AFRL-RV-PS-
TM-2023-0004

AFRL-RV-PS-
TM-2023-0004

OBJECT TOOLKIT VERSION 4.5 USER'S MANUAL

V.A. Davis, et al.

Leidos, Inc.
10260 Campus Point Drive, Mailstop C4
San Diego, CA 92121

31 May 2023

Interim Report

APPROVED FOR PUBLIC RELEASE; DISTRIBUTION IS UNLIMITED.

AIR FORCE RESEARCH LABORATORY
Space Vehicles Directorate
3550 Aberdeen Ave SE
AIR FORCE MATERIEL COMMAND
KIRTLAND AIR FORCE BASE, NM 87117-5776

DTIC COPY

NOTICE AND SIGNATURE PAGE

Using Government drawings, specifications, or other data included in this document for any purpose other than Government procurement does not in any way obligate the U.S. Government. The fact that the Government formulated or supplied the drawings, specifications, or other data does not license the holder or any other person or corporation; or convey any rights or permission to manufacture, use, or sell any patented invention that may relate to them.

This report was cleared for public release by AFMC/PA and is available to the general public, including foreign nationals. Copies may be obtained from the Defense Technical Information Center (DTIC) (http://www.dtic.mil).

AFRL-RV-PS-TM-2023-0004 HAS BEEN REVIEWED AND IS APPROVED FOR PUBLICATION IN ACCORDANCE WITH ASSIGNED DISTRIBUTION STATEMENT.

//SIGNED//	//SIGNED//
_____	_____
Dr. David L. Cooke	Mark E. Roverse, Chief
Program Manager/AFRL/RVB	AFRL Geospace Technologies Division

This report is published in the interest of scientific and technical information exchange, and its publication does not constitute the Government's approval or disapproval of its ideas or findings.

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing this collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden to Department of Defense, Washington Headquarters Services, Directorate for Information Operations and Reports (0704-0188), 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302. Respondents should be aware that notwithstanding any other provision of law, no person shall be subject to any penalty for failing to comply with a collection of information if it does not display a currently valid OMB control number. **PLEASE DO NOT RETURN YOUR FORM TO THE ABOVE ADDRESS.**

1. REPORT DATE *(DD-MM-YYYY)*	**2. REPORT TYPE**	**3. DATES COVERED** *(From - To)*
31-05-2023	Interim Report	01 Sep 2020 – 31 May 2023

4. TITLE AND SUBTITLE
Object Toolkit Version 4.5 User's Manual

5a. CONTRACT NUMBER
FA9453-19-C-0400

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBER
62601F

6. AUTHOR(S)
V. A. Davis, B. M. Gardner, and M. J. Mandell

5d. PROJECT NUMBER
1010

5e. TASK NUMBER
EF132583

5f. WORK UNIT NUMBER
V17X

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Leidos
10260 Campus Point Drive, Mailstop C4
San Diego, CA 92121

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)
Air Force Research Laboratory
Space Vehicles Directorate
3550 Aberdeen Avenue SE
Kirtland AFB, NM 87117-5776

10. SPONSOR/MONITOR'S ACRONYM(S)
AFRL/RVBX

11. SPONSOR/MONITOR'S REPORT NUMBER(S)
AFRL-RV-PS-TM-2023-0004

12. DISTRIBUTION / AVAILABILITY STATEMENT
Approved for public release; distribution is unlimited (AFRL-2023-3674 dtd 27 Jul 2023).

13. SUPPLEMENTARY NOTES

14. ABSTRACT
Object Toolkit is a finite-element model builder specifically designed for creating representations of spacecraft surfaces for spacecraft environment interactions computer codes. It is used by Nascap-2k, EPIC (Electric Propulsion Interactions Code), and MEM (Meteoroid Engineering Model). This document describes the use of the code.

15. SUBJECT TERMS
object toolkit, Nascap-2k, EPIC, GMsh

16. SECURITY CLASSIFICATION OF:			**17. LIMITATION OF ABSTRACT**	**18. NUMBER OF PAGES**	**19a. NAME OF RESPONSIBLE PERSON**
					Dr. David L. Cooke
a. REPORT	**b. ABSTRACT**	**c. THIS PAGE**			**19b. TELEPHONE NUMBER** *(include area code),*
Unclassified	Unclassified	Unclassified	Unlimited	168	

Standard Form 298 (Rev. 8-98)
Prescribed by ANSI Std. 239.18

This page is intentionally left blank.

Approved for public release; distribution is unlimited.

TABLE OF CONTENTS

Section **Page**

List of Figures v

List of Tables xiii

I Introduction 1

1 What Is *Object Toolkit*? 1

2 Installation 4

II Using *Object Toolkit* 5

3 Components and Assemblies 5

4 General Guidance 5

5 Cursor Tools and Direct Movement and Rotation 6

6 Standard Components 7

6.1 Box 8

6.2 Panel 9

6.3 Boom 10

6.4 Cylinder 11

6.5 Dish 11

7 Aligning and Positioning Components 12

8 Grouping Components 15

9 Moving, Aligning, and Grouping Components Using Wizards 16

10 Materials 24

10.1 Changing Surface Material Assignments 25

11 Mesh Editing 25

11.1 Local Mesh Refinement 26

11.2 Element Compatibility 27

11.3 Deleting and Creating Elements 27

11.4 Changing Connectivity (Creating and Deleting Nodes) 32

11.5 Combining Elements 34

11.6 Changing Surface Material Assignments 36

12 Importing Components from Files 37

12.1 Importing Components Built Using *Object Toolkit* 37

12.2 Importing Components from a *PATRAN* Neutral File 37

12.3 Importing Components from a *NX I-DEAS* TMG ASCII VUFF File 37

TABLE OF CONTENTS (continued)

Section			Page
	12.4	Importing Components from a *Gmsh* MSH Formatted File	37
	12.5	Importing Components from a *Nastran* BDF or DAT File	38
	12.6	Importing Components from a *I-DEAS* Universal File	38
	12.7	Importing Components from a STEP File	38
13	Building Custom Components Using Advanced Mesh Editing and Primitives		38
	13.1	Creating a Cavity	39
	13.2	Creating a User-Specified Shape	41
	13.3	Scaling a Component	42
14	Editing Components Within an Assembly		43
15	Application-Specific Attributes		45
	15.1	Conductor Number	45
	15.2	Solar Array Wizard	45
	15.3	Conductivity	46
	15.4	Particle Instruments (Emitters and Detectors)	47
	15.5	Antenna Surface Currents (Injection Point)	47
	15.6	Pointing (Subsystems)	48
	15.7	Thrusters	49
	15.8	Neutralizers	49
	15.9	Magnetic dipoles	49
	15.10	Move Wizards	50
	15.11	Orbits	50
	15.12	Materials	52
III	*Object Toolkit* Menus		53
16	Menus		53
IV	Examples		60
17	Quick Start		60
	17.1	Create a Box	60
	17.2	Rotate the View	61
	17.3	Change the Dimensions and Zoning	62
	17.4	Change Surface Material	63
	17.5	Save and Reopen Object File	65
18	Typical Geosynchronous Spacecraft for *Nascap-2k*		67

TABLE OF CONTENTS (continued)

Section **Page**

18.1 Create the Body 68
18.2 Create and Add the Omni-Antenna 69
18.3 Create and Add the Dish Antenna 75
18.4 Create and Attach the Solar Arrays 76
19 Half Cylinder Sunshade 82
19.1 Create Basic Component 84
19.2 Remove Extra Surfaces 84
19.3 Create Inside Surface 85
19.4 Create Object With Inner and Outer Surfaces 86
19.5 Join Inner and Outer Surfaces 89
19.6 Add Instrument 90
19.7 Consolidate to Simplify Editing 92
19.8 Complete Instrument 93
20 Triangular Solar Sail 96
21 Attaching a Boom to a Boom 99
21.1 Make a T Shape 99
21.2 Make a Y Shape 100
22 Using *Gmsh* to Create Finite Element Model from STEP File 106
22.1 *Gmsh* Interface 106
22.2 Open a Geometric Model 107
22.3 Create a Mesh on the Object 107
22.4 Export the Mesh and Import to *Object Toolkit* 109
22.5 Common Manipulations 110
22.6 Tweaking the Mesh Size 110
23 Spacecraft for *EPIC* 112
23.1 Main Body 112
23.2 Adding a Boom 114
23.3 Adding a Solar Array 116
23.4 Adding the –Y Solar Array 118
23.5 Adding a Thruster 120
23.6 Setting the Solar Arrays to Track the Sun 121
23.7 Adding an Antenna 122

Approved for public release; distribution is unlimited.

TABLE OF CONTENTS (continued)

Section **Page**

24 Spacecraft with an Orbit 123
References 125

Appendix A: Input file formats 126

A.1 *PATRAN* Neutral File 126
A.2 *NX I-DEAS* TMG ASCII VUFF File 128
A.3 *Gmsh* MSH Formatted File 129
A.4 *Nastran* BDF or DAT File 132
A.5 *I-DEAS* Universal File 134
A.6 STEP files 135
A.7 SEE Interactive Spacecraft Charging Handbook Material Definition File 137
Appendix B: Output file format 138
B.1 XML 138
B.2 *Object Toolkit* File Format 138
Appendix C: Customization 141
C.1 General 141
C.2 Optional Capabilities 143
C.3 Application-specific Surface Attributes 143
C.4 Special Components 144
C.5 Default Material Definitions 145
Glossary 146

LIST OF FIGURES

Figure **Page**

Figure 1. Simple Spacecraft Model. 1

Figure 2. *Nascap-2k* Model of the MESSENGER Spacecraft, Showing Biased Solar Array Surfaces. 2

Figure 3. DMSP Spacecraft Model Constructed in *Object Toolkit* 2

Figure 4. *Object Toolkit* Screen Showing the Five Standard Components. 8

Figure 5. Definition of a 3 m by 4 m by 5 m Rectangular Box with a Different Material on Each Side. 9

Figure 6. Definition of a Nonrectangular Box. 9

Figure 7. Definition of Rectangular Panel. 10

Figure 8. Definition of Nonrectangular Panel. 10

Figure 9. Definition of a Twisted Boom. 11

Figure 10. Definition of a Tapered Cylinder 11

Figure 11. Definition of a Dish. The Screen on the Right Shows Two Dishes Viewed from Different Directions. 12

Figure 12. Three Boxes Used to Illustrate Alignment and Positioning Components. The Left-hand Box Is Centered at the Origin. The Right-hand Box Is Centered at (6, 0, 0). The Small Upper Box Is at (10, 5, 0). The View Is Shifted Slightly to the Right and Perspective Is Turned on. 13

Figure 13. Using "Place" on the Component Menu, Which Operates on the Selected Component, to Move the Middle Box Vertically So That Its Center Is at (6, 2, 0) 13

Figure 14. Using "Node Relative Move" on the Component Menu, Which Operates on the Component Containing the Selected Node, to Move the Middle Box to the Left Side of the Other Box with the Upper Left Node at (–5, 4, 2.5). Selected Node Is Circled. 14

Figure 15. Using "Align Edge" on the Component Menu, Which Operates on the Component Containing the Selected Edge, to Rotate the Left-hand Box (Which Was on the Right in Figure 13). The Rotation to Align the Edge in the (0.5, 1.0, 0.0) Direction Takes Place in the Component Local Coordinate System 14

Figure 16. The Results of Translating and Then Rotating the Box Using the Cursor Tools 15

Figure 17. Alignment of an Assembly (Multiple Components That Have Been Combined into a Single Component) So That the Edge Is Oriented in the (0.5, 1.0, 0.0) Direction 16

Figure 18. Moving, Aligning, and Grouping Two Components Using the "Element to Element Wizard." The Four Trapezoidal Elements Belong to the Weld. In the Third Frame, the Fixed Component Is Automatically Hidden. 17

Figure 19. Moving, Aligning, and Grouping Two Components Using the "Element to Edge Wizard." In the Third Frame, the Fixed Component Is Automatically Hidden. 19

Approved for public release; distribution is unlimited.

LIST OF FIGURES (continued)

Figure **Page**

Figure 20. Moving, Aligning, and Grouping Two Components Using the "Edge to Edge Wizard." In the Second Frame, the Fixed Component Is Automatically Hidden. 21

Figure 21. Moving, Aligning, and Grouping Two Components Using the "Surface to Surface Wizard." In the Second Frame, the Fixed Component Is Automatically Hidden. In the Last Frame, the Outlines of Elements with Normals Facing into the Screen Are Seen. 23

Figure 22. Material Property Definition Dialog Box. 24

Figure 23. Changing the Surface Materials Using "Materials" on the Mesh Menu. 25

Figure 24. Creating Smaller Elements Using "Divide Element" on the Mesh Menu. 27

Figure 25. Deleting an Element Using "Delete Element" and Replacing It with a Quadrilateral and a Triangle by Selecting Edges and Using "Create Element" on the Mesh Menu. In b, c, and d, the Outlines of Elements on the Far Side with Normals Facing into the Screen Are Seen. 29

Figure 26. Deleting an Element Using "Delete Element" and Replacing It with a Quadrilateral and a Triangle by Selecting Nodes and Using "Create Element" on the Mesh Menu. In b, c, and d, the Outlines of Elements on the Far Side with Normals Facing into the Screen Are Seen. 30

Figure 27. Filling in the Gap Left by the "Surface to Surface Wizard." In a, b, c, and d, the Outlines of Elements with Normals Facing into the Screen Are Seen. 31

Figure 28. Resolving Incompatibilities by Dividing an Element Using "Divide Element" and "Combine Nodes" to Create Two Quadrilaterals and a Triangle in Place of the Original Single Element. 33

Figure 29. Resolving Incompatibilities by Dividing an Element, Combining Nodes Twice to Create Two Quadrilaterals and Two Triangles, and, Finally, Combining the Two Triangles into a Single Quadrilateral to Give Three Quadrilaterals in Place of the Original Single Element. In e, the Outlines of Elements on the Far Side with Normals Facing into the Screen Are Seen. 35

Figure 30. Changing the Surface Materials of Newly Created Elements by Going to the Mesh Menu and Choosing "Materials." 36

Figure 31. Creating a Cavity Using "Reverse Element" on the Mesh Menu. In b, c, d, and e, the Outlines of Elements with Normals Facing into the Screen Are Seen. 40

Figure 32. Changing the Edge of a Panel from a Straight Edge to a Cosine Curve by Repeated Use of "Edit Node" on the Mesh Menu. 41

Figure 33. Creation and Scaling of a Primitive. A Dish Is Converted to a Primitive and Then Scaled Anisotropically.....42

Approved for public release; distribution is unlimited.

LIST OF FIGURES (continued)

Figure **Page**

Figure 34. Editing of a Box Several Layers Down in the Assembly Structure. The Surface Materials and the Extent of the Box in the Y Direction Are Changed. Upon Returning to the Top Assembly, the Surface Material Changes Appear As Desired, but the Change in the Proportions of the Box Results in a Renumbering That Scrambles the Final Elements Added to Connect the Cylinder and the Box....44

Figure 35. Defining a Solar Array String using the "Solar Array Wizard." 46

Figure 36. Detector Properties Dialog Box. 47

Figure 37. Subsystem Properties Dialog Box. 48

Figure 38. Thruster Properties Dialog Box. 49

Figure 39. Heliocentric Orbit Dialog Box. 50

Figure 40. Geocentric Orbit Dialog Box. 51

Figure 41. Time and Stabilization Dialog Box. 51

Figure 42. Dialog Box for Creating a Box 3.0 m by 4.0 m by 5.0 m in Size and Five Elements in Each Direction. The ±Y Sides Are Defined to Be Material OSR and the Other Sides Are Black Kapton. 60

Figure 43. The Box Created from the Definition in Figure 42. 61

Figure 44. Rotated View of the Created Box. 61

Figure 45. Editing the Box to Reduce Its Size and Number of Elements in the Y Direction. 62

Figure 46. Box Resulting from Modifications Shown in Figure 45. 63

Figure 47. Box with Elements Selected. 64

Figure 48. Box with a Different Material on Selected Elements. 64

Figure 49. Save As Dialog Box. 65

Figure 50. Open Dialog Box. 66

Figure 51. Box after Reopening. 66

Figure 52. Box Viewed from a Different Direction. 67

Figure 53. *Object Toolkit* Model of a Typical Geosynchronous Spacecraft to Be Built in this Example. 67

Figure 54. Defining a Box to Form the Body of the Spacecraft. The Box is 1.76 m by 2.64 m by 2.64 m in Size. Three Elements in the X Direction, and Five Elements in Each of the Other Two Directions. The ±X Sides Are Teflon. The ±Y Sides Are OSR. The ±Z Sides Are Black Kapton. 68

Figure 55. Changing the Materials on the Body Surfaces to Be as Desired. 69

Figure 56. Defining a New Material, NPaint, Which Is Used to Cover the Omni-antenna 69

Approved for public release; distribution is unlimited.

LIST OF FIGURES (continued)

Figure **Page**

Figure 57. Defining a Cylinder to Represent the Omni-antenna. The Cylinder is 0.88 m in Height, 0.99 m in Diameter, Two Elements High, and Eight Elements in Circumference. All the Elements Are of the Newly Defined Material NPaint 70

Figure 58. Deleting the Bottom Surface of the Cylinder and Five Elements on the Top of the Body So That the Two Can Be Joined. The Outlines of Elements with Normals Facing into the Screen Are Seen. 71

Figure 59. Placing the Cylinder So That Its Bottom Surface Is in the Plane of the Top of the Body (Z = 1.32) and the Center Lines Coincide (X = 0 and Y = 0). The Outlines of Elements with Normals Facing into the Screen Are Seen. 72

Figure 60. Saving the Resulting Object, Which Has Errors in It. The Outlines of Elements with Normals Facing into the Screen Are Seen. 72

Figure 61. Moving the Nodes of the Body That Are Near the Nodes of the Cylinder to Be Coincident with the Nodes of the Cylinder. The Outlines of Elements with Normals Facing into the Screen Are Seen. 73

Figure 62. Joining the Omni-antenna and the Body. The Outlines of Elements with Normals Facing into the Screen Are Seen 74

Figure 63. Changing the Materials Assigned to the Top Surface 74

Figure 64. Defining Graphite Dishes to Represent the Dish Antennas. The Antenna Diameter Is 2 m, the Depth Is 0.1 m, and the Antenna Is Three Elements Across 75

Figure 65. Placing the Dish Antennas So That the Top of the Disks and the Bottom of the Body Are in the Same Plane, the Dishes Are Centered in the Y Direction, and the Nearest Edges of the Disks are 0.88 m from the Body. Selected Nodes Are Moved to (1.76, -0.259, -1.32) and (-1.76, -0.259, -1.32). 76

Figure 66. Defining a Boom to Support a Solar Array. The Boom Is 2.0 m Long, Has Two Elements in the Long Direction, and Has a Twist of 33°. The Boom Material Is Kapton. 77

Figure 67. Defining a Panel to Represent a Solar Array Wing. The Panel Is 4.4 m in Width and 7 m in Length. It Is Seven Elements Wide and Fifteen Elements Long. The Front Side Is Material Solar Cells and the Back Is Black Kapton. 77

Figure 68. Attaching a Solar Array Panel to a Boom Using the "Edge to Element Wizard." 78

Figure 69. Copying the First Array to Create the Second. 79

Figure 70. Changing the Twist Angle on the Second Boom and Saving the Resulting Object. 80

Figure 71. Attaching a Solar Array Panel and Boom to the Spacecraft Body Using the "Element to Element Wizard." 81

Figure 72. Attaching the Second Solar Array to the Body, and Rotating It So That It Is Parallel to the First Solar Array 82

Figure 73. *Object Toolkit* Model of the MESSENGER Spacecraft. 83

Approved for public release; distribution is unlimited.

LIST OF FIGURES (Continued)

Figure **Page**

Figure 74. Simplified MESSENGER Sunshade, Including DSAD Instrument, to Be Built in this Example....83

Figure 75. Creating the Cylinder....84

Figure 76. Deleting the Extraneous Top, Bottom, and Side Elements. The Outlines of Elements with Normals Facing into the Screen Are Seen....85

Figure 77. Saving the Half-Cylinder Shell. The Outlines of Elements with Normals Facing into the Screen Are Seen....85

Figure 78. Changing the Surface Material from Gold to Aluminum. The Selected Elements Are Displayed in White. The Other Elements Have Not Been Changed Yet. The Outlines of Elements with Normals Facing into the Screen Are Seen....86

Figure 79. Reversing the Normal of the Elements. In the Figure, Some of the Distant Elements Have Already Been Reversed and Some of the Near Ones Have Been Selected and Are About to Be Reversed. The Outlines of Elements with Normals Facing into the Screen Are Seen....86

Figure 80. Adding the Outer Surface of the Sunshade from the Previously Saved File. The Outlines of Elements with Normals Facing into the Screen Are Seen....87

Figure 81. Converting the Outer Surface to a Primitive. The Outlines of Elements with Normals Facing into the Screen Are Seen....88

Figure 82. Scaling the Outer Edge by Expanding It in the X and Y Directions by 10%. The Outlines of Elements with Normals Facing into the Screen Are Seen....88

Figure 83. Resulting Inner and Outer Surfaces of the Sunshade. The Outlines of Elements with Normals Facing into the Screen Are Seen....89

Figure 84. Creating an Element That Joins the Two Sides of the Sunshade. The Outlines of Elements with Normals Facing into the Screen Are Seen....90

Figure 85. Hemispherical Sunshade....90

Figure 86. Dividing a Corner Element....91

Figure 87. Result after Deleting Three Elements and Replacing Them with Two to Eliminate the Incompatibility along the Face of the Sunshade. Incompatible Elements Remain along the Edge....91

Figure 88. Result after Dividing the Corner Element and Then Dividing Each of the Newly Formed Elements....92

Figure 89. Result after Deleting the Center Four Elements of the Finely Divided Region and Then Selecting the Four Corner Nodes....92

Figure 90. Here, *Object Toolkit* Misinterpreted the Attempt to Create a New Element Using the Four Corner Nodes....93

Figure 91. New Element Created after Consolidation and Then Selecting Four Nodes and Requesting “Create Element.”....93

Approved for public release; distribution is unlimited.

LIST OF FIGURES (continued)

Figure **Page**

Figure 92. Result after Deleting Seven Small Elements and Replacing Them with Two Compatible, Larger Ones. The Outlines of Elements with Normals Facing into the Screen Are Seen. 94

Figure 93. Replacing the Existing Side Element with Two New Elements, a Quadrilateral and a Triangle. The Outlines of Elements with Normals Facing into the Screen Are Seen. 94

Figure 94. Replacing the Small Elements with Larger Compatible Ones to Form the DSAD Frame. The Outlines of Elements with Normals Facing into the Screen Are Seen. 95

Figure 95. Remeshed Side Element. 95

Figure 96. Completed Model of the DSAD Instrument. 95

Figure 97. Top View of Sample Solar Sail Created in This Example 96

Figure 98. A Four-by-four Panel is the Starting Point for a Small Solar Sail 96

Figure 99. Deleting Half of the Panel Leaves Open Edges. 97

Figure 100. Adding Triangular Elements by Going to the Mesh Menu and Choosing "Create Element." 97

Figure 101. Closing the Open Edge in the Center By Going to the Mesh Menu and Choosing "Combine Nodes." The Outlines of Elements with Normals Facing into the Screen Are Seen. 97

Figure 102. Filling in the Edge with Additional Triangular Elements, While Making Sure That the Two Sides Remain Separate. The Outlines of Elements with Normals Facing into the Screen Are Seen. 98

Figure 103. Replacing Each of the Square Elements in the Corners with a Triangular One, One at a Time. The Outlines of Elements with Normals Facing into the Screen Are Seen. 98

Figure 104. Adding Additional Triangular Elements to Fill in the Final Holes. 98

Figure 105. Moving Each Node to Form the Curved Edge by Going to the Mesh Menu and Choosing "Move Node." 99

Figure 106. Booms to Be Connected into a T Shape 99

Figure 107. Joining Two Booms into a T Shape Using the "Element to Element Wizard." 100

Figure 108. Booms to Be Connected into a Y Shape 100

Figure 109. Rotation of Short Boom to Be 45^{o} from the Long Boom. (Change Edge Vector from (0.0, 0.0, 0.5) to (0.0, 0.5, -0.5). 101

Figure 110. Moving the Short Side of the Y into Position by Going to the Component Menu and Selecting "Node Relative Move." The Bottom Corner Node Is Moved to (0.05, 0.05, -0.293) 101

Figure 111. Consolidating Booms into a Single Assembly and Moving Nodes to Coincide 102

Figure 112. Deleting Elements in Preparation for Completing Y 103

Approved for public release; distribution is unlimited.

LIST OF FIGURES (Continued)

Figure **Page**

Figure 113. Moving End Nodes of the Short Boom to Be Along Face of the Long Boom. In These Figures, Perspective Is Turned Off. 104
Figure 114. Moving Nodes of Long Boom to Coincide with Nodes on Short Boom. 105
Figure 115. Moving Nodes on Long Boom So That All Elements Are Approximately Rectangular. 105
Figure 116. Resulting Y Shape Created from Two Booms 106
Figure 117. Gmsh user interface 106
Figure 118. The imported object football.stp, as initially shown in Gmsh. 107
Figure 119. The expanded **Mesh** menu, and the triangular mesh created on the soccer ball by the "2D" command. 108
Figure 120. The soccer ball with a quadrilateral mesh created using the "Recombine 2D" command. 108
Figure 121. Gmsh Export dialog. 109
Figure 122. Meshed soccer ball as imported into Object Toolkit. 110
Figure 123. Options menu used for adjusting mesh size. 111
Figure 124. Coarse gridded (left) and fine gridded (right) soccer balls as shown in *Object Toolkit*. (Compare with Figure 142.) 111
Figure 125. *EPIC* Object Colored by Subsystem Name. 112
Figure 126. Specifying a Box to Represent the Main Body 113
Figure 127. Assigning a Subsystem Name to the Box. 113
Figure 128. Displaying the Object Colored by Subsystem Name. 114
Figure 129. Boom Component for *EPIC* Object. 114
Figure 130. Assigning a Subsystem Name to the Boom 115
Figure 131. Placing the Boom Next to the Box Using the "Element to Element Move Wizard." 115
Figure 132. Solar Array for *EPIC* Object. 116
Figure 133. Assigning a Subsystem Name to the Solar Array 117
Figure 134. Placing the Solar Array at the End of the Boom Using the "Element to Element Move Wizard." 118
Figure 135. *EPIC* Object Colored by Subsystem Name, Showing Both Solar Arrays. 119
Figure 136. *EPIC* Object Colored by Material Name. 119
Figure 137. Thruster Definition for *EPIC* Object. Thruster Is Located at (0.0, -1.4, 0.6) and Is Pointed in the –Y Direction. 120

Approved for public release; distribution is unlimited.

LIST OF FIGURES (Continued)

Figure **Page**

Figure 138. *EPIC* Object with Thruster. Thruster Is Located at (0.0, -1.4, 0.6) and Is Pointed in the –Y Direction. 120

Figure 139. Setting SolarArray1 to Track the Sun. The Fixed Point Is (0.0, 0.0, 0.0). The Pointing Vector Is (0.0, 0.0, 1.0). The First Axis Is (0.0, 1.0, 0.0) and the Second Axis Is the Same. 121

Figure 140. Dish Antenna for *EPIC* Object. 122

Figure 141. Moving Antenna to the +Z Side at (0.44, -0.44, 1.46) 122

Figure 142. Geocentric Orbit Dialog Box. 123

Figure 143. Applying Pointing Rotations Using the Time and Stabilization Dialog Box. 124

Figure 144. *EPIC* Object with Rotated Solar Arrays. 124

Figure 145. *PATRAN* Neutral File Defining an Eight-node, Six-element Cube 127

Figure 146. *Gmsh* MSH Formatted File Defining an Eight-node, Six-element Cube 130

Figure 147. *Nastran* BDF (or DAT) File Defining an Eight-node, Six-element Cube 132

Figure 148. *I-DEAS* Universal File Defining an Eight-node, Six-element Cube 134

Figure 149. Selected Lines from a STEP File Defining an Eight-node, Six-element Cube 136

Figure 150. *SEE Handbook* Material Definition File for Material Kapton with Default Properties. 137

Figure 151. Example of an *Object Toolkit* XML Output File 138

Figure 152. Series of Node Tags. 139

Figure 153. Series of Element Tags. 139

Figure 154. Attribute Properties for a Specified Value of "Subsystem." 140

Figure 155. Special Component Properties for an Instance of "Thruster." 140

Figure 156. Material Properties Specification in Object Definition File 141

Figure 157. *EPIC* Application-specific Parameter File, EPIC_OTKSpecs.xml 142

Figure 158. *Nascap-2k* Application-specific Parameter File, Nascap2k_OTKSpecs.xml 143

Figure 159. Application-specific Attribute Property Edit Dialog Box. 144

Figure 160. Special Component Edit Dialog Box for "Thruster1." The Object Appears As a Sphere at the Right of the Figure 145

Approved for public release; distribution is unlimited.

LIST OF TABLES

Table **Page**

Table 1. Formatting Conventions. 3
Table 2. Cursor Tool and Direct Movement and Rotation Buttons. 7
Table 3. File Menu....53
Table 4. Edit Menu. 54
Table 5. View Menu. 55
Table 6. Component Menu. 56
Table 7. Mesh Menu. 57
Table 8. Wizard Menu. 58
Table 9. Material Menu. 59
Table 10. Orbit* Menu. 59
Table 11. Help Menu. 59
Table 12. Correspondence between Material Property Numbers and Names 137

Approved for public release; distribution is unlimited.

This page is intentionally left blank.

I INTRODUCTION

1 What Is *Object Toolkit*?

Object Toolkit is a finite-element model builder specifically designed for creating representations of spacecraft surfaces for spacecraft environment interactions computer codes. *Object Toolkit* was originally developed to build an object (generally a spacecraft) for the spacecraft-plasma environment interactions computer code *Nascap-2k*. It is also used to define spacecraft for *EPIC* (*Electric Propulsion Interactions Code*)[1] and *MEM (Meteoroid Engineering Model)* and can be customized to create geometric models for other analysis codes.

The motivation behind the development and distribution of *Object Toolkit* is twofold. First, because *Object Toolkit* is provided with the distribution of codes such as *Nascap-2k* and *EPIC*, the user is not required to purchase or learn expensive finite element generators to create system models. Second, *Object Toolkit* is tailored for space environments effects modeling with these codes, rather than being primarily intended for some other purpose. As such, it can serve the needs of such tools more effectively than other available software.

Object Toolkit provides five standard components that can be rezoned, resized, and assembled. In addition, *Object Toolkit* can import a finite element model from standard finite-element preprocessors such as *PATRAN* and *Gmsh* or an object previously created with *Object Toolkit*. Direct mesh editing capabilities are available to subdivide surface elements and to build components with complex shapes. *Object Toolkit* allows the user to create a database of materials and edit their properties.

Figure 1 through Figure 3 show some examples of spacecraft defined using *Object Toolkit*. While all of these figures show spacecraft, the object generated is not limited to spacecraft. A single instrument can be an *Object Toolkit* object as can a combined solar array wing and ion thruster subsystem.

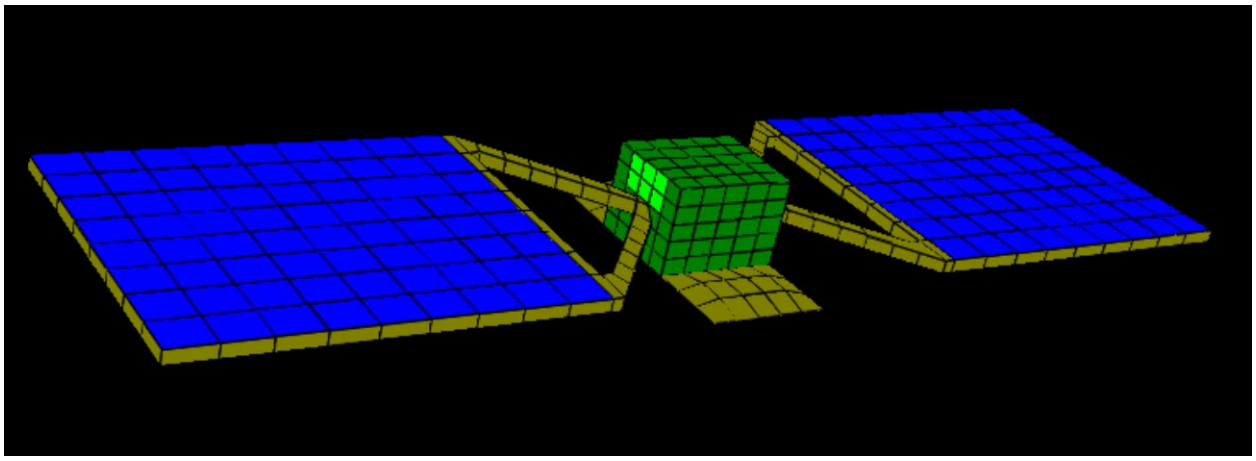


Figure 1. Simple Spacecraft Model.

Approved for public release; distribution is unlimited.

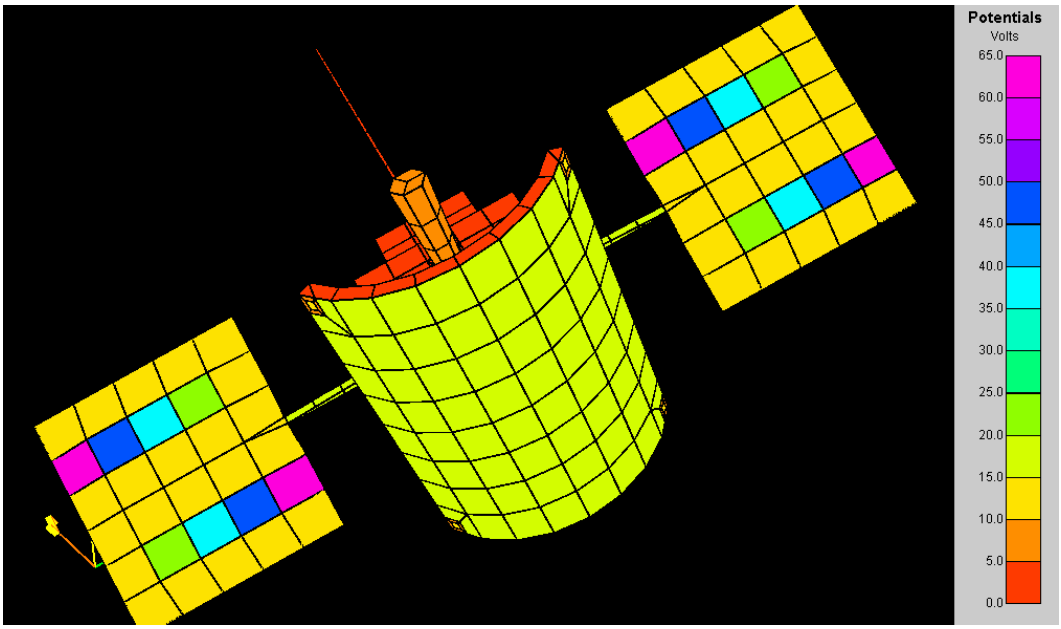


Figure 2. *Nascap-2k* Model of the MESSENGER Spacecraft, Showing Biased Solar Array Surfaces.

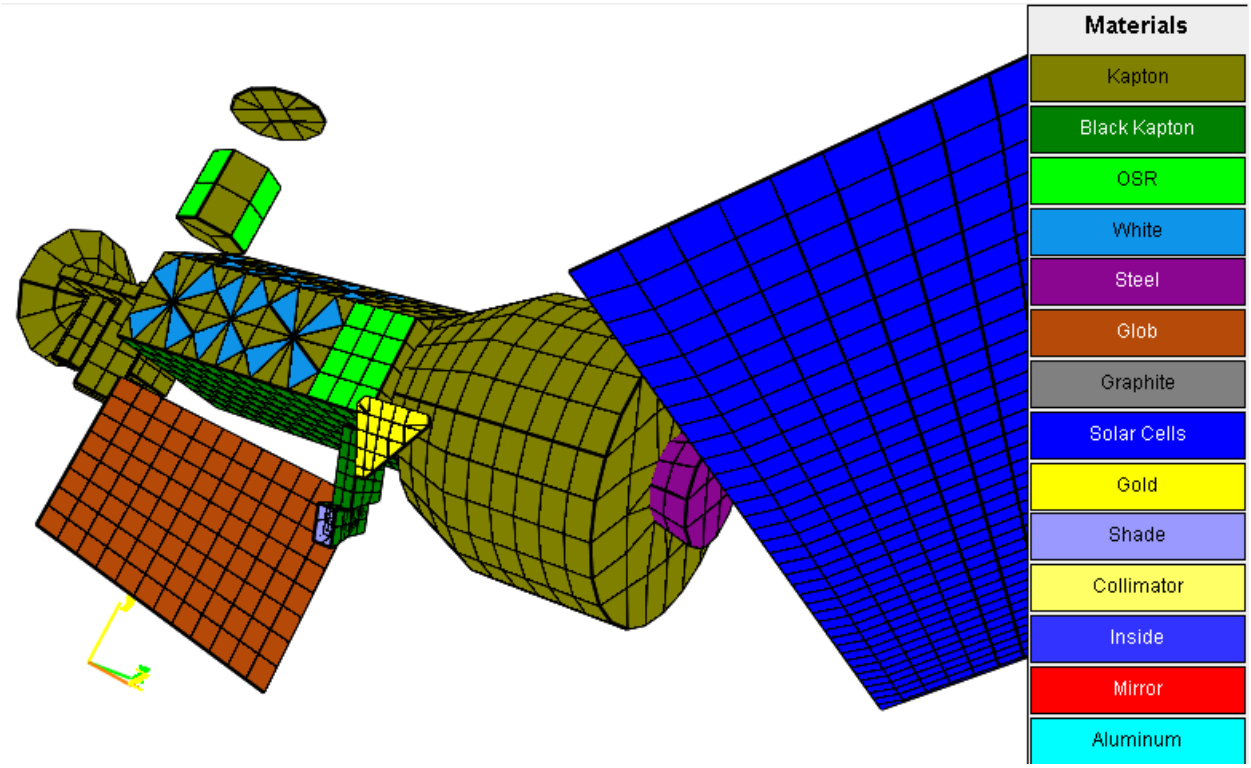


Figure 3. DMSP Spacecraft Model Constructed in *Object Toolkit*.

Object Toolkit writes an eXtensible Markup Language (XML) file that contains the finite element representation of the object, the recipe for re-creating/reassembling the object, and material definitions.

Approved for public release; distribution is unlimited.

How to Use This Manual

This manual is divided into five major sections. Part I is this Introduction.

Part II describes how to use the various capabilities of *Object Toolkit.* Within Part II are sections describing the creation and modification of Components and Assemblies (Section 3) and General Guidance on the use of *Object Toolkit* (Section 4). These provide basic information that all *Object Toolkit* users should be familiar with. Later sections within Part II are ordered by the probable frequency of use by a typical user. These sections can be read or treated as reference.

Part III provides information on all of the menu items in the *Object Toolkit* user interface.

Part IV provides several detailed examples that illustrate how to build objects and specialized components.

The Glossary defines specialized terms used in this manual. The appendices provide background material on file format and customization.

The most effective way to use this manual depends on the user's particular learning style and experience with this and other finite element modelers. One approach that works for many is to go directly to the Examples in Part IV and follow them step by step, using other sections of the manual as a reference for definition of terms and commands. Another approach is to read the first several chapters of Part II and then make a simple object using the methods described in those chapters.

Conventions Used in This Manual

For purposes of clarity, particular terms and text formatting styles are used in this manual.

The term "object" is used to denote the entire geometrical entity the user is creating. It may be a single spacecraft, more than one spacecraft, an instrument, or a subsystem. The object the user creates need have nothing to do with space, though the only tools presently using an *Object Toolkit*-generated object are space environment effects codes.

A "component" is a collection of nodes and surface elements that can be treated as a unit (e.g., moved, rotated). An object is composed of one or more components. Part II includes a more detailed discussion of these terms and others. Definitions can also be found in the Glossary.

Table 1 identifies the formatting conventions used in this manual. Italic font is also used for emphasis.

Table 1. Formatting Conventions.

Text Style	Meaning
Italics	Names of software, variable portion of filenames and input
Boldface	Names of menus and dialog boxes
"Quotation Marks"	Text on interface, including menu items, text box labels, etc.
Lucida Sans Font	Filenames, folder names, web addresses
`Monotype font`	Large quantities of text that represent computer output

Approved for public release; distribution is unlimited.

2 Installation

Object Toolkit is automatically installed with *EPIC* and with *Nascap-2k*. See the *EPIC* and *Nascap-2k* manuals for instructions. Most of the difficulties that users have encountered with *Object Toolkit* are related to the Java3D needed to display the three-dimensional image.

Approved for public release; distribution is unlimited.

II USING *OBJECT TOOLKIT*

3 Components and Assemblies

An *Object Toolkit* object is composed of one or more *components*. A component is a set of nodes and surface elements that can be acted upon as a unit (e.g., moved, rotated). A component can be one of five *standard components*, an *assembly*, or a *Primitive*. *Object Toolkit* has five *standard components*: Box, Panel, Boom, Dish, and Cylinder. Each standard component has a set of user-editable properties that specify its size, shape, gridding, and surface materials.

An *assembly* is a combination of two or more components that have been consolidated. Combining components into an assembly is akin to "grouping" in a drawing or presentation program, although it is not possible to "ungroup" an assembly and the editing capability within an assembly is limited. The assemblies form a tree-structured hierarchy, with each assembly comprised of components at the next lower level.

A *Primitive* is a component described by its nodes and elements; a Primitive can be scaled along each of its local coordinate axes. Any component can be converted into a Primitive, giving the ability to easily scale it, but losing the ability to edit the standard components from which it may have been assembled.

4 General Guidance

There are a number of practices that simplify building objects with *Object Toolkit*. The two most important are to (1) save often and (2) build the object in pieces and then assemble.

Constructing any object beyond the simplest involves numerous operations. As the object becomes more complex, the possibility of error and even of unexpected code termination increases. Therefore, it is *highly recommended* that the user save often and use different filenames when doing so. This allows the user to back up to a simpler object and start over if problems occur.

When making an object with a moving part, such as a solar array, it is best to add the moving part last to facilitate making several models to represent the various geometries.

Converting an object to a Primitive increases execution speed and reduces the likelihood of code failures. However, because a Primitive does not contain the instructions used to build it, it is good practice to save the object to a separate file immediately before converting it to a Primitive. Building parts of the object, saving each part to a separate file, and then assembling them into a single object also minimizes difficulties. Converting each part to a Primitive before assembling the total object is even more effective.

If a situation arises in which a complex component has the wrong size, shape, or meshing, it is often quicker and easier to rebuild that component from scratch than to try to modify it.

Once a complex component has been created, it can be saved to a Primitive and later scaled and reused.

Most operations can be undone using the "Undo" on the **Edit** menu.

Approved for public release; distribution is unlimited.

When the object becomes complex, it can become difficult to pick nodes, edges, and elements. When this occurs, "Consolidate Components into Assembly" on the **Components** menu can be used to consolidate all the components (including existing assemblies) into a single assembly resulting in a new "Top Assembly." This command has been observed occasionally to cause apparent loss of certain instructions. In particular, nodes moved using the "Edit Node" command have been seen to return to their original positions. This can often be corrected by selecting the component in which the error appears and selecting "Rebuild Mesh" from the **Mesh** menu.

It can be difficult to find a view in which it is possible to see all the nodes, edges, or elements that need to be picked for a specific operation. The direct rotation buttons can be used to reorient the view while the cursor remains in select mode. Also, the "Hide" command from the **View** menu can be used to temporarily remove from the display components that block the line of sight to the needed mesh entities.

Free edges are indicated by the presence of red lines. A free edge indicates that there is a hole in the object or that the elements are incompatible. A compatible mesh is a set of elements in which each edge has exactly one element on its right and exactly one on its left. Some codes that rely on *Object Toolkit* for object definition prohibit incompatible elements (e.g., *Nascap-2k*). Other codes allow them (e.g., *EPIC*). Incompatibilities may be resolved by mesh editing. Section 11 provides examples of resolving incompatibilities.

5 Cursor Tools and Direct Movement and Rotation

The buttons along the top of the screen control cursor mode and object view. The buttons on the top right move, zoom, or rotate the view when clicked. The buttons on the top left are cursor tools that place the cursor in a specified mode when clicked. Table 2 lists the functionality of each button. The first four cursor tool buttons on the left put the cursor in select mode. In select mode, a mouse click selects the component, element, edge, or node nearest the cursor, depending on which of the four cursor tools is active. Multiple entities of the same kind can be selected by holding down the CTRL key while clicking. The next two cursor tool buttons (disabled by default) place the cursor in a mode in which a single component can be moved or rotated using the mouse. The last two cursor tool buttons place the cursor in translate and rotate mode, respectively. In these modes, movement of the mouse while the left mouse button is pressed changes the view of the object. Note the difference between these two cursor tool buttons and the two (disabled by default) just to the left. The two on the right only move or rotate the view. The two on the left actually move or rotate a component.

Approved for public release; distribution is unlimited.

Table 2. Cursor Tool and Direct Movement and Rotation Buttons.

Buttons	Function
	Put the cursor in a mode to select specific components, elements, edges, or nodes.
	Put the cursor in a mode to modify the object by translating or rotating an individual component. Disabled by default. These tools are enabled by going to the **Component** menu and choosing "Enable Tools."
	Put the cursor in a mode to rotate or translate the view of the object.
	Translate the view left, right, up, or down by a fixed amount.
	Scale the view by zooming in or out.
	Rotate the view about the horizontal or vertical axis by a fixed amount.
	Rotate the view in-plane by a fixed amount.

6 Standard Components

Object Toolkit has five standard components, which are shown in Figure 4. When *Object Toolkit* is tailored for specific applications, additional special components may be available. Their use is discussed in Section 15. The Box, Panel, and Boom are shown in the top row of the figure. The Dish and the Cylinder are shown along the bottom of the figure. The user specifies the size, the surface materials, the conductor numbers, some aspects of the shape, and the number of mesh elements in each direction. For simplicity, each standard component has only a limited number of user-specified parameters. Each component has its own local coordinate system, which is *not* the

Approved for public release; distribution is unlimited.

object coordinate system. The local coordinate system is used when modifying a component as described below in Section 13.2. Generally, the origin is at the center of the component.

Standard components are created by choosing "Add New…" on the **Component** menu. Any of the parameters of a standard component can be changed by selecting the component and choosing "Edit Component" on the **Edit** menu.

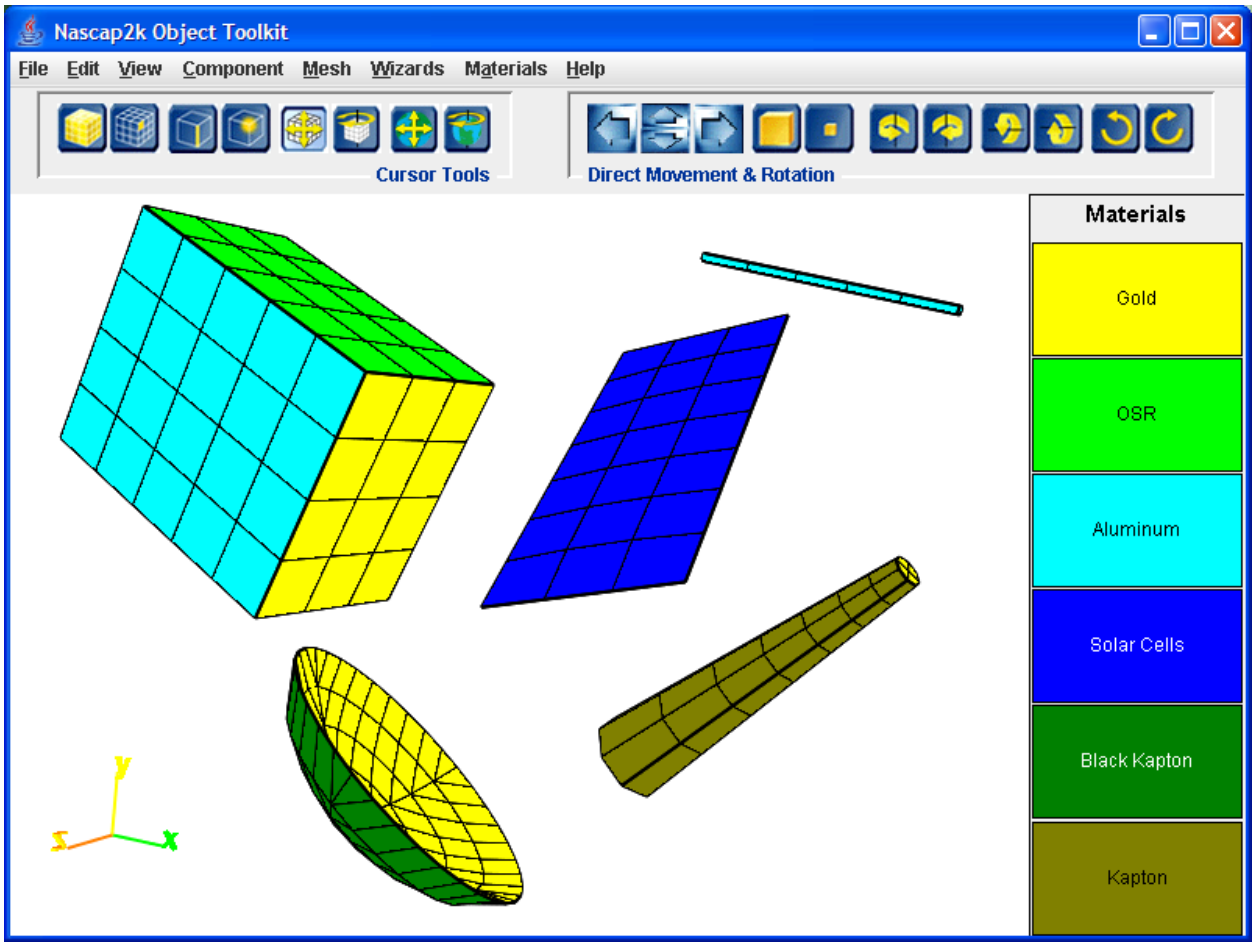


Figure 4. *Object Toolkit* Screen Showing the Five Standard Components.

6.1 Box

A Box (commonly called a "hexahedron" or "brick" by the finite element community) has six flat sides. The number of elements in each of the three directions is user-specified. The surface material and conductor number of each side is separately specified. A Box can be rectangular (and therefore defined by its extent in the three directions) or nonrectangular, defined by its eight corners. When a new Box is added, a dialog window appears for specifying dimensions, gridding, and materials, as shown in Figure 5 and Figure 6.

Approved for public release; distribution is unlimited.

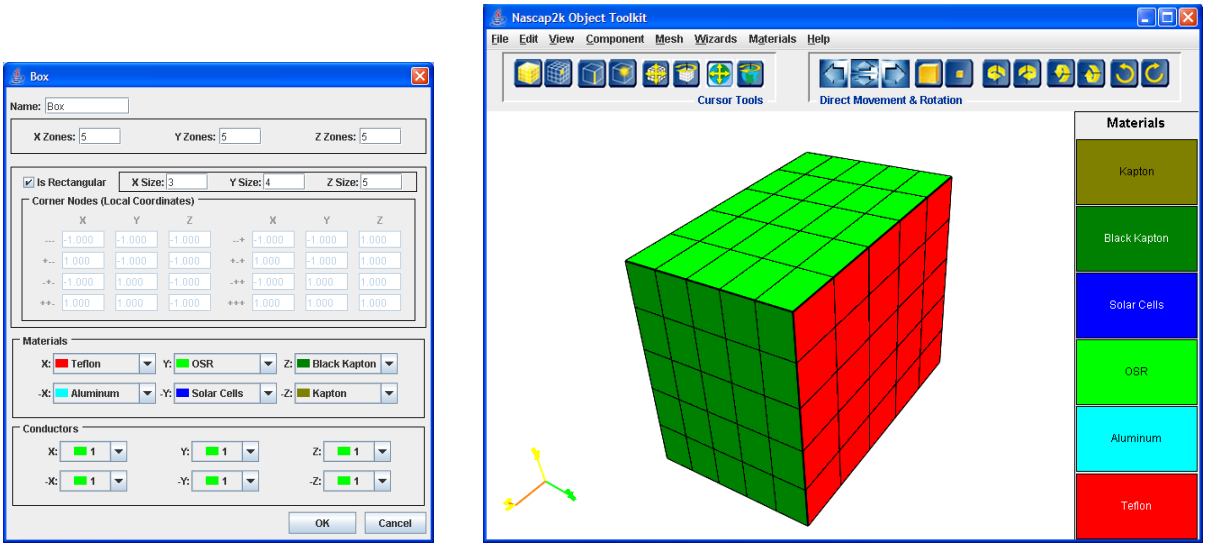


Figure 5. Definition of a 3 m by 4 m by 5 m Rectangular Box with a Different Material on Each Side.

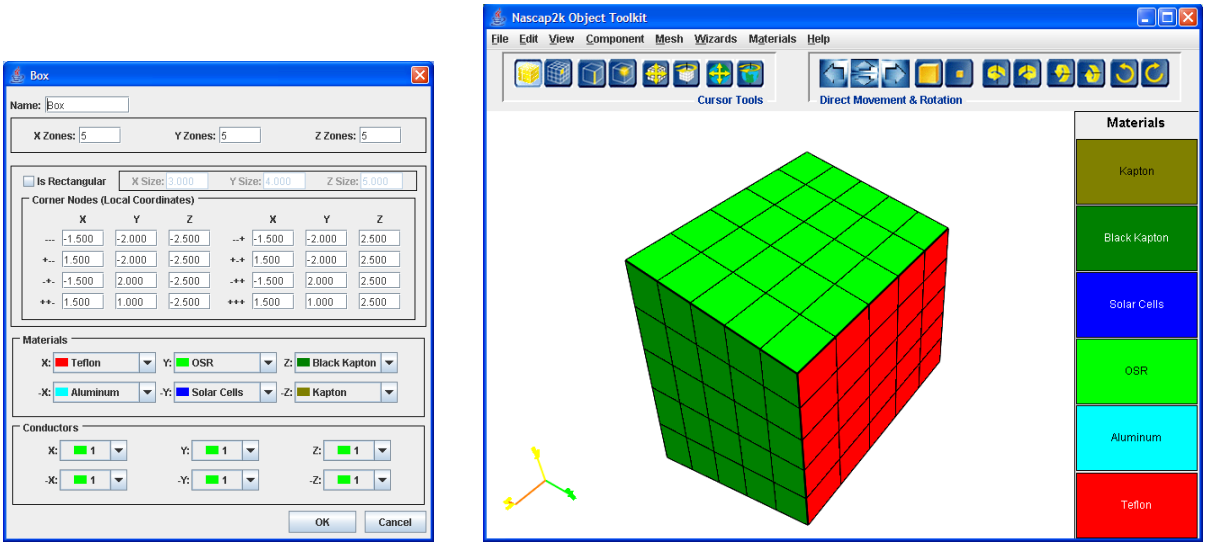


Figure 6. Definition of a Nonrectangular Box.

6.2 Panel

A Panel has only two almost-flat sides. The two sides are separated in the middle by a specified thickness; one side points toward +Z and the other toward –Z in the component local coordinate system. For most geometries, it is better to use a box that is one element thick instead of a panel. When a new Panel is added, a dialog window appears for specifying dimensions, gridding, and materials, as shown in Figure 7 and Figure 8. The nodes on the outer edges of the Panel are shared by the two sides at Z=0. Thus the normals of the elements along the edges point slightly off the Z axis, which can cause problems in some applications. The number of elements in each of the two directions is user-specified. The minimum size of a Panel is two elements by two

Approved for public release; distribution is unlimited.

elements. The surface material and conductor number of each side is separately specified. A Panel can be rectangular or defined by its four corners.

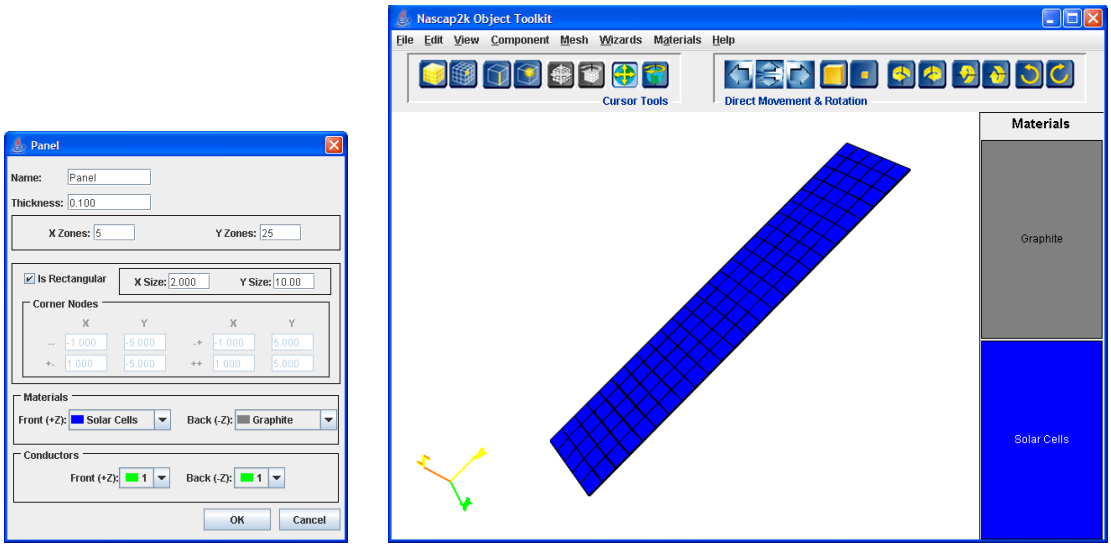


Figure 7. Definition of Rectangular Panel.

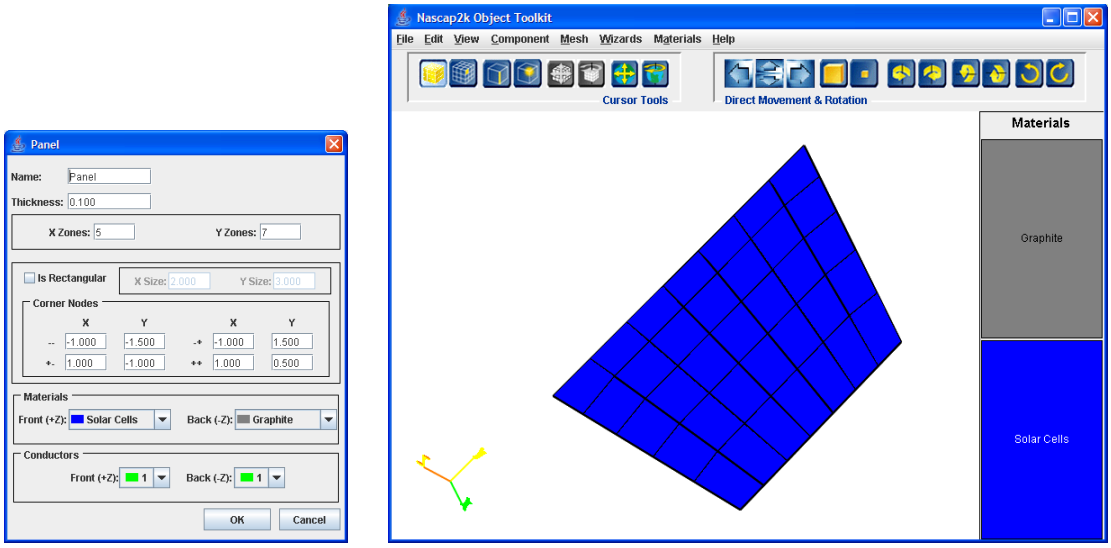


Figure 8. Definition of Nonrectangular Panel.

6.3 Boom

The Boom is a variation of the Box. It is intended to be used for long, skinny shapes and can have a twist. A boom with a twist can be used to add a component (such as a solar array) onto a boom and orient it to a non-orthogonal angle. All the surfaces of a Boom are the same material. A Boom can be rectangular or defined by its eight corners. When a new Boom is added, a dialog window appears for specifying dimensions, gridding, and materials, as shown in Figure 9. The twist is applied to the high-index end of the Boom after the corners are defined. The high-index end is the one with the highest Z value when originally created.

Approved for public release; distribution is unlimited.

