

Graphical User Interface for High Energy Multi-Particle Transport

**Final Report
November 30th 2008**

PREPARED BY:

Visual Editor Consultants

P.O. Box 1308

Richland, WA 99352-1308

PHONE: (509) 539-8621

FAX: (509) 946-2001

Email: randyschwarz@mcnpvised.com

Home Page: <http://www.mcnpvised.com>

Contract Number: NNL07AA10C

November 2008

Project Summary

Computer codes such as MCNPX now have the capability to transport most high-energy particle types (34 particle types now supported in MCNPX) with energies extending into the teravolt energy range. The efficient use of these types of Monte Carlo tools is very important for modeling the effects of space radiation on humans, spacecraft, and equipment. With this innovation, users of the MCNPX code will have access to a powerful graphical user interface for efficient creation and interrogation of their input files, which will significantly reduce the amount of time required to create and debug input files.

Table of Contents

	<u>PAGE</u>
PROJECT SUMMARY.....	II
TABLE OF CONTENTS	III
2 IDENTIFICATION AND SIGNIFICANCE OF THE INNOVATION	1
3 TECHNICAL OBJECTIVES	1
4 WORK PLAN	2
4.1 TECHNICAL APPROACH.....	2
4.2 TASK DESCRIPTIONS	2
4.2.1 <i>Task 1: Incorporate the Los Alamos Quark-Gluon String Model Module into the MCNPX Code</i>	<i>2</i>
4.2.2 <i>Task 2: Update the MCNPX Visual Editor to the Latest Version of MCNPX...3</i>	<i>3</i>
4.2.3 <i>Task 3: Add Support for Multiple Particles</i>	<i>3</i>
4.2.4 <i>Task 4: Add Source Creation</i>	<i>4</i>
4.2.5 <i>Task 5: Investigate the Implementation of Spline Surfaces in MCNPX.....4</i>	<i>4</i>
4.2.6 <i>Task 6: Implement Data Visualization</i>	<i>4</i>
4.2.7 <i>Task 7: Improve 3D Visualization Tools.....5</i>	<i>5</i>
4.2.8 <i>Task 8: General Enhancements to MCNPX Visual Editor.....5</i>	<i>5</i>
4.2.9 <i>Task 9: Create a Set of Test Problems.....6</i>	<i>6</i>
4.2.10 <i>Task 10: Documentation.....6</i>	<i>6</i>
4.3 MEETING THE TECHNICAL OBJECTIVES	6
4.4 TASK LABOR CATEGORIES AND SCHEDULES	6
5 POTENTIAL APPLICATIONS.....	7
5.1 POTENTIAL NASA APPLICATIONS	7
5.2 POTENTIAL NON-NASA APPLICATIONS	7
6 CONTACTS	8
6.1 KEY CONTRACTOR PARTICIPANTS	8
6.2 KEY NASA PARTICIPANTS	8
6.3 NASA AND NON-NASA ADVISORS	9
7 TECHNICAL ACTIVITIES	9
7.1 RECENT TECHNICAL ACTIVITIES	9
7.1.1 <i>Task 1: Incorporate the Los Alamos Quark-Gluon String Model Module into the MCNPX Code</i>	<i>9</i>
7.1.2 <i>Task 2: Update the MCNPX Visual Editor to the Latest Version of MCNPX..17</i>	<i>17</i>
7.1.3 <i>Task 3: Add Support for Multiple Particles</i>	<i>17</i>
7.1.4 <i>Task 4: Add Source Creation</i>	<i>35</i>
7.1.5 <i>Task 5: Investigate the Implementation of Spline Surfaces in MCNPX.....52</i>	<i>52</i>
7.1.6 <i>Task 6: Implement Data Visualization</i>	<i>76</i>
7.1.7 <i>Task 7: Improve 3D Visualization Tools.....90</i>	<i>90</i>
7.1.8 <i>Task 8: General Enhancements to MCNPX Visual Editor.....95</i>	<i>95</i>

7.1.9 Task 9: Create a Set of Test Problems.....95

List of Figures

	<u>PAGE</u>
Figure 1. Carbon Beam Therapy Problem.	15
Figure 2. Iron Ions on a Water Phantom.	16
Figure 3. Uranium Isotopes on Lithium.	17
Figure 4. Particle Track Plotting Panel in the MCNPX Visual Editor.	19
Figure 5. Plotting the Source in the MCNPX Visual Editor.	20
Figure 6. Plotting the Source in the MCNPX Visual Editor with Equal Extents.	21
Figure 7. Plotting Collision Points.	22
Figure 8. Plotting Collision Points with Different Colors.	23
Figure 9. Plot of Neutron Collision Points.	24
Figure 10. Plot of Photon Collision Points.	25
Figure 11. Plot of Muon Collision Points.	25
Figure 12. Plot of Pion Collision Points.	26
Figure 13. Plot of Neutral Pion Collision Points.	26
Figure 14. Plot of Deuteron Collision Points.	27
Figure 15. Plot of Triton Collision Points.	27
Figure 16. Plot of He-3 Collision Points.	28
Figure 17. Plot of He-4 Collision Points.	28
Figure 18. Plot of He-4 Collision Points Including Tracks.	29
Figure 19. Particles Colored by Weight in a Cask.	30
Figure 20. Particles that Contribute to a Tally Outside the Cask.	31
Figure 21. Plot of High Weight Particles Contributing to a Tally Region.	32
Figure 22. Plot of Biasing Particles Toward an Air Gap in a Cask.	33
Figure 23. Plot of Particles that Contribute to the Tally Region.	34
Figure 24. Source Reading and Display Capabilities.	36
Figure 25. Dependent Distributions.	37
Figure 26. Creation of a Source.	38
Figure 27. (x,y) View of g4 Surface for Sphere Flattened Some in + and - x, Input File ig4ca. Or=(0,0,0), ext=11	56
Figure 28. (x,y) View of g4 Surface for Sphere Flattened Some in +x, Input File ig4da. Or=(0,0,0), ext=11	57
Figure 29. (x,y) View of g4 Surface for Sphere Flattened Some in +x, Input File ig4da Plotted with Circular Source. Or=(0,0,0), ext=11	58
Figure 30. (y,z) View of g4 Surface for x-Torus Input File ig4tx. Or=(0,0,0), ext=100	59
Figure 31. View of g4 Surface for Torus Input File ig4txr with Slanted Axis. Or=(0,0,0), ext=100, b=(.707,-.707,0,0,0,1)	60
Figure 32. (x,y) View of g4 an Approximate Cube that is Nearly Flat on the Edges with Rounded Corners, Input File ig44a Plotted with Circular Source. Or=(0,0,0), ext=100	61
Figure 33. (x,y) View of G4 Cylinder with Radius of 80 cm and Negligible 4 th Order Terms. ..	62
Figure 34. (x,y) View of G4 with Axis Crossings of 80 cm and Small Positive 4 th Order Terms.	63

Figure 35. (x,y) View of G4 with Axis Crossings of 80 cm and Moderate Positive 4 th Order Terms.	64
Figure 36. (x,y) View of G4 with Axis Crossings of 80 cm and Only Positive 4 th Order Terms [second order terms are zero].	65
Figure 37. (x,y) View of G4 with Axis Crossings of 80 cm and Negative 2 nd Order Terms [4 th order terms increase more].	66
Figure 38. (x,y) View of G4 with Axis Crossings of 80 cm and Increasing 2 nd Order Terms [4 th order terms are negative].	67
Figure 39. (x,y) View of G4 with Axis Crossings of 80 cm and Moderate Positive 4 th Order Terms Along with Small Negative x^2y^2 Term.	68
Figure 40. (x,y) View of G4 with Axis Crossings of 80 cm and Moderate Positive 4 th Order Terms Along with Small Positive x^2y^2 Term.	69
Figure 41. (x,y) View of G4 with Axis Crossings of 80 cm and Moderate Positive 4 th Order Terms Along with Larger Positive x^2y^2 Term.	70
Figure 42. (x,y) View of GQ Elliptical Cylinder with Major x-Radius of 80 cm and Minor y-Radius of 40 cm.	71
Figure 43. (x,y) View of G4 Elliptical Cylinder with Major x-Radius of 80 cm and Minor y-Radius of 40 cm Using Negligible 4 th Order Terms.	72
Figure 44. (x,y) View of G4 Elliptical Cylinder [with Major x-Radius of 80 cm and Minor y-Radius of 40 cm] Turned into a Near Rectangle by Using Only 4 th Order Terms.	73
Figure 45. (x,y) View of G4 Slanted Plane Surface Using Terms (x, x^2, x^3, x^4, y)	74
Figure 46. Updated Tally Plotting Panel.	76
Figure 47. Run-Time Tally Plotting.	77
Figure 48. Mesh Tally Plot in the MCNPX Visual Editor.	78
Figure 49. Mesh Tally Plot on Top of the Geometry.	79
Figure 50. Zoomed in Region of the Mesh Tally Plot.	80
Figure 51. MCNPX Contour Plot.	81
Figure 52. Visual Editor Contour Plot.	81
Figure 53. MCNPX Contour Plot with a Fixed Variable.	82
Figure 54. Visual Editor Contour Plot with a Fixed Variable.	82
Figure 55. MCNPX Mesh Tally Plot.	83
Figure 56. Visual Editor Mesh Tally Plot.	83
Figure 57. MCNPX Mesh Tally Plot.	84
Figure 58. Visual Editor Mesh Tally Plot.	84
Figure 59. MCNPX Mesh Tally Plot.	85
Figure 60. Visual Editor Mesh Tally Plot.	85
Figure 61. MCNPX Mesh Tally Plot.	86
Figure 62. Visual Editor Mesh Tally Plot.	86
Figure 63. MCNPX Mesh Tally Plot.	87
Figure 64. Visual Editor Mesh Tally Plot.	87
Figure 65. MCNPX Mesh Tally Plot.	88
Figure 66. Visual Editor Mesh Tally Plot.	88
Figure 67. Enlarged View of a Visual Editor Mesh Tally Plot.	89
Figure 68. Dynamic 3D View Showing Universes.	90
Figure 69. Plot of a Simple Lattice Geometry Containing Universe Cell 3.	91
Figure 70. 3D Dynamic View of the Lattice.	92

Figure 71. 3D Dynamic View with different transparencies specified.....	93
Figure 72. 3D Dynamic View of a Glove Box	94
Figure 73. 2D Plots of KTMAN	95

2 Identification and Significance of the Innovation

Computer codes such as MCNPX now have the capability to transport most high-energy particle types (34 particle types now supported in MCNPX) with energies extending into the teravolt energy range. The efficient use of these types of Monte Carlo tools is very important for modeling the effects of space radiation on humans, spacecraft, and equipment. This proposal would develop a graphical user interface for high-energy multi-particle transport. With this innovation, users of the MCNPX code would have access to a powerful graphical user interface for efficient creation and interrogation of input files, which would significantly reduce the amount of time required to create and debug input files.

The MCNPX graphical user interface will be adapted from the MCNP Visual Editor, which is internationally recognized as the best code for visually creating input files for MCNP.

The Visual Editor for MCNP already has many powerful features to help the user create and display MCNP geometries. These features include the ability to display multiple cross-sectional views of the geometry, optional displays of the geometry in 3D using either wire mesh or ray tracing, plotting of the source, and optional displays of particle tracks during the random walks. Because MCNPX essentially has the same geometry package as MCNP, the powerful geometry features of the Visual Editor could be extended to MCNPX in a fairly straightforward manner by adapting the interface statements that are needed in the FORTRAN for communication with the C from the Visual Editor for MCNP to a new Visual Editor for MCNPX.

3 Technical Objectives

The technical objectives fall into three categories, all of which will result in an advanced production-working version of the Graphical User Interface (GUI) for MCNPX that was initially developed in Phase 1. These categories cover the creation of new capabilities, the enhancement of current capabilities, and the investigation into future capabilities.

The first objective consists of creating new capabilities in MCNPX and the GUI, specifically:

1. The Los Alamos Quark-Gluon String Model (LAQGSM) will be implemented within MCNPX to provide heavy-ion interaction physics. The GUI will be upgraded to support all the particle types in MCNPX and to provide visualization tools for this new feature.
2. The ability to create the source will be added to the GUI.
3. Advanced data visualization features will be added to the GUI.

The second objective is to enhance current capabilities in the GUI, including upgrading the latest version of MCNPX with support for all particles throughout the GUI and improving the 3D visualization tools.

The third objective is to investigate future upgrades to MCNPX and the GUI, specifically, the implementation of spline surfaces.

The first objective will be accomplished with tasks 1, 4, and 6.

The second objective will be accomplished with tasks 2, 3, 7, 8.

The third objective will be accomplished in task 5 and partly in task 8 as additional enhancements are considered.

4 Work Plan

4.1 Technical Approach

Table 4-1 shows the proposed work plan. The end deliverable will be an updated version of MCNPX and the GUI.

Table 4-1. Work Plan

Task	Description	Effort
1	Incorporate the LAQGSM into the MCNPX code	25%
2	Update the MCNPX Visual Editor to latest version of MCNPX	5%
3	Add support for multiple particles	10%
4	Add source creation	10%
5	Investigate the implementation of spline surfaces in MCNPX	10%
6	Implement data visualization	10%
7	Improve 3D visualization tools	5%
8	General enhancements to MCNPX Visual Editor	10%
9	Create a set of test problems	10%
10	Documentation	5%

Table 4-2 shows the timeline in which the different tasks will be accomplished.

4.2 Task Descriptions

4.2.1 Task 1: Incorporate the Los Alamos Quark-Gluon String Model Module into the MCNPX Code

Task 1 will be performed by Los Alamos National Laboratory (LANL) and will result in the implementation of the LAQGSM patch within MCNPX as a permanent feature. This patch is very large and interacts with many features of the code. Some of the details are:

1. Modify LAQGSM Fortran to comply with F90 standards (for compilation on all platforms).
2. Provide improvements in the heavy-ion slowing-down treatment (currently only a function of the number of protons).

3. Incorporate a standard tally feature that enables segregation based on heavy-ion identification.
4. Enable plotting of heavy-ion tally results.
5. Enable parallel execution of heavy-ion physics.
6. Identify and incorporate appropriate charge pick-up physics.

Once this patch has been implemented, additional work will be performed by Carter Monte Carlo Analysis, Inc. (CMCAI) and Visual Editor Consultants (VEC) to allow for the creation and display of these particles in the MCNPX Visual Editor.

This is a major effort that will span the entire two years of the grant.

4.2.2 Task 2: Update the MCNPX Visual Editor to the Latest Version of MCNPX

Task 2 involves updating the MCNPX Visual Editor to the latest version of the Fortran code. In Phase 1, MCNPX 2.6.a was used to generate the MCNPX Visual Editor. MCNPX has since been updated to MCNPX 2.6.b and it may be further updated by the time Phase 2 begins. CMCAI will perform most of this work initially, but will need to work with VEC once the update is completed to test and debug the updated code for the MCNPX Visual Editor.

Task 2 will be completed in the first 6 months. If an additional beta version is released from the MCNPX development team after this initial upgrade, a second upgrade may be required toward the end of the contract period.

4.2.3 Task 3: Add Support for Multiple Particles

Task 3 will add support for all of the different particle types available in MCNPX.

In Phase 1 an initial effort was made to support the different particle types by updating the treatment of the mode card. This work needs to be continued to accomplish the following:

1. Expand the treatment of the importances so that they can be set for different particle types; only neutrons, photons, and electrons can currently be set.
2. The cell dependent data cards need to handle the larger set of particle types (ext:n, fcl:n, wwni:n). The data routines for storing this information need to be set up dynamically to accommodate the addition of new particles in the future. Other data cards also need to be adapted to make sure they can work with MCNPX. All of these changes will need to be tested.
3. The display of collision points will need to be updated to accommodate the different particle types. The collisions can now be plotted for the starting particle if it is a neutron,

photon, or electron. This needs to be expanded to allow for a larger set of starting particles and to allow for the visualization of the particles that are created in a manner that will help the user understand the particle transport by distinctly displaying the different particle types. This will also need to be tested for the different particle types.

This task will be accomplished by VEC with support from CMCAI as needed. This is a major milestone that will be under development for the first 18 months of the grant.

4.2.4 Task 4: Add Source Creation

The Visual Editor still lacks the ability to create the source. This will become more necessary with MCNPX. As part of this task the MCNPX Visual Editor will be expanded to include the ability to create a source. This may come in the form of a wizard that allows the user to specify the type and location of the source.

Some research will need to be conducted to determine the best way to implement the source creation capability, because the source description is very generic and source descriptions can be very large. One possible approach is to support most of the common source configurations (perhaps with a wizard) and then have an advanced option that allows for experienced users to create more complex source descriptions.

This is expected to be a significant effort that will be under development for the entire two years of the grant. The work will be completed by VEC with support from CMCAI to modify the MCNPX Fortran source code as needed.

4.2.5 Task 5: Investigate the Implementation of Spline Surfaces in MCNPX

Many complex objects do not conform to the surfaces currently available in MCNPX. Many of these surfaces can be more accurately defined by a spline surface; additionally, many CAD geometries contain spline surfaces that cannot currently be converted to a MCNPX geometry using the CAD to MCNP conversion tool. This is complicated by the fact that most spline surfaces are cubic and MCNP currently only supports quadric surfaces. This task will investigate how to implement a spline surface in the MCNPX code and how to make an initial implementation, if possible.

This task will be accomplished in the second year of the grant and will be performed mostly by CMCAI with support from LANL and VEC as needed.

4.2.6 Task 6: Implement Data Visualization

Task 6 will investigate ways to help the user better visualize data. This will include implementing the radiograph tally, implementing mesh tallies, and adding support for the weight window mesh.

Additional research will be conducted to determine if there are additional data visualization tools that would be useful to the user. This task will be completed at the end of the two-year grant and will be implemented by VEC with support from CMCAI as needed.

4.2.7 Task 7: Improve 3D Visualization Tools

The MCNPX Visual Editor has some support for the 3D display of geometries, through 3D ray tracing and the dynamic display of 3D geometries. This was an initial effort designed to show proof of concept, but it needs to be expanded to make it more user friendly and capable of handling large problems efficiently. Items that will be addressed in this task include:

1. Make it easier to select cells for both ray-tracing and dynamic 3D display.
2. Upgrade the dynamic display to work for more complex cells.
3. Investigate and implement, if possible, the display of universes and lattices.
4. Improve the efficiency of the display of 3D dynamic geometries; this can be very slow to rotate for complex geometries.

Task 7 will be accomplished in the second year of the contract and will be implemented by CMCAI with support for visualization from VEC as needed.

4.2.8 Task 8: General Enhancements to MCNPX Visual Editor

If time and funding are available, general enhancements to the MCNPX Visual Editor will be investigated and implemented if possible. Possible enhancements include:

1. Implement tally creation and visualization.
2. Update the material library to be more comprehensive and to include materials of interest to the MCNPX user community. The current material library consists of 29 materials. This library should be expanded to allow users to select from a larger material database.
3. Investigate and implement the mix and match ability available in MCNPX (MXn:p).
4. Upgrade the input for a transformation to include visual aids and perhaps features like the optional use of Eulerian angles to determine the MCNP transformation coefficients.
5. Enhance the volume calculator to handle more complex cells.
6. Include an option to create a number of MCNP input files starting with one input file and specifying the parameters on the input file that are to be varied and how to vary them.

Additional general enhancements may be pursued depending on input from NASA, LANL, and users.

4.2.9 Task 9: Create a Set of Test Problems

Task 9 will consist of creating a comprehensive set of test problems to test the MCNPX Visual Editor. These test problems will need to test the many different features available to make sure the editor is reliable and can handle a diverse set of situations. This is essential to make sure that users will be getting a reliable program that does not die easily. If the program is unreliable it will not be used.

To test the graphical user interface a number of problems will need to be generated that need to be performed by hand. These problems are more complex than the test problems used to test the MCNPX code, because these tests cannot be easily automated.

This will be a significant task because it will take time to create the problems that can adequately test the graphical user interface, and then it will take time to run the problems. This task will start in the first year and continue until the end of the grant.

4.2.10 Task 10: Documentation

Task 10 will document the changes in the form of a user's manual. The existing Visual Editor Manual will be modified to include the enhancements performed as part of this grant. An initial version of the MCNPX manual was generated from Phase 1. This manual will be enhanced to cover all of the new capabilities that will be added from the work accomplished in this proposal.

4.3 Meeting the Technical Objectives

The above tasks will meet the technical objectives outlined in Section 3.0 by providing a path to creating a graphical user interface for the MCNPX computer code as specified in Section 3.0.

4.4 Task Labor Categories and Schedules

Table 4-2. Timeline

Task Number (Lead)	Months 1 - 6	Months 7 - 12	Months 13 - 18	Months 19 - 24
1 (LANL)	—————→			Task 1 complete
2 (CMCAI)	Task 2 complete			
3 (VEC)	—————→		Task 3 complete	
4 (VEC)	—————→			Task 4 complete
5 (CMCAI)		—————→		Task 5 complete
6 (VEC)		—————→		Task 6 complete
7 (CMCAI)			—————→	
8 (VEC)			—————→	
9 (VEC)		—————→		Task 9 complete
10 (VEC)				Task 10 complete

LANL will have the primary responsibility of completing the first objective. LANL will work with Dr. Lee Carter of CMCAI and Randy Schwarz of VEC to ensure compatibility of this module with the MCNPX Visual Editor. The remaining items will be performed by CMCAI and VEC with support from LANL as needed. The Fortran upgrades to the MCNPX code will be performed primarily by CMCAI, and the GUI development will be performed by VEC. LANL will provide additional support and advice as needed.

5 Potential Applications

VEC does not sell the Visual Editor; it instead releases this code to the Department of Energy code center (RSICC) to make the code available to interested users. It is believed that this provides the maximum benefit to the user community and enhances the international reputations of both VEC and CMCAI.

MCNPX is available to users around the world; any enhancements to the GUI enable all users of the code to be more effective in creating their geometries and performing their calculations. The improvements proposed in this document can substantially reduce the amount of time to create a computer model, benefiting users both within and outside of NASA.

VEC will obtain financial reimbursement from teaching how to use this tool and from contracts with users of the tool that would like specific features added or help using the software for performing calculations.

5.1 Potential NASA Applications

Any applications that require the transport of high-energy particles will benefit from this effort. These applications include:

1. Space shielding applications.
2. Evaluation of damage to electronic components in space.
3. Health physics applications for people living in space.
4. Mapping of planets.
5. Investigations of cosmic-ray radiation backgrounds and shielding for high altitude aircraft and spacecraft.
6. Charged-particle propulsion concepts for spaceflight.
7. Single-event upset in semiconductors, from cosmic rays in spacecraft or from the neutron component on the earth's surface.

5.2 Potential Non-NASA Applications

Any physics applications that utilize the transport of high-energy particles will benefit from this work. These applications include:

1. Investigations for accelerator isotope production and destruction programs, including the transmutation of nuclear waste.
2. Design of accelerator spallation targets, particularly for neutron scattering facilities.
3. Research into accelerator-driven energy sources.
4. Medical physics, especially proton and neutron therapy.
5. Accelerator-based imaging technology such as neutron and proton radiography.
6. Design of shielding in accelerator facilities.
7. Activation of accelerator components and surrounding groundwater and air.
8. High-energy dosimetry and neutron detection.
9. Design of neutrino experiments.
10. Charged-particle tracking in plasmas.

6 Contacts

6.1 Key Contractor Participants

Randy Schwarz, Principal Investigator
Visual Editor Consultants
P.O. Box 1308
Richland, WA 99352
Website: <http://www.mcnpvised.com>
Email: randyschwarz@mcnpvised.com
Phone: (509) 539-8621, Fax: (509) 946-2001

Dr. Leland (Lee) Carter, Consultant
Carter M.C. Analysis, Inc.
6636 Hogan Dr. N.
Keizer, OR 97303
Email: leecarter6@comcast.net
Phone: (503) 304-2966

6.2 Key NASA Participants

None.

6.3 NASA and Non-NASA Advisors

Dr. Gregg W. McKinney
MCNPX Project Leader and Radiation Transport Team Leader
Nuclear Design & Risk Analysis (D-5)
Los Alamos National Laboratory
P.O. Box 1663, MS K575
Los Alamos, NM 87545
Email: gwm@lanl.gov, Phone: (505) 665-8367

John Hendricks
Nuclear Design & Risk Analysis (D-5)
Los Alamos National Laboratory
P.O. Box 1663 MS K575
Los Alamos, NM 87545
Email: jxh@lanl.gov, Phone: (505) 667-6997

7 Technical Activities

7.1 Recent Technical Activities

The latest version of the MCNPX Visual Editor code generated for this reporting period is version 23Q. During this reporting period a number of bugs with the code were solved and the 3D dynamic plotting capability was modified to be more robust and allow more user control of the movement of the geometry and the colors. Section 7.1.7 provides additional details on these improvements.

7.1.1 Task 1: Incorporate the Los Alamos Quark-Gluon String Model Module into the MCNPX Code

A Work For Others agreement (FIA-07-012) was generated by LANL and signed by both VEC and LANL.

Dr. Michael R. James of LANL received a version of LAQGSM from Stepan Mashnik (v 3.01, dated 6/8/2006). The code was pulled together into one file, and a framework of preprocessor statements was integrated that allowed the same file to be compiled standalone or as a module to MCNPX.

Dr. James incorporated the linking routines to couple LAQGSM to MCNPX into the module. These are also wrapped with preprocessor statements.

Dr. Michael James of LANL migrated the heavy ion transport capabilities into MCNPX version 2.6.c. A physics module of LAQGSM 3.01 was created that integrates into MCNPX. Dr. James and Dr. Gregg McKinney updated the LAQGSM 3.01 code, enabling

the code to be ported to other compilers/architectures, and they have verified that it runs consistently as a standalone package.

The LAQGSM 3.01 physics model was incorporated into MCNPX 2.6.e so that the transport of heavy ions is possible. LANL has benchmarked the performance of LAQGSM in MCNPX and verified that the code gives answers consistent with the answers that come out of the stand-alone version of LAQGSM.

The heavy-ion transport capability was integrated into MCNPX 2.6.e. LANL ran benchmark calculations on integrated problems that involve a combination of slowing-down energy loss and nuclear interactions. These problems give excellent agreement with the experimental data.

LANL has also run this new version of MCNPX 2.6.e with all of their test problems and have verified that they do not substantially change the tracking of the code (any small changes that do exist can generally be attributed to improvements or bug fixes in the code).

The LAQGSM physics model and Heavy Ion transport feature were integrated into MCNPX v2.6.0 and was sent to RSICC for release.

The Los Alamos team has also conducted a number of calculations and benchmarking tests to explore the physics regime up to 1 TeV/nucleon. Specifically, a prototypic cosmic ray spectrum of protons through Ni-58 was used as a source to compute dose in a human phantom model. From this, a number of issues have come to light that demonstrated limitations in the original release. Several bug fixes have been made and the latest beta version of the code, v2.7.a, is much more stable and reliable. The latest version of MCNPX has demonstrated calculations up to hundreds of millions of source histories from a cosmic ray spectrum.

LANL has also explored new physics options that are not yet handled in the code such as elastic collisions involving heavy ions and charge pick-up during transport.

In terms of the work goals, there status report for each goal follows:

1. Modify LAQGSM Fortran to comply with F90 standards (for compilation on all platforms).
Done
2. Provide improvements in the heavy-ion slowing-down treatment (currently only a function of the number of protons).

Incomplete. Currently, the heavy ions compute stopping powers from preexisting MCNPX routines.
3. Incorporate a standard tally feature that enables segregation based on heavy-ion identification.
Done. FT RES cards provide this feature.

4. Enable plotting of heavy-ion tally results.
Done.
5. Enable parallel execution of heavy-ion physics.
Done.
6. Identify and incorporate appropriate charge pick-up physics.
Incomplete.

7.1.1.1 Operation of Heavy Ion Transport Feature of MCNPX

Transport and interactions of heavy ions are now available as an option in MCNPX. The symbol “#” is used to represent the heavy ion particle type. The “#” represents all possible heavy ion types (basically any ion that is not one of the existing light ion types in MCNPX that are deuterons, tritons, He-3 and alpha particles). Although the “#” is generic to all heavy ions, the identities of different heavy ions are tracked by their appropriate Z (charge) and A (mass number). The user cannot choose to transport any particular heavy ions (other than specific source particles), although tally specifications can be made for specific ions.

MCNPX Particles

IPT *	Name of Particle	Symbol	Mass (MeV)	Low Kinetic Energy Cutoff (MeV)	Mean Lifetime (seconds)
35	heavy ion	#	varies	5.0	Huge

The “#” may be used wherever a normal particle type is called for. If placed on the mode card with other particles types, heavy ion transport is turned on. If placed in a tally such as *f4:#*, a track length tally of the heavy ions will be printed in the output. The “#” may also be used in mesh tallies.

To use a heavy ion as a source particle, the ZA identifier is put into the *par* field of the *sdef* card such as *sdef par = 6012* which will start a C-12 ion. Note that the energy is still input as total energy and not energy/nucleon.

Heavy ions can also be used a distribution of source particle types. The specification:

```
sdef par = d1
si1 L 1 9 3006 26056 92238
sp1 1 1 0.1 0.3 0.5
```

would sample five different source particles : neutron, proton, Li-6, Fe-56, and U-238 in the relative frequencies given on the *sp* card. Dependent distributions can be used to specify different energies for different heavy ions.

The heavy ions will use LAQGSM as the default nuclear cascade/evaporation model. The LAQGSM model can be used for other particles by setting the 10th entry on the *Ica* card to 1. There are no ENDF data tables for the heavy ion reactions; therefore, there is no transition energy between table and model physics such as there is for protons and neutrons.

When running with heavy ions turned on ('#' on the mode card), any residuals that are produced from any model physics (such as CEM, Bertini, etc.) will be transported even if the source particle is not a heavy ion.

For example, a proton source that is directed into a target and uses CEM as the model physics will produce residuals from the nuclear reactions. If the “#” sign is on the mode card, these residuals will be banked and transported. Note that because the table physics regime does not normally create residuals, any interaction using of proton and neutron libraries will not produce transportable heavy ions.

Due to structural limitations in the code, the list of possible heavy ions that can be transported must be pre-defined. However, the list is quite large. Most of the isotopes listed on the Table of Isotopes (~2205 isotopes) are available to be tracked. On the rare occasion when an isotope is created that is not on this list (e.g. something far from the line of stability) the ion is stopped immediately and its energy is all deposited in the local region.

7.1.1.2 Tallies

The FT option will be available to the user to create user bins for tallying specific isotopes of heavy ions. This feature is still under development, but its general form will use the FT RES option to specify the ZA identifiers of interest. The general form is *FTn:<pl> RES*, where n is the tally number that the FT card applies and <pl> is the particle type, in this case usually a heavy ion (#). Inputs to the FT RES option can be performed in three ways:

1. Specifying *FT RES* with no isotopes listed automatically creates a bin for each of the 2205 ion types.
2. Specifying *FT RES z1 z2* will create user bins for all isotopes between the proton numbers z1 and z2.
3. Identifiers can be explicitly listed after the RES keyword for each user bin.

For example:

```
F4:# 6  
FT4 RES 8016 20040 26000 92238
```

creates a track length tally in cell 6 and then creates four users bins for the isotopes 8016, 20040, 26000, and 92238. Note that specifying the elemental Z Aid for Fe (26000) will lump all Fe isotopes into one bin. The FT RES option will work with type 1, 2, 4, and 6 tallies when using the heavy ion particle type. The FT RES still works as a special feature for f8 tallies where it functions as a residual tally.

Mesh tallies with heavy ions are possible. The type 1 (particle track) and type 3 (energy deposition) mesh tallies work with heavy ions; however, there is no capability to separate out contributions from particular heavy ion species.

7.1.1.3 Other Cards

The PHYS:# has the input features for a generic charged particle. Note: Light-ion recoil control still only applies to light ions. There is no heavy ion recoil for proton elastic scattering events.

SSR, SSW: Surface source reading and writing are compatible with heavy ions. The charge and mass for each heavy ion are stored in a surface source file and will be read back in to reconstruct the proper source distribution.

HSTEP: Applies to all charged particles (except electrons which use ESTEP), including heavy ions.

7.1.1.4 Variance Reduction

The heavy ions, similarly to the light ions, are able to use any variance reduction technique that is applicable to charge particles that use physics models for interactions. Any variance reduction technique used for heavy ions will apply equally to all heavy ion particles.

Weight windows, Cell importance, SPABI, ESPLT: All work with heavy ions.

FCL, EXT: Only applies to table physics regime. They are not functional with heavy ions.

DD, PD, DXTRANS: These cards are not applicable for charged particles.

7.1.1.5 Changes to PRINT Tables

The stopping powers for the primary or default heavy ion are printed with the normal stopping power tables in Print Table 85. An optional table, Print Table 87, prints the stopping powers that are generated for all of the secondary heavy ions. The stopping power tables for secondary heavy ions are simplified such that only the proton number of the heavy ion will determine which table to use. Note that all heavy ions specified with a distributed source will use the secondary stopping power tables. The Z and A identifiers for each table are shown in the particle identity field at the top of the table.

MCNPX Output Tables

Table Number	Type	Table Description
87	optional	Secondary heavy ion stopping powers and straggling

Changes have also been made to the Print Table 110 so that when a distributed source of heavy ions is requested, the Z and A identifiers for the sampled source particle are printed in the source particle field.

7.1.1.6 Limitations

The heavy ion feature has some restrictions when used with other features in MCNPX. No heavy transport information is written to the histp file (aside from the usual recoils from which the heavy ions are started). The PTRAC file will contain the heavy ion particles and their track information, but the individual HI identities (Z and A) are **not** recorded.

In a similar manner, the summary tables contain all the heavy ion information summed together; no method of breaking out individual heavy ion contributions for these tables is currently available.

7.1.1.7 LAQGSM Physics

The Los Alamos Quark-Gluon String Model v. 3.01 has been implemented as a physics model option in MCNPX. This model contains elements of the CEM INC physics model and GEM evaporation model but is intended to handle a wider variety of particles and energies. LAQGSM can simulate the interactions of A1+A2 interactions, as well as nucleons (protons and neutrons) and gammas. In MCNPX by default, LAQGSM handles all the heavy-ion interactions as well as light-ion interactions above 1 GeV/nucleon (below this, ISABEL is still the default model). However, setting the 10 entry (*ilaq*) on the lca card to 1 will redirect all light ions to LAQGSM. Neutrons and protons will also be sent to LAQGSM (instead of FLUKA) when their energies are above the transition energy.

LCA Input Parameters

Input Parameter	Description
<i>ilaq</i>	Choose alternative physics model. If <i>ilaq</i> =0, use LAQGSM for light-ion interactions only above 940 MeV/nucleon. ISABEL will handle light-ion interactions below this energy. Use FLUKA for proton and neutron interactions above the energy cutoff specified by <i>flenb1</i> and <i>flenb2</i> on LCB card. LAQGSM will be used for heavy ion interactions at all energies (DEFAULT) If <i>ilaq</i> =1, use LAQGSM to handle all light-ion interactions as well as all heavy-ion interactions. LAQGSM also replaces FLUKA for high-energy proton and neutron reactions.

LAQGSM requires two new data files (*atab.dat* and *channell.tab*) in addition to three of the standard CEM data files (*shell.tbl*, *gamman.tbl*, and *mass.tbl*). If these files cannot be found in one of the standard paths, the code may fail to execute. Note that there are some situations where LAQGSM is invoked automatically, even when the *ilaq* variable is set to zero.

7.1.1.8 LAQSM Particle Track Plotting

The Visual Editor has been updated to support the plotting of particle tracks from heavy ions. Figure 1 shows a particle track plot of a carbon beam therapy problem. Carbon nuclei come in from the top of the geometry. The carbon beam stops at the tumor region in the center of the sphere. Notice that the carbon beam loses energy in the tumor region as shown by the particle beam turning blue. Secondary protons are shown in purple and secondary neutrons are shown in light green.

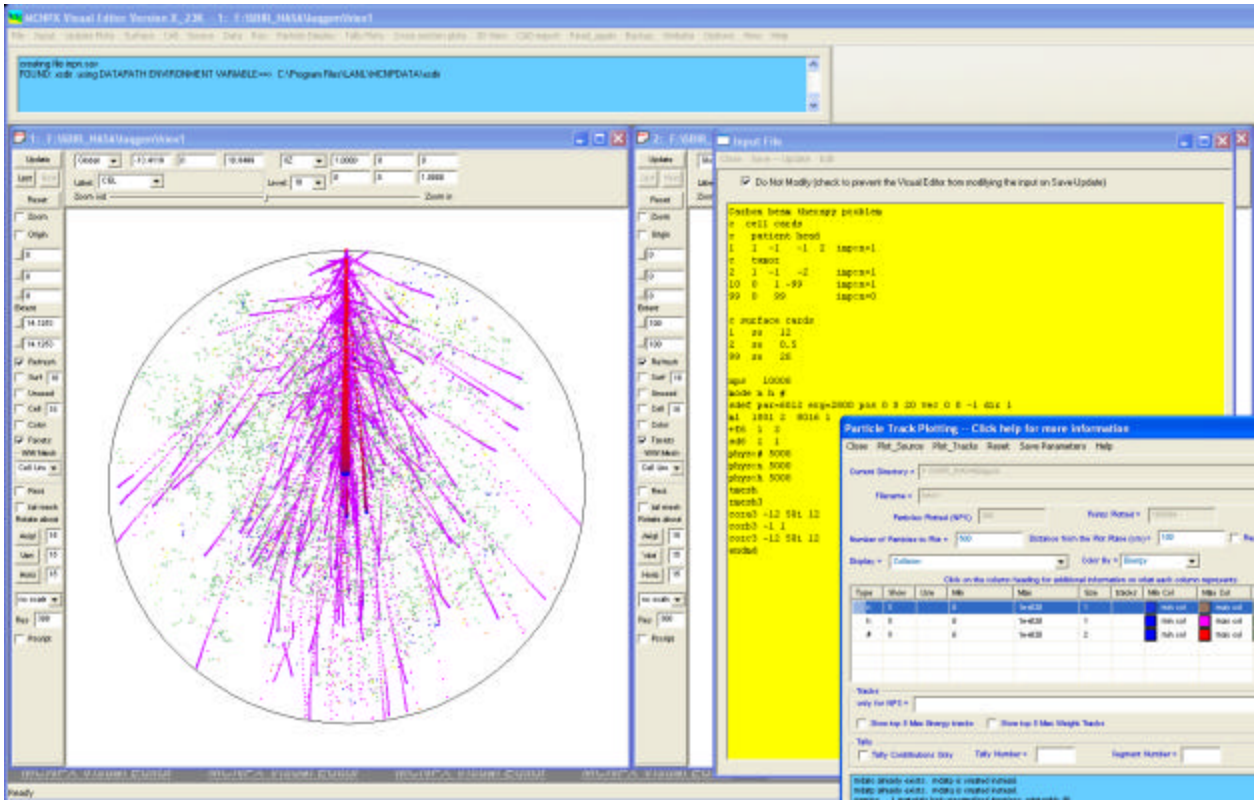


Figure 1. Carbon Beam Therapy Problem.

Figure 2 shows collision points for iron ions incident on a water phantom. The iron ions are shown in red. The secondary protons are shown in purple. A few alphas are shown as cyan points. Secondary neutrons are shown as small green dots in the background.

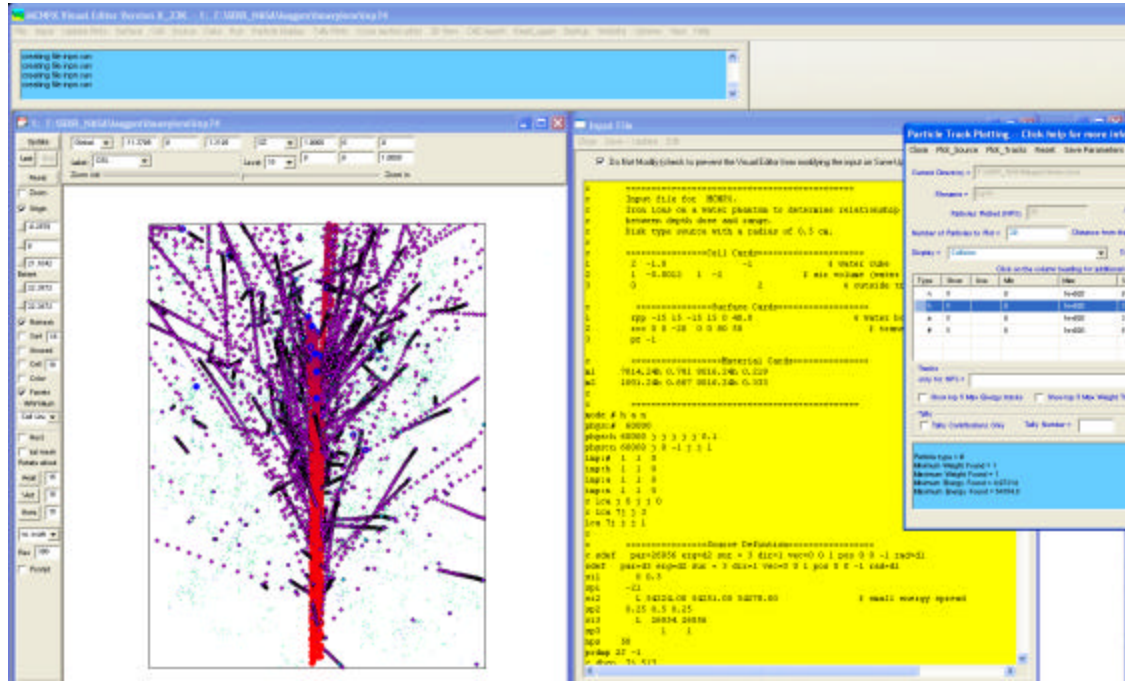


Figure 2. Iron Ions on a Water Phantom.

