conicoids do not affect the imaging) and always produces an image even if it is virtual or unphysical. The resulting imaging transform is stigmatic (points go into points) but not necessarily collinear (lines go into lines). Therefore, it is only an approximation since in any real optical system, the image will be aberrated (not a perfect point focus). This is most useful for SCATTER TOWARDS edges that represent images of important areas.

3.4.2.13 Rational Bezier Surface Patches -

```
--- PATCHES k l [ m n [ SEPARATE ] ]
```

Make current curve/edge points a surface mesh of Bezier degree "k" by "l" (both not to exceed 20) with "m" by "n" continuously joined or SEPARATE rational patches. The total number of points must be exactly equal to:

$$\begin{array}{ll} (1+k*m)*(1+l*n) & \text{continuous} \\ (1+k)*(1+l)*m*n & \text{SEPARATE} \end{array}$$

The connection "q" factors for each point will be interpreted as actual rational Bezier weights "w".

3.4.2.14 Curves Defined In The Parametric Space Of An Object -

```
--- UVSPACE [ OFF ]
      i
```

Assume current edge is defined in the UV parametric space of object "i" (or the first object that uses it). Currently, this is only useful if it is used as a closed BOUND on a CURVE-based object. Alternatively, any planar edge can be directly defined in parametric space by substituting "UV" for the second (axis) and third (location) entries on its command. For example,

```
POINTS Z z ... ==> POINTS UV ...
```

The OFF option returns the curve definition to the Z=0 plane in global three-dimensional space so that it, for example, can be PLOTted.

3.4.2.15 Linearly Extend One Or Both Ends Of A Curve -

```
--- EXTEND d [ d' ]
```

Linearly extend the start and end of the current edge a distance "d" and "d'" respectively. The default for "d'" is "d". Enter a zero distance if you only want to extend the other end.

3.4.3 Surface Or Function Entities

The program is capable of handling any "implicit" surface that can be represented by an arbitrary function of a general N-th order polynomial in the global coordinates X,Y,Z with arbitrary reference point x,y,z and real coefficients c.

$$f(X,Y,Z) = \text{fcn} \left(\sum_{i=0}^N \sum_{j=0}^i \sum_{k=0}^j c_{ijk} (X-x)^{i-j} (Y-y)^{j-k} (Z-z)^k \right) = 0$$

Currently the program can go to tenth order (N=10), i.e. 286 coefficients. Therefore, anything from a plane to an arbitrarily oriented elliptical toroid can be modeled. If the reference point lies on the surface, then the first coefficient (the constant term) will always be zero. "fcn" is an optional macro function (intrinsic, e.g. SIN, or user-defined \$FCN).

An "explicit" surface function (sag equation) can also be represented by an arbitrary function of a general N-th order polynomial in the local coordinates X,Y with arbitrary reference point x,y,z and real coefficients c.

$$Z-z = \text{fcn} \left(\sum_{i=0}^N \sum_{j=0}^i \sum_{k=0}^{i-j} c_{ijk} (X-x)^{i-j-k} (Y-y)^j \right)$$

The maximum value for N is 20 (231 coefficients).

```

--- SURFACES [ i ]
    FUNCTIONS
    ENTITIES

```

Signals the program that surface function definition commands follow. "i" is the surface number to be assigned to next command. The default value is one more than the highest surface number previously defined (set to one at start of program execution). EDGE, LENS, and SURFACE data currently reside in the same internal storage locations. Therefore, a SURFACE number cannot be the same as an already defined EDGE or LENS number.

```

--- GENERAL      [ X ] [ x y z ] [ c c' c" ... ]
    COEFFICIENTS  Y
                  Z
                  -X
                  -Y
                  -Z
                  EXPLICIT
--- XiYjZk c [ c' c" ... ]
    :

```

An entirely general surface may be created by entering a reference point and any of the up to 286 single precision polynomial coefficients directly. This can be done either on the GENERAL command starting with the lowest order coefficient or on successive commands that begin with the coefficient of the term represented by $X^i Y^j Z^k$ where i, j, k are integer powers. Any coefficients not entered are set to zero.

ORDER	TERMS
-----+-----	
0 Constant	C
1 Linear	X Y Z
2 Quadric	X ² XY XZ Y ² YZ Z ²
3 Cubic	X ³ X ² Y X ² Z XY ² XYZ XZ ² Y ³ Y ² Z YZ ² Z ³
4 Quartic	X ⁴ X ³ Y X ³ Z X ² Y ² X ² YZ X ² Z ² XY ³ XY ² Z XYZ ² XZ ³ Y ⁴ Y ³ Z Y ² Z ² YZ ³ Z ⁴
5 th order	X ⁵ X ⁴ Y X ⁴ Z X ³ Y ² X ³ YZ X ³ Z ² X ² Y ³ X ² Y ² Z X ² YZ ² X ² Z ³ XY ⁴ XY ³ Z XY ² Z ² XYZ ³ XZ ⁴ Y ⁵ Y ⁴ Z Y ³ Z ² Y ² Z ³ YZ ⁴ Z ⁵
6 th order	X ⁶ X ⁵ Y X ⁵ Z X ⁴ Y ² X ⁴ YZ X ⁴ Z ² X ³ Y ³ X ³ Y ² Z X ³ YZ ² X ³ Z ³ X ² Y ⁴ X ² Y ³ Z X ² Y ² Z ² X ² YZ ³ X ² Z ⁴ XY ⁵ XY ⁴ Z XY ³ Z ² XY ² Z ³ XYZ ⁴ XZ ⁵ Y ⁶ Y ⁵ Z Y ⁴ Z ² Y ³ Z ³ Y ² Z ⁴ YZ ⁵ Z ⁶
7 th order	X ⁷ X ⁶ Y X ⁶ Z X ⁵ Y ² X ⁵ YZ X ⁵ Z ² X ⁴ Y ³ X ⁴ Y ² Z X ⁴ YZ ² X ⁴ Z ³ X ³ Y ⁴ X ³ Y ³ Z X ³ Y ² Z ² X ³ YZ ³ X ³ Z ⁴ X ² Y ⁵ X ² Y ⁴ Z X ² Y ³ Z ² X ² Y ² Z ³ X ² YZ ⁴ X ² Z ⁵ XY ⁶ XY ⁵ Z XY ⁴ Z ² XY ³ Z ³ XY ² Z ⁴ XYZ ⁵ XZ ⁶ Y ⁷ Y ⁶ Z Y ⁵ Z ² Y ⁴ Z ³ Y ³ Z ⁴ Y ² Z ⁵ YZ ⁶ Z ⁷
8 th order	X ⁸ ...
9 th order	X ⁹ ...
10 th order	X ¹⁰ ...

The optional literal axis entry on the GENERAL command instructs the program to create a surface function with symmetry about some plane or axis. This is done by substituting one or two of the coordinates in the functional equation with their squares, effectively doubling the order of the polynomial.

Entry	Mathematical Effect	Geometrical Effect
-----+-----		
X	$X-x \rightarrow (X-x)^2$	symmetry about local x PLANE
:		
-X	$Y-y \rightarrow (Y-y)^2, (Z-z) \rightarrow (Z-z)^2$	symmetry about local x AXIS
:		

The alternate EXPLICIT entry essentially removes all Z-dependent terms from the polynomial except an implied linear z coefficient of -1. This general 2-dimensional polynomial can go as high as 20th order in X and Y.

For example, a sphere of radius 5 centered at 0,0,2 could be defined in the following ways:

```

GENERAL 0 0 2 -25 0 0 0 1 0 0 1 0 1
GENERAL 0 0 2; C -25; X2 1; Y2 1; Z2 1
GENERAL -Z 0 0 2; C -25; X2 1; Y2 1; Z2 1

```

Because it would be tedious for the user to enter even simple geometrical shapes in this manner, the program is equipped with special commands (for the more common surfaces) that generate the polynomial coefficients with a minimal amount of input. More complicated geometries can be programmed with macro commands.

3.4.3.1 Input For Surfaces With No Preferred Symmetry -

```
--- REPEAT [ i ]
```

The new surface created is the same as surface number "i" (default is last). The linear transformation or surface altering commands described in a later section can then be used to make this surface distinct from its "parent".

3.4.3.1.1 Arbitrarily Oriented Planes -

```
--- PLANE NORMAL a,b,c x y z
```

A plane surface containing the point (x,y,z) and with unit normal vector (a,b,c). The normal vector always points into the space on the positive side of the surface where $f(x,y,z) > 0$. As a protective measure, the program always renormalizes the vector length to one.

```
--- PLANE POINTS x y z x' y' z' x'' y'' z''
```

A plane surface that contains the three points (x,y,z), (x',y',z'), and (x'',y'',z''). The positive side of the plane is found by applying the right-hand rule to the three points in the order that they are entered. The second point becomes the surface's reference point.

3.4.3.1.2 An Ellipsoid (or Sphere) -

```

--- ELLIPSOID u v w [ x y z ] [ -X ]
                                -Y
                                -Z
                                X
                                Y
                                Z

```

A general ellipsoid with semi-lengths along each axis of (u,v,w) and center at (x,y,z). The normal vector points out away from the center. Normally a full closed ellipsoid is created (e.g. a complete sphere). However, the additional literal entry can be used instead to specify which half of the ellipsoid is wanted (e.g. a hemisphere).

3.4.3.1.3 Least Squares Fit Of An Arbitrary Set Of Points To A Surface -

```

--- FITTED PLANE      [ EDGES -n      ] [ VECTOR [ c ] ]
    PARABOLOID        i i' i"
    SPHERE
    ELLIPSOID
--- [ coord coord' coord" ]
--- x y z [ ABS ] x' y' z' x" y" z" ...
    REL
--- x y z x' y' z' x" y" z" ...
--- :

```

The given specific surface type is fitted in a least squares sense to the set of points given on successive commands, the points on the last "n" EDGES, or EDGES i through i' by i". When entering points directly, the first point always becomes the surface's reference point. However, if the literal option is placed after the first three entries, this reference point is not on the surface (constant coefficient non-zero) and all other points entered are either measured ABSolutely or RELatively from it. The VECTOR options puts each data point in the current 3D file as a dot of color "c" (default 1) for later plotting by a REPLOT, DRAW or \$VIEW command.

```

--- FITTED [ X ] [ k ] [ t ] [SVD [m]] [FIXTERM [n]] [EDGES -n ] [VECTOR c]
      Y                      i i' i"
      Z
      -X
      -Y
      -Z
      EXPLICIT
--- [ coord coord' coord" ]
--- x y z [ ABS ] x' y' z' x" y" z" ...
      REL
--- x y z x' y' z' x" y" z" ...
--- :
```

A polynomial surface of degree "k" (default 2) is fitted in a least squares sense to the set of points given on successive commands, the points on the last "n" EDGES, or EDGES i through i' by i". The optional literal entry is the same as the one on the GENERAL command and specifies some required symmetry. If the highest order surface coefficients are all zero, the degree of the surface will be reduced accordingly. The program saves the maximum extents in each coordinate direction and uses these as the default LOCAL box. "t" is an optional threshold between 1.E-8 and 1.E-2 (default 1.E-5) below which the relative contribution of a surface coefficient is assumed to be negligible so that it is reset to zero.

The rarely needed FIXTERM option specifies the number of the term to be fixed to 1. The default is the largest component of the normal to the surface through the first point (if it lies on the surface) or zero (constant term).

By default, the EXPLICIT solution is a Cholesky decomposition of the normal matrix which can accept any number of data points. One can elect to do a more robust SVD (Singular Value Decomposition) solution but the maximum number of data points it will accept is limited (see output from DIMENSION command). If "m" is given then only every mth point is used which may be necessary to get below this limit.

When entering points directly, the first point always becomes the surface's reference point and by default the program assumes that the surface also passes exactly through this point, i.e. the constant coefficient of the surface function is set to zero. However, if the literal option is placed after the first three entries, this reference point is not on the surface (constant coefficient non-zero) and all other points entered are either measured ABSolutely or RELatively from it. The VECTOR options puts each data point in

the current 3D file as a dot of color "c" (default 1) for later plotting by a REPLOT, DRAW or \$VIEW command.

3.4.3.2 The User-programmable Surface Intersection -

```
--- USERSURF x y z c [ c' c" ... ]
```

A user-programmable surface with reference point x,y,z and double precision coefficients whose intersection with an arbitrary ray can be coded by the user in the Fortran function USERSURF. The following restrictions apply to this surface:

- o A ray is not allowed to intersect the surface more than once consecutively.
- o The FCN, BEND, FMAP, MULTIPLE/ARRAY, and TEST commands cannot be used in conjunction with this command.
- o Non-orthogonal transformations are not allowed, e.g. SCALE or SKEW
- o They cannot be used later as BOUND, GRIN, or alternate surfaces/normals.

The default definition is a conicoid with vertex at x,y,z and normal direction c,c',c". The fourth and fifth coefficients are the vertex curvature (inverse radius) and the conic constant (0=sphere, -1=parabola), respectively.

3.4.3.3 The User-programmable Function -

```
--- USERFUNC x y z [ fcn ] [ c c' c" ... ]
```

A user-programmable function with reference point x,y,z and double precision coefficients. Its value and gradient at any point (and wavelength) can be coded by the user in the \$FCN named "fcn" or the Fortran function USERFUNC. If the function is continuous in both value and gradient everywhere in space, then there are no restrictions on the use of this function in the program except possibly the application of non-orthogonal transformations to it, e.g. SKEW or non-isotropic SCALE.

If the "fcn" is specified, then the current WAVELENGTH (or the associated ray/beam's wavelength) is passed in the "_0" register, the local "x y z"

coordinates are passed in the "_1 _2 _3" registers, and up to 63 coefficients "c c' ..." in "_4 _5 ... _66". If 4 or more values are returned, then the last 4 entries of the executed function must be the functional value and its gradient vector. If only one value is returned, then it is the functional value and the gradient will be calculated numerically using finite-differences. For example, a sphere of radius 10 centered about the reference point would be done as follows:

```
$FCN SPH _1^2+_2^2+_3^2-_4^2 2*_1 2*_2 2*_3
:
SURFACE; USERFUNC .5 0 -1 SPH 10
```

Otherwise, the default USERFUNC is a aspheric conicoid with vertex at x,y,z and normal direction c,c',c". The fourth and fifth coefficients are the vertex curvature (inverse radius) and the conic constant (0=sphere, -1=parabola), respectively. The remaining coefficients are for a straight polynomial in an offset radial distance including odd terms, i.e. the seventh coefficient is the linear term, the eighth is the quadratic, etc. For example,

```
USERFUNC x y z 0 0 1 c k h d d' d" ...
```

$$f(X,Y,Z) = (Z-z) - \frac{cR^2}{1 + \sqrt{1 - (1+k)cR^2}} - dr - d'r^2 - d''r^3 - \dots$$

$$R^2 = (X-x)^2 + (Y-y)^2$$

$$r = R - h$$

An exception has been implemented when the curvature "c" is zero, i.e.

$$f(X,Y,Z) = k(Z-z) - dr - d'r^2 - d''r^3 - \dots$$

3.4.3.4 Input For Surfaces Initially Normal To A Coordinate Axis -

The second entry on the following commands designates the axis of symmetry (either "X", "Y" or "Z") for the surface. Many of the surfaces created by these commands can extend to infinity unless a LOCAL command follows or where permitted, a trailing aperture option of the following form is specified:

```

--- ... X ... ELLIPSE  a [ a' [ o [ s [ s' ] ] ] ]
      Y      RECTANGLE
      Z      HEXAGONAL a [ o [ s [ s' ] ] ]

```

"a a'" are the heights in the other two transverse directions, "o" an optional central hole ratio, and "s s'" the transverse coordinates of the center of the aperture. For the HEXAGONAL form "a" is the center-to-vertex distance (maximum height).

3.4.3.4.1 Orthogonal Planes -

```

--- PLANE X x [ aperture ... ]
      Y y
      Z z

```

A plane normal to the axis of symmetry at the value given by the third entry. The normal vector points along the positive coordinate direction.

```

--- CORNER X x HEIGHT  d
      Y y  HEXAGONAL
      Z z  LENGTH

```

Creates an axis-aligned corner of a cube (three mutually perpendicular planes) with apex at the location given by the third entry. The size of the corner cube can be specified by either a circular HEIGHT, HEXAGONAL height or an axial LENGTH (maximum cross-section will then be triangular). Note that a small hole is created at the apex since the normal to the function becomes undefined there. Therefore, an on-axis ray will never hit the surface(s).

3.4.3.4.2 Circular Cross Section Horns With Polynomial Profiles -

```

--- HORN X x r [ r' r" ... ] [ FIT n x [ x' x" ... ] ]
      Y y      [ y [ y' y" ... ] ]
      Z z      [ z [ z' z" ... ] ]

```

Creates a surface symmetric about the given axis with a radial profile determined by the following polynomial in the radial distance squared:

$$r^2(d) = r^2 + r'd + r''d^2 + \dots$$

where d represents the particular axial distance coordinate relative to the third entry. The program can also FIT the given set of radial and axial positions to an "n"th order polynomial (n less than or equal to 10).

Rectangular/elliptical horns with different polynomial profiles in the two cross sections can be created with variations of the following format (only the Z axis form is shown for sake of brevity):

```

--- HORN Z z x0 [ x1 x2 ... ] [ FIT n z0 [ z1 z2 ... ] ]
---      y0 [ y1 y2 ... ] [ n' z0' [ z1' z2' ... ] ]
---      q0 [ q1 q2 ... ] [ n'' z0'' [ z1'' z2'' ... ] ]

```

$$X^2 = x0^2 + x1^2 (Z-z)^2 + x2^2 (Z-z)^2 + \dots \quad Y^2 = y0^2 + y1^2 (Z-z)^2 + y2^2 (Z-z)^2 + \dots$$

$$q = q0 + q1 (Z-z)^2 + q2 (Z-z)^2 + \dots$$

The "q" parameter controls whether the cross section is elliptical, rectangular or something in between. See the TUBE command for more details. As before each one of these polynomials can be determined by a least squares FIT to a set of values and positions. Either way you do it, remember that the coefficients or the fit points are relative to the reference point (the third entry). The following restrictions on the polynomial orders apply:

n	less than or equal to 10
n'	8
n+n'	10
n''	6

3.4.3.4.3 General Rectangular/elliptical Cross Section Tubes -

```

--- TUBE X x y z x' y' z' [ q [ q' ] ]
        Y y z x y' z' x'      INNER
        Z z x y z' x' y'      OUTER

```

Using the first form as an example, creates general tube-like surface symmetric about the "X" axis that passes through a closed curve normal to the axis at "x" with semi-major widths of "y" and "z" and a parallel closed curve represented by the primed quantities. The "q" entries control the type of closed curve (0=ellipse,1=rectangle, and a "rounded rectangle" for values in between). In this example, "q" is for the "x" end of the tube and "q'" for the "x'" end so that it is possible to have a fifth order tube whose cross-section varies linearly from an ellipse on one end to a rectangle on the other. If $yz'=zy'$ and $q=q'=0$, then the tube is a second order conical surface. Otherwise, it is of fourth ($q=q'$) or fifth order and there will be several branches to the surface that may or may not have to be excluded later on. The normal vector points out away from the symmetry axis.

```

q=0      elliptical
0<q<1    in between
q=1      rectangular

```

If the two ends of the tube are of different size, then instead of a sloped cone, the two ends can be joined by an INNER or OUTER cylinder and an annular plane at one end.

```

--- TUBE X x y z [ ANGLES b c [ q ] ]
        Y y z x      c a
        Z z x y      a b

```

The form of these commands is the same as the previous ones except that the "a", "b", "c" entries for this general tube-like surface are the angles (in degrees) that the two orthogonal projections of the sides of the tube make with the symmetry axis.

3.4.3.4.4 A Torus (doughnut) -

```

--- TORUS X x h r [ r' [ q ] ]
          Y y
          Z z

```

Defines a torus (doughnut) surface. "h" is the height to the center of the torus ring. "r" and "r'" are the (perpendicular to the axis and parallel to the axis, respectively) semi-major widths of the cross-sectional ellipse ($q=0$) at that height. If the optional "q" is not zero then the cross-section can be varied all the way up to a rectangle ($q=1$).

3.4.3.4.5 A Classical Symmetric Optical Surface -

```

--- OPTICAL X x r['p] [ c d e f g h i j k l ] [ EXPAND ] [ aperture ... ]
          Y y
          Z z
---      [ t r['p] [ c d e f g h i j k l ] [ EXPAND ] ]

```

Defines a classical "optical" surface normal to the axis of symmetry at a value given by the third entry. "r" is the vertex radius of curvature and is negative if the center of curvature is on the negative side of the surface. A zero or very large "r" corresponds to a planar flat. "p" is the vertex radius of curvature of a parabolic term to be optionally subtracted from the base surface. "c" is the conic constant (e.g. 0 is a sphere, -1 is a parabola, etc.). "d" is the 4th order aspheric deformation coefficient, "e" the 6th, and so on up to "l" the 20th. The program can model all surfaces up to 10th order (or 20th order if the conic constant is -1) exactly. Otherwise, the surface is approximated with a truncated formula and if an "aperture" is specified, a table of sag points will be generated in order to check the accuracy of this formula. At the vertex of the surface the normal vector points along the positive coordinate direction. If the highest order aspheric entry is the literal EXPAND, then the conic is expanded into aspheric components up to that order.

Optionally, a second arbitrary "optical" surface a relative distance "t" away can also be included in this one function. For example, this permits modeling both the front and back surfaces of a singlet lens with a single entity.

3.4.3.4.6 Surface From Two Distinct Conic Profiles -

```

--- BICONIC X x r r' k k' [ 3 ] [ aperture ... ]
            Y y              4
            Z z

```

Cubic (default) or quartic bi-conic surface with given vertex radii of curvature and conic constants (0 circle, -1 parabola, etc.). In the "Z" case, the actual equations of the surfaces are given by:

Implicit Cubic (3):

$$Z-z = \frac{x^2}{2 - e(Z-z)} + \frac{y^2}{2 - e'(Z-z)}$$

Explicit Quartic (4):

$$Z-z = \frac{x^2 + y^2}{1 + \sqrt{1 - e x^2 - e' y^2}}$$

where:

$$c = 1/r, \quad c' = 1/r', \quad e = (1+k)c, \quad e' = (1+k')c', \quad x^2 = cX^2, \quad y^2 = c'Y^2$$

The two representations differ in how they blend the two conic profiles together (see the USERAG command for yet another distinct blending). For "e(Z-z)" and "e'(Z-z)" much less than one, the differences between the two versions are normally negligible. However, the cubic is also conic (usually an ellipse) in any plane perpendicular to the given axis and its ray intersection is more robust. On the other hand, the quartic remains a conic in any plane containing the axis, but degenerates on-axis when converted to implicit polynomial form.

3.4.3.4.7 Surface With A User-defined Radial Or Toric Profile -

```

--- USERSAG X x [ fcn ] c c' ... [ TORIC r [ k ] ] [ aperture ... ]
            Y y
            Z z

```

Creates at the given plane, a surface with a user-programmable radial or TORIC

(of radius "r") profile specified by a sag formula. If the "fcn" is specified, then the local transverse height coordinate is passed in the "_0" (or "_") register and up to 66 coefficients "c c' ..." in registers "_1 _2 ... _66". The last expression of the executed function must be the sag value at that height. If "k" is specified, the profile is instead shifted along a conic curve with vertex radius of curvature "r" and conic constant "k" (0=circular, -1=parabolic). Note that specifying a "k" of zero is not the same as an absent "k".

Otherwise, the default is defined in the USERSAG Fortran routine and is a conic plus aspheric terms. The fourth and fifth entries are the vertex curvature (inverse radius) and the conic constant (0=sphere, -1=parabola), respectively. The remaining coefficients are for a straight polynomial including odd terms, i.e. the sixth entry is the linear term, the seventh is the quadratic, etc. For the Z-axis case,

$$\begin{aligned} &\text{USERSAG } Z \ z \ c \ k \ d \ d' \ d'' \ \dots \\ &Z - z = \frac{ch^2}{1 + \sqrt{1 - (1+k)c}h} + dh + d'h^2 + d''h^3 + \dots \quad h = X^2 + Y^2 \end{aligned}$$

3.4.3.4.8 Surfaces Swept Out By A Rotated Curve -

```
--- REVOLUTION X x c0 z1 x1 [z2 zx x2 [z3... [z4... [z5... ]]]] [DECENTER d]
                Y y c0 x1 y1 [x2 xy y2 [x3... [x4... [x5... ]]]]
                Z z c0 y1 z1 [y2 yz z2 [y3... [y4... [y5... ]]]]
```

Takes the general two-dimensional curve defined relative to the given coordinate by the ascending coefficients (up to 21, i.e. fifth), optionally DECENTERS it a distance "d", and then rotates it about the given axis to form a three-dimensional surface of up to twice the order of the curve.

Alternatively, the coefficients of a 1ST, 2ND, 3RD, 4TH or 5TH order curve can be determined by fitting in a least squares sense to a series of points (up to 5000 coordinate pairs).

```

--- REVOLUTION X 1ST z x z' x' ... [ VECTOR [ c ] ]
                Y 2ND x y x' y'
                Z 3RD y z y' z'
                4TH
                5TH
                FIT

```

The FIT option searches for the optimum degree surface automatically. The axial location of the point furthest from the axis becomes the surface's reference point. The threshold on the last LSQFIT command is used to determine if any of the resulting coefficients can be considered negligible (i.e. zero). The VECTOR options puts each data point in the current 3D file as a dot of color "c" (default 1) for later plotting by a REPLOT, DRAW or \$VIEW command.

```

--- AXICONIC X x h x' h' s [ x" h" ] [ LENGTH t ] [ LIST ]
              Y y h y' h' s [ y" h" ]   HEIGHT
              Z z h z' h' s [ z" h" ]

```

Generates a surface by rotating an arbitrarily oriented conic curve about the given axis. This command is similar to the REVOLUTION command except that the second order curve (whose coefficients can be LISTed) is determined from the foci of a conic. This means that the distance in such a plane from the first focus (primed coordinates) to the second focus (double primed coordinates) through any point on the surface (unprimed coordinates) is stationary with respect to small changes, i.e. constant. If the second focus is not specified, then it is assumed to be at infinity and "s" is the angle (in degrees) that it makes with respect to the axis of symmetry. Otherwise, only the sign of "s" affects the generation of the finite foci conic by determining whether the sum or difference of the distances is stationary, i.e. an ellipse or hyperbola, respectively. The resulting normal vector points out away from the symmetry axis.

An optimal LOCAL box for the surface can be automatically created if the other end of the curve is specified either by an axial LENGTH (distance from 3rd entry) or a HEIGHT (distance from axis). The program solves a quadratic for the other coordinate taking the root closest to the first unprimed point (if there are no real roots, a box is not created). The TEST and PARAMETERIZATION will also be automatically set so that one branch (maybe not the desired one) of the surface should mesh properly for PLOTting.

3.4.3.4.9 The Perfect Cartesian Ovaloid -

```
--- CARTOVAL X x d n d' n' [ aperture ... ]  
            Y y  
            Z z
```

Creates a Cartesian Oval surface at the given location on the given axis. This surface perfectly images an axial point a distance "d" in the first medium of index "n" into an axial point a distance "d'" in the second medium of index "n'".

3.4.3.4.10 Explicit Zernike Polynomial Surface -

```
--- ZERNIKE X x c c' c" ... [ aperture ... ]  
            Y y  
            Z z
```

Creates an explicit surface by transforming the University of Arizona FRINGE Zernike coefficient set "c c' ..." to a polynomial. The zeroth coefficient "c" is the constant term (piston). The 36th or last coefficient is the 12th order radial term (spherical). Since these coefficients are usually relative to a unit circle area, it may be necessary to SCALE the surface to its proper dimensions. The first 9 terms are related to the classical third-order optical aberrations:

Term	Dependency	Name
-----+-----+-----		
c	1	Piston
c'	r cosa	x tilt
c''	r sina	y tilt
	2	
:	2r - 1	defocus
	2	
	r cos2a	x astigmatism
	2	
	r sin2a	y astigmatism
	3	
	(3r - 2r)cosa	x coma
	3	
	4 2	
	6r -6r +1	spherical

3.4.3.4.11 Explicit Surface Interpolated From Sampled Data -

--- SAMPLED [name]

Sets up an explicit surface defined by deformation samples found in the binary distribution "name.DIS" (default BRO009.DAT). The surface (and slopes if present in file) are linearly interpolated between data points. The file header should contain directional labels ("X", "Y", and "Z") and data ranges that determine the orientation and size of the surface in its local coordinate system. If not, it will assume it is a z(x,y) surface located at the origin. The file can be created by an external user program, a \$ITER command, or by a MAP command. For example, an approximation to a shallow conical surface could be defined as follows:

```

$ITER X -1. 1. -11 Y -1. 1. -11 Z !negative sample numbers required
Z=.1*SQRT(X^2+Y^2) !z proportional to radial value
SURFACE; SAMPLED ITER !default file name when $ITER not in a macro
LOCAL AXIS Z !convert box from default rectangular to cylindrical
SHIFT Z ... !move it into global position

```

Alternatively, use the following form of the SAMPLED command to create the file directly.

```

--- SAMPLED name X x y y' z z' [ SLOPES ]
                Y y z z' x x'   DIFFS
                Z z x x' y y'
--- d [ s' s" ] d' ...
--- :

```

The third and fourth entries define the reference plane. The fifth through eighth define the surface extent in the other two directions. The command is then followed by the lines of deformation values that cover the given area uniformly up to (and including) the given extents. If SLOPES are to be included, then each sample point has not only a deformation value associated with it but also the slopes in the two other directions. Alternatively, only the deformation values can be given and the slopes then estimated using finite DIFFerences.

3.4.3.4.12 Deformed Cylindrical Surface Interpolated From Sampled Data -

```

--- SAMPLED name RX r x x' a a' [ SLOPES ]
                RY y y'   DIFFS
                RZ z z'
--- d [ s' s" ] d' ...
--- :

```

This variation of the SAMPLED surface is a departure from a cylinder instead of a plane. The third and fourth entries define the reference cylinder (i.e. its axis and radius). The fifth and sixth entries define its extent in the axial direction. The seventh and eighth define the angular (degrees) extent around the axis. The command is then followed by the lines of deformation values that cover the given area uniformly up to (and including) the given extents. If SLOPES are to be included, then each sample point has not only a radial

deformation value associated with it but also the slopes in the axial and angular directions. The angular slope has units of length per radian. Alternatively, only the deformation values can be given and the slopes then estimated using finite DIFFerenceS.

3.4.3.4.13 Circle Swept Along An Explicit Cubic Planar Curve -

```

--- CONDUIT X x x' r [ y [ a [ a' ] ] ]
            Y y y'   z
            Z z z'   x

```

Creates up to a 6th-order surface formed by sweeping a circle of radius $|r|$ along an explicit cubic planar curve starting at the first axial location and ending at the second. If "r" is positive, the circle is perpendicular to the curve at every point. If it is entered as a negative number, the circle is always perpendicular to the specified axis. The end of the curve can be displaced from the axis by the amount given on the sixth entry (default 0). The optional last two entries are the starting and ending angles in degrees of the curve relative to the axis. Their defaults are zero and any values entered must be (usually much) less than 90 degrees in magnitude.

3.4.3.4.14 Superconic Asphere Used In Optical Design -

```

--- SUPERCONIC X x d c c' c" a b' b" [ aperture ... ]
               Y y
               Z z

```

"d" is usually one (or zero). "c" is the vertex curvature (inverse radius). In the Z-axis case, the full implicit equation of the surface is given by:

$$A (Z-z)^2 - 2 B (Z-z) + C S^2 = 0$$

where

$$A = a \quad B = 1 + b'S^2 + b''S^4 \quad C = c + c'S^2 + c''S^4 \quad S^2 = X^2 + Y^2 + d(Z-z)^2$$

Some special cases are:

Case	Non-zero Coefficients
-----+-----	
Plane	None
Parabola	c
"	d=1, a=-c
Sphere	d=1, c
"	c, a=c
Conic (constant k)	d=1, c, a=ck
"	c, a=c(1+k)
Cartesian Oval	d=1, c, c', a, b'
Rational Polynomial	All but d and a

3.4.3.5 Altering Surfaces -

The polynomial coefficients of a surface can be altered to form a differently shaped or oriented surface. Besides the standard linear transformations, the following arbitrary "transformations" can be applied to surfaces.

```
--- FCN fcn
```

Specifies a macro function (intrinsic, e.g. SIN, or user-defined \$FCN) to wrap around the previous surface's base function.

$$f(X,Y,Z) ==> fcn(f(X,Y,Z))$$

3.4.3.5.1 Nonlinear Distortion Of A "normal" Surface -

```
--- ASYM u v
```

This command can be used after the TUBE, OPTICAL, TORUS, REVOLUTION, HORN, CARTOVAL, AXICONIC, BICONIC, CONDUIT, and SUPERCONIC commands to

essentially distort the rotational symmetry of the surfaces. For example, if "Z" is the axis of symmetry, then the surfaces are only a function of the distance from the axis squared:

$$r^2 = x^2 + y^2 \implies u x^2 + v y^2$$

This command does the above replacement so that a more general class of surfaces is possible such as anamorphic OPTICAL surfaces, e.g.

```
OPTIC Z 2.8 3.7 !radius of curvature in X = 3.7
ASYM 1 3.7/4.6 !radius of curvature in Y = 4.6
```

Another example is setting either "u" or "v" to zero so that a surface with cylindrical symmetry becomes an infinite trough in the non-zero direction.

3.4.3.5.2 Creating A Multiple Sheet Version Of A Surface -

```
--- MULTIPLE n f' [ EXPONENT p ]
    d x y z
```

The surface is converted into multiple "parallel" surfaces. In other words, the equation of the surfaces becomes:

$$f(X,Y,Z) = \sum_j^p f' \quad j=0,n$$

Note that the zeroth sheet is just the original surface. Alternatively, the program can calculate f' such that the distance from a point x,y,z on the original surface to the first sheet is d . The exponent "p" is defaulted to 1, but can be used, for example, to get evenly spaced cylinders or spheres ($p=2$).

3.4.3.5.3 Creating Simple Arrays Of Surfaces -

```

--- ARRAY n x y z [ n' x' y' z' ] [ EXPONENT p [ p' ] ] [ BOUNDS
      X n s      Y n' s'      SEARCH [ k ]
      Z
      Y      Z
      X
      Z      X
      Y

```

This command turns the last surface into an $n+1$ by $n'+1$ set of identical but linearly spaced (s and s' apart) surfaces. For the first, more general (possibly skew) syntax:

$$f(X-ax-bx', Y-ay-by', Z-az-bz') = 0 \quad \begin{matrix} p & p' \\ a=i & i=0, n & b=j & j=0, n' \end{matrix}$$

Each instance (i.e. element) can be optionally treated just like a separate entity on a BOUNDS command. Otherwise, if used as a base object surface, the SEARCH for any ray intersection is restricted to the instance/element nearest to where the ray intersects the plane of the array (default or "k" zero). "k" is the number of additional "rings" of instances about this nearest one to also consider. For large arrays, this can speed up the trace calculation by orders of magnitude but runs the risk of some rays unphysically missing the object. Larger values of "k" reduce this risk but slow down the trace.

3.4.3.5.4 Parabolic Bending Along A Given Axis -

```

--- BEND X c [ c' ]
      Y
      Z

```

Bends the surface (relative to its reference point) by replacing the given coordinate in its polynomial function with a quadratic in the other two coordinates, e.g. for a BEND Z:

$$Z ==> Z + .5(c^2 X + c'^2 Y)$$

The c coefficients are the curvatures (inverse radii) of the bending. Note that this operation may double the order of the polynomial.

3.4.3.6 Localizing A Surface With A Simple Bounding Box -

```

--- LOCAL x x' [ y y' z z' ] [ X [ f ] ]
                                Y
                                Z
                                SHORTEST
                                LONGEST
                                NORMAL

```

With the LOCAL command, the previous surface can be simply bounded by the specification of allowable ranges on each of the current coordinates. The unprimed entries are the lower bounds, primed the upper. If a coordinate direction is also given, then the surface is constrained by a cylinder of constant elliptical (as opposed to rectangular) cross-section in planes perpendicular to the given coordinate. This axis can also be determined by the SHORTEST limit box dimension, the LONGEST, or the coordinate direction nearest to the surface NORMAL. An optional inner boundary of fractional height "f" [default is zero] can be used to put a proportional hole in the surface in this direction.

```

--- LOCAL [ OFF      ]
      REPEAT i [ X [ f ] ]
      STATS      Y
      AXIS       Z
      -X x       SHORTEST
      -Y y       LONGEST
      -Z z       NORMAL
      +X x'      OFF
      +Y y'
      +Z z'
      EXPAND r
            X r
            Y
            Z

```

The above short form can also be used to temporarily turn OFF future limits checking, to set the limits on the current surface to those of a previous surface "i", to the ranges of ray intersections found from the previous TRACE

STATS command (assuming AXIS LOCAL command in effect), or to reset the limits AXIS or just one of the six limits values. EXPAND can be used to enlarge (or shrink) by a relative amount "r" the entire limits box or just in one direction, e.g. "EXPAND -.1" shrinks the entire box by 10%.

After a LOCAL command, any linear transformation commands will be applied to the surface's local-to-global transformation matrix and not to the surface coefficients themselves.

3.4.3.7 Contour Map Of A Cut Of The 3-dimensional Surface Function -

```
--- FMAP f [ f' n ]  
          [ r x y z ]
```

This command can follow any surface definition commands and will generate a contour map of one (or more) cuts of the current three-dimensional surface function. The paper coordinates and ranges are given by the last WINDOW command. The depth coordinate's cutting plane value is "f" (or "f" to "f'" in "n" steps). The map will indicate the different branches, sheets, and regions of the function and its sign so that any problems with signs of normals and unexpected intersections can be graphically verified. Optionally, one can display the difference between the surface and a sphere of radius "r" centered at "x y z". In either case, a distribution file of the surface data is also created on logical unit 9 (BRO009.DAT) for later processing by the DISPLAY commands.

3.4.3.8 Testing For A Particular Branch Of A Surface Function -

```
--- TEST  OFF  
          -POINT      x y z  
          +POINT  
          -DIRECTION  a,b,c  
          +DIRECTION  
          -AXIS       a,b,c [ x y z ]  
          +AXIS
```

This command sets up a test for a particular branch of a surface function or turns OFF a previous or default test. A surface function of order 2 or greater can have multiple branches of which usually only one is the desired one. A good example is the two distinct branches of a surface formed from a hyperbola. For each intersection of the surface, the program takes the DIRECTION or the vector from the POINT or AXIS to the intersection and "dots" it with the gradient of the surface function at the intersection. The intersection is then valid only if the result has the indicated sign.

3.4.3.9 Renormalizing The Coefficients Of A Surface Function -

```
--- RENORM [ c ]  
          [ term c ]  
          [ MIN  c ]  
          [ MAX  c ]  
          [ FUNC c [ x y z ] ]
```

Renormalizes the coefficients of the previous surface function. If actually a surface (function=0), then the renormalization will not affect the location of the surface, but may change the direction of the normal to the surface or avoid numerical overflow problems. If no additional entries are made, the coefficients are normalized such that the largest and smallest ones are equally spaced in magnitude about 1. If a single numerical value is entered, all the coefficients are divided by "c". If a "term" is specified (i.e. $X_i Y_j Z_k$ or the MIN / MAX), then the coefficients are normalized so that particular coefficient becomes exactly equal to "c". The last form allows one to set the value of the FUNC to "c" at the point x,y,z (default is the reference point).

3.4.3.10 Bounding Volume Formed From Surface And Its Local Box -

```
--- SOLID PLUS [ NOHOLE ]  
      MINUS  
      OFF
```

Make current surface a SOLID formed from the interior of its LOCAL box and the given side of the function. Currently, only affects the BOUNDS command. The

NOHOLE option excludes the interior hole of the LOCAL box if one exists. If the volume needed can be represented by just a LOCAL box, then use the above command with a dummy degree zero surface, e.g.

```
SURFACE n
  GENERAL x y z +1
  LOCAL ...
  SOLID PLUS
:
OBJECT
:
BOUNDS +-n
```

3.4.3.11 Controlling Parameterization For Meshing Surface -

```
--- PARAMETERIZE -X
                    -Y
                    -Z
                    +X
                    +Y
                    +Z
```

Sets the parameterization of the current surface to be either parallel "-" or perpendicular "+" to the given local axis. A hint given to the program so it can figure out how to mesh the surface (for PLOT SURFACE/MESH/FACETS, VUFACETS, and GRID/EMIT OBJECT/ENTITY commands) and is meaningless if the surface does not have a LOCAL box defined either implicitly or explicitly. Since this is automatically set for most surfaces, it is rarely necessary to override with this command. If you do, remember that the program cannot mesh (facet) the surface if the LOCAL box clips the surface along a parameterization direction. However, do not set the LOCAL box too large in this direction since this can also prevent the meshing or facetting of the surface. This command and whether a surface can be meshed (facetted) have absolutely no effect on tracing rays/beams to or through the surface.

3.4.3.12 Altering Specific Polynomial Coefficients -

```
--- ALTER term a [ b ] [ term' a' [ b' ] ...
```

Change the coefficient of the given polynomial "term" to be "a" plus "b" (default 0) times its current value. For example, to multiply the constant term by 2 and change the Z term to -1:

```
ALTER C 0 2 Z -1
```

3.4.3.13 Converting To Explicit Form -

```
--- EXPLICIT [ X ] m [ SVD ]  
           Y  
           -Z
```

Converts the current surface function to an explicit polynomial of degree "m" (maximum 20). This may be needed if the function is to be used as a DEFORM on an OBJECT. The optional axis entry allows order doubling of certain coordinates in order to enforce symmetry. The conversion is actually done by a Cholesky or SVD least squares fit operation on the original surfaces' mesh points. Therefore, the original surface must have a LOCAL box and should probably have PARAMETERIZE set to -Z. The RMS and MAXimum sag errors are always displayed upon completion.

3.4.4 Lens Entities

ASAP defines a lens entity to be a sequence of refractive or reflective conicoid surfaces bounded by circular apertures and separated by homogeneous MEDIA. When assigned as a whole to an OBJECT, rays from other objects can only intersect the first or last conicoid of the lens. The ray then propagates sequentially through the lens until it fails or exits at one of the ends. The advantage of a LENS object over individual SURFACE (e.g. OPTICAL) objects is not only simplicity but also speed in both raytracing and 3D graphical representation (see PLOT LENS).

```

--- LENSES [ i ]
    ENTITIES

```

Start defining lens entities at number "i" (default is one more than highest currently defined). EDGE, LENS, and SURFACE data currently reside in the same internal storage locations. Therefore, a LENS number cannot be the same as an already defined EDGE or SURFACE number.

3.4.4.1 General Sequences Of Conicoids -

```

--- SEQUENCE [ CURV ] [ HEIGHT ]
          RADI      DIAM
--- x y z a,b,c h [ r['p] k['s] o m ]
--- d h [ r['p] k['s] o m ]
--- :

```

Enter the lens directly as a sequence of conicoids (one line of input per surface). There are two formats, a long and short. In either case, the last 5 entries are the same. "h" is the aperture HEIGHT (semi-diameter) or DIAMeter. "r" is either a vertex CURVature or RADIus. "k" is the conic constant (e.g. 0=sphere, -1=parabola). "o" is the central obscuration (or hole) ratio. "m" is the number or name of the medium that follows this conicoid (use -1 or REFL for a reflector).

The first optional paired entry "p" is either a vertex CURVature or RADIus of a paraboloid that is subtracted from the conicoid. To get an aspheric "flat" (e.g. corrector plate), set "p" equal to "r". The fourth order and higher aspheric terms are then identical to those of the series-expanded conicoid times "s".

For the long format, "x y z" are the global coordinates of the vertex (center) of the conicoid and "a,b,c" is the normal to the conicoid at that point. Alternatively, these six entries can be replaced with a distance "d" from the previous conicoid. Therefore, the first conicoid of a SEQUENCE must always be entered in the long format.

```

--- REPEAT [ i ]

```


The "i"th lens (default last) is duplicated so that it may be displaced and/or rotated separately of the parent using the linear transformation commands.

```
--- COMPOSITE [ -n ]
               i [ i' ]
```

Combine the last "n" lenses, lens "i" through "i'" or all defined since LENS command into one lens. Note: IDEAL or PERFECT lenses are not allowed in a composite set.

3.4.4.2 Standard Lens Elements -

The following lens entities start out normal to one of the global coordinate axes (2nd entry) and at a particular location (3rd entry) on that axis. They are provided to simplify the input of standard optical elements.

```
--- MIRROR X x h [ f k o ]
           Y y
           Z z
```

A mirror of aperture height "h", focal length "f", conic constant "k" (e.g. 0=sphere, -1=parabola), and central hole ratio "o".

```
--- SINGLET X x t h m [ RD r r' ]
           Y y          CV c c'
           Z z          FL f b [ a      ]
                           a      APLANAT
```

A two refractive surface singlet lens of thickness "t", aperture height "h", and internal medium "m" (number or name). The shapes of the two surfaces are given either directly in terms of curvatures "c c'" or radii "r r'", or indirectly in terms of overall focal length "f" and bending factor "b". The bending factor is defined as the ratio of the sum of the two curvatures to the difference. For a positive lens, 0=bi-convex, -1=plano-convex, and 1=convex-plano. For a negative lens, just substitute concave for convex in the previous examples. "a" an optional conjugate factor, i.e. one plus the object-to-image magnification divided by one minus the magnification

(0=one-to-one imaging, 1=infinite object distance, -1=infinite image distance). If it is specified, one conic constant is automatically calculated to produce zero third-order spherical aberration. If the APLANAT option is used, the bending factor is automatically determined for the given "a" so that third-order coma is also eliminated (assuming the thin lens approximation applies).

```

--- MANGIN X x t h m [ RD r r ]
              Y y          CV c c'
              Z z          FL f b [ a      ]
                          a      APLANAT

```

The format of this command is identical to that of the SINGLET except that the second surface is reflective, i.e. rays are refracted twice at the first surface.

```

--- DOME X x t m [ r [ r' ] ]
          Y y
          Z z

```

Creates a single refractive element of axial thickness "t" and media "m" with front and back radii of curvature "r" and "r'", respectively. The side with the shortest radius will be a complete hemisphere while the other will be truncated at the same plane. The default for "r" is "t" (solid hemisphere) while the default for "r'" is r-t (concentric shell).

```

--- DOUBLET X x t h m m' [ f b r ]
            Y y
            Z z

```

A cemented doublet lens (a positive and negative lens in contact) of overall thickness "t", aperture height "h", media "m m'", focal length "f", and overall bending factor "b". "r" is the ratio of the focal length of the first element to the second. For an achromatic doublet this is also the ratio of dispersions (the default if "r" not given).

```

--- TELESCOPE X x h FL1 f [ SEP s FL2 d ] [ STOP p [ m ] ] [ FOV a [ l ] ]
              Y y  FL      MAG    BWD
              Z z

```

Creates a one or two mirror telescope with or without a corrector plate and/or field flattener. The final design will always be corrected for third-order

spherical aberration and as many other aberrations as possible. The primary mirror is located at the plane given by second and third entries. "h" is the entrance pupil height. The remaining parameters are:

	Sign
FL1 Focal length of primary mirror	+
FL Overall focal length of telescope	+C, -G *
SEP Separation distance between mirrors	+
MAG Secondary magnification factor	+C, -G *
FL2 Focal length of the secondary mirror	-C/+G *
BWD Position of final image relative to primary	+/-
STOP Distance from real object space aperture stop to primary	+
"m" media number or name for a corrector plate at stop	+
FOV Semi-field angle in degrees	+
"l" media number or name for a field flattening lens	+

* C = 2-mirror Cassegrain configuration, G = 2-mirror Gregorian configuration

```

--- RIGHT X x h m [ Y,Z ]
      PENTA Y y      Z,X
            Z z      X,Y
  
```

Creates the specified 90 degree deviation prism with aperture height "h" and medium "m" (name or number), and output coordinate direction given by the last entry. Actually, the best way to model any prism in ASAP is to write a macro that creates a planar faced object from one or two EDGES (see supplied UTIL.LIB).

```

--- WEDGE X x h m a [ t ]
      Y y
      Z z
  
```

Wedge of glass of aperture height "h", medium "m" (name or number), and angle "a" between the two faces. "t" is an optional thickness (default is "h").

```

--- AFOCAL X x l h h' m m'
      Y y
      Z z
  
```

A two element afocal telescope of overall length "l", input height "h", output height "h'", and media "m m'". If the heights are of opposite sign, then there will be an intermediate real focus. The lenses (or mirrors) used will have

conic constants assigned to them so that there is no axial aberration.