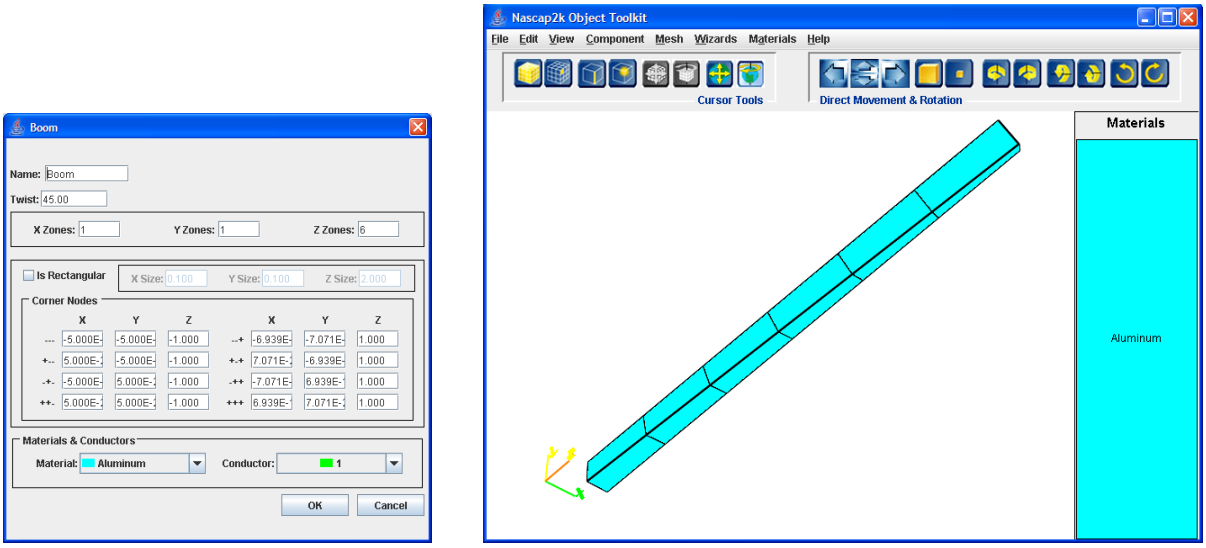


Figure 9. Definition of a Twisted Boom.

6.4 Cylinder

A Cylinder that is *not* tapered is a right regular prism. A tapered Cylinder is a frustum of a right regular pyramid. When a new Cylinder is added, a dialog window appears for specifying dimensions, gridding, and materials, as shown in Figure 10. The user-specified number of "circumferential zones" gives the number of sides of the regular polygons that form the upper and lower bases. The number of sides must be six or a multiple of four that is equal to or greater than eight. Other numbers are rounded upward. The surface materials of the top and bottom are specified separately from that of the sides.

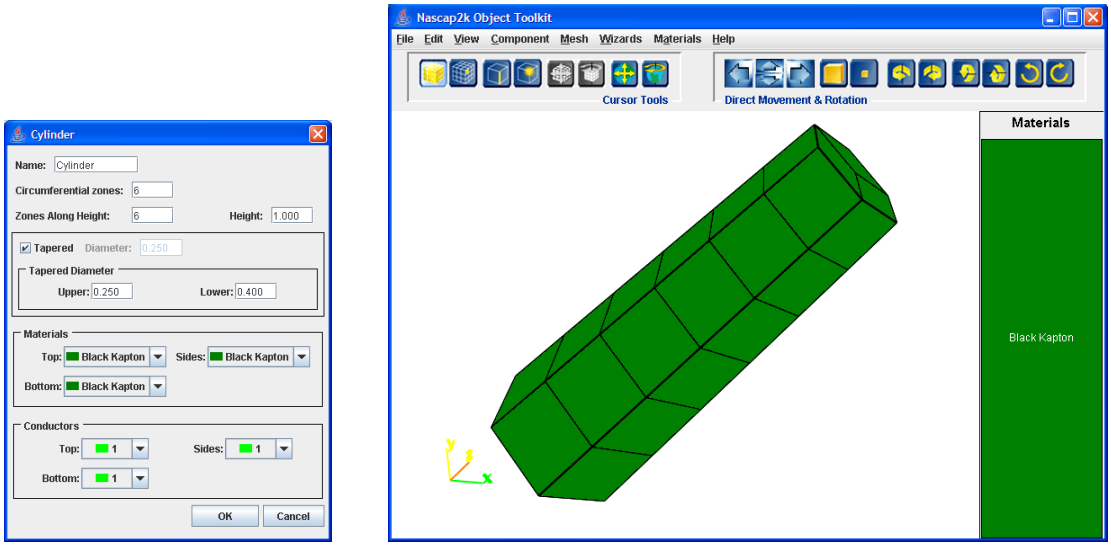


Figure 10. Definition of a Tapered Cylinder.

6.5 Dish

A Dish is an approximation of a circular paraboloid. It is intended primarily for modeling dish antennas. All the nodes are on two paraboloids, which form the concave and convex sides of the

Approved for public release; distribution is unlimited.

Dish. When a new Dish is added, a dialog window appears for specifying dimensions, gridding, and materials, as shown in Figure 11. The paraboloids intersect at a circle of the specified diameter. This circle circumscribes the top surface. The depth is the distance between the center of this circle and the vertex of the inner paraboloid. The distance between the center vertices of the two paraboloids is fixed. The user-specified "X-Y Zones" gives the number of elements across the diameter. The maximum number allowed is 24. The front and back surface materials are separately specified.

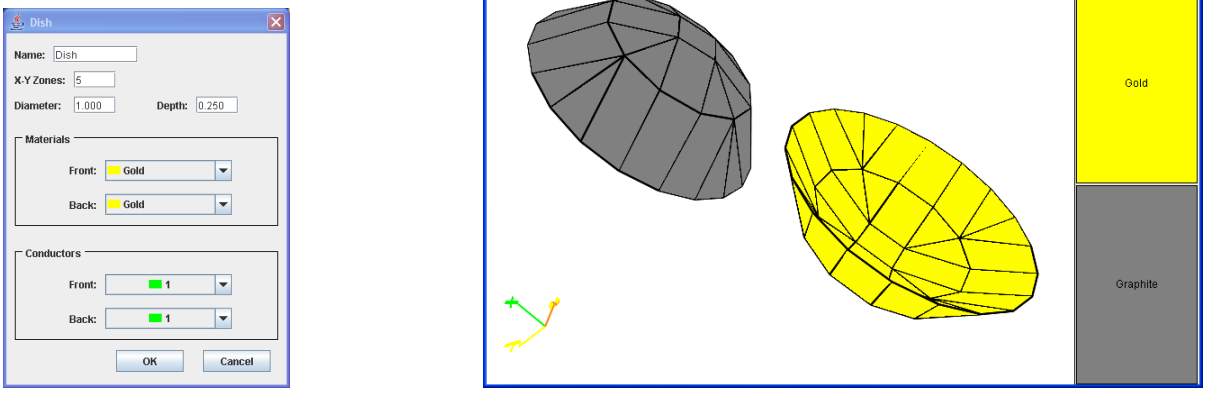


Figure 11. Definition of a Dish. The Screen on the Right Shows Two Dishes Viewed from Different Directions.

7 Aligning and Positioning Components

There are a number of different ways to align and position components. Four items on the **Component** menu ("Node Relative Move," "Align Edge," "Center at Origin," and "Place") and two cursor tools can be used to move or rotate components. (The cursor tools provide imprecise positioning, and should be used only sparingly.) "Node Relative Move" translates (moves) a component so that the node selected (using the select node cursor tool) is at the user-specified location (or the location of a second selected node). "Align Edge" orients (rotates) a component so that the edge selected is parallel to the user-specified direction (or to a second selected edge). "Place" translates the component so that the center of its bounding box is at the user-specified location. "Center at Origin" places the center of the component's bounding box at position (0, 0, 0). Figure 12 through Figure 15 illustrate successive use of the "Place," "Node Relative Move," and "Align Edge" commands.

Approved for public release; distribution is unlimited.

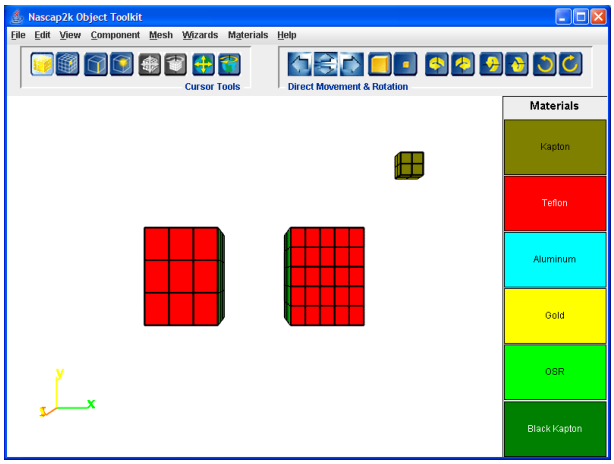


Figure 12. Three Boxes Used to Illustrate Alignment and Positioning Components. The Left-hand Box Is Centered at the Origin. The Right-hand Box Is Centered at (6, 0, 0). The Small Upper Box Is at (10, 5, 0). The View Is Shifted Slightly to the Right and Perspective Is Turned on.

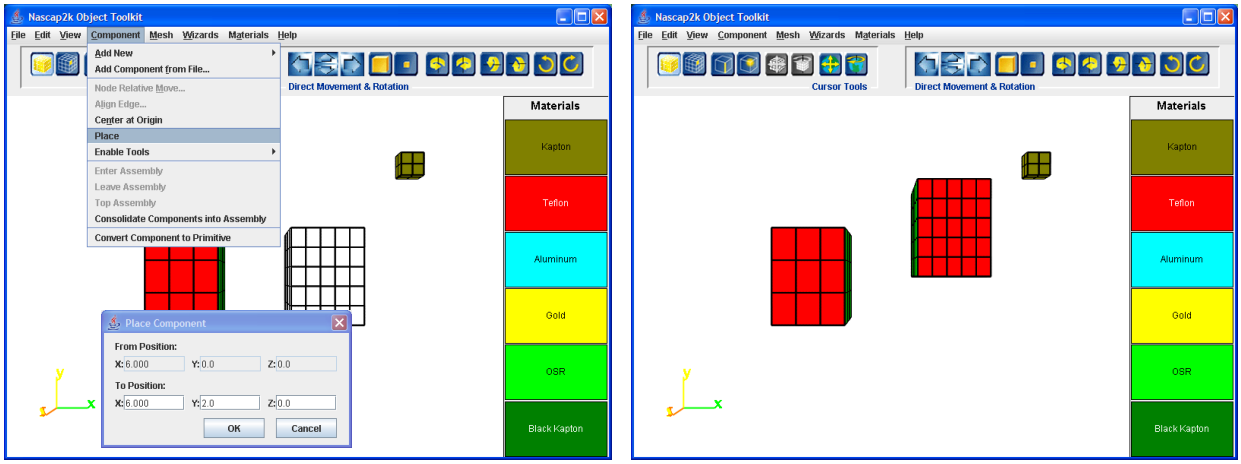


Figure 13. Using “Place” on the Component Menu, Which Operates on the Selected Component, to Move the Middle Box Vertically So That Its Center Is at (6, 2, 0).

Approved for public release; distribution is unlimited.

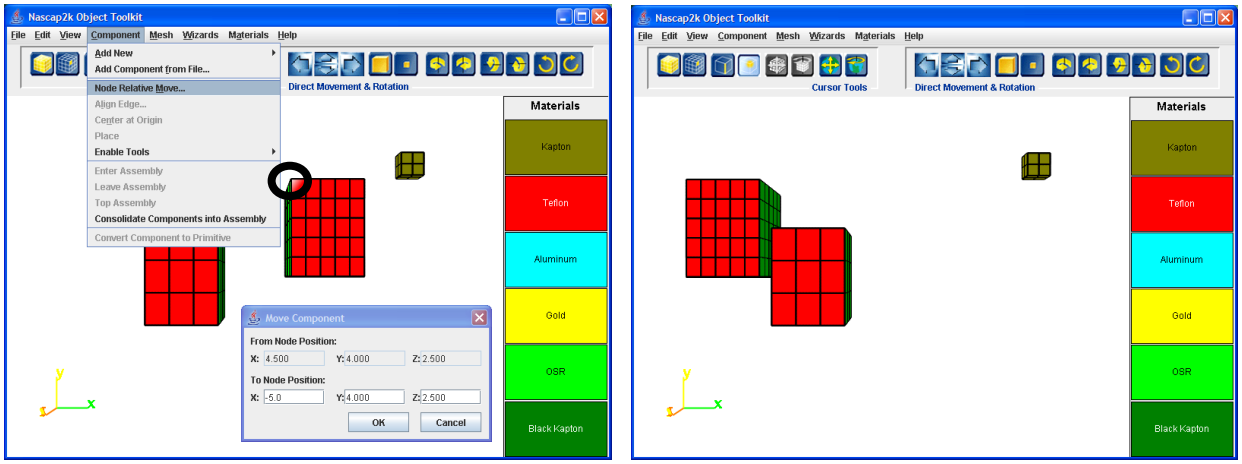


Figure 14. Using “Node Relative Move” on the Component Menu, Which Operates on the Component Containing the Selected Node, to Move the Middle Box to the Left Side of the Other Box with the Upper Left Node at (–5, 4, 2.5). Selected Node Is Circled.

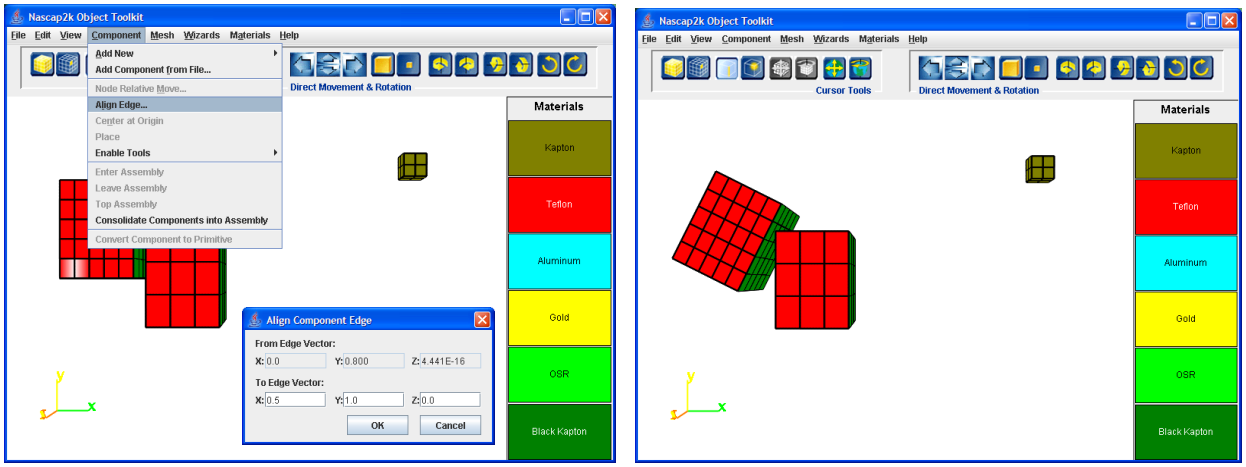


Figure 15. Using “Align Edge” on the Component Menu, Which Operates on the Component Containing the Selected Edge, to Rotate the Left-hand Box (Which Was on the Right in Figure 13). The Rotation to Align the Edge in the (0.5, 1.0, 0.0) Direction Takes Place in the Component Local Coordinate System.

Approved for public release; distribution is unlimited.

Figure 16 shows the results of translating and then rotating one component using the cursor tools. These tools are generally disabled because they can be easily confused with the object rotation and translation tools, resulting in unwanted changes to the assembly. They may be enabled from the **Component** menu and used for approximate placement of components.

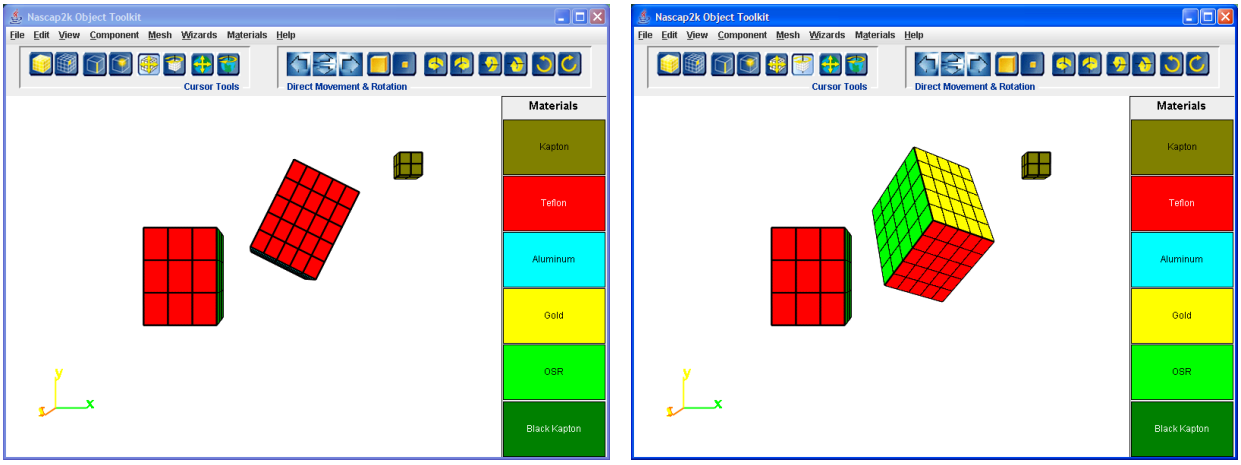


Figure 16. The Results of Translating and Then Rotating the Box Using the Cursor Tools.

8 Grouping Components

The "Consolidate Components into Assembly" item on the **Component** menu is used to group all the components (including existing assemblies) into a single component (an assembly). This command consolidates the entire object, excluding any special components, and builds a mesh for that assembly. Once two or more components are grouped into an assembly, the assembly functions as a single component and all the component commands operate on that assembly component. Specifically, mesh editing commands are applied to the assembly's mesh. Figure 17 shows an example of using "Align Edge" on an assembly. This figure can be compared with Figure 15 in which the same edge was selected and "Align Edge" was chosen when the components have *not* been combined into an assembly. Individual components that have been combined into an assembly can be separately edited by selecting the assembly and choosing "Enter Assembly." Section 14 discusses this capability.

Approved for public release; distribution is unlimited.

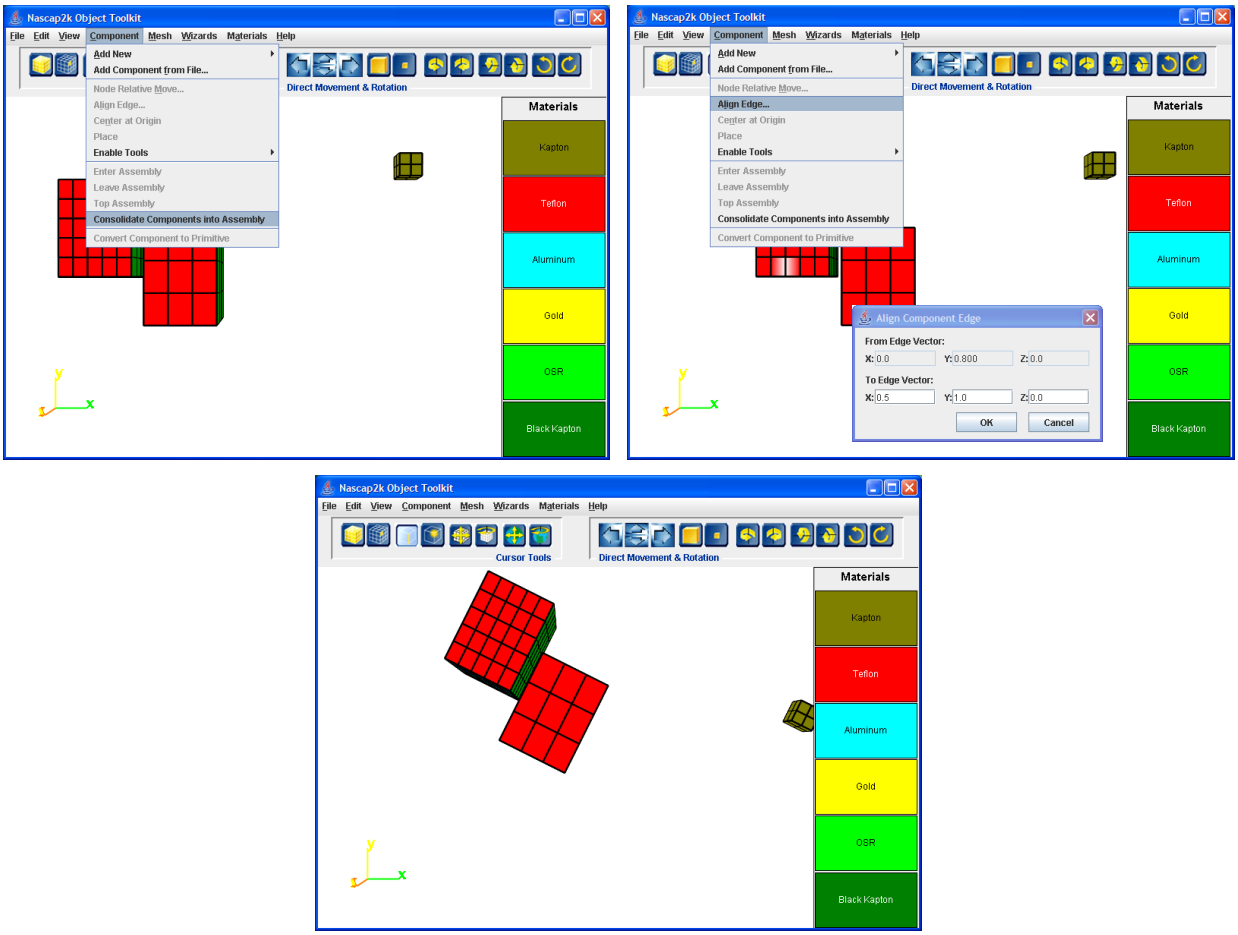


Figure 17. Alignment of an Assembly (Multiple Components That Have Been Combined into a Single Component) So That the Edge Is Oriented in the (0.5, 1.0, 0.0) Direction.

9 Moving, Aligning, and Grouping Components Using Wizards

The items on the **Wizard** menu automate the process of moving, aligning, and grouping components for common operations. In each case, one component is moved to be next to another component and oriented appropriately with respect to the nonmoving component. The two components are then grouped into an assembly, and elements and nodes are moved, added, and deleted to form a compatible meshing. Section 11.2 discusses compatible meshing. The four Move Wizards available in *Object Toolkit* for EPIC or MEM skip this last step. These Wizards are discussed further in Section 15.10. It is important to note the difference between the component being moved and the one that remains fixed. In all cases, the first item you choose in a wizard—element, edge, or surface—is on the component to remain fixed in the object reference frame.

Approved for public release; distribution is unlimited.

The “Element to Element Wizard,” shown in Figure 18, joins a quadrilateral element to the center of a larger quadrilateral element. A common use for this wizard is to join the end of a Boom to the side of a Box. The user first selects an element of the fixed component (Figure 18b) and then an element of the moving component (Figure 18c). The second component chosen is automatically moved and reoriented so that the centers of the two elements are coincident. A new component, called a Weld, is created to compatibly mate elements of the original components. The user is provided with an opportunity to specify any necessary reorientation (Figure 18d). The original two components and the Weld are combined to form a single assembly.

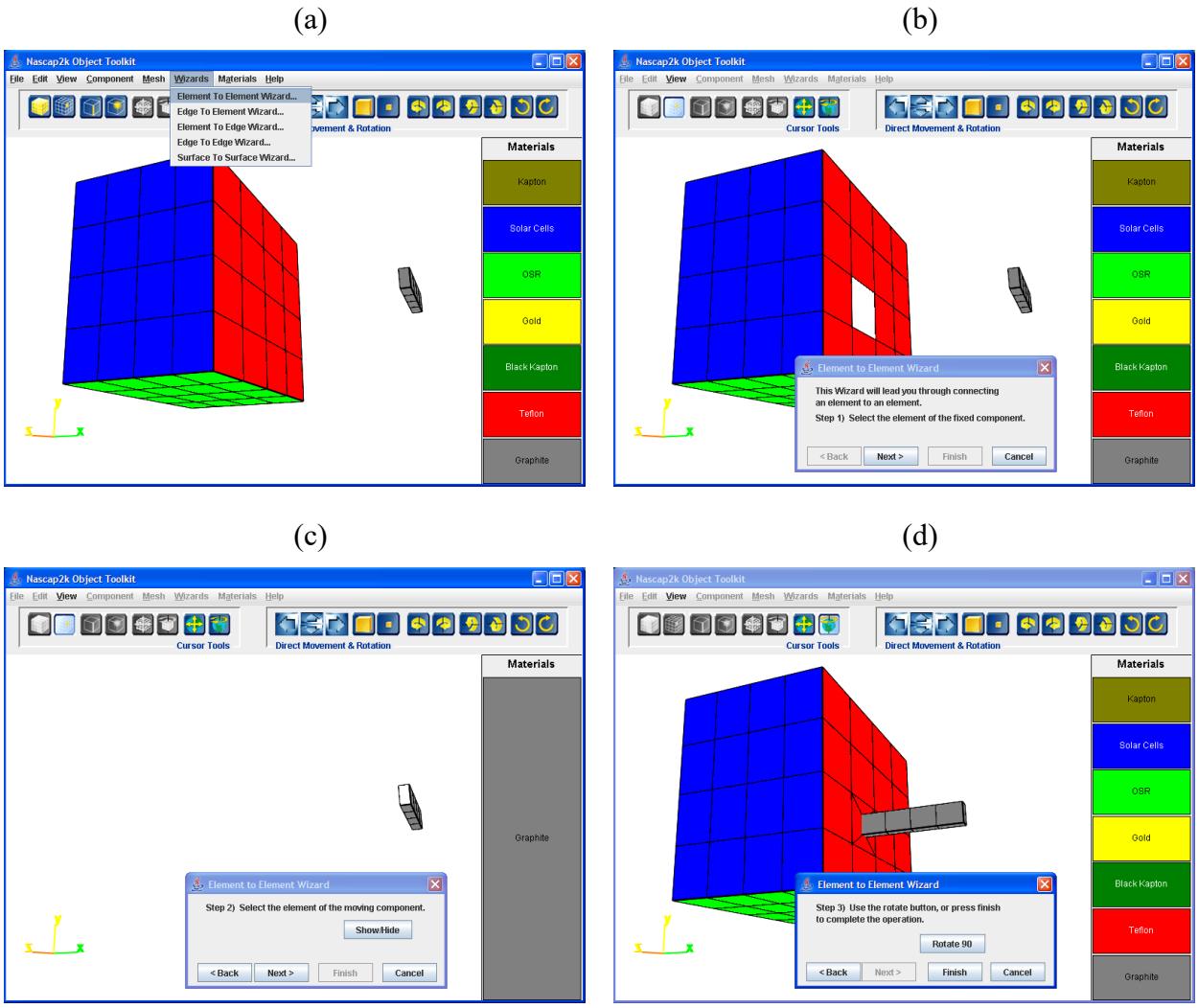


Figure 18. Moving, Aligning, and Grouping Two Components Using the “Element to Element Wizard.” The Four Trapezoidal Elements Belong to the Weld. In the Third Frame, the Fixed Component Is Automatically Hidden.

Approved for public release; distribution is unlimited.

The “Element to Edge Wizard” joins two components by placing an element of one component next to an edge of another, fixed component. The “Edge to Element Wizard,” shown in Figure 19, joins two components by placing an edge of one component next to an element of another, fixed component. The user first selects an element of the fixed component (Figure 19b) and then an edge of the moving component (Figure 19c). The second component chosen is automatically moved and reoriented so that the centers of the element and the edge are coincident. The user specifies which of the four possible orientations is desired by pushing the “Rotate 90 degrees” button until the desired orientation is achieved (Figure 19d and Figure 19e). The two components are combined into a single assembly. The selected element is then deleted, and the edge is “opened” by replacing each of the two elements that form the selected edge with a new element. Each of the new elements has one edge coincident with an edge of the element of the fixed component. This is necessary to avoid the possibility of a single point in space having two different potentials. Triangles are added to form a compatible meshing, as shown in Figure 19f.

Approved for public release; distribution is unlimited.

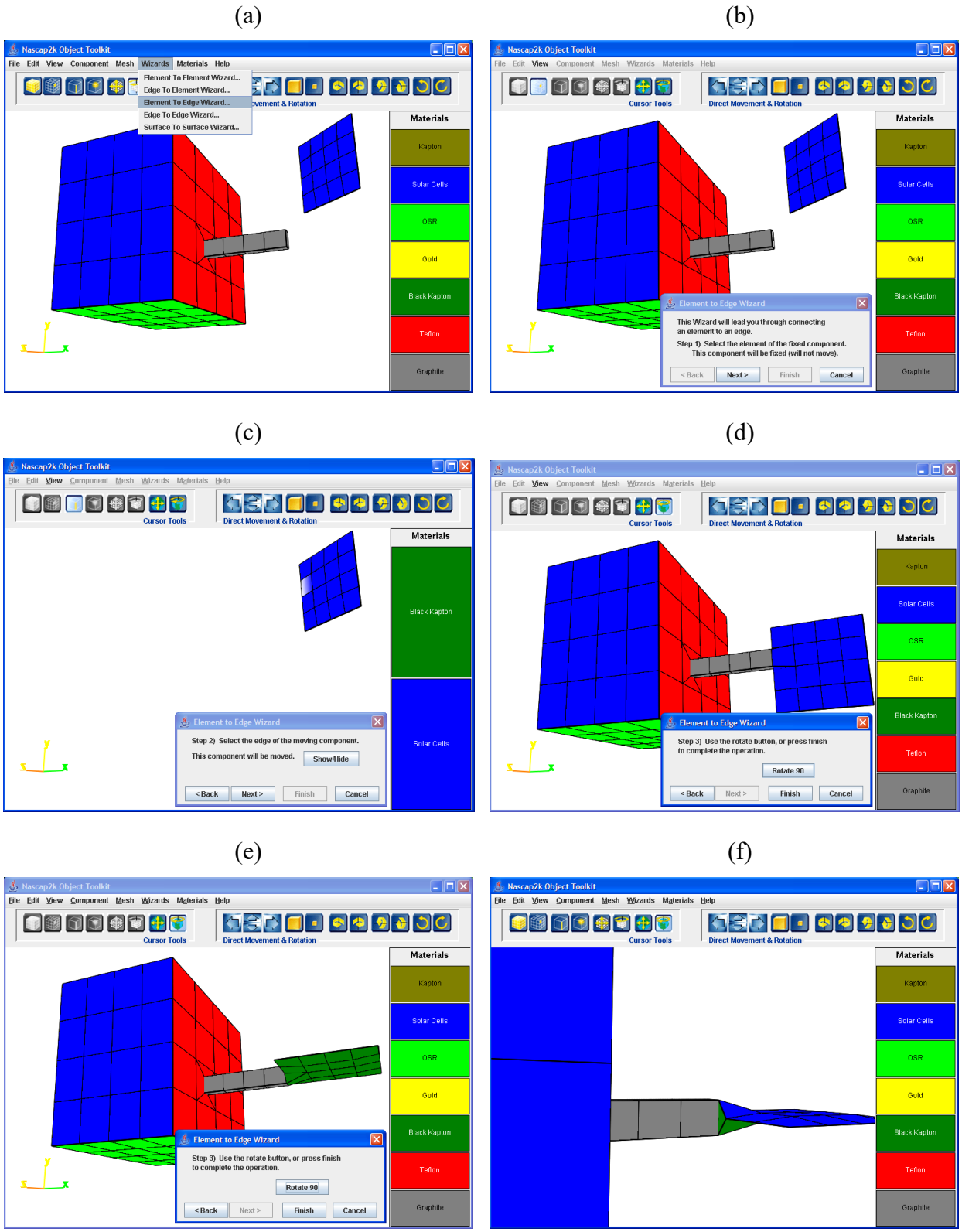


Figure 19. Moving, Aligning, and Grouping Two Components Using the "Element to Edge Wizard." In the Third Frame, the Fixed Component Is Automatically Hidden.

Approved for public release; distribution is unlimited.

The “Edge to Edge Wizard,” shown in Figure 20, joins two components by placing an edge of one component next to an edge of another. The user first selects an edge of the fixed component (Figure 20a) and then an edge of the moving component (Figure 20b). The second component chosen is automatically moved and reoriented so that the centers of the two edges are coincident. One node of the longer edge is then automatically moved, so that the two edges are of equal length. *Under some circumstances, this can lead to an unintended distortion of the object*. The user is provided with an opportunity to specify any necessary reorientation (Figure 20c and Figure 20d). The two components are combined into a single assembly. The edge is opened by the addition of two new nodes a small distance apart and the division of each of the four adjoining elements into a quadrilateral and a triangle using the new nodes. Figure 20e and f show the opening up of the edge and the additional triangular elements. This is necessary to avoid the possibility of a single point in space having two different potentials.

Approved for public release; distribution is unlimited.

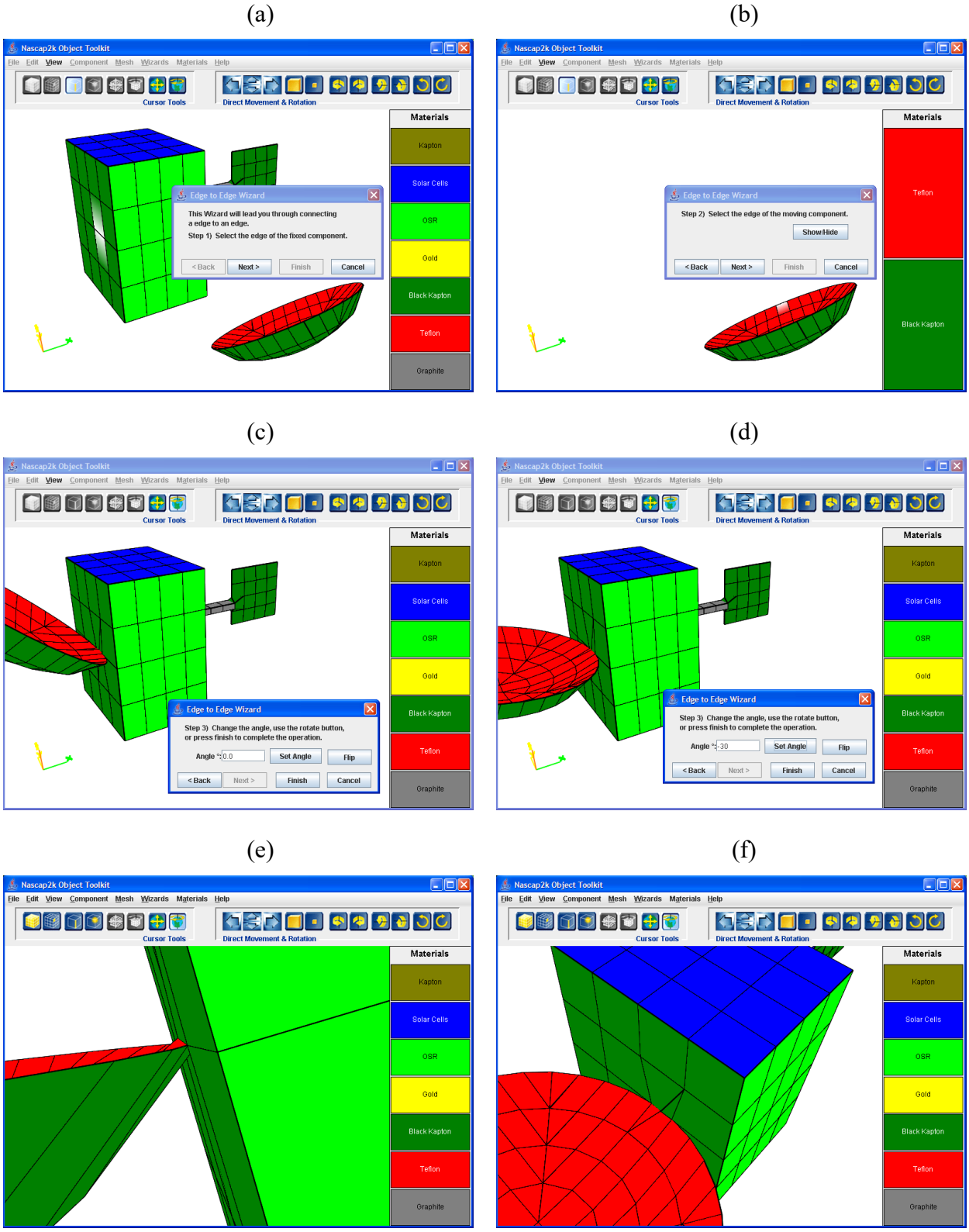


Figure 20. Moving, Aligning, and Grouping Two Components Using the "Edge to Edge Wizard." In the Second Frame, the Fixed Component Is Automatically Hidden.

Approved for public release; distribution is unlimited.

The “Surface to Surface Wizard,” shown in Figure 21, joins two components by moving one component so that a surface of the first is coplanar with a surface of the second component. (This is a difficult operation, so the user should not expect this Wizard to work perfectly.) The user first selects an element of the first component (Figure 21a) and then an element of the second component (Figure 21b). The second component chosen is automatically moved and reoriented so that the centers of the two elements are coincident. A pop-up dialog box then appears in which the user can specify the location and rotation of the moving component with respect to the fixed component (Figure 21c and Figure 21d). The two components are combined into a single assembly. Overlapping elements in the plane of the shared surface are deleted. Figure 21e shows the resulting object. The free edges are indicated by the presence of red lines along the edges. The wizard may move nodes along the free edges and add new nodes to divide elements adjoining the free edges. These changes sometimes distort the object, so *this wizard should be used with caution*. The addition of new elements to form a continuous meshing is left to the user and discussed in Section 11.2.

Approved for public release; distribution is unlimited.

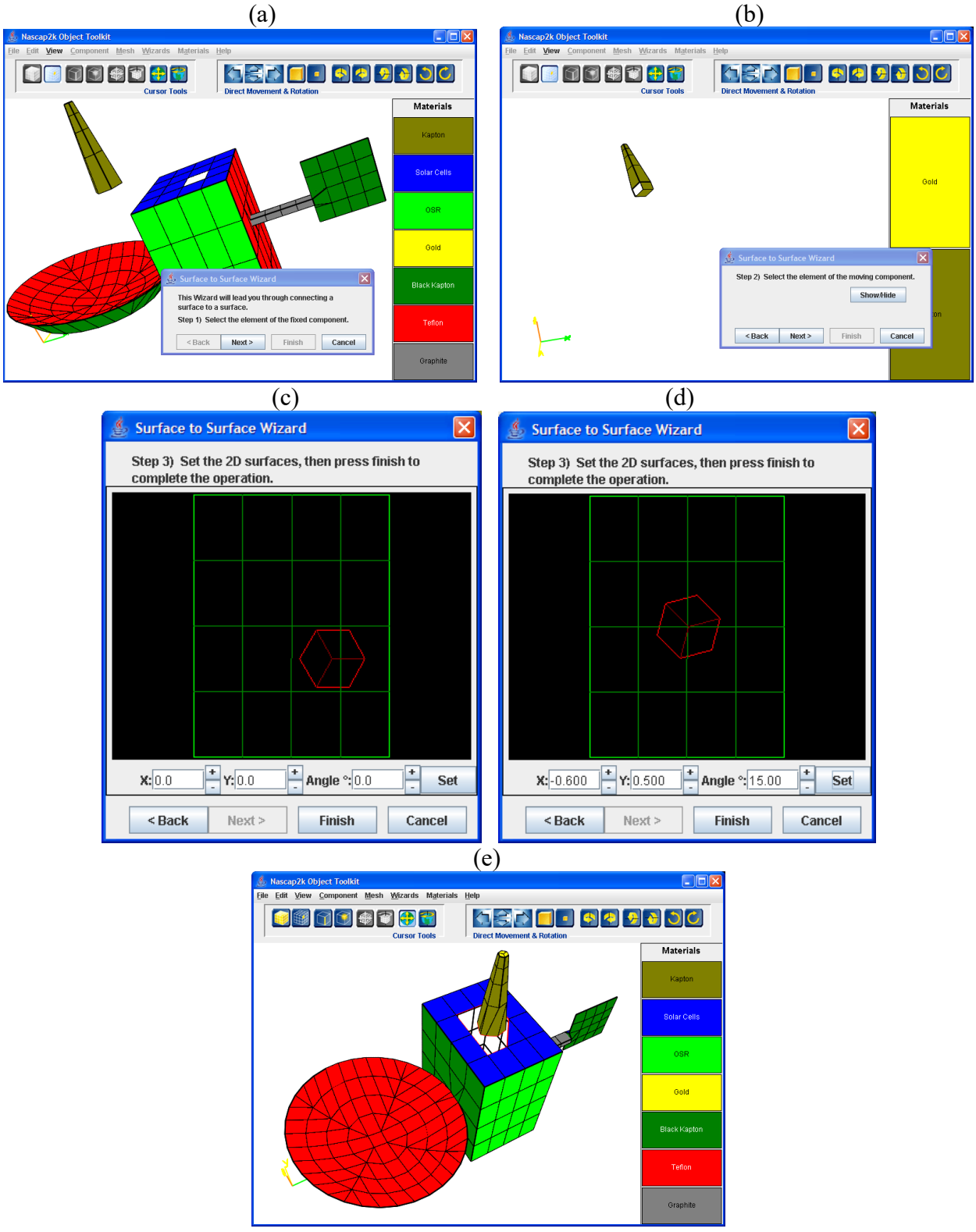


Figure 21. Moving, Aligning, and Grouping Two Components Using the “Surface to Surface Wizard.” In the Second Frame, the Fixed Component Is Automatically Hidden. In the Last Frame, the Outlines of Elements with Normals Facing into the Screen Are Seen.

Approved for public release; distribution is unlimited.

10 Materials

In *Object Toolkit*, a material specification consists of a name, display color, and a set of material properties. The properties are edited by changing values displayed in the **Material** dialog box, accessed by selecting "Edit" on the **Materials** menu. Figure 22 shows this dialog box. Different applications may have different properties. Different applications have different requirements for material names.

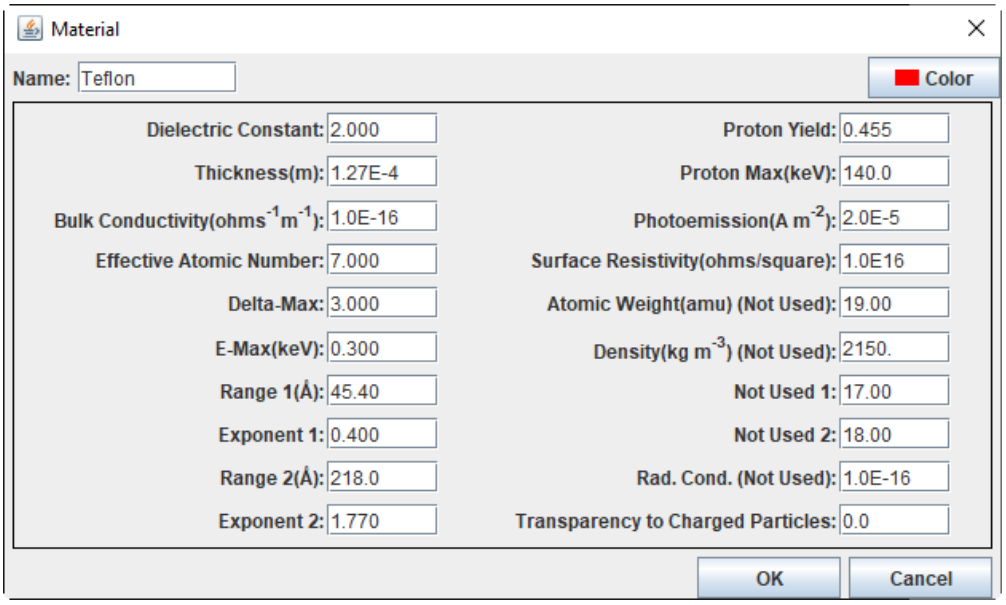


Figure 22. Material Property Definition Dialog Box.

Presently, *Nascap-2k* is the only program that imports material properties from the object definition file. The properties are the same as those used by the *SEE Interactive Spacecraft Charging Handbook,* and material definitions can be imported from the *SEE Handbook*. To import material definitions from the *SEE Handbook* into *Object Toolkit*, first define the materials in the Handbook. Save the Handbook parameters by clicking the "Save" button along the top and note where the file is saved. In *Object Toolkit*, select "Import SEE Handbook Materials File" on the **File** menu and browse to the file written by the *SEE Handbook*. Material property sets for more than a dozen materials of interest have been provided in this format to the SEE program by Utah State University. These are installed with *Nascap-2k*, and placed in the "Materials" folder.

Each application has a default list of materials. New materials can be added to the list using "New…" on the **Materials** menu. Materials can be deleted using "Delete" on the **Materials** menu. When a material is deleted, it is removed from the list of materials the user can edit, delete, or assign to an element, and its properties are not saved. If the deleted material is assigned to any element, the surface material is not changed. Therefore, it is possible for an element to have a material name that does *not* have any material properties associated with it, in which case *Nascap-2k* treats it as the default material for the application. Also, if a material on the default list of materials is deleted, it is re-created with default properties the next time *Object Toolkit* is started.

Approved for public release; distribution is unlimited.

Any material definitions included in an imported *Object Toolkit* file are also imported. Elements imported from most standard element preprocessors are assigned the default material for the application. Elements imported from a *NX I-DEAS* TMG ASCII VUFF file are assigned a material name that consists of the first four letters of the `RDFGROUPNAME` specified in the file. The material properties are set to those of the default material for the application.

10.1 Changing Surface Material Assignments

The surface material assigned to one or more elements can be changed by selecting the elements, going to the **Mesh** menu, selecting "Materials," and choosing the desired material from the fly-out menu that appears. Figure 23 shows how to change the surface material assigned to elements from OSR to Teflon.

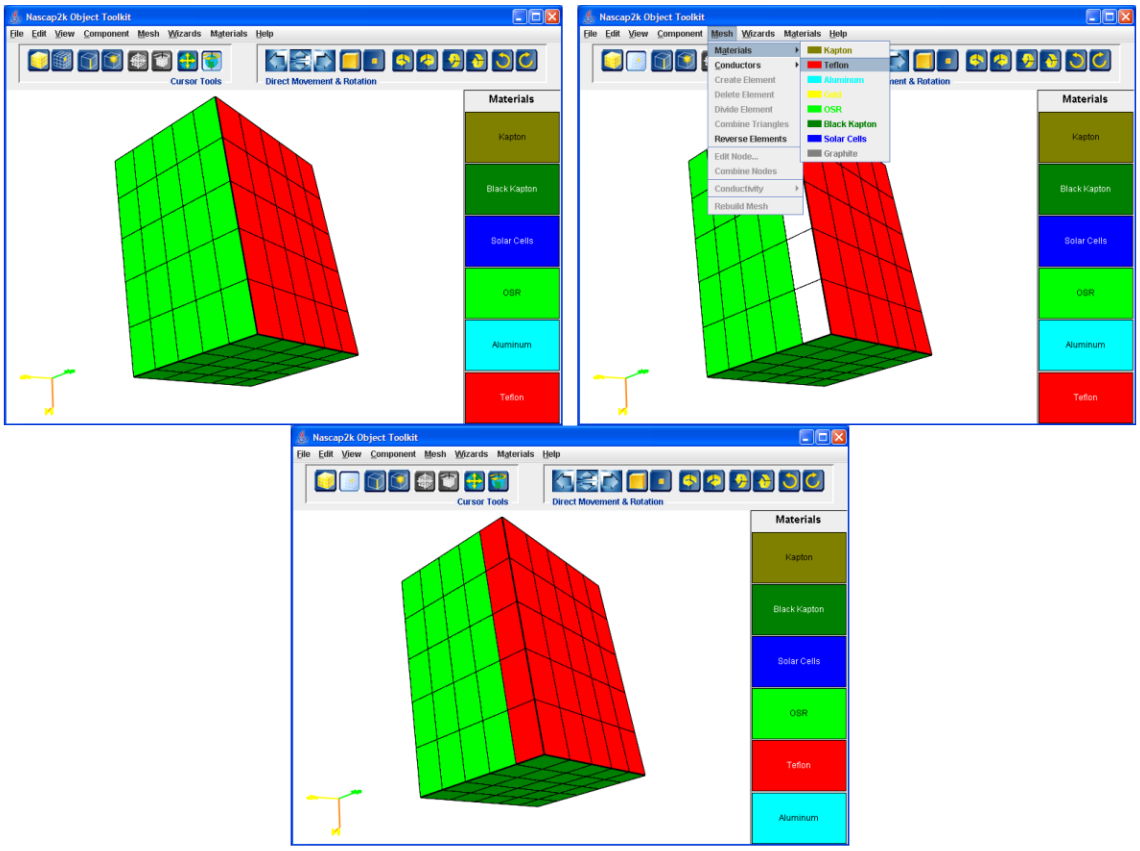


Figure 23. Changing the Surface Materials Using "Materials" on the Mesh Menu.

11 Mesh Editing

To expand beyond the shapes that can be built from the standard components, *Object Toolkit* provides the means for working directly with surface elements. The **Mesh** menu provides these capabilities, which include dividing, deleting, and reversing elements, assigning material names and conductor indices to elements, and creating new elements.

Because mesh editing can go wrong, particularly for complex objects, it is important to save the object before any extensive mesh editing. For mesh editing sessions beyond about 20 operations,

Approved for public release; distribution is unlimited.

it is also a good idea to save the object to a different filename periodically. This process makes it possible to revert to an earlier configuration.

Mesh editing is performed on the components of an assembly, therefore when combining the meshes of two components, it is necessary to consolidate the components into an assembly before beginning mesh editing. When creating a new element, all of the nodes or edges to be combined need to be in the same assembly. The same applies to the combining of triangles and the combining of nodes. If “Create Element”, “Combine Triangles”, or “Combine Nodes” is not available on the **Mesh** menu when expected, consolidating all the components usually resolves the difficulties.

All of the nodes or edges used to create a new element must be in the same component. Any two nodes that are in same component and are within the user specified tolerance (default value of one millimeter (0.001 m)) of each other are automatically combined into a single node. When all the elements to which a particular node belongs are deleted, that node is also deleted and unavailable for the creation of new elements.

It can sometimes be difficult to orient the view so that all the desired edges or nodes can be viewed at once. The direct rotation tools can be helpful in reorienting the view during selection.

11.1 Local Mesh Refinement

Often it is desirable for the elements in one portion of the object to be smaller than in the surrounding region, either for locally improved resolution or to model a smaller geometric feature. Figure 24 shows the division of four elements of a five by five surface using “Divide Element” on the **Mesh** menu. Figure 24a shows the selection of one element and the choosing of “Divide Element” on the **Mesh** menu. The result is shown in Figure 24b. Each of the smaller elements is one-quarter the size of the surrounding elements. Figure 24d shows the result after all four elements are divided. Elements must be divided one at a time.

Approved for public release; distribution is unlimited.

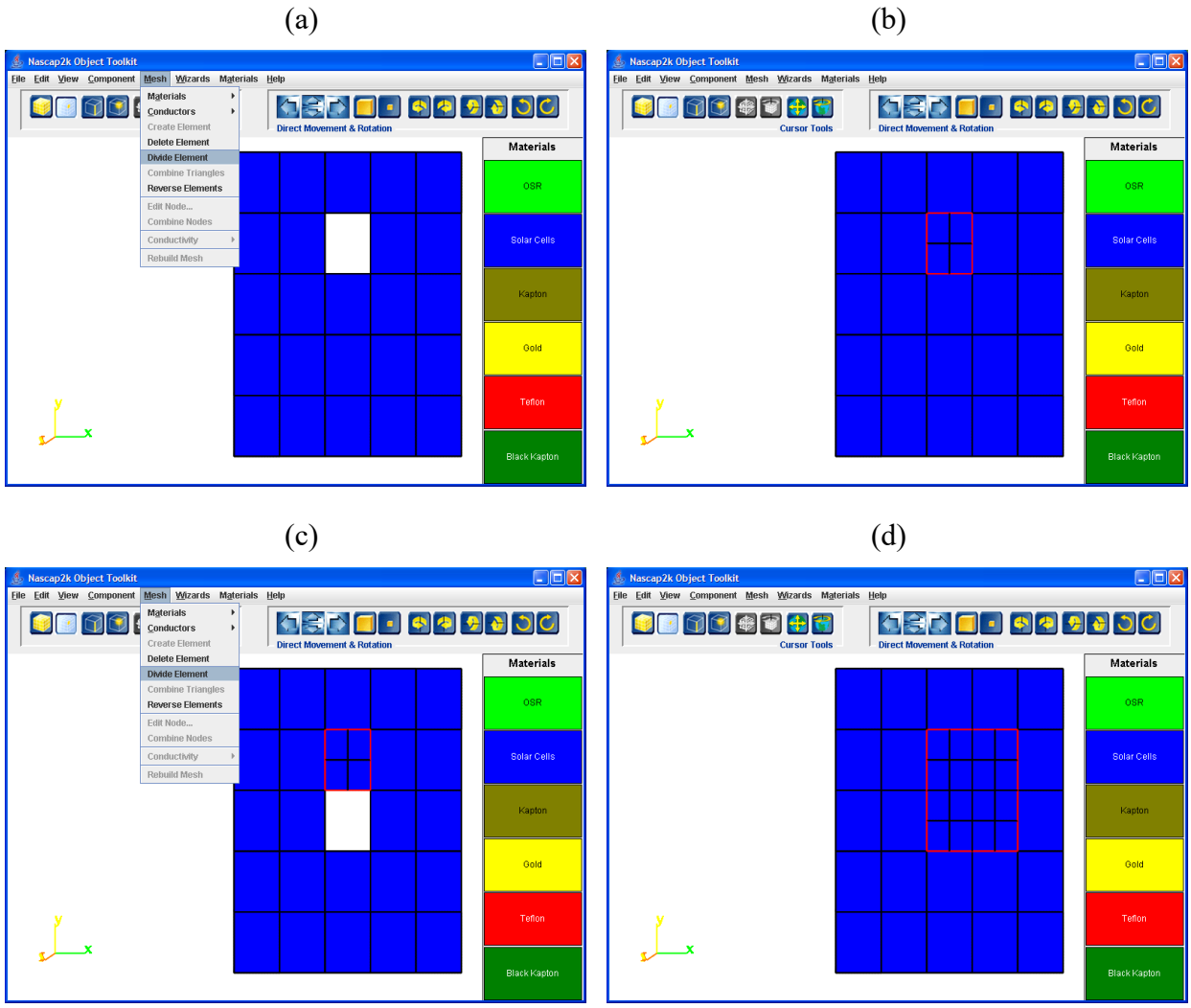


Figure 24. Creating Smaller Elements Using “Divide Element” on the Mesh Menu.

11.2 Element Compatibility

The region with the smaller elements is bordered by a red line, which indicates that the elements are incompatible. Elements are compatible if each edge has exactly one element on its right and exactly one on its left. In Figure 24d when traversing the element counterclockwise, the long red edges have two elements on the right and the short red edges have less than one element on the right. Some codes that rely on *Object Toolkit* for object definition prohibit incompatible meshing (e.g., *Nascap-2k*). Other codes allow it (e.g., *EPIC*). Further mesh editing is needed to resolve the incompatibilities.

11.3 Deleting and Creating Elements

The surface mesh can be revised by deleting and creating individual elements. Figure 25 and Figure 26 show how to delete and create elements and how deleting and creating elements can be used to resolve incompatibilities in a mesh.

Approved for public release; distribution is unlimited.

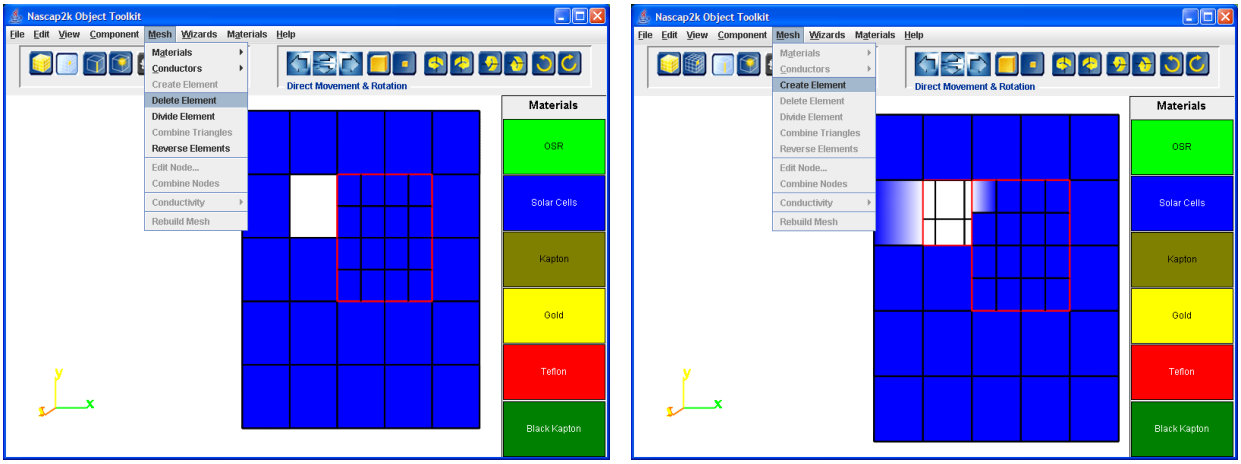
An element is deleted by selecting the element and then choosing “Delete Element” on the **Mesh** menu. Multiple elements can be selected and deleted together. When an element is deleted, any node of that element that is not shared with another element is also deleted.

Elements can be created either from two adjacent edges (to form a triangle), or two opposite edges (to form a quadrilateral) or from three or four nodes. All of the edges or nodes must belong to the same component, which also contains the newly created element. It is recommended that elements be created from edges rather than from nodes whenever possible. Figure 25 shows the creation of two elements from edges. To create a new element, the user selects two edges (holding the CTRL key down for the second selection) and then chooses “Create Element” on the **Mesh** menu. If the two edges have a node in common, a triangular element is created; otherwise, a quadrilateral element is created. Figure 25b and Figure 25c show the selection of two edges and the resulting new quadrilateral element. Figure 25d and Figure 25e show the selection of two edges and the resulting new triangular element. New elements always have the default material for the application. Element materials can be changed by selecting the elements and choosing “Materials” on the **Mesh** menu.

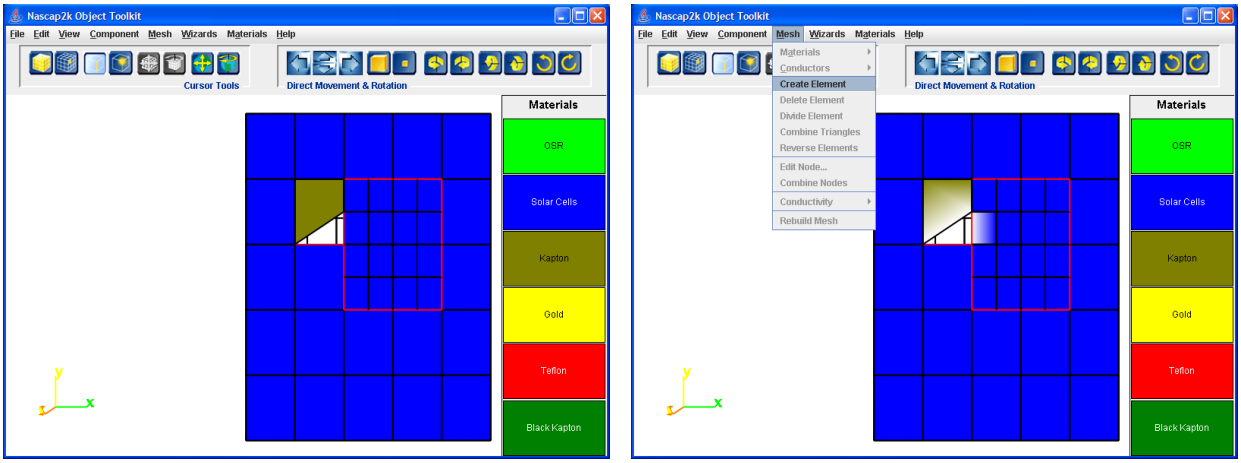
Figure 26 shows the creation of elements from nodes. To create a new element, the user selects three or four nodes in the counterclockwise direction, and then chooses “Create Element” on the **Mesh** menu. *If the nodes are selected in the clockwise direction, the element will face in the wrong direction*. If four nodes are selected, a quadrilateral element is created. If the three nodes are selected, a triangular element is created. Figure 26b and Figure 26c show the selection of four nodes and the resulting new quadrilateral element. Figure 26d and Figure 26e show the selection of three nodes and the resulting new triangular element.

Figure 27 shows how the gap between elements that was created by the use of the “Surface to Surface Wizard” in Section 9 is filled in by creating new elements and then changing their material to solar cells.