Figure 3 shows a plot of uranium isotopes in lithium. The uranium ions come in from the bottom and lose energy as they move upward producing a smooth red to purple spectrum going from bottom to top. Secondary protons are shown in purple. A few neutrons can be seen as olive colored points. The orange collisions represent deuterons moving outward from the uranium ion beam.

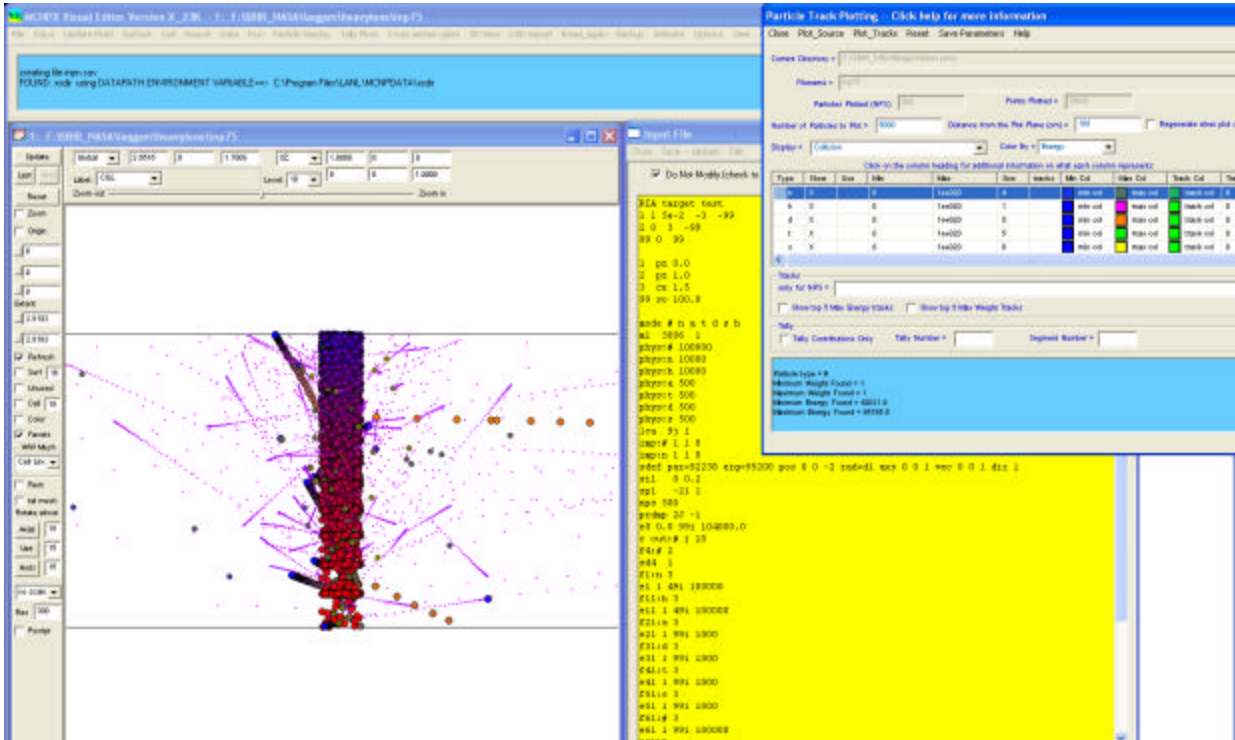


Figure 3. Uranium Isotopes on Lithium.

These plots demonstrate the capability of the Visual Editor to plot collision points generated by heavy ions interacting with matter.

7.1.2 Task 2: Update the MCNPX Visual Editor to the Latest Version of MCNPX

The latest beta version of the MCNPX code (version 26f) was downloaded from the MCNPX website. An updated version of the Visual Editor was compiled with the 26f beta Fortran code. Most of the improvements and testing of the code for this reporting period use this version of the code. Version 26f has support for the LAQGSM physics used to generate the particle collision points from heavy ions.

7.1.3 Task 3: Add Support for Multiple Particles

CMCAI has modified the Visual Editor Fortran patch file to provide additional particle information to the C++ code to allow the user to differentiate different particle types. Figure 4 shows the upgraded particle track-plotting panel. This panel can be accessed by selecting Particle Display->Plot Particle Tracks from the main menu.

Particle track plotting in the Visual Editor works by having the Fortran MCNPX code calculate the particle position and then projecting this position onto the 2D plot window. Because the particles are projected onto a 2D plot window plane, a sphere and a cylinder may look the same. The user may have to display several 2D slices to fully visualize the source region. The user can specify the distance from the 2D slice displayed in the plot window on which particles will be projected by changing the value of the “Distance from the plot plane (cm)” value. The default distance is 100 cm on either side of the 2D plot window plane.

In Figure 4, test input file inp110 has been read in. Ten particle types are identified in the mode card (mode n p | h / z d t s a). Within the “Particle Track Plotting” window, each particle type is listed. The user has a number of options that can be set for each particle type. The “show” column is used to request that the particle type be displayed when the collisions are plotted. This can be toggled on (indicated with an “X”) and off with a left click of the mouse button. The default is to show all particles. The next column has a heading of “use”. This column indicates the min and max values, as specified in the following two columns, which are to be used to limit the particles displayed to the range specified by the user. This can be toggled on (indicated with an “X”) and off with a left click of the mouse button. The default is to not use the specified ranges.

The min and max columns allow the user to specify an upper and a lower value for the particles plotted. The meaning of these values depends on the “color by” option. The values are set by clicking on the “Min” or “Max” value for the particle type listed in the row. If “energy” is selected, the values represent a lower and upper energy, if “weight” is selected the values represent a lower and upper weight. The default is to use energy.

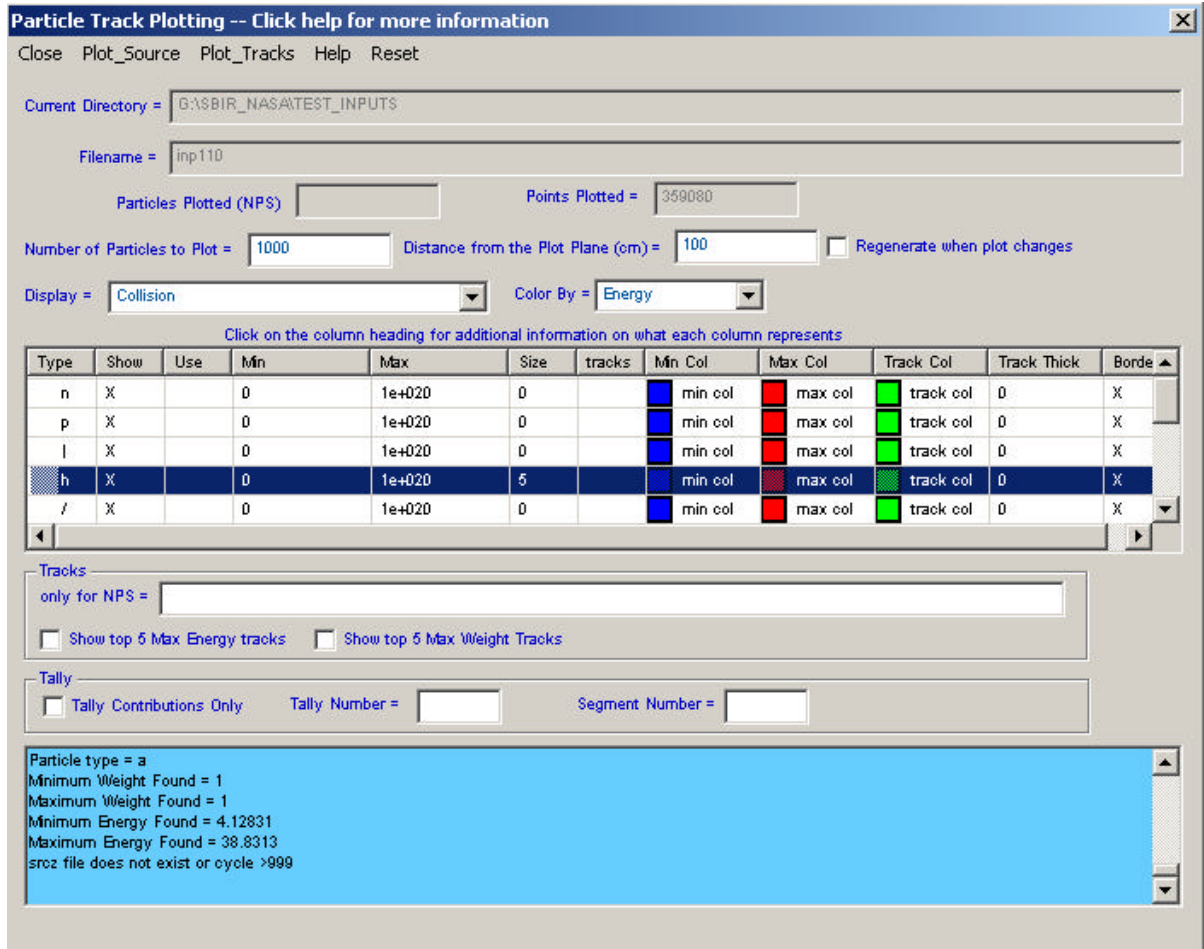


Figure 4. Particle Track Plotting Panel in the MCNPX Visual Editor.

The next column, labeled “size”, allows the user to set the size of the particle. The user can select a row in this column with the left mouse button, and a menu will appear that will allow the user to change the relative size from any value between 0 and 5. The default value is 0, which will plot each point as a pixel.

Following the “size” column is a “tracks” column. If this column is selected, the user is requesting that tracks be plotted for the particle selected. This can be toggled on (indicated with an “X”) and off with a left click of the mouse button. The default is to not plot the particle tracks.

The next three columns allow the user to set the minimum color, the maximum color, and the track color. When the user selects a row in this column, a panel will be displayed that allows the user to specify the color for each of these three entries.

The “Track Thick” column allows the user to specify the thickness of the track line if the user has requested that particle tracks be plotted. The user can select a row in this column with the left mouse button and a menu will appear that will allow the user to change the relative size from any value between 0 and 5. The default value is 0, which will plot the thinnest line for the particle tracks.

The final column allows the user to set a black border around the collision points. The default is to show this border for all particle sizes greater than 0.

In Figure 5, the source for inp110 is plotted by selecting “Plot_Source” from the menu at the top of the particle track-plotting panel previously shown in Figure 4. The source definition is defined as follows:

```
sdef sur = 1 erg = 800 par = h dir = 1 pos = 0 0 0 rad = d1
sil 2
```

This defines a source on surface 1 at x=0, y=0, and z=0. It has a radius of 2 cm defined by “si1”. The particles are generated in the positive z direction, because this is a surface source and the default reference vector (VEC) is in the positive direction of the surface normal, which is (0,0,1) for surface 1. The dir vector indicates that all particles are generated in this normal direction.

The source particle is a proton (h); to see the source it is necessary to set the particle size to 5 for protons and then select Plot_Source.

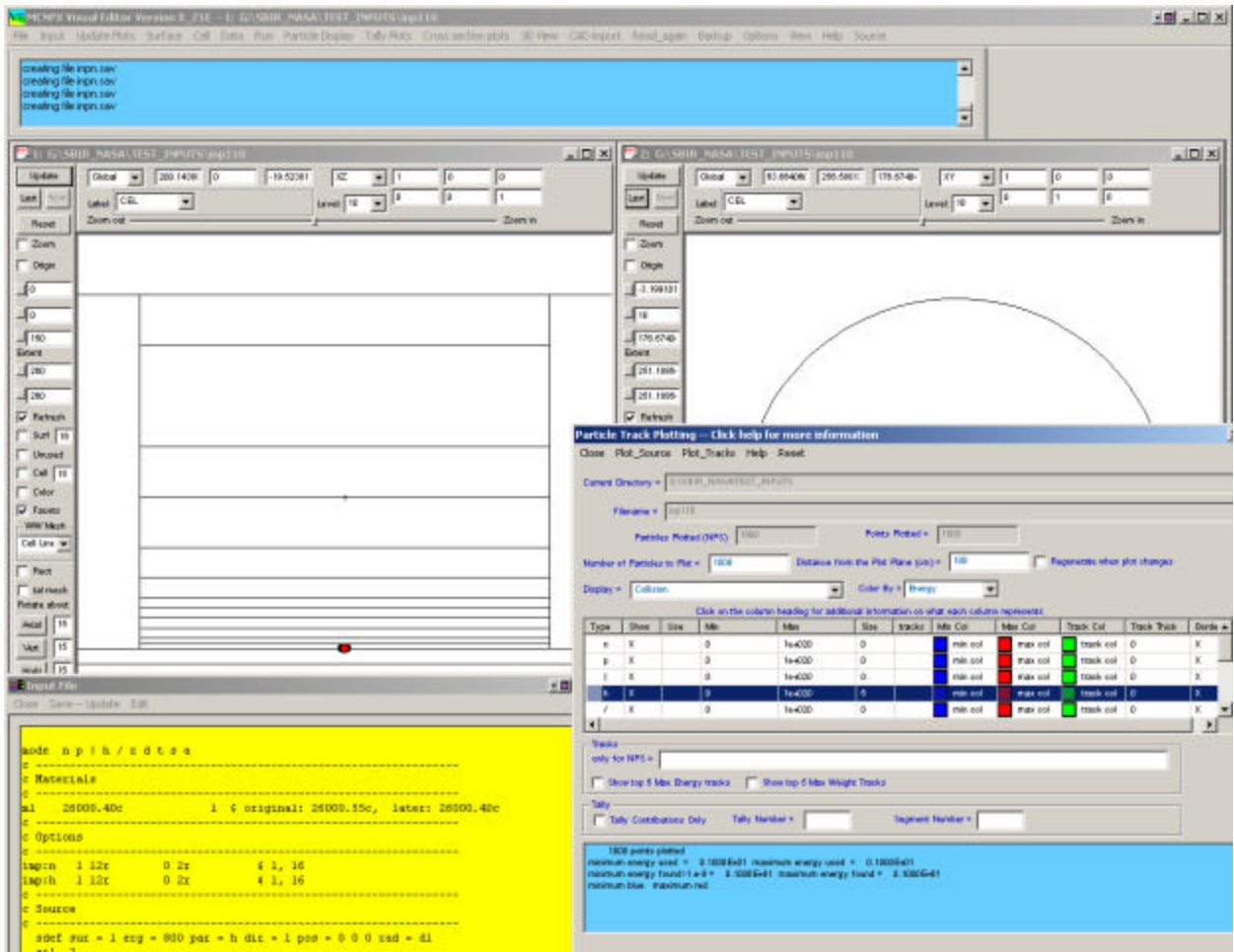


Figure 5. Plotting the Source in the MCNPX Visual Editor.

Figure 6 shows a close-up view of the surface source. When the particles are plotted they now appear as a line instead of a point. The line represents the surface source viewed from the side.

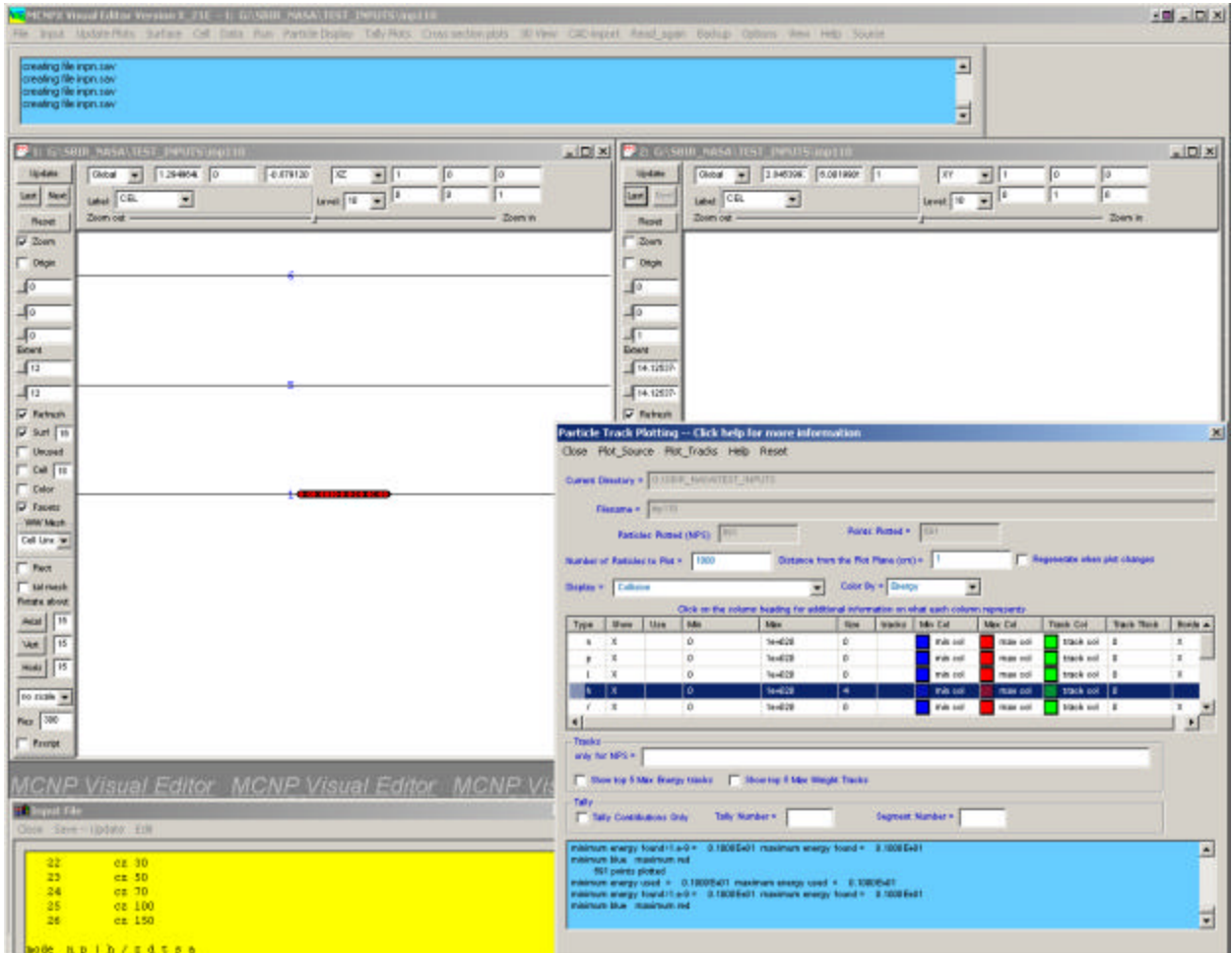


Figure 6. Plotting the Source in the MCNPX Visual Editor with Equal Extents.

In Figure 7, “Plot tracks” is selected to plot the particle tracks. Because the source particle (protons) is enlarged (set to size 5), the protons can be observed to be moving upward and generating the secondary particles that are still set to a pixel size. The low energy is set to blue, and the high energy is set to red.

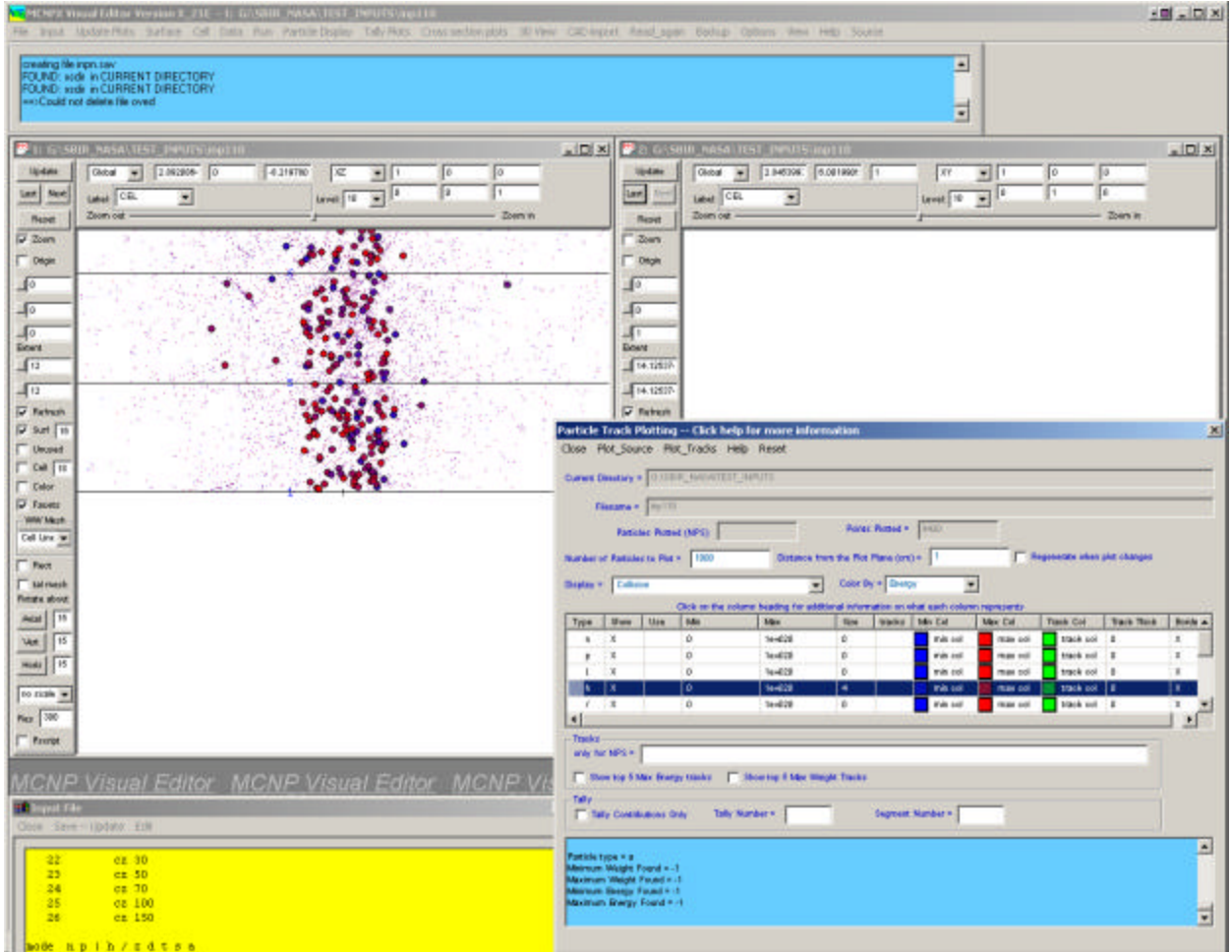


Figure 7. Plotting Collision Points.

In Figure 8 the particles are colored by weight and the extents are set to 100. To differentiate the different particles that are generated, the neutrons are yellow, the photons are green, and the muons are cyan colored. The color of the particle is set by clicking on the color box and setting it to the desired color. The color picker is shown in Figure 8.

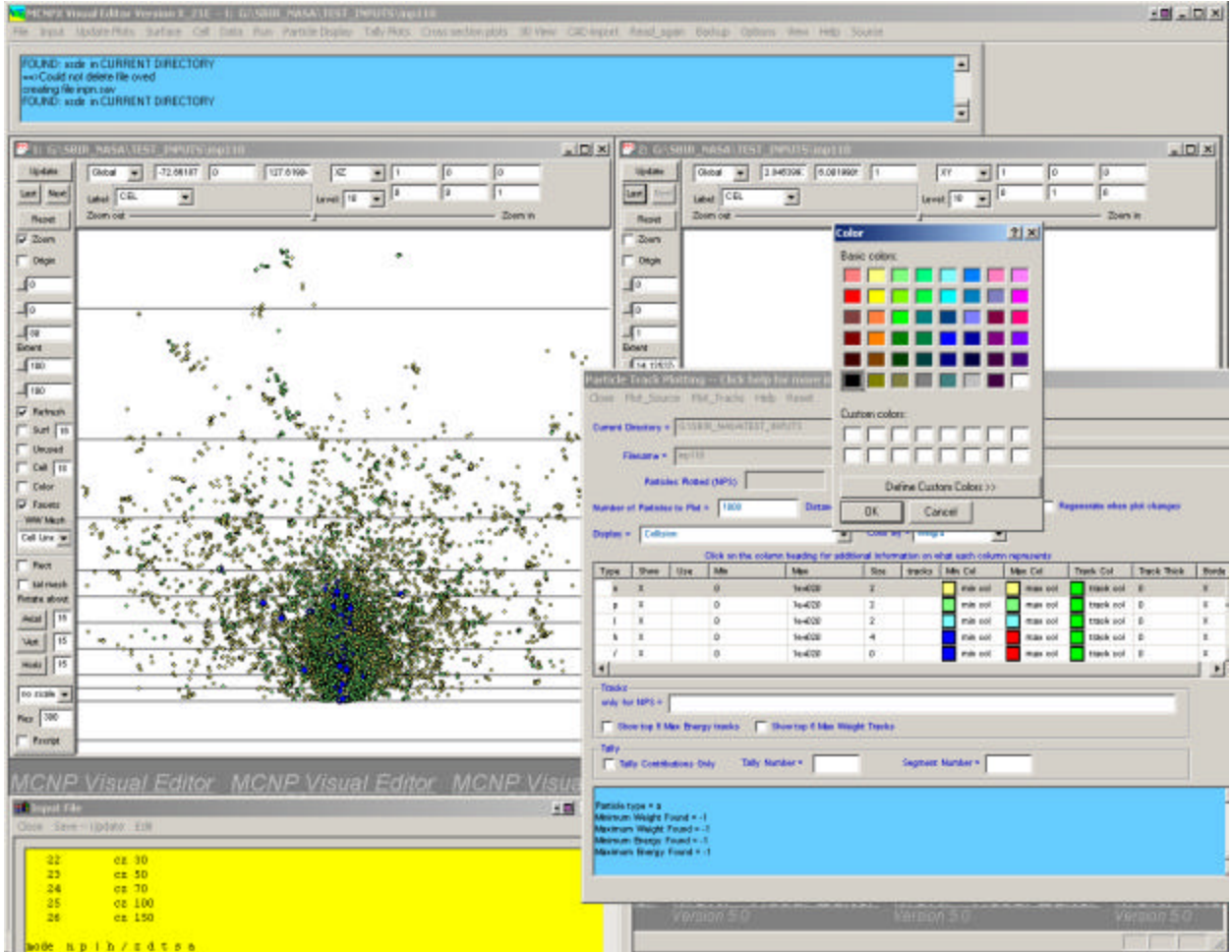


Figure 8. Plotting Collision Points with Different Colors.

In Figure 9, only the source protons and the generated neutrons are shown. All of the other particles are hidden (by unselecting the show option). The source protons are red and set to the smallest size, while the neutrons go from blue to green (depending on the energy) with a size of 3. The distance from the plot plane is set to 10 cm.

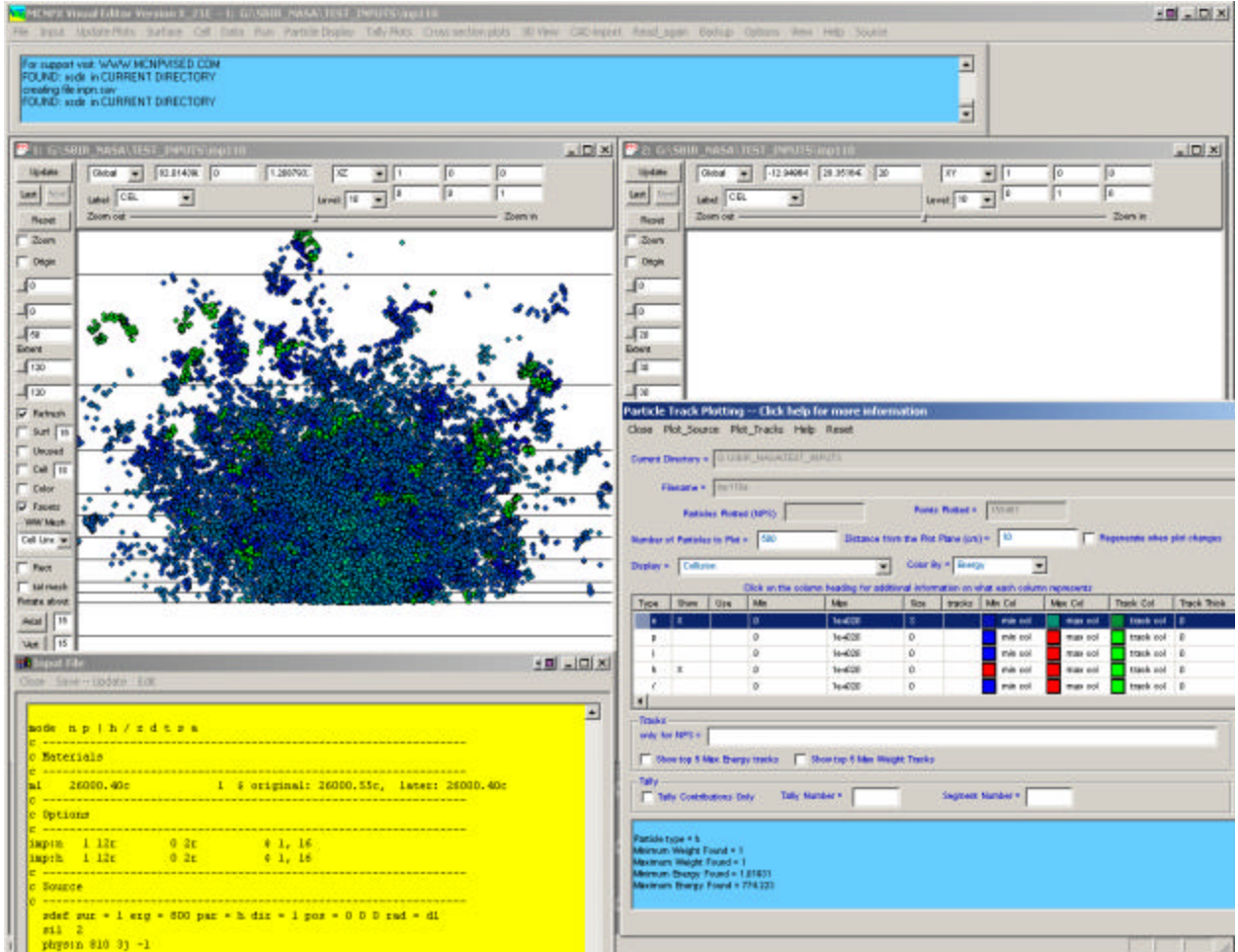


Figure 9. Plot of Neutron Collision Points.

Figure 10 shows the photons generated for this input file by selecting photons and increasing the size of the photon particles.

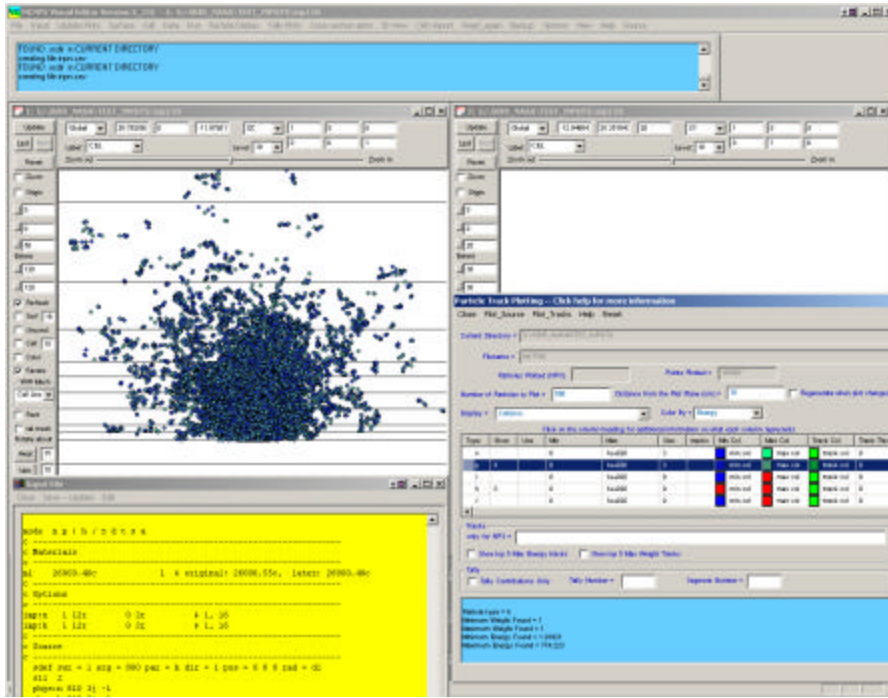


Figure 10. Plot of Photon Collision Points.

Figure 11 shows the muons generated for this input file.

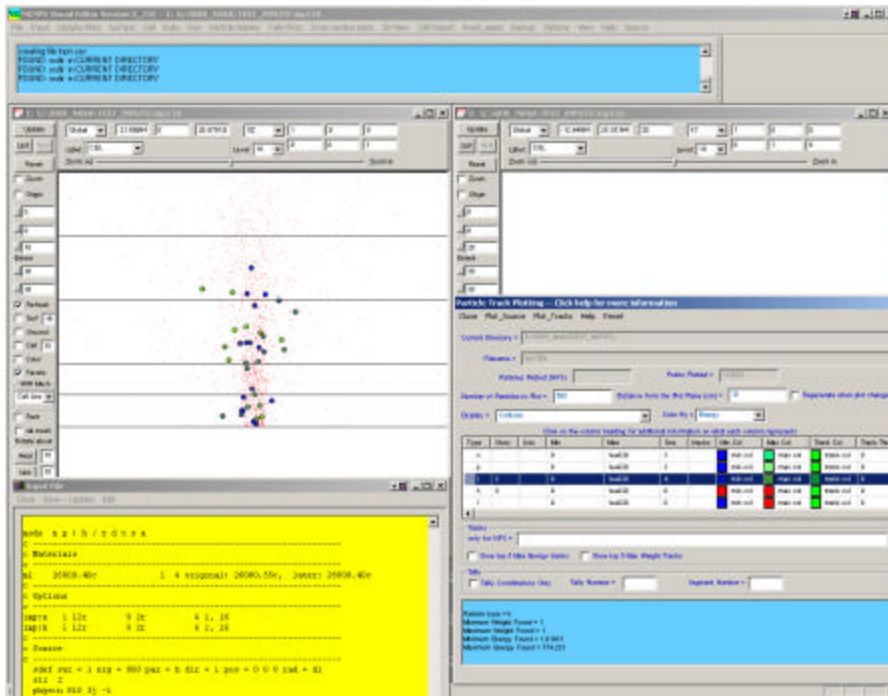


Figure 11. Plot of Muon Collision Points.

Figure 12 shows the +/- pions generated for this input file.

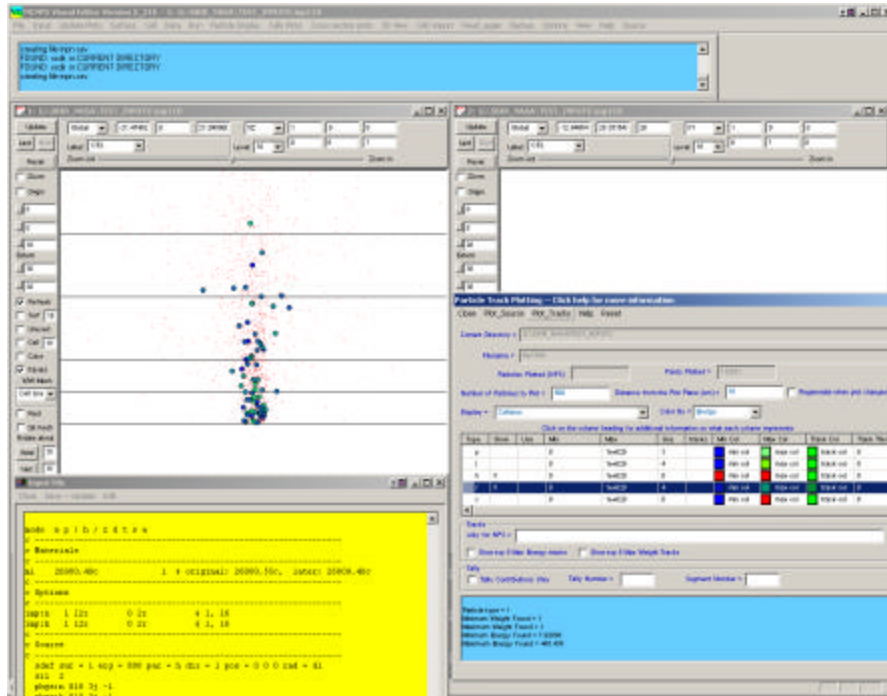


Figure 12. Plot of Pion Collision Points.

Figure 13 shows the neutral pions generated for this input file.

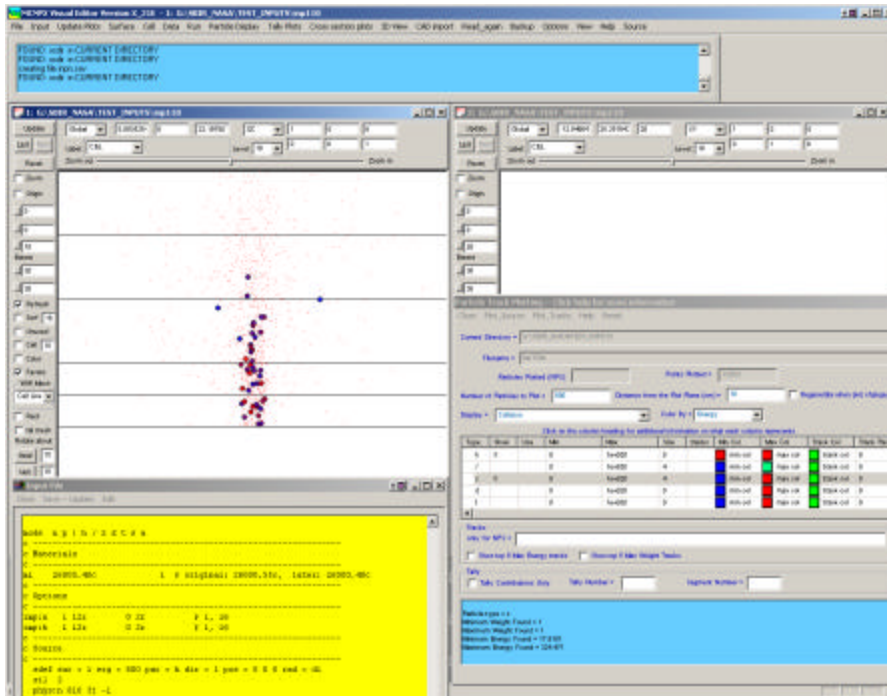


Figure 13. Plot of Neutral Pion Collision Points.

Figure 14 shows the deuterons generated for this input file.

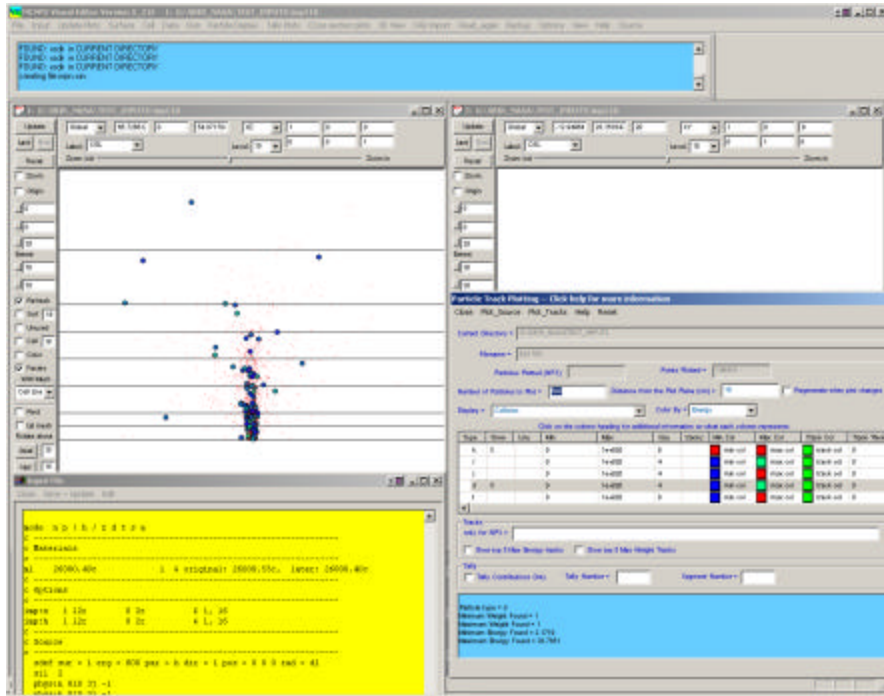


Figure 14. Plot of Deuteron Collision Points.

Figure 15 shows the tritons generated for this input file.

