



12-2004

## Automated Grid Generator for MHD Flow Simulations Made With the Gems Code

Konstantin Kolokolnikov  
*University of Tennessee - Knoxville*

Follow this and additional works at: [https://trace.tennessee.edu/utk\\_gradthes](https://trace.tennessee.edu/utk_gradthes)



Part of the [Engineering Commons](#)

---

### Recommended Citation

Kolokolnikov, Konstantin, "Automated Grid Generator for MHD Flow Simulations Made With the Gems Code. " Master's Thesis, University of Tennessee, 2004.  
[https://trace.tennessee.edu/utk\\_gradthes/2270](https://trace.tennessee.edu/utk_gradthes/2270)

This Thesis is brought to you for free and open access by the Graduate School at TRACE: Tennessee Research and Creative Exchange. It has been accepted for inclusion in Masters Theses by an authorized administrator of TRACE: Tennessee Research and Creative Exchange. For more information, please contact [trace@utk.edu](mailto:trace@utk.edu).

To the Graduate Council:

I am submitting herewith a thesis written by Konstantin Kolokolnikov entitled "Automated Grid Generator for MHD Flow Simulations Made With the Gems Code." I have examined the final electronic copy of this thesis for form and content and recommend that it be accepted in partial fulfillment of the requirements for the degree of Master of Science, with a major in Engineering Science.

Roy J. Schulz, Major Professor

We have read this thesis and recommend its acceptance:

Boris A. Kupershmidt, Charles L. Merkle

Accepted for the Council:

Carolyn R. Hodges

Vice Provost and Dean of the Graduate School

(Original signatures are on file with official student records.)

To the Graduate Council:

I am submitting herewith a thesis written by Konstantin Kolokolnikov entitled “AUTOMATED GRID GENERATOR FOR MHD FLOW SIMULATIONS MADE WITH THE GEMS CODE.” I have examined the final electronic copy of this thesis for form and content and recommend that it be accepted in partial fulfillment of the requirements for the degree of Master of Science, with a major in Engineering Science.

Roy J. Schulz  
Major Professor

We have read this thesis  
and recommend its acceptance:

Boris A. Kupershmidt

Charles L. Merkle

Accepted for the Council:

Anne Mayhew  
Vice Chancellor and Dean of  
Graduate Studies

(Original signatures are on file with official student records.)

AUTOMATED GRID GENERATOR FOR MHD FLOW SIMULATIONS  
MADE WITH THE GEMS CODE

A Thesis  
Presented for the  
Master of Science  
Degree  
The University of Tennessee, Knoxville

Konstantin Kolokolnikov

December 2004

## ACKNOWLEDGEMENTS

The author is indebted to his current major professor, [Dr. Roy Schulz](#) for his professional guidance and support.

The author wishes to express his gratitude to the other members of his committee, who all contributed with their special knowledge to the completion of this work.

The author expresses special gratitude to [Dr. Ding Li](#), work with whom was unforgettable, and, who gave so much help to finish the creation of the code represented in this work. The author also expresses his hope that future work with Dr. Ding Li will be possible. Together with Dr. Ding Li, the author would like to thank [Dr. Charles L. Merkle](#) for guidance and everything done during his tenure at [UTSI](#) and at a Purdue University.

The discussion of complicated issues in mathematics with [Dr. Boris Kupersmidt](#) was extremely important for the author during his graduate studies.

The author also appreciates the opportunity to complete this work at [UTSI](#).

The work was done in support of the [HVEPS](#) Project at [UTSI](#) which was supported by [General Atomics Corporation](#) under contract number SCDC 123501. Certain other affiliated companies were involved into this Project.

Author would like to express his gratitude to [Dr. Dennis Keefer](#), [Mr. Robert P. Rhodes](#) and [Dr. Rodger Crawford](#) for the opportunity to communicate with them on general scientific and engineering Problems of MHD Power Generation. Unfortunately, the retirement schedule of Dr. Dennis Keefer did not allow communicating with him for a longer time. Author has his hope that UTGRID® will be very useful for further work with this and many other projects.

## ABSTRACT

Numerical simulations for Hypersonic Vehicle Power System ([HVEPS](#)) Project were based on a multi-domain, general Navier-Stokes Solver Code called the GEMS® Code. The GEMS® Code was in the process of being extended to solve for plasma flows with both self-induced as well as externally-applied magnetic fields. GEMS® is also capable of simulating both laminar and turbulent flow in unbounded as well as ducted flows. For application to the ducted Plasma flows generated experimentally in the [HVEPS](#) program, GEMS® was set up to calculate turbulent flow in 1-3D (dimensional) duct geometries in general and in 2D, 3D ones for purposes of numerical simulation in [HVEPS](#) Project.

To model a real small scale experiment for Conductivity and MHD plasma channel flows it was decided to consider turbulent flow in both 2D and 3D duct geometries and to compare the results to experimental data obtained in [HVEPS](#), as well as with numerical results from other known codes, such as the Mach 2 Code in 2D duct geometry.

Accurate MHD channel flow simulations should require only 3D calculations, since MHD power generation is a completely three-dimensional phenomenon.

In 2002, the latest generation GEMS® code ([GEMS – General Equations and Mesh Solver](#)), for CFD problems was created by Dr. Ding Li and Dr. Charles Merkle. This code can run in 1D, 2D and 3D as options and will be mentioned below.

The problem attacked by the [author of this thesis](#) was to prepare a numerical method to generate appropriate and acceptable computational domains with acceptable grid formats that provide for convergence of numerical simulations made with [GEMS®](#) Code.

In numerical modeling the HVEPS facility at UTSI the following computational domains were required:

- a combustor chamber area;
- a supersonic nozzle;
- an adaptor fitted to the nozzle (ceramic ring as nozzle extension);
- a straight conductivity channel with 6 dielectrics rings;
- a conical MHD plasma channel with 6 dielectric rings;
- an air surrounding nozzle and channel;
- a channel extension as option for grid generation algorithm.

In addition to the variety of computational domains modeled, boundary layers required near the walls must be adequately resolved by the computational grids. Flexibility of grid generation by UTGRID®, which will be discussed later, in section 3.9, boundary layer issues, allows choosing of appropriate grid aspect ratios to provide convergence of the solutions in boundary layer regions. Known grid generators are incapable of working in an automated mode, and the process of grid generation for complicated domains, like the diagonal wall MHD channel, is an extremely time-consuming and difficult task.

The author developed a semi-automated process for creating the meshed domains for the CFD modeling in the HVEPS project. Thus, within the period 2002-2004 a new generation code, UTGRID®, for automatic and flexible grid generation was created by the author, which and links the set of the following programs together:

- [GEMS®](#) (Created by Dr. Ding Li and Dr. Charles Merkle)
- [PGRID®](#) (Created by Dr. Ding Li)
- [UTGRD®](#) (Created by the author of this thesis)
- [GAMBIT™](#) (a commercial product associated with FLUENT®)

- [TECPLOT™](#) (a commercial product)

UTGRID® ties these separate codes into a grid generation process that can provide excellent quality grids in a relatively easy process. It is fully flexible in geometry and allows for the input of all relevant grid parameters which are controllable from its input file.

Using UTGRID®, the author and his colleagues have been able to get converged solutions for 2D turbulent combustor-nozzle flow computations, for 3D plasma conductivity channel flows and for 3D MHD plasma flows in a conical channel.

To date, all runs have been done based on an ideal gas model; future numerical solutions will be done on an appropriate real gas model. Selected results of numerical simulations for combustor-nozzle turbulent flows are attached to this thesis, as well as ones for 3D conductivity channel, and a 3D MHD channel, to illustrate the complex grids that can be generated by the process developed by the author.

In addition, some recently published results with this process for a diagonal-wall MHD generator are presented in the thesis.

The UTGRID® is so flexible that its contemporary module structure allows a user to attach other modules and generate grids for a wide set of computational domains, practically any, using all features of [Fortran 90](#) compilers, for example, the [Portland](#) compiler and the algorithm for semi-automatic grid generation process with preprocessor [GAMBIT™](#).

UTGRID® allows a new qualitative jump in grid generation processes, a new approach to CFD applications in general and to other computational areas.

## Table of Contents

CHAPTER	PAGE
1. INTRODUCTION .....	1
1.1. Background .....	1
1.2. Hardware modeling .....	3
2. THE APPROACH TAKEN TO CREATE A GENERAL, FLEXIBLE AND POWERFUL MESH GENERATION ALGORITHM .....	7
2.1. Overview of the approach .....	7
2.2. GAMBIT™ .....	9
2.3. PGRID® .....	12
2.4. GEMS® .....	14
3. AUTOMATED MESH GENERATOR UTGRID® .....	17
3.1. File system and module structure of UTGRID® .....	19
3.2. Listing of modules and their functions, input file .....	24
3.3. Listing of modification, temporary created files and external subroutines .....	32
3.4. Nozzle input wall or contour profile and generated profiles of its first and second derivatives .....	39
3.5. Manual .....	49
3.5.1. The work directory .....	49
3.5.2. Compilation and installation of UTGRID® .....	53

3.5.3.	Input file .....	56
3.5.4.	The execution of UTGRID® with l_mod, l_chcalc and l_ch set to 1 .....	65
3.5.5.	The execution of UTGRID® with l_mod = 0 and l_ch, l_chcalc set to 1 .....	73
3.5.6.	Modification files creation .....	83
3.9.3.1.	Introduction .....	83
3.5.6.2.	Modification file ‘modify1’ .....	86
3.5.6.3.	Modification file ‘modify2’ .....	89
3.5.6.4.	Modification file ‘modify3_d’ or ‘modify3_s’ .....	90
3.5.6.5.	Recommendations .....	99
3.6.	Output of UTGRID® .....	100
3.7.	The procedure of mesh generation with linked programs: UTGRID®, GAMBIT™, PGRID® for grid generation in formats acceptable by GEMS® .....	103
3.8.	Compatibility with postprocessor TECPLOT™ .....	109
3.9.	The results of mesh generation and boundary layer modeling issues .....	111
3.9.1.	Combustor-nozzle domain .....	112
3.9.2.	Straight conductivity channel domain .....	116
3.9.3.	Diagonal MHD channel domain .....	124
3.9.3.1.	Introduction .....	124

3.9.3.2. Generalization of the DCW Generator Electrode	
Slant Angle .....	125
3.9.4. Mesh quality for equiangular skews .....	132
4. NUMERICAL RESULTS WITH GEMS® AND UTGRID® .....	138
4.1. Combustor-nozzle turbulent flow .....	138
4.2. Conductivity channel .....	141
4.3. Diagonal MHD power generator .....	145
5. SUMMARY AND CONCLUSIONS .....	150
LIST OF REFERENCES .....	152
LIST OF FIGURES .....	156
APPENDIX A1 .....	157
A1.1. FIGURES ILLUSTRATING INPUT GEOMETRY FOR THE CHANNEL	
AND GEOMETRY AND MESH INPUT PARAMETERS FOR THE NOZZLE	158
• Figure 3.5.6 .....	159
Inlet Input Diameters for the Channel.	
• Figure 3.5.7 .....	160
Axial and Angular Parameters in a XY – Plane for the Channel.	
• Figure 3.5.8 .....	161
Primary and Secondary Slope Angles for the Electric Elements respectively to the	
Z – axis in 3D view.	
• Figure 3.5.9 .....	162
Inputted and calculated Data for the Combustor – Nozzle Tract of the Driven Flow.	

<b>A1.2. FIGURES ILLUSTRATING INPUT MESH PARAMETERS FOR THE CHANNEL</b>	163
• <b>Figure 3.5.10</b>	164
Aspect Ratio for Plasma, Solid, Air Zones and the Thickness of the Plasma's Boundary Layer.	
• <b>Figure 3.5.11</b>	165
Axial Grids Distribution with inputted Precision and Axial Aspect ratio, independent on Radial ones.	
• <b>Figure 3.5.12</b>	166
Axial Grid Distribution for the Air Domain Zone.	
<b>A1.3. FIGURES ILLUSTRATING THE CREATION OF THE FILE 'modify1'</b>	167
<b>A1.3.1. FIGURES FOR THE SLOPE ANGLES: 45, 60</b>	167
• <b>Figure 3.5.13</b>	168
Radial Air Faces after the run of the journal file 'temp1.jou' with slope angles: 45, 60.	
• <b>Figure 3.5.14</b>	169
Radial Plasma Faces after the run of the file 'temp1.jou' with the slope angles: 45, 60.	
• <b>Figure 3.5.15</b>	170
Radial Solid Faces after the run of the file 'temp1.joy' with the slope angles: 45, 60.	

• <b>Figure 3.5.16</b> .....	171
Axial Plasma Faces after the run of the file ‘temp1.jou’ with the slop angles: 45, 60.	
• <b>Figure 3.5.17</b> .....	172
Axial Solid faces after the run of the file ‘temp1.jou’ with the slope angles: 45, 60.	
• <b>Figure 3.5.18</b> .....	173
Axial Air Face after the run of the journal file ‘temp1.jou’ with the slope angles: 45, 60.	
• <b>Figure 3.5.19</b> .....	174
The Volumes of the Plasma after the run of the journal file ‘temp1.jou’ with the slope angles: 45, 60.	
• <b>Figure 3.5.20</b> .....	175
The Volumes of the Solid Zone after the run of the journal file ‘temp1.jou’ with the slope angles: 45, 60.	
<b>A1.3.2. FIGURES FOR THE SLOPE ANGLES: 45, 90</b> .....	176
• <b>Figure 3.5.21</b> .....	177
Radial Air Faces after the run of the journal file ‘temp1.jou’ with the slope angles: 45, 90.	
• <b>Figure 3.5.22</b> .....	178
Radial Plasma Faces after the run of the journal file ‘temp1.jou’ with the slope angles: 45, 90.	
• <b>Figure 3.5.23</b> .....	179

	Axial Solid Faces after the run of the journal file ‘temp1.jou’ with the slope angles: 45, 90.	
•	<b>Figure 3.5.24</b> .....	180
	Axial Plasma Faces after the run of the journal file ‘temp1.jou’ with the slope angles: 45, 90.	
•	<b>Figure 3.5.25</b> .....	181
	Axial Solid Faces after the run of the journal file ‘temp1.jou’ with the slope angles: 45, 90.	
•	<b>Figure 3.5.26</b> .....	182
	Axial Air Face after the run of the journal file ‘temp1.jou’ with the slope angles: 45, 90.	
•	<b>Figure 3.5.27</b> .....	183
	The Plasma Volumes after the run of the journal file ‘temp1.jou’ with the slope angles: 45, 90.	
•	<b>Figure 3.5.28</b> .....	184
	The Solid Volumes after the run of the journal file ‘temp1.jou’ with the slope angles: 45, 90.	
•	<b>Figure 3.5.29</b> .....	185
	The Air Volume after the run of the journal file ‘temp1.jou’ with the slope angles: 45, 90.	
	<b>A1.3.3. FIGURES FOR THE SLOPE ANGLES: 90, 90</b> .....	186
•	<b>Figure 3.5.30</b> .....	187

Radial Air Faces after the runoff the journal file ‘temp1.jou’ with the slope angles:  
90, 90.

- **Figure 3.5.31** .....188  
Radial Plasma Faces after the run of the journal file ‘temp1.jou’ with the slope  
angles: 90, 90.
- **Figure 3.5.32** .....189  
Axial Plasma Faces after the run of the journal file ‘temp1.jou’ with the slope  
angles: 90, 90.
- **Figure 3.5.33** .....190  
Axial Solid Faces after the run of the journal file ‘temp1.jou’ with the slope angles:  
90, 90.
- **Figure 3.5.34** .....191  
Axial Air Face after the run of the journal file ‘temp1.jou’ with the slope angles:  
90, 90.
- **Figure 3.5.35** .....192  
The Plasma Volumes after the run of the journal file ‘temp1.jou’ with the slope  
angles: 90, 90.
- **Figure 3.5.36** .....193  
The Solid Volumes after the run of the journal file ‘temp1.jou’ with the slope  
angles: 90, 90.
- **Figure 3.5.37** .....194

The Air Volume after the run of the journal file ‘temp1.jou’ with the slope angles:  
90, 90.

#### **A1.4. FIGURES ILLUSTRATING THE CREATION**

#### **OF THE FILE ‘modify2’ ..... 195**

##### **A1.4.1. FIGURES FOR THE SLOPE ANGLES: 45, 60 ..... 195**

##### **• Figure 3.5.38 .....196**

Axial Air Faces after the run of the journal files consequently ‘temp1.jou’,  
‘modify1’, ‘temp2.jou’.

##### **• Figure 3.5.39 .....197**

Axial Air Faces after the run of the journal files consequently ‘temp1.jou’,  
‘modify1’, ‘temp2.jou’.

##### **A1.4.2. FIGURES FOR THE SLOPE ANGLES: 45, 90 ..... 198**

##### **• Figure 3.5.40 .....199**

Axial Air Faces after the consequent run of the files: ‘temp1.jou’, ‘modify1’,  
‘temp2.jou’.

##### **• Figure 3.5.41 .....200**

Axial Air Faces after the consequent run of the files: ‘temp1.jou’, ‘modify1’,  
temp2.jou’.

##### **A1.4.3. FIGURES FOR THE SLOPE ANGLES: 90, 90 ..... 201**

##### **• Figure 3.5.42 .....202**

Axial Air Faces after the consequent run of the files: ‘temp1.jou’, ‘modify1’,  
‘temp2.jou’.

##### **• Figure 3.5.43 .....203**

Axial Air Faces after the consequent run of the files: ‘temp1.jou’, ‘modify1’, ‘temp3.jou’.

## **A1.5. FIGURES ILLUSTRATING THE CREATION OF THE FILE**

**‘modify3\_d’ and ‘modify3\_s’** ..... 204

### **A1.5.1. FIGURES FOR THE CTREATION OF THE FILE ‘modify3\_d’**

**FOR THE SLOPE ANGLES: 45, 60** ..... 204

- **Figure 3.5.44** .....205  
Axial Plasma Edges for the “maximum”.
- **Figure 3.5.45** .....206  
Axial Plasma Edges for the “minimum”.
- **Figure 3.5.46** .....207  
Axial Solid Edges for the “maximum”.
- **Figure 3.5.47** .....208  
Axial Solid Edges for the “minimum”.
- **Figure 3.5.48** .....209  
Axial Air Edges after for the “maximum”.
- **Figure 3.5.49** .....210  
Axial Air Edges for the “minimum”.
- **Figure 3.5.50** .....211  
Radial Solid Edges for the “maximum”.
- **Figure 3.5.51** .....212  
Radial Solid Edges for the “minimum”.

• <b>Figure 3.5.52</b> .....	213
Radial Air Edges for the “maximum”.	
• <b>Figure 3.5.53</b> .....	214
Radial Air Edges for the “minimum”.	
• <b>Figure 3.5.54</b> .....	215
Elliptic Solid Edges for the “left”.	
• <b>Figure 3.5.55</b> .....	216
Elliptic Solid Edges for the “right”.	
• <b>Figure 3.5.56</b> .....	217
Elliptic Air Edges for the “left”.	
• <b>Figure 3.5.57</b> .....	218
Elliptic Air Edges for the “right”.	
• <b>Figure 3.5.58</b> .....	219
Elliptic Plasma Edges for the “left”.	
• <b>Figure 3.5.59</b> .....	220
Elliptic Plasma Edges for the “right”.	
• <b>Figure 3.5.60</b> .....	221
Axial Plasma Faces for the “left”.	
• <b>Figure 3.5.61</b> .....	222
Axial Plasma Faces for the “left”.	
• <b>Figure 3.5.62</b> .....	223
Axial Solid Faces for the “left”.	

• <b>Figure 3.5.63</b> .....	224
Axial Solid Faces for the “right”.	
• <b>Figure 3.5.64</b> .....	225
Axial Air Faces for the “left”.	
• <b>Figure 3.5.65</b> .....	226
Axial Air Faces for the “right”.	
• <b>Figure 3.5.66</b> .....	227
Radial Air Faces for the “left”.	
• <b>Figure 3.5.67</b> .....	228
Radial Air Faces for the “right”.	
• <b>Figure 3.5.68</b> .....	229
Radial Solid Faces for the “left”.	
• <b>Figure 3.5.69</b> .....	230
Radial Solid Faces for the “right”.	
• <b>Figure 3.5.70</b> .....	231
The “shift” angle for the channel with two slope angles for the electric elements for the cut plane.	

#### **A1.5.2. FIGURES FOR THE CREATION OF THE FILE ‘modify3\_d’**

<b>FOR THE SLOPE ANGLES: 45, 90</b> .....	232
• <b>Figure 3.5.71</b> .....	233
Axial Plasma Edges for the “maximum”.	
• <b>Figure 3.5.72</b> .....	234

	Axial Plasma Edges for the “minimum”.	
•	<b>Figure 3.5.73</b> .....	235
	Axial Solid Edges for the “maximum”.	
•	<b>Figure 3.5.74</b> .....	236
	Axial Solid Edges for the “minimum”.	
•	<b>Figure 3.5.75</b> .....	237
	Axial Air Edges for the “maximum”.	
•	<b>Figure 3.5.76</b> .....	238
	Axial Air Edges for the “minimum”.	
•	<b>Figure 3.5.77</b> .....	239
	Radial Solid Edges for the “maximum” and “minimum”.	
•	<b>Figure 3.5.78</b> .....	240
	Radial Air Edges for the “maximum”.	
•	<b>Figure 3.5.79</b> .....	241
	Radial Air Edges for the “minimum”.	
•	<b>Figure 3.5.80</b> .....	242
	Elliptic Solid Edges for the “left”.	
•	<b>Figure 3.5.81</b> .....	243
	Elliptic Solid Edges for the “right”.	
•	<b>Figure 3.5.82</b> .....	244
	Elliptic Air Edges for the “left”.	
•	<b>Figure 3.5.83</b> .....	245

Elliptic Air Edges for the “right”.	
• <b>Figure 3.5.84</b> .....	246
Elliptic Plasma Edges for the “left”.	
• <b>Figure 3.5.85</b> .....	247
Elliptic Plasma Edges for the “right”.	
• <b>Figure 3.5.86</b> .....	248
Axial Plasma Faces for the “left”.	
• <b>Figure 3.5.87</b> .....	249
Axial Plasma Faces for the “right”.	
• <b>Figure 3.5.88</b> .....	250
Axial Solid Faces for the “left”.	
• <b>Figure 3.5.89</b> .....	251
Axial Solid Faces for the “right”.	
• <b>Figure 3.5.90</b> .....	252
Axial Air Faces for the “left”.	
• <b>Figure 3.5.91</b> .....	253
Axial Air Faces for the “right”.	
• <b>Figure 3.5.92</b> .....	254
Radial Air Faces for the “left”.	
• <b>Figure 3.5.93</b> .....	255
Radial Air Faces for the “right”.	
• <b>Figure 3.5.94</b> .....	256

Radial Solid Faces for the “left”.

### **A1.5.3. FIGURES FOR THE CREATION OF THE FILE ‘modify3\_s’**

<b>FOR THE SLOPE ANGLES: 90, 90</b>	<b>257</b>
• <b>Figure 3.5.95</b>	<b>258</b>
Radial Solid Faces for the “right”.	
• <b>Figure 3.5.96</b>	<b>259</b>
Axial Plasma Edges for the “left”.	
• <b>Figure 3.5.97</b>	<b>260</b>
Axial Plasma Edges for the “right”.	
• <b>Figure 3.5.98</b>	<b>261</b>
Axial Solid Edges for the “left”.	
• <b>Figure 3.5.99</b>	<b>262</b>
Axial Solid Edges for the “right”.	
• <b>Figure 3.5.100</b>	<b>263</b>
Axial Air Edges for the “left”.	
• <b>Figure 3.5.101</b>	<b>264</b>
Axial Air Edges for the “right”.	
• <b>Figure 3.5.102</b>	<b>265</b>
Radial Edges for the “left” and “right”.	
• <b>Figure 3.5.103</b>	<b>266</b>
Radial Edges for the “left” and “right”.	
• <b>Figure 3.5.104</b>	<b>267</b>

Radial Air Edges for the “right”.

- **Figure 3.5.105** .....268  
Round (Elliptic) Solid Edges for the “down”.
- **Figure 3.5.106** .....269  
Round (Elliptic) Solid Edges for the “up”.
- **Figure 3.5.107** .....270  
Round (Elliptic) Air Edges for the “down”.
- **Figure 3.5.108** .....271  
Round (Elliptic) Air Edges for the “up”.
- **Figure 3.5.109** .....272  
Round (Elliptic) Plasma Edges for the “down”.
- **Figure 3.5.110** .....273  
Round (Elliptic) Plasma Edges for the “up”.
- **Figure 3.5.111** .....274  
Axial Plasma Faces for the “down”.
- **Figure 3.5.112** .....275  
Axial Plasma Faces for the “up”.
- **Figure 3.5.113** .....276  
Axial Solid Faces for the “down”.
- **Figure 3.5.114** .....277  
Axial Solid Faces for the “up”.
- **Figure 3.5.115** .....278

Axial Air Faces for the “down”.	
• <b>Figure 3.5.116</b> .....	279
Axial Air Faces for the “up”.	
• <b>Figure 3.5.117</b> .....	280
Radial Air Faces for the “down”.	
• <b>Figure 3.5.118</b> .....	281
Radial Air Faces for the “up”.	
• <b>Figure 3.5.119</b> .....	282
Radial Solid Edges for the “down”.	
• <b>Figure 3.5.120</b> .....	283
Radial Solid Faces for the “up”.	
• <b>Figure 3.5.121</b> .....	284
The “shift” angle for the Straight Channel (slope angles are: 90, 90).	
<b>APPENDIX A2</b> .....	285
<b>FIGURES ILLUSTRATING NOZZLE – COMBUSTOR GEOMETRY AND MESH</b>	
<b>APPENDIX A3</b> .....	296
<b>FIGURES ILLUSTRATING STRAIGHT CONDUCTIVITY CHANNEL GEOMETRY AND MESH</b>	
<b>APPENDIX A4</b> .....	315
<b>FIGURES ILLUSTRATING DCW MHD CHANNEL GEOMETRY AND MESH</b>	
<b>APPENDIX A5</b> .....	335

**FIGURES ILLUSTRATING MESH QUALITY OF STRAIGHT CONDUCTIVITY  
AND DCW MHD CHANNELS**

**APPENDIX A6 ..... 339**

**FIGURES ILLUSTRATING COMPUTATIONAL DATA FOR NOZZLE -  
COMBUSTOR**

**APPENDIX A7 ..... 350**

**FIGURES ILLUSTRATING COMPUTATIONAL DATA FOR STRAIGHT  
CONDUCTIVITY CHANNEL**

**APPENDIX A8 ..... 355**

**FIGURES ILLUSTRATING COMPUTATIONAL DATA FOR DCW MHD  
CHANNEL**

**APPENDICES (links)**

**B. GEMS**

- [gems\\_input\\_1](#) (3D conductivity/MHD plasma channel)
- [gems\\_input\\_2](#) (2D combustor-nozzle driven flow)
- The description of [GEMS®](#)

**C. NOZZLE WALL CONTOUR GENERATOR**

- [create\\_profile](#)

**D. SOURCE CODE UTGRID®, SAMPLE OUTPUT**

- [source\\_1](#) (available at the request)
- [source\\_2](#) (available at the request)
- [source\\_3](#) (available at the request)

- [source\\_4](#) (available at the request)
- [source\\_5](#) (available at the request)
- [source\\_6](#) (available at the request)
- [source\\_7](#) (available at the request)
- [source\\_8](#) (available at the request)
- [source\\_9](#) (example)
- [source\\_10.1](#) (example)
- [source\\_10.2](#) (example)
- [source\\_10.3](#) (example)
- [source\\_11](#) (example)
- [source\\_12](#) (example)
- [source\\_13](#) (example)
- [source\\_14](#) (example)
- [source\\_15](#) (example)
- [source\\_16](#)
- [source\\_17](#)
- [source\\_18](#) (example)
- [source\\_19](#) (example)
- [source\\_20](#) (example)
- [source\\_21](#) (example)
- [source\\_22](#) (example)
- [source\\_23](#) (example)

- [source\\_24](#) (example)
- [source\\_25](#) (available at the request)
- [source\\_26](#) (available at the request)
- [source\\_27](#) (available at the request)
- [source\\_28](#) (available at the request)
- [source\\_29](#) (available at the request)

**VITA** ..... 365



## CHAPTER 1

### INTRODUCTION

#### 1.1. Background.

A numerical model was needed of a small-scale, short-duration plasma generator that was being used to provide experimental plasma conditions for research into magneto-hydrodynamic (MHD) flows. Experimental research on MHD flow conditions is being done at [UTSI](#) in support of an overall research investigation of a Hypersonic Vehicle Electric Power System ([HVEPS](#)) Program. The subtask performed by [UTSI](#) personnel is to develop and operate the small scale MHD plasma generator and to take experimental data that defines the plasma parameters of temperature, pressure, electrical conductivity, and other key physical properties of the plasma.

Concurrently at [UTSI](#), a numerical model of MHD flows is being developed based on the General Equation and Mesh Solver (GEMS®) code [1-4]. The GEMS® code is being used to model not only the equations of motion of plasma, but also the attendant Maxwell equations which describe the generation and propagation of the electromagnetic fields induced by the electric currents in the MHD flow field as well as effects due to applied external magnetic fields. Since time-dependant electric and magnetic fields propagate not only through the fluid regimes but also into and through the solid regions (anode and cathode frames of an MHD channel, through the walls of a conductivity channel, through metallic duct components, etc.) of the apparatus used to contain and direct the MHD flows, GEMS® is configured to compute the evolution of magnetic and electric field in

space and time. Both 2 and 3-dimensional models are being developed. Sub-models for the physical processes in a plasma flow, such as a turbulence model [5], and an electric current model, will be required for implementation into the GEMS® code. In addition to physical sub-models, another important part of the development of an MHD-based version of the GEMS® code is a computational mesh generation algorithm. MHD plasma flows are compressible, turbulent, electrically-conducting gas flows with boundary layer regions near the insulator and conductor wall surfaces. Within the boundary layers, turbulence, heat, and electrical parameters vary significantly. Therefore, from a numerical modeling requirement, the boundary layer regions require high numerical resolution, that is, fine meshes in the directions transverse to the direction of flow. This requirement of high resolution within the wall boundary layers, coupled with the need to resolve electric current streams along planes diagonal to the main flow direction, in diagonal Conducting Wall channels (DCW), pose significant challenges to the construction of numerical meshes for the GEMS® code flow simulations. System components that must be represented by numerical mesh systems include a cylindrical combustion chamber, a converging-diverging supersonic nozzle, a nozzle extension (as ceramic ring fitted to the nozzle), a cylindrical conductivity channel, and either a segmented Faraday channel or a diagonal conducting wall (DCW) channel. Each component requires its own optimized mesh. The meshes must interface one another properly where the components are mated together. Different boundary layer resolution requirements are needed for each component.

Therefore, a powerful, flexible, 2 and 3-dimensional mesh generation algorithm was required in order to support the adaptation of the GEMS® code to the numerical

simulation of the experimental facility that has been developed and operated at [UTSI](#) in support of the [HVEPS](#) research program. The purpose of this thesis research was to develop and validate such a mesh generation algorithm in support of the [HVEPS](#) project, which was supported by [General Atomics Corporation](#) and other affiliated companies.

## **1.2. Hardware Modeling.**

Shown below are photographs of the experimental facility at [UTSI](#) used to generate plasma flow. The figures (**Figure 1.2.1**, **Figure 1.2.2** and **Figure 1.2.3**) show the cylindrical combustor, supersonic nozzle, and water-cooled exhaust diffuser coupled together. The water-cooled exhaust diffuser exhausts to the atmosphere as shown in **Figure 1.2.1**.

**Figure 1.2.2** shows the head (fuel-oxygen system) end of the combustor. **Figure 1.2.3** shows the supersonic nozzle and straight-walled, cylindrical conductivity channel attached to the ceramic ring fitted to the nozzle.

**Figure 1.2.4** shows the control room with the combustor and data gathering equipment based on using the [National Instrument Labview™](#) program.

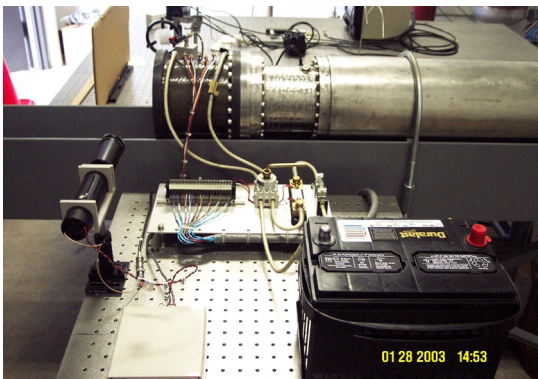
A schematic diagram of the electrical conductivity measurement system for the HVEPS real experiment on plasma conductivity in a straight channel is represented in **Figure 1.2.5**.



**Figure 1.2.1.** Full flow tract.



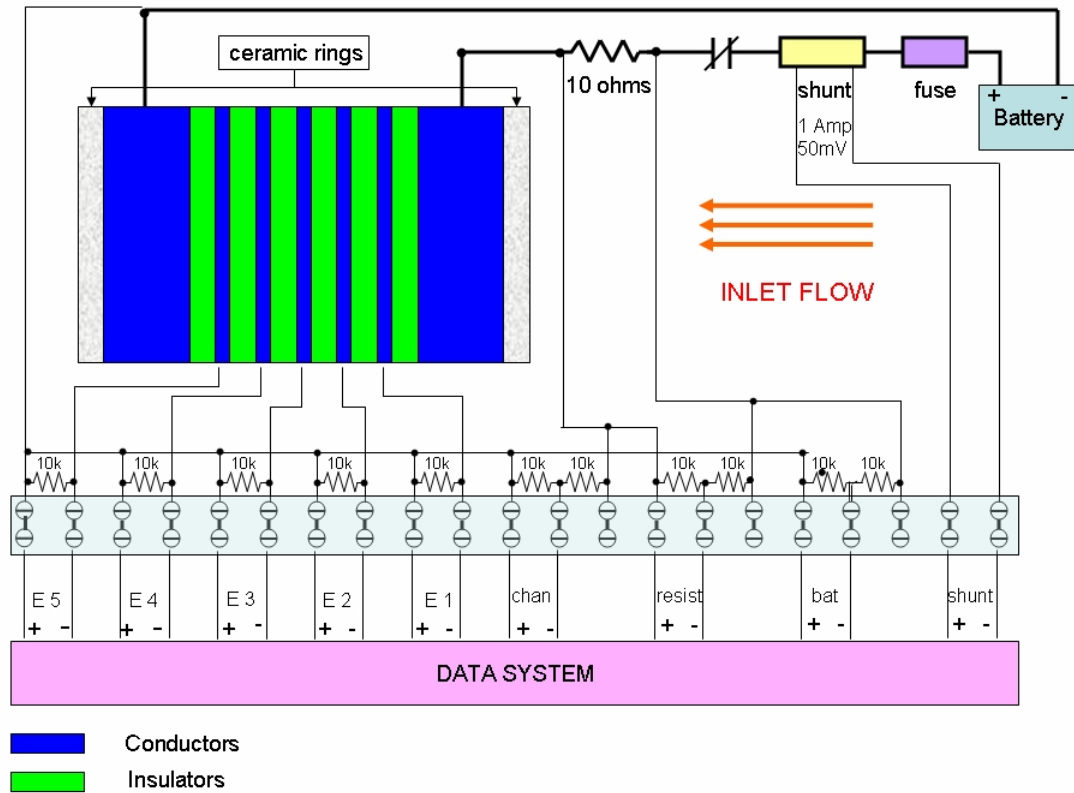
**Figure 1.2.2.** Combustor chamber.



**Figure 1.2.3.** Conductivity channel.



**Figure 1.2.4.** Control room.



**Figure 1.2.5.** Schematic diagram of the electrical conductivity measurement system for the HVEPS experiment, using a the straight conductivity channel .

Briefly, the HVEPS plasmas are generated by firing a gaseous fuel mixture of MAPP gas (a stabilized mixture of propadiene and methylacetylene in approximately 50:50 mixture by weight) with pure gaseous oxygen. The combustion products are seeded with an aqueous solution of potassium carbonate to about 3% of the total flow. The typical oxygen-fuel ratio is about 2.55, producing a (seeded) adiabatic temperature of about 3270K. Details of this facility and discussion of its performance as a plasma generation are provided by Dr. Roger Crawford [5].

UTSI Combustion Driven Flow Facility parameters:

- High enthalpy flow simulation, typical of high Mach number, hydrocarbon fueled systems.
- Oxygen (gas) and MAPP-C<sub>3</sub>H<sub>4</sub> (liquid) combustor operating 10% fuel rich with 3% potassium seed.
- Stagnation enthalpy equivalent to flight at  $M = 8$   
 $T_0 = 3274\text{K}$  at 2 ATM,  $T_0 = 3438\text{K}$  at 6 ATM.
- Electrical conductivity range representative of Mach numbers 5 - 7 flight with combustion augmentation and potassium seeding.
- Maximum utilization of existing hardware from previous MHD projects at UTSI.

## CHAPTER 2

### **THE APPROACH TAKEN TO CREATE A GENERAL, FLEXIBLE, POWERFUL MESH GENERATION ALGORITHM**

#### **2.1. Overview of the approach.**

The approach taken to create a powerful, flexible, and general mesh generation algorithm was to create a computer program that acts as an interface for two other programs that can be used to generate an optimum mesh geometry that the [GEMS®](#) code can use. The new program, and its implementation with the other two existing codes, a mesh generation code and a mesh translation code, is the basis of this thesis. The new program was developed by the author and is called UTGRID®. It interfaces with a commercially-available mesh preprocessor code called GAMBIT™, and a proprietary mesh post-processor code, called PGRID®, developed by Dr. Ding Li [6]. Basically, the approach to providing suitable mesh geometries for representing components of the [UTSI](#) plasma generation facility is to use UTGRID® to prepare a table of input values to GAMBIT™ that represent all of the parameters that GAMBIT™ needs in order to construct a mesh of a particular type. The types of meshes that GAMBIT™ can construct will be discussed later. UTGRID® greatly simplifies the problem of identifying all of the needed characteristics of a mesh required to represent a complex flow field of the type used to make calculations of compressible turbulent flows with boundary layers and MHD effects. UTGRID®, therefore, becomes the principal interface for the user in order to

create the necessary meshes for the GEMS® code. Details of UTGRID® will also be discussed later.

PGRID® is a grid translation program written by Dr. Ding Li, who is one of the co-developers of the GEMS® code. PGRID® translates the computational mesh created by GAMBIT™, which was based on data input via UTGRID®, and converts the GAMBIT™ mesh into a form recognized by the GEMS® code. A brief discussion of PGRID® will also be provided later.

Besides creating the input data interface code UTGRID®, the author was also responsible for interlinking GAMBIT™ and PGRID®, with assistance and consultation from Dr. Ding Li. Thus, the author developed the seamlessly coupled, interactive algorithm that links UTGRID®, GAMBIT™, and PGRID® to rapidly, and accurately construct well-designed computational meshes, with predefined regions of high numerical resolution for use by the modern, state-of-the-art GEMS® CFD code. The linking process is carried out using standard Fortran language and compile process [7-15].

In summary, the approach taken in this research to construct a rapid, flexible algorithm for constructing computational meshes of the kind needed to represent compressible, turbulent MHD combustor, nozzle, and channel flows, was to link three separate, and sequentially used, but interactive, software codes for the mesh construction: UTGRID®, GAMBIT™ and PGRID®. UTGRID® was entirely developed by the author, and the interactive linking of UTGRID®, GAMBIT™ and PGRID® was also the author's responsibility.

In the following sections the various codes used to model the MHD plasma flows are described: [GAMBIT™](#), [PGRID®](#), [GEMS®](#) and [UTGRID®](#).

## **2.2. GAMBIT™.**

GAMBIT™ is the preprocessor of the commercially-available [FLUENT™](#) Computational Fluid Dynamics suite of codes. GAMBIT™ can be used as a separate mesh generation process. Graphical interface of this product can be used for visualization of any resultant grid generated, or the grid structure can be analyzed in a text mode.

Different CFD problems require different mesh types, and GAMBIT™ provides all of the options need in a single package. GAMBIT™'s meshing toolkit lets you decompose geometries for structured hex meshing or perform automated hexahedral meshing with the controls. Triangular surface meshes and tetrahedral volume meshes can be created within a single environment, along with mesh elements such as a Cartesian core, pyramids and prisms needed for hybrid meshing, using automatic size distribution, to correctly capture sharp curvature and small gaps in geometries modeled. GAMBIT™'s semi-automatic cleanup tools can be used to repair and prepare mesh geometry for high quality meshing. Holes, overlapping faces, small features, sharp corners, etc. are quickly identified and various tools are directly accessible to resolve these kinds of geometric features.

For 2D grid generation the author used a quadrilateral element algorithm, and for 3D, elements were created based on the hexahedral Cooper's algorithm, described in the reference manual of [GAMBIT™](#). In the present study, mesh generation in 2D has been used for computational modeling of combustor-nozzle-extension domains. 3D computational meshes were used for the straight conductivity channel and the diagonal MHD plasma channel to create grid files acceptable by PGRID® [6]. GAMBIT™ contains routines for outputting mesh construction files in various formats.

The GAMBIT™ grid solvers and generated with them output formats used as input for PGRID® and for data presentation are:

- **GENERIC** – outputs mesh files in the neutral file format with extension: \*.neu for input to PGRID®
- **FLUENT 5/6** – outputs mesh files in the format with extension: \*.msh, for input to a graphical visualization and presentation program called TECPLOT™.

Input to GAMBIT™ is required in files with formats determined by the extensions: \*.dbs and \*.jou (called a journal file). The creation of these files by UTGRID® allows for much shorter modeling times and for a step-by-step mesh generation process, respectively. The files with the extension: \*.dbs are the output of GAMBIT™ in its general format for the geometry (not necessarily meshed, and if meshed, geometry plus mesh). According to its creators, GAMBIT™ provides a mesh generation process that is modern, convenient, flexible, allowing complex geometries to be represented by discrete mesh systems, appropriately interfaced for representing complicated geometries.

However, geometries as complicated as that required to represent the diagonal MHD plasma channel with arbitrary or variable amounts of electric elements and/or duct and channel segments, such as electrodes with fully controllable parameters for boundary layer representation in all geometrical parts, require extensive element identification and numbering with all of the multiple zones of the computational domain. Manual input of all of these mesh identification and control variables thus requires a very large amount of time to be fully done. Each part of the mesh construction process for this grid would be represented by specific text commands, the whole set of which would give a

complete list of directives for what GAMBIT™ should do at each step to construct the mesh. That is, each step of construction using GAMBIT™ has to be separately defined by the user. Once the user has created the mesh files for this geometry, he must save the mesh files in journal (\*.jou) format, which for very complicated geometry and mesh, might be a very large file to keep it flexible for parametric changes.

Thus the problem of grid generation with GAMBIT™ requires the creation of a set of acceptable and formatted input commands, and necessary flexibility has to be organized into the command input files to GAMBIT™ to affect the process of journal file generation.

This problem of manual input of command and control parameters for GAMBIT™ was solved by creation of UTGRID® (**University of Tennessee Grid Generator**, described in Chapter 3), entirely created by the author of this Thesis.

GAMBIT™ also incorporates a separate numerical procedure for generation of the wall contour or nozzle profile for the converging-diverging supersonic nozzle component of the HVEPS facility. This nozzle profile was generated by first entering in only a few real measured nozzle contour points. Then the numerical model of the nozzle and the nozzle wall profile or contour was constructed by an approximation algorithm (“**nerb**” – approximation) that ensures that the profile has continuous first and second derivatives. This profile can be created with the journal file for GAMBIT™ ‘[create\\_profile](#)’ included into the **APPENDIX C**. The procedure to create nozzle wall contour profile is described in section **3.4** in this thesis and represents the optional operation of the UTGRID® and created by the author of this Thesis.

With smoothly varying slope and the rate of change of slope, the numerical solutions of the nozzle flow field do not show or contain false shock or expansion waves. This gives a much higher quality numerical representation of the actual nozzle flow fields. GAMBIT™'s mesh generation algorithm is thus fully defined by a journal file generated by UTGRID®, which contains the complete set of directives for GAMBIT™ with all required parameters for GAMBIT™'s mesh generation process.

The output mesh files are saved in formats: \*.neu and \*.msh and for general geometry and mesh data saving purposes in format: \*.dbs. These files represent the input files for the PGRID®, TECPLOT™ and GAMBIT™ respectively (see below).

### **2.3. PGRID®.**

PGRID® is a utility created by Dr. Ding Li at [UTSI](#) to convert a mesh system that is defined in one of the acceptable formats listed below, to a format acceptable by the GEMS® Code, and also, for factorization of a complete meshed computational domain into interfaced subsystems or sub-domains consisting of one or more smaller domains defined by the user for parallel computations. PGRID® has to be compiled with a certain library ('libmetis.o') with a [Fortran 90 Compiler](#) [16] or [Lahey Fortran 95 Compiler](#) [21] for the particular computer architecture where the executable file is going to be run. PGRID® needs a large memory allocation for the process of conversion, partitioning and organization of the mesh sub-domains for the parallel numerical computations carried out by GEMS®. PGRID® has an interactive mode, and is easy to use.

Acceptable mesh input formats for PGRID® are:

- Uniform meshes;
- Structured meshes;
- Unstructured hybrid (GEMS-DFD) meshes;
- Unstructured hybrid (GEMS-DFD) meshes with volume indices;
- VGRID meshes [17];
- CFDRC meshes [18];
- GAMBIT neutral mesh file format [19];
- FIELDVIEW unstructured mesh format (for 3D only) [20].

An interactive choice of the user is the dimension of the problem – 1D, 2D or 3D.

Another interactive choice is the amount of partitions used for parallel calculations – it can be any arbitrary number, but it has to be chosen with respect to amount of available computational nodes, or computers in a cluster group. Utility code PGRID® redefines all elements, faces, nodes and cells for all zones. Normal termination of the execution of PGRID® will not give any error messages.

The unique utility, PGRID®, is a powerful tool for creating input files that define the meshes utilized by the GEMS® Code.

For any questions about this utility, refer to the author:

- [dli@purdue.edu](mailto:dli@purdue.edu)
- [dli@utsi.edu](mailto:dli@utsi.edu)
- <http://tspcpc110.ecn.purdue.edu/>

## 2.4. GEMS®.

[GEMS – General Equations and Mesh Solver](#) was created by Dr. Ding Li and Dr. Charles Merkle at [UTSI](#) and extended to plasma and magneto-hydrodynamic (MHD) flow by them in 2002. GEMS® can be applied to a wide spectrum of numerical simulations in general CFD and MHD areas. The GEMS® code has been described by its authors in various papers.

The following description was provided by Dr. Charles Merkle and Dr. Ding Li.

- The **GEMS®** (General Equation and Mesh Solver) code is a flexible new unstructured grid code that is designed for the numerical solution of arbitrary sets of conservation equations on arbitrary meshes. The primary focus of **GEMS®** is on computational fluid dynamics (CFD) where it has been used for perfect gases, incompressible fluids, real gases, supercritical fluids, cavitations and multi-component and multi-phase fluids. In addition it has been used for combustion, MHD and conjugate heat transfer problems and has a preliminary fluid-structure interaction capability. The code is fully three-dimensional but can be run in two-dimensional or axisymmetric modes also. Both time-accurate and steady-state computations can be handled.
- The code is written in Fortran 95 using all the modern language constructs and runs on a variety of platforms (Unix, Linux and Windows). The code uses MPI for parallel processing (limited to **UNIX** and **Linux**) and is fully implemented on PC clusters.

- The unstructured grid capability in the code uses a cell-centered format that is applicable to cells with any number of faces. Cells of different shapes (hexahedrons, tetrahedrons, prisms, etc.) are routinely mixed in a single grid, and either structured or unstructured grids can be included. The code also includes a preprocessing routine to convert multi-block structured (or unstructured) grids to single block grids for the ‘unstructured’ computation. This single-block capability minimizes load-balancing issues in parallel computations as the total number of grids is allocated equally among processors. Capabilities are included for both fixed and moving grids.
- The code involves several solution algorithms including Euler explicit, Gauss-Seidel, line Gauss-Seidel and GMRES. Line Gauss-Seidel is most often used and is implemented on an unstructured grid by tracing lines through the field to be used for tri-diagonal solution. These lines are constructed in such a manner that they become identical to a structured grid algorithm when implemented on a structured grid. We are currently implementing a ‘structured-grid algorithm’ that adds a pointer to second nearest neighbors to enable third-order biased flux reconstruction in identical fashion with structured grid solvers when the grid is labeled as ‘structured’.
- The code also allows capability for multiple ‘zones’ in which a different set of conservation equations are specified in each zone. For example, in a conjugate heat transfer problem, the continuity, momentum and energy equations are solved in the ‘fluid’ zone, while only the energy equation is solved in the ‘solid’

zone. Similarly, MHD problems are divided into fluid and solid zones with continuity, momentum, energy, species and the magnetic diffusion equation solved in the fluid zone, but only the energy and magnetic diffusion equation specified in the solid zone. A vacuum zone is also frequently employed where only the magnetic diffusion equation is solved. The zonal decomposition also potentially provides for solving the Euler equations in one zone, the Navier-Stokes in a second and the RANS equations in a third, should this capability be of interest. [2,3].

Representative applications of the code have been made to a very wide variety of problems. The results with UTGRID® mesh and GEMS® are published [1].

Generally the GEMS® is code runs on unstructured meshes.

## CHAPTER 3

### **AUTOMATED MESH GENERATOR UTGRID®**

UTGRID® was created by the author within the period 2002-2004 as part of a **Complete Set of Tools (CST)** for numerical simulation of flow fields experimentally produced in the HVEPS Project.

It is linked together independently with the set of program products:

- [GAMBIT™](#)
- [PGRID®](#)
- [TECLOT™](#)
- [GEMS®](#)
- [UTGRID®](#)

All program products in this set represent a **Complete Set of Tools (CST)** for numerical simulation of CFD and Magneto-Hydrodynamic flow fields, including magnetic and electric fields and electric current in plasmas.

Also a wide spectrum of complicated computational domains can be modeled with the best quality of mesh, and with fully controllable parameters from input files.

UTGRID® by itself can also be extended to a hexahedral mesh generation with additional module(s) included into the source for the mesh generation process for other specific domains. This thesis attempts to provide a complete and detailed users' manual

so that anyone using the assembly of codes (UTGRID®, GAMBIT™, PGRID® and GEMS®) can do so in an efficient manner.

The current status and configuration of program UTGRID® is indicated and detailed by and within the following list of Fortran-90 source files, input and output files. Source files with the extension: \*.f are Fortran-90 source files requiring compilation before the user can apply UTGRID® to a computational mesh construction project. In the list below, the source files and input/output files are described or listed by their relative order (source\_1, source\_2, etc.), by the name of their listings (**hveps.f**, **utgrid\_input.f**, etc.), and by a very brief description of what function each Fortran-90 module performs and input/output files represent. The actual Fortran 90 source code listings, and input/output file format structures are provided in **APPENDIX D**.

For further clarify the reader's understanding, the author would like to make a short comment about module structure in Fortran 90.

A module is a separate, closed program unit which can contain: data declarations, namelists of variables, and sub-program units ("subroutines" or "functions"). Modules can use other modules which have to be declared in a very first module declaration, which comes as the first statement, by construction "**use** <name\_of\_module>". The main program unit can use modules by use of statement of the form "**use** <name\_of\_module>". By using a module structure to create the source code, a user can avoid any applications of "**common**" statements and yet provide a flexible tool for data communication between modules and between modules and main program unit. Using modules represents either calling subroutines or functions contained in the module or/and sharing data by using "**use**<name\_of\_module>" statement. Thus, for all modules used by main program unit

any variable will be common including main program unit and, in turn, if some module X uses a certain set of other modules, e.g., A, B, etc., all variables with the same names in X, A, B, etc. will be common and will have the same values, which means that their names will be assigned to the same memory cell.

### **3.1. File system and module structure of UTGRID®.**

The process of the modification, which will be described in the section 3.5, is the process of changing the names assigned by GAMBIT™ to the geometrical elements, such as edges, faces and volumes.

All files are located and named as below in **APPENDIX D**:

source_1	‘UTGRID.f’	- contains main program unit
source_2	‘utgrid_input.f’	- contains module: ‘input_data’
source_3	‘chan_calc.f’	- contains module: ‘channel_calculation’
source_4	‘chan_grid.f’	- contains module: ‘channel_grid’
source_5	‘noz_calc.f’	- contains module: ‘nozzle_calculation’
source_6	‘noz_grid.f’	- contains module: ‘nozzle_grid’
source_7	‘litteral.f’	- contains module: ‘char’
source_8	‘spint.f’	- contains module: ‘print_data’
source_9	‘input’	- input data for UTGRID®
source_10.1	‘nozzle.jou’	- output file for nozzle
source_10.2	‘channel_45_90.jou’	- output file for channel
source_10.3	‘UTGRID.dat’	- output file for data printed
source_11	‘modify1’	- file for first modification

source_12	'modify2'	- file for second modification
source_13	'modify3_s'	- file for third (last) modification for straight channel
source_14	'modify3_d'	- file for third (last) modification for diagonal MHD plasma channel
source_15	'noz_prof.dat'	- actual 2D nozzle profile or nozzle wall contour file, represented in 3D. That means this is the array (x, y, z), where for 2D representations, all coordinates $z = 0$
source_16	'Makefile'	- for compilation and objective files deletion
source_17	'go-utgrid'	- script file for executable file creation
source_18	'temp1.jou'	- temporary created file after first modification
source_19	'temp2.jou'	- temporary created file after second modification
source_20	'temp3.jou'	- temporary created file after third modification
source_21	'temp4_s.jou'	- temporary created file after the final journal file generation for straight channel
source_22	'temp4_d.jou'	- temporary file created after final journal file is generated
source_23	'der1.dat'	- first derivative of the nozzle profile or wall contour
source_24	'der2.dat'	- second derivative of the nozzle profile or wall contour

The following are files with user-defined arbitrary names; in the present study, the author defined them as follows:

- 'input' (source\_9)

- ‘nozzle.jou’ (source\_10.1, extension can not be changed)
- ‘channel\_45\_90.jou’ (source\_10.2, extension can not be changed)
- ‘UTGRID.dat’ (source\_10.3, extension can be any acceptable to text editor)
- ‘noz\_prof.dat’ (source\_15, extension can be any acceptable to text editor and/or by TECPLOT™)
- ‘der1.dat’ (source\_23, extension can be any acceptable to text editor and/or by TECPLOT™)
- ‘der2.dat’ (source\_24, extension can be any acceptable to text editor and/or by TECPLOT™)

All the above specified above files have to be consol input typed with the names given by the user when UTGRID® requests them. The author would suggest choosing mnemonically understandable file names.

The main program in UTGRID® asks for the names of the data files and locations (if the user has only one work directory, he can enter just the names of appropriate files without their locations) in accordance to ‘control’ section in input namelist in the file: ‘source\_9’. The user can give any name for input file but must use the data input format defined in the existing input file, ‘source\_9’, shown in **APPENDIX D**.

The file ‘source\_15’ is the nozzle’s input wall or contour profile and has to have extension: \*.dat

The output file ‘source\_10.3’, in which some output data is being printed, can have any extension and can be read with any text editor.

Output files for nozzle's and channel's mesh have to have the extension: \*.jou, and be given names by the user, like: '<any\_user\_name>.jou'. In the present Thesis these files are: 'source\_10.1' and 'source\_10.2' respectively.

Output files for the first and second derivatives of the nozzle can have any extension acceptable to a text editor, and/or by TECPLOT™. Thus, for TECPLOT™, they can have extension: \*.dat for its appropriate data loader. Note that these files will be automatically generated by UTGRID® if the appropriate control parameter is as follows: l\_der = 1, and not generated otherwise.

Source files should be compiled with the Fortran 90 Compiler.

In the present study, the suite of compiled Fortran 90 sources and modules were compiled using a [Portland Fortran 90 Compiler](#) [7-16], for operation on a UNIX-based cluster of desktop computers.

For detailed instructions about the process of compilation, and the creation of the executable file, see the section **3.5**.

The suite of sources after compilation can be run in an interactive mode, or can be done according to the set of control parameters in the input file (section 'control' in the namelist).

It is not necessary to keep the set of input and output files in the work directory, but the locations of these files has to be consol input typed when UTGRID® requests them. Only 'modification' files have to be in the work directory. Certainly, the easiest way to run UTGRID® is to keep all needed files in one work directory; in this case the user will have to type just the names of the files requested by UTGRID®, and not their locations. All objective files are deleted automatically by the Fortran 90 compiler. All temporary

files will be created in the work directory. All Fortran 90 files must be located in the work directory.

The extension \*.f for the Fortran-90 files was chosen to activate the colour representation of the text in an Emacs text editor in a Linux operating system. It can be changed to the extension \*.f90, without any affects on compilation.

After the output files are created, any additional temporary files can be deleted manually.

With the next run of UTGRID® new temporary files are created.

When in the namelist of variables the parameter l\_mod is set to zero, the user will have to run both GAMBIT™ and UTGRID® in interactive mode, and the user must answer all required questions posed by UTGRID®. The user must manually check the names of the created geometry: edges, faces and volumes which then consequently requires the running in GAMBIT™ of temporary files created by UTGRID® with the extension \*.jou. These files are consequently: temp1.jou, temp2.jou, temp3.jou and temp4.jou. The order of running of those files in GAMBIT™ is established by the order of messages of UTGRID® which are output on the consol terminal.

This entire operating procedure for the suite of codes is described in detail in the “**Manual**” in the section **3.5**.

When modification files are created, the user can arbitrarily choose values for all other variables in the namelist of the input variables. For MHD channel representations, the only crucial input value is the number of electric elements (insulators) for either diagonal or straight channels. All other meshing parameters are arbitrary when the number of electric elements (insulators) has been defined and suitable modification files are created.

### 3.2. Listing of modules and their functions, input file.

The following subsection lists the modules, gives their source destinations, and briefly describes their functions.

---

File order	Source list name	Functional description
------------	------------------	------------------------

---

source_1	<b>UTGRID.f</b>	Main module which organizes input/output, interactively requests for the all needed files names and their locations, provides communications with modules and between modules.
----------	-----------------	--

Uses the modules:

- ‘input\_data’
- ‘nozzle\_calculation’
- ‘nozzle\_grid’
- ‘channel\_calculation’
- ‘channel\_grid’
- ‘char’
- ‘print\_data’

External program unit is subroutine ‘rows’ which calculates number of rows and depth of boundary layer, and incorporates the routine for mesh spacing in the boundary layer called a **Geometrical Progression Solver**. All modules in statement ‘include’ can be commented and compiled separately and independently. However, the appropriate file with the name ‘Makefile’

(source\_16) and script file '**go-utgrid**' (source\_17) undergo automatic compilation and creation of executable file when the user runs the utility '**make**' in a **UNIX** - like operating system.

source\_2      **utgrid\_input**

Module, consists of:

- namelist variables
- data declarations
- subroutine 'read\_input' (called from main)  
which reads input data from input file  
(source\_9)

Uses modules: no modules are used

source\_3      **chan\_calc.f**

Module, consists of:

- data declarations
- subroutine 'calc\_chan' (called from main)  
which provides all needed calculations of  
geometry and mesh for either straight or  
diagonal channel according to input file  
parameters
- subroutine 'allocate\_arrays' (called from  
main) which allocates memory for all needed  
arrays

- subroutine 'deallocate\_arrays' (reserved)  
which can deallocate memory for all needed arrays

Uses modules:

- 'input\_data'
- 'char'

source\_4      **chan\_grid.f**

Module, consists of:

- data declarations
- subroutine 'channel' (called from main)  
which provides grid generation process for either straight or diagonal channel according to input file parameters in input file  
(see source\_9)
- subroutine 'label' (called from main) which provides appropriate and needed names (if  $l\_mod = 0$ , see source\_9) for all geometric elements for the channel: faces, edges and volumes. It also has an interactive mode.  
(See section 3.5)

Uses modules:

- 'input\_data'
- 'char'

		<ul style="list-style-type: none"> <li>• ‘channel_calculation’</li> </ul>
source_5	<b>noz_calc.f</b>	<p>Module, consists of:</p> <ul style="list-style-type: none"> <li>• data declarations</li> <li>• subroutine ‘calc_noz’ (called from main) which provides all needed calculations for geometry and mesh for the nozzle (but does not provide nozzle mesh yet)</li> </ul> <p>Uses modules:</p> <ul style="list-style-type: none"> <li>• ‘input_data’</li> <li>• ‘channel_calculation’</li> <li>• ‘char’</li> </ul>
source_6	<b>noz_grid.f</b>	<p>Module, consists of:</p> <ul style="list-style-type: none"> <li>• data declarations</li> <li>• subroutine ‘nozzle’ (called from main) which provides the grid generation process for the nozzle and creates, as output of UTGRID®, journal input file for the GAMBIT™ grid generation process</li> </ul> <p>Uses modules:</p> <ul style="list-style-type: none"> <li>• ‘input_data’</li> <li>• ‘channel_calculation’</li> <li>• ‘nozzle_calculation’</li> </ul>

- ‘char’

source\_7

**litteral.f**

Module, consists of:

- data declarations
- function ‘i2s’ named in modules:
  - ‘chan\_grid’
  - ‘noz\_grid’
  - ‘chan\_calc’
  - ‘noz\_calc’
  - ‘print\_data’

This function converts integer numbers to the ASCII characters (created by Dr. Ding Liand and modified by the author of the Thesis)

- function ‘r2s’ named in modules:
  - ‘chan\_grid’
  - ‘noz\_grid’
  - ‘chan\_calc’
  - ‘noz\_calc’
  - ‘print\_data’

This function converts real numbers to the characters (created by Dr. Ding Li and modified by the author of the Theses)

source\_8

**sprint.f**

Module, consists of:

- data declarations
- subroutine ‘information’ (called from main) which provides printing of some of the channel’s and nozzle’s (reserved) data (geometry and mesh)

Uses modules:

- ‘input\_data’
- ‘char’
- ‘channel\_calculation’
- ‘nozzle\_calculation’

source\_9      **input**      This is an Input file consisting of data for the variables shown in the namelist in the module ‘utgrid\_input’ (see source\_9 in the **APPENDIX D**)

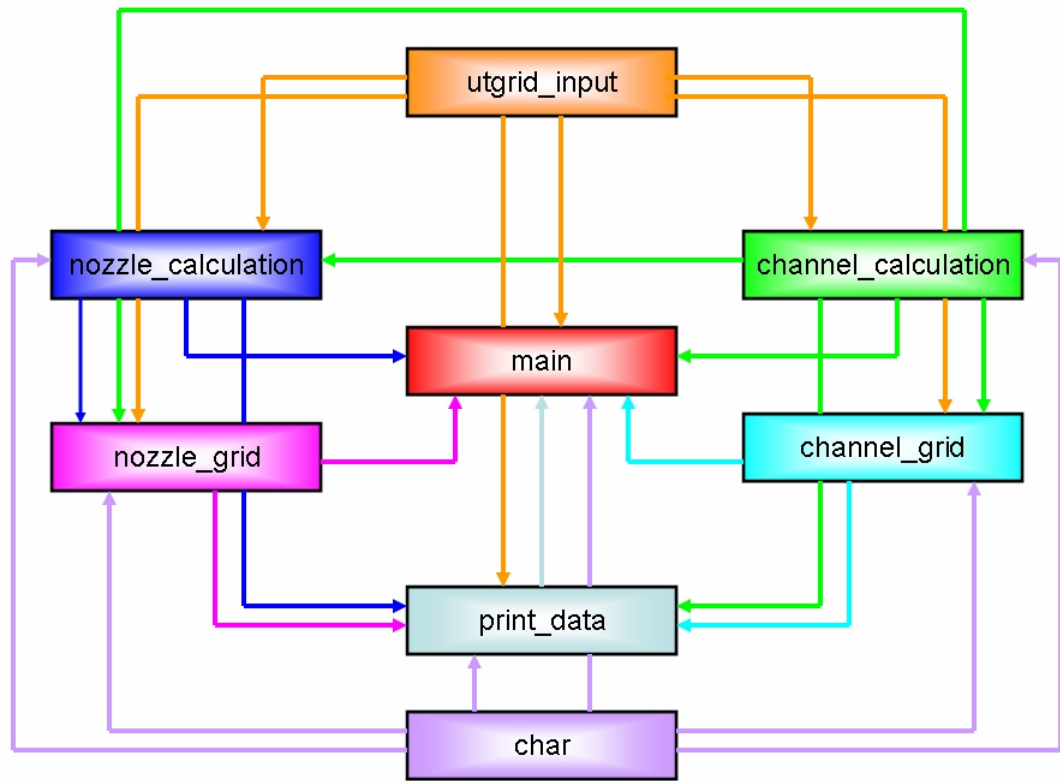
---

For the input file of the UTGRID® the user can choose any name, but has to console input type it and its location (if it is located not in work directory) when it is requested interactively by UTGRID®. The user also has to use only the format that is declared in source\_9 when inputting the needed numerical data in free form. All variables in the namelist are described by corresponding comment at the bottom of the input file and in the section 3.5.3, and are should be clearly understood by the user.

Schematically, the module structure of UTGRID® is represented in **Figure 3.2.1** below.

On this figure, the relation “arrow” (symbolically as “→”) represents the relation (“use”) which is described in module listing above. That means that the data and variables from

all used modules became global variables or common for the modules which use them. In turn, for all modules used by the main program unit, all data and variables become global or common together with the main program. And symbolically, the relation  $\langle \text{module\_A} \rangle \rightarrow \langle \text{module\_B} \rangle$  in the diagram represents that  $\langle \text{module\_B} \rangle$  uses  $\langle \text{module\_A} \rangle$ , which means that all data from  $\langle \text{module\_A} \rangle$  are global or common for  $\langle \text{module\_B} \rangle$ . Another example  $\langle \text{module\_A} \rangle \rightarrow \langle \text{main} \rangle \leftarrow \langle \text{module\_B} \rangle$  represents that all data and variables from these modules are global for them and for the main program unit. The color for each relation “arrow” represents which module is being used by relative module. This gives a clear visual representation for data communications. To be global means that all variables with the same names occupy the same memory cells in the computer architecture. As was mentioned in previous sections, the module structure of Fortran 90 eliminates the use of “common” statements and provides very flexible and convenient data communications between all program units in the source. Additionally the author would like to point out that the module structure allows inserting or use of other set of modules, for example, created at a later time and for certain different purposes. Thus UTGRID® allows insertion of other modules for different computational domains. While current configuration and functionality of UTGRID® has been specifically adapted and set up to represent components of the UTSI HVEPS facility, UTGRID® can be modified to accept other modules for creation of meshes for other types of flow geometries. Contact the [author of this Thesis](#) for details or assistance in expanding the capabilities of UTGRID®.



**Figure 3.2.1.** Module structure of UTGRID® and functional relations between modules.

### 3.3. The Listing of modification, temporary created files and external subroutines.

This section lists files that have been specifically defined and set up by the author to create mesh configurations for numerical modelling of the HVEPS facility components, such as the number of electric elements in the channel which is set to 6 and the geometry of the straight channel which is defined in the input file of UTGRID®. The purposes and functions of these files are also briefly described. For more details refer to the section 3.5.6.

---

File order	Source list name	Functional description
------------	------------------	------------------------

---

source_11	<b>modify1</b>	This is a geometrical elements modification file that arises at the first stop (STOP1) event when executing UTGRID®. The user must create this file if control parameter l_mod is set to zero in the input namelist (source_9) according to the “ <b>Manual</b> ” in section 3.5 below. This file is a command file for GAMBIT® for the purpose of setting the controllable labels or names for all of the geometrical elements of a mesh, which are: edges, faces and volumes. During execution of UTGRID®, this file will be added to or concatenated with the file ‘temp1.jou’ (see below).
source_12	<b>modify2</b>	This file is also a geometrical elements name modification file at the second stop (STOP2) event when executing

program UTGRID®. The user must create this file if control parameter `l_mod` is set to zero in the input namelist according to the “**Manual**” in section **3.5** below. This is also a command file for GAMBIT® for the purpose of setting the controllable names to all of geometrical elements: edges, faces and volumes. During execution of the UTGRID® this file will be added to or concatenated with the file ‘temp2.jou’ (see below).

source\_13     **modify3\_s**     Also a geometrical elements label or name modification file at the third stop (STOP3) event when executing UTGRID®. The user must create this file if control parameter `l_mod` is set to zero in the input namelist according to the “**Manual**” in section **3.5** below.

This file is also a command file for GAMBIT® for the purpose of setting controllable names to all of geometrical elements: edges, faces and volumes. This file is associated with the straight geometry (slope angles: 90, 90; see also the section **3.5.3**). During execution of UTGRID® this file will be added to or concatenated with the file ‘temp3.jou’ according to the type of the channel (see below).

source\_14     **modify3\_d**     A geometrical elements label or name modifications file at the third stop (STOP3) event when executing UTGRID®. The user must create this file if control parameter `l_mod` is

set to zero in the input namelist according to the “Manual” in section 3.5 below.

This file is also a command file for GAMBIT® for purpose of setting the controllable names to all of geometrical elements: edges, faces and volumes. This file is associated with the diagonal wall channel with the values of the slope angles for the electric elements (insulators and conductors) set according to the input namelist for the channel called diagonal. During execution of the UTGRID® this file will be added to or concatenated with the file ‘temp3.jou’ according to the type of the channel (see below).

source_18	<b>temp1.jou</b>	<p>This is an output file produced <b>at the first stop</b> event (<b>STOP1</b>) when executing the program UTGRID®. This is a <b>geometry</b> file output by GAMBIT™ after the set of names are given for the geometrical elements created by the command file generated by UTGRID®. These names are weakly predictable in general and are also weakly algorithmic for the geometry creation and for further mesh generation process. That is why the names mentioned above have to be re-defined in order to support the geometry creation and further mesh generation algorithms of UTGRID®. After final execution, this file can be deleted or saved depending on purposes of the user. On any new</p>
-----------	------------------	--

run of UTGRID® this file will be re-written in a work directory of the source. This is the journal file for GAMBIT® and must be used at the very first step before the creation of the file ‘modify2’ according to the control parameter l\_mod in the input namelist, see also the “**Manual**” below in the section 3.5.

source\_19      **temp2.jou**      This is an output file produced **at the second stop** event (**STOP2**) when executing the program UTGRID®. This is a **geometry** file which will be added to or concatenated with the file ‘modify1’, which in turn, will be added to or concatenated with the file ‘temp1.jou’ described above. This is a journal input file for GAMBIT® and must be used at the second step before the creation of the file ‘modify2’ according to the control parameter l\_mod in the input namelist, the “**Manual**” below in the section 3.5.

This file is the second set the geometry commands for GAMBIT® generated by UTGRID® during the execution.

source\_20      **temp3.jou**      Also an output file produced **at the third stop** event (**STOP3**) when executing the program UTGRID®. This is a **geometry** file which will be added to or concatenated with the file ‘modify2’, which in turn, will be added to or concatenated with the file ‘temp2.jou’ described above. This is a journal file for GAMBIT® and must be used as

the input at the third step before the creation of the file ‘modify3’ according to the control parameter l\_mod in the input namelist, the “**Manual**” below in the section 3.5.

source\_21      **temp4\_s.jou** This is an output file generated **after the third stop** event (**STOP4**) when executing the program UTGRID®. This is a **geometry** and **mesh** file for the channel with the straight geometry with both straight slope angles (90, 90) for the electric elements (conductors and insulators). This is a journal file for GAMBIT™ and it will be added to or concatenated with the file ‘modify3\_s’, which in turn, will be added to or concatenated with the file ‘temp3.jou’ (see above). This file is created during the last step of execution of the program UTGRID® for this type of the channel.

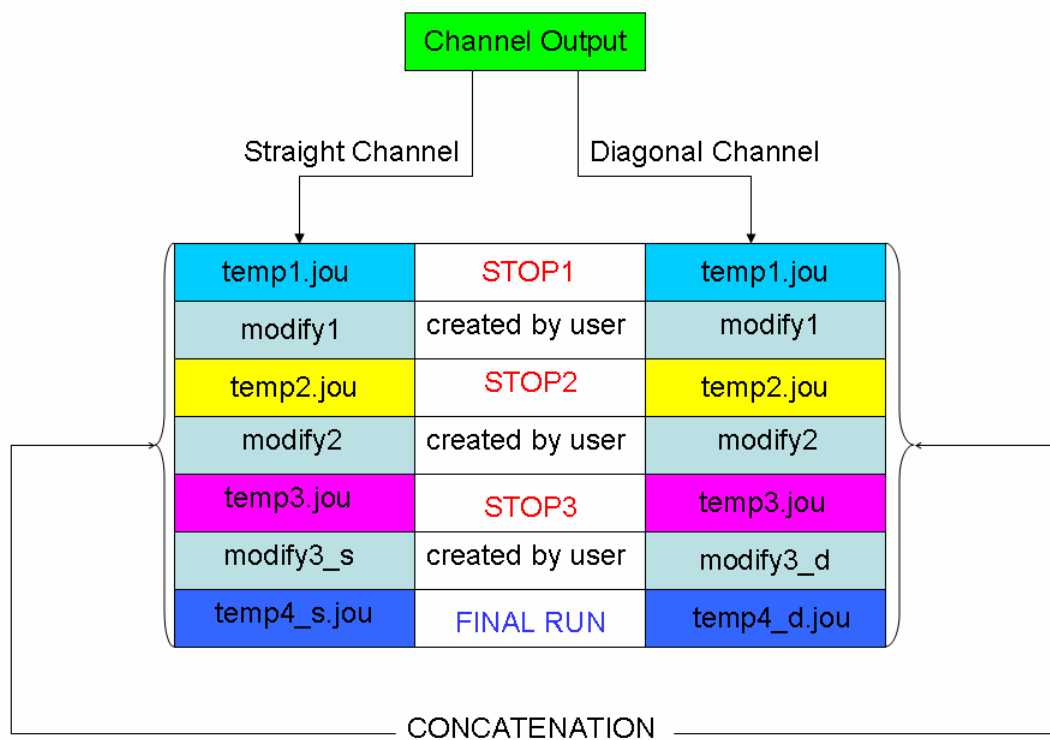
source\_22      **temp4\_d.jou** This an output file generated **after the third stop** event when executing the program UTGRID®. This is a **geometry** and **mesh** file for the channel with diagonal conducting wall geometry with two slope angles for the electric elements (conductors and insulators) according to the input namelist of variables. This is a journal file for GAMBIT™ and it will be added to or concatenated with the file ‘modify3\_d’, which in turn, will be added to or concatenated with the file ‘temp3.jou’ (see above). This file is created during the last step of execution of the program

UTGRID® for this type of the channel (diagonal MHD channel).

source_25	<b>'create1.f'</b>	'modify1' file creation (external subroutine)
source_26	<b>'create2.f'</b>	'modify2' file creation (external subroutine)
source_27	<b>'create3_d'</b>	'modify3_d' file creation (external subroutine)
source_28	<b>'create3_s'</b>	'modify3_s' file creation (external subroutine)
source_29	<b>'geom_prog.f'</b>	geometrical progression solver (external subroutine)

---

The schematic diagram of a final output file for the channel is represented in the **Figure 3.3.1** below in this section. All files described above in this section are concatenated as shown in this figure. This way of concatenation allowed full control of the names which GAMBIT® creates for its final mesh geometry and mesh file generation by the UTGRID®. All files from the listing above have an ASCII format and can be read by any appropriate text editor regardless the extension: \*.jou for the journal files.



**Figure 3.3.1.** Generating the output files when executing UTGRID®

### **3.4. Nozzle input wall or contour profile and generated profiles of its first and second derivatives.**

In constructing a numerical mesh for the converging-diverging supersonic nozzle, to present this component of the HVEPS facility, it was found that the input nozzle wall or profile contour needed smoothing. Generally, such nozzle profiles are described by a relatively small number (typically less than 50-100) of discrete (x, y) points, because this data is difficult to measure. Physical nozzle contours are designed to avoid the generation of flow irregularities and shock waves in the nozzles. Therefore, numerical smoothing of the input wall coordinate data is necessary so that the computed supersonic nozzle flow fields do not contain irregularities or shock waves unless they are actually or physically present.

Thus, to obtain an accurate representation of the HVEPS nozzle contour, a curve-fitting and smoothing routine was applied to the discrete contour data. This procedure ensured that both the first and second derivatives of the numerically generated wall contour varied in a smooth and monotonic manner from inlet to outlet. Specifically, the procedure eliminated jumps, steps, and waves or bumps in the numerical representation of the nozzle contour.

The smoothing procedure used was as following:

1. The profile which was received from the hardware facility manager Mr. James R.

Goodman is represented on **Figure 3.4.1** below. The points of the profile marked as red circles were used as the input or data basic points for smoothing procedure. This file was defined in an Excel format and consisted of only eight (8) real, physical measured contour points of the nozzle.

These points were converted to the SI system of measurement and with simple auxiliary Fortran code they were converted to a 3D representation as coordinate array

$(x, y, 0)$  where:

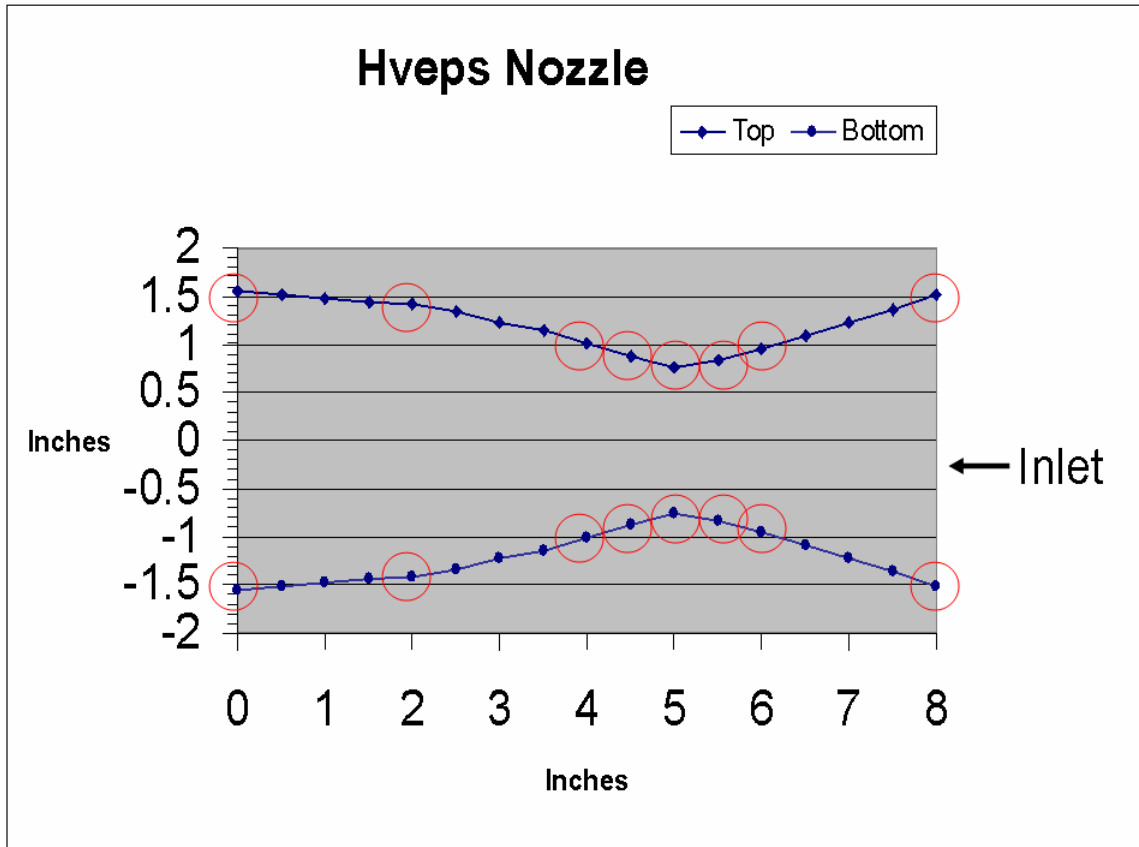
$x = 0., 0.051, 0.0635, 0.0762, 0.0889, 0.1016, 0.1524, 0.2032,$

$y = 0.038608, 0.024384, 0.02109, 0.019444, 0.022276, 0.036119, 0.039497$

and

$z = 0$

respectively.



**Figure 3.4.1.** Measured nozzle wall contour of the real nozzle and basic position points for a smoothing procedure.

2. These converted points were entered manually to GAMBIT™ as input vertices.

The set of commands which were used for vertices creation were:

---

```
vertex create "v_1" coordinates 0 0.038608 0
vertex create "v_2"coordinates 0.051 0.024384 0
vertex create "v_3"coordinates 0.0635 0.02109 0
vertex create "v_4"coordinates 0.0762 0.019444 0
vertex create "v_5"coordinates 0.0889 0.022276 0
vertex create "v_6"coordinates 0.1016 0.036119 0
vertex create "v_7"coordinates 0.1524 0.039497 0
vertex create "v_8"coordinates 0.2032 0.039497 0
```

---

The user can also organize an ASCII journal file for the GAMBIT® with the extension: \*.jou and run it either from the interface or from the command line. For all questions about the format of the commands of the GAMBIT™ refer to the Fluent “Command reference guide” of the [GAMBIT™](#).

3. The commands of GAMBIT™ which were used for the approximation of the nozzle wall contour or the profile are as following:

---

```
edge create "profile" nurbs "v_1" "v_2" "v_3" "v_4" "v_5" "v_6" "v_7" \
v_8" interpolate
```

---

The user can also do it manually using interface of GAMBIT™.

4. It is impossible to export from GAMBIT™ only one meshed edge, therefore it is necessary to create 2D geometry in order to create a 2D mesh. The commands of GAMBIT™ for this purpose are as following:

---

**vertex create “x\_inlet” coordinates 0 0 0**

**vertex create “x\_outlet” coordinate 0.2032 0 0**

**edge create “r\_inlet” nurbs “x\_inlet” “v\_1” interpolate**

**edge create “r\_outlet” nurbs “v\_8” “x\_outlet” interpolate**

**face create “nozzle” wireframe “x\_inlet” “x\_outlet” “r\_inlet” “r\_outlet”**

**face mesh “nozzle” map intervals 500**

**solver select “FLUENT 5/6”**

**export fluent5 “<full location of the file for saving>” nozval**

---

Instead of 500 intervals the user can choose any reasonable number of mesh intervals.

The names which are in quotes can be chosen by the user. Instead of **<full location of the file for saving>** the user must type the location like as follows:

**c:\ <user\_directory>\<user\_file\_name>.msh** in Windows and respectively:

**c/<user\_directory/<user\_file\_name>.msh** in the UNIX-based system.

5. Open TECPLOT™ and import mesh file <user\_file\_name>.msh with the option “load boundary only”. Instead of the extension for the case file \*.cas and data file the user must put <user\_file\_name>.msh. When the boundary is loaded, the user must write it to the ASCII file from the TECPLOT™ file menu.

After that the user must manually edit the saved ASCII file for the nozzle wall boundary.

6. The procedure of editing the boundary ASCII file created by TECPLOT™ was done in the following steps:

- 6.1. Find the very first (**x, y**) couple (for this particular case)

$$\mathbf{x} = \mathbf{0.0}, \mathbf{y} = \mathbf{0.038608}$$

- 6.2. Find the line with (**x, y**) couple with (for this particular case)

$$\mathbf{x} = \mathbf{0.2032}, \mathbf{y} = \mathbf{0.039497}$$

- 6.3. Delete exactly the second line immediately after the line found in **6.1**

- 6.4. Copy all data coordinates between the lines found in 6.1 and the ones in **6.2** to the chosen another empty ASCII file.

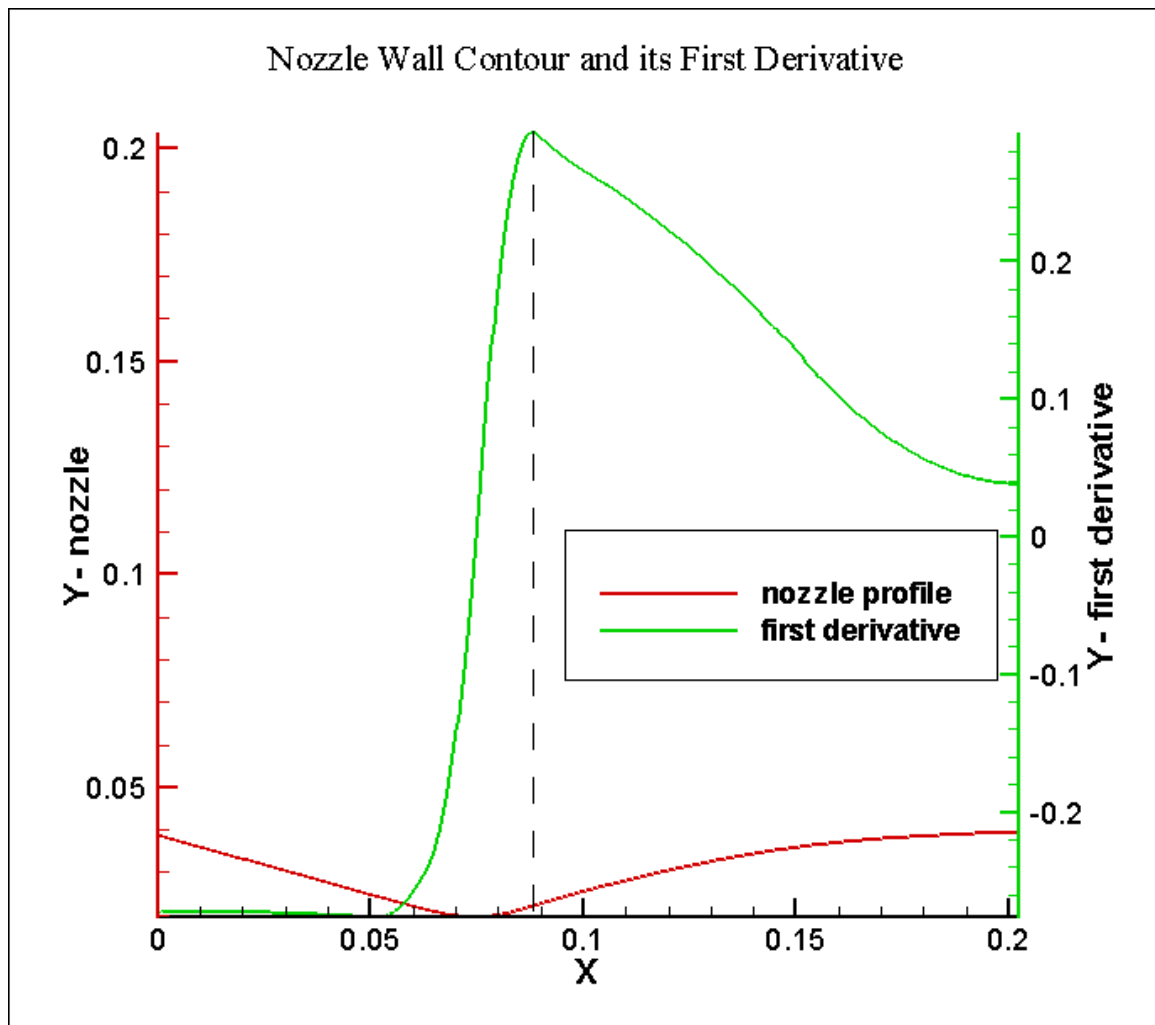
This new file will represent the 2D wall nozzle contour or the profile.

7. A simple auxiliary Fortran code was written by the author to convert this 2D array to a 3D array with Z-components set to zero, like (**x, y, 0**).

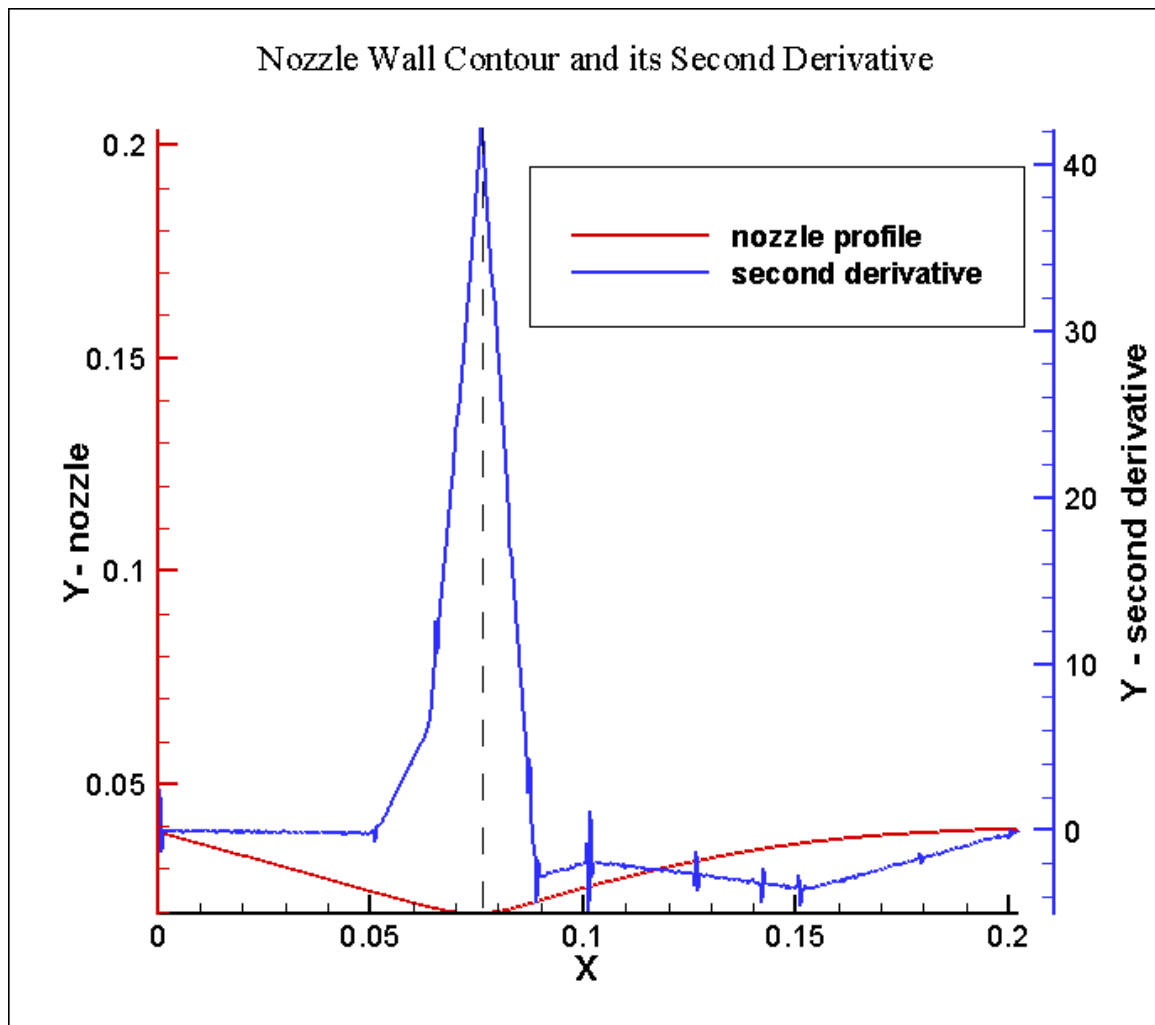
This was the final array used in UTGRID® to represent nozzle coordinates which defined 'noz\_prof.dat' (see 'source\_15'). As was mentioned in section **3.1**, this file can have any extension as ASCII file, but the user must type in the consol terminal this exact name when answering the requested questions during the interactive part of the UTGRID® execution. Also the author would like to suggest that to export command files from GAMBIT™, to use the interface and appropriate file menus to avoid any possible bugs that have been encountered in previous versions of GAMBIT™.

If the user is familiar with the neutral file format, and also with the mesh file format: \*.msh, mesh file can be edited manually without changing the solver from **GENERIC** to **FLUENT 5/6**, or the user could manually edit a mesh file not loading only boundary to **TECPLOT™**.

The appropriate journal file for nozzle profile generation is represented in the file [create\\_profile](#) in the **APPENDIX C**. The plots of the wall nozzle contour and the profile with the first and second derivatives are represented below in **Figure 3.4.2** and in **Figure 3.4.3** respectively.



**Figure 3.4.2.** Generated nozzle wall contour or the profile and its first derivative.



**Figure 3.4.3.** Generated nozzle wall contour or the profile and its second derivative.

Finally it should be pointed out that UTGRID® accepts any number of nozzle profile array points; for example, in the present study 500 grid points were used to represent the length of the nozzle. A greater array of points would be expected to give better quality representation for the wall contour of the nozzle.

If in the ‘**control**’ section, in the input namelist, parameter **l\_der** is set to **1**, the profiles for the first and the second derivatives are generated as useful auxiliary information. Certainly, an array with a greater number of wall coordinates for the wall nozzle contour would also give better quality for the derivatives which may improve the convergence of the solution during computations.

The derivatives were calculated with second order numerical schemes. The number of points in the nozzle coordinates does not affect on the mesh generation process since meshing is done based on the control set by UTGRID®, and then by the GAMBIT™ “continuously” (in the sense of the precision of the approximation by GAMBIT™). All intermediate vertices are deleted, except for the inlet and outlet vertices, and then meshed during the execution of the algorithm in UTGRID®.

When parameter **l\_der** is set to one, the user must type in the consol terminal the desired location of the output files (if they are in another directory), or just their names if they are to be created in the work directory. For any possible questions about the smoothing procedure for any XY - curve contact the [author of this Theses.](#)

### **3.5. Manual.**

The following is an outline of a user's manual for the suite of codes that form the automated meshing procedure developed by the author. Where specific files are named or given or referenced, they pertain to the specific test facility components of the HVEPS project: combustor, nozzle, nozzle extension as an adaptor (ceramic ring), and either constant area conductivity channel, or DCW channel.

#### **3.5.1 The work directory.**

The work directory must contain the following files:

- 'UTGRID.f'
- 'utgrid\_input.f'
- 'chan\_calc.f'
- 'chan\_grid.f'
- 'noz\_calc.f'
- 'noz\_grid.f'
- 'litteral.f'
- 'sprint.f'
- 'modify1' (if created by the user)
- 'modify2' (if created by the user)
- 'modify3\_s' (if created by the user)
- 'modify3\_d' (if created by the user)
- 'create1.f'

- ‘create2.f’
- ‘create3\_s.f’
- ‘create3\_d.f’
- ‘Makefile’
- ‘go-utgrid’
- ‘geom\_prog.f’

The source files described in the list above are given in detail in the **APPENDIX D**. The examples of the created modification files as well as temporary created files are included in the **APPENDIX D**, including the channel with the amount of the insulators of the channel set to 6. The work directory must be created by the user. The directives for the creation of modification files will be described below in section **3.5.5** of this “**Manual**”. An input file given as an example, ‘input\_45\_60’, can be located in the work directory; if not in the work directory the user must type its location on the consol terminal during the interactive procedure of the execution of UTGRID®.

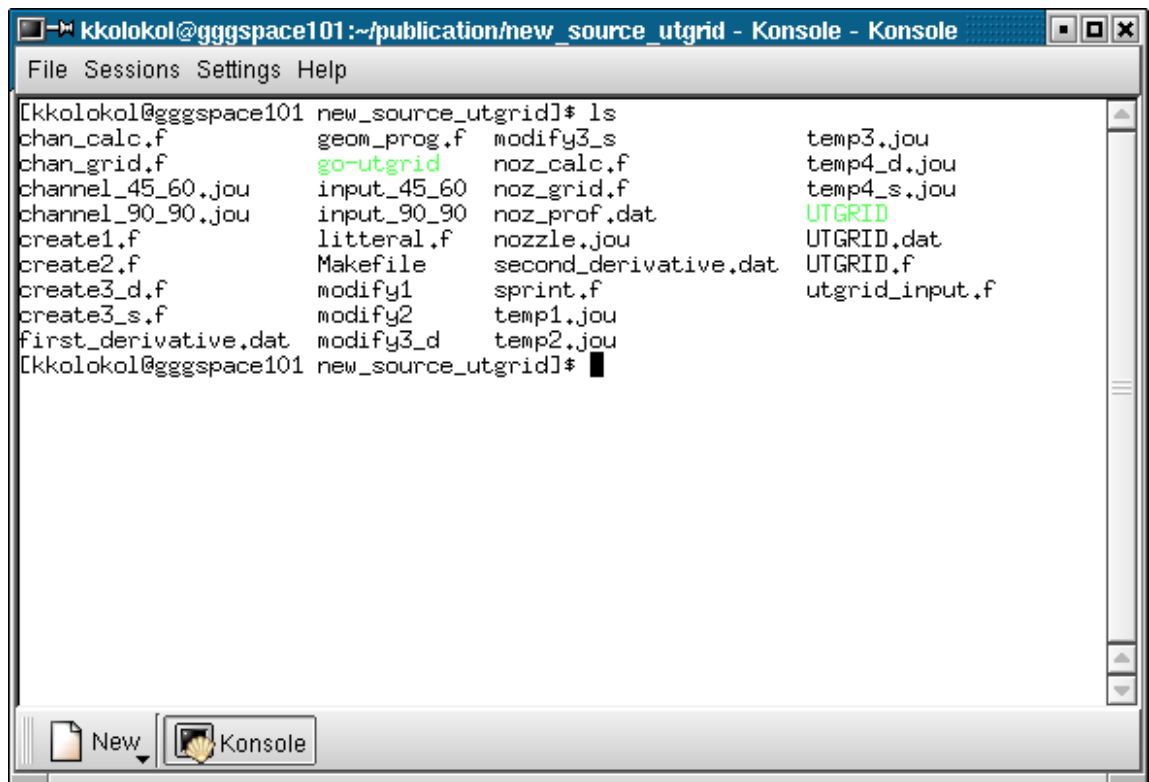
According to the section ‘**control**’ (control variables) of the input file namelist, and the answers typed by the user on the consol terminal during the interactive process of the execution of the UTGRID®, the following files will be created in the work directory:

- ‘temp1.jou’
- ‘temp2.jou’
- ‘temp3.jou’
- ‘temp4\_s.jou’
- ‘temp4\_d.jou’

- ‘first\_derivative.dat’
- ‘second\_derivative.dat’
- ‘UTGRID.dat’
- ‘channel\_45\_60.jou’
- ‘nozzle.jou’
- ‘modify1’ (if not created by the user previously)
- ‘modify2’ (if not created by the user previously)
- ‘modify3\_s’ (if not created by the user previously)
- ‘modify3\_d’ (if not created by the user previously)

As was mentioned before in the section **3.1** of the current chapter, the last five files represented in the list above can have any names and locations defined by the user. However, in this case, their locations must be typed on the consol terminal when requested if they are located not in the work directory. The procedure of the execution of UTGRID® will be discussed later in more details in sections **3.4** and **3.5** as well as the output, which will be described in section **3.6**. The work directory of the author is illustrated in **Figure 3.5.0** below in this section. The author would suggest locating all files with user-definable names in one work directory which will certainly simplify the execution procedure.

Note, if a modification file (or one of them) is not located in the work directory, and the corresponding parameter **l\_mod** from the section ‘**control**’ in the input namelist is set to **1**, an appropriate error message will be printed on the consol terminal.



The image shows a terminal window titled "Konsol - Konsol" with a menu bar containing "File", "Sessions", "Settings", and "Help". The terminal text shows a user at the prompt "kkolokol@gggspace101" in the directory "new\_source\_utgrid" running the command "ls". The output lists 24 files in a three-column format. The files include source code (.f), data files (.dat), and job files (.jou). Notable files include "UTGRID.dat" and "UTGRID.f" in green text, and "go-utgrid" in green text. The prompt returns to "kkolokol@gggspace101 new\_source\_utgrid\$". At the bottom, there are buttons for "New" and "Konsol".

```
kkolokol@gggspace101 new_source_utgrid]$ ls
chan_calc.f          geom_prog.f      modify3_s        temp3.jou
chan_grid.f          go-utgrid        noz_calc.f       temp4_d.jou
channel_45_60.jou    input_45_60      noz_grid.f       temp4_s.jou
channel_90_90.jou    input_90_90      noz_prof.dat     UTGRID
create1.f            litteral.f        nozzle.jou       UTGRID.dat
create2.f            Makefile          second_derivative.dat UTGRID.f
create3_d.f          modify1           sprint.f          utgrid_input.f
create3_s.f          modify2           temp1.jou
first_derivative.dat modify3_d          temp2.jou
kkolokol@gggspace101 new_source_utgrid]$
```

**Figure 3.5.0.** The work directory of the author for UTGRID®.

### 3.5.2. Compilation and installation of UTGRID®.

The source code of UTGRID® must be compiled and installed, which means that an executable file must be created. For the compilation procedure the user must use a Fortran 90 compiler. The code UTGRID® was created and tested with certain Fortran 90 compiler: [Portland Group Fortran 90 Compiler](#) [7]. The user can also check the compatibility with other compilers and the conformance to standards [8-16]. The source code of UTGRID® was adapted to the UNIX-like operating system denoted **Linux: Red Hat 7.2 Professional** (Kernel 2.4.7-10 on i686). It was not tested with **Microsoft Windows** Fortran 90 compilers; however, it can be run with a Portland Group Fortran 90 Compiler in Windows with the **CYGWIN** installed. It is implied for the installation in operating system Windows for this compiler and described in the guide [7].

The consequent list of the commands provides the compilation and installation:

- **make (gmake)**
- **go-utgrid**

Those commands will be executed if the appropriate source files (except for input and output files) are located in the work directory created by the user. After the execution of the commands above the source files will be compiled and an executable file will be created. The work directory will be cleaned of all objective files.

It is also acceptable to not locate the file ‘Makefile’ and the script file ‘go-utgrid’ in the work directory. In this case the user must run the following commands:

- **make -f <location of the ‘Makefile’> or gmake -f <location of the ‘Makefile’>**
- **<the path of the file ‘go-utgrid’>**

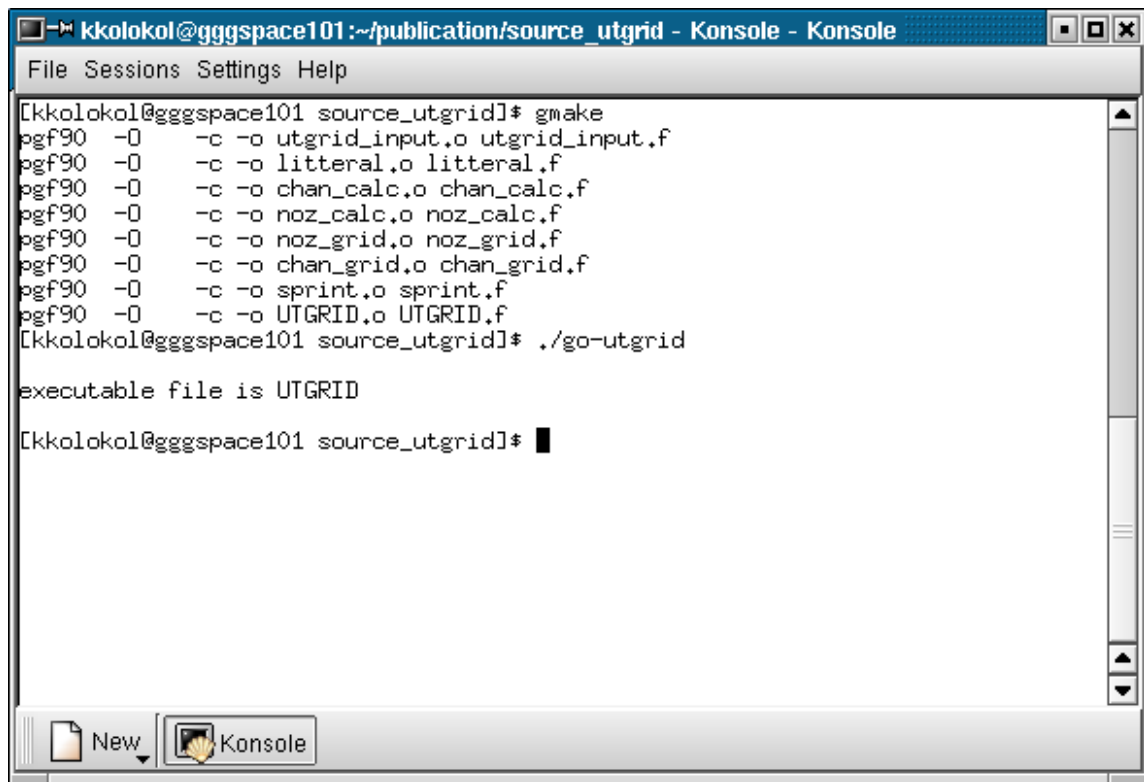
The author would suggest locating of those files in the work directory for more simplicity

and convenience. The compilation procedure is illustrated on **Figure 3.5.1** below in this section. In the example considered, the executable file is named 'UTGRID' which is typed on the consol terminal by the UTGRID®. The path of the file 'go-utgrid' in this example is:

**~/publication/new\_source\_utgrid/go-utgrid.**

'Makefile' and 'go-utgrid' are located in the work directory: **new\_source\_utgrid.**

Normal termination of the compilation and installation process will not give any error messages and must look like it is represented on **Figure 3.5.1**, depending on operating system and appropriate graphical interface (if X-server is running).



The image shows a terminal window titled "Kkonolokol@gggospace101:~/publication/source\_utgrid - Konsole - Konsole". The window contains the following text:

```
[kkolokol@gggospace101 source_utgrid]$ gmake
pgf90 -O      -c -o utgrid_input.o utgrid_input.f
pgf90 -O      -c -o litteral.o litteral.f
pgf90 -O      -c -o chan_calc.o chan_calc.f
pgf90 -O      -c -o noz_calc.o noz_calc.f
pgf90 -O      -c -o noz_grid.o noz_grid.f
pgf90 -O      -c -o chan_grid.o chan_grid.f
pgf90 -O      -c -o sprint.o sprint.f
pgf90 -O      -c -o UTGRID.o UTGRID.f
[kkolokol@gggospace101 source_utgrid]$ ./go-utgrid

executable file is UTGRID

[kkolokol@gggospace101 source_utgrid]$
```

The terminal window has a menu bar with "File", "Sessions", "Settings", and "Help". At the bottom, there are buttons for "New" and "Konsole".

**Figure 3.5.1.** The compilation and installation of UTGRID®.

### 3.5.3. Input file.

The format of the input file must be as it is represented in ‘source\_9’ in **APPENDIX D**.

The user can put numerical data for all variables in free form for real numbers. The dimension of all variables is SI metric system. For explanation of the variables in the input namelist, the input file is commented at the bottom of the namelist. The user must not change anything except numerical data. Because of the large number of figures needed to clarify the nature and identities of the control parameters, all figures in the range: **Figure 3.5.6 – Figure 3.5.121** in this section, have been put into an appendix, **APPENDIX A1**, pages 157 - 284.

The definitions of the variables in the input namelist are provided below, along with a set of figures in the range: **Figure 3.5.6 – Figure 3.5.9**, located also in **APPENDIX A1** to clarify their meaning.

- **ch\_diam**
  - **d1** – internal inlet diameter of the channel;
  - **d12** – external inlet diameter of the channel for the solid zone;
  - **d3** – the diameter of the air domain.

The **Figure 3.5.6** illustrates the variables in the list above.

- **ch\_len**
  - **chlin** – the distance between the inlet plane and the closest edge of the first insulator on the solid surface of the channel in the ZY – plane;
  - **chlns1** – the distance between the edges of each insulator in the ZY – plane;
  - **chlcd1** – the distance between the edges of each conductor in the ZY – plane;
  - **chlout** – the distance between the last edge of the outlet conductor and the

outlet in the ZY – plane.

The **Figure 3.5.7** illustrates the variables in this list. For the secondary slope angle **ange** for the electric elements, which will be discussed later in this section, the variables from the list above are relative to the ZY –plane; for the “**shift**” plane they will differ and are calculated. The verbal notation “**shift**” will be introduced and described later in section **3.5.6**. This approach was chosen for convenience, and the thickness of insulators as well as conductors can be easily calculated if needed. It is possible that in further versions of UTGRID® the thickness of the electric elements will be inputted and the axial distance on the solid surface will be calculated. It will depend on the particular needs for numerical modeling.

- **ch\_rd**
  - **prate** – the multiplier for the boundary layer of the channel increase which means, how much thicker the user would like to input boundary layer. In computation algorithm of the UTGRID® it is used for the boundary layer: **delp1\*prate**. This parameter can be very useful for numerical modeling of the real experiment. If this parameter is set to 1 the computed boundary layer thickness is **delp1**;
  - **delp1** – the inlet thickness of the plasma boundary layer of the channel;
  - **delp2** – the outlet thickness of the plasma boundary layer of the channel;
  - **axar** – the axial aspect ratio which is the denominator of the Geometrical Progression used for the axial mesh generation distribution;
  - **bl1** – the aspect ration for the plasma boundary layer distribution;
  - **bl2** – the aspect ration for the radial stretch mesh distribution for the solid

zones (conductors and insulators);

- **bl3** – the aspect ratio for the radial stretch mesh distribution for the air zone;
- **axprec** – the first row of the Geometrical Progression used for the axial stretch mesh distribution for all plasma solid and air zones. This parameter is called “axial precision” and represents the minimal step of the mesh distribution in axial flow direction

The **Figure 3.5.10** illustrates the variables: **prate**, **delp1**, **delp2**, **bl1**, **bl2** and **bl3**. The **Figure 3.5.11** illustrates the axial mesh distribution for the solid and plasma zones along the edge located on the solid and plasma faces in the ZY – plane. The **Figure 3.5.12** illustrates the axial mesh distribution for the air zone along the air edge located on the faces of the air zone in the ZY – plane.

- **ch\_ang**
  - **anga** - the angle, called “primary slope angle”, between the bigger axis of the ellipse, defined by internal electric element, and the axial Z – axis;
  - **angb** – the angle of the divergence (if positive) or convergence (if negative) of the channel;
  - **ange** – the angle, called “secondary slope angle”, between the smaller axis of the ellipse, defined by the internal electric element, and the axial Z – axis.

The **Figure 3.5.8** illustrates the angular variables for the channel in this list. If the user would like to input only one slope angle he must set the variable **ange** to 90. The Straight channel is defined as soon as **anga** and **ange** are set to 90. For the double slope channel the variables in the section ‘**ch\_len**’ of the input namelist are defined for the appropriate edges in the ZY – plane; it is very important issue.

The following section and variables in the section are included into the namelist of the input file for the nozzle:

- **noz\_dat**
  - **delth** – the radial rate of the throat which defines the length of the stretched area, called “throat corridor”, and its coordinates from the left and from the right of the throat point. The coordinates (left and right) are calculated accordingly to the rule:  $yprof = yth/delth$ , where **yprof** is the Y – coordinate of the throat point (minimal radius);
  - **axprec1** - the first row of the Geometrical Progression used for the grid generation distribution in the axially stretched areas in the throat corridor of the nozzle;
  - **axprec2** - the first row of the Geometrical Progression used for the grid generation distribution in the axially stretched area for the discontinuity point, called “sharp angle”, where combustor is connected to the nozzle;
  - **axprec3** - the first row of the Geometrical Progression used for the grid generation distribution in the axially for the discontinuity point in the combustor where cylindrical and conical part are connected;
  - **arnoz** - aspect ratio for the axially stretched areas of the nozzle zone used as a denominator of the Geometrical Progression;
  - **arcomb** - aspect ratio for the axially stretched areas of the combustor as denominator of the Geometrical Progression;
  - **archan** - aspect ratio for the channel extension axially stretched mesh distribution;

- **fnoz** - the first row of the nozzle boundary layer mesh distribution as the first row of the Geometrical Progression;
- **blar** - aspect ratio for the nozzle boundary layer;
- **throat** - the rate of the boundary layer thickness to the **rth**;
- **rcomb** - the radius of the cylindrical part of the combustor;
- **cyl\_c** - the length of the cylindrical part of the combustor;
- **cone\_c** - the length of the conical part of the combustor;

The **Figure 3.5.9** illustrates the input namelist variables in the above list above. The extent of the uniform mesh distribution is represented by the blue color arrows in this figure. The radial mesh distributions are uniform for combustor, nozzle, adaptor and the channel extension parts.

The last and the first rows of the axial stretched areas are automatically adjusted to be respectively equal in the mesh distribution from the left: the last row of previous mesh is equal to the first row for the next area meshed, etc. The axial length of the cell in the axial uniform mesh distribution area is equal, respectively to: the minimal last row between the area of the “sharp angle” (connection combustor – nozzle) and the starting point of the throat corridor; the minimal last row between the throat corridor and the last row of the mesh distribution in the area which stretches the discontinuity point, where adaptor is attached to the nozzle. The inlet and outlet are also being stretched in the channel extension.

The last row of radial boundary later is equal to the radial mesh in uniform mesh distribution radial areas below the boundary layer.

The author hopes that the description of the mesh generation process for the nozzle is

clear for understanding and does not bring any inconveniences to the user.

The following sections are included into the namelist of the input file for the control of the execution:

- **l\_der** – the control parameter for the calculation of the first and second derivatives of the nozzle wall contour to estimate its quality in its approximation. The role of this parameter is explained in the section **3.4**;
- **l\_print** – the control parameter for the generation of the ASCII file which contains some important printed data about the channel and for the nozzle as well (reserved);
- **l\_ch** – the control parameter for the grid generation for the full channel domain;
- **l\_chcalc** – the control parameter for the calculations of all required geometrical parameters and mesh data needed for the grid generation process for the channel and nozzle as well;
- **l\_noz** – the control parameter for the nozzle grid generation process;
- **l\_adapt** – the control parameter for the grid generation process which includes the nozzle extension or adaptor (the ceramic ring in the real experiment) attached to the nozzle;
- **l\_ext** – the control parameter for the grid generation process for channel extension attached to the adaptor;
- **l\_mod** – the control parameter for creation of the modification files done by the interactive routine procedure of UTGRID®, for the arbitrary amount of the electric elements in the channel, which will be discussed in details later in the

section **3.5.6.**

All control parameters in the list above represent numerically-logical parameters which can be assigned a value only 0 or 1. They define the appropriate functions of the UTGRID® to be executed.

The functions of UTGRID® can be described as the following logical predicates:

- **Predicate 1**

Calculate the first and second derivatives of the nozzle wall contour and write them to an organized ASCII file (each to a separate file). These files can be located in some other directory chosen by the user and typed on the consol terminal.

- **Predicate 2**

Print some geometrical and mesh data (reserved) of the channel and nozzle into an organized ASCII file which must be chosen by the user and its name (if in the work directory) or location (if not in the work directory) must be typed on the consol terminal.

- **Predicate 3**

Generate the journal file for GAMBIT™ which completely defines the mesh generation process for the channel, and write it into an organized ASCII file with the extension \*.jou. The user must type either the name of this file in the work directory or its chosen location.

- **Predicate 4**

Generate the journal file for GAMBIT™ which completely defines the mesh generation process for the combustor and nozzle, and write it into an organized ASCII file with the extension \*.jou. The user must type either the name of this file in the work directory or its chosen location.

- **Predicate 5**

Generate the journal file for GAMBIT™ which completely defines the mesh generation process for the adaptor attached to the nozzle, and write it into an organized ASCII file with the extension \*.jou. The user must type either the name of this file in the work directory or its chosen location.

- **Predicate 6**

Generate the journal file for GAMBIT™ which completely defines the mesh generation process for the channel extension attached to the adaptor, and write it into an organized ASCII file with the extension \*.jou. The user must type either the name of this file in the work directory or its chosen location.

- **Predicate 7**

Calculate all geometrical and mesh data for the channel.

- **Predicate 8**

Create the modification files and write each of them into separate organized ASCII files in the work directory. The extension for these files is not needed.

- **Predicate 9**

Terminate the execution of UTGRID® and provide the user gets a final message on the consol terminal which represents the end of run of the UTGRID®.

- **Predicate 10**

Terminate the execution of UTGRID® and provide the user a message on the consol terminal that the modification files will not be created, and a final message, which will represents the end of run of the UTGRID®.

- **Predicate 11**

Terminate the execution and provide the user a message on the consol terminal that manifests an error message and asks the user to repeat the run.

▪ **End of list of predicates**

The logical outcomes of activating different options in UTGRID® are shown symbolically below. The symbol ‘ $\wedge$ ’ is used to mean logical ‘and’, the symbol ‘ $\Rightarrow$ ’ is used to denote logical implication. Moreover, the results below indicate the value of the control parameters in the data input file (source\_9) required to achieve each option.

The logical variable **answer** below represents the answers which the user must type on the consol terminal if: **l\_mod = 0** and the logical variable **ans** below represents the answers which the user must type on the consol terminal if: **l\_mod = 1**.

The following logical implications are available for the execution of the UTGRID®:

L.1:  $((l\_der = 1) \wedge \text{Predicate 4}) \Rightarrow \text{Predicate 1}$

L.2:  $((l\_print = 1) \wedge \text{Predicate 3} \wedge \text{Predicate 4}) \Rightarrow \text{Predicate 2}$

L.3:  $(\text{Predicate 7} \wedge (l\_ch = 1) \wedge (l\_mod = 1)) \Rightarrow \text{Predicate 3}$

L.4:  $(\text{Predicate 7} \wedge (l\_ch = 1) \wedge (l\_mod = 0) \wedge \text{Predicate 8}) \Rightarrow \text{Predicate 3}$

L.5:  $(\text{Predicate 7} \wedge (l\_noz = 1)) \Rightarrow \text{Predicate 4}$

L.6:  $(\text{Predicate 4} \wedge (l\_adapt = 1)) \Rightarrow \text{Predicate 5}$

L.7:  $(\text{Predicate 5} \wedge (l\_ext = 1)) \Rightarrow \text{Predicate 6}$

L.8:  $(l\_chcalc = 1) \Rightarrow \text{Predicate 7}$

L.9:  $((l\_mod = 1) \wedge \text{Predicate 7} \wedge (l\_ch = 1) \wedge (ans = 'yes')) \Rightarrow \text{Predicate 3}$

L.10:  $((l\_mod = 1) \wedge \text{Predicate 7} \wedge (l\_ch = 1) \wedge (ans = 'no')) \Rightarrow \text{Predicate 9}$

L.11:  $((l\_mod = 0) \wedge \text{Predicate 7} \wedge (l\_ch = 1) \wedge (answer = 'yes')) \Rightarrow \text{Preicate 8}$

L.12:  $((l\_mod = 0) \wedge \text{Predicate 7} \wedge (l\_ch = 1) \wedge (answer = 'no'))$

⇒ **Predicate 10**

L.13:  $((\text{I\_der} = 0) \wedge (\text{I\_print} = 0) \wedge (\text{I\_ch} = 0) \wedge (\text{I\_chcalc} = 0) \wedge (\text{I\_noz} = 0))$

⇒ **Predicate 9**

L.14:  $((\text{I\_ch} = 0) \wedge (\text{I\_noz} = 0)) \Rightarrow \text{Predicate 9}$

All other possible and meaningful logical predicates can be formally calculated from the kernel: L1-L14. This kernel represents the functions of the UTGRID® which can be performed during its execution process. The author hopes that the list of logical implications above is clear to the user and would be happy to provide assistance in case of any user-encountered questions.

#### **3.5.4 The execution of UTGRID® with I\_mod, I\_ch and I\_ch\_calc set to 1.**

If parameter **I\_mod** is set to 1 the execution of UTGRID® will require the user to type on the consol terminal interactively the names or locations of the input file and the output files, and other control information as shown in **Figure 3.5.2-1** and **Figure 3.5.2-2**. The names and/or locations of the files must be entered respectively, depending on whether they are in the work directory, or another directory, chosen by the user.