3.4.4.3 Perfect Eikonal-based (realistic) Lens -

```

--- PERFECT X x f h [ t [ h' ] ]
            Y y
            Z z

```

A "perfect" (but realistic) lens of focal length "f" (signed), input height "h" (less than magnitude of f), output distance "t" (default 0), and output height "h'" (default $f \cdot \tan(\arcsin(h/f))$). The output ray vectors are determined from the input ray vectors by the solutions to the eikonal (characteristic) function for perfect imaging of an object plane at infinity (image plane at back focal plane) and no spherical aberration of the principal points. As opposed to the the IDEAL lens, there will be blurring ray aberrations at all other conjugates. Two back-to-back PERFECT lenses with a small collimated space between them can be used to perfectly reimage a finite-distance object to a finite-distance plane with a magnification equal to the ratio of their focal lengths.

3.4.4.4 Ideal Lens Element (ABCD And Jones Matrices) -

```

--- IDEAL X x h [ t [ h' ] ] [ RANA ] [ fcn ]
            Y y                RANB
            Z z
--- [ a b c d [ o ] ]
--- [ p q r s ]

```

An "ideal" (but unphysical) lens of input height "h", output distance "t", and output height "h'". The input ray vectors are linearly related to the output ray vectors by the 2x2 ABCD matrix given on the next line of input (default is an identity matrix). For a perfect lens of focal length f, $a=1$ $b=0$ $c=-1/f$ $d=1$. For an afocal system of angular magnification m, $a=1/m$ $b=0$ $c=0$ $d=m$. For a "nonlens" where the output rays are extensions of the input rays, $a=1$ $b=t$ $c=0$ $d=1$. For a telecentric lens of focal length f and semi-field angle u, $t=f$ $a=1$ $b=f$ $c=-1/f$ $d=0$ $h'=h+f \tan u$. The input and output media are assumed to be the same isotropic medium. Therefore, the determinant of the matrix should be

unity, i.e. $ad - cb = 1$. There are no ray aberrations at all conjugates. However, the wavefronts from this lens can only be perfectly spherical for a single conjugate. "o" is the object distance from the first surface for this perfect imaging. For an infinite object distance, enter zero (the default).

Ray Matrix

a b

c d

Jones Matrix

p q

r s

The optional additional line of input are the four elements of the 2x2 complex Jones matrix for linearly altering the polarization states of transmitted beams. It is important to point out that in general Jones matrices are really only defined for beams normally incident on the reference plane. Therefore, non-normal incidence beams may have their polarizations (and fluxes) altered in unexpected ways.

The RAN options randomize azimuthally (around the axis) the output rays. RANA does just the angle by rotating the output ray about its exit point. RANB does both position and direction by rotating about the center of the output aperture. These can be used to simulate some non-imaging effects. The fluxes of the rays can be apodized both radially and directionally if a \$FCN "fcn" is specified. The function takes as its first argument _1 the normalized (between 0 and 1) radial position of the ray. The optional second argument _2 is the sine of the angle from normal for the ray direction. The value returned (last expression) is multiplied with the ray flux to get the new ray flux. For example, these two options could be combined to do an efficient simulation of an incoherent multi-mode fiber:

```
NA=.5 !numerical aperture of fiber
$FCN CUTOFF RECT(_2/(2*NA)) !directional cutoff
IDEAL Z 0 .1 1000 .1 RANB CUTOFF !1000mm long 100um fiber
```

3.4.4.5 Altering Specific Data On A Set Of Lens Conicoids -

```
--- ALTER n [ n' [ n" ] ] data a [ b [ j ] ] data' a' [ b' [ j' ] ] ...
```

Alters the given data on conicoids "n" through "n'" (default n) for the current lens, i.e. for $i = n$ to n' by n'' (default 1):

```
data(i) = a + b data(j)      [ defaults: b=0,j=i ]
data'(i) = a'+ b'data'(j')
      :
```

where

"data"	Description
-----+	-----
X	x coordinate of vertex position
Y	y
Z	z
U	x component of vertex normal
V	y
W	z
H	Aperture height
C R	Vertex curvature or radius
K	Conic constant
O	Obscuration (hole) ratio
M	Medium number
C0 R0	Curvature or radius of subtracted parabola
S	Aspheric scale factor

3.4.4.6 Imaging A Global Point Through The Lens -

--- IMAGE k x y z k'

Images the given point "x y z" in space "k" of the current lens into space "k'" and displays the coordinates. The spaces are numbered from 1 (before the first conicoid) to N+1 (after the last conicoid N). It uses the auxiliary axis technique that steps the image through the centers of curvature of the conicoids (the aperture semi-diameters, conic constants and obscuration ratios of the conicoids do not affect the imaging) and always produces an image even if it is virtual or unphysical. The resulting imaging transform is stigmatic (points go into points) but not necessarily collinear (lines go into lines). Therefore, it is only an approximation since in any real optical system, the image will be aberrated (not a perfect point focus).

3.4.4.7 Displaying The Aberrations Of A Centered Imaging System -

```
--- ABERRATIONS [ l l' ] [ LIST ] [ PLOT ] [ datum d ] [ datum' d' ] ...
```

Display the image aberrations of all of the current lens or just conicoids "l" through "l'" (up to 120 total). Since the analysis is valid only for centered lens systems, the conicoids must have a common axis. If not, the lens is temporarily "unfolded" to make it so. The first-order operating data must be supplied with the additional entries, e.g. the data for the marginal/axial ray from the center of the "object" through the edge of the limiting aperture stop and the chief/principal ray from the edge of the "object" to the center of this stop.

Object Space (before first conicoid):

datum	default	description
U0	0	Marginal ray slope (tangent of angle)
UB0	.01	Chief ray slope (tangent of semi-field angle)
H0	1st apert	Marginal ray height at entrance pupil plane
H1	"	Marginal ray height at first conicoid
HB1	0	Chief ray height at first conicoid
TH0	0	Distance from entrance pupil to first conicoid
KTH0	none	Conicoid number of actual aperture stop (0 to vary, .1 for telecentric in image space)
FB	0	Fractional evaluation fields (up to 5, e.g. 0, .5, .7, .866, 1)

Image Space (after last conicoid):

datum	default	description
UF	initial	Marginal ray slope after last conicoid (curvature solve)
BKF	NA	Fix back focal distance (last conicoid to paraxial focus)
THF	0	Distance from paraxial focus to desired focal surface
CVF	0	Curvature of focal surface
CCF	0	Conic constant of focal surface
KTHF	NA	Solve for best axial position of focal surface
KCVF	NA	Solve for best curvature of focal surface
KCCF	NA	Solve for best conic constant of focal surface

If the LIST option is present, the following surface-by-surface tables are produced:

- Paraxial marginal and chief ray traces.
- Seidel primary/secondary aberrations (3rd-order, axial/lateral color).
- Real marginal and chief ray traces.
- Glass and conicoid data (after solves).

In any case, the final 1st-order (primary color), 3rd-order, 5th-order, and selected 7th/9th-order aberration coefficients (in waves) are listed for actual stop plane coordinates and relative to the ideal paraxial focus. These coefficients are computed using both analytical formulas and real-ray data matching. The Pro version carries out all these calculations in double precision. However, since Basic uses only single precision here, the accuracy can suffer for either a large number of surfaces or delicately balanced configurations.

The optional PLOT displays the traced lens and the following graphs:

- Ray fans (image deviation versus aperture) for each evaluation field.
- Chief ray sagittal (S) and tangential (T) foci as a function of field.
- Chief ray (D) and spot centroid (C) fractional distortion versus field.
- Polychromatic (A)ctual and monochromatic (B)est-focused RMS spot vs. field.

Except for the RMS spot size graphs, solid curves are computed from the approximate aberration coefficients (with color terms) while dotted curves are center-wavelength real-ray points.

3.4.4.8 Locally Or Globally Optimizing A Centered Imaging System -

A lens can be optimized using a brute-force algorithm that finds either the nearest minimum or (given enough time) the global minimum without requiring the merit function to be continuous (i.e. no derivatives are used). This technique also allows constraints to be handled in a straight-forward and exact manner. All this is made possible by an extremely fast method for determining the RMS spot size at any field location (e.g. it does not have to do iterative ray-aiming to hit the internal stop surface of a wide angle lens). On a typical system, over a million merit function evaluations per minute can be achieved, independent of the number of fields.

This ASAP feature is not intended to replace a full-blown lens-design program since it is currently limited to unvignetted, centered, focal systems (the vast majority of manufactured units) and does not have any built-in multi-configuration or tolerancing capability. However, these and other advanced capabilities can be easily emulated with some additional macro-language programming (e.g. \$ITER). At the very least, this specific optimization feature can be used to quickly find (possibly unique) designs that may then be tweaked (if necessary) elsewhere.

The ABERRATIONS command can be followed by any number and combination of these subsequent commands:

```
--- VARIABLES [ OFF ] [ LIST ] [ TH [ i i'... ] ] ...  
CV  
CC  
CP  
AS  
BN  
GL
```

Declares what lens construction parameters are to be varied during optimization. The basic ones are the THicknesses, CurVatures, and Conic Constants on any set of conicoids listed after each. If the conicoid number is positive, the parameter is free to vary. If it is entered as a negative number, that particular parameter is subtracted from the variable list (i.e. it is frozen at its current value). If no conicoid set is given, then all but the last conicoid is allowed to vary. These commands are cumulative until another ABERRATION command or an OFF option resets all the parameters to their default frozen state. The new variable set can also be LISTed where asterisks indicate each active one.

The sag of higher-order aspheric surfaces (like corrector plates) is given by AS (default 1) times the difference between the conic defined by CV/CC and a parabola with vertex curvature CP.

BeNding is a composite variable that changes the curvature of the given conicoid(s) while keeping constant the difference in curvatures between it and the next conicoid in the lens. This approximately fixes the power (and focal length) of the two adjacent conicoids while allowing their aberration contributions to vary. It works with both glass elements and air spaces.

In global optimization mode, the optimum GLass (or combination of GLasses) can be found from all the currently defined MEDIA. Since there can be potentially 6 variables per conicoid, an optimization can use up to 720 variables! A large number of variables can quickly become impractical even for a local optimization because runtimes go approximately as it's cube.

```
--- MINIMIZE [ w w'... ] [ control c ] [ control c' ] ...
```

Proceed to minimize the maximum RMS spot size or the average of the previously specified fields weighted by the "w"s.

Constraint controls:

control	defaults	description
-----+-----+-----		
DIST	1 (100%)	Maximum allowed fractional spot-centroid distortion
GLTH	0	Minimum center and/or edge glass thickness
	infinite	Maximum "
TLEN	infinite	Maximum total length of conicoid set
	1	Optional starting conicoid
	last	Optional ending conicoid
SDIA	infinite	Maximum unvignetted semi-diameter of conicoid set
	1	Optional starting conicoid
	last	Optional ending conicoid
BKFD	0	Minimum back focal distance (last conicoid to focus)
BKWD	-infinite	Minimum back working distance
UBAR	1	Maximum magnitude of final chief ray slope

If the image quality of a design does not get better during an optimization, it is usually due to conflicting constraint violations or constraint violations not affected by the specified variables.

Normal optimization controls:

control	default	description
-----+-----+-----		
MULT	1 (local)	Number of multiple trial solutions (large for global)
SEED	987654321	Random (large odd integer) or quasi-random (zero) seed
TARG	1.E-9	Take first solution that drops below this RMS spot size

Advanced optimization controls:

control	typical	description
-----+-----+		
TOLR	.0001	Fractional change in merit function at local minima
DELV	.001-1	Maximum allowable randomization of normalized variables
ITER	10	Minimum number of iterations per local optimization

These advanced controls are automatically determined by the MULT value and the number of variables. In rare circumstances, they may have to be set explicitly.

In addition, the underlying design engine actually has a comprehensive "pickup" (a conicoid variable is constrained to follow one at a previous conicoid) and "paraxial solve" (a conicoid variable is determined from paraxial ray-trace requirements) capability. These are currently hidden from the user but are used in the following circumstances:

1. The focal length of the lens is always held by a paraxial marginal ray solve for the curvature of the last conicoid and sometimes a previous one coupled to it (see 3 and 5).
2. The nominal focal plane position is found from a paraxial marginal ray solve for the back focal distance that results in a zero ray height.
3. If the starting design is symmetric and only the conicoid parameters for the front half are varied, then the lens will automatically remain symmetric during optimization.
4. If more than one element references the same glass and only the first instance of that glass is varied, then all other instances will follow automatically.
5. If an element is actually a reflector where the surface before and after it are geometrically the same and even though internally two conicoids are used to represent the duplicate surface, the second instance will always replicate any changes in the first. A Mangin back-surface mirror and a Maksutov telescope (where the secondary is an aluminized spot on the back of the meniscus corrector) are common examples of this condition.

Stores the current state of the lens back into the ASAP internal database OR optionally in an ASAP input file, ASAP MACro file, CodeV SEquence file, ZeMaX file, and/or OSLO LENS file. The commercial lens design programs can usually read these simple versions of their files except for possibly the private glasses which can be avoided if standard catalog glasses like Schott are used. The standalone ASAP files automatically create lens edges, do a basic plot with ray traces, and produce spot diagrams across the image surface.

3.5 DEFINING OPTICAL PROPERTIES

3.5.1 System Geometry Length Units And Flux Label

```
--- UNITS [ length ] [ 'flux' ]
```

Sets the length units of the system geometry to one of the following "length"s:

IN	or	INCHES
MM	or	MILLIMETERS
YD	or	YARDS
FT	or	FEET
MI	or	MILES
M	or	METERS
KM	or	KILOMETERS
MIL	or	MILS
UM	or	MICRONS
CM	or	CENTIMETERS
UIN	or	MICROINCHES

The labeling of "flux" units can be set to the given string (up to 11 arbitrary characters). Some typical ones would be WATTS, LUMENS or PHOTONS/SEC.

3.5.2 Interpolation Wavelengths

```
--- WAVELENGTH w [ w' w" ... [ name ] ]  
                                UNITS u
```

Sets the wavelength for any future calculations to "w" in normally system units. This command must precede any ray definitions in order to properly set up the Gaussian beams used to calculate the diffracted field.

"u" is an optional scale factor that converts "w" into system units and is defaulted to 1. Some input parameters have units that are the same as "w" (see INTERFACE command) so that it is sometimes more convenient for these parameters to have different units than the physical system. Alternatively, if a basic UNITS command has already be issued, then any of the following "name"s can be used to calculate "u" automatically:

```
A or ANGSTROMS  
NM or NANOMETERS  
UM or MICRONS  
MM or MILLIMETERS  
CM or CENTIMETERS  
M or METERS
```

Multiple wavelengths in either ascending or descending order should be entered for any MEDIA or COATING commands that have multiple refractive indices or properties so as to identify the wavelength for each entry.

3.5.3 Refractive Media

```
--- MEDIA [ m ]  
--- n [ n' n" ... ] [ options ... ] [ 'name' ]  
    catalog_glass  
--- ...  
    :
```

Starting with medium "m", define media with real (or complex) refractive

indices "n" or from its literal designation in a glass catalog (e.g. SCHOTT_BK7). The default value for "m" is one more than the largest medium number defined and is set to one at the start of program execution. "name" is a descriptive designator that the user can assign to this medium (only the first 16 characters are stored for future reference by the program). Optionally, separate indices can be entered (or computed) that correspond to the wavelengths entered on the last multiple WAVELENGTHS command. The program will then interpolate to get the refractive index at any desired wavelength.

3.5.3.1 Absorption Or Gain Option -

```
--- ... [ ABSORB a [ j q t ] ]
```

The volume absorption "a" is in inverse length units and is positive for an absorbing medium or negative for a gain (lasing) medium. If an absorption is not given (or zero) the program uses the wavelength and the imaginary part of the complex refractive index (if specified) to calculate the absorption. Inhomogeneous absorption or gain can be handled by assigning to the medium a GENERAL polynomial in the global coordinates X,Y,Z or USERFUNC function (with additional wavelength w dependence). The magnitude of "j" is the "SURFACE" designation for this function. The absorption coefficient at each point in the medium is then given by:

$$a(X,Y,Z) = a^q([w;]X,Y,Z)$$

"t" is the step length to be used by the program when tracing a ray in this inhomogeneous medium.

3.5.3.2 Gradient Option -

```
--- ... [ GRIN k p t [ l ] ]
```

In a similar fashion, Gradient index (GRIN) materials can be handled by assigning to the medium a GENERAL polynomial in the global coordinates X,Y,Z or USERFUNC function (with additional wavelength dependence). The magnitude of

"k" is the "SURFACE" designation for this function. The refractive index squared at each point in the medium is then given by:

$$n^2(w; X, Y, Z) = n^2(w) f([w;]X, Y, Z)$$

where w is the wavelength. If the constant coefficient of the function f is unity (1), then the refractive indices entered after the MEDIA command will correspond to those at the function's reference point. "t" is again the step length to be used by the program when tracing a ray in this inhomogeneous medium. "l" is the maximum number of steps a ray will be allowed to take in the medium (default 1000).

3.5.3.3 Birefringent Uniaxial Crystal Option -

```
--- ... [ CRYSTAL a,b,c [ m' ] ]
```

This option identifies the medium as a birefringent uniaxial crystal with optical axis direction "a,b,c" and given extraordinary refractive indices. The ordinary indices are specified on a previous medium "m'" (default is last). For example, a calcite crystal with its optical axis initially aligned with the Y direction would be defined as follow:

```
MEDIA
1.66 !ordinary indices for following extraordinary
1.49 CRYSTAL 0 1 0 =CALCITE
```

If SPLIT is one or higher, the program will automatically generate an ordinary and extraordinary ray / beam at each crystal interface.

3.5.3.4 Inhomogeneous Monte-Carlo Scattering Option -

```
--- ... [ SCATTER m [ k e t [ l ] ] ]
      USER
```

In the case of SCATTER, "m" is a reference to a VOLUME scattering MODEL.

Inhomogeneous scattering can be handled by assigning to the medium a GENERAL polynomial in the global coordinates X,Y,Z or USERFUNC function (with additional wavelength w dependence). The magnitude of "k" is the "SURFACE" designation for this function. The scattering at each point in the medium is then multiplied by:

$$\frac{e}{f([w;]X,Y,Z)}$$

For the case of USER, "m" AND "e" are arbitrary integer and float parameters, respectively, passed to the user-supplied Fortran function USERSCAT. If "k" is zero, the positional coordinates of the ray are also passed. Otherwise, the value and gradient of FUNCTION entity "k" are passed instead. With the additional passing of the current complex refractive index, the Fortran function then allows one to change the wavelength, flux, and / or direction of the ray while returning a mean-free pathlength for the next interaction.

"t" is always the step length to be used by the program to sample the inhomogeneous medium when tracing a ray.

3.5.4 Coatings

Optical INTERFACE COATings can be specified (and optionally assigned an up to 16 character 'name') either in terms of their wavelength PROPERTIES, angular MODELS, or their actual LAYER structure.

```
--- COATINGS [ k ] [ PROPERTIES [ a ] ]
--- r [ t r' t' r" t" ... ] [ 'name' ]
:
```

Starting with coating "k", coatings with real energy (or complex amplitude) reflectances "r" and transmittances "t" are entered. The default value for "k" is one more than the largest coating number defined and is set to one at the start of program execution. Optionally, separate coefficients can be entered that correspond to the wavelengths entered on the last multiple WAVELENGTHS command. The program will then interpolate (linearly in complex amplitude) to get the coefficients at any desired wavelength. For example, the actual reflectivities and transmissions at a wavelength "v" between the first two

WAVELENGTHS "w w'" would be:

$$R(v) = |r(1-u) + r'u|^2 \quad T(v) = |r(1-u) + r'u|^2 \quad u=(v-w)/(w'-w)$$

The "r"s and "t"s in above equations are the entered complex amplitudes or the the square roots of the real energy coefficients.

```

--- COATINGS [ k ] MODELS i j m n
--- r t r' t' r" t" ... [ 'name' ]
:
--- COATINGS [ k ] MODELS
--- r i j t m n r' i' j' t' m' n' r" ... [ 'name' ]
:
```

Starting with coating "k", coatings with real energy (or complex amplitude) reflectances "r" and transmittances "t" are entered. The default value for "k" is one more than the largest coating number defined and is set to one at the start of program execution. Also, separate angular properties can be specified by using previously defined (usually RAWDATA) MODELS where:

```

i Model for reflected S polarization
j Model for reflected P polarization
m Model for transmitted S polarization
n Model for transmitted P polarization
```

Optionally, groups of 6 numbers can be entered so that each group corresponds to a wavelength entered on the last multiple WAVELENGTHS command. For example, the actual reflectivities and transmissions at an incidence angle "a" and a wavelength "v" between the first two WAVELENGTHS "w w'" would be:

$$R(a, v) = \sqrt{|r(1-u)f(i, a) + r'u f(i', a)|^2}$$

$$R(a, v) = \sqrt{|r(1-u)f(j, a) + r'u f(j', a)|^2}$$

$$T(a, v) = \sqrt{|t(1-u)f(m, a) + t'u f(m', a)|^2}$$

$$T(a, v) = \sqrt{|t(1-u)f(n, a) + t'u f(n', a)|^2}$$

where $u=(v-w)/(w'-w)$. The "r"s and "t"s in above equations are the entered complex amplitudes or the square roots of the real energy coefficients. "f" is a normalized angular amplitude:

$$f(i, a) = \sqrt{\text{BSDF}_i(a, a) / \text{BSDF}_i(0, 0)}$$

```

--- COATINGS [ k ] LAYERS [ w [ a ] ]
--- d m [ d' m' d" m" ... ] [ 'name' ]
:
```

Starting with coating "k", coatings with layer thicknesses "d" and MEDIA "m" (numbers or names) are entered. The default value for "k" is one more than the largest coating number defined and is set to one at the start of program execution. Normally, the thicknesses are in absolute WAVELENGTH UNITS. However, if a reference wavelength "w" is specified, then the thicknesses are fractional optical waves in the particular media at that vacuum wavelength and angle of incidence "a" (default is 0 degrees, i.e. normal incidence).

Since the MEDIA command can define complex refractive indices at several wavelengths, the effects of absorption and dispersion in each layer can be handled precisely. Groups of layers can be replicated any number of times by following a set of layer definitions with a negative "d" for the number of immediately previous layers to replicate and "m" for the number of times the layers are to be replicated. For example, a typical Fabry-Perot filter is constructed from quarterwave layers of Zinc Sulfide (H) and Cryolite (L) as follows:

```

                    5      5
                glass/(HL) HH (LH) /glass

! define materials
MEDIA
  1.35 =CRYOLITE
  1.52 =GLASS
  2.35 =ZNS
! define layers in terms of waves of optical thickness
COATING LAYERS 700 nm
  1/4 ZNS 1/4 CRY -2 5, 1/2 ZNS, 1/4 CRY 1/4 ZNS -2 5
! assign to interface sandwiched between two glass surfaces
:
OBJECT
  INTERFACE COATING .1 GLASS GLASS

```

3.5.5 Directional Scatter Models

ASAP can handle a very general set of Bi-directional Scattering Distribution Functions (BSDF). A particular scatter MODEL can then be assigned to an OBJECT (or set of objects) via the SCATTER command.

```
--- MODELS [ i [ PLOT [ a a' ... ] ] ]
```

Begin defining scatter MODELS at number "i" (default is one more than highest defined so far). The PLOT option is described below.

3.5.5.1 Standard Options For Scatter Model Input -

These options can be used on any model definition and only apply to that particular model.

```
--- ... [ PLOT [ a a' ... ] [ OVERLAY ] ]
```

Plot the model's BSDF for up to 7 specular angles in ascending order (default

0,15,30,45,60,75,89.9 degrees). A unit reflectivity surface in the current IMMERSE medium and at the current WAVELENGTH is assumed when a model is plotted. The current PIXELS setting controls the resolution of these plots in direction cosine space. For the default angles and a model sharply peaked at specular, the optimal setting for PIXELS is 580. Optionally, the multiple graphs plotted can be OVERLAYed into separate quadrants. If the angles "a a' ..." are explicitly given, PLOT will also create a full-hemispherical 2D distribution file "name_angle.DIS" for each of the angles. Otherwise, a 3D distribution file "name.DIS" is created containing the hemispherical data for all the default specular angles.

```
--- ... [ MINMAX b b' ]
```

Set the minimum "b" (floor) and maximum "b'" (ceiling) values for the BSDF.

3.5.5.2 Direct Data Entry Options And Format -

Several BSDF models allowing fitting to or interpolation from actual data. They all use the following common options and format:

```
--- ... [ ANGLES ] [ LOG ] [ COS ]
--- [ ao bo ]           !first specular direction (defaults to output)
--- a b f [ a' b' f' ...] !triplets of output direction and BSDF
--- :
--- ao' bo'            !second specular direction
--- :                  !more triplets
```

The "a"s and "b"s are either direction cosine space coordinates or the spherical ANGLES from and around normal, respectively. The "f"s are either actual BSDF (times scatter angle COSine) values or the common (base 10) LOG of the BSDF (times scatter angle COSine). Except for the RAWDATA model, BSDF values below 1.E-9 (-9 LOG) are ignored.

3.5.5.3 Isotropic-surface Scattering -

3.5.5.3.1 Simple Specific Models -

--- LAMBERTIAN t

"t" is the total hemispherical diffuse scatter ratio (less than 1). The BSDF will be independent of wavelength, incidence direction, and scatter direction.

--- HARVEY b s [l [m [n [w]]]]

"b" is the BSDF at .01 radians (.573 degrees) from specular and "s" the asymptotic fall-off with angle (typically between -1 to -2.5). If the optional B-Bo shoulder point "l" (in radians) is given, "b" is the maximum BSDF (at specular). These parameters describe a shift-invariant generalized Lorentzian function of B-Bo that normally fits the scatter from smooth surfaces (RMS much less than wavelength) extremely well. "m" and "n" are the two additional invariance parameters postulated for rough surfaces, i.e. if

$$\text{BSDF} = b f^s\{(B-B_0)/l\} \quad f^s\{x\} = \sqrt{1+x^2}$$

for m=0 and n=0 (the default), then

$$g \text{ BSDF} = b f^m\{(B-B_0)/(lg^n)\} \quad g = (C+Co)/2$$

where B,C are the sine,cosine of the scatter angle from normal and Bo,Co are the sine,cosine of the specular angle. For typical rough surfaces, "m" is approximately 2 and "n" around 1. "w" is the wavelength (in current or eventual wavelength units) at which this model is defined (or was measured). The default is the current value from the last WAVELENGTH command. If it is greater than zero, then any ray whose wavelength is different will automatically have its scatter scaled according to the smooth surface approximation.

3.5.5.3.2 General Mathematical Models For Isotropic-surface Scattering -

Scattering from an isotropic surface must be symmetric with regard to the plane of incidence and surface normal, i.e. that scatter pattern does not change when the surface element is rotated about its normal. If the BSDF is only a function of the following variables, then these symmetry properties are guaranteed.

$$U = A^2 + B^2 < 1, \quad V = B \cdot B_0, \quad W = B_0^2 < 1$$

where (A,B)=(sinR sinF , cosR sinF) are the direction cosine coordinates of the scatter direction and (0,B₀)=(0, sinF₀) is the specular direction (R,F are the scatter angles around and from normal respectively, and F₀ is the specular angle from normal). Note that the "distance" squared (in direction cosines) from specular is given by:

$$T = A^2 + (B - B_0)^2 = U - 2V + W < 4$$

Any physical BSDF must obey reciprocity, i.e. be the same when the scatter (U) and specular (W) directions are interchanged. Therefore, the BSDF's dependence on U should be identical to its dependence on W. Unfortunately due to errors, there are a lot of models and even measured data that violate reciprocity.

3.5.5.3.2.1 Linear In Free Parameters -

```

--- POLYNOMIAL n m [ l [ l' [ d ] ] ]
    TRINOMIAL
    BINOMIAL
--- c [ c' c" ... ]

```

The scattering is modeled by a general polynomial of 2 or 3 symmetry variables with coefficients "c ..." (up to 286), i.e.

POLYNOMIAL:

$$\log(\text{BSDF}) = \sum_{k=0}^n \left\{ \sum_{i=0}^{m[-k]} \sum_{j=0}^{i} c_{ijk} (U^i W^j + U^j W^i) + \sum_{i=l'}^l c_{ik} \log(1+d T^i) \right\} V^k / 2$$

TRINOMIAL:

$$\log(\text{BSDF}) = \sum_{k=0}^n \left\{ \sum_{i=0}^{m[-k]} \sum_{j=0}^{i-j} c_{ijk} U^i V^j + .5 \sum_{i=l'}^l c_{ik} \log(1+d T^i) \right\} W^k$$

BINOMIAL:

$$\log(\text{BSDF}) = \sum_{k=0}^n \left\{ \sum_{i=0}^{m[-k]} c_{ik} \sqrt{T}^i + .5 \sum_{i=l'}^l c_{ik} \log(1+d T^i) \right\} V^k$$

If m is entered as a negative number, then the upper limit of the second sum becomes $|m| - k$. The coefficients are entered in the order in which they appear in the above equation for the given " $n \ m \ l \ l'$ ". Note that if " d " is zero, then all the Lorentzian-like (log specular) terms vanish.

The user can optionally FIT (up to a finite number if using the SVD algorithm, unlimited otherwise) BSDF values to the above model by entering the data on successive commands:

```

--- POLYNOMIAL n m [ l [ l' [ d ] ] ] FIT [ k ] [ options ... ]
    TRINOMIAL                                     SVD
    BINOMIAL
--- data ...
--- :
```

Optionally, only every " k "th data value can be used. The restricted variables used in the BINOMIAL form and the term symmetry in the POLYNOMIAL form assure that the resulting BSDF exactly obeys reciprocity. The TRINOMIAL form is not by definition reciprocal but during a FIT it replicates each input point by interchanging the specular and scatter directions. This in effect tries to force the resulting BSDF to obey reciprocity.

If the peak of the BSDF remains nearly centered on all specular directions (as is the case for relatively smooth or glossy surfaces) or only in-plane data is

available, then try the following Lorentzian only fit:

```
POLYNOMIAL n 0 14 -4 2. FIT
TRINOMIAL
BINOMIAL
```

where "n" is less than or equal to the number of non-normal incident directions in the data. This will generate a model with $20(n+1)$ coefficients. So make sure there are at least as many BSDF values (a few times more is better).

Another more general technique would be to use \$ITER to do a "regressive" fit.

```
$ITER N 1 5 -5, M 1 5 -5, E
{ MODEL 1
  BINOMIAL INT[N] INT[M] SVD
  :
  data
  :
  RETURN
  $GRAB 'RMS=' E }
```

3.5.5.3.2.2 Nonlinear In Free Parameters -

```
--- NONLINEAR a b c d e [ a' b' c' d' e' [ a" b" c" d" e" ... ] ] ]
```

Generalizes the combination of the Harvey (sharp peak) and Phong (broad peak) models and as such is applicable to both smooth and rough surfaces. It is defined by the following relatively simple formula:

$$\text{BSDF} = \sum_{i=1}^N \left\{ a_i (U+W) + b_i V + c_i \sqrt{[(1-U)(1-W)]} + d_i \right\} e_i$$

where from the previous definition of the isotropic-surface direction cosine variables:

$$\sqrt{(1-U)(1-W)} = \cos F \cos F_o \quad R = \text{Angle around normal}$$

The "e" exponents do not have to be integer or positive. Note that the resulting BSDF is guaranteed to have all the correct positivity, symmetry, and reciprocity properties. The total number of parameters must be less than or equal to 285, i.e. "N" less than or equal to 57 quintuples (five or less terms is usually sufficient for most surfaces). The specific cases of this model are:

a>>1	b=-2a	c=0	d>0	e<0	Forward Harvey
a>>1	b=2a	c=0	d>0	e<0	Retro Harvey
a=0	b=0	c=0	d>0	e=1	Lambertian
a=0	b=c	c>0	d=0	e>1	Forward Phong
a=0	b=-c	c>0	d=0	e>1	Retro Phong

The user can optionally fit BSDF values to the above model by entering the data on successive commands:

```

--- NONLINEAR FIT  [ n      ] [ options ... ]
                   FRAC    e e' ...
--- data ...
--- :

```

"n" is the number of 5-parameter terms (default 5) or optionally the "e"s are the starting guesses for the exponents of each term. Since this model is not defined in logarithm space (like POLYNOMIAL), the FIT may have a tough time accurately reproducing any BSDF with a high dynamic range. Optionally, the FRACTIONAL error at each data point can be used instead of the absolute error. This has about the same effect as fitting in logarithm space. In either case, the fit is done using an iterative non-linear damped least-squares algorithm. Therefore, it will converge to one of the local minima and not necessarily the best. An off-line time-consuming global optimization technique (e.g. simulated annealing) could be used if the data only needs to be fit once.

3.5.5.3.3 User Definable Isotropic-surface Scatter Model -

```
--- USERBSDF [ fcn ] c c' c" ...
```

Up to 63 coefficients "c" can be used by the predefined \$FCN function or up to 286 coefficients can be passed to the USERBSDF Fortran function. If "fcn" is specified, then the incident vacuum wavelength and the 3 isotropic-surface symmetry variables "U V W" are passed in the "_0 _1 _2 _3" registers, the 63 input coefficients "c c' ..." in "_4 _5 ... _66". The last entry on the \$FCN definition is returned as the actual BSDF value.

For example, a BSDF that varies as the cosine of the angle from specular raised to some arbitrary power (Phong-like model) could be implemented with the following:

```
$FCN FONG C=_2+SQRT((1-_1)*(1-_3)), !cosine of angle from specular
      _4*((C>0)^_5)+_6
MODEL; USERBSDF FONG .3 10 .1
```

Or, the simplest BSDF model with constant TIS (total integrated scatter) that is Lambertian at normal incidence but becomes more and more specular as it approaches grazing incidence:

```
$FCN ROUGH _4*(1+_2)/3.1416, !_4 will be the TIS
MODEL; USERBSDF ROUGH .9 !90% TIS
```

Note that both these examples as defined obey reciprocity (same result if _1 and _3 are interchanged). Another example of USERBSDF would be the GUERAPV standard model:

```
$FCN GUERAPV C=SQRT(1-_1) A=ACOS(_2+C*SQRT(1-_3)),
      A1=_4 B1=_5 C1=_6 D1=_7 A2=_8 B2=_9 C2=_10 D2=_11 D=_12,
      (A1*EXP(-B1*A)*COS(C1*A+D1)+A2*EXP(-B2*A)*COS(C2*A+D2))/C+D/3.1416
MODEL; USERBSDF GUERAPV ...
```

Note that this is an unphysical model that does not obey reciprocity, cannot have the proper directional dependency for smooth surfaces, and could even become negative.

Besides a \$FCN definition, the user can program the USERBSDF routine to return the BSDF as a function of the wavelength and isotropic-surface scattering variables. The default definition for USERBSDF is the crude APART model for

scattering from a random collection of homogeneous spherical particles where:

c = particle radius divided by wavelength

c' = fractional area obscured by particles

c'' = real refractive index of particle (default=100, i.e. opaque)

Enter a negative index value for particles on a reflecting surface.

3.5.5.3.4 Models Based On Physical Properties -

3.5.5.3.4.1 Physical Optics Models Of Surface With Random Height Variations

```
--- RMS r l s
```

"r" is the RMS height variation of the surface in WAVELENGTH units. "l" is the autocorrelation length of the surface variations (again in WAVELENGTH units). The B-Bo shoulder point is then the ray's wavelength divided by "l". "s" is the asymptotic fall-off with angle from specular (0=Lambertian, otherwise typically between -1 and -2.5). This is an approximate (violates reciprocity) theoretical model primarily for smooth surfaces ("r" much less than a wavelength and "l" much greater than a wavelength) but is well-behaved otherwise (although it may not represent any actual rough surface). The BSDF will obviously vary with wavelength, incidence direction, scatter direction, and the specific object's INTERFACE properties. It will also automatically conserve energy (within standard statistical variations) if a "ROUGHNESS r l" command is also assigned to the object.

```
--- PHYSICAL s l [ r ] [ GAUSSIAN ]
      coat
```

A comprehensive physical reflection model that is "good" even for rough surfaces at grazing incidence. Defined in terms of the statistical properties of the surface height variation, i.e. RMS roughness "s" and autocorrelation length "l" in WAVELENGTH UNITS for either a fractal or GAUSSIAN Power Spectral Density Function. Also requires the normal incidence ideal reflectivity "r" or polychromatic COATING name (default is the particular objects specular

reflectivity).

3.5.5.3.4.2 A Random Distribution Of Spherical Particles -

```

--- PARTICLES g q['p] f
    VOLUME    MIE m a a' f [ fcn n c c' ... ]
                r s

```

The surface/volume scatters as if it had a uniform random distribution of particles on/in it. The 3-entry short form is based on the simple Henyey-Greenstein model where "g" is a directional factor between -1 (perfect backscatter) to 1 (perfect forward scatter) exclusive. A value of zero corresponds to isotropic scatter. "q" and "p" are the scattering and absorption efficiencies per particle (typically near or less than 1). For PARTICLES on a surface, "f" is the overall fractional area obscuration at normal incidence and is equivalent to the number of particles per unit area times the average particle cross sectional area. Therefore, it is usually a small number much less than one. For particles in a VOLUME, "f" is the overall fractional area obscuration per unit length and is equivalent to the number of particles per unit volume times the average particle cross sectional area. Whether the scattering is from actual real particles or something else, only the product of "q+p" and "f" really matters since this is just equal to the standard "extinction" coefficient ("q" times "f" is inversely proportional to the mean free path length).

Two other models for sphere scattering are available. One is an exact time-consuming MIE calculation with particle refractive indices specified by MEDIA "m". The other is a fast approximation for large (radii much greater than a wavelength) opaque (indices much greater than one) rough ("white" Gaussian statistics) spheres with intrinsic surface reflectivity "r" and RMS surface roughness "s" (relative to the sphere radius). In both cases, the particle radii lie between "a a'" (in wavelength units) corresponding to the one over e-squared points of a default Gaussian normal size distribution. Optionally, any size distribution function "fcn" can be defined via a previous \$FCN command. The main "_" argument of the function is a normalized size between the integration limits -1 to 1, i.e.

$$\frac{r - (a + a')/2}{(a' - a)/2}$$

where "r" is the actual size (again in wavelength units). "n" is the number of integration samples to use in the given size range (enter zero for a minimal default). The additional (and optional) "c c' ..." command parameters are passed in the "_1 _2 ..." registers (up to 66). For example, an equivalent to the default distribution would be simply defined as follows:

```
$FCN E2GAUS EXP(-2*_^2)
```

Also, if the distribution function is known in terms of actual size "R", then just define it in terms of the normalized radius "_", e.g.

```
$FCN PSDIST R=_*( _2-_1)/2+( _1+_2)/2 ...
MODEL
PARTICLE MIE m a a' f PSDIST 0 a a' ...
```

3.5.5.3.4.3 Rough Reflective Surface Of Random V-cavities -

```
--- VCAVITY s r r'
      -k -h
```

Scatter from a rough surface geometrically modeled as a random collection of v-cavities. It includes the effects of shadowing and multiple reflections within a cavity. The slopes of the cavities are assumed to follow a Gaussian-normal probability distribution with zero mean and RMS slope "s" (in radians). "r" is the intrinsic normal-incidence specular reflectivity of the material. "r'" is the corresponding Lambertian diffuse reflectivity (i.e. the sum of these reflectivities should be equal to one minus the material absorption). Alternatively (for polychromatic effects), the specular reflectivity can be determined from COATING "k" (in vacuum/air) and the diffuse from the RMS height roughness "h" (in wavelength units).

3.5.5.3.5 Indirectly Interpolating From Entered BSDF Data -

```
--- BSDFDATA [ n ] [ d ] [ options ... ]
    RAWDATA
--- data ...
--- :
```

Creates a model that indirectly interpolates from the entered BSDF data. The four-dimensional direction space is collapsed to just "n" (default 2) dimensions that enforce isotropic-surface symmetry. "d" is a small directional distance (default .001 radians for BSDFDATA, .000001 for RAWDATA) used to determine if two input data points have collapsed to the same "n"D point. The BSDF or log(BSDF) (RAWDATA or BSDFDATA, respectively) is then interpolated as a function of these "n" directional coordinates as follows:

n	Coordinates	Interpolation
-----+-----+-----		
1	sqrt(T)	Linear
2	sqrt(T),V	Bilinear
3	U,V,W	Nearest point or reciprocal point

Therefore, if the entered data is not from an isotropic surface or if it violates reciprocity, then the interpolation may not match the original data very well (i.e. it may be noisy). One last note, this model uses an internal 10000 point buffer. If the final number of data points the model needs to store exceeds this, then a disk file is used and runtimes can increase dramatically. In this case, use a larger value of "d" to further collapse the input data to fewer than 10000 final data points.

3.5.5.4 Anisotropic-surface Scattering -

Scattering from anisotropic surfaces (e.g. brushed, diamond-turned) is not rotationally symmetric at normal incidence and not necessarily symmetric about the plane of incidence otherwise. Therefore, the orientation of the model on the surface is important and is generically specified by an axis for the second command entry:

```

--- model X ...
      Y
      Z
      U
      V
      W
      R
      T
      A
    -X
      :

```

where:

Entry Anisotropic Axis

```

-----+-----
X Y Z    Global coordinate direction
U V W    CURVE object parametric direction or local x,y,z of SURFACE object
R T A    Radial, angular, or axial direction for LENS, SWEEP AXIS, or
          PARAMETERized/LOCALized SURFACE object

```

The local "Alpha" direction is perpendicular to the given axis and the surface normal. The "Beta" direction is then perpendicular to the "Alpha" direction and the surface normal. The local direction cosines are the projections of the global direction vector onto these locally orthogonal directions.

Direction	Components	Magnitude
-----+-----		
		$\sqrt{A^2 + B^2}$
Scatter	A, B	$\sqrt{A^2 + B^2} = \sin(\text{scatter angle from normal})$
		$\sqrt{A_o^2 + B_o^2}$
Specular	Ao,Bo	$\sqrt{A_o^2 + B_o^2} = \sin(\text{specular angle from normal})$

3.5.5.4.1 Effective Surface Scattering Of Vane Structure -

```

--- VANES X a s d t [ t' [ c [ w ] ] ] [ EDGES r [ t' ] ] [ DIFFRACT [ h n ] ]
Y
Z
:
```

The surface scatters as if it is the locus of vane tips parallel to the local "Alpha" direction. The angle of the vanes is "a" in degrees from the surface (i.e. 0 to 180). Their separation is "s" and their depth is "d". The vane cavity surfaces are assumed to be Lambertian with TIS values of "t" and "t'" for the sides and bottom, respectively. "c" is an optional cylindrical radius of curvature of the vaned surface and "w" is an optional vane cavity width (defaults for both are zero, i.e. infinite).

The vane tips may have rounded EDGES of radius "r" and the given Lambertian TIS. They may also DIFFRACT as if they are optionally serrated with "h" the half-angle in degrees of each vertex and "n" the number of vertices per unit length (i.e. two over the serration period).

Note: This is currently an approximate model that ignores the scatter between the sides and the bottom of the cavity. Therefore, under certain circumstances it may underestimate the effective BRDF.

3.5.5.4.2 Elliptical Harvey Model -

```

--- HARVEY X b s l l' [ w ]
Y
Z
:
```

"b" is the maximum BSDF (at specular). "s" is the asymptotic fall-off with angle (typically between -1 to -2.5). "l" and "l'" are the A-Ao and B-Bo shoulder points in radians. More precisely,

$$\text{BSDF} = b f\{(A-A_o)/l, (B-B_o)/l'\}^s \quad f\{x,y\} = \sqrt{1+x^2+y^2}$$

where A,B are the scatter direction cosines and Ao,Bo are the specular. "w" is the wavelength (in current or eventual wavelength units) at which this model is defined (or was measured). The default is the current value from the last

WAVELENGTH command. If it is greater than zero, then any ray whose wavelength is different will automatically have its scatter scaled according to the smooth surface approximation.

3.5.5.4.3 Nonlinear In Free Parameters -

```

--- NONLINEAR X p q a b c d e [ p' q' a' b' c' d' e' ... ] ]
      Y
      Z
      :

```

Generalizes the combination of the Harvey (sharp peak) and Phong (broad peak) models and as such is applicable to both smooth and rough surfaces. It assumes the surface anisotropy is aligned with the local Alpha or Beta direction and is defined by the following relatively simple formula:

$$BSDF = \sum_{i=1}^N \left\{ p_i (A_i^2 + A_o^2) + q_i (B_i^2 + B_o^2) + a_i A_i A_o + b_i B_i B_o + c_i C_i C_o + d_i \right\} e_i$$

where from the previous definition of the anisotropic-surface direction cosine variables:

$$C = \sqrt{1 - A^2 - B^2} = \cos F \quad F = \text{Scatter angle from normal}$$

$$C_o = \sqrt{1 - A_o^2 - B_o^2} = \cos F_o \quad F_o = \text{Specular " " "}$$

If the quantity in braces {} is less than zero, then the term is set to zero. The "e" exponents do not have to be integer or positive. Note that the resulting BSDF is guaranteed to have all the correct positivity and reciprocity properties. The total number of parameters must be less than or equal to 285, i.e. "N" less than or equal to 40 terms (five or less terms is usually sufficient for most surfaces). Note that when the "p"s equal the "q"s and the "a"s equal the "b"s for each term, then this model reduces to the isotropic-surface version.