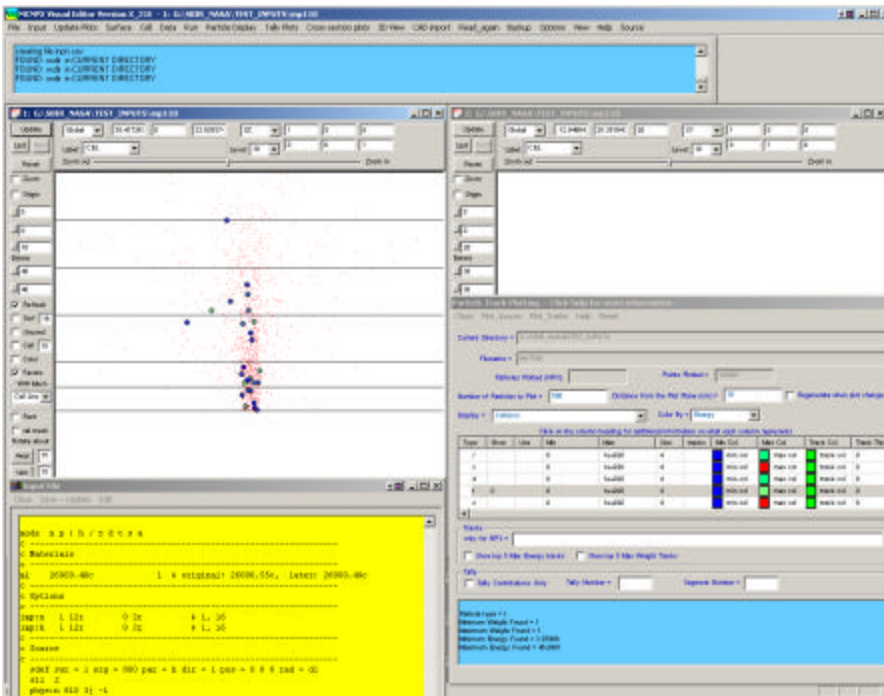


Figure 15. Plot of Triton Collision Points.

Figure 16 shows the He-3 collision points generated for this input file.

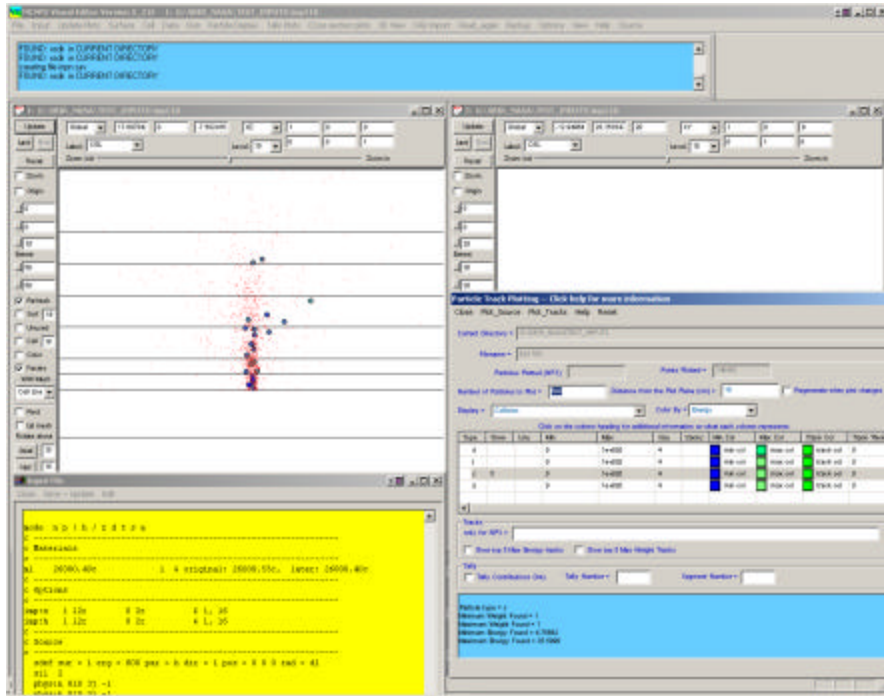


Figure 16. Plot of He-3 Collision Points.

Figure 17 shows the He-4 collision points generated for this input file.

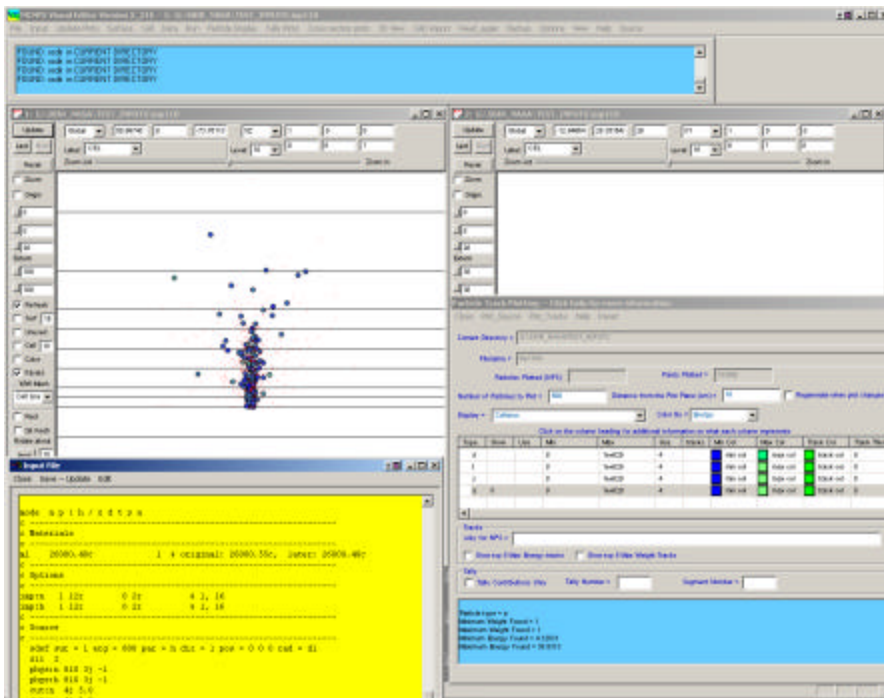


Figure 17. Plot of He-4 Collision Points.

Figure 18 shows the He-4 collision points generated for this input file. The collision points have been set to a cyan color, tracks have been turned on, and the color of the track line has been set to magenta.

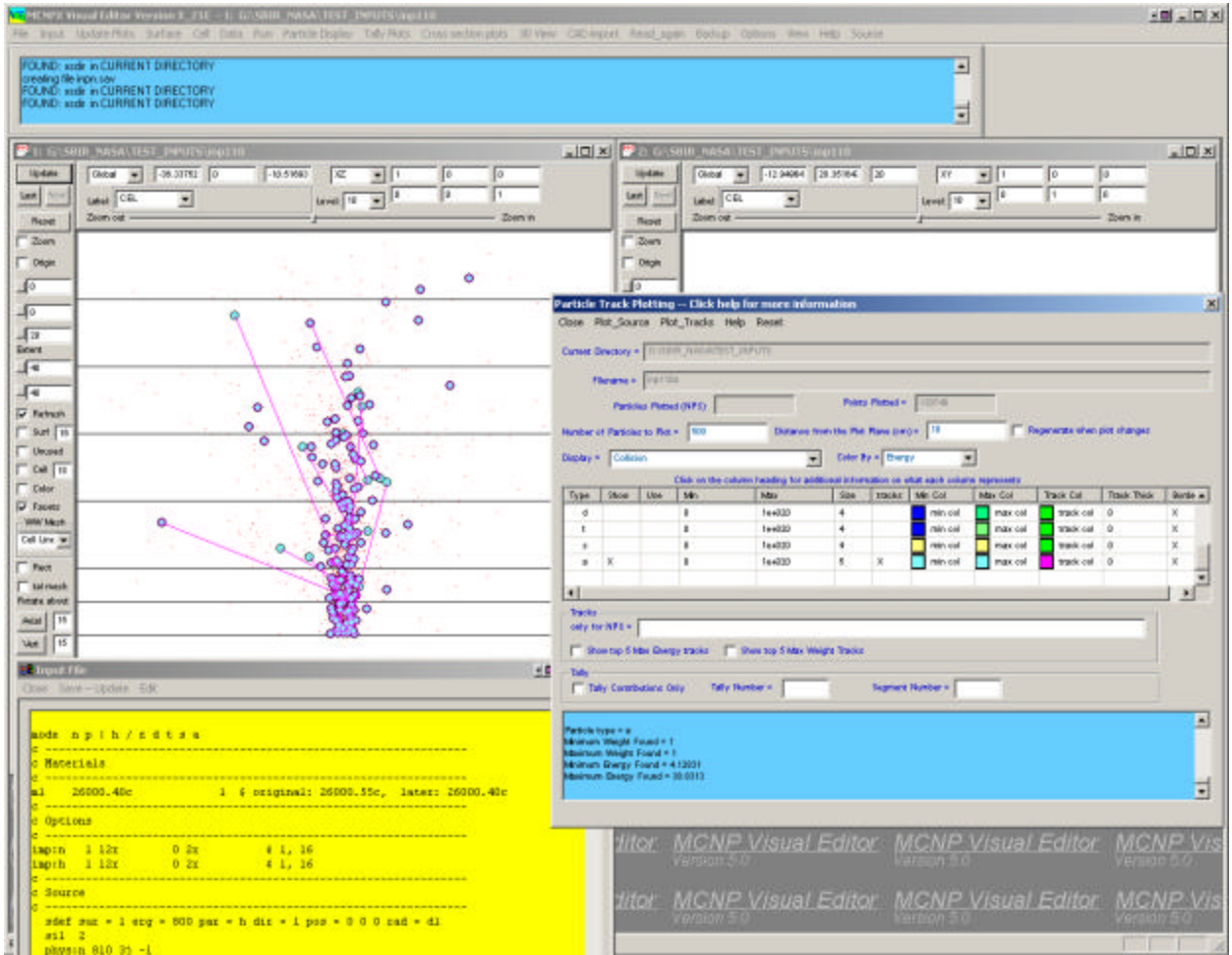


Figure 18. Plot of He-4 Collision Points Including Tracks.

Figure 19 shows photons in a cask with the particles colored by weight. The particles are biased through the cask shield and can be observed going from a high weight red color to a low weight blue color as the biasing in the shield lowers the weight.

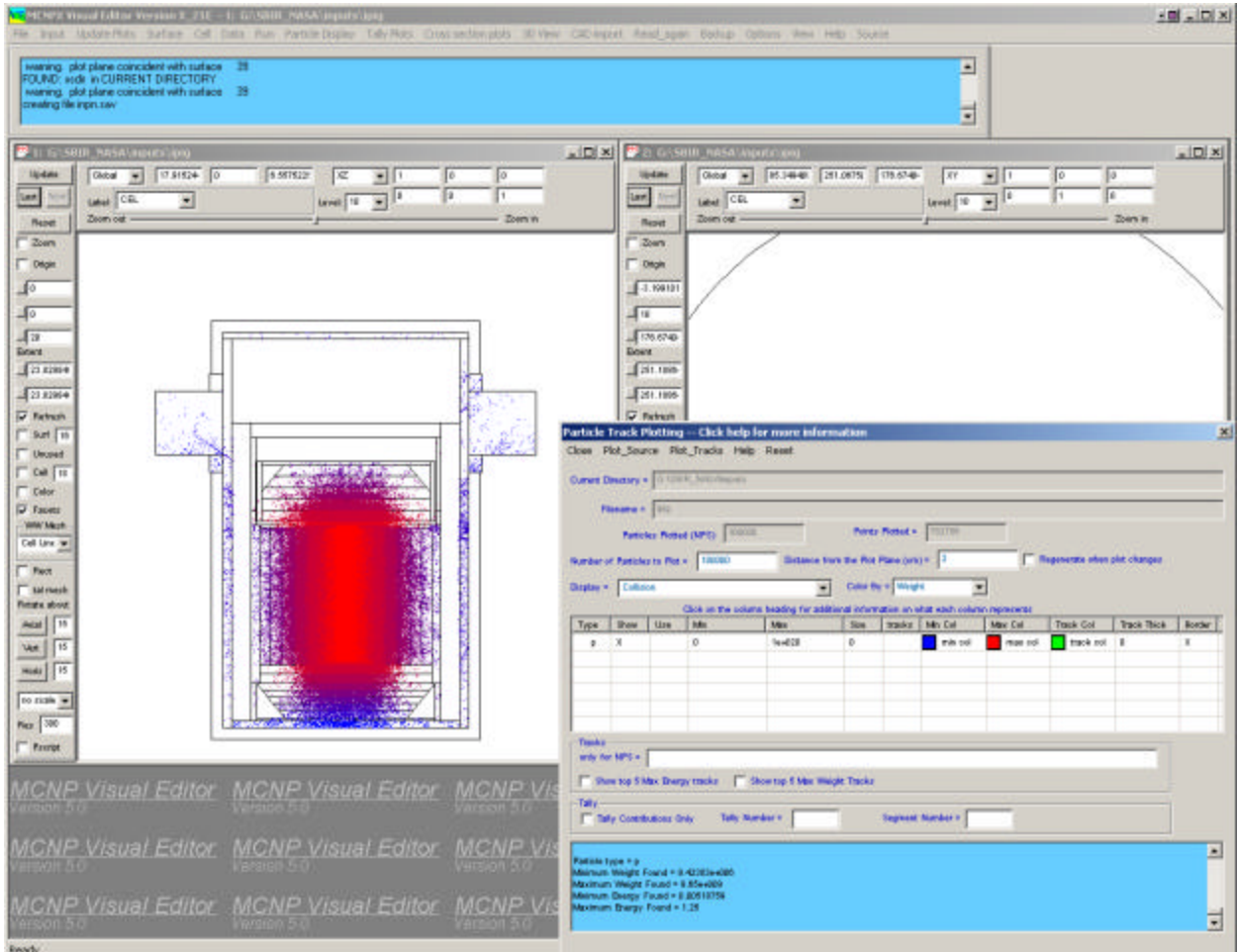


Figure 19. Particles Colored by Weight in a Cask.

In Figure 20, the “Tally Contributions Only” checkbox has been selected to only show those particle tracks that eventually lead to a contribution to a tally. The tally in this case is on the outside of the cask.

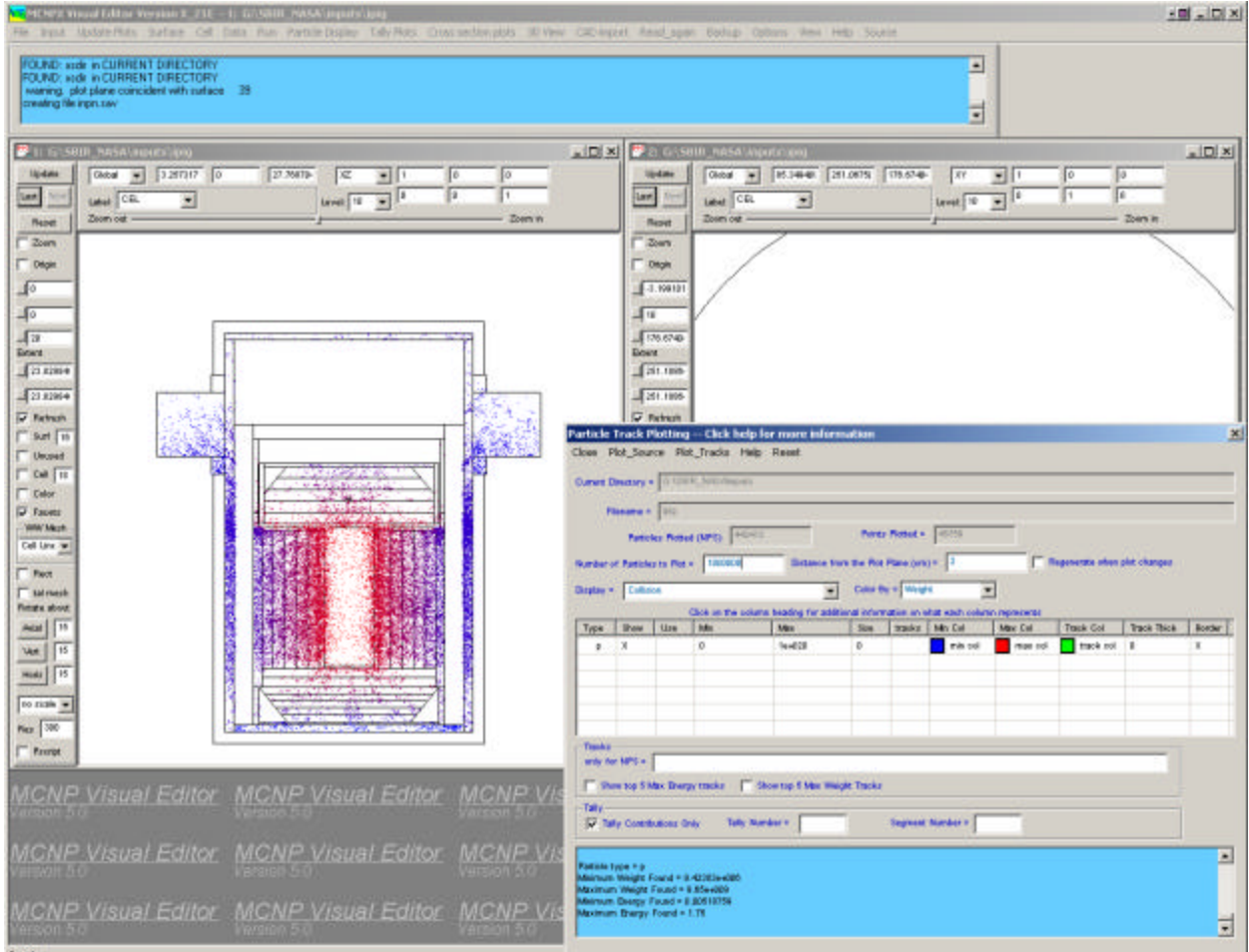


Figure 20. Particles that Contribute to a Tally Outside the Cask.

In Figure 21, the high weight particles have been selected by setting the range of weights to plot between $4e9$ and $1e10$. The weights are high because they have been multiplied by a weight scale factor in the source of $9.65e9$. High weight particles can be observed traveling through a steel streaming path (shown in yellow) and going to the outside tally region of the cask. Because the steel is less effective as a shield than the lead, the particles can go through this steel region more easily than through the lead cask.

This additional information indicates that the dose rate near the top of the cask should be higher than the dose rate outside the center of the cask. Additional calculations show that for this case, the dose rate is a factor of two higher in this region. This is an example of how being able to display particle tracks can lead to additional insight into the problem being analyzed.

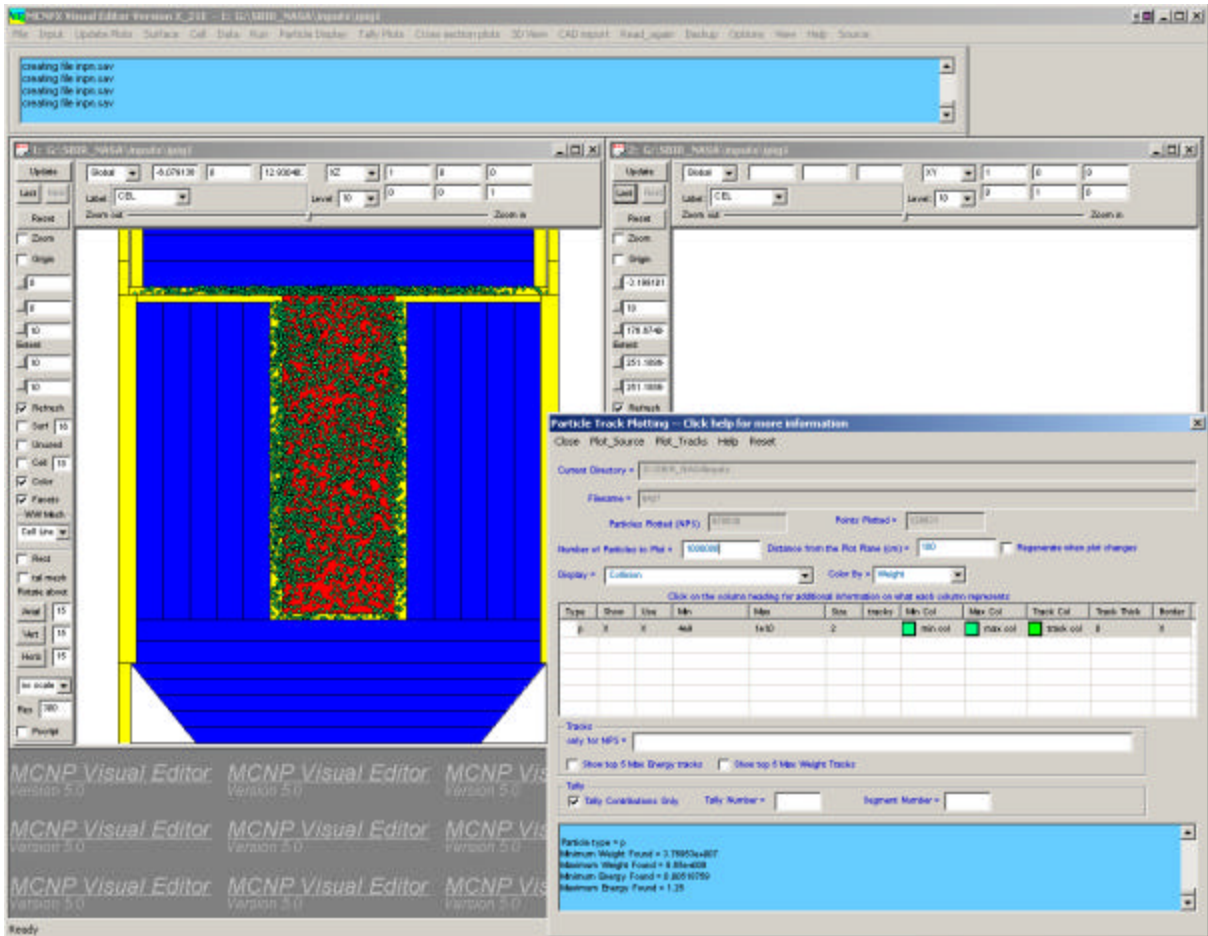


Figure 21. Plot of High Weight Particles Contributing to a Tally Region.

Figure 22 shows a cask containing cesium capsules with a cylindrical air chamber in the top left corner of the cask. This plot shows the source and collision points colored according to weight. The biasing of the source region can be observed. The left pin has more source than the right pin. The upper regions of both pins have more source points than the lower regions. The source biasing lowers the weight.

As the particles go through the steel shield, the particles are split using importance biasing that further reduces the weight. The end result is a plot showing the particles being biased towards the air chamber and changing from red to blue as they lose importance. This plot verifies that the particles are going in the expected direction.

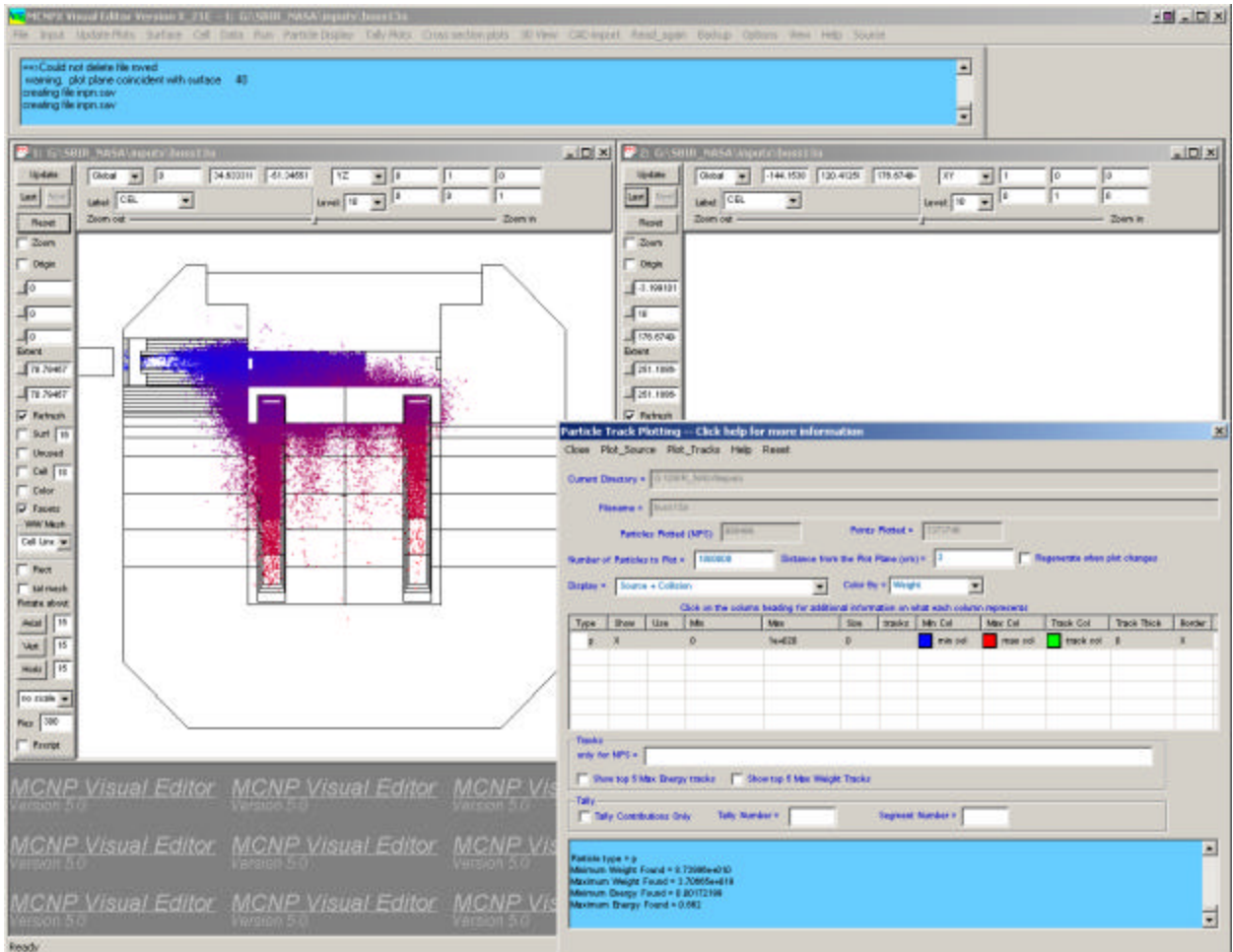


Figure 22. Plot of Biasing Particles Toward an Air Gap in a Cask.

Figure 23 shows the same cask but now only shows those particles that contribute to the detector outside the cask.

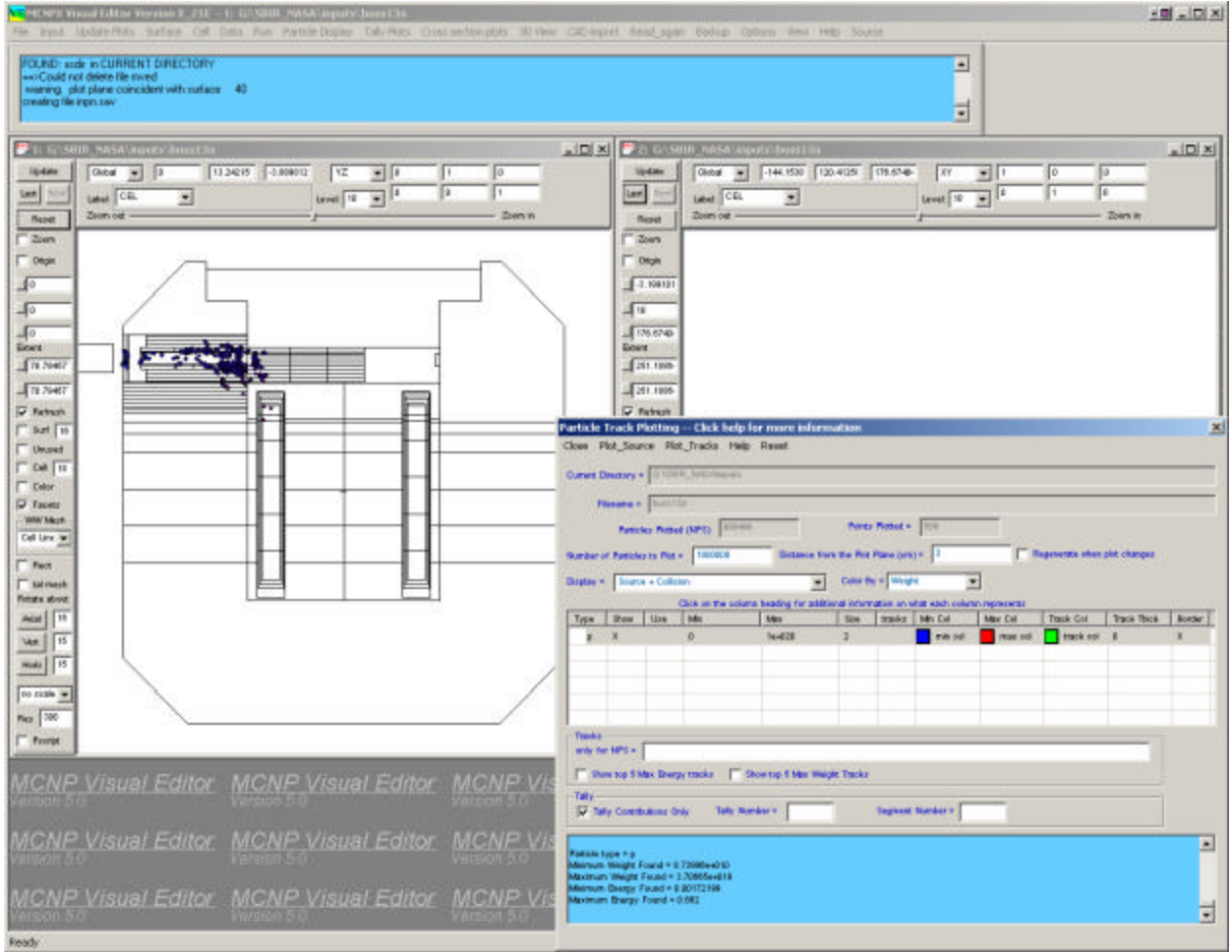


Figure 23. Plot of Particles that Contribute to the Tally Region.

7.1.4 Task 4: Add Source Creation

The creation of sources in the Visual Editor includes the reading, display, creation, and writing of source cards. The scope is initially limited to sdef cards. The source creation dialog is able to create and display source in much the same manner as the surface dialog can create and display surfaces. Work is continuing on a useful wizard that will assist in their creation for beginning users. This will be much like the surface wizard provides useful help in the creation of surfaces.

7.1.4.1 Reading the Source from an Input File

Complex sources can be read from input files. The Visual Editor can read si, sp, sb, ds, and sc cards. These cards are stored in memory and then used to populate the dialogs for display or for writing the source.

7.1.4.2 Display of Source Data

The sdfef line parameters are displayed on the “Create a Source” dialog. The distribution may be displayed by clicking the “D” button next to the appropriate variable. Distributions may be displayed either vertically or horizontally. Figure 24. Source Reading and Display Capabilities. shows a complicated source for a buss cask. The sdfef parameters are displayed in the top left dialog. The bottom left displays distribution 2. Distribution 3 is shown in the top right.

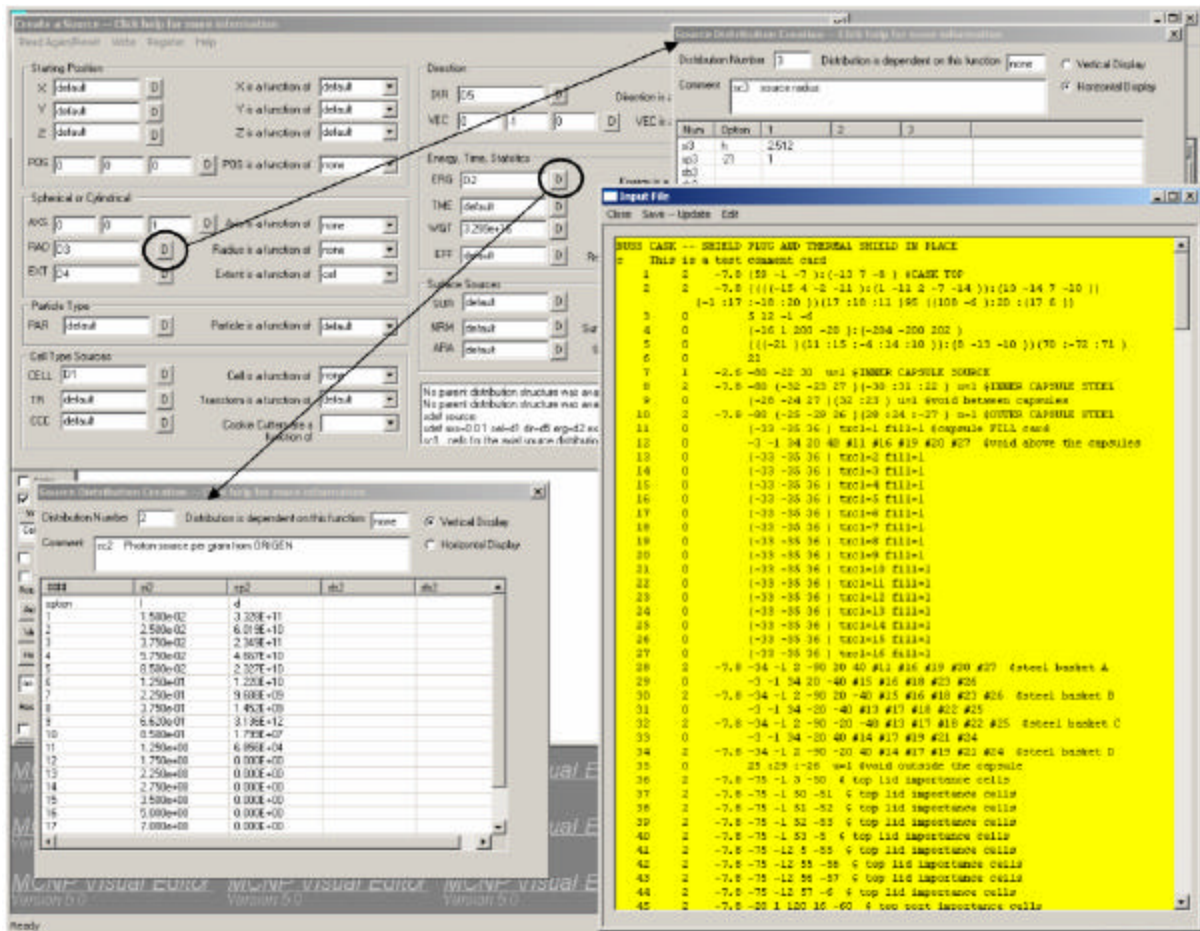


Figure 24. Source Reading and Display Capabilities.

From the distributions displayed, dependent distributions may be displayed by right-clicking on the parent distribution. In Figure 25 the sdfef card is displayed in the source creation dialog. Distribution 4 was displayed by clicking the button labeled with a “D” next to the EXT variable showing D4. Once Distribution 4 was displayed in a child dialog, Distribution 22 was displayed by clicking on the 22 in the Distribution 4 dialog.

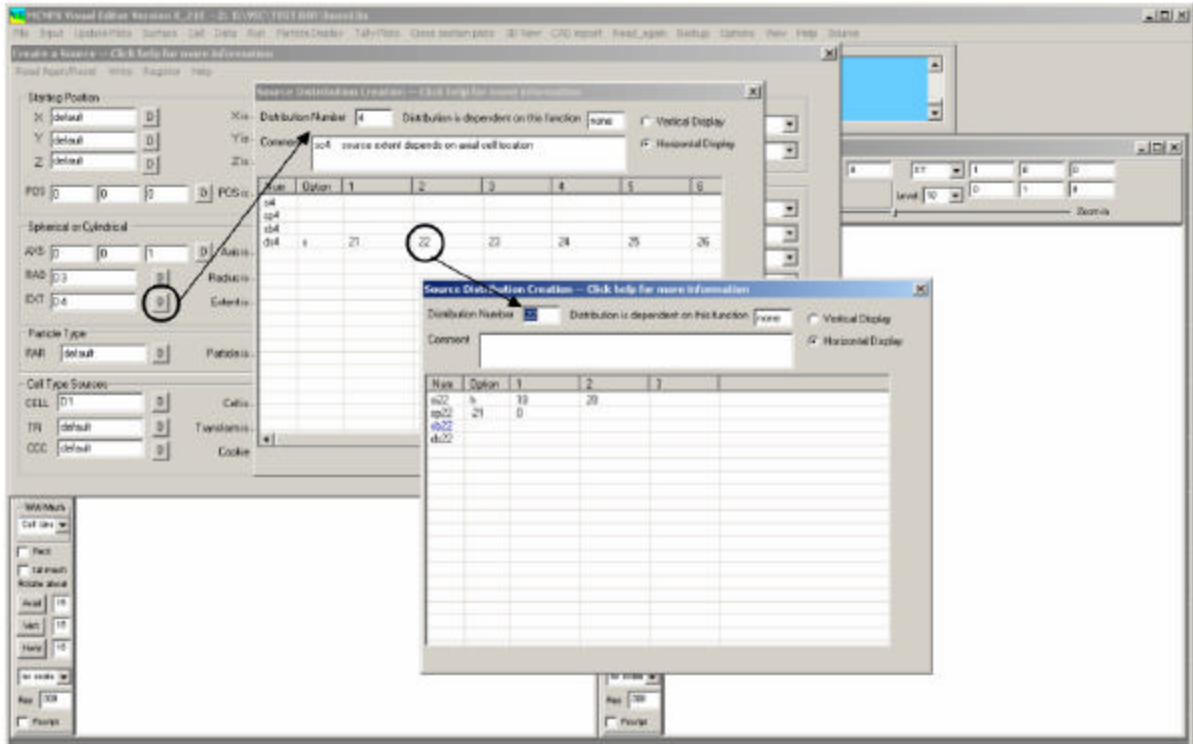


Figure 25. Dependent Distributions.

7.1.4.3 Source Creation

Most sources can now be created from the dialogs.

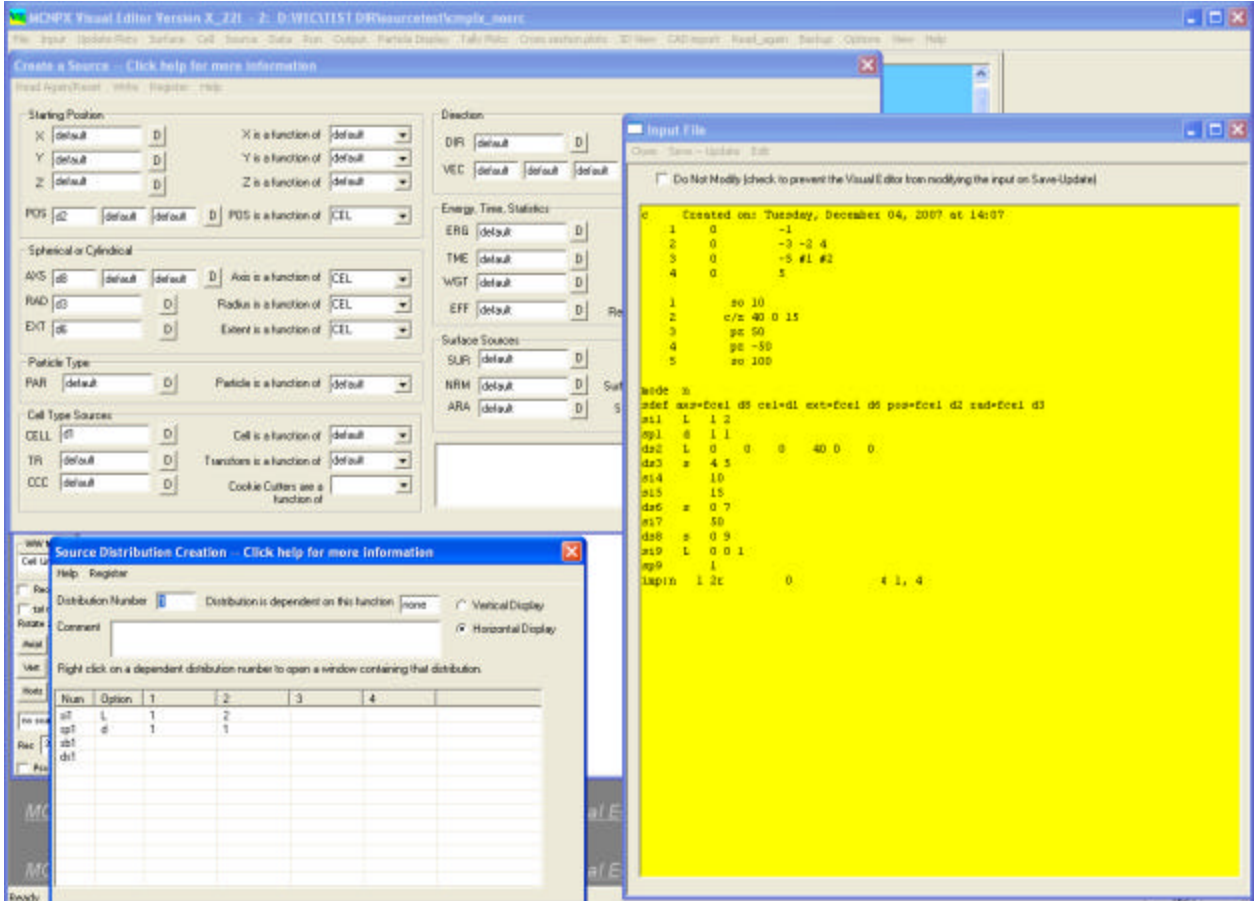


Figure 26. Creation of a Source.

In Figure 26, an sdef source with nine distributions was specified. This source was created entirely using the Visual Editor.