The second section in the terminal output represents options selectable from the list of the logical implications in section **3.5.3**, and the number of grid points on the axial edges of the solid, the air and the plasma faces in the ZY – plane. This is very important indicator which allows testing of the compiler and its optimization (**O**, **O1**, **O2**, **O3** or **O4**). Currently, in ‘Makefile’, optimization **O** is used. Certain changes in this file allows the use of double precision compilation and then execution. Those possible options are: **-r8**, **-i8**, which will define respectively double precision for real or integer numbers.

```
File Sessions Settings Help

[kkolokol@gggospace101 new_source_utgrid]$ ./UTGRID
> please, enter input file name:
input_45_60
> please, enter print output file name:
UTGRID.dat
> please, enter channel output file name:
channel_45_60.jou
> please, enter nozzle profile file name:
noz_prof.dat
> please, enter nozzle output file name:
nozzle.jou
> please, enter first derivative file name:
> of nozzle profile output file name:
first_derivative.dat
> please, enter second derivative file name:
> of nozzle profile output file name:
second_derivative.dat

-----
> for "basic" slope anga = 45.00000000,
  you can see the number of grids on the axial edges of conductors
  and insulators respectively: n_c(1,...,n), n_ins(1,...,n):
  n_con(0) = 51
[kkolokol@gggospace101 new_source_utgrid]$ ./UTGRID
> please, enter input file name:
input_90_90
> please, enter print output file name:
UTGRID.dat
> please, enter channel output file name:
channel_90_90.jou
> please, enter nozzle profile file name:
noz_prof.dat
> please, enter nozzle output file name:
nozzle.jou
> please, enter first derivative file name:
> of nozzle profile output file name:
first_derivative.dat
> please, enter second derivative file name:
> of nozzle profile output file name:
second_derivative.dat

-----
> for "basic" slope anga = 90.00000000,
  you can see the number of grids on the axial edges of conductors
  and insulators respectively: n_c(1,...,n), n_ins(1,...,n):
  n_con(0) = 43
  n_ins(1) = 19
  n_con(2) = 35
  n_ins(3) = 19
  n_con(4) = 35
```

input and output file names in the work directory

indicator

**Figure 3.5.2-1.** The first and second sections of the consol output from execution of UTGRID for the straight channel, and the top of the important indicator for the axial grid points.

```
kkolokol@gggspace101:~/publication/new_source_utgrid - Konsole - Konsole
File Sessions Settings Help

n_con(0) = 43
n_ins(1) = 19
n_con(2) = 35
n_ins(3) = 19
n_con(4) = 35
n_ins(5) = 19
n_con(6) = 35
n_ins(7) = 19
n_con(8) = 35
n_ins(9) = 19
n_con(10) = 35
n_ins(11) = 19
n_con(12) = 47

-----

CONTROL RUN1
> have you prepared file modify1?(yes/no):
yes
CONTROL STOP1

-----

CONTROL RUN2
> have you prepared file modify2?(yes/no):
yes
CONTROL STOP2

-----

CONTROL RUN3
> you run STRAIGHT channel,
> have you prepared file modify3_s?(yes/no):
yes
CONTROL STOP3

-----

CONTROL RUN4
CONTROL END OF JOB

-----

> you can delete temporarily created files: temp*.* using command:
rm -f temp*

> UTGRID GENERATOR PERFORMED ALL ASSIGNED ACTIONS
> EXIT UTGRID

[kkolokol@gggspace101 new_source_utgrid]$
```

indicator

First control section

Second control section

Third control section

Final control section

Final message

**Figure 3.5.2-2.** The table of the axial grid points, the mesh generation sections, the control sections, exit the execution and final messages on the consol terminal for the straight channel.

The next three sections in **Figure 3.5.3** indicate an additional control for the grid generation process for the channel. The answer ‘yes’ confirms the existence of the modification files and their previous creation, and their location in the work directory as well. If the user will type ‘no’, the execution of the program will be terminated accordingly:

- First section (the file ‘temp1.jou’ is created):
  - ‘no’ – the execution is terminated with the final message represented on the consol terminal is indicated in **Figure 3.5.3** and the consol shown in **Figure 3.5.5** at the very bottom. The file ‘temp1.jou’ will be written into the output file of the channel;
  - ‘yes’ – the execution will be continued. The file ‘modify1’ is added into the channel output file. At this step the concatenation of the files: ‘temp1.jou’ and ‘modify1’ is written into the channel output file;
- Second section (the file ‘temp2.jou’ is created):
  - ‘no’ – the execution is terminated with the final message discussed above. The content of the channel output file at this step is the previous concatenation of the files. The file ‘temp2.jou’ is added into the channel output file, the content of which is now the concatenation of the files:  
‘temp1.jou’, ‘modify1’ and ‘temp2.jou’;
  - ‘yes’ – the execution will be continued. The file ‘modify2’ will be added into the channel output file, the content of which at this step is the concatenation of the files: ‘temp1.jou’, ‘modify1’, ‘temp2.jou’ and ‘modify2’;
- Third section (the file ‘temp3.jou’ is created):

- ‘**no**’ – the execution is terminated with the final message printed on the consol terminal. The file ‘temp3.jou’ is added into the channel output file, the content of which at this step is the concatenation of the files:  
‘temp1.jou’, ‘modify1’, ‘temp2.jou’ and ‘modify2’;
- ‘**yes**’ – the execution will be continued. The file ‘modify3\_d’ or ‘modify3\_s’ (respectively to the case for the channel: straight or diagonal) will be added into the channel output file, the content of which at this step is the concatenation of the files:  
‘temp1.jou’, ‘modify1’, ‘temp2.jou’, ‘modify2’, ‘temp3.jou’ and ‘modify3\_s’ or ‘modify3\_d’;
- Fourth section (the file ‘temp4\_s.jou’ or ‘temp4\_d.jou’ is created).  
This file will be added into the channel output file, the content of which at this step is the final concatenation of the files:  
‘temp1.jou’, ‘modify2’, ‘temp3.jou’, ‘modify3\_s’ or ‘modify3\_d’ and ‘temp4\_s.jou’ or ‘temp4\_d.jou’ respectively.

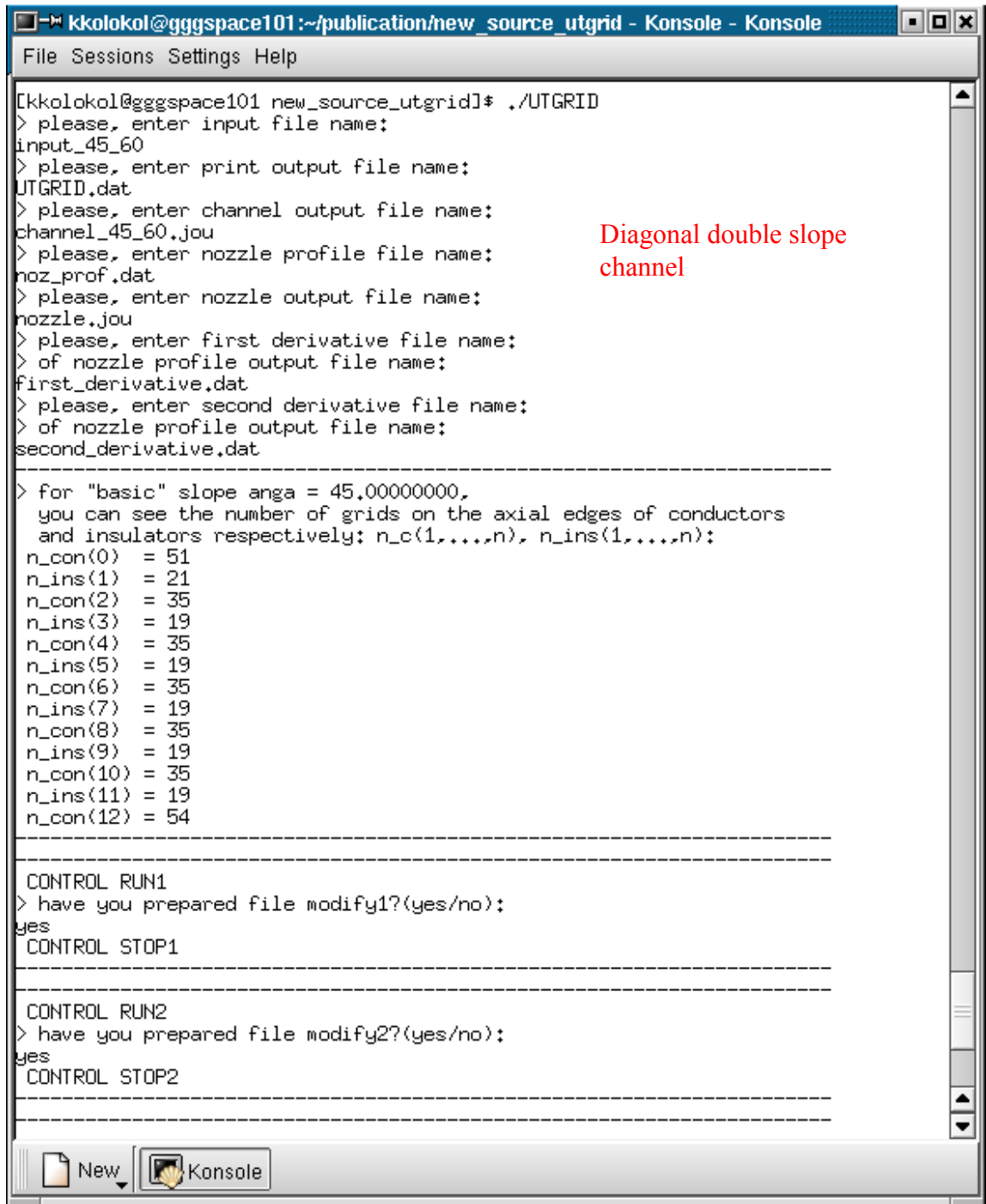
The user can delete temporary created files manually and the printed message on the consol terminal informs the user about this action, if they are not needed any more in the work directory. Their status defines them as ‘rewritten’ for any next execution.

However, having had experience with the [Portland Fortan 90 Compiler](#) [16], the author would suggest deleting all temporary created files manually before the next execution is performed. Contact the author for the further comments with this concern.

The user will also be informed about the type of the channel being considered: straight or diagonal. This message is illustrated on the consol output screen shown in **Figure 3.5.2-2**

above for the straight channel, and in **Figure 3.5.3-2** below for the diagonal channel. The list of control sections is illustrated in **Figure 3.5.2-2** and in **Figure 3.5.3-2**, and the final message as well.

For nozzle grid generation the user the user will not get any messages on the consol terminal except for those associated with **Predicate 9**.



```
kkolokol@gggspace101:~/publication/new_source_utgrid - Konsole - Konsole
File Sessions Settings Help

[kkolokol@gggspace101 new_source_utgrid]$ ./UTGRID
> please, enter input file name:
input_45_60
> please, enter print output file name:
UTGRID.dat
> please, enter channel output file name:
channel_45_60.jou
> please, enter nozzle profile file name:
noz_prof.dat
> please, enter nozzle output file name:
nozzle.jou
> please, enter first derivative file name:
> of nozzle profile output file name:
first_derivative.dat
> please, enter second derivative file name:
> of nozzle profile output file name:
second_derivative.dat

-----
> for "basic" slope anga = 45.00000000,
  you can see the number of grids on the axial edges of conductors
  and insulators respectively: n_c(1,...,n), n_ins(1,...,n):
n_con(0) = 51
n_ins(1) = 21
n_con(2) = 35
n_ins(3) = 19
n_con(4) = 35
n_ins(5) = 19
n_con(6) = 35
n_ins(7) = 19
n_con(8) = 35
n_ins(9) = 19
n_con(10) = 35
n_ins(11) = 19
n_con(12) = 54

-----

CONTROL RUN1
> have you prepared file modify1?(yes/no):
yes
CONTROL STOP1

-----

CONTROL RUN2
> have you prepared file modify2?(yes/no):
yes
CONTROL STOP2

-----

New Konsole
```

**Figure 3.5.3-1.** The first and second sections of the consol output for execution of UTGRID for the diagonal double slope channel, and the top of the important indicator for the axial grid points.

```
kkolokol@gggspace101:~/publication/new_source_utgrid - Konsole - Konsole
File Sessions Settings Help

n_con(0) = 51
n_ins(1) = 21
n_con(2) = 35
n_ins(3) = 19
n_con(4) = 35
n_ins(5) = 19
n_con(6) = 35
n_ins(7) = 19
n_con(8) = 35
n_ins(9) = 19
n_con(10) = 35
n_ins(11) = 19
n_con(12) = 54

-----

CONTROL RUN1
> have you prepared file modify1?(yes/no):
yes
CONTROL STOP1

-----

CONTROL RUN2
> have you prepared file modify2?(yes/no):
yes
CONTROL STOP2

-----

CONTROL RUN3
> you run DIAGONAL channel,
> have you prepared file modify3_d?(yes/no):
yes
CONTROL STOP3

-----

CONTROL RUN4
CONTROL END OF JOB

-----

> you can delete temporarily created files: temp*.jou using command:
rm -f temp*

> UTGRID GENERATOR PERFORMED ALL ASSIGNED ACTIONS
> EXIT UTGRID

[kkolokol@gggspace101 new_source_utgrid]$
```

**Figure 3.5.3-2.** The table of the axial grid points the mesh generation section, exit the execution and final messages on the consol terminal for the diagonal double slope channel.

### 3.5.5 The execution of UTGRID® with **l\_mod = 0** and **l\_ch, lch\_calc** set to **1**.

If the control parameter **l\_mod** in the input namelist of variables is set to 1 the user must answer either ‘**yes**’ or ‘**no**’ on the consol terminal to the requests about the modification files creation and their availability in the work directory.

The following listing of actions which will be performed during the execution is available for UTGRID® according to the interactive answers typed by the user on the consol terminal:

- **Control run section 1:**
  - Answer ‘**yes**’:
    - **Control confirmation section 1:**
      - Answer ‘**yes**’ – **Predicate 3**;
      - Answer ‘**no**’ – **Predicate 9** and **Predicate10**;
      - Wrong answer – **Predicate 11**.
  - Answer ‘**no**’ :
    - **Control assistance section 1:**
      - Answer ‘**yes**’ – **Predicate 8** and **Predicate 3**;
      - Answer ‘**no**’ - **Predicate 9**;
      - Wrong answer – **Predicate 11**.
  - Wrong answer – **Predicate 11**.
- **Control run section 2:**
  - Answer ‘**yes**’:
    - **Control confirmation section 2:**
      - Answer ‘**yes**’ – **Predicate 3**;

- Answer '**no**' – **Predicate 9** and **Predicate10**;
- Wrong answer – **Predicate 11**.
- Answer '**no**':
  - **Control assistance section 2:**
    - Answer '**yes**' – **Predicate 8** and **Predicate 3**;
    - Answer '**no**' - **Predicate 9**;
    - Wrong answer – **Predicate 11**.
- Wrong answer – **Predicate 11**.
- **Control run section 3:**
  - Answer '**yes**':
    - **Control confirmation section 3:**
      - Answer '**yes**' – **Predicate 3**;
      - Answer '**no**' – **Predicate 9** and **Predicate10**;
      - Wrong answer – **Predicate 11**.
  - Answer '**no**':
    - **Control assistance section 3:**
      - Answer '**yes**' – **Predicate 8** and **Predicate 3**;
      - Answer '**no**' - **Predicate 9**;
      - Wrong answer – **Predicate 11**.
  - Wrong answer – **Predicate 11**.

The predicates used in this list were described previously in section **3.5.3**.

To describe the figures which illustrate the actions during the execution, there a triple of logical variables (L1,L2,L3) is introduced, where the logical variable L1, L2 and L3 can

each be assigned respectively a ‘**yes**’ or a ‘**no**’. The variable L1 represents the answer in the **control run section**, the variable L2 represents the answer on the consol terminal for the **control confirmation section** and the variable L3 represents the **control assistance section**.

The triple (‘**no**’, \*, ‘**yes**’) for all three control run sections defines the actions described in **Figures: 3.5.4-1, 3.5.4-2 and 3.5.4-3** below. The section marked by the red rectangular box is the body of the interactive procedure for modification file creation and will be discussed in detail in section **3.5.6** of this “**Manual**”. For a more clear understanding at this stage, it is shown only as the text of this interactive routine and not the directives.

```
[kkokolok@gggspace101 ~:/publication/new_source_utgrid - Konsole - Konsole]
File Sessions Settings Help

[kkolokol@gggspace101 new_source_utgrid]$ ./UTGRID
> please, enter input file name:
input_45_60
> please, enter print output file name:
UTGRID.dat
> please, enter channel output file name:
channel_45_60.jou
> please, enter nozzle profile file name:
noz_prof.dat
> please, enter nozzle output file name:
nozzle.jou
> please, enter first derivative file name:
of_nozzle_profile_output_file_name:
first_derivative.dat
> please, enter second derivative file name:
of_nozzle_profile_output_file_name:
second_derivative.dat

-----
> for "basic" slope anga = 45.00000000,
you can see the number of grids on the axial edges of conductors
and insulators respectively: n_c(1,...,n), n_ins(1,...,n);
n_con(0) = 51
n_ins(1) = 21
n_con(2) = 35
n_ins(3) = 19
n_con(4) = 35
n_ins(5) = 19
n_con(6) = 35
n_ins(7) = 19
n_con(8) = 35
n_ins(9) = 19
n_con(10) = 35
n_ins(11) = 19
n_con(12) = 54

-----

CONTROL RUN1
> have you prepared file modify1?(yes/no):
no
CONTROL STOP1

STOP1

=====
! MAKE FIRST CHANGES !
=====
```





The triple ('yes', 'no', 'no') for the **control run section 1** is represented on the **Figure 3.5.5-1** which illustrates the list above. The interactive answers are marked as red ellipses on this figure.

The triple ('yes', wrong answer,\*) is illustrated on the **Figure 3.5.5-2**, where the wrong answer is not equivalent to one of the 'yes' or 'no'.

The triple ('no', wrong answer,\*) is represented on the **Figure 3.5.5-3**. The logical value '\*' means that it does not define any actions of the execution and is not used. All referenced figures in this section are located below.

For the nozzle grid generation the user will not get any messages on the consol terminal except that of **Predicate 9**.

```
kkolokol@gggspace101:~/publication/update_source_utgrid - Konsole - Konsole
File Sessions Settings Help

> please, enter channel output file name:
test

-----
> for "basic" slope anga = 90,00000000,
you can see the number of grids on the axial edges of conductors
and insulators respectively: n_c(1,...,n), n_ins(1,...,n):
n_con(0) = 43
n_ins(1) = 19
n_con(2) = 35
n_ins(3) = 19
n_con(4) = 35
n_ins(5) = 19
n_con(6) = 35
n_ins(7) = 19
n_con(8) = 35
n_ins(9) = 19
n_con(10) = 35
n_ins(11) = 19
n_con(12) = 47

-----
CONTROL RUN1
> have you prepared file modify1?(yes/no):
yes
CONTROL STOP1
> are you sure you created file modify1?(yes/no):
no

-----
> Assistance with the creation of modify1:

-----
> Would you like to create interactively modify1? (yes/no):
no
> modify1 will not be created
> END OF RUN

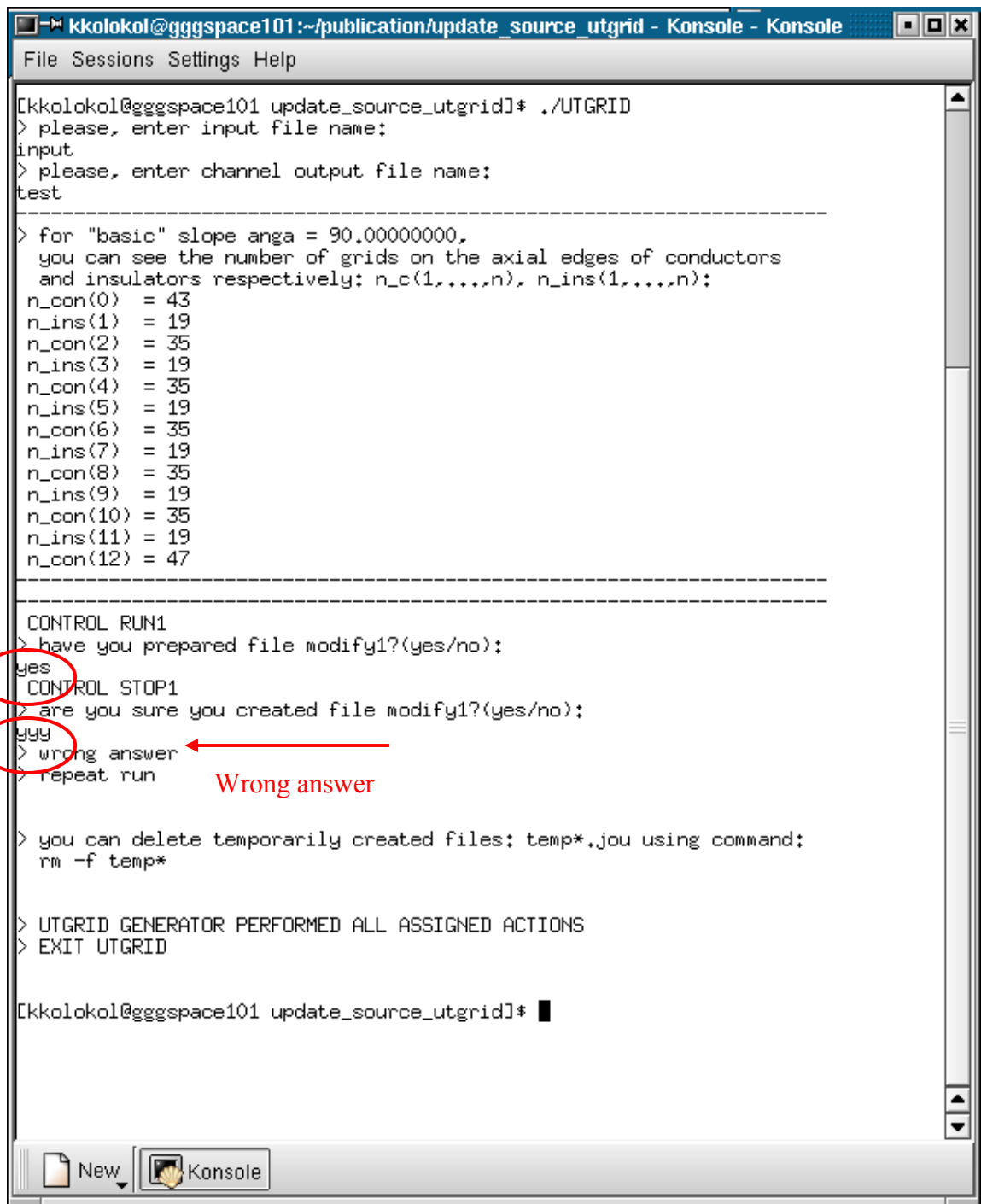
-----
> you can delete temporarily created files: temp*.jou using command:
rm -f temp*

-----
> UTGRID GENERATOR PERFORMED ALL ASSIGNED ACTIONS
> EXIT UTGRID

[kkolokol@gggspace101 update_source_utgrid]#
```

Control confirmation  
section

**Figure 3.5.5-1.** The control step shown on the consol that represents the case of answer ‘yes’ in case if  $l\_mod = 0$ .



```
[kkolokol@gggospace101 update_source_utgrid]# ./UTGRID
> please, enter input file name:
input
> please, enter channel output file name:
test
-----
> for "basic" slope anga = 90.00000000,
  you can see the number of grids on the axial edges of conductors
  and insulators respectively: n_c(1,...,n), n_ins(1,...,n):
n_con(0) = 43
n_ins(1) = 19
n_con(2) = 35
n_ins(3) = 19
n_con(4) = 35
n_ins(5) = 19
n_con(6) = 35
n_ins(7) = 19
n_con(8) = 35
n_ins(9) = 19
n_con(10) = 35
n_ins(11) = 19
n_con(12) = 47
-----

CONTROL RUN1
> have you prepared file modify1?(yes/no):
yes
CONTROL STOP1
> are you sure you created file modify1?(yes/no):
yyy
> wrong answer
> repeat run
Wrong answer

> you can delete temporarily created files: temp*.jou using command:
rm -f temp*

> UTGRID GENERATOR PERFORMED ALL ASSIGNED ACTIONS
> EXIT UTGRID

[kkolokol@gggospace101 update_source_utgrid]#
```

**Figure 3.5.5-2.** The control statements on the consol that indicate a wrong answer on the confirmation request about availability of the modification file in the work directory, if  $l\_mod = 0$ .



### **3.5.6. Modification files creation.**

#### **3.5.6.1. Introduction.**

Again, because of the extensive number of figures required to illustrate, define, and identify control and geometrical parameters required in this section, all figures in this section are located in the **APPENDIX A1**, pages 157 - 284.

As was mentioned already in this thesis, the only crucial issue for full flexibility of the input parameters of UTGRID® is the creation of modification files by the user for the pre-selected number of electric elements of the channel. When those files are created, all other input parameters are fully flexible.

To created modification files the user must follow the interactive procedure and type the answers to the program requests on the console terminal. All geometrical elements used for the modifications of their labels are represented and shown in figures in **APPENDIX A1**, pages 157 - 284. The changes are different for the straight, diagonal and double slope channels for the ‘modify3\_s’ and ‘modify3\_d’ files. For a full demonstration and illustration of the use of the modification files all three cases will be discussed below.

First of all the user must perform the request to start GAMBIT™ simultaneously with the execution of UTGRID® and run, respectively, the correct control input needed at each step to convert the channel output file to a new journal file.

As it was described in the sections **3.5.4** and **3.5.5**, the content of the channel output file at each **control assistance section** is as follows:

- **control assistance section 1** – ‘templ.jou’

The modification file ‘modify1’ is created at this step

- **control assistance section 2** – the concatenation of the following files:

‘temp1.jou’, ‘modify1’ and ‘temp2.jou’

The modification file ‘modify2’ is created at this step

- **control assistance section 3** – the concatenation of the following files:

‘temp1.jou’, ‘modify1’, ‘temp2.jou’, ‘modify2’ and ‘temp3.jou’

The modification file ‘modify3\_s’ or ‘modify3\_d’ is created at this step

The user can also open and keep the files consequently: ‘temp1.jou’, ‘temp2.jou’ and ‘temp3.jou’ that is saving the previous output of GAMBIT™ after the creation of the appropriate modification file. However the author of this Thesis has believes that his method which is to open the journal files as the new working files is more convenient for the user.

The beginning of the mesh geometry element numeration process starts at the coordinate origin, the inlet plane of the channel for all geometrical elements: edges, faces and volumes.

In each **control assistance section** of the interactive procedure, the user must type on the consol terminal the number of the first, the second and the last geometrical element delimiting them by space.

For an example, consider entering the labels for the axial plasma faces which are among the geometrical elements mentioned above. These faces are labeled as: “**face.71**”, “**face.66**” and “**face.2**”. The user must type on the consol terminal: **71 66 2** and press “**Enter**” on the keyboard. The code UTGRID® will give the labels for the axial plasma faces as: “**xfp0**”, ... , “**xfp<2\*n>**” respectively, where n is the number of the insulators in the channel. The next request will be for the axial solid faces etc. for all needed geometrical elements and modification files. The procedure for the construction

of the file ‘modify2’ is slightly different, and the procedure of the modifications for all files will be described below in this section.

The following notations will be used to introduce the classes of the geometrical elements:

- “**maximum**” – the class of the geometrical elements (edges) located on the upper half plane or space, where the distance from the inlet to the first insulator of the channel is maximal as represented on **Figure 3.5.77**;
- “**minimum**” – the class of the geometrical elements (edges) located on the upper half plane or space, where the distance from the inlet to the first insulator of the channel is minimal as represented on **Figure 3.5.77**;
- “**left**” - the class of the geometrical elements (edges and faces) located on the left half of the channel geometry as represented on **Figure 3.5.57** for the elliptic air edges and on **Figure 3.5.68** for the radial solid faces;
- “**right**” - the class of the geometrical elements (edges and faces) located on the right half of the channel geometry as represented on **Figure 3.5.84** for the elliptic plasma edges and on **Figure 3.5.94** for the radial solid faces;
- “**up**” – the class of the geometrical elements for the straight channel (edges and faces) located on the upper half of the channel geometry as represented on **Figure 5.5.42**;
- “**down**” – the class of the geometrical elements for the straight channel (edges and faces) located on the lower half of the channel geometry as represented on **Figure 3.5.43**;
- “**shift**” – the angle between the ZY –plane and the cut plane for the channel

geometry which is defined by the radial edges of the air, and solid zones. For the double slope channel, the example is represented on **Figure 3.5.70**; for a one slope channel, this angle is equal zero; for the straight channel the example is illustrated on **Figure 3.5.121**.

This interactive procedure requires accuracy from the user and will not take too much time for the creation of modification files even for a very big number of electric elements in the channel. This task is very difficult to perform manually and had not been done before the creation of UTGRID® by the author of this Thesis.

In the list of the directives below, there are represented classes and subclasses of geometry labels for all modification files for three cases: straight channel (slope angles are: 90, 90), diagonal, one slope channel (slope angles are: 45, 90) and diagonal double - slope channel (slope angles are: 45, 60). The 60 degrees second slope angle was an arbitrary choice of the author. The qualitative differences between the straight channel, one slope channel and double – slope channel will be in detail described in the list below, where the appropriate figures will be consequently and respectively referenced.

### **3.5.6.2. Modification file ‘modify1’.**

The labels of modification file ‘modify1’ must be as follows:

- **Modification file ‘modify1 (slope angles: 45, 60)**

- Axial plasma faces:

**– Figure 3.5.16**

- Axial solid faces:

**– Figure 3.5.17**

- Axial air faces:  
“**xfa**”; only one number must be entered – **Figure 3.5.18**
- Radial air faces:  
“**rfa0**”, “**rfa<2\*n+1>**”; index = 0, 2\*n+1 – **Figure 3.5.13**
- Radial plasma faces:  
“**rfp0**”, ... , “**rfp<2\*n+1>**”; index = 0, ... , 2\*n+1 – **Figure 3.5.14**
- Radial solid faces:  
“**rfs0**”, ... , “**rfs<2\*n+1>**”; index = 0, ... , 2\*n+1 – **Figure 3.5.15**
- Plasma volumes:  
“**vp0**”, ... , “**vp<2\*n>**”; index = 0, ... , 2\*n – **Figures 3.5.19**
- Solid volumes:  
“**vs0**”, ... , “**vs<2\*n>**”; index = 0, ... , 2\*n – **Figure 3.5.20**
- **Modification file ‘modify1’ (slope angles: 45, 90)**
  - Axial plasma faces:  
“**xfp0**”, ... , “**xfp<2\*n>**”; index = 0, ... , 2\*n – **Figure 3.5.24**
  - Axial solid faces:  
“**xfp0**”, ... , “**xfp<2\*n>**”; index = 0, ... , 2\*n – **Figure 3.5.25**
  - Axial air faces:  
“**xfa**”; only one number must be entered – **Figure 3.5.26**
  - Radial air faces:  
“**rfa0**”, “**rfa<2\*n+1>**”; index = 0, 2\*n+1 – **Figure 3.5.21**
  - Radial plasma faces:  
“**rfp0**”, ... , “**rfp<2\*n+1>**”; index = 0, ... , 2\*n+1 – **Figure 3.5.22**

- Radial solid faces:

“**rfs0**”, ... , “**rfs<2\*n+1>**”; index = 0, ... , 2\*n+1 - **Figure 3.5.23**

- Plasma volumes:

“**vp0**”, ... , “**vp<2\*n>**”; index = 0, ... , 2\*n - **Figures 3.5.27**

- Solid volumes:

“**vs0**”, ... , “**vs<2\*n>**”; index = 0, ... , 2\*n - **Figure 3.5.28**

- Air volumes:

“**va**” - **Figure 3.5.29**

The air volume is not changed and showed here for illustration only.

- **Modification file ‘modify1’ (slope angles: 90, 90)**

- Axial plasma faces:

“**xfp0**”, ... , “**xfp<2\*n>**”; index = 0, ... , 2\*n - **Figure 3.5.32**

- Axial solid faces:

“**xfp0**”, ... , “**xfp<2\*n>**”; index = 0, ... , 2\*n - **Figure 3.5.33**

- Axial air faces:

“**xfa**”; only one number must be entered - **Figure 3.5.34**

- Radial air faces:

“**rfa0**”, “**rfa<2\*n+1>**”; index = 0, 2\*n+1 - **Figure 3.5.30**

- Radial plasma faces:

“**rfp0**”, ... , “**rfp<2\*n+1>**”; index = 0, ... , 2\*n+1 - **Figure 3.5.31**

- Radial solid faces:

“**rfs0**”, ... , “**rfs<2\*n+1>**”; index = 0, ... , 2\*n+1 – not showed, but similar

- Plasma volumes:

**“vp0”, ... , “vp<2\*n>”; index = 0, ... , 2\*n** - **Figures 3.5.35**

- Solid volumes:

**“vs0”, ... , “vs<2\*n>”; index = 0, ... , 2\*n** - **Figure 3.5.36**

- Air volume:

**“va”** - **Figure 3.5.37**

The label for the air volume is not changed and represented for illustration.

### 3.5.6.3. Modification file ‘modify2’.

The labels of modification file ‘modify2’ must be as follows:

- **Modification file ‘modify2’ (slope angles: 45, 60)**

- Axial air face of the **“left”**:

**“xfa2”** - **Figure 3.5.39**

Only one number has to be entered (for the left face). In the particular example where  $n = 6$ , **“face.76”** will be assigned the label **“xfa2”**. The user will be requested to enter the number **76** on the consol terminal.

- Axial face of the **“right”**:

By the program code the face **“xfa”** will be assigned to the label: **“xfa1”**

- **Figure 3.5.38**

- **Modification file ‘modify2’ (slope angles: 45, 90)**

- Axial air face of the **“left”**:

**“xfa2”** - **Figure 3.5.41**

Only one number has to be entered (for the left face). In particular example where  $n = 6$ . **“face.76”** will be assigned the label **“xfa2”**. The user will be

requested to enter the number **76** on the consol terminal.

- Axial face of the “**right**”:

By the program code the face “**xfa**” will be assigned the label: “**xfa1**”

- **Figure 3.5.40**

- **Modification file ‘modify2’ (slope angles: 90, 90)**

- Axial air face of the “**up**”:

“**xfa1**”

- **Figure 3.5.42**

Only one number has to be entered (for the left face). In the particular example where  $n = 6$  “**face.76**” will be assigned the label “**xfa2**”. The user will be requested to enter the number **76** on the consol terminal.

- Axial face of the “**down**”:

By the program code the face “**xfa**” will be assigned to the label: “**xfa2**”

- **Figure 3.5.43**

#### **3.5.6.4. Modification file ‘modify3\_d’ or ‘modify3\_s’.**

The labels of modification file ‘modify2’ must be as follows:

- **Modification file ‘modify3\_d’ (slope angles: 45, 60)**

- Axial plasma edges of the “**maximum**”:

“**xep1\_0**”, ... , “**xep<2\*n>**”; index = 0, ... ,  $2*n$

- **Figure 3.5.44**

- Axial plasma edges of the “**minimun**”:

“**xep2\_0**”, ... , “**xep<2\*n>**”; index = 0, ... ,  $2*n$

- **Figure 3.5.45**

- Axial solid edges of the “**maximum**”:

“**xes1\_0**”, ... , “**xes1\_<2\*n>**”; index = 0, ... ,  $2*n$

- **Figure 3.5.46**

- Axial solid edges of the “**minimum**”:  
“**xes2\_0**”, ... , “**xes2\_<2\*n>**”; index = 0, ... , 2\*n - **Figure 3.5.47**
- Axial air edges of the “**maximum**”:  
“**xa1\_0**”, ... , “**xa1\_<2\*n>**”; index = 0, ... , 2\*n - **Figure 3.5.48**
- Axial air edges of the “**minimum**”:  
“**xa2\_0**”, ... , “**xa2\_<2\*n>**”; index = 0, ... , 2\*n - **Figure 3.5.49**
- Radial solid edges of the “**maximum**”:  
“**res1\_0**”, ... , “**res1\_<2\*n+1>**”; index = 0, ... , 2\*n+1 - **Figure 3.5.50**
- Radial solid edges of the “**minimum**”:  
“**res2\_0**”, ... , “**res2\_<2\*n+1>**”; index = 0, ... , 2\*n+1 - **Figure 3.5.51**
- Radial air edges of the “**maximum**”:  
“**ra1\_0**”, “**ra1\_<2\*n>**”; only two edges - **Figure 3.5.52**
- Radial air edges of the “**minimum**”:  
“**ra2\_0**”, “**ra2\_<2\*n>**”; only two edges - **Figure 3.5.53**
- Elliptic solid edges of the “**left**”:  
“**es1\_0**”, ... , “**es1\_<2\*n+1>**”; index = 0, ... , 2\*n+1 - **Figure 3.5.54**
- Elliptic solid edges of the “**right**”:  
“**es21\_0**”, “**es22\_0**”, ... , “**es21\_<2\*n+1>**”, “**es22\_<2\*n+1>**”;  
index = 0, ... , 2\*n+1 - **Figure 3.5.55**

The edge “**es21\_0**” is relative to the longest elliptic edge on the inlet.

The edge “**es22\_0**” is relative to the shortest elliptic edge on the inlet.

The edge “**es21\_13**” is relative to the longest elliptic edge on the outlet.

The edge “**es22\_13**” is relative to the shortest elliptic edge on the outlet.

In this example:  $2*n+1 = 13$  for  $n = 6$ .

- Elliptic air edges of the “**left**”:

“**ea1\_0**”, ... , “**ea1\_<2\*n+1>**”; index = 0, ... ,  $2*n+1$  - **Figure 3.5.56**

- Elliptic air edges of the “**right**”:

“**ea21\_0**”, “**ea22\_0**”, ... , “**ea21\_<2\*n+1>**”, “**ea22\_<2\*n+1>**”;  
index = 0, ... ,  $2*n+1$  - **Figure 3.5.57**

The edge “**ea21\_0**” is relative to the longest elliptic edge on the inlet.

The edge “**ea22\_0**” is relative to the shortest elliptic edge on the inlet.

The edge “**ea21\_13**” is relative to the longest elliptic edge on the outlet.

The edge “**ea22\_13**” is relative to the shortest elliptic edge on the outlet.

In this example:  $2*n+1 = 13$  for  $n = 6$ .

- Elliptic plasma edges of the “**left**”:

“**ep1\_0**”, ... , “**ep1\_<2\*n+1>**”; index = 0, ... ,  $2*n+1$  - **Figure 3.5.58**

- Elliptic plasma edges of the “**right**”:

“**ep21\_0**”, “**ep22\_0**”, ... , “**ep21\_<2\*n+1>**”, “**ep22\_<2\*n+1>**”;  
index = 0, ... ,  $2*n+1$  - **Figure 3.5.59**

The edge “**ep21\_0**” is relative to the longest elliptic edge on the inlet.

The edge “**ep22\_0**” is relative to the shortest elliptic edge on the inlet.

The edge “**ep21\_13**” is relative to the longest elliptic edge on the outlet.

The edge “**ep22\_13**” is relative to the shortest elliptic edge on the outlet.

In this example:  $2*n+1 = 13$  for  $n = 6$ .

- Axial plasma faces of the “**left**”:

“**xfp1\_0**”, ... , “**xfp<2\*n>**”; index = 0, ... ,  $2*n$  - **Figure 3.5.60**

- Axial plasma faces of the “**right**”:  
“xfs2\_0”, ... , “xfs2\_<2\*n>”; index = 0, ... , 2\*n - **Figure 3.5.61**
- Axial solid faces of the “**left**”:  
“xfs1\_0”, ... , “xfs1\_<2\*n>”; index = 0, ... , 2\*n - **Figure 3.5.62**
- Axial solid faces of the “**right**”:  
“xfs2\_0”, ... , “xfs2\_<2\*n>”; index = 0, ... , 2\*n - **Figure 3.5.63**
- Axial air faces of the “**left**”:  
“xfa1\_0”, ... , “xfa1\_<2\*n>”; index = 0, ... , 2\*n - **Figure 3.5.64**
- Axial air faces of the “**right**”:  
“xfa2\_0”, ... , “xfa2\_<2\*n>”; index = 0, ... , 2\*n - **Figure 3.5.65**
- Radial air faces of the “**left**”:  
“rfa1\_0”, “rfa1\_<2\*n+1>”; only two faces - **Figure 3.5.66**
- Radial air faces of the “**right**”:  
“rfa2\_0”, “rfa2\_<2\*n+1>”; only two faces - **Figure 3.5.67**
- Radial solid faces of the “**left**”:  
“rfs1\_0”, ... , “rfs1\_<2\*n+1>”; index = 0, ... , 2\*n+1 - **Figure 3.5.68**
- Radial solid faces of the “**right**”:  
“rfs2\_0”, ... , “rfs2\_<2\*n+1>”; index = 0, ... , 2\*n+1 - **Figure 3.5.69**
- The “shift” angle is represented on figure **Figure 3.5.70**  
This angle is introduced in section **3.5.6.1**.
- **Modification file ‘modify3\_d’ (slope angles: 45, 90)**
  - Axial plasma edges of the “**maximum**”:  
“xep1\_0”, ... , “xep1\_<2\*n>”; index = 0, ... , 2\*n - **Figure 3.5.71**

- Axial plasma edges of the “**minimun**”:  
“xep2\_0”, ... , “xep<2\*n>”; index = 0, ... , 2\*n - **Figure 3.5.72**
- Axial solid edges of the “**maximum**”:  
“xes1\_0”, ... , “xes1\_<2\*n>”; index = 0, ... , 2\*n - **Figure 3.5.73**
- Axial solid edges of the “**minimum**”:  
“xes2\_0”, ... , “xes2\_<2\*n>”; index = 0, ... , 2\*n - **Figure 3.5.74**
- Axial air edges of the “**maximum**”:  
“xa1\_0”, ... , “xa1\_<2\*n>”; index = 0, ... , 2\*n - **Figure 3.5.75**
- Axial air edges of the “**minimum**”:  
“xa2\_0”, ... , “xa2\_<2\*n>”; index = 0, ... , 2\*n - **Figure 3.5.76**
- Radial solid edges of the “**maximum**”:  
“res1\_0”, ... , “res1\_<2\*n+1>”; index = 0, ... , 2\*n+1 - **Figure 3.5.77**
- Radial solid edges of the “**minimum**”:  
“res2\_0”, ... , “res2\_<2\*n+1>”; index = 0, ... , 2\*n+1 - **Figure 3.5.77**
- Radial air edges of the “**maximum**”:  
“ra1\_0”, “ra1\_<2\*n>”; only two edges - **Figure 3.5.78**
- Radial air edges of the “**minimum**”:  
“ra2\_0”, “ra2\_<2\*n>”; only two edges - **Figure 3.5.79**
- Elliptic solid edges of the “**left**”:  
“es1\_0”, ... , “es1\_<2\*n+1>”; index = 0, ... , 2\*n+1 - **Figure 3.5.54**
- Elliptic solid edges of the “**right**”:  
“es21\_0”, “es22\_0”, ... , “es21\_<2\*n+1>”, “es22\_<2\*n+1>”;  
index = 0, ... , 2\*n+1 0 - **Figure 3.5.55**

The edge “**es21\_0**” is relative to the longest elliptic edge on the inlet.

The edge “**es22\_0**” is relative to the shortest elliptic edge on the inlet.

The edge “**es21\_13**” is relative to the longest elliptic edge on the outlet.

The edge “**es22\_13**” is relative to the shortest elliptic edge on the outlet.

In this example:  $2*n+1 = 13$  for  $n = 6$ .

- Elliptic air edges of the “**left**”:

“**ea1\_0**”, ... , “**ea1\_<2\*n+1>**”; index = 0, ... ,  $2*n+1$  - **Figure 3.5.56**

- Elliptic air edges of the “**right**”:

“**ea21\_0**”, “**ea22\_0**”, ... , “**ea21\_<2\*n+1>**”, “**ea22\_<2\*n+1>**”;  
index = 0, ... ,  $2*n+1$  - **Figure 3.5.57**

The edge “**ea21\_0**” is relative to the longest elliptic edge on the inlet.

The edge “**ea22\_0**” is relative to the shortest elliptic edge on the inlet.

The edge “**ea21\_13**” is relative to the longest elliptic edge on the outlet.

The edge “**ea22\_13**” is relative to the shortest elliptic edge on the outlet.

In this example:  $2*n+1 = 13$  for  $n = 6$ .

- Elliptic plasma edges of the “**left**”:

“**ep1\_0**”, ... , “**ep1\_<2\*n+1>**”; index = 0, ... ,  $2*n+1$  - **Figure 3.5.58**

- Elliptic plasma edges of the “**right**”:

“**ep21\_0**”, “**ep22\_0**”, ... , “**ep21\_<2\*n+1>**”, “**ep22\_<2\*n+1>**”;  
index = 0, ... ,  $2*n+1$  - **Figure 3.5.59**

The edge “**ep21\_0**” is relative to the longest elliptic edge on the inlet.

The edge “**ep22\_0**” is relative to the shortest elliptic edge on the inlet.

The edge “**ep21\_13**” is relative to the longest elliptic edge on the outlet.

The edge “**ep22\_13**” is relative to the shortest elliptic edge on the outlet.

In this example:  $2*n+1 = 13$  for  $n = 6$ .

- Axial plasma faces of the “**left**”:  
“**xfp1\_0**”, ... , “**xfp<2\*n>**”; index = 0, ... ,  $2*n$  - **Figure 3.5.60**
- Axial plasma faces of the “**right**”:  
“**xfp2\_0**”, ... , “**xfp2\_<2\*n>**”; index = 0, ... ,  $2*n$  - **Figure 3.5.61**
- Axial solid faces of the “**left**”:  
“**xfs1\_0**”, ... , “**xfs1\_<2\*n>**”; index = 0, ... ,  $2*n$  - **Figure 3.5.62**
- Axial solid faces of the “**right**”:  
“**xfs2\_0**”, ... , “**xfs2\_<2\*n>**”; index = 0, ... ,  $2*n$  - **Figure 3.5.63**
- Axial air faces of the “**left**”:  
“**xfa1\_0**”, ... , “**xfa1\_<2\*n>**”; index = 0, ... ,  $2*n$  - **Figure 3.5.64**
- Axial air faces of the “**right**”:  
“**xfa2\_0**”, ... , “**xfa2\_<2\*n>**”; index = 0, ... ,  $2*n$  - **Figure 3.5.65**
- Radial air faces of the “**left**”:  
“**rfa1\_0**”, “**rfa1\_<2\*n+1>**”; only two faces - **Figure 3.5.66**
- Radial air faces of the “**right**”:  
“**rfa2\_0**”, “**rfa2\_<2\*n+1>**”; only two faces - **Figure 3.5.67**
- Radial solid faces of the “**left**”:  
“**rfs1\_0**”, ... , “**rfs1\_<2\*n+1>**”; index = 0, ... ,  $2*n+1$  - **Figure 3.5.68**
- Radial solid faces of the “**right**”:  
“**rfs2\_0**”, ... , “**rfs2\_<2\*n+1>**”; index = 0, ... ,  $2*n+1$  - **Figure 3.5.69**
- The “**shift**” is represented on **Figure 3.5.70**

This angle is introduced in the section **3.5.6.1**.

- **Modification file ‘modify3\_s’ (slope angles: 90, 90 – straight channel)**

- Axial plasma edges of the “left”:

“xep1\_0”, ... , “xep1\_<2\*n>”; index = 0, ... , 2\*n - **Figure 3.5.96**

- Axial plasma edges of the “right”:

“xep2\_0”, ... , “xep2\_<2\*n+1>”; index = 0, 2\*n - **Figure 3.5.97**

- Axial solid edges of the “left”:

“xes1\_0”, ... , “xes1\_<2\*n>”; index = 0, ... , 2\*n - **Figure 3.5.98**

- Axial plasma edges of the “right”:

“xes2\_0”, ... , “xes2\_<2\*n+1>”; index = 0, 2\*n - **Figure 3.5.99**

- Axial air edges of the “left”:

“xa1\_0”, ... , “xa1\_<2\*n>”; index = 0, ... , 2\*n - **Figure 3.5.100**

- Axial air edges of the “right”:

“xa2\_0”, ... , “xa2\_<2\*n+1>”; index = 0, 2\*n - **Figure 3.5.101**

- Radial solid edges of the “left”:

“res1\_0”, ... , “res1\_<2\*n>”; index = 0, 2\*n - **Figure 3.5.102**

- Radial solid edges of the “right”:

“res2\_0”, ... , “res2\_<2\*n>”; index = 0, 2\*n - **Figure 3.5.102**

- Radial air edges of the “left”:

“ra1\_0”, “ra1\_<2\*n>”; only two elements – **Figure 3.5.103**

2\*n = 12 for n = 6 in this example.

- Radial air edges of the “right”:

“ra2\_0”, “ra2\_<2\*n>”; only two elements – **Figure 3.5.104**

$2*n = 12$  for  $n = 6$  in this example.

- Elliptic (round) solid edges of the “**down**”;  
“**es1\_0**”, ... , “**es1\_<2\*n+1>**”; index = 0, ... ,  $2*n+1$  - Figure 3.5.105
- Elliptic (round) solid edges of the “**up**”;  
“**es2\_0**”, ... , “**es2\_<2\*n+1>**”; index = 0, ... ,  $2*n+1$  - Figure 3.5.106
- Elliptic (round) solid edges of the “**down**”;  
“**es1\_0**”, ... , “**es1\_<2\*n+1>**”; index = 0, ... ,  $2*n+1$  - Figure 3.5.105
- Elliptic (round) air edges of the “**up**”;  
“**es2\_0**”, ... , “**es2\_<2\*n+1>**”; index = 0, ... ,  $2*n+1$  - Figure 3.5.108
- Elliptic (round) plasma edges of the “**down**”;  
“**ep1\_0**”, ... , “**ep1\_<2\*n+1>**”; index = 0, ... ,  $2*n+1$  - Figure 3.5.109
- Elliptic (round) plasma edges of the “**up**”;  
“**ep2\_0**”, ... , “**ep2\_<2\*n+1>**”; index = 0, ... ,  $2*n+1$  - Figure 3.5.110
- Axial plasma faces of the “**down**”:  
“**xfp1\_0**”, ... , “**xfp1\_<2\*n>**”; index = 0,  $2*n$  - Figure 3.5.111
- Axial plasma faces of the “**up**”:  
“**xfp2\_0**”, ... , “**xfp2\_<2\*n>**”; index = 0,  $2*n$  - Figure 3.5.112
- Axial solid faces of the “**down**”:  
“**xfs1\_0**”, ... , “**xfs1\_<2\*n>**”; index = 0,  $2*n$  - Figure 3.5.113
- Axial solid faces of the “**up**”:  
“**xfs2\_0**”, ... , “**xfs2\_<2\*n>**”; index = 0,  $2*n$  - Figure 3.5.114
- Axial air faces of the “**down**”:  
“**xfa1\_0**”, ... , “**xfa1\_<2\*n>**”; index = 0,  $2*n$  - Figure 3.5.115

- Axial air faces of the “**up**”:

“**xfa2\_0**”, ... , “**xfa2\_<2\*n>**”; index = 0, 2\*n - **Figure 3.5.116**

- Radial air faces of the “**down**”:

“**rfa1\_0**”, “**rfa1\_<2\*n>**”; only two elements - **Figure 3.5.117**

2\*n = 12 for n = 6 in this example.

- Radial air faces of the “**up**”:

“**rfa2\_0**”, “**rfa2\_<2\*n>**”; only two elements - **Figure 3.5.118**

2\*n = 12 for n = 6 in this example.

- Radial solid faces of the “**down**”:

“**rfs1\_0**”, ... , “**rfs1\_<2\*n>**”; index = 0, ... , 2\*n - **Figure 3.5.119**

- Radial solid faces of the “**up**”:

- “**rfs2\_0**”, ... , “**rfs2\_<2\*n>**”; index = 0, ... , 2\*n - **Figure 3.5.40**

- “**shift**”:

**Figure 3.5.121**

### 3.5.6.5. Recommendations.

When only two geometrical elements must be entered on the consol, the user will be notified and requested appropriately by UTGRID®. In turn, when it is necessary to enter the longest and shortest edges for the double slope channel, the user also will be notified and requested about this terminal input. The routine for the modification files creation is not the short procedure and the user must be extremely accurate entering data on the console terminal. The user must respond to each particular request of UTGRID® and type “Enter” on the keyboard after responding. The GAMBIT® must be open during this

interactive procedure and work in parallel.

After the user has created the necessary modification files, he would be encouraged, since all other geometrical parameters in the input file are fully flexible. It is necessary to create modification files just once for each specific case based on the number of insulators in the channel. The user can also create a library of modification files for a variety of the insulator cases, for example, channels with 6, 10, or even 1000 insulators.

If the user makes an error during the interactive procedure (like, entering the wrong number of the geometrical elements) nothing obviously wrong will happen except that the geometry will not be meshed as assigned and allowed by the grid algorithm.

For any comments or suggestions about using UTGRID® for creating the geometrical modification files, contact the author for further assistance.

### **3.6. Output of UTGRID®.**

The available output files of UTGRID® which control the meshing process of GAMBIT™, are described below and are associated with the corresponding control parameters described in section 3.5:

- **Nozzle:**
  - Nozzle without the adaptor and channel extension (ASCII journal file for GAMBIT™);
  - Nozzle with the adaptor attached to the nozzle (ASCII journal file for GAMBIT™);
  - Nozzle with the channel extension attached to the adaptor which is attached to the nozzle outlet (ASCII journal file for the GAMBIT™);

- The first derivative of the nozzle wall contour or the profile (ASCII file);
- The second derivative of the nozzle wall contour or the profile (ASCII file).
- **Channel:**
  - Double slope channel grid generation file (ASCII journal file for GAMBIT™);
  - One slope channel grid generation file (ASCII journal file for GAMBIT™);
  - Straight channel grid generation file (ASCII journal file for GAMBIT™).
- **Modification files:**
  - File 'modify1' (ASCII file);
  - File 'modify2' (ASCII file);
  - File 'modify3\_d' ASCII file);
  - File 'modify3\_s' (ASCII file).
- **Temporary created files:**
  - File 'temp1.jou' (ASCII journal file for GAMBIT™);
  - File 'temp2.jou' (ASCII journal file for GAMBIT™);
  - File 'temp3.jou' (ASCII journal file for GAMBIT™);
  - File 'temp4\_d.jou' (ASCII journal file for GAMBIT™);
  - File 'temp4\_d.jou' (ASCII journal file for GAMBIT™);
- **Print file which contains (ASCII file):**
  - Calculated and inputted geometrical data for the channel;
  - Calculated data for test and debug purposes;
  - Calculated and inputted mesh data for the channel (reserved);
  - Calculated and inputted geometrical data for the nozzle (reserved);

- Calculated and inputted mesh data for the nozzle (reserved);
- Other data (reserved).
- **Text output on the consol terminal:**
  - Control run section;
  - Control assistance section;
  - Control confirmation section;
  - Service suggested directives;
  - Final message.

The structure of the execution process of UTGRID® is completely described in sections 3.5.4 and 3.5.5 in the respective lists of allowed logical operations.

In general, UTGRID® could be developed to automate meshing of a multi-component grid generator, if some additional modules and possible external subroutines are added to mesh the specific computational domains. In this Thesis, two-component grid generation algorithm was developed: for the nozzle and for the channel. As was mentioned already in this Thesis, UTGRID® allows addition of other modules and external subroutines for grid generation for other specific computational domains, which implies multi-component mesh-generation process.

GAMBIT™ is the postprocessor for UTGRID® and GAMBIT™ provides its output in a variety of formats. One format that is used by PGRID® is the neutral file format with the extension: \*.neu. This format is acceptable by the mesh processor code PGRID®, as the input for grid file generation in a format acceptable to GEMS®. The complete procedure of grid generation for GEMS® will be discussed later in the section 3.5.7.

GAMBIT™ also allows for saving the mesh file it produced, based on the input received

form UTGRID®, in another format with the extension: \*.msh. This is a mesh format and it is acceptable by the postprocessor TECPLOT™ for numerical analyses of the mesh file by the tools of this postprocessor.

ASCII files can be opened with any text editor, however the extension for journal files must be as follows: \*.jou.

### **3.7. The procedure of grid generation in the format acceptable by GEMS® with linked programs: UTGRID®, GAMBIT™, PGRID®.**

1. Compile UTGRID® with the Portland Fortran 90 Compiler and install (create the executable file);
2. Choose the set of parameters needed to construct or modify the data and control variables input file for UTGRID®, accordingly to section 3.5;
3. Execute UTGRID® and generate modification files if they are not created for the channel, working with GAMBIT™ in parallel accordingly to the manual in the section 3.5;
4. Execute UTGRID® and generate the output of the UTGRID® for channel and/or nozzle if modification files are created;
5. Open the output file of the UTGRID® as new journal file in GAMBIT™  
(it will take some certain appropriate time for the execution depending on the size of the file);
6. Save the output of GAMBIT™ in neutral file format; this will assign the extension: \*.neu for it;
7. Compile the program PGRID® with the Fortran 90 Compiler (Portland[16] or

Lahey [21]) with the source file, linked to the library: 'libmetis.o' and install (create the executable file) this program on the specific computer where it will be executed;

8. Execute the pre-utility PGRID® and type all needed answers on the requests of PGRID® on the consol terminal:

- 8.1. Type the location of the input file (this is the output of GAMBIT™ in neutral format, produced from the output of UTGRID®);

**Figure 3.7.1** below illustrates this action;

- 8.2. Type the dimension of the domain;

**Figure 3.7.1** illustrates this action as well **Figure 3.7.2**

- 8.3. Type the amount of the partitions assigned to the computational domain;

**Figure3.7.3** and **Figure 3.7.4** illustrate this action.

The output file of the pre-utility PGRID® represents the mesh input file, in the format acceptable, to GEMS® and also in format acceptable by TECPLOT™ which will be additionally described in the next section.

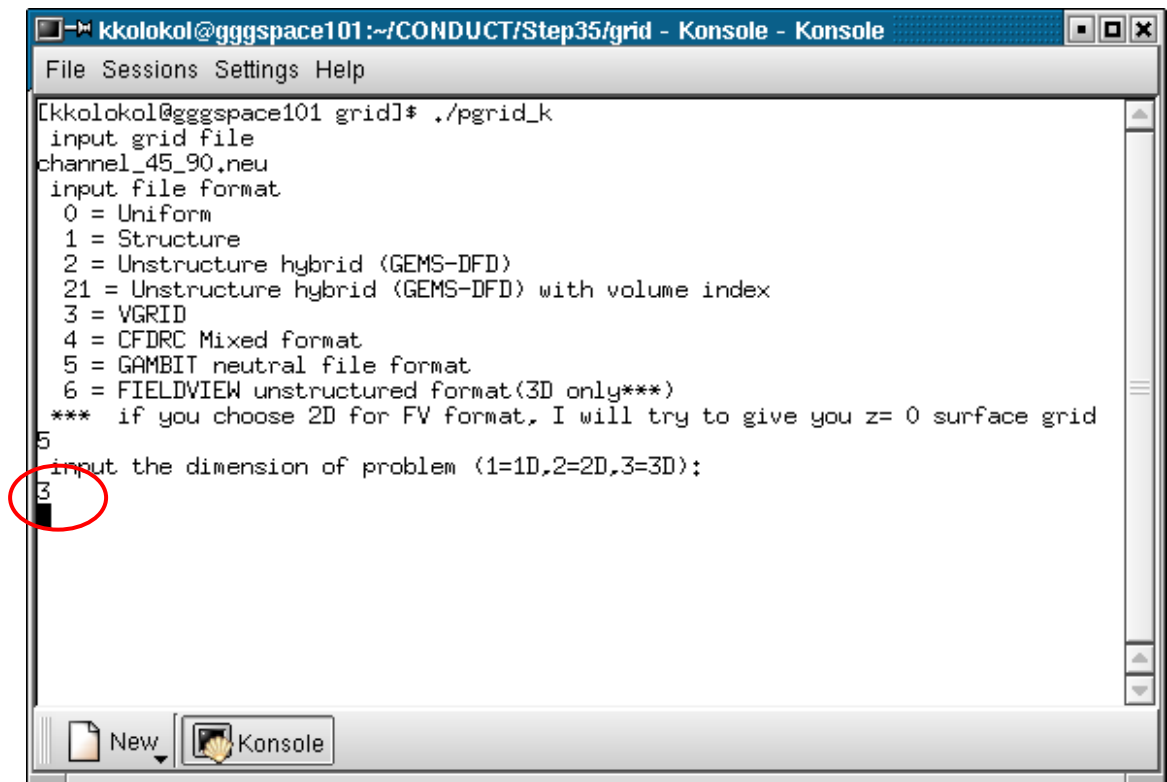
```
kkolokol@gggspace101:~/CONDUCT/Step35/grid - Konsole - Konsole
File Sessions Settings Help
[kkolokol@gggspace101 grid]* ./pgrid_k
input grid file
channel_45_90.new
input file format
 0 = Uniform
 1 = Structure
 2 = Unstructure hybrid (GEMS-DFD)
21 = Unstructure hybrid (GEMS-DFD) with volume index
 3 = VGRID
 4 = CFDRC Mixed format
 5 = GAMBIT neutral file format
 6 = FIELDVIEW unstructured format(3D only***)
*** if you choose 2D for FV format, I will try to give you z= 0 surface grid
5
input the dimension of problem (1=1D,2=2D,3=3D);
```

Input file

Acceptable formats of the input

The dimension of the domain

**Figure 3.7.1.** The request of PGRID® on the available dimension of the input.



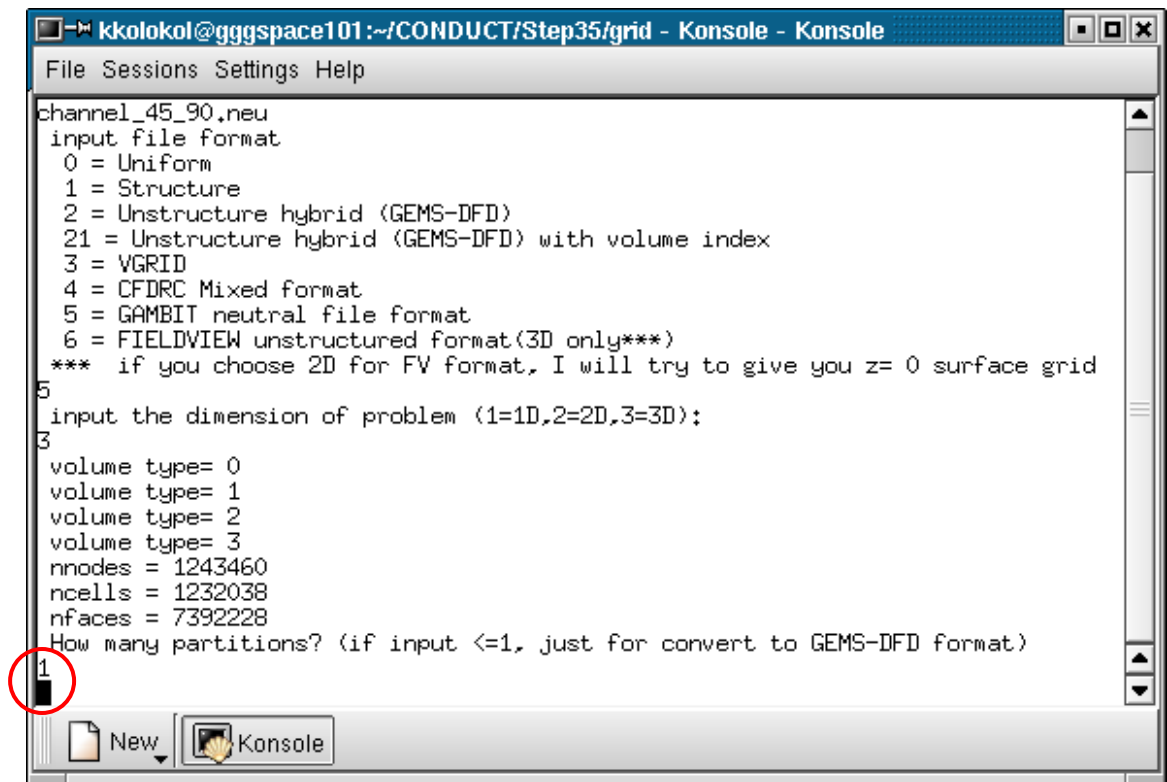
```
kkolokol@gggspace101:~/CONDUCT/Step35/grid - Konsole - Konsole
File Sessions Settings Help
[kkolokol@gggspace101 grid]* ./pgrid_k
input grid file
channel_45_90.neu
input file format
  0 = Uniform
  1 = Structure
  2 = Unstructure hybrid (GEMS-DFD)
  21 = Unstructure hybrid (GEMS-DFD) with volume index
  3 = VGRID
  4 = CFDRM Mixed format
  5 = GAMBIT neutral file format
  6 = FIELDVIEW unstructured format(3D only**)
*** if you choose 2D for FV format, I will try to give you z= 0 surface grid
5
input the dimension of problem (1=1D,2=2D,3=3D):
3
```

**Figure 3.7.2.** The entered dimension of the input is set to 3 in the example.

```
input grid file
channel_45_90.neu
input file format
0 = Uniform
1 = Structure
2 = Unstructure hybrid (GEMS-DFD)
21 = Unstructure hybrid (GEMS-DFD) with volume index
3 = VGRID
4 = CFDRC Mixed format
5 = GAMBIT neutral file format
6 = FIELDVIEW unstructured format(3D only**)
*** if you choose 2D for FV format, I will try to give you z= 0 surface grid
5
input the dimension of problem (1=1D,2=2D,3=3D):
3
volume type= 0
volume type= 1
volume type= 2
volume type= 3
nnodes = 1243460
ncells = 1232038
nfaces = 7392228
How many partitions? (if input <=1, just for convert to GEMS-DFD format)
```

Sorting control of the amount of the mesh elemets

**Figure 3.7.3.** The request of the PGRID® on the assigned amount of partitions of the computational domain.



```
kkolokol@gggspace101:~/CONDUCT/Step35/grid - Konsole - Konsole
File Sessions Settings Help
channel_45_90.neu
input file format
0 = Uniform
1 = Structure
2 = Unstructure hybrid (GEMS-DFD)
21 = Unstructure hybrid (GEMS-DFD) with volume index
3 = VGRID
4 = CFDRM Mixed format
5 = GAMBIT neutral file format
6 = FIELDVIEW unstructured format(3D only***)
*** if you choose 2D for FV format, I will try to give you z= 0 surface grid
5
input the dimension of problem (1=1D,2=2D,3=3D):
3
volume type= 0
volume type= 1
volume type= 2
volume type= 3
nnodes = 1243460
ncells = 1232038
nfaces = 7392228
How many partitions? (if input <=1, just for convert to GEMS-DFD format)
1
```

**Figure 3.7.4.** The assigned amount of partitions is 1 for the channel computational domain for this particular example.

### **3.8. Compatibility with postprocessor TECPLOT™ and CST – Complete Set of Tools.**

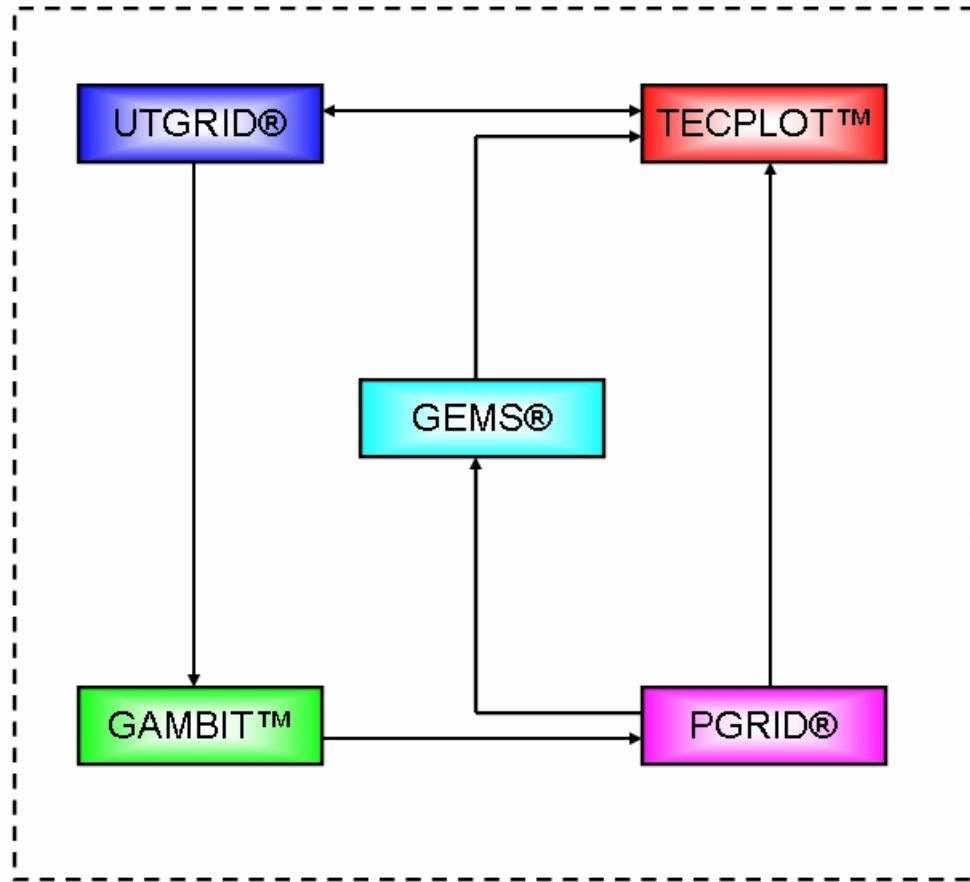
The programs described in previous section of the Thesis, together with the postprocessor TECPLOT™, represent the **Complete Set of Tools** for numerical simulations of the combustor-nozzle driven flow and the MHD and conductivity channels as well. Symbolically the **CST (Complete Set of Tools)** is represented in the diagram on **Figure 3.7.5** below.

Compatibility with TECPLOT™ was achieved by the actions:

- Generation of the nozzle wall contour can also be made with TECPLOT™ although in the present study it was made with GAMBIT™;
- Output of GAMBIT™ which is described in the section **3.6**, is accepted by TECPLOT™;
- The ASCII files for the first and the second nozzle wall contour derivatives are compatible with TECPLOT™.

PGRID® also additionally generates TECPLOT™ acceptable files. In summary, **CST** has the module structure and contains the set of programs: GEMS®, UTGRID®, GAMBIT®, PGRID® and TECPLOT™.

### CST - Complete Set of Tools



**Figure 3.7.5. Complete Set of Tools** for numerical simulation of the flows through a combustor-nozzle-extension, and an MHD and conductivity channel.

### **3.9. Application of CST to Mesh and Model the HVEPS Facility Flows.**

The development and details of the use of a suite of programs to model the flow fields in the components of the UTSI HVEPS facility have been presented in the previous sections of this Thesis. Because of the geometric complexity of these facility components, many computational domains had to be defined, not just for the internal flow fields, but also for the combustor, nozzle and channel walls (or solid zones) and for the air domain surrounding the facility. This led to the need for generation of modification files wherein the geometrical elements were re-named in order to put them in a sequential form that can be processed in a grid generation algorithm developed by the author and incorporated in UTGRID®, change to mnemonically understandable names to be attached to these geometrical elements (edges, faces and volumes) to make them easily understood by the user. Also, these numerous (very numerous, in the case of MHD and conductivity channels) computational domains had to interface one another appropriately. Therefore, the author presented the “user’s manual” of the last section in order to show graphically and pictorially the names and numbering of the edges, faces and volumes of the computational elements used to model each facility component.

But, that is not all that needs to be discussed. There is also the issue of obtaining satisfactory resolution of internal flow field details with an appropriate mesh configuration. Boundary layers, shock waves, and electric arcs, as well as diffuse electric currents, must be resolved with an acceptable mesh configuration. Therefore, the author also had to develop mesh schemes which obtained the necessary grid resolution in the appropriate flow field domains inside the combustor, nozzle, nozzle-extension, straight

conductivity channel and proposed MHD channel components, and, make these meshes interface properly at their mating surfaces.

To simplify the presentation, the meshing issues are presented and referred to the appropriate figures shown at the end of each subsection. Only important details are mentioned.

### **3.9.1. Mesh resolution scheme for the Combustor-nozzle Domain.**

This subsection of the Thesis discusses and attempts to clarify, using figures, some of the critical issues involving the development of adequate meshes for internal flow field resolution in the HVEPS facility component model and points out the most important details of meshing that impact the Combustor-nozzle-nozzle extension components.

- **Figure 3.9.1:**

This figure shows that a longer conical extension can be added to the nozzle computational domain for estimating boundary layer thickness and growth. Appropriate inlet and outlet values can be obtained in preliminary 2D flow simulation without any external magnetic field effect in order to use those values as input data for final 3D mesh generation. Input data for 3D MHD computations can be generated, if needed, by rotation of a 2D data array around 360 degrees angle around  $Z$  – axis. For example, a 2D array which could be taken (extracted) at the axial position marked on this figure.

- **Figure 3.9.2:**

This figure attempts to show the dense mesh region near the wall for boundary layer resolution and it also shows how UTGRID® automatically stretches grids across the

component interfaces (discontinuity points of the nozzle wall contour) of the full flow tract. UTGRID® allows flexible and controllable meshing based on the control parameters of its input file, and thus ensures that the necessary conditions for convergence of the solutions are met. UTGRID®'s input file contains three independent aspect ratios for mesh stretching using the Geometrical Progression, for the combustor, the nozzle and the channel extension areas, and also for the boundary layer of the nozzle profile.

- **Figure 3.9.3:**

This figure shows how the overall computational domain is factorized into sub-domains or partitions by PGRID®. Twelve (12) partitions are shown in this example. The partitions are shown with different colors. Parallel computation by GEMS® is performed on each partition or sub-domain independently. The nozzle extension can be chosen to interface with a straight channel, to estimate thickness of the boundary layer for inlet and outlet, or to a diagonal-wall MHD channel as well. As dated previously, if needed, input data for a 3D MHD computations can be generated by rotation of a 2D data array around 360 degrees angle around Z – axis. For example, a 2D array which could be taken (extracted) at the axial position marked on this figure.

- **Figure 3.9.4:**

This figure shows that the author used UTGRID® to stretch the mesh covering area immediately after the nozzle extension, or adaptor (the ceramic ring after nozzle), because of a geometric discontinuity of the nozzle contour or profile at this point, which represents the connection of the adaptor to the channel. The adaptor outlet is being stretched as well. The aspect ratio for a constant area nozzle extension, called

as channel imitation, is also available and independent of other aspect ratios and it is also controllable from the input file to UTGRID®.

- **Figure 3.9.5:**

This figure shows five sub-domain regions (sectors) where mesh stretching was required:

- Cylindrical section of the combustor with its conic section;
- Conic section of the combustor with the nozzle inlet;
- Throat corridor of the nozzle;
- Nozzle outlet with the nozzle extension or adaptor;
- Nozzle extension or adaptor with channel extension.

The longitudinal profile has a discontinuity at each interface between those parts. Note that PGRID® automatically chooses boundaries for the sub-domain partitions (shown with different colors). It should not be straightly defined, but it follows the algorithm for parallel computation.

The following five figures show the mesh stretching in the above five regions in a magnified scale.

- **Figure 3.9.6:**

This figure illustrates the independent aspect ratios that stretch the grids at the interface between the cylindrical section of the combustor and the conic part. Boundary layer thickness for the total profile was chosen independently, with an independent aspect ratio. For this case, it was chosen as **10E-7** in SI metric system (see appropriate input file in the **APPENDIX D**).

- **Figure 3.9.7:**

This figure shows the interface region between the conical and cylindrical parts of the combustor being axially stretched up to the beginning of the nozzle's inlet. The reader can see the computational boundaries that correspond to the partitions of this domain into sub-domains in more detail in this magnified area on the figure. Those boundaries are generated by PGRID®.

- **Figure 3.9.8:**

This figure shows the mesh stretching in the region of the contraction corner of the nozzle. Note that the input parameters for stretching can be changed to improve the convergence of the solution. The user can stop GEMS® and analyze the solution, then change the mesh aspect ratios and first rows in UTGRID®'s input file and repeat the mesh generation process. GEMS® can be run with new mesh to improve convergence if needed. Mesh factorization by partitions does not affect any mesh parameters obtained with UTGRID® and GAMBIT™. PGRID® just factorizes the computational domain and converts grid file to GEMS®'s acceptable format.

- **Figure 3.9.9:**

Experience has shown that it is crucial to stretch the throat corridor for the nozzle flow. This figure shows that the throat radius varies greatly in the throat region, and so mesh resolution in the boundary layer is very important. Appropriate radial rate to define the length of this corridor is controlled from UTGRID®'s input file (see section 3.5.3). During the computation, the code UTGRID® automatically chooses the axial locations for the points which define the throat corridor. All performed computations for the nozzle are represented in the source\_5 file in the **APPENDIX D**.

- **Figure 3.9.10:**

Another location with a discontinuity in the wall profile is the point of connection of the nozzle to its extension, or to an adaptor. This region is also being stretched independently with the same aspect ratio as mesh stretching inside the nozzle. The code UTGRID® applies mesh stretching to all four components, the combustor, the nozzle, the nozzle extension and the wall boundary layer of all components. The boundary layer was meshed with the same thickness in all parts of the total flow tract. The user can exclude adaptor and channel extension for the mesh generation as option in the input namelist in UTGRID® and mesh the domains only of the combustor and nozzle, if desired.

---

The figures described in this list are represented in **LIST OF FIGURES**, in **APPENDIX A2**, pages 285 – 295.

### **3.9.2. Straight conductivity channel.**

We now can come to two components of the HVEPS facility that are more difficult to model from the standpoint of generating mesh systems. These are the straight conductivity channel and the diverging-wall, MHD channel. These components are much more difficult to model and mesh because they contain many thin electrodes, that is, anodes and cathodes, or more properly anode and cathode “frames”, positioned in sequential parallel planes between the channel inlet and the channel outlet. These anode and cathode frames are the constant electric potential electrodes into which flow the electrical power extracted from the plasma flow, or from which electrical power is input

to the plasma flow, depending on the imposed external voltage scheme. The theory of MHD electrical power generators or accelerators is too complex to discuss here, but, the reader may refer to various references, for example [22, 23].

The problem with modeling and meshing these components using the approach based on the GEMS® code is that GEMS® calculates not only the internal fluid flow field and electromagnetic fields in the fluid, but also, GEMS® calculates the evolution and development of the electromagnetic fields in the solid duct walls and electrode frames, and, out into the free air environment surrounding the channel. Therefore, the mesh system must be stretched not only internally to the fluid flow (the plasma core and its boundary layer), but also the mesh must be stretched in the electrode frames and in the air domain outside the channel to allow convergence of the numerical scheme used by GEMS® to solve the Maxwell equations in these regions.

Because the meshing process is so complex, it was extremely important that UTGRID® be developed to allow for straight forward input of all the necessary mesh-stretching aspect ratios for the GAMBIT™ Geometric Progression Mesh Stretching Procedure. Note also that some MHD channels use slanted or diagonally inclined electrode frames [23]. Thus the author had to develop the procedures that would allow GAMBIT™ to develop and mesh multiple. Thin, vertical and diagonally-slanted electrode frames in either a straight walled, constant area conductivity, or in a diverging-walled MHD channel. These procedures have been incorporated into UTGRID®, so that for future geometric changes to HVEPS-type facility components, new mesh systems can be very easily generated for solution by GEMS®.

What follows in the present section are a group of figures that illustrate some of the meshing difficulties as well as meshing requirements that the author incorporated as control parameters into UTGRID®, and which GAMBIT™ subsequently used to construct very complicated 2D and/or 3D meshes of conductivity, and MHD channel components.

The list of the figures and their description represented in this section is:

- **Figure 3.9.11:**

This figure identifies some of the key zones or sub-domains in the plasma volume which have common boundary faces or interfaces with the electrode frames, and have two common boundary edges. These edges are the geometrical boundary edges for the plasma sub-domain volume and the electrode frame element volumes. Also, boundary layer resolution requires that the plasma radial aspect ratio be defined to properly mesh the boundary layer of the plasma. All plasma sub-domains were stretched corresponding to the boundary plasma face (see section 3.5, and look at the appropriate figures in the **APPENDIX A**). The overall grid distribution is defined by the both a radial mesh generation process, and an axial mesh generation scheme from inlet to outlet. UTGRID® defines first the mesh stretching requirements for the plasma boundary layer thickness where the boundary layer thickness is under control from the input file. Then, UTGRID® meshes all of the axial faces with a stretching process for each sub-domain volume respectively, with solid elements' length and a separate axial aspect ratio. Axial first row is under control from input file (parameter **axprec** in the section **ch\_rd** of the namelist of variables). UTGRID® automatically chooses the number of rows with its internal optimization. Cooper's algorithm is used

for all sub-domain volumes (cylindrical in this case). Boundary layer radial grid lines are perpendicular to the wall. The magnified sectors are shown as **1**, **2**, **3** and **4**, will be expanded, shown and discussed in subsequent figures.

- **Figure 3.9.12:**

This figure is a magnification of sector **1** on the previous figure and shows the mesh stretching used in the radial boundary layer for plasmas at the inlet plane and the axial mesh stretch process on the lateral surface of the plasma volume. UTGRID® uses independent aspect ratios, called axial and radial aspect ratios for the Geometric Progression mesh stretching process used in GAMBIT™. The axial ratio is also called the axial precision. They are in the input namelist of variables to UTGRID® and fully controllable from the input.

- **Figure 3.9.13:**

This figure is a magnification of the sector **2** on **Figure 3.9.10** and shows the axial stretch process on the plasma face with the axial aspect ratio for the areas on the face which are adjacent to the electrode frames (the conductors and insulators), which are also called electric elements.

- **Figure 3.9.14:**

This figure is a magnification of sector **3** on **Figure 3.9.10** and shows the axial stretch process for the grid distribution over the electric elements with constant axial aspect ratio (axial precision), which is the variable in the input namelist.

- **Figure 3.9.15:**

This figure is magnification of sector **4** on **Figure 3.9.10**, and shows the axial stretch process on the plasma face at the outlet.

The next section of figures will show the mesh details and the mesh stretching requirements of just the insulators or insulator disks that are sequentially stacked in the conductivity channel between the electrode frames. To clarify these components, the electrode solid surfaces and volumes are not shown. Only the plasma volume and the solid insulators are shown in the subsequent figures.

- **Figure 3.9.16:**

This figure shows the axial stretch process for the electric element faces (insulator volumes between the electrode volumes in a constant-area conductivity channel). The radial mesh distribution is also shown for the insulators. The aspect ratio for the axial distribution is the same as that for plasma. The radial aspect ratio for the electric elements is an independent parameter and is a variable in the input namelist of UTGRID®. Sectors **2** and **3** on this figure are magnified and described in subsequent figures below.

- **Figure 3.9.17:**

This figure is a magnification of sector **2** on the previous figure and shows the axial double sided (respective to the edge-boundary) stretch grid process for the electric element. The aspect ratio is much less at this example than the axial precision. They are independent inlet parameters.

- **Figure 3.9.18:**

This figure shows a magnification of sector **3**, **Figure 3.9.16**, and shows the axial and radial mesh configuration on adjacent insulator sub-domains. The mesh stretching in the axial and radial directions was achieved with different and independent aspect ratios.

- **Figure 3.9.19:**

The same algorithm used to generate mesh systems representing insulators was used for mesh generation for the corresponding set of solid conductor elements. Both insulator and conductor mesh representations use the same axial and radial precision in the axial and radial mesh distributions on the solid faces. The mesh aspect ratio for the plasma boundary layer regime was independent of those used to mesh the solid regimes, and was defined in the input namelist set. The same Cooper's algorithm was used for this zone (conductors) for volume mesh generation. The sectors **1**, **2**, **3**, and **4** are magnified and described below.

- **Figure 3.9.20:**

This figure shows a the magnification of sector **1** of **Figure 3.9.19**, and shows the radial mesh distribution in the boundary layer regime of the plasma cross-section, and the solid conductor wall interface at the inlet plane. The boundary layer regime and the conductor wall were meshed with equal but independent aspect ratios to achieve high resolution at the plasma wall interface.

- **Figure 3.9.21:**

This figure shows the magnification of sector **2**, **Figure 3.9.19**, and shows the axial mesh of the first conductor segment of the constant cross-sectional area conductivity channel.

- **Figure 3.9.22:**

This figure shows a magnification of sector **4**, **Figure 3.9.19**, and shows the axial mesh of the last (outflow) conductor segment of the constant area conductivity channel.

- **Figure 3.9.23:**

This figure shows an overview of the air domain that had to be meshed that surrounds the constant area conductivity channel. This domain has axial stretching that matches that of the conductivity channel segments, but does not need any external boundary layer regime to be represented, except that which stretches radial mesh respectively to the external boundary of the solid conductor and insulator elements (one sided radial mesh stretching). The author would like to point out that for straight wall geometry of conductivity channel boundary layer regime is equivalent to mesh stretching process for the air domain (radial grid lines of the air mesh are perpendicular to the solid boundary face), but for diagonal wall geometry it is not true, and non-perpendicularity can be sufficiently compensated by the appropriate radial aspect ratio for air mesh. The difference between the straight and diagonal wall geometry the reader can see in section 3.9.3. Again this external air domain is required because the GEMS® code calculates the developing electro-magnetic field completely through the encompassing the channel.

- **Figure 3.9.24:**

This figure shows a magnification of sector 1, **Figure 3.9.23**, and shows the radial mesh for plasma flow, the conductor, and surrounding air domain. The radial mesh stretching was done with three equal but opposite and independent aspect ratios for the radial grid generation process.

- **Figure 3.9.25:**

This figure shows a magnification of sector **2**, on the **Figure 3.9.23**, and shows on the front side the radial mesh stretch process for the air zone at the inlet, and on the top surface, the axial stretch process at the inlet plane.

- **Figure 3.9.26:**

This figure shows a magnification of sector **3**, on the **Figure 3.9.23**, and shows that the axial mesh stretching and grid generation become matches with corresponding solid elements for the conductivity channel, the conductors and insulators.

- **Figure 3.9.27:**

This figure shows a magnification of sector **4**, **Figure 3.9.23**, and shows the axial mesh and stretch process for the air domain at the outlet.

- **Figure 3.9.28:**

This figure shows cut through the entire meshed domains needed to represent the constant cross-sectional area conductivity channel. The mesh was generated using finite elements in TECPLOT™. The author would like to point out that boundary faces can not be continuously blanked which illustrates the restriction for finite elements representation with TECPLOT™.

- **Figure 3.9.29:**

This figure shows a radial projection of the geometry and mesh distribution of the inlet plane. This projection clearly shows how UTGRID® stretches the mesh radially across the radial faces. For the straight conductivity channel it was not important to define very precise mesh for the air and therefore the air domain was obtained with significantly larger meshed elements. For the radial components it was better to

choose equal but opposite aspect ratios for the plasma and solid mesh distributions, to have the same length of the first row in both stretch and boundary layer regimes.

---

The figures described in this list are represented in **LIST OF FIGURES**, in **APPENDIX A3**, pages 296 – 314.

### **3.9.3. Diagonal MHD channel domain.**

#### **3.9.3.1. Introduction.**

The next subsection of the Thesis describes the meshing of elements or sub-domains of a diagonal conducting wall (DCW), magneto-hydrodynamic (MHD) power generation channel. This computational model does not represent physical hardware of the UTSI HVEPS facility. Such a channel would be designed using engineering design codes that would predict the optimum loft (expansion) of the channel based on plasma velocity and conductivity, and on the field created by a magnet surrounding the channel. Magnetic field strength of the magnet would depend on whether it is iron-cored or a superconductor wound magnet. These details and design have not, at present, been developed for the HVEPS facility.

However, it was thought worthwhile to go ahead and create a mesh system that would model a DCW MHD generator. This meshing process exhibits many difficult geometrical features including diagonal or slating solid conductors and insulators. Moreover, the cross-sectional plasma flow area must increase with axial distance to satisfy the static pressure recovery of the power extraction process by MHD forces. Thus, from a geometrical modeling viewpoint, the creation of suitable mesh that would allow the

GEMS® Code to calculate the electro fluid dynamic process in an MHD channel was considered a formidable task. Never the less, this difficult task would allow the usefulness of UTGRID® to be demonstrated on a challenging project.

The data input file of UTGRID® was modified to allow the input of meshing variables needed to create a multi-electrode, DCW MHD generator, with inclined or slanted electrodes. The same issues that were faced with the straight conductivity channel were also faced in the modeling of the DCW generator, namely

- boundary layer resolution
- multiple, thin conductor and insulators stacked in the axial direction
- the need to model the air domain surrounding the channel.

The resolution of the meshing problem for a DCW MHD generator is discussed below with very brief descriptions of the technical features of the resulting mesh geometry for the entire domain (plasma volume, solid volumes, and external air volume or domain). Specific sub-domain features and details are described and shown in corresponding figures.

### **3.9.3.2. Generalization of the DCW Generator Electrode Slant Angle.**

In the DCW MHD generators, the slant angle of the electrode is imposed in order to take advantage of the Hall current, the axial component of the electric current. Thus, the total current vector is not perpendicular to the axis of flow, but is inclined to it at an angle, which is typically the angle of incline of the electrodes in a multi-electrode DCW generator. This geometry results also from the typical orientation of the fixed magnetic field passing through the plasma; it is usually oriented perpendicular to the plasma bulk

velocity vector which is aligned with the axes of the channel. The author decided to generalize the allowed inclination angle of the electrodes to include two slant angles, or actually, a slant angle and an angle of rotation of the electrode about one of its axes. This generalization may, or may not, permit optimization of the DCW MHD generator to produce power in the event of a non-axially aligned plasma velocity field.

However, the author is sure that future computational studies should be done to investigate whether any benefit accrues from generalizing the inclined orientation of electrodes in DCW and similar generators. In any case, when presenting grid and mesh data that corresponds to inclined electrodes, two angles are specified for these mesh sub-domains: the primary DCW electrode angle that corresponds to the angle quoted in papers and references on DCW generator theory, and a second angle which permits rotation of the inclined electrode with respect to the imposed magnetic field and plasma velocity field. The author also believes that further generalization of DCW MHD conception allows considering the external magnetic field direction, together with the second slope angle, as flexible parameter for power generation optimization.

---

The list of the figure descriptions represented in this section is as follows:

- **Figure 3.9.30:**

This figure shows the plasma volume or domain which consists of the sub-domains that interfaces with and corresponds to the solid electric element sub-domains. The mesh and grid stretching process was similar to those used to represent the straight channel for radial and for axial faces, and volume grid distributions. UTGRID® was used to create radial boundary layer regimes in the plasma near the solid surfaces.

Axial and radial grid stretching in the boundary layer region was achieved with independent aspect ratio input parameters. The user assigns thicknesses of the boundary layer regimes in the UTGRID® input file, and the code automatically calculates the first and the last row of the mesh domain. The author would suggest choosing the value of the aspect ratio such that the last row in the mesh of the inlet plasma face is comparable to the length of the arc defined by two adjacent grid points in the round edge (see section 3.5.6) at the inlet, which can be easily estimated. This is important so as not to have too big a jump on the radial grid line. As for the straight channel, the radial and axial grid generation process defines the mesh generation process for the volumes. The primary difference for the diagonal channel was establishing the skew angle of the finite elements representing the slanted electrodes. The quality of this effort will be described later in the next section. The sectors 1, 2, 3 and 4 of **Figure 3.9.30** are magnified and represented on four figures that follow. Algorithmically the grid generation process is similar to one for the straight channel except that the notation “radial” is used for the edges and faces in the plane inclined to the axial Z – axis (primary and secondary slope angles) and the notation “axial” is used for the edges and faces on other plasma boundary and solid faces.

- **Figure 3.9.31:**

This figure represents a magnification of sector 1, **Figure 3.9.30**, and shows the radial mesh for resolving the boundary layer in plasma, and also the axial mesh stretching process on plasma faces. The axial mesh distribution is a double-sided process, respectively, to the appropriate edge-boundaries.

- **Figure 3.9.32:**

This figure represents a magnification of sector **2**, **Figure 3.9.30**, and shows the axial stretch process for this sector on the plasma faces.

- **Figure 3.9.33:**

This figure represents a magnification of sector **3**, **Figure 3.9.30**, and shows the axial mesh stretching process for this sector.

- **Figure 3.9.34:**

This figure represents a magnification of sector **4**, on the **Figure 3.9.30**, and shows the axial mesh stretch process for the outlet on plasma at the channel outlet. The main difference between the DCW channel axial mesh and that of the straight channel is the skew of the cells on the surfaces and volumes.

- **Figure 3.9.35:**

This figure shows the insulator electric element extending outward from the plasma volume, and shows the radial mesh stretch process for the insulators, where grid line are not perpendicular to the boundary faces. Resolution can be maintained by choosing the appropriately smaller aspect ratio for radial mesh distribution and by the axial aspect ratio (axial precision) as well. The axial mesh stretch process goes with the same axial aspect ratio as for the plasma faces and with the same algorithm. As for the straight channel, the axial and radial aspect ratios are independent input parameters and controllable from the input file namelist. This figure shows the sectors **1**, **2**, **3** and **4** for magnification. However, it is only necessary to magnify sector **2**, which is represented and described on the next figure.

- **Figure 3.9.36:**

This figure represents the magnification of sector **2**, **Figure 3.9.35**, and shows the axial mesh stretch process for the plasma volume, and for the radial mesh of the insulators, obtained with independent aspect ratios.

- **Figure 3.9.37:**

This figure shows the outer surface of the meshed DCW generator with two groups of solid electric elements (conductors and insulators), and the inlet plane containing both the plasma and the solid element (conductor). The sectors for magnification are: **1**, **2**, **3** and **4**, which are represented and described consequently on figures **3.9.38 – 3.9.42**.

- **Figure 3.9.38:**

This figure represents the magnification of sector **1**, **Figure 3.9.37**, and shows radial mesh distribution for the plasma and the solid and axial mesh stretch process on the solid at the inlet plane. All three aspect ratios were independently chosen input parameters. The visual distortion of the grid lines on the inlet solid face is just the graphical effect of image representation; they are actually and appropriately straight.

- **Figure 3.9.39:**

This figure represents the magnification of sector **2**, **Figure 3.9.37**, and shows the axial mesh stretch process for inlet conductor which is the longest solid volume together with the outlet conductor, and the first adjacent insulator. They have been generated with the same axial aspect ratio.

- **Figure 3.9.40:**

This figure represents the magnification of sector **3**, **Figure 3.9.37**, and shows the axial mesh with a double-sided mesh stretch process for the internal solid elements, based on the same aspect ratio.

- **Figure 3.9.41:**

This figure represents the magnification of sector **4**, **Figure 3.9.37**, and shows the axial mesh stretch process at the outlet for the last solid electric element (conductor).

- **Figure 3.9.42:**

The air zone for the DCW MHD channel consists of one volume which is defined by all axial air faces and inlet and outlet radial air faces. The “compensators” (the elliptic air edges) are extremely important for modeling the diagonal MHD channel. They allow finalizing the mesh generation process as a necessary condition. Without these air edges the mesh generation process for the air could not be done with the chosen grid solver of GAMBIT™ (GENERIC). The axial mesh stretch process on the air faces was done with the same axial aspect ratios as for corresponding plasma and solid faces, and then for volumes. The grid lines are not perpendicular to the solid faces for the air domain, but they were compensated for establishing sufficiently smaller radial aspect ratio for the air. The radial mesh stretch process for the air was one sided, which is reasonable for the boundary condition formulation in MHD numerical modeling. The direction of external magnetic field is parallel to the radial air grid lines in ZY – plane which represents one of the very important conditions for convergence of the solution. This is very important notice. However, further generalization of DCW MHD conception together with secondary slope angle allows changing the magnetic field direction. This figure has sectors **1**, **2**, **3** and **4** noted for further magnification and description.

- **Figure 3.9.43:**

This figure represents the magnification of sector **1**, **Figure 3.9.42**, and shows the radial mesh representing the boundary layer for the plasma, the radial mesh stretch process for the solid, and the radial one-sided mesh stretch process for the air face at the inlet plane. The radial stretching was based on three equal, but independent radial aspect ratios, which are independent input parameters.

- **Figure 3.9.44:**

This figure represents the magnification of sector **2**, **Figure 3.9.42**, and shows the last row in the one sided radial mesh stretch process for the air inlet and the two-sided stretch distribution on the faces (“**left**” and “**right**”, see section **3.5**) adjacent to the inlet elliptic air boundary edge.

- **Figure 3.9.45:**

This figure represents the magnification of sector **3**, **Figure 3.9.42**, and shows the axial mesh stretch process on the air faces obtained with the same algorithm as the axial, double-sided stretch process for the solid electric elements.

- **Figure 3.9.46:**

This figure represents the magnification of sector **4**, **Figure 3.9.42**, and shows the axial stretch process on the air faces at the outlet boundary.

- **Figure 3.9.47:**

This figure shows a cut plane for a slice through the plasma zone, at the inclined angle of 45 degrees along the boundary face of a diagonal electric element for one slope MHD channel. The next figure, **Figure 3.9.48** will show the plasma mesh geometry on this cut plane.

- **Figure 3.9.48:**

This figure shows plasma volume mesh in the cut plane of **Figure 3.9.47**, which represents the boundary face of a volume contained by a solid electric element and the adjacent insulator. It is very important that the algorithm of UTGRID® generates the same grid quality at this plane as it does for the perpendicular inlet and outlet planes of the channel. Tests showed that for a one slope and a double slope channel, and for the straight channel, the grid quality remained the same. The issue of grid quality will be discussed in the next section.

---

The same algorithm of UTGRID® creates the mesh for the straight constant area conductivity channel, for diagonal wall conductivity channel and for the DCW MHD channel. The author would suggest analyzing the inlet geometrical and mesh parameters first to compensate the diagonal skews, as sufficiently as possible and reasonable for getting the converged solutions in MHD numerical simulation. The mesh quality, which is necessary condition to get the converged solution, will be discussed in the next section of this Thesis.

The figures described in the list above are represented in **LIST OF FIGURES**, in **APPENDIX A4**, pages 315 – 334.

#### **3.9.4. Mesh quality for equiangular skews.**

In computational fluid mechanics, it is possible to represent a computational domain, such as a flow field region, with many different, possible mesh configurations. The primary objective in meshing a computational domain is to obtain a mesh that results in proper resolution of the object being modeled with as few mesh points, or finite elements,

or finite volumes, as possible, thus providing for sufficient computational accuracy while minimizing computational resources required. This objective, and techniques for obtaining it, have been the subject of general discussion and analysis in the field of computational modeling since its beginning. The general topic is well beyond the scope of the present thesis. An overview of this problem is covered in many references, for example, reference [24].

In the present study, the quality of the mesh representations of the components of the HVEPS facility is evaluated by use of some tools provided by GAMBIT™ to assess the “goodness” of regions in the meshed domains. These “goodness” criteria simply refer to the quality of the geometry of the mesh used to represent the corresponding physical domains. In this sense, mesh quality has a very narrow definition. It does not refer to the overall quality or goodness of the global flow field and electromagnetic field solutions obtained ultimately with the GEMS® code. Rather, it refers to local mesh element geometry that is known from practical experience to provide for sufficiently accurate numerical resolution on the local mesh level, and also tends to ensure that convergence of numerical solutions on the local mesh would be reasonably well guaranteed. Consequently, these local mesh element goodness criteria in GAMBIT™ are specifically related to acceptable values of the mesh geometrical parameters listed below.

The GAMBIT™ allows checking for grid quality using the following mesh element criteria:

- aspect ratio
- diagonal ratio
- edge ratio

- equiangular skew
- equisize skew (uses original notation)
- middle angle skew
- taper
- volume

All criteria in this list are described in the reference manual of GAMBIT™. On the figures represented and described below grid quality is represented by the skew criterion for mesh regions of the straight conductivity channel and the DCW MHD channel. The degree of the “goodness” of the mesh is scaled by the color in GAMBIT™’s graphical interface and ranges from “blue”, the highest quality or best mesh configuration, to “red”, which is the lowest quality. Better mesh quality can be obtained with more grid points on the circumference of the channel, with the smaller aspect ratios, and with a smaller axial precision. The user can create a variety of grid files, trying to reach a compromise between the grid quality and the total amount of the finite elements for computation. The problem for the user is that computational time can dramatically increase with an increase in the number of mesh elements in the grid file, which might not be reasonable in many cases. However, increased numerical resolution is often required to achieve convergence of the numerical solution. The user has choices for the computational experiment. For any possible questions about the definitions of the criteria of mesh element quality, refer to the [GAMBIT™](#) user’s manual.

The three figures presented below show mesh representations of an actual HVEPS facility component, the constant area conductivity channel, and two hypothetical or future components, which represent DCW MHD channels. The two diagonal wall channels are

of the some basic channel design, but one mesh representation is based on a single incline angle for the conductors and insulators, denoted to the one-slope channel, and the other mesh representation is based on two reference of slope angles for the conductors and insulators, and is called the two-slope channel. The purpose of presenting these figures is to illustrate the level of mesh quality achievable in the most complex mesh configurations required to represent HVEPS facility components. The mesh elements displayed are those representing the plasma flow, the solid conductor and insulator walls, and the air domain surrounding the channels. Thus, three colors are indicated for mesh (not for mesh quality!) in each of these figures: the mesh color for the plasma region turns out to be “yellow”, the color of the insulator sub-domains turns out to be “magenta” and the color of the conductor sub-domain turns out to be “blue” for each case shown. The reader has to distinguish the mesh quality color and mesh representation color. This is very important issue for understanding mesh quality in figures shown below. Thus, the reader can see that the color which represents mesh quality (separated from the color which represents the mesh) is in the middle of the scale for the most skewed mesh elements, between “blue” and “red”. That means that the degree of “goodness” of the mesh is better than average for the most of skewed mesh finite elements. The best mesh quality can be achieved for the constant area conductivity channel (straight channel).

Note again, the mesh quality color is the color of mesh finite elements, and the mesh representation color is the color of the meshes or grid lines.

The list of figures and their description in this section is:

- **Figure 3.9.49:**

This figure shows the grid quality for constant area conductivity channel. The quality is the best for this channel because of the straight channel geometry. Mesh color (not mesh quality color!) for plasma is “yellow” in this example and mesh color for insulators is “magenta”. Mesh color for air is “cyan”. The upper picture in this figure shows the mesh finite elements on ZX – plane and second picture on this figure shows the finite mesh elements on ZY – plane. Slope angles in this example: 90, 90 (straight channel). The color of finite mesh elements is mostly “blue” (the best quality).

- **Figure 3.9.50:**

This figure shows the grid quality for the one slope DCW MHD channel. The quality is better than average on the color scale. Do not mix the color mesh with the color of the finite elements. Thus, mesh representation color for plasma is “yellow” in this example, mesh representation color for insulators is “magenta” and mesh representation color for air is “blue”. The upper picture in this figure shows the mesh finite elements on the ZX – plane, and lower picture on this figure shows mesh finite elements on ZY – plane. Slope angles in this example: 45, 90, respectively to primary and secondary angles.

- **Figure 3.9.51:**

This figure shows the level of grid quality for the two-slope DCW MHD channel. The mesh quality is better than average on the color scale (mesh quality color is more “blue”). As stated previously, mesh representation color for plasma is “yellow”, mesh representation color for insulators is “magenta” and mesh representation color for the air is “blue”. The upper picture in this figure shows

the mesh finite element on  $ZX$  – plane, and lower picture on this figure shows the mesh finite elements on the  $ZY$  –plane. Slope angles in this example: 45 and 60 degrees respectively to primary and secondary angles.

---

The figures described in the list above are represented in **LIST OF FIGURES**, in **APPENDIX A5**, pages 335 – 338.

## CHAPTER 4

### NUMERICAL RESULTS WITH GEMS® AND UTGRID®

#### 4.1. Combustor-nozzle turbulent flow.

This chapter of the thesis displays some numerical results obtained using the **Complete Set of Tools (CST)** for solving for the flow fields in the HVEPS facility components. The **CST** consists of UTGRID®, GAMBIT™, PGRID® and GEMS® for obtaining the solutions and TECPLOT™ for representing the solutions graphically.

Detailed descriptions of these flow fields will be provided elsewhere in other papers and reports. The main focus of the descriptions of the flow field solutions presented herein will be on the implications for, and the need for mesh resolution in specific flow field regions. Also, the interfaces between the components of the HVEPS facility numerical models will be specifically identified in order to show the dependence of the flow field solutions for the channel on the flow field solutions for the combustor and nozzle upstream.

The input data for the numerical results represented and described below are in the file ‘gems\_input\_1’ in **APPENDIX B**. Note, that absolute pressure must be calculated as the gage pressure plus reference pressure which was specified as input parameter for GEMS® code computation. Outlet pressure for non-slip boundary conditions was specified with zero pressure gradient.

The list of figures and their description in this section is as follows.

- **Figure 4.1.1:**

This figure shows the Mach number distribution in the combustor outlet, the nozzle, the nozzle extension section or adaptor and the channel extension. The figure shows that the Mach number reaches supersonic values of about 2.8 in the nozzle exit region. The nozzle extension together with the channel extension acts as a short supersonic diffuser for the nozzle. The channel extension can also represent a conductivity channel without any applied external voltage or current extraction, for data comparison to the simulation results obtained with the constant area conductivity channel.

- **Figure 4.1.2:**

This figure shows an expanded or magnified view of the Mach number distribution in the nozzle and channel extension. Boundary layer growth and thickening is indicated near the wall. Some degree of supersonic diffusion is evident near the axis of the extensions. Note, that it is very important to obtain an adequate representation of the nozzle wall boundary layer since the entire wall-to-axis flow field represents the inlet data for the conductivity channel or DCW MHD channel calculation. The accuracy and resolution of the nozzle wall boundary layer profile will affect the stability and the rate of convergence of the channel flow field solution.

- **Figure 4.1.3:**

This figure shows the Mach number distribution in the combustor outlet and nozzle, where the nozzle extension or adaptor (ceramic ring) was attached, where the channel extension was not. This calculation was made with a reduced chamber pressure,  $P = 4$  Bar, thus, the Mach number distribution in the supersonic nozzle is indicating some

influence of the exit ambient pressure. To show a clear presentation of the effect of external ambient pressure on the nozzle Mach number distribution, an expanded or magnified is presented below.

- **Figure 4.1.4:**

This figure shows a magnified view of the nozzle Mach number distribution. It clearly shows the effect of external pressure feed back upstream through the nozzle wall boundary layer, causing compression in the interior of the nozzle flow field. Again, adequate mesh resolution of the wall boundary layer is required in order to have realistic flow field solutions inside the nozzle and for any downstream component such as the conductivity channel.

The figures in the range **4.1.5 – 4.1.9** will be described in the text below.

The following figures present the static pressure distributions and the static temperature distributions that correspond to the Mach number distributions just presented. **Figure 4.1.5** for static pressure corresponds to **Figure 4.1.1** for Mach number. **Figure 4.2.6** for static pressure distribution corresponds to **Figure 4.1.3** for Mach number distribution. **Figure 4.1.7** for static pressure distribution corresponds to **Figure 4.1.4** for Mach number distribution.

The static temperature distributions that correspond to the previously discussed Mach number distributions are as follows. **Figure 4.1.8** for static temperature corresponds to **Figure 4.1.1** for Mach number. **Figure 4.1.9** for static temperature corresponds to **Figure 4.1.3** for Mach number. Finally, **Figure 4.1.10** for static temperature distribution corresponds to **Figure 4.1.4** for Mach number distribution.

Since the main results of these computational studies presented with the discussions of the Mach number distributions, no further comment will be provided here. It should be noted that these solutions were obtained by the GEMS® code after numerical residuals had decreased to values of about 1E-08 in the numerical solution process. The solutions were obtained in an axially-symmetric coordinate system and hence represented 2D calculations. Input data are in the GEMS® input file ‘gems\_input\_2’ in appendix, **APPENDIX B**.

---

The figures described above in this section are represented in **LIST OF FIGURES**, in **APPENDIX A6**, pages 339 – 349.

#### **4.2 Conductivity Channel.**

The following set of figures shows selected results obtained from numerical calculations, made using the suite of codes (CST) assembled by the author, of the constant cross-sectional area conductivity channel, and the one-slope diagonal conducting wall (DCW) magneto-hydrodynamic (MHD) channel. The constant area conductivity channel was modeled as an axisymmetric system, when it is in physical reality. Therefore, its inlet conditions could be taken directly from the previously obtained nozzle exit plane solutions, or they can be defined arbitrarily for parametric studies.

On the other hand, the numerical model of the MHD channel is fully three-dimensional. Their channel inlet conditions have to be defined over an inlet plane. In the present study these channel inlet conditions were set arbitrarily and uniformly with selected values of pressure, temperature and Mach number. However, the inlet conditions for the numerical

model of the representative HVEPS DCW MHD channel should to be obtained from the 2D axisymmetric nozzle exit solutions by an interpolation process based on rotating the 2D axisymmetric solution through 360 degrees to give a full, 3D symmetric inlet condition in terms of plasma Mach number, pressure, temperature, and density profiles, including the wall boundary layer regimes, and this is the approach recommended for future studies.

Since detailed features of the channel solutions are provided elsewhere, reference [1], only an overview of the numerical solutions of the plasma flow, the magnetic and electric fields, and the electric current distributions obtained will be presented in this thesis. It should also be noted that the plasma and MHD flow field solutions obtained with the GEMS® code were based on an ideal perfect gas model, with simplified electric current conductivity and electric current models. Therefore, these solutions must be regarded as preliminary results until the GEMS® code is upgraded with a more physically accurate gas model, electrical conductivity model, and electric current model. The reader should refer to reference [1] for further details.

The first set of figures below will refer to the constant cross-sectional area conductivity channel. This device has been built and tested in the UTSI HVEPS facility. At present, the experimental data is still subject to interpretation, and suffers from data measurement uncertainty. Therefore, experimental data will not be presented in this thesis compared to the calculations made with GEMS®.

However, since the system modeled in GEMS® was axisymmetric, the GEMS® results were directly compared to equivalent calculations made with the UTSI version of the MACH2® code. The MACH2® code, reference [25], is a code developed to represent a

class of inviscid 2D planar and axisymmetric plasma and MHD flows in simple geometries, such as in rail-gun arc, reference [26], a general discussion of the MACH2® code and its modeling capabilities is also, beyond the scope of the present thesis. Therefore, the GEMS® - based solutions will only be qualitatively compared to corresponding MACH2® – based solutions to indicate their nominal agreement.

One final remark should be made before presenting the calculated results for the channels. Some of the figures described below present calculated current contour data. The current contours were obtained by the application of Maxwell laws to relate electromagnetic field geometries to current flux. For example in an axisymmetric system, Maxwell's laws in magnetostatics give [27]:

$$\oint_{C(R)} \vec{B} \cdot d\vec{l} = \frac{2\pi R B_{\theta}}{\mu_0} = I(R) \text{ amps} , \text{ where}$$

$C(R)$  represents the circular or circumferential path at a radius  $R$  from the centerline. Note that the value of  $I(R)$  at the wall of the conductivity channel should represent the total current input to the conductivity channel. A current contour, therefore, is a level surface of current, that is, the contour represents a fixed amount of current flowing axially, encompassed between the centerline of the conductivity channel and the level surface.

Even though the color data represents current contours, the plots of current contour in the ZY – cross-section of the conductivity channel also indicate the current concentrations and paths taken by the current from inlet to outlet.

Furthermore, in a conductivity channel without any imposed external magnetic field, the magnetic field induced by the forced current flow, due to the applied voltage, can be calculated from the current flow. This allows for a check of the ability of the GEMS®

code to predict the resulting magnitude of the relatively weak, steady-state magnetic field against the known total current flow.

The list of figures and their descriptions in this section are provided below.

- **Figure 4.2.1:**

This figure shows the predicted current density,  $J$ , in the cross-sectional  $ZY$  – plane of the HVEPS conductivity channel. The current flows from the first upstream conductor to the conductor electrodes downstream. This conductor – insulator configuration corresponds to the set up shown in **Figure 1.2.3** in section 1.2 on the HVEPS facility components.

- **Figure 4.2.2:**

This figure shows the same as **Figure 4.2.1**, with the zones representing the solid conductors included for clarification.

- **Figure 4.2.3:**

This figure represents the  $ZY$  – slice of the 3D current distribution in the straight conductivity channel with the axial symmetry. Current contours are represented by constant color levels or surfaces. Solid electric elements and boundaries are included so that their relationship to current flow is clearly shown.

- **Figure 4.2.4:**

This figure also represents the current contour distributions in the  $ZY$  – plane made with viscous GEMS® code in 3D for comparison to the results made with the 2D inviscid MACH2® code, which are described in the description of **Figure 4.2.5**. Close inspection of the GEMS® results (**Figure 4.2.5**) shows reasonably good

qualitative agreement. Quantitatively, there are differences near the electrode surfaces. Detailed comparisons of these solutions will be made in future studies.

- **Figure 4.2.5:**

This figure displays the current contours in 2D computations made with MACH2® for the comparison with the GEMS® results in **Figure 4.2.4**.

---

The figures described in the list above are represented in **LIST OF FIGURES**, in **APPENDIX A7**, pages 350 – 354.

#### **4.3 Diagonal MHD power generator.**

The next set of figures presents results obtained using **CST** to predict flow, plasma, electric current density and electromagnetic field conditions in representative DCW MHD generator, operating in an imposed, 2 Tesla external magnetic field.

These results were obtained in a collaboration of the author with his colleagues, Dr. Dennis Keefer, Dr. Charles Merkle, and Dr. Ding Li. The results have been more completely presented and discussed by these authors [1]. The results are presented herein, only briefly, to illustrate the complexity of mesh generation for these kinds of physically and geometrically complex flows.

Inlet boundary conditions for all solutions below were as follows:

- $T = 2940$  K for temperature;
- $P = 4$  Bar for pressure;
- $M = 2$  for Mach number;
- $B = 2$  T for B-field.

Two different axial distributions for the externally imposed B-field were modeled, in an attempt to gauge the effect of B-field on plasma flow properties. The B-field distributions will be discussed below.

The list of figure descriptions is provided below:

- **Figure 4.3.1:**

This figure shows the predicted plasma flow density distribution in the DCW MHD generator, corresponding to the Mach number, pressure, and temperature distributions described below. It is included for reference.

- **Figure 4.3.2:**

This figure shows the predicted Mach number distribution made with GEMS® code in the plasma flow of a diagonal conducting wall, one-slope, magneto-hydrodynamic generator inside an imposed external magnetic field. The solution shows considerable Mach number reduction in the boundary layer but it increases in the core of the plasma flow in the downstream part of the channel. The reason for the increase in core Mach number is that the effect of channel divergence (acceleration) is greater than the effect of the electrical power (energy) extraction (deceleration) from the flow by the conductor walls. The B-field orientation with respect to the flow field is shown in the figure.

- **Figure 4.3.3:**

This figure shows the predicted static pressure distribution in the DCW MHD generator that corresponds to the Mach number distribution described above. Note that the static pressure is decreasing in the flow direction due to the Mach number increase and the power extraction.

- **Figure 4.3.4:**

This figure shows the plasma static temperature distribution in the DCW MHD generator that corresponds to the Mach number and static pressure distributions shown previously. Although static temperature is decreasing in the core region of for reasons described above, however, the plasma's temperature stays fairly hot in the boundary layers near the wall. The boundary layer temperatures are high because they are recovery temperatures of the flow, nearly equal to the stagnation temperature of the flow, which is approximately 5300 K, at its highest value.

- **Figure 4.3.5:**

This figure shows the B-field distribution in the mid-plane of the DCW MHD generator. This externally-imposed B-field has a linear ramp-up to constant value, followed by a linear tail-off or ramp down to zero. The B-field is weak in the central plasma flow region but grows in strength out near the walls.

- **Figure 4.3.6:**

This figure shows the other B-field distribution utilized in the study, also in the mid-plane of the DCW MHD generator. This externally-imposed B-field has an exponential ramp-up to constant value, followed by an exponential tail-off or ramp down to zero. This B-field is also weak in the central flow region but grows in strength out near the walls. It is included for comparison with previous B-field distribution.

- **Figure 4.3.7:**

This figure shows the current density distribution and also tangential component of the current vector in the ZY – plane of a DCW MHD generator made with the

GEMS® code. Note, how current density increases on the leading and traveling edges of the conductors, and in the plasma near these points. This kind of current bunching or concentration is a serious technical issue for the design of MHD channels, and this demonstrates the need for computational modeling of these complex flows, when either designing such systems or analyzing their behavior and performance. A portion of the total current travels across the plasma to enter the solid conductors on the opposite side. There is also evidence that a substantial amount of the current simply enters the plasma, jumps across the dielectric slab and re-enters the next conductor downstream of the conductor it left. Grids that used for computation were pretty coarse, and author has a hope that an opportunity to perform future calculations with finer meshes will be given to him.

- **Figure 4.3.8:**

This figure also shows the predicted electric current density contours in the ZY – plane of a DCW MHD generator. Again, this figure, like the previous figure, illustrates the concentration of current near the leading edges of the anode conductors and the trailing edge of the cathode conductors. In general the current has both an axial (Hall) component, and a transverse (Faraday) component. For conductors with diagonal angles of 45 degrees, the axial and transverse currents are about of equal magnitude.

In any case, details of these results are presented and described elsewhere [1]. The specific point of presenting these figures in this thesis is to illustrate the need for great numerical resolution in repeated and relatively small regions or domains of the plasma flow. The **Complete Set of Tools (CST)** assembled by the author greatly

facilitates the creation or generation of new and/or modified meshing systems for the solution of complex flow fields in geometrically very complex configurations such as a DCW MHD channels.

- **Figure 4.3.9:**

This figure shows a magnified sector **Figure 4.3.8** for purpose of clearly showing current vectors. It is the last figure in the set of figures above. It is hoped that the figures presented in this section convey the reader the importance of UTGRID® to future computational modeling of HVEPS facility components and related facilities.

---

The figures described in the list above are represented in **LIST OF FIGURES**, in **APPENDIX A8**, pages 355 – 364.

## CHAPTER 5

### SUMMARY AND CONCLUSIONS

The suite of codes was assembled by the author to allow efficient and accurate numerical modeling of MHD Power Generator processes in the UTSI HVEPS facility. The codes together form a **Complete Set of Tools (CST)**, which was comprised of UTGRID®, created by the author, GAMBIT™ which is part of a set of codes offered commercially, PGRID® created by Dr. Ding Li, and the GEMS® code created by Dr. Charles Merkle and Dr. Ding Li. UTGRID® ties all of the other codes together in a seamless process of two and three dimensional mesh generation for numerical modeling of turbulent, compressible plasma flow in a combustion-driven MHD facility developed at UTSI for the HVEPS project.

UTGRID® was based on the contemporary programming architecture and the modern module structure of Fortran 90, which allowed the creation of a flexible, general and automated algorithm for mesh generation, not only for HVEPS project, but for other uses. UTGRID® was specifically designed to create grids acceptable to GEMS®, though UTGRID® can be used in wide spectrum of computational areas by adding new program modules for specific domains. Detailed analysis of current density, current vectors and plasma properties, in both the core flow and boundary layer regions of both a conductivity channel and an MHD became possible only with the **CST** codes at UTSI.

The **Complete Set of Tools (CST)** is capable of numerical CFD simulation in MHD flows as well as a spectrum of other physical areas. GEMS®, GAMBIT™, PGRID®, UTGRID® and also TECPLOT® give the possibility for numerical simulation and analysis for very complicated problems, and helps ensure that converged solutions are obtained for coupled flow and electromagnetic field problems.

The **CST** provides the opportunity to mesh both transverse and diagonal conductor MHD channel flows with an arbitrary number of electric elements, and with any conductor slope angles, which allows performing a wide variety of computational experiments, and the possibility of optimizing channel geometry for the MHD power generation.

The flexibility of the mesh generation algorithm controlled by UTGRID® allows new features to be considered like a second slope angle for electric elements in MHD generator, which has never been done before. The author hopes that it might be considered as qualitatively new step and approach to the MHD power generation general physical problem of channel optimization. Detailed instructions in the complete “**Manual**” in section 3.5 and friendly interactive interface of UTGRID® are intended to help any user of **CST** to attack very complicated and extremely important problems requiring a complex grid generation process.

## **LIST OF REFERENCES**

1. Analysis of MHD Generator Power Generation  
Ding Li, Dennis Keefer, Robert Rhodes, Charles L. Merkle, Konstantin Kolokolnikov  
University of Tennessee Space Institute.  
Department of Mechanical and Aerospace Engineering  
Tullahoma, TN 37388, USA (AIAA 2003-5050).
2. Li, D. and Merkle, C.L.  
“Analysis of Real Fluid Flows in Converging Diverging Nozzles”, AIAA-2003-4132, Accepted for publication, 33<sup>rd</sup> AIAA Fluid Dynamics Conference, Orlando FL, June23-26, 2003.
3. Li, D. and Merkle, C. L.  
“Application of a General Structured-Unstructured Solver to Flows of Arbitrary Fluids”, ICCFD’2000, Japan.
4. Wilcox, D. C., “Turbulence Modeling for CFD,” DCW Industries, Inc., ISBN 0-9636051-5-1, 1998.
5. Crawford, R. A., “FINAL REPORT FOR AUGUST 26, 2001 – MARCH 26, 2003”, submitted to the General Atomics Corporation, San Diego, California.
6. Li, Ding “PGRID”, Private Communication, 2002.
7. *American National Standard Programming Language FORTRAN*, ANSI X3.-1978 (1978).
8. *American National Standard Programming Language FORTRAN*, ANSI X3.-1991 (1991).
9. *International Language Standard ISO Standard 1539-199 (E)*.

