ANL-18/39

IBR-2M REACTOR MODELING WITH MCNP, SERPENT, CUBIT, AND GIMP COMPUTER PROGRAMS

Nuclear Science and Engineering Division

IBR-2M REACTOR MODELING WITH MCNP, SERPENT, CUBIT, AND GIMP COMPUTER PROGRAMS

ANL-18/39

IBR-2M REACTOR MODELING WITH MCNP, SERPENT, CUBIT, AND GIMP COMPUTER PROGRAMS

prepared by

Alberto Talamo, Yousry Gohar, Zhaopeng Zhong, Yan Cao,
Argonne National Laboratory, 9700 South Cass Avenue, Argonne, IL 60439, USA

Igor Bolshinsky,
Idaho National Laboratory, P. O. Box 2528, Idaho Falls, Idaho 83403, USA

Yu. N. Pepelyshev, Alexander Vinogradov
Joint Institute for Nuclear Research, 141980 Dubna, Russia

October 2018

IBR-2M REACTOR MODELING WITH MCNP, SERPENT, CUBIT, AND GIMP COMPUTER PROGRAMS

Table of Contents

IBR-2M REACTOR MODELS WITH MCNP, SERPENT, CUBIT, AND GIMP COMPUTER PROGRAMS

List of Figures

List of Tables

Table No. **Page**

IBR-2M REACTOR MODELING WITH MCNP, SERPENT, CUBIT, AND GIMP COMPUTER PROGRAMS

Abstract

IBR-2M is a fast research reactor located in Dubna (Russia), which operates in supercritical condition for ~800 μs every 200 ms. Two reflector parts are rotating in opposite directions generating 1.8 GW_{th} peak power when the two parts align with the fuel zone. The reactor core uses plutonium fuel and sodium coolant. The control rods are located in stationary reflector zones.

This report describes the computational modeling of the IBR-2M reactor. Two independent computational models were generated using MCNP and SERPENT computer programs. The MCNP model uses combinatorial geometry based on macro bodies, e.g. boxes, cylinders, hexagons, spheres, cones, etc., and surfaces. The SERPENT model is based on the STereoLithographic (STL) geometry representation that is used by 3D printers. The STL geometry was constructed using the CUBIT computer program.

The water cooling system of the stationary and the rotating reflectors has a very complicated geometry, involving serpentines and irregular surfaces profiles. The materials in these areas have been homogenized using the GIMP image editor according to the material distributions obtained from the colors of the design drawings. The MCNP and the SERPENT inputs consist of ~1500 and ~1750 lines, respectively. The calculated effective multiplication factors from these two Monte Carlo computer programs with different models agree within 50 pcm.

IBR-2M REACTOR MODELING WITH MCNP, SERPENT, CUBIT, AND GIMP COMPUTER PROGRAMS

1. Introduction

The IBR-2M research facility is located in Dubna, Russia, and the current reactor configuration has been in operation since the end of 2011. It is a fast research reactor, which operates in supercritical condition for ~800 μs every 200 ms. The reactor core uses high-enriched plutonium fuel and it is cooled by sodium. The moving reflector consists of two blades rotating in opposite directions that change the reactor from deep subcritical to supercritical status. The supercritical status occurs when both nickel reflector blades are aligned with the fuel zone. The average thermal power during the pulse period is 2 MW_{th} and the peak power is 1830 MW_{th}. When the reflector blades are not aligned with the fuel, the power reduces to 0.16 MW. The reactor is used for experimental research in different physics areas, including: condensed matter, biology, chemistry, and material science[1-16].

The IBR-2M reactor consists of the core containing the fuel assemblies, the stationary and the rotating reflectors, the control rods, the cooling system of the stationary and rotating reflectors, the vessel, and the reactor shield. All the control rods are located in the stationary reflector. Section 2 of this report discusses the details of each component except the reactor shield, which has not been included in the computational models. Section 3 highlights the techniques used to develop the computational models from the technical specifications and the colored design drawings. Finally, Section 4 compares the effective multiplication factor results calculated by the MCNP and the SERPENT Monte Carlo computer programs.

2. IBR-2M Computational Model

The first computational model of the IBR-2M reactor was developed using the MCNP computer program[17] and the ENDF/B-VII.0 nuclear data library[18]. Figures 1 through 48 illustrate the details of the MCNP model by showing different cross sections at planes perpendicular to the x, y, and z axes, as specified in the captions of these figures. The z-axis is parallel to the fuel rods axes. The figures have been cropped by the VISUALBASIC macro[19] listed in Appendix A to remove the MCNP text on the left side of the figures. Tables I and II summarize the technical specifications used in the MCNP input and the colors of the headings of Table II distinguish the different materials used in Figs. 1 to 48. In these figures and the tables, the model origin is located at the center of Fig. 48 in the x-y plane and at the middle of the active fuel length in the z-direction. The scales on the figures frames refer to local origins, as specified in the figure captions. The MCNP input has been updated using the Revision Control System (RCS) of Linux[20] and the commands summarized in Appendix B. Eight MATLAB scripts[21] were used to prepare the MCNP model, as discussed in Section 3 and Appendix C.

The CUBIT program[22] has been used for a three-dimensional visualization of the MCNP model, as shown in Figs. 49 to 66. The CUBIT model of the IBR-2M reactor has 4861 volumes, 27433 surfaces, 29508 curves, and 26951 vertices (key points). The MCNP and

CUBIT inputs have 1500 and 2600 lines, respectively; however, 1412 lines of the CUBIT input were the output of the three MATLAB scripts, listed in Appendix D. These scripts use the central fuel assembly to build the hexagonal array of fuel assemblies in the fuel zone.

2.A Fuel Rods

The fuel rod geometry is shown in the top view of Fig. 20. The fuel material is plutonium oxide and each fuel rod contains 185 g of plutonium oxide with 95.7% ^{239}Pu enrichment. The plutonium oxide density is 10.103 g/cm^3. The active and the total fuel rod lengths are 44.4 and 77.8 cm, respectively. The fuel pellets have a central hole with 0.75 mm radius to accommodate the fission gases and to reduce the peak temperature in the fuel material. Helium gas fills this hole and the interstitial gap between the fuel pellets and the clad. The fuel pellets radius and the clad thickness are 0.37 cm and 0.15 mm, respectively. The wrapping wire on the fuel clad has been taken into account by increasing the clad nominal radius by 63 μm, as explained in Section 3.

The top inactive part of the fuel rods has a 6 cm tungsten reflector and a stainless steel zone with a spring to accommodate the thermal expansion of the fuel material and the fission gas products, as shown at the bottom of Fig. 13. Above the spring zone, the fuel rod radius is reduced to 0.125 cm through a conical transition over 0.2 cm length, as shown at the top of Fig. 14.

The bottom inactive part of the fuel rod consists of a stainless steel zone, a reduced radius zone, and a sodium-steel zone. These zones have 0.8, 0.5, and 0.7 cm lengths, respectively, as shown at the top of Fig. 15. Two conical transitions unite the previous three zones. The sodium and steel materials in the 0.7 cm long zone have been homogenized using GIMP and MATLAB, as described in Section 3, using Script 7 listed in Appendix C.

2.B Fuel Assemblies

The fuel assembly has a hexagonal shape and contains 7 fuel rods and 6 axial steel spacers, as shown in Fig. 48. The zone above the fuel rods, shown at the bottom of Fig. 13, has been modeled with hollow steel cylinders, as shown at the bottom of Fig. 42 and in Fig. 43, followed by solid steel cylinders immerged in the sodium coolant, as shown at the top of Fig. 44. The zone below the fuel rods, shown in Fig. 19, has been modeled as hollow steel cylinders, as shown at the bottom of Fig. 36. These hollow steel cylinders penetrates the steel support grid, as shown at the top of Fig. 36. The latter is perforated by sodium channels with 1.125 cm radius, as shown at the bottom of Fig. 35, that shrink due to the penetration of the steel hollow cylinders of the fuel assemblies, as shown at the top of Fig. 36 top.

The computational model of the zones above and below the fuel rods does not account for:

- the transition at the top and bottom axial boundaries of the fuel rods where the fuel assembly shape changes from hexagonal to cylindrical;

- the holes, which allow sodium to pass through the hollow steel cylinders in the zones above and below the fuel rods; and
- the transition from hollow to solid steel cylinders in the zone above the fuel rods.

In this study, the IBR-2M reactor core is loaded with 64 fuel assemblies and 5 copper assemblies, as shown in Fig. 47. The copper assemblies are the same as the fuel assemblies with the exception that copper replaces plutonium oxide and there is no central hole in the copper material.

2.C Reactor Vessel

The steel vessel around the fuel zone has a hexagonal shape with three different apothems as shown in Fig. 47 and its length in the vertical z direction is 116 cm, as shown at the top of Fig. 30. The vessel consists of two steel layers separated by a thin air layer. The thickness of the inner and the outer steel layers are 0.7 and 0.4 cm, respectively; the air layer thickness is 0.3 cm.

Under the steel grid supporting the fuels rods, the vessel shape changes from hexagonal to cylindrical with 5.7 cm inner radius, as shown in Figs. 31 through 34. Above the fuel assemblies, the vessel shape also changes from hexagonal to cylindrical with 15.2 cm inner radius, as shown in Figs. 43 and 44. Above the fuel assemblies, the vessel contains two solid zones. One zone is made of steel and extends for 12.2 cm, as shown at the bottom of Fig. 44. The other zone is made of iron and boron carbide and it extends up to the end of the vessel, as shown at the top of Fig. 45. At the end of this zone, the vessel radius increases from 15.2 to 17.25 cm through a conical transition, as illustrated in Fig. 12. The total vessel length is 696 cm; this value includes 552.9 cm length of the cylindrical zones, as shown at the bottom of Fig. 11.

2.D Stationary Reflector and Control Rods

The stationary steel reflector has pairs of SCRAM, compensating, and manual control rods; and a single automatic control rod, as illustrated in Figs. 67 and 55. The stationary reflector length along the z-axis is 100 cm and the length of the control rods varies from 40 to 42.6 cm, as reported in Table I. All the control rods are made of tungsten alloy with the exception of the automatic control rod, which is made of pure beryllium, as shown in Fig. 22. The control rods are held from the top by steel support parts, which have 38 cm length along the z-axis, as shown in Fig. 7. All control rods and steel support parts have rectangular prism shapes with the exception of the automatic control rod, which has cylindrical shape.

2.E Water Cooling System of the Stationary Reflector

Three boron carbide layers, allocated next to the stationary reflector shown in Figs. 55 through 58, prevent thermal neutrons from entering the fuel zone from the stationary reflector. The thickness and the axial length of the boron carbide layers are 1 and 44 cm, respectively.

The width of the boron carbide central layer is 21.6 cm, whereas the width of the right and the left layers is 25.8 cm, as shown in Fig. 46. All the boron carbide layers are axially centered at the middle of the active fuel length.

The cooling system of the stationary reflector has aluminum structure. The aluminum and the water materials are homogenized into four solid blocks, as illustrated in Figs. 46, 52, 54, and 58 left. The homogenization process was performed with GIMP[23] and MATLAB[21], as discussed in Section 3.

2.F Rotating Reflectors

The blades of the two rotating reflector parts are made of nickel alloy; the main rotating reflector blade has two rounded rectangle holes, whereas the auxiliary rotating reflector blade has just one rounded rectangle hole. The nickel blades are attached to steel blade-holders, as illustrated in Figs. 25 top, 26 top, 68 and 69. The rotating shaft has 26 cm radius and 11.76 cm length, as shown in Figs. 1 and 26 bottom. The main and auxiliary rotating reflectors and the shaft are inside a cylindrical steel casing, referred to as cylindrical shell in Fig. 75, with 121.6 and 130 cm inner and outer radii, respectively, as shown in Figs. 25 top, 26 top, and 75. Along the y-axis, the width of the cylindrical casing matches the rotating shaft.

The cylindrical steel casing has inner and outer triple-layer covers (jackets), as shown in Fig. 52. The inner triple-layer cover is next to the core. Inside this cylindrical casing and between the inner and outer triple-layer covers there are helium gas, the rotating reflector shaft, the two rotating reflectors blades, and the blade-holders. The triple-layer covers include two steel layers separated by a water layer, as shown in Figs. 46 top and 52. The water in the covers cools down the rotating reflectors materials. The thicknesses of the layers have been calculated by taking into account the covers design shown in Figs. 70 to 75 using the GIMP and MATLAB, as discussed in Section 3. The cylindrical steel casing and the triple-layer covers are shown in Fig. 75.

The inner and outer triple-layer covers have four and one rectangular cuts, respectively, as shown in Table I, that approximate the design of the jackets, as shown in Fig. 22 bottom, 23, 27 bottom, and 28. Without these cuts, the covers would have a solid cylinder shape.

2.G Water Cooling System of the Rotating Reflectors

The water cooling system of the rotating reflectors is located inside the cut of the outer triple-layer cover and is centered on the middle of the fuel zone along the x and the z-axes. This system is made of three aluminum rectangular layers separated by a water layer and a boron carbide layer. The latter layer lies on the core side as shown in Fig. 52, and it prevents thermalized neutrons from the water layer to enter the fuel zone.

3. GIMP Homogenizations and MATLAB Scripts

The GIMP computer program is an open-source image editor[23]. In the development of the MCNP model, GIMP was used to:

- calculate the distance in pixels between two points in a design image;
- calculate the angle between two lines in a design image;
- calculate the position of the centroid of a colored zone relative to the bottom-left corner of a design image[24];
- calculate the red, green, and blue colors distributions of a colored design image; for images with three solid colors, these distributions are proportional to the colors areas ratios;
- set transparent regions in design images; this allows the calculation of the red, green, and blue colors distributions;
- open multiple pages of a postscript file as image layers and automatically export them at once[25].

Appendix C lists the eight MATLAB scripts used for the preparation of the MCNP computational model. MATLAB was used to process the GIMP data and to write sections of the MCNP input by the fprintf instruction. In the MATLAB scripts the rgb variable, whenever defined, sets the distribution of red, green, and blue colors of an image from the GIMP data.

The first script has been used to increase the fuel clad outer radius to account for the steel wrapping wire.

The second script calculates the geometrical parameters of the planes that define the transition of the bottom vessel shape. These planes converge towards the z-axis at the center of the cylindrical section of the vessel, as shown in Fig. 55.

The third script calculates the geometrical parameters of the planes that define the stationary reflector. These planes are parallel to the z-axis, as shown in Figs. 47 top and 54.

The fourth script calculates the thickness of the triple-layer covers (jackets) of the rotating reflectors. In the MCNP model discussed in Section 2, the inner and the outer triple-layer covers consist of two pure steel layers and one pure water layer; the water layer is in-between the two steel layers. In the casing design, the central layer of the covers contains both water and steel materials, as illustrated in the central plot of Figs. 71 and 72. In the MCNP model, the steel in the central layers of the design covers, illustrated in Fig. 75, has been allocated to the outer steel layers by increasing their 0.8 cm nominal thickness. Accordingly, the nominal thickness of the central layers of the design covers has been diminished in the MCNP modeling (Fig. 75 versus Table I data).

The fifth script calculates the radii of the sodium channels and the steel cylinders at the top section of the fuel assembly as shown at the bottom of Fig. 13. The sixth script calculates the radius of the sodium channels at the bottom of the fuel assembly as shown at the bottom of Fig. 14. The MCNP input dimensions for the previous two scripts were taken from Figs. 76 and 77, which represent the design.

The seventh script calculates the homogenized composition of the steel and the sodium zone at the bottom of the fuel rods according to the design illustrated in plot B of Fig. 78. Plot A of Fig. 78 has been used to obtain the coordinates (in pixel unit) of the image center relative to the bottom-left corner. Plot B of Fig. 70 has been used to obtain the area ratio between steel and sodium zones. Plot C of Fig. 70 has been used to obtain the centroid coordinates of the sodium zone. Plot D of Fig. 70 has been used to obtain the centroid coordinates of the steel and sodium zones (altogether). The coordinates of the centroid center are relative to the bottom-left corner of plot A since the left side of plots B, C, and D of Fig. 78 is transparent. All plots of Fig. 78 have the same size in pixel.

The eighth script calculates the volumes and the compositions of the four blocks of the cooling system of the stationary reflector. These blocks consist of homogenized aluminum and water materials and contain a pure boron carbide layer. The design of the blocks is illustrated in Figs. 79 to 85. In the MCNP model, the boron carbide layers have not been homogenized in order to keep their function of absorbing thermal neutrons and they are explicitly modeled next to the boundary of the stationary reflector. In the blocks design, there is a thin pure aluminum layer between the boron carbide layer and the stationary reflector. This thin pure aluminum layer has not been explicitly modeled. However, the aluminum material of this thin layer has been redistributed in the blocks during the homogenization process, so that the total aluminum mass of the design is preserved in the MCNP model.

The volumes of aluminum and water were calculated by using the colors areas ratios from the left plots of Figs. 83 to 85, these plots are along sections parallel to the z-axis. The volumes of aluminum and water were calculated by multiplying the areas previously discussed by the length along the perpendicular direction. The latter is orthogonal to the z-axis. For instance, the length of the central block shown in Figs. 46 and 54 is 44 cm, as shown in the central plot of Fig. 83. In the design, each block has, at the beginning and end, two pure aluminum layers with thicknesses of 0.4 and 0.6 cm, as shown in the central plot of Figs. 83 and 84 and in the right plot of Fig 85. In the MCNP model, these pure aluminum layers were homogenized with water. The boron carbide volume has been calculated using the design geometry dimensions instead of the colors areas ratios.

For each block, aluminum and water volumes have been corrected by a factor that takes into account the design geometry of the cooling blocks, as shown in Figs. 80 to 82. For instance, for the central block shown in Figs. 46 and 54, the factor was calculated as the sum of all green areas divided the sum of all green and red areas of Fig. 80 plots. A similar procedure has been followed for the other three blocks.

For each block, the volume of the homogenized region is the sum of aluminum and water volumes. For the central block shown in Figs. 46 and 54, the x and the z dimensions have been taken from the central and left plots of Fig. 83, respectively. The y dimension, orthogonal to the boron carbide layer and parallel to the rotating shaft axis, was obtained by dividing the total volume, including water and aluminum volumes, by the x and z dimensions. A similar procedure has been followed for the other three blocks.

In the design, the top-right and the bottom-right plots of Fig. 79 refer to the same cooling block, as shown in the left plot of Fig. 85. In the MCNP model, this cooling block has been divided into two pieces separated by a plane orthogonal to the z-axis, as shown in Fig. 58 left.

4. MCNP and SERPENT Test Simulations

A comparison between MCNP (version 6.1.0) and SERPENT (version 2.1.23)[26] computer programs was performed for the MCNP model discussed in Section 2. This input has all materials at 300 K temperature and homogenized steel-sodium zones at the top and bottom ends of the fuel assemblies. No homogenization was applied in the central part of the fuel assemblies. The two Monte Carlo codes share the same geometrical model and the material specifications using ENDF/B-VII.0 nuclear data[27].

The SERPENT computer program can use either the MCNP model based on spheres, cylinders, prisms, etc., and combinatorial geometry, or the STereoLithographic (STL) geometry format used for 3D printers[28]. The STL geometry describes volumes by triangular facets (meshed surfaces). The CUBIT computer program generated the STL geometry input file used in the SERPENT calculation. The feature angle parameter can be defined in CUBIT, since release 14.1, before exporting the solid model into the STL format. This parameter represents the allowed maximum angle between two successive segments approximating a curved line. The accuracy of the STL geometry model relative to the solid model geometry increases as the value of the feature angle parameter decreases.

CUBIT is internally interfaced with the PYTHON programming language[29]. The PYTHON scripts listed in Appendix E have been used in CUBIT to write the color and volume indexes for each volume of the STL model. The CUBIT model has been constructed using different colors for each volume of the geometry according to the filling material. The C program listed in Appendix F has been used to read the output of the PYTHON scripts and map the colors of the CUBIT model bodies (volumes) into SERPENT materials. The SERPENT input consists of 4820 lines; 4520 of them were written by the C program listed in Appendix F.

All SERPENT computations were performed on a cluster with 32 cores per node. Each core has an Intel Xeon E5-4610 CPU with 2.3 GHz speed. SERPENT was compiled with Intel Fortran compiler[30] version 12.0.2 and MPICH[31] version 3.0.4. The operating system is Linux CentOS[32] release 5.11 and the system RAM memory is 128 GB.

The MCNP computation was performed on a cluster with 8 cores per node. Each core consists of an AMD 6273 CPU with 2.3 GHz speed. MCNP was compiled with Intel Fortran compiler version 11.1 and Intel MPI version 4.0.0.027. The operating system is Red Hat Enterprise[33] release 4 and the system RAM memory per computing node is 32 GB.

MCNP and SERPENT simulations used the MPI and the OPENMP[34] parallel computing platforms. MPI MCNP calculation used 103 computing cores instead of 104, since the master core processed the data for the slave cores without performing any neutron transport simulation. In the MPI SERPENT calculations, the master core, in addition to the slave cores, is used to perform neutron transport. Generally, MCNP and SERPENT have better performances using MPI and OPENMP parallel computing platforms, respectively[35].

The combinatorial geometrical model requires less RAM memory relative to the STL geometrical model. Consequently, MCNP computations were performed with 16 million neutrons per kcode cycle, which reduces standard deviation of the effective multiplication factor (k_{eff}) down to 1 pcm. SERPENT simulations with more than 0.4 million neutrons per kcode cycle can only performed with the OPENMP parallel computing platform due to the RAM memory limitation. In OPENMP there is a single copy of the simulation data for all cores (shared memory feature) of the node, whereas in MPI each core creates its own copy of the simulation data (distributed memory feature).

The results summarized in Table III show that the agreement between MCNP and SERPENT k_{eff} results is within 50 pcm. The SERPENT simulation with the STL feature angle equal to 2 took longer than 20 days; this long computing time is due to the increase of the STL file size. For the feature angle parameter equal to 2, SERPENT simulations could only use the OPENMP parallel computing platform due to the RAM memory limitation previously discussed. The SERPENT simulation with the STL feature angle equal to 10 did not accurately reproduce the MCNP k_{eff}. The SERPENT simulation with the STL feature angle equal to 5 took about one day and produced a k_{eff} that agrees very well with the MCNP result, within 20 pcm.

5. Conclusions

This report describes the modeling of the IBR-2M reactor by MCNP and SERPENT Monte Carlo computer programs. The MCNP geometry is based on basic volumes and combinatorial geometry, whereas the SERPENT geometry is based on triangular facets (meshed surfaces) using the STereoLithographic (STL) format of 3D printers. The MCNP model took advantage of GIMP and MATLAB to describe the complicated design shapes, e.g. the serpentines and the irregular profiles of the cooling system surfaces. The GIMP image editor has been used to calculate the area of each color in the design drawings, where each color represents a specific material. GIMP estimates the areas covered by three colors, red, green, and blue, of a drawing. Therefore, this approach can only be used to account up to three different materials. Additional materials can be considered by using additional images with different color schemes. In addition, the GIMP editor can calculate the centroid of a colored zone, which can be used to homogenize materials confined in solid of revolution (solids generated by rotating a surface around some straight line). The SERPENT model has been created using CUBIT, MATLAB, PYTHON, and C computer programs. The CUBIT and MATLAB programs were used to replicate the MCNP geometry model, whereas the PYTHON and the C languages were used to map the colors of the CUBIT volumes into SERPENT materials.

The number of handwritten input lines is 1500 and 1750 for the MCNP and SERPENT models, respectively. The effective multiplication factors k_{eff} from these two Monte Carlo codes agree within 50 pcm. The MCNP simulation requires 17 hours to get 1 pcm standard deviation. The SERPENT simulation with CUBIT STL feature angle parameter equal to 2 requires 20 days to get 5 pcm standard deviation. If the feature angle is increased to 5, then the computing time of the SERPENT simulations is only 30 hours. In this case, the k_{eff} standard deviation increases to 11 pcm and the k_{eff} value is within 100 pcm from the MCNP value. A CUBIT STL feature angle parameter equal to 10 produces inaccurate SERPENT results. MCNP and SERPENT simulations have been performed on different computer clusters, therefore it is not possible to compare their performances. An accurate comparison of the computing time of these two programs can be found in other studies[36].

ACKNOWLEDGMENTS

This work is supported by the U.S. Department of Energy, Office of Material Management and Minimization (M3), National Nuclear Security Administration.

References

[1] JOINT INSTITUTE FOR NUCLEAR RESEARCH International Intergovernmental Scientific Research Organization, Technical description of the IBR-2M Reactor, Dubna, Russia, 2014.

[2] JOINT INSTITUTE FOR NUCLEAR RESEARCH International Intergovernmental Scientific Research Organization, Sistematization of data on PuO_2 fuel used at the IBR-2M research reactor, Dubna, Russia, 2015.

[3] JOINT INSTITUTE FOR NUCLEAR RESEARCH International Intergovernmental Scientific Research Organization, Evaluation of LEU fuel based on its neutron-physical characteristics for use in the IBR-2M, Dubna, Russia, 2014.

[4] Y.N. PEPYOLYSHEV, A.K. POPOV, and D. SUMKHUU, Model of the IBR-2M pulsed reactor dynamics for investigating transition processes in a wide range of power variation, Annals of Nuclear Energy **85**, pp. 488-493, 2015.

[5] Y.N. PEPELYSHEV and T.S. TSOGTSAIKHAN, Investigation of the pulse energy noise dynamics of IBR-2M using cluster analysis, Annals of Nuclear Energy **83**, pp. 50-56, 2015.

[6] Y.N. PEPYOLYSHEV and A.K. POPOV, Safe external reactivity perturbation limits in IBR-2M, Annals of Nuclear Energy **38/5**, pp. 1109-1114, 2011.

[7] A.A. MARACHEV, Y.N. PEPYOLYSHEV, and A.K. POPOV, Statistically optimal filter for IBR-2M mean power estimation, Annals of Nuclear Energy **35/9**, pp. 1779-1784, 2008.

[8] Y.N. PEPYOLYSHEV, Method of experimental estimation of the effective delayed neutron fraction and of the neutron generation lifetime in the IBR-2M pulsed reactor, Annals of Nuclear Energy **35/7**, pp. 1301-1305, 2008.

[9] Y.N. PEPELYSHEV and A.K. POPOV, Influence of the nearest environment of the core on the power pulse dynamics in the IBR-2M reactor, Annals of Nuclear Energy **33/9**, pp. 813-819, 2006.

[10] W. DZWINEL, Y.N. PEPYOLYSHEV, and K. JANICZAK, Predicting of slow noise and vibration spectra degradation in the IBR-2M pulsed neutron source using a neural network simulator, Progress in Nuclear Energy **43**, pp. 145-150, 2003.

[11] A.K. POPOV, Y.N. PEPYOLYSHEV , and E.A. BONDARCHENKO, The model of the IBR-2M pulsed reactor dynamics and investigation of pulse energy stabilization, Nuclear Technology **139/1**, pp. 21-29, 2002.

[12] E.A. BONDARCHENKO, Y.N. PEPYOLYSHEV, and A.K. POPOV, Influence of automatic regulator parameters on power transition processes of the IBR-2M reactor, Annals of Nuclear Energy **28/1**, pp. 63-78, 2001.

[13] A.K. POPOV, YU.N. PEPELYSHEV, and E.A. BONDARCHENKO, The model of the IBR-2M pulsed reactor of periodic operation for investigations of transitional processes, Annals of Nuclear Energy **27/7**, pp. 563-574, 2000.

[14] Y.G. DRAGUNOV et al., Modernization of the IBR-2M Pulsed Research Reactor, Atomic Energy **113/1**, 2012.

[15] Y.N. PEPELYSHEV and A.K. POPOV, Neutron physics calculations for upgraded IBR-2M reactor during the first criticality and power reactor startup, Proceedings of the XXIII Inter-Industry Seminar, Neutron Physics Problems of Nuclear Power

Engineering with Closed Fuel Cycle, Obninsk, Russia, October 30 to November 2, 2012.
[16] V.D. ANAN'EV et al., The first criticality of modernized IBR-2 (IBR-2M) reactor, International Intergovernmental Scientific Research Organization JOINT INSTITUTE FOR NUCLEAR RESEARCH, P13-2012-41, Dubna, Russia, 2012.
[17] D.B. PELOWITZ et al., MCNP6 user's manual, code version 6.1.1beta, Los Alamos National Laboratory, LA-CP-14-00745, 2014.
[18] M.B. CHADWICK et al., ENDF/B-VII.1 nuclear data for science and technology: cross sections, covariances, fission product yields and decay data, Nuclear Data Sheets **112/12**, 2011.
[19] VISUALBASIC Macro, URL https://support.office.com/en-us/article/Automate-tasks-with-Visual-Basic-macros-925942c5-4961-4af7-a0c0-08b4a4db9e97,2015.
[20] RCS Revision Control System, URL https://www.gnu.org/software/rcs/, 2015.
[21] MATLAB, URL http://www.mathworks.com/products/matlab/, 2015.
[22] CUBIT, URL https://cubit.sandia.gov/, 2015.
[23] GIMP, URL http://www.gimp.org/, 2015.
[24] GIMP Centroids Plugin, URL http://registry.gimp.org/node/27529, 2015.
[25] GIMP Export Layers Plugin, URL https://github.com/khalim19/gimp-plugin-export-layers, 2015.
[26] J. LEPPÄNEN et al., The Serpent Monte Carlo Code: Status, Development and Applications in 2013, Annals of Nuclear Energy **82**, pp. 142-150, 2015.
[27] M.B. CHADWICK et al., ENDF/B-VII.0: Next Generation Evaluated Nuclear Data Library for Nuclear Science and Technology, Nuclear Data Sheets **107/12**, pp. 2931-3060, 2006.
[28] STL File Format, URL https://en.wikipedia.org/wiki/STL_(file_format), 2015.
[29] PYTHON Programmin Language, URL https://www.python.org/, 2015.
[30] INTEL FORTRAN Compiler, URL https://software.intel.com/en-us/fortran-compilers, 2015.
[31] MPICH Parallel Computing Platform, URL https://www.mpich.org/, 2015.
[32] LINUX CENTOS, URL https://www.centos.org/, 2015.
[33] RED HAT ENTERPRISE, URL http://www.redhat.com/en/technologies/linux-platforms/enterprise-linux, 2015.
[34] OPENMP Parallel Computing Platform, URL http://openmp.org/wp/, 2015.
[35] A TALAMO et al., SERPENT Stereolithography Modeling of the GIACINT Critical Assembly Experimental Facility, PHYSOR Conference Proceedings, Sun Valley, ID, USA, 2016.
[36] A TALAMO et al., SERPENT Validation and Optimization with Mesh Adaptive Search on Stereolithography Geometry Models, Annals of Nuclear Energy **115**, pp. 619-632, 2018.

Table I: Geometry specifications of the IBR-2M reactor model; dimensions are in cm

Fuel Rod	
Radii active zone (materials) (Fig. 48)	0.075 (He); 0.37 (PuO_2); 0.385 (He); 0.4373 (Steel)
Total and active lengths (Fig. 18 bottom)	77.8; 44.4
Tungsten zone length (Fig. 13 bottom)	6
Spring zone length (Fig. 13 bottom)	22.3
Conical transition at top length (Fig. 14 top)	0.2
Reduced radius zone at top length (Fig. 14 top)	0.125
Inactive length at top (Fig. 13 bottom)	29.5 = [6 (tungsten; cylindrical shape) + 22.3 (reduced density steel; cylindrical shape) + 0.2 (steel; conical shape) + 1 (steel; cylindrical shape)]
Reduced radius at top (Fig. 14 top)	0.25
Inactive steel zone at bottom length (Fig. 15 top)	0.8
Conical zone at bottom length (Fig. 15 top)	0.3
Reduced radius zone at bottom length (Fig. 15 top)	0.5
Conical zone at bottom (next to the steel and sodium zone) length (Fig. 15 top)	0.5
Steel and sodium zone at bottom length (Fig. 15 top)	0.7
Inactive length at bottom (Fig. 15 top)	2.8 = [0.8 (steel; cylindrical shape) + 0.3 (steel; conical shape) + 0.5 (steel; cylindrical shape) + 0.5 (steel; conical shape) + 0.7 (steel and sodium; cylindrical shape)]
Reduced radius at bottom (Fig. 15 top)	0.15
Fuel Assembly with Fuel Rods	
Displacer radius (Fig. 48)	0.2
Fuel rods pitch (Fig. 48)	0.911
Apothems (materials) (Fig. 48)	1.27 (sodium/fuel rods); 1.31 (steel); 1.36 (sodium)
Length (Figs. 3 to 5)	77.8
Fuel Assembly above the Fuel Rods	
Hollow cylinders radii (materials) (Figs. 43 and 13 bottom)	0.8913 (sodium); 1.2414 (steel)
Hollow cylinders length (Figs. 43 and 13 bottom)	11.891
Steel cylinders radius (Figs. 44 top and 13 bottom)	0.36694
Steel cylinders length (Figs. 44 top and 13 bottom)	3.109

Total length (Fig. 13 bottom)	15 = [11.891 (steel; hollow cylinders; Fig. 43) + 3.109 (steel; solid cylinders; Fig. 44 top)]
Fuel Assembly below the Fuel Rods and Support Grid	
Hollow cylinders radii (materials) (Figs. 36 bottom and 19)	0.7797 (sodium); 1.3255 (steel)
Hollow cylinders length (Figs. 36 bottom and 19)	3.274
Sodium channels radius (Figs. 36 top and 19)	0.93564
Sodium channels length (Figs. 36 top and 19)	24.8
Sodium channels radius (Figs. 35 bottom and 19)	1.125
Sodium channels length (Figs. 35 bottom and 19)	3.2
Total length (Fig. 19)	31.274 = [3.274 (steel; coolant channels; Fig. 35 bottom) + 24.8 (steel; support grid and coolant channels; Fig. 36 top) + 3.2 (steel; support grid; Fig. 36 bottom)]
Vessel	
Apothems in the hexagonal zone (direction) (Fig. 47)	11.275 (north and south); 11.277 (south-west); 13.514 (north-west)
Thicknesses (materials) in all zones [mm] (Fig. 47)	0.7 (steel); 0.3 (air); 0.4 (steel)
Length hexagonal zone (z extension) (Fig. 30 top)	116 (from -60 to 56)
Transition from hexagonal to top cylindrical zone length (Fig. 30 top)	10
Transition increasing the radius of the cylindrical zone length (Fig. 30 top)	7.6
Transition from hexagonal to bottom cylindrical zone length (Fig. 30 top)	9.5
Cylindrical zone inner radii (z extension) (Figs. 29 bottom and 30 top)	5.7 (from -400 to -69.5) 15.2 (from 66 to 92) 17.25 (from 99.6 to 296)
Total length (Fig. 29 bottom and Fig. 50)	696 = [330.5 (bottom; cylindrical shape) + 9.5 (transition from cylindrical to hexagonal shape) + 116 (hexagonal shape) + 10 (transition from hexagonal to cylindrical shape) + 26 (cylindical shape) + 7.6 (conical shape) + 196.4 (top; cylindrical shape)]
z extension of the steel solid zone inside the vessel (Figs. 30 top and 44 bottom)	from 69.8 to 82
Stationary Reflector and Control Rods	
Length (z extension) (Fig. 6)	100 (from -74 to 26)

XY dimensions	Fig. 67
SCRAM control rods length (Fig. 18 top)	42.6
Compensation control rods length (Fig. 20)	42
Manual control rods length (Fig. 21 bottom)	42
Automatic control rod (beryllium) length (Fig.21 bottom)	40
Control rods support length (Fig. 21 bottom)	38
Water Cooling System of the Stationary Reflector (central block in Figs. 46 and 54)	
Boron carbide thickness	1
Boron carbide length	21.6
Boron carbide height (z extension)	44 (from -22 to 22)
H_2O and Al width	12.97
H_2O and Al length	44
H_2O and Al height (z extension)	49 (from -24.5 to 24.5)
Water Cooling System of the Stationary Reflector (right block in Figs. 46 and 54)	
Boron carbide thickness	1
Boron carbide length	25.8
Boron carbide height (z extension)	44 (from -22 to 22)
H_2O and Al width	10.156
H_2O and Al length	29.4
H_2O and Al height (z extension)	46.6 (from -23.3 to 23.3)
Water Cooling System of the Stationary Reflector (top-left block in Figs. 39 and 58 left)	
Boron carbide thickness	1
Boron carbide length	25.8
Boron carbide height (z extension)	44 (from -22 to 22)
H_2O and Al width	6.7251
H_2O and Al length	29.4
H_2O and Al height (z extension)	22.782 (from 0.51778 to 23.3)
Water Cooling System of the Stationary Reflector (bottom-left block in Figs. 46 and 58 left)	
Boron carbide thickness	1
Boron carbide length	25.8
Boron carbide height (z extension)	44 (from -22 to 22)
H_2O and Al width	10.946
H_2O and Al length	29.4

H_2O and Al height (z extension)	46.6 (from -23.3 to 0.51778)
Rotating Reflectors	
Center of the rotating shaft (x,y,z) (Fig. 46 and Fig. 24 bottom)	-45, 19.505, -92.26
Rotating shaft radius, y length (Figs. 26 and 1)	26, 0.48 + 5 + 0.8 + 5 + 0.48
Main rotating reflector y length (y extension) (Figs. 46 and 4)	5 (from 14.105 to 19.105)
Auxiliary rotating reflector y length (y extension) (Fig. 14 bottom)	5 (from 19.905 to 24.905)
Inner and outer radii of the steel cylindrical shell (Figs. 24 bottom and 70 to 74)	121.26; 130
Thicknesses (materials) (y extension) of the inner jacket (Figs. 46 top and 75)	1.2697 (steel) (from 8.612 to 9.8817); 1.7433 (water) (from 9.8817 to 11.625); 2 (steel) (from 11.625 to 11.625)
Thicknesses (materials) (y extension) of the outer jacket (Figs. 46 top and 75)	1.5 (steel) (from 25.385 to 26.885); 1.796 (water) (from 26.885 to 28.681); 1.728 (steel) (from 28.681 to 30.409)
Cut 1 of the inner jacket x, y, x extensions relative to the center of the rotating axis (Figs. 22 bottom, 23 top, 70 top, and 73)	from 0 to 92.95 from 8.612 to 11.625 from 0 to 42.25
Cut 2 of the inner jacket x, y, x extensions relative to the center of the rotating axis (Figs. 22 bottom, 23 top, 70 top, and 73)	from 0 to 131 from 8.612 to 11.625 from 42.25 to 131.26
Cut 3 of the inner jacket x, y, x extensions relative to the center of the rotating axis (Figs. 22 bottom, 23 top, 70 top, and 73)	from 0 to 54.6 from 8.612 to 11.625 from -22.1 to 0
Cut 4 of the inner jacket x, y, x extensions relative to the center of the rotating axis (Fig. 23 bottom and 70 top)	from 32.2 to 57.8 from 11.625 to 13.375 from 18.26 to 131.26
Cut of the outer jacket x, y, x extensions relative to the center of the rotating axis (Figs. 27 bottom, 28, 70 bottom, and 74)	from 12.062 to 62.543 from 25.985 to 30.409 from 50.928 to 131.26
XZ dimensions of the main rotating reflector (Fig. 25 top)	Fig. 68
XZ dimensions of the auxiliary rotating reflector (Fig. 26 top)	Fig. 69

Water Cooling System of the Rotating Reflectors	
Aluminum inner layer x, y, z extensions (Figs. 16 top and 46 bottom)	from -15 to 15 from 26.885 to 27.385 from -22.5 to 22.5
Boron carbide layer x, y, z extensions (Figs. 16 top and 46 bottom)	from -15 to 15 from 27.385 to 28.385 from -22.5 to 22.5
Aluminum central layer x, y, z extensions (Figs. 16 top and 46 bottom)	from -15 to 15 from 28.385 to 28.885 from -22.5 to 22.5
Water layer x, y, z extensions (Figs. 16 top and 46 bottom)	from -15 to 15 from 28.885 to 33.885 from -22.5 to 22.5
Aluminum outer layer x, y, z extensions (Figs. 16 top and 46 bottom)	from -15 to 15 from 33.885 to 34.385 from -22.5 to 22.5
Air channel in the water layer radius (Fig. 46 bottom)	1.75

Table II: Materials specifications of the IBR-2M reactor model; atomic and weight fractions are positive and negative, respectively

ISOTOPES	ATOMIC/WEIGHT FRACTIONS
Plutonium Oxide **active zone of the fuel rods** **Color = Brown – T = 600 K – Density = 10.103 g/cm^3**	
^{238}Pu	0.00002241
^{239}Pu	0.02135588
^{240}Pu	0.00089566
^{241}Pu	0.00003763
^{242}Pu	0.00000220
^{16}O	0.04559839
Steel **fuel rods clad, non-active zone of the fuel rods, fuel assemblies, vessel, stationary reflector, rotating reflectors jackets and cylindrical shell, rotating shaft, blade-holders** **Color = Blue – T = 293/600 K – Density = 0.08495329 atoms/cm^3**	
C	6.79625E-5
^{28}Si	6.268190598E-4
^{29}Si	3.17385342E-5
^{30}Si	2.1068406E-5
^{31}P	2.97336E-5
^{32}S	1.6143898E-5
^{33}S	1.27425E-7
^{34}S	7.15279E-7
^{36}S	3.398E-9
^{46}Ti	3.7379408E-5
^{47}Ti	3.41087098E-5
^{48}Ti	3.448250388E-4
^{49}Ti	2.5698343E-5
^{50}Ti	2.52311004E-5
^{50}Cr	0.000664419
^{52}Cr	0.0128127
^{53}Cr	0.00145285
^{54}Cr	0.000361646
^{55}Mn	0.001699064
^{54}Fe	0.00337593
^{56}Fe with $S(\alpha,\beta)$	0.0533863
^{57}Fe	0.00128053
^{58}Fe	0.000162976
^{58}Ni	0.00578336
^{60}Ni	0.00222773
^{61}Ni	9.68467E-5
^{62}Ni	0.00030872
^{64}Ni	7.86667E-5

Steel **spring zone inside the fuel rods** **Color = DeepSkyBlue (CUBIT) Blue (MCNP) – T = 600 K – Density = 0.64 g/cm^3**	
Steel composition	
Tungsten Alloy **inside the fuel rods** **Color = DarkSlateGray – T = 600 K – Density = 0.06890289 atoms/cm^3**	
^{54}Fe	0.00022935
^{56}Fe	0.00460347
^{57}Fe	0.00053185
^{58}Fe	0.00013475
^{58}Ni	0.00822984
^{60}Ni	0.00325456
^{61}Ni	0.00014288
^{62}Ni	0.00046162
^{64}Ni	0.00121023
^{182}W	0.01327765
^{183}W	0.00716993
^{184}W	0.01541209
^{186}W	0.01424466
Tungsten Alloy **control rods inside the stationary reflector** **Color = Pink – T = 293 K – Density = 0.05999262 atoms/cm^3**	
^{50}Cr	0.00010308
^{52}Cr	0.00206893
^{53}Cr	0.00023903
^{54}Cr	0.00006056
^{54}Fe	0.00054564
^{56}Fe with $S(\alpha,\beta)$	0.01095209
^{57}Fe	0.00126531
^{58}Fe	0.00032058
^{58}Ni	0.00635118
^{60}Ni	0.00251163
^{61}Ni	0.00011027
^{62}Ni	0.00035625
^{64}Ni	0.00093397
^{182}W	0.00905614
^{183}W	0.00489031
^{184}W	0.01051196
^{186}W	0.00971570
Steel **support of the control rods** **Color = DarkSeaGreen – T = 293 K – Density = 0.00726794 atoms/cm^3**	
^{50}Cr	0.00005764
^{52}Cr	0.00115697
^{53}Cr	0.00013367

^{54}Cr	0.00003387
^{54}Fe	0.00021765
^{56}Fe with S(α,β)	0.00436869
^{57}Fe	0.00050472
^{58}Fe	0.00012788
^{58}Ni	0.00041267
^{60}Ni	0.00016319
^{61}Ni	0.00000716
^{62}Ni	0.00002315
^{64}Ni	0.00006068
Iron and Boron Carbide inside the top part of the vessel Color = PaleGreen – T = 293 K – Density = 0.07453 atoms/cm^3	
^{10}B	0.00005764
^{11}B	0.00115697
C	0.00013367
^{54}Fe with S(α,β)	0.00003387
Nickel Alloy blades of the rotating reflectors Color = Green – T = 293 K – Density = 0.0915681 atoms/cm^3	
^{10}B	1.823628E-6
^{27}Al	0.0007331168
^{28}Si	5.0711512625E-4
^{29}Si	2.567741125E-5
^{30}Si	1.70449625E-5
^{46}Ti	1.9794152E-4
^{47}Ti	1.80621637E-4
^{48}Ti	1.826010522E-3
^{49}Ti	1.36084795E-4
^{50}Ti	1.33610526E-4
^{50}Cr	0.000816257
^{52}Cr	0.0157407
^{53}Cr	0.00178487
^{54}Cr	0.000444292
^{55}Mn	5.3151E-5
^{54}Fe	0.000840518
^{56}Fe	2.01607E-5
^{57}Fe	2.56591E-6
^{58}Fe	0.0003665584
^{58}Ni	0.0461153
^{60}Ni	0.0177635
^{61}Ni	0.000772236
^{62}Ni	0.00246167
^{64}Ni	0.000627272

Boron Carbide **cooling system of the stationary and rotating reflectors** **Color = Red – T = 293 K – Density = 0.0926375 atoms/cm^3**	
^{10}B	0.0146590
^{11}B	0.0594510
C	0.0185275
Sodium **inside the vessel** **Color = Yellow – T = 600 K – Density = 0.873994792 g/cm^3**	
^{23}Na	1
Helium **central hole and gap of the fuel rods and inside the rotating reflectors casing** **Color = LightCyan – T = 293/600 K – Density = 0.1664 mg/cm^3**	
^{3}He	0.000137
^{4}He	99.999863
Air **gap in-between the steel layers of the vessel** **Color = LightYellow – T = 293 K – Density = 1.29 mg/cm^3**	
^{7}N	-0.7885
^{16}O	-0.2115
Beryllium **automatic control rod in the stationary reflector** **Color = Magenta – T = 293 K – Density = 1.85 mg/cm^3**	
^{9}Be with $S(\alpha,\beta)$	1
Steel and Sodium **at the bottom end of the fuel rods** **Color = DarkOliveGreen – T = 600 K – Density = 0.0645651 atoms/cm^3**	
C	6.79625E-5
^{28}Si	6.268190598E-4
^{29}Si	3.17385342E-5
^{30}Si	2.1068406E-5
^{31}P	2.97336E-5
^{32}S	1.6143898E-5
^{33}S	1.27425E-7
^{34}S	7.15279E-7
^{36}S	3.398E-9
^{46}Ti	3.7379408E-5
^{47}Ti	3.41087098E-5
^{48}Ti	3.448250388E-4
^{49}Ti	2.5698343E-5
^{50}Ti	2.52311004E-5
^{50}Cr	0.000664419
^{52}Cr	0.0128127
^{53}Cr	0.00145285
^{54}Cr	0.000361646
^{55}Mn	0.001699064

^{54}Fe	0.00337593
^{56}Fe with $S(\alpha,\beta)$	0.0533863
^{57}Fe	0.00128053
^{58}Fe	0.000162976
^{58}Ni	0.00578336
^{60}Ni	0.00222773
^{61}Ni	9.68467E-5
^{62}Ni	0.00030872
^{64}Ni	7.86667E-5
^{23}Na	6.5241E-3
Aluminum and Water cooling system of the stationary reflector, central block in Figs. 46 and 54 Color = SkyBlue – T = 293 K – Density = 0.082113 atoms/cm^3	
^{27}Al with $S(\alpha,\beta)$	4.8502E-1
^{1}H with $S(\alpha,\beta)$	6.6667E-1
^{16}O	3.3333E-1
Aluminum and Water cooling system of the stationary reflector, right block in Figs. 46 and 54 Color = SkyBlue – T = 293 K – Density = 0.083193 atoms/cm^3	
^{27}Al with $S(\alpha,\beta)$	4.3367E-1
^{1}H with $S(\alpha,\beta)$	6.6667E-1
^{16}O	3.3333E-1
Aluminum and Water cooling system of the stationary reflector, top-left block in Figs. 39 and 58 Color = SkyBlue – T = 293 K – Density = 0.080037 atoms/cm^3	
^{27}Al with $S(\alpha,\beta)$	5.9943E-1
^{1}H with $S(\alpha,\beta)$	6.6667E-1
^{16}O	3.3333E-1
Aluminum and Water cooling system of the stationary reflector, bottom-left block in Figs. 46 and 58 Color = SkyBlue – T = 293 K – Density = 0.082038 atoms/cm^3	
^{27}Al with $S(\alpha,\beta)$	4.8876E-1
^{1}H with $S(\alpha,\beta)$	6.6667E-1
^{16}O	3.3333E-1
Aluminum cooling system of the rotating reflectors Color = Purple – T = 293 K – Density = 2.7 g/cm^3	
^{27}Al with $S(\alpha,\beta)$	1
Copper inside the fuel rods Color = Magenta (in the core) – T = 600 K – Density = 8.96 g/cm^3	
^{63}Cu	0.6915
^{65}Cu	0.3085

Water cooling system of the rotating reflectors Color = Cyan – T = 293 K – Density = 0.9977735 g/cm^3	
^{1}H with S(α,β)	-0.1111110
^{16}O	-0.8888890

Table III: Monte Carlo simulation results. The statistical error on k_{eff} results is in pcm units. The simulations were performed with 200 and 40 total and inactive kcode cycles, respectively

Code	**STL Feature Angle**	**STL File Size [Gb]**	k_{eff}	**Cores**	**Parallel Computing Platform**	**Computing Time [h]**	**Neutrons per kcode Cycle**
MCNP	–	–	1.01056±1	103	Intel MPI	17.1	16E6
SERPENT	2	2.6	1.01106±4.8	32	OPENMP	487.9	6.4E6
SERPENT	5	1.3	1.01072±11	32	MPICH	30.2	4E5
SERPENT	10	0.325	1.00641±11	32	MPICH	5.1	4E5