The user can optionally fit BSDF values to the above model by entering the data on successive commands:

```

--- NONLINEAR FIT [ n      ] [ options ... ]
                FRAC   e e' ...
--- data ...
--- :
```

"n" is the number of 7-parameter terms (default 5) or optionally the "e"s are the starting guesses for the exponents of each term. If the given data does not cover most of the input and output hemispheres, then the fitted model can do unexpected things in the missing regions, e.g. have a TIS greater than one. Since this model is not defined in logarithm space (like POLYNOMIAL), the FIT may have a tough time accurately reproducing any BSDF with a high dynamic range. Optionally, the FRACTIONAL error at each data point can be used instead of the absolute error. This has about the same effect as fitting in logarithm space. In either case, the fit is done using an iterative non-linear damped least-squares algorithm. Therefore, it will converge to one of the local minima and not necessarily the best. An off-line time-consuming global optimization technique (e.g. simulated annealing) could be used if the data only needs to be fit once.

3.5.5.4.4 User Definable Anisotropic-surface Scatter Model -

```

--- USERBSDF X [ fcn ] c c' c" ...
                Y
                Z
                :
```

Up to 62 coefficients "c" can be used by the predefined \$FCN function or up to 286 coefficients can be passed to the USERANIS Fortran function. If "fcn" is specified, then the following registers are passed:

Registers	Data
-----+-----	
_0	Incident vacuum wavelength
_1	Scatter Alpha direction cosine
_2	Scatter Beta direction cosine
_3	Specular Alpha direction cosine
_4	Specular Beta direction cosine
_5 ... _66	c c' c" ... input entries

The last entry on the \$FCN definition is returned as the actual BSDF value.

Besides a \$FCN definition, the user can program the USERANIS routine to return the BSDF as a function of the wavelength and local direction cosine variables. The default definition for USERANIS is a simple elliptical Gaussian centered on the specular direction:

c = Peak BSDF value
 c' = Width in local Alpha direction
 c" = Width in local Beta direction

3.5.5.4.5 Directly Interpolating From Entered BSDF Data -

```

--- BSDFDATA X [ d ] [ options ... ]
    RAWDATA  Y
              Z
              :
--- data ...
--- :
```

Creates a model that directly interpolates from the entered BSDF data. "d" is a small directional distance (default .001 radians for BSDFDATA, .000001 for RAWDATA) used to determine if two input data points are reciprocal. The BSDF or log(BSDF) (RAWDATA or BSDFDATA, respectively) is then interpolated as a function of the four directional coordinates. Therefore, if the entered data violates reciprocity, then the interpolation may not match the original data very well (i.e. it may be noisy). Also, since no assumptions concerning symmetry are made, data must be provided over most of the output hemisphere for most of the input hemisphere. One last note, this model uses an internal 10000

point buffer. If the final number of data points the model needs to store exceeds this, then a disk file is used and runtimes can increase dramatically. In this case, use a larger value of "d" to further collapse the input data to fewer than 10000 final data points.

3.5.5.5 Sum Of Other Scattering Models -

```

--- SUM [ X ] i i' [ i" ... ]
      Y
      Z
      :
```

Adds the BSDFs of the given list of previously defined models to form a composite model. It is fully recursive so that one SUM model can reference another. Only one anisotropy orientation is used for all the models, either the one specified or the one for the first anisotropic model listed.

3.6 DEFINING OBJECTS

A geometry entity (SURFACE, EDGE, or LENS) must be assigned to an OBJECT for it to be "visible" to rays or beams.

```

--- OBJECT [ j ] [ 'name' ]
--- OBJECTS [ j ]
--- i [ i' ] [ 'name' ]
      :
```

Define a new OBJECT using either the last entity or begin defining (or redefining) OBJECTS starting with number "j". The default value for "j" is one more than the largest object number defined and is set to one at the start of program execution. The first entry "i" is the surface, edge, or lens number that defines the geometry of the object. Simple mesh objects can be formed by one or (if "i'" also given) two edges. "name" is a descriptive designator that the user can assign to this object.

More complex objects can use multiple edges or surfaces to form a mesh or solid, respectively.

```
--- OBJECT; i [ q i' q' ... ]  
            -n q [ q' ... ] q"
```

If "i"s refer to EDGES, then a two-dimensional mesh is formed from these edges with the edge-to-edge connection specified by the "q"s (see the POINTS command for definitions but note that the inter-curve Bezier degree cannot be greater than 2). Alternatively, the last "n" EDGES will be used to form a mesh with "q" the connection factor for odd edges, "q'" the even, and "q'" the last to first. The EDGES in the mesh should be similar, i.e. have the same number of segments.

If the "i"s refers to SURFACES, then a solid is formed by bounding each surface with the others. The signs of the "q"s determine the proper "side" of each surface (see BOUNDS command). A zero "q" means the surface is not to be used to bound the others in the object. Note that each surface's LOCAL box (if defined) always clips it.

3.6.0.1 Redefining Some Basic Options -

```
--- REDEFINE [ SURFACE j ] [ THICKNESS t ] [ COLOR k ] [ 'name' ]  
            NORMAL
```

The SURFACE option forces the program to move any intersection points on a non-lens object to a nearby continuous surface "j" after first intersecting the original object surface. "j" is also then used to calculate the normal for refraction and/or reflection. The NORMAL option allows the user to designate a continuous surface number "j" to be used for calculating just the normal to the object that is different than the actual object normal. These options are useful for modeling Fresnel optics and more accurately tracing rays/beams through discrete faceted approximations to objects.

The THICKNESS option is used to thicken the object surface(s), i.e. effectively turn it into two surfaces spaced a small distance "t" apart. The sign of "t" determines in which direction along the surface normal the second surface is located.

The COLOR option assigns a specific color (1 to 26) to the object that is used in all subsequent graphical displays unless overridden by the global COLOR command.

3.6.0.2 Controlling Subdivision Of Mesh Patches Into Facets -

```
--- FACETS  n [ n' ]
           -n  -n'
           +n  +n'
```

Sets the number of facet subdivisions in each mesh patch direction. If the entry is unsigned, then this is the number that is always used. If the entry has a plus sign, it is the maximum number of subdivisions relative to the corresponding number on a PLOT FACETS or VUFACETS command. If it has a minus sign, its magnitude is the minimum number of subdivisions per patch. Also controls the density of rays when referenced by a GRID OBJECT or EMITTING OBJECT.

3.6.0.3 Assigning Surface Roughnesses -

```
--- ROUGHNESS [ r [ l ] ] [ MODEL i [ FRACTION [ f ] ] ]
                      RANDOM h [ s [ m ] ]
```

"r" is the RMS micro-roughness of the object surface(s) in WAVELENGTH units and is used to transfer energy out of the specularly reflected / refracted beams (and usually into any scatter beams). It does not affect the optical path lengths or directions of the specular beams. If the optional autocorrelation length "l" (also in WAVELENGTH units) is given, then the effects of shadowing at high angles of incidence are included.

For a FRACTION "f" (default 1 or given MODEL T.I.S.) of the incident rays, RANDOM assigns a macroscopic random height variation "h" in WAVELENGTH units and slope error "s" in radians (both defaulted to 0) to the surface(s) of the object. The random height errors affect only the position of a point on the object (and thus the optical paths lengths of any reflected / refracted beams) while the slope errors affect only the normal (and thus the beam directions).

For near normal incidence rays, if the RMS slope error exceeds about one fifth (.2), then unexpected raytrace results may occur at the surface, i.e. "wrong side" warnings because a ray may randomly reflect or refract into the surface. For near grazing incidence rays, the RMS slope may have to be much smaller than this to avoid these raytrace errors. Both errors are by default generated according to an approximately Gaussian-normal distribution (type 2 below). However, the slope error distribution function can be any one of the following:

m	Slope distribution	(Maximum/RMS)	Equivalent
-3	Two deltas	1	RAN(-15)
-2	Lambertian		
-1	Ramp	2	RAN(-1)
0	Uniform	3	RAN(0)
1	Triangular	6	RAN(2)
2	Gaussian-like	9	RAN(3)
3	Cosine	5	RAN(1)
4	Near-Gaussian	15	RAN(5)
5	Gaussian	$2 \ln(2^{32})$	RAN(15)

To be more precise, if the unperturbed normal points along z, then the components of the randomly perturbed normal will be:

$$[s \cos(u) , s \sin(u) , \sqrt{1-s^2}]$$

$$-1 < s < 1$$
$$-\pi < u < \pi$$

where "u" is a uniformly distributed random number, and "s" is a random number determined by the given slope distribution (RMS and probability function). Note the maximum absolute slope error is always limited to one, i.e. 45 degrees.

Alternatively, the surface slopes can be randomized such that the shape of the normal incidence reflected pattern matches that of the BRDF specified by scattering MODEL "i".

3.6.0.4 Adding Small Deformations To An Object Surface -

```
--- DEFORM k [ k' ]
```

With this short format, one or two general deformation functions defined entirely by the given explicit surface functions (GENERAL EXPLICIT, FITTED EXPLICIT, ZERNIKE, SAMPLED, ...EXPLICIT or USERSAG) are added to the object surface.

```
--- DEFORM x y z a [ a' a" ... ] [ AXIS a,b,c ] [ FCN fcn ]
```

With this long format, a small user-definable aspheric deformation is added to the previous object. This deformation is rotationally symmetric about the axis defined by either the normal vector that passes through the point "x,y,z" on the object's surface or the given point and AXIS. The "a"s are the up to 20 aspheric deformation coefficients. The deformation or sag value as a function of perpendicular distance from the aspheric axis (r below) is given by:

$$\text{sag} = \text{fcn} \left(ar + a' r^2 + a'' r^3 + \dots \right)$$

where "fcn" is the name of an optional macro function (intrinsic, e.g. SIN, or user-defined \$FCN).

3.6.0.5 Bounding (and Slicing) Of Objects -

The LIMITS command can be used for very simple bounding of an object. For more exotic surface boundaries, BOUND surfaces or edges can also be employed.

3.6.0.5.1 Rough Orthogonal Bounds -


```

--- LIMITS x x' [ y y' z z' ] [ X [ f ] ]
                                Y
                                Z
                                SHORTEST
                                LONGEST
                                NORMAL

```

With the LIMITS command, the previous object can be crudely bounded by the specification of allowable ranges on each of the global coordinates. The unprimed entries are the lower bounds, primed the upper. If a coordinate direction is also given, then the object is constrained by a cylinder of constant elliptical (as opposed to rectangular) cross-section in planes perpendicular to the given coordinate. This axis can also be determined by the SHORTEST limit box dimension, the LONGEST, or the coordinate direction nearest to the surface NORMAL. An optional inner boundary of fractional height "f" [default is zero] can be used to put a proportional hole in the object in this direction.

```

--- LIMITS [ OFF      ]
      REPEAT i [ X [ f ] ]
      STATS      Y
      AXIS       Z
      -X x       SHORTEST
      -Y y       LONGEST
      -Z z       NORMAL
      +X x'      OFF
      +Y y'
      +Z z'
      EXPAND r
            X r
            Y
            Z

```

The above short form can also be used to temporarily turn OFF future limits checking, to set the limits on the current object to those of a previous object "i", to the ranges of ray intersections found from the previous TRACE STATS command, or to reset the limits AXIS or just one of the six limits values. EXPAND can be used to enlarge (or shrink) by a relative amount "r" the entire limits box or just in one direction, e.g. "EXPAND -.1" shrinks the entire box by 10%.

Any linear transformation commands that follow an object definition are applied

to the object surface and all boundary surfaces, i.e. the entire object. The coordinate ranges entered with the LIMITS command are also adjusted so that the object lies entirely in a new 3-dimensional orthogonal box. Therefore, the shape of the object can change with a ROTATE or SKEW if only the LIMITS are used to bound it. To avoid this, use a LOCAL box on the base SURFACE of the object.

3.6.0.5.2 General Boundary Surfaces And Edges -

```
--- BOUNDS [ k [ k' ... ] [ MULTIPLE m [ m' ... ] ] ] [ POINT x y z ]  
RBOUNDS OFF
```

The "k"s are the numbers of the surface and /or edge entities that are to be used to define the boundaries of the object surface. If these numbers are positive, then the object is assumed to be on the positive side of the corresponding surface, outside the surface's SOLID or outside the closed edge. A negative number means that the object is on the surface's negative side, inside the surface's SOLID or inside the edge. The MULTIPLE option makes a group of surfaces /edges equivalent to one standard constraint, i.e. a point that is valid for one or more of the given "m" bounds is a valid object point. To efficiently enter a very large number of boundaries, use instead the RBOUNDS version where every pair of "k"s or "m"s specifies a range of entity numbers instead of individual entities.

The POINT option switches the signs of the entities if necessary in order to make "x y z" a valid point on the object. A BOUNDS OFF command can be used to temporarily turn off all boundary surfaces for the current object. A BOUNDS command by itself will turn them back on.

3.6.0.6 Assigning Optical Properties To Object Interfaces -

If the object surface is an optical boundary through which rays are to be traced, the optical properties of the interface must be specified using the following command after the definition command for that object.

```

--- INTERFACE r [ t m m' ] ...
      COAT k
      -k
      +k

```

"r" is the relative energy (or complex amplitude) reflection coefficient for the surface (or the reflectors in a lens structure). "t" is the transmission coefficient of the surface (or the refractors in a lens structure, their reflectivities are then assumed to be 1-t). Optionally these coefficients can be determined from COATING number or name "k" (0 for a BARE SUBSTRATE) and the current WAVELENGTH. The signs in front of "k" can be used to suppress either the reflected component (+k: r=0) or the transmitted component (-k: t=0). If "t" is nonzero, then "m" and "m'" are the numbers (or names) of the media on each side of the object surface. If an INTERFACE command does not follow an object definition, the surface is assumed to be perfectly absorbing and all rays reaching the surface will be trapped there.

```

--- INTERFACE ... DIFFRACT i j [ e j' e' ... ]
      coat coat'

```

DIFFRACTive lines are created by the intersection of the object surface with the different sheets of a MULTIPLE surface "i", e.g. a ruled linear grating is created if "i" is a plane, a zone plate if the surface is a cylinder, etc. If "i" is positive, the MULTIPLE sheet spacing is taken to be the grating spacing in system units. If "i" is negative, the spacing is assumed to be in the same units as the last WAVELENGTH specification. The number of sheets entered on the MULTIPLE surface command has no bearing on this application. The program will generate diffracted rays/beams for the diffraction order numbers given by the "j"s with relative efficiencies given by the corresponding positive "e"s. If an "e" is entered as a negative number or as a name then it is a COATING PROPERTY possibly containing polychromatic complex amplitudes. Also, a named COATING MODEL can be used to specify the angular variation of the diffraction order efficiencies, i.e.

```

--- INTERFACE ... DIFFRACT i coat j j' j" ...

```

Multiple exposure holograms can be modeled by using more than one DIFFRACT option (the zeroth order should only be specified once).

```

--- INTERFACE ... DIFFRACT i ... DIFFRACT i'... [ DIFFRACT i" ... ]

```

--- INTERFACE REPEAT [i]

An INTERFACE REPEAT command will assign the interface characteristics of object "i" (or the previous object) to the current object. If "i" is zero, then all the interface properties are removed.

3.6.0.6.1 Scatter Characteristics -

--- SCATTER REPEAT [i]

An SCATTER REPEAT command will assign the scatter characteristics of object "i" (or the previous object) to the current object. If "i" is zero, then all the scatter properties are removed.

3.6.0.6.1.1 Deterministic -

**--- SCATTER RMS b [s a]
BSDF**

An incoherent near-specular scattering component can be assigned to the interface. "b" is either the RMS micro-roughness of the surface in wavelength units or the actual BSDF in inverse steradians at 0.573 degrees from specular. "s" is the asymptotic falloff of the surface Power Spectral Density Function (also the Bi-directional Scattering Distribution Function in direction cosine space) and is typically a number between -1 and -2. If scattered ray / beam generation has been turned on by the LEVEL command, the program will generate a diverging near-specular beam at any interface with a non-zero "b" value. In addition, if the cone half-angle in radians "a" is also non-zero, the program generates a back scattered beam that propagates back along the incoming direction with this maximum divergence half-angle. The scatter in this direction is calculated using the "b s" or "m" parameters. A negative "a" suppresses the forward scatter beam so that only the backscattered beam is generated at the interface.

3.6.0.6.1.2 Random -

```

--- SCATTER [ MODEL m ] [ RANDOM [ r' n a' ] ] [ ABS r t ]
          MODELS m m'                                rel

```

A general scattering MODEL "m" can also be assigned to the interface. If the reflective "m" and transmissive "m'" scatter MODELS are different, specify both. While the LEVEL command is in effect, the RANDOM option causes the program to generate a set of randomly distributed diffuse scatter rays for each incoming ray on the object. "r'" is the additional total hemispherical diffuse reflectivity of the scatter (default=1, i.e. white Lambertian). "n" is the number of rays to be scattered into the hemisphere centered on the ray point

If both reflective and transmissive (or multiple diffraction orders) scatter rays will be generated (see LEVEL ALL command), the specified scatter can be separately scaled in magnitude for the two components by the given "r" and "t" factors. The scatter specified can also be either ABSOLUTE (referenced to the incident power) or RELATIVE to the specular power. The default is "ABS 1 1" (except for SCATTER RMS which is "REL 1 1").

```

--- SCATTER MODEL m
          MODELS m m'
--- TOWARDS [ EDGE ] i      [ n' f t a" [ REL] ]
          POINT
              X
              Y
              Z
          -X
          -Y
          -Z
          SPEC
          REFL
          TRAN
--- [ TOWARDS ... ]
      :
```

If the MODEL and/or RANDOM options are used, more diffuse scatter rays directed TOWARDS "important" areas can be specified with an additional lines of input. The "n" entries are the number of random rays per important area or direction. The "a"" entries are the divergence half-angles (in radians) to be used for these rays (default is 1.5708 when PARABASALs 0, .1 otherwise) and are either absolute or RELative to the cone angle of the importance area.

If the second entry is EDGE or a number, then absolute value of the "i" entries are the numbers of the edges to used as important areas. If "i" is entered as a positive number, radiation is scattered towards the designated real area. If "i" is negative, then radiation is scattered away from the designated virtual area. "f" and "t" are the fractional lower and upper bounds (default 0 to 1) of the importance area relative to the defining edge. The program tries to take into account "important" areas whose solid angles overlap in order to maintain conservation of energy.

Instead of defining an important area with a polygon (EDGE), an important direction can specified as lying between the surface normal and the following:

Entry	Description
-----+-----	
X,Y,Z, -X, -Y, -Z	Global coordinate axis direction
REFL, TRAN, SPEC	Reflected, transmitted, or either specular direction
POINT i	Direction from ray point to entity "i" reference point

In these cases, "f" is the cone half-angle in radians of the solid angle centered on the direction. "t" is a parameter that varies the scattering direction from the surface normal (t=0) to the specified direction (t=1, default). A value of t=-1 corresponds to the anti-direction (e.g. retroREFlection).

3.6.0.7 Refraction/reflection Controls -

The following override the defaults for the current object.

```
--- FRESNEL [ TIR ]
    OFF
    AVE
    S
    P
    BOTH
    k
```

Signals the program to take into account the variation of the reflection and transmission coefficients with incidence angle at the object's interface and

reduce the ray flux accordingly. The "k" parameter or equivalent literal entry can be used to control which Fresnel coefficient to use in the refraction calculation, i.e.

Literal Integer k		
-----+-----+-----		
TIR	-2	Allow TIR rays but no angular flux variation for others
OFF	-1	Use normal incidence properties for all rays (default)
AVE	0	Incoherent average of polarizations
S	1	S polarization coefficient only
P	2	P polarization only
BOTH	3	Both separately

```
--- SPLIT [ n ] [ MONTECARLO ] [ c ]
      TRAN
      REFL
      NORM
      OFF
```

"n" is the maximum number times a ray can be split into specular components (OFF resets n to zero). When greater than zero, rays that encounter this object with non-zero reflection AND transmission coefficients will be split into separate reflected and transmitted rays. The most energetic at NORMal incidence, the TRANSmitted, or the REFLected component continues to be traced while the other component is added to the top of ray storage and is eventually traced at a later time. NORM is the default.

With the MONTECARLO option, no additional specular and/or scattered rays are actually created, the new parent is probabilistically selected from the possible child rays. Otherwise, for a modest number of initial rays and a refractive/scatter object, the total number of rays eventually traced can be quite large. Even though the program has virtual ray storage, one is practically limited by the total free disk storage and long run times. Therefore, the default deterministic splitting should be used with due appreciation to its possible consequences. For a typical application such as ghost image analysis, a value of 2 for "n" is sufficient. If the fractional energy in the split-off ray/beam is below "c" (default 1.E-6), splitting does not occur, i.e. only the main component is propagated.

To control the level of scattered component splitting, use the following parallel command:

```
--- LEVEL [ n ] [ OFF ] [ c ]  
      ALL
```

"n" is the maximum number of scattering levels allowed for random diffuse rays. Deterministic near-specular and back-scattered beams are never rescattered. If the fractional scattered energy (relative to the incident ray) drops below "c" (default 1.E-12), the scattered ray is not generated.

Normally, scattered rays are not generated for any specular child rays, only the parent. If ALL is specified, scattered rays are generated for each specular child ray (including multiple diffraction orders, extraordinary rays, etc.), so that "bi-directional" scatter can occur at a partially reflecting/transmitting interface.

```
--- HALT [ n ] [ OFF ] [ c ]  
      +-X  
      Y  
      Z
```

Forces the tracing of certain rays to be halted. "n" is the maximum number of times a ray can consecutively interact with this non-LENS object. The literal entry picks out a certain coordinate direction or turns it OFF. If a ray attains a component in this direction after reflection or refraction or the number of consecutive interactions with this non-LENS object reaches the integer "n" (default 12), then tracing halts for that particular ray. Tracing of a ray whose ratio of current flux to initial flux falls below the decimal number "c" (defaulted to 1.E-6) at this object will also be stopped.

3.6.0.8 Streamlined Entity And Object Input -


```

--- ENTITIES [ OBJECTS ]
---   lens ... [ 'name' ]
      edge
      surf
---   entity modifiers (e.g. LOCAL,SWEEP)
---   [ object modifiers (e.g. INTERFACE,BOUNDS) ]
---   lens' ...
      edge'
      surf'
---   :
```

Begin defining mixed geometrical entities without using the EDGE/CURVE, LENS, or SURFACE/FUNCTION commands. Optionally, make each subsequently defined entity an OBJECT automatically without using the OBJECT command. Obviously, this means that only single entity objects can be defined in this manner (although BOUNDS can still be applied if the bounding entities are defined in the normal manner before the ENT OBJECTS command). Also, note that all entity modifiers must precede any object modifiers.

3.6.0.9 Moving Groups Of Objects And Sources As A Whole -

```

--- GROUP [ i [ i' ... ] ] [ SOURCES j [ j' ... ] ]
          -n                      -n
          0
```

This command temporarily declares the objects/sources whose numbers (or object names possibly with "?" wildcards) are entered on it, the last "n" objects/sources, the last group (0), or all objects (no entries) as belonging to the same "group". The SOURCES option can only be abbreviated to SOURCE in order to avoid confusion with an object name. Any linear transformation commands that immediately follow this command are applied to every object and source in the group. In addition, the CONSIDER command can reference the last (current) group of objects by default.

A few specific examples are:

```
GROUP      !all objects
GROUP ?    !all objects
GROUP -10000 !all objects
GROUP .?   !objects in same branch as last object
```

3.6.0.10 Exploding Out Lenses Into Separate Surface Or Curve Based Objects

```
--- EXPLODE [ l ] [ macro ]
      +l
      -l
      ALL
      +ALL
      -ALL
```

Explodes the "l"th (default last) LENS or ALL lens entities into separate SURFACE-based objects (or CURVE-based objects if the current macro library is the standard CADEQUIV.LIB). If the entry is preceded by a sign, then additional objects are created that represent mirror backs and the baffle, mounting, or edge surfaces that connect each coaxial lens surface. Use "-" for direct sloped cones or "+" for right cylinders. The resulting input commands either go into the file "macro.MAC" or are immediately executed from an internal buffer. The interfaces for the surfaces are preset as follows:

Refractive	Last primarily transmissive COATING PROPERTY or BARE
Reflective	Last primarily reflective COATING PROPERTY or unit reflectivity
Back/Edges	Totally absorbing

3.6.0.11 Specifying A Branch Of The Name Tree For Future Objects -

```
--- BRANCH [ name ]
      .name
      ^
      ^.name
```

Specify an absolute or relative branch of the object name tree for future object definitions and display the current or new full branch name. Periods "." are used as node separators in the name. Carets "^" are used to move up one node. If the entry begins with either of these two characters, then the branch is defined relative to the current branch. Otherwise, it is an absolute complete branch specification (with possibly many embedded periods "."). The number of levels is limited by the total object name length of 344 characters. See also the TREE command.

3.7 INPUT/OUTPUT CONTROL COMMANDS

3.7.1 Controlling Execution

--- RESET

Reinitializes all control settings to those at program startup. Does not affect system description (use SYSTEM NEW to reset this).

--- DOMACROS FIRST LAST NEVER

Determines whether the user's MACRO commands are checked FIRST, LAST or NEVER relative to the top level ASAP commands. The default is LAST, but FIRST could be used if a user macro name conflicts with a ASAP command.

--- LSQFIT [t] [LIST] [NORM] [OFF]

This command controls the Singular Value Decomposition (SVD) algorithm used by any of the commands that do a least squares fit. Any singular values that are "t" (default 3.E-5) times less than the maximum are removed. If the LIST option is present, then the singular values, variable values, and fit errors are printed. Sometimes an improved fit is obtained if the variables are first NORMALized. OFF sets both of these options back to their default states.

--- FTSIZE [m]

Sets the Fourier Transform size used by future DECOMPOSE DIR, OPDMAP PSF, and FFT commands to the maximum recommended (as shown by DIMENSION command) or raised to the "m"th power. The default size at startup (or after a RESET or END) is 9, i.e. 512 samples. The absolute maximum allowed is 16 (65536 samples).

--- RETURN

Returns the user to the top most command level or if already there, terminates graphics (i.e. OVERLAY) mode and returns the user to text mode.

--- END [OFF] NOW

Immediately terminates the current session of the program and does either of two things:

- If in batch mode, returns control to the operating system
- or if interactive, executes a SYSTEM NEW and RESET.

The optional OFF entry temporarily turns off the processing of the END command. In other words, any END command following an END OFF command does not terminate the session but simply switches the input to interactive mode. This is useful for executing input files in interactive mode that also work properly in batch mode. To terminate the session under these conditions, enter an END NOW or \$EXIT command.

3.7.2 Displaying Important Program Information

```
--- PRINT [ i ]  
SPRINT SURFACE [ i ]  
QPRINT  EDGE  
        LENS  
        ALL  
        MEDIA  
        COATING  
        MODEL  
        OBJECT
```

Prints the system data for each surface/edge/lens, coating, medium, scatter model, object or just the selected information. The "i" entry prints all the information pertaining to only that entity name or number. The amount of information displayed depends on the command:

PRINT	Complete information
SPRINT	Excludes main geometry data (i.e. coefficients, conicoids, points)
QPRINT	Only one line per entity

--- DIMENSIONS

Display a table of maximum dimensions for the most important program arrays. Values listed in the form "number-1" usually indicate a limitation due to decimal encoding (either internally or for text output).

--- NUMBERS [NAMES] SUMMARY

This command displays either a short SUMMARY of storage usage or tables of the current valid SURFACE/EDGE and OBJECT numbers. Unused numbers appear as blank areas in the tables. Temporarily disabled OBJECTS (via the CONSIDER command) are shown as negative numbers. A table of numbers and NAMES for all MEDIA, COATINGS, and OBJECTS will be listed with this option.

--- SHOW [ALL]

Prints out the current status of some or ALL of those commands that set flags or data used by other commands.

--- TREE [ENTITIES] [i] ['delimiter']

Displays object name tree indented by "i" (default 2) at each level assuming they are separated by the single character "delimiter" (default is a period "."). The ENTITIES option also displays the base entities and signed BOUNDS entities for each object. The corresponding object (and entity) numbers are shown inside angle brackets < >.

3.7.3 Saving/retrieving System Data To/from A Disk File

```
--- SYSTEM NEW
      TO [ file ]
      FROM
```

Write current system data TO an unformatted binary "file", or read new system data FROM "file", or reinitialize data for NEW system input. The default for name part of "file" is taken from the last SYSTEM or \$IO command (or is assumed to be LASTEXEC). The file extension is will always be "SYS". After an END command and just before exiting, the program always writes out the system data a file LASTEXEC.SYS just in case the user forgot to save their changes. A menu of all the SYStem files in the current directory can be obtained by using the following syntax:

```
      SYSTEM FROM "*"
or SYSTEM FROM _
```

IMPORTANT NOTE: Do not use this command for archive purposes since normally the file format changes with each release.

3.7.4 Exporting A CAD Representation Of The System

```
--- CADEXPORT [ IGS [ name          ] ]
      DXF      [ block ]
      VCR
```

Writes out a 3D CAD surface representation of the currently considered objects or the 3D graphics in the current VCR file. This command supports the fixed IGES "IGS" (the default) and AutoCAD 12 "DXF" file standards (including the option to put everything into a named "block"). The file created will be "name" followed by one of these extensions. The current limitations on surface export are:

1. Advanced surface effects (REDEFINE SURF/NORM, DEFORM) and trimming information (LIMITS, BOUNDS) are not translated.
2. IGS usually requires that the system UNITS must have already been set.

3. All base surfaces in DXF are by necessity approximated by facets.
4. LENS, connected CURVE/EDGE, and SAMPLED base surfaces are exported exactly to IGES. Since no CAD standard supports general algebraic surfaces, all other SURFACES are approximated by cubic spline patches or revolved cubic spline curves when possible.
5. An often better translation to AutoCAD can be done by processing the current VCR file (usually created by PLOT FACETS or VUFACETS).

3.7.5 Limiting The Current Set Of "visible" Objects

```
--- CONSIDER [ ALL ]  
             NONE  
             ONLY  [ i i' ... ]  
             EXCEPT  
             ADD  
             REMOVE
```

This command gives the user control over the current set of objects the program is to consider in all calculations and output. The ALL option forces consideration of all known objects, NONE for no objects. Particular object numbers (or names possibly with "?" wildcards) can be either excluded with the EXCEPT option or the ONLY ones considered. If no object list is given, then the last GROUP is used. A CONSIDER command by itself (with no entries) produces a list of the currently considered objects. The current state (i.e. a previous CONSIDER command) can be updated by using ADD or REMOVE instead of ONLY or EXCEPT, respectively.

3.8 GRAPHICS CONTROL AND GENERATION

```
--- TITLE [ userid ] [ 'default_title' ]
```

Up to an 8-character "userid" literal can be displayed in the upper-left corner of each graphics screen. If a title is not supplied on a graphics generating command (e.g. a comment string on a PROFILE or PLOT), the "default_title"

specified here will be used.

3.8.1 Sampling

```
--- PIXELS n [ r ] [ ON ] [ FILL ]  
                        OFF
```

Sets the number of equally-spaced pixels across (vertically on) the WINDOW to "n" for all subsequent plots. Initially, 39 is assumed by the program. "r" is the ratio of the pixels per system unit down (horizontally on) to the number per unit across. This is initially assumed to be 1 and can be used to assure that the character plots are of the correct proportion. This command also controls the number of data pixels generated by the SPREAD (and possibly FIELD) commands (within the current WINDOW) and therefore determines the record structure of the data files they create and use. The number of "down" pixels is calculated as follows (see WINDOW command also):

$$a'' = (a' - a)/n \quad d'' = a''/r \quad m = (d' - d)/d'' = n \cdot r \cdot (d' - d)/(a' - a)$$

By default, all high resolution (non-printer) graphics are mapped into the plotting window such that the aspect of the physical entities are properly maintained. However, physically distorted views, where the entity completely FILLS the plotting window in both directions, can also be obtained.

The plotting of a pixel sampling box with some graphics commands (e.g. PROFILES) can be turned ON or OFF (the initial default) with this command.

3.8.2 Geometrical Data Window


```

--- WINDOW [ X [ a  a' ] Y [ d  d' ] ]
              OBJ i  Z  OBJ j
              Y      Z
              X
              Z      X
              Y

```

Used before most plot commands to specify the data directions and area to be plotted as "a" to "a'" across the text page (vertically on the graphics device) and "d" to "d'" down the page (horizontally on device). If either $a=a'$ or $d=d'$, then the next graphics command will attempt to autoscale the particular direction. A WINDOW command with no numerical entries on it sets $a=a'$ and $d=d'$, i.e. autoscaling in both directions. Alternately, The window size can also be set to the LIMITS range on the OBJECT numbers given.

The current (or next autoscaled) window can be modified by specifying a window magnification factor "m" and/or a relative pan factor "p" for one particular direction or both.

```

--- WINDOW [ ACROSS      ] m p  [ ACROSS      m' p' ]
            VERTICAL      CEN  VERTICAL      CEN
            DOWN
            HORIZONTAL      HORIZONTAL
            BOTH

```

ACROSS=VERTICAL and DOWN=HORIZONTAL depending on whether one is referring to the printer or plot output. As an alternative to the numeric pan factor, the program can automatically calculate a pan factor that results in CENTERing the particular data being plotted within the window.

3.8.3 Oblique (nonorthographic) Views

```

--- OBLIQUE [ f ] [ a a' ]
            OFF

```

With this command, future plots will be an OBLIQUE type projection pivoted about the third depth coordinate value "f". The default for "f" is determined by the program if autoscaling is in effect, otherwise a value of zero is

assumed. The normal orthogonal projections are restored when this is turned OFF (the default at program startup). The optional entries are the angles of rotation about the vertical "a" and horizontal "a'" axes and determine the desired oblique view. These angles are initially set at program startup to 45 and 30 degrees, respectively. Setting both these angles to zero has the same effect as the OFF option.

3.8.4 Perspective Views

```
--- VIEW [ X ] [ h ] [ CENTER x y z ] [ EYE x' y' z' ]
          Y                               DIR a,b,c
          Z
```

Make all future plots a true perspective view instead of using the current WINDOW (and OBLIQUE) settings. The view is completely determined by an up coordinate axis, a vertical full height "h" in system units, the global center of the view "x y z", and eye point "x' y' z'" (or DIRection "a,b,c" equivalent to an eye point a distance 1000*h away). The current view can be modified with the following forms:

```
--- VIEW [ OFF ] [ ZOOM f ] [ DOLLY d ] [ ORBIT o ] [ FOV a ]
          0
```

The current perspective view can be modified a variety of ways. Turn perspective viewing OFF, i.e. return to normal WINDOW (and OBLIQUE) settings. The form "VIEW 0" calculates the closest perspective view to the current WINDOW (and OBLIQUE) settings assuming an eye distance the is twice the vertical height (closely approximates a standard 35mm camera lens). ZOOM divides the current vertical height by a factor "f" (>1 zooms in, <1 zooms out). DOLLY changes the current height and eye distance by the signed fraction "d" (positive moves toward the center point, negative away). ORBIT rotates the eye point an angle "o" degrees about the up coordinate axis. FOV changes the eye distance to yield a full field-of-view equal to "a" degrees (enter zero for parallel projection).

3.8.5 Arrows On Ray And Polarization State Plots

```
--- ARROWS [ s ]
      OFF
```

The PLOT RAYS, TRACE PLOT, and PLOT POLARIZATION commands can have arrows added to their basic lines by preceding them with this command. "s" is a rescaling factor used to change the size of the arrows relative to the internal setting (default=1). The adding of arrows can also be turned OFF.

3.8.6 Number Of Segments Plotted Per Arc

```
--- SEGMENTS [ n ]
```

Some PLOT commands draw conic arcs. "n" is the number of straight line segments used to approximate a 45 degree arc (minimum=1, default=3, maximum=45).

3.8.7 Coloring Of Objects By Group And/or Interface Properties

```
--- COLORS [ i [ j [ k [ l [ m [ m' ... ] ] ] ] ] ]
```

Reassigns the color used to display each object according to the last group assignment or dominant interface property:

Color	Interface
-----+-----	
i	Group
j	Absorbing
k	Reflecting
l	Transmitting
m	Media
:	"

3.8.8 Plotting Slices Through Objects

```
--- PROFILES [ f f' n ] [ NOOPTIMIZE ] [ QUICK ]
```

Produces a set of profiles of the system. "n" is the number and "f" to "f'" is the range of the third depth coordinate (the one different than the two specified on the last WINDOW command) values for which slices through the current set of objects are plotted. If "n" is entered as a negative number, then the "f"s are ignored on objects with LIMITS and "n" equally spaced cuts are done across the individual LIMIT ranges of those objects. In the absence of any of the numeric entries, a value of -1 is assumed for "n" when OBLIQUE is off and -5 when its on.

Normally the program tries and connect up any short vectors generated by a PROFILE command. This slows down the immediate generation of the graphics but produces a "cleaner" plot file that won't drive a pen plotter crazy. Use the NOOPTIMIZE option to generate the original "raw" plot commands.

The QUICK option turns off exact boundary clipping to speed up the plot.

3.8.9 Geometrical Entities That Can Be Directly Plotted

```
--- PLOT [ thing ... ] [ thing' ... ] ...
```

The projections of the specified things in the directions given by the last WINDOW command are sent to the 2D plot device. More than one thing can now be entered on a single plot command. In addition to the 2D plot that is generated, the full 3-dimensional data is written to the VECTOR file (default logical unit 30). See MESHES for the default when nothing is specified.

EDGES, LENSES, SURFACES, LOCAL boxes, or any ENTITIES (whether they are used by an object or not) are directly plotted. The actual ones plotted is either all, the last n, just k, or k through k' incrementing by k" (default 1). CURVES is the same as EDGES except that control points are also plotted.

For LIMITS, the object limits boxes are plotted.

If BEAMS is entered, the 1/2 amplitude ellipses of each Gaussian beam described in current ray storage is plotted according to the last WINDOW command.

The POLARIZatIon state ellipses for each ray / beam are plotted. The size of the ellipses is proportional to the ray / beam's flux raised to the "e" (default 1) power.

For RAYS, a line is plotted from the current position along the direction of each ray. The actual distance plotted is "d" times the flux of the ray divided by the maximum ray flux. The default for "d" is approximately 1/10 the window size. If "d" is entered as a negative number, then only the endpoint of this vector is plotted. This is useful for producing 3D far-field intensity distribution point clouds when "d" is much larger than any inter-ray distance. Optionally, only every "i"th ray can be plotted.

For FACETS, "n n" is the minimum or maximum number of subdivisions in the two parametric directions for each "patch" of an object (see also FACETS command).

If possible, the maximum number of total facets on any object is kept below "m" (default is 1000, use 0 to turn off). Each facet will be trimmed by any BOUNDS and/or LIMITS into at most an 8-sided convex polygon. Therefore, it may be necessary to increase the facet density in order to see the effects of fine-scale trimming. The REMOVE option can be used to remove all facet edges from the plot except for the main patch edges and any trimming edges. This produces a just as accurate but less crowded plot. Alternatively, the HIDE option can be used to produce a hidden-line perspective view of the faceted objects. "d" is the viewing distance relative to the maximum lateral size (the default for "d" is 2 which simulates a typical 35mm camera lens). Use a "d" of 1000 to produce a nearly parallel projection. Currently, the following restrictions apply to HIDE:

1. There is a limit to the total number of facets that can be processed so you may only get a partial plot.
2. All objects are plotted in the same color.
3. The plot is always autoscaled to fill the window (or scaled window, i.e. "WINDOW 1.1" command before PLOT command).
4. The VECTOR unit is restarted so previous 3D plot data will be lost.
5. Unlike other PLOTS, an OBLIQUE view will always be true (right-handed). Therefore, OBLIQUE may not produce the expected view.

```
--- PLOT ... MESHES [ m ] ...
```

For MESHES, "m" is 0, 1 (the default) or 2 interpolation curves between segments of an object mesh (wireframe). The plotted meshes are not trimmed by any BOUNDS or LIMITS. However, any BOUND edges for an object will also be shown.

3.8.10 Replotting Accumulated 3D Graphics Data

Besides the external \$VIEW procedure, the following commands replot the current accumulated 3D VECTOR graphics (plus the current CONSIDER settings are taken into account).

```
--- REPLOT [ NORAYS ] [ OPTIMIZE ]
```

Replots within the current graphics device WINDOW all 3D vector data generated

so far (e.g. by the `PROFILES` and/or `PLOT` commands). Useful for zooming in on particular areas by changing the `WINDOW` and `REPLoT`ting. The `NORAYS` option suppresses the replotting of the rays themselves but does plot the intersection points of the rays on each object. The `OPTIMIZE` option forces the program to try and connect up any short vectors generated by a previous `PROFILE` command. This slows down the immediate generation of the graphics but produces a "cleaner" plot file that won't drive a pen plotter crazy.

For direct graphics devices that support an interactive graphics input mode, a graphical cursor (e.g. cross-hairs) will appear at the end of plotting. The user can then position this cursor and hit a key (or mouse button) to perform the following operations:

Key	Operation
-----+-----	
? (shift left mouse)	Display 3d system coordinates of cursor position
# (shift right mouse)	Display number of object nearest to cursor
space (left mouse button)	Mark the lower left-hand corner of a new window
enter (right mouse button)	Terminate replot

The `#` operation also replaces the current `GROUP` with all objects so-tagged (a `GROUP` and/or `CONSIDER` command can then be used to operate on these selected objects). The `?` operation also puts the coordinates in the output buffer for access by `$GRAB` or display by `$PLAY`.

For the last operation, the cursor should then be placed at the upper right-corner of the desired window. After striking another space key (or left mouse button), the program will draw a box on the current plot to indicate the extent of the new window and then beep. Hitting a carriage return (enter or right mouse button) will cause the program to clear the screen and to begin replotting the data in the new window. This "zooming in" process can be repeated indefinitely. If the two corners of the new window are entered as the same point, then the program zooms back to include all the plot.

--- DRAWING [xyz] [DIMENSIONS] [NORAYS]

Plots a black-and-white four-view drawing of the current data in the 3D `VECTOR` file. The default views are:

Quadrant	View
-----+-----	
Top Left	X vs. Z
Top Right	Y vs. X
Bottom Left	Y vs. Z
Bottom Right	YX Oblique

The three non-oblique views are aligned plus identically scaled, and optionally, can have simple DIMENSIONING added. The optional "xyz" argument is a three-letter permutation of the default XYZ coordinate ordering and can be used to obtain a different set of views. The NORAYS option suppresses the replotting of the rays themselves but does plot the intersection points of the rays on each object.

```
--- VUFACETS [ n n' [ m ] ] [ LIST ]
```

This command is a shortcut for the \$IO VECTOR REWIND, PLOT FACETS [n n'[m]], and \$VIEW command sequence. However, it temporarily suppresses any 2D plotting in order to minimize computation time and disk space. It also remembers the facetting numbers "n n'm" from the previous VUFACETS command (an END command resets them to their defaults). The numbers of facets created for each object can also be LISTed (a \$VIEW is not automatically issued in this case).

3.8.11 Rendering Of The System Geometry

```
--- LIGHTS e a,b,c [ e' a',b',c' [ e" ... ] ]
      -e x y z -e' x y z -e"
```

Sets up optional light sources (up to 10) for the RENDER command. if "e" is positive, then it is the relative irradiance of the collimated beam and "a,b,c" its direction of propagation. If it is negative, the light is an isotropic point source at "x y z". Note that the addition of these light sources can significantly increase the time to render a scene.

```
--- RENDER [ d ] [ DEPTH [ d' [ d" ] ] ] [ RAYS [ s ] ] [ MODEL [ m ] ]
      x y z
```

Performs a rendering of the system geometry given the current WINDOW, PIXEL,

and LIGHT setting. The observation (eye/camera) point is either given absolutely as "x y z" or as a distance "d" (default 10 times maximum scene span) in front of the scene. The scene is always illuminated by a light source of unit irradiance emanating from this point (i.e. a camera mounted flash bulb). Therefore, unless a second LIGHT source is specified, no shadowing will be visible. Normally, only the "outside" of the objects will be visible. However, they can be cut away in the DEPTH direction at the hither and yon coordinate planes d' and d'' .

RAYs can be added to the rendering if SAVE was turned on before the last TRACE. Only those portions of the ray paths not hidden by any part of any surface (whether it is transmissive or not) will be visible. Note that the addition of rays can significantly increase the time to render a scene, therefore the number of rays should be kept to a minimum. "s" is an optional scale factor for the "width" of the rays (default is 1, i.e. one pixel). If "s" is entered as a negative number, then each ray segment is rendered as a smooth cylinder (i.e. anti-aliasing).

Normally all object surfaces are considered to be perfect 100% Lambertian diffuse reflectors. With the MODEL option, any reflective SCATTER MODELS assigned to objects are used in the rendering. Objects with no SCATTER MODELS assigned to them will either default to 100% Lambertian or the given MODEL "m".

A somewhat crude representation of the scene is written to the 2D plot device during rendering process. A precise "picture" is also written to the binary distribution file BRO009.DAT (see DISPLAY command). Each pixel point is a single floating point value where the integer part is a color number and the fractional part its intensity (shade). A standard COLOR option can be used to set the background color to something different than the default black. Previously, a separate utility program (such as DIS2PS or DIS2VGA) was needed to display this file on a device capable of representing many intensities of the colors. This utility can still be run automatically if it is specified with the IMAGER switch or environment variable.

3.8.12 Mapping Of The System Geometry

```
--- MAP [ name ] [ DEPTH [  $d'$  [  $d''$  ] ] ] [ SLOPES ]
```

Produces a "depth" map of the current object surfaces using the last WINDOW and PIXEL settings. The map is saved to distribution file "name.DIS" (default BRO009.DAT) for later processing and can optionally contain not only the depth value but also the two SLOPES at each scan point. If the actual range of depth values (d' to d") are not given, then they are automatically calculated from the current object geometry (and reversed if a DEPTH option is present but without any numerical entries).

3.8.13 Standard Plot Command Options

These standard modifier can be used on the PROFILE, REPLOT, PLOT, TRACE PLOT, HISTORY PLOT, and SPOTS commands.

3.8.13.1 Overlaid And Multiple Window Graphics -

```
--- ...OVERLAY [ k ]
```

This option tells the program not to issue an end-of-plot so that the next plot done by the program will be overlaid with this one. For example, this is useful for combining system plots with ray trace plots (assuming the data WINDOW is the same for all plots), multiple spot diagrams, etc. The optional entry is an integer between 1 and 4 inclusive that tells the program to scale the plot down to half its size and place it in the "k"th quadrant of the normal plotting area. This feature can be used to overlay but not overlap successive plots (e.g. spot diagrams for different source points). The quadrants are numbered in the following manner:

1 = Upper left 2 = Upper right

3 = Lower left 4 = Lower right

Due to the scaled down size, no ordinary border text is placed on the plot. Also, the default for "k" is 0 (normal full size plot) if the last plotting command did not have an OVERLAY option on it. Otherwise, it is one more than the last quadrant.