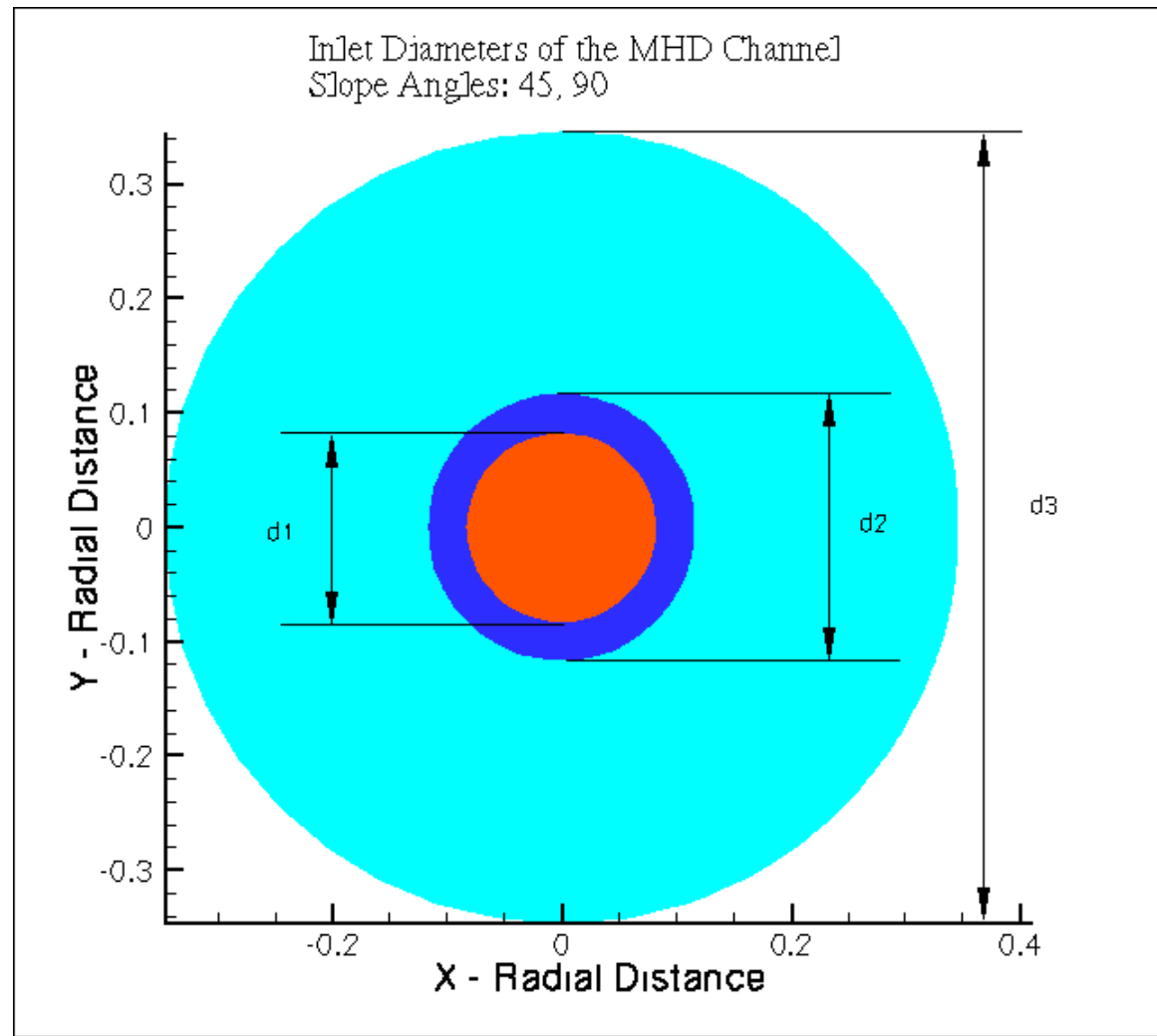
10. *Fortran 90 Handbook*, Intertext-McGraw Hill, New-York, NY, 1992.
11. *High Performance Fortran Language Specification*, Revision 1.0, Rice University, Houston, Texas (1993), <http://www.crpc.rice.edu/HPFF>.
12. *OpenMP Fortran Application Program Interface*, Version 1.1, November 1999.  
<http://www.openmp.org>.
13. *Programming in VAX Fortran*, Version 4.0, Digital Equipment Corporation (September, 1984).
14. *IBM VS Fortran*, IBM Corporation, Rev. GC26-4119.
15. Military Standard, Fortran, DOD Supplement to American National Standard Programming Language Fortran, ANSI x.3-1978, MIL-STD-1753 (November 9, 1978).
16. PGI User's Guide, *Parallel Fortran, C and C++ for Scientists and Engineers*, STMicroelectronics, Inc., Release 4.0, May 2002, <http://www.pgroup.com>.
17. VIGYAN, Inc., <http://www.vigyan.com/>
18. CFD Research Corporation, <http://www.cfdrc.com/>
19. Fluent, Inc., <http://www.fluent.com>
20. Schlumberger Limited, <http://www.slb.com/>
21. Fortran Lahey Computer Systems, Inc., <http://www.lahey.com>
22. Rosa, R. T. "Physical principles of Magneto-hydrodynamic Power Generation", *Physics of Fluids*, V.4, No. 2, pp 182 – 198, (1961).
23. Kirillin, V. A., and Shoyndlin, A. E. "MHD Energy Conversion, Physic-technical Problem", *AIAA Progress in Astronautics and Aeronautics*, Vol. 101, 1986.

24. Stein, Erwin, De Borst, René, and Hughes, Thomas J. R. "Encyclopedia of Computational Mechanics", Wiley Interscience, ISBN 0-470-84699, October, 2004.
25. Peterkin, R.E. and Frese, M.H., and Sovenec, C. R., "Transport of Magnetic Flux in an Arbitrary Coordinate ALE Code," Journal of Computational Physics Vol. 140, 148-171, (1998).
26. Rhodes, Robert P, Private communication, UTSI, 2003, Tullahoma, Tennessee.
27. Jackson, John David, 1925 – Classical Electrodynamics, John Wiley & Sons, Inc., 3<sup>rd</sup> edition. ISBN 0-471-30932-X, 1998.

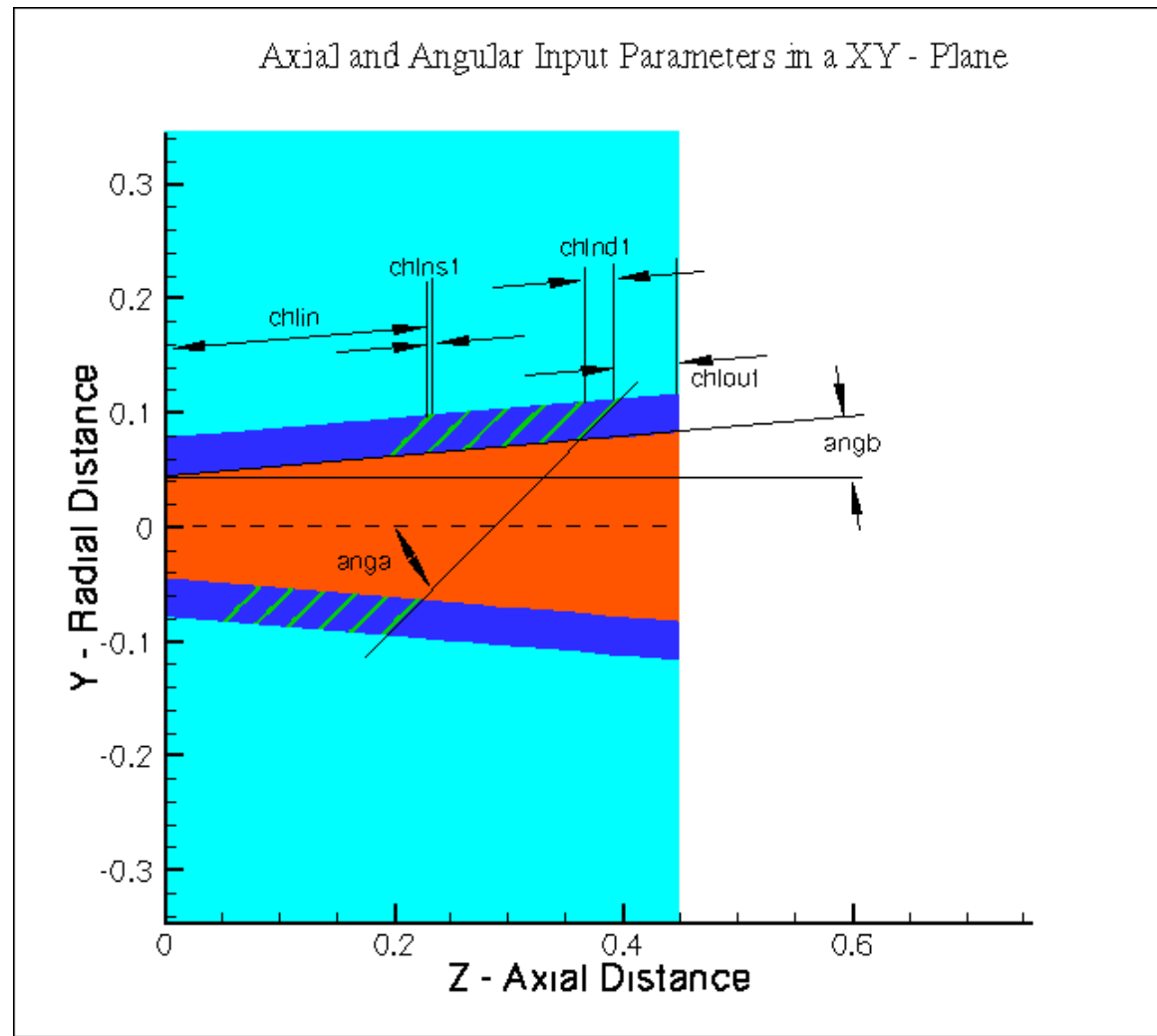
## **LIST OF FIGURES**

## **APPENDIX A1**

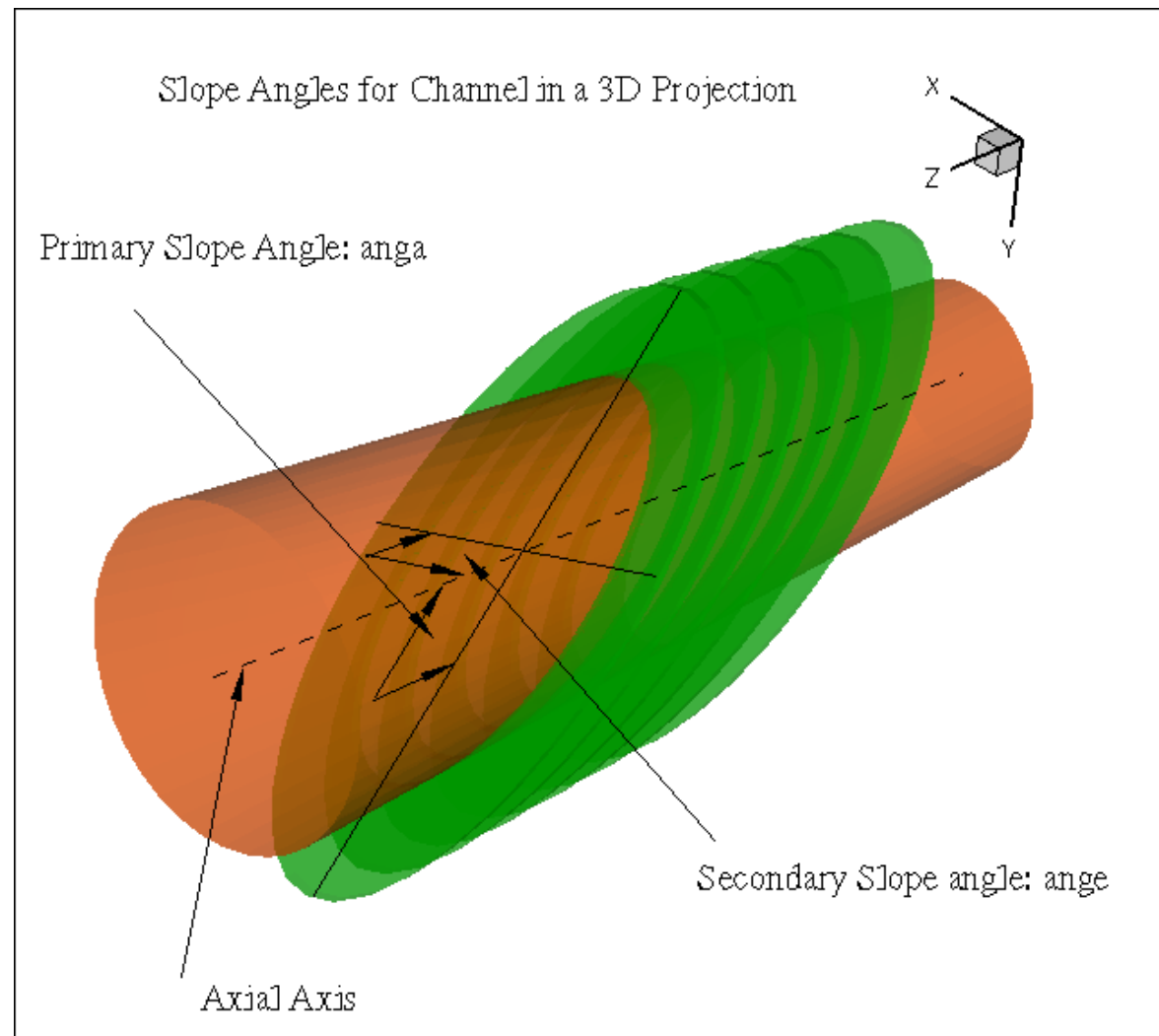
**A1.1. FIGURES ILLUSTRATING INPUT GEOMETRY FOR THE CHANNEL AND GEOMETRY AND MESH  
INPUT PARAMETERS FOR THE NOZZLE**



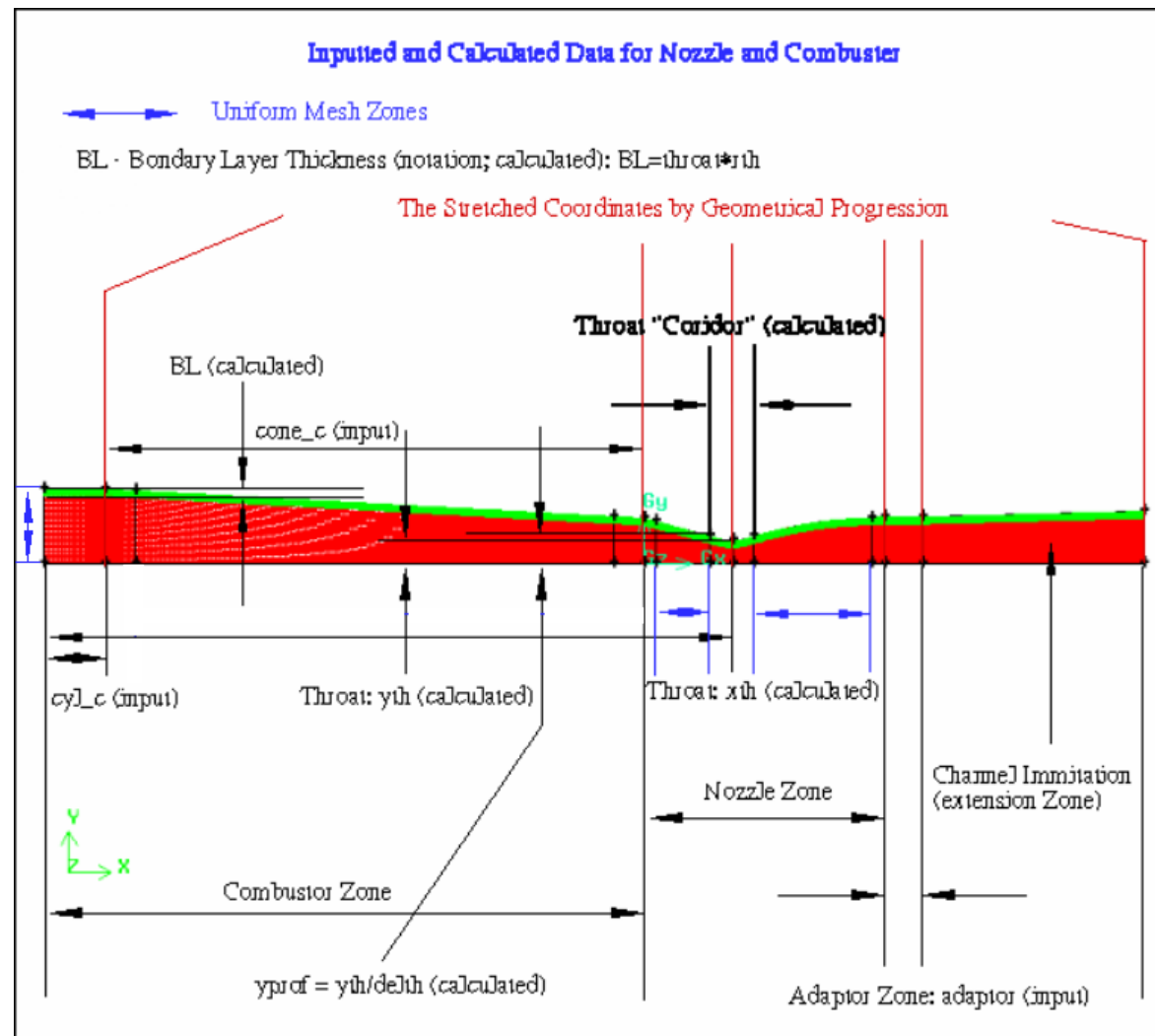
**Figure 3.5.6.** Inlet Input Diameters for the Channel.



**Figure 3.5.7.** Axial and Angular Parameters in a XY – Plane for the Channel.

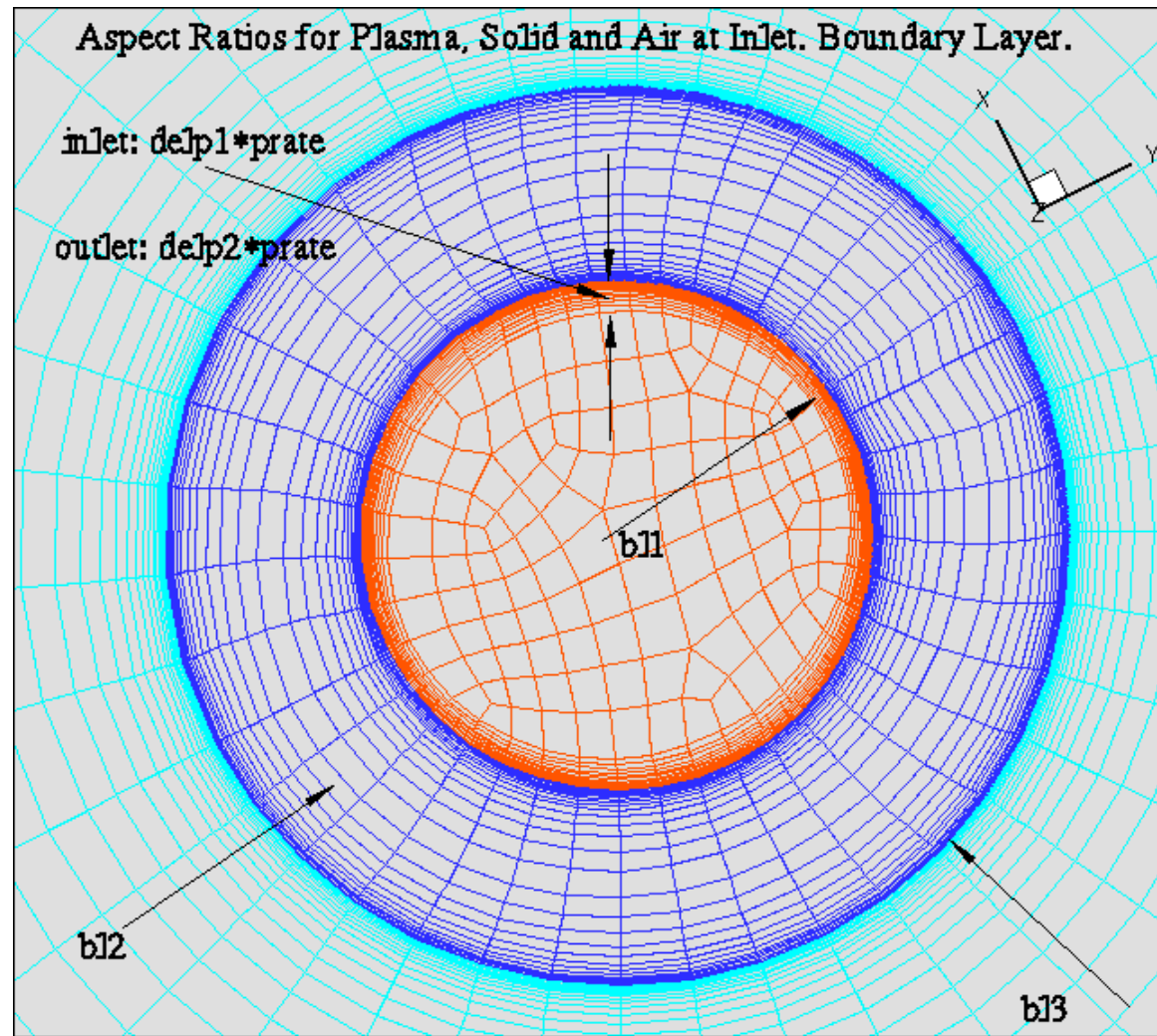


**Figure 3.5.8.** Primary and Secondary Slope Angles for the Electric Elements respectively to the Z – axis in 3D view.

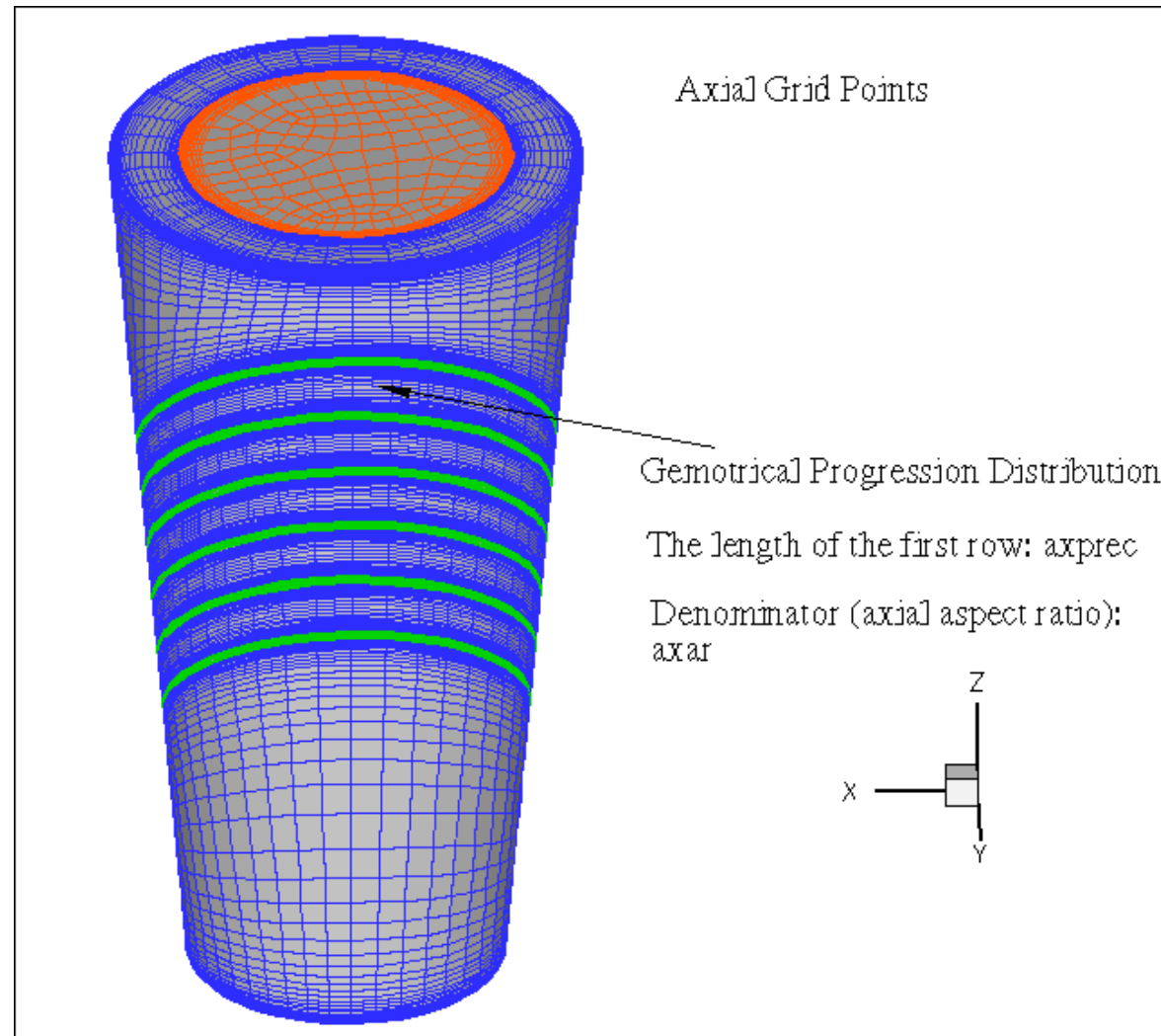


**Figure 3.5.9.** Inputted and calculated Data for the Combustor – Nozzle Tract of the Driven Flow.

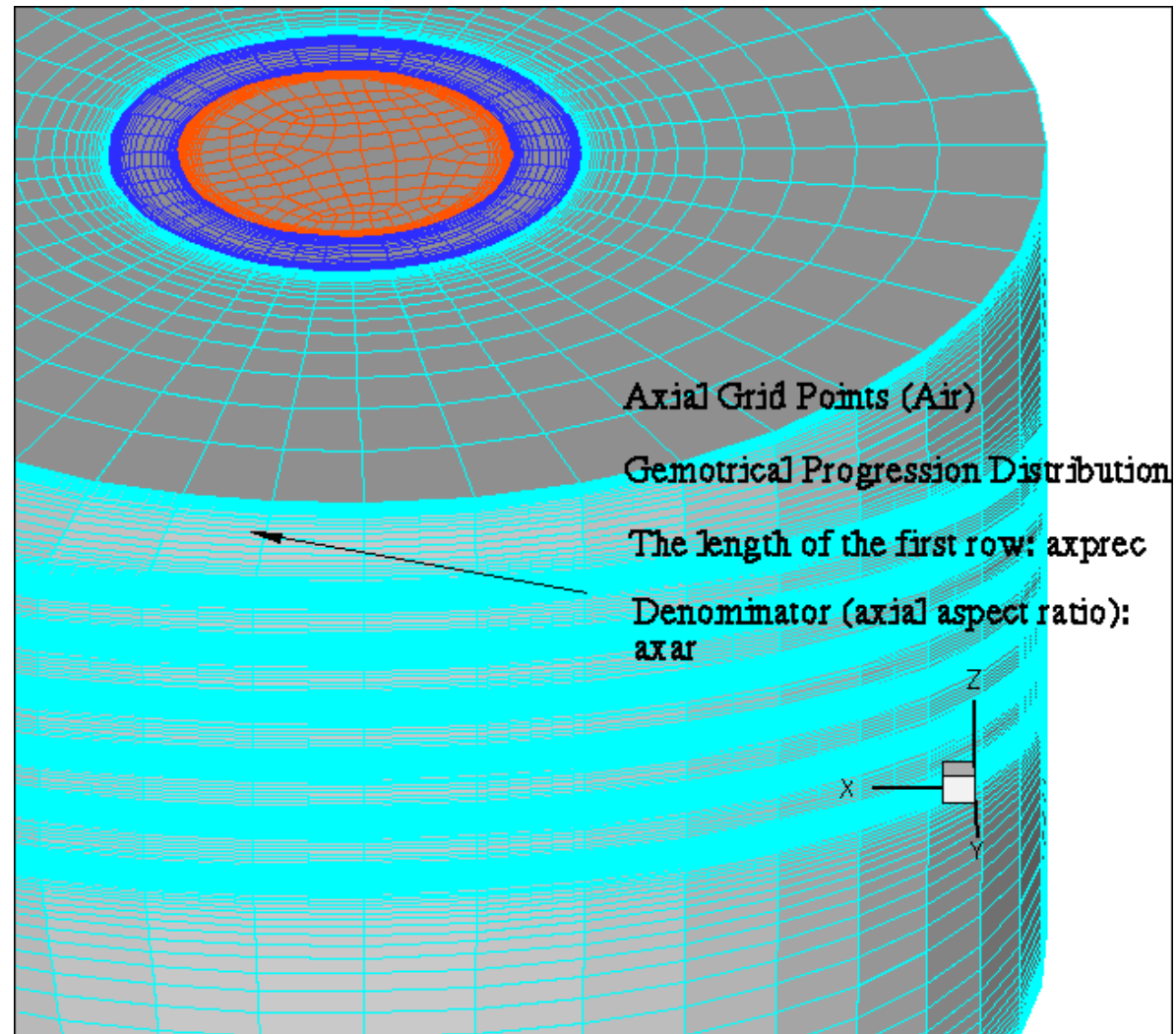
#### **A1.2. FIGURES ILLUSTRATING INPUT MESH PARAMETERS FOR THE CHANNEL**



**Figure 3.5.10.** Aspect Ratio for Plasma, Solid, Air Zones and the Thickness of the Plasma's Boundary Layer.



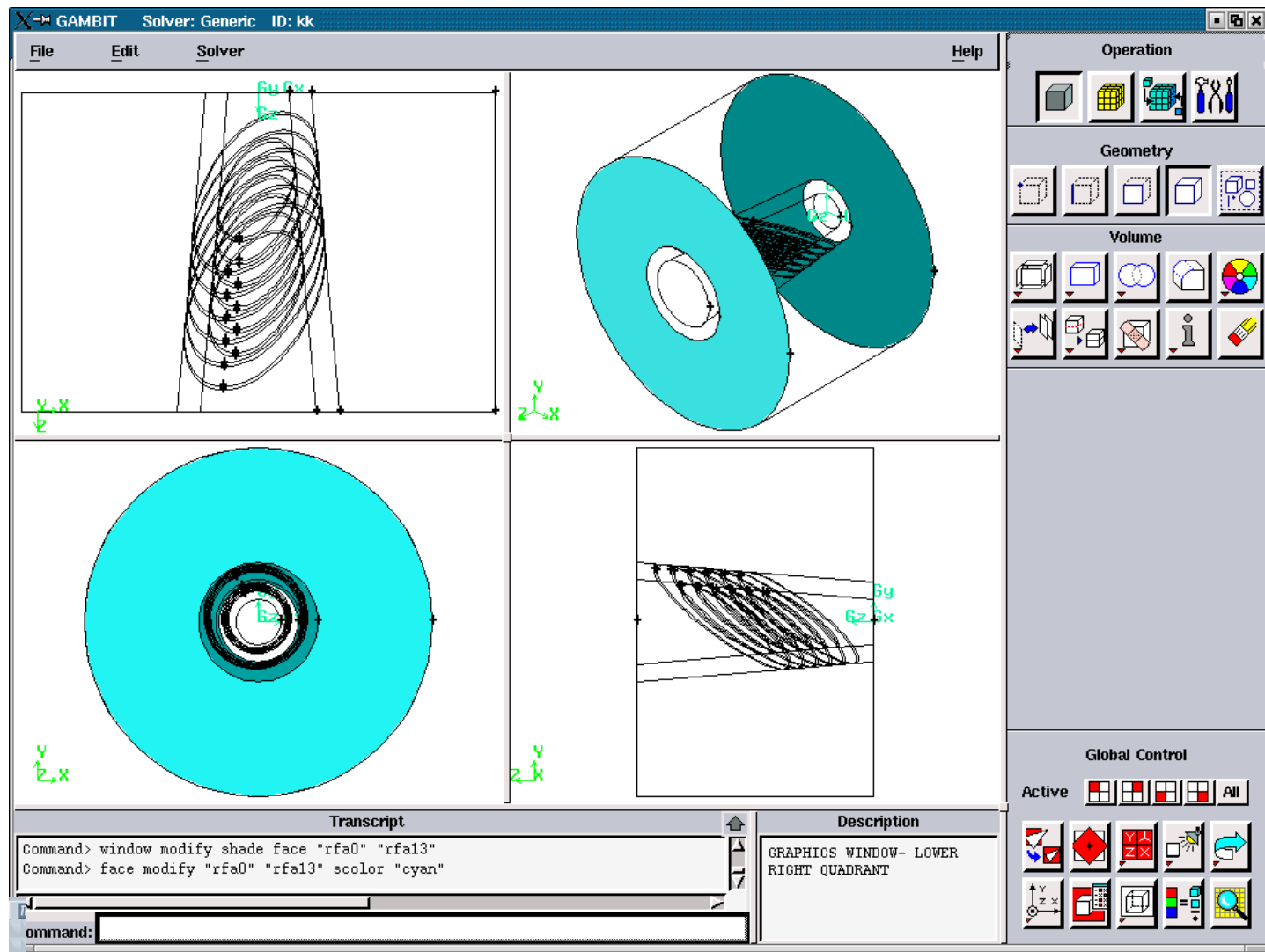
**Figure 3.5.11.** Axial Grids Distribution with inputted Precision and Axial Aspect ratio, independent on Radial ones.



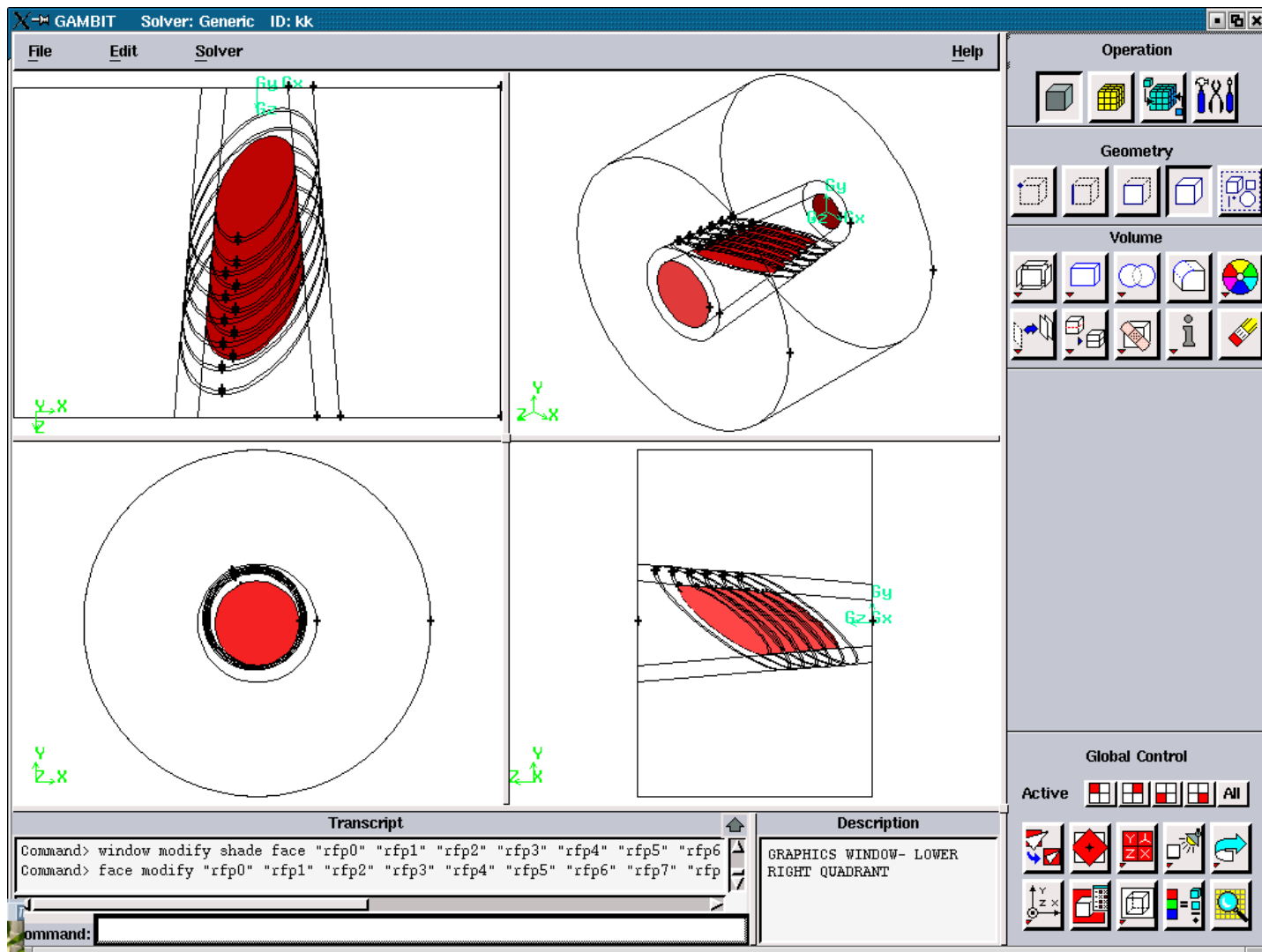
**Figure 3.5.12.** Axial Grid Distribution for the Air Domain Zone.

### **A1.3. FIGURES ILLUSTRATING THE CREATION OF THE FILE ‘modify1’**

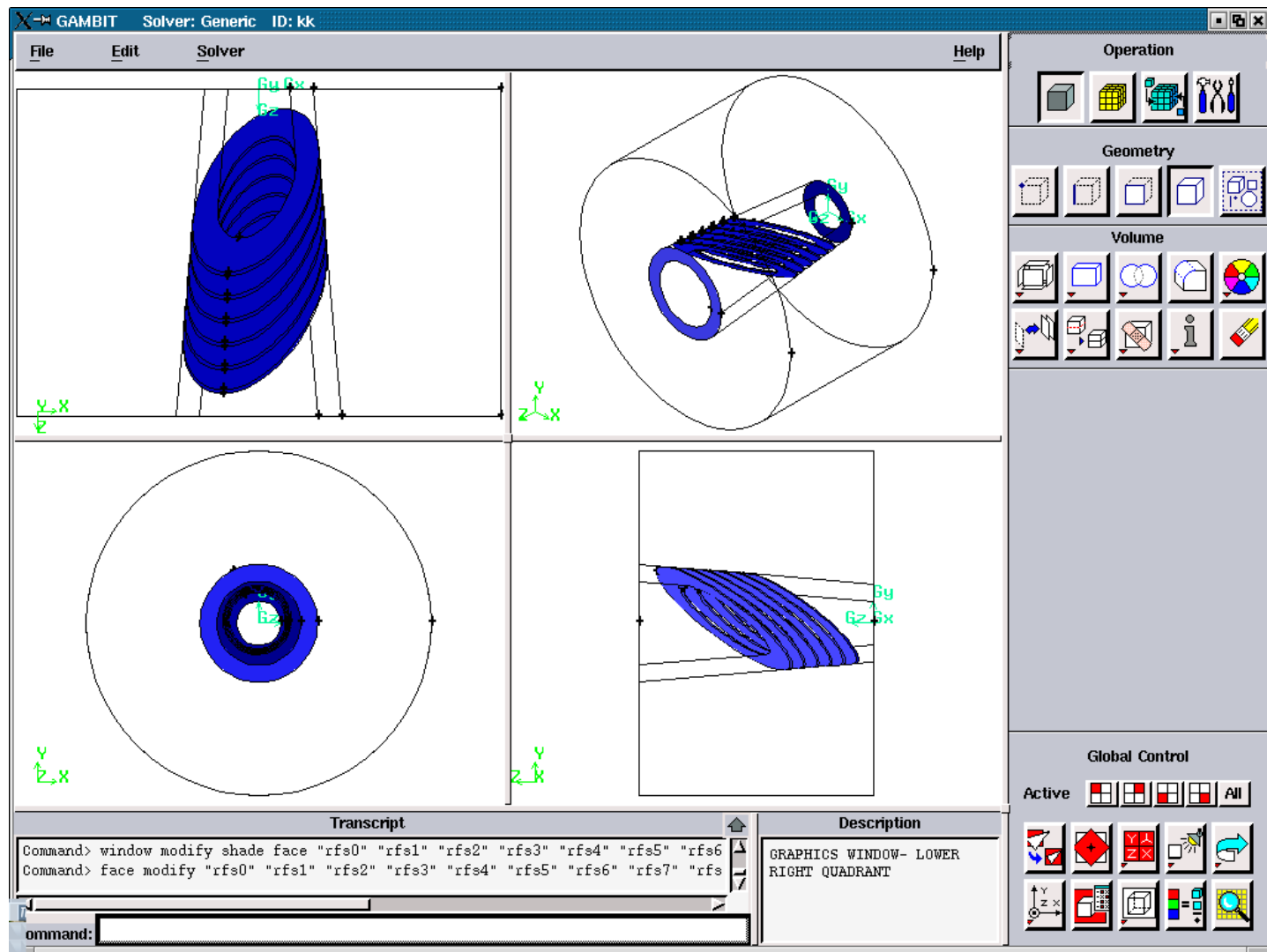
#### **A1.3.1. FIGURES FOR THE SLOPE ANGLES: 45, 60**



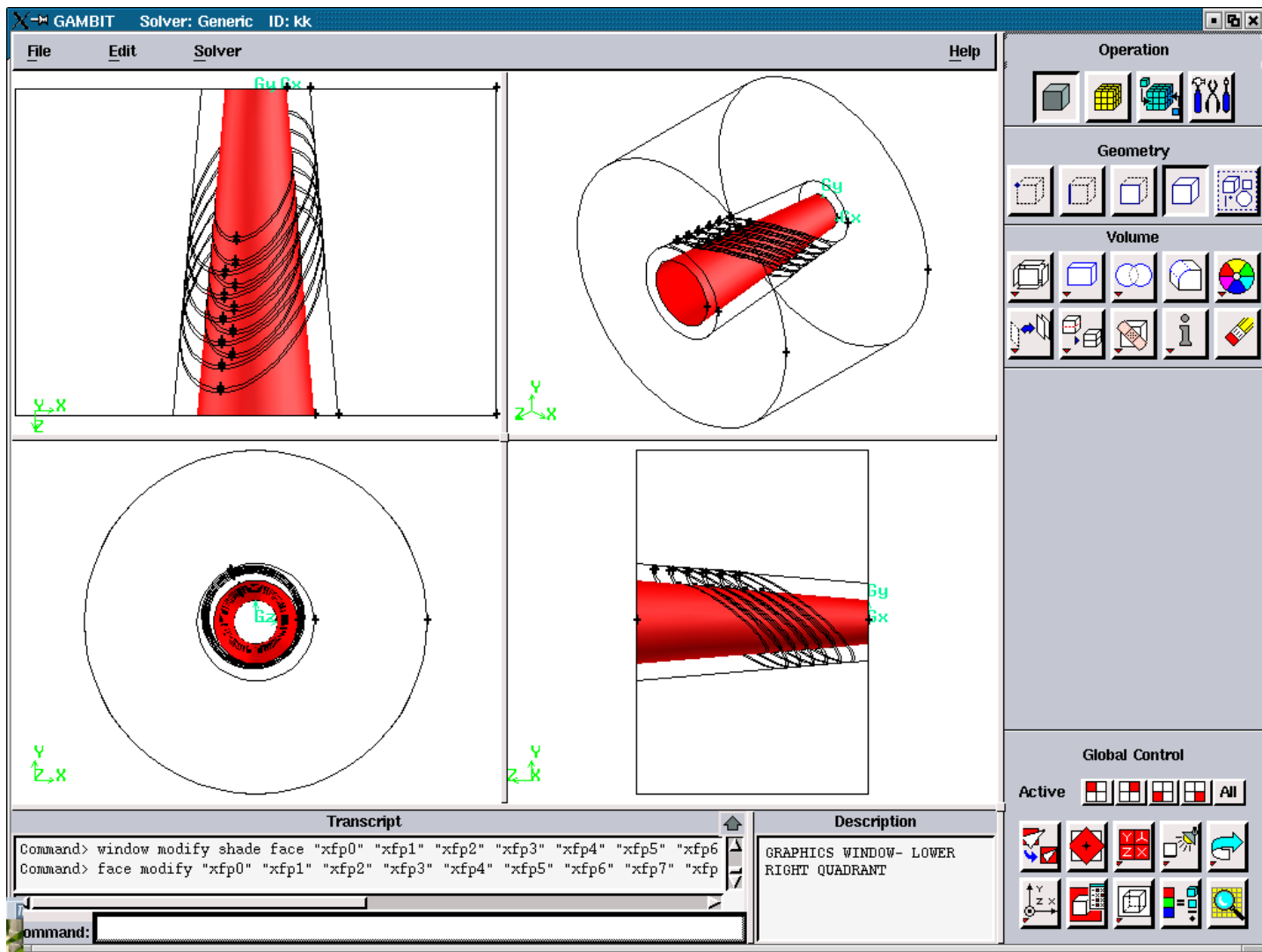
**Figure 3.5.13.** Radial Air Faces after the run of the journal file 'temp1.jou' with slope angles: 45, 60.



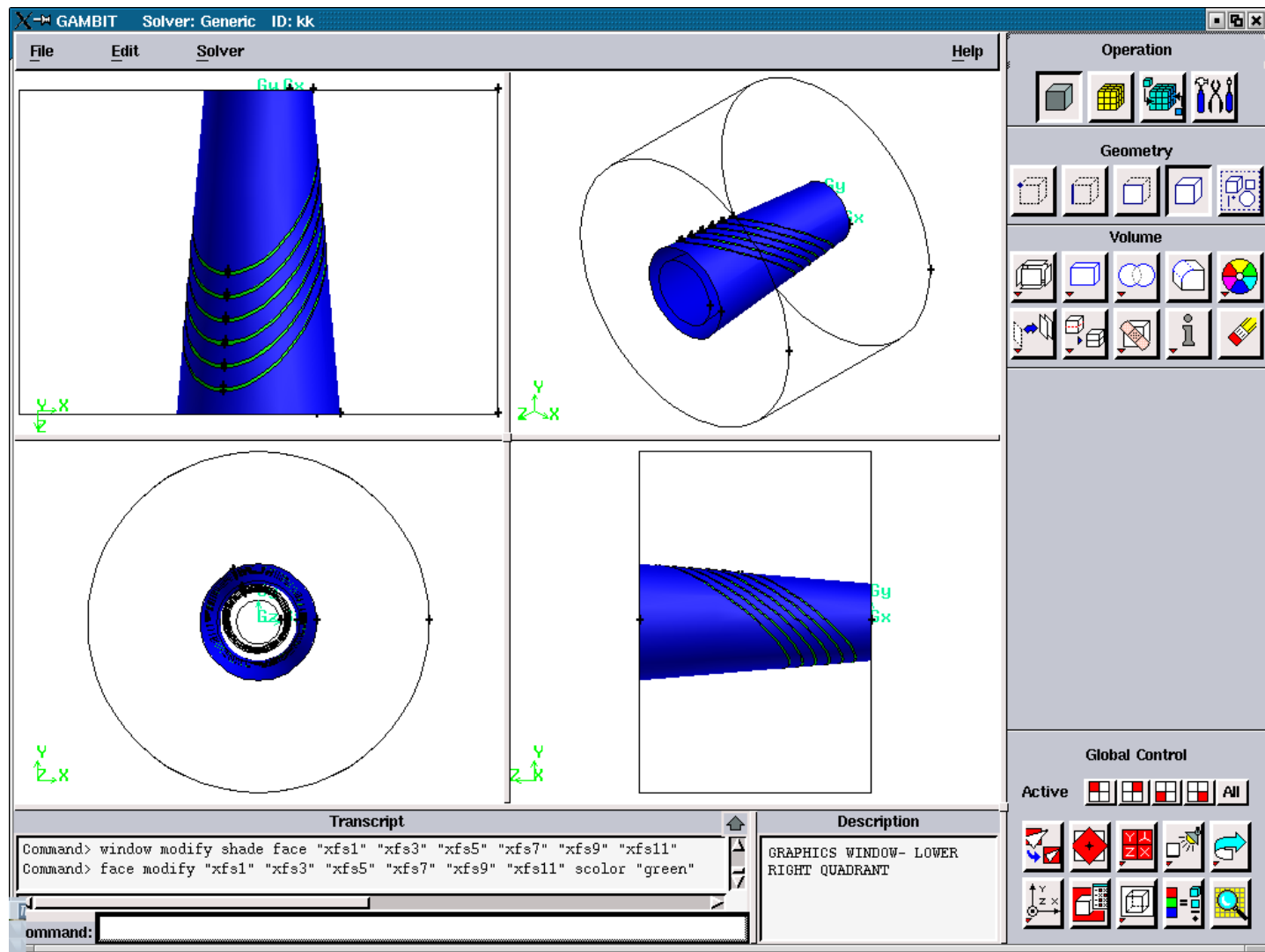
**Figure 3.5.14.** Radial Plasma Faces after the run of the file 'temp1.jou' with the slope angles: 45, 60.



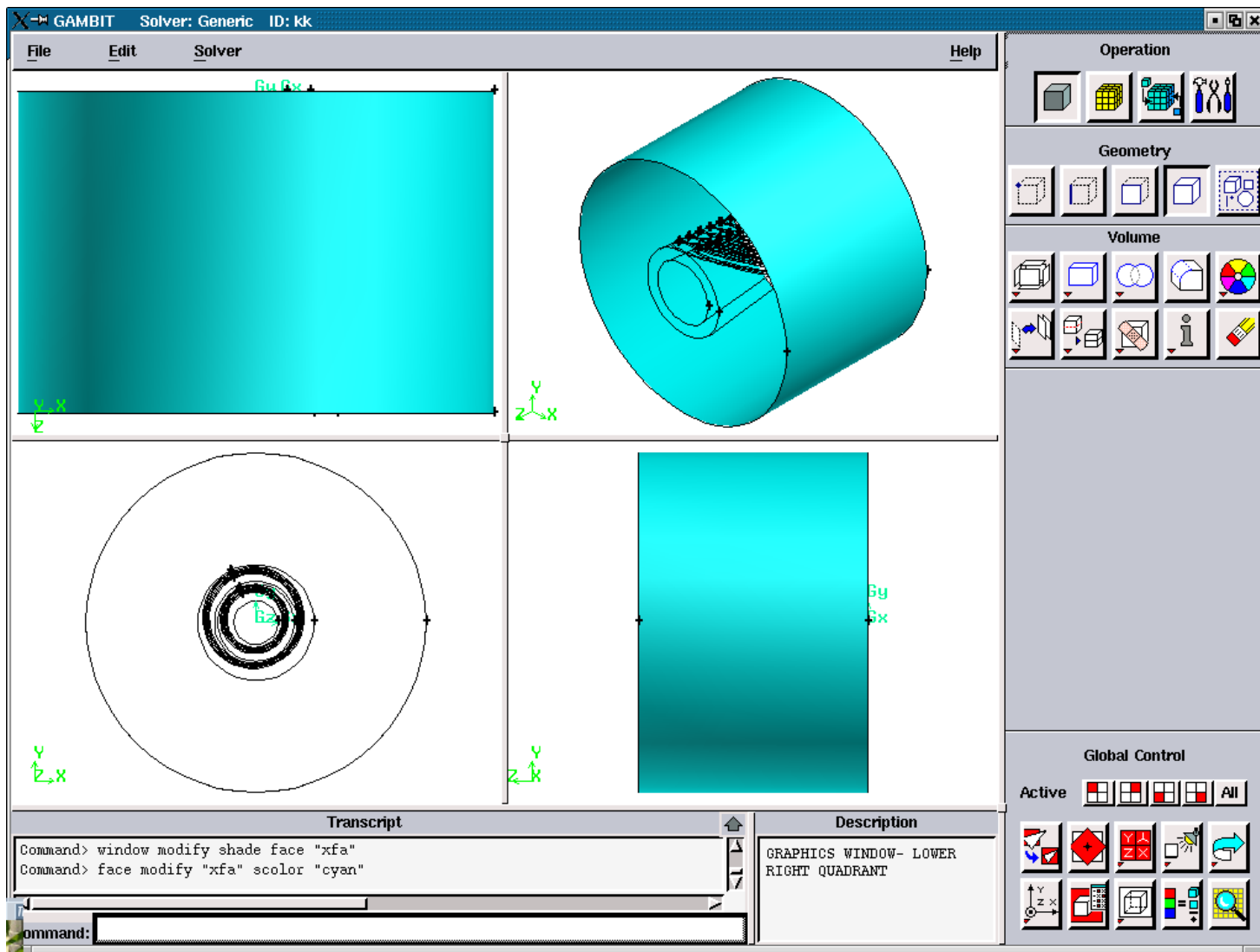
**Figure 3.5.15.** Radial Solid Faces after the run of the file 'templ.joy' with the slope angles: 45, 60.



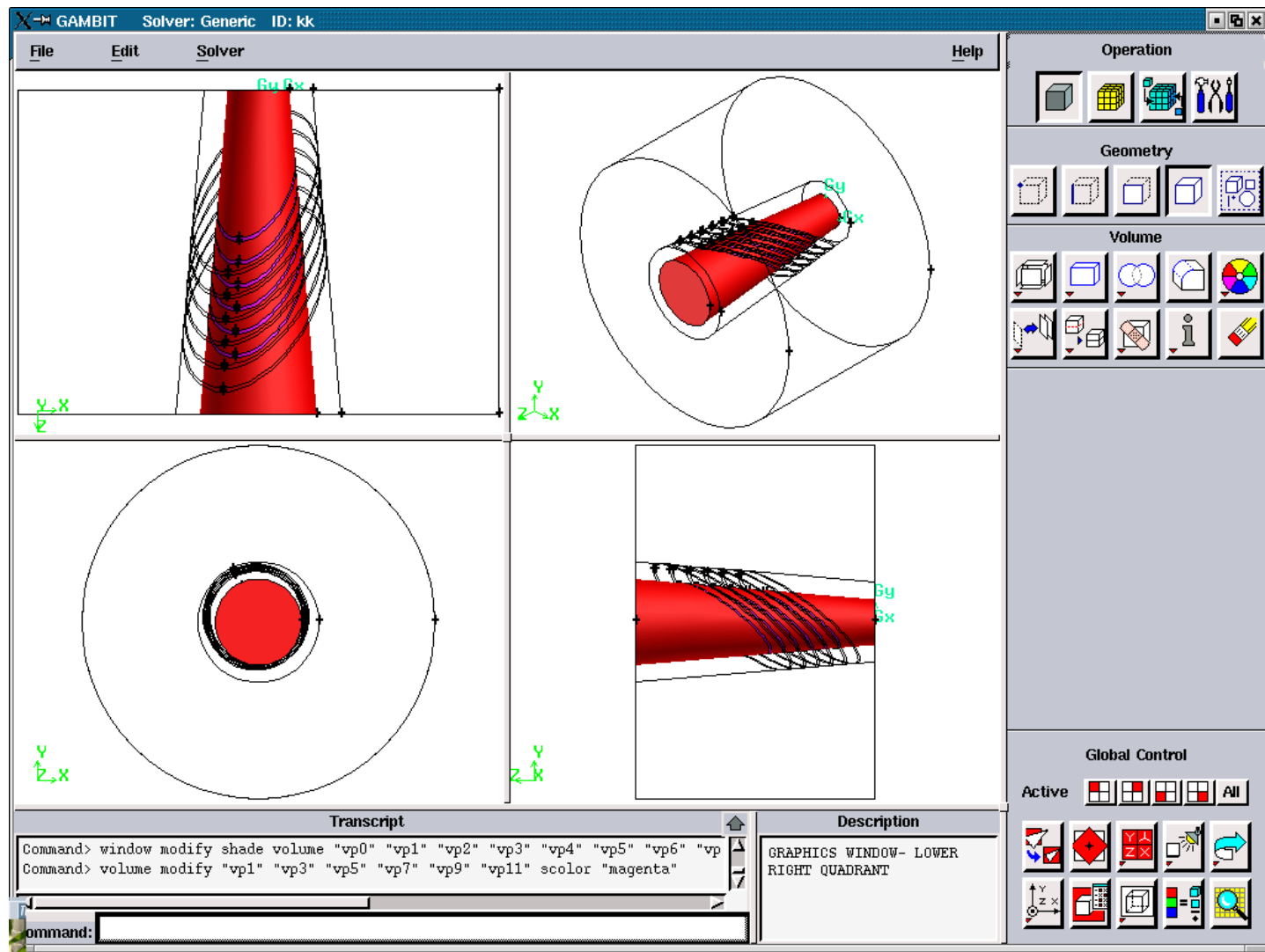
**Figure 3.5.16.** Axial Plasma Faces after the run of the file 'temp1.jou' with the slop angles: 45, 60.



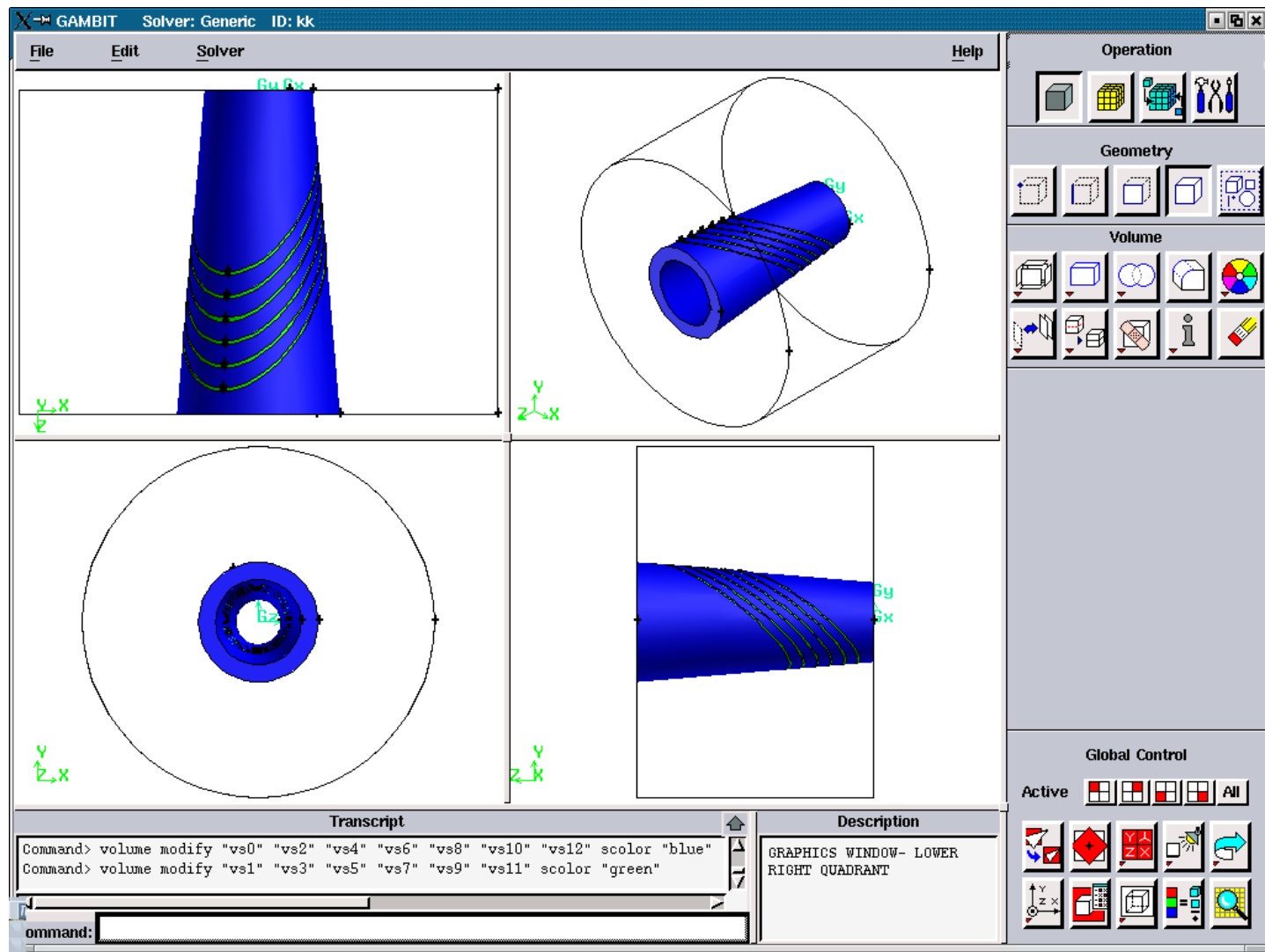
**Figure 3.5.17.** Axial Solid faces after the run of the file 'temp1.jou' with the slope angles: 45, 60.



**Figure 3.5.18.** Axial Air Face after the run of the journal file 'temp1.jou' with the slope angles: 45, 60.



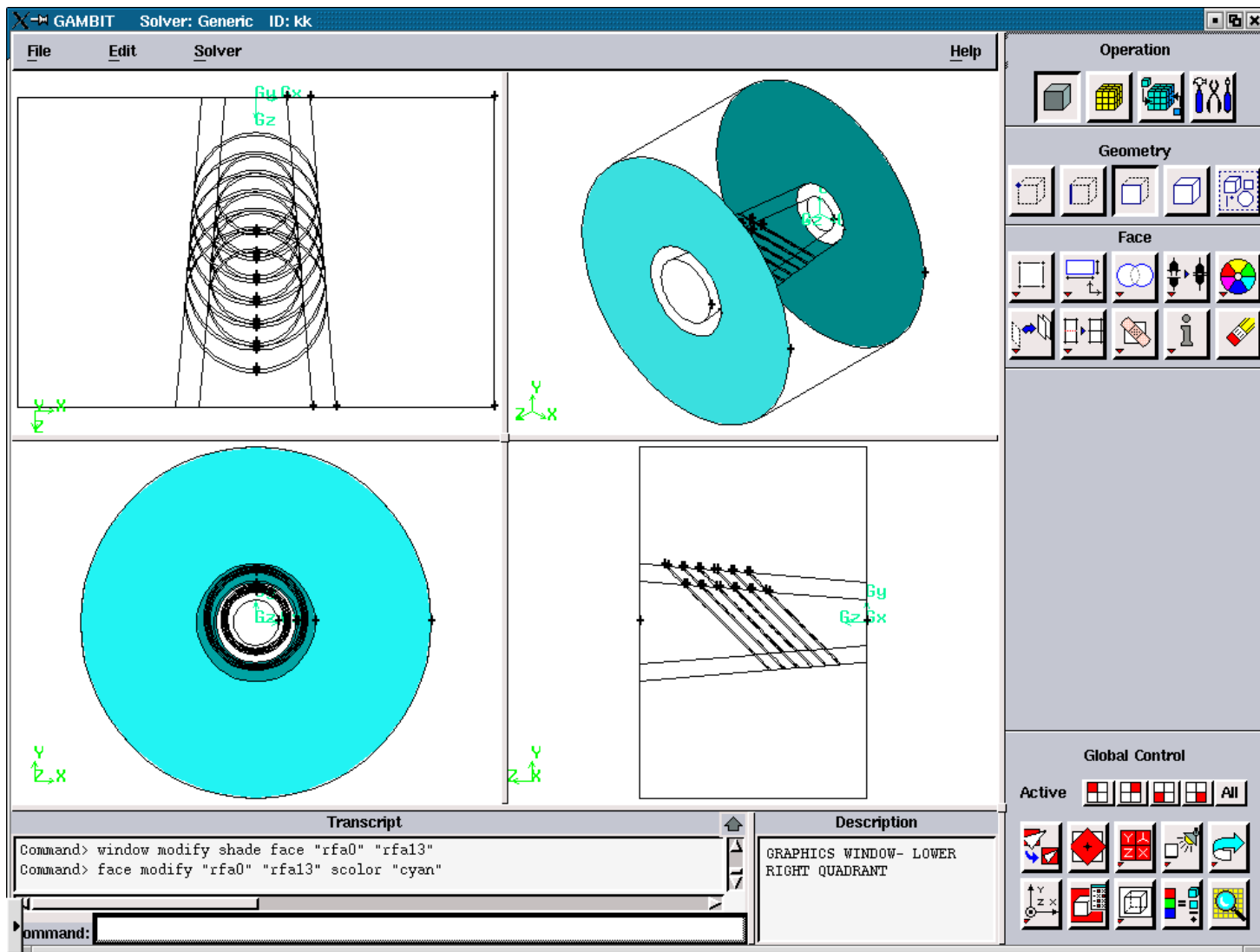
**Figure 3.5.19.** The Volumes of the Plasma after the run of the journal file 'temp1.jou' with the slope angles: 45, 60.



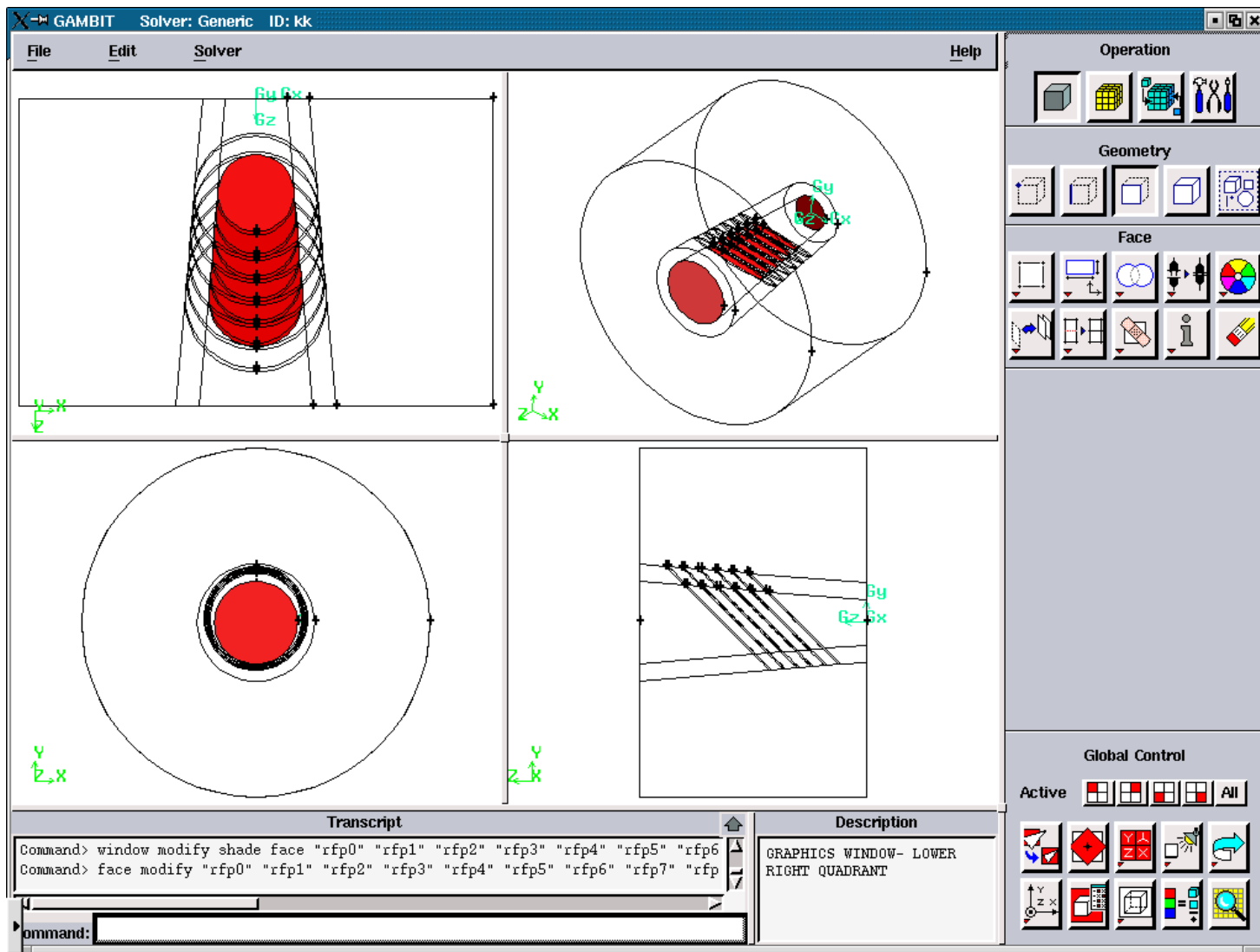
**Figure 3.5.20.** The Volumes of the Solid Zone after the run of the journal file 'templ.jou' with the slope angles: 45, 60.

**A1.3. FIGURES ILLUSTRATING THE CREATION OF THE FILE ‘modify1’**

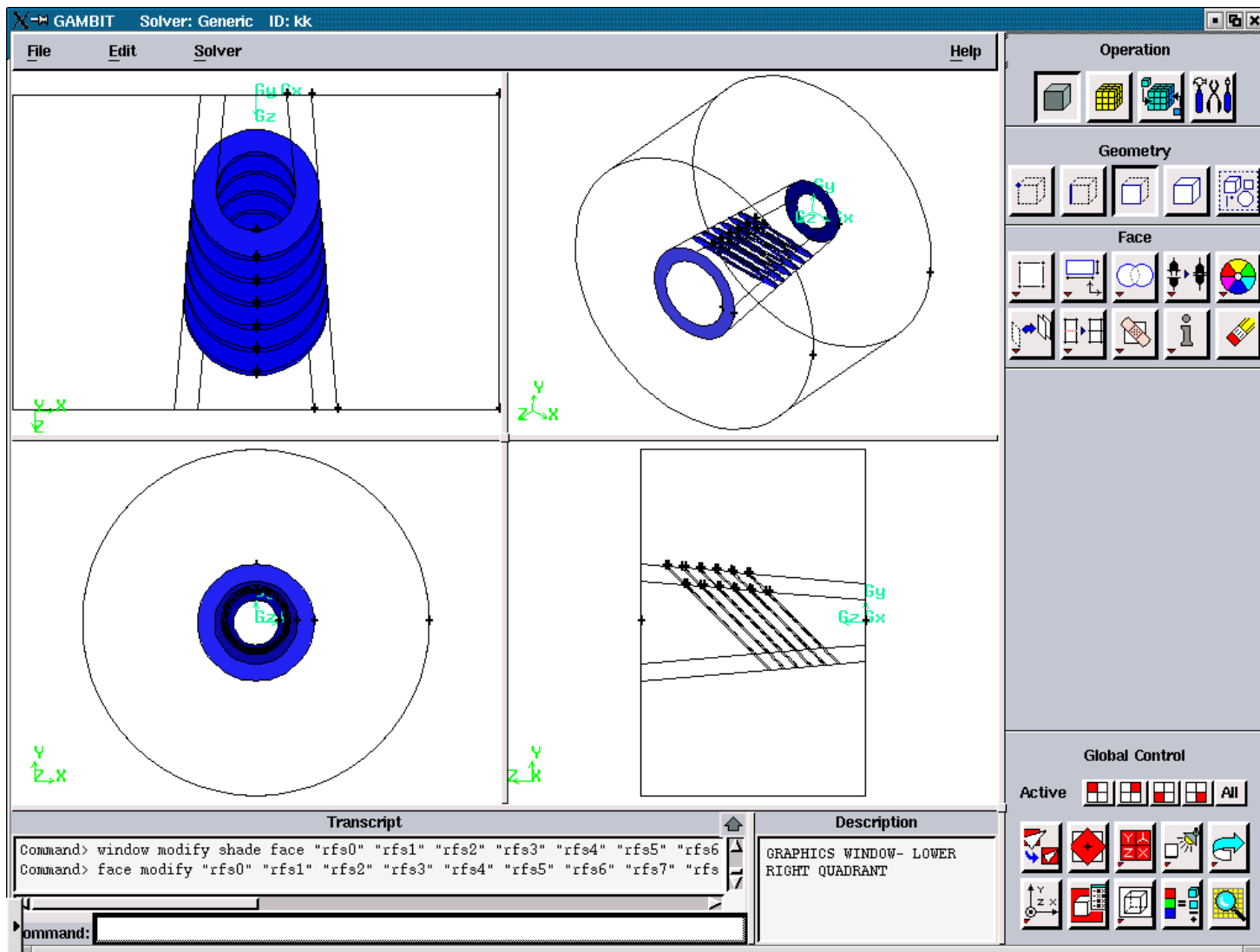
**A1.3.2. FIGURES FOR THE SLOPE ANGLES: 45, 90**



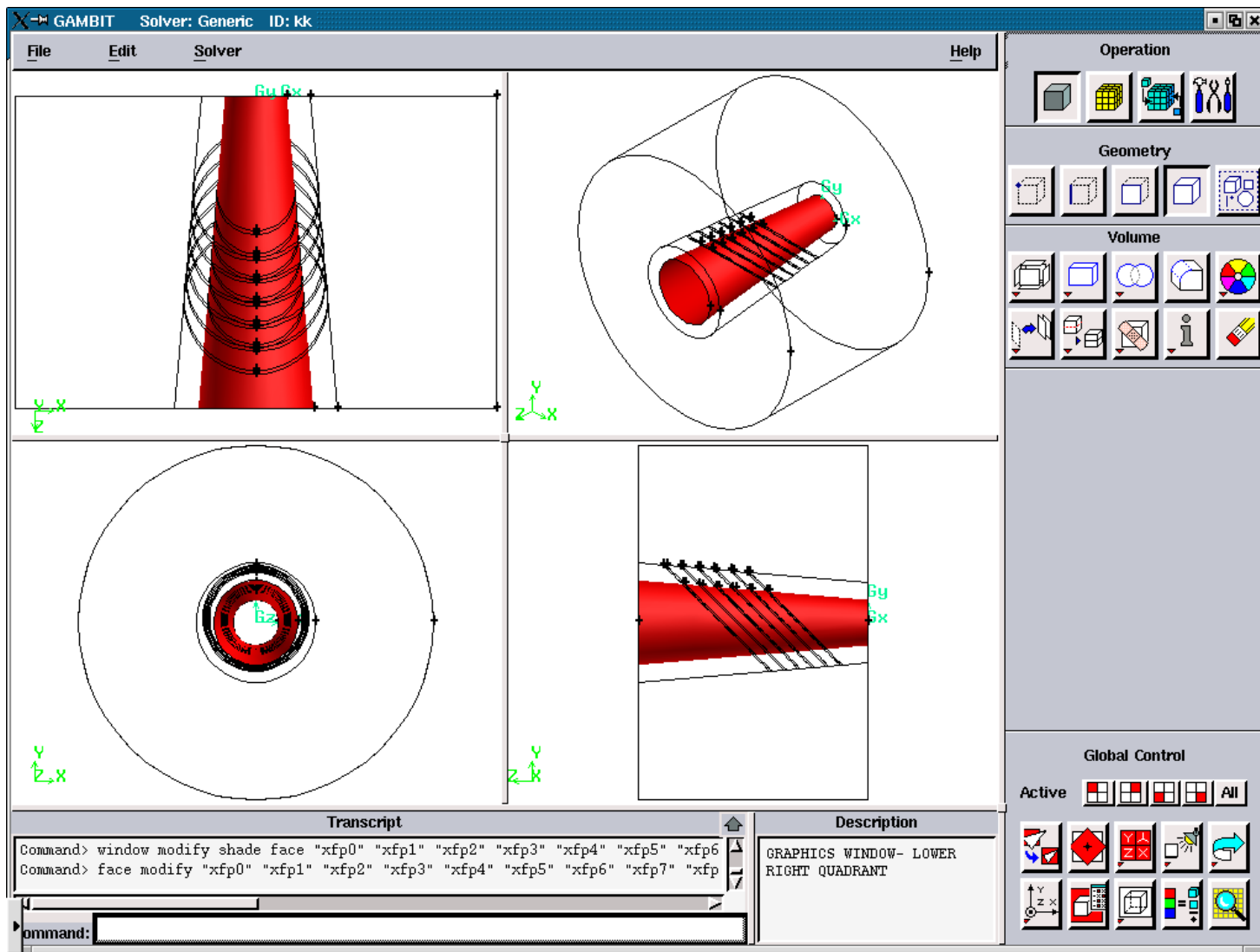
**Figure 3.5.21.** Radial Air Faces after the run of the journal file 'temp1.jou' with the slope angles: 45, 90.



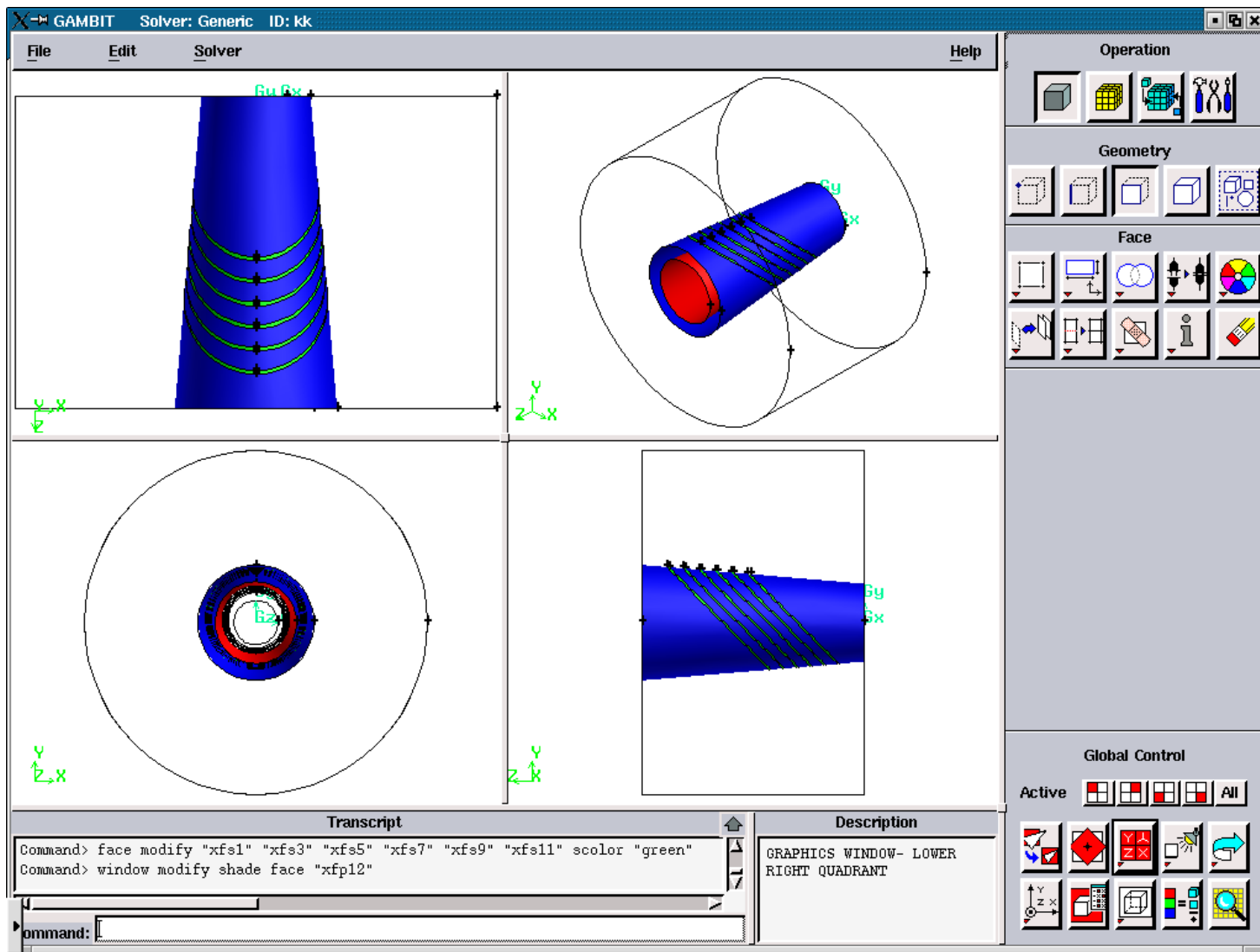
**Figure 3.5.22.** Radial Plasma Faces after the run of the journal file ‘temp1.jou’ with the slope angles: 45, 90.



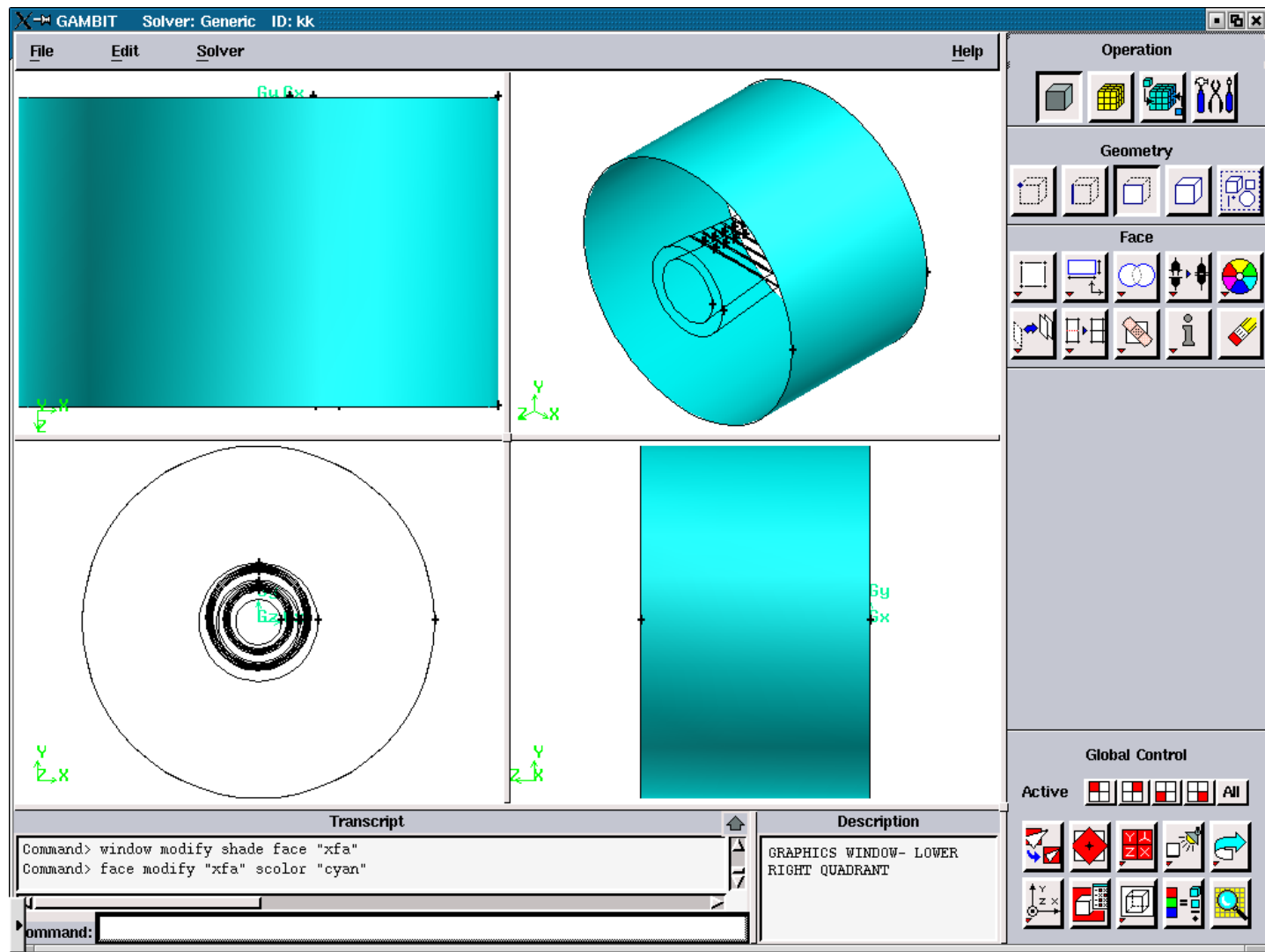
**Figure 3.5.23.** Axial Solid Faces after the run of the journal file 'temp1.jou' with the slope angles: 45, 90.



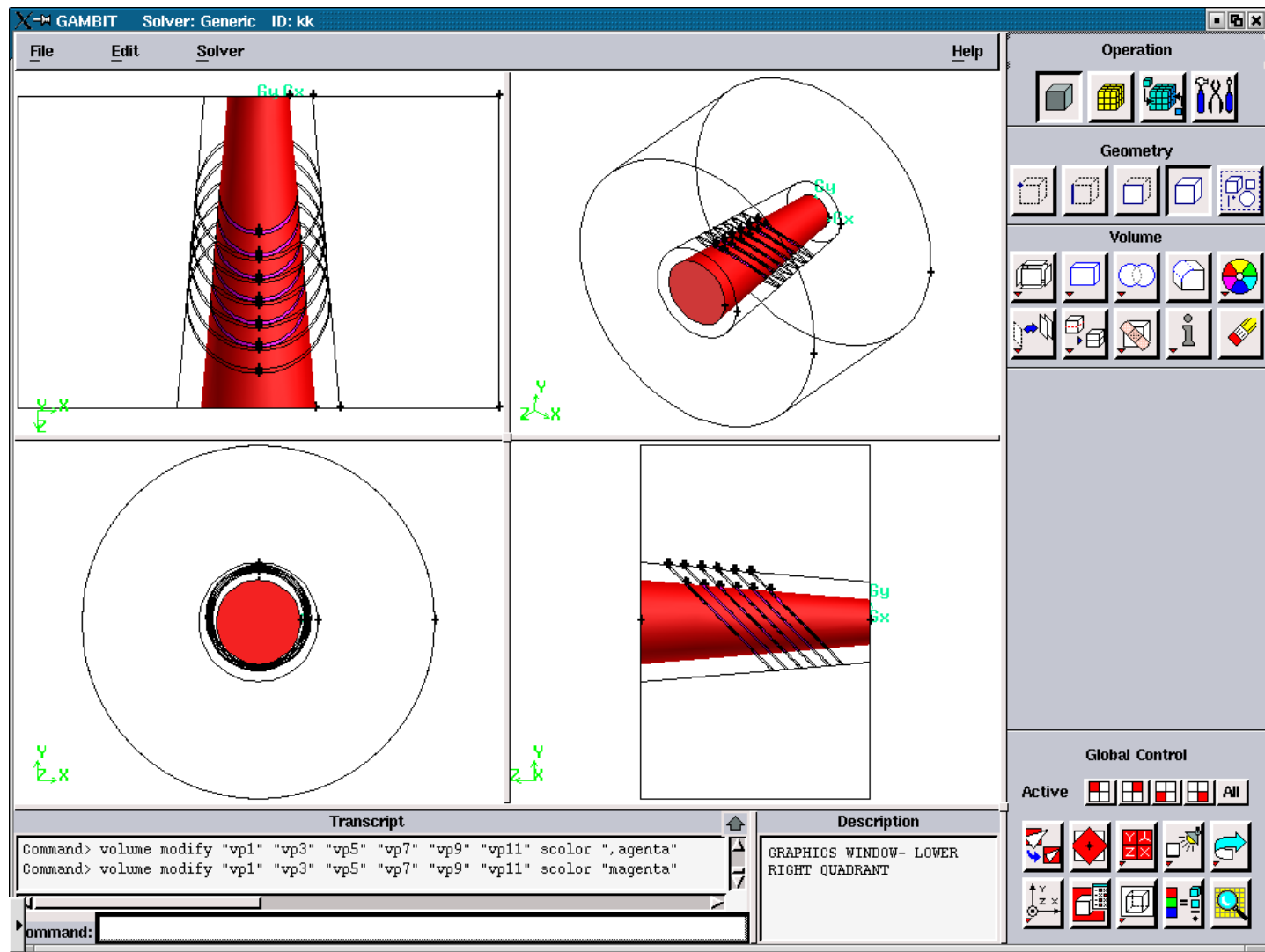
**Figure 3.5.24.** Axial Plasma Faces after the run of the journal file ‘temp1.jou’ with the slope angles: 45, 90.



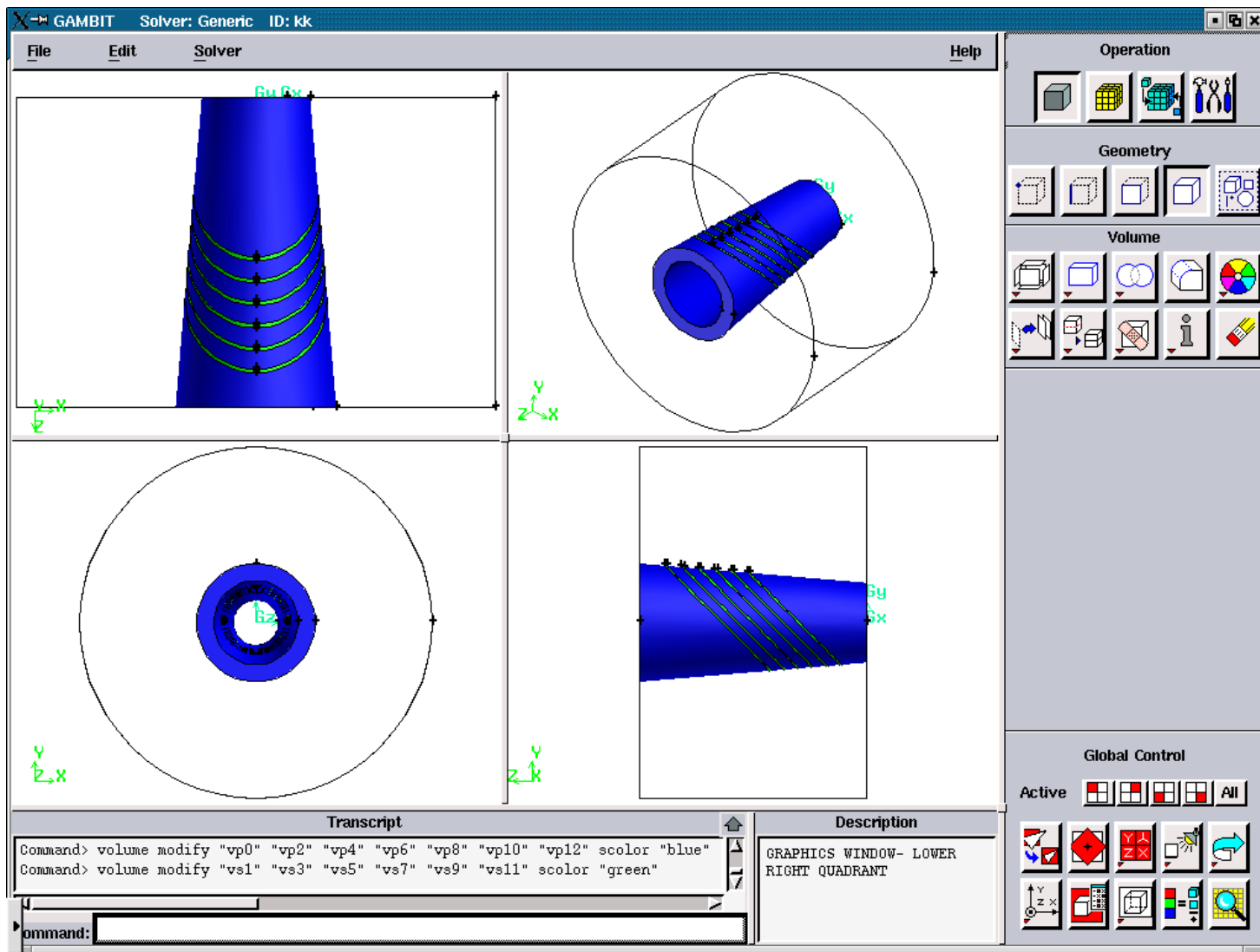
**Figure 3.5.25.** Axial Solid Faces after the run of the journal file 'temp1.jou' with the slope angles: 45, 90.



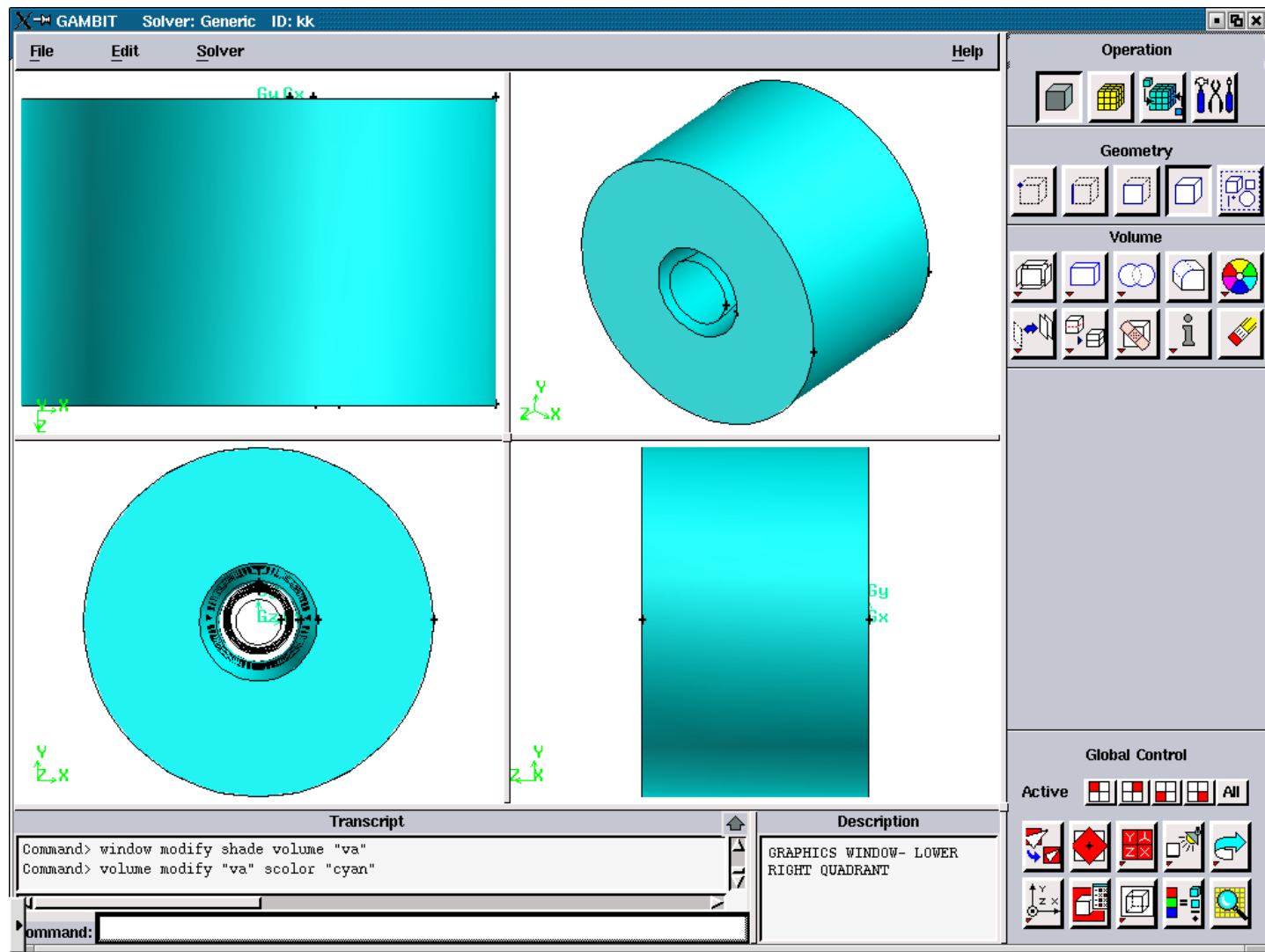
**Figure 3.5.26.** Axial Air Face after the run of the journal file 'temp1.jou' with the slope angles: 45, 90.



**Figure 3.5.27.** The Plasma Volumes after the run of the journal file 'temp1.jou' with the slope angles: 45, 90.



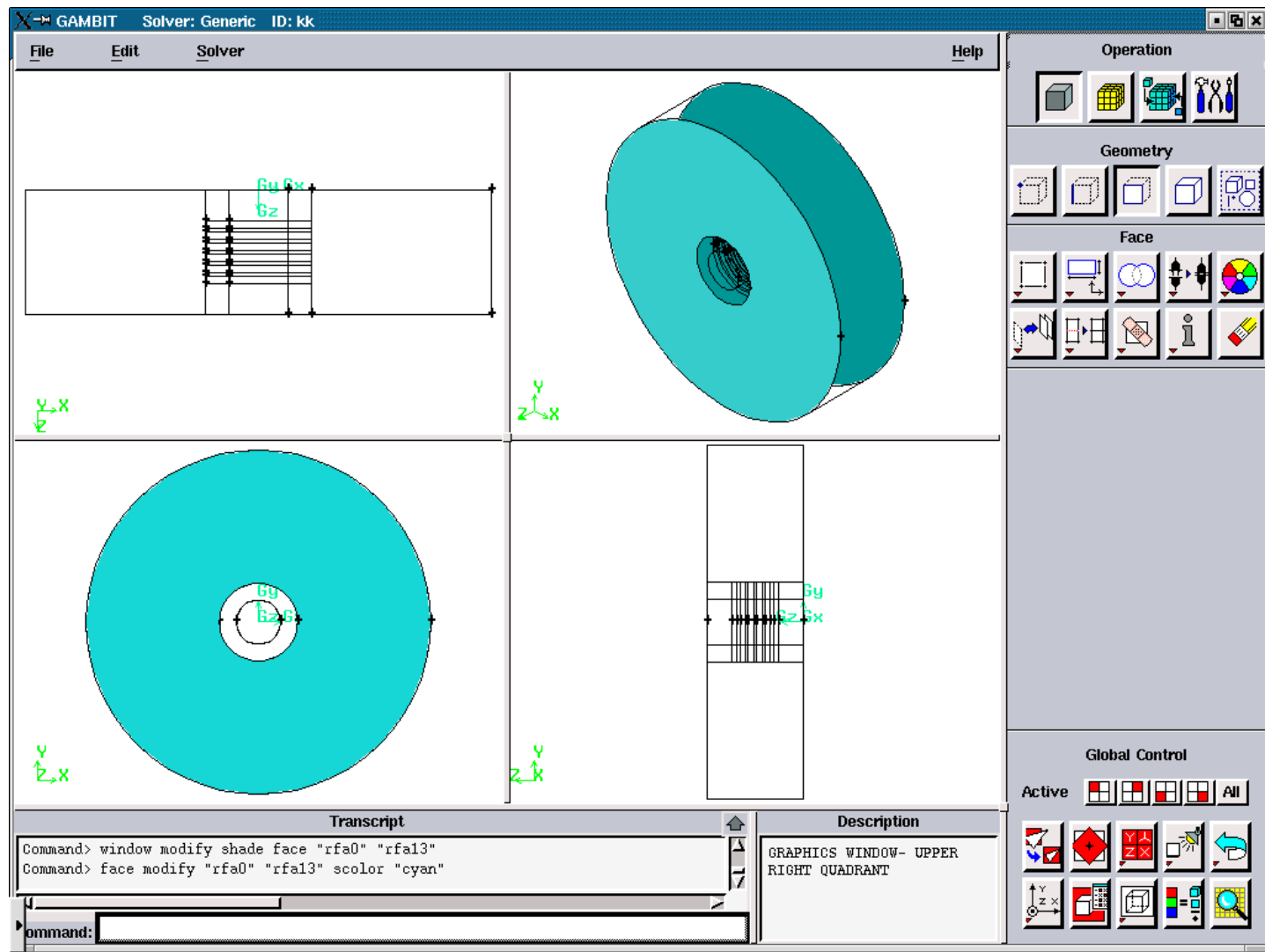
**Figure 3.5.28.** The Solid Volumes after the run of the journal file 'temp1.jou' with the slope angles: 45, 90.



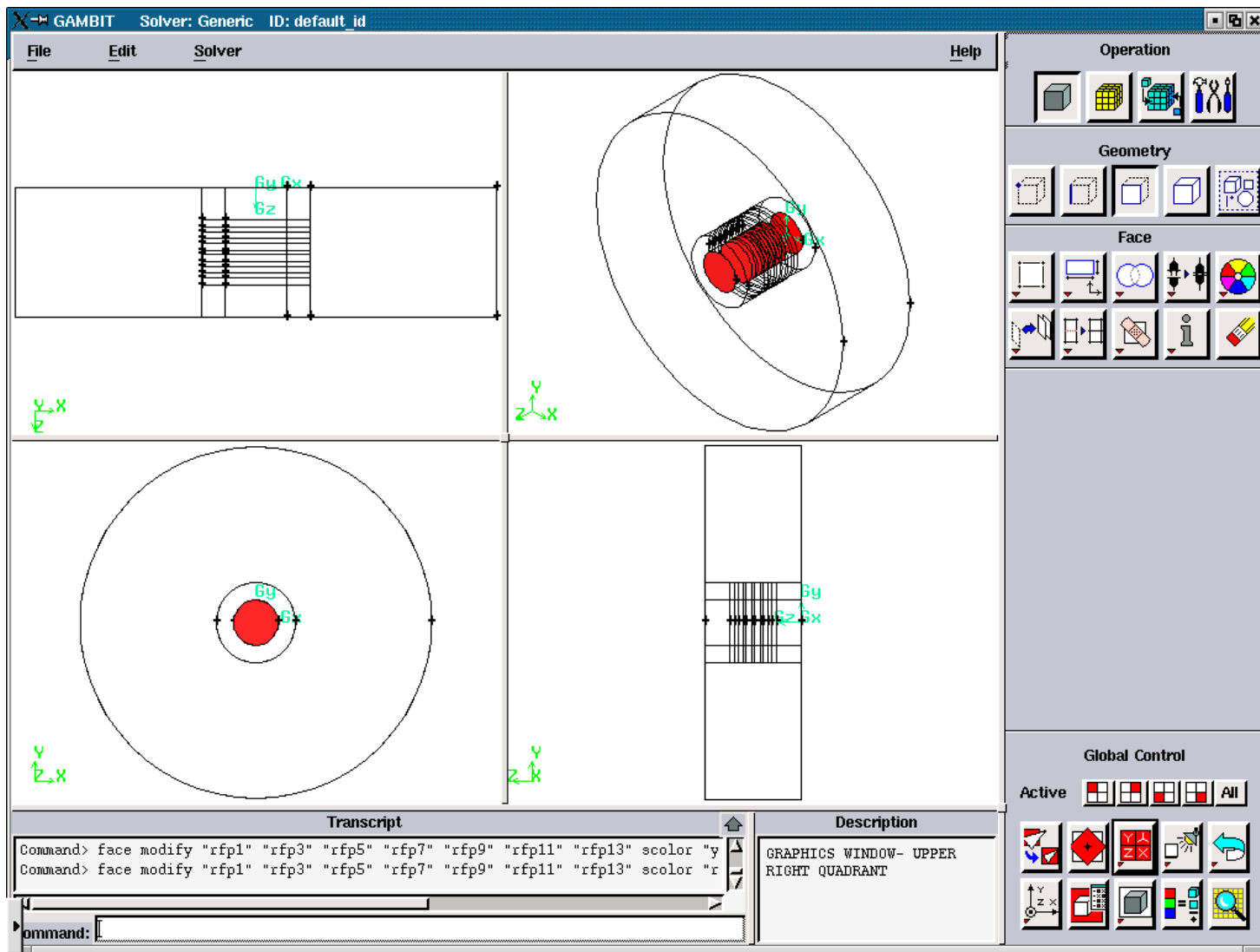
**Figure 3.5.29.** The Air Volume after the run of the journal file 'templ.jou' with the slope angles: 45, 90.

**A1.3. FIGURES ILLUSTRATING THE CREATION OF THE FILE ‘modify1’**

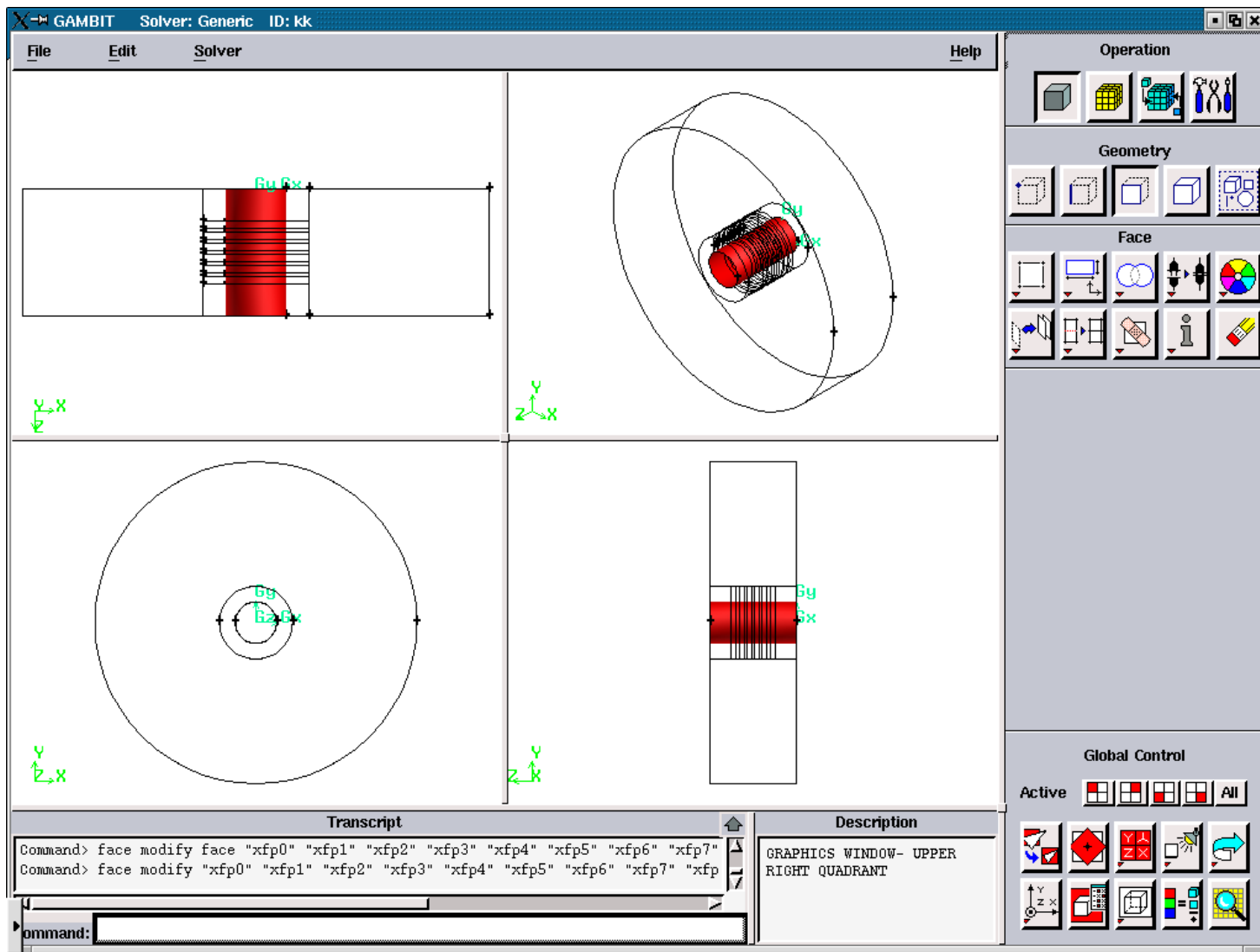
**A1.3.3. FIGURES FOR THE SLOPE ANGLES: 90, 90**



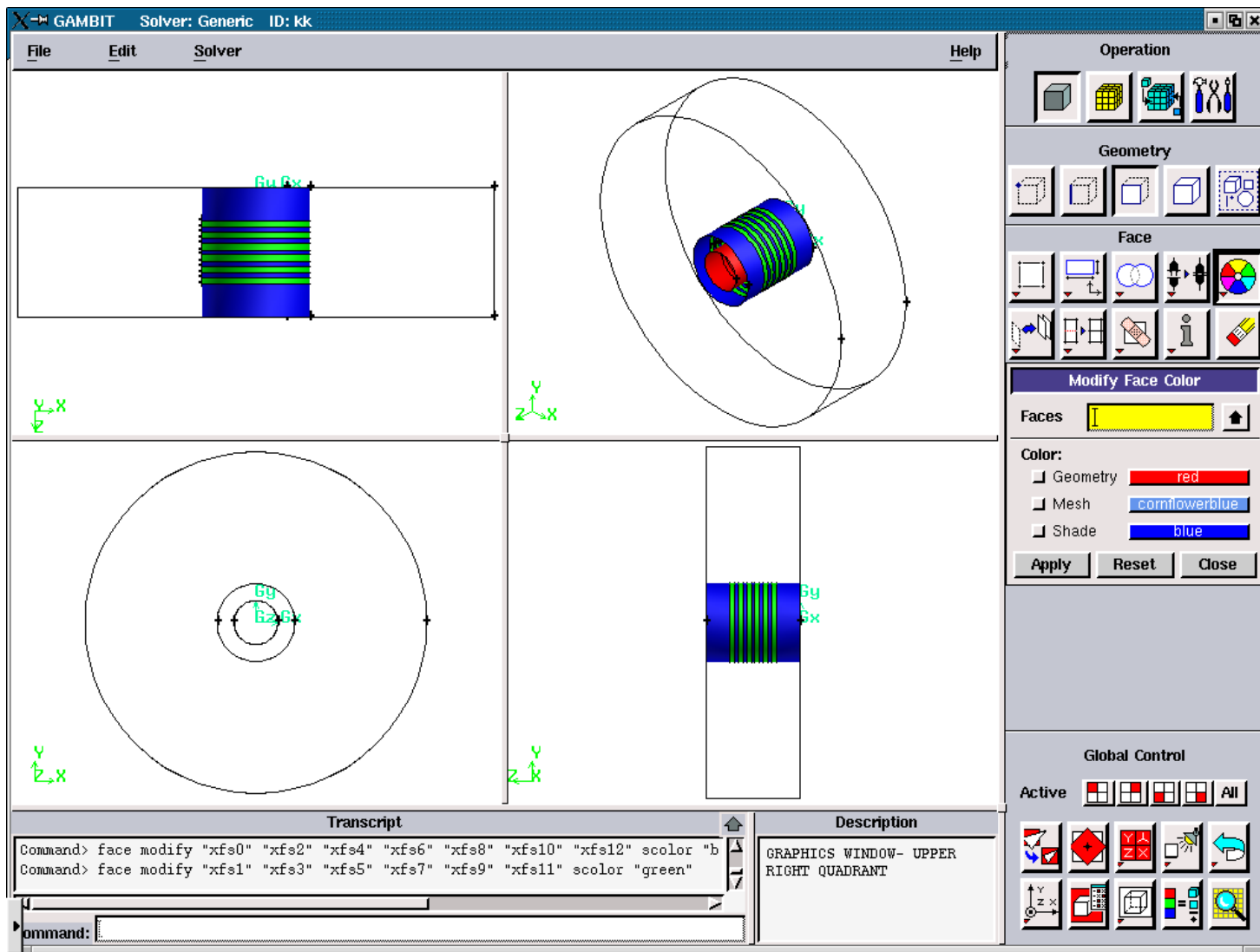
**Figure 3.5.30.** Radial Air Faces after the runoff the journal file 'temp1.jou' with the slope angles: 90, 90.



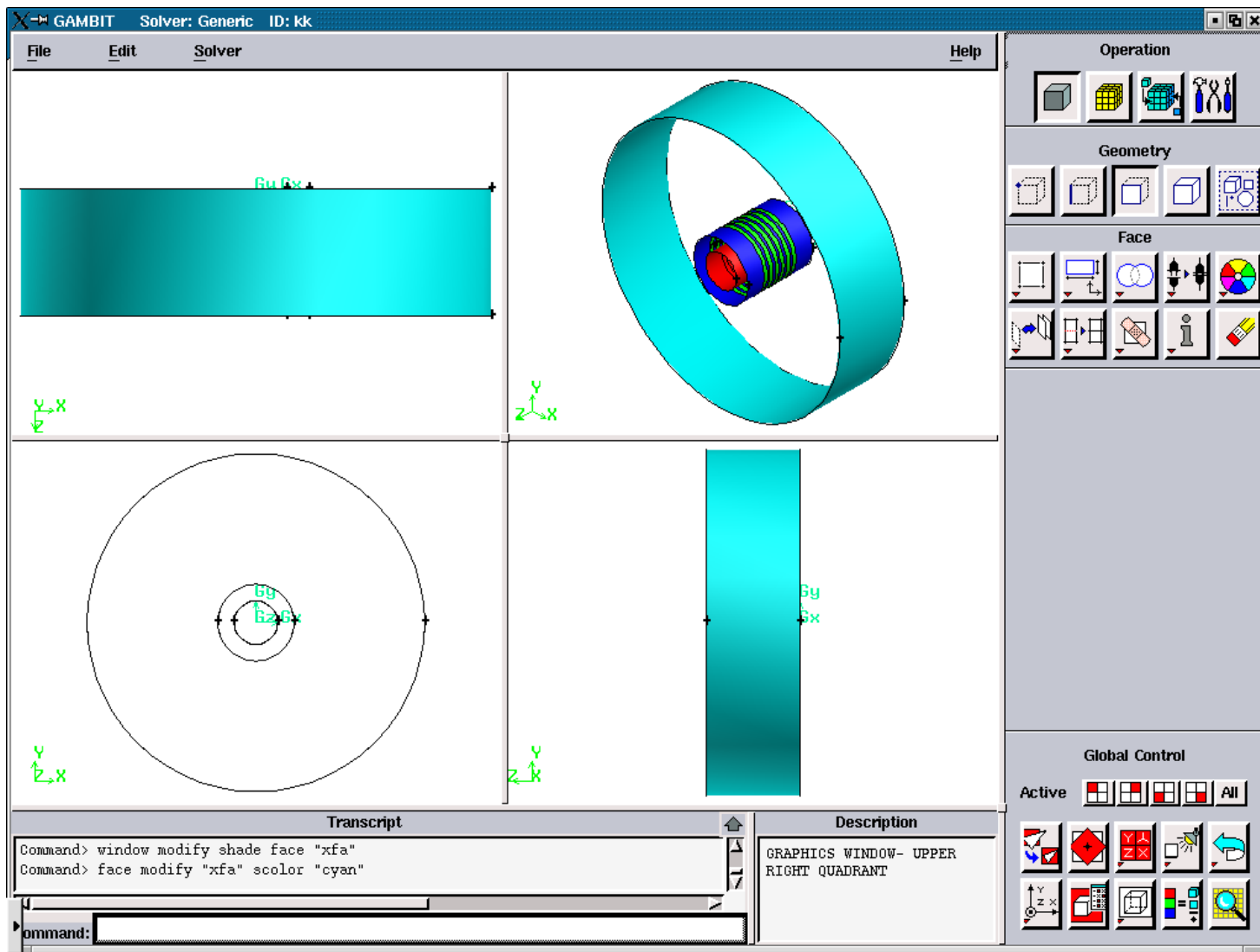
**Figure 3.5.31.** Radial Plasma Faces after the run of the journal file ‘temp1.jou’ with the slope angles: 90, 90.



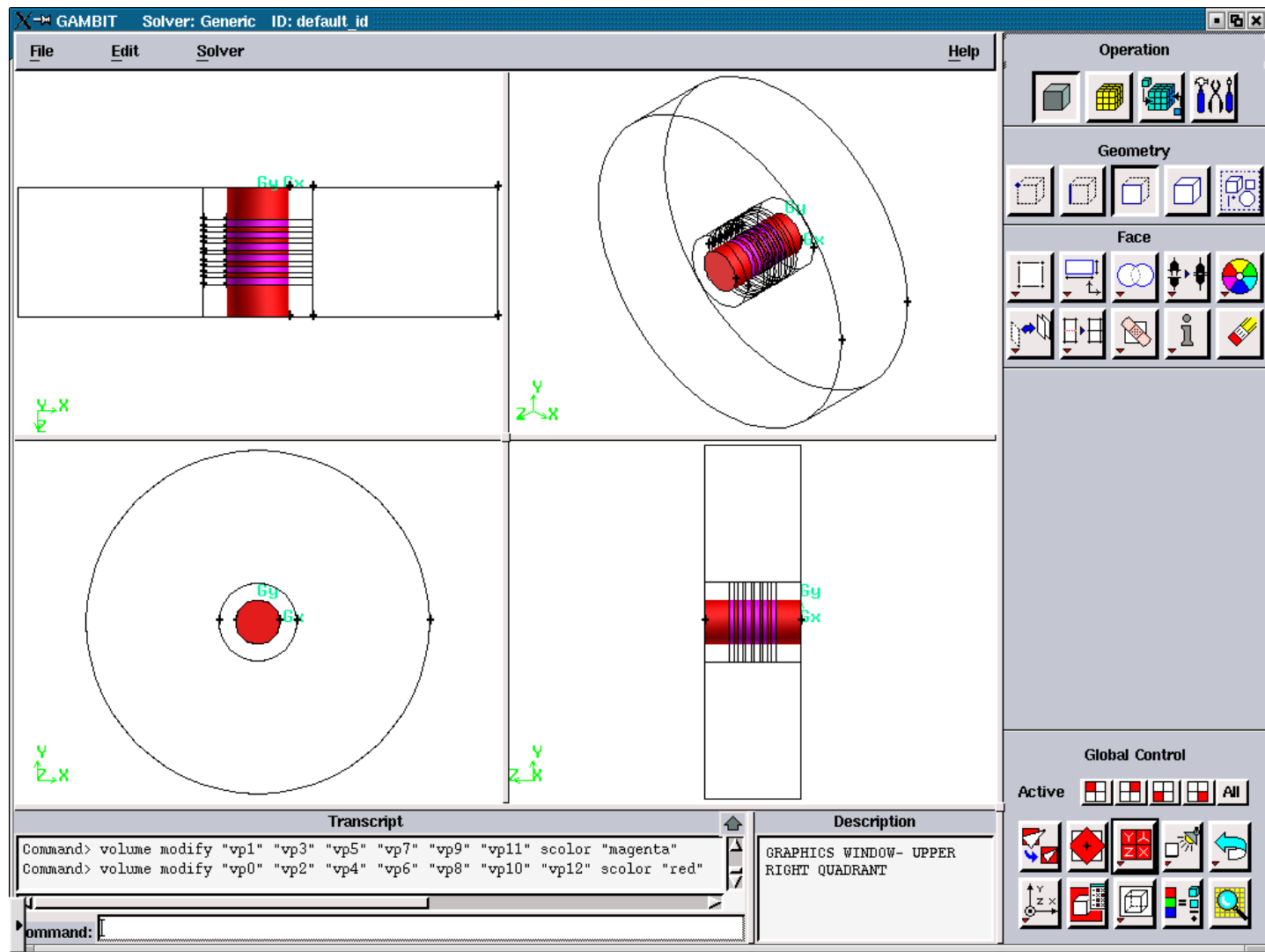
**Figure 3.5.32.** Axial Plasma Faces after the run of the journal file 'temp1.jou' with the slope angles: 90, 90.



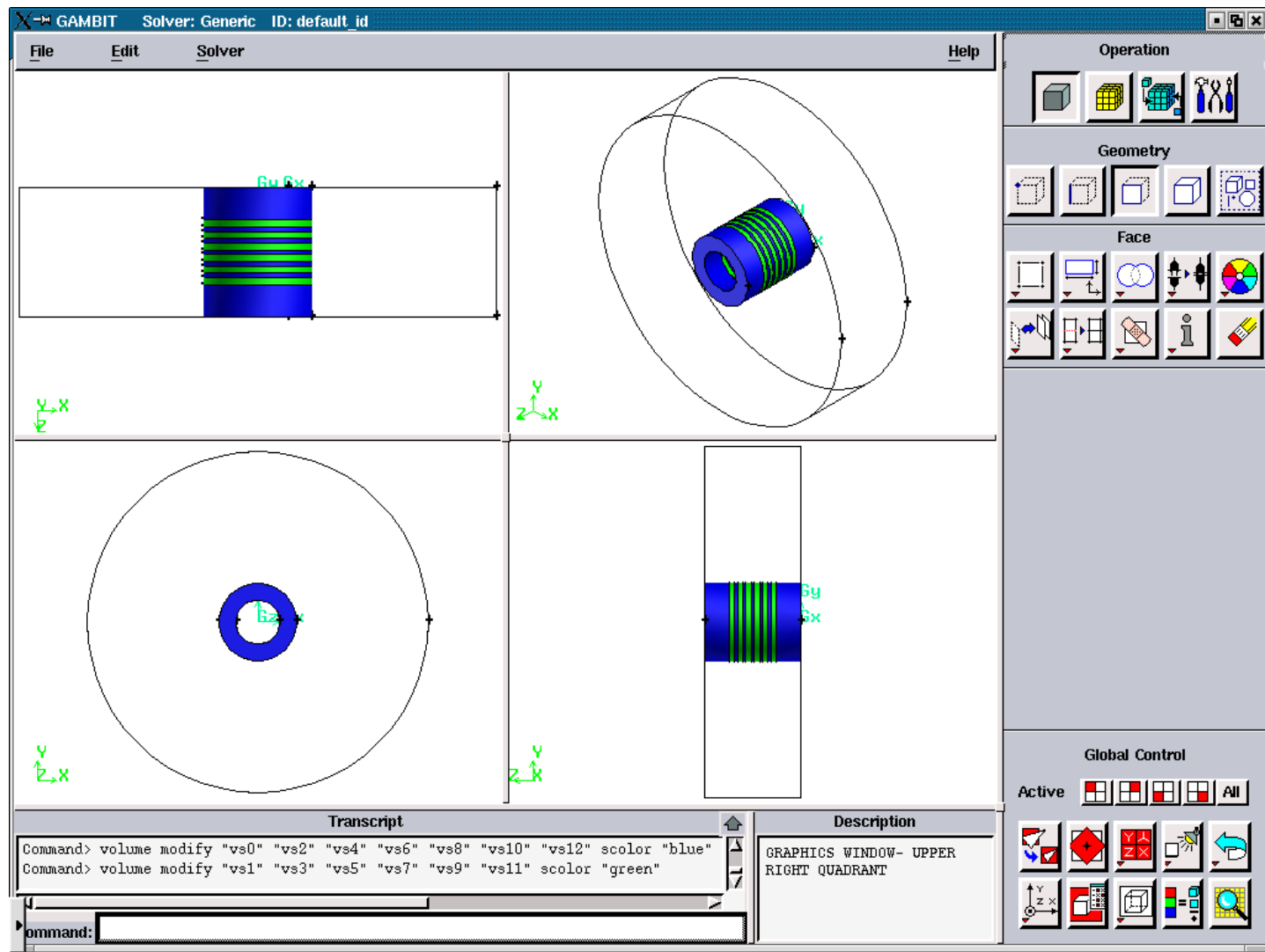
**Figure 3.5.33.** Axial Solid Faces after the run of the journal file 'temp1.jou' with the slope angles: 90, 90.