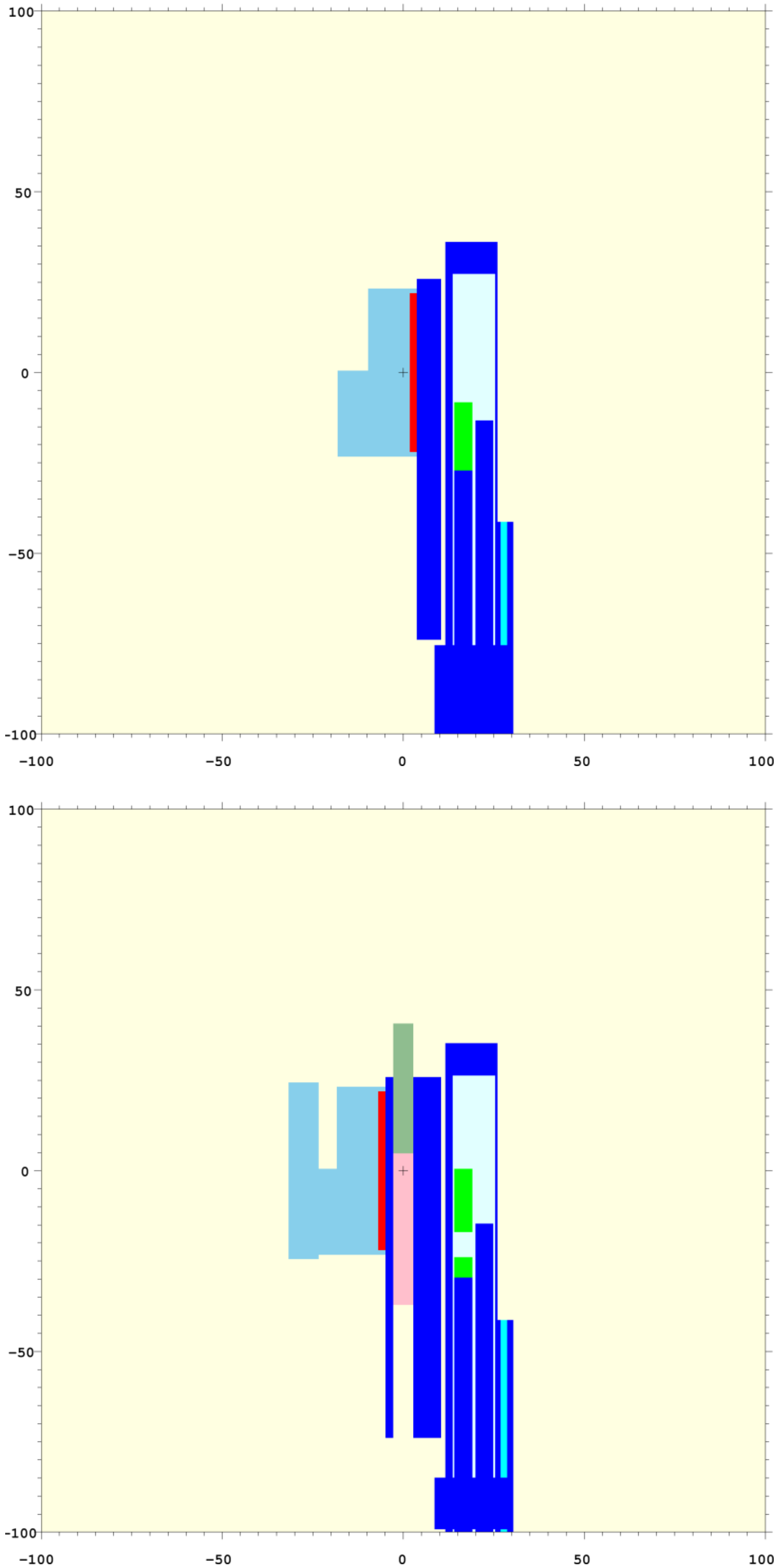


Figure 1. Top plot: px plane with origin at (-25, 0, 0) cm and 100 cm extent; bottom plot: px plane with origin at (-20, 0, 0) cm and 100 cm extent

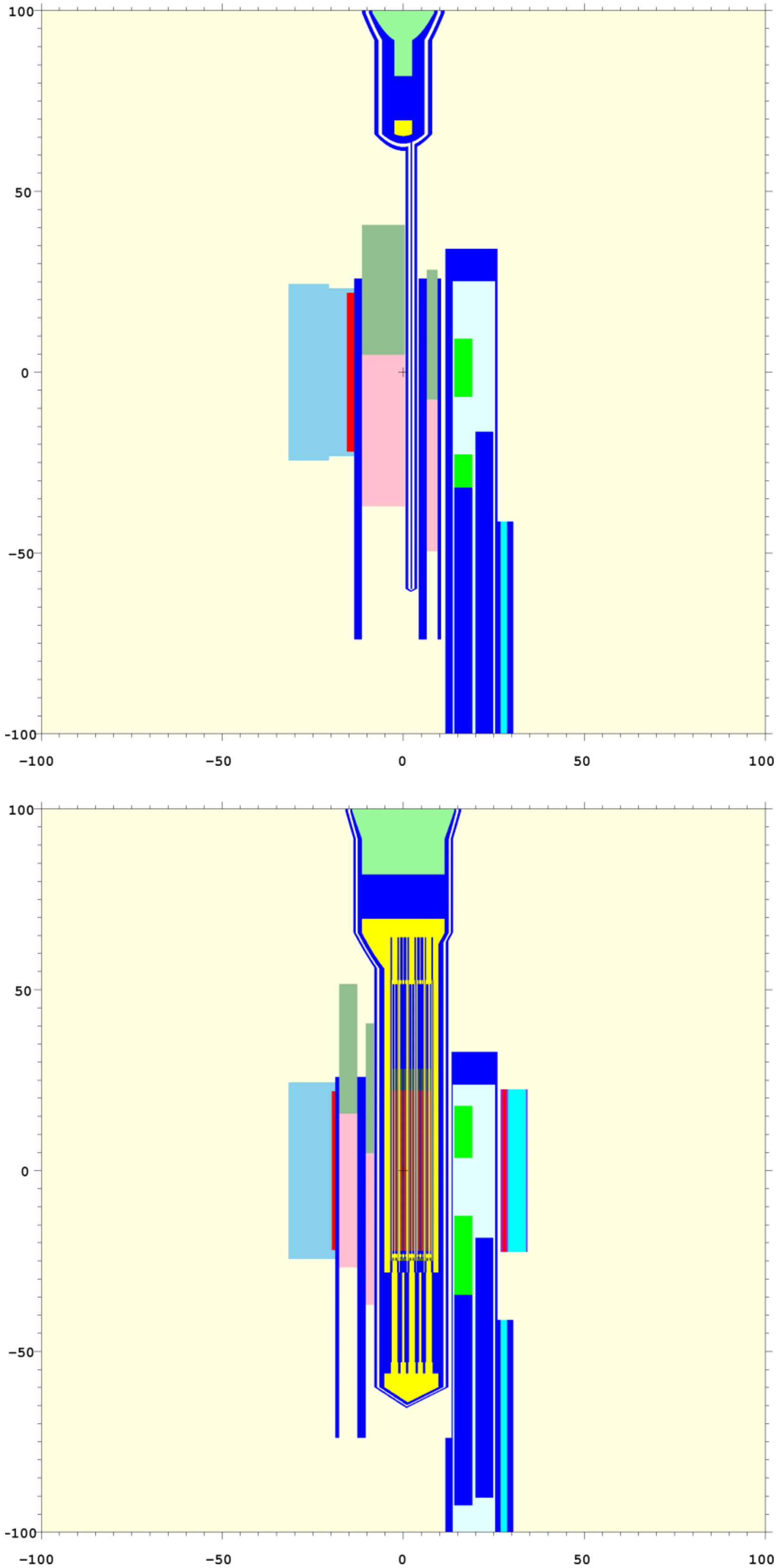


Figure 2. Top plot: px plane with origin at (-15, 0, 0) cm and 100 cm extent; bottom plot: px plane with origin at (-10, 0, 0) cm and 100 cm extent

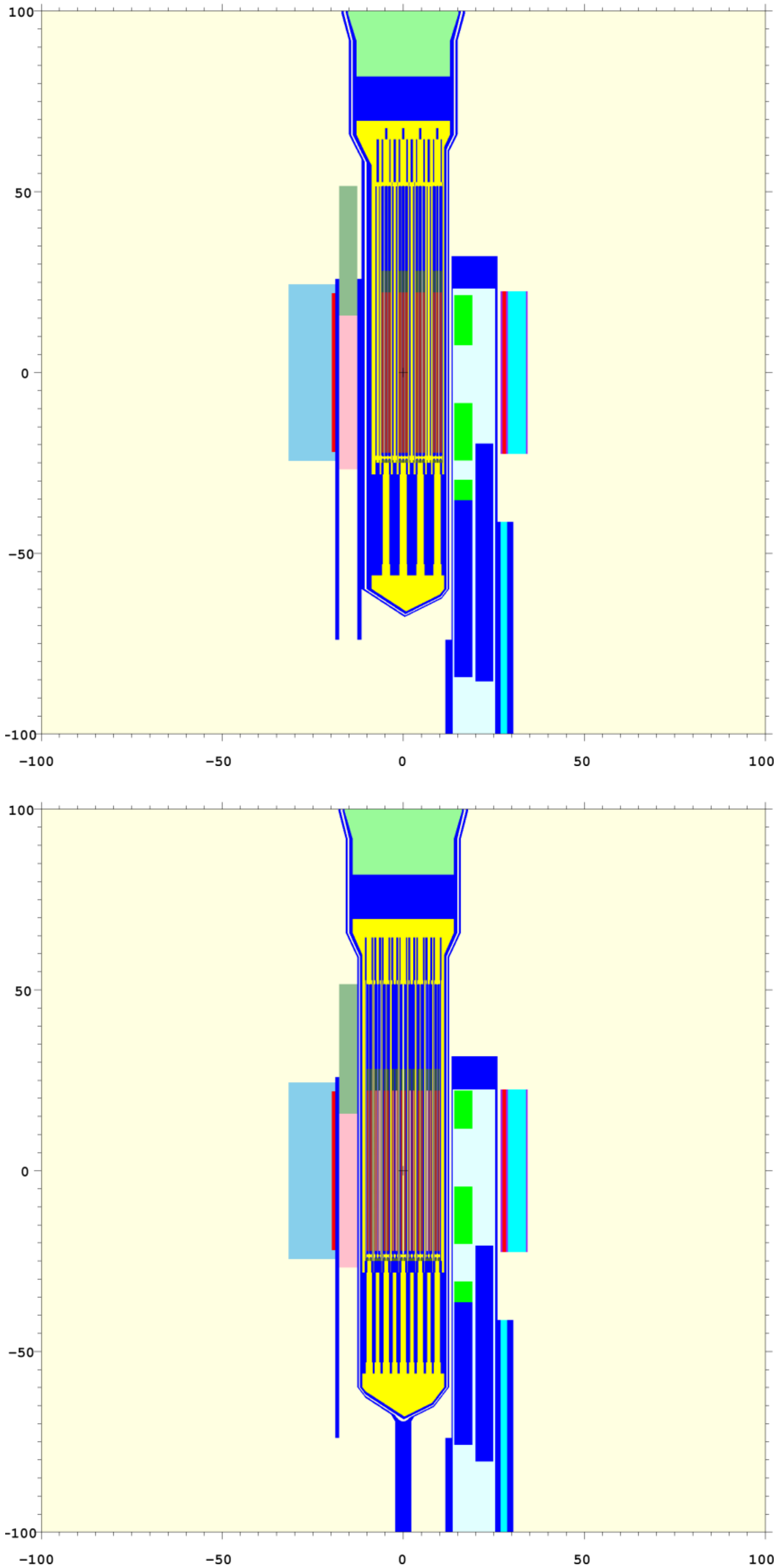


Figure 3. Top plot: px plane with origin at (-8, 0, 0) cm and 100 cm extent; bottom plot: px plane with origin at (-6, 0, 0) cm and 100 cm extent

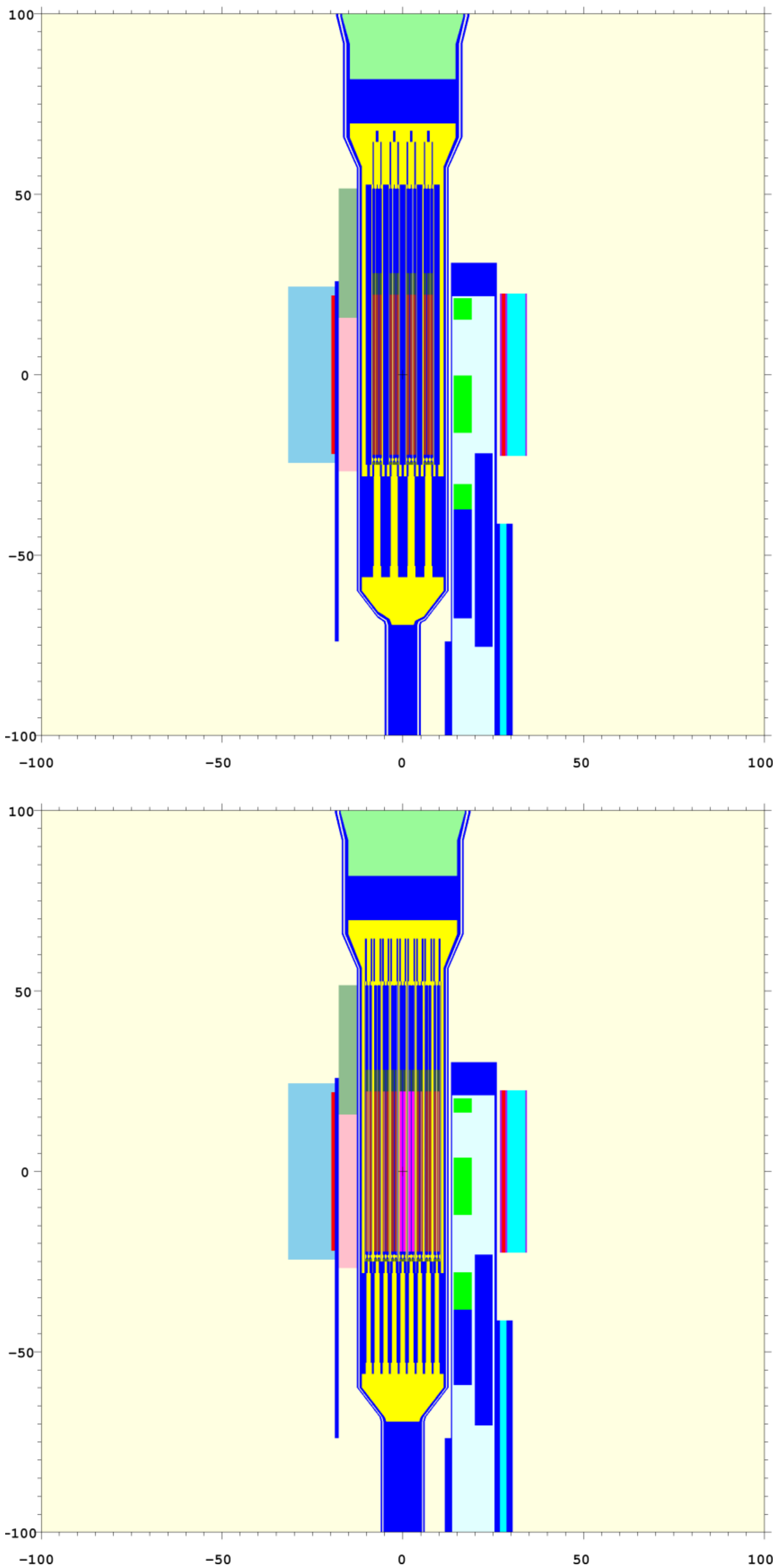


Figure 4. Top plot: px plane with origin at (-4, 0, 0) cm and 100 cm extent; bottom plot: px plane with origin at (-2, 0, 0) cm and 100 cm extent

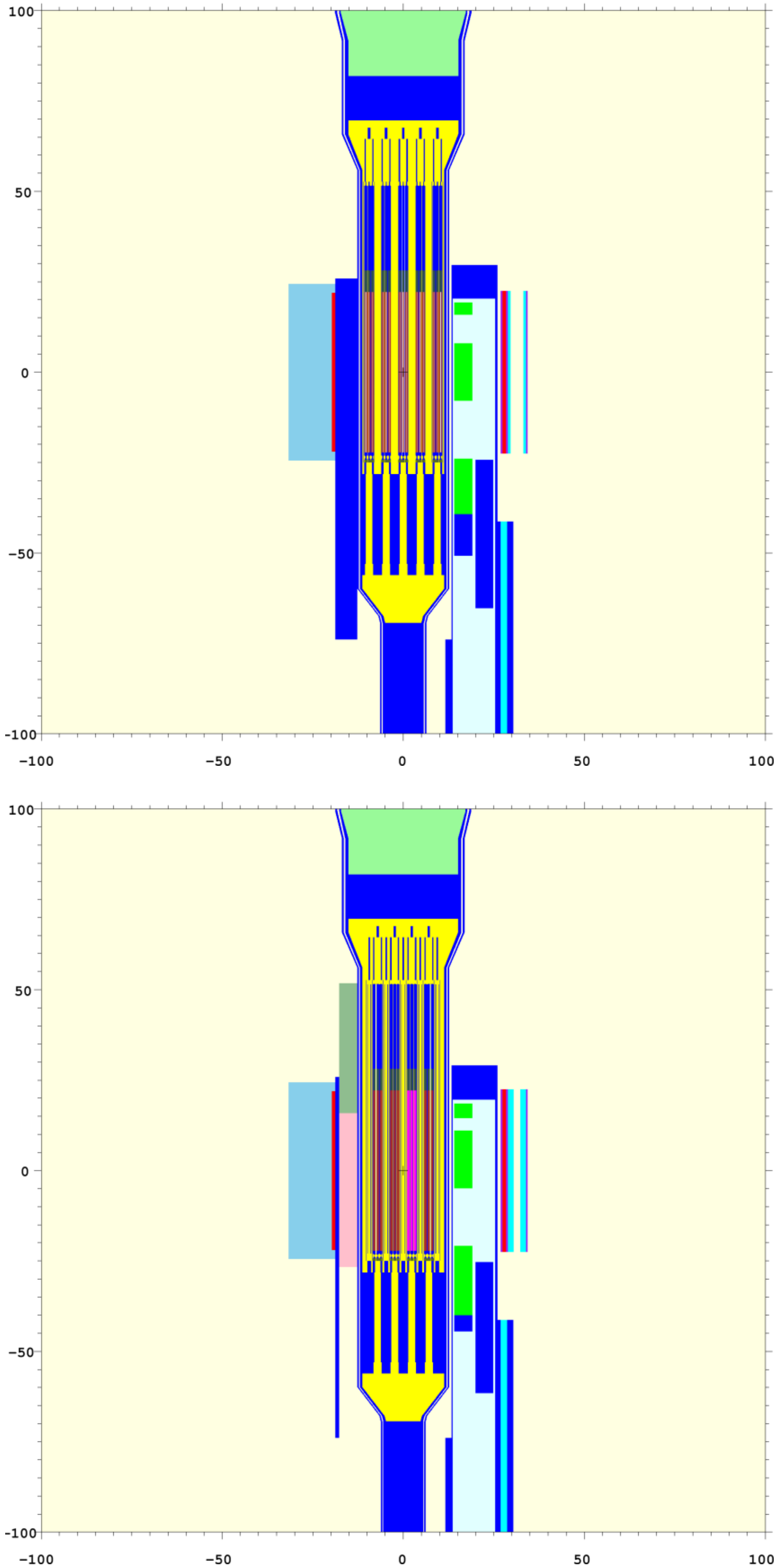


Figure 5. Top plot: px plane with origin at (0, 0, 0) cm and 100 cm extent; bottom plot: px plane with origin at (1.5, 0, 0) cm and 100 cm extent

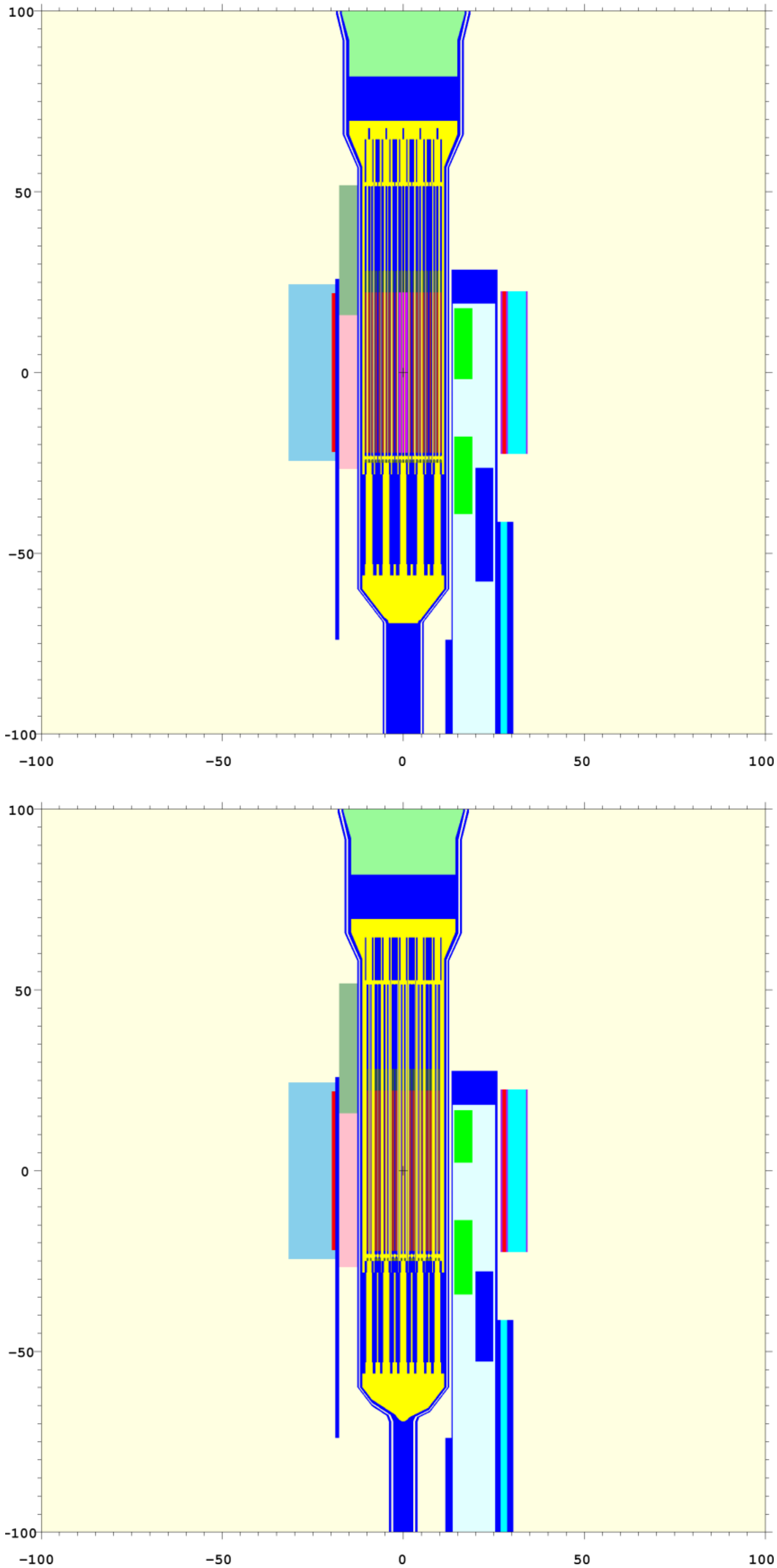


Figure 6. Top plot: px plane with origin at (3, 0, 0) cm and 100 cm extent; bottom plot: px plane with origin at (5, 0, 0) cm and 100 cm extent

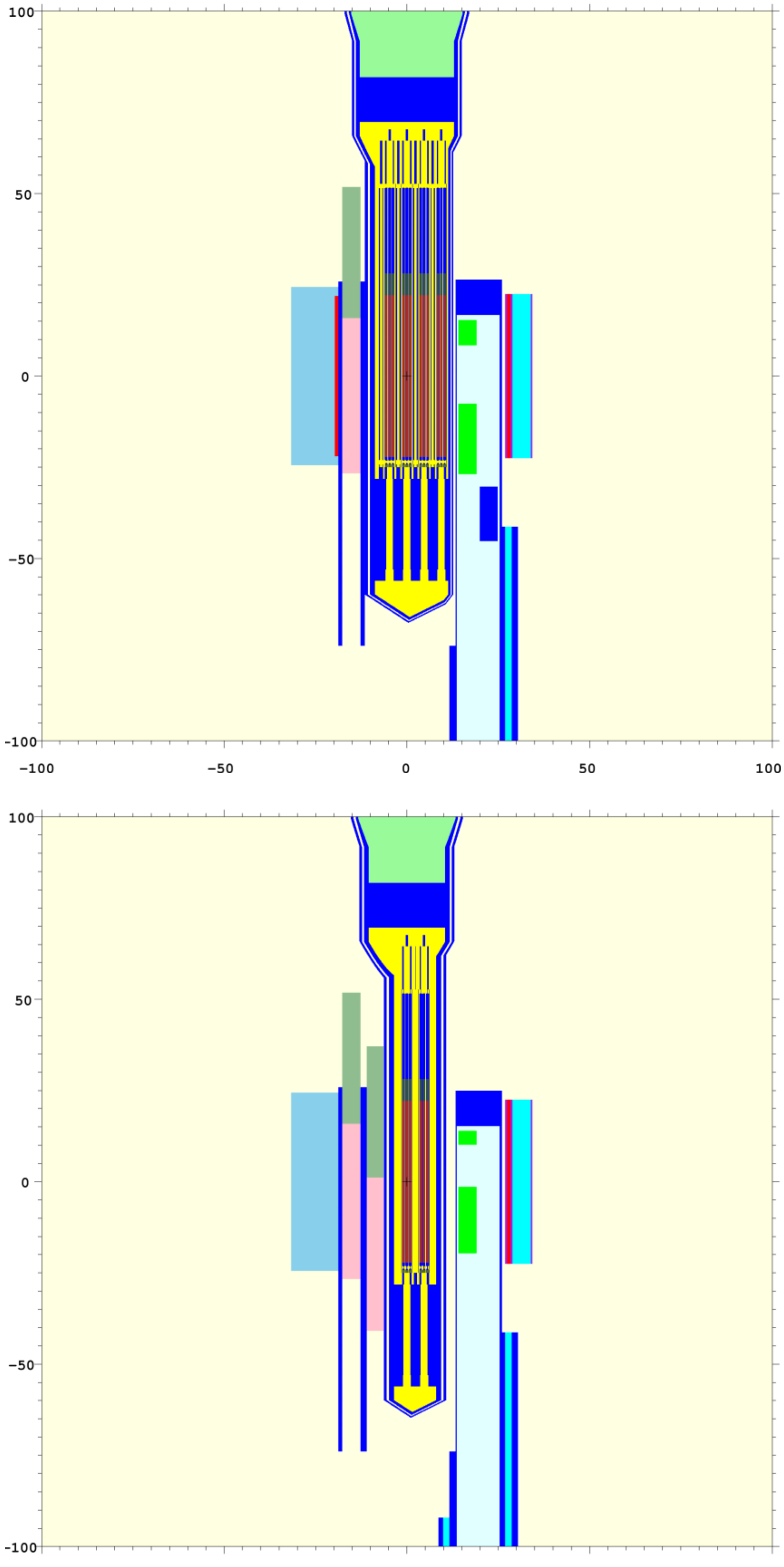


Figure 7. Top plot: px plane with origin at (8, 0, 0) cm and 100 cm extent; bottom plot: px plane with origin at (11, 0, 0) cm and 100 cm extent

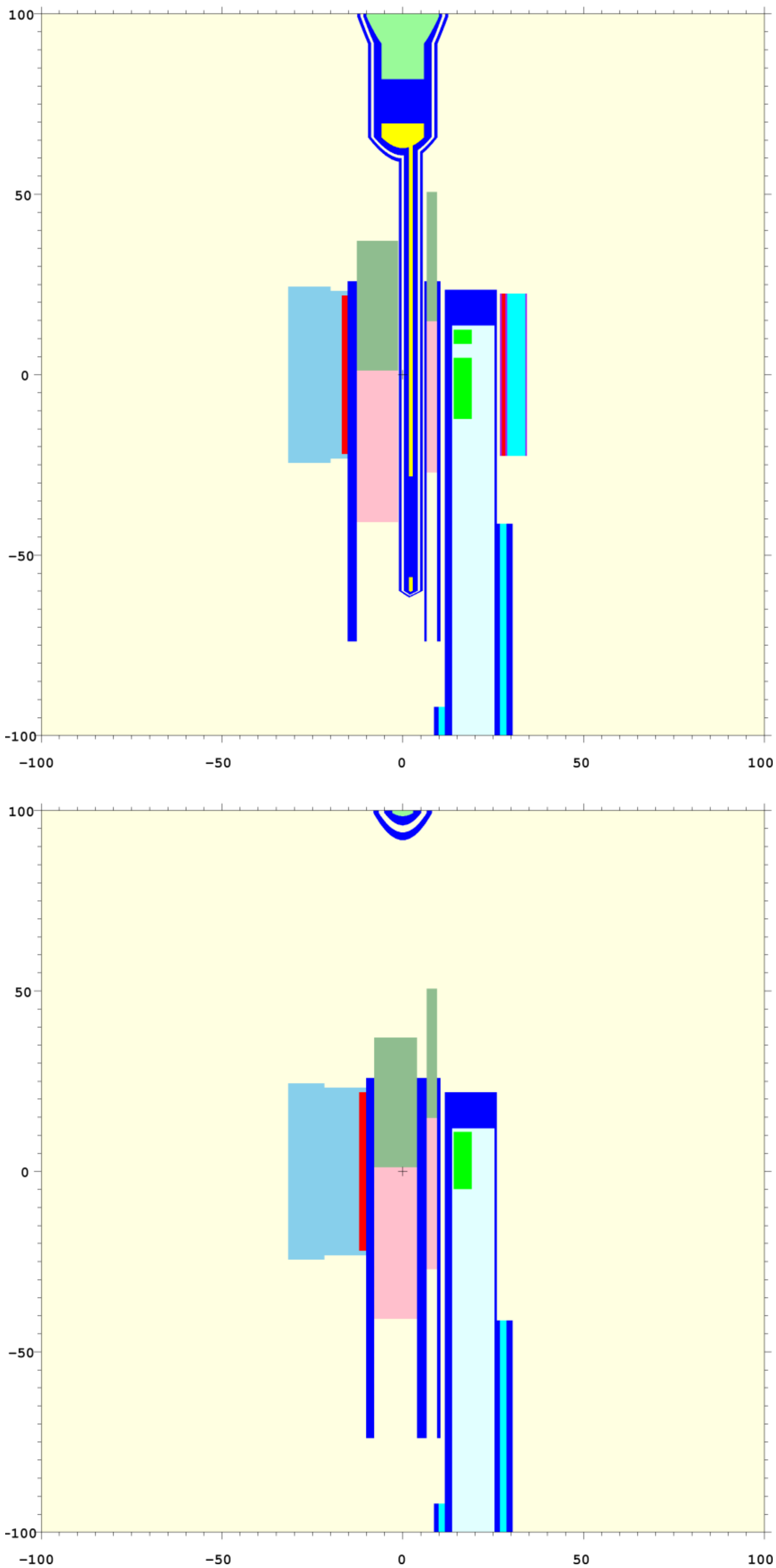


Figure 8. Top plot: px plane with origin at (14, 0, 0) cm and 100 cm extent; bottom plot: px plane with origin at (17, 0, 0) cm and 100 cm extent

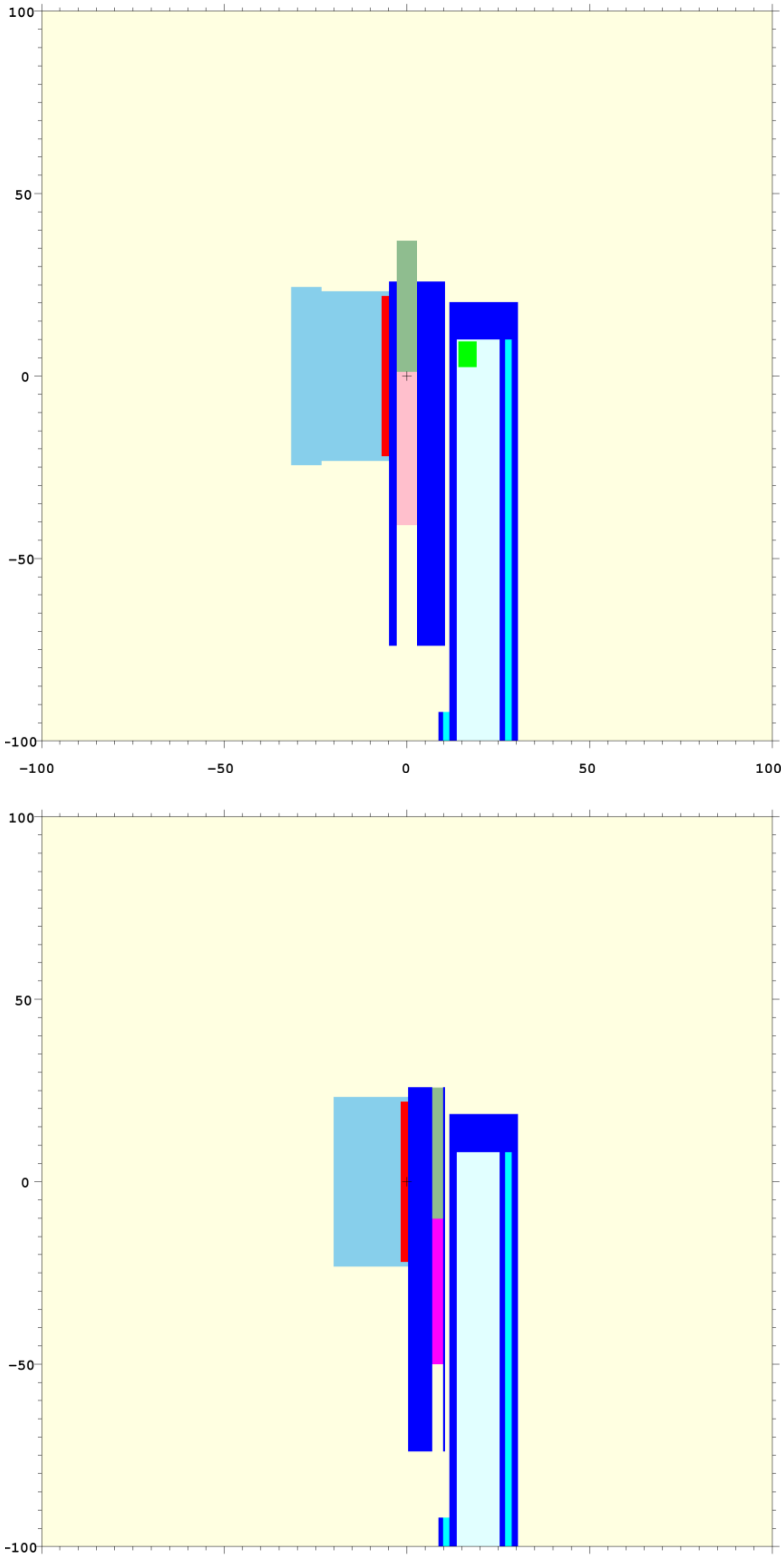


Figure 9. Top plot: px plane with origin at (20, 0, 0) cm and 100 cm extent; bottom plot: px plane with origin at (23, 0, 0) cm and 100 cm extent

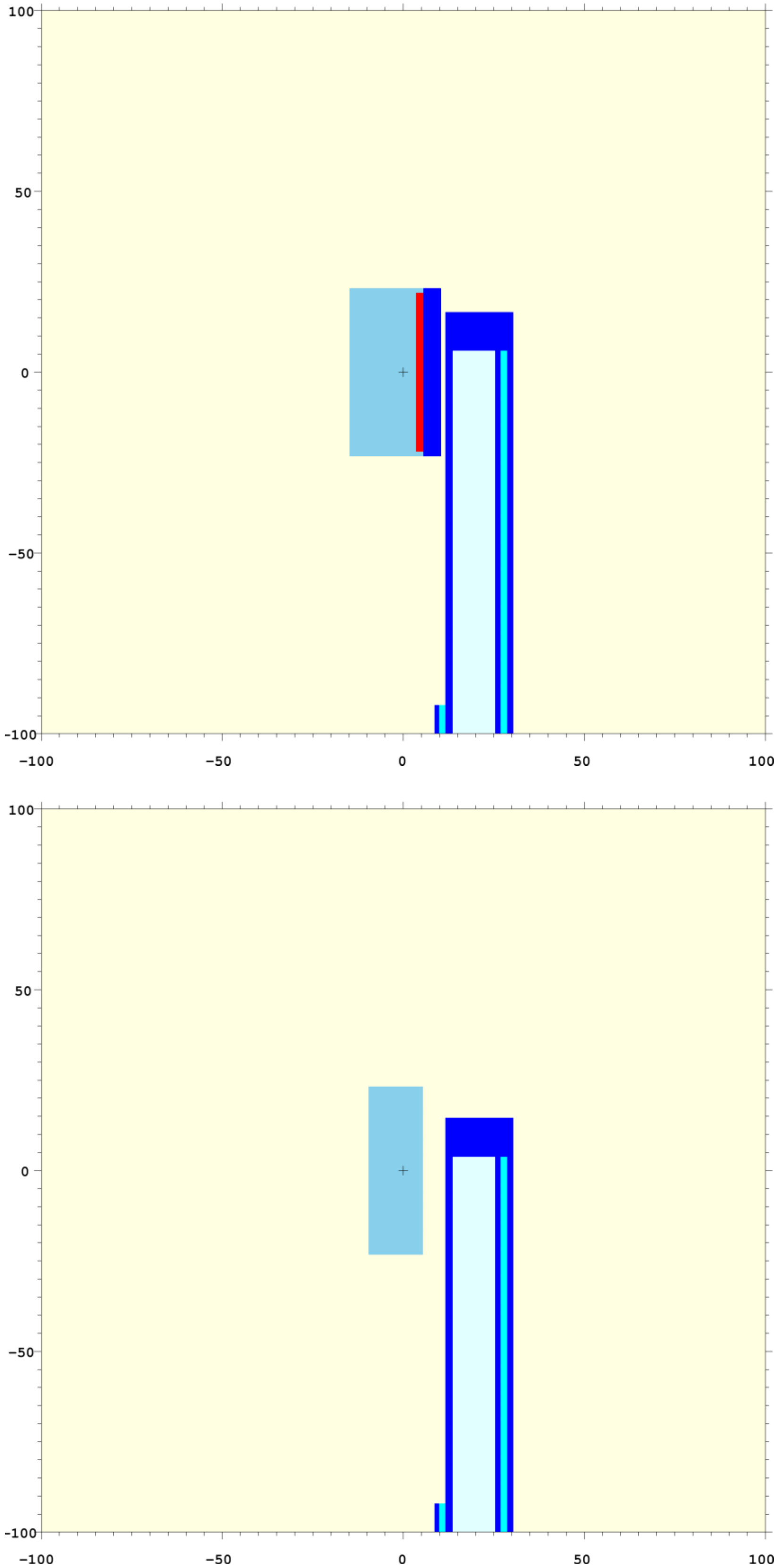


Figure 10. Top plot: px plane with origin at (26, 0, 0) cm and 100 cm extent; bottom plot: px plane with origin at (29, 0, 0) cm and 100 cm extent

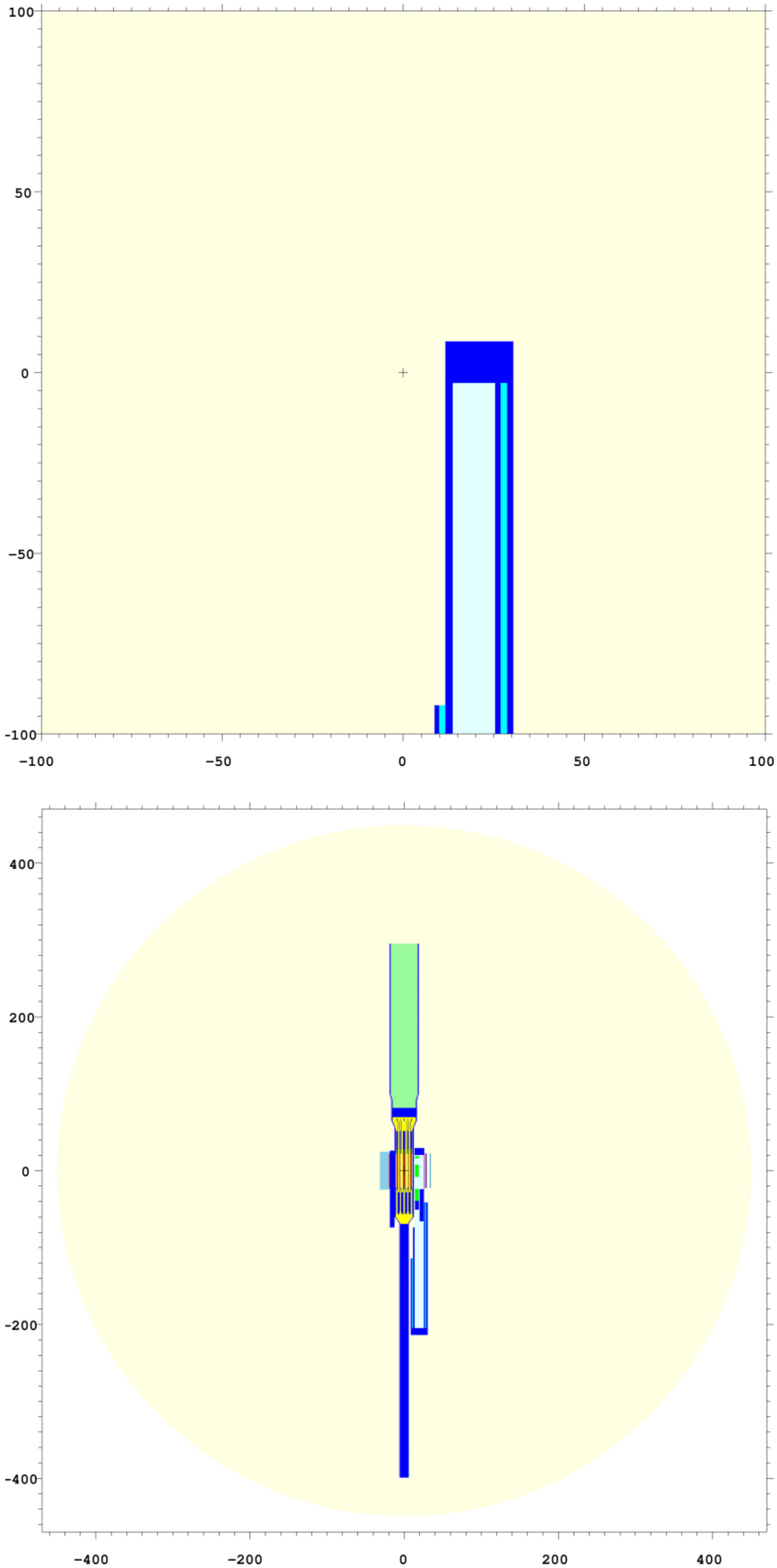


Figure 11. Top plot: px plane with origin at (37, 0, 0) cm and 100 cm extent; bottom plot: px plane with origin at (0, 0, 0) cm and 470 cm extent

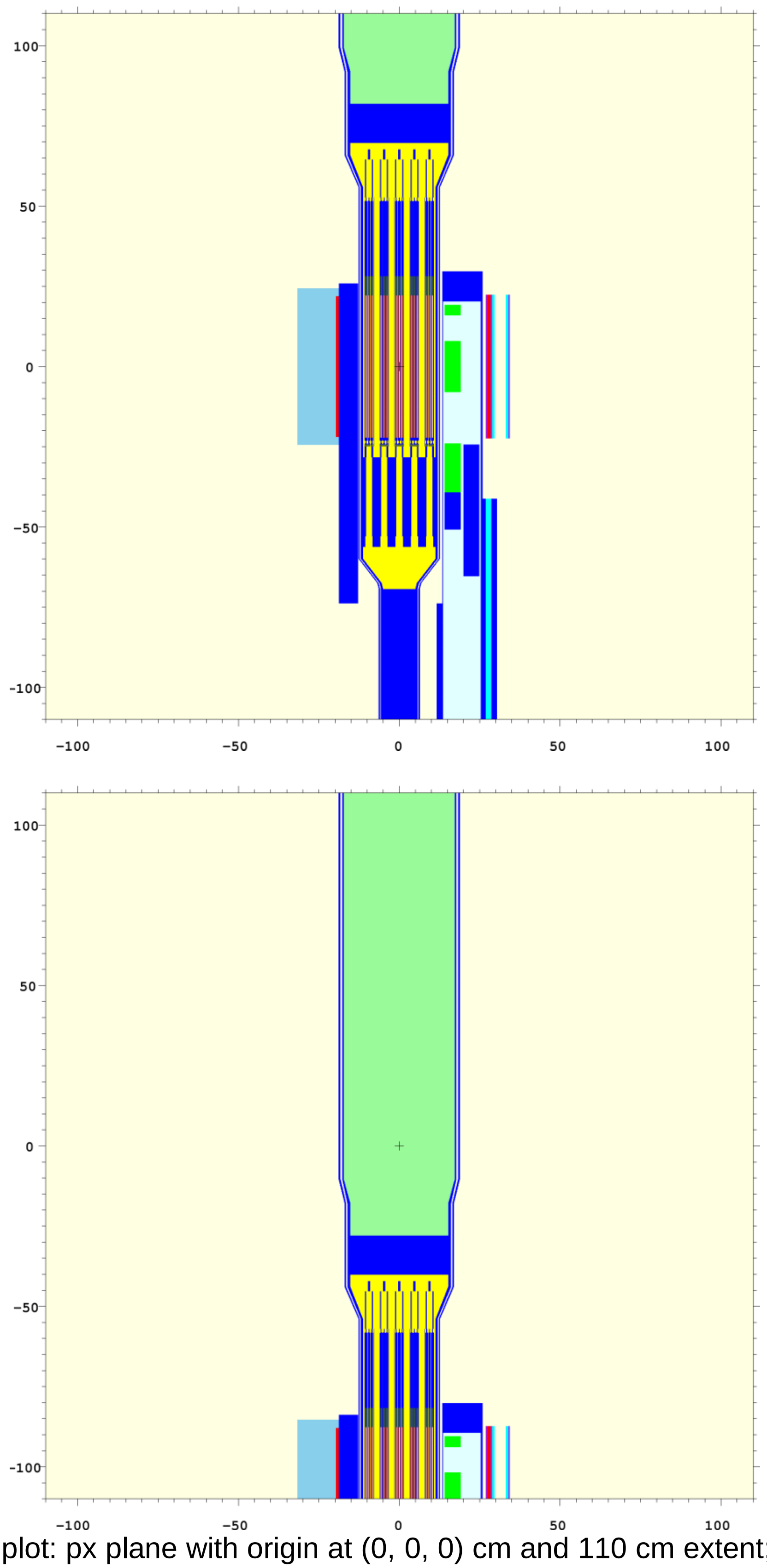


Figure 12. Top plot: px plane with origin at (0, 0, 0) cm and 110 cm extent; bottom plot: px plane with origin at (0, 0, 110) cm and 110 cm extent

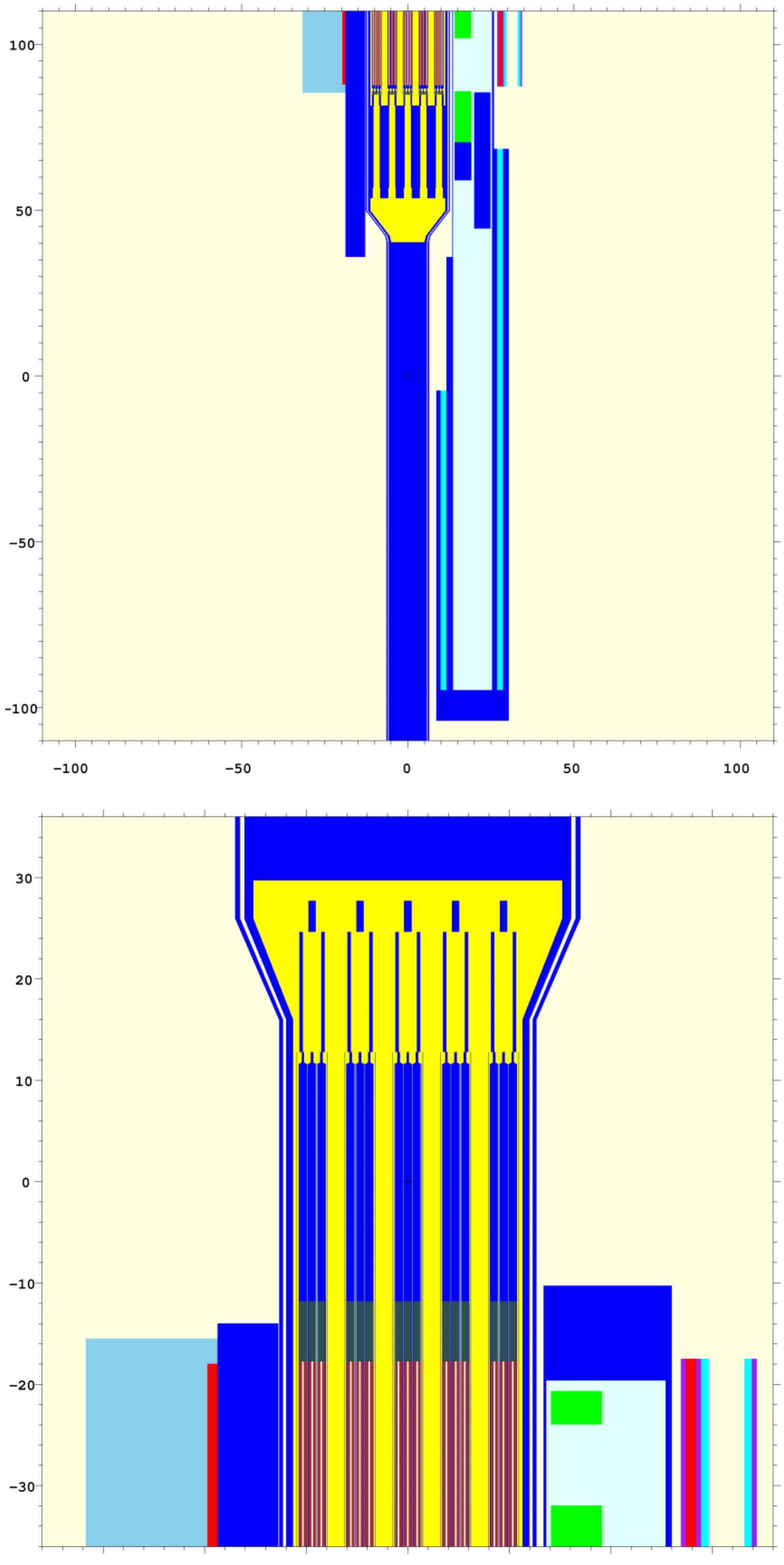


Figure 13. Top plot: px plane with origin at (0, 0, -110) cm and 110 cm extent; bottom plot: px plane with origin at (0, 0, 40) cm and 36 cm extent

