Argonne NATIONAL LABORATORY

ANL-NE-10/17

Deterministic and Monte Carlo Modeling and Analyses of Yalina-Thermal Subcritical Assembly

Nuclear Engineering Division

About Argonne National Laboratory
Argonne is a U.S. Department of Energy laboratory managed by UChicago Argonne, LLC under contract DE-AC02-06CH11357. The Laboratory's main facility is outside Chicago, at 9700 South Cass Avenue, Argonne, Illinois 60439. For information about Argonne, see www.anl.gov.

Availability of This Report
This report is available, at no cost, at http://www.osti.gov/bridge. It is also available on paper to the U.S. Department of Energy and its contractors, for a processing fee, from:

U.S. Department of Energy
Office of Scientific and Technical Information
P.O. Box 62
Oak Ridge, TN 37831-0062
phone (865) 576-8401
fax (865) 576-5728
reports@adonis.osti.gov

Disclaimer
This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency thereof, nor UChicago Argonne, LLC, nor any of their employees or officers, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise, does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof. The views and opinions of document authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof, Argonne National Laboratory, or UChicago Argonne, LLC.

June 2010 ANL-NE-10/17

DETERMINISTIC AND MONTE CARLO MODELING AND ANALYSES OF YALINA-THERMAL SUBCRITICAL ASSEMBLY

By

Alberto Talamo and Yousry Gohar

Nuclear Engineering Division
Argonne National Laboratory
9700 South Cass Avenue
Argonne, IL 60439

Work supported by the
Office of Global Nuclear Material Threat Reduction
U.S. Department of Energy
Under Contract to DE-AC02-06CH11357

DETERMINISTIC AND MONTE CARLO MODELING AND ANALYSES OF YALINA-THERMAL SUBCRITICAL ASSEMBLY

Table of Contents

Page

Abstract 1

1. Introduction 2

2. The YALINA thermal Assembly 3
   2.A. Monte Carlo Modeling 3
   2.B Deterministic Modeling 4
   2.C External Neutron Source Modeling 7
      2.C.1 Californium Neutron Source 7
      2.C.2 Deuterium-Deuterium Neutron Source 7
      2.C.3 Deuterium-Tritium Neutron Source 8

3. Fuel and Moderator Unit Cell 8

4. Neutron Multiplication Factor 9
   4.A Criticality Analyses 9
   4.B Californium Neutron Source 10
   4.C Deuterium-Deuterium Neutron Source 11
   4.D Deuterium-Tritium Neutron Source 11

5. Delayed Neutron Fraction 12
   5.A Adjoint-Weighted (Effective) Delayed Neutron Fraction 12
   5.B Adjoint-Unweighted (Average) Delayed Neutron Fraction 16

6. Neutron Lifetime 17
   6.A Adjoint-Weighted Prompt Neutron Lifetime 17
   6.B Adjoint-Unweighted Prompt Neutron Lifetime 19
   6.C Adjoint-Weighted Neutron Lifetime 19
   6.D Adjoint-Unweighted Neutron Lifetime 20

7. Mean Neutron Generation Time 20

8. Spatial Neutron Flux Distributions 21
   8.A MCNP/MCNPX Calculations for Cf, D-D, and D-T Neutron Sources 21
   8.B MCNP/MCNPX, MONK, PARTISN, and TORT Calculations for Criticality and D-D Neutron Source 21

9. Neutron Spectrum 22
   9.A MCNP/MCNPX Calculations for Cf, D-D, and D-T Neutron Sources 22
   9.B MCNP/MCNPX, MONK, PARTISN, and TORT Calculations for Criticality and D-D Neutron Source 23

10. Reaction Rate Spatial Distribution 24
10.A Criticality Calculations 24
10.B Californium, D-D and D-T Neutron Sources Calculations 24
10.C Comparison between Numerical and Experimental Results for Cf Neutron Source .. 25

11. Time Dependent Reaction Rates from D-D and D-T Pulsed Neutron Sources 26
11.A MCNP/MCNPX Results for D-D and D-T Pulsed Neutron Sources Calculations 26
11.B Comparison between MCNP/MCNPX and Experimental Results for D-D Pulsed Neutron Source 27
11.C Comparison between MCNP/MCNPX and PARTISN Results for D-D Pulsed Neutron Source 27
11.D Bell and Glasstone Correction Factors for Reactivity Measurements 28

12. Conclusions 31

Acknowledgements 32

References 33

Appendix A: DRAGON Modeling 146

Appendix B: Alternative Definition of the Kinetics Parameters for Subcritical Systems 150

Appendix C: DRAGON2PARTISN Source Code 152

Appendix D: DRAGON2TORT Source Code 167

Appendix E: Source Code for the Calculation of the Effective Delayed Neutron Fraction ... 182

Appendix F: Source Code for the Calculation of the Prompt Neutron Generation Time 234

Appendix G: Source Code for the Pulse Superimposition 238

Appendix H: Sample for Rate.txt 244

Appendix I: Sample for Time.txt 246

DETERMINISTIC AND MONTE CARLO MODELING AND ANALYSES OF YALINA-THERMAL SUBCRITICAL ASSEMBLY

List of Figures

Figure No. Page

Figure 1. YALINA thermal facility 44
Figure 2. External view of the YALINA thermal subcritical assembly model 45
Figure 3. External view of the YALINA thermal subcritical assembly model with corner cut 46
Figure 4. Zoom of the external view of the YALINA thermal subcritical assembly model with corner cut 47
Figure 5. External view of the YALINA thermal subcritical assembly model with left half-side cut 48
Figure 6. External view of the YALINA thermal subcritical assembly model with right half-side cut 49
Figure 7. Horizontal section of the YALINA thermal subcritical assembly model at the middle of the active fuel height 50
Figure 8. Horizontal section of the YALINA thermal subcritical assembly model at the height of the borated polyethylene axial reflector 51
Figure 9. Vertical section of the YALINA thermal subcritical assembly model 52
Figure 10. Zoom of the vertical section of the YALINA thermal subcritical assembly model 53
Figure 11. Horizontal section of the YALINA thermal subcritical assembly model with 216 fuel rods 54
Figure 12. Horizontal section of the YALINA thermal subcritical assembly model with 245 fuel rods 55
Figure 13. Horizontal section of the YALINA thermal subcritical assembly model with 280 fuel rods 56
Figure 14. Horizontal section of the YALINA thermal subcritical assembly MCNP/MCNPX model 57
Figure 15. Zoom of the horizontal section of the YALINA thermal subcritical assembly MCNP/MCNPX model 58
Figure 16. Vertical section of the YALINA thermal subcritical assembly MCNP/MCNPX model 59
Figure 17. Zoom of the vertical section of the YALINA thermal subcritical assembly MCNP/MCNPX model 60
Figure 18. Vertical section of the YALINA thermal subcritical assembly MCNP/MCNPX model at x=0 61
Figure 19. PARTISN model of YALINA Thermal 62

Figure 20. PARTISN model of YALINA Thermal with right side corner cut 63
Figure 21. PARTISN model of YALINA Thermal with left side corner cut 64
Figure 22. PARTISN model of YALINA Thermal with corner cut between z=0 and z=12.4 ..65
Figure 23. PARTISN model of YALINA Thermal with corner cut between z=12.4 and z=35 66
Figure 24. PARTISN model of YALINA Thermal with corner cut between z=35 and z=35.4 67
Figure 25. PARTISN model of YALINA Thermal with corner cut between z=35.4 and z=39.9 68
Figure 26. PARTISN model of YALINA Thermal with corner cut between z=39.9 and z=64.9 69
Figure 27. PARTISN model of YALINA Thermal with corner cut between z=64.9 and z=66.35 70
Figure 28. PARTISN model of YALINA Thermal with corner cut between z=66.35 and z=71.4 71
Figure 29. PARTISN model of YALINA Thermal with corner cut between z=71.4 and z=89.9 72
Figure 30. PARTISN model of YALINA Thermal with corner cut between z=89.9 and z=93 73
Figure 31. PARTISN model of YALINA Thermal with corner cut between z=93 and z=94.4 74
Figure 32. PARTISN model of YALINA Thermal with corner cut between z=94.4 and z=94.5 75
Figure 33. PARTISN model of YALINA Thermal with corner cut between z=94.5 and z=103.5 76
Figure 34. Angular distribution of the D-D neutrons obtained with 250 KeV deuterons, the deuteron beam is at 270° with upward direction (the vertical section of the YALINA thermal assembly has been added as semi-transparent background) ...77
Figure 35. Angular distribution of D-T neutrons obtained with 250 KeV deuterons, the deuteron beam is at 270° with upward direction 77
Figure 36. Microscopic differential cross section of D-D and D-T neutrons obtained with 250 KeV deuterons, the angle is between the deuteron beam and the neutron flight path (0° in the current figure corresponds to 90° in Figures 34 and 35 and it means the neutron is emitted in the forward direction relative to the deuteron beam direction) 78
Figure 37. Energy of D-D and D-T neutrons obtained with 250 KeV deuterons, the angle is between the deuteron beam and the neutron flight path (0° in the current Figure corresponds to 90° in Figures 34 and 35 and it means the neutron is emitted in the forward direction relative to the deuteron beam direction) 78
Figure 38. Delayed neutron fraction as a function of incident neutron energy 79
Figure 39. Total number of fission neutrons as a function of incident neutron energy 79

Figure 40. Total macroscopic cross section of the fuel and moderator unit cell 80
Figure 41. Absorption macroscopic cross section of the fuel and moderator unit cell 80
Figure 42. Fission macroscopic cross section of the fuel and moderator unit cell 81
Figure 43. Fission neutrons multiplied by the fission macroscopic cross section of the fuel and moderator unit cell 81
Figure 44. Neutron spectra of the fuel and moderator unit cell 82
Figure 45. Neutron flux in the EC1 experimental channel calculated by MCNPX 82
Figure 46. Neutron flux in the EC2 experimental channel calculated by MCNPX 83
Figure 47. Neutron flux in the EC3 experimental channel calculated by MCNPX 83
Figure 48. Neutron flux in the EC5 experimental channel calculated by MCNPX 84
Figure 49. Neutron flux in the EC6 experimental channel calculated by MCNPX 84
Figure 50. Neutron flux in the EC7 experimental channel calculated by MCNPX 85
Figure 51. Neutron flux in the EC1 experimental channel calculated by MCNPX with Cf neutron source 85
Figure 52. Neutron flux in the EC2 experimental channel calculated by MCNPX with Cf neutron source 86
Figure 53. Neutron flux in the EC3 experimental channel calculated by MCNPX with Cf neutron source 86
Figure 54. Neutron flux in the EC5 experimental channel calculated by MCNPX with Cf neutron source 87
Figure 55. Neutron flux in the EC6 experimental channel calculated by MCNPX with Cf neutron source 87
Figure 56. Neutron flux in the EC7 experimental channel calculated by MCNPX with Cf neutron source 88
Figure 57. Neutron flux in the EC1 experimental channel from MCNPX and PARTISN criticality calculation for the YALINA thermal with 216 fuel rods 88
Figure 58. Neutron flux in the EC2 experimental channel from MCNPX and PARTISN criticality calculation for the YALINA thermal with 216 fuel rods 89
Figure 59. Neutron flux in the EC3 experimental channel from MCNPX and PARTISN criticality calculation for the YALINA thermal with 216 fuel rods 89
Figure 60. Neutron flux in the EC5 experimental channel from MCNPX and PARTISN criticality calculation for the YALINA thermal with 216 fuel rods 90
Figure 61. Neutron flux in the EC6 experimental channel from MCNPX and PARTISN criticality calculation for the YALINA thermal with 216 fuel rods 90
Figure 62. Neutron flux in the EC7 experimental channel from MCNPX and PARTISN criticality calculation for the YALINA thermal with 216 fuel rods 91
Figure 63. Neutron flux in the EC1 experimental channel from MCNPX and PARTISN criticality calculation for the YALINA thermal with 280 fuel rods 91
Figure 64. Neutron flux in the EC2 experimental channel from MCNPX and PARTISN criticality calculation for the YALINA thermal with 280 fuel rods 92

Figure 65. Neutron flux in the EC3 experimental channel from MCNPX and PARTISN criticality calculation for the YALINA thermal with 280 fuel rods 92
Figure 66. Neutron flux in the EC5 experimental channel from MCNPX and PARTISN criticality calculation for the YALINA thermal with 280 fuel rods 93
Figure 67. Neutron flux in the EC6 experimental channel from MCNPX and PARTISN criticality calculation for the YALINA thermal with 280 fuel rods 93
Figure 68. Neutron flux in the EC7 experimental channel from MCNPX and PARTISN criticality calculation for the YALINA thermal with 280 fuel rods 94
Figure 69. Neutron flux in the EC1 experimental channel calculated by MCNPX, MONK, PARTISN, and TORT for the YALINA thermal with 216 fuel rods and D-D neutron source.....94
Figure 70. Neutron flux in the EC2 experimental channel calculated by MCNPX, MONK, PARTISN, and TORT for the YALINA thermal with 216 fuel rods and D-D neutron source.....95
Figure 71. Neutron flux in the EC3 experimental channel calculated by MCNPX, MONK, PARTISN, and TORT for the YALINA thermal with 216 fuel rods and D-D neutron source.....95
Figure 72. Neutron flux in the EC5 experimental channel calculated by MCNPX, MONK, PARTISN, and TORT for the YALINA thermal with 216 fuel rods and D-D neutron source.....96
Figure 73. Neutron flux in the EC6 experimental channel calculated by MCNPX, MONK, PARTISN, and TORT for the YALINA thermal with 216 fuel rods and D-D neutron source.....96
Figure 74. Neutron flux in the EC7 experimental channel calculated by MCNPX, MONK, PARTISN, and TORT for the YALINA thermal with 216 fuel rods and D-D neutron source.....97
Figure 75. Neutron spectra in the EC1 experimental channel calculated by MCNPX with Cf neutron source 97
Figure 76. Neutron spectra in the EC2 experimental channel calculated by MCNPX with Cf neutron source 98
Figure 77. Neutron spectra in the EC3 experimental channel calculated by MCNPX with Cf neutron source 98
Figure 78. Neutron spectra in the EC5 experimental channel calculated by MCNPX with Cf neutron source 99
Figure 79. Neutron spectra in the EC6 experimental channel calculated by MCNPX with Cf neutron source 99
Figure 80. Neutron spectra in the EC7 experimental channel calculated by MCNPX with Cf neutron source 100
Figure 81. Neutron spectra in the EC1 experimental channel calculated by MCNPX with Cf neutron source and with and without the beam tube.....100
Figure 82. Neutron spectra in the EC1 experimental channel calculated by MCNPX with D-D neutron source 101