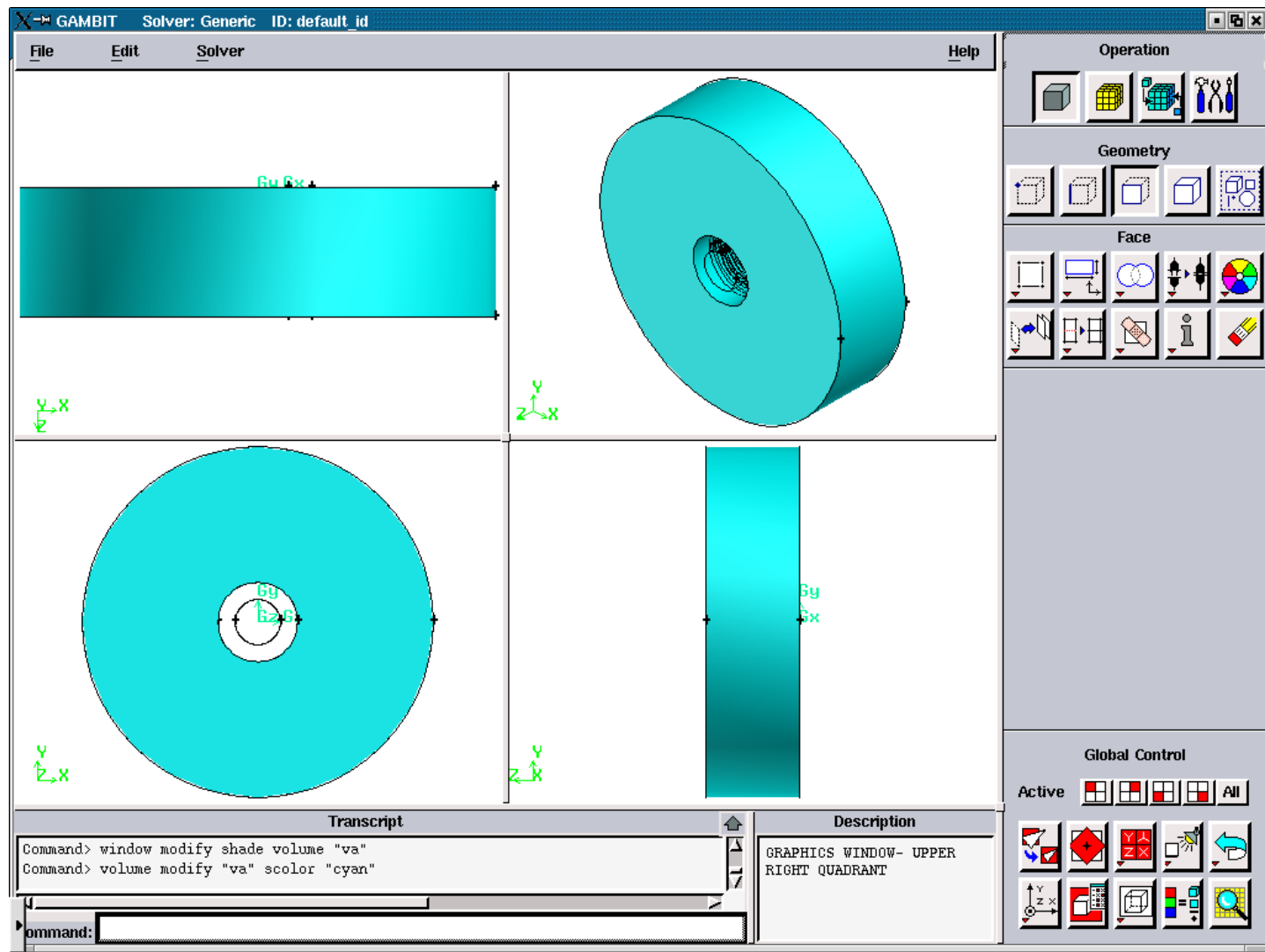
**Figure 3.5.34.** Axial Air Face after the run of the journal file 'temp1.jou' with the slope angles: 90, 90.



**Figure 3.5.35.** The Plasma Volumes after the run of the journal file 'temp1.jou' with the slope angles: 90, 90.



**Figure 3.5.36.** The Solid Volumes after the run of the journal file 'temp1.jou' with the slope angles: 90, 90.



**Figure 3.5.37.** The Air Volume after the run of the journal file 'templ.jou' with the slope angles: 90, 90.

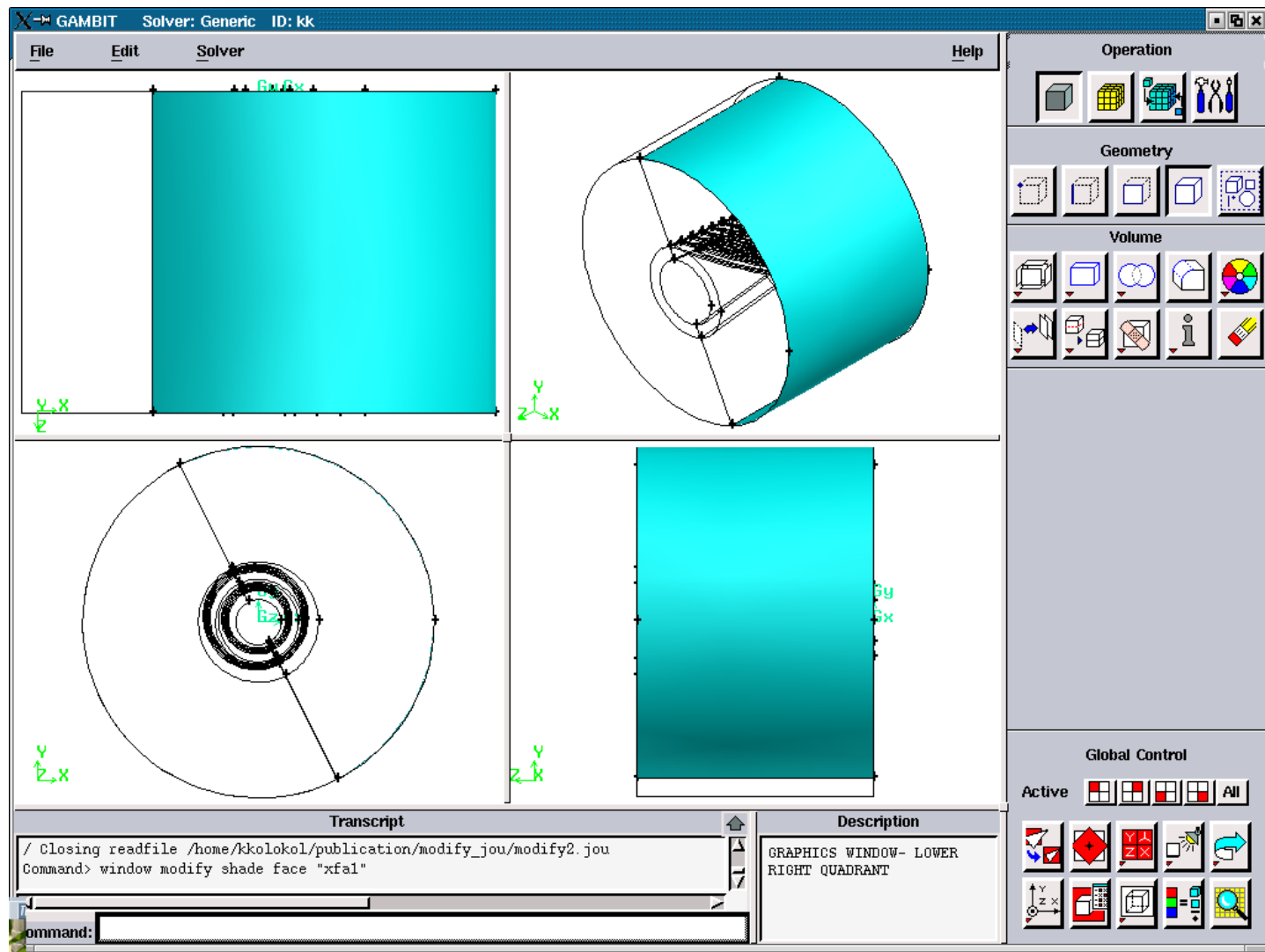
#### **A1.4. FIGURES ILLUSTRATING THE CREATION OF THE FILE ‘modify2’**

##### **A1.4.1. FIGURES FOR THE SLOPE ANGLES: 45, 60**

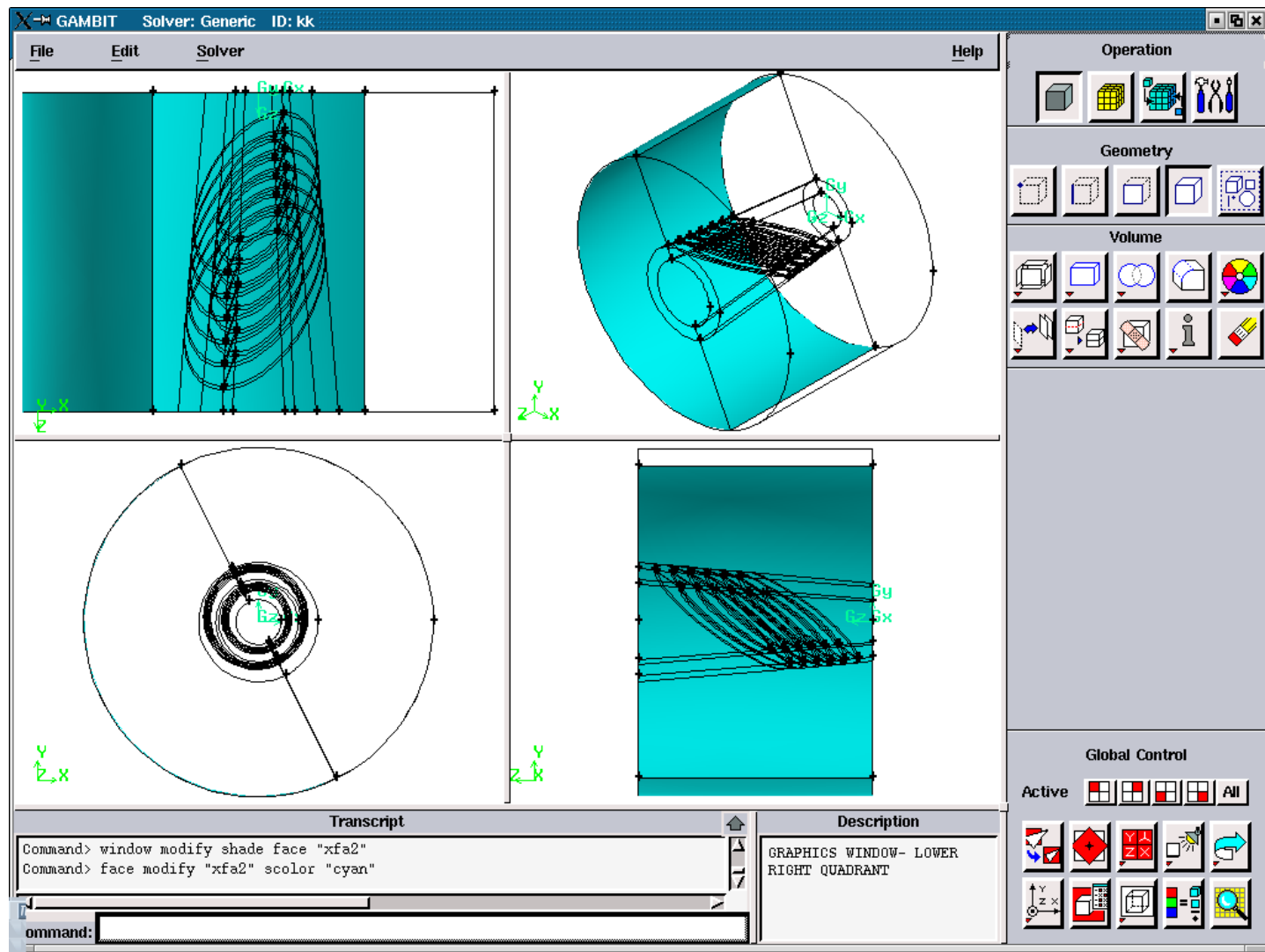
---

All figures in this section represent the geometry after the consequent run of the files:

‘temp1.jou’, ‘modify1’, temp2.jou’



**Figure 3.5.38.** Axial Air Faces after the run of the journal files consequently 'temp1.jou', 'modify1', 'temp2.jou'.



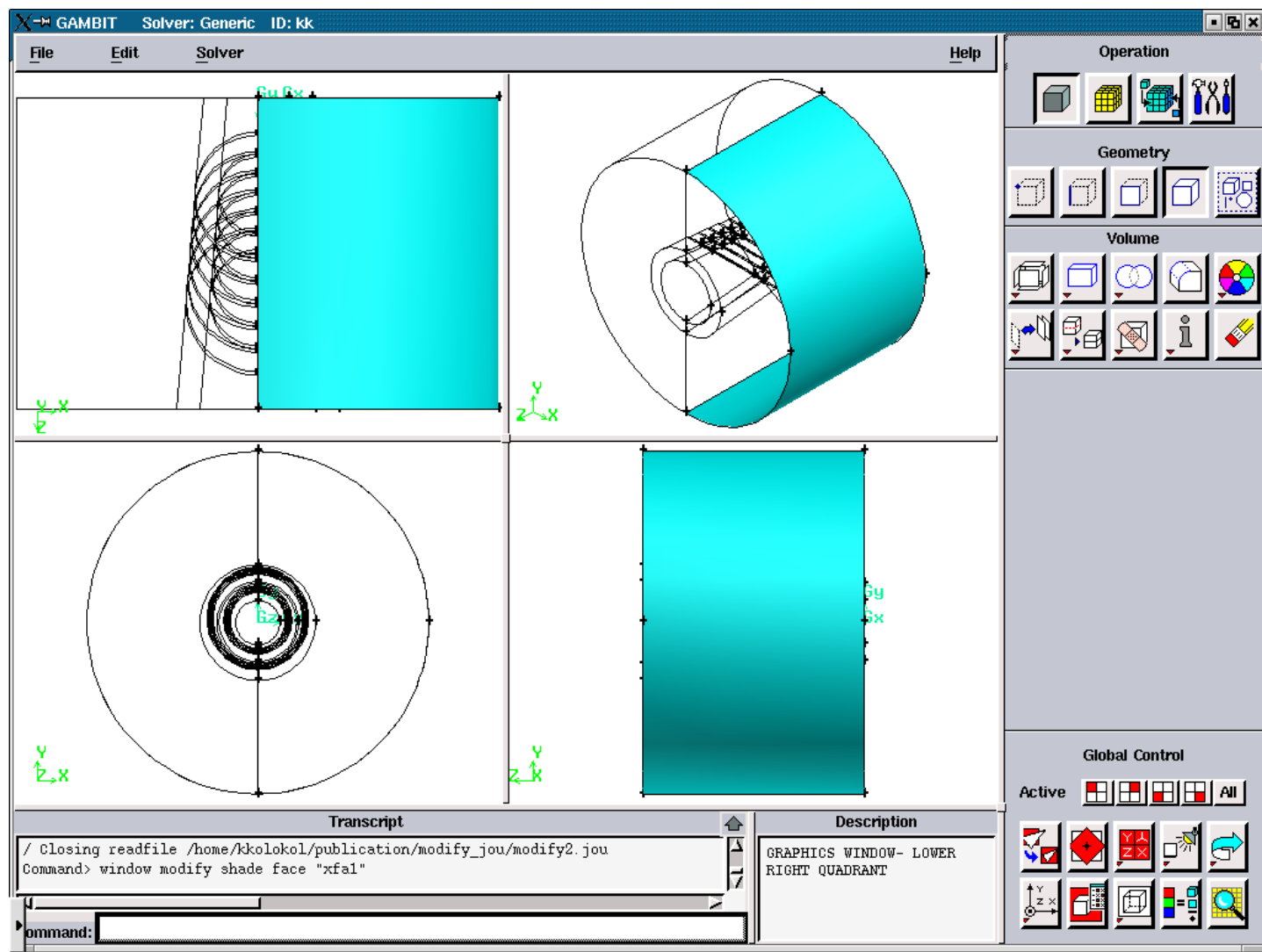
**Figure 3.5.39.** Axial Air Faces after the run of the journal files consequently 'temp1.jou', 'modify1', 'temp2.jou'.

#### **A1.4.2. FIGURES FOR THE SLOPE ANGLES: 45, 90**

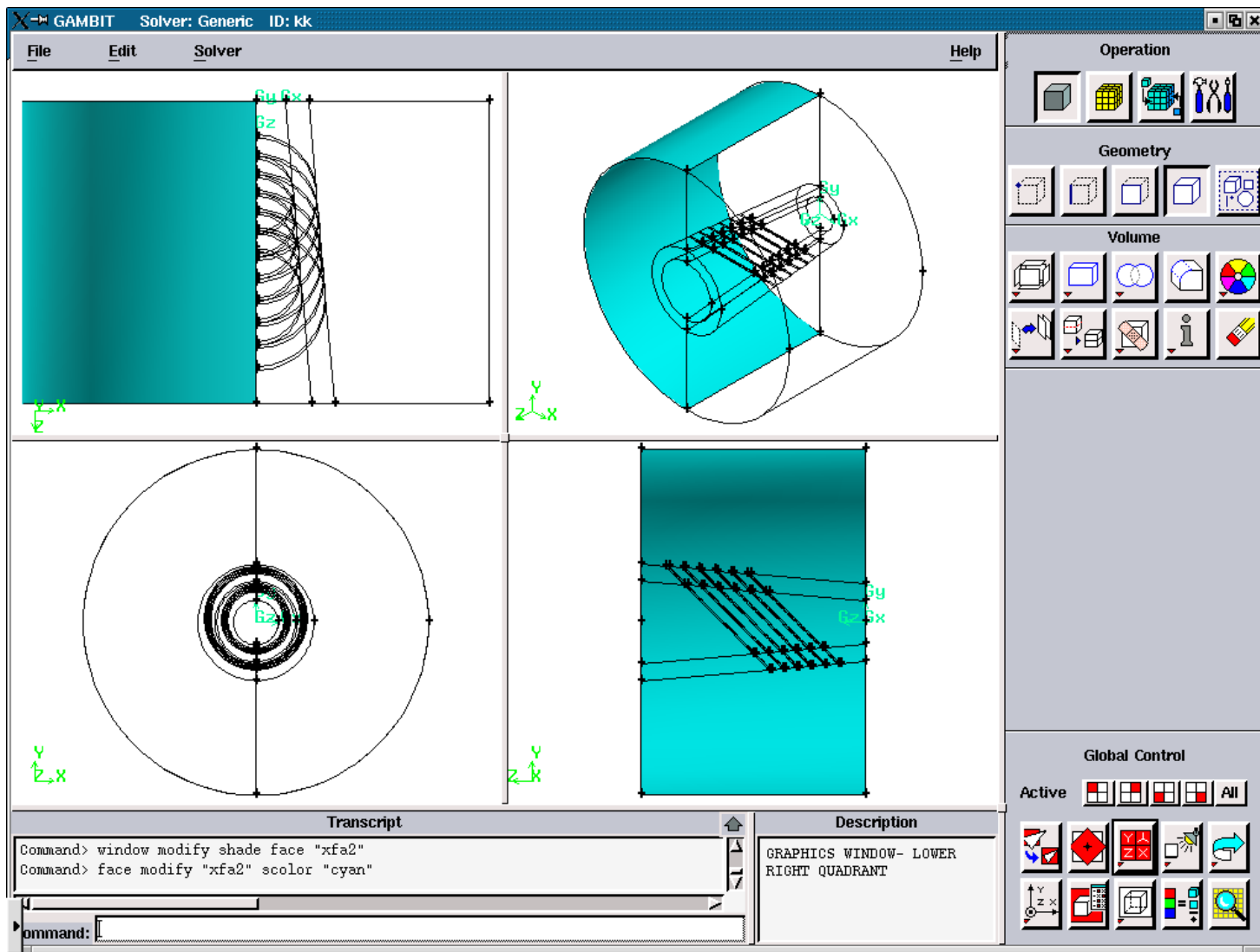
---

All figures in this section represent the geometry after the consequent run of the files:

‘temp1.jou’, ‘modify1’, temp2.jou’



**Figure 3.5.40.** Axial Air Faces after the consequent run of the files: 'temp1.jou', 'modify1', 'temp2.jou'.



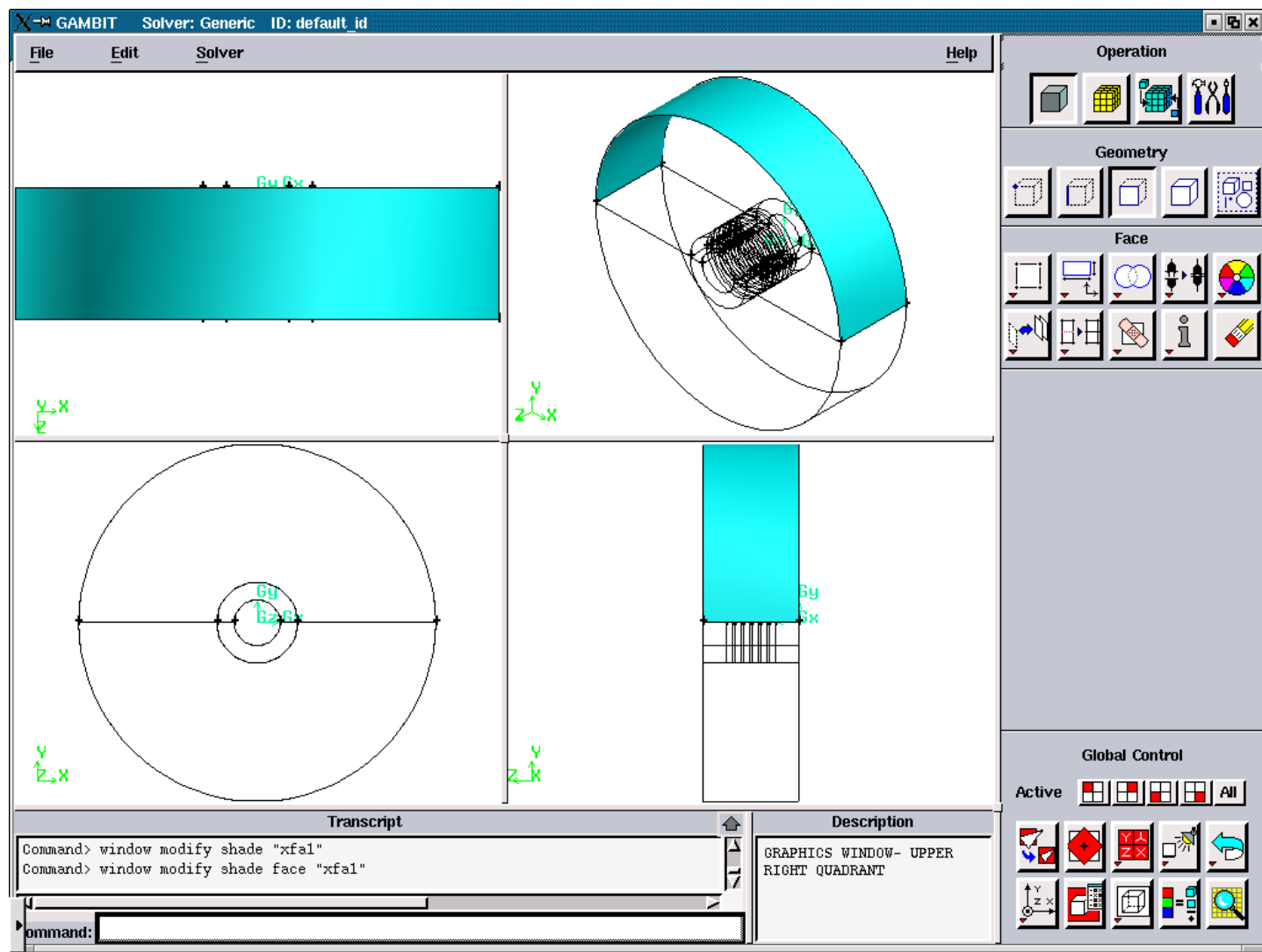
**Figure 3.5.41.** Axial Air Faces after the consequent run of the files: 'temp1.jou', 'modify1', temp2.jou'.

### **A1.4.3. FIGURES FOR THE SLOPE ANGLES: 90, 90**

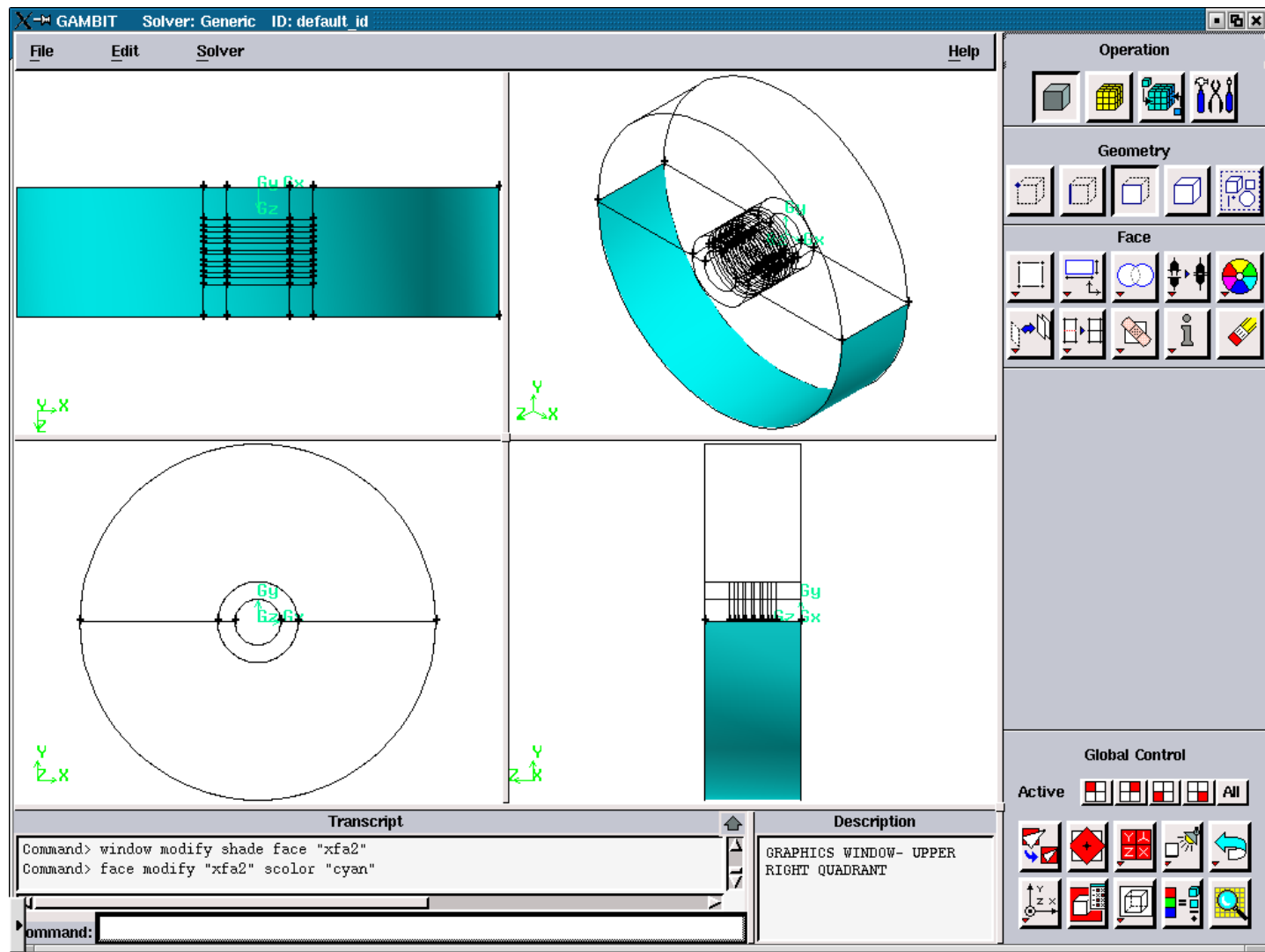
---

All figures in this section represent the geometry after the consequent run of the files:

‘temp1.jou’, ‘modify1’, temp2.jou’



**Figure 3.5.42.** Axial Air Faces after the consequent run of the files: 'temp1.jou', 'modify1', 'temp2.jou'.



**Figure 3.5.43.** Axial Air Faces after the consequent run of the files: 'temp1.jou', 'modify1', 'temp3.jou'.

## **A1.5. FIGURES ILLUSTRATING THE CREATION OF THE FILE**

**‘modify3\_d’ and ‘modify3\_s’**

### **A1.5.1. FIGURES FOR THE CTREATION OF THE FILE ‘modify3\_d’**

**FOR THE SLOPE ANGLES: 45, 60**

---

All figures in this section represent the geometry after the consequent run of the files:

‘temp1.jou’, ‘modify1’, temp2.jou’, modify2, ‘temp3.jou’

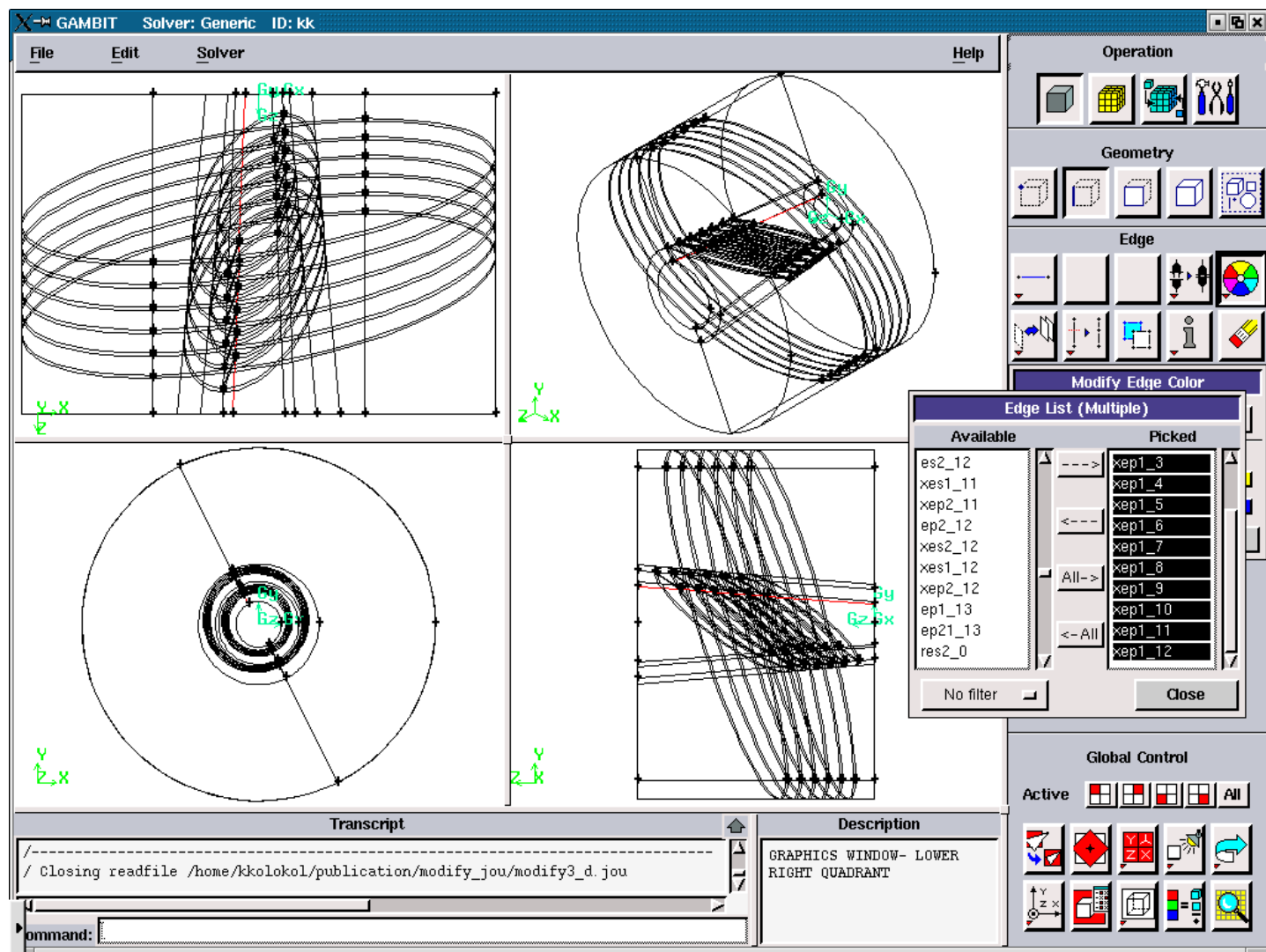


Figure 3.5.44. Axial Plasma Edges for the “maximum”.

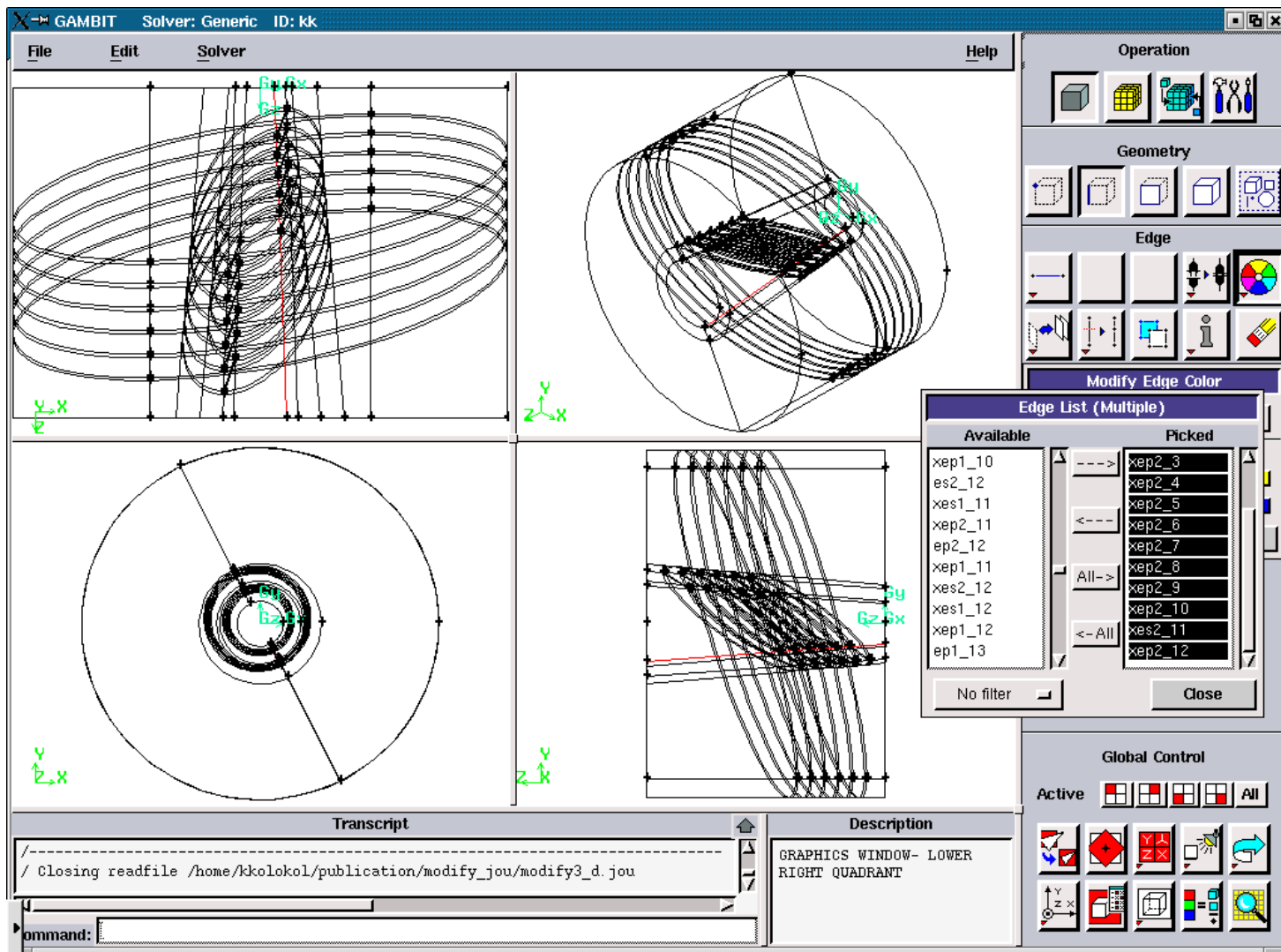


Figure 3.5.45. Axial Plasma Edges for the “minimum”.

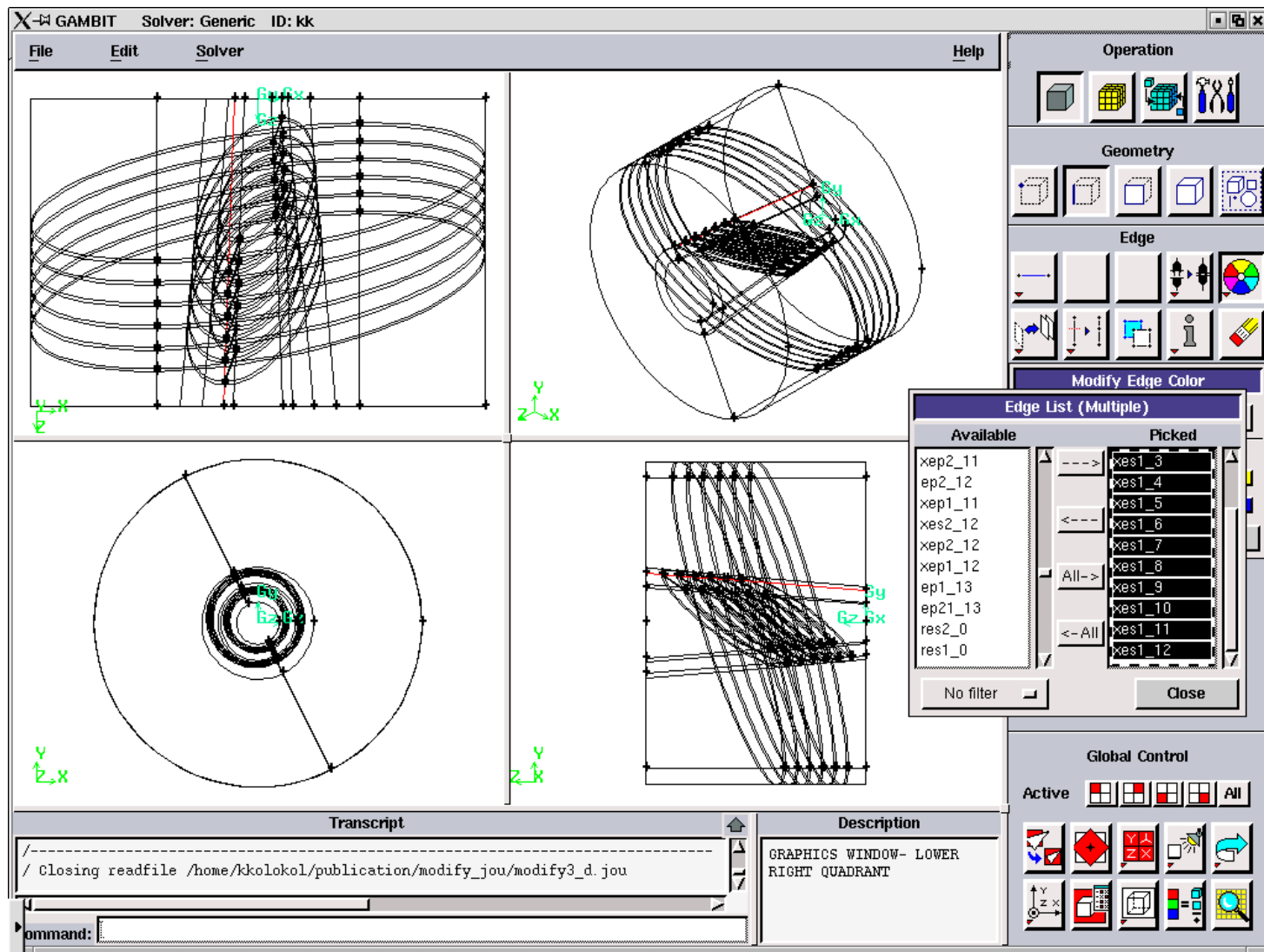


Figure 3.5.46. Axial Solid Edges for the “maximum”.

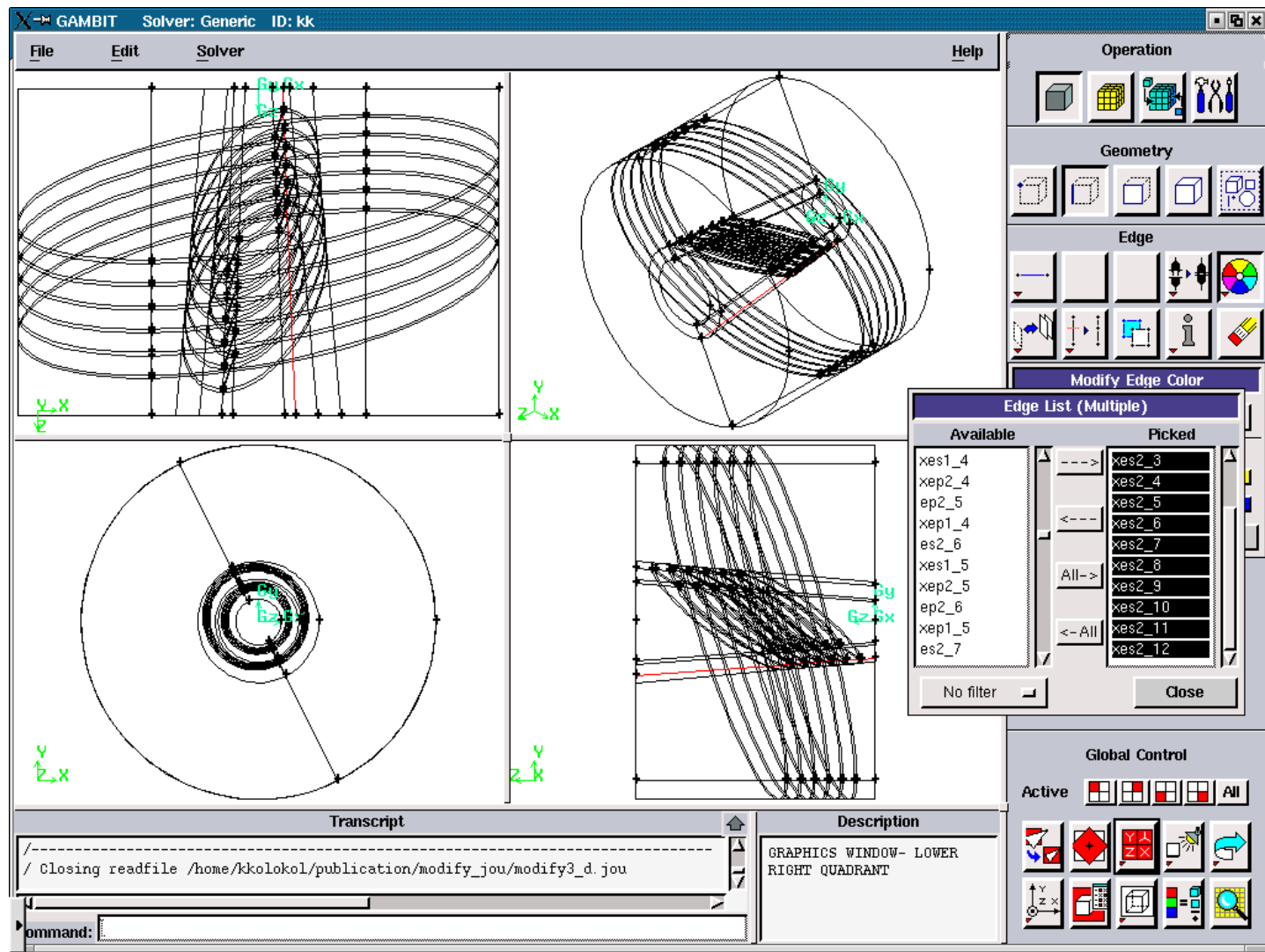


Figure 3.5.47. Axial Solid Edges for the “minimum”.

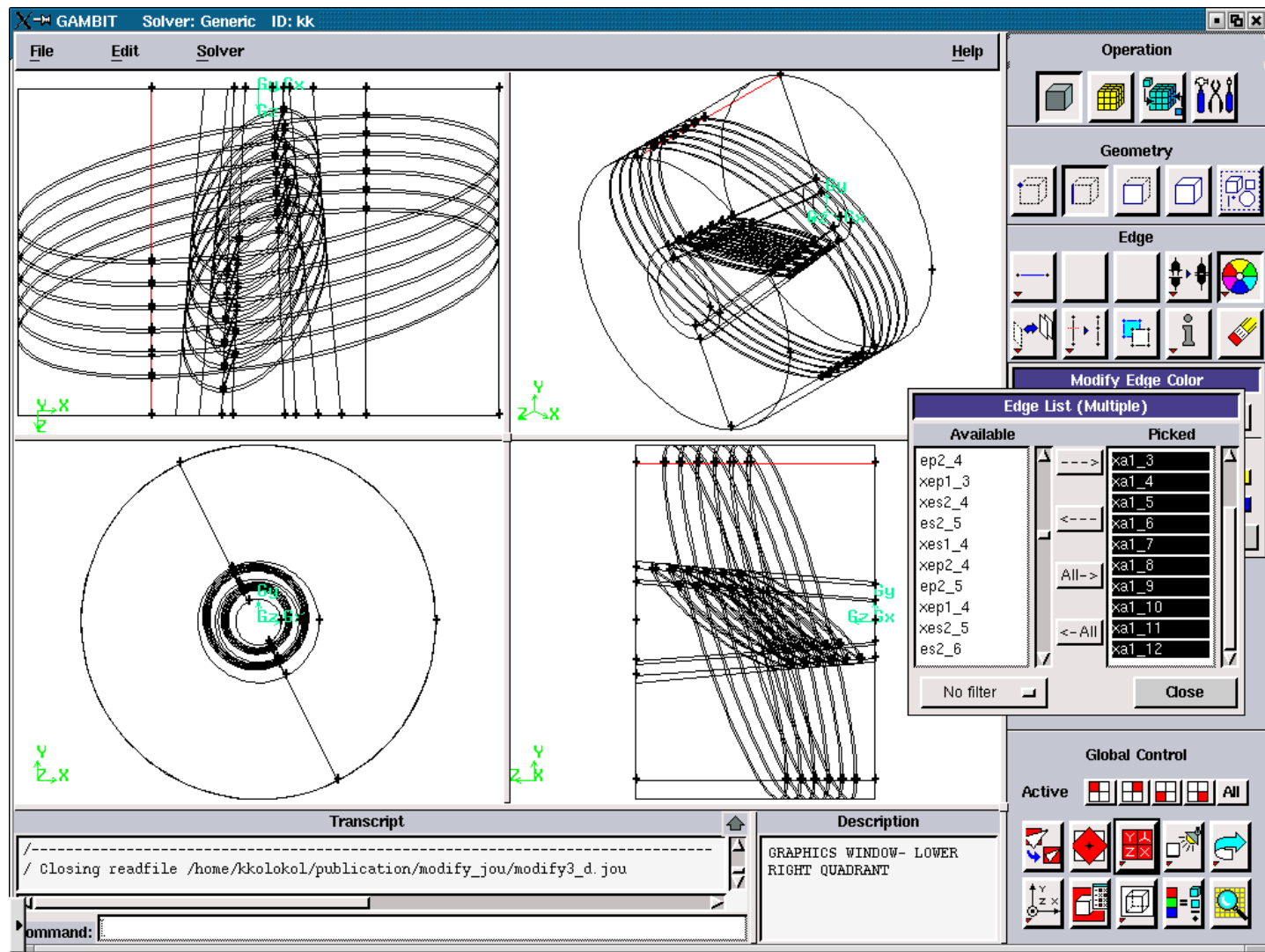


Figure 3.5.48. Axial Air Edges after for the “maximum”.

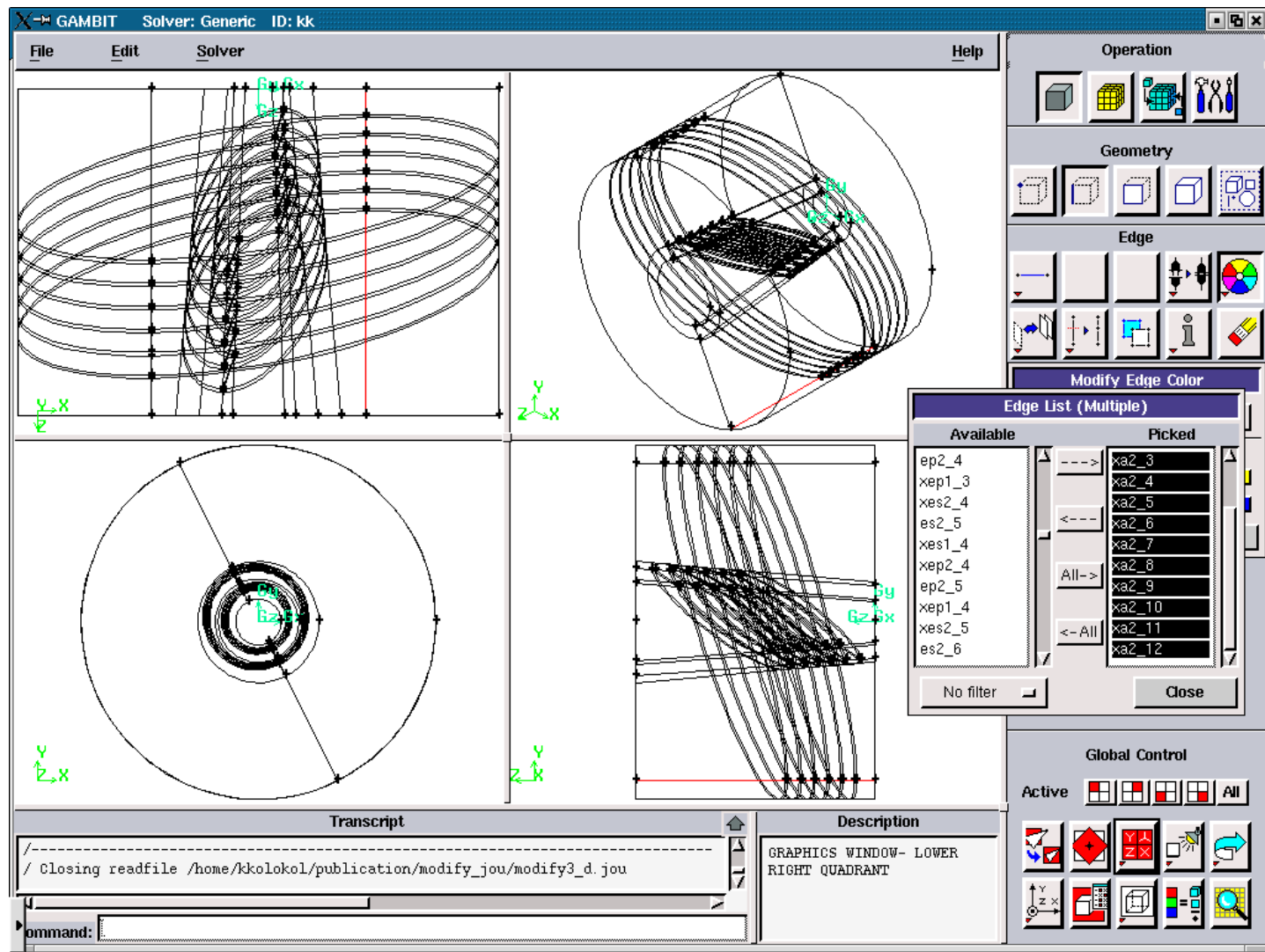


Figure 3.5.49. Axial Air Edges for the “minimum”.

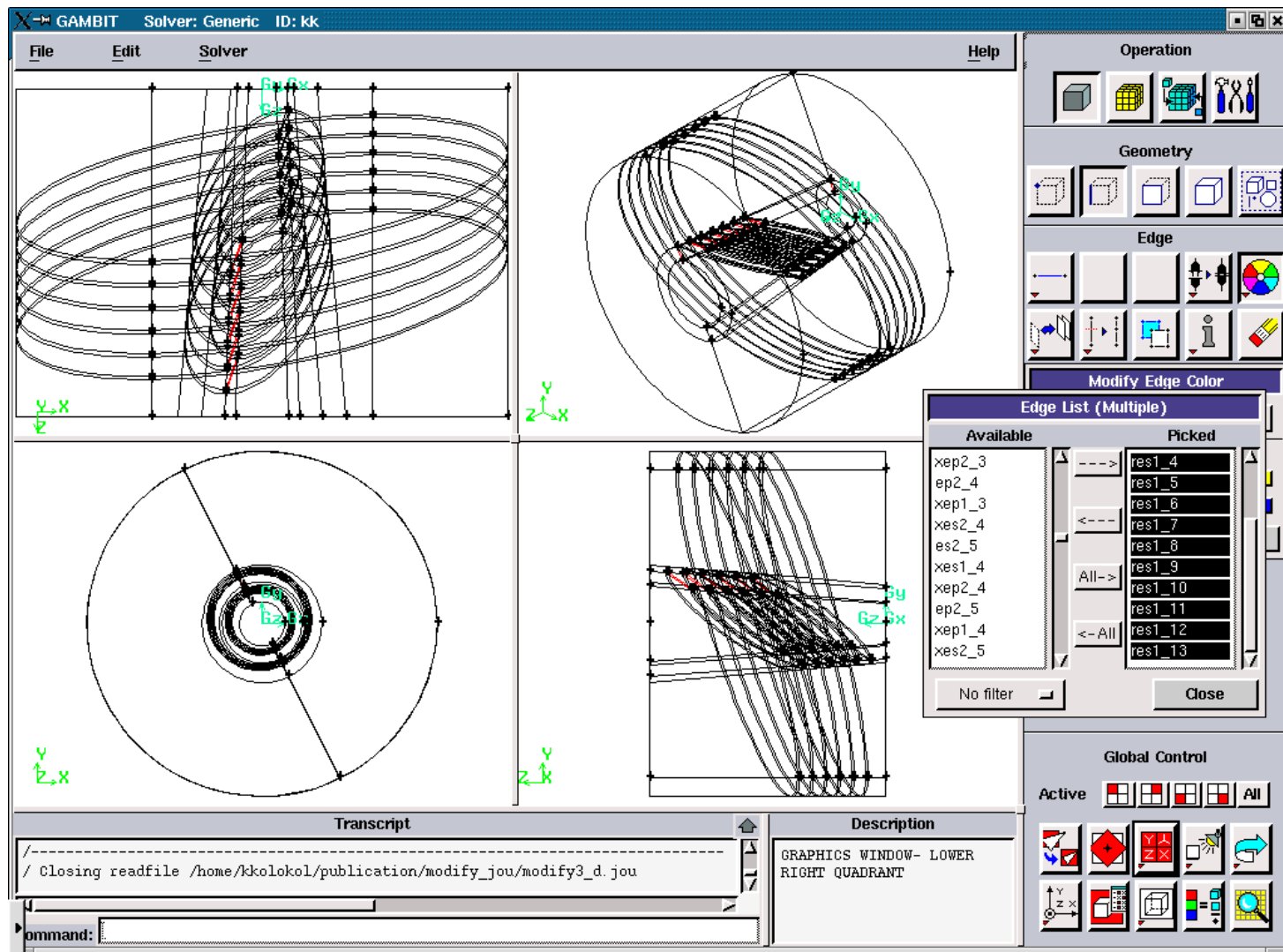


Figure 3.5.50. Radial Solid Edges for the “maximum”.

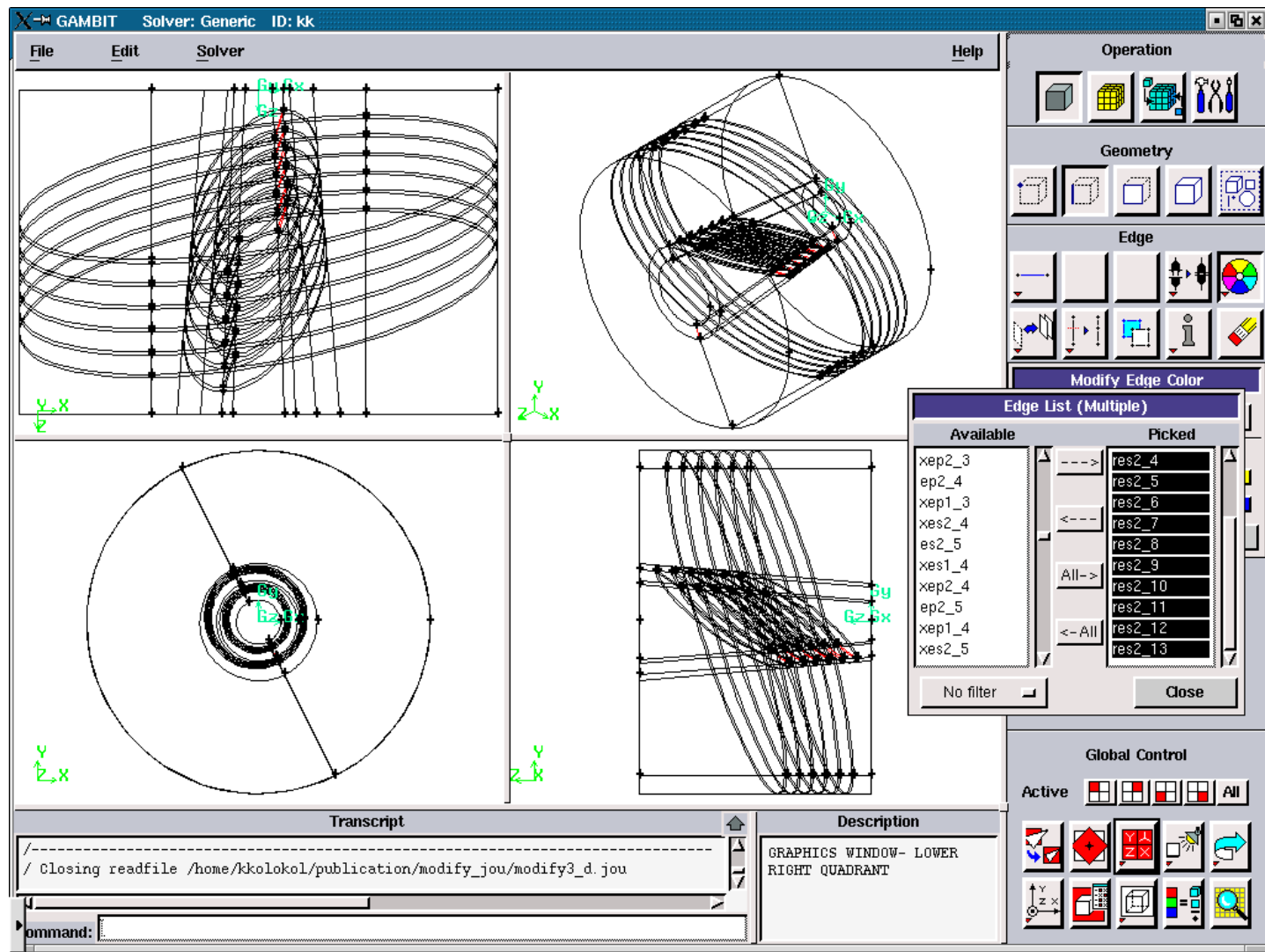


Figure 3.5.51. Radial Solid Edges for the “minimum”.

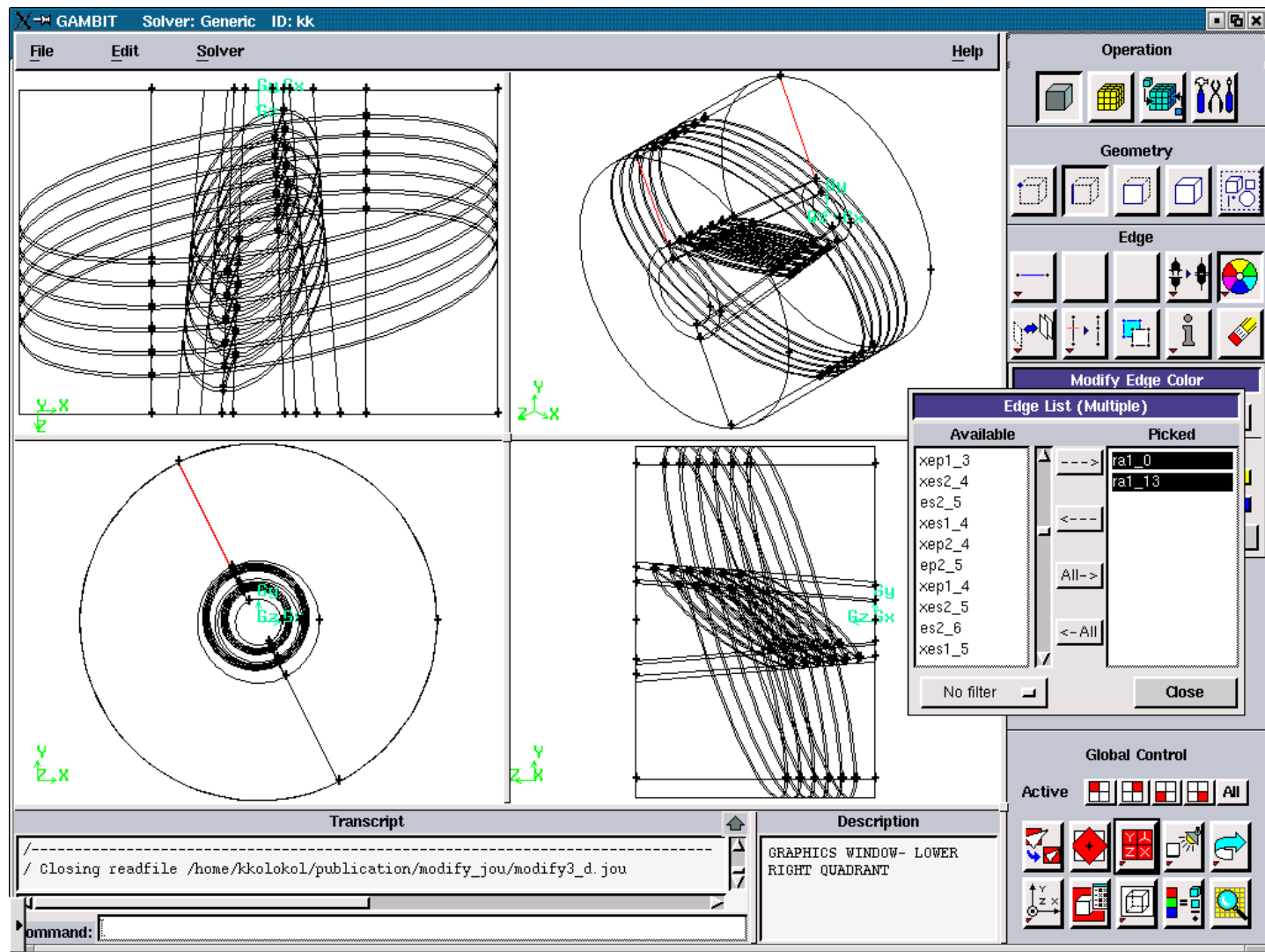


Figure 3.5.52. Radial Air Edges for the “maximum”.

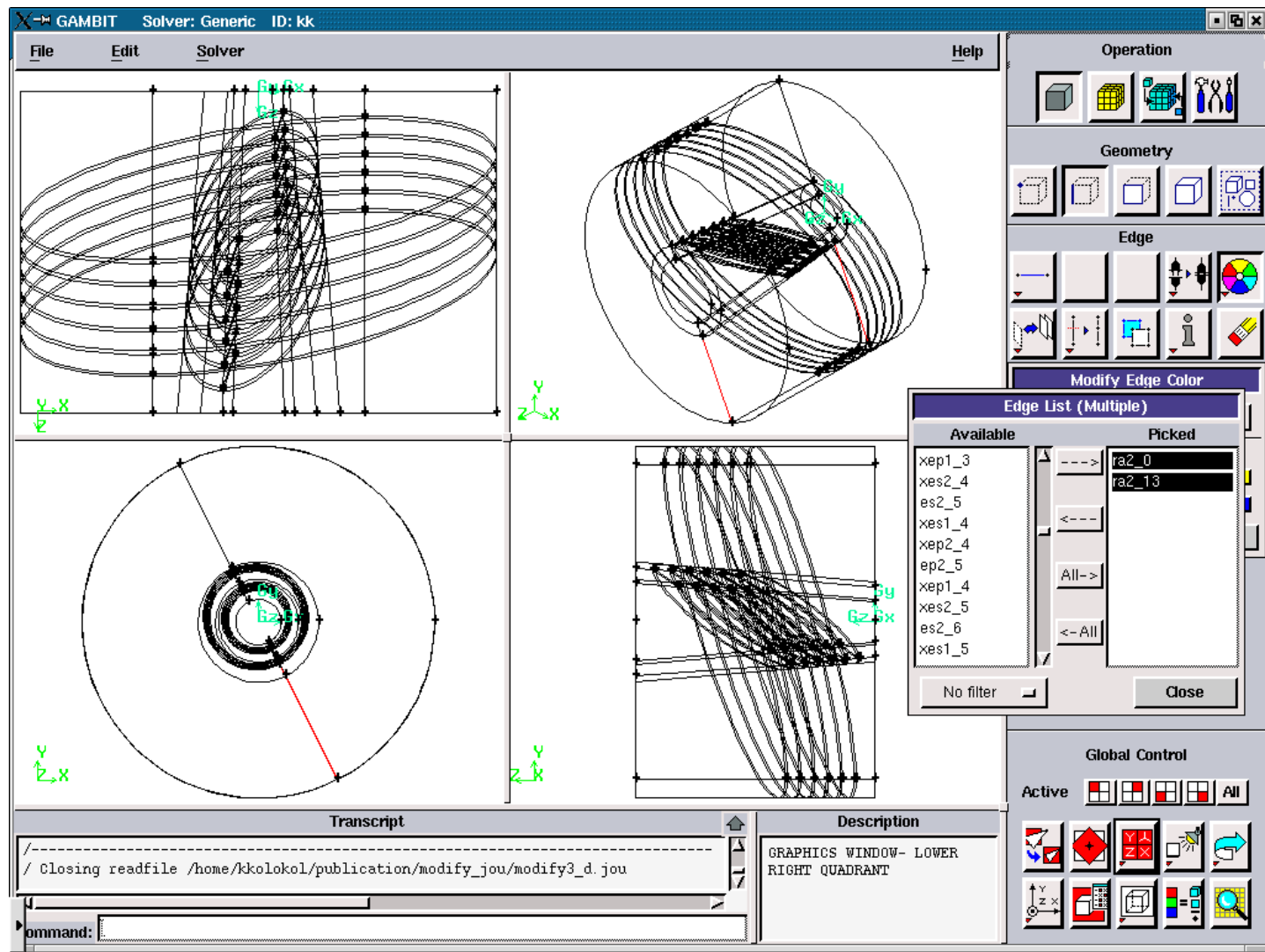


Figure 3.5.53. Radial Air Edges for the “minimum”.

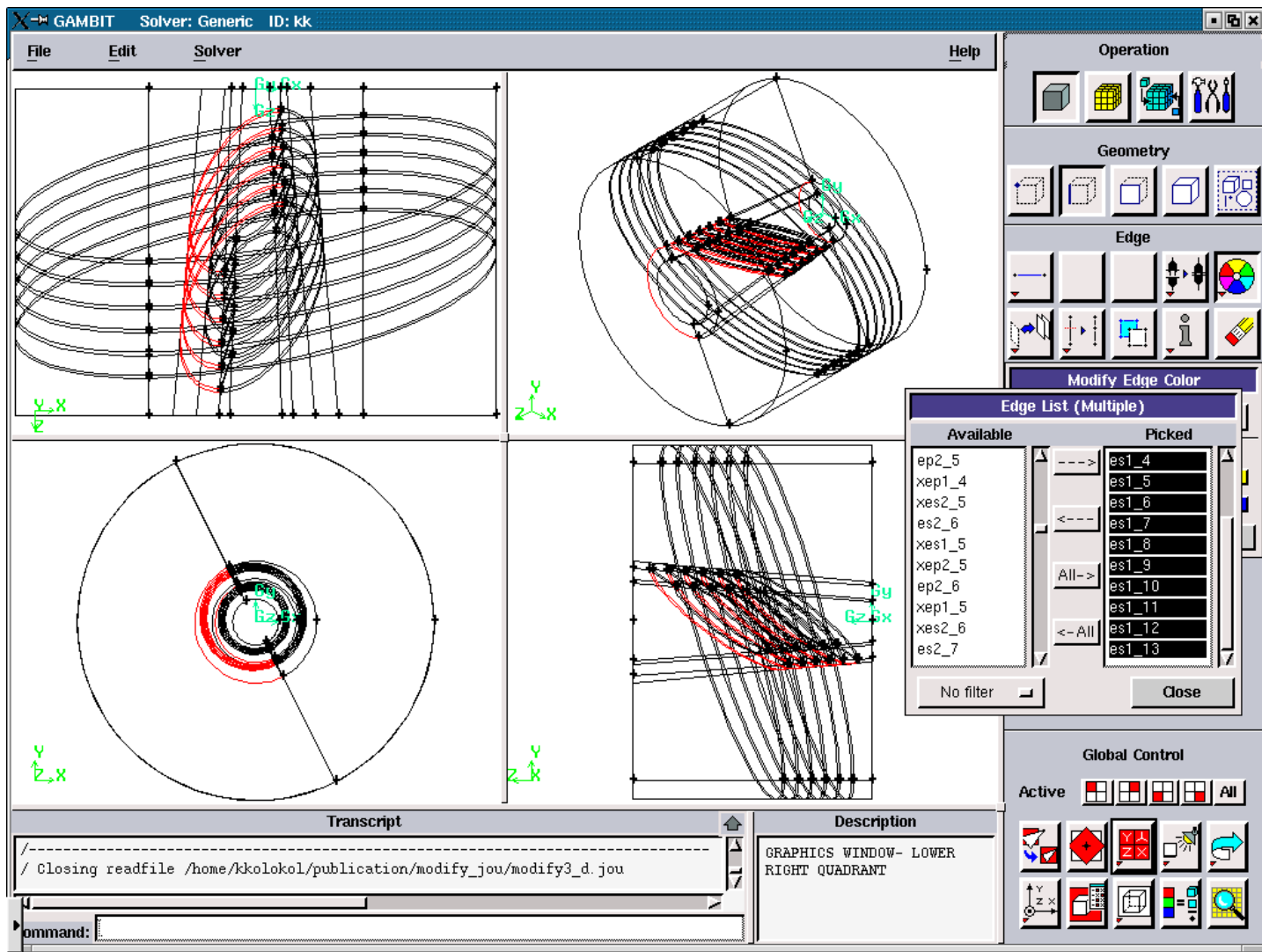


Figure 3.5.54. Elliptic Solid Edges for the “left”.

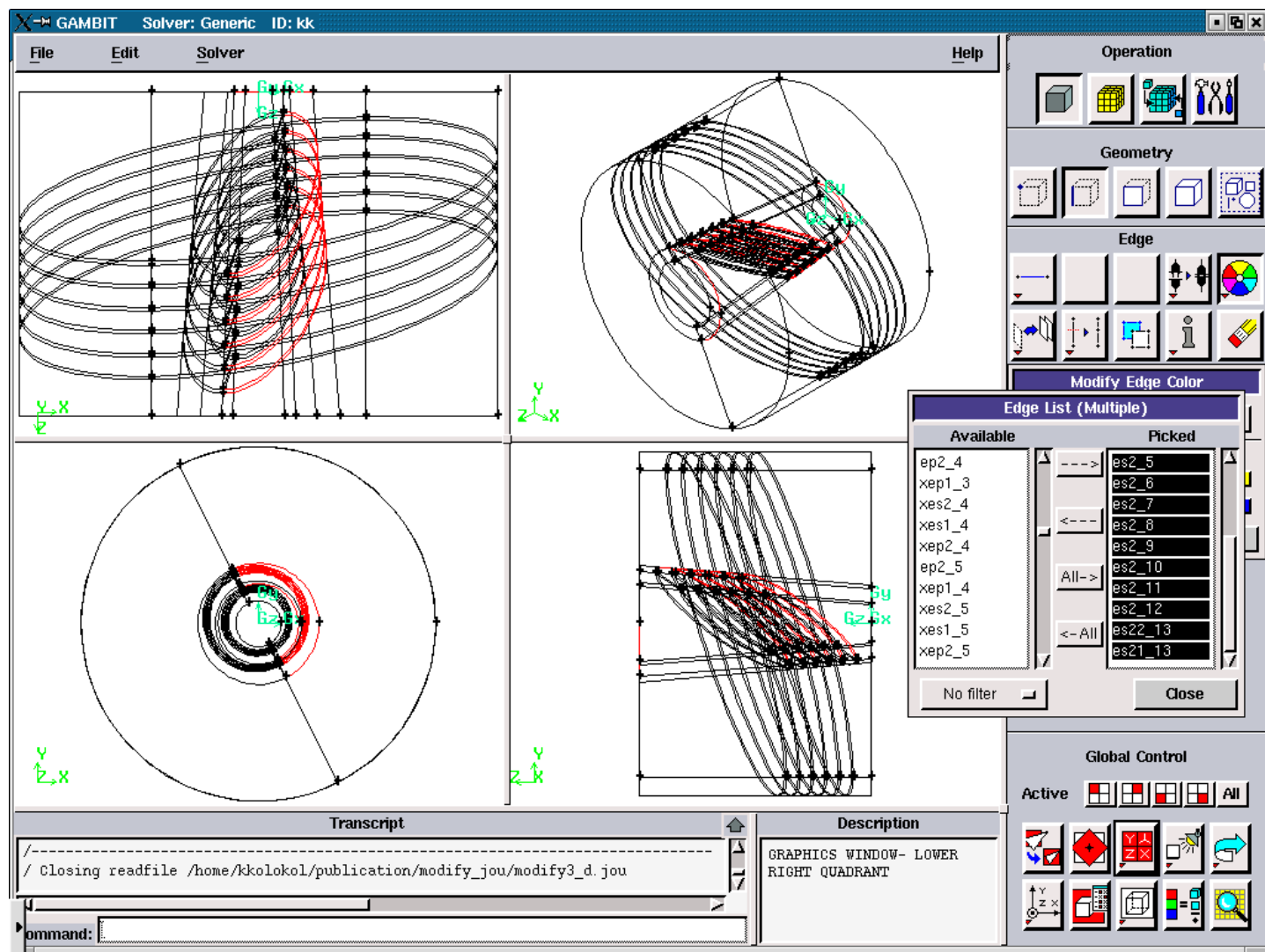


Figure 3.5.55. Elliptic Solid Edges for the “right”.

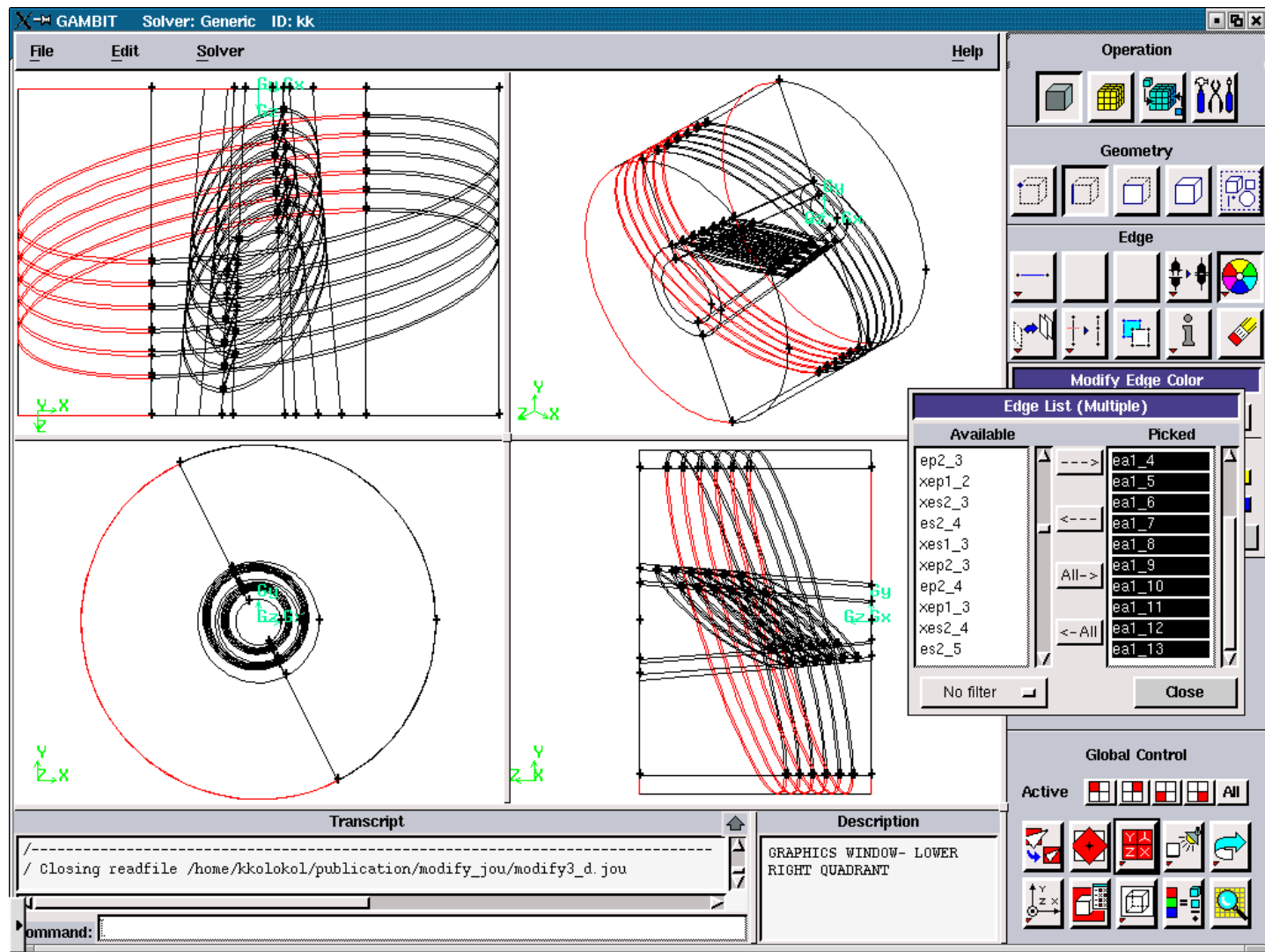


Figure 3.5.56. Elliptic Air Edges for the “left”.

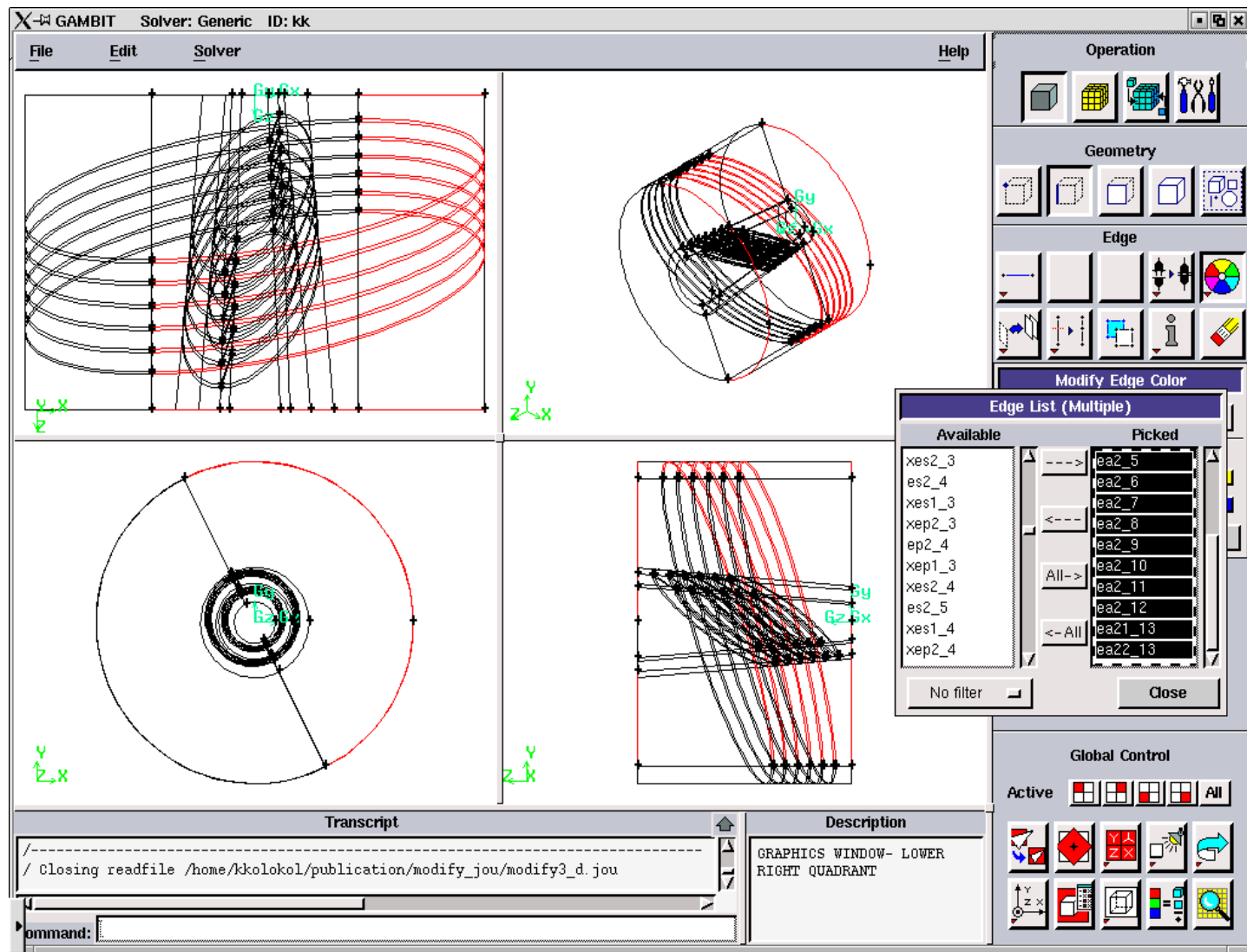


Figure 3.5.57. Elliptic Air Edges for the “right”.

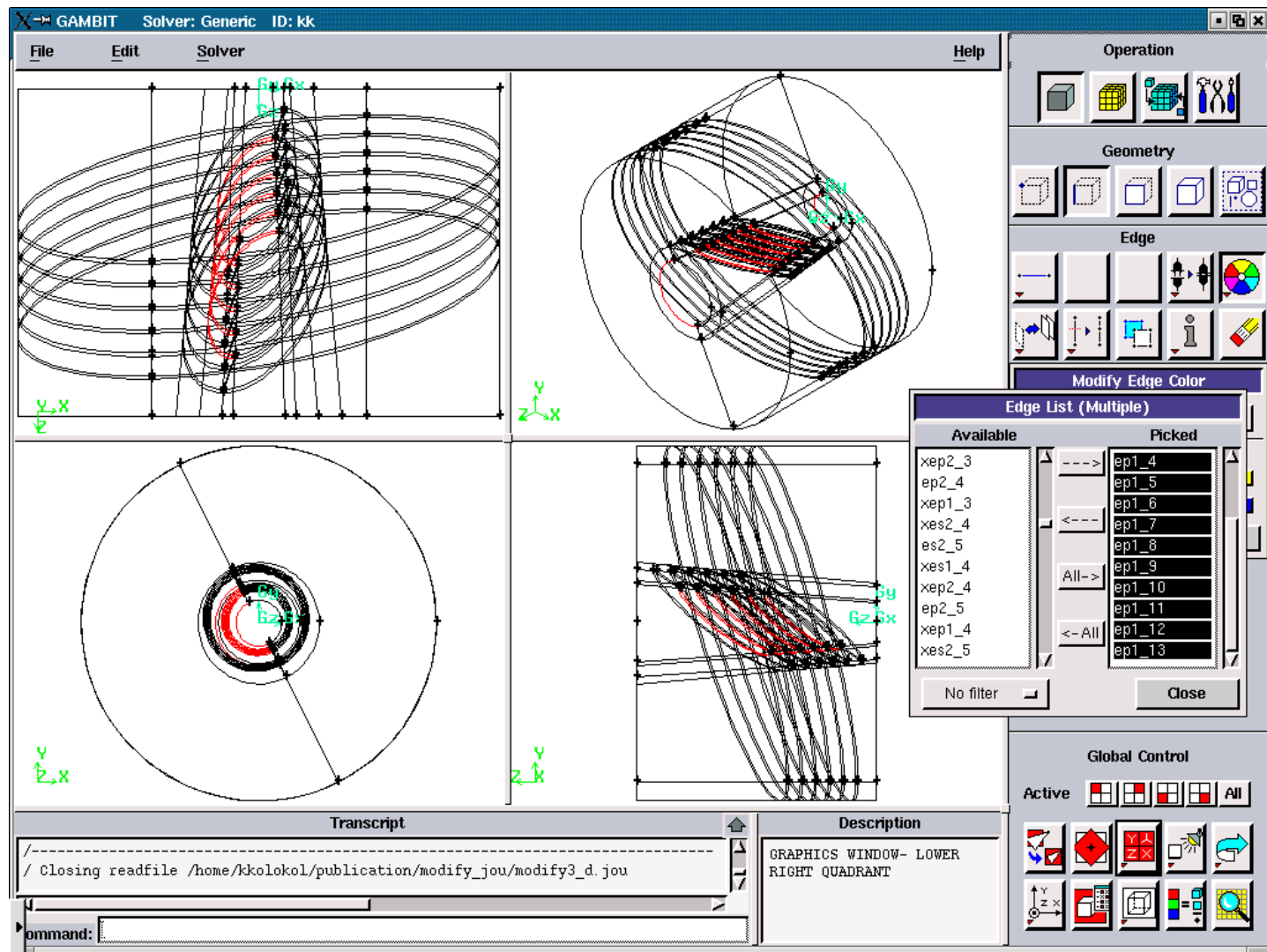


Figure 3.5.58. Elliptic Plasma Edges for the “left”.

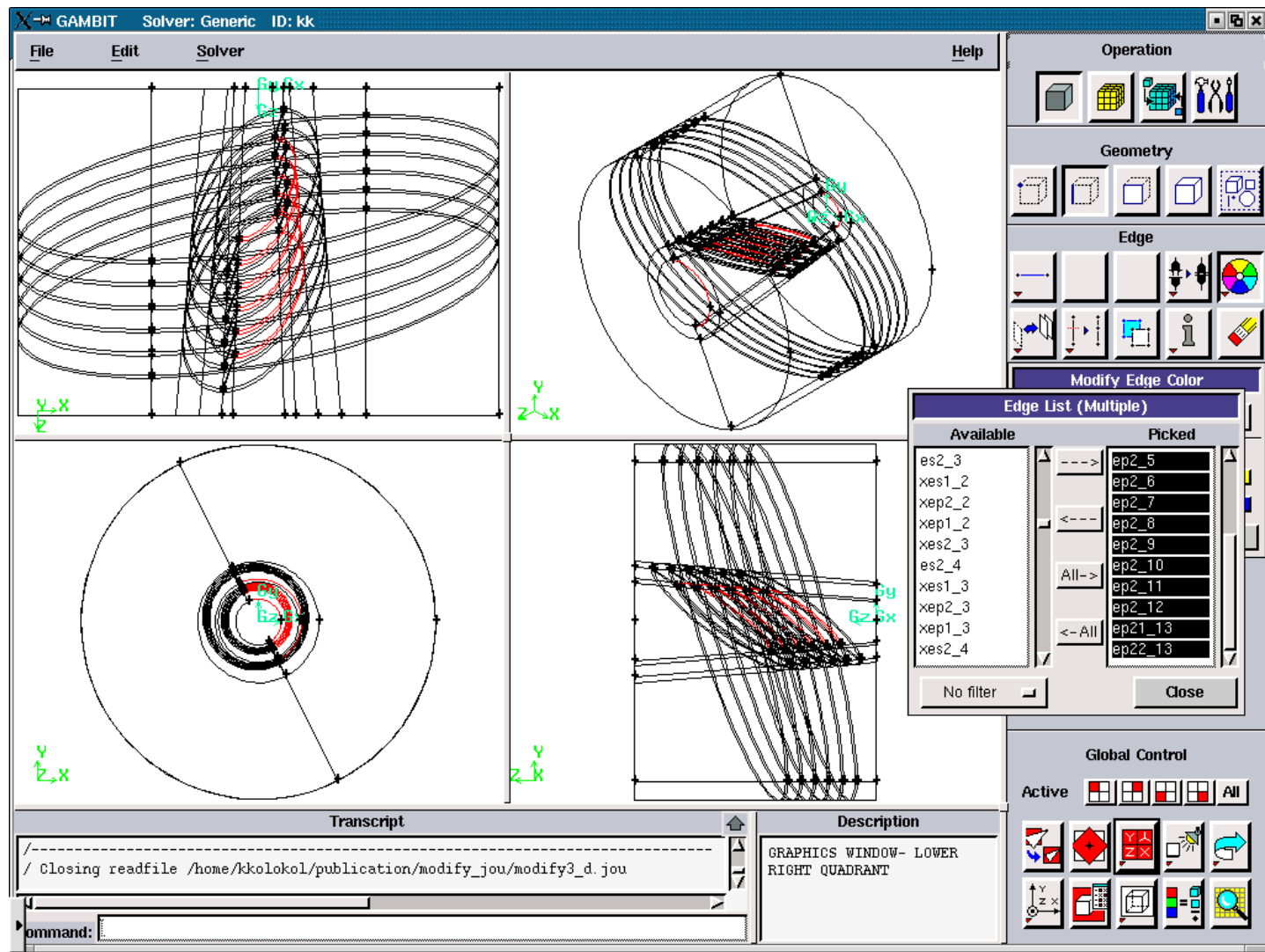


Figure 3.5.59. Elliptic Plasma Edges for the “right”.

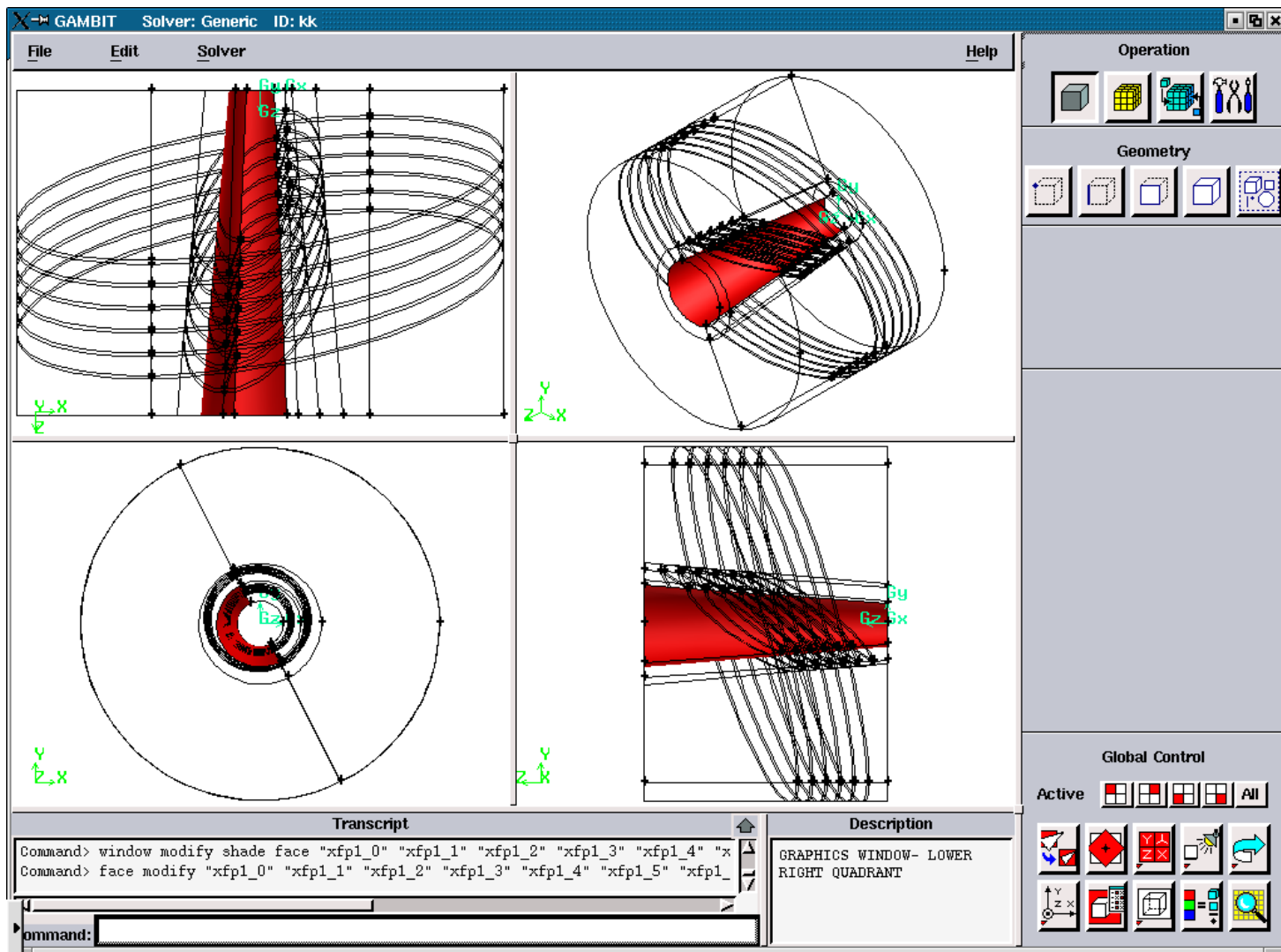


Figure 3.5.60. Axial Plasma Faces for the “left”.

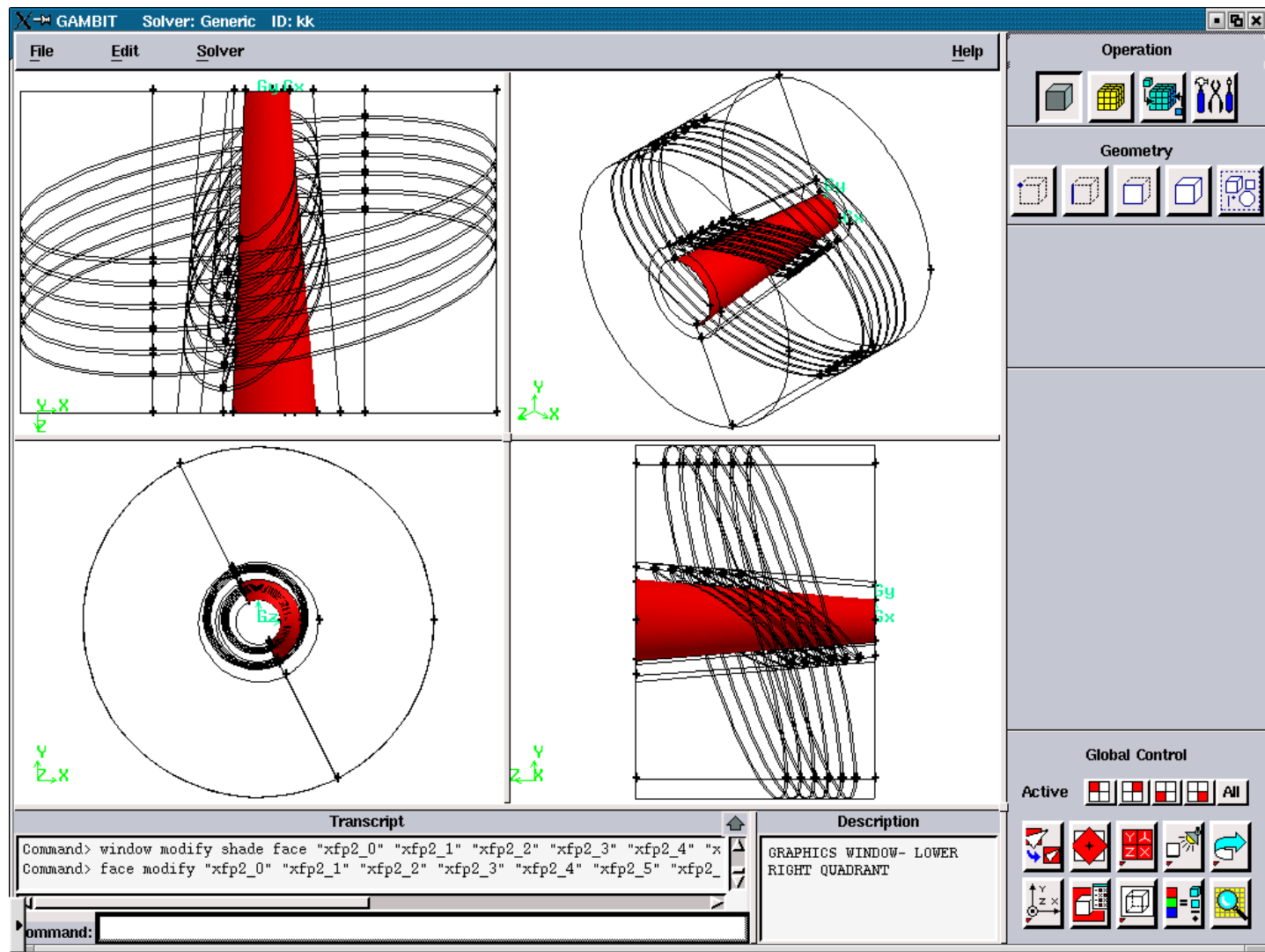


Figure 3.5.61. Axial Plasma Faces for the “left”.

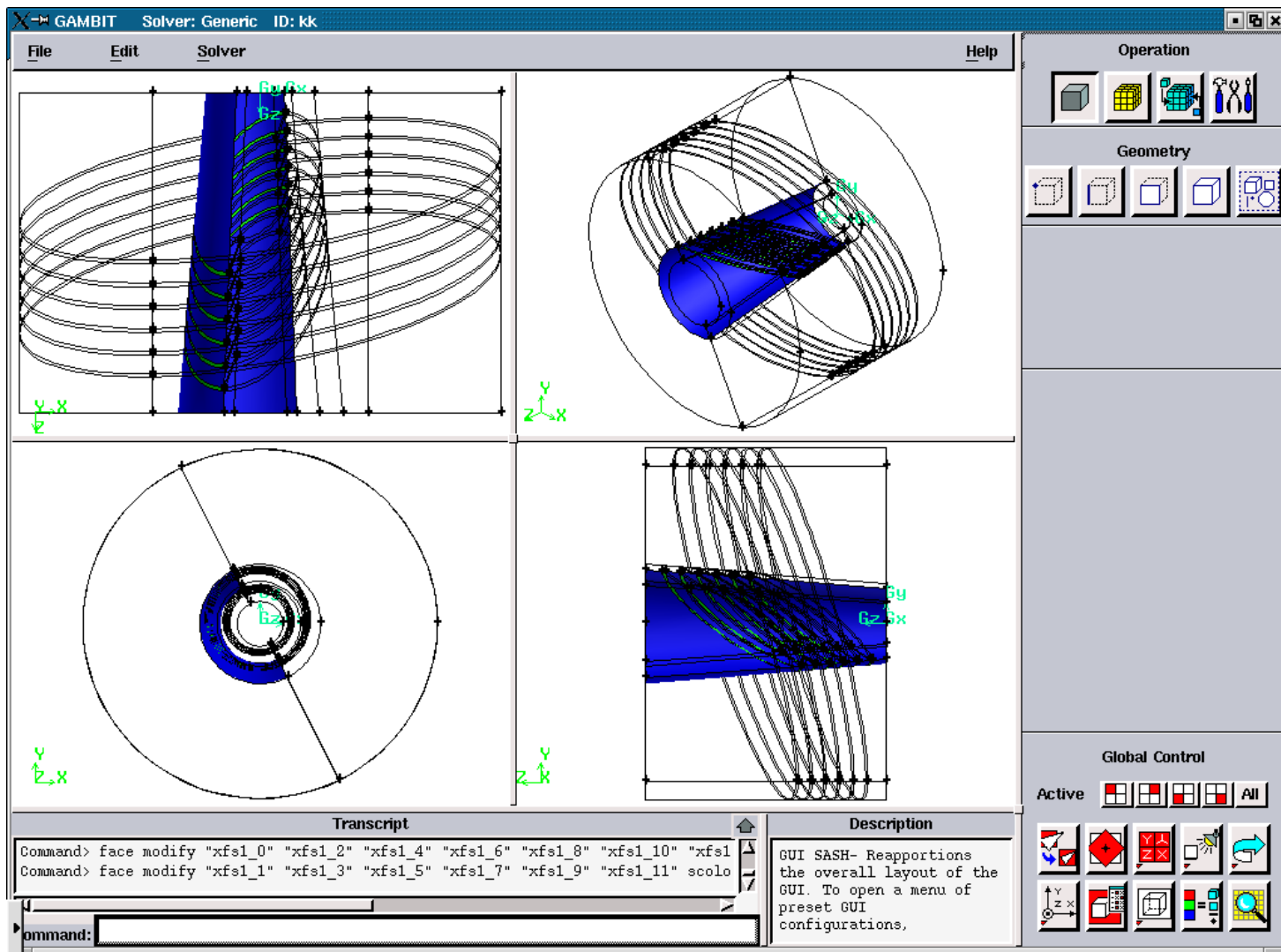


Figure 3.5.62. Axial Solid Faces for the “left”.

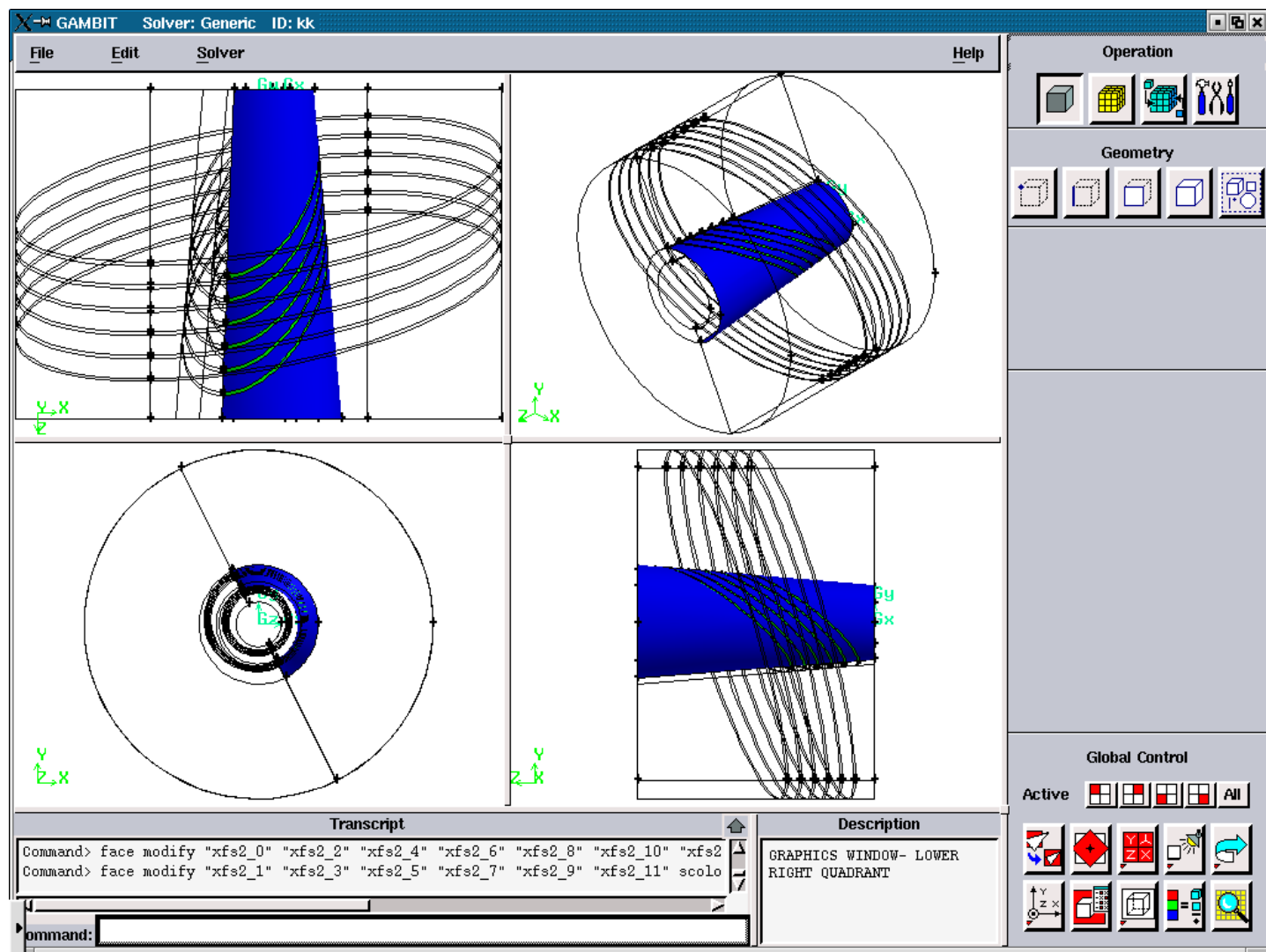


Figure 3.5.63. Axial Solid Faces for the “right”.

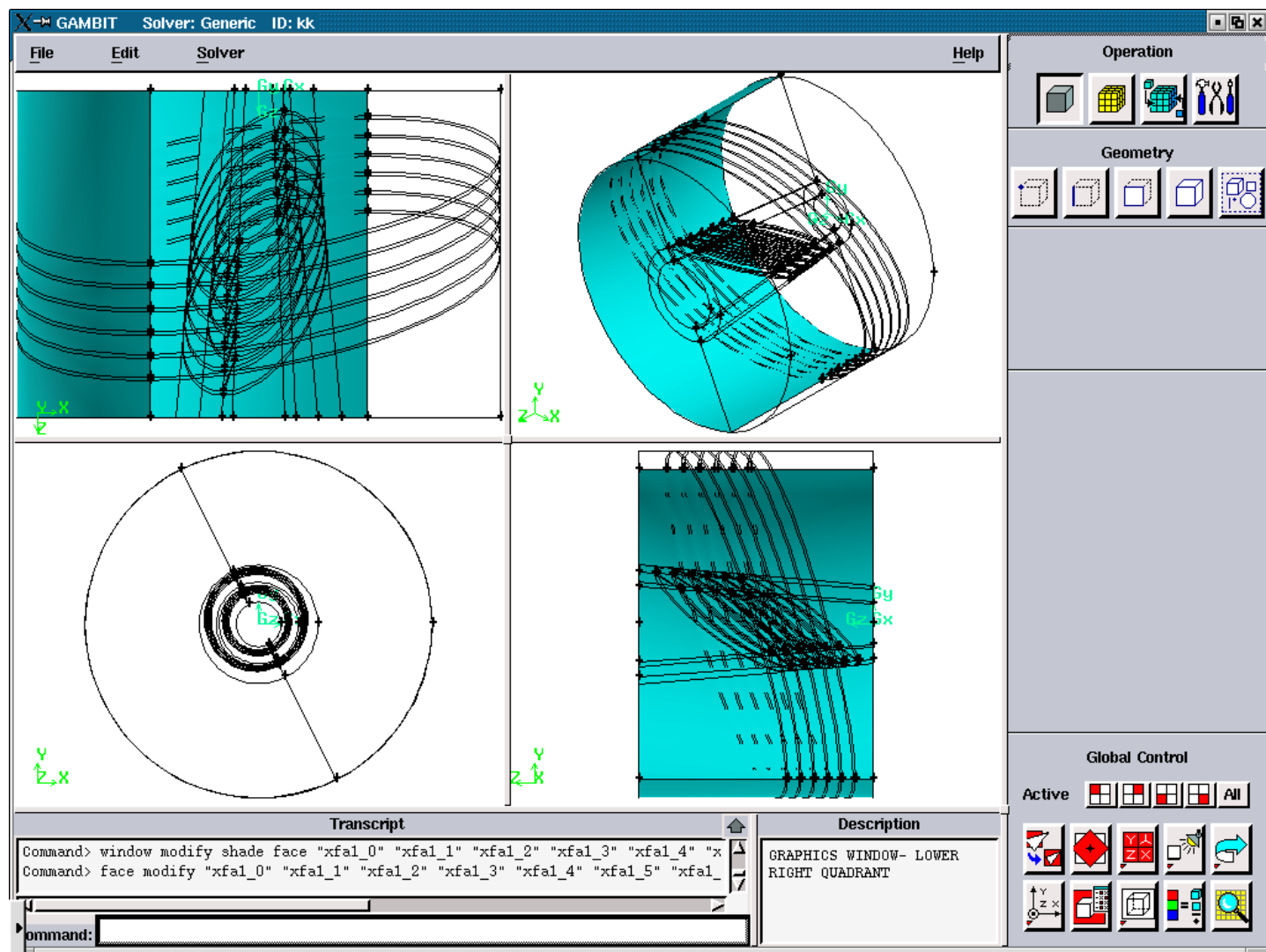


Figure 3.5.64. Axial Air Faces for the “left”.

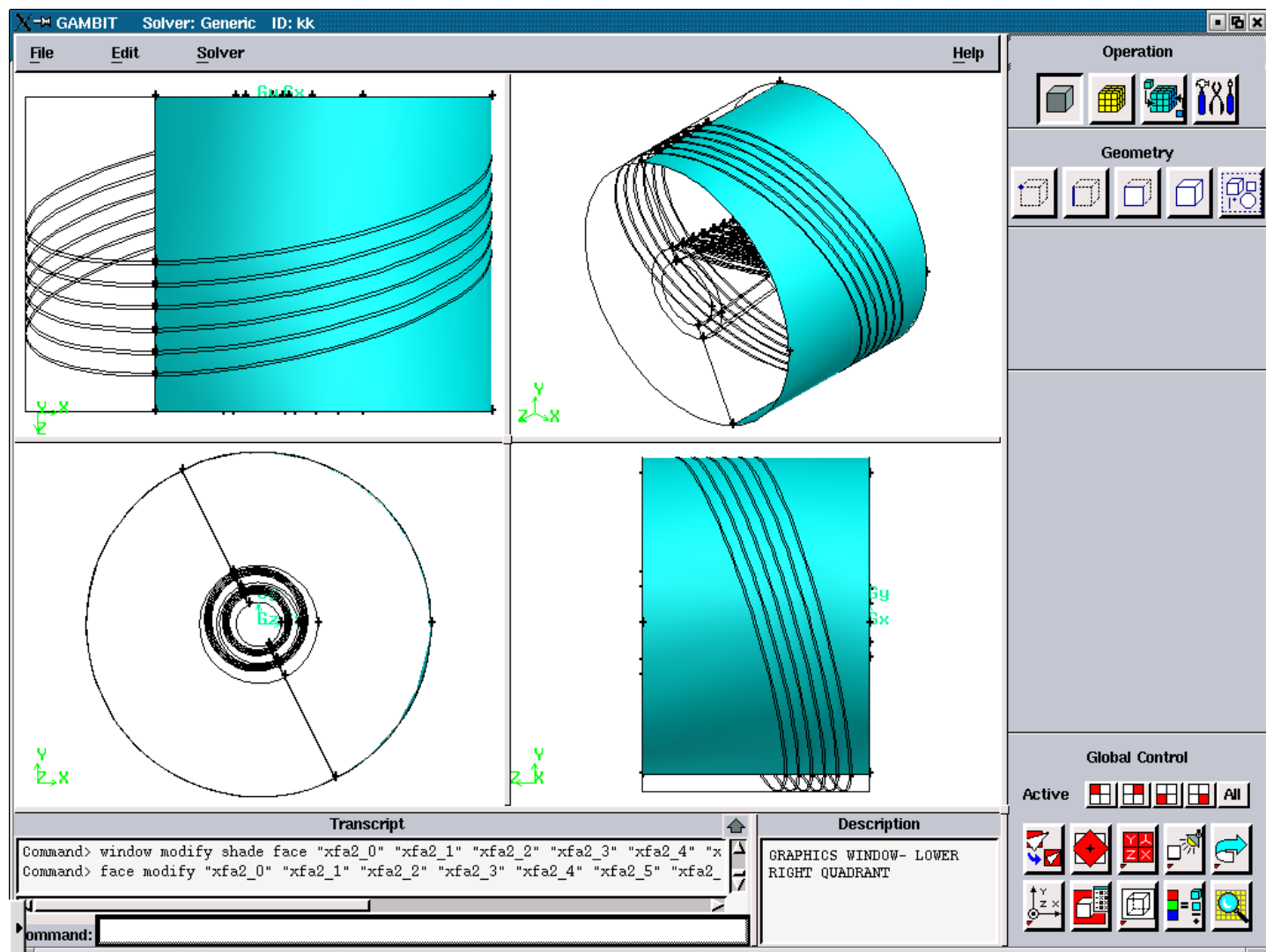


Figure 3.5.65. Axial Air Faces for the “right”.

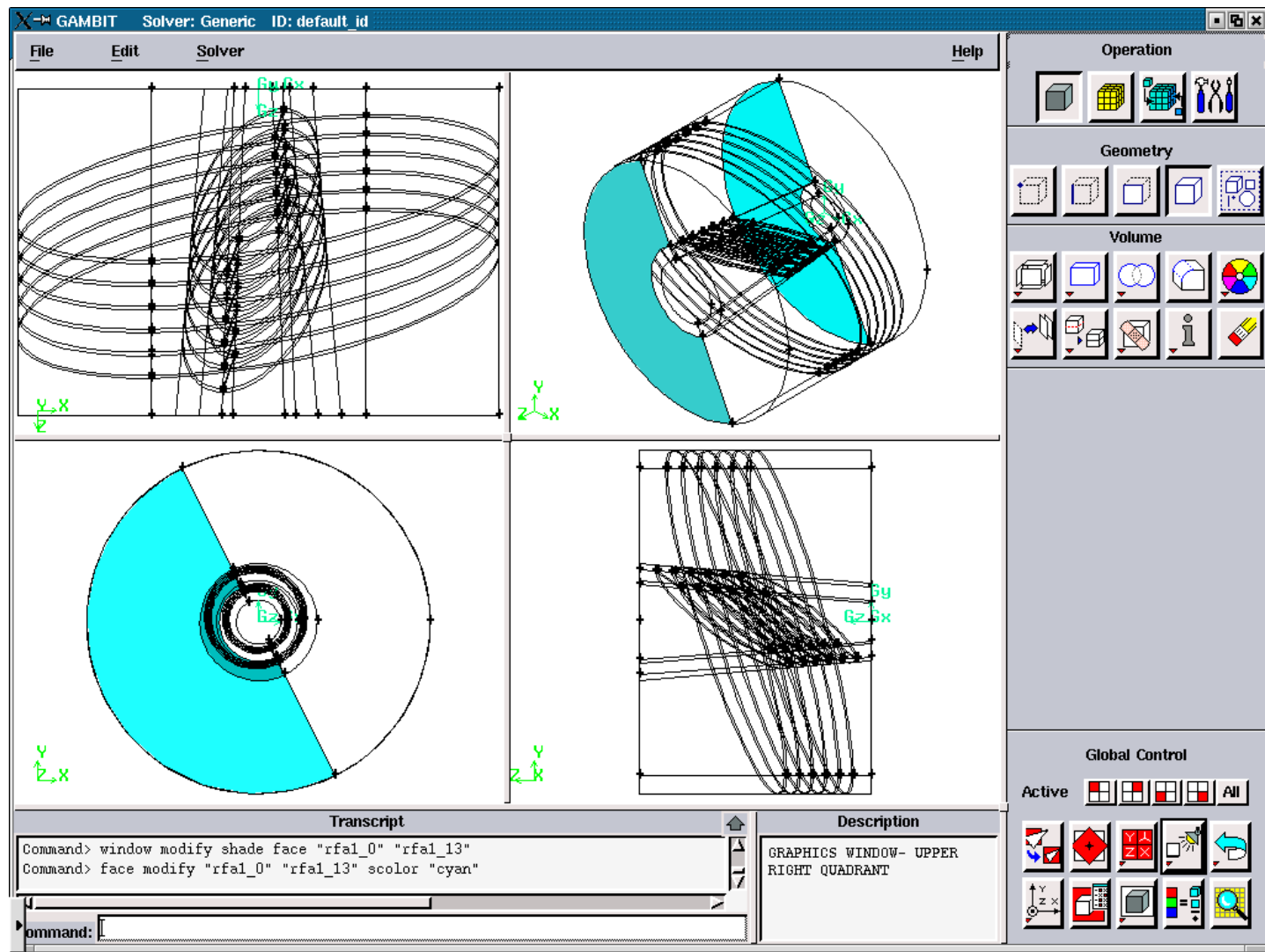


Figure 3.5.66. Radial Air Faces for the “left”.

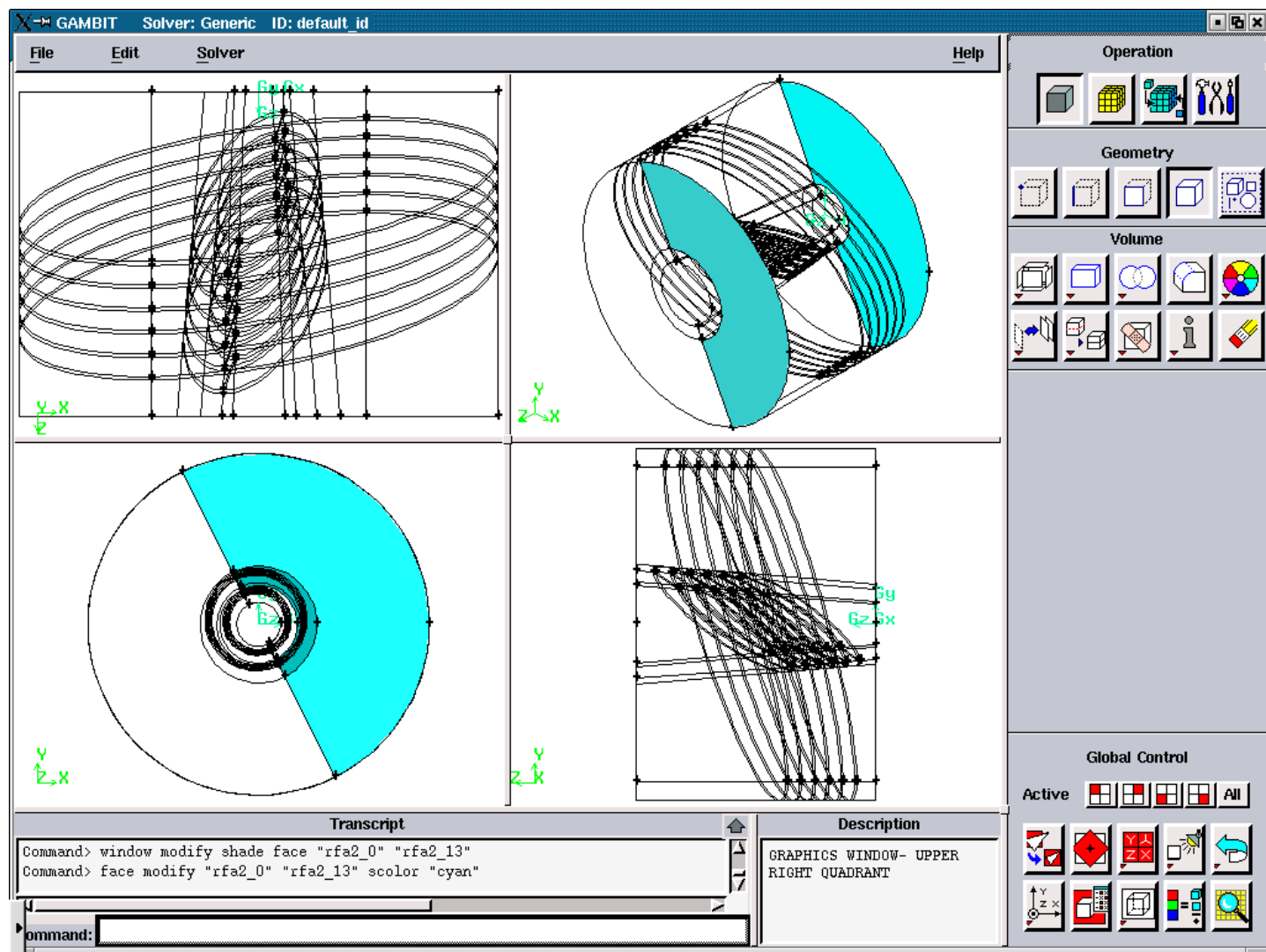


Figure 3.5.67. Radial Air Faces for the “right”.