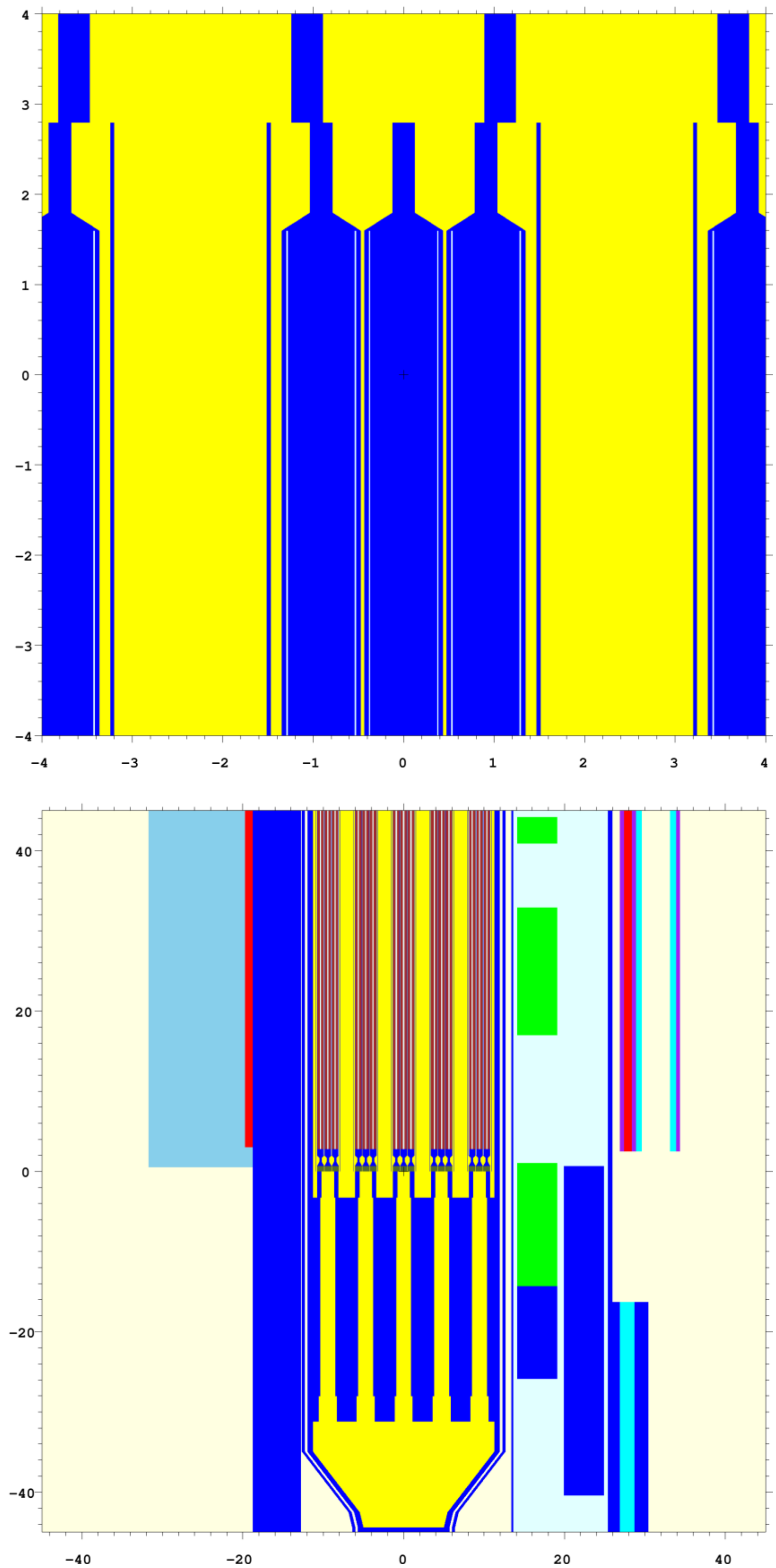


Figure 14. Top plot: px plane with origin at (0, 0, 50) cm and 4 cm extent; bottom plot: px plane with origin at (0, 0, -25) cm and 45 cm extent

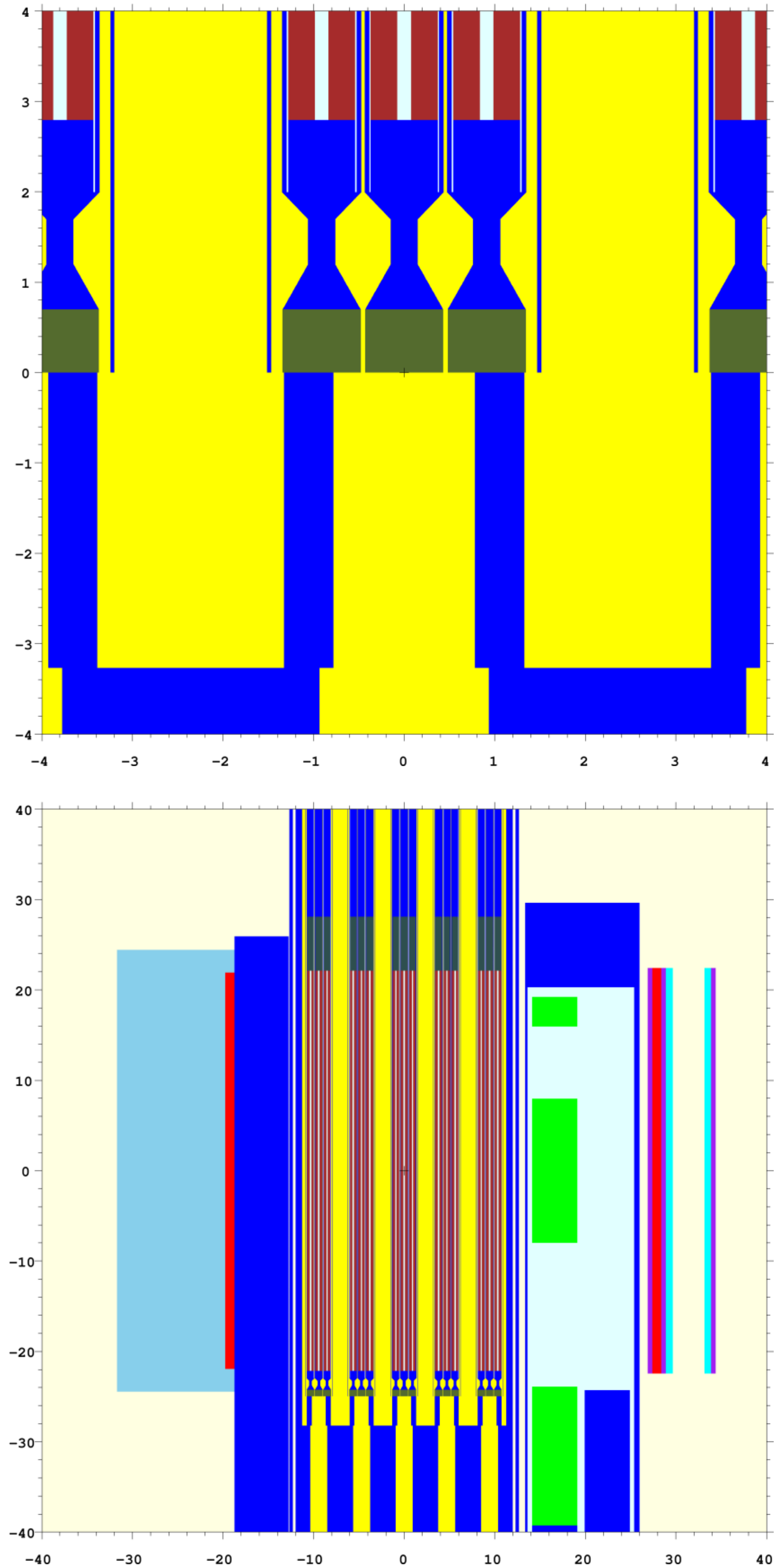


Figure 15. Top plot: px plane with origin at (0, 0, -25) cm and 4 cm extent; bottom plot: px plane with origin at (0, 0, 0) cm and 40 cm extent

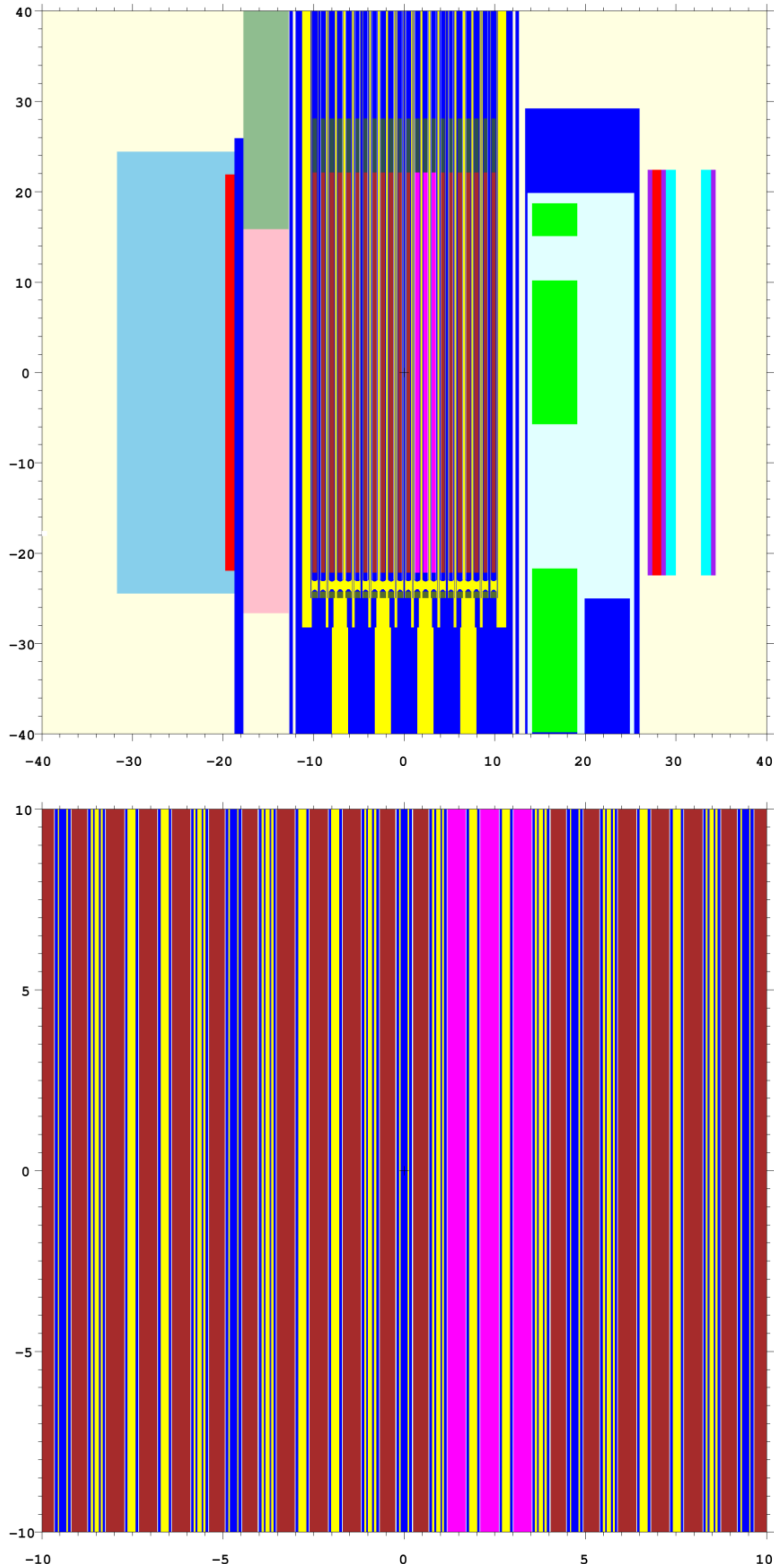


Figure 16. Top plot: px plane with origin at (1.09, 0, 0) cm and 40 cm extent; bottom plot: px plane with origin at (1.09, 0, 0) cm and 10 cm extent

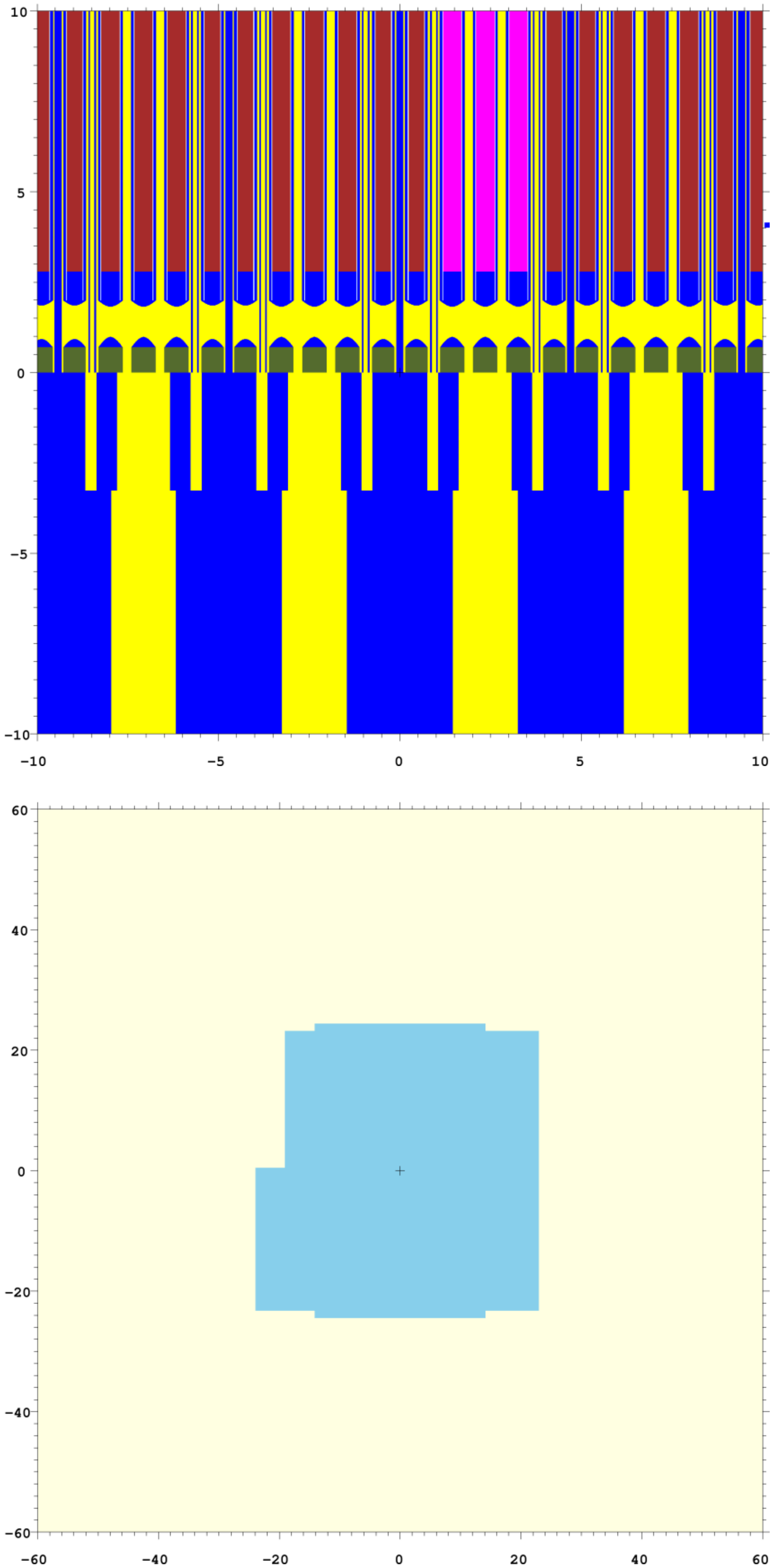


Figure 17. Top plot: px plane with origin at (1.09, 0, -25) cm and 10 cm extent; bottom plot: py plane with origin at (0, -20, 0) cm and 60 cm extent

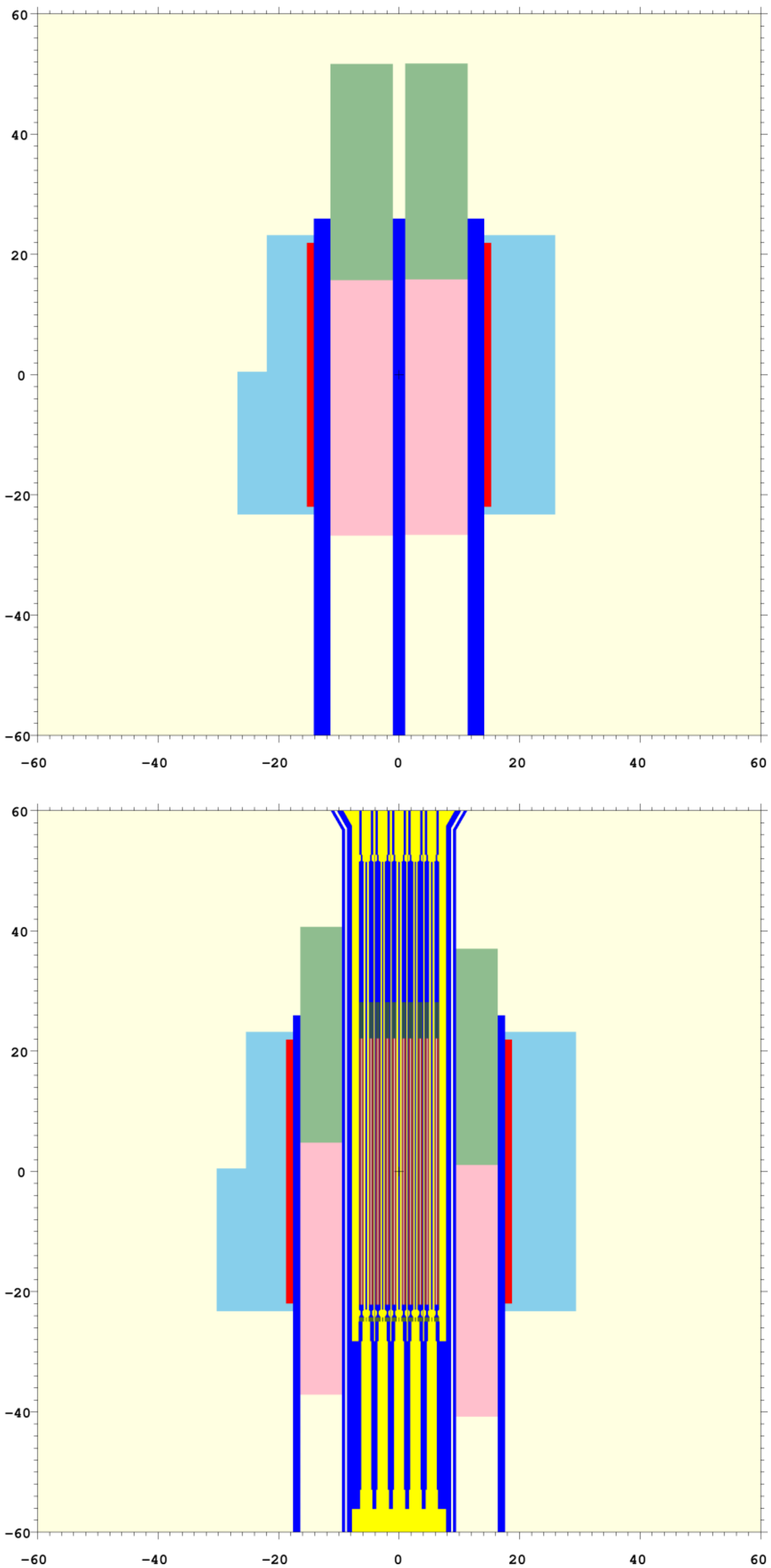
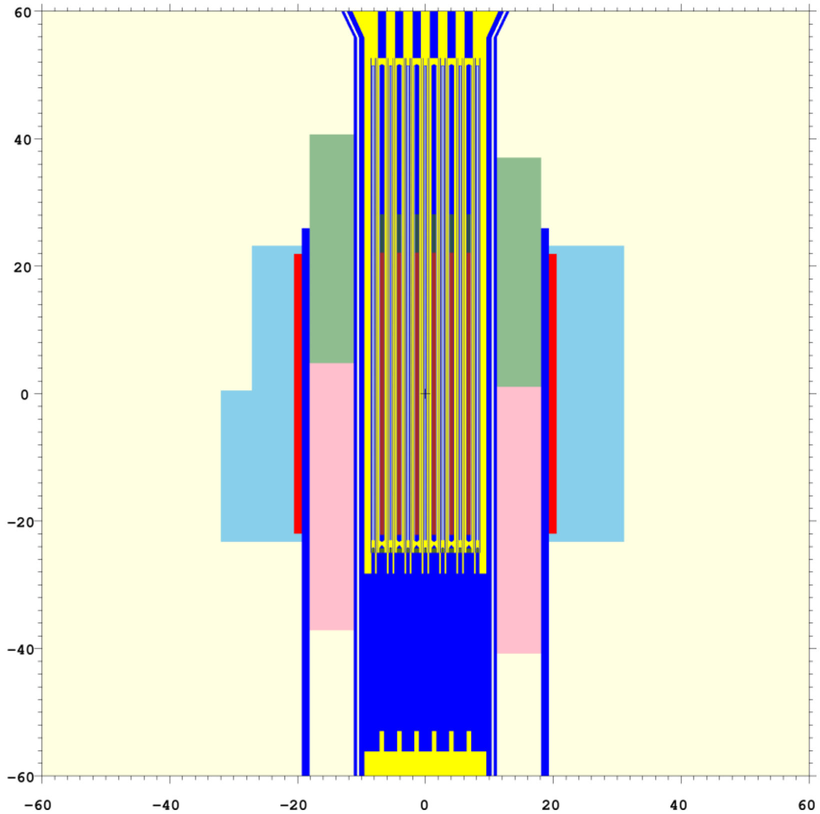


Figure 18. Top plot: py plane with origin at (0, -15, 0) cm and 60 cm extent; bottom plot: py plane with origin at (0, -9, 0) cm and 60 cm extent



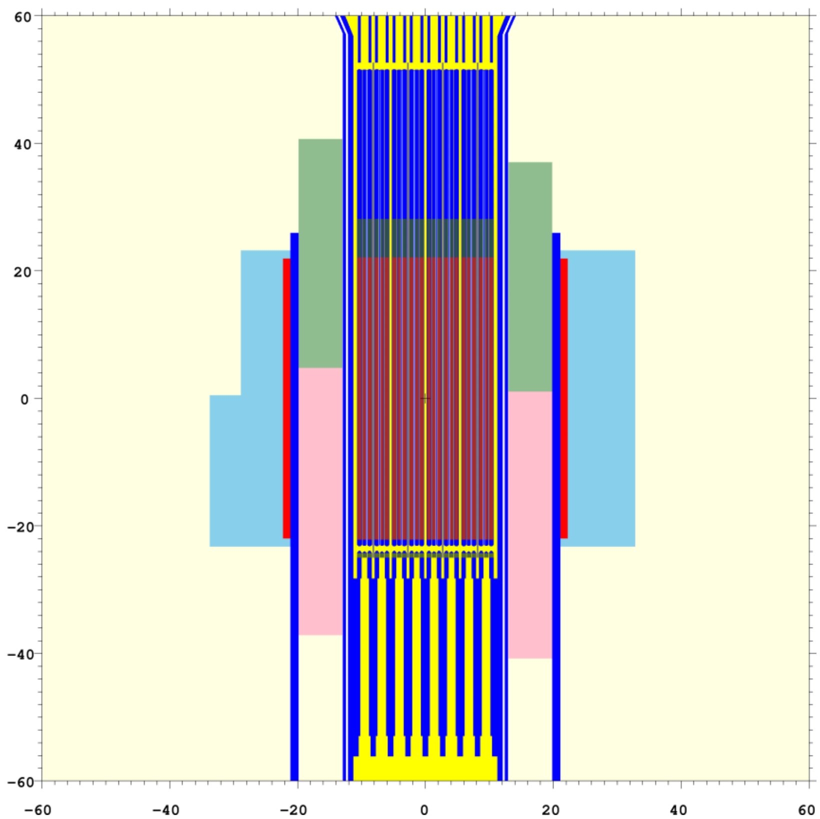
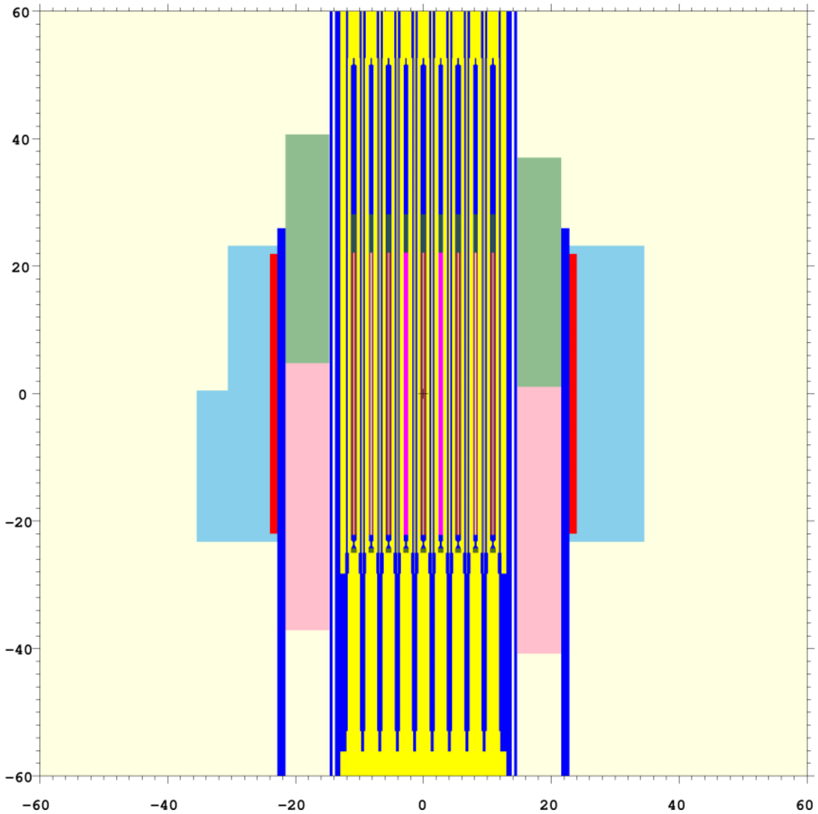


Figure 19. Top plot: py plane with origin at (0, -6, 0) cm and 60 cm extent; bottom plot: py plane with origin at (0, -3, 0) cm and 60 cm extent



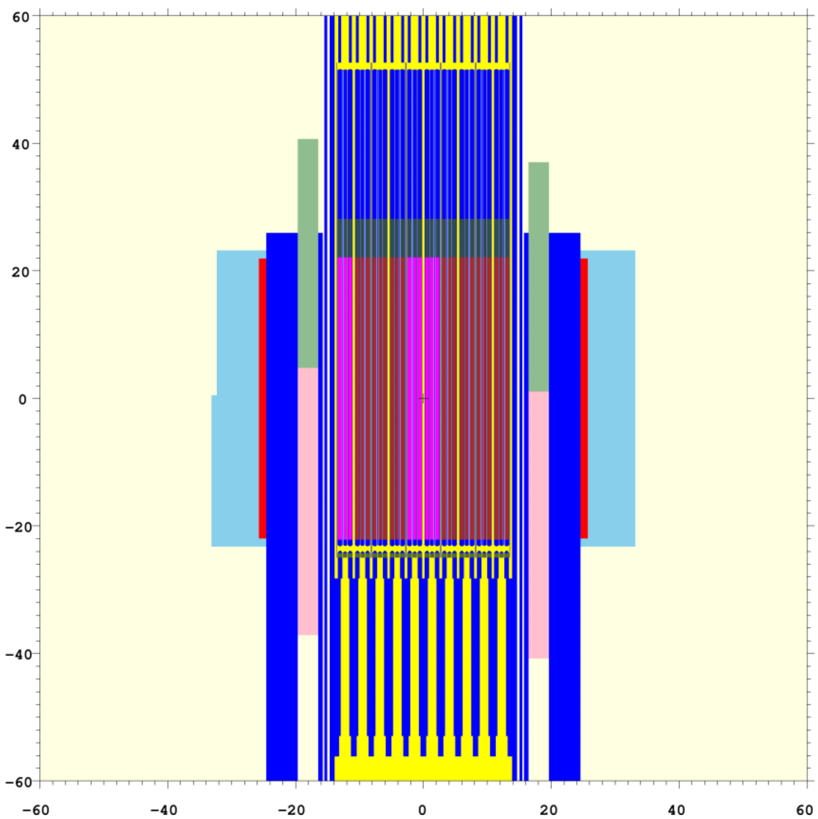


Figure 20. Top plot: py plane with origin at (0, 0, 0) cm and 60 cm extent; bottom plot: py plane with origin at (0, 3, 0) cm and 60 cm extent

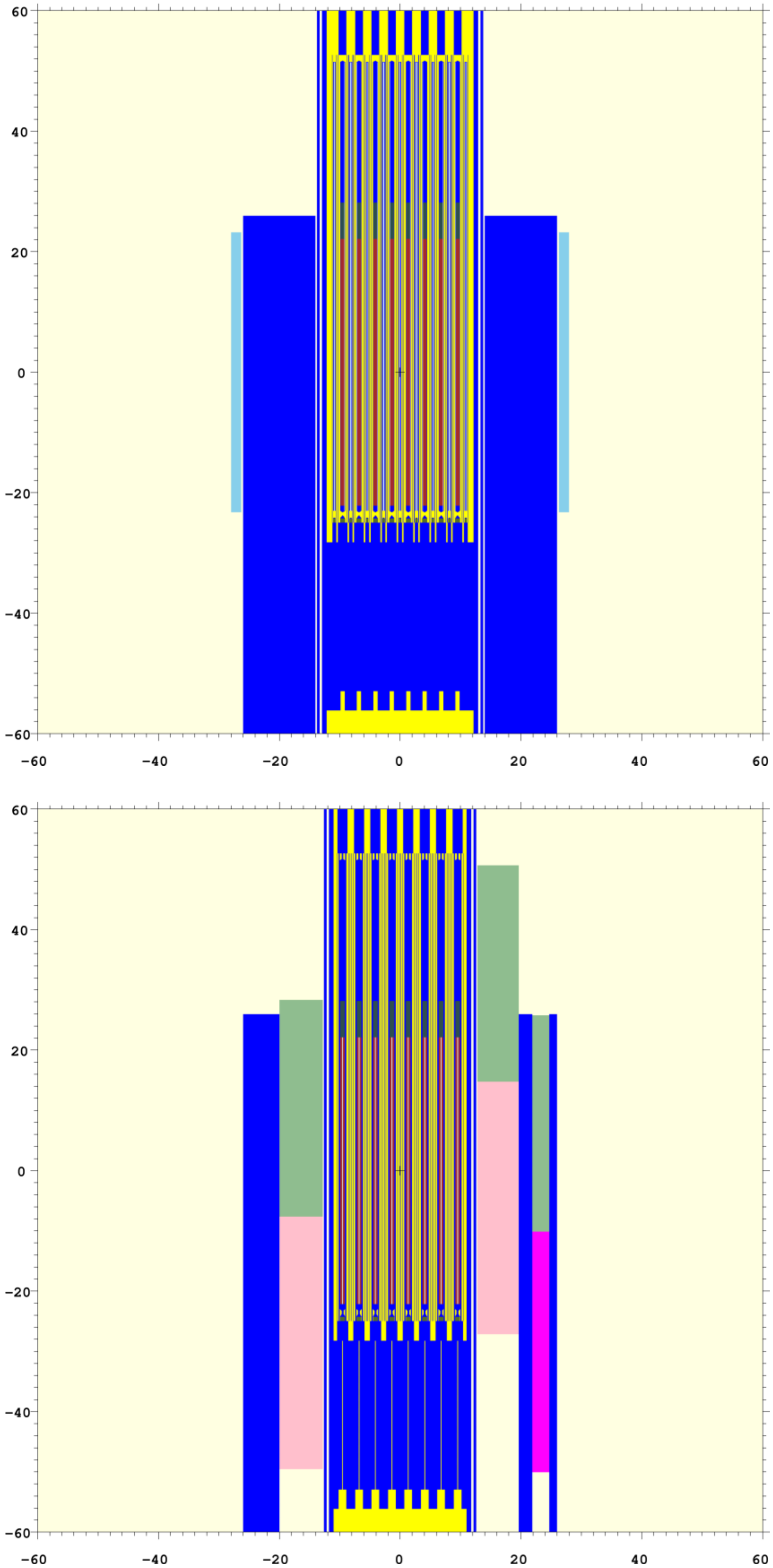


Figure 21. Top plot: py plane with origin at (0, 6, 0) cm and 60 cm extent; bottom plot: py plane with origin at (0, 8, 0) cm and 60 cm extent

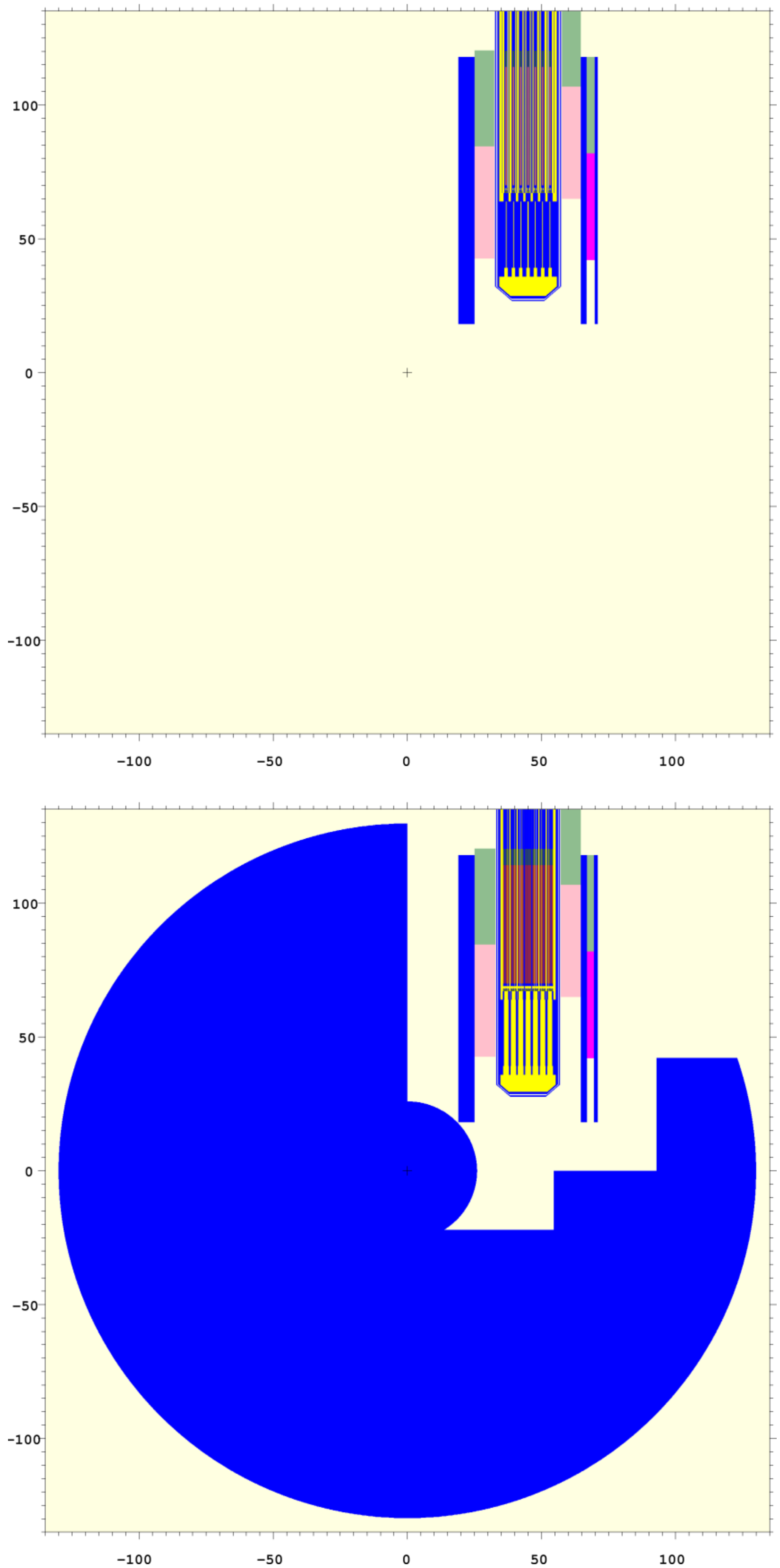


Figure 22. Top plot: py plane with origin at (-45, 8.5, -92.26) cm and 135 cm extent; bottom plot: py plane with origin at (-45, 9.2, -92.26) cm and 135 cm extent

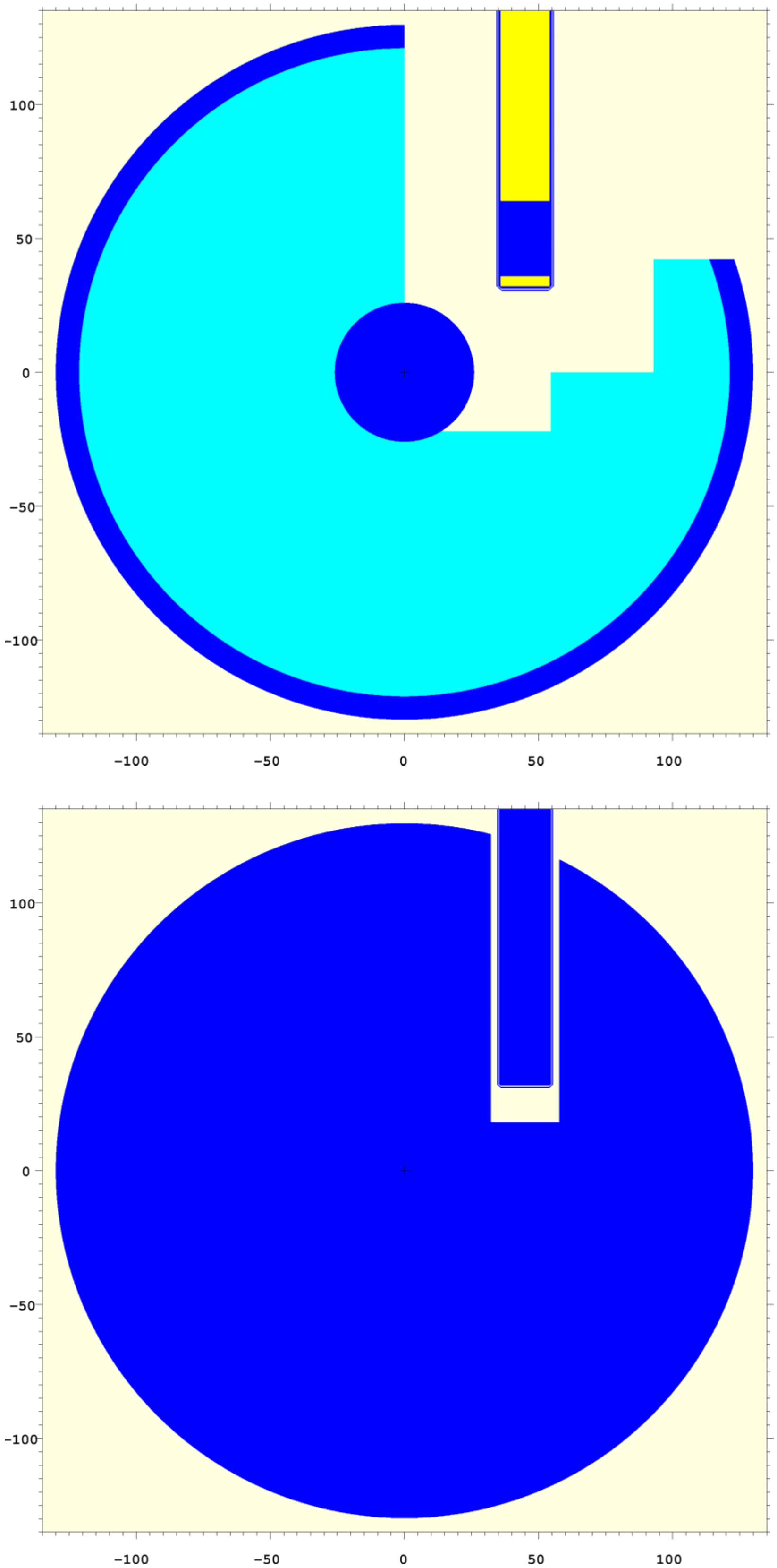


Figure 23. Top plot: py plane with origin at (-45, 11.2, -92.26) cm and 135 cm extent; bottom plot: py plane with origin at (-45, 11.7, -92.26) cm and 135 cm extent

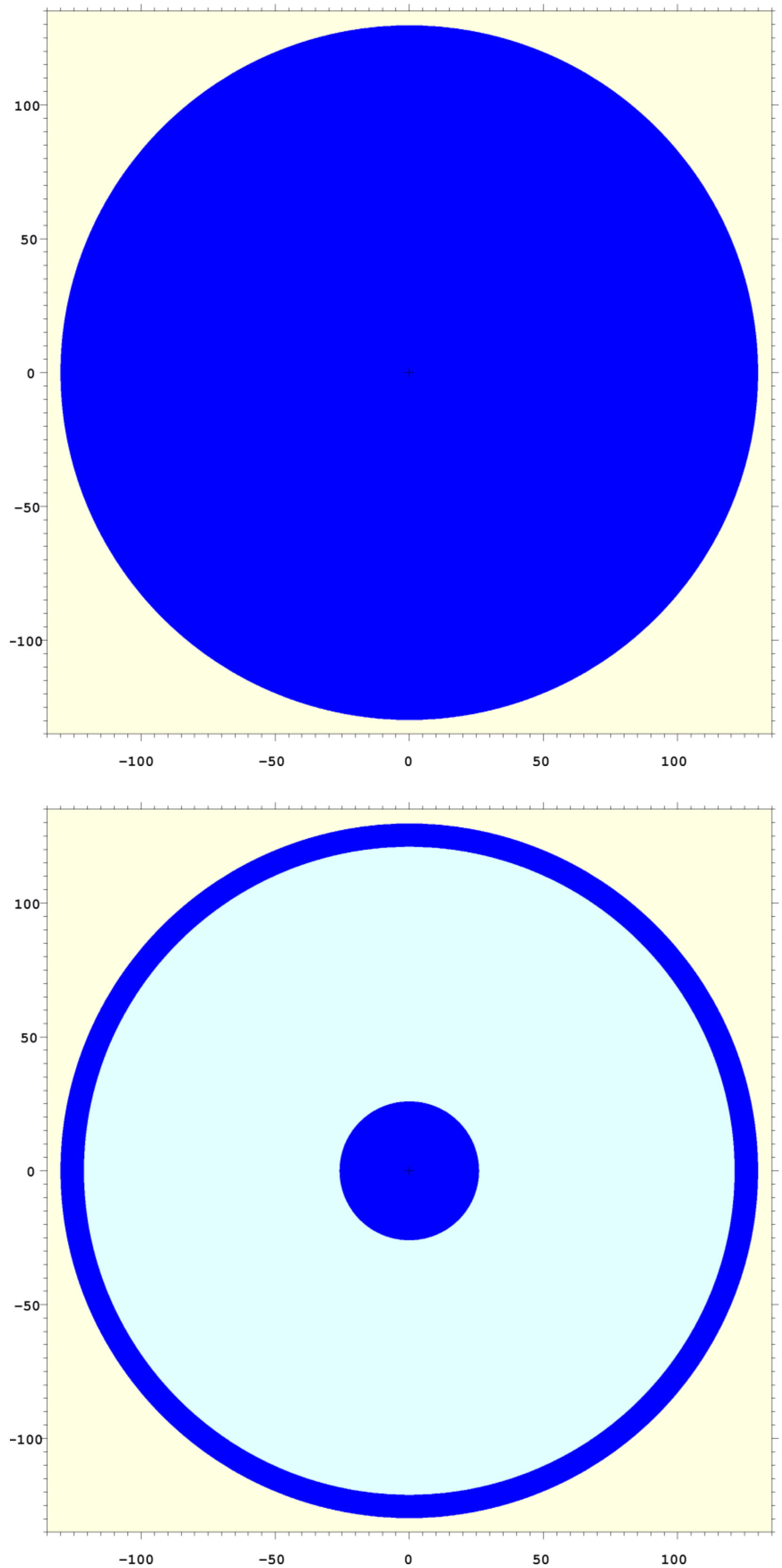
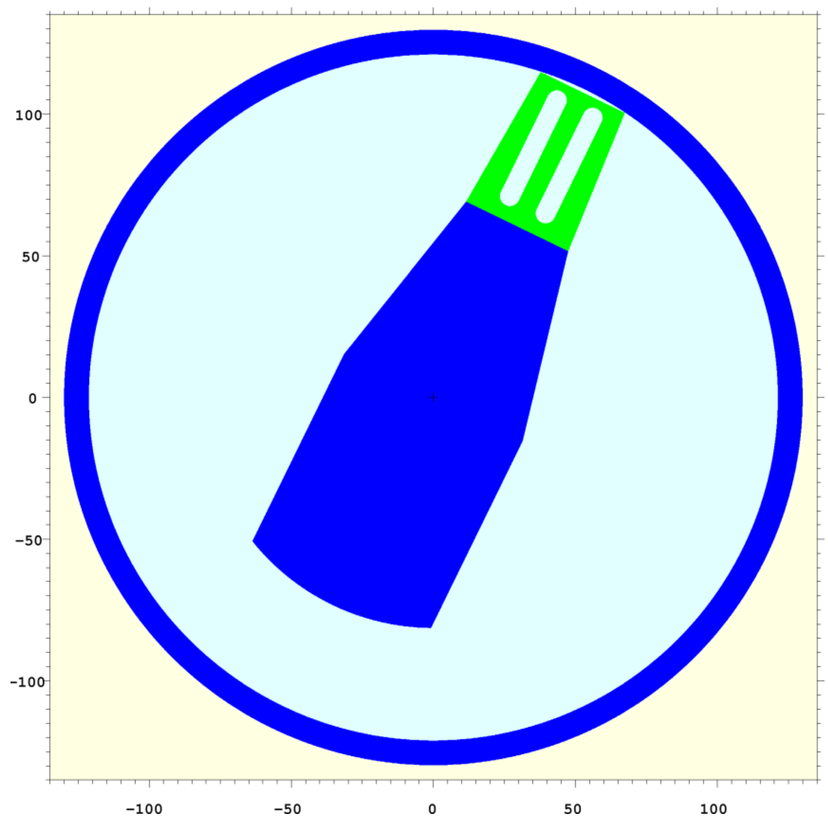


Figure 24. Top plot: py plane with origin at (-45, 13.4, -92.26) cm and 135 cm extent; bottom plot: py plane with origin at (-45, 13.7, -92.26) cm and 135 cm extent