7.1.4.4 Source Writing

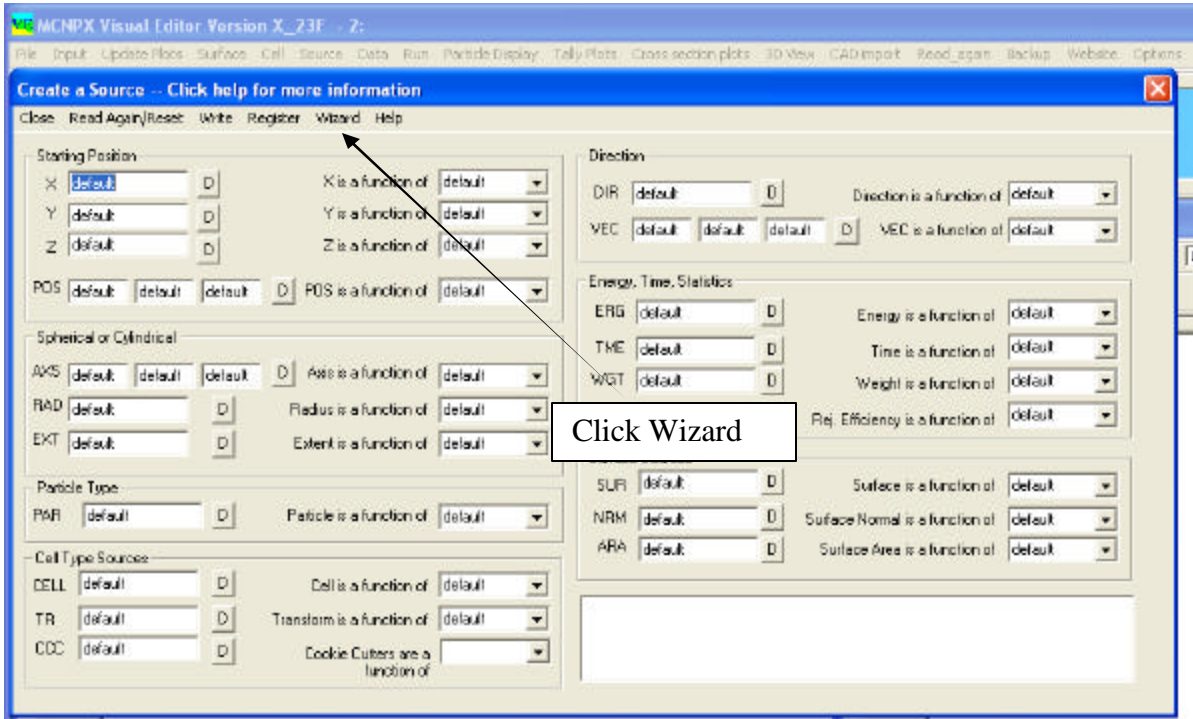
The source is now being written to the input file. A comparison of the two sources for a buss cask is shown in Table 7-1. Because comments designated by a “c” rather than an “sc” are not currently read, they are also not written. Distributions are written in numeric order. If they were originally in a different order in the input file, they are reordered.

Table 7-1. Comparison of Input File and Output Generated by Source Creation Program

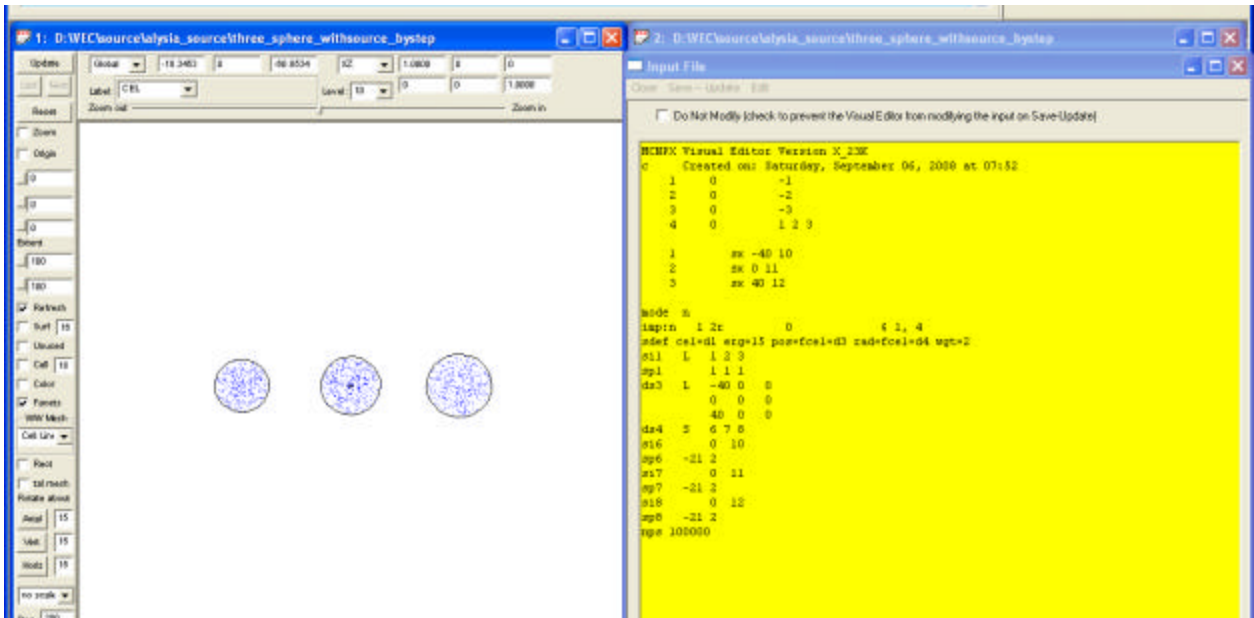
Source as Written by Program	Source from Input File
<pre>sdef source: sdef axs=0 0 1 cel=d1 dir=d5 erg=d2 ext=fcel d4 pos=0 0 0 rad=d3 vec=0 -1 0 wgt=3.299e+16 sc1 cells for the axial source distribution si1 s 11 12 13 14 15 16 17 sp1 157.957 198.239 198.239 198.239 198.239 99.1195 39.648 42.6016 sb1 .001 .005 .05 .1 0.25 0.5 1 sc2 Photon source per gram from ORIGEN # si2 sp2 1 1.500e-02 3.328E+11 2 2.500e-02 6.019E+10 3 3.750e-02 2.349E+11 4 5.750e-02 4.667E+10 5 8.500e-02 2.327E+10 6 1.250e-01 1.220E+10 7 2.250e-01 9.688E+09 8 3.750e-01 1.452E+09 9 6.620e-01 3.136E+12 10 8.500e-01 1.799E+07 11 1.250e+00 6.856E+04 12 1.750e+00 0.000E+00 13 2.250e+00 0.000E+00 14 2.750e+00 0.000E+00 15 3.500e+00 0.000E+00 16 5.000e+00 0.000E+00 17 7.000e+00 0.000E+00 18 1.100e+01 0.000E+00 sc3 source radius si3 2.512 sp3 -21 1 sc4 source extent depends on axial cell location ds4 s 21 22 23 24 25 26 27 sc5 angle distribution si5 h -1 0 1 sp5 d 0 1 1 sb5 d 0 1 100 sc6 cells for each pin location si6 1 11 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 sp6 1 15r sb6 25 50 50 25 10 100 10 10 10 10 10 10 50 50 10 10 si11 1 d6:7 sp11 1 si12 1 d6:53 sp12 1 si13 1 d6:54 sp13 1 si14 1 d6:55 sp14 1 si15 1 d6:56 sp15 1 si16 1 d6:57 sp16 1 si17 1 d6:58 sp17 1 si21 2.032 10 sp21 -21 0 si22 10 20 sp22 -21 0 si23 20 30 sp23 -21 0 si24 30 40 sp24 -21 0 si25 40 45 sp25 -21 0 si26 45 47 sp26 -21 0 si27 47 49.149 sp27 -21 0</pre>	<pre>sdef cel = d1 erg = d2 rad = d3 ext = fcel d4 pos 0 0 0 axs = 0 0 1 wgt = 3.299e+16 vec = 0 -1 0 dir=d5 c 3.858E+12 per gram * 8550.56 = 3.299e+16 c 744.07KCi / 87.02/gram (Cs 137) = 8550.56 sc1 cells for the axial source distribution si1 s 11 12 13 14 15 16 17 sp1 157.957 198.239 198.239 198.239 99.1195 39.648 42.6016 sb1 .001 .005 .05 .1 0.25 0.5 1 c TOTAL PHOTON SPECTRUM sc2 Photon source per gram from ORIGEN # si2 sp2 1 1.500e-02 3.328E+11 2 2.500e-02 6.019E+10 3 3.750e-02 2.349E+11 4 5.750e-02 4.667E+10 5 8.500e-02 2.327E+10 6 1.250e-01 1.220E+10 7 2.250e-01 9.688E+09 8 3.750e-01 1.452E+09 9 6.620e-01 3.136E+12 10 8.500e-01 1.799E+07 11 1.250e+00 6.856E+04 12 1.750e+00 0.000E+00 13 2.250e+00 0.000E+00 14 2.750e+00 0.000E+00 15 3.500e+00 0.000E+00 16 5.000e+00 0.000E+00 17 7.000e+00 0.000E+00 18 1.100e+01 0.000E+00 c TOTAL 3.858E+12 per gram sc3 source radius si3 2.512 sp3 -21 1 sc4 source extent depends on axial cell location ds4 s 21 22 23 24 25 26 27 sc5 angle distribution si5 h -1 0 1 sp5 d 0 1 1 sb5 d 0 1 100 sc6 cells for each pin location si6 1 11 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 sp6 1 15r sb6 25 50 50 25 10 100 10 10 10 10 10 10 50 50 10 10 si11 1 d6:7 sp11 1 si12 1 d6:53 sp12 1 si13 1 d6:54 sp13 1 si14 1 d6:55 sp14 1 si15 1 d6:56 sp15 1 si16 1 d6:57 sp16 1 si17 1 d6:58 sp17 1 si21 2.032 10 sp21 -21 0 si22 10 20 sp22 -21 0 si23 20 30 sp23 -21 0 si24 30 40 sp24 -21 0 si25 40 45 sp25 -21 0 si26 45 47 sp26 -21 0 si27 47 49.149 sp27 -21 0</pre>

7.1.4.5 Source Wizard

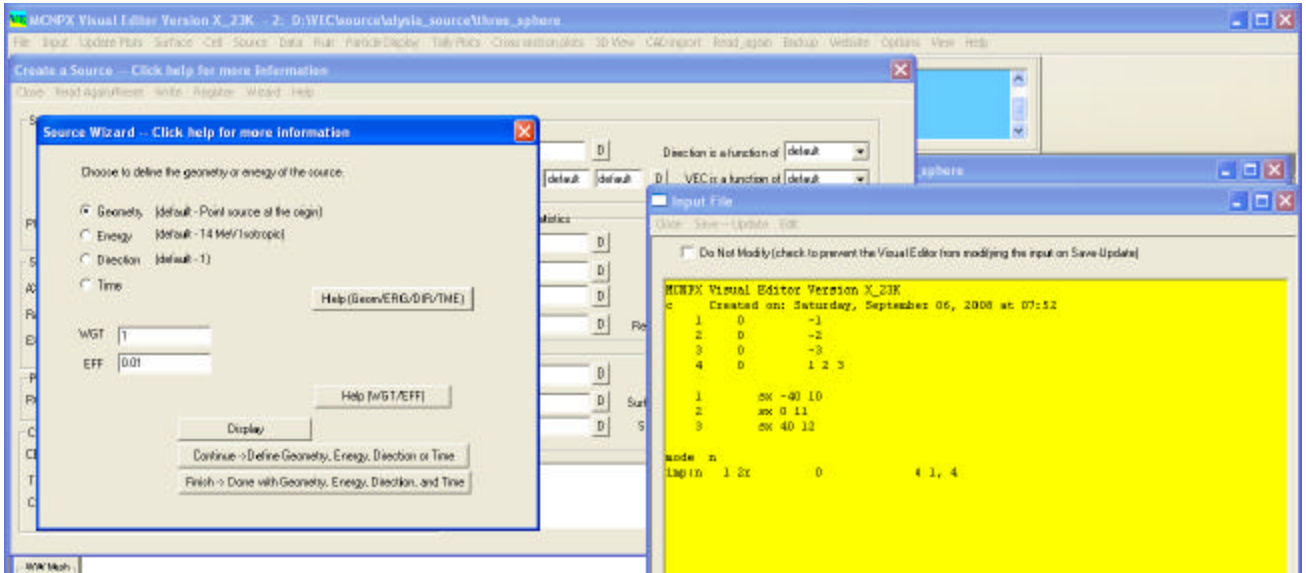
The wizard supports the creation of several types of sources. Points, spheres, cylinders, lines, areas, boxes, and surfaces are included in the scope of this wizard.



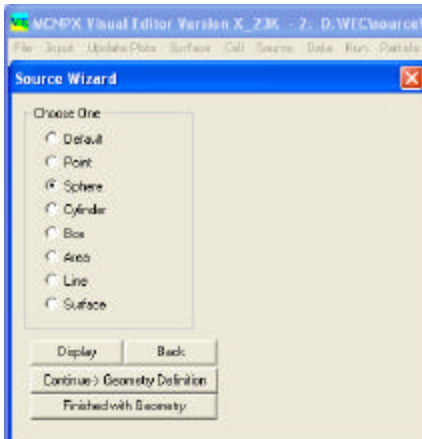
As an example on how to use the wizard, it will be used to create a source in three spheres as shown in the plot below.



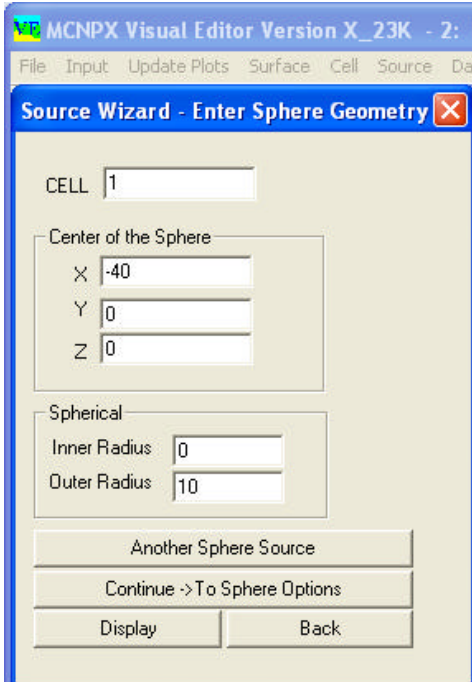
To start the process, from the main source creation panel, **Click Wizard**.



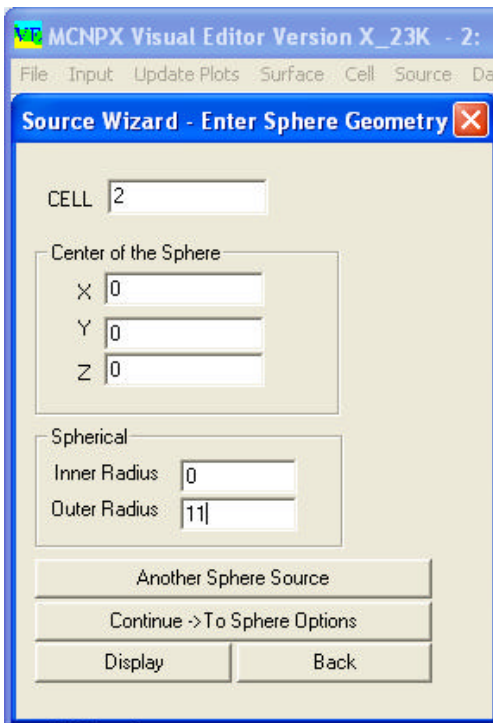
The initial panel lets the user select what part of the source is to be defined. In this example a source will be created in three spheres defined by cells 1, 2, and 3. Click the **Continue->Define Geometry, Energy, Direction or Time** button to go to the next panel. The panel below will appear.



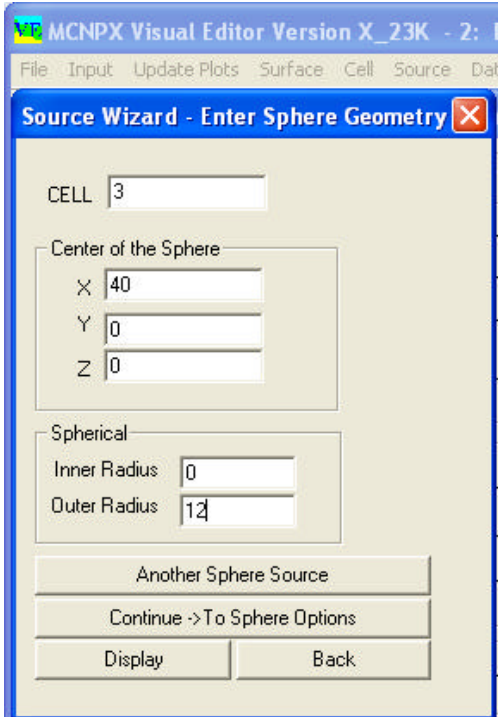
Verify that sphere has been checked and then select **Continue->Geometry Definition** to go to the next panel.



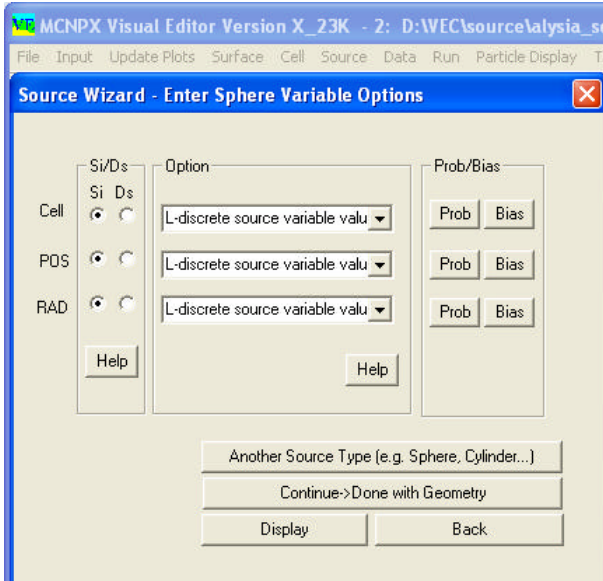
Here the cell number and position of the first sphere must be entered along with the radius. Click **Another Sphere Source** and enter the information for the second sphere as shown below.



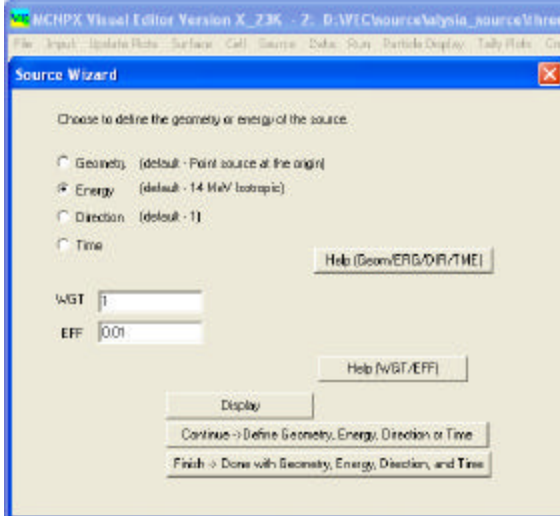
Notice the cell number, position and radius must be entered. Select **Another Sphere Source** and enter information for the final sphere as shown below.



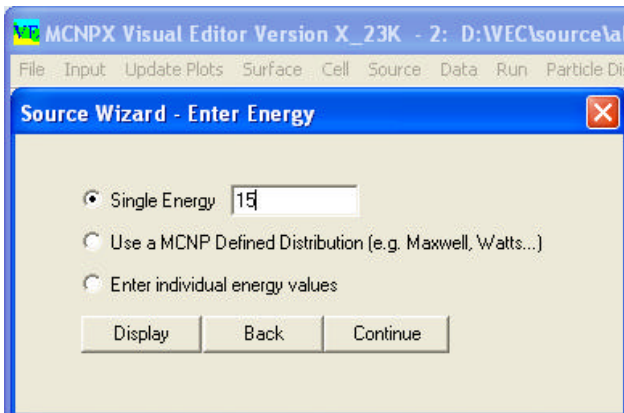
Now that the three sphere source regions have been defined, click **Continue->To Sphere Options**.



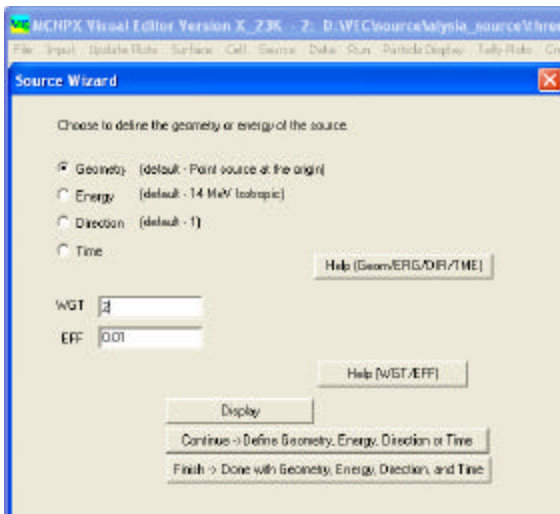
Verify that all distributions are Si distributions. The option for all three distributions should be L, which is supplied as a default on this panel. Click **Continue->Done with Geometry**.



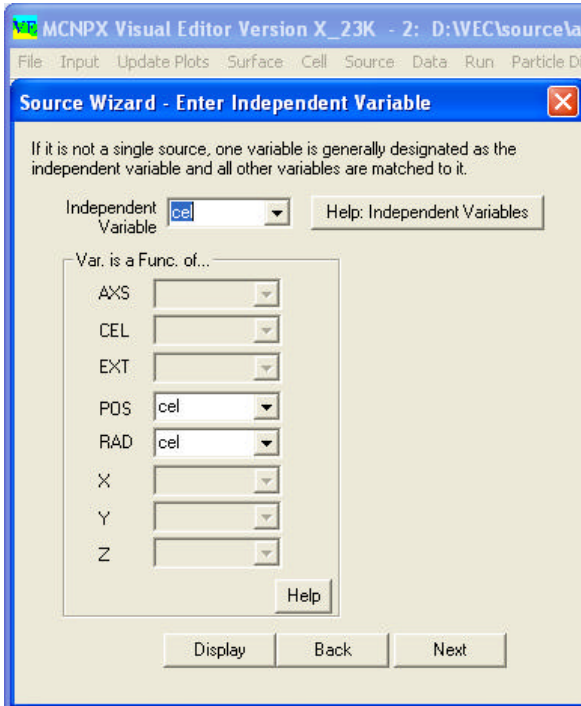
Click to select Energy. Click **Continue -> Define Geometry, Energy, Direction, or Time**.



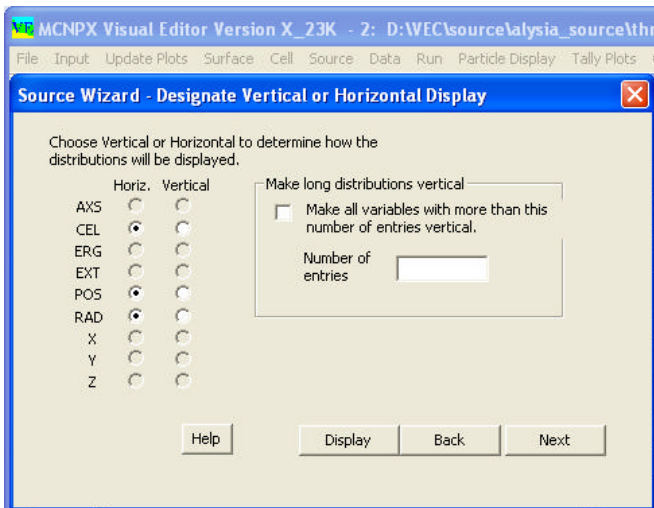
On the Energy Panel, Change the Single Energy entry to 15 MeV from the 14 MeV default. Click **Continue**.



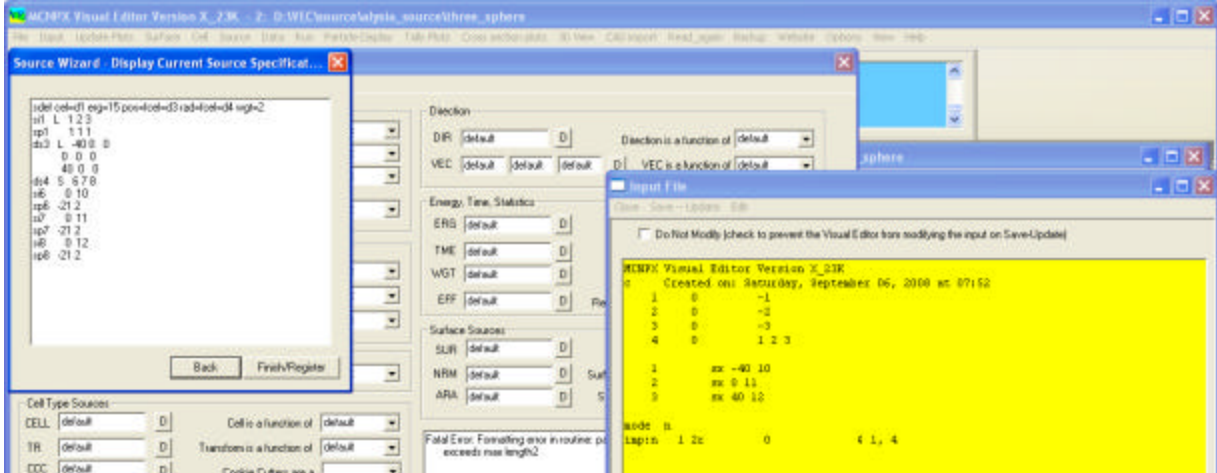
On the Source Wizard Main panel, change the weight (WGT) to 2. Click **Finish->Done with Geometry, Energy, Direction, and Time**.



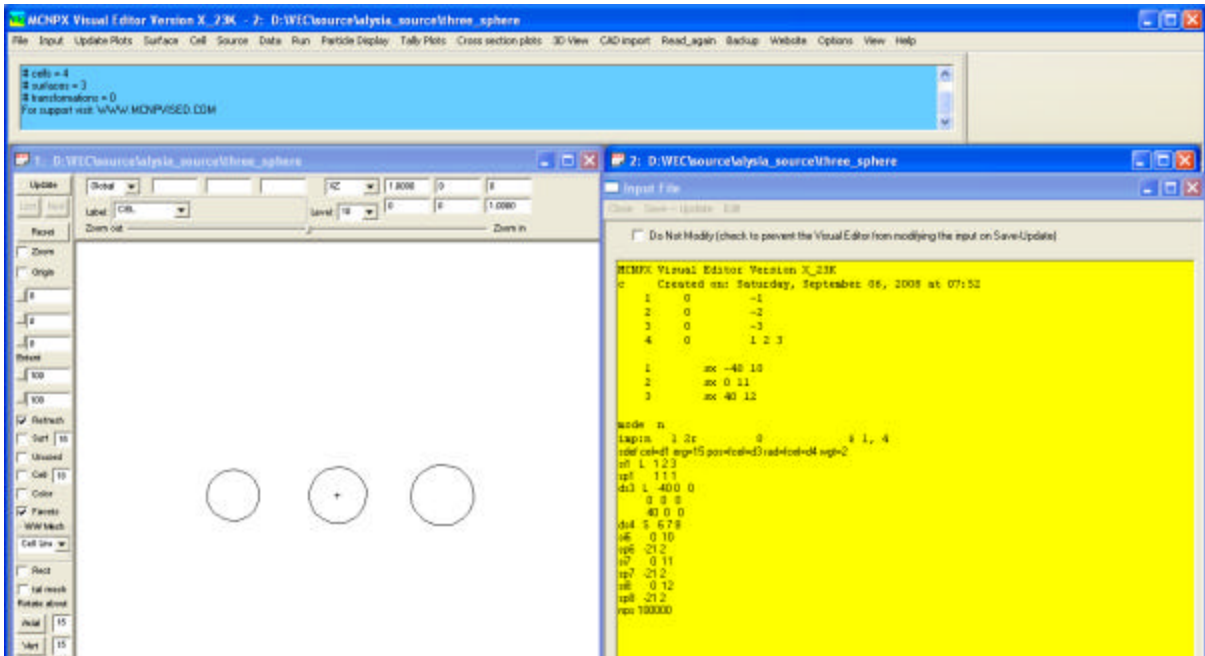
On the Independent Variable panel, Select Cel. The other variables (POS and RAD) are automatically assigned CEL for their functional dependency. Variables that have only a single entry or are not used and disabled. Click **Next**.



This panel controls the display of the variables in the input file. Verify that all distributions are marked horizontal. Click **Next**.



The wizard displays the resulting input file. Currently, the source definition must be manually copied into the input file. This is being done while the wizard is still being tested.



Copy the source string at the bottom of the input file. Add the following line after it:

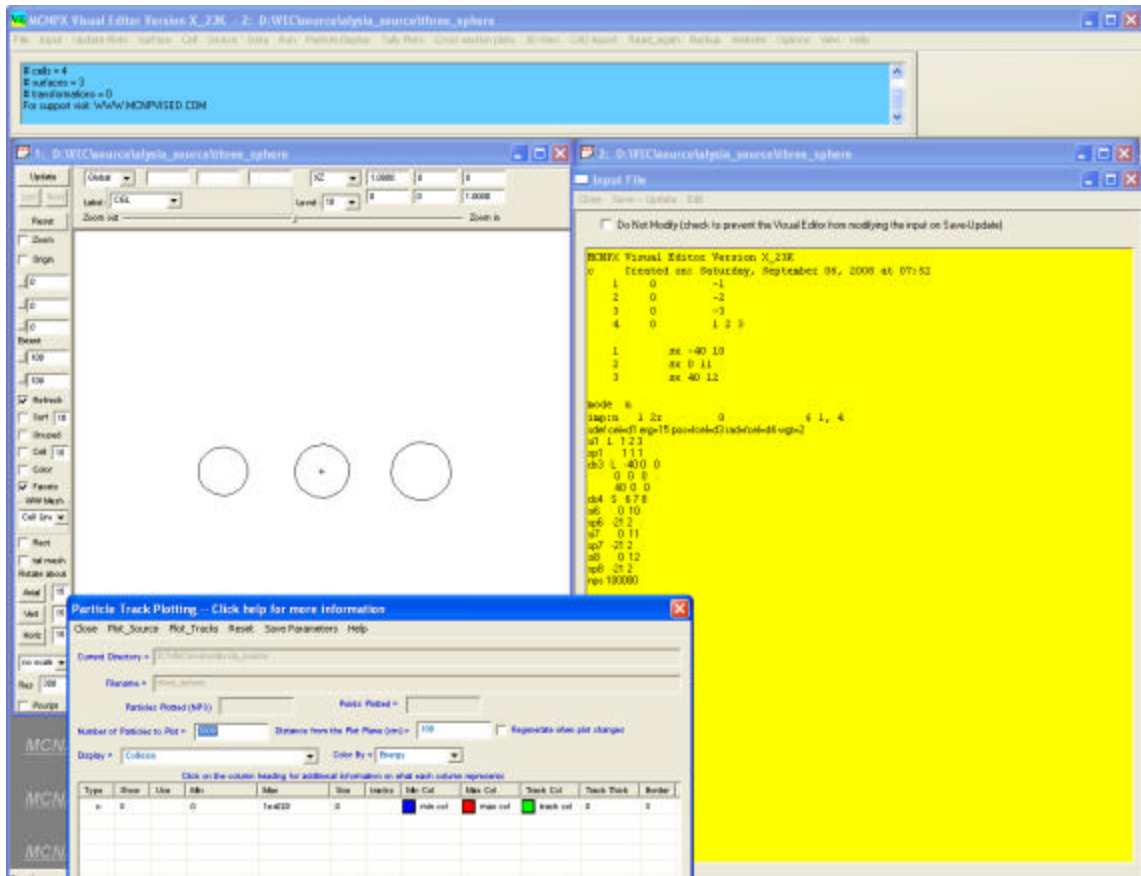
nps 100000

Verify that there is a blank line at the end of the input file.

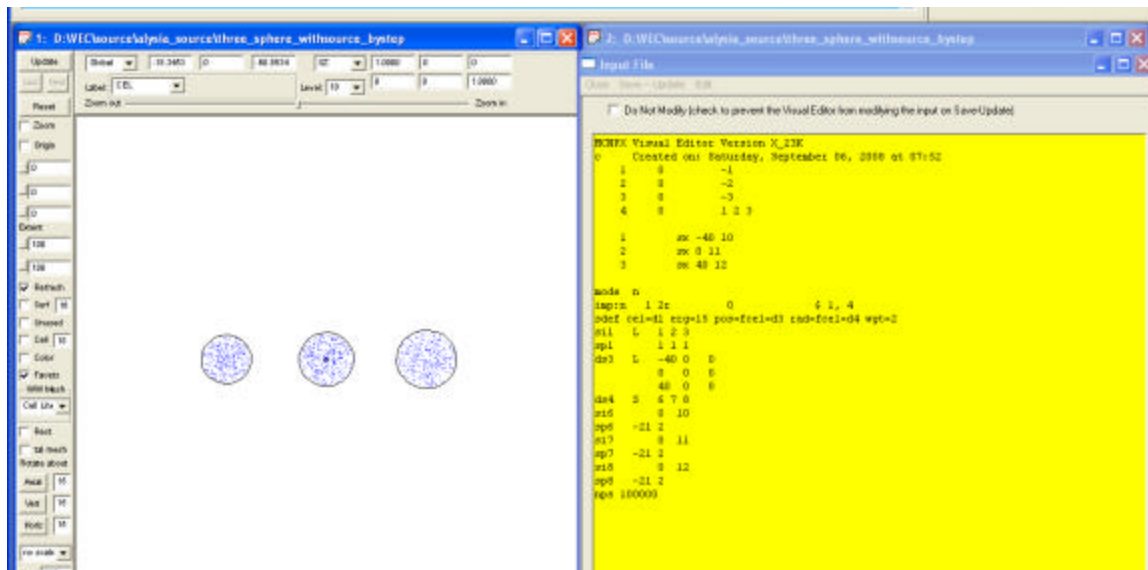
Click **Save-Update**.

Click **Particle Display**.

Click **Plot Particle Tracks**.

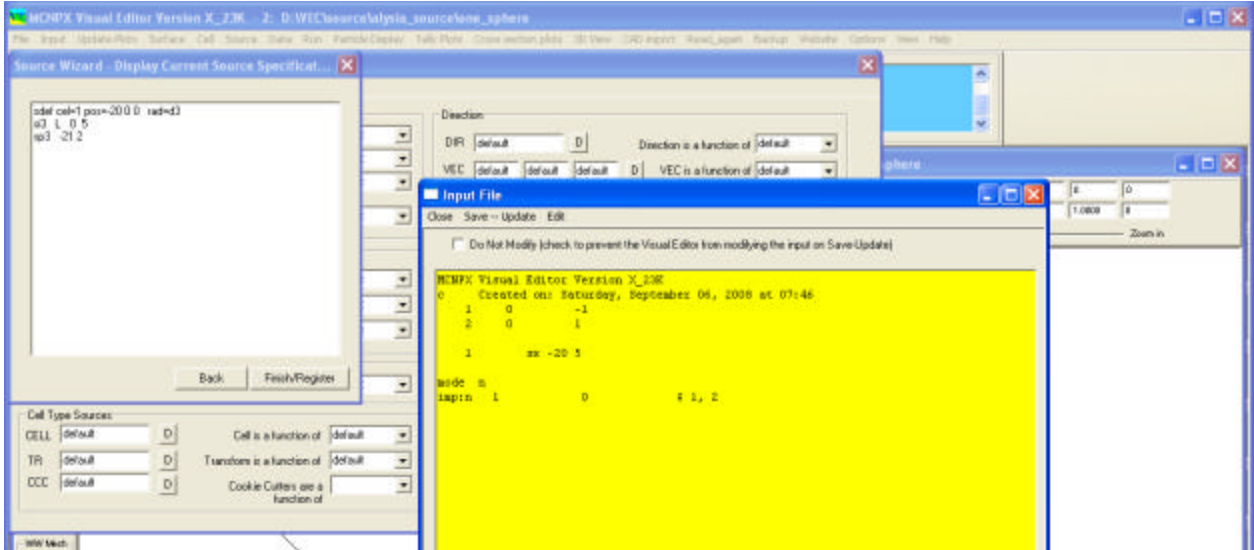


Click **Plot_Source**.

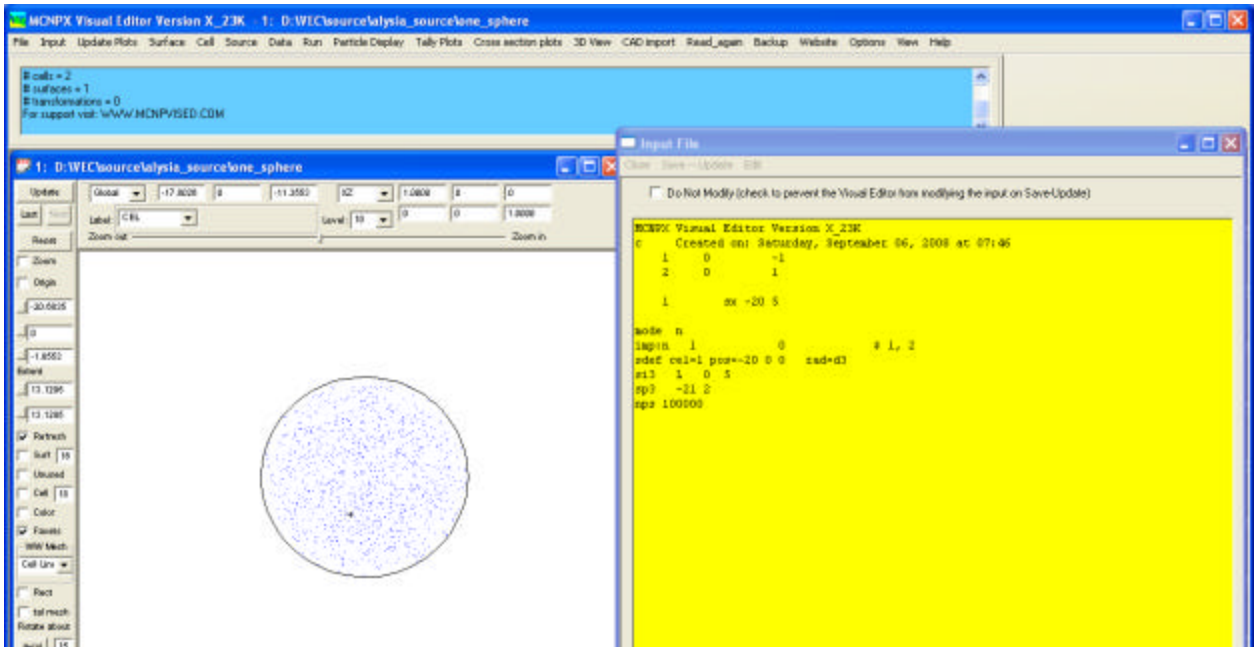


The source wizard can now handle most geometries containing spheres, cylinders, and boxes. It also allows for entry of an energy spectrum. Below are some examples of input files and sources that have been successfully created in the Source Wizard. In each case, the source, once inserted into the input file, is plotted using particle track plotting.

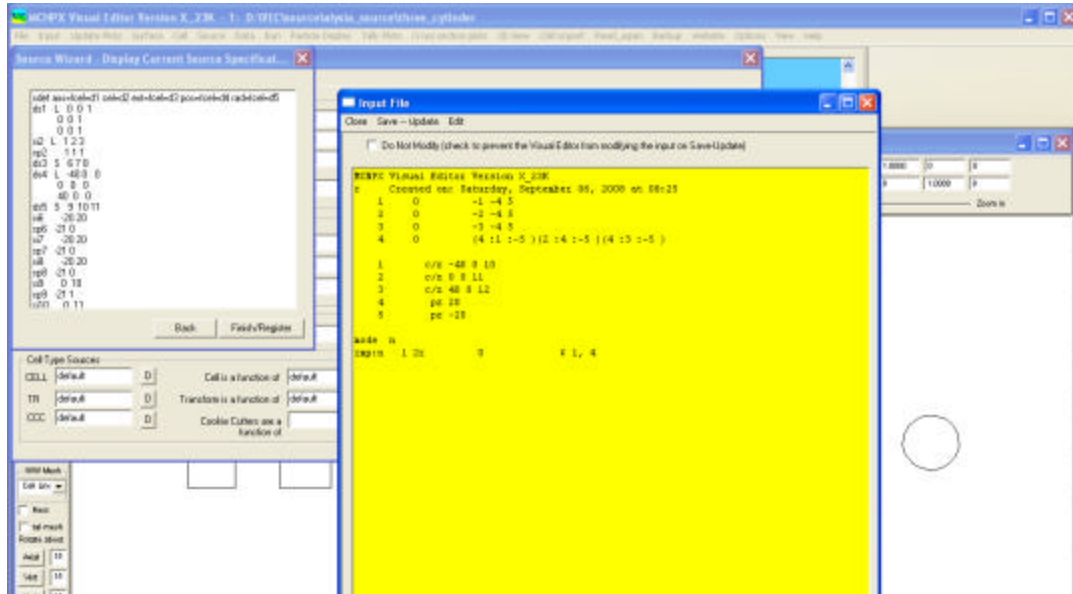
The plot below shows the source for 1 sphere.