(a) (b)



(c) (d)



(e)

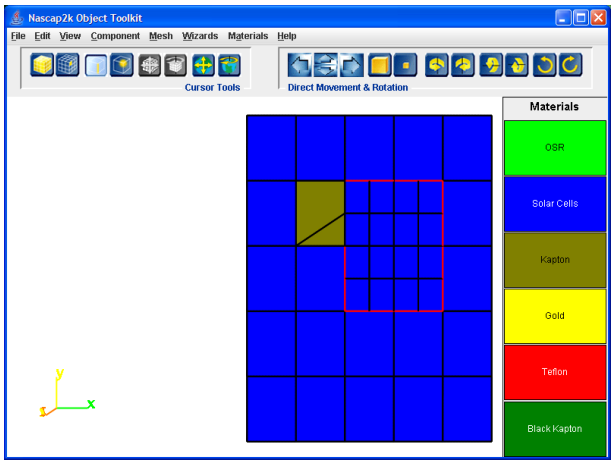
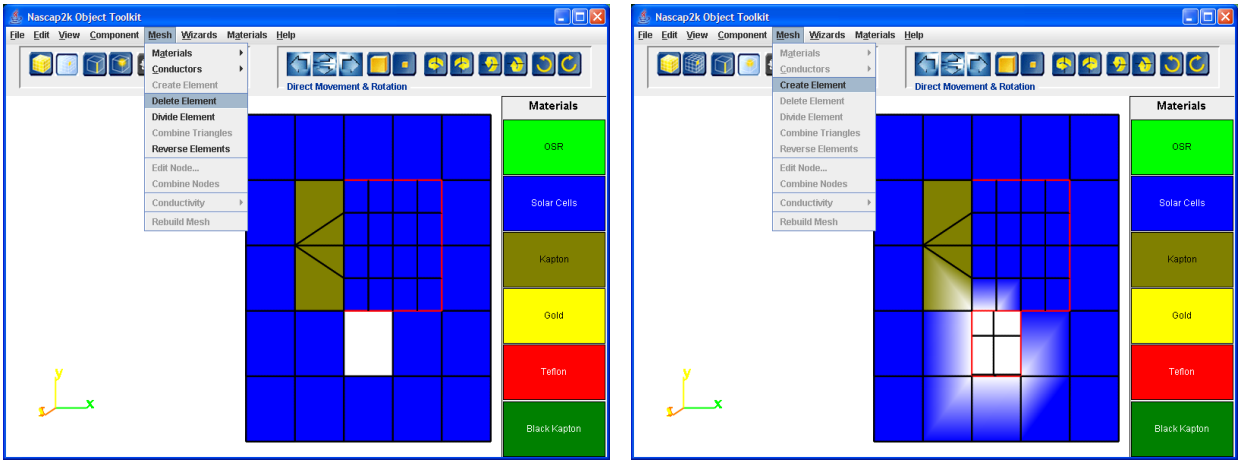


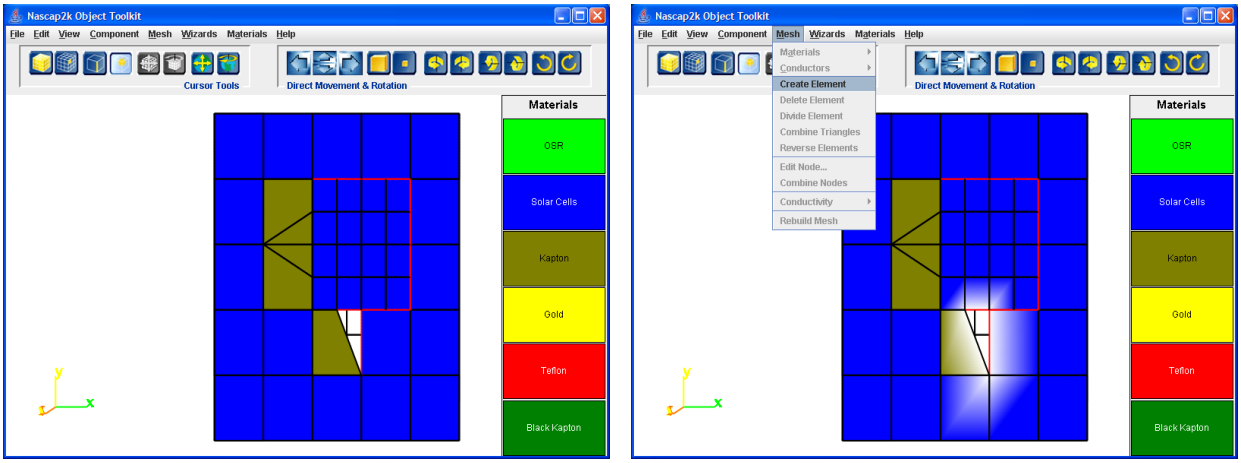
Figure 25. Deleting an Element Using “Delete Element” and Replacing It with a Quadrilateral and a Triangle by Selecting Edges and Using “Create Element” on the Mesh Menu. In b, c, and d, the Outlines of Elements on the Far Side with Normals Facing into the Screen Are Seen.

Approved for public release; distribution is unlimited.

(a) (b)



(c) (d)



(e)

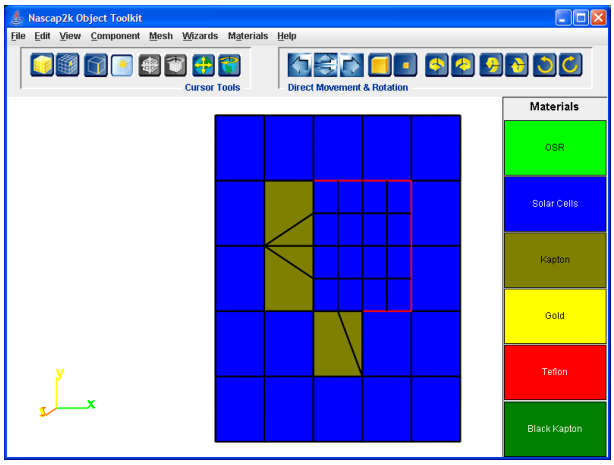


Figure 26. Deleting an Element Using “Delete Element” and Replacing It with a Quadrilateral and a Triangle by Selecting Nodes and Using “Create Element” on the Mesh Menu. In b, c, and d, the Outlines of Elements on the Far Side with Normals Facing into the Screen Are Seen.

Approved for public release; distribution is unlimited.

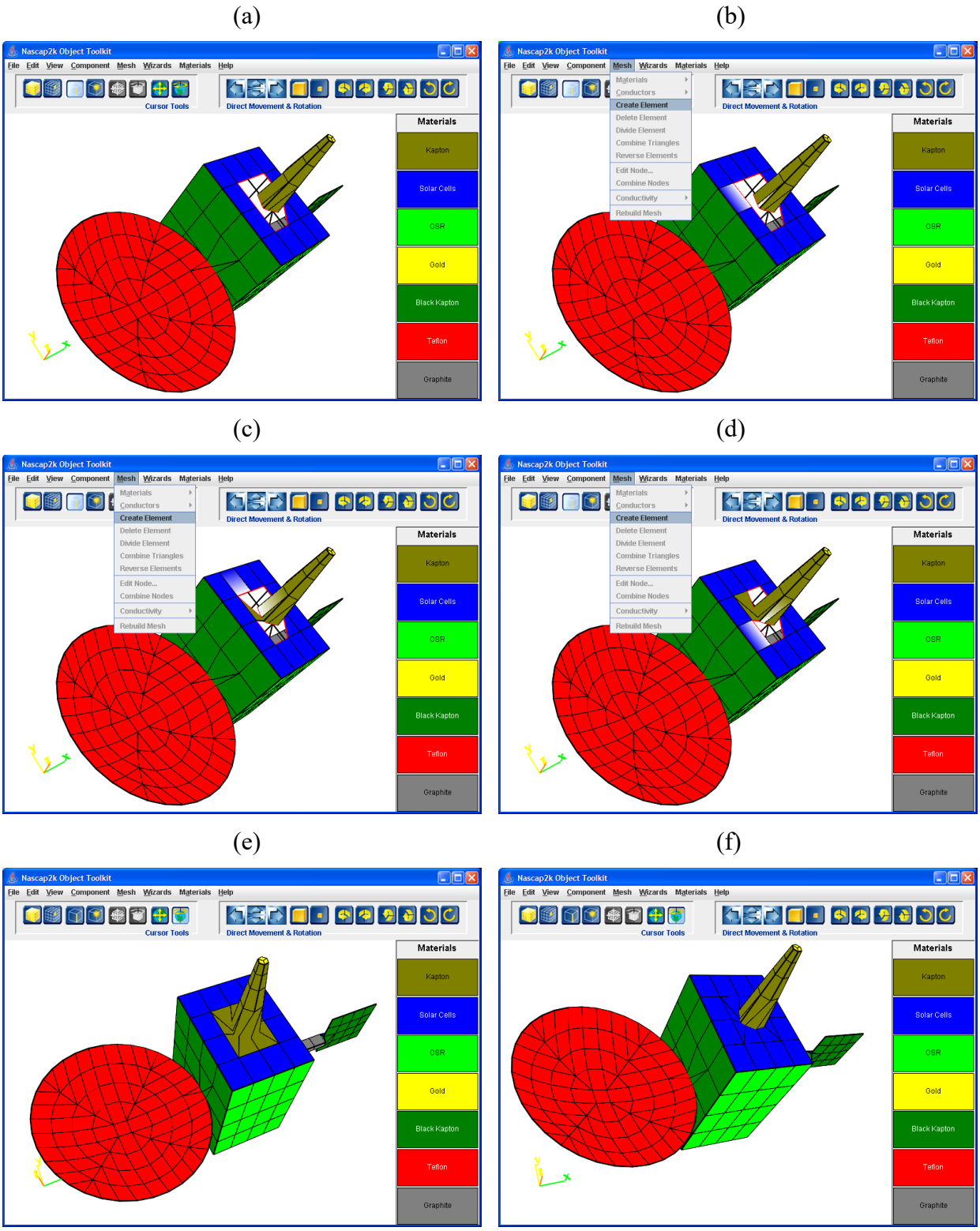


Figure 27. Filling in the Gap Left by the "Surface to Surface Wizard." In a, b, c, and d, the Outlines of Elements with Normals Facing into the Screen Are Seen.

Approved for public release; distribution is unlimited.

11.4 Changing Connectivity (Creating and Deleting Nodes)

Because nodes are always assigned to elements, the easiest way to create new nodes is to divide an element. Dividing a quadrilateral creates four new quadrilaterals and five new nodes. Dividing a triangle creates four new triangles and three new nodes. (Any node that is created at the location of an existing node is automatically merged with the previously existing node.) The newly created nodes can be moved by selecting the node and choosing "Edit Node" on the **Mesh** menu. (See Section 13.2.)

A node that is no longer assigned to an element is automatically deleted.

Connectivity can be changed by merging nodes. Two nodes are merged by selecting them and then choosing "Combine Nodes" on the **Mesh** menu. The second node selected is moved to the location of the first node and the two nodes are combined. Figure 28 shows some examples of how merging nodes can be used to resolve incompatible elements. In Figure 28a, the element on the side is divided. In Figure 28b, two nodes on the side are selected for combining. The top node is selected first because the desired result is that seen in Figure 28c—the quadrilateral has been changed into a triangle. In Figure 28d two nodes are again chosen for combining. The left-hand node is selected second because it is an extra node that needs to be eliminated. The result after the merge is shown in Figure 28e. Figure 28e also shows two more nodes selected for merging. This final merge eliminates a triangle and gives the result shown in Figure 28f.

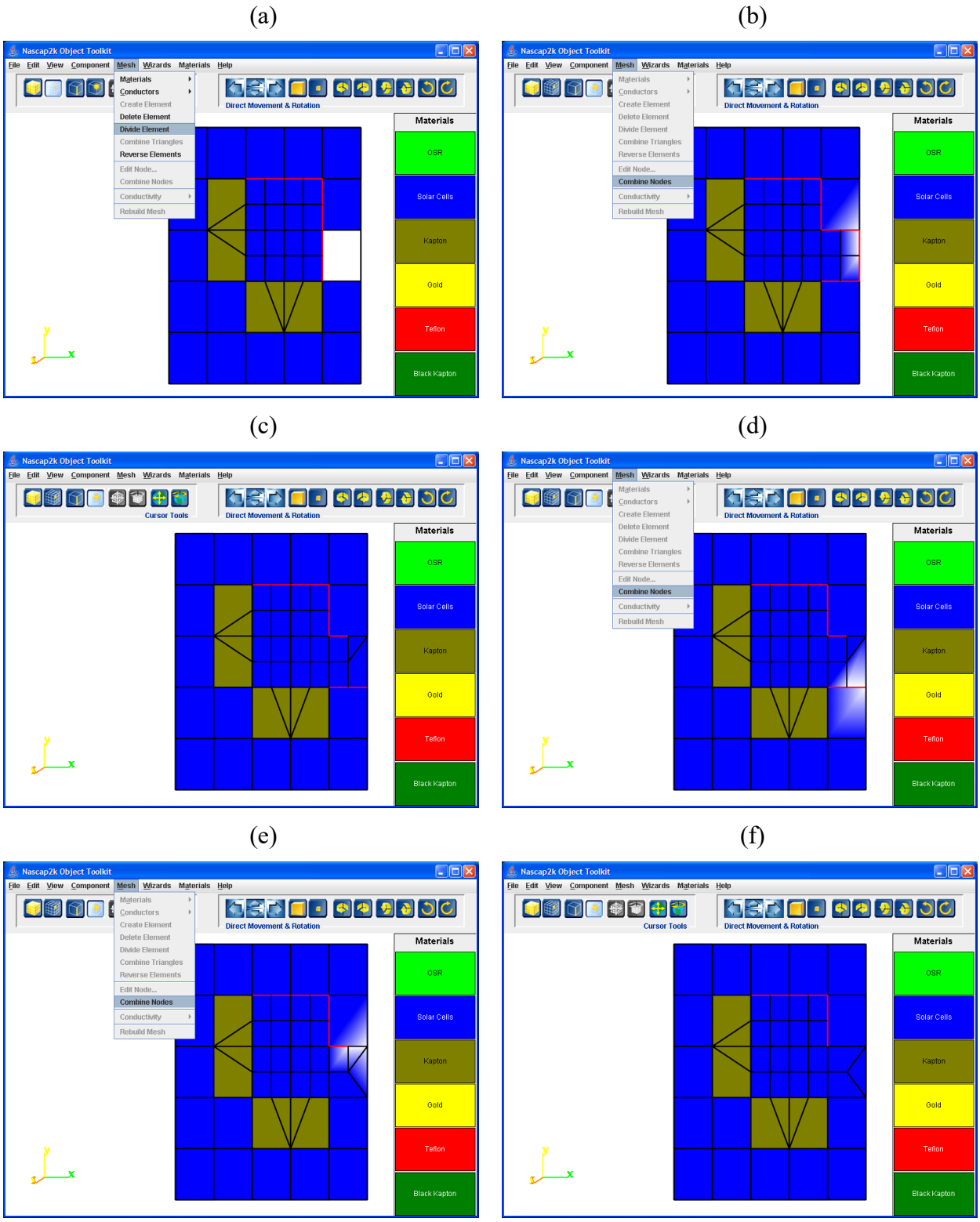


Figure 28. Resolving Incompatibilities by Dividing an Element Using “Divide Element” and “Combine Nodes” to Create Two Quadrilaterals and a Triangle in Place of the Original Single Element.

Approved for public release; distribution is unlimited.

11.5 Combining Elements

Two triangles that share an edge can be combined into a single quadrilateral. The last step in Figure 29 (e and f) shows how two triangles can be selected and then combined by selecting the triangles and choosing “Combine Triangles” on the **Mesh** menu. In Figure 29a, the first of the top two surfaces is selected for dividing. In Figure 29b, the two side nodes are selected for combining. The lower node, in the center of the edge, was selected second in order to eliminate it. In Figure 29c, two more nodes are again selected for merging in order to eliminate an extra node. The result is the two triangles seen in Figure 29d. In Figure 29e, the two triangles are selected, ready for merging into a single quadrilateral. The result after the merge is shown in Figure 29f.

Approved for public release; distribution is unlimited.

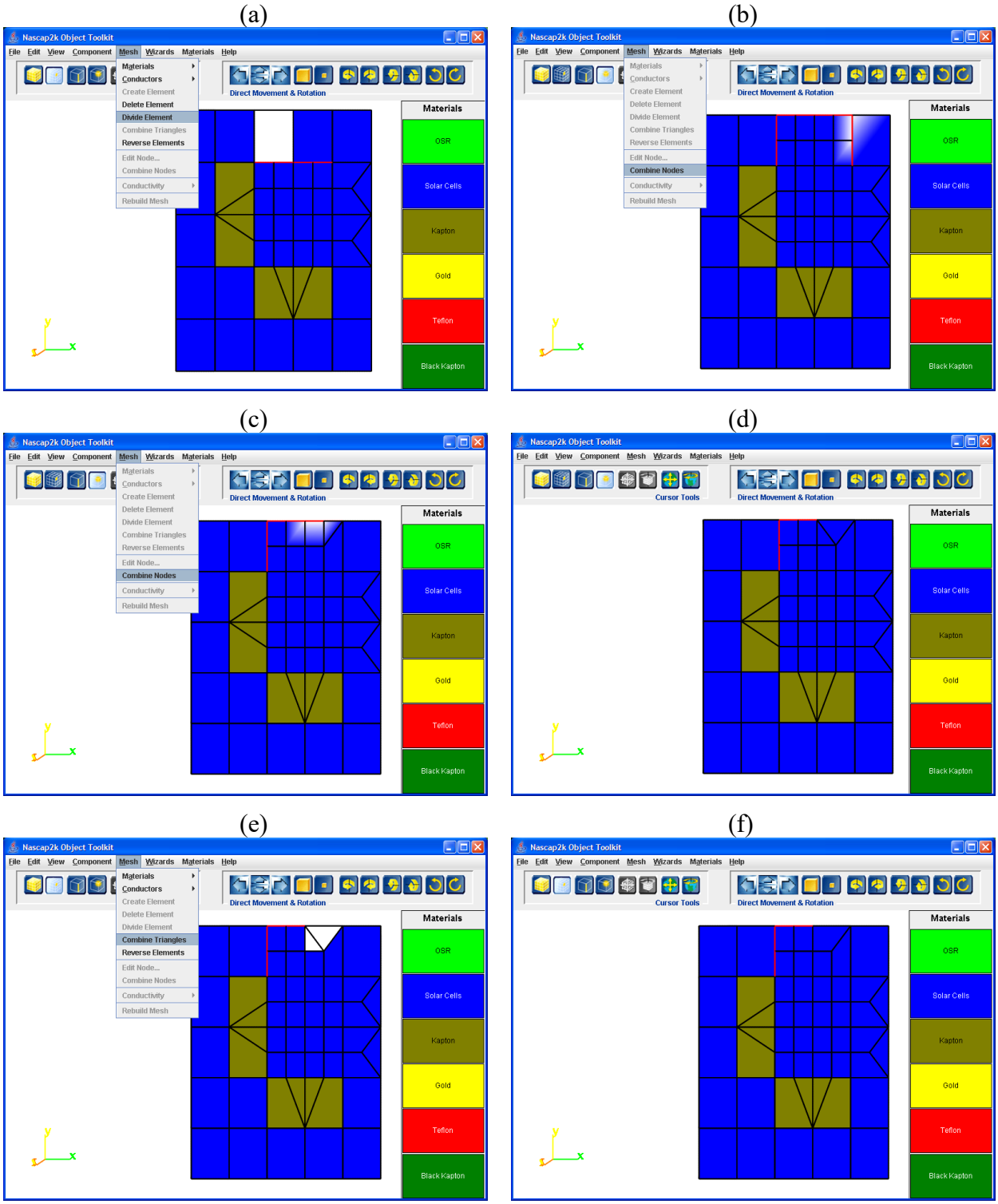


Figure 29. Resolving Incompatibilities by Dividing an Element, Combining Nodes Twice to Create Two Quadrilaterals and Two Triangles, and, Finally, Combining the Two Triangles into a Single Quadrilateral to Give Three Quadrilaterals in Place of the Original Single Element. In e, the Outlines of Elements on the Far Side with Normals Facing into the Screen Are Seen.

Approved for public release; distribution is unlimited.

11.6 Changing Surface Material Assignments

When an element is created, a surface material name is associated with it. The default material is "Kapton." Figure 30 shows how to change the surface material assigned to one or more elements by selecting the elements, going to the **Mesh** menu, selecting "Materials," and choosing the desired material from the fly-out menu.

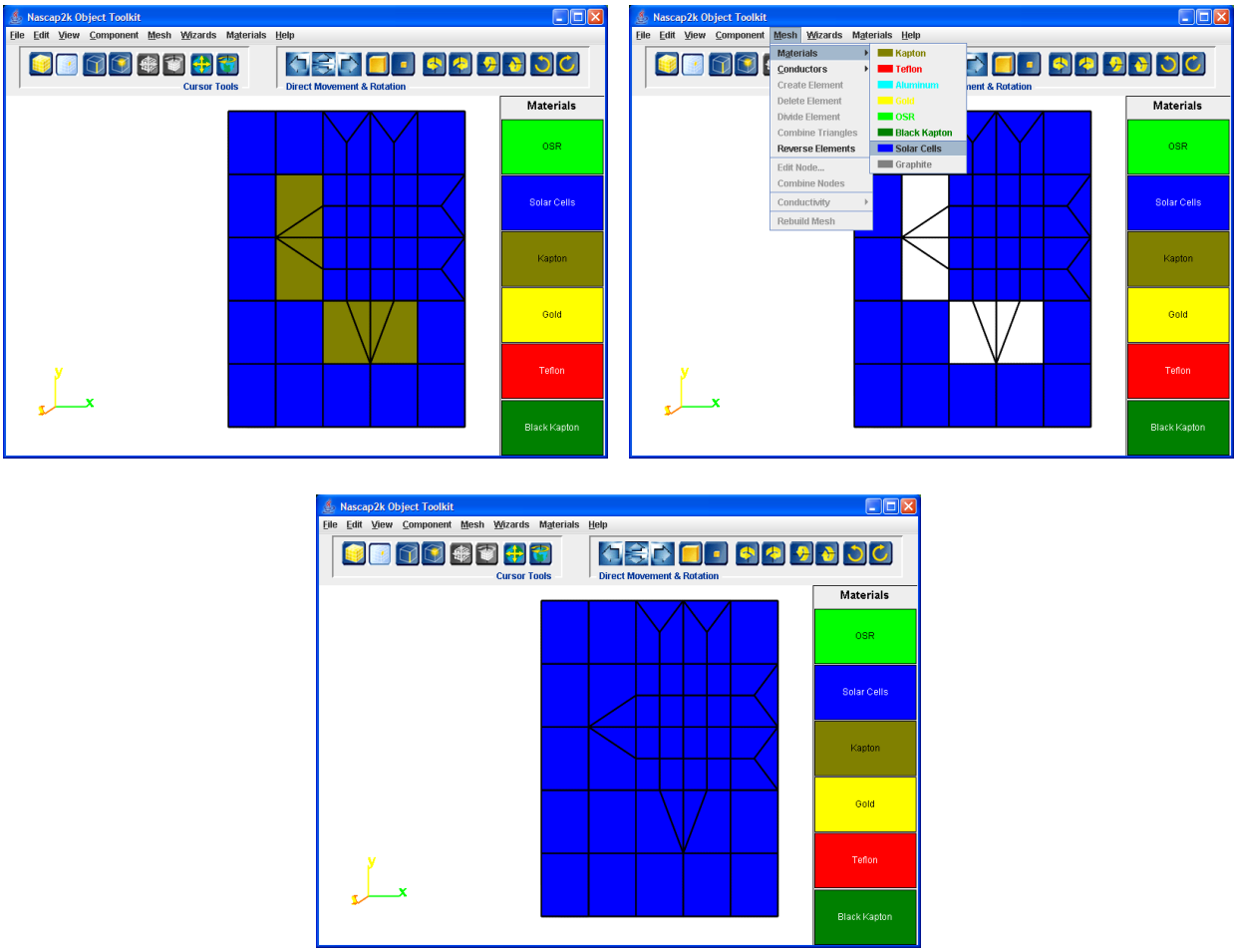


Figure 30. Changing the Surface Materials of Newly Created Elements by Going to the Mesh Menu and Choosing "Materials."

Approved for public release; distribution is unlimited.

12 Importing Components from Files

Object Toolkit can import finite element components written in any of several common formats, including

- Patran Neutral File (*.neu, *.pat)
- *NX I-DEAS* TMG ASCII VUFF File (*.tmg)
- *Gmsh* File (*.msh)
- *Nastran* Bulk Data File (*.bdf)
- *I-DEAS* Universal File (*.unv)
- STEP File (*.stp)

Geometry files that do not contain finite element descriptions must be surface meshed by other software in order to be imported into *Object Toolkit*. Section 22 provides an example of using the open source *Gmsh* program to mesh a STEP geometry file and import the result into *Object Toolkit*.

12.1 Importing Components Built Using *Object Toolkit*

"Add Component from File" on the **Component** menu is used to import components built using *Object Toolkit*. "Add Component from File" combines all the components saved in the specified file into one assembly and adds it to the present object. The original location and orientation are retained during this operation. Any material definitions are also imported and replace the existing (or default) material properties with the definitions found in the imported file.

12.2 Importing Components from a *PATRAN* Neutral File

PATRAN neutral files can be read into *Object Toolkit* by selecting "Open…" on the **File** menu or "Add Component from File" on the **Component** menu. The *PATRAN* file is identified by the extension ".neu" or ".pat." (This may require renaming the file.) Elements imported from a *PATRAN* neutral file are assigned the default material for the application. The *PATRAN* object surface must be meshed with linear triangles (TRI/3/icond) and linear quadrilaterals (QUAD/4/icond).

12.3 Importing Components from a *NX I-DEAS* TMG ASCII VUFF File

Users of the *NX I-DEAS* TMG thermal analysis program can import the ASCII version of the VUFF file into *Object Toolkit* by selecting "Open…" on the **File** menu or "Add Component from File" on the **Component** menu. The file is identified by the extension ".tmg". (This will require renaming the file.) Elements imported from a *NX I-DEAS* TMG ASCII VUFF file are assigned a material name consisting of the first four letters of the `RDFGROUPNAME` specified in the file. The corresponding properties are set to those of the default material for the application. The object surface must be meshed with linear triangles and linear quadrilaterals.

12.4 Importing Components from a *Gmsh* MSH Formatted File

Files in the *Gmsh* MSH format can be read into *Object Toolkit* by selecting "Open…" on the **File** menu or "Add Component from File" on the **Component** menu. *Gmsh* is an open source three-

Approved for public release; distribution is unlimited.

dimensional finite element mesh generator with a built-in Computer aided design engine and post-processor. The MSH formatted file is identified by the extension ".msh". Imported elements are assigned the default material for the application. The object surface must be meshed with 3-node triangles (element type 2) and 4-node quadrangles (element type 3). Elements of other types are ignored.

12.5 Importing Components from a *Nastran* BDF or DAT File

Nastran BDF (Bulk Data Format) or DAT (DATa) files can be read into *Object Toolkit* by selecting "Open…" on the **File** menu or "Add Component from File" on the **Component** menu. The BDF or DAT file is identified by the extension ".bdf". (This may require renaming the file.) Imported elements are assigned the default material for the application. The object surface must be meshed with quadrilaterals (CQUAD, CQUAD4, CQUAD8, CQUADR, CQUADX) and triangles (CTRIA3, CTRIA6, CTRIAR, CTRIAX, CTRIAX6). *Object Toolkit* only accommodates linear elements. For the higher order elements (CQUAD8 and CTRIAX6) only the corner nodes are read.

12.6 Importing Components from a *I-DEAS* Universal File

I-DEAS universal files can be read into *Object Toolkit* by selecting "Open…" on the **File** menu or "Add Component from File" on the **Component** menu. The universal file is identified by the extension ".unv". Imported elements are assigned the default material for the application. The object surface must be meshed with linear triangles (element types 41 and 91) and linear quadrilaterals (element types 44 and 94).

12.7 Importing Components from a STEP File

Finite element models in some STEP (Standard for the Exchange of Product model data) files can be read into *Object Toolkit* by selecting "Open…" on the **File** menu or "Add Component from File" on the **Component** menu. A STEP file is an ASCII data exchange form of STEP, which can represent three dimensional objects in Computer-aided design (CAD) and related information. The STEP file is identified by the extension ".stp". Only STEP files containing a finite element model can be imported as *Object Toolkit cannot* mesh an externally defined geometric description. (See Section 22 for an example of meshing a STEP geometry file. Imported elements are assigned the default material for the application. The object surface must be meshed with linear triangles and linear quadrilaterals. If only three nodes are present, the element is a triangle, otherwise the first four nodes are taken to be the corners of a quadrilateral surface element in counterclockwise order as viewed from the object exterior. If more than four nodes are present, only the first four are read.

13 Building Custom Components Using Advanced Mesh Editing and Primitives

Object Toolkit provides tools for creating and saving a wide variety of shapes. Section 6 addresses the use of standard components. Section 11 describes basic mesh editing. This section describes advanced mesh editing and the creation and use of Primitives. Once a Primitive is created, it can be scaled and reused in models of different objects. It should be remembered, however, that all information about how the component was originally created is lost once it is turned into a Primitive.

13.1 Creating a Cavity

A common use of "Reverse Elements" on the **Mesh** menu is in the creation of convex surfaces. The desired shape is built with the elements pointing outward and then "Reverse Elements" is used to invert the selected elements. Figure 31 shows how to create a hollowed-out box. Between Figure 31a and Figure 31b, the +X surfaces of both Boxes are removed using "Delete Element" on the **Mesh** menu. Figure 31b shows the smaller Box being moved inside the larger Box using "Node Relative Move" on the **Component** menu. In Figure 31d the elements of the inner Box are reversed by selecting the elements and choosing "Reverse Element" on the **Mesh** menu. The components are then grouped into an assembly by selecting "Consolidate Components into Assembly" on the **Component** menu. Between Figure 31e and Figure 31f, additional elements to join the inside to the outside are added by successively selecting two edges and then going to the **Mesh** menu and choosing "Add Element." The Direct Rotation tools are used to move the view to select the desired edges.

Approved for public release; distribution is unlimited.

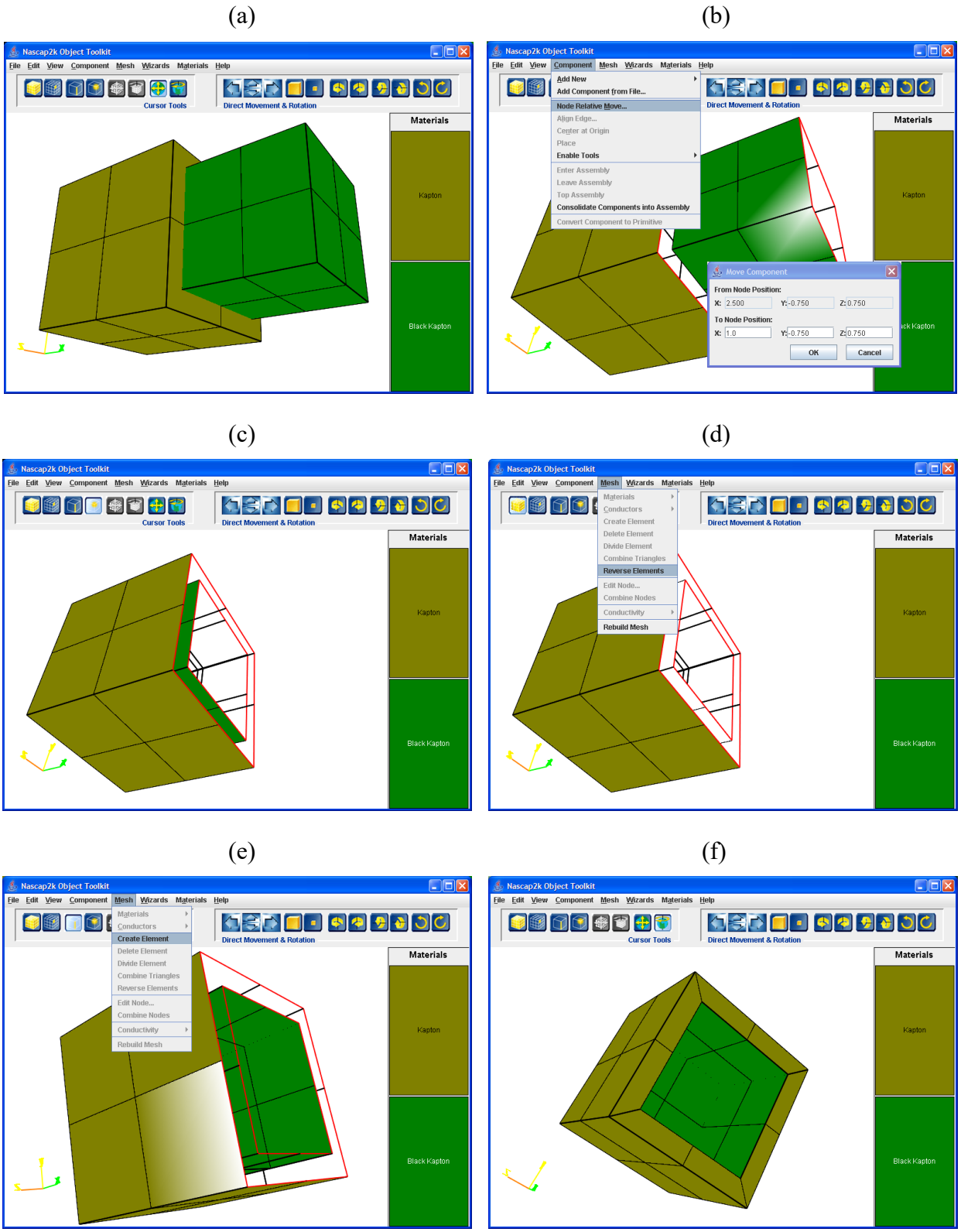


Figure 31. Creating a Cavity Using “Reverse Element” on the Mesh Menu. In b, c, d, and e, the Outlines of Elements with Normals Facing into the Screen Are Seen.

Approved for public release; distribution is unlimited.

13.2 Creating a User-Specified Shape

The shape of a component is changed by placing nodes at specific locations. This is done by selecting a node, going to the **Mesh** menu, choosing "Edit Node," and then specifying the location in the pop-up dialog box. The position is specified in the component local coordinate system. Figure 32 shows how to use "Edit Node" to create a user-specified shape.

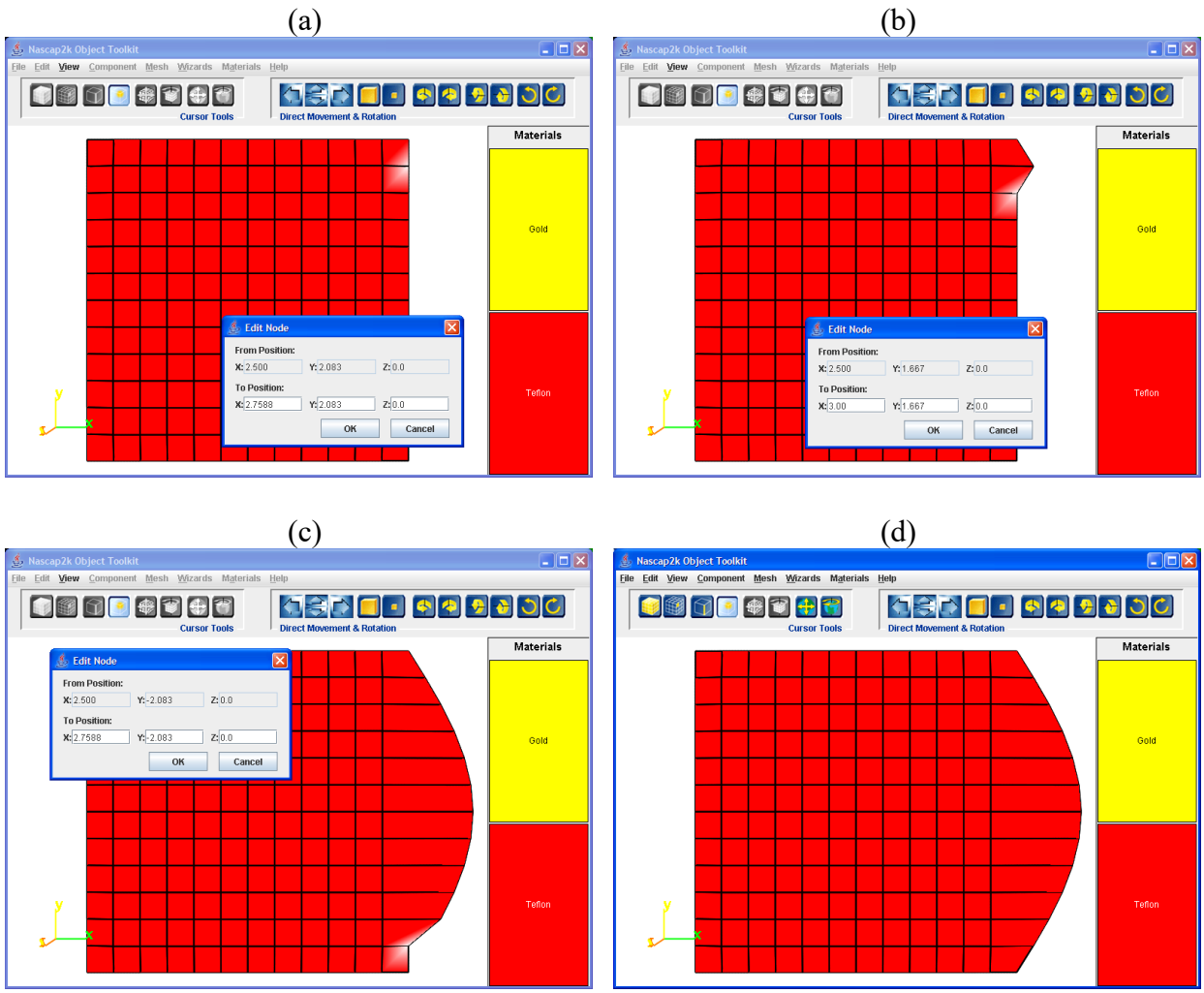


Figure 32. Changing the Edge of a Panel from a Straight Edge to a Cosine Curve by Repeated Use of "Edit Node" on the Mesh Menu.

Approved for public release; distribution is unlimited.

13.3 Scaling a Component

The Primitive is the only type of component that can be scaled. Once a component has been converted to a Primitive, any assembly structure is lost. Primitives are scaled by selecting the Primitive and choosing “Edit Component” on the **Edit** menu. The Primitive can be scaled by a different factor in each of the X, Y, and Z directions. Figure 33 shows the conversion of a Dish to a Primitive and the stretching of the Dish in the Y direction to make it ellipsoidal.

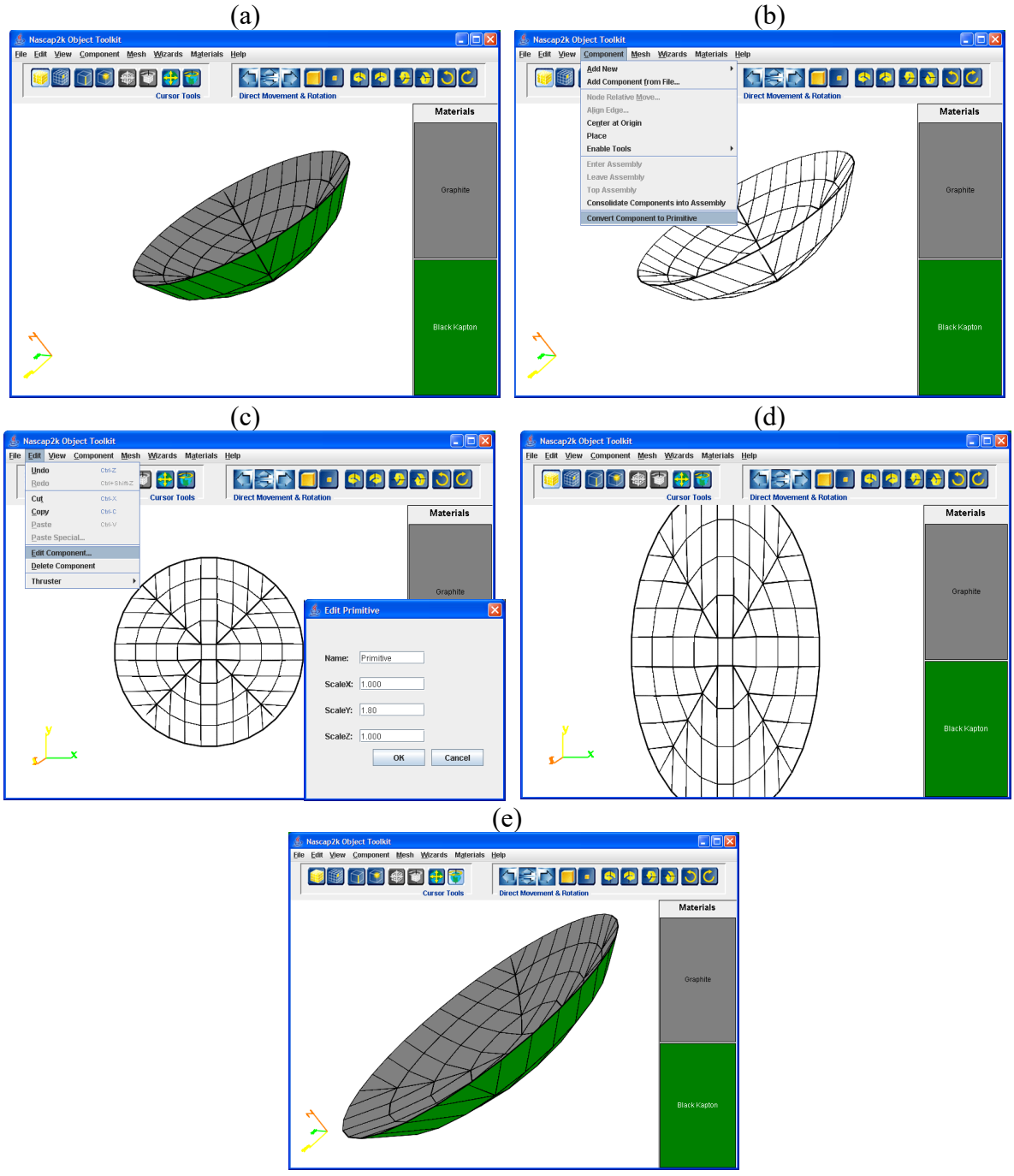


Figure 33. Creation and Scaling of a Primitive. A Dish Is Converted to a Primitive and Then Scaled Anisotropically.

Approved for public release; distribution is unlimited.

14 Editing Components Within an Assembly

An assembly is a component created by combining two or more components. Assemblies are created by the wizards and by "Consolidate Components into Assembly" on the **Component** menu. An assembly can be manipulated in *Object Toolkit* just as any other component. Figure 15 in Section 7 and Figure 17 in Section 8 show the effect of selecting an edge and then going to the **Component** menu and choosing "Align Edge" when the edge is part of a simple component and when the edge is part of an assembly.

Once separate components have been combined into an assembly, the user may wish to edit one of the constituent components separately. To some extent this can be done with caution as long as the assembly has *not* been converted to a Primitive. "Enter Assembly," "Leave Assembly," and "Top Assembly," which appear on the **Component** menu, allow the user to edit a constituent component. Changes to surface attributes (such as material name) can always be done in this way. However, changes to meshing or geometry may result in a change in the numbering of the elements or nodes, possibly leading to bizarre errors in the top-level object.

Figure 34 shows how changing the proportions of a standard component results in a scrambling of the final mesh editing. In the figure, the Box is selected and "Enter Assembly" on the **Component** menu is chosen until it is possible to select the Box alone. In Figure 34c through Figure 34e, the **Edit** menu item "Edit Component" is used to change surface materials and the extent of the Box in the Y direction. Upon returning to the top assembly (Figure 34f), the surface material changes appear as desired, but the change in the proportions of the Box results in a renumbering that scrambles the final elements added to connect the Cylinder and the Box.

Approved for public release; distribution is unlimited.

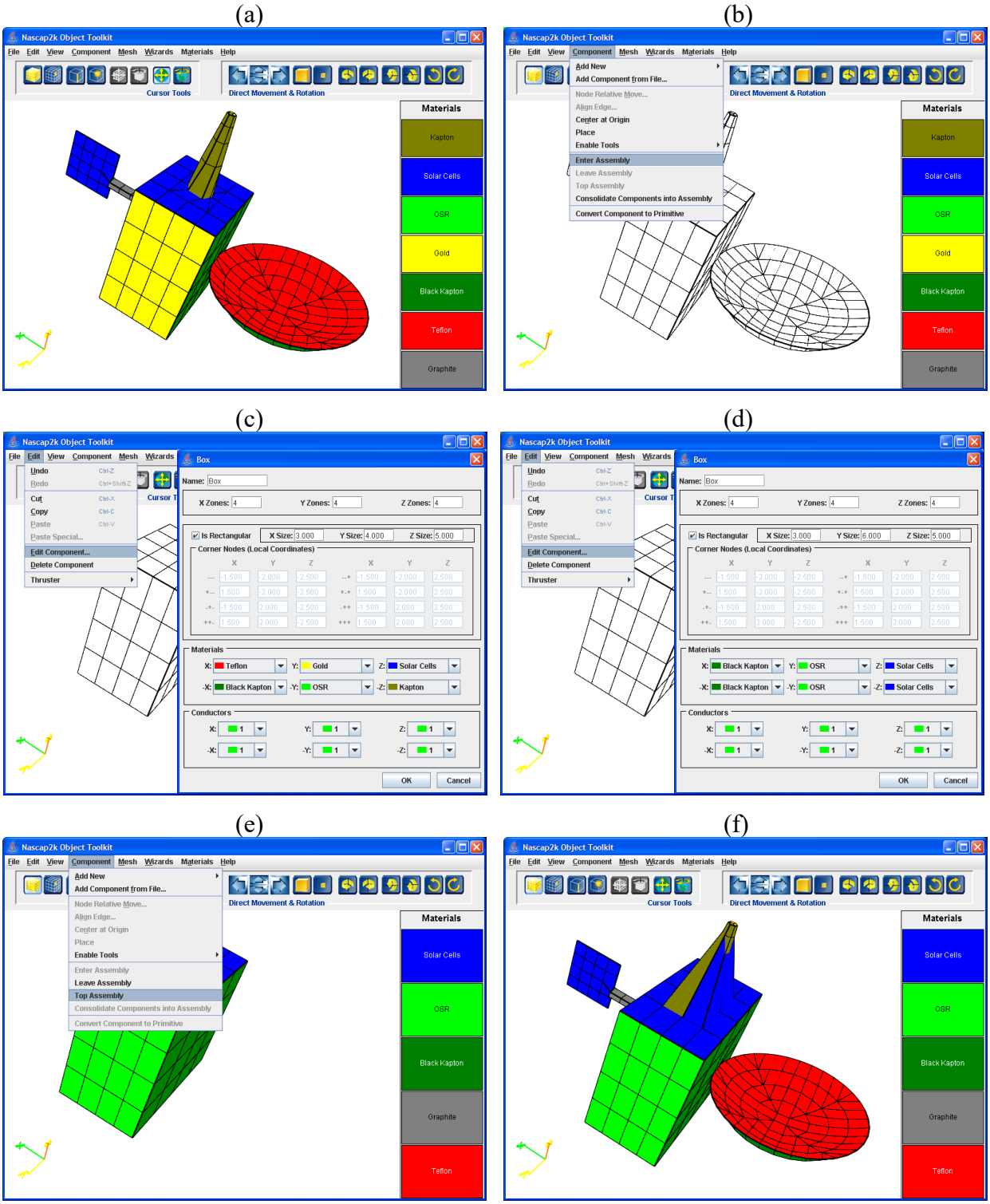


Figure 34. Editing of a Box Several Layers Down in the Assembly Structure. The Surface Materials and the Extent of the Box in the Y Direction Are Changed. Upon Returning to the Top Assembly, the Surface Material Changes Appear As Desired, but the Change in the Proportions of the Box Results in a Renumbering That Scrambles the Final Elements Added to Connect the Cylinder and the Box.

Approved for public release; distribution is unlimited.

15 Application-Specific Attributes

Several software packages use *Object Toolkit* to generate an object description for their use. Capabilities associated each application only appear when *Object Toolkit* is initiated with a command line argument indicating the specific application, (-app *App*). *Object Toolkit* then searches for an *App*_OTKSpecs.xml file that specifies the customizations for the requested application. Files that include the appropriate command line arguments (".bat" files) are installed along with *Nascap-2k* and *EPIC* and along with *Object Toolkit* itself when installed for *MEM*.

15.1 Conductor Number

The conductor number is an integer (1–25) attribute used by *Nascap-2k* to define electrical connectivity. (While only used by *Nascap-2k*, this attribute always appears.) Each element has a conductor number. These assignments may be used for purely diagnostic purposes or as a basis for internal spacecraft circuitry. Conductor number 1 is considered "spacecraft ground." Higher-numbered conductors can be specified as biased relative to ground or floating.

Conductor numbers for elements in standard components are assigned to elements in the same manner as materials. The conductor number on selected elements is changed by going to the **Mesh** menu and choosing "Conductors."

The conductor numbers assigned to elements can be viewed by going to the **View** menu and choosing "Conductors."

15.2 Solar Array Wizard

The "Solar Array Wizard" simplifies the specification of conductor numbers for solar arrays. The conductor numbers increase sequentially along the user-specified path, as might be desired when simulating a string of solar cells. The interface is shown in Figure 35. The user selects "Solar Array Wizard" from the **Wizard** menu. The conductor number of the low-end of the string is specified in the dialog box and the surface element to have this conductor number is selected on the object. In the second step, the high-end element of the string is selected. Conductor numbers are assigned along a path between the two surface elements. If there are more than 100 elements, some elements will be assigned the same conductor number. As there are only 32 colors, if there are more than 32 conductor numbers, the same color is used for multiple conductor numbers. Multiple strings of conductor numbers can be assigned at the same time.

Approved for public release; distribution is unlimited.

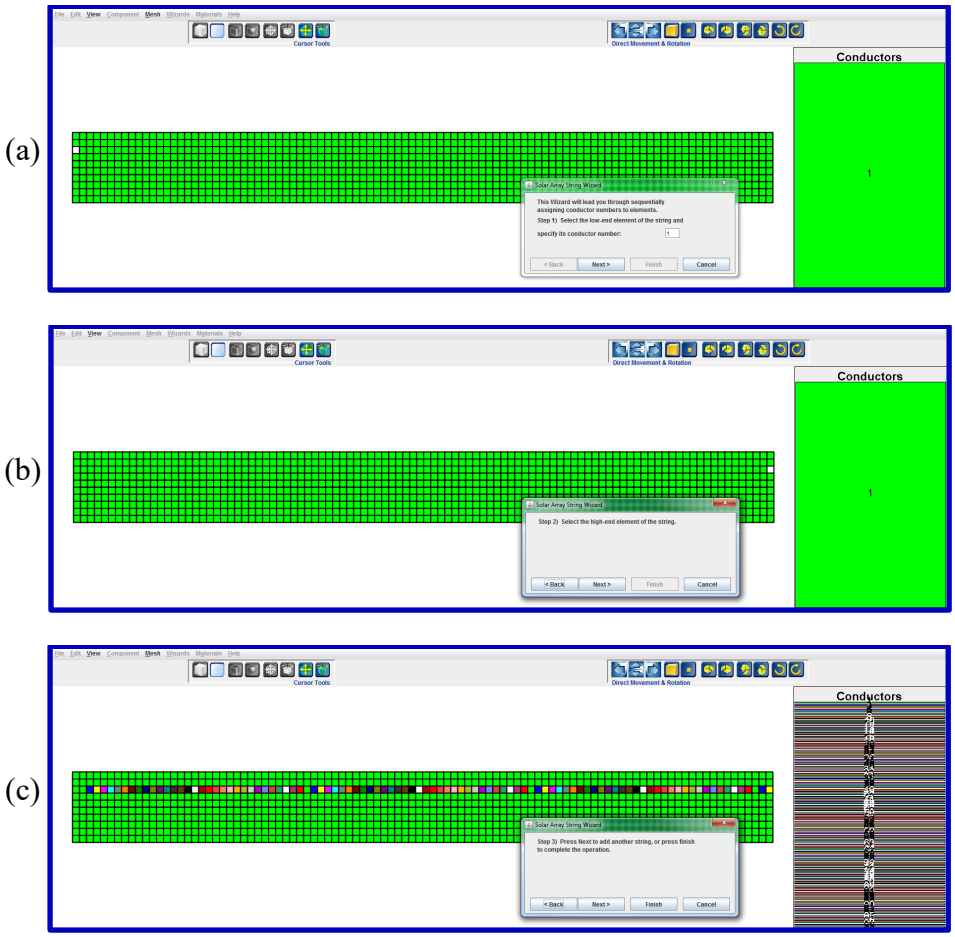


Figure 35. Defining a Solar Array String using the “Solar Array Wizard.”

15.3 Conductivity

The menu item “Conductivity” on the **Mesh** menu is used to specify that a selected node or edge is to be a grounding element in *Nascap-2k*. (While only used by *Nascap-2k*, this capability always appears Surface conductivity operates between insulating elements of a common material and with a common edge, thus covering transport over a wide expanse of such material. To ground the material, the user can specify grounding by a strip at an element edge or by a circular contact located at a node. These grounding elements can be specified only for Primitive components, because other types of components frequently have their meshes re-created. A grounding edge establishes a conductance of $L/D\kappa$ to ground from each of the two neighboring elements, where L is the length of the edge, D is the distance from the center of the element to the center of the edge, and κ is the surface resistivity of the material. A grounding node establishes a conductance to ground of $\theta/\ln(D/r)/\kappa$ from each of the neighboring elements, where θ is the angle the element subtends at the node, D is the distance from the element center to the node, and r is the node radius. The node radius is user-specified.

Approved for public release; distribution is unlimited.

15.4 Particle Instruments (Emitters and Detectors)

Emitters and Detectors can be defined for use by *Nascap-2k* in simulations of charged particle instruments. They are created by assigning either an emitter or detector name to one or more surface elements. They are assigned to surface elements in a manner similar to how materials are assigned. “Emitter” and “Detector” on the **Mesh** menu can be used to specify the Emitter or Detector name for selected surface elements. “Assign Emitter” and “Assign Detector” on the **Component** menu can be used to specify the Emitter or Detector name for all the elements of the selected component. The Emitter and Detector assignments can be viewed by selecting “Emitter” or “Detector” on the **View** menu.

The **Emitter Properties** and **Detector Properties** dialog boxes are viewed by selecting “Emitter” or “Detector” respectively on the **Edit** menu and then selecting the specific Emitter or Detector on the fly-out menu. The properties can also be edited within the *Nascap-2k* user interface. The meaning of these properties is discussed in the *Nascap-2k User’s Manual*.

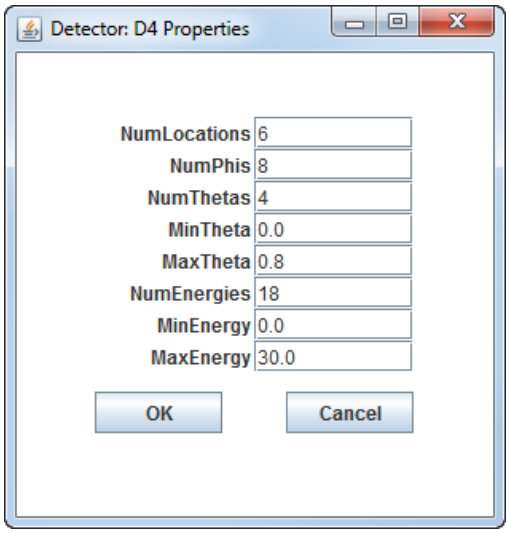


Figure 36. Detector Properties Dialog Box.

15.5 Antenna Surface Currents (Injection Point)

Injection Points can be defined for use by *Nascap-2k* in simulations of the current that flows along the surface of a high voltage antenna in a plasma environment. They are created by assigning an Injection Point name to one or more surface elements. They are assigned to surface elements in a manner similar to how materials are assigned. “Injection Point” on the **Mesh** menu can be used to specify the Injection Point name for selected surface elements. “Assign Injection Point” on the **Component** menu can be used to specify the Injection Point name for all the elements of the selected component.

The Injection Point assignments can be viewed by selecting “Injection Point” on the **View** menu. While “Injection Point” appears on the **Edit** menu, an Injection Point has no properties.

Approved for public release; distribution is unlimited.

15.6 Pointing (Subsystems)

If *Object Toolkit* is running customized for *EPIC* or *MEM*, subsystems can be defined. Subsystems are used to specify rotating objects. Each rotating object rotates with respect to the rest of the object in order to track the sun or another body. An *EPIC* spacecraft is composed of objects, which correspond to subsystems in *Object Toolkit*.

Each element has a subsystem attribute. By default, elements are assigned the subsystem "defaultSubsystem." The menu items "Assign Subsystem" on the **Component** menu and "Subsystem" on the **Mesh** menu are used to change the subsystem attribute for the surfaces of a component and for selected surface elements, respectively. The choice "Other" brings up a dialog box on which to specify a new subsystem. A subsystem only exists as long as elements are assigned to it.

Just as object surfaces can be viewed color-coded by material name or conductor number, color-coding by "Subsystem" can be selected on the **View** menu.

The **Subsystem Properties** dialog box, seen in Figure 37, defines object tracking. All of the elements with the same subsystem name move as a single object. If the target value is "(None)" the subsystem does not track. If the target value is "sun," the subsystem rotates with respect to the rest of the spacecraft to track the sun. Additional targets are available for *MEM*. Four vectors are used to define the tracking:

Fixed Point: Corresponds to the location of the joint about which the object rotates.

Pointing Vector: A directional vector that originates at the geometric center of the object and passes through the point on the surface that is to be aimed at the target.

First Axis: Defines the axis about which the first rotation takes place. Note that the first axis of rotation should not be parallel to the pointing vector. This is the equivalent of rotating a vector about itself and defines rotation about the second axis only.

Second Axis: Defines the axis about which the second rotation takes place. To rotate about one axis only, set both axes the same. Note that the second rotation takes place about the rotated second axis.

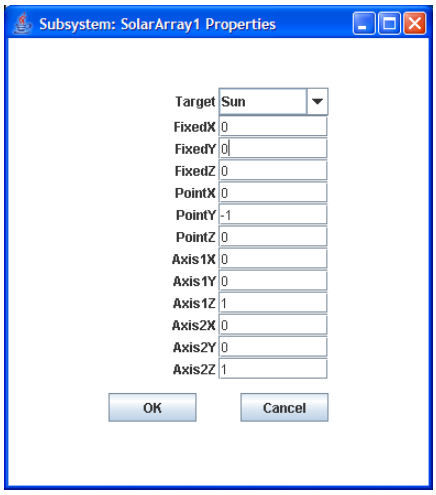


Figure 37. Subsystem Properties Dialog Box.

Approved for public release; distribution is unlimited.

15.7 Thrusters

An *EPIC* or *Nascap-2k* object may include one or more thrusters. Attributes of a thruster are a name, a type, a position (x, y, z) in meters relative to the *Object Toolkit* origin, a color (r, g, b), a display size, and a direction (dirx, diry, dirz). For asymmetric thrusters, a second direction is added within the *EPIC* user interface. The "type" property is not used by either *Nascap-2k* or *EPIC* at present. The color and display size give the color and radius of the sphere that *Object Toolkit* uses to display the thruster location on the screen. A thruster is added to the object by going to the **Component** menu and selecting "Add New" and then choosing "Thruster" from the fly-out menu that appears.

The properties of thrusters are edited by going to the **Edit** menu, selecting "Thrusters," and changing the values that appear in the dialog box (Figure 38). The position and direction can also be changed within the *EPIC* user interface, and the direction can be changed on the *Nascap-2k* interface to allow for gimbaling. The *Nascap-2k* interface also allows thrusters to be turned on or off.

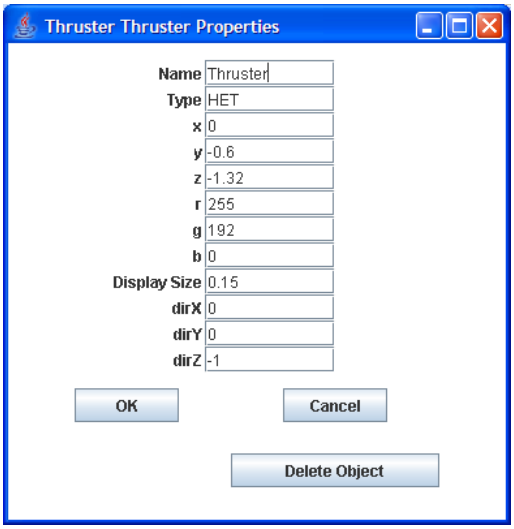


Figure 38. Thruster Properties Dialog Box.

15.8 Neutralizers

A *Nascap-2k* object may include one or more neutralizers. Attributes of a neutralizer are a name, position (x, y, z) in meters relative to the *Object Toolkit* origin, display color (r, g, b), display size, and direction (dirx, diry, dirz). A neutralizer is added to the object by going to the Component menu and selecting "Add New" and then choosing "Neutralizer" from the fly-out menu that appears.

15.9 Magnetic dipoles

Magnetic fields from electric propulsion devices and magnetic torque rods can be considerable. It is common to model spacecraft-generated magnetic fields using a set of point dipoles with appropriate locations and magnetic moments. A *Nascap-2k* object may include one or more magnetic dipoles. Attributes of a magnetic dipole are a name, position (x, y, z) in meters relative to the *Object Toolkit* origin, display color (r, g, b), display size, and a vector moment (Px, Py, Pz)

Approved for public release; distribution is unlimited.

in units of ampere-meters^2. A magnetic dipole is added to the object by going to the **Component** menu and selecting "Add New" and then choosing "MagDipole" from the fly-out menu that appears.

15.10 Move Wizards

When building an object for *EPIC* or *MEM*, *Object Toolkit* has four wizards that move components to be next to each other, but don't group them into a new assembly. These wizards are generally preferred for constructing objects for *EPIC* and *MEM* because they refrain from removing the surface elements that abut the neighboring object and would be interior to the assembly. The wizards appear on the **Wizards** menu and are "Element to Element Move," "Edge to Edge Move," "Edge to Element Move," and "Element to Edge Move."

15.11 Orbits

When building an object for *MEM*, *Object Toolkit* has an **Orbit** menu that allows the user to specify and edit a heliocentric or geocentric orbit. The dialog boxes are shown in Figure 39 and Figure 40, respectively. For heliocentric orbits, the semi-major axis is expressed in astronomical units. Orbits can be saved and retrieved from the files Orbits/HelioFixParamOrbits.xml and GeoCentricOrbits.xml.