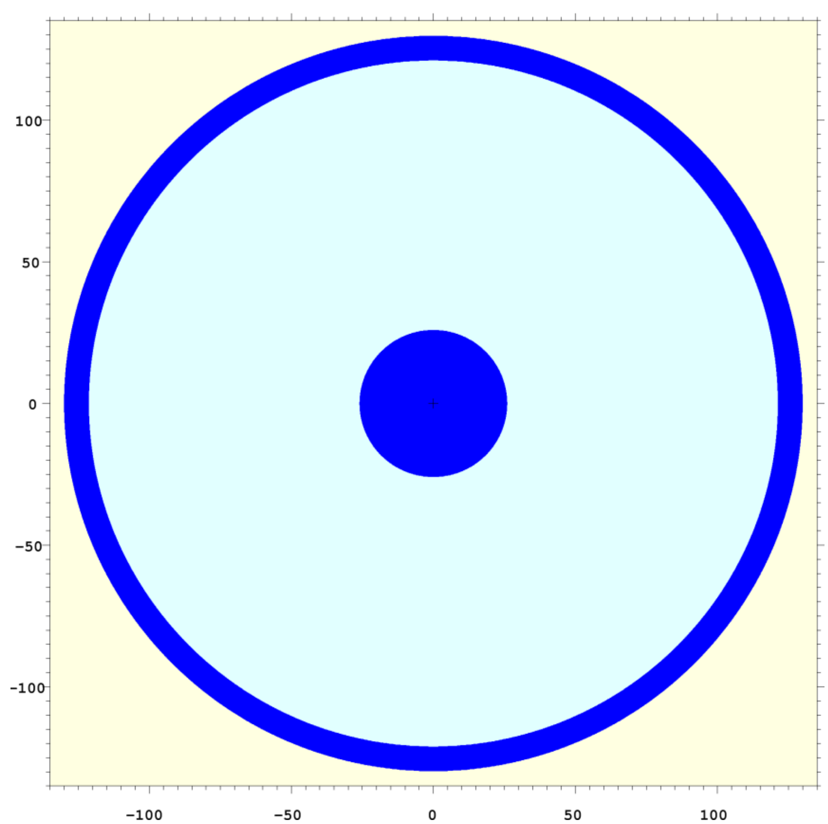
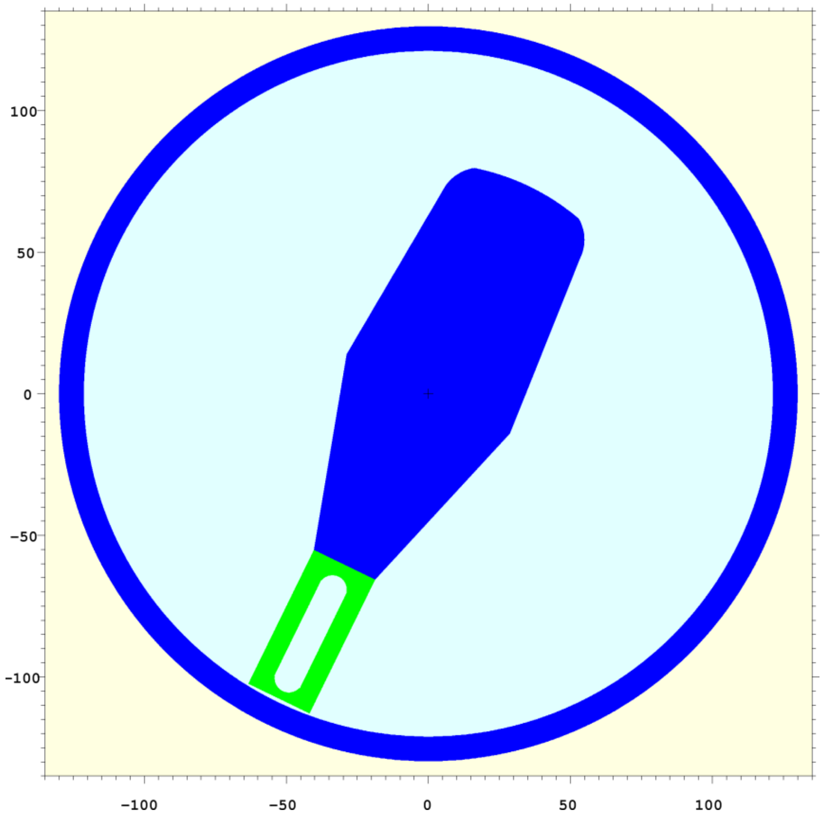


Figure 25. Top plot: py plane with origin at (-45, 15, -92.26) cm and 135 cm extent; bottom plot: py plane with origin at (-45, 19.9, -92.26) cm and 135 cm extent



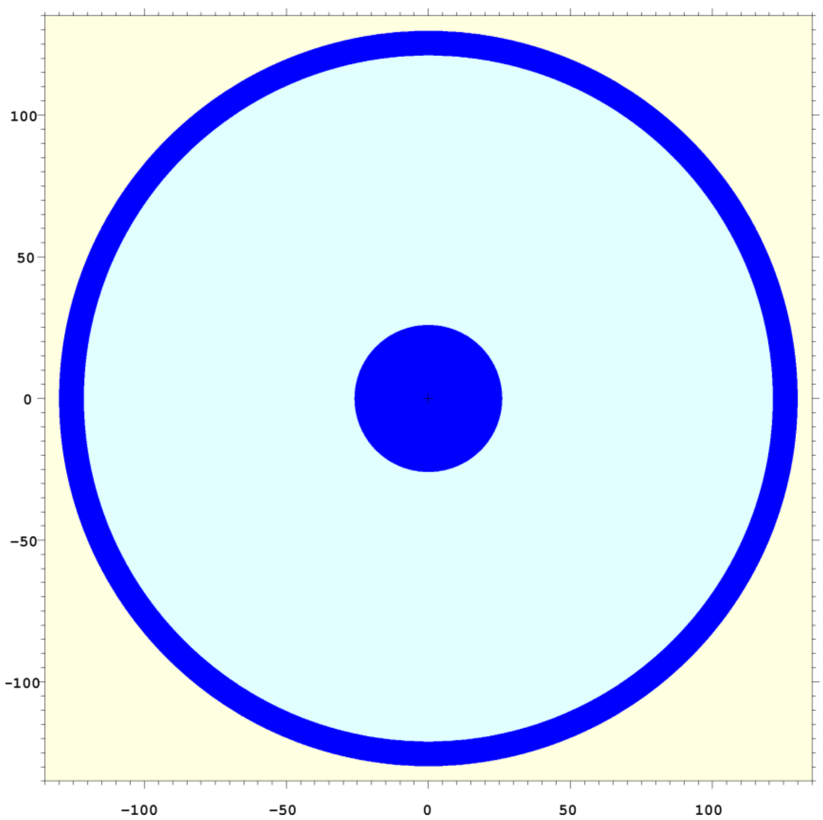


Figure 26. Top plot: py plane with origin at (-45, 20, -92.26) cm and 135 cm extent; bottom plot: py plane with origin at (-45, 25, -92.26) cm and 135 cm extent

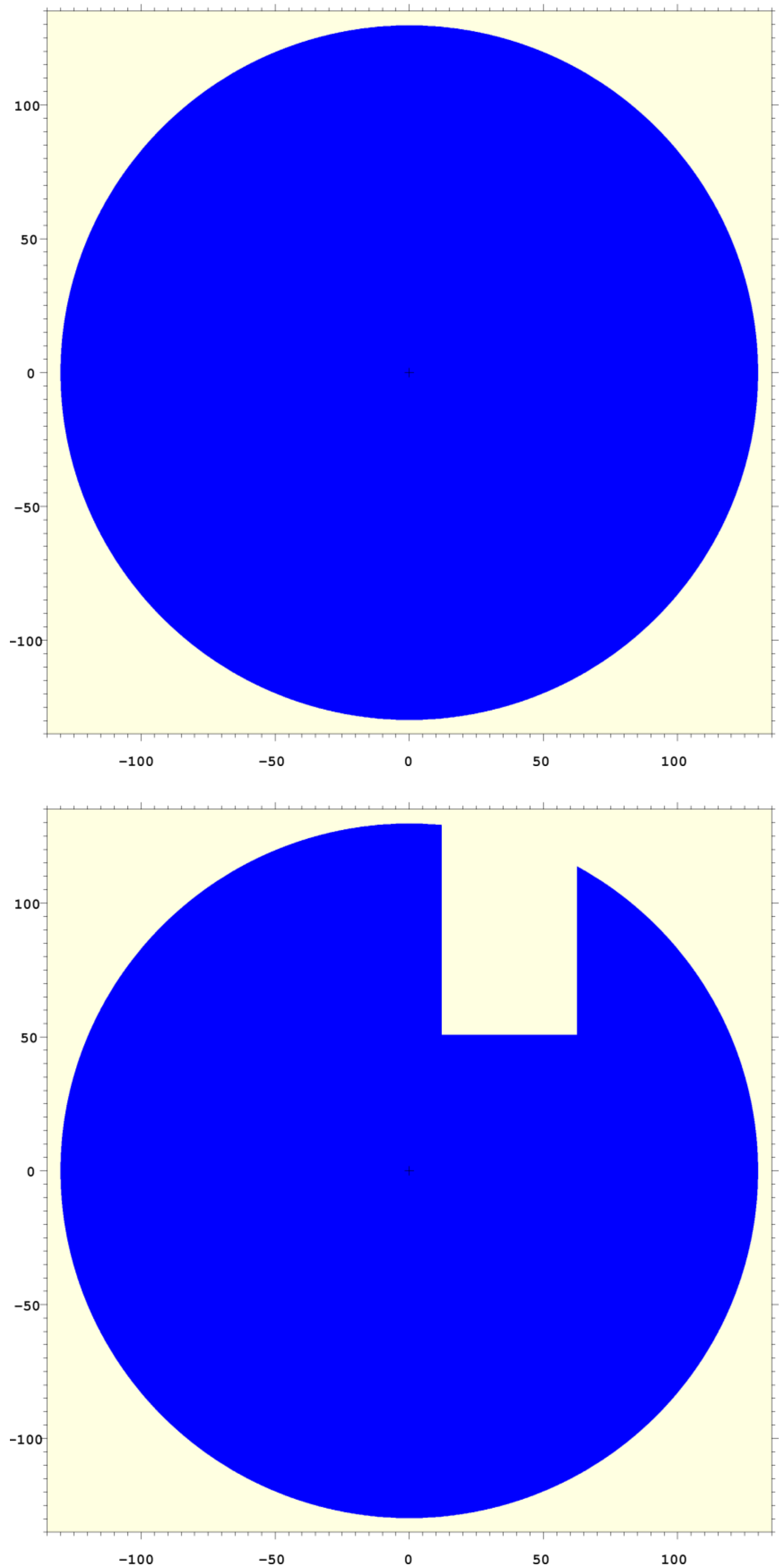


Figure 27. Top plot: py plane with origin at (-45, 25.4, -92.26) cm and 135 cm extent; bottom plot: py plane with origin at (-45, 26, -92.26) cm and 135 cm extent

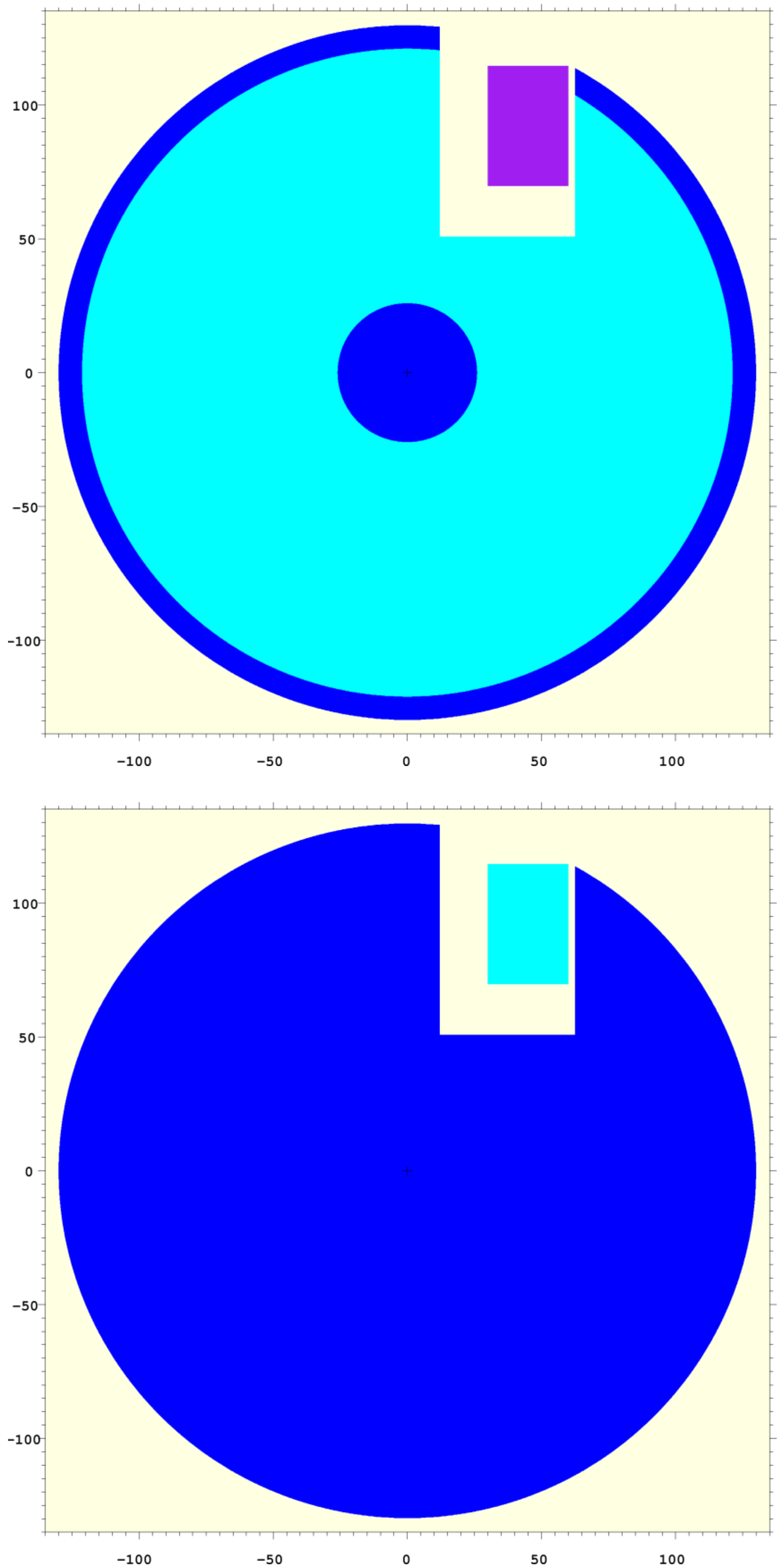


Figure 28. Top plot: py plane with origin at (-45, 27, -92.26) cm and 135 cm extent; bottom plot: py plane with origin at (-45, 29, -92.26) cm and 135 cm extent

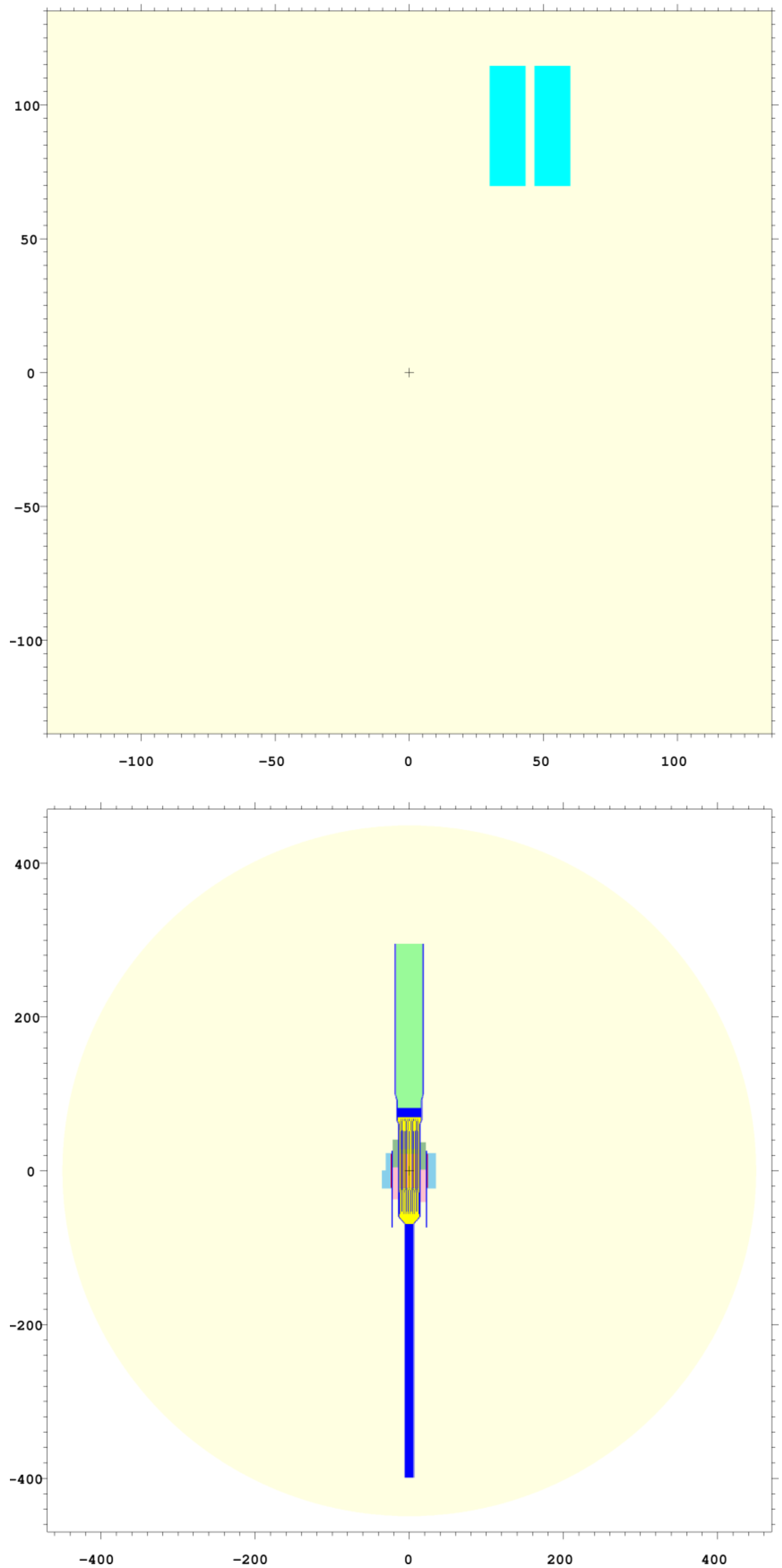


Figure 29. Top plot: py plane with origin at (-45, 31, -92.26) cm and 135 cm extent; bottom plot: py plane with origin at (0, 0, 0) cm and 470 cm extent

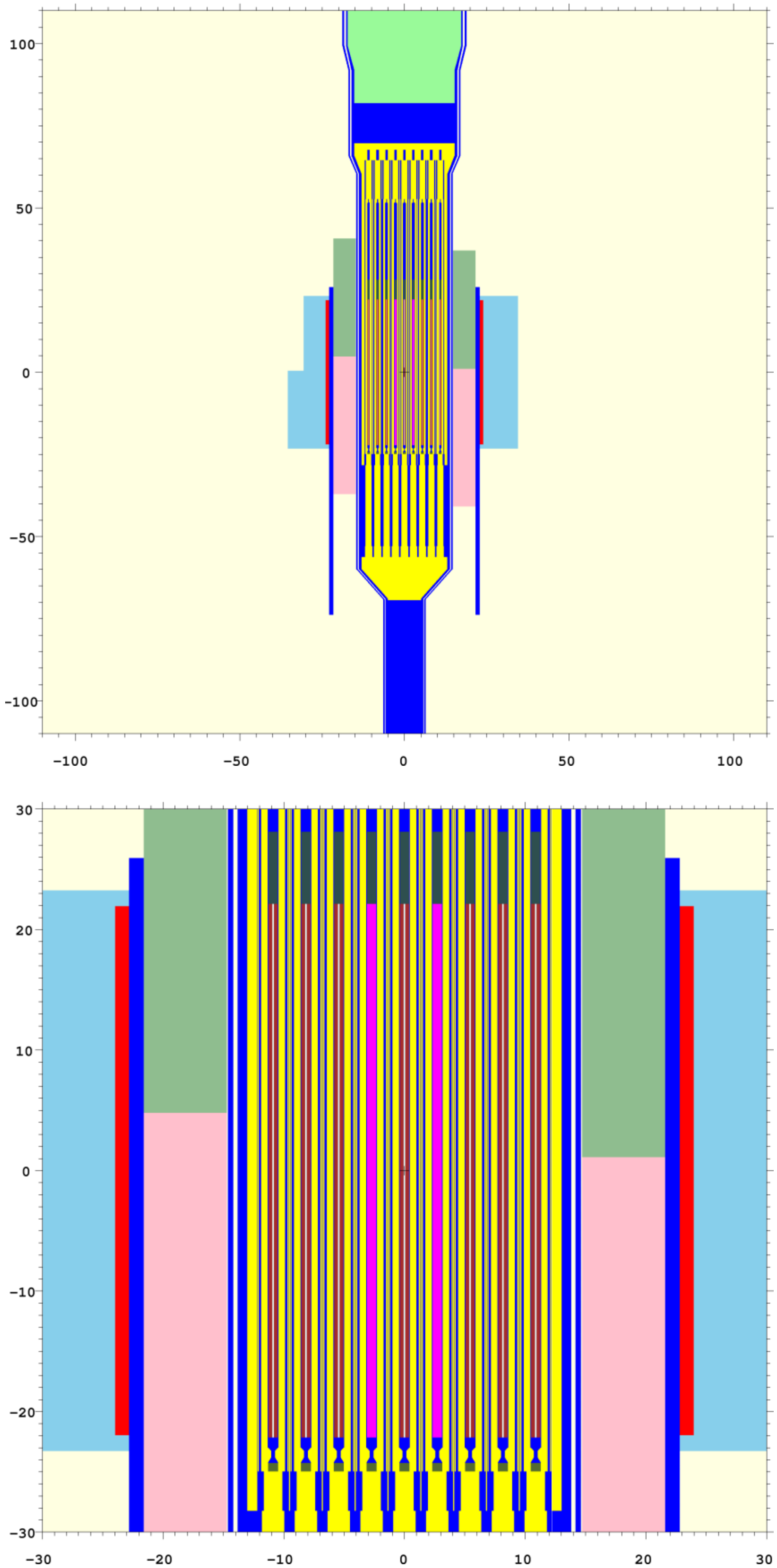


Figure 30. Top plot: py plane with origin at (0, 0, 0) cm and 110 cm extent; bottom plot: py plane with origin at (0, 0, 0) cm and 30 cm extent

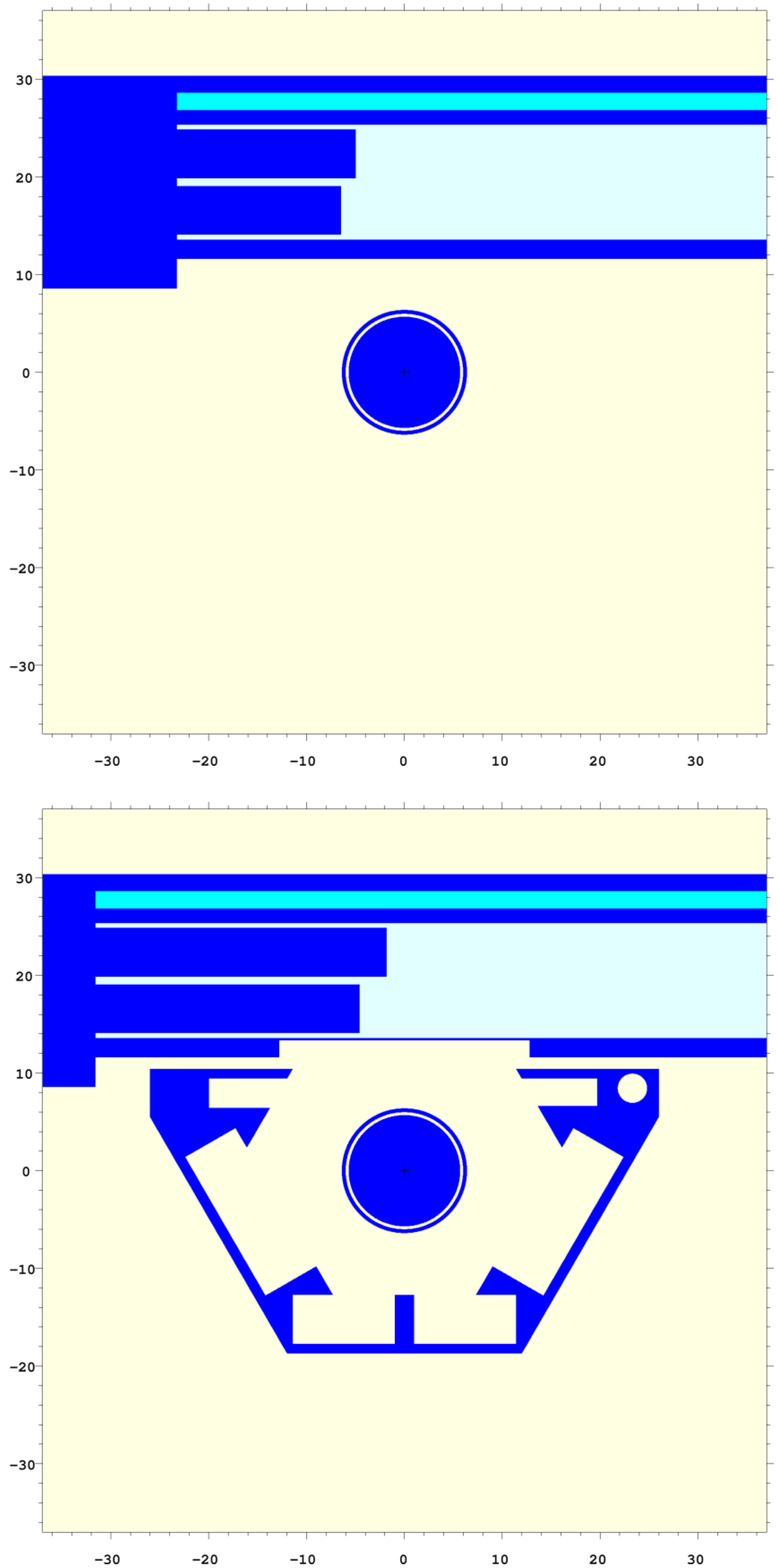


Figure 31. Top plot: pz plane with origin at (0, 0, -78) cm and 37 cm extent; bottom plot: pz plane with origin at (0, 0, -70) cm and 37 cm extent

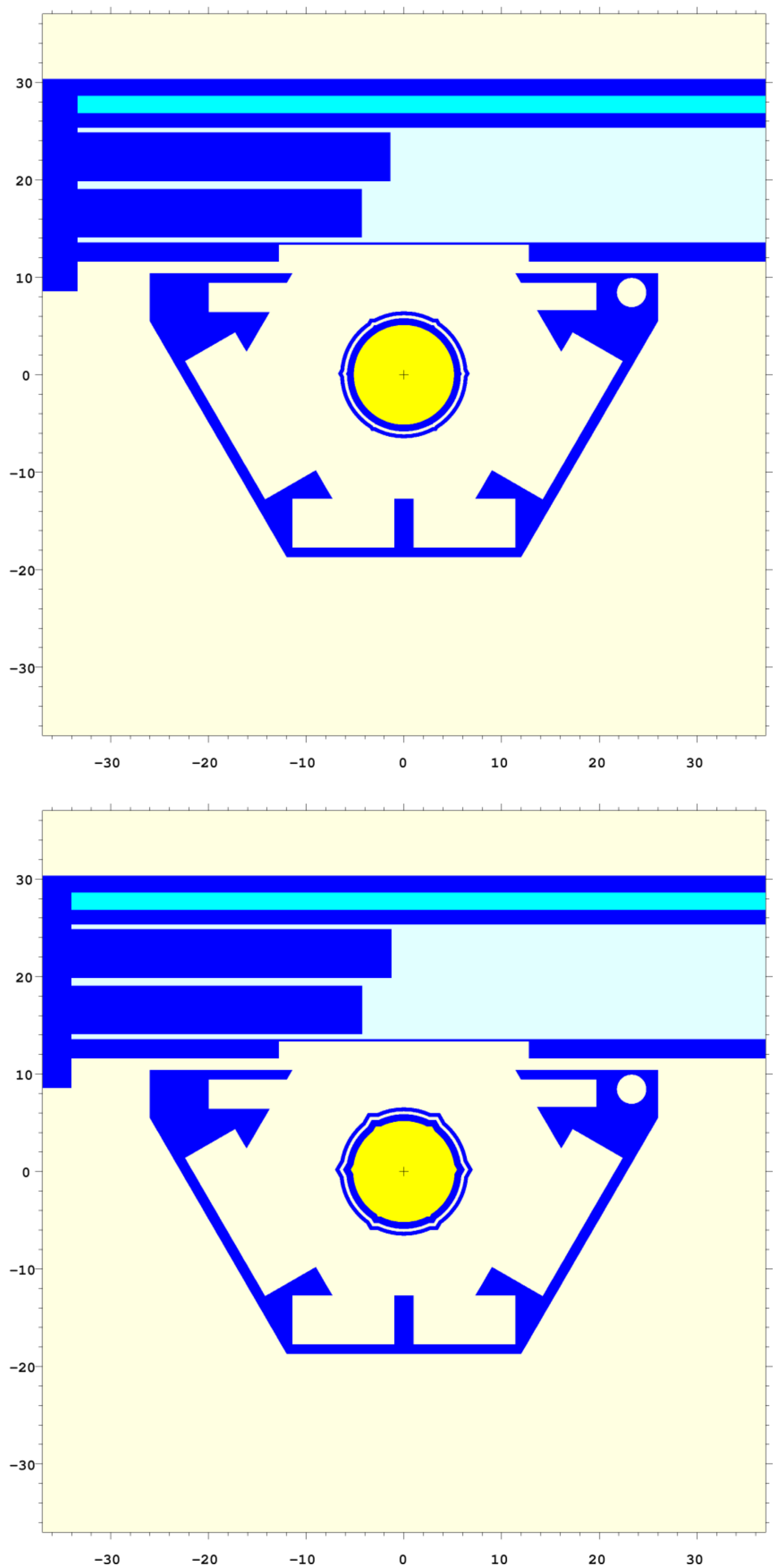


Figure 32. Top plot: pz plane with origin at (0, 0, -69) cm and 37 cm extent; bottom plot: pz plane with origin at (0, 0, -68.7) cm and 37 cm extent

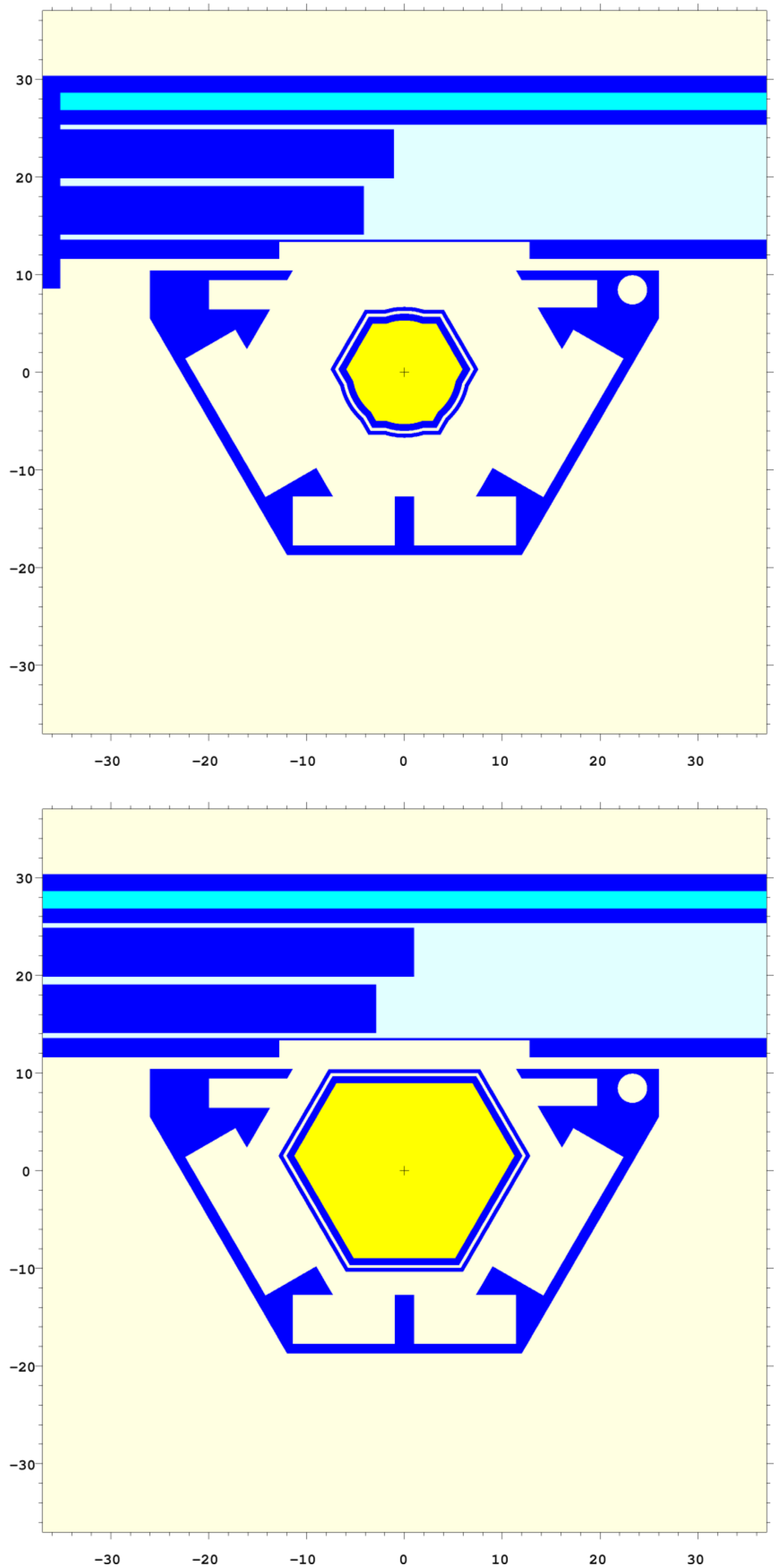


Figure 33. Top plot: pz plane with origin at (0, 0, -68.2) cm and 37 cm extent; bottom plot: pz plane with origin at (0, 0, -63) cm and 37 cm extent

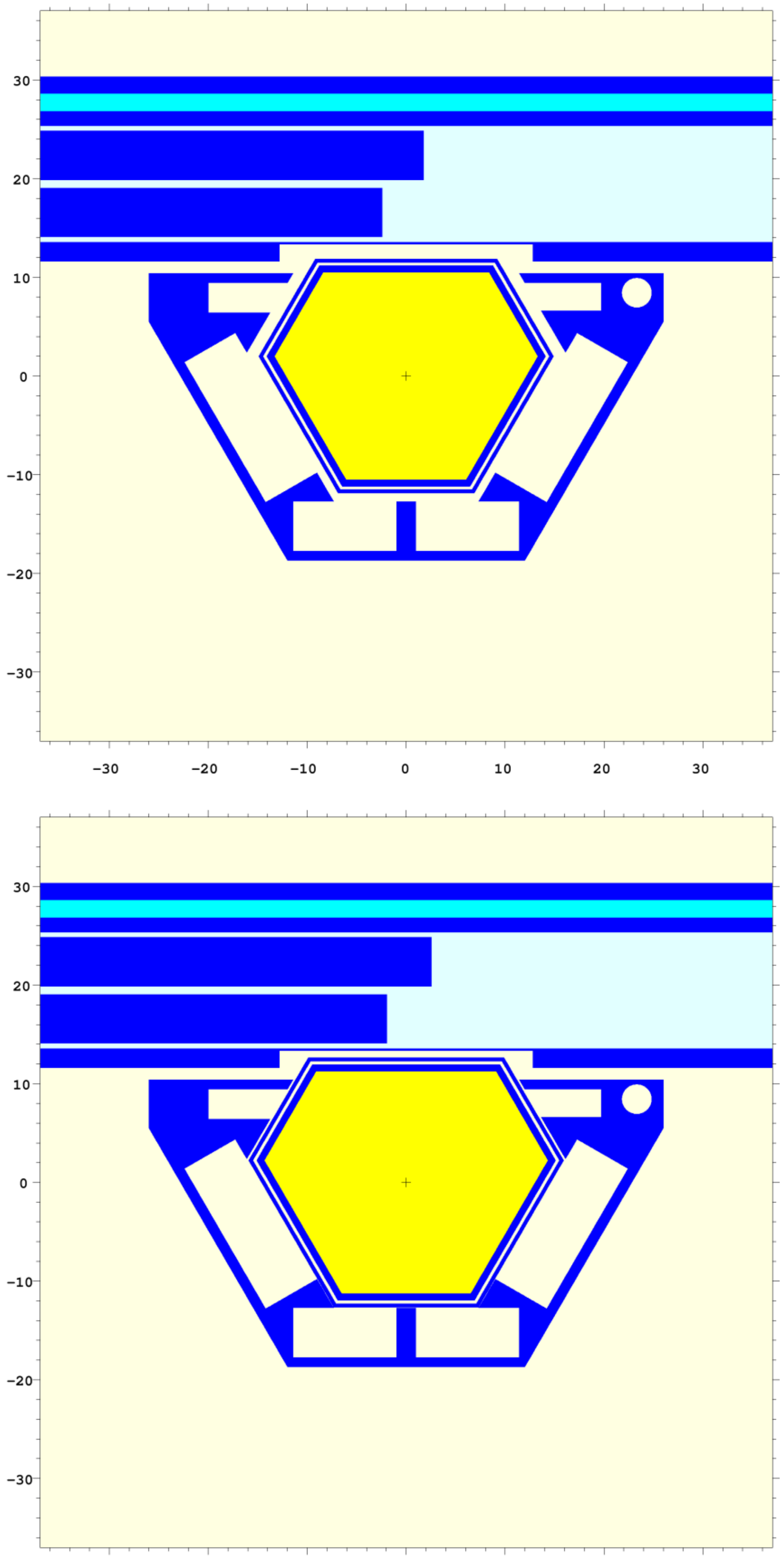


Figure 34. Top plot: pz plane with origin at (0, 0, -61) cm and 37 cm extent; bottom plot: pz plane with origin at (0, 0, -59) cm and 37 cm extent

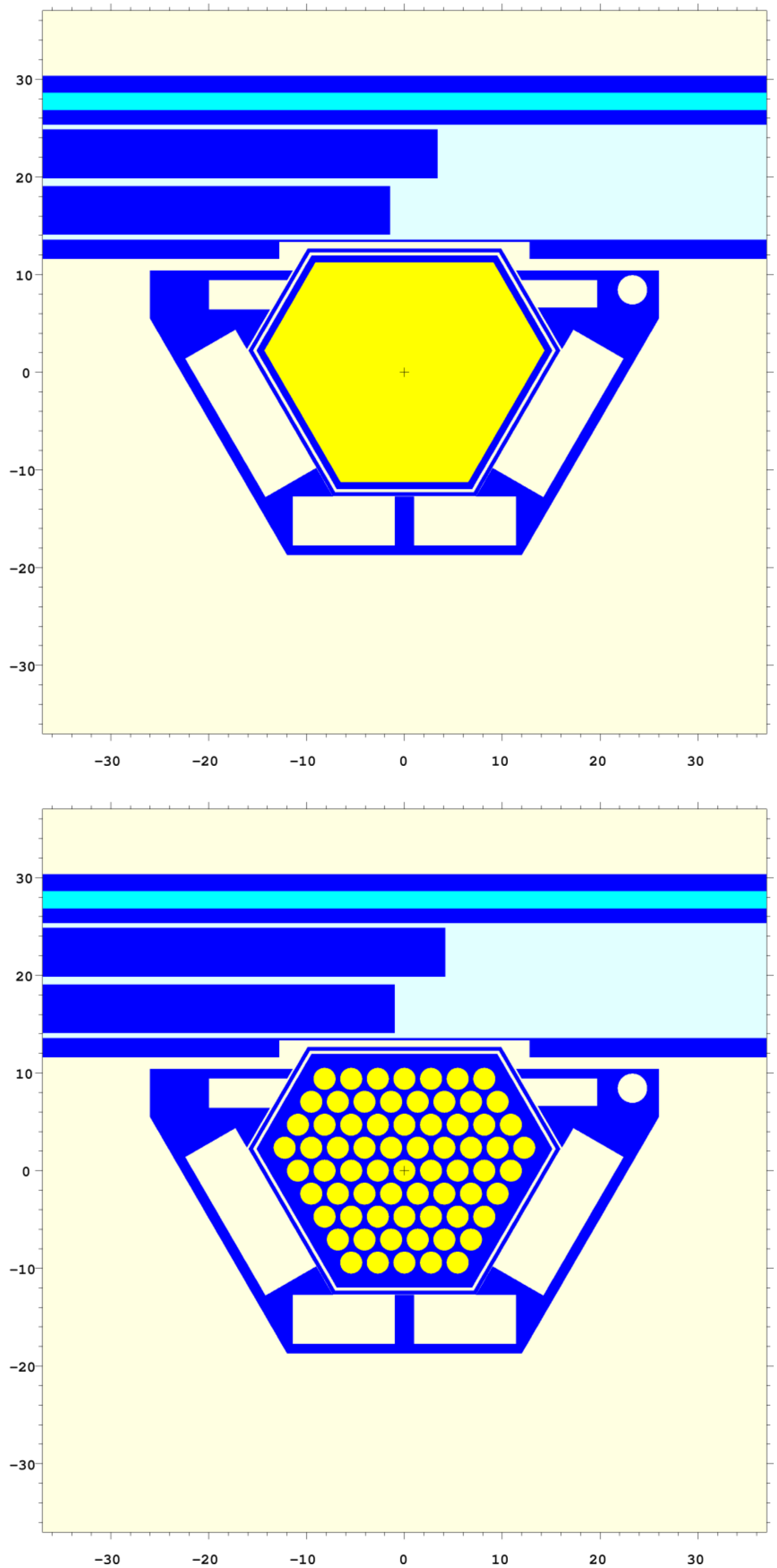


Figure 35. Top plot: pz plane with origin at (0, 0, -57) cm and 37 cm extent; bottom plot: pz plane with origin at (0, 0, -55) cm and 37 cm extent

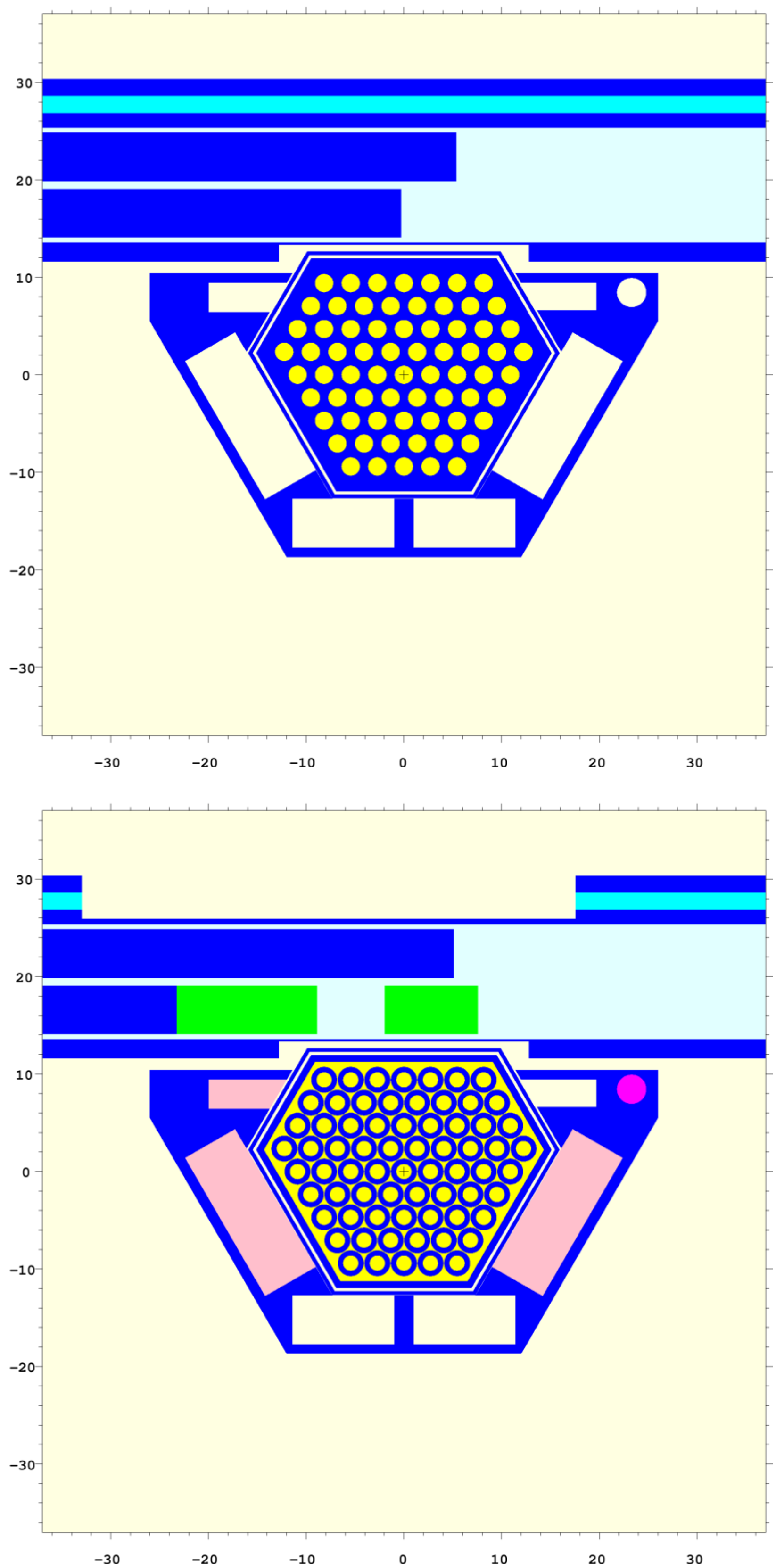


Figure 36. Top plot: pz plane with origin at (0, 0, -52) cm and 37 cm extent; bottom plot: pz plane with origin at (0, 0, -28) cm and 37 cm extent

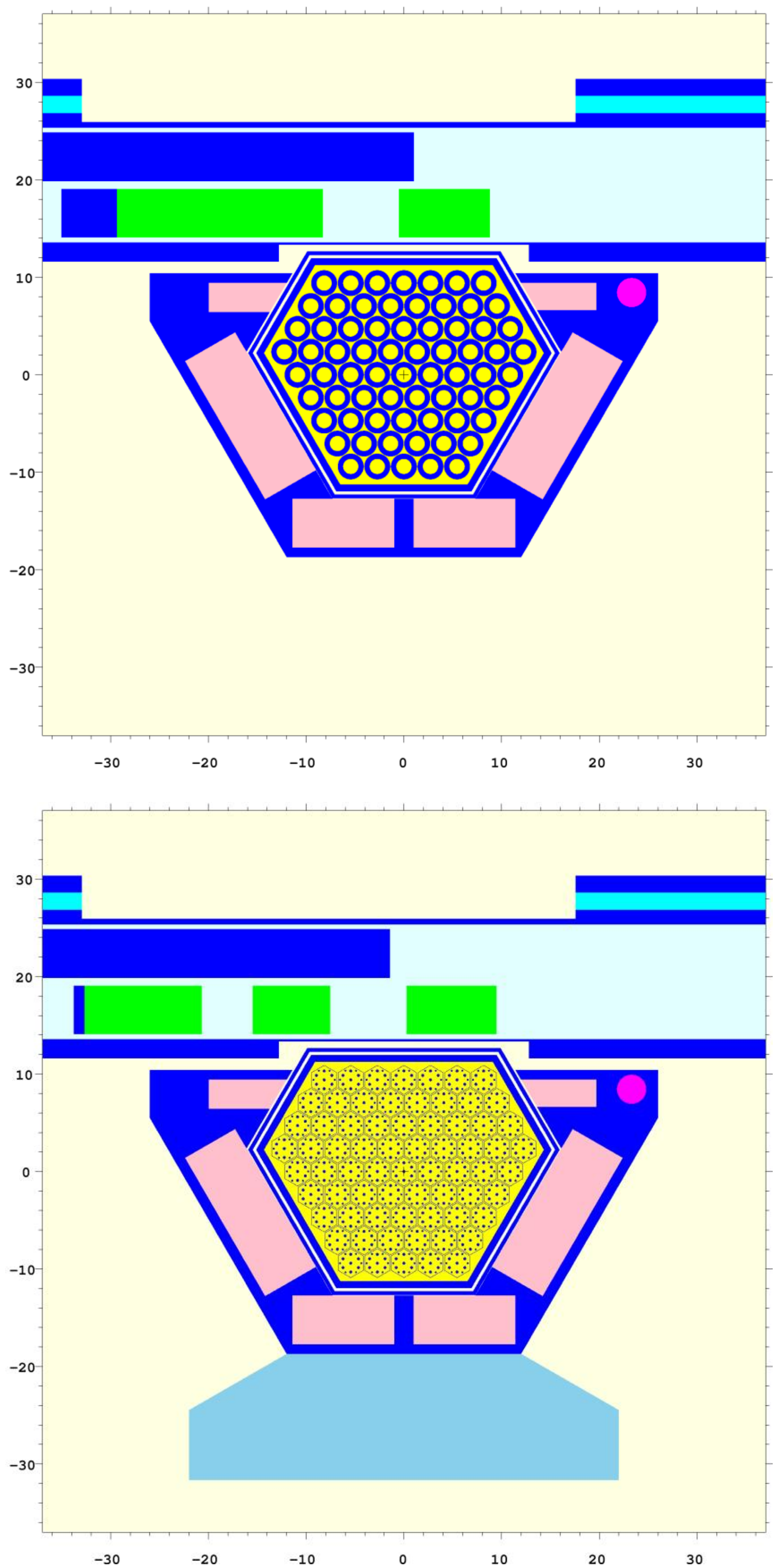


Figure 37. Top plot: pz plane with origin at (0, 0, -25) cm and 37 cm extent; bottom plot: pz plane with origin at (0, 0, -23.4) cm and 37 cm extent