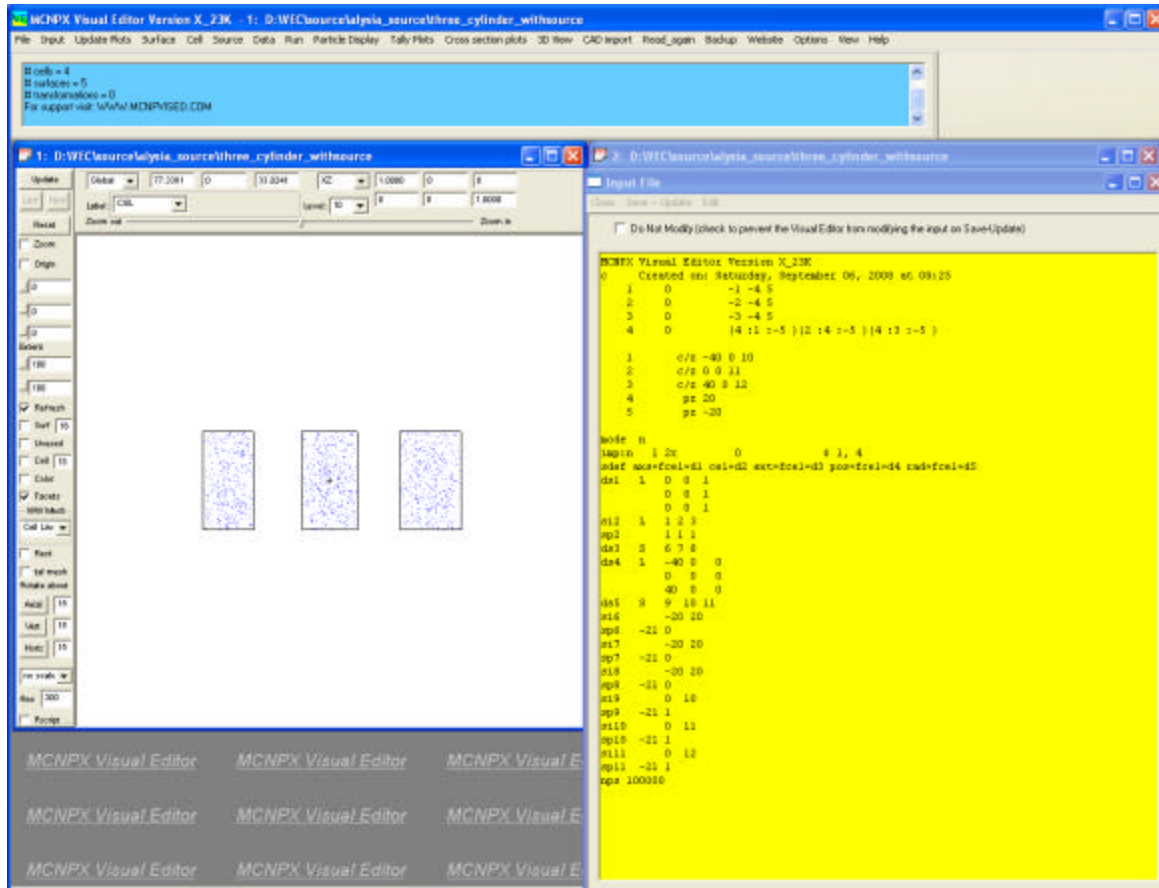
Once the source is inserted into the input file, the following collision plot can be made.



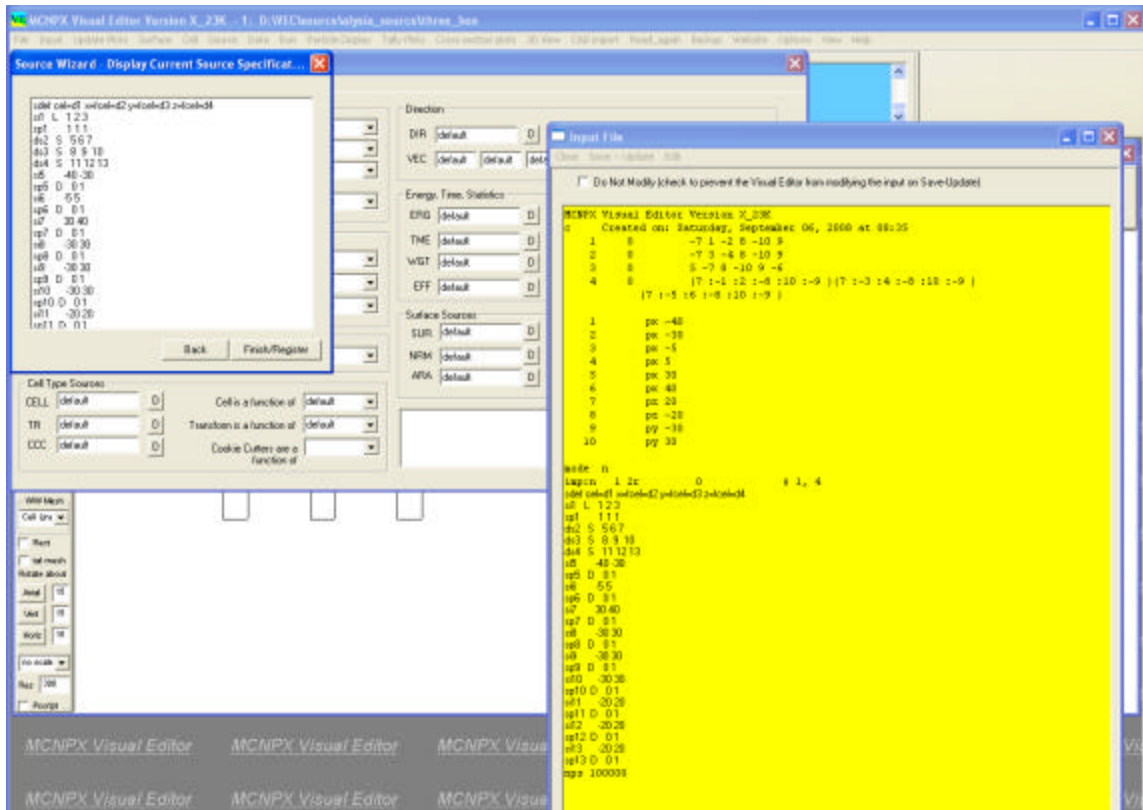
Below is an example of a source for three cylinders created by the wizard.



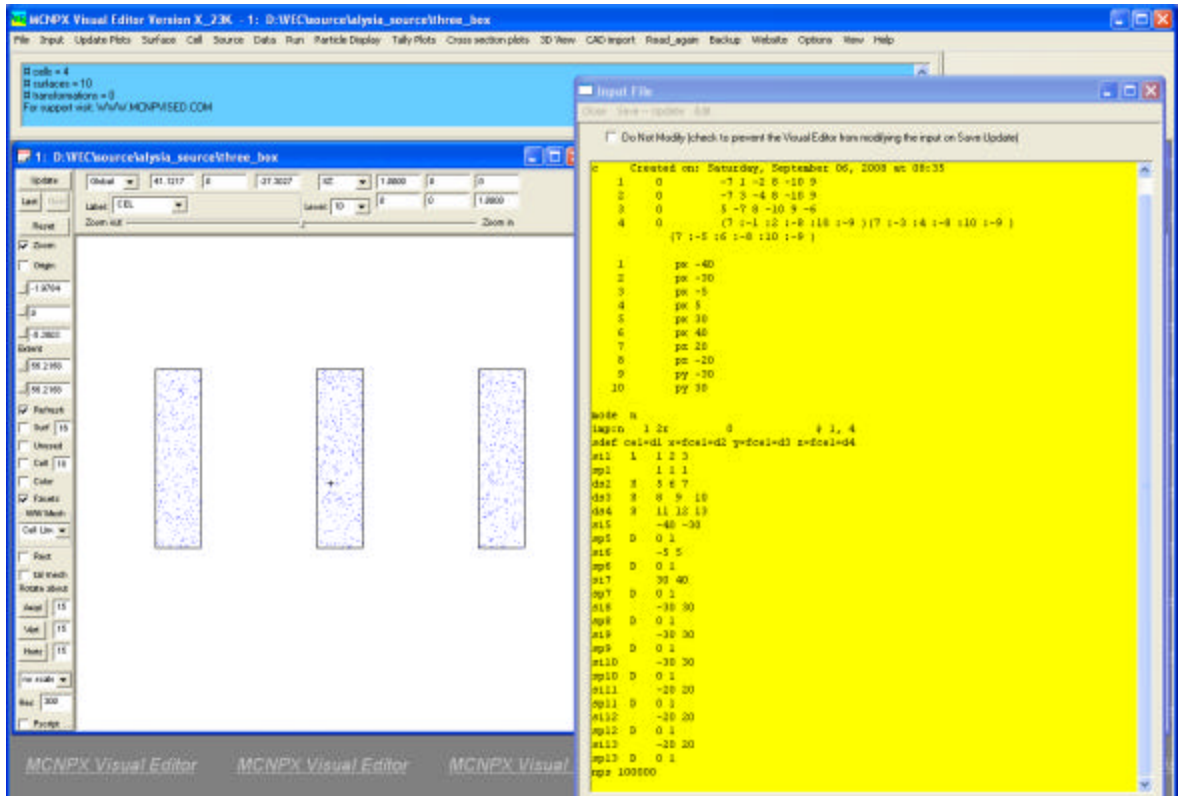
Once the source is placed in the input file, collision points can be plotted inside the cylinders as shown below.



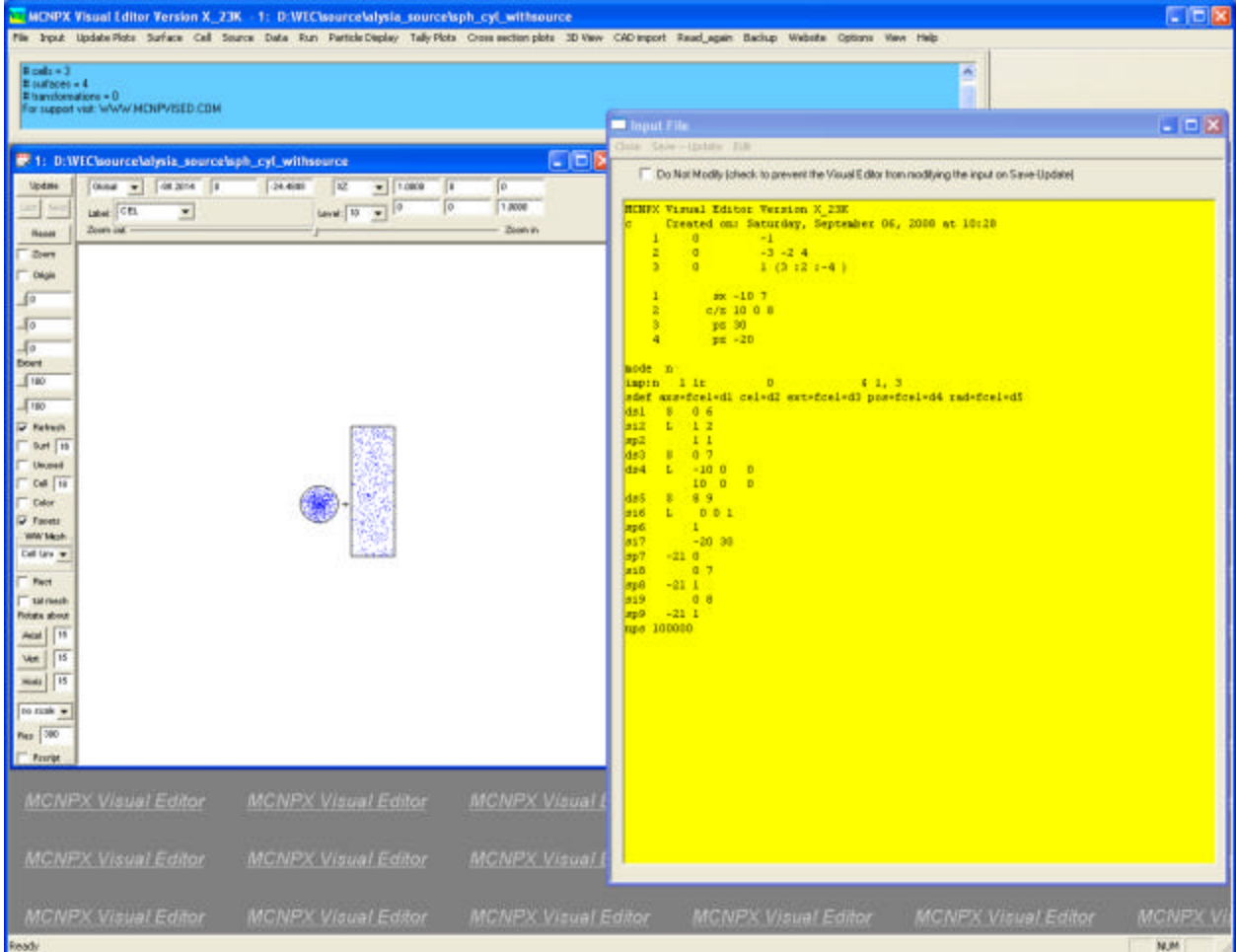
Below is an example of a source with three boxes created by the wizard.



Once the source is placed in the input file, collision points can be plotted inside the boxes as shown below.



The wizard can even handle mixed geometry source configurations. The example below shows both a cylinder and a box with the source defined inside both cells.



7.1.5 Task 5: Investigate the Implementation of Spline Surfaces in MCNPX

7.1.5.1 Introduction and Conclusion

A preliminary, general fourth order surface has been implemented in the MCNPX Visual Editor. Such a fourth order surface treatment may be useful in expanding the geometry capabilities of MCNPX/MCNP and is being examined with a preliminary implementation in the Visual Editor in this regard. The MCNPX/MCNP geometry description is currently based on second order surfaces, except for an optional fourth order torus surface with a requirement that the major axis be parallel to either x, y, or z. A much more general preliminary fourth order surface implementation in the Visual Editor is summarized in this section. Section 7.1.5.2, "Plots of Fourth Order Surfaces," presents plots of the fourth order surface obtained by varying the coefficients of the fourth order surface and considers why such surface descriptions could be useful. This summary could form a basis for a future full implementation of fourth order surfaces in MCNPX/MCNP.

The fourth order surface, G4, that has been implemented in the Visual Editor for testing, is much like the general quadratic GQ surface currently in MCNP for a general second order surface,

except the G4 surface includes all combinations of terms up to and including fourth order, i.e., $(x-x)^4$, $(x-x)^3$, $(x-x)^3(y-y)$, $(x-x)^2(y-y)^2$,etc. These 35 terms with a user-supplied coefficient for each term to define a desired surface are presented in Section 7.1.5.6, "Appendix: Details of the Preliminary Fourth Order Surface Implementation". This section also includes details about the preliminary implementation in the Visual Editor of MCNPX, including a ray tracing method to obtain the 2D plots.

A full fourth order surface implementation would presumably aid the user by allowing for a variety of simplified surface forms similar to what is available in MCNPX/MCNP for second order surfaces, i.e., the SO, S, SX, SY, SZ sphere surface types, the six cylinder surface types, and the six cone surface types. Such a choice of surface types enables ease of user input for second order surfaces compared to the more general SQ or GQ surface types. Similar advantages would likely occur for fourth order surfaces, and, in fact, may be even more important because such choices would presumably give the user insights into the types of fourth order surfaces that might be useful in a specific application.

7.1.5.2 Plots of Fourth Order Surfaces

Figure 27 is a plot of a "g4" surface that has second order terms describing a sphere of radius 10 cm, and the coefficient #1 for the x^4 term is included with a large enough value so the radius of the sphere is reduced on the + and - x ends from 10 cm to about 8.75 cm. Therefore, this fourth order surface is similar to an ellipse in the (x,y) view. The ray tracing utilized to obtain this plot starts 100 particles uniformly along the left edge of the plot in the +horizontal direction and plots short lines along the surface at each crossing of the surface. Consequently, this plot has an essentially continuous line for small y values and only has interspaced lines near the top and bottom. Note that the input file shown at the bottom of the figure has no cells and only has the one "g4" surface for this plotting of the fourth order surface.

Figure 28 is a plot of a "g4" surface that is similar to that of Figure 27 except the coefficient #16 for the x^3 term is utilized along with a reduction in coefficient #1 for the x^4 term in such a way that the radius is still about "8.75" at +x when y and z are zero, but at the same time the radius is "10" at -x when y and z are zero so it retains more of a spherical shape in the -x direction.

The plot presented in Figure 29 is the same as the plot presented in Figure 28 except the ray tracing source along the left of the plot is replaced with a circular inward source of radius 100 cm. This gives a much more uniform distribution of short lines for the surface for a near circular surface. The patch file was upgraded from pvisedc.14 to pvisedc.15 for this circular sampling with the G4 plotting flag negative as "iwins(32)=-jsu" in the patch file. This circular source option is used for the subsequent plots and is the recommended option. It can be changed to the other option by simply changing "iwins(32)=-jsu" to "iwins(32)=+jsu" in the pvisedc.15 patch file and recompiling. All of the subsequent plots use this inward circular source for the ray tracing.

Figure 30 is a plot of a "g4" surface representing a torus (axis parallel to x-axis) with a major axis radius of 50 and a minor axis radius of 20. The corresponding MCNP TX surface for a torus was used to determine the 35 coefficients for the "g4" surface. The approximate dimensions

along the two axes, obtained with the mouse on the plot, are as expected. This plot is a good test for confirming that the main new subroutine fourg4(), which is used to obtain the intersection of a random walk track with the surface, is functioning as expected. Note that on the plot the density of lines is greater for the smaller radius because the ray tracing source is inward in the direction of the origin.

The plot presented in Figure 31 is the same as the plot presented in Figure 30, except the axis of the torus is now along the 45 degree $x=y$ line with $z=0$, i.e., this is a torus with axis not along a major axis so the coefficients for the “g4” surface have been transformed appropriately and the basis vectors for the plot have been changed appropriately.

The plot presented in Figure 32 is a possible practical application of a fourth order surface, where only fourth $[x^4, y^4, z^4]$ terms are used in this “g4” surface along with the constant. This results in an approximate cube that is nearly flat on the edges with rounded corners. A similar approximate square shape could be obtained by setting the axial term to zero. This approximate square shape could have application for this type of duct penetrating the wall and would be easy to describe with the fourth order surface. The transition to more rounded sides could be obtained by including second $[x^2, y^2, z^2]$ order terms.

7.1.5.3 Fourth Order Surface (X,Y) Plots Constrained to a Fixed Distance (Radius) at the Four Axis Crossings [Z Coefficients Zero]

Figure 33 is basically a cylinder, where a G4 surface is used with the fourth order coefficients small enough to be negligible compared to the second order coefficients and the radius of the cylinder is 80 cm. All “z” coefficients are zero in this and the subsequent plots so only the “x” and “y” coefficients are being changed.

For Figure 34, the two second order coefficients [C_{25} for x^2 and C_{26} for y^2] are decreased from “1.0” to “0.8” while the fourth order coefficients [for x^4 and y^4] are increased just enough to compensate so that the radial distances at the crossings of the x or y axis are retained at 80 cm, i.e.,

$$C_1 80^4 + C_{25} 80^2 - 6400 = 0$$

is used to determine the C_1 coefficient (or C_2 the same way for y) with the constant term held constant at “-6400”. The surface shown in Figure 34 is still nearly a circle, although positioning the mouse at various azimuth locations shows that the distance from the origin increases off axis (this was also shown on a spreadsheet). Further decreasing the C_{25} and C_{26} coefficients to “0.4” gives Figure 35, where now it is beginning to look more like a square than a circle. Finally, for Figure 36, because the second order coefficients C_{25} and C_{26} are decreased all the way to “0.0”, there are only the fourth order coefficients and it now has the appearance of a square with rounded corners (the distances to the origin from the axis crossings of the surface are still 80 cm), where there is symmetry in the different quadrants because only even order terms are used in the G4 surface. As shown by Figure 37, little change in the shape of the surface is obtained by going to negative C_{25} and C_{26} values when they are “-1.0”.

A large change is obtained by going to the larger positive C_{25} and C_{26} values of “2” [negative C_1 and C_2], as shown by Figure 38. Outside zones also now show up on the plot, which has similarities to a second order cone above and below the apex, except with fourth order many outside zones are possible. This would need to be understood to use such a G4 surface. A spreadsheet verified that for some of these points along the surface on an outside zone the surface equation indeed goes to zero.

Another method of making the well rounded square of Figure 35 more like a square is to introduce a small negative coefficient for the x^2y^2 term as shown in Figure 39. This can be made more like a circle by changing this x^2y^2 term to a small positive coefficient, as shown in Figure 40. The “circle” is brought in dynamically on the diagonals to resemble a four-sided star with a larger positive coefficient for the x^2y^2 term as shown in Figure 41.

7.1.5.4 Surface (X,Y) Plots Constrained to Fixed Distances Corresponding to Those for an Ellipse at the Four Axis Crossings [Z Coefficients Zero]

Figure 42 shows an ellipse with major radius 80 cm and minor radius 40 cm, where the surface is the usual second order GQ MCNP surface with center at the origin and the plot is created with the usual 2D plotter. Figure 43 is basically the same except a G4 surface is used with the fourth order coefficients small enough to be negligible compared to the second order coefficients. All “z” coefficients in these plots are zero; therefore, only the “x” and “y” coefficients are changed.

Changes could be made for this ellipse similar to what was done before for perturbing the circle, where the values at the ellipse axes are preserved with an appropriate balance between the second and fourth order coefficients. Here only an example is shown of the ellipse turning into a near rectangle in Figure 44 by using only the fourth order coefficients.

7.1.5.5 Surface (X,Y) Plots Showing a Variable Two Dimensional Slanted Plane Surface

Figure 45 shows a slanted plane surface, where the slant of the plane changes according to the coefficients of the (x, x^2, x^3, x^4, y) terms. It starts out with a small positive linear slope at $x=0$ with the slope changing according to the values of the higher order x coefficients. Because the x^4 coefficient has a small negative value, the surface curves downward for large x . The fourth order G4 surface allows greater detail in the surface variation than could be obtained with second order.

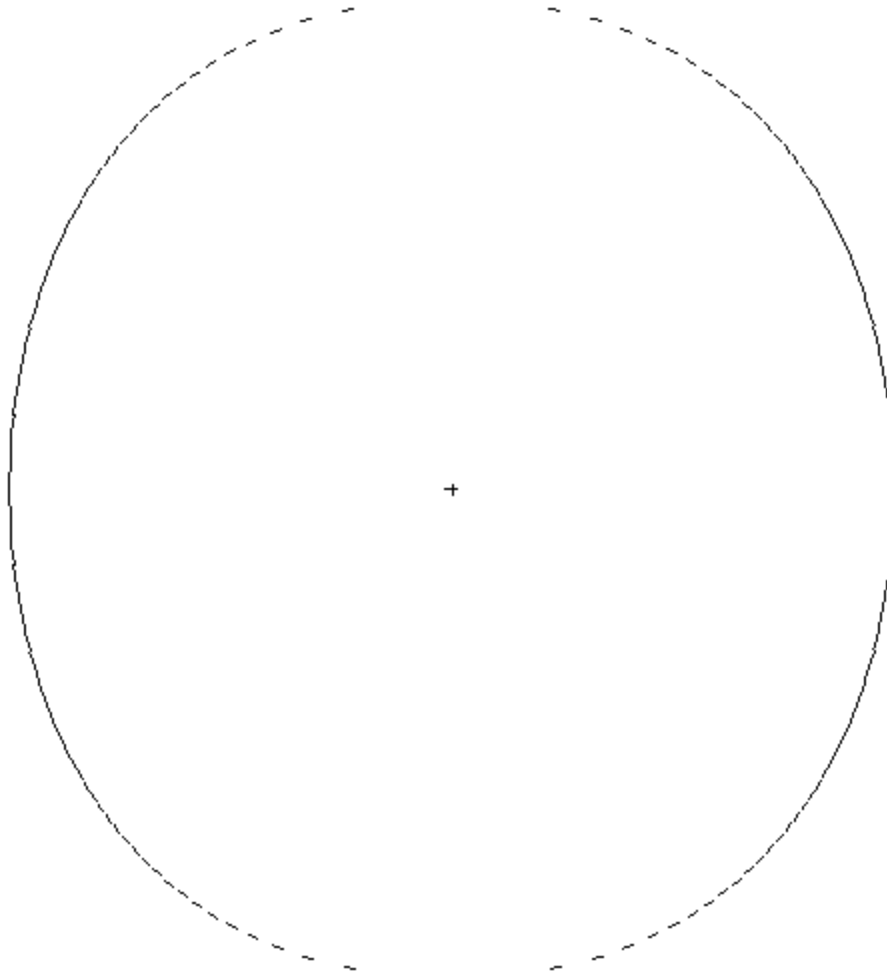


Figure 27. (x,y) View of g4 Surface for Sphere Flattened Some in + and - x, Input File ig4ca. Or=(0,0,0), ext=11

Input file ig4ca:
test with only fourth order surface, ig4ca

```
1  g4  .004  0  22r
      1.  1.  1.  0. 6r  -100.  0.  0.  0.
```

```
mode    n
sdef    erg=1.01
nps     10
print
```

Note from the g4 equation, $.004x^4 + x^2 + y^2 + z^2 - 100 = 0$,
with $y=z=0$, it becomes:
 $.004x^4 + x^2 = 100$ or $x=8.75$, which is the maximum x values on the plot.

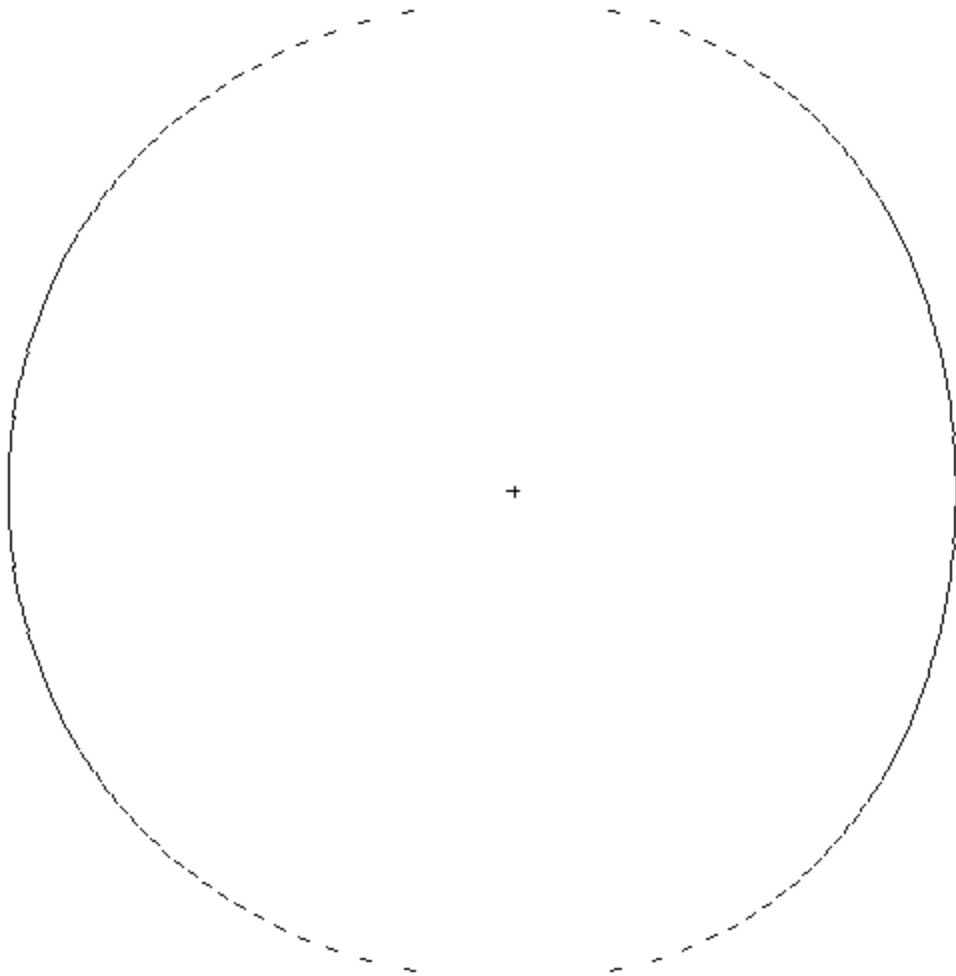


Figure 28. (x,y) View of g4 Surface for Sphere Flattened Some in +x, Input File ig4da. Or=(0,0,0), ext=11

Input file ig4da:

test with only fourth order surface, ig4da

```
1 g4 .001866 0 13r .01866 0. 7r
      1. 1. 1. 0. 6r -100. 0. 0. 0.
```

```
mode n
sdef erg=1.01
nps 10
print
```

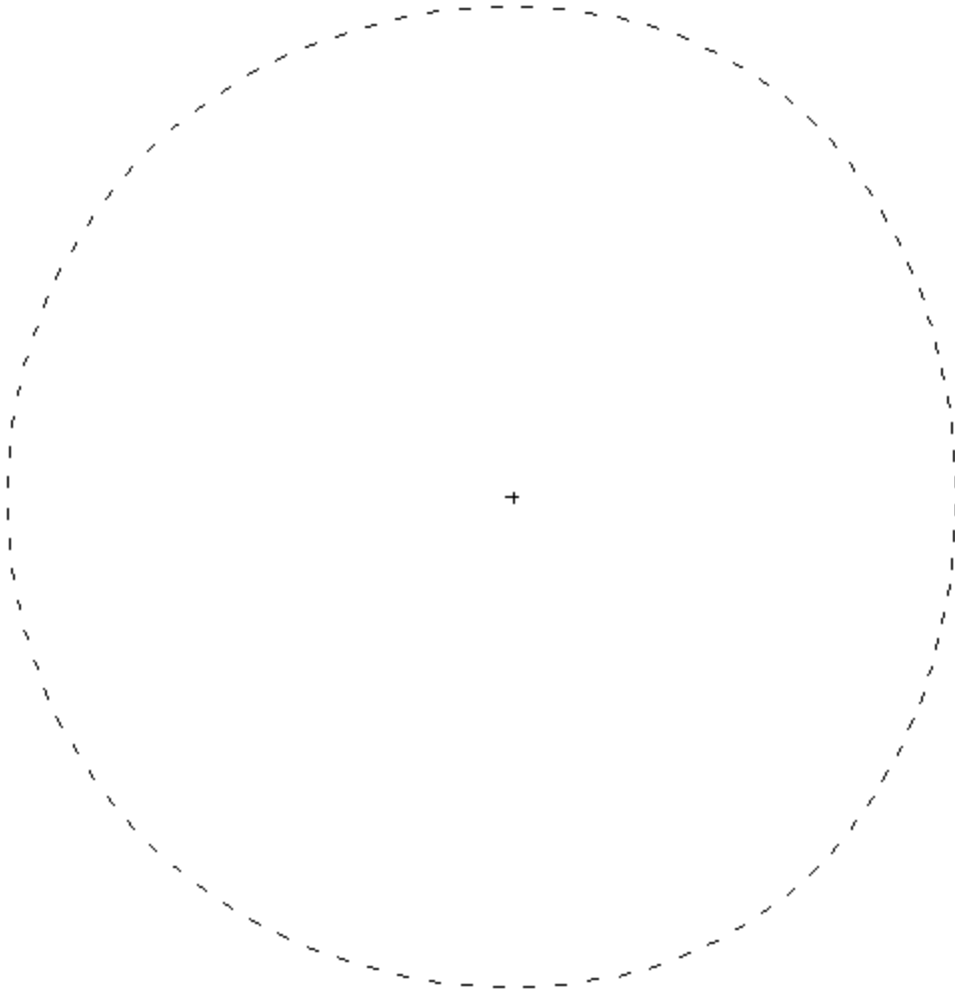


Figure 29. (x,y) View of g4 Surface for Sphere Flattened Some in +x, Input File ig4da Plotted with Circular Source. Or=(0,0,0), ext=11

Input file ig4da:
test with only fourth order surface, ig4da

```
1 g4 .001866 0 13r .01866 0. 7r  
1. 1. 1. 0. 6r -100. 0. 0. 0.
```

```
mode n  
sdef erg=1.01  
nps 10  
print
```

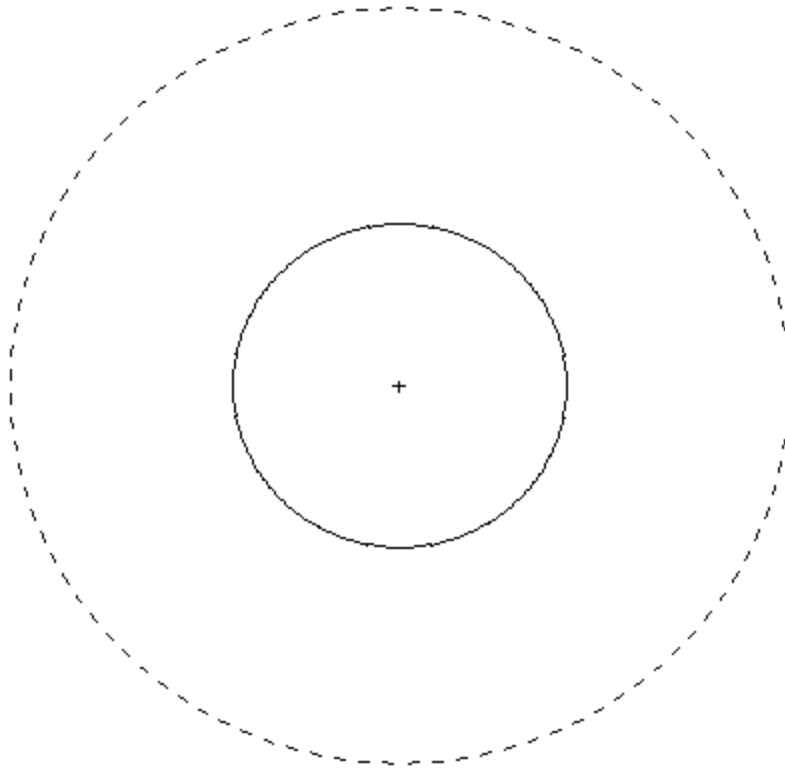


Figure 30. (y,z) View of g4 Surface for x-Torus Input File ig4tx. $O_r=(0,0,0)$, $ext=100$

Input file ig4tx:

test fourth order surface, like tx A=50 B=C=20, ig4tx

```
1  g4  6.25e-6 6.25e-6 6.25e-6 0 5r 1.25e-5 1.25e-5 1.25e-5
    0 11r 0.02625 -0.03625 -0.03625 0 6r 27.5625 0. 0. 0.
```

```
mode    n
sdef    erg=1.01
nps     10
print
```

Note: On the Visual Editor plot the radius of the outer circle is about 70, and the radius of the inner circle is about 30 as expected.

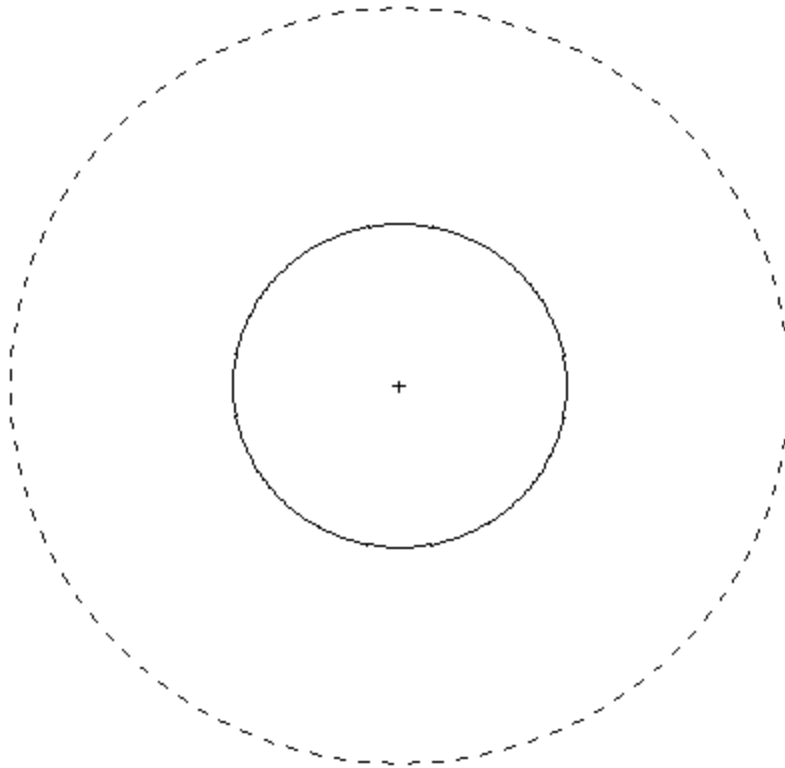


Figure 31. View of g4 Surface for Torus Input File ig4txr with Slanted Axis. $O_r=(0,0,0)$, $ext=100$, $b=(.707,-.707,0,0,0,1)$

Input file ig4txr:

```
test fourth order surface, like tx A=50 B=C=20,rotate 45d, ig4txa

1  g4  6.25e-6 6.25e-6 6.25e-6 0 5r 1.25e-5 1.25e-5 1.25e-5
   0 11r -.005  -0.005  -0.03625 .0625 0 5r 27.5625  0. 0. 0.

mode      n
sdef      erg=1.01
nps       10
print
```

Note: Axis of torus is along the line $x=y$ with $z=0$.

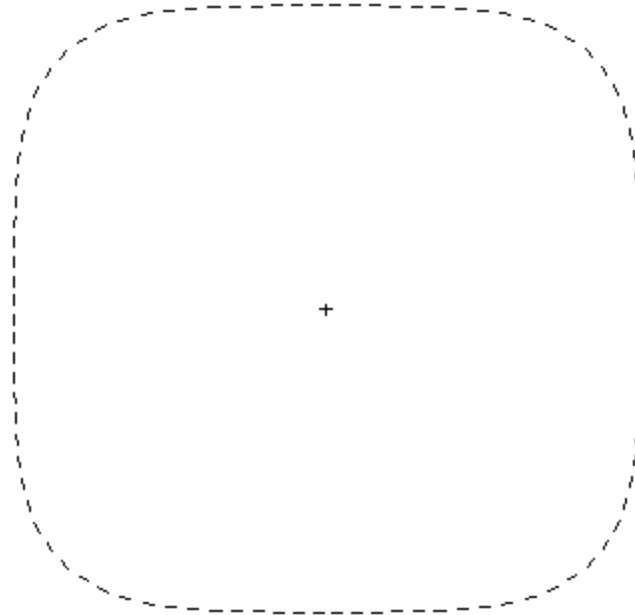


Figure 32. (x,y) View of g4 an Approximate Cube that is Nearly Flat on the Edges with Rounded Corners, Input File ig44a Plotted with Circular Source. Or=(0,0,0), ext=100

Input file ig44a:

```
test with only fourth order surface,only 4th terms, ig44a

  1  g4  1  1  1          0  20r
      0.  0.  0.  0.  6r  -10000000.  0.  0.  0.

mode    n
sdef    erg=1.01
nps     10
print
```

Note: Only fourth $[x^4,y^4,z^4]$ terms are used along with the constant. The transition to more rounded sides could be obtained by including second $[x^2,y^2,z^2]$ order terms.

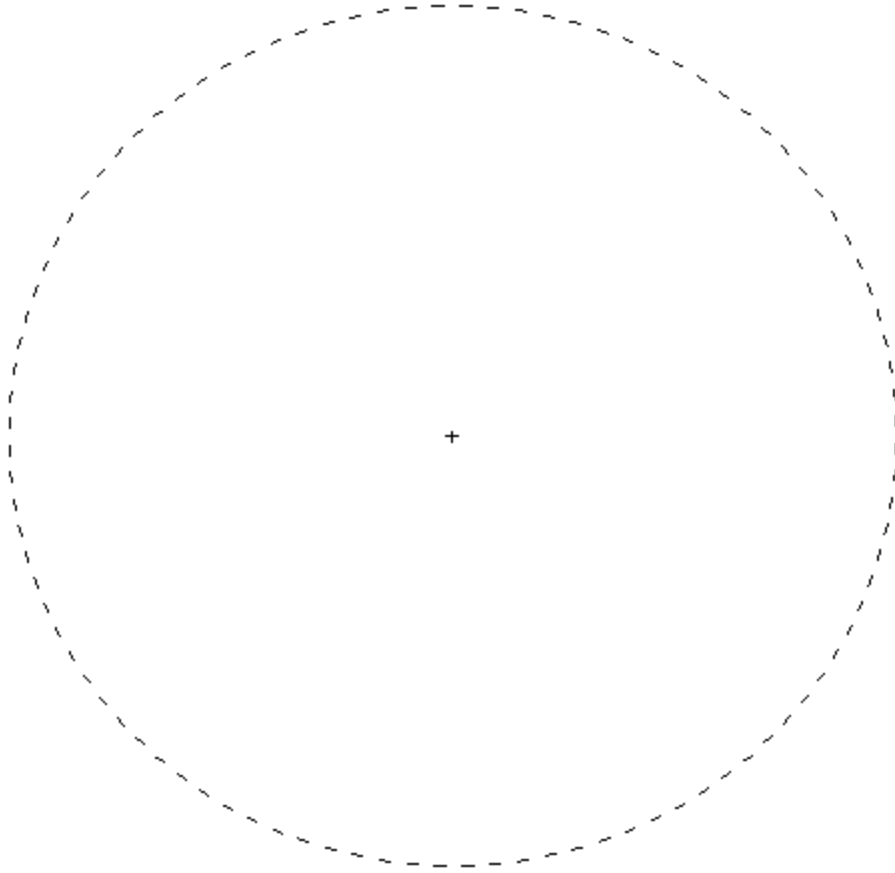


Figure 33. (x,y) View of G4 Cylinder with Radius of 80 cm and Negligible 4th Order Terms.