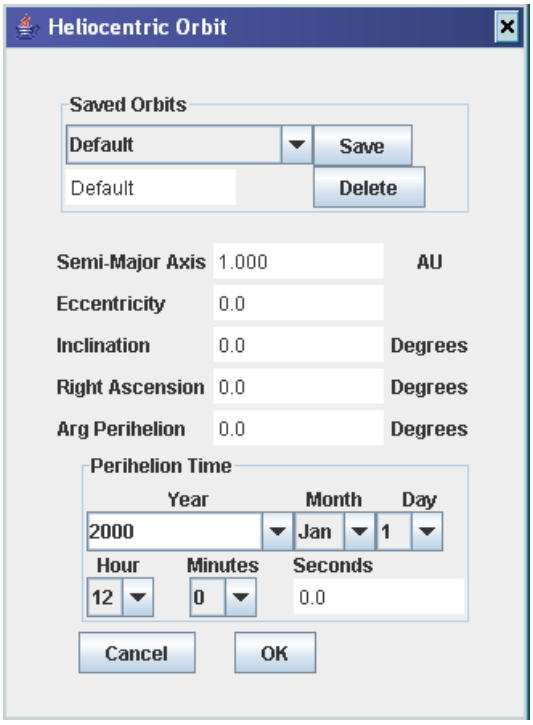


Figure 39. Heliocentric Orbit Dialog Box.

Approved for public release; distribution is unlimited.

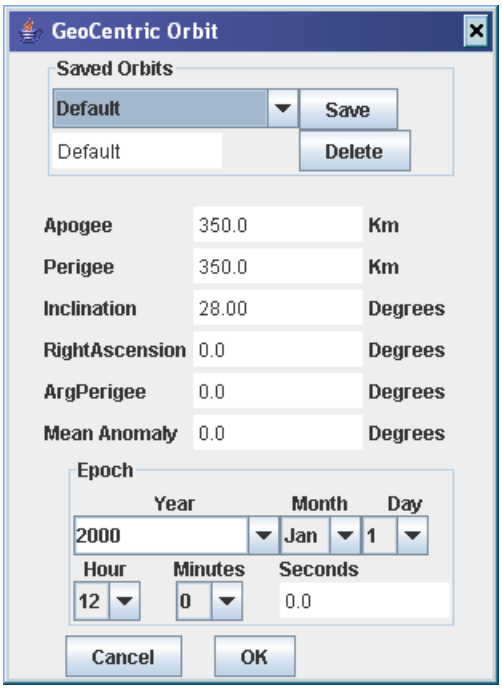


Figure 40. Geocentric Orbit Dialog Box.

Once an orbit has been defined, the time at which the orbit is to be displayed, the type of rotations to apply, and the type of stabilization are specified on the **Time and Stabilization** dialog box shown in Figure 41. While either "Pointing Rotations" or "Space and Pointing Rotations" is specified, *Object Toolkit* disables all editing features. "Pointing Rotations" are those in which a subsystem of the object moves with respect to the rest of the object to point at a target, such as solar arrays pointing at the sun. "Space Rotations" are those in which the entire object rotates.

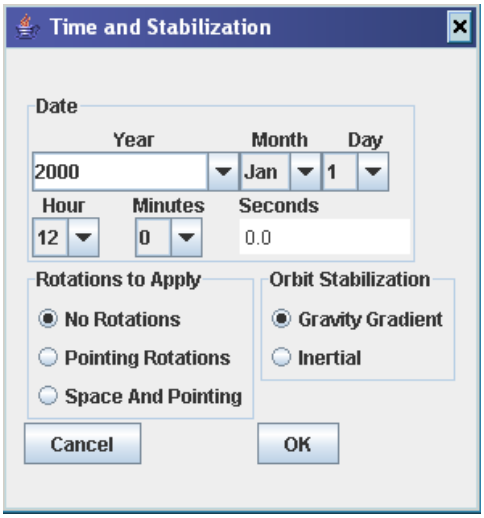


Figure 41. Time and Stabilization Dialog Box.

Approved for public release; distribution is unlimited.

The user-specified heliocentric orbits are Keplerian with no time perturbations and are calculated using *Astro-Tool Orbit*. Given the orbit parameters, including a time of perihelion, the code calculates the position for any other time using Kepler's law. Geocentric orbits are calculated by propagating from the instantaneous parameters that are specified in the **Geocentric Orbit** dialog box using the NORAD 8 propagator [2].

15.12 Materials

The default materials and their properties, along with default properties for new materials are specified in the *App*_OTKSpecs.xml file. If no materials are defined, default materials and material properties appropriate to *Nascap-2k* are used.

The object definition file produced by *Object Toolkit* contains the material properties for multiple applications. An object created in *Object Toolkit* tailored for one application can be read into *Object Toolkit* tailored for another application. When the object is saved to a file, it contains the material properties for both applications. By repeating this process, material properties for any number of applications can be incorporated into the object file.

16 Menus

Table 3 through Table 11 list the various items available on the *Object Toolkit* menus.

Table 3. File Menu.

Menu Item	Action
New Object…	Start building an object from scratch.
Open Object…	Open an existing *Object Toolkit* object file or a finite element model from standard finite-element preprocessor of known format. Elements must be meshed with linear triangles and linear quadrilaterals. (In some cases higher order elements are read as linear elements.) Material names and conductor numbers are *not* imported. (Except that for *NX I-DEAS* TMG files, the first four letters of the `RDFGROUPNAMES` are imported as material names.) *PATRAN* files, should have extension ".neu" or ".pat". *NX I-DEAS* TMG ASCII VUFF files, should have extension ".tmg". *Gmsh* MSH formatted files should have extension ".msh". Nastran BDF or DAT files should have extension ".bdf". I-DEAS universal files should have extension ".unv". STEP files should have extension ".stp".
Save Object	Save object.
Save Object As…	Save object to disk file with a specified name.
Import SEE Handbook Materials File…	Import material definitions from an XML file of the format saved by the *SEE Interactive Spacecraft Charging Handbook*. See Section 10.
Exit	Shut down *Object Toolkit*. Provides user with an option to save existing open object.

Approved for public release; distribution is unlimited.

Table 4. Edit Menu.

Menu Item	Action
Undo	Undo most recently executed command. Repeated Undos may be performed. A few complex operations cannot be undone.
Redo	Redo most recent undo action.
Cut	Copy selected component to clipboard and then delete original.
Copy	Copy selected component to clipboard leaving the original intact.
Paste	Paste component on clipboard into object.
Paste Special…	Paste component on clipboard into object. An opportunity to change any of its properties is provided before pasting.
Edit Component…	Edit properties of selected component.
Delete Component	Delete selected component.
Injection Point†	Not used, as injection points have no properties.
Emitter†	Set properties of emitter.
Detector†	Set properties of detector.
Subsystem*	Set properties of subsystem. See Section 15.6.
Thruster**	Edit properties of existing thruster or delete thruster. See Section 15.7.
Neutralizer†	Edit properties of existing neutralizer or delete neutralizer. See Section 15.8
MagDipole†	Edit properties of existing magnetic dipole or delete magnetic dipole. See Section 15.9.
Other Special Component‡	Edit properties of *special component*. See Appendix C.

**EPIC* and *MEM*.
***EPIC* and *Nascap-2k*.
†*Nascap-2k* only
‡Available if *Object Toolkit* has been tailored for a specific application by adding *special component*.

Approved for public release; distribution is unlimited.

Table 5. View Menu.

Menu Item	Action
Hide	Temporarily hide the selected component from display.
Hide Others	Temporarily display only the selected component.
Show All	Display all elements.
Material	Color elements by material name. Enable editing of element materials.
Conductor	Color elements by conductor number. Enable editing of element conductor numbers.
Injection Point†	Color elements by injection point name.
Emitter†	Color elements by emitter name.
Detector†	Color elements by detector name.
Subsystem*	Color elements by subsystem name.
Other Surface Property‡	Color elements by *surface property*. See Appendix C.
Select View	Reorient view so that object is seen from one of the coordinate axes or a user specified direction, or rescale view so that object fits comfortably within the 3-D view area.
Set Background Color	Set the background color of the 3-D view area to black or white.
Set Outline Color	Set the color of the lines outlining individual elements to black or white. Also used to eliminate element outlining (choice none).
Perspective	Turn perspective view on or off. (On by default.)
Mouse Position	Bring up a small window that displays the present mouse position. This is useful for quick determinations of component size and location. Turns off perspective.
Legend Font Size	Increase or decrease font size used in legend.

**EPIC* and *MEM*.
†*Nascap-2k* only.
‡ Available if *Object Toolkit* has been tailored for a specific application by adding *surface property*.

Approved for public release; distribution is unlimited.

Table 6. Component Menu.

Menu Item	Action
Add New	Add a new predefined component: Dish, Panel, Cylinder, Box, Boom, or Thruster**. See Section 6.
Add Component from File…	Add the object contained in the specified *Object Toolkit* XML file, *PATRAN* file, *NX I-DEAS* TMG ASCII VUFF file, *Gmsh* MSH formatted file, Nastran BDF or DAT file, *I-DEAS* universal file, or STEP file to the current object as a single assembly. The original location and orientation are retained during this operation. See Section 12.
Node Relative Move…	Move the component containing the selected node to a new location (or the location of a second selected node) so that the selected node is at the specified location. See Section 7.
Align Edge…	Reorient the component containing the selected edge so that the selected edge is oriented in the specified direction (or the direction of a second selected edge). See Section 7.
Center at Origin	Move the selected component so that the center of its bounding box is located at (0, 0, 0). Also moves all special components by the same amount.
Place	Move the selected component so that the center of its bounding box is located at the specified location.
Enable Tools	Enable or disable the move component buttons on the tool bar.
Assign Injection Point†	Assign selected injection point name to elements of the selected component. If "Other" is chosen, create a new injection point name and assign it to the selected elements. See Section 15.5.
Assign Emitter†	Assign selected emitter name to elements of the selected component. If "Other" is chosen, create a new emitter name and assign it to the selected elements. See Section 15.4.
Assign Detector†	Assign selected detector name to elements of the selected component. If "Other" is chosen, create a new detector name and assign it to the selected elements. See Section 15.4.
Assign Subsystem*	Assign elements of selected component to subsystem with the name chosen from the menu. If "Other" is chosen, create a new subsystem and assign selected elements to it. See Section 15.6.
Enter Assembly	View and edit selected component as it exists one level down in the assembly hierarchy. Caution must be exercised when editing components at any level except the top level to ensure that surface elements continue to match up in the same way after the size of any component is changed. Mysteriously distorted elements and mysteriously changed materials can be caused by elements no longer matching up as intended. See Section 14.
Leave Assembly	Move one level up in the assembly hierarchy. See Section 14.
Top Assembly	View and edit object at the top level of the assembly hierarchy. See Section 14.
Consolidate Components in Assembly	Combine all components, excluding special components, into a single assembly. See Section 8.
Convert Component to Primitive	Convert selected component into a Primitive. Because Primitives consist of only nodes and elements, this operation destroys the instructions used to build the component. Primitives are the only components that can be scaled. Converting complex objects into Primitives speeds up code operations. See Section 13.3.

**EPIC* and *MEM*.
***EPIC* and *Nascap-2k*.
†*Nascap-2k* only.

Approved for public release; distribution is unlimited.

Table 7. Mesh Menu.

Menu Item	Action
Materials	Change material of selected elements. Active when Materials are viewed. See Section 10.
Conductors	Change conductor number of selected elements. Active when conductors are viewed. See Section 15.1.
Injection Point†	Assign selected injection point name to selected elements. If "Other" is chosen, create a new injection point name and assign it to the selected elements. See Section 15.5.
Emitter†	Assign selected emitter name to selected elements. If "Other" is chosen, create a new emitter name and assign it to the selected elements. See Section 15.4.
Detector†	Assign selected detector name to selected elements. If "Other" is chosen, create a new detector name and assign it to the selected elements. See Section 15.4.
Subsystem*	Assign selected elements to subsystem with the name chosen from the menu. If "Other" is chosen, create a new subsystem and assign selected elements to it. See Section 15.6.
Create Element	Create an element from two selected edges, four selected nodes, or three selected nodes. The edges or nodes must be in the same component. Available only if the appropriate number of edges or nodes has been selected. See Section 11.2.
Delete Element	Delete the selected element. Available if and only if at least one element is selected. See Section 11.2.
Divide Element	If the selected element is a quadrilateral, divide it into four quadrilaterals. If the selected element is a triangle, divide it into four triangles. Available if and only if exactly one element is selected. See Section 11.
Combine Triangles	Combine the two selected triangles into a single quadrilateral. Available only if two triangles have been selected. The two triangles must be in the same assembly and share an edge. See Section 11.5.
Reverse Elements	Reverse the outward normal of all the selected elements. See Section 13.1.
Edit Node…	Move the selected node. The location is specified in the coordinate system of the assembly. See Section 13.2.
Combine Nodes	Move second selected node to location of first selected node. The two nodes must be in the same assembly. See Section 11.4.
Conductivity	Specify that the selected node or edge is to be a grounding element in *Nascap-2k*. To ground the material, the user can specify grounding by a strip at an element edge or by a circular contact with user-specified radius located at a node. These grounding elements can be specified only for Primitive components. See Section 15.3.
Rebuild Mesh	Rebuild the mesh of the selected component from the bottom of the assembly hierarchy. Ideally, this menu item should never be needed; however, if the object suddenly looks very strange (misshapen surfaces, mixed-up surface attributes, or misplaced nodes), it can sometimes be helpful. In particular, if nodes that have been moved using the "Edit Node" command mysteriously return to their original positions, a "Rebuild Mesh" operation may correct the problem. The entire mesh is automatically rebuilt every time the code is restarted.
Set Tolerance	Specify the minimum distance between nodes of the same assembly. Any two nodes that are in same component and are within the specified tolerance of each other are automatically combined.

*EPIC and MEM only.
†*Nascap-2k* only.

Approved for public release; distribution is unlimited.

Table 8. Wizard Menu.

Menu Item	Action
Element to Element Wizard…	Join a quadrilateral element to the center of a larger quadrilateral element. Its most common use is when the smaller quadrilateral element is the end of a Boom. The second component chosen is moved and reoriented so that the centers of the two elements are coincident. A new component that consists of one element on one side and five on the other, which exactly match with the original elements, is created. The original two components and the new component are combined into a single assembly. The abutting, matching elements are deleted. See Section 9.
Edge to Element Wizard and Element to Edge Wizard…	Join a small quadrilateral element (like the end of a Boom) to the edge of a larger element. Attaching a Boom to a solar panel is a common application. The second component chosen is moved and reoriented so that the centers of the selected element and the selected edge are coincident. Joining the two components requires "opening" the edge to make an attachment element and dividing the elements neighboring the edge. The two elements adjoining the edge are converted to three quadrilaterals and six triangles. The two components are combined into a single assembly. Which component is moved is the only difference between the two wizards. See Section 9.
Edge to Edge Wizard…	Join specified edges of two components and separate the resulting elements to avoid zero-thickness problems. This is commonly used to attach a dish antenna. The second component chosen is moved and reoriented so that the centers of the two edges are coincident. The end nodes of the longer edge are moved so that the two edges have the same length. This move *can* distort the object shape. A new node is added at the midpoint of the shared edge and the four adjoining elements are each divided into a quadrilateral and a triangle. The new node is split into two 0.05 units apart to provide a thickness to the joint. The two components are combined into a single assembly. See Section 9.
Surface to Surface Wizard…	Place one flat surface on another and attempt to correct the meshing of the resulting object. The two components are combined into a single assembly. This wizard should be used with caution, because the resulting meshing is generally non-optimal. The object should always be saved before using this wizard. See Section 9.
Solar Array Wizard…	Assign conductor numbers to series of surface elements so that they increase sequentially along the user-specified path, as might be desired when simulating a string of solar cells.
Element to Element Move…*	Place a quadrilateral element at the center of a larger quadrilateral element. The second component chosen is moved and reoriented so that the centers of the two elements are coincident. See Section 15.8.
Edge to Edge Move…*	Place specified edges of two components next to each other. The second component chosen is moved and reoriented so that the centers of the two edges are coincident. See Section 15.8.
Edge to Element Move and Element to Edge Move…*	Place a small quadrilateral element (like the end of a Boom) at the edge of some larger element. Moving a Boom to a solar panel is a common application. The second component chosen is moved and reoriented so that the centers of the selected element and the selected edge are coincident. Which component is moved is the only difference between the two wizards. See Section 15.8.

**EPIC* and *MEM* only. These wizards create incompatible meshing.

Table 9. Material Menu.

Menu Item	Action
New…	Create a new material name and associated properties.
Edit	Modify the material properties associated with the material name.
Delete	Delete existing material. There is no change to any existing elements of the deleted material, but the material no longer appears in any of the **Material** pull-down menus and has no properties.

Table 10. Orbit* Menu.

Menu Item	Action
Heliocentric	Define a heliocentric orbit. See Section 15.11.
Geocentric	Define a geocentric orbit. See Section 15.11.
Time and Stabilization	Specify various parameters regarding the orientation of tracking components included in the object. These parameters include the time at which the orientation of tracking components is to be displayed, the type of stabilization, and the rotations that are to be displayed. See Section 15.11.

**MEM* only.

Table 11. Help Menu.

Menu Item	Action
Object Toolkit Help…	Display help file.
About…	Display version number and contact information.

Approved for public release; distribution is unlimited.

17 Quick Start

17.1 Create a Box

To create a Box, go to the **Component** menu, select "Add New" and choose "Box" from the fly-out menu. Make this Box 3.0 m by 4.0 m by 5.0 m and five elements in each direction, as shown in Figure 42. Make the ±Y sides material OSR and the other sides Black Kapton.

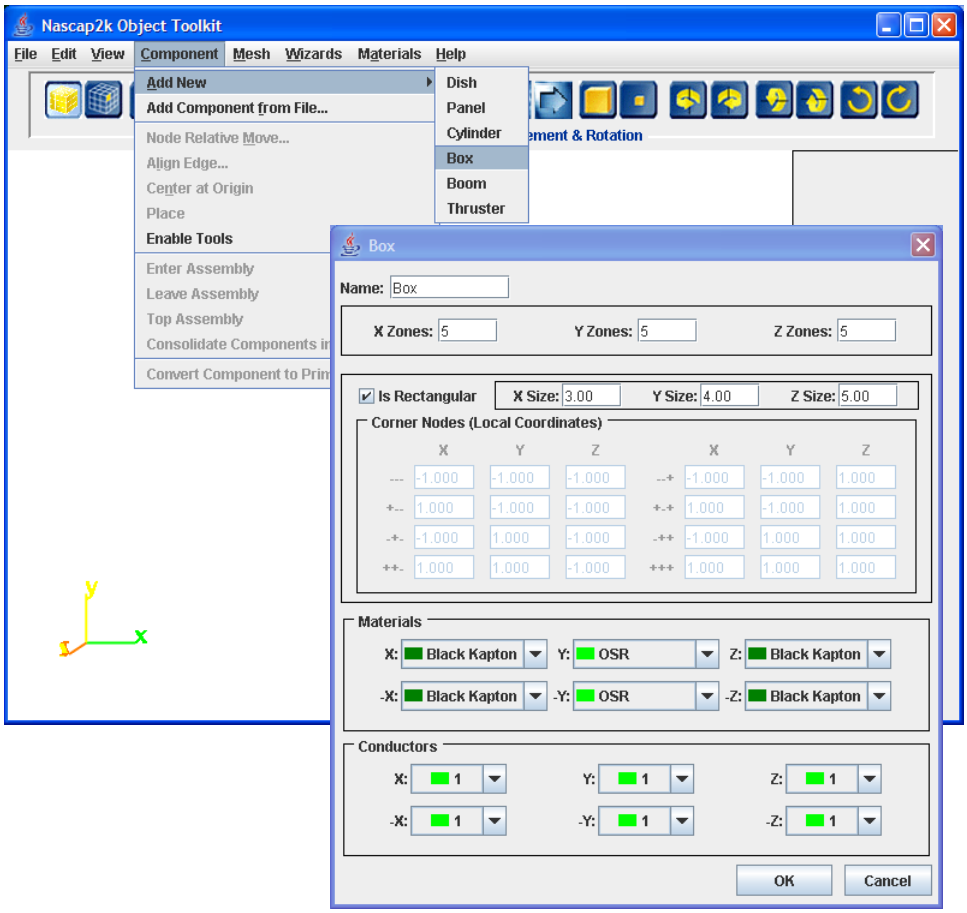


Figure 42. Dialog Box for Creating a Box 3.0 m by 4.0 m by 5.0 m in Size and Five Elements in Each Direction. The ±Y Sides Are Defined to Be Material OSR and the Other Sides Are Black Kapton.

Approved for public release; distribution is unlimited.

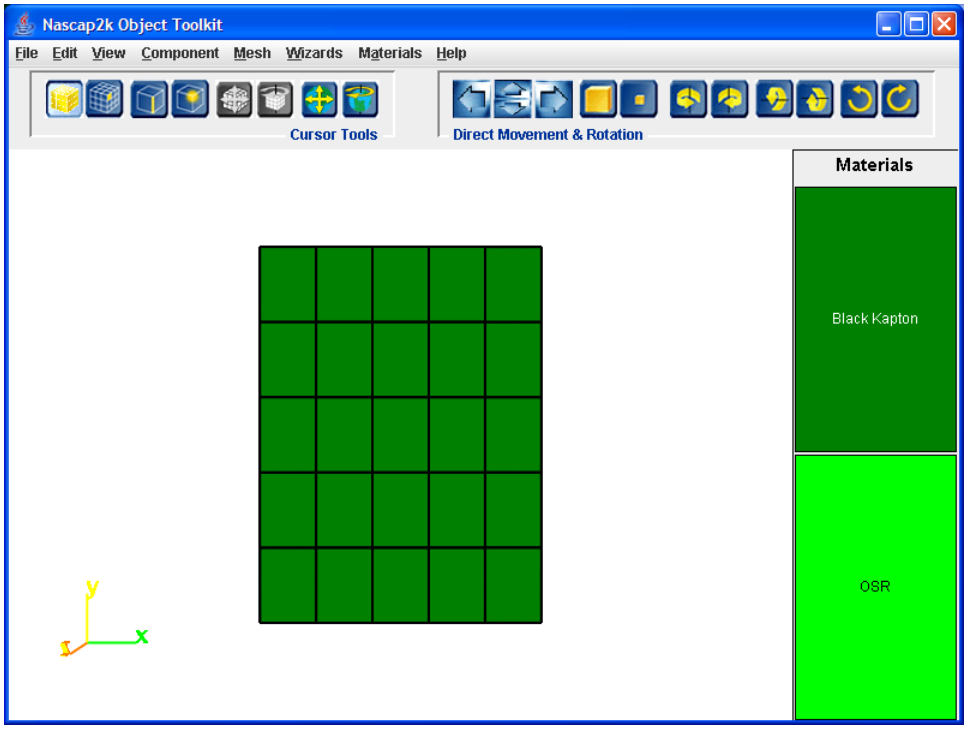


Figure 43. The Box Created from the Definition in Figure 42.

17.2 Rotate the View

Click the right-most cursor tool button and rotate the view of the Box using the mouse.

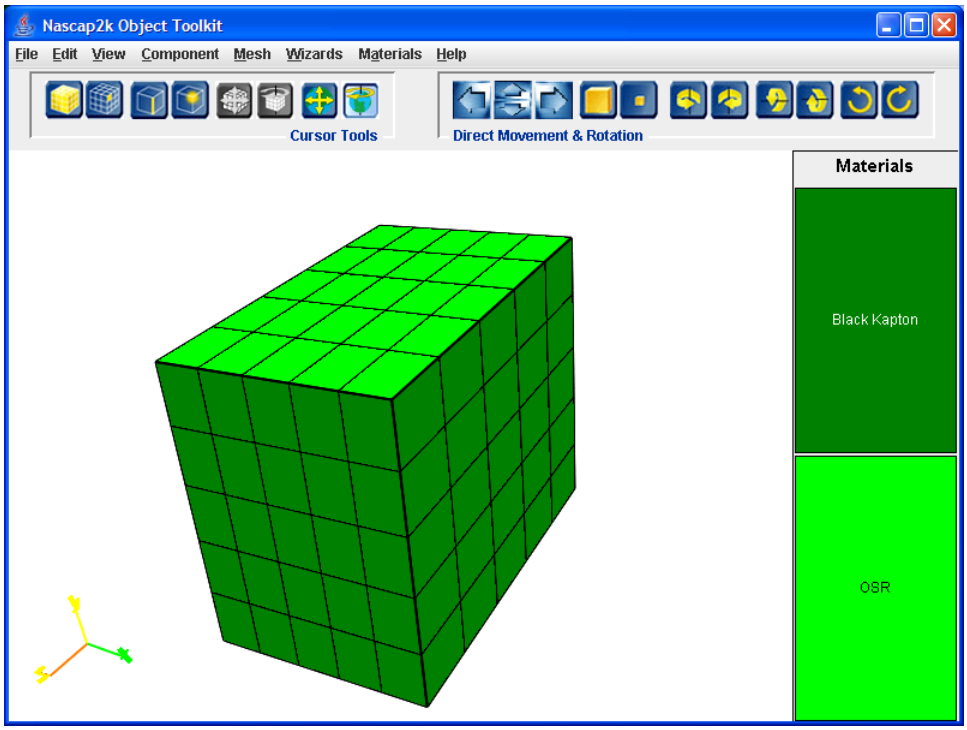


Figure 44. Rotated View of the Created Box.

Approved for public release; distribution is unlimited.

17.3 Change the Dimensions and Zoning

Click the left-most cursor tool button to place the mouse in component selection mode. Then select the Box by clicking on it. Go to the **Edit** menu, and select "Edit Component." This brings up the **Box** definition dialog box shown in Figure 45. Change the number of elements in the Y direction to four. Change the distance in the Y direction to be 2.0 m. The result is shown in Figure 46.

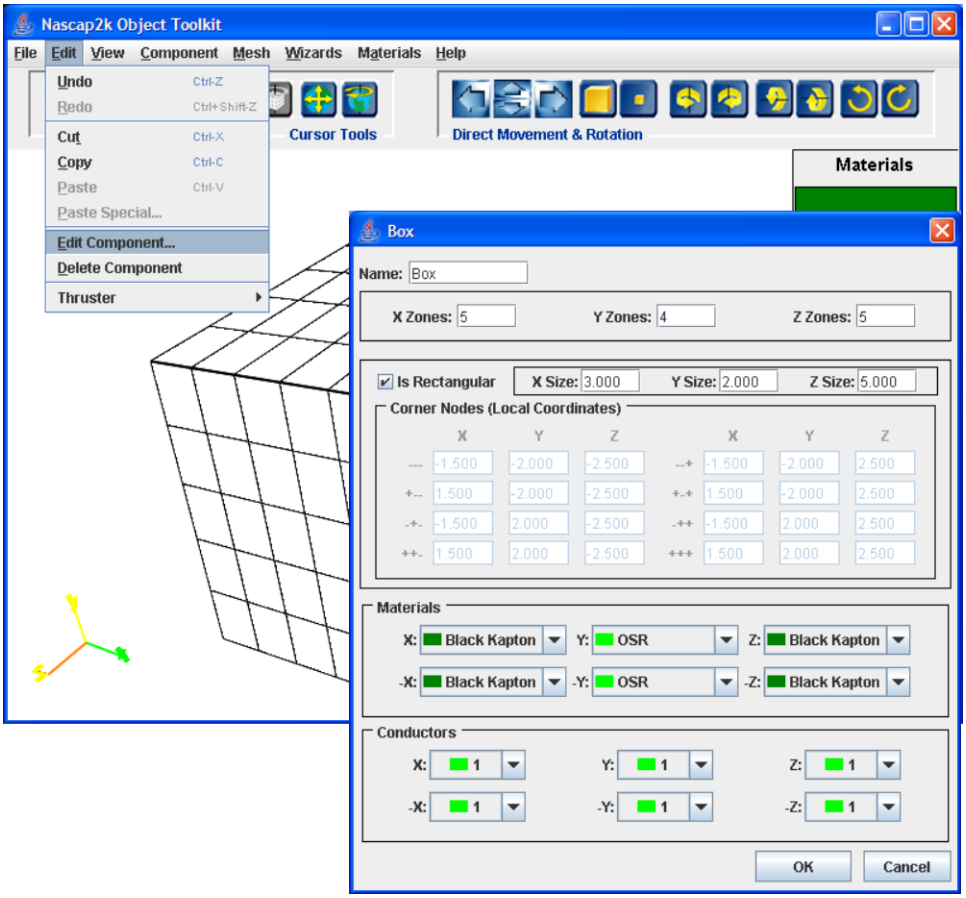


Figure 45. Editing the Box to Reduce Its Size and Number of Elements in the Y Direction.

Approved for public release; distribution is unlimited.

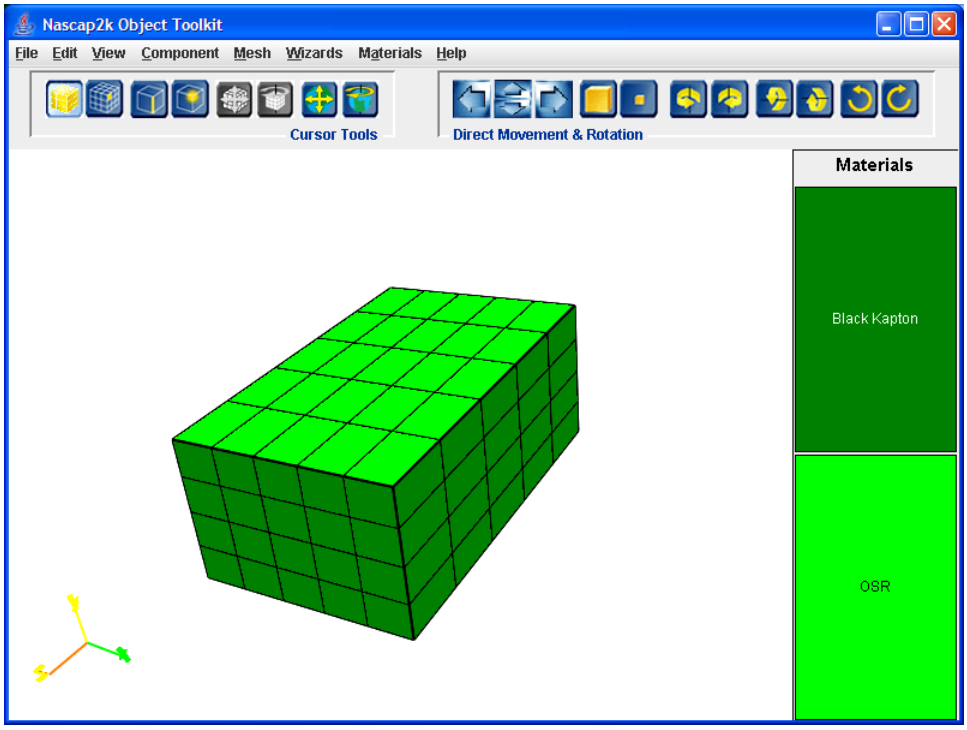


Figure 46. Box Resulting from Modifications Shown in Figure 45.

17.4 Change Surface Material

Click the second cursor tool button from the left to put the mouse in select element mode. Select several elements on the +X side of the Box as shown in Figure 47 (the CTRL key allows for multiple selections). On the **Mesh** menu, select "Materials" and choose "Solar Cells" from the fly-out menu to obtain the result in Figure 48.

Approved for public release; distribution is unlimited.

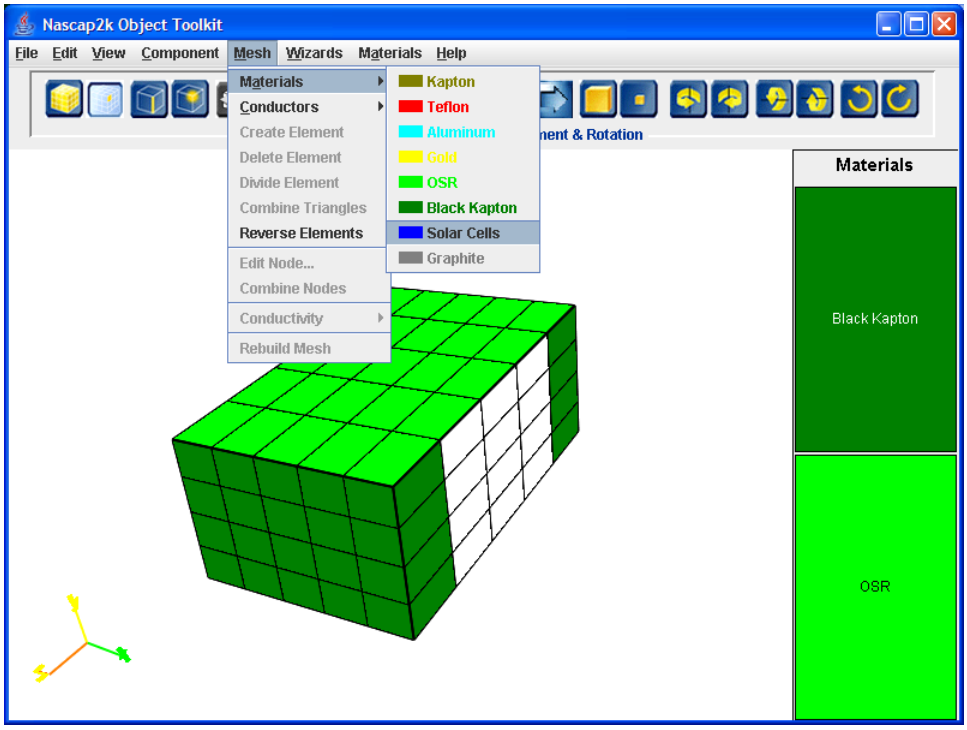


Figure 47. Box with Elements Selected.

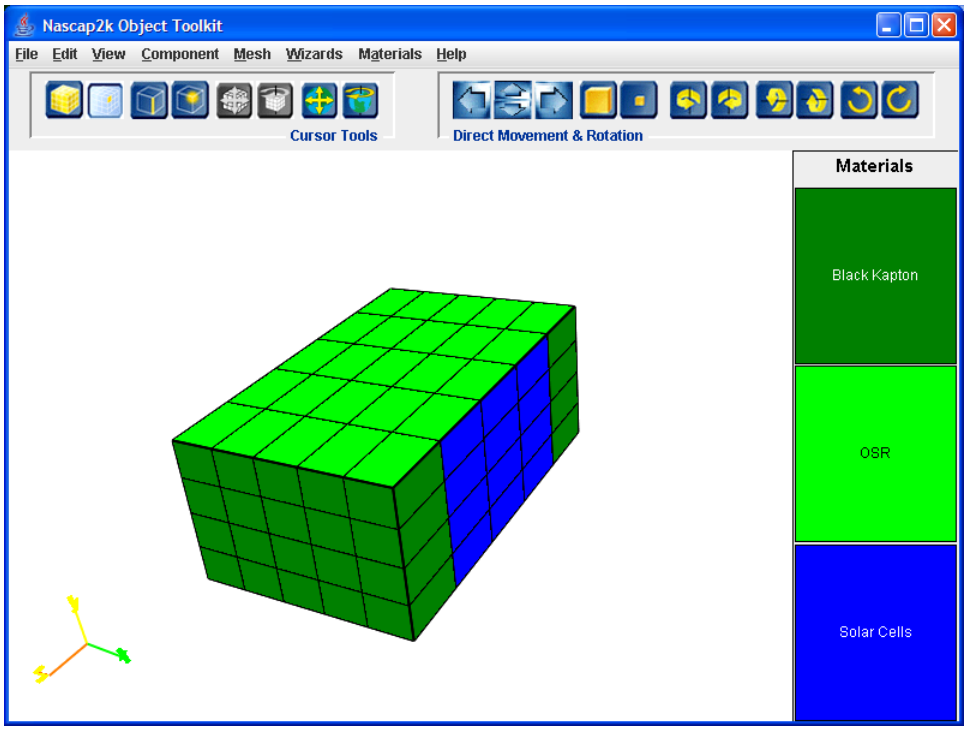


Figure 48. Box with a Different Material on Selected Elements.

Approved for public release; distribution is unlimited.

17.5 Save and Reopen Object File

On the **File** menu, select "Save." Navigate to the desired folder and save the object definition file to the desired name as shown in Figure 49. Close *Object Toolkit* and restart it. On the **File** menu, select "Open" to open the previously saved object file as shown in Figure 50 and Figure 51. Use the mouse to rotate the view of the Box as illustrated in Figure 52.

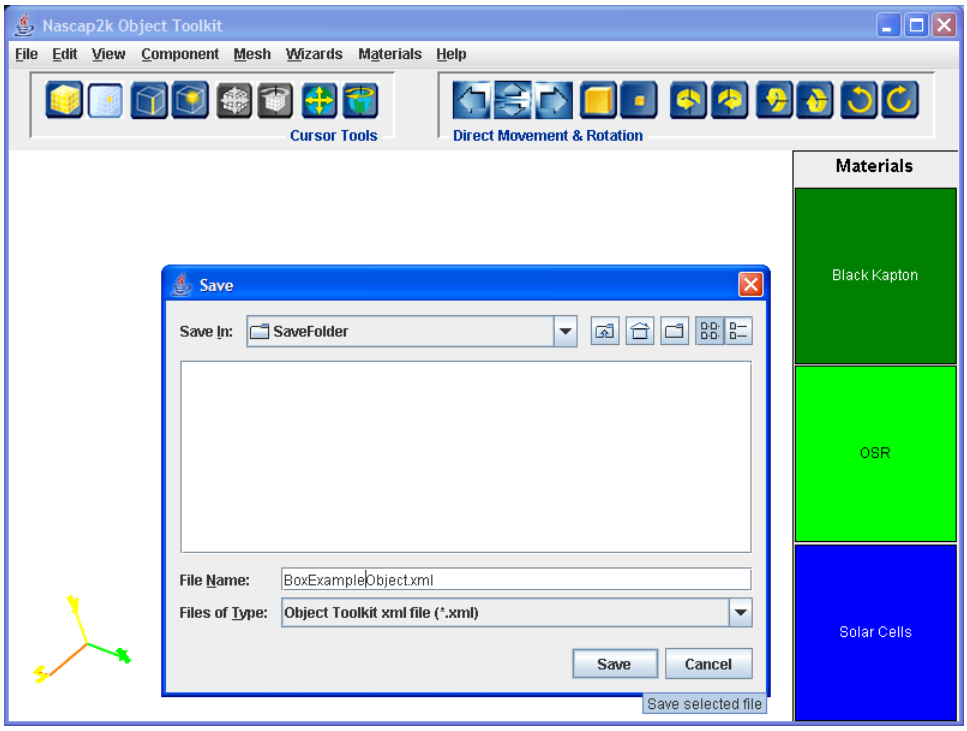


Figure 49. Save As Dialog Box.

Approved for public release; distribution is unlimited.

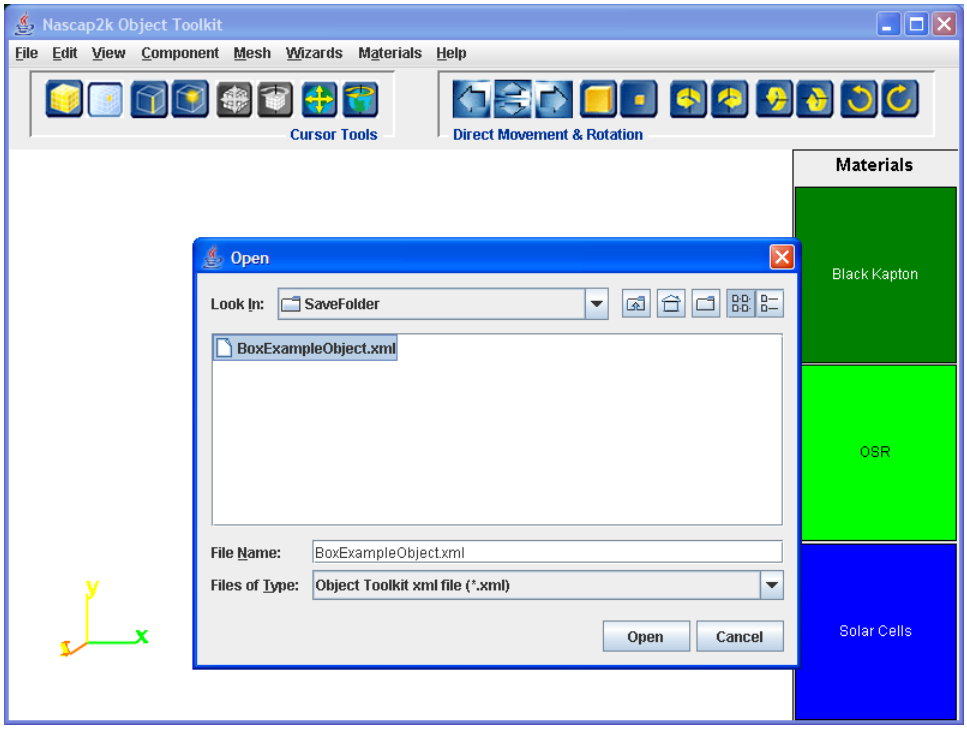


Figure 50. Open Dialog Box.

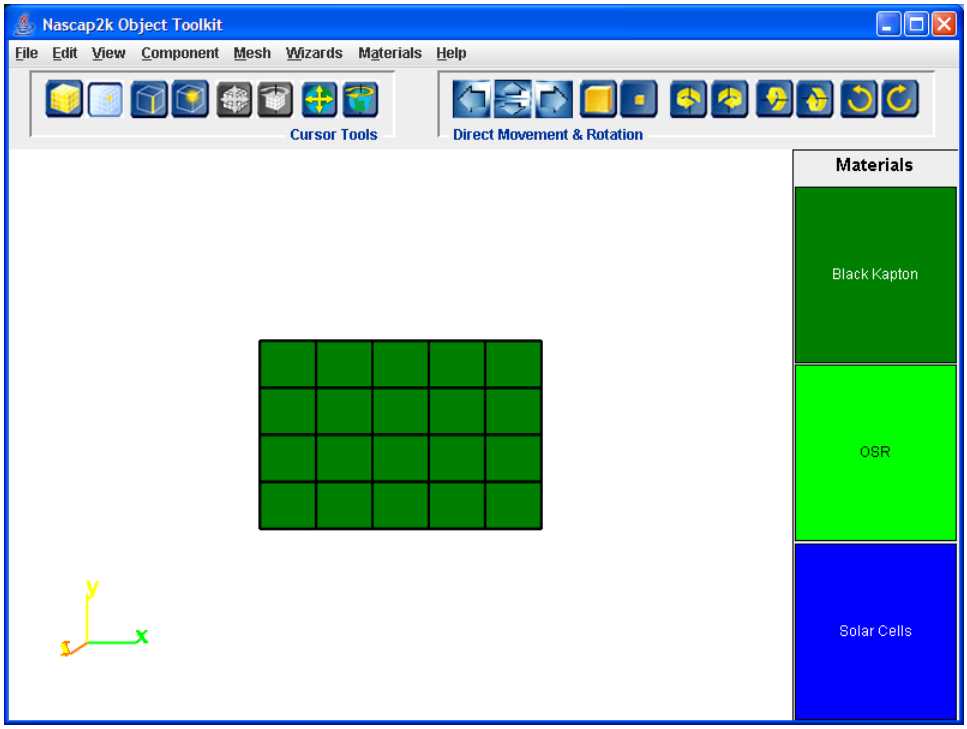


Figure 51. Box after Reopening.

Approved for public release; distribution is unlimited.

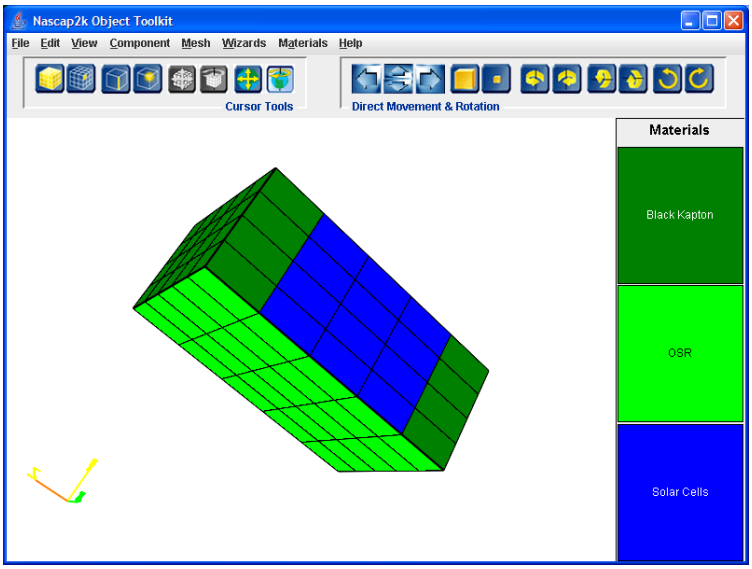


Figure 52. Box Viewed from a Different Direction.

18 Typical Geosynchronous Spacecraft for *Nascap-2k*

Figure 53 shows an *Object Toolkit* model of a typical geosynchronous spacecraft. It consists of a Box to represent the body, two Panels to represent the solar arrays, two Booms to support the solar arrays, two Dishes to represent antennas, and a Cylinder on the earth-facing surface of the body to represent an omni-antenna.

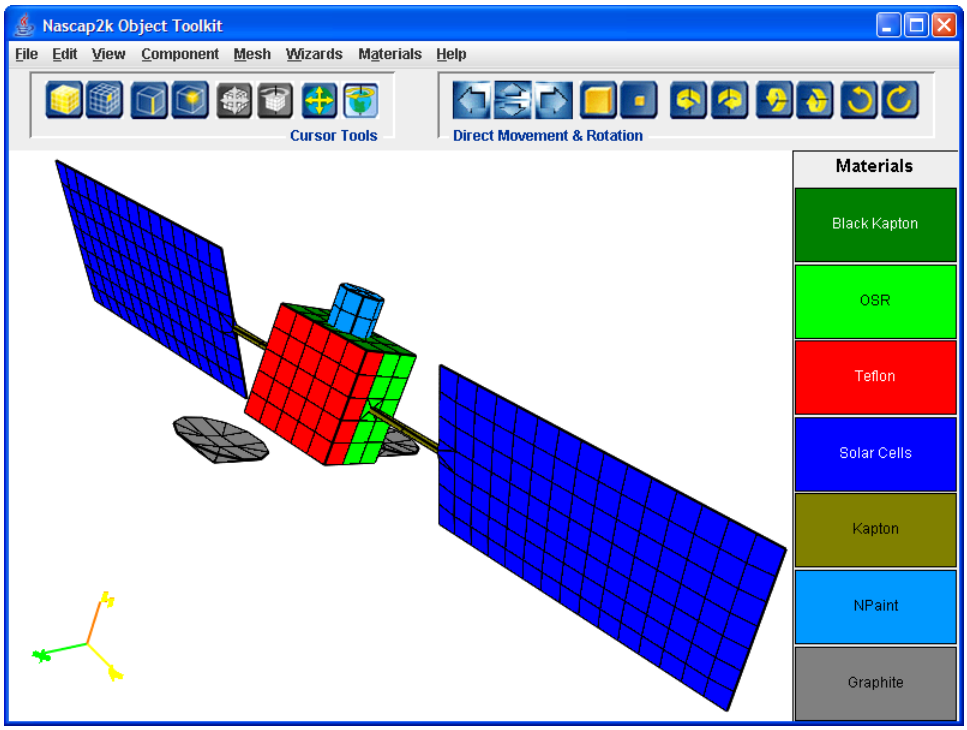


Figure 53. *Object Toolkit* Model of a Typical Geosynchronous Spacecraft to Be Built in this Example.

Approved for public release; distribution is unlimited.

18.1 Create the Body

To create a Box to represent the spacecraft body, go to the **Component** Menu, select "Add New" and choose "Box" from the fly-out menu. Make this Box 1.76 m by 2.64 m by 2.64 m, with three elements in the X direction and five in the Y and Z directions. Make the ±X sides Teflon, the ±Y sides OSR, and the ±Z sides Black Kapton, as shown in Figure 54.

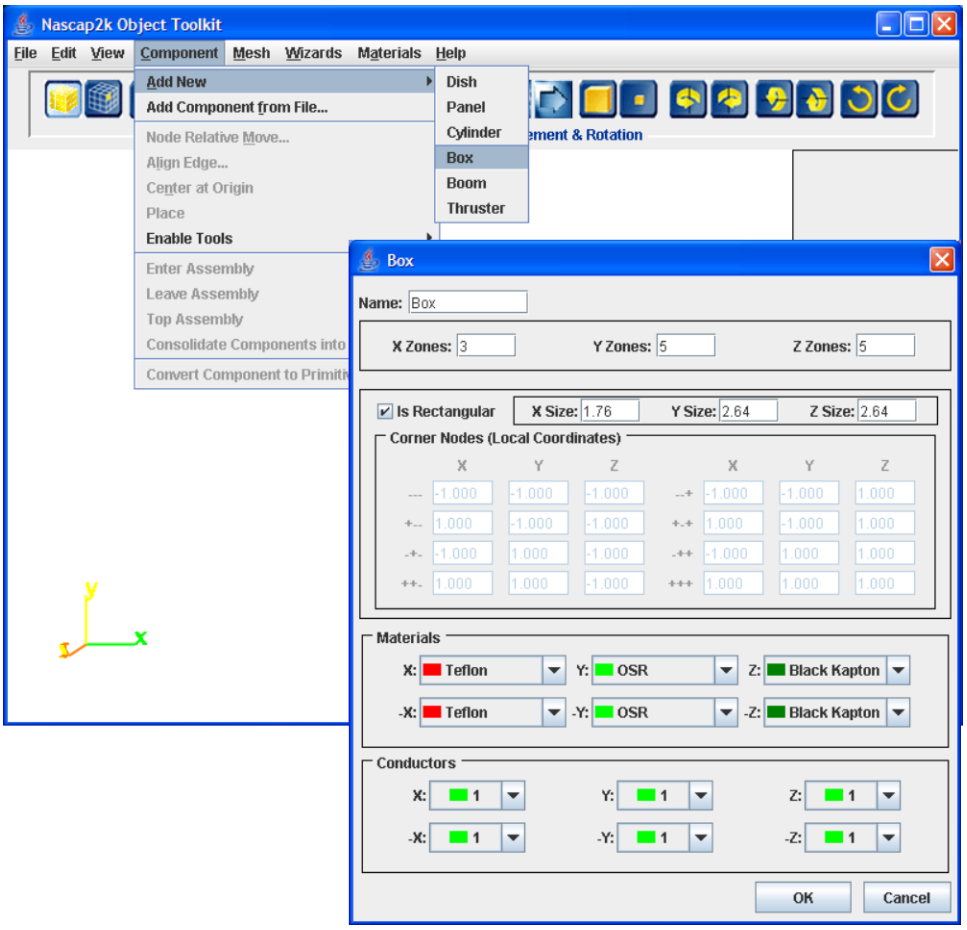


Figure 54. Defining a Box to Form the Body of the Spacecraft. The Box is 1.76 m by 2.64 m by 2.64 m in Size. Three Elements in the X Direction, and Five Elements in Each of the Other Two Directions. The ±X Sides Are Teflon. The ±Y Sides Are OSR. The ±Z Sides Are Black Kapton.

Change surface materials so that the five elements along the +X edge of the north and south sides (±Y sides) of the body are Teflon and are as shown in Figure 55.

Approved for public release; distribution is unlimited.

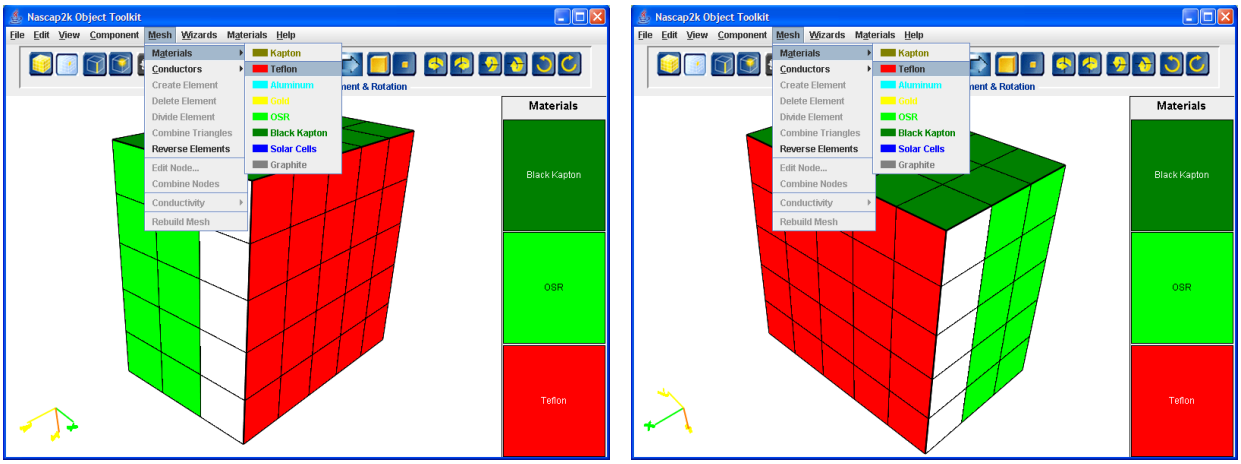


Figure 55. Changing the Materials on the Body Surfaces to Be as Desired.

Then save the result.

18.2 Create and Add the Omni-Antenna

Create a new material to represent the nonconducting paint used on the omni-antenna by going to the **Material** menu and selecting "New." Assign a name, color for display, and values for the material properties, as shown in Figure 56.

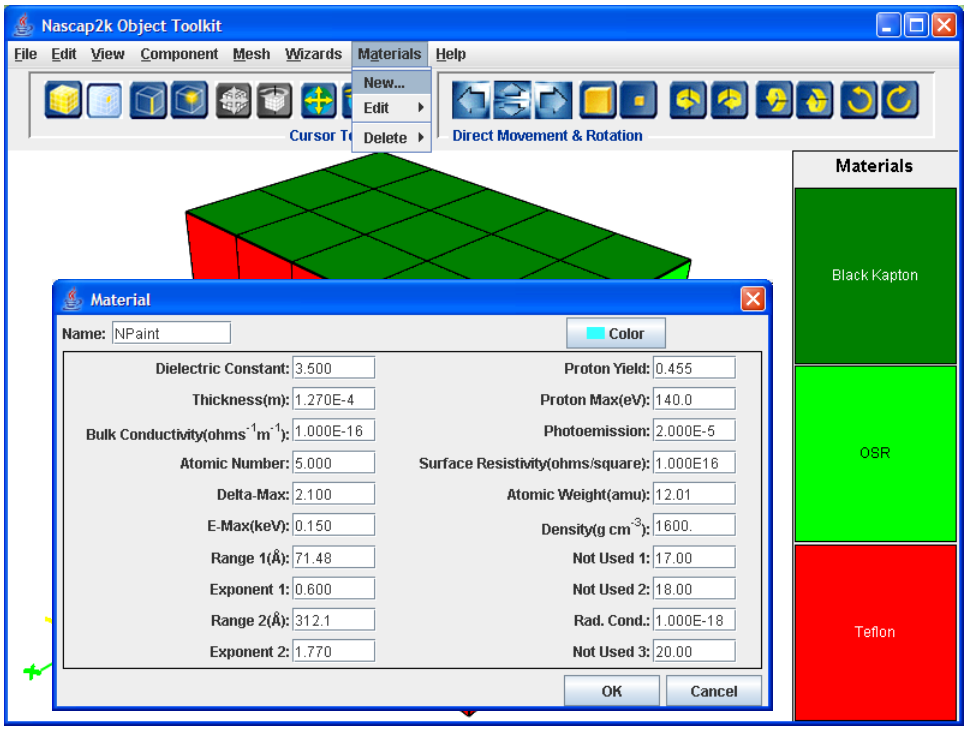


Figure 56. Defining a New Material, NPaint, Which Is Used to Cover the Omni-antenna.

Create a Cylinder to represent the omni-antenna by going to the **Component** menu, selecting "Add New," and choosing "Cylinder" from the fly-out menu. Make this Cylinder 0.88 m in

Approved for public release; distribution is unlimited.

height and 0.99 m in diameter, two elements high, and eight elements in circumference. Assign the material NPaint to all the surfaces, as shown in Figure 57.

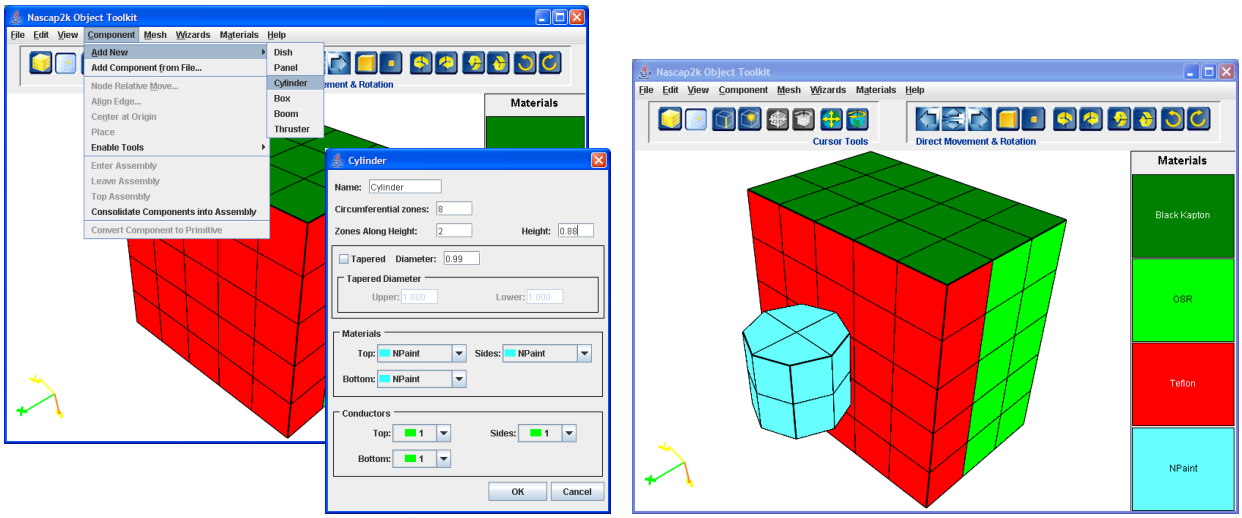


Figure 57. Defining a Cylinder to Represent the Omni-antenna. The Cylinder is 0.88 m in Height, 0.99 m in Diameter, Two Elements High, and Eight Elements in Circumference. All the Elements Are of the Newly Defined Material NPaint.

Attaching the omni-antenna to the body is a multistep process. Begin by deleting the elements along the –Z side of the Cylinder and the center five elements on the +Z side of the body, as shown in Figure 58. Then move the omni-antenna so that the bottom of the omni-antenna is in the same plane as the +Z side of the body and the centerline of the omni-antenna is the same as that of the body (which happens to be centered), as seen in Figure 59. Use the "Node Relative Move" menu item on the **Component** menu with a "Cancel" to determine the location of the top surface of the body.

Approved for public release; distribution is unlimited.

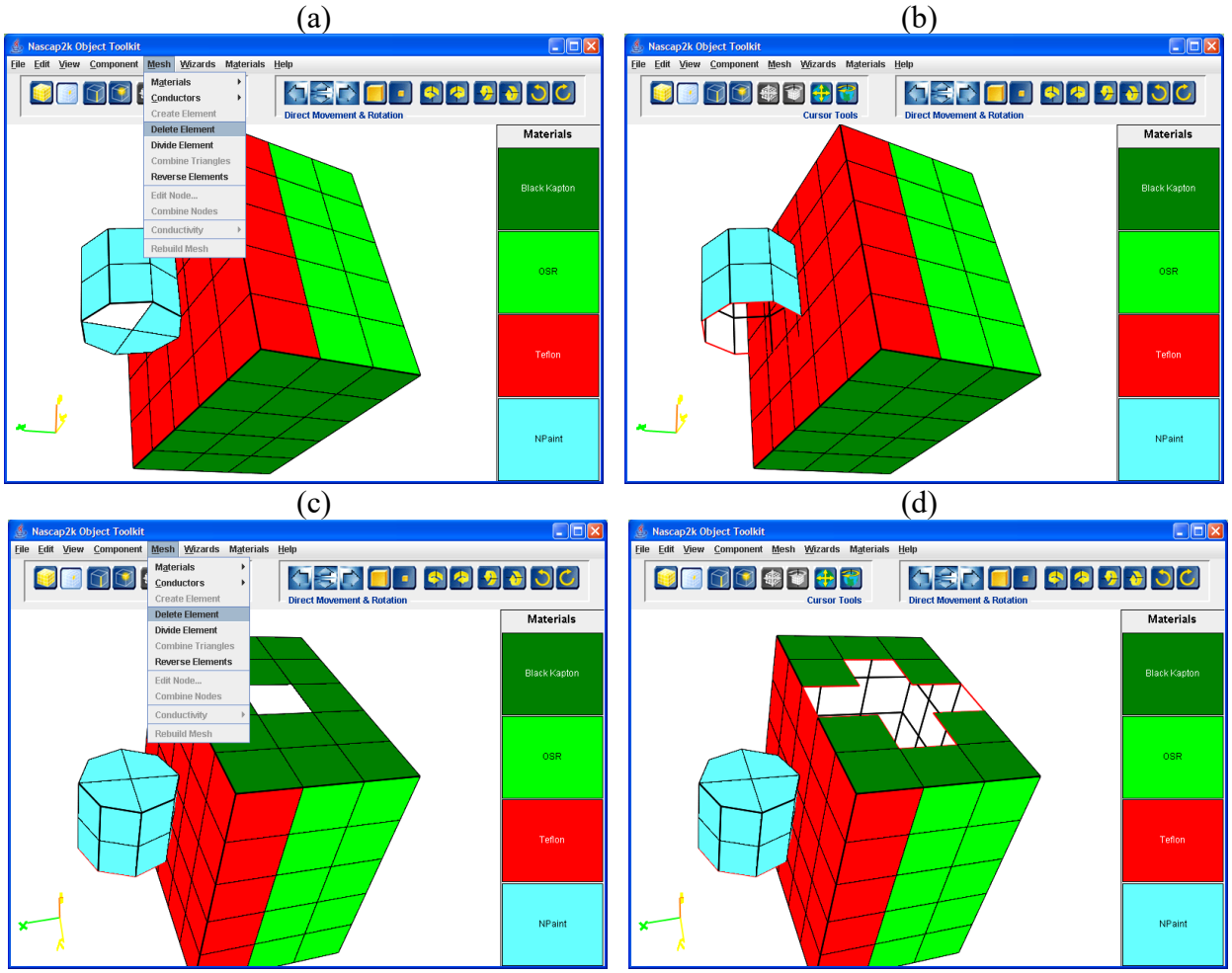


Figure 58. Deleting the Bottom Surface of the Cylinder and Five Elements on the Top of the Body So That the Two Can Be Joined. The Outlines of Elements with Normals Facing into the Screen Are Seen.