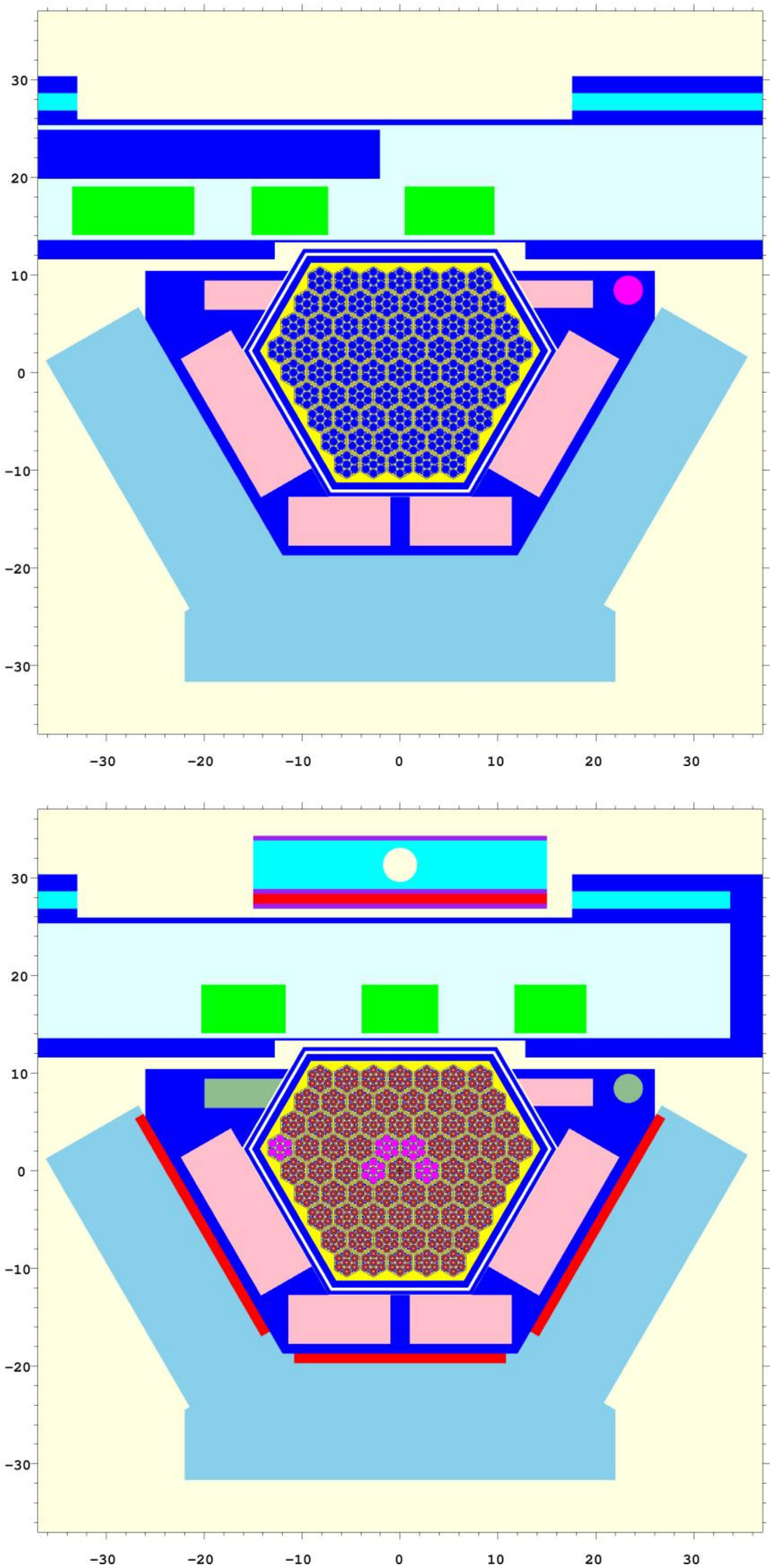


Figure 38. Top plot: pz plane with origin at (0, 0, -23) cm and 37 cm extent; bottom plot: pz plane with origin at (0, 0, 0) cm and 37 cm extent

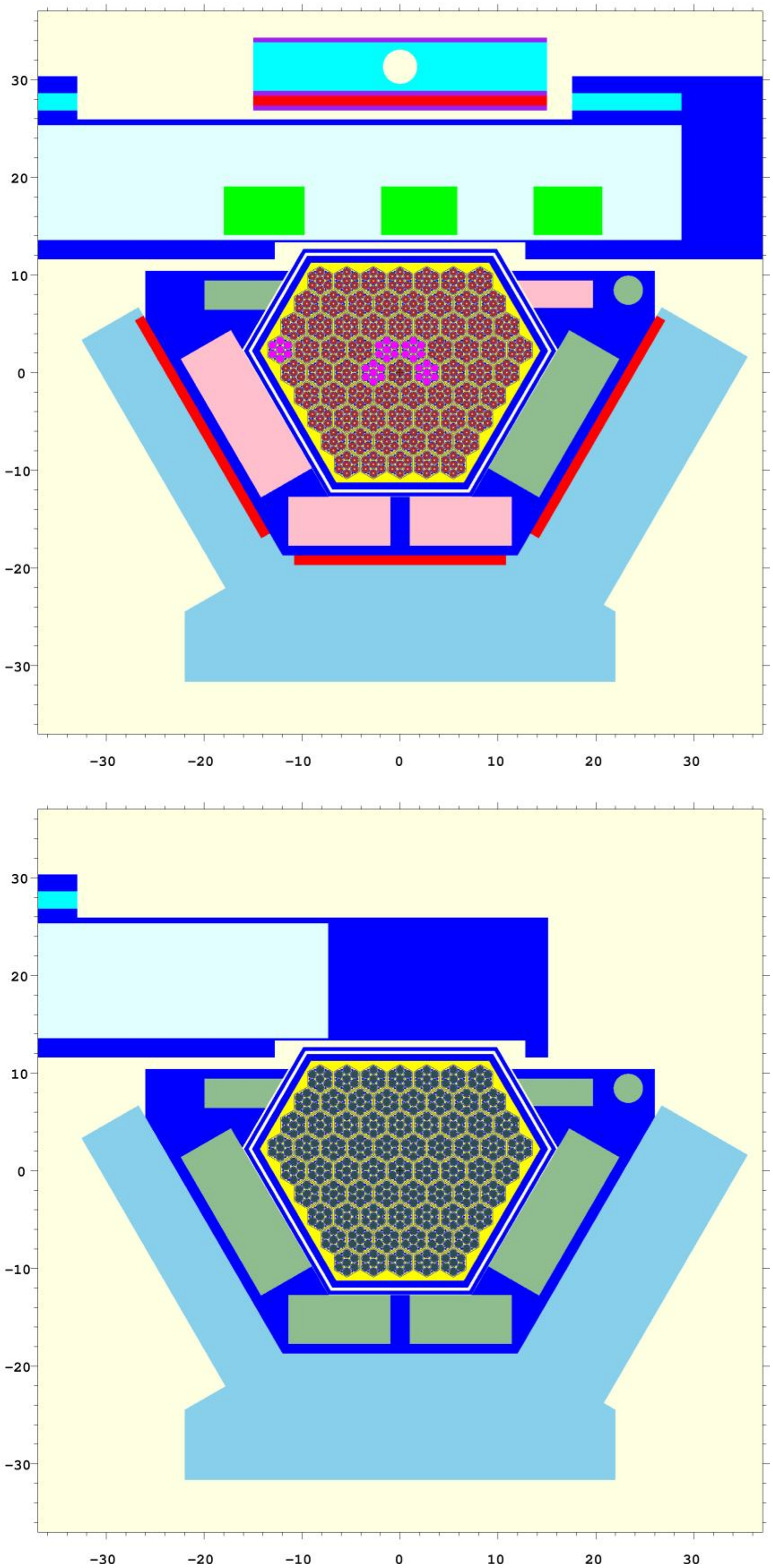


Figure 39. Top plot: pz plane with origin at (0, 0, 4) cm and 37 cm extent; bottom plot: pz plane with origin at (0, 0, 23) cm and 37 cm extent

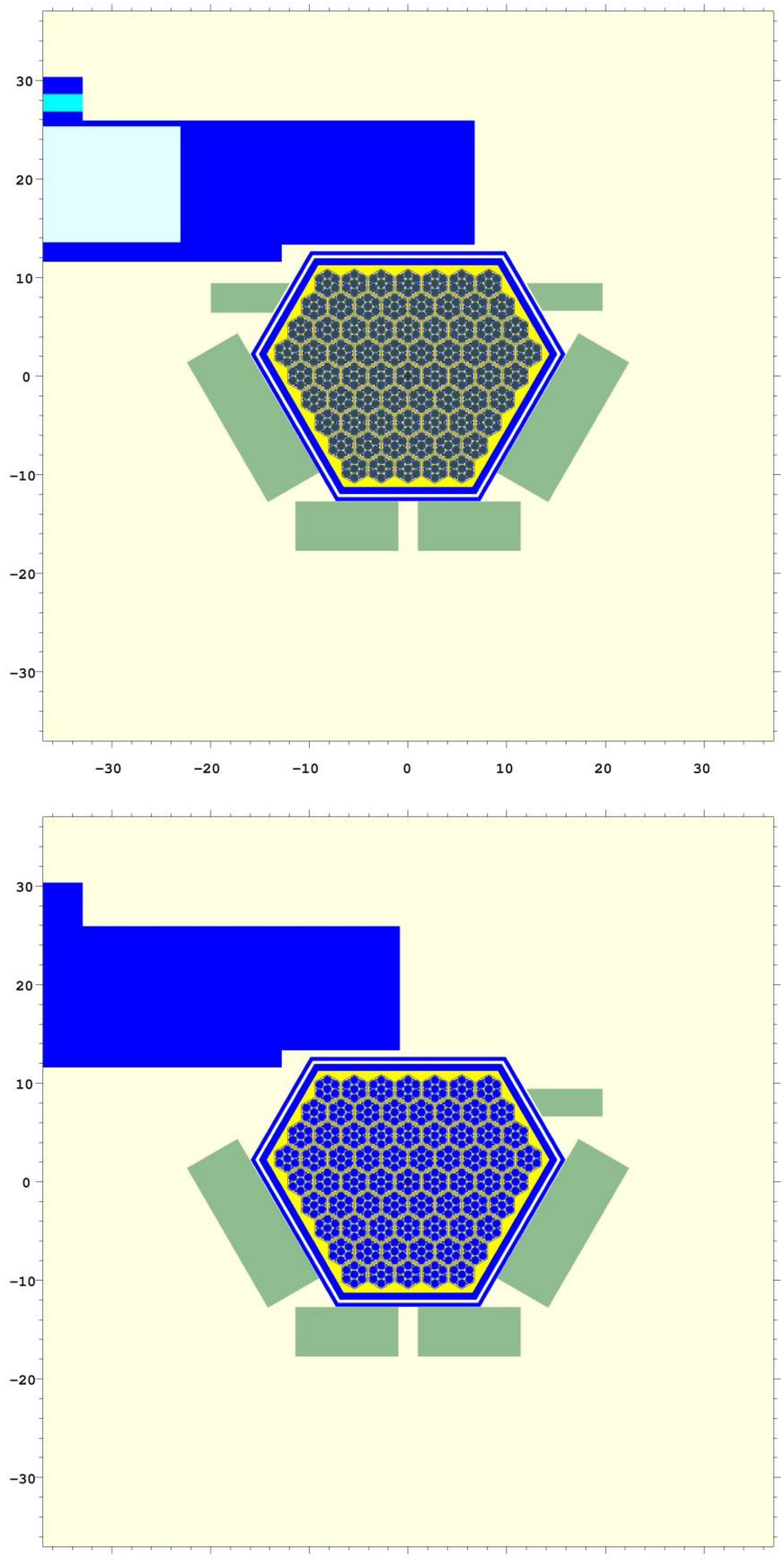


Figure 40. Top plot: pz plane with origin at (0, 0, 27) cm and 37 cm extent; bottom plot: pz plane with origin at (0, 0, 30) cm and 37 cm extent

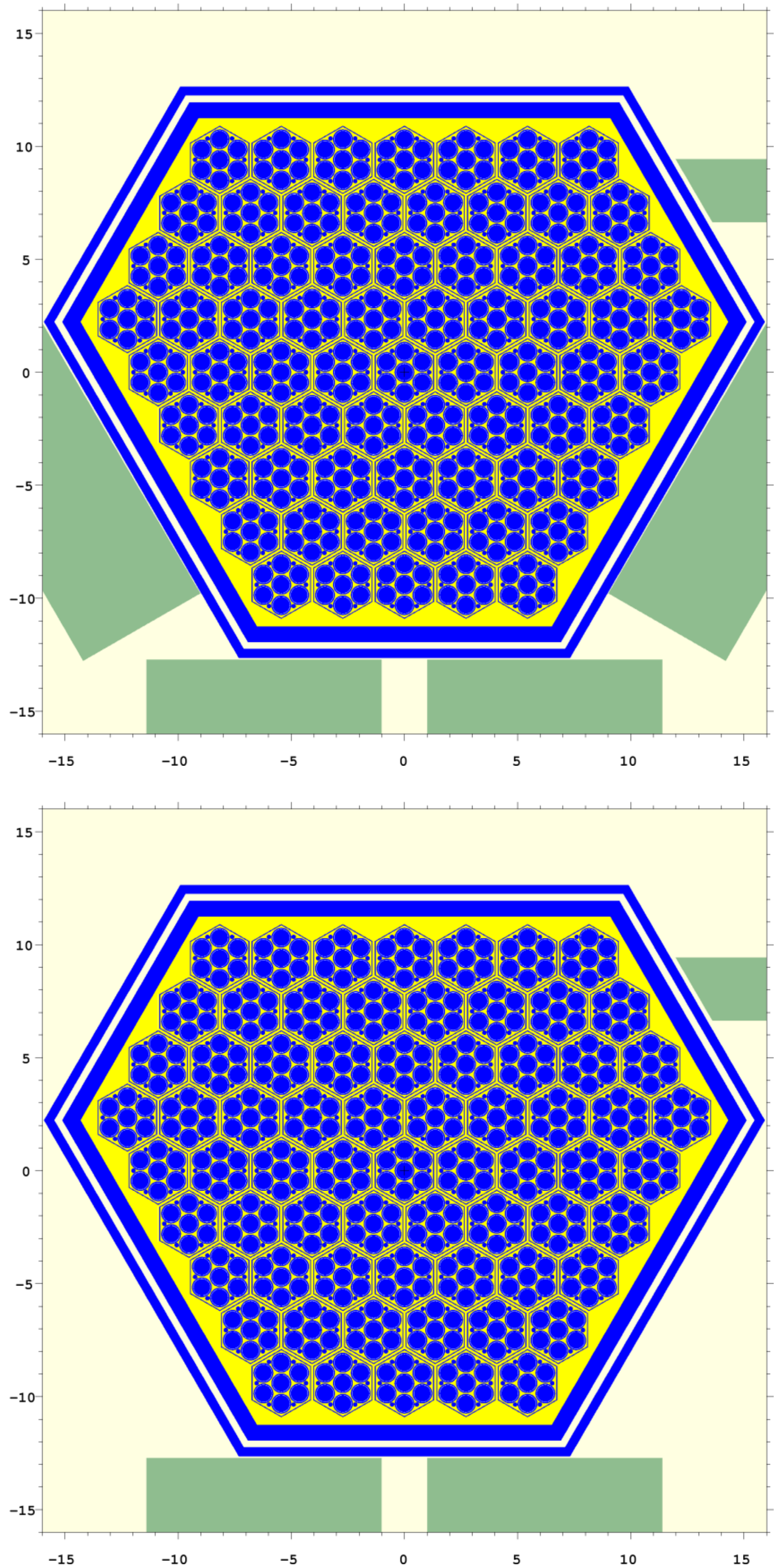


Figure 41. Top plot: pz plane with origin at (0, 0, 35) cm and 16 cm extent; bottom plot: pz plane with origin at (0, 0, 50) cm and 16 cm extent

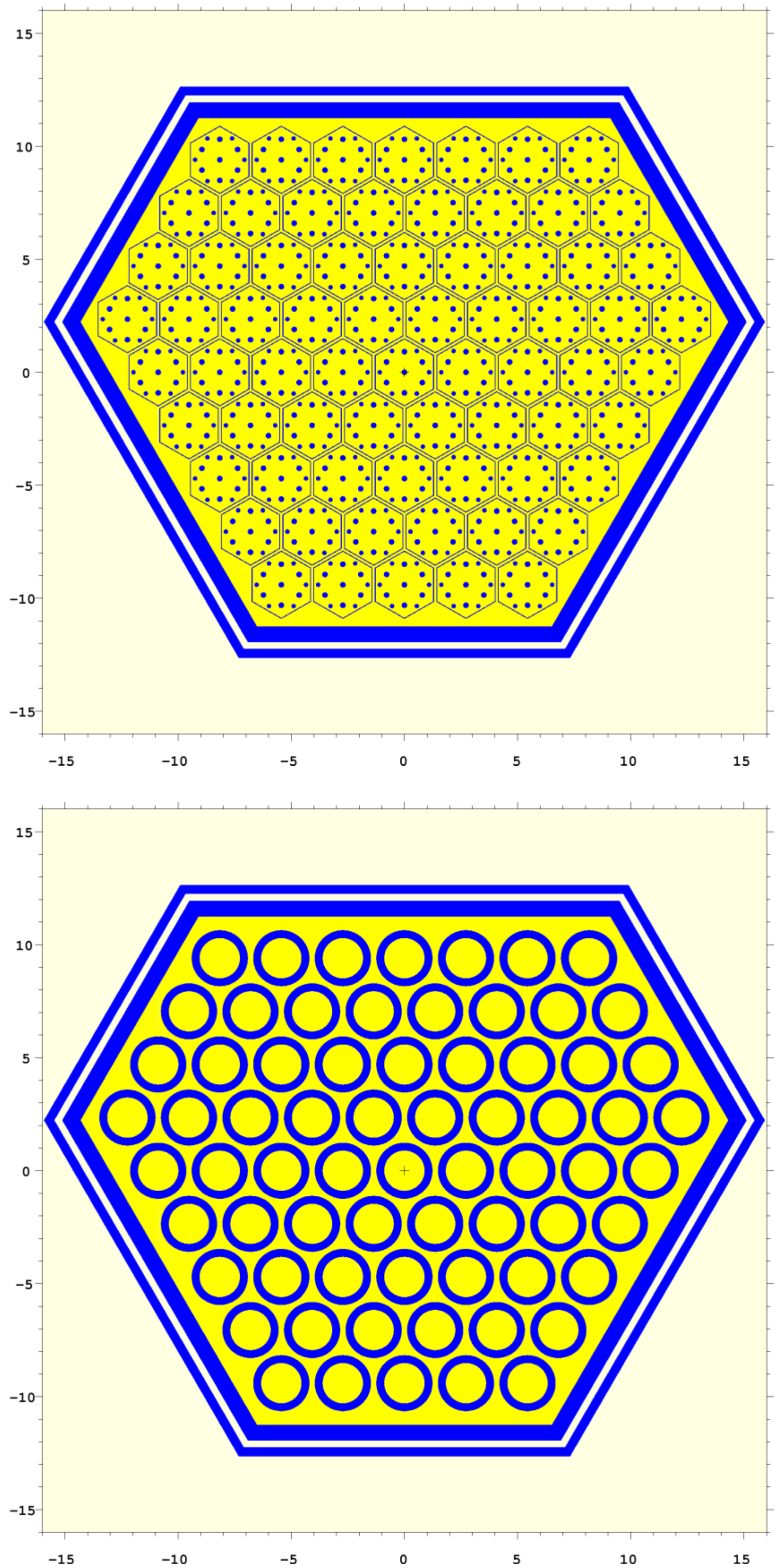
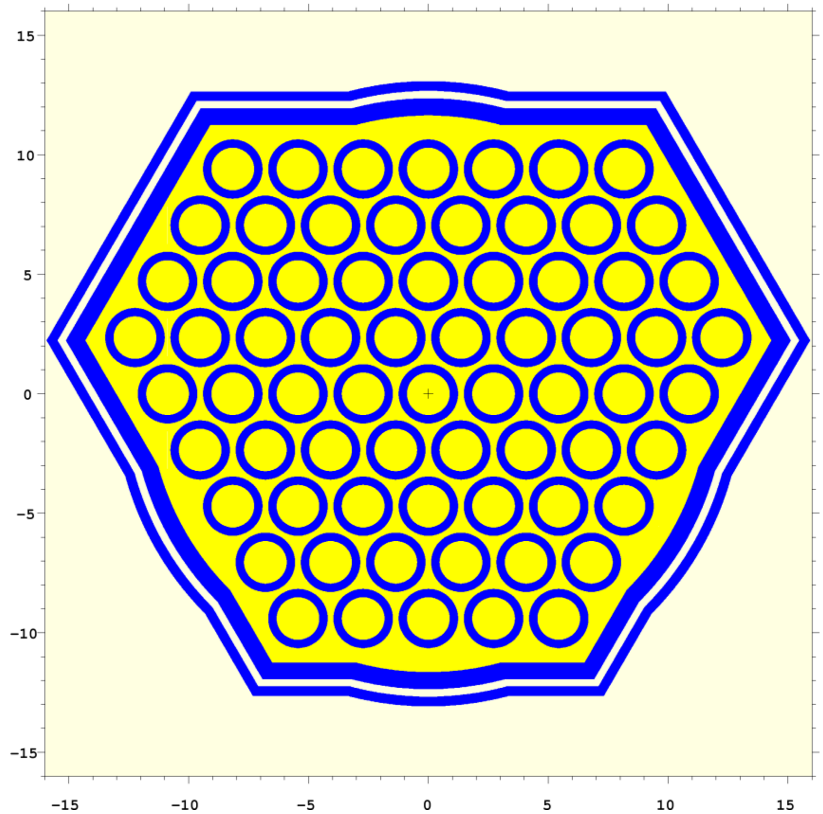


Figure 42. Top plot: pz plane with origin at (0, 0, 52) cm and 16 cm extent; bottom plot: pz plane with origin at (0, 0, 55) cm and 16 cm extent



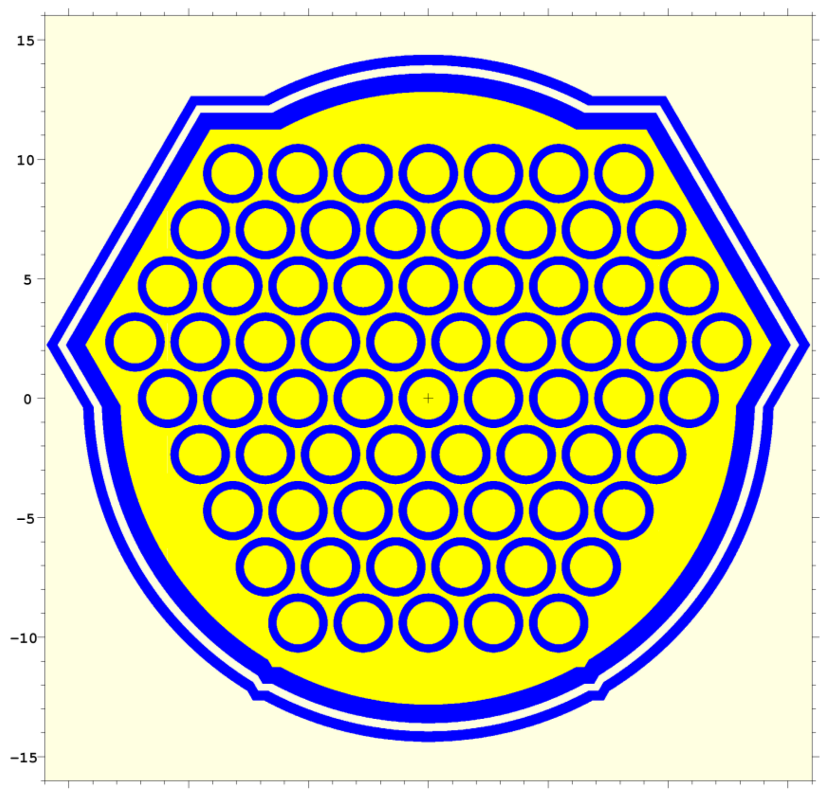


Figure 43. Top plot: pz plane with origin at (0, 0, 57) cm and 16 cm extent; bottom plot: pz plane with origin at (0, 0, 60) cm and 16 cm extent

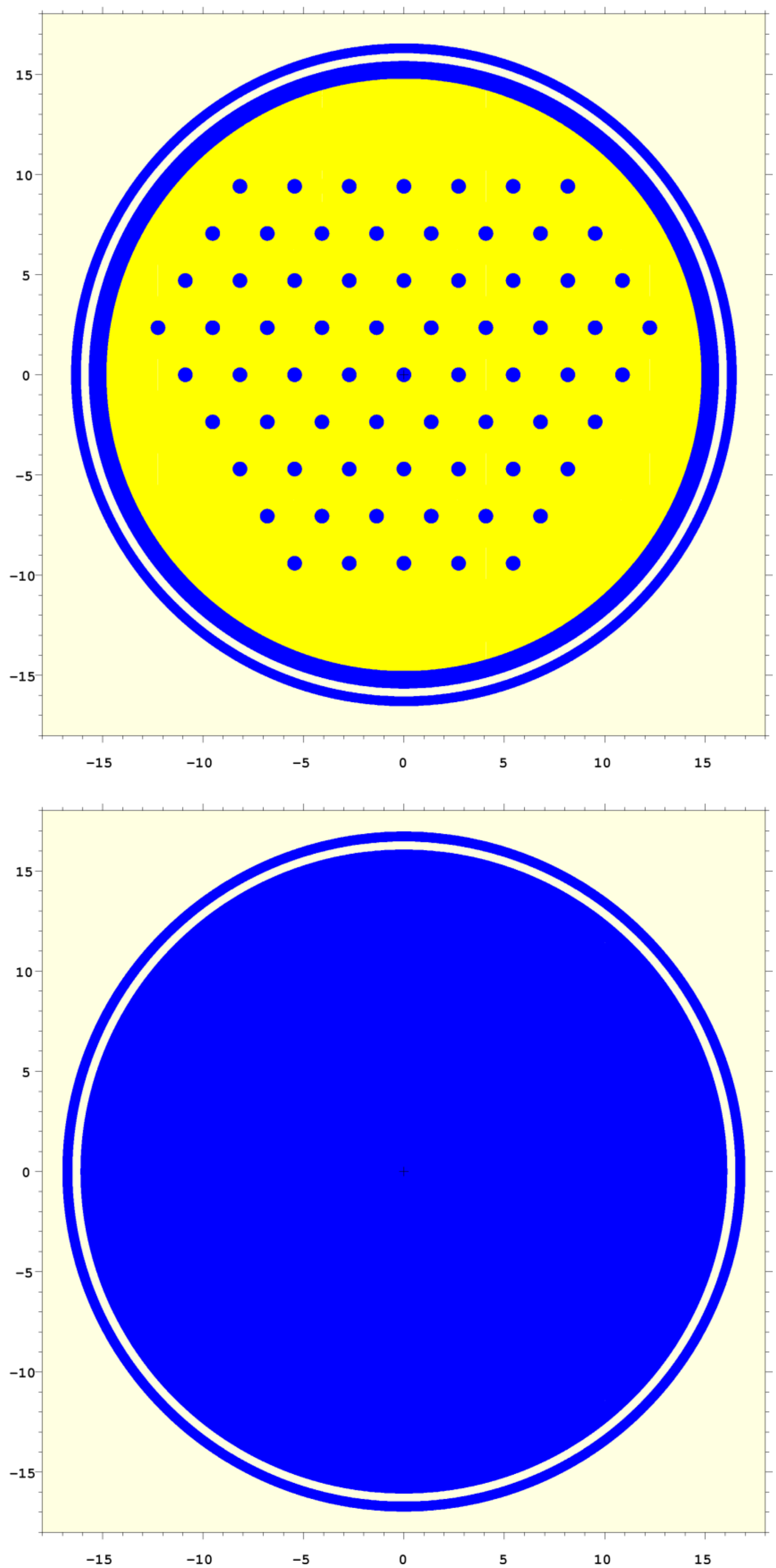


Figure 44. Top plot: pz plane with origin at (0, 0, 65) cm and 18 cm extent; bottom plot: pz plane with origin at (0, 0, 70) cm and 18 cm extent

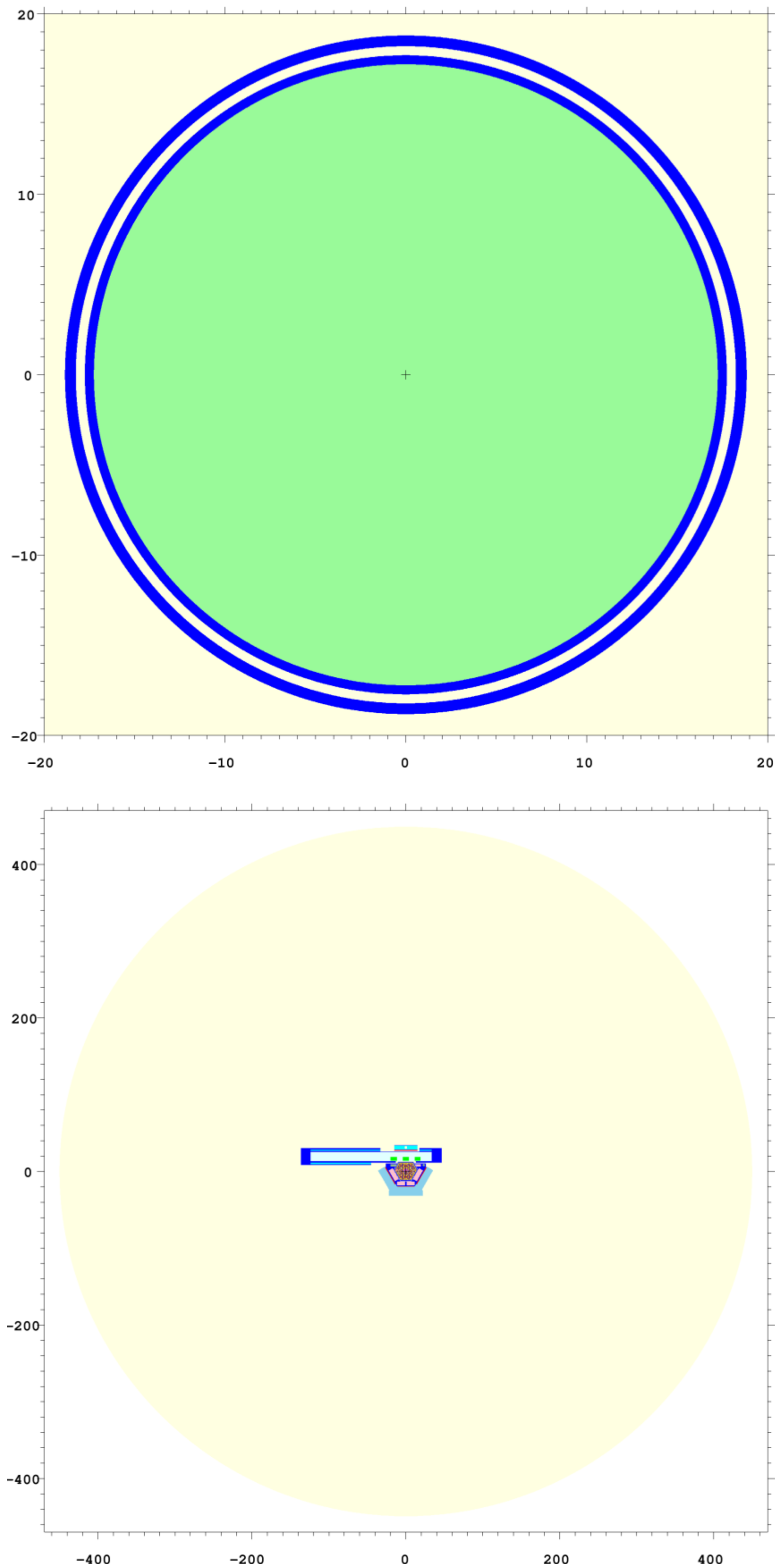


Figure 45. Top plot: pz plane with origin at (0, 0, 100) cm and 20 cm extent; bottom plot: pz plane with origin at (0, 0, 0) cm and 470 cm extent

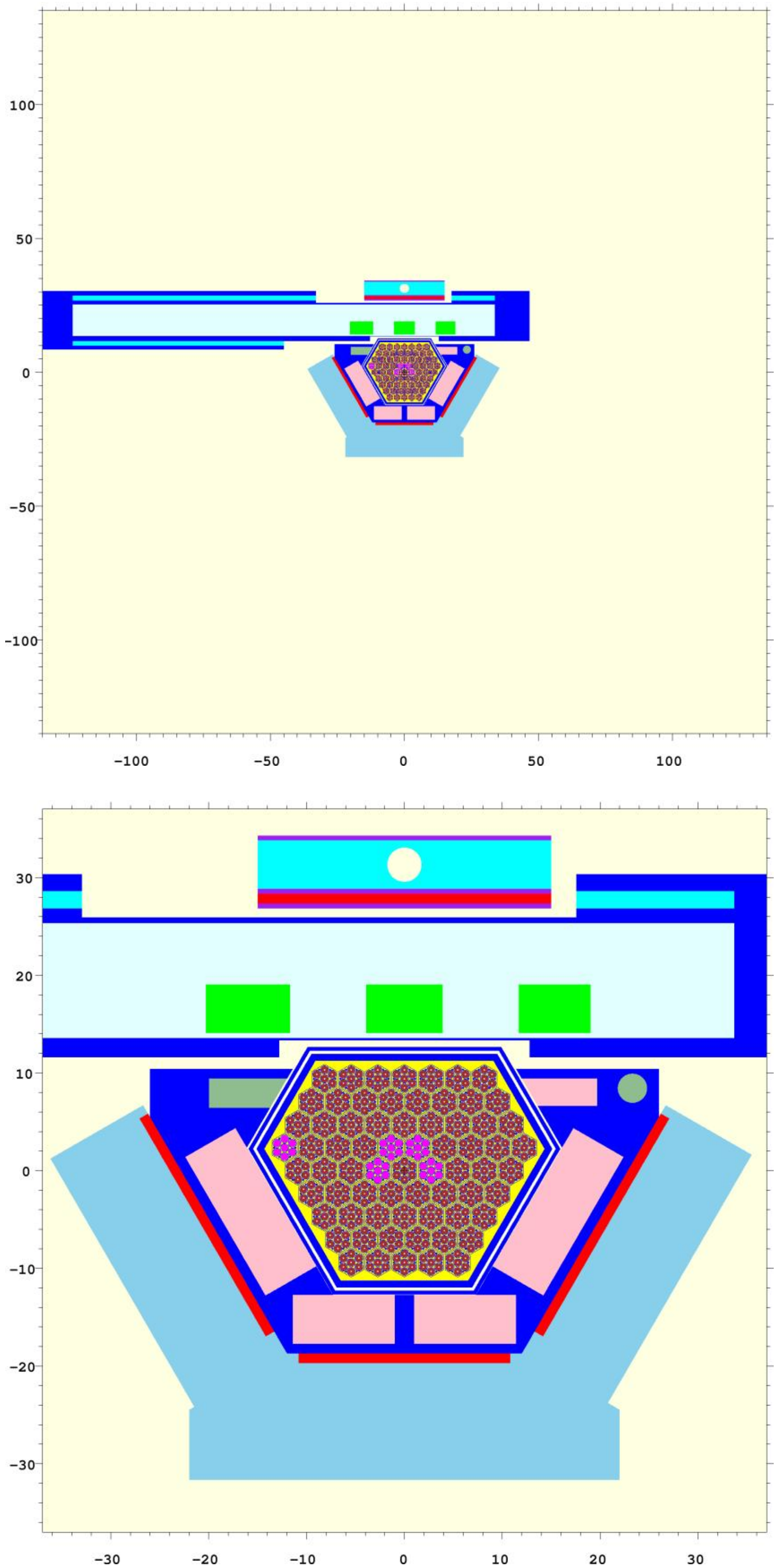


Figure 46. Top plot: pz plane with origin at (0, 0, 0) cm and 135 cm extent; bottom plot: pz plane with origin at (0, 0, 0) cm and 37 cm extent

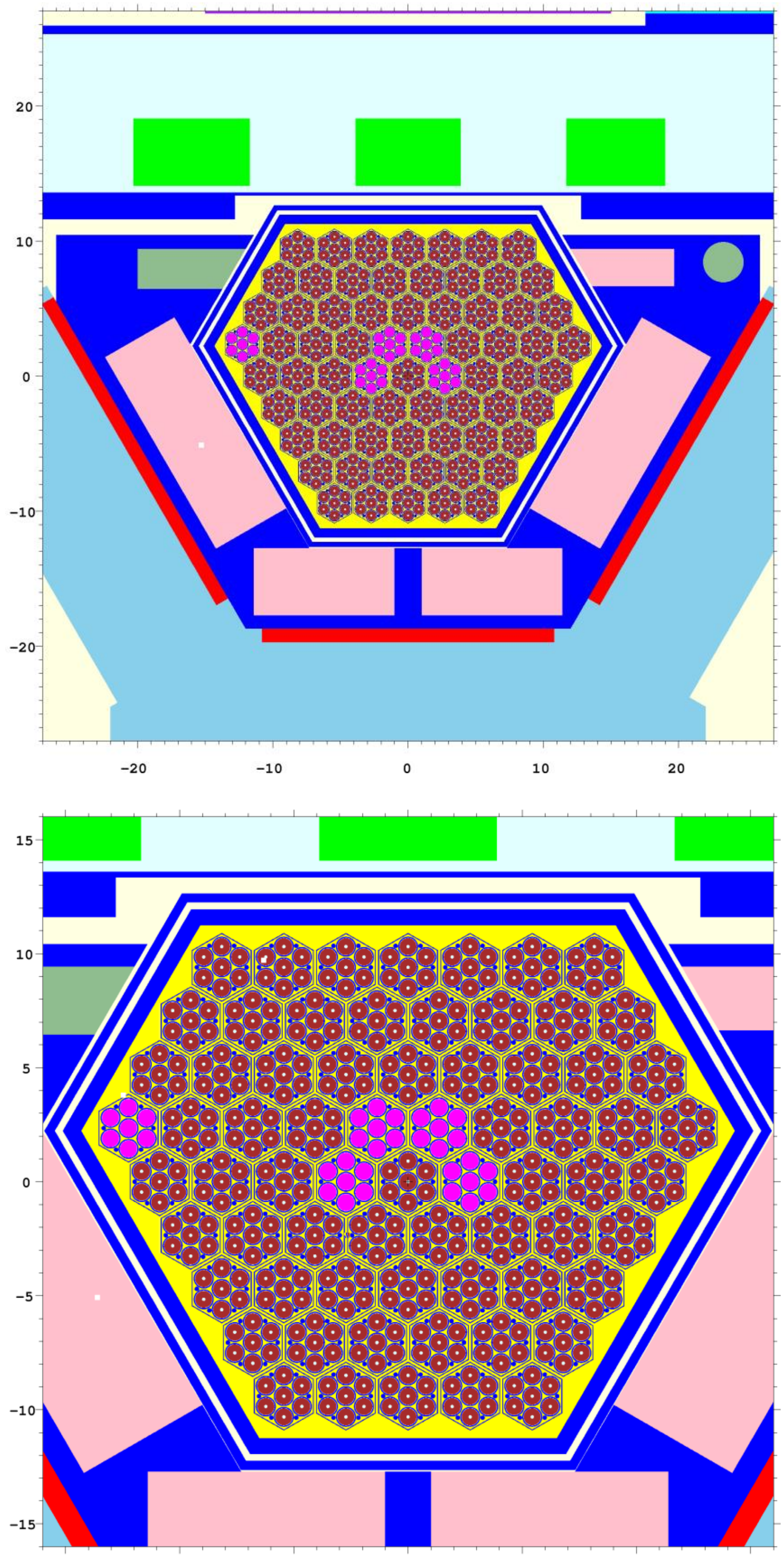


Figure 47. Top plot: pz plane with origin at (0, 0, 0) cm and 27 cm extent; bottom plot: pz plane with origin at (0,0, 0) cm and 16 cm extent

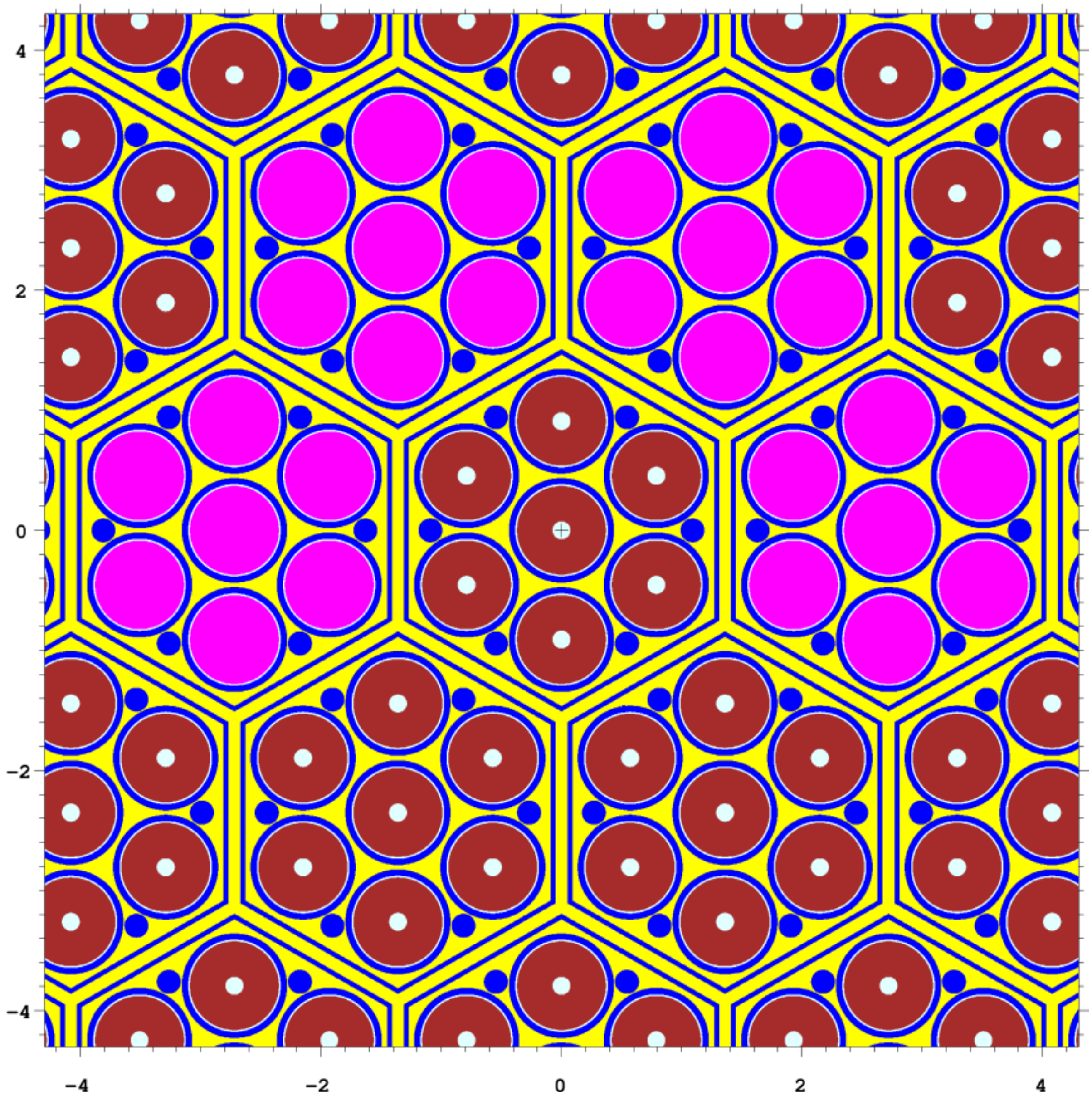


Figure 48. pz plane with origin at (0, 0, 0) cm and 4.3 cm extent

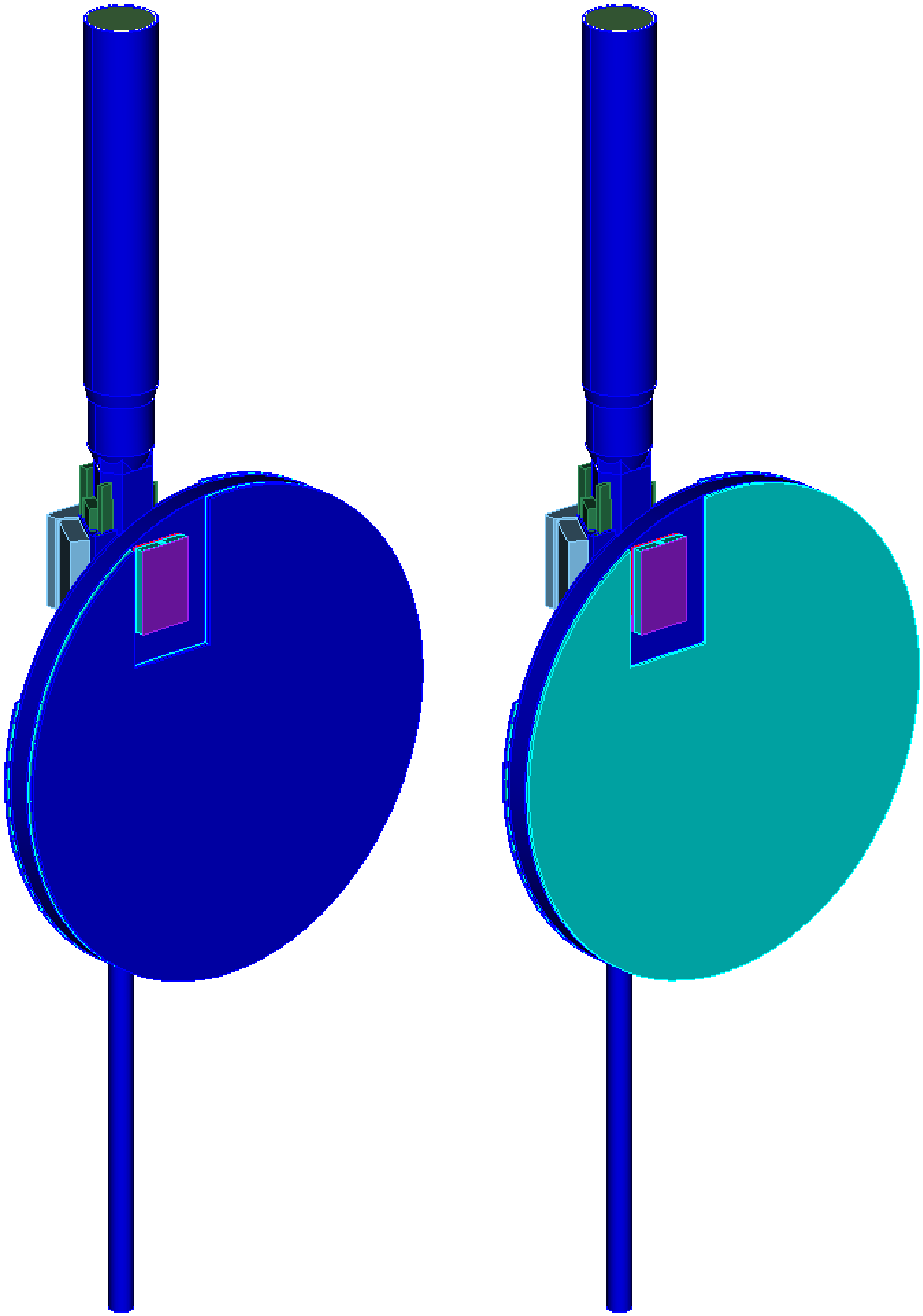


Figure 49. Left plot: overview of the IBR-2M reactor; right plot: overview of the reactor without the steel layer of the outer cover (jacket) in the rotating reflectors

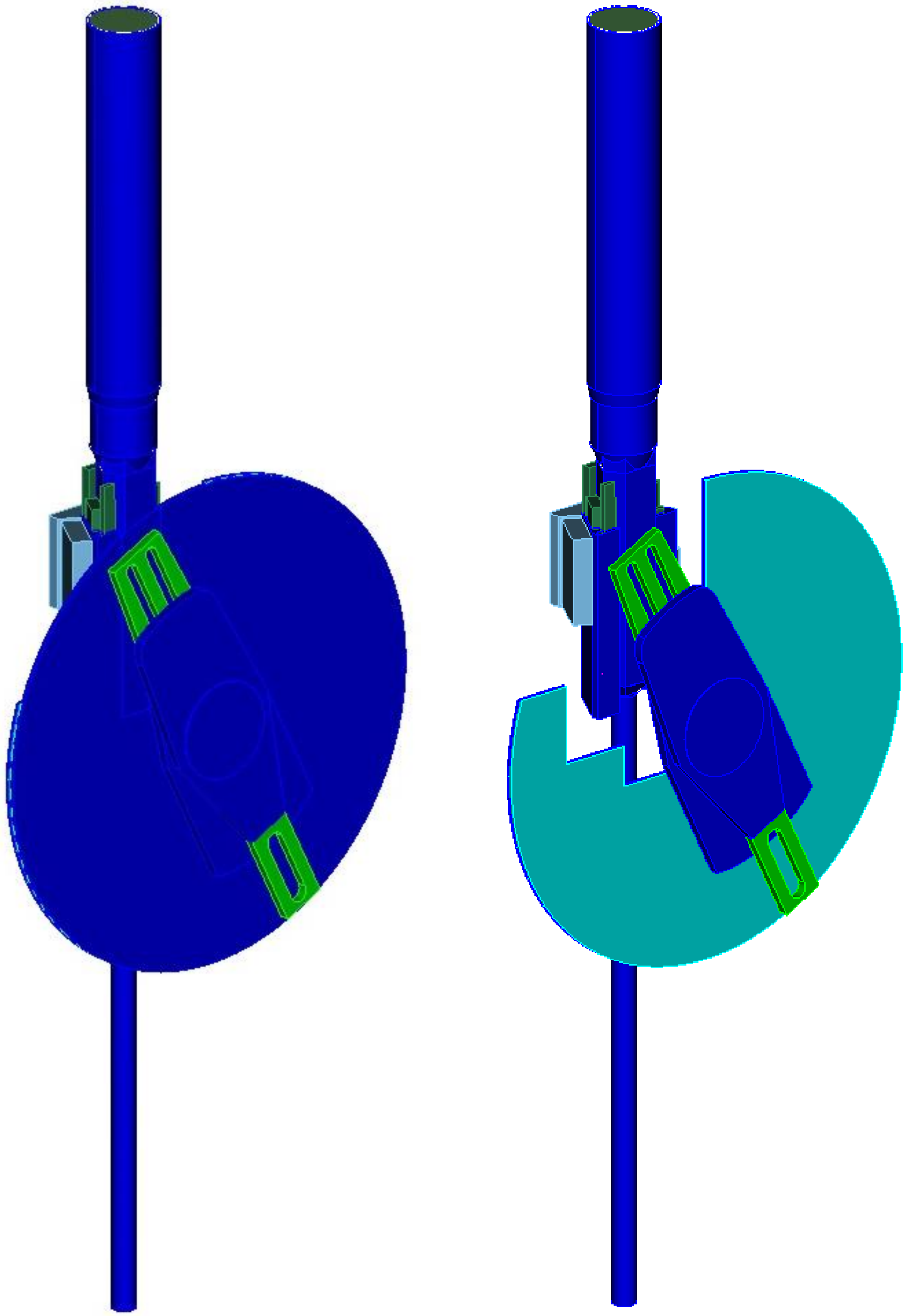


Figure 50. Left plot: overview of the reactor without the outer jacket and helium gas in the rotating reflectors; right plot: overview of the reactor without the outer jacket, the helium gas, and the steel layer (next to the rotating blades) of the inner jacket in the rotating reflectors