3.8.13.2 Drawing 2D And/or 3D Graphics Text -

```
--- ...TEXT
--- x y [ z ] x' y' [ z' ] x" y" [ z" ] [ k ] 'string'
  :
```

The text "string" of color "k" (default 1) is written to either the 2D graphics device or both the 2D and 3D devices. The number of entries determines whether the starting point (unprimed coordinates), character spacing vector (primed), and character height vector (double primed) are in 2D plot units or 3D system units.

Number of entries	Interpretation of coordinate entries
-----+-----	
<8	2D only, "paper" coordinates (0-10,0-7.5)
>7	3D (and 2D projection), system coordinates
x,y[,z]	Starting point
x',y'[,z']	Character spacing vector
x",y"[,z"]	Character height vector
k	Color number (1-26)

3.8.13.3 Overriding Default Colors With A New Set Of Colors -

```
--- ...COLORS k [ k' ... ]
```

The entities displayed during the current command will all be plotted in ONLY the colors given instead of any default coloring. Remember that the maximum color number 27 (or 0) is equivalent to the background and can be used for erasing in the graphics display.

3.8.13.4 Temporarily Overriding Number Of Pixels -

```
--- ...PIXELS n [ r ]
```

Sets the number of PIXELS to "n" for this plot only. "r" is the desired aspect ratio of the pixels, i.e. the ratio of the vertical pixel width to the horizontal (default 1).

3.8.13.5 Temporarily Overriding Current Plot Window -

```
--- ... XY[Z] [ m      ]
        YZ[X]   a a' d d'
        ZX[Y]
        YX[Z]
        ZY[X]
        XZ[Y]
```

Sets the window to the given directions and size (see WINDOW command) for this plot only. If the third direction is given, then an OBLIQUE plot will be generated. If the plot ranges "a a' d d'" are not given, the plot will be autoscaled (and then the window optionally magnified by a factor "m"). A good use of the option is to control the direction cosine space window without affecting the current spatial WINDOW, e.g.

```
SPOTS DIR YX -4@1
```

3.9 RAY TRACING COMMANDS

3.9.1 Commands For Setting Options Before Trace

The commands that actually initiate ray trace execution (RAY and TRACE) are described in later sections. The commands in this section are options that must be turned on or off before ray tracing is initiated.

3.9.1.1 Accuracy Versus Speed Of Ray/surface Intersection Calculations -

```
--- ACCURACY [ HIGH ]  
              MEDIUM  
              LOW
```

This command primarily controls the tradeoff between ray / surface intersection accuracy and calculational speed. HIGH (the startup default) should be used when parabal rays must be as accurate as the base rays and optical path lengths are required to be accurate to a small fraction of a wavelength over great distances. This is usually necessary for coherent high-resolution analyses. With LOW, parabal rays are left on the base ray's tangent plane. This is usually sufficient for incoherent straylight and illumination analyses. A PROFILE and other calculations are also affected by this command. Typical relative error values used in iterative solvers are:

LOW	1.E-4
MEDIUM	1.E-7
HIGH	1.E-10

If no option is specified, the setting reverts to the one before the previous ACCURACY command.

3.9.1.2 Controlling The Object Intersection Search -

```
--- SEARCH [ LIST ]  
          [ ALL FORWARD ]  
          SEQUENTIAL BACKWARD  
--- i      j      j'      j"  
   :
```

Determines the way that rays will be traced through the system by specifying the set of objects that the program will consider as possible nearest intersections for rays leaving an object. Additional entries on the SEARCH command itself apply to all objects. The range of possible objects can be selectively set by additional commands with the first entry "i" being the number for the particular object followed by a pair of entries. The range of possible objects can be specified using the following combinations of input (m=largest possible object number).

Input		Object Range and Increment		
-----+-----		-----		
j	j'	j''	j to j' by j''	
ALL			1	m 1
ALL	FORWARD		i	m 1
ALL	BACKWARD		i	1 -1
SEQUENTIAL			i-1	i+1 2
SEQUENTIAL	FORWARD		i+1	i+1
SEQUENTIAL	BACKWARD		i-1	i-1

Note that for any ALL input, the object itself is considered for the next intersection for the rays. The LIST option prints the current SEARCH settings for each object.

```

--- ALLOWED [ LIST ]
          [  ALL    FORWARD    ]
          [  SEQUENTIAL BACKWARD ]
---  i      j      j'      j''
   :
```

Identical in format to the SEARCH command except instead of controlling what group of objects to search for the next ray intersection, this command determines whether a ray is allowed to proceed once it has reached the next object.

3.9.1.3 Setting Conditions On Terminating Trace -

```

--- CUTOFF [ t ] [ n ]
```

Sets the absolute flux threshold below which rays are ignored by the program to the decimal number "t" (default 1.E-18). The maximum number of total object intersections for any ray can be set to the integer "n" (default 1000).

```

--- HALT [ n ] [ OFF ] [ c ]
      +-X
      Y
      Z
```

Forces the tracing of certain rays to be halted. "n" is the maximum number of times a ray can consecutively interact with the same non-LENS object. The literal entry picks out a certain coordinate direction or turns it OFF. If a ray attains a component in this direction after reflection or refraction or the number of consecutive interactions with the same non-LENS object reaches the integer "n" (default 12), then tracing halts for that particular ray. Tracing of a ray whose ratio of current flux to initial flux falls below the decimal number "c" (defaulted to 1.E-6) will also be stopped.

3.9.1.4 Refraction/reflection Controls -

The previous HALT control and the following interface controls become the defaults used by any object for which they are not explicitly set. Since they are identical to the OBJECT subcommands, make sure you are not at the OBJ> prompt. If you are, issue a RETURN command first.

```
--- FRESNEL [ TIR ]
                OFF
                AVE
                S
                P
                BOTH
                k
```

Signals the program to take into account the variation of the reflection and transmission coefficients with incidence angle at each interface and reduce the ray flux accordingly. The "k" parameter or equivalent literal entry can be used to control which Fresnel coefficient to use in the refraction calculation, i.e.

Literal Integer k		
-----+-----+-----		
TIR	-2	Allow TIR rays but no angular flux variation for others
OFF	-1	Use normal incidence properties for all rays (default)
AVE	0	Incoherent average of polarizations
S	1	S polarization coefficient only
P	2	P polarization only
BOTH	3	Both separately

```
--- SPLIT [ n ] [ MONTECARLO ] [ c ]  
      TRAN  
      REFL  
      NORM  
      OFF
```

"n" is the maximum number times a ray can be split into specular components (OFF resets n to zero). When greater than zero, rays that encounter an object with non-zero reflection AND transmission coefficients will be split into separate reflected and transmitted rays. The most energetic at NORMAL incidence, the TRANSMitted, or the REFlected component continues to be traced while the other component is added to the top of ray storage and is eventually traced at a later time. NORM is the default.

With the MONTECARLO option, no additional specular and /or scattered rays are actually created, the new parent is probabilistically selected from the possible child rays. Otherwise, for a modest number of initial rays and refractive /scatter objects, the total number of rays eventually traced can be quite large. Even though the program has virtual ray storage, one is practically limited by the total free disk storage and long run times. Therefore, the default deterministic splitting should be used with due appreciation to its possible consequences. For a typical application such as ghost image analysis, a value of 2 for "n" is sufficient. If the fractional energy in the split-off ray / beam is below "c" (default 1.E-6), splitting does not occur, i.e. only the main component is propagated.

To control the level of scattered component splitting, use the following parallel command:

```
--- LEVEL [ n ] [ OFF ] [ c ]  
      ALL
```

"n" is the maximum number of scattering levels allowed for random diffuse rays. Deterministic near-specular and back-scattered beams are never rescattered. If the fractional scattered energy (relative to the incident ray) drops below "c" (default 1.E-12), the scattered ray is not generated.

Normally, scattered rays are not generated for any specular child rays, only the parent. If ALL is specified, scattered rays are generated for each specular child ray (including multiple diffraction orders, extraordinary rays, etc.), so that "bi-directional" scatter can occur at a partially reflecting / transmitting interface.

3.9.1.5 Trace Output Control -

```
--- WARNINGS [ n ] [ LIST ]
```

Controls output of parent ray warning messages. If "n" is zero or greater and more than "n" ray warnings occur, the raytrace is terminated. The default for "n" is a negative number which means there is no limit on the number of warnings. If the "LIST" entry is present, then all warning messages are printed during future ray traces.

```
--- AXIS [ OFF ]
      X
      Y
      Z
     -X
     -Y
     -Z
    LOCAL [ i ]
```

Allows future ray positions to be printed in either local and /or cylindrical coordinates relative to the specified axis. OFF resets the program to the default global Cartesian coordinates. LOCAL causes data to be output in a local Cartesian coordinate system whose origin is the reference point of object "i"'s base entity. The default for "i" is each ray's current object. If the axis designation is preceded by a minus sign, then a local cylindrical coordinate system is used. The cylindrical coordinates are "R" (radial distance from the axis), "T" (angle theta in degrees around the axis), and the actual axial coordinate ("X", "Y" or "Z" if positions, "A", "B" or "C" if directions).

```
--- MISSED [ OFF ] [ d ]
      ARROW
      LINE
```

Determines what type of vector is drawn during ray plotting to indicate a missed ray. Normally, the missed rays are extended only to the edge of the 2D plotting window. "d" is an optional distance in system units for extending the rays in 3D.

3.9.1.6 Random Ray Generation -

```
--- SEED [ n ] [ QUASI ]  
      OFF
```

Initializes the seed value for the random number generator used by the ROUGHNESS RANDOM, GRID RANDOM, SCATTER RANDOM/TOWARDS, and EMITTER commands. "n" should be some large odd integer and is set to 2000000001 at program startup. Optionally for the EMITTERS, a QUASI random number sequence can be used instead. In this case, only the least significant digits of "n" are used to select between 50 possible sequences. If a zero is entered for "n", then the QUASI number generator is no longer random and returns a fixed value of .5 every time.

3.9.1.7 Saving All Future Ray Intersections To A Binary Data File -

```
--- SAVE [ k [ name ] ]  
      OFF
```

Directs the program to save on an unformatted binary file information about every ray intersection found during any future ray traces. If "k" is present, it becomes the file extension (otherwise "his" is assumed) and is automatically incremented after each TRACE command unless another SAVE command is used to override it or turn it OFF. The default name part of the file is "rayinter" or it is taken from the last SYSTEM or \$IO command. Otherwise, it can be directly specified with the additional "name" entry.

This file is used by the next HISTORY or RENDER...RAYS command. After program completion, it can be processed by another program such as STRAP. Each of the fixed length records in the file corresponds to one ray intersection and contains the following 6 or 12 words (24 or 48 bytes) of information:

WORDS	CONTENT
-----+-----	
0	Previous record number
1	Object number (negative if starting or split object)
2	Incident power/flux of ray (negative if scattered ray)
3-5	Global X,Y,Z coordinates of ray on object
6	Cross-sectional size (height on surface) of beam centered on ray
7	Average radius of curvature of beam wavefront
8-10	X,Y,Z components of unit normal to object surface
11	Square root of the cosine of the incidence angle from normal

The first word (0) is a signed integer, all others are floating point. The short 6-word record is used for a SAVE command with no additional entries or if XMEMORY is set to MIN. The long 12-word record is the minimum amount of information needed by the STRAP module for calculating stray light paths.

The following is a sample program for reading a SAVE file and listing the results:

```

      PROGRAM DUMPS
*
*  dumps basic SAVE file information to text
*
      PARAMETER (LNRECD=1) !Digital/Compaq Fortran default
!      PARAMETER (LNRECD=4) !otherwise
      DIMENSION RADATA(11)
!      REAL NUMREC !ASAP versions 6.0 or earlier
      CHARACTER*100 FIL
      LOGICAL EX
*  ask for file
      WRITE(*,*) 'Enter file name:'
      READ(*,'(A)') FIL
*  determine record length from OS
      LENREC=0
      INQUIRE(FILE=FIL,EXIST=EX,RECL=LENREC)
      IF (.NOT.EX) STOP 'File does not exist'
      IF (LENREC.EQ.0) THEN
*  or from "zeroth" record
      OPEN(1,FILE=FIL,ACCESS='DIRECT',FORM='UNFORMATTED',
&      RECL=LNRECD,STATUS='OLD')
      READ(1,REC=1,IOSTAT=IOS) LENREC

```

```

        CLOSE(1)
        IF (IOS.NE.0) STOP 'Cannot read first record'
        IF (MOD(LenREC,256).NE.247) STOP 'Not a BRO binary file'
        LenREC=LenREC/256
    ENDIF
* check record length and open file
    LenREC=LenREC/4
    IF ((LenREC.NE.6).AND.(LenREC.NE.12)) STOP 'Wrong record size'
    OPEN(1,FILE=fil,ACCESS='DIRECT',FORM='UNFORMATTED',
        & RECL=LenREC*LnRECD,STATUS='OLD')
* get number of records
    READ(1,REC=2,IOSTAT=IOS) NumREC
    IF (IOS.NE.0) STOP 'Error reading second record'
    N=NumREC+1
    L=LenREC-1
* loop through records
    DO J=2,N
        READ(1,REC=J,IOSTAT=IOS) NumREC,(RADATA(I),I=1,L)
        IF (IOS.NE.0) STOP 'Error reading all the records'
        WRITE(*,*) NumREC,(RADATA(I),I=1,L)
    ENDDO !J
    CLOSE(1)
END

```

3.9.1.8 Setting Up Volume Energy Tracking And Visualization -

```

--- VOXELS FLUENCE [ d d' [ n ] ]
    ABSORBED  x x' y y' z z' [ n n' n" ]
    X
    Y
    Z
    -X
    -Y
    -Z
    +X
    +Y
    +Z

```

During the next TRACE, accumulate in a 3D array of volume elements (voxels), the flux per unit volume (total absolute FLUENCE or ABSORBED in media) or irradiance on a area perpendicular to the given coordinate axis. Only accumulates irradiance from rays going in the direction specified by the sign of the axis (no sign is equivalent to a bi-directional irradiance). The volume is either the enclosing box of the currently considered objects, determined by the current WINDOW settings and the specified depth range, or given directly in terms of a range for each of the global coordinates. The number of samples per coordinate is then either taken from the current PIXELS setting and a depth number or specified directly. If any of the "n"s is entered as zero, then the PIXELS setting at the time of the TRACE is used. The total number of voxels is limited to approximately 81 cubed in Basic or 203 cubed in the Pro version. After the completion of the next TRACE, the results are written to a 3D version of the standard binary distribution file (BRO009.DAT), and any 2D slice can then be processed by DISPLAY and its subcommands.

```
--- VOXELS [ OFF          ]
          READ [ u  ]
          file
```

Turn on or OFF the volume flux tracking during the next TRACE. Optionally, read in a previous 3D binary distribution given by number "u" (default 9) or name "file" (default DIS extension). The results of the next TRACE are added to these values and the total is then written to the standard distribution file (BRO009.DAT).

3.9.2 Commands For Setting Up Beam Characteristics

```
--- IMMERSE [ m ]
```

Sets the starting MEDIA for future rays to "m" (name or number, default 0).

```
--- PARABASAL n [ d  [ d' ] ] [ CLIP ]
--- [ g  h  u  v  ]
--- [ g' h' u' v' ]
--- [ g" h" u" v" ]
--- :
```

This command tells the program to trace "n" secondary rays close to each primary ray and should precede any ray definition commands. These rays will follow that same path through the system as the primary ray and are never clipped by a boundary unless the primary ray is also clipped. "n" usually takes on values of 0, 2, 4 or 8. When $n=4$ or 8 and a WAVELENGTH has been specified, the parabasal rays are used to define general Gaussian beams centered on each primary ray. A CLIP option signals the program to check for clipping of the parabasal rays by boundaries and reduce the beam flux accordingly.

"d" sets the default for the semi-divergence of each geometrical incoherent beam in radians. This value is only used if $n=0$ or 4 and the current WAVELENGTH is zero, i.e. a geometrical optics radiant beam.

Normally, the transverse positions and direction of the parabasals are automatically set by the program depending on the properties of the beams being created. However, an override table can be entered, one line per parabasal. The first two entries are the relative heights in the two orthogonal directions. The last two are the orthogonal convergence / divergence angles in radians.

```
--- WIDTHS w [ h ] [ EDGE ]
```

The "w" parameter is factor that the program scales the actual width of each beam by. When $w=1$ (the default), a GRID command will create a set of beams that just touch each other, i.e. unit packing density. When $w<1$ there are gaps between adjacent beams and when $w>1$ they overlap. The latter case is especially useful for overlapping a Gaussian beam set so that the overall beam shape is much smoother. For this application, a value of 1.414 is usually sufficient and a value of 2 is more than enough.

"h" is the ratio of the parabasal ray heights to the half-max half-width of the beam they define. The default is 1 but it can be set smaller if the exact parabasal rays depart too much from the first-order approximation on which the beam calculations are based.

The EDGE option causes the program to add narrow odd-order Hermite-Gaussian beams at the edge of a GRID (RECT or POLAR) of beams in order to preserve the sharpness of the overall beam pattern.

```

--- POLARIZ X [ a a' ]
      Y
      Z
      OFF

```

Sets the polarization direction for future ray creation (RAYSET and GRID) and analyses (SPREAD and FIELD). Also can be used to initialize the complex coefficients "a" and "a'" for the two orthogonal components. Used with FRESNEL BOTH command to analyze polarization effects.

```

--- BEAMS [ INCOHERENT ] [ GEOMETRIC ] [ SHAPES k [ s ] ]
          COHERENT      DIFFRACT

```

Sets the default coherence, propagation, and shape characteristics for future beam creation (except EMITTING and SCATTER rays which are incoherent, geometric, and of a fixed shape). See the SHAPE command for an explanation of "k" and "s" (except the sign of "k" has no meaning here). In most circumstances INCOHERENT should be used with GEOMETRIC and COHERENT with DIFFRACT (this is the default if only one is specified). Although it is possible to specify the other two "cross" combinations, the results can be unexpected and therefore useful only in a few rare cases (e.g. BEAMS COHERENT GEOMETRIC SHAPE FIBR/SLAB for a non-diffracting guided mode). The default in the absence of a BEAMS command depends on the WAVELENGTH and XMEMORY settings at the time of the first ray/beam creation:

WAVELENGTH 0 or XMEMORY MIN		INCOHERENT GEOMETRIC
	= BEAMS	
otherwise		COHERENT DIFFRACT

From then on, the BEAM settings are not affected by future WAVELENGTH and XMEMORY commands unless a BEAMS command with no arguments is issued.

3.9.2.1 Positional And/or Directional Clipping Of Rays During Creation -

CLIP specifies POSitional or DIRectional limits boxes for clipping rays during creation, i.e. any ray coordinates falling outside the given box will not be created. If set, positional clipping is done in all cases except the single RAY command; directional clipping only for the SOURCE DIR GRID, EMITTING, and DECOMPOSE commands.

```
--- CLIP OFF
```

```

POS [ x x' [ y y' z z' ] [ X [ f ] ] ]
DIR  a a'   b b' c c'      Y
                                Z
                                SHORTEST
                                LONGEST
```

Turns OFF any future positional and directional ray creation clipping. Otherwise, the unprimed entries are the lower bounds of the box, primed the upper. If a coordinate direction is also given, then the coordinates are constrained by a cylinder of constant elliptical (as opposed to rectangular) cross-section in planes perpendicular to the given coordinate. This axis can also be determined by the SHORTEST limit box dimension, or the LONGEST. An optional inner boundary of fractional height "f" [default is zero] can be used to put a proportional hole in this direction.

```

--- CLIP POS [ OFF      ]
      DIR  AXIS      X [ f ] ]
          -X  x      Y
              a      Z
          -Y  y      LONGEST
              b      SHORTEST
          -Z  z      OFF
              c
          +X  x'
              a'
          +Y  y'
              b'
          +Z  z'
              c'
      EXPAND r
              X r
              Y
              Z
```

The above short form can also be used to temporarily turn OFF future clipping or to reset the limits AXIS or just one of the six limits values. EXPAND can be used to enlarge (or shrink) by a relative amount "r" the entire limits box or just in one direction, e.g. "EXPAND -.1" shrinks the entire box by 10%.

For CLIP POS, more complex ray bounding can be done by following it with:


```

--- BOUNDS [ k [ k' ... ] [ MULTIPLE m [ m' ... ] ] ] [ POINT x y z ]
      RBOUNDS OFF

```

The "k"s are the numbers of the surface and/or edge entities that are to be used to define the spatial boundaries for future ray creation. If these numbers are positive, then a valid ray is assumed to be on the positive side of the corresponding surface, outside the surface's SOLID or outside the closed edge. A negative number means that a valid ray is on the surface's negative side, inside the surface's SOLID or inside the edge. The MULTIPLE option makes a group of surfaces/edges equivalent to one standard constraint, i.e. a point that is valid for one or more of the given "m" bounds is a valid ray point. To efficiently enter a very large number of boundaries, use instead the RBOUNDS version where every pair of "k"s or "m"s specifies a range of entity numbers instead of individual entities.

The POINT option switches the signs of the entities if necessary in order to make "x y z" a valid ray point. A BOUNDS OFF command can be used to temporarily turn off all boundary surfaces for future ray creation. A BOUNDS command by itself will turn them back on.

3.9.2.2 Spectral Apodization -

```

--- SPECTRUM [ OFF          ]
              FCN fcn [ s ]
              VISUAL
              SCOTOPIC
              THERMAL t
              PHOTONS t
              w w' w'' w''' ...

```

Sets up or turns OFF spectral apodization (flux weighting as a function of wavelength) for future ray/beam creation. Two user-definable and four standard spectral curves are provided. The user-definable one can be either a \$FCN called "fcn" or a table of weights "w w' ..." (one for each wavelength specified on the last WAVELENGTHS command). The standard ones are defined as follows:

VISUAL Standard Luminosity Curve (photopic eye in bright light)

SCOTOPIC Eye response in dim light

THERMAL Power output of a blackbody at temperature "t" (degrees K)

PHOTONS Photons/second of a blackbody "

These four curves are by default normalized to unity at their maximum. "s" can optionally be used to scale to some other peak value. The following is an example of creating a polychromatic source with visual weighting:

```
UNITS ...
SPECTRUM VISIBLE
$DO 400 700 25
{ WAVELENGTH ? NM
  :
  ray creation commands (e.g. EMIT)
  :
}
```

3.9.2.3 User Apodization Of Ray Distributions -

APODIZE operates immediately on all the currently selected or considered rays and therefore, can be used before or after a TRACE. USERAPOD will only be applied to rays when they are created and just for the following combinations:

Ray Creation Commands	USERAPOD			
	POS	DIR	ANG	BOTH
-----+-----+-----+-----+-----				
GRID OBJECT/DATA				
All other GRIDs	x			
RAYSETs				
SOURCE POS GRID	x			
SOURCE DIR GRID	x	x		
All other SOURCES				
DECOMPOSE				
EMITTING RECT/DISK	x	x		
EMITTING IES				
EMITTING OBJ/ENT		x	x	x
EMITTING DATA plane		x		
EMITTING DATA volume			x	

All other EMITTING

x

x

```

---  USERAPOD POS [ fcn ] [ c c' c" ... ] [ 'string' ]
      APODIZE  DIR  OFF
      OFF

```

The user can define his or her own POSitional and/or DIRectional apodization function to be applied to a set of rays or beams created in a GRID or EMITTING surface command. This is done by either using the given \$FCN function "fcn" or rewriting the Fortran function USERAPOD and relinking the program. This command can also be used to turn on or OFF the use of the currently defined apodization function by the program.

If "fcn" is specified, the in-plane two-dimensional POSitional or DIRectional coordinates are passed in the "_1" and "_2" registers. The 50 coefficients "c c'..." are passed in "_3 _4 ... _52". The last entry in the \$FCN definition of "fcn" will be used as the apodization factor for scaling the ray / beam fluxes. For example, to define and use a Gaussian apodization (identical to the default one discussed below):

```

$FCN APOD _5*GAUS[_1/_3]*GAUS[_2/_4]/(_3*_4)
:
USERAPOD POS APOD ...
:
GRID ...
SOURCE ...

```

Otherwise, up to 50 coefficients and the "string" can be passed to USERAPOD where the default apodizing function is a Gaussian, e.g. for a Z grid the default unit flat top intensity is multiplied by:

$$[c''/(cc')] e^{-\pi[(x/c)^2 + (y/c')^2]}$$

Therefore, c and c' are the semi-major widths of the Gaussian envelope. "c" is normally the total flux in the beam (assuming that the default POS irradiance or DIR radiance distribution is unity, and it is not truncated appreciably by the finite size of the GRID or EMITTING surface).

```

---  USERAPOD POS c c'
      APODIZE DIR
---  [ e p [ f ] [ PLOT ] ]
      PROD
---  [ e' p' [ f' ] ]
      :

```

Also, the user can enter a table of measured energy flux in the two orthogonal grid directions "e,f" versus POSition or DIRection cosine "p" from which the program will interpolate to get the needed amplitude value in the GRID plane. The "c" and "c'" values are then used to scale the "p" value into the two coordinate values at the GRID or EMITTER plane. A default distribution file ("BRO009.DAT") can be created for future PLOTting by DISPLAY (the current PIXELS setting controls the resolution). Optionally, the PRODuct of the two orthogonal profiles can be used instead of the normal weighted sum.

```

---  USERAPOD ANGLES X [ fcn ] [ c c' c" ... ] [ 'string' ]
      APODIZE      Y
                  Z
                  OFF

```

Apodize as a function of spherical angles for the given axis the 3D radiation patterns of emitters (except RECT, DISK and IES) according to the predefined \$FCN function or the user-supplied Fortran in the function USERANGS. If "fcn" is specified, then the zenith angle from the axis (0 to pi radians) and the azimuth angle around the axis (-pi to pi radians) are passed in the "_1 _2" and the 50 coefficients "c c'..." in "_3 _4 ... _52".

Otherwise, the optional coefficients and "string" are passed to USERANGS. The default definition for USERANGS is a spherical angle clipping and/or interpolation, i.e.

```

---  USERAPOD ANGLES X [ f f' t t' ] [ i t" s      ]
      APODIZE      Y      -1      'filename'
                  Z

```

"f f'" are the limits of the zenith angle (in degrees) measured from the axis (default 0 to 180). "t t'" are the limits of the azimuth angle measured around this axis (default 0 to 360).

"i" is an optional number (non-zero) of a file that contains an interpolation table in spherical coordinates. If "i" is negative, the complete

case-sensitive name of the file is taken from the comment string. If "i" is positive and less than 80, then the file's name must be "usap3d" and its extension is the character whose ASCII value is $i+48$ (e.g. for $i=2$, the file is "usap3d.2"). Otherwise, the file name is the number itself and the extension is "ua3" (e.g. for $i=9007$, the file is "9007.ua3"). "t" is an azimuthal angle data offset and "s" a data scale factor for the data in the file.

```
--- USERAPOD BOTH [ fcn ] [ c c' c" ... ] [ 'string' ]
      APODIZE          OFF
```

Apodize in both position and /or direction the full 3D radiation patterns of emitters (except RECT, DISK and IES) according to the predefined \$FCN function or the user-supplied Fortran in the function USERBOTH. If "fcn" is specified, then the global positional coordinates are passed in the "_1 _2 _3" registers, the unit direction vector in "_4 _5 _6", and the 50 coefficients "c c' ..." in "_7 _8 ... _56".

Otherwise, the optional coefficients and "string" are passed to USERBOTH. The default definition for USERBOTH is for the present the same as USERANGS, a spherical angle clipping and /or interpolation, i.e.

```
--- USERAPOD BOTH k f f' t t' [ i t" s          ]
      APODIZE      X          -1      'filename'
                   Y
                   Z
```

"k" is an axis specification ($1=x, 2=y, 3=z$) for a spherical coordinate system.
 "f f'" are the limits of the zenith angle (in degrees) measured from this axis.
 "t t'" are the limits of the azimuth angle measured around this axis.

"i" is an optional number (non-zero) of a file that contains an interpolation table in spherical coordinates. If "i" is negative, the name of the file is taken from the comment string. If "i" is positive and less than 80, then the file's name must be "usap3d" and its extension is the character whose ASCII value is $i+48$ (e.g. for $i=2$, the file is "usap3d.2"). Otherwise, the file name is the number itself and the extension is "ua3" (e.g. for $i=9007$, the file is "9007.ua3"). "t" is an azimuthal angle data offset and "s" a data scale factor for the data in the file.

3.9.2.4 Paging Of Extended Ray/beam Memory -

```
--- XMEMORY [ OFF ] [ 'pagefile' ]
              MIN      m
              NORM
              FULL
```

Controls the amount of ray data stored in the virtual memory paging file (default VIRTUAL.PGS) or entirely in RAM for a given maximum of "m" rays. Can reduce the file/RAM size (i.e. allow more rays to be traced for a given file/RAM size) and speed up execution.

XMEMORY WORDS(BYTES)		
setting	per ray	COMMENTS
-----+-----+-----		
FULL	128 (512)	Automatically turned on by PARABASAL command
NORM,OFF	48 (192)	no parabasals and higher-order beam modes (default)
MIN	16 (64)	no polarization, path lengths, paths, and beam sizes also LIST ELLIP,PLOT BEAMS/POLARIZ,SPREAD,FIELD,OPDMAP disabled

3.9.3 Executing A Single Ray Trace

```
--- RAY x y z a,b,c [h] [DIR] [PLOT [d]] [SEARCH] [KEEP] [GALOP] [LENS [n]]
```

Traces a single ray /beam through the entire system starting in the IMMERSE medium at the point "x y z", with a direction given by the vector "a,b,c" and cross-sectional semi-diameter "h" (default 1). The program always lists the ray coordinates at each surface intersected. The DIR option adds the ray directions to the listing. The PLOT option plots the ray (and any PARABASALS) in the current graphics WINDOW. If this command is entered interactively by the user and the ray is being overlayed on a previous system graphics screen, then the user can trace more rays by marking with the graphics cursor the starting point (depth "x","y" or "z") and a second point (depth "d") for each desired ray. Hit return to terminate this mode. The SEARCH option resets the allowable object intersections for future rays according to the sequence of objects encountered during this single ray trace. Unless the KEEP option is

used, any rays traced with this command are deleted from storage after they have been traced. The sequence of surfaces encountered by the parent ray can be optionally written to a GALOP input (".ING") file and/or used to create LENS entity "n" (default is one more than current highest entity number).

3.9.4 Tracing A Group Of Rays Or Beams

Any number of rays or beams may be traced at one time. The procedure involves initializing the ray and beam data (coordinates, directions, and sizes) at a plane in the system which the program designates as object zero. The commands RAYSET and GRID set the coordinates and sizes while the SOURCE command then sets the directions and starting optical path lengths. The FLUX command initializes the energy flux per ray or beam. Actual execution of the ray / beam trace is accomplished with the TRACE command. At any time the new ray data may be analyzed using the STATS and SPOTS commands. Coherent beam data is analyzed using the SPREAD and FIELD commands.

3.9.4.1 Initialization Of Ray And Beam Data -

```
--- RAYS [ n ]
```

This shortened version of the RAYSET command allows one to reset the total number of rays in storage to "n". The default is the number of rays in the current virtual ray file, i.e. the rays from a previous run can be restored via this command. Thus, the following two commands can be used to restart an analysis from the last time the program was run within the current directory:

```
SYSTEM FROM !restore system from LASTEXEC.SYS file
RAYS       !restore rays/beams from VIRTUAL.PGS file
```

3.9.4.1.1 Ray Tables -

```
--- RAYSET X x
--- y [ z f y' z' k [ a a' a" ... ] ] [ NOSPLIT ]
      :                      s
```

```
--- RAYSET Y y
--- z [ x f z' x' k [ a a' a" ... ] ] [ NOSPLIT ]
      :                      s
```

```
--- RAYSET Z z
--- x [ y f x' y' k [ a a' a" ... ] ] [ NOSPLIT ]
      :                      s
```

The second and third entries on the RAYSET command designate the plane from which the rays will originate (object zero). The ray coordinates in this plane are then entered on successive commands. "f" is the flux (energy per unit time) to be assigned to each corresponding ray (default=1). The primed coordinates are the relative positions of the 2 orthogonal parabal rays to be traced if the number of parabal rays has been previously set by a PARABASAL command. "k" sets the beam shape and coherence of the "ray" (see SHAPE command definition of "s" and further information). If k=0, the beam is a coherent set of Hermite-Gaussian modes with the "a"s being the complex amplitude coefficients for each mode. The modes are ordered as follows:

```
00 10 01 20 11 02 30 21 12 03 40 31 22 13 04 50 ...
```

The default amplitude for the fundamental mode (00,"a") is 1. The higher order modes are defaulted to 0.

The NOSPLIT option tells the program to never split this ray, regardless of what the SPLIT or LEVEL commands indicate. The short numeric form for the RAYSET command resets the number of rays in storage to "n". The RAYSET and GRID commands always add their rays to the ones already in storage until a TRACE is performed.

3.9.4.1.2 Ray Grids -

There are a large number of ways to create grids of rays. When in doubt about the exact distribution that will be created by one of the following commands, use the SPOTS POSITION command to display the grid.

```
--- GRID RECT X x y y' z z' n n' [ c ] [ RANDOM f ]
              Y y z z' x x'
              Z z x x' y y'
```

Generates a uniform rectangular grid of rays on the plane given by the second and third entries. The fourth and fifth entries determine the maximum extent of the grid in one direction and the sixth and seventh in the other direction on that plane. The number of rays in each direction are given by the "n" and "n'" entries. The optional "c" entry specifies a fractional height for a central hole in the grid. Each position on the grid can be uniformly RANDOMized within a region "f" times the local grid spacing.

```
--- GRID ELLIPTIC X x y y' z z' n n' [ c ] [ RANDOM f ]
                 Y y z z' x x'
                 Z z x x' y y'
```

Similar to GRID RECT in that it first generates a uniform rectangular grid of rays on the plane given by the second and third entries. The fourth and fifth entries determine the maximum extent of the grid in one direction and the sixth and seventh in the other direction on that plane. The number of rays in each direction are given by the "n" and "n'" entries. This rectangular grid is then clipped by an inscribed elliptical aperture. The optional "c" entry specifies a fractional height for a central hole in the grid. Each position on the grid can be uniformly RANDOMized within a region "f" times the local grid spacing.

```
--- GRID POLAR X x r r' a a' n n' [ w ] [ RANDOM f ]
               Y y
               Z z
```

Generates a circular grid of rays on the plane given by the second and third entries when the extra entry "w" is not unity. The plane is normal to one of the given axis. "r" and "a" entries determine the maximum extent of the grid on that plane in the radial and angular directions respectively. The number of rays in the radial direction is given by the "n" entry. "n'" is the number of rays in the first non-zero radial "ring". The sign of the "n" determine how

the range entries are used, i.e.

```
+n  divide r to r' into n rings
-n  divide 0 to r' into n rings, eliminate rings inside r
+n' divide a to a' into n sectors (first ring)
-n' divide 0 to 360 into n sectors, eliminate sectors outside a to a'
```

For $r=0$ or $n<0$, the rays are distributed radially according to the formula:

$$R(i) = r'(i/n)^w$$

The rays are distributed evenly in the angular direction with the number in each "ring" selected so that they all represent an equal area on the plane. Therefore, a value of 1 for "w" (the default) yields an equipolar distribution while a .5 corresponds to a standard straight spokes distribution. If "r" is equal to zero, there will be a ray/beam at the center of the grid. Each position on the grid can be uniformly RANDOMized within a region "f" times the local grid spacing.

An alternate shorter form is available that simply creates a full circular (of radius "r") hexapolar grid (with "n" rings):

```
--- GRID POLAR X x r' n [ RANDOM f ]
      Y y
      Z z
```

```
--- GRID HEX X x y z h n [ RANDOM f ]
      Y y z x
      Z z x y
```

Creates a hexagonal grid of rays normal to the given direction and centered at the given coordinates. "h" is the height of the hexagon as measured from the center to one of its vertices. "n" is the number of rays along one side of the hexagon. Each position on the grid can be uniformly RANDOMized within a region "f" times the local grid spacing.

```
--- GRID OBJECT k [ n [ +-X ] ] [ RANDOM f ]
      Y
      Z
```

Generates a uniform grid of rays based on the location and dimensions of the

given object "k". Normally, ray positions lie at the centers of the object's facets. Otherwise, approximately "n" rays are distributed over the LIMITS box end specified by the last entry. Each position on the grid can be uniformly RANDOMized within a region "f" times the local grid spacing.

```

--- GRID WINDOW [ f [ m n [ RANDOM r ] ] ]
                MIN
                MAX
                CEN
                OBL

```

Creates a rectangular raster-type ray grid using the current WINDOW dimensions and sampling "m" rays vertical by "n" horizontal (or current PIXEL settings). The grid depth plane is given by "f" (default 0), the MINimum, MAXimum, CENTER, or last OBLique value. Each position on the grid can be uniformly RANDOMized within a region "r" times the local grid spacing.

```

--- GRID DATA [ file [ n ] [ RANDOM r ] ]
                u

```

Creates a rectangular grid of rays based on the coordinates found in the header of the distribution "file" (default .DIS extension) or unit number "u" (default 9). A ray is created at each sample location in the file (or optionally every "n"th location). Each ray's flux will be proportional to the corresponding data value in the file. Each position on the grid can be uniformly RANDOMized within a region "r" times the local grid spacing. In addition to the usual CLIP POSITION, the current flux CUTOFF and HALT settings may also truncate the distribution. If the labels in the file do not contain coordinates, then the data is mapped to the current plotting WINDOW at the zero depth plane.

3.9.4.1.3 Setting Ray Directions According To Source Configuration -

```

--- SOURCE POSITION x y z [ x,y,z' x,y,z" ... ]
                FOCUS
                LINE
                DIRECTION a,b,c   a,b,c' a,b,c" ... [ RANDOM f ]
                        GRID [ ONE ]

```

Used after the GRID and RAYSET commands to initialize the directions and optical path lengths of the rays. (x,y,z) is the point from which the rays appear to emanate (POSITION) or converge (FOCUS). The vector (a,b,c) specifies a direction assigned to all the rays, i.e. a collimated beam. For SOURCE LINE, each pair of coordinates locates the foci of an ellipsoidal wavefront. If the foci are sufficiently separated, one effectively gets a line source.

The optical path length of the first ray is always chosen to be zero. More than one source location or direction can be entered on this command. For SOURCE DIRECTION, the directions can be RANDOMized within an angle "f" times the divergence angle given on the PARABASAL command.

The GRID option will use the coordinates defined in the last GRID command as the SOURCE coordinates and not as starting ray coordinates. If these coordinates are to be DIRECTIONS, then the sum of their squares should never exceed 1. If they do, they are eliminated from the grid. The sign of the third direction is taken from the 4th entry on that GRID command. By default each entry in the source grid is treated as a separate source or the entire grid can be treated as ONE source.

3.9.4.1.4 Setting Ray Directions According To Desired Wavefront Function -

```
--- SOURCE WAVEFUNC k [ p k' p' ... ]  
      X  
      Y  
      Z  
      N
```

The wavefront function or "eikonal" is specified by the previously defined surface function "k" raised to "p"th power (default 1). The normalized gradient of the function (times the sign of "k") becomes the ray's direction, the normalized value it's optical path length. Alternatively, the ray positions will be moved along the given direction ("N" means normal to "k") to the actual wavefront surface "k". Multiple source wavefronts can be created with additional pairs of entries.

3.9.4.1.5 Creating Rays To Simulate A Random Emitter -

The EMITTING commands simultaneously set the ray positions and directions in a random manner in order to simulate natural emitters, e.g. blackbody radiators.

3.9.4.1.5.1 Emitting Surfaces -

```

--- EMITTING DISK X x y z n [ a [ a' ] [ RECT ] ] [ ISO ]
      RECT Y y z x
            Z z x y
            -X
            -Y
            -Z

```

The emitting surface is either an elliptical / circular DISK or a RECTangular plate perpendicular to the given axis (third entry, sign determines emission direction) and at the given location (fourth entry) on that axis. The semi-major heights in the other two directions are given by the fifth and sixth entries. "n" is the total number of rays that will be created by this command. "a a'" are optional half-angles in degrees for the elliptical or RECTangular cone of emitted radiation centered on the axis. The default for "a'" is "a", i.e. circular or square light cone. The default for "a" is 90 degrees, i.e. a full hemisphere. By default, the surface emits directionally in a Lambertian fashion or ISOtropically. This pattern can be further apodized in position and / or direction by the current USERAPOD settings.

3.9.4.1.5.2 Emitting Geometry -

```

--- EMITTING ENTITY k [ n ] [ ISO ]
      OBJECT -k -n [ d ] NORM
            +n

```

Randomly distributes rays over the faceted surface(s) of entity number "k" or object "k" (number or name). If the integer entry "n" is negative, its absolute value is the maximum total number of uniformly distributed rays that

will be created. If preceded by a plus sign, "n" is the number of rays per facet and the total number of rays created depends on the entity's intrinsic patching, the object's FACETS setting, and this value. In the absence of a leading sign, the previous default was plus but this has been changed to minus, i.e. the far more common total rays usage. The default for "n" is still +100. The resulting ray positions will lie very nearly on the actual smooth surface(s) of the entity or for an object, a distance "d" (default is a small fraction of largest limits box dimension) from the actual smooth surface(s). If floating point number "d" is entered as exactly "0.", then the rays will be tagged as being on the actual object instead of object zero. If NORM is specified, the rays will be directed exactly parallel to the local surface normal. Otherwise, the rays will directionally radiate in a random Lambertian or ISotropic fashion from the surface(s). The sign of "k" controls the direction of the hemispherical emission relative to the local surface normal, i.e. whether the emission is, for example, outward or inward from a sphere. Before apodization, the flux emitted by a facet is equal to its area. Therefore, the total flux in the rays will approach the total exact surface area of the object as the number of facets increases. For example, to simulate the thermal emission from an object of a given temperature:

```

UNITS CM
:
WAVELENGTH1=.4 microns
WAVELENGTH2=.7 microns
TEMPERATURE=300 degrees kelvin
EMISSIVITY=.5 normally same as absorption ratio
SIGMA=5.672E-12 stefan boltzmann constant for cm units
FRACTION=FBI(WAVELENGTH2*TEMPERATURE)-FBI(WAVELENGTH1*TEMPERATURE) for watts
FRACTION=LPW(TEMPERATURE) for lumens
EMITTING
    FRACTION*EMISSIVITY*SIGMA*TEMPERATURE^4  OBJECT ... !no ISO/NORM option
RETURN
:
```

3.9.4.1.5.3 Emitting Volumes -

The following commands randomly create rays within a specified volume. The flux of each ray can be modified via an "apodization" function programmed by the user USERAPOD ANGLES/BOTH commands.

```

--- EMITTING CONE      X x y z x' y' z' n [ ISO ]
                        PYRAMID Y y z x y' z' x'
                        Z z x y z' x' y'

```

Creates a ray set distributed within the elliptical CONE or rectangular PYRAMID centered on the given axis with one end defined by entries 4 through 6 and the other end by 7 through 9. "n" is the total number of rays created. The number of rays per unit length along the axis is held constant so that the ray density will be higher at the smaller end of the volume. This closely simulates the plasma in a compact arc lamp. Normally, the far-field radiation pattern is a donut, i.e. it varies as the sine of the angle from the axis. The "ISO" option will yield an isotropic radiation pattern.

```

--- EMITTING BOX      x y z u v w n [ X ]
                        SPHEROID Y
                        Z

```

Creates a ray set uniformly distributed within the BOX or SPHEROID centered at "x y z" with semi-major heights "u v w". "n" is the number of rays created. By default the radiation pattern is isotropic. However, if the optional axis entry is given, the radiation pattern is a donut centered on that axis, i.e. varies as the sine of the angle from that axis.

3.9.4.1.5.4 Emitting Space Curves (wires) -

```

--- EMITTING HELIX X x x' t h r n [ ISO ]
                    Y y y'
                    Z z z'

```

Creates "n" random rays that follow a helical (spiral, spring, coil) curve. The helix extends along the given axis from the fourth to fifth entries. "t" is the number of turns (not necessarily integer). "h" is the major radius and "r" the minor radius (semi-thickness) of the helix. Normally, each segment of the helix emits like a Lambertian cylinder. The ISO option will force each point to radiate isotropically. Besides a true helix, this command can also be used for some special cases of interest:

1. Torus - set the two axial coordinates to be equal and "t" to one.
2. Hollow Cylinder - set "t" to a high number and "r" to $|x'-x|/2t$ (X case).

The following long format is more flexible since it lets the helix vary from one elliptical shape (unprimed coordinates) to another (primed):

```
--- EMITTING HELIX X x y z x' y' z' t r n [ ISO ]
                  Y y z x y' z' x'
                  Z z x y z' x' y'
```

To get a planar spiral, set the two axial coordinates to be equal.

```
--- EMITTING FILAMENT [ i ] [ fcn t' t" ] n [ r ] [ ISO ]
```

Creates "n" rays randomly but uniformly distributed either along previously defined CURVE "i" or the arbitrary curve:

$$X, Y, Z[, W] = \text{fcn}(t) \quad t' < t < t''$$

where the function name is defined by a previous \$FCN command. The last three or four entries created by its expressions are the three coordinates and an optional flux weighting (apodization) value (default 1). The optional "r" is a crosssectional semi-thickness of the wire. For example, the commands for a helical filament might be:

```
--- R=1 radius  T=5 turns  L=5 length
--- $FCN EFCN x R*COS(6.2832*T*_ ) y R*SIN(6.2832*T*_ ) z L*_ w ! 0<_<1
--- EMITTING FILAMENT EFCN 0 1 2000 .1
```

Normally, each segment of the filament emits like a Lambertian cylinder. The ISO option will force each filament point to radiate isotropically.

3.9.4.1.5.5 Emitter Specified By An IES Standard Photometric Data File -

```
--- EMITTING IES name X x y z n [ a ]
                      Y
                      Z
```

Creates a ray set that emits according to the data in an IESNA Standard

Photometric Data file called "name.IES". The third entry specifies the coordinate axis to associate with the file's polar axis ("vertical" if type A or C photometry, "horizontal" if B). "x y z" are the coordinates of the emitter and "n" is the number of rays to created. An optional azimuthal angle "a" (in degrees) can also be specified to rotate the angular pattern around the polar axis. The USERAPOD settings do not effect this command since the equivalent information is found in the IES file.

3.9.4.1.5.6 Arbitrary Emitting Ray Set -

```
--- EMITTING RAYS [ n dx dy dz da db dc ]
--- x y z a,b,c f s d
--- x' [ y' z' a',b',c' f' s' d' ]
--- :
```

Creates an arbitrary set of rays (one line of following input per ray) in terms of their positions, directions, fluxes, sizes, and divergences. Any trailing entries that are omitted take their value from the previous ray. Optionally, instead of each of these defined rays, "n" rays randomly distributed within the given half-intervals about each defined ray can be created.

3.9.4.1.5.7 Emit According To Distribution In A Binary File -

```
--- EMITTING DATA [ file ] [ n ] [ a [ a' ] [ RECT ] ] [ ISO ]
      -DATA      u
      +DATA
```

Creates "n" rays at the current WAVELENGTH based on the distribution stored in "file" (default extension is .DIS) or unit number "u" (default 9). The default for "n" is the total number of samples in the file. This command currently supports the following distributions:

1. General list of rays in full (.DIS) or compressed (.DMP) DUMP command format. If the current WAVELENGTH is zero, then it is reset to the shortest one stored in the DUMP file header.