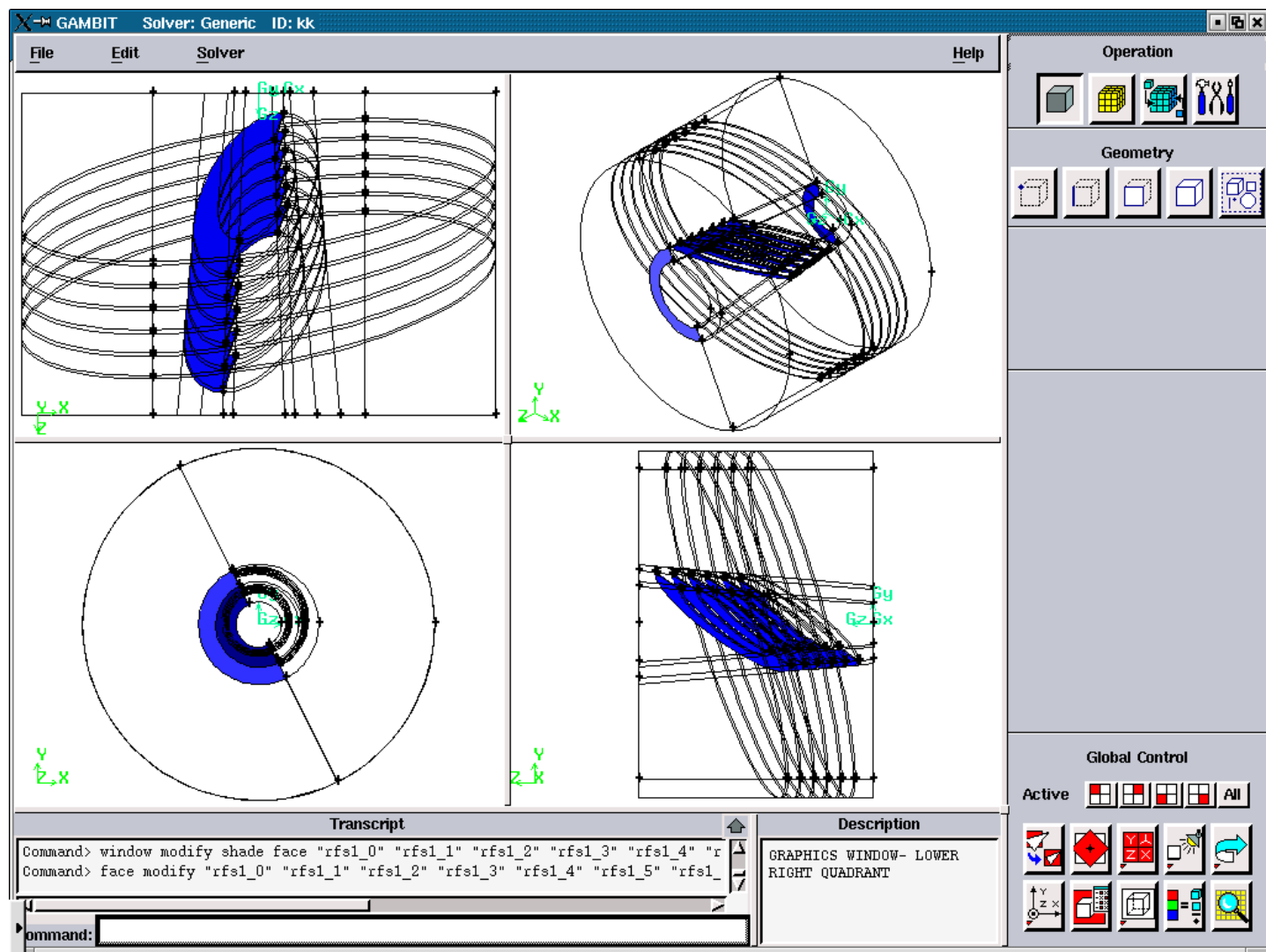


Figure 3.5.68. Radial Solid Faces for the “left”.

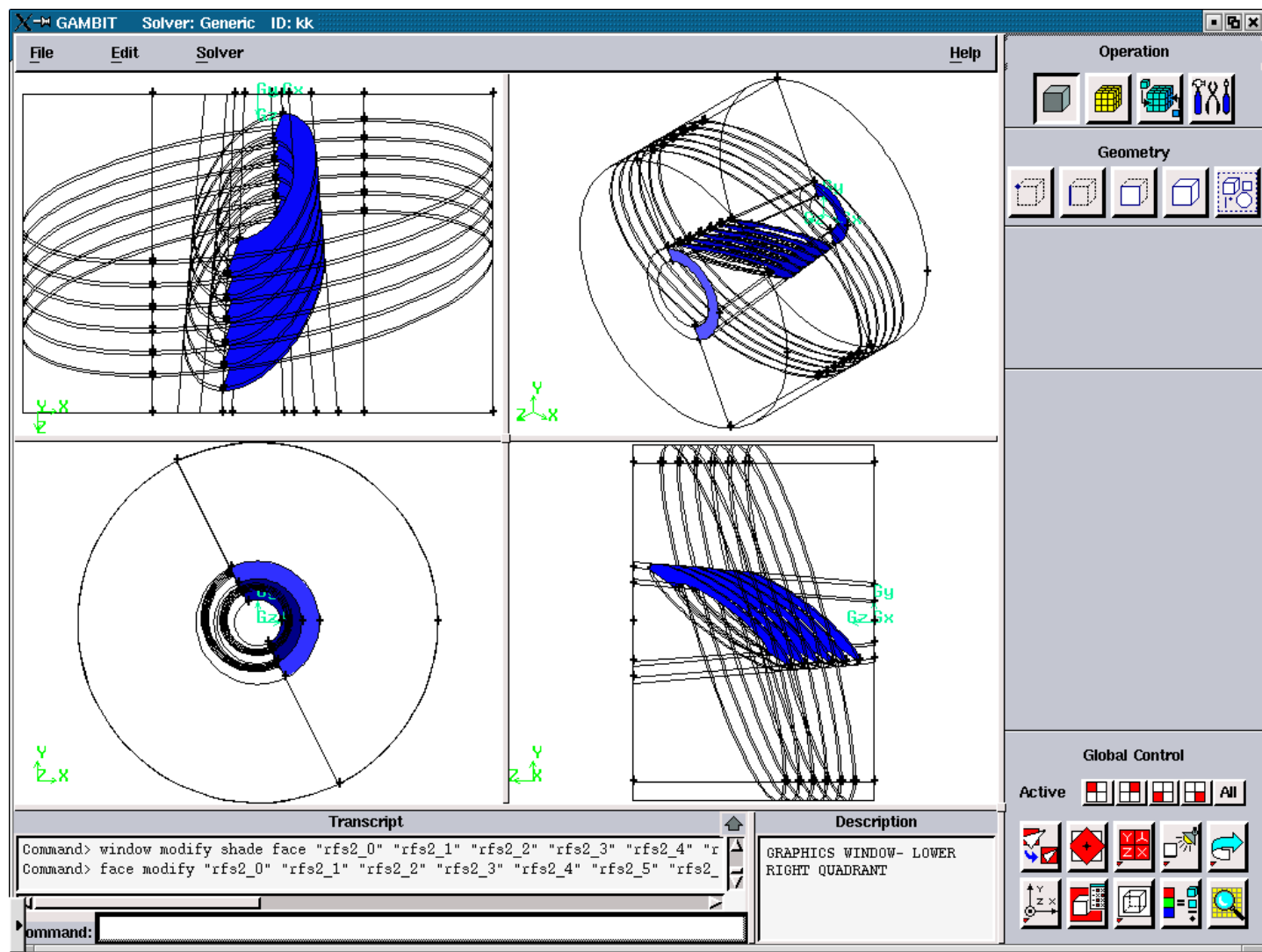
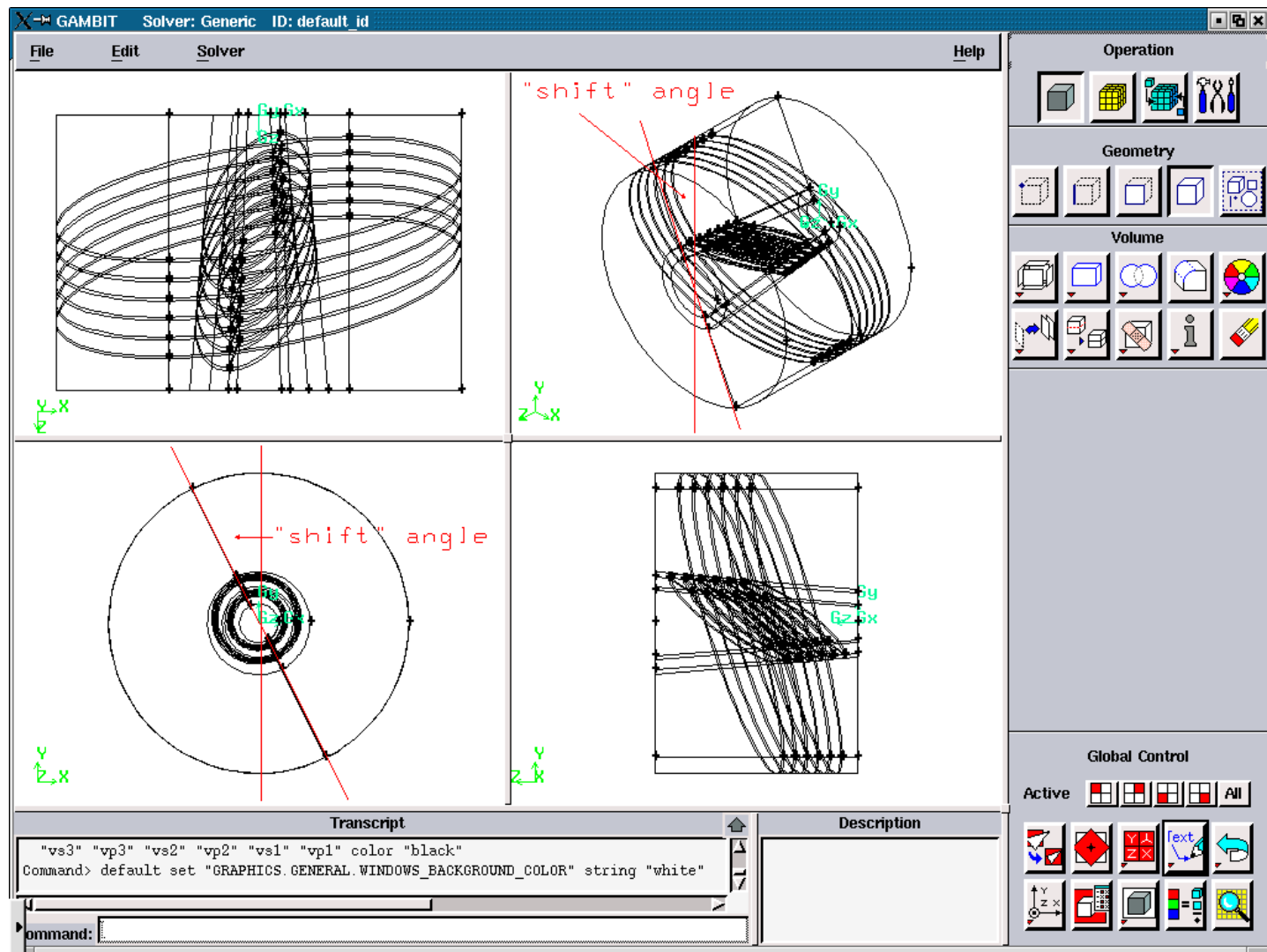


Figure 3.5.69. Radial Solid Faces for the “right”.



**Figure 3.5.70.** The “shift” angle for the channel with two slope angles for the electric elements for the cut plane.

## **A1.5. FIGURES ILLUSTRATING THE CREATION OF THE FILE**

**‘modify3\_d’ and ‘modify3\_s’**

### **A1.5.2. FIGURES FOR THE CREATION OF THE FILE ‘modify3\_d’**

**FOR THE SLOPE ANGLES: 45, 90**

---

All figures in this section represent the geometry after the consequent run of the files:

‘temp1.jou’, ‘modify1’, temp2.jou’, modify2, ‘temp3.jou’

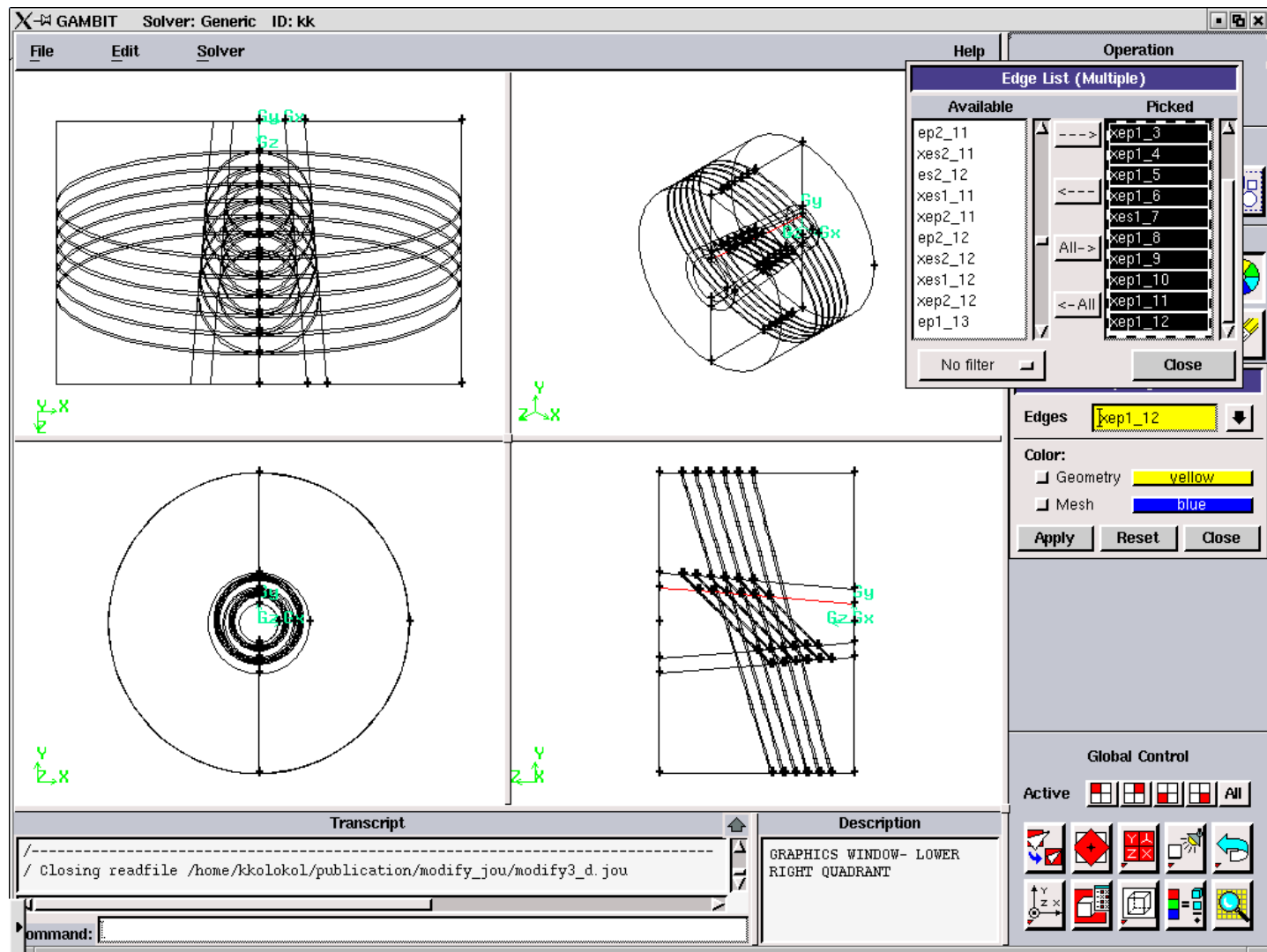


Figure 3.5.71. Axial Plasma Edges for the “maximum”.

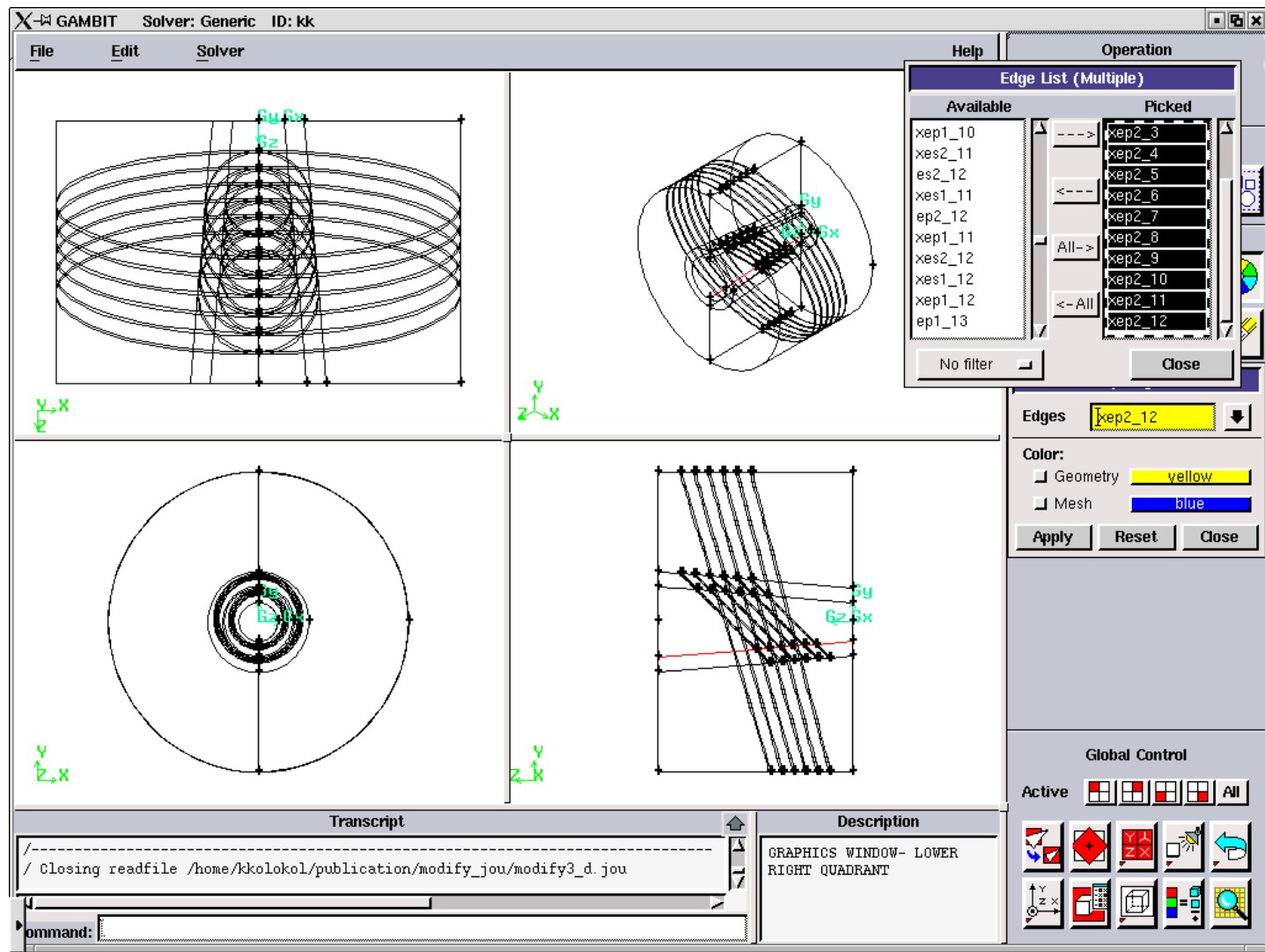


Figure 3.5.72. Axial Plasma Edges for the “minimum”.

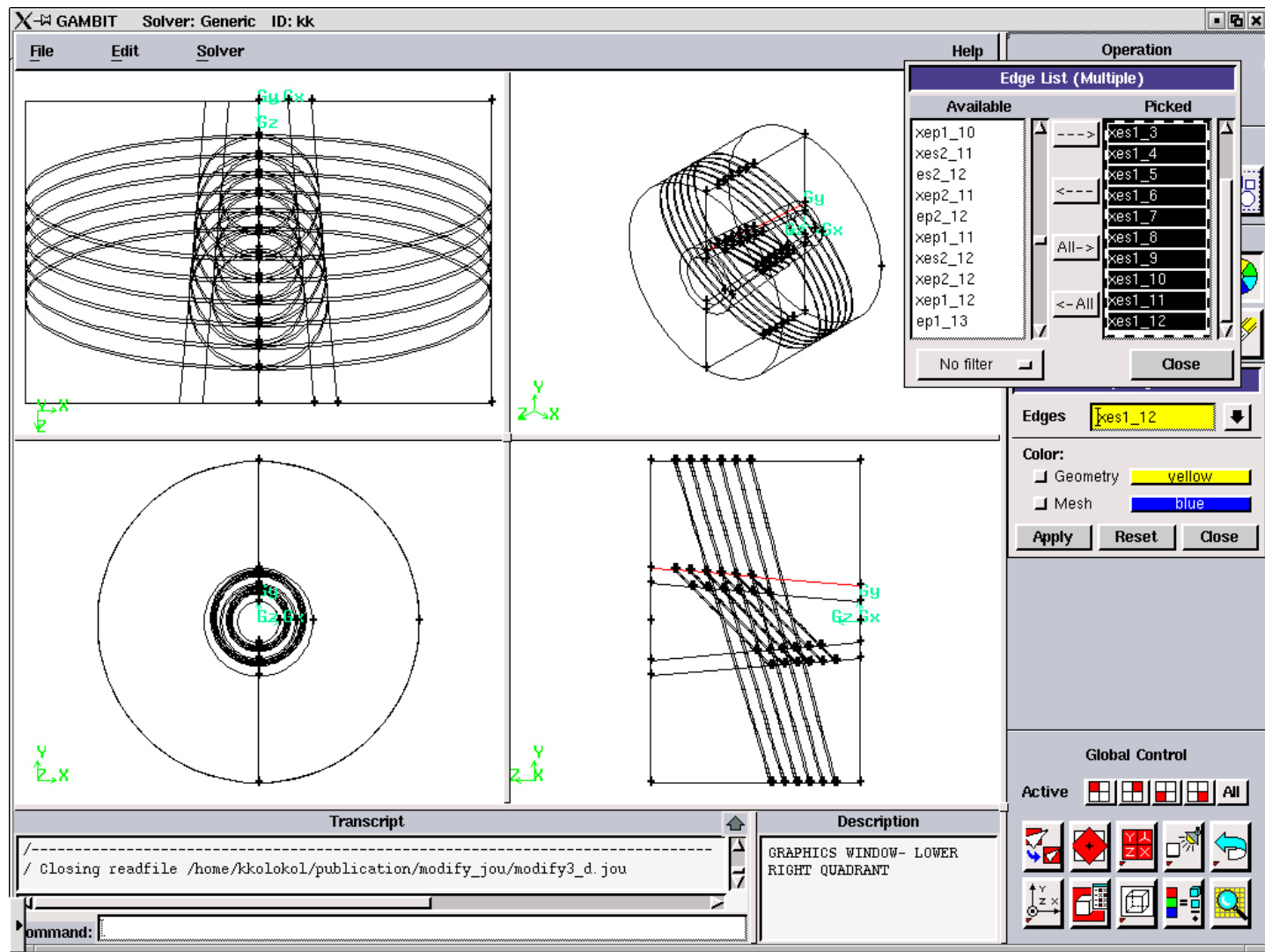


Figure 3.5.73. Axial Solid Edges for the “maximum”.



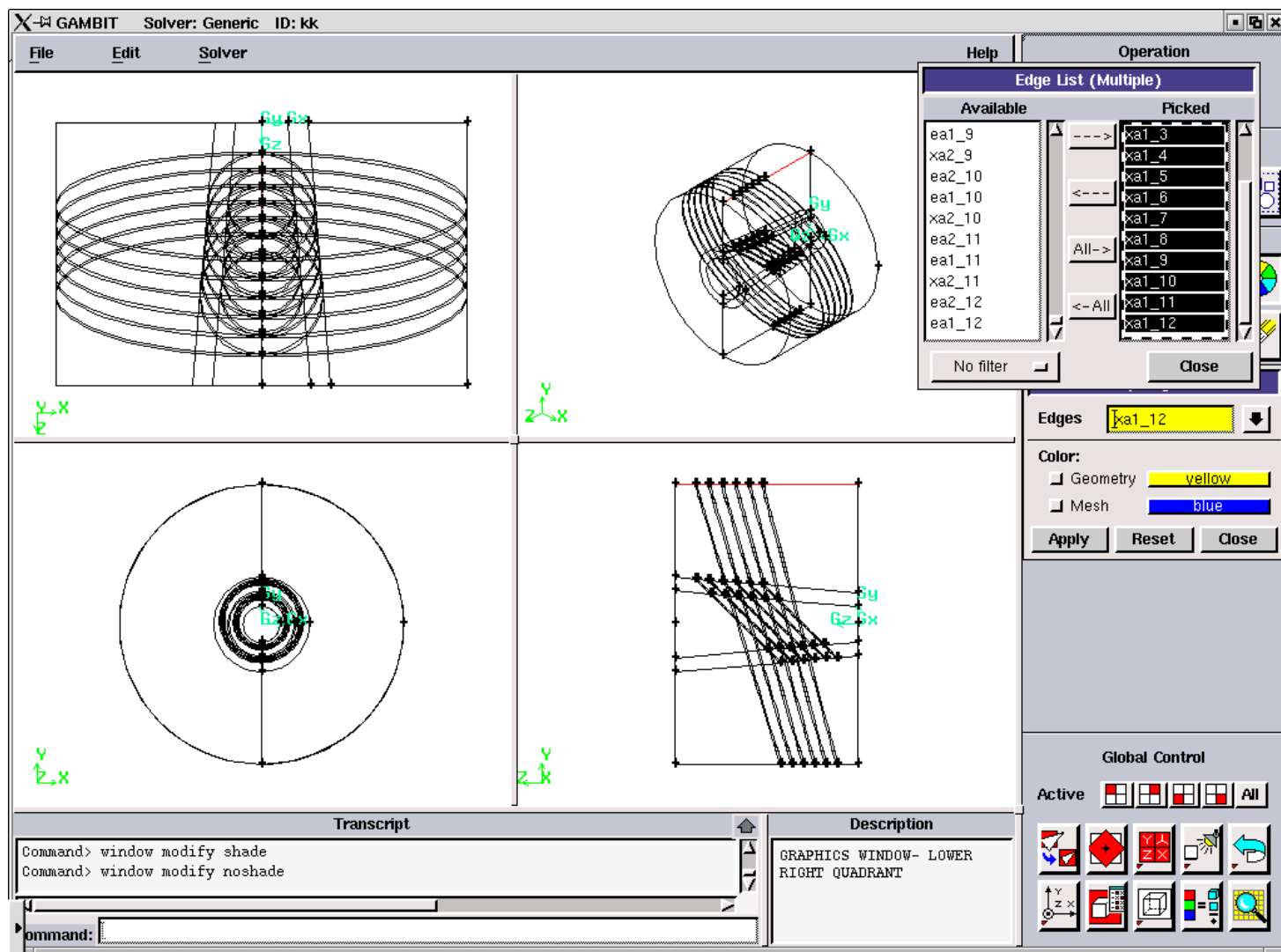


Figure 3.5.75. Axial Air Edges for the “maximum”.

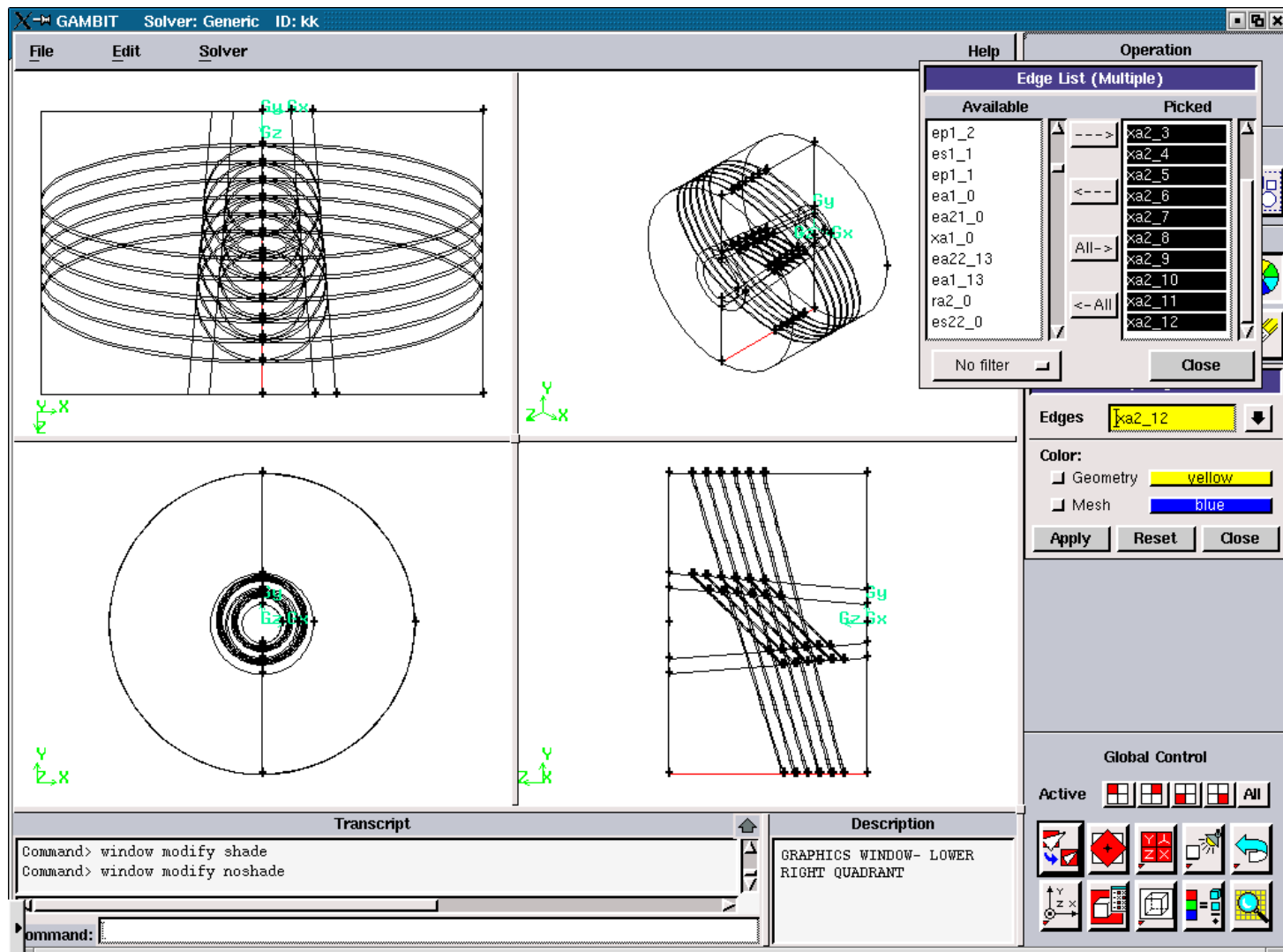
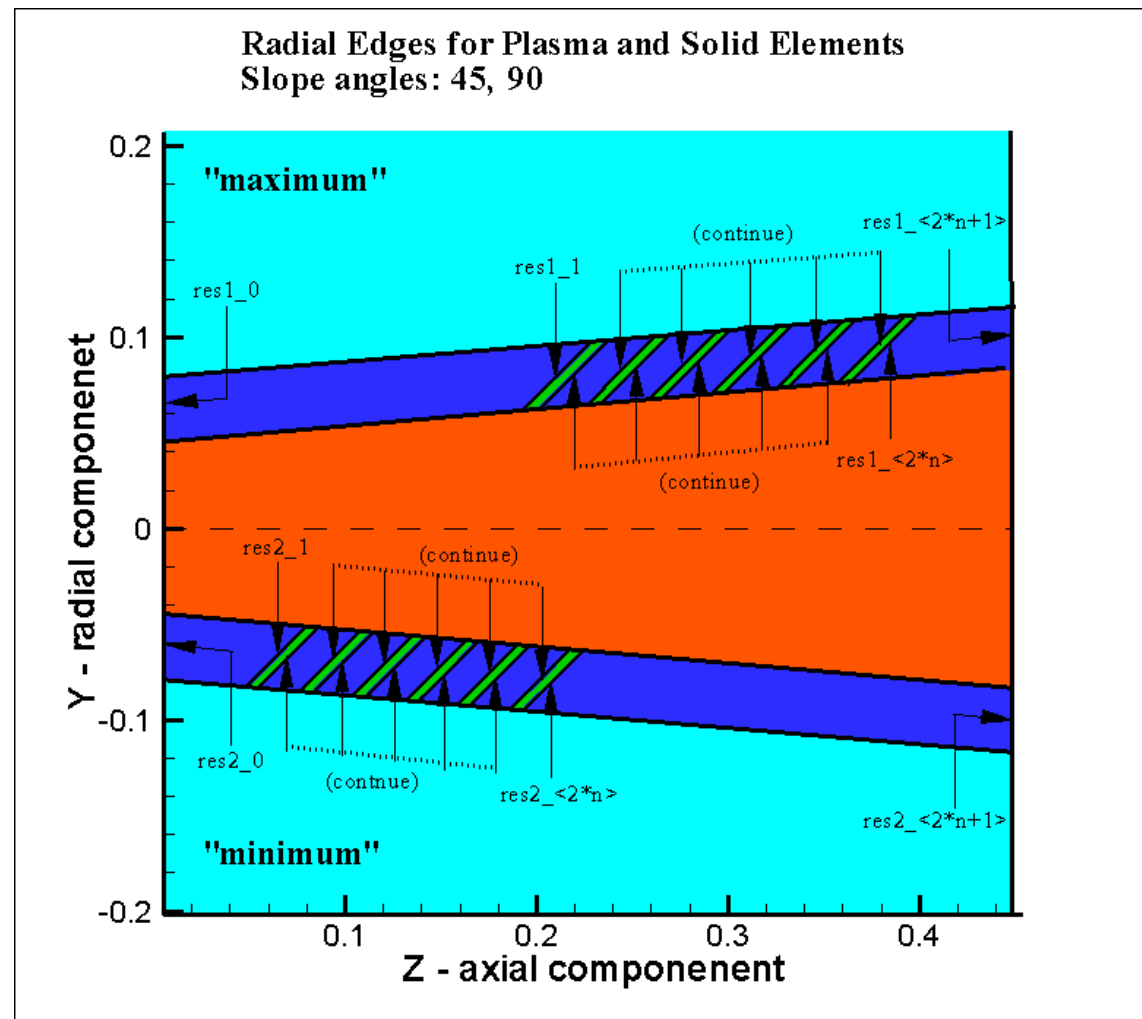


Figure 3.5.76. Axial Air Edges for the “minimum”.



**Figure 3.5.77.** Radial Solid Edges for the “maximum” and “minimum”.

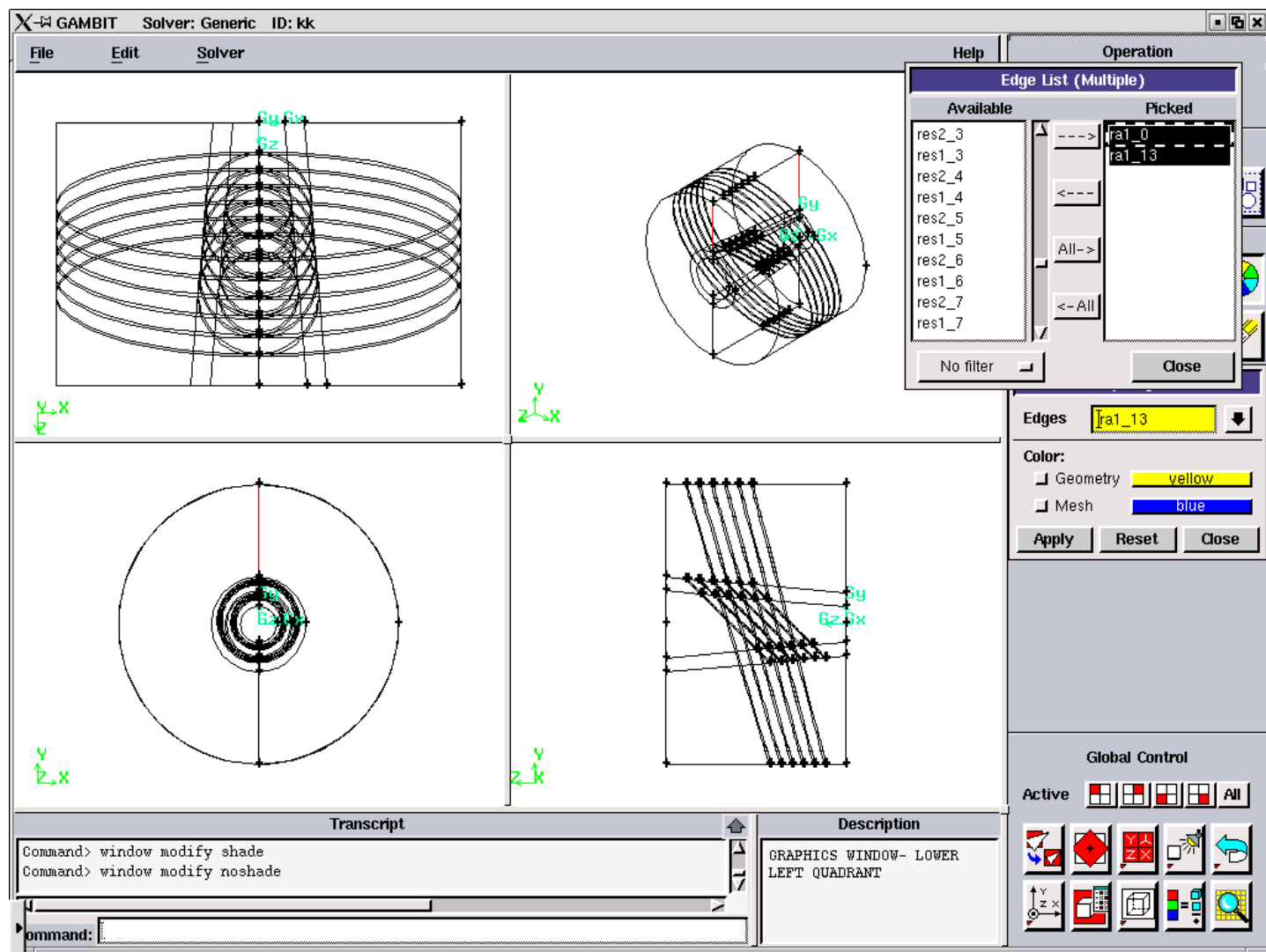


Figure 3.5.78. Radial Air Edges for the “maximum”.

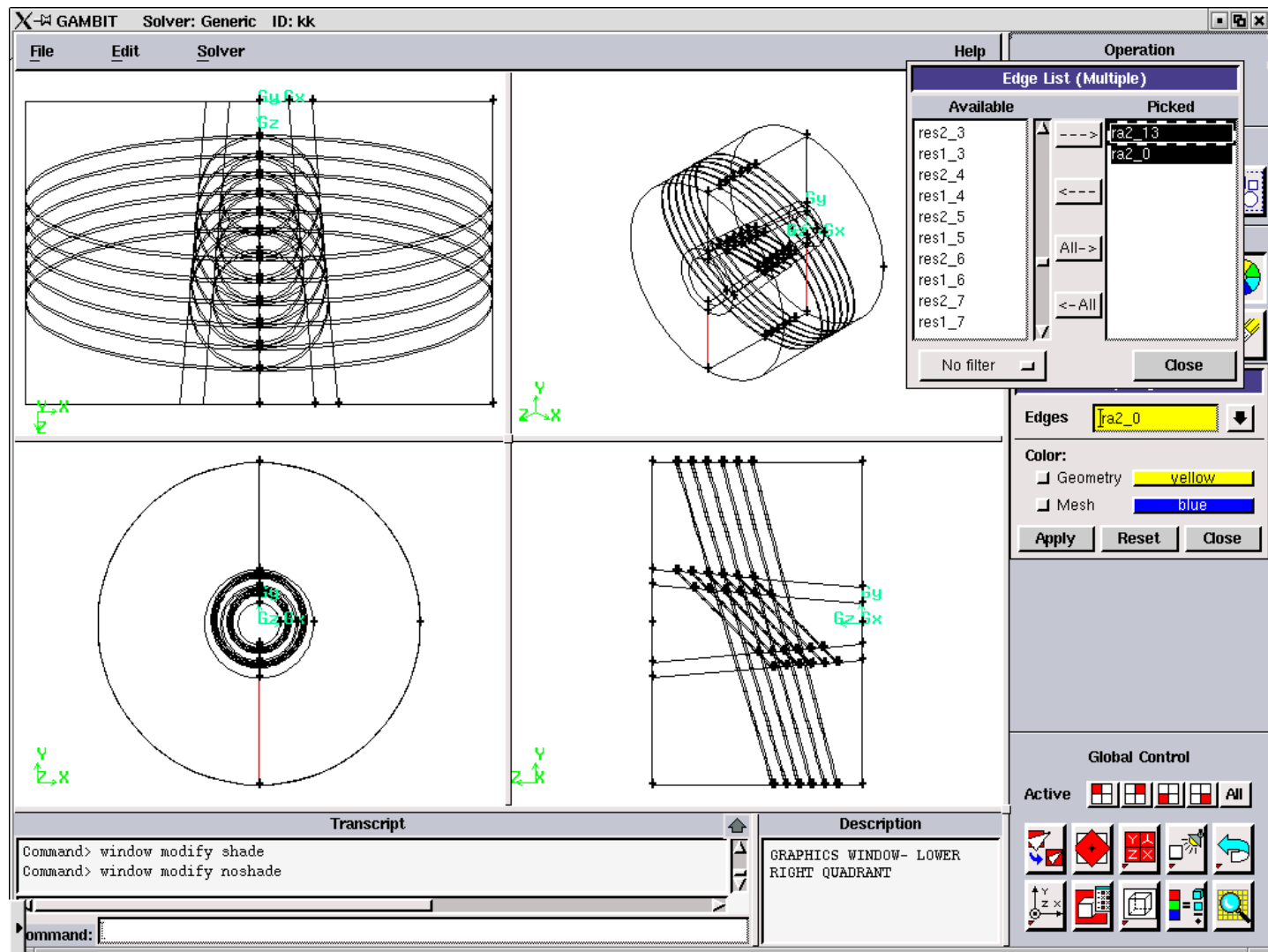


Figure 3.5.79. Radial Air Edges for the “minimum”.

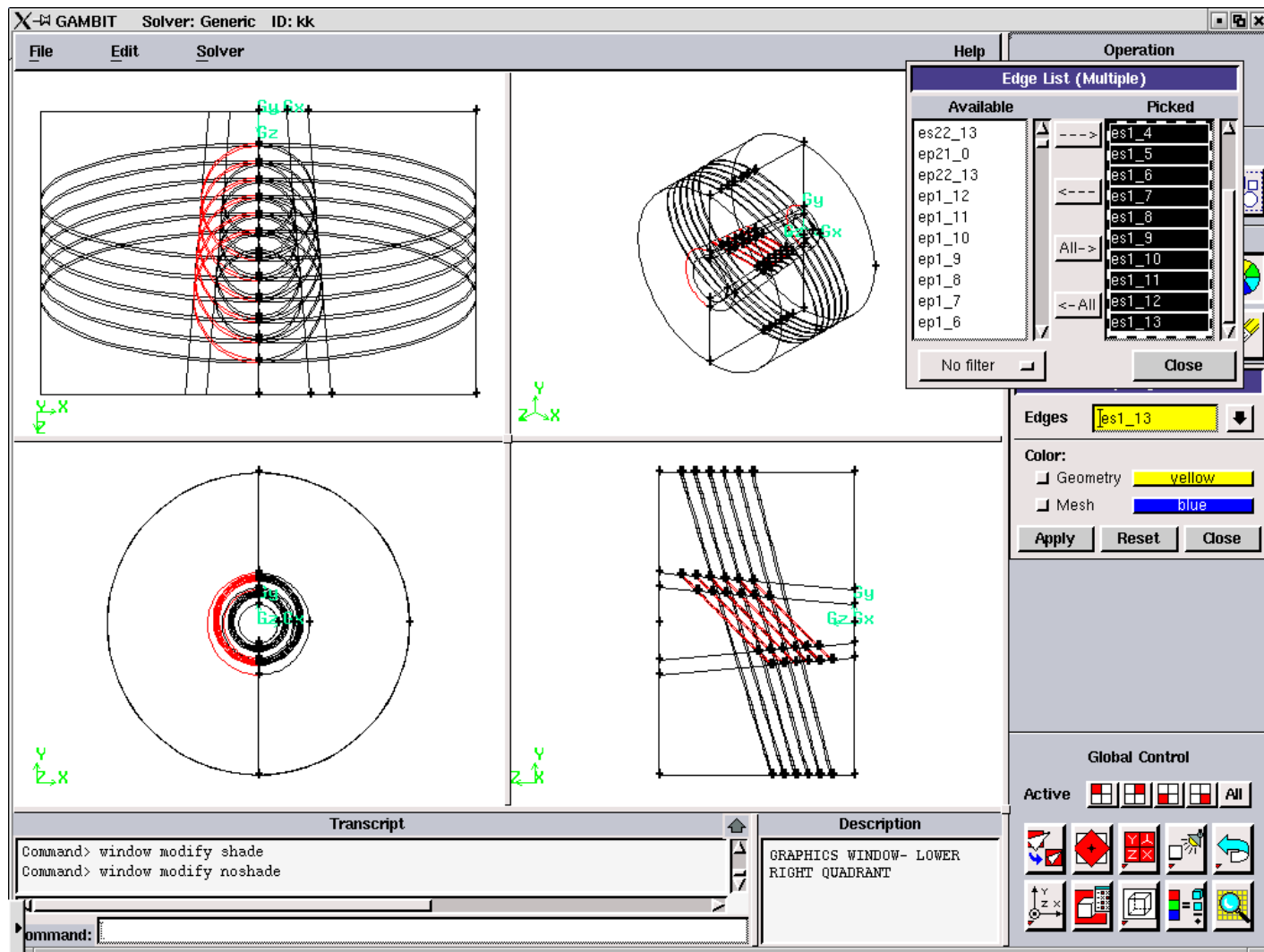


Figure 3.5.80. Elliptic Solid Edges for the “left”.

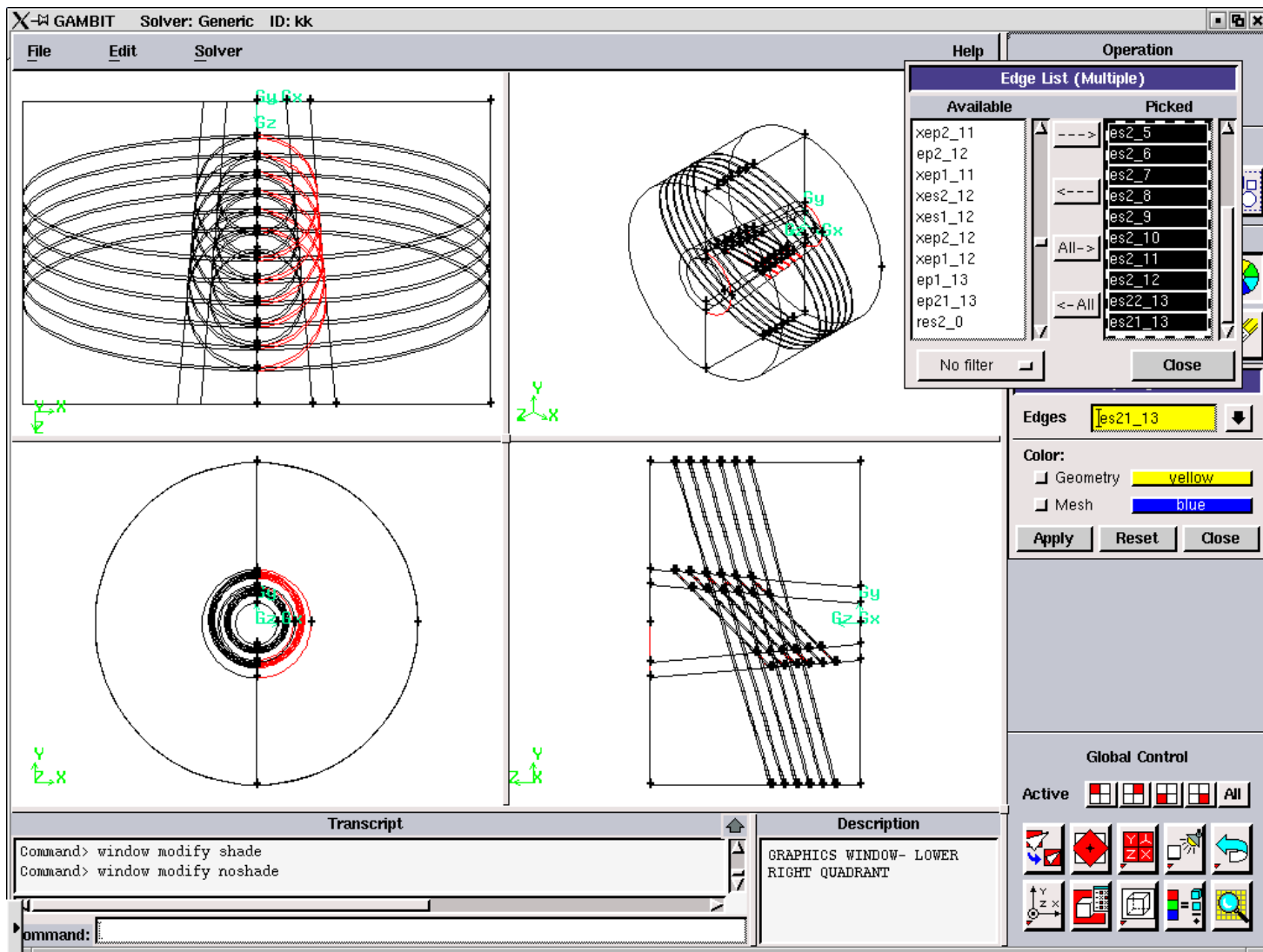


Figure 3.5.81. Elliptic Solid Edges for the “right”.

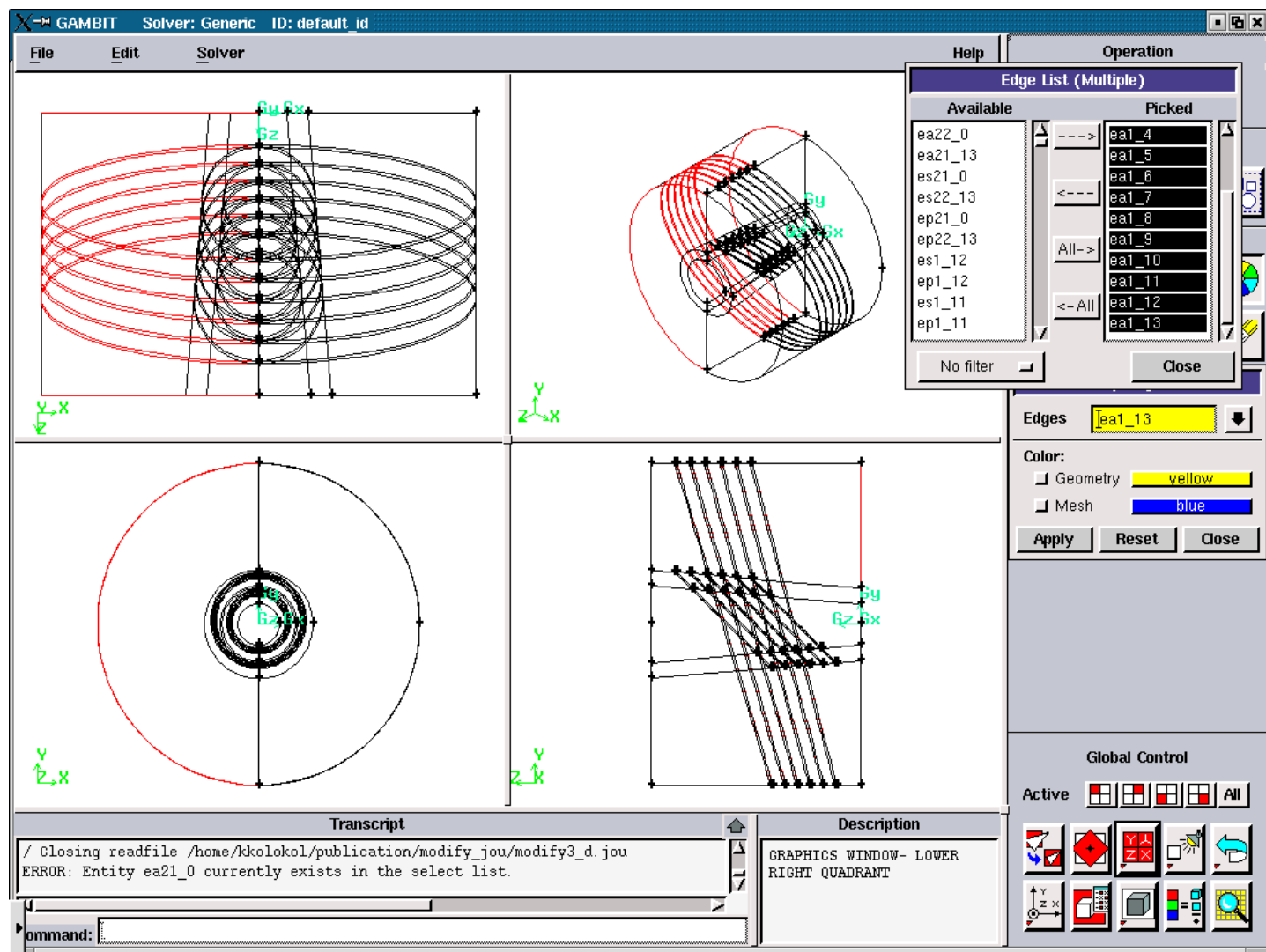


Figure 3.5.82. Elliptic Air Edges for the “left”.

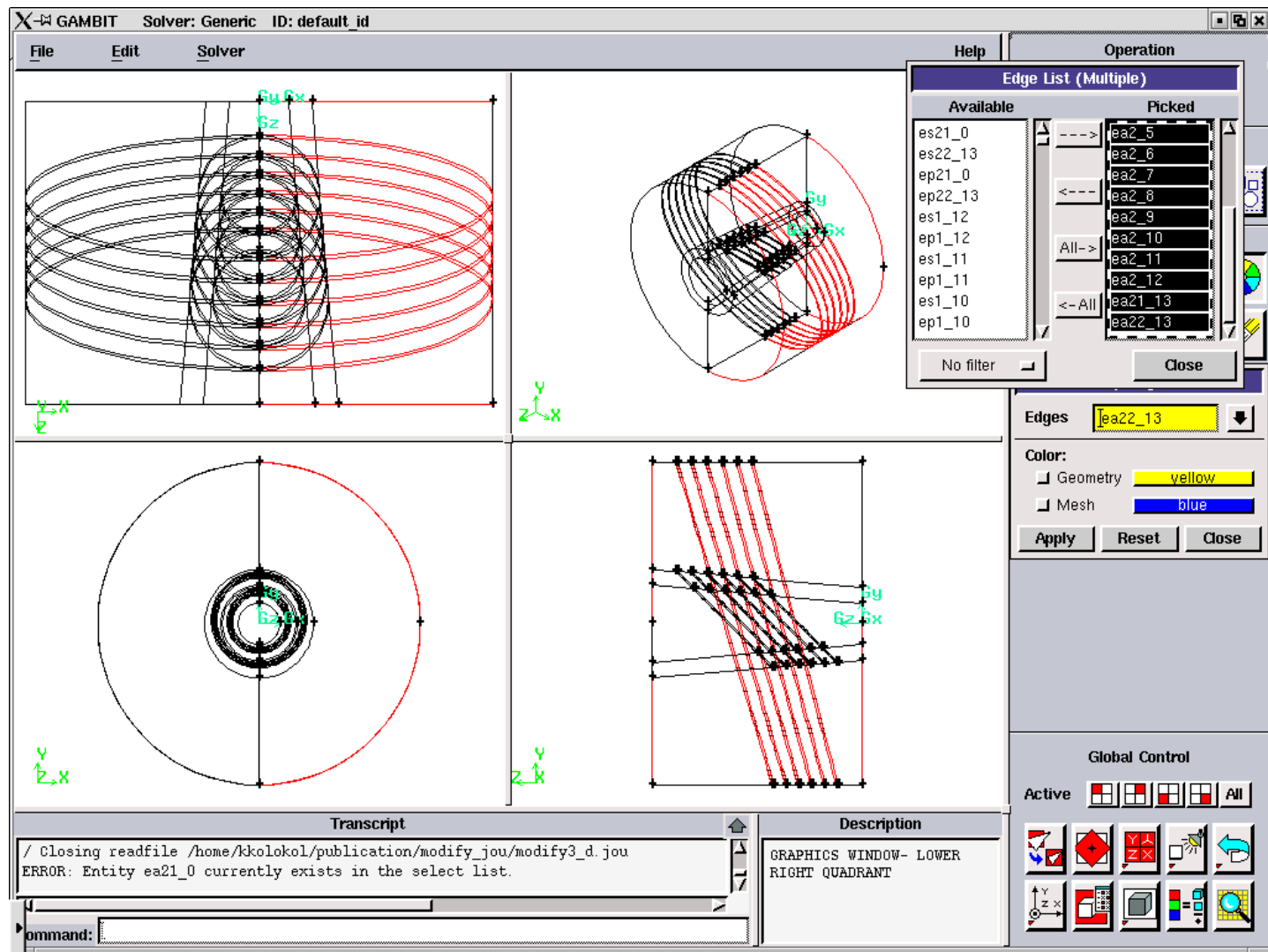


Figure 3.5.83. Elliptic Air Edges for the “right”.

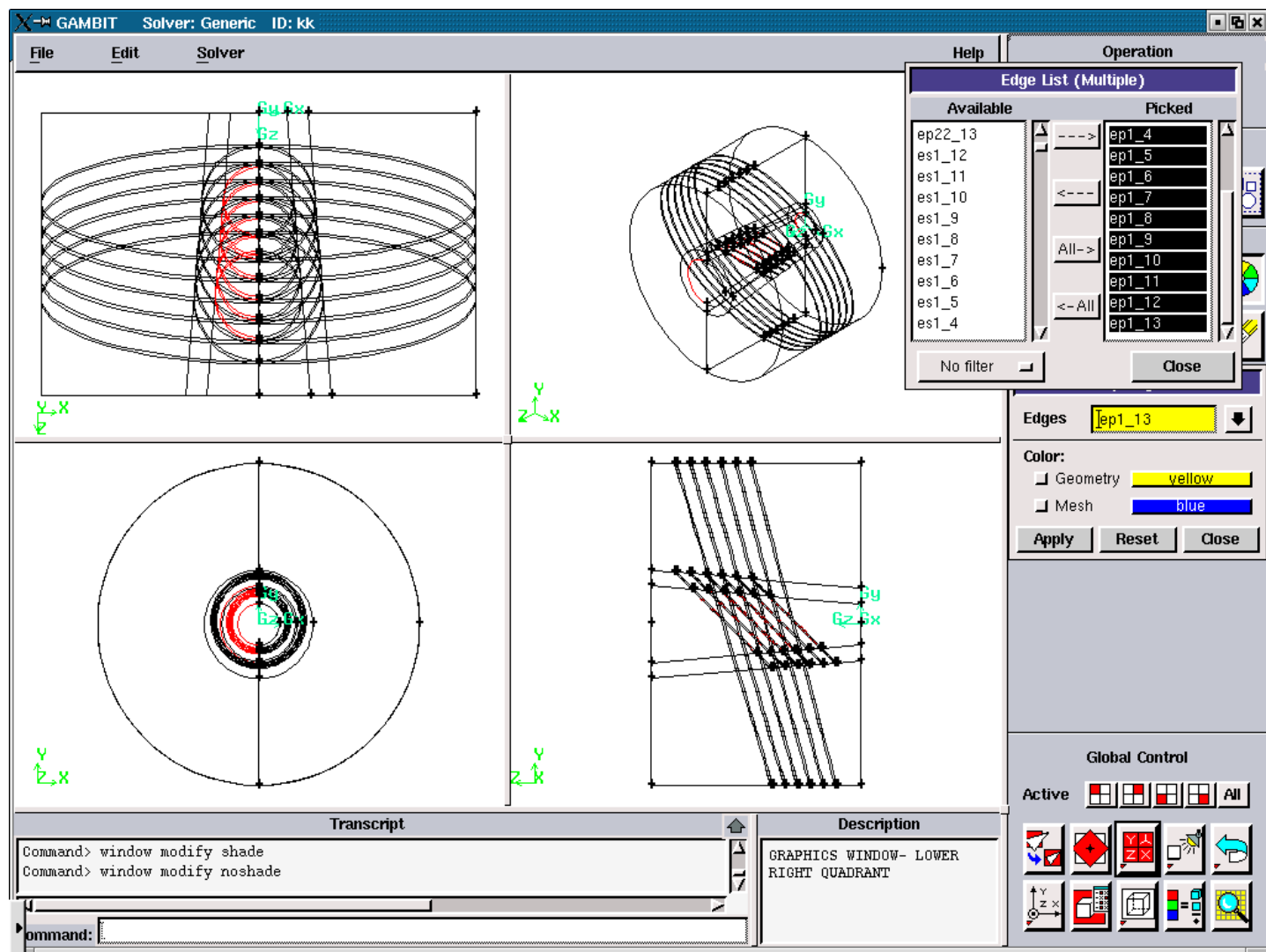


Figure 3.5.84. Elliptic Plasma Edges for the “left”.

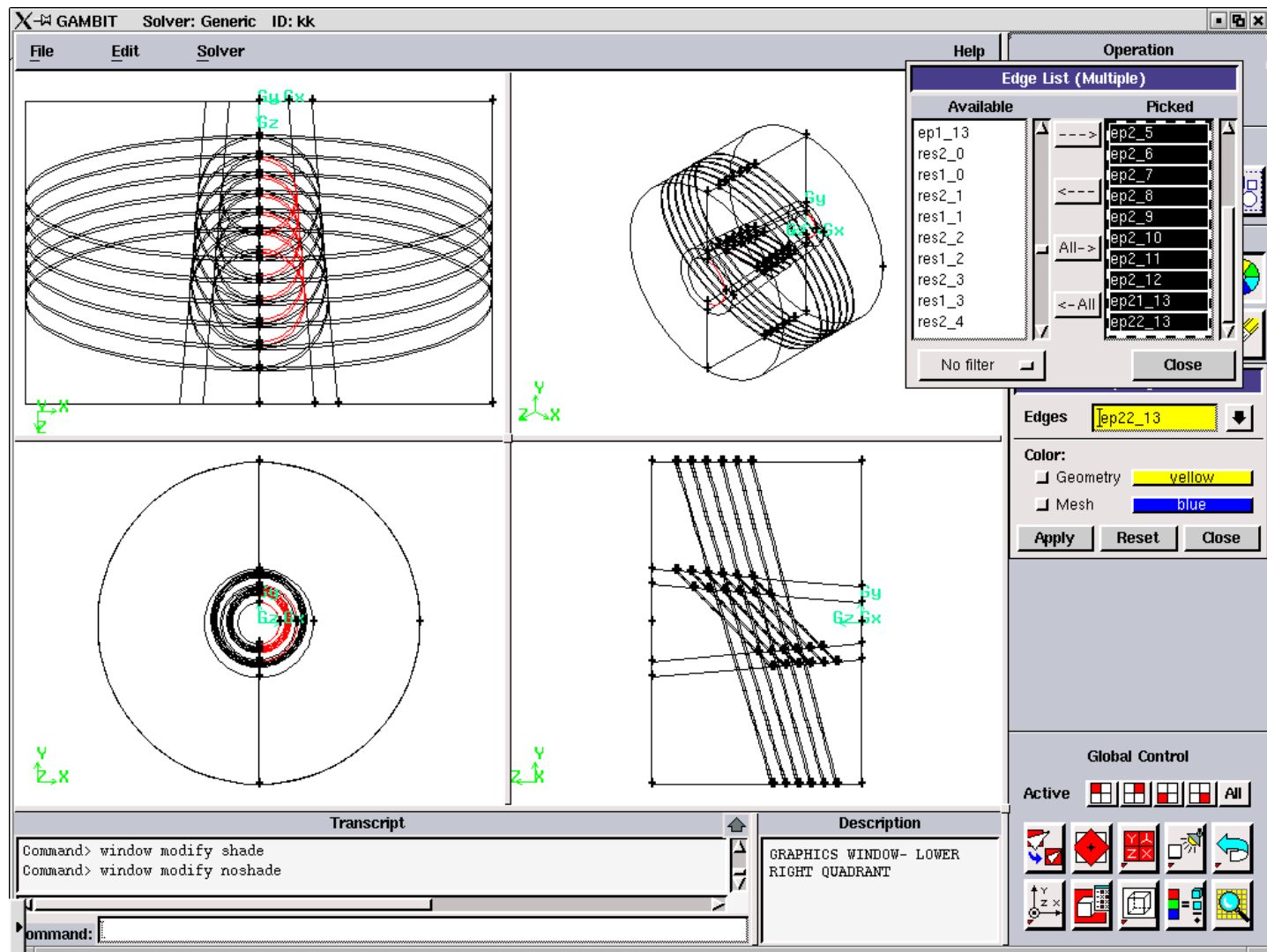


Figure 3.5.85. Elliptic Plasma Edges for the “right”.

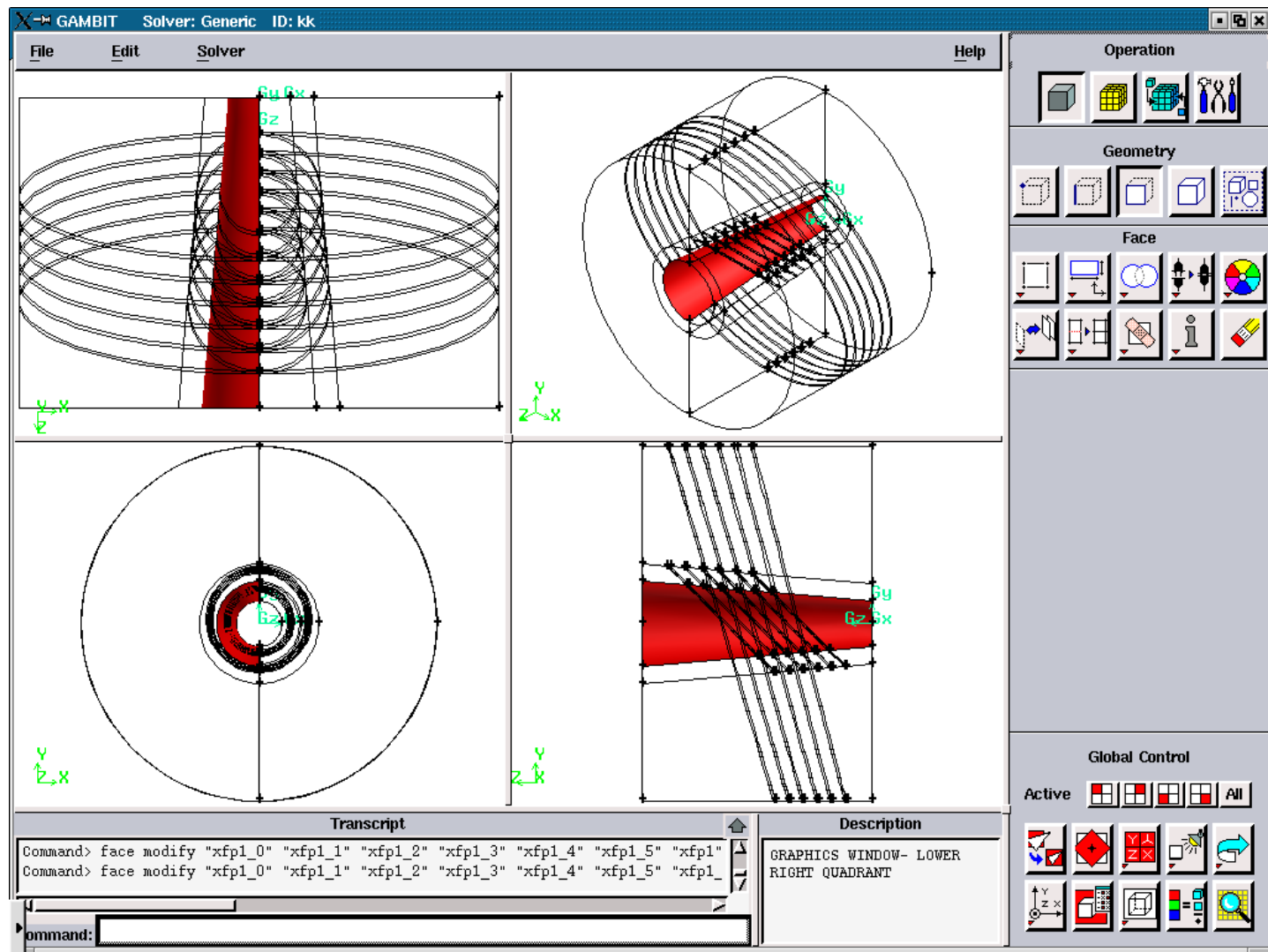


Figure 3.5.86. Axial Plasma Faces for the “left”.

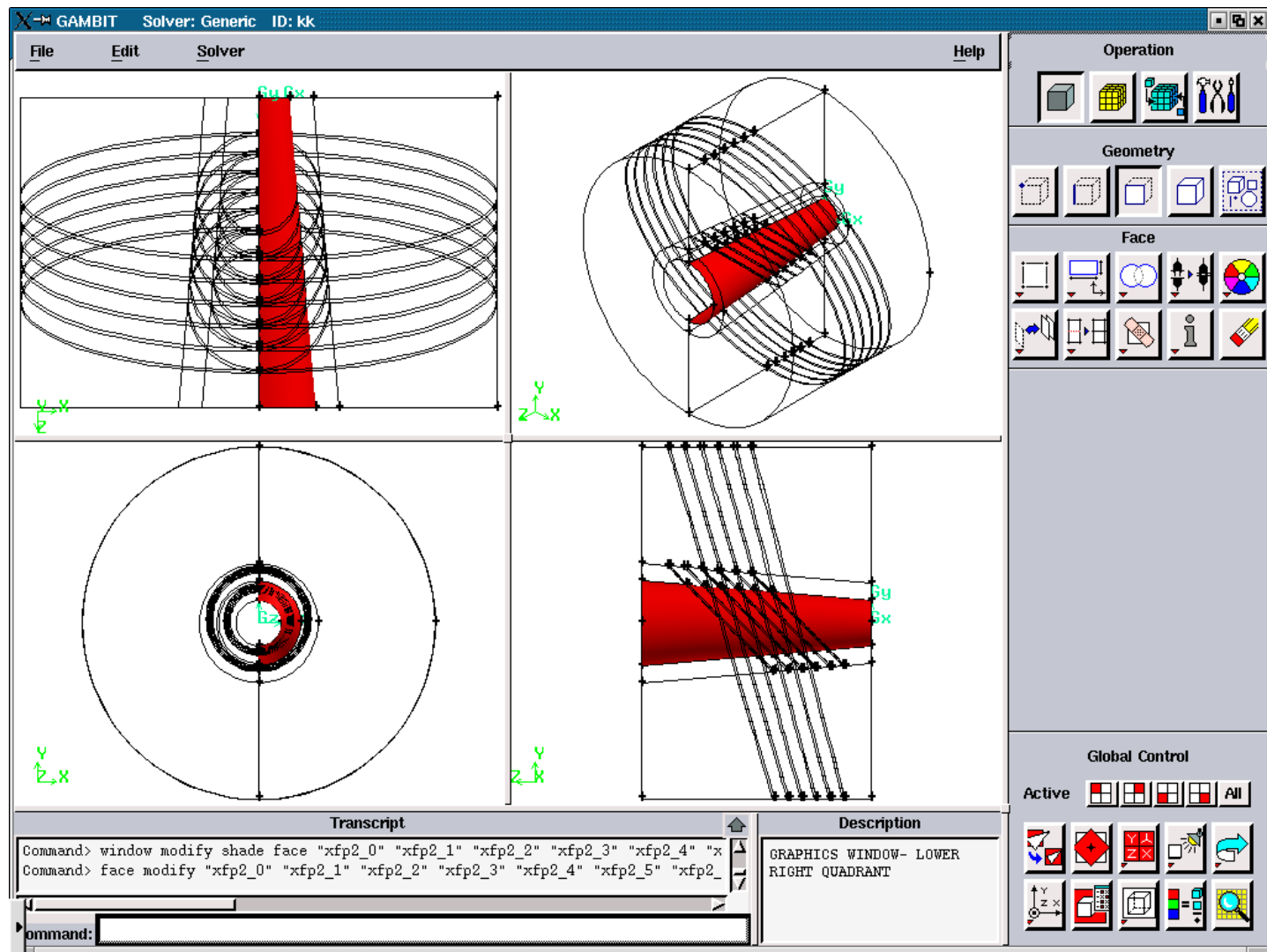


Figure 3.5.87. Axial Plasma Faces for the “right”.

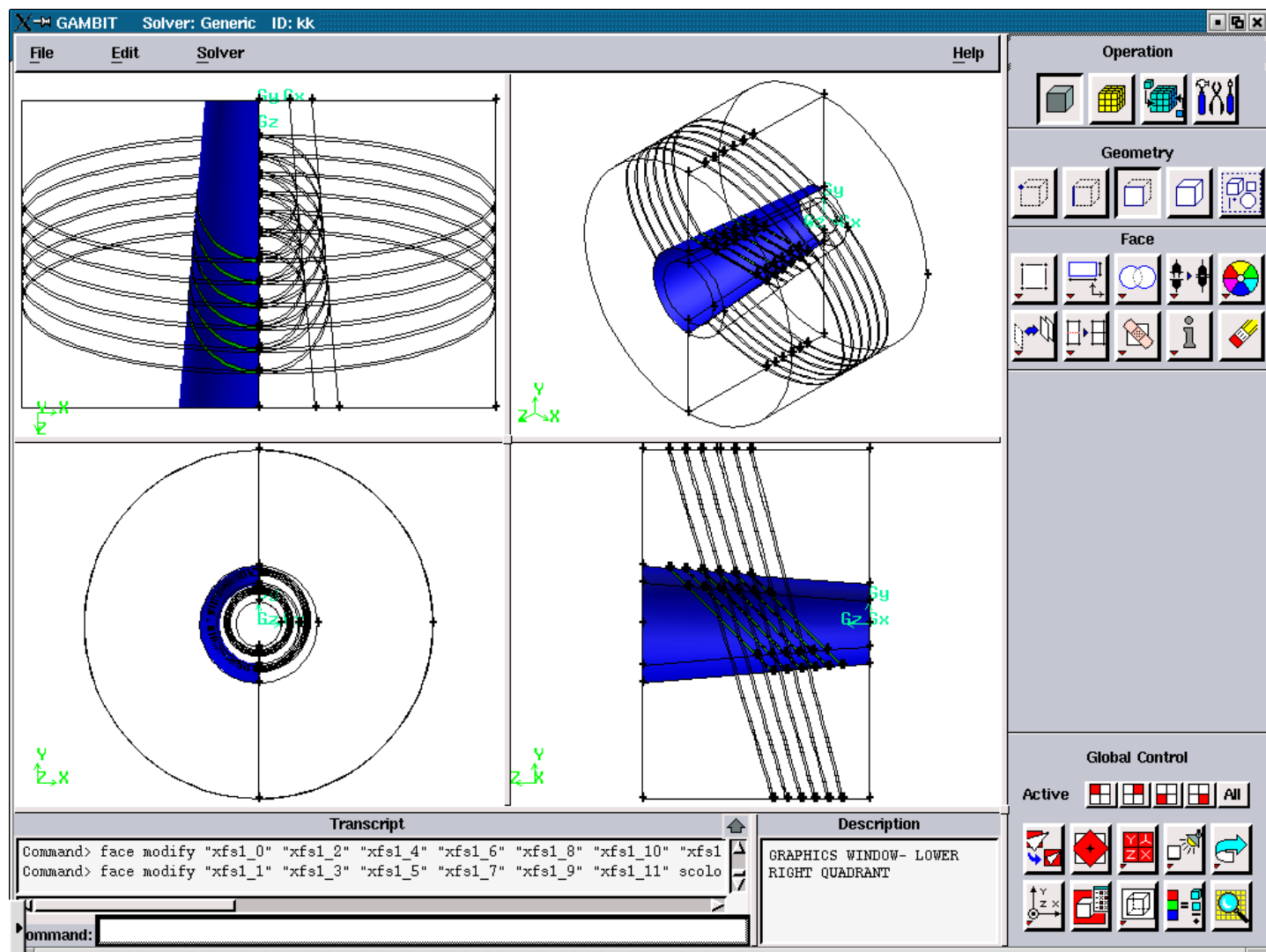


Figure 3.5.88. Axial Solid Faces for the “left”.

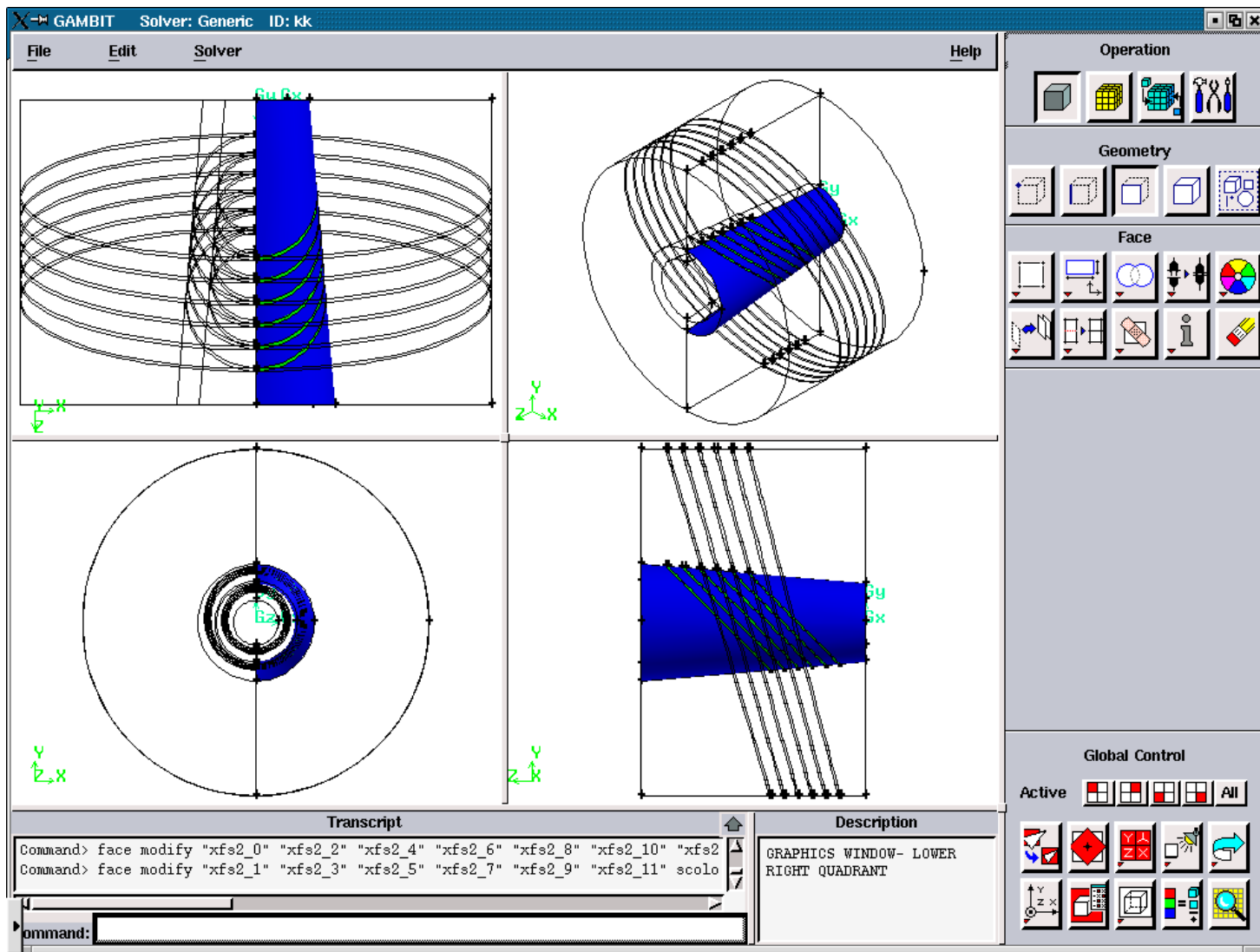


Figure 3.5.89. Axial Solid Faces for the “right”.

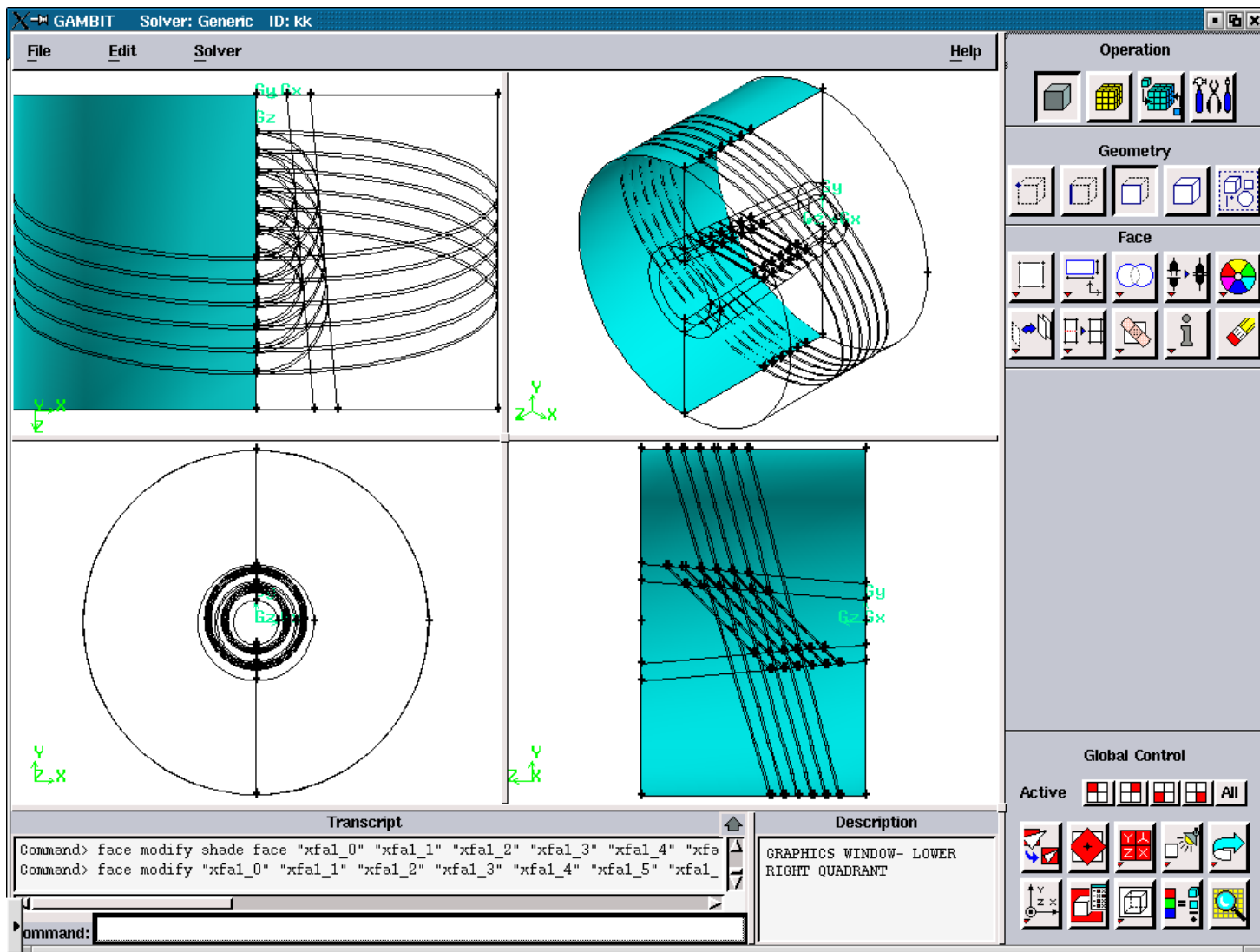


Figure 3.5.90. Axial Air Faces for the “left”.

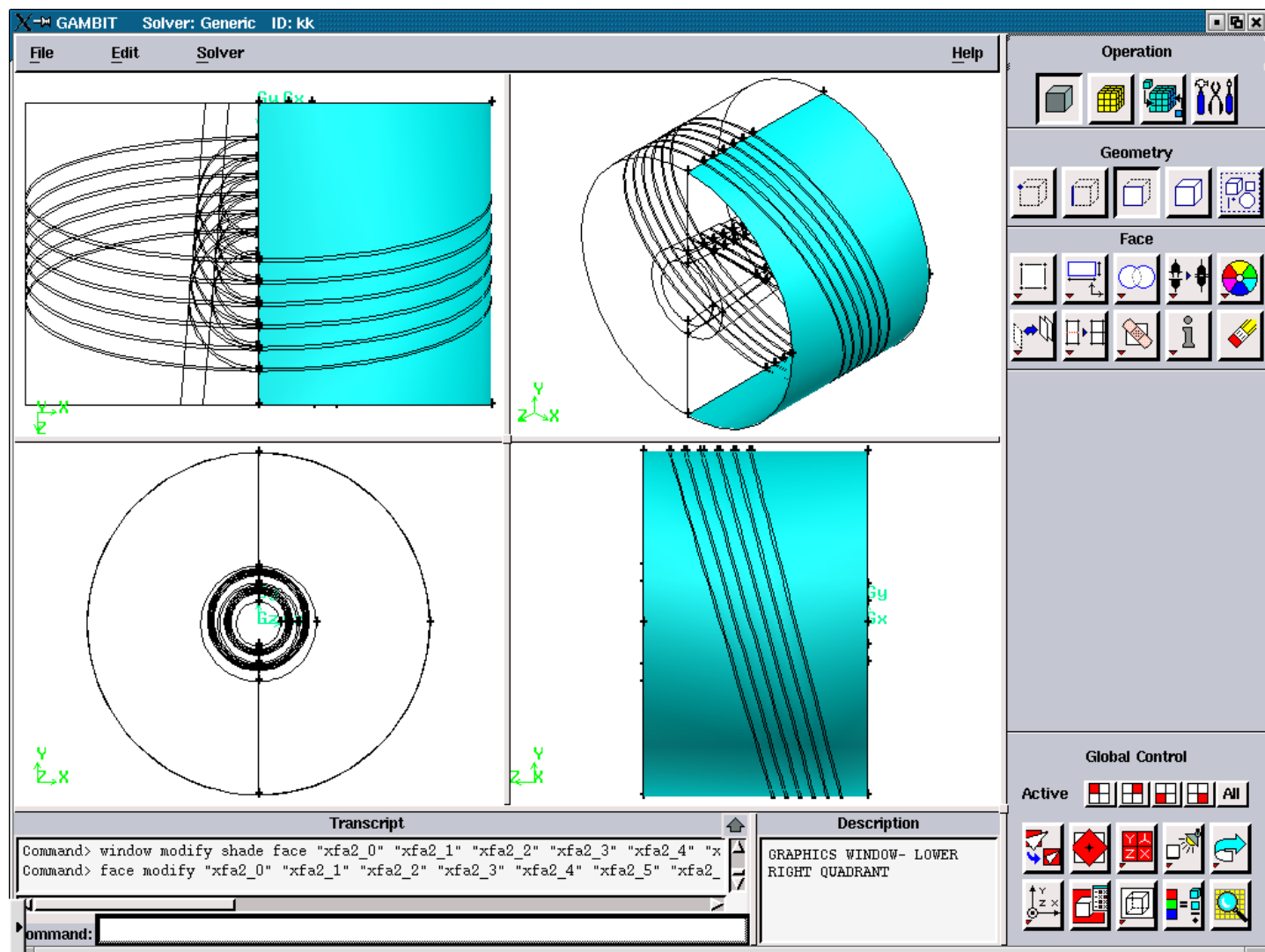


Figure 3.5.91. Axial Air Faces for the “right”.

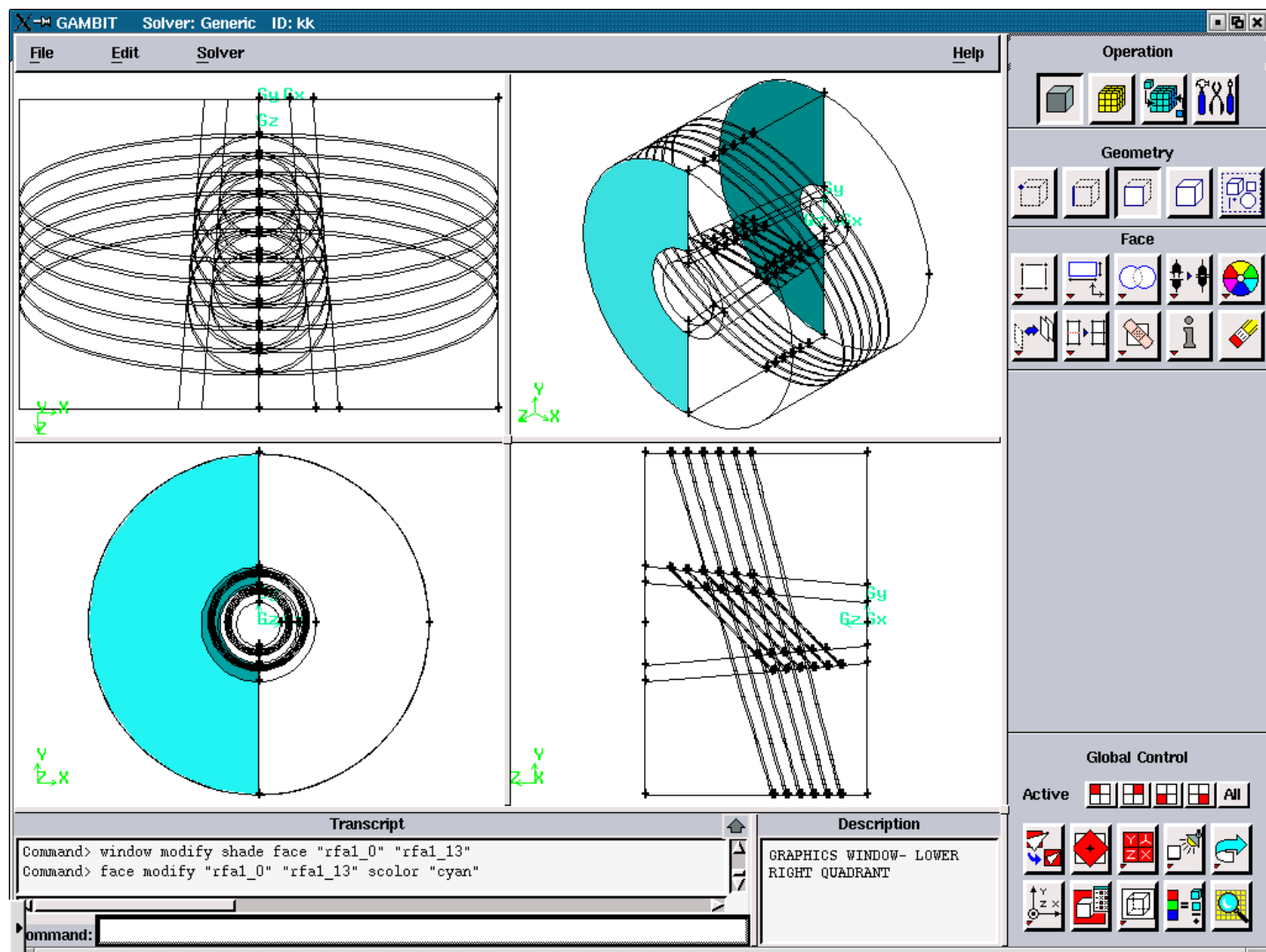


Figure 3.5.92. Radial Air Faces for the “left”.

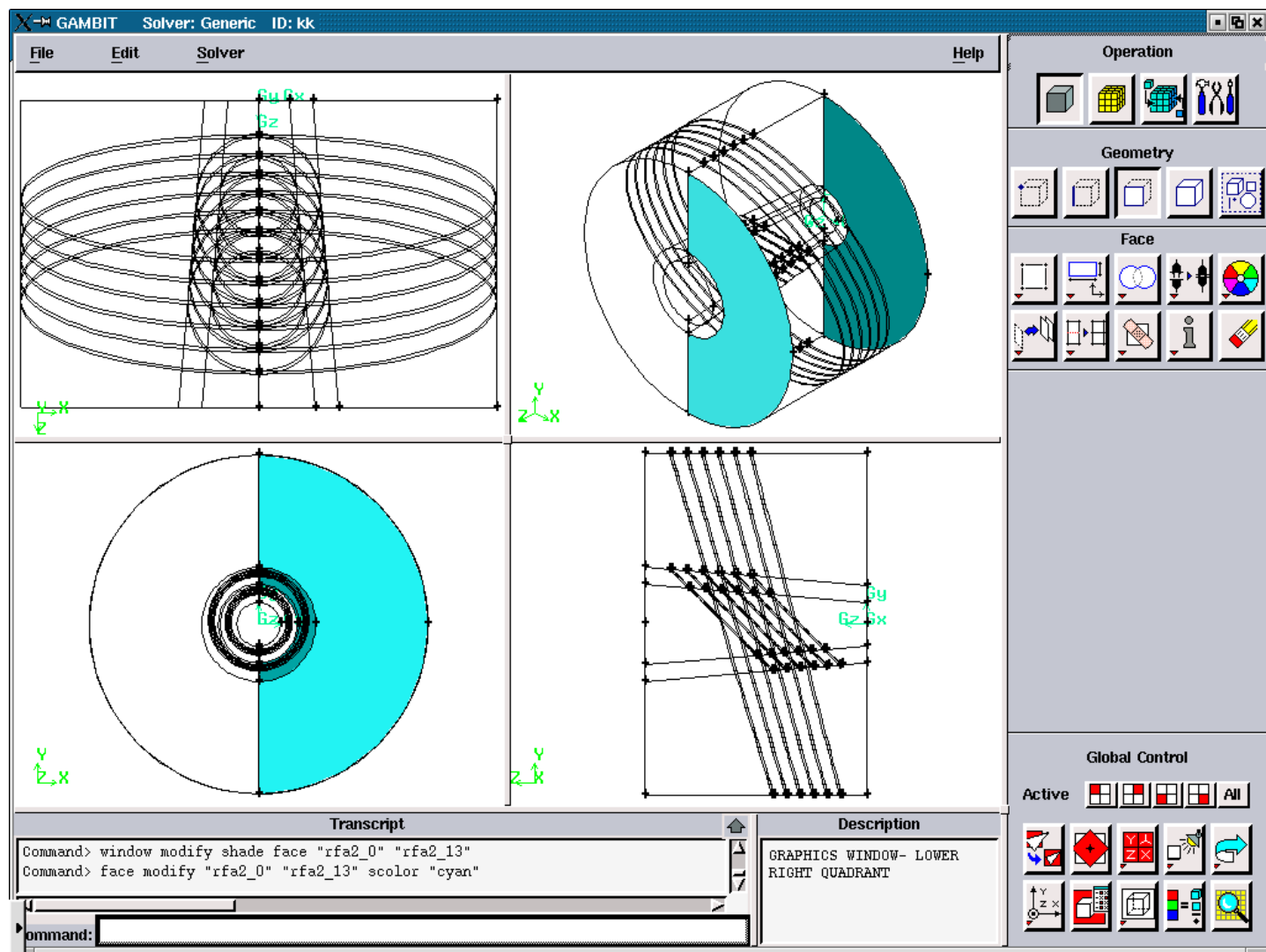


Figure 3.5.93. Radial Air Faces for the “right”.

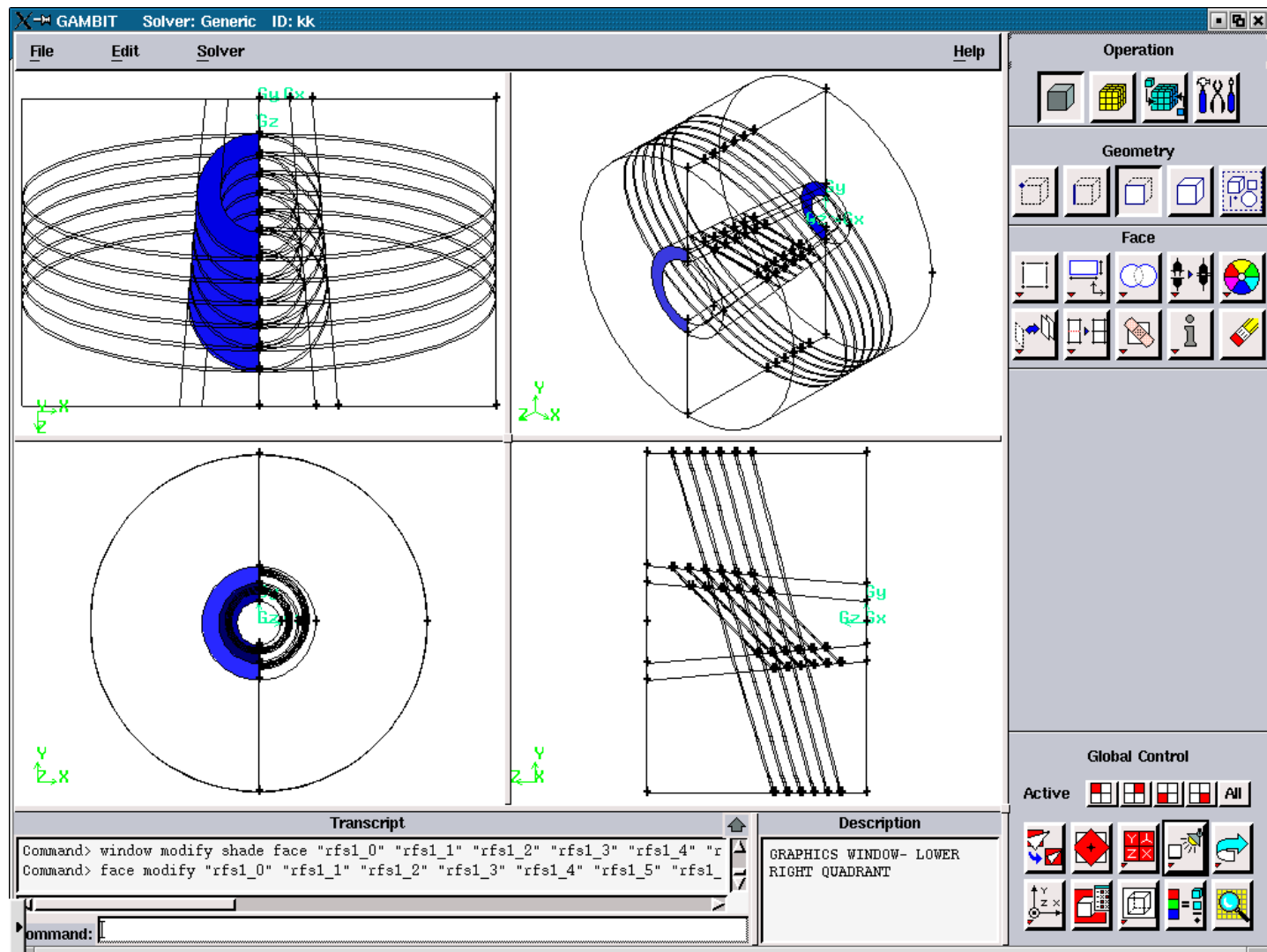


Figure 3.5.94. Radial Solid Faces for the “left”.

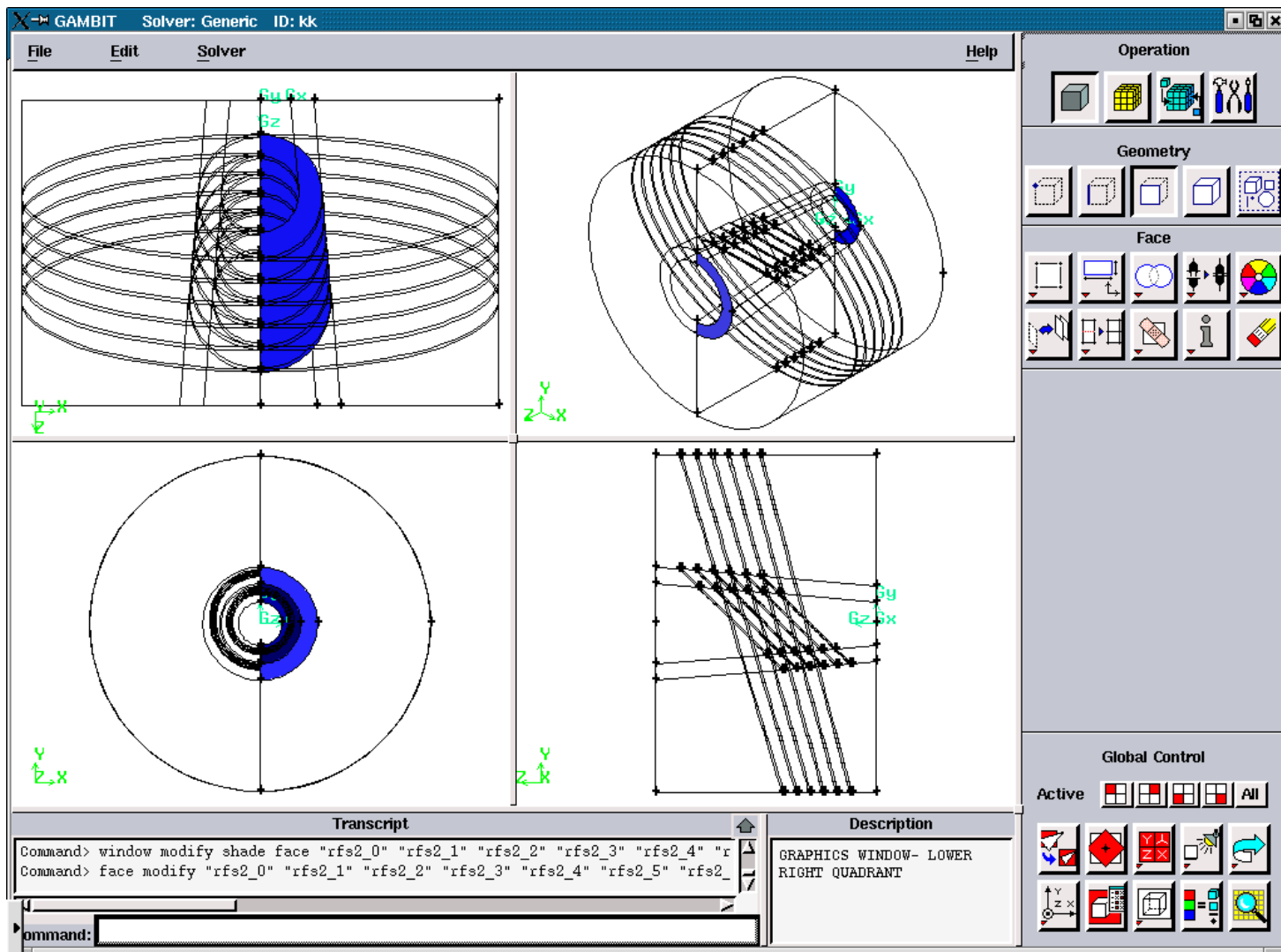


Figure 3.5.95. Radial Solid Faces for the “right”.

## **A1.5. FIGURES ILLUSTRATING THE CREATION OF THE FILE**

**‘modify3\_d’ and ‘modify3\_s’**

### **A1.5.3. FIGURES FOR THE CREATION OF THE FILE ‘modify3\_s’**

**FOR THE SLOPE ANGLES: 90, 90. (STRAIGHT CHANNEL)**

---

All figures in this section represent the geometry after the consequent run of the files:

‘temp1.jou’, ‘modify1’, temp2.jou’, modify2, ‘temp3.jou’

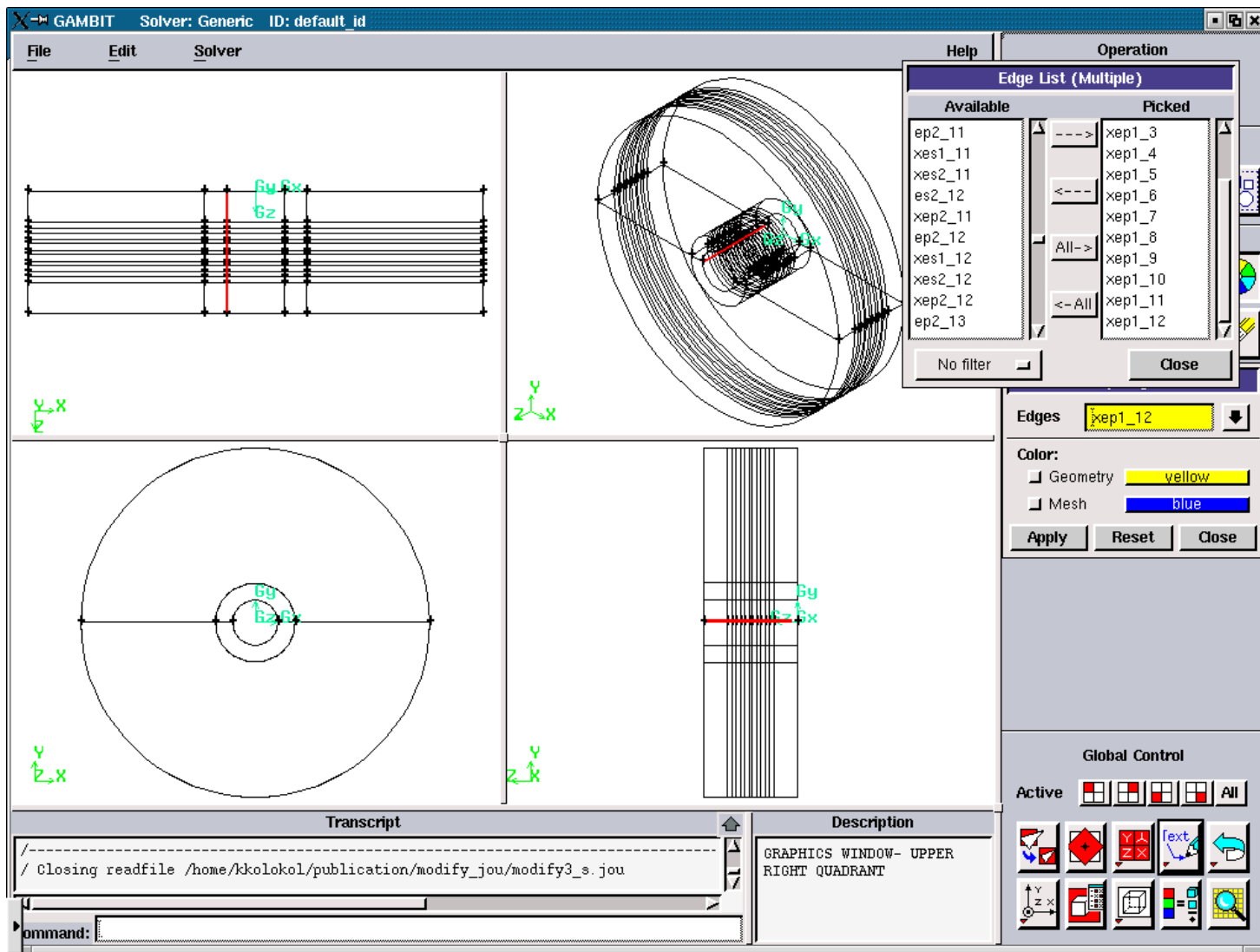


Figure 3.5.96. Axial Plasma Edges for the “left”.

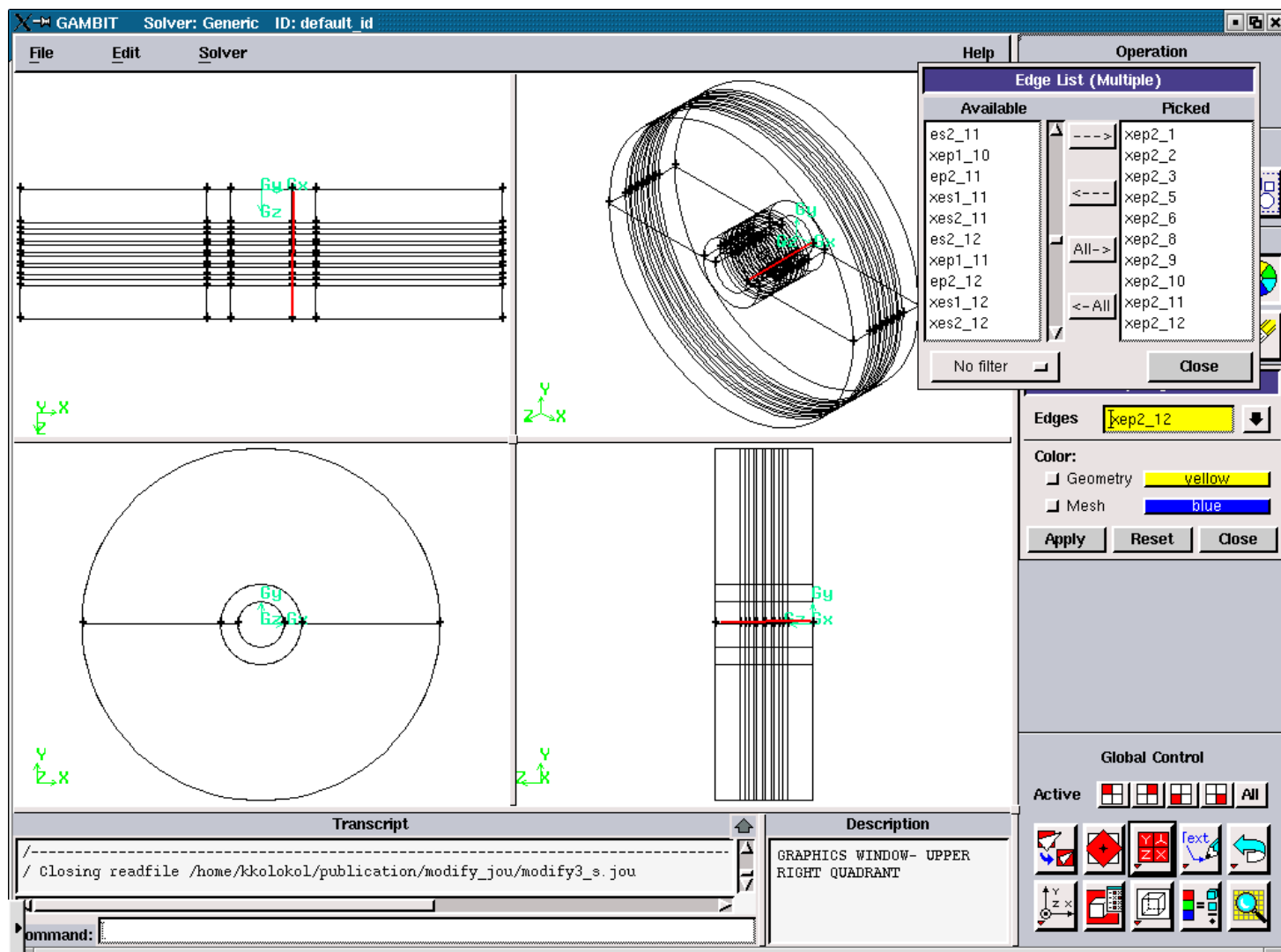


Figure 3.5.97. Axial Plasma Edges for the “right”.

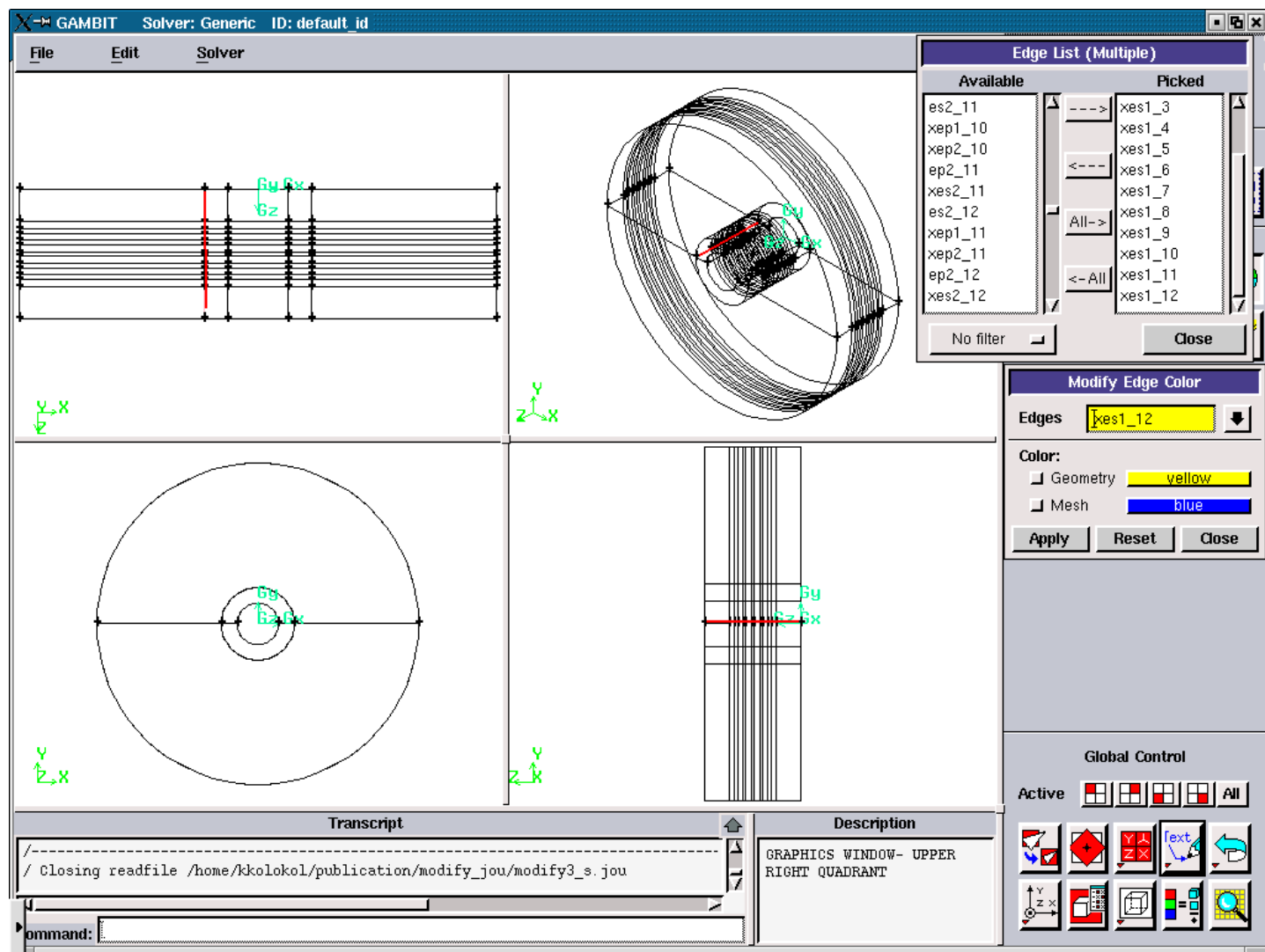


Figure 3.5.98. Axial Solid Edges for the “left”.

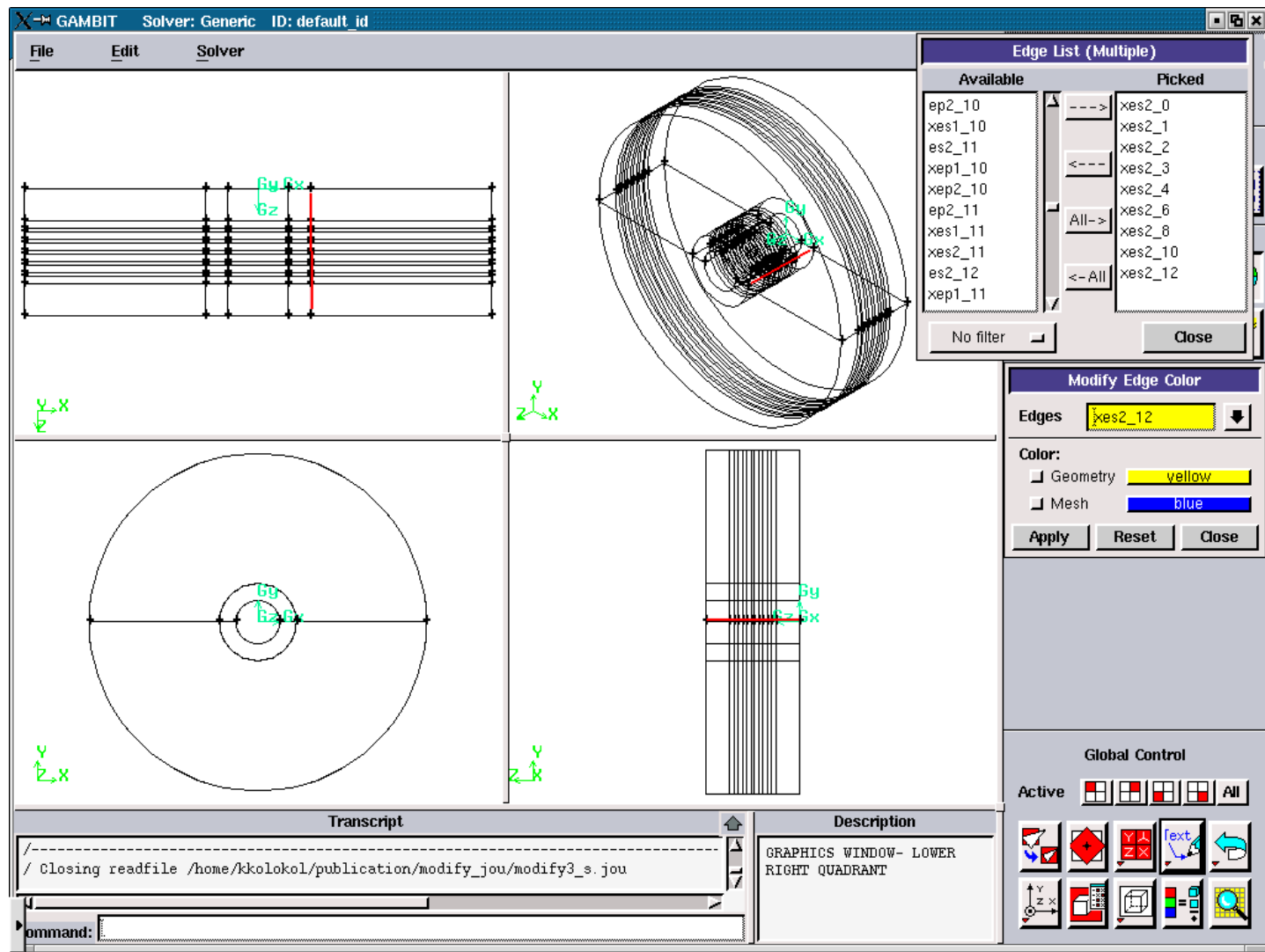


Figure 3.5.99. Axial Solid Edges for the “right”.