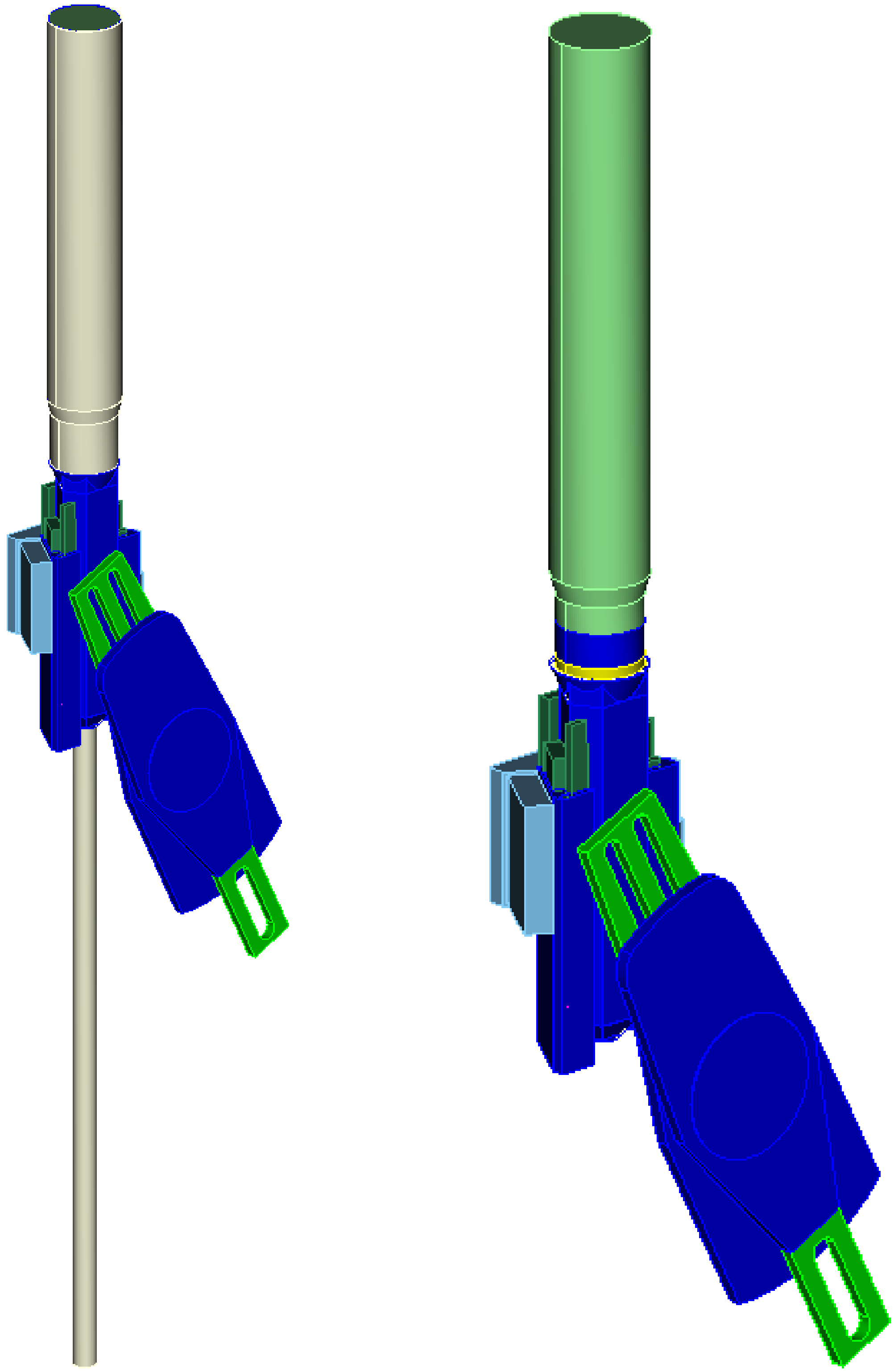


Figure 51. Left plot: overview of the reactor without the outer steel layer of the vessel at the top and the bottom and without the rotating reflectors casing and the jackets; right plot: overview of the reactor without the bottom part, the vessel in the top, and the rotating reflectors casing and the jackets

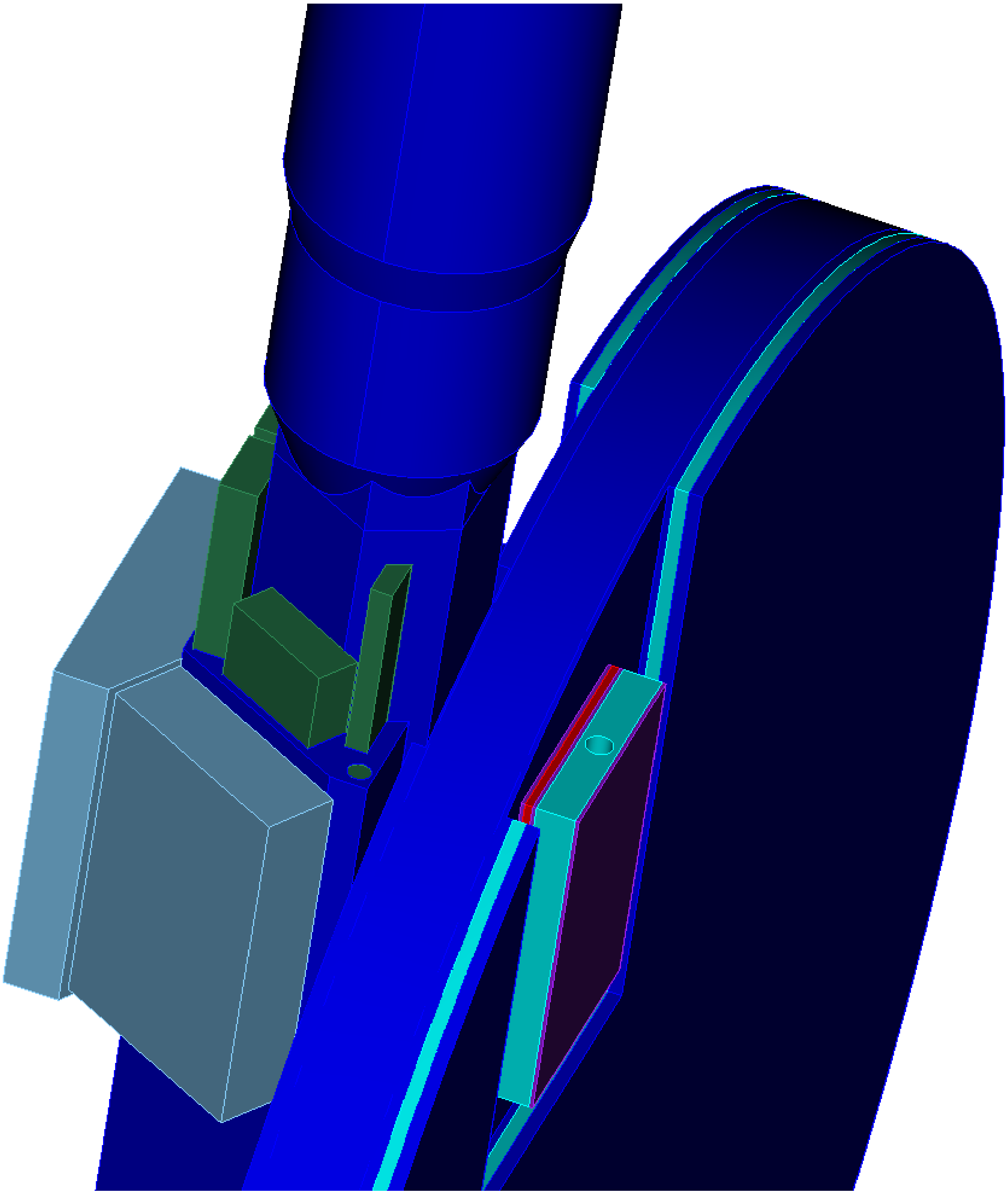


Figure 52. Zoom of the top part of the reactor

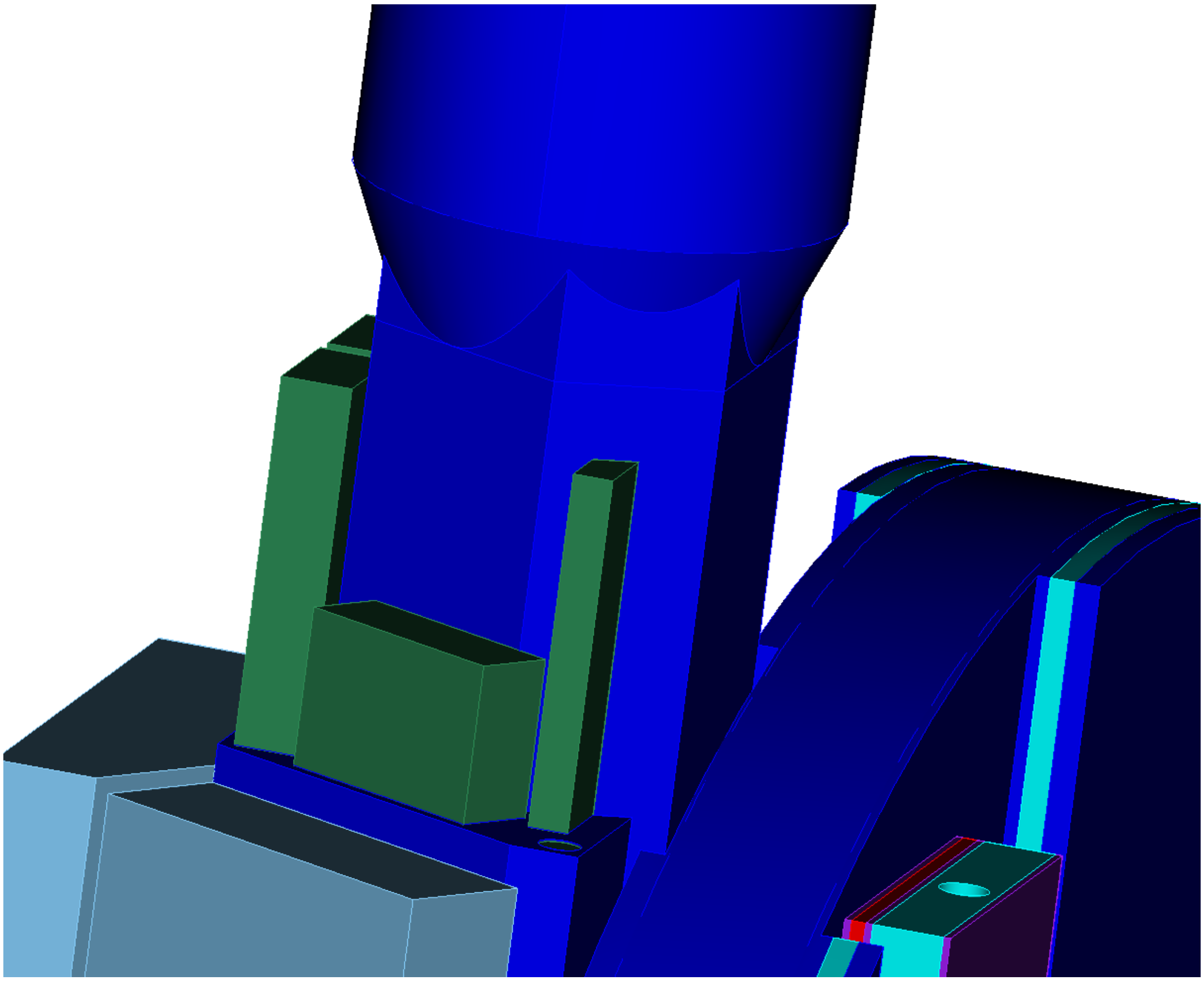


Figure 53. Zoom of the transition from the hexagonal to the circular shape of the reactor vessel at the top

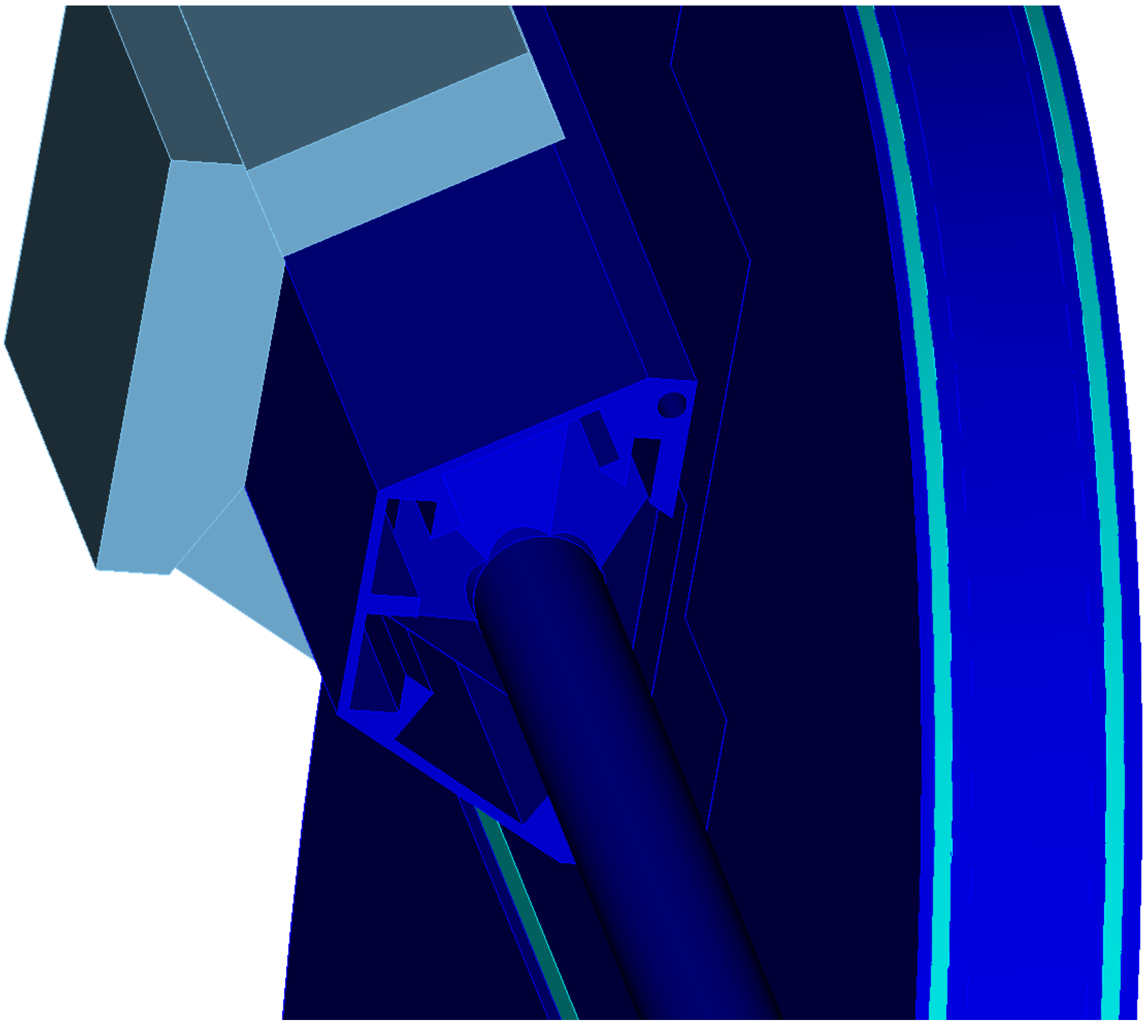


Figure 54. Zoom of the bottom part of the reactor

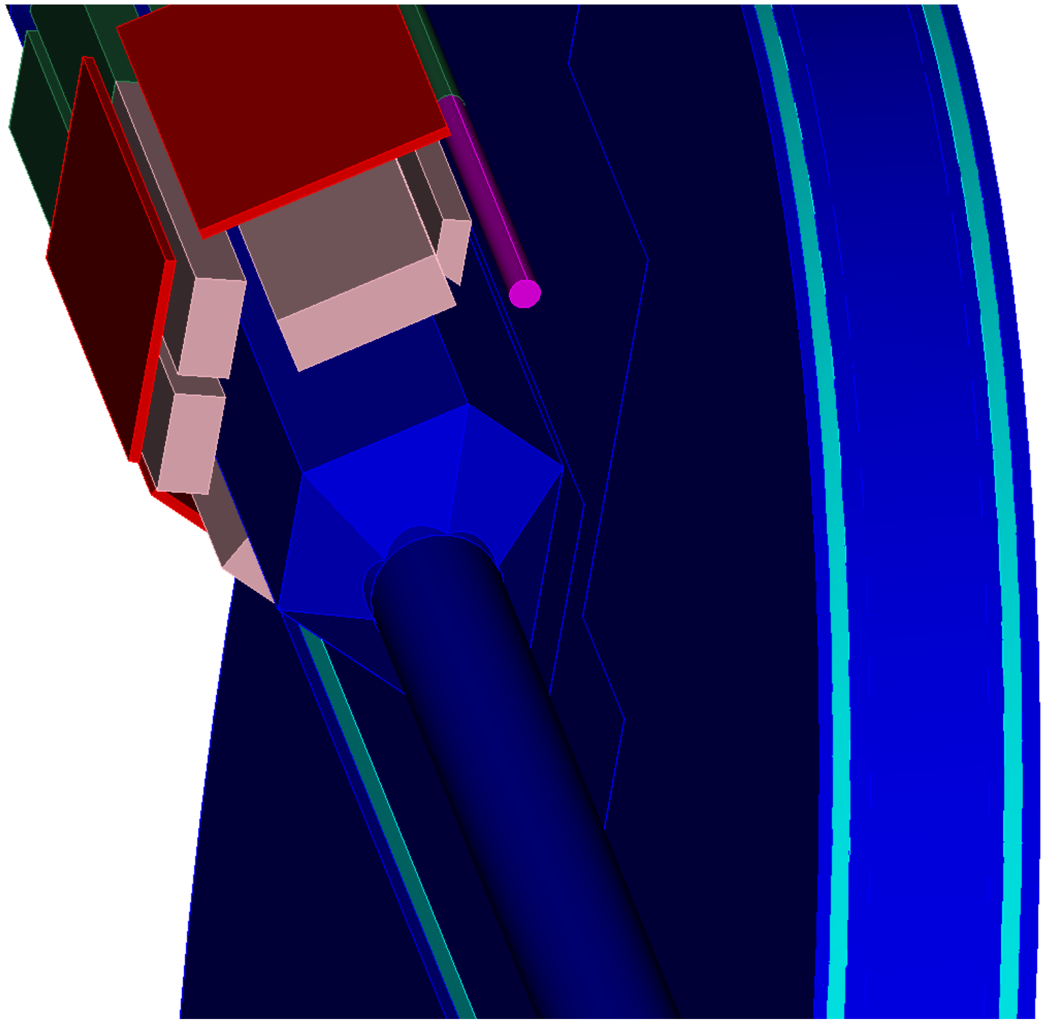


Figure 55. Zoom of the transition from the hexagonal to the circular shape of the reactor vessel at the bottom; the water and the aluminum blocks of the cooling system of the stationary reflector are hidden

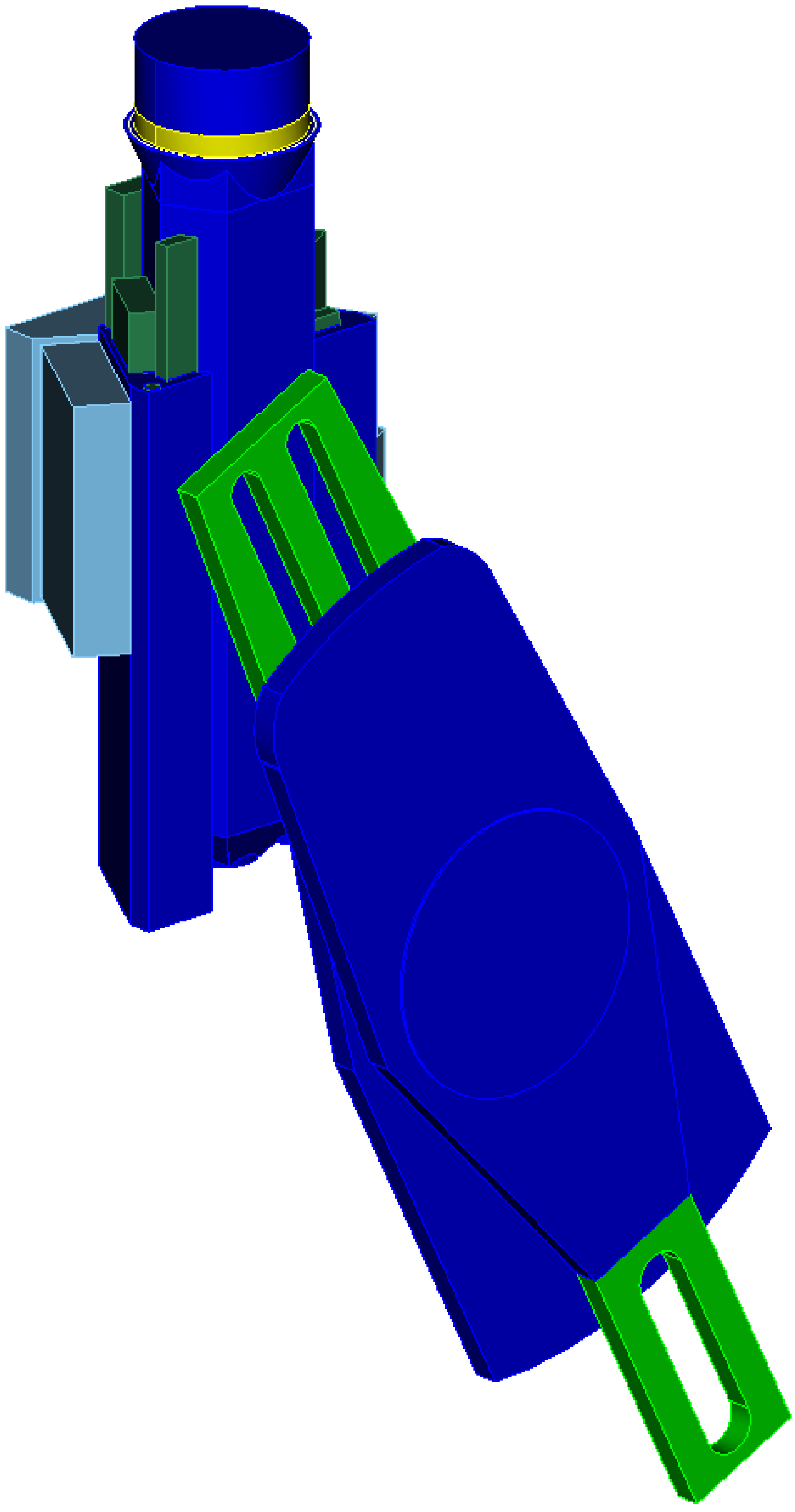


Figure 56. Overview of the reactor without the top and the bottom parts and the rotating reflectors casing and the jackets

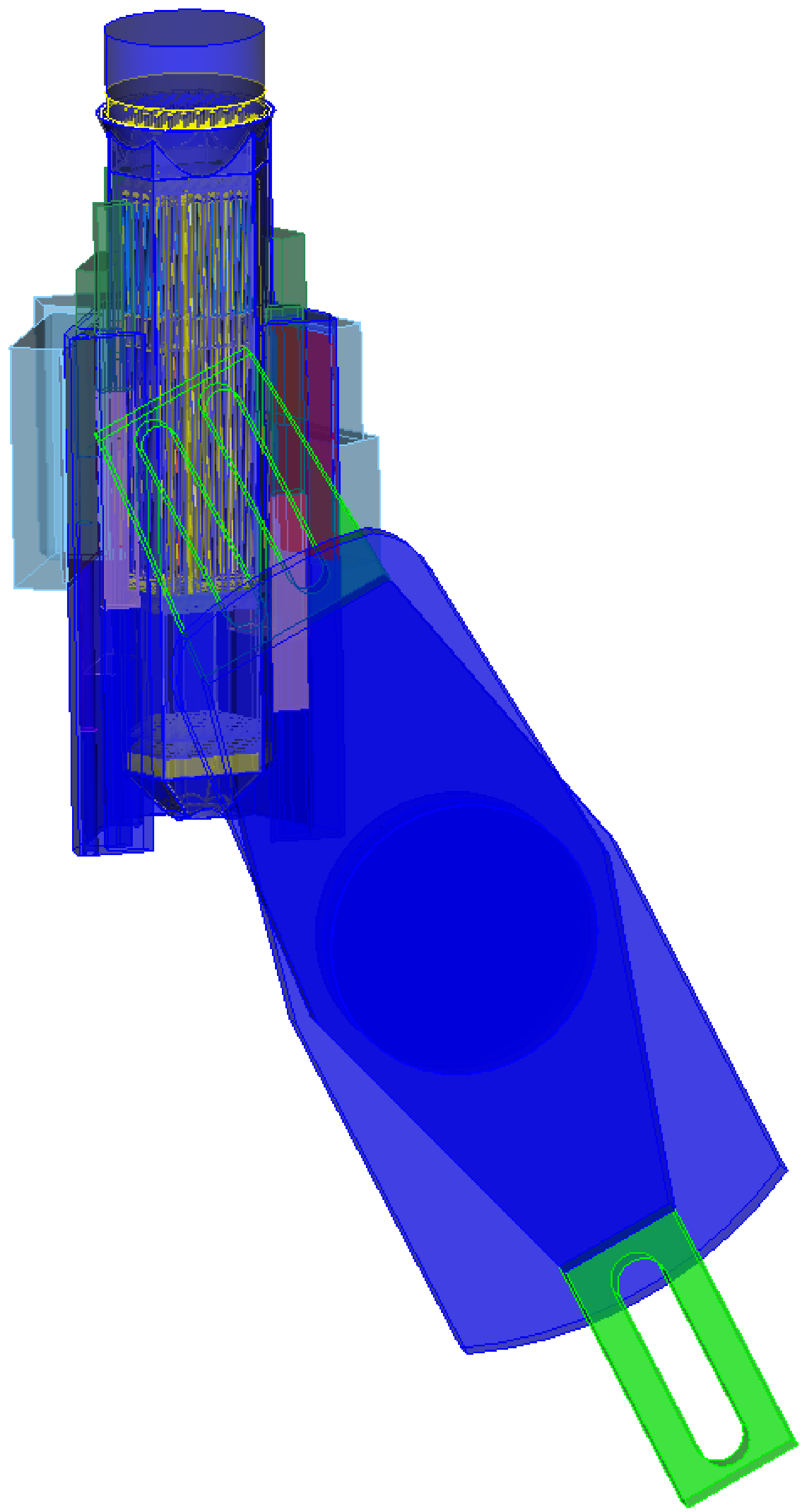


Figure 57. Overview of the reactor without the top and the bottom parts and the rotating reflectors casing and the jackets with a semi-transparent filter

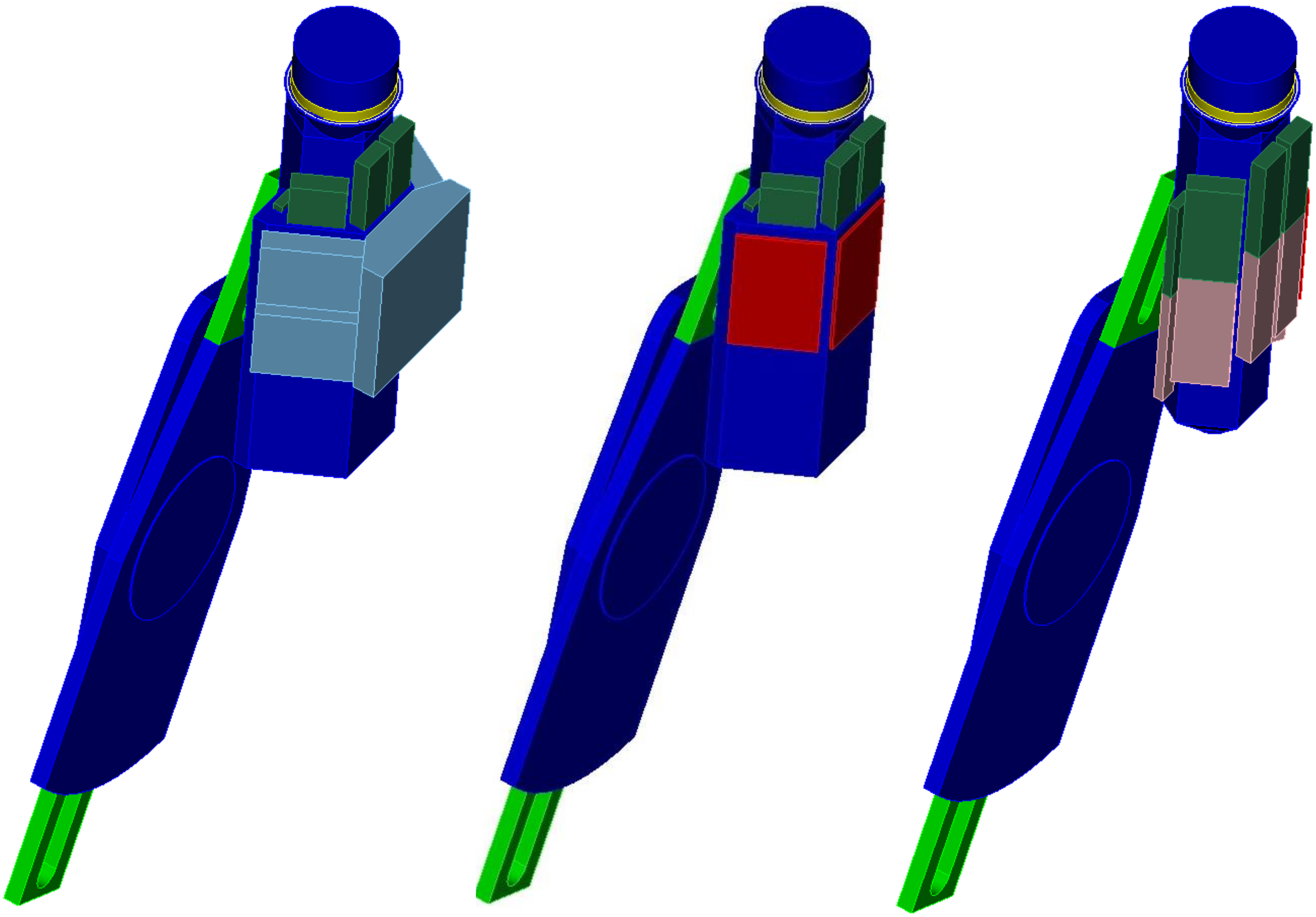


Figure 58. Left plot: overview of the reactor without the top and the bottom parts, and the rotating reflectors casing and the jackets. Central plot: overview of the reactor without the top and the bottom parts, the rotating reflectors casing and the jackets, and the water and the aluminum blocks of the cooling system of the stationary reflector. Right plot: overview of the reactor without the top and the bottom parts, the rotating reflectors casing and the jackets, the cooling system of the stationary reflector, and the stationary reflector

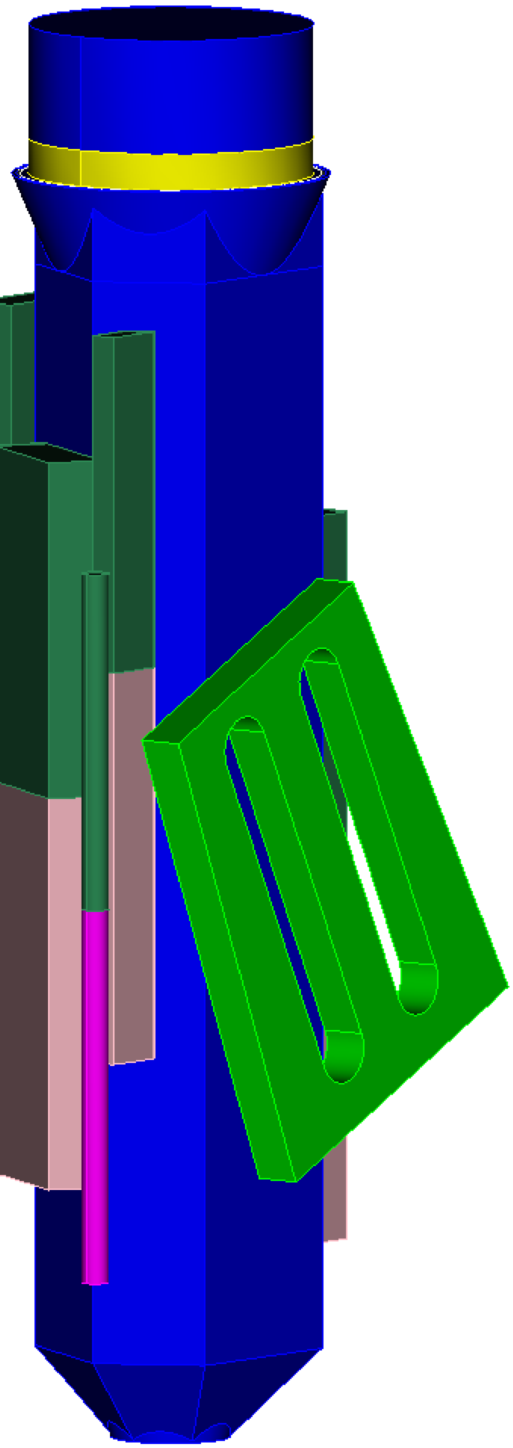


Figure 59. Overview of the reactor without the top and the bottom parts, the rotating reflectors with exception of the two-hole blade, the cooling system of the stationary reflector, and the stationary reflector

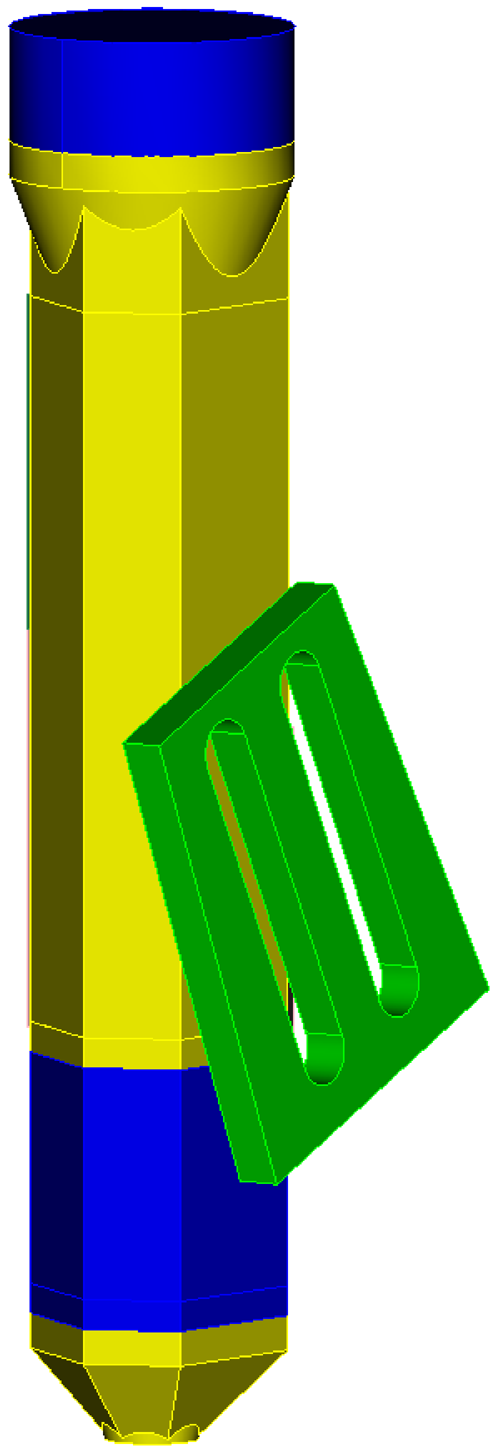


Figure 60. Overview of the reactor without the top and the bottom parts, the rotating reflectors with exception of the two-hole blade, the cooling system of the stationary reflector, the stationary reflector, the control rods, and the vessel

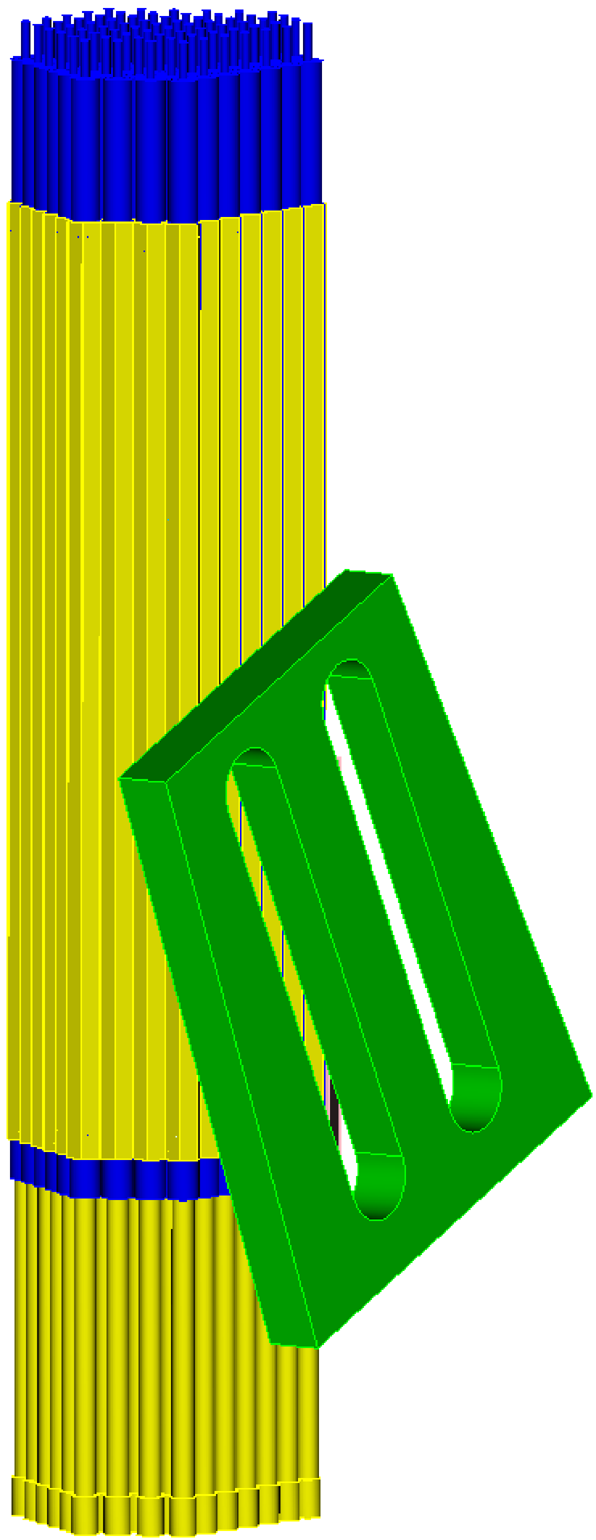


Figure 61. Overview of the fuel assemblies and the two-hole blade; the steel hollow cylinders at the bottom of the fuel assemblies (Figs. 36 and 37 top) are hidden

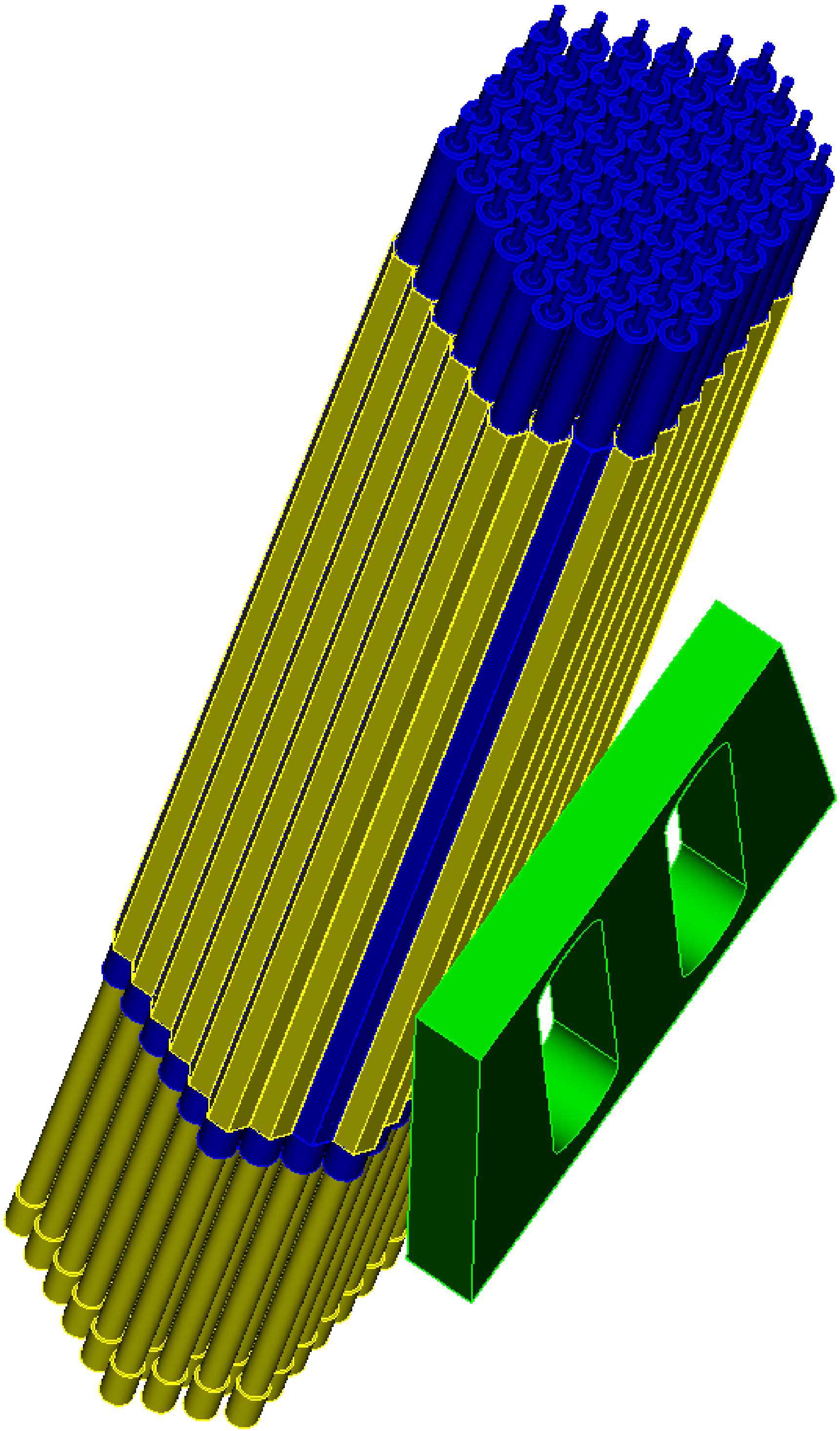


Figure 62. Overview of the fuel assemblies and the two-hole blade reflector; the hollow steel cylinders at the bottom of the fuel assemblies (Figs. 36 and 37 top) and sodium in the frontal assembly (in the fuel rods zone) is hidden

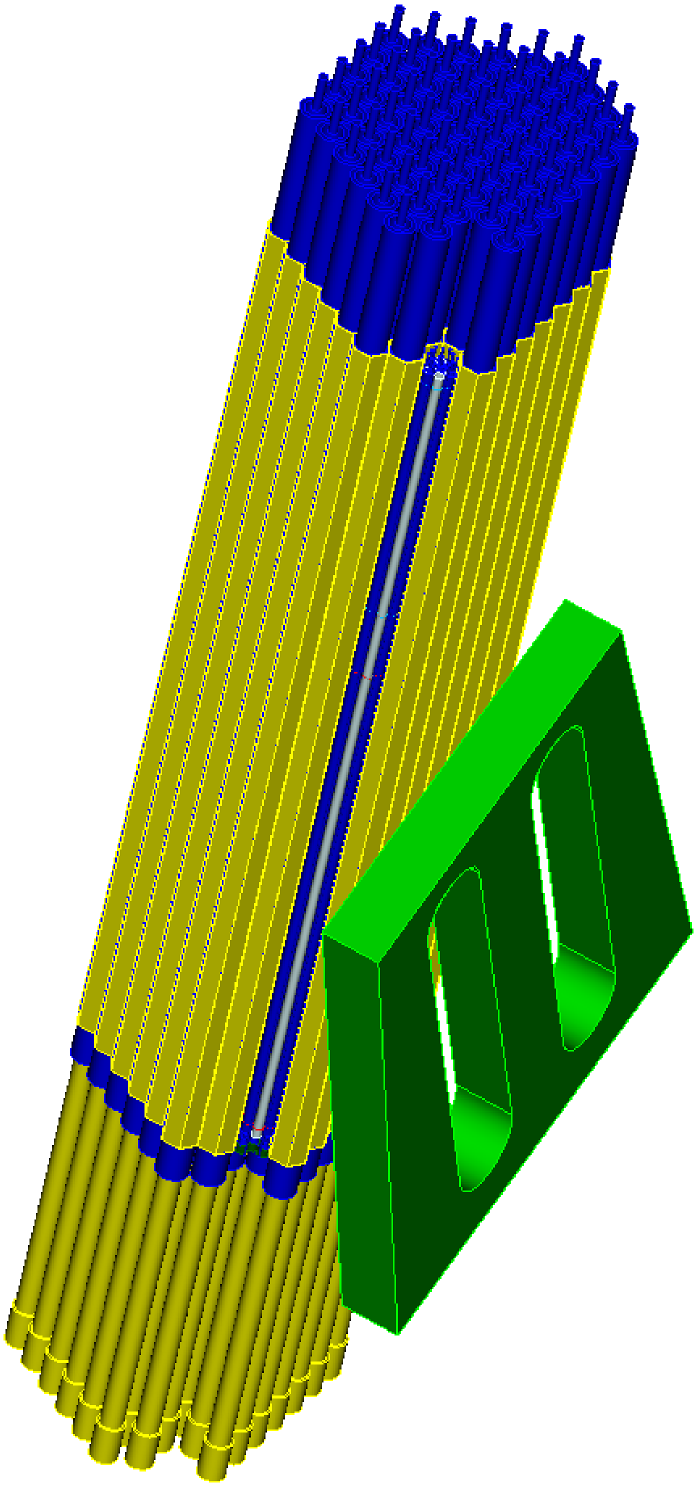


Figure 63. Overview of the fuel assemblies and the two-hole blade of the rotating reflectors; sodium in the frontal assembly and the steel clad in the frontal fuel rod are hidden