Approved for public release; distribution is unlimited.

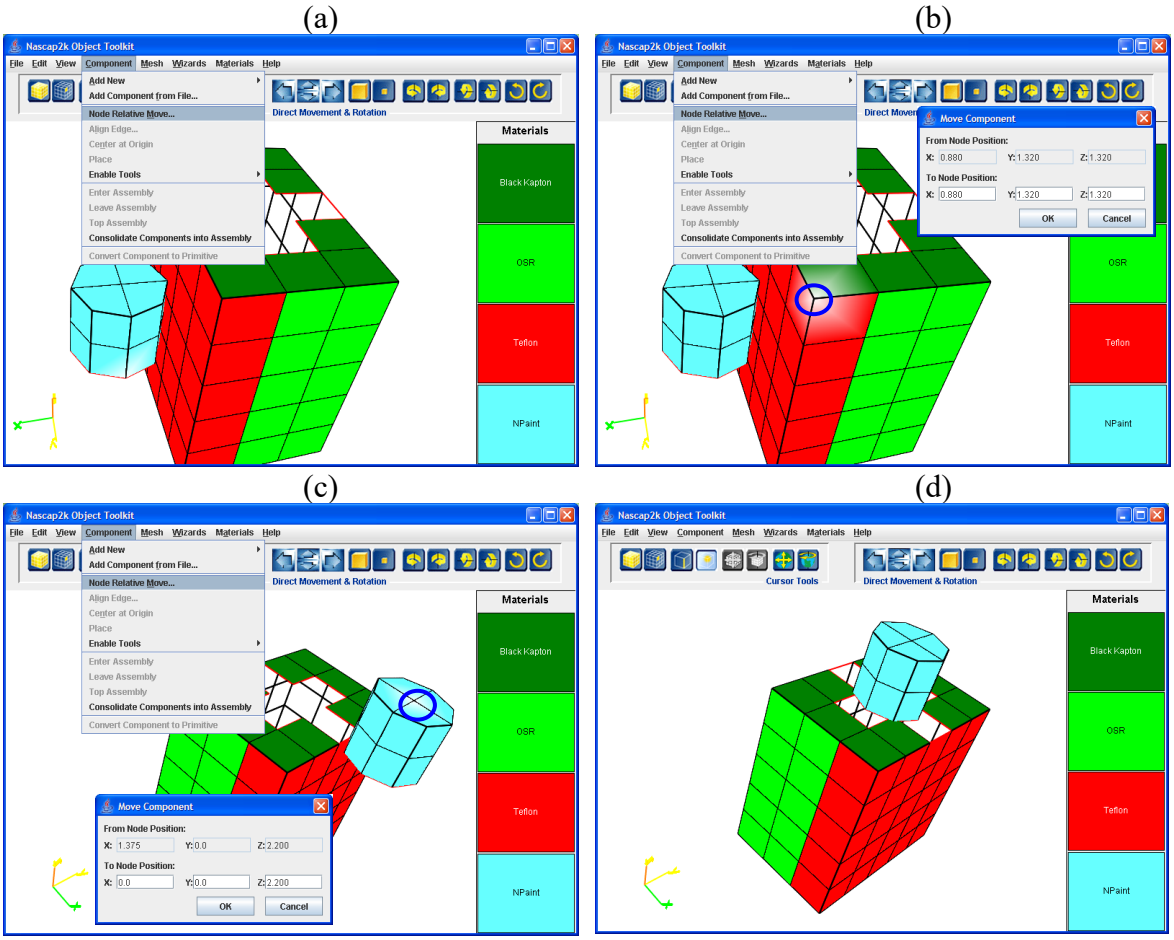


Figure 59. Placing the Cylinder So That Its Bottom Surface Is in the Plane of the Top of the Body (Z = 1.32) and the Center Lines Coincide (X = 0 and Y = 0). The Outlines of Elements with Normals Facing into the Screen Are Seen.

Before joining the components, save the results. Because some of the edges have an element on only one side, the error message seen in Figure 60 is generated.

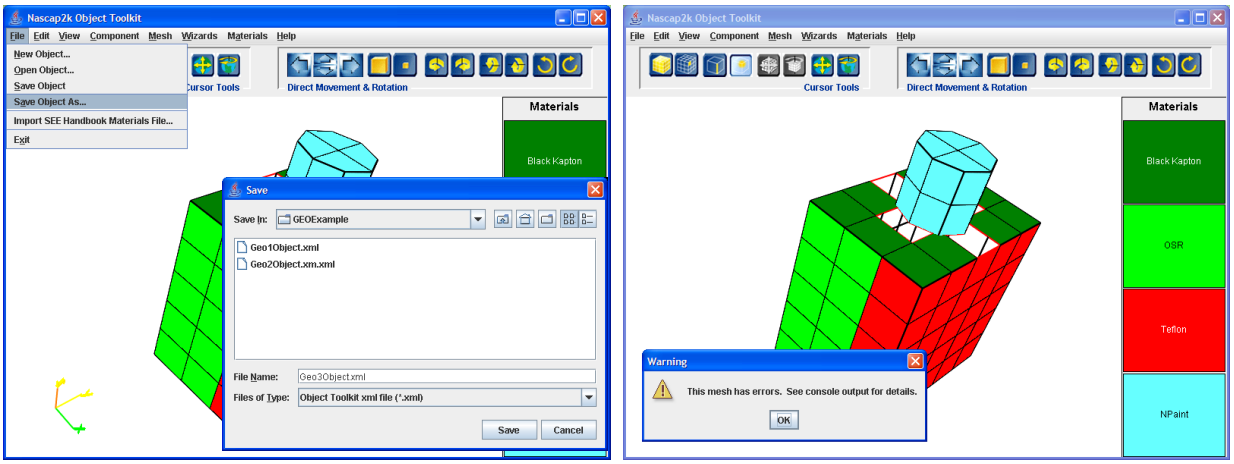


Figure 60. Saving the Resulting Object, Which Has Errors in It. The Outlines of Elements with Normals Facing into the Screen Are Seen.

Approved for public release; distribution is unlimited.

There are four nodes of the body that are just inside the Cylinder. Move these nodes to be coincident with the nodes on the Cylinder, as shown in Figure 61. Begin by grouping the two parts into an assembly. Select the node on the corner of the Cylinder, then select the node on the body, and finally go to the **Mesh** menu and select "Combine Nodes." Repeat for all four corners.

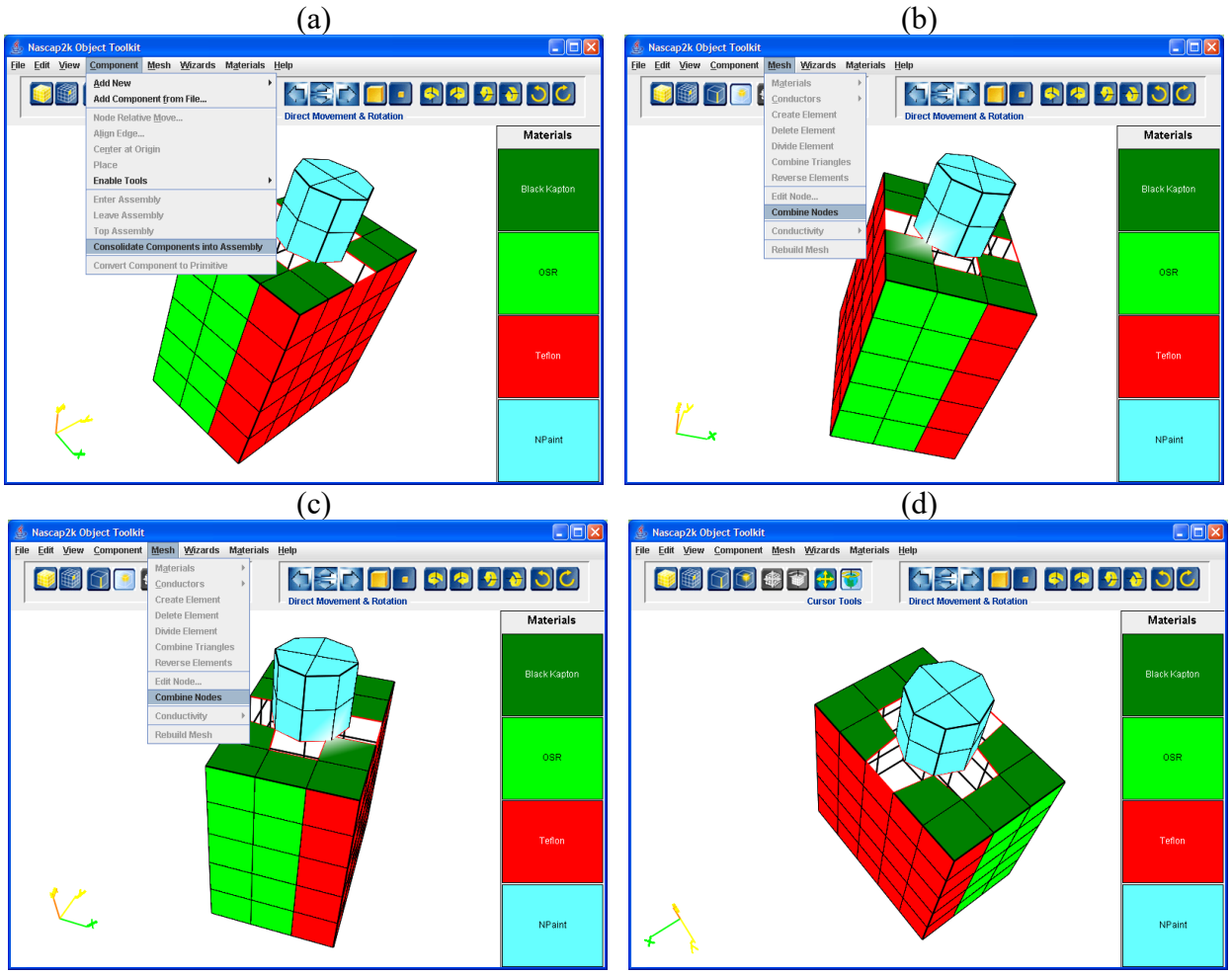


Figure 61. Moving the Nodes of the Body That Are Near the Nodes of the Cylinder to Be Coincident with the Nodes of the Cylinder. The Outlines of Elements with Normals Facing into the Screen Are Seen.

Save the result.

Add elements to connect the omni-antenna and the body as shown in Figure 62. Then change the material on the new elements to Black Kapton as shown in Figure 63. Finally, save the resulting object.

Approved for public release; distribution is unlimited.

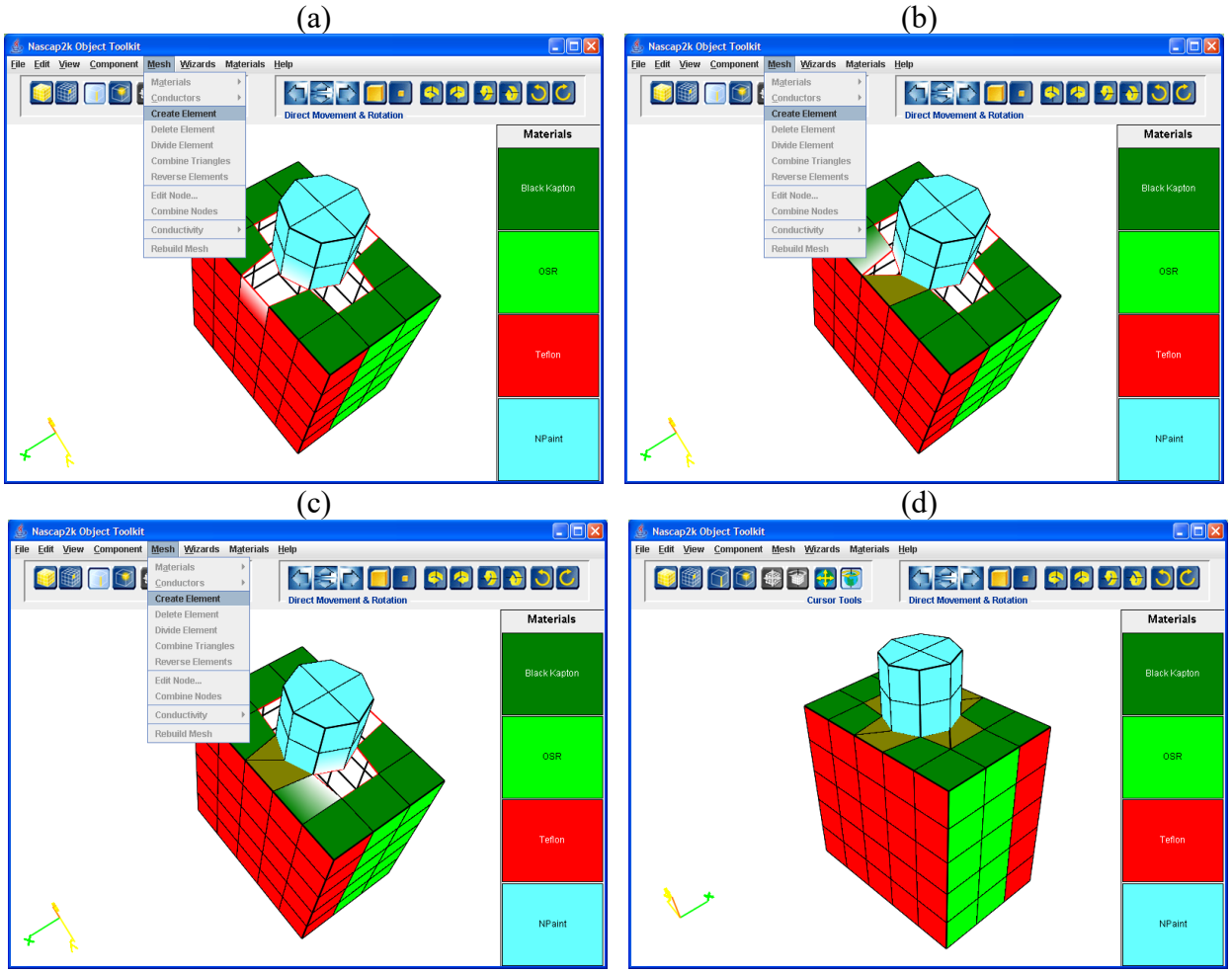


Figure 62. Joining the Omni-antenna and the Body. The Outlines of Elements with Normals Facing into the Screen Are Seen.

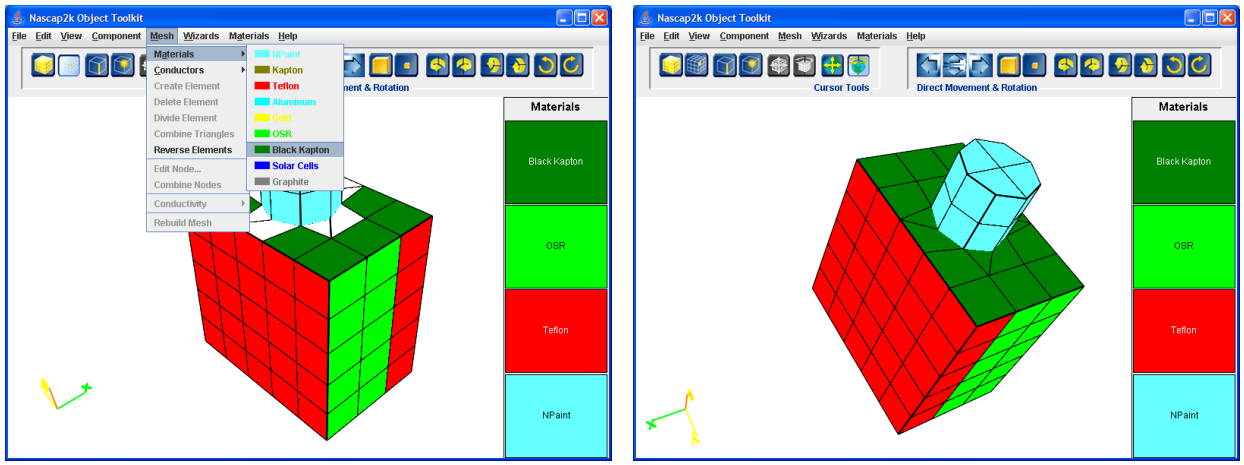


Figure 63. Changing the Materials Assigned to the Top Surface.

Then save the result.

Approved for public release; distribution is unlimited.

18.3 Create and Add the Dish Antenna

To create two Dishes to represent the dish antennas, go to the **Component** Menu, select "Add New," and choose "Dish" from the fly-out menu (Figure 64). Set the diameter of each Dish to be 2 m and the depth 0.1 m. Make each Dish three elements across, and set the surface material to be Graphite.

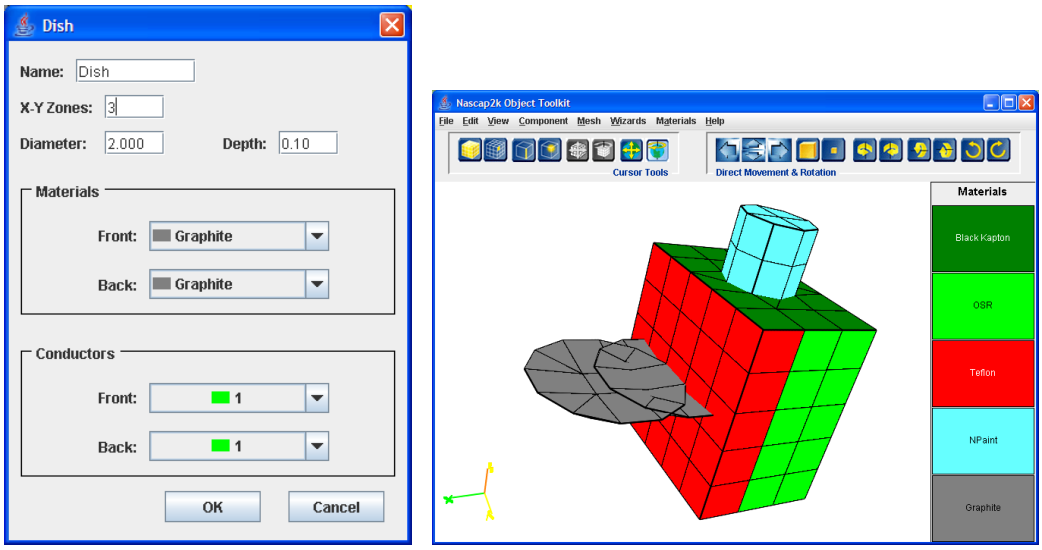


Figure 64. Defining Graphite Dishes to Represent the Dish Antennas. The Antenna Diameter Is 2 m, the Depth Is 0.1 m, and the Antenna Is Three Elements Across.

Place the dish antennas so that the top of the Dishes and the bottom of the body are in the same plane, the Dishes are centered in the Y direction, and the nearest edges of the disks are 0.88 m from the body (Figure 65).

Approved for public release; distribution is unlimited.

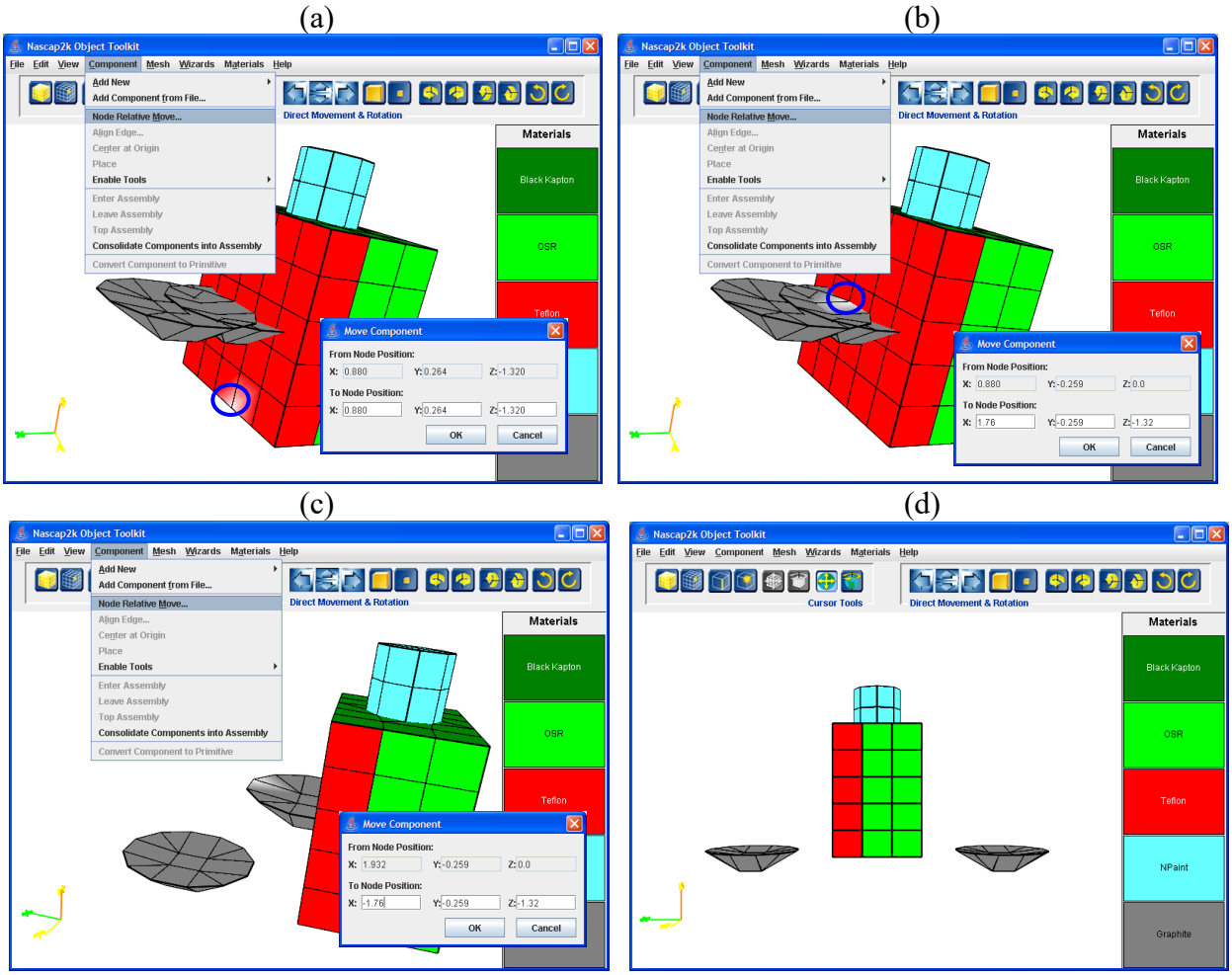


Figure 65. Placing the Dish Antennas So That the Top of the Disks and the Bottom of the Body Are in the Same Plane, the Dishes Are Centered in the Y Direction, and the Nearest Edges of the Disks are 0.88 m from the Body. Selected Nodes Are Moved to (1.76, -0.259, -1.32) and (-1.76, -0.259, -1.32).

Save the resulting model.

18.4 Create and Attach the Solar Arrays

As we may wish to create several models with different solar array orientations, we add the solar arrays last.

To create a Boom to support a solar array; go to the **Component** Menu, select "Add New," and choose "Boom" from the fly-out menu. Make the Boom 2.0 m long with two elements in the long direction. Set the twist to be 33°. Set the material on the Boom to be Kapton, as shown in Figure 66.

Approved for public release; distribution is unlimited.

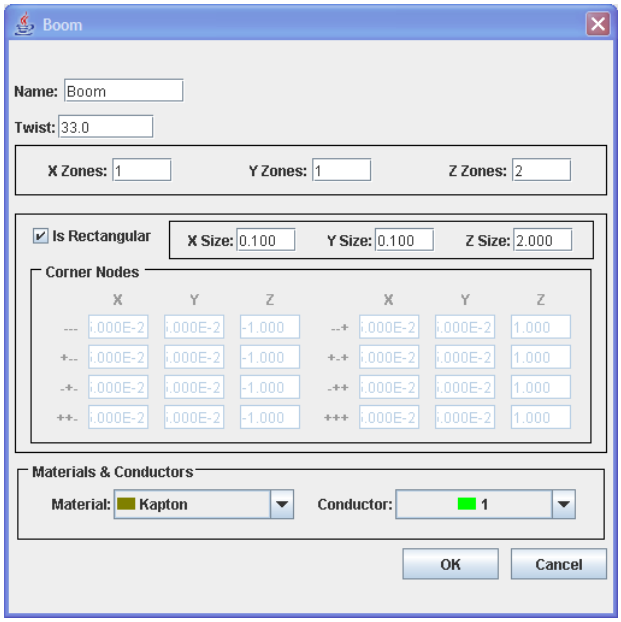


Figure 66. Defining a Boom to Support a Solar Array. The Boom Is 2.0 m Long, Has Two Elements in the Long Direction, and Has a Twist of 33º. The Boom Material Is Kapton.

Next create a Panel to represent a solar array wing; go to the **Component** Menu, select "Add New," and choose "Panel" from the fly-out menu. Make the Panel 4.4 m in width and 7 m in length. Make the Panel seven elements wide and fifteen elements long. Make the front side Solar Cells and the back Black Kapton, as shown in Figure 67.

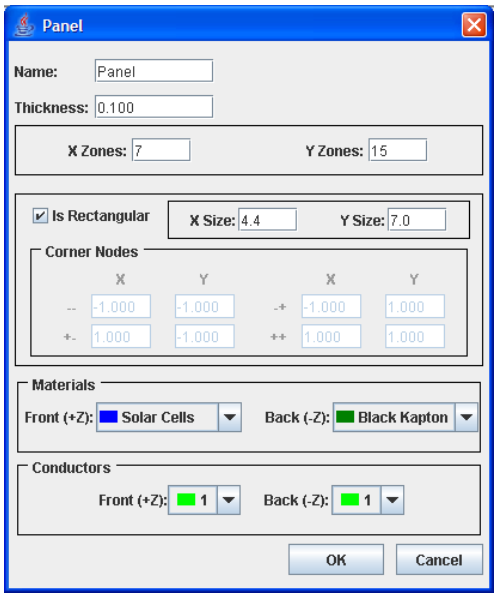


Figure 67. Defining a Panel to Represent a Solar Array Wing. The Panel Is 4.4 m in Width and 7 m in Length. It Is Seven Elements Wide and Fifteen Elements Long. The Front Side Is Material Solar Cells and the Back Is Black Kapton.

Approved for public release; distribution is unlimited.

Reorient the view to be able to select the –Z end of the boom. Attach the solar array to the Boom using the "Edge to Element Wizard," as shown in Figure 68. Select the edge of the solar array as the first component and the –Z end of the Boom as the second component.

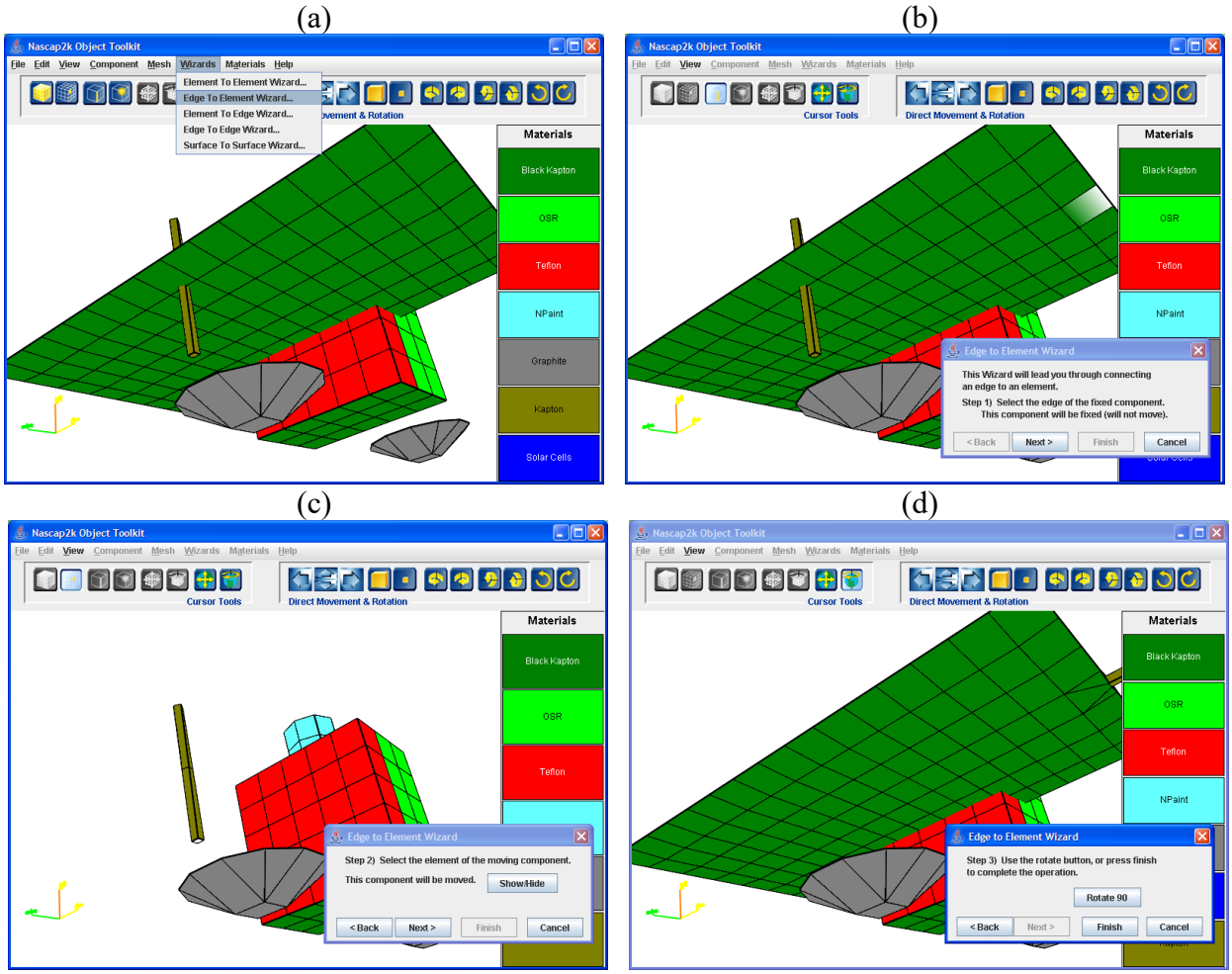


Figure 68. Attaching a Solar Array Panel to a Boom Using the "Edge to Element Wizard."

Figure 69 shows how to make the second solar array and support boom. Select the solar array, go to the **Edit** menu, and select "Copy." Then go to the **Edit** menu and select "Paste." The second solar array is displayed on top of the first.

Approved for public release; distribution is unlimited.

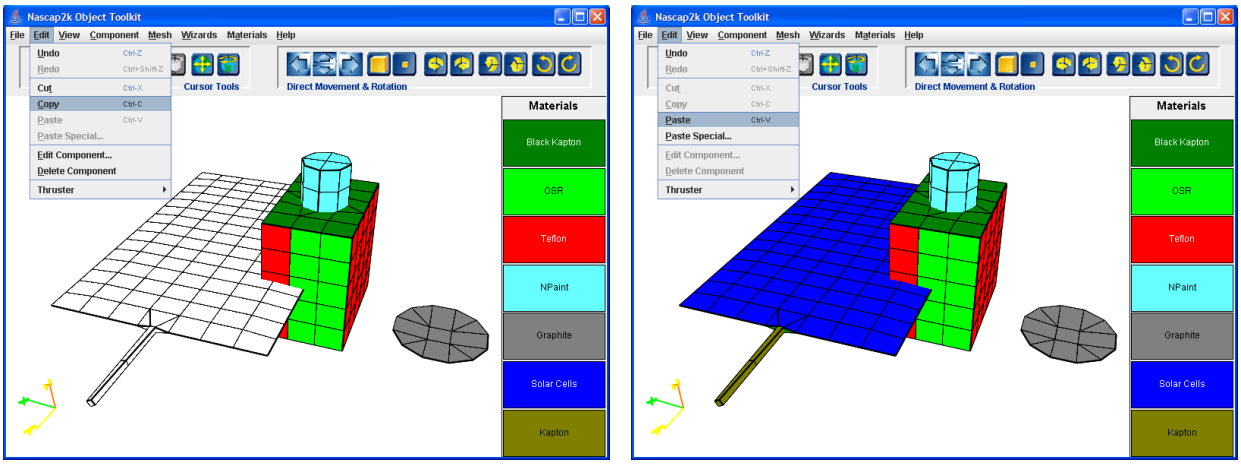


Figure 69. Copying the First Array to Create the Second.

To separate the two arrays, go to the **Component** menu, select "Enable Tools," and choose "Enable" from the fly-out menu to enable the move/rotate component cursor tool buttons. Move one array so that the two are not at the same location. The top two boxes in Figure 70 depict this process. The twist on the Boom for the second array needs to be negative the twist on the first Boom. The bottom four boxes in Figure 70 show how to accomplish this. Select the second solar array, go the **Component** menu, and select "Enter Assembly." Then select the Boom. Go to the **Edit** menu and select "Edit Component." Change the twist value in the dialog box. Finally, go to the **Component** menu and select "Top Assembly." Save the resulting object.

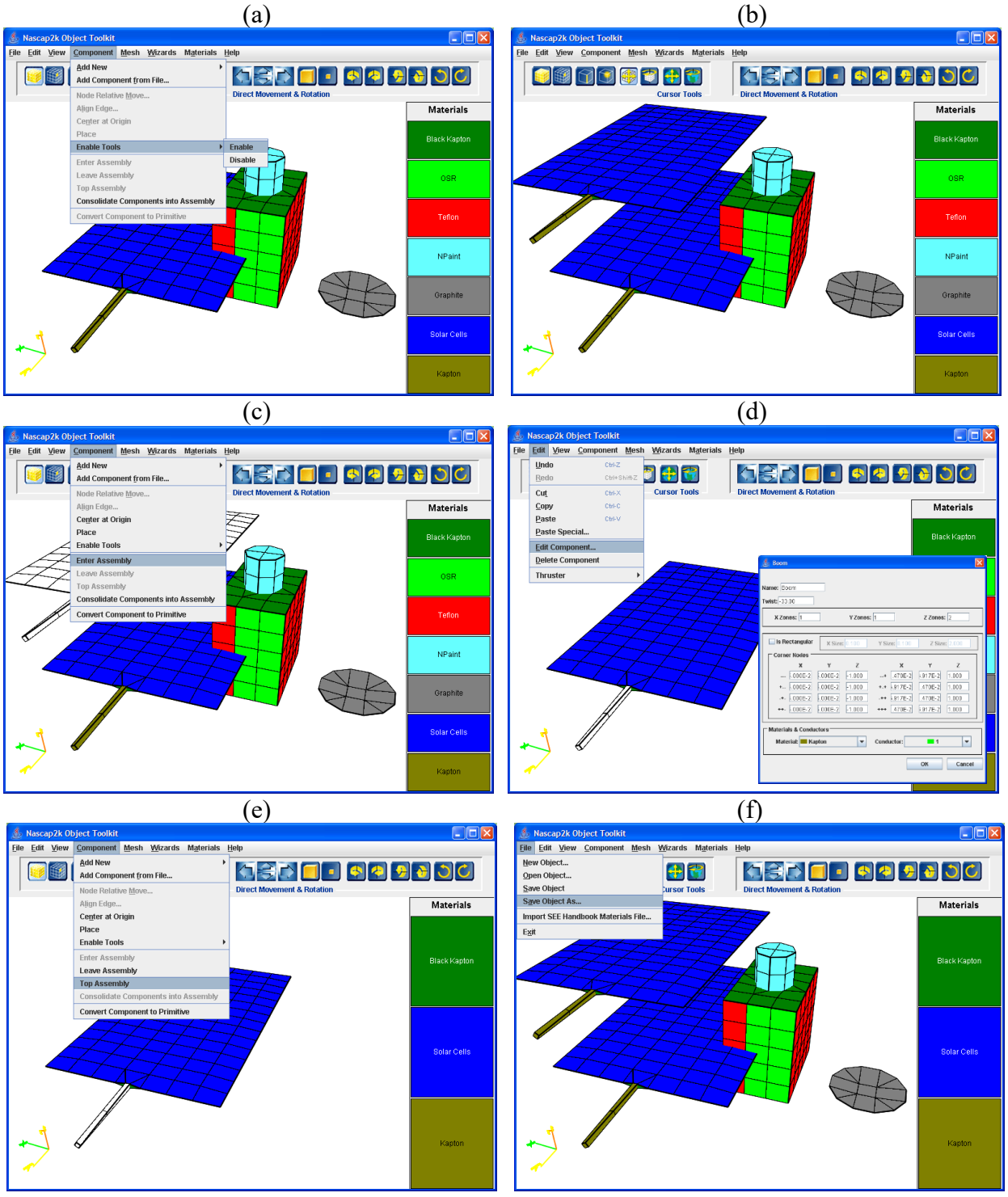


Figure 70. Changing the Twist Angle on the Second Boom and Saving the Resulting Object.

Attach each solar array and Boom to the spacecraft body using the "Element to Element Wizard," making sure that the arrays are oriented as shown in Figure 71 and Figure 72. The Boom with the –33° twist is attached to the –Y face of the body.

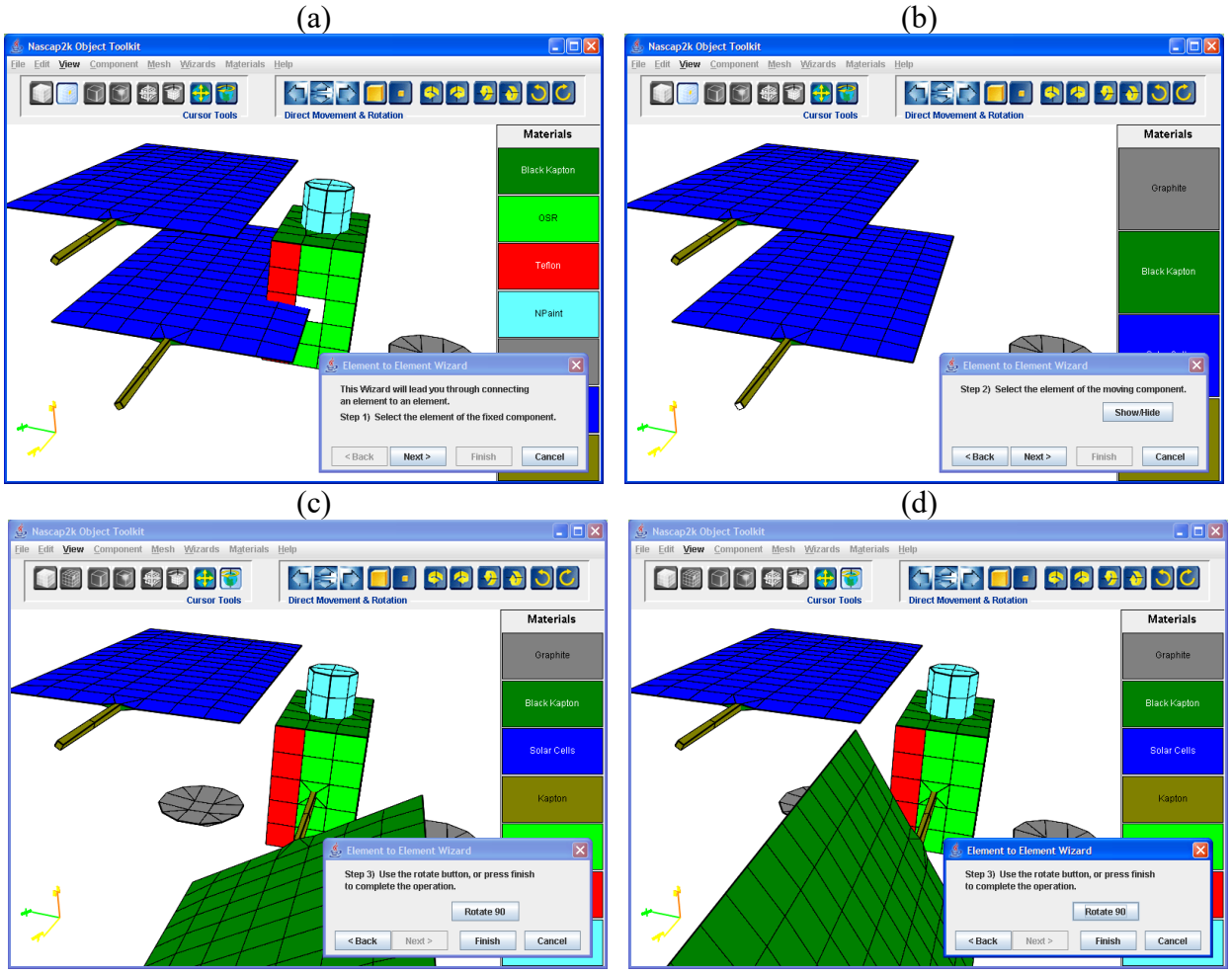


Figure 71. Attaching a Solar Array Panel and Boom to the Spacecraft Body Using the "Element to Element Wizard."

Approved for public release; distribution is unlimited.

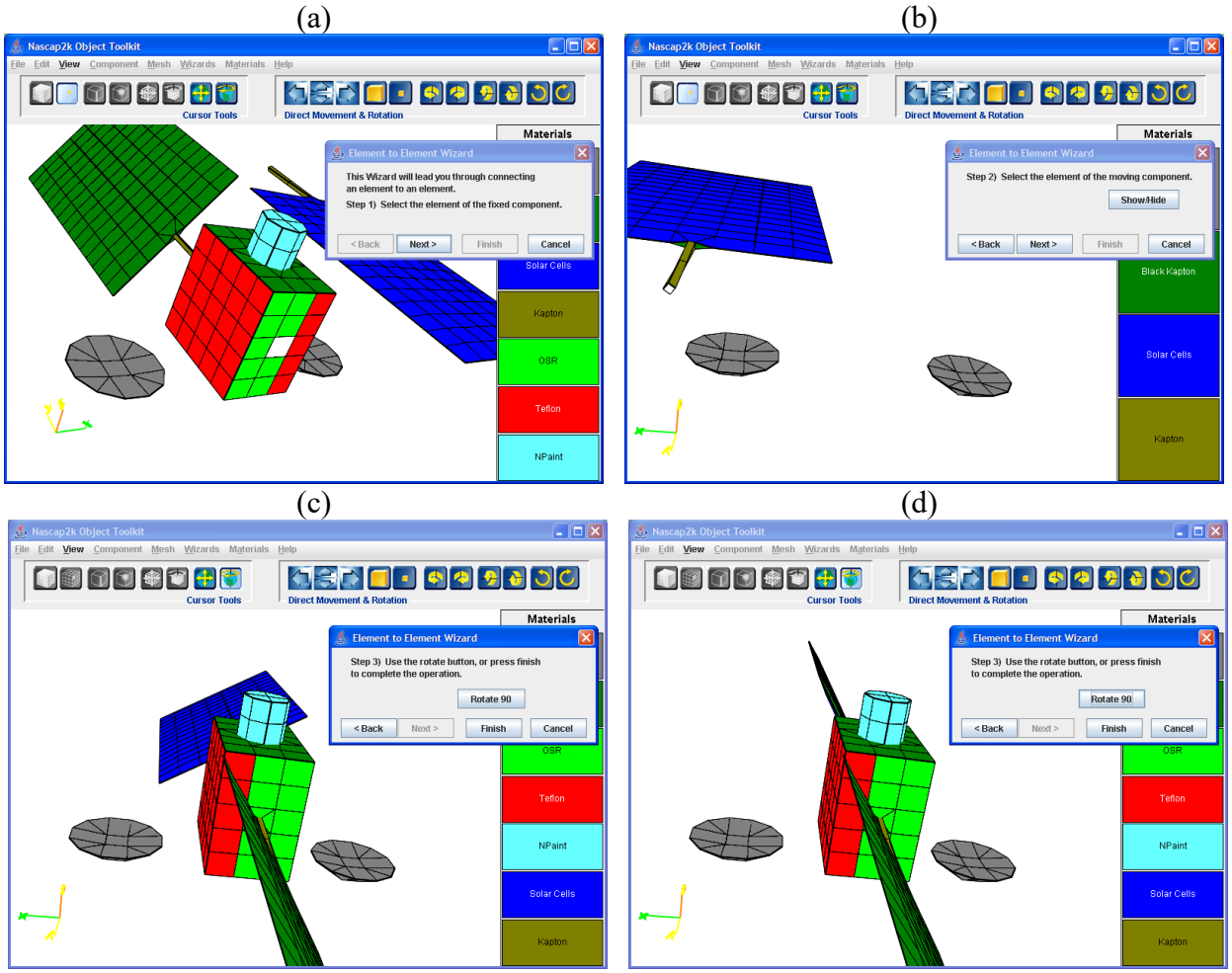


Figure 72. Attaching the Second Solar Array to the Body, and Rotating It So That It Is Parallel to the First Solar Array.

Finally, save the result.

19 Half Cylinder Sunshade

In this example, we build a half Cylinder shape similar to that used to model the MESSENGER sunshade. The MESSENGER model is shown in Figure 73 and the example object is shown in Figure 74. The small resolved surface in the corner represents the Digital Solar Altitude Detector (DSAD). This is an advanced example that uses element editing and Primitives.

Approved for public release; distribution is unlimited.

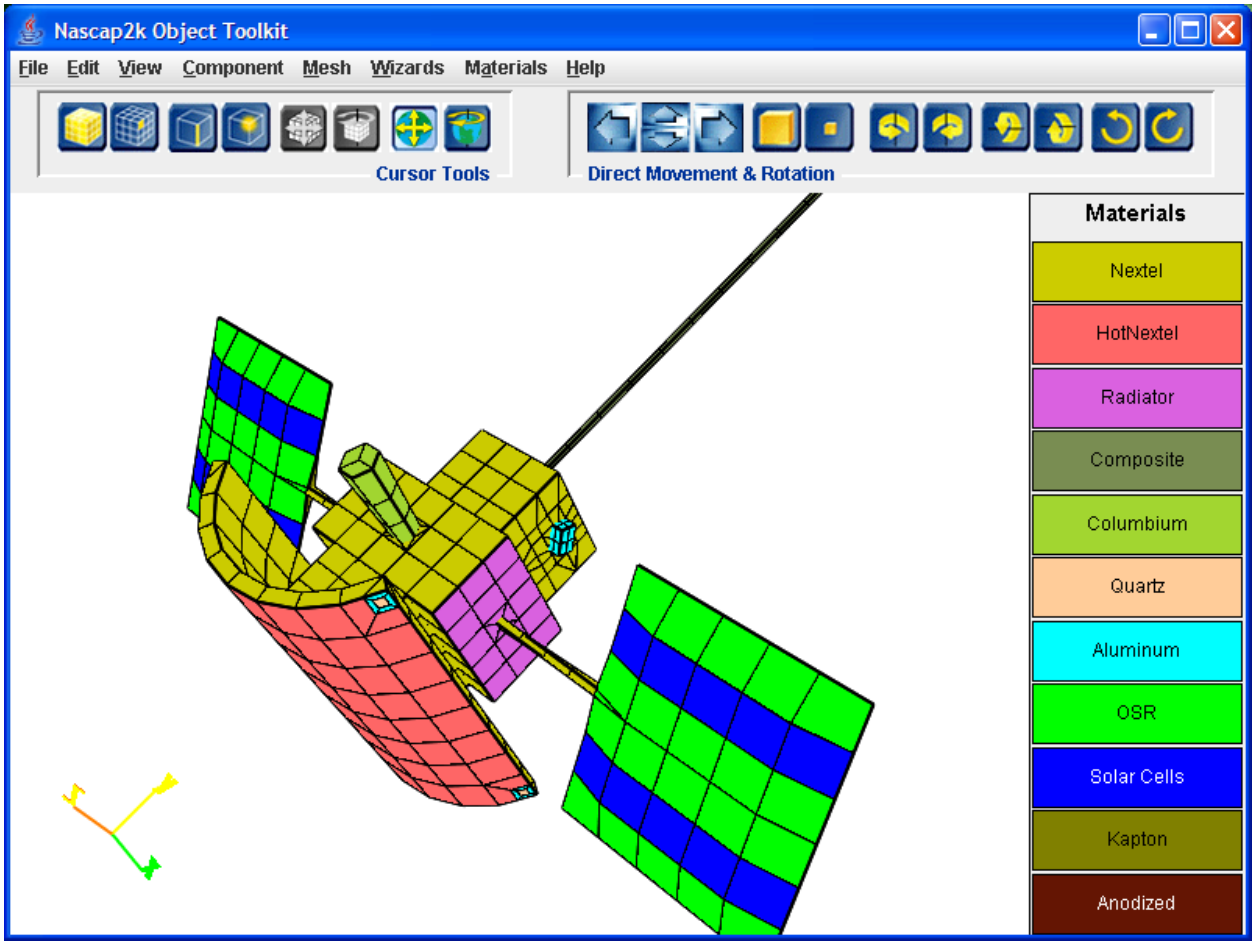


Figure 73. *Object Toolkit* Model of the MESSENGER Spacecraft.

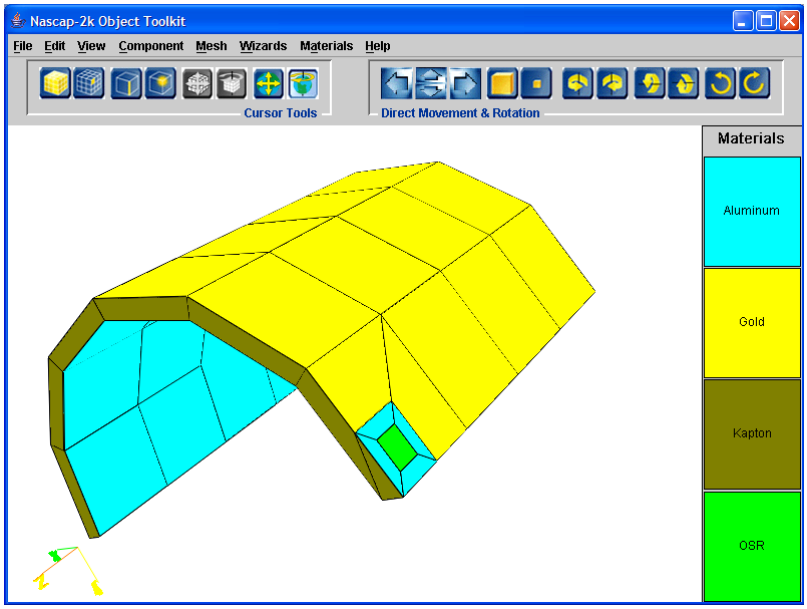


Figure 74. Simplified MESSENGER Sunshade, Including DSAD Instrument, to Be Built in this Example.

Approved for public release; distribution is unlimited.

19.1 Create Basic Component

Begin by creating a Cylinder that is 2 m in diameter and 2.0 m in height with a Gold circumference and Kapton top and bottom, as shown in Figure 75. Set the number of sides to be twelve to make the Cylinder moderately rounded. Set the number of elements in the vertical direction to four, so that the elements are approximately square.

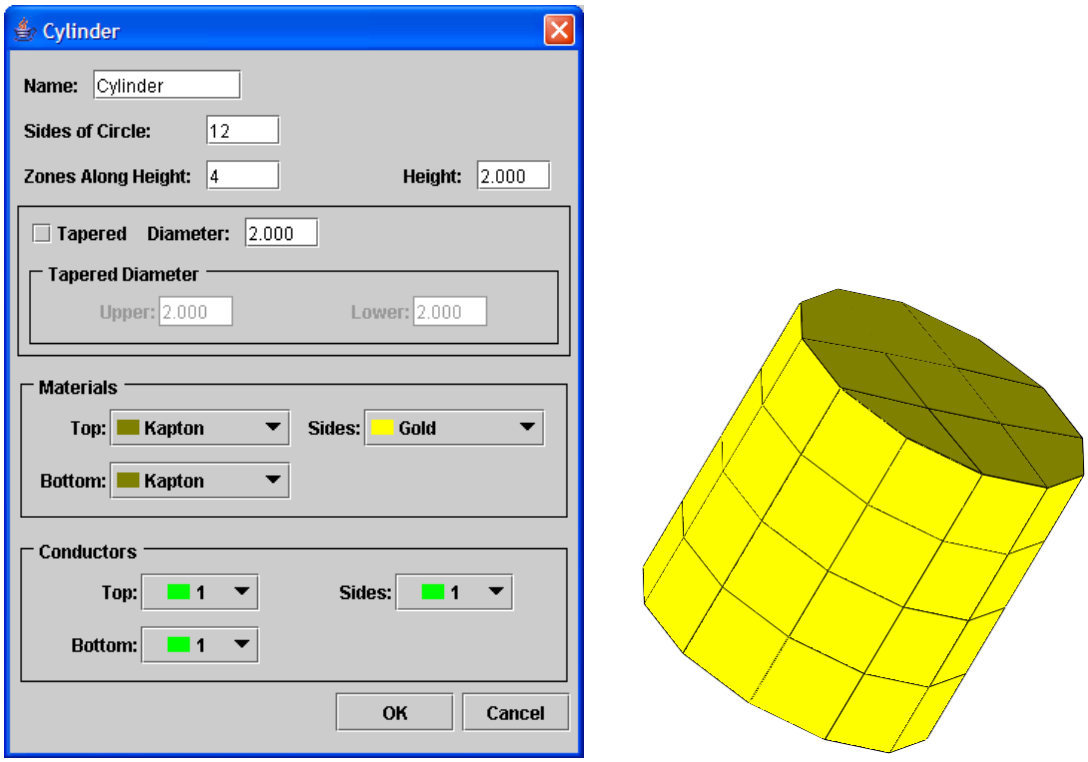


Figure 75. Creating the Cylinder.

19.2 Remove Extra Surfaces

The sunshade is made from one-half of the side of the Cylinder. Delete all the elements on the top and bottom and on half of the side surface. To delete an element, make sure the "Element Cursor Tool" button is selected, click the element to be deleted, go to the **Mesh** menu, and select "Delete Element" (Figure 76). You end up with a semicircle the height of the Cylinder (Figure 77). Save the result—it is good practice to save often—and you will import it later.

Approved for public release; distribution is unlimited.

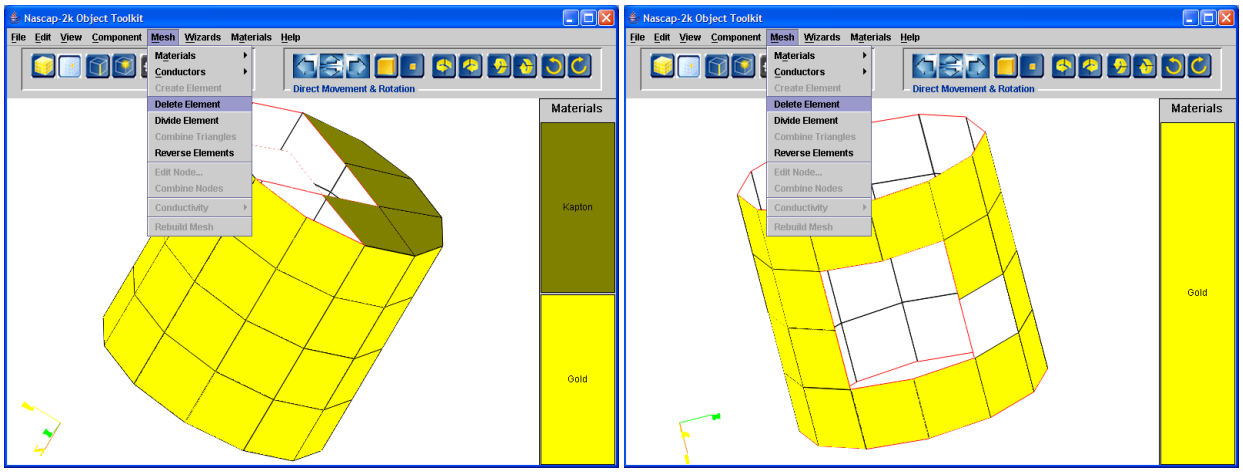


Figure 76. Deleting the Extraneous Top, Bottom, and Side Elements. The Outlines of Elements with Normals Facing into the Screen Are Seen.

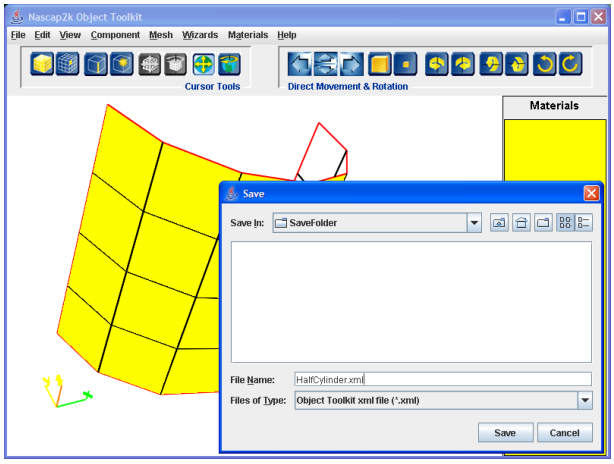


Figure 77. Saving the Half-Cylinder Shell. The Outlines of Elements with Normals Facing into the Screen Are Seen.

19.3 Create Inside Surface

The inside surface of the sunshade is created by changing the material and reversing all the elements. First, change the material on all the elements to Aluminum. Multiple elements are selected using the CTRL key. Once elements are selected, the desired material can be chosen by going to the **Mesh** menu and selecting "Materials" (Figure 78).

The next step is to reverse the normals of all the elements. Again, multiple elements can be selected to reverse all at once. In Figure 79, some of the distant elements have already been reversed and some of the near ones have been selected and are about to be reversed.

Approved for public release; distribution is unlimited.

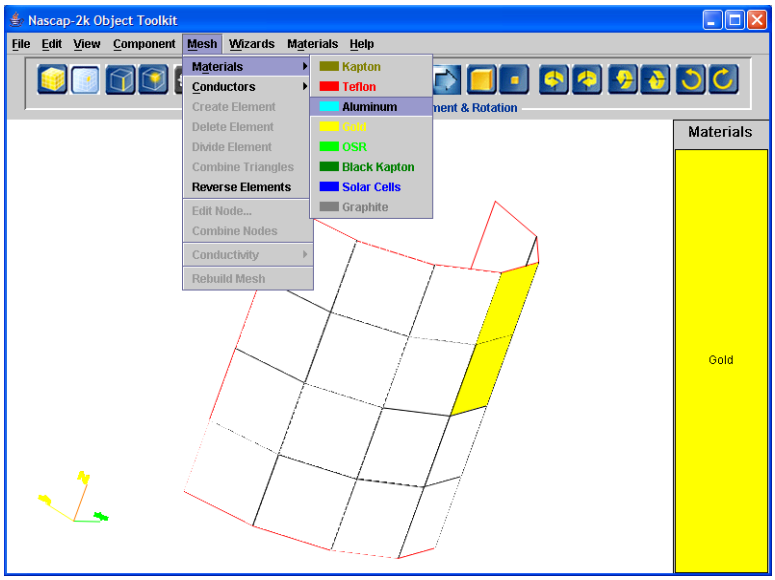


Figure 78. Changing the Surface Material from Gold to Aluminum. The Selected Elements Are Displayed in White. The Other Elements Have Not Been Changed Yet. The Outlines of Elements with Normals Facing into the Screen Are Seen.

