
PLOT3D User's Manual

Pamela P. Walatka, Pieter G. Buning, Larry Pierce,
and Patricia A. Elson

March 1990

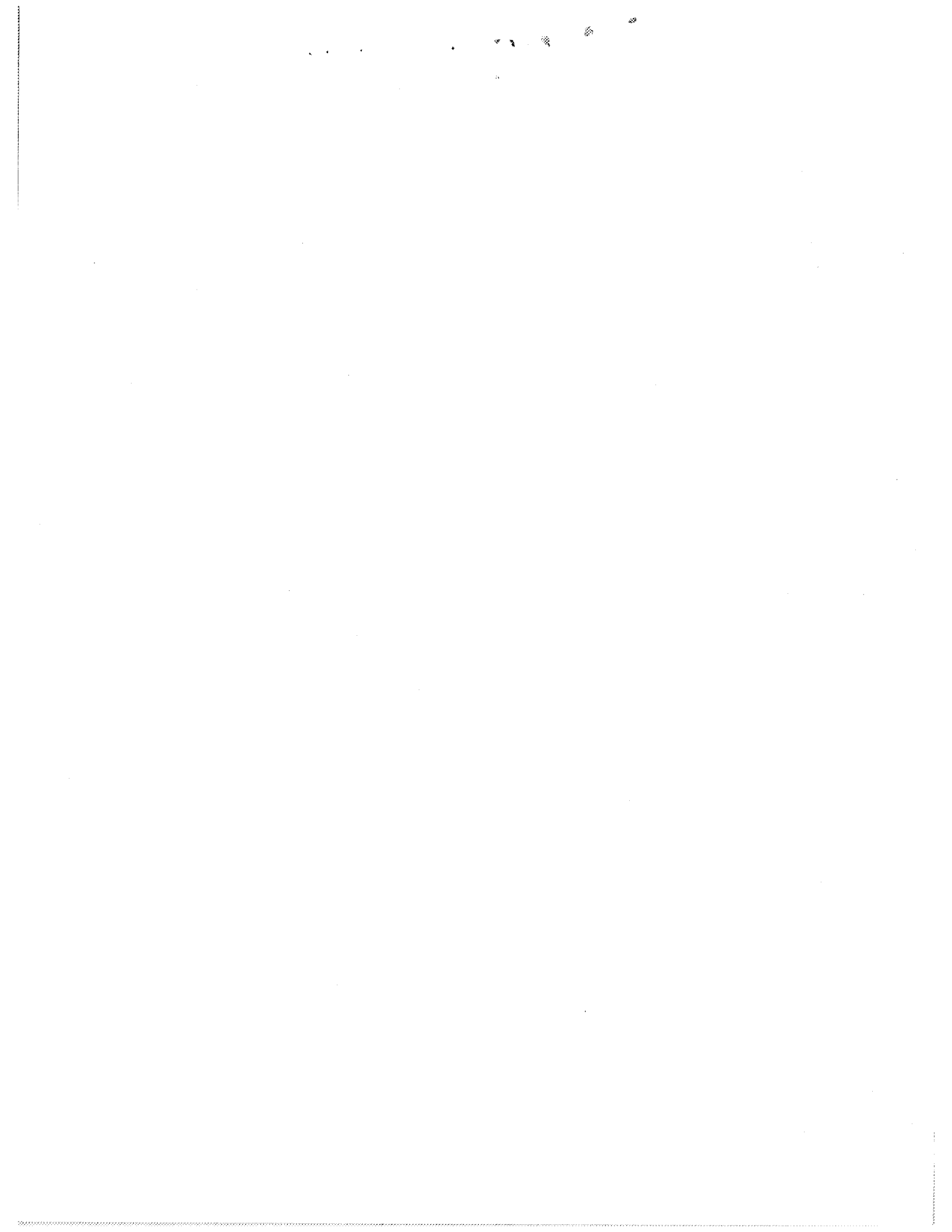
(NASA-TM-101067) PLOT3D USER'S MANUAL
(NASA) ~~3007~~ CSCL 09B

N90-23090

308p

238108

[Handwritten signature]
Unclas
G3/61 0279303



**ORIGINAL CONTAINS
COLOR ILLUSTRATIONS**

PLOT3D User's Manual

Pamela P. Walatka, Computer Sciences Corporation, Moffett Field, California
Pieter G. Buning, Ames Research Center, Moffett Field, California
Larry Pierce and Patricia A. Elson, Sterling Federal Systems, Moffett Field, California

March 1990



National Aeronautics and
Space Administration

Ames Research Center
Moffett Field, California 94035

INSTANTANEOUS TEMPERATURE CONTOURS IN 2.5 STAGE COMPRESSOR

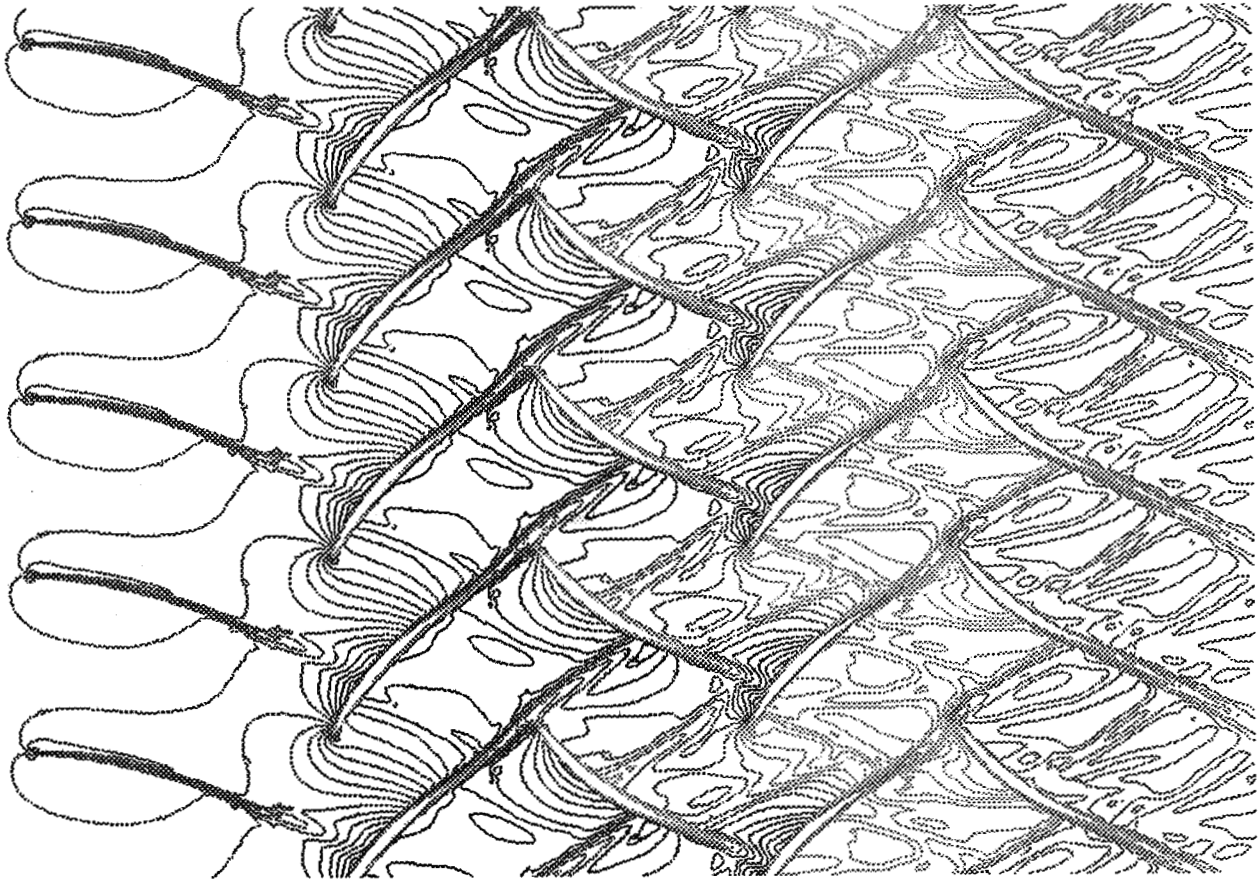


Figure 1. Temperature contours in 2.5 stage compressor. PLOT3D was used to visualize pressure solutions on the model of five rotor stator blades; then GAS was used to copy and repeat the image and to add the title. PIXPS was used to create a black and white PostScript file (PLOT3D also makes PostScript files). Computations by Karen Gundy-Burlet. Graphics by Guy Haskin and Paul Kelaita.

**ORIGINAL PAGE IS
OF POOR QUALITY**

Foreword

The PLOT3D software program was developed at NASA Ames Research Center by Pieter Buning. He released the first code in 1982. Version 3.6 was released in February 1989, and 3.6b, for IRIS workstations, in December 1989.

The PLOT3D USER'S MANUAL and the PLOT3D software are distributed without copyright to any institution or business within the United States. The software is distributed for the use of the recipient only and is not to be redistributed.

- To obtain copies of the code, contact:

Larry Pierce
M/S T045-2
NASA Ames Research Center
Moffett Field, CA 94035-1000

(415) 604-4492
FTS 464-4492
pierce%prandtl@ames.arc.nasa.gov

- For help with PLOT3D problems, call Larry Pierce.

- For manuals only, contact:

NAS Documentation Center
M/S 258-6
NASA Ames Research Center
Moffett Field, CA 94035-1000

(415) 604-4632
(FTS) 464-4632
doc-center@prandtl.nas.nasa.gov

Most of the other software described in this manual is available from COSMIC (404) 524-3265.

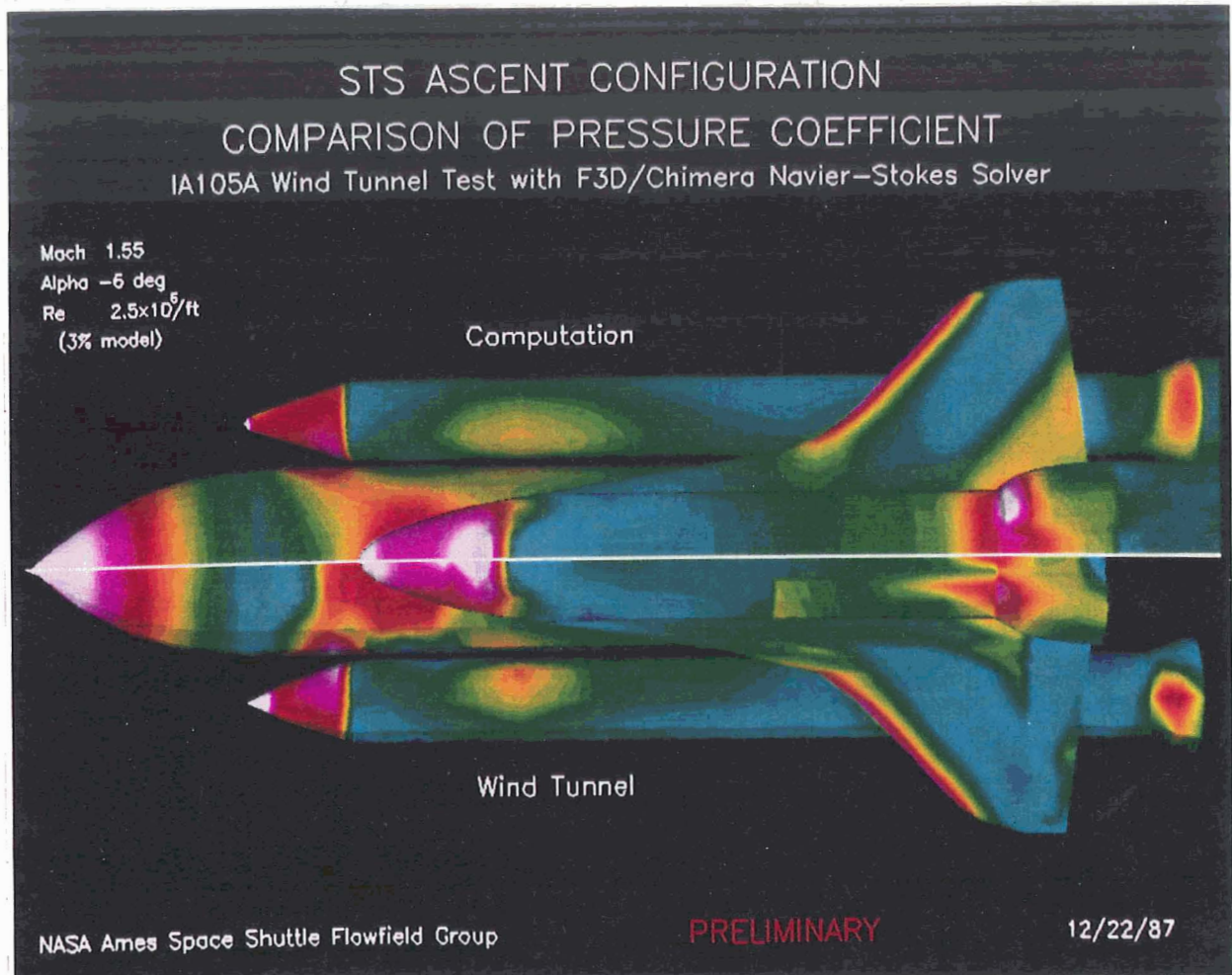


Figure 2. Color contours on the surface of the Space Shuttle. The colors indicate pressure coefficient. Wind tunnel results are compared to computational results, using PLOT3D to read and display the grids and fluid dynamics solutions, then using GAS to combine and title the parts. See Chapter 11.3 for details. Computations and graphics by Pieter Buning, Ing-Tsau Chiu, Yehia Rizk, and Joe Steger.

ORIGINAL PAGE
COLOR PHOTOGRAPH



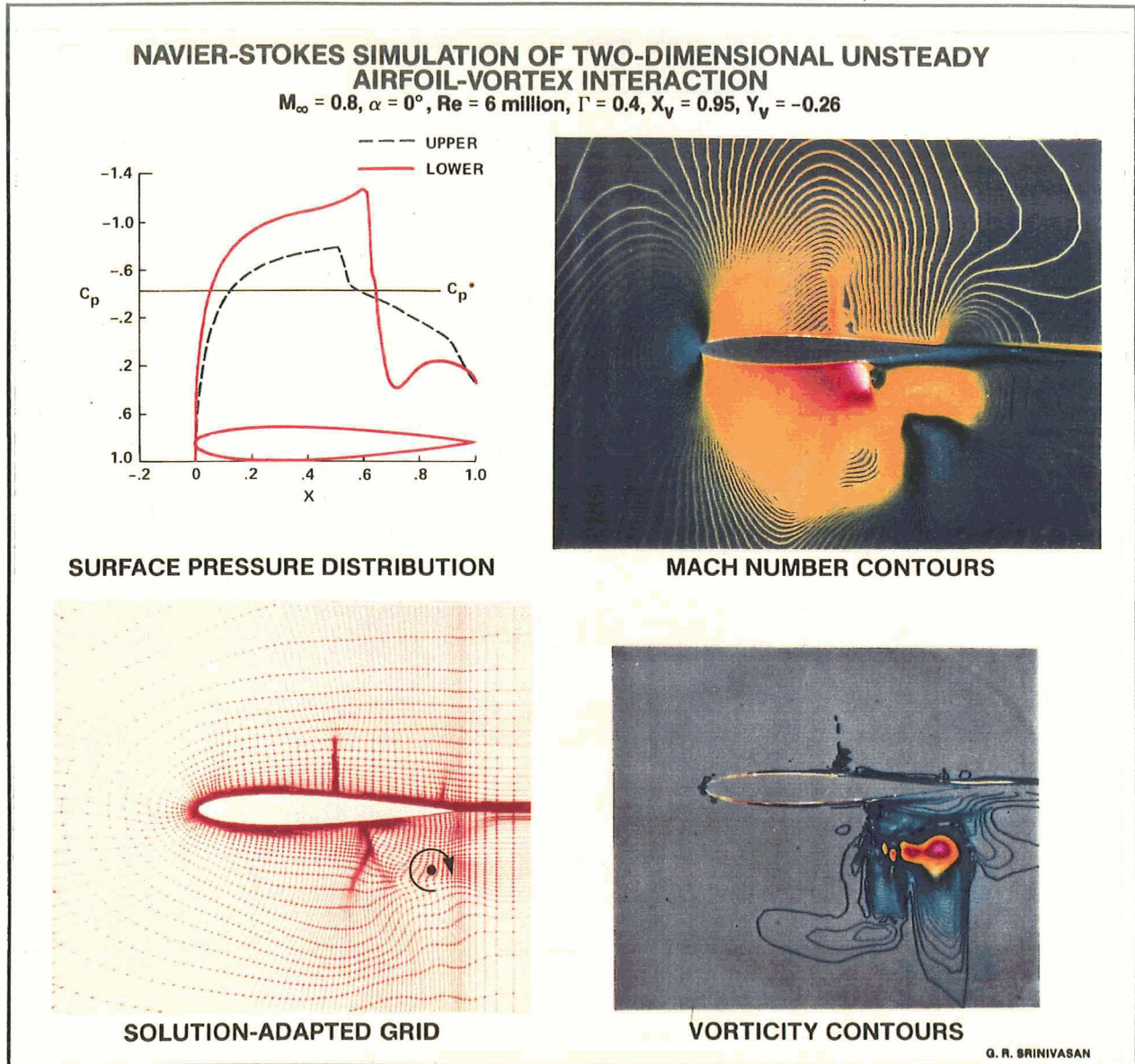


Figure 3. Four plots created with PLOT3D and then combined with GAS. The plots are from a two-dimensional data set. SOLUTION-ADAPTED GRID shows the grid upon which the solutions are based. SURFACE PRESSURE DISTRIBUTION, created with an earlier version of PLOT3D and enhanced with GAS, is a function surface plot, showing a graph of the pressure coefficients on both surfaces of the airfoil. The others are contour plots, with color indicating variations of Mach number and vorticity magnitude. Computations and graphics by G. R. Srinivasan.

ORIGINAL PAGE
 COLOR PHOTOGRAPH

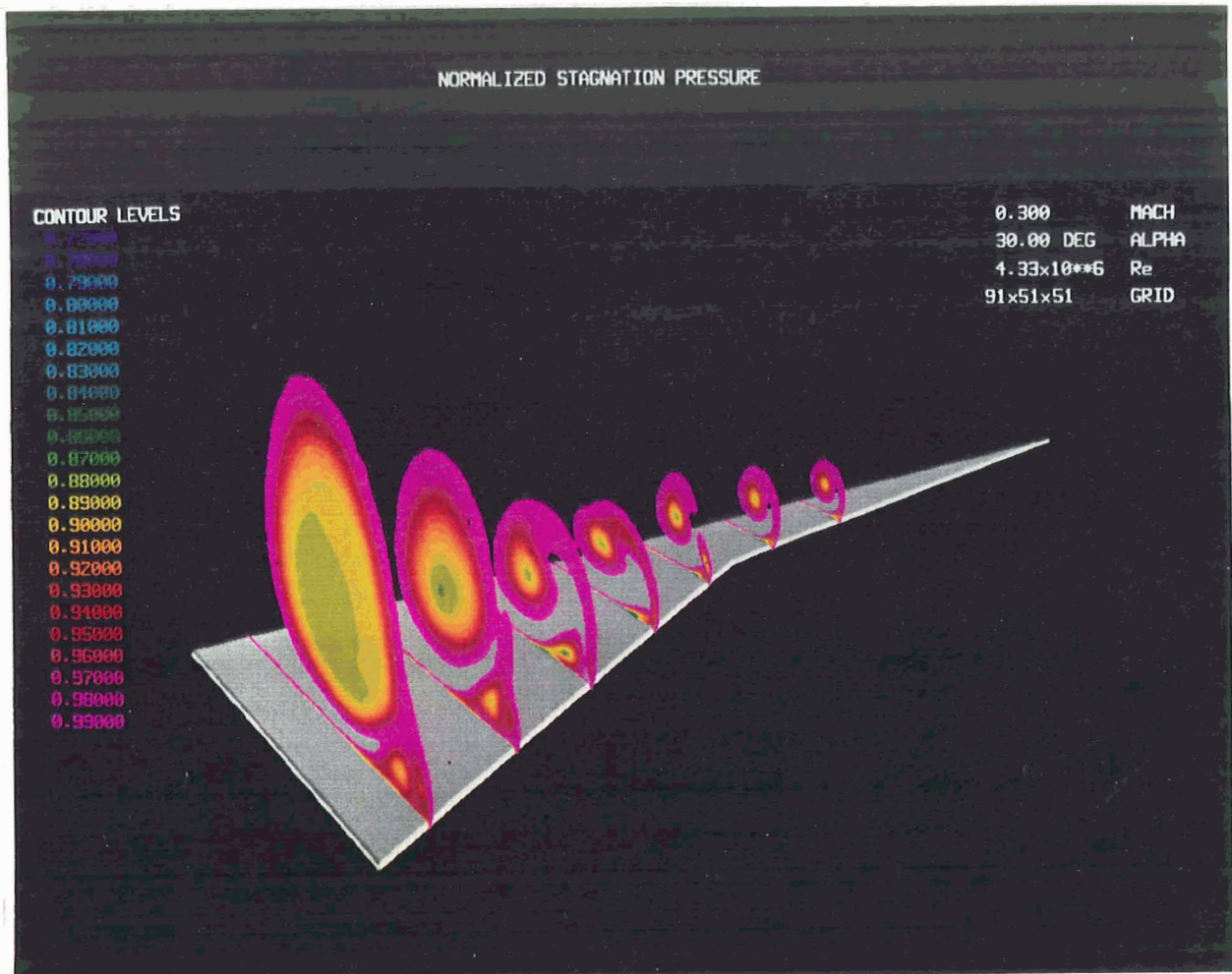


Figure 4 Color contours on subset slices. This is a photograph of a PLOT3D screen. The colors in the subset slices indicate variations in stagnation pressure. The legend at left shows the values represented by the colors. Computations by Ko Fujii. Graphics by Ko Fujii and Pieter Buning.

ORIGINAL PAGE
 COLOR PHOTOGRAPH

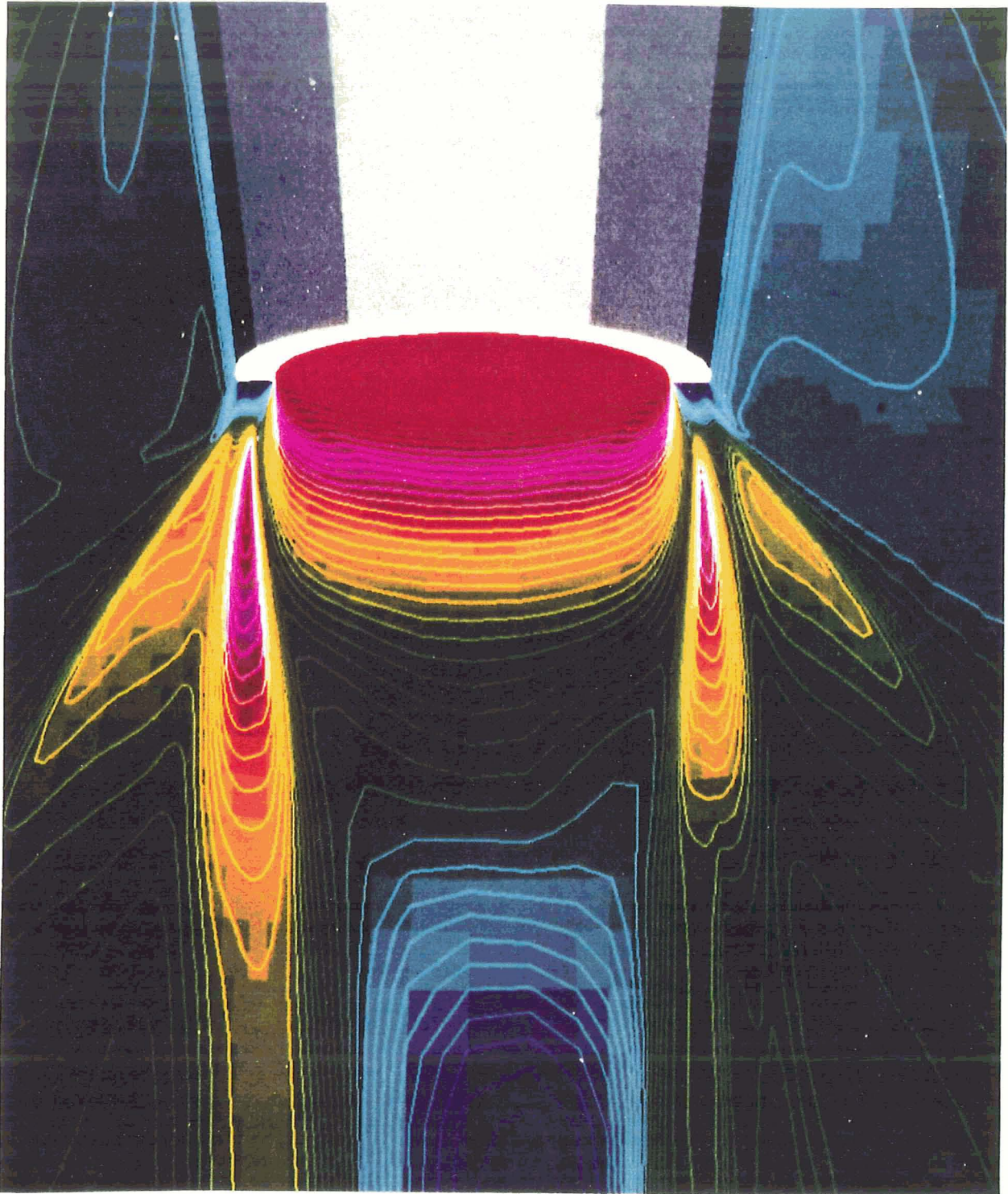


Figure 5. Flow past conical afterbody. Density is shown with lines over color contours. The /MANUAL qualifier to the CONTOURS command was used. See Chapter 4.7. Computations by Steve Deiwert and Herb Rothmund. Graphics by Pieter Buning, Adrain Borja, and Steve Deiwert.

ORIGINAL PAGE
COLOR PHOTOGRAPH

xi

Acknowledgements

The manual was reviewed by Vilia Collet, Charles Dickens, Sandy Emerson, Christine Gong, Zack Jalamani, Robert Meakin, Jacques Middlecoff, Merritt Smith, and David Yip. Mike Bridges helped with the PostScript figures. Todd Plessel, Scott Thomas and Val Watson provided guidance.

Thank you.

P.W.

PLOT3D User's Manual

Table of Contents

Foreword.....	iii
Color plates	v
Acknowledgements	xii
Table of Contents.....	xv
List of Figures	xx
Symbols.....	xxii
Summary.....	xxiii
Chapter 1. Introduction	1
1.1 Summary	1
1.2 Scientific Visualization at NASA Ames.....	1
1.3 PLOT3D Features	4
1.4 Host Systems	5
TUTORIAL.....	6
2.1 Summary	7
2.2 Introduction	7
2.3 Key to Tutorial Symbols and Styles.....	7
2.4 Getting Wingbodytail on an Ames IRIS	8
2.5 Getting Wingbodytail at Ames in VMS	8
2.6 Getting Wingbodytail Off-site.....	9
2.7 Demonstration Testdrive.....	9
2.8 Geometry	12
2.9 How To Use the Mouse	13
2.10 Return to Command File — <return> or <delete>	15
2.11 Example of a Surface Plot.....	15
2.12 Mach Contours on Slices	16
2.13 Mach 1	17
2.14 Contour Plot of Pressure on Surface.....	18
2.15 Particle Traces.....	19
2.16 End of the Testdrive.....	20
2.17 Command Basics	20
2.18 File Names and Extensions.....	22
2.19 On-line Help.....	23
2.20 Exit — exit or exit/save	26
Chapter 3. Loading and Displaying Data	27
3.1 Summary	27
3.2 How To Do It.....	27
3.3 Data In Current Directory.....	27
3.4 Loading Data — read/x=gridfile/q=solutionfile.....	29

Table of Contents, Continued

3.5	Text — text.....	31
3.6	Defining Geometry	32
3.7	Defining Plot Range — minmax.....	41
3.8	Plotting Viewpoint — vpoint/angle.....	42
3.9	Grid Chart.....	43
3.10	Plot — plot.....	46
3.11	Return to PLOT3D Prompt — ↵ or 	47
3.12	Exit — exit or exit/save.....	48
3.13	Conclusion of Chapter 3.....	48
Chapter 4.	Functions.....	49
4.1	Summary	49
4.2	Types of Functions	49
4.3	Grid Functions — Walls, Subsets, IBLANK Holes, and Grid checks.....	50
4.4	Simple Walls.....	58
4.5	Scalar Functions	60
4.6	Comparison of Contour and Function Surface or Carpet Plots	61
4.7	Scalar PLOT/CONTOUR.....	63
4.8	Examples of Contour Plots.....	64
4.9	Scalar plot/Fsurface	71
4.10	Vector Functions.....	75
4.11	Particle Traces.....	76
4.12	Shock Wave.....	82
4.13	References to Other Chapters	83
4.14	Functions Conclusion.....	84
Chapter 5.	Saving Data.....	85
5.1	Summary	85
5.2	Saving Keyboard Work.....	85
5.3	Command (Script/Batch) Files.....	86
5.4	Initialization File — plot3dini.com.....	87
5.5	PLOT/SCRIPT.....	88
5.6	Saving Particle Traces.....	88
5.8	Converting .com files to .gra for GAS.....	89
5.9	LISTing Function Data.....	90
5.10	Conclusion	91

Table of Contents, Continued

6.1	Summary	92
6.2	IRIS Screen Dumps to Files.....	92
6.3	Screen Dumps at NAS — Printers and Cameras	93
6.4	Videos at NAS	94
6.5	DISSPLA Version and IRISRGB Screen Dumps	94
6.6	Device Independent Plot — .dip Files	94
6.7	Conclusion	94
	REFERENCE.....	99
	Chapter 7. Keywords / Help File.....	101
7.1	Summary	101
7.2	Information.....	101
	General information.....	101
	Journal file	102
	Initialization file.....	102
	Installation of PLOT3D.....	103
	Release notes.....	104
	Known problems.....	111
	Special characters.....	115
	Blanking.....	115
	Attributes	116
	@file	119
	End-of-file.....	120
	\$system-command	120
7.3	Commands	121
	CLEAR.....	121
	CONTOURS.....	122
	EXIT.....	123
	FSURFACE	124
	FUNCTION	126
	Function numbers	126
	Nondimensionalizations	129
	Fluid constants.....	129
	Function definitions	129
	HELP	136
	LIST	137
	MAP	138
	MINMAX.....	139
	PLOT	140
	QUIT.....	149

Table of Contents, Continued

RAKES.....	150
READ.....	153
SHOW.....	155
SUBSETS.....	156
TEXT.....	157
VECTORS.....	158
VIEW.....	159
VPOINT.....	160
WALLS.....	161
Chapter 8. Data File Formats	162
8.1 Introduction	162
8.2 Grid XYZ Files.....	162
xyz file single grid.....	162
xyz file with IBLANK.....	163
xyz file multiple grid	163
8.3 Solution Q Files.....	164
Q file.....	164
Q file single grid (without Jacobian)	165
Q file multiple grid (without Jacobian).....	165
8.4 Function Files.....	166
Function_file single_grid.....	166
Function_file multiple_grid	167
Chapter 9. Changing PLOT3D	168
9.1 Summary	168
9.2 Getting More Space on Host System.....	168
9.3 Code Organization.....	169
9.4 Changing PLOT3D to Use Less Space	172
9.5 Different Form of Input Data.....	172
9.6 Different Operating System.....	176
9.7 Different Graphics Library	176
Chapter 10. Error Messages and Warnings	177
10.1 Organization.....	177
10.2 Error Messages.....	177
10.3 Warning Messages	206
Chapter 11. Sample Command Files.....	219
11.1 Summary	219
11.2 2D Pressure Coefficient Using TWING (plot/line).....	220
11.3 3D Comparison of Computational and Experimental Data, Space Shuttle.....	221

Table of Contents, Continued

11.4 CGL—Running PLOT3D on the Cray.....	227
11.5 Limiting Contour Lines to Mach 1.....	229
Chapter 12. Conclusion	232
12.1 Summary	232
12.2 Filename Extension Chart	232
12.3 Typical Command Sequence	234
Appendix A. Version 3.6	235
Appendix B. New Tools	250
Appendix C. PLOT3D Without an IRIS...plot3ddi.....	253
Appendix D. Incompressible Fluids	255
Appendix E. IRIS Differences.....	256
Appendix F. Directory Structures and Wingbodytail	258
Glossary.....	261
INDEX	271

List of Figures

1. Temperature contours. *ii*
2. Color contours on the surface of the shuttle. *v*
3. Four plots created with PLOT3D. *vii*
4. Color contours on subset slices. *ix*
5. Flow past conical afterbody. *xi*
6. Contour plot of pressure coefficients around an airfoil. *xxiv*
7. Numerical Aerodynamic Simulation software flow. 3
8. Key to tutorial symbols. 7
9. nfsmount remote workstation. 8
10. Change directory to wingbodytail on a workstation. 8
11. Getting the wingbodytail directory into your VMS directory. 9
12. Starting the testdrive. 11
13. First plot in the tutorial. 12
14. Mouse buttons. 13
15. Example of fsurface plot of pressure coefficient. 15
16. Mach contours on slice subset of wing. 16
17. Mach 1. 17
18. Color contour plot of pressure. 18
19. Particle traces, wingbodytail. 19
20. Another plot of particle traces. 20
21. Command basics. 21
22. Help . 24
23. Help subtopics. 25
24. Exit from PLOT3D. 26
25. nfsumount. 26
26. Starting PLOT3D; the READ command. 29
27. Text. 31
28. WALL command. 32
29. Wall 2, grid 2. 34
30. Walls 3 and 4, grid 2, line and shaded surface. 35
31. Walls 1 and 2, grid 3. 37
32. More walls - shaded surface, grid 3. 38
33. Walls 1 and 2, grid 4. 39
34. Walls 3 and 4, grid 4. 40
35. MINMAX. 41
36. Minmax box. 41
37. VPOINT/ANGLE. 42
38. Blank grid chart. 44
39. Tutorial grid chart . 45
40. PLOT. 46
41. Wingbodytail geometry. 47
42. EXIT. 48
43. Walls for grid function 1. 51
44. First and last walls of grid 2. 52
45. Optional experiment. 52
46. Walls and subsets comparison chart. 53
47. Eliminating subsets. 54
48. Subset = all of grid three. 54
49. More optional experiments. 55
50. Subsets for function 2, IBLANK holes. 56
51. Plot of function 2 showing IBLANKing holes. 57
52. Simple walls for grid 2. 58
53. Outline of the wing. 59
54. Outline of the tail. 60
55. Scalar function possibilities chart. 62
56. Subsets for slices on the wing. 65
57. FUNCTION, CONTOURS and SHOW commands. 66
58. function 115. 67
59. Example of stagnation pressure plotted with color contours on slices. 68
60. Subsets for symmetry plane. 69
61. Mach number, function 154. 70

List of Figures Continued

- 62. Plot of Mach number. 70
- 63. Subsets for rings around the surface
of wing and tail. 72
- 64. PLOT/FSURFACE and
FSURFACE. 73
- 65. 3D function surface plot. 74
- 66. Commands for vector plot. 75
- 67. VECTOR command for attributes.
76
- 68. Subsets set to everything for
particle traces. 77
- 69. More subsets, and RAKE
command. 79
- 70. Particle trace plot. 80
- 71. Oil flow particle trace. 81
- 72. Shock function specification. 82
- 73. Shock function plot. 83
- 74. Exit/save to create .jou file. 86
- 75. Running a command file. 87
- 76. Using PLOT3X to create a .gra file
for GAS. 90
- 77. Filename extensions involved in
saving data. 91
- 78. File types for making stills. 95
- 79. Extra Reference card, Functions.
97
- 80. Extra Reference card, Commands.
98
- 81. Line plot of 2D pressure
coefficient, DISSPLA version of
PLOT3D. 220
- 82. Distributed processing example:
color_contours. 227
- 83. Filename extension chart,
summary. 233
- 84. Right-handed coordinate system.
262

Symbols

boldface type	tutorial user input	subscripts:
c	speed of sound	x before shock
Cp	pressure coefficient	y after shock
Cp*	critical pressure coefficient	0 stagnation
(D)	default	inf free stream
e	energy per unit mass	i internal
grad	gradient	k kinetic
h	enthalpy per unit mass	
I	vector indicating grid segment	
J	vector indicating grid segment	
K	vector indicating grid segment	
M	Mach number	
p	pressure	
pp	pitot pressure	
Q1	first variable in the solution (q) file; density	
Q2	second variable in the solution (q) file; x-momentum	
Q3	third variable in the solution (q) file; y-momentum	
Q4	fourth variable in the solution (q) file; z-momentum (not present in 2D)	
Q5	fifth variable in the solution (q) file; stagnation energy per unit volume	
Re	Reynolds Number	
s	entropy	
T	temperature	
u	Cartesian component of velocity	
V	velocity	
v	volume, or Cartesian component of velocity	
w	Cartesian component of velocity	
↵	return	
gamma	ratio of specific heats	
ζ zeta	a curvilinear coordinate direction	
η eta	a curvilinear coordinate direction	
ξ xi	a curvilinear coordinate direction	
rho	density	
omega	vorticity	

Summary

PLOT3D is a computer graphics program designed to visualize the grids and solutions of Computational Fluid Dynamics (CFD). Versions are written for many systems. Seventy-four functions are available. PLOT3D can read multiple grids with almost any number of grid points, and renders models as wireframe, flat shaded, or color-mapped. Shading and perspective are used to show depth. Output from PLOT3D is displayed on the screen and (on the IRIS) can be moved with the mouse. Files can be created for scripting PLOT3D, or for transferring output to GAS (for animation), or for driving various printers and plotters. The graphics laboratory in the Numerical Aerodynamic Simulation facility at NASA Ames is equipped to make negatives, prints, viewgraphs, movies.

The first part of this manual (Chapters 2 - 6) is a tutorial that takes the reader, keystroke by keystroke, through a sample PLOT3D session. The reader runs a scripted demonstration using a data base called *wingbodytail*. Then, still using *wingbodytail*, the reader creates plots, making all the entries necessary to plot geometry and functions, and to save data.

The second part of the manual (Chapters 7 - 11) contains reference chapters, including the on-line help file, data file formats, advice on modifying the PLOT3D source code, and a listing, with explanations, of error and warning messages. Chapter 11 offers sample command files. Chapter 12 has a chart of the various file name extensions, and an example of typical PLOT3D command sequence. Appendix A describes new features of Version 3.6, many not listed elsewhere.

In brief, the basic instructions are as follows. From the system prompt, type **plot3d**, which will take you into PLOT3D if it is installed on your system. To load data, use the **read** command with appropriate qualifiers (Chapter 3 and listed under **READ** in Chapter 7). Use the **wall** command to define geometry. The region you will see is defined by **minmax**, and **vpoint** determines where you will see it from. Use **plot** to graphically display what you have. On an IRIS workstation, you can use the mouse to translate, zoom, or rotate. Press <return> or to get back to the keyboard.

Before plotting functions, use the **subset** command to specify the regions where the function will be calculated. Then use **function** to state the type of function you want. The **contours** command offers attribute and increment choices for plotting the function. Then type **plot/contour** for a contour plot, or **plot/fsurface** (version 3.6) or **plot/surface** (version 3.5) for a carpet plot. Other commands offer a variety of adjustments.

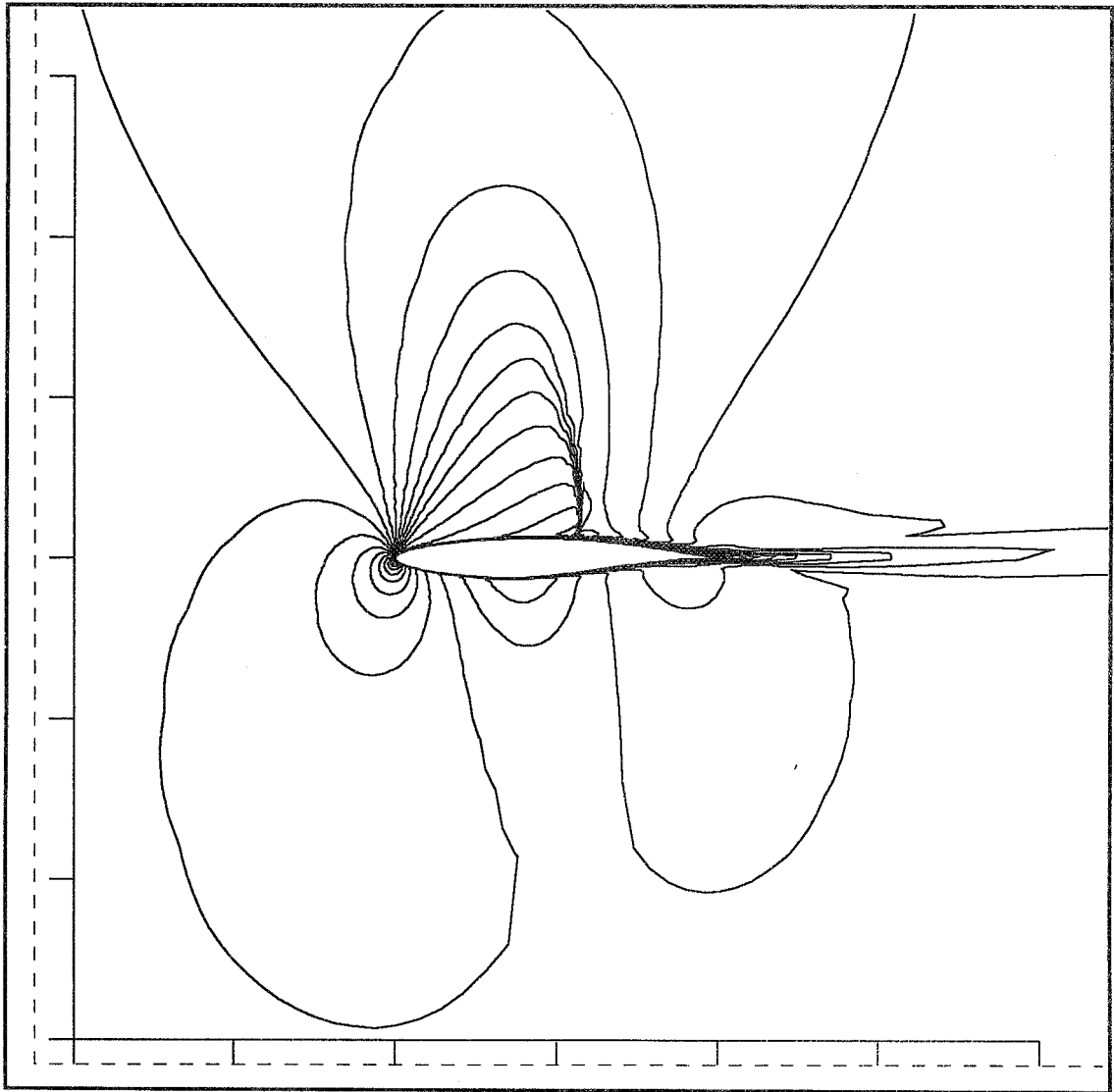


Figure 6. Contour Plot of Pressure Coefficients Around An Airfoil.

Chapter 1. Introduction

1.1 Summary

PLOT3D is a graphics program for interactive examination of fluid dynamics data. It helps scientists make their solutions visible. Using graphics workstations and other powerful computers, scientists use PLOT3D to *see* their work, to interactively zoom in or out, and to view the computational models from any angle. The quality of the solutions and the physics of the problem can be observed.

PLOT3D reads a grid file and a solution file with pre-computed density, x- y- and z-momentum, and stagnation energy for each grid point. From those, PLOT3D can calculate any one of 74 functions, including shock waves, surface pressure, velocity vectors, and particle traces. The grid file can have 50 grids.

Pieter Buning developed PLOT3D for computational fluid dynamics (CFD) research at NASA Ames Research Center. The program is also in use at many educational and industrial sites throughout the United States.

1.2 Scientific Visualization at NASA Ames

The primary mission of the Fluid Dynamics Division at NASA Ames Research Center is to research air flow over aircraft and spacecraft. Traditionally, Ames has used simplified analytical models and wind tunnels to test aeronautic designs. Computers have now become powerful enough to do some of the preliminary testing. Using the supercomputers at the Numerical Aerodynamic Simulation (NAS) Facility at Ames, scientists across the United States can remotely log in to study the fluid flow over a variety of complex shapes and use PLOT3D or other applications to turn their computations into colorful graphics.

The fluid is usually, but not necessarily, air. CFD physicists start by designing computational grids that model an object and the surrounding space. The object could be a wing or an entire complex aerospace vehicle. Using x y z coordinates to specify three dimensions, the scientists design complex grids with higher resolution in the strategic areas of study. Grids sometimes have as many as 2 million points, each of which will be used to calculate air flow characteristics. The grid lines need not be straight or regularly spaced; the grids may be bent to put extra grid points

(finer definition) in the regions of greater interest. The Cartesian coordinate location (x, y, z) of each grid point is arranged in the grid file according to **I, J, K** indexing. The **I, J, K** indices are set in relation to the object. **I** might start out in the x -direction, and **J** in the y -direction, but they follow the shape of the object. For example, **I** might go around the wing; **J** might go from the midline of the fuselage to the tip of the wing. **K** usually represents vectors that extend out radially from the object. The **I, J, K** indices correspond to the boundary conforming coordinate directions $\zeta, \eta,$ and ξ .

Scientists in the Fluid Dynamics Division develop 'flow-solvers', e.g., TNS, INS, PNS, and F3D. These are FORTRAN programs that run on supercomputers, such as the Cray2, and solve the system of equations (e.g., Navier-Stokes) governing fluid flow. When the grids have been designed and conditions such as angle of attack, Mach number, and Reynolds number have been specified, the flow-solver calculates the solution.

The solution file contains density, momentum, and stagnation energy values for each grid point. "Scientific visualization" is the process of turning research solutions into graphics, so that the results can be *seen*.

PLOT3D is the most popular and comprehensive program for visualizing the solution files. Alternatively, SURF may be used to interactively create Gouraud-shaded and smooth-colored function mapped surfaces; SURF offers fewer functions, but extra shading and coloring enhancements. The next version of SURF will accept function files from PLOT3D. RIP is an interactive particle-tracer that uses the Cray supercomputer. ISOLEV offers display-only viewing of planes sliced through the data. GAS combines objects created in PLOT3D, SURF, and RIP, and records animation sequences. A new program, FAST, being developed for the Silicon Graphics 4D, will provide an environment that supports modules to do the work of all of these existing visualization programs. ARCGraph (Ames Research Center Graphics) is a device-independent graphics library that enables data created on one machine-environment to be readable by another. ARCGraph is the Ames standard for graphics file format. See figure 7.

The ability to visualize the physical properties of air flow during flight is a tremendous aid in designing aircraft. Interactive computer graphics, by showing turbulence, temperature, reverse flow, pressure, and dozens of other functions, make the critical areas obvious so that they can be studied more closely using a finer grid. Thus, the aeronautic designs can be improved before the models are made for wind-tunnel testing.

Computational fluid dynamics and scientific visualization at Ames are in a state of exhilarating growth. The limits are being pushed.

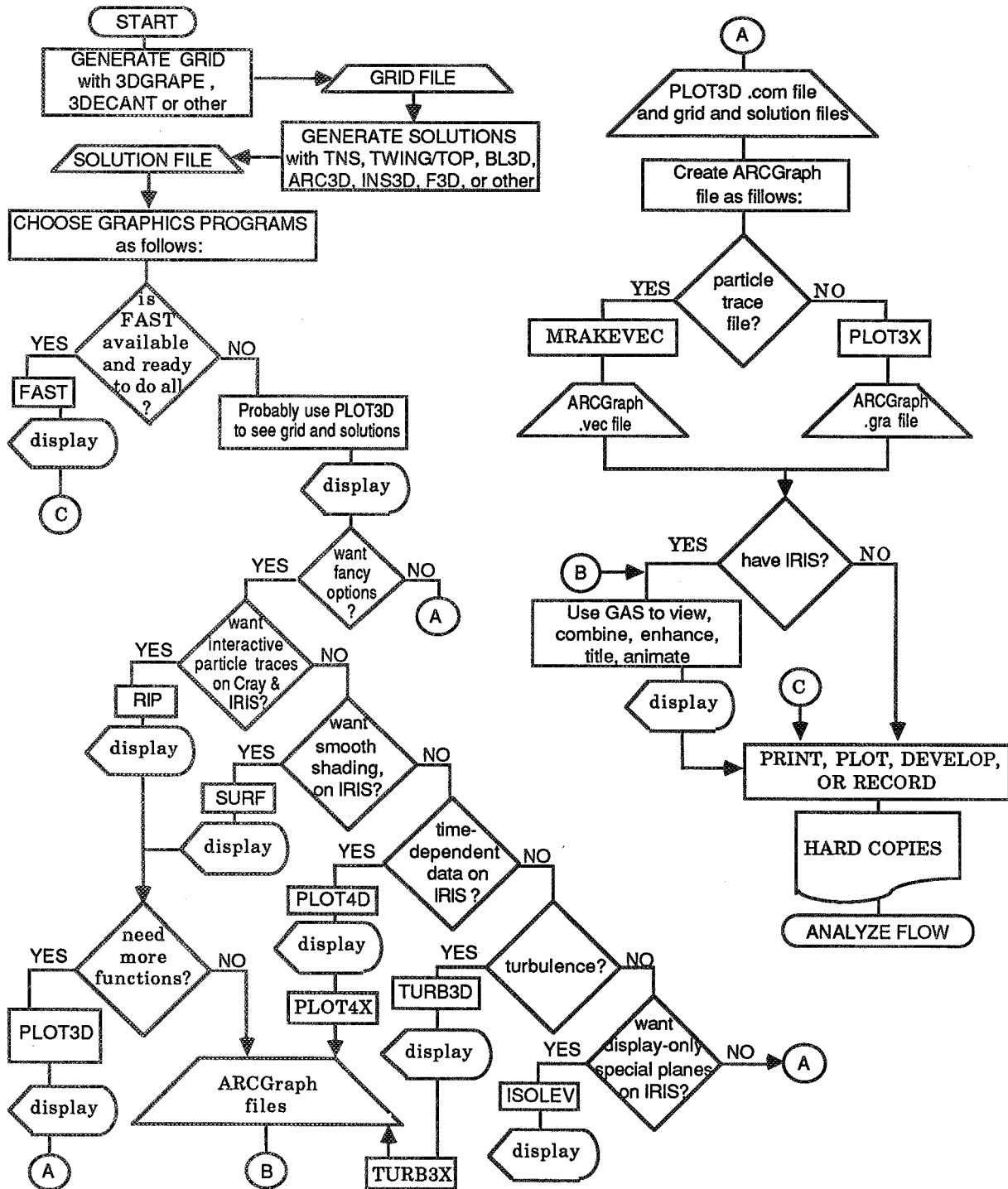


Figure 7. Numerical Aerodynamic Simulation software flow

1.3 PLOT3D Features

PLOT3D is the graphics workhorse of CFD. It offers more functions and features, and has been adapted for more types of computers than any other CFD graphics program. It is very widely used.

Pieter Buning released the first version in 1980 and version 3.6 in February 1989. This manual includes the release notes of version 3.6 as Appendix A and a note about new tools as Appendix B. The tutorial is suitable for versions 3.5 or 3.6; differences are mentioned.

PLOT3D reads grid and solution files and plots functions to produce on-screen graphics and files from which to produce animation and still photos or screen dumps. PLOT3D helps scientists understand their grids and their solutions. How will the simulated fluid flow? Is there enough grid resolution? Are boundary conditions correct?

PLOT3D lets scientists view and compare computed or experimental results. CFD output can be compared with wind tunnel results, (a) by putting wind-tunnel data into grid form and running it in PLOT3D, or, (b) taking photos of CFD on-screen plots and comparing them with oil flow photos.

PLOT3D has 74 functions divided into five groups:

- 1) GRID FUNCTIONS for walls (geometry), grids, outlines of IBLANK, and new grid-checking.
- 2) SCALAR FUNCTIONS for contour or carpet plots of density, pressure, temperature, Mach number, vorticity magnitude, helicity, etc.
- 3) VECTOR FUNCTIONS for vector plots of velocity, vorticity, momentum, and density gradient.
- 4) PARTICLE TRACE FUNCTIONS for rake-like plots of particle flow or vortex lines
- 5) SHOCK WAVE FUNCTION based on pressure gradient

The objects can be plotted with lines (wireframe), hidden lines, points, or flat-shaded surfaces. Various types and widths of line are available. Within the point line type, symbols can be used instead of dots. Shading and perspective are used to show depth. Color is used to denote data values.

Solutions are assumed to be computed on the grid points (as opposed to in the middle of the grid cell).

PLOT3D can accept multiple grids and an enormous complexity of data.

PLOT3D was first developed for the VAX computer; the user interface is based on VMS. Commands are entered from the keyboard or command files (rather than with a mouse/menu). The code is written in FORTRAN77.

On most workstations that have a mouse, the mouse can be used in PLOT3D to manipulate the *view* of the plot. With the touch of a mouse button, you can drag (translate), zoom, or rotate. You also can manipulate the view with PLOT3D keyboard commands. The actual dimensions of the object being plotted can only be changed by altering the grid file.

You can use command files to supply all the commands and parameters that PLOT3D requests. Most users develop command files to run their standard work. You also can create a *plot3dini.com* file that PLOT3D will read whenever you start a session. Command and initialization files are discussed in Chapter 5.

1.4 Host Systems

PLOT3D is available for the following computers and graphics libraries. Version 3.6 is now being distributed (except Sun/SunCore for which only 3.4 is available). See the Forward for distribution information.

- Silicon Graphics IRIS 2xxx or 3xxx /Unix (W3.5-GL2)
- Silicon Graphics IRIS 4D/70, 4D/70 GT/Unix (4D1-3.0)
- Generic Unix and IRIS (suitable for Cray2/Unicos, Convex, or Alliant with remote IRIS 2xxx/3xxx).
- VAX/VMS (V4.5) and DISSPLA (V10.5)
- VAX/VMS (V4.5) and DI-3000 (Version 4)
- Generic Unix and DI-3000 (Version 4)
- Apollo/Unix and GMR3D (Version 2.0) (no text on plots)
- Sun/Suncore (8 bit-plane color systems only). PLOT3D version 3.4 only.

Appendix C has information on `plot3ddi` for running PLOT3D on non-IRIS terminals. `Plot3ddi` uses the DI-3000 graphics library. At NAS, it is under `usr/local/bin` and there is a manual page on `navier`.

TUTORIAL

This manual assumes, unless otherwise noted, that PLOT3D is being run on a Silicon Graphics IRIS workstation. There are versions of PLOT3D for many types of machine: during the tutorial, the user may encounter variations.

Chapters 2 through 6 of this PLOT3D manual will take you, keystroke by keystroke, through a sample PLOT3D session, using a 3-D multiple grid data set with IBLANKING.

In order to use this tutorial, you will need:

1. Access to a computer on which the PLOT3D program has been installed.
2. Access to the data set *wingbodytail*. Details for copying *wingbodytail* into your current working directory are given in Chapters 2 and 3.
3. About 4 megabytes of memory for data. If you do not have that much, you can reconfigure PLOT3D by modifying the source code (see Chapter 9), and reconfigure the disk for more swap. Version 3.6 has dynamic memory allocation; therefore you no longer need to change NWORK in the source code, but you still may need to modify MGRIDS, and reconfigure the disk. If you think you may have enough memory, go ahead and try the tutorial and see what happens.

Chapter 2. Getting Started

2.1 Summary

This chapter explains how to get the directory *wingbodytail*, and gives you the keys to a demonstration testdrive of PLOT3D. Use of the mouse is described. Command structure, filename extensions, and the help file are discussed.

2.2 Introduction

Before you start PLOT3D, your data files must be accessible. PLOT3D reads two data files: an *xyz* file containing the grid points, and a *q* file with solutions for those grid points. For this tutorial, we will use a data set in the directory called *wingbodytail*.

Your job now is to help PLOT3D find the *wingbodytail* directory.

The preferred methods of file transfer or remote access are subject to frequent change; if the following suggestions do not work, ask a colleague for help, or call PLOT3D Support at (415) 694-4492 or User Services at (415) 694-4444.

For Version 3.5, you should make 4 megabytes of space available for running PLOT3D; do some housekeeping if necessary, or change PLOT3D as described in Chapter 9. Version 3.6 has dynamic memory allocation. This tutorial assumes that PLOT3D is installed on your system. If it is not, see the installation notes in 7.1.