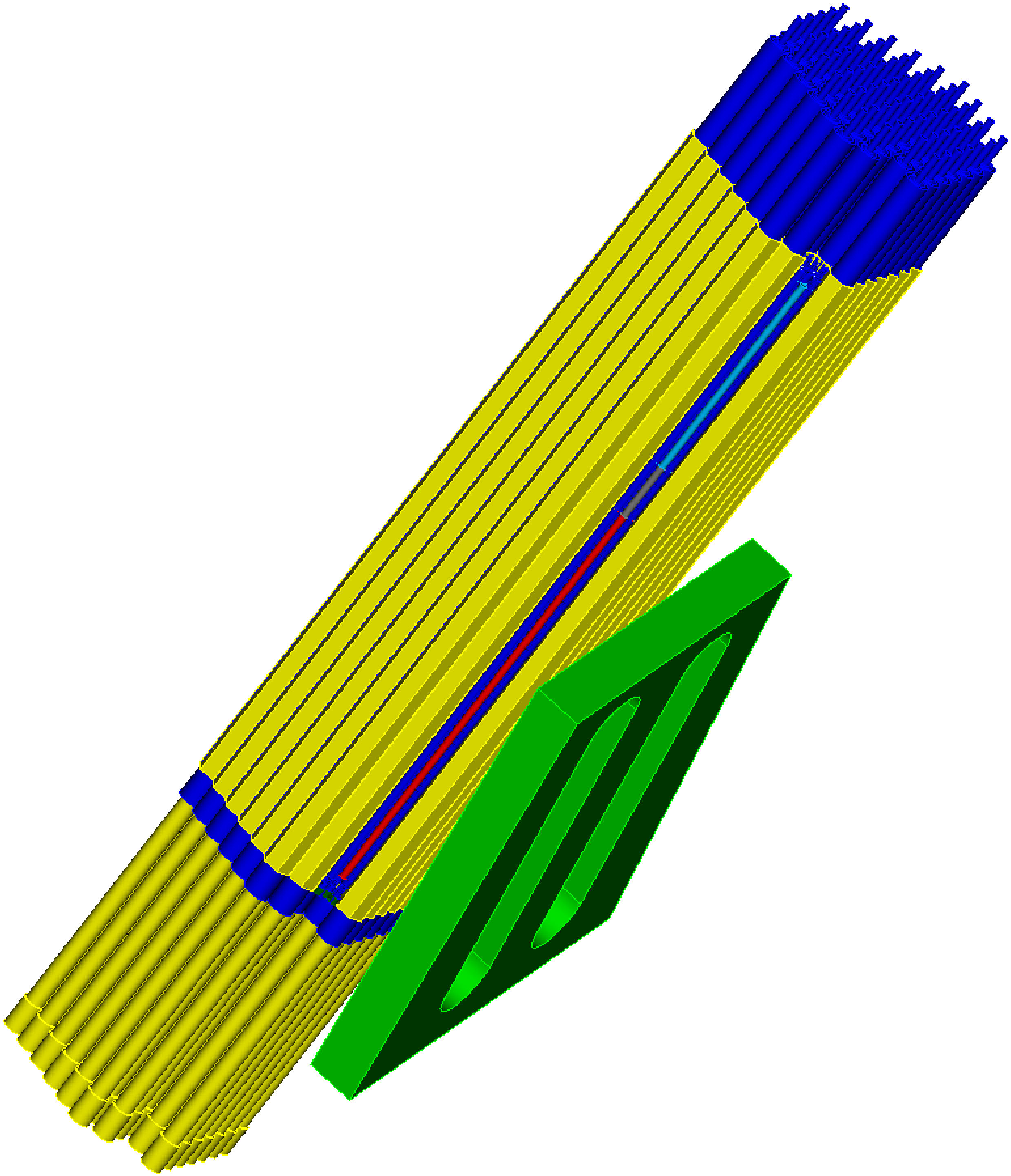


Figure 64. Overview of the fuel assemblies and the two-hole blade; the steel hollow cylinders at the bottom of the fuel assemblies (Figs. 36 and 37 top), the frontal assembly, and the steel clad and the helium gap in the frontal fuel rod are hidden

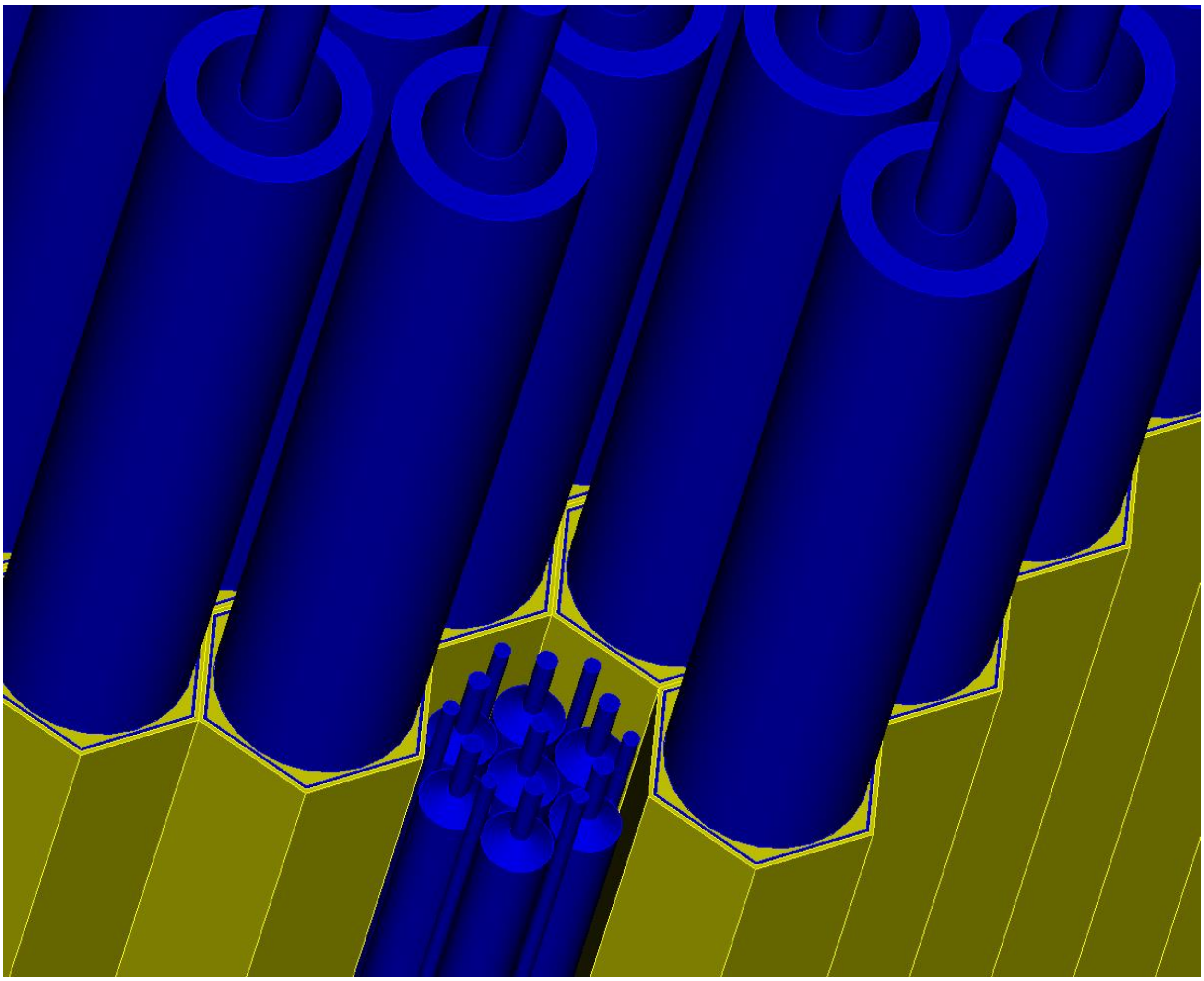


Figure 65. Zoom of the top part of the fuel rods and the assemblies; sodium and steel in the frontal assembly are hidden

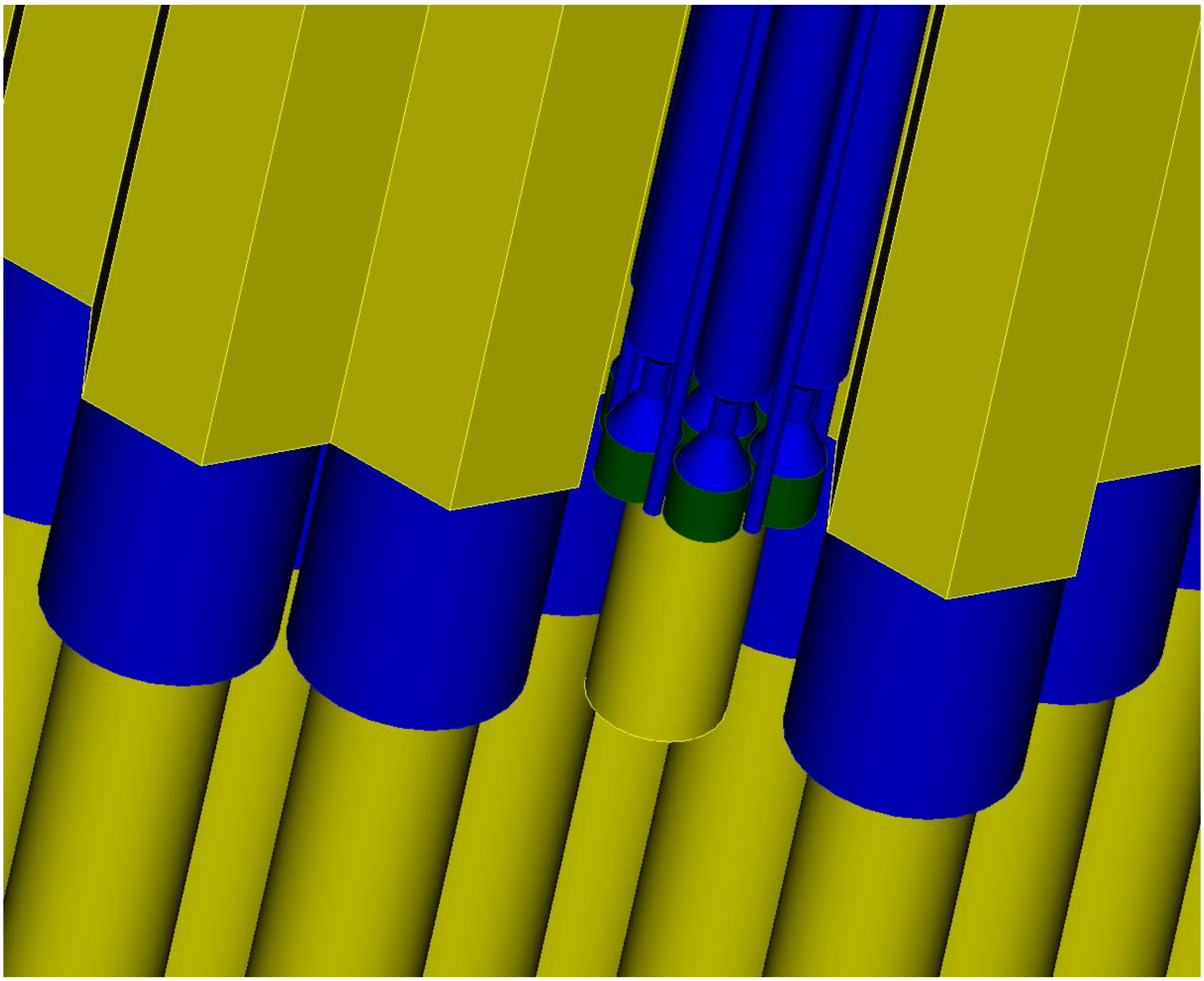


Figure 66. Zoom of the bottom part of the fuel rods and the assemblies; sodium and steel in the frontal assembly are hidden with exception of the sodium flowing inside the steel hollow cylinder

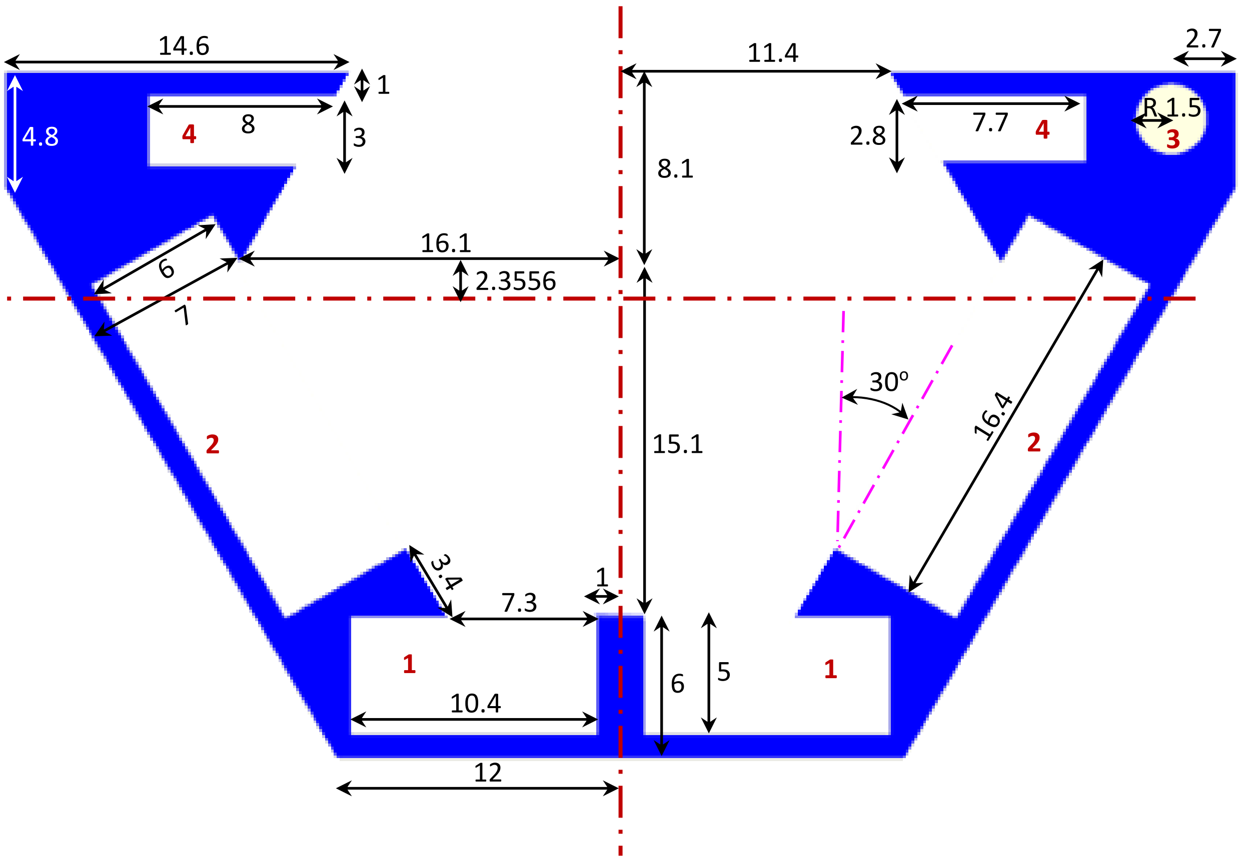


Figure 67. Dimensions [cm] on the x-y plane of the stationary reflector; the red axes mark the center of the core; red numbers legend: 1) SCRAM rods; 2) compensation rods; 3) automatic control rod (beryllium); 4) manual control rods; R is acronym for radius

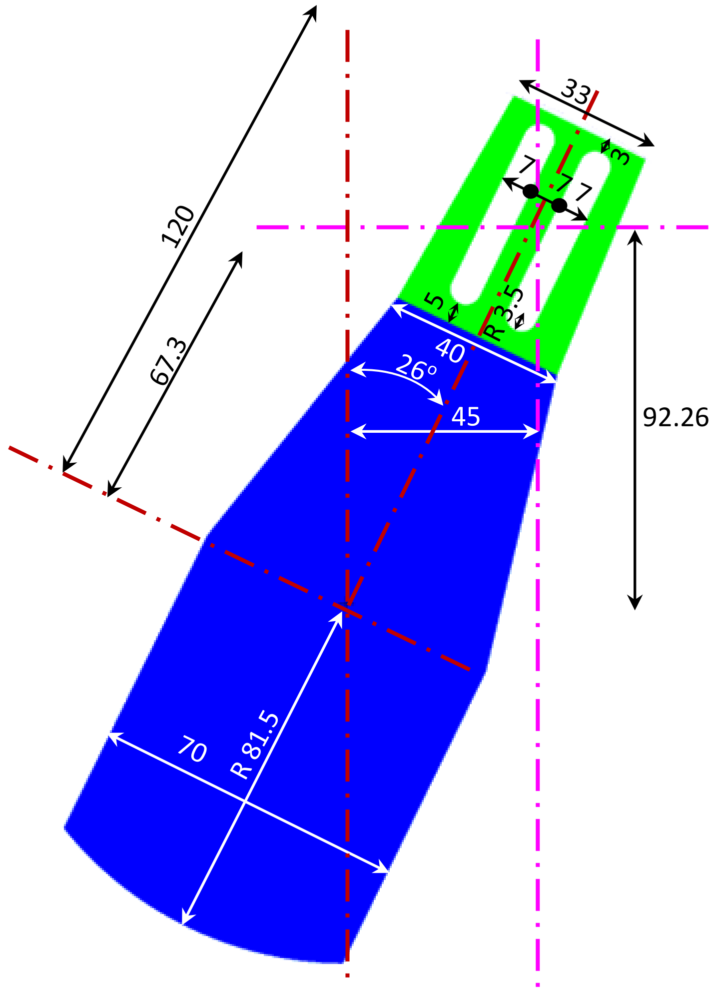


Figure 68. Dimensions [cm] on the x-z plane of the main rotating reflector part; the magenta and the red axes mark the centers of the core and the rotating shaft, respectively; R is acronym for radius; the nickel blade has green color; the steel blade-holder has blue color

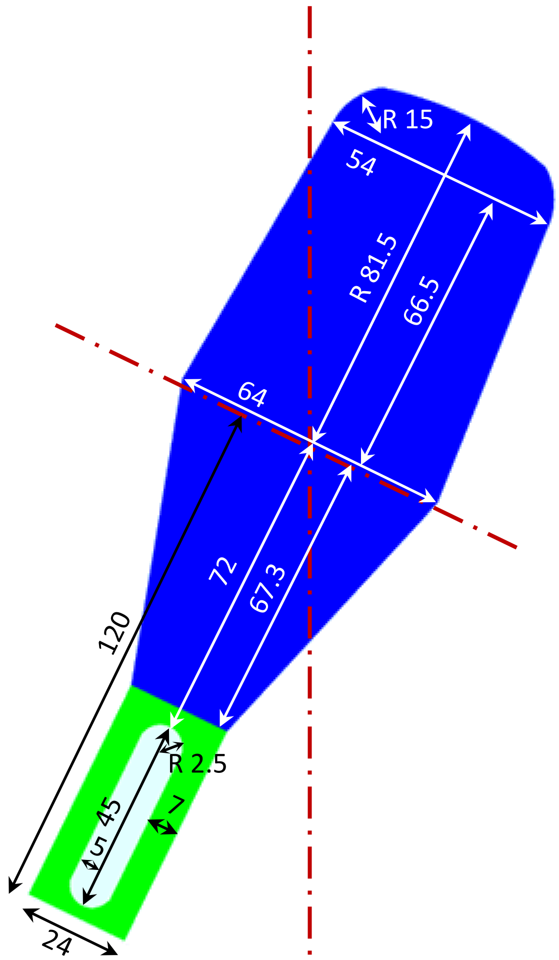


Figure 69. Dimensions [cm] on the x-z plane of the auxiliary rotating reflector part; the red axes mark the center of the rotating shaft; R is acronym for radius; the nickel blade has green color; the steel blade-holder has blue color

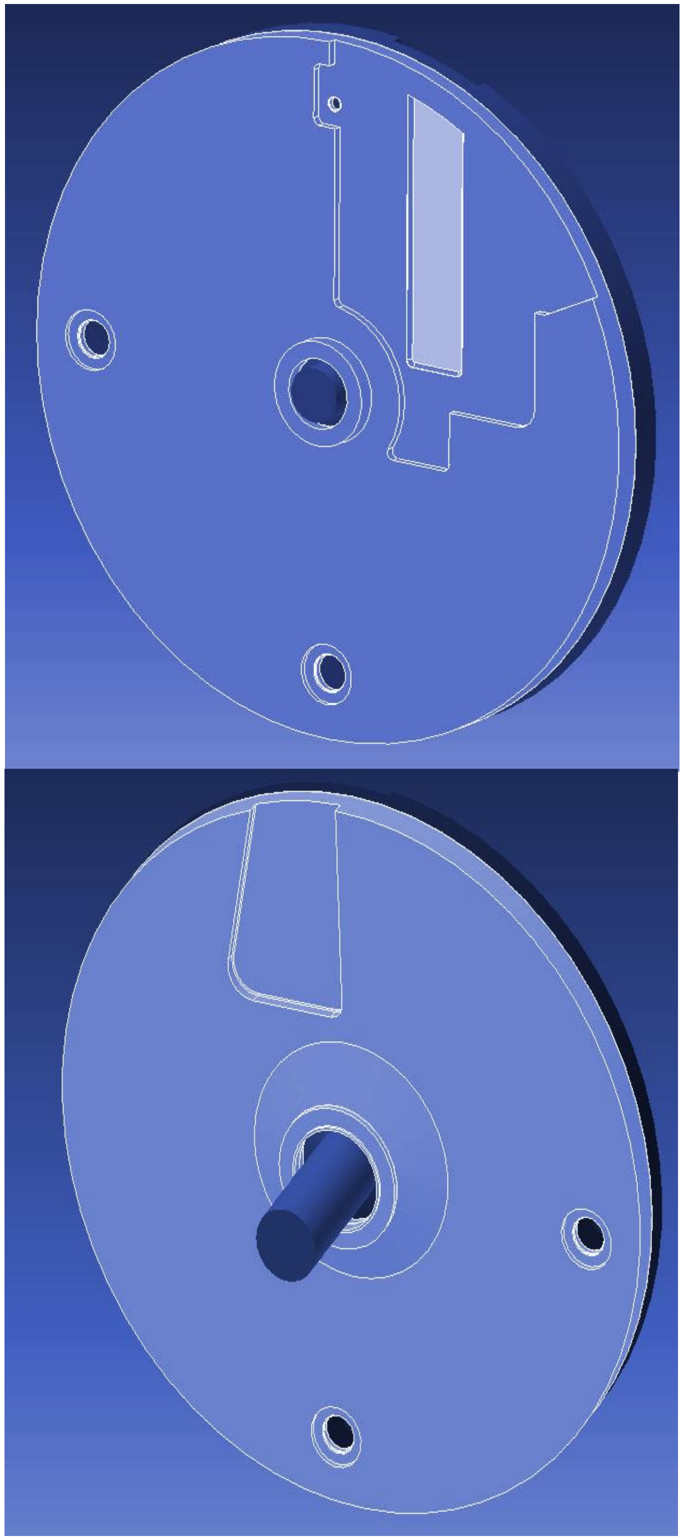


Figure 70. Illustration of the jackets and the cylindrical casing from the CAD model of the design; the top and the bottom plots illustrate the inner and the outer jackets, respectively

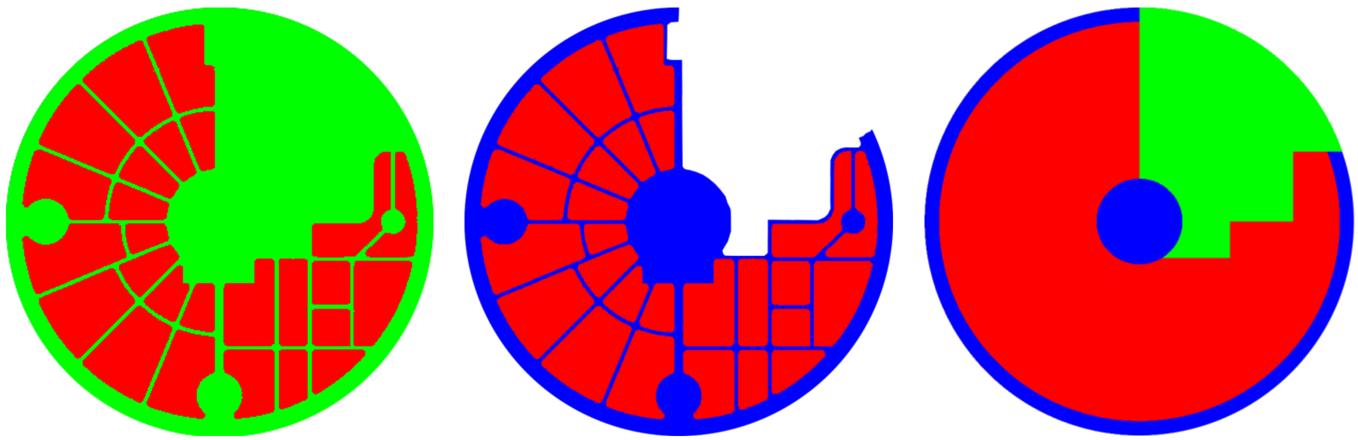


Figure 71. Pictures used to calculate the thickness (along the y-axis parallel to the rotating shaft axis) of the steel and the water layers of the inner jacket by GIMP and MATLAB. The central plot shows the design of the jacket; the blue and the red colors define steel and water, respectively. The right plot is from the MCNP model and it has air (green), steel (blue), red (water). The left plot has been used by GIMP to calculate the ratio between the water and the circle areas; the circle area has been calculated from the technical specifications

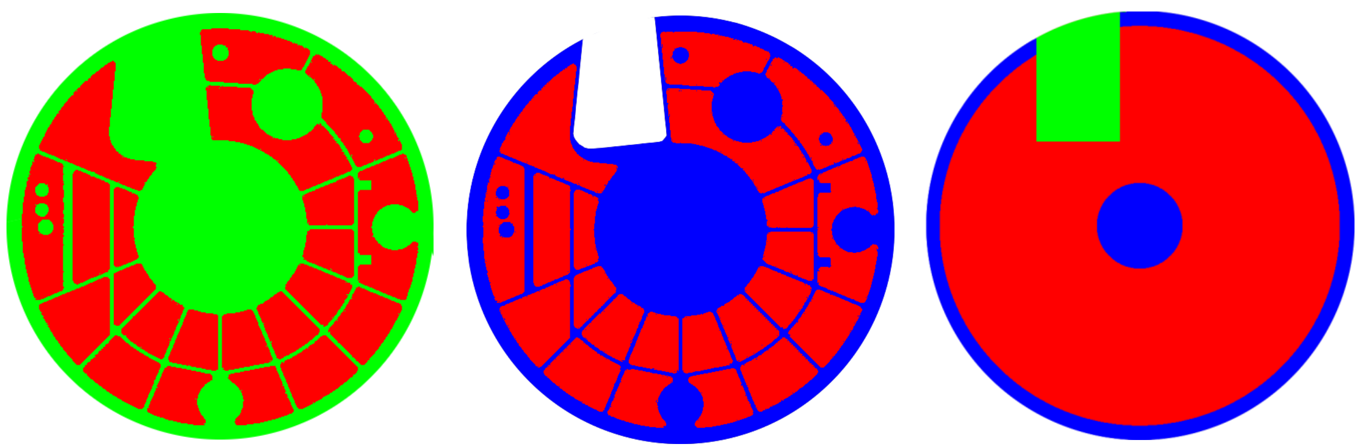


Figure 72. Pictures used to calculate the thickness (along the y-axis parallel to the rotating shaft axis) of the steel and the water layers of the outer jacket by GIMP and MATLAB. The central plot shows the design of the jacket; the blue and the red colors define steel and water, respectively. The right plot is from the MCNP model and it has air (green), steel (blue), red (water). The left plot has been used by GIMP to calculate the ratio between the water and circle areas; the circle area has been calculated from the technical specifications

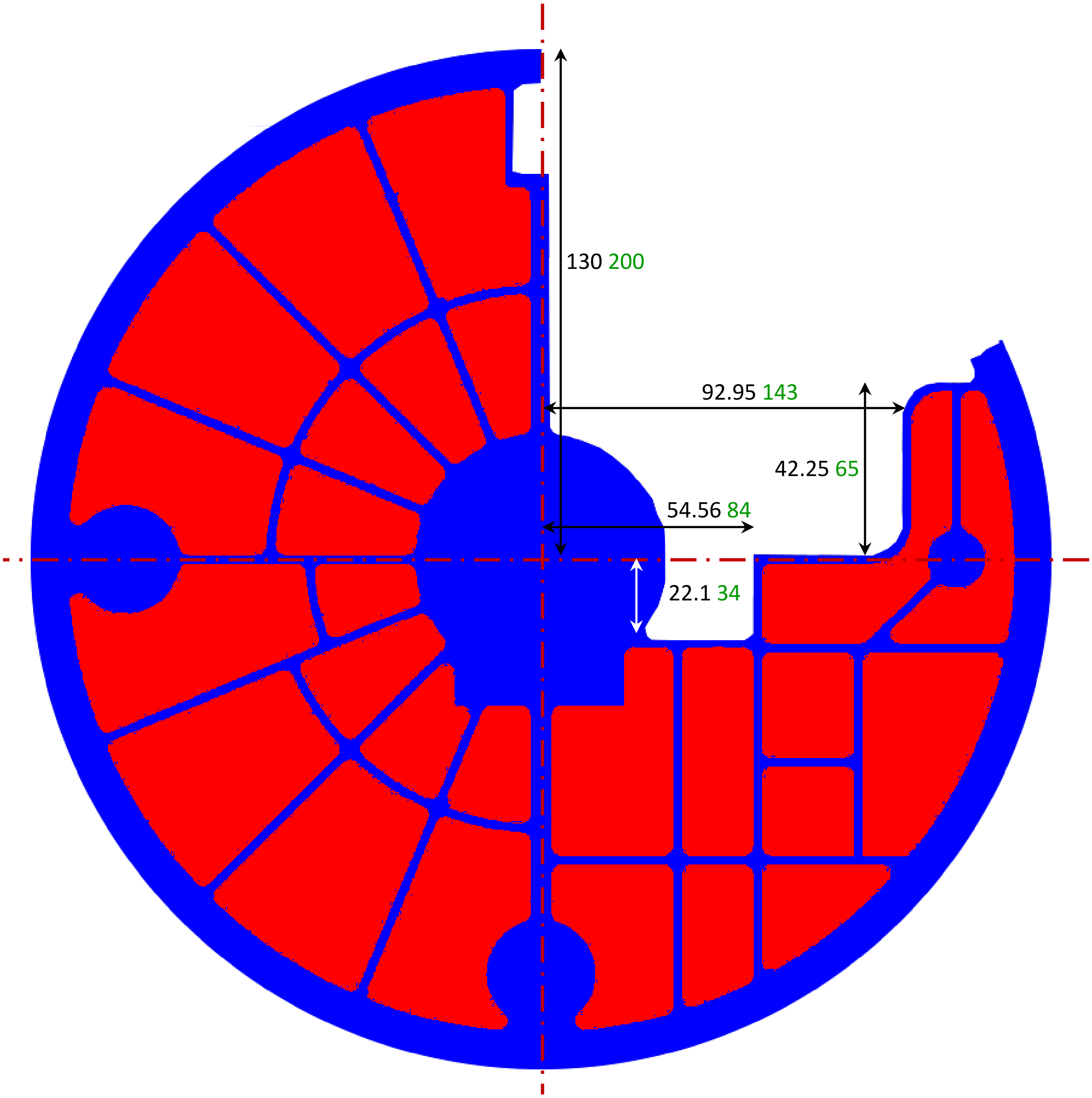


Figure 73. Dimensions [cm] on the x-z plane of the steel and the water central layer of the inner jacket; the plot shows the jacket design; numbers in green color are in pixel unit (from GIMP measurements); the red axes mark the center of the rotating shaft

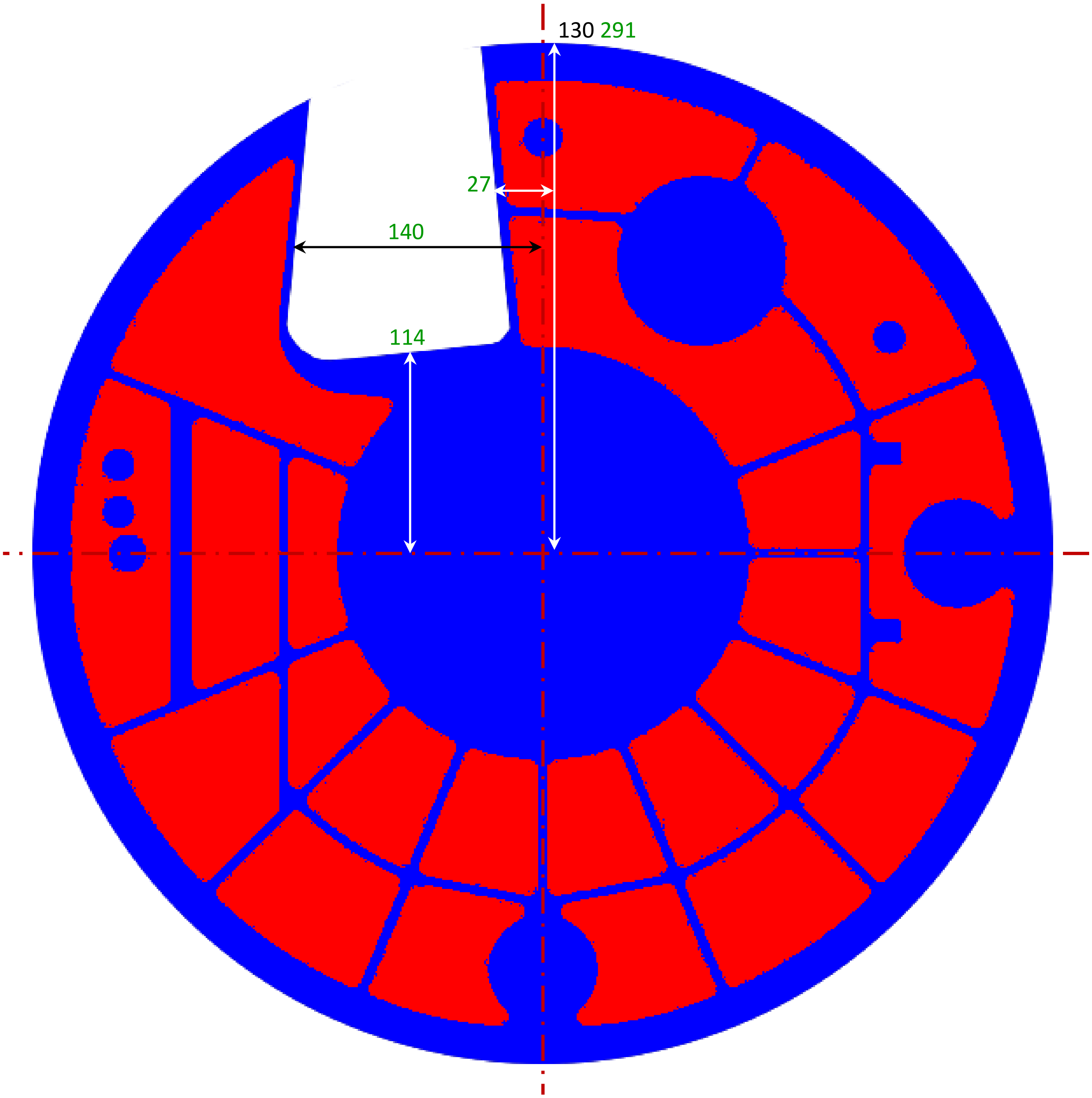


Figure 74. Dimensions [cm] on the x-z plane of the steel and the water central layer of the outer jacket; the plot shows the jacket design; numbers in green color are in pixels (from GIMP measurements); the red axes mark the center of the rotating shaft

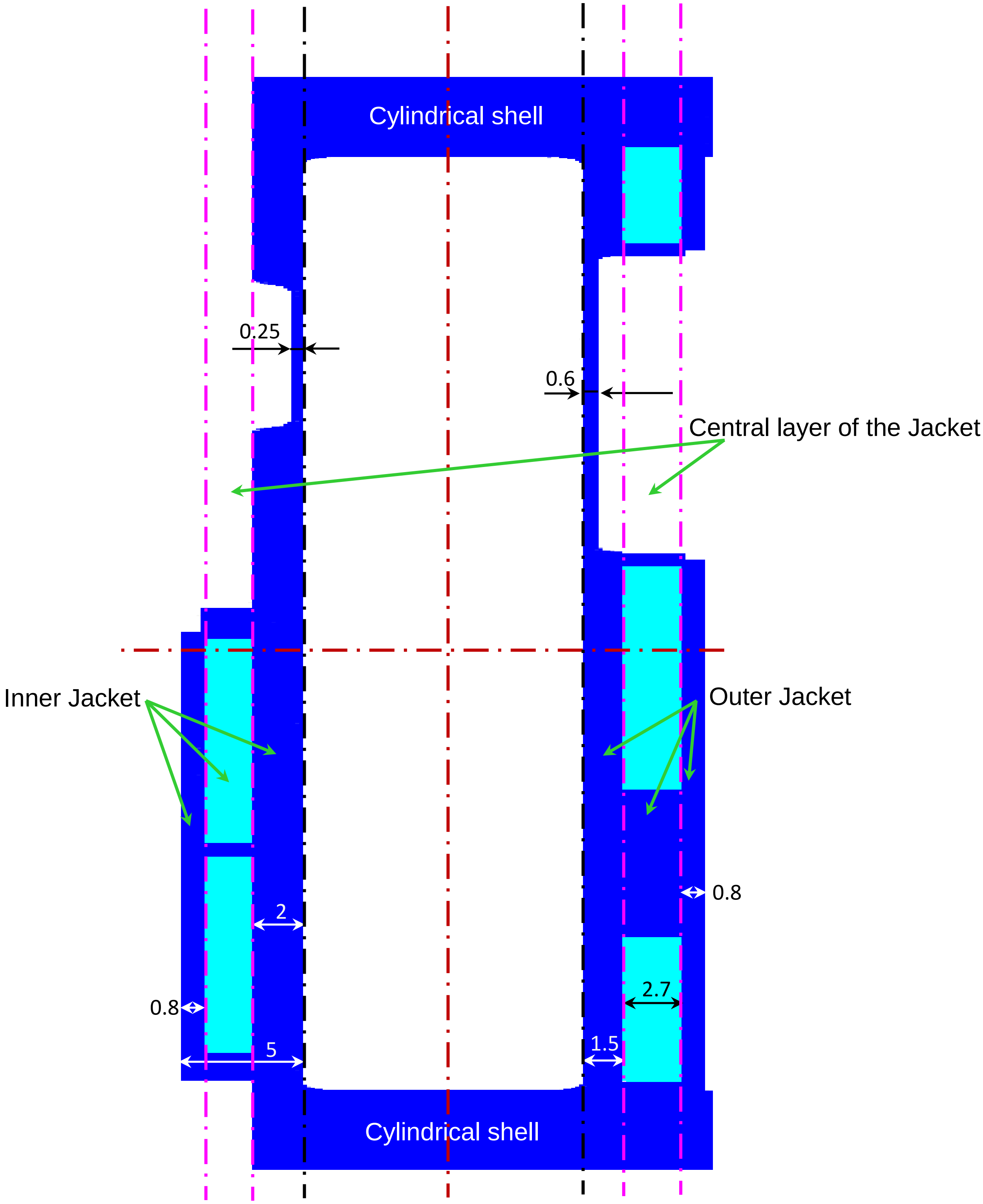


Figure 75. Dimensions [cm] on the x-y plane of the inner and the outer jackets; the plot represent the jacket design. The red axes mark the center of the rotating shaft and the black lines mark the boundaries of the jackets and the cylindrical shell. The magenta lines mark the boundaries of the central layer of the inner and the outer jackets; blue and cyan colors indicate steel and water, respectively. The figure is not drawn to scale

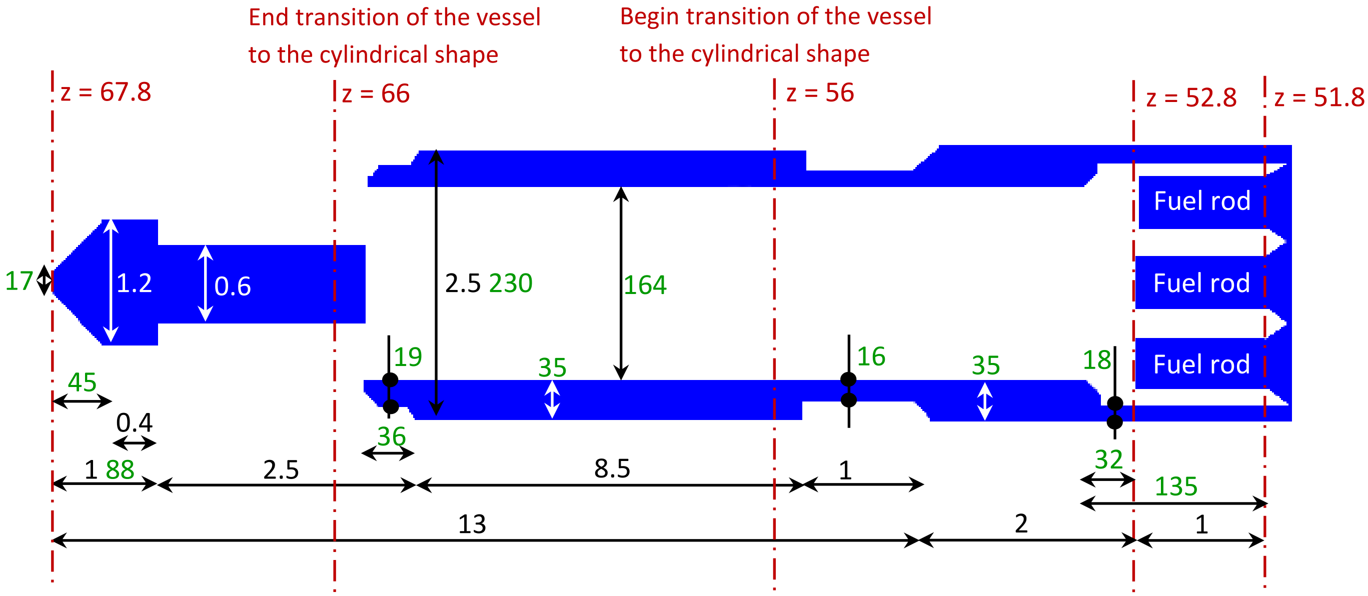


Figure 76. Illustration of the fuel assembly at the top used in the modeling process. Numbers in green color are in pixel unit from GIMP measurements. Numbers in black and red colors are in cm unit. Above z=52.8 cm the fuel assembly has cylindrical instead of hexagonal, shape. The figure is not drawn to scale.

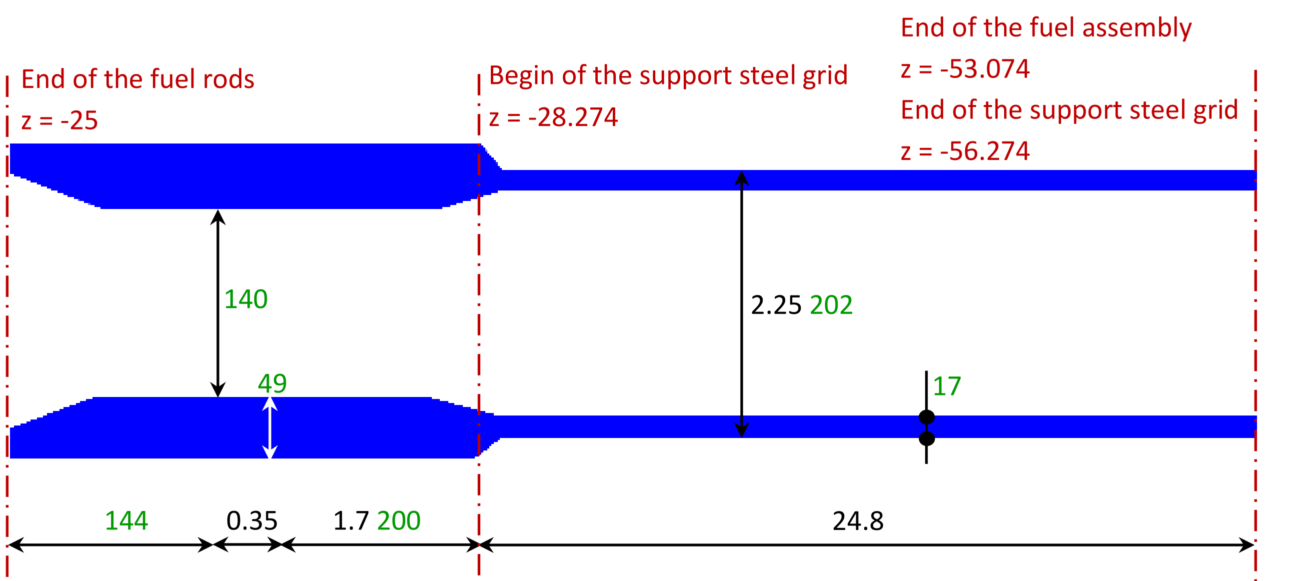


Figure 77. Illustration of the fuel assembly at the bottom used in the modeling process. Numbers in green color are in pixel unit from GIMP measurements. Numbers in black and red colors are in cm. Below z=-25 cm the fuel assembly has cylindrical, instead of hexagonal, shape. The figure is not drawn to scale

Figure 78. Pictures used to homogenize the steel and the sodium zones at the bottom end of the fuel rod by GIMP and MATLAB. The red axis marks the axis of the fuel rod. In the B plot, the red and green colors define the sodium and steel zones, respectively. The pictures refer to planes parallel to the z-axis; plot B shows design plot. The other images are used as discussed in Section 3

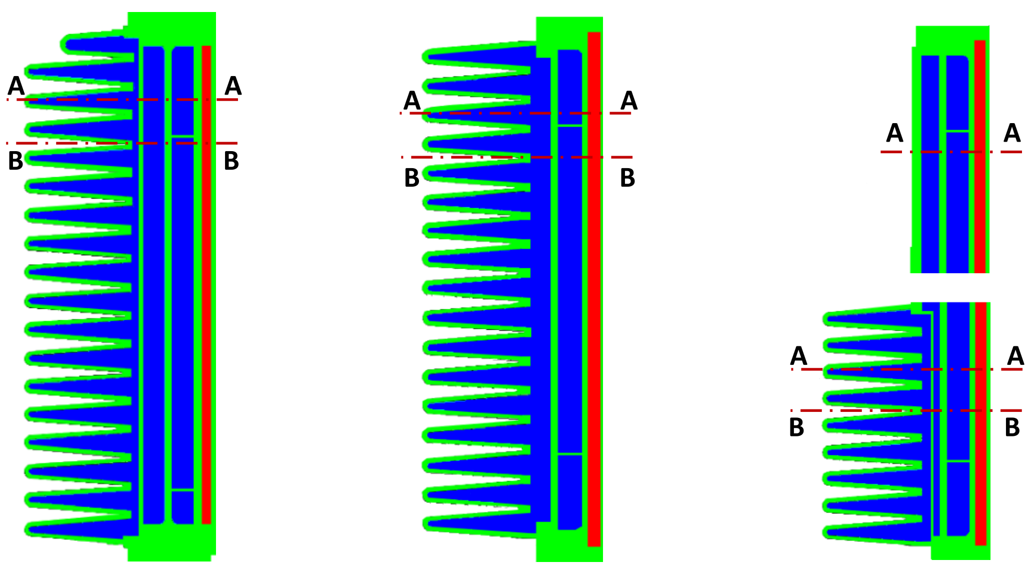


Figure 79. Plots used to homogenize aluminum (green) and water (blue) of the cooling system of the stationary reflector by GIMP and MATLAB. Boron carbide (red) has not been homogenized. The left picture refers to the aluminum and the water of the central block shown in Figs. 46 and 54. The central picture refers to the aluminum and water of the right block shown in Figs. 46 and 54. The top-rigth picture refers to the top-left aluminum and water of the block shown in Figs. 39 and 58 left. The bottom-rigth picture refers to aluminum and water of the bottom-left block shown in Figs. 46 and 58 left. All plots refer to planes parallel to the z-axis

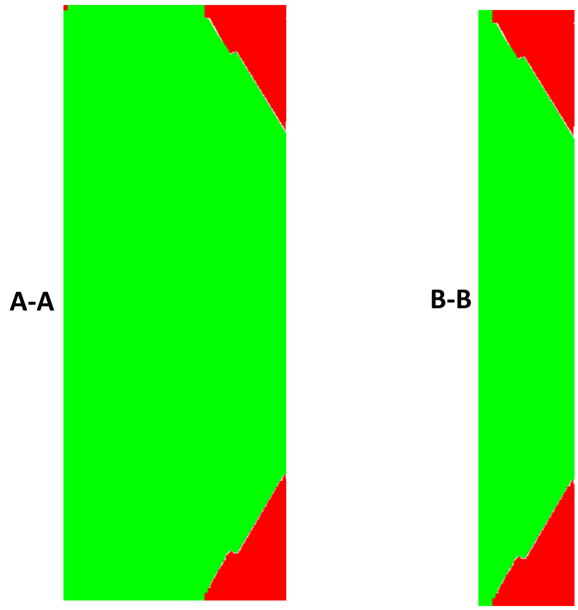


Figure 80. Plots used to homogenize aluminum and water in the cooling system of the stationary reflector by GIMP and MATLAB. The red color (air) defines the cut area in the aluminum and the water block in the left plot of Fig. 79. The left and right plots are sections A-A and B-B, respectively, of the left plot of Fig. 79. The plots refer to planes perpendicular to the z-axis

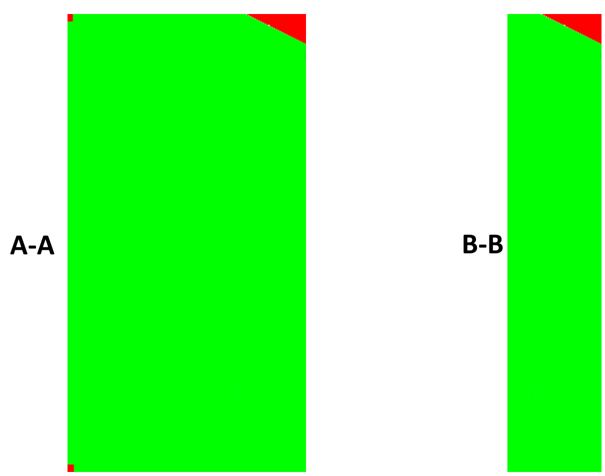


Figure 81. Plots used to homogenize aluminum and water in the cooling system of the stationary reflector by GIMP and MATLAB. The red color (air) defines the cut area in the aluminum and water block in the central and bottom-left plots of Fig. 79. The left and right plots are sections A-A and B-B, respectively, of the central and bottom-left plots of Fig. 79. The plots refer to planes perpendicular to the z-axis

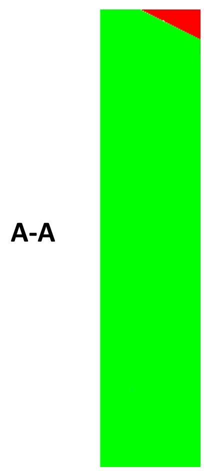


Figure 82. Plot used to homogenize aluminum and water in the cooling system of the stationary reflector by GIMP and MATLAB. The red color (air) defines the cut area in the aluminum and water block in the top-right plot of Fig. 79. The plot is section A-A of the top-right plot of Fig. 79. The plots refers to a plane perpendicular to the z-axis