Input file ig4.2

cyl radius of 80 cm and negligible 4th order terms, ig4.2

```
1  g4  .000001  .000001  0  0  20r
      1  1  0  0  6r  -6400  0  2r
```

```
mode  n
print
```

Note: Plot is made inside of circle of radius 100 cm: extents=100.

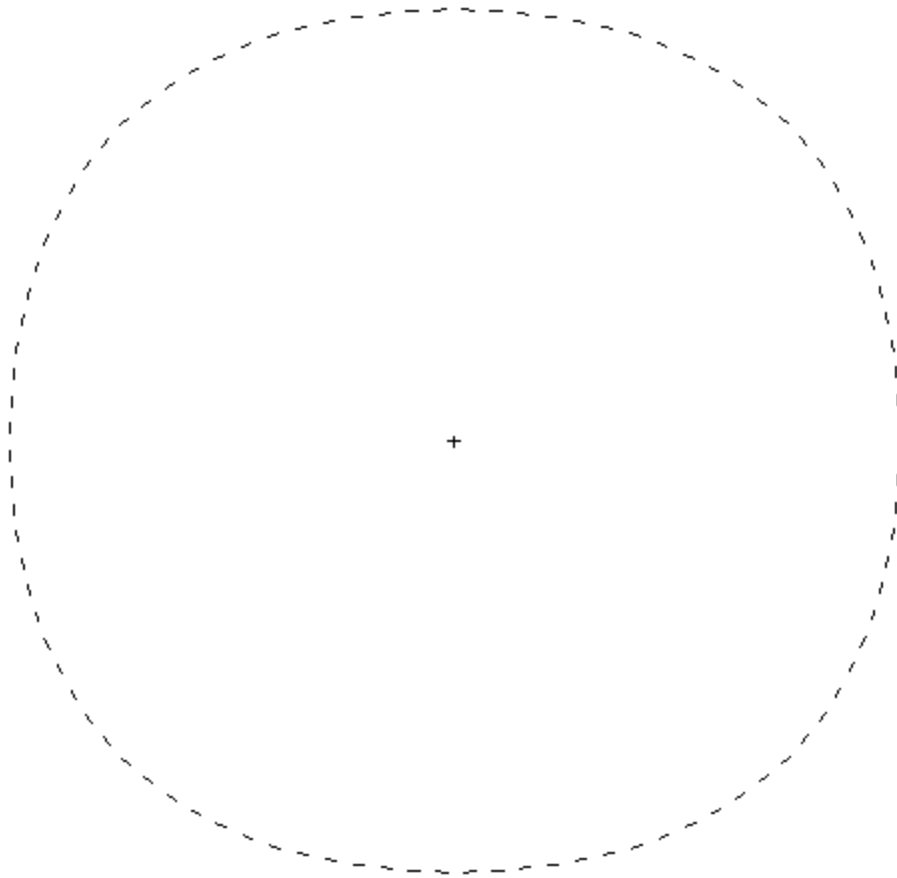


Figure 34. (x,y) View of G4 with Axis Crossings of 80 cm and Small Positive 4th Order Terms.

Input file ig4.3

cyl radius of 80 cm at axis crossings, 4th order terms, ig4.3

```
1  g4  .00003125 .00003125  0      0 20r
      .8   .8      0      0 6r  -6400    0 2r
```

```
mode   n
print
```

Note: Distance from origin at 45 degrees has increased from the 80 cm to about 84.2 cm. Plot is made inside of circle of radius 100 cm; extents=100.

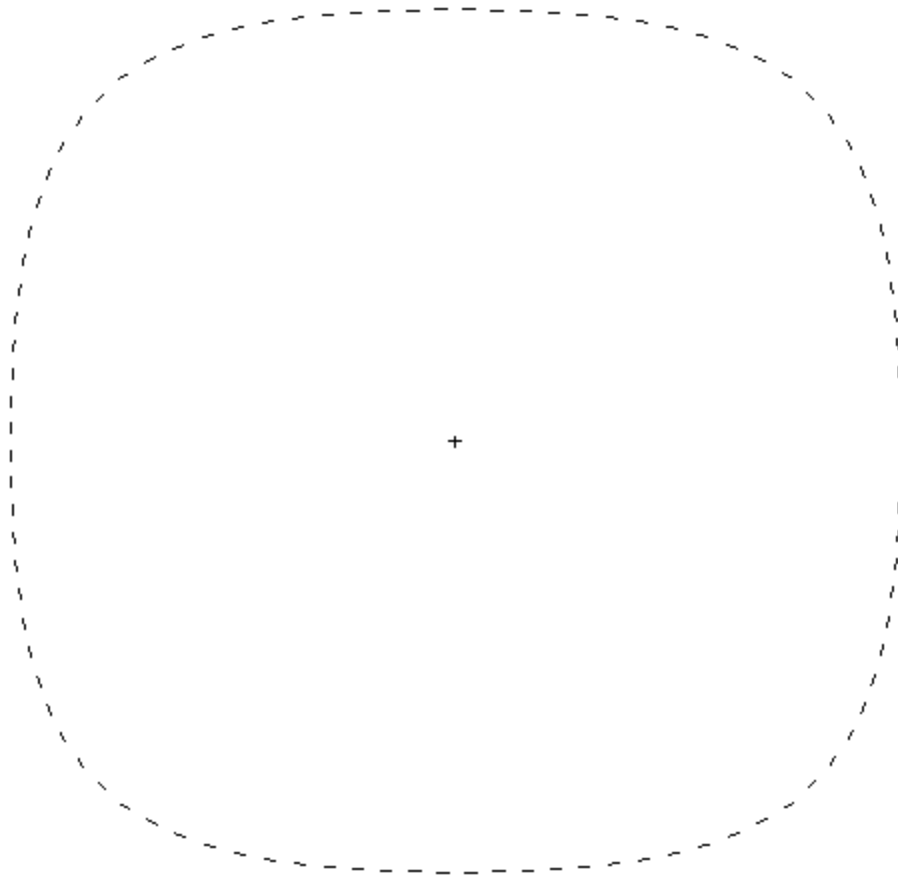


Figure 35. (x,y) View of G4 with Axis Crossings of 80 cm and Moderate Positive 4th Order Terms.

Input file ig4.4:

cyl radius of 80 cm at axis crossings, 4th order terms, ig4.4

```
1  g4  .00009375 .00009375  0      0 20r
      .4   .4      0      0 6r  -6400    0 2r
```

```
mode    n
print
```

Note: Plot is made inside of circle of radius 100 cm; extents=100.

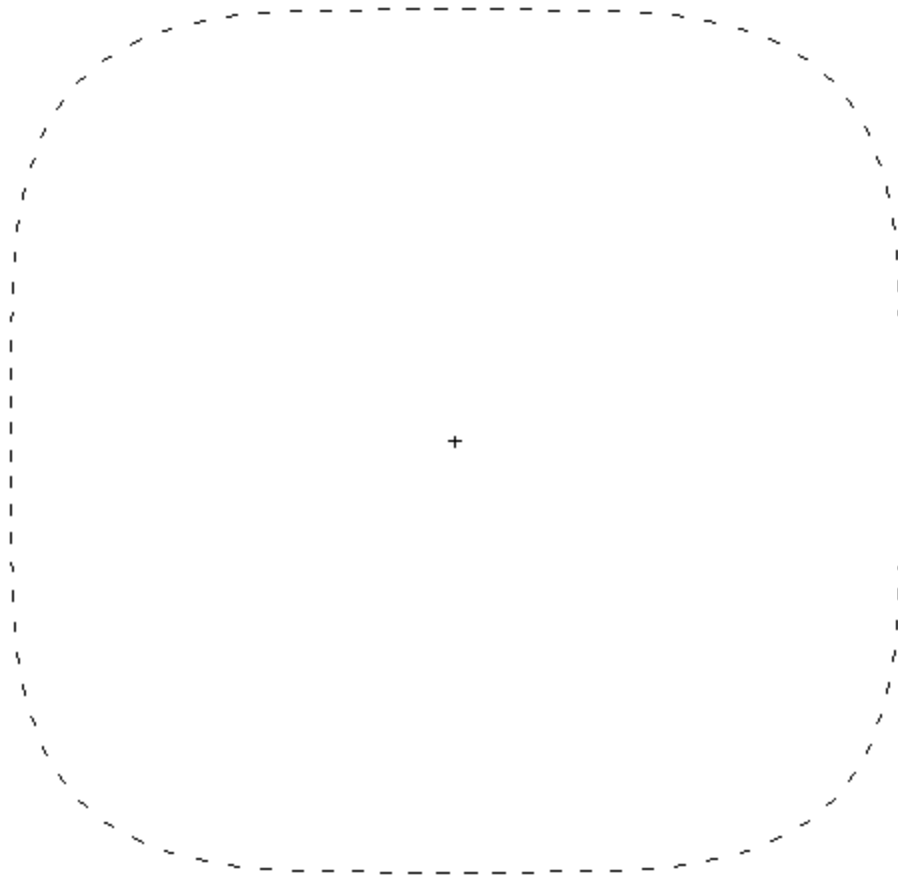


Figure 36. (x,y) View of G4 with Axis Crossings of 80 cm and Only Positive 4th Order Terms [second order terms are zero].

Input file ig4.5

cyl radius of 80 cm at axis crossings, 4th order terms, ig4.5

```
1  g4  .00015625 .00015625  0      0 20r
      .0      .0      0      0 6r -6400  0 2r
```

```
mode  n
print
```

Note: Coefficients of second order terms are zero. Plot is made inside of circle of radius 100 cm; extents=100.

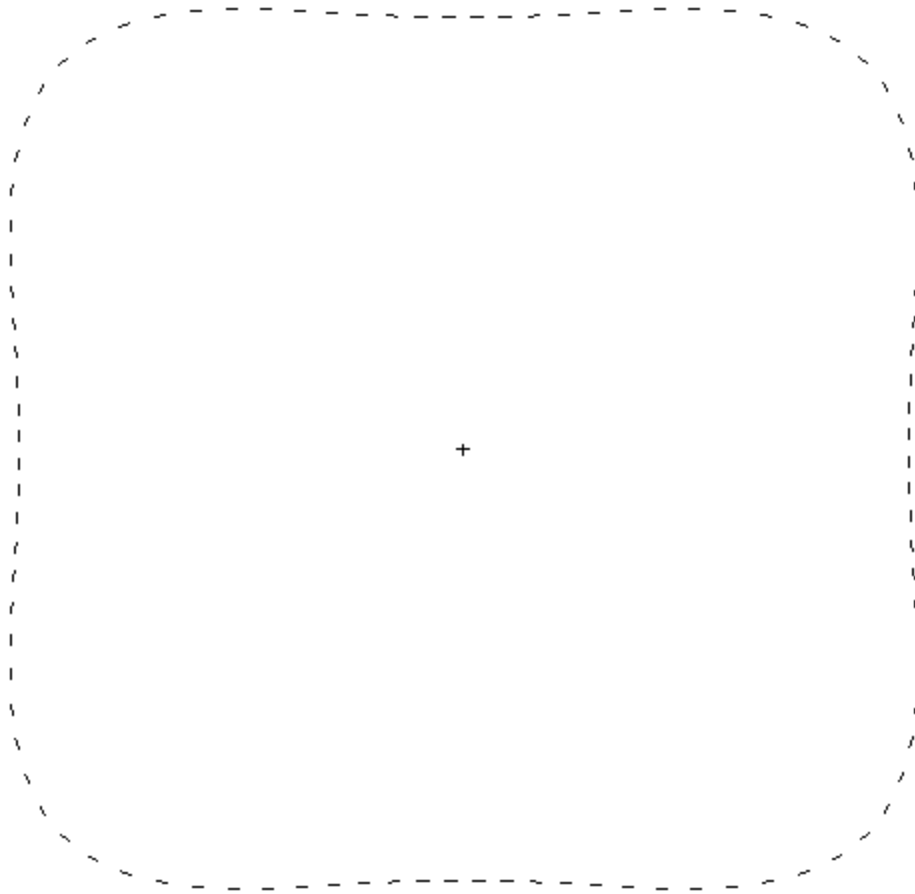


Figure 37. (x,y) View of G4 with Axis Crossings of 80 cm and Negative 2nd Order Terms [4th order terms increase more].

Input file ig4.6

cyl radius of 80 cm at axis crossings, 4th order terms, ig4.6

```
1  g4  .00031250 .00031250  0      0 20r
      -1.    -1.      0      0 6r -6400    0 2r
```

```
mode   n
print
```

Note: Plot is made inside of circle of radius 100 cm.

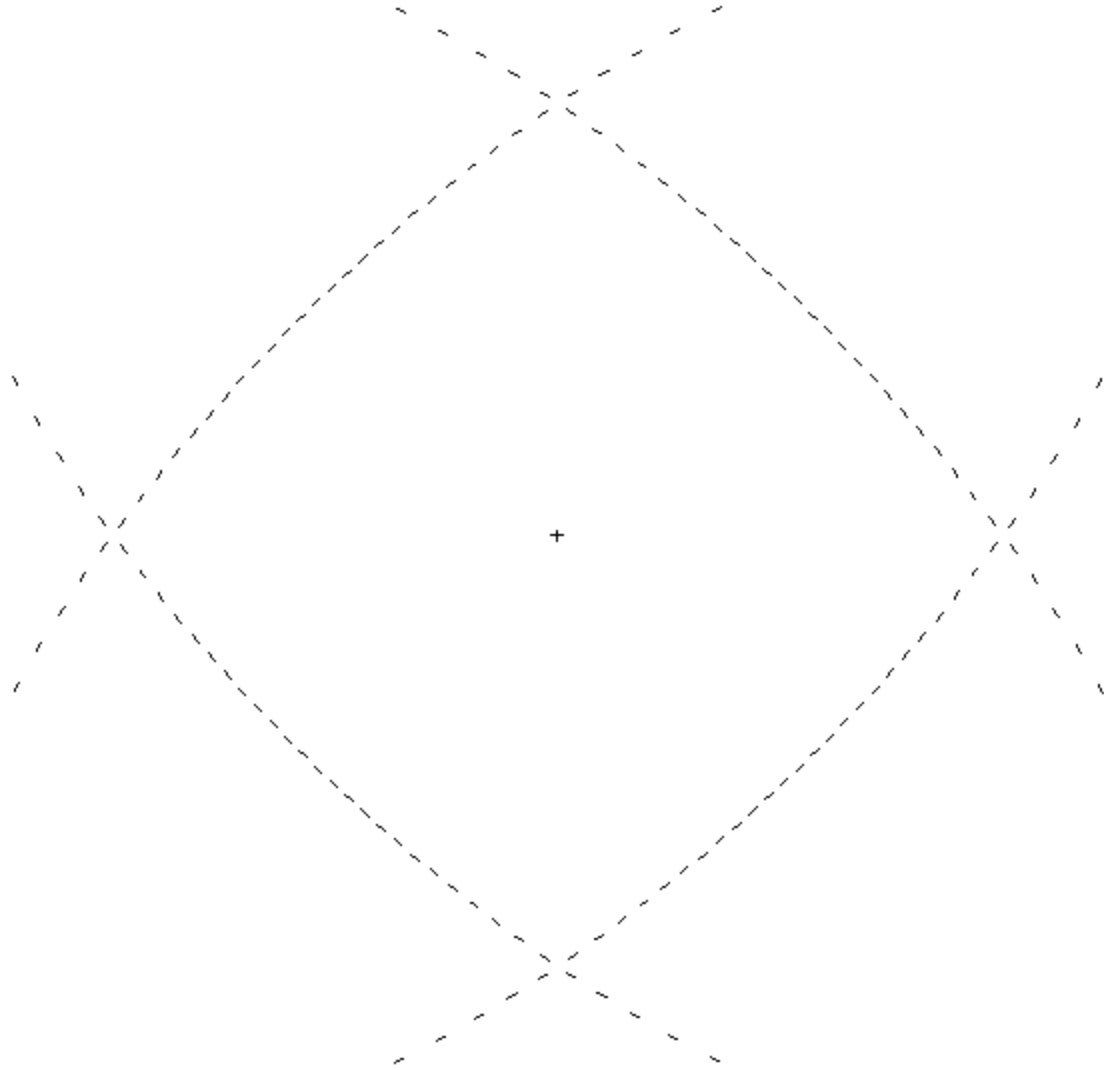


Figure 38. (x,y) View of G4 with Axis Crossings of 80 cm and Increasing 2nd Order Terms [4th order terms are negative].

Input file ig4.7

cyl radius of 80 cm at axis crossings, 4th order terms, ig4.7

```
1  g4  -.00015625  -.00015625  0  0  20r
      2  2  0  0  6r  -6400  0  2r
```

```
mode  n
print
```

Note: Plot is made inside of circle of radius 100 cm; extents=100.

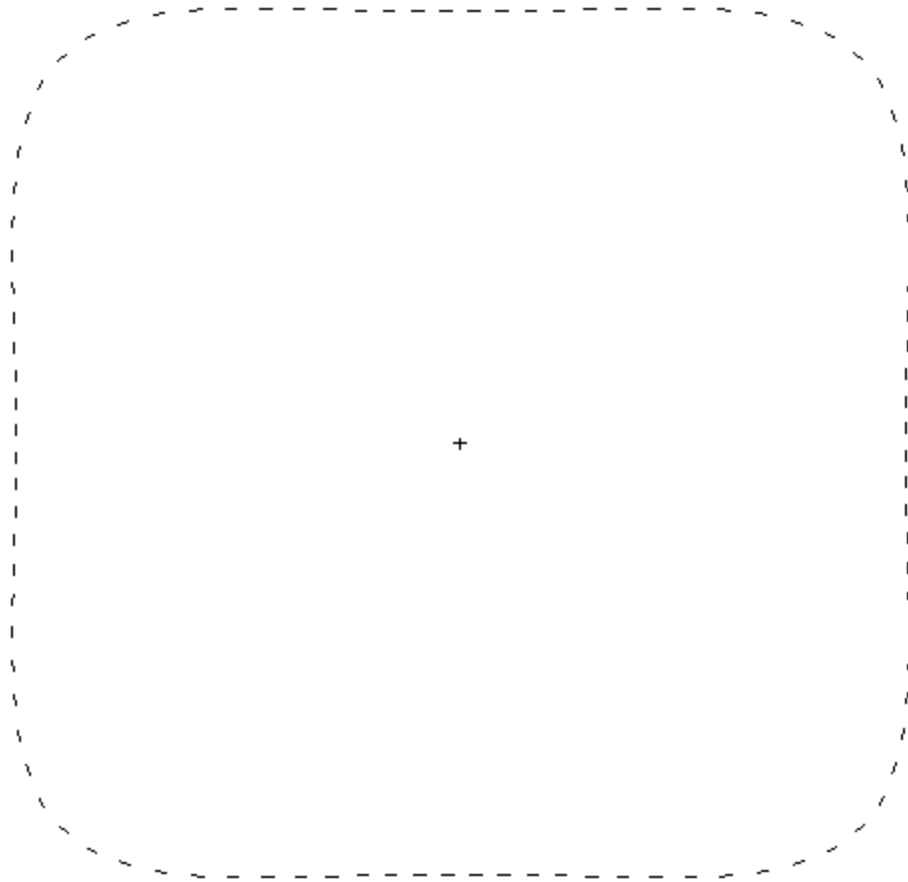


Figure 39. (x,y) View of G4 with Axis Crossings of 80 cm and Moderate Positive 4th Order Terms Along with Small Negative x^2y^2 Term.

Input file ig4.8

cyl radius of 80 cm at axis crossings, 4th order terms, ig4.8

```
1  g4  .00009375 .00009375  0  0 5r -.0001 0  0 12r
      .4   .4     0      0 6r -6400  0 2r
```

```
mode  n
print
```

Note: Plot is made inside of circle of radius 100 cm; extents=100.

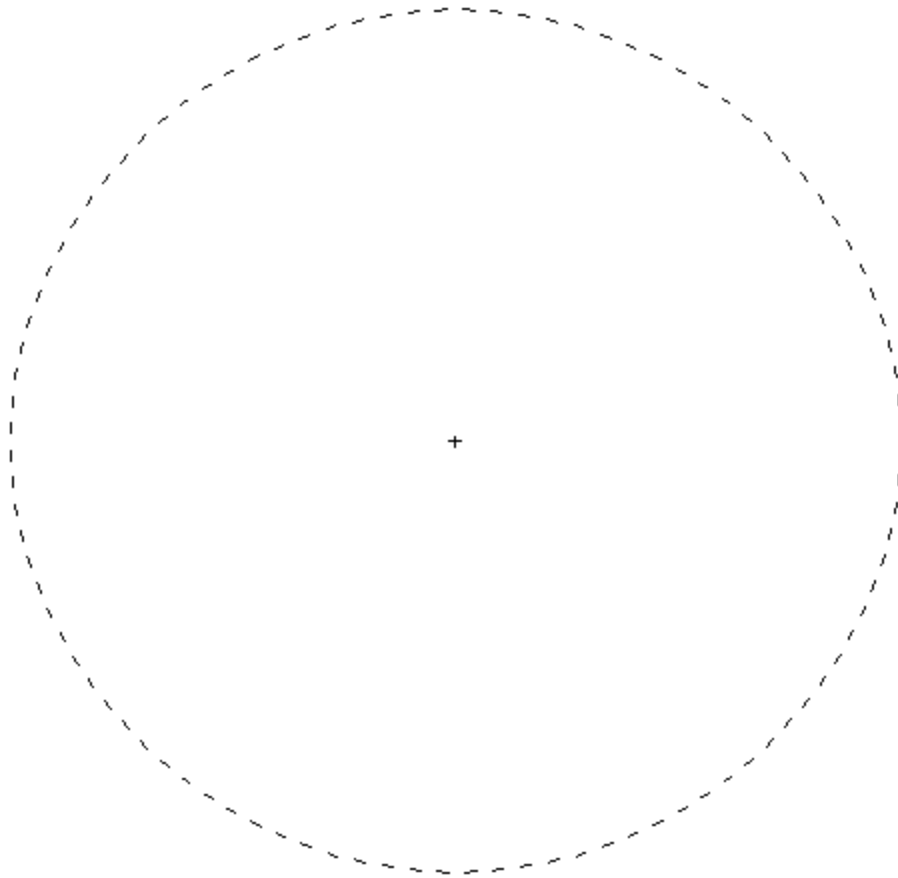


Figure 40. (x,y) View of G4 with Axis Crossings of 80 cm and Moderate Positive 4th Order Terms Along with Small Positive x^2y^2 Term.

Input file ig4.9

cyl radius of 80 cm at axis crossings, 4th order terms, ig4.9

```
1  g4  .00009375 .00009375  0  0 5r  .0002 0  0 12r
      .4    .4    0    0 6r  -6400    0 2r
```

```
mode  n
print
```

Note: Plot is made inside of circle of radius 100 cm; extents=100.

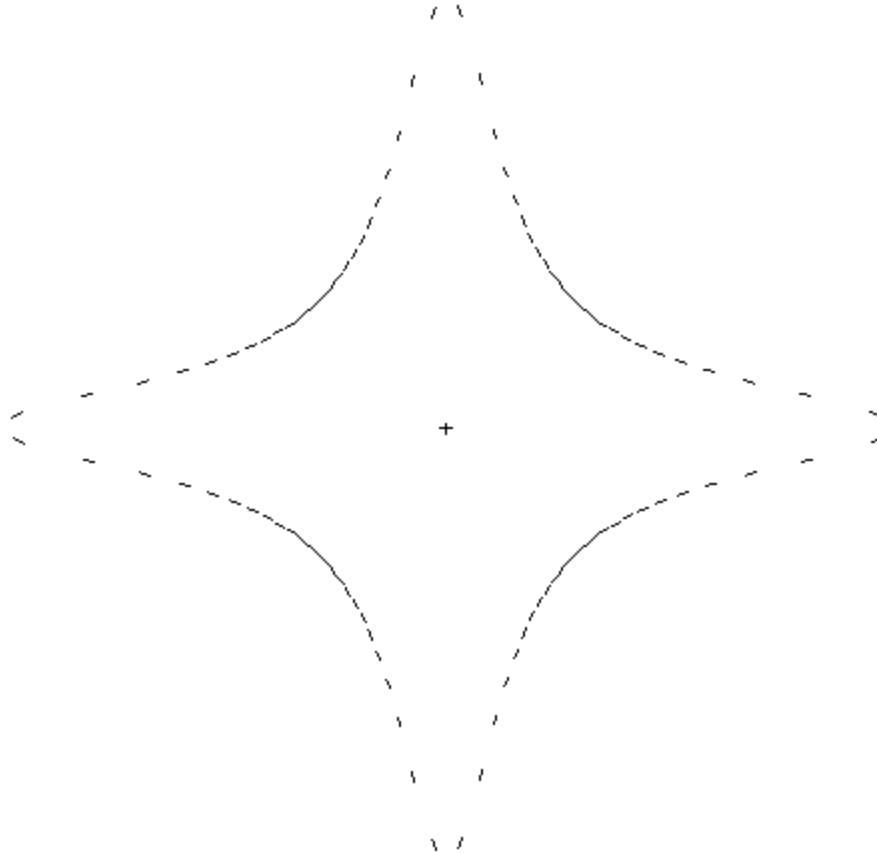


Figure 41. (x,y) View of G4 with Axis Crossings of 80 cm and Moderate Positive 4th Order Terms Along with Larger Positive x^2y^2 Term.

Input file ig4.10

cyl radius of 80 cm at axis crossings, 4th order terms, ig4.10

```
1  g4  .00009375 .00009375  0  0 5r  .0200 0  0 12r
      .4   .4     0     0 6r  -6400  0 2r
```

```
mode  n
print
```

Note: Plot is made inside of circle of radius 100 cm; extents=100.

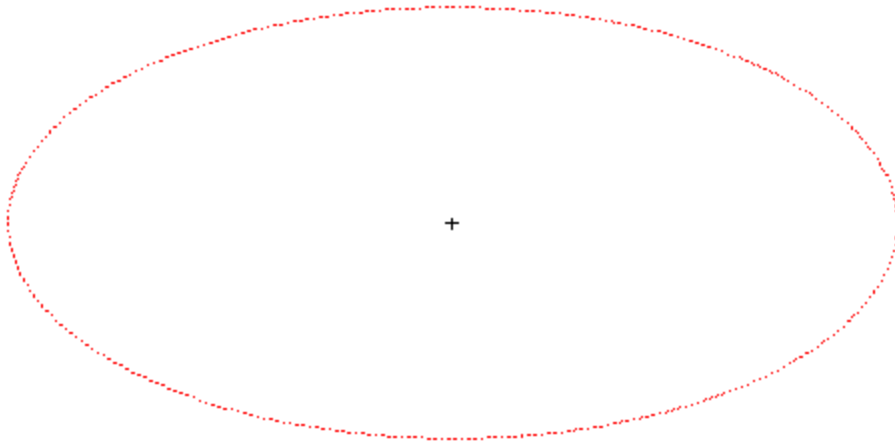


Figure 42. (x,y) View of GQ Elliptical Cylinder with Major x-Radius of 80 cm and Minor y-Radius of 40 cm.

Input file igq.11

ellipse of major x-radius 80 and minor y-radius 40, igq.11

```
1      gq 0.00015625 0.000625 0 0 0 0 0 0 0 -1
```

Note: Extents=100 cm with normal 2D plotter.

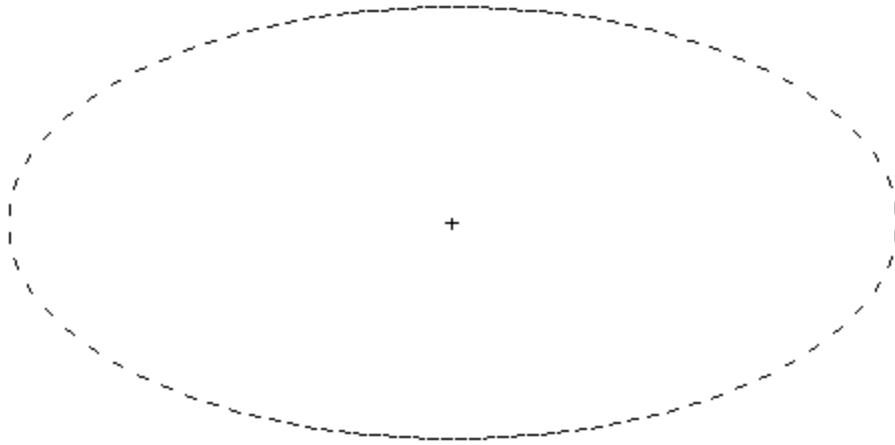


Figure 43. (x,y) View of G4 Elliptical Cylinder with Major x-Radius of 80 cm and Minor y-Radius of 40 cm Using Negligible 4th Order Terms.

Input file ig4.12

ellipse with negligible 4th order terms, ig4.12

```
1  g4  .0000000001  .0000000001  0  0  20r
      .00015625  .000625  0  0  6r  -1  0  2r
```

```
mode  n
print
```

Note: Plot is made inside of circle of radius 100 cm: extents=100.

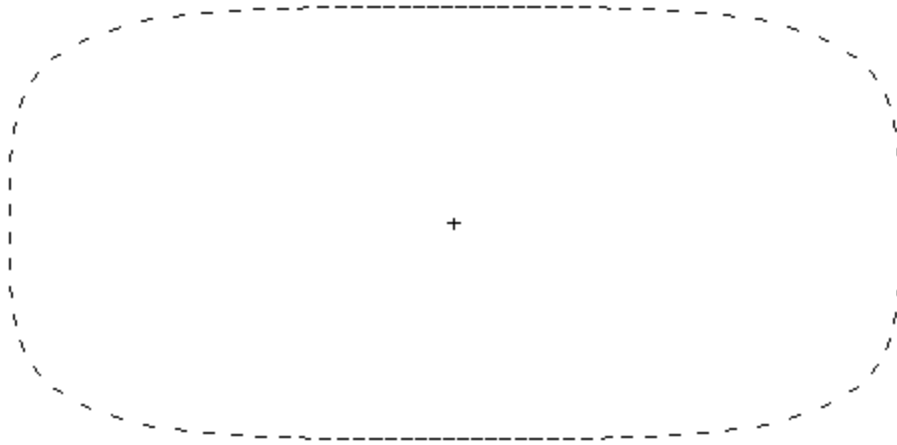


Figure 44. (x,y) View of G4 Elliptical Cylinder [with Major x-Radius of 80 cm and Minor y-Radius of 40 cm] Turned into a Near Rectangle by Using Only 4th Order Terms.

Input file ig4.13

ellipse turned into near rectangle with 4th order terms, ig4.13

```
1  g4  2.4414e-8  3.9063e-7  0  0  20r
      0  0  0  0  6r  -1  0  2r
```

```
mode  n
print
```

Note: Plot is made inside of circle of radius 100 cm: extents=100.

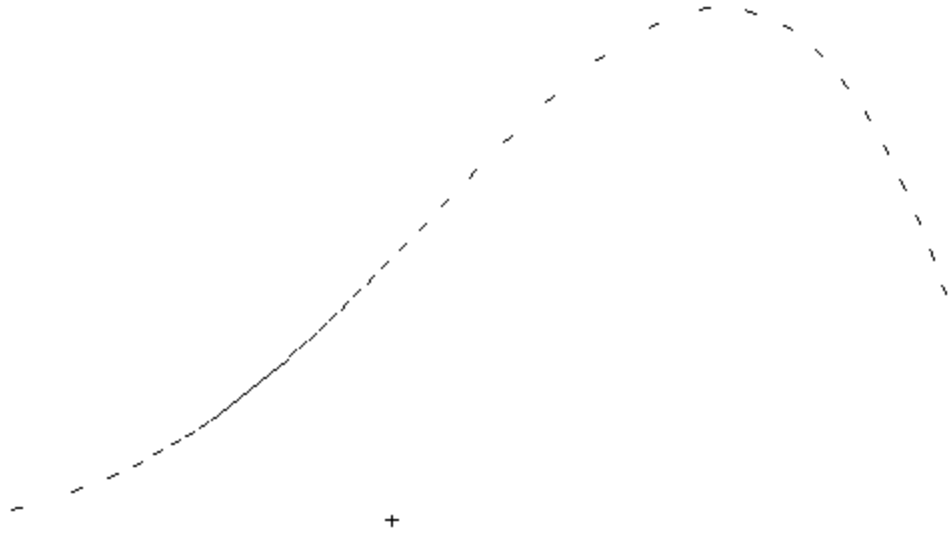


Figure 45. (x,y) View of G4 Slanted Plane Surface Using Terms (x, x^2, x^3, x^4, y) .

Input file ig4.14

variable two dimensional slanted plane surface, ig4.14

```
1 g4  -.000004  0 13r  .0003  0 7r .0065  0 4r .125 -1 0 5r
```

```
mode    n  
print
```

Note: Plot is made inside of circle of radius 50 cm: origin=(40,0,0) extents=50.

Values along surface:	x	y
	0	0
	20	6.9
	40	24.3
	60	43.9
	80	41.4

7.1.5.6 Appendix: Details of the Preliminary Fourth Order Surface Implementation

The fourth order surface, G4, which has been implemented for testing, is much like the general quadratic GQ surface currently in MCNP for a general second order surface, except the G4 surface includes all linear combinations of terms up to and including fourth order. This requires the following 35 terms for the G4 surface equation:

$$\begin{aligned}
 & C_1 (x-\underline{x})^4 & + & C_2 (y-\underline{y})^4 & + & C_3 (z-\underline{z})^4 & + & \\
 & C_4 (x-\underline{x})^3 (y-\underline{y}) & + & C_5 (y-\underline{y})^3 (z-\underline{z}) & + & C_6 (z-\underline{z})^3 (x-\underline{x}) & + & \\
 & C_7 (x-\underline{x})^3 (z-\underline{z}) & + & C_8 (y-\underline{y})^3 (x-\underline{x}) & + & C_9 (z-\underline{z})^3 (y-\underline{y}) & + & \\
 & C_{10} (x-\underline{x})^2 (y-\underline{y})^2 & + & C_{11} (y-\underline{y})^2 (z-\underline{z})^2 & + & C_{12} (z-\underline{z})^2 (x-\underline{x})^2 & + & \\
 & C_{13} (x-\underline{x})^2 (y-\underline{y}) (z-\underline{z}) & + & C_{14} (y-\underline{y})^2 (z-\underline{z}) (x-\underline{x}) & + & C_{15} (z-\underline{z})^2 (x-\underline{x}) (y-\underline{y}) & + & \\
 & C_{16} (x-\underline{x})^3 & + & C_{17} (y-\underline{y})^3 & + & C_{18} (z-\underline{z})^3 & + & \\
 & C_{19} (x-\underline{x})^2 (y-\underline{y}) & + & C_{20} (y-\underline{y})^2 (z-\underline{z}) & + & C_{21} (z-\underline{z})^2 (x-\underline{x}) & + & \\
 & C_{22} (x-\underline{x})^2 (z-\underline{z}) & + & C_{23} (y-\underline{y})^2 (x-\underline{x}) & + & C_{24} (z-\underline{z})^2 (y-\underline{y}) & + & \\
 & C_{25} (x-\underline{x})^2 & + & C_{26} (y-\underline{y})^2 & + & C_{27} (z-\underline{z})^2 & + & \\
 & C_{28} (x-\underline{x}) (y-\underline{y}) & + & C_{29} (y-\underline{y}) (z-\underline{z}) & + & C_{30} (z-\underline{z}) (x-\underline{x}) & + & \\
 & C_{31} (x-\underline{x}) & + & C_{32} (y-\underline{y}) & + & C_{33} (z-\underline{z}) & + & \\
 & C_{34} (x-\underline{x}) (y-\underline{y}) (z-\underline{z}) & + & C_{35} & & & = & 0
 \end{aligned}$$

Here, the C_1 to C_{35} coefficients are input by the user for the desired G4 surface plus the \underline{x} , \underline{y} , \underline{z} to specify an origin that would typically be three zeros. Hence, the user inputs 38 coefficients for a G4 surface, in contrast to the 10 coefficients required for the second order SQ or GQ MCNP surfaces.

The implementation of a 2D plotting capability in the Visual Editor for fourth order surfaces is quite complicated and was beyond this preliminary work scope. However, a preliminary ray tracing surface plotting capability was implemented when there are no cells and there is one “g4” surface (using the “SURFACE SHOW UNUSED” feature of the Visual Editor to show a plot of the surface). Some examples of plots of 4th order surfaces have been shown in the main sections of this report.

7.1.6 Task 6: Implement Data Visualization

Tally plotting has been significantly updated to display mesh tallies. Tally mesh plotting has been partially implemented and the run-time tally has been updated. Additional work is required to make sure the tally plotting features work.

Figure 46 shows the revised tally-plotting panel for a 2D plot. The different tally-plotting modes are now contained in tabs to show only the information for the tally being plotted.

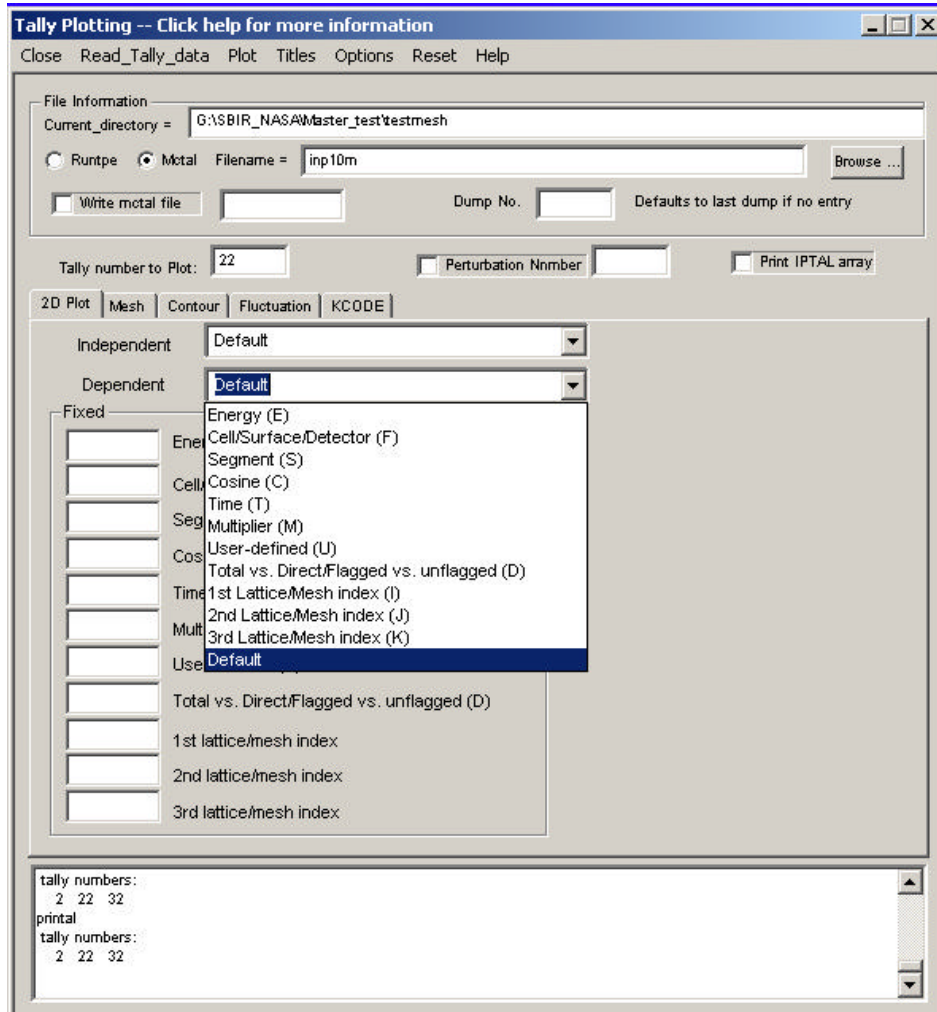


Figure 46. Updated Tally Plotting Panel.

Figure 47 shows the run-time plotting capability. The input file is read in and initialized in the run-time tally panel by selecting the input file and selecting “initiate”. Once the file has been read in and has run for an initial number of particles (default 5,000), the user can pull up the tally plotting panel to set up the tally to plot. Once the tally has been set up, the user can go back to the run-time tally panel and either run for one additional cycle (default 5,000 particles) or run continuously (plot gets updated every 5,000 particles). The user can stop the running code by selecting “stop” from the run-time tally panel. Once the code has been stopped, the tally being plotted can be modified.