Figure 83. Neutron spectra in the EC2 experimental channel calculated by MCNPX with D-D neutron source 101
Figure 84. Neutron spectra in the EC3 experimental channel calculated by MCNPX with D-D neutron source 102
Figure 85. Neutron spectra in the EC5 experimental channel calculated by MCNPX with D-D neutron source 102
Figure 86. Neutron spectra in the EC6 experimental channel calculated by MCNPX with D-D neutron source 103
Figure 87. Neutron spectra in the EC7 experimental channel calculated by MCNPX with D-D neutron source 103
Figure 88. Neutron spectra in the EC1 experimental channel calculated by MCNPX with D-T neutron source.....104
Figure 89. Neutron spectra in the EC2 experimental channel calculated by MCNPX with D-T neutron source.....104
Figure 90. Neutron spectra in the EC3 experimental channel calculated by MCNPX with D-T neutron source.....105
Figure 91. Neutron spectra in the EC5 experimental channel calculated by MCNPX with D-T neutron source.....105
Figure 92. Neutron spectra in the EC6 experimental channel calculated by MCNPX with D-T neutron source.....106
Figure 93. Neutron spectra in the EC7 experimental channel calculated by MCNPX with D-T neutron source.....106
Figure 94. Neutron spectra in the EC1 experimental channel of the YALINA thermal with 216 fuel rods from criticality calculation by MCNPX and PARTISN 107
Figure 95. Neutron spectra in the EC2 experimental channel of the YALINA thermal with 216 fuel rods from criticality calculation by MCNPX and PARTISN 107
Figure 96. Neutron spectra in the EC3 experimental channel of the YALINA thermal with 216 fuel rods from criticality calculation by MCNPX and PARTISN 108
Figure 97. Neutron spectra in the EC5 experimental channel of the YALINA thermal with 216 fuel rods from criticality calculation by MCNPX and PARTISN 108
Figure 98. Neutron spectra in the EC6 experimental channel of the YALINA thermal with 216 fuel rods from criticality calculation by MCNPX and PARTISN 109
Figure 99. Neutron spectra in the EC7 experimental channel of the YALINA thermal with 216 fuel rods from criticality calculation by MCNPX and PARTISN 109
Figure 100. Neutron spectra in the EC1 experimental channel of the YALINA thermal with 280 fuel rods from criticality calculation by MCNPX and PARTISN 110
Figure 101. Neutron spectra in the EC2 experimental channel of the YALINA thermal with 280 fuel rods from criticality calculation by MCNPX and PARTISN 110
Figure 102. Neutron spectra in the EC3 experimental channel of the YALINA thermal with 280 fuel rods from criticality calculation by MCNPX and PARTISN 111

Figure 103. Neutron spectra in the EC5 experimental channel of the YALINA thermal with 280 fuel rods from criticality calculation by MCNPX and PARTISN 111

Figure 104. Neutron spectra in the EC6 experimental channel of the YALINA thermal with 280 fuel rods from criticality calculation by MCNPX and PARTISN 112

Figure 105. Neutron spectra in the EC7 experimental channel of the YALINA thermal with 280 fuel rods from criticality calculation by MCNPX and PARTISN 112

Figure 106. Neutron spectra in the EC1 experimental channel of the YALINA thermal with 216 fuel rods with D-D neutron source calculated by MCNPX, MONK, PARTISN, and TORT.....113

Figure 107. Neutron spectra in the EC2 experimental channel of the YALINA thermal with 216 fuel rods with D-D neutron source calculated by MCNPX, MONK, PARTISN, and TORT.....113

Figure 108.Neutron spectra in the EC3 experimental channel of the YALINA thermal with 216 fuel rods with D-D neutron source calculated by MCNPX, MONK, PARTISN, and TORT.....114

Figure 109. Neutron spectra in the EC5 experimental channel of the YALINA thermal with 216 fuel rods with D-D neutron source calculated by MCNPX, MONK, PARTISN, and TORT.....114

Figure 110. Neutron spectra in the EC6 experimental channel of the YALINA thermal with 216 fuel rods with D-D neutron source calculated by MCNPX, MONK, PARTISN, and TORT.....115

Figure 111. Neutron spectra in the EC7 experimental channel of the YALINA thermal with 216 fuel rods with D-D neutron source calculated by MCNPX, MONK, PARTISN, and TORT.....115

Figure 112. ^{3}He reaction rate in the EC1 experimental channel of the YALINA thermal with 216 fuel rods from criticality calculation by MCNPX and PARTISN.....116

Figure 113. ^{3}He reaction rate in the EC2 experimental channel of the YALINA thermal with 216 fuel rods from criticality calculation by MCNPX and PARTISN.....116

Figure 114. ^{3}He reaction rate in the EC3 experimental channel of the YALINA thermal with 216 fuel rods from criticality calculation by MCNPX and PARTISN.....117

Figure 115. ^{3}He reaction rate in the EC1 experimental channel of the YALINA thermal with 280 fuel rods from criticality calculation by MCNPX and PARTISN.....117

Figure 116. ^{3}He reaction rate in the EC2 experimental channel of the YALINA thermal with 280 fuel rods from criticality calculation by MCNPX and PARTISN.....118

Figure 117. ^{3}He reaction rate in the EC3 experimental channel of the YALINA thermal with 280 fuel rods from criticality calculation by MCNPX and PARTISN.....118

Figure 118. ^{3}He(n,p) reaction rate per source particle in the EC1 experimental channel with Cf neutron source calculated by MCNPX, PARTISN, and TORT 119

Figure 119. ^{3}He(n,p) reaction rate per source particle in the EC1 experimental channel with D-D neutron source calculated by MCNPX, PARTISN, and TORT 119

Figure 120. ^{3}He(n,p) reaction rate per source particle in the EC2 experimental channel with Cf neutron source calculated by MCNPX, PARTISN, and TORT 120

Figure 121. ^{3}He(n,p) reaction rate per source particle in the EC2 experimental channel with D-D neutron source calculated by MCNPX, PARTISN, and TORT 120
Figure 122. ^{3}He(n,p) reaction rate per source particle in the EC3 experimental channel with Cf neutron source calculated by MCNPX, PARTISN, and TORT 121
Figure 123. ^{3}He(n,p) reaction rate per source particle in the EC3 experimental channel with D-D neutron source calculated by MCNPX, PARTISN, and TORT 121
Figure 124. ^{3}He(n,p) reaction rate per source particle with D-T neutron source calculated by MCNPX 122
Figure 125. ^{235}U (n,f) reaction rate per source particle in the EC1 experimental channel with Cf neutron source calculated by MCNPX, PARTISN, and TORT 122
Figure 126. ^{235}U (n,f) reaction rate per source particle in the EC1 experimental channel with D-D neutron source calculated by MCNPX, PARTISN, and TORT 123
Figure 127. ^{235}U (n,f) reaction rate per source particle in the EC2 experimental channel with Cf neutron source calculated by MCNPX, PARTISN, and TORT 123
Figure 128. ^{235}U (n,f) reaction rate per source particle in the EC2 experimental channel with D-D neutron source calculated by MCNPX, PARTISN, and TORT 124
Figure 129. ^{235}U (n,f) reaction rate per source particle in the EC3 experimental channel with Cf neutron source calculated by MCNPX, PARTISN, and TORT 124
Figure 130. ^{235}U (n,f) reaction rate per source particle in the EC3 experimental channel with D-D neutron source calculated by MCNPX, PARTISN, and TORT 125
Figure 131. ^{235}U (n,f) reaction rate per source particle with D-T neutron source calculated by MCNPX 125
Figure 132. In (n,γ) reaction rate per source particle in EC2 experimental channel with Cf neutron source calculated by MCNPX and PARTISN.....126
Figure 133. In (n,γ) reaction rate per source particle in EC2 experimental channel with D-D neutron source calculated by MCNPX and PARTISN 126
Figure 134. In (n,γ) reaction rate per source particle in EC7 experimental channel with Cf neutron source calculated by MCNPX and PARTISN.....127
Figure 135. In (n,γ) reaction rate per source particle in EC7 experimental channel with D-D neutron source calculated by MCNPX and PARTISN 127
Figure 136. ^{197}Au (n,γ) reaction rate per source particle in the EC2 experimental channel with Cf neutron source calculated by MCNPX and PARTISN.....128
Figure 137. ^{197}Au (n,γ) reaction rate per source particle in the EC2 experimental channel with D-D neutron source calculated by MCNPX and PARTISN 128
Figure 138. ^{55}Mn (n,γ) reaction rate per source particle in the EC2 experimental channel with Cf neutron source calculated by MCNPX and PARTISN.....129
Figure 139. ^{55}Mn (n,γ) reaction rate per source particle in the EC2 experimental channel with D-D neutron source calculated by MCNPX and PARTISN 129
Figure 140. In (n,γ) reaction rate per source particle in the EC2 experimental channel with explicit modeling of the irradiation samples calculated by MCNPX.....130

Figure 141. In (n,γ) reaction rate per source particle in the EC7 experimental channel with explicit modeling of the irradiation samples calculated by MCNPX.....130

Figure 142. ^{197}Au (n,γ) reaction rate per source particle in the EC2 experimental channel with explicit modeling of the irradiation samples calculated by MCNPX 131

Figure 143. ^{55}Mn (n,γ) reaction rate per source particle in the EC2 experimental channel with explicit modeling of the irradiation samples calculated by MCNPX 131

Figure 144. Comparison of MCNP and PARTISN neutron flux in the EC1 experimental channel of YALINA thermal with 245 fuel rods from Cf neutron source 132

Figure 145. Comparison of MCNP, PARTISN and experimental results for the ^{3}He reaction rate in the EC1 experimental channel of YALINA thermal with 245 fuel rods from Cf neutron source 132

Figure 146. Comparison of MCNP and PARTISN neutron flux in the EC2 experimental channel of the YALINA thermal with 245 fuel rods from Cf neutron source 133

Figure 147. Comparison of MCNP, PARTISN and experimental results for the ^{3}He reaction rate in the EC2 experimental channel of the YALINA thermal with 245 fuel rods from Cf neutron source 133

Figure 148. Comparison of MCNP and PARTISN neutron flux in the EC3 experimental channel of the YALINA thermal with 245 fuel rods from Cf neutron source 134

Figure 149. Comparison of MCNP, PARTISN and experimental results for the ^{3}He reaction rate in the EC3 experimental channel of the YALINA thermal with 245 fuel rods from Cf neutron source 134

Figure 150. Comparison of MCNP and PARTISN neutron flux in the EC1 experimental channel of the YALINA thermal with 280 fuel rods from Cf neutron source 135

Figure 151. Comparison of MCNP, PARTISN and experimental results for the ^{3}He reaction rate in the EC1 experimental channel of the YALINA thermal with 280 fuel rods from Cf neutron source 135

Figure 152. Comparison of MCNP and PARTISN neutron flux in the EC2 experimental channel of the YALINA thermal with 280 fuel rods from Cf neutron source 136

Figure 153. Comparison of MCNP, PARTISN and experimental results for the ^{3}He reaction rate in the EC2 experimental channel of the YALINA thermal with 280 fuel rods from Cf neutron source 136

Figure 154. Comparison of MCNP and PARTISN neutron flux in the EC3 experimental channel of the YALINA thermal with 280 fuel rods from Cf neutron source 137

Figure 155. Comparison of MCNP, PARTISN and experimental results for the ^{3}He reaction rate in the EC3 experimental channel of the YALINA thermal with 280 fuel rods from Cf neutron source 137

Figure 156. ^{3}He(n,p) and ^{235}U(n,f) reaction rates in the YALINA thermal with 216 fuel rods from D-D and D-T pulsed neutron sources 138

Figure 157. ^{3}He(n,p) and ^{235}U(n,f) reaction rates in the YALINA thermal with 245 fuel rods from D-D and D-T pulsed neutron sources 138

Figure 158. ^{3}He(n,p) and ^{235}U(n,f) reaction rates in the YALINA thermal with 280 fuel rods from D-D and D-T pulsed neutron sources 139