2.3 Key to Tutorial Symbols and Styles

boldface = user input
\$, PLOT3D V3:, small plain type = prompts
↵ = <return>
^ = <control>
<i>italics</i> = user substitution, or emphasis, or directory name

Figure 8. Key to tutorial symbols.

Within sentences, PLOT3D commands and qualifiers are usually capitalized, but PLOT3D is *not* case-sensitive; there is no need to capitalize commands when you use them.

2.4 Getting Wingbodytail on an Ames IRIS Workstation

If you are at Ames, on an IRIS in the NAS system, use `nfsmount` to access `wingbodytail`. The directory is located on `wk23`. See figure 9.

```

                                     NAS WKS
system: 51> nfsmount wk23 ↵
system: 52>
```

Figure 9. `nfsmount` remote workstation.

If you get "command not recognised," call User Services at (415) 694-4444.

The `nfsmount` utility will give you access to all the files on `wk23`. When you run the testdrive, you do not need to be in your own directory, so you can simply change directories to `wingbodytail` on `wk23`. See figure 10. On the IRIS, the `cd` command should fit on one line; if not, use a hyphen (-) and <return> at the end of the first line. Sections 2.5 and 2.6, and Appendix F have directory information.

```

                                     NAS WKS
system: 52> cd /r/wk23/wk/pierce/examples/
               wingbodytail ↵
```

Figure 10. Change directory to `wingbodytail` on a workstation.

2.5 Getting Wingbodytail at Ames in VMS

On the Ames VAX/VMS network, copy `wingbodytail` from EARTH, as shown in figure 11. This is new; the directory was moved to EARTH, one of the VAX computers at Ames, in the spring of 1990. Be sure you have all the punctuation

correct. The last asterisk, preceded by a space means to copy the directory into your current directory. After you copy the *wingbodytail* directory, you need to reset its protection, so that PLOT3D can write a journal file in that directory. Then set your directory to *wingbodytail* and you are ready to start PLOT3D.

The "copy" command should fit on one line on the VAX; if so, you can omit the hyphen (-) and <return> after buning. Never put a space before "examples." See Appendix F if you need more information.

VMS

```

$ create/dir [.wingbodytail] ↵
$ sd [.wingbodytail] ↵
  STA0: [WALATKA.WINGBODYTAIL]
$ copy ear::stt0:[buning.- ↵
examples.wingbodytail]*.* * ↵
$ sd^ ↵
  STA0: [WALATKA]
$ set prot=(o:rwed) wingbodytail.dir ↵
$ home ↵
$ sd [.wingbodytail] ↵
  STA0: [WALATKA.WINGBODYTAIL]
$ █

```

Figure 11. Getting the *wingbodytail* directory into your VMS directory.

2.6 Getting Wingbodytail Off-site

The *wingbodytail* directory is on the PLOT3D tape distributed to off-site users. IRIS users look under `plot3d/3.6b/unix+iris2/examples/wingbodytail`. See Appendix F for alternatives.

2.7 Demonstration Testdrive


PLOT3D has the ability to read command files; most users eventually create their own files of PLOT3D keyboard work. Command files are designated by the extension `.com`.

Pieter Buning has written a command file for *wingbodytail*. The following tutorial screen, figure 12, will show you how to run the .com file. Before you start, be sure you are in the *wingbodytail* directory. The word "wingbodytail" should appear wherever your current working directory is indicated. In UNIX, the command **pwd** will show your working directory.

Throughout the tutorial, type the words that are printed in bold; PLOT3D should produce the non-bold words on your screen. In figure 12, you just type **plot3d** to start PLOT3D running, and then the PLOT3D banner should appear. If you get "command not recognised," call User Services at (415) 694-4444. Then type **@wbt.com**. The **@** tells PLOT3D that you are running a command file.

If you are using **nfsmount**, you will see an ERROR or WARNING message: "unable to start journal file plot3d.jou." For the testdrive, you can ignore that; in the next chapter, you can not. In the DISSPLA versions of PLOT3D, the opening banner always includes a warning message about the use of control-C.

If your testdrive is running smoothly, you will see PLOT3D report the files it is reading, as in figure 12.

You should see the PLOT3D prompts and responses whizzing by, and then the screen will go blank except for the cursor in the upper left corner. Within a few seconds, you should see parts of figure 13 appear on your screen. How long the plot takes depends on how busy your CPU is with tasks for other users. On a good day, the plot will be completed in half a minute or less. On a bad day, you can go get a drink of water and still have time to read much of this manual. The plot will ding when done, that is, you will hear a tone when the plot is complete, and the plot will stay on the screen until you press **<return>**. Do not press **<return >** while you are waiting.  When running on an IRIS and using the window manager (MEX or 4sight), use **<delete>** instead of **<return>**. That is, if you keep getting new windows and the old ones won't go away, use **** with the graphics cursor in the window you wish to delete.

As long as *something* is happening on your screen, you are successfully into PLOT3D. If the cursor just hangs there for a long time, use **exit <return>** to get out, check to make sure you have access to *wingbodytail*, and start over. If **exit <return>** does not work, use **control-C** (followed by **control-Y** on VAX/VMS). Or try the **break** key.

If your system will not start up PLOT3D, check to see that PLOT3D is installed, then see Chapter 9, or call User Services. Chapter 10 lists and explains all error messages.

Good luck!

```
system prompt > plot3d ↵

                PLOT3D Version 3.6      4 May 89
***An interactive graphics program for 2D and 3D CFD datasets***

Type "HELP" for a list of commands and general information,
"HELP Release_notes 3.6" for info on new features, bug fixes, etc.

PLOT3D U3: @wbt.com ↵
  New line:
  !
PLOT3D U3:
  ! Chimera scheme wing/body/tail demo.
PLOT3D U3:
  !
PLOT3D U3:
  ! Read wing/body/tail grid and solution.
PLOT3D U3:
  !
PLOT3D U3:
read/mgrid/blank/nocheck/x=wbtg/q=wbtr
SUBSETS reset to everything.
WALLS reset to nothing.
RAKES reset to nothing.
  Reading XYZ DATA for grid 1, 74x25x2
  Reading XYZ DATA for grid 2, 74x35x30
  Reading XYZ DATA for grid 3, 66x28x15
  Reading XYZ DATA for grid 4, 56x18x15
  Reading Q DATA for grid 1, 74x25x2
  Reading Q DATA for grid 2, 74x35x30
```

Figure 12. Starting the testdrive.

2.8 Geometry

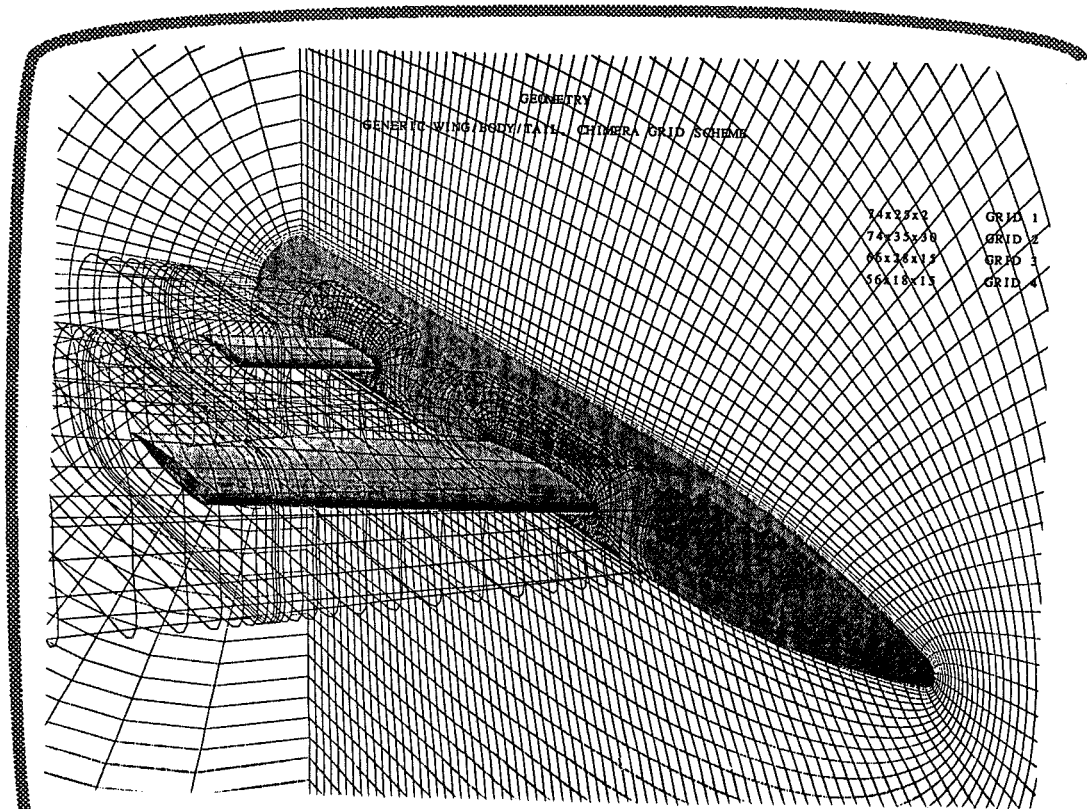


Figure 13. First plot in the tutorial: geometry.

The first plot you will see is titled:

GEOMETRY
GENERIC WING/BODY/TAIL, CHIMERA GRID SCHEME

The plot will stay on the screen until you press `<return>` or `<enter>`.

This plot shows the shapes of the grids. Grid one (irrelevant) is not shown. Grid two is the fuselage (body) of the airplane. Grid three is the wing, and grid four, the green grid in the back, representing a horizontal stabilizer, is the "tail."

"Chimera" (ki-mer-a), a composite monster in Greek mythology, is used in CFD to mean multiple grid.

The testdrive file uses the `WALL` command to specify certain parts of the grids to make visible. *Geometry* is the term used to describe the visible parts of the grid. As you can see, the grid extends far beyond the surface of the object.

Walls are specified with **I**, **J**, and **K** vectors, which are built into the design of the grid. The **I**, **J**, and **K** directions have no implied relation to *x*, *y*, and *z*. **IJK** for each grid are totally independent and need not have any relation to each other. In grid 2, the **I**-direction goes from the nose to the back of the fuselage, and **J** goes from the top midline of the fuselage around to the bottom, and **K** outward from the surface. In grid 3, **I** starts at the front of the wing and goes all the way around and back to the front (not staying in a straight line), and **J** goes out the wing. As in most CFD grids, **K=1** is the surface of the model, and other **K** vectors go radially out from the surface.

Only parts of the grids are shown here; trying to show all the lines in all four grids could crash your machine, and be too dense to interpret. For grid two, the first **K** layer (model surface) is shown in gray-filled polygons (solid), and the last **I** layer is shown in white grid lines. The surface of the wing, grid 3, is shown in solid red, with the surrounding grid in red lines. Grid four is shown in green.

The legend in the upper right corner shows the overall **IJK** dimensions of all the grids in the data set.

While this plot is still on your screen, go on to the next section.

2.9 How To Use The Mouse

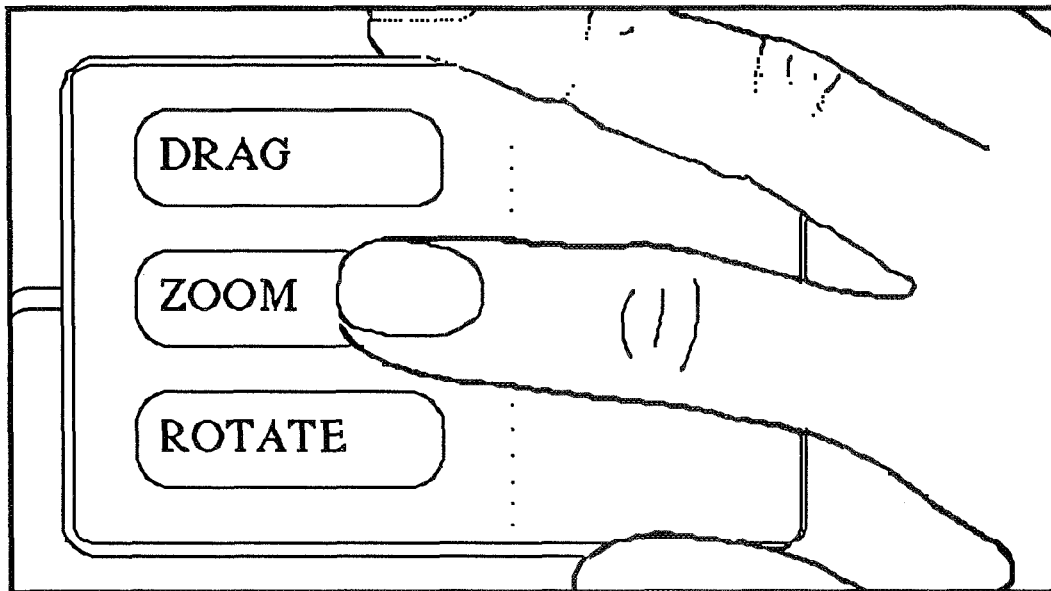



Figure 14. Mouse buttons: drag, zoom, rotate

On the IRIS and some other systems, the mouse can be used to manipulate the view of the plot. The mouse buttons are used to activate different degrees of freedom (types of movement) for manipulation, while the mouse (x,y) *displacement* controls the RATE of motion in the respective degree of freedom.  In other words: the *position* of the mouse controls the *rate* of motion (speed) of the object. To move the object farther, *hold the mouse button down longer, without moving the mouse farther.*

- Which button you press determines type of movement.
- The direction you move determines the direction of movement.
- How long you hold the button down determines amount of movement.
- How far you move the mouse determines speed of movement.

While the left mouse button is being depressed, x-motion of the mouse is associated with rotation (about the center of the MINMAX box) in phi, while y-motion controls rotation in theta. (Phi and theta are cylindrical coordinate angles using the DISSPLA convention). The MINMAX box is explained in section 3.7.

The middle mouse button activates zooming (toward the center of the MINMAX box) controlled with the y-motion of the mouse. X-motion is not used.

Pressing the right button allows the picture to be dragged left or right (x-motion of the mouse), and up or down (y-motion) on the screen. Dragging is another word for translating.

The sensitivity of the mouse to zooming and screen dragging is a function of the distance of the VPOINT initial viewpoint from the center of the MINMAX box. The closer the viewpoint, the less sensitive the mouse. While the plot is on the screen, this can be adjusted by typing the "up-arrow" (increase) or "down-arrow" (decrease) keys.

More than one button can be depressed at one time. The mouse cannot be used to pick a specific point within the plot.


On IRIS 4D's, put the graphics cursor in the window of the object to be moved.

Experiment now with the different motions of the mouse. Press the appropriate mouse button, move the mouse, and hold the button down until you get the viewpoint you want.

As you change your viewpoint, you can see that PLOT3D automatically uses perspective for depth cueing, that is, to show which part of the grid is farther away from the viewpoint.

If you are not on an IRIS or other system that allows use of the mouse with PLOT3D, you will have to manipulate your viewpoint with keyboard commands, as explained in Chapter 3.

2.10 Return to Command File — <return> or <delete>

When you have finished looking at the plot, press <return> (put the graphics cursor in the window and use the **DELETE** key to close the window (<return> opens a new window). You will see PLOT3D zip through more of the command file.  If you want out of the command file, press <break> or control C.

2.11 Example of a Surface Plot: Pressure Coefficient

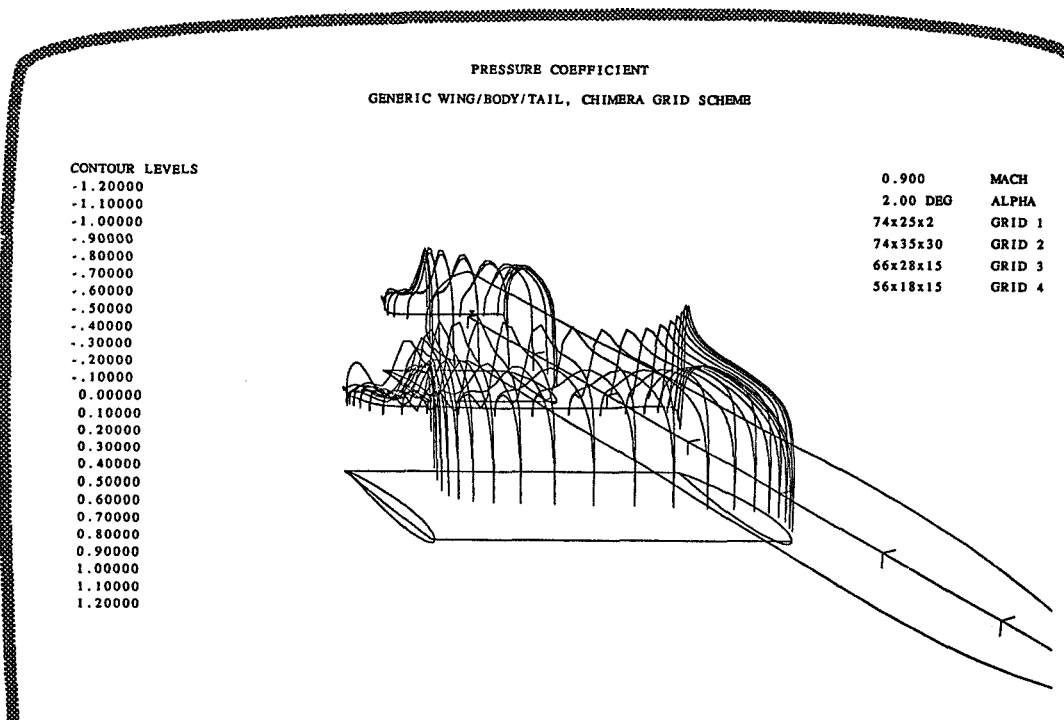


Figure 15. Example of fsurface plot of pressure coefficient.

The next screen, figure 15, will show a scalar plot. Scalar plots represent functions, such as pressure or temperature, which can be expressed as values on a linear scale. Scalar values are often plotted with contour lines, but sometimes, as in this example, scalar values are represented with an fsurface (carpet) plot. In this case, the pressure coefficient function has been calculated for a series of rings around the surface of the wing and the horizontal tail (stabilizer). The value of the function is shown by two lines, one for the top surface and one for the bottom. The lines form a *graph* of the function; the vertical axis of the graph is the value of the function,

and the horizontal axis is the chord length of the wing (from the front to the back) and the tail.

Pressure coefficient is shown inversely; a high point in the graph indicates low pressure (lift). The difference between the pressure coefficients for the top and bottom of the wing represents the overall lift .

Use the mouse to zoom in and have a close look at the lines. The fact that there are various colors within each line indicates that this is a *fsurface* plot. A contour plot would have had only one color per line. (Except grid contour plots do have multiple colors per line). Notice that the legend on the left is labeled "contour levels" even though it is not a contour plot. The legend shows which colors indicate which values.

The difference between *fsurface* and contour plots is discussed in Chapter 4.

This plot uses simple outline walls of the wingbodytail.

When you have finished looking at this plot, press return to continue with the demonstration.

2.12 Mach Contours on Slices

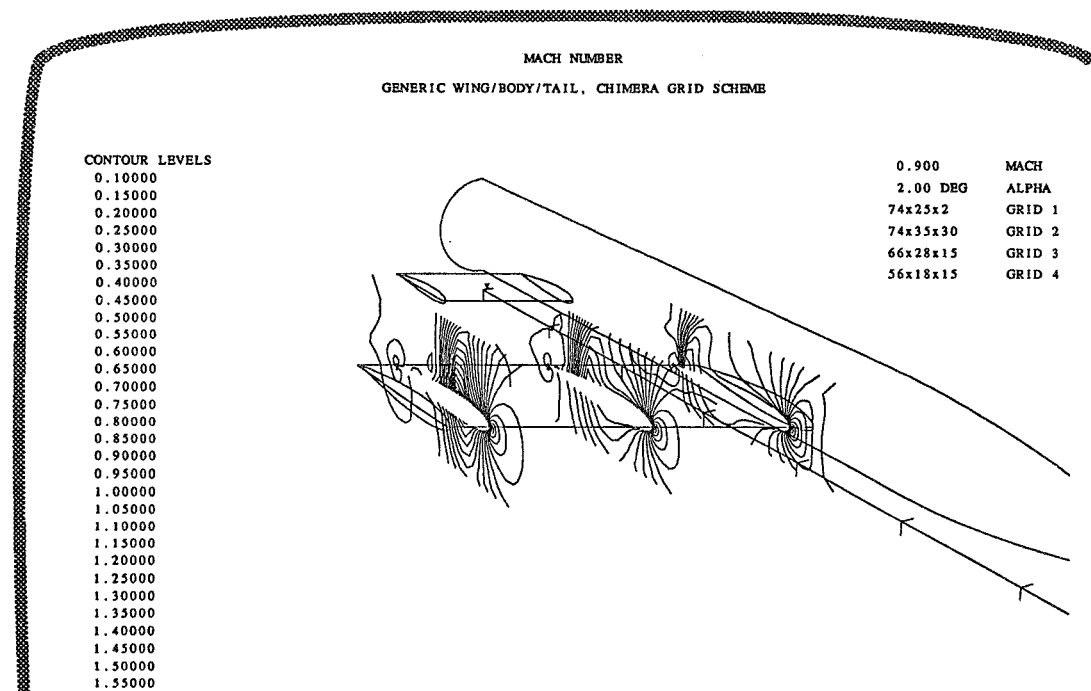


Figure 16. Mach contours on slice subset of wing.

ORIGINAL PAGE IS
OF POOR QUALITY

This plot, figure 16, is a *contour* plot. Each line shows the *location* of one value of the function, which in this case is Mach number. The subsets (active parts of the grids) have been set to slices through the wing; within the subset, each colored line shows where a particular value is found. Notice that each line has only one color.

Press return to continue.

2.13 Mach 1

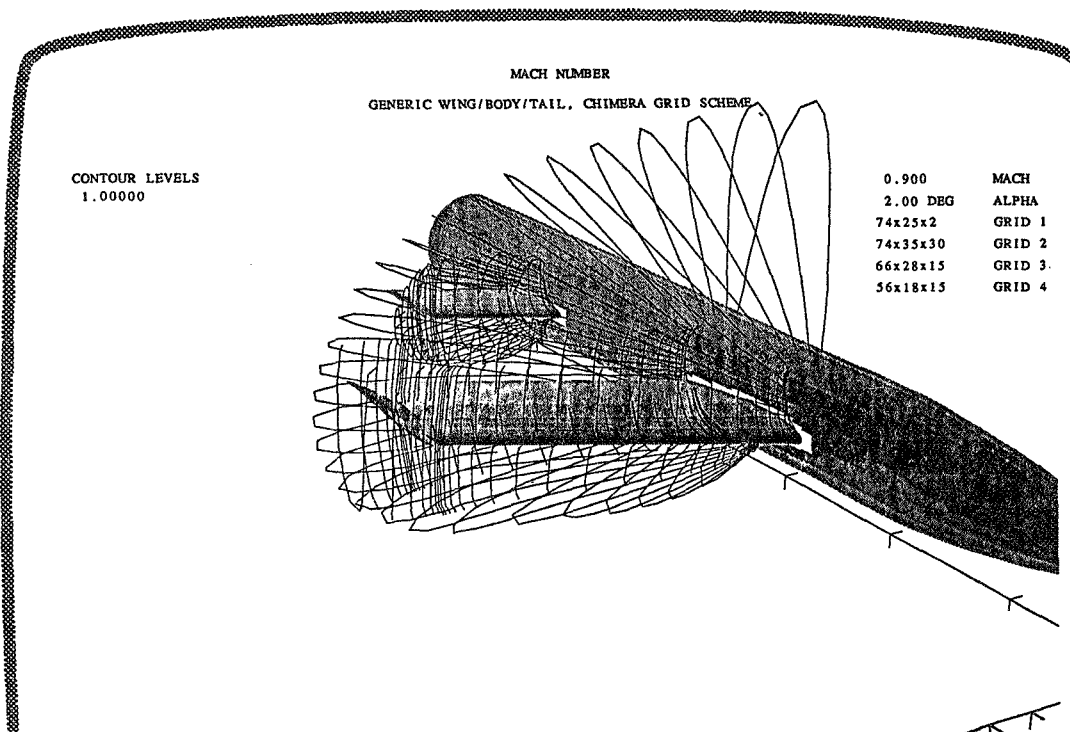


Figure 17. Mach 1.

You will have to wait a little longer for this plot, figure 17, because it uses solid walls. You know a plot is finished when you see the color legend or hear "beep."

The subsets here are set to everything, that is, to all parts of the grids. The /MANUAL qualifier is used with the CONTOURS command to limit the plotted function to Mach 1 (see Chapter 11, Mach 1, for more on this technique).

Press return to continue.

2.14 Contour Plot of Pressure on Surface

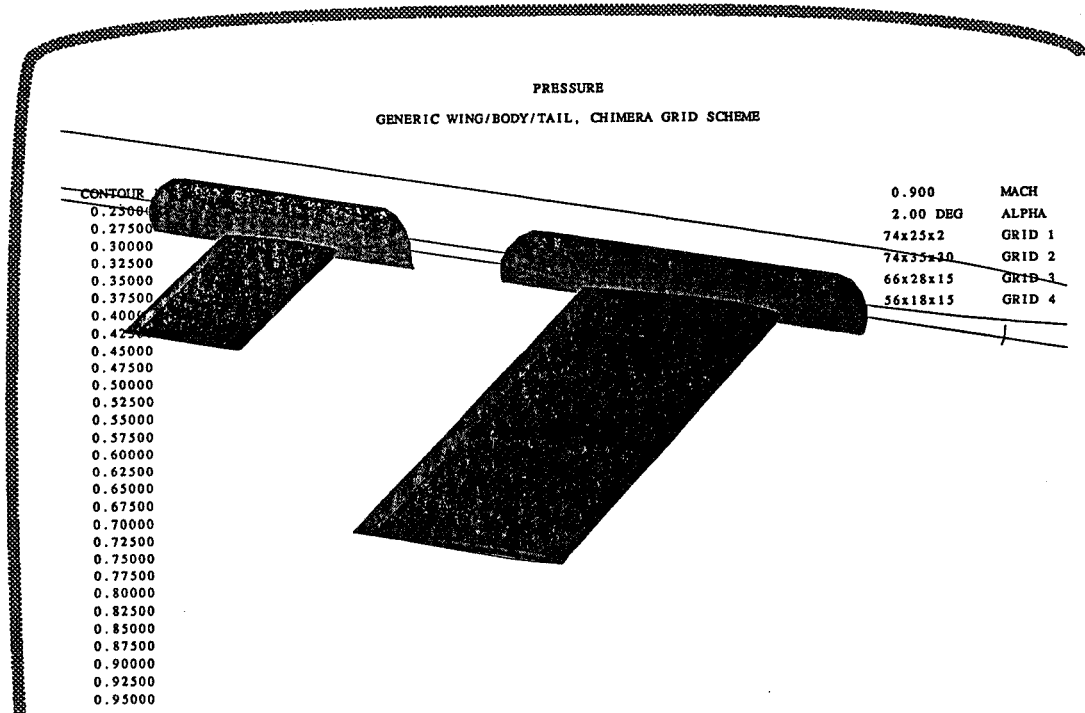


Figure 18. Color contour plot of pressure.

Figure 18 shows a color contour plot of the pressure function. This is another contour plot; the function appears within and throughout the subset. The subset has been set to the surface of the wing and tail, but that does not make it an "fsurface" plot. This style is called "function mapped." The attribute "color_contours," under the CONTOURS command, was used. Hit <return> or to continue.

ORIGINAL PAGE IS
OF POOR QUALITY

2.15 Particle Traces

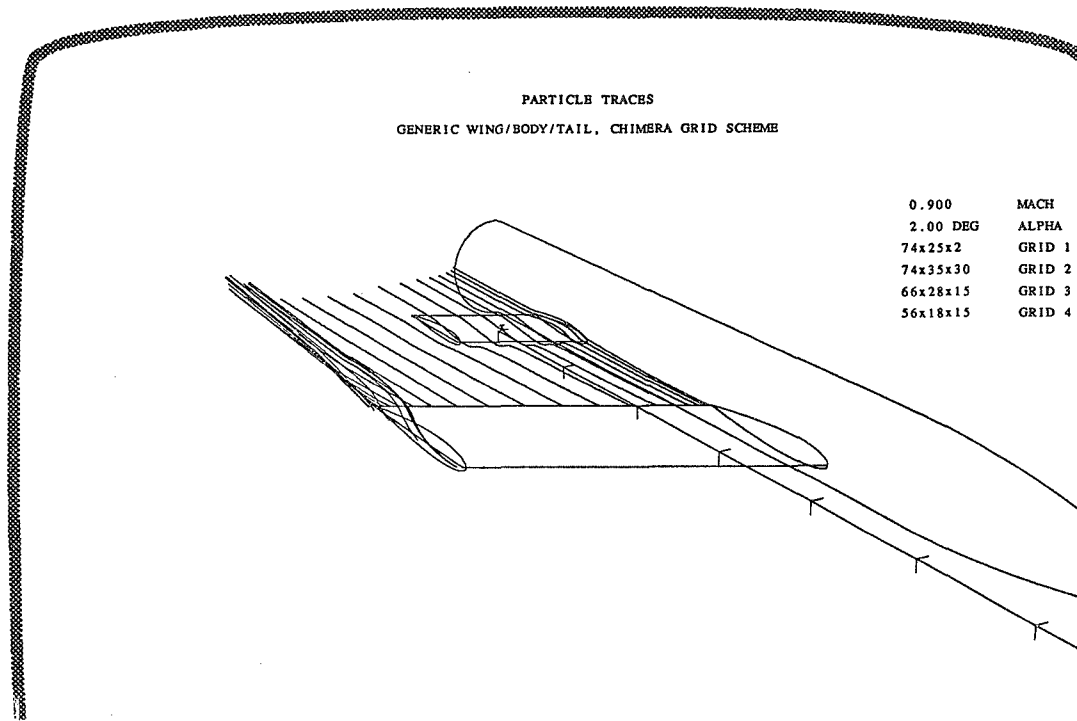


Figure 19. Particle traces, wingbodytail.

If you do not have a lot of memory on your machine, this plot, figure 19, will probably crash. If it crashes, press **control-C** (followed by **control -Y** on VAX/VMS) to get out of PLOT3D or back to PLOT3D prompt. You may try changing "rake/xyz" in the wbt.com file to "rake/xyz/write=filename"... on some systems the default temp files cannot be created, so you must supply a file spec of your own. Or just go on to the next section of this tutorial. If necessary, type **plot3d** from the system prompt when you go on to the next section. You will have a chance to work with particle traces in Chapter 4. Chapter 9 discusses ways of getting PLOT3D to use less memory... or you could make more memory available.

For particle traces, the subsets are usually set to everything, and then the **RAKE** command is used to set the release points of the traces. See section 4.11.

The solution files used for PLOT3D are calculations for the flow at one moment of time; they are like a snapshot of the flow. "Particle traces" are the paths taken by neutral particles moving through a flow field that is unchanging in time (steady flow). That is, the particle traces are streamlines.

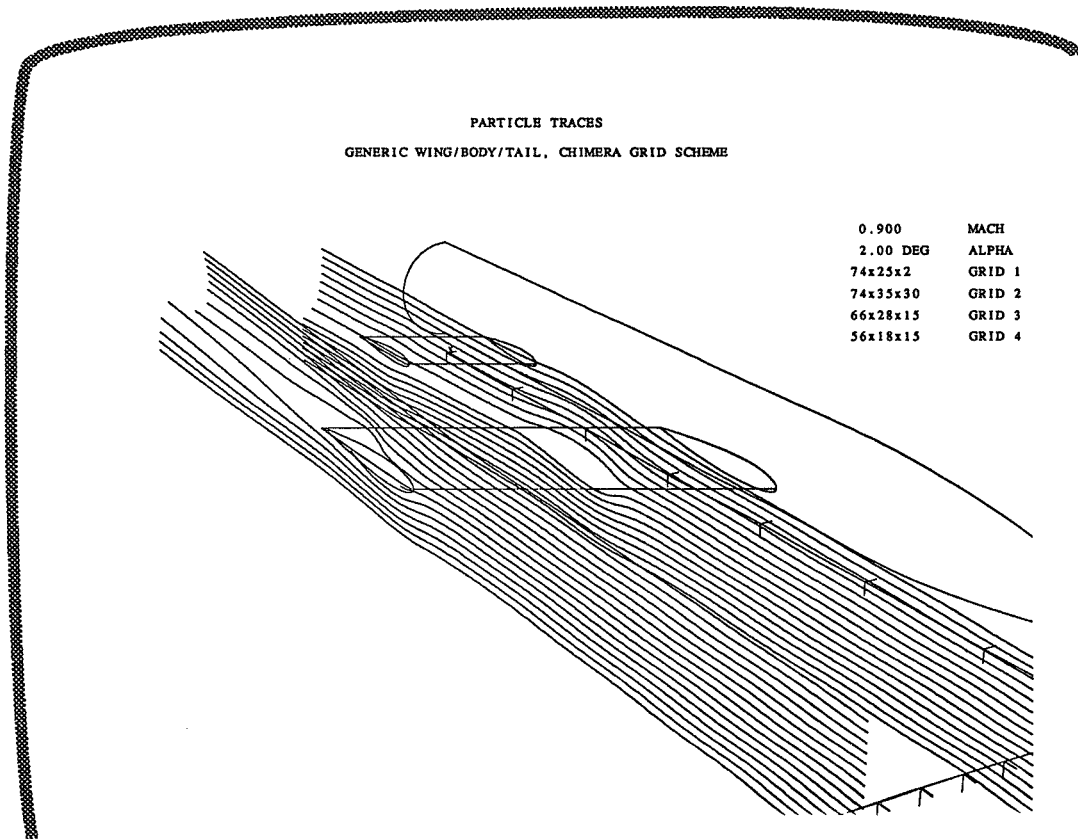


Figure 20. Another plot of particle traces.

2.16 End of the Testdrive

Hit **<return>** or **** to continue. If your testdrive of PLOT3D made it through the particle trace plots, you will have reached the end of the testdrive. Depending on the version you are using, at the end of a command file, PLOT3D will prompt you for a new line, or exit you from PLOT3D. If you are at "new line", just stay there while you read the section on command basics, then try some of the suggestions about the help file. If you are out at the system prompt, just type **plot3d** to get back in. If you want to quit for a while, see section 2.20.

2.17 Command Basics

PLOT3D command syntax is based on DCL, the DEC Command Language. This is the basic form: **command/qualifier=argument arguments .**

Here is an example:

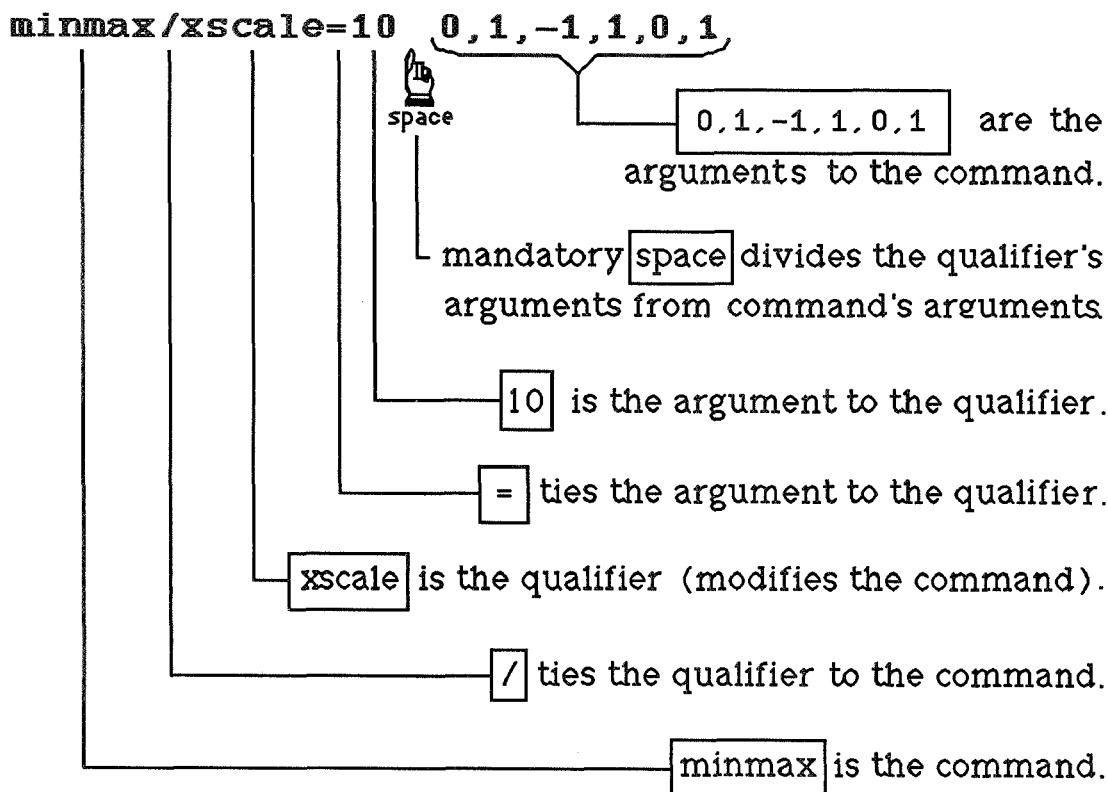


Figure 21. Command basics.

This command line means that the minmax box (the area to be plotted) will include those parts of the grid from $x = 0$ to $x = 1$, from $y = -1$ to $y = 1$, and from $z = 0$ to $z = 1$, with the qualification that the x-axis scale factor equals 10.

The order of these elements can be changed somewhat, as long as the command is first. Thus, the same command line could be written as:

```
minmax 0,1,-1,1,0,1 /xscale=10
```

Spaces before a slash (/) or after an equals sign (=) are optional. A slash followed by no blank indicates a qualifier, for example, `/xscale`. An equals sign preceded by no blank indicates an argument to the qualifier, for example, `xscale=10`. A blank precedes the arguments to the command. Arguments are values, sometimes called parameters. If the *qualifier* expects a group of values, enclose them in parentheses, for example, `/figure=(10,20,30)`. Arguments are separated by commas, blanks or both: `0 1 -1 1` or `0,1,-1,1`.

Identify distinct text strings with double quotes :

```
text "NASA Ames Research Center" "Moffett Field "
```

Double quotes are also used to enclose a path name in the READ command line. See section 3.3.

Continuation character : if the last character on a line is a hyphen () , the next line is assumed to be a continuation of this line.

PLOT3D accepts abbreviations. You need only type enough letters to specify a unique command. Thus, `e` suffices for `exit`, and `plot` can be simply `p`.

PLOT3D commands and qualifiers are NOT case-sensitive .

2.18 File Names and Extensions

In PLOT3D, file names follow the form: base.extension. There are no restrictions on the base of the file name, except that in most UNIX systems, a file name with its extension must be 14 characters or less. PLOT3D assigns the default extensions. The inclusion of an `x` in grid file names and a `q` in solution file names is customary and useful; for example: `shuttlex.bin` and `shuttleq.bin`. Most of the following extensions are applied automatically.

Extensions:

.bin	binary data
.com	command file
.dat	FORTRAN unformatted data
.dip	Device Independent Plotter file
.fmt	formatted data
.gra	general ARCGraph file (use PLOT3X to create)
.jou	journal file; record of all keyboard entries
.par	particle trace file
.pol	ARCGraph file for flat shaded polygons (subset of .gra)
.ps	PostScript (new in Version 3.6)
.prs	ARCGraph file for pressure or other function (sub. of .gra)
.qms	screen save (QMS QUIC format)
.sei	screen save for Seiko color printer
.ras	pixel file screen save
.vec	ARCGraph file for lines (subset of .gra)

ARCGraph is the Ames Research Center Graphics device independent library of standard graphics utilities.

Data file formats for input to PLOT3D are explained in Chapter 8; most grid and solution programs format the data correctly. Files for saving PLOT3D output are explained in Chapters 5 and 6, and a chart showing origin and uses of file types is in Chapter 12.

2.19 On-line Help

To access the on-line manual while running PLOT3D, type **help** or **help *command optionalqualifier***. See figure 22.

```
PLOT3D U3: help ↵
```

```
HELP
```

```
PLOT3D Version 3.5    20 May 1987
```

```
Format:  HELP [keyword [keyword [keyword ...]]]
```

The HELP command prints information on a list of keywords. "*" is a wildcard, while "*..." matches anything at the current level or below. Thus "HELP *..." prints all HELP information available. When responding to a "Topic?" or "Subtopic?" prompt, a "?" causes information for the current level to be repeated; a <RETURN> pops HELP up one level, and an end-of-file terminates the HELP session.

Additional information available:

HELP	General_information	Journal_file	Initialization_file				
Installation_of_PLOT3D	Release_notes	Special_characters	Blanking				
Attributes	@file	End-of-file	\$system-command	CLEAR			
CONTOURS	EXIT	FSURFACE	FUNCTION	LIST	MAP	MINMAX	PLOT
VPOINT	WALLS						

```
Topic? function ↵
```

```
FUNCTION
```

```
Format:  FUNCTION [function number]
```

Allows the selection of the function number to be plotted. Numbers between 0 and 99 are for grid-type information, between 100 and 199 are scalar functions (suitable for contour plots), and numbers between 200 and 299 are vector fields. Note that currently 200-299 refer to vector plots, while 300-399 generate particle trace-type plots. Functions from 400 on up are special functions (such as shock wave locations).

Additional information available:

Grid_functions	Scalar_functions	Vector_functions
Particle_trace_functions	Shock_waves	Nondimensionalizations
Fluid_constants	Function_definitions	

Figure 22. Help.

If you type just **help**, you will get a list of the major topics of help available, as in the preceding example. Each topic lists subtopics. When you use more than one key word, the topic must precede the subtopic, as shown in figure 23.

```

Topic? function scalar
FUNCTION
  Scalar_functions
    100 Density (or Q1).
    101 Normalized density.
    102 Stagnation density.
    103 Normalized stagnation density.
    110 Pressure.
    111 Normalized pressure.
    112 Stagnation pressure.
    113 Normalized stagnation pressure.
    114 Pressure coefficient.
    115 Stagnation pressure coefficient.
    116 Pitot pressure.
    117 Pitot pressure ratio.
    118 Dynamic pressure.
  
```

Figure 23. Help subtopics.

(A complete listing of function categories and numbers is on the reference card and in Chapter 7.)

An asterisk, after the key word and a space, gives you more information than would be provided with just the key word. An asterisk followed by three dots gives you all the information available for that file. Thus,

```

help function
gives a paragraph about functions and lists the categories.
help function *
lists all the functions.
help function *...
lists all the functions and gives all the information available about functions.
  
```

Typing a question mark causes the preceding information to be repeated.

The help file is organized as a hierarchy of levels; each topic has subtopics. After giving information on a topic, PLOT3D will prompt you for a subtopic. If you want to go to a different topic, first press <return>.

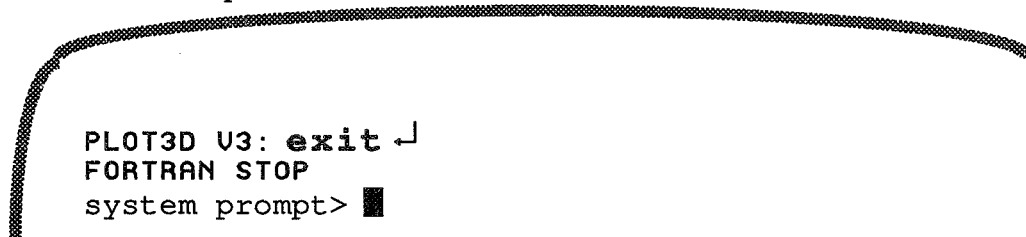
To exit the help file, press <return> again.

If you need help that is not in the help file, call PLOT3D Support at (415) 694-4492 or FTS 464-4492.

For help with a UNIX operating system on an IRIS workstation, call User Services at (415) 694-4444.

2.20 Exit — exit or exit/save

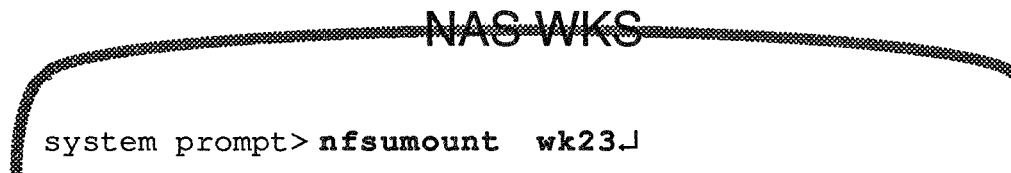
If you need to end your session now, type `e <return>` or `exit <return>` as shown in figure 24; otherwise just go ahead to the next chapter. EXIT/SAVE is discussed in Chapter 5.2.



```
PLOT3D U3: exit ↵
FORTRAN STOP
system prompt> █
```

Figure 24. Exit from PLOT3D.

If you are on a NAS workstation, and you are not going to continue the tutorial today, use `nfsumount`, as shown:



```
system prompt> nfsumount wk23 ↵
NAS WKS
```

Figure 25. `nfsumount`

When you start up again, use `nfsmount wk23`, as in section 2.4.

Chapter 3. Loading and Displaying Data

3.1 Summary

This chapter shows how to load data and specify geometry, that is, how to use the READ command to read grid and solution files and how to make parts of the grid visible with the WALL command. The TEXT, MINMAX, VPOINT, PLOT, and EXIT commands are also illustrated.

3.2 How To Do It

There are five steps in a PLOT3D session:

- load data
 - specify geometry
 - specify function (covered in Chapter 4)
 - plot / view interactively
 - save to file (covered in Chapter 5)

This tutorial continues to work with *wingbodytail*, the directory containing grid and solution files describing a simplified generic fuselage with wing and tail. This data set gives you the opportunity to work with multiple grids.

3.3 Data In Current Directory

Usually, when you run PLOT3D, you must either have your grid and solution files in your current working directory or use a path name. For the testdrive, if you were on a NAS IRIS, you only needed to access the files; you changed directories into Pieter Buning's wingbodytail directory because you were just running a command file. But when you create new plots, PLOT3D keeps a journal file of your work and if you crash or type "exit/save" PLOT3D will save the journal file ("plot3d.jou") in your current working directory. If you are in someone else's directory, without "write" permission, PLOT3D cannot create a journal file and most

versions of PLOT3D will not create a new plot. On the IRIS 4D, you may be able to create a plot without being able to create a .jou file.

If you do have *wingbodytail* as your own current directory, skip this section. The following assumes you have read Chapter 2, sections 2.4, 2.5, or 2.6.

For the main tutorial, those of you using nfsmount will have to either give a path name in the READ statement, or copy the data into your own directory. Because the data set is large, the path name method is preferred. Enclose the path name in double quotes. Thus, nfsmount users substitute this statement for the read statement in the following section:

```
read/mgrid/blank/nocheck/x="/r/wk23/wk/pierce/examples/-
wingbodytail/wbtg"/q="/r/wk23/wk/pierce/examples/-
wingbodytail/wbtr"
```

Note the slashes before the (remote) r's, and the hyphens (followed by <returns> to continue lines. PLOT3D needs the quotes to differentiate directories from qualifiers.



See Appendix F for directory details.

In general, if you need to use a file not in your current working directory, enclose the path name in double quotes. Alternately, you could alias the file names in your login file, or type only `read/mgrid/blank/nocheck` and PLOT3D will prompt you for the file names, and you would not need quotes:

```
PLOT3D V3: read/mgrid/blank/nocheck <return>
Enter xyz DATA filename: /r/wk23/wk/pierce/examples-
/wingbodytail/wbtg <return>
Enter Q DATA filename: /r/wk23/wk/pierce/examples-
/wingbodytail/wbtr <return>
```

Getting the data may be the hardest part of PLOT3D; if you are having trouble, ask for help from a colleague familiar with your system, or call PLOT3D Support at (415) 694-4492, or, for nfsmount help, User Services at (415) 694-4444.

Before you start the next section, check to see that you are in your own *wingbodytail* directory, or that you are in your own directory and are prepared to use a path name in the READ statement. If you are using a VMS or off-site system, refer to sections 2.5, and 2.6 for path names to the directory.

3.4 Loading Data — read/x=gridfile/q=solutionfile

If you are still running PLOT3D from the tutorial, and you are in your own directory, you can skip the `plot3d` command in figure 26, but do enter the `read` command in figure 26. If you are using `nfsmount`, you are probably not in your own directory; be sure you have read section 3.3. If you don't know what directory you are in, look for clues in the system prompt; if still in doubt, type `exit` from the PLOT3D prompt, then `pwd` from the UNIX system prompt.

```

system prompt > plot3d ↵

                PLOT3D Version 3.6      4 May 89
***An interactive graphics program for 2D and 3D CFD datasets***

Type "HELP" for a list of commands and general information,
"HELP Release_notes 3.6" for info on new features, bug fixes, etc.

PLOT3D U3: read/mgrid/blank/nocheck/x=wbtg/q=wbtr ↵
SUBSETS reset to everything.
WALLS reset to nothing.
RAKES reset to nothing.
  Reading XYZ DATA for grid 1, 74x25x2
  Reading XYZ DATA for grid 2, 74x35x30
  Reading XYZ DATA for grid 3, 66x28x15
  Reading XYZ DATA for grid 4, 56x18x15
  Reading Q DATA for grid 1, 74x25x2
  Reading Q DATA for grid 2, 74x35x30
  Reading Q DATA for grid 3, 66x28x15
  Reading Q DATA for grid 4, 56x18x15
PLOT3D U3: █

```

Figure 26. Starting PLOT3D: the READ command.

From the PLOT3D prompt, your first command should be `read`, followed by a slash (/) followed by the qualifiers: `mgrid` means there will be multiple grids; `blank` means there will be IBLANKing, which creates holes or solid boundaries (turns off grid points or indicates grid adjacency) in one or more of the grids such that flow can move realistically from one grid to another. Each qualifier is preceded by a slash. `Nocheck` is used here to speed up the running of the READ

command; when you are using data that has not been checked for negative or inconsistent pressures or densities, omit this qualifier.

The fundamental qualifiers for the READ command are *x* and *q*, which specify the grid and solution files to be used. The form is:

/x=gridfile/q=solutionfile

If you are using path names rather than having the data set in your current working directory, substitute the full path name, in quotes, for the *x* and *q* file names.

If you use READ without *xyz* or *q* file names on the command line, PLOT3D will prompt you for them. This is often easier than typing one long statement.

Chapter 7 gives a complete listing and explanation of all the qualifiers to READ. After the READ command line (and the */x* qualifier), PLOT3D resets SUBSETS to everything, which means that when you specify a function, the function will be calculated for all parts of the grid. Functions are covered in Chapter 4; when you get there, you will see how to specify subsets. Walls are reset to nothing; in order to see the geometry, you must specify walls, as shown later in this chapter. Rakes are reset to nothing; they will be used when you plot particle traces. (READ/*q* does not reset anything).

WARNING: if you get an error message, do not proceed; stop and fix the problem. If it says "unable to access *xyz* file," you either do not have *wingbodytail* as your current working directory, or you misspelled the file name or path name. See section 3.3 or the first few pages of Chapter 2 for help in locating *wingbodytail*. See Chapter 10 for explanations of other error messages. DO NOT proceed if you see "error: unable to open .jou file." You must be in your own directory; otherwise, PLOT3D cannot write the journal file.

PLOT3D will tell you the IJK dimensions of the *xyz* and *q* files it is reading. If you find yourself at a new PLOT3D prompt without having entered the *q* filename (*wbtr*), use the read statement again with all the qualifiers, but with the *q* filename instead of the *xyz* (for example `read/mgrid/blank/noc/q=wbtr.`).

3.5 Text — text

```

PLOT3D U3: text ↵
Enter up to 2 lines of text.
? GENERIC WING/BODY/TAIL, CHIMERA GRID SCHEME ↵

? ↵
PLOT3D U3: ■

```

Figure 27. Text.

Enter text as shown in figure 27. If you run out of room and want to continue to a new line on your screen, use a hyphen (-) at the end of the first line. PLOT3D will automatically print a title in capital letters, across the top, indicating the function that is shown. In this case, we are just looking at the grid, so the automatic title will be "GEOMETRY." You can add one or two lines of text to be centered above future plots. The text will continue to be used until another TEXT command is given. Current text can be deleted by pressing return (↵) in response to the prompt for text. If text is entered on the command line, *be sure to enclose each line in double quotes*:

```
text "Wingbodytail" "NASA Ames Research Center"
```

Titles are often added later, when a plot has been converted to GAS (see Chapter 5).

3.6 Defining Geometry — wall/grid=2

Be careful not to skip the <returns> (↵) in the next screen, figure 28. When you get to the bottom of this illustration, do NOT press <return>; go on to figure 29. Notice that if you press <return> by mistake, you get another chance.

```

PLOT3D V3: wall/grid=2
  Wall  1,  grid  2
Enter I startI,endI,incI): last ↵
                        : ↵
Enter J startJ,endJ,incJ): 2 34 ↵
                        : ↵
Enter K startK,endK,incK): all ↵
                        : ↵
LINES, HIDDEN_LINES, POINTS, or SHADED_SURFACE? line ↵

Enter color: rgb .6 .6 .6 ↵
Enter line type: ↵
Enter line thickness factor: ↵
Symbol type? ↵
Enter symbol size factor: ↵

  Wall  2,  grid  2
Enter I startI,endI,incI): █

```

Figure 28. WALL command.

From the PLOT3D prompt, specify the geometry (which parts of the grid you want to see) with the *wall* command, followed by */grid=gridnumber*. PLOT3D will say "Wall 1, grid *gridnumber*" and then prompt you to indicate where in the I-direction you want the wall to start. In *wingbodytail*, in grid 2, the I-direction goes from the nose to the back of the fuselage, and J goes from the top midline of the fuselage around to the bottom, and K outward from the surface. In this case, we want the last I-vectors in grid 2. This wall will appear as semicircular grid lines at the back of the airplane. PLOT3D uses square brackets [] to prompt

for optional input; when stating walls, you have the option to specify the end and the increment.

For this tutorial, just follow the screens, but in your own work, consider the following options. If you want to see 2, 4, 6, and 8, you would write `2 8 2 ↵` (start-end increment arguments separated by spaces or commas). If you state just one number, that will be the only level shown, that is, "start" is also "end" if no end is stated. What you want may not be separated by even increments; PLOT3D gives you another chance: if you want 2, 3, and 7, type `2 ↵`, and PLOT3D will give you the "Enter I start . . ." prompt again. Enter `3 ↵`, get the prompt again, and enter `7 ↵`. You keep getting the same prompt until you simply press `↵`.

Remember to always press an extra `<return>` after entering the last I, J, or K.

If you have entered a line that is not what you want, just keep pressing return until you get back to the level you need to re-specify.

In grid 2, wall 1, we want to go from 2 to 34 in the J-direction, and we want all the K's (you can use "a" for "all").

Next, PLOT3D wants to know whether you want your wall expressed in lines, hidden lines, points, or as a shaded surface. Enter `L` for lines (`P`, `S`, or `H` suffice for points, shaded_surface, or hidden_lines.)

The next prompt is "Enter color ." You may specify a particular mix of the primary colors of light: red, green, and blue. In the preceding example, `rgb .6 .6 .6` means red at 60% intensity mixed with green at 60% mixed with blue at 60% (it comes out gray). Alternately, you may state one of the following colors:

- magenta
- red
- yellow
- blue
- green
- cyan
- white
- black.

```

Wall  2,  grid  2
Enter I start[,end[,inc]]: all↵
                               : ↵
Enter J start[,end[,inc]]: 34↵
                               : ↵
Enter K start[,end[,inc]]: all↵
                               : ↵
LINES, HIDDEN_LINES, POINTS, or SHADED_SURFACE? line↵
Enter color:  rgb  .6  .6  .6↵
Enter line type: ↵
Enter line thickness factor: ↵
Symbol type? ↵
Enter symbol size factor: ↵
Wall  3,  grid  2
Enter I start[,end[,inc]]: ■

```

Figure 29. Wall 2, grid 2

PLOT3D automatically states the next wall of the grid you are working on, and asks the same questions. The wall specified in figure 29 will be grid lines extending straight down from the bottom of the fuselage.