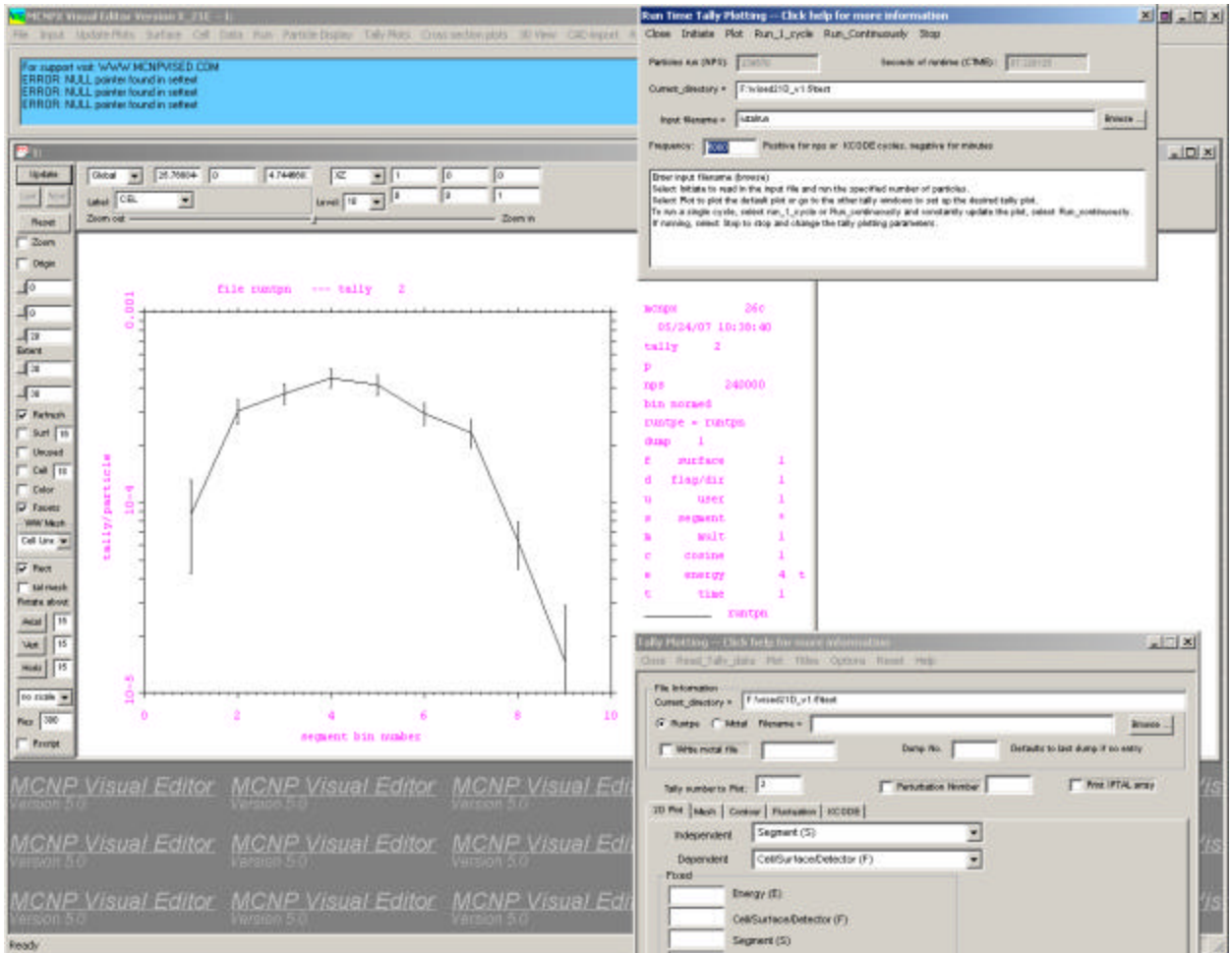


Figure 47. Run-Time Tally Plotting.

Figure 48 shows a mesh tally plot of a critical configuration of seven barrels of fissionable material. This input file is the same one described in the MCNPX release notes for version 2.5.f (LA-UR-05-0891). To generate this plot the code was run and a mctal file generated. The tally plotting window was then brought up by selecting Tally Plots->2D Tally Plots from the main menu.

At the command prompt this plot is generated using the commands “rmctal <mctal fname>” followed by “tal 12 free ik”. In the Visual Editor, the user must first read in the mctal file by entering the name of the mctal file, selecting the “mctal” option, and then selecting “Read_tally_data” from the main menu. Next the user must set the tally number to 12 and then select the mesh tab to bring up the mesh tally plotting options. In the mesh tally window, the user must select IK from the pull down menu to specify the view to plot. Finally, the user must select the “Plot” menu option to plot the mesh tally. The user can then set the window to rectangular by selecting the “Rect” option in the plot window.

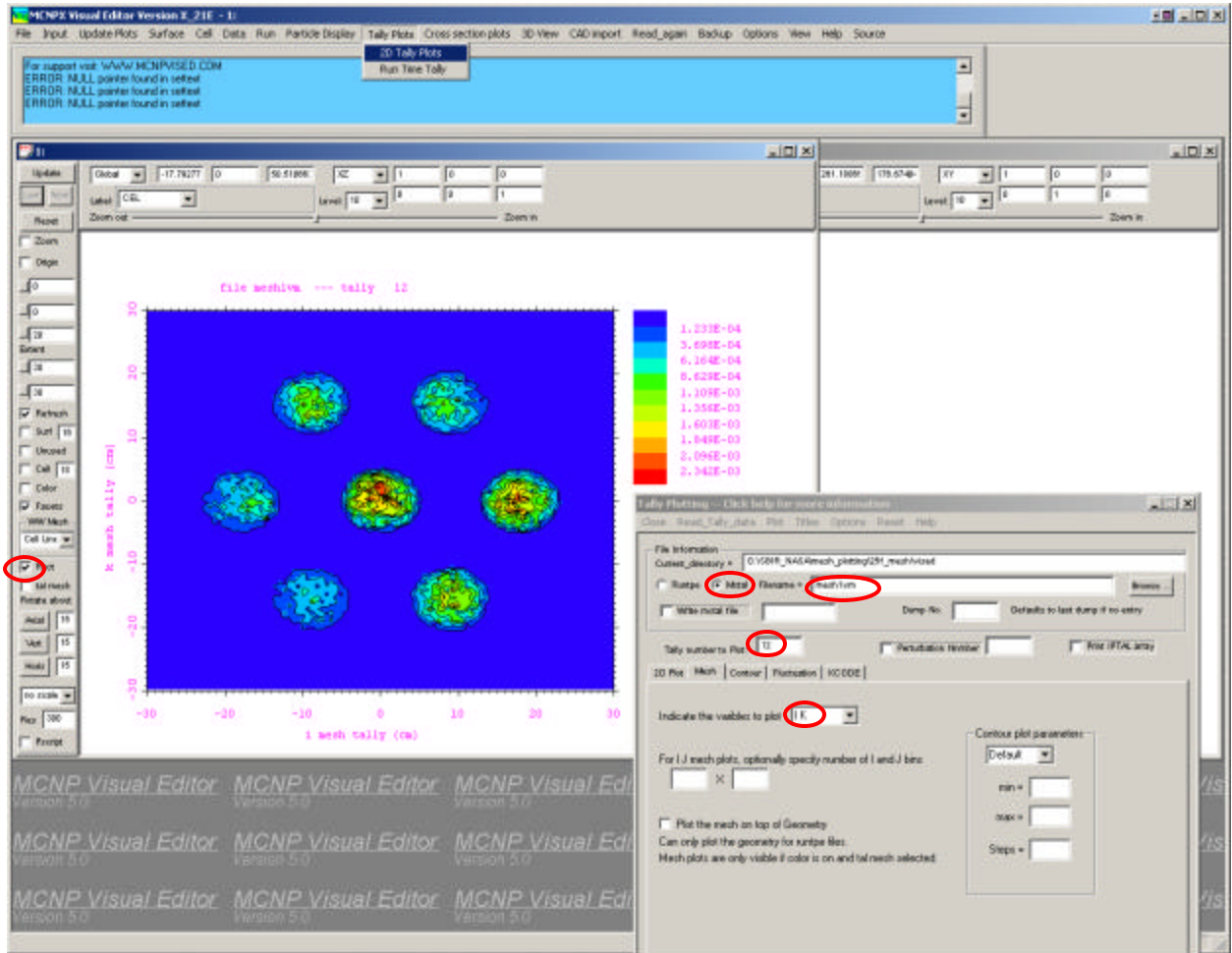


Figure 48. Mesh Tally Plot in the MCNPX Visual Editor.

Figure 49 shows a plot of the same mesh tally plotted on top of the geometry. This plot is also taken from the plot shown in MCNPX release notes for version 2.5.f (LA-UR-05-0891). To generate this plot from the command prompt the user first reads in the runtpe using “Runtpe = <runtpe filename>” followed by “PLOT” to get into the geometry plot mode. Then the user enters the command “py 4 ex 40 or 0 4 0 la 0 1 tal 12 color on la 0 0 con 0 100 %”.

In the Visual Editor, the user must read in the runtpe by setting the runtpe file name and then selecting “Read_tally_data”. Next, the user must set the tally number to “12” and then select the “Plot the mesh on top of the Geometry” check box. This will then set the “talmesh” toggle button in the plot window and will also set the “color” to “on” in the plot window, because tally mesh plots will not be visible if color is not on. Next it is necessary to set the plot view so that the mesh can be seen. For this example, the Y origin is set to “4” and the extents are set to “40”.

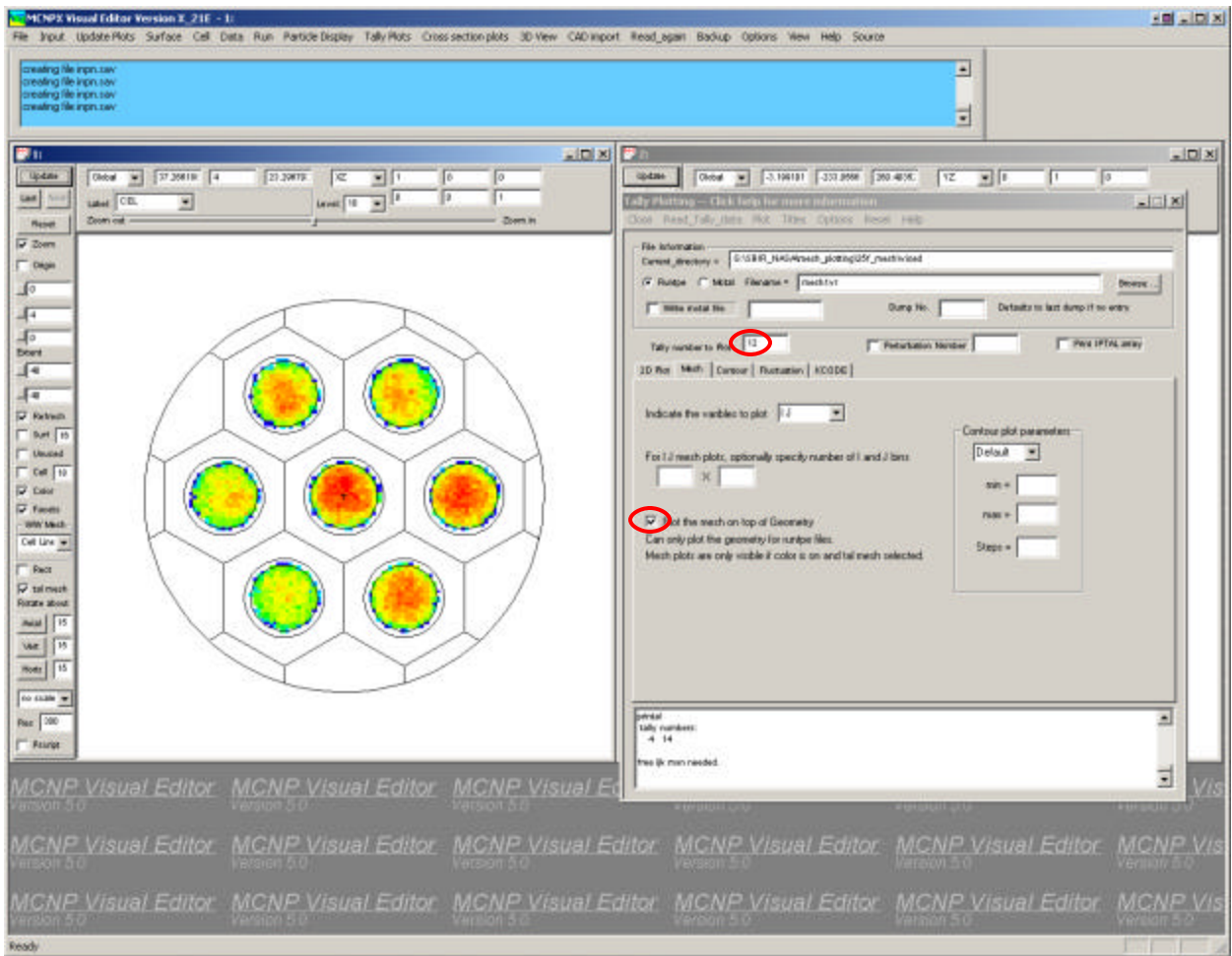


Figure 49. Mesh Tally Plot on Top of the Geometry.

Once the plot has been generated, the view can be changed to show different regions of the tally mesh. In Figure 50 the zoom toggle button was selected and the mouse was dragged across one of the barrels to zoom in on a single barrel.

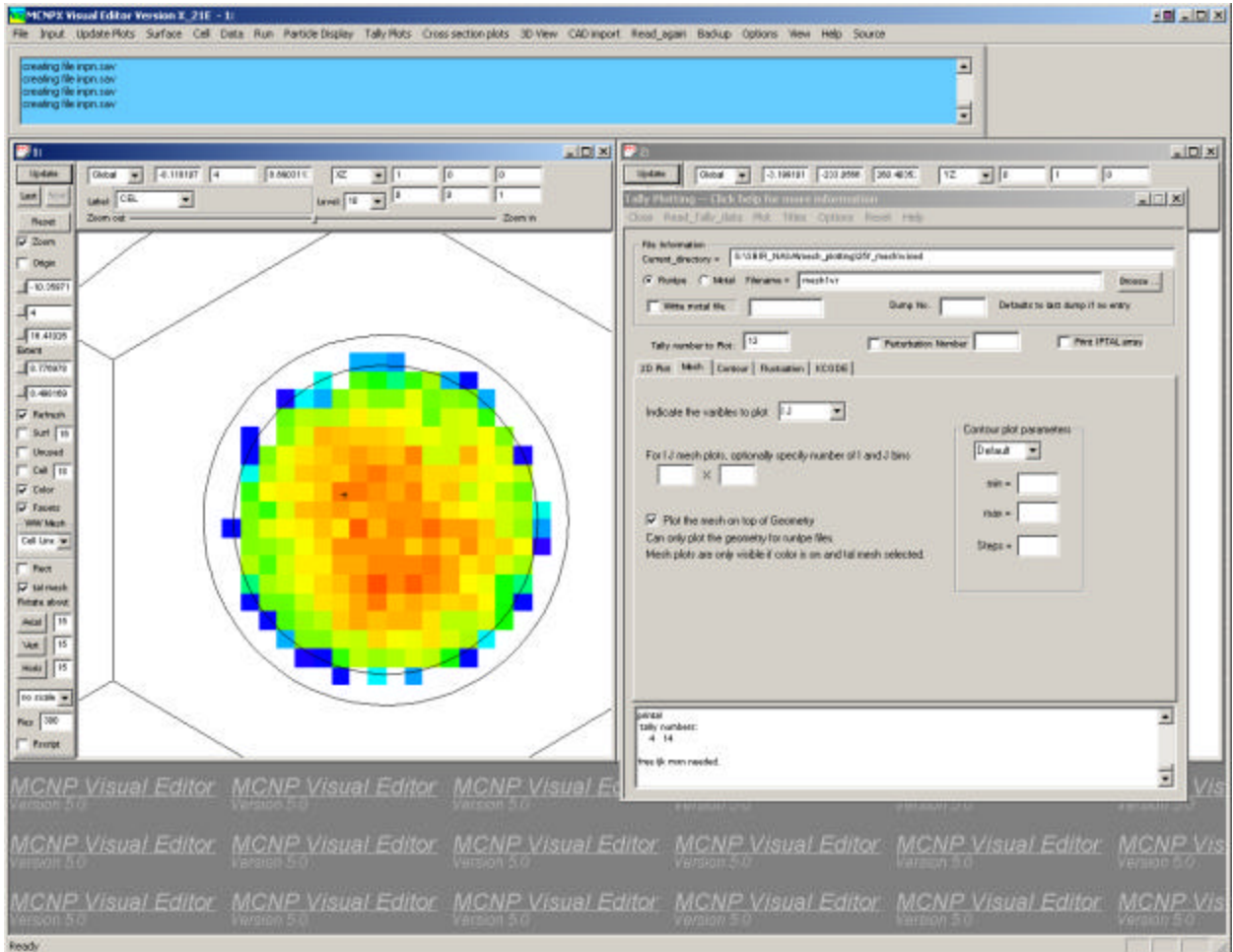


Figure 50. Zoomed in Region of the Mesh Tally Plot.

The latest MCNPX beta version has a number of mesh test problems contained in the testmesh directory that are downloadable from the MCNPX beta site. An attempt was made to reproduce the test plots in both the MCNPX plotter and in the Visual Editor. The following figures show plots made both in the MCNPX tally plotter and the Visual Editor.

Figure 51 shows a contour plot generated with the MCNPX tally plotter using the command “rnc inp01m tal 5 free sc contour line”. Figure 52 shows the same plot generated in the MCPNX Visual Editor.

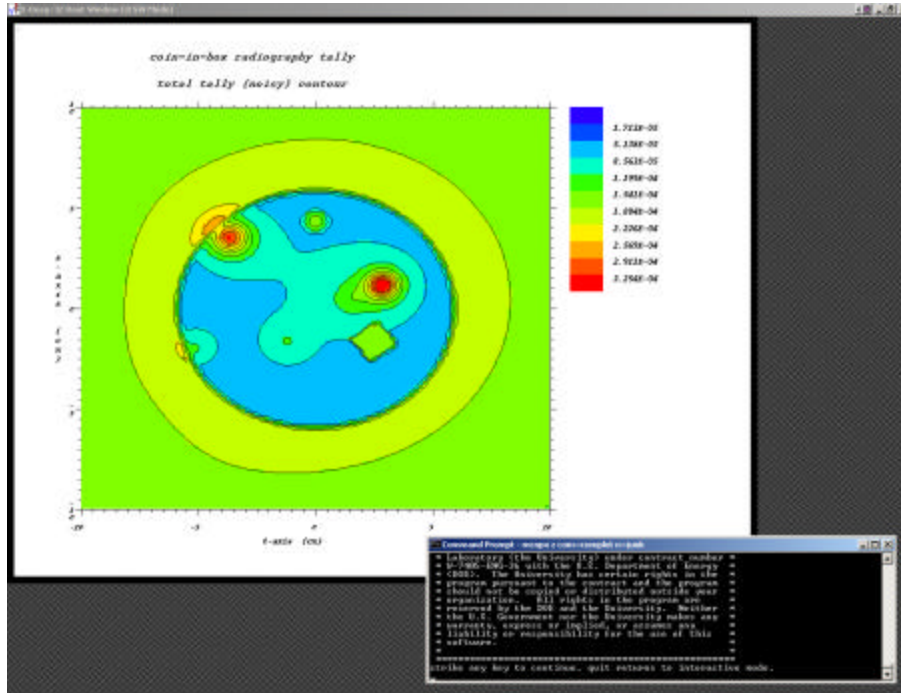


Figure 51. MCNPX Contour Plot.

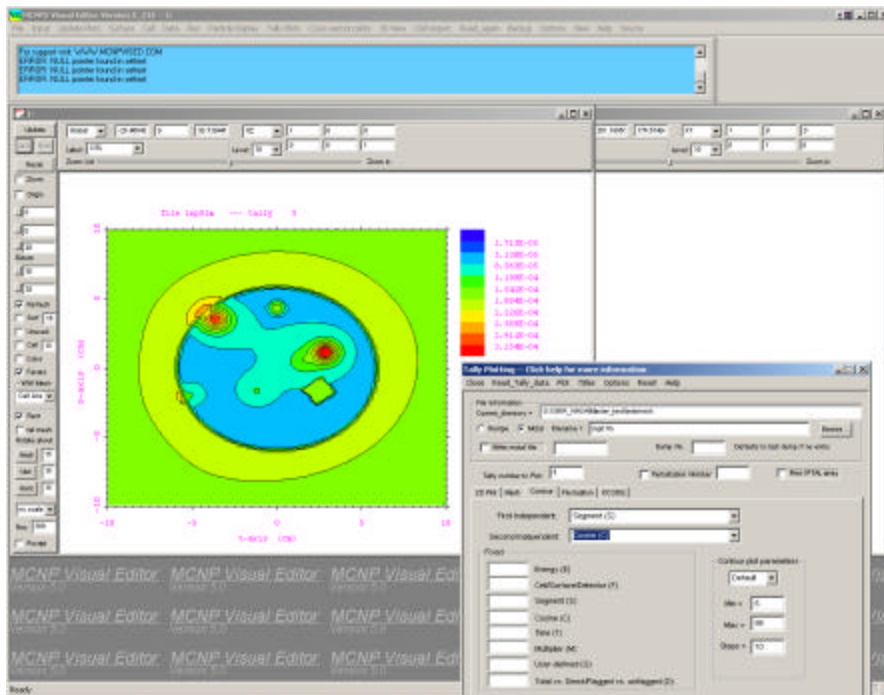


Figure 52. Visual Editor Contour Plot.

Figure 53 shows a contour plot generated with the MCNPX tally plotter using the command “rnc inp01m tal 5 free sc contour line fixed d 2”. Figure 54 shows the same plot generated in the MCPNX Visual Editor.

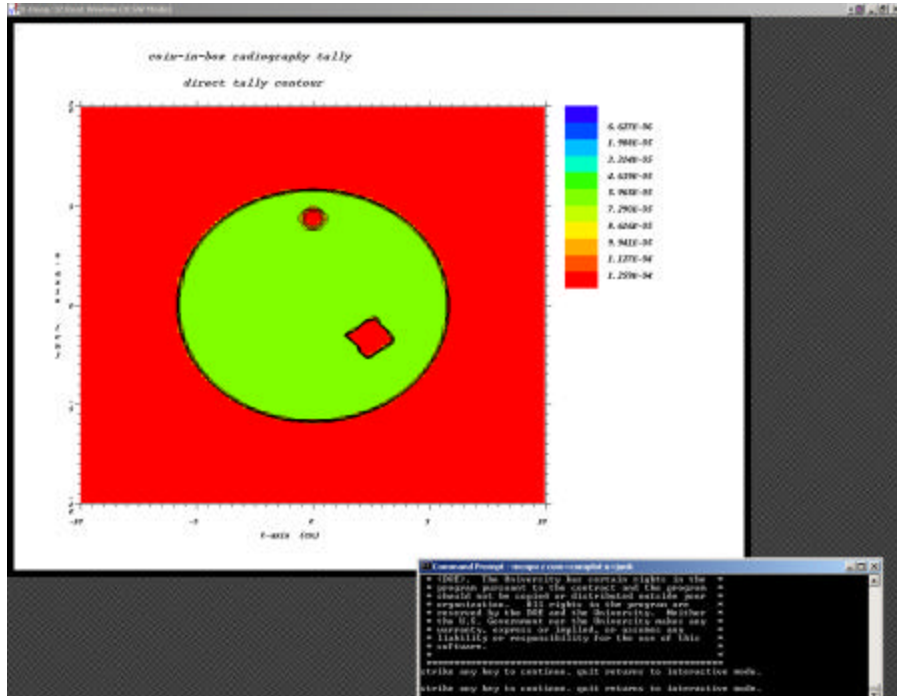


Figure 53. MCNPX Contour Plot with a Fixed Variable.

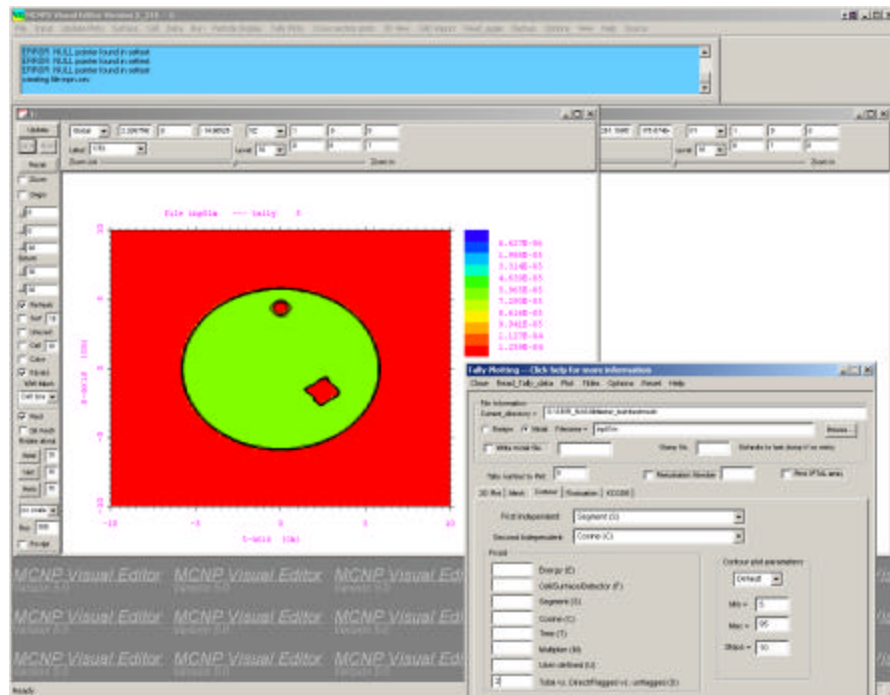


Figure 54. Visual Editor Contour Plot with a Fixed Variable.

Figure 55 shows a contour plot for testmesh input file 4 generated with the MCNPX tally plotter using the command “tal 3 free ik”. Figure 56 shows the same plot generated in the MCPNX Visual Editor.

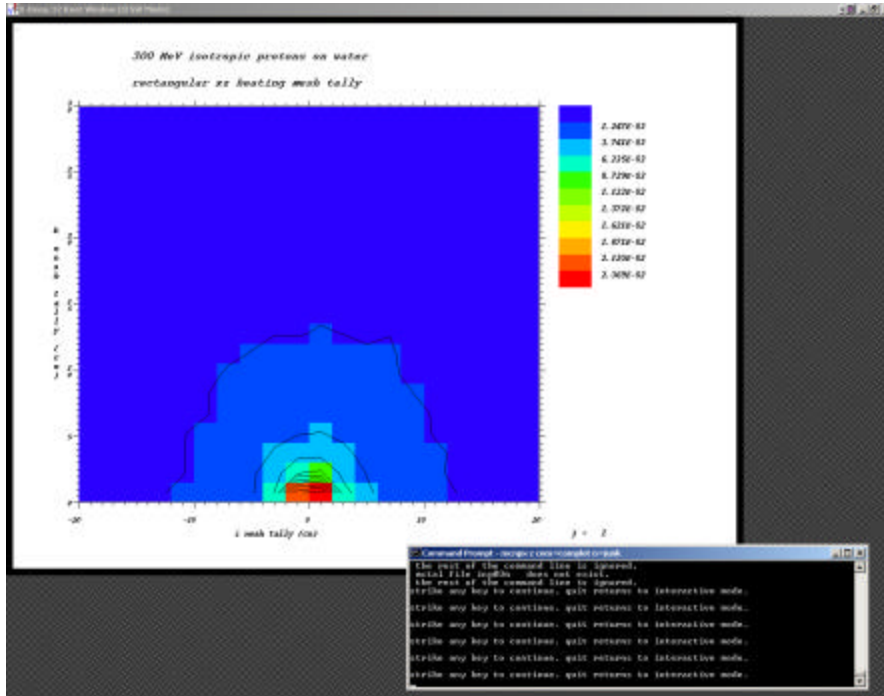


Figure 55. MCNPX Mesh Tally Plot.

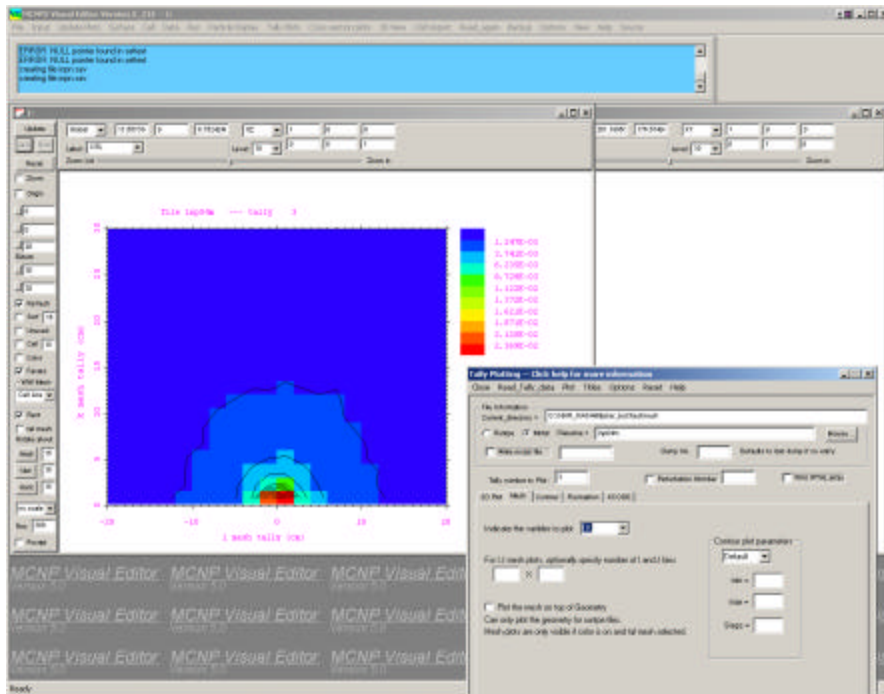


Figure 56. Visual Editor Mesh Tally Plot.

Figure 57 shows a contour plot for testmesh input file 4 generated with the MCNPX tally plotter using the command “tal 1 free jk contour log”. Figure 58 shows the same plot generated in the MCPNX Visual Editor.

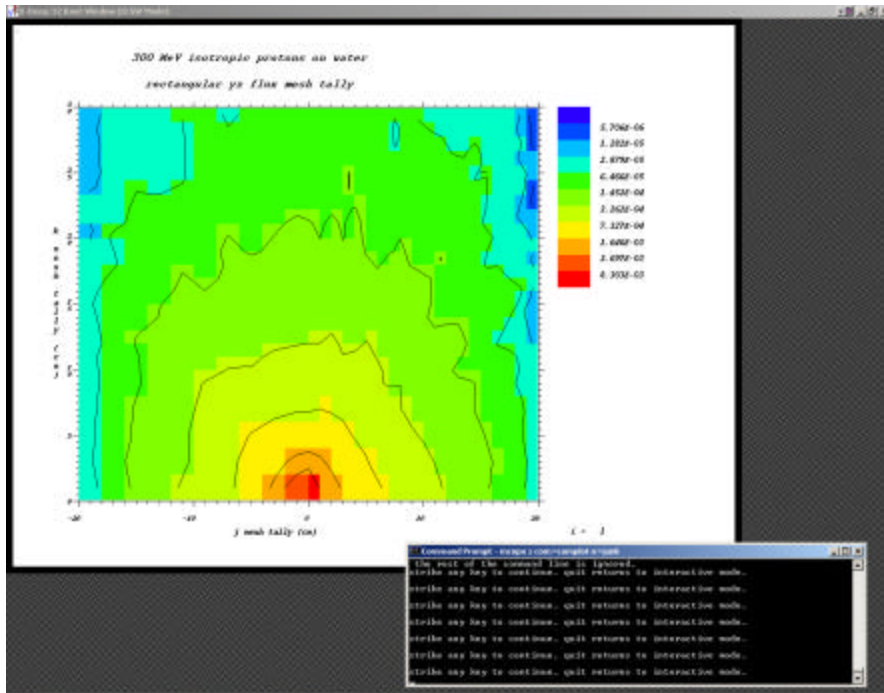


Figure 57. MCNPX Mesh Tally Plot.

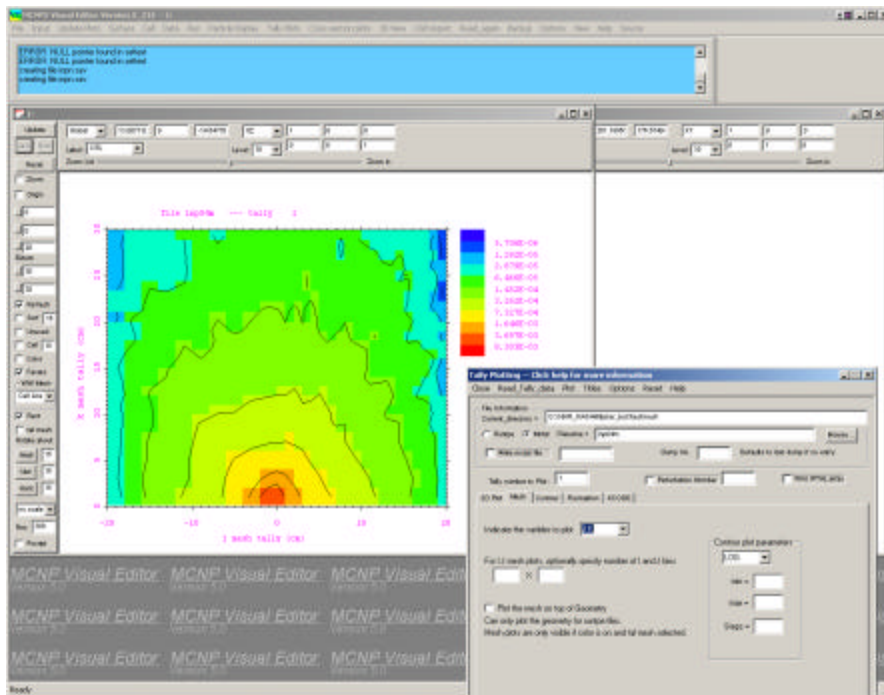


Figure 58. Visual Editor Mesh Tally Plot.

Figure 59 shows a contour plot for testmesh input file 4 showing the same mesh in cylindrical profile generated with the MCNPX tally plotter using the command “tal 11 free ij contour 5e-5 .02 20 log”. Figure 60 shows the same plot generated in the MCPNX Visual Editor.

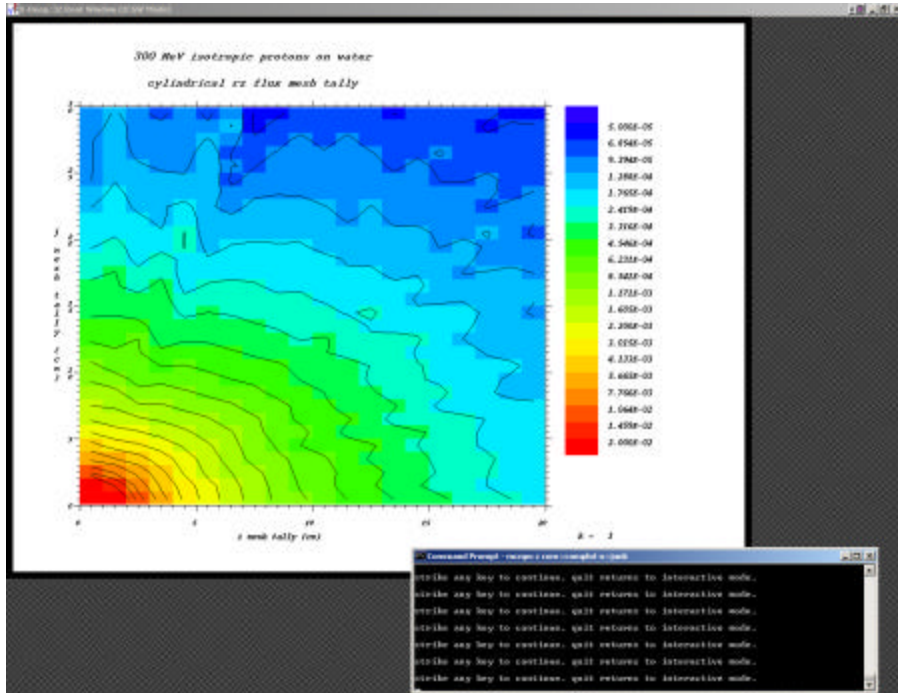


Figure 59. MCNPX Mesh Tally Plot.

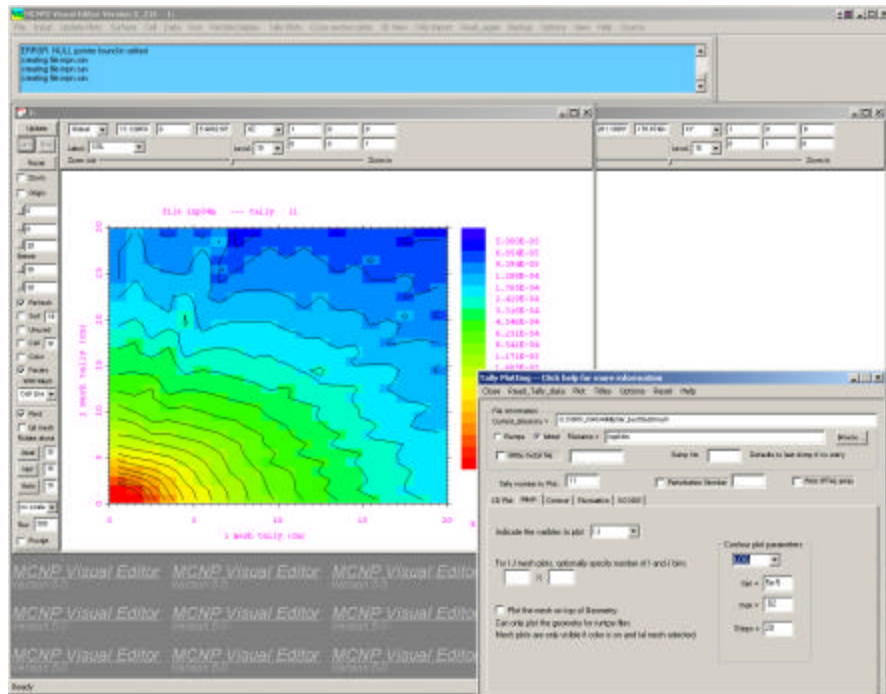


Figure 60. Visual Editor Mesh Tally Plot.

Figure 61 shows a contour plot for testmesh input file 5 generated with the MCNPX tally plotter using the command “rnc inp05m tal 1 free jk contour 5 95 10 %”. Figure 62 shows the same plot generated in the MCPNX Visual Editor.

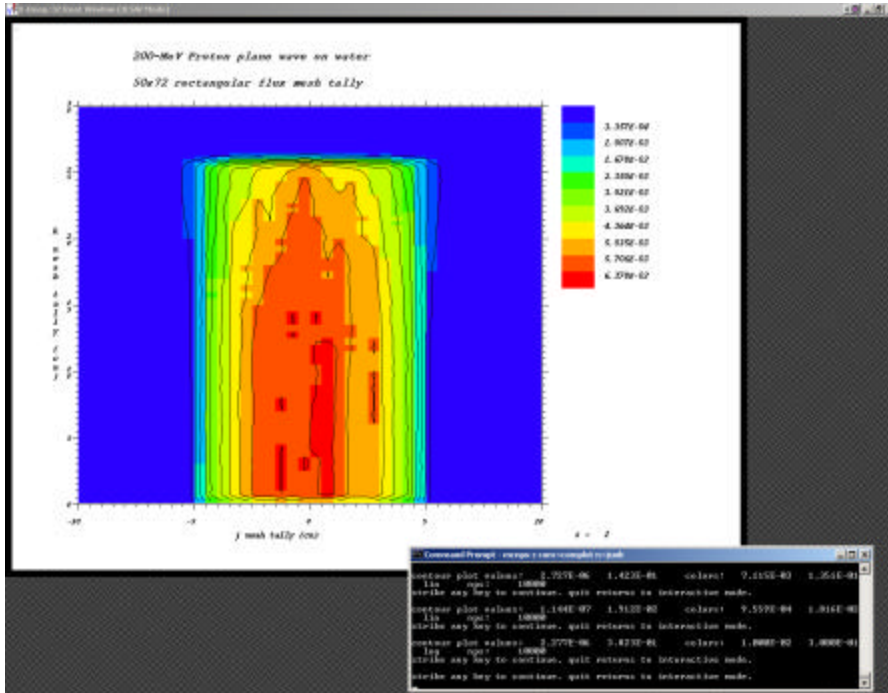


Figure 61. MCNPX Mesh Tally Plot.

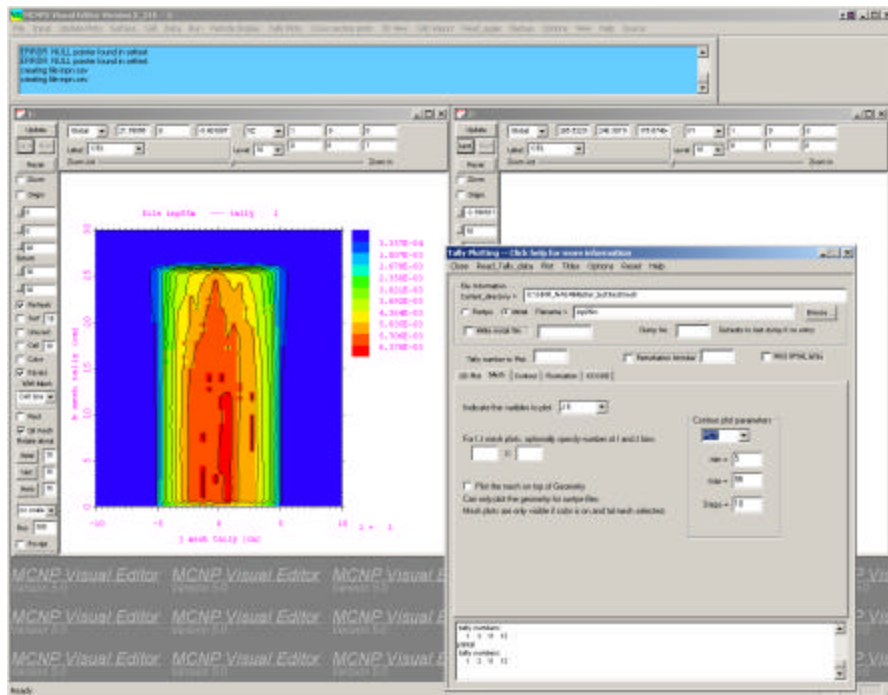


Figure 62. Visual Editor Mesh Tally Plot.