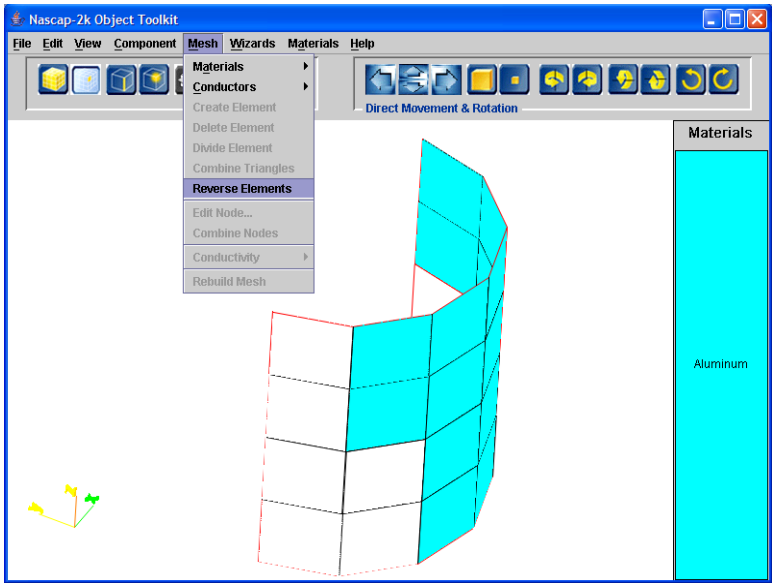


Figure 79. Reversing the Normal of the Elements. In the Figure, Some of the Distant Elements Have Already Been Reversed and Some of the Near Ones Have Been Selected and Are About to Be Reversed. The Outlines of Elements with Normals Facing into the Screen Are Seen.

19.4 Create Object With Inner and Outer Surfaces

The next step is to read in the component saved earlier to serve as the outer surface with Gold, non-reversed elements (see Figure 80), and to enlarge it so that we have a smaller inner and larger outer surface. Read in the file saved earlier as a new component by going to the **Component** menu and selecting "Add Component from File…."

Approved for public release; distribution is unlimited.

Because the Gold outer surface is a Cylinder and only Primitives can be scaled, the Cylinder first needs to be converted to a Primitive. A Primitive is a component that is described solely by its mesh rather than by a standard geometry. Select the Gold Cylinder, go to the **Component** menu, and choose "Convert Component to Primitive" (Figure 81). Now that the Gold component is a Primitive, enlarge it in the X and Y directions by 10 percent as shown in Figure 82. Select the gold component and choose "Edit Component" from the Edit menu. Z is the height (parallel to the Cylinder axis), so it can be left alone.

Save the object, shown in Figure 83, to the file DisjointCylinders.xml.

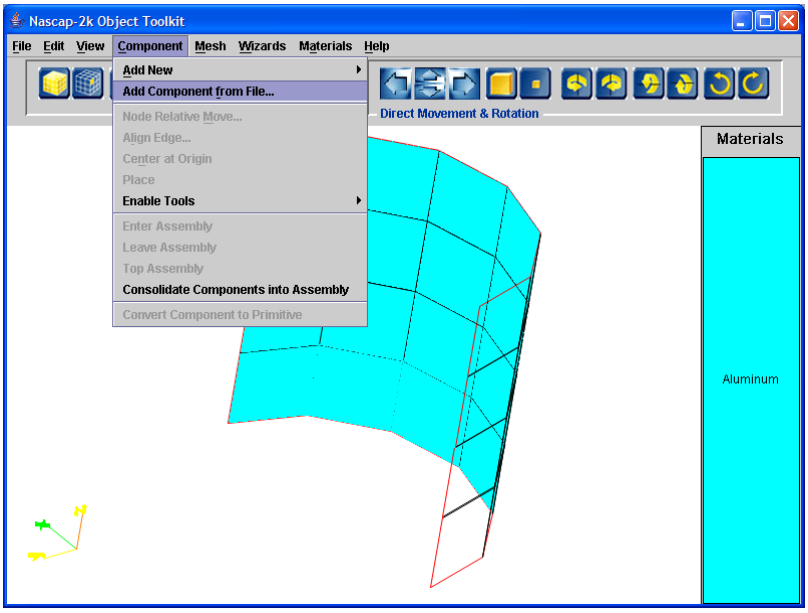


Figure 80. Adding the Outer Surface of the Sunshade from the Previously Saved File. The Outlines of Elements with Normals Facing into the Screen Are Seen.

Approved for public release; distribution is unlimited.

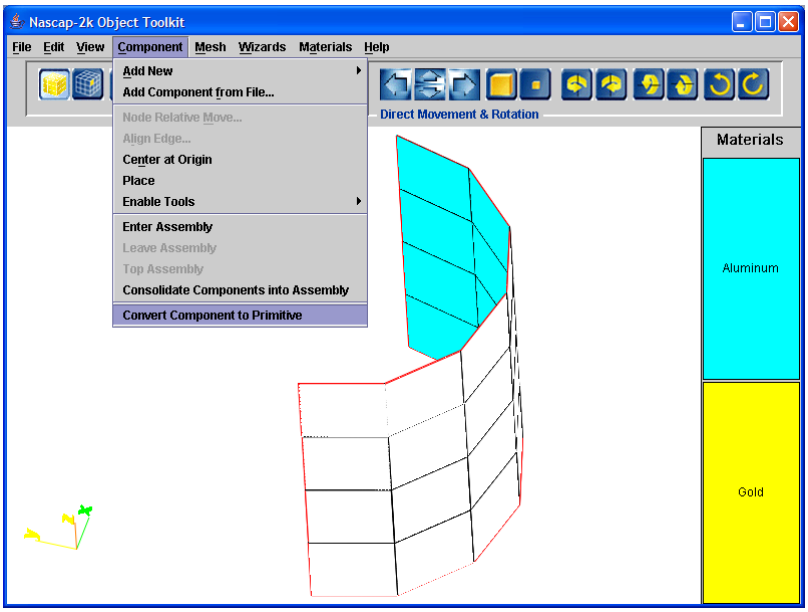


Figure 81. Converting the Outer Surface to a Primitive. The Outlines of Elements with Normals Facing into the Screen Are Seen.

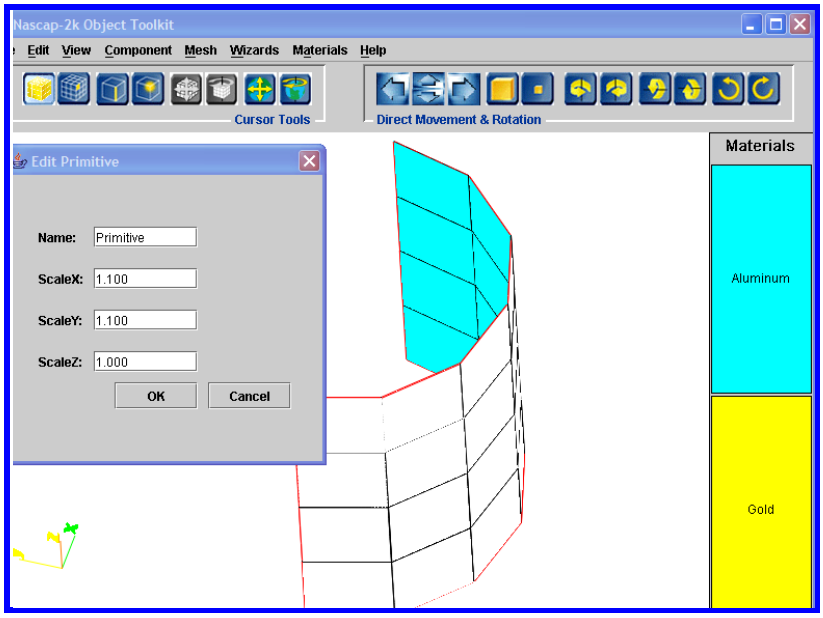


Figure 82. Scaling the Outer Edge by Expanding It in the X and Y Directions by 10%. The Outlines of Elements with Normals Facing into the Screen Are Seen.

Approved for public release; distribution is unlimited.

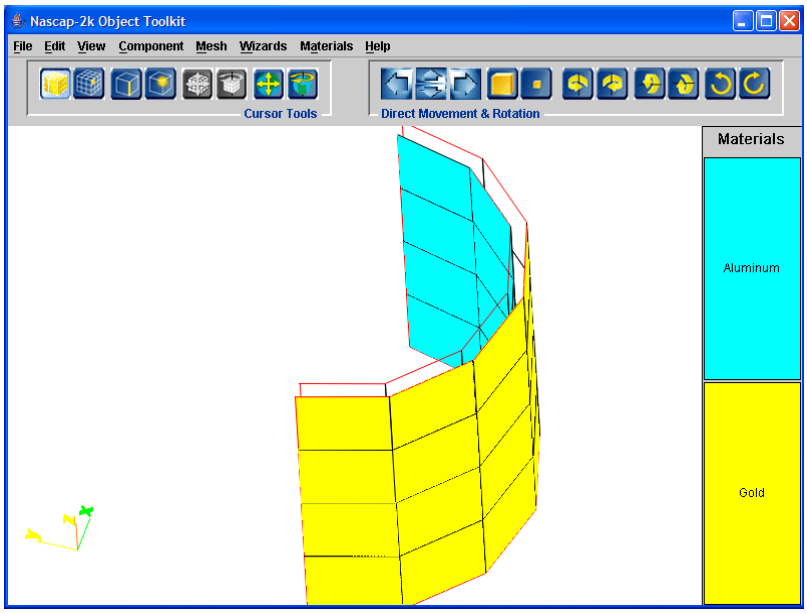


Figure 83. Resulting Inner and Outer Surfaces of the Sunshade. The Outlines of Elements with Normals Facing into the Screen Are Seen.

19.5 Join Inner and Outer Surfaces

At this point the sunshade has disjoint inner and outer surfaces. The next task is to join the surfaces by adding elements along the sides. The first step is to consolidate the components into a single assembly. This is done by choosing "Consolidate Components into Assembly" on the **Component** menu.

Now add elements to connect the inner and outer surfaces. Once done, this has the effect of getting rid of all the red lines, which are indicators of free edges. To do this, select an open edge on the Gold side (using the Select-Edge Cursor Tool), rotate the object view so the Aluminum side is visible and select the corresponding open edge. Go to the **Mesh** menu and select "Create Element" which is now enabled, as shown in Figure 84. Newly created elements have material Kapton, and the sides of the sunshade can be left as Kapton.

After all twenty joining elements have been created and you have a half-cylindrical sunshade, shown in Figure 85, save the object to the file JointCylinders.xml.

Approved for public release; distribution is unlimited.

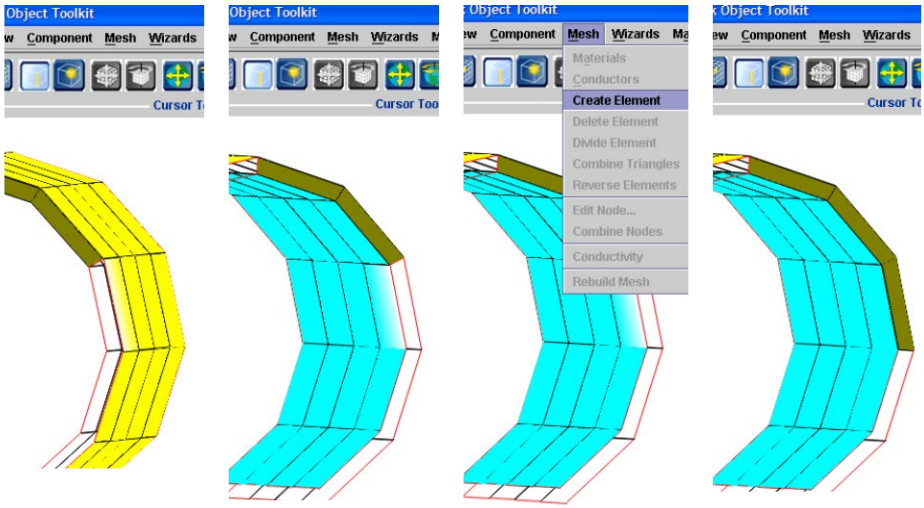


Figure 84. Creating an Element That Joins the Two Sides of the Sunshade. The Outlines of Elements with Normals Facing into the Screen Are Seen.

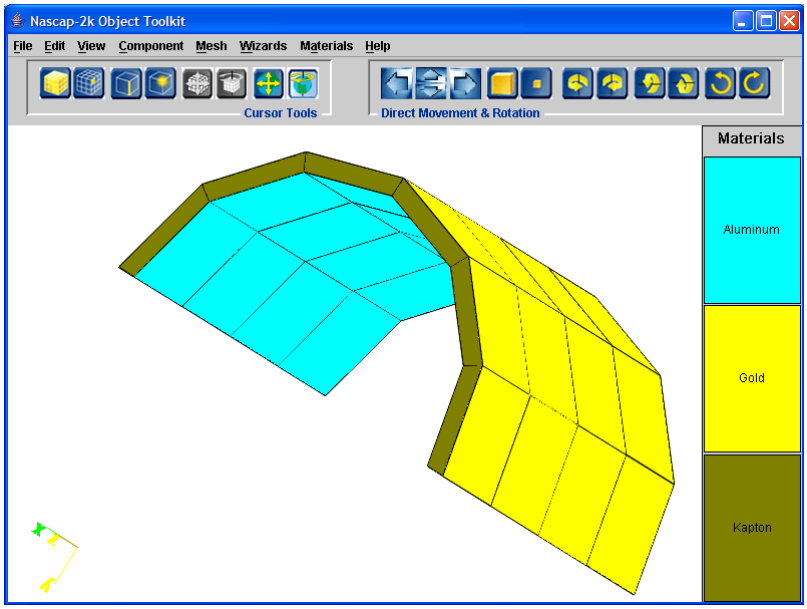


Figure 85. Hemispherical Sunshade.

19.6 Add Instrument

The last step is to create one of the DSADs that appear in each corner of the MESSENGER sunshade. Start by dividing one of the corner elements. Select the element to be divided, go to the **Mesh** menu, and select “Divide Element” (Figure 86).

Keeping the corner element of the four new ones, delete the other three. Fill in the empty region with two new elements to eliminate element incompatibility indicated by red lines. Create the new elements from the free edges as shown in Figure 87. The other two edges of the original element remain red and are dealt with later.

Approved for public release; distribution is unlimited.

Divide the small DSAD element just created, and divide each of the resulting smaller elements, to end up with sixteen small elements as shown in Figure 88. The object of this is to have nodes defining the inner quarter of the DSAD. Now delete the inner elements and add a single element where the four have been deleted by selecting four nodes counterclockwise, as shown in Figure 89. This element represents the active area of the DSAD.

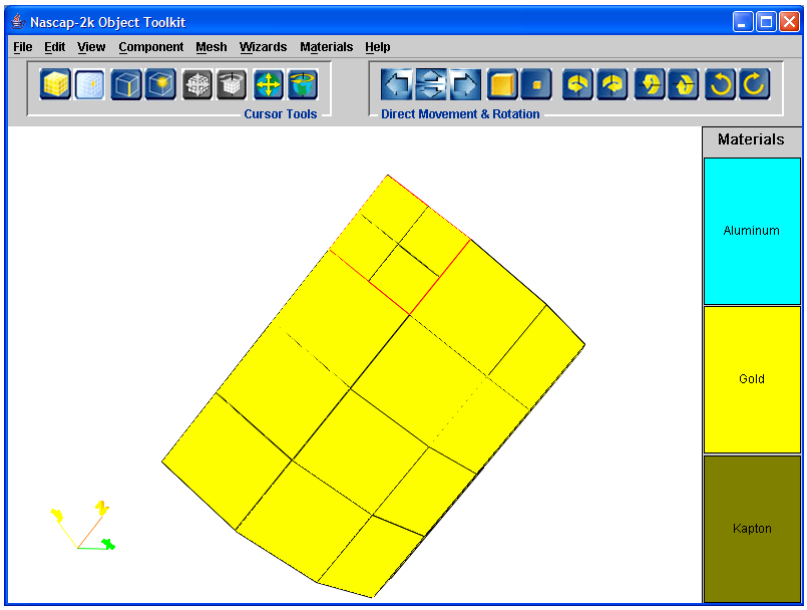


Figure 86. Dividing a Corner Element.

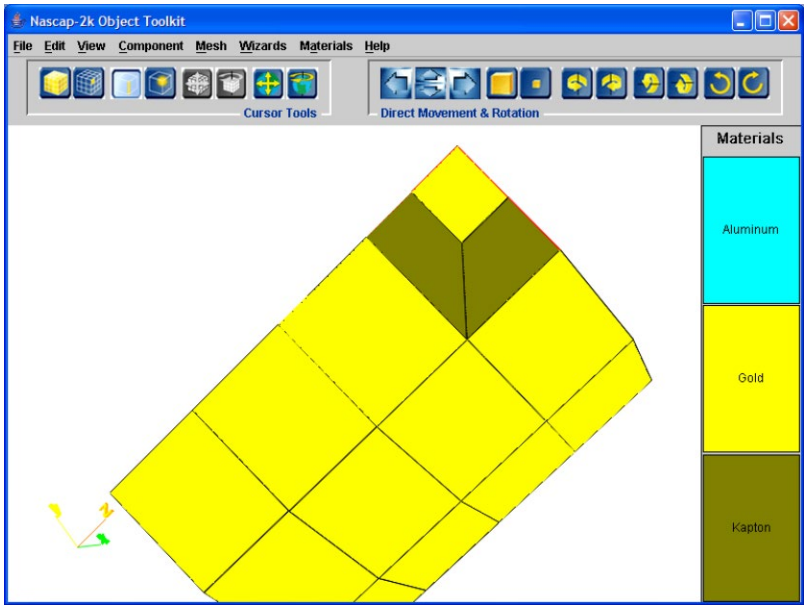


Figure 87. Result after Deleting Three Elements and Replacing Them with Two to Eliminate the Incompatibility along the Face of the Sunshade. Incompatible Elements Remain along the Edge.

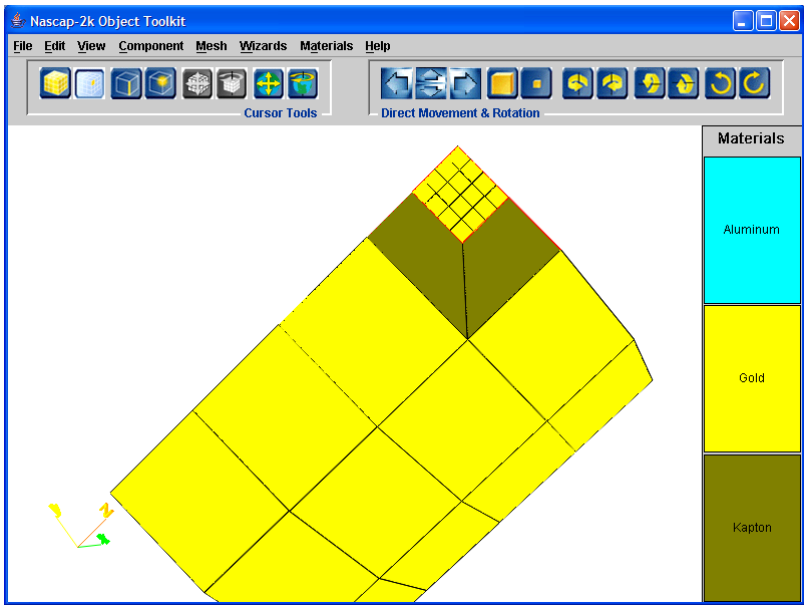


Figure 88. Result after Dividing the Corner Element and Then Dividing Each of the Newly Formed Elements.

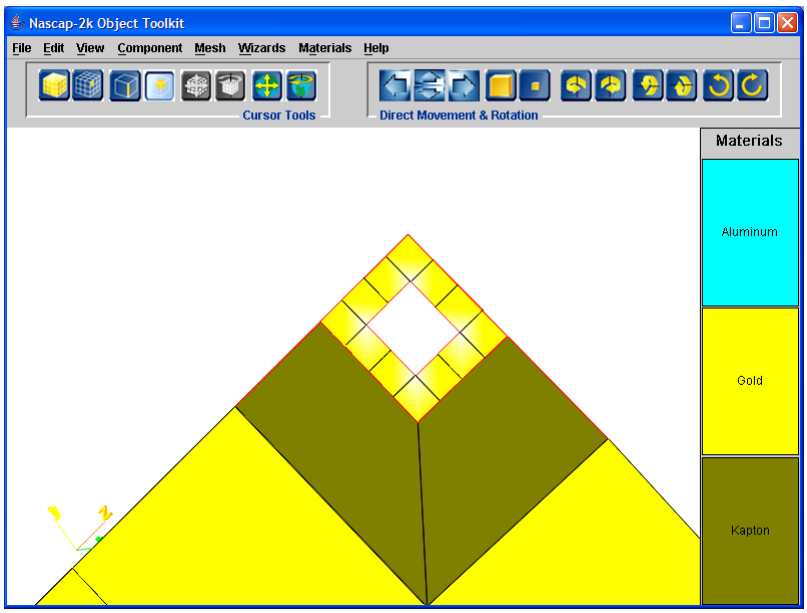


Figure 89. Result after Deleting the Center Four Elements of the Finely Divided Region and Then Selecting the Four Corner Nodes.

19.7 Consolidate to Simplify Editing

While building complex models such as this one, *Object Toolkit* can become confused and give strange results. An example occurred while developing this example. Figure 90 shows the result obtained after creating a new element from the four nodes selected in Figure 89. The diagonal red line and the appearance of a bit of turquoise indicate that something has gone wrong. If you "undo" the last step and repeat and the results still aren't correct, it is probably because too much editing has been done at this assembly/component level. To proceed, it is necessary to move all

the editing down one level by going to the **Component** menu and selecting "Consolidate Components into Assembly." After doing this, the addition of the new element can proceed, as shown in Figure 91.

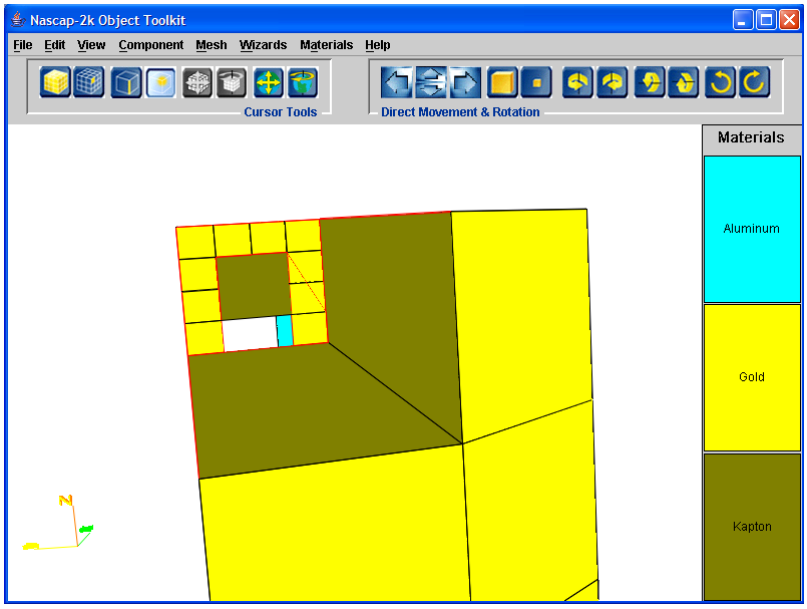


Figure 90. Here, *Object Toolkit* Misinterpreted the Attempt to Create a New Element Using the Four Corner Nodes.

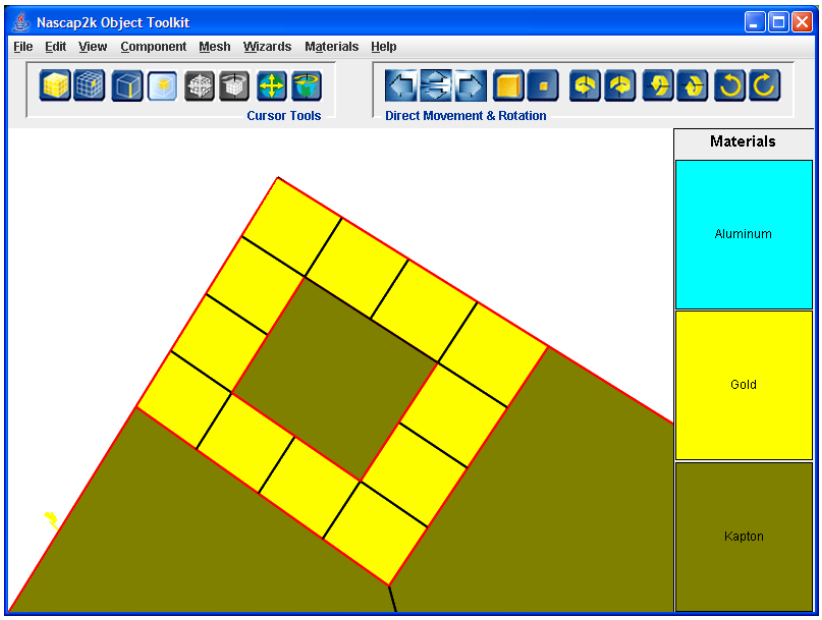


Figure 91. New Element Created after Consolidation and Then Selecting Four Nodes and Requesting "Create Element."

19.8 Complete Instrument

Complete the model of the DSAD by creating the frame. Begin by deleting the smaller elements on the inner edge and replacing them with newly created larger ones, as shown in Figure 92.

Approved for public release; distribution is unlimited.

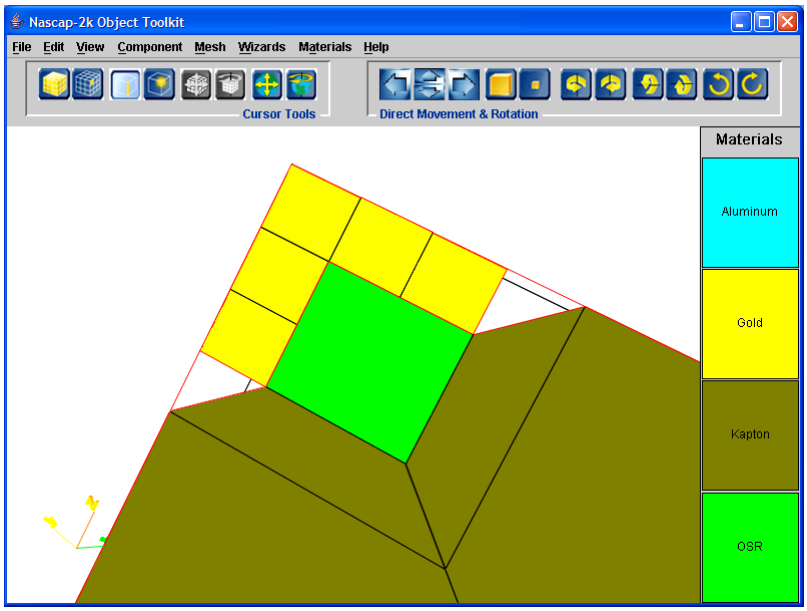


Figure 92. Result after Deleting Seven Small Elements and Replacing Them with Two Compatible, Larger Ones. The Outlines of Elements with Normals Facing into the Screen Are Seen.

Before constructing the outboard part of the frame, it is necessary to fix up the side elements. Delete the existing side element shown in Figure 93a, and connect the free edge on the front with the back. This leaves the unfilled triangle shown in Figure 93b. Then create a triangular element that fills in the rest of the side by selecting the two adjacent free edges, as shown in Figure 93c, to get the result shown in Figure 93d.

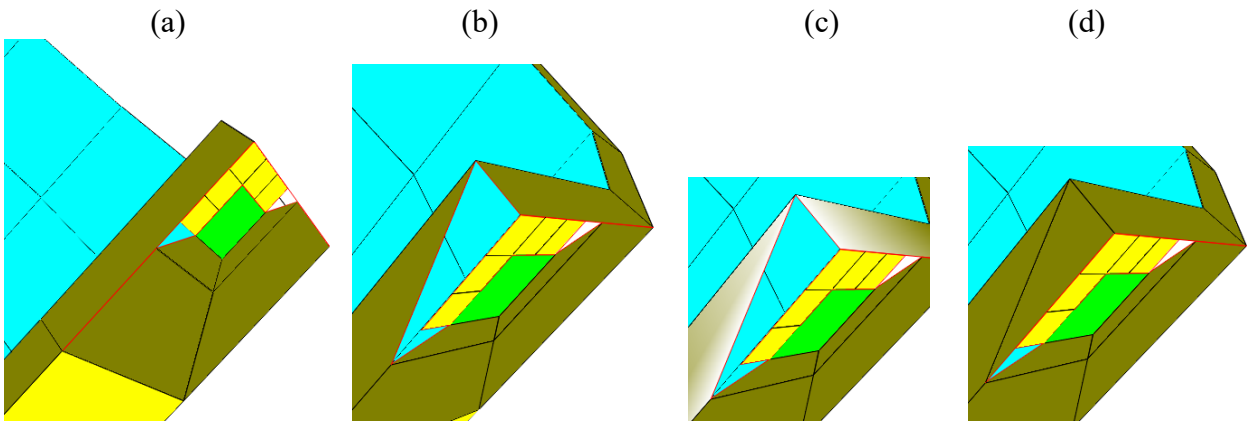


Figure 93. Replacing the Existing Side Element with Two New Elements, a Quadrilateral and a Triangle. The Outlines of Elements with Normals Facing into the Screen Are Seen.

Delete the remaining small elements. Create an element between the left side of the OSR and the free edge on the triangle. Then, as shown in Figure 94, select the two slanted free edges (upper left) to create the final element of the frame.

Approved for public release; distribution is unlimited.

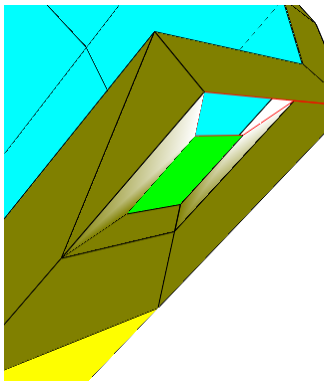


Figure 94. Replacing the Small Elements with Larger Compatible Ones to Form the DSAD Frame. The Outlines of Elements with Normals Facing into the Screen Are Seen.

Delete the remaining side element and remesh it as a quadrilateral and triangle to get the result shown in Figure 95

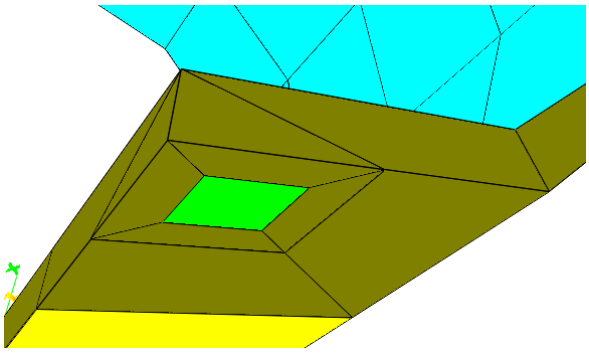


Figure 95. Remeshed Side Element.

Finally, correct the materials so that the sunshade is Gold and the DSAD has an Aluminum frame and an active OSR center.

Save the final result shown in Figure 96.

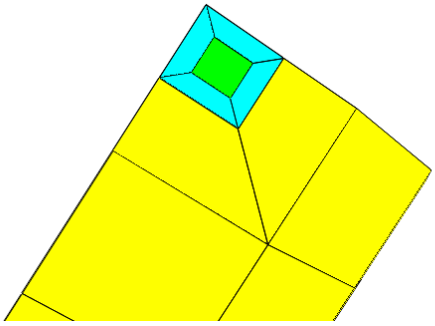


Figure 96. Completed Model of the DSAD Instrument.

Approved for public release; distribution is unlimited.

20 Triangular Solar Sail

To illustrate how to create a large triangular solar sail with a curved edge, we create the four-by-four example shown in Figure 97. The sail is constructed to be as thin as possible but thick enough to be used in a *Nascap-2k* calculation.

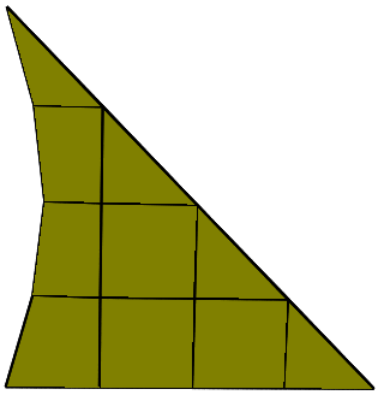


Figure 97. Top View of Sample Solar Sail Created in This Example.

Begin by creating a Panel as shown in Figure 98 (one of the standard components selected on the "Add New" fly-out menu.) The solar sail is constructed using half the Panel by deleting half of the elements and leaving the corners, as shown in Figure 99. The next step is to add triangular elements to fill in along the diagonal, as shown in Figure 100. This is done by selecting two edges (remember to hold down the CTRL key to select the second edge), going to the **Mesh** menu, and selecting "Create Element." Repeat this process for the other side of the panel.

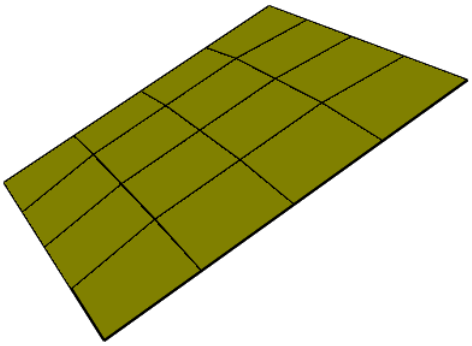


Figure 98. A Four-by-four Panel is the Starting Point for a Small Solar Sail.

Approved for public release; distribution is unlimited.

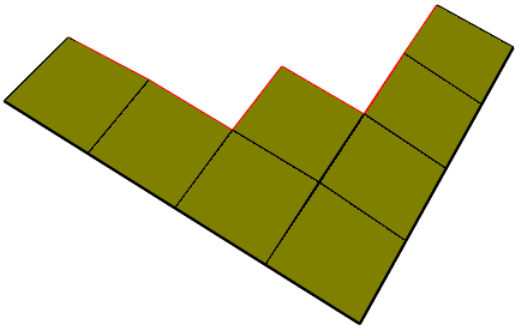


Figure 99. Deleting Half of the Panel Leaves Open Edges.

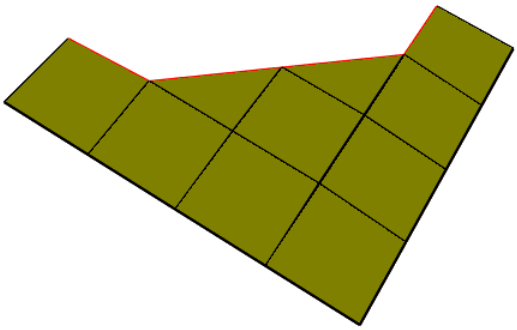


Figure 100. Adding Triangular Elements by Going to the Mesh Menu and Choosing “Create Element.”

Now that the basic shape has been formed, it is necessary to join the front and back in such a way that no two triangles share more than two nodes in common. (The existence of two elements with the same nodes, but facing opposite directions is not allowed in any code that uses *Object Toolkit* for object definition.) First, to avoid extra elements along the side, merge the front and back points along the diagonal as shown in Figure 101. In this case, only the center points of the two sides can be merged together. To merge the nodes together, select the two nodes, go to the **Mesh** menu, and select “Combine Nodes.” Next, create new triangular elements that fill in along the diagonal, as shown in Figure 102.

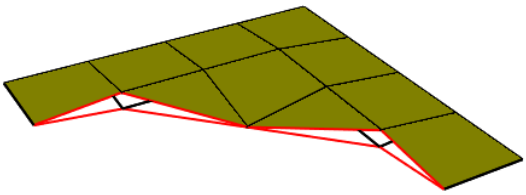


Figure 101. Closing the Open Edge in the Center By Going to the Mesh Menu and Choosing “Combine Nodes.” The Outlines of Elements with Normals Facing into the Screen Are Seen.

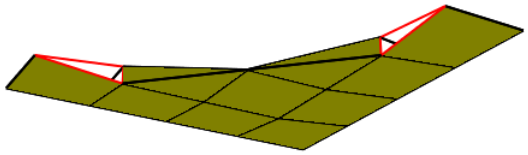


Figure 102. Filling in the Edge with Additional Triangular Elements, While Making Sure That the Two Sides Remain Separate. The Outlines of Elements with Normals Facing into the Screen Are Seen.

Next, replace the corner quadrilateral elements with triangles by deleting the quadrilateral and creating a new triangle, as shown in Figure 103. The front and back elements need to be replaced one at a time, so that two sides of the triangle exist before it is created. And finally, additional triangular elements need to be added to fill in on the sides, as shown in Figure 104.

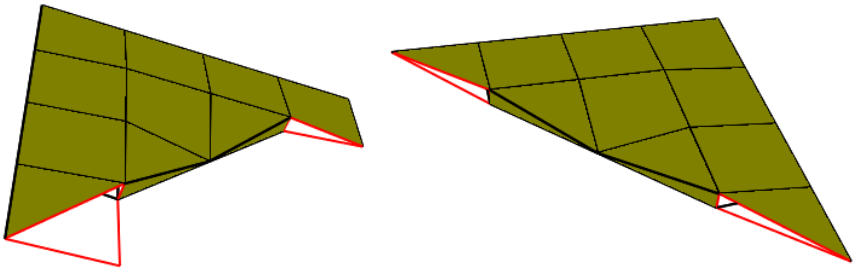


Figure 103. Replacing Each of the Square Elements in the Corners with a Triangular One, One at a Time. The Outlines of Elements with Normals Facing into the Screen Are Seen.

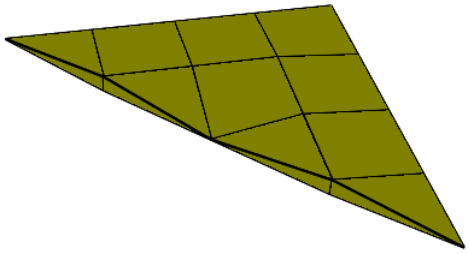


Figure 104. Adding Additional Triangular Elements to Fill in the Final Holes.

The final object has a sine wave curve to one of edges. This is done by moving each node to the desired position using “Edit Node” on the **Mesh** menu, as shown in Figure 105.

Approved for public release; distribution is unlimited.

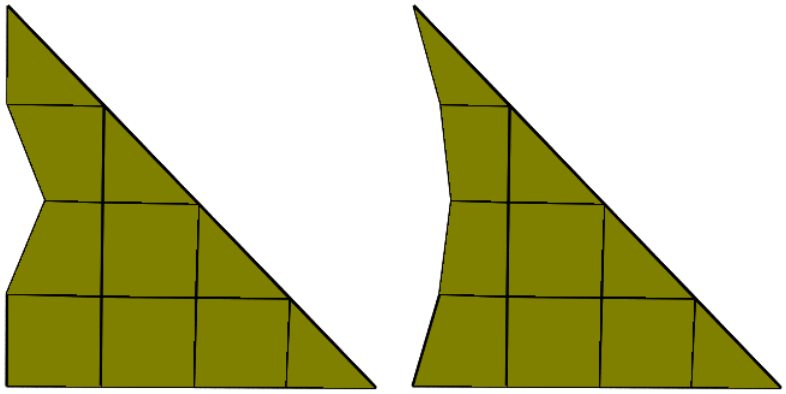


Figure 105. Moving Each Node to Form the Curved Edge by Going to the Mesh Menu and Choosing “Move Node.”

21 Attaching a Boom to a Boom

21.1 Make a T Shape

To attach one Boom to the side of another Boom, make the second Boom slightly larger in cross section than the first. For example, if the first Boom is 0.1 m by 0.1 m, the second could be 0.12 m by 0.12 m.

First, create two Booms, one 0.1 m by 0.1 m and the other 0.12 m by 0.12 m. Make the larger Boom three elements in length so that the smaller boom can be attached to its middle, as shown in Figure 106. Use the “Element to Element Wizard” to connect the two as shown in Figure 107.

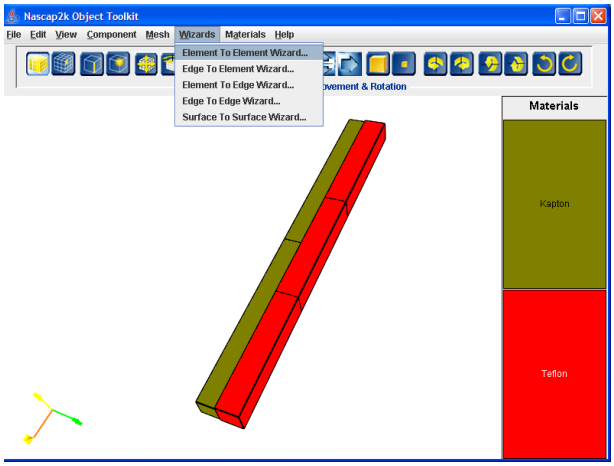


Figure 106. Booms to Be Connected into a T Shape.

Approved for public release; distribution is unlimited.

Figure 107. Joining Two Booms into a T Shape Using the “Element to Element Wizard.”

21.2 Make a Y Shape

The mesh editing capabilities can be used to join Booms into more complex shapes. To form a Y shape, start with a 2 m by 0.1 m by 0.1 m Boom and a 1 m by 0.1 m by 0.1 m Boom. Make the longer Boom three elements long (Figure 108).

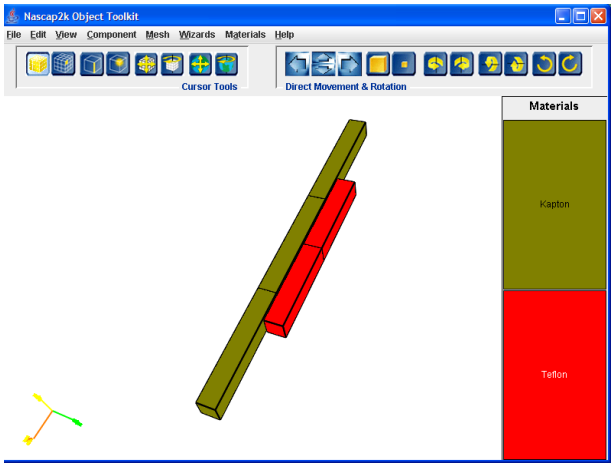


Figure 108. Booms to Be Connected into a Y Shape.

Approved for public release; distribution is unlimited.

Booms are created with the long axis along the Z axis of the object. Leave the long Boom oriented this way. Select the long edge of the short boom. The sign of the vector in the **Align Component Edge** dialog box depends on which side of the edge is selected. To simplify following these instructions, select the edge by clicking slightly to the right of the edge. Then go to the **Component** menu and select "Align Component" to reorient the short Boom so that it points 45° off the Z axis in the Y-Z plane, as shown in Figure 109. Then go to the **Component** menu and use "Node Relative Move" to move the short Boom so that its nodes correspond to the nodes of the long Boom, as shown in Figure 110.

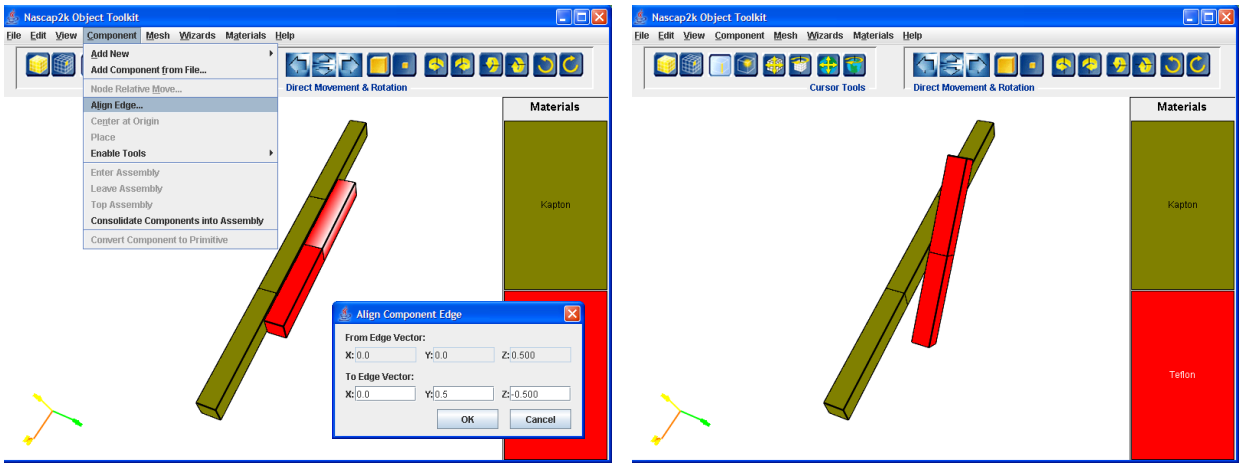


Figure 109. Rotation of Short Boom to Be 45° from the Long Boom. (Change Edge Vector from (0.0, 0.0, 0.5) to (0.0, 0.5, -0.5).

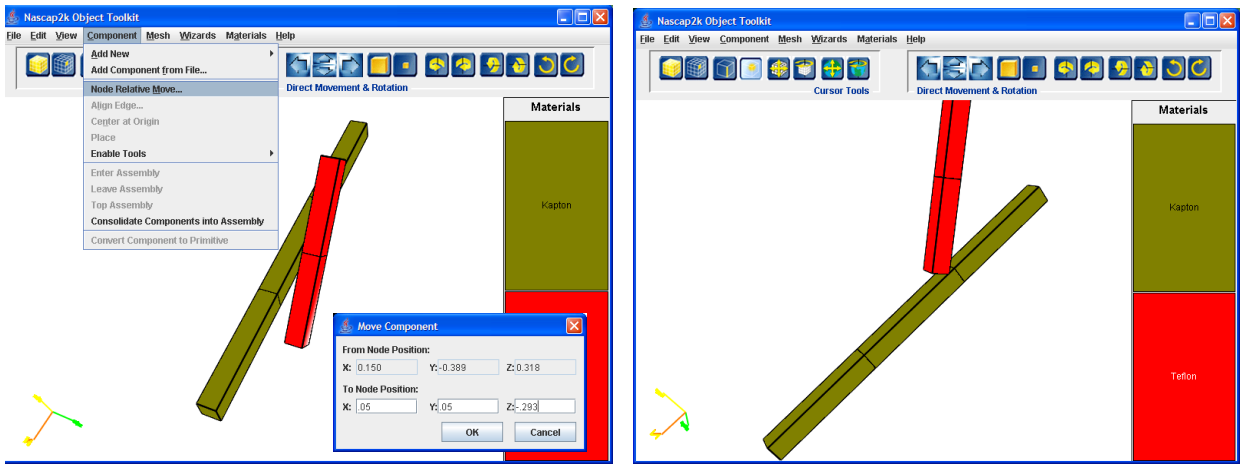


Figure 110. Moving the Short Side of the Y into Position by Going to the Component Menu and Selecting "Node Relative Move." The Bottom Corner Node Is Moved to (0.05, 0.05, -0.293).

Approved for public release; distribution is unlimited.

To make a compatibly meshed object, the nodes of the two Booms need to be coincident. First, go to the **Component** menu and select "Consolidate Components into Assembly" to combine the two Booms into a single component, which allows the mesh editing commands to be used to combine the parts. Then go to the **Mesh** menu and select "Combine Nodes" to move the two nodes on the long Boom to be coincident with the corresponding nodes on the end of the short Boom. Figure 111 illustrates these two steps.

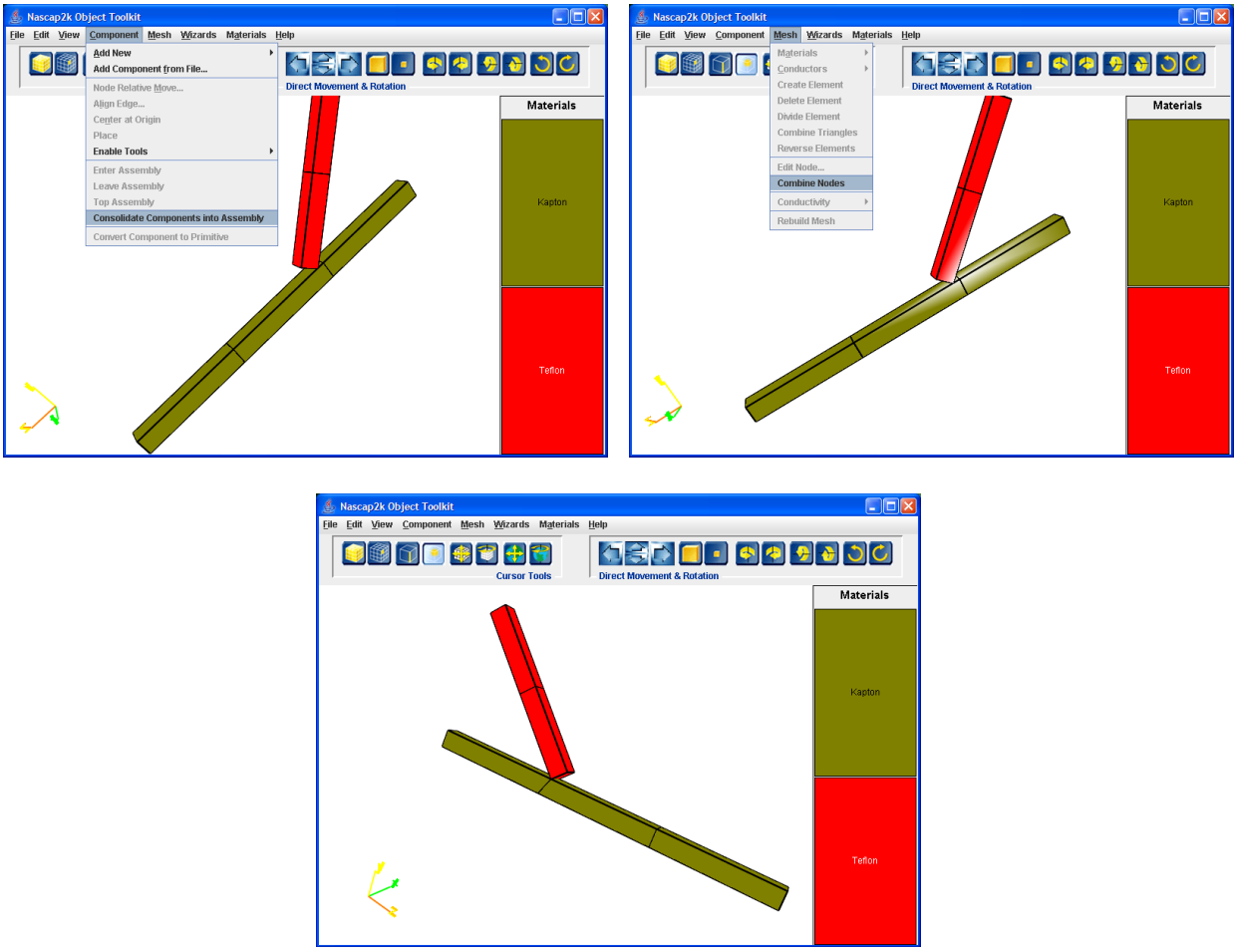


Figure 111. Consolidating Booms into a Single Assembly and Moving Nodes to Coincide.

The next step is to extend the other side of the short Boom to meet the long Boom, move the long Boom nodes to be at the join, and eliminate the extra elements. It is easiest to remove the extra elements first (Figure 112). Then move the end nodes of the short Boom to be at Y = 0.05 and Z = -0.151, shown in Figure 113. Finally, move the nodes on the long Boom to be coincident with those of the short Boom (Figure 114). The Y is now complete.

Approved for public release; distribution is unlimited.

Figure 112. Deleting Elements in Preparation for Completing Y.

Approved for public release; distribution is unlimited.

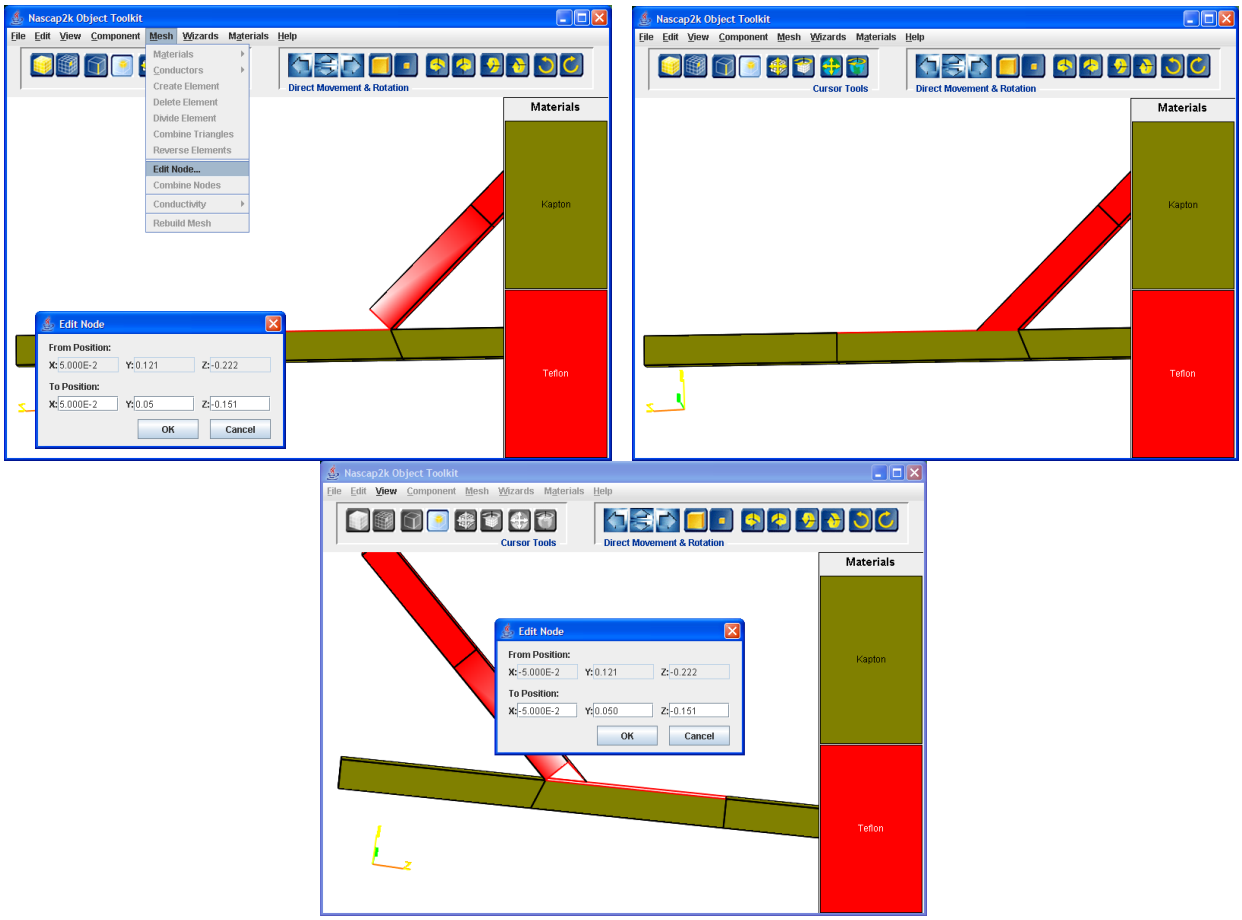


Figure 113. Moving End Nodes of the Short Boom to Be Along Face of the Long Boom. In These Figures, Perspective Is Turned Off.

Approved for public release; distribution is unlimited.

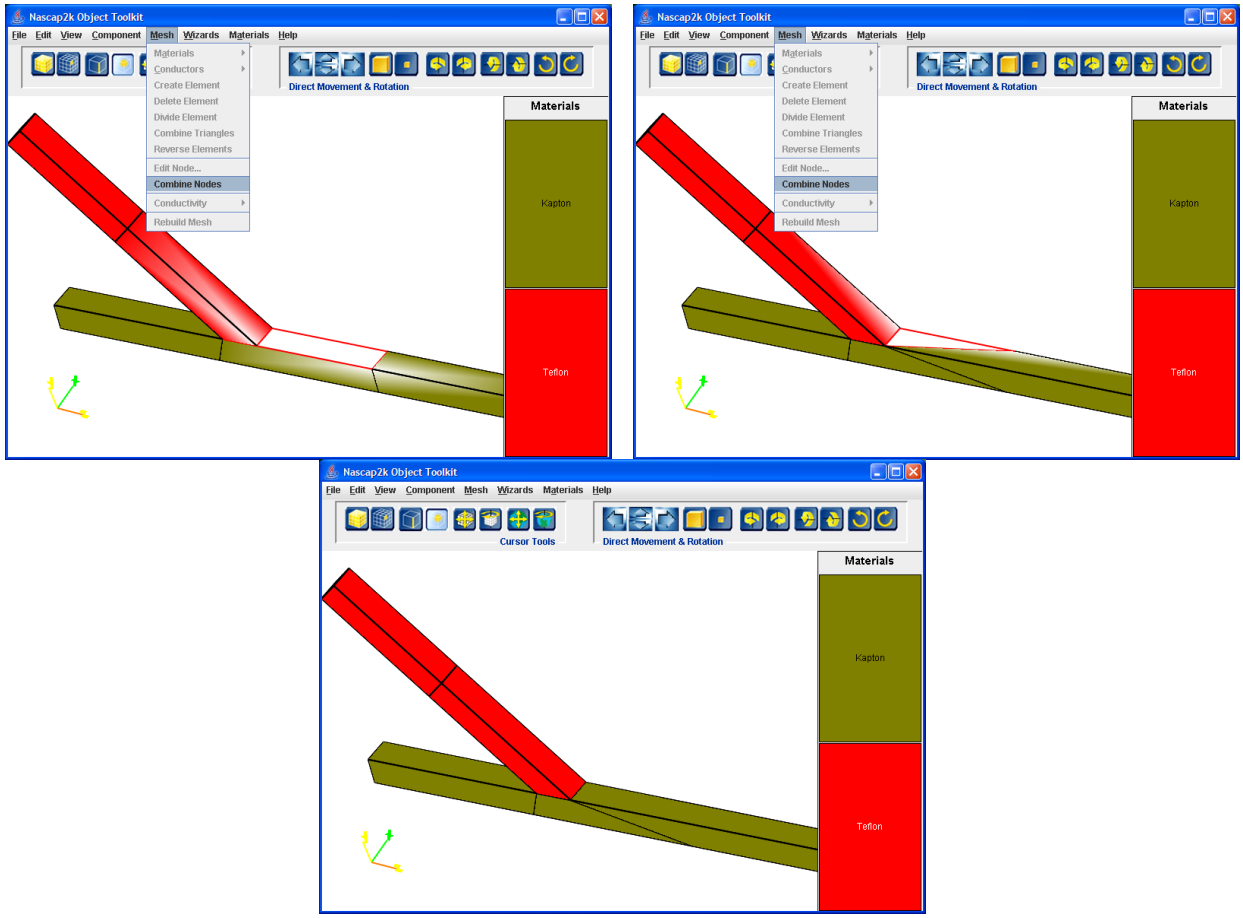


Figure 114. Moving Nodes of Long Boom to Coincide with Nodes on Short Boom.

Generally, it is desirable for elements to be approximately rectangular. A final step is to move the nodes on the back of the long Boom to have the same Z value as those in common with the short Boom (Figure 115).

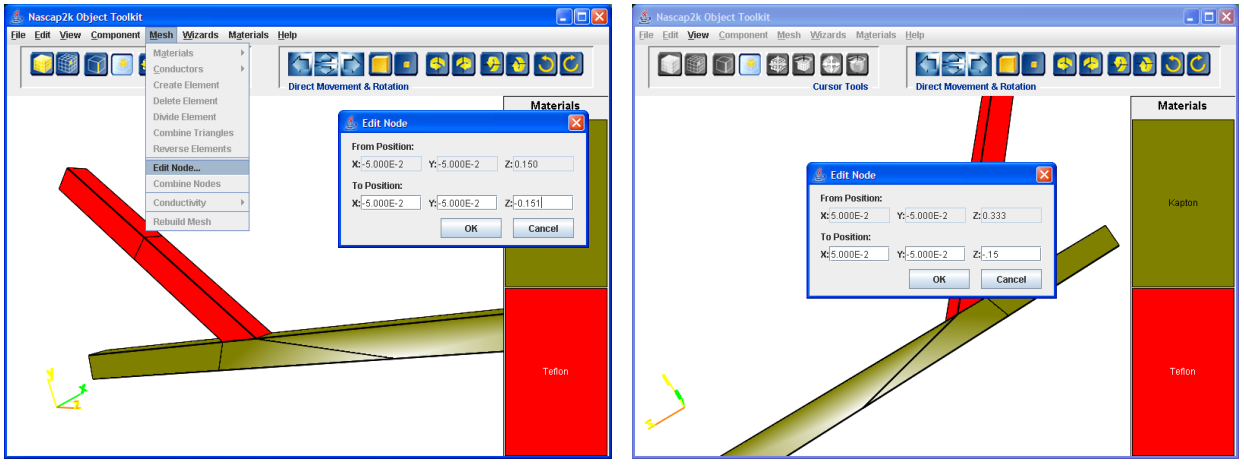


Figure 115. Moving Nodes on Long Boom So That All Elements Are Approximately Rectangular.

The final object is shown in Figure 116.

Approved for public release; distribution is unlimited.

Figure 116. Resulting Y Shape Created from Two Booms.

22 Using *Gmsh* to Create Finite Element Model from STEP File

Gmsh is an open source three-dimensional finite element mesh generator with a built-in computer aided design engine and post-processor. It is capable of reading a geometric description of an object in the form of a STEP (.stp) file, and creating and exporting a finite element surface model of that object suitable for import into *Object Toolkit*. The following provides a step-by-step example of meshing a geometric object using *Gmsh* and importing the mesh into *Object Toolkit*. Downloads and more thorough tutorials on *Gmsh* can be found at http://gmsh.info.

Note that all surface elements of objects imported in external formats are of a single material and conductor number. Surface materials and conductor numbers are assigned in *Object Toolkit*.