2. Two-dimensional planar spatial distributions (e.g. SPOTS POS, SPREAD). "a a'" are optional half-angles in degrees for the elliptical or RECTangular cone of emitted radiation centered on the positive normal axis. The default for "a'" is "a", i.e. circular or square light cone. The default for "a" is 90 degrees, i.e. a full hemisphere. By default, the surface emits directionally in a Lambertian fashion or ISOtropically (the sign on the DATA literal determines into which hemisphere). This pattern can be further apodized in direction by the current USERAPOD DIR settings.

3. General three-dimensional volume distributions that emit isotropically in direction (unless USERAPOD ANGLES is set).

4. Volumes with cylindrical symmetry (e.g. ABEL INVERSE). Normally, the far-field radiation pattern is a donut, i.e. it varies as the sine of the angle from the axis. The "ISO" option will yield an isotropic radiation pattern.

As needed, more will be added in the future. Except for the first form, the rays are distributed spatially in a random fashion using the given data as the probability density function.

3.9.4.1.5.8 Composite Single-source Emitter -

```

--- EMITTING
--- w  DISK      ...
      RECT
      ENTITY
      OBJECT
      CONE
      PYRAMID
      BOX
      SPHEROID
      RAYS
      FILAMENT
      HELIX
      IES
      DATA
--- w'  ...
--- w'' ...
--- :
```

Any of the previous EMITTING types can be used in a composite (single source) emitter using the above format. The "w"s are the flux weighting factors for the individual component emitters. Only linear transformation commands (e.g. SHIFT, ROTATE, etc.) are allowed between the individual emitter commands.

3.9.4.1.6 Ray/beam Set For A Coherent Astigmatic Gaussian Mode -

```

--- GAUSSIAN X x  x' x"  n' n"  m' m"  a' a" [ DEGREES ]
              Y y  y' y"                      RADIANS
              Z z  z' z"

```

Creates an astigmatic Hermite-Gaussian field distribution propagating in the specified direction and starting at the given plane. The remaining pairs of parameters are the waist locations, number of rays/beams, mode numbers, and waist semi-widths (or angular semi-divergences) for the two transverse directions. The WAVELENGTH must be previously defined and the field created will have total flux identical to a unit peak-irradiance fundamental mode. Also, the current CLIP POSITION and flux CUTOFF / HALT settings may truncate the distribution. When the beam is the fundamental mode, circularly symmetric, and not astigmatic, then the following short format can be used:

```

--- GAUSSIAN X x  x' n a [ DEGREES ]
              Y y  y'      RADIANS
              Z z  z'

```

3.9.4.1.7 Decomposing An Existing Arbitrary Field Into A New Gaussian Set -

Spatial filtering can be done in ASAP if the truncated field (e.g. FIELD ... CLIP calculation) at a focal plane is resynthesized into a new Gaussian ray/beam set.

```

--- DECOMPOSE [ u ] ...
              n.e

```

This command reads the complex coherent field stored in the given distribution

file (number or name with extension). The default is 29 or BRO029.DAT which is the one normally created by a FIELD command. This field is then decomposed into a new set of Gaussian beams. If the field is a complex vector (i.e. FRESNEL BOTH set before FIELD calculation), then only the component specified by the last POLARIZtion command is decomposed. The new ray set is added to the current and a new source (with the current WAVELENGTH) is created. If immediately preceded by a "RAYS 0" command, the current ray set can be deleted before the new one is created. Negligible beams are not created, i.e. their absolute flux is below the CUTOFF command floating entry or their relative flux is below the HALT command floating entry. For example, to decompose properly a vector field propagating mostly in the Z-direction:

```

      RAYS 0
      POLARIZ X; DECOMPOSE ...
      POLARIZ Y; DECOMPOSE ...

```

```

--- DECOMPOSE [ u ]  POSITION [ x y z CON ] [ ADJUST ]
      n.e  -POSITION          DIV
      +POSITION      a,b,c PLA

```

The POSITIONal decomposition creates a straight forward spatial distribution of beams, one per input field pixel, and should be used only when the pixel size is a few wavelengths or larger. The direction of each beam in the grid is adjusted to be normal to the local phase-front. In order to make this determination more robust, a CONverging or DIVerging spherical (centered at "x y z") or PLAnar (with DIRection "a,b,c") reference wavefront can also be specified. The overlap of the beams is controlled by the first setting on the last WIDTH command. The ADJUST option takes into account this overlap and produces a more accurate decomposition in most cases.

```

--- DECOMPOSE [ u ]  DIRECTION [ m [ a [ a' ] [ RECT ] ] ]
      n.e  -DIRECTION
      +DIRECTION

```

The DIRECTIONal decomposition Fourier transforms the field and creates up to "m" (actual integer number or floating point fraction of maximum possible, default .1) Gaussian beams whose sum closely approximates the original field distribution. The centers and waists of the beams are all located at the center of the original field distribution. The beams all have the same widths (chosen such that they are uniformly distributed in the far field) but different propagation directions. The "a"s are limiting cone angles in degrees (default is 90, i.e. full hemisphere) for the Gaussian directions and should be

set to the acceptance cone of the subsequent optical system. If the RECT option is used, this cone is rectangular instead of the default elliptical. The following technical notes apply:

1. The accuracy of the decomposition is related to the number of beams and their maximum far field angle. As a check, it is a good idea to issue a SPOTS DIRECTION and a SPREAD NORMAL command just after DECOMPOSing and before TRACING the ray/beam set further.
2. If you want the number of beams given to just fill the given angular cone, then (assuming $a=a'$ and a square WINDOW) the spatial sampling interval of the original field must be:

$$d = \frac{\text{sqrt}(m/c) * \text{wavelength}}{N * \sin(a)}$$

where N is the size of the FFT used (set by the last FTSIZE command) and c is either 4 for RECT or PI otherwise. Therefore, the PIXEL setting for the original field should be the WINDOW size divided by d.

3. The overlap of the Gaussians in the far field is controlled as usual by the WIDTH command.

3.9.4.2 Selecting And Modifying The Current Ray Set -

```

--- SELECT [ ALL ]
          ONLY   [ entry entry' [ AND entry entry' ...
          EXCEPT                                OR

```

This command gives the user control over the current set of rays the program is to consider in all calculations and output. No entries or the ALL option forces consideration of all known rays. Particular ray sets are selected by entering pairs of entries separated by a logical operator (AND/OR) according to the following table:

```

entry entry'
-----+-----+-----
  i      j      Ray number is between i through j inclusive
OBJECT  n      Ray comes from previous object n (name or number)
        -n     Ray was scattered from object n
        +n     Ray was split from object n
SOURCE  k      Ray originated from source number k
        -k     Scattered ray from source number k
MEDIA   m      Ray is in MEDIA m (name or number)
        -m     Scattered ray is in MEDIA m
GENERATION n    Ray was split+scattered n times
        -n     Ray was scattered n times
        +n     Ray was split n times
HITS    n      Ray has hit objects n times
        -[n]   "                and incoming to last object
        +[n]   "                and outgoing from last
EVERY   n      Ray number modulo n equals one
PATH    l      Ray belongs to "l"th path from last PATH command
        0      Ray belongs to a path not listed by last PATH command
        w      W      Ray has wavelength greater than w
        W      w      wavelength less than w
        f      F      Ray has flux greater than f
        F      f      flux less than f
        d      L      Ray has optical path length greater than d
        L      d      optical path length less than d
        r      R      Ray has AXIS radial positions greater than r
        R      r'     AXIS radial positions less than r'
        t      T      Ray has AXIS angular positions greater than t degrees
        T      t'     AXIS angular positions less than t' degrees
        x      X      Ray has X coordinates greater than x
        X      x'     X coordinates less than x'
        :      :
        a      A      Ray has X direction cosines greater than a
        A      a'     X direction cosines less than a'
        :      :

```

Remember that the sequence of logical tests specified on the command are applied to each ray individually, one at a time. The final number of rays selected is displayed after all rays have been tested.

```

--- SUBSET [ RESET [ m ] ]

```

Collapses ray storage to include only those rays determined by the last CONSIDER, SELECT, and CUTOFF commands. All other ray information is lost. This command is useful when a lot of rays have been traced and only a small subset will be analyzed extensively. You can optionally RESET the following information:

```

Current object => 0
Previous objects => 0
Number of hits => 0
Specular splits => 0
Scatter levels => 0
Parent ray => itself
Optical Path Length => 0 for ray 1
Current medium => "m" if specified

```

This would be useful for preparing the output from one optical system (e.g. a complex source) to be the input for a completely different one.

3.9.4.2.1 Ray Energy Flux -

```

--- FLUX    a  [ b  [ i j ] ]
      TOTAL      SOU m

```

Changes the fluxes for the currently SELECTed ray set (or the rays specified by the optional fourth and fifth entries) to "a" plus "b" (default=0) times the current ray flux or to produce a TOTAL flux "b". Rays with fluxes less than or equal to the CUTOFF value are ignored by the program in any calculation.

3.9.4.2.2 Local Beam Shapes -

```

--- SHAPE   k [ s ]
            beam
            0 [ n a ] [ a' a" ... ]
            MODE

```

Resets the beam shape to the literal "beam" or integer "k" for the currently

selected ray set. The program currently allows the following beam shapes (the shape is either an amplitude profile for coherent beams or a flux density profile for incoherent beams).

"beam" "k"		BEAM SHAPE
-----+-----+-----		
MODE	0	Complex Hermite-Gaussian beam (default if WAVEL>0)
ELIP	1	Uniform ellipse (default when WAVEL=0), central hole ratio s
GAUS	2	External real Gaussian
RECT	3	Uniform parallelogram with central hole ratio "s"
BELL	4	Bell-shaped (cosine squared) approximation to Gaussian
LRNZ	5	General Lorentzian (inverse "s" asymptotic falloff)
SINC	6	Sinc (sinx/x) with Fourier central hole ratio "s"
SECH	7	Sech (hyperbolic secant)
SOMB	8	Sombrero or Airy disk function with Fourier central hole "s"
FIBR	9	Fundamental mode of circular fiber at normalized frequency "s"
SLAB	10	Fundamental mode of symmetric waveguide at normalized frequency "s"

The sign of "k" (or "beam") determines whether the beam will be incoherently added (intensity summing) or coherently added (complex field summing) to other beams of similar designation.

- k < 1
- Coherent beam
- k > 0
- Incoherent beam

For MODE or "k" equal to 0, "n" is the number of higher-order Hermite-Gaussian beam modes and the "a"s are the corresponding complex amplitudes (see RAYSET command for mode ordering).

3.9.4.2.3 Folding Over Ray Coordinates Due To Symmetry -

```
--- BILATERAL POSITION X [ Y,Z ]
                        DIRECTION Y [ X,Z ]
                                Z [ Y,X ]

OFF
```

In systems with bilateral symmetry the above commands can be used to reduce the number of rays traced and still produce meaningful plots from the SPOTS command. Any ray, whose data specified by the second and third is negative, is

mirror imaged about the zero coordinate plane so that the particular data becomes positive. For systems with symmetrical quadrants, two coordinate directions should be entered.

3.9.4.2.4 Moving Along Straight Ray Paths To A New Location -

```
--- MOVE BY  d
          TO  X  d
              Y
              Z
          OPL
```

Moves the ray data to a new reference surface either BY an amount "d" (relative move) or TO a position "d" (absolute move) along the individual ray directions, given coordinate direction, or Optical Path Length.

```
--- MOVE TO POINT x y z
```

Moves rays along their trajectories to their closest approach to the given point "x y z".

```
--- MOVE TO SPHERE [ r [ x y z ] ]
          PLANE [ x y z ] [ a,b,c ]
```

It is also possible to transfer the current ray/beam set to the surface of a SPHERE of radius "r" centered on the point "x y z" or a PLANE with normal "a,b,c" through the point "x y z" by using the above variation of the MOVE command. If none of the SPHERE data is given, a best-fit spherical wavefront is used. The default for the normal to the PLANE is the average of the current base ray directions. The default point is the average of the current base ray coordinates. The Peak-to-Valley and RMS optical path difference on this surface are also printed so that normally it represents either the reference sphere of a focussing system or the reference plane of an afocal system.

```
--- MOVE TO FOCI [ n n' n" ... ]
```

Moves each ray/beam to the centroid of the intersections of the given parabal rays with the base ray.

--- MOVE PARABASALS

Moves the parabasal rays for each ray / beam to its base ray plane. This does not in any way change the beam characteristics but does affect the PLOT BEAMS display since by default the parabasals lie on the current object surface.

3.9.4.2.5 Reversing Ray Directions -**--- REVERSE [OPL]**

Reverses the directions of the currently selected rays. Useful for doing "virtual" ray traces. The OPL option signals the program to also start subtracting optical path lengths instead of adding them for ALL rays.

3.9.4.2.6 Imaging Ray Positions Through A Lens -**--- IMAGE l k k'**

Images the current ray positions (not directions) from space "k" to space "k'" of LENS entity "l". The spaces are numbered from 1 (before the first conicoid) to N+1 (after the last conicoid N). It uses the auxiliary axis technique that steps the image through the centers of curvature of the conicoids (the aperture semi-diameters, conic constants and obscuration ratios of the conicoids do not affect the imaging) and always produces an image even if it is virtual or unphysical. The resulting imaging transform is stigmatic (points go into points) but not necessarily collinear (lines go into lines). Therefore, it is only an approximation since in any real optical system, the image will be aberrated (not a perfect point focus). This is most useful for mapping a grid of rays at an internal stop position of an optical system into its object space (i.e. entrance pupil). For example,

```

LENS 99; ... !equivalent imaging train
:
GRID ... !covers internal stop
    IMAGE 99 4 1 !image into entrance pupil
SOURCE DIR ... !now set directions
    MOVE TO ... !out in front of first surface
TRACE

```

3.9.4.3 Execution Of Trace -

```

--- TRACE [j j'] [STEP [k]] [LIST [n]] [GRAPH [n  ]] [STATS [m]] [KEEP]
          DIR      PLOT    [d]    ACCUM

```

Traces the current ray set in storage through the system starting at object "j" and stopping at object "j'" (the defaults are "0 0", i.e. from all objects to all possible objects). The sign of "j'" will determine if a refraction or reflection calculation will be done at the stopping object surface. In particular, a negative "j'" results in no such calculation. Alternatively, the rays can be STEPped "k" objects intersection. Again, the sign of this entry determines whether one steps through the object (+k) or up to the object (-k). The default for "k" is -1. The LIST option should be used if object-by-object ray position data printout is desired (use DIR instead if you want ray directions also). Otherwise, no output is generated except for possibly warning messages. The PLOT option will plot the rays as they are being traced and also store their vectors on the 3D graphics file for later plotting. The numerical option "n" specifies that only every nth ray is PLOTted and / or LISTed. Optionally, only those rays whose starting depth coordinate lies within a distance "d" of the zero plane will be PLOTted. The STATS or ACCUM option keeps track of ray statistics (base and up to the first "m" parabazals where the default is all) for each GRIN / absorbing media, scatter model, and object encountered during the trace and in the case of STATS prints them out on completion of the trace. Instead of PLOTting the rays, the progress of the trace can be monitored GRAPHically. In this case "n" is the maximum ray number magnitude (default is 6, i.e. 1 million rays). By default, the first ray definition command (e.g. GRID, EMIT) executed after tracing automatically flushes any rays (i.e. RAYS 0). To override this behavior, use the KEEP option on the TRACE command.

3.10 ANALYZING THE CURRENT RAY AND BEAM DATA

3.10.1 Listing Data On A Ray-by-ray Basis

```
--- LIST POSITION [ k ]  
    P#  
    DIRECTION  
    D#  
    RAYS  
    SOURCES  
    INTEGER  
    ELLIPSE
```

List the ray data specified by the second literal entry for the current ray set. By default the positions or directions of each base ray is used. Any particular parabal ray may be selected by specifying its number "#" (i.e. "P0" means base ray position, "D1" first parabal ray direction, etc.). If "#" is an "S", then the differences between the base ray and all the parabal rays are also listed. The flux, size, and optical path length of each beam is listed along with the coordinate data. "k" specifies an optional ray number whose coordinates and optical path length are subtracted from each base ray before they are listed. The CONSIDER and SELECT commands can be used to restrict the list to rays that are currently on certain objects.

The RAYS form lists the position, direction, flux, size, and divergence of each ray/beam in a format compatible with the input to the EMITTING RAYS command.

The SOURCE data includes the wavelength and ray numbers for each source. The INTEGER data includes the current medium, split level, source number, number of objects, current object and previous starting objects for each ray. Actually, in order to reduce the output, the program only lists the break points where the integer data changes from one ray to the next.

If FRESNEL is set to BOTH, the major axis orientation (in global coordinates) and minor-to-major axis ratio (ellipticity) of the polarization ELLIPSE for each ray can also be listed. The sign of the ellipticity determines the "handedness" of the polarization state (zero corresponds to pure linear).

3.10.2 Listing Data On The Extreme Rays

```
--- EXTREMES POSITION [ k ]  
    P#  
    DIRECTION  
    D#  
    FLUX  
    OPL
```

Determines the extremes in the ray data specified by the second literal entry and lists the corresponding rays. By default the positions or directions of each base ray is used. Any particular parabasal ray may be selected by specifying its number "#" (i.e. "P0" means base ray position, "D1" first parabasal ray direction, etc.). The flux, current object, and optical path length of each extreme ray is listed along with all base coordinate data. "k" specifies an optional ray number from which the extremes are measured. If "k" is less than zero, the average of the ray data is used.

3.10.3 Statistics On Current Ray Set

```
--- STATS [ ALL      ]  
    POSITION  
    P#  
    DIRECTION  
    D#
```

By default this command produces a one line per object summary of the flux and number of rays on each object (ALL considered objects even those with no rays and zero flux on them). If the second entry is specified, it also produces a statistical analysis of the positional or directional ray data including centroids, full variances (RMS deviations), and maximum spreads in each coordinate direction on each object. By default the positions or directions of each base ray is used. Any particular parabasal ray may be selected by specifying its number "#" (i.e. "P0" means base ray position, "D1" first parabasal ray direction, etc.). The actual objects analyzed may be controlled by the CONSIDER command.

3.10.4 Statistics On Previous Trace

```
--- STATS TRACE [ MEDIA ] [ MODELS ] [ OBJECTS ]
```

If the previous TRACE had a STATS or ACCUM option on it, then selectively print the given statistics. For OBJECTS (the default), only those currently CONSIDERed are printed.

3.10.5 Grouping And Listing Current Ray Set By Paths

```
--- PATHS [ p ] [ n ] [ AVERAGE ] [ m ] [ SHORT ]  
      PEAK          MEDIUM  
      TOTAL          LONG  
      OBJECT
```

Prints out a table of ray paths according to current, previous, and initial objects. A value of "p" greater than zero causes the program to sort the paths by AVERAGE irradiance, PEAK irradiance, TOTAL flux (default), or OBJECT flux and print out only those paths whose percentage contribution is greater than the decimal number "p". Any paths containing less than "n" rays are also not printed. The value of the integer "m" determines whether the above flux related quantity for each distinct path is the minimum (-1), sum (default 0), or maximum (+1) of the individual contributions. The length of the printed output can be SHORT (all integer object numbers), the default MEDIUM (current and previous objects integer) or LONG (all decimal object numbers).

In the list of "Split/Scatter" objects, a scattered ray generation is indicated by a negative number. Except for SHORT output, the three digits following the decimal point can be used to determine more precisely the mechanism that generated that ray at that object. The code numbers and their meanings are as follows:

FIRST DIGIT	SECOND & THIRD DIGITS
0 Random Scatter	00 Hemispherical ## TOWARDS number
1 Ordinary Reflection	## DIFFRACT number
2 " Transmission	"
3 Deterministic Scatter	00 Retro-reflected
"	01 Near Reflected
"	02 Near Transmitted
4 Extraordinary Reflection	## DIFFRACT number
5 " Transmission	"
6 Volume Scatter	## MEDIA number

3.10.6 Complete Positional Ray Histories

```
--- HISTORY [ k ] [ PLOT [ m ] ]
                SPOTS
                LIST
```

If a SAVE file number "k" exists from the last TRACE, this command can be used to LIST (the default) or PLOT the reverse trajectories of every "m"th ray (default 1) in the current ray set. Alternatively, just the SPOTS where the rays struck the objects can be plotted instead of the trajectory lines.

3.10.7 Best Focus Determination

```
--- FOCUS [ MODE [ m ] ] [ MOVE [ r ] ]
```

Determines the "best" 3-dimensional focal point for the current ray bundle

using one of the following methods (listed from fastest to slowest):

MODE	METHOD
-----+-----	
m<0	Centroid of individual ray foci
m=0 (default)	Least squares closest approach of base rays
m>0	Above plus first "m" parabasal rays

Also calculates the RMS deviation of the rays from this point. The MOVE option causes the rays to be transferred to a plane that passes through this focal point. The SPOTS command can then be used to generate a spot diagram at this "best" focus. The optional "r" entry is the radius of a reference sphere centered on the best focal point to which the ray data will be moved. The Peak-to-Valley and RMS OPD (optical path difference) is then printed.

3.10.8 Obtaining A Full Field Aberration Display

```
--- FFAD [ SPOTS [ s ] ] [ REFERENCE w [ k ] ]
```

This command produces a real Full Field Aberration Display. This is a plot of the 3D best-focussed RMS spot ellipses for each field position entered on the SOURCE command. The ellipses are scaled up so that the largest one just touches the spot from an adjacent field position. These plots are useful for identifying aberration nodes on the image surfaces of non-centered or perturbed systems. The corresponding unfocussed ray SPOTS can also be plotted. "s" (default=.3) is an additional scale factor used to make room for the spot patterns since they will always be larger than the RMS ellipses. The REFERENCE option specifies the width (in system units) of a square that will be drawn around the smallest (or "k"th) spot. This command also prints out the statistics of the spots including the best-fit planar and curved surface parameters.

3.10.9 Accessing Internal Ray Data Directly

```
--- GET [ k [ k' ] ]
```

Get the current data for all rays, ray "k", or rays "k" to "k'" and place it into the input registers (see sections in Chapter 2 on these registers). If more than one ray selected, then registers contain flux-weighted averages (except total flux). The register assignments are as follows:

REGISTER	LITERAL	RAY/BEAM DATA
-----+-----+-----		
A0,B0,C0	X,Y,Z_DIR_B	Absolute X,Y,Z direction cosines of base ray
Ai,Bi,Ci	X,Y,Z_DIR_i	Relative direction vector of ith parabalasal ray
D0	OPL	Optical path length from start of base ray
E1,E2,E3	X,Y,Z_EPOL	Components of unit polarization vector
F0	FLUX	Total flux in ray/beam
G0	DIVERG	Average divergence angle of beam
H0	HEIGHT	Average height of beam centered on base ray
Ii	PREV_0_i	ith previous split object for ray/beam
J0	SOURCE	Source number from which ray/beam originated
K0	CURR_OBJ	Current object at which ray/beam is located
L0	HITS	Total number of surfaces ray has hit (intersected)
M0	MEDIUM	Medium that ray/beam is in
N0	SPLITS	Number of times ray/beam has been split
N1	LEVELS	Number of times ray/beam has been scattered
P0	POLAR_0	Relative modulus of fundamental beam mode
P1,P2	POLAR_1,2	Relative moduli of polarization components
Q0	NUM_RAYS	Total number of ray/beams
Q1	NSOURCES	Total number of original sources
R0	PARENT	Number of ray from which this ray was split (parent)
S0	SHAPE	Beam shape number (see SHAPES command)
S1	FACTOR	Beam shape factor or number of higher modes
T0	PHASE_0	Relative phase angles of fundamental beam mode
T1,T2	PHASE_1,2	Relative phase angles of polarization components
U0,V0	U,VPARAMB	Parametric coordinates of base ray position
W0	WAVELEN	Wavelength of ray/beam
Wi	WAVLNS_i	Wavelength for ith source
X0,Y0,Z0	X,Y,Z_POS_B	Global X,Y,Z coordinates of base ray
Xi,Yi,Zi	X,Y,Z_POS_i	Relative coordinates of ith parabalasal ray

This data can then be used in any future commands and can even be modified if

the following command is used.

```
--- PUT k
```

Puts the current values in the input registers back into the storage locations for ray "k".

3.10.10 Geometrical Spot Diagrams

```
--- SPOTS POSITION [ u ] [ATTRIBUTE i] [OBJECT] [NUMBER [s]] [EVERY n]
      Pc          ADD
      DIRECTION
      Dc
```

For current ray set, this command produces a "spot diagram" of the ray data specified in type by the second entry and in coordinate directions by the last WINDOW command. By default the positions or directions of each base ray is used. Otherwise, the last character of the second entry can be used to select the following specific data:

```
      c          Coordinates Plotted
---+-----
      Current base ray (default)
      S All the current parabasals
      0 Initial base ray
      1 First parabasal only
      2 Second parabasal only
      :
```

The flux data for each pixel written to logical unit "u" in unformatted direct access binary records whose length is the same as the current PIXEL setting. If "u" is negative, then any previous data on the specified unit is overwritten. Otherwise, the flux data is added pixel by pixel to what's already in the file. The default value of "u" is -9. ADD is equivalent to +9. This file can be manipulated, plotted, and named with the DISPLAY command and its associated subcommands.

With the ATTRIBUTE option, the ASCII equivalent to "i"+48 is printed where the

ray strikes instead of the flux sum of all rays in the pixel. It also controls what symbols (instead of the default dot) are used for the spots on the plot, i.e.

ATTRIBUTE	Symbol
-----+-----	
-1	Connecting line
0	No plot generated
1	Plus sign
2	X cross
3	Up triangle
4	Down triangle
5	Square
6	Diamond

With the OBJECT option, the color associated with the ray’s current object (instead of the ray’s source) is used for plotting the spot. In addition, the distribution file will contain object number information and not flux information.

Optionally, the ray’s NUMBER can also be drawn on the plot. "s" an optional scale factor for the character size to be used. Also, only EVERY "n"th ray can be plotted instead of all the currently selected ones (which are still summed in the distribution file).

3.10.11 Full-Spherical Far-field Incoherent Radiant Intensity Or Radiance Pattern

```
--- RADIANT [ X [ f f' n a a' n' ] ] [ MAP [ s ] ] [ LOW ]
              Y              AREA              HIGH
              Z
            -X
            -Y
            -Z
```

Normally calculates in spherical coordinates the far-field incoherent radiant intensity (flux per solid angle) pattern of the current ray set. The second entry specifies the "polar" axis of the coordinate system (default Z). "f f'

is the range in degrees for the angles from this axis (zenith / latitude angles). The defaults are 0 to 180 degrees. "n" is the number of subdivisions of this angular range (default 36, i.e. 5 degree increments). "a a" is the range in degrees for the angles about the axis (azimuth / longitude angles). The defaults are -180 to 180 degrees. "n" is the number of subdivisions of this angular range (default 36, i.e. 10 degree increments). The pattern is written to the standard distribution file "BRO009.DAT" and can therefore processed further by the DISPLAY commands (e.g. DIRECTION or MESH will produce a 2D or 3D polar diagram of the radiation pattern).

The MAP or AREA option will also use the current PIXELS and WINDOW settings to calculate the full 4-dimensional pattern, i.e. a function of not only the two angles but also the two spatial coordinates. MAP outputs radiant intensity (flux per solid angle) while AREA outputs radiance (flux per solid angle per projected area). The 3D polar-diagrams for each point on the spatial grid will be written to the current VECTOR file (use REPLOT or \$VIEW to see this data) as either simple wireframes or LOW / HIGH resolution shaded "surfaces". "s" is an optional rescaling factor for altering the sizes of the polar diagrams relative to the spatial grid spacing. If the angular sampling is 1x1, then the radiant intensity or radiance map for this solid angle is written to the standard distribution file "BRO009.DAT".

3.10.12 Collection Efficiency As A Function Of Aperture Size And Cone Angle

```
--- COLLECTION [ PERCENT ELLIP HEIGHT m ANGLE n a ]
                  FLUX      RECT  AREA      SINE  MAX
                  GCF
                  OMEGA
```

Calculates the collection PERCENT efficiency or FLUX of the current rayset for "m" (default 100) spatial samples and "n" (default 90) direction samples up to the angle "a" in degrees (default 90 or rayset MAX). The spatial limits are determined from the current WINDOW setting and are either its full RECTangle or the inscribed ELLIPse. The actual functional coordinates can be:

HEIGHT	Geometric-mean height of the RECTangle or ELLIPse (0 to max)
AREA	Actual area of the RECTangle or ELLIPse (0 to full area)
-----+-----	
ANGLE	Actual angle in degrees from the depth axis (0 to 90)
SINE	Sine of the angle (0 to 1)
GCF	Sine squared of angle (0 to 1)
OMEGA	Solid angle (0 to 2 pi)

The first choice listed for each literal item is its default if it is left off the end of the command. The result is written to the default distribution file (BRO009.DAT) and can then be processed by the DISPLAY command and its subcommands.

3.10.13 Interpolating OPD Data And Generating A Normalized PSF

```
--- OPDMAP [ name ] [ d ] [ PSF [ f ] ]
```

Interpolates optical path lengths (of the first source encountered in the current ray set) into a rectangular grid for storage into a distribution file with "name" and extension "DIS" (default BRO009.DAT). The data can then be manipulated and plotted using the DISPLAY commands. The average OPD value (piston) is always subtracted out. In order to get rid of focus and tilt errors, it may be necessary to precede this command with a MOVE TO or FOCUS MOVE command. If the "d" entry is entered as zero, the program generates a fast discrete map of the wavefront with no interpolation. Otherwise, the program uses a very general (and somewhat slower) linear interpolation (no extrapolation) technique that considers only a small set of ray points within a distance "d" (default is 3 times smallest ray separation) of each grid point.

With the PSF option, the program forms the complex geometrical pupil function from the interpolated OPD map and stores its Fourier Transform (i.e. the coherent Point Spread Function) on unit 29 (BRO029.DAT). "f" controls how much of this PSF to return. If it is less than one, it is a fractional energy (relative to the maximum) cutoff. If greater than one, it is the maximum distance in Airy units (default 5). This PSF is normalized relative to an ideal circular aperture of equivalent area. A "DISPLAY 29 ENERGY" command can then be used to manipulate and plot this far-field distribution.

3.10.14 Dumping Currently Selected Rays To A Binary File

```
--- DUMP [ name ]  
      +
```

Dumps the currently selected rays to the binary distribution file "name" (default LASTDUMP.DIS) or appends ("+") to the previous DUMP. Only the bare essential information about each ray is written to the file: position coordinates, direction cosines, flux (size and divergence if XMEMORY not MIN). This file can be efficiently read back in using the EMITTING DATA command.

3.10.15 Standard Distribution Clipping Option

```
--- ...CLIP [ i ]  
      -j  
      +j
```

This option when used on a SPOTS, OPDMAP, SPREAD or FIELD command will clip the distribution by either the specified LIMITS/BOUNDS of OBJECT "i" or the given side of entity "j". A "-" means the interior of a closed EDGE or the negative side of a SURFACE. A "+" means the exterior of a closed EDGE or the positive side of a SURFACE. Data points outside the specified regions have their distribution values set to zero.

3.10.16 Creating Distributions In The Local Coordinates Of An Entity

The combination of an AXIS LOCAL command and a window override on a SPOTS, RADIANT, OPDMAP, SPREAD or FIELD command will force the distribution to be calculated in a local Cartesian coordinate system.

```

--- AXIS LOCAL
--- SPOTS          ...window
    OPDMAP
    SPREAD
    FIELD
    RADIANT ... MAP
                AREA

```

To distinguish this from the global case, the coordinate axes in the header of the resulting distribution file (or files) will be in lower case. Also, the Euler angles and positional offsets needed to transform the distribution back into global coordinates will be stored in the header. If a reference object is not specified on the **AXIS LOCAL** command, then the distribution is in the local coordinates of the base entity associated with the object that the first valid ray/beam resides on.

3.11 COHERENT/INCOHERENT FIELD/ENERGY FROM CURRENT BEAMS

Using a (usually Gaussian) beam summation technique, the coherent/incoherent field/energy can be calculated on one or more uniform planar grids. The beams and grid points are always assumed to lie in the same homogeneous (no GRIN) and isotropic (no CRYSTAL) medium.

```

--- IRRADIANCE [ OFF ]
    X
    Y
    Z

```

By default, the **SPREAD** and **FIELD** commands calculate the energy or peak flux density relative to the local propagation direction at each sample point. This command will cause any future **SPREAD** or **FIELD** commands (until it is turned OFF) to calculate the **IRRADIANCE** (flux per unit area) relative to a given fixed direction for all sample points. If no entry is given, the normal to the sample grid is used.

```

--- VIOLATION g [ g' [ p [ p' [ s [ s' ] ] ] ] ]

```

Controls the handling of paraxial departure, positivity, and stability

violations during future SPREAD and FIELD commands.

Entry	Default	Description
g	.1 waves	Paraxial invariant departure before message issued
g'	1 waves	Paraxial invariant departure before calculation terminated
p	.01	Positivity violation before message issued
p'	1	Positivity violation before calculation terminated
s	.1	Stability violation before message issued
s'	10	Stability violation before calculation terminated

Setting these thresholds higher than the defaults may allow a calculation to finish but the results should be viewed as suspect.

```
--- SPREAD DIRECTION [ u ] [ DOWN d d' ] [ COLENGTH l ]
      POSITION      ADD
      APPROX
      NORMAL
```

Calculates the coherent/incoherent energy distribution for the current beam set in the directions given by the last window command. The particular calculation is selected by the second entry. They are (listed from fastest to slowest):

	FUNCTIONAL COORDINATES	OPTICAL METHOD	ASSUMED COHERENCE	UNITS
DIRECTION	d-cosines	rough geom	incoherent	Flux/Ster
POSITION	positions	rough geom	incoherent	Flux/Area
APPROX	positions	rough wave	coh/incoh	Flux/Area
NORMAL	positions	exact wave	coh/incoh	Flux/Area

If APPROX method is selected or the number of PARABASAL rays is 0, only symmetric beams are used, i.e. individual astigmatic effects are ignored. The NORMAL method is exact only if at least 4 PARABASAL rays were created and /or traced. If there are multiple wavelengths in the current ray set, the program automatically cycles through each distinct wavelength and incoherently (intensity) sums it with any other wavelengths.

The flux data for each pixel is written to logical unit "u" in unformatted direct access binary records whose length is the current PIXEL setting. If "u" is negative, then any previous data on the specified unit is overwritten. Otherwise, the flux data is added pixel by pixel to what's already in the file.

The default value of "u" is -9. ADD is equivalent to +9. This file can be manipulated, plotted, and named with the DISPLAY command and its associated subcommands.

The WINDOW and PIXELS commands affect the sampling used. For this calculation to take place, a WAVELENGTH (and possibly a PARABASAL) command must be entered before any ray definitions. The DOWN option can be used to further restrict the calculation to a fractional range given by "d" to "d'" relative to the current setting of the window in the down (horizontal) direction. For example, if $d=d'=.5$, then only the one-dimensional profile of the energy pattern along the center line of the current window will be calculated.

The COLENGTH option can be used to incoherently sum a subset of beams that would normally be coherently summed. Any beam whose optical path length is different from that of the first valid beam by the coherence length "l" will be incoherently summed. The default for "l" is a very large number, i.e. coherent summation for those beams whose shape factors are less than or equal to zero.

```

--- FIELD ... MODULUS ... [      ADD      ] [DELTA [t]] [CONTOUR k
      PHASE          MULTIPLY                c c' [c" ...]
      AMPLITUDE    COUPLE [r v [a]]
      WAVEFRONT
      REAL
      IMAGINARY
      ENERGY
      NONE

```

The original complex vector field is written to unit 29 (file BRO029.DAT). If the ADD option is present, then it is added to the field in the existing file. If the COUPLE option is present, the calculated field is coupled to the field previously stored in the file (i.e. the field in the file is multiplied by a single complex coupling coefficient calculated from the overlap integral of the two fields). Alternatively, one can directly COUPLE to the fundamental mode field of a circular fiber of core radius "r", normalized frequency "v", and optional gradient index power "a" (e.g. 2 for parabolic, absent for step index). If MULTIPLY is specified, the result is the product of the two fields. In all three cases, the two fields must be calculated under the same conditions, i.e. WAVELENGTH, PIXEL, and WINDOW settings.

A constant phase shift "t" (in cycles at the current WAVELENGTH) can be added to each beam. This is equivalent to calculating the field at a time other than

t=0. If this DELTA option is present and a vector field is being calculated (i.e. FRESNEL BOTH set), then the program will produce a plot of polarization ellipses or if "t" given, arrows representing the relative magnitude and direction of the instantaneous electric field at each sample point.

An additional CONTOUR plot can be generated with fractional contours "c c' ..." relative to the full range of the function being plotted. Alternatively, "k" equally spaced contours can be specified. If "k" is less than zero, a "grey scale" plot is produced instead of the line contour plot. And finally, if NONE is not selected, the scalar distribution is written to logical unit 9 and thus can be manipulated, replotted, and named by the DISPLAY commands.

3.11.1 Summing Coherent Beams (possibly Polychromatic And/or Polarized)

```
-- FIELDSUM component [ f [ f' n ] ] ...options
-- [ l l' m m' x y z x' y' z' x" y" z" ]
```

This command is a more general implementation of the (Gaussian) beam summation technique than the SPREAD NORMAL method. It sums ALL coherent beam contributions from the current ray set including multiple wavelengths. Any component of the resulting complex scalar or vector field can be calculated and displayed on an orthogonal or arbitrarily oriented skew planar grid.

Normally the grid is determined by the last WINDOW and PIXELS commands and the third depth coordinate value "f" (or a range from "f" to "f'" in "n" steps). If "f" is omitted and the next command starts with a number, then the grid information is read from that command. This skew planar grid is specified by an origin position vector (x,y,z) and two direction increment vectors, (x',y',z') and (x",y",z"). The integer ranges of the two grid coordinates are given by l to l' for the first direction and m to m' for the second. In other words, the actual grid coordinates in global coordinates are given by:

$$\begin{aligned}x(i,j) &= x + ix' + jx'' & i=l, l' \quad j=m, m' \\y(i,j) &= y + iy' + jy'' \\z(i,j) &= z + iz' + jz''\end{aligned}$$

The total number of grid points must not exceed the following:

Version	Scalar		Vector (FRESNEL BOTH)	
	-----+-----		-----+-----	
Basic	736x736/2 =	270848 ~ 520x520	736x736/6 =	90282 ~ 300x300
Pro	2896x2896/2 =	4193408 ~ 2047x2047	2896x2896/6 =	1397802 ~ 1182x1182

A printed map of the desired field component will be displayed if the GRAPHICS unit is active (see \$IO command).

3.11.2 Field Propagation Through Small Geometries And Inhomogeneous Media

There is a second version of the FIELD command that does not use any of the current ray / beam data.

```
--- FIELDBPM u component [ d [ n ] ] [ BC [ i [ i' ] ] ] ...options
                n.e                f f'                i j i' j'
```

Takes the field stored in the complex distribution file number "u" or name (with extension) "n.e" and propagates it a distance "d" (or from "f" to "f'") in the depth direction by directly solving the partial differential harmonic wave equation (the scalar or semi-vectorial Helmholtz equation). "n" is the number of equally-spaced intermediate propagation locations at which the field distributions will be written to the default binary files (BRO009.DAT and BRO029.DAT). The default is zero, i.e. just the final location.

A unique FD-BPM (Finite Difference Beam Propagation Method) technique is used that is not only fast and accurate but automatically determines the best reference refractive index and axial step size for the given media and lateral sampling. The field is assumed to start entirely within the current IMMERSE medium but will then interact with any other media it encounters. All media can be complex (ABSORBing), inhomogeneous (GRIN), and birefringent (CRYSTAL if the optical axis is parallel to one of the global coordinate axes). Any object geometry (including surfaces imported from CAD) can act as interfaces between the media, but the actual reflection and transmission coefficients of these objects are ignored. Whenever possible, order the media on each object's INTERFACE command so that the first media is before the surface and the second is after (relative to the positive X, Y or Z depth axis direction). Also, the two INTERFACE media entries must be separated by only a single comma "," to indicate that there is a preferred order to them. This permits nearly all

"MEDIA mismatch" errors to be fixed quickly and automatically.

To prevent spurious reflections at the edge of the finite computational window, a physical (BC option absent) or numerical (BC option alone) absorbing region is implemented by default that involves the outer 1/6 pixels at each edge of the window (i.e. 1/3 of the pixels in each lateral direction). Otherwise, the following explicit boundary conditions can be specified on the BC option for one ("i") or both ("i j") directions of the computational window (unprimed entries first end of window, primed second):

B(oundary)C(ondition)	at edge of computational window
<-1	Number of pixels in physically absorbing band
-1	Periodic
0	Zero field
1	Totally reflective (zero derivative)
>1	Number of pixels in numerically absorbing band

The normally integer "i"s and "j"s can also be entered as floating point fractions of the number of samples in the respective direction.

The specific BPM algorithm used is capable of operating outside the normal "paraxial" region (small departures from the factored plane wave) because it expands the lateral differential operator into higher order terms. The order number is controlled by the current ACCURACY setting and the possibilities are listed (from fastest to slowest) below:

ACCURACY setting	Operator order	Max Angle from axis	Comments
LOW	1	10	Fresnel/paraxial/weakly-guided approximation
MEDIUM	2	20	
HIGH	3	40	Wide-angle and/or large refractive index variations

There is a basic limitation inherent in the BPM method: it only tracks the "forward" propagating component of the field and thus ignores back-reflected waves and any flux lost to them. This means that it does a good job predicting the relative shape and phase of the main field but not necessarily the absolute magnitude. The advantage of the BPM method is that it is relatively efficient at propagating wavefields through three-dimensional volumes versus a more rigorous Time-Domain (TD) method that is essentially four-dimensional and thus orders of magnitude slower for the same size problem.

The accuracy of the calculation is most directly affected by the lateral sampling used, i.e. the number of PIXELS for a given size WINDOW. Many problems will require sub-wavelength sampling which means that one is practically limited to small volumes. A good rule of thumb is to use a sample spacing of one-third the shortest wavelength (the vacuum wavelength divided by the highest index in the volume). Some problems may allow coarser sampling while others will require finer sampling than this. The best indicator of accuracy is the two numbers displayed under the "Relative Flux" heading during the propagation. For a non-absorbing (actual and/or numerical) problem, the closer these numbers stay to unity during the entire propagation, the more accurate the final result. To reduce their fluctuation, decrease the lateral sample size (e.g. increase the number of PIXELS). Be careful though, the number of longitudinal steps required may increase dramatically along with run times.

As with virtually all types of numerical simulations, there is a tradeoff between speed and accuracy. It is suggested that the optimum sampling for a given problem be found first with a fast two-dimensional version (negligible width of second WINDOW direction) before proceeding to the full three-dimensional calculation. For three-dimensional problems with cylindrical symmetry, a fast radially symmetric calculation can be done by using the following command sequence:

```
WINDOW Y 0 rmax X -tiny tiny
FIELDSUM ENERGY ... !create 1D starting field
AXIS Z
FIELDDBPM 29 ENERGY dist ... !propagate a distance
```

where in this example Z is the symmetry and propagation axis.

3.11.3 Displaying Data Distributions

```
--- DISPLAY [ u [ MODULUS ] [ s ] ]  
           name PHASE  
                AMPLITUDE  
                WAVEFRONT  
                REAL  
                IMAGINARY  
                ENERGY  
                1ST  
                2ND  
                3RD  
                4TH  
                :  
name . BMP RED  
            GREEN  
            BLUE  
            MONO
```

The distribution data on logical unit "u" (default=9) or file "name.DIS" is read into memory and can be modified and / or displayed with the following sets of commands. If a BMP bitmap file is given, it is first translated to distribution file format (see description of BMP2DIS utility). A menu of all the DIS (or BMP) files in the current directory can be displayed with the following commands:

```
        DISPLAY "*"
or DISPLAY _
        DISPLAY "*.bmp"
or DISPLAY _ .BMP
```

If the distribution is multivalued (e.g. the 6-component complex vector that can be created by a FIELD command), the first value is extracted by default. Otherwise, a specific component given by the third entry can be extracted for processing. If the distribution is three-dimensional, then "s" is either an absolute (integer) or fractional slice number (the default is the last two-dimensional slice).