Notice in figure 30 (facing page) that in wall 4 the J-vectors are specified from 34 to 2 rather than the other way around. This is important because walls are painted in order : objects painted last cover objects painted first. When you use the WALL command, the part of the wall that is farthest from the viewpoint should come first, and the farther walls should come before the closer walls. Grids are always drawn in order, that is, grid one, grid two, etc; but within each grid, the order of walls matters.

```

Wall 3, grid 2
Enter I start[,end[,incl]: all ↵
                        : ↵
Enter J start[,end[,incl]: 2 ↵
                        : ↵
Enter K start[,end[,incl]: all ↵
                        : ↵

LINES, HIDDEN_LINES, POINTS, or SHADED_SURFACE? line ↵
Enter color: rgb .6 .6 .6 ↵
Enter line type: ↵
Enter line thickness factor: ↵
Symbol type? ↵
Enter symbol size factor: ↵

Wall 4, grid 2
Enter I start[,end[,incl]: all -1 ↵
                        : ↵
Enter J start[,end[,incl]: 34 2 ↵
                        : ↵
Enter K start[,end[,incl]: 1 ↵
                        : ↵

LINES, HIDDEN_LINES, POINTS, or SHADED_SURFACE? shaded ↵
Enter color: white ↵
Enter shading pattern: ↵
Enter transparency, shininess, and
                        material coefficient: 0 0 0 ↵

Wall 5, grid 2
Enter I start[,end[,incl]:

```

Figure 30. Walls 3 and 4, grid 2, line and shaded surface.

When PLOT3D prompts for a wall you do not want (the fifth wall in grid two), press **<return >** and you will get back to the PLOT3D V3 prompt. Then use the **WALL** command again with the next grid you want.

In wall 4, grid 2, we specify `shaded_surface` instead of `line`. This will show the body of the airplane as a solid . The first three walls of grid two will show how the grid surrounds the body. (Grid one is not used in this tutorial).

The grid numbers are set in the grid design. Walls are whatever you ask for in your PLOT3D session.

The attributes for shaded surface are different from line attributes. Enter a color. Alternatives for shading pattern are not yet implemented, so press return. You will need a number between 0 and 1 for transparency; 0 is opaque. Shininess and material coefficient are not yet implemented; enter any single digit. In figure 31, the first zero is for transparency; the following two digits are meaningless but necessary.

```

Wall 5, grid 2
Enter I start[,end[,inc[]: ↵
PLOT3D U3: wall/grid=3 ↵
Wall 1, grid 3
Enter I start[,end[,inc[]: all ↵
                        : ↵
Enter J start[,end[,inc[]: 1 ↵
                        : ↵
Enter K start[,end[,inc[]: all ↵
                        : ↵
LINES,HIDDEN_LINES,POINTS,or SHADED_SURFACE? shaded ↵
Enter color: white ↵
Enter shading pattern: ↵
Enter transparency, shininess, and
                        material coefficient: 0 0 0 ↵
Wall 2, grid 3
Enter I start[,end[,inc[]: all ↵
                        : ↵
Enter J start[,end[,inc[]: 1 ↵
                        : ↵
Enter K start[,end[,inc[]: all ↵
                        : ↵
LINES,HIDDEN_LINES,POINTS,or SHADED_SURFACE? line ↵
Enter color: red ↵
Enter line type: ↵
Enter line thickness factor: ↵
Symbol type? ↵
Enter symbol size factor: ↵
Wall 3, grid 3
Enter I start[,end[,inc[]: ■

```

Figure 31. Walls 1 and 2, grid 3.

```

Wall 3, grid 3
Enter I start[,end[,inc]]: all ↵
                          : ↵
Enter J start[,end[,inc]]: 1 21 ↵
                          : ↵
Enter K start[,end[,inc]]: 1 ↵
                          : ↵
LINES, HIDDEN_LINES, POINTS, or SHADED_SURFACE? shaded ↵
Enter color: red ↵
Enter shading pattern: ↵
Enter transparency, shininess, and
                    material coefficient: 0 0 0 ↵

Wall 4, grid 3
Enter I start[,end[,inc]]: 3 64 4 ↵
                          : ↵
Enter J start[,end[,inc]]: all ↵
                          : ↵
Enter K start[,end[,inc]]: last ↵
                          : ↵
LINES, HIDDEN_LINES, POINTS, or SHADED_SURFACE? line ↵
Enter color: red ↵
Enter line type: ↵
Enter line thickness factor: ↵
Symbol type? ↵
Enter symbol size factor: ↵

Wall 5, grid 3
Enter I start[,end[,inc]]: ■

```

Figure 32. More walls ...shaded surface, grid 3.


```

Wall 5, grid 3
Enter I start[,end[,incl]: ↵
PLOT3D U3: wall/grid=4 ↵
Wall 1, grid 4
Enter I start[,end[,incl]: all ↵
                        : ↵
Enter J start[,end[,incl]: 1 ↵
                        : ↵
Enter K start[,end[,incl]: all ↵
                        : ↵
LINES, HIDDEN_LINES, POINTS, or SHADED_SURFACE? shaded ↵
Enter color: white ↵
Enter shading pattern: ↵
Enter transparency, shininess, and
                        material coefficient: 0 0 0 ↵

Wall 2, grid 4
Enter I start[,end[,incl]: all ↵
                        : ↵
Enter J start[,end[,incl]: 1 ↵
                        : ↵
Enter K start[,end[,incl]: all ↵
                        : ↵
LINES, HIDDEN_LINES, POINTS, or SHADED_SURFACE? line ↵
Enter color: green ↵
Enter line type: ↵
Enter line thickness factor: ↵
Symbol type? ↵
Enter symbol size factor: ↵

Wall 3, grid 4
Enter I start[,end[,incl]: ■

```

Figure 33. Walls 1 and 2, grid 4.

```

Wall 3, grid 4
Enter I start[,end[,incl]]: all ↵
                        : ↵
Enter J start[,end[,incl]]: 1 11 ↵
                        : ↵
Enter K start[,end[,incl]]: 1 ↵
                        : ↵
LINES, HIDDEN_LINES, POINTS, or SHADED_SURFACE? shaded ↵
Enter color: green ↵
Enter shading pattern: ↵
Enter transparency, shininess, and
                        material coefficient: 0 0 0 ↵

Wall 4, grid 4
Enter I start[,end[,incl]]: 3 54 4 ↵
                        : ↵
Enter J start[,end[,incl]]: all ↵
                        : ↵
Enter K start[,end[,incl]]: last ↵
                        : ↵
LINES, HIDDEN_LINES, POINTS, or SHADED_SURFACE? line ↵
Enter color: green ↵
Enter line type: ↵
Enter line thickness factor: ↵
Symbol type? ↵
Enter symbol size factor: ↵

Wall 5, grid 4
Enter I start[,end[,incl]]: ↵
PLOT3D U3: ■

```

Figure 34. Walls 3 and 4, grid 4.

3.7 Defining Plot Range — minmax 0 14 -5 5 -1 1

```

PLOT3D U3: minmax 0 14 -5 5 -1 1 ↵
PLOT3D U3: █
    
```

Figure 35. MINMAX

Use MINMAX to specify the size of the box you want to see. The first value is the minimum x coordinate, the second value is the maximum x. The next two are for y, and the last two are for z. PLOT3D "clips" to the minmax box. Thus, the command in figure 35, above, would show this much of the grids (figure 36):

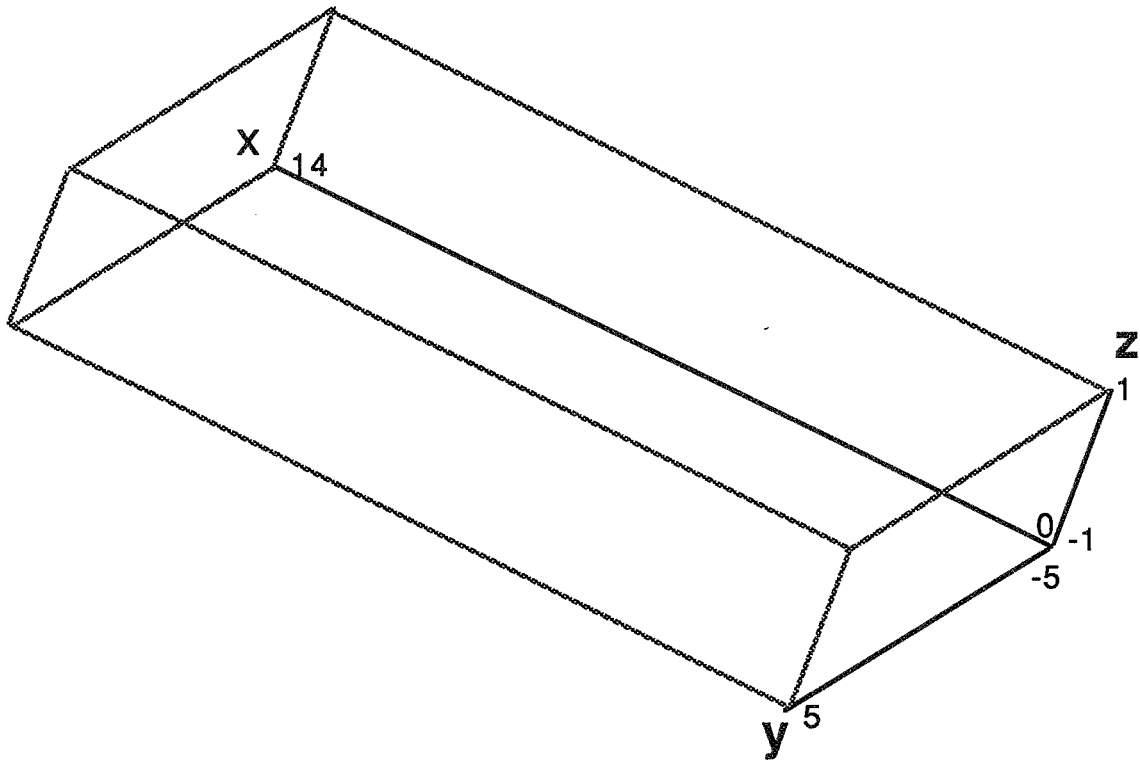


Figure 36. Minmax box.

On the IRIS, what is visible on the screen can be changed interactively with the mouse. You still should specify MINMAX because the center of the minmax box is always the center of rotation; an inappropriate minmax may cause your object to

rotate off the screen when moved with the mouse. The IRIS also uses MINMAX for z-buffering .

Use the SHOW command to see the current minmax (notice no slash):

```
show minmax
```

3.8 Plotting Viewpoint — vpoint/angle 150 20 25

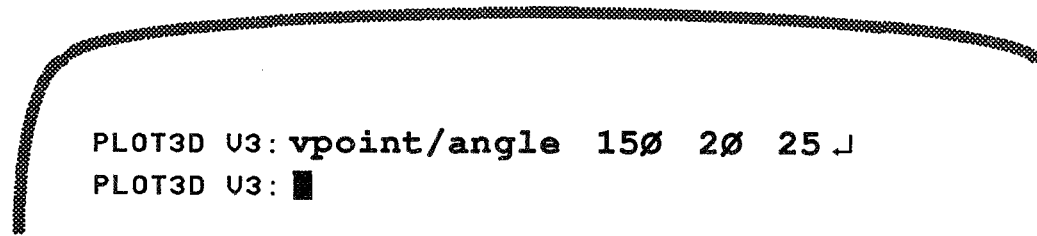


Figure 37. VPOINT/ANGLE.

In three dimensions, you are always looking into the center of the minmax box; VPOINT changes the location from which you are looking.

If you just use **vpoint**, you specify the x, y, and z coordinates of the viewpoint. If you use **vpoint/angle**, specify the polar coordinates of the viewpoint: phi, theta, and radius as shown in figure 37. If you imagine a circle drawn on a floor below the plot, phi is your position on the circle, in degrees counterclockwise from the right hand side. That is, changing *phi* takes you around the plot. *Theta* determines how high you are, with zero being level with the middle of the plot. *Radius* determines how far away you are (zoom). In the DISSPLA version, **vpoint/angle** does not zoom; use **minmax** to zoom (being careful to keep the same center if you want to keep the same orientation of the plot).

The default vpoint/angle (expressed in polar coordinates) is phi=150, theta=20, radius=20.

If you are working on an IRIS, the viewpoint is easily changed with the mouse. Use the SHOW command to see the current vpoint: **show vpoint**

3.9 Grid Chart

Here are two grid charts, one (figure 39) to show what we have done so far, and a blank one (figure 38) to copy for future work. The following abbreviations are used:

LHPS	lines, hidden_lines, points, or shaded_surface
ln tp	line type
ln thck	line thickness factor
sym	symbol type
sym size	symbol size factor
patern	shading pattern
transp	transparency, shininess, and material coefficient

PROJECT:		XYZ FILE:				Q FILE:			
Date:		GRID 1				GRID 2			
Wall:	1	2	3	4	1	2	3	4	
I									
I									
J									
J									
K									
K									
LHPS									
color									
ln tp									
ln thck									
sym									
sym sz									
pattern									
coeff.									
		GRID 3				GRID 4			
Wall:	1	2	3	4	1	2	3	4	
I									
I									
J									
J									
K									
K									
LHPS									
color									
ln tp									
ln thck									
sym									
sym sz									
pattern									
coeff.									
MINMAX:		/	/	/	/	/	/	/	
YPOINT/ANGLES:		/	/	/	/	/	/	/	
VIEW:		/	/	/	/	/	/	/	

Figure 38. Blank grid chart.

PROJECT: wbt				XYZ FILE: wbtg				Q FILE: wbttr			
Date:				GRID 1				GRID 2			
Wall:	1	2	3	4	1	2	3	4			
I					L	A	A	A	-1		
I											
J					2 34	34	2	34	2		
J											
K					A	A	A	1			
K											
LHPS					L	L	L	S			
color					.b .b .b	.b .b .b	.b .b .b	WHITE			
ln tp											
ln thck											
sym											
sym sz											
pattern											
transp									∅∅∅		
GRID 3				GRID 4							
Wall:	1	2	3	4	1	2	3	4			
I	A	A	A	3 64 4	A	A	A	3 54 4			
I											
J	1	1	1 21	A	1	1	1 11	A			
J											
K	A	A	1	L	A	A	1	L			
K											
LHPS	S	L	S	L	S	L	S	L			
color	WHITE	RED	RED	RED	WHITE	GREEN	GREEN	GREEN			
ln tp											
ln thck											
sym											
sym sz											
pattern											
transp	∅∅∅		∅∅∅		∅∅∅		∅∅∅				
MINMAX:	∅	14	-5	5	-1	1					
YPOINT/ANGLES:	150	20	25								
VIEW:											

Figure 39. Tutorial grid chart.

3.10 Plot — plot

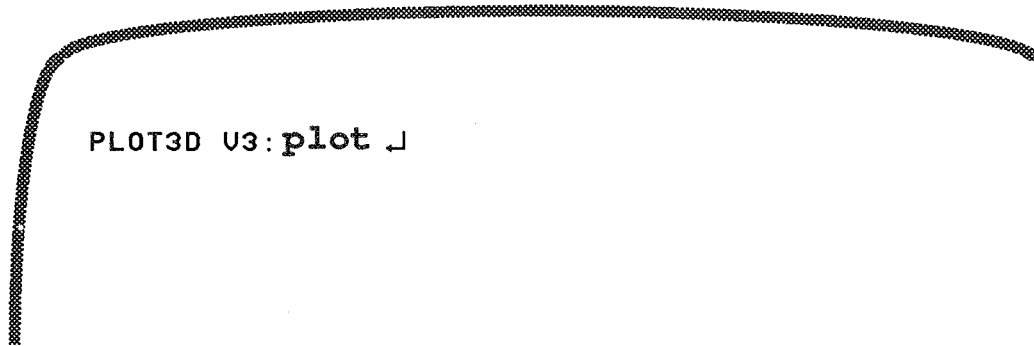


Figure 40. PLOT.

After you type **plot** (figure 40), or simply **p**, the screen should almost immediately go blank and then the word **GEOMETRY** should appear at the top of the screen. (A reader reports that this took almost a minute). Then your walls are plotted in the same order that you specified them. If your screen does not go blank and the cursor just hangs there, you have a problem. Perhaps you ignored warnings after the read command, and *wingbodytail* is not really your current working directory, and you didn't use a pathname. Usually, PLOT3D will not plot if it cannot open a journal file. To get out, press **^C** (and **^Y** in **DISSPLA** version) and **<break>** Return to **GO** (section 3.3). Do not collect \$200.

Figure 41 shows what you should see if you have entered everything correctly. Notice how shading and perspective are used. Refer back to Chapter 2, figure 14, for use of the mouse to zoom, drag, and rotate.

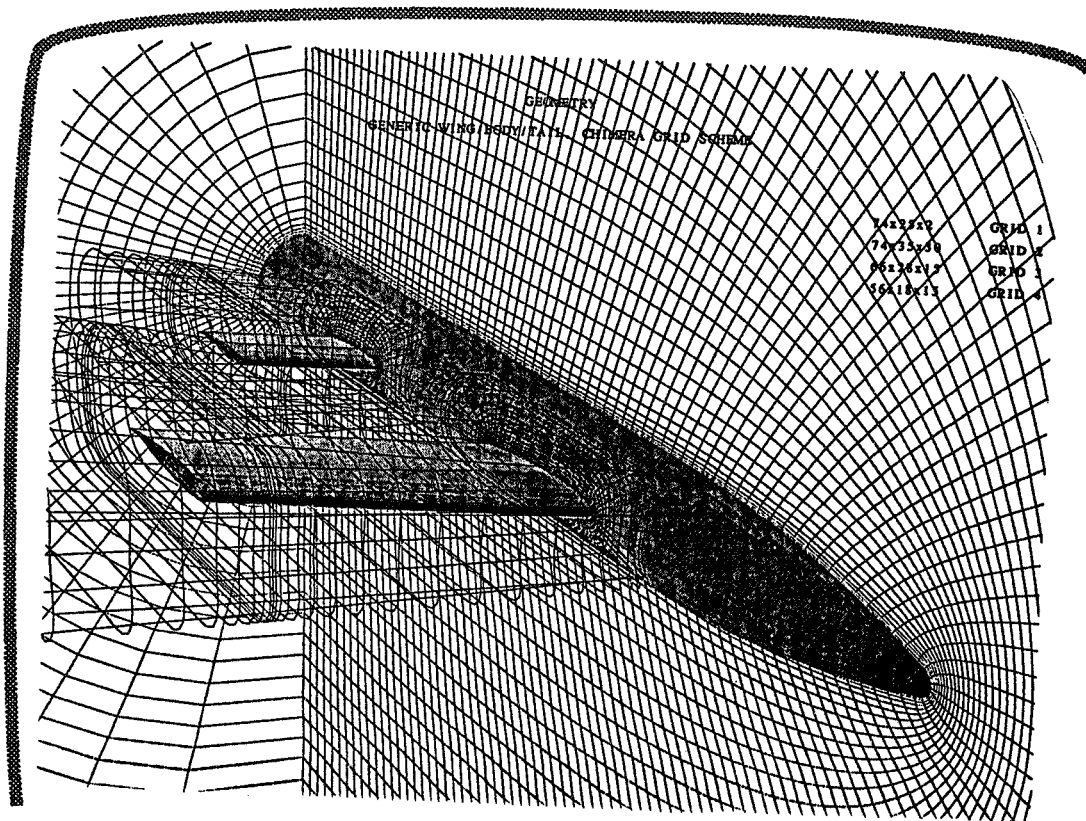


Figure 41. Wingbodytail geometry.

3.11 Return to PLOT3D Prompt — ↵ or

When you have finished looking at the plot, press <return> (↵) to get back to the PLOT3D prompt. On IRIS 4D's or in MEX, use (the delete key) to close the window, or <return> to get a new window..

3.12 Exit — exit or exit/save

If you need to end your session now, type `exit` or `e` as shown in figure 42; otherwise, just go ahead to the next chapter. Alternately, you can use `quit` or `q`. Or, to save your keyboard work, type `exit/save` (covered in Chapter 5).

ORIGINAL PAGE IS
OF POOR QUALITY

```
PLOT3D U3: exit ↵  
FORTRAN STOP  
system prompt> █
```

Figure 42. EXIT.

3.13 Conclusion of Chapter 3

These first commands, with their qualifiers and arguments, will show the geometry of the grid, including its descriptive title:

- read
- text
- wall
- minmax
- vpoint
- plot

Use of these commands is explained in this chapter and in Chapter 7.

Chapter 4. Functions

4.1 Summary

This chapter explains how to plot functions. First the grid functions (geometry, grid, and IBLANK) are discussed and tutorial examples are given for *wingbodytail*. WALLS and SUBSETS are compared; then scalar functions are illustrated. The difference between PLOT/CONTOUR and PLOT/FSURFACE is discussed. There are sections on vector plots, particle trace, and shock wave functions.

4.2 Types of Functions

The primary job of PLOT3D is to graph functions; that is, to graphically represent data such as the computed path of air over an aerospace vehicle.

PLOT3D functions are organized into five types: grid, scalar, vector, particle trace, and shock.

Once you specify a function number, it will be used for all of the following plots, or until you specify another. Before you specify any, PLOT3D defaults to function 0, which is geometry.

The command SHOW FUNCTION displays the name and number of the active function, plus pertinent data about your solutions for that function.

If you exited PLOT3D, you will have to read the data in again. See the previous chapter, sections 3.3 and 3.4. Make sure that your data set is in your current working directory, or use path name in quotes.


4.3 Grid Functions — Walls, Subsets, IBLANK Holes, and Grid checks

PLOT3D has six functions for viewing grids:

- Function 0 = walls alone (geometry). This is the default. We used it in Chapter 3 (we did not use the function command). It shows just the walls you have specified. *Walls are parts of the grid made visible with the wall command.*
- Function 1 = grids. This is what you use to see what your whole grid looks like. If you have a simple grid, you can use this function without specifying walls, and you will see your entire grid. The amount of available memory will limit the complexity of the grid you can display. Do not try to see all of wingbodytail; it is too much to plot. If you try it by mistake, use ^C or the break key. (You could use SUBSETS to look at grids).
- Function 2 = outline of IBLANK holes. This function shows you the part of the grid that has been turned off by an array in the xyz grid file. See "Blanking" in Chapter 7, section 7.2.
- Function 10* = 2d Crossing grid line check.
- Function 11* = Tetrahedron decomposition cell volume check.
- Function 12* = Tetrahedron decomposition grid crossing check.

*These functions are new in version 3.6. They are used for testing for errors in grids.

The best way to get a look at multiple grids is to specify walls that make an outline box of each grid, and use a different color for each one, as follows in fig. 43.

 Remember to press <return> when you want to clear the plot off the screen and return to the PLOT3D prompt. With the IRISwindow manager or the 4D, <return> only gets you a new window: press <delete> with the cursor in the graphics window to get rid of a plot.

```

PLOT3D U3: wall/grid=2↵
  Wall 1, grid 2
Enter I start[,end[,incl]]: 1↵
                               : last↵
                               : ↵
Enter J start[,end[,incl]]: 1↵
                               : last↵
                               : ↵
Enter K start[,end[,incl]]: 1↵
                               : last↵
                               : ↵

LINES, HIDDEN_LINES, POINTS, or SHADED_SURFACE? line↵
Enter color:red↵
Enter line type:↵
Enter line thickness factor:↵
Symbol type?↵
Enter symbol size factor:↵
  Wall 2, grid 1
Enter I start[,end[,incl]]:↵
PLOT3D U3: plot↵

```

Figure 43. Walls for grid function 1.

You should get a plot that looks like the following screen (fig. 44): an outline of grid two, which shows half a fuselage and the surrounding air space. This is function \emptyset , walls alone, the default.

Grid one is outside of grid two and is not used in this tutorial. If you want to see it, specify walls as above (fig. 43, except say `grid=1`), then use the mouse to zoom out. A plot is complete when the grid dimensions appear in the upper right corner; sometimes the plot is off-screen. You can always be sure a plot is complete when you hear the beep. If the terminal or workstation doesn't have a beeper, then one should watch closely. On a system without a mouse, use `minmax` to see more.

Grid three is the wing and grid four is the (horizontal) tail.

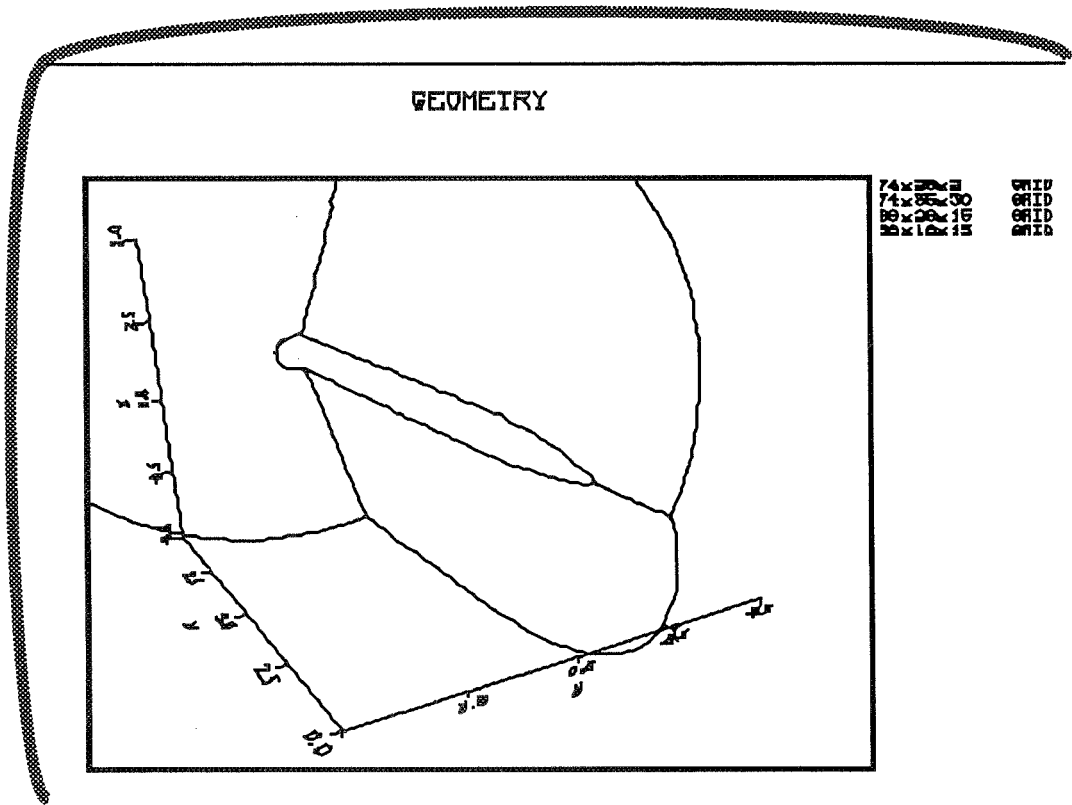


Figure 44. First and last walls of grid 2

Reminder: to get back to the PLOT3D prompt, press <return> or <enter>. On IRIS 4D's, press <enter> to get back to the prompt and <delete> to get rid of the plot. If you are using two monitors and two keyboards, such as on Chewbaka, use the <enter> and <delete> keys on the graphics keyboard.

(optional experiments)

**Repeat the previous pattern
with the other grids.**

Figure 45. Optional experiment

If you do the optional experiments, you will have more walls showing in your subsequent plots than are showing in the illustrations.

Another way to see multiple grids is to plot one at a time, using the `SUBSET` command. A *subset* is that part of a grid to which a function applies. For example, when we want to see the pressure on the top of the wing, we will designate the top surface of the wing as a subset, and the other grids as inactive subsets; the pressure function will be shown only on the top of the wing.

WALLS	SUBSETS
read statement resets to nothing	read statement resets to everything
parts of the grid made visible	part of grid to which function applies
prompts for attributes	defaults to no attributes


BOTH
use qualifier <code>/grid=gridnumber</code>
prompt for I, J, and K start and optional end, increment
remain until respecified

Figure 46. Walls and subsets comparison chart.

The parts of the grid specified with the `WALLS` command are always visible; walls alone can be plotted with function \emptyset , the default function. The parts of the grid specified with the `SUBSET` command do not become necessarily visible; they become active areas for the calculation of functions. But there are cases in which the subset grid lines become visible: when the function is a grid function, or when the `GRID` attribute is chosen for the `CONTOURS` command (see 4.6 and 4.7). Using subsets and function 1, we can see one grid at a time.

A word of caution: if you try to use function 1 with a multiple-grid data set, without using subsets, you are going to get much more than you want or more than the hardware can handle. Use `^C` to stop the plot.

We will eliminate grids one, two, and four so that we can plot all of grid three, which is the wing and some surrounding airspace.

Notice that the wall you specified from grid two is still plotting, even though grid two is now an inactive subset.  Subsets determine where functions are calculated; walls are not functions and, therefore, are not affected by subset commands. If you did some of the optional experiments, your plots will not look exactly like the following examples.

```
PLOT3D U3: subset/grid=1 ↵  
  Subset 1, grid 1  
  Enter I startI,endI,incI):↵  
PLOT3D U3: subset/grid=2 ↵  
  Subset 1, grid 2  
  Enter I startI,endI,incI):↵  
PLOT3D U3: subset/grid=4 ↵  
  Subset 1, grid 4  
  Enter I startI,endI,incI):↵  
PLOT3D U3: function 1 ↵  
PLOT3D U3: plot ↵
```

Figure 47. Eliminating subsets

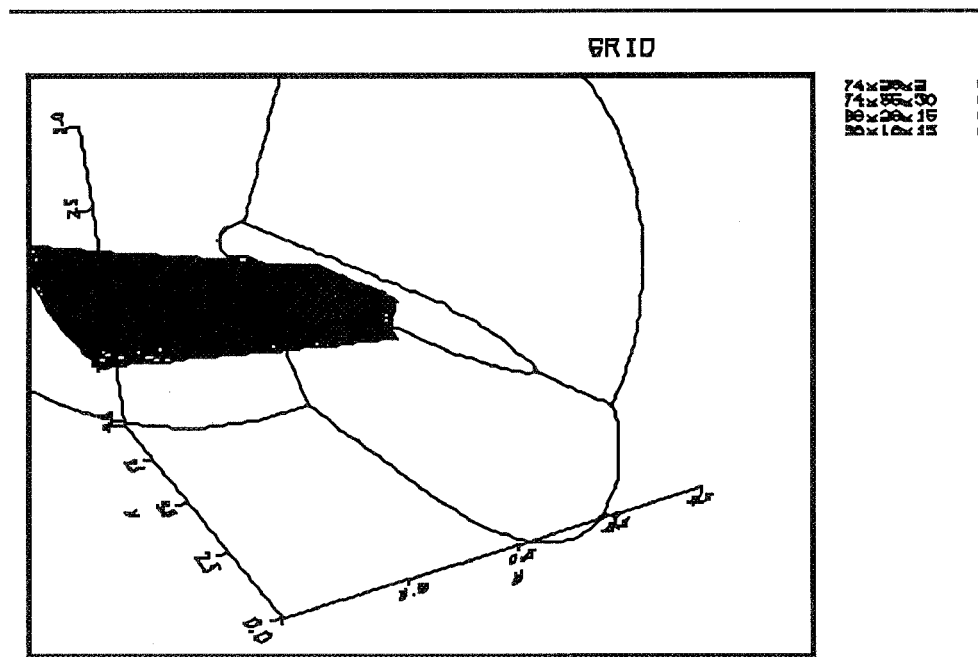



Figure 48. Subset = all of grid three; function = 1

(optional experiments)

**Repeat the previous pattern
with the other grids.**

Figure 49. More optional experiments.

Optional experiments: With the subset command, turn off all but one grid and plot with function 1, as in figure 47, but changing the numbers. You turn a grid off by typing `subset/grid=x` where `x` is the gridnumber, and then pressing `<return>` twice. When you start up PLOT3D, subsets are set to everything; in figure 47, we just turned several off, and left the other on. If you want to turn one back on, follow the examples in figure 50.

 If you experiment, turn grids one and three off before going on to the next exercise.

* * * * *

Holes can be defined in grids by using an "IBLANK " array. Holes are regions where the data on a grid is not considered valid, where perhaps another grid solution is more accurate. IBLANKing is an array in the xyz file; there are several options for how the holes work: 0 = always off, 1 = normal IBLANK, -n = space continues on grid n, 2 = solid wall boundaries. These options are set in the grid design. For more information on IBLANK , see Blanking section (7.2) in Chapter 7.

To see the IBLANK holes, use function two. But first we need to turn all the grids back on (except 1, which, on the VAX, does not plot right). See fig. 50. If you turned off grid three while you were experimenting, turn it on now, following the examples in Figure 50. If you are not sure, go ahead and turn it on again

```

PLOT3D U3:subset/grid=2 ↵
  Subset 1, grid 2
  Enter I start[,end[,incl]: all ↵
                               : ↵
  Enter J start[,end[,incl]: a ↵
                               : ↵
  Enter K start[,end[,incl]: a ↵
                               : ↵

  Subset 2, grid 2
  Enter I start[,end[,incl]: ↵

PLOT3D U3:su/gr=4 ↵
  Subset 1, grid 4
  Enter I start[,end[,incl]: a ↵
                               : ↵
  Enter J start[,end[,incl]: a ↵
                               : ↵
  Enter K start[,end[,incl]: a ↵
                               : ↵

  Subset 2, grid 4
  Enter I start[,end[,incl]: ↵
PLOT3D U3:fun 2 ↵
PLOT3D U3:p ↵

```

Figure 50. Subsets for function 2, IBLANK holes.

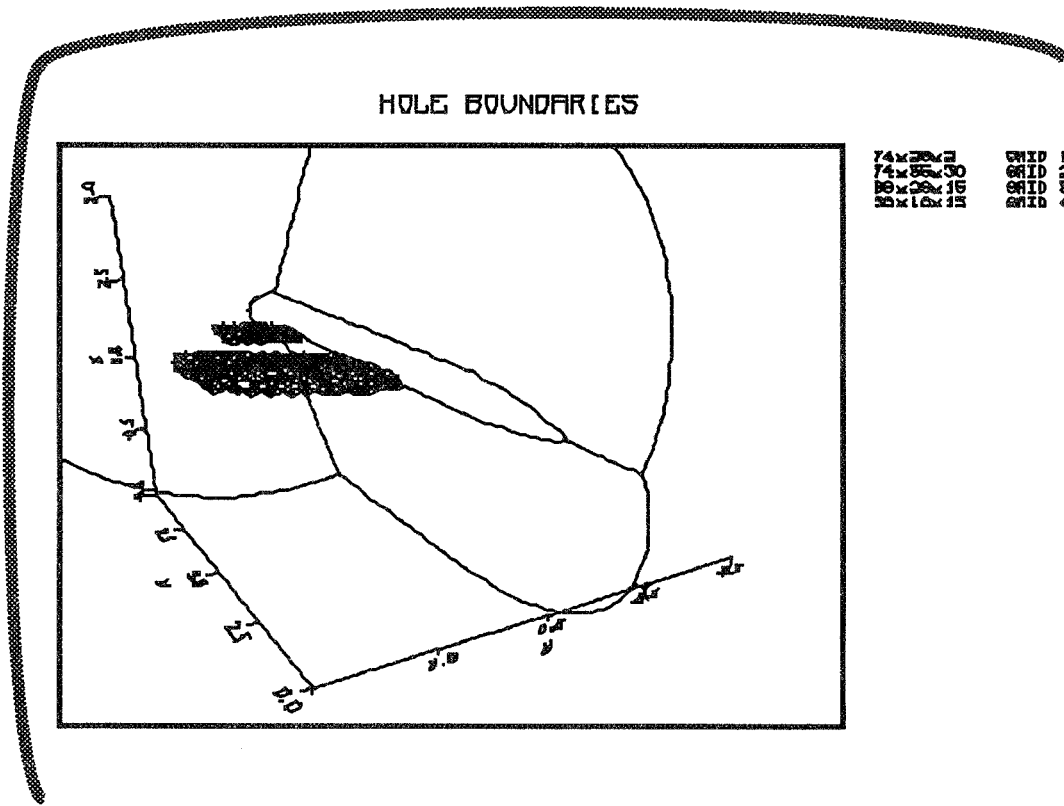


Figure 51. Plot of function 2 showing IBLANKing holes.

In figure 51, you can see that the wing and tail have been blanked out of the body grid.

Notice that the name of the function is always printed above the plot. In version 3.6, PLOT/NOTITLE turns off the printing of the function name; specified text is still printed.

4.4 Simple Walls

Before plotting functions, we will make walls that show a simple outline of the wing, body, and tail. Simple walls plot faster and move faster with the mouse.

In figure 52, we will use the fact that PLOT3D always offers another opportunity to enter I, J, or K start. We want the first and the last I, and the second and 34th J and the first K. Remember that you have to press an extra <return> when you have finished with each letter. If you make a mistake, just press <return> until you are offered another chance, or until you get back to the PLOT3D prompt, and enter the wall again. We only want one wall per grid, so press <return> when PLOT3D prompts for the next wall.

The wall from grid 2 will be a simple outline of the body of the plane.

```

PLOT3D U3: wall/gr=2 ↵
  Wall 1, grid 2
  Enter I start[,end[,inc]]: 1 ↵
                               : last ↵
                               : ↵
  Enter J start[,end[,inc]]: 2 ↵
                               : 34 ↵
                               : ↵
  Enter K start[,end[,inc]]: 1 ↵
                               : ↵

  LINES, HIDDEN_LINES, POINTS, or SHADED_SURFACE? line ↵
  Enter color: red ↵
  Enter line type: ↵
  Enter line thickness factor: ↵
  Symbol type? ↵
  Enter symbol size factor: ↵
  Wall 2, grid 2
  Enter I start[,end[,inc]]: ↵
PLOT3D U3: ■

```

Figure 52. Simple walls for grid 2.

```

PLOT3D U3: wa/gr=3 ↵
Wall 1, grid 3
Enter I start[,end[,incl]]: 3 ↵
                               : 33 ↵
                               : 64 ↵
                               : ↵
Enter J start[,end[,incl]]: 1 ↵
                               : 20 ↵
                               : 21 ↵
                               : ↵
Enter K start[,end[,incl]]: 1 ↵
                               : ↵

LINES, HIDDEN_LINES, POINTS, or SHADED_SURFACE? line ↵
Enter color: red ↵
Enter line type: ↵
Enter line thickness factor: ↵
Symbol type? ↵
Enter symbol size factor: ↵
Wall 2, grid 3
Enter I start[,end[,incl]]: ↵
PLOT3D U3: ■

```

Figure 53. Outline of the wing.

Continue as in figures 53 and 54. Chapter 5 will explain how to enter these values into a file, so that you do not have to retype them. (In brief: use `exit/save` instead of `exit`).

```

PLOT3D U3: wall/grid=4 ↵
  Wall 1, grid 4
  Enter I startI,endI,incI): 3 ↵
                               : 28 ↵
                               : 54 ↵
                               : ↵
  Enter J startJ,endJ,incJ): 1 ↵
                               : 10 ↵
                               : 11 ↵
                               : ↵
  Enter K startK,endK,incK): 1 ↵
                               : ↵

  LINES, HIDDEN_LINES, POINTS, or SHADED_SURFACE? L ↵
  Enter color: red ↵
  Enter line type: ↵
  Enter line thickness factor: ↵
  Symbol type? ↵
  Enter symbol size factor: ↵
  Wall 2, grid 4
  Enter I startI,endI,incI): ↵
PLOT3D U3: ■

```

Figure 54. Outline of the tail.

4.5 Scalar Functions

A scalar function represents a quantity, such as temperature or pressure, that is completely specified at each point by a single number on an appropriate scale, as opposed to vector functions, whose specifications include both magnitude and direction. PLOT3D function numbers 100 through 199 are for scalar functions.

PLOT3D offers 48 scalar functions. A complete list is on the reference card and in Chapter 7 under Functions. Here is a partial list (these indicate the major groups; for example, functions between 100 and 109 are types of density):

100 Density	154 Mach number
110 Pressure	160 X-momentum
120 Temperature	170 Entropy
130 Enthalpy	183 Vorticity magnitude
140 (Internal) energy	190 Shock

(see Appendix A for new functions)

4.6 Comparison of Contour and Function Surface or Carpet Plots

Scalar functions can be plotted in two ways: as *contour plots* or *function surface plots*. The difference between the two is a source of much confusion; this tutorial will take you through examples that illustrate the difference.

Contour plots are like a contour map; each line represents a value. Position of the line indicates location of the value. See section 4.7.

Function surface plots are like a graph shown against the walls of the model; the vertical position of the line indicates magnitude of the function, plotted against one or two of the axes. Subsets should be limited to the surface or one IJ, IK, or JK plane. The words "carpet" and "line" are also used to describe function surface plots. Function surface plots are discussed and illustrated in section 4.9. To help differentiate function surface plots from the WALL or CONTOURS *attribute* "surface" (which merely sets the method of drawing a plot, not the type of plot) the *command/qualifier* that creates carpet plots has been changed to PLOT/FSURFACE. The *attribute* "surface" means filled polygons. When you are plotting a scalar function, and you want to use other than the default attributes, use the CONTOURS command even if you are doing a surface plot. Then enter PLOT/CONTOUR (the default), or PLOT/FSURFACE.

PLOT/CONTOUR= command and qualifier to get a contour plot (default).

PLOT/FSURFACE= command and qualifier to get a carpet plot.

PLOT/SURFACE= command and qualifier to get a carpet plot before vers. 3.6.

PLOT/CARPET= alternate name to PLOT/FSURFACE for 3D.

PLOT/LINE= alternate name to PLOT/FSURFACE for 2D.

FSURFACE= command to make adjustments to a carpet plot.

CONTOURS= command to set attributes for contour OR carpet plot!

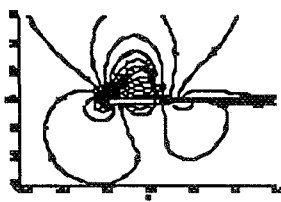
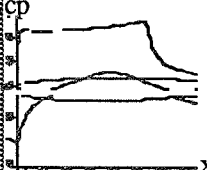
SCALAR FUNCTION POSSIBILITIES				
<p>ATTIBUTES*</p> <p>↓</p>	<p>PLOT/CONTOUR (D) MAP, FILLS ENTIRE SUBSET</p> <p>with 3d data and 3d subsets with 2d data or 3d slice</p>		<p>PLOT/FSURFACE or /CARPET or /LINE GRAPH, APART FROM SUBSET</p> <p>Subset=surface of model or IJ, IK, or JK plane.</p>	
<p>LINE (default)</p>	<p>Position of line indicates location of one value of function (isofunction), like a line on a contour map showing one elevation. Color indicates value. Each line is one color; e.g., a yellow line might indicate the location of Mach =.5.</p> 		<p>with 3d data and 3d subsets with 2d data or 3d slice</p>	<p>Position of line indicates value of function at subset. Like a graph of the function. Line changes color as value changes.</p> <p>"CARPET PLOT" "LINE PLOT"</p> <p>Undulating grid looks like wireframe carpet. Two spatial dimensions vs. a scalar function.</p> 
<p>SURFACE</p>	<p>As above, but 3d shell (filled polygons). Don't confuse with carpet.</p>	<p>Same as line attribute.</p>	<p>Solid carpet (filled polys) = graph.</p>	<p>Filled polys = graph.</p>
<p>GRID (usually set subsets = surface of model)</p>	<p>Grid colored by function value. In this case, line can have different colors.</p>		<p>Same as line</p>	
<p>COLOR CONTOURS (usually set subsets = surface of model, or slices)</p>	<p>Filled polygons colored by value of function. Like surface.</p>	<p>Filled polys throughout slice, colored by value.</p>	<p>Same as SURFACE attribute.</p>	<p>Graph made of filled polys in slice.</p>
<p>DOTS</p>	<p>Colored dots at each grid point indicate value of function.</p>		<p>Graph made of dots.</p>	
<p>* To be prompted for attributes, you must use the CONTOURS command, even if you do not want a contour plot. Then, use plot/contour (D) or plot/fsurface.</p> <p>The FSURFACE command is something else again, used to make adjustments. Use VIEW to set which spatial axes will be plotted vs. the function in function surface plots. See FSURFACE in the helpfile (Chapter 7) for surface info.</p>				

Figure 55. Scalar functions possibilities chart.

4.7 Scalar PLOT/CONTOUR and the CONTOURS Command

Contour lines connect points having equal values, as on a topographic map where each contour line represents one elevation. For example, in a pressure function plot, one line would connect the points with the same pressure; other lines would represent other pressures. *Colors are used to show the different values*; you can set magenta to be the lowest value, red the next, then yellow, green, cyan, then blue for the highest value. The default colormap is blue = lowest value, through cyan, green, yellow, and red, to magenta = highest. In general, it is good to have the hottest colors (red and magenta) represent the greatest magnitude of the function. On the screen, a numeric color scale to the left of the plot shows which color represents what value. For contour plots, there is never more than one color per line (except for grid line contours).

The specified function can be expressed in five different ways; these attribute choices are prompted when you use the CONTOURS command.

- 1) *LINES*, shown on a set of slices which have been specified with the subset command. See the color plates in the front of this book for an example (fig. 2, Mach number, and fig.5). Each line equals a set of constant function-value (isofunction); for example, one Mach number.
- 2) *SURFACE* (filled polygons) is equivalent to line except shown as a shell (filled polygons) in 3D. The shells are concentric, with one color/value per shell. In 2D, it looks the same as line. The filled polygons are in the "air", not on the surface of the object. Set subsets to a region (e.g., all of grid two) and select a few levels with CONTOURS. Please note that this is not the option to use for coloring the surface of your object; nor is it used here to get a "surface" or "carpet" plot.
- 3) *GRID*, grid lines colored by value.
- 4) *COLOR CONTOURS*, fills in the grid with polygons, not lines. This is the option to use for coloring a *surface*. Subsets are usually set to a plane (e.g., all I, all J, K=1). The surface of the shuttle orbiter, figure 2 in the color plates, is done with color contours. This attribute can also be used to fill a slice with solid color rather than lines.
- 5) *DOTS*.

You are prompted for these options after choosing the CONTOURS command. If you do not use the CONTOURS command, PLOT3D will default to lines. Even

if you are not planning to do a *contour* plot, use the CONTOURS command to specify attributes.

More levels give a more detailed picture of how the function is changing. You specify the number of levels on the contour command line, or use one of the qualifiers for more control.

These are the qualifiers:

- 1) */automatic* (default). If you ask for 40 levels, PLOT3D will look at your function, find the minimum and maximum, and pick increments to get roughly 40 levels.
- 2) */increment*. You specify the increment, but not number of levels.
- 3) */range*. Shows you the minimum and maximum of the function.
- 4) */manual*. Prompts you for minimum, maximum, and increment. After you specify one set, PLOT3D asks if you want another set. You can use this to divide the function into sections. Or, you can get a nice looking plot by putting contour lines on top of color contours, as in "Flow Past Conical Afterbody" in the color plates. Simply ask for color contours on the first set, and lines on the next.

4.8 Examples of Contour Plots

When you get back a new solution data set, and you do not know what is in it, there are recommended functions for getting pictures of the solution. We will try them out with *wingbodytail*. The first is scalar function 114, pressure coefficient, which is

$$\frac{\text{pressure} - \text{freestream pressure}}{\text{dynamic pressure}}$$

Like most of the PLOT3D functions, pressure coefficient is nondimensional; (Function 110 shows dimensional pressure).

```

PLOT3D U3: subset/grid=3↓

  Subset 1, grid 3
  Enter I start[,end[,inc]: a↓
                               :↓
  Enter J start[,end[,inc]: 5↓
                               :↓
  Enter K start[,end[,inc]: a↓
                               :↓

  Subset 2, grid 3
  Enter I start[,end[,inc]: a↓
                               :↓
  Enter J start[,end[,inc]: 10↓
                               :↓
  Enter K start[,end[,inc]: a↓
                               :↓

  Subset 3, grid 3
  Enter I start[,end[,inc]: a↓
                               :↓
  Enter J start[,end[,inc]: 15↓
                               :↓
  Enter K start[,end[,inc]: a↓
                               :↓

  Subset 4, grid 3
  Enter I start[,end[,inc]: ↓

PLOT3D U3: subset/grid=1↓
  Subset 1, grid 1
  Enter I start[,end[,inc]: ↓

PLOT3D U3: subset/grid=2↓
  Subset 1, grid 2
  Enter I start[,end[,inc]: ↓

PLOT3D U3: subset/grid=4↓
  Subset 1, grid 4
  Enter I start[,end[,inc]: ↓

PLOT3D U3: █


```

Figure 56. Subsets for slices on the wing

Specify subsets as in figure 56.

After specifying subset slices, we use the `FUNCTION` and `CONTOURS` commands, with the number of the function, and the number of contour levels (see

fig. 57). Our slices are oriented as IK planes. For scalar functions, the SHOW FUNCTION command gives minimum and maximum values.

 Margins in this manual necessitate moving some responses to new lines; you should be able to type them after the prompt. Use the hyphen (-) *only* if a line needs to be continued. If you need the hyphen, follow it with a <return>.

```

PLOT3D U3: function 114 ↵
PLOT3D U3: contours 40 ↵
  LINES, SURFACES, GRID_LINES, COLOR_CONTOURS, or DOTS? lines ↵
  Enter color sequence for color map: magenta red yellow- ↵
                                         green cyan blue ↵
  Enter line type: ↵
  Enter line thickness factor: ↵
  Symbol type? ↵
  Enter symbol size factor: ↵
  Plane orientations? ik ↵
PLOT3D U3: show function ↵
  Current function number is 114
  PRESSURE COEFFICIENT

  Minimum -1.125176 at point (47,15,1) in grid 3
  Maximum 1.129670 at point (32,1,1) in grid 3
PLOT3D U3: minmax 0 14 0 5 -1.25 1.25 ↵
PLOT3D U3 p ↵
  
```

Figure 57. FUNCTION, CONTOURS, and SHOW commands.

(You will probably get a different maximum than shown here).

PLOT/CONTOUR is the default, so PLOT or P will suffice (unless we had previously done a PLOT/FSURFACE).

Another good function to use, when examining a new solution data set, is stagnation pressure. We limited the subsets to three slices on the wing.(fig. 56). In general, contour plots are often limited to slices or surfaces.

```

PLOT3D U3: fun 115 ↵
PLOT3D U3: contour 50 ↵
  LINES, SURFACES, GRID_LINES, COLOR_CONTOURS, or DOTS?
                                                    color ↵

  Enter color sequence for color map: ↵
  Enter shading pattern: ↵
  Enter transparency, shininess, and material coefficient:
                                                    0 0 0 ↵

  Plane orientations? ik ↵
PLOT3D U3: p ↵

```

Figure 58. Function 115.

This time we chose `color_contours`, which plots polygons of color rather than lines. The following black and white plot (fig. 59) cannot express the effect, but it will show you where your slices should be. Notice that the slices are slightly curved, because the grid is curved.

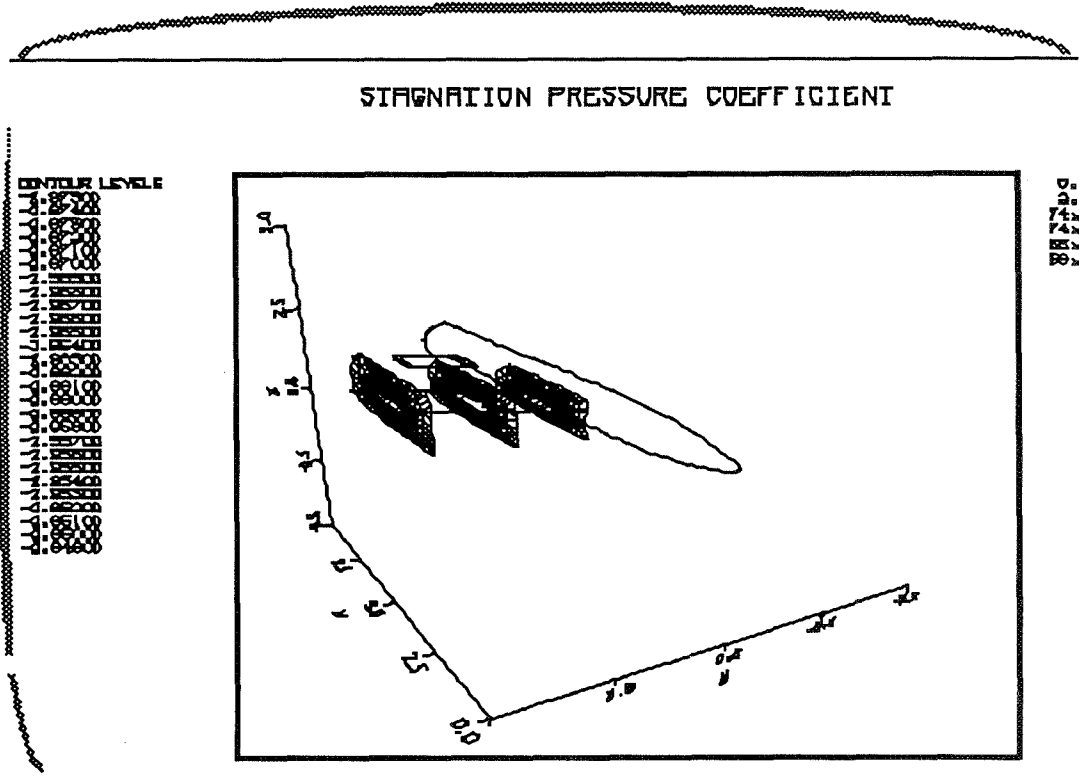


Figure 59. Example of stagnation pressure plotted with color contours on slices

Mach number, function 154, is also useful for examining a solution. For the subset (fig. 60), we use the symmetry plane (that plane about which the model is symmetrical) because we know that the flow there will be two-dimensional. Mach number is basically the speed of flow scaled to the local speed of sound, but we do not yet know the direction of flow for all the grid. On the symmetry plane, we eliminate the unknown.

```

PLOT3D U3: sub/gr=1 ↵
  Subset 1, grid 1
  Enter I start[,end[,incl]: ↵
PLOT3D U3: sub/gr=2 ↵
  Subset 1, grid 2
  Enter I start[,end[,incl]: a ↵
  : ↵
  Enter J start[,end[,incl]: 2 ↵
  : ↵
  Enter K start[,end[,incl]: a
  : ↵

  Subset 2, grid 2
  Enter I start[,end[,incl]: ↵
PLOT3D U3: sub/gr=3 ↵
  Subset 1, grid 3
  Enter I start[,end[,incl]: ↵
PLOT3D U3: sub/gr=4 ↵
  Subset 1, grid 4
  Enter I start[,end[,incl]: ↵
PLOT3D U3: ■

```

Figure 60. Subsets for symmetry plane.

```

PLOT3D V3: function 154 ↓
PLOT3D V3: contour 25 ↓
  LINES, SURFACES, GRID_LINES, COLOR_CONTOURS, or DOTS? L ↓
  Enter color sequence for color map: blue cyan green-
                                     yellow red magenta ↓
Enter line type: ↓
Enter line thickness factor: ↓
Symbol type? ↓
Enter symbol size factor: ↓
Plane orientations? ik ↓
PLOT3D V3: plot ↓
  
```

Figure 61. Mach number, function 154.