Figure 159. ^{3}He(n,p) reaction rate in the EC2 experimental channel of YALINA thermal with 245 fuel rods from D-D pulsed neutron source 139
Figure 160. ^{3}He(n,p) reaction rate in the EC5 experimental channel of YALINA thermal with 245 fuel rods from D-D pulsed neutron source 140
Figure 161. ^{3}He(n,p) reaction rate in the EC2 experimental channel of the YALINA thermal with 280 fuel rods from D-D pulsed neutron source 140
Figure 162. ^{3}He(n,p) reaction rate in the EC5 experimental channel of YALINA thermal with 280 fuel rods from D-D pulsed neutron source 141
Figure 163. Comparison of MCNP/MCNPX and PARTISN results for ^{3}He(n,p) reaction rate in the EC2 experimental channel of YALINA thermal with 216 fuel rods from D-D single pulse 141
Figure 164. Comparison of MCNP/MCNPX and PARTISN results for ^{3}He(n,p) reaction rate in the EC5 experimental channel of YALINA thermal with 216 fuel rods from D-D single pulse 142
Figure 165. Comparison of MCNP/MCNPX and PARTISN results for ^{3}He(n,p) reaction rate in the EC2 experimental channel of YALINA thermal with 280 fuel rods from D-D single pulse 142
Figure 166. Comparison of MCNP/MCNPX and PARTISN results for ^{3}He(n,p) reaction rate in the EC5 experimental channel of YALINA thermal with 280 fuel rods from D-D single pulse 143
Figure 167. ^{3}He(n,p) microscopic cross section from MCNPX calculations and from ENDF/B-VI.8 nuclear data files 143
Figure 168. Illustration of A_p and A_d of the area method 144
Figure 169. Correction factor as a function of the axial position calculated by PARTISN for the 245 configuration and the D-D neutron source 144
Figure 170. Correction factor as a function of the axial position calculated by PARTISN for the 280 configuration and the D-D neutron source 145
Figure A1. DRAGON geometry for the fuel and moderator unit cell 146
Figure A2. DRAGON geometry for one eight of the subcritical assembly 147
Figure A3. Horizontal section in the lead zone of the DRAGON geometry for the target region 148
Figure A4. Vertical section along the plane indicated by the red line of figure A3 for the target region 149

DETERMINISTIC AND MONTE CARLO MODELING AND ANALYSES OF YALINA-THERMAL SUBCRITICAL ASSEMBLY

List of Tables

Table No. Page

Table I: Neutron multiplication factor (k_{eff}); the standard deviation is reported in pcm units 39

Table II: Neutron source multiplication factor for the Californium neutron source with the beam tube; the standard deviation is reported in pcm units 39

Table III: Neutron source multiplication factor for the Californium neutron source without the beam tube; the standard deviation is reported in pcm units 39

Table IV: Neutron source multiplication factor for the D-D neutron source; $(E,\vec{\Omega})$ indicates that the energy-angle distribution (Figure 37) of the neutron source is modeled; the standard deviation is reported in pcm units 40

Table V: Neutron source multiplication factor for the D-T neutron source; $(E,\vec{\Omega})$ indicates that the energy-angle distribution (Figure 37) of the neutron source is modeled; the standard deviation is reported in pcm units 40

Table VI: Delayed neutron fraction [pcm] as a function of the number of the fuel rods calculated with MCNP - ENDF/B-VI.6 and PARTISN – ENDF/B-VI.8 41

Table VII: Prompt neutron lifetime [μs] as a function of the number of the fuel rods 41

Table VIII: Correction factors calculated by MCNP/MCNPX for the 245 and 280 configurations for ^{3}He detector, and D-D neutron source. All the modeled detectors have 0.45 cm radius and 25 cm length. For the two detectors case, the detectors are located in EC2 and EC3 or at EC5 and EC6 experimental channels 42

Table IX: Correction factor calculated by PARTISN for the 245 and 280 configurations for ^{3}He detector, and the D-D neutron source with static (source mode) simulations without detector modeling 42

Table X: Neutron multiplication factor (k_{eff}) obtained by MCNP/MCNPX (with ENDF/B-VI.6 nuclear data files) and EA-MC[109] (with JENDL-3.2 nuclear data library) for the 245 and 280 configurations. All the modeled detectors have 0.45 cm radius and 25 cm length. In the case two detectors are used the detectors are located at EC2 and EC3 or at EC5 and EC6 experimental channels. The standard deviation is reported in pcm 42

Table XI: Neutron multiplication factor (k_{eff}) obtained by pulsed neutron source experiments using the area method and two detectors. The detectors are located in the experimental channels EC2 and EC3 or EC5 and EC6. The detector acquiring the signal (EC2 or EC5) has 0.45 cm radius and 25 cm height. The error is reported in pcm units 43

Table XII: EK10 fuel composition (weight fractions) 43

Table B.I. Source delayed neutron fraction [pcm] as function of the fuel rods number calculated using PARTISN5.86, nuclear data library based on ENDF/B-VI.8 nuclear data files, and Equations B2 and B3 151

Table B.II. Source prompt neutron lifetime [μs] as function of the fuel rods number calculated using PARTISN5.86, nuclear data library based on ENDF/B-VI.8 nuclear data files, and Equations B1 and B3 151

DETERMINISTIC AND MONTE CARLO MODELING AND ANALYSES OF YALINA-THERMAL SUBCRITICAL ASSEMBLY

Abstract

The YALINA thermal facility of the Joint Institute for Power & Nuclear Research – SOSNY, Belarus consists of a subcritical assembly driven by a californium, deuterium-deuterium, or deuterium-tritium neutron source. This study analyzes the YALINA thermal assembly using deterministic and Monte Carlo calculation methodologies. In the deterministic analyses, the DRAGON computer program has been used to generate the macroscopic cross sections starting from the WIMSD nuclear data library with 69 energy groups. The WIMSD nuclear data library has been modified by adding few isotopes describing the polyethylene material, some impurity nuclides, and the uranium nuclides without delayed neutrons for calculating the effective delayed neutron fraction by the k-ratio method and the Bell and Glasstone spatial correction factor. The data added to the WIMSD library have been generated by the NJOY computer program using the ENDF/B nuclear data files. The macroscopic cross sections generated by DRAGON computer program describe more than sixty different zones of the subcritical assembly; in these zones, whenever necessary, the materials have been homogenized. The macroscopic cross sections generated by the DRAGON computer program have been then used in the PARTISN computer program, which modeled the YALINA thermal facility in details. In order to facilitate the coupling between the DRAGON and PARTISN computer programs, a new interface computer program (DRAGON2PARTISN) has been developed. For the time-dependent calculations, the PARTISN computer program has been modified to speed up the computing time. Some of the deterministic results have also been confirmed by the TORT computer program.

In addition, the YALINA thermal facility was modeled without any geometrical homogenization by two independent Monte Carlo computer programs, MONK and MCNP/MCNPX. The MONK computer program has nuclear data libraries, which are different from the libraries of MCNP/MCNPX computer programs.

The Deterministic and Monte Carlo analyses of the YALINA thermal assembly investigated: the neutron multiplication factor (k_{eff}), the neutron source multiplication factor, the effective delayed neutron fraction, the prompt neutron lifetime and generation time, the spatial neutron flux profile, the neutron spectrum, and the ^{3}He(n,p), ^{235}U(n,f), ^{197}Au(n,γ), ^{115}In(n,γ), and ^{55}Mn(n,γ) reaction rates profiles, the time dependent ^{3}He reaction rate from the pulsed deuterium-deuterium or the deuterium-tritium neutron source, and the Bell and Glasstone correction factors for the measured reactivity values. The analytical results have been compared with experimental measurements.

This report presents the computational models and the obtained results. The comparison of the deterministic and Monte Carlo results shows an excellent agreement with the experimental measurements.

DETERMINISTIC AND MONTE CARLO MODELING AND ANALYSES OF YALINA-THERMAL SUBCRITICAL ASSEMBLY

1. Introduction

Subcritical nuclear assemblies driven by an external neutron source for transmuting nuclear waste have been proposed.[1-7] These transmuters are referred to as Accelerators Driven Systems (ADS). Research programs for partitioning and transmuting nuclear waste by ADS are under development in Japan,[8-11] China,[12] Europe,[13-31] and Russia.[32-34] In Europe, the MUSE program was carried out at the MASURCA facility in Cadarache, France.[13-19] The MASURCA facility consists of a fast subcritical reactor, loaded with MOX fuel and sodium, coupled with a deuteron accelerator, and it operated at low power of less than 5 kW. Another set of experiments has been carried out within the TRIGA Accelerator Driven Experiment (TRADE) program.[20-25] Another related program is the RACE-T.[26] The current European research and development activity aims at the construction of a multipurpose prototype ADS for nuclear waste transmutation. Two major projects are under consideration: the PDS-ADS[27-28] and the MYRRHA/XT-ADS.[28-31] Both projects use eutectic lead-bismuth targets and a windowless target is considered for the MYRRHA/XT-ADS project. In Russia, the ADS experiments have been considered at the Subcritical Assembly in Dubna (SAD), which couples Russian fuel elements of the BN-600 reactor[35] with a proton accelerator. In the United States, the ADS experiments have been carried out under the Reactor-Accelerator Coupling Experiments Project (RACE). In this project, water-moderated uranium-aluminum fuel plates have been coupled with an electron accelerator at Idaho State University in Pocatello, ID, USA.[36-39]

In 2005, the International Atomic Energy Agency (IAEA) has started coordinated ADS research activities to validate computational methods, to characterize the performance of different subcritical systems, to develop and improve reactivity monitoring techniques, and the utilization of low enriched uranium instead of high enriched uranium in these systems. These activities include the YALINA booster and thermal subcritical experiments.[40-43] The YALINA subcritical assemblies are located at the Joint Institute for Power & Nuclear Research – SOSNY, Minsk, Belarus and the assemblies can be driven by californium (Cf), deuterium-deuterium (D-D), or deuterium-tritium (D-T) neutron source. The Argonne National Laboratory previous analyses focused on the YALINA booster assembly,[44-46] which has different zones with a fast and a thermal neutron spectra. This work examines the YALINA thermal assembly, which has a thermal neutron spectrum, and it extends the computational methodologies used in the previous studies. The YALINA thermal assembly has been modeled by the deterministic[47] computer programs DRAGON,[48] PARTISN,[49-53], and TORT.[54] The DRAGON computer program has been developed at the Polytechnique de Montréal, Canada, and it is used for licensing CANDU nuclear power plants in CANADA. In this work, the DRAGON computer program has been used to generate the macroscopic cross sections of sixty different material zones of the YALINA thermal assembly. The processing of the macroscopic cross sections started with the 69 energy groups WIMSD microscopic cross section library.[55] The macroscopic cross sections generated by DRAGON have been then

used in the PARTISN computer program, which modeled the facility in details using the 69 energy groups structure and about 60 different material zones. The TORT modeling of YALINA thermal is the same as the PARTISN one and the modeling details are discussed in Section 2.B.

The results of deterministic computer programs (PARTISN and TORT) have been compared with those from the Monte Carlo[56] computer programs MCNP/MCNPX[57-58] and MONK[59-61], in addition to the experimental measurements. The Monte Carlo modeling of the facility followed the approach used in the previous studies[44-46] for the YALINA booster assembly. The performed analyses include the neutron multiplication factor (k_{eff}), the Bell and Glasstone correction factors for the measured reactivities, the neutron source multiplication factor (k_{src}), the effective delayed neutron fraction, the prompt neutron lifetime and the generation time, the spatial neutron flux profiles, the neutron spectrum, and the ^{3}He(n,p), ^{235}U(n,f), ^{197}Au(n,γ), ^{115}In(n,γ), and ^{55}Mn(n,γ) reaction rate profiles. The comparison between the experimental measurements and the numerical results included the neutron multiplication factor of the assembly (k_{eff}), the ^{3}He reaction rate as a function of time from the pulsed deuterium-deuterium neutron source, and the ^{3}He reaction rate spatial profile from the Cf neutron source. The comparison between the deterministic results, the Monte Carlo results, and the experimental measurements showed an excellent agreement.

2. The YALINA thermal Assembly

2.A. Monte Carlo Modeling

The YALINA thermal assembly has polyethylene moderator and EK10 fuel rods with 10% enriched uranium. The EK10 fuel, the polyethylene moderator, the assembly unit cell, and the target zone of the YALINA thermal assembly are the same as those of the YALINA booster assembly.[41-46] The assembly fuel unit cell, the target, and fuel zones have a square geometry with a side of 2, 8 and 40 cm, respectively. The axial reflector is a 9-cm thick borated polyethylene at one side of the fuel rods. On the other side of the fuel rods, the deuteron beam tube penetrates the assembly without axial reflector. Air surrounds the deuteron beam tube and an inactive part of the fuel rods. The radial reflector is 40-cm graphite reflector. An organic glass layer covers the vertical outer surfaces of the graphite reflector. The other two external sides (top and bottom) of the radial graphite reflector are covered with cadmium and iron layers, the iron layer is on the outside. In addition, a thin (0.4 cm thick) organic glass layer covers the end of the fuel zone on the air side. The active fuel length and the total fuel rod length are 50 and 59 cm, respectively.