3.11.3.1 Data Processing -

```
--- NORMALIZE [ d [ c [ c' ] ] ] [ 'flabel' ]
                MAX  MAX
```

Renormalize distribution data (divide functional values by) given value "d" or data MAXimum. The default is the value from the last NORMALIZE command. The coordinate ranges can also be normalized in a similar manner by the optional third and fourth entries. Use the "flabel" option to relabel the functional data.

```
--- FORM [ f ] [ 'flabel' ]
```

"f" controls the form of the data to be plotted (default=1). If it is greater than zero, the data is raised to that power before plotting. If less than zero, the common logarithm of the data is plotted with "f" setting a lower cutoff relative to the data maximum. If the distribution has any negative values in it, a bias is added to make all values greater than or equal to zero before the requested operation takes place. Use the "flabel" option to relabel the functional data.

```
--- FFT [ f ] [ AMPLITUDE ] [ SIZE [ m ] ] [ 'flabel' ]
      n      PHASE
            MODULUS
            WAVEFRONT
            REAL
            IMAGINARY
            ENERGY
```

The FFT option generates the spatial frequency spectrum of the data. It does this by applying a Fast Fourier Transform to a M-by-M block of data points where M is the current Fourier Transform size (2 raised to the "m"th power or as set by FTSIZE command) minus one. If the original data area is smaller, then the data is centered and any excess is zero filled. If the original data area is larger, then one must make sure that most of the energy lies in the first M-by-M block. The program then takes the selected part (default MODULUS) of the center n-by-n or f*M-by-f*M area of the FFT as the current data set. If "f" is an integer "n" greater than one, then it is the actual number of pixels to use. The default for "n" is calculated by the program to maintain the original data size. If "f" or "n" is entered as a negative number, an inverse FFT operation takes place. Applying this command to the diffraction image of a point object generated by a SPREAD or OPDMAP command produces the classical MTF

(Modulation Transfer Function) of the system. The spacing between sample points after the FFT will be related to the original spacing by:

$$S' = ((M+1)*S)^{-1}$$

Use the "flabel" option to relabel the functional data.

```
--- AVERAGE [ i [ j ] ] [ WEIGHT ]
```

This option smooths the data before displaying it by averaging over "i" (default 1) adjacent pixels in the first direction and "j" (default=i) in the second. For example, i=1 and j=1 replaces each pixel with the average of the 3x3 box of pixels centered on it. i=2 and j=3 uses a 5x7 box and so on. This operation is also equivalent to convolving the distribution with a finite window. Optionally, a WEIGHTed average can be done assuming that the error in each pixel within the given window is proportional to the square root of its value, i.e. Monte Carlo statistics. If "i" and/or "j" are entered as negative numbers, then the median value within the smoothing window is used instead of the normal average value.

```
--- RADIAL [ i j ] [ FUNCTION [ REPLACE ] ]
          MAX      INTEGRAL [ p ]
          BOTH
```

Rotationally averages the data about the pixel "i,j", the MAXimum point, or the centroid. The resulting distribution will be radially symmetric about that point. The radial FUNCTION, its INTEGRAL (percent encircled), or both can also be plotted. If specified, the radius of "p" percent encircled integral is printed.

```
--- TRANSPOSE [ i ]
```

Transposes distribution data array about "i"th line (default is about diagonal) so that future PLOT3D and ISOMETRIC commands give a different view of function. Also, can be used before GRAPH command to see reversed profiles or perpendicular to usual ones.


```

--- MODIFY [ m m' n n' [ a b ] ] [ 'flabel' ]
                fcn
--- [ m m' n n' [ a b ] ]
                fcn
--- :
```

Modifies the data region specified by the two integer pixel ranges, m to m' and n to n'. The data in that region is replaced by a "a" plus "b" times the data value, i.e.

$$f'(i,j) = a + b f(i,j) \quad i=m, m' \quad j=n, n'$$

or

$$f'(i,j) = fcn(f(i,j)) \quad i=m, m' \quad j=n, n'$$

where "fcn" is a function name (either internal or user-defined via \$FCN). More complex modifications can be done with multiple commands. Use the "flabel" option to relabel the functional data.

```

--- REDUCE [ s [ s' ] ]
      -s
      +s
      m" [ n" ]
      m m' m" [ n n' n" ]
```

Reduces the number of data points by selecting out a subset of the current data, i.e. the new data points are given by:

$$f(i,j): \quad i=m \text{ to } m' \text{ by } m'', \quad j=n \text{ to } n' \text{ by } n''$$

The defaults for the increments m" and n" are 1. The default for n is m and n' is m'. A single entry "s" selects out a square region of points (either absolute number or fractional) at the center, lower corner (-s), or upper corner (+s). If no entries are given, the outer minimum values will be "cropped" off.

```

--- SECTION [ m m' [ n n' ] ] [ GET ]
                        PUT
```

Print, GET into registers or PUT from registers the section of data:

$$f(i,j): \quad i=m \text{ to } m', \quad j=n \text{ to } n'$$

If GET or PUT present, then the first data value is associated with register "A0". If $m'-m > 8$ and $n'-n > 24$, then the maximum allowable intrinsic register section is transferred:

$$\begin{array}{lll} A0 <=> f(m,n) & \dots & A9 <=> f(m+9,n) \\ \vdots & & \vdots \\ Z0 <=> f(m,n+25) & \dots & Z9 <=> f(m+9,n+25) \end{array}$$

--- ANGLES [RADIANCE]

Converts a direction cosines space radiance distribution (created by SPREAD / SPOTS DIR) to an angle space RADIANCE or radiant intensity distribution. The polar axis of the spherical angle coordinate system is assumed to be "horizontal" (IES type B photometry).

--- VALUES v h [reg] [v' h' [reg'] ...]

Lists the functional values that correspond to each pair of actual coordinates given. The values can be optionally placed in the given register names.

--- OFFSET [v [h]]

Shift the origin of the distribution to the given actual coordinates. Default is to make the new origin the center of the data set.

--- FOLD [FIRST [n]] SECOND BOTH

Averages the data about the given absolute or fraction pixel number "n" (default is center) of the FIRST (vertical), SECOND (horizontal) or BOTH (default) coordinates. Useful for distributions that should be perfectly symmetric about one or both directions but aren't due to sampling and/or statistical errors.

--- ABEL [c] [INVERSE]

Performs an Abel or INVERSE Abel transform on each line of data assuming that the center is located at "c" (absolute or fractional pixel value, default is centroid of data).

```
--- THRESHOLD a b [ c d ]
```

Reset all data values less than "b" to "a". Optionally, reset all data values greater than "c" to "d". Each of these value entries can be specified either directly, as the MIN, AVE or MAX value, or as a percentage of the range from minimum to maximum (e.g. "33.3%").

```
--- HEADER zlabel z flabel ylabel y y' xlabel x x' 'title'
```

Redefine the file header for the current data. Any entries left off the end of the command or any literal underscores "_" do not alter that part of the header. For example to just change the horizontal label and range:

```
HEADER _ _ _, "New Label" -12 12
```

```
--- REPLICATE [ n ]
```

Replicate distribution |n| times in horizontal (second) direction. If "n" is negative, alternate reversing (mirroring) the distribution for each replicant. The default for "n" is one.

3.11.3.2 Data Plotting -

```
--- TABLE [ m n ] [ l ] [ SKIP [ s ] ] [ PLOT [ c ] ]
```

Produces an "m" by "n" (default 10x10) table of the current data with "l" characters (default 7, maximum 13) for each number. If the SKIP option is used, table entries with the same value as "s" (default is the data's minimum value) are left blank. The table can also be PLOTted where "c" is a character rescaling factor (default 1). This plotted table can be OVERLAYed with other similar plots such as PLOT BEAMS/POL, SPOTS, FIELD DELTA, or CONTOUR. For example:

```
DISPLAY
CONTOUR 10 TICS OVERLAY
TABLE PLOT
```

```
--- RANGE [ d [ d' ] ]
```

Expands or recalculates the minimum "d" and/or maximum "d'" values to be plotted by GRAPH, PLOT3D, ISOMETRIC, CONTOUR, and MESH. Useful for plotting different data at the same fixed scale. This command can only make the plotting range larger than the actual data range. Use the THRESHOLD command to "clamp" the data values to a smaller range.

```
--- PLOT3D [ s ]
```

This option produces a 3D hidden line representation of the data and a set of one-dimensional cross-sections on the same plot. "s" can be used to readjust the vertical scale of the plot.

```
--- ISOMETRIC [ s ] [ n ]
```

This is similar to the PLOT3D command except that it produces a true isometric view of the data and no 1D cross-sections. The floating point entry "s" can be used to readjust the vertical scale. The integer entry can be used to plot every "n"th line instead of the default of every line (n=1).

```
--- DMAP [ n ]
```

Produces a character map of the current distribution data. "n" (default 10) is an optional number of distinct levels to be simulated with different characters.

```
--- GRAPH [ i [ i'... ] ] [APPEND] [AXIS [ x [ x' ]]] [AUTO] [DISCRETE [w]]
      f
      MIN
      MAX
--- [ ' comment... ]
      :
```

This option produces a fancier and faster series of up to 10 1D profiles. The particular profiles are specified either as the line that contains the MAXimum (the default) or MINimum distribution value, the integer "i"th line, or the floating point fraction "f" (between 0. and 1.) of the total number of lines. Also, up to three 72-character comments following this command can be placed at the bottom of the plot (as with any graphical command, a comment on the command itself gets placed at the top of the plot).

The APPEND option tells the program to add these profiles to those from the

last GRAPH command. The AXIS allows the user to override the range of values written on the abscissa axis. If no additional entries are given, the maximum absolute abscissa value is normalized to 1. If there is one entry, the abscissa values are divided by "x". If there are two entries, the new abscissa values are "x" to "x'". By default, the vertical functional limits of the graph are taken from the complete data set. If the AUTO option is used, then the limits are taken from only the specified line data. For a single dataset only, the DISCRETE option draws vertical lines (default w=0) or bars ($0 < w < .5$) from the horizontal (zero) axis to each data point.

```
--- CONTOUR n          [ LOW ] [ TICS t [ t' ] ] [ VECTOR ]
      c c' [ c" ... ]  HIGH   GRID
```

A contour plot will be generated with fractional contours "c c' ..." relative to the full range of the function being plotted (i.e. 0=minimum, 1=maximum). If any of the "c"s are less than zero or greater than one, then they are assumed to be absolute. Alternatively, "n" equally spaced contours can be specified. In any case, the number of contours is limited to 26. Also, a LOW or HIGH resolution "grey scale" or "color map" plot can be produced instead of the line contour plot. If the VECTOR option is specified, the contours in system coordinates will also be written into the 3D vector file. If the coordinate TIC mark or GRID line spacings "t t'" (vertical, horizontal) are specified, the plot is slightly reduced in size and more "window dressing" is added.

```
--- DIRECTIONAL [ UNWRAP ] [ RADIANCE ]
```

Produces a polar plot of the angular energy distribution created by a SPOTS DIRECTION or SPREAD DIRECTION command. By default (no RADIANCE option), it converts the radiance (flux per projected solid angle) as a function of direction cosines generated by these commands to integrated radiant intensity (flux per solid angle) as a function of angle. If any distribution value is negative, the data is assumed to be the common logarithm of energy and is handled accordingly. Optionally, the polar plot can be UNWRAPPED into a Cartesian plot of data versus angle.

```
--- ENCLOSED [ i j ]
      MAX
```

Sums the data within a square array of points centered about the pixel "i,j", the MAXimum point, or the centroid and plots the percent enclosed as a function of the size of the square.

```
--- MESH [ i [ j ] ] [ LOW ]  
HIGH
```

Writes a three-dimensional representation (wireframe or LOW / HIGH resolution shaded "surfaces") of the distribution into the 3D system vector file so that it can be REPLOTted or \$VIEWed with other geometrical entities. If given, only every "i"th across line or "j"th down line is written (defaults are 1).

```
--- PICTURE
```

Directly produces a "grey"-scale picture of the current distribution (or indirectly using the procedure defined with the IMAGER switch or environment variable).

```
--- HISTOGRAM [ n ] [ w ] [ NORMALIZED ]  
PROBABILITY
```

Displays a plot of the number of data values within "n" intervals (equally spaced between the minimum and maximum values). The default for "n" is 21. "w" controls the relative width of the bars plotted (see GRAPH DISCRETE option). The default for "w" is .4. The plot can be optionally NORMALIZED to a peak value of one or by the total number of samples thereby yielding the classical definition of a PROBABILITY density function.

3.11.3.3 Data File Operations -

```
--- COMBINE [ u ] [ c ]  
name fcn
```

Combine the current data with the data in file number "u" or "name.DIS". If "u" is less than 1, original file specified on entry to DISPLAY is used. If "c" is not equal to zero, the current data is added to "c" times the given file. If "c" is absent or zero, the two distributions are multiplied together. If instead of the number "c" a \$FCN function name "fcn" is given, then the two distributions can be combined using an arbitrary formula. At any given pixel, the current data value is placed in the "_1" register and the other data value in "_2". The final result of the function is then placed back in the current data location. For example, one could subtract two distributions (ONE.DIS and

TWO.DIS) in two ways:

```

      DISPLAY ONE          $FCN SUB _1-_2
      COMBINE TWO -1      or  DISPLAY ONE
                           COMBINE TWO SUB

```

```

--- APPEND [ u ]
      name

```

Appends the current data to the end of distribution file number "u" or "name.DIS" (the default is the one loaded on entry to DISPLAY) and updates the header accordingly. Due to the fixed structure of a BRO binary distribution file, the obvious restriction holds: each dataset must have the same number of rows and columns.

```

--- WRITE [ u ]          [ 'title' ]
      name    [ DIS ]
              DIN

```

This option writes the current data to the current file, unit number "u", binary file "name.DIS", or text file "name.DIN" (default is both). An optional comment string can be used to override the "title" record in the resulting file.

```

--- IESFILE [ name [ ELLIP w l h ] ] [ PAD [ p ] ] [ 'string' ]
      RECT

```

Writes out a standard IESNA photometric file ("name.IES") of an angular distribution created by either a RADIANT command (A or C photometry where the polar axis becomes "vertical") or a SPOTS DIR and ANGLES combination (B photometry). The devices RECTangular or ELLIPTical width "w", length "l", height "h", and a comment "string" may be added to the file's header. If the distribution does not cover the whole sphere, then it can be optionally PADded with "p" values (default is minimum).

```

--- TEXTFILE name
--- :
--- header lines
--- :
--- DATA [ k ] [ s ] [ 'format' ]
--- :
--- trailer lines
--- :
--- EOF

```

Writes the current data to a user-definable text file called "name" (default extension is ".txt"). The following registers are immediately loaded and can be used in the header and trailer definitions:

Values			
Register	Literal	Numeric	Description
-----+-----+-----+-----			
T	title	0	Data set title
F	flabel	0	Function label
Z	LABELZ	zval	3rd coordinate value
L1	labelx	0	1st coordinate label
L2	labely	0	2nd coordinate label
N1	NUMX	numx	Number of 1st samples
N2	NUMY	numy	Number of 2nd samples
A1	XMIN	xmin	Minimum 1st coordinate
A2	YMIN	ymin	Minimum 2nd coordinate
B1	XMAX	xmax	Maximum 1st coordinate
B2	YMAX	ymax	Maximum 2nd coordinate
D1	DELX	delx	1st coordinate spacing
D2	DELY	dely	2nd coordinate spacing
F1	FMIN	fmin	Minimum function value
F2	FMAX	fmax	Maximum function value

The actual DATA can be written out in virtually any format. "s" is an optional scale factor that will be multiplied with each data value before writing (default is 1, i.e. no scaling). "format" is an actual Fortran FORMAT specification to be used for each number (default G15.7). "k" is the number of columns to write per line (default is number of columns in data). Some values of "k" have special meaning:

k	output
-1	f
	:
-2	x f
	:
-3	x y f
	:
-4	x y z f
	:
numx+1	x x' ...
	y f ...
	y' f' ...
	:

EOF closes the text file. A RETURN can be used also to return to DISPLAY mode without closing the file. Another TEXTFILE command without a file name entry will then continue writing to the original file. This way multiple distribution data sets can be written to the same file.

The simplest use of TEXTFILE is to dump the data in straight matrix/array or row/column format for importing into spreadsheet software, e.g.

```
TEXTFILE name; DATA; EOF
```

A sophisticated example of TEXTFILE is a macro that takes the current distribution in DISPLAY and converts it to a PostScript image file:

```
PSFILE { 1 Convert current distribution to a PostScript image file
TEXTFILE #1.PS
R=ABS((YMAX-YMIN)/(XMAX-XMIN)) I=(700/R)<560 J=(560*R)<700
'%!PS-Adobe'
'%BoundingBox: 0 580 0 770'
'gsave'
"/picstr" (NUMX) "string def"
20 20 "translate" (I) (J) "scale"
(NUMX) (NUMY) 8
"[ (NUMX) 0 0 -(NUMY) 0 (NUMY) "]"
'{ currentfile picstr readhexstring pop }'
'image'
DATA 255/FMAX 'Z2.2'
'showpage grestore'
EOF
}
PostScript file name>
```

3.11.3.4 Distribution File Structure -

Distribution files are unformatted (binary) direct access files with a 5 record header followed by the data array. The record length is fixed at LENR=NUMF*NUMX or 7 words (4*LENR or 28 bytes), whichever is greater. The structure for the most common 2-dimensional format (NUMF positive) is:

RECORD	LENGTH (bytes)		CONTENT
-----+-----+-----			
1	7	28	247+256*(4*LENR),0,0,0,0,0,0
2	"	"	TITLE*24,NUMF
3	"	"	ZLABEL*8,ZVAL,FLABEL*16
4	"	"	YLABEL*16,YMIN,YMAX,NUMY
5	"	"	XLABEL*16,XMIN,XMAX,NUMX
-----+-----+-----			
6	LENR	4*LENR	F(1,1),F(2,1),... ,F(NUMX,1)
7	"	"	F(1,2),F(2,2),... ,F(NUMX,2)
:			
NUMY+5	"	"	F(1,NUMY),... ,F(NUMX,NUMY)
[NUMY+6			X(1), X(2),... , X(NUMX)]

NUMF specifies the size of each data point in real words, i.e. the function F can be anything from a scalar to complex vectors:

NUMF	FUNCTION
-----+-----	
1	Scalar
2	Complex
3	Vector or Triplet
4	Complex SemiVector
6	Complex Vector
7	Min Ray Data
9	Norm Ray Data
12	2 Complex Vectors

The optional last record can be used to override the normally equally-spaced "x" coordinates with a set of arbitrary ascending values. These coordinates will only be applicable to output from a GRAPH command.

The signs of NUMX and NUMY determine whether the sample points are at the center or edge of a cell, e.g. respectively for X:

```
if NUMX>0
    X = XMIN + (i-.5)*(XMAX-XMIN)/NUMX      i=1, NUMX
or if NUMX<0
    X = XMIN + (i-1)*(XMAX-XMIN)/(|NUMX|-1)  i=1, |NUMX|
```

3.11.3.5 Inputting An Arbitrary Data Distribution -

```
--- DISPLAY 0 [ 'title' ]
    -u
--- xlabel z flabel   ylabel y y' j   xlabel x x' i   [ 'title' ]
--- f f' f" ...
---      :
      j lines of i entries each
--- [ x x' x" ... ]
--- [ WRITE name ]
---      :
    other DISPLAY subcommands
```

The user can create his or her own distribution of data for plotting (and optionally write it to the BRO00u.DAT or name.DIS file) with the above command sequence. If the header "x" and "x'" values are equal, then the program looks for the additional line of "i" coordinates values (ordered but not evenly spaced) after the "j" line of functional data.

Another way to create a DIS file is by writing your own external program to do this. For example in Fortran:

```
* CODE FOR CREATING A BINARY DISTRIBUTION FILE
*
* record length multiplier MREC normally 4
*
* variable declarations
  IMPLICIT REAL (A-H,O-Z)
  IMPLICIT INTEGER (I-N)
  PARAMETER (MREC=4) !1 for VMS default
  CHARACTER*24 NAME,TITLE
  CHARACTER*8 ZLABEL
  CHARACTER*16 FLABEL,XLABEL,YLABEL
  PARAMETER (NUMF=1,NUMX=50,NUMY=50)
  DIMENSION F(NUMF,NUMX,NUMY)
* variable assignments
  XMIN=-2
  XMAX=2
  YMIN=-1.5
  YMAX=2.5
  ZVAL=0
  NAME='example'
  L=INDEX(NAME// ' ', ' ')-1
  TITLE='exponential'
  ZLABEL='z-axis'
  FLABEL='f(x,y)'
  XLABEL='x-axis'
  YLABEL='y-axis'
* function definition
  DO J=1,NUMY
    Y=YMIN+(J-.5)*(YMAX-YMIN)/NUMY
    DO I=1,NUMX
      X=XMIN+(I-.5)*(XMAX-XMIN)/NUMX
      DO N=1,NUMF
        F(N,I,J)=EXP(-X**2-Y**2)*X
```

```
        ENDDO !N
        ENDDO !I
        ENDDO !J
* open, write, and close file
        LENR=MAX(7,NUMF*NUMX)
        OPEN(9,FILE=NAME(1:L)//'.DIS',RECL=MREC*LENR,
& ACCESS='DIRECT',FORM='UNFORMATTED',STATUS='UNKNOWN')
        WRITE(9,REC=1) 247+256*LENR*4,(0.,I=1,6)
        WRITE(9,REC=2) TITLE,NUMF
        WRITE(9,REC=3) ZLABEL,ZVAL,FLABEL
        WRITE(9,REC=4) YLABEL,YMIN,YMAX,NUMY
        WRITE(9,REC=5) XLABEL,XMIN,XMAX,NUMX
        DO J=1,NUMY
            WRITE(9,REC=5+J) ((F(N,I,J),N=1,NUMF),I=1,NUMX)
        ENDDO !J
        CLOSE(9)
        END

*
* CODE FOR READING A BINARY DISTRIBUTION FILE
*
* variable declarations
        IMPLICIT REAL (A-H,O-Z)
        IMPLICIT INTEGER (I-N)
        PARAMETER (MREC=4) !1 for VMS default
        CHARACTER*24 NAME,TITLE
        CHARACTER*8 ZLABEL
        CHARACTER*16 FLABEL,XLABEL,YLABEL
        PARAMETER (MAXF=1,MAXX=362,MAXY=362)
        DIMENSION EULER(3),POSIT(3),F(MAXF,MAXX,MAXY)
        NAME='example'
        L=INDEX(NAME//',',' ')-1
* determine record length of file from OS if possible
        LENR=0
        INQUIRE(FILE=NAME(1:L)//'.DIS',RECL=LENR)
        IF (LENR.LT.1) THEN
* or from header
            OPEN(9,FILE=NAME(1:L)//'.DIS',RECL=MREC*7,
& ACCESS='DIRECT',FORM='UNFORMATTED',STATUS='OLD')
            READ(9,REC=1) LENR,EULER,POSIT
            CLOSE(9)
            IF (MOD(LENR,256).NE.247) STOP 'Not a BRO distribution file'
            LENR=LENR/256
```

```

ENDIF
LENR=LENR/4
* open file and read header
OPEN(9,FILE=NAME(1:L)//'.DIS',RECL=MREC*LENR,
& ACCESS='DIRECT',FORM='UNFORMATTED',STATUS='OLD')
READ(9,REC=2) TITLE,NUMF
NUMV=2
IF (NUMF.LT.0) THEN
* new multi-dimensional format
NUMV=-MOD(NUMF,10)
NUMF=-NUMF/10
ENDIF
* rest of header
READ(9,REC=3) ZLABEL,ZVAL,FLABEL
NUMY=1
IF (NUMV.GT.1) READ(9,REC=2+NUMV) YLABEL,YMIN,YMAX,NUMY
READ(9,REC=3+NUMV) XLABEL,XMIN,XMAX,NUMX
* basic checks
NX=ABS(NUMX)
IF (NX.GT.MAXX) STOP 'Too many X values'
NY=ABS(NUMY)
IF (NY.GT.MAXY) STOP 'Too many Y values'
* other checks
IF (NUMF.GT.MAXF) STOP 'Too many function components'
IF (NUMF*NX.NE.LENR) STOP 'Inconsistent record length'
* read data and close file
DO J=1,NY
READ(9,REC=3+NUMV+J) ((F(N,I,J),N=1,NUMF),I=1,NX)
ENDDO !J
CLOSE(9)
END

```

Here is a procedure for reading a file into an array for use by the data visualization software IDL from Research Systems in Boulder, Colorado.

```

function disfile,name
;
; return data array (up to 4-dimensional) for given distribution file
;
if n_params() eq 0 then $
name=dialog_pickfile(/read,filter='*.DIS',/must_exist)

```

```
    openr,unit,name,/get_lun
;record length in bytes and words
    recl=long(0)
    readu,unit,recl
    recl=recl/256
    lenr=recl/4
;number of function values per pixel
    numf=long(0)
    point_lun,unit,recl+24
    readu,unit,numf
    numv=2
;determine number of variables
    if numf lt 0 then begin
        numv=-numf mod 10
        numf=-numf/10
    endif
    nums=[1,1,1,1]
;array size in each direction
    num=long(0)
    for i=1,numv do begin
        point_lun,unit,(i+2)*recl+24
        readu,unit,num
        nums(numv-i)=num
    endfor
    nums=abs(nums)
;actual data (real or complex)
; if numf eq lenr then numf=lenr/nums(0) ;for old format
    if numf eq 2 $
        then fdata=complexarr(nums(0),nums(1),nums(2),nums(3)) $
        else fdata=fltarr(nums(0),nums(1),nums(2),nums(3))
    point_lun,unit,(numv+3)*recl
    readu,unit,fdata
;finish
    close,unit
    free_lun,unit
    return, fdata
end
```

ASAP COMMANDS

INDEX

a,b,c, 3-12
ABEL, 3-189
ABERRATIONS, 3-66
ACCURACY, 3-128
AFOCAL, 3-62
ALIGN, 3-16
ALLOWED, 3-129
ALTER, 3-29, 3-58, 3-64
ANGLES, 3-189
...aperture, 3-40
APODIZE, 3-142
APODIZE ANGLES, 3-143
APODIZE BOTH, 3-144
APPEND, 3-194
ARC, 3-24
ARRAY, 3-53
ARROWS, 3-118
ASCALE, 3-28
ASYM, 3-51
AVERAGE, 3-187
AXICONIC, 3-46
AXIS, 3-132

BEAMS, 3-138
BEND, 3-53
BEZIER, 3-26
BICONIC, 3-44
BILATERAL, 3-163
Binary file, 3-197
BOUNDS, 3-101, 3-140
BRANCH, 3-109
BSDFDATA, 3-89
BSDFDATA anisotropic, 3-94

CADEXPORT, 3-113
CARTOVAL, 3-47
CHARACTER, 3-24
...CLIP, 3-177

CLIP, 3-139
COARSEN, 3-30
COATING LAYERS, 3-77
COATING MODELS, 3-76
COATING PROPERTIES, 3-75
COLLECTION, 3-175
...COLORS, 3-126
COLORS, 3-118
COMBINE, 3-193
COMPOSITE, 3-27, 3-60
CONDUIT, 3-50
CONIC, 3-25
CONSIDER, 3-114
CONTOUR, 3-192
CORNER, 3-40
CURVES, 3-18
CUTOFF, 3-129

data, 3-79
DECOMPOSE, 3-158
DECOMPOSE DIREC, 3-159
DECOMPOSE POSIT, 3-159
DEFORM, 3-99
DIMENSION, 3-112
DIRECTIONAL, 3-192
DISPLAY, 3-185
DMAP, 3-191
DOMACROS, 3-110
DOME, 3-61
DOUBLET, 3-61
DRAWING, 3-122
DUMP, 3-177

EDGES, 3-18
ELLIPSE, 3-21
ELLIPSOID, 3-36
EMIT BOX, 3-154
EMIT CONE, 3-154

EMIT DATA, 3-156
EMIT DISK, 3-152
EMIT ENTITY, 3-152
EMIT FILAMENT, 3-155
EMIT HELIX, 3-154
EMIT IES, 3-155
EMIT OBJECT, 3-152
EMIT PYRAMID, 3-154
EMIT RAYS, 3-156
EMIT RECT, 3-152
EMIT SPHEROID, 3-154
Emitters, 3-152
EMITTING, 3-157
ENCLOSED, 3-192
END, 3-111
ENT OBJECTS, 3-108
ENTITIES, 3-108
EXPLICIT, 3-58
EXPLODE, 3-109
EXTEND, 3-32
EXTREME, 3-168

FACETS, 3-97
FCN, 3-51
FFAD, 3-171
FFT, 3-186
FIELD, 3-180
FIELDBPM, 3-182
FIELDSUM, 3-181
FITTED, 3-36 to 3-37
FLUX, 3-162
FMAP, 3-55
FOCUS, 3-170
FOLD, 3-189
FORM, 3-186
FRESNEL, 3-105, 3-130
FTSIZE, 3-110
FUNCTIONS, 3-33

GAUSSIAN, 3-158
GENERAL COEFFICIENTS, 3-33
GET, 3-172

GRAPH, 3-191
GRID DATA, 3-150
GRID ELLIPTIC, 3-148
GRID HEX, 3-149
GRID OBJECT, 3-149
GRID POLAR, 3-148
GRID RECT, 3-148
GRID WINDOW, 3-150
GROUP, 3-108

HALT, 3-107, 3-129
HARVEY, 3-80
HARVEY anisotropic, 3-91
HEADER, 3-190
HELIX, 3-27
HISTOGRAM, 3-193
HISTORY, 3-170
HORN, 3-41

IDEAL, 3-63
IESFILE, 3-194
IMAGE, 3-30
IMMERSE, 3-136
INTERF DIFFRACT, 3-102
INTERF REPEAT, 3-103
INTERFACE, 3-102
INVERT, 3-29
IRRADIANCE, 3-178
ISOMETRIC, 3-191

LAMBERTIAN, 3-80
LENSES, 3-59
LEVEL, 3-107, 3-131
LIGHTS, 3-123
LIMITS, 3-100
LINE, 3-19
LIST, 3-167
LOCAL, 3-54
LSQFIT, 3-110

MANGIN, 3-61
MAP, 3-124

MATRIX, 3-16	PIXELS, 3-115
MEDIA, 3-72	PLACE, 3-14
MEDIA ABSORB, 3-73	PLANE, 3-40
MEDIA CRYSTAL, 3-74	PLANE NORMAL, 3-35
MEDIA GRIN, 3-73	PLANE POINTS, 3-35
MEDIA SCATTER, 3-74	...PLOT, 3-78
MEDIA USER, 3-74	PLOT, 3-119
MESH, 3-193	PLOT BEAMS, 3-120
MINIMIZE, 3-69	PLOT FACETS, 3-120
...MINMAX, 3-79	PLOT LIMITS, 3-120
MIRROR, 3-60	PLOT MESHES, 3-121
MISSED, 3-132	PLOT POLARIZATION, 3-120
MODELS, 3-78	PLOT RAYS, 3-120
MODIFY, 3-188	PLOT3D, 3-191
MOVE, 3-164	POINTS 2d, 3-23
MULTIPLE, 3-52	POINTS 3d, 3-18
	POLARIZ, 3-138
NONLIN FIT, 3-84	POLYNOM FIT, 3-82
NONLIN FIT anisotropic, 3-93	POLYNOMIAL, 3-81
NONLINEAR, 3-83	PRINT, 3-111
NONLINEAR anisotropic, 3-92	Prisms, 3-62
NORMALIZE, 3-186	PROFILE, 3-119
NUMBERS, 3-112	PUT, 3-173
OBJECTS, 3-95	QPRINT, 3-111
OBLIQUE, 3-116	
OFFSET, 3-189	RACETRACK, 3-23
OPDMAP, 3-176	RADIAL, 3-187
OPTICAL, 3-43	RADIANT, 3-174
OVAL, 3-22	RANGE, 3-190
...OVERLAY, 3-125	RAWDATA, 3-89
	RAWDATA anisotropic, 3-94
PARABASAL, 3-136	RAY, 3-145
PARAMETERIZE, 3-57	RAYS, 3-146
PARTICLES, 3-87	RAYSET, 3-147
PATCHES, 3-31	RBOUNDS, 3-101, 3-140
PATHS, 3-169	RECTANGLE, 3-22
PENTA, 3-62	REDEFINE, 3-96
PERFECT, 3-63	REDUCE, 3-188
PHYSICAL, 3-86	RENDER, 3-123
PICTURE, 3-193	RENORM, 3-56
...PIXELS, 3-127	REPEAT, 3-19, 3-35, 3-59

REPLICATE, 3-190
RELOT, 3-121
RESET, 3-110
RETURN, 3-111
REVERSE, 3-165
REVOLUTION, 3-45
RIGHT, 3-62
RMS, 3-86
ROTATE, 3-16
ROUGHNESS, 3-97
ROUNDED, 3-23

SAG, 3-28
SAMPLED, 3-48
SAMPLED cylinder, 3-49
SAVE, 3-133
SAWTOOTH, 3-25
SCALE, 3-15
SCALE FROM, 3-15
SCATTER, 3-103
SCATTER MODEL, 3-104
SCATTER REPEAT, 3-103
SEARCH, 3-128
SECTION, 3-188
SEED, 3-133
SEGMENTS, 3-118
SELECT, 3-160
SEQUENCE, 3-59
SHAPE, 3-162
SHIFT, 3-14
SHOW, 3-112
SINGLET, 3-60
SKEW, 3-15
SMOOTH, 3-29
SOLID, 3-56
SOURCE, 3-150
SOURCE WAVEFUNC, 3-151
SPECTRUM, 3-140
SPLINE 2d, 3-26
SPLINE 3d, 3-20
SPLIT, 3-106, 3-131
SPOTS, 3-173

SPREAD, 3-179
SPRINT, 3-111
STATS, 3-168
STATS TRACE, 3-169
STORE, 3-70
SUBSET, 3-161
SUM, 3-95
Summary, 3-2
SUPERCONIC, 3-50
SURFACES, 3-33
SWEEP, 3-30
SYSTEM, 3-113

TABLE, 3-190
TELESCOPE, 3-61
TEST, 3-55
...TEXT, 3-126
TEXTFILE, 3-195
THRESHOLD, 3-190
TITLE, 3-114
TORUS, 3-43
TOWARDS, 3-104
TRACE, 3-166
Transformations, 3-13
TRANSPPOSE, 3-187
TREE, 3-112
TUBE, 3-42

UNITS, 3-71
UPDATE, 3-18
USERAPOD, 3-142
USERAPOD ANGLES, 3-143
USERAPOD BOTH, 3-144
USERBSDF, 3-85
USERBSDF anisotropic, 3-93
USERCURVE, 3-20
USERFUNC, 3-38
USERSAG, 3-44
USERSURF, 3-38
UVSPACE, 3-31

VALUES, 3-189

VANES, 3-91

VARIABLES, 3-68

VCAVITY, 3-88

VIEW, 3-117

VIOLATION, 3-178

VOLUME, 3-87

VOXELS, 3-135

VUFACETS, 3-123

WARNINGS, 3-132

WAVELENGTH, 3-72

WEDGE, 3-62

WIDTHS, 3-137

WINDOW, 3-116

...window, 3-127

WRITE, 3-194

XEQ, 3-17

XMEMORY, 3-145

ZERNIKE, 3-47

CHAPTER 4 SEPARATE UTILITY PROGRAMS

The utility programs are primarily used to process files, either those generated by, or those to be used as inputs to ASAP.

4.1 CONVERSION OF 3D VECTOR FILE TO AUTOCAD DXF FORMAT

The VECT2DXF utility takes a BRO 3D file and converts it to POINT, 3DLINE, 3DFACE, CIRCLE and POLYLINE/VERTEX entities in a standard DXF file format. This file can then be read by software compatible with AutoCAD version 11 or above. Each object (color) in the file is assigned to a different LAYER. The user has the option of specifying a BLOCK name that will contain all the entities. Otherwise, they are written out as separate entities.

4.2 PROGRAMS FOR CONVERTING APART FILES INTO ASAP INPUT

Two programs are provided to turn system data stored by the first APART program on its BASICA (TAPE10) file into ASAP input. The AP2RBS program translates all objects into the smooth continuous SURFACE input to ASAP. AP2RBE, on the other hand, generates faceted objects described by EDGES. However, there is no

reason why the user could not combine the outputs from these two programs to form a single input to ASAP.

Both programs prompt the user for two file specifications. The first is the APART BASICA file. This obviously assumes that APART is also licensed on the same computer system. The second is the desired name for the ASAP input file that will be generated. The programs then ask if you want to keep the original object numbers as assigned in APART or simply renumber the objects sequentially.

4.3 CONVERSION OF AUTOCAD DXF FILES TO ASAP INPUT

The DXF2ASAP utility will read an AutoCAD file (with DXF extension), convert all surface definitions (and BLOCKS) into ASAP CURVE based objects and write them to INR and LIB files of the same name.

4.4 SUPPORTED ENTITIES AND RESTRICTIONS ON IGES IMPORT

===== Currently Supported Translations =====

IGES Entity	----->	ASAP Commands
Global Section Model Space Scale		GROUP; SCALE
Global Section Units Name		GROUP; SCALE FROM
Directory Section Color Number		OBJECT; REDEFINE COLOR
100 Circular Arc		CURVE; ARC
102 Composite Curve		CURVES; COMPOSITE
104 Conic Arc		CURVE; CONIC
106 Copious Data		CURVE; POINTS
106.63 Simple Closed Planar Curve		CURVE; POINTS; OBJECT
108 Plane (bounded)		CURVE; OBJECT
110 Line		CURVE; LINE
112 Parametric Spline Curve		CURVE; POINTS
114 Parametric Spline Surface		CURVES; PATCHES; OBJECT
116 Point		CURVE; POINT
118 Ruled Surface		CURVE; CURVE; OBJECT

120 Surface of Revolution	CURVE; SWEEP AXIS; OBJECT
122 Tabulated Cylinder	CURVE; SWEEP DIR TO; OBJECT
123 Direction	... a,b,c ...
124 Transformation Matrix	SHIFT/MATRIX
126 Rational B-Spline Curve	CURVE; POINTS
128 Rational B-Spline Surface	CURVES; PATCHES; OBJECT
132 Connect Point	CURVE; POINT
140 Offset Surface	OBJECT; SHIFT
141 Boundary	CURVES
142 Curve on Parametric Surface	CURVE; UVSPACE
143 Bounded Surface	CURVES; OBJECT; (R)BOUNDS
144 Trimmed Parametric Surface	CURVES; OBJECT; (R)BOUNDS
190 Plane Surface	SURFACE; PLANE; OBJECT
192 Right Circular Cylinder	SURFACE; TUBE; OBJECT
194 Right Circular Cone	SURFACE; TUBE; OBJECT
196 Spherical Surface	SURFACE; ELLIPSOID; OBJECT
198 Toroidal Surface	SURFACE; TORUS; OBJECT
308 Subfigure Definition	
402 Group Associativity	OBJECTS
406.3 Level Function	OBJECT =layer...
406.15 Name Property	OBJECT =...name
408 Subfigure Singular Instance	OBJECTS
510 Face	CURVES; OBJECT
514 Shell	OBJECTS

===== Other Restrictions =====

1. Accepts only fixed ascii IGES files with default delimiters.
2. By default, translates only visible independent surfaces and related child entities.
3. Model space versions of trimming curves are normally required.
4. Offset surface approximated by simple shift.
5. Trimmed closed or semi-closed surfaces may require manual correction of SWEEP POINT for model space (R)BOUNDS curve.
6. Signs of (R)BOUNDS entities for 143 Bounded Surface set somewhat arbitrarily. 144 Trimmed Surfaces are preferred.
7. Currently, no attempt is made to match actual colors.
8. Shell Faces require the /all option and do not yet include their bounding Loops.
9. Nesting of Subfigures is currently not allowed.

4.5 CONVERTING BMP BITMAP IMAGE FILES TO/FROM DISTRIBUTION DATA

BMP2DIS converts one color in a BMP bitmap image to a DIStribution data file and sets the full "width" (in system length units) of the distribution:

```
BMP2DIS file [ Red ] [ width ]  
             Green  
             Blue  
             Mono
```

The "Mono" version converts the 3 colors into a grey-scale using the standard formula $.30*Red + .59*Green + .11*Blue$.

DIS2BMP converts one or three DIStribution data files to a monochrome or color BMP bitmap image file, respectively:

```
DIS2BMP disfile [ bmpfile ]  
DIS2BMP redfile greenfile bluefile [ bmpfile ]
```

APPENDIX A DEVICE INDEPENDENT GRAPHICS PROTOCOLS

A.1 TWO-DIMENSIONAL PLOT FILES

A.1.1 Basic Organization Of Graphics Routines

Graphics are performed either directly to a graphics device, an intermediate file (for later processing) or both. The work is divided between a set of high-level device-independent routines written in Fortran and two driver routines written specifically for a particular graphics device (or set of devices).

```

      |
      V
INITIA  High level routines
LINE
SYMBOL
NUMBER
WHERE
PLTEXT
ENDPLT
      |
      V
PLCLIP  Clip vectors at "window" edge
--> PLOT  Central routine
      |
      V
OPTPLT  Optional short vector optimization
OPTVEC  Unnecessary vector removal
WRPLOT  Relative to absolute conversion and/or write to file
      |
      V
--> GDRIVR  Immediate graphics driver
<-- GINPUT  Interactive graphics cursor input

```

A.1.2 The Central Plot Routine Protocol

The two-dimensional plot data follows very closely the CALCOMP protocol. Each line in the file corresponds to a call to the PLOT routine and contains essentially the calling arguments, i.e.

```

CALL PLOT(X,Y,IPEN)      -99.999 < X < 99.999
                        -9.999 < Y < 99.999  (maximum)
                        0. < Y < 11.        (typical)

```

Except for an IPEN value of 0, X and Y are the floating point coordinates in inches in the long and short directions of the device, respectively. If the origin is not changed, then X can take on values from -99.999 to 99.999 and Y typically from 0.000 to 11.000 inches. IPEN is the integer "pen" control

parameter and can take the following values:

IPEN	ACTION
-----+-----	
0	Set character height to X inches and rotation to Y degrees.
1	Move to X,Y with pen up, put pen down (simple point plot).
2	Move to X,Y with pen down (draw line).
3	Move to X,Y with pen up (no drawing, simple move).
4	Fill a rectangle from last point to diagonal corner at X,Y.
5-31	Change pen attribute* to IPEN-4 then move to X,Y with pen up.
32-127	Place corresponding ASCII character at X,Y (lower left corner).
128-998	Reserved for other attributes and/or character fonts.
<0	Perform above functions then place new origin at X,Y.
999	End of plot and get ready for new plot indicator (exit graphics mode then set flag to enter graphics mode and clear device on next write to plot). X and Y not used.

*The pen attribute is assumed to be a color number from 1 to 27 with 1 being completely opposite of the devices background color (the default) and 27 being the same as the background. For example, a useful default definition for the colors in terms of relative primary content would be:

Blue = $\text{mod}(k, 3)$	$k=27-n$ for Black background
Red = $\text{mod}(k/3, 3)$	$k=n-1$ for White background
Green= $\text{mod}(k/9, 3)$	$n=1, 27$

A.1.3 Directly Accessing The Immediate Graphics Driver

It is possible to directly access the graphics drivers for faster, more interactive displays. The main graphics driver routine is used primarily for writing to the display.

CALL GDRIVR(X,Y,IP)

The coordinates are again assumed to be approximately equivalent to inches on a standard page, i.e.

0. < X < 10. (Horizontal, across)

0. < Y < 7.5 (Vertical, up)

However, both coordinates are automatically scaled by the driver to fill the device's display. The third argument is a mode value that specifies the action to be taken and is similar to but not exactly equivalent to the IPEN value of the previous section.

IP	ACTION
-----+-----	
<0	Enter graphics mode, initialize device, and clear display. If IP comes back positive, the device is not the console or user's terminal and IP is the logical unit number associated with it.
0	Set character size to X (default=.125) and rotation to Y degrees.
1	Move to X,Y and write point.
2	Move to X,Y with writing (draw line).
3	Move to X,Y with no writing (simple move).
4	Fill a rectangle from last point to diagonal corner at X,Y.
5-31	Change writing attribute* to IP-4 then move to X,Y.
32-126	Write corresponding ASCII character at X,Y (lower left corner).
127	Prepares device to write a text string starting at X,Y and returns required string terminator ASCII code in IP. The application program can then write a string directly to device.
128-899	Reserved for other attributes and/or character fonts.
900	Immediately exit graphics mode.
9XX	Wait XX seconds (01-98), then exit graphics mode.
999	Beep, wait for <CR>, then exit graphics mode.

*The writing attribute is assumed to be a color number from 1 to 27 with 1 being completely opposite of the devices background color (the default) and 27 being the SAME as the background (for erasing).

The second graphics driver routine is used to get information back from the display in the form of the coordinates of a user-positioned cursor, i.e.

CALL GINPUT(X,Y,IP)

Input	
IP	MEANING
-----+-----	
<-31	Define home position to be X,Y and activation key to be -IP.
<0	Enter interactive graphics cursor mode -IP at position X,Y.
=0	Enter interactive graphics cursor mode at position X,Y.
>0	Enter interactive graphics cursor mode IP at current position.

The currently allowed cursor modes are:

- 1 Small crosshair cursor at interactive point
- 2 Straight line dragged from start to interactive point
- 3 Full window crosshair at interactive point
- 4 Rectangle (box) dragged from start to interactive point
- 5 Inscribed ellipse dragged from start to interactive point

The values returned by GINPUT are given by the following table.

Return	
IP	MEANING
-----+-----	
<0	Device does not have interactive graphics cursor capability.
0	<CR> key was pressed at X,Y.
32-127	ASCII code of key pressed at X,Y.

A.2 THREE-DIMENSIONAL GRAPHICS FILES

The protocol for the three-dimensional data files is a simple one devised by BRO. Each line in the file corresponds to a coordinate in 3-dimensional space.

```
CODE,X,Y,Z [,A,B,C] [ /identifier ]
```

X,Y, and Z are the floating point coordinates in system units. A,B, and C specify an optional associated vector (usually a surface normal). They will have to be converted to pixels, inches, etc. for your particular device. The integer part CODE parameter is defined as follows:

Integer Part of CODE	ACTION
-----+-----	
0	Move to X,Y,Z (no drawing).
>0	Draw a dot at X,Y,Z.
<0	Draw a line to X,Y,Z.

The actual CODE value assigned to each point is application dependent but is usually assumed in most BRO software to be an integer color number with a fractional object number.

APPENDIX B

REVISION HISTORY AND RELEASE NOTES

B.1 VERSION 7.5

Release Notes for ASAP 7.5 Kernel as of January 3, 2003.

Changed commands (affects backward compatibility)

EMITTING OBJECT default is now total ray number instead of rays per facet.

Renamed Commands (but with backward compatibility)

CAD extended to more descriptive CADEXPORT (abbreviation will still work).

FIELD split into FIELDSUM and FIELDBPM (abbreviation still works for both).

New Commands

DISPLAY

HISTOGRAM plot (number of values in each of specified number of intervals).

SURFACE

SUPERCONIC rotationally symmetric asphere used in optical design.

MEDIA scatter USERprog routine can even change wavelength (e.g. fluorescence).

Enhanced Geometry Commands

CADEXPORT

Improved ACCURACY of IGES rotationally symmetric SURFACES.

Can call VECT2DXF utility to create DXF version of current VCR file.

SURFACE

ARRAY

Now has simpler alternative syntax for orthogonal case.

New SEARCH option can dramatically speedup tracing of "tiled" surfaces.

CORNER cube can now have a HEXAGONAL aperture.

CURVE

USERCURVE can produce smooth rational or non-rational cubic SPLINE curve.

SPLINE with CURVatures will use rational quadratic segments when possible.

LENS

SINGLET/MANGIN aberration elimination for any bending and conjugate factors.

ABERRATIONS solve to maintain back focal distance (last conicoid to focus).

VUFACETS can also LIST the number of facets created for each object.

PLOT FACETS can limit, if possible, the total number of facets on any object.

Enhanced Properties Commands

OBJECT

ROUGHNESS has additional probability distributions to select from.

SCATTER cone can now be back TOWARDS incidence direction (retro-reflection).

FUNCTION

USERFUNC passed current WAVELENGTH (or associated ray/beam's wavelength).

MODELS

RAWDATA default collapse threshold changed from .001 to .000001.

Enhanced Analysis Commands

DISPLAY

Also prints out RMS variance and widths of data.

GRAPH DISCRETE option displays data as vertical bars of controllable width.

MESH now defaults to monochrome but also accepts standard COLOR option.

Automatically calls BMP2DIS utility if given file has a .BMP extension.

SELECT by previous OBJECT now accepts wildcards in specified name.

AXIS LOCAL can refer to a specific object not just a ray/beam's current one.

SPOTS/OPDMAP/SPREAD/FIELDSUM/RADIANT can be in local coordinates of an entity.

SAMPLED/MESH/DECOMPOSE/EMIT DATA can process local coordinate distributions.

FIELD BPM Finite Difference Beam Propagation Method (FD-BPM).

BC option

Expanded to asymmetric computational window boundary conditions.

Easy option for (and defaulted to) physically absorbing region at edges.

Size of absorbing regions can be specified as fraction of total samples.

Ordering INTERFACE media allows "mismatch" errors to be fixed automatically.

Semi-vectorial version of wide-angle algorithm added for polarized fields.

GROUP/CONSIDER special case of ".*" wildcard added.

CLIP POS followed by a BOUNDS\RBOUNDS command allows complex ray bounding.

DECOMPOSE accepts any complex distribution file not just BRO029.DAT.

STATS ALL option also lists objects with no rays and zero flux on them.

FIELD COUPLE more easily to fundamental mode of step or gradient index fiber.