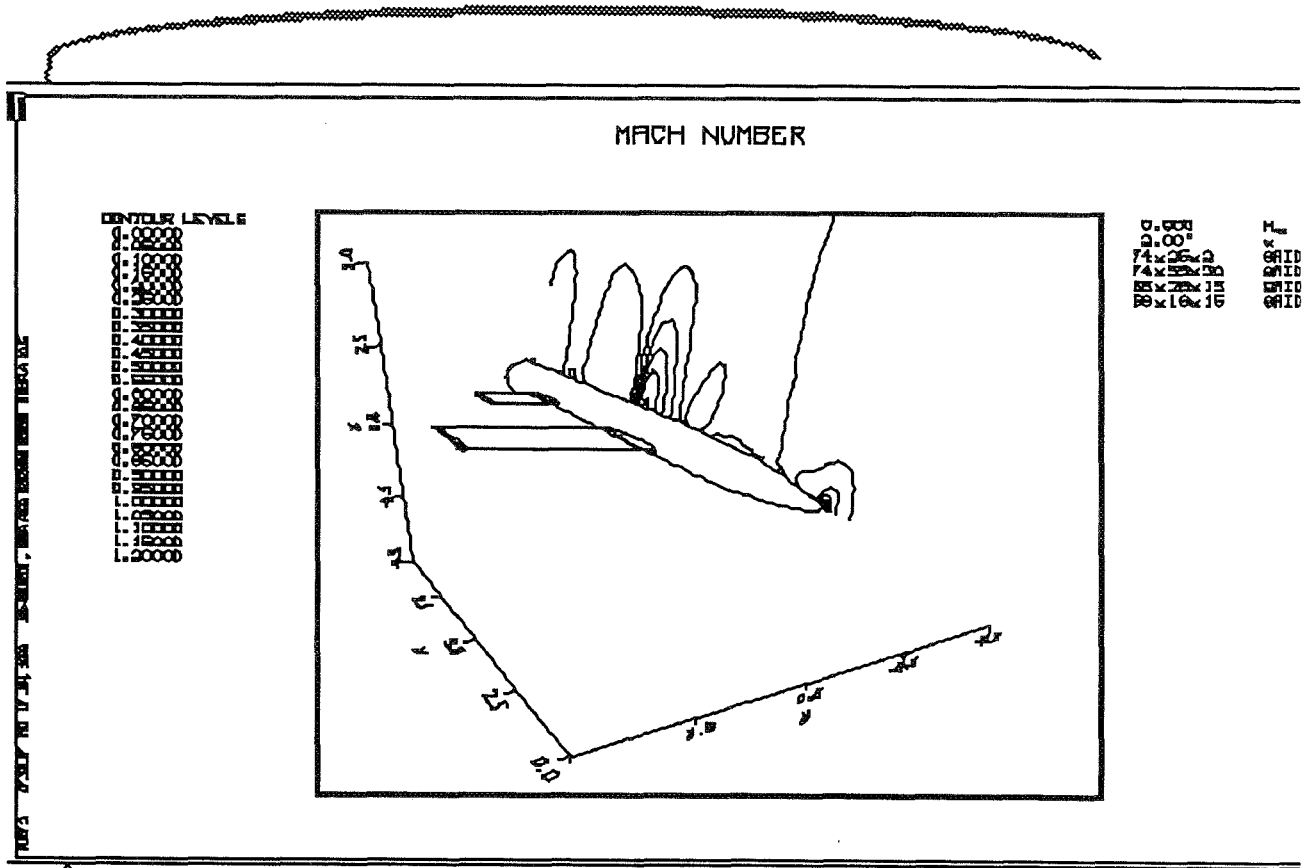


Figure 62. Plot of Mach number, function 154.

4.9 Scalar PLOT/FSURFACE or /CARPET or /LINE

Surface plots show a graph of the function, typically using the surface of the model as the subset. For 2D plots, the graph uses a specified xyz (default=x) dimension as the horizontal axes and the function as the vertical axis; for 3D, two of the xyz axes are used for the horizontal axis and the function value is plotted on the vertical axis. The VIEW command specifies which axes to use if other than default.

Another name for a 3D function surface plot is carpet plot, and a 2D function surface plot can be called a line plot. Be careful: this use of the word "surface" is easily confused with the WALL or CONTOURS *attribute* surface, which merely sets the method of drawing a plot, not the type of plot.

We want to see how much pressure is on the surface; we use subsets to specify the surfaces (figure 63). To simplify matters, we will plot the function only on the wing and tail. These subsets are lines which wrap around the surface of the wing and tail, so we will get graph lines representing the top and the bottom of the wing and tail.

Before version 3.6, PLOT/SURFACE was used instead of PLOT/FSURFACE.

```

PLOT3D U3: subset/grid=1 ↵
  Subset 1, grid 1
  Enter I start[,end[,inc[]:↵

PLOT3D U3: subset/grid=2 ↵
  Subset 1, grid 2
  Enter I start[,end[,inc[]:↵

PLOT3D U3: su/gr=3 ↵
  Subset 1, grid 3
  Enter I start[,end[,inc[]: 3 ↵
                               : 64 ↵
                               : ↵
  Enter J start[,end[,inc[]: 1 21 ↵
                               : ↵
  Enter K start[,end[,inc[]: 1 ↵
                               : ↵

  Subset 2, grid 3
  Enter I start[,end[,inc[]: ↵

PLOT3D U3: su/gr=4 ↵
  Subset 1, grid 4
  Enter I start[,end[,inc[]: 3 ↵
                               : 54 ↵
                               : ↵
  Enter J start[,end[,inc[]: 1 11 ↵
                               : ↵
  Enter K start[,end[,inc[]: 1 ↵
                               : ↵

  Subset 2, grid 4
  Enter I start[,end[,inc[]: ↵
PLOT3D U3: █
  
```

Figure 63. Subsets for rings around the surface of wing and tail.

Continue as in figure 64. The `FSURFACE` command is used to adjust the position of the walls and the scaling of the function axis. The `/FSURFACE` qualifier is used after `PLOT` to make this a surface plot (graph) rather than a contour plot (map). Versions before 3.6 use `/SURFACE`.

```

PLOT3D U3: fun 114 ↵
PLOT3D U3: contours 40 ↵
LINES, SURFACES, GRID_LINES, COLOR_CONTOURS, or DOTS? line ↵
Enter color sequence for color map: magenta red yellow
                                   green cyan blue ↵

Enter line type: ↵
Enter line thickness factor: ↵
Symbol type? ↵
Enter symbol size factor: ↵
Plane orientations? ↵

PLOT3D U3: show function ↵
    Current function number is 114
    PRESSURE COEFFICIENT
PLOT3D U3: minmax 0 14 0 5 1.25 -1.25 ↵
PLOT3D U3: fsurface/scale=1/walls_origin=1.25 ↵
PLOT3D U3: plot/axes/fsurface ↵

```

Figure 64. `PLOT/FSURFACE` and `FSURFACE`.

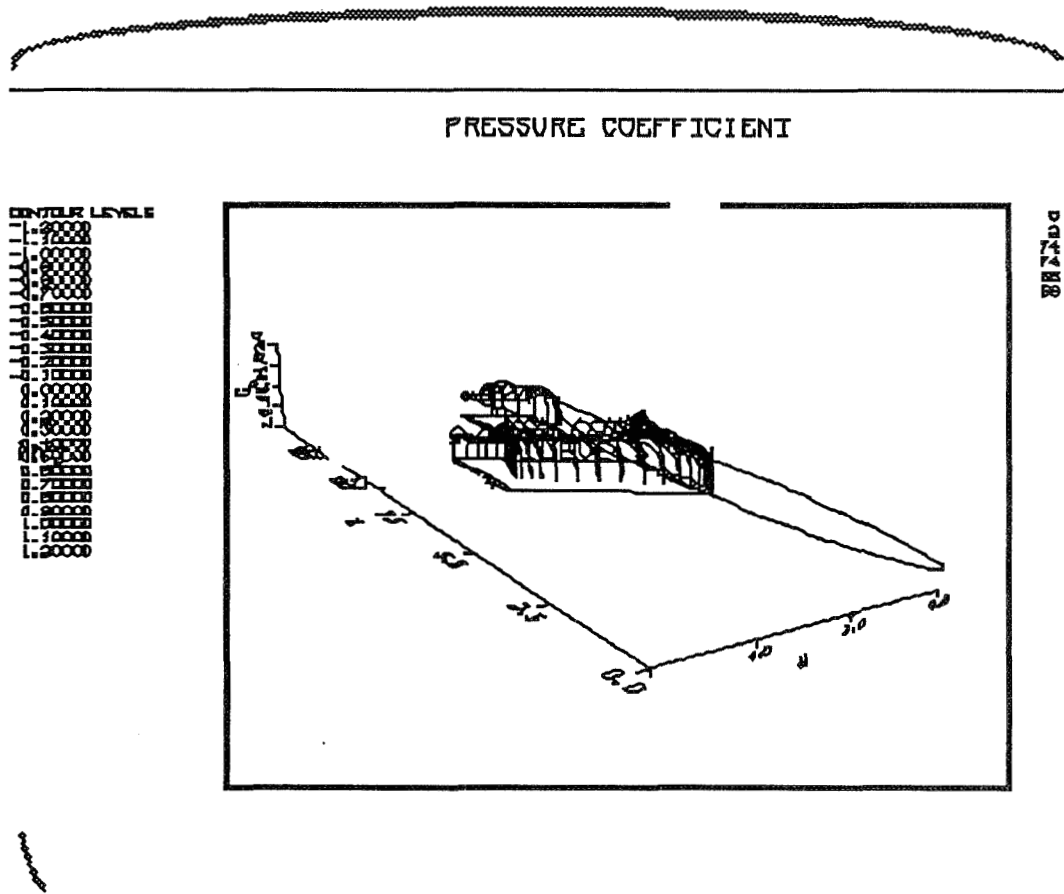


Figure 65. 3D function surface plot.

Figure 65 shows the *function surface* plot of function 114, the same plot you saw in the testdrive (fig. 15). If you zoom in and look at the lines, you will see that there are two lines per subset, because the subset goes all the way around the wing and tail, so the pressure coefficient is being graphed for both upper and lower surfaces. Pressure is shown inversely; a high point in the graph indicates low pressure (suction). The difference between the pressure coefficients for the top and bottom of the wing represents the overall lift .

Use function 1 if you want to see the subsets.

4.10 Vector Functions

Vector functions use line segments with arrow heads to show both direction and magnitude, as opposed to scalar functions that only show magnitude.

To illustrate vector functions, we will plot velocity vectors on the fuselage (grid 2) of *wingbodytail*.

As with any PLOT3D function, subsets must be defined. This example (fig. 66) eliminates grids 1, 3, and 4 from the subsets, and specifies the subset for grid 2 such that parts will be skipped. The start-end-inc prompt of the subset command is used to specify every sixth point along I and every third point along J. This is where the vectors will be calculated and drawn.

MINMAX and VPOINT are explained in Chapter 3, sections 3.7 and 3.8, respectively.

```

PLOT3D U3: function 200 ↵
PLOT3D U3: subsets/grid=1 ↵

Subset 1, grid 1
Enter I start[,end[,inc]]: ↵
PLOT3D U3: sub/grid=3 ↵

Subset 1, grid 3
Enter I start[,end[,inc]]: ↵
PLOT3D U3: sub/grid=4 ↵

Subset 1, grid 4
Enter I start[,end[,inc]]: ↵
PLOT3D U3: sub/gr=2 ↵

Subset 1, grid 2
Enter I start[,end[,inc]]: 1 last 6 ↵
                        : ↵
Enter J start[,end[,inc]]: 1 last 3 ↵
                        : ↵
Enter K start[,end[,inc]]: 1
                        : ↵

Subset 2, grid 2
Enter I start[,end[,inc]]: ↵

PLOT3D U3: vpoint/ang 90 0 30 ↵
PLOT3D U3: minmax 0 13 -5 5 -5 5 ↵
PLOT3D U3: plot ↵

```

Figure 66. Commands for vector plot.

Now (fig. 67) we will use the VECTOR command to override the default vector attributes. We will scale the vectors and arrowheads by .5.

```

PLOT3D U3: vector ↵
Enter color: ↵
Scale vector length with magnitude? [y/n] ↵
Vector length scale factor? .5 ↵
Arrowhead type LINES, FILLED, or NONE? ↵
Scale arrowhead size with vector length? [y/n] ↵
Arrowhead size scale factor? .5 ↵
Enter line type: ↵
Enter line thickness factor: ↵
Symbol type? ↵
Enter symbol size factor: ↵
PLOT3D U3: plot ↵
    
```

Figure 67. VECTOR command for attributes.

There are two types of scaling available under the VECTOR command:

- 1) Vector scaling : determines whether the vector length is scaled with the magnitude. Default is scaled.
- 2) Vector length scale factor : shrinks or expands all vectors compared to default size. We entered .5, which reduced the length of the vectors by half. This is the option to use when you are having trouble with the vectors all running together.

Using the qualifier /SCALAR_FUNCTION=function number, you can have the vectors colored by a scalar function. See Chapter 7, under VECTOR, for more information on qualifiers and attributes.

4.11 Particle Traces

(If you have access to a Cray, and you are only doing particle traces, and your traces may be complex, then RIP is the recommended program to use.)

Each q file used in PLOT3D represents one moment in time. Particle trace plots show the equivalent of a snapshot of neutral particles moving through a steady flow field. The RAKE command is used to say where the particles are "released." PLOT3D is designed for steady-state flow; the particle traces are streamlines.

Oil flow patterns, that is, particle traces one level up from the surface, are good for examining new data, but they take a lot of memory. We will do a very simple particle trace.

All particle traces are very disk-space-consuming. If your plot crashes, use ^C to get out. On the VAX, use ^Y after ^C. If it takes you all the way out of PLOT3D, PLOT3D will save your work in a .jou file (see Chapter 5). While at the system level, do some housekeeping in your account; clean out unnecessary files. *You can run PLOT3D on a Cray: see Chapter 11.*

In order to let the traces go anywhere in the flow field, we reset the subsets to everything. Follow figure 68.

```

PLOT3D U3: subset/grid=2

  Subset 1, grid 2
  Enter I start[,end[,inc]]: a↵
  : ↵
  Enter J start[,end[,inc]]: a↵
  : ↵
  Enter K start[,end[,inc]]: a↵
  : ↵

  Subset 2, grid 2
  Enter I start[,end[,inc]]: ↵
PLOT3D U3: sub/gr=3 ↵

  Subset 1, grid 3
  Enter I start[,end[,inc]]: a↵
  : ↵
  Enter J start[,end[,inc]]: a↵
  : ↵
  Enter K start[,end[,inc]]: a↵
  : ↵

  Subset 2, grid 3
  Enter I start[,end[,inc]]: ↵
PLOT3D U3: █

```

Figure 68. Subsets set to everything for particle traces.

In figure 69, we use the RAKE command to determine where the particle traces will be released. We specify three traces, released at three different J points, each a bit back from the nose (at I = 5), and one level up from the surface (K = 2). K = 2 is used to keep the particle traces from "hitting" the body and stopping.

(Wingbodytail uses Navier-Stokes flow solvers, resulting in no-flow, or "sticky", walls. Euler solvers would not do this).

We will use two qualifiers to the RAKE command: /WRITE= filename , which saves the particle trace calculations in a .par file , (and also gives additional memory space if the traces are too long for the allotted internal memory) and /SCALAR_FUNCTION=function number , which colors the traces by the values of the chosen function. Chapter 5 discusses how to use the .par files: in brief, they are run through another program called MRAKEVEC, which makes ARCGraph .vec files to use in GAS. MRAKEVEC offers a few options, such as coloring the traces by x, y, or z dimensions, which could color the traces by elevation. MRAKEVEC is used to prepare the particle traces to be animated (which makes them appear to flow) in GAS. The .par files also can be read into PLOT3D to save recalculating (use RAKE/READ = filename).


```

PLOT3D U3: sub/gr=4 ↓

  Subset 1, grid 4
  Enter I start[,end[,incl]: a ↓
                        : ↓
  Enter J start[,end[,incl]: a ↓
                        : ↓
  Enter K start[,end[,incl]: a ↓
                        : ↓

  Subset 2, grid 4
  Enter I start[,end[,incl]: ↓
PLOT3D U3: function 300 ↓
PLOT3D U3: rake/write=wbtrake/scalar_function=154 ↓

  Rake 1
  Grid number for rake points? 2 ↓
  Enter I start[,end[,incl]: 5 ↓
                        : ↓
  Enter J start[,end[,incl]: 10 ↓
                        : 20 ↓
                        : 30 ↓
                        : ↓
  Enter K start[,end[,incl]: 2 ↓
                        : ↓

  Enter IJ, IK, or JK to restrict particle traces,
  or <RETURN> for no restriction: ↓
  LINES or DOTS? L ↓
  Enter color: red ↓
  Enter line type: ↓
  Enter line thickness factor: ↓
  Symbol type? ↓
  Enter symbol size factor: ↓

  Rake 2
  Grid number for rake points? ↓
PLOT3D U3: vpoint/angle 150 20 25 ↓
PLOT3D U3: plot ↓

```

Figure 69. More subsets, and RAKE command.

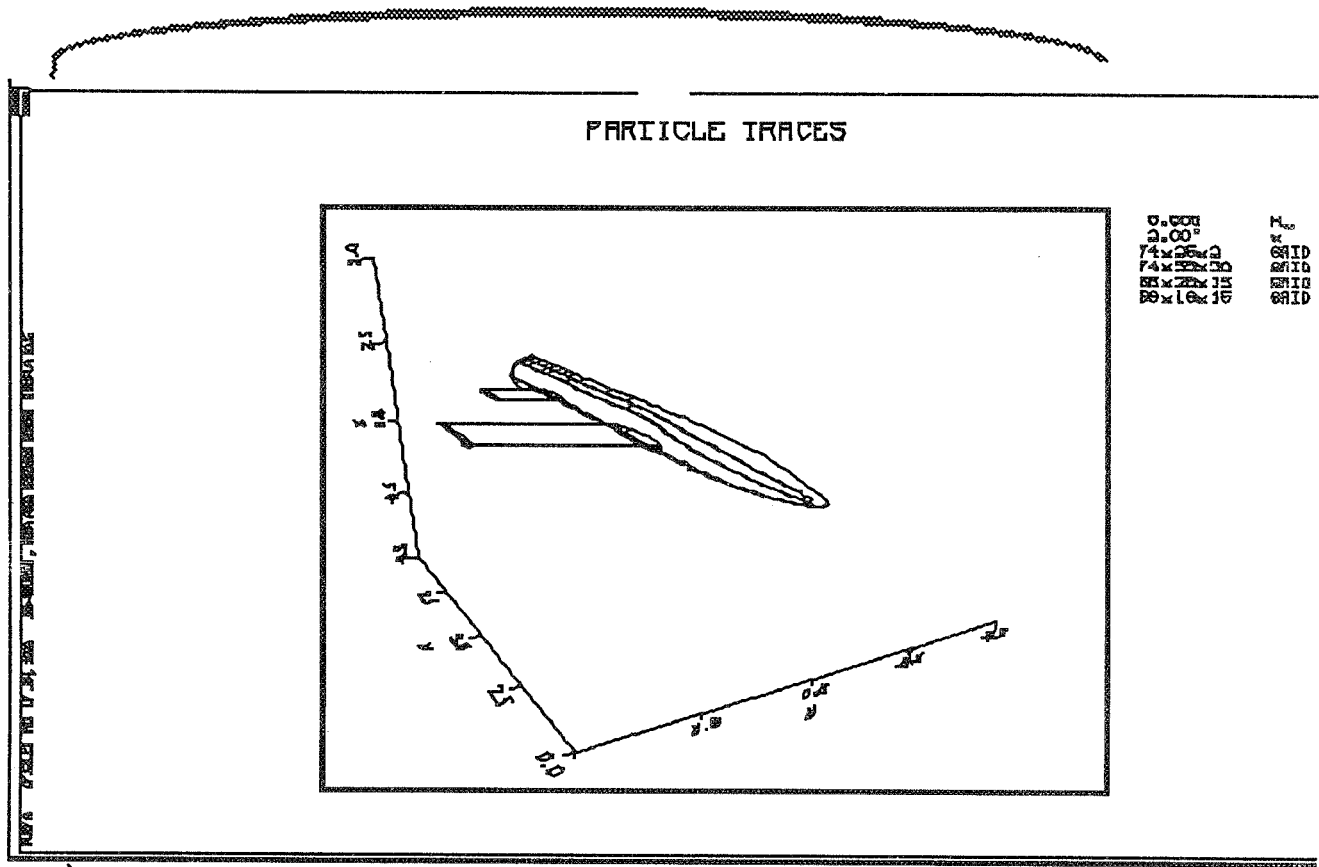


Figure 70. Particle trace plot.

ORIGINAL PAGE IS
OF POOR QUALITY

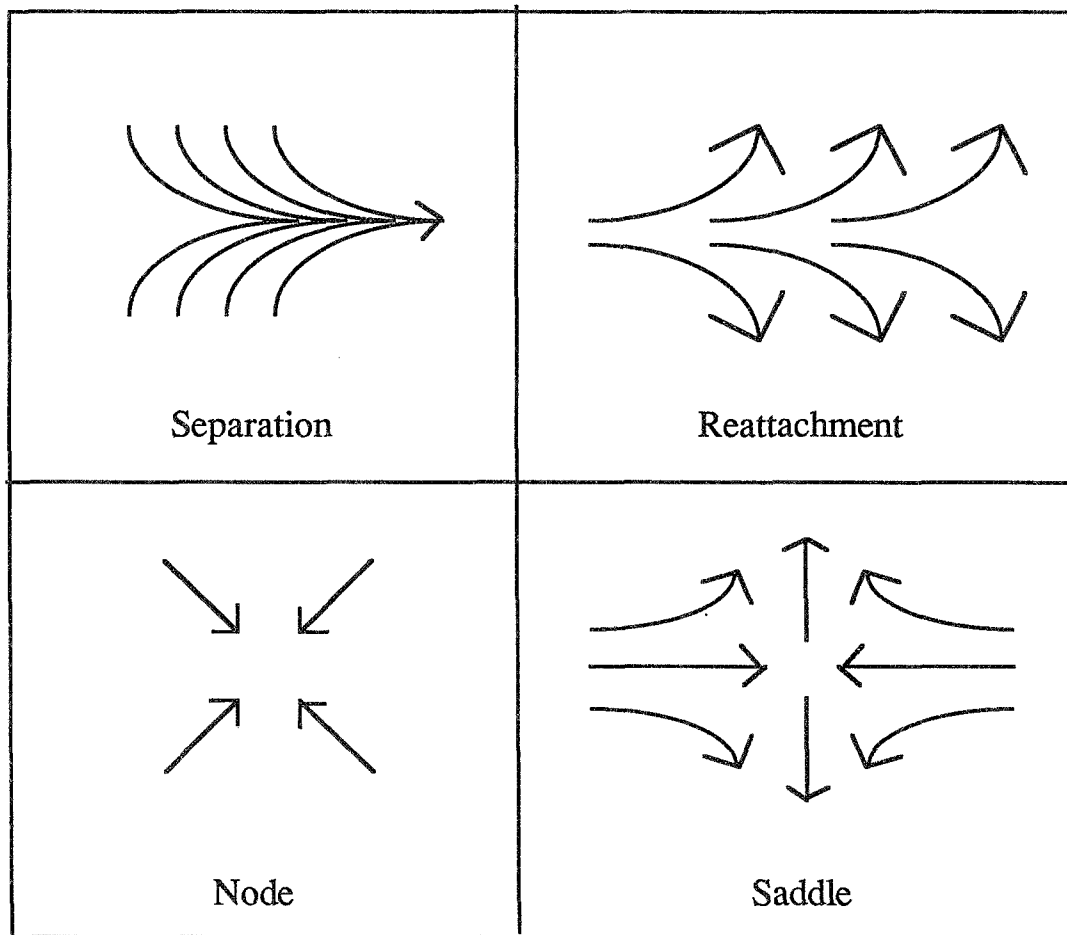
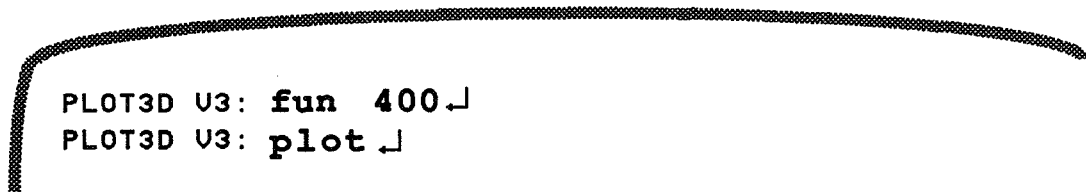


Figure 71. Oil flow particle trace patterns.

Figure 71 shows some patterns that you may see when you do oil flow particle traces, that is, where the flow is restricted to $K=1$ or $K=2$ ($K=1$ does not work with some solutions). Separation lines indicate where flow comes together and lifts off the surface. The opposite of separation is reattachment, where air blows down onto the surface. Air flows into or out of the same point at a "node," and a saddle is as illustrated.

4.12 Shock Wave

Function 400 indicates shock wave locations in the flow. This has been a problematic function. On IRIS 2xxx you won't get this plot due to a floating point exception; IRIS 4D should give this plot just fine.



```
PL0T3D U3: fun 400 ↵  
PL0T3D U3: plot ↵
```

Figure 72. Shock function specification.

Use the mouse to to see the shock function in 3D. Movement may be slow.

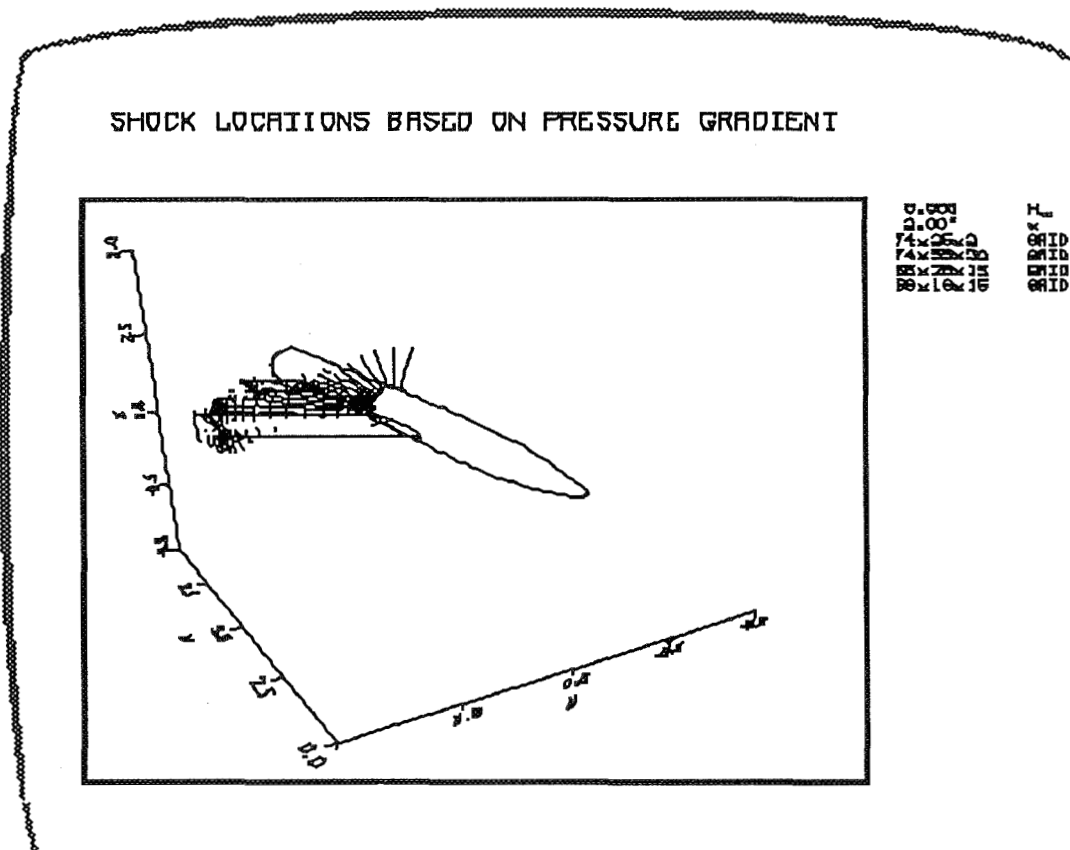


Figure 73. Shock function plot.

4.13 References to Other Chapters

See Chapter 7, Help File, under FUNCTION, FSURFACE, RAKE, VECTOR, CONTOURS, LIST, PLOT, SHOW, SUBSETS, and VIEW for more information on functions.

Formulae and fluid constants are in Chapter 7 at the end of the FUNCTIONS section, except for the new functions, which are in Appendix A.

Chapter 5 explains methods of saving PLOT3D work.

ORIGINAL PAGE IS
OF POOR QUALITY

4.14 Functions Conclusion

To plot functions, first define the WALLS of the object as you want them to show beneath the function. Simple outline walls are good for preliminary examination of new solutions; you might want to use solid walls for your final plots. Also select SUBSETS, the parts of the grid where the function will be calculated and plotted. Then use the FUNCTION command in the following form:

function <space> *function_number*

For example: **function 154**

or: **fun 154**

Functions are divided into five groups: grid, scalar, vector, particle trace, and shock. Scalar functions can be plotted as contour or surface plots. As types of plots, /CONTOUR and /FSURFACE are both qualifiers to the PLOT command. A contour plot usually has many lines throughout the subset. A surface plot graphs the value of the function versus a spatial dimension. The word CONTOURS is a command, used to specify attributes and ranges, and SURFACE is an attribute under the CONTOURS command. FSURFACE is a command and a qualifier.

After specifying a scalar function (numbers 100 to 190), use the CONTOURS command followed by the number of levels you want, or else followed by one of the qualifiers, /RANGE, /INCREMENT, or /MANUAL. Answer the prompts for attributes, or press <return> for defaults.

Set the minmax box (as in Chapter 3), then type PLOT/CONTOUR or PLOT/FSURFACE.

Recommended functions for examining new solutions are:

- 114 Pressure coefficient
- 115 Stagnation pressure coefficient
- 154 Mach number
- 300 Particle traces
- 400 Shock locations

Chapter 5. Saving Data for PLOT3D and GAS

5.1 Summary

This chapter will explain how to save your keyboarding work for use in future PLOT3D sessions, how to use command and PLOT/SCRIPT files, how to save particle trace files, how to use the LIST command to save calculations or reformat data, and how to make files to use for animation sequences in GAS. Saving screen images for making hard copies will be covered in Chapter 6.

These file types are discussed in this chapter: .com, .gra, .jou, .par, .vec.

5.2 Saving Keyboard Work — .jou Files

If your Chapter 4 session of PLOT3D is still running, you have something to save. If you have exited PLOT3D, re-enter by typing `plot3d` from the system prompt. Use the **READ** command as in 3.4, and **WALLS** specifications as in 4.4.

Whenever you work with PLOT3D, a journal file is made of all your keyboard entries. If you have an abnormal exit (crash), the journal file is saved for you. On normal exit, you can save the journal file by using the **SAVE** qualifier with the **EXIT** command. The file will be named `plot3d.jou`.

Users often experiment with different specifications for a plot until they get what they want, then **EXIT/SAVE** as in figure 74. Assuming you are running PLOT3D and have done some work, for example, the **READ** and **WALL** commands, type `exit/save` from the PLOT3D prompt.

An example is given of how to rename (move) the .jou to a .com file for the next session. Command file names on UNIX systems *must* be changed from "plot3d.jou" before running PLOT3D again. The program opens a new file and this kills any previous .jou file. PLOT3D assumes ".com" if no extension is given on the @ line.

In figure 74, the "plot3d.jou" is the file just saved. There is a "wbt.com" file that came with the wingbodytail directory; we used it in the testdrive. We rename the .jou file to a file with the name "wbtout.com" because it gives an outline of

wingbodytail. If you are not using a UNIX environment, you will have to follow the conventions of your system.

UNIX

```

PLOT3D U3: exit/save ↵
Programmed STOP
chewbaka:/u/wk/guest/walatka/wingbodytail:56 > ls ↵
plot3d.jou refs.txt wbt.com wbtg.bin wbttr.bin wbttrake.par
chewbaka:/u/wk/guest/walatka/wingbodytail:57 >
                                mv plot3d.jou wbtout.com ↵
chewbaka:/u/wk/guest/walatka/wingbodytail:58 > ls ↵
refs.txt wbt.com wbtg.bin wbtout.com wbttr.bin wbttrake.par
chewbaka:/u/wk/guest/walatka/wingbodytail:59 > █
    
```

Figure 74. Exit/save to create .jou file then save it as a .com file

5.3 Command (Script/Batch) Files— .com

PLOT3D has the capability of reading command files, that is, files that contain the keyboard entries for a PLOT3D session. In general computer use, and sometimes in PLOT3D, these files are called "script" files, but in PLOT3D "script file" also has the meaning of a file to control the display. See section 5.5 for PLOT/SCRIPT. "Batch" is another common term for a command file.

You can create the file from scratch, or use a .jou file. When creating from scratch, do not put in system prompts. Do put in returns (but they will not show). Use exclamation points (!) for comments.

A sample command file for a very simple 2D plot is shown below:

```

! this is a complete command file
read/x=airfoilx/q=airfoilq
function 110
minmax -1 2 -1.5 1.5
plot/2d
    
```


Command files usually have .com extensions. The usual procedure is to rename a .jou file to a .com file, then edit it to retain only the parts that you want. Whether or not you can leave the "exit/save" command line in depends on the version you are using; sometimes, when PLOT3D is running a command file, any end-of-file message returns PLOT3D to the previous source of input; when the command files are finished, you will see the "New Line:" prompt, at which point you can go ahead with keyboard input.

Our journal/command file was saved (figure 74) after entering the READ statement and the WALLS for the simple outline of wingbodytail.

Now run your command file as shown in figure 75.

```

chewbaka: /u/wk/guest/walotka/wingbodytail: 59 > plot3d␣
                PLOT3D Version 3.6      4 May 89
***An interactive graphics program for 2D and 3D CFD datasets***

Type "HELP" for a list of commands and general information,
"HELP Release_notes 3.6" for info on new features, bug fixes, etc.

PLOT3D U3: @wbtout.com␣
New line:  read/mgrid/blank/nocheck/x=wbtg/q=wbtr
SUBSETS reset to everything.
WALLS reset to nothing.
RAKES reset to nothing.
  Reading XYZ DATA for grid 1, 74x25x2
  Reading XYZ DATA for grid 2, 74x35x30

```

Figure 75. Running a command file

When you use @ with a .com file, you will see PLOT3D reading through the commands. You can get out of the command file before it is done by pressing the <break> key. If the program crashes, use ^C to get out.

5.4 Initialization File — plot3dini.com

Whenever you start running PLOT3D, PLOT3D will look in the current directory for an initialization file before it begins. This is a good place to put anything you always use in PLOT3D, for example, minmax or a recurring READ statement. Create the file using a text editor and name the file **plot3dini.com**, or rename a .jou file.

5.5 PLOT/SCRIPT

In the IRIS version of PLOT3D, the word "script" has a special meaning. If you use the /SCRIPT qualifier after the PLOT command, further input will be interpreted as commands for controlling the display. This can be done from the keyboard or from a .com file. On the IRIS, this can be used for simple animation when you do not have access to GAS and the ABEKAS.

Appropriate commands are: CENTER, GO, HOLD, INTERACTIVE, STOP, and VPOINT. See PLOT/SCRIPT in Chapter 7 for more information.

5.6 Saving Particle Traces — .par Files

A lot of computation goes into tracing particles. To save redoing the effort, the computations can be saved in a .par file. A .par file is not graphical information; it contains, for each point on the trace, x, y, z, time, and real number I, J, K.

In section 4.11, we saved a .par file when we said

rake/write=wbtrake

Whenever you use function 300 or 301, PLOT3D starts computing the traces when you type **plot**. You save the calculations by using the /WRITE qualifier after the RAKE command. You name the file, and PLOT3D automatically gives it a .par extension.

In PLOT3D, you use the .par file by typing **rake/read=filename**. To prepare particle trace files for use in GAS, see the next section.

5.7 Converting .par files to .vec for GAS — MRAKEVEC

GAS is a graphics animation and enhancement program developed by the Workstation Applications Office at NASA Ames. GAS is used to view, combine, enhance, title, and animate 3D or 2D scenes. Output from GAS is used to make films or video tapes. GAS is distributed for IRIS 2xxx/3xxx by COSMIC: for ordering information, telephone (404) 542-3265 or write to

COSMIC

University of Georgia

Athens, GA 30602

For the IRIS 4D and Personal computers, GAS requests should be sent to

NAS users should contact Sterling Software at (415) 964-9900, except for NAS users, who should contact Pat Elson at (415) 604-4463 or:

Pat Elson
Mail Stop 258-2
NASA Ames
Moffett Field CA 94035-1000

MRAKEVEC is a program for converting .par files to .vec files for GAS. (On NAS wks, look under /usr/local/bin, or, on tapes: plot3d/3.6b/unix+iris2/mrake/). A .vec file is a subset of .gra files (see next section).

This is the procedure:

- 1) Create a .par file by using `rake/write=filename`, followed by `plot` (because the calculation are made when you give the plot command) while running PLOT3D.
- 2) Run MRAKEVEC, using the .par file as input. Create .vec (graphic)file.
- 3) Run GAS to create an animation sequence.
- 4) Use the ABEKAS (with assistance) to record the animation to video tape.

MRAKEVEC allows you to divide the traces into segments for animation, so that the traces appear to flow from front to back. Other options include coloring by x, y, or z (elevation) dimension. Each solution file used by PLOT3D represents one moment in time; if you want to show a function over time, you must load in another solution file, or use PLOT4D (available for the IRIS 2xxx/3xxx, not 4D, from Merritt Smith 415-604-4493). GAS can combine particle traces made in PLOT3D with smooth-shaded models made in SURF.

5.8 Converting .com files to .gra for GAS— PLOT3X

PLOT3X is a special version of PLOT3D that plots to a .gra file instead of to the screen. A .gra file is a graphics file in the ARCGraph format. ARCGraph is the device-independent Ames Research Center Graphics library.

The usual procedure is to experiment in PLOT3D until you have what you want, then `exit/save`. Rename your .jou file to something.com, then run PLOT3X and `@` the .com file. PLOT3X will create a .gra file. A .gra file can have multiple frames, i.e., more than one plot.

The .gra files are not pixel files; they contain graphics primitives in the form of binary op-codes and data such as `move x y z`, `draw x y z`, and `ppoly`. These files represent a graphical interpretation of an image in terms of color, lines, and polygons. ARCGraph files can be loaded into GAS where they can be interactively viewed, rotated, zoomed, mirrored, animated, and recorded to digital disk, video tape, or 16-mm film.

You should still have a .com file from section 5.3. PLOT3X does not display the plots; you will see the cursor blinking while the functions are calculated, after the `plot` command.

For the following example, figure 76, the `wbtout.com` file was used.

```

chewbaka: /u/wk/guest/walatka/wingbodytail:55 >plot3x␣
                PLOT3D Version 3.6      4 May 89
***An interactive graphics program for 2D and 3D CFD datasets***

Type "HELP" for a list of commands and general information,
"HELP Release_notes 3.6" for info on new features, bug fixes, etc.

PLOT3D U3: @wbtout.com␣

                (you see file running)
                -----
Enter line thickness factor:  2
Symbol type?
Enter symbol size factor:

                Rake  3
                Suggested grid numbers to look for rake points?
PLOT3D U3:  plot
PLOT3D U3:  exit

chewbaka: /u/wk/quest/walatka/wingbodytail:56 >ls␣
refs.txt      wbtg.bin      wbtr.bin      wbtrake.par
wbt.com       wbtout.com    wbtr.gra

chewbaka: /u/wk/quest/walatka/wingbodytail: 57 >█
    
```

Figure 76. Using PLOT3X to create a .gra file for GAS

For making hard copies — videos and film — from GAS, see Chapter 6.

5.9 LISTing Function Data for PLOT3D

The LIST command in PLOT3D can be used to save xyz, q, or current function data. Output is directed to the screen (default) or to a file. An instance where you might want to try it is if you have made a 2d contour plot on the IRIS and you want labels on the contour lines showing values. The PLOT/LABELS qualifier is only

available on the VAX/VMS-DISSPLA version of PLOT3D. You could use LIST/FORMATTED to convert your files from IRIS binary (.bin) to VAX/VMS formatted (.fmt). The .fmt files should run on any system that runs PLOT3D.

MRAKEVEC uses function file listings for coloring particle traces by function values. PLOT3D version 3.6 implemented more uses for LIST files, as will version 2.1 of SURF; that is, SURF 2.1 will be able to read LISTed PLOT3D functions. The uses of LIST are still under development.

See LIST in Chapter 7 for more information.

5.10 Conclusion

From:	Ext:	Definition:	To:
file translator on grid or flow data or from LIST command in PLOT3D	.bin	binary data	PLOT3D or SURF on IRIS
system editor or .jou file	.com	command file to run PLOT3D (plot3dini.com = initialization)	PLOT3D or PLOT3X
see .bin	.dat	FORTTRAN unformatted data	PLOT3D or SURF
see .bin	.fmt	FORTTRAN formatted data	PLOT3D
PLOT3X SURF	.gra	ARCGraph file, 3D graphical information	GAS
PLOT3D exit/save	.jou	journal file; record of all keyboard entries	PLOT3D or PLOT3X
PLOT3D or RIP rake/write=filename	.par	particle trace calculation	PLOT3D rake/read=filename or MRAKEVEC
MRAKEVEC, SURF	.vec	colors, moves and draws, for lines (subset of .gra)	GAS

Figure 77. Filename extensions involved in saving data

The various ways of saving PLOT3D data are immensely useful. The .jou and .com files eliminate the need to repeat keyboard work. PLOT/SCRIPT files or keyboard commands can provide simple animation. .Par files save recalculating particle traces, and can be converted by MRAKEVEC into .vec files for GAS. Other types of PLOT3D work are prepared for GAS by running .com files through PLOT3X which produces .gra files. The PLOT3D LIST command is used to save xyz, q, or function data.

Chapter 6. Making Hard Copies

6.1 Summary

Chapter 5 was about saving PLOT3D data for use in subsequent sessions of PLOT3D or other programs (usually GAS). This chapter is about creating stills on paper or film, and, very briefly, about the equipment for recording GAS animations to video tape or film.

How you create stills (save screen images) depends on what machine you are using. Many of the image-saving methods are pixel dumps that is, dot-by-dot copies of what is on the screen. The following file types are involved in making stills: .dip, .qms, .sei, .ras, parm.plv, vectr1.plv, initscrt1.tmp, imagen.imp, .d48, and popfil.dat.

GAS .gra files are used for recording animations.

Postscript files are available from version 3.6 of PLOT3D.

6.2 IRIS Screen Dumps to Files

On the IRIS, there are several ways to save still screen images to a file:

- .ras files** While the picture is on the screen, press **r** (in versions before 3.6, press **p**), which will cause the colors of the pixels to be stored in a file. (When the plot is on the screen, PLOT3D is in a different mode, and does not confuse this command with the plot command.) The file will be given the name of your q file (if you used one), with a .ras extension. Ras is short for raster, the type of CRT screen used. The .ras files can have more than one frame, that is, more than one plot. The utility RASP is used to view or prepare the .ras file for printing. They can also be displayed in GAS (for adding titles).
- .sei files** While the picture is on the screen, press **s**. This will create a run-length encoded pixel dump in Seiko format suitable for printing on a Seiko color printer. The frame will be saved as

- filename.frame_number.sei. Each file holds only one frame. Used on Seiko or Tektronix color thermal printers. Can be displayed with `pixedit`.
- .qms files** Press `q` to save a file for the QMS 1200 black and white laser printers.
- .ps, .psd files** Version 3.6 of PLOT3D has the ability to make files in black and white PostScript format. For `.ps`, press `p` while the plot is on the screen. The `.ps` file is based on moves and draws. For `.psd`, press `d` (this creates a pixel-based file that can tie up the printer for hours). These files are used by any PostScript printer, including Apple Laserwriters, QMS 2400, New Imagen, Data Products 2600, and some of the LN03's. On IRIS 4D's, the `.ps` capability is iffy; on GT's, non-existent.
- .sei files from MEX** All the above will dump the active window. If you are using the window manager MEX and have more than one plot on the screen, press `f` to dump the entire screen to a Seiko file.

At NAS, the Seiko files are sent to the printer with the UNIX command `lpr (lpr -Pdevicename filename)`. See the printer manuals for further information.

6.3 Screen Dumps at NAS — Printers and Cameras

The above information applies. In the graphics lab, Room 125, which is near the front lobby in building 258 (NAS) at NASA Ames, there are personnel and equipment to help produce professional quality graphics. If you are a NAS user, call Vee Collet or Chris Gong for expert help with the cameras and other graphic equipment: (415) 604-4450, 604-4459, or 604-4422.

On the NAS workstations, the system command `man printers` will give the locations of the printers. There are various kinds of printers throughout the building, and Seiko color printers that accept `.sei` files. Seiko1 can make prints on paper or on transparencies for viewgraphs. Use `pixsave` to save a pixel dump of the screen or `pixps` for PostScript (see on-line UNIX man pages).

There is a Dunn camera hooked up to wk27 in the visiting scientists' room, and a Focus camera set up for Chewy. Users make prints with the Seiko until they get exactly what they want, then use the cameras for a high-quality reproduction, which can be a 35-mm slide or negative, or an 8x10 Polaroid print or transparency.

By special arrangement, 8x10 negatives can be made for printing by Imaging Technology. Cameras in other areas of the lab make 16-mm movies from the screen while you are running GAS or PLOT3D in PLOT/SCRIPT mode, with assistance, by appointment, from Vee or Chris.

A Sharp color scanner is in the area near Chewy. On the workstations, check the man pages for information on `pixedit`, `pixps`, `pixqms` and `pixscan`.

6.4 Videos at NAS

In the graphics lab, for NAS users, there is an ABEKAS digital disk recorder and other equipment to record GAS displays onto video tape. See Vee Collett and Chris Gong for help. See Chapter 5 for information on converting PLOT3D files for use by GAS.

6.5 DISSPLA Version and IRISRGB Screen Dumps

In general, in PLOT3D, the `plot` command can be qualified with the type of device to plot to. For example, you can say `plot/IRISRGB` if you have 24 or more bit planes on your IRIS (but not 4D). Then proceed as in 6.2.

In DISSPLA, this capability is more useful because you can plot to any device that the program is able to access. Get the plot you want on the screen, then type `plot/appropriate_qualifier`. The default is `plot/tektronix`, but you can specify `plot/qms` (QMS 1200 printer), `/LN03` or `/imagen` (laser printers), `plot/dicomed` (for film recorder), or `plot/compress` (for a device-independent file for POP). The output is put in a correctly formatted file which you then send to the printer, plotter, or film recorder. See PLOT in Chapter 7 for more information and more possibilities. While running PLOT3D, experiment until you get the plot you want, then press `<return>`, then type `plot/` with the appropriate qualifier.

6.6 Device Independent Plot — .dip Files

Device Independent Plot is a much used conversion routine for VAX/VMS, producing a file like `.gra`, except only for plotting two-dimensional plots. If you have a `.dip` file, you can run it through a utility to plot to a variety of devices. It is created by typing `plot/dip` in the DISSPLA version of PLOT3D. Theoretically, `.dip` file can be converted to PostScript.

6.7 Conclusion

Figure 78 shows the types of files involved in making stills. Not all printer/device/- file types are available on all versions of PLOT3D.

Section 6.2 discusses IRIS screen dumps; see 6.5 for DISSPLA.

From:	File.ext:	Definition:	Version	To:
PLOT3D plot/dicomed	.D48	file for film recorder	DISSPLA	D48
PLOT3D plot/compress	popfil.dat	device-independent file for POP	DISSPLA	POP
PLOT3D plot/imagen	imagen.imp	file for Imagen	DISSPLA	Imagen 8/300 laser printer
PLOT3D plot/ln03	initscrt1.tmp	file for DEC laser printers	DISSPLA	DEC LN01S LN03
PLOT3D plot/dip	.dip	Device-Independent- Plotter file	DISSPLA	graphic device
PLOT3D plot/versatec	parm.plv vectrl.plv	files for a Versatec plotter connected to VAX	DISSPLA	Versatec plotter
PLOT3D. while plot is on screen, press p (version 3.6 only)	.ps	PostScript screen save moves and draws	IRIS	PostScript laser printer
PLOT3D. while plot is on screen, press d (version 3.6 only)	.psd	PostScript pixel dump (slow to print)	IRIS	PostScript laser printer
PLOT3D. while plot is on screen, press q	.qms	screen save (one frame)	IRIS	QMS 1200 printer
PLOT3D plot/qms	.qms	screen save (one frame)	DISSPLA	QMS 1200 printer
PLOT3D. while plot is on screen, press s for graphics window or f for full MEX screen	.sei	screen save (one frame)	IRIS	Seiko thermal color printer or Tektronix thermal color printer
PLOT3D. while plot is on screen, press p (versions through 3.5) or r (ver.3.6)	.ras	pixel file screen save can hold multiple frames		RASP GAS

Figure 78. File types involved in making stills.

PLOT3D User's Manual

That's all for the tutorial.
Good-bye and good luck. Over and out.

Next two pages: Figure 79. Extra Reference Card, Functions, and
Figure 80. Extra Reference Card, Commands.

GRID FUNCTIONS

- 0 Walls alone (Geometry)
- 1 Grids
- 2 Outline of IBLANK holes
- 10* 2D Crossing grid line check
- 11* Tetrahedron decomposition cell volume check
- 12* Tetrahedron decomposition grid crossing check

SCALAR FUNCTIONS

- 100 Density (Q1)
- 101 Normalized density
- 102 Stagnation density
- 103 Normalized stagnation density
- 104* Log of normalized density
- 110 Pressure
- 111 Normalized pressure
- 112 Stagnation pressure
- 113 Normalized stagnation pressure
- 114 Pressure coefficient
- 115 Stagnation pressure coefficient
- 116 Pitot pressure
- 117 Pitot pressure ratio
- 118 Dynamic pressure
- 119* Log of normalized pressure
- 120 Temperature
- 121 Normalized temperature
- 122 Stagnation temperature
- 123 Normalized stagnation temperature
- 124* Log of normalized temperature
- 130 Enthalpy
- 131 Normalized enthalpy
- 132 Stagnation enthalpy
- 133 Normalized stagnation enthalpy
- 140 (Internal) energy
- 141 Normalized (internal) energy
- 142 Stagnation energy
- 143 Normalized stagnation energy
- 144 Kinetic energy
- 145 Normalized kinetic energy
- 150 u component of velocity
- 151 v component of velocity
- 152 w component of velocity
- 153 Velocity magnitude
- 154 Mach number
- 155 Speed of sound
- 156 Cross flow velocity

SCALAR FUNCTIONS continued

- 157 Normalized 2D stream function
- 158* Divergence of velocity
- 160 x-momentum (Q2)
- 161 y-momentum (Q3)
- 162 z-momentum (Q4)
- 163 Stagnation energy per unit volume (Q5)
- 170 Entropy
- 171 Entropy measure s1
- 180 x-component of vorticity
- 181 y-component of vorticity
- 182 z-component of vorticity
- 183 Vorticity magnitude
- 184 Swirl
- 185 Velocity \times vorticity magnitude
- 186* Helicity density
- 187* Relative helicity
- 188* Filtered relative helicity
- 190 Shock function based on pressure gradient
- 191* Filtered shock function (on pressure gradient)
- 192* Pressure gradient magnitude (on pressure gradient)
- 193* Density gradient magnitude (on pressure gradient)

VECTOR FUNCTIONS

- 200 Velocity
- 201 Vorticity
- 202 Momentum (Q2, Q3, Q4)
- 203 Perturbation velocity
- 204 Velocity \times vorticity
- 210 Pressure gradient
- 211* Density gradient

PARTICLE TRACE FUNCTIONS

- 300 Particle traces
- 301 Vortex lines

SHOCK WAVE FUNCTIONS

- 400 Shock locations based on pressure gradient
- 401* Filtered shock locations (on pressure gradient)

PLOT3D IRIS 2xxx/3xxx Screen Dumps or Draws

- | | | |
|-------|------|------------------------------------|
| Press | file | Save current screen display |
| d | .psd | PostScript pixel dump* (dangerous) |
| f | .sei | Seiko or Tek. (All mex windows) |
| p | .ps | PostScript moves and draws* |
| q | .qms | QMS QUIC format (lines only) |
| r | .ras | Raster dump (use p in v3.5) |
| s | .sei | For Seiko or Tektronix printers |
- (s and f work on IRIS 4Ds; d, p, r, q might work)

Reference Card

versions 3.5 and 3.6

Typical PLOT3D Command Sequence:

```

READ
TEXT
WALLS
MINMAX
VPOINT
SUBSET
FUNCTION
CONTOURS
PLOT
EXIT/SAVE
    
```

ATTRIBUTES of Walls

```

color Name or RGB components (see below)
line_thickness Real number multiplier
line_type Solid(D), dash, dot, chaindash,
chaindot, or none
material_coefficient Not implemented
shaded_surface Filled grid cells for faceted shading
shaded_pattern Solid(D) Others not implemented
shininess Real number 0 (dull) to 1 (shiny)
symbol_size Real number multiplier
symbol_type Integer representing symbol (See below)
transparency Real number 0 (opaque) to 1 (trans.)
    
```

COLORS - Defined by name (in color map order)

BLACK, WHITE, BLUE, CYAN, GREEN,
YELLOW, RED, MAGENTA

RGB COLORS - Set RGB with real numbers 0 to 1

RGB (red green blue) components (ex: RGB .2 .3 .4)

SPECIAL CHARACTERS

```

- Continue line
' Surround symbol to be translated
" Surround character string to be treated as single
argument, e.g., text or pathname
! Comment
@ Invoke .com file
EOF End of file (same as ^D in UNIX or ^Z in VMS)
$ Temporarily escape to system (in UNIX, use exit to
get back to PLOT3D, in VMS, use logout)
    
```

NASA/Ames Research Center
Mail Stop T45-2
Moffett Field, CA 94035-1000
(415) 604-4492

COMMAND - Description
 COMMAND/QUALIFIER=*argument*
 QUALIFIER[optional]
 /QUALIFIER (D) Default

CLEAR - Clear computed arrays
 CLEAR

CONTOURS - Set increment, attributes of function
 CONTOURS [*max number of levels*]
 CONTOURS/INC [*inc*] Increment
 CONTOURS/MANUAL [*start[,end,inc]*]
 /ATTRIBUTES (D) Prompt for attributes
 /AUTOMATIC (D) Increment number of contour levels
 /INCREMENT Set space between contour levels
 /LINEAR* (D) Linear (regular) interpolation
 /CUBIC* Use parametric cubic interpolation
 /MANUAL Multiple start, end, increment
 /NOATTRIBUTES Use default settings
 /RANGE Set min and max of current function

EXIT - Terminate session (Same as QUIT)
 EXIT
 /SAVE Save journal (.jou) file

FSURFACE - Set scale, origin (offset) of surface plots
 FSURFACE
 /CONTOUR Use contour rather than grid lines
 /GRID (D)
 /SCALE_FACTOR=AUTO (D)
 /SCALE_FACTOR=*scale* For function axis
 /WALLS_ORIGIN=AUTO (D)
 /WALLS_ORIGIN=*origin* Set function graph location

FUNCTION - Select function to be plotted, see back.
 FUNCTION [*function number*]

HELP - On-line information on keywords
 HELP [*keyword [keyword [keyword ...]]*]

LIST - Show or save current data
 LIST [*XYZfile* or *Qfile* or FUNCTION]
 /BINARY Save as binary (.bin) file
 /FORMATTED Save as formatted (.fmt) file
 /OUTPUT=*file* Redirect output to file or device
 /TEXT (D) Show data with heading
 /UNFORMATTED Save as unformatted (.dat) file

MAP - Map work array usage
 MAP

MINMAX - Set plot ranges
 MINMAX [*xmin,xmax,ymin,ymax[,zmin,zmax]*]
 MINMAX/INC [*xmin,xmax,xinc,ymin,ymax,yinc[,zmin,zmax,zinc]*]
 /INCREMENT Specify tick mark increment
 /X (D) /Y (D) /Z (D) Control axis to change

MINMAX continued
 /NOX /NOY /NOZ
 /XSCALE=*scale* Set scale factor (default is 1)
 /YSCALE=*scale* Set scale factor (default is 1)
 /ZSCALE=*scale* Set scale factor (default is 1)

PLOT - Display the plot
 PLOT
 /2D /3D (D) Set dimensions used
 /ADDITIONAL_TEXT* /NOADDITIONAL_TEXT*
 Turn legend on/off
 /AXES (D) /NOAXES Include/omit axes in plot
 /BACKGROUND=*color* Set background color
 /BAR* /NOBAR* Turn on/off contour level color bar
 /CARPET Same as /SURFACE
 /CONTOUR (D) Use many lines, each for a function value
 /FIGURE=(*areax, areay, charht*) /NOFIGURE (D)
 Use default plot size Control size of plot
 /FSURFACE* Replaces "SURFACE" graph function vs axis
 /FULLSCREEN Show enlarged plot without text
 /LABELS Show values on contour lines (2D DISSPLA only)
 /LINE Same as /SURFACE
 /METAFILE* Create Di3000 file
 /OVERLAY=*window number** Overlay plot (IRIS only)
 /SCRIPT /NOSCRIPT* Turn on/off script process
 /SURFACE Use 1 or 2 lines to graph function vs. axis
 /TITLE* /NOTITLE* Turn on/off function name
 /UP=*axis* Specify vertical axis
 See User's Manual for qualifiers related to hardware.