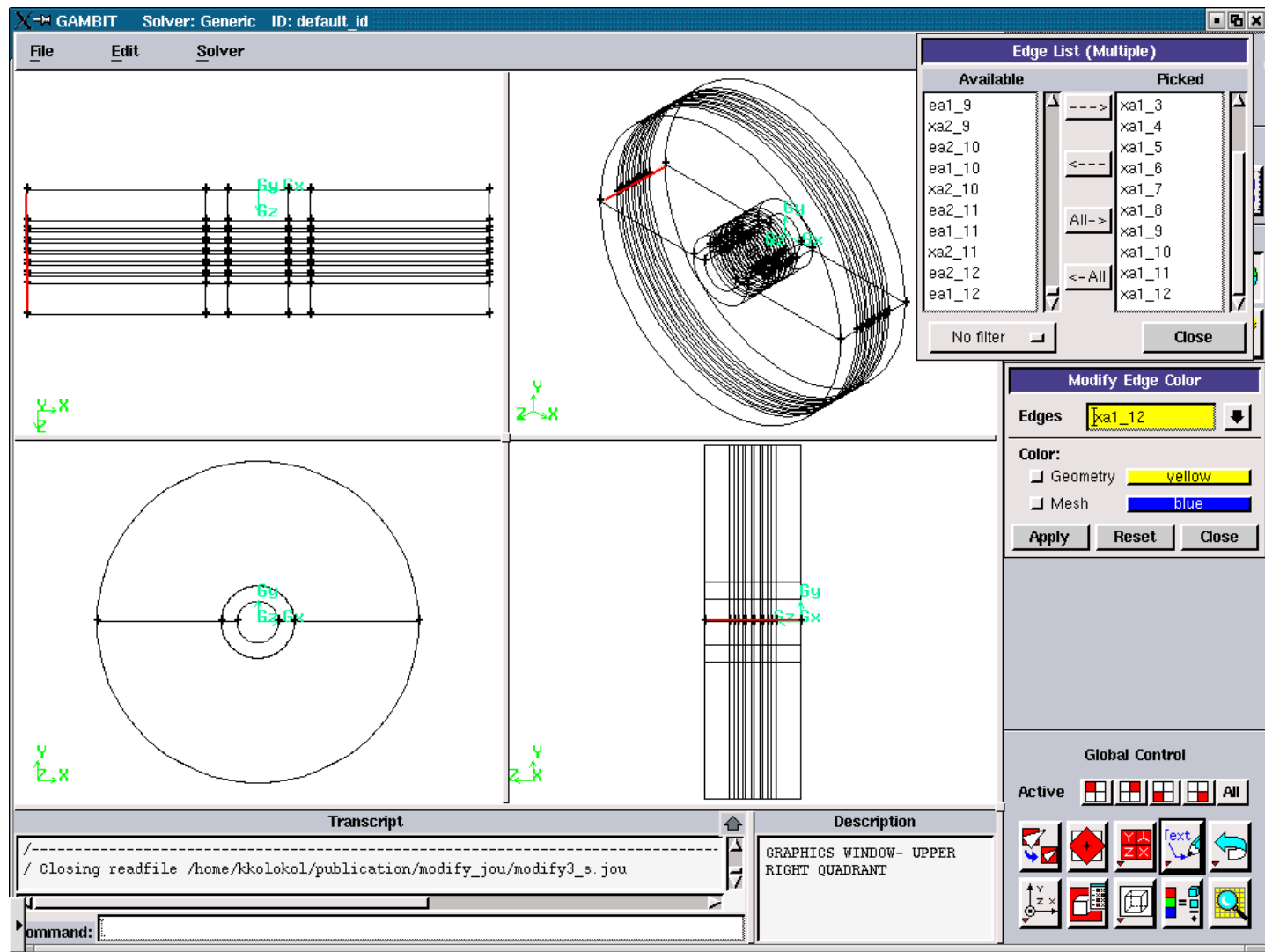


Figure 3.5.100. Axial Air Edges for the “left”.

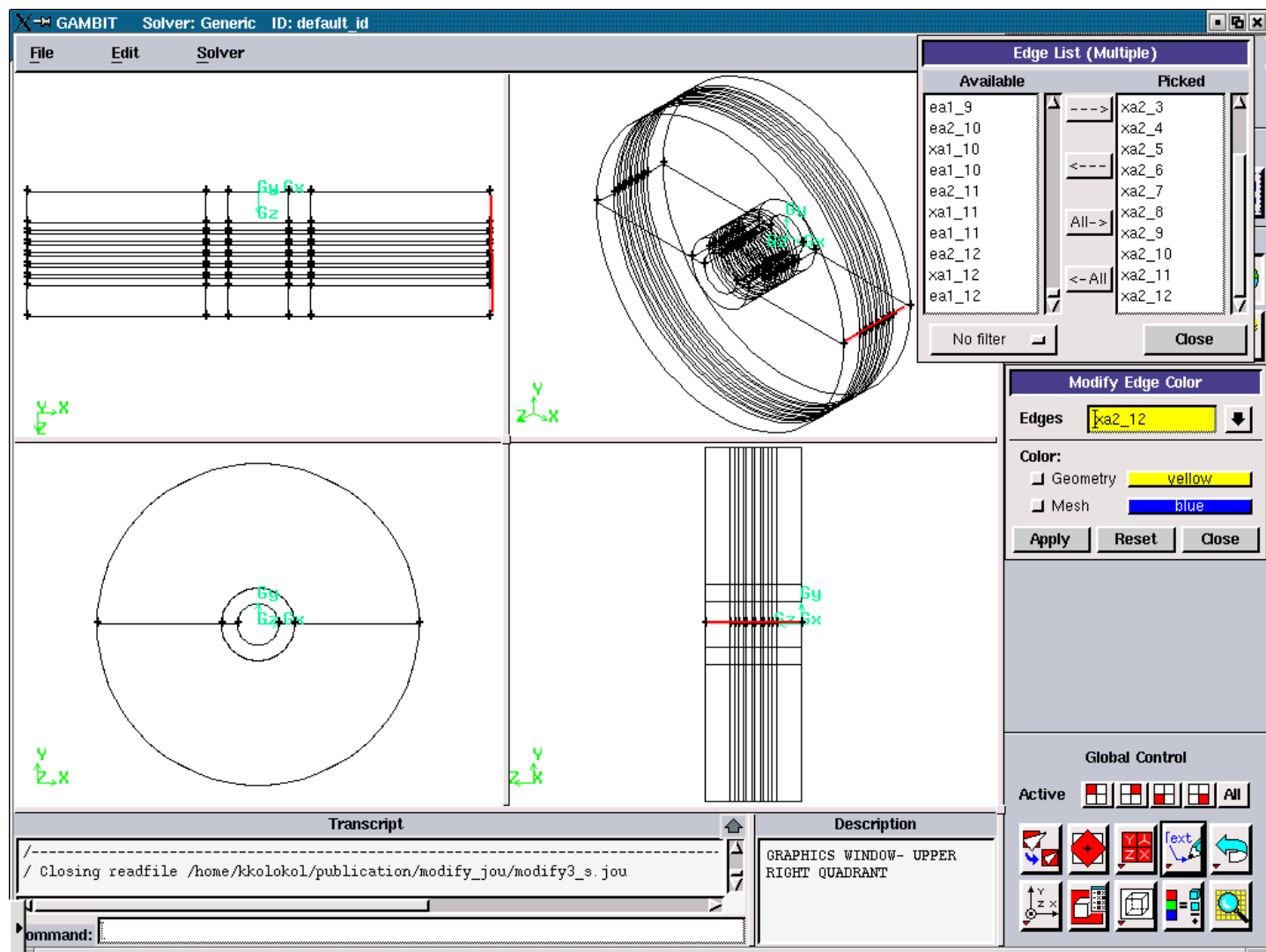
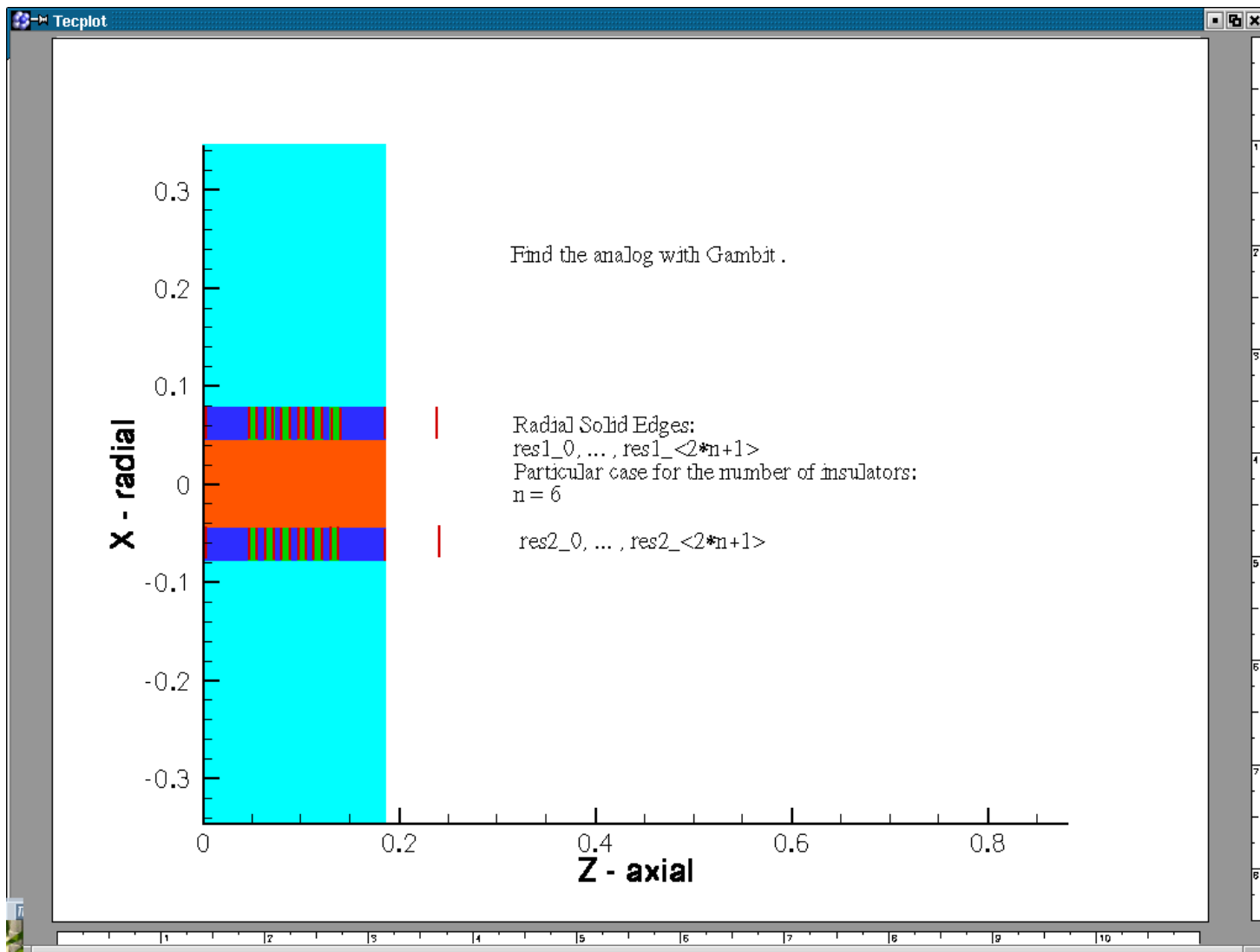


Figure 3.5.101. Axial Air Edges for the “right”.



**Figure 3.5.102.** Radial Edges for the “left” and “right”.

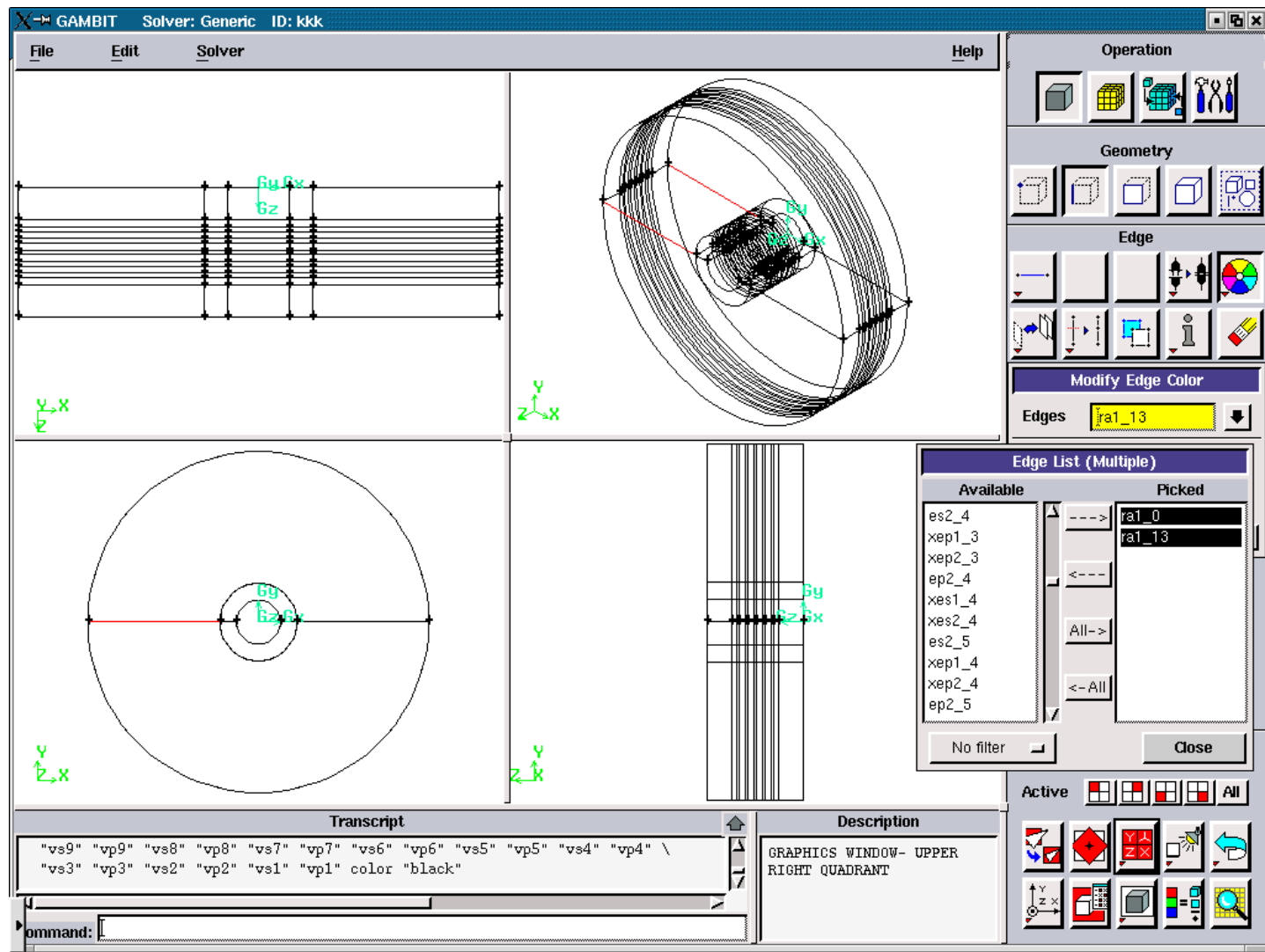


Figure 3.5.103. Radial Edges for the “left” and “right”.

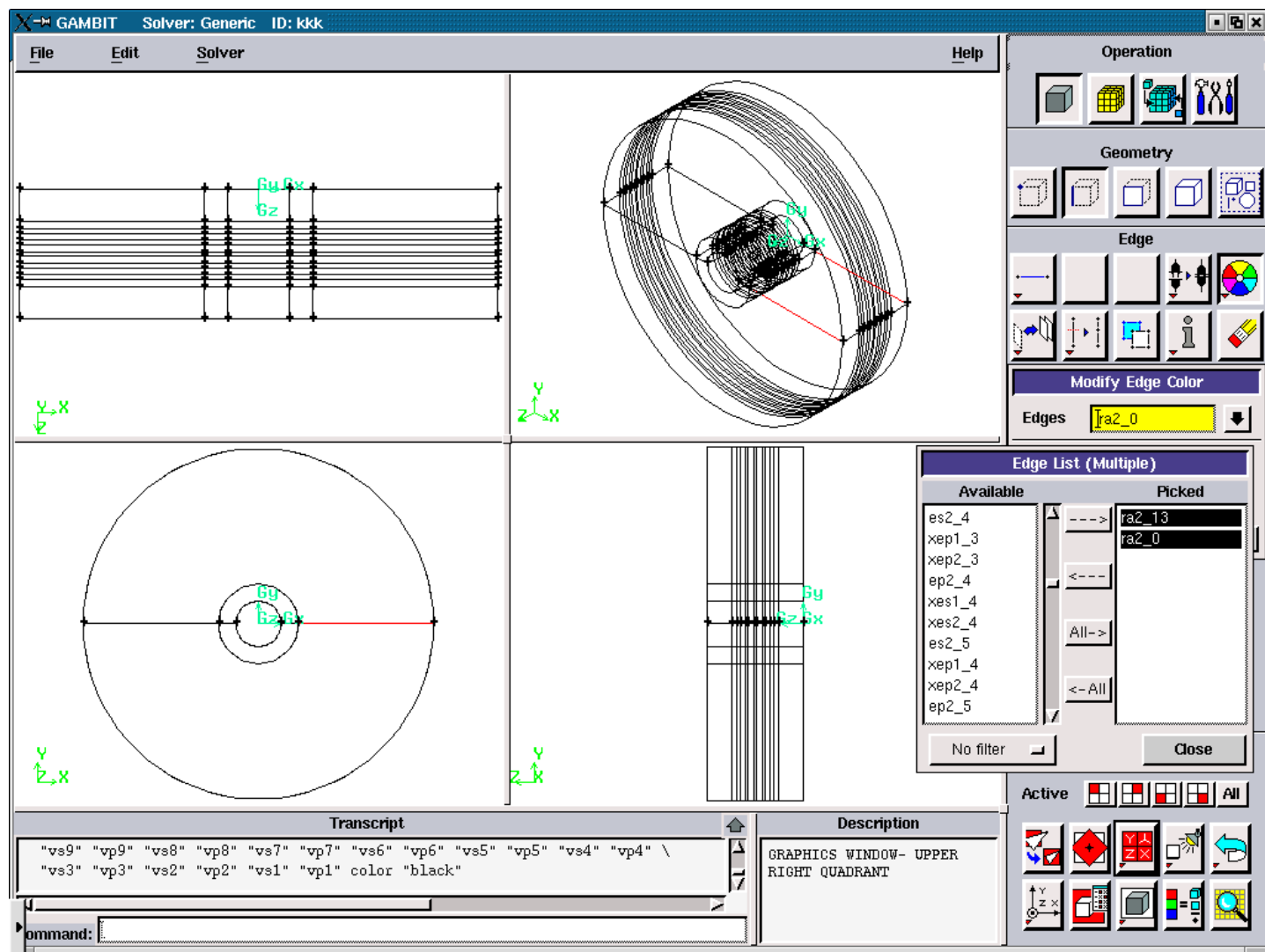


Figure 3.5.104. Radial Air Edges for the “right”.

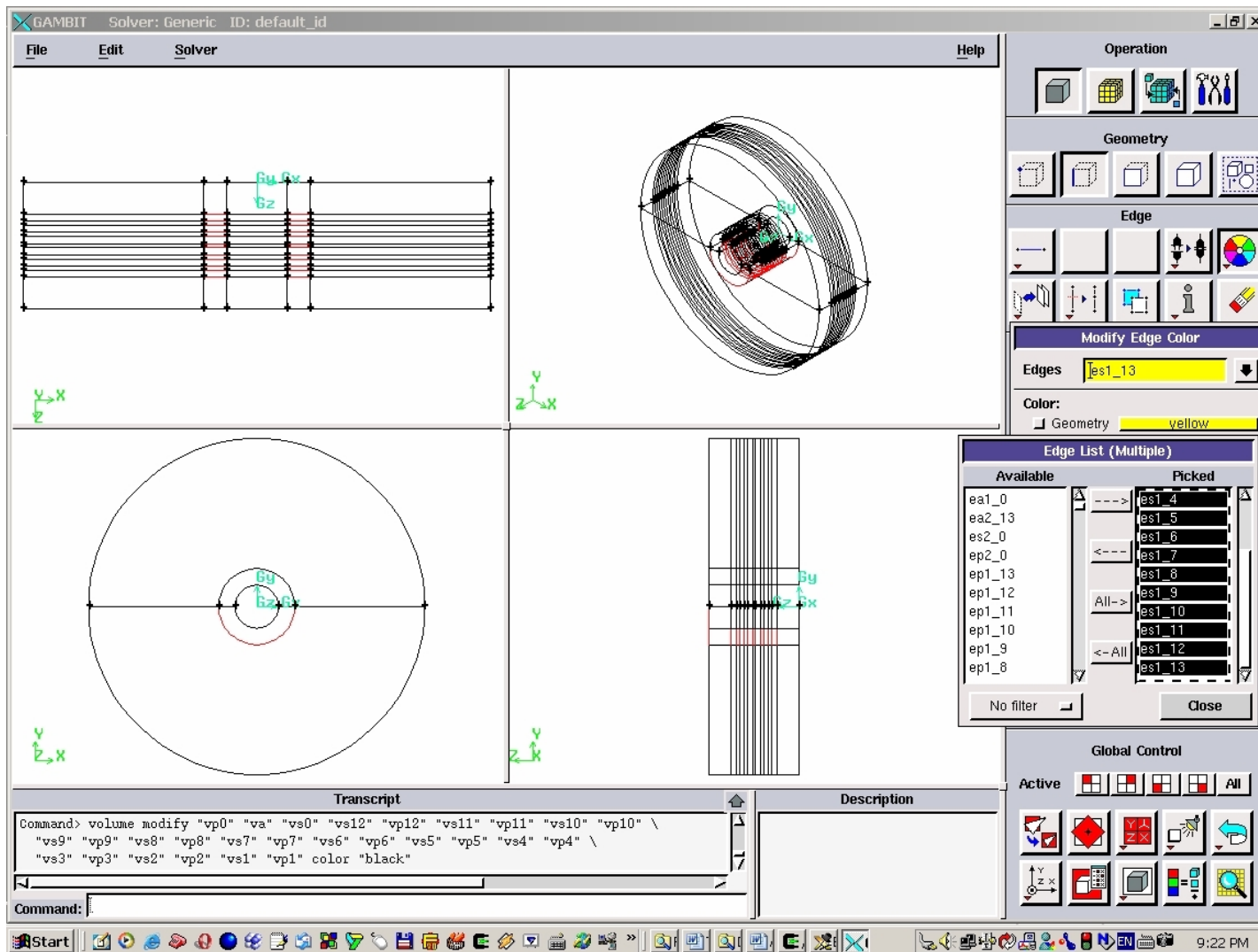


Figure 3.5.105. Round (Elliptic) Solid Edges for the “down”.

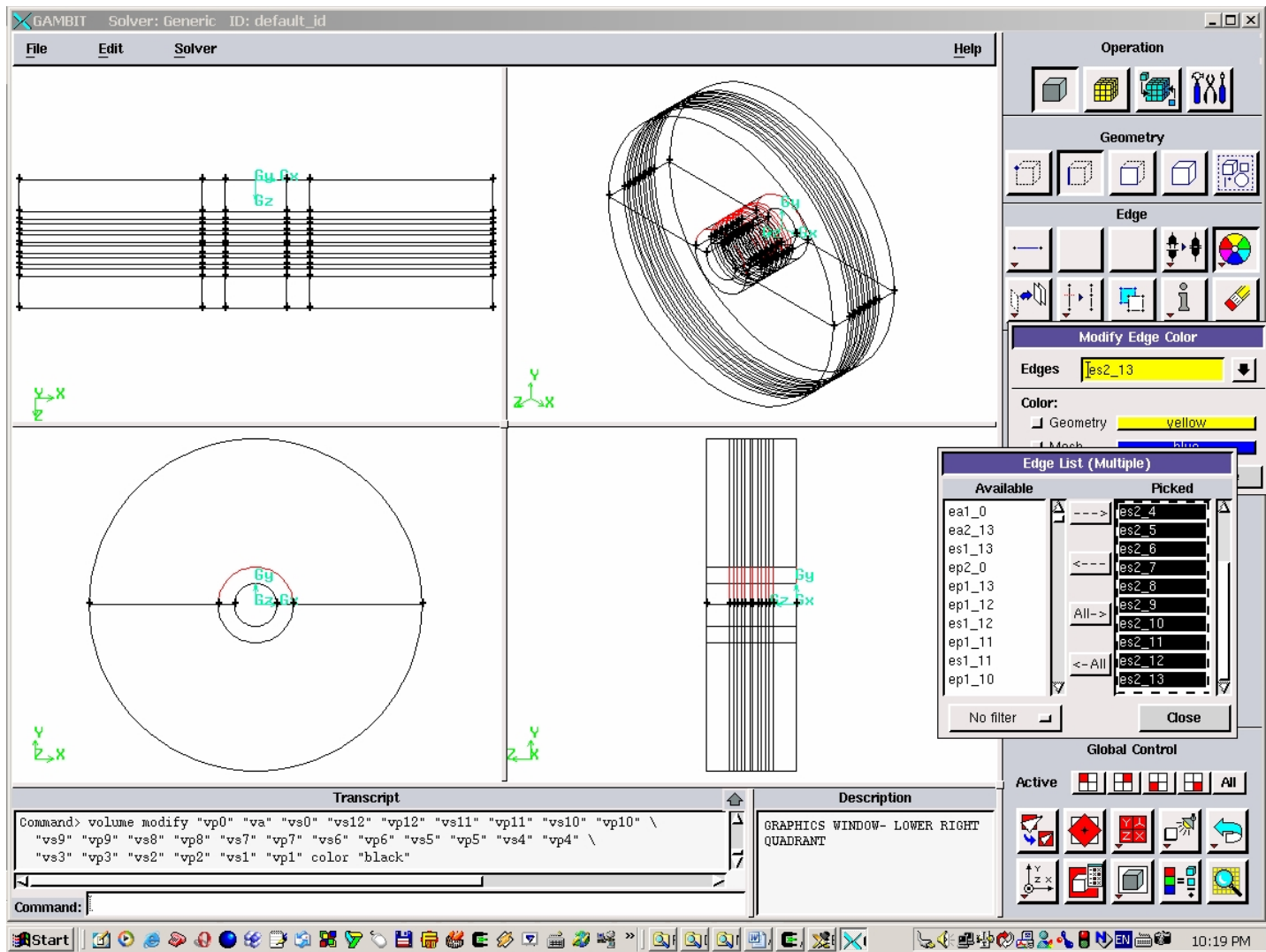


Figure 3.5.106. Round (Elliptic) Solid Edges for the “up”.

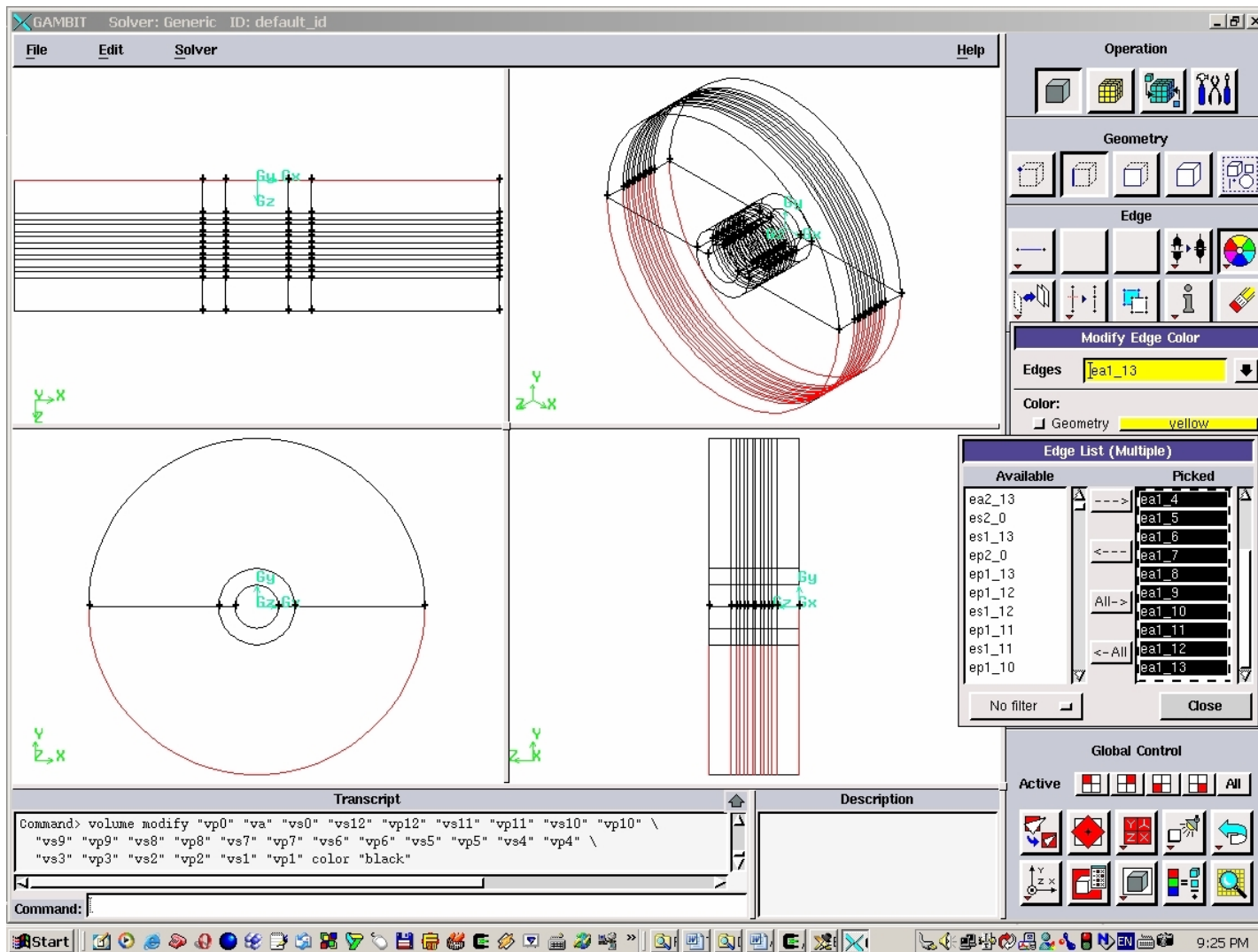


Figure 3.5.107. Round (Elliptic) Air Edges for the “down”.

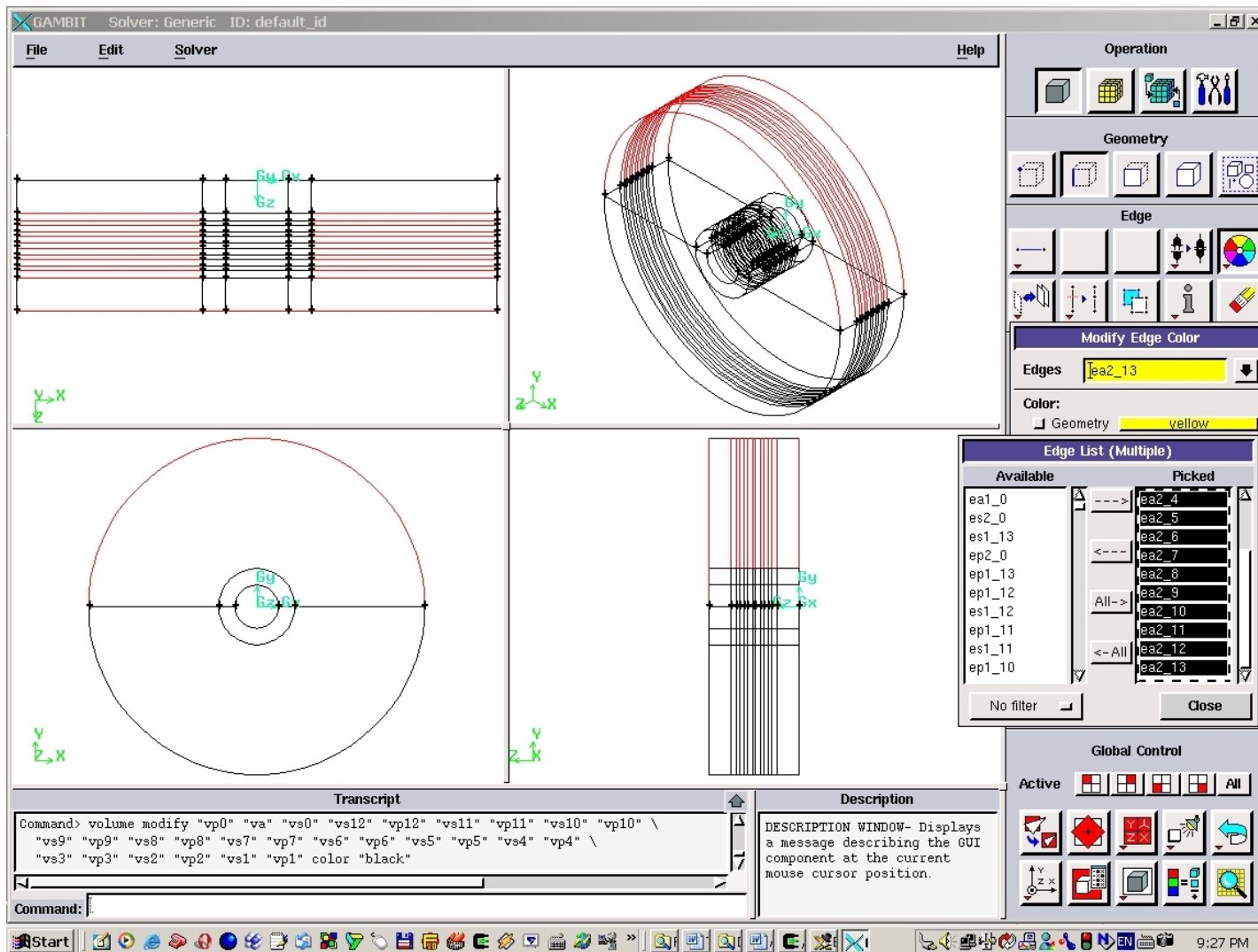


Figure 3.5.108. Round (Elliptic) Air Edges for the “up”.

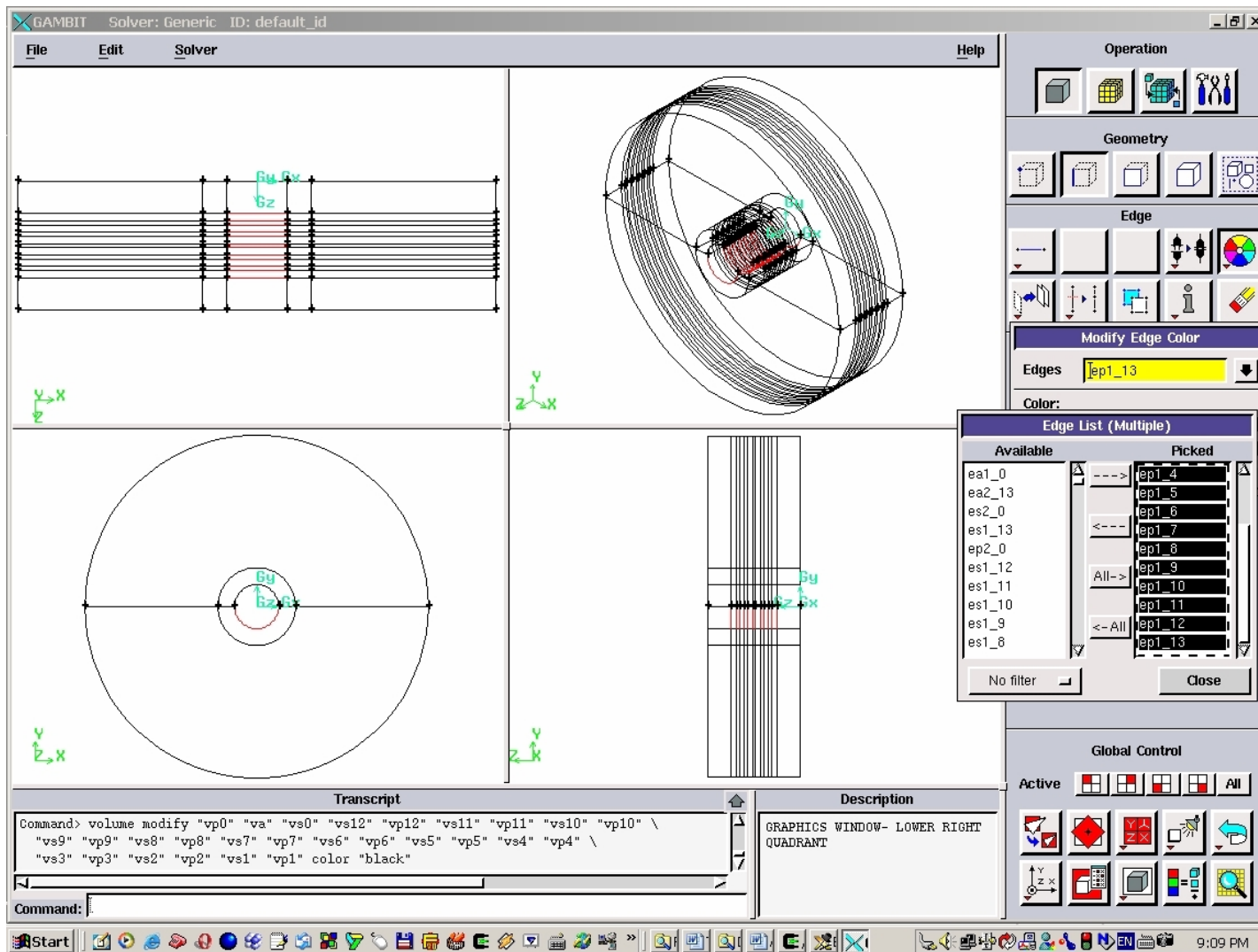


Figure 3.5.109. Round (Elliptic) Plasma Edges for the “down”.

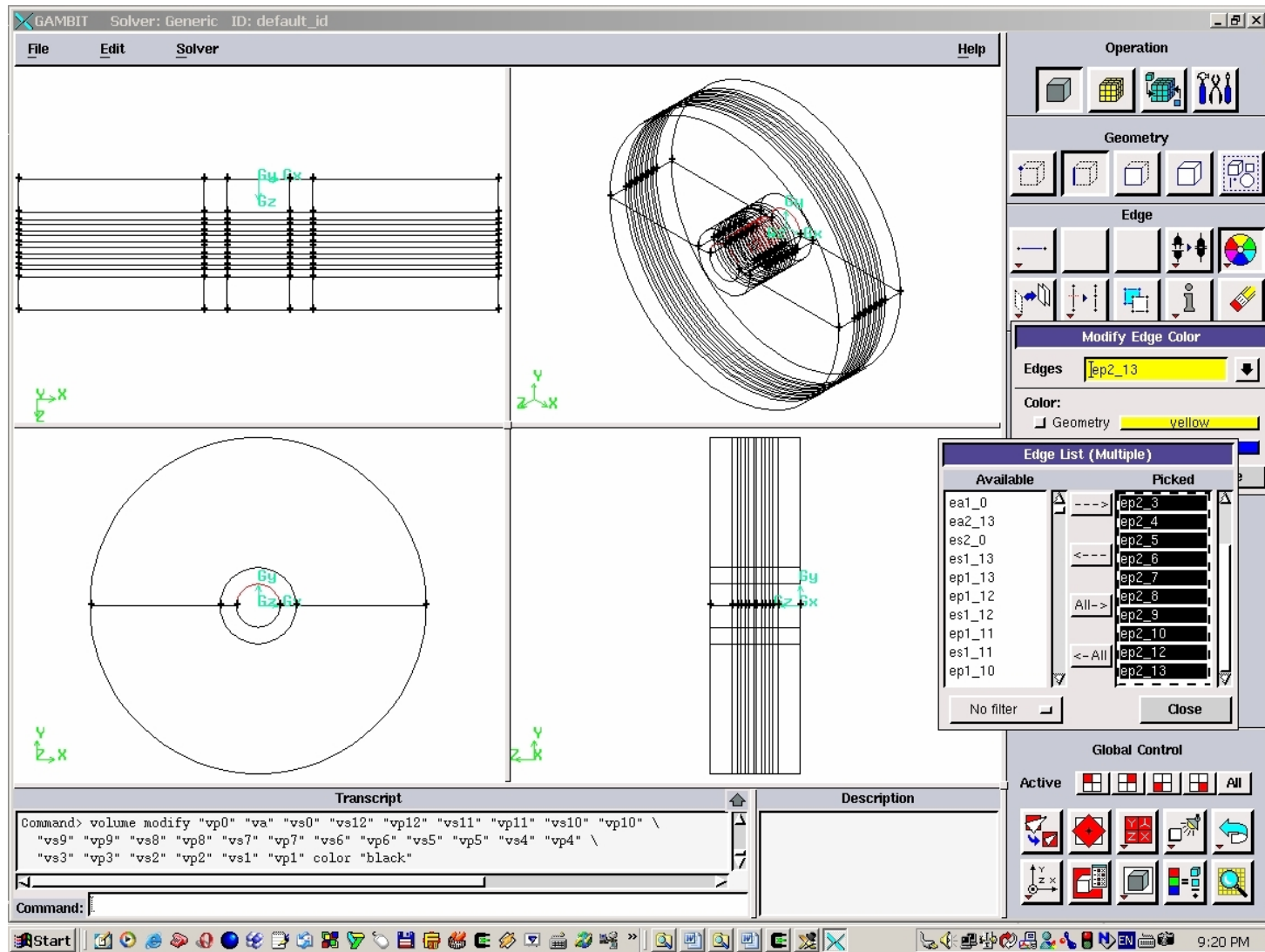


Figure 3.5.110. Round (Elliptic) Plasma Edges for the “up”.

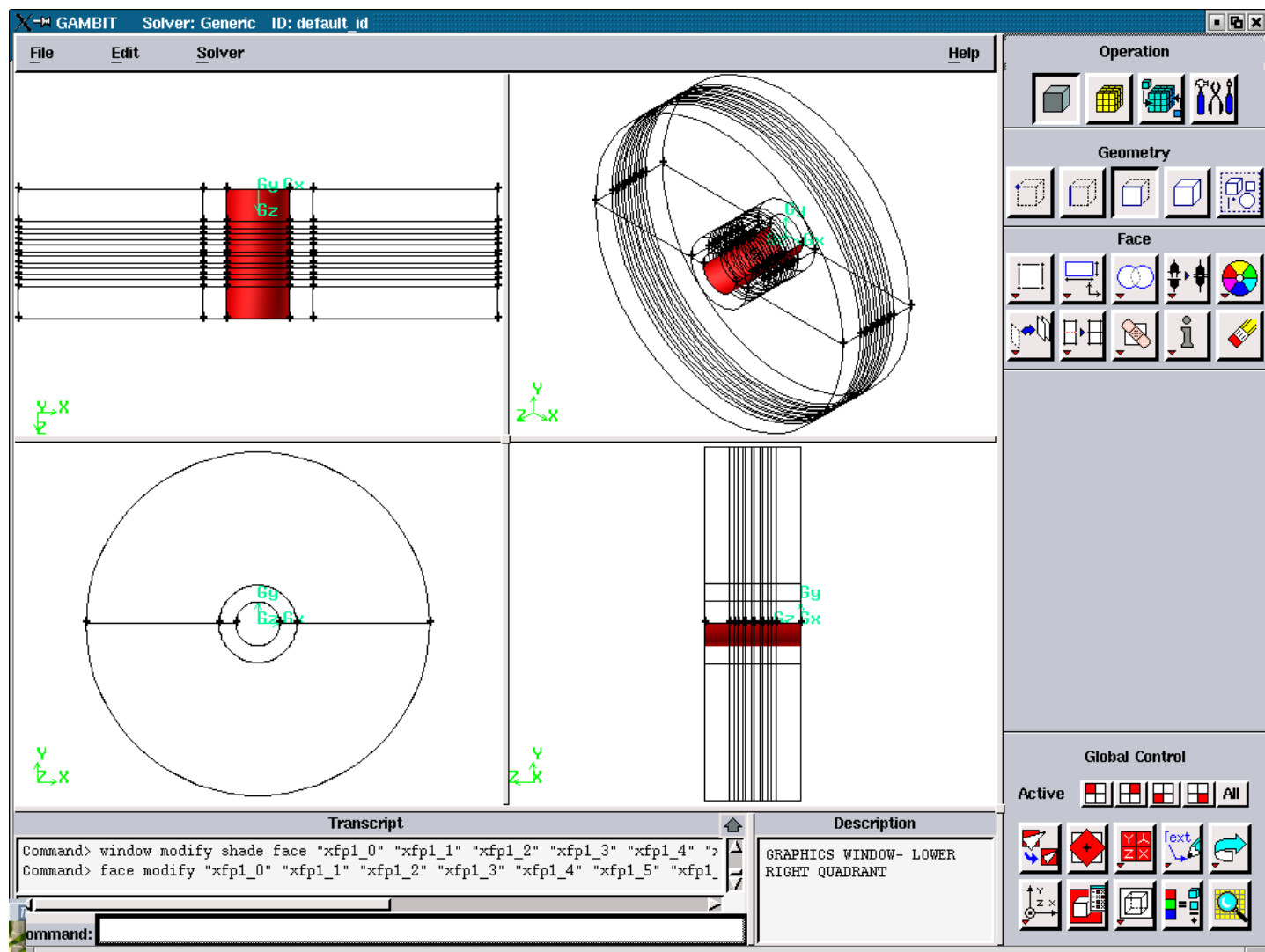


Figure 3.5.111. Axial Plasma Faces for the “down”.

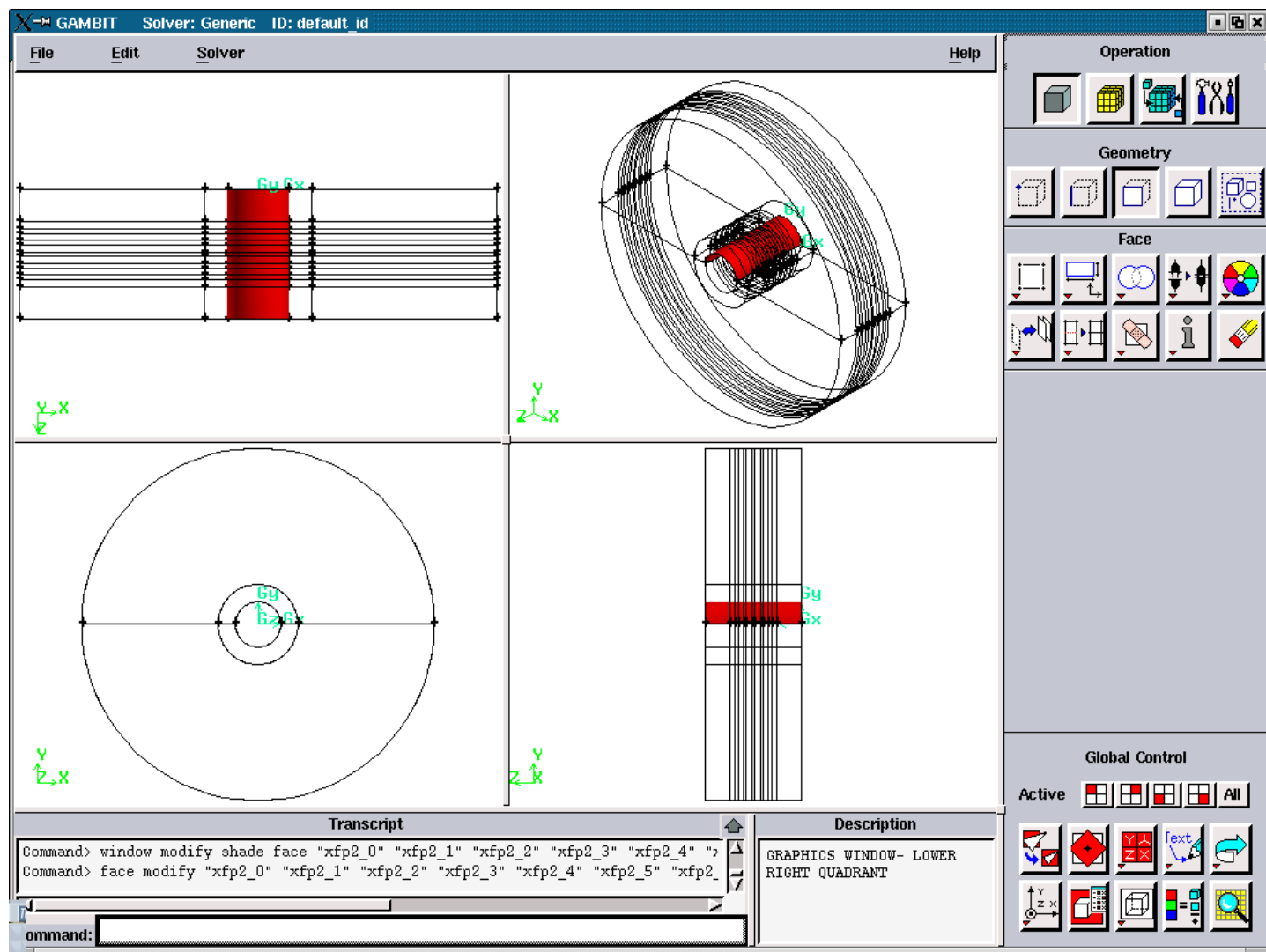


Figure 3.5.112. Axial Plasma Faces for the “up”.



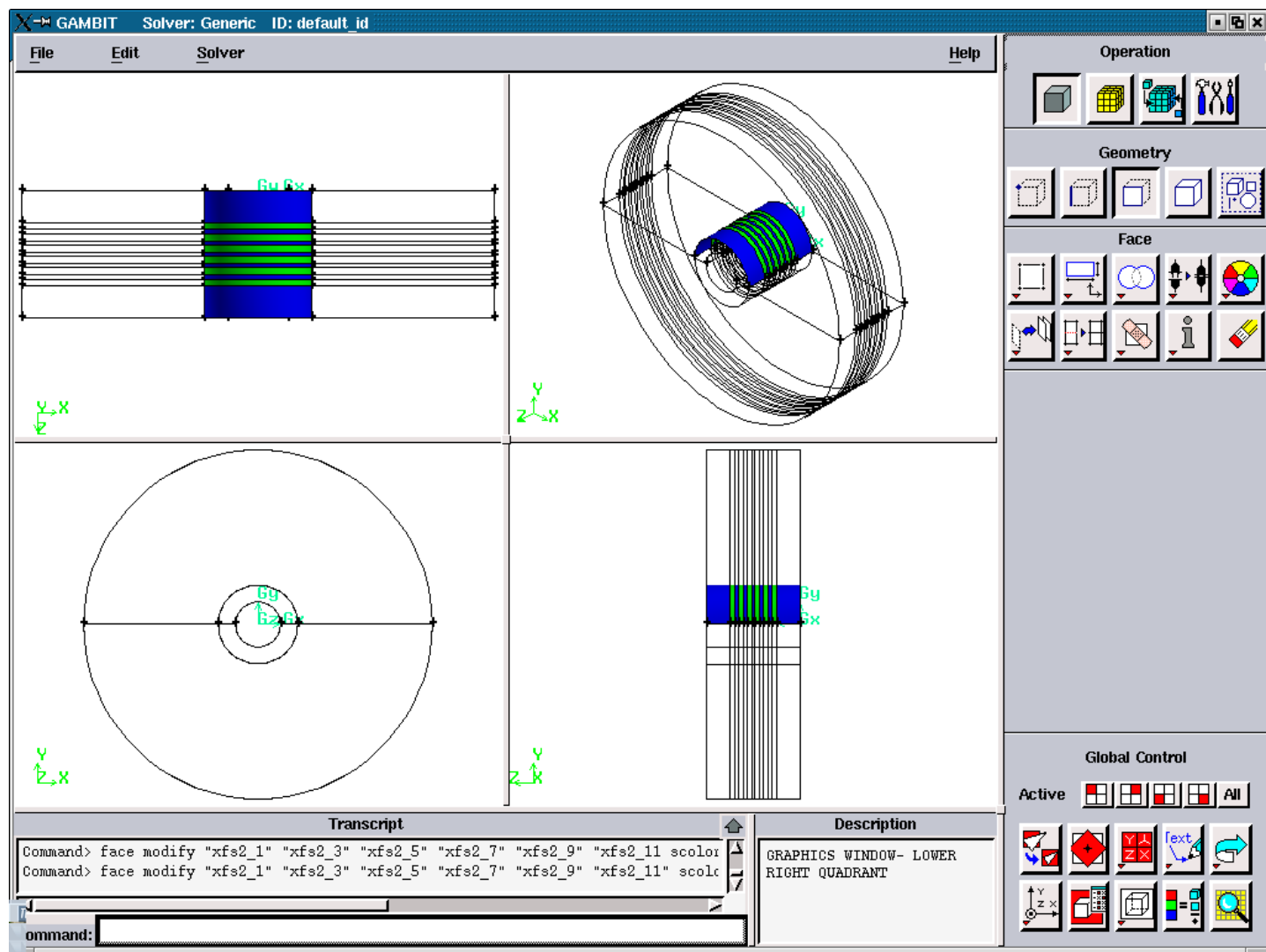


Figure 3.5.114. Axial Solid Faces for the “up”.

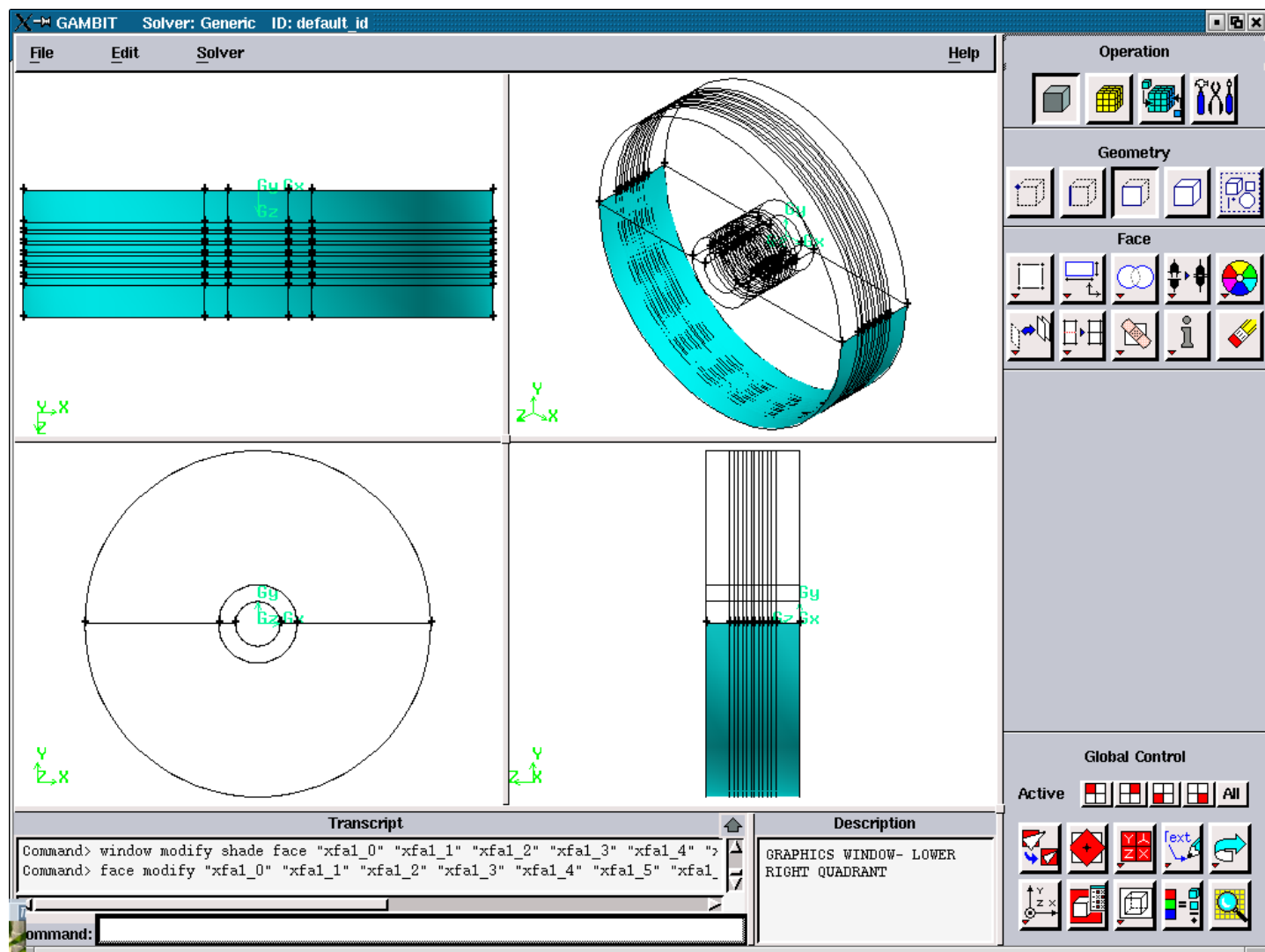


Figure 3.5.115. Axial Air Faces for the “down”.

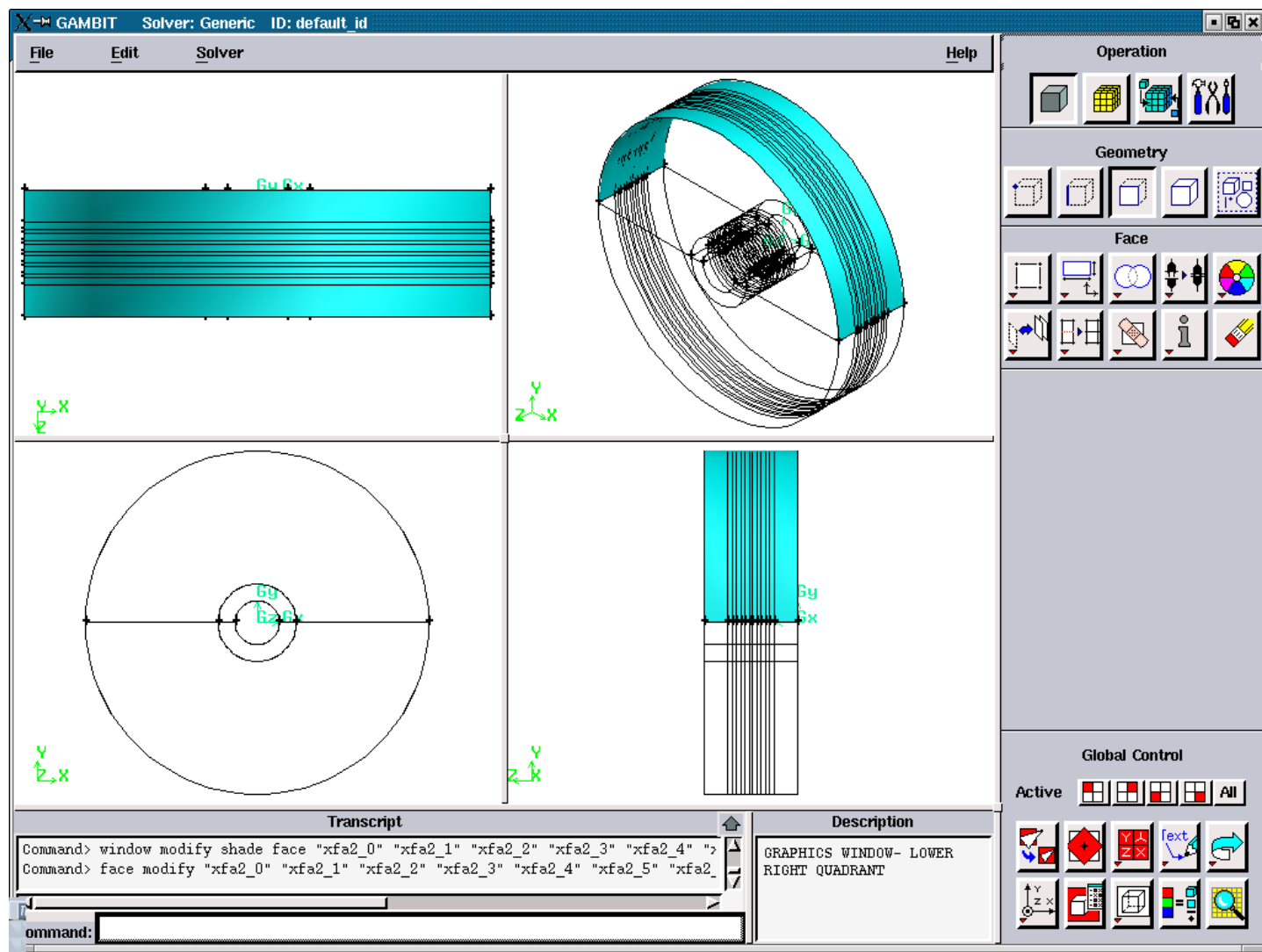


Figure 3.5.116. Axial Air Faces for the “up”.

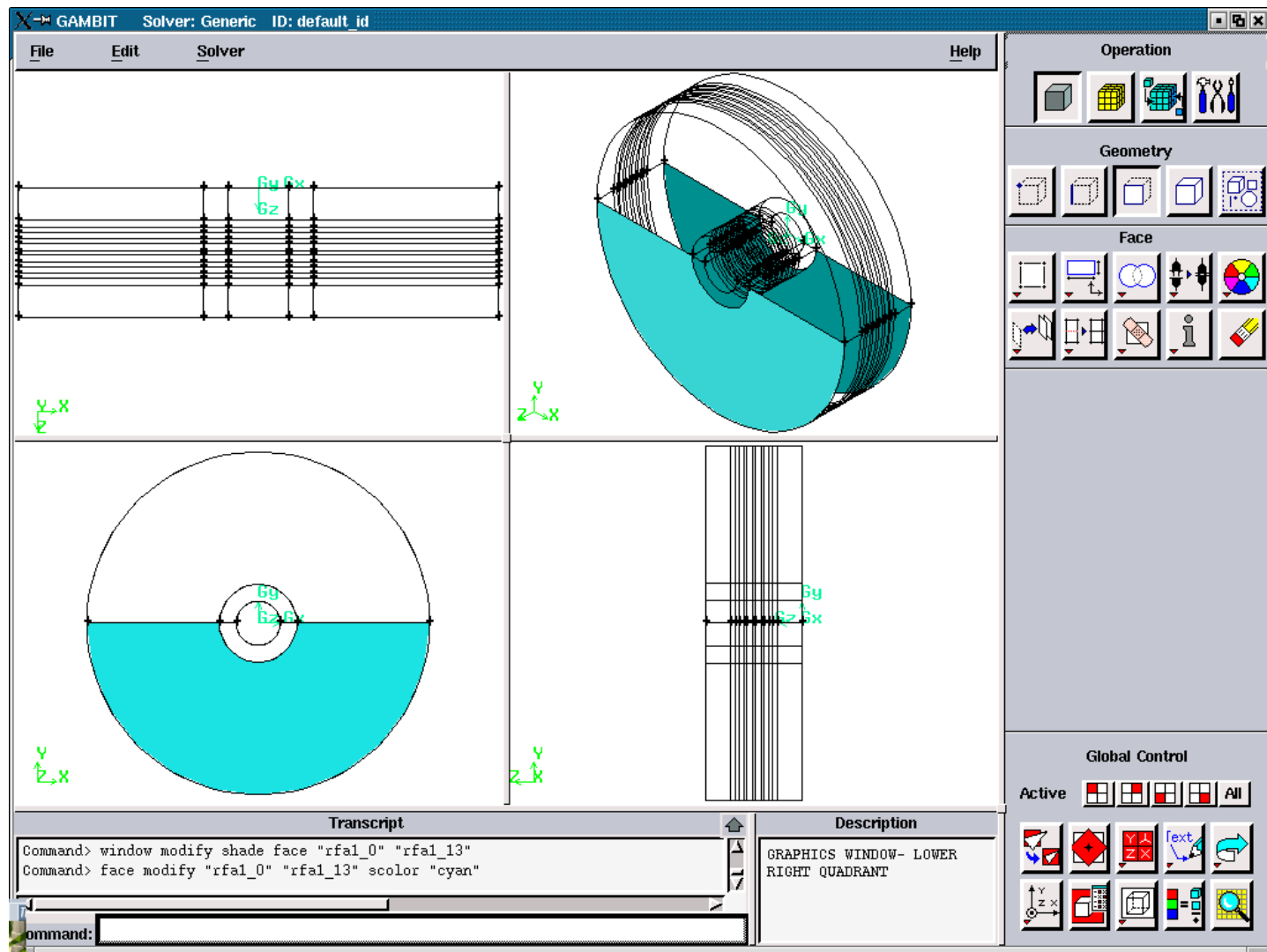


Figure 3.5.117. Radial Air Faces for the “down”.

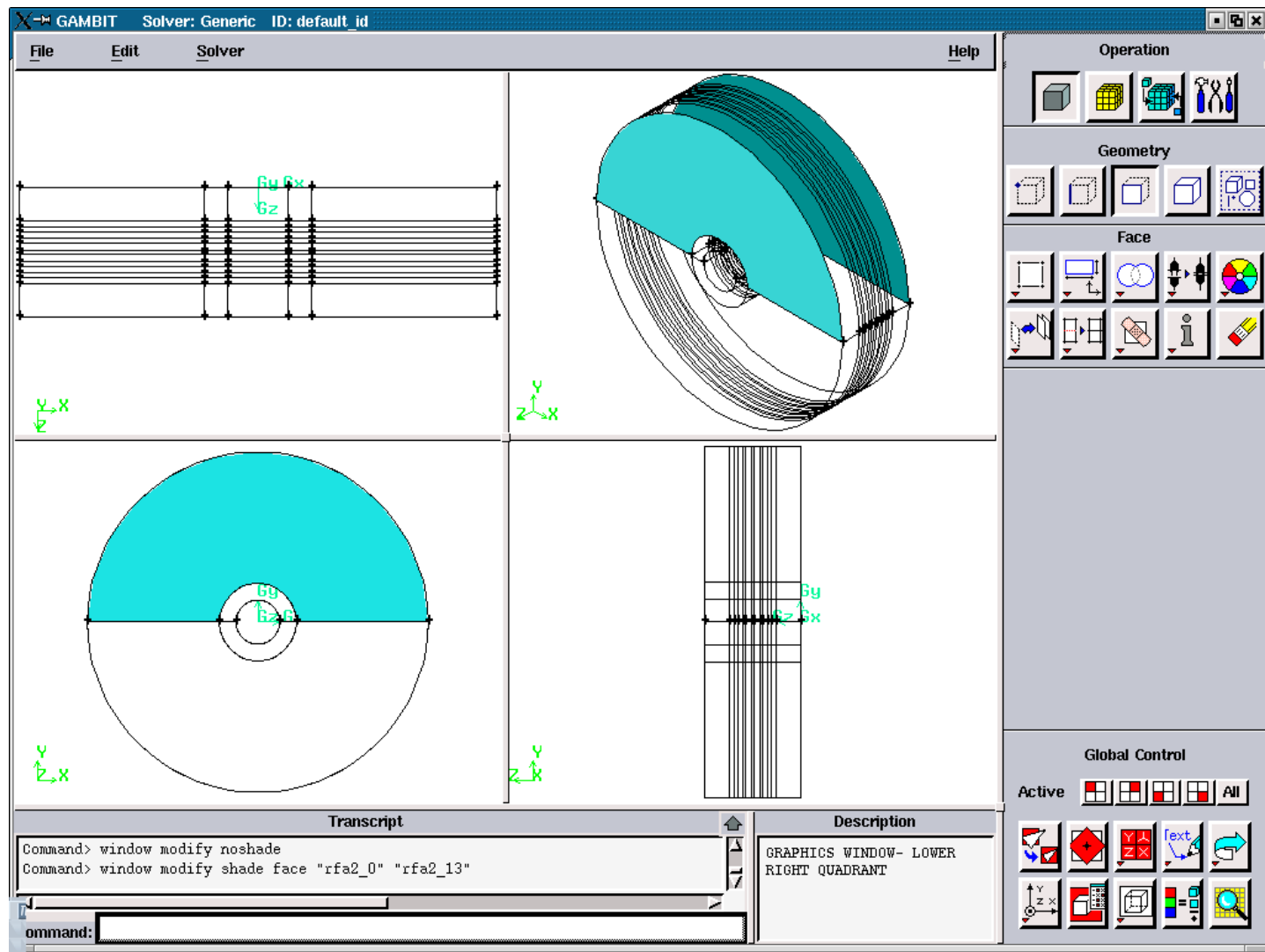


Figure 3.5.118. Radial Air Faces for the “up”.





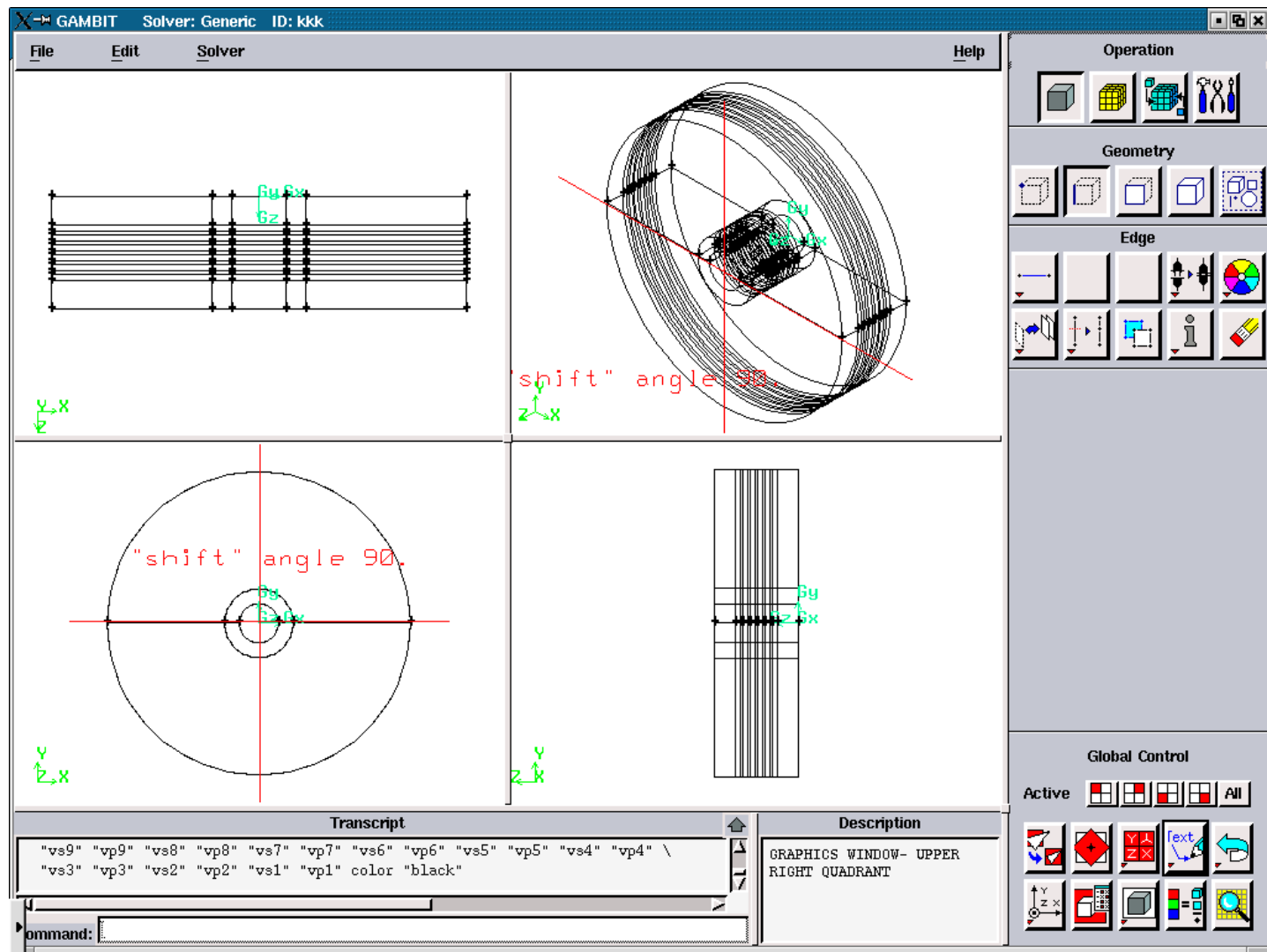
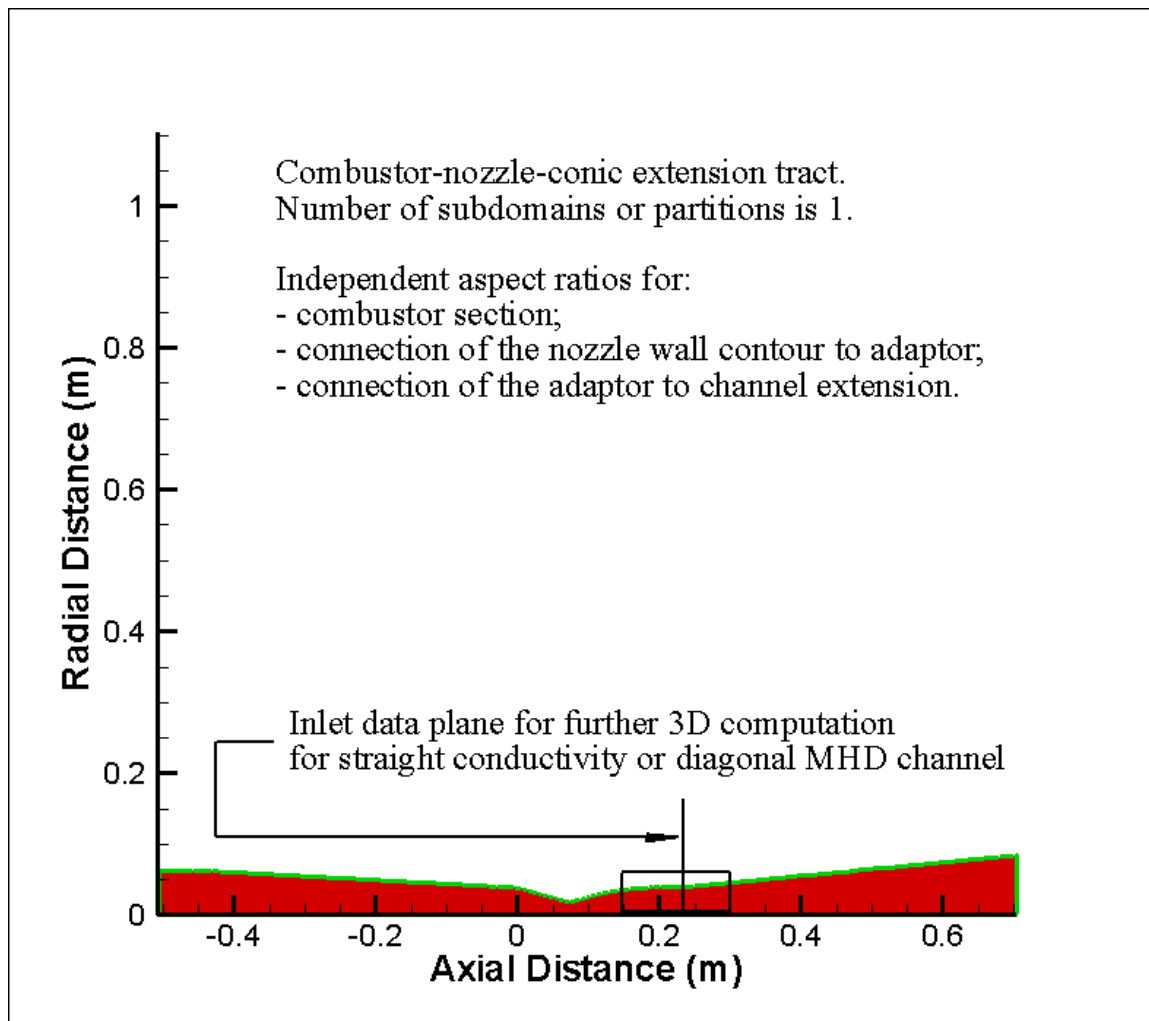


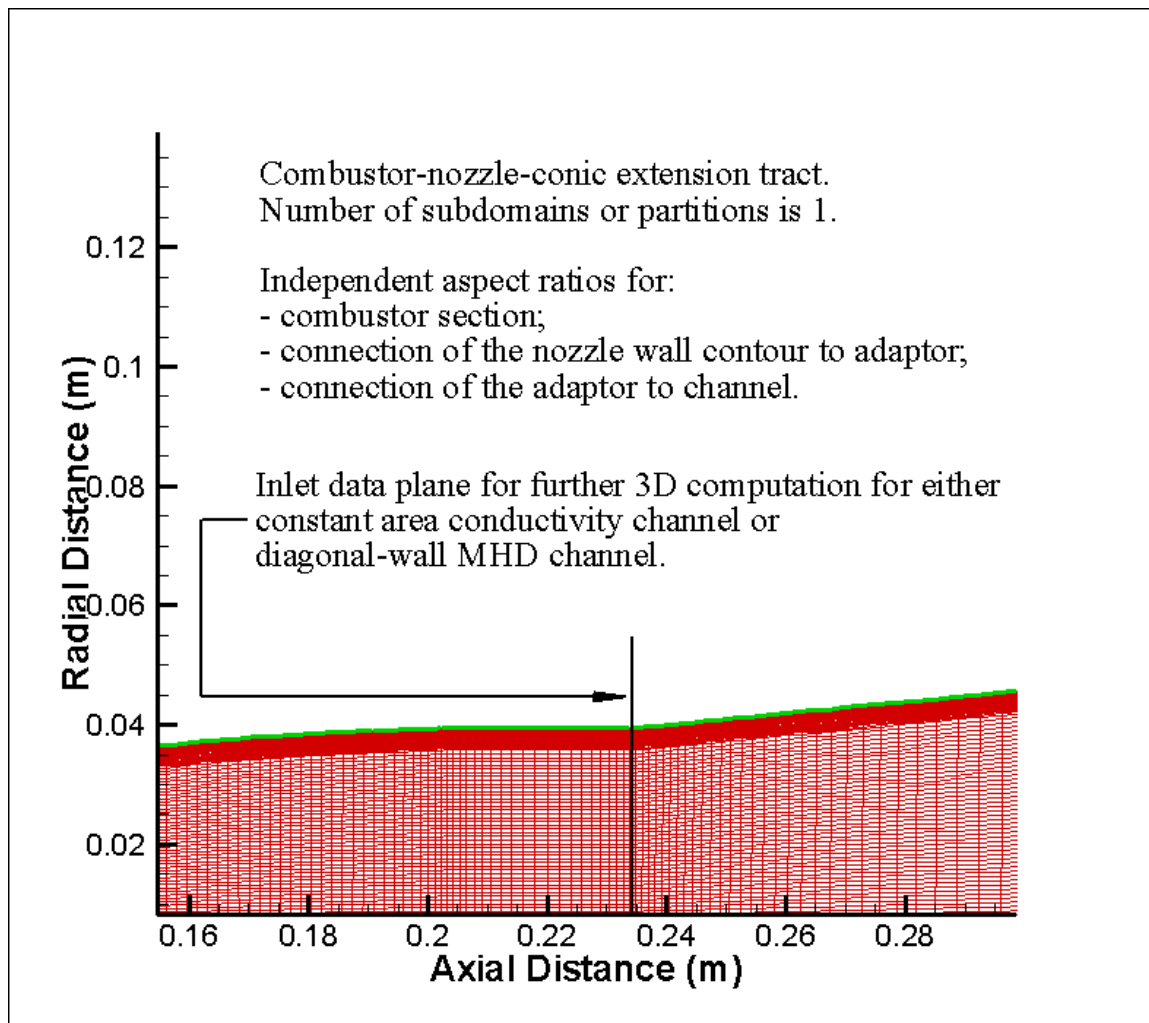
Figure 3.5.121. The “shift” angle for the Straight Channel (slope angles are: 90, 90).

## **APPENDIX A2**

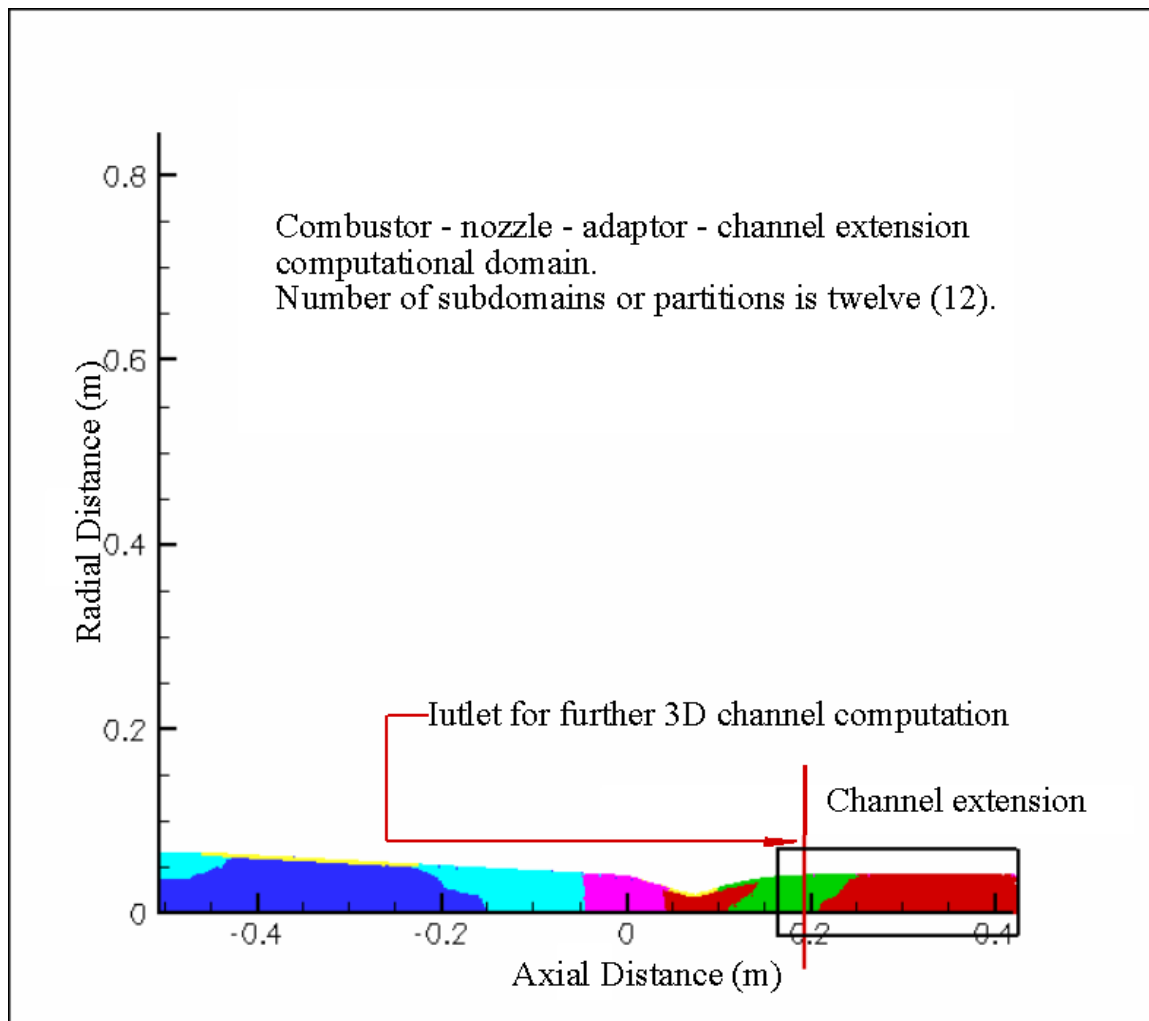
### **FIGURES ILLUSTRATING NOZZLE – COMBUSTOR GEOMETRY AND MESH**



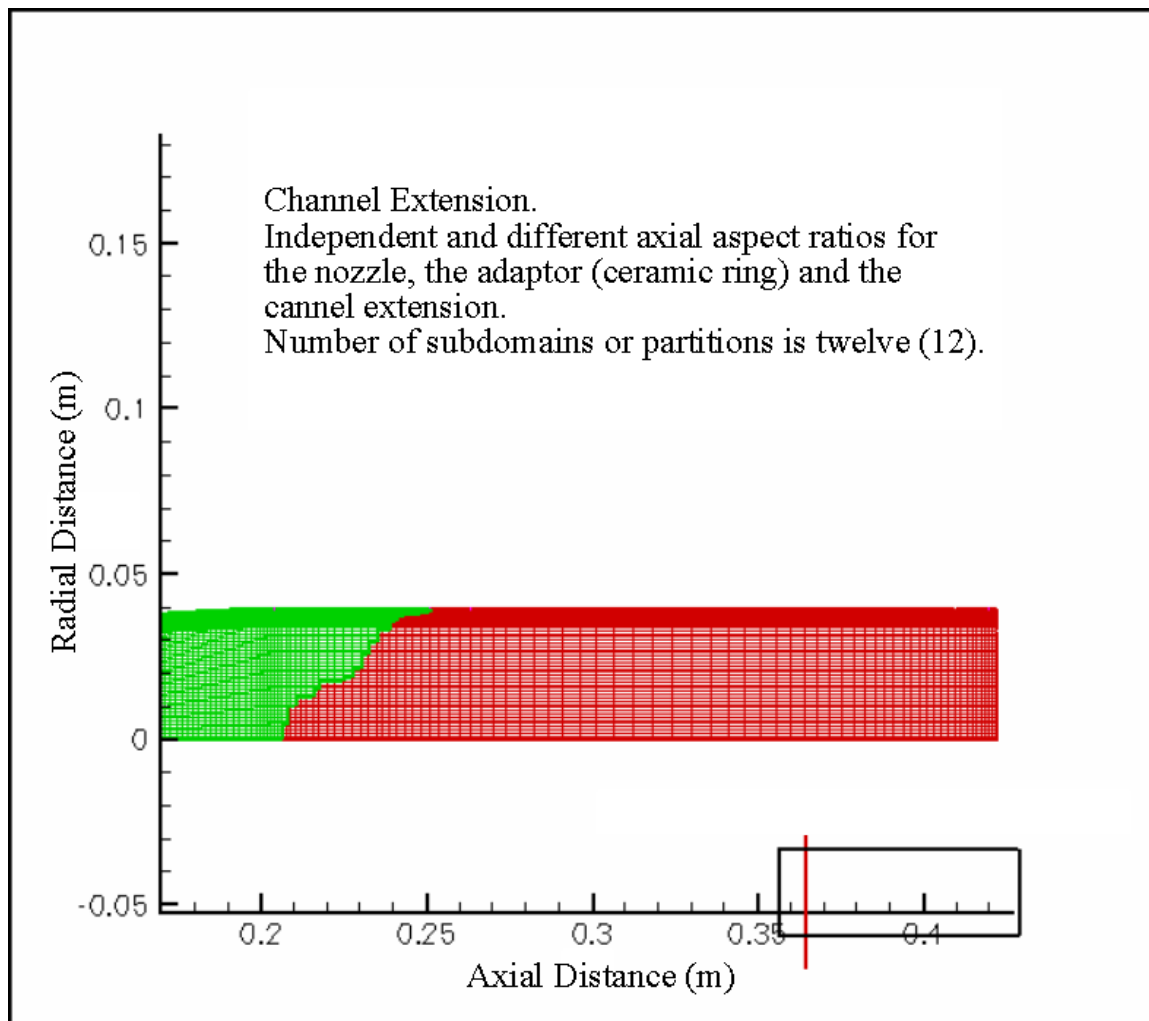
**Figure 3.9.1.** Combustor, nozzle, and nozzle extension attached to the outlet of the nozzle.



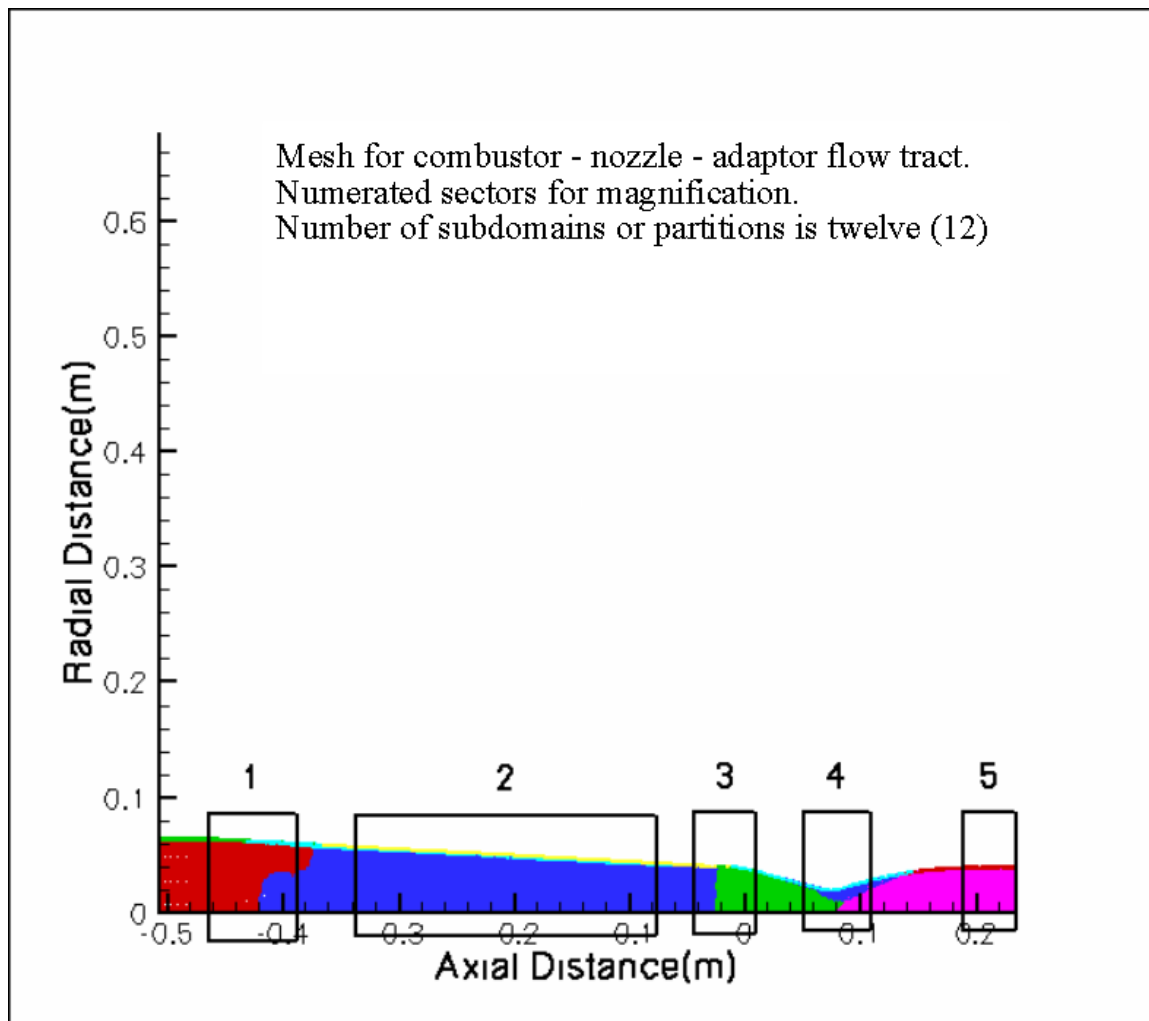
**Figure 3.9.2.** Non-Uniform and axially stretched mesh at the nozzle extension – channel interface.



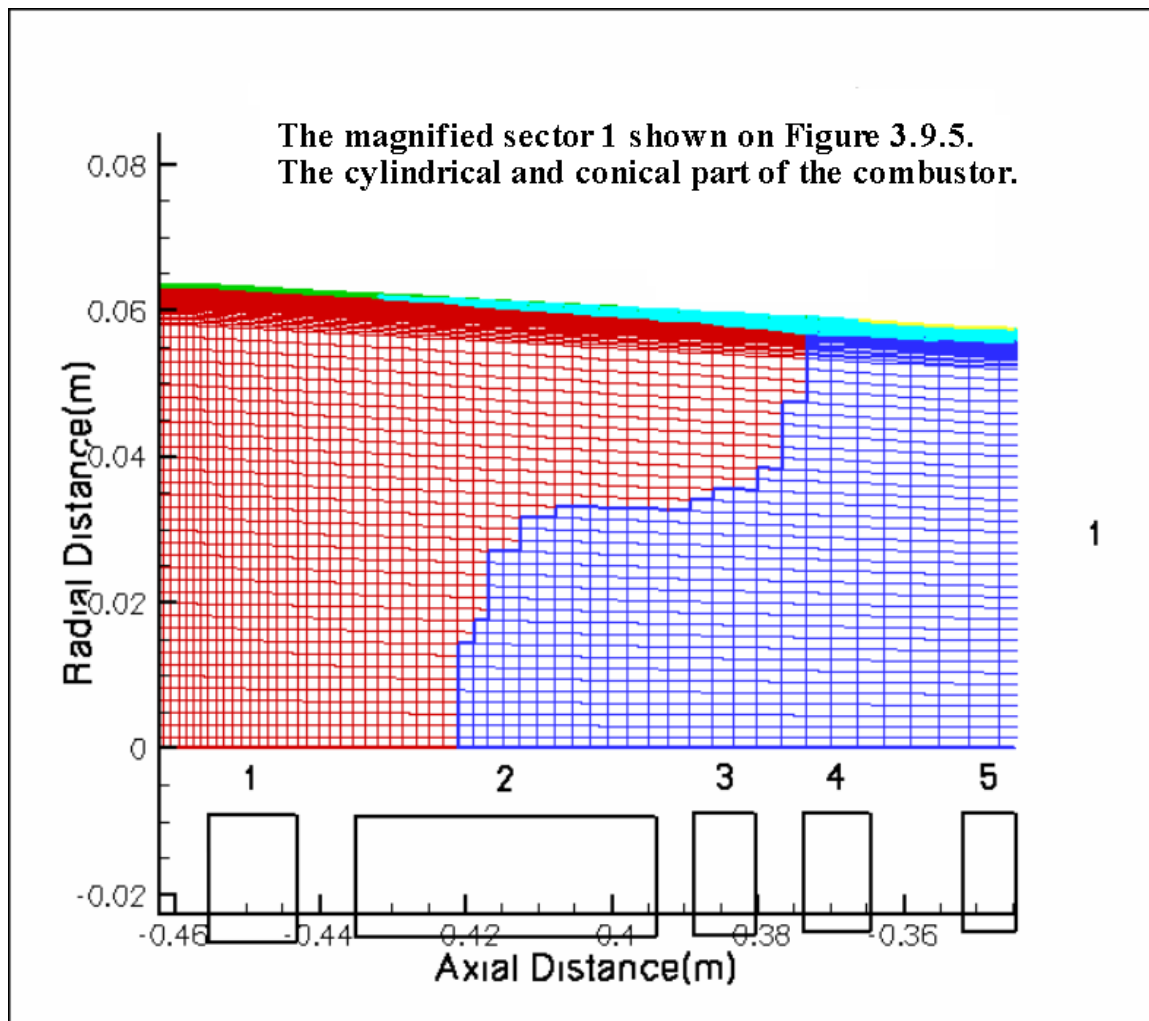
**Figure 3.9.3.** Computational sub-domains created as partitions by PGRID®, including combustor, nozzle, and straight channel components.



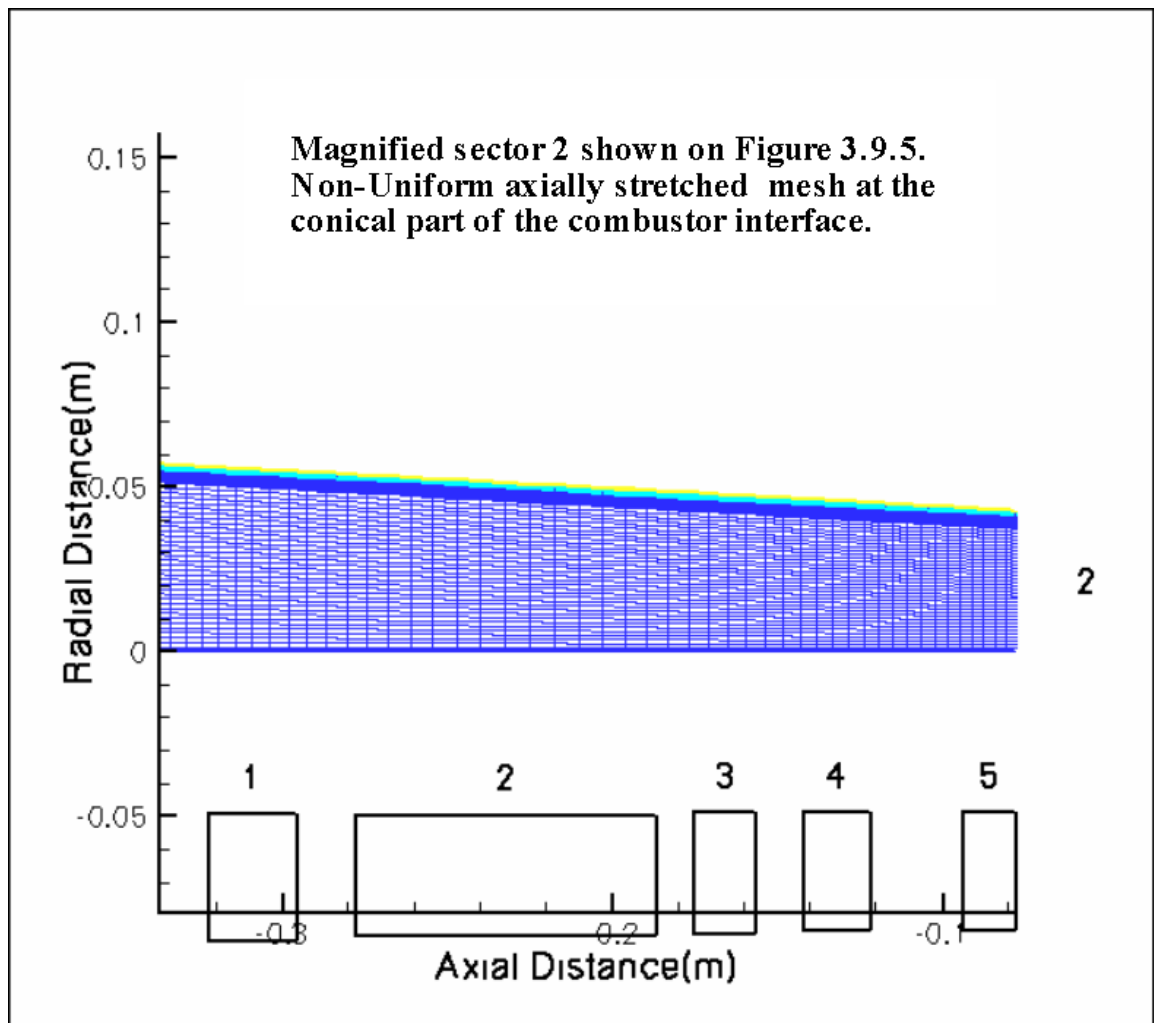
**Figure 3.9.4.** The magnified channel extension area shown on **Figure 3.9.3**.



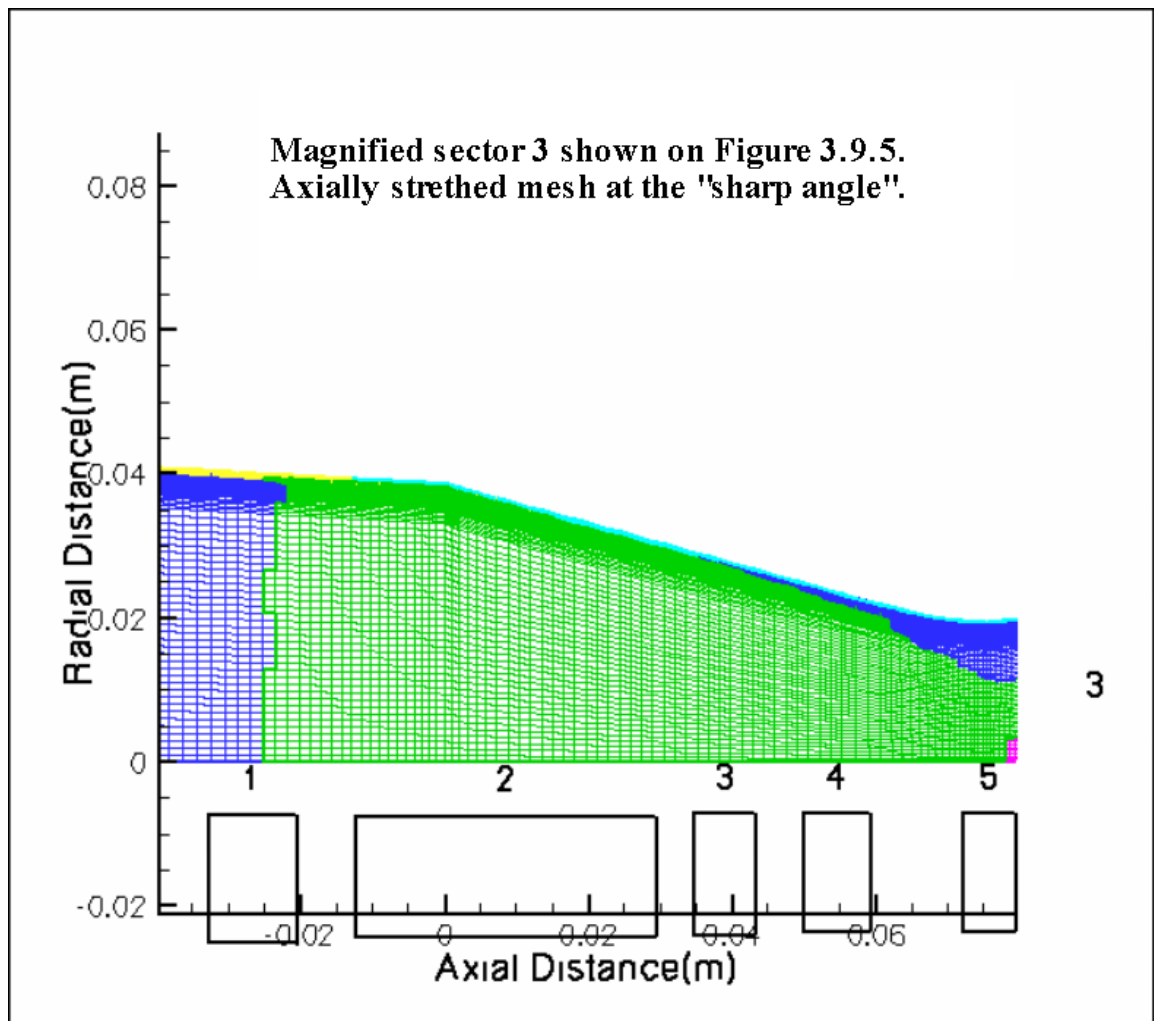
**Figure 3.9.5.** Five sectors of the multi-partition combustor - nozzle - adaptor domain requiring stretched mesh configuration.



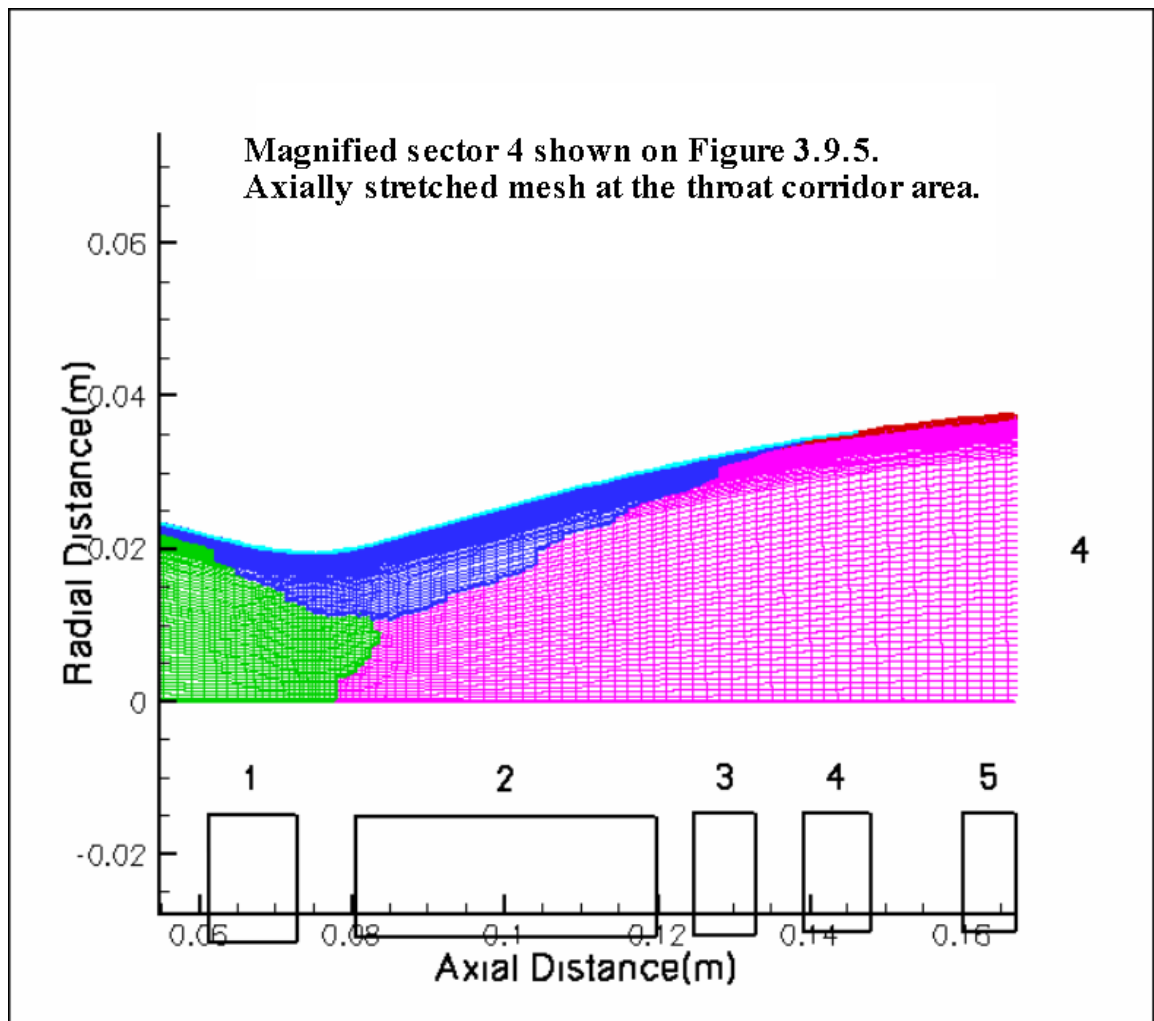
**Figure 3.9.6.** Magnified view of mesh stretching in the region where the conical section of the combustor is attached to the cylindrical section, representing sector **1** of **Figure 3.9.5**.



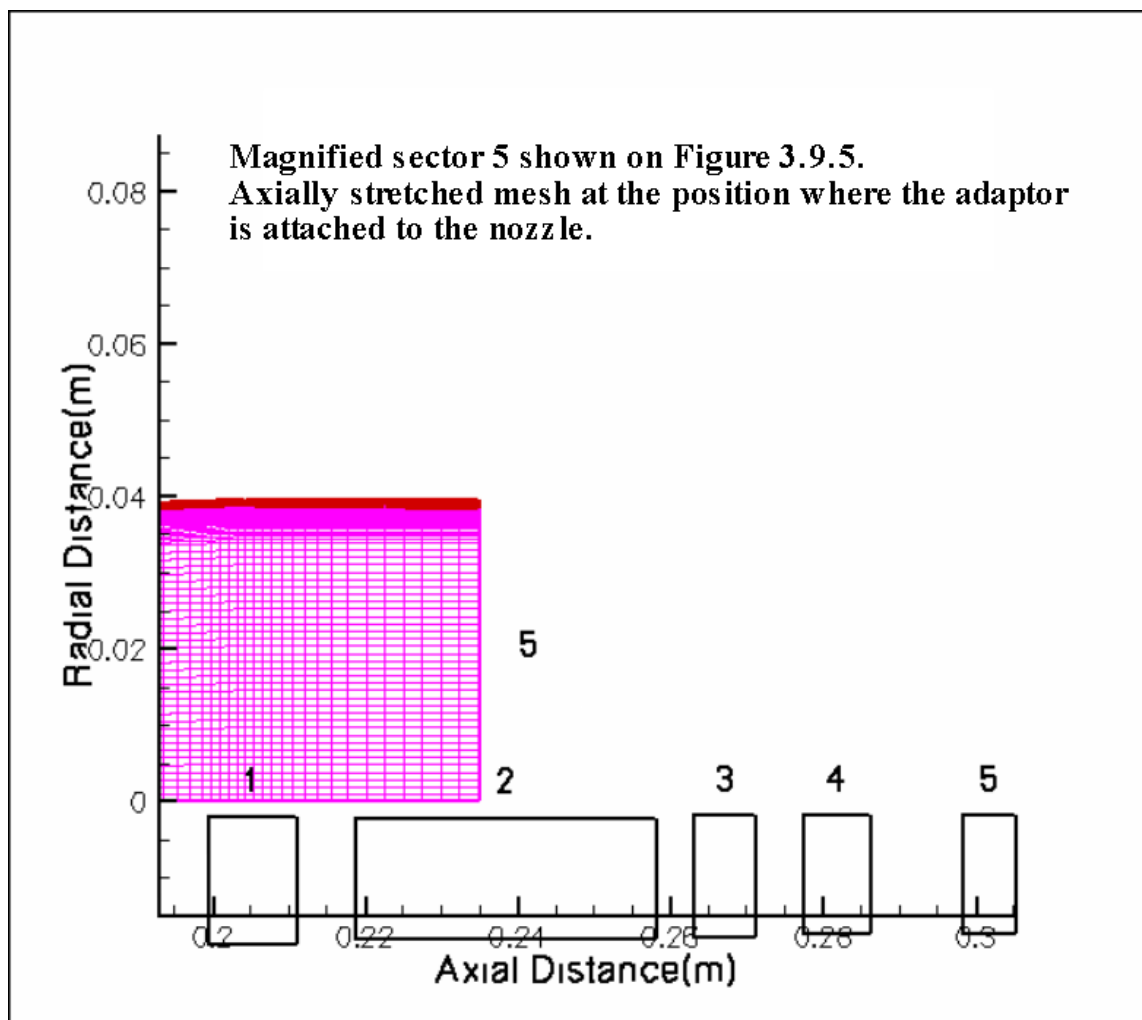
**Figure 3.9.7.** Magnified view of the axial and radial mesh stretching in the conical section of the combustor, representing sector 2 of **Figure 3.9.5**.



**Figure 3.9.8.** Magnified view of the axial and radial mesh corner stretching of the nozzle wall contour, where the inlet of the nozzle is attached to the outlet of the combustor, representing sector 3 of **Figure 3.9.5**.



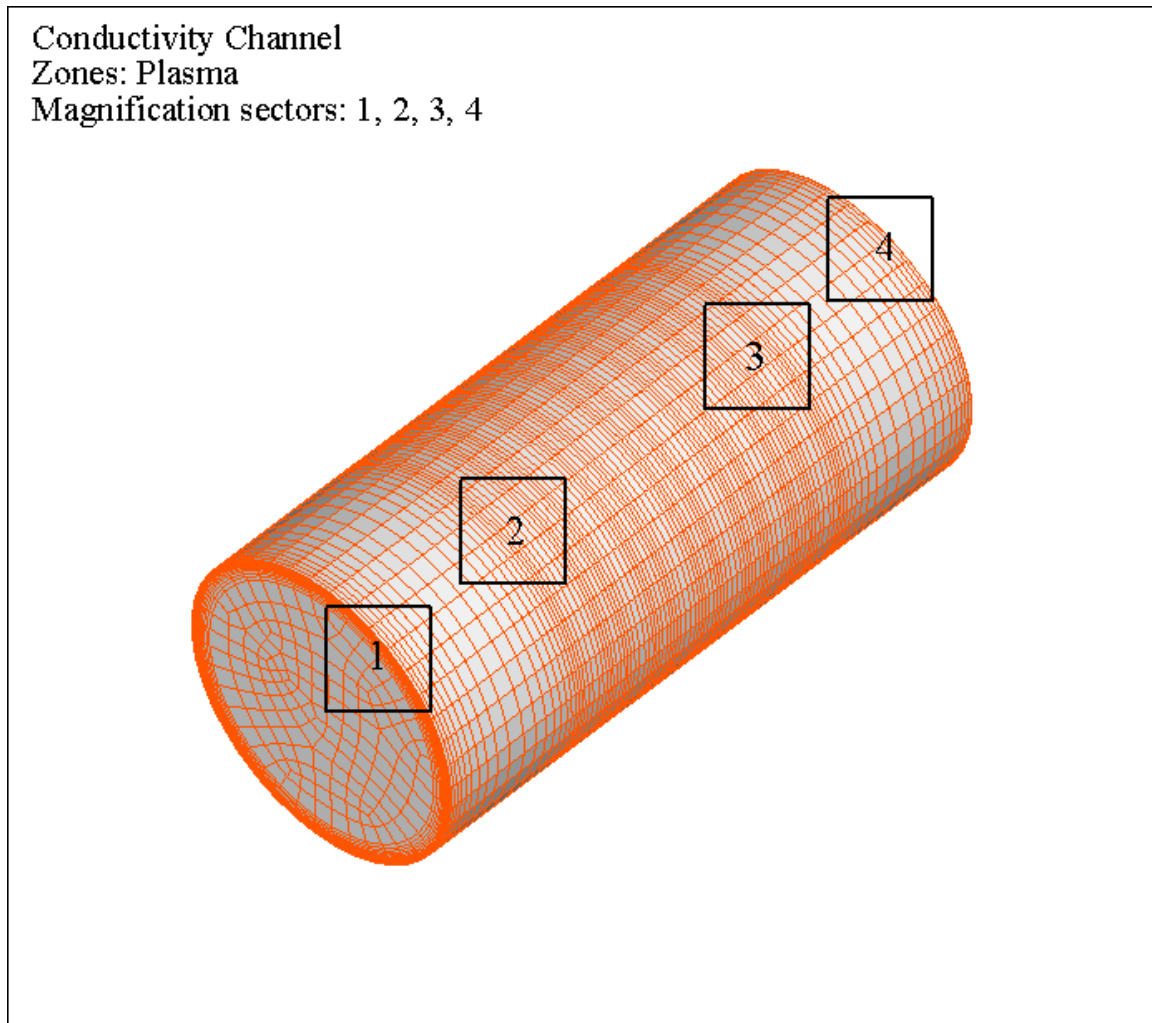
**Figure 3.9.9.** Magnified view of axial and radial mesh stretching in the throat region of the nozzle, which represents sector 4 of **Figure 3.9.5**.



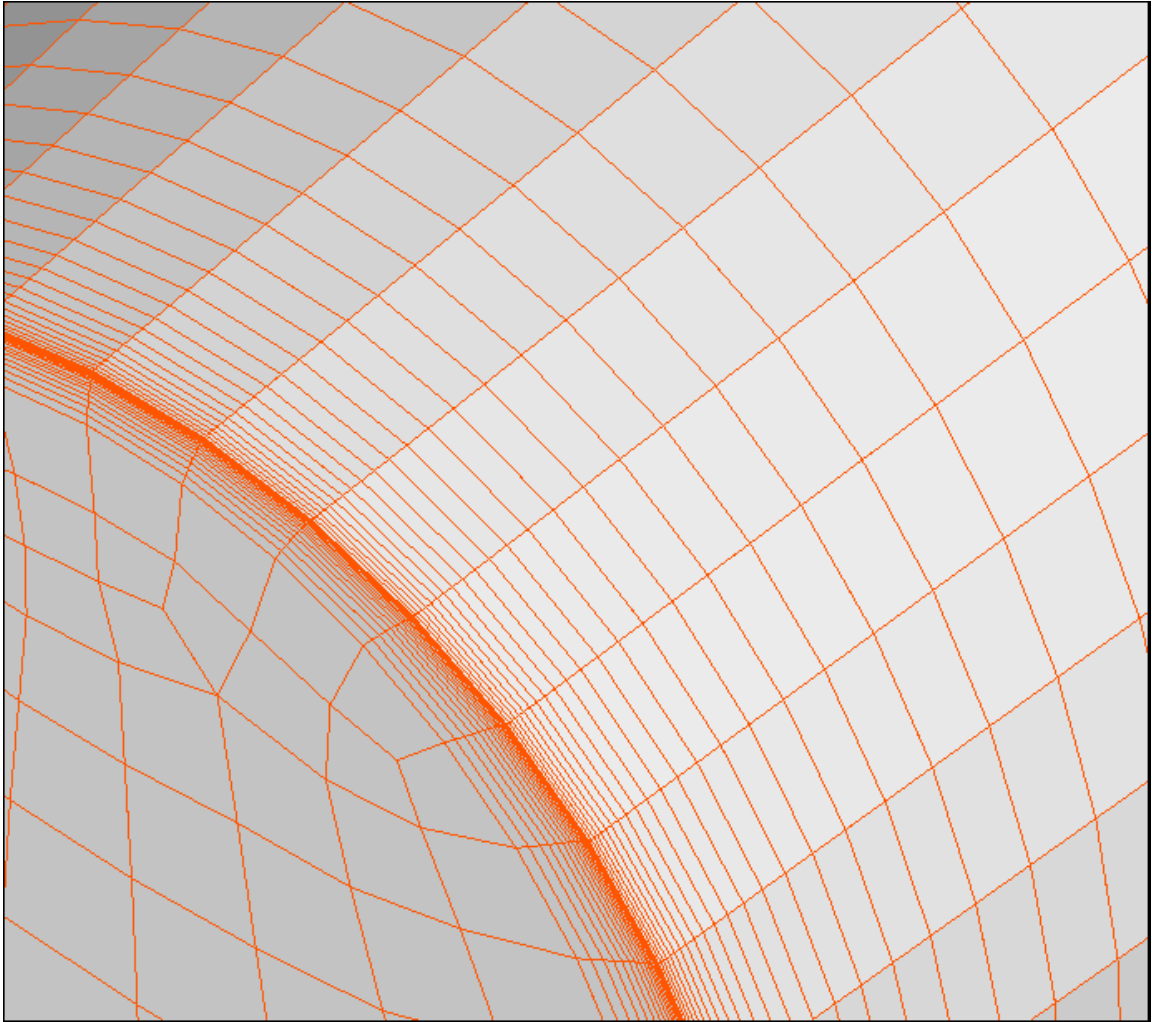
**Figure 3.9.10.** Magnified view of axial and radial mesh stretching in the nozzle adaptor or extension, representing sector **5** of **Figure 3.9.5**.