Macro Language

&GUI parses any expression results into their string equivalents.

Added ability to access a macro in a library other than current \$IO LIB one.

Higher-order Bessel functions added (BJn and BKn for n up to 9).

RAN function generates random numbers with various probability distributions.

\$RAN option for default DEC generator or new BEST one.

\$TIC option allows user to set the time units for this and all future output.

\$ITER/\$EVAL new form that generates random perturbations of given variables.

&GRAB strings instead of numbers from output buffer.

More intelligent and flexible ":" operator.

\$GRAB can now also find numbers that immediately follow an equal sign (=).

Other Changes

As usual, *.SYS files are incompatible with previous versions.

Spatial distribution files can now have an arbitrary global orientation.

IGES Import

When possible will use wildcard on GROUPS instead of number.

Can now handle (very rare) case of nested COMPOSITE curves.

Ray selection flags increased (SELECT still works on any number of rays!)

Pro from 32 to 100 million, Basic from 9.6 to 10 million.

Faster tracing of semi-quadratic patches (e.g. IGES imported surfaces).

Experimental code for "intelligently" setting the default FACETing on objects.

B.2 VERSION 7.1

Release Notes for ASAP 7.1 Kernel as of February 28, 2002

New Commands

MODELS

RAWDATA version of BSDFDATA that does not interpolate in logarithmic space.

OBJECT

RBOUNDS version of BOUNDS allows virtually unlimited entity ranges.

Enhanced Commands

MOVE

BY/TO a specified optical path length.

TO a best-fit SPHERical wavefront.

DECOMPOSE

POSITION CONVERGE/DIVERGE accuracy improved by using non-planar beamlets.

No longer creates "left-handed" beams that have anomalous $\pi/2$ phase shift.

MODELS

VOLUME uses (and MIE displays) absorption efficiency if non-zero.

SURFACES

SOLID faster and now set automatically by TUBE, semi-ELLIPSOID, and CONDUIT.
...HEXAGONAL aperture option in addition to RECTANGLE and ELLIPSE.

SAMPLED distribution files are now fully buffered for a tremendous speedup.

OPTICAL allows new parabolic term to be subtracted from base definition.

DISPLAY

REDUCE now has a shorter form for picking out every nth pixel.

GRAPH DISCRETE option draws vertical lines from axis to each data point.

SELECT on number of times ray has HIT objects and/or AXIS angular range.

EMIT DATA

Now allows control of which hemisphere from a 2D distribution.

Can read directly new compressed DUMP format.

EDGE

POINTS now more thoroughly checks consistency of input data.

SPLINE 3D version optionally allows specification of CURVatures at points.

LINE/ELLIP/RECT/OVAL/ROUND redefining default subdivision can be suppressed.

CONIC now allows a parabola to be subtracted from base curve.

FIELD

More accurate (and sometimes faster) version of the FD-BPM algorithm.

Validity extended by using non-paraxial invariants for Gaussian beamlets.

Restored applicability and improved speed of the "fast" method.

MEDIA SCATTER rays are now uniquely tagged for breakdown by PATHS.

SCATTER

TOWARDS temporary limitation to 32767 scattered rays per incident removed.

OBJECT now issues warning for duplicate names.

HISTORY can process only every nth ray instead of all rays.

CAD IGES

Now exports ARRAYS of surfaces in the form of 308/408 Subfigures.

Displays residual angular errors for SURFACE entity spline approximations.

LENS

ABERRATIONS

STORE can now write ZEMAX and OSLO lens files (in addition to CodeV).

VUFACETS defaults tessellation of unBOUNDED planar EDGE objects to 3D viewer.

Macro Language

Macro argument "#" substitution no longer occurs beyond "!!" comment start.

Macro, function, and variable name lengths increased from 16 to 32 characters.

\$MOVE same as \$COPY except the source file is deleted afterwards.

\$EVAL can now take a \$FCN that returns a complex (real,imaginary) result.

&SHOW will also list the internal stack code for any \$FCN definitions.

Real/imaginary entries can be assigned to two related registers/variables.

Other Changes

*.SYS (as usual) and *.REG files are incompatible with previous versions.

IGES import

Added option to compress (COMPOSITE) multiple trimming curves into one.

ALL geometry option now includes unbounded Faces (510) of solid Shell (514)

Unnested Subfigures (308/408) are now processed.

Total geometry storage doubled in both Basic and Pro versions.

Most coherent SPREAD/FIELD Gaussian beamlet calculations 3 to 5 times faster.

Maximum allowed FFT size set to 2^{16} (65536) in both Basic and Pro versions.

B.3 VERSION 7.0

Release Notes for ASAP 7.0 Kernel as of January 19, 2001

Important Changes (no backward compatibility)

LENSES no longer treat zero curvature as special case of 4th-order aspheric.
PARTICLES with \$FCN size distribution has new entry for integration samples.
FTSIZE defaults to 512 samples at startup (and after a RESET or END).
EMITTING DATA uses DUMP file wavelength only if current WAVELENGTH is zero.

New Commands

COLLECTION efficiency as a function of aperture size and acceptance angle.
LENS

ABERRATIONS of a centered set of conicoids.

VARIABLES for locally or globally optimizing a basic imaging system.

MINIMIZE weighted average or maximum RMS spot size subject to constraints.

STORE current design in an ASAP file and/or a CodeV sequence file.

DOMES forms a single refractive element with at least one side a hemisphere.

DISPLAY

APPEND current data to the end of an existing distribution file.

EDGE

EXTEND linearly one or both ends of a curve.

VOXELS for tracking and visualizing volume fluence, absorption or irradiance.

MODEL

VOLUME scatter version of PARTICLE for use with new MEDIA SCATTER option.

SURFACE

CONDUIT sweeps a circle along a explicit cubic planar curve.

Enhanced Commands

SURFACE

REVOLUTION

Can optionally put input data points into current 3D VECTOR file.

Smart FIT option automatically finds optimum degree surface.

USERFUNC/USERSAG/SAMPLED now handle ROTATED versions better.

Different SAMPLED surfaces can now access same distribution file.

Cartesian (not polar) parameterization on holeless RECTangular apertures.

ELLIPSOID can optionally be only a half instead of a full one.

OBJECT

DEFORM short form

Now exact when ACCURACY HIGH and the base is a SURFACE entity.

Can use a general USERSAG definition as the deformation.

FFAD now exactly autoscales RMS spot-size ellipses.

MEDIA

Can now be directly created from any of the glass catalogs.

Optional SCATTER for direct inhomogeneous Monte-Carlo volume scattering.

LENS

Basic surface extended to a conicoid minus a paraboloid (e.g. an aspheric).

SEQUENCE can optionally take DIAMeters instead of HEIGHTS (semi-diameters).

MANGIN can now be an APLANAT (to third-order) just like SINGLET.

RADIANT

Automatically averages to compensate for smaller solid angles near poles.

Optional LOW or HIGH resolution 3D shaded patterns in place of wireframes.

DISPLAY can pick out a specific 2D slice of a 3D distribution data file.

MESH LOW or HIGH resolution 3D shaded display in place of simple wireframes.

GRAPH uses new (hopefully improved) algorithm for axes numbers/tic-marks.

ACCURACY LOW/MED for some Bezier intersections improved without speed loss.

XMEMORY option to allocate ray storage entirely in RAM for optimum speed.

SCATTER

TOWARDS can now treat more precisely an odd-shaped importance-area EDGE.

ALTER now has an option for every "n"th EDGE point or LENS conicoid.

CURVE

SPLINE 3D smoother and can operate with some or all tangent vectors missing.

COMPOSITE has new option for merging GAP points.

...CLIP option can now reference a (possibly ARRAYed) SURFACE entity.

HALT settings can now be applied on an object-by-object basis.

GAUSSIAN astigmatic mode can now be created from just one ray/beam.

MODEL

...PLOT puts hemispherical output for all default angles into one DIS file.

FIELD can take an existing file and propagate it through a micro-structure.

VIOLATION now also controls stability errors for above FD-BPM algorithm.

TRACE STATS prints out range of indices in any inhomogeneous (GRIN) media.

...PIXELS option can now override aspect ratio (vertical/horizontal widths).

SHAPE of a beam can be the fundamental mode of a symmetric step-index SLAB.

Macro Language

\$CLEAR clears the output buffer for future \$GRAB and/or \$PLAY commands.

\$COPY copies or appends an arbitrary existing file to one of a different name.

\$SCR can write up to 344 characters per line to the current output device.

\$VIEW can optionally send a file to a new viewer or add it to the current one.

\$DISP immediately or automatically displays binary distribution files.

\$PATH now closes default files in old directory and then reopens them in new.

Other Changes

ASAP-Basic can now handle more complex systems (e.g. imported from CAD):

	Basic 6.6	Basic 7.0	Pro 7.0
Maximum number of objects	300 ==>	1,000 (	9,999)
Maximum number of entities	1,000 ==>	3,000 (	30,000)
Total geometry storage	100,000 ==>	300,000 (	3,000,000)
Maximum Fourier transform size	256 ==>	512 (	2048)
Total pixel storage	362x362 ==>	736x736 (	2896x2896)
Maximum possible voxel storage		81x81x81	(203x203x203)

As usual, *.SYS files are incompatible with previous versions.

Maximum input/output line and file/object name length upped from 128 to 344.

Several speed improvements making this the fastest version of ASAP yet.

B.4 VERSION 6.6

ASAP 6.6 Kernel Enhancements as of August 24, 1999

New Commands

MODELS

PHYSICAL reflective scatter for random rough surfaces even at grazing.
VIEW sets perspective-based instead of WINDOW-based graphics displays.

SURFACE

CORNER of a cube aligned to an axis (e.g. retroreflector).

LENS

PERFECT eikonal-based (realistic) lens for object at infinity.
FTSIZE sets the Fourier Transform size to be smaller than the default maximum.

CURVE

SPLINE of G2 continuous cubic segments given points and slopes/directions.

Enhanced Commands

TRACE can PLOT only those rays whose starting depth coordinate is near zero.
FRESNEL can allow TIR rays without splitting and angular flux variation.
PLOT POLARIZATION takes an optional flux weighting factor.
CAD can now export most SURFACE-based objects (BOUNDS still not implemented).
PLOT FACETS HIDE default is now 2 (typical camera) instead of 1000 (parallel).
RENDER

Can now take into account any reflective scatter MODELS.

RAY segments can be rendered as smooth shaded cylinders (anti-aliasing).

SURFACE

ARRAY instances can be handled identically to other BOUNDS entities.
TRACE ACCUM keeps track of ray STATS but does not print them afterwards.
STATS TRACE allows selective display of previous trace statistics.

MODELS

BSDFDATA has a choice of coordinates and interpolation methods.

Anisotropic-surface versions of HARVEY, NONLINEAR, USERBSDF, BSDFDATA, and SUM.
RAY PLOT will now also draw any paraboloidal ray trajectories.

CURVE

BEZIER/POINTS/PATCH maximum degree has been increased from 10 to 20.

PATCH smaller of two degrees can now be cubic or higher.

DISPLAY

FFT can now use a different Fourier Transform SIZE than the default.
GRID POLAR now has an alternative shorter and simpler form.

Macro Language

\$READ commands can now be nested within macros and other \$READ files.
\$EVAL is similar to \$ITER but rapidly evaluates \$FCN functions.
\$ITER maximum variables increased from 25 to 40, targets from 100 to 125.
\$FCN maximum number of definitions increased from 20 to 60.

Other Changes

As usual, *.SYS files are incompatible with previous versions.
Intrinsic "patching" of SURFACES set more intelligently (usually higher).
Most USERPROG routines are now tested before actual use.
ASAP-Pro capabilities increased to keep pace with typical 128MB workstations:

Maximum number of objects	3,000 ==>	10,000(-1)
Maximum number of entities	20,000 ==>	30,000
Total geometry storage	2,000,000 ==>	3,000,000 FP numbers

Eliminated many "fatal" program terminations when exceeding certain limits.
IGES import:

- More accurate degree-reduced surfaces that raytrace just as fast.
- New option to increase ORDER (degree) of ASAP Bezier curve segments.
- Exact versions of Cubics (112), Bicubics (114), Beziers (126/128 subset).

Improved PARABASAL and/or GRIN ray intersection with SWEEP AXIS 360 object.

B.5 VERSION 6.5

ASAP 6.5 Kernel Enhancements as of February 19, 1999

New Commands

DISPLAY

HEADER information can be selectively redefined.

REPLICATE distribution one or more times with or without "mirroring".

MOVE PARABASALS to base ray plane of each beam.

MODEL

RMS, autocorrelation length, and fall-off of surface variations.

COATING MODELS specify angular properties for reflection and transmission.

VIOLATION controls SPREAD and FIELD paraxial and positivity messages.

Enhanced Commands

SAVE

File now always uses short records when XMEMORY set to MIN.

Maximum record number increased from 2^{24} (16 million) to 2^{31} (2 billion).

OBJECT

ROUGHNESS

MODEL resolution changed from 1 to .2 milliradians in Pro version.

Optional correlation length of micro-roughness for high angle shadowing.

SCATTER

TOWARDS

Center direction can now be parallel to surface.

Default beam half-angle is now 90 degrees when PARABASAL number is 0.

Simple solid angle towards an entity's reference POINT.

INTERFACE DIFFRACT has short form for efficiencies given by a COATING MODEL.

DISPLAY

GRAPH axes numbering changed from multiples of 2 decades to 3.

COMBINE

Record buffer increased to 10,000 pixels per line.

Added ability to do an arbitrary combination with a \$FCN definition.

RADIAL averaging of center pixel and first ring improved slightly.

AVERAGE can be WEIGHTed by estimated error in each pixel.

NORMALIZE now allows different scale factors for the two coordinate ranges.

DIRECTION/ANGLES can output RADIANCE instead of default radiant intensity.

MODEL

BSDFDATA

Now works with just a single normal-incidence scan.

Processed points now written TO and read FROM SYSTEM file.

Small data sets (<95 points) always kept in main memory storage.

Any measured data values can be BSDF times COSine of angle from normal.

...PLOT will take an additional OVERLAY option.

HARVEY

Extended with 2 additional parameters postulated for rough surfaces.

Wavelength scaling (smooth surface approximation) done automatically.

PARTICLE

New short form based on simple Henyey-Greenstein model.

Particle size distribution can be user-specified via a \$FCN definition.

EMIT

OBJECT can also take total number of evenly distributed rays.

FILAMENT parametric \$FCN weight is now optional (default 1).

FIELD/SPREAD

Now calculates axial pathlength phase much more accurately.

Positivity violations no longer abort program (just current calculation).

LENS

IDEAL Jones matrix behavior is now virtually independent of incidence angle.

WIDTH...EDGE now produces much sharper edges on coherent GRID RECT or POLAR.

OPDMAP defaults have changed to make them better for most applications.

EXPLODE

Can now create glass inner edge, Mangin outer edge, and mirror back.

Added option to do ALL lens entities, not just one at a time.

Automatically creates CURVE-based objects if standard CADEQUIV.LIB loaded.

PLOT BEAMS no longer uses a crude approximation for skew beams.

SURFACE

USERFUNC

Gradient done by finite-differences if \$FCN returns only function value.

Default routine now handles isotropic SCALE transformation.

SAMPLED uses smooth positional interpolation with slopes and ACCURACY HIGH.

AXICONIC can specify other end of curve with LENGTH or HEIGHT option.

USERSAG can optionally shift the given profile along a conic curve.

GRID POLAR does better job of creating equal area beams for non-unity weight.

TRACE also lists warning summary for suppressed child rays.

WARNINGS

Limit now only terminates current TRACE not entire program.

Default is now unlimited warnings instead of number of ray selection flags.

MEDIA;...CRYSTAL

Reflected extraordinary beam now created along with ordinary.

Interface reflectivity/transmissivity values handled more accurately.

PATHS default output format is now MEDIUM instead of LONG.

SPLIT...MONTECARLO for probabilistic "one ray in, one ray out" tracing.

DUMP now works with "+" entry after an END command reinitialization.

Macro Language

\$RAN sets "~" random operator seed.

\$FCN uses separate instances of any locally initialized intrinsic registers.

Math functions added:

FACT returns factorial of argument (actually continuous Gamma function).

STEP and RECT binary functions.

ERF error function and its complement.

Other Changes

As usual, *.SYS files are incompatible with previous versions.

Eliminated most null length "for###.dat" files and renamed rest "bro###.dat".

Allow reading of input file lines with "extra" CRs and/or LFs.

Increases in Pro version:

Simultaneous grating orders per object 9 ==> 99

Important scattering areas per object 9 ==> 20

USERSURF subroutine in USERPROG library changed to a logical function.

Virtual.pgs file no longer limited to 2GB (33 million XMEMORY MIN rays).

New IGES import options:

Use uniform non-rational splines instead of non-uniform rational.

Display debug information (first Parameter record of used entities).

IGES trimmed planar spline surface imported as more efficient ASAP object.

UVSPACE BOUNDing has been sped up considerably.

For consistency, TIR rays get normal incidence reflectivity if FRESNEL OFF.

B.6 VERSION 6.0

Release Notes for ASAP 6.0 Kernel as of May 28, 1998

Important changes (no backward compatibility)

LENS conicoid with 0 curvature treats "conic" entry as 4th-order deformation.
DEFORM no longer accesses a user-programmable Fortran routine.
POLYNOMIAL scatter model redefined to guarantee reciprocity.

New commands

BRANCH changes default (current) position in object name hierarchy.
TREE displays indented object name (and optionally entity) hierarchy.
CURVE/EDGE

PATCHES makes current data actually define a rational Bezier surface mesh.
UVSPACE tags current curve to be defined in parametric-space of an object.
COARSEN converts edge to a coarser (less dense) piecewise linear curve.
BEZIER converts explicit polynomial to the equivalent-order Bezier segment.
USERCURVE piecewise linear parametric curve or surface from \$FCN definition.

LENS

TELESCOPE one/two mirrors with/without corrector and/or field flattener.

DISPLAY

ABEL transform (or INVERSE) of current distribution.

THRESHOLD can be used to reset the floor and/or ceiling of data values.

PLOT CURVES same as EDGES except control points are also displayed.

GRID

WINDOW creates a rectangular grid of rays that fills the current window.

DATA creates rectangular grid of rays weighted by distribution in file.

VUFACETS shortcut for facetting and viewing current objects.

SURFACE/FUNCTION

USERSAG creates a surface with a user-defined radial or toric profile.

EXPLICIT converts current function to explicit form (e.g. for DEFORM).

DUMP currently selected rays to a binary file (DIS format).

EMITTING DATA creates rays from data in given binary distribution file.

Linear Transformations

SCALE FROM given length units to the current system UNITS.

MODEL

NONLINEAR in free parameters (generalizes combination of Phong and Harvey).

Enhanced commands

LENS

SINGLET...APLANAT option bends and aspherizes to remove spherical and coma.

IDEAL

RANdomize output azimuthally (e.g. incoherent fiber optic simulation).

Specify \$FCN function for radial position and direction apodization.

CAD IGES

Now exports COLOR index information.

Upgraded to ANSI 5.3 standard (e.g. year 2000 dates).

System UNITS don't have to be specified in advance anymore.

TRACE STATS also displays any flux absorbed by MEDIA.

PATH

Output can now be controlled with SHORT/MEDIUM/LONG option.

Suppress listing paths containing less than a specified number of rays.

CURVE/EDGE

SMOOTH can now do cubic interpolation in addition to quadratic.

COMPOSITE can now also contain closed edges like RACETRACK.

INTERFACE

DIFFRACT function can now be ARRAYed (holographic lens arrays).

REPEAT can also reset/remove current properties.

OBJECT

...NORMAL/SURFACE alternate function can now be ARRAYed.

FACETS can specify either minimum, fixed, or maximum number.

ROUGHNESS allows for only a FRACTION of the incident rays to be RANDOMized.

SPOTS

Can now also mark each ray with its NUMBER.

Only EVERY nth ray can be plotted without affecting distribution file.

CONSIDER now has ADD/REMOVE forms for updating current state.

NUMBERS can now show current storage usage with new SUMMARY option.

FLUX can now alter ray fluxes to give desired TOTAL flux.

SCATTER

REPEAT copies a previous object's or removes current scatter properties.

MODELS for specifying separate reflective and transmissive BSDFs.

TOWARDS SPECular can be just for REFlected or TRANsmitted rays.

SELECT

Split or scatter GENERATION number.

If ray was split at a certain object.

Those paths not listed in last PATH command.

USERAPOD ANGLES has a new short form with no angle clipping entries.

VUFACETS and PLOT FACETS can minimize facetting on some objects when possible.

APODIZE POS/DIR can set apodization axis (default is average axis of rays).

SURFACE/FUNCTION

REVOLUTION can now do a better fit on up to 1000 coordinate pairs.

ZERNIKE default aperture is unit circle but can be reset to RECT or ELLIP.

FITTED FIXTERM option no longer needed in most circumstances.

DISPLAY

FOLD can now take an optional pixel number.

CONTOUR LOW/HIGH and VECTOR options can now be used together.

GRAPH can now AUTOScale the specified data subset.

DECOMPOSE

POS can optionally use a nearby wavefront to avoid phase ambiguity.

Overall beam direction can be set by sign in front of POS or DIR entry.

SELECT ray flags has been increased in number by a factor of 4.

MODEL

BSDFDATA internal buffer increased from 2000 to 10000 BSDF values.

Number of PLOTted incident/specular directions increased from 5 to 7.

SUM is now fully recursive, i.e. it can reference other SUM models.

USERBSDF now also passes wavelength to user-defined function.

SYSTEM now stores and recalls any required user \$FCN definitions.

SHAPE of a beam can be the fundamental mode of a circular step-index FIBRE.

FIELD can now COUPLE or MULTIPLY the current distribution with a previous one.

Option to temporarily override window directions and size on any plot.

MATRIX can recall the orientation (or inverse) of a given surface entity.

EMITTING

OBJECT/ENTITY can direct rays parallel to local surface normal.

DISK/RECT can now be of zero size (point source).

SEED QUASI with zero seed value implements a special non-random case.

PLOT ENTI/SURF/EDGE/CURV/LENS now autoscale on the given set of entities.

REPLOT interrogation coordinates also put in output buffer for \$GRAB and

\$PLAY.

Macro Language

&STO/&RCL write/read variables to a portable text version of REG file.

&REG with no text string now outputs numeric values in assignment format.

\$FAST can now handle complex number input and scaling.

\$GUI sends window-related commands to the GUI front-end.

\$IO OUTPUT file ... now has complement to ONLY action called BOTH.

New math functions

BK0 and BK1 modified Bessel functions.

\$PLOT EACH option added.

\$READ input from a file and automatically return to calling macro or file.

\$ITER

Can act as a simple nested loop structure when no function is specified.

&ITER same but automatically cancels output during execution.
\$UNVAR controls uninitialized variable/register messages.
\$ABORT returns (from any depth and as soon as possible) to the command prompt.
\$LEAVE can optionally specify the number of construct levels.

Others Changes

*.REG and (as usual) *.SYS files are incompatible with previous versions.
Added option to stop or resume an interrupted (paused) calculation.

ASAP-Pro changes in maximum numbers:

Objects	2,000 ==>	3,000
Pixels	1,000 ==>	1,600
Paths	50,000 ==>	160,000
SVD fit	3,484 ==>	8,919
FFT size	512 ==>	1024
Ray Flags	8M ==>	32M (SELECT still works with any number of rays)

All timing outputs display both elapsed and CPU times.

USERPROG user-programmable Fortran routines

Compile with Digital Visual Fortran (Microsoft Powerstation discontinued).

Additional arguments have been added to scatter and apodization routines.

USERSAG use and meaning has completely changed.

Default USERANGS and USERBOTH can now access data files of any size.

SAVE file record length increased from 11 to 12 words (44 to 48 bytes).

Some source code takes advantage of the latest Fortran-95 features.

Eliminated all use of the "config.bro" file.

Superfluous EMIT, ROUGH, and TOWARDS calls to random number generator removed.

Bezier division of full circles (ARC and SWEEP AXIS) increased from 3 to 4.

Old DIS file format (version 5.0 and lower) no longer supported.

Outputs adjusted to handle ray numbers greater than 100 million.

IGES to ASAP Translator

Now imports COLOR index information.

Added new (command line) options:

Bring in ALL independent geometry not just visible surfaces.

Set maximum degree-product of any PATCHES.

Suppress ASAP plot commands near end of file.

Route messages to a LOG file instead of the console.

Plot denser MESH for spline surfaces.

Specify ROOT name for object hierarchy.

Create specific SURFace-based objects when possible.

Use parametric space TRIMming instead of model space bounds when possible.

Use a FUZZier spatial tolerance than global section Model Space Resolution.
Uses Levels and Groups to form object name hierarchy.
Automatically simplifies piecewise linear curves.
Very dense higher-order splines are automatically sub-sampled.
Simplifies spline surfaces that are actually extruded or revolved curves.
More intelligently sets the SWEEP point for model space bounding curves.
Automatically scales IGES units to ASAP system UNITS at time of import.

B.7 VERSION 5.1

Release Notes for ASAP 5.1 Kernel as of April 9, 1997

Mandatory Changes (NO BACKWARDS COMPATABILITY!)

DISPLAY

IMAGE command has been renamed PICTURE (see new IMAGE transform commands).

TABLE command has been renamed SECTION (see new TABLE command).

USERAPOD BOTH default may change in future (switch to new USERAPOD ANGLES).

New Commands

MODEL

BINOMIAL/TRINOMIAL variants of POLYNOMIAL help enforce reciprocity.

BSDFDATA indirectly interpolates from entered values.

EMITTING ...

ENTITY randomly distributes rays over the surface(s) of SURF, LENS or EDGE.

OBJECT randomly distributes rays over the surface(s) of an object.

IMAGE the current ray positions through a specified LENS entity.

LENS

IMAGE a given global point through all or part of the current lens.

CURVE/EDGE

IMAGE the current curve/edge through a specified LENS entity.

SAWTOOTH edge given shape and number of teeth.

HELIX creates a general helical curve including an elliptical spiral.

DISPLAY

TEXTFILE can write data to a file in a completely user-definable format.

TABLE numeric display of current data for arbitrary number of rows/columns.

SPECTRUM simplifies flux weighting of rays as a function of wavelength.

OBJECT

FRESNEL/SPLIT/LEVEL settings can now be different for every object.

IMMERSE allows creation of rays in MEDIA other than number 0.

SURFACE/FUNCTION

ALTER allows editing of polynomial coefficients.

USERAPOD ANGLES replaces default USERAPOD BOTH definition.

APODIZE is like USERAPOD but immediately alters fluxes in current ray set.

Enhanced Commands

SUBSET ...

Will actually produce a smaller VIRTUAL.PGS file on PC platform.

Can also optionally RESET to just "created" rays.

PATH output can now handle larger ray and object numbers.

SCATTER;TOWARDS large subtense areas handled more accurately.

EMITTING ...

HELIX has an optional longer but more flexible format.

IES can optionally rotate angular pattern about polar axis.

USERAPOD

DIR/POS profile interpolation can be product instead of sum (and plotted).

Comment string is now passed to routines (can be data file name).

Function definitions can now be passed up to 50 user coefficients.

AXIS now affects output from STATS, EXTREME, HISTORY, and LIST RAYS.

SELECT ...

EVERY nth ray has been added.

F(lux), optical path L(ength), and (W)avelength thresholds have been added.

DISPLAY

CONTOUR options

...VECTOR now puts color of contours into 3D file.

...GRID draws complete lines instead of short TICS.

IESFILE can be optionally PADded in those regions of sphere not covered.

RAY can create LENS entity from trace information (e.g. for IMAGE transform).

HALT can now turn OFF coordinate direction check.

RADIANT can MAP the radiant intensity or true radiance over an AREA.

SURFACE

CARTOVAL can now take ELLIPSE/RECTANG aperture option.

USERFUNC \$FCN definition can now be passed up to 63 user coefficients.

FITTED

Can optionally put input data points into current 3D VECTOR file.

Added direct fitting to a PLANE, PARABOLOID, SPHERE or ELLIPSOID.

AXICONIC can LIST intermediate curve coefficients.

LENS

IDEAL Jones matrix behavior for non-normal incidence has been improved.

GROUP

0 option does not redefine group, uses last (current) group definition.

SOURCES can now be moved along with any objects.

REPLOTT automatically GROUPs all objects interrogated for their numbers (#).

CONSIDER by default uses last GROUP of objects.

WAVELENGTHS maximum number for interpolation increased from 15 to 25 in Plus.

COATING maximum number of unique LAYERS increased from 60 to 100 in Plus.

SPREAD/FIELD

Doubled internal complex precision to avoid roundoff positivity violations.

Severe paraxial invariant departures (>1 wave) now abort calculation.

DECOMPOSE DIR number of beams can be entered as fraction of maximum possible.
UNITS can now take optional flux label (e.g. Lumens or Watts).
MEDIA;...GRIN function can now be ARRAYed.
PLACE can also locate something at a given parametric value on a CURVE/EDGE.
MODEL

USERBSDF \$FCN definition can now be passed up to 63 user coefficients.
TRACE ...

New KEEP option allows definition of additional rays after tracing.

STATS also displays BSDF statistics for each scatter model.

OBJECT

BOUNDS has new option to force a given POINT to be valid.

CURVE/EDGE

SAG option can now take a vertex position different from reference point.
OBJECT maximum name length has been increased from 24 to 128 characters.
GROUP/CONSIDER can now use wildcard "?" in object names.
LIGHT can define multiple collimated or point sources for the RENDER command.
CAD IGES attaches OBJECT name to surface using a NAME Property (406.15).
WARNINGS limit number can be ignored or non-fatal when exceeded.

Other Improvements or Changes

SEARCH on object 0 or other affects creation of EMITTING or child rays.
Entity geometry storage doubled in Plus version.
TRACE no longer STEPs generated child rays.
A parent ray that dies at an interface can still generate children.
A ray can now consecutively hit the same LENS conicoid many times.
Each MEDIA and COATING can now have different interpolation WAVELENGTHS.
Length and flux UNITS have been added to most plot labels.
"a,b,c" direction triplet can be the TANGENT to a CURVE at given parameter.
DIS file format has changed slightly but previous version can still be read.
EMIT IES and DISPLAY;IESFILE now follow LM-63-1995 standard.
Explicit surface functions can now have either 0 or -1 implied Z coefficient.
Material CATALOGs can contain data in wavelength/index PAIRS format.
More robust intersection algorithm for parametric (CURVE-based) surfaces.
As usual, *.SYS files are incompatible with previous versions.

Macro Language

\$WAIT pauses program execution for a specified number of seconds.
Nesting of Macros, \$DO/\$ITER, \$IF..THEN increased to 99 levels.
\$NEXT branches to top of a \$DO/\$ITER loop.
New intrinsic functions

LPW returns Lumens per Watt for blackbody of given temperature.

EYE returns normalized photopic or scotopic eye response given wavelength.

FILE returns next available file (logical unit) number.

\$SHOW also displays number of registers/variables used and storage left.

Maximum number of registers/variables increased from 858 to 1768.

Maximum number of \$FCN definitions increased from 10 to 20.

\$IO OUTPUT redirection can suppress console echo and write ONLY to given file.

\$SCR

Can now have optional on-line help text.

Boolean (toggles,checkboxes) fields have been added.

GUI cancel button controlled/processed with SCR_CANCEL variable.

\$LOC makes given registers/variable names local to the current macro.

Utilities

DISFILE.PRO reads distribution file data into an IDL array.

IGES to ASAP translator

Now filters out unnecessary COMPOSITES and BOUNDS.

Added partial support for 141 Boundary and 143 Bounded Surface entities.

Limit on number of trimming curves per object has been removed.

Name Property (406.15) attached to surface entity now used as OBJECT name.

APART 9.7 translators also pass equivalent imaging LENS to ASAP.

APARTDAT.LIB contains BSDF data for all static APART models.

DXF translator now creates only those BLOCK macros that reference surfaces.

B.8 VERSION 5.0

ASAP 5.0 Release Notes as of May 24, 1996

General

New graphical user interface and on-line documentation on most platforms.

Macro Language

\$PLAY can now turn ON or OFF the capturing of output into the replay buffer.

\$PATH added for setting default directory for files.

ASAP Enhanced Commands

SHOW can now display ALL settings and not just a select set.

REPLOTT allows object number interrogation by pointing and hitting # key.

CONSIDER can now produce a list of the currently considered objects.

INTERFACE DIFFRACT efficiency can reference a polychromatic COATING PROPERTY.

GAUSSIAN command has shorter form for symmetric fundamental mode.

SELECT ray by current MEDIA has been added.

MOVE can now specify the closest approach TO a POINT.

...PIXELS option will temporarily override number on any pertinent command.

ASAP New Commands

DRAWING produces 4 views of 3D VECTOR file data (like old ENDRAW program).

SPRINT and QPRINT are shorter versions of PRINT.

DISPLAY;IES writes out a Standard Photometric file of angular distribution.

ASAP Changes

Forward referencing of undefined entities is now strictly forbidden.

Maximum number of sources/wavelengths has been increased from 127 to 1024.

Maximum BOUNDS entities per object increased from 20 to 30 (100 in Plus).

VIRTUAL.PGS and *.SYS files are incompatible with previous releases.

ASAP Known Problems That Will Be Fixed in 5.1

Rays can leak through a parametric CURVE-based object in rare circumstances.

Fluxes in rays split at a CRYSTAL interface are off (especially at non-normal)

Utility Programs

DXF2ASAP will now read Release 13 files but will not process any new entities.

B.9 VERSION 4.1

ASAP 4.1 Release Notes as of July 25, 1995

IMPORTANT CHANGE (requires modifying any affected user files)

OBJECT; signs of solid "q" factors have been reversed to conform with BOUNDS.

Other changed commands

PLOT ...

LOCALS/LIMITS plots only boxes, i.e. no longer INTERSECTS surface.

OBJECTS has been deleted since MESHES is now the equivalent (and default).

SEGMENTS now sets number of line segments in a 45 (instead of 90) degree arc.

Undocumented 4.0 Commands

EMITTING HELIX creates a straight helical filament source directly.

New Commands

SURFACE

PARAMETERIZE given local coordinate axis for meshing surface.

PLOT ...

ENTITIES plots all entity types (SURFACES, LENSES, and EDGES).

SURFACES directly plots surface entities which have local boxes.

DISPLAY

IMAGE produces "grey"-scale picture of distribution using SHADED process.

MODEL

VCAVITY is a geometrical model for a rough (random v-cavities) surface.

Enhanced commands

\$EDIT can now edit arbitrary files (not just inputs and macros).

\$HELP system is now menu-driven.

DISPLAY

WRITE can now optionally produce only the DIN or DIS version of file.

DIRECTION now has a legend to help distinguish the two curves.

AVERAGE can optionally smooth data using a median window technique.

PLOT ...

MESHES/FACETS work not only for EDGE objects but also LENS and SURFACE ones.

FACETS...HIDE option produces hidden-line perspective plot.

USERAPOD BOTH "usap3d.#" file can now accept data every degree in both angles.
XMEMORY can specify any name for page file (not just VIRTUAL.PGS).
LEVEL can generate scatter rays for ALL specular child rays (not just parent).
SCATTER can be ABSolute or RELative with separate reflect/transmit scaling.
SELECT now works on any number of rays (not just ones with bit flags).
FIELD can now ADD its results to a previous field calculation.
HISTORY can show ray SPOTS on objects instead of PLOTting trajectories.

Code Changes

As usual *.SYS files are incompatible with previous versions.
All known bugs have been fixed.

New Platforms

AIX 4.1 on IBM RISC/6000 and PowerPC.
Microsoft Windows 3.1 on Intel CPUs.

B.10 VERSION 4.0

ASAP 4.0 features as of 30-Nov-1994

Macro Language

Old MACRO files now called LIBRARY files

\$IO MACRO name[.MAC] ==> \$IO LIBRARY name[.LIB] (new expression syntax)
or \$IO MACRO name[.LIB] (old " ")

- *New simplified MACRO files contain just one macro of same name (no header)
On-the-fly macros put into individual files, current library not touched
Curly braces {} can no longer be used in expressions (use square brackets [])
- *Macro definitions can use curly braces {} to delimit body
\$MENU and \$SCR now take an optional title
Register storage increased to approximately 300 fixed plus 600 variable.
- *Variable and \$FCN names can now be up to 16 characters
\$SHOW displays current states
\$FAST can be used in a macro and take optional label and/or scaling factor
\$ITER now has two different ways to step through variables
\$ARGS SCR pops up an editable argument template when a macro is called
\$GRAB now issues a non-fatal warning and zeros register when number not found
\$PAGE sets page size for pausing output
\$PLOT can now suppress screen graphics and control immediate plot prompting
- *Nested \$IF .. THEN, \$ELSEIF .. THEN, \$ELSE, \$END structures have been added
- *\$EXP selects OLD left-right or NEW operator-precedence expression syntax
\$ASK can prompt for more than one numeric input
\$FCN definitions can now be recursive i.e. call each other
- *Multi-record blocks for \$DO and \$ITER delimited by curly braces {}
\$LEAVE forces immediate exit from DO loop, macro, or input file.

ASAP changed commands (no backward compatibility)

SURFACE;OPTICAL semi-diameter "a" entry removed (use aperture option)
EMIT FILAMENT function (\$FCN) must return an additional apodization value
USERAPOD BOTH direction cosine clipping moved to new CLIP DIR command
OBJECT by default uses only last entity (not last two EDGES)
TRACE and RAY no longer automatically PLOT if a previous ...OVERLAY
FRESNEL command no longer has any effect on INTERFACE command

ASAP new commands

MODEL;SUM defines a model that is the sum of previous ones

*ENTITIES [OBJECTS] for simple input without SURFACE/EDGE/LENS/OBJECT commands
MAP scans objects and produces depth (and optionally SLOPES) map
CURVE
 ARC defines a portion or all of a circle
*SWEEP last curve into a surface for a single entity object
COMPOSITE combines a set of previously defined curves into one
SMOOTH a piecewise linear curve
RACETRACK ... SPLIT option added for backward compatibility
INVERT reverses a curve's parametric direction
CONIC creates single quadratic segment from implicit coefficients of conic
SURFACE
*SAMPLED explicit surface by interpolating deformation (and slope) data
BICONIC creates a surface from two distinct conic profiles
 *ARRAY is a 2D version of MULTIPLE, i.e. planar arrays of surfaces
 SOLID sets up surface and its local box for use as a BOUNDing volume
*EMITTING RAYS accepts an arbitrary table of rays
 IES uses data from an IES Standard Photometric file
 HELIX creates a helical curve (coiled wire)
PLOT LOCAL limit boxes for surfaces
 MESH/FACETS also handles new SURFACE;SAMPLED objects
*CLIP POSitional and/or DIRectional ray coordinates during creation
LENS;COMPOSITE combines a set of previously defined lenses into one
OBJECT
 FACETS sets number of facet subdivisions per mesh patch
 REDEFINE basic options (SURFACE/NORMAL, THICKNESS, COLOR, name)
*RADIANT calculates pattern in full spherical angles (and spatial MAP)
UNITS set the length units (e.g. INCHES, MM) for the system geometry
DISPLAY;FOLD averages the data about one or both centers
*GAUSSIAN creates the ray/beam set for a coherent astigmatic Gaussian mode
UPDATE controls how geometrical entities are update in new storage scheme
BEAMS sets default coherence, propagation, and shape characteristics

ASAP enhanced commands

SURFACE

*PLANE,OPTICAL can take RECTANG or ELLIPSE aperture option
TORUS cross-section can vary from elliptical to rectangular
REVOLUTION can take optional DECENTER before revolving
FITTED can remap the order of coordinate data
OPTICAL can take a optional second surface (e.g. both sides of a lens)
LIST ELLIPS displays polarization ellipse information
RAYS outputs basic ray/beam data in EMITTING RAYS input format

FIELD DELTA can plot polarization ellipses when FRESNEL BOTH set
INTERFACE command can take explicit refractive index and define media name
MODEL;POLYNOMIAL FIT/SVD direction input can be optionally in spherical ANGLES
also generates plot of given raw in-plane data
DISPLAY;VALUES can optionally place results directly in specified registers
ACCURACY setting not overridden anymore (was on some calculations)
RAYS can now properly restart from an XMEMORY MIN paging file.
...COLORS option can now be used to temporarily redefine the set of colors
PLOT EDGES/LENSES/LOCALS can take a specific range of entities
FFAD also prints best-fit planar/curved focal surface (center,normal,radius)
STATS with no argument produces a one line per object summary
PLOT RAYS can optionally plot every nth ray
CURVE
POINTS has shorter planar version
CHARACTER can take complete strings instead of just one character
SAG to surface can be either ABSolute or RELative
PROFILE can do a QUICK plot with no exact boundary clipping
SURFACE;TEST can now be perpendicular vector from any given AXIS
LENS;WEDGE now takes an optional thickness
*EMITTING FILAMENT can reference a CURVE instead of a \$FCN definition
EMITTING DISK/RECT surfaces can optionally radiate ISOTropically
*SELECT test pairs take the logical operators AND,OR between them
also issues message if ray flag storage exceeded
OBJECT
DEFORM can be combination of two explicit surface functions
ROUGHNESS RANDOM slope error can select from several distribution functions
*MODEL for random normal deviations that match any scatter model
BOUNDS can use the interior or exterior of SOLID volumes
CAD can now do a basic IGES export of curve, lens, and sample based objects
WAVELENGTH...UNITS can be set by name (e.g. ANGSTROMS, MICRONS)
*EMITTING can produce a single-source weighted composite of individual emitters
SOURCE DIR GRID now takes into account USERAPOD DIR (and new CLIP DIR)
TRACE GRAPH now allows control of maximum magnitude
GRID OBJECT now takes standard RANDOM option
MODEL ... PLOT creates an additional plot of log BSDF versus angle from normal
EXPLODE can optionally store commands in a macro.
SCATTER; TOWARDS can designate virtual edges for important areas

ASAP general enhancements

Implicit polynomial surfaces and rational Bezier curves can go to 10th order
(explicit and order-doubled polynomial surfaces to 20th)

*New much more efficient memory management for object surface data
Replaced Bezier intersection algorithm with a new more-robust one
Built-in interactive run/edit loop with file menu

ASAP specific enhancements

*The following are virtually unlimited in number now:

- Conicoids per LENS
- Points per CURVE/EDGE
- Entities per OBJECT
- SURFACE;FITTED EXPLICIT points
- MODEL;POLYNOMIAL FIT values

*Plot can be sent directly to PLOTTER and/or file at end of screen drawing
BOUND "outer" entities now used in default LIMITS and plot autoscaling
SCHOTT.CAT has been updated with new Sellmeier coefficients
COATINGS can be named and then referenced by that name

ASAP code changes

VIRTUAL.PGS, *.SYS, and *.REG files are incompatible with previous versions
Old character graphics are now suppressed by default (except for DISPLAY;DMAP)
User-programmable Fortran routines (now in USERPROG.FOR) have been redone.
Some new code written entirely in Fortran-90 syntax

Utilities

Plot utilities now start at top when putting more than one plot on a page
Color PostScript support has been added to PSCRIP and DIS2PS
AutoCAD importer (DXF2ASAP) upgraded to make use of new ASAP 4.0 capabilities
New utility (IGS2ASAP) for importing all IGES parametric surfaces

B.11 VERSION 3.3

ASAP 3.3 release, November 1993

Approximately two dozen minor bugs have been fixed

Macros files now supplied for components from the following manufacturers:

Edmund, Newport, Melles-Griot, and Spindler-Hoya

Maximum number of geometrical entities increased from 600 to 900

Compatible with new 9.3 APART release for DOS and Unix systems

Access to user-programmable Fortran routines available again on some platforms

AutoCAD translator significantly improved

PC version:

Now uses Pharlap 5.1 extender which eliminates slow load under DOS 6.0

ASAP.BAT temporarily remaps arrow and escape keys for use during \$SCR input

Bulletin board system now in place for technical support (bug reports, patches)

B.12 VERSION 3.2

March 1993

Changes to Macro Language:

- Number of lines entry on macro call eliminated (remove dummy ALL)
- FBI Fractional Blackbody Integral function added
- \$PLAY replays previous output or input
- \$GRAB reads numbers from output buffer into registers
- Up to 16-character macro names now allowed
- Maximum number of lines per macro now 9999
- Optional end-of-macro literal for in-line and file definition
- \$ITER for plotting or minimizing registers as a function of other registers
literal:literal construct
- \$DATIM command turns ON or OFF date and/or clock time outputs
- \$IO OUTPUT redirection can send output to screen and a file simultaneously
- \$LOOK and \$SCR now allow user to control attributes (colors) of display
- \$BEEP beeps!
- \$EDIT for on-line editing of macros and selected portions of input file
- \$PLOT and \$VIEW for on-line processing 2D and 3D graphics files
- Numerical entries can start with a parentheses (leading + not needed)
- Comments can no longer be delimited with parentheses
- &MENU allows user to select a macro for execution with the graphics cursor

Renamed ASAP Commands:

CURVE ==> REVOLUTION (CURVE is now an alias for EDGE)

Moved ASAP Commands:

- OBJECT;...RANDOM option moved to new ROUGHNESS command
- DEFORM moved from a SURFACE to an OBJECT modifier

New ASAP Commands:

- MISSED controls missed ray plotting
- DECOMPOSE field into a POSITION or DIRECTION grid of beams
- SURFACE;FCN defines a macro "wrapping" function
- ACCURACY of calculations can be traded off against speed
- LIGHT a scene for rendering

RENDER the current system geometry from an arbitrary camera point
COLORS can be reassigned to group or objects by interface properties
ALLOWED same as SEARCH but applied after finding next intersection
EMITTING FILAMENT creates an arbitrary "wire" of rays
SUBSET reduces the rayset to only the currently active rays
CAD command creates Autocad DXF file of EDGE and LENS objects
OBJECT;ROUGHNESS for micro and/or macro RMS roughness
SURFACE;ZERNIKE polynomial sag
DISPLAY
 TABLE displays or transfers to registers a section of data
 ANGLES converts direction cosine space data to spherical angle space
 VALUES list specific functional values given actual coordinate pairs
 OFFSET shifts origin of distribution
EXPLODE lens conicoids into separate object surfaces
RESET program settings to startup values
EDGE; ALTER coordinates and/or connection factor of points
GRID HEX creates a hexagonal grid of rays

Changed ASAP Commands:

EDGE;POINTS are quadratic rational Bezier (conic) segments instead of ellipse
PLOT FACETS entries have different meaning
SPOTS...ATTRIBUTE 0 suppresses all plotting now

Enhanced ASAP Commands:

RAY now creates a GALOP input file for the sequence of surfaces encountered
FIELD/SPREAD/SPOTS/OPDMAP distributions with object or edge CLIPPING
DISPLAY;FFT absolute output size added to relative
SOURCE WAVEFUNC can be actual wavefront surface
SHAPE and RAYSET allow literal beam shape names in addition to integer
USERAPOD with GRID RECT or ELLIP improved
SURFACE;FITTED has several addition options
Direction specification can be in LOCAL system or NORMAL at limits box center
ROTATE ABOUT an arbitrary 3D axis added
OBJECT can reference up to 60 edges or surfaces to form meshes or solids
PLOT MESH and OBJECTS options added
FFAD can add a REFERENCE box to plot
LENS;IDEAL takes a complex 2x2 Jones matrix for altering polarization
NUMBERS [NAMES] lists numbers and names of media and objects
PARABASAL can now take override table of transverse heights and angles
DISPLAY;MODIFY can now use a macro function for non-linear operations

OBJECT;INTERFACE takes multiple DIFFRACTs for multiplexed HOEs
 MEDIA;...CRYSTAL for uniaxial birefringent materials
 FIELD...DELTA will plot vector E-field if FRESNEL BOTH set
 SELECT has new PATH and radial options
 USERAPOD BOTH for 3d apodization of random emitting volumes
 MODEL;VANES with DIFFRACTION from serrated edges
 SURFACE;GENERAL EXPLICIT surface definition (sag equation to 14th order)
 OBJECT;DEFORM with user-defined macro or explicit surface function
 DISPLAY;WRITE also generates editable/portable ASCII *.DIN file
 MODEL;HARVEY can take additional shoulder entry
 DISPLAY;CONTOUR can do absolute levels and tic marks
 OBJECT;INTERFACE COAT 0 handles polychromatic properties of bare surfaces

better

SURFACE;TUBE can optionally create INNER or OUTER right cylinder
 DISPLAY;REDUCE by default will "crop" a distribution
 SEED allows selection of QUASI random number for EMITTERS
 XMEMORY now controls 3 different VIRTUAL.PGS file sizes
 CONTOUR...VECTOR option writes contours into 3D vector file
 USER... commands can optionally use a \$FCN definition

ASAP Code Changes:

VIRTUAL.PGS, *.SYS, and *.DIS files are incompatible with previous releases

Maximum number of objects increased from 200 to 300

Coefficients per Surffunc	84	120
Conicoids per Lens	10	12
Points per Edge	30	999
Parabasals per Beam	8	9
Pixels per line	256	1000
Pixel map	256x256	362x362

Linear transformations carried out in double precision now

Distribution file headers now slightly larger for better labeling

Source code now uses Fortran-90 DO [WHILE] ... ENDDO through out

Ray/beams now properly SCALE and/or SKEW to maintain wavefront invariants

VIRTUAL.PGS deleted before creating new ray set

XMEMORY OFF now startup default and reset by PARABASAL

Faster more compact SAVE for HISTORY (but not STRAP)

SAVE record pointer now first number instead of eighth

TEST POINT now translated with MULTIPLE surfaces

DISPLAY.DEF and CONFIG.CFG files replaced by CONFIG.BRO file

SOURCE POS/FOC now adjusts fluxes for isotropic point source

Control-C will now interrupt most calculations without exiting program

New Utilities

DXF2RBT utility creates ASAP surface input from an AutoCAD DXF file

DIS2VGA displays distribution files with shading on VGA adapters

HEADER converts old distribution file headers to new format

Platform Specifics

Extended DOS version will run in foreground or background under WINDOWS 3.1

Workstation versions have new multi-window color graphics driver

B.13 VERSION 3.0

May 1991

Recent GALOP enhancements:

- Chromatic aberration correction
- Complete Schott catalog for glass selection
- Much improved hybrid optimization algorithm (SVD with PSD damping)
- New variables
 - Stop position
 - Edge (marginal) ray height (power redistribution)
 - Tilted/decentered aspherics that maintain 1st order layout
 - SINGLET and DOUBLET thicknesses
- Generation of 2D plot file in ANALYZE for later hardcopy

New RABET commands:

- LENSES are a sequence of up to 10 arbitrary conicoids (one object)
 - SEQUENCE entered directly
 - MIRROR surface
 - SINGLET lens
 - MANGIN mirror
 - DOUBLET cemented (achromat)
 - RIGHT angle prism
 - PENTA prism
 - WEDGE of glass
 - AFOCAL beam expander or reducer (mirrors and/or lenses)
 - ALTER any lens parameter
 - IDEAL lens module from ABCD matrix

- MODELS for surface scatter in extended version
 - LAMBERTIAN diffuse
 - HARVEY shift-invariant polished surface
 - POLYNOMIAL with actual BSDF data FIT option
 - VANES angled with rounded tips
 - USERBSDF programmed by user

EMITTING surfaces or volumes from a randomly created ray set

- TITLE and identifier for plots
- DISPLAY

COMBINE two data sets
REDUCE current data to smaller subset
... COLOR override option on all plot commands
MESH write to 3D vector file
ENCLOSED energy within a rectangle
RANGE reset for minimum and maximum functional value
LSQFIT variable normalization and singular value threshold
HISTORY of ray trajectories from SAVE file
SCATTER with MODEL assignment
TOWARDS an important area bounded by an edge
DOMACROS determines priority of user macro commands
SEGMENTS sets number used to draw elliptical arcs
SHOW current commands settings

Enhanced RABET Commands:

SEARCH LIST option added
EDGES can now have smooth elliptical arc segments
Dashed LINE option
OVAL is shape factored polygonal edge
RTRACK is racetrack
SURFACES
MULTIPLE linear displacements
HORN FIT to point/derivative pairs option added
BOUNDS checking with EDGES added
SPREAD automatically sums multiple wavelengths incoherently
FIELD sums all coherent contributions, just not current WAVELENGTH
FIELD ... DELTA option allows varying "time"
Relative indexing for ray and source numbers added to GET/PUT and SELECT
GET totals and averages for all the current rays
RAY, REPLOT graphical inputs
"?" displays current cursor coordinate
Equivalent commands written to USR file
DISPLAY
FFT any part not just modulus
RADIAL averages and/or plots function and/or integral
GRAPH has new options
CONTOUR level values written on plot
OPDMAP ... PSF option
TRACE GRAPH option displays progress of ray storage
USERAPOD input apodization both in position and directional space
SOURCE LINE and WAVEFUNC for arbitrary input wavefronts

Code changes:

RABET default array sizes increased:

Pixel sampling	128x128 to 256x256
Objects	160 to 200
Surfaces+edges+lenses	480 to 600
Media/coatings	40 to 50
Interpolation wavelengths	5 to 9
Simultaneous HOE orders	3 to 5
Important areas per object	3 to 10

SPREAD and FIELD from 2x to 20x faster

Faster SELECT command avoids ray paging

Glass MEDIA referred by name can be automatically created from SCHOTT.CAT

USERFUNC,USERSURF coefficients passed in double precision

Concave facets/polygons now allowed

BSI is now a set of subcommands in STRAP

Macro Language:

SGN math function added

\$TIC internal macro for elapsed CPU time

\$ARG controls macro argument prompting

\$SCR displays and/or edits register fields

\$CASE allows upper, lower, or both cases for commands

\$ECHO controls/overrides echo of input

Macro libraries:

UTIL contains many useful macros for lenses, prisms, etc.