22.1 *Gmsh* Interface

On startup the *Gmsh* user interface appears as shown in Figure 117. The **File** menu is used to import the geometric model and later to export the finite element model. The **Mesh** module is used to create the finite element surface mesh. Among other things, the **Tools** menu gives access to options that control the size of the elements produced.

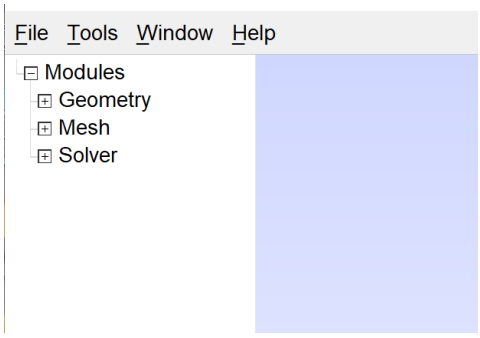


Figure 117. Gmsh user interface.

Approved for public release; distribution is unlimited.

22.2 Open a Geometric Model

On the **File** menu, choose “Open”, and browse to the file containing the geometry to be meshed. This example uses the file football.stp, which is included with the *Nascap-2k* distribution. The soccer ball appears as shown in Figure 118. It can be rotated with the mouse. The image consists of points and edges defining pentagonal and hexagonal surfaces, which are not suitable for *Nascap-2k.*

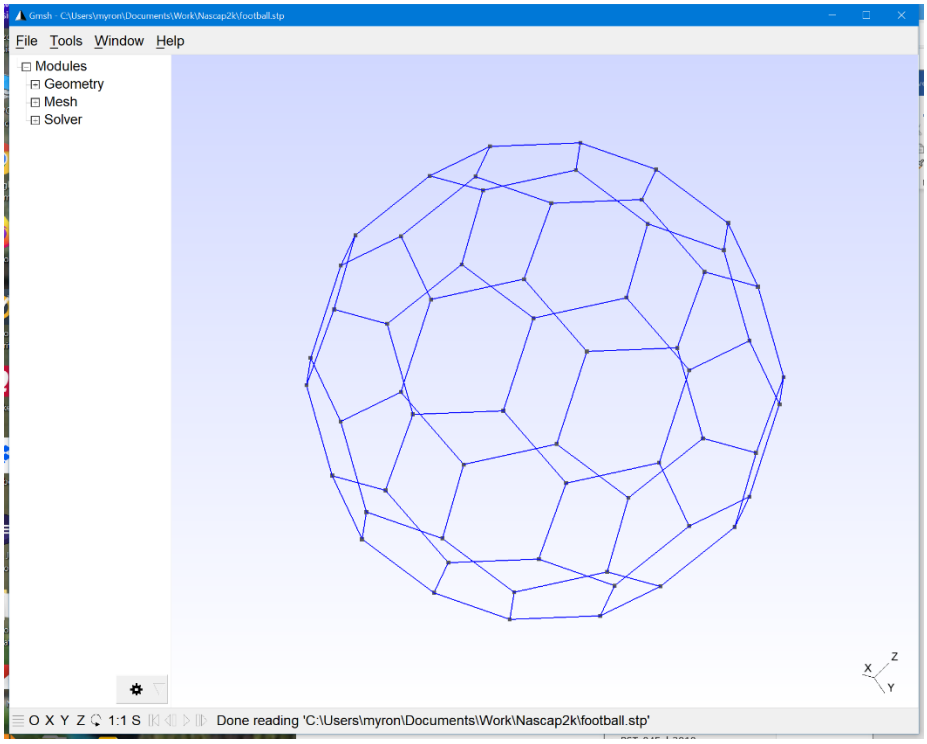


Figure 118. The imported object football.stp, as initially shown in Gmsh.

22.3 Create a Mesh on the Object

To create a mesh on the object, (1) expand the Mesh module by clicking the “+” sign, and (2) click “2D” in the menu that appears (Figure 119). This creates a mesh of triangular elements, as shown in Figure 119.

To convert the triangle mesh into a quadrilateral mesh, choose “Recombine 2D” in the **Mesh** Module menu. The resulting quadrilateral mesh is shown in Figure 120.

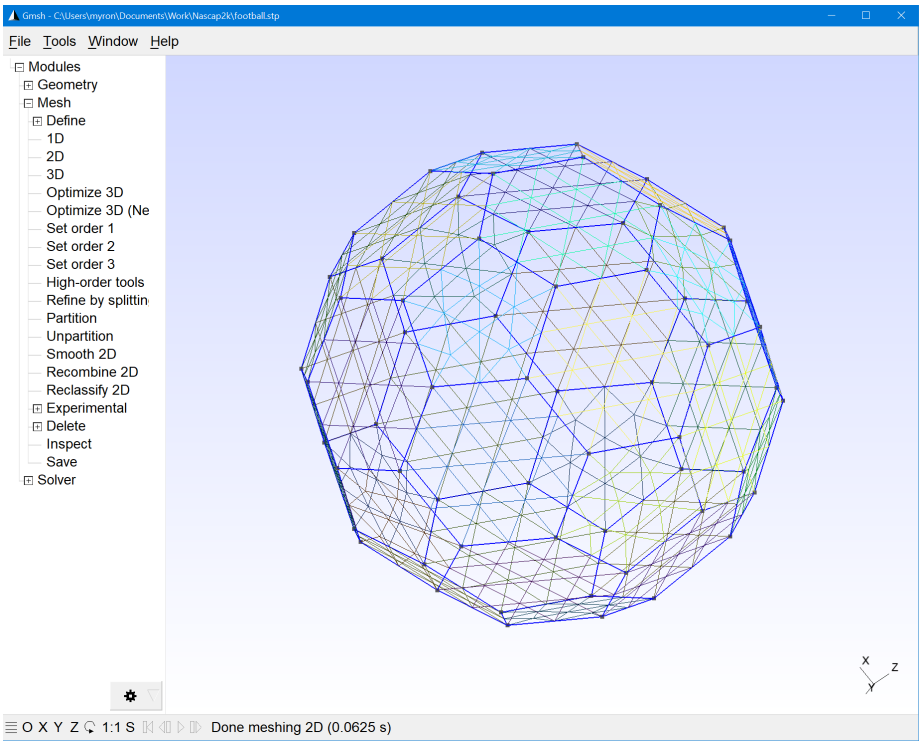


Figure 119. The expanded **Mesh** menu, and the triangular mesh created on the soccer ball by the “2D” command.

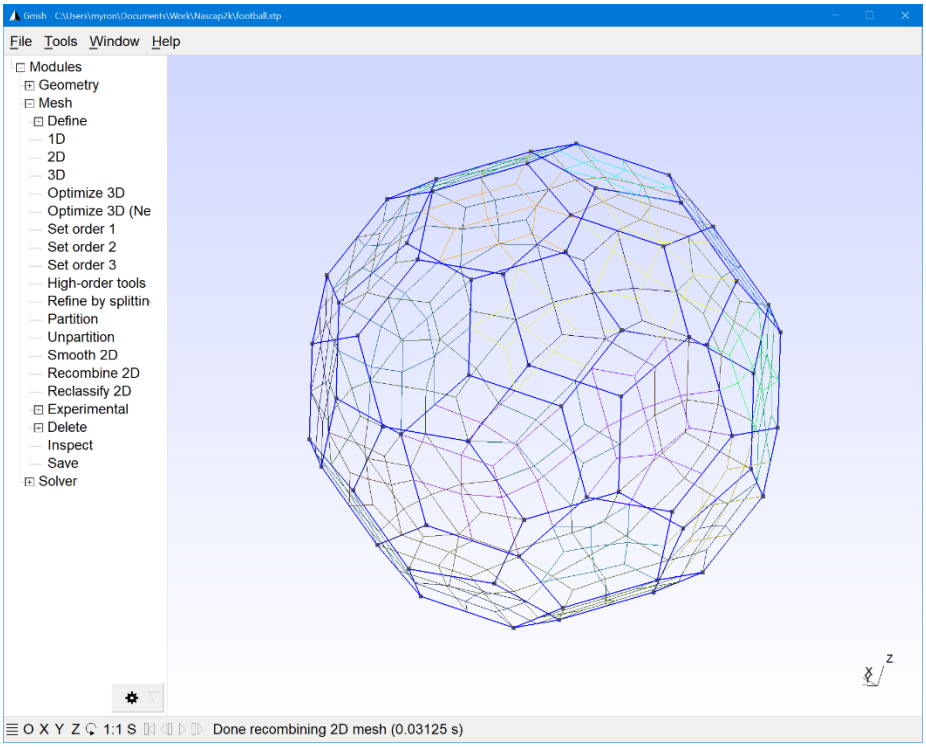


Figure 120. The soccer ball with a quadrilateral mesh created using the “Recombine 2D” command.

Approved for public release; distribution is unlimited.

22.4 Export the Mesh and Import to *Object Toolkit*

The final step is to export the mesh to a file format that can be read by *Object Toolkit*. Selecting the "Export" menu item under the **File** menu brings up the **Export** dialog shown in Figure 121. Expanding the "Save as type:" combo box gives a large selection of Mesh file formats from which to choose. The native *Gmsh* MSH (.msh) format works well. *Object Toolkit* also accepts the *I-DEAS* Universal (.unv) and *Nastran* Bulk Data File (.bdf) formats. Check "Save all elements" on the dialog that appears. The filename entered by the user should include the extension, as *Gmsh* does *not* add the file extension automatically.

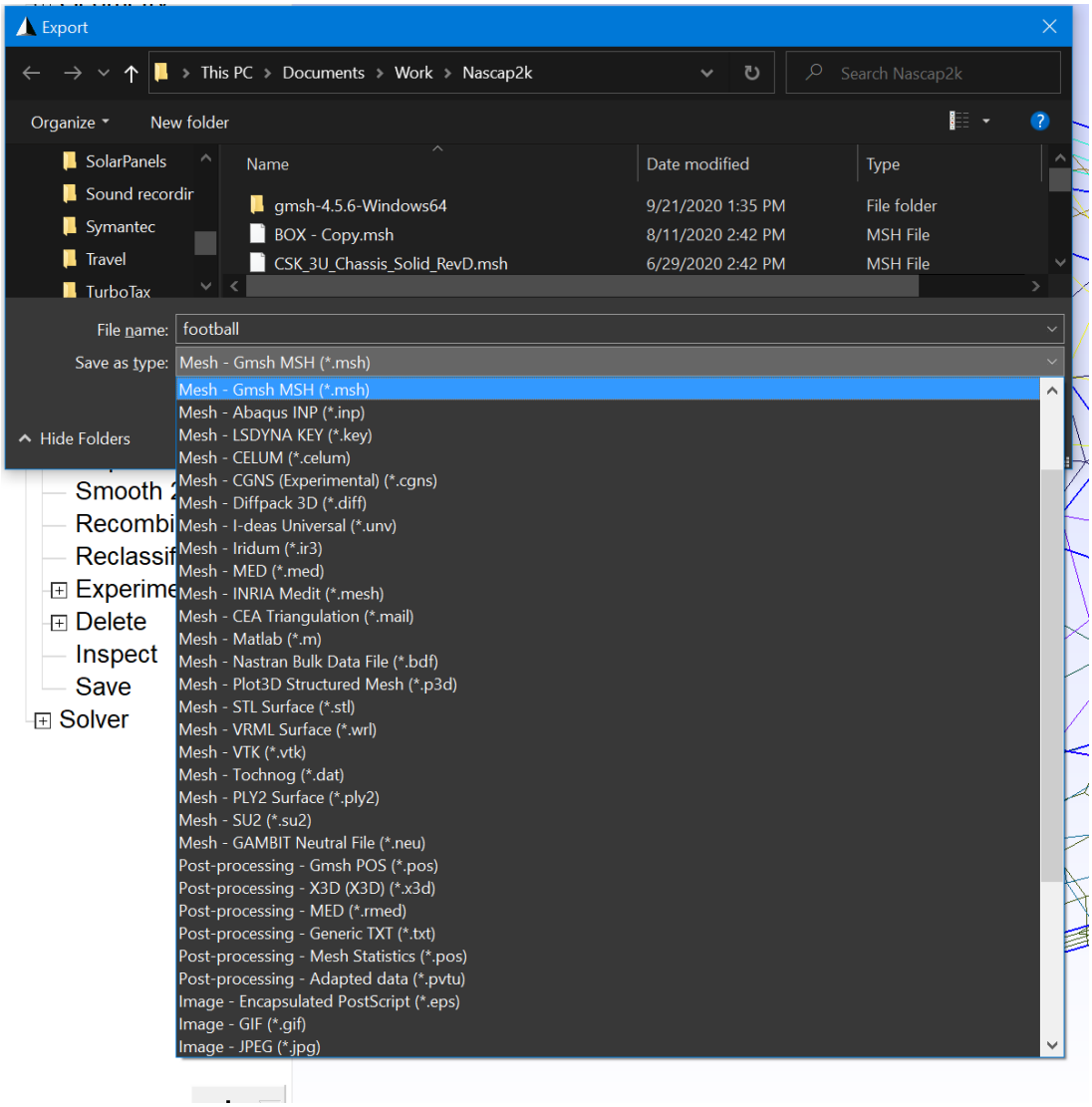


Figure 121. Gmsh Export dialog.

To import the exported file into *Object Toolkit*, select "Add Component from File…" from the **Component** menu. In the **Open** dialog box that appears, expand the "Files of Type:" combo box and select the type of file exported. Then navigate to the file and open it. The imported soccer ball is shown in Figure 122.

Approved for public release; distribution is unlimited.

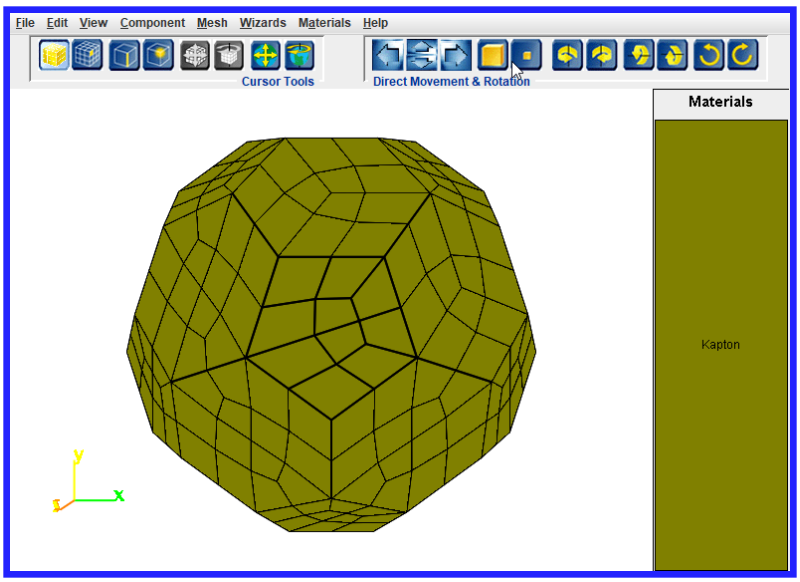


Figure 122. Meshed soccer ball as imported into Object Toolkit.

22.5 Common Manipulations

In some cases, after import some or all of the elements of the object are reversed, i.e., surface elements are pointing toward the interior of the object. In *Object Toolkit* you can see through the reversed surface elements on the near side to view the reversed surface elements on the far side. For the case that all elements are reversed, select the entire component (using the component selection tool, which is the leftmost icon) and choose "Reverse Elements" from the **Mesh** menu. If only some of the elements are reversed, select the reversed elements using the element selection tool and choose "Reverse Elements" from the **Mesh** menu. (Also see Section 13.1.)

If the object was defined in units other than meters, it needs to be scaled. The soccer ball was defined in millimeters, which you can tell by using *Object Toolkit*'s "Mouse Position" dialog. To convert the object to meters, select the component and choose "Edit Component…" from the **Edit** menu. (If the name is not "Primitive", use "Convert Component to Primitive" from the **Component** menu.) The object can be scaled in the X, Y, and Z directions. For the soccer ball, enter "0.001" in all three boxes, click "OK", and then zoom in to view the object. (Also see Section 13.3.)

If the object is off-center, it may be centered by selecting the object and then choosing "Center at Origin" on the **Component** menu. (Also see Section 7.)

Material names and conductor numbers can be assigned to elements by first selecting the elements and then choosing the "Materials" and "Conductors" menu items from the **Mesh** menu. (Also see Sections 10.1 and 15.1.)

22.6 Tweaking the Mesh Size

To create a finer or coarser mesh than *Gmsh* provides by default, use the **Options** menu shown in Figure 123. To reach this menu, select "Options" from the **Tools** menu, select "Mesh" in the left pane, and then select the **General** tab. The options of most interest are the two numbers

designated “Min/Max element size”, whose defaults are shown in Figure 123. As shown in Figure 119, Figure 120, and Figure 122, this produces one additional node on each edge of the soccer ball, and ten quadrilateral surface elements in each pentagon.

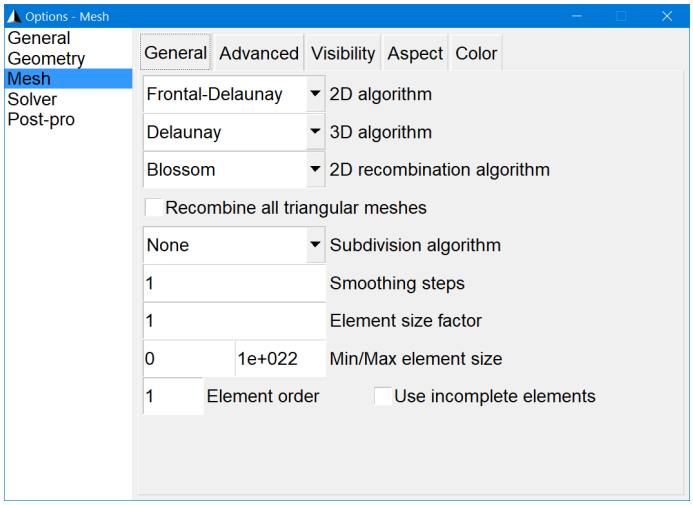


Figure 123. Options menu used for adjusting mesh size.

The mesh can be deleted by clicking the “settings” icon (gear-like object in Figure 120) at the bottom of the left panel and then choosing “Reset Database.” Then, a new mesh can be created with different “Min/Max element size” parameters.

To obtain a coarser mesh, the minimum element size can be set to 150 mm, so the “Min/Max element size” becomes [150, 1e+022]. With this setting no additional nodes are added, as seen on the left in Figure 124, producing only three surface elements in each pentagon or hexagon.

To obtain a finer mesh, the maximum element size can be set to 40 mm, so the “Min/Max element size” becomes [0, 40]. With this setting, two additional nodes are added to each edge, producing twenty-four surface elements in each pentagon, as seen on the right in Figure 124.

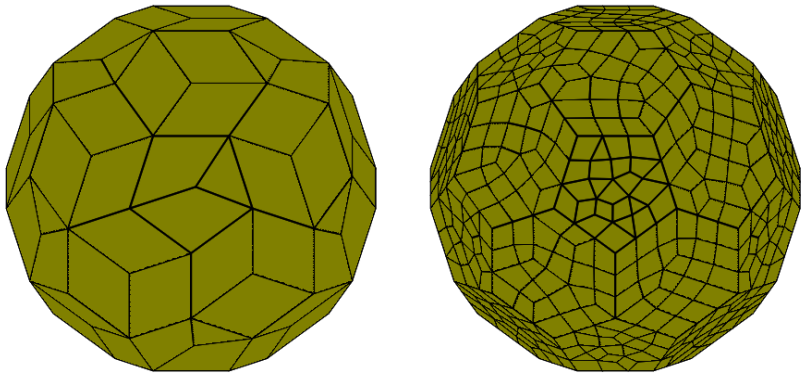


Figure 124. Coarse gridded (left) and fine gridded (right) soccer balls as shown in *Object Toolkit*. (Compare with Figure 122.)

Approved for public release; distribution is unlimited.

23 Spacecraft for *EPIC*

There are three main differences to keep in mind when creating objects for *EPIC*. First, it is generally *better not to group the components* together but instead to *move the components* into position. Second, it is necessary to assign subsystem names to the components for use by *EPIC*. A subsystem in *Object Toolkit* corresponds to an object in *EPIC*. In *EPIC* (and *MEM),* subsystems can rotate and point at something. Third, the object must be defined in the coordinate system that *EPIC* expects. The Z direction is toward the earth and the X direction is the direction of motion. It should also be noted that an element in *Object Toolkit* corresponds to a surface in *EPIC*. For additional details, see the *EPIC User's Manual*. In this example, we build the object shown in Figure 125, which has a thruster and whose solar arrays point at the sun.

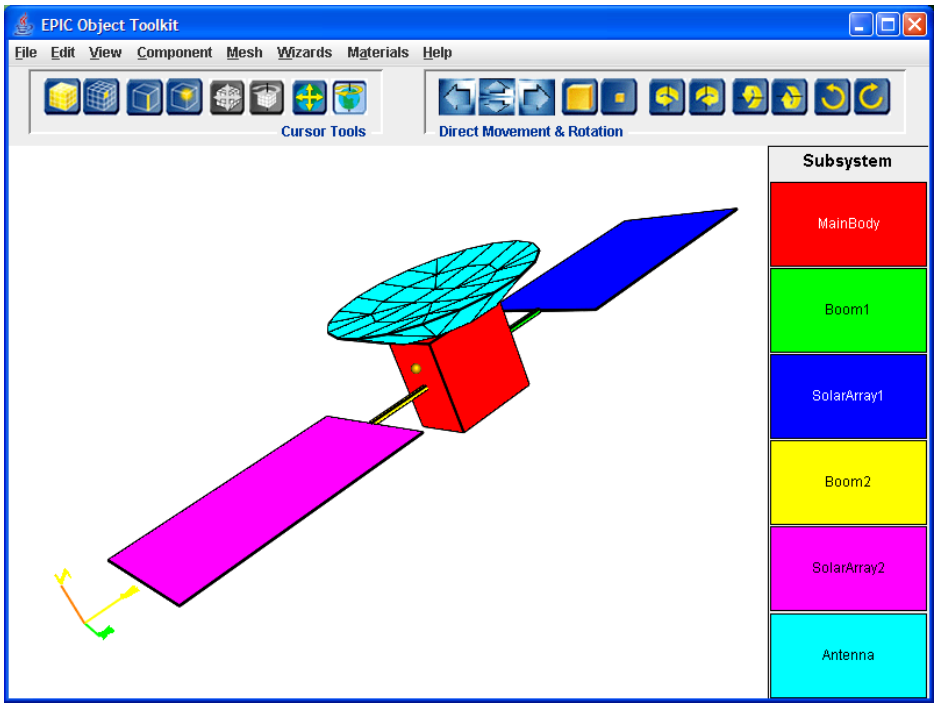


Figure 125. *EPIC* Object Colored by Subsystem Name.

23.1 Main Body

We use a Box as the main body as in previous examples. Create a Box by going to the **Component** menu, selecting "Add New" and choosing "Box" from the fly-out menu. Make the Box 1.76 m by 2.64 m by 2.64 m in size. Specify that the +X and –X surfaces are Teflon, that the +Y and –Y surfaces are OSR, and the +Z and –Z surfaces are Black Kapton. Set the number of zones to 1 for each direction because the surfaces do not need to be zoned up for *EPIC*. Figure 126 shows the **Box** dialog box with the appropriate values and the resulting Box.

Approved for public release; distribution is unlimited.

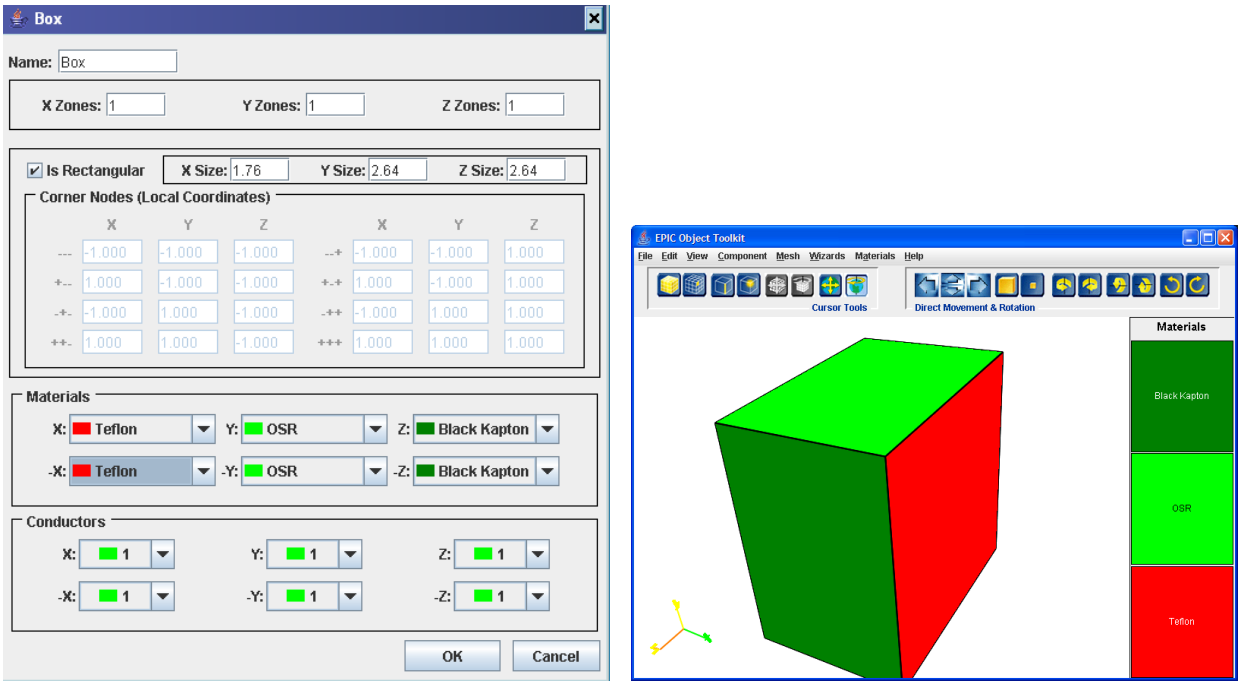


Figure 126. Specifying a Box to Represent the Main Body.

Assign a subsystem name to the Box by selecting it, going to the **Component** menu, selecting "Assign Subsystem" and choosing "Other" from the fly-out menu, as shown in Figure 127. Set the name of the subsystem to MainBody. To display the object color-coded by the subsystem names, go to the **View** menu and select "Subsystem" as shown in Figure 128.

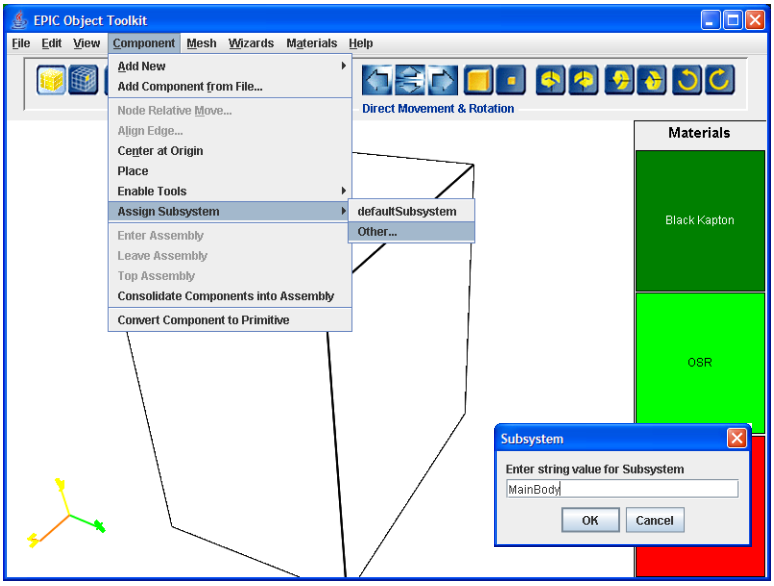


Figure 127. Assigning a Subsystem Name to the Box.

Approved for public release; distribution is unlimited.

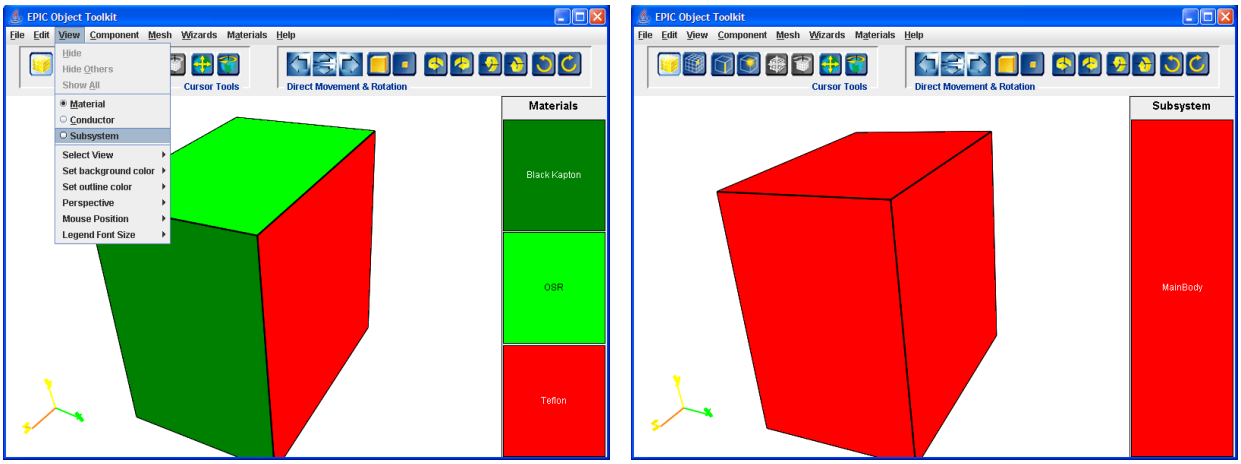


Figure 128. Displaying the Object Colored by Subsystem Name.

23.2 Adding a Boom

Add a Boom by going to the **Component** menu, selecting "Add New" and choosing "Boom" from the fly-out menu. Again, set zoning to be one element in each direction. Make the Boom 2 m long and 0.1 m in the other two directions and finally, make the surface Kapton (see Figure 129). Select the newly added Boom and then assign the subsystem name Boom1 as shown in Figure 130. Go to the **Wizard** menu and use the "Element to Element Move Wizard" to move the Boom to the +Y side of the main body. Note that this wizard *moves* the Boom rather than *joining* it. Select the +Y side of the main body as the fixed element and select one end of the Boom as the moving element. The Boom moves to the center of the +Y side as shown in Figure 131.

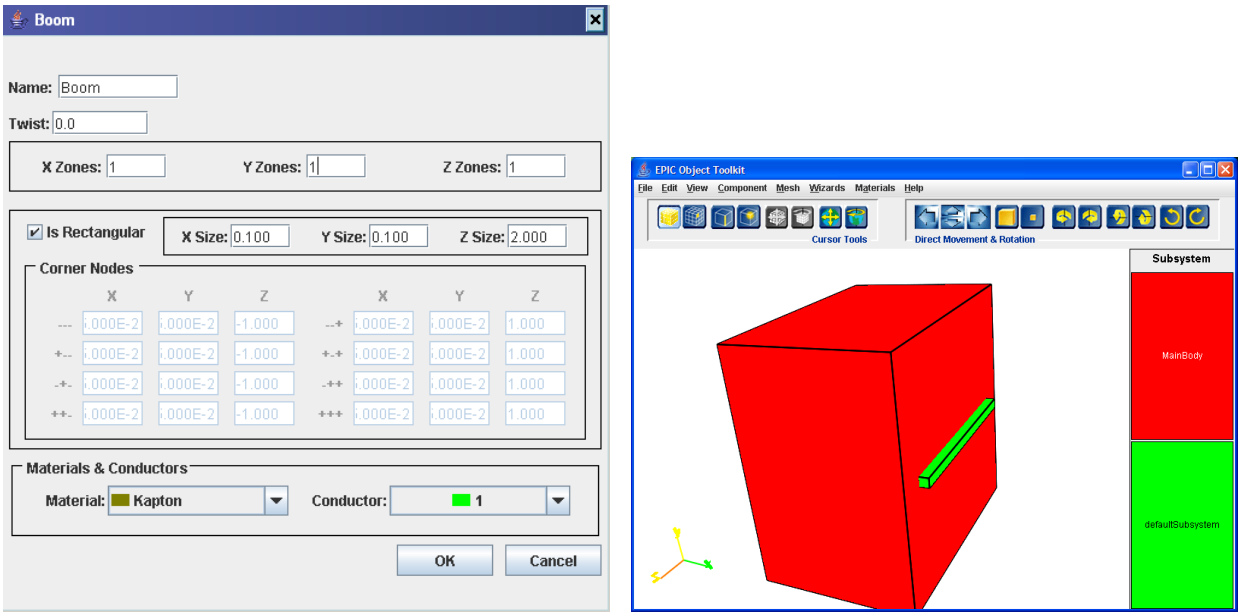


Figure 129. Boom Component for *EPIC* Object.

Approved for public release; distribution is unlimited.

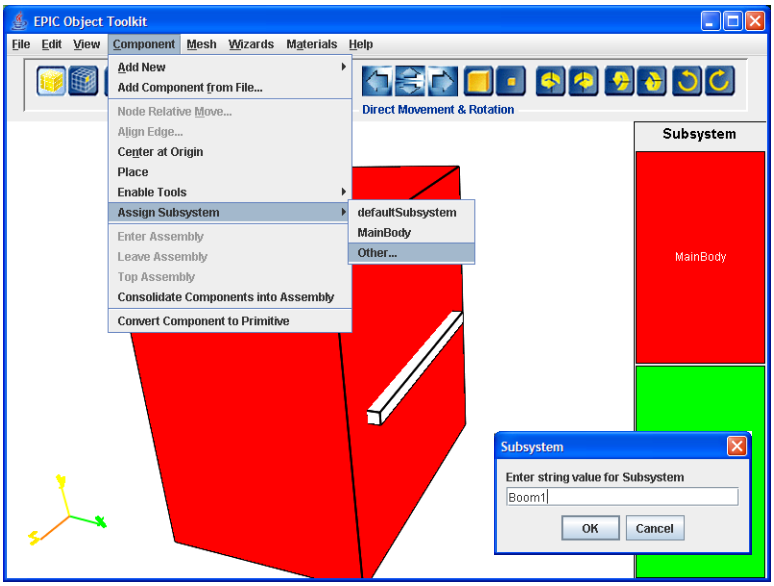
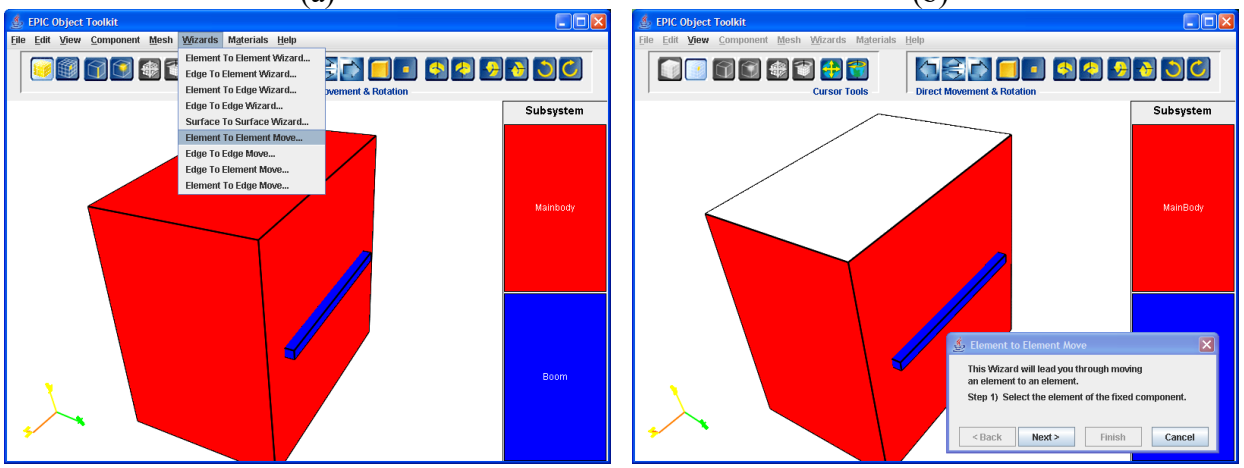


Figure 130. Assigning a Subsystem Name to the Boom.

(a) (b)



(c) (d)

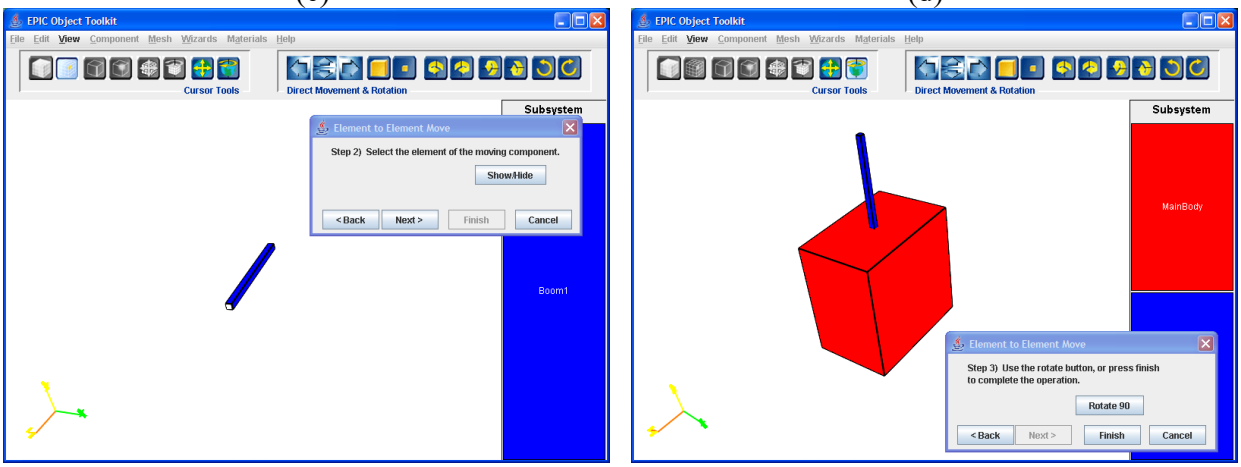


Figure 131. Placing the Boom Next to the Box Using the “Element to Element Move Wizard.”

Approved for public release; distribution is unlimited.

23.3 Adding a Solar Array

Create a Box to use as the solar array by going to the **Component** menu, selecting "Add New" and choosing "Box" from the fly-out menu. Make a 4.4 m by 7.0 m by 0.01 m Box, covered with Solar Cells on the +Z side and Black Kapton on the other sides, as shown in Figure 132. Again, set the component to be one element in each direction. Select the newly added Box and assign the subsystem name SolarArray1 as shown in Figure 133. Use the "Element to Element Move Wizard" to move the array to the end of the Boom. It is important to *move* the array rather than *joining* it because it will rotate. Select the end of the Boom as the fixed element and then select one end of the array as the moving element. The array reorients and moves to the end of the Boom as shown in Figure 134.

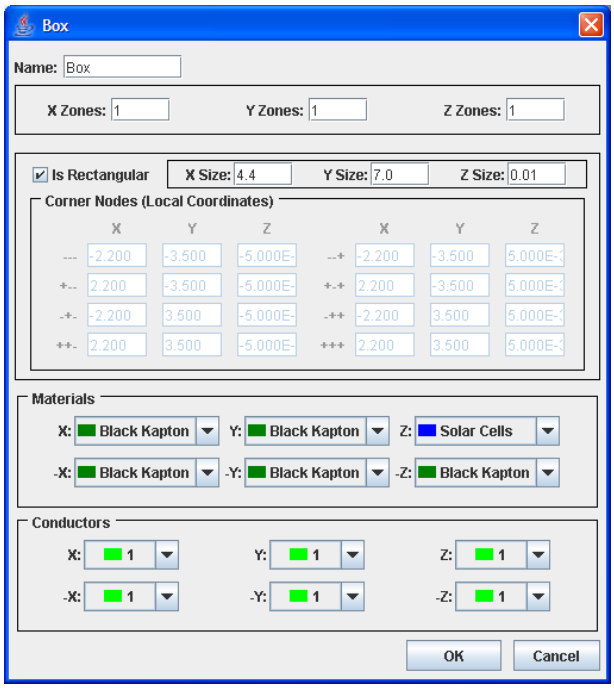


Figure 132. Solar Array for *EPIC* Object.

Approved for public release; distribution is unlimited.

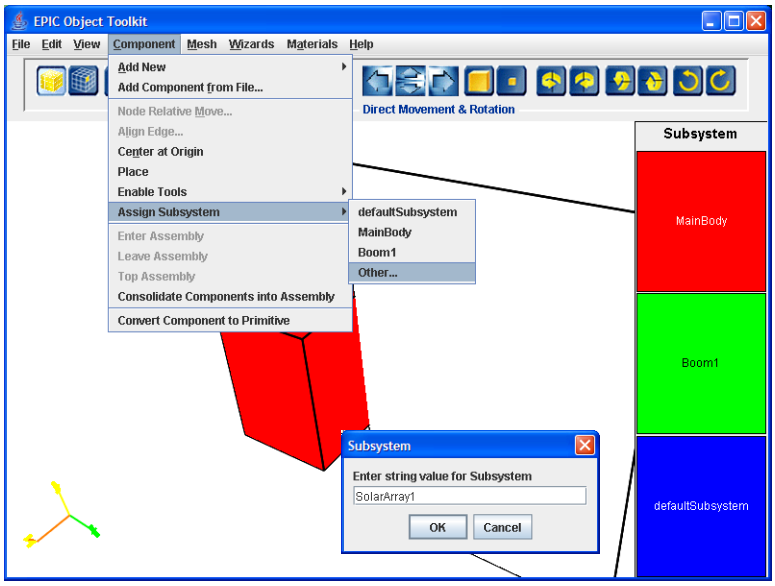


Figure 133. Assigning a Subsystem Name to the Solar Array.

Approved for public release; distribution is unlimited.

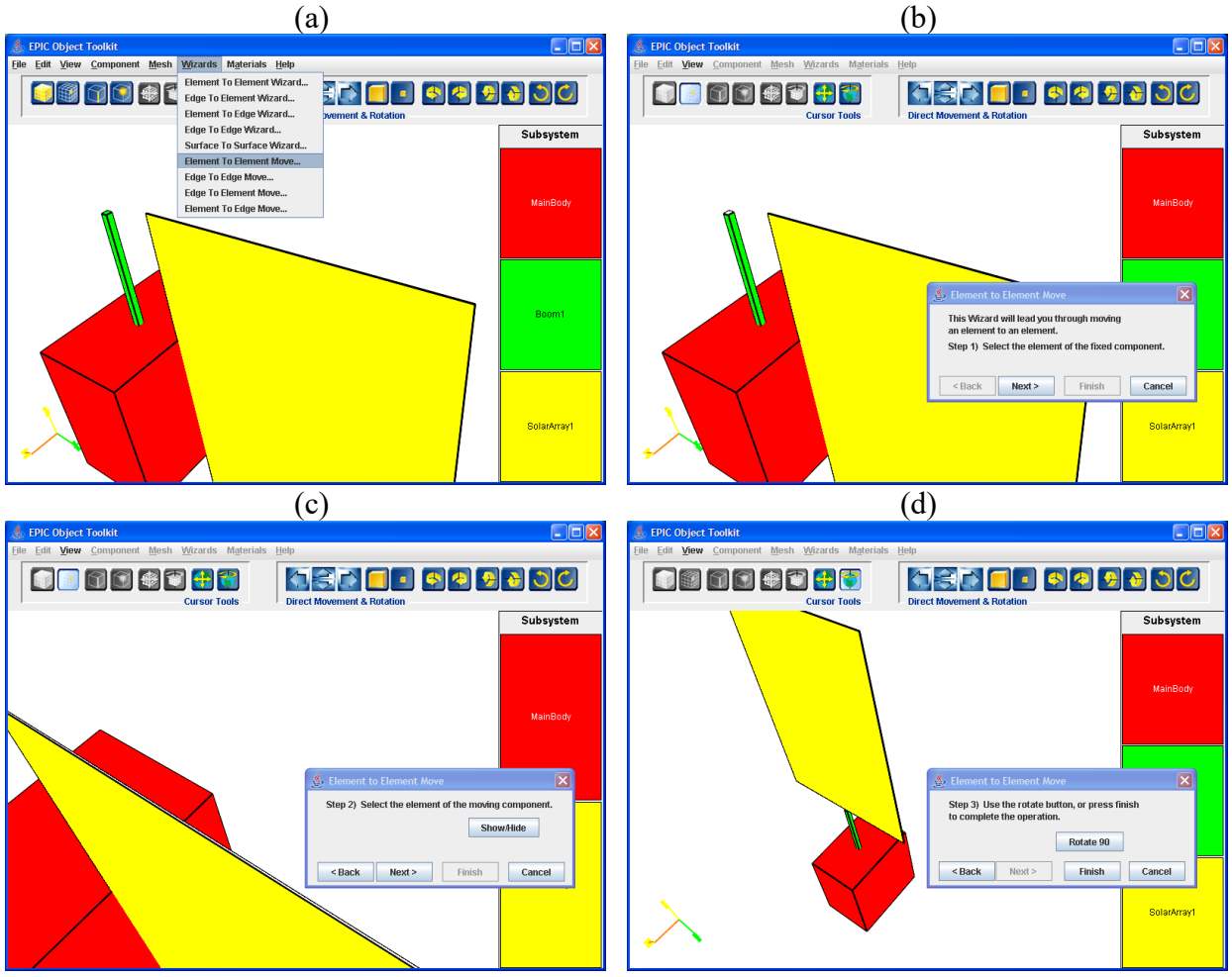


Figure 134. Placing the Solar Array at the End of the Boom Using the "Element to Element Move Wizard."

23.4 Adding the –Y Solar Array

Add the other support Boom and solar array in the same way, this time attaching the support Boom to the –Y side of the body and naming the objects Boom2 and SolarArray2. The object should now appear as shown in Figure 135 and Figure 136. If the solar arrays are facing the wrong way, they can be reversed by changing the original definition. The color assigned to a subsystem depends on the order the names are assigned, is not saved, and is not under user control.

Approved for public release; distribution is unlimited.

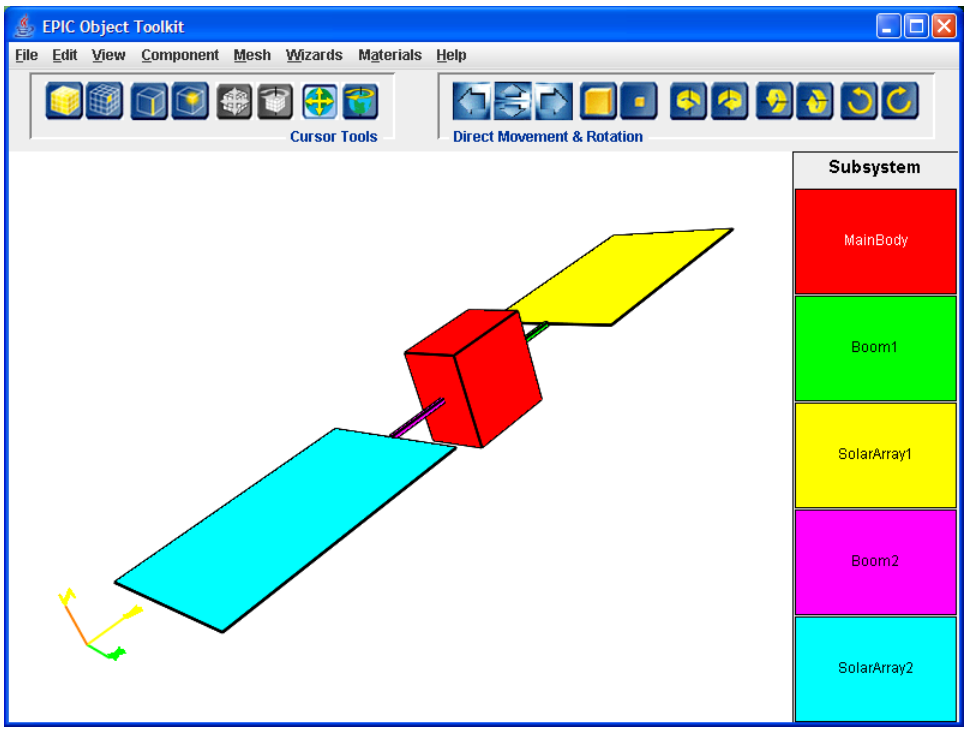


Figure 135. *EPIC* Object Colored by Subsystem Name, Showing Both Solar Arrays.

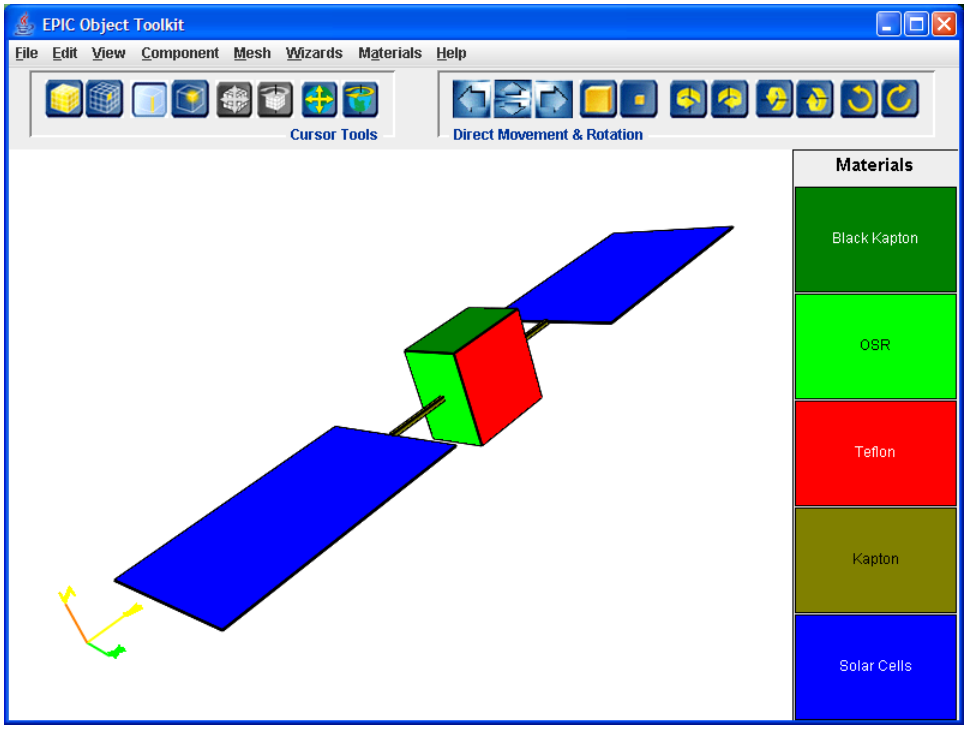


Figure 136. *EPIC* Object Colored by Material Name.

Approved for public release; distribution is unlimited.

23.5 Adding a Thruster

Add a thruster located on the –Y surface of the main body and pointing in the –Y direction by going to the **Component** menu, selecting "Add New" and choosing "Thruster" on the fly-out menu. Place the thruster at the location (0.0, –1.4, 0.6), which is on the face pointing in the –Y direction. The (r, g, b) parameters on the **Thruster Properties** dialog box shown in Figure 137 indicate the color of the sphere used to indicate the thruster location. The *EPIC* object with the thruster is shown in Figure 138.

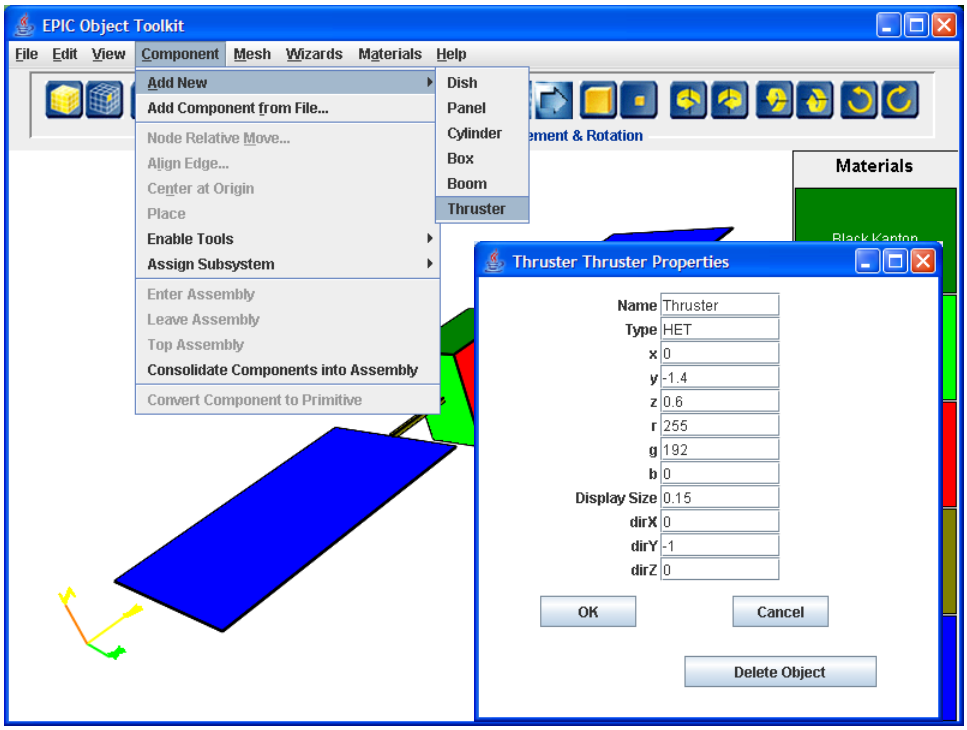


Figure 137. Thruster Definition for *EPIC* Object. Thruster Is Located at (0.0, -1.4, 0.6) and Is Pointed in the –Y Direction.

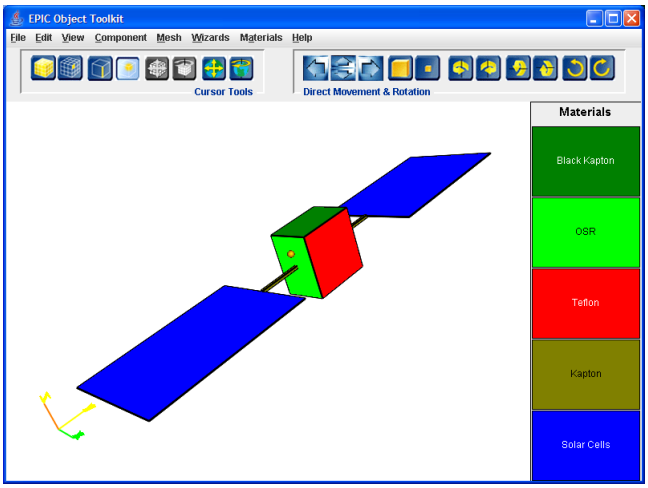


Figure 138. *EPIC* Object with Thruster. Thruster Is Located at (0.0, -1.4, 0.6) and Is Pointed in the –Y Direction.

Approved for public release; distribution is unlimited.

23.6 Setting the Solar Arrays to Track the Sun

To set the solar arrays to track the sun, go to the **Edit** menu, select "Subsystem," and choose "SolarArray1" from the fly-out menu. To set up a pointing object, first specify the vector in the unrotated system that is to point at something (here, the sun). In this case, that vector is the direction normal to the side of the solar panel covered with Solar Cells, or the +Z direction (0.0, 0.0, 1.0). The dialog box allows for the specification of one or two axes for the rotations. For this geometry, the solar arrays can only rotate around the Z axis, so set one rotation axis the Z axis, (0.0, 1.0, 0.0) and the other to either the same (0.0, 1.0, 0.0) or to zeros, (0.0, 0.0, 0.0). Because there is only one rotation axis, the fixed point can be any point on the rotation axis. If there were two axes of rotation, the fixed point would be the intersection of those two axes. Here (0.0, 0.0, 0.0) is used. Finally, set the target to be the sun. Figure 139 shows the **Subsystem: SolarArray1 Properties** dialog box.

Set the rotation of the other solar array in the same way. This object can now be used by *EPIC*, for example, to evaluate surface erosion and material redeposition on rotating solar arrays due to the plume of an ion thruster.

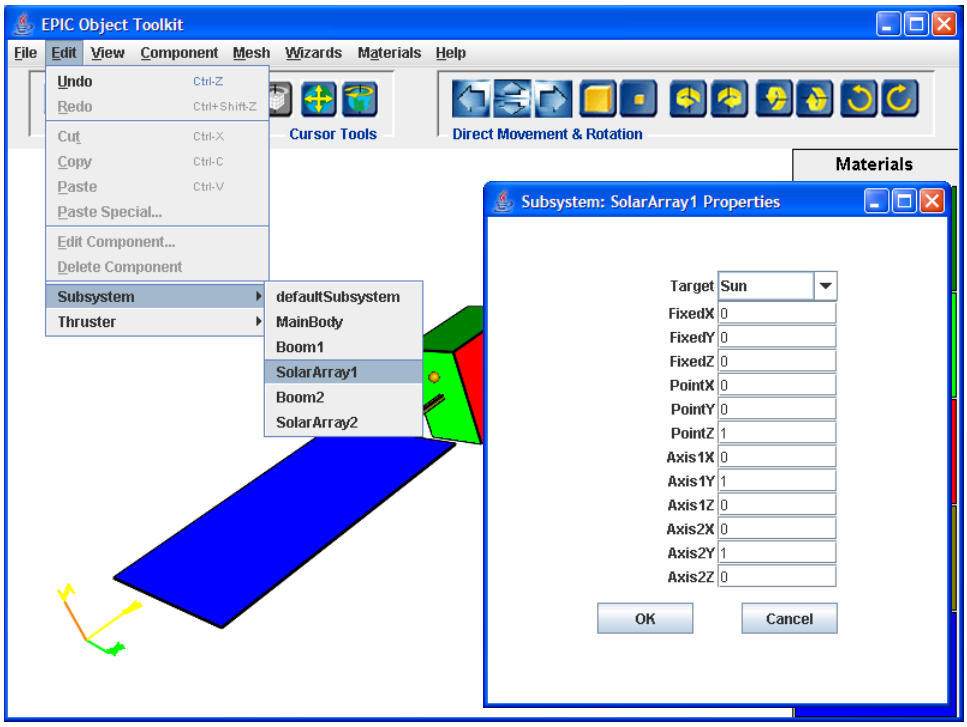


Figure 139. Setting SolarArray1 to Track the Sun. The Fixed Point Is (0.0, 0.0, 0.0). The Pointing Vector Is (0.0, 0.0, 1.0). The First Axis Is (0.0, 1.0, 0.0) and the Second Axis Is the Same.

Approved for public release; distribution is unlimited.

23.7 Adding an Antenna

Finally, we want to add a dish antenna on the +Z facing side. Go the **Component** menu, select "Add New" and choose "Dish" from the fly-out menu. Make a 6-m diameter Graphite Dish with five elements across, as shown in Figure 140. Select the newly added Dish and then assign the subsystem name Antenna in the same manner as for the other components. Now the antenna needs to be moved into position. Select a point on the base of the antenna, go to the **Component** menu, and choose "Node Relative Move" as shown in Figure 141. Enter (±0.44, ±0.44, –1.46) in the dialog box that appears, where the choice of the "+" or "–" sign depends on which of the four base nodes you choose.

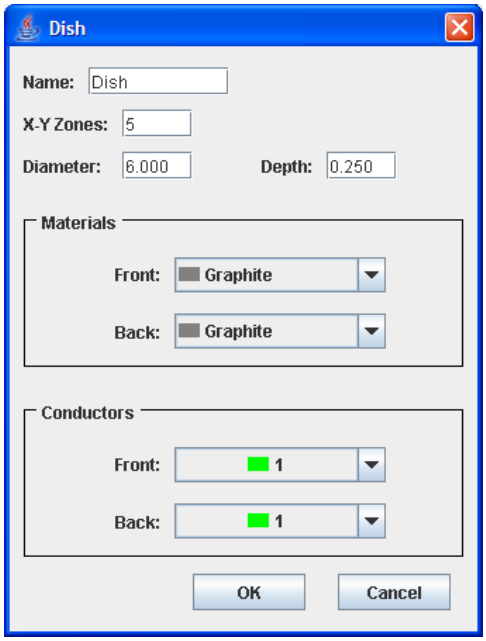


Figure 140. Dish Antenna for *EPIC* Object.