The assembly has an experimental channel in the target zone (EC4), three experimental channels in the fuel zone (EC1, EC2, EC3), and three experimental channels in the reflector zone (EC5, EC6, EC7). All the experimental channels are parallel to the fuel rods except EC7, which is perpendicular to the fuel rods. EC1, EC2, EC3, EC5, EC6, and EC7 experimental channels have 1.2 cm radius. In addition to the experimental channels, the facility has also four measurement channels located at the corners of the fuel zone. Three

control rods for emergency shutdown are located in the fuel zone next to the target zone. The control rods are always extracted during normal operation. A further detailed description of the assembly specifications is reported in the IAEA benchmark specifications.[40,42]

In this study, three different fuel loading patterns have been analyzed. The three configurations have different number of EK10 fuel rods, which are 216, 245, and 280. No experimental measurements have ever been performed with the 216 configuration. A picture of the YALINA thermal assembly is given in Figure 1. Figures 2 to 6 illustrate three-dimensional views of the assembly model with MONK computer program. In these figures the colors correspond to the different materials as follows:

- pale green – borated polyethylene
- blue – organic glass
- cyan – graphite
- gray – iron
- yellow – air
- pink – lead
- purple – polyethylene
- green – stainless steel
- orange – copper
- light blue – water
- white – polystyrene
- light yellow – vacuum
- pale blue – fuel

Figures 7 to 13 show cross section details of the MONK assembly model. The experimental channels of the facility are labeled in Figures 6, 7, and 8. The three different fuel loading configurations, 216, 245, and 280, are illustrated in Figures 11, 12, and 13, respectively. In the MONK assembly model, the material approximations discussed in appendix A of the previous Argonne report[44] are used. Figures 14 to 18 show cross sections of the MCNP/MCNPX assembly model. Both MONK and MCNP/MCNPX computer programs modeled the facility in details as specified in the IAEA benchmark report without any material homogenization. MONK and MCNP/MCNPX models use the same geometry and material specifications. The MCNP/MCNPX and MONK codes have been used (unless otherwise specified) with nuclear data libraries based ENDF/B-VI nuclear data files.[62-63] In all Figures, the statistical error has been indicated (whenever reported) with two standard deviation confidence (95% confidence). In all Tables, the statistical error has been indicated (whenever reported) with one standard deviation confidence (68% confidence).

2.B Deterministic Modeling

A deterministic model of the YALINA thermal assembly has been developed for the PARTISN computer code using a macroscopic cross section library with a 69 energy groups. The model has about sixty different material zones, as shown in Figures 19 through 33. In the PARTISN model, three approximations have been introduced:

1. The cadmium and the iron layers at the top and the bottom sides of the graphite reflector boundaries are not included in the geometrical model. This approximation has been introduced because the vacuum boundary condition at the iron boundary is equivalent to an infinite absorber. So removing the thin cadmium and iron layers do not impact the modeling results.
2. The organic glass layer at the four outer surfaces of the graphite reflector boundaries has been approximated by a graphite layer to simplify the geometrical model. The organic glass material has carbon and it is thickness is very small. In addition, it is located at the outer boundary of the geometrical model. Therefore, this approximation does not impact the accuracy of the deterministic model.
3. The three experimental channels in the reflector zone are filled with graphite to simplify the spatial mesh of the model and to reduce the numerical errors from using fine and coarse meshes next to each other.

These three approximations have been investigated by comparing the multiplication factor calculated with and without these three approximations using the MCNPX computer program. The obtained difference in the multiplication factor is within the statistical error of the Monte Carlo calculations.

The PARTISN simulations have been usually performed with the S_8 built-in angular quadrature set[64-65] and 60, 60, and 41 spatial mesh intervals along the x, y, and z axes, respectively. The fuels axes are parallel to the z axis. In source driven configurations with Cf, D-D, or D-T neutron sources, the angular quadrature set is S_{16} and the first collision technique is utilized.[66]

The PARTISN computer program has been modified for performing the time-dependent calculations. In this modification, the writing of the restart file at each time step has been suppressed in order to speed the calculation and save disk space. PARTISN writes a restart file at each time step, which is extremely expensive for large problems with large number of time steps.

The macroscopic cross sections for the PARTISN material zones have been generated by the DRAGON computer program using the WIMSD ENDF/B-VI.8 nuclear data library with 69 energy groups. The WIMSD nuclear data library has been modified to include hydrogen data in polyethylene; free carbon data in polyethylene, stainless steel, polystyrene and organic glass materials; ^{75}As, ^{130}Ba, ^{132}Ba, ^{134}Ba, ^{135}Ba, ^{136}Ba, ^{137}Ba, ^{138}Ba, ^{209}Bi, natural zinc data to describe the material impurities; uranium isotopes without delayed neutrons for calculating the effective delayed neutron fraction by the k-ratio method. The update of the WIMSD library has been performed by the NJOY[67-68] and WILLIE[69] computer programs. The NJOY code used the ENDF/B-VI.8 nuclear data files[63] for hydrogen, carbon, ^{235}U, and ^{238}U isotopes and ENDF/B-VII.0 nuclear data files[63] for the other isotopes.

In the calculation of the effective delayed neutron fraction of Section 5 by the k-ratio method (sometimes also referred to as the two-runs method), two separate WIMSD libraries have been created by the NJOY and WILLIE computer programs: one library contains the

total fission neutrons with their spectrum and another includes only the prompt fission neutrons with their spectrum. The latter WIMSD library has been obtained by replacing the total fission neutrons by the prompt fission neutrons and by removing the delayed neutrons from the ENDF/B nuclear data file. These ENDF/B delayed neutron data are described in FILE 1 and MT 452 (total fission neutrons), MT 455 (delayed fission neutrons), and MT 456 (prompt fission neutrons).[62] In addition, the WIMSD library has been edited to replace the default fission spectrum with the one obtained by NJOY calculations.

Three types of DRAGON models have been used to generate the macroscopic cross sections. The first model has a single EK10 fuel rod inside a 2 x 2 cm polyethylene square box, a two-dimensional model. This model generated the homogenized cross section for the fuel and moderator unit cell. The second model has a one eight of the assembly including the target, fuel, and reflector zones using a two-dimensional geometry. This model has been modified several times to generate the homogenized cross sections for the fuel rods next a fuel hole, the fuel hole, the experimental and the measurement channels, the fuel zone with the empty channels for the control rods, and the graphite reflector regions. The third model has a one quarter of the target zone surrounded by two concentric square rows of fuel pins using a three-dimensional geometry. This model has the total axial length of the facility and it was used for generating the homogenized cross sections of the target, the beam tube, the lead, the borated polyethylene, and the inactive fuel regions. Appendix A describes in details the models used in the DRAGON calculations. All the DRAGON simulations used reflected or symmetric boundary conditions with zero buckling. An angular quadrature sets with 12, 32 and 16 angles has been used for the first, second, and third DRAGON models, respectively. The method of characteristics algorithm has been used for the fuel and moderator unit cell calculation; the probability collision algorithm has been used for the other two models. The scattering matrix expansion in orthogonal Legendre polynomials has been truncated to the first moment.

In order to facilitate the coupling between DRAGON and PARTISN codes, a new software interface DRAGON2PARTISN[70] has been written in C language.[71] The DRAGON2PARTISN software reads the ASCII macroscopic cross sections for a specific material from the DRAGON output and it rearranges them into a new file according to the material section syntax of the PARTISN input. The DRAGON2PARTISN software can utilize multiple DRAGON outputs files and for each of the output file the user can choose which material to process. According to the DRAGON2PARTISN material processing sequence, the macroscopic cross sections are sequentially stored (appended) in the material section of the PARTISN input. The DRAGON2PARTISN software also processes the scattering matrix up to the first moment. The source code of DRAGON2PARTISN is listed in Appendix C.

The TORT code used the same cross sections, geometry, and external neutron source (for the source driven cases) details as the PARTISN code. In this case, the DRAGON2PARTISN source code has been modified so that the P1 terms of the scattering matrix are multiplied by 3 ($2\ell+1$ factor). The source code of DRAGON2TORT is reported in Appendix D. TORT cross sections have been prepared by the GIP code,[72] which loaded the output of DRAGON2TORT code. The S_4 and S_8 TORT quadrature sets[73] have been used for criticality and external neutron source calculations, respectively.

2.C External Neutron Source Modeling

2.C.1 Californium Neutron Source

The YALINA thermal assembly can be driven by Cf, D-D, or D-T neutron source. In the first case, when experiments are performed, the beam tube is removed from the facility. Consequently, some of the results for the Cf neutron source have been calculated with the beam tube inserted (Table II, Figures 45-56, 75-81, 118, 120, 122, 125, 127, 129, 132, 134, 136, 138, and 140-143; these numerical results are compared with the numerical results from the D-D and the D-T neutron source), and the other results obtained without the beam tube (Table III, Figures 51-56, 81, 144-155; these numerical results are compared with the experimental results).

In MCNPX and MONK inputs, the californium neutron source has isotropic angular distribution and it located at the origin of the assembly. The fission neutron spectrum is used to describe the neutron energy of the Cf neutrons. The Appendix G data of the MCNP5 manual[57] have been used to describe the spontaneous Watt fission neutron spectrum of californium in the MCNP/MCNPX calculations.

In the PARTISN inputs, neutrons are emitted in a parallelepiped located at the center of the facility. The parallelepiped dimensions are 2 cm in the x and y axes and 6.45 cm in the z axis. It extends from z = -5 to 1.45 cm; z = 0 cm marks the location of the copper disk surface facing the deuteron beam. In the Monte Carlo model, the source neutrons are located at the origin (x = 0, y= 0, and z = 0). In the PARTISN calculations, as in Monte Carlo ones, the source neutrons have an isotropic distribution and a fission spectrum. In order to compare accurately the results from MCNP/MCNPX and PARTISN calculations, the MCNP/MCNPX calculations sometimes are performed with the californium neutron source definition of PARTISN calculations. In these MCNP/MCNPX calculations, a volume source instead of a point source is used as shown in Figures 138 and 144.

In MCNP/MCNPX results, the source multiplication factor (k_{src}) has been calculated from the weights summarized in the neutron creation table using the methodology of Reference 44. This table does not include the statistical error and therefore the standard deviation of the neutron source multiplication factor is not reported. A similar methodology has been used for the deterministic results by using the *source multiplication* parameter from the PARTISN output. This methodology has been also used for D-D and D-T k_{src} calculations.

2.C.2 Deuterium-Deuterium Neutron Source

For the D-D and D-T neutron sources, the beam tube is always modeled and the neutrons are emitted at the copper disk surface facing the deuteron beam. The D-D neutrons are generated by 250 KeV deuterons. Figures 34 and 35 show the angular distributions of the D-D and the D-T neutrons, respectively, which are obtained from the microscopic differential cross sections. The distributions show a peak in the forward direction. The angular distribution of the D-D source neutrons is more anisotropic than the angular distribution of the D-T neutrons. Figures 36 and 37 plot the microscopic differential cross section of the D-D reaction and the neutron energy angular distribution, respectively. The microscopic differential cross section exhibits a dip when neutrons are emitted perpendicular to the deuteron beam direction. The neutron energy angular distribution has a single dip

when neutrons are emitted in the backward direction, relative to the deuteron beam direction. The data in Figures 34 through 37 have been obtained by using the DROSG software.[74]

In the Monte Carlo calculations, D-D source neutrons have an isotropic distribution and 2.45 MeV energy. The neutrons are emitted from the copper disk surface facing the deuteron beam. In the deterministic calculations, D-D source neutrons have also isotropic distribution but their energy spreads in the energy range of 2.23 to 3.68 MeV, which matches the energy group width. In addition, D-D source neutrons are uniformly emitted from a small parallelepiped, as Cf source neutrons.

In order to compare accurately the results from MCNP/MCNPX and PARTISN calculations, the MCNP/MCNPX calculations sometimes used the D-D neutron source definition as PARTISN calculations. In these MCNP/MCNPX calculations, a volume source instead of a surface source and an energy bin instead of a mono-energetic source distribution are used for obtaining the results shown in Figures 69 through 74 and 139.

2.C.3 Deuterium-Tritium Neutron Source

The angular distribution of D-T source neutrons, the microscopic differential cross section of the D-T reaction, and the energy angular distribution of D-T neutrons are plotted in Figures 35, 36, and 37, respectively. The D-T neutrons have almost an isotropic distribution, which is different from the D-D neutrons because of the higher mass of the target nuclei. The D-T microscopic differential cross section and the energy angular distribution have a single dip for the neutrons emitted in the backward direction relative to the deuteron beam direction. The D-T microscopic differential cross section is more than one order of magnitude higher than the D-D one.

In the Monte Carlo calculations, D-T source neutrons have an isotropic distribution and 14.1 MeV energy. They are emitted from the copper disk surface facing the deuteron beam. The PARTISN and TORT codes have not been considered for the D-T neutron source calculations because the upper energy boundary of the WIMSD 69 energy group library is 10 MeV, which is less than the average energy of D-T source neutrons.

3. Fuel and Moderator Unit Cell

The first comparison between Monte Carlo and deterministic results has been performed for the fuel unit cell. The two dimensional unit cell model contains only one EK10 fuel rod inside a 2 x 2 cm polyethylene square box with reflecting boundary conditions. The MCNP/MCNPX multiplication factor of the unit cell is 1.36520±5. This value is only 48 pcm higher than the DRAGON multiplication factor value. Karthikeyan and Hebert (one of the authors of the DRAGON computer program) have observed a 280 pcm difference between MCNP5 and DRAGON multiplication factors of a CANDU fuel unit cell.[75]

The effective delayed neutron fraction calculated by MCNP/MCNPX and DRAGON with the k-ratio method[76-77] is 690±10 and 688 pcm, respectively. In the DRAGON calculation, two different WIMSD nuclear data libraries have been used, as described in Section 2.B. The effective delayed neutron fraction is close to the ^{235}U average delayed neutron fraction, as

visible in Figure 38. U^{238} contributes little to the fission rate, as discussed in section 5.A. The sharp decrease of the ^{235}U (and ^{238}U) delayed neutron fraction above 1 MeV is due to the corresponding sharp increase of prompt fission neutrons as shown in Figures 38 and 39. The effective prompt neutron lifetime, calculated by MCNP/MCNPX and PARTISN, of the fuel unit cell is equal to 59 μs.

Figures 40 to 42 show the total, the absorption, and the fission macroscopic cross sections calculated by DRAGON and MCNP/MCNPX; the curves of the deterministic results superimpose to those one of Monte Carlo results. Figures 43 and 44 plot the number of secondary neutrons (ν) multiplied by the fission cross section and the neutron spectrum, respectively, calculated by DRAGON and MCNP/MCNPX; also in this case the agreement between Monte Carlo and deterministic results is excellent.