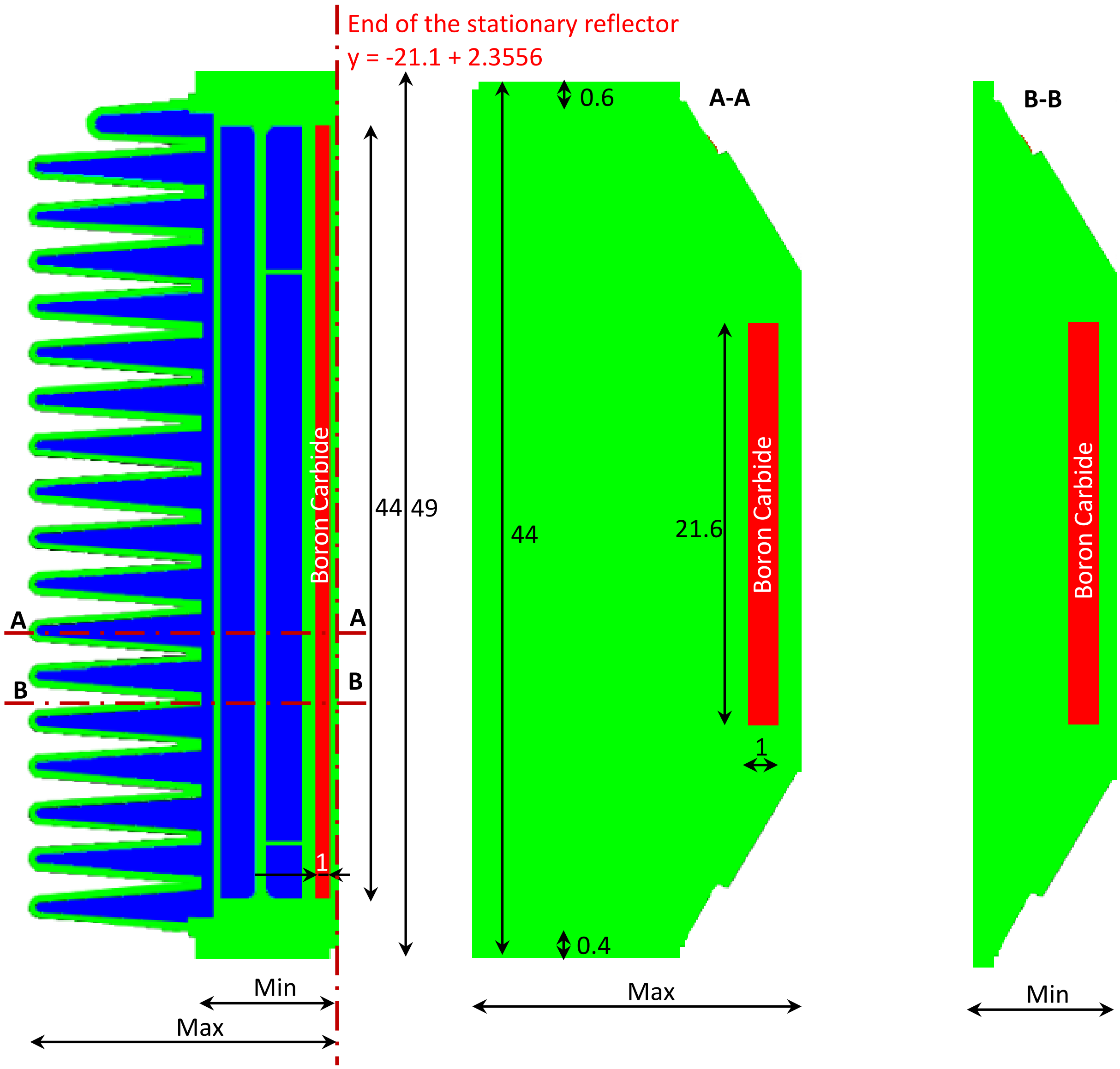


Figure 83. Dimensions [cm] of the left plot of Fig. 79 (left plot), left plot of Fig. 80 (central plot), and right plot of Fig. 80 (right plot); the 0.4 and 0.6 cm lengths in the central plot mark the pure aluminum layers used in Script 8 of Appendix C. Figures are not drawn to scale

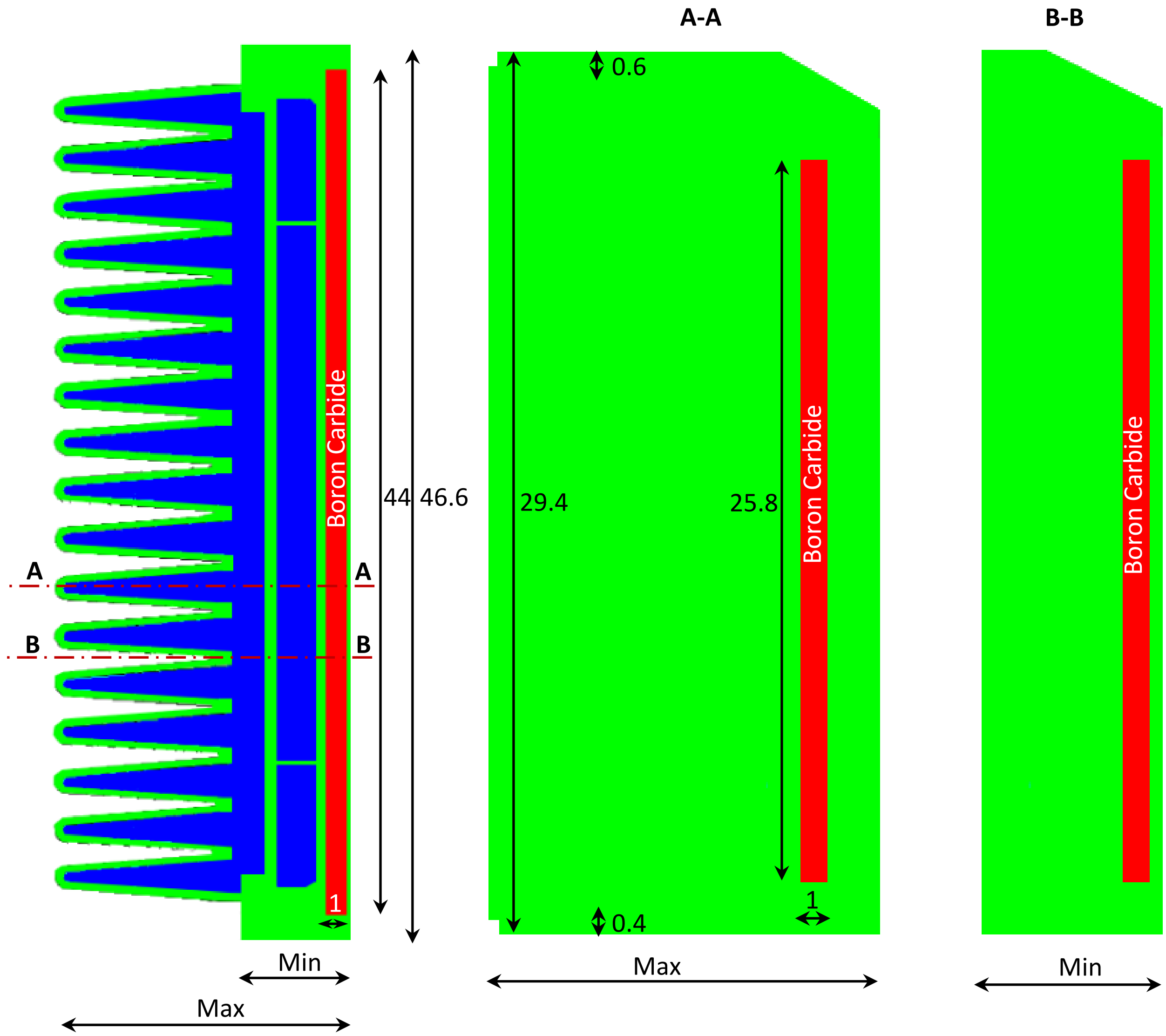


Figure 84. Dimensions [cm] of the central plot of Fig. 79 (left plot), left plot of Fig. 81 (central plot), and right plot of Fig. 81 (right plot); the 0.4 and 0.6 cm lengths in the central plot mark the pure aluminum layers used in Script 8 of Appendix C. Figures are not drawn to scale

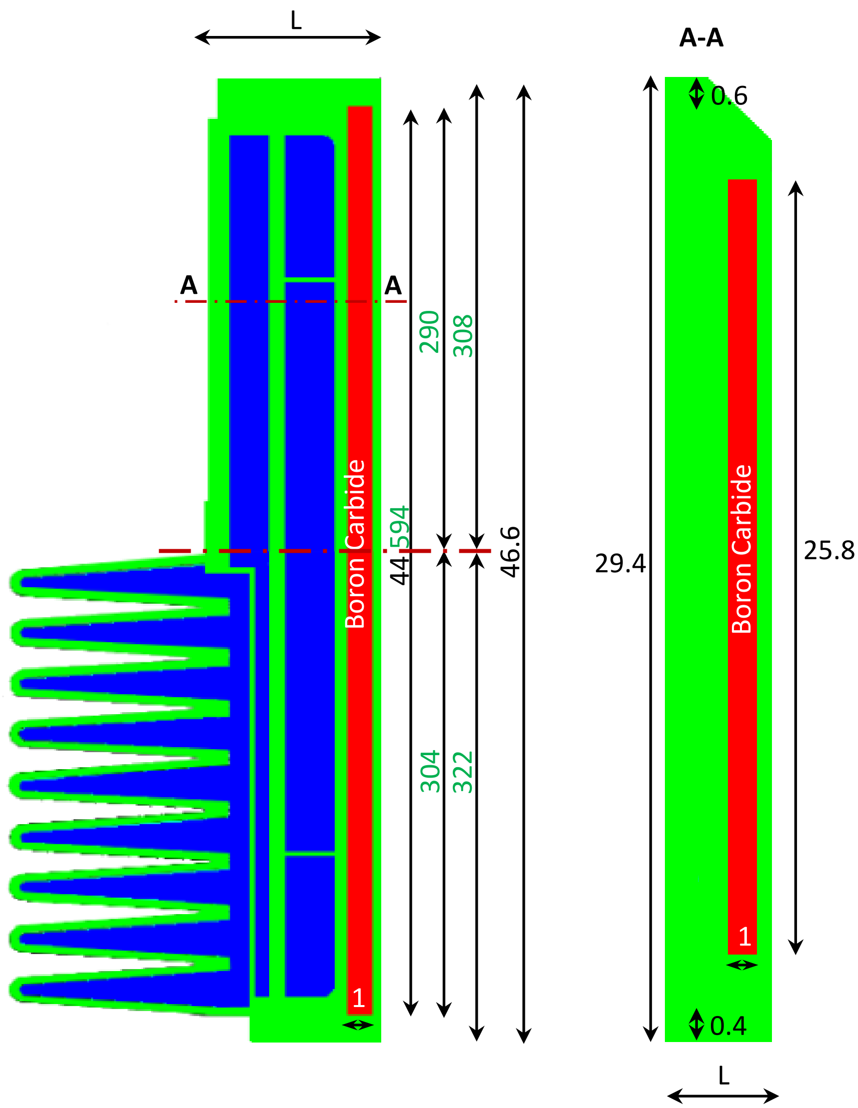


Figure 85. Dimensions [cm] of the top-right and bottom-right plots of Fig. 79 (left plot) and Fig. 82 (right plot); numbers in green color are in pixel unit from GIMP measurements; numbers in black color are in cm unit; the 0.4 and 0.6 cm lengths in the central plot mark the pure aluminum layers used in Script 8 of Appendix C. The red axis on the left plot marks the boundary of the top-right and bottom-right plots of Fig. 79. Figures are not drawn to scale

Appendix A: VISUALBASIC Macro to Crop and Resize Images in Microsoft Word

This macro removes MCNP text in the MCNP images by cropping the left and top sides. For changing the hot key in an existing macro, select: 1) options; 2) customize ribbon; 3) customize keyboard.

By default, key F4 in Microsoft Word repeats the last command and ALT+F9 toggles field codes.

```
Sub Macro1()
Dim myHeight, myWidth, myCropL, myCropT As Single
myCropL = 0.29
myCropT = 0.07
With Selection.InlineShapes(1)
myHeight = .Height
myWidth = .Width
With .PictureFormat
.CropLeft = myWidth * myCropL
.CropTop = myHeight * myCropT
End With
' to use only for resizing
' .Height = .Height * resizefactor
' .Width = .Width * resizefactor
End With
End Sub
```

Appendix B: Basics Commands of the RCS Revision Control System of Linux

Build the directory database
mkdir RCS; chmod 700 RCS

Add the file filename.txt to the database
ci filename.txt

Set the locking to non-strict in file filename.txt
rcs -U filename.txt

Lock the file filename.txt (to use only if the locking is non-strict)
rcs -l filename.txt

Unlock the file filename.txt (to use only if the locking is non-strict)
rcs -u filename.txt

Retrieve the file filename.txt from the database
co filename.txt

Retrieve revision X.Y of the file filename.txt from the database
co –rX.Y filename.txt

List all modifications to the file filename.txt
rlog filename.txt

List all modifications between revisions X.Y and Z.W
rcsdiff –rX.Y –rZ.W filename.txt

Add revision X.Y to the database
ci –rX.Y filename.txt

Change comments for revision X.Y
rcs –mX.Y:"new comments" filename.txt

Change comments on multiple lines for revision X.Y
rcs –mX.Y:"new comments first line (hit enter)
new comments second line (hit enter)
new comments third line (hit enter)
" filename.txt (hit enter)

Appendix C: MATLAB Scripts for MCNP Modeling

Script 1 includes the wrapping wire in the clad radius

```
                                     % 0.431 cm is the nominal clad radius
A=pi*0.13*0.04;                      % area of the elliptic wire from p. 50 of Ref. 1
p=10;                                % 10 cm is the wire pitch on the z-axis
R=0.431+0.5*(0.13+0.04);             % radius of the center of the wire
z=(8+444+60+223+(23-12))/10;         % z length of the fuel rod with the wrapping wire
L=sqrt(p^2+(2*pi*R)^2);              % length of the wire taking into account only one rotation in the clad
L=L*z/p;                             % total length of the wire
V=A*L;                               % total volume of the wire
rnew=sqrt(0.431^2+V/(z*pi))          % adjusted radius of the steel clad taking into account the wire volume
```

Script 2 calculates the planes of the vessel shape transition (from hexagonal to circular) at bottom

```
dxy1 =[11.275 11.275 11.277 13.514 11.277 13.514]; % Table I and Fig. 55
phi  =[-90/180*pi 90/180*pi -30/180*pi 30/180*pi -150/180*pi 150/180*pi];
dxy  =[dxy1;dxy1+0.7;dxy1+0.7+0.3;dxy1+0.7+0.3+0.4];
theta(1,:)=atan((dxy(1,:)- 4              )/9.5);
theta(2,:)=atan((dxy(2,:)-(4+0.7)         )/9.5);
theta(3,:)=atan((dxy(3,:)-(4+0.7+0.3)     )/9.5);
theta(4,:)=atan((dxy(4,:)-(4+0.7+0.3+0.4))/9.5);
for i=1:size(dxy,1)
  n(i,:)=dxy(i,:).*cos(theta(1,:));          % distance to the normal vector of the plane
end
m=72;
for i=1:size(dxy,1)
  for j=1:size(dxy,2)
    x= n(i,j)*cos(theta(i,j))*cos(phi(j));  % x coordinate of the normal vector
    y= n(i,j)*cos(theta(i,j))*sin(phi(j));  % y coordinate of the normal vector
    z=-n(i,j)*sin(theta(i,j));              % z coordinate of the normal vector
    if (abs(x)<1e-10)
      x=0;
    end
    fprintf('%d   p   %+e %+e %+e %+e\n',m,x,y,z,n(i,j)^2);
    m=m+1;
  end
end
```

Script 3 calculates the vertical planes of the stationary reflector

```
c=cos(30/180*pi); % Figs. 47 top and 54
s=sin(30/180*pi);
p0=[-7.3,-15.1];
p1=p0+3.4*[-s,+c];
p2=p1+6*[-c,-s];
p3=p2+16.4*[-s,+c];
p4=p1+16.4*[-s,+c];
fprintf('323  p  %e %e %e\n       %e %e %e\n       %e %e %e\n', ...
p1(1), p1(2),0,p2(1),p2(2),0,p2(1),p2(2),1);
fprintf('324  p  %e %e %e\n       %e %e %e\n       %e %e %e\n', ...
p3(1), p3(2),0,p2(1),p2(2),0,p2(1),p2(2),1);
fprintf('325  p  %e %e %e\n       %e %e %e\n       %e %e %e\n', ...
p3(1), p3(2),0,p4(1),p4(2),0,p4(1),p4(2),1);
p1=[-p1(1),p1(2)];
p2=[-p2(1),p2(2)];
p3=[-p3(1),p3(2)];
p4=[-p4(1),p4(2)];
fprintf('326  p  %e %e %e\n       %e %e %e\n       %e %e %e\n', ...
p1(1), p1(2),0,p2(1),p2(2),0,p2(1),p2(2),1);
fprintf('327  p  %e %e %e\n       %e %e %e\n       %e %e %e\n', ...
p3(1), p3(2),0,p2(1),p2(2),0,p2(1),p2(2),1);
fprintf('328  p  %e %e %e\n       %e %e %e\n       %e %e %e\n', ...
p3(1), p3(2),0,p4(1),p4(2),0,p4(1),p4(2),1);
```

Script 4 calculates the thickness of the jackets layers

```
% inner jacket (next to the core)
% 130 cm are 200 pixel; Fig. 73
p=130/200;     % Fig. 73
x_right1=84*p  % Fig. 73
x_right2=143*p % Fig. 73
y_up=65*p      % Fig. 73
y_down=34*p    % Fig. 73
                                      % thickness array; left side of Fig. 75
t=[0.8 5-2-0.8 2 0.25]                % [steelcover water steel thinsteel] the thickness of thinsteel is included in steel
rgb_real_all=[127.5 127.5 0];         % Fig. 71 left;   water red ; green everything else
rgb_real_cut=[163.9 0     91.1];      % Fig. 71 center; water red ; steel blue
rgb_mcnp    =[160.9 57.8  36.3];      % Fig. 71 right;  water red ; steel blue ; green air
v_water_real=pi*130^2*t(1,2)*rgb_real_all(1,1)/sum(rgb_real_all);  % real volume of water
t_water=v_water_real/(pi*130^2*rgb_mcnp(1,1)/sum(rgb_mcnp))   % thickness of water in mcnp ; 2nd element in vector
v_steel_real=v_water_real*rgb_real_cut(1,3)/rgb_real_cut(1,1);    % real volume of steel in the central layer
v_steel_real=v_steel_real-pi*130^2*t_water*rgb_mcnp(1,3)/sum(rgb_mcnp);       % subtract the steel volume which was modeled in the water layer in MCNP
t_steel=v_steel_real/(pi*130^2*(rgb_mcnp(1,1)+rgb_mcnp(1,3))/sum(rgb_mcnp)) % this extra thickness adds to the 0.8 cm nominal value steelcover in MCNP, Fig. 75
%
% outer jacket (far from to the core)
% 130 cm are 291 pixel; Fig. 74
p=130/291;    % Fig. 74
y_up=114*p    % Fig. 74
x_left1=27*p  % Fig. 74
x_left2=140*p % Fig. 74
% thickness array; right side of Fig. 75
t=[0.6 1.5 2.7 0.8] % [thinsteel steel water steelcover] the thickness of thinsteel is included in steel
rgb_real_all=[131   124    0];      % Fig. 72 left;   water red ; green everything else
rgb_real_cut=[139.5 0    115.5];  % Fig. 72 center; water red ; steel blue
rgb_mcnp    =[196.9 17.7  40.4]; % Fig. 72 right;  water red ; steel blue ; green air
v_water_real=pi*130^2*t(1,3)*rgb_real_all(1,1)/sum(rgb_real_all);                      % real volume of water
t_water=v_water_real/(pi*130^2*rgb_mcnp(1,1)/sum(rgb_mcnp))                            % thickness of water in mcnp ; 3rd element in vector
%
v_steel_real=v_water_real*rgb_real_cut(1,3)/rgb_real_cut(1,1);                         % real volume of steel in the central layer
v_steel_real=v_steel_real-pi*130^2*t_water*rgb_mcnp(1,3)/sum(rgb_mcnp);        % subtract the steel volume which was modeled in the water layer in MCNP
t_steel=v_steel_real/(pi*130^2*(rgb_mcnp(1,1)+rgb_mcnp(1,3))/sum(rgb_mcnp)) % this extra thickness adds to the 0.8 cm nominal value steelcover in MCNP, Fig. 75
```

Script 5 calculates the radii at the top of the fuel assemblies

```
% 2.5 cm are 230 pixel; Fig. 76
p=2.5/230;
h=2+1+8.5+36*p % Fig. 76
r_outer=((135*p-1)*(82+18)*p+... % Fig. 76
        (3-135*p)*(82+35)*p+...
         1*      (82+16)*p+...
         8.5*     (82+35)*p+...
         36*p*    (82+19)*p)/h  % mean  outer radius
r_inner=82*p                  % fixed inner radius; Fig. 76
PI=3.14159265358979323846;
v=PI*0.3^2*(2.5-36*p)+...
  PI*0.6^2*0.4+...
  PI*(1/3)*(0.6^2+0.6*17*p/2+17^2*p^2/4)*0.6; % volume truncated cone = 1/3*h*(r1^2+r1*r2+r2^2)
h=h+2.5-36*p+1
r=sqrt(v/(PI*(1+2.5-36*p)))
```

Script 6 calculates the radii at the bottom of the fuel assemblies

```
p=1.7/200; % 1.7 cm are 200 pixel; Fig. 77
h=144*p+0.35+1.7  % Fig. 77
p=2.25/101/2 % 2.25/2 cm are 101 pixel; Fig. 77
r_outer=(70+49)*p % Fig. 77
r_inner= 70    *p % Fig. 77
h=24.8
r_outer=2.25/2
r_inner=r_outer-17*p
```

Script 7 homogenizes the steel and sodium zone at the bottom of the fuel rod

```
% the volume of a solid of revolution is given by the area multiplied by the centroid
% see Fig. 78
rgb=[76.3 179.1 0]; % red sodium; green steel from Fig. 78B; Fig. 78B has red and green only on the right side (transparent on the left side); rgb estimates the areas
origin=[60.5 50]; % center position (in pixel unit) of Fig. 78B relative to (0,0) which is the bottom-left corner of Fig. 78A; Fig. 78A has green everywhere (left and right sides) no red;
red_centroid  =[82.77 56.86]-origin; % centroid position (in pixel unit) of red color zone in the right side of Fig. 78C; Fig. 78C has red on the right side and no green (transparent on the left side)
green_centroid=[90.5 50]-origin; % centroid position (in pixel unit) of green color in the right side of Fig. 78D; Fig. 78D has green on the right side; both steel and sodium are colored in green in the right side (transparent on the left side)
volume_fraction_red=(rgb(1,1)*red_centroid(1,1))/(sum(rgb)*green_centroid(1,1));
f=volume_fraction_red/(1-volume_fraction_red) % f = sodium volume / steel volume
```

Script 8 calculates the volume of the four blocks of the cooling system of the stationary reflector

```
% this script calculates the surfaces for the 4 aluminum&water and boron carbide zones around the stationary reflector
% rgb variable data are the red green blue values from GIMP color histogram; red=boroncarbide green=aluminum blue=water
rgb=[15.1 104.9 138.3   % Fig. 79 left         ; piece 1
    23.4  92.4 141.5   % Fig. 79 center       ; piece 2
    34.7 105.1 115.1   % Fig. 79 top-right    ; piece 3
    21.8 100.5 135.6]; % Fig. 79 bottom-right ; piece 4
% rgb_cut variable data are the uncut volume fractions for pieces 1 2 3 4; red green color values from GIMP color histogram; red=cut area (air), green=uncut area (aluminum+water+boroncarbide)
rgb_cut=[
% average of Figs. 80 left and 80 right; Fig. 80 right has been generated from Figs. 80 left and 79 left (see Fig. 83); 20.7 and 48.1 are the red values in Figs. 80 left and 80 right respectively; 234.3 and 206.9 are the green values in Figs. 80 left and 80 right respectively
       1-0.5*(20.7/(20.7+234.3)+48.1/(48.1+206.9))   % piece 1
% average of Figs. 81 left and 81 right; Fig. 81 right has been generated from Figs. 81 left and 79 central (see Fig. 84); 2.3 and 5.4 are the red values in Figs. 81 left and 81 right respectively; 252.7 and 249.7 are the green values in Figs. 81 left and 81 right respectively
       1-0.5*( 2.3/( 2.3+252.7)+ 5.4/( 5.4+249.7))   % piece 2
% value from Fig. 82; 5 and 250 are the red and green values, respectively, in Fig. 82
       1-     5  /( 5  +250)                        % piece 3
% average of Figs. 81 left and 81 right; Fig. 81 right has been generated from Figs. 81 left and 79 central (see Fig. 83); 2.3 and 5.4 are the red values in Figs. 81 left and 81 right respectively; 252.7 and 249.7 are the green values in Figs. 81 left and 81 right respectively
       1-0.5*( 2.3/( 2.3+252.7)+ 5.4/( 5.4+249.7))]; % piece 4
% boron carbide volume calculated with exact dimensions
v_B4C=[            44 * 1 * 21.6
                  44 * 1 * 25.8
      290/(290+304)*44 * 1 * 25.8   % 290 pixels height from GIMP in Fig. 85
      304/(290+304)*44 * 1 * 25.8]; % 304 pixels height from GIMP in Fig. 85
% aluminum volumes calculated using red/green/blue areas ratios from GIMP histograms data
% it is assumed that at the boundaries there are 0.4 and 0.6 pure aluminum layers (Figs. 83 center, 84 center, and 85 left); these layers are accounted in the last term
v_Al =[v_B4C(1,1)/21.6*rgb(1,2)./rgb(1,1)*(44  -0.6-0.4)*rgb_cut(1,1)+v_B4C(1,1)/21.6*sum(rgb(1,:))./rgb(1,1)*(0.6+0.4)*rgb_cut(1,1)
      v_B4C(2,1)/25.8*rgb(2,2)./rgb(2,1)*(29.4-0.6-0.4)*rgb_cut(2,1)+v_B4C(2,1)/25.8*sum(rgb(2,:))./rgb(2,1)*(0.6+0.4)*rgb_cut(2,1)
      v_B4C(3,1)/25.8*rgb(3,2)./rgb(3,1)*(29.4-0.6-0.4)*rgb_cut(3,1)+v_B4C(3,1)/25.8*sum(rgb(3,:))./rgb(3,1)*(0.6+0.4)*rgb_cut(3,1)
      v_B4C(4,1)/25.8*rgb(4,2)./rgb(4,1)*(29.4-0.6-0.4)*rgb_cut(4,1)+v_B4C(4,1)/25.8*sum(rgb(4,:))./rgb(4,1)*(0.6+0.4)*rgb_cut(4,1)];

% water volumes calculated using red/green/blue areas ratios from GIMP histograms data
v_H2O=[v_B4C(1,1)/21.6*rgb(1,3)./rgb(1,1)*(44  -0.6-0.4)*rgb_cut(1,1)
      v_B4C(2,1)/25.8*rgb(2,3)./rgb(2,1)*(29.4-0.6-0.4)*rgb_cut(2,1)
      v_B4C(3,1)/25.8*rgb(3,3)./rgb(3,1)*(29.4-0.6-0.4)*rgb_cut(3,1)
      v_B4C(4,1)/25.8*rgb(4,3)./rgb(4,1)*(29.4-0.6-0.4)*rgb_cut(4,1)];
v_tot=v_B4C+v_Al+v_H2O; % total volume for the 4 pieces
%
```

```
fprintf('c piece 1; Fig. 83 and Fig.9 of Ref. 1\n');
xyz=[-44/2 44/2 -21.1-v_tot(1,1)/(44*49) -21.1 -49/2 49/2];
fprintf('501  px  %+g\n',xyz(1,1));
fprintf('502  px  %+g\n',xyz(1,2));
fprintf('503  py  %+g\n',xyz(1,3));
fprintf('504  py  %+g\n',xyz(1,4));
fprintf('505  pz  %+g\n',xyz(1,5));
fprintf('506  pz  %+g\n',xyz(1,6));
%
c=cos(30/180*pi);
s=sin(30/180*pi);
fprintf('c piece 2; Fig. 84 and Fig.10 of Ref. 1\n');
p1=[12, -15.1-6]; % same technique as in Script 3
p2=p1+v_tot(2,1)/(29.4*46.6)*[c,-s];
p3=p2+29.4*[s,+c];
p4=p1+29.4*[s,+c];
fprintf('507  p  %+e %+e %+e\n       %+e %+e %+e\n       %+e %+e %+e\n',p1(1), p1(2),0,p2(1),p2(2),0,p2(1),p2(2),1);
fprintf('508  p  %+e %+e %+e\n       %+e %+e %+e\n       %+e %+e %+e\n',p3(1), p3(2),0,p2(1),p2(2),0,p2(1),p2(2),1);
fprintf('509  p  %+e %+e %+e\n       %+e %+e %+e\n       %+e %+e %+e\n',p3(1), p3(2),0,p4(1),p4(2),0,p4(1),p4(2),1);
fprintf('510  pz %+g\n',-46.6/2);
fprintf('511  pz %+g\n', 46.6/2);
%
fprintf('c piece 3; Fig. 85 and Fig. 11 of Ref. 1 top part\n');
p1=[-12, -15.1-6]; % same technique as in Script 3
p2=p1+v_tot(3,1)/(29.4*46.6*308/(308+322))*[-c,-s]; % 308 and 322 pixels height from GIMP in Fig. 85
p3=p2+29.4*[-s,+c];
p4=p1+29.4*[-s,+c];
fprintf('512  p  %+e %+e %+e\n       %+e %+e %+e\n       %+e %+e %+e\n',p1(1), p1(2),0,p2(1),p2(2),0,p2(1),p2(2),1);
fprintf('513  p  %+e %+e %+e\n       %+e %+e %+e\n       %+e %+e %+e\n',p3(1), p3(2),0,p2(1),p2(2),0,p2(1),p2(2),1);
fprintf('514  p  %+e %+e %+e\n       %+e %+e %+e\n       %+e %+e %+e\n',p3(1), p3(2),0,p4(1),p4(2),0,p4(1),p4(2),1);
fprintf('515  pz %+g\n',46.6/2-46.6*308/(308+322));
%
fprintf('c piece 4; Figs. 85 left and 84 central and right plots and Fig.11 of Ref. 1 bottom part\n');
p1=[-12, -15.1-6]; % same technique as in Script 3
p2=p1+v_tot(4,1)/(29.4*46.6*322/(308+322))*[-c,-s]; % 308 and 322 pixels height from GIMP in Fig. 85
p3=p2+29.4*[-s,+c];
p4=p1+29.4*[-s,+c];
fprintf('516  p  %+e %+e %+e\n       %+e %+e %+e\n       %+e %+e %+e\n',p1(1), p1(2),0,p2(1),p2(2),0,p2(1),p2(2),1);
fprintf('517  p  %+e %+e %+e\n       %+e %+e %+e\n       %+e %+e %+e\n',p3(1), p3(2),0,p2(1),p2(2),0,p2(1),p2(2),1);
fprintf('518  p  %+e %+e %+e\n       %+e %+e %+e\n       %+e %+e %+e\n',p3(1), p3(2),0,p4(1),p4(2),0,p4(1),p4(2),1);
%
fprintf('c piece 1 B4C\n');
fprintf('519 rpp  %g %g %g %g %g %g\n',-21.6/2,21.6/2,-21.1+2.3556-1,-21.1+2.3556,-22,22);
fprintf('c piece 2 B4C\n');
p0=[12,-15.1-6]; % same technique as in Script 3
p1=p0+(29.4-25.8/2-13.9)*[s,+c];
p2=p1+1*[c,-s];
p3=p2+25.8*[s,+c];
p4=p1+25.8*[s,+c];
fprintf('520  p  %+e %+e %+e\n       %+e %+e %+e\n       %+e %+e %+e\n',p1(1), p1(2),0,p2(1),p2(2),0,p2(1),p2(2),1);
fprintf('521  p  %+e %+e %+e\n       %+e %+e %+e\n       %+e %+e %+e\n',p3(1), p3(2),0,p2(1),p2(2),0,p2(1),p2(2),1);
fprintf('522  p  %+e %+e %+e\n       %+e %+e %+e\n       %+e %+e %+e\n',p3(1), p3(2),0,p4(1),p4(2),0,p4(1),p4(2),1);
fprintf('523  pz %+g\n',-22);
fprintf('524  pz %+g\n', 22);
fprintf('c pieces 3 and 4 B4C\n');
p1=[-p1(1),p1(2)];
p2=[-p2(1),p2(2)];
p3=[-p3(1),p3(2)];
p4=[-p4(1),p4(2)];
fprintf('525  p  %+e %+e %+e\n       %+e %+e %+e\n       %+e %+e %+e\n',p1(1), p1(2),0,p2(1),p2(2),0,p2(1),p2(2),1);
fprintf('526  p  %+e %+e %+e\n       %+e %+e %+e\n       %+e %+e %+e\n',p3(1), p3(2),0,p2(1),p2(2),0,p2(1),p2(2),1);
fprintf('527  p  %+e %+e %+e\n       %+e %+e %+e\n       %+e %+e %+e\n',p3(1), p3(2),0,p4(1),p4(2),0,p4(1),p4(2),1);
%
N_Al =6.02616E-02; % aluminum atomic [density atoms/cm3]
N_H2O=9.96361E-02; % water atomic [density atoms/cm3]
% atomic density of homogenized water and aluminum for the 4 pieces
% data are in an array of 4 units
N_hom=(N_Al*v_Al+N_H2O*v_H2O)./(v_Al+v_H2O)
% atoms of aluminum per atom of water
N_al_hom=(6.02616E-02/9.96361E-02)*v_Al./v_H2O
```

Appendix D: MATLAB Scripts for CUBIT Modeling

Script 1 copies and move the central fuel assembly into a hexagonal lattice in the fuel rods region

```
clear all; close all;
% 64 fuel assemblies configuration
xy=[
                    0 0 0 0 0 1 1 1 1 1
                   0 0 0 0 1 1 1 1 1 1
                  0 0 0 1 1 1 1 1 1 1
                 0 0 1 1 1 1 1 1 1 1
                0 1 1 1 1 1 1 1 1 1
               1 1 1 1 1 1 1 1 1 1
              1 1 1 1 1 1 1 1 1 0
             1 1 1 1 1 1 1 1 0 0
            1 1 1 1 1 1 1 0 0 0
];
v=11; % CHANGE ACCORDING TO THE NUMBER OF VOLUMES IN THE CUBIT MODEL
fid1=fopen('matlab_output_group.txt','w+');
p=1.36*2;          % horizontal increment - fuel pitch
h=cos(30*pi/180)*p; % vertical increment
v=65+58;
for j=1:size(xy,1)
  for i=1:size(xy,2)
    copper=0;
    center=0;
    x=(i-6)*p+(j-5)*p/2;
    y=(5-j)*h;
    if (((i==5)&&(j==5))||((i==7)&&(j==5))||((i==5)&&(j==6))||((i==6)&&(j==6)))
      copper=1;
    end
    if (((i==6)&&(j==5))||((i==1)&&(j==6)))
      center=1;
    end
    if (xy(j,i)==1)
      if ((center==0)&&(copper==0))
        fprintf(fid1,'group fuel_assembly_6_5 copy move %3d %3d  0\n',x,-y);
        fprintf(fid1,'group "fuel_assembly_%d_%d" add volume %d to %d\n',i,j,v+1,v+65);
        fprintf(fid1,'color volume  %d %d %d %d %d %d %d red\n'       ,v+2,v+2+8,v+2+2*8,v+2+3*8,v+2+4*8,v+2+5*8,v+2+6*8);
        fprintf(fid1,'color volume  %d %d %d %d %d %d %d lightcyan\n'  ,v+1,v+1+8,v+1+2*8,v+1+3*8,v+1+4*8,v+1+5*8,v+1+6*8);
        fprintf(fid1,'color volume  %d %d %d %d %d %d %d lightcyan\n'  ,v+3,v+3+8,v+3+2*8,v+3+3*8,v+3+4*8,v+3+5*8,v+3+6*8);
        fprintf(fid1,'color volume  %d %d %d %d %d %d %d blue\n'       ,v+4,v+4+8,v+4+2*8,v+4+3*8,v+4+4*8,v+4+5*8,v+4+6*8);
        fprintf(fid1,'color volume  %d %d %d %d %d %d %d blue\n'       ,v+8,v+8+8,v+8+2*8,v+8+3*8,v+8+4*8,v+8+5*8,v+8+6*8);
        fprintf(fid1,'color volume  %d %d %d %d %d %d %d darkgreen\n'  ,v+5,v+5+8,v+5+2*8,v+5+3*8,v+5+4*8,v+5+5*8,v+5+6*8);
        fprintf(fid1,'color volume  %d %d %d %d %d %d %d grey\n'       ,v+6,v+6+8,v+6+2*8,v+6+3*8,v+6+4*8,v+6+5*8,v+6+6*8);
        fprintf(fid1,'color volume  %d %d %d %d %d %d %d deepskyblue\n',v+7,v+7+8,v+7+2*8,v+7+3*8,v+7+4*8,v+7+5*8,v+7+6*8);
        fprintf(fid1,'color volume  %d %d yellow\n',v+65,v+63);
        fprintf(fid1,'color volume  %d %d %d %d %d %d %d blue\n',v+64,v+62,v+61,v+60,v+59,v+58,v+57);
        v=v+65;
      end
      if ((center==0)&&(copper==1))
        fprintf(fid1,'group fuel_assembly_1_6 copy move %3d %3d  0\n',x,-y);
        fprintf(fid1,'group "fuel_assembly_%d_%d" add volume %d to %d\n',i,j,v+1,v+58);
        fprintf(fid1,'color volume  %d %d %d %d %d %d %d magenta\n'    ,v+1,v+1+7,v+1+2*7,v+1+3*7,v+1+4*7,v+1+5*7,v+1+6*7);
        fprintf(fid1,'color volume  %d %d %d %d %d %d %d lightcyan\n'  ,v+2,v+2+7,v+2+2*7,v+2+3*7,v+2+4*7,v+2+5*7,v+2+6*7);
        fprintf(fid1,'color volume  %d %d %d %d %d %d %d blue\n'       ,v+3,v+3+7,v+3+2*7,v+3+3*7,v+3+4*7,v+3+5*7,v+3+6*7);
        fprintf(fid1,'color volume  %d %d %d %d %d %d %d darkgreen\n'  ,v+4,v+4+7,v+4+2*7,v+4+3*7,v+4+4*7,v+4+5*7,v+4+6*7);
        fprintf(fid1,'color volume  %d %d %d %d %d %d %d grey\n'       ,v+5,v+5+7,v+5+2*7,v+5+3*7,v+5+4*7,v+5+5*7,v+5+6*7);
        fprintf(fid1,'color volume  %d %d %d %d %d %d %d deepskyblue\n',v+6,v+6+7,v+6+2*7,v+6+3*7,v+6+4*7,v+6+5*7,v+6+6*7);
        fprintf(fid1,'color volume  %d %d %d %d %d %d %d blue\n'       ,v+7,v+7+7,v+7+2*7,v+7+3*7,v+7+4*7,v+7+5*7,v+7+6*7);
        fprintf(fid1,'color volume  %d %d yellow\n',v+58,v+56);
        fprintf(fid1,'color volume  %d %d %d %d %d %d blue\n',v+57,v+55,v+54,v+53,v+52,v+51,v+50);
        v=v+58;
      end
    end
  end
end
fclose(fid1);
```

Script 2 copies and move the central fuel assembly into a hexagonal lattice above the fuel rods region

```
clear all; close all;
% 64 fuel assemblies configuration
xy=[
                          0 0 0 0 0 1 1 1 1 1
                         0 0 0 0 1 1 1 1 1 1
                        0 0 0 1 1 1 1 1 1 1
                       0 0 1 1 1 1 1 1 1 1
                      0 1 1 1 1 0 1 1 1 1
                     1 1 1 1 1 1 1 1 1 1
                    1 1 1 1 1 1 1 1 1 0
                   1 1 1 1 1 1 1 1 0 0
                  1 1 1 1 1 1 1 0 0 0
];

v=4518; % CHANGE ACCORDING TO THE NUMBER OF VOLUMES IN THE CUBIT MODEL
fid1=fopen('matlab_output_topassembly.txt','w+');
p=1.36*2;          % horizontal increment - fuel pitch
h=cos(30*pi/180)*p; % vertical increment
for j=1:size(xy,1)
    for i=1:size(xy,2)
        x=(i-6)*p+(j-5)*p/2;
        y=(5-j)*h;
        if (xy(j,i)==1)
            fprintf(fid1,'top_assembly_center copy move %3d %3d  0\n',x,-y);
            fprintf(fid1,'group "top_fuel_assembly" add volume %d to %d\n',v+1,v+2);
            fprintf(fid1,'color volume  %d %d blue\n',v+1,v+2);
            v=v+2;
        end
    end
end
fclose(fid1);
```

Script 3 copy and move the central fuel assembly into a hexagonal lattice below the fuel rods region

```
clear all; close all;
% 64 fuel assemblies configuration
xy=[
                          0 0 0 0 0 1 1 1 1 1
                         0 0 0 0 1 1 1 1 1 1
                        0 0 0 1 1 1 1 1 1 1
                       0 0 1 1 1 1 1 1 1 1
                      0 1 1 1 1 0 1 1 1 1
                     1 1 1 1 1 1 1 1 1 1
                    1 1 1 1 1 1 1 1 1 0
                   1 1 1 1 1 1 1 1 0 0
                  1 1 1 1 1 1 1 0 0 0
];
v=4662; % CHANGE ACCORDING TO THE NUMBER OF VOLUMES IN THE CUBIT MODEL
fid1=fopen('matlab_output_bottomassembly.txt','w+');
p=1.36*2;          % horizontal increment - fuel pitch
h=cos(30*pi/180)*p; % vertical increment
for j=1:size(xy,1)
    for i=1:size(xy,2)
        x=(i-6)*p+(j-5)*p/2;
        y=(5-j)*h;
        if (xy(j,i)==1)
            fprintf(fid1,'bottom_hole_center copy move %3d %3d  0\n',x,-y);
            fprintf(fid1,'group "bottom_hole" add volume %d\n',v+1);
            fprintf(fid1,'color volume  %d blue\n',v+1);
            fprintf(fid1,'bottom_grid_center copy move %3d %3d  0\n',x,-y);
            fprintf(fid1,'group "bottom_grid" add volume %d\n',v+2);
            fprintf(fid1,'color volume  %d yellow\n',v+2);
            v=v+2;
        end
    end
end
fclose(fid1);
```

Appendix E: PYTHON Scripts for CUBIT

Script 1 prints the color index and volume value for all volumes
In this script: i is a counter, number_of_volumes is the number of volumes in the CUBIT model, color_index is the color index of the volume, volume is the index of the volume, cubit.get_entity_color_index is a built-in python function of CUBIT that correlates colors and volumes, and cubit.get_total_volume is a built-in python function of CUBIT that calculates the volume value (e.g. in cm^3). The output of this script was used to generate an ASCII file containing three columns of data: the index, the color index, and the volume index.

```
for  i in range(1,number_of _volumes):
    color_index = cubit.get_entity_color_index("volume",i)
    volume = cubit.get_total_volume([i])
    print "%d %d %f" % (i,color_index,volume)
```

Script 2 prints the volume and body indexes for all volumes
In this script: vol is a counter, number_of_volumes is the number of volumes in the CUBIT model, body_id is the body index of the volume, and cubit.get_owning_body is a built-in python function of CUBIT that correlates bodies and volumes. The output of this script was used to generate an ASCII file containing two columns of data: the volume index and the body index.

```
for  vol in range(1,number_of _volumes):
    body_id = cubit.get_owning_body("volume",vol)
    print(vol,body_id)
```

Appendix F: C Program to Map CUBIT Colors into SERPENT Materials

This program reads a file with two columns of integer data: the body index and the color index, generated by the PYTHON scripts in Appendix E. In addition, the program writes the material section of the SERPENT input. It is assumed that in the CUBIT model there is a unique correspondence between colors and materials.

```c
#include <stdio.h>
#include <math.h>
#include <stdlib.h>
#include <errno.h>
#include <string.h>
#include <ctype.h>

void finish(unsigned int error);
char *mapcolor(unsigned int i,unsigned int j);

int main(int argc,char *argv[])
{
FILE            *fi=NULL,*fo=NULL;
char            line[20],*head,*material=NULL;
unsigned int       vol=9999999,color,k;

if (argc<3) finish(1);
if ((fi=fopen(argv[1],"r" ))==NULL) finish(2);
if ((fo=fopen(argv[2],"w+"))==NULL) finish(3);

fprintf(fo,"%% finput = %s\n",argv[1]);
fprintf(fo,"%% foutput= %s\n",argv[2]);

k=0;
while (1)
  {
  if (fgets(line,20,fi)==NULL) break;
  k++;
  head=strtok(line," ");
  sscanf(head,"%d",&vol);
  vol=atoi(head);
  if ((vol<=0)||(vol==9999999)) finish(4);
  if (isdigit(vol)==1)         finish(5);
  head=strtok(NULL," ");
  sscanf(head,"%d",&color);
  color=atoi(head);
  if ((color<=0)||(color>60))   finish(6);
  if (isdigit(color)==1)       finish(7);
  material=mapcolor(color,vol);
  fprintf(fo,"body Body_%d solidcell %s\n",k,material);
  }
printf("The file %s contains %d volumes\n",argv[1],k);
fclose(fi);
fclose(fo);
}

char *mapcolor(unsigned int i,unsigned int j)
{
char *mat;

mat="not_found";
if (i==3 )    mat="m1 ";
if (i==8 )    mat="m2 ";
if (i==1 )    mat="m3 ";
if (i==16)    mat="m4 ";
if (i==23)    mat="m5 ";
if (i==57)    mat="m6 ";
if (i==4 )    mat="m7 ";
if (i==5 )    mat="m10";
if (i==2 )    mat="m11";
if (i==11)    mat="m12";
if (i==7 )    mat="m13";
if (i==33)    mat="m14";
```

```
if (i==58)    mat="m15";
if (i==6 )    mat="m23";
if (i==37)    mat="m17";
if (i==26)    mat="m18";
if (i==17)    mat="m22";
if (i==24)    mat="m2x";
if ((j==4491)||(j==4489)||(j==4490)||(j==4514)) mat="m8 ";
if (j==4478) mat="m16";
if (j==4493) mat="m19";
if (j==4494) mat="m20";
if (j==4495) mat="m21";
return(mat);
}

/*
mat cubitcolor     cubitvolume        cubitcolor mcnp
m1  red           2                    3       $ Fuel PuO2 rho=-10.103 to have 185 g mass per rod (p.24 fuel report)
m2  blue          4                    8       $ Steel cladding fuel element FE stationary reflector SR rho=0.08495329
m3  grey          6                    1       $ Tungsten alloy top fuel rod rho=0.06890289
m4  pink          4475                 16      $ Steel control rods rho=0.05999262
m5  seagreen      4484                 23      $ Steel control rods cr1 cr2 rho=0.00726794
m6  palegreen     4469                 57      $ Fe and BC4 thermal shield rho=0.07453
m7  green         4498                 4       $ Nickel alloy rotating reflector MR rho=0.0915681
m8  red           4491 4489 4490 4514    3       $ BC4 rho=0.0926375
m10 yellow        1291                 5       $ Sodium rho=-0.873994792
m11 orange        4452                 2       $ =0.0065*2.28938E-02/2.47347E-02 ; Iron and sodium bottom reflector rho=0.0676
m12 gold          4465                 11      $ =0.007*2.28938E-02/2.47347E-02 ; Iron and sodium top reflector rho=0.047
m13 cyan          4517                 7       $ Water rho=-0.9977735
m14 lightcyan     1                    33      $ Helium rho=-0.1664e-3
m15 lightyellow   4456                 58      $ Air rho=-1.29e-3
m16 magenta       4478                 6       $ Beryllium rho=-1.85
m17 darkgreen     5                    37      $ Steel 69.2% sodium 30.8% in bottom fuel rod rho=6.6406e-2
m18 lightskyblue  4492                 26      $ Al 13027.80c   4.8502e-001
m19 lightskyblue  4493                 26      $ Al 13027.80c   4.3367e-001
m20 lightskyblue  4494                 26      $ Al 13027.80c   5.9943e-001
m21 lightskyblue  4495                 26      $ Al 13027.80c   4.8876e-001
m22 purple        4513 4515 4516        17      $ Aluminum rho=-2.7
m23 magenta       66                   6       $ dummy fuel assembly copper rho=-8.96
m24 same as m14                                $ Helium rho=-0.1664e-3
m25 same as m2                                 $ Steel cladding fuel element FE stationary reflector SR rho=0.08495329
mXX deepskyblue   7                    24      $ steel spring reduced density
FIX DUPLICATE RED AND MAGENTA
#ffd600 in cubit gold   top
#ffa400 in cubit orange bottom
python script:
for i in range(1,10):
..      color_index = cubit.get_entity_color_index("volume",i)
..      print(i,color_index)
*/
void finish(unsigned int error)
{
if (error==1)
   {
   printf("Usage \"bodies input output\"\n");
   }
else
   {
   printf("The program error is %d\n", error);
   printf("The system  error is %d\n", errno);
   perror("Execution aborted.\n");
   }
exit(error);


}
```

Nuclear Science and Engineering Division
Argonne National Laboratory
9700 South Cass Avenue, Bldg. 208
Argonne, IL 60439

www.anl.gov