ACCS2RBT permits reading or converting AccosV input files

OSL02RBT " OSLO "

2D graphics drivers:

TERMSP now supports UNIX X-Windows through XTERM emulator.

HPGLNP can output separate plotter files

PSCRIP creates encapsulated PostScript files, portrait or landscape

EPSONP now handles 24-pin printers as well as 9-pin

Documentation:

New format and packaging

TUTORIAL is now two volumes totalling almost 700 pages with an index

B.14 VERSION 2.9

June 1989

Easy-to-use GALOP interactive optimizing preprocessor is now available:

Graphical 1st-order layout, real-ray design and analysis
Contact us for specs and prices

Renamed RABET commands:

DIFFUSE ==> INTERFACE ... RANDOM
CONSTRAINTS ==> BOUNDS (Avoids any conflict with CONSIDER)

New or enhanced RABET commands:

SURFACES

HORN with polynomial cross sections
OPTICAL ... EXPAND conic into aspheric terms
TEST for desired surface branch
LOCAL coordinates and limits box
BEND a surface parabolically
RENORMALize coefficients
MULTIPLE power entry

EDGES; CHARACTER

SAG with default axis override
EXTREME rays in set listed
GRID ... optional RANDOMizing of points
FRESNEL 3 causes both polarizations to be traced
POLARIZ sets initial state
PLOT POLARIZATION displays current state
... TEXT can be added to most graphics display
LEVEL of random scatter
ARROWS on rays
OPDMAP of current wavefront
DISPLAY part of complex vector FIELD
DIRECTIONAL (polar or angular) diagram
SELECT enhanced
RAY enhanced with KEEP
COATING LAYERS and groups of layers can be physically defined
TRACE ... OVERLAY/TEXT options added

XMEMORY controls paging of extend ray/beam information

Miscellaneous changes or enhancements to RABET:

Basic polarization capability added

Several diffraction orders can be generated at each interface

More than one random ray and LEVEL of scatter allowed now

Order doubled symmetric polynomials (within LOCAL coordinates)

Parabasals stored in relative form (more accurate)

Title record added to distribution file header

Miscellaneous changes or enhancements to utilities:

ACADIN creates AutoCAD DXF files from 3D vector files

HIDDEN program now has on-line graphics and autoscaling

Composite terminal driver with optional before pause

Miscellaneous changes or enhancements to input language:

Nested user macros (up to 6 levels)

Relative jumps with \$GO

Labels for \$ERR and \$GO can now be up to 8 characters

Single line \$IF structure

Transcendental math functions (e.g. SQRT,SIN,LOG,EXP,ATAN,BJ0 ...)

Repeated entries @ with alternating signs

\$ASK, interactive prompting for any register value

Interactive prompting for unspecified macro arguments

\$MENU executes current user macros from a menu

\$SCR allows editing any set of registers via a programmable screen

Still in the works:

Non-lambertian COATING/MODELS in STRAP

\$ITER internal macro varies any group of program inputs
and plots or minimizes any program output

B.15 VERSION 2.8

July 1988

RENAMED COMMANDS:

STATS OBJECTS	==> PATHS
INTERFACE ... SCATTER	==> INTERFACE ... RMS/BSDF
INTERFACE ... RANDOM	==> OBJECT; # RANDOM ...
DISPLAY;REPLACE	==> DISPLAY;WRITE

ENHANCED COMMANDS:

\$GO	can jump directly to a particular macro line
\$IF	AND/OR constructs now permitted
MEDIA	Inhomogeneous (GRIN) added
MOVE	TO PLANE added
CUTOFF	Upper limit on # of objects encountered
CURVE	Additional entry and least-squares fit
FFAD	Ray SPOTS can be added to display
PARABASAL	Beam scaling entry replaced by beam divergence
RAY	SEARCH reset and interactive tracing
OBJECT	One edge faceted objects allowed
TRACE	Can STEP a given number of objects
LIST	Info on SOURCES and all parabasal rays
SPOTS	Plotting all parabasals, distribution file
END	Automatic system save to LASTEXEC.SYS
SYSTEM	FROM defaulted to LASTEXEC.SYS
RAYSET	can be completely recovered from last run
SELECT	based on previous ray objects added
USERSURF	rotations now saved and passed to user
SOURCE	can use last GRID to define distribution
DISPLAY	Allows the naming of distribution files
SHIFT	a distance along a given direction
FIELD	Saves specified distribution to file
PROFILES	New default operation

NEW COMMANDS:

WIDTHS	Sets beam width scaling factors
EDGE;LINE	Equally divided straight line
USERFUNC	User-programmable function (GRIN, surface)

IRRADIANC Sets irradiance direction for SPREAD/FIELD
DISPLAY

MODIFY Allows modification of data regions

CONTOUR Contour plot

RADIAL Averaging distribution radially about a point

CODE CHANGES:

VIRTUAL.PGS and *.SYS files are incompatible

Nested parentheses can be used in input arithmetic

Input registers can now be accessed by assigned literal

On-the-fly macro definitions require 2 prefixes, e.g. \$\$

Up to 6th-order 84-term SURFACES and 25-point EDGES

Parabasal ray errors now halt trace of that beam

New header for unformatted distribution files (FOR009)

Improved accuracy in OPL calculations

Numerous bugs have been squashed (SOP)

Default DEFORM and USERSURF definitions changed

General clipping and device support added to VIEW3D

ACADIN program for 3D file to AutoCAD-9 DXF conversion

IN THE WORKS:

Non-lambertian COATING/MODELS in STRAP

\$ITER internal macro varies any group of program inputs
and plots or minimizes any program output

More "bells and whistles" (SOP)

Interactive optimizing graphical preprocessor (4th qtr.)

ACADIN program for 3D file to AutoCAD-10 DXF conversion

B.16 VERSION 1.8

February 1988

COMMAND CHANGES:

PAGE ==> PIXELS
DUMP ==> SAVE (now with optional file name)

ENHANCED COMMANDS:

GENERAL requires reference point
RAY interactive input
FRESNEL polarization selection
DEFORM new default definition and AXIS option
SHAPE additional parameter
INTERFAC can reference COATING definitions
WAVELENG multiple wavelengths, UNITS option
ROTATE new default center of rotation
SOURCE multiple types now allowed

NEW COMMANDS:

\$FAST internal macro for fast numerical input
ALIGN new linear transformation
PLACE " " "
XEQ " " "
SELECT selects ray set or source number
COATINGS define spectral properties of interface
USERSURF user-programmable surface intersection

CODE CHANGES:

VIRTUAL.PGS and *.SYS files are incompatible
TEK10 and REGIS drivers are buffered to reduce I/O
Graphics codes 4, 5-30, 31, and 127 are different
Hundreds of new error/warning messages have been added
Numerous bugs have been squashed
Commands can now be entered while in graphics mode
Higher-order Hermite-Gaussian beams can be propagated
More general ways to specify direction vectors
OBJECTS and MEDIA can now be referenced by name

B.17 VERSION 1.0 REFERENCE MANUAL (1983)

B.17.1 INTRODUCTION

This manual describes a general raytracing computer program. The program has been designed to accurately model complex opto-mechanical systems. Similar to the GUERAP straylight analysis programs, it utilizes a very powerful geometrical approach that permits a nearly limitless variety of systems to be handled in a straightforward manner. As opposed to most other raytracing algorithms, all surfaces and ray data are referenced to a single global Cartesian coordinate system. Object surfaces are represented by a general 35-term polynomial (taken to fourth-order in the three Cartesian components) and can be bounded by other surfaces of the same general form. Therefore, anything from a simple plane with a polygonal boundary to an arbitrarily oriented elliptical toroid can be modeled precisely.

Bundles of rays can be traced through the system such that after every reflection or refraction each ray always transfers to the nearest object. Also, rays are allowed to interact with an object any number of times, i.e. multiple bounces. Therefore, the program easily handles the "funneling" effect of non-imaging radiation collectors as well as imaging systems. In addition, each ray may be assigned an arbitrary total flux that is reduced by volume and surface absorption (or increased in a gain medium) as the ray propagates through the system. The standard Fresnel equations are used to not only calculate (as a function of incidence angle) transmission losses at interfaces between two dielectric media, but also reflection losses at any dielectric/conductor interfaces.

B.17.2 PROGRAM COMMANDS

In this section, the format of each command will be shown followed by a detailed explanation of the command's function. A consistent notation is used. Any upper case letters or numbers represent the actual form of the entry. Lower case letters represent numerical entries to be determined by the user. Trailing lower case letters on literals and anything enclosed within square brackets are optional and will not trigger a program error if

omitted. Unless otherwise stated, the default value for most optional numeric entries is zero.

B.17.2.1 Defining The Geometry Of The System -

The program uses a single global Cartesian coordinate system with components x, y , and z . "Surfaces" are the basic geometrical building blocks of the program. They are represented mathematically by a single equation $f(x,y,z)=0$ and are either infinite in extent or self-enclosed (e.g. a plane or sphere, respectively). The volumes of space between surfaces can be assigned isotropic properties and are called "media". An "object" is a surface bounded by other surfaces and represents an actual physical entity that can refract or reflect radiation.

B.17.2.1.1 Surfaces -

The program is capable of handling any surface that can be represented by a general fourth-order polynomial in the coordinates x, y , and z with 35 constant real coefficients. Therefore, anything from a plane to an arbitrarily oriented elliptical toroid can be modeled.

```
--- SURFACES [ i ]
```

Signals the program that surface definition commands follow. "i" is the surface number to be assigned to next command. The default value is one more than the highest surface number previously defined (set to one at start of program execution).

```
--- GENERAL [ c c' c" .. ]  
--- XiYjZk c c' c" ..  
:
```

An entirely general surface may be created by entering any of the 35 polynomial coefficients directly. This can be done either on the GENERAL command starting with the lowest order coefficient or on successive commands that begin with the coefficient of the term represented by $XiYjZk$ where

i, j, k are integer powers ($i-j > j-k > k$). Any coefficients not entered are set to zero.

Because it would be tedious for the user to enter even simple geometrical shapes in this manner, the program is equipped with special commands (for the more common surfaces) that generate the polynomial coefficients with a minimal amount of input. More complicated geometries can be built up from these special surface commands and then referenced using a single macro command.

B.17.2.1.1.1 Special Surfaces With No Preferred Symmetry -

```
--- PLANE NORMa1 a b c x y z
      X a a'
      Y a a'
      Z a a'
```

A plane surface containing the point (x,y,z) and with unit normal vector (a,b,c) or zenith and azimuth angles (a,a') as measured in degrees from the axis defined by the third entry. The normal vector always points into the space on the positive side of the surface where $f(x,y,z) > 0$. As a protective measure, the program always renormalizes the vector length to one.

```
--- PLANE POINTs x y z x' y' z' x" y" z"
```

A plane surface that contains the three points (x,y,z) , (x',y',z') , and (x'',y'',z'') . The positive side of the plane is always opposite the origin of the global coordinate system.

```
--- ELLIpso1d a b c [ x y z ]
```

A general ellipsoid with semi-lengths along each axis of (a,b,c) and center at (x,y,z) . The normal vector points out away from the center.

```
--- REPEat i
```

The new surface created is the same as surface number "i". The surface altering commands described in a later section can then be used to make this surface distinct from its "parent".

B.17.2.1.1.2 Special Surfaces With An Initial Axis Of Symmetry -

The second entry on the following commands designates the axis of symmetry (either "X", "Y" or "Z") for the surface.

```
--- PLANE X x
      Y y
      Z z
```

A plane normal to the axis of symmetry at the value given by the third entry. The normal vector points along the positive coordinate direction.

```
--- CONE X x y z x' y' z'
      Y y z x y' z' x'
      Z z x y z' x' y'
```

Using the first form as an example, a general conical surface symmetric about the "X" axis that passes through an ellipse normal to the axis at "x" with semi-major widths of "y" and "z" and a parallel ellipse represented by the primed quantities. If $yz'=zy'$, then the cone is a second order surface. Otherwise, it is of fourth order. The normal vector points out away from the symmetry axis.

```
--- CONE X x y z x b c
      Y y z x y c a
      Z z x y z a b
```

The form of these commands is the same as the previous ones except that the last two entries for this general conical surface are the angles (in degrees) that the two orthogonal projections of the sides of the cone make with the symmetry axis.

```
--- OPTIC X x r [ k d h e f g ]
      Y y
      Z z
```

Standard "optical" surface normal to the axis of symmetry at a value given by the third entry. "r" is the vertex radius of curvature and is negative if

the center of curvature is on the negative side of the surface. "k" is the conic constant (e.g. 0 is a sphere, -1 is a parabola, etc.). "d" is the fourth order aspheric deformation coefficient. "h" is the clear aperture height used in fitting the surface to the higher order aspherics "e", "f", and "g". At the vertex the normal vector points along the positive coordinate direction.

```

--- CURVe X a a' ...
          Y
          Z

```

Takes the two-dimensional curve defined by the ascending coefficients (up to 6, i.e. second order) and rotates it about the given axis to form a three-dimensional surface of twice the order of the curve.

```

--- FERMat X x h x' h' s [ x" h" ]
          Y y h y' h' s [ y" h" ]
          Z z h z' h' s [ z" h" ]

```

A surface that obeys Fermat's principle in any plane containing the axis of symmetry. This means that the distance in such a plane from the first focus (primed coordinates) to the second focus (double primed coordinates) through any point on the surface (unprimed coordinates) is stationary with respect to small changes, i.e. constant. If the second focus is not specified, then it is assumed to be at infinity and "s" is the angle (in degrees) that it makes with respect to the axis of symmetry. The normal vector points out away from the symmetry axis.

B.17.2.1.1.3 Altering Surfaces -

The polynomial coefficients of a surface can be altered to form a differently shaped or oriented surface. However, the degree of the polynomial never changes.

B.17.2.1.1.4 Nonlinear Transformations -

Nonlinear transformations change the surface's shape and type.

--- ASYMetric u v

This command can be used after the CONE, OPTIC, and FERMA commands to essentially distort the rotational symmetry of the surfaces. For example, if "Z" is the axis of symmetry, then the surfaces are only a function of the distance from the axis squared, $r^2 = x^2 + y^2$. This command replaces this with $ux^2 + vy^2$ so that a more general class of surfaces is possible such as "toric" OPTIC surfaces. Another example is setting either "u" or "v" to zero so that a surface with cylindrical symmetry becomes an infinite trough in the non-zero direction.

B.17.2.1.1.5 Linear Transformations -

Linear transformations change the scaling or orientation of a surface.

Any number of the following commands can be applied after any surface definition. The LIST option causes the resulting 4 by 4 transformation matrix to be printed.

```
--- SHIFt x y z [ LIST ]
      X x
      Y y [ LIST ]
      Z z
```

Translates the surface by the relative components (x,y,z).

```
--- SCALE x y z [ LIST ]
      X x
      Y y [ LIST ]
      Z z
```

Scales the surface by the relative components (x,y,z).

```
--- SKEW X a Y [ c LIST ]
      Z
      Y a Z [ c LIST ]
      X
      Z a X [ c LIST ]
      Y
```

Skews the surface in the first direction by an angle "a" measured in degrees from the second direction. The corresponding coordinate of the skew center is optionally given by "c".

```
--- ROTate X a [ y z LIST ]
      Y a [ z x LIST ]
      Z a [ x y LIST ]
```

Rotates the surface through an angle "a" (in degrees) about an axis displaced by the fourth and fifth entries from the coordinate axis given by the second entry. The sign of the angle is determined by the right-hand rule where a positive value results when the thumb points along the positive direction of the rotation axis.

B.17.2.1.2 Media -

```
--- MEDIA [ m ]
--- n a [ *name ]
:
```

Starting with medium "m", media with real refractive indices "n" and volume absorptions "a" are entered. The default value for "m" is one more than the largest medium number defined and is set to one at the start of program execution. The refractive index is always a positive number greater than or equal to one. The volume absorption is in inverse length units and is positive for an absorbing medium or negative for a gain (lasing) medium. "name" is a descriptive designator that the user can assign to this medium (only the first 10 characters after the asterisk are stored for future reference by the program).

B.17.2.1.2.1 Special Case Of Reflecting Substrate -

For a medium on the substrate side of a reflecting object surface, the format of the command is identical to the one in the previous section except that "n" is the negative of the surface reflectivity at normal incidence and "a" is a value that controls the way the reflectivity varies with angle from the normal. A value of one approximates the variation in reflectivity of a polished metal surface for radiation polarized perpendicularly to the plane of incidence. Use a value of two to approximate unpolarized radiation. In any case, the reflectivity at grazing incidence is always unity.

B.17.2.1.3 Objects -

```
--- OBJECTs [ j ]  
--- i [ m- m+ k k' k" .. *name ]  
  :
```

Begin defining objects starting with number "j". The default value for "j" is one more than the largest object number defined and is set to one at the start of program execution. "i" is the surface number that defines the actual surface of the object. "m-" and "m+" are the numbers of the media on the negative and positive sides of the object surface. The "k"s are the numbers of the surfaces that are to be used to define the boundaries of the object surface. If these numbers are positive, then the object is assumed to be on the positive side of the corresponding surface. A negative number means that the object is on the negative side. "name" is a descriptive designator that the user can assign to this object (only the first 20 characters after the asterisk are stored for future reference by the program). Any SHIFT, SCALE and /or ROTATe commands that follow an object definition are applied to the object surface and all boundary surfaces, i.e. the entire object.

B.17.2.1.4 Relative Indexing -

A relative indexing scheme is available when referencing surface, media, or object numbers within commands. Instead of using the actual absolute

number, one can specify a number relative to the largest number defined by using a decimal entry of the form ".i" where i is an integer between 0 and 9 inclusive. ".i" is equivalent the largest number defined minus i. For example, "-.1" is the negative of the next to the largest number defined. If the user wants to specify the negative of the largest number defined, then a "-.01" must be used.

B.17.2.2 Input/Output Control Commands -

--- EXIT

Immediately terminates execution of the program and returns control to the system monitor.

```
--- PRINT [ SHORT      ]
          SURFACES
          MEDIA
          OBJECTS
```

Prints all the data specifying each surface, medium, and object or just the selected information. The SHORT option prints everything but the surface coefficients.

```
--- PLOT X a a' Z d d' Y [ f f' n ]
          Y      Z
        Y a a' Z d d' X [ f f' n ]
          X      Z
        Z a a' X d d' Y [ f f' n ]
          Y      X
```

Produces plots of the crosssections of all the objects in the system. "a" to "a'" is the range of corresponding coordinate values to be plotted across the page. "d" to "d'" is the range of corresponding coordinate values to be plotted down the page. "n" is the number and "f" to "f'" is the range of corresponding coordinate values that the system is to be cut into the plane of the paper. A value greater than one for "n" produces a three-dimensional effect.

```
--- PAGE l [ r ]
```

Sets the width of the page to "l" columns for all subsequent plots. Initially, 132 columns is assumed by the program. "r" is the ratio of the lines per inch down the page to the number of characters per inch across. This is initially assumed to be 6/10 and used to assure that the plots are of the correct proportion.

```
--- TI820 [ OFF ]
```

Speeds up the outputting of subsequent plots by replacing blank areas with horizontal tabbing. The user's terminal must be a Texas Instruments 820 KSR terminal equipped with the DFC (Device Forms Control) option and Configuration Code 81 enabled. Use the "OFF" entry to turn off the feature (the default condition).

B.17.2.3 Raytracing Commands -

B.17.2.3.1 Optional Commands -

The commands that actually initiate raytrace execution (RAY and TRACE) are described in later sections. The commands in this section are options that must be turned on or off before raytracing is initiated.

```
--- ORDER FORWARD
      ALL
      SEQUential
```

Determines the way that rays will be traced through the system. The objects that the program will consider as possible nearest intersections is controlled by the second literal entry. Use "ALL", if all objects in the system are to be considered (the default). Use "SEQUential", if only the next highest object number is to be considered. And use "FORWARD", if all objects with object numbers greater than or equal to the last number intersected are to be considered.

```
--- HALT n [ k ]
```

Forces the tracing of certain rays to be halted. "n" is the maximum number of times a ray can consecutively interact with the same object. "k" is a direction specified by 1=x, -1=-x, 2=y, etc. If a ray attains a component in this direction after reflection or refraction or the number of consecutive interactions with the same object reaches "n", then tracing halts for that particular ray.

--- DIGIts n

An iterative technique is employed by the program to find the intersection with an object surface of order 3 or more. The iterative process stops when the solution has at least "n" significant digits of accuracy. The default at start of program execution is 5. This is more than sufficient for non-imaging systems. However for high resolution imaging systems, a greater accuracy should be used but should not exceed the precision of the machine.

--- FRESnel [OFF]

Signals the program to calculate the unpolarized Fresnel losses at all refractive interfaces and reduce the ray flux accordingly. Use the "OFF" entry to turn this option off (the default). The calculation will only be done at interfaces at which both the refractive index and absorption change.

--- ERRORs [n LIST]

Controls output of ray error messages. If more than "n" ray errors occur, program execution is terminated. If the "LIST" entry is present, then all error messages are printed.

--- DRAW k k'

The projection of each ray in the directions given by the literal entries "k k' " is plotted on a penplotter. For example, if the literals are "X Y", then the x and y components are plotted on the paper. The actual plot commands are written to FORTRAN unit 20 using the standard CALCOMP format. These commands can then be read and sent to a penplotter or accessed by a program that does the plotting on a Texas Instruments 820 KSR terminal with the DFC (Device Forms Control) option and Configuration Code 81 enabled. For the latter the required monitor commands are:

```
$ SET TER / WIDTH=218  
$ RUN LINPLT
```

This program will then prompt the user for a scaling factor which is the number of inches on the paper per unit system length.

```
--- RANDom [ r j j' j" .. ]
```

Assigns an RMS angular error "r" in degrees (default=0) to the surfaces of the objects j, j', j", etc. If the object numbers are not given then all objects are assigned the same random surface error.

B.17.2.3.2 Single Ray Trace -

```
--- RAY i  
    x y z a b c  
    x y z X a a'  
        Y  
        Z
```

Traces either the "i"th previously defined ray or a single ray through the entire system starting in the first medium at the point (x,y,z) and with a direction given by the unit vector (a,b,c) or zenith and azimuth angles (in degrees) a and a'. The program always prints out the ray data at each surface intersected.

B.17.2.3.3 Tracing A Group Of Rays -

A number of rays may be traced at one time. The procedure involves initializing the ray data (coordinates and directions) at a plane in the system which the program designates as object zero. The commands RAYS and GRID set the coordinates while the SOURCE command then sets the directions and starting optical path lengths. The FLUX command initializes the flux per ray. Actual execution of the ray tracing is accomplished with the TRACE command. Afterwards the new ray data may be analyzed using the STATS and SPOTS commands.

B.17.2.3.3.1 Initialization Of Ray Data -

```

--- RAYS OFF
      X x
--- y [ z f ]
      :

```

```

--- RAYS Y y
--- z [ x f ]
      :

```

```

--- RAYS Z z
      x [ y f ]
      :

```

The second and third entries on the RAYS command designate the plane from which the rays will originate (object zero). The ray coordinates in this plane are then entered on successive commands. "f" is the flux (energy per unit time) to be assigned to each corresponding ray (default=1). The OFF option resets the number of rays in storage to zero. Otherwise, the RAYS and GRID commands always add their rays to the ones already in storage.

```

--- GRID X x y y' z z' n n'
      Y y z z' x x'
      Z z x x' y y'

```

Generates a uniform rectangular grid of rays on the plane given by the second and third entries. The fourth and fifth entries determine the maximum extent of the grid in one direction and the sixth and seventh in the other direction on that plane. The number of rays in each semi-direction are given by the last two entries.

```

--- GRID X x r r' a a' n n' w [ k k' ... ]
      Y y
      Z z
      k d

```

Generates a circular grid of rays on the plane given by the second and

third entries when the extra entry "w" is specified. The plane is normal to one of the axes and can be specified explicitly or by an already defined object "k" (the "d" entry is an additional shift that can be given to objects plane surface). In the latter case the boundary surfaces of the object are also applied to the grid as the ray coordinates are being generated. The "r" and "a" entries determine the maximum extent of the grid on that plane in the radial and angular directions respectively. The number of rays in the radial direction is given by the "n" entry. "n" is the number of rays in the first non-zero radial "ring". The rays are distributed radially according to the formula $r(i)=r(n)*(i/n)**w$. The rays are distributed evenly in the angular direction with the number in each "ring" selected so that they all represent an equal area on the plane. Therefore, a value of 1 for "w" yields an equipolar distribution while a .5 corresponds to a standard straight spokes distribution. If "r" is equal to zero, the first

--- STORE [k]

Stores the present ray data onto FORTRAN unit "k" in sequential, unformatted, variable length records. If "k" is less than zero, then the unit is rewound first. The default for "k" is -20.

--- RECALL [k]

Recalls the ray data stored on FORTRAN unit "k". The default for "k" is 20.

--- FOCUS

Determines the 3-dimensional focal point (in a least squares sense) for the current ray bundle. Also calculates the RMS deviation of the rays from this point.

--- FLUX f

--- FLUX -j

--- FLUX LAST

Assigns the same flux "f" to every ray. The "-j" entry changes the sign of the fluxes of any rays that are not on object number "j". Rays with fluxes less than or equal to zero are ignored by the program in any calculation.

The LAST entry replaces the fluxes by the present object number of each ray.

```
--- SOURCE POINT x y z [ m ]
```

```
--- SOURCE DIREction a b c [ m ]
```

```
--- SOURCE DIREction X,Y,Z a a' [ m ]
```

Used after the GRID and RAYS commands to initialize the directions and optical path lengths of the rays. (x,y,z) is the point from which the rays appear to emanate. The unit vector (a,b,c) or zenith and azimuth angles (a,a') measured in degrees from the given axis specify a direction assigned to all the rays, i.e. a collimated beam. The optical path length of the first ray (the principal ray) is always chosen to be zero. "m" is the number of the medium in which the rays originate (default=1).

```
--- REVERse
```

Reverses the directions of all the rays presently in storage. Useful for doing "virtual" ray traces.

B.17.2.3.3.2 Execution Of Trace -

```
--- TRACE [ j j' n LIST ]
```

Traces the first n rays in storage through the system starting at object "j" and stopping at object "j'" (the defaults are all the rays from object zero to the highest defined object respectively). The signs of j and j' determine if a refraction or reflection calculation will be done at the starting and stopping object surfaces. In particular, a positive j and negative j' result in no such calculation. The LIST option should be used if object by object ray data printout is desired. Otherwise, no output is generated except for possibly error messages.

B.17.2.3.3.3 Analyzing The Final Ray Data -

```
--- STATS POSITioN [ j 1 ]
```

```
--- STATS DIRECtioN [ j 1 ]
```

Produces a statistical analysis of the positional or directional ray data including total flux, centroids, full variances (RMS deviations), and maximum spreads in each coordinate direction. "j" is the number of the object for which the analysis will be done. If this number is less than or equal to zero all objects will be analyzed. The "1" entry is the number of the ray to which all ray data is to be referenced and is used to produce an additional printout of the individual ray data for every ray. If it is absent or zero, then no listing is generated. If it is negative, the ray data is listed in absolute form.

```
--- SPOTS POSITioN X Y,Z [ u n OFF ]  
                  Y Z,X  
                  Z X,Y
```

```
--- SPOTS DIRECtioN X Y,Z [ u n OFF ]  
                  Y Z,X  
                  Z X,Y
```

Produces a "spot diagram" of the ray data specified in type by the second entry and in coordinate direction by the third and fourth entries. The flux data for each pixel on the page is written to FORTRAN unit "u" in unformatted direct access records whose length is the same as the page width (see PAGE command). If "u" is negative, then any previous data on the specified unit is overwritten. Otherwise, the flux data is added pixel by pixel to what's already in the file. The default value of "u" is -19. The ASCII equivalent to "n" can be plotted at the spot where the ray strikes instead of the flux sum of all rays in the pixel. No diagram is plotted (although pixel flux accumulation is still done) if the OFF entry is employed. The short form of the command plots the data already on file "u" regardless of what type it is. However, the record length (i.e. the page width) must be the same as it was when the data was written on the file.

```
--- WINDOW a a' d d'
```

Used before the SPOTS command to set the data area to be plotted as "a" to "a'" across the "d" to "d'" down the page. If either a=a' or d=d', then the

SPOTS command autoscales the particular direction. Otherwise, this command is used to override any autoscaling.

```
--- BILATeral POSITion X [ Y,Z ]  
                        Y [ X,Z ]  
                        Z [ Y,X ]
```

```
--- BILATeral DIREction X [ Y,Z ]  
                        Y [ X,Z ]  
                        Z [ Y,X ]
```

In systems with bilateral symmetry the above commands can be used to reduce the number of rays traced and still produce meaningful plots from the SPOTS command. Any ray, whose data specified by the second and third is negative, is mirror imaged about the zero coordinate plane so that the particular data becomes positive. For systems with symmetrical quadrants, two coordinate directions should be entered.

```
--- PLOT3d [ u ]
```

Produces a pen plot of the 3-dimensional projection of the data created by a SPOTS command on FORTRAN unit "u" (default=19). The pen movement commands are written onto FORTRAN unit 20 and can be either sent to a CALCOMP type plotter or to a Texas Instruments 820 KSR terminal via the LINPLT program (see DRAW command).

INDEX

a,b,c, 3-12
ABEL, 3-189
ABERRATIONS, 3-66
\$ABORT, 2-37
ACCURACY, 3-128
AFOCAL, 3-62
ALIGN, 3-16
ALLOWED, 3-129
ALTER, 3-29, 3-58, 3-64
ANGLES, 3-189
AP2RBE, 4-1
AP2RBS, 4-1
...aperture, 3-40
APODIZE, 3-142
APODIZE ANGLES, 3-143
APODIZE BOTH, 3-144
APPEND, 3-194
ARC, 3-24
\$ARGS, 2-15
Arithmetic, 2-4
ARRAY, 3-53
ARROWS, 3-118
ASCALE, 3-28
\$ASK, 2-24
ASYM, 3-51
AVERAGE, 3-187
AXICONIC, 3-46
AXIS, 3-132

BEAMS, 3-138
\$BEEP, 2-32
BEND, 3-53
BEZIER, 3-26
BICONIC, 3-44
BILATERAL, 3-163
Binary file, 3-197
BMP2DIS, 4-4
BOUNDS, 3-101, 3-140

BRANCH, 3-109
BSDFDATA, 3-89
BSDFDATA anisotropic, 3-94

CADEXPORT, 3-113
CARTOVAL, 3-47
\$CASE, 2-14
CHARACTER, 3-24
\$CLEAR, 2-29
...CLIP, 3-177
CLIP, 3-139
COARSEN, 3-30
COATING LAYERS, 3-77
COATING MODELS, 3-76
COATING PROPERTIES, 3-75
COLLECTION, 3-175
...COLORS, 3-126
COLORS, 3-118
COMBINE, 3-193
COMPOSITE, 3-27, 3-60
CONDUIT, 3-50
CONIC, 3-25
CONSIDER, 3-114
CONTOUR, 3-192
\$COPY, 2-17
CORNER, 3-40
CURVES, 3-18
CUTOFF, 3-129

data, 3-79
\$DATIM, 2-32
\$DBG, 2-27
DECOMPOSE, 3-158
DECOMPOSE DIREC, 3-159
DECOMPOSE POSIT, 3-159
DEFORM, 3-99
DIMENSION, 3-112
DIRECTIONAL, 3-192

DIS2BMP, 4-4	\$EXP, 2-14
\$DISP, 2-34	EXPLICIT, 3-58
DISPLAY, 3-185	EXPLODE, 3-109
DMAP, 3-191	EXTEND, 3-32
\$DO, 2-18	EXTREME, 3-168
DOMACROS, 3-110	
DOME, 3-61	FACETS, 3-97
DOUBLET, 3-61	\$FAST, 2-26
DRAWING, 3-122	\$FCN, 2-28
DUMP, 3-177	FCN, 3-51
DXF2ASAP, 4-2	\$FF, 2-23
	FFAD, 3-171
\$ECHO, 2-14	FFT, 3-186
EDGES, 3-18	FIELD, 3-180
\$EDIT, 2-33	FIELDBPM, 3-182
ELLIPSE, 3-21	FIELDSUM, 3-181
ELLIPSOID, 3-36	FITTED, 3-36 to 3-37
\$ELSE, 2-22	FLUX, 3-162
\$ELSEIF, 2-22	FMAP, 3-55
EMIT BOX, 3-154	FOCUS, 3-170
EMIT CONE, 3-154	FOLD, 3-189
EMIT DATA, 3-156	FORM, 3-186
EMIT DISK, 3-152	FRESNEL, 3-105, 3-130
EMIT ENTITY, 3-152	FTSIZE, 3-110
EMIT FILAMENT, 3-155	FUNCTIONS, 3-33
EMIT HELIX, 3-154	
EMIT IES, 3-155	GAUSSIAN, 3-158
EMIT OBJECT, 3-152	GENERAL COEFFICIENTS, 3-33
EMIT PYRAMID, 3-154	GET, 3-172
EMIT RAYS, 3-156	\$GO, 2-20
EMIT RECT, 3-152	\$GRAB, 2-29
EMIT SPHEROID, 3-154	GRAPH, 3-191
Emitters, 3-152	GRID DATA, 3-150
EMITTING, 3-157	GRID ELLIPTIC, 3-148
ENCLOSED, 3-192	GRID HEX, 3-149
END, 3-111	GRID OBJECT, 3-149
\$END IF, 2-22	GRID POLAR, 3-148
ENT OBJECTS, 3-108	GRID RECT, 3-148
ENTITIES, 3-108	GRID WINDOW, 3-150
\$ERR, 2-20	GROUP, 3-108
\$EVAL, 2-31	\$GUI, 2-36
\$EXIT, 2-23	

HALT, 3-107, 3-129	MEDIA, 3-72
HARVEY, 3-80	MEDIA ABSORB, 3-73
HARVEY anisotropic, 3-91	MEDIA CRYSTAL, 3-74
HEADER, 3-190	MEDIA GRIN, 3-73
HELIX, 3-27	MEDIA SCATTER, 3-74
\$HELP, 2-23	MEDIA USER, 3-74
HISTOGRAM, 3-193	\$MENU, 2-17
HISTORY, 3-170	MESH, 3-193
HORN, 3-41	MINIMIZE, 3-69
	...MINMAX, 3-79
IDEAL, 3-63	MIRROR, 3-60
IESFILE, 3-194	MISSED, 3-132
\$IF, 2-21	MODELS, 3-78
\$IF...THEN, 2-22	MODIFY, 3-188
IGS2ASAP, 4-2	\$MOVE, 2-17
IMAGE, 3-30	MOVE, 3-164
IMMERSE, 3-136	MULTIPLE, 3-52
INTERF DIFFRACT, 3-102	
INTERF REPEAT, 3-103	\$NEXT, 2-19
INTERFACE, 3-102	NONLIN FIT, 3-84
INVERT, 3-29	NONLIN FIT anisotropic, 3-93
\$IO, 2-15	NONLINEAR, 3-83
IRRADIANCE, 3-178	NONLINEAR anisotropic, 3-92
ISOMETRIC, 3-191	NORMALIZE, 3-186
\$ITER, 2-29	NUMBERS, 3-112
LAMBERTIAN, 3-80	OBJECTS, 3-95
\$LEAVE, 2-19	OBLIQUE, 3-116
LENSES, 3-59	OFFSET, 3-189
LEVEL, 3-107, 3-131	OPDMAP, 3-176
LIGHTS, 3-123	OPTICAL, 3-43
LIMITS, 3-100	OVAL, 3-22
LINE, 3-19	...OVERLAY, 3-125
LIST, 3-167	
\$LOC, 2-35	\$PAGE, 2-34
LOCAL, 3-54	PARABASAL, 3-136
LSQFIT, 3-110	PARAMETERIZE, 3-57
	PARTICLES, 3-87
MANGIN, 3-61	PATCHES, 3-31
MAP, 3-124	\$PATH, 2-35
Math functions, 2-5	PATHS, 3-169
MATRIX, 3-16	PENTA, 3-62

PERFECT, 3-63
PHYSICAL, 3-86
PICTURE, 3-193
...PIXELS, 3-127
PIXELS, 3-115
PLACE, 3-14
PLANE, 3-40
PLANE NORMAL, 3-35
PLANE POINTS, 3-35
\$PLAY, 2-28
\$PLOT, 2-33
...PLOT, 3-78
PLOT, 3-119
PLOT BEAMS, 3-120
PLOT FACETS, 3-120
PLOT LIMITS, 3-120
PLOT MESHES, 3-121
PLOT POLARIZATION, 3-120
PLOT RAYS, 3-120
PLOT3D, 3-191
POINTS 2d, 3-23
POINTS 3d, 3-18
POLARIZ, 3-138
POLYNOM FIT, 3-82
POLYNOMIAL, 3-81
PRINT, 3-111
Prisms, 3-62
PROFILE, 3-119
PUT, 3-173

QPRINT, 3-111

RACETRACK, 3-23
RADIAL, 3-187
RADIANT, 3-174
\$RAN, 2-37
RANGE, 3-190
RAWDATA, 3-89
RAWDATA anisotropic, 3-94
RAY, 3-145
RAYS, 3-146
RAYSET, 3-147

RBOUNDS, 3-101, 3-140
\$READ, 2-15
RECTANGLE, 3-22
REDEFINE, 3-96
REDUCE, 3-188
\$REG, 2-24
RENDER, 3-123
RENORM, 3-56
REPEAT, 3-19, 3-35, 3-59
REPLICATE, 3-190
REPLOTT, 3-121
RESET, 3-110
RETURN, 3-111
REVERSE, 3-165
REVOLUTION, 3-45
RIGHT, 3-62
RMS, 3-86
ROTATE, 3-16
ROUGHNESS, 3-97
ROUNDED, 3-23

SAG, 3-28
SAMPLED, 3-48
SAMPLED cylinder, 3-49
SAVE, 3-133
SAWTOOTH, 3-25
SCALE, 3-15
SCALE FROM, 3-15
SCATTER, 3-103
SCATTER MODEL, 3-104
SCATTER REPEAT, 3-103
\$SCR, 2-25
SEARCH, 3-128
SECTION, 3-188
SEED, 3-133
SEGMENTS, 3-118
SELECT, 3-160
SEQUENCE, 3-59
SHAPE, 3-162
SHIFT, 3-14
\$SHOW, 2-34
SHOW, 3-112

SINGLET, 3-60	UNITS, 3-71
SKEW, 3-15	\$UNVAR, 2-36
SMOOTH, 3-29	UPDATE, 3-18
SOLID, 3-56	USERAPOD, 3-142
SOURCE, 3-150	USERAPOD ANGLES, 3-143
SOURCE WAVEFUNC, 3-151	USERAPOD BOTH, 3-144
SPECTRUM, 3-140	USERBSDF, 3-85
SPLINE 2d, 3-26	USERBSDF anisotropic, 3-93
SPLINE 3d, 3-20	USERCURVE, 3-20
SPLIT, 3-106, 3-131	USERFUNC, 3-38
SPOTS, 3-173	USERSAG, 3-44
SPREAD, 3-179	USERSURF, 3-38
SPRINT, 3-111	UVSPACE, 3-31
STATS, 3-168	
STATS TRACE, 3-169	VALUES, 3-189
\$STO \$RCL, 2-26	VANES, 3-91
STORE, 3-70	VARIABLES, 3-68
SUBSET, 3-161	VCAVITY, 3-88
SUM, 3-95	VECT2DXF, 4-1
Summary, 3-2	\$VIEW, 2-34
SUPERCONIC, 3-50	VIEW, 3-117
SURFACES, 3-33	VIOLATION, 3-178
SWEEP, 3-30	VOLUME, 3-87
\$SYS, 2-24	VOXELS, 3-135
SYSTEM, 3-113	VUFACETS, 3-123
TABLE, 3-190	\$WAIT, 2-35
TELESCOPE, 3-61	WARNINGS, 3-132
TEST, 3-55	WAVELENGTH, 3-72
...TEXT, 3-126	WEDGE, 3-62
TEXTFILE, 3-195	WIDTHS, 3-137
THRESHOLD, 3-190	WINDOW, 3-116
\$TIC, 2-27	...window, 3-127
TITLE, 3-114	WRITE, 3-194
TORUS, 3-43	
TOWARDS, 3-104	
TRACE, 3-166	XEQ, 3-17
Transformations, 3-13	XMEMORY, 3-145
TRANSPOSE, 3-187	
TREE, 3-112	
TUBE, 3-42	ZERNIKE, 3-47