4. Neutron Multiplication Factor

4.A Criticality Analyses

The multiplication factor results for the YALINA thermal assembly calculated by Monte Carlo and deterministic codes are summarized in Table I. The neutron multiplication factor increases as the number of fuel rods increases from 216, 245, up to 280. Generally, a very good agreement has been obtained between the results obtained by different computer codes (MCNP/MCNPX, MONK, PARTISN, and TORT) and different nuclear data files (ENDF/B-VI, ENDF/B-VII, and JEFF-3.1).[62-63,78-79] The maximum difference is less than 100 pcm. Consequently, the ENDF/B-VI nuclear data library (Mod 6 for MCNP/MCNPX; Mod 2 for DICE semi-continuous energy format for MONK; and Mod 8 for PARTISN and TORT) has been used in all the results obtained in this study. The nuclear data library based on ENDF/B-VI.2 produces a 300 pcm smaller neutron multiplication factor since the carbon is modeled as free carbon (C_{free}) and the $S(\alpha,\beta)$ scattering function has been ignored. This library has been used only for MONK calculations. The PARTISN calculation has been performed both in direct and adjoint modes; the difference between the two modes is less than 50 pcm. Filling the experimental channels EC5, EC6, and EC7 with graphite does not change the MCNP/MCNPX multiplication factor since the amount of the additional graphite is very small.

The PARTISN code uses the S_N method[47] to solve the neutron transport equation. This method is based on selecting a discrete set of directions and their associated weights. The PARTISN results have been obtained with the S_8 built-in quadrature set[50, 64-65] which contains 10 directions per solid angle octant.[80] The use of S_4 equal weight quadrature set increases the neutron multiplication factor by 76 pcm for the 216 fuel rods configuration. The neutron multiplication factor obtained with the S_8 triangular Chebychev-Legendre[50, 64-65] quadrature set is only 9 pcm higher than the value obtained with the S_8 built-in quadrature set.

Table I also reports the multiplication factor obtained by the area method[81-82] for the D-D pulsed neutron source experiments as described in Section 11.B. The area method can be applied with Gozani[83] or Garelis and Russel[84-85] corrections. The Gozani or Garelis and

Russel corrections have never been used in this study. The discrepancy between MCNP/MCNPX and the experimental results is less than 400 and 1100 pcm for the 245 and 280 configurations, respectively. The measured reactivity needs a spatial correction factor for accurate comparison with the calculated values.[18,45-46,86]

4.B Californium Neutron Source

The source multiplication factor has been calculated using the two different definitions given in Equations 1 and 2. Equation 1 assumes that the (n,xn) reactions contribute to the absorption term of the neutron transport equation with negative sign.[87] Equation 2 assumes that the (n,xn) reactions contribute to the fission term of the neutron transport equation with positive sign.

$$k_{src}^{f} \equiv \frac{\int_{4\pi} d\vec{\Omega} \int_{V} d\vec{r} \int_{0}^{20MeV} dE \nu \Sigma_{f}(\vec{r},E)\varphi(\vec{r},E,\vec{\Omega})}{\int_{4\pi} d\vec{\Omega} \int_{V} d\vec{r} \int_{0}^{20MeV} dE \nu \Sigma_{f}(\vec{r},E)\varphi(\vec{r},E,\vec{\Omega}) + \int_{4\pi} d\vec{\Omega} \int_{V} d\vec{r} \int_{0}^{20MeV} dE s(\vec{r},E,\vec{\Omega})} \quad (1)$$

$$k_{src}^{fx} \equiv \frac{\int_{4\pi} d\vec{\Omega} \int_{V} d\vec{r} \int_{0}^{20MeV} dE \left[\nu \Sigma_{f}(\vec{r},E) + 2\Sigma_{n,2n}(\vec{r},E) + 3\Sigma_{n,3n}(\vec{r},E)\right]\varphi(\vec{r},E,\vec{\Omega})}{\int_{4\pi} d\vec{\Omega} \int_{V} d\vec{r} \int_{0}^{20MeV} dE \left[\nu \Sigma_{f}(\vec{r},E) + 2\Sigma_{n,2n}(\vec{r},E) + 3\Sigma_{n,3n}(\vec{r},E)\right]\varphi(\vec{r},E,\vec{\Omega}) + \int_{4\pi} d\vec{\Omega} \int_{V} d\vec{r} \int_{0}^{20MeV} dE s(\vec{r},E,\vec{\Omega})} \quad (2)$$

In Equations 1 and 2, $\vec{r}$ is the neutron position, $\vec{\Omega}$ is the neutron flight direction, E is the neutron energy, φ is the angular neutron flux, ν is the total number of fission neutrons, Σ_f is the macroscopic fission cross section, $\Sigma_{n,2n}$ is the (n,2n) macroscopic cross section, $\Sigma_{n,3n}$ is the (n,3n) macroscopic cross section, and s is the external neutron source. The two definitions of the source multiplication factor lead to different neutron multiplication values. The neutron multiplication obtained by the source multiplication factor defined in Equation 1 includes only neutrons created by fission reactions (Equation 3). The neutron multiplication obtained by the source multiplication factor defined in Equation 2 includes not only neutrons created by fission reactions but also neutrons created by (n,xn) reactions (Equation 4).

$$M^{f} \equiv \frac{1}{1-k_{src}^{f}} = \frac{\int_{4\pi} d\vec{\Omega} \int_{V} d\vec{r} \int_{0}^{20MeV} dE \nu \Sigma_{f}(\vec{r},E)\varphi(\vec{r},E,\vec{\Omega}) + \int_{4\pi} d\vec{\Omega} \int_{V} d\vec{r} \int_{0}^{20MeV} dE s(\vec{r},E,\vec{\Omega})}{\int_{4\pi} d\vec{\Omega} \int_{V} d\vec{r} \int_{0}^{20MeV} dE s(\vec{r},E,\vec{\Omega})} \quad (3)$$

$$M^{fx} \equiv \frac{1}{1-k_{src}^{fx}} = \frac{\int_{4\pi} d\vec{\Omega} \int_{V} d\vec{r} \int_{0}^{20MeV} dE \left[\nu \Sigma_{f}(\vec{r},E) + 2\Sigma_{n,2n}(\vec{r},E) + 3\Sigma_{n,3n}(\vec{r},E)\right]\varphi(\vec{r},E,\vec{\Omega}) + \int_{4\pi} d\vec{\Omega} \int_{V} d\vec{r} \int_{0}^{20MeV} dE s(\vec{r},E,\vec{\Omega})}{\int_{4\pi} d\vec{\Omega} \int_{V} d\vec{r} \int_{0}^{20MeV} dE s(\vec{r},E,\vec{\Omega})} \quad (4)$$

The neutron multiplication factor reported in Table II shows that there is a negligible difference between the k_{src} results obtained by Equations 1 and 2 for the californium neutron

source. The (n,2n) reactions have a threshold energy above 5.5 MeV for ^{235}U and ^{238}U, therefore an insignificant contribution is expected from (n,2n) reactions. The californium k_{src} is higher than the fission multiplication factor because the source neutrons are emitted in a high importance zone of the assembly, the assembly center. For the 280 configuration, the californium k_{src} is closer to k_{eff} because source neutrons contribute less to the fission chain; the 280 configuration has the highest neutron multiplication factor. The cross section remarks of the criticality calculations in Section 4.A hold also for the californium source multiplication factor.

Table III reports the californium source multiplication factor calculated for the assembly when the beam tube is removed and air fills the target region. The beam tube behaves as a neutron absorber and its removal increases k_{src} by about 700 pcm. The removal of the beam tube has been analyzed because the experimental measurements with the californium neutron source were performed without the beam tube.

4.C Deuterium-Deuterium Neutron Source

The results for the D-D neutron source are summarized in Table IV. These results are very similar to the ones obtained for the californium neutron source with the beam tube inserted. The D-D source multiplication factor is smaller than the californium source multiplication factor because D-D neutrons have higher average energy (~2.45 MeV) relative to the californium neutrons (~2 MeV), which increases the neutron leakage. The remarks observed in sections 4.A and 4.B hold also for the D-D source multiplication factor.

When the energy-angular distribution shown in Figure 37 is explicitly modeled in the analyses for the 216 configuration, the k_{src} decreases by 60 pcm because of the increased axial leakage. The angular distribution of the D-D neutron source has more anisotropic distribution than the D-T neutron source and it has a minimum when neutrons are emitted perpendicular to the deuteron beam and the fuel rods. The energy-angular dependency distribution of the source neutrons has been modeled by using 180 angles in the $[0,\pi]$ interval, which starts from the deuteron beam direction.

4.D Deuterium-Tritium Neutron Source

The k_{src} set by the D-T neutron source, shown in Table V, is lower than the one set by the D-D neutron source shown in Table IV. The higher average energy of the D-T source neutrons (~14.1 MeV), relative to D-D source neutrons (~2.45 MeV) increases the neutrons leakage. For the YALINA booster subcritical assembly, the k_{src} with D-T neutron source is higher than the k_{src} with the D-D neutron source because the YALINA booster assembly volume is much larger than the YALINA thermal assembly volume.[40-42]

The results summarized in Table V show that the D-T source multiplication factor calculated by Equation 2 is 40-400 pcm higher than the one calculated by Equation 1. This is due to the difference in the inclusion of the (n,2n) and (n,3n) contributions in the transport equation and the corresponding definition of k_{src}. The D-T source neutrons have an average

energy much higher than the threshold energy of the (n,2n) reaction of ^{235}U (~5.5 MeV). In addition, the (n,3n) threshold energy for ^{235}U is 13 MeV, therefore this reaction is also possible. The difference between the results obtained by Equations 1 and 2 increases with the decrease of the fuel rods number because the contribution to the neutron population of D-T source neutrons, relative to fission neutrons, increases.

Close to criticality, the k_{src} converges to k_{eff} since source neutrons are outnumbered by fission neutrons. This behavior has been numerically tested by filling all fuel holes of the assembly with fuel rods. When the number of fuel rods is 317 and the ^{235}U mass fraction in the fuel material is increased up to 8.17% by increasing the ^{235}U enrichment from 9.86% to 10.08%, the k_{eff} is 0.99952±6, which is only 10 pcm less than the k_{src}. These values have been calculated by MCNPX-2.6b using ENDF/B-VI.6 data.

For the 216 configuration of YALINA-Thermal, the energy-angular distribution of the source increases the k_{src} by about 40 to 130 pcm, because neutrons are more likely to be emitted in the forward direction (relative to the deuteron beam), which is on the side of the axial reflector, the opposite side in the backward direction does not have any reflector. On the contrary of D-D source neutrons, D-T source neutrons do not have an angular distribution with a minimum for neutrons emitted perpendicular to the beam tube towards the center of the active fuel length. This explains why the modeling of the angular distribution decreases k_{src} for the D-D neutron source and increases k_{src} for the D-T neutron source.

5. Delayed Neutron Fraction

5.A Adjoint-Weighted (Effective) Delayed Neutron Fraction

The adjoint-weighted delayed neutron fraction (β_{eff}) (also referred to as the effective delayed neutron fraction) is defined as the ratio of the production rate of delayed neutrons weighted with the adjoint flux, and the production rate of all fission neutrons weighted with the adjoint flux, as described in Equation 5:[88-90]

$$\beta_{eff} \equiv \frac{\int_{4\pi} d\vec{\Omega} \int_{4\pi} d\vec{\Omega}' \int_0^{20MeV} dE \int_0^{20MeV} dE' \int_V d\vec{r} \varphi^+(\vec{r},E',\vec{\Omega}') \sum_{k=1}^{6} \chi_{d,k}(\vec{r},E') \cdot \nu_{d,k}(\vec{r},E) \cdot \Sigma_f(\vec{r},E) \cdot \varphi(\vec{r},E,\vec{\Omega})}{\int_{4\pi} d\vec{\Omega} \int_{4\pi} d\vec{\Omega}' \int_0^{20MeV} dE \int_0^{20MeV} dE' \int_V d\vec{r} \varphi^+(\vec{r},E',\vec{\Omega}') \cdot \chi(\vec{r},E') \cdot \nu(\vec{r},E) \cdot \Sigma_f(\vec{r},E) \cdot \varphi(\vec{r},E,\vec{\Omega})} \quad (5)$$