## **APPENDIX A3**

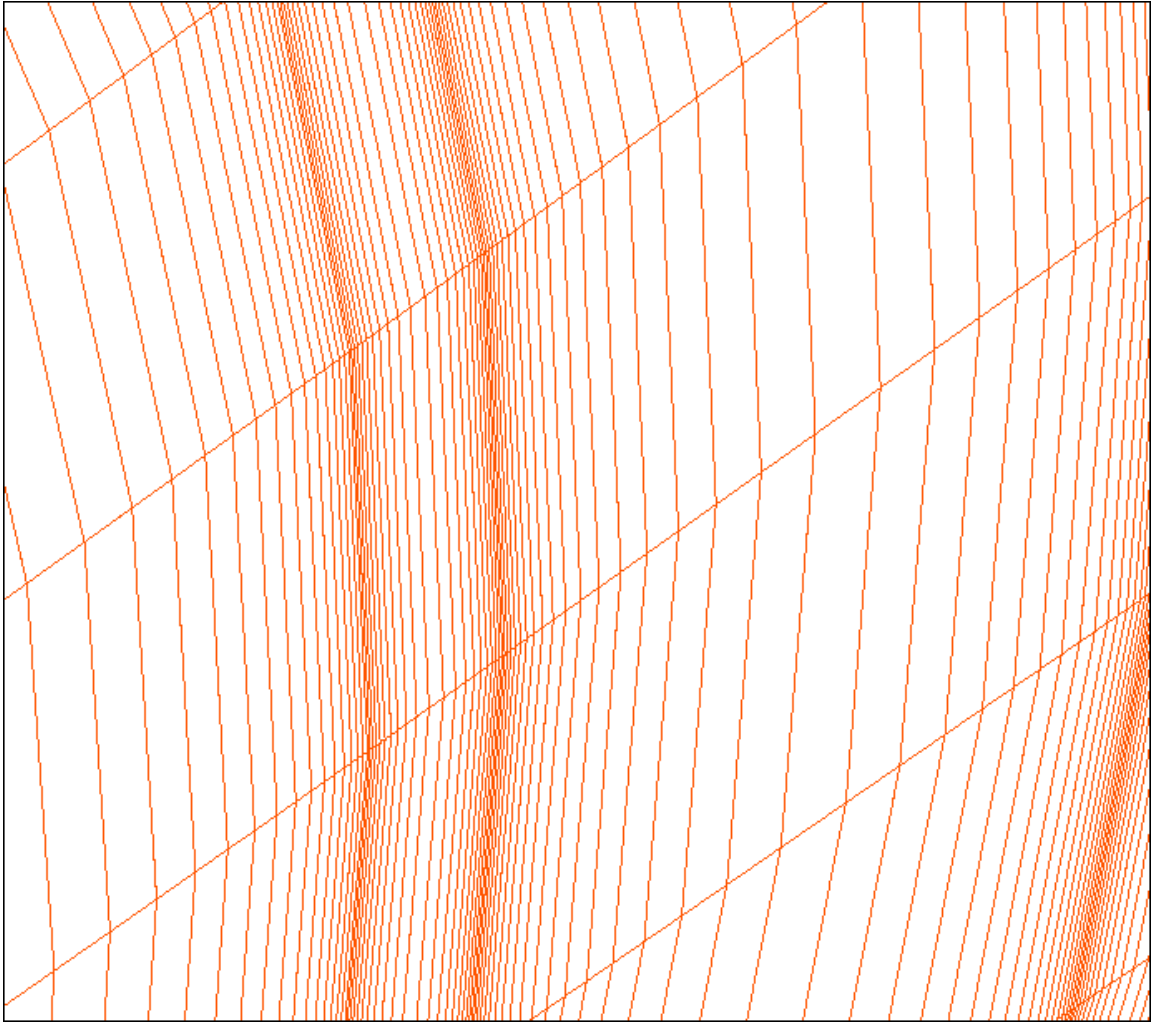
**FIGURES    ILLUSTRATING    STRAIGHT    CONDUCTIVITY    CHANNEL  
GEOMETRY AND MESH**



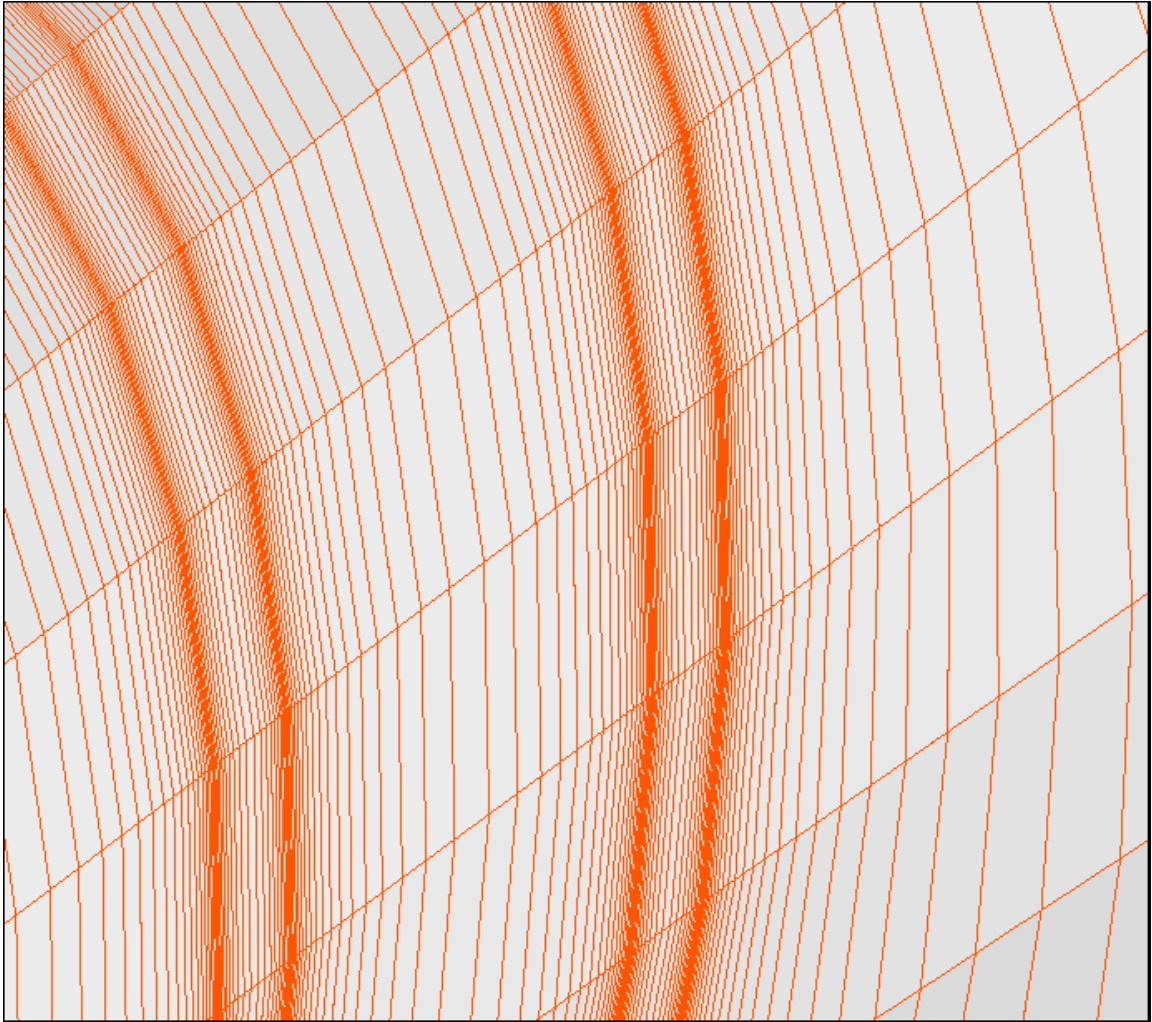
**Figure 3.9.11.** The meshed plasma domain of the straight conductivity channel with four marked sectors: the radial boundary layer (**1**); the axially stretched faces and volumes in sectors of magnification **1**, **2**, **3** and **4**.



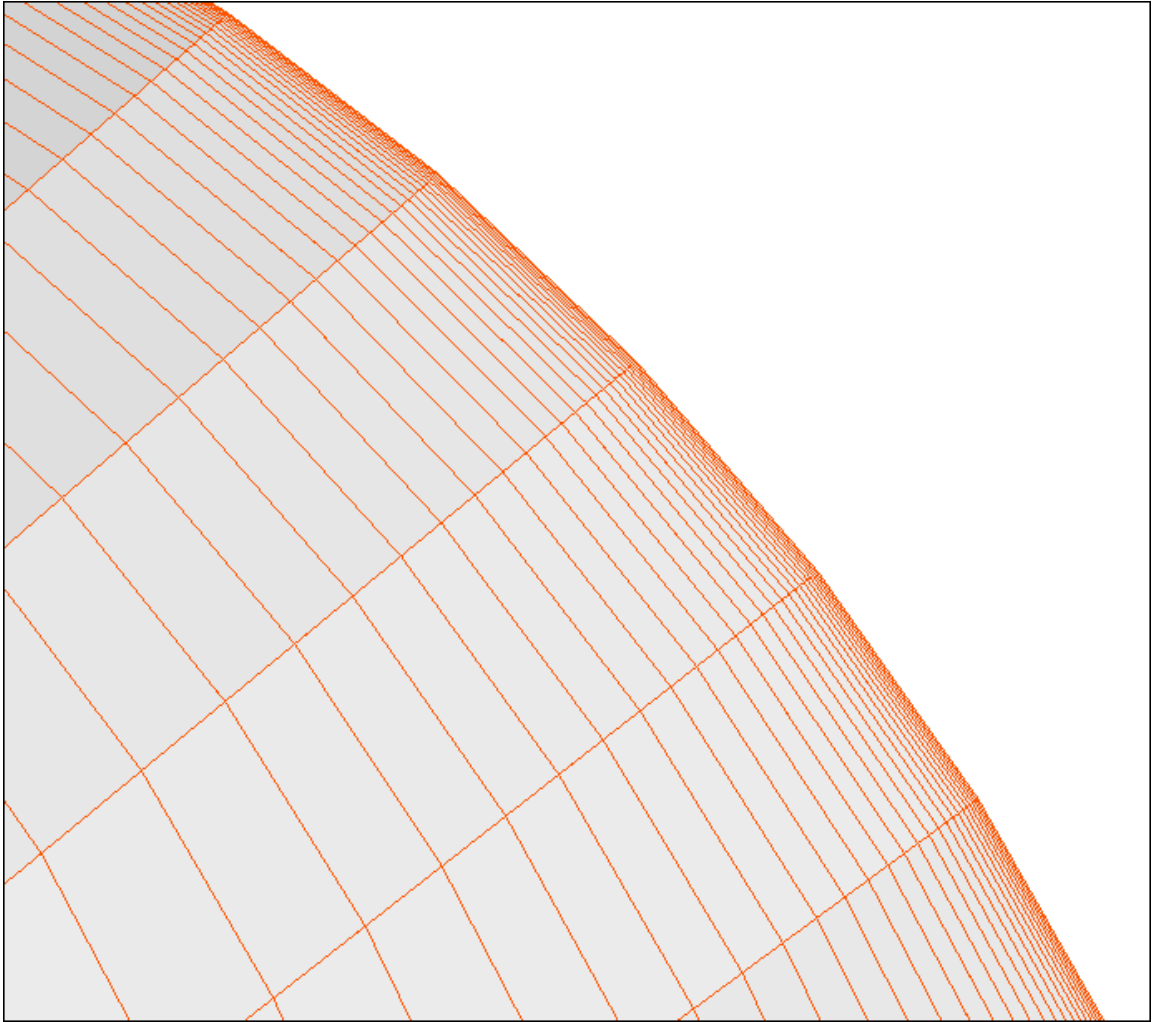
**Figure 3.9.12.** The meshed plasma domain, based on Cooper's algorithm for the interior volume of the plasma. Both an axial and radial mesh stretching distribution is shown with the radial stretching of plasma face at the inlet, and the axial stretching on the lateral surface. This figure is magnified sector **1** shown on **Figure 3.9.11**.



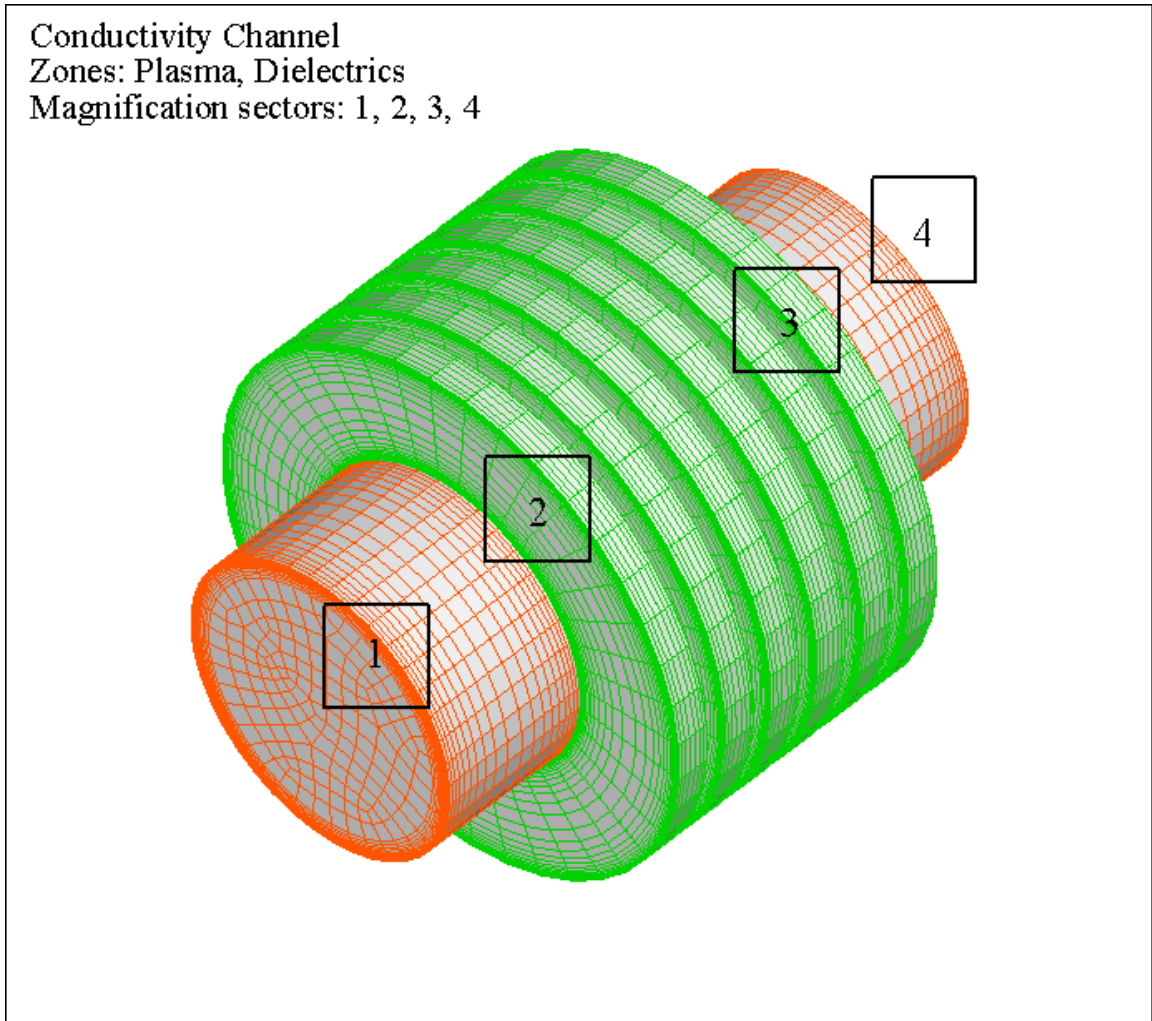
**Figure 3.9.13.** The axial stretch process for the plasma volumes adjacent to the electrode frames (conductors and insulators) on the axial plasma face, which represents the magnified sector 2 shown on **Figure 3.9.11**.



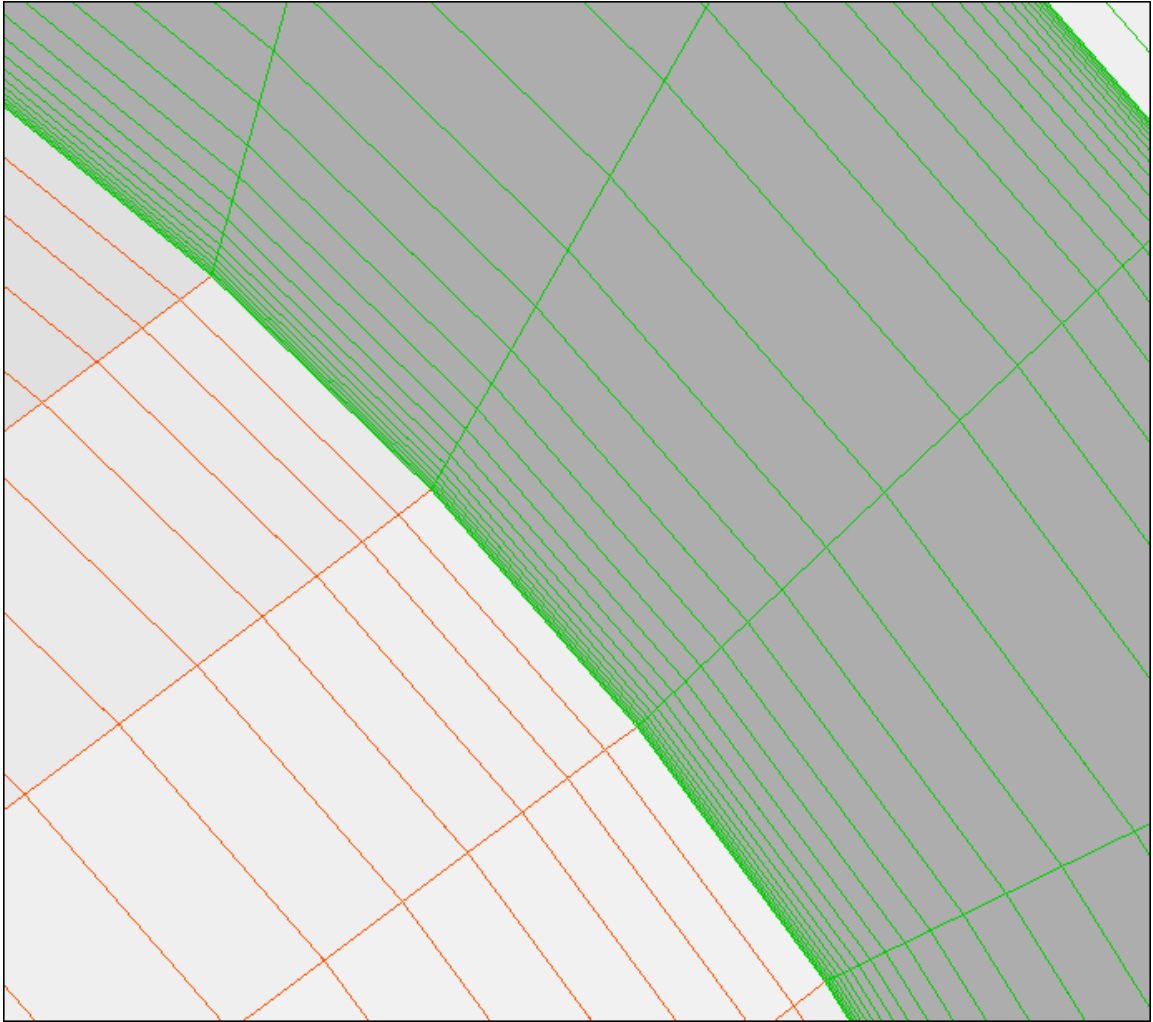
**Figure 3.9.14.** The axial stretch process for the plasma volume adjacent to additional electric elements of the channel, representing the magnified sector **3** shown on **Figure 3.9.11**, (the volume between included boundary between the last insulator and the last conductor in the channel, closest to the outlet plane).



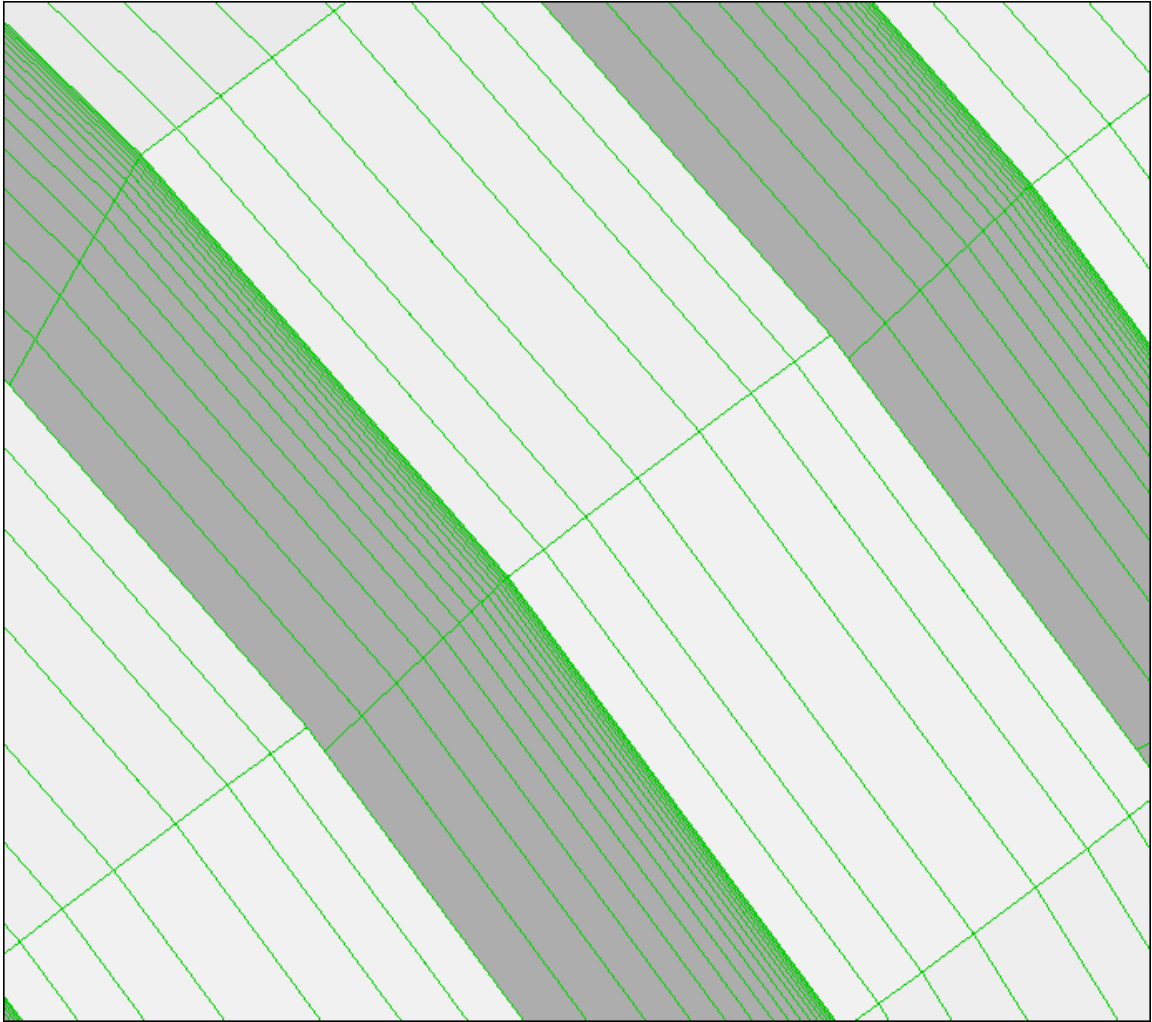
**Figure 3.9.15.** The axial stretch process on the plasma face with respect to the outlet boundary of the plasma, which represents the magnified sector **4** shown on **Figure 3.9.11**.



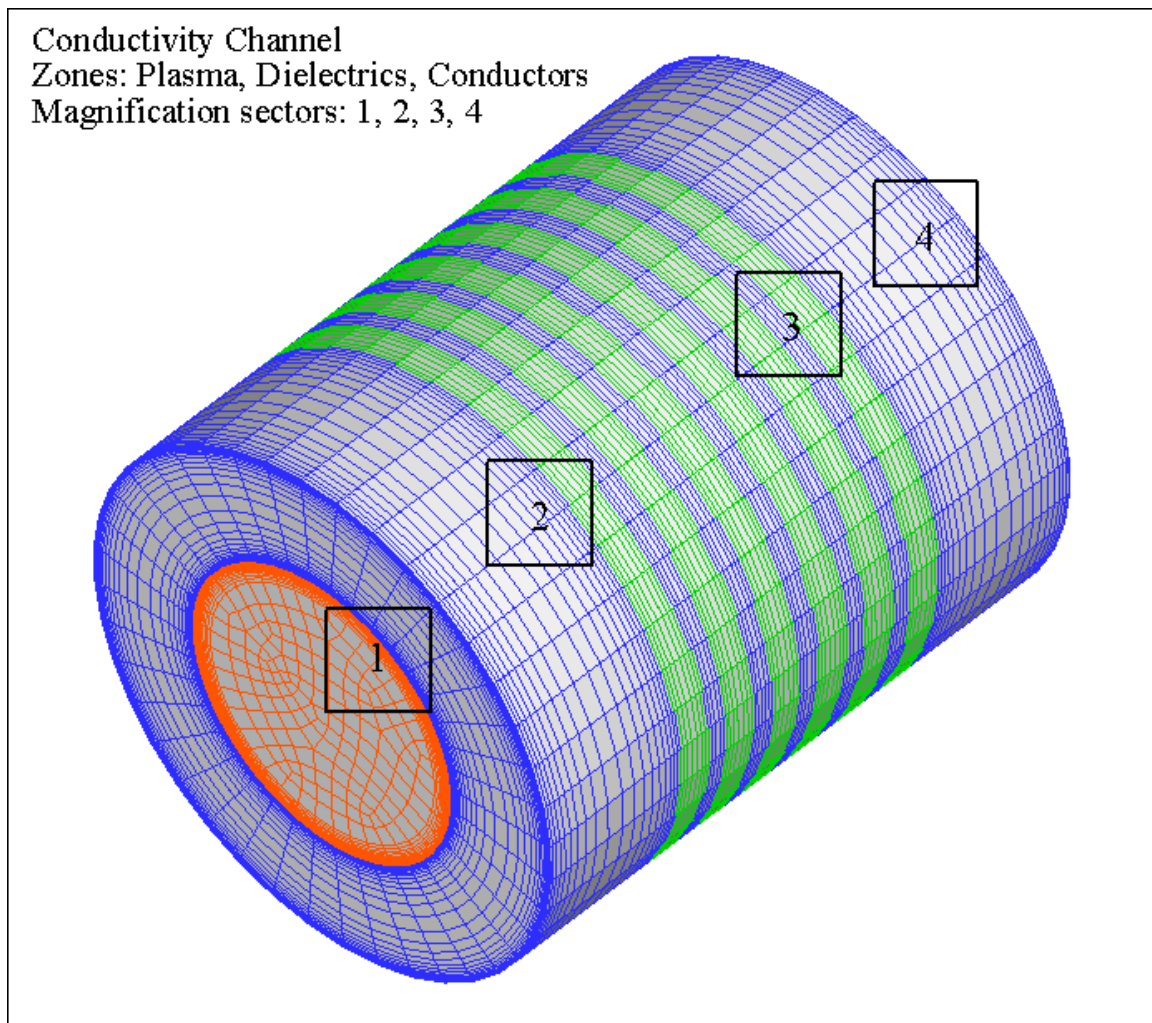
**Figure 3.9.16.** The mesh system for the sub-domain volumes representing the insulators in the straight conductivity channel; both axial and radial mesh stretching is indicated.



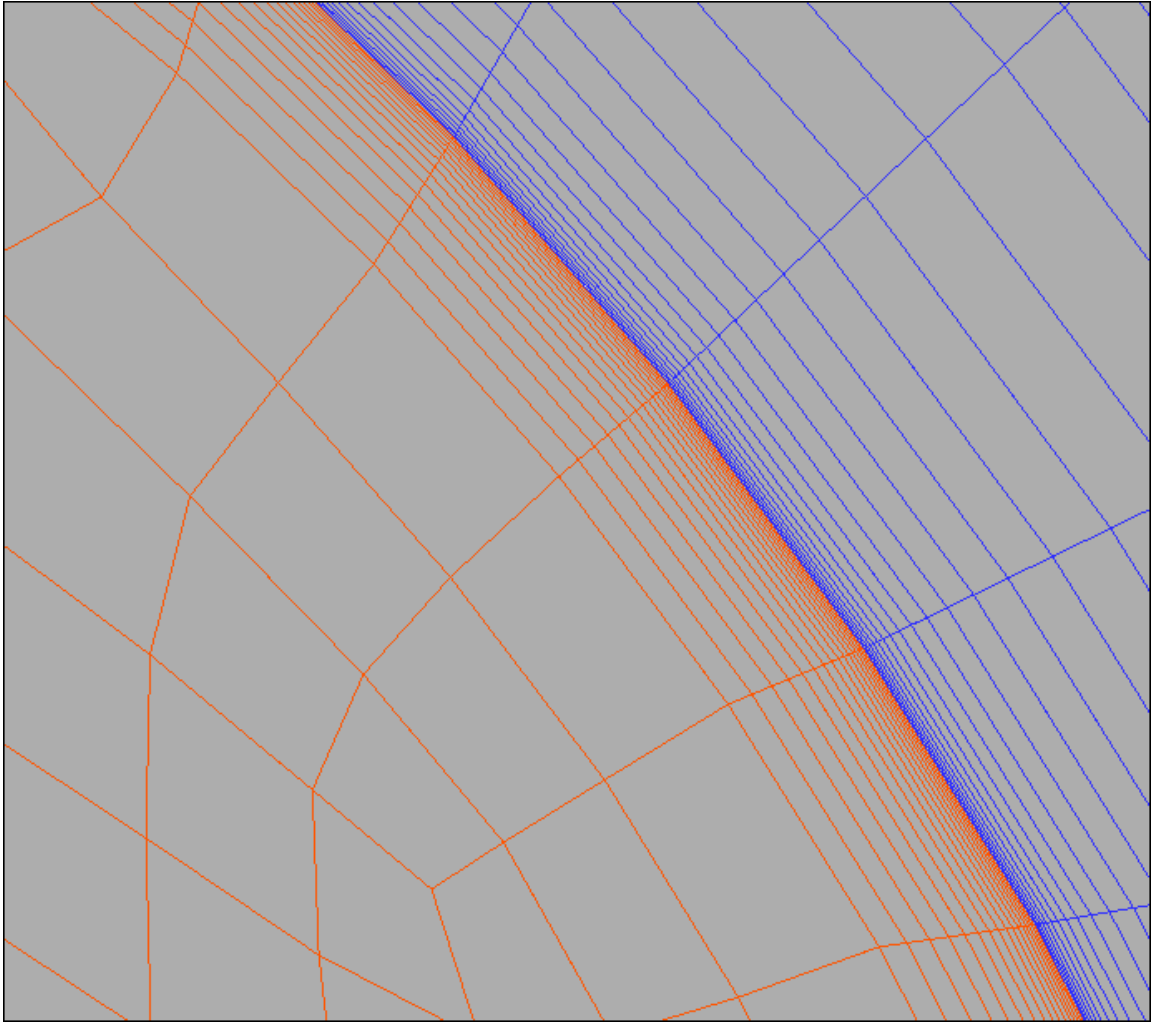
**Figure 3.9.17.** Magnified view of sector 2, **Figure 3.9.16**, showing the mesh of an axial plasma sub-domain corresponding to an axial conductor, and an insulator sub-domain with the radial mesh stretching, based on different and independent aspect ratios in the radial and axial directions.



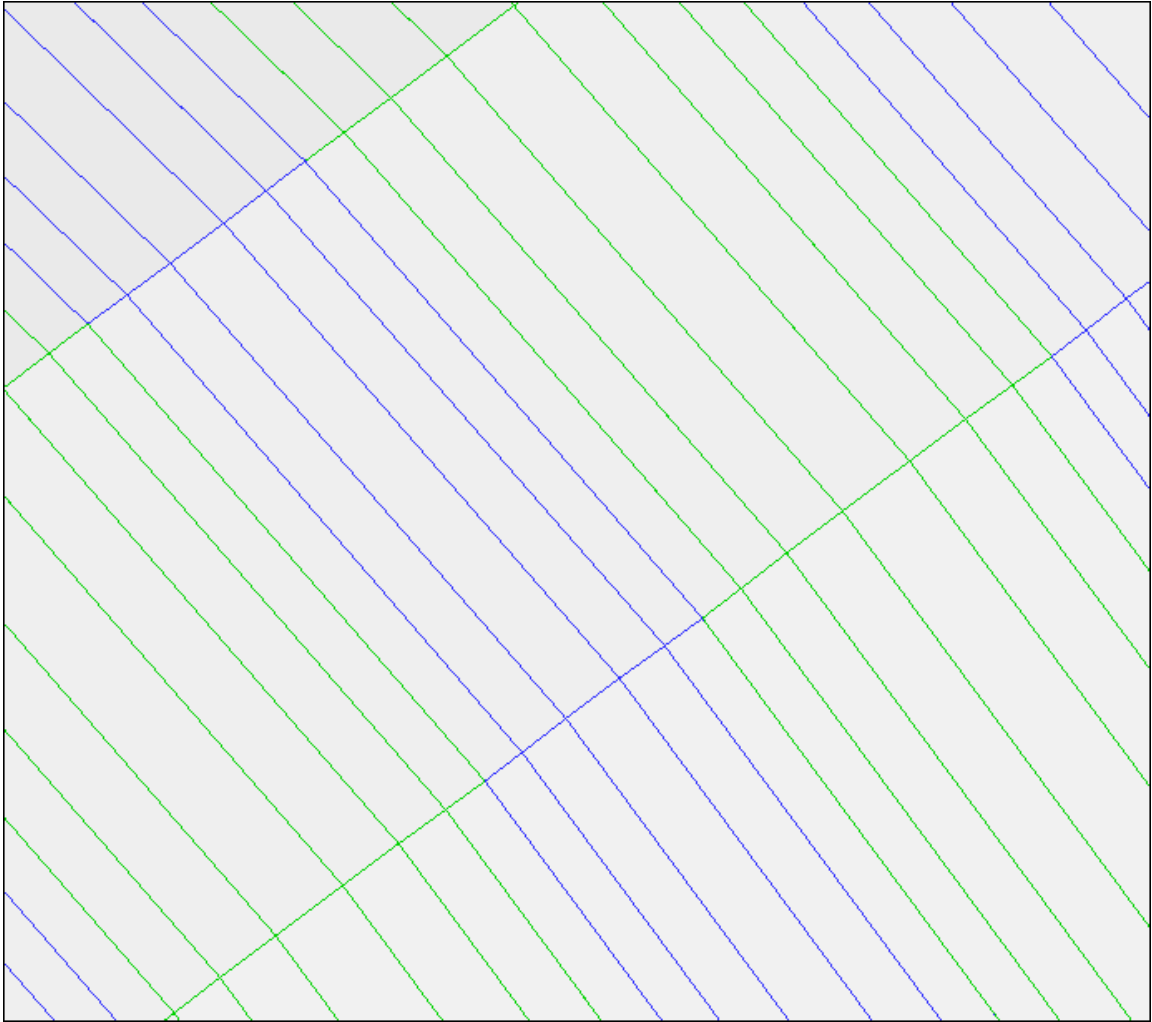
**Figure 3.9.18.** Magnified view of sector 3, **Figure 3.9.16**, showing the axial and radial mesh of adjacent insulator sub-domains, where the stretching is based on different and independent stretching aspect ratios in the radial and axial directions.



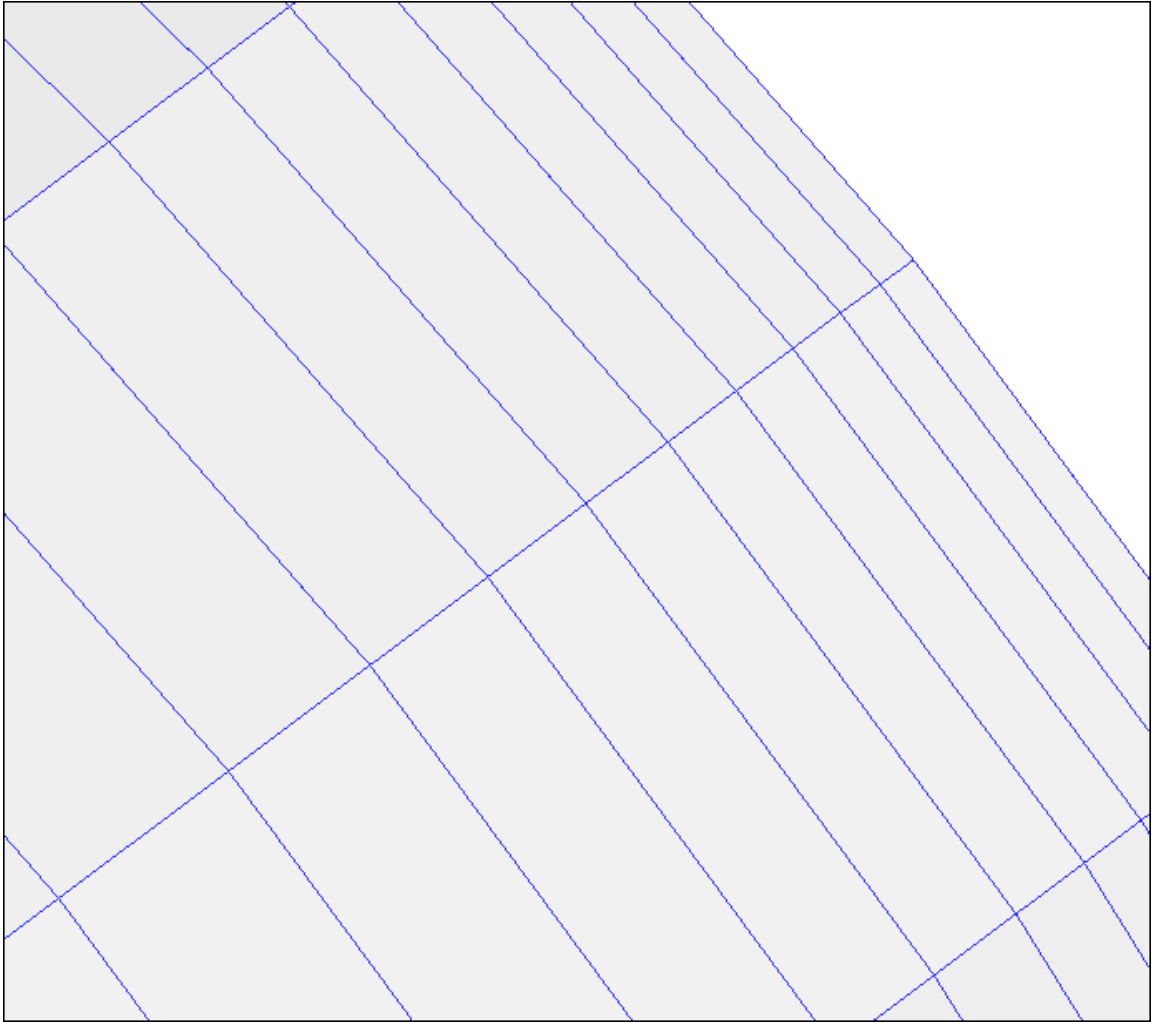
**Figure 3.9.19.** Mesh configuration for the central plasma flow domain, and the surrounding conductors and insulators forming the constant area conductivity channel.



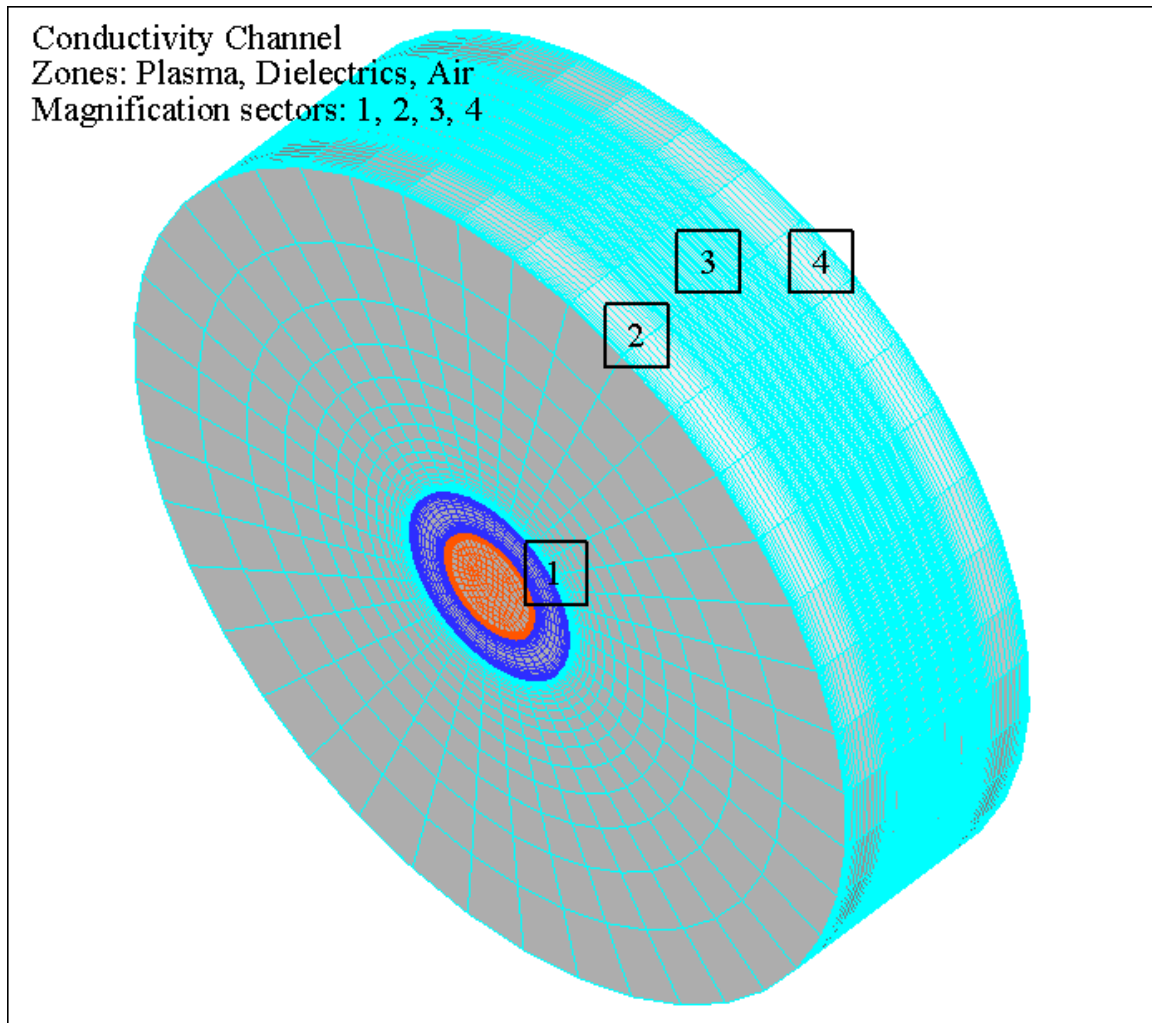
**Figure 3.9.20.** Magnification of sector 1, **Figure 3.9.19**, showing boundary layer regime in the cross-section of the plasma flow, and the solid surrounding conductor wall, obtained with equal and opposite, but independent radial aspect ratios where the meshes interface.



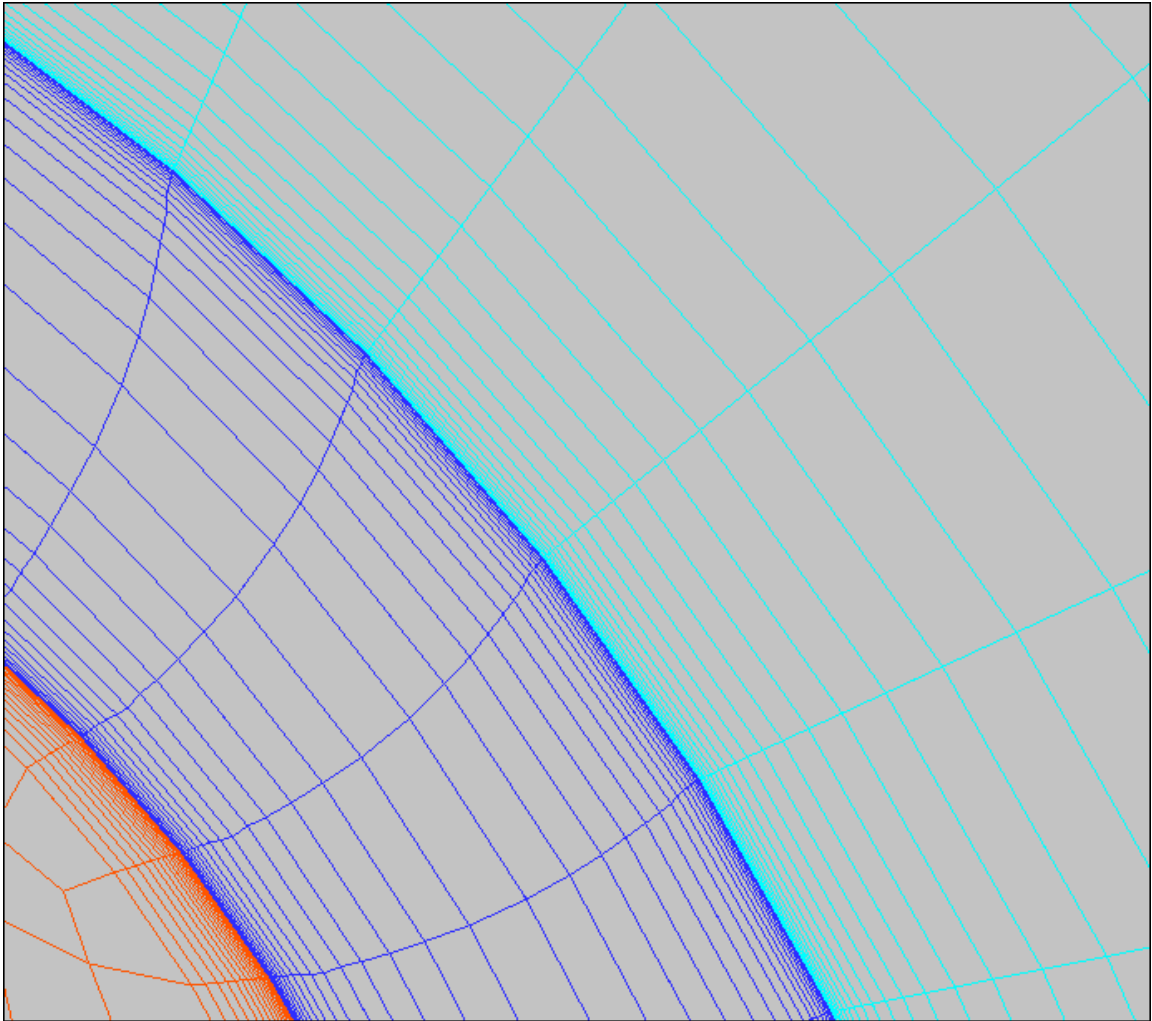
**Figure 3.9.21.** Magnification of sector **3** (sector **2** has similar view), **Figure 3.9.19**, showing the axial mesh representing the conductor and insulator segments of the constant cross-sectional area conductivity channel.



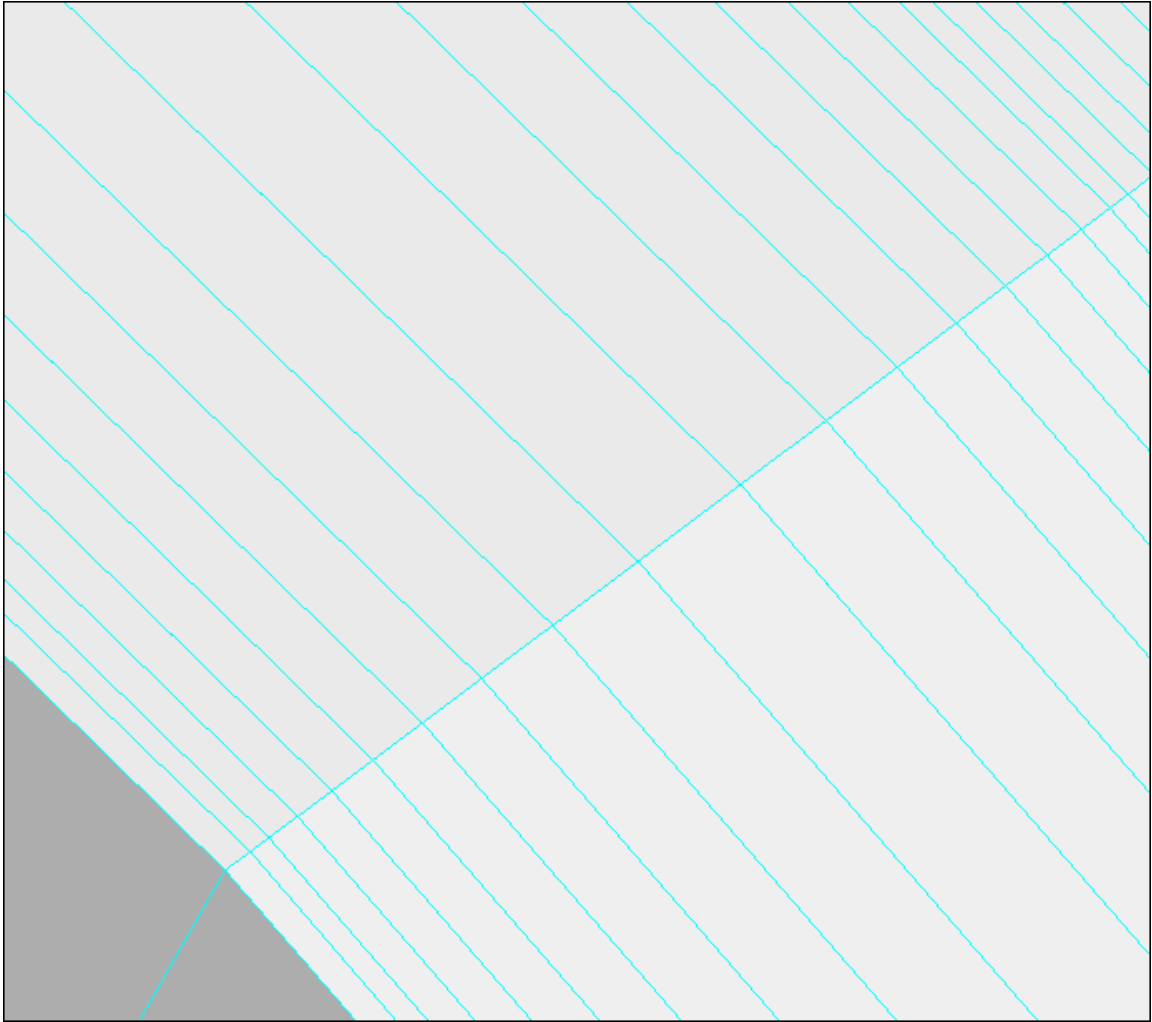
**Figure3.9.22.** Magnification of sector 4, **Figure 3.9.19**, showing the axial mesh of the last (outflow) conductor segment of the constant area conductivity channel.



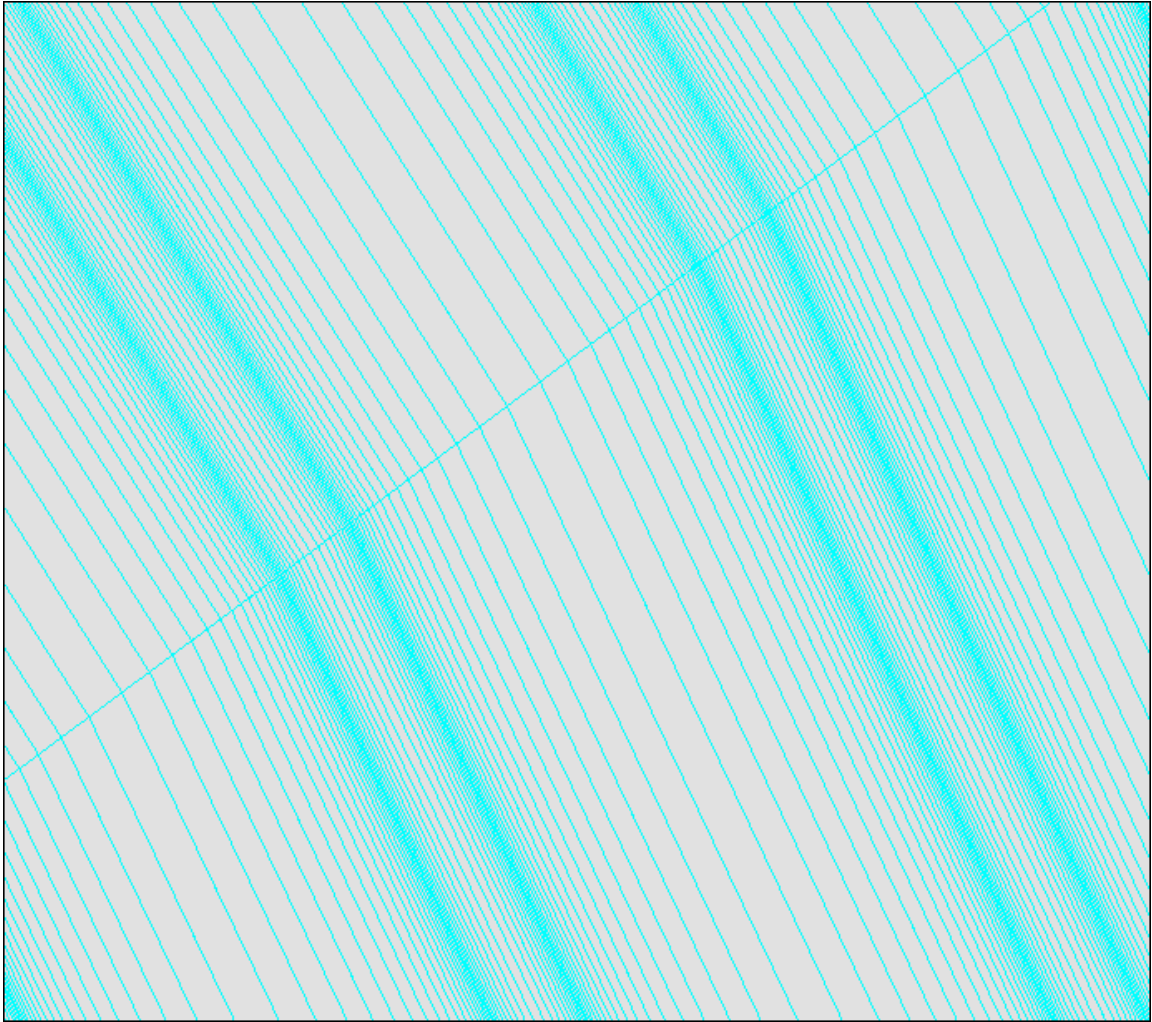
**Figure 3.9.23.** Mesh representation of the air domain surrounding the constant area conductivity channel, showing the radial and axial mesh stretching needed to interface the air domain with the conductivity channel segments.



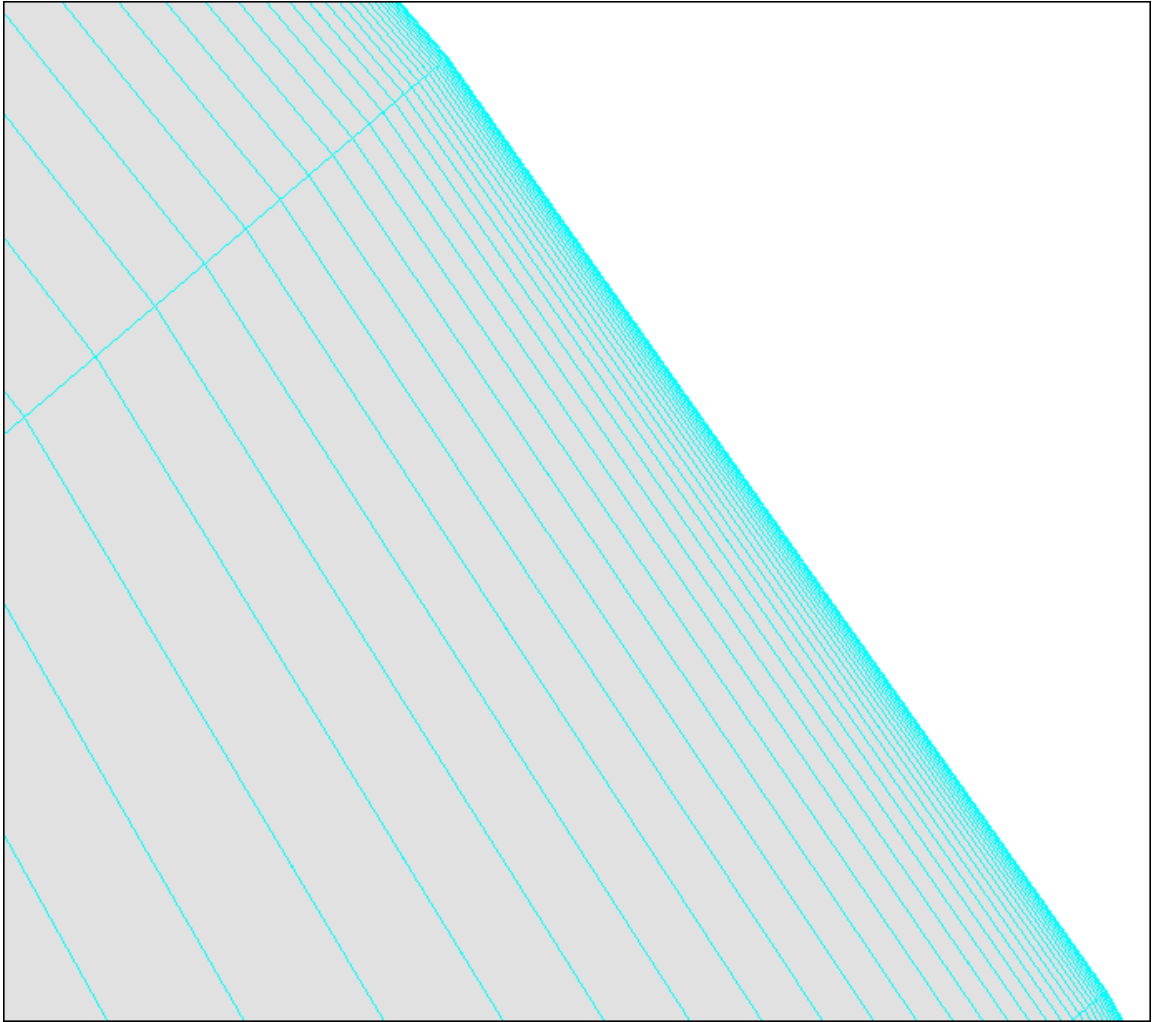
**Figure 3.5.24.** Magnification of sector **1**, **Figure 3.9.23**, showing the radial mesh representing plasma flow, solid conductor, and surrounding air domain, generated with three equal and opposite, but independent aspect ratios.



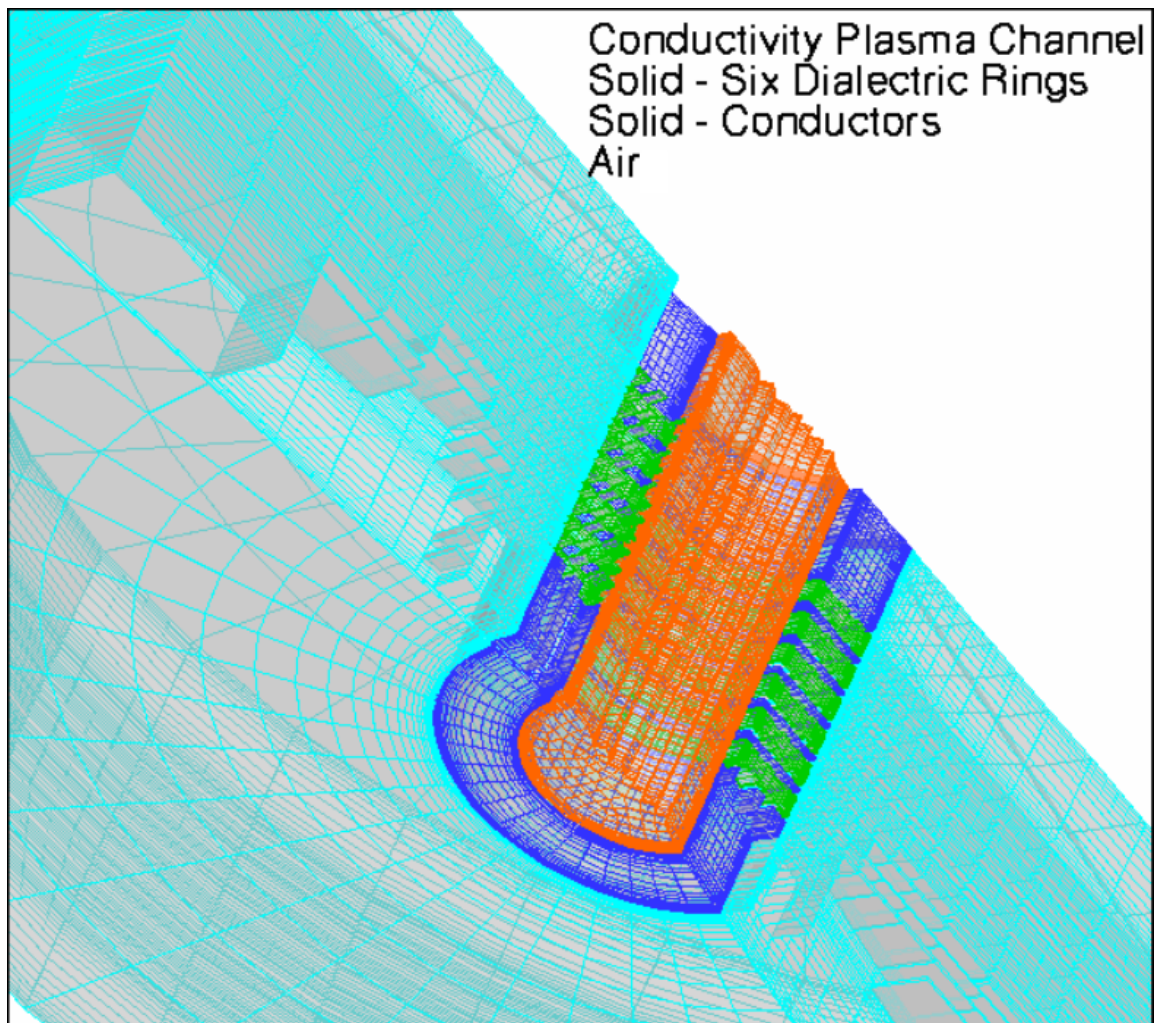
**Figure 3.9.25.** Magnification of sector **2**, **Figure 3.9.23**, showing the radial mesh stretching on the front face of the inlet plane in the air domain, and the axial mesh stretching on the other surface of the air domain.



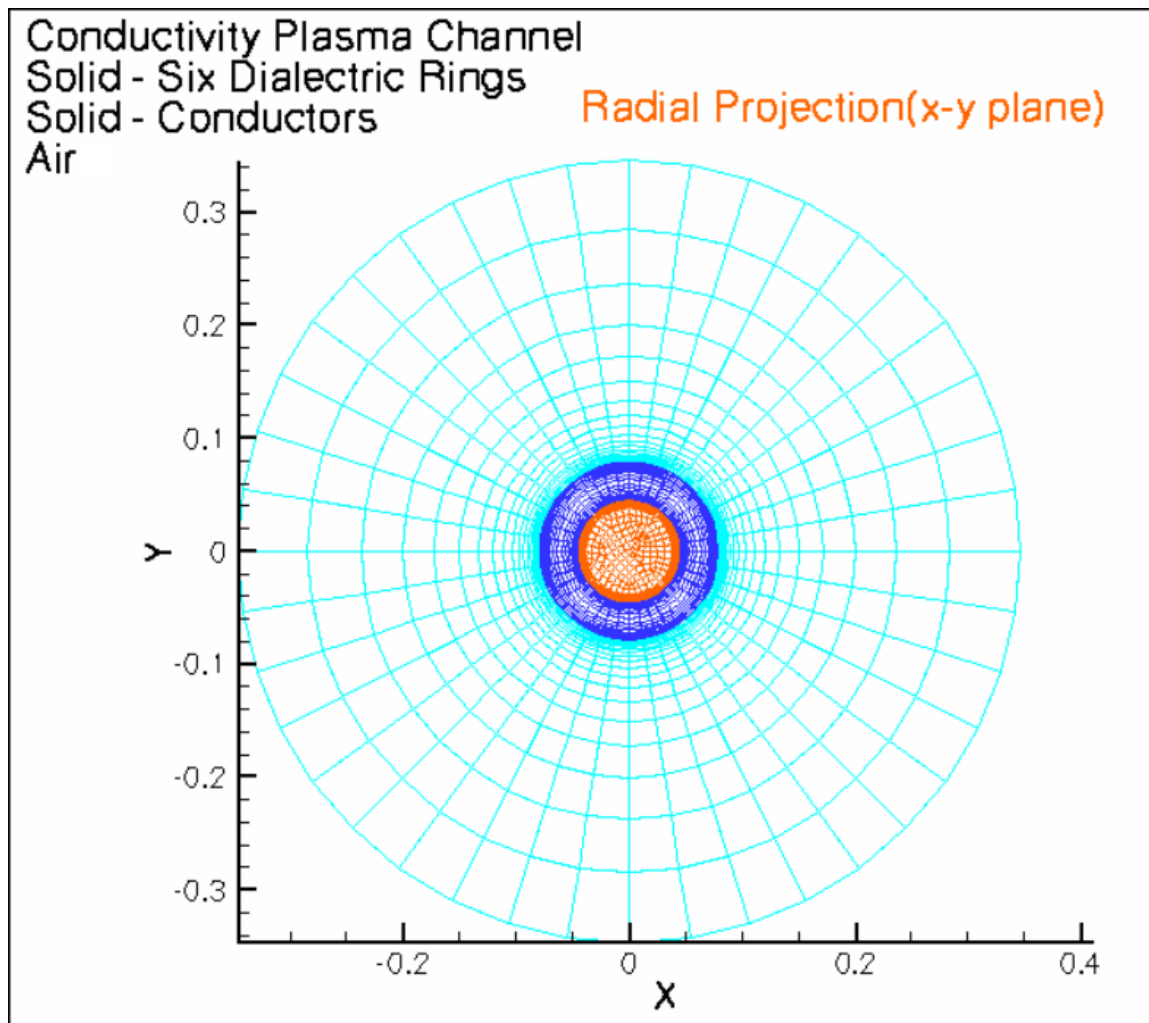
**Figure 3.9.26.** Magnification of sector 3, **Figure 3.9.23**, showing that the axial mesh stretching and grid generation scheme matches with the corresponding solid elements of the constant area conductivity channel, the conductors and insulators.



**Figure 3.9.27.** Magnification of sector 4, **Figure 3.9.23**, showing the axial mesh configuration of the last segment of the air domain.



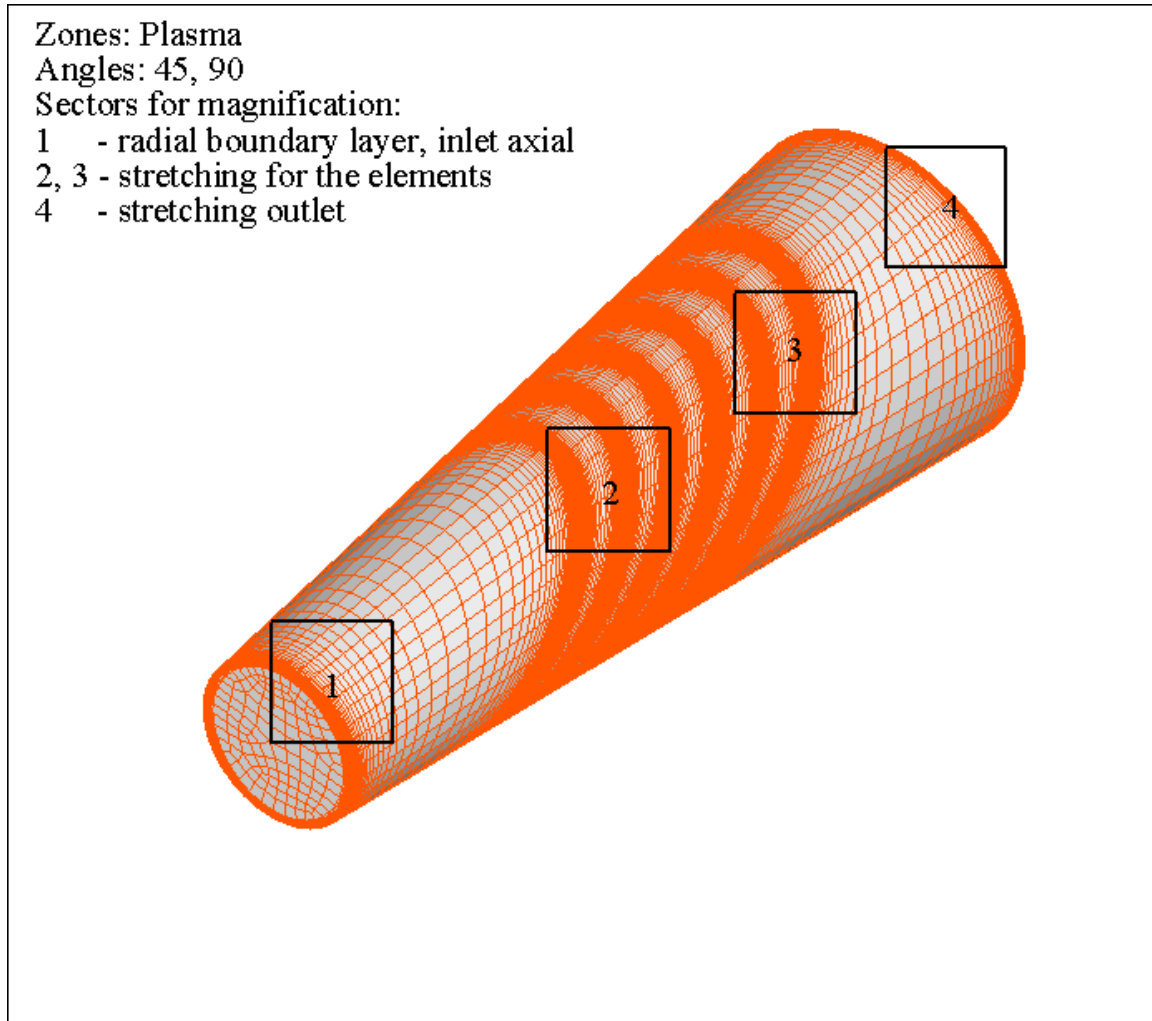
**Figure 3.9.28.** A finite element representation of the entire meshed domain of the constant area conductivity channel made with TECPLOT™ for the purpose of illustrating the complexity of the meshing process.



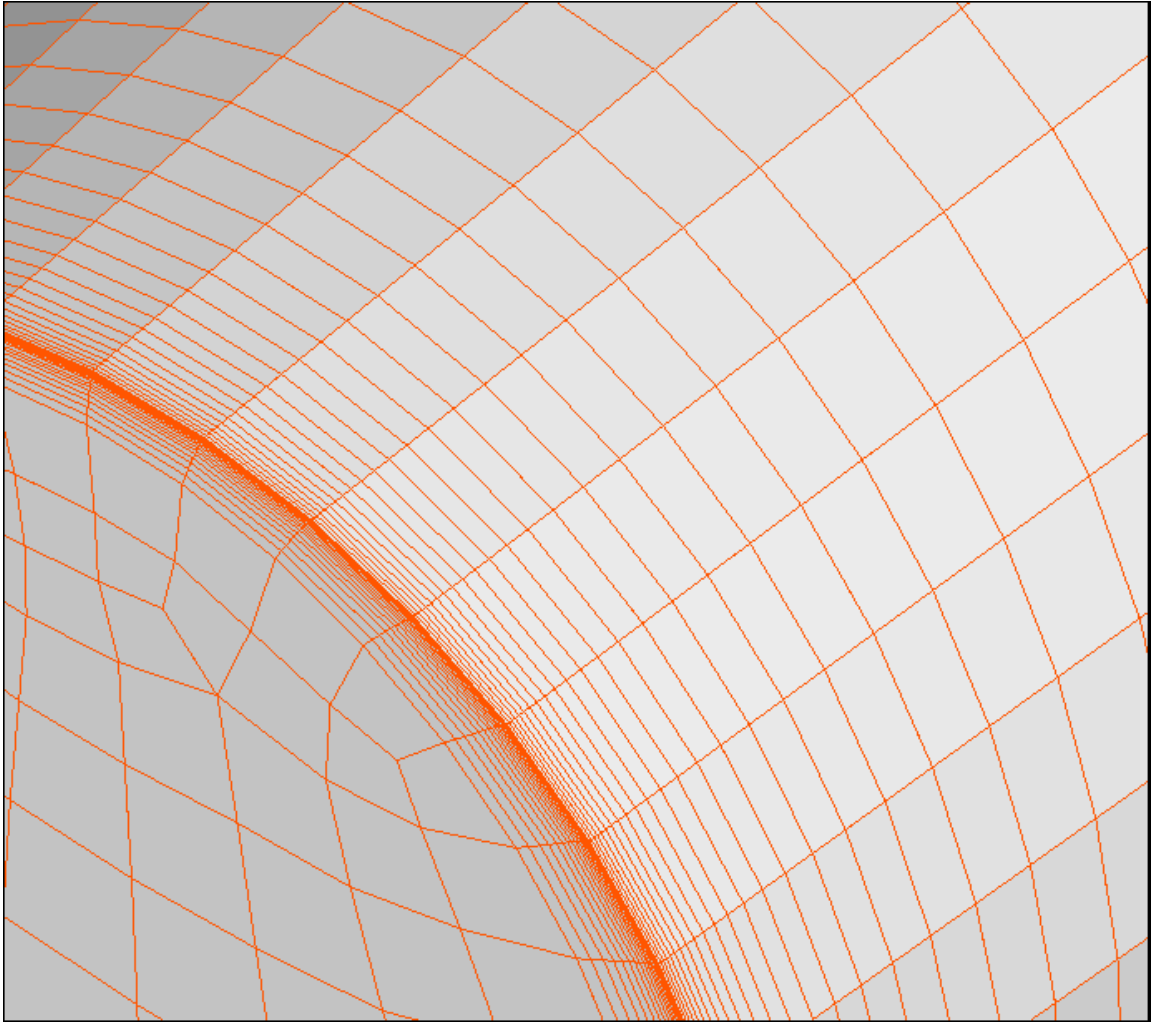
**Figure 3.9.29.** A radial projection of the entire meshed domains representing the inlet plane, showing plasma, solid and air volumes for the straight conductivity channel.

## **APPENDIX A4**

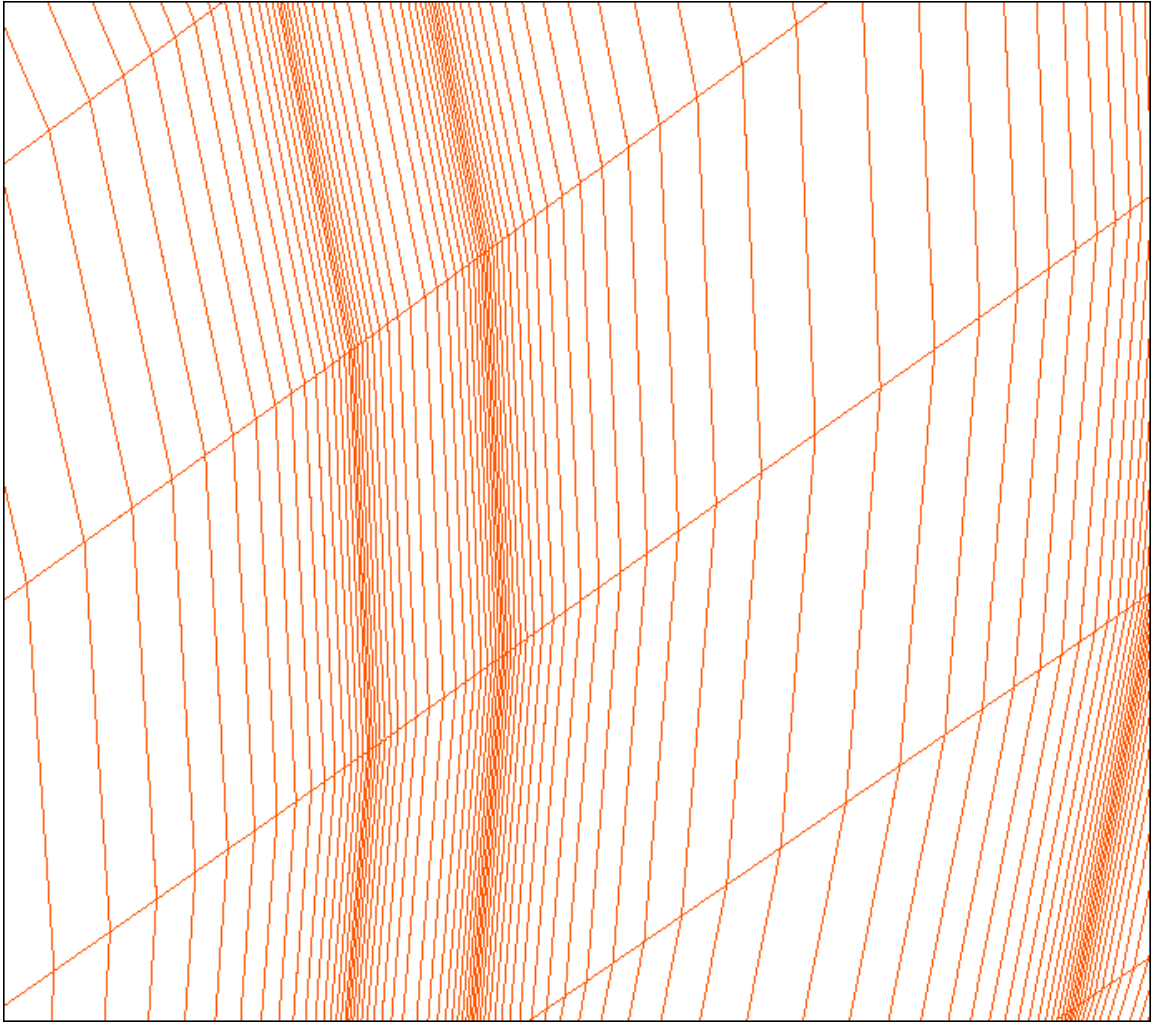
### **FIGURES ILLUSTRATING DCW MHD CHANNEL GEOMETRY AND MESH**



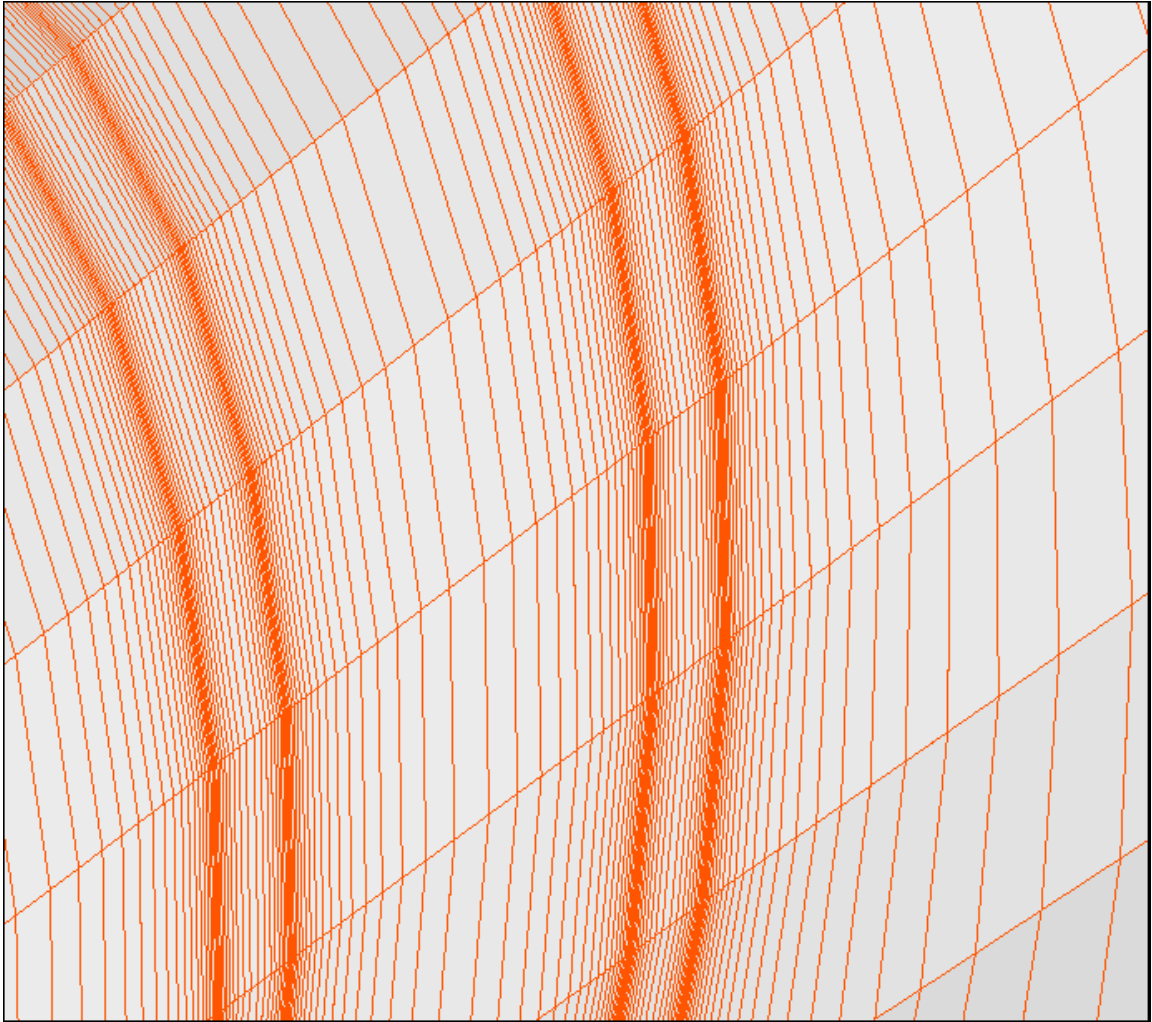
**Figure 3.9.30.** The meshed domains of the plasma flow field inside a DCW MHD channel with four sectors indicated where the boundary layer regimes will be illustrated in expanded figures. Slope angles: 45, 90.



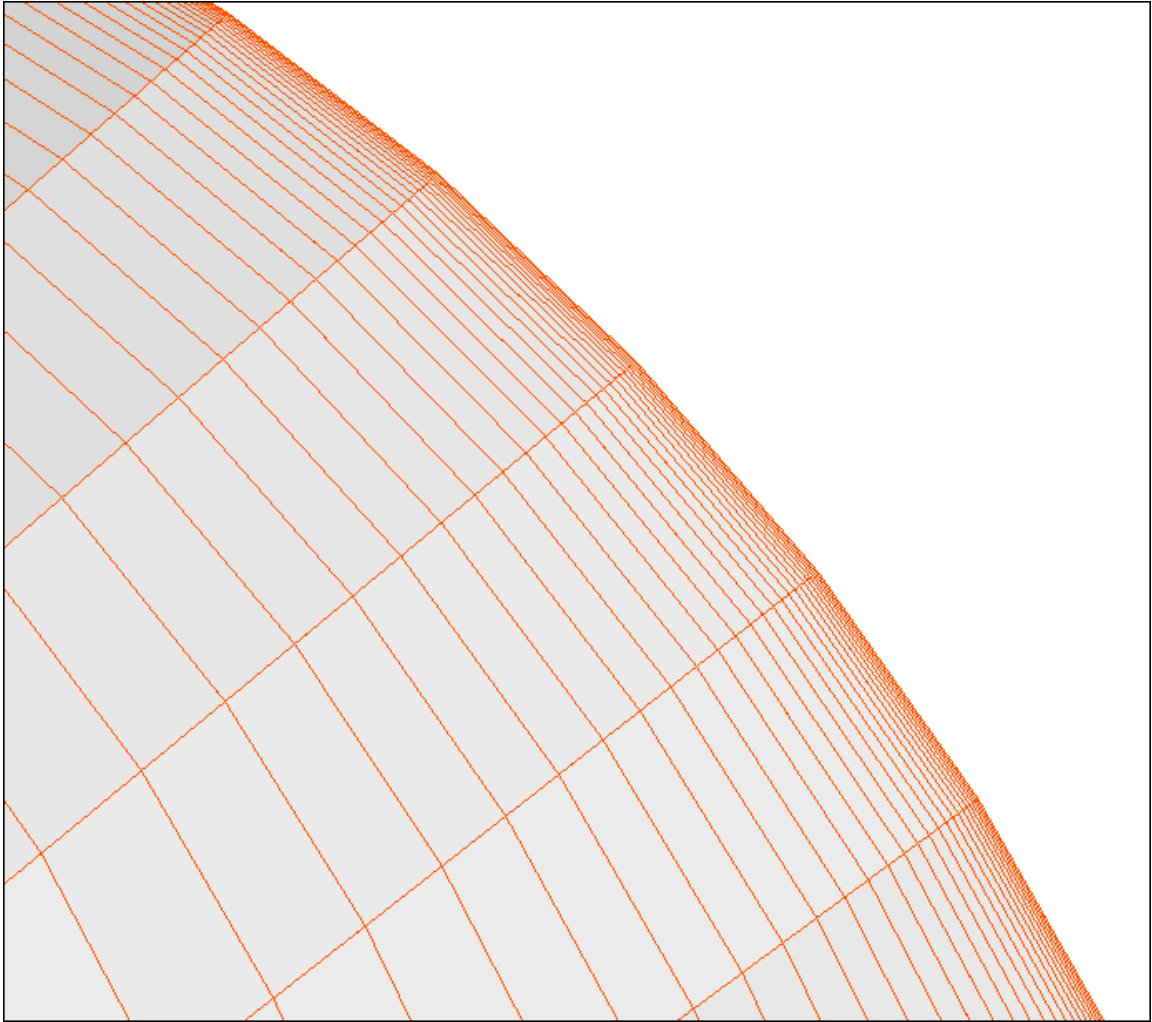
**Figure 3.9.31.** The radial mesh for resolving the plasma boundary layer and, also the axial mesh stretch distribution for the axial plasma face at the inlet, which represents the magnified sector **1**, **Figure 3.9.30**. Cooper's algorithm for the plasma volume mesh generation.



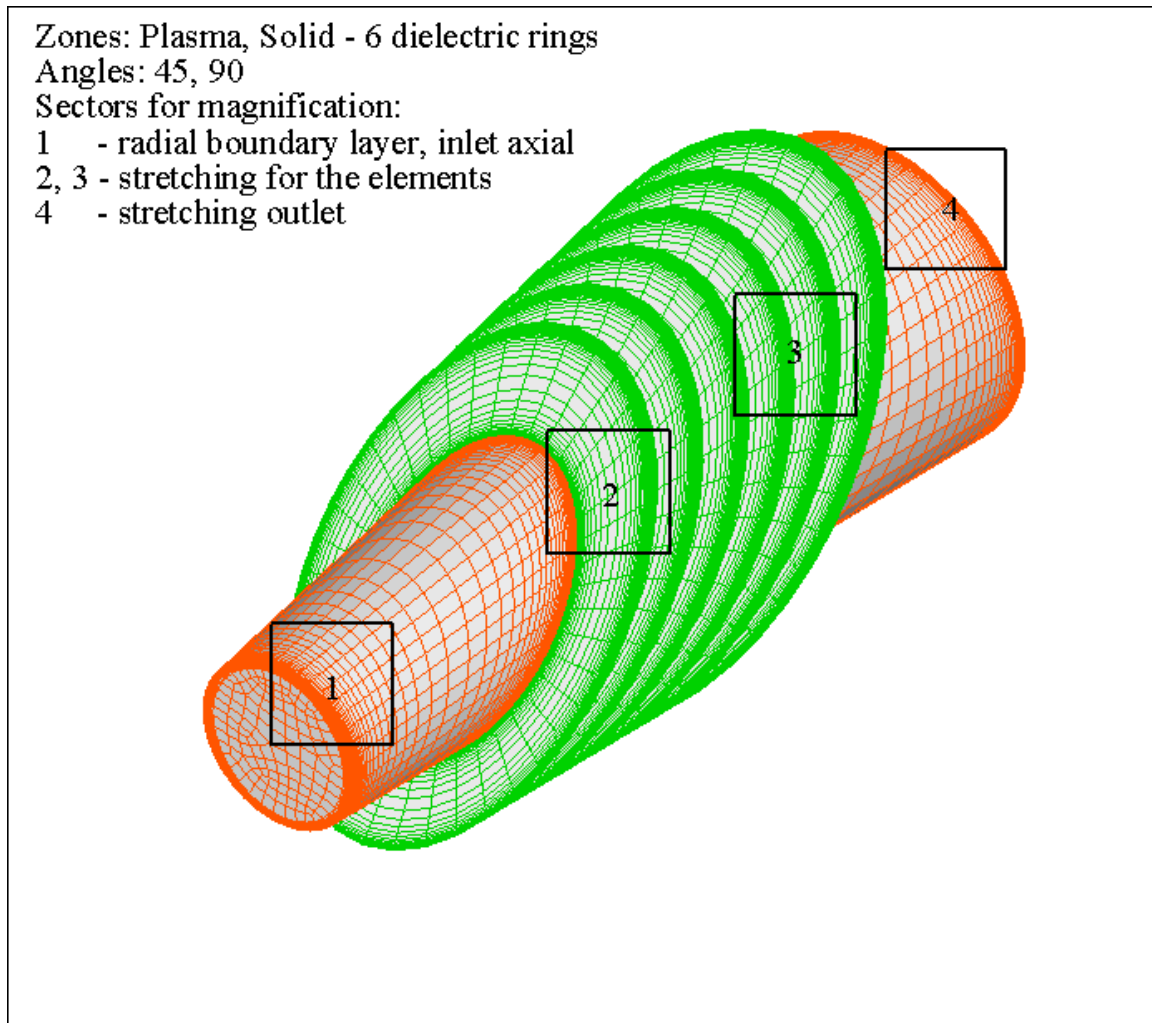
**Figure 3.9.32.** The axial mesh stretch process for the plasma volumes adjacent to conductor and insulator electric elements which represents the magnified sector 2, **Figure 3.9.30.**



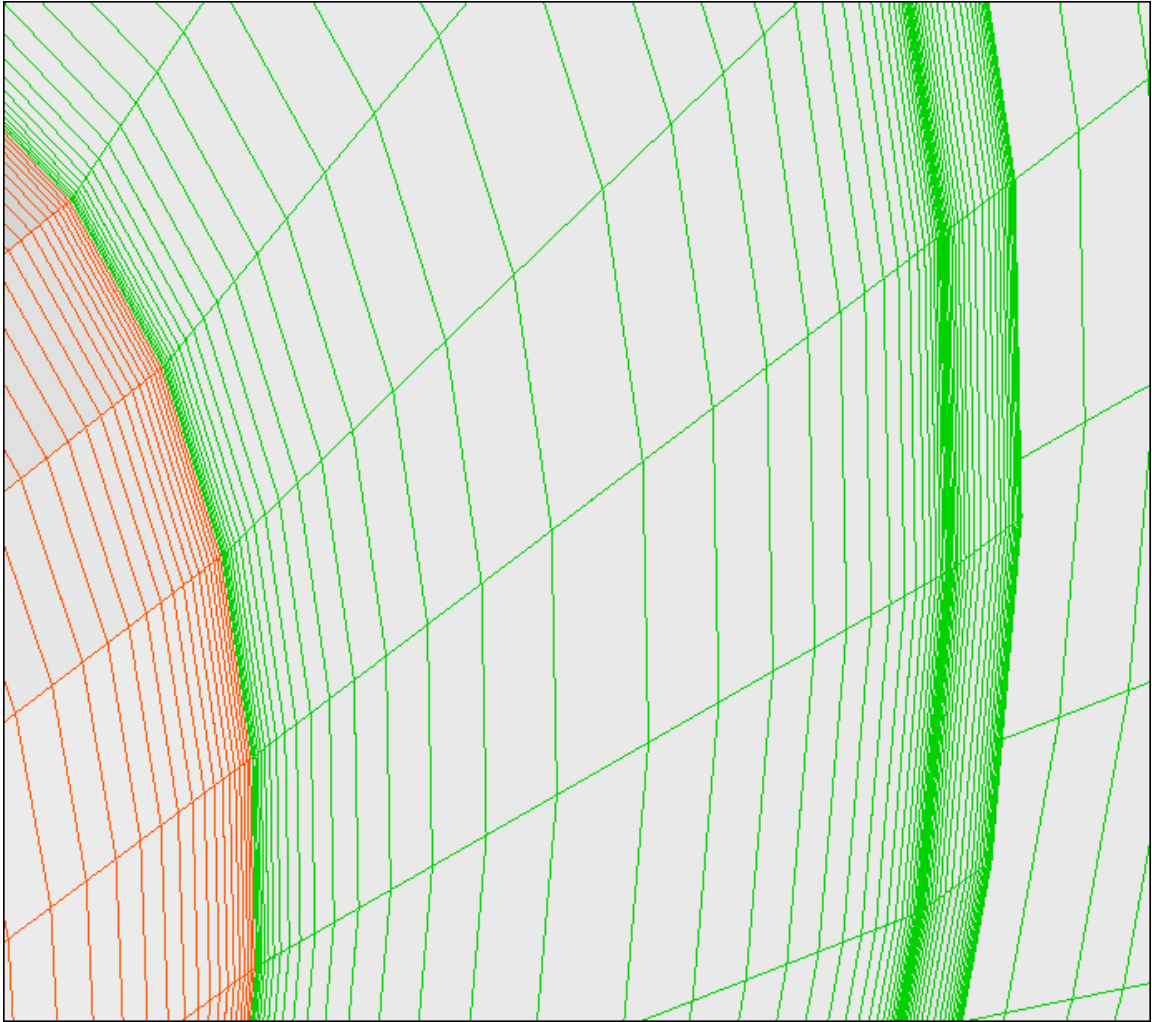
**Figure 3.9.33.** The axial mesh stretch process for plasma face, and also volumes, adjacent to electric elements of the channel, which represents the magnified sector 3, **Figure 3.9.30.**



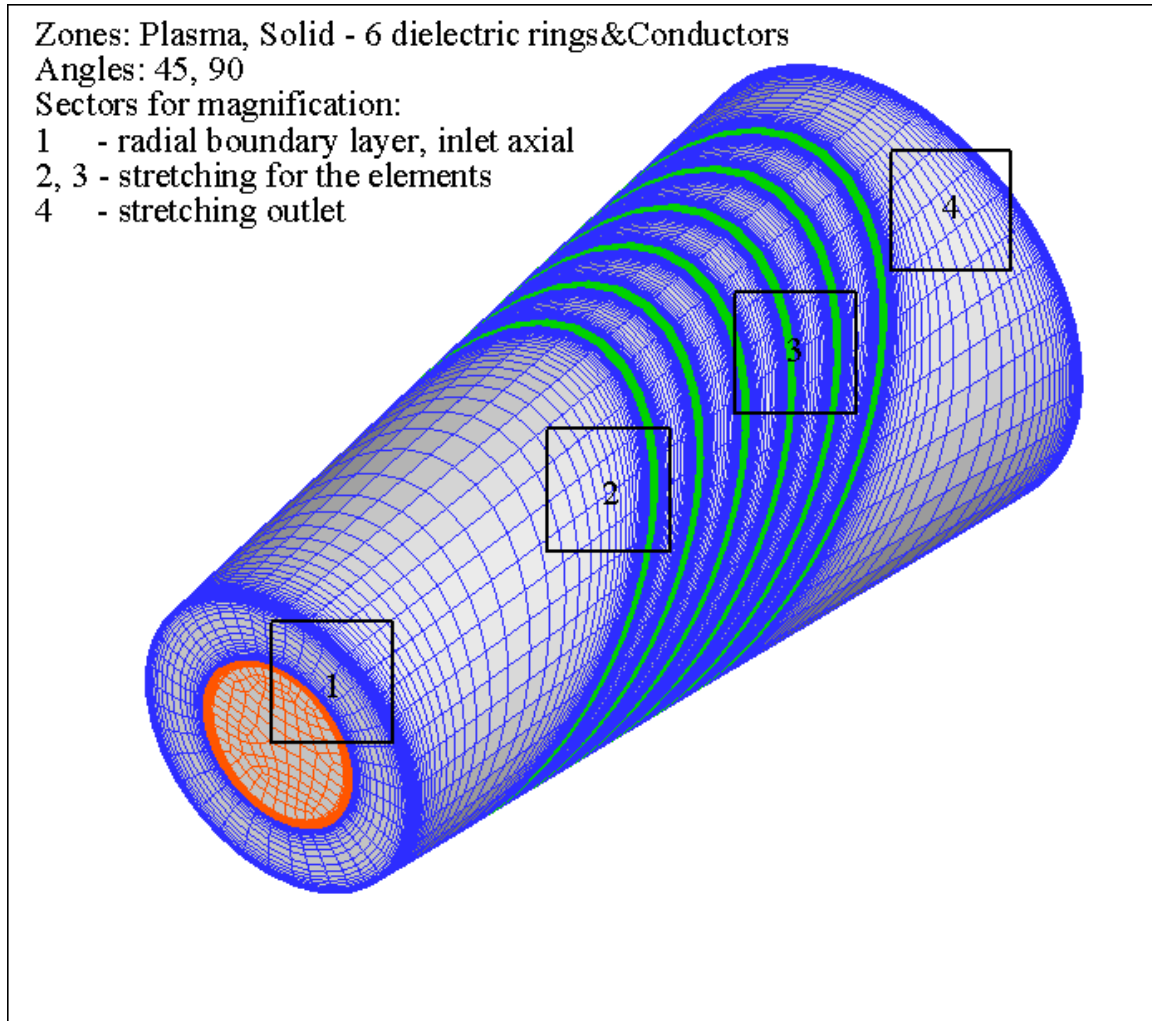
**Figure 3.9.34.** The axial mesh stretch process of the plasma face and volume at outlet boundary of the channel, representing sector 4, **Figure 3.9.30**.



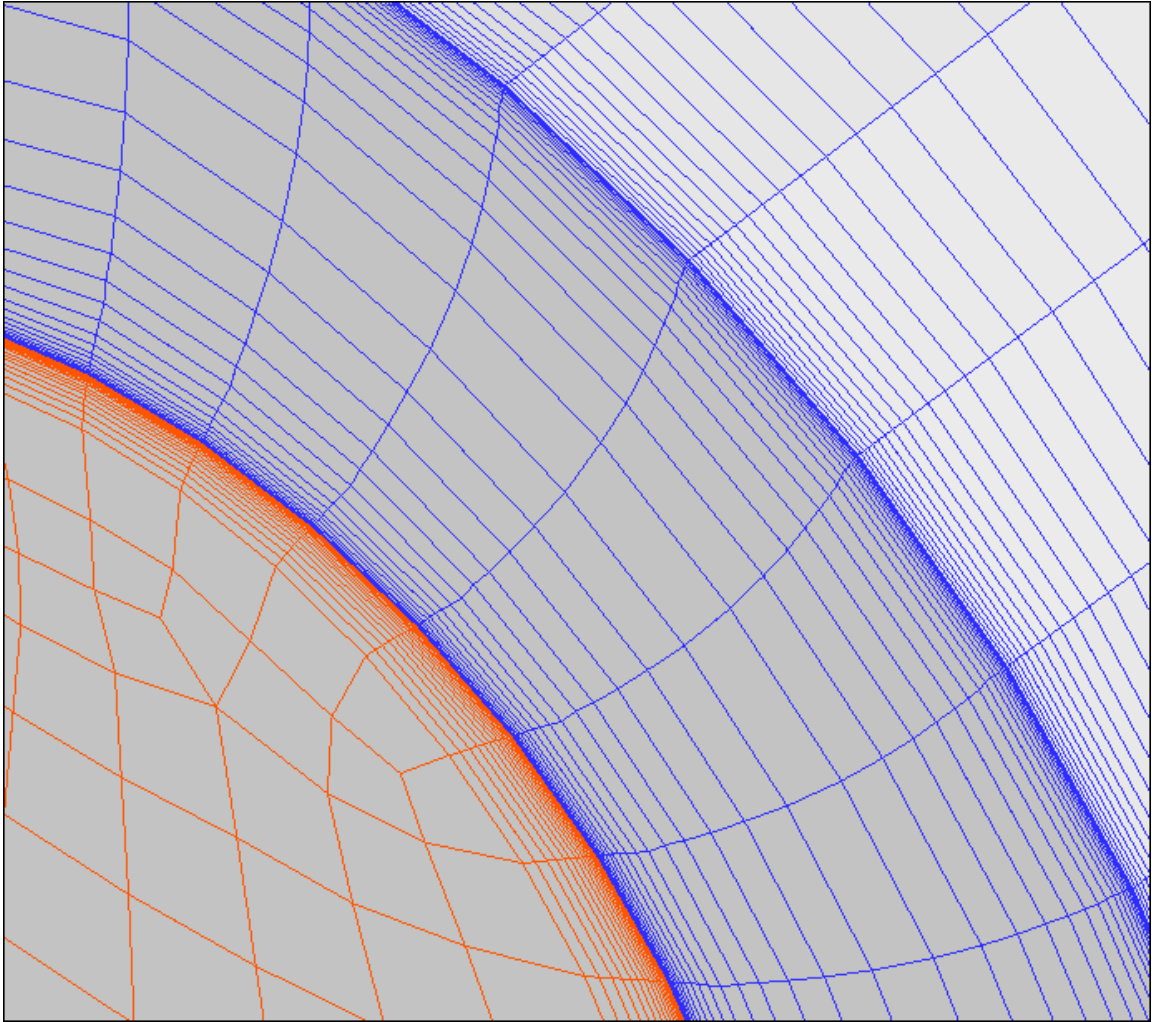
**Figure 3.9.35.** The mesh configuration used to represent the insulator elements of the diagonal wall MHD channel with sectors **1**, **2**, **3** and **4** noted.



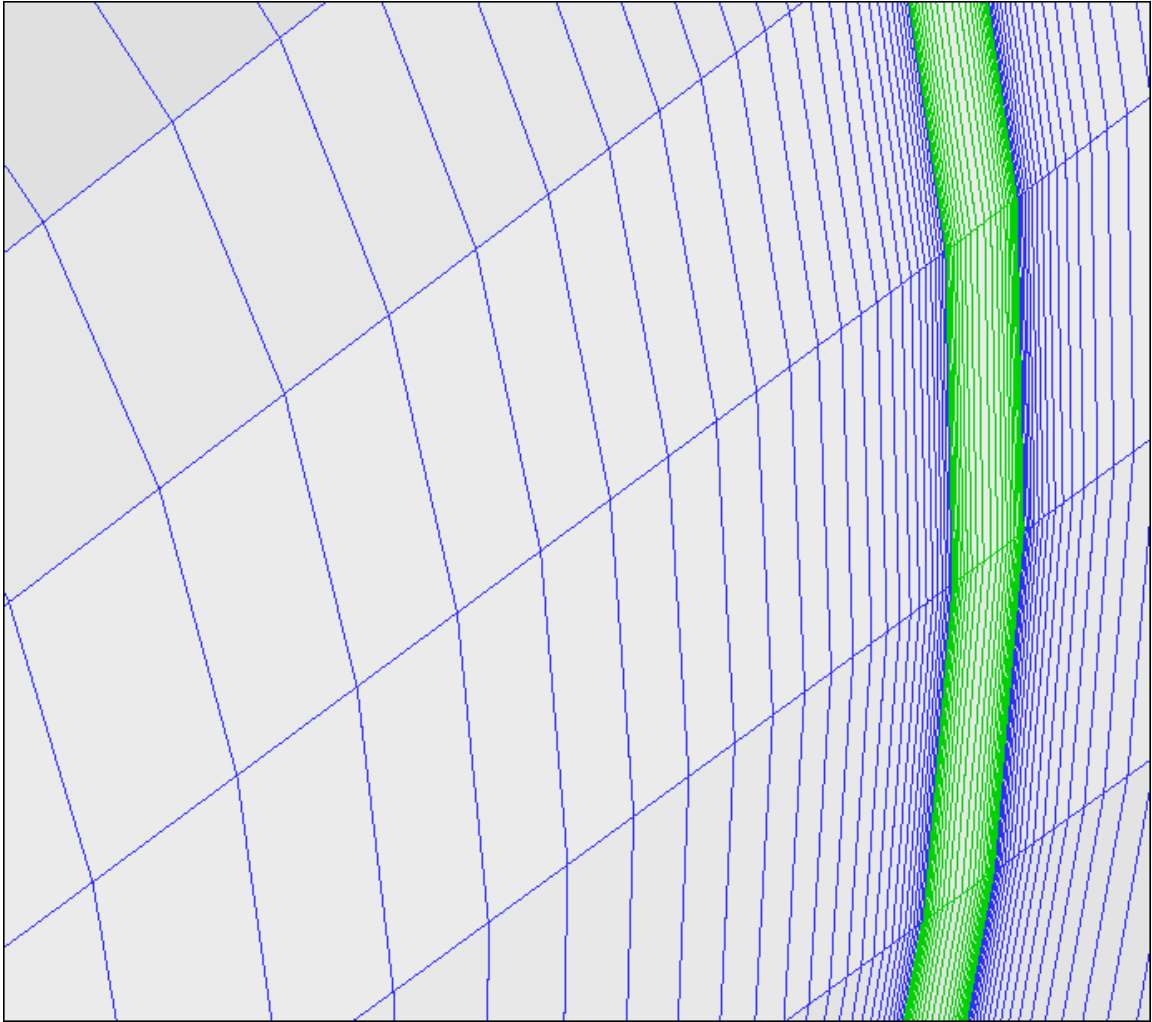
**Figure 3.9.36.** The radial mesh stretch process for the insulators and the axial mesh stretch for the adjacent plasma face and volume with different and independent aspect ratios, representing sector 2, **Figure 3.9.35**.



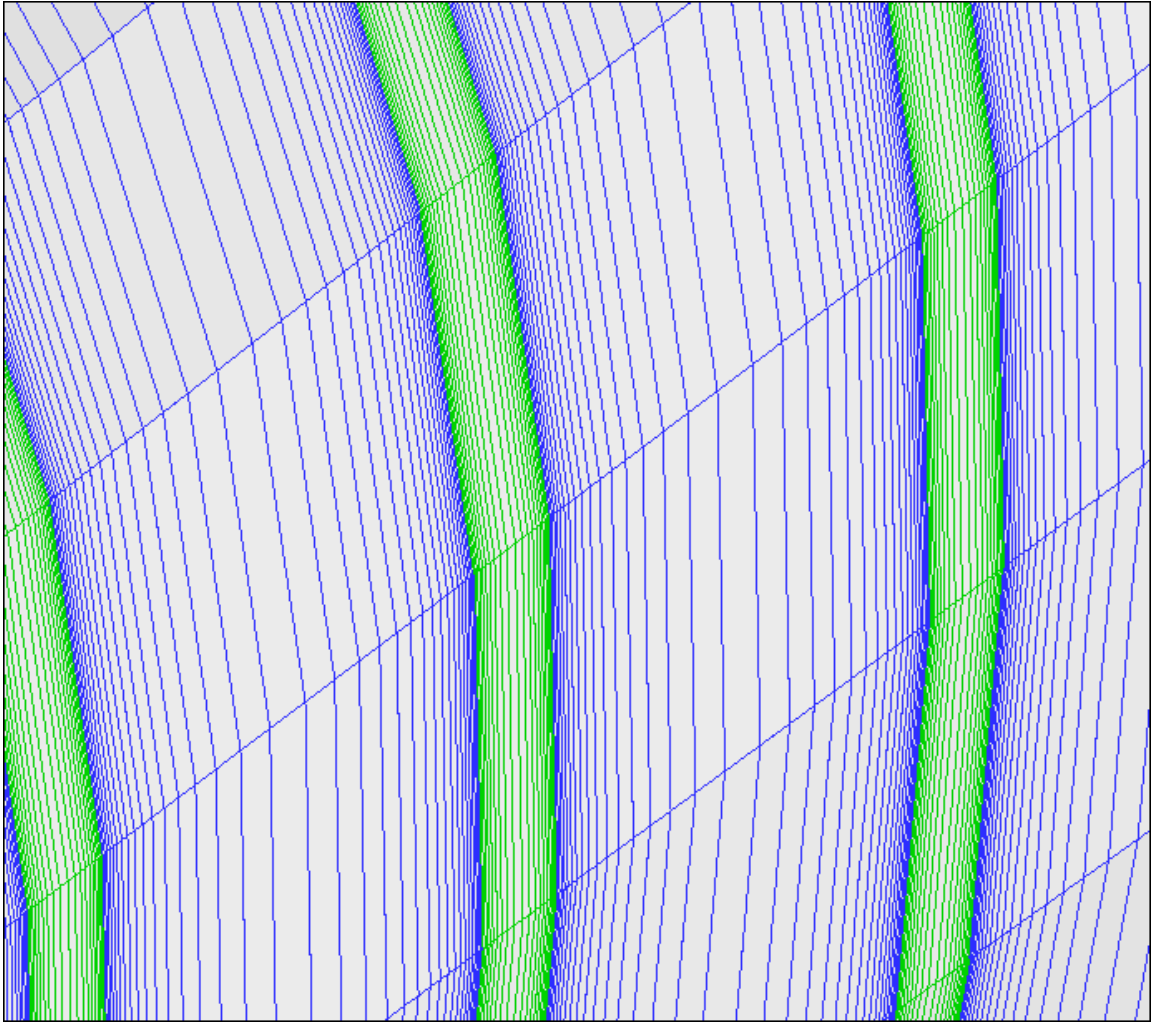
**Figure 3.9.37.** The mesh representing the DCW MHD generator with conductor and insulator elements indicated, and four sectors noted for further expanded representation.



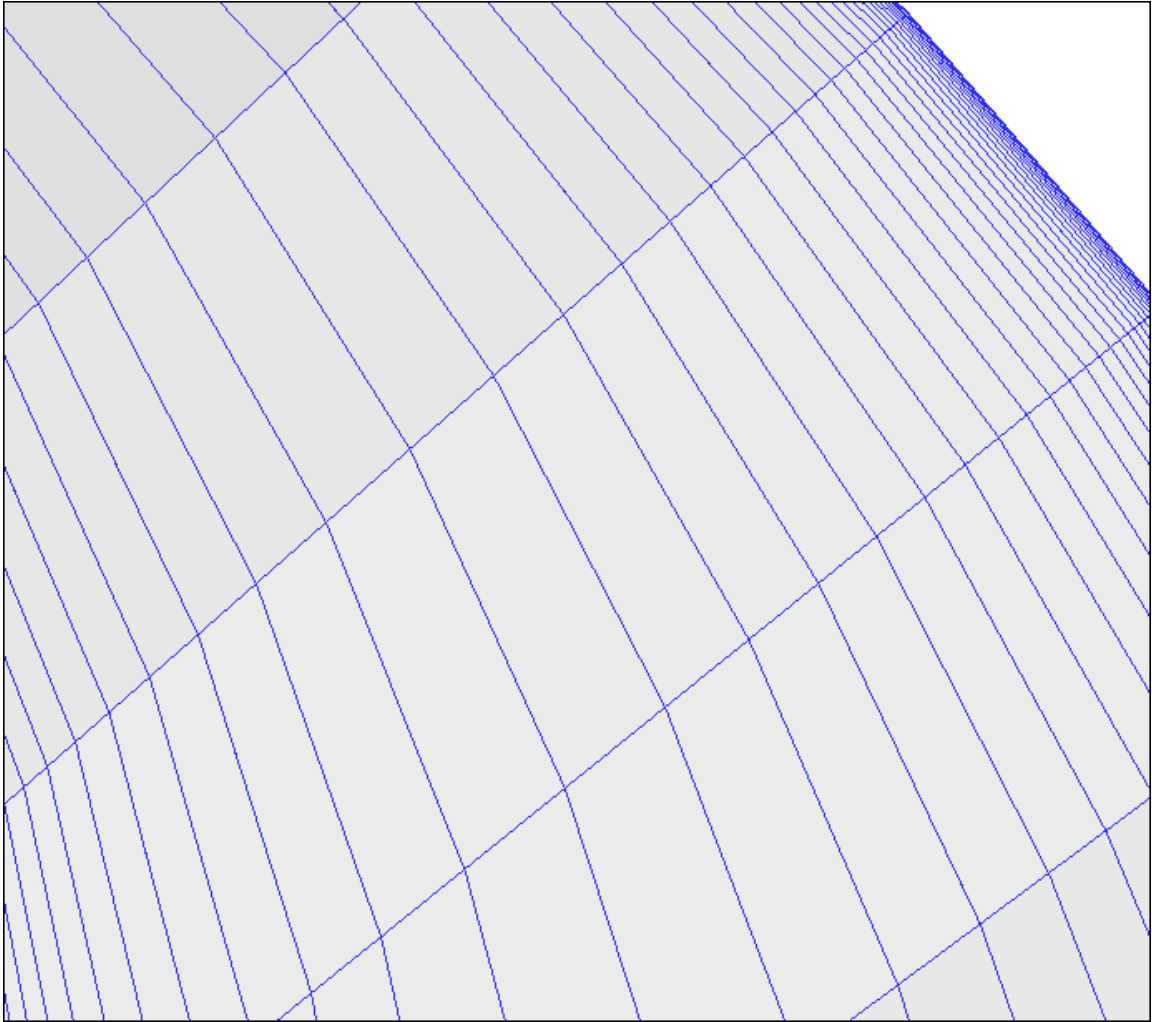
**Figure 3.9.38.** The radial mesh stretch for the plasma and the solid conductor, as well as the axial mesh stretching on the conductor outer surface, at the inlet plane, representing magnified sector **1**, **Figure 3.9.47**.



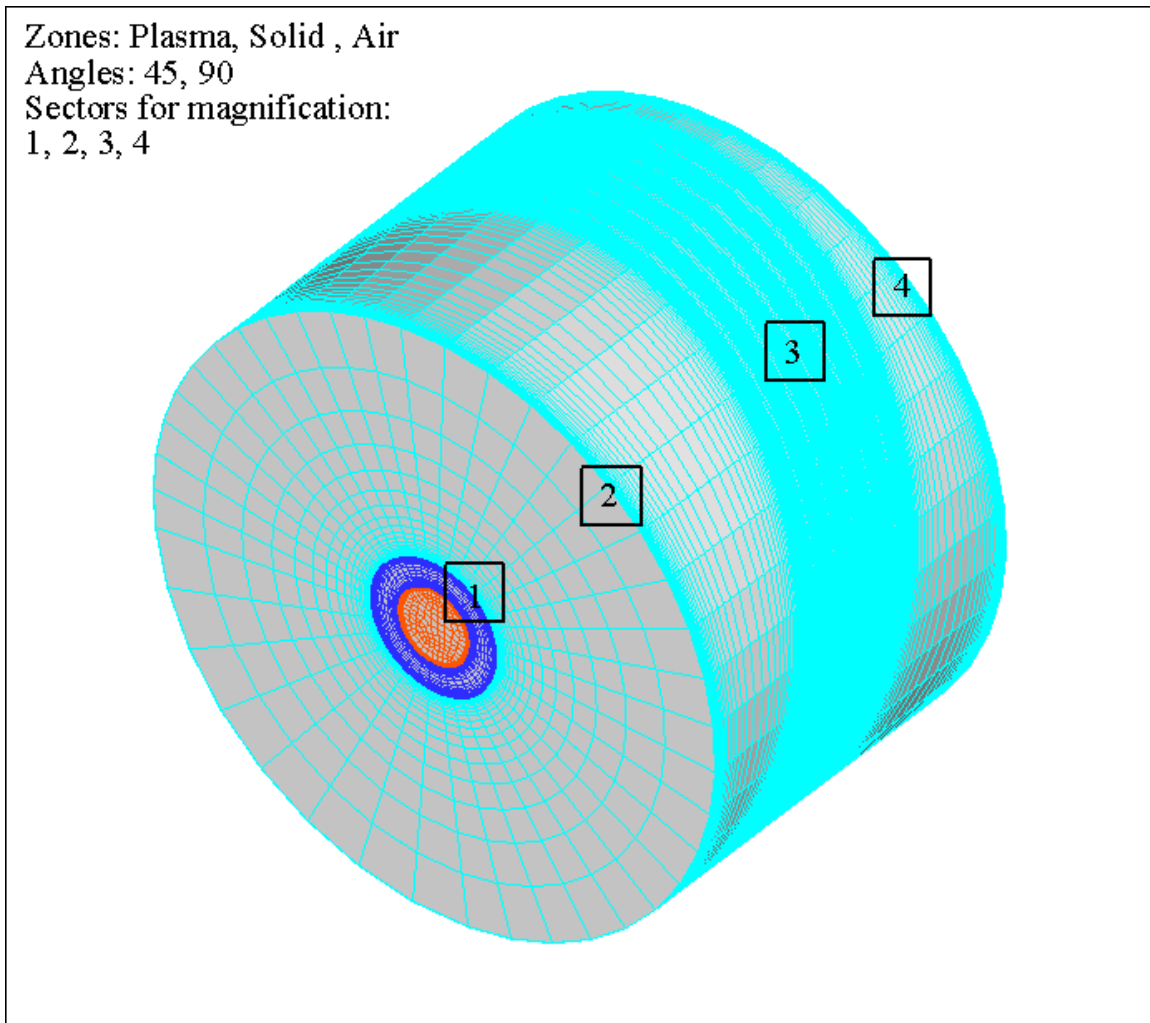
**Figure 3.9.39.** The axial mesh stretch process for the first (largest) solid conductor face and volume, and the first adjacent insulator interface, generated with the same axial aspect ratios, representing magnified sector 4, **Figure 3.9.37**.



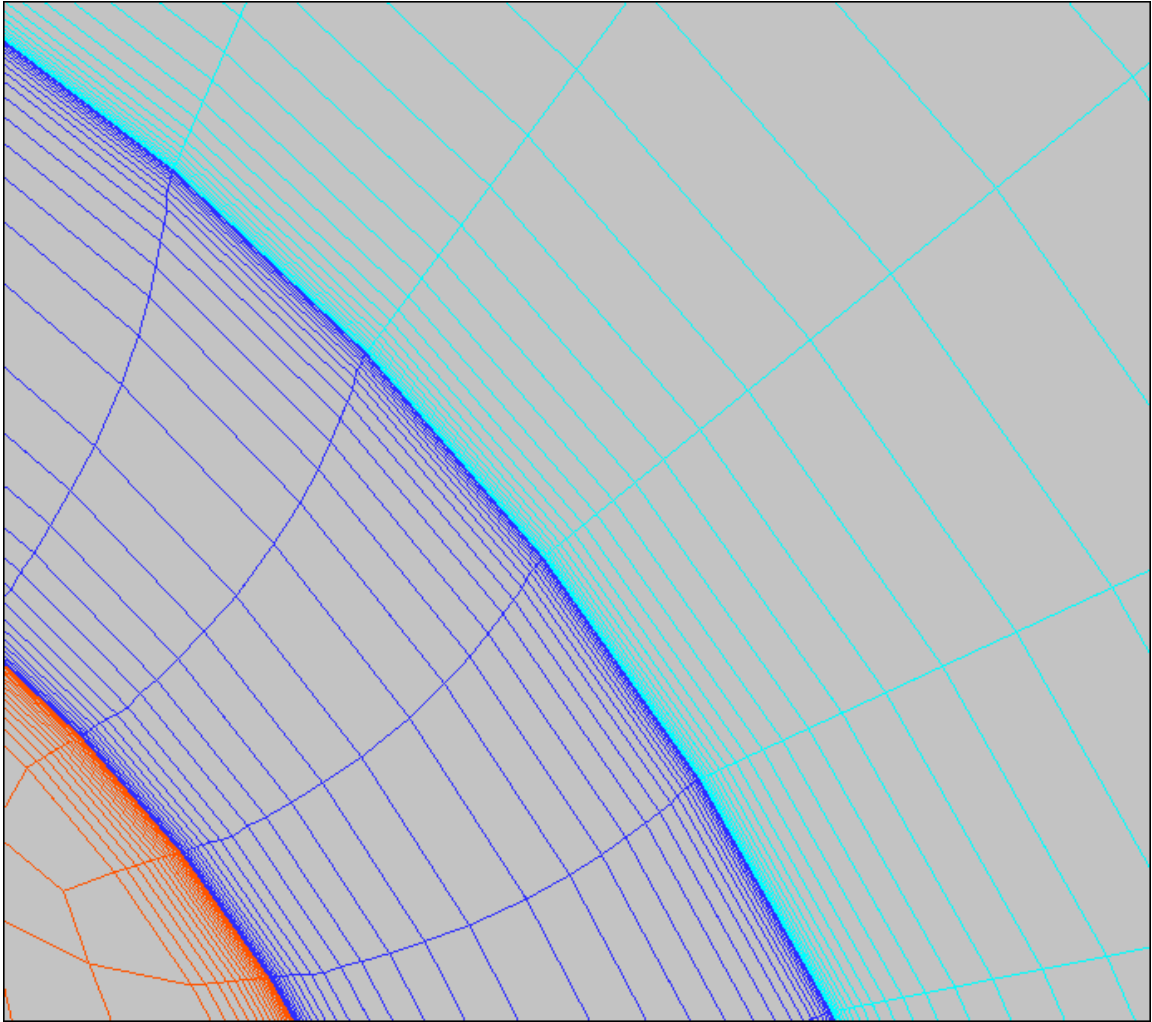
**Figure 3.9.40.** Two sided mesh stretch process on the solid electric elements (conductors and insulators), based on one aspect ratio called the axial precision; representing magnified sector 3, **Figure 3.9.37**.



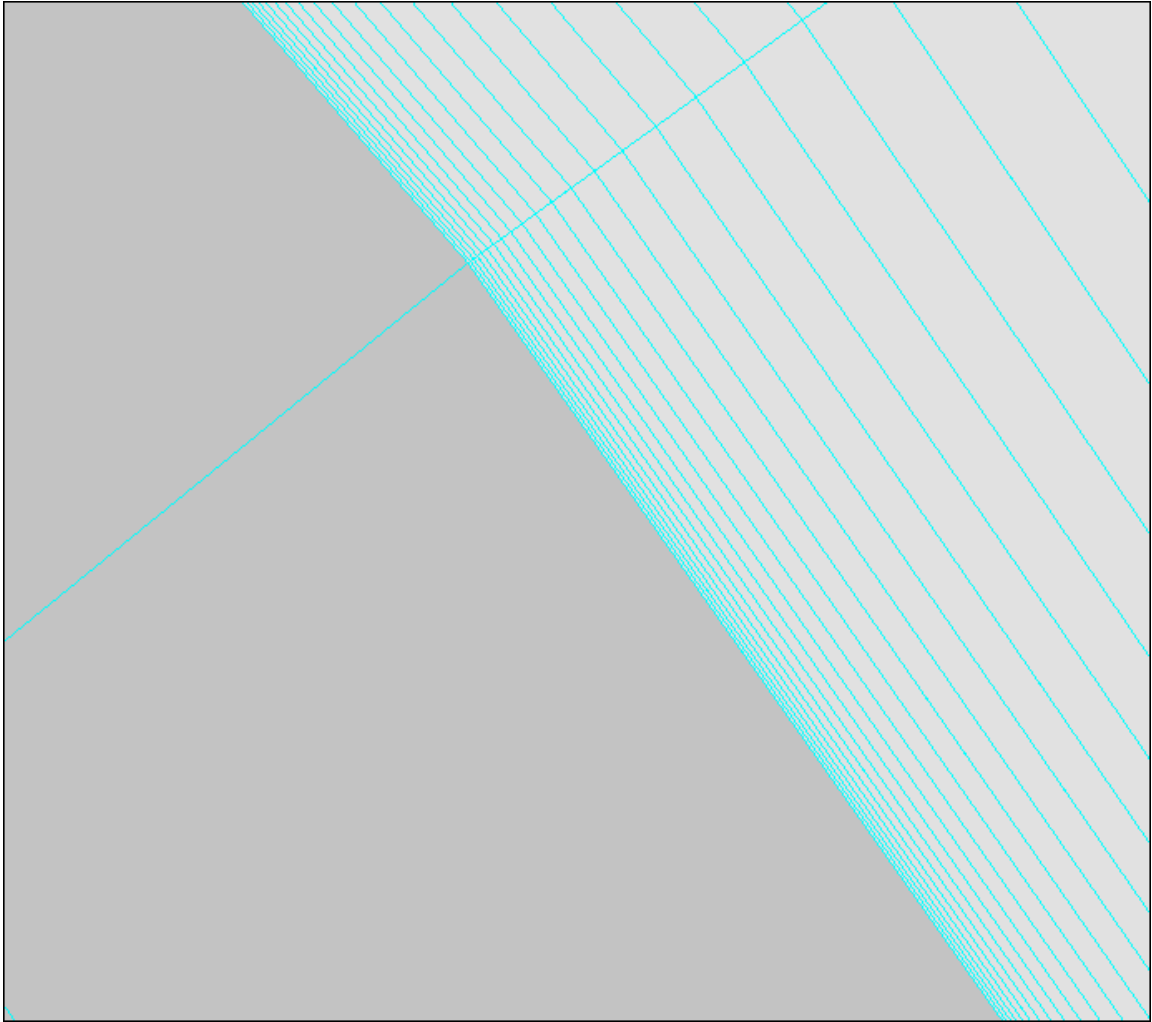
**Figure 3.9.41.** The axial mesh stretch process on outer surface of the last conductor at the outlet; magnified sector 4, **Figure 3.9.37**.