QUIT - Terminate session (Same as EXIT)
 QUIT
 /SAVE Save journal (.jou) file

RAKES - Specify start locations and attributes of streamlines
 RAKES
 /ADD Add rake to previous rake
 /ATTRIBUTES (D) Prompt for attributes
 /IJK (D) Start traces at I,J,K grid point
 /MAXPOINTS=*n* Control length of trace
 /NOATTRIBUTES Use default settings
 /NOSCALAR_FUNCTION (D) Use attribute color
 /READ=*file* Use previously saved particle trace (.par) file
 /+TIME (D) /-TIME /+TIME Show +, -, or both vectors
 /SCALAR_FUNCTION=*function number*
 Trace color is set by scalar function
 /WRITE=*file* Save in particle trace file (.par file)
 /XYZ Start particle trace at X, Y, Z coordinate

READ - Input data file
 READ
 /1D /2D /3D (D) Define 1D, 2D, or 3D file type

READ continued
 /BINARY (D IRIS only) For binary (.bin) file
 /CHECK (D) Check Q data validity
 /BLANK For file with IBLANK data
 /FORMATTED For formatted (.fmt) file
 /FUNCTION=*file* Not implemented
 /JACOBIAN Scaled Q variables
 /MDATASET For multiple data files. Not implemented
 /MGRID For files with multiple grids
 /NOBLANK (D) For files with no IBLANK data
 /NOCHECK Bypass Q data validity check
 /NOJACOBIAN (D)
 /PLANES For files in planes format
 /Q=*file* (D) For solution file
 /UNFORMATTED (D) For unformatted (.dat) file
 /WHOLE (D) For files in whole format
 /XYZ=*file* (D) For grid file

SHOW - Current status of some commands
 SHOW [*command*]

SUBSETS - Set active part of grid for function calculation
 SUBSETS
 /ADD Add to current list
 /ATTRIBUTES Prompt for attributes
 /GRID=*grid number* Specify which grid for subset
 /NOATTRIBUTES (D) Use default settings

TEXT - Add up to two lines of heading after title
 TEXT ["*line 1*" ["*line 2*"]]

VECTORS - Set vector characteristic
 VECTORS
 /ATTRIBUTES (D) Prompt for attributes
 /NOATTRIBUTES Use default settings
 /NOSCALAR_FUNCTION (D)
 /SCALAR_FUNCTION=*function number*
 Color by function number

VIEW - Specify axes for 2D or 3D surface plot
 VIEW [*view*]

VPOINT - Change viewpoint
 VPOINT [*x,y,z*]
 VPOINT/ANGLES [*phi,theta,radius*]
 /ANGLES Use polar coordinates
 /XYZ (D) Use Cartesian coordinates

WALLS - Specify part of grid to draw
 WALLS
 /ADD Add to current list of walls
 /ATTRIBUTES (D) Prompt for attributes
 /GRID=*grid number*
 /NOATTRIBUTES Use default settings

* New in PLOT3D v3.6

REFERENCE

Reference Summary

Chapter 7: Help file

Chapter 8: Data File Formats

Chapter 9: Changing PLOT3D and Host Systems

Chapter 10: Error Messages

Chapter 11: Examples of Other Command Files

Chapter 12: Filename Extensions and Command Sequence

Chapter 7. Keywords / Help File

By Pieter G. Buning

7.1 Summary

This is a verbatim reproduction of the PLOT3D help file from version 3.5. Section 7.2 gives information on the following topics: general information, journal file, initialization file, installation of PLOT3D, release notes, new features, bugs fixed, known problems, special characters, attributes, @ file, end-of-file, and \$system command.

Section 7.3 lists each command with its qualifiers and arguments and notes on use. These are the commands : CLEAR, CONTOURS, EXIT, FSURFACE, FUNCTION, HELP, LIST, MAP, MINMAX, PLOT, QUIT, RAKES, READ, SHOW, SUBSETS, TEXT, VECTORS, VIEW, VPOINT, and WALLS.

7.2 Information

General_information

In this documentation, information that can be included on a command line that is optional is enclosed in [square brackets]. Control characters are denoted by a preceding caret (^) (i.e., ^D for control-D).

The basic information needed for making a plot (after reading in the XYZ and Q data files) are: (1) the function to plot (FUNCTION command), and (2) the plot ranges, i.e., XMIN, XMAX, etc. (MINMAX command). A contour plot of Mach number can thus be made using the commands

```
READ/2D/XYZ=xyzfile/Q=qfile
FUNCTION 154
MINMAX -1 2 -1.5 1.5
PLOT/2D
```

PLOT3D User's Manual

Arguments can be delimited by either blanks or commas, and real numbers entered as integers (no decimal point) will be interpreted correctly.

The VT100 "green screen" and TEKTRONIX terminals are treated (and respond) identically.

Versions for the following machines/graphics libraries are available:

[see section 1.4 for current versions.]

If you have any suggestions or find any anomalies, please contact

Larry Pierce
MS T45-2
NASA Ames Research Center
Moffett Field, CA 94035
(415) 604-4492 or FTS 464-4492

Journal_file

During a PLOT3D session, a "journal file " named PLOT3D.JOU is kept of all commands entered to the program. In the case of an abnormal termination (i.e., crash), this file is saved. The file can also be saved by exiting from PLOT3D using EXIT/SAVE (or QUIT/SAVE).

Normally the journal file is deleted when the program terminates.

Initialization_file

When PLOT3D first starts up, it looks for a file "PLOT3DINI.COM " in the current directory. If the file is there, PLOT3D reads commands from it first, before taking input from the normal input device (the terminal).

Installation_of_PLOT3D

Installation notes for PLOT3D Version 3:

1. Check dimension of WORK in MAIN. It should be at least 13 times the maximum number of grid points. [Not necessary for Version 3.6.]
2. Change help library filename (HLBFIL) in SUBROUTINE HLPCMD.
3. COMMON blocks /WALL/ and /SUBS/ currently use ~400,000 words each, due to the maximum number of (multiple) grids being set to 50. To reduce this storage, change all occurrences of "MGRID=50" to "MGRID=5" (or whatever).
4. Look at SUBROUTINE PLTCMD for current plotting device names, SUBROUTINE PLTDEV for calls to routines which initialize devices.
5. Operating system-specific routines are at the end of the source file, and start with BLOCK DATA BCHKP. Check all these routines, but ones which will certainly need to be changed are:
 - BLOCK DATA BCHKP # characters/word.
 - SUBROUTINE FILNAM form of filenames on system, usually of the form "FILENAME.TYPE".
 - SUBROUTINE GETLUN sets available unit numbers.
 - LOGICAL FUNCTION INTERA interactive or not (see VMS version for what I really want to do).
 - CHARACTER*80 FUNCTION STDDEV sets standard I/O device names.
6. For the IRIS 2000 series terminals and workstations (GL2): subroutine IRIS contains a PARAMETER (IZNEAR=m,IZFAR=n) statement relating to z-buffering. IZNEAR is currently set to 0 and IZFAR to 32766 (for a 32 bitplane system), and should be set to 0 and 4095, respectively, for a 28 plane system. Z-buffering should not be attempted on a system with fewer than 28 bit planes: IZNEAR and IZFAR should be set to zero.

Release_notes

3.2

New features in version 3.2:

1. /CHECK/NOCHECK option on reading Q files.
2. /BINARY option (default) on reads for IRIS/Unix, as well as /FORMATTED and /UNFORMATTED.
3. Default file types are .FMT for formatted, .DAT for unformatted, and .BIN for binary.
4. Maximum number of grids increased from 10 to 50.
5. Faster initial search for RAKE/X particle points.
6. Use of C (binary) I/O for faster reading and writing of particle trace files on IRIS/Unix.
7. "P" for picture save (screen image dump) on IRIS version.
8. "Up-arrow" and "down-arrow" to increase or decrease mouse sensitivity on IRIS version.
9. Default contour colors changed from "WHITE" to "BLUE CYAN GREEN YELLOW RED MAGENTA".
10. Highlighting on surface-shading disabled.

3.3

New features in version 3.3:

1. 3D particle traces no longer require right-handed (i,j,k) system.
2. PLOT/SCRIPT option for movie scripts.
3. EXIT command (same as QUIT).
4. PLOT/IRISSINGLE and color maps for IRIS systems with fewer than 24 bit planes.
5. Special command-line characters explained in HELP.
6. Command files calling command files and returning fixed.
7. VPOINT/ANGLES fixed.

3.4

New in version 3.4:

New functions implemented:

Scalar functions:

- Cross-flow velocity
- Stagnation pressure coefficient
- Pitot pressure
- Pitot pressure ratio
- Dynamic pressure
- Normalized 2D stream function
- Swirl
- Speed of sound
- Shock function

Vector functions:

- Perturbation velocity
- Pressure gradient

Functions corrected:

- Normalized stagnation density
- Normalized stagnation temperature

Commands/options implemented:

- PLOT/2D - 2D plots implemented.
- PLOT/LABELS - works for 2D contour plots using DISPLA.
- PLOT/SURFACE (or /CARPET or /LINE) - for 2D line plots or 3D function surface plots.
- FSURFACE command - to define properties of line or function surface plot.
- VIEW command - for 2D plots or 3D function surface plots.
- MINMAX/XSCALE=scale/YSCALE=scale/ZSCALE=scale - to scale axes independently.

Release Notes continued

- PLOT/AXES/NOAXES - axes and axis labels added for IRIS too.
- PLOT/FIGURE=(areax,areay,char ht) - for figure-quality plots from DISSPLA.
- READ/PLANES - to read data one IJ plane per record
- Symbols and arrowheads have been implemented for the IRIS.
- CONTOUR grid-lines - for grid lines colored by contour value.

Defaults/initial values changed:

- SUBSETS/NOATTRIBUTES is now the default.
- Contour color map default is now "BLUE CYAN GREEN YELLOW RED MAGENTA" instead of "WHITE".
- MINMAX 0 10 -5 5 -5 5 is the initial MINMAX box.
- VPOINT/ANGLES 150 20 20 is the initial viewpoint for 3D plots.

IRIS features incorporated:

- PLOT3D runs under the MEX window manager (but does change the color map) (IRIS 2xxx only). You need a .mexrc file, with commands "reservebut 13", "bindfunc hogwhiledown 13", and "bindfunc menu 101" in it. See the Window Manager section of the IRIS User's Guide for more information.
- Keyboard attention interrupts (^C or <BREAK>) return PLOT3D to the command level prompt.
- \$system-command may work, depending on system swap space.
- Q key - dumps the screen to a file for printing on the QMS Lasergrafix 1200 printer (IRIS 2xxx only)
- Z key - toggles z-buffering mode (IRIS 2xxx only).
- 3,6,N keys - change to 30Hz, 60Hz, NTSC video out, respectively (IRIS 2xxx only).

IRIS bugs fixed:

- TBL6 6-plane color table fixed.
- IRIS 2xxx PXL screen dump and PLOT3DPIC fixed.

DISSPLA device drivers rearranged:

- PLOT/VER now produces files PARM.PLV and VECTR1.PLV for plotting on a VERSATEC connected to a VAX. At NASA Ames, see "\$HELP PLOT" on how to submit these files to be printed.
- PLOT/DIP produces a .DIP file for plotting on a VERSATEC connected to a PDP 11. At NASA Ames, a .DIP file can be converted to a .QMS file for printing on the QMS Lasergrafix printer with the "\$DIPQMS" command.
- PLOT/QMS produces a .QMS file directly, but this file may be very large.

VAX/VMS features restored:

- HELP, keyboard attention interrupts (^C), and error recovery now work under VMS 4.2.

ARCGraph GRAFIX interface written:

- A PLOT3D version (commonly known as PLOT3X) for NASA Ames device-independent graphics files has been written.

3.5

New_features

New commands/options:

- RAKE/MAXPOINTS=n option added to limit length of particle traces.
- RAKE/SCALAR_FUNCTION option added for coloring particle traces by a scalar function.
- PLOT/LN03 and PLOT/IMAGEN devices added to DISSPLA version.
- PLOT/UP=axis option added.
- PLOT/BACKGROUND=color option added {not in DISSPLA; background color must be set in GAS for GRAFIX}.
- SHOW PLOT command added.

Release Notes continued

- SHOW TEXT command added.
- SHOW FUNCTION also shows the (i,j,k) location of function min and max.
- LIST command added, supporting UNFORMATTED, FORMATTED, and BINARY XYZ and Q data file conversion.
- VECTOR attributes added to allow many different types of vector displays.
- READ prints an informational message when reading a file, including grid size.
- FSURFACE/NOCONTOUR option name changed to FSURFACE/GRID.

New features/versions:

- Particle tracing uses 2nd-order Runge-Kutta time -advance, still 5 steps per grid cell. Slow, and oil flow ain't always what we'd want.
- Removed XY,XZ,YZ restriction capability for particle tracing.
- CONTOUR plane orientation doesn't matter anymore if subset is just a plane.
- COLOR_CONTOURS now fill in between contour lines, instead of filling each grid cell with only one color.
- Contour GRID_LINES change color at contour levels, rather than having only one color for each grid line segment.
- SHADED_SURFACE walls that are just a single line will be drawn and filled as a single polygon. This can be used to fill in an airfoil outline, for instance.
- HIDDEN_LINE attribute enabled, as polygons filled with the background color, then outlined. (Ordering of the polygons is thus still important.)
- Polygon shininess attribute enabled. See "Attributes Shininess" for an explanation.
- Journal file capability added--file PLOT3D.JOU saved on abnormal exit or EXIT/SAVE.
- Auto-initialization file capability added--file PLOT3DINI.COM will be read and executed on startup of PLOT3D, if it is found in the current directory.

- HELP utility vastly improved.
- APOLLO version written, using AEGIS and GMR3D; it doesn't do everything, but it's workable.

New functions:

- Functions "Velocity x vorticity" and "Velocity x vorticity magnitude" added.
- Function "Entropy measure s1" added, a scaled, normalized version of exp(s).

IRIS improvements:

- IRIS screen saves rewritten to use run-length-encoding for smaller files, and to use the ARCGraph RAS file format. This format encodes RGB instead of color map entries, so that pictures can be restored in any mode. A new program RASP restores the screen saves, replacing the program PLOT3DPIC.
- Seiko color printer screen save added to the IRIS version. See PLOT.IRIS_interfaceSeiko_color_printer_screen_save".
- More colors available for 6, 8, and 10-plane plots on IRIS.
- Up to 8 simultaneous graphics windows under window manager on IRIS.
- Increased WORK array size from 2.5M to 3Mwords on IRIS. This will reduce maximum image complexity.
- Backface polygon removal implemented for IRIS, tied to "B" key during display.
- Unix SETENV symbols will be translated in PLOT3D when enclosed in 'single primes'.

DISSPLA improvements:

- 2D polygons now use shading in DISSPLA.

CRAY 2 improvements:

- The CRAY 2/IRIS version of PLOT3D now tries to start up PLOT3D_LCL on the workstation.

Release Notes continued

Bugs_fixed

Commands/options:

- READ/PLANES for XYZ file bug fixed (blanking vs. no blanking calls were reversed).
- READ/X="xfile"/Q="qfile" bug fixed (was getting misinterpreted).
- PLOT/AXES/NOAXES option fixed (wasn't always getting set right).

Features:

- Last contour level coming out white fixed, I hope for good (a roundoff problem).
- "RGB 0 0 0" color now treated correctly.
- Displaying IK planes in 2D plots should now work correctly (used J indices instead of K).

Functions:

- Function SWIRL fixed.
- ENTROPY calculation scaling changed to include specific heat cv.

IRIS interface:

- Color table problem with IRIS fixed (only red was displayed due to a FORTRAN compiler bug).
- Worked on near clipping problem with z-buffering on IRIS where z-buffering doesn't work when you're too close to the object).
- Device INPUTC changed to INPTCH on IRIS; commented out PARAMETER initialization of MODECH and INPTCH (these were due to changes in the GL2 FORTRAN include file "/usr/include/fgl.h").
- Mouse buttons should now work correctly under the window manager on the IRIS.
- NTSC display mode for IRIS now correctly maps to a subset of the screen.

- Better handling of keyboard interrupts during graphics initialization under MEX. (Would get message "no more gfports in system!!" twice when attempting another plot; would have to interrupt again. If a third plot was tried, system would hang.)

DISSPLA interface:

- MINMAX/xSCALE=scale for DISSPLA now works.

GRAFIX interface:

- 3D shading fixed (was incorrect if VPOINT was specified as (x,y,z)).
- Axis scaling from MINMAX/xSCALE=scale and PLOT/SURFACE fixed.



Version 3.6 Release notes: see Appendix A

Known_problems

Listed are known or potential problems with various versions of PLOT3D.

IRIS

- Z-buffering an image with dashed, dotted, chaindashed, or chaindotted linestyles will hang the IRIS.
- Changing from single- to double-buffer mode or vice-versa (such as when turning z-buffering on or off), when using the window manager and more than one plot is on the screen can cause the screen to be refreshed incorrectly. Move some windows slightly to force MEX to update the screen.
- In general, and especially at the command prompt level, PLOT3D does not keep its graphics windows "idling" with a "swapbuffers" (see "Silicon Graphics PIPELINE", Vol. II, No. 1, Fall 1985, p. 12). Thus under the window manager (and double buffering), another graphics program will hang until PLOT3D swaps buffers.

Known_problems continued

- For very complicated plots and/or particle trace plots, PLOT3D has been observed to respond VERY sluggishly. Response to the mouse may take on the order of a minute (keep that mouse button down!). Further screen updates may be somewhat faster.
- PLOT3D will not update the screen correctly if the window manager has been started up using "mex -d" (double-buffered).
- PLOT3D uses A LOT of (virtual) memory. IRISes at NASA Ames have been reconfigured with 24MB of swap space. This involves repartitioning the system disk (see the "IRIS Series 3000 Owner's Guide, Version 2.0" or your SGI rep.).
- Under W3.4-GL2, support for the Seiko color printer screen dump may not exist, indicated by unsatisfied externals (isetcolormap, isetname, iclose, iflsbuf, putrow, iopen) when linking. In this case, disable the Seiko color printer screen dump by commenting out the call to SCRDMPI in EXEQUE.
- Under W3.5-GL2, the INCLUDE file "/usr/include/fgl.h" does not define the Graphics Library function ISMEX to be type LOGICAL. If, in a future release, this is fixed, all "LOGICAL ISMEX" declarations in IRIS2 will have to be commented out.

VMS

- To link PLOT3D under VMS (with the current dimensions in MAIN), your page file quota must be something over 40,000 pages. To run PLOT3D, it must be at least 20,000 pages. You need special privilege to change this:

```
$$D SYSS$SYSTEM:  
$RUN AUTHORIZE  
MODIFY username /PGFLQUOTA=20000
```

COS

- The CRAY/COS version of PLOT3D is really not worth much due to the high memory overhead in PLOT3D and the relatively small amount of physical memory on the XMP. See

"Installation_of_PLOT3D" for tips on reducing the memory requirements of PLOT3D.

DISSPLA

- Divide-by-zero errors are sometimes generated by DISSPLA while making 2D contour plots with labels. The traceback may point to QQTLB(?), CONTUR, or ENDCO2. This is a DISSPLA internal bug. Try changing MINMAX or plotting without contour labels ("PLOT/NOLABELS")
- Spurious lines have been observed on QMS and LN03 output plots. The cause is unknown.

CRAY_2

The CRAY 2 version of PLOT3D has many idiosyncrasies, due to the poor condition of the CFT77 compiler under UNICOS. Things have improved somewhat under UNICOS 2.0. Some of the remaining problems are:

- The "\$" format descriptor does not work, so all prompts end with a <RETURN>.
- Multiple line messages, such as those printed by the SHOW command, are garbled.
- ^D end-of-files are often not trapped correctly by the FORTRAN READ statement. Type "EOF" instead (see " End-of-file").
- The utility programs CREHLPLIB and GETHELP do not work: the HELP Library file must be created on the IRIS.

APOLLO

- Titles and text have not been implemented.
- The C I/O routines (in the file CIO.C) have not been tested with PLOT3D; I don't know how to compile and link them; I've indicated something reasonable in the PLOT3DLINK file. They

Known_problems continued

are needed for the READ/BINARY command (but not READ/UNFORMATTED), and for particle tracing. C routines are included because C is faster for reading and writing large chunks of data than FORTRAN, and the Cbinary files are pure binary, as opposed to FORTRAN unformatted files, which include record length information. This latter feature facilitates the transfer of binary data files from a dissimilar type machine such as a CRAY.

- The FORTRAN compiler has a maximum number of lines of code that it will accept from one file.
- GMR3D V1.0 does NOT support filled polygons. This is in beta-test for the next release of GMR3D.
- BSP tree hidden surface removal is not supported or documented for GMR3D V1.0.
- I could not get more than 8 colors out of GMR3D in single-buffered mode (but could from GPR). This is a possible GMR3D problem. Only 8 colors are available in double-buffered mode, as documented.
- Mouse buttons work strangely--you may have to click the button a second time to turn off the rotation, rather than just releasing the button. This is a possible GMR3D problem.
- Keyboard attention interrupts are handled by PLOT3D using Aegis system calls. However, when interrupting a plot, the textport does not come back, but the "PLOT3D V3:" prompt does. This makes continued execution difficult.
- Line styles and filled polygons (beta-test) don't work correctly on the DN580--they originally come up correctly, but when the picture is complete (ready to be manipulated), dashed or dotted lines change to solid, and filled polygons disappear.
- When doing a "grow" or "pop" on the PLOT3D window, the viewport is cleared and redrawn. On the DN580, the viewport is incorrectly cleared to cyan. When the mouse buttons are used to rotate or translate the image (causing the viewport to be refreshed), the picture reappears. The refresh works correctly on a DN570.
- As a further note, when writing programs that write files, FORTRAN I/O requires that a maximum record length (RECL=#bytes) be specified on the OPEN statement when opening ANY new file. This is a BUG. This DOES affect PLOT3D, when

doing particle traces. Routines PAR2D and PAR3D must be modified to specify the record length when opening the SCRATCH file. For the rest, current coding uses C I/O routines to do the particle trace reads and writes; equivalent FORTRAN I/O (but WITHOUT the RECL parameter) is commented out beside the C calls.

Special_characters

The following characters are treated specially when encountered on ANY line of input.

- If the last character on a line of input, "-" means to consider the next input line to be a continuation of the current line.
- ' Expected to surround a symbol, to be translated (by calling an appropriate system routine) into a character string.
- " Expected to surround a character string, the contents of which are not to be disturbed. The string is treated as a single character argument.
- ! Denotes that all input following on this line is to be considered a comment and ignored.
- @ If the first (nonblank) character on a line of input, the rest of the line is taken as a file name to be read as input to PLOT3D. When an end-of-file (or EOF) is encountered, input continues from the previous source.
- EOF If the ONLY nonblank characters on an input line are EOF, this is equivalent to an end-of-file reading input. Further input is taken from the previous input file (or device).

Blanking

An array IBLANK can be included in the XYZ grid file which supplies PLOT3D with information on grid topology. This can be used to indicate that some points in the grid are "turned off" (so contours are not plotted through this region), or how multiple grids are connected (to aid in particle tracing). See "READ File_formats XYZ with_IBLANK" for information on including IBLANK in XYZ data files.[FILE FORMATS ARE IN CHAPTER 8.]

The use of IBLANK is as follows: where IBLANK is zero, the corresponding grid point is assumed turned off, and data at that point (XYZ, Q) are NEVER USED. If IBLANK is anything but zero, data at that point are assumed valid. The

NORMAL value for IBLANK in the interior of a grid is 1. Other values are used for particle tracing, and indicate grid topologies: around holes (blanked-out regions) and grid boundaries, an IBLANK value of -n means that physical space "continues" in grid n; in other words, a particle trace leaving the current grid here will continue in grid n. Note that periodic boundaries or C-mesh-type boundaries can be coded the same way: for grid m, use IBLANK=-m on the common boundaries.

Another IBLANK key is the value 2, which indicates the location of a solid wall boundary. This is used to (artificially) restrict particle traces from flowing into the wall and stopping. Particles are forced to remain some fraction of a point away from a wall. This restriction is only implemented for particle tracing through a velocity field.

Attributes

Various attributes of graphical objects can be set in commands such as WALLS SUBSETS , and CONTOURS . Examples of such attributes are color, line type and thickness, and surface transparency.

Hidden_lines

The HIDDEN_LINE attribute type is implemented as polygons filled with the background color, and outlined with the given line style. As a result, to generate a correct hidden line image, walls or subsets, as well as polygons within the walls or subsets must be ordered correctly (i.e., from back to front). Do this by setting the START, END, INCREMENT values appropriately in the WALLS or SUBSETS command, and/or split a wall or subset into two.

Shaded_surface

Implemented as filled grid cells, giving a faceted shading. Note that hidden surfaces are not by default implemented, so that for a correct hidden surface image, walls and subsets, as well as polygons within walls or subsets must be ordered correctly (i.e., from back to front). Do this by setting the START, END, INCREMENT values appropriately in the WALLS or SUBSETS command, and/or split a wall or subset into two. Shading is done assuming that the light source is at the initial viewpoint (specified with the VPOINT command). The color is ramped linearly from 0.3 times the base color at 90 deg. incidence of the light source to the polygon normal vector (angle cosine of 0), to full color at a cosine of 1 (1-shininess). As the cosine increases to 1, the color changes linearly from the base color to white.

Color

Colors may be entered as one of the eight standard (named) colors, BLACK, MAGENTA, RED, YELLOW, GREEN, CYAN, BLUE, and WHITE (D), or as a set of RGB (red-green-blue) components, in the range of 0 to 1. RGB colors are specified in the form "RGB .2 .3 .4".

Line_type

Line type (or style) can be SOLID (D), DASH, DOT, CHAINDASH, CHAINDOT, or NONE. {DOT, CHAINDASH, and CHAINDOT are not implemented for GRAFIX.}

Line_thickness

This is a real number that multiplies the standard line thickness. {Only factors of 1 or 2 are supported for IRIS 1000 series terminals and workstations. Not implemented for GRAFIX.}

Attributes continued

Symbol_type

Specifies a symbol (by integer number) to be put at each point.

Symbols are

- 1 dot (single pixel)
- 0 no symbol (default)
- 1 +
- 2 x
- 3 square
- 4 circle
- 5 diamond
- 6 triangle (delta)
- 7 triangle (nabla)
- 8 filled square
- 9 filled circle
- 10 filled diamond
- 11 filled Delta
- 12 filled nabla

{DISSPLA: Symbol 1 may not work; symbols 10-12 are implemented as unfilled.

GRAFIX : Symbols 1-12 are not implemented.

APOLLO : Symbol 3 is implemented as a *; symbols 5-12 are implemented as a circle.}

Symbol_size

Real number factor which multiplies the standard symbol size.

Shading_pattern

Shading pattern for polygon fills. Default is SOLID. {Others are not yet implemented.}

Attributes continued

Transparency

Real number between 0 (opaque) and 1 (completely transparent) controlling polygon transparency. {Not implemented for DISSPLA or GRAFIX.} The HIDDEN_LINE type is flagged via a negative transparency. A transparency value greater than 1 results in polygons being outlined and not filled.

Shininess

Shaded surface shininess; a real number between 0 and 1.

Shading is done assuming that the light source is at the initial viewpoint (specified with the VPOINT command). The color is ramped linearly from 0.3 times the base color at 90 deg. incidence of the light source to the polygon normal vector (angle cosine of 0), to full color at a cosine of (1-shininess). As the cosine increases to 1, the color changes linearly from the base color to white.

Material_coefficient

Polygon reflectivity. {Not implemented.}

@file

Use the given file as input for PLOT3D commands. On an end-of-file, control is returned to the previous source of input. This command may be entered at any time during a PLOT3D session, not necessarily at the main command level.

End-of-file

An end-of-file can be used in many commands to prematurely terminate a set of input, preventing that input from being incorporated into the program. Most computer systems have a control character for entering an "end-of-file" from the terminal; in VAX/VMS, it is ^Z. As an alternative, the letters "EOF" can be entered as an input line, signalling an end-of-file.

\$ system-command

If a "\$" is the first character of a command line (at the main command level), the rest of the line is passed to the operating system as a command (for example \$DIR when running under VAX/VMS would result in the "DIR" being sent to VMS). A "\$" alone means return to the system for a while. Upon completion (LOGOUT in VMS, ^D in Unix), return to PLOT3D. {Generally doesn't work for IRIS/Unix due to swap space limitations.}

7.3 Commands

CLEAR

Format: CLEAR

Clears all internally computed arrays (such as functions) from the WORK storage area. In general, this is handled automatically.

CONTOURS

Format: CONTOURS [max number of levels]
or CONTOURS/INCREMENT [contour increment]
or CONTOURS/MANUAL [start[,end,increment]]

Select contour levels or ranges.

Qualifiers

/RANGE

Show the range (minimum and maximum) of the current function over all active points in all active grids.

/AUTOMATIC (D)

/INCREMENT

/MANUAL

For the default AUTOMATIC mode, a maximum number of contour levels to be calculated is specified. In INCREMENT mode, a contour level increment is specified. The number of levels will depend on the function range. MANUAL entry of contour levels allows start-end-increment sets to be given.

/ATTRIBUTES (D)

/NOATTRIBUTES

Controls whether attributes such as color range, line type and thickness, and surface transparency are prompted for. {Symbols do not appear for PLOT/2D/LABELS using DISSPLA.}

EXIT

Format: EXIT

Exit out of PLOT3D. Same as QUIT .

Qualifiers

/SAVE

Save the journal file. Default is to delete it.

FSURFACE

Format: FSURFACE

Set the following properties of function surface (or line) plots:

- (1) the scale factor of the function axis ;
- (2) the origin (or offset) of walls drawn on the plot for reference; and
- (3) whether a 3D function surface will be drawn as grid lines or contour lines, if the contour attribute type is LINES.

Other properties of a function surface plot are set by other commands:
(1) the VIEW command sets which spatial (x,y,z) axes will be plotted vs. the function;

- (2) the CONTOUR command controls whether the function surface will be plotted as lines or a surface (using polygons), as well as the color map and line or surface attributes to be used;
- (3) the MINMAX command sets the range of the function to be plotted (XMIN,XMAX, YMIN,YMAX, or ZMIN,ZMAX), depending upon which axis will be used for the function; and
- (4) the /SURFACE qualifier on the PLOT command is used to select a function surface plot instead of the (default) /CONTOUR plot.

In 3D, a function surface is a plot of two spatial dimensions (x, y, or z) vs. a scalar function. This is also known as a carpet plot. Which two spatial dimensions are used is chosen using the VIEW command. A right-handed coordinate system is maintained; thus a plot with VIEW XZ would plot x vs. f vs. z, and the axis range for f would be controlled by YMIN and YMAX entered in the MINMAX command.

In 2D, a function surface can degenerate to a simple line plot, depending on the subset(s) active. A good example of a line plot is C_p (pressure coefficient) vs. x for an airfoil. Here the appropriate subset would be just the points on the airfoil surface. In any case, the plot is of one spatial dimension and the function value. (The spatial dimension used is the first axis given to the VIEW command. Currently this spatial dimension will be always be plotted on the horizontal axis, and the function along the vertical axis.)

Qualifiers

/SCALE_FACTOR=scale
/SCALE_FACTOR=AUTO (D)

Allows entry of the relative scaling of the function axis compared to the spatial (x, y, or z) axes. Thus a scale factor of 2 would mean that 1 unit of the scalar function will have the same length as 2 units in x, y, or z. Specifying "AUTO" for the scale factor means that the length of the function axis (from MINMAX) will be made equal to the longer of the other two axes (3D), or equal to the other axis (2D).

`/WALLS_ORIGIN=origin`

`/WALLS_ORIGIN=AUTO (D)origin (or offset) of walls`

Allows specification of the origin along the function axis for the walls which are drawn on the function surface or line plot. For example, if a value of 5 was entered for a plot of x vs. y vs. f, a wall at coordinates (x,y,0) would be plotted at (x,y,5) in (x,y,f) space. Specifying "AUTO" means that the origin will be set to zero.

`/CONTOUR`

`/GRID (D)`

Controls whether a function surface will be drawn as grid lines or contour lines if the contour attribute type is LINES. This flag is ignored for 2D plots.

FUNCTION

Format: FUNCTION [function number]

Allows the selection of the function number to be plotted. Numbers between 0 and 99 are for grid-type information, between 100 and 199 are scalar functions (suitable for contour plots), and numbers between 200 and 299 are vector fields. Note that currently 200-299 refer to vector plots, while 300-399 generate particle trace-type plots. Functions from 400 on up are special functions (such as shock wave locations).

Grid_functions

0 Walls alone (geometry)	10* 2D crossing grid check
1 Grids	11* Tetrahedron decomposition cell volume check
2 Outline of IBLANK holes	12*. Tetrahedron decomposition grid crossing check

Scalar_functions

100 Density (or Q1)	
101 Normalized density	
102 Stagnation density	
103 Normalized stagnation density	104* Log of normalized density
110 Pressure	
111 Normalized pressure	
112 Stagnation pressure	
113 Normalized stagnation pressure	
114 Pressure coefficient	
115 Stagnation pressure coefficient	
116 Pitot pressure	
117 Pitot pressure ratio	
118 Dynamic pressure	119* Log of normalized pressure
120 Temperature	
121 Normalized temperature	
122 Stagnation temperature	
123 Normalized stagnation temperature	124* Log of normalized temperature
130 Enthalpy	
131 Normalized enthalpy	
132 Stagnation enthalpy	
133 Normalized stagnation enthalpy	
140 (Internal) energy	*New in Version 3.6

- 141 Normalized (internal) energy
- 142 Stagnation energy
- 143 Normalized stagnation energy
- 144 Kinetic energy
- 145 Normalized kinetic energy
- 150 u velocity
- 151 v velocity
- 152 w velocity
- 153 Velocity magnitude
- 154 Mach number
- 155 Speed of sound
- 156 Cross-flow velocity
- 157 Normalized 2D stream function
- 158* Divergence of velocity
- 160 x-momentum (Q2)
- 161 y-momentum (Q3)
- 162 z-momentum (Q4)
- 163 Stagnation energy per unit volume (Q5)
- 170 Entropy
- 171 Entropy measure s1
- 180 x-component of vorticity
- 181 y-component of vorticity
- 182 z-component of vorticity
- 183 Vorticity magnitude
- 184 Swirl
- 185 Velocity x vorticity magnitude
- 186* Helicity density (degree of knottedness of tangled vortex)
- 187* Relative helicity
- 188* Filtered relative helicity
- 190 Shock function based on pressure gradient
- 191* Filtered shock function
- 192* Pressure gradient magnitude
- 193* Density gradient magnitude

*New in Version 3.6

Vector_functions

- 200 Velocity
- 201 Vorticity
- 202 Momentum (Q2,Q3,Q4)
- 203 Perturbation velocity

FUNCTION continued

204 Velocity x vorticity

210 Pressure gradient

211* Density gradient

Particle_trace_functions

300 Particle traces

301 Vortex lines

Particle traces are generated using trilinear interpolation of values of the vector function inside a computational cell, and second-order Runge-Kutta steps to advance the particle in space.

The particles are advanced using the velocity (or vector function) in real-space, but the step is limited so the particle will only advance some fraction of a computational cell.

Currently, five steps are taken per computational cell.

See the RAKE command for specification of starting points and attributes for the traces.

Shock_waves

400 Shock locations based on pressure gradient

401* Filtered shock locations

The current algorithm for "finding shocks" is to look at the Mach number component in the direction of the local pressure gradient. Where this value goes through one, AND the Mach number is decreasing, is plotted as a shock. See "FUNCTION Function_definitions Shock_function" for additional information.

The way that the shock structure is plotted is determined by the FIRST contour specification (attributes, not level). Thus one can specify LINES (and therefore plane orientations) or SURFACES, color, line thickness, surface transparency, etc.

*New in Version 3.6

FUNCTIONS continued**Nondimensionalizations**

Certain reference conditions are assumed in calculating some functions (the pressure coefficient, for example). These conditions are set in subroutines SCAFUN and VECFUN.

- | | | |
|-----|---------------------------------|-----------------------------|
| (1) | $\rho_{inf} = 1$ | (freestream density) |
| (2) | $c_{inf} = 1$ | (freestream speed of sound) |
| (3) | $p_{inf} = 1/\gamma$ | (freestream pressure) |
| (4) | $ V _{inf} = M_{inf} * c_{inf}$ | (freestream velocity magn.) |


Conditions 3 and 4 follow from 1 and 2.

Fluid_constants

The fluid is assumed to be air . The perfect gas law is also used. The following constants are used in computing functions, and are defined in BLOCK DATA BFLUID.

- | | |
|----------------|---------------------------|
| $\gamma = 1.4$ | (ratio of specific heats) |
| $R = 1$ | (gas constant) |

Function_definitions

 Version 3.6 function definitions: see Appendix A

Density

$$\rho = Q1$$

$$\rho_{inf} = 1$$

FUNCTION continued

$$\rho_0 = \rho * \left[1 + \frac{\gamma - 1}{2} * M^{**2} \right]^{**[1/(\gamma - 1)]}$$

Pressure

$$p = (\gamma - 1) * \rho * [e_0 - (1/2) * V^{**2}]$$

$$p_{inf} = 1/\gamma$$

$$p_0 = p * \left[1 + \frac{\gamma - 1}{2} * M^{**2} \right]^{**[\gamma/(\gamma - 1)]}$$

Pressure_coefficient

$$C_p = (p - p_{inf}) / (.5 * \rho_{inf} * V_{inf}^{**2})$$

$$C_{p_0} = (p_0 - p_{0inf}) / (.5 * \rho_{inf} * V_{inf}^{**2})$$

$$C_p^* = \frac{\left[\frac{2 + (\gamma - 1) * M_{inf}^{**2}}{\gamma + 1} \right]^{**[\gamma/(\gamma - 1)]} - 1}{\frac{\gamma}{2} * M_{inf}^{**2}}$$

Ref: A.H. Shapiro, THE DYNAMICS AND THERMODYNAMICS OF COMPRESSIBLE FLOW, The Ronald Press Company, New York, 1953, Vol. 1, p. 107.

Pitot_pressure

For $M < 1$, the pitot pressure is the same as the stagnation pressure. Above Mach 1, it is the equivalent stagnation pressure behind a normal shock:

$$pp = p_{\theta y} = (p_{\theta y}/p_x) * p_x \quad (x \text{ is before shock, } y \text{ is after})$$

$$\text{pitot pressure ratio} = pp/p_{inf}$$

$$p_{\theta y}/p_x = (p_{\theta y}/p_{\theta x}) * (p_{\theta x}/p_x)$$

$$= \frac{[\frac{\gamma+1}{2} * M_x^{**2}]^{**} [\gamma / (\gamma-1)]}{[\frac{2*\gamma}{\gamma+1} * M_x^{**2} - \frac{\gamma-1}{\gamma+1}]^{**} [1 / (\gamma-1)]}$$

Ref: A.H. Shapiro, THE DYNAMICS AND THERMODYNAMICS OF COMPRESSIBLE FLOW, The Ronald Press Company, New York 1953, Vol. 1, p. 119.

Dynamic_pressure

$$q = (1/2) * \rho * v^{**2} = \text{kinetic energy/unit volume}$$

Temperature

$$T = p / (\rho * R) \quad (\text{perfect gas law})$$

$$T/T_{inf} = (p/p_{inf}) / (\rho/\rho_{inf})$$

$$T_{\theta} = T * [1 + \frac{\gamma-1}{2} * M^{**2}]$$

FUNCTION continued

Enthalpy

$$\begin{aligned}
 h &= \text{gamma} * [e\emptyset - (1/2)*V**2] \\
 &= \text{gamma} * e_i \\
 &= [\text{gamma}/(\text{gamma}-1)] * (p/\text{rho}) \\
 \\
 h_{inf} &= \text{gamma}*e\emptyset_{inf} \\
 &= [\text{gamma}/(\text{gamma}-1)] * (p_{inf}/\text{rho}_{inf}) \\
 \\
 h\emptyset &= h + (1/2)*V**2 \\
 &= \text{gamma}*e\emptyset - (\text{gamma}-1)*(1/2)*V**2 \\
 &= e\emptyset + p/\text{rho}
 \end{aligned}$$

Energy

$$\begin{aligned}
 e_i &= e\emptyset - (1/2)*V**2 \\
 &= [1/(\text{gamma}-1)] * (p/\text{rho}) \\
 \\
 e_{iinf} &= [1/(\text{gamma}-1)] * (p_{inf}/\text{rho}_{inf}) \\
 \\
 e\emptyset &= Q5/\text{rho} \\
 \\
 e\emptyset_{inf} &= [1/(\text{gamma}-1)]*(p_{inf}/\text{rho}_{inf}) + (1/2)*V_{inf}**2 \\
 \\
 e_k &= (1/2)*V**2 \\
 &= q/\text{rho}
 \end{aligned}$$

Mach_number

$$\begin{aligned}
 M &= V/c \\
 \\
 c &= \text{sqrt}(\text{gamma}*p/\text{rho})
 \end{aligned}$$

Cross-flow_velocity

$$V_{\text{cross-flow}} = \text{sqrt}(v^{**2}+w^{**2})$$

2D_stream_function

Calculated by integrating the mass flow across a coordinate line. THE STREAM FUNCTION IS ASSUMED TO HAVE THE VALUE ZERO AT THE FIRST POINT IN EACH SUBSET. The values are normalized by the freestream Mach number. Only x and y, and u and v are used! See subroutine STREAM.

Momentum

$$\text{Momentum} = (Q2, Q3, Q4)$$

Entropy

$$s = c \cdot \ln \left[\frac{p}{p_{\text{inf}}} \right] + c \cdot \ln \left[\frac{\rho_{\text{inf}}}{\rho} \right]$$

$$= c \cdot \ln \left[\frac{(p/p_{\text{inf}})}{(\rho/\rho_{\text{inf}})^{\gamma}} \right]$$

$$s1 = \left[\frac{(p/p_{\text{inf}})}{(\rho/\rho_{\text{inf}})^{\gamma}} \right] - 1$$

= another measure of entropy

FUNCTION continued

Vorticity

$$\begin{aligned} \text{omega} &= \text{xi} \frac{w}{y} + \text{eta} \frac{w}{y} + \text{zeta} \frac{w}{y} - \text{xi} \frac{v}{z} - \text{eta} \frac{v}{z} - \text{zeta} \frac{v}{z} \\ &= d(w)/d(y) - d(v)/d(z) \end{aligned}$$

$$\begin{aligned} \text{omega} &= \text{xi} \frac{u}{z} + \text{eta} \frac{u}{z} + \text{zeta} \frac{u}{z} - \text{xi} \frac{w}{x} - \text{eta} \frac{w}{x} - \text{zeta} \frac{w}{x} \\ &= d(u)/d(z) - d(w)/d(x) \end{aligned}$$

$$\begin{aligned} \text{omega} &= \text{xi} \frac{v}{x} + \text{eta} \frac{v}{x} + \text{zeta} \frac{v}{x} - \text{xi} \frac{u}{y} - \text{eta} \frac{u}{y} - \text{zeta} \frac{u}{y} \\ &= d(v)/d(x) - d(u)/d(y) \end{aligned}$$

Swirl

$$\text{Swirl} = (\text{omega dot } V) / (\text{rho} * V^{**2})$$

Shock_function

The shock function based on pressure gradient, of which some parts of the level 1 contour are the shock waves, is computed as follows, where V is the velocity vector:

$$\text{Shock function} = \frac{V}{c} \cdot \frac{\text{grad}(p)}{|\text{grad}(p)|}$$

There may be a switch which sets the shock function to zero if $|\text{grad}(p)| < 0.1$. This avoids picking up preshock oscillations (see subroutine SHOCKP). It is agreed that this is not the proper place to do this. Another approach is to put the test in the contour finding/testing routines (SHK2L, SHK2LB, SHK3L, SHK3LB, SHK3S), but this requires carrying the pressure gradient along.

Perturbation_velocity

$$v' = v - v_{inf}$$

HELP

PLOT3D Version 3.5 20 May 1987

Format: HELP [keyword [keyword [keyword ...]]]

The HELP command prints information on a list of keywords. "*" is a wildcard, while "*..." matches anything at the current level or below. Thus "HELP *..." prints all HELP information available. When responding to a "Topic?" or "Subtopic?" prompt, a "?" causes information for the current level to be repeated; a <RETURN> pops HELP up one level, and an end-of-file terminates the HELP session.

LIST

Format: LIST [XYZ or Q or FUNCTION]

List the XYZ, Q, or current function data. Output can be directed to the screen (default) or to a file. FORMATTED , UNFORMATTED , or BINARY files suitable for reading into PLOT3D can be produced as well. [As of this writing, PLOT3D is unable to read these files, but SURF does read them. This is a good way to get PLOT3D to calculate functions for use in SURF, which provides many lighting and shading choices.]

Qualifiers

/TEXT (D)

/FORMATTED

/UNFORMATTED

/BINARY

Select the type of LIST output desired. TEXT output includes column headings and is suitable for viewing on the screen or printing out. FORMATTED, UNFORMATTED, and BINARY options produce files which can be read into PLOT3D using the READ command.

/OUTPUT=file

Redirect the list output to a file or device. A file name is required for FORMATTED, UNFORMATTED, or BINARY lists, and will be prompted for in case the /OUTPUT qualifier has not been included in the command line.

MAP

Format: MAP

Produce a map of WORK array usage, including grid number, variable names, and source file names.

MINMAX

Format: MINMAX [xmin,xmax,ymin,ymax[,zmin,zmax]] or
MINMAX/INCREMENT [xmin,xmax,xinc,ymin,ymax,yinc[,zmin,zmax,zinc]]

Set the plot ranges, or general region of interest for the plot.

Qualifiers

/X (D)

/NOX

/Y (D)

/NOY

/Z (D)

/NOZ

Controls which set of axis limits are to be changed. To change only the y-axis limits, for instance, type "MINMAX/Y ymin,ymax".

/INCREMENT

Allows the specification of a tick mark increment. An increment of zero implies automatic scaling of tick marks and possibly rounding of axis limits. (Axis drawing can be suppressed by using PLOT/NOAXES.)

/XSCALE=scale

Sets x-axis scale factor. Default is one.

/YSCALE=scale

Sets y-axis scale factor. Default is one.

/ZSCALE=scale

Sets z-axis scale factor. Default is one.

PLOT

Format: PLOT

Initiate a plot. When the plot is completed, type <RETURN> to return to PLOT3D command level.

Qualifiers

/2D

/3D (D)

Type of plot. Note that 2D data can be displayed in a 3D plot too.

/TEKTRONIX (D for DISSPLA)

/TK41XX

/ENVISION

/VERSATEC

/DIP

/QMS

/LN03

/IMAGEN

/DICOMED

/COMPRESS

/IRIS (D for IRIS)

/IRISRGB

/IRISSINGLE

/GRAFIX (D for GRAFIX)

/APOLLO (D for Apollo)

/APOLLOSINGLE

Output device.

DISSPLA ONLY devices :

TEKTRONIX (4014)(default)

TK41XX (Tektronix 41xx series terminals will prompt for terminal type)

ENVISION (215)

VERSATEC (connected to a VAX; produces files PARM.PLV and VECTR1.PLV -- see \$HELP PLOT for info on submitting to Versatec)

DIP (NASA Ames Device Independent Plot file; produces file xxx.DIP*)

QMS (QMS 1200 laser printer; produces file xxx.QMS*)

LN03 (DEC LN01S or LN03 Plus laser printer; produces file
INTSCRT1.TMP)
IMAGEN (IMAGEN 8/300 laser printer; produces file IMA-
GEN.IMP)
DICOMED (D48 film recorder; produces file xxx.D48*)
COMPRESS (DISSPLA device independent file to be used with
POP; produces file POPFIL.DAT)

(On the CRAY , only DIP and DICOMED are available; at NASA
Ames, use the COS JCL
AMESLIB.
DISSPLA,HELP.
for information on DISSPLA output files)

IRIS ONLY devices:

IRIS (double buffered) (default)
IRISSINGLE (single buffered)
IRISRGB (RGB mode)

GRAFIX ONLY devices:

GRAFIX (produces xxx.GRA*) (default)

APOLLO ONLY devices:

APOLLO (double buffered) (default)
APOLLOSINGLE (single buffered)

* For those devices that generate a file (such as DIP), the base filename used will be the Q filename (or the XYZ filename, or simply "PLOT3D") with an appropriate three-letter extension (such as ".DIP"). Exceptions are PLOT/COMPRESS, which produces a file, "POPFIL.DAT", PLOT/LN03, which generates "INTSCRT1.TMP", and PLOT/IMAGEN, which generates "IMAGEN.IMP". If the device specified does not apply to the version of PLOT3D being run, the default for that version is used instead.

/CONTOUR (D)
/SURFACE
/CARPET
/LINE

If this is a plot of a scalar variable, set whether the plot will be a CONTOUR plot (the default) or a function SURFACE plot. A CARPET plot is another common term for a function surface, and

PLOT continued

in 2D a function surface degenerates to a LINE plot; thus CARPET and LINE are synonyms for SURFACE. See HELP FSURFACE for more information on function surface plots.

/FULLSCREEN

/NOFULLSCREEN (D)

Allows for an enlarged plotting area to be used, at the expense of titles and text. Very helpful for 3D plots. {For the IRIS or GRAFIX, FULLSCREEN simply removes all text from the display.}

/AXES (D)

/NOAXES

Controls whether axes are put on the plot or not.

/FIGURE=(areax,areay,charht)

/NOFIGURE (D)

For /FIGURE, the plot size and character height is specified (in inches), title, contour bar, and additional text are not plotted, and other measures are taken to pretty up the plot. {Most of this can not be done for the IRIS or GRAFIX.}

/BACKGROUND=color

Set the background color of the plot. Note that if "color" is of the form "RGB r,g,b", it must be enclosed in parentheses (e.g., /BACKGROUND=(RGB .5,.5,.5)) so the entire string is associated with the /BACKGROUND qualifier. The default is BLACK. {Not implemented for DISSPLA. The display program (e.g. GAS) sets the background color for GRAFIX.}

/UP=axis

Specify which axis will be considered generally vertical for a plot. Valid "axes" are X, Y, Z, +X, +Y, +Z, -X, -Y, or -Z. (A right-handed system is always assumed.) A viewpoint specified in spherical coordinates (VPOINT/ANGLES) sets the angles phi, in the horizontal plane, and theta, above the plane. The default is /UP=Z for a 3D plot, /UP=Y for 2D.

Note that for a 2D plot, the "y-axis" really means the second axis on the plot: if VIEW XZ had been selected, /UP=Y would put the Z-valued axis up. Similarly, if VIEW ZX had been selected,

/UP=Y would put the x-axis up! There is the possibility of confusion here!

Example: We have a 2D airfoil, and would like velocity profiles with u horizontal and y vertical. We select proper subsets and function, type "VIEW YX" so the line plot will use Y as the spatial axis, "MINMAX 0 .2 -.5 1" so y will range from 0 to 0.2, u from -0.5 to 1. Then we say "PLOT/2D/LINE/UP=X": we're making a 2D line plot, and we want the axis which WOULD have been horizontal (and corresponds to the first set of MINMAX values) to be "up". Thus y will be up, u horizontal.

/SCRIPT

Further input will be interpreted as SCRIPT commands for controlling the orientation of the display on the screen. STOP will signify the end of SCRIPT commands. See SCRIPT (several pages past here) for information on these commands. {Not implemented for DISSPLA or GRAFIX.}

 See Appendix A, PLOT for new 3.6 PLOT qualifiers.

IRIS_interface

PLOT3D will run with or without the IRIS window manager MEX. Under MEX, PLOT3D can open multiple graphics ports. By default, each plot is NOT deleted when a <RETURN> is entered and the program returns to the PLOT3D command prompt. Subsequent plots are displayed as additional graphics ports connected to PLOT3D. Mouse and keyboard input can be attached to the different ports by using the window manager "attach" menu item while mouse and keyboard manipulation is active in PLOT3D (i.e., while a picture is being displayed).

To delete a graphics window, attach to the window (again, while a picture is being displayed, NOT in PLOT3D command mode) and type <DELETE>.

To make different graphics ports have the same viewpoint:

- (1) attach to the port with the desired viewpoint,
- (2) type "=", and
- (3) attach to the port(s) to be modified.

PLOT continued

NOTE: a "hogwhiledown " keyboard button can be identified in the user's ".mexrc" file to allow attaching to different PLOT3D graphics windows.

The .mexrc file lines

```
reservebut 13
```

```
bindfunc hogwhiledown 13
```

bind the "no scroll" key to the hogwhiledown function. For more information on the .mexrc file and hogwhiledown, read the Window Manager section in the IRIS User's Guide.

Mouse

On the IRIS, the mouse can be used to manipulate the picture on the graphics screen. The mouse buttons are used to activate different degrees of freedom for manipulation, while the mouse (x,y) displacement controls the RATE of motion in the respective degree of freedom.

While the left mouse button is being depressed, x-motion of the mouse is associated with rotation (about the center of the MINMAX box) in phi, while y-motion controls rotation in theta. (Phi and theta are cylindrical coordinate angles, using the DISSPLA convention.)

The middle mouse button activates zooming (toward the center of the MINMAX box), controlled with the y-motion of the mouse. X-motion is not used.

Depressing the right button allows the picture to be dragged left or right (x-motion of the mouse), and up or down (y-motion) on the screen. REMEMBER, the POSITION of the mouse controls the RATE of motion (speed) of the object.

The sensitivity of the mouse to zooming and screen dragging is a function of the distance of the VPOINT initial viewpoint from the center of the MINMAX box. The closer the viewpoint to the MINMAX box, the less sensitive the mouse. While the plot is on the screen, this can be adjusted by typing "up-arrow" (to increase the sensitivity by a factor of 2) or "down-arrow" (to decrease it).

More than one button can be depressed at one time. The mouse position will control the rates associated with the last button pushed, while the mo-

tions associated with the other buttons are maintained, as long as they remain depressed.

Type <RETURN> to return to PLOT3D command level.

Seiko_color_printer_screen_save

Type "S" to save the current graphics window.


Type "F" to save the full screen.

The following message will appear when the save is done: "Graphics window [or Screen image] saved as file xxx.N.sei" where xxx is the Q filename (or XYZ filename, or simply "PLOT3D"), and N is the frame number (starting from 1 for each PLOT3D run). These files can be printed by typing the Unix command "lp -dseiko xxx.N.sei".

[These commands are entered when the plot is on the screen]. Once the image has been saved, the picture can again be manipulated with the mouse.

This screen save option is a fixed-up version of the routine SCRSAVE supplied by Silicon Graphics under W3.5-GL2 and documented in the GL2-W3.5 Release Notes, p. 4-38. It still references routines in the library "/usr/people/gifts/mextools/implib/libimage.a".

RAS_screen_save

 Vers. 3.5 or earlier: type "P" to save the graphics window.

 Version 3.6 see Appendix A, PLOT for .ras and PostScript!

The following message will appear when the save is done: "Graphics window saved as frame N in file xxx.ras". The filename used is the Q filename (or XYZ filename, or simply "PLOT3D") with a .RAS extension.

Once the image has been saved, the picture can again be manipulated with the mouse.

One file is opened per PLOT3D run. Frames are added to the existing file, if there is one (so we don't delete it by creating a new one!).

Screen images can be restored by running the program RASP.

PLOT continued

The ARCGraph GRAFIX RAS file format is used. See the file "arcgraph/grafix.doc", ARCGraph Internal Functional Specifications for a description of the format.

Screen_dump_to_QMS

Type "Q" to save the graphics window.

The following message will appear when the save is done: "Graphics window saved as page N in file xxx.qms".

The filename used is the Q filename (or XYZ filename, or simply "PLOT3D") with a .QMS extension.

This is intended for line drawings only--anything that is not the background color (usually black) will be colored black on the QMS (where the background is, of course, white).

Once the image has been saved, the picture can again be manipulated with the mouse.

A new file is opened for every PLOT3D run. Thus if the file already exists, it is deleted!

Backfacing_polygon_removal

On the IRIS (2000 or 3000 series), typing "B" toggles the removal of backfacing polygons on or off. The default is off. See the IRIS User's guide for a description of backface polygon removal and its uses. Note that this is not an exact procedure, and some polygons may not get properly removed due to screen resolution. (The polygon normal is determined in screen coordinates.)

This is a new feature, and as such is not yet polished. The clockwise or counterclockwise ordering of polygon points in walls can be changed by using a negative increment for one of the indices. This is true for shocks and contour surfaces as well (I believe), by using a negative increment in the subset definition. Point ordering for color contours does NOT change with the sign of the increment, however.

Z-buffering

***** WARNING ***** Under GL2-W3.5 (as of 18 March 1987), z-buffering does not work for any displays using nonsolid linestyle! The machine may crash.

IRIS 2000 or 3000 series machines with 28 or more bitplanes support hardware z-buffering to generate hidden surface pictures. Type "Z", *while the plot is on the screen*, to toggle z-buffering on and off. The default is off. The MINMAX command should be set to include everything of importance, as the near and far clipping planes are activated with z-buffering, and are set to include somewhat more than the volume specified by the MINMAX command.