In Equation 5, $\vec{r}$ is the neutron position, $\vec{\Omega}$ is the neutron flight direction, E is the neutron energy, φ^+ is the adjoint angular neutron flux, φ is the forward angular neutron flux, ν is the total number of fission neutrons, $\nu_{d,k}$ is the number of delayed fission neutrons for the k delayed neutron precursor family, χ is the energy spectrum of all neutrons (both prompt and delayed neutrons), $\chi_{d,k}$ is the energy spectrum of delayed neutrons from the k delayed neutron precursor family, and Σ_f is the macroscopic fission cross section. In a critical system, the adjoint neutron flux at a point in the phase space (space, energy, and angle) is proportional to the expected steady state neutron population resulting from a hypothetical neutron introduced

at that point in phase space.[91-92] In Equation 5, both the forward and the adjoint neutron fluxes solve the homogeneous neutron transport equation, which holds for a critical system with no external neutron source. In the ENDF/B nuclear data files (both version 6 and 7), the delayed neutron precursors are grouped into 6 families.[63,93-94] Only the family with the longest decay half-life (about 55 seconds) contains only one isotope (^{87}Br); this affects the kinetics behavior of the reactor core.[94] For subcritical systems driven by an external neutron source, Equation 5 changes into Equation 6, which uses the forward and adjoint neutron fluxes solving the inhomogeneous neutron transport Equations 7 and 8, respectively.[88,95]

$$\beta_{eff}^{s} \equiv \frac{\int_{4\pi} d\vec{\Omega} \int_{4\pi} d\vec{\Omega}' \int_{0}^{20MeV} dE \int_{0}^{20MeV} dE' \int_{V} d\vec{r} \varphi_s^{+}(\vec{r},E',\vec{\Omega}') \sum_{k=1}^{6} \chi_{d,k}(\vec{r},E') \cdot \nu_{d,k}(\vec{r},E) \cdot \Sigma_f(\vec{r},E) \cdot \varphi_s(\vec{r},E,\vec{\Omega})}{\int_{4\pi} d\vec{\Omega} \int_{4\pi} d\vec{\Omega}' \int_{0}^{20MeV} dE \int_{0}^{20MeV} dE' \int_{V} d\vec{r} \varphi_s^{+}(\vec{r},E',\vec{\Omega}') \cdot \chi(\vec{r},E') \cdot \nu(\vec{r},E) \cdot \Sigma_f(\vec{r},E) \cdot \varphi_s(\vec{r},E,\vec{\Omega})} \quad (6)$$

$$\vec{\Omega} \cdot \vec{\nabla} \varphi_s(\vec{r},\vec{\Omega},E) + \Sigma_t(\vec{r},E)\varphi_s(\vec{r},\vec{\Omega},E) = \int_{0}^{+\infty} dE' \int_{4\pi} d\vec{\Omega}' \Sigma_s(\vec{r},\vec{\Omega}\cdot\vec{\Omega}',E' \to E)\varphi_s(\vec{r},\vec{\Omega}',E') +$$

$$+ \frac{\chi(\vec{r},E)}{4\pi} \int_{0}^{+\infty} dE' \int_{4\pi} d\vec{\Omega}' \nu(\vec{r},E')\Sigma_f(\vec{r},E')\varphi_s(\vec{r},\vec{\Omega}',E') + s(\vec{r},\vec{\Omega},E) \quad (7a)$$

$$\varphi(\vec{r}_b,\vec{\Omega},E,t)\Big|_{\vec{n}\cdot\vec{\Omega}<0} = \begin{cases} 0 & \text{vacuum boundary condition} \\ \varphi(\vec{r}_b,\vec{\Omega}',E,t)\Big|_{\vec{n}\cdot\vec{\Omega}'>0} \quad \vec{\Omega}' = \vec{\Omega} - 2\left(\vec{\Omega}\cdot\vec{n}\right)\cdot\vec{n} & \text{reflected boundary condition} \end{cases} \quad (7b)$$

$$-\vec{\Omega} \cdot \vec{\nabla} \varphi_s^{+}(\vec{r},\vec{\Omega},E) + \Sigma_t(\vec{r},E)\varphi_s^{+}(\vec{r},\vec{\Omega},E) = \int_{0}^{+\infty} dE' \int_{4\pi} d\vec{\Omega}' \Sigma_s(\vec{r},\vec{\Omega}\cdot\vec{\Omega}',E \to E')\varphi_s^{+}(\vec{r},\vec{\Omega}',E') +$$

$$+ \frac{\nu(\vec{r},E)\Sigma_f(\vec{r},E)}{4\pi} \int_{0}^{+\infty} dE' \int_{4\pi} d\vec{\Omega}' \chi(\vec{r},E')\varphi_s^{+}(\vec{r},\vec{\Omega}',E') + \nu(\vec{r},E)\Sigma_f(\vec{r},E) \quad (8a)$$

$$\varphi_s^{+}(\vec{r}_b,\vec{\Omega},E,t)\Big|_{\vec{n}\cdot\vec{\Omega}\geq 0} = \begin{cases} 0 & \text{vacuum boundary condition} \\ \varphi_s^{+}(\vec{r}_b,\vec{\Omega}',E,t)\Big|_{\vec{n}\cdot\vec{\Omega}'\leq 0} \quad \vec{\Omega}' = \vec{\Omega} - 2\left(\vec{\Omega}\cdot\vec{n}\right)\cdot\vec{n} & \text{reflected boundary condition} \end{cases} \quad (8b)$$

If one sets to zero the last term in the right side of Equations 7a and 8a, representing the external neutron source, then the homogeneous forward and adjoint fluxes, used in Equation 5, are obtained. The choice of $\nu\Sigma_f$ as external neutron source in Equation 8a allows the interpretation of the adjoint function as the neutron importance.[96-97]

Assuming the prompt neutron flux is approximately equal to the total neutron flux, the k-ratio method[76-77] evaluates the effective delayed neutron fraction as the ratio between the delayed and the total neutron multiplication factors. The delayed neutron multiplication factor is obtained as the difference between the total k_{eff} and the prompt neutron multiplication factor k_{prompt}, as shown in Equation 9.

$$\beta_{eff} \cong \frac{k_{eff} - k_{prompt}}{k_{eff}} = 1 - \frac{k_{prompt}}{k_{eff}} \qquad (9)$$

MCNP/MCNPX computer codes can be used to calculate β_{eff} by the k-ratio method since these computer codes have the capability to ignore the delayed neutrons. The MONK computer program does not have this capability. The k-ratio method has been applied also by PARTISN, using the procedure described in section 2.B by producing a nuclear data library without the delayed neutrons. In the case of MCNP/MCNPX, the delayed neutron fraction obtained by k-ratio method has a standard deviation that can be calculated using Equation 10.[98]

$$\sigma_{1-x/y} = \sigma_{x/y} = \frac{E(x)}{E(y)} \sqrt{\frac{\sigma_x^2}{E^2(x)} + \frac{\sigma_y^2}{E^2(y)} - 2\frac{\mathrm{cov}(x,y)}{E(x)E(y)}} \qquad (10)$$

In Equation 10, x is the random variable k_{prompt} and y is the random variable k_{eff}; σ_x, σ_y , $\sigma_{1-x/y}$, and $\sigma_{x/y}$ are the standard deviation of x, y, $1-x/y$ and x/y, respectively; $E(x)$ and $E(y)$ are the expected values of the random variables x and y, respectively; $\mathrm{cov}(x,y)$ is the covariance of the random variables x and y. In the present study, the covariance term in Equation 10 has been neglected and therefore the standard deviation is overestimated. The covariance term is not zero because the random variables x (k_{prompt}) and y (k_{eff}) are correlated. The k-ratio method requires two separate simulations: one for the k_{eff} and the other for the k_{prompt}. The Meulekamp and van der Marck method[89] requires only one simulation because it tallies, in the same Monte Carlo simulation,[99] the number of fissions for each prompt and delayed neutron history. The ratio between the fissions induced by delayed neutrons and the fissions induced by all neutrons (both quantities weighted by the fission probability obtained by the previous mentioned tally) represents the effective delayed neutron fraction, as defined in Equation 5. While using an equal number of neutron histories, the Meulekamp and van der Marck method produces a much lower statistical error than the k-ratio method.

In order to estimate the effective delayed neutron fraction by deterministic codes, Equations 5 and 6 have been evaluated by a C language[71] program which combined data from the ENDF/B-nuclear data files with PARTISN results. The C program used the total number of fission neutrons as a function of energy (ν), the number of delayed fission neutrons as a function of energy (ν_d), the delayed neutron spectrum of the six delayed neutron precursor families ($\chi_{d,k}$), and the fractional probability (w_k) of the six delayed neutron precursor families from the ENDF/B-VII.0 data for ^{235}U. [62,78,100] The ENDF/B-delayed neutron data of ^{238}U have not been used because the facility operates on a thermal neutron spectrum. In addition to the ^{235}U ENDF/B-VII.0 data, the C program used the total neutron spectrum (χ), the number of fission neutrons multiplied by the fission reaction rate integrated over the volume of the spatial mesh ($\mathrm{V}\,\nu\Sigma_f\Phi$), and the scalar adjoint flux (Φ^+) from the PARTISN output files, which include the ^{238}U contribution. By combining all the above data, the C program calculated the delayed neutron fraction according to Equation 11. The source code of the C program is listed in Appendix E.

$$\beta_{eff} \equiv \frac{\int_{4\pi} d\vec{\Omega} \int_{4\pi} d\vec{\Omega}' \int_0^{20MeV} dE \int_0^{20MeV} dE' \int_V d\vec{r} \varphi^+(\vec{r},E,\vec{\Omega}) \sum_{k=1}^{6} w_k \cdot \chi_{d,k}(\vec{r},E) \cdot \frac{\nu_d(\vec{r},E')}{\nu(\vec{r},E')} \cdot \nu\Sigma_f(\vec{r},E') \cdot \varphi(\vec{r},E',\vec{\Omega}')}{\int_{4\pi} d\vec{\Omega} \int_{4\pi} d\vec{\Omega}' \int_0^{20MeV} dE \int_0^{20MeV} dE' \int_V d\vec{r} \varphi^+(\vec{r},E,\vec{\Omega}) \cdot \chi(\vec{r},E) \cdot \nu\Sigma_f(\vec{r},E') \cdot \varphi(\vec{r},E',\vec{\Omega}')}$$

$$\cong \frac{\int_0^{20MeV} dE \int_0^{20MeV} dE' \int_V d\vec{r} \Phi^+(\vec{r},E) \sum_{k=1}^{6} w_k \cdot \chi_{d,k}(\vec{r},E) \cdot \frac{\nu_d(\vec{r},E')}{\nu(\vec{r},E')} \cdot \nu\Sigma_f(\vec{r},E') \cdot \Phi(\vec{r},E')}{\int_0^{20MeV} dE \int_0^{20MeV} dE' \int_V d\vec{r} \Phi^+(\vec{r},E) \cdot \chi(\vec{r},E) \cdot \nu\Sigma_f(\vec{r},E') \cdot \Phi(\vec{r},E')} \cong$$

$$\cong \frac{\sum_{i=1}^{I} \sum_{g=1}^{G} \sum_{k=1}^{6} \Phi^+_{g,i} \cdot w_k \cdot \chi_{d,k,g,i} \cdot \left(\sum_{g'=1}^{G} \frac{\nu_{d,g',k,i}}{\nu_{g',i}} \cdot V_i \cdot \nu\Sigma_{f,g',i} \cdot \Phi_{g',i} \right)}{\sum_{i=1}^{I} \sum_{g=1}^{G} \Phi^+_{g,i} \cdot \chi_{g,i} \cdot \left(\sum_{g'=1}^{G} V_i \cdot \nu\Sigma_{f,g',i} \cdot \Phi_{g',i} \right)} \qquad (11)$$

Equation 11 uses the same symbols as Equation 5. In addition, w_k represents the relative fraction of the delayed neutrons from the k delayed neutron precursor family (the sum of w_k from 1 to 6 is one); the indexes i, g, and k refer to the volume cell of the spatial mesh, the energy group, and the delayed neutron precursor family, respectively. A PARTISN sample input for the calculation of the adjoint neutron flux can be found in the of Favorite's article.[101] MCNP/MCNPX simulations have shown that ^{238}U contributes only about 0.28, 0.3, and 0.32% to the total fission rate for configurations 216, 245 and 280, respectively. Assuming a constant delayed neutron fraction of 680 and 1800 pcm for ^{235}U and ^{238}U from Figure 38, respectively, the contribution of ^{238}U to the effective delayed neutron fraction is less than 4 pcm. In addition, the change of the effective delayed neutron fraction between the three fuel configurations due to ^{238}U different contributions to the total fission rate would be about 0.2 pcm. These remarks confirm the accuracy of the PARTISN calculation assumptions and methodology.

All the deterministic and the Monte Carlo computer programs estimated an effective delayed neutron fraction between 770 and 800 pcm, as given in Table VI. No significant difference has been found between using the adjoint and forward fluxes calculated by PARTISN in criticality mode and the adjoint and forward fluxes calculated by PARTISN in source mode. In the latter case, the neutron source of the adjoint calculation, consisting of fission neutrons multiplied by the fission cross section ($\nu\Sigma_f$), was located in all fissile material zones of the assembly.

In the k-ratio method, the results from the two Monte Carlo simulations generating the prompt and total multiplication factors must have a very low statistical error, about 3 pcm, to accurately estimate the effective delayed neutron fraction. This high precision is required because the difference between the effective delayed neutron fraction of two successive configurations is only ~10 pcm. The use of the k-ratio method requires a very long

computation time to produce results with high precision. The k-ratio results overestimate the effective delayed neutron fraction both for Monte Carlo (Meulenkamp and van der Marck method) and deterministic (adjoint weighted method) results by 10 to 15 pcm.

Appendix B proposes a new definition of the delayed neutron fraction in which the external source and (n,xn) neutrons are counted as prompt neutrons.