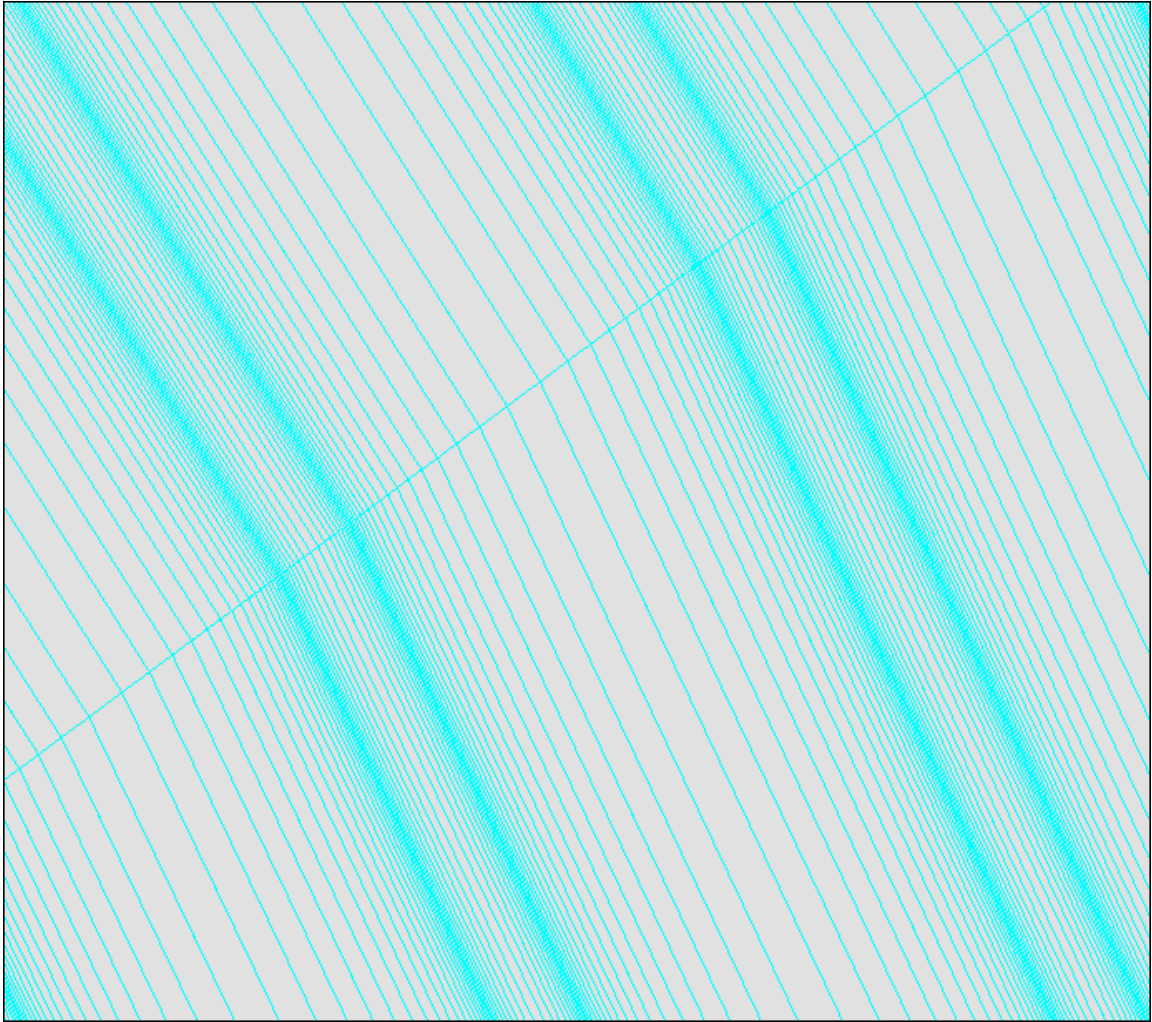
**Figure 3.9.42.** The meshed domain representing the volume of air surrounding the DCW MHD channel. Four sectors **1**, **2**, **3** and **4** are noted for further magnified representation.



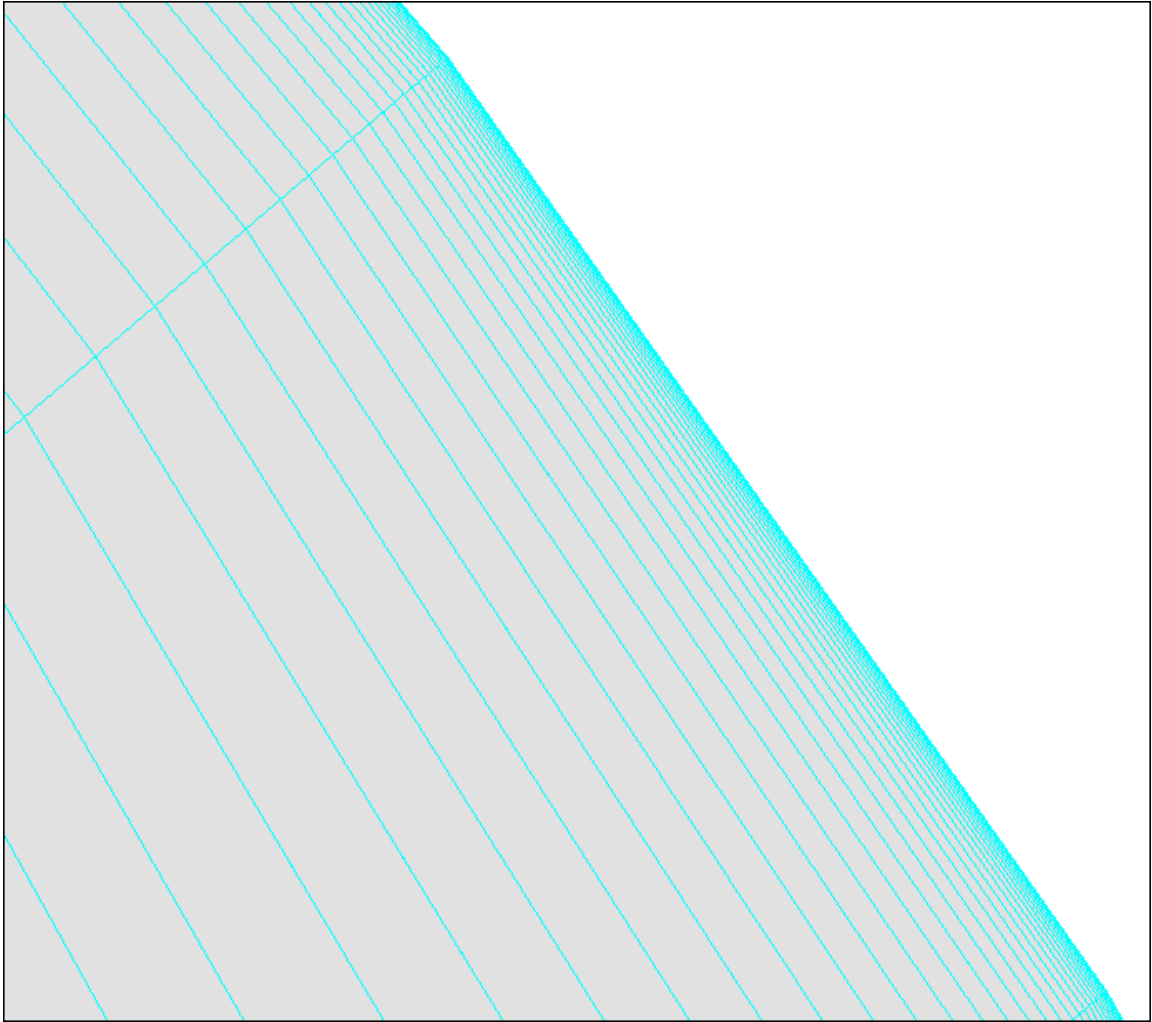
**Figure 3.9.43.** The radial mesh stretch or distribution across the inlet plane, showing the boundary layer regime of the plasma, the solid conductor face, and the air domain, based on three equal but independent aspect ratios; magnified sector **1**, **Figure 3.9.42**.



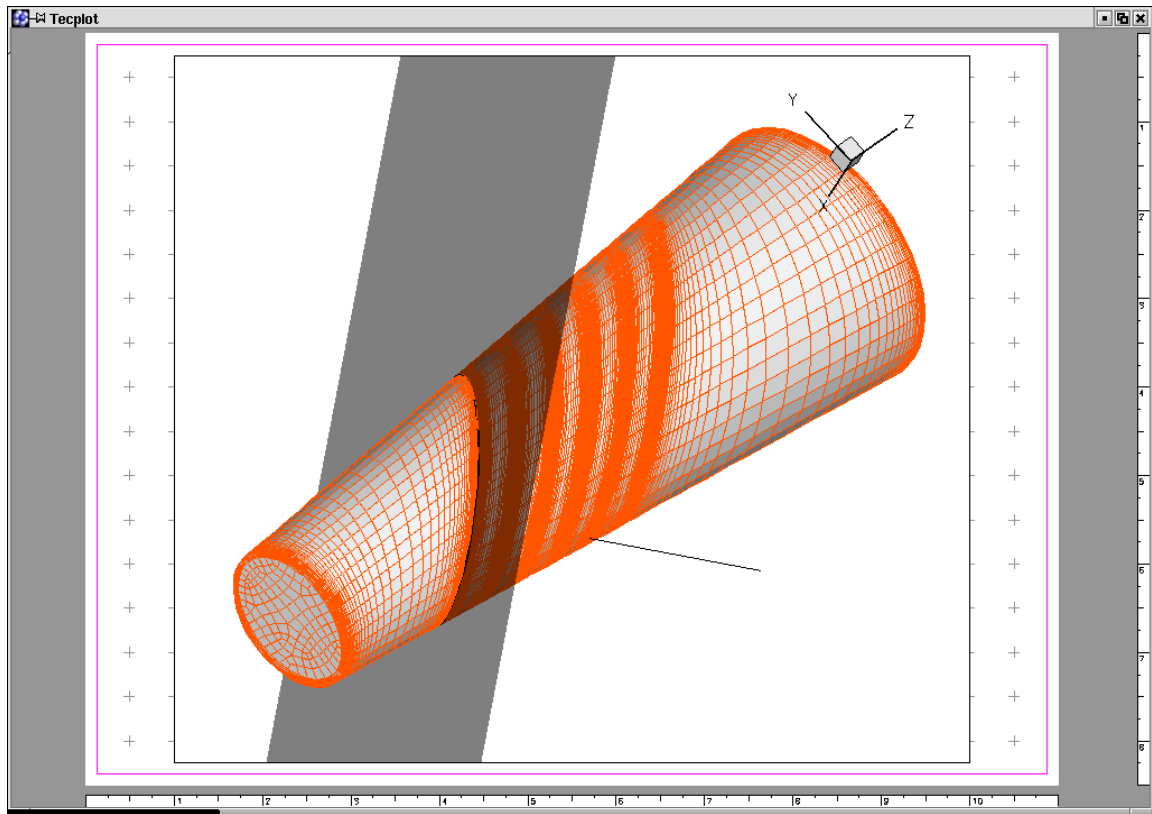
**Figure 3.9.44.** Magnification of the last row of one-sided mesh stretch process at the air inlet and the two-sided axial mesh stretch process on the air face at the inlet boundary; sector 2, **Figure 3.9.42**.



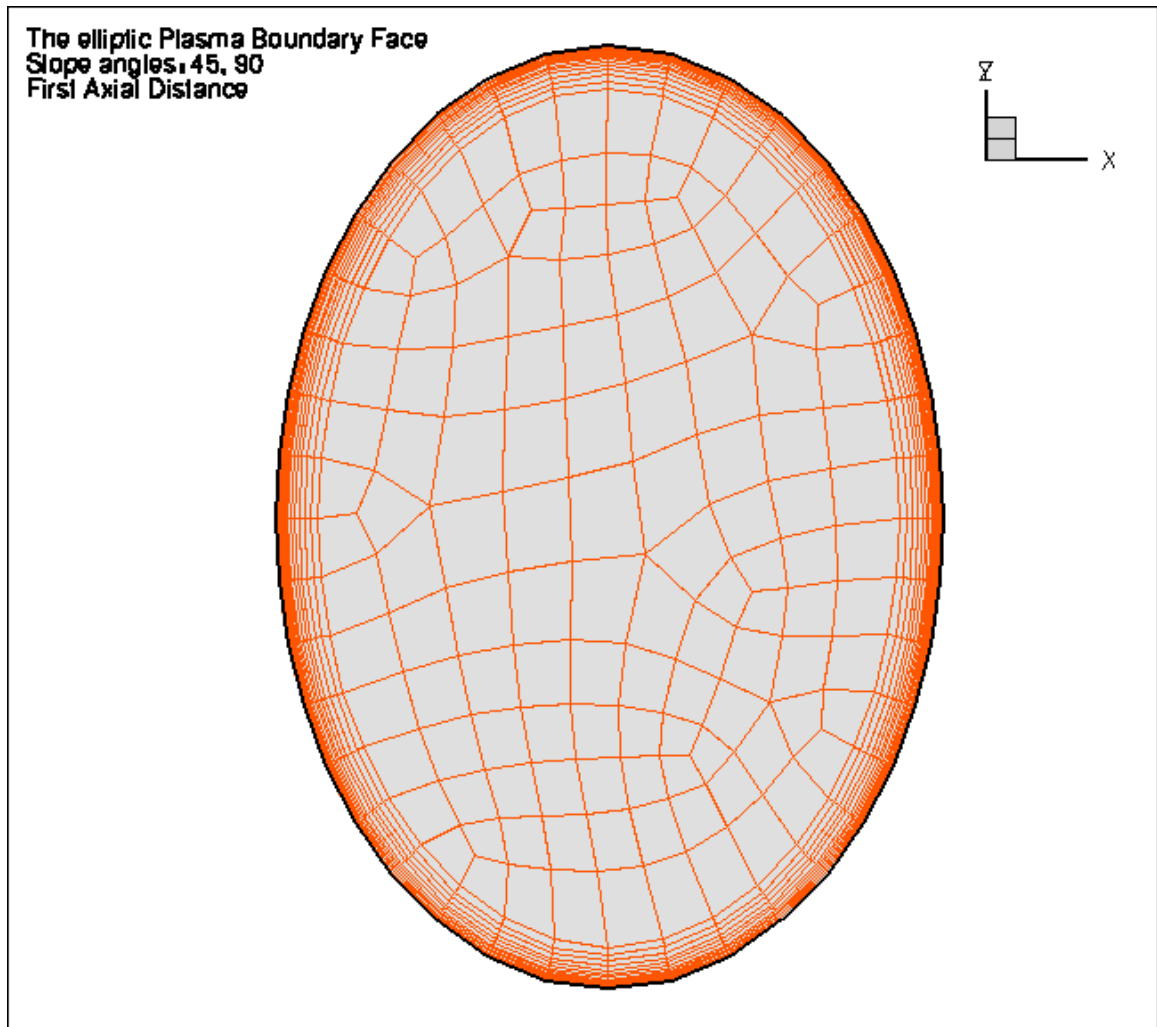
**Figure 3.9.45.** Magnification of the axial mesh stretch process for the air faces showing the same double-sided axial aspect ration as the corresponding solid electric elements, sector 3, **Figure 3.9.42**.



**Figure 3.9.46.** Magnification of the axial mesh stretch process on the air face at the outlet, sector 4, **Figure 3.9.42.**



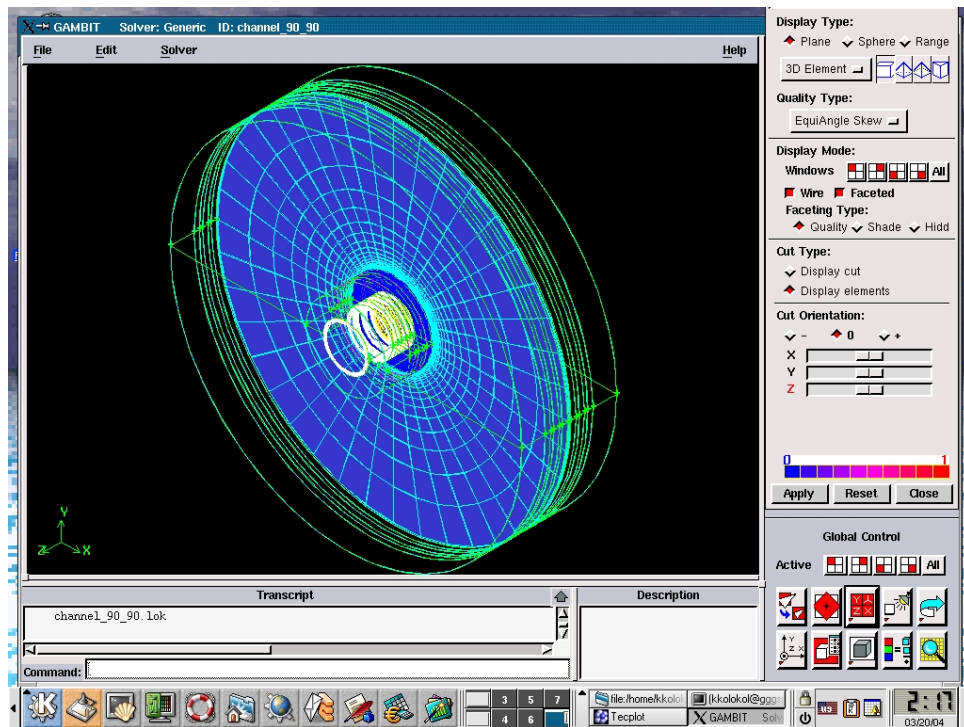
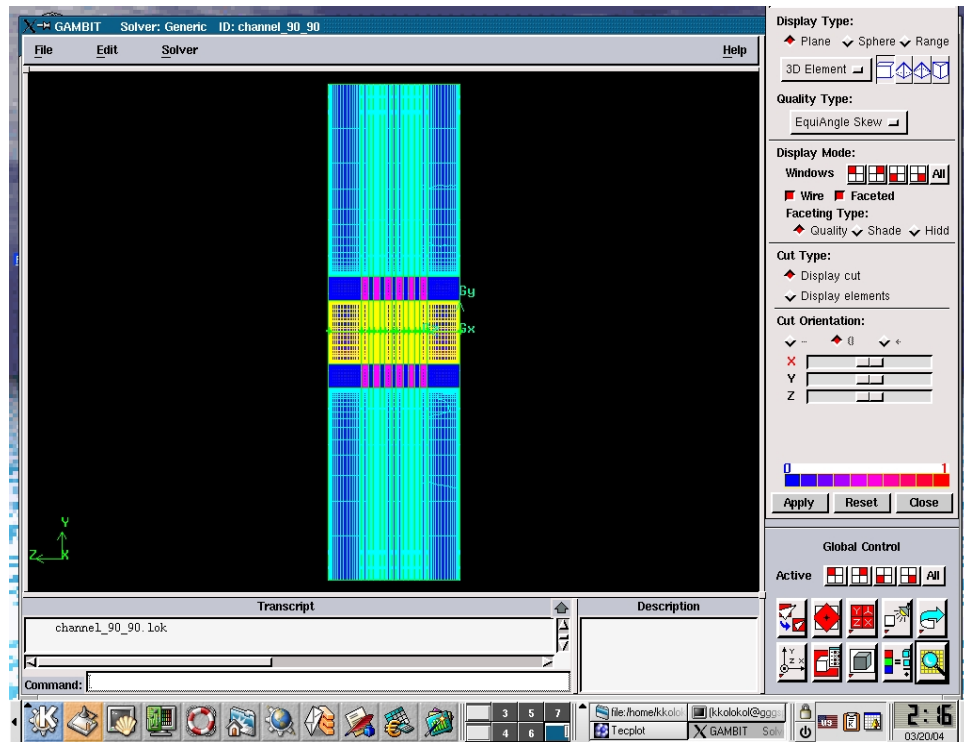
**Figure 3.9.47.** A cut plane or slice through the mesh representing the plasma volume, at an angle equal to the angle of inclination of an electric element, on the face plane of the electric element.



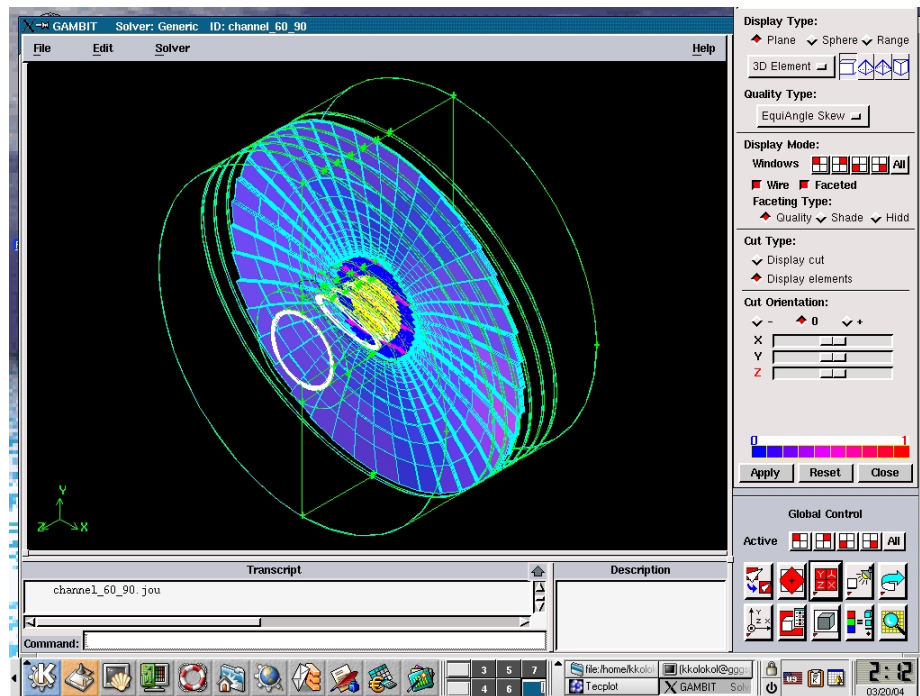
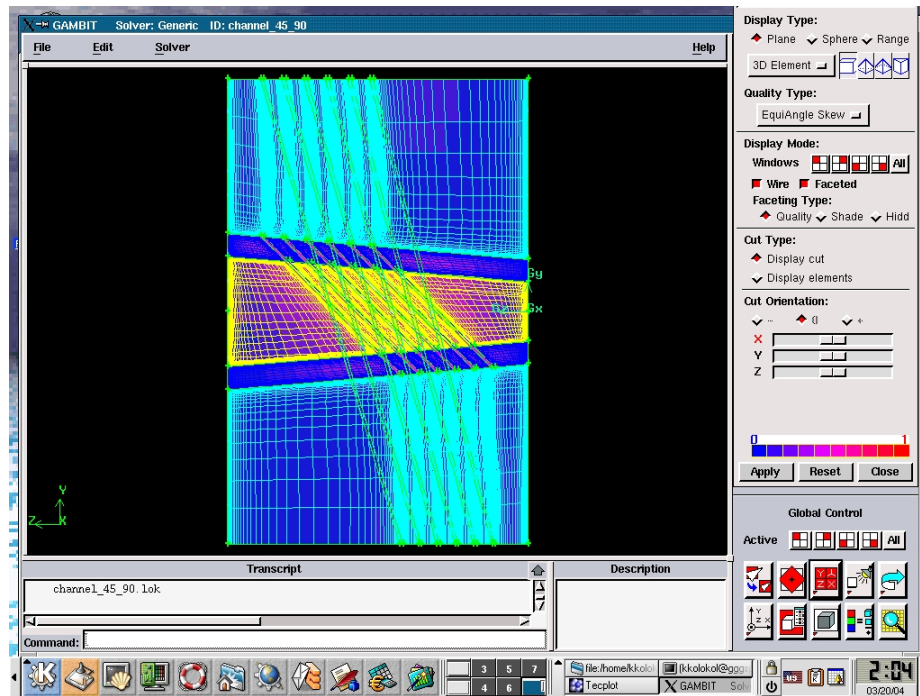
**Figure 3.9.48.** Plasma volume mesh in the plane inclined at the electrode element slant angle, and hence parallel to the solid elements. This slice can be done only at the plasma boundary faces corresponding to the electric elements.

## **APPENDIX A5**

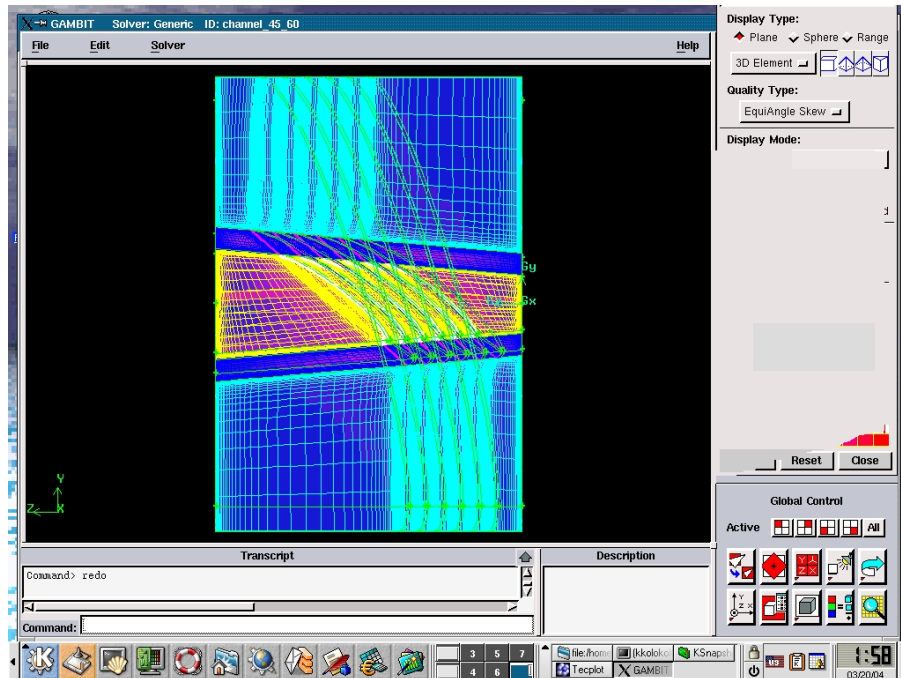
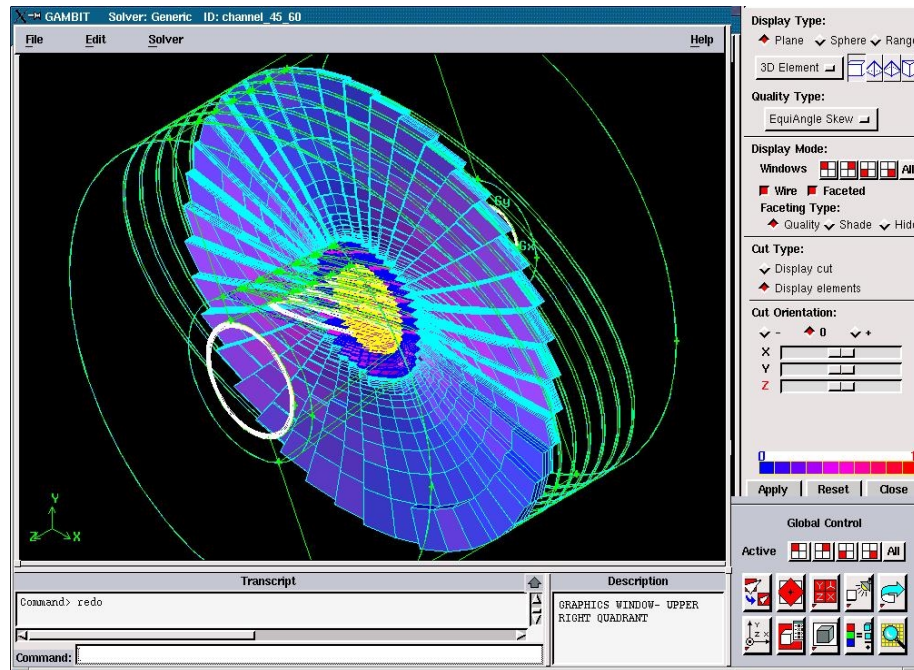
### **FIGURES ILLUSTRATING MESH QUALITY OF STRAIGHT CONDUCTIVITY AND DCW MHD CHANNELS**



**Figure 3.9.49.** Display of the level of quality of the equiangular mesh used to represent constant area conductivity channel surrounded by air domain.



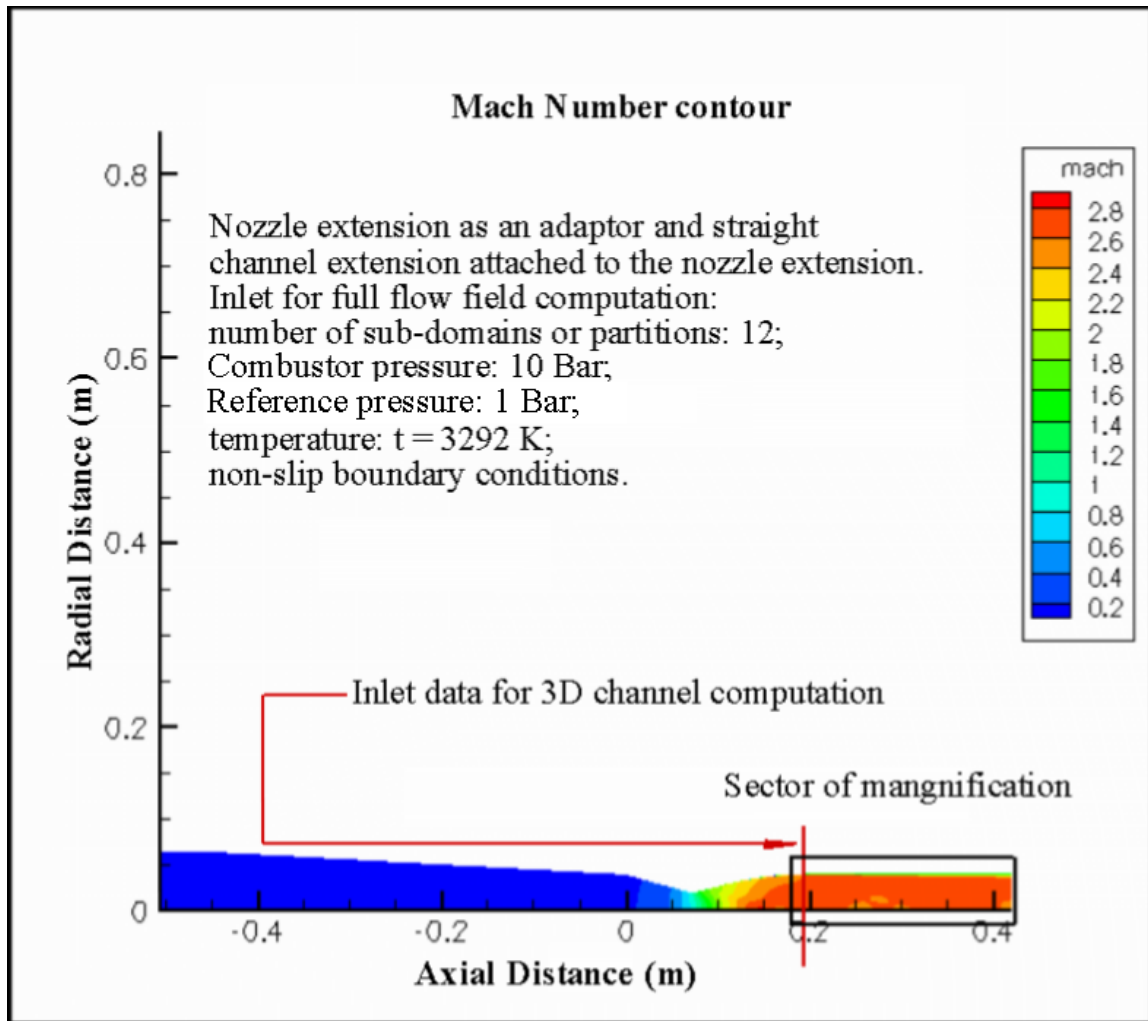
**Figure 3.9.50.** Display of the level of quality of the equiangular mesh used to represent the one-slope DCW MHD channel surrounded by air domain. Slope angles are 45 and 90 degrees respectively.



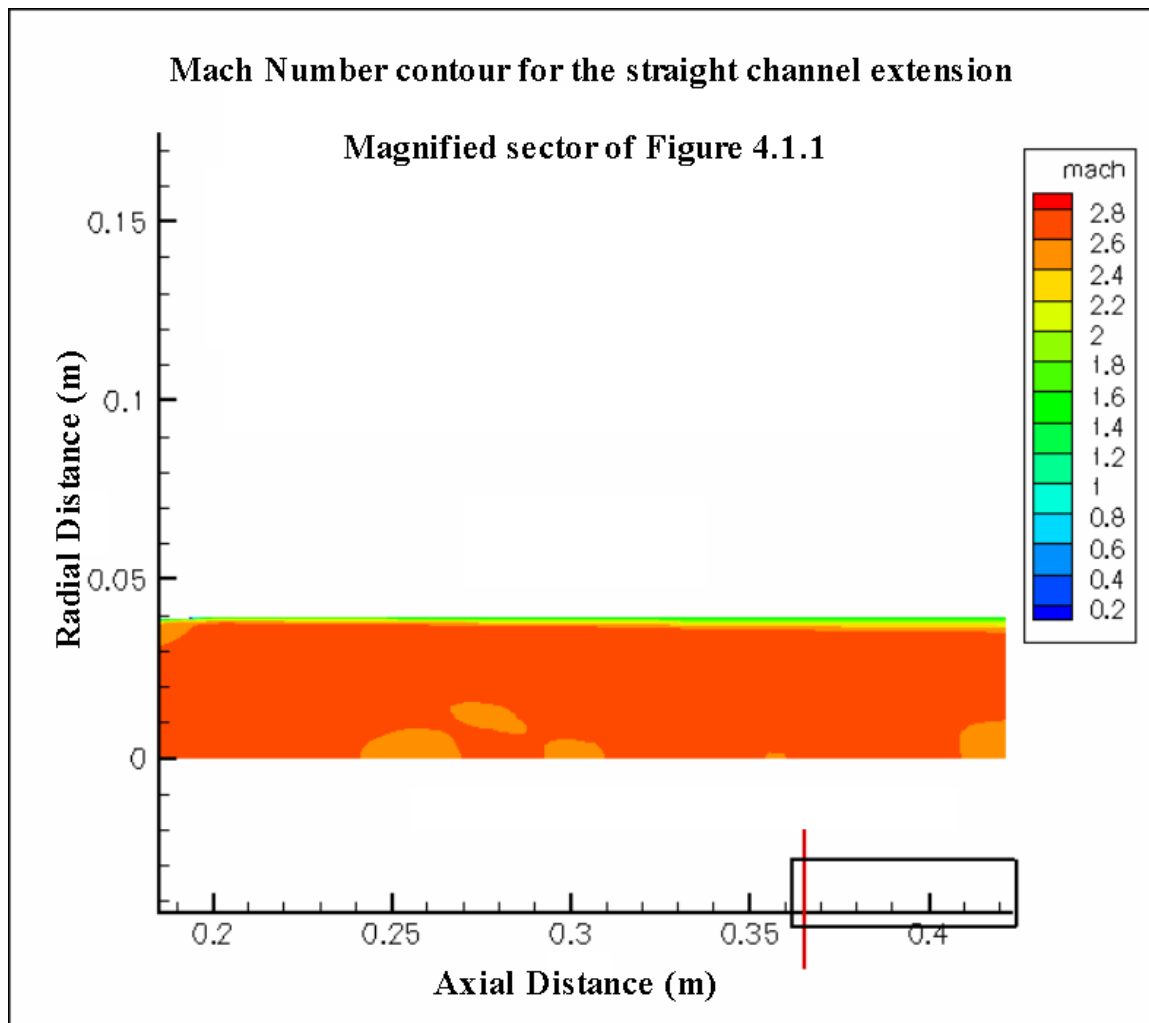
**Figure 3.9.49.** Display of the level of quality of the equiangular mesh used to represent two-slope DCW MHD channel surrounded by air domain. Slope angles are 45 and 60 degrees respectively.

## **APPENDIX A6**

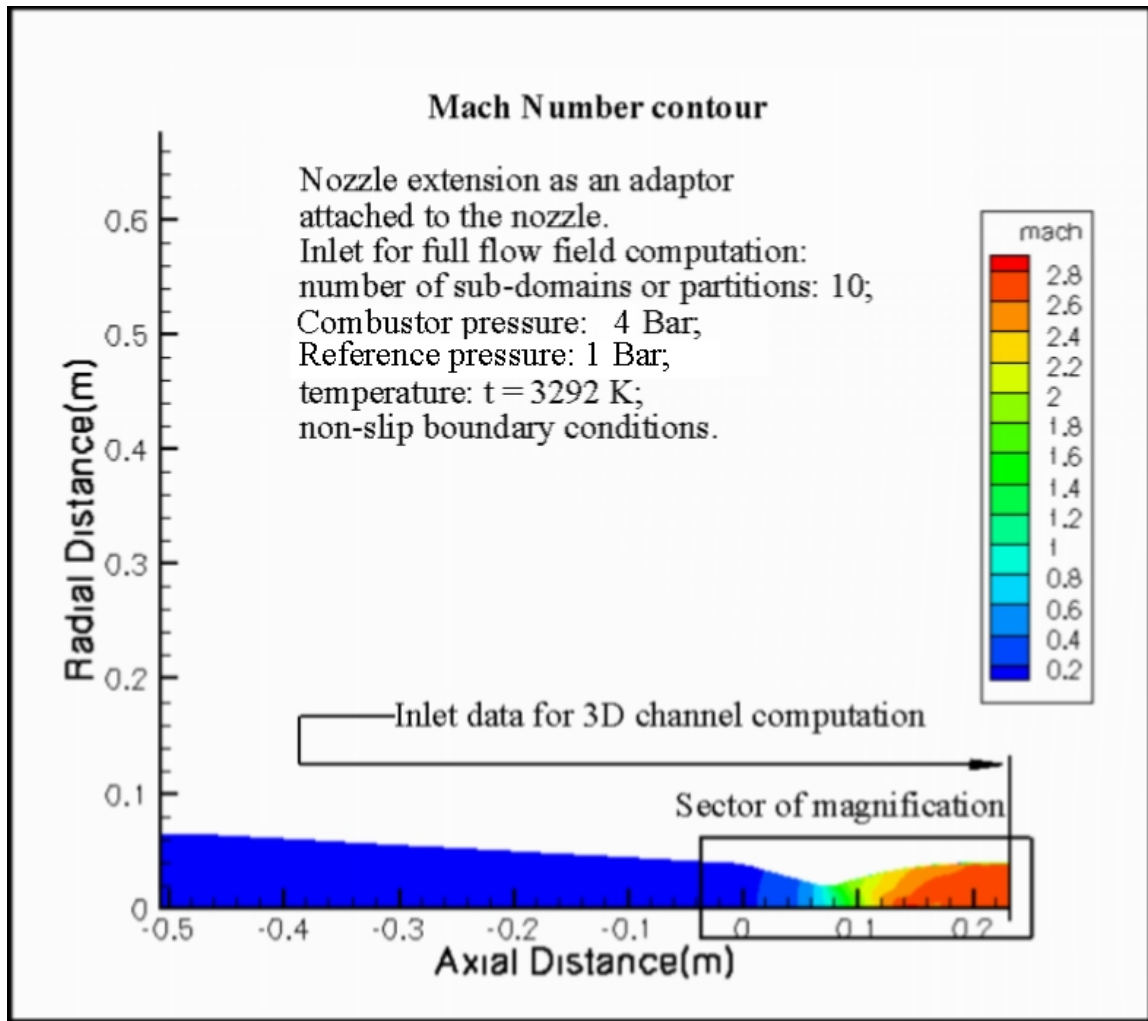
### **FIGURES ILLUSTRATING COMPUTATIONAL DATA FOR NOZZLE - COMBUSTOR**



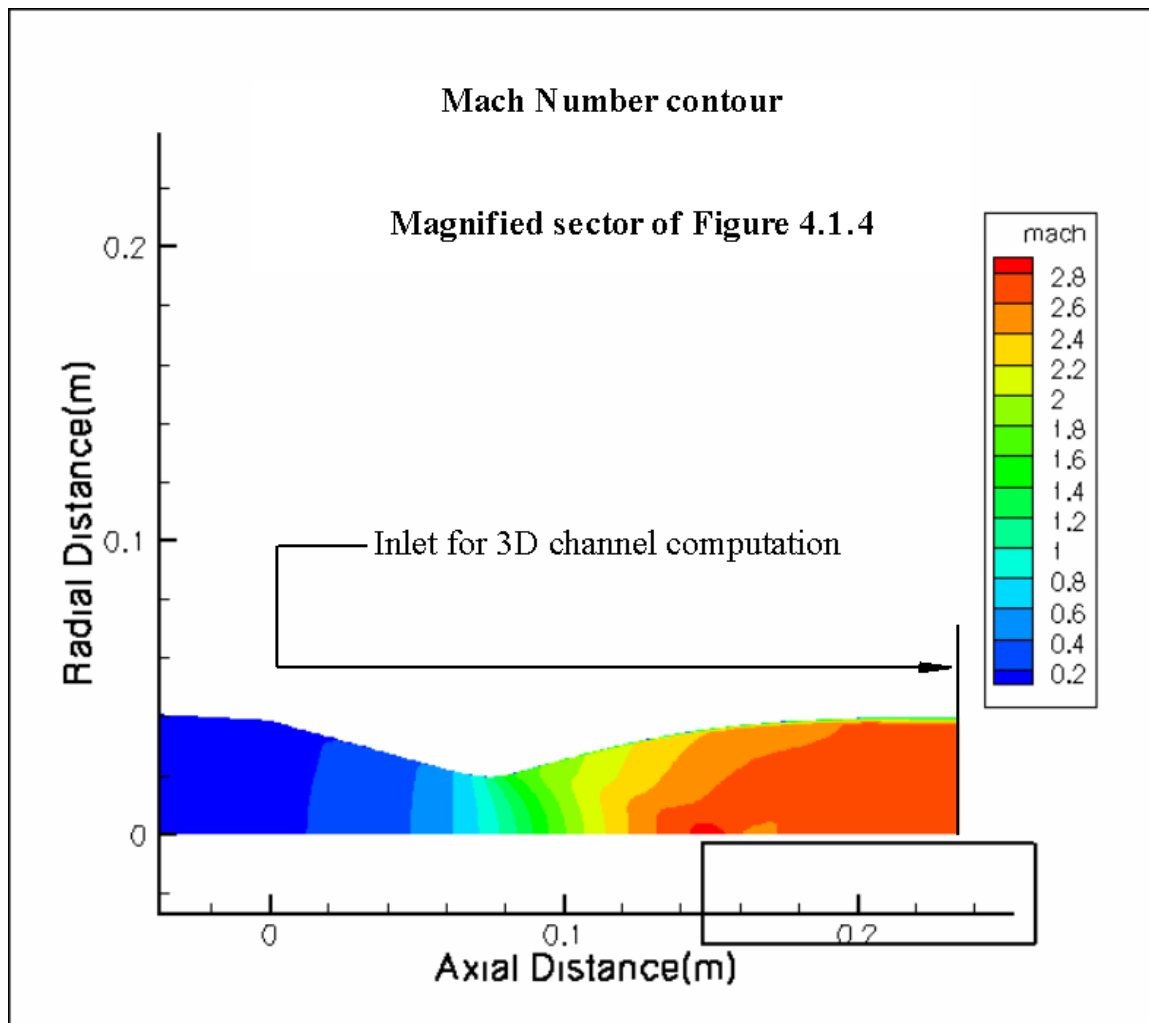
**Figure 4.1.1.** Mach number distribution in combustor outlet, nozzle extension or adaptor (supersonic diffuser) and channel extension.



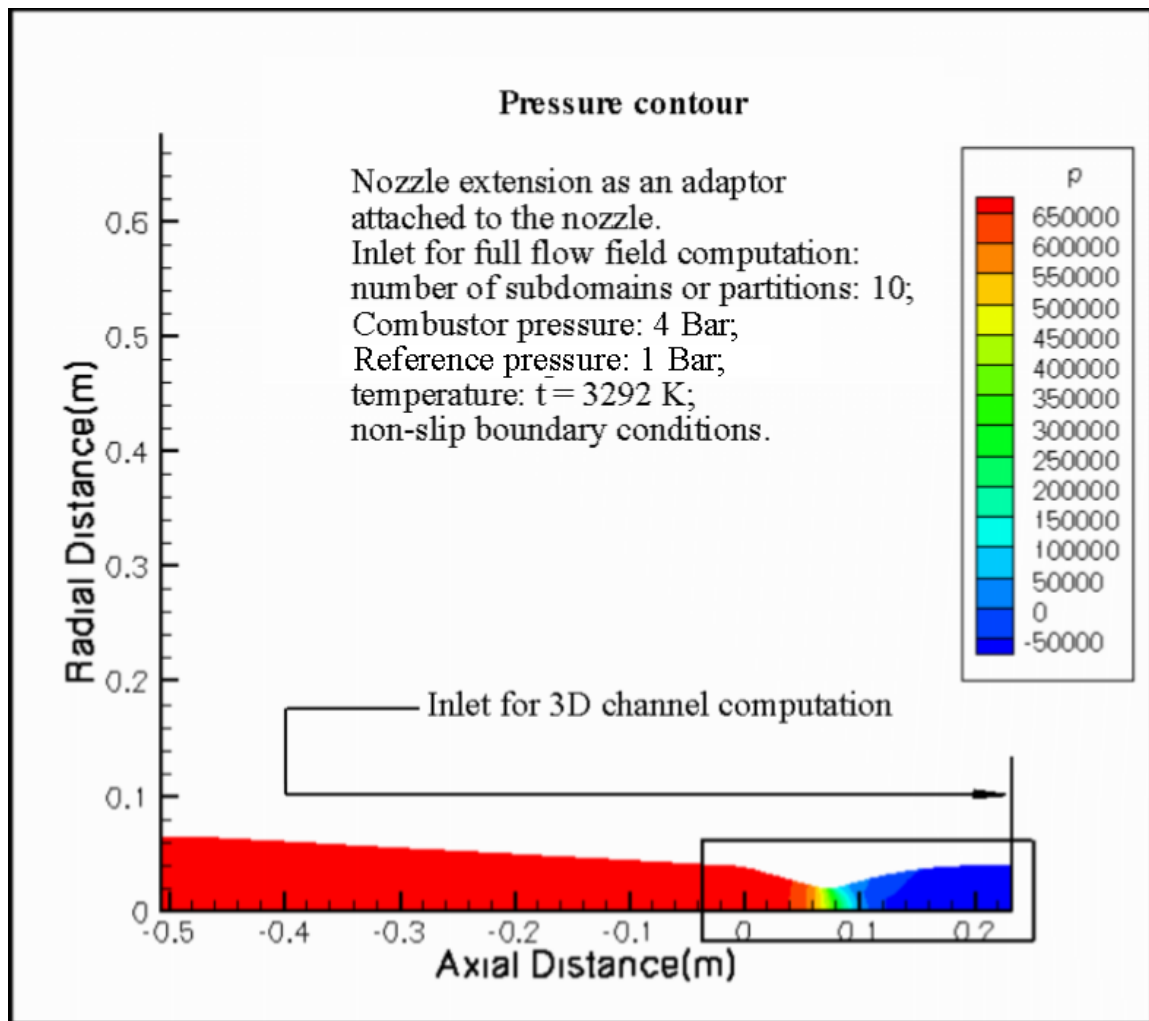
**Figure 4.1.2.** Much number distribution in the supersonic nozzle extension, showing boundary layer growth and thickening at the walls, and some supersonic diffusion near the axis of the extension; magnified view from sector of **Figure 4.1.1**.



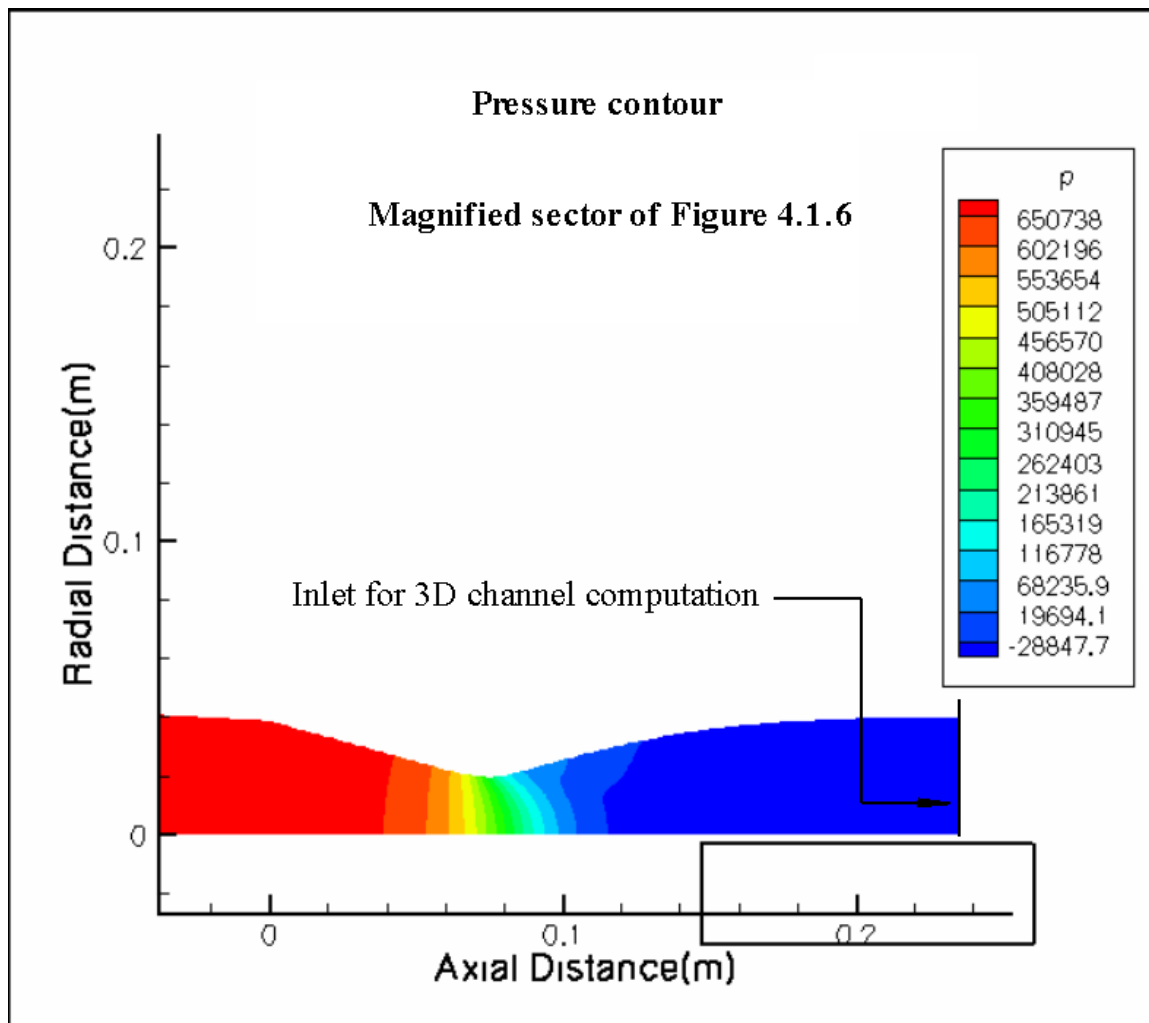
**Figure 4.1.3.** Mach number distribution in combustor outlet and supersonic nozzle without the channel extension, but with the nozzle extension or adaptor, and for reduced chamber pressure.



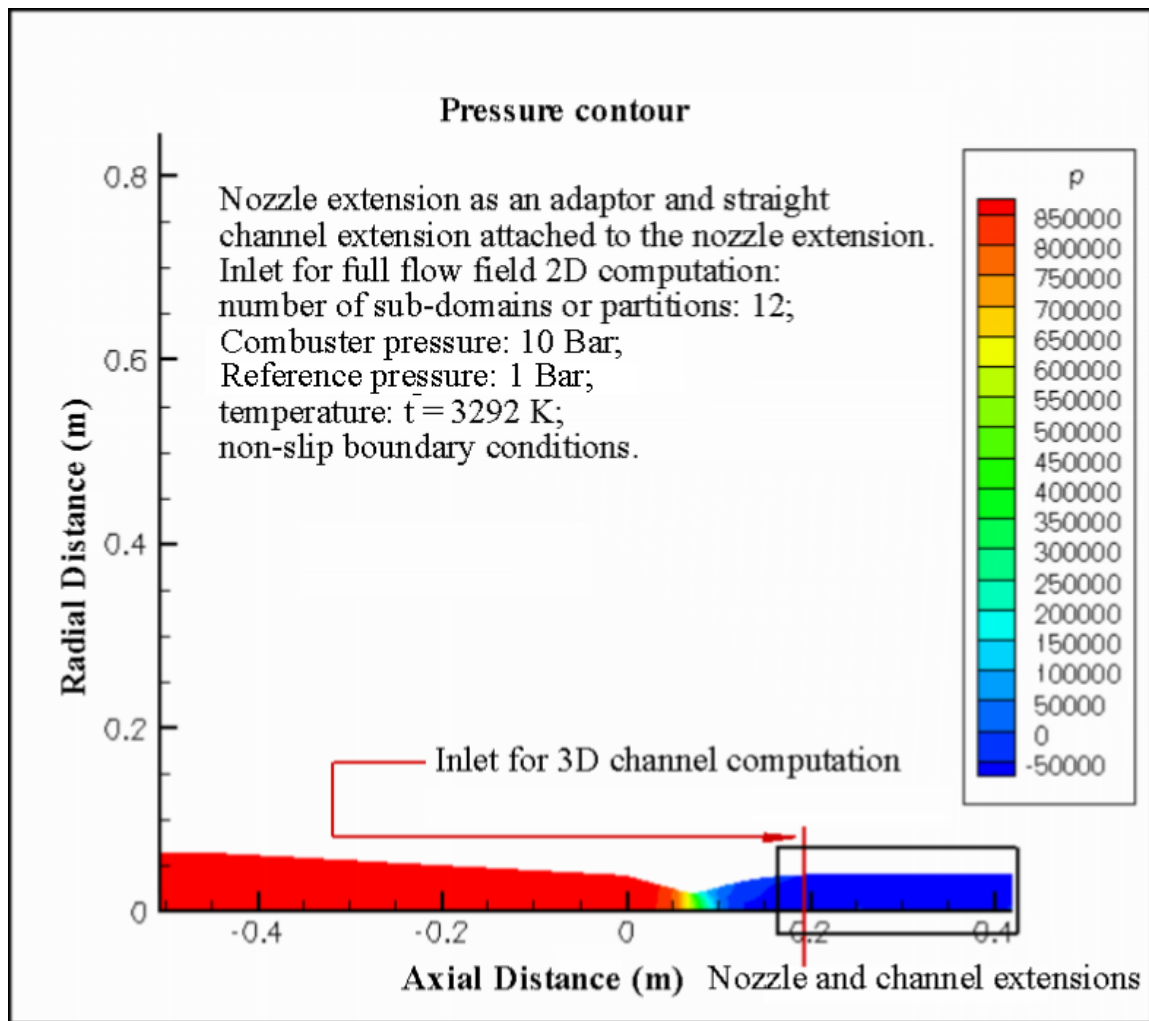
**Figure 4.1.4.** Mach number distribution in the supersonic nozzle without channel extension, but with the nozzle extension (adaptor or supersonic diffuser); magnified sector of **Figure 4.1.3**.



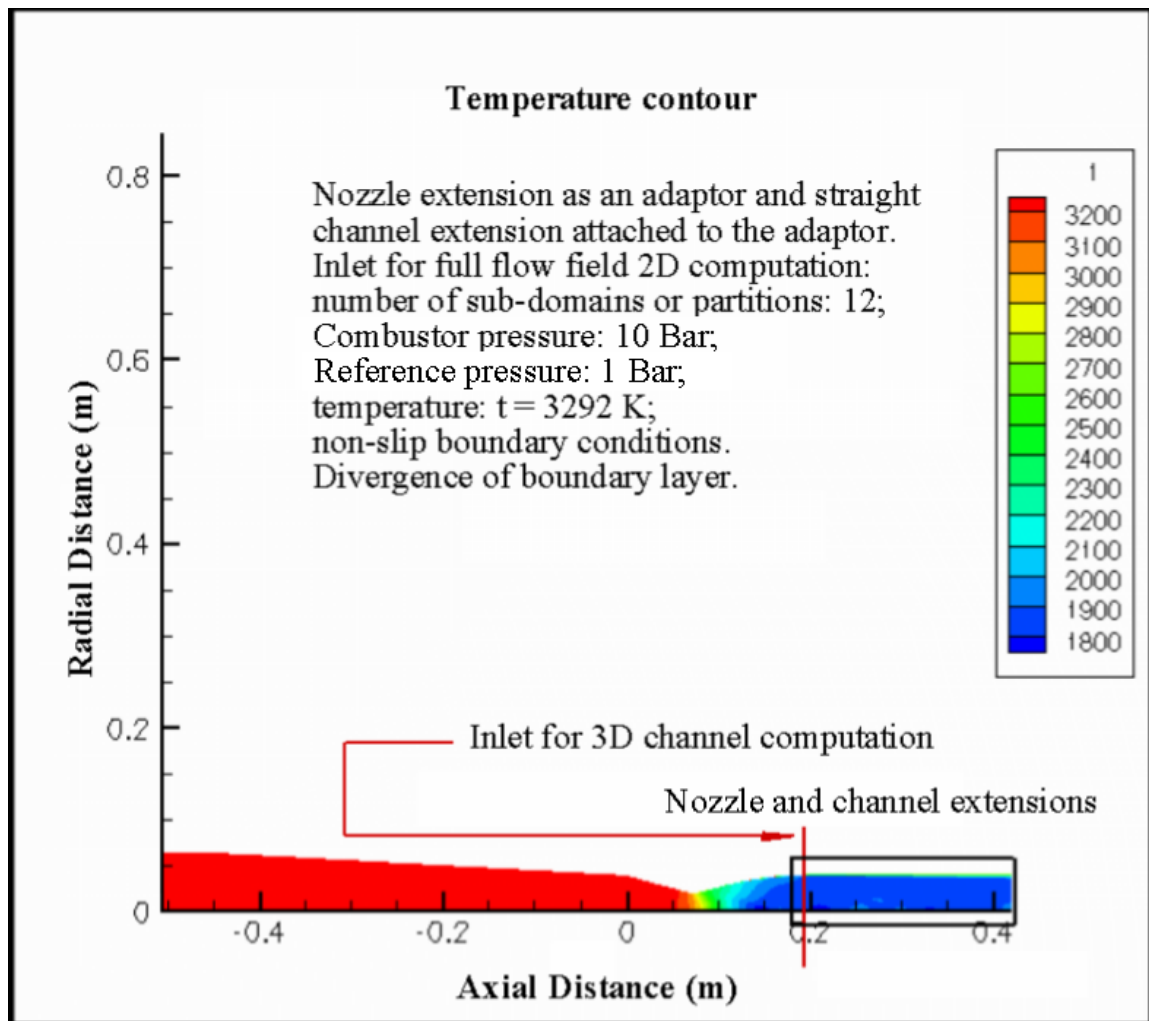
**Figure 4.1.5.** Pressure contour for the combustor and nozzle without channel extension; adaptor is attached to the nozzle. The sector of magnification includes “sharp” or construction angle, throat corridor and the area after the throat.



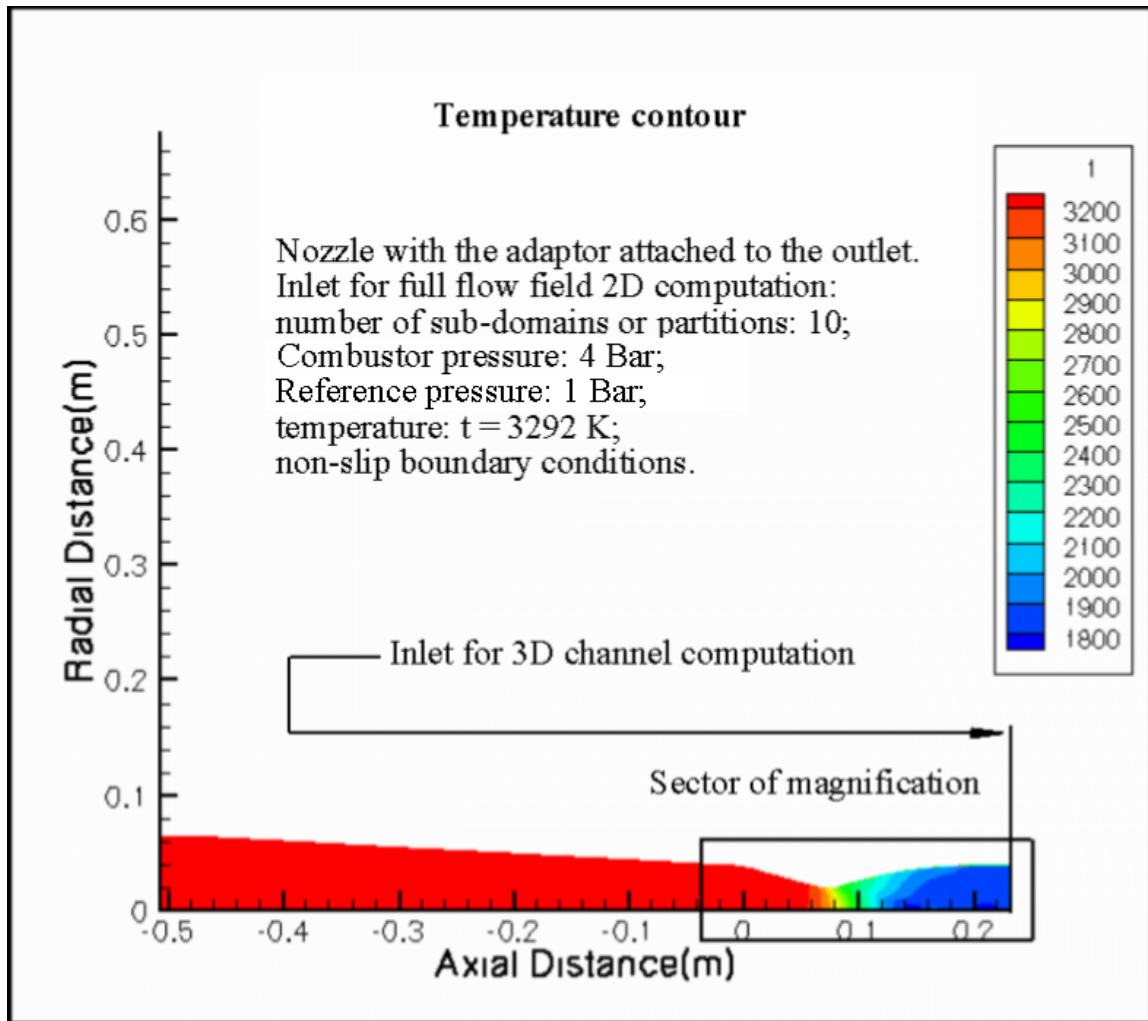
**Figure 4.1.6.** Static pressure distribution in the supersonic nozzle without channel extension, but with the nozzle extension (adaptor or supersonic diffuser), for reduced chamber pressure; magnified sector of **Figure 4.1.5**.



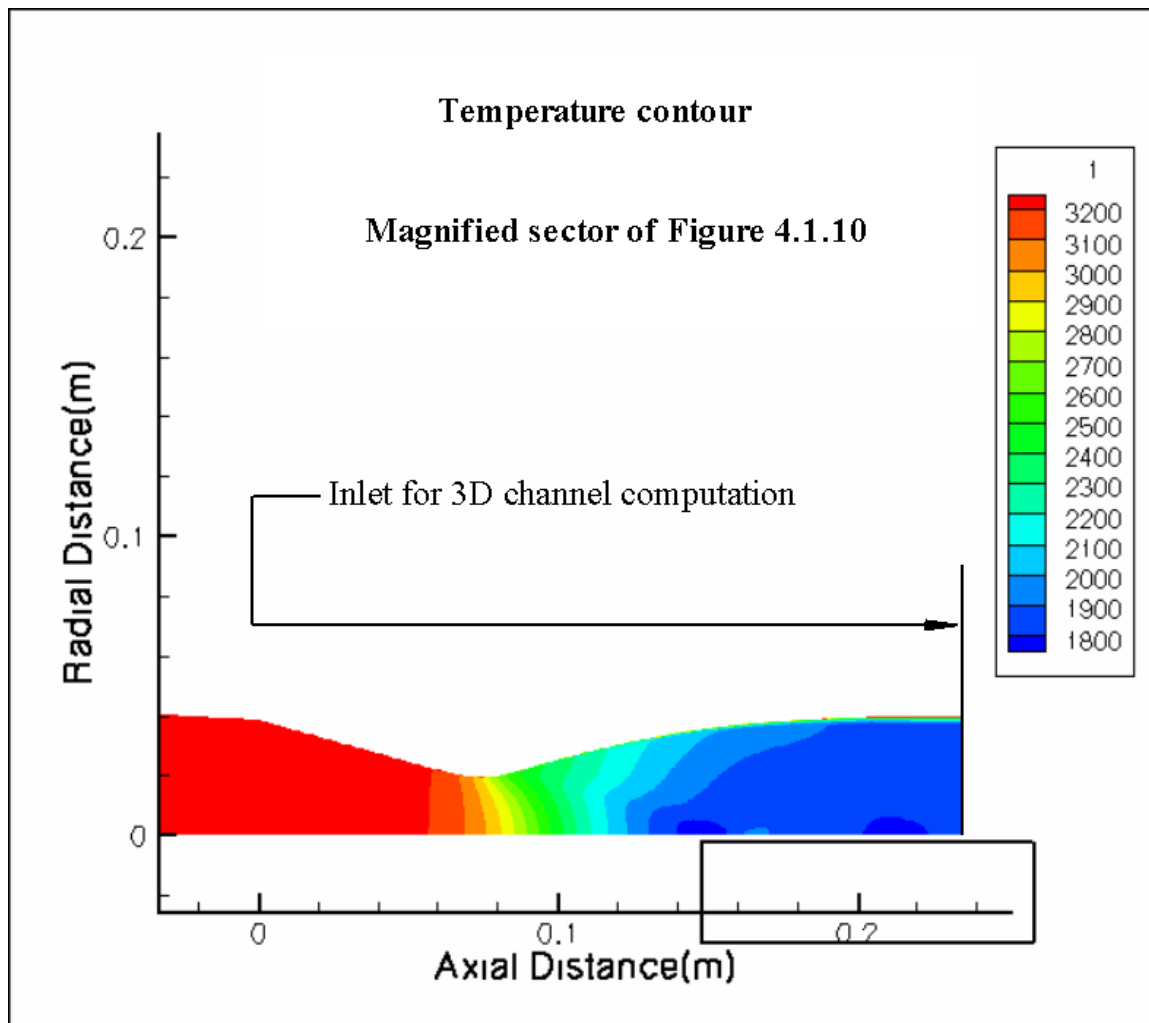
**Figure 4.1.7.** Static pressure distribution in combustor outlet, nozzle, nozzle extension or adaptor and channel extension corresponding to **Figure 4.1.1**.



**Figure 4.1.8.** Static temperature distribution in combustor outlet, nozzle, nozzle extension or adaptor and channel extension.



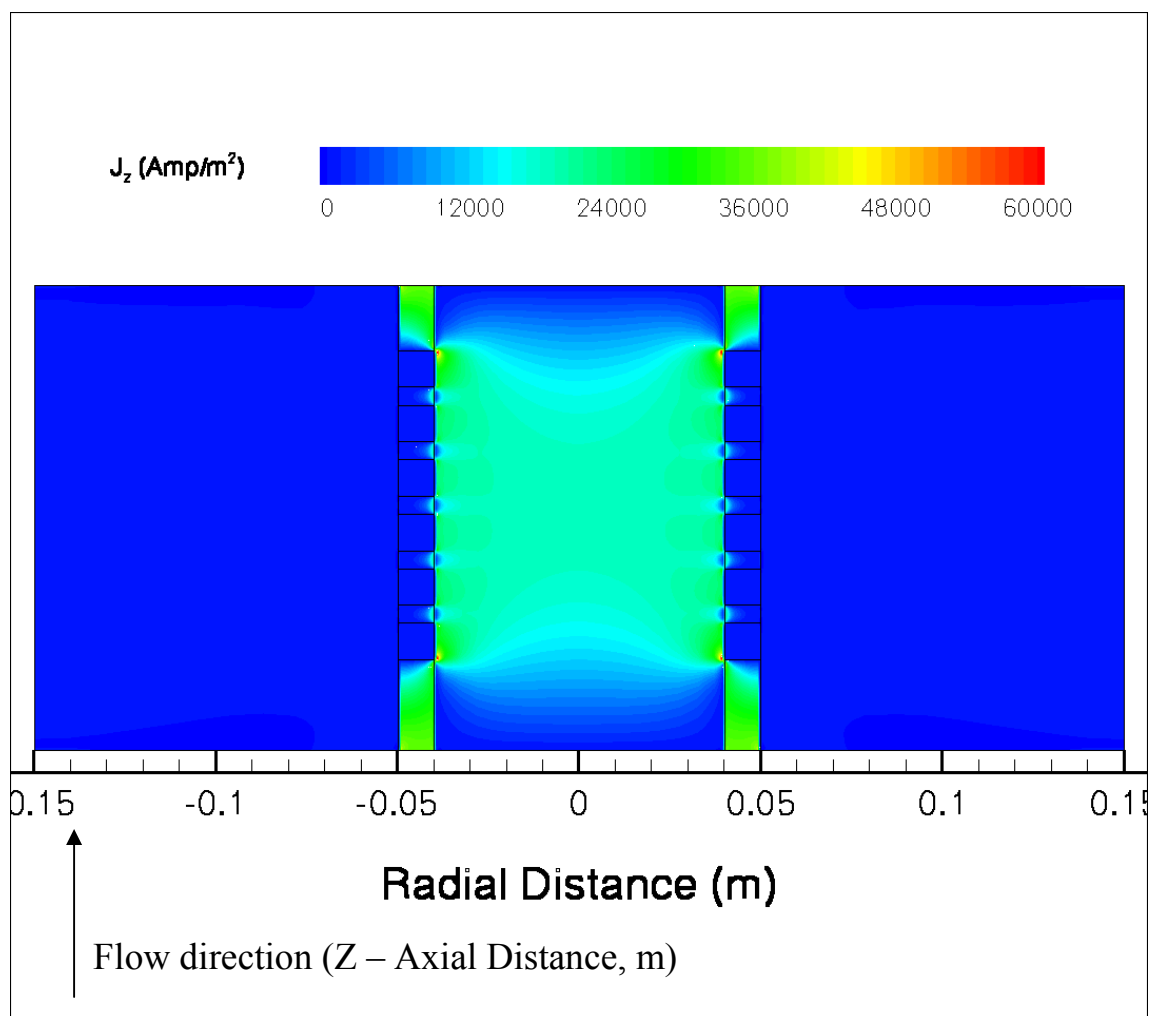
**Figure 4.1.9.** Static temperature distribution in combustor outlet and supersonic nozzle without the channel extension, but with the nozzle extension or adaptor, and for reduced chamber pressure.



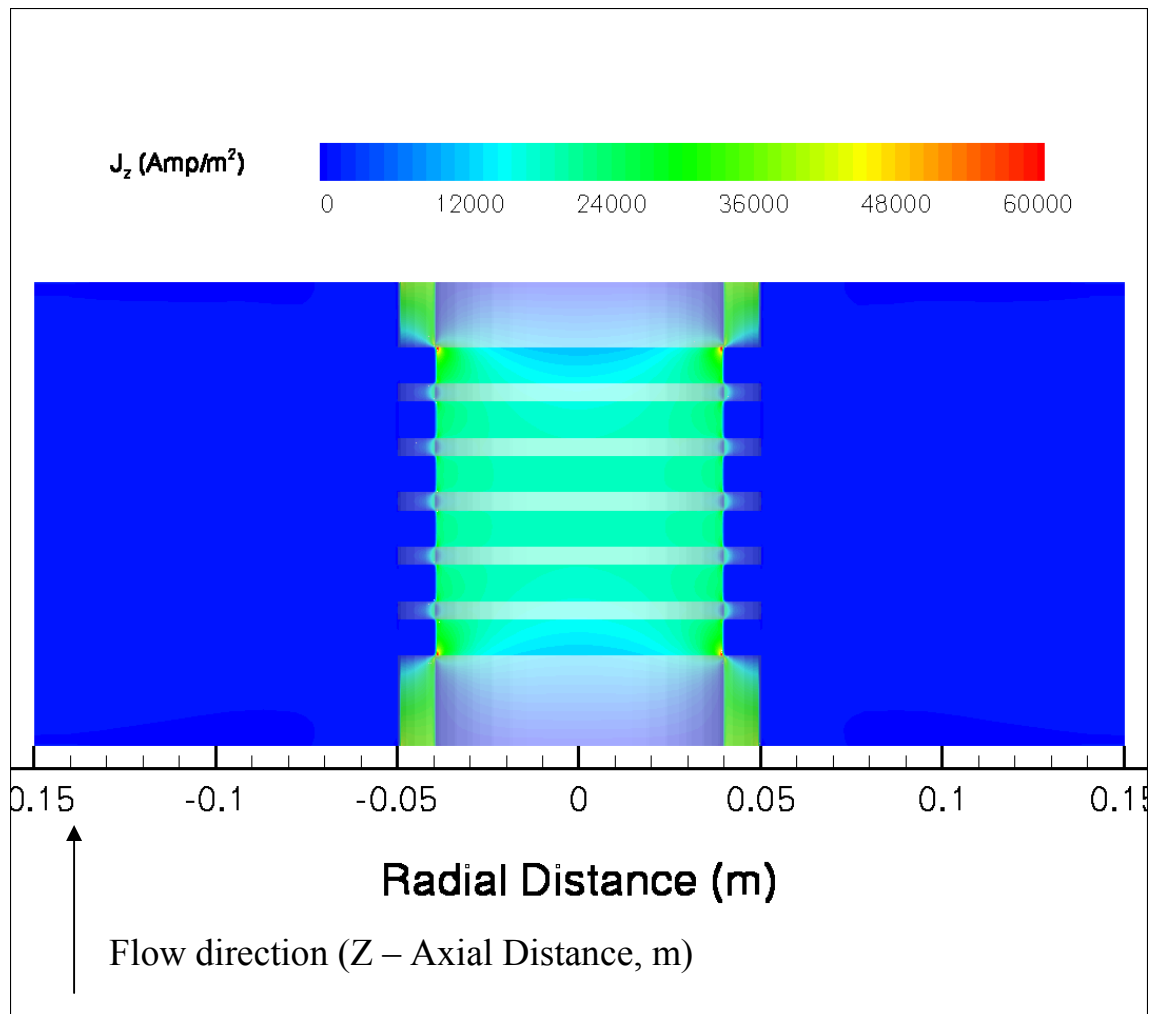
**Figure 4.1.10.** Static temperature distribution in the supersonic nozzle without the channel extension, but with the nozzle extension or adaptor; magnified sector of **Figure 4.1.9**.

## **APPENDIX A7**

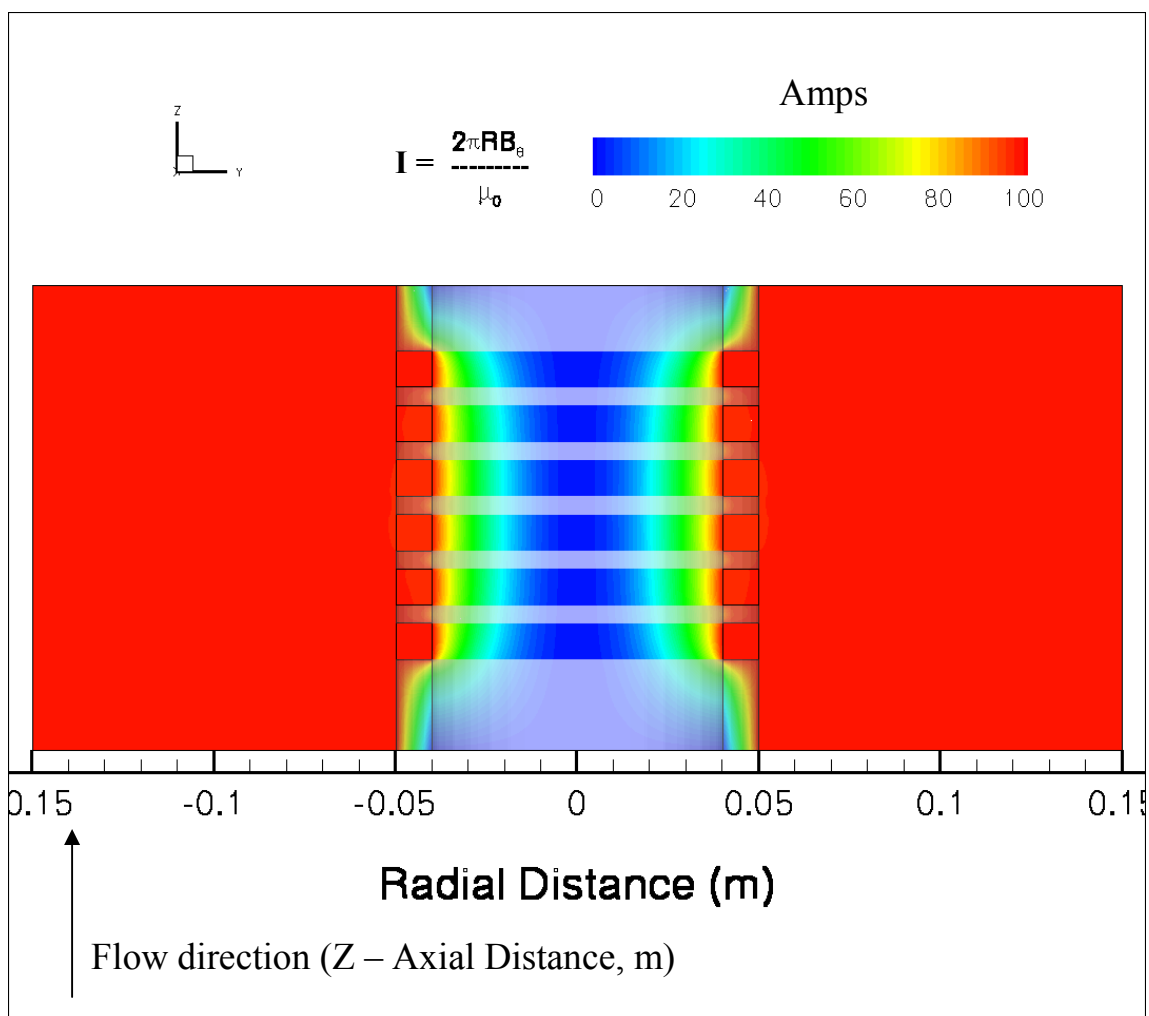
### **FIGURES ILLUSTRATING COMPUTATIONAL DATA FOR STRAIGHT CONDUCTIVITY CHANNEL**



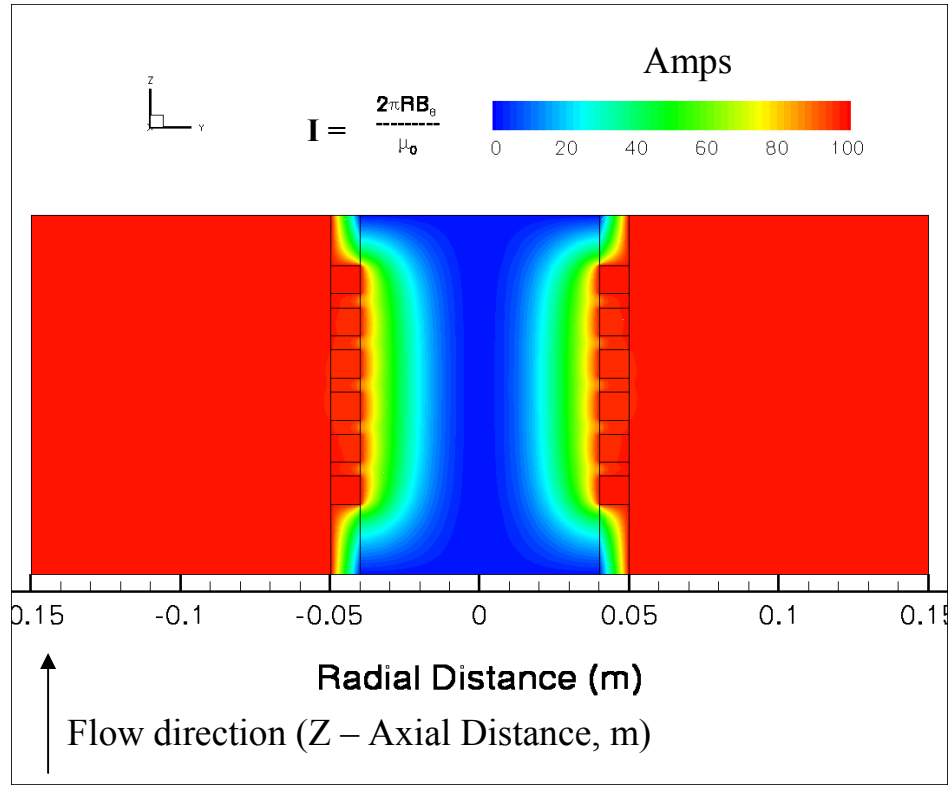
**Figure 4.2.1.** Predicted distribution of electric current density,  $J$ , in a cross-section of the HVEPS conductivity channel.



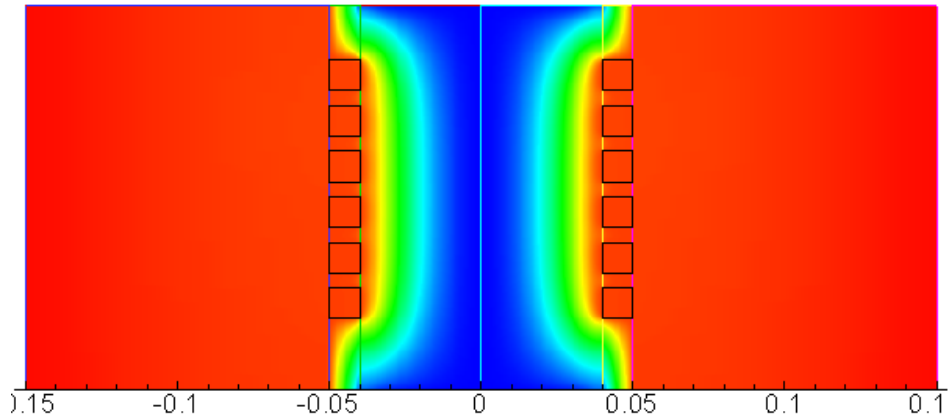
**Figure 4.2.2.** The cross-section of the predicted current density distribution in ZY – plane with the solid zones included.



**Figure 4.2.3.** Distribution of predicted axial electric current levels (contours) in a cross-section of the HVEPS conductivity channel made with GEMS® code.



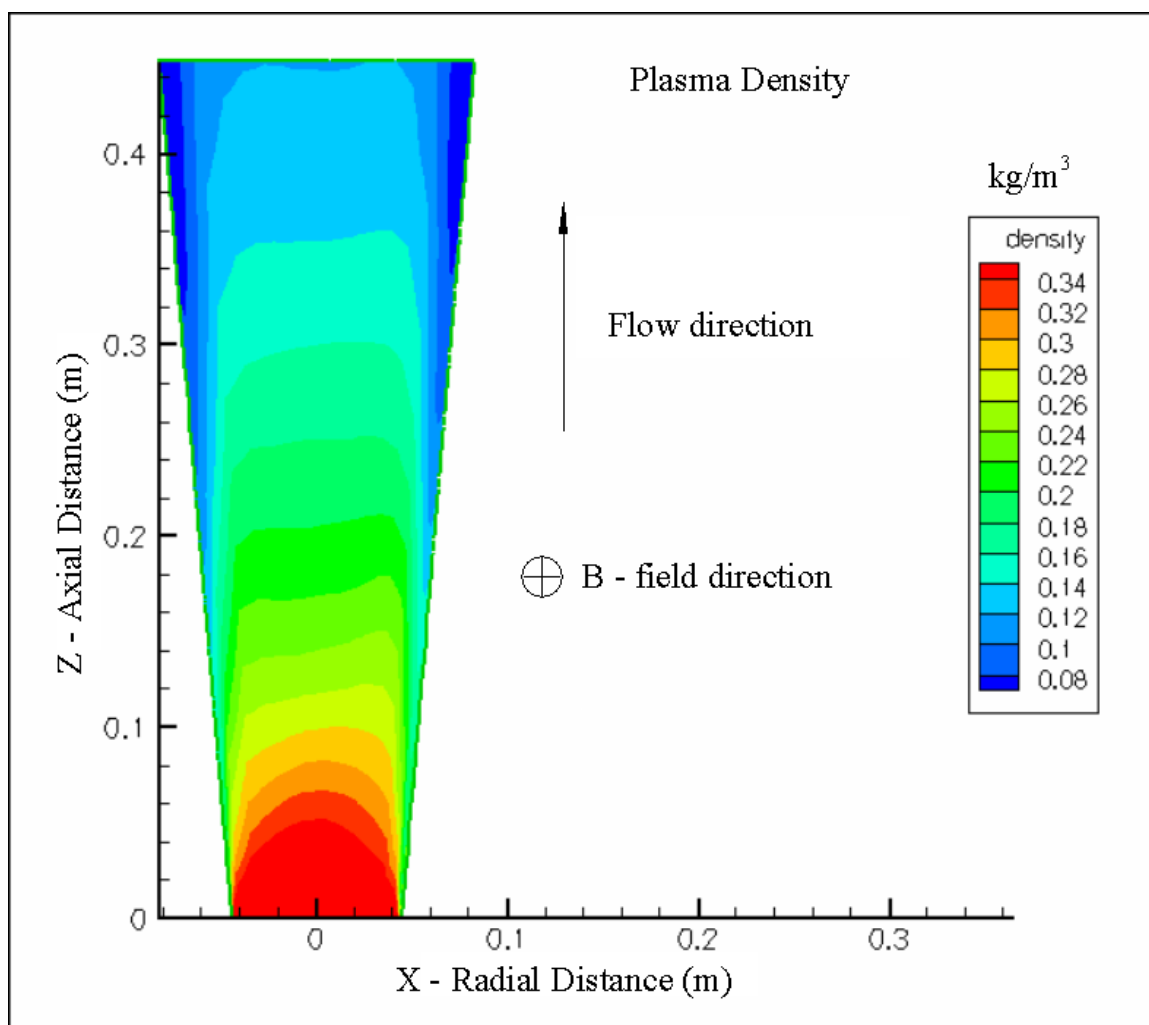
**Figure 4.2.4.** Current level distribution in ZY – plane predicted by the GEMS® code made in 3D.



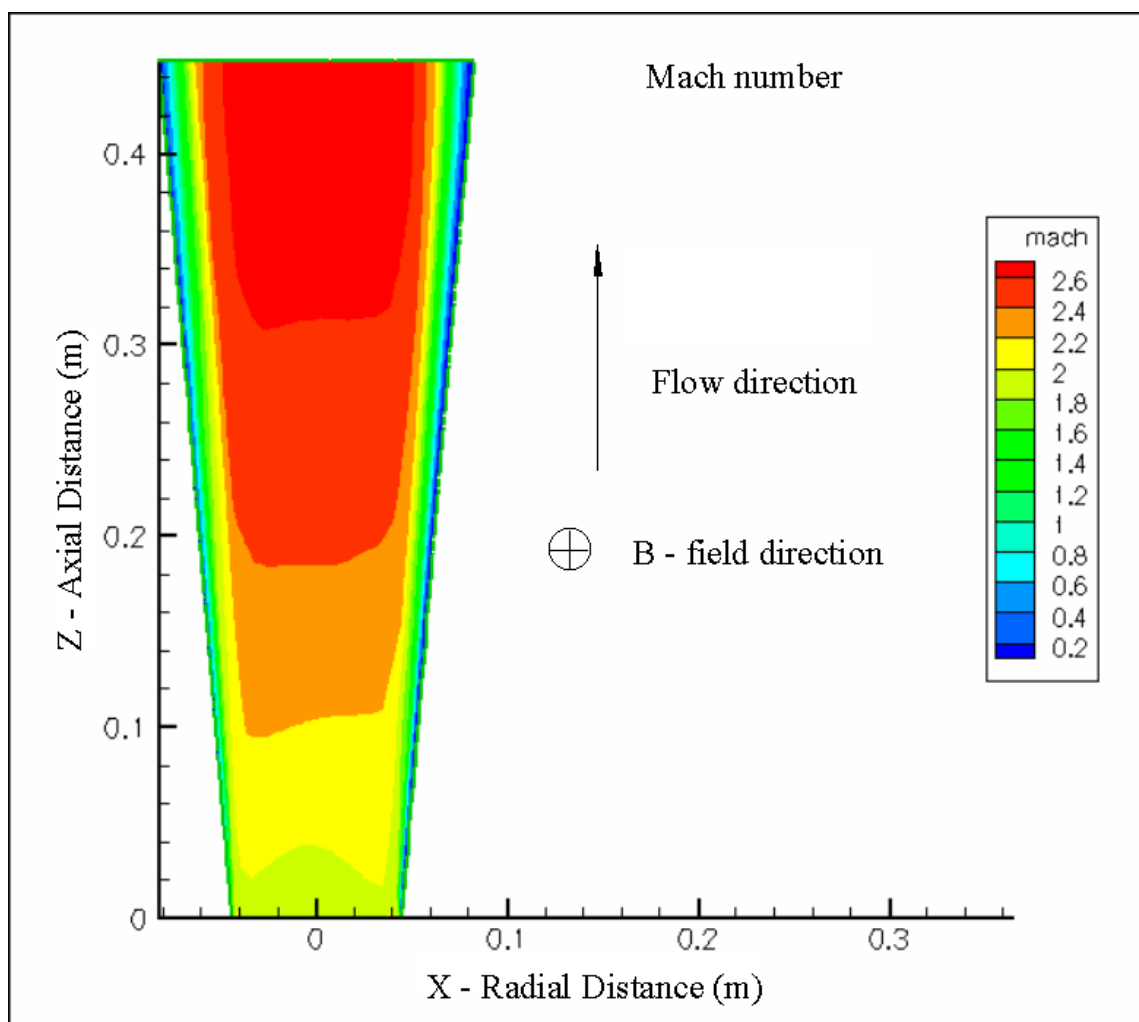
**Figure 4.2.5.** Current level distribution predicted by the MACH2® code made in 2D. Comparison of predicted current level distributions in the cross-section of the HVEPS conductivity channel obtained with GEMS®, **Figure 4.2.4** above.

## **APPENDIX A8**

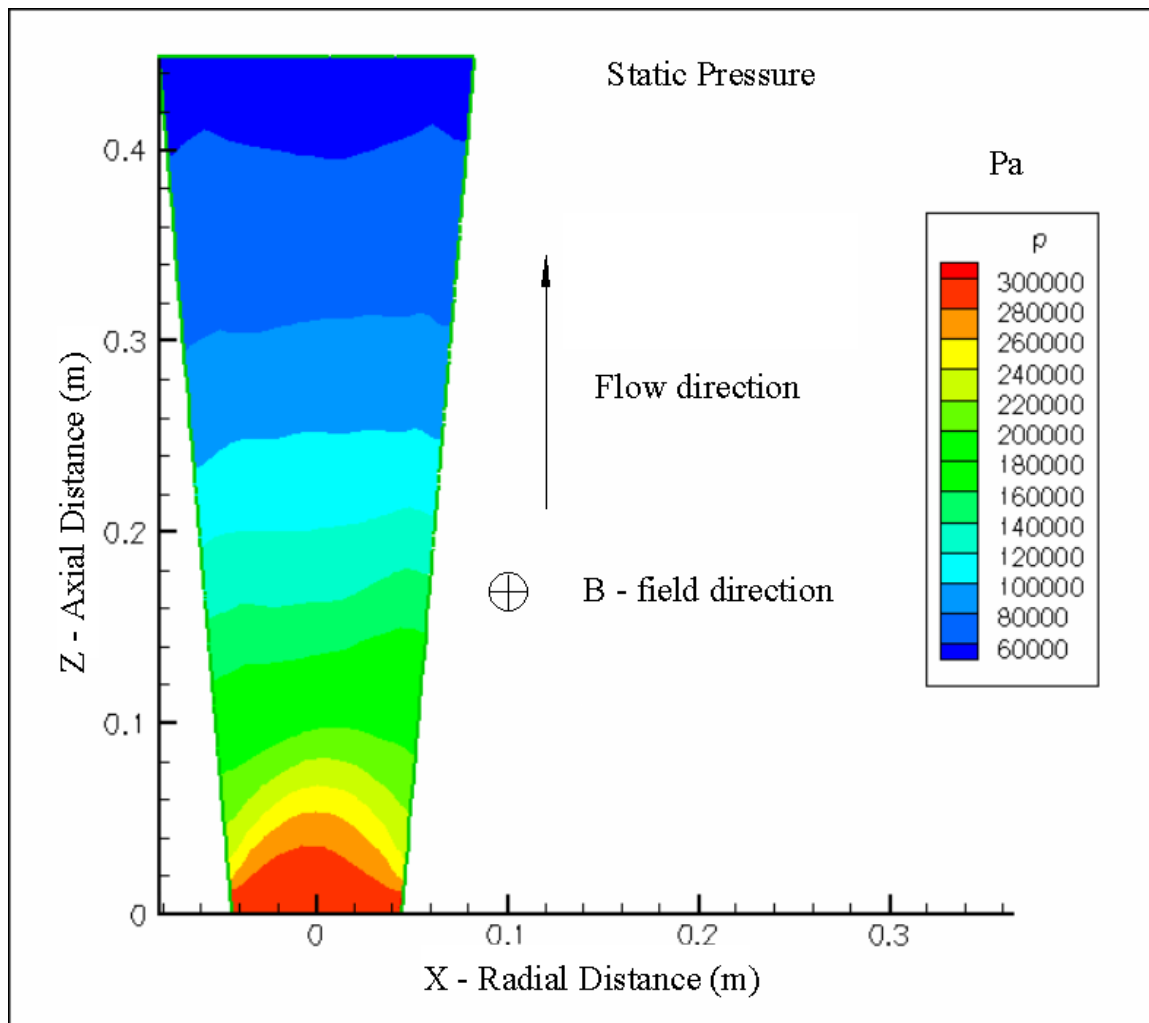
### **FIGURES ILLUSTRATING COMPUTATIONAL DATA FOR DCW MHD CHANNEL**



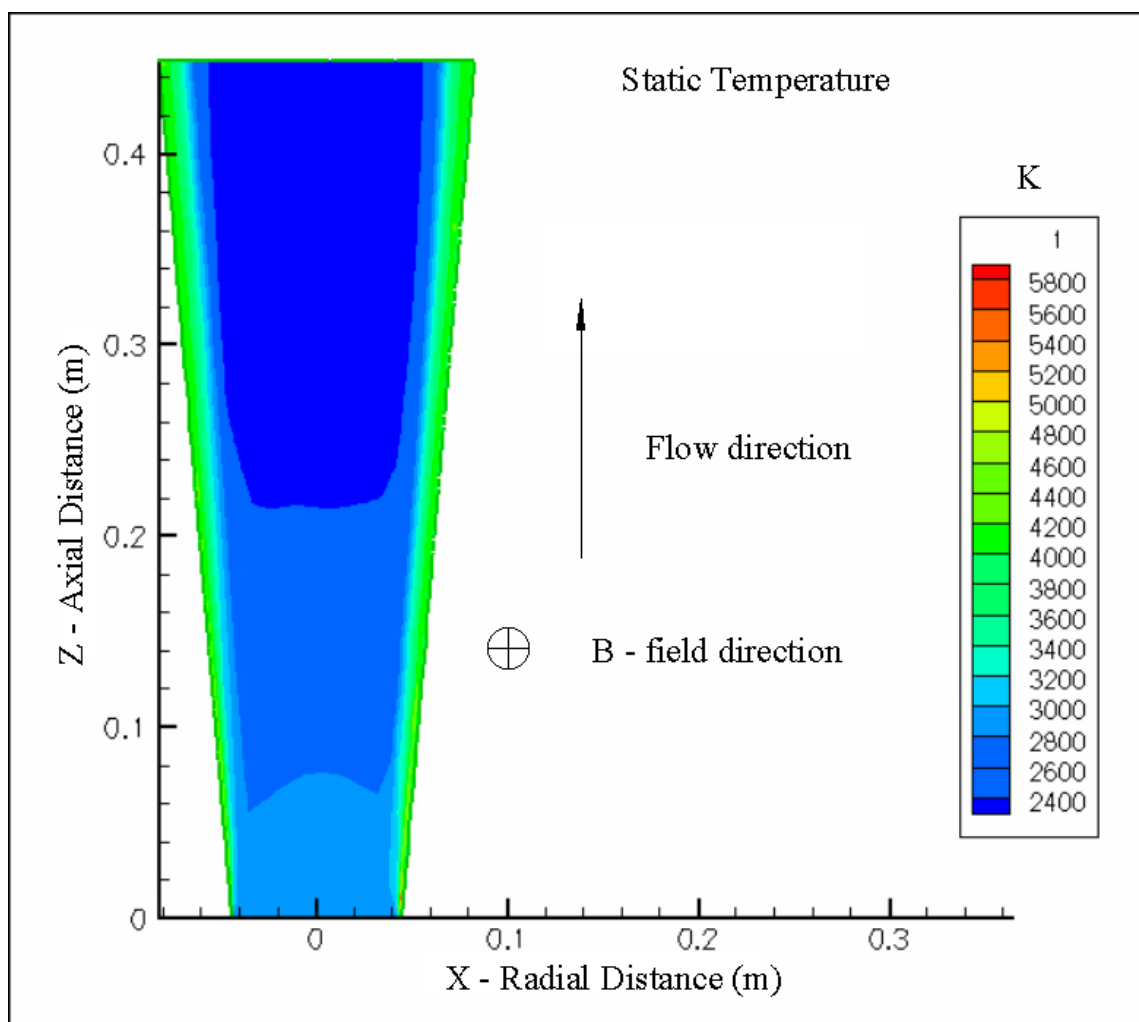
**Figure 4.3.1.** Plasma flow density distribution in the DCW MHD generator made with GEMS® code in 3D.



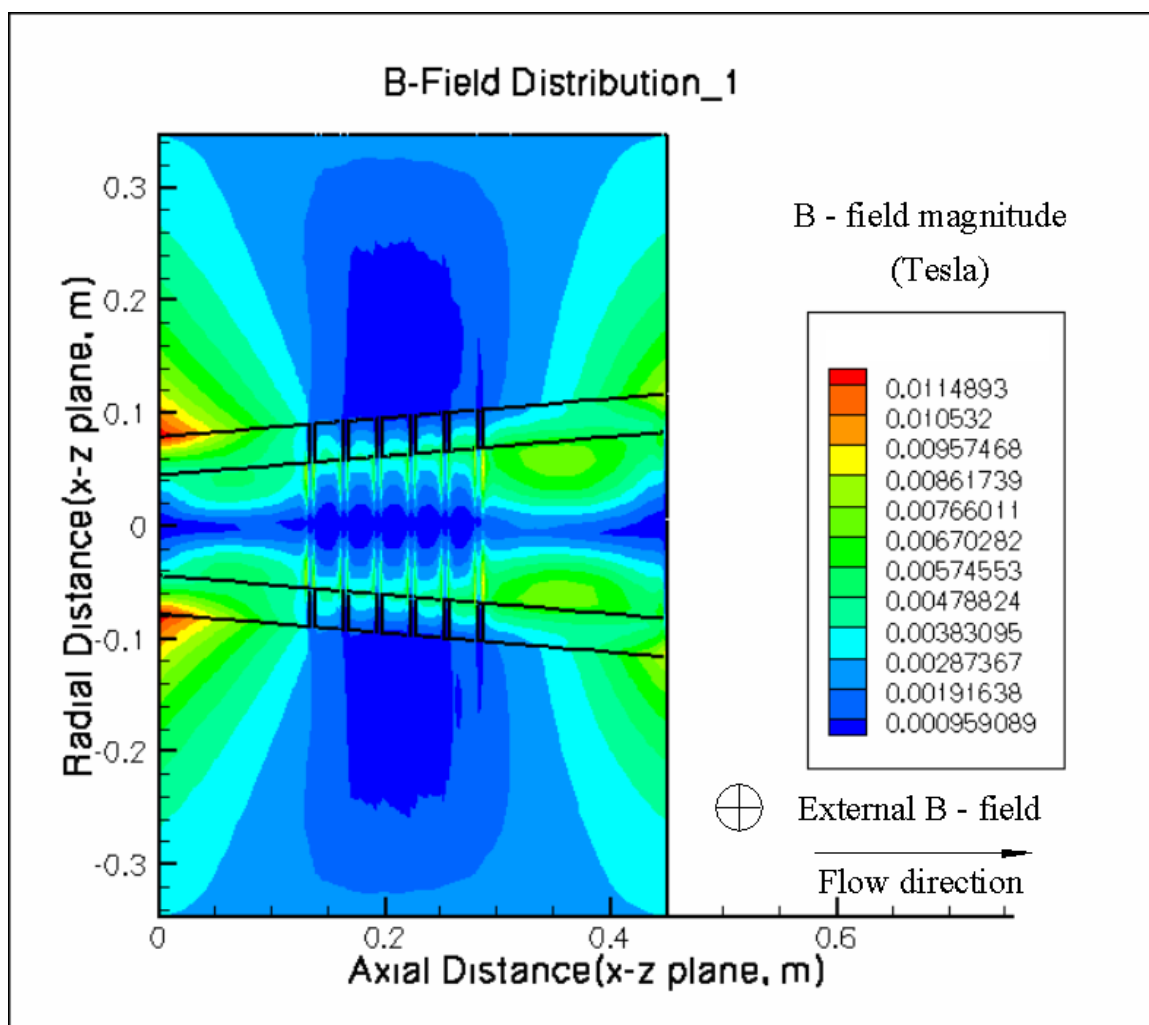
**Figure 4.3.2.** Mach number distribution predicted with the GEMS® code in 3D in the flow field of a diagonal conducting wall, one-slope MHD generator in a 2 Tesla external magnetic field.



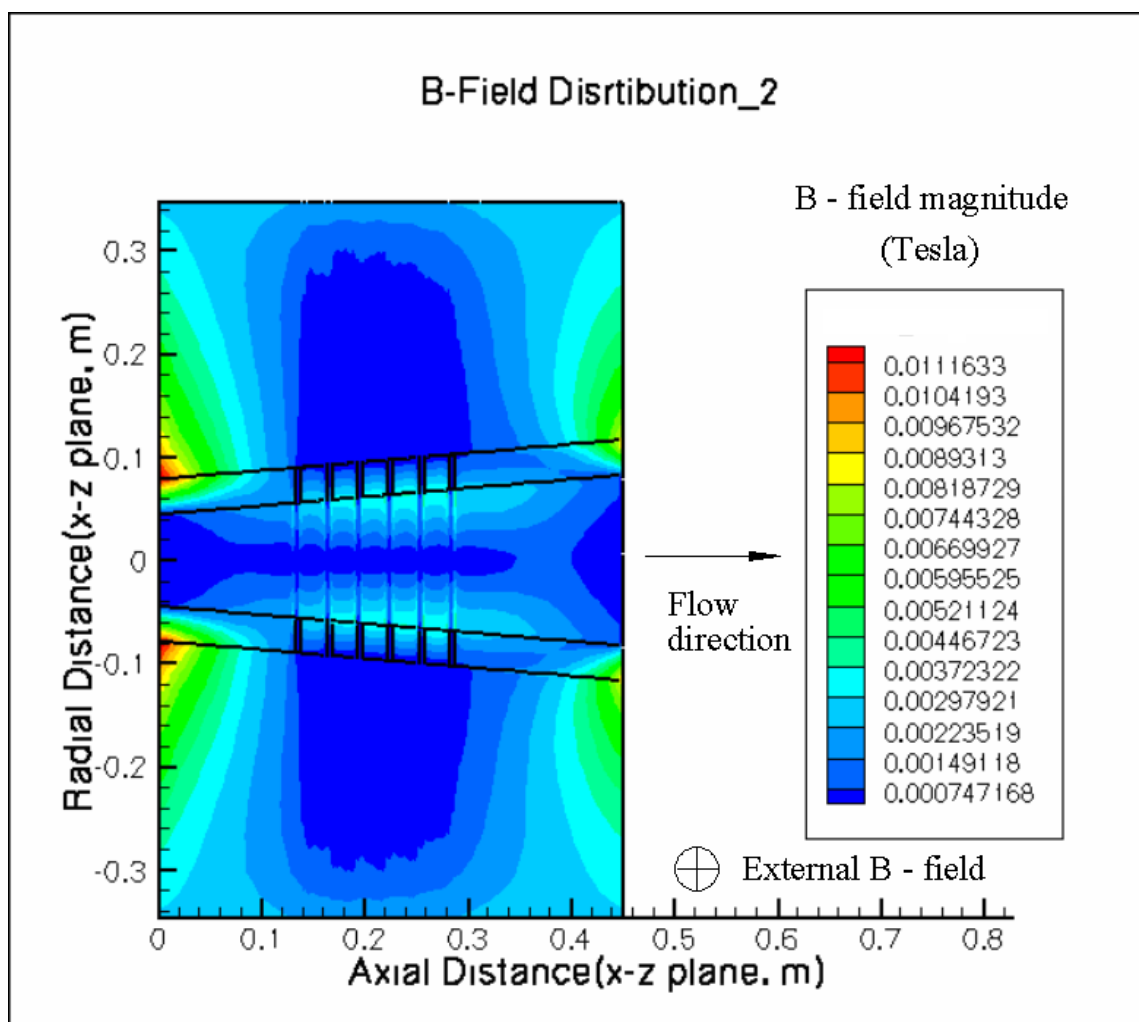
**Figure 4.3.3.** Static pressure distribution predicted with GEMS® code in 3D computations in the flow field of a diagonal conducting wall, one-slope MHD generator in a 2 Tesla external magnetic field.



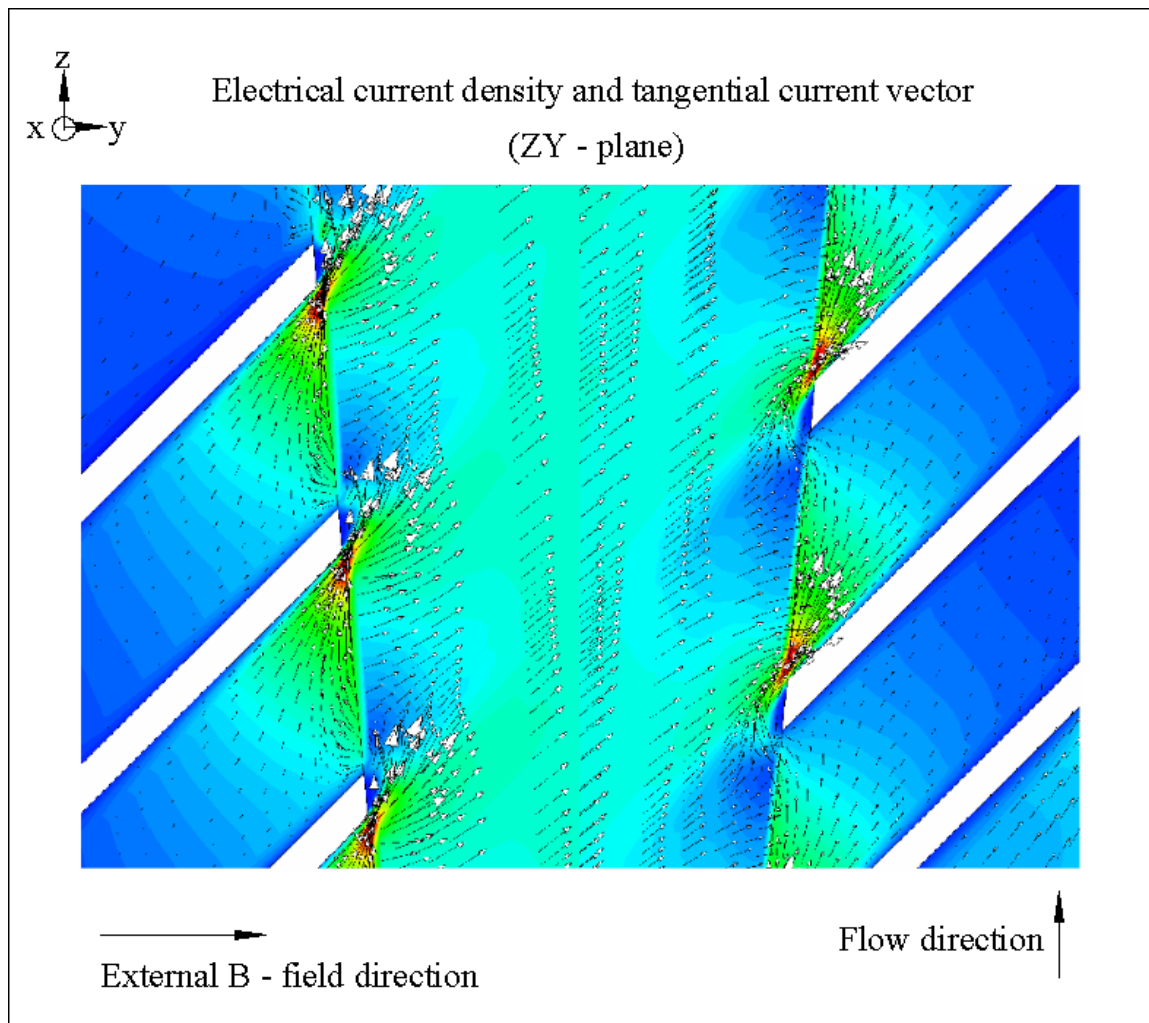
**Figure 4.3.4.** Static temperature distribution predicted with GEMS® code in 3D computations in the flow field of a diagonal conducting wall, one-slope MHD generator in a 2 Tesla external magnetic field.



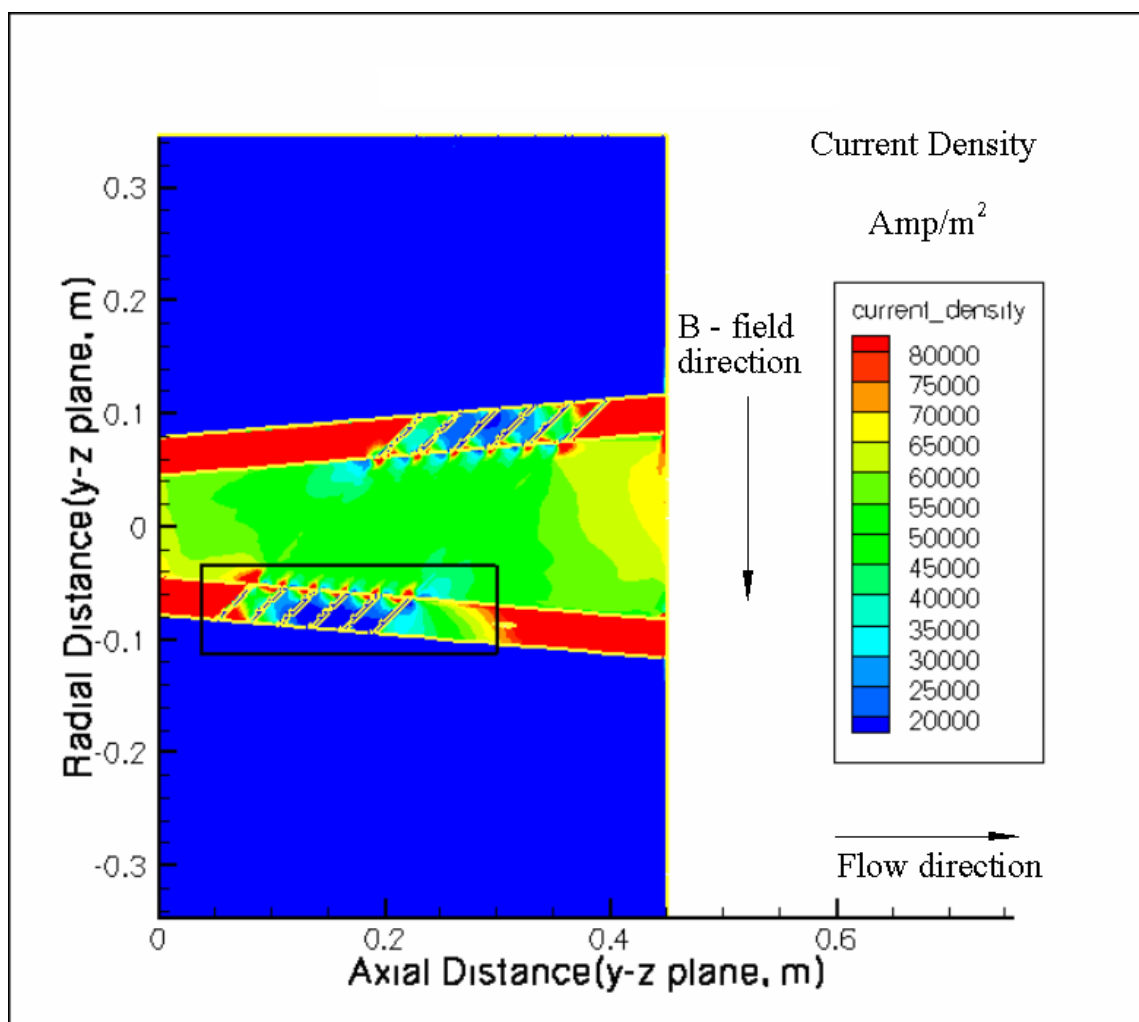
**Figure 4.3.5.** B-field distribution predicted by the GEMS® code in the ZY – plane of the DCW MHD generator with the uniform ramp-up and uniform ramp-down.



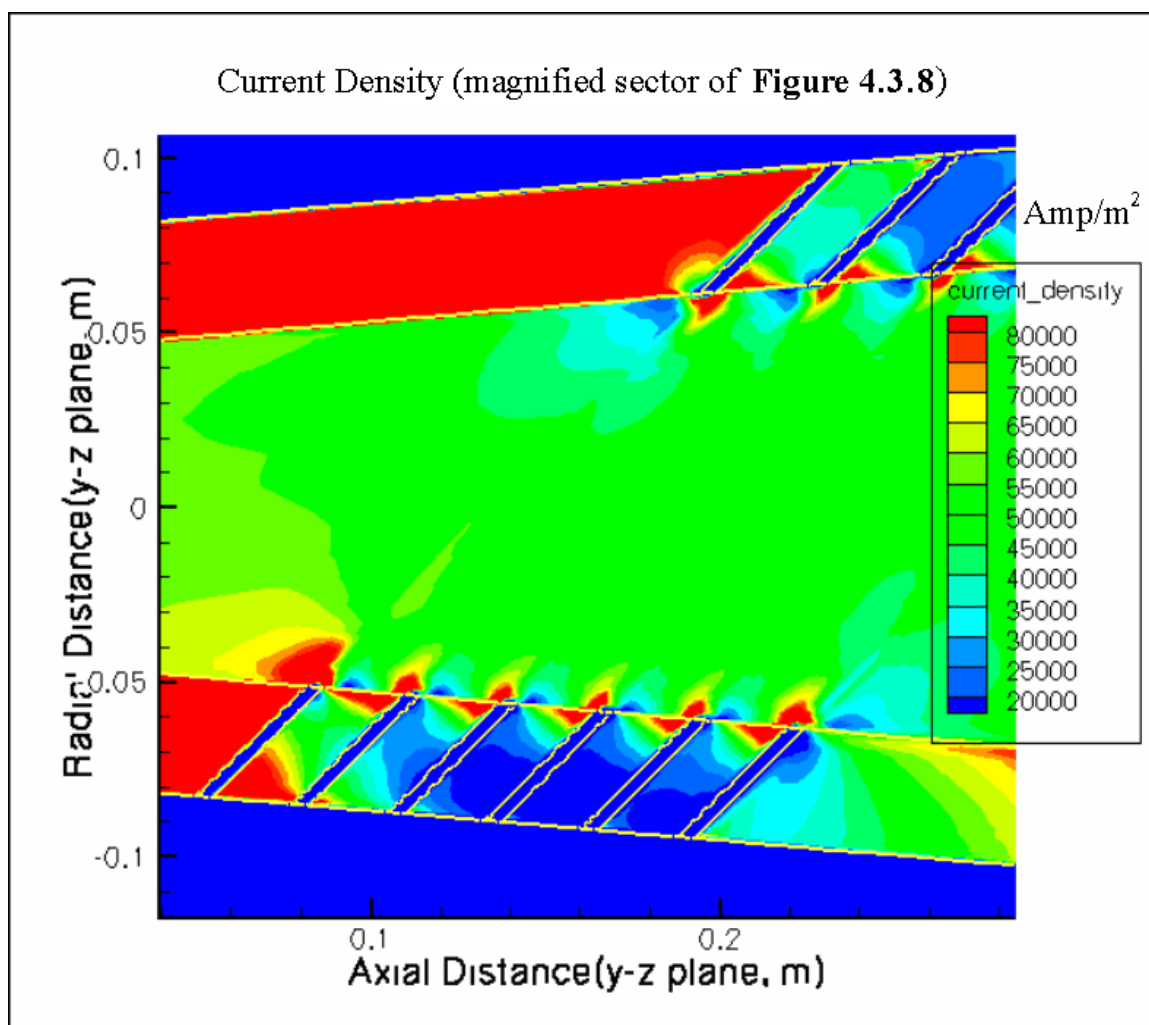
**Figure 4.3.6.** B-field distribution predicted by the GEMS® code in the ZY – plane of the DCW MHD generator with the exponential ramp-up and exponential ramp-down.



**Figure 4.3.7.** Predicted current density levels and tangential component of the current vectors in a DCW MHD channel made with the GEMS® code.



**Figure 4.3.8.** Current density distribution predicted by the GEMS® code in the plasma flow of a DCW MHD channel.



**Figure 4.3.9.** Magnified sector of **Figure 4.3.8**.

## VITA

Konstantin Kolokolnikov was born in New-York City, New-York, and after certain family circumstances he was brought to Moscow City, in the Russian Federation (former Soviet Union), where he received his elementary and high school education with specialization in mathematics. In August 1985 he graduated from the Moscow State Technical University of Electronics and Mathematics (MIEM) with major in Applied Mathematics. After his graduation he began working in Moscow Radio-Technical Institute of Academy of Science as an Engineer-Mathematician.

In 1999 he left the Russian Federation and came back to the U.S.A., his Motherland, and New-York City, New-York. From May, 1999, till September, 2001, he worked as a Research Assistant in a NASA Center for Photonic Materials at FISK University in Nashville, Tennessee. During his work there he continued to improve his knowledge of Physics and began to follow the requirements for an MA in Physics. Konstantin Kolokolnikov accomplished all those requirements with a GPA of 3.95. However, before graduating, he made an application to UTSI for admission to the PhD program in Engineering Science, and was accepted. He left FISK University and began his study as a Graduate Research Assistant at UTSI, working toward the PhD degree.

As a result of a set of circumstances which happened in a period 2003-2004 his PhD program was terminated by the Chair of Mechanical Engineering, Dr. Roy Schulz. In the authors opinion, this was an inappropriate action, which is the position of the author. After this, the author took some of his research work and applied it to obtain a Master of Science degree in Engineering Science. The topic of his Mater's thesis was a necessary and organic part of his PhD dissertation.