Z-buffering cannot be done from RGB mode (PLOT/IRISRGB); also, if the display is in double buffer mode, a side-effect of typing "Z" is that the display mode will change to single buffer mode. The mode will change back when z-buffering is turned off.

Parameters IZNEAR and IZFAR must be set correctly in subroutine IRIS.

SCRIPT

These commands, when entered following a PLOT/SCRIPT command, control the motion of the objects on the display screen. {Not implemented for DISSPLA or GRAFIX.}

CENTER

Format: CENTER [x, y, z]

Move the center of the display (equivalent to the center of the MINMAX box) to the given (x, y, z) point. Objects on the screen will rotate about this point.

Qualifiers

/FROM=(x, y, z)

Specify what point to move the center from. Default is the previous CENTER, or center of the MINMAX box.

/IN=n

Move the center to the new point in n frames.

PLOT continued

GO

Format: GO [number of frames]

"GO n" means advance the display n frames. The default value for n is one.

HOLD

Format: HOLD [number of frames]

"HOLD n" means hold the display still for n frames. Default value for n is one.

INTERACTIVE

Format: INTERACTIVE

Temporarily enable interactive manipulation of the display. Typing <RETURN> returns control to the SCRIPT input. See "PLOT Mouse" for a description of interactive control of the display.

STOP

Format: STOP

Signals the end of SCRIPT commands.

VPOINT

Format: VPOINT [x,y,z] or VPOINT/ANGLES [phi,theta,radius]

Change the viewpoint to the given (x,y,z) or (phi,theta,radius) position. Viewpoint is always looking toward the CENTER point.

Qualifiers

/ANGLES

Viewpoint (as well as FROM viewpoint, if supplied) is in (phi,theta,radius) spherical coordinates. Default is (x,y,z) Cartesian coordinates

/FROM=(x, y, z)

Viewpoint from which to move. Specified as (x,y,z) or (phi,theta,radius), depending on /ANGLES. Default is to move from previous viewpoint.

/IN=n

Move the viewpoint to the new viewpoint in n frames.

QUIT

Format: QUIT

Exit out of PLOT3D. Same as EXIT.

Qualifiers

/SAVE

Save the journal file. Default is to delete it.

RAKES

Format: RAKES

Specify the starting locations and attributes of particle traces (and vortex lines, both of which are traced through a vector field).

Qualifiers

/IJK (D)

/XYZ

Traces can be started at (i,j,k) grid points or at (x,y,z) physical coordinates.

/ADD

Rake specifications are to be added to the end of existing rake specs. This does not work when the particle traces are read in with **/READ=file**.

/ATTRIBUTES (D)

/NOATTRIBUTES

Controls whether attributes such as color, line type and thickness, and surface transparency are prompted for.

/READ=file

/WRITE=file

Particle traces can be written out or read in to PLOT3D, allowing traces to be replotted later, or calculated elsewhere. See "File_format" for information on file structure. The default file type is .PAR.

/+TIME (D)

/-TIME

/+-TIME

Controls whether traces in this rake are followed in the plus, minus, or both directions of the vector field.

/MAXPOINTS=n

Set the maximum number of points for each particle trace segment, allowing a crude control on the length of the trace. Each trace MIGHT be made up of more than one segment, as a new segment is started when a trace continues in another grid. Currently, a particle takes 5 steps to cross a computational cell; therefore, setting /MAXPOINTS=15 means that each trace will be roughly 3 cells long.

The default value (and maximum) is /MAXPOINTS=2000.

/SCALAR_FUNCTION=function number**/NOSCALAR_FUNCTION (D)**

Sets the function to be used to determine the color of the traces. Each trace will be colored according to the value of this function (which must be a scalar function) and the color map(s) specified using the CONTOUR command. If /NOSCALAR_FUNCTION is selected, the color attribute color will be used. This is the default.

Restriction

Traces can be restricted to remain in computational planes. This is useful for plotting particle traces on symmetry planes or for simulating oil flow. For oil flow pictures, traces are often restricted to remain one computational plane above the surface (as the velocity on the surface is zero).

Valid restrictions are IJ, IK, or JK.

File_format

Trace files are read and written using FORTRAN UNFORMATTED I/O (BINARY on IRIS/Unix) as follows:

```

10  CONTINUE
    READ (unit,END=20)  IRAKE, ITRACE, N, IGRID
    READ (unit)  (X(I), I=1, N), (Y(I), I=1, N), (Z(I), I=1, N), (T(I), I=1, N),
C          (RI(I), I=1, N), (RJ(I), I=1, N), (RK(I), I=1, N)
    ...
    GOTO 10
20  CONTINUE

```


RAKES continued

IRAKE is the rake number, and will be used to determine the attributes of the trace when it is drawn. **ITRACE** is the trace number inside this rake, and **N** is the number of points to be read in the following record. **IGRID** is the grid number for this part of the trace. The physical location (x,y,z) and time t is recorded for each point in the trace, as well as the (i,j,k) coordinates, saved as real numbers to indicate both the cell number and position within the cell of the point.

A trace may appear in the file as several segments, all with the same rake and trace number.

READ

Format: READ

Read input data files.

Qualifiers

/1D

/2D

/3D (D)

Controls whether the input files are expected to be 1-, 2-, or 3-dimensional.

/XYZ=file (D)

Input standard grid file.

/Q=file (D)

Input standard Q file, including density, momentum, and stagnation energy per unit volume.

/FUNCTION=file

Input an arbitrary function. {Currently there is no way to plot an arbitrary function.}

/MDATASET

For multiple datasets in one file. {Not implemented.} (Intended to support time-varying solutions.)

/MGRID

If the files contain multiple grids, in the multiple-grid form.

/FORMATTED

/UNFORMATTED (D)

/BINARY (D for Unix)

Controls whether the files are expected to be in FORTRAN FORMATTED or UNFORMATTED form. For character form, free-formats (READ(unit,*)) are used. Under IRIS/Unix, I/O is done much more efficiently using C and straight BINARY files (which have no record information). Thus BINARY is the default under Unix. (Note that BINARY I/O can be done from FORTRAN too, using OPEN(...,FORM='BINARY').)

READ continued

/PLANES

/WHOLE (D)

Specifies whether the data is to be read in planes (one plane per record) or the whole thing at once. See "READ File_formats" for some examples of reads with /PLANES and /WHOLE. Note that /PLANES has no effect for reading other than 3D datasets.

/CHECK (D)

/NOCHECK

Controls whether the default checking of Q data for zero or negative density or pressure is performed. Bypassing this check can speed up the Q file reading significantly.

/JACOBIAN

/NOJACOBIAN (D)

Indicates whether Q variables are scaled (divided) by the determinant of the metric Jacobian, in which case this determinant is expected as an additional Q variable. {This option may disappear, leaving the default of NOJACOBIAN.}

/BLANK

/NOBLANK (D)

Signals whether the XYZ file contains an integer array IBLANK as a fourth (or rather n-dimension plus one) variable. At (i,j,k) points where IBLANK=0, data will NEVER BE USED.

File_formats

[See Chapter 9, File Formats. In the on-line helpfile, file formats are given here, under READ.]

SHOW

Show the current status of some commands. Applicable commands are CONTOURS, FSURFACE, FUNCTION, MINMAX, PLOT, RAKES, SUBSETS, TEXT, VECTORS, VIEW, VPOINT, and WALLS.

[Note that there is no slash between SHOW and the command to be shown: SHOW FUNCTION.]

SUBSETS

Format: SUBSETS

Allows parts of grids to be selected as "active".

Qualifiers

/GRID=grid number

Specifies which grid these subsets are to be associated with (default is grid 1).

/ADD

These subsets are to be added on to the current list, rather than starting from subset number 1.

/ATTRIBUTES

/NOATTRIBUTES (D)

Controls whether attributes such as color, line type and thickness, and surface transparency are prompted for. Subset attributes are applied to GRID plots. {In the future, if NOATTRIBUTES is specified for certain types of plots, subset attributes would be used instead. }

TEXT

Format: TEXT ["line 1"["line 2"]]

Allows the entry of up to two additional lines of text (besides the title), to be centered above future plots. This text will continue to be used until another TEXT command is given. Current text can be "erased" by entering a <RETURN> in response to the prompt for text. (Text is prompted for if not entered on the command line.) If text is entered on the command line, be sure to enclose each line in double quotes.

VECTORS

Format: VECTORS

Sets vector plot characteristics, such as arrowheads, vector length, and associated scalar function for vector colors.

Qualifiers

`/SCALAR_FUNCTION=function number`

`/NOSCALAR_FUNCTION (D)`

Sets the function to be used to determine the color of the vectors. Each vector will be colored according to the value of this function (which must be a scalar function) and the color map(s) specified using the `CONTOUR` command. If `/NOSCALAR_FUNCTION` is selected, the color attribute color will be used. This is the default.

`/ATTRIBUTES (D)`

`/NOATTRIBUTES`

Sets various vector attributes. `NOATTRIBUTES` resets all everything to default values. Attributes are:

- Color - Vector color if `NOSCALAR_FUNCTION` is selected. Default is white.
- Vector scaling - whether or not the vector is scaled with the magnitude. Default is scaled.
- Vector length scale factor - default is 1 (a vector function of unit length will be drawn as one unit long.
- Arrowhead type - `LINES (D)`, `FILLED`, or `NONE`.
- Arrowhead scaling - whether or not the arrowhead is scaled with the vector length. Default is scaled.
- Arrowhead size factor - default value of 1 corresponds to 3/10ths the vector length for scaled arrowheads, 1/50th the longest axis if not scaled.
- Line type -
- Line thickness -
- Symbol type - causes a symbol to be put at the base of the vector.
- Symbol size -

See `HELP ATTRIBUTES` for information on line type, etc.

VIEW

Format: VIEW [view]

The VIEW command is used for two cases: (1) to specify which axes are to be horizontal and vertical for a 2D plot, and (2) to specify which two spatial axes are to be used for 3D function surface (carpet) plots. Choices are: XY,XZ,YZ,YX,ZX,ZY, TOP,SIDE,FRONT . TOP,SIDE,FRONT are equivalent to XY,XZ,YZ, respectively.

VPOINT

Format: VPOINT [x,y,z]
or VPOINT/ANGLES [phi,theta,radius]

Enter a viewpoint for a 3D plot.

Qualifiers

/XYZ (D)

/ANGLES

Controls whether the viewpoint will be interpreted as an (x,y,z) point in space, or as phi,theta,radius values, corresponding to DISPLA conventions.

WALLS

Format: WALLS

Specifies which parts of the grids should be drawn for all plots, indicating the grid "geometry" or "configuration" (hence the name WALLS).

Qualifiers

/GRID=grid number

Specifies which grid these walls are to be associated with (default is grid 1).

/ADD

These walls are to be added on to the current list, rather than starting from wall number 1.

/ATTRIBUTES (D)

/NOATTRIBUTES

Controls whether attributes such as color, line type and thickness, and surface transparency are prompted for.

Chapter 8. Data File Formats

By Pieter Buning

8.1 Introduction

This chapter shows the data file formats for PLOT3D. These formats are in the *on-line* help file, under READ file_formats. Most of the grid and solution generator programs supply the data already correctly formatted. One thing to check is that your solutions are calculated for the grid points, not at cell centers.

Because PLOT3D is written in FORTRAN, your XYZ and Q files must be in FORTRAN, which is no problem because they are normally written from a FORTRAN program. PLOT3D reads three types of FORTRAN files:

- 1) FORTRAN unformatted files — .dat extension.
- 2) FORTRAN formatted files — .fmt extension.
- 3) Binary files for the IRIS — .bin extension.

Standard FORTRAN supports .dat or .fmt files. The .dat (unformatted) are preferred because they are smaller than .fmt files. The .fmt files are used only when files that will work on any machine are desired. FORTRAN on the IRIS supports .dat, .fmt or .bin (binary); .bin is the default. On the IRIS 4D, FORTRAN unformatted is recommended (and READ with the /UNFORMATTED qualifier). Binary from FORTRAN doesn't work correctly on 4D's.

The formats below will work for .bin or .dat files. For .fmt files, substitute (IUNIT,*) for (IUNIT), and use the /FORMATTED qualifier with the READ command.

8.2 Grid XYZ Files

xyz_file single_grid

1D:

```
READ (IUNIT) IDIM  
READ (IUNIT) (X(I), I=1, IDIM)
```

2D:

```
READ (IUNIT) IDIM, JDIM
```

```

      READ(IUNIT) ((X(I,J), I=1, IDIM), J=1, JDIM),
C              ((Y(I,J), I=1, IDIM), J=1, JDIM)

```

3D (/WHOLE):

```

      READ(IUNIT) IDIM, JDIM, KDIM
      READ(IUNIT) ((X(I,J,K), I=1, IDIM), J=1, JDIM), K=1, KDIM),
C              ((Y(I,J,K), I=1, IDIM), J=1, JDIM), K=1, KDIM),
C              ((Z(I,J,K), I=1, IDIM), J=1, JDIM), K=1, KDIM)

```

3D (/PLANES):

```

      READ(IUNIT) IDIM, JDIM, KDIM
      DO 10 K= 1, KDIM
          READ(IUNIT) ((X(I,J,K), I=1, IDIM), J=1, JDIM),
C                  ((Y(I,J,K), I=1, IDIM), J=1, JDIM),
C                  ((Z(I,J,K), I=1, IDIM), J=1, JDIM)
10      CONTINUE

```

xyz_file with_IBLANK

3D (/WHOLE):

```

      READ(IUNIT) IDIM, JDIM, KDIM
      READ(IUNIT) (((X(I,J,K), I=1, IDIM), J=1, JDIM), K=1, KDIM),
C              ((Y(I,J,K), I=1, IDIM), J=1, JDIM), K=1, KDIM),
C              ((Z(I,J,K), I=1, IDIM), J=1, JDIM), K=1, KDIM),
C              (((IBLANK(I,J,K), I=1, IDIM), J=1, JDIM), K=1, KDIM)

```

3D (/PLANES):

```

      READ(IUNIT) IDIM, JDIM, KDIM
      DO 10 K= 1, KDIM
          READ(IUNIT) ((X(I,J,K), I=1, IDIM), J=1, JDIM),
C                  ((Y(I,J,K), I=1, IDIM), J=1, JDIM),
C                  ((Z(I,J,K), I=1, IDIM), J=1, JDIM),
C                  ((IBLANK(I,J,K), I=1, IDIM), J=1, JDIM)
10      CONTINUE

```

xyz_file multiple_grid

3D (/WHOLE):

```

    READ(IUNIT) NGRID
    READ(IUNIT) (IDIM(IGRID),JDIM(IGRID),KDIM(IGRID),IGRID=1,NGRID)
    DO 10 IGRID= 1,NGRID
        READ(IUNIT)
    C ((X(I,J,K),
    C   I=1, IDIM(IGRID)),J=1,JDIM(IGRID)),K=1,KDIM(IGRID)),
    C ((Y(I,J,K),
    C   I=1, IDIM(IGRID)),J=1,JDIM(IGRID)),K=1,KDIM(IGRID)),
    C ((Z(I,J,K),
    C   I=1, IDIM(IGRID)),J=1,JDIM(IGRID)),K=1,KDIM(IGRID))
10    CONTINUE

```

3D (/PLANES):

```

    READ(IUNIT) NGRID
    READ(IUNIT) (IDIM(IGRID),JDIM(IGRID),KDIM(IGRID),IGRID=1,NGRID)
    DO 10 IGRID= 1,NGRID
    DO 10 K= 1,KDIM(IGRID)
        READ(IUNIT)
    C ((X(I,J,K),I=1, IDIM(IGRID)),J=1,JDIM(IGRID)),
    C ((Y(I,J,K),I=1, IDIM(IGRID)),J=1,JDIM(IGRID)),
    C ((Z(I,J,K),I=1, IDIM(IGRID)),J=1,JDIM(IGRID))
10    CONTINUE

```

8.3 Solution Q Files

Q_file

Listed are some samples... The Q header information (extra information at the beginning of each grid) includes the following:

```

Freestream Mach number (FSMACH)
Angle-of-attack        (ALPHA)
Reynolds number        (RE)
Time (not used)        (TIME)

```

Q_file single_grid (without Jacobian)

1D:

```

READ(IUNIT) IDIM
READ(IUNIT) FSMACH, ALPHA, RE, TIME
READ(IUNIT) ((Q(I, NX), I=1, IDIM), NX=1, 3)

```

2D:

```

READ(IUNIT) IDIM, JDIM
READ(IUNIT) FSMACH, ALPHA, RE, TIME
READ(IUNIT) ((Q(I, J, NX), I=1, IDIM), J=1, JDIM), NX=1, 4)

```

3D (/WHOLE):

```

READ(IUNIT) IDIM, JDIM, KDIM
READ(IUNIT) FSMACH, ALPHA, RE, TIME
READ(IUNIT) (((Q(I, J, K, NX), I=1, IDIM), J=1, JDIM), K=1, KDIM),
C          NX=1, 5)

```

3D (/PLANES):

```

READ(IUNIT) IDIM, JDIM, KDIM
READ(IUNIT) FSMACH, ALPHA, RE, TIME
DO 10 K= 1, KDIM
    READ(IUNIT) ((Q(I, J, K, NX), I=1, IDIM), J=1, JDIM), NX=1, 5)
10 CONTINUE

```

Q_file multiple_grid (without Jacobian)

3D (/WHOLE):

```

READ(IUNIT) NGRID
READ(IUNIT) (IDIM(IGRID), JDIM(IGRID), KDIM(IGRID), IGRID=1, NGRID)
DO 10 IGRID= 1, NGRID
    READ(IUNIT) FSMACH, ALPHA, RE, TIME
    READ(IUNIT)
C (((Q(I, J, K, NX),
C   I=1, IDIM(IGRID)), J=1, JDIM(IGRID)), K=1, KDIM(IGRID)),
C   NX=1, 5)
10 CONTINUE

```

PLOT3D User's Manual

```
3D (/PLANES) :
  READ(IUNIT) NGRID
  READ(IUNIT) (IDIM(IGRID), JDIM(IGRID), KDIM(IGRID), IGRID=1, NGRID)
  DO 10 IGRID= 1, NGRID
    READ(IUNIT) FSMACH, ALPHA, RE, TIME
    DO 10 K= 1, KDIM(IGRID)
      READ(IUNIT)
    C      (((Q(I, J, K, NX), I=1, IDIM(IGRID)), J=1, JDIM(IGRID)), NX=1, 5)
10      CONTINUE
```

8.4 Function Files

Function_file single_grid

```
1D:
  READ(IUNIT) IDIM, NVAR
  READ(IUNIT) ((F(I, NX), I=1, IDIM), NX=1, NVAR)

2D:
  READ(IUNIT) IDIM, JDIM, NVAR
  READ(IUNIT) ((F(I, J, NX), I=1, IDIM), J=1, JDIM), NX=1, NVAR)

3D (/WHOLE) :
  READ(IUNIT) IDIM, JDIM, KDIM, NVAR
  READ(IUNIT) (((F(I, J, K, NX), I=1, IDIM), J=1, JDIM), K=1, KDIM),
  C      NX=1, NVAR)

3D (/PLANES) :
  READ(IUNIT) IDIM, JDIM, KDIM, NVAR
  DO 10 K= 1, KDIM
    READ(IUNIT) ((F(I, J, K, NX), I=1, IDIM), J=1, JDIM), NX=1, NVAR)
10  CONTINUE
```

Function_file multiple_grid

```

3D (/WHOLE) :
    READ(IUNIT) NGRID
    READ(IUNIT) (IDIM(IGRID), JDIM(IGRID), KDIM(IGRID), NVAR(IGRID),
C           IGRID=1, NGRID)
    DO 10 IGRID= 1, NGRID
        READ(IUNIT)
C      (((F(I, J, K, NX),
C        I=1, IDIM(IGRID)), J=1, JDIM(IGRID)), K=1, KDIM(IGRID)),
C        NX= 1, NVAR(IGRID))
10    CONTINUE

3D (/PLANES) :
    READ(IUNIT) NGRID
    READ(IUNIT) (IDIM(IGRID), JDIM(IGRID), KDIM(IGRID), NVAR(IGRID),
C           IGRID=1, NGRID)
    DO 10 IGRID= 1, NGRID
        DO 10 K= 1, KDIM(IGRID)
            READ(IUNIT)
C      ((F(I, J, K, NX), I=1, IDIM(IGRID)), J=1, JDIM(IGRID)),
C      NX=1, NVAR(IGRID))
10    CONTINUE

```


Chapter 9. Changing PLOT3D and Host Systems

By Larry Pierce

9.1 Summary

PLOT3D is very powerful; it can calculate complex functions on millions of grid points. Running PLOT3D takes a lot of memory. This chapter explains what you need from your system, and how you can modify the PLOT3D source code to take up less space. See also Appendix B "Tools" for ways to use less memory.

For those who want to change PLOT3D in other ways, there is a brief discussion of the overall structure of PLOT3D code, and discussions of changing to a different form of input data, different operating system, and different graphics library.

9.2 Getting More Space on Host System

You must have access to enough memory on your host system:

IRIS

On the IRIS, the amount of memory you can use is determined by the amount of swap space on the disk and by what else is running on the machine at the time. If you don't have enough swap space, your command to start PLOT3D (`plot3d`) will be answered by "plot3d:no space."

If you have enough space to run, but not to generate all the graphics to make the picture, then after "plot" you might get "out of space in gl_makeroom."

To get around these problems, make simpler pictures, or increase swap space, which involves reformatting the disk, as explained in the IRIS manuals. Call User Support (415-694-4444) for help.

VMS/VAX

The system may not allow you enough virtual memory. If you do not have enough to run PLOT3D, when you type "plot3d", it will say "quota exceeded." At this point, your system administrator needs to increase the page file quota on

your account, probably to 40,000 pages. See the "3.5 Release Notes, Known problems VMS" in section 7.2 of Chapter 7: Key Words/Help File.

There is an additional system-wide limit on the system as to how big a job can be; that also has to be large enough.

The most frequent problems are with particle traces, because they take a lot of disk space. You have to either do fewer traces or get more disk space quota.

There are two kinds of quota on VMS/VAX:

1. Virtual memory = page file quota.
2. Disk space quota = blocks.

If PLOT3D will not even start, you need more page quota; if it crashes, you may need more disk space.

9.3 Code Organization

Documentation of PLOT3D, Version 3 (by Pieter Buning, May 1986; modified by Larry Pierce, September 1988)

PLOT3D sections:

1. Commands

Includes PROGRAM PLOT3D (called MAIN and included dimension for WORK in previous versions), P3DINI, P3DERR, and P3DCOM (all in PLOT3D in previous versions), and all the other subroutines for handling all commands (definition of syntax, setting up of information for execution, and execution of most commands). That is, 3.6: PROGRAM PLOT3D and P3DCOM

3.5: unnamed main and PLOT3D

3. 2D

Make 2D plots, including contour lines, line plots, vectors, particle traces, and shocks.

a. Grid

Grid plotting routines, such as walls, grid subsets, and hole boundaries. (Function numbers 0-99.)

b. Scalar variables

Calculation of variables, calling contour routines, plotting contour lines or function surface (line) plots. (Functions 100-199.)

c. Vector variables

Calculation of variables and plotting of vectors. (Functions 200-299.)

d. Particle traces

Calculation of vector variable, handling of rake start and continuation points, saving and plotting of particle traces. (Functions 300-399.)

e. Shock waves

A "special" category for Euler/Navier-Stokes datasets. Calculation of necessary variables, calling special version of contour routines. (Functions 400-499.)

4. 3D

Make 3D plots, including contour lines, function surface (carpet) plots, vectors, particle traces, and shocks.

a. Grid

Grid plotting routines, such as walls, grid subsets, and hole boundaries. (Function numbers 0-99.)

b. Scalar variables

Calculation of variables, calling contour routines, plotting contour lines or function surface (carpet) plots. (Functions 100-199.)

c. Vector variables

Calculation of variables and plotting of vectors. (Functions 200-299.)

d. Particle traces

Calculation of vector variable, handling of rake start and continuation points, saving and plotting of particle traces. (Functions 300-399.)

e. Shock waves

A "special" category for Euler/Navier-Stokes datasets. Calculation of necessary variables, calling special version of contour routines. (Functions 400-499.)

5. Functions

All routines for calculating scalar and vector functions for Euler/Navier-Stokes datasets. FUNNAM returns the name associated with any function number, and FUNSYM returns its symbol. For undefined function numbers, a blank space is returned.

6. Particle tracing

Routines for finding and tracing particles through an arbitrary vector field.

7. Shock waves

Function calculation and special contour routines needed for shock wave plotting.

8. Contours

Routines for contour lines on 2D or 3D data planes, or surfaces through 3D data.

9. Storage

Routines to handle the initialization, allocation, and cleanup of WORK array space. This includes keeping track of where each array in WORK came from, whether they are currently in memory, and rereading them from the original input files if necessary to free space in WORK.

Actually, "virtual memory" has not been implemented, but some of the structure necessary for it may be in the code.

PLOT3D 3.5 can not page data out to a file (or page in).

All you can use is what is in memory ... hence not "virtual" memory. PLOT3D 3.6 can't page in/page out data either; since dynamic memory allocation is used in 3.6, this is less important.

10. Parser

Routines for defining an "interactive environment" (prompt string, I/O devices), command syntax (including command arguments, qualifiers, and qualifier arguments), translation of reals and integers to and from character strings, and handling of error, warning, and informational messages.

11. Utilities

General-purpose routines for copying or zeroing arrays, transposing arrays, computing the gradient of a vector field, and various other linear algebra routines (dot and cross product, normalize vector, etc.).

12. IRIS2, DISSPLA, or GRAFIX, ETC.

All routines dealing with plotting: initialization of graphics device, title, contour, and additional text routines, definition of viewing transformation, color changing, and point, line, and polygon drawing. All device-dependent routines, such as mouse interpretation and screen dumping for the IRIS.

13. Unix or VMS

Operating system-dependent routines, such as how to interpret file names and types (file.DAT, file.PAR), set up keyboard attention interrupt handling, execute operating system commands, translate symbols, and find out if the program input is interactive or batch.

9.4 Changing PLOT3D to Use Less Space

The first thing PLOT3D does is dimension the WORK array to equal the scratch space that the grids and Q files will be put into. That dimension should be large enough to hold 13 times the number of grid points. In versions 3.5 and before, that number can be changed, by editing the source code. Version 3.6 uses dynamic memory allocation, and changing NWORK will have no effect!

Change one FORTRAN statement:

```
PARAMETER (NWORK=3000000)
```

The number in the PARAMETER statement may be different than what is shown here (because of differences in systems). Change the number as desired, and then compile the program.

Usually this statement is located at the top of file "commands.f" (or "commands.for", "commands.ftn", etc., depending on what system you are running on). If there is no 'commands' file, search all the PLOT3D files using a pattern matching utility (e.g., use grep if your operating system is UNIX).

For your information, NWORK is the dimension of the real number array WORK, which is used as a space for grid, solution, and function data. Another thing that takes a lot of memory is the number of grids that PLOT3D can handle. To reduce required memory, change all occurrences of MGRID=50 to, e.g., MGRID=5. This is still effective in Version 3.6.

Changing MGRID is not easy, because you have to change every PARAMETER statement that defines MGRID. To find each occurrence, use a pattern matching utility on all the PLOT3D files. In every file that contains a PARAMETER statement defining MGRID, you must change the dimension to whatever you wish, and be consistent: give the same value to MGRID wherever it occurs.

9.5 Different Form of Input Data

There is a simple technique, called "fooling PLOT3D" that is frequently used to plot functions that are not on the list. There are five solution variables in the Q file that PLOT3D reads verbatim, without making further calculations:

Q1 = function 100

Q2 = function 160

Q3 = function 161

Q4 = function 162

Q5 = function 163

You could put almost any kind of solutions in the five places in the Q file, and then use the above function numbers to read them and plot them. Use the regular READ command, except be sure to use the /NOCHECK qualifier, because your solutions would not satisfy the check. Only the grid functions and your special functions would be valid; the normal PLOT3D function calculations would make no sense. Use PLOT/FULLSCREEN or PLOT/FIGURE to suppress printing of function name, or, in version 3.6, use PLOT/NOTITLE. As an alternative to fooling PLOT3D, you could modify the source code, as described below.

Users very often wish to define their own functions for their own application. A good example of this is converting PLOT3D to handle turbulence data (TURB3D). How does one approach these modifications?

Obviously, the most fundamental things that must be changed are the function definitions and the input format. Let us first tackle the function definitions.

Function definitions:

Many of the functions are defined in one module called "functions." Depending on which PLOT3D you have (e.g., VMS/DISSPLA, SUN, etc.) the "functions" module may or may not exist as a separate file. If it does exist as a separate file it will be called "functions.f*" (surprise!). If it doesn't exist as a separate file, then it is buried in a big file with a number of other modules. In this case, search the big file(s) for "SUBROUTINE SCAFUN". A search on this subroutine should land you somewhere in the middle of the "functions" module.

Once you are in the functions module, you can modify the definitions for scalar or vector functions, function names, and function symbols.

Function names are used within PLOT3D to check function numbers for meaning. The function number is passed to a function called FUNNAM, which returns the name of the function in a character string. If the function number is defined then the character string will be something like 'PRESSURE' or 'VORTICITY'. If the function number is not defined, then the character string will be a space and the user will get a message "Function number x is not implemented". Function numbers are checked in this manner before any space is allocated for a function variable, and before any function computation begins. So, if you plan to add/delete/change any PLOT3D functions, you should edit FUNCTION FUNNAM to reflect the additions/deletions/changes that you are making.

Function symbols might be used when doing function line or function surface plots (e.g., x vs coefficient of pressure, as a 2-D example). They are used as "shorthand" for the name of the function, as there is limited space on the screen or plotting page. These symbols are defined for some functions, and if you change the functions you may want to add/delete/change corresponding function symbols. Edit FUNCTION FUNSYM to do so.

Hints to be used when adding/changing/deleting function definitions:

- In the "functions" module are all the definitions for scalar and vector functions. All scalar functions are executed from SUBROUTINE SCAFUN; SCAFUN takes the function number as an input parameter and calls the appropriate subroutine(s) to calculate the scalar function (numbers 100-199). Likewise, all vector functions (numbers 200-299) are executed from SUBROUTINE VECFUN; VECFUN takes the function number as an input parameter and calls the appropriate subroutine(s) to calculate the vector function.
- In most cases, SCAFUN and VECFUN only call one subroutine to calculate a function; SCAFUN calls scalar subroutines and VECFUN calls vector subroutines, most of the time. There are some function numbers for which more than one routine is called. For example, function number 180 (x component of vorticity) requires that vorticity (function number 201) be calculated, and so VECFUN is called from within SCAFUN with a function number of 201 as a parameter; VECFUN in turn calls a routine to calculate the vorticity (a vector quantity consisting of x , y , and z components). In other cases vector functions are called directly from SCAFUN, or scalar functions are called directly from VECFUN. Keep your wits about you.
- There are a number of COMMON blocks that one should be mindful of when changing the "functions" module. COMMON/FREEST/ defines a few freestream values (see SCAFUN and VECFUN) that are used in the calculation of certain functions. COMMON/FLUID/ defines fluid constants (see BLOCK DATA BFLUID); note that COMMON/FLUID/ is referenced in other modules such as "commands". COMMON/GRIDS/ is referenced in the "functions" module, and contains the values input from the header record of your solution file (FSMACH, ALPHA, RE, TIME). If you want to add/change/delete values from the header record, you must change all occurrences of COMMON/GRIDS/ in all the files to reflect the changes.
- The particle trace functions are defined in a separate module, called "particles". PARTI2 and PARTI3 are two subroutines that reside in this module. The particle tracing capabilities of this module are believed to be general enough so

that particles can be traced through any (sensible) vector field. In the case of PLOT3D, velocity and vorticity are the vector fields used for making particle traces. However, one could conceivably use the vector field from some kind of magnetics study to trace "magnetic lines."

- There is also a separate module, "shocks," used for calculating shock locations based on pressure gradient. These routines are modifications of the contour routines.

Input Format

If you are changing PLOT3D to fit your application, you'll probably need to change the input format. To change the input format of the solution file you'll have to modify the "commands" module, and possibly others. Anyway, in the "commands" module is a subroutine READQ. This subroutine is a good place to start when attempting to modify the input format of your solution file; it reads the problem dimensions, reads the header information, and reads all the solution variables; note that space must be allocated in the WORK array for the solution data (space is not allocated for the dimensions or header info), and it is in subroutine READQ that the space requirement is computed and space in WORK allocated. Hence, you must make changes to READQ (and possibly to routines that it calls) if you add/change/delete solution header values, or if you add/change/delete solution data values. The header information is read in READQH, and stored in COMMON/GRIDS/, as mentioned previously.

You could change the format of the grid file as well, but this could require vast changes in the PLOT3D code. For example, if you wanted to use a triangular grid scheme rather than a rectangular one, you'd probably have to make quite a few changes throughout the code. READX is the routine in the "commands" module that takes care of grid file input.

QCHECK (functions.f) {formerly READCK} for Checking Solutions

Another note: subroutine QCHECK in the "functions" module, checks for bad pressure, and bad density for every grid point as the solution data is read in. If you are changing the nature of the solution data, you'll probably want to change QCHECK.

Yet another note: in Version 3.6, subroutine REMJAC in the "commands" module checks for bad Jacobian (if you have it) and then removes Jacobian

information from the solution data. You may need to modify or eliminate this routine as dictated by your application.

9.6 Different Operating System

Changing to a different operating system will require changes in the operating system dependent module(s) of the PLOT3D code. For example, if you have the VAX/VMS/DISSPLA version of PLOT3D and wanted to port the code from VAX/VMS to VAX/Unix then you would have to make changes in the file "vms.for". In some cases you may have to modify a number of FORTRAN, C, or assembly modules. See also section 9.3.

9.7 Different Graphics Library

The graphics library dependent code is typically organized into a single module. For example, if you are porting your VAX/VMS DISSPLA version of PLOT3D to use another graphics library, you would need to change file "displa.for". See also section 9.3.

Chapter 10. Error Messages and Warnings

By Larry Pierce

Errors from version 3.6 are included in this chapter.

10.1 Organization

This chapter is divided into two sections: error messages and warnings. Lower case letters such as "x" are used to indicate variables.

For IRIS/Cray2 error messages, users should check to see that the version numbers on PLOT3D and PLOT3D_LCL match; there might be the wrong version of one.

10.2 Error Messages

Error messages have the following format:

```
Error.  
  blah blah blah . . .  
  . . .
```

Here is the list of PLOT3D error messages, with comments and suggestions:

/SCALE_FACTOR argument x not recognized as a real number or AUTO.

You issued a command (like FSURFACE) with the /SCALE_FACTOR=x qualifier. The argument x you gave was not the string "AUTO" and was not a real number. Try the command again with "AUTO" or with a real number.

/WALLS_ORIGIN argument not recognized as a real number or AUTO.

You issued a command (like FSURFACE) with the /WALLS_ORIGIN=x qualifier. The qualifier argument x you specified was not the string "AUTO" and was not a real number. Try again with "AUTO" or a real number.

A filename is REQUIRED for non-TEXT files.

If you are LISTing your grid, solution, or function data in a format such as FORMATTED, UNFORMATTED, or BINARY, you must specify an output file name (listing will not go to the screen unless you use the default list format, TEXT). You can specify a filename with the /OUTPUT=*x* qualifier.

A negative point limit is not valid.

If you are using the RAKE command with the /MAXPOINTS=*n* qualifier, you must specify a positive *n* for the maximum number of points per trace segment (a trace segment is that part of a trace that lies within one grid).

Ambiguous command *x*.

You tried to issue a command, but didn't supply enough characters for PLOT3D to uniquely identify the command. For example, if you typed in "R" as a command, PLOT3D would not know if you wanted "RAKES" or "READ" or whatever. Supply more characters in the command next time.

Ambiguous qualifier *x*

You issued a command with a qualifier that could not be uniquely identified. For example, suppose that you gave the command "CONTOURS" with the qualifier "/A." PLOT3D would not know if you meant "CONTOURS/AUTOMATIC" or "CONTOURS/ATTRIBUTES." Supply more characters in your qualifier next time ("/AU" or "/AT" would be enough in this case).

An increment is required if you give both a start and an end.

You issued a command like "CONTOUR/MANUAL" which requires that real number set(s) be input (a real number set is three real numbers in this order: start, end, increment). You specified two real numbers (a start and an end) but did not give an increment. Try it again. If you typed "1. 2." you should have typed "1. 2. *x*" (where *x* is a real increment).

Argument length exceeds internal storage capacity.

You typed in a command or qualifier argument that was just too long. You might see this message if you typed something like this: "FSURFACE/SCALE_FACTOR=supercalifragilisticexpialidocious" where "supercalifragilisticxpialidocious" is the qualifier argument.

Arguments are not allowed with qualifier *x*

You typed in a command with a qualifier and a qualifier argument, but no qualifier argument was expected. You would see this message if you typed "LIST FUNCTION /BINARY=density.bin" (where "density.bin" is the qualifier argument). In this case, you meant this:
"LIST FUNCTION/BINARY/OUTPUT=density.bin".

Arrowhead type *x* is ambiguous.

You probably should not see this error message, because the arrowhead types are "LINES", "FILLED", or "NONE", and there is no chance for ambiguity. If you do see this error message, call me.

Arrowhead type *x* not recognized.

You input an arrowhead type (look up the VECTOR command) that is not recognized as "LINES", "FILLED", or "NONE". Try again with one of these arrowhead types (can be abbreviated "L", "F", or "N", respectively).

Associated scalar function number *x* is not implemented.

You supplied a scalar function number that is not implemented. Try "HELP FUNCTION SCALAR" in PLOT3D to see a list of implemented scalar functions. Or see FUNCTIONS in Chapter 7.

Attribute type *x* is ambiguous.

You supplied some kind of attribute type that PLOT3D could not uniquely identify. I do not have a handy example of this, but try typing in more characters (the golden rule of PLOT3D and VAX/VMS).

Attribute type *x* not recognized.

You supplied some kind of attribute type that PLOT3D could not recognize. For example, if you responded to the following prompt, "LINES, SURFACES, GRID_LINES, COLOR_CONTOURS, or DOTS?" with this, "PINK_ELEPHANTS" then PLOT3D would give you this error message!!

Axis *x* is ambiguous.

You gave a command like PLOT/UP=*a* where *a* specifies the "up axis". The axis *a* you specified could not be uniquely identified. For example, if you typed "PLOT/UP=+", PLOT3D wouldn't know if you meant "PLOT/UP=+X", "PLOT/UP=+Y", or "PLOT/UP=+Z". Again, use enough characters.

Axis *x* not recognized.

You gave a command like PLOT/UP=*a* where *a* specifies the "up axis". The axis *a* you specified could not be identified. For example, if you typed "PLOT/UP=K", you would get this error message. Try "HELP PLOT /UP" in PLOT3D for a list of valid axis names. Or see third page of PLOT in Chp. 7.

Bad RGB value, color reset to BLACK.

You specified a bad RGB color somewhere while using PLOT3D. For example, if you typed "RGB -1. 1. 1." you would see this message. Try "HELP ATTRIBUTES COLOR" for help with colors in PLOT3D. RGB values must be between 0 and 1.

Bad RGB value, color reset to WHITE.

You specified a bad RGB color somewhere while using PLOT3D. For example, if you typed "RGB -1. 1. 1." you would see this message. Try "HELP ATTRIBUTES COLOR" for help with colors in PLOT3D. RGB values must be between 0 and 1. See Attributes, near the end of section 7.2 in this manual.

Bad RGB value.

You specified a bad RGB color somewhere while using PLOT3D. RGB values must be between 0 and 1. See Attributes, section 7.2 in this manual. Try "HELP ATTRIBUTES COLOR" for help with colors in PLOT3D.

Bad grid number.

You gave a bad grid number in some PLOT3D command (like SUBSETS). Make sure that your grid number is a positive integer, and that it corresponds to an actual grid that has been read in.

Bad status from RCVGRA.

You tried to PLOT something while running the IRIS/Cray2 version of PLOT3D. For some reason, the plot information did not get from the Cray2 to the IRIS, or was garbled. Try PLOTing again. See Chapter 11, section 11.4. If problems persist, call me.

Bad status from SNDMSG sending "plot3d_lcl done" message.

You tried to PLOT something while running the IRIS/Cray2 version of PLOT3D. The PLOT3D_LCL program on the IRIS could not send a message to the Cray2 PLOT3D. See Chapter 11, section 11.4.

Bad status from SNDMSG sending "plot3d_lcl ready" message.

You tried to PLOT something while running the IRIS/Cray2 version of PLOT3D. The PLOT3D_LCL program on the IRIS could not send a message to the Cray2 PLOT3D. See Chapter 11, section 11.4. If problems persist, call me.

Call to Unix SYSTEM(3S) failed in SYSCOM.

You tried to: 1) execute an operating system command from the PLOT3D command line, by typing "\$" followed by the system command, or 2) escape to the operating system by typing "\$" followed by <return> while running PLOT3D. For some reason, PLOT3D was unable to get the operating system to cooperate. If you were trying to execute a system command, check your typing to make sure that you didn't misspell. Also check swap space limit (see Chapter 9.2). Let me know if this continues to be troublesome.

Can't do z-buffering from IRISRGB mode. Use PLOT/IRIS or IRISSINGLE instead.

You did a plot with PLOT/IRISRGB and then hit the "Z" key in an attempt to Z-buffer the image. IRIS hardware does not permit z-buffering with RGB mode, so re-plot with PLOT/IRIS or PLOT/IRISSINGLE instead.

Can't have more than x graphics ports. Please delete one or more ports. . .

You are running PLOT3D under MEX, and have the maximum (x) number of graphics ports open already. Delete at least one before you ask for another one.

Can't make a QMS page from RGB mode.

You tried to make a QMS screen dump ("Q" key while plot is on the screen) while in RGB mode. Try "PLOT/IRIS" or "PLOT//IRISSINGLE" instead. Note that your colors may change (but the QMS dump is in black and white).

Can't use ALL as an end value.

You issued a command like WALLS that asks for an integer set (3 integers or "ALL"). You typed something like "1 ALL" when you should have typed "ALL". Read up on the "WALLS" or "SUBSETS" commands.

Color x is ambiguous.

You specified a color name like "BL" which could mean "BLUE" or "BLACK". Use enough characters to identify the color name uniquely, like "BLU" or "BLA" in this case.

Color x not recognized.

You specified a color name that was not recognized. Valid color names are: BLACK, WHITE, BLUE, CYAN, GREEN, YELLOW, RED, MAGENTA, and RGB.

Command length exceeds internal storage capacity.

Perhaps you gave a command line that was just too long for PLOT3D to handle. See if you can get the same job done by using two or more shorter command lines.

Command *x* not recognized.

You gave a command like "GET_ME_COFFEE", which PLOT3D does not recognize. Try giving PLOT3D a valid command (like "WALLS", "SUBSETS", "PLOT", or whatever), or get your own coffee. See Chapter 12, section 12.3 for a complete list of commands, and Chapter 7, section 7.3 for explanations of the commands.

Contour surfaces are not implemented for 2D plots.

Contour surfaces do not make sense for 2D plots.

Couldn't make window *x* current.

You were running PLOT3D under mex or 4sight (on an IRIS), and got this message. Perhaps you gave a bad argument to PLOT/OVERLAY=*x*... check your window numbers. If this one is troublesome, give me a call.

DICOMED output filename *x* is too long. Limit is 23 characters.

Shorten your DICOMED filename to at most 23 characters in all.

Data is in a nonstandard form of 2D--IDIM, JDIM, or KDIM is 1 and the corresponding X, Y, or Z is not all zero. Subroutine VORTIC is not set up to calculate this vorticity.

Basically what you have is a 3D data set (an I, J, and K) that has only one computational plane (e.g., KDIM is 1) but varies in X, Y, and Z (i.e., computationally it is 2D, but geometrically it is 3D). PLOT3D cannot calculate vorticity because it cannot be sure what all the derivatives should be (it makes no assumptions).

Data is in a nonstandard form of 2D--IDIM, JDIM, or KDIM is 1 and the corresponding X, Y, or Z is not all zero. Subroutine *x* is not set up to calculate this divergence.

Basically what you have is a 3D data set (an I, J, and K) that has only one computational plane (e.g., KDIM is 1), but varies in X, Y, and Z (i.e., computationally it is 2D, but geometrically it is 3D). PLOT3D cannot calculate divergence because it cannot be sure what all the derivatives should be (it makes no assumptions).

Data is in a nonstandard form of 2D--IDIM, JDIM, or KDIM is 1, and the corresponding X, Y, or Z is not all zero. Subroutine GRAD is not set up to calculate this vorticity.

Basically what you have is a 3D data set (an I, J, and K) that has only one computational plane (e.g., KDIM is 1) but varies in X, Y, and Z (i.e., computationally it is 2D, but geometrically it is 3D). PLOT3D cannot calculate vorticity because it cannot be sure what all the derivatives should be (it makes no assumptions).

End-of-file reading FUNCTION DATA file, grid *x*, plane *y*.

You were READING a FUNCTION DATA file, and an end-of-file was encountered. Your data file is probably screwed up, so it would be a good idea to re-do it. Make sure that the grid dimensions agree with the number of function values in the file.

End-of-file reading FUNCTION DATA file, grid *x*.

Same as the above message, except this is WHOLE format, not PLANES.

End-of-file reading problem dimensions in FUNCTION DATA file.

Same as the above message, but you know that the end-of-file was encountered in the problem dimensions (IDIM, JDIM, KDIM, . . .).

End-of-file reading Q DATA file, grid *x*, plane *y*.

You were READING a Q DATA file, and an end-of-file was encountered. Your data file is probably screwed up, so it would be a good idea to re-do it. Make sure that the grid dimensions agree with the number of values in the file.

End-of-file reading Q DATA file, grid *x*.

Same as the above message, except this is WHOLE format, not PLANES.

End-of-file reading Q DATA header record, grid *x*.

Same as above, but you know that the end-of-file was encountered in the Q DATA header record (FSMACH, ALPHA, RE, TIME).

End-of-file reading problem dimensions in Q DATA file.

Same as the above message, but you know that the end-of-file was encountered in the problem dimensions (IDIM, JDIM, KDIM, . . .).

End-of-file reading XYZ DATA file, grid *x*, plane *y*.

You were READING a Q DATA file, and an end-of-file was encountered. Your data file is probably screwed up, so it would be a good idea to re-do it. Make sure that the grid dimensions agree with the number of values in the file.

End-of-file reading XYZ DATA file, grid *x*.

Same as the above message, except this is WHOLE format, not PLANES.

End-of-file reading problem dimensions in XYZ DATA file.

Same as the above message, but you know that the end-of-file was encountered in the problem dimensions (IDIM, JDIM, KDIM, . . .).

End-of-file reading frame *x* in RAS file *y* Picture save aborted. File must be fixed before it can be used.

You hit the "P" key ("R" in 3.6) in an attempt to save the plot you are looking at. PLOT3D hit an unexpected end-of-file in the existing part of the RAS file you are writing to. You will probably have to start from the beginning again.

Error closing file x for deletion in subroutine DELFIL.

PLOT3D will often attempt to delete an existing file if it needs to replace it with a new one. For some reason it was unable to CLOSE the file x for deletion. Check the file protections of file x to make sure it is read-able and write-able. Call me if this is troublesome.

**Error opening file x
FORM=y for deletion in subroutine DELFIL.**

PLOT3D will often attempt to delete an existing file if it needs to replace it with a new one. For some reason it was unable to OPEN the file x with format y for deletion. Check the file protections of file x to make sure it is read-able and write-able. Call me if this is troublesome.

Error defining symbol x.

You probably did this: "\$symbol := definition" (on VAX/VMS version). For some reason the symbol definition failed. Try it again, and check your VMS syntax.

Error from RCVMSG, expecting "plot3d starting plot" or "plot3d done".

You tried to PLOT using the IRIS/Cray2 version of PLOT3D. Try plotting again. See Chapter 11.4. If problems persist, call me.

Error opening HELP file x.

You did "HELP" and it won't work. Maybe the help files are not installed. Call me.

Error opening RAS file x. Picture save aborted.

You hit the "P" key ("R" in version 3.6) in an attempt to save the plot you are looking at. PLOT3D was unable to open a RAS file for writing. Maybe you don't have permission to create a file in the current working directory, or the disk is full.

Error positioning RAS file x in frame y Picture save aborted. File must be fixed before it can be used.

You hit the "P" key ("R" in version 3.6) hoping to save your current plot (and you have probably hit the "P" key previously in this PLOT3D session), and PLOT3D didn't like the RAS file. You will probably have to start over from the beginning.

Error reading FUNCTION DATA file, grid x , plane y .

An error was encountered while READING a FUNCTION data file. Most likely a bad integer or real number was encountered, so you will have to re-do the FUNCTION data file.

Error reading FUNCTION DATA file, grid x .

Same as above, but the file is in WHOLE format.

Error reading problem dimensions in FUNCTION DATA file.

Same as above, but you know that the error occurred while reading the problem dimensions (IDIM, JDIM, . . .).

Error reading HELP file x

You are using the PLOT3D "HELP", and for some reason PLOT3D doesn't like the HELP file it is reading. Call for help.

Error reading Q DATA file, grid x , plane y .

An error was encountered while READING a Q data file. Most likely a bad integer or real number was encountered, so you will have to re-do the Q data file.

Error reading Q DATA file, grid x .

Same as above, but the file is in WHOLE format.

Error reading Q DATA header record, grid x .

Same as above, but you know that the error occurred while reading the header record (FSMACH, ALPHA, RE, TIME).

Error reading problem dimensions in Q DATA file.

Same as above, but you know that the error occurred while reading the problem dimensions (IDIM, JDIM, . . .).

Error reading XYZ DATA file, grid x , plane y .

An error was encountered while READING a Q data file. Most likely a bad integer or real number was encountered, so you will have to re-do the Q data file.

Error reading XYZ DATA file, grid x .

Same as above, but the file is in WHOLE format.

Error reading problem dimensions in XYZ DATA file.

Same as above, but you know that the error occurred while reading the problem dimensions (IDIM, JDIM, . . .).

Error reading frame x in RAS file y . Picture save aborted. File must be fixed before it can be used.

You hit the "P" key ("R" in version 3.6) in an attempt to save the plot you are looking at. PLOT3D didn't like something it found in the existing part of your RAS file. You will probably have to start from the beginning again.

Error writing frame x in RAS file y . Picture save incomplete. File must be fixed before it can be used.

You hit the "P" key ("R" in version 3.6) in an attempt to save the plot you are looking at. For some reason PLOT3D could open the RAS file, but could not write to it. You will probably have to start from the beginning again.

Expecting a qualifier.

You typed something weird like "PLOT/", which made PLOT3D think that a qualifier should be there. Type "PLOT" or "PLOT/something" instead.

Expecting qualifier arguments.

You typed something like "PLOT/UP=", which made PLOT3D expect an axis name as an argument. Type "PLOT/UP=*axis_name*" instead. See Chp.7, PLOT .

Feedback buffer OVERFLOW in subroutine PSDRW.

Buffer is dimensioned x.

Program may be corrupted.

IRIS manual says this shouldn't happen!

You hit the "P" key while looking at a plot on your IRIS in hopes of getting a PostScript draw file. The complexity of the image being processed was great enough to cause an array overflow, and so the PLOT3D program may be corrupted. Terminate your PLOT3D session (use *exit/save*). The PostScript file may be incomplete. If you really need a PostScript draw file of that particular image, call me and we'll talk about re-dimensioning the arrays.

Feedback buffer OVERFLOW in subroutine PSDRW. Buffer is dimensioned x. Program may be corrupted. IRIS manual says this shouldn't happen!

This message would indicate a bug in the IRIS graphics interface. Let me know if you ever see it.

File x does not exist.

Probably you told PLOT3D to READ something from a file that it could not find. If files are not in your current working directory, give the path name in quotes. Check your spelling. See Chapter 3, section 3.3.

Function number x has not been implemented.

You specified a function number (with the FUNCTION command or some other command) that does not correspond to an implemented PLOT3D function. Try "HELP FUNCTION" in PLOT3D for a list of implemented functions.

Function number x is not implemented.

You specified a function number (with the FUNCTION command or some other command) that does not correspond to an implemented PLOT3D function. Try "HELP FUNCTION" in PLOT3D for a list of implemented functions.

Function x is not a scalar function.

You specified a function number (in RAKES or VECTOR perhaps) that does not correspond to a scalar function. Recall that scalar function numbers are 100-199, and not all function numbers in that range are implemented. Try "HELP FUNCTION SCALAR" in PLOT3D for a list of implemented functions.

Function x is not a scalar or vector function. LIST FUNCTION is only supported for scalar or vector functions.

You tried to LIST FUNCTION, but your current function (set in the most recent FUNCTION command you gave) is not a scalar or vector function, so PLOT3D won't do it. Recall that scalar function numbers are 100-199, and vector function numbers are 200-299, and not all function numbers in those ranges are implemented. Try "HELP FUNC SCAL" or "HELP FUNC VECTOR" for help with function numbers. Or see FUNCTION in Chapter 7.

Function x is not implemented.

You specified a function number (with the FUNCTION command or some other command) that does not correspond to an implemented PLOT3D function. Try "HELP FUNCTION SCALAR" in PLOT3D for a list of implemented functions.

Incompatible qualifier combination x and y

You gave a command like "PLOT/IRISRGB/IRISSINGLE" in which the qualifiers are in direct conflict. In this case note that the IRIS can only be in one mode at a time, and so it does not make sense to specify two modes. Choose one .

Input file stack is full.

You did an @command_file in PLOT3D, and that command file called another command file, etc., so many times that PLOT3D could not handle it. Change

your command files so that the calling stack is not so deep (make the calling structure "flatter").

Input line was too long.

You input a command line that was simply too long for PLOT3D to store. Can you find a way to get the same job done with two or more shorter command lines? If not, maybe you can reconfigure PLOT3D to accept longer command lines (call me for help on this).

Insufficient number of arguments.

You input a command, but did not supply enough arguments. Try using "HELP" with your command, to find out how many arguments it is expecting.

Integers less than 1 are not valid indices.

You input an integer less than one (probably in a WALLS, SUBSETS, or RAKES command) when PLOT3D was expecting something like an I, J, or K grid index. Do the command again, and make sure your indices make sense (i.e., are at least one, and are within the grid).

Internal storage exhausted while getting space for variable x grid y . Variable length was z , available free space was n . Automatic paging of variables has not been implemented.

This basically means that PLOT3D is out of memory. No more grids, solutions, or computed functions can be stored, and the best thing to do is quit. It might be possible to reconfigure PLOT3D so that it can store more data. See Chapter 9.

Invalid index x .

You were probably in a WALLS, SUBSETS, or RAKES command, and you input an index that PLOT3D didn't like. Make sure that your index is a positive integer (and is within the grid if it is an I, J, or K index).

Invalid integer *x*

PLOT3D was expecting an integer, and what it took as input was not an integer. Don't give decimal points with integers, and don't give exponential notation. Make sure your integers are separated from other arguments by spaces or commas.

Invalid real number *x*

PLOT3D was expecting a real number, and got something else as input. Try again, and make sure that your real number does not contain any extraneous characters.

It will be difficult to make a plot with no data.

You tried to do a screen dump of your current plot, and for some reason PLOT3D thinks that there are no data. Call me.

LIST type *x* is not unique.

If you get this command call me! LIST types are XYZ, Q, and FUNCTION, and you should not get this error message!

LIST type *x* not recognized.

You gave the LIST *x* command, but PLOT3D could not recognize *x* as "XYZ", "Q", or "FUNCTION". Try again.

Line type *x* is ambiguous.

You gave a line type that can not be uniquely recognized. For example, if you type "CHAIND" PLOT3D wouldn't know if you meant "CHAINDOT" or "CHAINDASH". Give more characters. Try "HELP ATTR LINE" for a list of valid line types. Or see Attributes in Chapter 7, section 7.2.

Line type *x* not recognized.

You gave a line type that can not be recognized. Try "HELP ATTR LINE" for a list of valid line types. Or see Attributes in Chapter 7, section 7.2

Matching right parenthesis not found.

Try giving the same command, but make sure that your parentheses match.

Message from PLOT3D_LCL not recognized. Message was x Expecting "plot3d_lcl done."

This message should only appear if you are running the IRIS/Cray2 version of PLOT3D. It means that the PLOT3D running on the IRIS sent a message to the PLOT3D running on the Cray2 that was garbled. Call me.

Message from PLOT3D_LCL not recognized. Message was x Expecting "plot3d_lcl ready."