Figure 63 shows a contour plot for testmesh input file 4 generated with the MCNPX tally plotter using the command “tal 13 free ij contour .01 .3 20 log”. Figure 64 shows the same plot generated in the MCPNX Visual Editor.

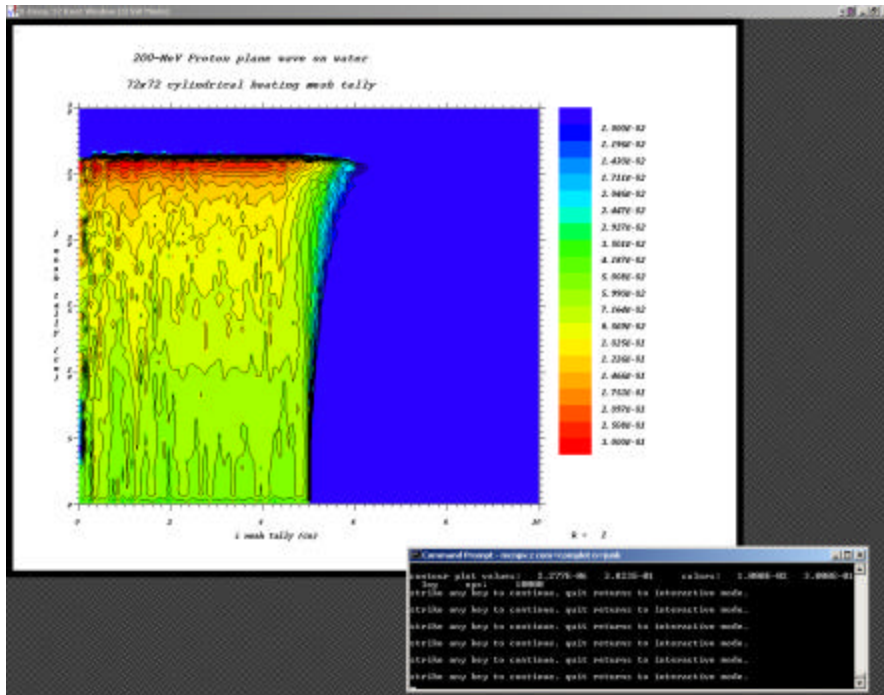


Figure 63. MCNPX Mesh Tally Plot.

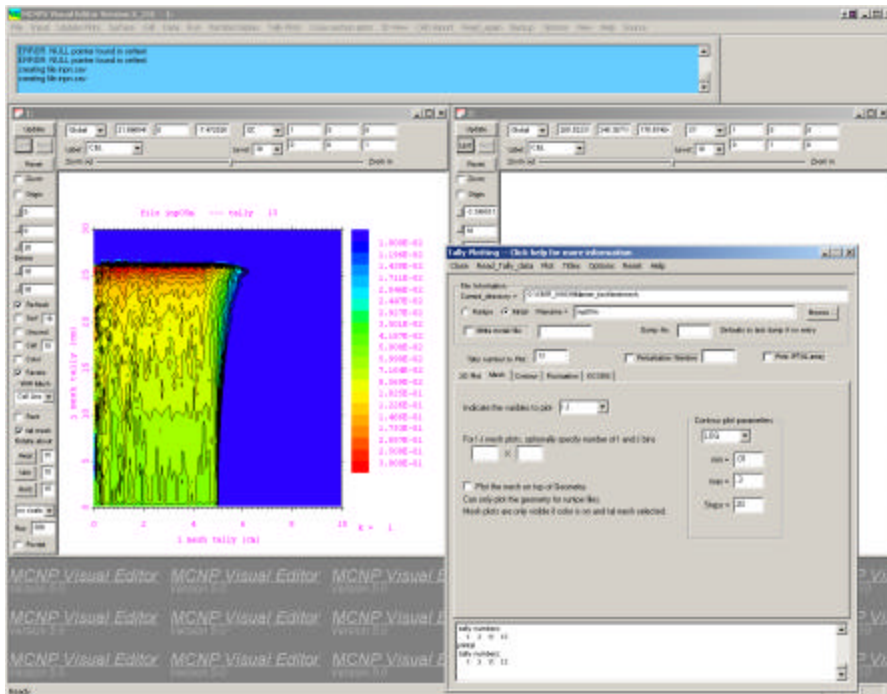


Figure 64. Visual Editor Mesh Tally Plot.

Figure 65 shows a contour plot for testmesh input file 10 generated with the MCNPX tally plotter using the command “rnc inp10m tal 2 free jk con 1e-5 .05 20 log”. Figure 66 shows the same plot generated in the MCPNX Visual Editor.

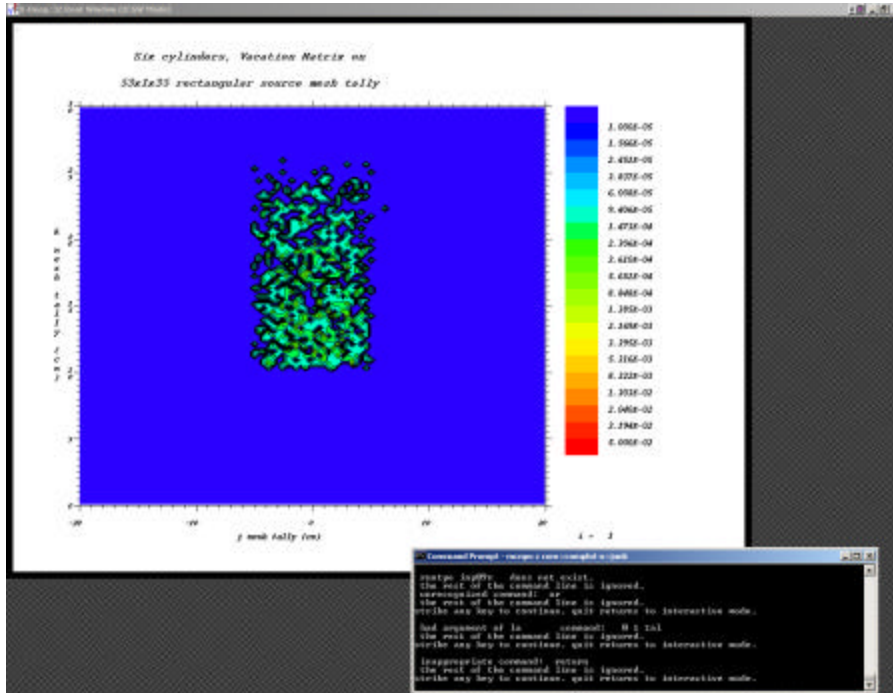


Figure 65. MCNPX Mesh Tally Plot.

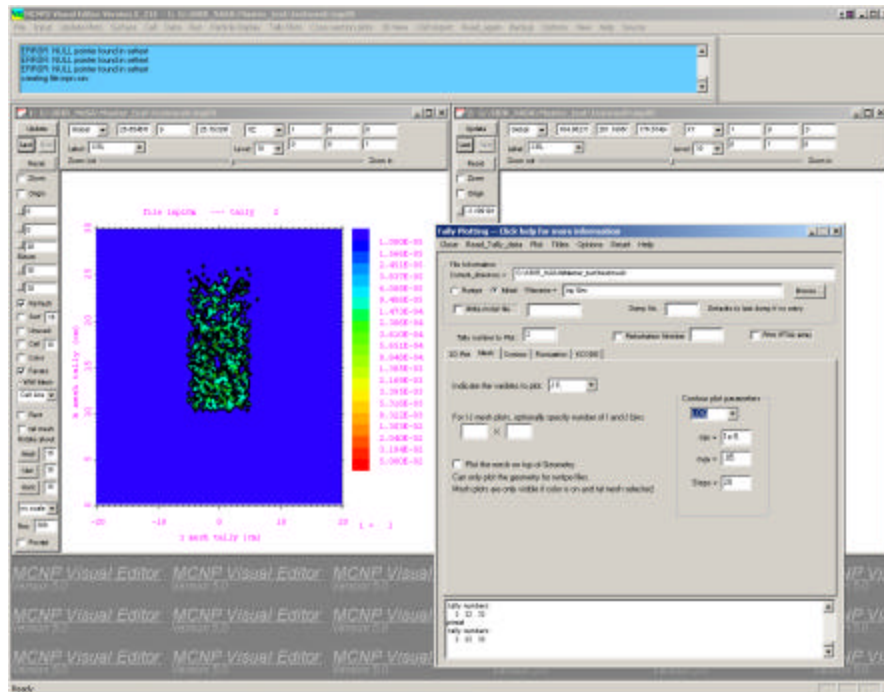


Figure 66. Visual Editor Mesh Tally Plot.

Figure 67 shows an expanded view of the previous tally plot within the MCNPX Visual Editor.

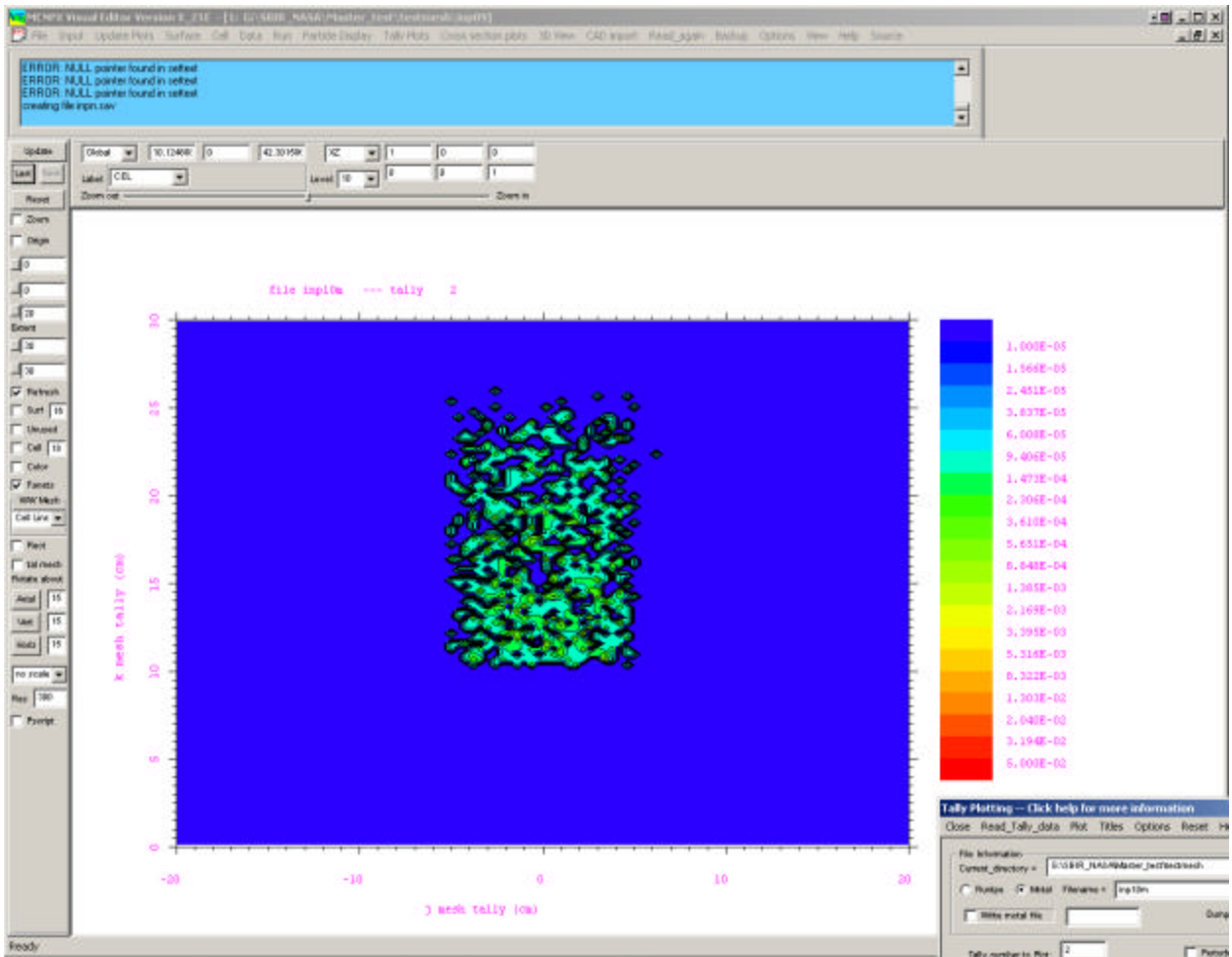


Figure 67. Enlarged View of a Visual Editor Mesh Tally Plot.

7.1.7 Task 7: Improve 3D Visualization Tools

The dynamic 3D visualization capability has been upgraded to include the ability to display both universes and lattices. Figure 68 shows a plot of a cask that contains 16 pins; each of these pins is contained in a universe. The user must specify both the universe and non-universe cells to display and then select “3D display” to show the plot. In the 3D dynamic view, the user can rotate the image, zoom in and out, and change the transparency of the cells.

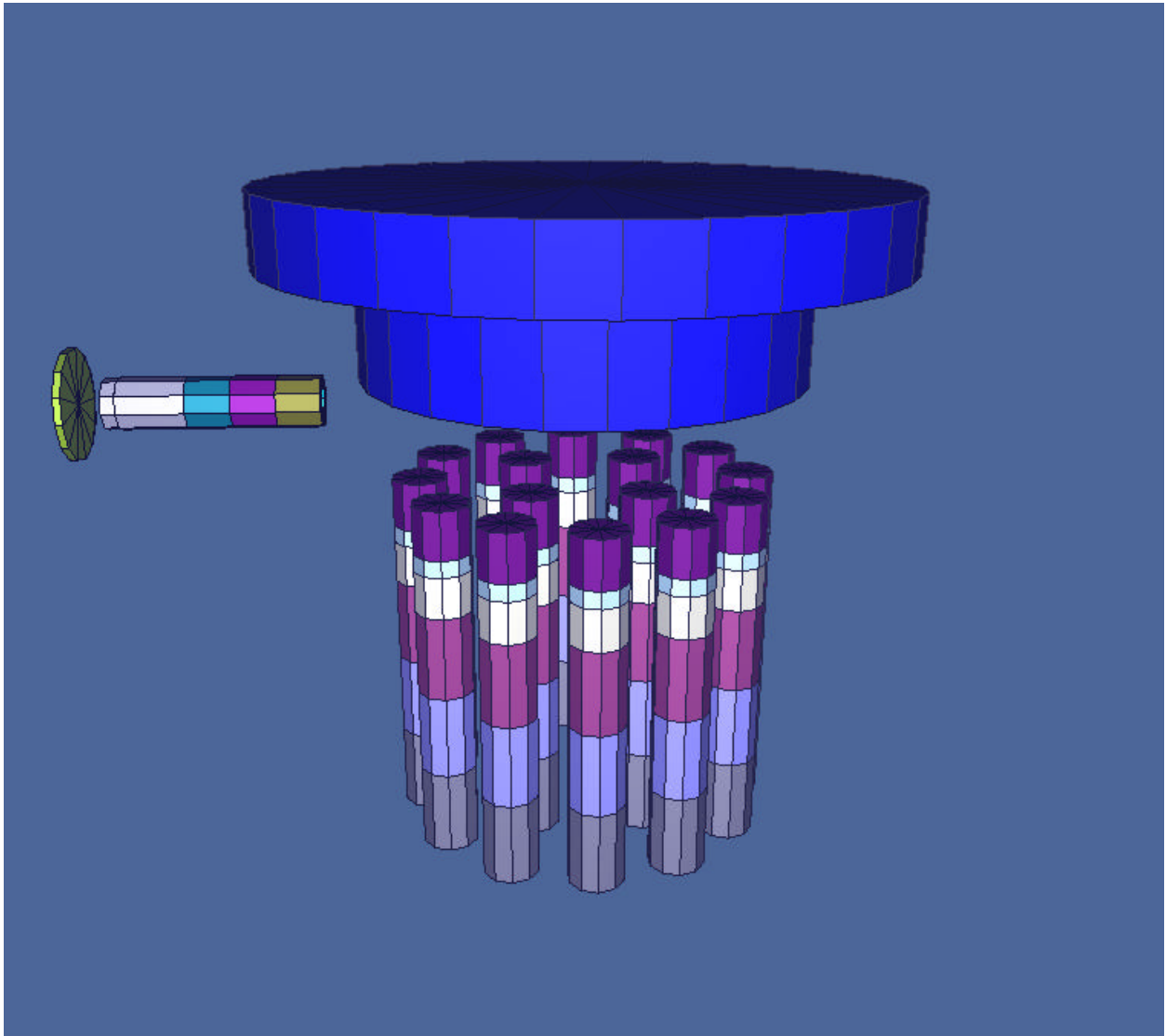


Figure 68. Dynamic 3D View Showing Universes.

Figure 69 shows a 2D geometry plot of a lattice that contains 5 cylinders that are defined as cell 3.

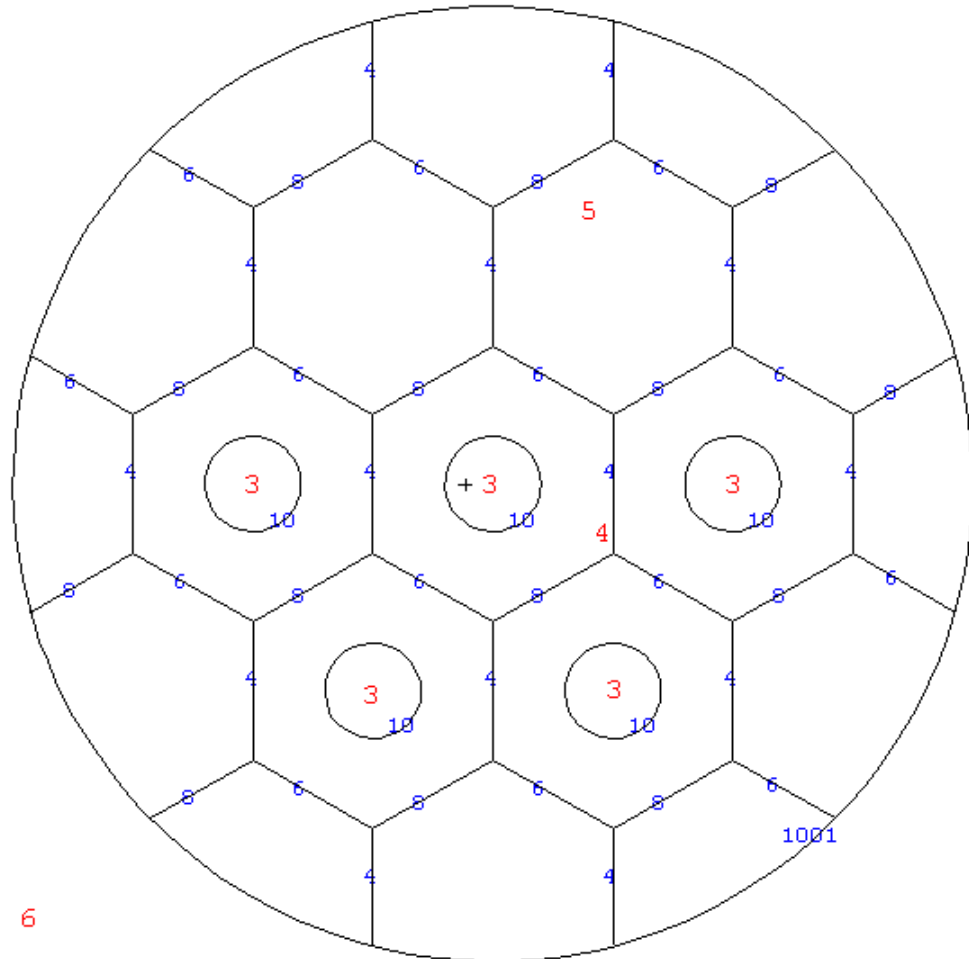


Figure 69. Plot of a Simple Lattice Geometry Containing Universe Cell 3.

Figure 70 shows a dynamic 3D view of the lattice where the outer fill cylinder has been made transparent.

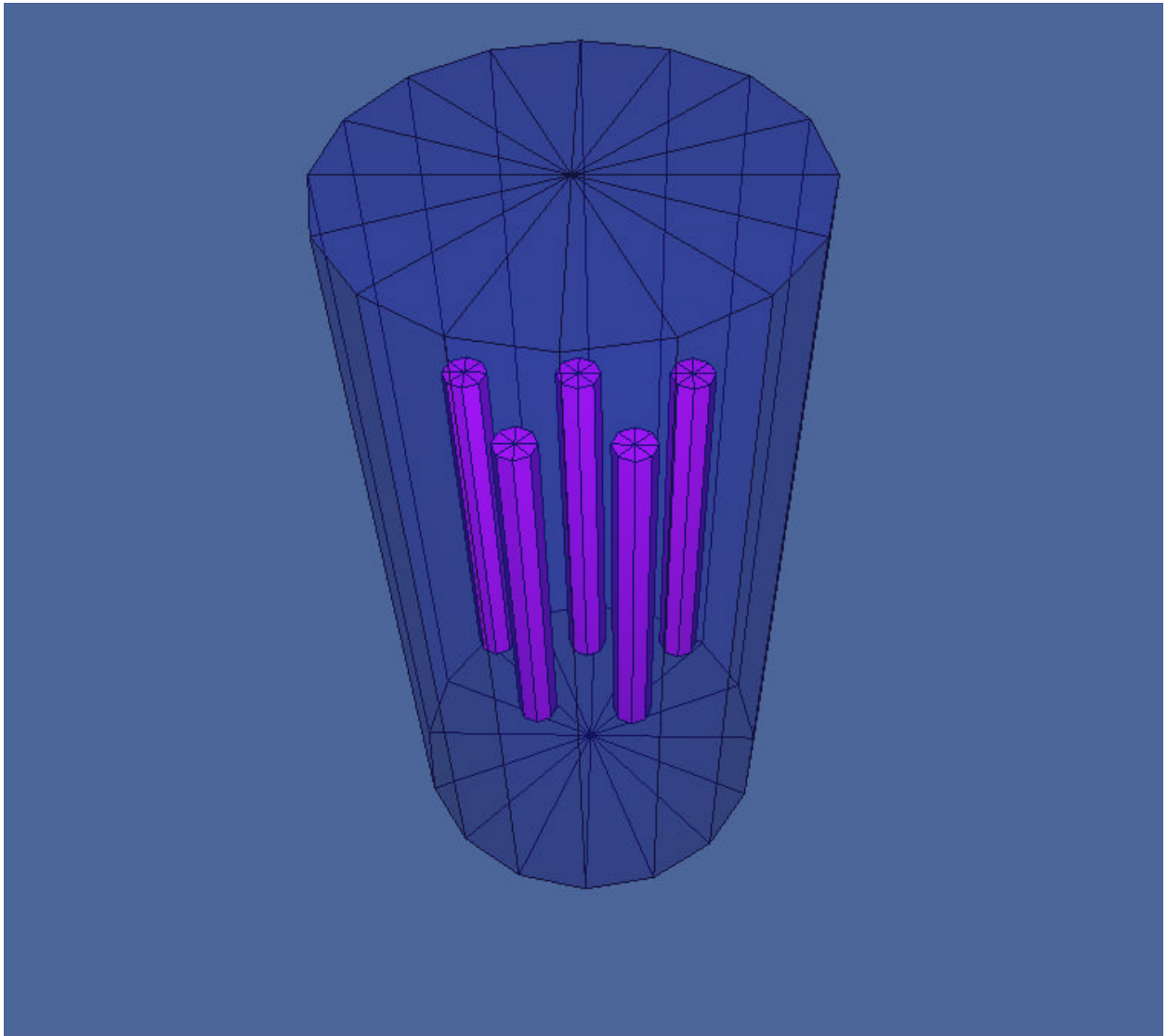


Figure 70. 3D Dynamic View of the Lattice.

Additional improvements that have been made to the 3D dynamic view include allowing the user to specify which transparency each cell should have and allowing users to change color of the 3D cells. Figure 71 shows a view of a cask with different cells set to solid, wireframe and transparent. Once these are specified, the user can save these values for this input file by selecting the “save parameters” menu option.

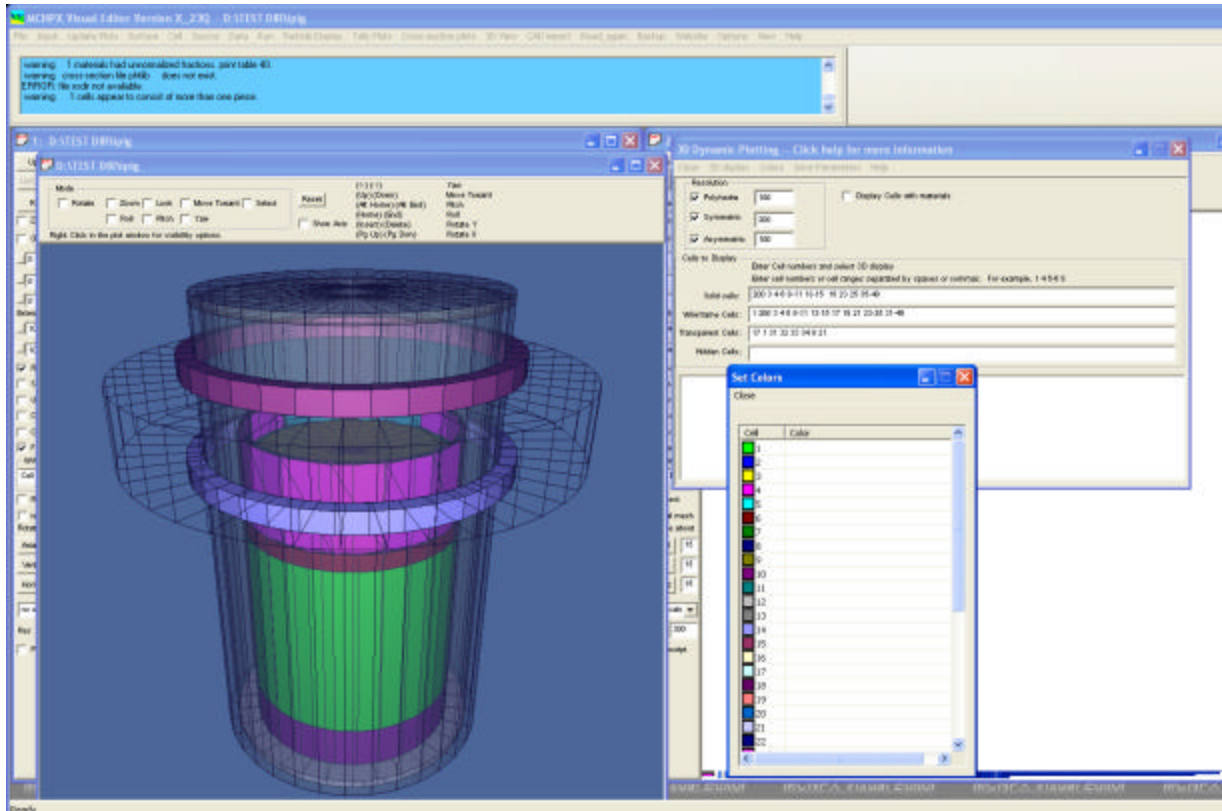


Figure 71. 3D Dynamic View with different transparencies specified

Also shown in Figure 71 is a list of colors for each of the cells that the user can set to a specific value.

Figure 72 shows another 3D dynamic view. In this figure, the source is shown as solid and the outlines of the glovebox are shown in wireframe.

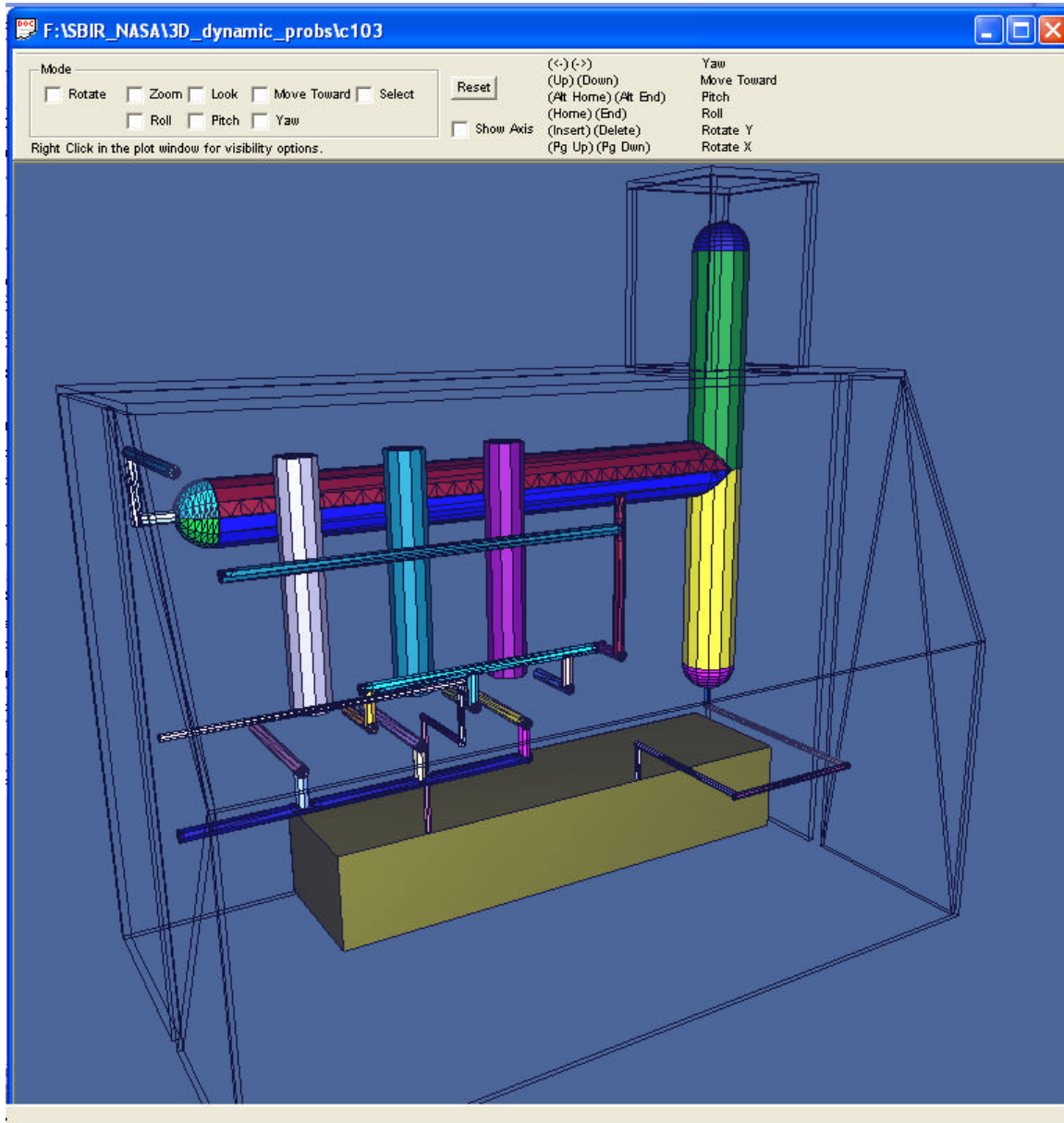


Figure 72. 3D Dynamic View of a Glove Box

Keyboard shortcuts have been added to allow the object to be rotated using keys. These shortcuts are shown at the top of the plot window.

7.1.8 Task 8: General Enhancements to MCNPX Visual Editor

An item that is of particular interest to NASA is the ability to read in large voxelized human phantoms to study the effects of space radiation on humans. One example of such a phantom is the Korean Typical Man-2 (KTMAN-2) model that contains a three-dimensional lattice with a voxel array size of 300 X 150 X 344. This comes out to more than 1.5 million voxels. Initial research was performed to see what modifications had to be made to the Visual Editor to view this geometry.

Figure 73 shows a plot of the KTMAN geometry displayed in the Visual Editor. Some of the features of the Visual Editor had to be disabled to make enough room in memory to hold this geometry. More research should be done to improve the memory management in both MCNPX and the Visual Editor to allow these types of geometries to be read in. A full resolution of this problem goes beyond the scope of the current Phase 2 proposal.

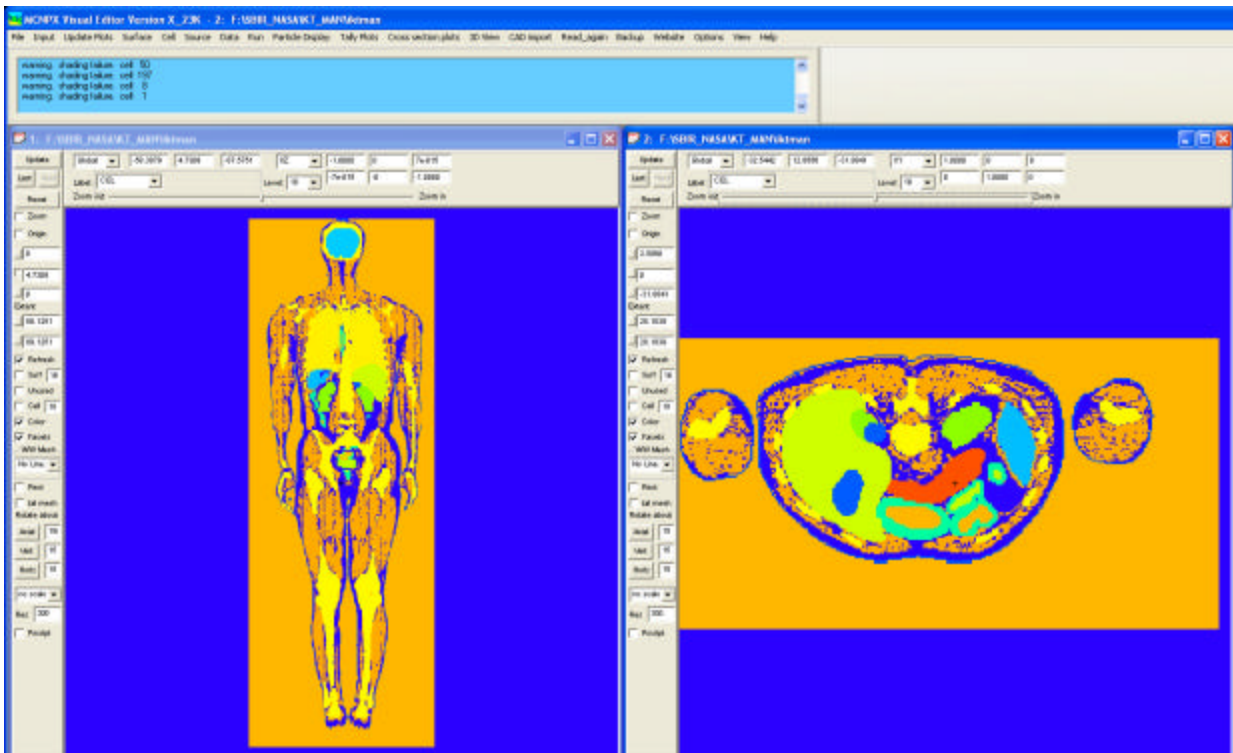


Figure 73. 2D Plots of KTMAN

7.1.9 Task 9: Create a Set of Test Problems

A review of the test problems from the Phase 1 effort was performed and a number of problems have been solved. A bug with 3D ray-tracing was fixed along with a number of other bugs involved in reading in the 65 test problems. All of these input files, except for the continue runs, now read into the Visual Editor and generate a valid plot. This means the Fortran code is

accurately passing the input file to the Visual Editor. Several cases where the Visual Editor is not passing the correct information back to the Fortran are currently being investigated.

To test the MCNPX Visual Editor for MCNPX data, the test input files that are sent out with the distribution of the beta code were read into the Visual Editor to see what problems were encountered. Table 7-1 shows a list of these 65 test problems and the results.

Four different tests were performed on each input file. The first test was to open the file in the Visual Editor. The Visual Editor will read the input file using the MCNPX Fortran code and then the information for the input is passed to the Visual Editor C++ code. If the Fortran has a problem reading the input or passing the information, this test will fail. All of the test problems pass this test.

The second test is to use the input window to save and then update the input file. This will show if the information passed from the Fortran to the Visual Editor is correct. If the information is incorrect in the Visual Editor, when it is read in again, the code will have a fatal error or die. Table 7-2 shows there are eleven problems that do not pass this test.

The third test is to try to run the problem within the Visual Editor. This will show if the Visual Editor can run MCNPX from the GUI. This must work or particle track plotting and 3D ray tracing will not work, because both of these features involve running the MCNPX code from the GUI. All but two of the test problems that do not have an intentional fatal error will now run in the Visual Editor.

The fourth test is to run the Visual Editor from the command line. The Visual Editor can be run from the command line to test running the problem using the same script used to test the MCNPX code. This works for all but a couple of problems.

Table 7-2. Test Problems Included with the MCNPX Source Code

(Input files from: ..\MCNPX26a\v26a\src\Test\Test_sources)

Case	Name	Input Has Fatal Error	Test 1 Opens in Vised	Test 2 Save-Update in Vised	Test 3 Runs in Vised GUI	Test 4 Runs in Vised Command Line	Comments
1	inp01		OK	OK	YES	YES	
2	inp02	X	OK	OK	NO	YES	
3	inp03		OK	OK	YES	YES	
4	inp04		OK	OK	YES	YES	
5	inp05		OK	OK	YES	YES	
6	inp06		OK	OK	YES	YES	
7	inp07		OK	OK	YES	YES	
8	inp08		OK *	OK	YES*	YES	*rssa & wwinp must be defined
9	inp09		OK	OK	YES	YES	
10	inp10		OK	OK	YES*	YES	*wwinp must be defined
11	inp11	X	OK	OK	NO	YES	
12	inp12		OK	OK	YES	YES	
13	inp13		OK	OK	YES	YES	
14	inp14		OK	OK	YES	YES	
15	inp15		OK	FE ¹	YES	YES	FE on S-U ² cell tr wrong
16	inp16		OK	FE	YES	YES	FE on S-U lat tr wrong
17	inp17		OK	GE ³	YES	YES	GE on S-U
18	inp18		OK	OK	YES	YES	
19	inp19		OK	OK	YES	YES	
10	inp20	X	OK	OK	NO	YES	
21	inp21		OK	OK	YES	YES	
22	inp22		OK	OK*	YES*	YES	*Must specify rssa file
23	inp23	X	OK	OK	NO	YES	
24	inp24		OK	OK	YES	YES	
25	inp25		NA	NA		YES	Continue run
26	inp26		NA	NA		YES	Continue run
27	inp27		OK	OK*	YES*	YES	*Must specify rssa file
28	inp28		OK	OK	YES	YES	
29	inp29	X	OK	DIES	NO	YES	Dies on S-U
30	inp30		OK	FE	YES	YES	FE on S-U
31	inp31		OK	OK	YES	YES	
32	inp32		OK	OK	YES	YES	
33	inp33		OK	OK	YES	YES	
34	inp34		OK	OK	YES	YES	Must specify rssa file
35	inp35		OK	OK	YES	YES	
36	inp36		OK	OK	YES	YES	
37	inp37		OK	OK	YES*	YES	*must delete files
38	inp38		OK	OK	YES	YES	
39	inp39		OK	DIES	YES	YES	bad tmesh data
40	inp40		OK	OK	YES	YES	
41	inp41		OK	OK	YES	YES	

Case	Name	Has Fatal Error	Test 1 Opens in Vised	Test 2 Save-Update in Vised	Test 3 Runs in Vised GUI	Test 4 Runs in Vised Command Line	Comments
42	inp42		OK	OK	YES	YES	
43	inp102		OK	OK	NO	YES	Dies in nps=8
44	inp103		OK	OK	YES	YES	
45	inp104		OK	OK	YES	YES	
46	inp105		OK	OK	YES	YES	
47	inp106		OK	OK	YES	YES	
48	inp107		OK	OK	YES	YES	
49	inp108		OK	FE	YES	YES	FE on S-U
50	inp109		OK	OK	YES	YES	
51	inp110		OK	OK	YES	YES	
52	inp117		OK	OK	YES	YES	
53	inp118		OK	OK	NO	YES	Dies in nps=18
54	inp119		OK	OK	YES	YES	
55	inp120		OK	OK	YES	YES	
56	mes01		OK	DIES	YES	DIES	Bad mesh data
57	mes02		OK	DIES	YES	YES	Unexpected EOF
58	mes03		OK	DIES	YES	YES	Unexpected EOF
59	mes04		OK	DIES	YES	DIES	Bad mesh data
60	htp111		OK	OK	YES	YES	
61	htp112		OK	OK	YES	YES	
62	htp113		OK	OK	YES	YES	
63	htp114		OK	OK	YES	YES	
64	htp115		OK	OK	YES	YES	
65	htp116		OK	OK	YES	YES	

¹ FE – fatal error

² S-U – Save-Update

³ GE – geometry error