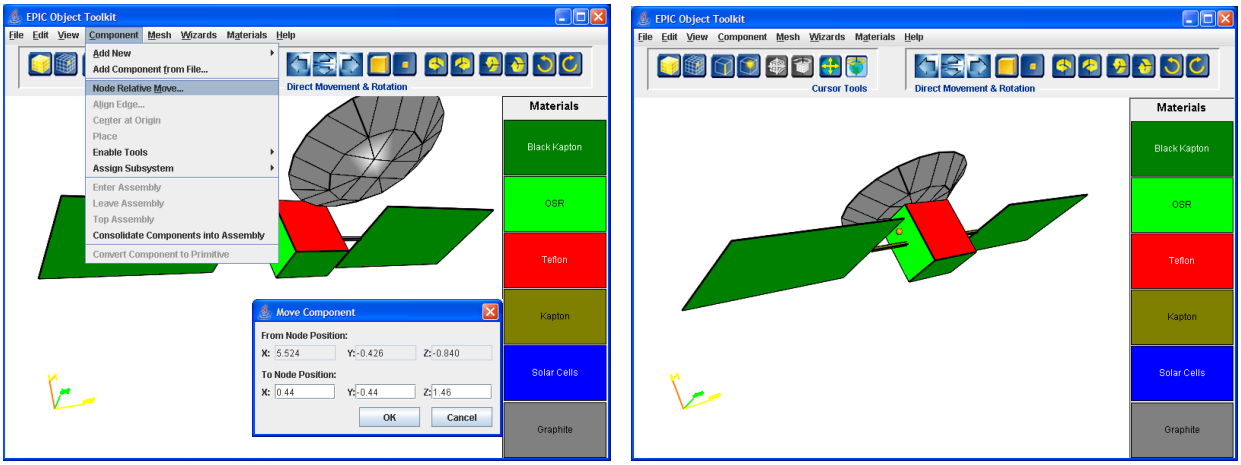


Figure 141. Moving Antenna to the +Z Side at (0.44, -0.44, 1.46).

Approved for public release; distribution is unlimited.

24 Spacecraft with an Orbit

In this section we assign an orbit to the object developed in the previous section in order to view the rotation of the solar arrays. The *MEM* code requires that the object include an orbit. This capability is only available when *Object Toolkit* is tailored for *MEM*. To assign the orbit go to the **Orbit** menu, select "NewOrbit," and choose "GeoCentric" on the fly-out menu. The **GeoCentric Orbit** dialog box appears, as shown in Figure 142. This dialog box allows you to specify the orbit and save and retrieve orbits. For this example, accept the default parameters by clicking "OK."

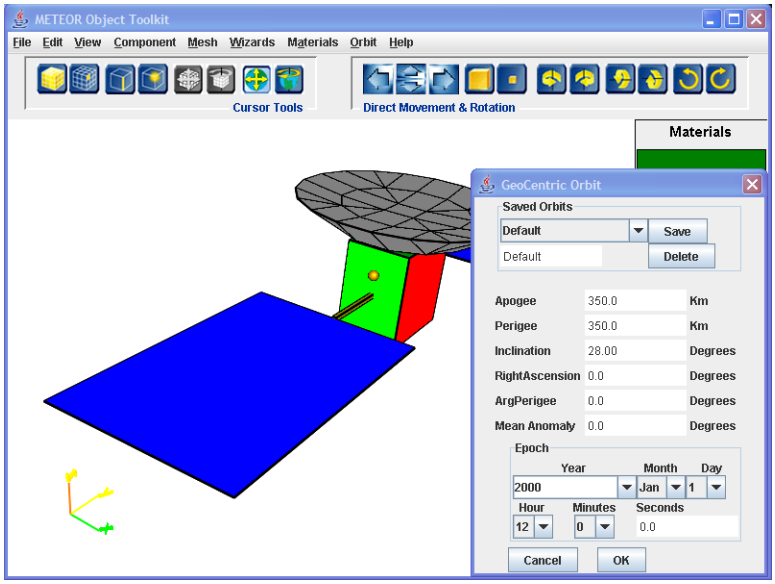


Figure 142. Geocentric Orbit Dialog Box.

To view the rotation of the solar arrays go to the **Orbit** menu, select "Time and Stabilization," and select "Pointing Rotations" on the dialog box that appears (Figure 143). The solar arrays rotate around the Z axis to point at the sun, as shown in Figure 144. You can modify the time on the **Time and Stabilization** dialog box and view the reoriented solar arrays. Note: *To edit the object, "No Rotations" must be selected on the* ***Time and Stabilization*** *dialog box.*

Approved for public release; distribution is unlimited.

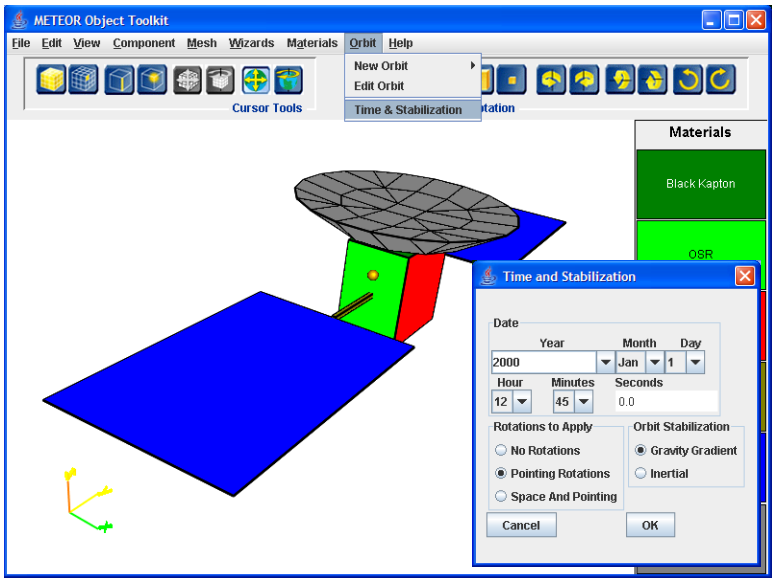


Figure 143. Applying Pointing Rotations Using the Time and Stabilization Dialog Box.

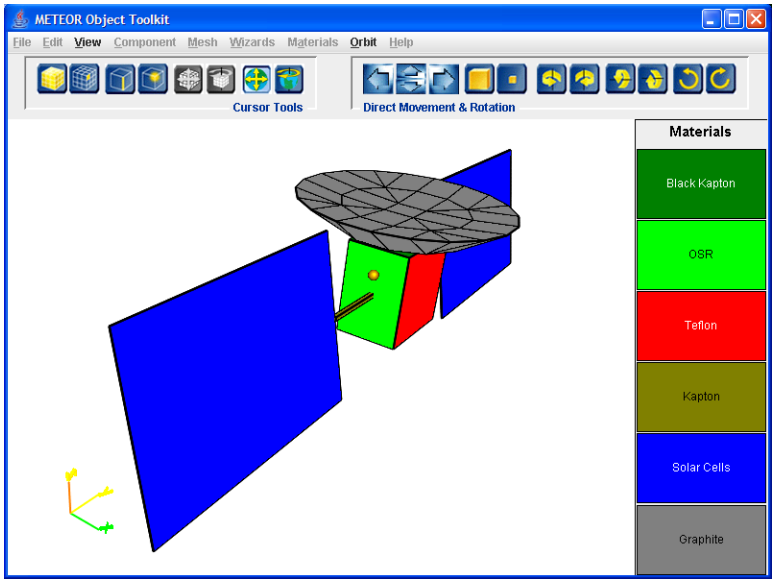


Figure 144. *EPIC* Object with Rotated Solar Arrays.

Approved for public release; distribution is unlimited.

REFERENCES

1. Mikellides, I.G., *et al.*, "Assessment of Spacecraft Systems Integration Using the Electric Propulsion Interactions Code (EPIC)," AIAA Paper 02-3667, 38th Joint Propulsion Conference, Indianapolis, Indiana, July 2002.

2. T.S.Kelso, "Celestrack.com" http://celestrack.com, accessed on February 20, 2014.

Approved for public release; distribution is unlimited.

Appendix A: Input file formats

A.1 *PATRAN* Neutral File

A *PATRAN* neutral file consists of data packets, each of which is two or more lines long. Each data packet consists of one header record followed by one or more data records. The first two characters of the first record is the packet type. *Object Toolkit* recognizes five packet types, 25, 26, 01, 02, and 99. Other packet types are ignored. Figure 145 shows the *PATRAN* neutral file for an eight-node, six-element cube.

Approved for public release; distribution is unlimited.

```
25       0       0       1       0       0       0       0       0
CUBE FOR SAMPLE PROBLEM
26       0       0       1       8       6       0       0       0
18-OCT-90   17:06:01     2.4
 1       1       0       2       0       0       0       0       0
 0.000000000E+00 0.000000000E+00 0.000000000E+00
1G       6       0   65535  000000
 1       2       0       2       0       0       0       0       0
 0.000000000E+00 0.000000000E+00 0.200000033E+00
1G       6       0   65535  000000
 1       3       0       2       0       0       0       0       0
 0.000000000E+00 0.200000033E+00 0.000000000E+00
1G       6       0   65535  000000
 1       4       0       2       0       0       0       0       0
 0.000000000E+00 0.200000033E+00 0.200000033E+00
1G       6       0   65535  000000
 1       5       0       2       0       0       0       0       0
 0.200000003E+00 0.000000000E+00 0.000000000E+00
1G       6       0   65535  000000
 1       6       0       2       0       0       0       0       0
 0.200000003E+00 0.000000000E+00 0.200000033E+00
1G       6       0   65535  000000
 1       7       0       2       0       0       0       0       0
 0.200000003E+00 0.200000033E+00 0.000000000E+00
1G       6       0   65535  000000
 1       8       0       2       0       0       0       0       0
 0.200000003E+00 0.200000033E+00 0.200000033E+00
1G       6       0   65535  000000
 2       1       4       2       0       0       0       0       0
       4       1       0       0 0.000000000E+00 0.000000000E+00 0.000000000E+00
       1       2       4       3
 2       2       4       2       0       0       0       0       0
       4       1       0       0 0.000000000E+00 0.000000000E+00 0.000000000E+00
       5       7       8       6
 2       3       4       2       0       0       0       0       0
       4       1       0       0 0.000000000E+00 0.000000000E+00 0.000000000E+00
       1       5       6       2
 2       4       4       2       0       0       0       0       0
       4       1       0       0 0.000000000E+00 0.000000000E+00 0.000000000E+00
       3       4       8       7
 2       5       4       2       0       0       0       0       0
       4       1       0       0 0.000000000E+00 0.000000000E+00 0.000000000E+00
       1       3       7       5
 2       6       4       2       0       0       0       0       0
       4       1       0       0 0.000000000E+00 0.000000000E+00 0.000000000E+00
       2       6       8       4
99       0       0       1       0       0       0       0       0
```

Figure 145. *PATRAN* Neutral File Defining an Eight-node, Six-element Cube.

The format of each line of the *PATRAN* neutral file that is used by *Object Toolkit* is as follows:

Header record format

Each data type starts with a header card in the format (I2, 8I8) with the following information:

IT = Packet type

ID = Identification number

IV = Additional identification

KC = Number of following records within the packet

N1—N5 = Additional integer values as needed

The format of the following records depends on the packet type.

Packet type 25: Title Card

The following record gives a description of the object.

Packet type 26: Summary Data

The header record contains the number of nodes in the N1 location and the number of elements in the N2 location. The following record contains the date and time of file creation.

Packet type 1: Node data

The header record gives the node number in the ID location. The second record specifies the global Cartesian x,y,z-coordinates of the node in meters. The content of the third record is ignored by *Object Toolkit*.

Packet type 2: Element data

The header record gives the element number in the ID location and the shape ID for this element in the IV location. *Object Toolkit* only recognizes two shapes, the triangle (shape ID=3) and the quadrilateral (shape ID=4). The second record is ignored. The third record holds the node numbers of the corners of this element.

Packet type 99: End of file

A.2 *NX I-DEAS* TMG ASCII VUFF File

An ASCII VUFF file consists of data packets, each of which consists of a string followed by data. The string specifies the packet type. *Object Toolkit* recognizes four packet types, `RDFVARIABLENAME`, `RDFGROUPNAMES`, `GRD`, and `NGR`. Other packet types are ignored.

The format of each line of the VUFF file that is used by *Object Toolkit* is as follows:

RDFVARIABLENAME records

Records that start with `RDFVARIABLENAME` specify general information. Of these, two are read. The number of nodes and the number of elements are specified in the records with the content

```
RDFVARIABLENAME %MAXNODE                    numberofnodes
```

and

```
RDFVARIABLENAME %MAXELEM                    numberofelements
```

respectively.

Approved for public release; distribution is unlimited.

RDFGROUPNAMES records

These records are used to specify the material names. The material name is taken to be the first four characters of the second entry. The record gives the element number for the first and last elements of the group and the increment between elements to be included. The last entry is ignored. Records specifying more elements than half of the total are ignored.

```
RDFGROUPNAMES NAME       e1         e2         delta         counter
```

GRD records: Node data

The record gives the node number and the global Cartesian x,y,z-coordinates of the node in meters.

```
GRD         nnn x y z
```

NGR records: Element data

The record gives the element number, the number of nodes, and the node numbers of the corners of this element. *Object Toolkit* only recognizes elements with 3 (triangle) or 4 (quadrilateral) nodes.

```
NGR  nnn     numnodes       n1       n2       n3       n4
```

A.3 *Gmsh* MSH Formatted File

A *Gmsh* MSH formatted text file consist of three of more sections. *Object Toolkit* reads the $MeshFormat, $Nodes, and $Elements sections. Other sections are ignored (including the $Entities section). Sections can be repeated, as long as each node is defined before it is used in an element definition. Figure 146 shows the MSH formatted file for an eight-node, six-element cube.

Approved for public release; distribution is unlimited.

```
$MeshFormat
4.1 0 8
$EndMeshFormat
$Entities
.
.
.
$EndEntities
$Nodes
8 8 1 8
0 1 0 1
1
-0.5 -0.7 1
0 2 0 1
2
-0.5 -0.7 0
0 3 0 1
3
-0.5 0.3 1
0 4 0 1
4
-0.5 0.3 0
0 5 0 1
5
0.5 -0.7 1
0 6 0 1
6
0.5 -0.7 0
0 7 0 1
7
0.5 0.3 1
0 8 0 1
8
0.5 0.3 0
$EndNodes
$Elements
6 6 1 6
2 1 3 1
1 4 2 1 3
2 2 3 1
2 5 6 8 7
2 3 3 1
3 1 2 6 5
2 4 3 1
4 8 4 3 7
2 5 3 1
5 2 4 8 6
2 6 3 1
6 7 3 1 5
$EndElements
```

Figure 146. *Gmsh* MSH Formatted File Defining an Eight-node, Six-element Cube.

Object Toolkit's requirements for the format of the sections used is as follows:

$MeshFormat Section

This section has one line. The first item on the line must be a real number equal to or greater than 4.1.

$Nodes Section

The first line of the $Nodes section must contain four integers, the number of entity blocks, the number of nodes, the minimum node tag, and the maximum node tag. *Object Toolkit* uses only the number of entity blocks and the maximum node tag. All nodes within the section must have a node tag no greater than the value given in this first line.

The specified number of entity blocks follows the first line of the section. Each block has a header line consisting of four integers, the entity dimension, the entity tag, a flag with allowed values of 0 or 1 (parametric), and the number of nodes in the block. *Object Toolkit* ignores the first three parameters, using only the number of nodes in the block. Within each entity block, the header line is followed by the node tags, one per line. The node tags are in turn followed by the node locations, one per line. Therefore, each entity block consists of a header and two lines per node.

The example above has 8 entity blocks, each with one node. All 8 nodes could also all have been placed within a single entity block, with the eight lines with a single integer apiece followed by the eight lines with three real numbers each.

Object Toolkit requires that the imported node tags be non-negative and unique. (*Gmsh* requires positive node tags.) After reading in the object, *Object Toolkit* equivalences and renumbers the nodes to be consecutive starting with zero.

$Elements Section

The first line of the $Elements section must contain four integers, the number of entity blocks, the number of elements, the minimum element tag, and the maximum element tag. *Object Toolkit* uses only the number of entity blocks.

The specified number of entity blocks follows the first line of the section. Each block has a header line consisting of four integers, the entity dimension, the entity tag, the element type, and the number of elements in the block. *Object Toolkit* ignores the first two parameters. *Object Toolkit* uses only element types "2" for a 3-node triangle and "3" for a 4-node quadrangle. Elements of other types are ignored. Within each entity block, the header line is followed by lines specifying the elements. Each element is specified by an integer element tag followed by the tags of the nodes forming the element in counterclockwise order as viewed from the object exterior.

The example above has 6 entity blocks, each with one element. All 6 elements could also have been placed within a single entity block.

A.4 *Nastran* BDF or DAT File

The *Nastran* BDF (Bulk Data Format) and DAT (DATa) text files have the same format. The files consist of a list of entries (nodes, elements, and other items) with each entry on one or two lines. The name of the entry starts in the first column. If the name is immediately followed by an asterisk, the following line is a continuation. If no asterisk is present, all of the information regarding the item is on a single line. Three formats are available and they can be intermixed. The formats are known as "small field format", "large field format", and "free field format". A node must be defined before it can be used in an element definition.

Object Toolkit's requirements are somewhat looser than the published *Nastran* standards. *Object Toolkit* does *not* handle what the *Nastran* documentation refers to as replication. Only lines that start with "GRID", "CQUAD", or "CTRI" or, for large field format, follow such a line, are read. Lines using free field format are identified by the presence of at least one comma. Lines using large field format are identified by the presence of an asterisk within the first eight characters of the line.

Figure 147 shows the BDF file for an eight-node, six-element cube. In this example, the nodes are defined using the large field format, and each requires two lines. The first three elements are defined using the free field format, the last three elements are defined using the small field format.

```
$ Example mesh of a cube
GRID*   1               0               -0.5            -0.7
*       1
GRID*   2               0               -0.5            -0.7
*       0
GRID*   3               0               -0.5            0.3
*       1
GRID*   4               0               -0.5            0.3
*       0
GRID*   5               0               0.5             -0.7
*       1
GRID*   6               0               0.5             -0.7
*       0
GRID*   7               0               0.5             0.3
*       1
GRID*   8               0               0.5             0.3
*       0
CQUAD4,9,1,4,2,1,3
CQUAD4,10,2,5,6,8,7
CQUAD4,11,3,1,2,6,5
CQUAD4  12      4       8       4       3       7
CQUAD4  13      5       2       4       8       6
CQUAD4  14      6       7       3       1       5
ENDDATA
```

Figure 147. *Nastran* BDF (or DAT) File Defining an Eight-node, Six-element Cube.

Nodes

The characters "GRID" at the beginning of the entry specifies that a node is being defined. In small field format, the node number is given in columns 9 through 16. Columns 17 through 24 contain a number ignored by *Object Toolkit*. The three numbers in columns 25 through 32, columns 33 through 40, and columns 41 through 48 are the x, y, and z values of the location in

Approved for public release; distribution is unlimited.

meters. The numbers can be anywhere within the columns specified. Anything after column 48 is ignored by *Object Toolkit*.

In large field format, the node number is given in columns 9 through 24. Columns 25 through 39 contain a number ignored by *Object Toolkit*. The two numbers in columns 40 through 55 and columns 56 through 71 are the x and y values of the location in meters. The z value is in columns 9 through 24 of the following line. The numbers can be anywhere within the columns specified. All other columns are ignored by *Object Toolkit*.

In free field format, the items are separated by commas and the entire definition is on a single line. (The format does allow for free field format on multiple lines, but *Object Toolkit* does not recognize it.) The second item is the node number. The third item is ignored by *Object Toolkit*. The fourth, fifth, and sixth items give the location in meters. Consecutive commas indicate a missing field. A dollar sign ($) indicates a comment, so all characters after a dollar sign are ignored.

Object Toolkit requires that the imported node numbers be non-negative and unique. After reading in the object, *Object Toolkit* equivalences and renumbers the nodes to be consecutive starting with zero.

Elements

A surface element is defined by the node numbers of its corner nodes in counterclockwise order. The only elements that *Object Toolkit* recognizes are quadrilaterals (CQUAD, CQUAD4, CQUAD8, CQUADR, CQUADX) and triangles (CTRIA3, CTRIA6, CTRIAR, CTRIAX, CTRIAX6). *Object Toolkit* only accommodates linear elements. For the higher order elements (CQUAD8 and CTRIAX6) only the corner nodes are read. In all formats, the element number (first item after the keyword) is ignored.

In small field format, the node numbers for the corner nodes are given in columns 25 through 32, columns 33 through 40, columns 41 through 48, and columns 49 through 56 (quadrilaterals only). The numbers can be anywhere within the columns specified and are in counterclockwise order as viewed from the object exterior. Anything after column 48 is ignored by *Object Toolkit*.

In large field format, the node numbers for the first two corner nodes are given in columns 41 through 56 and columns 57 through 72. For quadrilaterals, the node numbers of the other two nodes are in columns 9 through 24 and 25 through 40 of the following line. For triangles, the number of the third node is in columns 9 through 24. The numbers can be anywhere within the columns specified. All other columns are ignored by *Object Toolkit*. The nodes are in counterclockwise order as viewed from the object exterior.

In free field format, the items are separated by commas and the entire definition is on a single line. (The format does allow for free field format on multiple lines, but *Object Toolkit* does not recognize it.) The second and third items are ignored by *Object Toolkit*. The fourth, fifth, and sixth, and seventh (for quadrilaterals) items are the node numbers of the corner nodes in counterclockwise order as viewed from the object exterior. Consecutive commas indicate a missing field. A dollar sign ($) indicates a comment, so all characters after a dollar sign are ignored.

A.5 *I-DEAS* Universal File

An *I-DEAS* Universal file consists of datasets. Only datasets 2411 and 2412 are read by *Object Toolkit*. Nodes are read from dataset 2411 and elements from dataset 2412. All other datasets are ignored. Figure 148 shows the Universal file for an eight-node, six-element cube. A new dataset is indicated by the presence of "-1" as the first entry of a line. The following line has one integer, the dataset number. The end of a dataset is also indicated by a "-1" as the first entry of the line.

```
    -1
  2411
         1          1          1         11
-5.0000000000000000D-001 -6.9999999999999996D-001  1.0000000000000000D+000
         2          1          1         11
-5.0000000000000000D-001 -6.9999999999999996D-001  0.0000000000000000D+000
         3          1          1         11
-5.0000000000000000D-001  3.0000000000000004D-001  1.0000000000000000D+000
         4          1          1         11
-5.0000000000000000D-001  3.0000000000000004D-001  0.0000000000000000D+000
         5          1          1         11
 5.0000000000000000D-001 -6.9999999999999996D-001  1.0000000000000000D+000
         6          1          1         11
 5.0000000000000000D-001 -6.9999999999999996D-001  0.0000000000000000D+000
         7          1          1         11
 5.0000000000000000D-001  3.0000000000000004D-001  1.0000000000000000D+000
         8          1          1         11
 5.0000000000000000D-001  3.0000000000000004D-001  0.0000000000000000D+000
    -1
    -1
  2412
        23         94          3          0          7          4
         1          2          6          5
        24         94          4          0          7          4
         8          4          3          7
        25         94          5          0          7          4
         2          4          8          6
        26         94          6          0          7          4
         7          3          1          5
    -1
    -1
  2477
    -1
```

Figure 148. *I-DEAS* Universal File Defining an Eight-node, Six-element Cube.

Object Toolkit requirements for the datasets read is as follows.

Dataset 2411

Pairs of lines follow the "2411" indicating the node dataset. The first entry in the first line of the pair is an integer specifying the node number. The second line has three real numbers specifying the node location in meters. The numbers can be given in decimal, exponential, or exponential with a lower or upper case "d" format.

Object Toolkit requires that the imported node numbers be non-negative and unique. After reading in the object, *Object Toolkit* equivalences and renumbers the nodes to be consecutive starting with zero.

Approved for public release; distribution is unlimited.

Dataset 2412

The elements are in dataset 2412. Each element is described in two or three lines. The first line consists of six integers. The first integer is the element number, which is ignored. The second is the element type. The following three integers are ignored. The six integer is the number of nodes that make up the element. *Object Toolkit* reads only element types “44” and “94” for a 4-node quadrangle and types “41” and “91” for a 3-node triangle. Each of these is defined by a pair of lines. The second line consists of four or three integers specifying the number of the nodes in counterclockwise order as viewed from the object exterior.

The Universal format allows for two node elements, which are defined using three lines each. All other elements are defined on two lines. *Object Toolkit* ignores all of these.

A.6 STEP files

Object Toolkit can read a finite element model from some STEP formatted files. In the future, as more examples are obtained, more capabilities may be added. *Object Toolkit cannot* mesh a geometric description. The specifications used to build the interpreter are for STEP AP209.

A STEP file is an ASCII file that represents 3D objects in Computer-aided design (CAD) and related information using a format that conforms to ISO 10303-21. Some STEP Application Protocols can include data associated with a finite element analysis (FEA) model and analysis results.

Each entry in a STEP file starts with a hash mark (#) followed by a line number. The line number is followed by an equal sign (“=”) and then a string describing the type of entry. The string is followed by arguments within parentheses and separated by commas. Each entry ends with a semicolon (;). A single entry can extend over multiple lines. Entries may be in any order.

The second item in each entry is a string specifying the type of entry. *Object Toolkit* recognizes “CARTESIAN_POINT", “NODE”, and "SURFACE_3D_ELEMENT_REPRESENTATION". Figure 149 shows a selection of lines from a STEP formatted files that define an eight-node, six-element cube.

```
#1000=CARTESIAN_POINT('1',(1,0,-0.50000,-0.70000,1.000000));
#1002=CARTESIAN_POINT('2',(0,-0.50000,-0.70000,0.00E+00));
#1003=CARTESIAN_POINT('3',(-0.50000,0.300000,1.000000));
#1004=CARTESIAN_POINT('4',(-0.50000,0.300000,0.00E+00));
#1005=CARTESIAN_POINT('5',(0.500000,-0.70000,1.000000));
#1006=CARTESIAN_POINT('6',(0,0.500000,-0.70000,0.00E+00));
#1007=CARTESIAN_POINT('7',(0,0.500000,0.300000,1.000000));
#1008=CARTESIAN_POINT('8',(0,0.500000,0.300000,0.00E+00));
#2001=NODE('1',(#1001),#4000,#2000);
#2002=NODE('2',(#1002),#4000,#2000);
#2003=NODE('3',(#1003),#4000,#2000);
#2004=NODE('4',(#1004),#4000,#2000);
#2005=NODE('5',(#1005),#4000,#2000);
#2006=NODE('6',(#1006),#4000,#2000);
#2007=NODE('7',(#1007),#4000,#2000);
#2008=NODE('8',(#1008),#4000,#2000);
#3001=SURFACE_3D_ELEMENT_REPRESENTATION('1',(#2200),#4100,
(#2004,#2002,#2001,#2003),#2000,#2102,#2103,#2106);
#3002=SURFACE_3D_ELEMENT_REPRESENTATION('2',(#2200),#4100,
(#2005,#2006,#2008,#2007),#2000,#2102,#2103,#2106);
#3003=SURFACE_3D_ELEMENT_REPRESENTATION('3',(#2200),#4100,
(#2001,#2002,#2006,#2005),#2000,#2102,#2103,#2106);
#3004=SURFACE_3D_ELEMENT_REPRESENTATION('4',(#2200),#4100,
(#2008,#2004,#2003,#2007),#2000,#2102,#2103,#2106);
#3005=SURFACE_3D_ELEMENT_REPRESENTATION('5',(#2200),#4100,
(#2002,#2004,#2008,#2006),#2000,#2102,#2103,#2106);
#3006=SURFACE_3D_ELEMENT_REPRESENTATION('6',(#2200),#4100,
(#2007,#2003,#2001,#2005),#2000,#2102,#2103,#2106);
```

Figure 149. Selected Lines from a STEP File Defining an Eight-node, Six-element Cube.

Object Toolkit's requirements for the format of the entities read are as follows.

CARTESIAN_POINT

A Cartesian point represents a point in three-dimensional space. The string CARTESIAN_POINT is followed by two arguments in parentheses and separated by commas. The first argument is ignored. The second argument is the position, expressed as a triplet of real numbers separated by commas and within parentheses. *Object Toolkit* assumes that the position is given in meters. *Object Toolkit* ignores the parentheses and allows the numbers to be separated by either spaces or commas.

NODE

A node connects a node number with a location specified by a Cartesian point. The line number (the number immediately following the hash sign) is the node number. The string NODE is followed by multiple arguments separated by commas and within parentheses. The second argument is the line number of the Cartesian point specifying the location of the node. The line number is prefaced by a hash sign and the hash sign and line number are contained with parentheses.

SURFACE_3D_ELEMENT_REPRESENTATION

The entity type SURFACE_3D_ELEMENT_REPRESENTATION defines a surface element. The string is followed by several arguments in parentheses and separated by commas. The first three arguments are ignored. The fourth argument is a list of node numbers (line numbers for

NODE lines) separated by commas and within parentheses. The first four nodes are taken to be the corners of a quadrilateral surface element in counterclockwise order as viewed from the object exterior. If only three nodes are present, the element is a triangle. If more than four nodes are present, only the first four are read.

A.7 SEE Interactive Spacecraft Charging Handbook Material Definition File

Material definitions in the XML format shown in Figure 150 can be read into *Object Toolkit* by choosing "Import SEE Handbook Materials File" on the **File** menu. The name is unrestricted, although some software has constraints. (*Nascap-2k* treats material names as four-character significant and case insensitive.) Table 12 shows the correspondence between the labels used in the file and the property names.

```
<ROOTSTUB><Materials>Material Nodes<Mkapton name="Kapton" type="Insulator"
p0="3.5" p1="0.000127" p2="1e-16" p3="5" p4="2.1" p5="0.15" p6="71.48"
p7="0.6" p8="312.1" p9="1.77" p10="0.455" p11="140" p12="0.00002"
p13="10000000000000000" p14="12.01" p15="1600" p16="17" p17="18" p18="1e-18"
p19="20">Kapton</Mkapton></Materials></ROOTSTUB>
```

Figure 150. *SEE Handbook* Material Definition File for Material Kapton with Default Properties.

Table 12. Correspondence between Material Property Numbers and Names.

p0	Dielectric Constant
p1	Thickness
p2	Bulk Conductivity
p3	Effective Atomic Number
p4	Delta-Max (Secondary yield)
p5	E-Max (Secondary yield)
p6	Range 1 (Electron range)
p7	Exponent 1 (Electron range)
p8	Range 2 (Electron range)
p9	Exponent 2 (Electron range)
p10	Proton Yield (Ion induced secondary emission)
p11	Proton Max (Ion induced secondary emission)
p12	Photoemission
p13	Surface Resistivity
p14	Atomic Weight
p15	Density
p16	Not used 1
p17	Not used 2
p18	Radiation-Induced Conductivity
p19	Transparency to Charged Particles

Appendix B: Output file format

Object Toolkit uses the XML file format to store the spacecraft description.

B.1 XML

Microsoft Corporation defines XML this way: “Extensible Markup Language (XML) is the universal format for data on the Web . . . XML allows developers to easily describe and deliver rich, structured data from any application in a standard, consistent way.”

Using XML, it is easy to structure data in a way that is both logical and flexible and that is easily interpreted by both human and artificial intelligence. Java and C# contain integrated, standard, W3C-compliant software to read, interpret, construct, modify, and write XML structured data. XML parsers are also available for other languages and platforms.

Microsoft Internet Explorer is a convenient tool for displaying XML data files, allowing parts of the data tree to be expanded and collapsed. While XML files can be edited using an ordinary text editor, it is usually more convenient to use commercial software designed for that purpose. XMLShell (http://www.softgauge.com/xmlshell/index.htm) is an excellent aid for working with XML files.

B.2 *Object Toolkit* File Format

The *Object Toolkit* output file provides the “Nodes” (points) and “Elements” (elemental surfaces) that define the geometry of the object, attributes of those surfaces, and the properties associated with the attributes. Figure 151 shows a nearly collapsed version of an *Object Toolkit* output file.

```
<?xml version="1.0" encoding="UTF-8"?>
- <Assembly Name="Project">
  + <Assembly Name="Consolidated">
    <Transformation> 1.0 0.0 0.0 0.0 1.0 0.0 0.0 0.0 1.0 0.0 0.0 0.0 </Transformation>
  - <Mesh NNodes="284" NElt="296">
      + <Nodes>
      + <Elements>
        <Commands/>
    </Mesh>
    <Tolerance Value="0.001"/>
  + <MaterialProperties Name="Kapton" Color="32896" App="Nascap2K">
  + <MaterialProperties Name="Teflon" Color="255" App="Nascap2K">
  + <MaterialProperties Name="Aluminum" Color="16776960" App="Nascap2K">
  + <MaterialProperties Name="Gold" Color="65535" App="Nascap2K">
  + <MaterialProperties Name="OSR" Color="65280" App="Nascap2K">
  + <MaterialProperties Name="Black Kapton" Color="32768" App="Nascap2K">
  + <MaterialProperties Name="Solar Cells" Color="16711680" App="Nascap2K">
  + <MaterialProperties Name="Graphite" Color="8421504" App="Nascap2K">
  + <AttributeProperties Value="SA" Attribute="Subsystem">
  + <SpecialObjectProperties ObjectType="Thruster" ObjectName="SPT100">
</Assembly>
```

Figure 151. Example of an *Object Toolkit* XML Output File.

The enclosing tag (“Document Element”) of the file is an “Assembly” with the name of “Project” whose mesh represents the full model that has been defined. Constituent components of this assembly are included as child elements of “Project.” The XML element for a constituent component (excepting Primitive components) does not contain the component’s full mesh, but rather contains the directions used by *Object Toolkit* to rebuild the mesh, so that *Object Toolkit*

Approved for public release; distribution is unlimited.

can be used to edit the model at the component level (with the caveats given in the text of this manual).

The tags of interest to applications that read the file are “Mesh” (defining the Nodes and Elements of the model), “AttributeProperties” (defining the pointing properties of portions of a spacecraft), “MaterialProperties” (defining material properties), and “SpecialObjectProperties” (used to define position and direction of thrusters and the properties of any other special components).

The “Mesh” tag (whose attributes include the numbers of Nodes and Elements) encloses the Nodes and Elements that define the object geometry, as well as commands that specify additional operations to be performed when assembling the component meshes. Each Node (Figure 152) has attributes of “index” (by which it can be referenced) and “x,” “y,” and “z” (specifying its absolute position in space in meters). Figure 153 shows a series of Element tags. Each Element has attributes that include the indices of the three or four nodes that define its geometry, and the name of its surface material. The “Subsystem” parameter (“SolarArray1” for the elements shown) is a special attribute included as a “Param” child element.

```
- <Nodes>
    <Node index="0" x="0.40815289726477266" y="-0.4933721350833382"
      z="0.37499676406649896" />
    <Node index="1" x="0.40487264290363767" y="-0.29339903706431464"
      z="0.3749967629466572" />
    <Node index="2" x="0.24489416451861146" y="-0.29602324055271007"
      z="0.3749998715249687" />
    <Node index="3" x="0.24817441887974662" y="-0.4959963385717336"
      z="0.3749998726448105" />
    <Node index="4" x="0.08491568613358555" y="-0.2986474440411052"
      z="0.37500298010328037" />
```

Figure 152. Series of Node Tags.

```
- <Element index="163" Material="Graphite" InitialPotential="0.0" Conductor="1"
    Node_0="155" Node_1="163" Node_2="164" Node_3="156" EFieldCondition="false">
    <Param ParamName="Subsystem" ParamValue="SA" />
  </Element>
- <Element index="164" Material="Graphite" InitialPotential="0.0" Conductor="1"
    Node_0="156" Node_1="164" Node_2="165" Node_3="157" EFieldCondition="false">
    <Param ParamName="Subsystem" ParamValue="SA" />
  </Element>
- <Element index="165" Material="Solar Cells" InitialPotential="0.0" Conductor="2"
    Node_0="166" Node_1="167" Node_2="168" Node_3="169" EFieldCondition="false">
    <Param ParamName="Subsystem" ParamValue="SA" />
  </Element>
- <Element index="166" Material="Solar Cells" InitialPotential="0.0" Conductor="2"
    Node_0="169" Node_1="168" Node_2="170" Node_3="171" EFieldCondition="false">
    <Param ParamName="Subsystem" ParamValue="SA" />
  </Element>
- <Element index="167" Material="Solar Cells" InitialPotential="0.0" Conductor="2"
    Node_0="171" Node_1="170" Node_2="172" Node_3="173" EFieldCondition="false">
    <Param ParamName="Subsystem" ParamValue="SA" />
```

Figure 153. Series of Element Tags.

Figure 154 shows the properties to be associated with a particular value of the subsystem attribute. The properties describe the “Target” (determined as the first “Item” of the enumeration)

Approved for public release; distribution is unlimited.

toward which the subsystem is to point, and the rotation operations available to achieve that pointing.

```
- <AttributeProperties Attribute="Subsystem" Value="SA" xmlns="">
  - <Property name="Target" value="ENUM">
      <Item value="Sun" />
      <Item value="(None)" />
    </Property>
    <Property name="FixedX" value="0.0" />
    <Property name="FixedY" value="0.0" />
    <Property name="FixedZ" value="-0.6375" />
    <Property name="PointX" value="0" />
    <Property name="PointY" value="0" />
    <Property name="PointZ" value="1" />
    <Property name="Axis1X" value="0" />
    <Property name="Axis1Y" value="1" />
    <Property name="Axis1Z" value="0" />
    <Property name="Axis2X" value="0" />
    <Property name="Axis2Y" value="1" />
    <Property name="Axis2Z" value="0" />
  </AttributeProperties>
```

Figure 154. Attribute Properties for a Specified Value of “Subsystem.”

Figure 155 shows the properties to be associated with a particular instance of a “SpecialObject” of type “Thruster.” The properties include the location and pointing direction, as well as the display color.

```
- <SpecialObjectProperties ObjectType="Thruster" ObjectName="SPT100" xmlns="">
    <Property name="Name" value="SPT100" />
    <Property name="Type" value="Hall Thruster" />
    <Property name="x" value="0.0" />
    <Property name="y" value="0.4" />
    <Property name="z" value="-0.2875" />
    <Property name="r" value="255" />
    <Property name="g" value="192" />
    <Property name="b" value="0" />
    <Property name="dirX" value="0" />
    <Property name="dirY" value="1" />
    <Property name="dirZ" value="0" />
  </SpecialObjectProperties>
```

Figure 155. Special Component Properties for an Instance of “Thruster.”

Each material is defined by a separate “MaterialProperties” element. Each “MaterialProperties” element has three attributes. The “Name” attribute specifies the name for the material. If the “App” attribute is not present, *Object Toolkit* assumes the material is for the current application. The material color is used when displaying the object in *Object Toolkit.* If the “Color” attribute is not present *Object Toolkit* assigns a random color to the material. The allowed value are strings specifying the defined colors (Blue, Green, Red, Yellow, Magenta, Cyan, Grey, Dark Green, Dark Red, Dark Blue, Dark Yellow, Dark Magenta, Dark Cyan, Black) and numbers specifying RGB color definitions.

The properties of the material are specified as child elements of the “MaterialProperties” element. A property element has three attributes “Name,” “Index,” and “Value.” The “Name” attribute is the value displayed in the **Edit** dialog box for the material. The “Index” attribute contains an integer index for the property. A specific application may use either the name or the

index to identify a property. *Nascap-2k* uses the "Index" to identify each property. The "Value" attribute, specifies the value for the property and must be a number.

```
+ <MaterialProperties Name="Kapton" Color="32896" App="Nascap2K">
+ <MaterialProperties Name="Teflon" Color="255" App="Nascap2K">
+ <MaterialProperties Name="Aluminum" Color="16776960" App="Nascap2K">
+ <MaterialProperties Name="Gold" Color="65535" App="Nascap2K">
+ <MaterialProperties Name="OSR" Color="65280" App="Nascap2K">
+ <MaterialProperties Name="Black Kapton" Color="32768" App="Nascap2K">
- <MaterialProperties Name="Solar Cells" Color="16711680" App="Nascap2K">
    <Property Name="Dielectric Constant" Value="3.8" Index="0"/>
    <Property Name="Thickness(m)" Value="1.25E-4" Index="1"/>
    <Property Name="Bulk Conductivity(ohms<sup>-1</sup>m<sup>-1</sup>)" Value="1.0E-13" Index="2"/>
    <Property Name="Atomic Number" Value="10.0" Index="3"/>
    <Property Name="Delta-Max" Value="5.8" Index="4"/>
    <Property Name="E-Max(keV)" Value="1.0" Index="5"/>
    <Property Name="Range 1(&#197)" Value="77.5" Index="6"/>
    <Property Name="Exponent 1" Value="0.45" Index="7"/>
    <Property Name="Range 2(&#197)" Value="156.1" Index="8"/>
    <Property Name="Exponent 2" Value="1.73" Index="9"/>
    <Property Name="Proton Yield" Value="0.244" Index="10"/>
    <Property Name="Proton Max(keV)" Value="230.0" Index="11"/>
    <Property Name="Photoemission(A m<sup>-2</sup>)" Value="2.0E-5" Index="12"/>
    <Property Name="Surface Resistivity(ohms/square)" Value="1.0E19" Index="13"/>
    <Property Name="Atomic Weight(amu)" Value="20.0" Index="14"/>
    <Property Name="Density(kg m<sup>-3</sup>)" Value="2660.0" Index="15"/>
    <Property Name="Not Used 1" Value="17.0" Index="16"/>
    <Property Name="Not Used 2" Value="18.0" Index="17"/>
    <Property Name="Rad. Cond." Value="1.0E-18" Index="18"/>
    <Property Name="Transparency" Value="0.0" Index="19"/>
  </MaterialProperties>
+ <MaterialProperties Name="Graphite" Color="8421504" App="Nascap2K">
```

Figure 156. Material Properties Specification in Object Definition File.

When an orbit is specified, *Object Toolkit*'s XML file includes the orbit parameters along with the position and velocity at the evaluation time. The evaluation time is Julian time. For heliocentric orbits, the position is in astronomical units, the velocity is in astronomical units per day, and the Solar Ecliptic Coordinate System (SE) is used. For geocentric orbits the position is in earth radii, the velocity is in earth radii per day, and the Geocentric Equatorial Inertial System (GEI) is used. It also contains the rotation to go from body to ram space coordinates.

Appendix C: Customization

Object Toolkit, originally developed to create spacecraft surface models for the *Nascap-2k* plasma interactions code, can be tailored to define spacecraft models for analysis by other applications. Presently there are two such applications, *EPIC* and *MEM*. There is only one version of *Object Toolkit*, rather than a specialized version for each application. All the application-specific information is contained in *Object Toolkit*'s normal XML output file, rather than in any special XML or other specially formatted files.

C.1 General

The application-specific parameters are specified in an XML file named *ApplicationName*_OTKSpecs.xml. The application name is specified as the first argument to *Object Toolkit* or, alternatively, as an argument preceded by "–app." The root element of the XML is tagged "ApplicationParameters" and has a "Name" attribute giving the name of the application. Presently there are four child elements. "Options" contains commands to provide the user with access to various optional capabilities. "ElementParams" contains information about

the application-specific surface attributes. “SpecialObjects” contains information about application-specific features whose location is to be specified in *Object Toolkit*. “MaterialProperties” contains a list of default materials and their properties along with default properties for new materials.

```
<?xml version="1.0" ?>
- <ApplicationParameters Name="EPIC">
  - <Options>
      <Option>UseMoveWizards</Option>
    </Options>
  - <ElementParams>
    - <Param name="Subsystem" value="defaultSubsystem">
      - <Property name="Target" value="ENUM">
          <Item value="(None)" />
          <Item value="Sun" />
        </Property>
        <Property name="FixedX" value="0" />
        <Property name="FixedY" value="0" />
        <Property name="FixedZ" value="0" />
        <Property name="PointX" value="0" />
        <Property name="PointY" value="0" />
        <Property name="PointZ" value="1" />
        <Property name="Axis1X" value="1" />
        <Property name="Axis1Y" value="0" />
        <Property name="Axis1Z" value="0" />
        <Property name="Axis2X" value="0" />
        <Property name="Axis2Y" value="1" />
        <Property name="Axis2Z" value="0" />
      </Param>
    </ElementParams>
  - <SpecialObjects>
    - <Object Type="Thruster">
        <Property name="Name" value="Thruster" />
        <Property name="Type" value="HET" />
        <Property name="x" value="0" />
        <Property name="y" value="0" />
        <Property name="z" value="0" />
        <Property name="r" value="255" />
        <Property name="g" value="192" />
        <Property name="b" value="0" />
        <Property name="Display Size" value="0.15" />
        <Property name="dirX" value="0" />
        <Property name="dirY" value="0" />
        <Property name="dirZ" value="1" />
      </Object>
    </SpecialObjects>
</ApplicationParameters>
```

Figure 157. *EPIC* Application-specific Parameter File, EPIC_OTKSpecs.xml.

```
<?xml version="1.0"?>
- <ApplicationParameters Name="N2k">
  - <ElementParams>
      <Param name="Injection Point"> </Param>
    + <Param name="Emitter">
    + <Param name="Detector">
    </ElementParams>
  - <SpecialObjects>
    + <Object Type="Thruster">
    + <Object Type="Neutralizer">
    + <Object Type="MagDipole">
    </SpecialObjects>
  - <MaterialProperties Name="Kapton" Color="Dark Yellow" App="Nascap2K">
      <Property Name="Dielectric Constant" Value="3.5" Index="0"/>
      <Property Name="Thickness(m)" Value="1.27E-4" Index="1"/>
      <Property Name="Bulk Conductivity(ohms<sup>-1</sup>m<sup>-1</sup>)" Value="1.0E-16" Index="2"/>
      <Property Name="Atomic Number" Value="5.0" Index="3"/>
      <Property Name="Delta-Max" Value="2.1" Index="4"/>
      <Property Name="E-Max(keV)" Value="0.15" Index="5"/>
      <Property Name="Range 1(&#197)" Value="71.48" Index="6"/>
      <Property Name="Exponent 1" Value="0.6" Index="7"/>
      <Property Name="Range 2(&#197)" Value="312.1" Index="8"/>
      <Property Name="Exponent 2" Value="1.77" Index="9"/>
      <Property Name="Proton Yield" Value="0.455" Index="10"/>
      <Property Name="Proton Max(keV)" Value="140.0" Index="11"/>
      <Property Name="Photoemission(A m<sup>-2</sup>)" Value="2.0E-5" Index="12"/>
      <Property Name="Surface Resistivity(ohms/square)" Value="1.0E16" Index="13"/>
      <Property Name="Atomic Weight(amu)" Value="12.01" Index="14"/>
      <Property Name="Density(kg m<sup>-3</sup>)" Value="1600.0" Index="15"/>
      <Property Name="Not Used 1" Value="17.0" Index="16"/>
      <Property Name="Not Used 2" Value="18.0" Index="17"/>
      <Property Name="Rad. Cond." Value="1.0E-18" Index="18"/>
      <Property Name="Transparency" Value="20.0" Index="19"/>
    </MaterialProperties>
  + <MaterialProperties Name="Teflon" Color="Red" App="Nascap2K">
  + <MaterialProperties Name="Aluminum" Color="Cyan" App="Nascap2K">
  + <MaterialProperties Name="Gold" Color="Yellow" App="Nascap2K">
  + <MaterialProperties Name="OSR" Color="Green" App="Nascap2K">
  + <MaterialProperties Name="Black Kapton" Color="Dark Green" App="Nascap2K">
  + <MaterialProperties Name="Solar Cells" Color="Blue" App="Nascap2K">
  + <MaterialProperties Name="Graphite" Color="Gray" App="Nascap2K">
</ApplicationParameters>
```

Figure 158. *Nascap-2k* Application-specific Parameter File, Nascap2k_OTKSpecs.xml.

C.2 Optional Capabilities

Object Toolkit has two optional capabilities, “UseMoveWizards” and “Orbit.” If the option “UseMoveWizards” is specified, the “Move” wizards appear on the **Wizard** menu. These wizards are discussed in Section 15.7. If the option “Orbit” is specified, the following occurs: the **Orbit** menu, described in Section 15.11, appears on the menu bar and the “Move” wizards appear on the **Wizard** menu.

C.3 Application-specific Surface Attributes

A surface attribute is a parameter associated with every object surface. The surface attribute has a name (e.g., “Material”), a value (e.g., “Aluminum”) with a default (e.g., “Kapton”), and possibly a set of properties that has both default values and specific values for each value of the attribute. An example of an application-specific surface attribute is “Subsystem” used by *Object Toolkit* tailored for *EPIC* and *MEM*. To the extent possible, effort was made to treat application-specific attributes like the native attributes of “Material” and “Conductor.” A target application may define multiple application-specific attributes. Because application-specific attributes do not appear on the **Component Edit** dialog boxes, the user is permitted to assign an attribute value to

all surfaces of a selected component (**Component** menu) as well as to a set of selected surface elements (**Mesh** menu).

Each application-specific attribute has a selection of properties, each of which has a value. Figure 159 shows the property editor for an application-specific attribute, which is brought up from the main **Edit** menu. The caption bar states the attribute name and the value to which the properties belong. The user can change the value of specific properties using the text boxes and pull-down menu. The property names and their default values are specified in the application-specific XML file. If the default value for a property is specified as "ENUM," a combo box listing the available choices is shown rather than a text box.

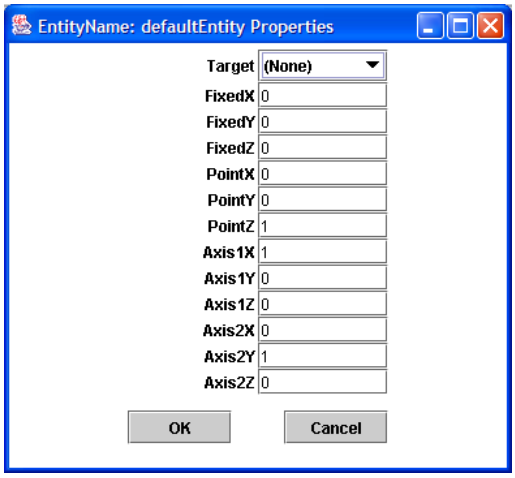


Figure 159. Application-specific Attribute Property Edit Dialog Box.

In the output file, the attribute name and value are included with the attributes for each surface element, and the attribute property value sets appear in their own section, on a par with material names and properties.

C.4 Special Components

Special components (for *EPIC* and *Nascap-2k*, "Thrusters") are components understood by the target application that are included in *Object Toolkit* primarily to specify and display their location. They are not a part of the spacecraft surface model. (If they were, there would be no need to treat them specially.) An application may define multiple special components. Special components appear on the **Add Component** menu below the standard component types.

Each instance of a special component has a type name (e.g., "Thruster"), an instance name (e.g., "Thruster1"), and a set of properties (Figure 160) that includes, at a minimum, its position ("x," "y," "z") and the color with which it is to be displayed ("r," "g," "b"). The properties can be edited for each special component instance in similar fashion to the attribute property values, and the special component instances with their properties are listed in the output file.

Approved for public release; distribution is unlimited.

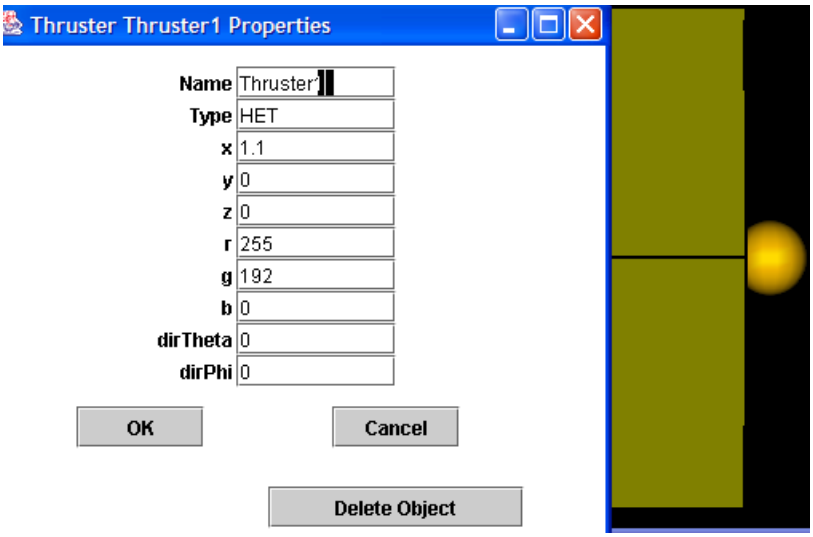


Figure 160. Special Component Edit Dialog Box for "Thruster1." The Object Appears As a Sphere at the Right of the Figure.

C.5 Default Material Definitions

Figure 158 shows an example of a default material properties specification. Each default material is defined by a separate "MaterialProperties" element. Each "MaterialProperties" element has three attributes. The "Name" attribute specifies the name for the material. The "App" attribute is optional. If it is not present, *Object Toolkit* assumes the material is for the current application. The material color is used when displaying the object in *Object Toolkit.* If the "Color" attribute is not present *Object Toolkit* assigns a random color to the material. The allowed values are strings representing the defined colors (Blue, Green, Red, Yellow, Magenta, Cyan, Grey, Dark Green, Dark Red, Dark Blue, Dark Yellow, Dark Magenta, Dark Cyan, Black) and numbers specifying RGB color definitions.

The properties of the material are specified as child elements of the "MaterialProperties" element. Applications can have as many properties as desired. A "Property" element has three attributes "Name," "Index," and "Value." The "Name" attribute is displayed in the **Edit** dialog box for the material. It is processed as html, so superscripts, subscripts, and special characters can be used. The "Index" attribute contains an integer index for the property. An application may use either the name or the index to identify a property. In applications that use the name to identify the property, the index need not be present. *Nascap-2k* uses the "Index" to identify each property, so the "Index" attribute is required. The "Value" attribute, which specifies the value for the property, must be present and must be a number.

The first material specified in the *ApplicationName*_OTKSpecs.xml file is the default material. New materials are created with the same properties as this material.

Object Toolkit does not require that each material have the same number of properties or that the names of the properties be the same for different materials.

Regardless of the content of the Nascap2k_OTKSpecs.xml file, *Object Toolkit* tailored for *Nascap-2k* always has 20 material properties. The property names are those specified for the first material given in the Nascap2k_OTKSpecs.xml file.

Approved for public release; distribution is unlimited.

GLOSSARY

Assembly: A collection of two or more components forming a new "assembly" component. An assembly is a component composed of other components.

Astronomical unit (AU): 149,597,871 km, the mean distance between the sun and the earth.

Attribute: Property associated with an element or XML tag. For example, *Object Toolkit* surface elements have attributes of material name, conductor number, four pointers to nodes, and (for *EPIC*) subsystem.

Compatible elements: Set of surface elements in which each edge has exactly one element on its right and exactly one on its left. Some codes that rely on *Object Toolkit* for object definition prohibit incompatible elements (e.g., *Nascap-2k*). Other codes allow them (e.g., *EPIC*). Incompatibilities may be resolved by mesh editing. Incompatible elements are displayed with the relevant edges in red.

Component local coordinate system: Coordinate system in which a component is defined. The dialog boxes used to specify the parameters of the standard components and "Edit Node" on the **Mesh** menu use the component coordinate system, not the object coordinate system, to specify node locations.

Component: An *Object Toolkit* object is built of components. A component can be a standard component, an assembly, a Primitive, or a special component.

Conductor number: Identifying number used by *Nascap-2k* to define electrical connectivity. Conductor 1 is almost always chassis ground. All conducting elements with the same conductor number have the same surface potential. Each element must have a conductor number as an attribute.

Consolidate: Combine two or more components into an assembly.

Constituent component: Component that is part of an assembly.

DMSP: Defense Meteorological Satellite Program. Series of earth-observing satellites with 101 minute, sun-synchronous, near-polar orbit at an altitude of 830 km. Some of the DMSP spacecraft have carried particle detectors that are able to measure charging events.

***DynaPAC*:** Dynamic Plasma Analysis Code, developed under Air Force contract (1989—1999). Many of *Nascap-2k*'s computational modules were originally developed for *DynaPAC*.

Element: An elemental surface defined by three (a triangle) or four (a quadrilateral) nodes. The nodes are ordered counterclockwise as viewed from an exterior point.

***EPIC*:** *Electric Propulsion Interactions Code*. Engineering tool to model interactions between electric propulsion effluents and spacecraft systems developed by Leidos (as SAIC) and distributed by the SEE Program at NASA/MSFC. *EPIC* uses *Object Toolkit* to define object geometry and materials. Available from https://software.nasa.gov/software/MFS-32165-1. Presently (2020) only works on outdated operating systems.

Finite element model: Representation of geometry consisting of nodes and elements.

Finite element method: Technique for solving elliptical equations (such as Poisson's equation) in which the computational domain is divided into small elements within each of which the spatial variation of a trial function is defined by interpolation of a small number of nodal values. The solution to the discretized elliptical equation is determined by finding the set of nodal values that minimizes an integral functional of the trial function.

Free edge: A free edge indicates that there is a hole in the object or that the elements are incompatible. *Object Toolkit* indicates free edges by the presence of red lines.

Frustum of a pyramid: Truncated pyramid in which the plane cutting off the apex is parallel to the base.

Geocentric orbit: Orbit described with respect to earth.

***Gmsh*:** Open source three-dimensional finite element mesh generator with a built-in computer aided design engine and post-processor.

Heliocentric orbit: Orbit described with respect to the sun.

***I-DEAS*:** Integrated Design and Engineering Analysis Software, a computer-aided design software package originally produced by SDRC in 1982. It is no longer being developed as a separate produce by its present owner Siemens PLM Software.

Instance name: Name associated with an instance of a component.

Material: An entity representing a surface material that is identified by a unique name and that has a display color and a set of material properties.

Material Name: An attribute of each element pointing to a set of associated properties (e.g., conductivity, secondary yield). Each element has a material name as an attribute.

Mesh: A set of elements together with their defining nodes.

MEM: *Meteoroid Engineering Model*, a computer code developed for the SEE Program at the University of Western Ontario that uses *Object Toolkit* to define the object. Available from https://software.nasa.gov/software/MFS-32205-2.

MESSENGER: MErcury Surface, Space ENvironment, Geochemistry, and Ranging. A NASA scientific spacecraft to Mercury launched August 2004.

***Nascap-2k*:** Interactive tool developed by Leidos for studying plasma interactions with realistic spacecraft in three dimensions. *Nascap-2k* is distributed by the SEE Program at NASA/MSFC. *Nascap-2k* uses *Object Toolkit* to define object geometry and materials. Available from https://software.nasa.gov/software/MFS-32056-1.

***Nastran*:** Finite element analysis (FEA) program that was originally developed for NASA in the late 1960s under United States government funding for the aerospace industry. The source code is publicly available. Commercial versions of Nastran are currently available from several companies, and *Nastran* libraries are integrated in several commercial FEA packages.

Node: An entity representing a point in space.

Object: Entire model defined in *Object Toolkit*.

Paraboloid: Surface of revolution of a parabola.

***PATRAN*:** Popular, general purpose, three-dimensional, finite element modeling software package distributed by MSC Software Corporation.

Primitive: A component defined solely by its mesh (nodes and elements).

Prism: Solid with bases that are polygons equal in size and shape and are parallel to each other. The lateral faces of a prism are all parallelograms or rectangles.

Pyramid: Polyhedron with a polygonal base and lateral faces that taper to an apex.

RGB color definition: An integer that specifies a color as a mixture of red, green, and blue. The RGB integer is given by the formula R + 256(G + 256B), where R, G, and B are integers between 0 and 255 inclusive.

Right prism: Prism that has bases aligned one directly above the other and has lateral faces that are rectangles.

Right pyramid: Pyramid that has its apex aligned directly above the center of the base.

Right regular prism: Right prism with bases that are regular polygons.

Right regular pyramid: Right pyramid with a base that is a regular polygon.

***SEE Interactive Spacecraft Charging Handbook*:** Interactive handbook used to assess material models, environment models, and their interactions developed by Leidos (as SAIC) and distributed by the SEE Program at NASA/MSFC. Uses same material and environment models as *Nascap-2k*. Available from https://software.nasa.gov/software/MFS-31675-1. Note that it only works on outdated operating systems.

SEE Program: NASA Space Environments and Effects Program at Marshall Space Flight Center.

Simple component: Component that is not an assembly.

Special component: A component with a position and an RGB color definition for display in *Object Toolkit*, together with a set of application-specific properties. Special components are application-specific.

Standard component: Simple component predefined within *Object Toolkit*. Each has a set of user-definable properties to specify its shape, gridding, surface material, and conductor number. (Conductor number is only applicable to use with *Nascap-2k*.) There are five standard components available: Box, Dish, Panel, Boom, and Cylinder.

STEP file: Representation of three-dimensional objects in computer-aided design (CAD) conforming to ISO 10303-21. The file extensions .stp and .step indicate that the file contains data

conforming to a STEP Application Protocol (standardized representation of product data in a specific application context). The application protocol AP-209 specifies computer-interpretable composite and metallic structural product definition including their shape, their associated finite element analysis (FEA) model and analysis results, and material properties. This includes the finite element analysis model, analysis controls, and analysis results information.

Subsystem: Application-specific surface attribute. A subsystem is described by a name and set of properties. *Object Toolkit* tailored for *EPIC* and *MEM* uses subsystems to enable specifying rotations.

Surface: Several contiguous elements with approximately the same normal.

Surface normal: Vector normal to a surface element and pointing outward. The nodes are ordered counterclockwise when viewed from the direction the normal points.

VUFF: File used by the *NX I-DEAS* TMG thermal analysis program. TMG can create an ASCII version that includes the object description.

Weld: Component automatically created by the "Element to Element Wizard" that acts as an interface between the two components being combined.

XML: eXtensible Markup Language, a hierarchal text format for data following specifications maintained by the World Wide Web Consortium (W3C).

XMLShell: XML text editor available from http://www.softgauge.com/xmlshell/index.htm.

DISTRIBUTION LIST

DTIC/OCP
8725 John J. Kingman Rd, Suite 0944
Ft Belvoir, VA 22060-6218 1 cy

AFRL/RVIL
Kirtland AFB, NM 87117-5776 1 cy

Official Record Copy
AFRL/RVB/Dr. David L. Cooke 1 cy

Approved for public release; distribution is unlimited.