This message should only appear if you are running the IRIS/Cray2 version of PLOT3D. It means that the PLOT3D running on the IRIS sent a message to the PLOT3D running on the Cray2 that was garbled. Call me.

Message from remote PLOT3D not recognized. Message was x Expecting "plot3d starting plot." or "plot3d done."

This message should only appear if you are running the IRIS/Cray2 version of PLOT3D. It means that the PLOT3D running on the Cray2 sent a message to the PLOT3D running on the IRIS that was garbled. Call me.

Multiple qualifier arguments must be enclosed in parentheses.

When using a command like "VPOINT/FROM=(1.,2.,3)", note that the arguments 1., 2., and 3. must be enclosed in parentheses.

NVAR exceeds internal array dimension for scratch space in routine PCLINE.

You did a CONTOUR/CUBIC followed later by a PLOT command. The routine that does the parametric cubic interpolation ran out of scratch space. If you really need to do this, call me and we'll try to make more scratch space.

No IBLANK data for grid x. LIST aborted.

You did a "LIST" command, and PLOT3D was expecting some IBLANK data, but couldn't find it. You probably did a "READ/BLANK" command on an grid file that didn't contain IBLANK data.

No Q data for grid x. LIST aborted.

You did a "LIST Q" command, and PLOT3D could not find any Q data. Try reading in the Q data! Make sure that the number of grids is the same in your XYZ and Q data files.

No XYZ data for grid x. LIST aborted.

You did a "LIST XYZ" command, and PLOT3D could not find any XYZ data. Try reading in the XYZ data! Make sure that the number of grids is the same in your XYZ and Q data files.

No argument between commas.

You typed in something strange like "1,,10". Try it again, but make sure you put arguments between your commas!

No command on line.

PLOT3D was looking for a command, but didn't find one. Hit <RETURN> a few times, and continue where you left off.

No free unit numbers available.

This might indicate a PLOT3D bug. There are routines GETLUN and FRELUN that get and free-up logical unit numbers for FORTRAN I/O, and if they are not properly matched, PLOT3D could run out of unit numbers. Call me or look at your PLOT3D source.

No good way to abort at this point. Please try again.

This message should only appear if you are running the IRIS/Cray2 version of PLOT3D. It means that you gave a bad IRIS workstation name, and so you should start over again.

No hardware for 30Hz mode.

The IRIS has 3 possible display modes out of which at most 2 can be installed. The 3 possible display modes are 60Hz, 30Hz, or NTSC. This message indicates that your IRIS cannot do 30Hz mode. Try 60Hz or NTSC.

No hardware for 60Hz mode.

The IRIS has 3 possible display modes out of which at most 2 can be installed. The 3 possible display modes are 60Hz, 30Hz, or NTSC. This message indicates that your IRIS cannot do 60Hz mode. Try 30Hz or NTSC.

No hardware for NTSC mode.

The IRIS has 3 possible display modes out of which at most 2 can be installed. The 3 possible display modes are 60Hz, 30Hz, or NTSC. This message indicates that your IRIS cannot do NTSC mode. Try 30Hz or 60Hz.

No qualifier argument between commas.

You typed in something strange like "1,,10". Try it again, but make sure you put arguments between your commas!

No qualifiers allowed.

You issued a command like "MAP/OUTPUT=map.txt", but there are no qualifiers allowed with the MAP command!

**No room in information tables for another variable name for grid *x*.
Maximum number of variables per grid is *y*; trying to get space for new
variable *z*.**

Indicates that PLOT3D is out of memory. Quit, and see if PLOT3D can be reconfigured to meet your larger memory requirements (call me for help).

No translation for symbol *x*.

You did an '*x*' where *x* is some symbol you wanted to use. PLOT3D is telling you that the symbol *x* is undefined. Try redefining it, by doing "\$*x* := definition" on VAX/VMS (PLOT3D running or not running), or by doing "setenv" from your UNIX shell (PLOT3D not running).

No translation for zero-length symbol ''.

You did a "", which does not make sense. There must be at least one character between the quotes.

Not enough arguments for qualifier *x*.

You did not supply enough arguments for the command qualifier you used. Try using HELP with the command and qualifier in question, for information about what qualifier arguments are expected.

Number of arguments exceeds internal dimension limit of *x*.

You used too many arguments with the last command you issued. If you are confused, try HELP with the command in question.

Number of frames must be greater than or equal to 1.

You were doing PLOT/SCRIPT, and in one of your script commands (such as CENTER, GO, HOLD, or VPOINT) you specified a number of frames that was less than 1. You cannot animate anything in less than one frame(s). Do the command again, and this time make sure to specify a sensible number of frames.

Number of multiple grids exceeds internal dimension of x . y grids are in the file.

You probably tried to read in a multiple-grid XYZ file and PLOT3D is telling you that it doesn't have enough memory for all those grids. It is possible to "re-configure" PLOT3D by changing all occurrences of "MGRID= x " to "MGRID= y " (or larger). See Chapter 9.

Number of qualifier arguments exceeds internal dimension limit of x .

You used too many qualifier arguments (the maximum number is x) with a qualifier in your last PLOT3D command. Try "HELP" in PLOT3D with the command and qualifier, so PLOT3D can tell you what qualifier arguments are expected. Or look up the command in Chapter 7.

Number of qualifiers exceeds internal dimension limit of x .

You used too many qualifiers in your last command. Try "HELP" in PLOT3D with the command, so PLOT3D can tell you what qualifiers are valid. Or Chp. 7.

Opcode x not recognized in frame y of RAS file z . Picture save aborted. File must be fixed before it can be used.

You tried to save your current plot (by hitting the "R" key) and for some reason bad information was encountered in the RAS file. You will probably have to start over again (EXIT or EXIT/SAVE from PLOT3D, and do it all over again).

Orientation x is ambiguous.

You input a plane-orientation (e.g., with the CONTOUR command) that PLOT3D cannot uniquely identify. This could happen if you input "I", which is ambiguous because it could mean "IJ" or "IK".

Orientation x not recognized.

You input a plane orientation (e.g., with the CONTOUR command) that PLOT3D does not recognize. Valid plane orientations are "IJ", "IK", "JK", "XI", "ETA", "ZETA", and "Y". See CONTOURS in Chapter 7.

... Programming error (many of these, call for help)

If you get an error message of this form, report the exact message to me!

Qualifier argument length exceeds internal storage capacity.

You typed in a command with a qualifier argument that is too big to be stored in PLOT3D (as configured on your system). This could happen if you typed "READ/XYZ=superduperfantasticdataset.bin", where "superduperfantasticdataset" would be the questionable qualifier argument.

Qualifier length exceeds internal storage capacity.

You typed in a command with a qualifier that is too long for PLOT3D to store (as it is configured on your system). This could happen if you typed something like: "READ/XYZPDQTURKEYGIZZARDS", where "XYZPDQTURKEYGIZZARDS" would be the questionable qualifier.

Qualifier *x* not recognized.

You typed in a command with a qualifier that is not valid with that command. For example "RAKE/OUTPUT" would give this error.

Ran out of object indices; dimensioned *x*.

You probably are trying to define too complex an image for PLOT3D. Try specifying fewer walls, subsets, et cetera, and see if you can get by. If not, try to reconfigure PLOT3D. Change all occurrences of "MOBJ=100" to "MOBJ=200" (or whatever). See Chapter 9.

Ran out of text object indices; dimensioned *x*.

If you get this error, call me.

Restriction *x* is ambiguous.

You issued a RAKES command, and gave PLOT3D an ambiguous restriction, such as "I", which could mean "IJ" or "IK".

Restriction x not recognized.

You issued a RAKES command, and gave PLOT3D an invalid restriction. Try "HELP RAKES RESTRICT" for a list of valid restrictions.

Shock surfaces are not implemented for 2D plots.

Shock surfaces do not make sense in 2D.

String length too long in routine FSYSTE.

You tried to execute an operating system command from the PLOT3D command line (by typing a "\$" followed by the system command). The command you gave was so long that PLOT3D could not store it or process it. Try to find a more economical way to type the command, or escape to the operating system (simply type "\$" followed by <return>) and try the command there.

String length x too long in call to SYSCOM.

You did a "\$system_command" and system_command is too long. Try something shorter.

String to be translated is too long for routine FGETEN.

First, a definition or two: a symbol in this context is an operating system variable that takes a character string as its value; in PLOT3D, you denote symbols on the command line by enclosing them in single quotes; the PLOT3D command interpreter attempts to expand all symbols into character strings before executing the command line.

If you get this error it means that you gave PLOT3D a symbol name that was too long (i.e., PLOT3D cannot store the name and therefore cannot do anything with it). You must use shorter symbol names. Note: do not confuse the symbol's NAME with its CONTENTS. The NAME you gave was too long. Use a shorter symbol name!

String was truncated on translation. No value returned.

You are probably expanding a symbol (with ' 's), and the definition of that symbol is too long for PLOT3D to handle. Make symbol definition shorter. (Try "HELP SPECIAL")

Symbol translation exceeds internal limit on string length.

You are probably expanding a symbol (with ' 's), and the definition of that symbol is too long for PLOT3D to handle. Make symbol definition shorter. (Try "HELP SPECIAL")

Three numbers are required.

Supply three numbers. Use HELP in PLOT3D if you are not sure which 3 numbers to supply.

Too many arguments for qualifier *x*.

Try giving the command with fewer arguments to qualifier *x*.

Too many arguments: *x* . . .

Try giving the command with fewer arguments.

Too many qualifier arguments: *x* . . .

Try giving the command with fewer qualifier arguments.

Too many bus errors. Program terminated.

The above error message is fairly serious (results in program termination) so you should call me.

Too many floating point exceptions. Program terminated.

The above error message is fairly serious (results in program termination) so you should call me.

Too many operand range errors. Program terminated.

The above error message is fairly serious (results in program termination) so you should call me.

Too many program range errors. Program terminated.

The above error message is fairly serious (results in program termination) so you should call me.

Too many segmentation violations. Program terminated.

The above error message is fairly serious (results in program termination) so you should call me.

Translated string is too long for routine CGETEN.

First, a definition or two: a symbol in this context is an operating system variable that takes a character string as its value; in PLOT3D, you denote symbols on the command line by enclosing them in single quotes; the PLOT3D command interpreter attempts to expand all symbols into character strings before executing the command line.

If you get this error, it means that the character string contained in the symbol was too long for PLOT3D's internal storage. Try using two or more symbols, each one containing part of the original character string.

Unable to find HELP Library File x.

PLOT3D could not find the HELP library file after looking for it in the standard directories. The on-line HELP utility will not work until this is fixed. Call me if you need assistance creating and/or placing the HELP library file.

Unable to make system calls.

You were probably doing a "\$" command and this version of PLOT3D is not set up to do system calls.

Unable to open FUNCTION DATA file *x*.

You were probably trying to READ/FUNC=*x*, and the function file *x* cannot be opened. Perhaps you gave a wrong or misspelled filename. (Quote pathnames).

Unable to open FUNCTION LIST file *x*.

You were probably trying to LIST FUN /OUTPUT=*x*, and for some reason the file *x* cannot be opened for writing. Perhaps the directory you are in is protected.

Unable to open Q DATA file *x*.

You were probably trying to READ/Q=*a*, and the Q file *a* cannot be opened. Perhaps you gave the wrong or misspelled filename. (Quote pathnames).

Unable to open Q LIST file *x*.

You were probably trying to LIST Q /OUTPUT=*x*, and for some reason the file *x* cannot be opened for writing. Perhaps the directory you are in is protected.

Unable to open RAS file *x*.

You tried to save the current plot (you hit the "R" key) and for some reason, PLOT3D was unable to open the RAS file. Make sure that the directory you invoked PLOT3D in is not write-protected, and make sure that you haven't run out of space on your disk. EXIT or EXIT/SAVE out of PLOT3D, check for the aforementioned problems, and start over again. [R is used in version 3.6; use P for 3.5].

Unable to open RAS file *x*. Picture save aborted.

You tried to save the current plot (you hit the "R" key) and for some reason, PLOT3D was unable to open the RAS file. Make sure that the directory you invoked PLOT3D in is not write-protected, and make sure that you haven't run out of space on your disk. EXIT or EXIT/SAVE out of PLOT3D, check for the aforementioned problems, and start over again. [R is used in version 3.6; use P for 3.5]

Unable to open XYZ DATA file *x*.

You were probably trying to READ/XYZ=x, and the XYZ file *x* cannot be opened. Perhaps you gave the wrong filename or misspelled it, or the file is not in your current working directory. Enclose pathnames in quotes. See Chapter 3.3.

Unable to open XYZ LIST file *x*.

You were probably trying to LIST XYZ /OUTPUT=x, and for some reason the file *x* cannot be opened for writing. Perhaps the directory you are in is protected.

Unable to open file *x*.

The error message above has to do with the execution of PLOT3D command files. You probably have them nested too deep. Call me.

Unable to open input file *x*.

The error message above has to do with the execution of PLOT3D command files. You probably have them nested too deep. Call me.

Unable to open output file *x*.

The error message above has to do with the execution of PLOT3D command files. You probably have them nested too deep. Call me.

Unable to start journal file *x*.

NOTE: this could be an ERROR or WARNING depending on whether you are running 3.5 or 3.6.

PLOT3D keeps a journal file that is a record of all the commands you execute during a session. For some reason, PLOT3D cannot start this journal file. Perhaps you are running PLOT3D in a write-protected directory! Get into your own directory and see Chapter 3.3 for directions on quoting pathnames. Also check to make sure you aren't out of disk space. Some versions of PLOT3D *will not plot* without a journal file: if you intend to proceed without a journal file, try a very simple plot before you do a lot of keyboard work.

Unix signalled bus error. Core dump suppressed.

You were running PLOT3D under the UNIX operating system. This error is fairly serious, so give me a call.

Unix signalled floating point exception. Core dump suppressed.

You were running PLOT3D under the UNIX operating system. This error is fairly serious, so give me a call.

Unix signalled operand range error. Core dump suppressed.

You were running PLOT3D under the UNIX operating system. This error is fairly serious, so give me a call.

Unix signalled program range error. Core dump suppressed.

You were running PLOT3D under the UNIX operating system. This error is fairly serious, so give me a call.

Unix signalled segmentation violation. Core dump suppressed.

You were running PLOT3D under the UNIX operating system. This error is fairly serious, so give me a call.

Unknown function number *x*.

The function number you specified does not correspond to an implemented function. Try "HELP FUNCTION" in PLOT3D to get a list of function numbers.

View *x* is not unique.

You did the VIEW command, and gave a view like "X" which PLOT3D cannot uniquely identify; "X" could mean "XY" or "XZ". Give more characters.

View *x* not recognized.

You did the VIEW command, and gave an invalid view. Try "HELP VIEW" in PLOT3D for a list of valid views. Or see VIEW in Chapter 7.

You need at least 28 bit planes to do z-buffering.

You did a PLOT and attempted to z-buffer it on the IRIS, but your workstation is not equipped with enough bit planes (image memory) to handle z-buffering.

Zero is NOT a good increment.

You input a (start, end, increment) set (in WALLS, SUBSETS, or whatever) and specified 0 as the increment, which does not make sense.

Zero length argument.

Perhaps you typed in a command, and forgot to give it an argument. Make sure that you supply argument(s) for commands that need 'em. Commands, with qualifiers and arguments, are listed alphabetically in Chapter 7.

Zero length qualifier argument.

Perhaps you typed in something like "PLOT/UP=" by mistake. Try the command again, and make sure you give qualifier argument(s) for each qualifier that needs 'em. Commands, with qualifiers and arguments, are listed alphabetically in Chapter 7.

Zero length qualifier.

Perhaps you typed in something like "PLOT/" by mistake. Try the command again with proper qualifiers.

Zero length string for START, END, or INC.

You were doing a command such as WALLS that requires (START, END, INC) sets. The parser could not make your input into a valid (START, END, INC) set, so start all over again. See tutorial in Chapter 3.6 for WALLS and Chapter 4, sections 4.8 through 4.11 for SUBSETS.

WARNINGS

10.3 Warning Messages

PLOT3D warning messages have the following format:

Warning.

blah blah blah. . .

...

Here is the list of warning messages with comments and suggestions:

^C interrupts from the keyboard work, except when PLOT3D is prompting for input. A ^C then is fatal!

Do not use ^C while PLOT3D is waiting for input. Interrupting a FORTRAN READ statement puts PLOT3D in an infinite loop (VAX/VMS only)! ^Y should work to kill PLOT3D.

***x* is the maximum number of rakes allowed, due to internal program dimension MRAKE.**

You have the maximum number of RAKES allowed right now (i.e., no more can be added). If you want more traces, try getting more into each RAKE, or reconfigure PLOT3D so that it can deal with more RAKES. Try "SHOW RAKE" in PLOT3D to see all the specifications for the RAKES you have created.

***x* is the maximum number of subsets allowed per grid, due to internal program dimension MSUBS.**

You have the maximum number of subsets allowed per grid right now (i.e., you cannot add any more subsets to whatever grid you just specified a subset for). If you need more subsets, reconfigure PLOT3D (call me for help on reconfiguration, or see Chapter 9).

***x* is the maximum number of walls allowed per grid, due to internal program dimension MWALS.**

You have the maximum number of walls allowed per grid right now (i.e., you cannot add any more walls to whatever grid you just specified a wall for). If you need more walls, reconfigure PLOT3D (call me for help on reconfiguration, or see Chapter 9).

2 plane color map used. Only available colors are BLACK,RED,GREEN,YELLOW. Since this is a little ridiculous, may I suggest you use PLOT/IRISSINGLE instead?

You are running PLOT3D on an IRIS workstation that has only a few bit planes of image memory. You did a PLOT/IRIS or a PLOT (with default of /IRIS). If you want more than four colors, do PLOT/IRISSINGLE.

**A complex shaded image can tie up the printer for hours!
Please be courteous to other users.**

You did a PostScript screen dump ("D" key) while using PLOT3D on your IRIS. This creates a PostScript file that is basically a raster image of the graphics window. Try to print these files when printer usage is low, because these files can take a long time to print. The "P" key (PostScript draw) might produce a file that will print quicker, but be courteous just in case.

Ambiguous command *x*.

The command you entered could not be uniquely identified. For example "R" could be "RAKES" or "READ".

Attributes for contour level *x* not found for label processing in subroutine MYCON. Default attributes will be used.

You are running the DISSPLA version of PLOT3D and were doing some kind of contour plot. Try using slightly different contour levels, or try doing a contour plot with the same attributes for each contour level. If the problem persists, give me a call.

BINARY writes not implemented, using UNFORMATTED.

I'll bet you tried to do a LIST/BINARY in PLOT3D. For whatever reason, BINARY data format is not supported for your system, so FORTRAN UNFORMATTED is being used instead!

BINARY reads not implemented, using UNFORMATTED.

I'll bet you tried to do a READ/BINARY in PLOT3D. For whatever reason, BINARY data format is not supported for your system. Use FORTRAN UNFORMATTED or FORTRAN FORMATTED files instead (READ/UNFORMATTED or READ/FORMATTED respectively).

Blanking is not supported for this function.

You probably are using function number 157, Normalized 2D stream function. Your XYZ data have IBLANK data, but PLOT3D will not support BLANKing for this function. If you want to use this function, make a new XYZ file that does not contain IBLANK data.

Can't get interpolation for point x in trace y , rake z , corresponding to grid point p q r in grid n Point will be blanked out.

The above warning means that you were doing particle traces (probably colored by a scalar function) and for some reason PLOT3D cannot get a proper interpolation. If your plot looks weird, give me a call.

Can't get interpolation for point x in trace y , rake z , corresponding to grid point p q in grid n Point will be blanked out.

The above warning means that you were doing particle traces (probably colored by a scalar function) and for some reason PLOT3D cannot get a proper interpolation. If your plot looks weird, give me a call.

Changing display modes for more than one window. This may cause an incorrect blanking of windows on the screen. Relocate windows to restore correct visibility.

When running PLOT3D (IRIS version) under MEX, with several plots on the screen, you must be careful when changing display modes (such as when you toggle z-buffering on or off with the "Z" key) because some of the windows may not be drawn correctly. Move the windows slightly to force MEX to redraw them correctly.

**Data is in a nonstandard form of 2D--IDIM, JDIM, or KDIM is 1,
and the corresponding X, Y, or Z is not all constant.
Derivatives are incorrect.**

You did a PLOT or a LIST FUNCTION or a SHOW FUNCTION and the last FUNCTION you selected was divergence of velocity (e.g. 158) or perhaps a vorticity function (e.g., 180-188,201,204,301), or perhaps a function in which a gradient was calculated (e.g., 190-193,210-211). Your data set is 3D (you used READ/3D), but at least one of IDIM, JDIM, or KDIM (try HELP READ File_Formats in PLOT3D) is equal to 1, and the corresponding Cartesian component (x, y, or z) is not constant. This is a non-standard format that falls somewhere in between 2D and 3D, hence the confusion. You should use a data set that is genuine 3D (i.e. where IDIM, JDIM, and KDIM are all greater than 1), or use a data set that is genuine 2D (i.e. use the 2D file format, or use the 3D file format and make sure that your grid lies in a Cartesian plane (xy, xz, or yz)).

**Didn't get a good solution from CUBIC:
Parametric cubic matrix (PCMAT)
n1 n2 n3 n4
Cubic polynomial coefficients
m1 m2 m3 m4
CUBIC returned k roots: r1, r2, r3, ...**

You did a CONTOUR/CUBIC followed later by a PLOT. For whatever reason, the routine that does the parametric cubic interpolation was unable to get a good solution. This means that your plot may look funny. Try using CONTOUR/LINEAR instead.

Don't know how to handle subsets that are not a single plane.

You tried to PLOT and your current FUNCTION is 10 (2D crossing grid line check). You specified a subset that is not a single plane (should be IJ, IK, or JK). Specify a SUBSET that is a single plane (e.g. for an IJ plane at K=1, type "all" for the I and J subsets, and type "1" for the K subset).

WARNINGS

Extrapolating for point x in trace y , rake z , corresponding to grid point $p q$ in grid n .

PLOT3D is telling you that it is computing a trace point outside of the grid. This is probably nothing to worry about.

Extrapolating for point x in trace y , rake z , corresponding to grid point $p q r$ in grid $.n$

PLOT3D is telling you that it is computing a trace point outside of the grid. This is probably nothing to worry about.

Feedback buffer full in subroutine PSDMP. Buffer is dimensioned x . Plot may be incomplete.

You tried to make a postscript file by hitting the "P" key. The plot you are trying to save is so complex that it cannot be stored in PLOT3D's memory. You can either make your plot less complex, or you can try to reconfigure PLOT3D by increasing parameter MBUF in PSDRW.

Feedback buffer full in subroutine PSDRW. Buffer is dimensioned x . Plot may be incomplete.

You tried to make a postscript file by hitting the "P" key. The plot you are trying to save is so complex that it cannot be stored in PLOT3D's memory. You can either make your plot less complex, or you can try to reconfigure PLOT3D.

Have to extrapolate for point x in trace y , rake z , corresponding to grid point $p q$ in grid n .

PLOT3D is telling you that it is computing a trace point outside of the grid. This is probably nothing to worry about.

High-quality PostScript screen save has not been implemented for the IRIS 4D/GT.

You tried to do a PostScript high-quality screen save ("P" key in PLOT3D 3.6) on an IRIS 4D/GT. This form of PostScript output is not supported for these machines (but is supported on IRIS 2xxx's and 3xxx's). You could try using PostScript screen dump instead ("D" key), which produces a PostScript raster image type of file rather than a PostScript file with move's and draw's in it; note however that PostScript screen dump files can take HOURS to print, especially if the image is complex.

In grid x , IBLANK(I, J,1) points to grid y which does not exist. Value will be ignored.

Your IBLANK data are apparently incorrect. Make sure that the IBLANK values in your XYZ file make sense (try "HELP BLANK" in PLOT3D). If your IBLANK values are ok, maybe your XYZ and/or Q data are not correct.

In grid x , IBLANK(I, J, K) points to grid y which does not exist. Value will be ignored.

Your IBLANK data are apparently incorrect. Make sure that the IBLANK values in your XYZ file make sense (try "HELP BLANK" in PLOT3D). If your IBLANK values are ok, maybe your XYZ and/or Q data are not correct.

Internal dimensions in HELP prevent accepting more than x keywords.

You tried to use the PLOT3D HELP command with quite a few keywords (e.g., "HELP PLOT /IRISSINGLE /IRIS /IRISRGB /UP /SCRIPT VPOINT SHOW MAP", which is incorrect anyway). Try using fewer keywords, and let PLOT3D tell you what subtopics (keywords) you can use from there.

Invalid color number x in COLNAM, reset to WHITE.

If you get this, there is a bug in PLOT3D.

Invalid color number x in NAMRGB, value returned for BLACK.

If you get this, there is a bug in PLOT3D.

WARNINGS

Invalid line style number x in LINSTY, reset to SOLID.

If you get this, there is a bug in PLOT3D.

Invalid symbol type x in SYMTYP, reset to 0.

If you get this, there is a bug in PLOT3D.

Keyboard attention interrupts not enabled.

Some versions of PLOT3D (like the VMS/DISSPLA version) allow the user to do things like abort a plot by hitting the <BREAK> key. The version you are running does not allow things like <BREAK>ing.

Line stipple pattern h not recognized in subroutine PSDRW. Resetting line style to SOLID.

You did a PostScript high-quality screen save ("P" key in PLOT3D 3.6, IRIS only). The PostScript output subroutine encountered a line-style that it did not recognize, and therefore is going to produce solid lines instead. Try using a different line style (try "HELP ATTRIBUTES" in PLOT3D or see Attributes, Chp. 7.2). Some examples of PLOT3D line styles are "SOLID", "DASH", "DOT", "CHAINDASH", "CHAINDOT", and "NONE"..

Line style pattern x not recognized in subroutine PSDRW. Resetting line style to SOLID.

You have a plot on the IRIS monitor that has non-solid lines in it. You hit the "P" or "D" key to get a Postscript file, and for some reason PLOT3D was unable to draw the line style on the monitor to the Postscript file, so it drew a solid line to the Postscript file instead. Try printing the Postscript file. If you really want non-solid line styles, try using different non-solid line styles. See Attributes, Chp. 7.2.

New problem dimensions do not match old problem dimensions.

You probably read in a grid file and a solution file whose dimensions (IDIM, JDIM, KDIM, etc.) don't match. Check the files carefully. PLOTing will probably not work until you straighten this out. Make sure that you are reading in grid and solution files that are supposed to go together!

No current window to overlay onto-- making new plot instead.

You did a PLOT/OVERLAY, but there were no plots being displayed. Since there is nothing to overlay onto, PLOT3D is making a new plot.

No points active in any of the grids.

Your SUBSETS command specifies nothing (you probably did "SUBSETS" followed by a couple of <RETURN>'s). Try setting your subsets to make sure that some points are active. To check your subsets type "SHO SUBSETS". For help with subsets type "HELP SUBSETS". Chapter 4 gives SUBSETS examples.

No points in the rake that way. Start over.

Probably you did a RAKE command and hit a <RETURN> by mistake at the J or K prompt. Just do it over again. Chapter 4.11 shows an example.

No points in the subset that way. Start over.

Probably you did a SUBSET command and hit a <RETURN> by mistake at the J or K prompt. Just do it over again. Chapter 4 gives SUBSETS examples

No points in the wall that way. Start over.

Probably you did a WALL command and hit a <RETURN> by mistake at the J or K prompt. Just do it over again. Chapter 3.6 gives WALLS examples.

No Q data for grid x

You have not READ in any Q data for grid number x .

No XYZ data for grid x

You have not READ in any XYZ data for grid number x .

Only true 2D data will be computed correctly. Grid x is not standard 2D.

Basically what you have is a 3D data set (an I, J, and K) that has only one computational plane (e.g., KDIM is 1) but varies in X, Y, and Z (i.e., computationally

WARNINGS

it is 2D, but geometrically it is 3D). PLOT3D cannot calculate the current function because it cannot be sure what all the derivatives should be (it makes no assumptions).

Particle trace length exceeds internal dimension of x Trace will be truncated.

You PLOTEd particle traces (FUNCTION number 300), and one of the traces has so many points that it cannot be stored in PLOT3D. Try to reconfigure PLOT3D so that it can handle your traces, if you need all the points in the trace.

PLOT/OVERLAY has not been implemented for GRAFIX.

You were running PLOT3X (version of PLOT3D that creates ARCGRAPH files), and tried to do a PLOT/OVERLAY. The /OVERLAY qualifier does not make sense in PLOT3X, so don't use it.

Prompt string truncated to x .

This warning would indicate a surprising PLOT3D bug. Call me.

SCRIPT files have not been implemented for DISSPLA.

PLOT/SCRIPT will not work on VAX/VMS (DISSPLA) version of PLOT3D.

SCRIPT files have not been implemented for GRAFIX.

PLOT/SCRIPT will not work on GRAFIX (ARCGRAPH) version of PLOT3D.

SCRIPT files have not been implemented for IRIS_RGLA.

PLOT/SCRIPT will not work on the Cray2/IRIS (RGL) version of PLOT3D.

SHOW not available for command x .

You tried to do something like "SHOW READ", and SHOW does not work for commands like READ. Try "HELP SHOW" in PLOT3D for a list of commands that will work with SHOW. Or see SHOW in Chapter 7.

Unable to overlay onto requested window-- making new plot instead.

You did a PLOT/OVERLAY=n, where n is a PLOT3D graphics window number. Perhaps you typed in a bogus number by mistake... the window you were trying to overlay onto probably doesn't exist. Anyway, PLOT3D will simply open a new graphics window and make a new plot. Call me if this is troublesome.

Unable to start journal file x.

NOTE: this could be an ERROR or WARNING depending on whether we are running 3.5 or 3.6.

PLOT3D keeps a journal file that is a record of all the commands you execute during a session. For some reason, PLOT3D cannot start this journal file. Perhaps you are running PLOT3D in a write-protected directory! Get into your own directory and see Chapter 3.3 for directions on quoting pathnames. Also check to make sure you aren't out of disk space. Some versions of PLOT3D *will not plot* without a journal file: if you intend to proceed without a journal file, try a very simple plot before you do a lot of keyboard work.

Unable to use RGB mode while running under the window manager. Using IRISSINGLE instead.

You tried to PLOT/IRISRGB while running PLOT3D in MEX. IRIS RGB mode cannot be used if you are running PLOT3D under mex (the window manager), so IRIS single buffer mode is used instead.

Variable name x for grid y already exists, but will be deleted.

This message has to do with PLOT3D's method of storing computed data. It is probably nothing to worry about. Continue as you were.

Z-buffering is enabled--using single buffer mode instead.

You tried to PLOT after PLOTing and z-buffering a previous image. Z-buffering is still enabled, and so PLOT3D cannot use double-buffer mode (the default). To return to double-buffer mode, hit the "Z" key.

Zero or negative density at point x , in grid y .

You probably got this message while READING your Q data. It means that the density (Q1) value supplied in your Q file at point x , grid y , is zero or negative, which makes no physical sense. Try listing your Q data out to a text file (try "HELP LIST" in PLOT3D for help on how to do this) so you can find the offending value(s). Better yet, try plotting density! See LIST in Chapter 7.

Zero or negative metric Jacobian at point x , in grid y .

The metric Jacobian supplied in your Q file is zero or negative, and hence makes no physical sense. Perhaps you read in a Q file that does not contain Jacobian information with the /JACOBIAN option active; if this is the case, try READING in your Q data with the /NOJACOBIAN option. If you meant to read in a Q file that has Jacobian information, then you have bad Jacobian value(s). Try LISTing out your Q data to find the offending value(s) (try "HELP LIST" in PLOT3D). Or see LIST in Chapter 7.

Zero or negative pressure at point x , in grid y .

The pressure at point x , in grid y is zero or negative, which makes no physical sense. Perhaps your Q data are wrong. Try LISTing out your Q data and/or LISTing out FUNCTION data for the pressure function (try "HELP LIST" in PLOT3D, and remember to set the current function to pressure (with the FUNCTION command) if you want to LIST pressure data). Better yet, try plotting pressure!

^C interrupts from the keyboard work, except when PLOT3D is prompting for input. A ^C then is fatal!

Do not use ^C while PLOT3D is waiting for input. Interrupting a FORTRAN READ statement puts PLOT3D in an infinite loop (VAX/VMS only)! ^Y should work to kill PLOT3D.

x is the maximum number of rakes allowed, due to internal program dimension MRAKE.

You have the maximum number of RAKEs allowed right now (i.e., no more can be added). If you want more traces, try getting more into each RAKE, or

reconfigure PLOT3D so that it can deal with more RAKES. Try "SHOW RAKE" in PLOT3D to see all the specifications for the RAKES you have created.

x is the maximum number of subsets allowed per grid, due to internal program dimension MSUBS.

You have the maximum number of subsets allowed per grid right now (i.e., you cannot add any more subsets to whatever grid you just specified a subset for). If you need more subsets, reconfigure PLOT3D (call me for help on reconfiguration, or see Chapter 9).

x is the maximum number of walls allowed per grid, due to internal program dimension MWALS.

You have the maximum number of walls allowed per grid right now (i.e., you cannot add any more walls to whatever grid you just specified a wall for). If you need more walls, reconfigure PLOT3D (call me for help on reconfiguration, or see Chapter 9).

Chapter 11. Sample Command Files

11.1 Summary

Several command files are presented here, including some that are only a few lines long. The 2D example in 11.2 shows how simple a PLOT3D session can be. Two advanced topics in PLOT3D (comparison of computational and experimental data, and using the Computer Graphics Library to run PLOT3D on the Cray) are illustrated by example. A picture of each resultant plot is shown or referenced.

The creation and use of command files are discussed in Chapter five; this chapter just gives examples, with explanations specific to the examples.

PLOT3D prompts are not part of a command file; see the tutorial (Chapters 2-6) for an idea of how these scripts fit into PLOT3D. In this chapter, a blank line indicates a <return>, and each line ends in a return; in the tutorial, the returns were printed out for you, but they are not visible in a command file. I have added in the horizontal lines, to indicate how many returns are used. Your command files should look like those below, but without the horizontal lines. As discussed in Chapter 5, command files are created in an editor utility, then executed in PLOT3D with the *@filename* command.

Before you could run a command file, you would have to have the data. The last command file in these examples uses the *wingbodytail* data set, which, along with the *wingbodytail* command file (*wbt.com*) is available in the *wingbodytail* directory. Directions for obtaining *wingbodytail* are in Chapters 2.4, 2.5, 2.6, 3.3 and 3.4, and in Appendix F. The *examples* directory also contains the data sets *airfoil* (2D) and *bluntfin*, and related command files. New users often experiment with these examples.

The following list indicates section numbers, topics and authors:

- 11.2 2D pressure coefficient using TWING: Scott Thomas
- 11.3 3D comparison of computational and experimental data, Space Shuttle: Pieter Buning
- 11.4 RGL— running PLOT3D on the Cray: Robert Meakin
- 11.5 Limiting contour lines to Mach 1: Pieter Buning


```
TWING Solution
wall
a
1
last
subset
a
1
last
plot/line/2d
```

11.3 3D Comparison of Computational and Experimental Data, Space Shuttle

Command file by Pieter Buning

These are the command files for the picture in the first color plate, showing the comparison of computational and experimental data for the space shuttle. Before these files were used, a simple program was written that combined the computational and experimental data into one multiple-grid data set. That is, the combined data set contained one group of grids for the experimental data, and another group of grids for the computational data.

PLOT3D USER'S MANUAL

There are three .com files involved: *top.com* (the top view of the shuttle) calls *surf.com* (the subsets for the surface) and *cons.com* (color contours). Together, these command files produce three plots, which are combined in GAS to produce the final image. Notice that the first command file (*top.com*) plots three times: the first plot shows the grid lines of all the subsets that are going to get color contours. The purpose of this wireframe model is to produce something simple to manipulate in GAS, to position the parts correctly. A color-contoured model would have been much too cumbersome to move around. The next plot shows the color contours, that is, the colors on the surface of the model, representing the various pressure coefficients on the surface. The third plot creates a thick white line to divide the halves of the final image; the white line is used as an overlay (foreground) in GAS.

The other two command files are called from within the *top.com* file, with the @ symbol.

These plots were prepared for GAS by running these command files through PLOT3X. Chapter 5 gives more information on GAS and PLOT3X. GAS is frequently used (on IRIS workstations) to combine objects created with PLOT3D and SURF, and to add titles. The titles on this image were added in GAS.

See figure 1, at the beginning of this manual, to see what this looks like.

top.com

```
re/mq/noc/x=surfx/q=surfq
min/inc -100 3000 100 -500 500 100 0 800 100
vp/a -90 90 3200
@surf
fu 1
p/full/noax
fu 114
@cons
p
wall/gr=5
a
2
1
```

wh

2

a

38

1

wh

2

wall/gr=7

a

2

1

wh

2

a

76

PLOT3D USER'S MANUAL

1

wh

2

fu 0

p

surf.com

subs/gr=1
a

1 17

1

subs/gr=5
a

2 38

1

subs/gr=6
a

a

1

11.4 CGL: Running PLOT3D on the Cray

Command file by Robert Meakin

CGL is a distributed graphics library that is sent out with any request for Cray/IRIS PLOT3D. To run PLOT3D on a Cray while sitting at a workstation, simply log into the Cray, and type `plot3d`. When you are ready to use the `plot` command, type `plot` and PLOT3D will prompt you for the ID of your workstation. You would respond `wk00`, for example, to tell the Cray what workstation you are using. This works only if you have identical accounts on the Cray and the workstation. If you do not (for example, if you are on Chewbaka), you need to do a manual startup of LCL. Get back on the IRIS, type `plot3d_lcl &` then telnet to the Cray and proceed as usual. The program `plot3d_lcl` has to be running in the background on the IRIS. In version 3.6, you can hit `<return>` when PLOT3D asks for workstation number, and that should start LCL.

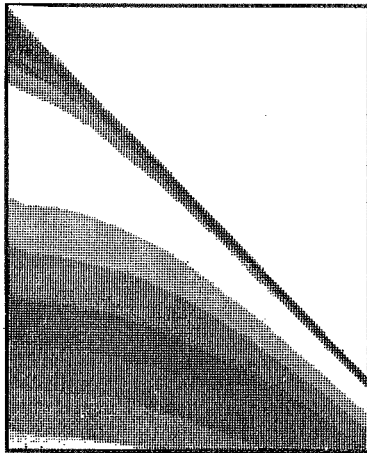


Figure 82. Distributed processing example: color_contours. The output is the same as non-distributed processing; the difference is in the speed of calculations.

"Distributed processing" is the term used when the process of making plots is distributed between the workstation and the supercomputer. The resultant plot is no different than it would be if it had been created on the workstation alone; it is just created in less time (unless you are in a long queue for the Cray!). The following command file is no different from ordinary command files, except for the line (immediately after the PLOT statement, near the end) that identifies the workstation. It runs on the supercomputer because you are logged on to the supercomputer when you run it.


```

whi
_____
_____
_____
_____
_____
_____
_____
p/noax/full
_____
wk25.nas.nasa.gov
_____
_____
quit/save
_____

```

11.5 Limiting Contour Lines to Mach 1

Command file by Pieter Buning

This is a fragment from the command file we used in the testdrive. The complete file, and two others are in the *wingbodytail* directory. See sections 2.4, 2.5, and 2.6, and Appendix F for more information about the location of this directory.

The read command is not included in this segment; this part just shows how to create solid walls and plot only Mach 1. See Chapter 2.13 for an illustration of this plot. Lines beginning with ! are comments.

```

! Mach 1 surface.
_____
!
_____
! Use solid walls.
_____
!
_____
wall/grid=2
_____
a
_____
_____
a
_____
_____
1
_____
_____
s
_____

```

PLOT3D USER'S MANUAL

```
_____  
_____  
0 0 0  
_____  
wall/grid=3  
3 64  
_____  
1 21  
_____  
1  
_____  
s  
_____  
_____  
0 0 0  
_____  
wall/grid=4  
3 54  
_____  
1 11  
_____  
1  
_____  
s  
_____  
_____  
0 0 0  
_____  
subset/grid=2  
a  
_____  
a  
_____  
a  
_____  
_____  
_____
```

subset/grid=3

a

a

a

subset/grid=4

a

a

a

contour/man 1

line

red

2

ik

plot

Chapter 12. Conclusion

12.1 Summary

This chapter presents a chart of the origin and uses of different types of files in PLOT3D, and a list of commands in typical sequence.

12.2 Filename Extension Chart

The following chart, figure 83, shows the various types of files associated with PLOT3D, organized alphabetically by filename extensions. The first column lists where the files come from, and the last column lists where they go. Two other filetypes, .pol and .prs, which are used in GAS, are subsets of .gra files, and represent flat shaded polygons and function data (e.g., pressure), respectively.

From:	Ext:	Definition:	To:
grid and flow solver programs, or from LIST in PLOT3D	.bin	binary data	PLOT3D or SURF or other
system editor or .jou file	.com	command file to run PLOT3D (plot3dini.com = initialization)	PLOT3D
see .bin	.dat	FORTTRAN unformatted data	PLOT3D
PLOT3D plot/dip DISSPLA only	.dip	Device Independent Plotter file	various output devices
see .bin	.fmt	formatted data	PLOT3D
PLOT3X	.gra	ARCGraph file, graphic calculations for each grid point	GAS
PLOT3D exit/save	.jou	journal file; record of all keyboard entries (rename it .com!)	PLOT3D
PLOT3D rake/write=filename	.par	particle trace calculation	PLOT3D or MRAKEVEC
PLOT3D. while plot is on screen, type p (version 3.6 only). IRIS only.	.ps	PostScript screen save (moves and draws). (Typing d => pixel dump, NOT recommended)	PostScript laser printer
PLOT3D. while plot is on screen, type q	.qms	screen save (one frame)	QMS 1200 laser printer
PLOT3D. while plot is on screen, type s for graphics window or f for full screen. IRIS only.	.sei	screen save (one frame)	Seiko or Tektronix
PLOT3D. while plot is on screen, type p (versions through 3.5) or r (version 3.6). IRIS only.	.ras	pixel file screen save can hold multiple frames	RASP GAS
MRAKEVEC, SURF	.vec	colors, moves and draws, for lines	GAS

Figure 83. Filename extension chart, summary.

12.3 Typical Command Sequence

The following list shows all the PLOT3D commands, in the order in which they might appear during a PLOT3D session. The most-used commands are in boldface.

READ
TEXT
WALLS
SUBSETS
FUNCTION
CONTOURS
SHOW
HELP
MINMAX
VPOINT
VIEW
PLOT
FSURFACE
PLOT
LIST
FUNCTION
VECTORS
CLEAR
PLOT
FUNCTION
RAKES
MAP
PLOT
EXIT
QUIT

Appendix A. Version 3.6

By Pieter Buning

(Pieter Buning released version 3.6 of PLOT3D in February 1989, and Version 3.6b in November 1989. Following are excerpts from the new on-line HELP file, indicating most of the new features. To receive version 3.6, follow the directions in the Forward of this manual. A complete Version 3.6b HELP document is available on-line in the *documentation* directory; it can be accessed with the HELP command. See Chapter 2.19.)

Release_notes

3.6

New_features

New versions:

- o DI-3000 versions for both VAX/VMS and UNIX available, courtesy LTV Aircraft Products Group, Dallas TX, with enhancements by Flow Industries, Inc., Kent, WA, and Jacques Middlecoff, General Electric Company, NAS User Services.
- o SUN/SunCore version of PLOT3D (Version 3.4 only) available, courtesy General Electric Company and Sun Microsystems, Inc.
- o Apollo/UNIX and GMR3D version is available; Apollo/AEGIS will no longer be supported.
- o CRAY/COS and IRIS 1xxx versions are no longer supported.

New commands/options:

- o CONTOURS/LINEAR/CUBIC allows the use of parametric cubic interpolation for contour lines/shock wave lines instead of the usual linear interpolation. This helps remove "scalloping" from contours on highly stretched, high aspect ratio grids.
- o PLOT/OVERLAY =window# allows plots to be overlaid on top of each other. Implemented for IRIS only.
- o PLOT/TITLE/NOTITLE turns on/off the function name (top line of the title text). Title lines entered with the TEXT command will still be plotted.
- o PLOT/BAR/NOBAR turns on/off the contour level text.
- o PLOT/ADDITIONAL_TEXT/NOADDITIONAL_TEXT turns on/off the additional text information (Mach number, grid size, etc.) (Please feel free to use abbreviations for all commands and qualifiers!)
- o LIST/FORMATTED fixed to produce files that are compatible across different machines, and to produce files which do not exceed 80 characters per record.
- o PLOT/NOSCRIP implemented to turn off script command processing.
- o DELETE and RETURN commands added for PLOT/SCRIPT mode to delete or leave a graphics window. See "HELP PLOT Script *".
- o PLOT/SURFACE option renamed to PLOT/FSURFACE to avoid confusion with the contour type SURFACE.

New features:

- o Dynamic memory allocation has been implemented for all Unix and VAX/VMS versions.
- o Use of PLOT3D with incompressible flow documented. See "HELP FUNCTIONS Nondimensionalizations".
- o Search for Help file uses simple path, (\$HOME/bin,/usr/local/bin, /usr/unsupported/bin,/usr/unsupported/man) for Unix, SYS\$LOGIN, SYS\$SYSTEM, SYS\$HELP, STT0:[BUNING.PLOT3D], DUA0:[CFDLIB.PLOT3D] for VAX/VMS.
- o Maximum number of points in a particle trace increased to 4000; default set to 2000.
- o Perspective field-of-view has been reduced from 30 deg. to

- 20 deg. (except DISSPLA). This matches GAS.
- o Arrowhead half-aspect ratio changed from 11 to 6.67.

New functions:

- o Added functions "Helicity density" (function 186), "Relative helicity" (187), and "Filtered relative helicity" (188), for plotting vortex cores. See "HELP FUNCTIONS Function_definitions Helicity".
- o Added "Divergence of velocity" (158), useful for plotting shocks. See "HELP FUNCTIONS Function_definitions Divergence_of_velocity".
- o Added "2D crossing grid line check" (function 10), "Tetrahedron decomposition cell volume check" (11), and "Tetrahedron decomposition grid crossing check" (12). See "HELP FUNCTIONS Function_definitions Grid_checks".
- o Added "Log of normalized density" (104), "Log of normalized pressure" (119), and "Log of normalized temperature" (124), useful for plotting high-supersonic flows.
- o Added vector function "Density gradient" (211), scalar functions "Pressure gradient magnitude" (192) and "Density gradient magnitude" (193).

IRIS improvements:

- o Added PostScript screen DRAW hardcopy capability, tied to "P" key (.RAS file screen dump has been moved to "R" key). This is a high-quality image, as it uses moves and draws rather than a (pixel) screen dump. See "HELP PLOT IRIS_interface PostScript_screen_saves". IMPLEMENTED FOR IRIS 2XXX/3XXX; MAY work on IRIS 4D/70; will NOT work on 4D/70GT.
- o PLOT/FIGURE=(areax,areay,charht) option implemented (fixed plot size, no text).
- o INCLUDE file "fglbugs.h" created to isolate and correct bugs in IRIS Graphics Library INCLUDE file <fgl.h>.
- o IRIS 2xxx,3xxx and 4D versions consolidated by using CPP (UNIX C Preprocessor) statements.
- o Added PostScript screen DUMP (pixel image) hardcopy capability,

tied to "D" key ("D" for dangerous). This can tie up a printer for HOURS! See "HELP PLOT IRIS_interface PostScript_screen_saves".

UNIX improvements:

- o The UNIX interface has been totally rewritten to be portable: the interface is identical between the IRIS, CRAY 2, CONVEX, SUN, and BSD 4.3 VAX. Several standard FORTRAN/C calling conventions have been coded, and a particular convention can easily be selected via the INCLUDE file "fortcall.h". Use of CPP (UNIX C Preprocessor) with both FORTRAN and C source files is required.
- o Keyboard interrupts (^C) have been enabled.
- o Floating point exception trapping has been enabled where possible.

REMOTE (e.g. CRAY 2)/IRIS improvements:

- o Added option for manual startup of PLOT3D_LCL.
- o Startup of PLOT3D_LCL via the UNIX "rsh" or "remsh" command now includes a crude PATH capability, checking \$HOME/bin, /usr/local/bin, and /usr/unsupported/bin directories.

Bugs_fixed

Commands/options:

- o LIST/FORMATTED fixed to produce files that are compatible across different machines, and to produce files which do not exceed 80 characters per record.
- o Added PLOT/NOSCRIPt option to turn off script command processing.

Features:

- o IRIS hidden line attribute (WALLS, SUBSETS) has been fixed.
- o Reading .PAR files with too many points per line is now handled (trace lines are truncated).
- o If a new file is to be created and a file of that name already exists, it is deleted first (otherwise a FORTRAN run-time error

- would result).
- o Changed confusing error message about an "insidious bug" and "automatic paging" to read "Internal storage exhausted while getting space for variable xxx, grid n... Automatic paging of variables has not been implemented."
- o Fixed possible divide-by-zero in contour level scaling.

Functions:

- o Computation of "Stagnation pressure coefficient" (115) corrected.
- o Several 2D particle tracing bugs have been fixed.
- o Handling of extrapolation improved for coloring of particle traces by a scalar function.
- o Bug in restoring subsets after drawing particle traces colored by a scalar function corrected.
- o Particle tracing at grid singularities fixed.
- o Bad warning message from particle tracing colored by a scalar function corrected (had caused some spectacular IRIS crashes).
- o Fixed bad dimensions in 3x2 matrix inversion, used for restricted particle tracing.
- o Added a maximum number of steps to initially find a point for particle tracing.

IRIS interface:

- o Switching from 60Hz to 30Hz or NTSC display modes fixed.
- o Backface polygon removal initialization fixed.
- o Seiko screen dump now reads the correct screen buffer.

GRAFIX interface:

- o Fixed bug in drawing translated 3D polygons (used when plotting function surfaces).

GMR3D interface:

- o Bug in 2D axis routine fixed.

VMS interface:

- o Help File access has been corrected.
- o Translation of logical device names into physical device names fixed.

AEGIS interface:

- o Added code-arounds for FORTRAN compiler bugs.

CRAY 2 interface:

- o Removed all code-arounds for CFT77 (compiler has improved).

Known_problems

Listed are known or potential problems with various versions of PLOT3D.

IRIS

- o Output to the journal file "plot3d.jou" is not flushed if PLOT3D is killed, so the file may be incomplete.
- o On the IRIS 2xxx/3xxx, while running under MEX, holding the "hogwhiledown" button down when PLOT3D tries to attach to the graphics window makes MEX go bye-bye. To recover, type "gclear", kill PLOT3D (if it ain't dead yet), and restart MEX. The "attached" window is the one with the highlighted (usually red) border. For info on "hogwhiledown", see "HELP PLOT IRIS_interface".
- o PostScript screen draw MAY NOT work for IRIS 4D/70, and WILL NOT work for 4D/70GT.
- o IRIS 2xxx/3xxx version ignores ILLEGALOP floating point

- exceptions because these are sometimes generated incorrectly due to a FORTRAN compiler bug.
- o Z-buffering an image with dashed, dotted, chaindashed, or chaindotted linestyles will hang the IRIS 2xxx/3xxx.
- o Changing from single- to double-buffer mode or vice-versa (such as when turning z-buffering on or off), when using the window manager and more than one plot is on the screen can cause the screen to be refreshed incorrectly. Move some windows slightly to force MEX to update the screen. (IRIS 2xxx/3xxx)
- o In general, and especially at the command prompt level, PLOT3D does not keep its graphics windows "idling" with a "swapbuffers" (see "Silicon Graphics PIPELINE", Vol. II, No. 1, Fall 1985, p. 12). Thus under the window manager (and double buffering), another graphics program will hang until PLOT3D swaps buffers.
- o For very complicated plots and/or particle trace plots, PLOT3D has been observed to respond VERY sluggishly. Response to the mouse may take on the order of a minute (keep that mouse button down!). Further screen updates may be somewhat faster.
- o PLOT3D will not update the screen correctly if the window manager has been started up using "mex -d" (double-buffered). (IRIS 2xxx/3xxx)
- o PLOT3D uses A LOT of (virtual) memory. IRISes at NASA Ames have been reconfigured with 24MB of swap space. This involves repartitioning the system disk (see the "IRIS Series 3000 Owner's Guide, Version 2.0" or your SGI rep.).

REMOTE/IRIS

- o When PLOT3D is interrupted by a ^C while making a plot, PLOT3D_LCL may continue to run. This PLOT3D_LCL should be terminated before making another plot.
- o Other problems with communication between PLOT3D and PLOT3D_LCL may occur after interrupting PLOT3D.

CRAY_2

- o When compiling on the CRAY 2, ignore warning messages about

duplicate entries.

VMS

- o You may need to increase your page file quota to link and run PLOT3D. You need special privilege to do this:

```
$SD SYSS$SYSTEM:  
$RUN AUTHORIZE  
MODIFY username /PGFLQUOTA=40000
```

DISSPLA

- o Divide-by-zero errors are sometimes generated by DISSPLA while making 2D contour plots with labels. The traceback may point to QQTLB(?), CONTUR, or ENDCO2. This is a DISSPLA internal bug. Try changing MINMAX or plotting without contour labels ("PLOT/NOLABELS")
- o Spurious lines have been observed on QMS and LN03 output plots. The cause is unknown.

APOLLO

- o Titles and text have not been implemented.
- o PLOT3D is set up to use a maximum of 8 bitplanes of color.
- o The call structure of the C I/O routines has been modified in this version to conform to several de facto standards for calling C from FORTRAN. See file "fortcall.h" for the currently implemented conventions.

EXERPTS FROM COMMAND DESCRIPTIONS:

CONTOURS

(New Qualifier)

/LINEAR (D)

/CUBIC

For contour LINES, optionally use parametric cubic interpolation to find the intersections of contour lines with edges of cells. This smooths out "scaloping" caused by highly stretched, high aspect ratio cells. Takes longer to plot, as might be expected.

FUNCTION

(*New functions*)

Grid_functions

- 10 2D crossing grid line check.
- 11 Tetrahedron decomposition cell volume check.
- 12 Tetrahedron decomposition grid crossing check.

Scalar_functions

- 104 Log of normalized density
- 119 Log of normalized pressure.
- 124 Log of normalized temperature.
- 158 Divergence of velocity.
- 186 Helicity density.
- 187 Relative helicity.
- 188 Filtered relative helicity.
- 191 Filtered shock function based on pressure gradient.
- 192 Pressure gradient magnitude.
- 193 Density gradient magnitude.

Vector_functions

- 211 Density gradient.

Shock_functions

- 401 Filtered shock.

Function_definitions

Grid_checks

2D crossing grid line check:

Take each logical rectangle of grid points and check to see if two kitty-corner points are on opposite sides of the diagonal created by the other two points. (The two spatial dimensions used are determined from the VIEW command.)

Tetrahedron grid checking algorithms were originally coded by Dennis C. Jespersen, Computational Fluid Dynamics Branch, NASA Ames Research Center. Jespersen's comments follow:

Tetrahedron decomposition cell volume check:

Imagine each cell decomposed into six tetrahedra as in Fig. 3(b) of the reference. Compute the volume of each tetrahedron using the vector formula. If any of these volumes turn out to be "negative," there are probably problems with the grid. If all of these volumes are positive, the grid is probably OK. (I have no theorem about this.) The word "negative" is in quotes because of the nature of finite precision arithmetic. A tolerance is declared as a parameter.

Tetrahedron decomposition grid crossing check:

Imagine each cell decomposed into 5 tetrahedra as in Fig. 2 of the reference. For each planar face of the "core" tetrahedron, check whether each diagonally opposite pair of points is on opposite sides of the plane. With the notation of points as in the reference, we check

Plane	Points
138	4 6
168	3 5
136	2 8
368	1 7

Reference: W. Kordulla and M. Vinokur, "Efficient Computation of Volume in Flow Predictions," AIAA J 21 (1983), pp 917-918.

Helicity

Helicity density = (V dot omega)

Relative helicity = cos(phi), where phi is the angle between the velocity vector and the vorticity vector.

$$= \frac{(V \text{ dot } \omega)}{|V| |\omega|}$$

The relative helicity is set to zero when either |V| or |omega| is zero.

Filtering the relative helicity attempts to get rid of spurious maxima due to the "wandering" of the vorticity vector in regions where |omega| is basically small. NOTE THAT THIS FILTER IS DIMENSIONAL, AND IS BASED ON A LENGTH SCALE ON THE ORDER OF ONE.

```

Filtered relative helicity = {
    { cos(phi), abs(V dot omega) >= filter
    { 0          , abs(V dot omega) < filter

    filter = 0.1*Vinf**2

```

The inclusion of (and definitions for) the helicity functions are due to Mr. Yuval Levi, Aero Department, Technion Israel Institute of Technology, Technion City, Haifa, ISRAEL.