5.B Adjoint-Unweighted (Average) Delayed Neutron Fraction

If a constant adjoint flux equal to unity is used, equation 5 produces the average delayed neutron fraction (β_{ave}), which is the fraction of delayed neutrons averaged over the phase space, as described in Equation 12.

$$\beta_{ave} \equiv \frac{\int_{4\pi} d\vec{\Omega} \int_0^{20MeV} dE \int_0^{20MeV} dE' \int_V d\vec{r} \sum_{k=1}^{6} \chi_{d,k}(\vec{r},E') \cdot \nu_{d,k}(\vec{r},E) \cdot \Sigma_f(\vec{r},E) \cdot \varphi(\vec{r},E,\vec{\Omega})}{\int_{4\pi} d\vec{\Omega} \int_0^{20MeV} dE \int_0^{20MeV} dE' \int_V d\vec{r} \cdot \chi(\vec{r},E') \cdot \nu(\vec{r},E) \cdot \Sigma_f(\vec{r},E) \cdot \varphi(\vec{r},E,\vec{\Omega})}$$

$$= \frac{\int_0^{20MeV} dE \int_0^{20MeV} dE' \int_V d\vec{r} \sum_{k=1}^{6} \chi_{d,k}(\vec{r},E') \cdot \nu_{d,k}(\vec{r},E) \cdot \Sigma_f(\vec{r},E) \cdot \Phi(\vec{r},E)}{\int_0^{20MeV} dE \int_0^{20MeV} dE' \int_V d\vec{r} \chi(\vec{r},E') \cdot \nu(\vec{r},E) \cdot \Sigma_f(\vec{r},E) \cdot \Phi(\vec{r},E)}$$

$$= \frac{\int_0^{20MeV} dE \int_V d\vec{r} \sum_{k=1}^{6} \cdot \nu_{d,k}(\vec{r},E) \cdot \Sigma_f(\vec{r},E) \cdot \Phi(\vec{r},E)}{\int_0^{20MeV} dE \int_V d\vec{r} \nu(\vec{r},E) \cdot \Sigma_f(\vec{r},E) \cdot \Phi(\vec{r},E)} \qquad (12)$$

In Equation 12, $\nu_{d,k}$ is equal to $w_k \nu_d$ of Equation 11. Equation 12 has been derived taking into accounts Equations 13 and 14 and it is the same as Equation 2 in the article of Meulekamp and van der Marck (Ref. 89). Meulekamp and van der Marck referred to the average delayed neutron fraction as the “fundamental” delayed neutron fraction.

$$\int_0^{20MeV} dE \chi_{d,k}(\vec{r},E') = 1 \qquad \forall k \in [1,2...6] \qquad (13)$$

$$\int_0^{20MeV} dE \chi(\vec{r},E') = 1 \qquad (14)$$

If delayed neutrons would be emitted with the prompt fission spectrum, then the average delayed neutron fraction would be the same as the effective delayed neutron fraction.

Delayed fission neutrons born with a softer spectrum relative to prompt fission neutrons; consequently, the effective delayed neutron fraction is higher than the average delayed neutron fraction because thermal neutrons are more likely to induce a fission reaction. Figure 38 shows that a pure ^{235}U fuel cannot have an average delayed neutron fraction higher than 700 pcm. By contrast, the effective delayed neutron fraction of a very high enriched uranium fuel can be higher than 700 pcm, as in the case of the FRM-II reactor which has 93% enriched fuel.[102] The higher value of the effective delayed neutron fraction, relative to the average value can be also understood from Equation 5. The spectrum of the delayed fission neutrons is softer than the spectrum of the prompt fission neutrons, therefore the numerator of equation 5 weights more the thermal component of the adjount flux, which is much higher than the fast component and this increases the effective delayed neutron fraction relative to the average one. For the YALINA thermal assembly, the average delayed neutron fraction calculated by MCNP/MCNPX and PARTISN is between 670 and 700 pcm, as reported in Table VI. The calculation of the average delayed neutron fraction by the MCNP/MCNPX code has been performed by two independents methods: the first method is the k-ratio and the second method tallies the prompt and total fission rate, the delayed fission rate is obtained as the difference. Both methods set the delayed neutron spectrum equal to the prompt neutron spectrum and provide the same results. The calculation of the average delayed neutron fraction by PARTISN has also been performed by two independent methods: the first method is the k-ratio and the second method sets a constant adjoint flux equal to one in Equation 5. The PARTISN value of the average delayed neutron fraction is a little smaller than the MCNPX value because ^{238}U fissions have been neglected. In addition, the deterministic results (last two rows of Table VI; average beta) show that the k-ratio method results are 10 pcm higher relative to the application of Equation 12.

6. Neutron Lifetime

6.A Adjoint-Weighted Prompt Neutron Lifetime

The adjoint-weighted (effective) prompt neutron lifetime $\ell_{p,eff}$, sometimes referred to as the adjoint-weighted prompt removal lifetime, is the average time between the emission of a prompt neutron with an average importance by fission and its removal by either absorption or leakage.[103] It is defined by equation 15.[103]

$$\ell_{p,eff} \equiv \frac{\int_{4\pi} d\vec{\Omega} \int_0^{20MeV} dE \int_V d\vec{r}\, \varphi^+(\vec{r},E,\vec{\Omega}) \frac{1}{\mathrm{v}(E)} \varphi(\vec{r},E,\vec{\Omega})}{\frac{1}{k_{eff}} \int_{4\pi} d\vec{\Omega} \int_{4\pi} d\vec{\Omega}' \int_0^{20MeV} dE \int_0^{20MeV} dE' \int_V d\vec{r}\, \varphi^+(\vec{r},E',\vec{\Omega}') \cdot \chi(\vec{r},E') \cdot \nu(\vec{r},E) \cdot \Sigma_f(\vec{r},E) \cdot \varphi(\vec{r},E,\vec{\Omega})} \qquad (15)$$

Symbols of Equation 15 have the same physical meaning as in Equation 5; v is the neutron velocity. The use of Equation 15 for calculating of the effective prompt neutron lifetime ignores the contributions from (n,xn) and (γ,n) reactions and from the external neutron source; Appendix B addresses this issue.

In analogy with the YALINA booster analyses,[44] the adjoint-weighted prompt neutron lifetime has been calculated with Monte Carlo computer programs by introducing a small concentration of ^{10}B homogeneously distributed in the whole assembly. This calculation methodology is referred to as the 1/v method[104] and it is based on the first order perturbation theory and on the linear behavior of the ^{10}B(n,α) cross section as a function of the neutron energy. The linearity holds up to 0.5 MeV, therefore the insertion of ^{10}B for the application of the 1/v method not suitable in fast reactors. Under the previous assumptions, Equation 15 can be reduced to Equation 16.[76,104]

$$\ell_{p,eff} = \lim_{N^B \to 0} \frac{k_{eff} - k_{eff}^B}{k_{eff}^B} \frac{1}{N^B \overline{\sigma}_a^B \overline{\mathrm{v}}} \qquad (16)$$

In Equation 16, N^B is the small boron concentration (~10^{-7} atoms/barn·cm^3), $\overline{\sigma}_a^B$ is the ^{10}B microscopic (n,α) cross section (3837 barn) evaluated at the neutron speed $\overline{\mathrm{v}}$ of 2200 m/s, k_{eff}^B is the multiplication factor with boron, k_{eff} is the multiplication factor without boron. The statistical error associated with the prompt neutron lifetime has been calculated by equation 10 with x and y equal to k_{eff} and k_{eff}^B, respectively; the covariance term has been neglected.

The deterministic calculation of the adjoint-weighted prompt neutron lifetime followed the same approach used for the delayed neutron fraction illustrated in Section 5. For the prompt neutron lifetime, the algorithm used by the C program followed equation 17, where v_g represents the neutron speed for the g energy group. Appendix F lists the source code of the C program for calculating the prompt neutron lifetime. Equation 17 is the multi-group version of Equation 15, which is valid for a continuous energy domain.

$$\ell_{p,eff} \cong \frac{\sum_{i=1}^{I}\sum_{g=1}^{G} \Phi_{g,i}^{+} \cdot \frac{1}{\mathrm{v}_g} \cdot V_i \cdot \Phi_{g,i}}{\frac{1}{k_{eff}} \sum_{i=1}^{I}\sum_{g=1}^{G} \Phi_{g,i}^{+} \cdot \chi_{g,i} \cdot \left(\sum_{g'=1}^{G} V_i \cdot \nu\Sigma_{f,g',i} \cdot \Phi_{g',i} \right)} \qquad (17)$$

Equation 17 assumes that the adjoint and forward angular neutron fluxes have an isotropic distribution, since it uses the scalar neutron fluxes instead of the angular ones. This approximation has been also applied in other studies.[105] Of course, the exact calculation of the inner product involves the adjoint and forward angular fluxes as rigorously calculated by Favorite.[106] At present, PARTISN computer program does not have the capability to compute the inner product of the adjoint and forward angular fluxes.

The Monte Carlo and deterministic results (both based on the first order perturbation theory), are summarized in Table VII. The adjoint-weighted prompt neutron lifetimes calculated by MCNP/MCNPX and PARTISN show good agreement. The adjoint-weighted prompt neutron lifetime increases with the increase of the fuel rods number. As the number of fuel rods increases, the neutron multiplication and the number of neutrons scattered back from the reflector zone into the fuel zone increases. The scattered back neutrons have a

longer prompt neutron lifetime and generate fission neutrons in the fuel assembly, which increases the adjoint-weighted prompt neutron lifetime.

6.B Adjoint-Unweighted Prompt Neutron Lifetime

The adjoint-unweighted (average) neutron lifetime $\ell_{p,ave}$ (sometimes referred to as the adjoint-unweighted prompt removal lifetime) is the average time between the emission of a prompt neutron by fission and its removal by either absorption or leakage regardless of its importance.[103] It is defined by equation 18.[103]

$$\ell_{p,ave} \equiv \frac{\int_0^{20MeV} dE \int_V d\vec{r}\, \frac{1}{\mathrm{v}(E)} \Phi(\vec{r},E)}{\frac{1}{k_{eff}} \int_0^{20MeV} dE \int_V d\vec{r}\, \nu(\vec{r},E) \cdot \Sigma_f(\vec{r},E) \cdot \Phi(\vec{r},E)} \qquad (18)$$

Equation 18 is derived from Equation 15 by using a constant adjoint flux as a function of energy, angle, and space. The MCNP/MCNPX computer programs calculate the adjoint-unweighted prompt neutron lifetime (referred to as the *prompt removal lifetime*);[107] its statistical error is negligible. As expected, the adjoint-unweighted neutron lifetime calculated by MCNP/MCNPX and PARTISN computer programs is much larger than the effective neutron lifetime. The adjoint-weighted prompt neutron lifetime weights more the neutrons in the fuel zone (because they have higher importance), which have shorter prompt neutron lifetime relative to the neutrons in the reflector zone. The neutrons in the reflector zone have longer lifetime because of the lower absorption cross section of the reflector.

The adjoint-unweighted prompt neutron lifetime increases with the decrease of the fuel rods number. In this case, the neutrons of the fuel and the reflector zones are equally weighted for calculating the adjoint-unweighted prompt neutron lifetime and the neutron life time in the reflector zone is much larger than the fuel zone. Therefore, as the dimension of the active fuel zone decreases, the contribution of the reflector neutrons increases, this increases the adjoint-unweighted prompt neutron lifetime.

6.C Adjoint-Weighted Neutron Lifetime

The adjoint-weighted neutron life time (ℓ_{eff}) is the average time between the emission of a fission neutron with an average importance and its removal by either absorption or leakage and it is defined by Equation 19.[94]

$$\ell_{eff} \equiv \left(1-\beta_{eff}\right)\ell_{p,eff} + \sum_{k=1}^{6} \beta_{eff}\, w_k \left(\frac{1}{\lambda_k} + \ell_{p,eff}\right) \qquad (19)$$

In Equation 19, λ_k is the decay constant of the k delayed neutron precursor family. Using the effective delayed neutron fraction and prompt neutron lifetime obtained by

PARTISN without external neutron source, the relative fractions and delayed constants from the ENDF/B- nuclear data files for ^{235}U,[100] the adjoint-weighted neutron lifetime is about 83-85 ms for the three configurations, three order of magnitudes longer than the adjoint-weighted prompt neutron lifetime.

6.D Adjoint-Unweighted Neutron Lifetime

The adjoint-unweighted neutron life time (ℓ_{ave}) is defined as the average time between the emission of a fission neutron and its removal by either absorption or leakage regardless of its importance and it is defined by Equation 20.[94]

$$\ell_{ave} \equiv (1-\beta_{ave})\ell_{p,ave} + \sum_{k=1}^{6} \beta_{ave} w_k \left(\frac{1}{\lambda_k} + \ell_{p,ave} \right) \qquad (20)$$

Using the average delayed neutron fraction and prompt neutron lifetime obtained by PARTISN, the relative fractions and delayed constants from the ENDF/B nuclear data files for ^{235}U,[100] the average neutron lifetime is 73 ms for the three configurations. The adjoint-unweighted neutron lifetime is smaller than the adjoint-weighted neutron lifetime because the average delayed neutron fraction is smaller than the effective delayed neutron fraction and $1/\lambda$ is much larger than ℓ_p.

7. Mean Neutron Generation Time

The adjoint-weighted mean generation time Λ_{eff} is defined as the average time between the emission of a fission neutron with average importance and its removal by another fission reaction normalized to the number of neutrons in one neutron generation. The adjoint-unweighted mean generation time Λ_{ave} is defined as the average time between the emission of fission neutron with any importance and its removal by another fission reaction normalized to the number of neutrons in one neutron generation. The adjoint-weighted or the adjoint-unweighted neutron generation time can be calculated by dividing the adjoint-weighted or the adjoint-unweighted, prompt fission lifetime (defined as the average time between the emission of a fission neutron and its removal by another fission reaction; sometime referred to as the prompt fission lifespan) by the average number of neutrons per fission (ν),[107] respectively. Alternatively, the adjoint-weighted or the adjoint-unweighted mean generation time can also be calculated by dividing the adjoint-weighted or the adjoint-unweighted prompt neutron lifetime by the multiplication factor (k_{eff}),[94,107] respectively. For the YALINA thermal assembly, the adjoint-weighted mean generation time without external neutron source calculated by PARTISN is 85, 84, and 83 μs and the adjoint-unweighted value is 273, 258 and 238 μs for 216, 245, and 280 fuel configurations, respectively.