Divergence_of_velocity

The divergence of the velocity field is also known as the total dilatation or the normal-strain rate. As shown below, it gives some normalized measure of the local rate-of-change (particle derivative) of the density.

Where this function is negative, the flow is undergoing compression. By choosing suitable contours to accentuate these values, shock waves can be displayed!

Note that from the conservation of mass equation,

$$\frac{D(\rho)}{D(t)} + \rho \operatorname{div}(V) = 0,$$

we see that

$$\operatorname{div}(V) = -\frac{D(\rho)}{\rho D(t)},$$

which for steady flows is

$$= -\frac{1}{\rho} V \cdot \operatorname{del}(\rho).$$

This is how this function is actually computed in subroutine DIVV.

PLOT

(*New Qualifiers*)

/METAFILE

Output device.

DI-3000 ONLY devices: METAFILE

(whatever other device driver is loaded)

/OVERLAY=window#

Overlay this plot onto the indicated window. If no window number is supplied, overlay onto the most recently active window.

IMPLEMENTED FOR IRIS ONLY.

/CONTOUR (D)

/FSURFACE (*replaces* /SURFACE)

/CARPET

/LINE

If this is a plot of a scalar variable, set whether the plot will be a CONTOUR plot (the default) or a function surface (FSURFACE) plot. A CARPET plot is another common term for a function surface, and in 2D a function surface degenerates to a LINE plot; thus CARPET and LINE are synonyms for FSURFACE. See HELP FSURFACE for more information on function surface plots.

/TITLE (D)

/NOTITLE

Optionally turns off plotting of the top line of the title (the function name). Text specified using the TEXT command is still displayed.

/BAR (D)

/NOBAR

Optionally turns off plotting of the contour level color bar (list of contour levels).

/ADDITIONAL_TEXT (D)

/NOADDITIONAL_TEXT

Optionally turns off plotting of the block of additional text in the upper right of the plot, which includes grid size and other case information.

/SCRIPT

/NOSCRIPT (D)

Further input will be interpreted as **SCRIPT** commands for controlling the orientation of the display on the screen. **STOP** will signify the end of **SCRIPT** commands. See **PLOT SCRIPT** for information on these commands. **/NOSCRIPT** means that **SCRIPT** commands will not be expected. {**SCRIPT** is not implemented for **DISSPLA** or **GRAFIX**.}

RAS_screen_save

Type "R" to save the graphics window. (*was p in version 3.5*).

PostScript_screen_saves

Two types of PostScript screen saves are implemented. They use the "P" key (high-quality line/polygon image) and the "D" key (screen dump). Both convert the image to a gray-shaded image, and both can tie up a reasonably powerful printer for a looong time, based on the complexity of the image (and that doesn't necessarily mean the size of the file)!

The color of any object is converted to a "gray" value based on its difference from the background color, and the background color becomes equivalent to an "erase":

$$\text{gray} = 1 - \max(|R-R_{\text{background}}|, |G-G_{\text{background}}|, |B-B_{\text{background}}|)$$

where a value of zero is black and one is white. Thus to generate a shaded image, make the body to be shaded white, and plot with a white background (as it should appear on the page).

Once the image has been saved, the picture can again be manipulated with the mouse.

A new file is opened for every **PLOT3D** run. Thus if the file already exists, it is deleted!

High_quality

Type "P" to save the graphics window.

The following message will appear when the save is done:

"Graphics window saved as page N in file xxx.ps"

The filename used is the Q filename (or XYZ filename, or simply "PLOT3D") with a .PS extension.

This is intended for high-quality line drawings, or half-tone shaded images. Shading capability (probably) depends on the printer. Z-buffering DOES NOT work with this screen save, but backface polygon removal DOES.

Screen_dump

A bitmapped screen dump to a PostScript file is available as well. Type "D" to save the graphics window.

"D" stands for both "dump" and "dangerous"! A fullsize shaded image can tie up the printer for hours! Please be courteous to other users.

The following message will appear when the save is done:

"Graphics window saved as page N in file xxx.psd"

The filename used is the Q filename (or XYZ filename, or simply "PLOT3D") with a .PSD extension.

Appendix B. New Tools

By Pieter Buning

New tools were created for version 3.6 of PLOT3D. Some of these tools are used to run PLOT3D using less memory space.

```
prandtl/u/ps/buning/plot3d/tools> ls
```

Makefile	chopxb.f	int3d.f	rotateq.f	splitx.f
Makefile.c2	combineq.f	mirrorq.f	rotatex.f	splitxb.f
chopq.f	combinex.f	mirrorx.f	scalex.f	translatex.f
chopx.f	combinexb.f	refinex.f	splitq.f	(and others)

README

Various tools for PLOT3D grids and Q files. These can be used as-is or as examples to be edited.

CHOPQ

Cut a subset out of a (3D, single grid) PLOT3D Q file and write it out as a new Q file.

CHOPX

Cut a subset out of a (3D, single grid) PLOT3D XYZ file and write it out as a new XYZ file.

CHOPXB

Cut a subset out of a (3D, single grid) PLOT3D XYZ+IBLANK file and write it out as a new XYZ+IBLANK file.

COMBINEQ

Combine several (3D, single grid) PLOT3D Q files into a new multiple grid Q file.

COMBINEX

Combine several (3D, single grid) PLOT3D XYZ files into a new multiple grid XYZ file.

COMBINEXB

Combine several (3D, single grid) PLOT3D XYZ+IBLANK files into a new multiple grid XYZ+IBLANK file.

IJK

Generate a (3D, single grid) PLOT3D XYZ file which is simply the computational grid, i.e. $(x,y,z)=(i,j,k)$. Good for looking at flow quantities in the computational domain.

INT3D

Interpolate a (3D, single grid) PLOT3D Q file from one grid onto another. Old and new XYZ files may have IBLANK. Various options available on what to do if a new grid point isn't found within the old grid. Uses trilinear interpolation.

MIRRORQ

Flip a (3D, single grid) PLOT3D Q file about the x-, y-, or z-axis.

MIRRORX

Flip a (3D, single grid) PLOT3D XYZ file about the x-, y-, or z-axis.

PROPER2D

Perform 2D grid line crossing check on a (2D, single grid) PLOT3D XYZ file.

PROPER3D

Perform tetrahedron decomposition cell volume check on a (3D, single grid) PLOT3D XYZ file.

PROPER3DN

Perform tetrahedron decomposition grid crossing check on a (3D, single grid) PLOT3D XYZ file.

REFINEX

Generate a new (3D, single grid) PLOT3D XYZ file which is an integer refinement of an existing grid file. Uses parametric cubic interpolation.

ROTATEX

Rotate a (3D, single grid) PLOT3D XYZ file about the x-, y-, or z-axis.

SCALEX

Scale a (3D, single grid) PLOT3D XYZ file.

SCALEX

Scale a (3D, single grid) PLOT3D XYZ+IBLANK file.

SPLITQ

Split a (3D) multiple grid PLOT3D Q file into separate single grid Q files.
Can skip grids if desired.

SPLITX

Split a (3D) multiple grid PLOT3D XYZ file into separate single grid XYZ files. Can skip grids if desired.

SPLITXB

Split a (3D) multiple grid PLOT3D XYZ+IBLANK file into separate single grid XYZ+IBLANK files. Can skip grids if desired.

TRANSLATEX

Translate a (3D, single grid) PLOT3D XYZ file.

TRANSLATEXB

Translate a (3D, single grid) PLOT3D XYZ+IBLANK file.

Use "make" to compile and link tools as follows:

2xxx/3xxx: make -f Makefile.i2

CRAY 2: make -f Makefile.c2

VAX/VMS: @MAKEFILE.VMS

(There are more make files).

Appendix C. PLOT3D Without An IRIS...plot3ddi

By Jacques Middlecoff

A DI-3000 version of PLOT3D is now operational on Navier and Reynolds. It is called plot3ddi and is located in /usr/local/lib/pvi/bin/. The advantage of using plot3ddi is that it can send plots to any device supported by a DI-3000 driver (such as a graphics terminal), and is not limited to plotting on an IRIS workstation. The disadvantage is that on most devices plot3ddi does not support the full range of graphic operations available on an IRIS workstation, such as high-speed, mouse-controlled graphics manipulation.

The plot3ddi program is executed by entering the command:

```
plot3ddi [<term>] [<prntr>] [<meta>] [<megawords>]
```

Plot3ddi allows the user to plot on a terminal of type <term> and/or make a file for printer type <prntr> and/or make a metafile depending on <meta> and allocate <megawords> of memory. If the <megawords> argument is omitted, plot3ddi defaults to 1 megaword (1,000,000 words) of memory. (Some jobs may require 10 megawords or more.) Typing plot3ddi with no arguments produces a list of the available drivers. Currently supported drivers on the Cray are:

<term>	device
4014	Tektronix 4014 (Mono)
4105	Tektronix 4105 (8 Colors)
4115	Tektronix 4115 (255 Colors)
G407	GraphOn 407HR(256 Colors)

<prntr>	device
imgn	Imagen printer
post	PostScript printer
LNØ3	LNØ3 plus printer

<meta>	device
meta	metafile

Dynamically allocating memory at runtime provides two major benefits: first, using a minimum amount of memory will significantly increase interactive response time; and second, plots can be made from jobs with memory requirements too large for interactive execution.

You can set the memory allocation by using the command

```
Plot3ddi 4105 1.5
```

which will allocate 1,500,000 words of memory and plot to a Tektronix 4105 terminal. It is important to keep memory allocation to a minimum to increase response time. In this case, you will get much faster interactive response time than you would by allocating the maximum interactive memory limit of 20 megawords.

A job requiring more than 20 megawords of memory can be submitted through NQS, producing a metafile which can be viewed interactively later. A program called p3dtrans is provided on Navier and Reynolds to allow the user to view a metafile that has been created by plot3ddi.

The program p3dtrans is executed by entering the command:

```
p3dtrans [<term>] [<prntr>]
```

Where <term> is the destination terminal and [<prntr>] are as defined above.

Plot3ddi and p3dtrans are documented in the man pages on Navier and Reynolds. If you have any questions or find any bugs, please contact Jacques Middlecoff through NAS User Services at (415) 694-4444 or via electronic mail addressed to : middleco@prandtl.nas.nasa.gov.

Appendix D. Incompressible Fluids

By Domingo Tavella and Stuart Rogers

1. General case

Assuming the density is constant everywhere, define the Q-vectors as follows:

$$Q1 = 1$$

$$Q2 = u$$

$$Q3 = v$$

$$Q4 = w$$

$$Q5 = \frac{p}{\gamma-1} + \frac{Q1}{2} (Q2^{**2} + Q3^{**2} + Q4^{**2})$$

where u, v, and w are the x, y, and z components of the velocity, respectively, and p is pressure. Since PLOT3D assumes $\gamma = 1.4$, Q5 is written as follows:

$$Q5 = \frac{p}{0.4} + 0.5 (Q2^{**2} + Q3^{**2} + Q4^{**2})$$

2. Irrotational Flows

Assuming that the pressure at infinity is zero, and that the modulus of the velocity at infinity is one, define

$$Q5 = \frac{1}{2(\gamma-1)} [1 + (2-\gamma) (Q2^{**2} + Q3^{**2} + Q4^{**2})]$$

For $\gamma = 1.4$, Q5 is written as:

$$Q5 = \frac{1}{0.8} [1 + 0.6 (Q2^{**2} + Q3^{**2} + Q4^{**2})]$$

Appendix E. IRIS Differences

By Larry Pierce

Here are some notes on differences between IRIS 2xxx/3xxx's and IRIS 4D's:

1. The IRIS 2xxx/3xxx can run under the window manager (mex) or without it. Under the window manager it is possible to get several plots (up to 8) on the screen at the same time. To remove a plot, press <delete> on the graphics keyboard. Pressing <return> gets you back to the PLOT3D prompt, but does not close the window.

vs.

The IRIS 4D's are ALWAYS running under the window manager (Max/4sight). It is possible to get up to 8 plots on the screen at the same time. To remove a plot, press <delete> on the graphics keyboard. Pressing <return> gets you back to the PLOT3D prompt, but does not close the window.

2. IRIS 2XXX/3XXX: to attach to a window (while running mex), you must move the cursor into that window and then depress (don't release) the right mouse button; a mex menu will pop up, and then the user should move the cursor over the menu item "attach" or the menu item "select" (select is equivalent to an "attach" followed by a "pop"); then release the right mouse button.

vs.

IRIS 4D's: to attach to a window, you must move the cursor into that window. That's it. Whenever you move the cursor to a different window, that window is automatically attached.

3. There is a strange bug with the windowing in PLOT3D on the 4D's: if you are in plot-mode (i.e. the mode in which you can use the mouse) and you are using the mouse (particularly x/y translation) AND there is a window overlapping (i.e. partially obscuring) your current graphics window, the mouse may go out of control. No hint of a solution yet. For now we recommend that the user "pop" any window that he wants to mouse in.

4. IRIS 2xxx/3xxx: It is possible to create .bin (pure binary) data files for PLOT3D by writing a FORTRAN program and using FORM='BINARY' in the OPEN statement. Then one can write the data and close the file in the same manner as for a FORTRAN UNFORMATTED file.

IRIS 4D's: It is possible to create .bin data files for PLOT3D by writing a FORTRAN program and using FORM='BINARY' in the OPEN statement... but the resulting file is NOT pure binary, and therefore PLOT3D won't READ it with the /BINARY (D) option! It may be possible to READ such files with the /UNFORMATTED option, but we aren't sure about that. For now we recommend that FORTRAN programs on the IRIS 4D's use FORM='UNFORMATTED' in the OPEN statement; then they can READ the files using the /UNFORMATTED option. NOTE: if the PLOT3D user is a C programmer, he can write C code to create genuine .bin (pure binary) files for PLOT3D. Such users are probably a minority.

5. IRIS 2xxx/3xxx: The PLOT3D program is compiled/linked using the command `make -f Makefile.i2` on the Unix command line.

IRIS 4D/70, or 4D/70G: The PLOT3D program is compiled/linked using the command `make -f Makefile.i4` on the Unix command line. This command will probably work for IRIS 4D/20, IRIS 4D/20G, IRIS 4D/50, IRIS 4D/50G, etc.

IRIS 4D/70GT: The PLOT3D program is compiled/linked using the command `make -f Makefile.i4gt` on the Unix command line. This command will probably work for IRIS 4D/20GT, IRIS 4D/50GT, etc.

PLOT3D_LCL (IRIS process for distributed version) is set up similarly.

6. Hardcopy capabilities: the IRIS 2xxx/3xxx version is probably the most robust in this sense. In particular the PostScript High-Quality ("P") works on the IRIS 2xxx/3xxx, MAY work on IRIS 4D/70 or IRIS 4D/70G (with some bugs), and will NOT work on IRIS 4D/70GT.
7. Some bugs (particularly for the IRIS versions) with PLOT3D version 3.6 are documented in the file "bugs.txt" (in the documentation directories).

Appendix F. Directory Structures and Wingbodytail

By Larry Pierce

F.1 Finding Wingbodytail

IRIS (stand-alone) implementation:

Users who receive IRIS PLOT3D source code from us will find Version 3.6 examples in "plot3d/3.6b/unix+iris2/examples/". NAS workstation users who want to access the files on wk23 can simply use "/r/wk23/wk/pierce/examples" to find the various IRIS PLOT3D examples (with nfsmount). Version 3.5 examples are not available for IRIS.

VAX/VMS (DISSPLA or DI3000) implementation:

Users who receive the PLOT3D source code for one or both of the VMS implementations will find the examples in "[...PLOT3D_3_6.VMS.EXAMPLES]". Users on NASA/Ames VAX/VMS systems will find the examples (for version 3.6 only) in EUL::STT0:[BUNING.EXAMPLES]. The 3.5 examples were deleted.

Sun implementation:

Users who receive the PLOT3D source code for Sun will find the examples in "plot3d/3.4/suncore/plot3demos/". Wingbodytail is not available for this one. We don't have the Sun examples in a standard place for NAS (or NASA) users.

Stellar implementation:

Users who receive the PLOT3D source code for Stellar will find the examples in "plot3d/3.6_Stellar/examples/". There aren't many users (that I know of) here at Ames, so we won't be putting the examples in a standard place for our local users.

implementation independent (ASCII) data:

Users who receive PLOT3D source code with the FORMATTED (ASCII) examples will find them in "plot3d/3.6/examples_fmt/" (Unix syntax) or "[...PLOT3D_3_6.EXAMPLES_FMT]" (VAX/VMS syntax). NAS PLOT3D users who want access to these examples will find them on prandtl (alias pam) in "/u/pi/pierce/examples_fmt/".

F.2 Directory Structures December 89

```

plot3d
  3.4
    suncore
      plot3demos
        airfoil
        bluntfin
  3.6
    documentation
    examples_fmt           (machine independent)
    airfoil
    bluntfin
    wingbodytail
    tools
    tutorial
    vms
      displa
      di3000
      examples             (VMS examples)
        airfoil
        bluntfin
        wingbodytail
    unix+gmr3d
    unix+di3000
  3.6b
    documentation         (3.6b has augmented documentation)
    unix+iris2
      arcgraph
      mrake
    examples             (IRIS examples)
      airfoil

```

```

        bluntfin
        wingbodytail
    unix+iris2_rgl
        local
        remote
        cgl_dist      (CGL distributed graphics library)
3.6_Stellar        (Stellar's stuff does not
                   match our 3.6 stuff exactly, so
                   gets its own master directory)

    documentation
    examples        (Stellar examples)
        airfoil
        bluntfin
        wingbodytail
    tools
    tutorial
    unix+stellar

```

turb3d

```

    3.6
        documentation
        vms
            displa
            di3000
        unix+gmr3d
        unix+di3000
    3.6b
        documentation      (3.6b has augmented documentation)
        unix+iris2
        unix+iris2_rgl
        remote

```

Also, on VMS systems some of the directories will be concatenated vertically, e.g.

UNIX	VMS
plot3d/3.6_Stellar/	-> [...plot3d_3_6_Stellar]
plot3d/3.6/	-> [...plot3d_3_6]
plot3d/3.6b/	-> [...plot3d_3_6b]
plot3d/3.4/	-> [...plot3d_3_4]
turb3d/3.6/	-> [...turb3d_3_6]
turb3d/3.6b/	-> [...turb3d_3_6b]

Glossary

By Patricia Elson

.bin file	binary file
.com file	command file for PLOT3D keyboard entry
.dat file	unformatted FORTRAN file
.dip file	NASA Ames device-independent plotter (dip) file
.fmt file	formatted FORTRAN file
.gra file	graphics ARCGraph file for a mixture of data types
.jou file	journal file created by PLOT3D recording all keyboard work during a session; if an abnormal end occurs, PLOT3D saves the file; the file may also be saved by EXIT/SAVE or QUIT/SAVE
.par file	particle trace file storing x, y, z, time and real i, j, k of each point of the trace
.pol file	polygon ARCGraph file for flat shaded parts; subset of .gra
.prs file	ARCGraph file for function-mapped or contour function data; subset of .gra
.ps file	PostScript file (available for IRIS 2xxx/3xxx and some IRIS 4Ds in version 3.6)
.qms file	QMS format file to store a screen save for a QMS printer
.ras file	raster file created by a screen save using ARCGraph RAS format
.sei file	Seiko color printer file
.vec file	vector ARCGraph file for lines
@ <i>filename</i>	reads <i>filename</i> for PLOT3D commands; on finding an end-of-file character, control is returned to the previous source of input
alpha	angle of attack; this value is contained in the solution file

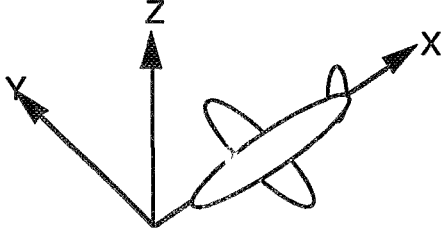
ARCGraph	Ames Research Center Graphics: device-independent, standard function libraries and utilities developed by the Advanced Computer Research Division at NASA Ames Research Center
attribute	choices to be made after the WALLS, CONTOURS, VECTOR, SUBSET or RAKE commands to determine color, shading, line or solid, length of vector, etc.
bit plane	area of memory to store pixel information: a bit plane supplies one bit of color information per pixel on the display; thus, an 8 bit plane system allows 2^8 (256) different colors to be displayed
Boolean	value of true or false: true equals 1, false equals 0
carpet plot	type of plot called with the PLOT/SURFACE command showing a graph of a scalar function (e.g. x versus density or xy versus pressure, etc.); use PLOT/FSURFACE in version 3.6
CFD	computational fluid dynamics: the discipline that attempts to solve the differential equations of fluid flow numerically
clip	omit drawing part of an object outside the clipping planes
clipping planes	sides of a rectangular 3D space where objects are drawn; objects outside this space are not drawn at all; objects that extend partially outside this box are "clipped" so only the portion inside is drawn; see MINMAX box
coefficient, material	polygon reflectivity entered for each wall (currently prompted, but not implemented)
color	one of the eight standard colors (entered by name): black, blue, cyan, green, yellow, red, magenta, white; color may also be (Red Green Blue) intensity components in the range from 0 to 1 specified, e.g., "RGB .2 .3 .4"
command file	file of commands for entry into PLOT3D; see .com file

contour plot	plot showing color-coded values of a function throughout the subset either on planes slicing through the scene or on the surface of an object; used for scalar functions such as pressure, Mach number
CPU	central processing unit; the part of a computer where instructions are executed
Cray2	model of supercomputer
CRT	cathode ray tube; a raster display monitor
data multiplier	real number factor that will be multiplied with every x, y, z coordinate
delta	Greek letter; Δ and \triangle are used as symbol types
density	concentration of matter
DIP	NASA Ames device-independent plotter file convention
display mode	setting for a display screen which can be 60 Hz, 30 Hz, or NTSC
double buffer	mode in which two buffers are alternately displayed and updated
dynamic display	display with options to move, rotate and zoom in and out of the display
enthalpy	heat content
entropy	capacity factor for isothermally unavailable energy
Ethernet	type of high-speed communication link
fsurface	(1) in v3.6 a <i>qualifier</i> to the plot command for creating a function surface plot, also called a carpet or line plot; (in v3.5 use /SURFACE); (2) a PLOT3D <i>command</i> for making adjustment in a function surface plot
flow field	fluid flow described by the calculations in the solution file: these may be obtained either from experimental measurements or computer simulation
FSMACH	free-stream Mach number; the speed of the air beyond the disturbance of the vehicle; this value is in the solution file

GAS	Graphics Animation System: a menu driven program for viewing, animating, and recording computed results created by PLOT3D (or other graphics package)
geometry	visible parts of a grid: in PLOT3D, the user defines the visible part of the geometry with the WALL command
Geometry Engine	IRIS workstation hardware consisting of 12 VLSI chips which perform the floating point calculations needed to transform, project, and clip geometrical data for display on a CRT
Gouraud shading	an advanced method of shading in which each polygon is smoothly color shaded; available in SURF, not PLOT3D
grid	array of x, y, z points at which the flow is defined
grid file	file that supplies PLOT3D the grid
helicity	a measure of vortex strength
hidden line	line of the wireframe of a geometric object that is not visible because it is obscured by "surfaces" of the wireframe object
hidden surface	surface of a geometric object that is not visible because it is obscured by other surfaces
HYPERchannel	local network that allows workstations to interface with the Cray2
i, j, k	vectors indicating units of the grid: grid line segments; usually i is one unit in the x direction; j is one unit counterclockwise around the object; k is one unit radially out from the surface of the object
IBLANK data	data following the Z values in the grid (XYZ) file showing which grid points are "turned off" or blanked out; the iblank data defines holes in a grid made to accommodate another grid; also defines the location of solid wall or where grids meet or overlap, for particle tracing
IJ, IK, or JK	plane orientations; determines which computational planes will be used for plotting contour lines, for example

initialization file	optional command file to start PLOT3D: when PLOT3D starts it looks for a file "plot3dini.com" in the current directory; PLOT3D reads commands in that file before taking input from the normal input device (the terminal)
Jacobian	the determinant of the coordinate transformation matrix
journal file	the file PLOT3D.jou is created when PLOT3D ends abnormally or after an EXIT/SAVE command; the .jou file is usually renamed <i>filename.com</i> and used as a command file
legend	color-mapped list displaying the range of the function and value represented by different colors
LHPS	Lines/Hidden_lines/ Points/Surface: an abbreviation used on the PLOT3D grid chart
line thickness	real number which multiplies the standard line thickness
line type	style of line such as SOLID, DASH, DOT, CHAINDASH, CHAINDOT or NONE; the default is SOLID; PLOT3X does not allow CHAINDASH, CHAINDOT, or DOT line types
ln tp	line type: an abbreviation used on the PLOT3D grid chart
Mach	velocity in relation to the speed of sound
material coefficient	polygon reflectivity (not yet implemented)
mgrid	qualifier used with the READ command when working with multiple grids
MINMAX box	user-defined 2D or 3D space that specifies plot ranges and defines a center of rotation
mirroring	creation of a mirror image of an object
momentum	quantity of motion measured by the product of mass and velocity
MRAKEVEC	program that converts .par files into .vec files that can be read by GAS (Graphic Animation System)
nabla	inverted triangle; ∇ and \blacktriangledown are used as symbol types
NAS	Numerical Aerodynamic Simulation
nocheck	option of the READ command to omit testing for inconsistent data to make READ command run faster

NTSC	National Television Standards Code: the signal used by television broadcast and standard video recorders and players in the United States
particle trace	path a particle will travel past an object
perspective	viewing projection that shows distant objects as smaller
pixel	picture element: a display screen is composed of an array of pixels; in a black and white system, pixels are turned on and off to form images; in a color system, each pixel has three color components: red, green and blue; see RGB
pixel dump	file containing a saved screen image
PLANES format	three-dimensional flow field or grid data format where each K-plane is a new file record (available in version 3.5 on)
PLOT3D	interactive graphics program for CFD flow fields illustrating 74 functions such as particle tracing, pressure contours, Mach number, and shock finding: PLOT3D makes extensive use of color and 3D cues such as perspective; it is written primarily in FORTRAN and runs on IRIS workstations and several other systems
PLOT3X	modified version of PLOT3D that produces .gra graphic output files that can be assembled into a movie using GAS; PLOT3X does not display plots
PLOT4D	modified version of PLOT3D that allows some animation capability
polar coordinates	coordinate system in the plane related to the the standard Cartesian (x, y) coordinates by the following equations: $x = r \cos \theta$, $y = r \sin \theta$ (r and θ are polar coordinates)
polygon shininess	real number between 0 (dull) and 1 (shiny) controlling shaded surface shininess; available in version 3.6
polygon transparency	value between 0 (opaque) and 1 (completely transparent); available on the IRIS version of PLOT3D only
pressure	force applied to or distributed over a surface

pressure map	method of representing a range of values by using color shades; applies to scalar functions in general
Q file	see solution file
rake	set of computed particle trace paths
raster display	screen display that produces an image using a series of scan lines
RE or Re	Reynolds number, magnitude of viscosity; in the solution file
RGB	Red Green Blue: an additive color system that generates colors by varying the intensities of red, green, and blue
RGB values	for PLOT3D, a set of real number from 0 to 1 for the (red green blue) intensities that compose a color, (e.g., RGB .5 .5 .5)
RGL	Remote Graphics Library: a Cray2 library that sends graphic commands to an IRIS workstation
right-handed system	if the fingers of the right hand curl in the direction of rotation from positive x to positive y, the thumb points to positive z; in CFD grids, x is often downstream, y out the starboard wing, and z up
	
	<i>Figure 84. Right-handed coordinate system.</i>
RIP	Remote Interactive Particle-tracer: a menu-driven program to interactively calculate particle traces and plot them on an IRIS or output them to data files (.par) for input into the MRAKEVEC program; RIP is written in C and executes on IRIS and Cray2 simultaneously (i.e. in a distributed fashion)
rotate	revolve about a given axis or origin
scalar	quantity such as density, pressure, or temperature that can be specified by a single number

scale	change the overall size of an object: an object may be scaled to retain its relative dimensions or distorted
scaling	uniform stretching of an object along an axis
screen save	procedure to save the image on the screen
script file	a file used by PLOT/SCRIPT to "animate" in PLOT3D
shading pattern	shading pattern for polygon fills: the default is SOLID
shininess	real number between 0 (dull) and 1 (shiny) controlling shaded surface shininess; available in version 3.6
solution file	file containing the flow-field to be plotted by PLOT3D; this file contains the conditions under which the solution was generated (FSMACH, ALPHA, RE, TIME) and flow density normalized to free-stream, x-flow momentum, y-flow momentum, z-flow momentum, and stagnation energy per unit volume; also called Q file
spectrum plot	plot showing color-coded levels (for pressure, Mach number, etc.) with no line separation between colors.
subset	part of a grid selected to be "active"; the portion of a grid to which the function applies
SURF	SURFace shading program: an interactive graphics program that creates wireframe, flat shaded, Gouraud shaded, and smoothly colored function-mapped 3D models which can be loaded into GAS; SURF is written in C and runs on IRIS workstations
surface	(1) the surface of a model; (2) as an attribute, filled polygons instead of lines; (3) as a qualifier to the plot command, carpet plot (replaced by /FSURFACE in version 3.6)
symbol size	real number which multiplies the standard symbol size
symbol type	specific symbol represented by an integer (-1 to 12)
T1 channel	channel with a speed of 1.5 MBits/sec
text	up to two lines of centered text created by TEXT command
TIME	solution time; this value is contained in the solution file and is used only for additional text on plots

title	name of current function printed at top of plot; turned off with PLOT/NOTITLE
trace segment	part of a trace that lies within a particular grid; with multiple grids the trace can extend though several grids
transformation	any matrix multiplication that changes the orientation of an object; see rotate, scale, translate, zoom
translate	to move a given distance along the x, y, or z axis
transparency	real number between 0 (opaque) and 1 (completely transparent) controlling polygon transparency
unsteady flow	time-dependent flow
vector field	field property specified by more than one number, usually magnitude and direction
velocity	time rate of motion specified as a magnitude and direction
viewpoint	the point in space from which the plotted object is viewed
visualization	application of graphics and imaging techniques to the understanding of physical phenomenon; scientific data visualization
vortex	three-dimensional circular flow field caused by the motion of a rigid body through a fluid
vorticity	amount of rotation about a given center; whirling
wall	visible part of the grid; walls define the visible part of the geometry in PLOT3D
WHOLE format	three-dimensional flow-field data format where the entire grid or flow-field is one large file record
wireframe	model of an object that shows only the edges of the polygons that make up the surface of the object
workstation	computer with high-performance graphics and large disk storage capacity
x, y, z	Cartesian coordinates referring to physical points in the grid
XYZ file	See grid file

PLOT3D User's Manual

z-buffering	method of hidden surface (polygon) removal that displays points of the surface based on distance from the viewpoint; a hardware feature of some computers
zoom	to move the viewpoint closer to focus in on details or move farther away to get the total picture

Index

- (D) indicates default
- / indicates qualifier
- . dot indicates filename extension

- " 115
- ! 115
- \$ 120
- 115
- .bin binary data 22, 233
- .com command file 22, 86—88
- .D48 files 95
- .dat FORTRAN unformatted data 22, 233
- .dip Device Independent Plotter file 22, 94, 95, 140—141, 233
- .fmt formatted data 22, 233
- .gra general ARCGraph file (use PLOT3X to create) 22, 89, 233
- .jou journal file 28, 85—86, 102, 233
- .par particle trace file 22, 88, 233
- .pol ARCGraph file for flat shaded polygons 22, 233
- .prs ARCGraph file for pressure or other function 22, 233
- .ps files 93, 95, 233
- .psd files 95
- .qms screen save (QMS QUICK format) 22, 93, 95, 233
- .qqq 233 (see .bin)
- .ras pixel file screen save 22, 92, 95, 233
- .sei screen save for Seiko color printer 22, 92, 95, 233
- .vec ARCGraph file for lines 22, 233

- /
- /+TIME/+TIME (D)/-TIME 150
- /1D 153
- /2D 140, 153
- /3D (D) 140, 153

- /ADD 150, 156, 161
- /ANGLES 42, 160
- /ANGLES in SCRIPT 148
- /APOLLO (D for Apollo) 140
- /APOLLOSINGLE 140
- /ATTRIBUTES 122
- /ATTRIBUTES in RAKES 150
- /ATTRIBUTES in SUBSETS 156
- /ATTRIBUTES in VECTORS (D) 158
- /ATTRIBUTES in WALLS (D) 161
- /AUTOMATIC 122
- /AXES (D) 142
- /BACKGROUND=color 142
- /BINARY (D for Unix) 153
- /BLANK 154
- /CARPET 141
- /CHECK (D) 154
- /COMPRESS 140
- /CONTOUR (D) 61, 125, 141
- /DICOMED 140
- /DIP 22, 94, 95, 140—141, 233
- /ENVISION 140
- /FIGURE=(areax,areay,charht) 142
- /FORMATTED
 - with LIST 137
 - with READ 153
- /FROM=(x, y, z) in SCRIPT 147, 148
- /FULLSCREEN 142
- /FUNCTION=file in READ statement 153
- /GRAFIX (D for GRAFIX) 140—141
- /GRID=grid number 32, 125, 156, 161
- /IJK (D) 150
- /IMAGEN 140
- /IN=n in SCRIPT 147, 148
- /INCREMENT 122, 139
- /IRIS (D for IRIS) 140
- /IRISRGB 140
- /IRISSINGLE 140
- /JACOBIAN 154
- /LINE 141
- /LN03 140
- /MANUAL 122
- /MAXPOINTS=n 151

- /MDATASET 153
 - /MGRID 153
 - /NOATTRIBUTES (D) 122, 150, 156, 158, 161
 - /NOAXES 142
 - /NOBLANK (D) 154
 - /NOCHECK 154
 - /NOFIGURE (D) 142
 - /NOFULLSCREEN (D) 142
 - /NOJACOBIAN (D) 154
 - /NOSCALAR_FUNCTION (D) 151, 158
 - /NOX 139
 - /NOY 139
 - /NOZ 139
 - /OUTPUT=file 137
 - /PLANES 154
 - /Q=file (D) 153
 - /QMS 140
 - /RANGE 122
 - /READ=file 150
 - /SAVE 123
 - /SCALAR_FUNCTION
 - with RAKE 151
 - with VECTORS 158
 - /SCALE_FACTOR 124
 - /SCRIPT 143, 147
 - /SURFACE 61, 141
 - /TEKTRONIX (D for DISSPLA) 140
 - /TEXT (D) 137
 - /TK41XX 140
 - /UNFORMATTED
 - with LIST 137
 - with READ 153
 - /UP=axis 142
 - /VERSATEC 140
 - /WALLS_ORIGIN 125
 - /WHOLE (D) 154
 - /WRITE=file 150
 - /X (D) 139
 - /XSCALE=scale 139
 - /XYZ
 - in RAKE 150
 - in READ 153
 - in VPOINT 160
 - /Y (D) 139
 - /YSCALE=scale 139
 - /Z (D) 139
 - /ZSCALE=scale 139
 - 2D_stream_function 133
 - 16-mm film 89
 - @ symbol 10, 115, 119
 - [square brackets] 101
 - ^ 101
 - 1D 153
 - 2D 140, 153
 - 2d Pressure Coefficient using TWING
 - (plot/line) 220
 - 3D (D) 140, 153
- A**
- abbreviations 22
 - ADD 150, 156, 161
 - ANGLES 160
 - ANGLES in SCRIPT 148
 - animation 88, 89, 94
 - APOLLO (D for Apollo) 140
 - APOLLO ONLY devices 141
 - APOLLOSINGLE 140
 - ARCGraph 22, 23, 89, 146
 - Arrowheads 158
 - ATTRIBUTES 33, 62, 116
 - in CONTOURS (D) 122
 - in RAKES (D) 150
 - in SUBSETS 156
 - in VECTORS (D) 158
 - in WALLS (D) 161
 - AUTOMATIC 122
 - AXES (D) 142
- B**
- Backfacing_polygon_removal 146
 - BACKGROUND=color 142
 - basic information needed for making a
 - plot 101
 - BINARY (D for Unix) 137, 153
 - Binary files for the IRIS — .bin
 - extension. 162
 - BLANK (blanking) 21, 29, 115, 154
 - BLOCK DATA BFLUID. 129
 - bugs ("Known Problems") 111
 - Buning, Pieter 102
 - button 144
- C**
- c 129, 132
 - CARPET (carpet plot) 61, 71, 141

- case-sensitive (not) 22
 - CENTER in SCRIPT 147
 - CGL 227—229, and :
 - error messages 177, 181, 185, 192, 193, 211
 - Changing PLOT3D and Host Systems
 - 168 — 176
 - To Use Less Space 172
 - CHECK (D) 154
 - Chewbaka (wkØØ) 93
 - Chimera 12
 - CLEAR 121
 - clipping planes 41, 147
 - code
 - organization 169 — 171
 - requests 3
 - Color, vector 158
 - color contours 62—63, 67
 - colors 33, 117
 - Command Basics 20
 - command files 9, 14—15, 86
 - sample command files 219—231
 - command/qualifier arguments 20
 - commands
 - basics 20
 - explanations 121—161
 - list of 98
 - typical sequence 234
 - comparison of computational and experimental data 221—226
 - COMPRESS 94, 95, 141
 - Continuation character 22
 - CONTOUR (D) qualifier to PLOT 125, 141
 - compared to surface 61
 - contour vs. surface chart 63
 - overlay two kinds 64
 - CONTOURS command 63, 66, 122
 - attributes 66, 116
 - levels 66
 - control-C 19
 - Converting .com files to .gra for GAS 89
 - converting .par files to .vec files for GAS 88
 - COS JCL 141
 - Cp 130
 - Cp* 130
 - CpØ 130
 - crash 19, 85
 - CRAY
 - error messages 177, 181, 185, 192, 193, 211
 - output files 141
 - running PLOT3D on 227—229
 - Cross-flow_velocity (function #127) 133
 - current working directory 28
- D**
- d 93
 - D48 film recorder 141
 - data File Formats 162—167
 - data In Current Directory 28
 - data, differnet form of 172
 - data files 1, 7
 - default contour colors 104
 - density (or Q1) (function #100) 126, 129
 - density gradient (function # 211) 243
 - depth cueing 14
 - Device Independent Plot — .dip Files 22, 94, 95, 140—141, 233
 - device to plot to 94—95
 - device-independent Ames Research Center Graphics library 89
 - DICOMED 94—95, 140—141
 - Different Form of Input Data 172
 - Different Graphics Library 176
 - Different Operating System 176
 - digital disk 89
 - DIP 22, 94, 95, 140—141, 233
 - DISSPLA output files 94, 140—141
 - divergence of velocity (function # 158) 243, 246
 - drag 13—14, 144
 - Dunn camera 93—94
 - Dynamic_pressure 126, 131
- E**
- EARTH (wingbodytail dir) 8, 258, 259
 - ei 132
 - End-of-file 120
 - Energy 127, 132
 - Enthalpy 126, 132
 - Entropy 127, formula 133
 - Entropy measure s1 127, 133

ENVISION 140
 EOF 120
 equals sign (=) 21
 Error Messages and Warnings 177 —
 218
 exit 26, 85, 123, ^C to stop plot 53
 EXIT/SAVE 85—86, 89, 123
 experimental data 221—226
 extensions 22, 233

F

f (mex screen dump) 93, 95
 FIGURE=(areax,areay,charht) 142
 File types and extensions 22, 91, 95,
 summary chart 233
 .bin binary data 22, 233
 .com command file 22, 86
 .D48 files 95
 .dat FORTRAN unformatted data
 22, 233
 .dip Device Independent Plotter file
 22, 94, 95, 140—141, 233
 .fmt formatted data 22, 233
 .gra general ARCGraph file (use
 PLOT3X to create) 22, 89, 233
 .jou journal file 27, 85—86, 102,
 233
 .par particle trace file 22, 78, 88,
 233
 .pol ARCGraph file for flat shaded
 polygons 22
 .prs ARCGraph file for pressure or
 other function 22, 233
 .ps 93, 95, 233
 .psd 95
 .qms screen save (QMS QUIC
 format) 22, 93, 95, 233
 .ras pixel file screen save 22, 92,
 95, 233
 .sei screen save for Seiko color
 printer 22, 92, 95, 233
 .vec ARCGraph file for lines 22,
 88, 233
 File_format for particle trace files
 151—152
 filenames generated by PLOT qualifiers
 141
 film recorder (DISSPLA) 95
 films 89

flow-solvers 2—3
 Fluid Dynamics Division 1
 fluid is assumed to be air 129
 Fluid_constants 129
 fooling PLOT3D 172
 formats, data file 153, 162—167,
 using LIST 137
 FORTRAN formatted files — .fmt
 extension. 153, 162
 FORTRAN unformatted files — .dat
 extension. 153, 162
 freestream density, pressure, speed of
 sound, velocity magn. 129
 FROM=(x, y, z)in SCRIPT 147, 148
 FSURFACE command 73, 124—125
 FSURFACE qualifier, version 3.6
 61—62, 71—74, 236, 247
 FULLSCREEN 142
 FUNCTION
 command 66, 126—135
 definitions 129—135
 definitions, changing code 173—
 175
 numbers 126—128
 tutorial 49—84
 Function Files formats 166
 function surface 124, compared to
 contour 61—62, tutorial 71—74
 FUNCTION=file in READ statement
 153
 Function_file multiple_grid formats
 167
 Function_file single_grid formats 166

G

gamma 129
 GAS program for animation 2, 88
 for adding titles 92
 gas constant 129
 General_information from help file 101
 Geometry 12, 32—40, 50—57
 GO in SCRIPT 148
 GRAFIX ONLY devices 140, 141
 graph of the function 71—74
 Graphics Library, Different 176
 graphics file 89
 graphics lab 93
 graphics ports, multiple 143
 Grid Chart 43—45

grid checks (functions # 10, 11, 12)
243—245
Grid XYZ Files 1—2
formats 162—164
READING (loading) 27—30
GRID=grid number qualifier
with WALLS 32, 125, 161
with SUBSETS 53—54, 156
Grid_functions
help file 126
new, version 3.6 243—245
tutorial 50—57

H

h 132
Hard Copies 92—95
helicity (functions # 186, 187, 188)
definition 127
formulaes 245
new functions 243
HELP command 136
help file, on-line
how to use 23—26, 136
reproduced 101—167
hidden_lines 33, 116
fixed, version 3.6 238
hogwhiledown 144
HOLD in SCRIPT 148
Holes, in grid (IBLANK) 50, 55—57,
115—116
Host Systems 5, 168
hyphen (-) 22

I

IBLANK 50, 55—57, 115—116
I, J, and K 2, 13
IJ, IK, or JK planes
with CONTOURS 66
with RAKE 151
IJK (D) qualifier for RAKE 150
IMAGEN 94, 95, 140, 141
imagin.imp files 95
IN=n in SCRIPT 147, 148
incompressible flow 255
INCREMENT qualifier
with CONTOURS 122
with MINMAX 139

Initialization File — plot3dini.com 87,
102
initscrt1.tmp files 95
input data
normal 27ff.
different form 172
Installation_of_PLOT3D 103
INTERACTIVE in SCRIPT 148
Internal energy (function #140) 126,
127
IRAKE 152
IRIS devices 140—141
IRIS Screen Dumps to Files 92—93,
95, 145—146
version 3.6 240—241, 256—257
IRIS swap space 168
IRIS_interface 143
IRISRGB Screen Dumps 94, 140—
141, 145—147,
IRISSINGLE 140—141
IZNEAR and IZFAR 103

J

JACOBIAN 154, 176
journal file 28, 85—86, 102

K

K=1 is the surface of the model 13, 77,
81
Kinetic energy (function #144) 127,
131

L

labels on the contour lines 90
laser printers 93, 95
LCL 227
error messages 177, 181, 193,
LHPS lines, hidden_lines, points, or
shaded_surface 43
lift 16, 74
Limiting Contour Lines to Mach 1 229
LINE 141
line (carpet plot) 61, 71, 141
Line_thickness 117
Line_type 117
LIST 90—91, 137
ln thck line thickness factor 43
ln tp line type 43

LN03 94, 95, 140, 141
 Loading Data —
 read/x=gridfile/q=solutionfile 27—
 30
 log of ... functions 243
 lpr (lpr -Pdevicefilename filename) 93

M

M, Mach number 68—70, function
 #154) 127, 132
 Mach 1. 229
 MANUAL qualifier to CONTOURS
 122
 MAP 138
 Material_coefficient (not implemented)
 36, 119
 MAXPOINTS=n 151
 MDATESET 153
 memory 168
 MEX 93, 143
 MGRID 29, 153
 MINMAX 41—42, 139
 mistake entering walls 58
 Momentum (Q2,Q3,Q4) 127, 133
 Mouse 13—14, 144
 movies 89, 94
 MRAKEVEC 78, 88—89
 multiple graphics ports 143
 multiple grid
 experimental and computational
 221—227
 IBLANK 50, 55—57, 115—116
 Q file format 165
 READ statement 29
 SUBSETS see SUBSETS
 xyz file format 163

N

NAS 1
 NASA Ames 1
 nfsmount 8, 28
 new tools 250
 NOATTRIBUTES (D) qualifier
 with CONTOURS 122
 with RAKES 150
 with SUBSETS 156
 with VECTORS 158
 with WALLS 161

NOAXES 142
 NOBLANK (D) 154
 NOCHECK qualifier with READ 29,
 154
 NOFIGURE (D) 142
 NOFULLSCREEN (D) 142
 NOJACOBIAN (D) 154
 Nondimensionalizations 129
 Normalized (internal) energy 127
 Normalized 2D stream function 127
 Normalized density 126
 Normalized enthalpy 126
 Normalized kinetic energy 127
 Normalized pressure 126
 Normalized stagnation density 126
 Normalized stagnation energy 127
 Normalized stagnation enthalpy 126
 Normalized stagnation pressure 126
 Normalized stagnation temperature 126
 Normalized temperature 126
 NOSCALAR_FUNCTION (D) 151,
 158
 NOX 139
 NOY 139
 NOZ 139
 Numerical Aerodynamic Simulation
 software flow 1,3

O

off-site users 9
 oil flow 81, 151, restrict with
 IBLANK 116
 omega *xvi*, 134, 245
 On-line Help
 how to use 23—26, 136
 reproduced 101—167
 Operating Systems
 PLOT3D versions for 5
 VAX/VMS 9
 UNIX 8, 86
 Changing PLOT3D for different
 176
 order of walls 34
 origin (or offset) of walls 125
 outline of the wing, body, and tail
 58—60
 Output device: PLOT qualifiers 140
 OUTPUT=file qualifier to LIST 137

P

p for screen save 92, 93, 145
 version 3.6 249
 p pressure 129
 page file quota, VMS 112, 176
 parm.plv files 95, 140
 Particle Traces 76—81
 colored by function 78
 functions listed 127
 path name in double quotes 22, 28
 perfect gas law 129
 perspective 14, to defeat 14
 Perturbation_velocity (function #203)
 127, 135
 Pierce, Larry 102
 Pinf 129
 Pitot_pressure (function # 116) 126,
 131
 Pitot_pressure ratio (function # 117)
 126, 131
 pixedit, pixps, pixqms and pixscan.
 93, 94
 pixel dumps 92—95
 PLANES file type 154
 planes orientation
 with CONTOURS 66
 with RAKE 151
 PLOT 46, 140—148
 Plot Range = minmax 41
 PLOT/CONTOUR 63—64
 PLOT/DIP 22, 94, 95, 140—141, 233
 PLOT/FSURFACE 61—62, 71-73
 PLOT/LABELS qualifier 90
 PLOT/LINE sample command file
 220—221
 PLOT3D code 3
 plot3d.jou 27, 85—86, 102
 plot3ddi 253
 plot3dini.com 87, 102
 PLOT3X 89—90
 PLOT4D (time) 89
 plotter 94—95
 points instead of lines 33
 polar coordinates 42
 polygon, backfacing, removal 146
 polygons of color 67
 POP 141
 popfil.dat files 95

PostScript (version 3.6 only) 95,
 192—193
 pp 131
 Pressure 126, 130
 Pressure gradient 128
 Pressure_coefficient 15, 64, 126, 130
 printers 93, 95
 prompt 47
 p0 130
 p0y 131

Q

q (dynamic pressure) 131
 q for QMS 93, 95, 146
 q solutions file 7, 164—166
 Q=file (D)qualifier for READ 153
 QCHECK 175
 Q_file formats 164—166
 QMS 95, 140
 QUIT 123, 149
 quota, VMS 112
 quotes 21—22

R

r for .ras screen save, version 3.6 248
 R 129
 RAKE/READ/WRITE 77—79, 88,
 150—152
 RANGE qualifier to CONTOURS 122
 RAS_screen_save 95, 145, 248
 RASP 92
 ratio of specific heats 129
 READ statement 28—30, 150, 153
 READCK (now QCHECK) 175
 Release_notes 104 ff., 235—242
 REMJAC 176
 Restrict
 RGL see CGL
 rho *xvi*, 129, 130
 right-handed system 124, 262
 rotation 13, 144
 Runge-Kutta 108

S

s for Seiko files 92
 s entropy formula 134
 Sample Command Files 87, 219—231
 SAVE qualifier 85, 123

- saving data and plots 85—95
 - for GAS animation 89—90
 - keyboard work 85—88
 - particle traces 88
 - screen dumps and draws 92—95
 - version 3.6 235—252
 - scalar plot 15
 - SCALAR_FUNCTION=function
 - number
 - RAKE 78, 151
 - VECTORS 158
 - Scalar_functions
 - tutorial 60—74
 - listed 126
 - SCALE_FACTOR of function axis 125
 - scanner 94
 - Scientific Visualization 1
 - screen dumps and draws (save) 92—
 - 95, 145—146
 - version 3.6 292—293
 - SCRIPT qualifier to PLOT 88, 143, 147
 - Seiko_color_printer_screen_save 93, 95, 145
 - sensitivity of the mouse 144
 - shaded_surface 33, 36, 117
 - shading pattern (solid only) 36, 118
 - Shininess 36, 119
 - Shock function 82—83, 128, 134
 - SHOW 155
 - Simple walls 58
 - slash (/) 21
 - solid 36
 - solution file 1
 - Spaces 21
 - Special_characters 115
 - Speed of sound (function #155)127
 - Stagnation density, enthalpy, pressure, pressure coefficient, temperature 126—127
 - Stagnation energy per unit volume (Q5) 127
 - steady-state flow 77
 - stills 92—95
 - STOP in SCRIPT 148
 - streamlines 77
 - SUBSETS 53 ff., 156
 - attributes 116
 - compared to walls 53
 - for slices on the wing 65
 - for rings around the surface of wing and tail 72
 - READ statement resets to everything 30
 - set to everything for particle traces 77
 - SURF graphics application 2, 91
 - SURFACE (carpet) *qualifier* to PLOT 61—62, 73, 141
 - changed to FSURFACE in v.3.6 236, 247
 - surface plots, compared to contour 61—62
 - surface as *attribute* of WALLS, SUBSETS or CONTOURS 61—62, 117
 - swap space 168
 - Swirl (function # 184) 127, 134
 - sym size symbol size factor 118
 - sym symbol type 118
 - Symbol size and type 118
 - for VECTORS 158
 - symmetry plane subset 68—69
 - symmetry planes with RAKE (IJ, IK, JK) 151
 - system-command 120
- T**
- T 132
 - tape, wingbodytial on 9, 258— 259
 - TEKTRONIX (4014) 94—95, 140
 - Temperature (function #120) 126, 131
 - Testdrive 9—20
 - TEXT 21—22, 31, 157
 - as format type 137
 - time 76, 89, 108, 150
 - TK41XX 140
 - TOP,SIDE,FRONT 159
 - tools 250
 - translating (dragging) 14
 - transparency 36, 119
 - TUTORIAL 6—96
 - Tutorial Symbols and Styles 7
 - Typical Command Sequence 234

U

u velocity 127
 UNFORMATTED 136, 153
 UP=axis 142

V

IVI 129
 V 134
 v velocity (function #150) 127
 VAX
 getting tutorial 8—9
 location of wingbodytail 8, 258
 versions available 5
 VMS/VAX virtual memory 168
 Vector Functions 75—76, 127—128
 Vector length scale factor 76, 158
 Vector scaling 76, 158
 vector field directions in RAKE. 150
 VECTORS command 75—76, 158
 vectors colored by a scalar function 76, 158
 VECTR1.PLV files 95, 140
 Velocity magnitude (function #153) 127
 Velocity x vorticity (function #204) 128
 Velocity x vorticity magnitude (function #185) 127
 Velocity (function #200) 75, 127
 VERSATEC 95, 140
 video tape 89
 videos at NAS 94
 VIEW command 159
 in fsurface plots 71
 viewgraphs 93
 viewpoint, see VPOINT
 VMS/VAX virtual memory 168—169
 Vortex lines 127
 Vorticity 127, 134
 Vorticity magnitude 127
 VPOINT command 42, 160
 VPOINT in SCRIPT 148

W

w velocity (function #152) 127
 WALLS command 12—13, 32—40, 50—53, 161
 attributes 116
 compared to subsets 53
 order of 34

Walls alone (geometry) (function #0) 50—52, 126

WALLS_ORIGIN qualifier to FSURFACE 73, 125

Warning Messages 206—218

wbt.com (testdrive) 9—20

wbtout.com 85—86

WHOLE (D) 154

wind tunnel results 4, 221—226

window manager 10, 95, 143

wingbodytail 7 ff., 258—259

wingbodytail, location of 8—9, 258—259

wk23 8

WRITE qualifier to RAKE78, 88, 150

X

X (D) qualifier to MINMAX 138

x, y, and z coordinates in VPOINT 42

x-component of vorticity (f. #180) 127

x-momentum (Q2) (function #160) 127

XSCALE qualifier to MINMAX 139

XY,XZ,YZ,YX,ZX,ZY in VIEW command 159

XYZ

 file 7

 file in READ 153

 in RAKE 150

 VPOINT 160

xyz_file multiple_grid formats 164

xyz_file single_grid formats 162

xyz_file with_IBLANK formats 163

Y

Y (D) qualifier to MINMAX 139

y-component of vorticity (function #181) 127

y-momentum (Q3) (function #161) 127

YSCALE qualifier to MINMAX 139

Z

Z (D) qualifier to MINMAX 139

z-buffering 103, 147

z-component of vorticity (function #182) 127

z-momentum (Q4) (function #162) 127

zooming 13—14, 144

ZSCALE qualifier to MINMAX 139

1. Report No. NASA TM-101067		2. Government Accession No.		3. Recipient's Catalog No.	
4. Title and Subtitle PLOT3D User's Manual				5. Report Date March 1990	
				6. Performing Organization Code	
7. Author(s) Pamela P. Walatka*, Pieter G. Buning, Larry Pierce†, and Patricia A. Elson†				8. Performing Organization Report No. A-89029	
				10. Work Unit No.	
9. Performing Organization Name and Address Ames Research Center, Moffett Field, CA 94035-1000 *Computer Sciences Corporation, Moffett Field, CA 94035 †Sterling Software, Moffett Field, CA 94035				11. Contract or Grant No. NAS2-12961	
				13. Type of Report and Period Covered Technical Memorandum	
12. Sponsoring Agency Name and Address National Aeronautics and Space Administration Washington, DC 20546-0001				14. Sponsoring Agency Code	
15. Supplementary Notes Point of Contact: Pieter G. Buning, Ames Research Center, MS 258-2, Moffett Field, CA 94035-1000 (415) 604-4492 or FTS 464-4492					
16. Abstract PLOT3D is a computer graphics program designed to visualize the grids and solutions of computational fluid dynamics. Seventy-four functions are available. Versions are available for many systems. PLOT3D can handle multiple grids with a million or more grid points, and can produce varieties of model renderings, such as wireframe or flat shaded. Output from PLOT3D can be used in animation programs. The first part of this manual is a tutorial that takes the reader, keystroke by keystroke, through a PLOT3D session. The second part of the manual contains reference chapters, including the helpfile, data file formats, advice on changing PLOT3D, and sample command files.					
17. Key Words (Suggested by Author(s)) PLOT3D, Computational fluid dynamics, Computer graphics, Scientific visualization, Numerical aerodynamic simulation			18. Distribution Statement Unclassified-Unlimited Subject Category-61		
19. Security Classif. (of this report) Unclassified		20. Security Classif. (of this page) Unclassified		21. No. of Pages 304	22. Price A13