8. Spatial Neutron Flux Distributions

8.A MCNP/MCNPX Calculations for Cf, D-D, and D-T Neutron Sources

The neutron flux distribution in the axial direction (parallel to the fuel rods) has been calculated for the assembly driven by the Cf, the D-D, and the D-T external neutron sources in the different experimental channels. Figures 45 through 49 show the neutron flux distributions in the EC1, EC2, EC3, EC5, and EC6 experimental channels. In all the distributions, the negative coordinate refers to the beam tube side and the origin is located at the copper disk surface facing the deuteron beam. Figure 50 shows the radial flux distribution in the EC7 experimental channel, which is perpendicular to the fuel rods. All results in these Figures have been obtained by MCNP/MCNPX calculations. In all plots of this study, the neutron flux unit is neutrons per source particle and square centimeter ($n/sp \cdot cm^2$). The following remarks concern the neutron flux spatial distributions:

- In all the experimental channels, the flux intensity from the Cf neutron source is always higher than the flux intensity from the D-D neutron source and the flux intensity from the D-D neutron source is always higher than the flux intensity from the D-T neutron source. This is due to the higher leakage from the D-D source neutrons relative to the Cf source neutrons and the higher leakage from the D-T source neutrons relative to the D-D source neutrons. The average source neutron energy increases in the following order: Cf < D-D < D-T. Neutron leakage is proportional to the average source neutron energy since the assembly has a small volume and thin reflector thickness.
- In all the experimental channels, the neutron flux intensity increases as the number of the fuel rods increases due to the higher neutron multiplication.
- The neutron flux intensity in the experimental channels decreases as the distance between the experimental channel and the neutron source increases.
- The axial neutron flux distributions have cosine shapes in all the experimental channels except the radial channel (EC7), which shows an exponential attenuation.

For the californium neutron source, the calculation of the neutron flux distribution has been repeated without the beam tube and air filling the beam tube volume. The results are shown in Figures 51 through 56. As observed in Section 4.B the beam tube behaves as a neutron absorber and its removal increases the neutron flux.

In Figures 45 through 56, the neutron flux has been tallied over a cylindrical volume with 1.2-cm radius and 5-cm length. For all the experimental channels except the EC7 experimental channel, the cylindrical tally volume is parallel to the fuel rods and it is centered at the z axis ticks. For the EC7 experimental channel, the cylindrical tally volume is perpendicular to the fuel rods and it is centered at the x axis ticks.

8.B MCNP/MCNPX, MONK, PARTISN, and TORT Calculations for Criticality and D-D Neutron Source

The comparisons between the neutron flux spatial distributions calculated by MCNP/MCNPX and PARTISN in criticality mode are shown in Figures 57 through 62 and Figures 63 through 68 for the 216 and 280 configurations, respectively. The agreement between MCNP/MCNPX and PARTISN is within a few percent.

Figures 69 through 74 illustrate the neutron flux profile, obtained by MCNP/MCNPX, MONK, PARTISN, and TORT for the 216 configuration driven by the D-D neutron source. This configuration has been investigated because it has the lowest neutron multiplication. The agreement between the Monte Carlo codes MCNP/MCNPX and MONK is within the statistical errors (which is indicated by two standard deviations). The results obtained by the PARTISN code are very close to the ones obtained by the TORT code. When the MCNP/MCNPX code uses the same neutron source model as PARTISN and TORT (as described in Section 2.C.2), the agreement between MCNP/MCNPX and deterministic results is improved.

In the PARTISN and TORT simulations, the fuel unit cell with 2-cm side has two intervals in the x and y axes and several intervals in the z axis parallel to the fuel rod. In the x-y plane, the fuel unit cell is divided into four equal squares with 1 cm side. Each experimental channel in the fuel zone covers one fuel and moderator unit cell in the x-y plane, therefore the neutron flux at each z interval is the average of the four flux values in the unit cell. At each z interval, the neutron flux in the experimental channels of the moderator zone has been calculated by interpolating (instead of averaging) four flux values in the xy plane spatial mesh. This interpolation calculated the neutron flux at the center of each experimental channel in the reflector zone. The values of the neutron flux in the z intervals have been averaged over a 5-cm length.

In the MCNP/MCNPX and MONK simulations, the neutron flux in the experimental channels is tallied over a cylindrical volume with 1.2 cm radius (which matches the circular cross section of the experimental channel) and 5 cm length, as in Section 8.A.

9. Neutron Spectrum

9.A MCNP/MCNPX Calculations for Cf, D-D, and D-T Neutron Sources

The neutron spectra in the experimental channels of the facility (EC1, EC2, EC3, EC5, EC6, and EC7) driven by the Cf, the D-D, and the D-T neutron source are illustrated in Figures 75 through 93. In the experimental channels of the fuel zone, the neutron spectra have been averaged over a cylindrical volume (the axis of the cylinder is parallel to the fuel rod) with 1.2 cm radius (the radius of the experimental channel) and 10 cm length centered at the middle of the active fuel length. In the EC7 experimental channel, the neutron spectra have been averaged over a cylindrical volume (the axis of the cylinder is perpendicular to the fuel rod) with 1.2 cm radius and 5 cm length located next to the fuel zone. The neutron spectra have been sampled using F4 flux tally with 172 energy groups. In all neutron spectra plots of this study, the neutron flux has been first normalized to unity and then to lethargy. Several observations can be made from the neutron spectrum plots:

- The neutron spectrum in the experimental channel is thermalized further as the distance between the neutron source and the experimental channel increases.
- The thermal neutron flux in the EC3, EC5, EC6, and EC7 experimental channels increases as the number of the fuel rods in the assembly decreases. The fuel mass

decreases but the polyethylene and graphite mass do not change, which increases the reflector/moderator around the active fuel zone.

- The neutron spectrum in the EC2 experimental channel is insensitive to the variation of the number of fuel rods since this channel is in the middle of the active fuel zone.
- The fast component of the spectrum in the EC1 experimental channel increases as the number of the fuel rods decreases. This channel is very close to the neutron source. The decrease of the fuel mass in the assembly enhances the contribution from the external neutron source, which emits fast neutrons.
- In all experimental channels, the neutron spectra due to the Cf, D-D or D-T neutron source are very similar except for the peaks at 2.45 MeV for the D-D source neutrons and at 14.1 MeV for the D-T source neutrons. The relative contribution of the peak to the neutron spectrum increase as the number of fuel rods decreases and/or the channel distance from the source decreases.
- The D-T neutron source peak is visible in all the experimental channels, including the ones in the reflector. This indicates that a fraction of the D-T source neutrons reaches all the experimental channels without collision. In addition, some of the D-T source neutrons leak outside the assembly as already observed in Section 4.D.
- For the Cf neutron source, the neutron spectrum in the experimental channels does not change when the beam tube is removed except for the EC1 experimental channel. This channel is close to the neutron source where the beam tube acts as neutron moderator and absorber. Figure 81 shows the impact of the beam tube removal on the neutron spectrum of the EC1 experimental channel. The fast and thermal parts of the neutron spectrum are increased.

9.B MCNP/MCNPX, MONK, PARTISN, and TORT Calculations for Criticality and D-D Neutron Source

The comparison between the MCNP/MCNPX and PARTISN neutron spectra for the 216 and the 280 configurations obtained from criticality calculations is shown in Figures 94 through 99 and 100 through 105, respectively. The comparison between MCNP/MCNPX, MONK, PARTISN, and TORT neutron spectra for the 216 configuration driven by the D-D neutron source is shown in Figures 106 through 111. The 216 configuration has been investigated because it has the lowest k_{eff}, therefore the effect of the external neutron source is enhanced. By contrast, the 280 configuration is the closest one to criticality. Generally, the Monte Carlo and deterministic results show an excellent agreement. However, the deterministic codes cannot accurately represent the D-D neutron source peak at 2.45 MeV due to the discrete representation of the energy domain. The PARTISN and TORT modeling approximation of filling the experimental channels of the reflector with graphite does not impact the neutron spectrum. The neutron spectrum calculated by the MONK code in the epithermal region exhibits some statistical fluctuations because the MONK code can run only on a single computer processor, which limits the number of source neutrons per simulation.

The PARTISN and TORT calculation methodology of the neutron spectra followed the one illustrated at the end of Section 8.B, with the only exception that the neutron spectrum

has been averaged over an axial length of 10 cm. The Monte Carlo calculation methodology followed the one illustrated in Section 9.A.

The present results improve the ones previously obtained[108] by collapsing the WIMSD 172 energy group nuclear data library into 69 energy group library. The neutron spectrum calculated by the DRAGON code after the collapsing process to 69 energy groups exhibits some spikes and dips which have no physical explanation.[108] The spikes and dips disappear if DRAGON uses the WIMSD 69 energy group library, as always performed in the work of this report. The WIMSD 172 energy groups nuclear data library has never been used in the work of this report.

10. Reaction Rate Spatial Distribution

10.A Criticality Calculations

Figures 112-114 and 115-117 plot the ^{3}He(n,p) reaction rate profile in the experimental channels (EC1, EC2, and EC3) of the fuel zone for the 216 and 280 configurations, respectively. The MCNP/MCNPX and PARTISN results, which have been obtained by criticality calculations, show a good agreement. The MCNP/MCNPX results for the reaction rate have been obtained by tallying the neutron flux in 11 volumes, each with 0.5 cm radius and 1 cm length. The ticks on the z axis mark the center of the MCNP/MCNPX tally volumes. The PARTISN results for the reaction rate have been obtained by multiplying the one group microscopic cross section calculated by MCNP/MCNPX (averaged in the tally volume) by the one group flux calculated by PARTISN. The PARTISN flux has been calculated by averaging four values in the x-y plane as described in Section 8.B and by interpolating two values on the z axis. The two values of the PARTISN flux along the z axis have been linearly interpolated to obtain the values at the center of the MCNP/MCNPX tally volumes. This calculation methodology has been applied to all reaction rate results obtained by deterministic calculations. The unit of all the time independent reaction rates of this study is barn per source particle and square centimeter (10^{-24}/sp).

10.B Californium, D-D and D-T Neutron Sources Calculations

The ^{3}He(n,p) and ^{235}U(n,f) reaction rates calculated by MCNP/MCNPX, PARTISN, and TORT for the three YALINA thermal configurations driven by the three neutron sources (Cf, D-D, and D-T) are plotted in Figures 118 through 124 and 125 through 131, respectively. In these calculations, the irradiation samples have not been explicitly modeled, therefore the self-shielding effect is not considered. As stated in Section 2.C.3, the PARTISN and the TORT calculations for the D-T neutron source have not been performed since the upper energy of the 69 energy groups WIMSD nuclear data library is 10 MeV, which is much smaller than the average energy of the D-T source neutrons. The MCNP/MCNPX and the deterministic results show a good agreement. The agreement is better for the Cf neutron source relative to the D-D neutron source. As explained in Section 2.C, PARTISN and TORT use a volume neutron source whereas MCNP/MCNPX uses a point or surface neutron source (located at z = 0 cm) for Cf or D-D source neutrons, respectively. The higher energy of D-D neutrons, relative to Cf neutrons, enhances the difference in the results because of the different spatial modeling of the source neutrons. D-D source neutrons have a higher

average mean free path relative to Cf source neutrons, which increase neutron leakage probability. The MCNP/MCNPX and PARTISN reaction rate calculation methodologies are the same as the ones used in Section 10.A.

The In(n,γ), ^{197}Au(n,γ), and ^{55}Mn(n,γ) reaction rates in the experimental channels EC2 and EC7 obtained by MCNP/MCNPX and PARTISN for the californium and D-D neutron sources are plotted in Figures 132 through 139. In these calculations the irradiation samples have not been explicitly modeled, the self-shielding effect is ignored. In these Figures, the ticks on the z axis indicate the center of the MCNP/MCNPX flux tally volume. The samples have 0.5-cm radius and a thickness of 350-μm for In(n,γ), 33-μm for ^{197}Au(n,γ), and 171-μm for ^{55}Mn(n,γ). In this set of MCNP/MCNPX calculations, the EC2 and the EC7 experimental channels have been filled with polyethylene. The PARTISN calculations have been performed with EC2 filled by polyethylene and EC7 filled by graphite. The agreement between PARTISN and MCNP/MCNPX results improves when the latter code uses a distributed source, as depicted in Figures 138 and 139 and described in Section 2.C.

In the EC2 experimental channel, the PARTISN calculation methodology followed the same approach used for the criticality calculations in Section 10.A. In the EC7 experimental channel, this approach has been modified as follows:

- In the x axis direction: A linear interpolation between neutron flux values is used to generate the values at the center of the MCNP/MCNPX tally volumes along the x axis.
- In the y axis direction: An average between two neutron flux values is used to generate the value at y = 0 since the two intervals have the same width.
- In the z axis: A single value of the neutron flux is used covering the interval from 0 to 0.725 cm in the z axis.

Figures 140-143 show the MCNP/MCNPX results for the same calculations of Figures 132-139 with the exception that the irradiation samples have been explicitly modeled; therefore the self-shielding effect is considered.

The values of the reaction rates and the self-shielding attenuation factors are proportional to the magnitude of the microscopic cross sections in the thermal energy region, in the following order: ^{3}He(n,p) > ^{235}U(n,f) > In(n,γ) > ^{197}Au(n,γ) > ^{55}Mn(n,g).[44]

10.C Comparison between Numerical and Experimental Results for Cf Neutron Source

The comparison between MCNP/MCNPX and PARTISN results for the neutron flux and between numerical and experimental results for the ^{3}He(n,p) reaction rate are shown in Figures 144 through 149 and 150 through 155 for the 245 and 280 configurations, respectively. In the reaction rate plots, the maximum of the experimental values has been normalized to the maximum of the MCNP/MCNPX value since the external neutron source strength has not been measured in the experiments. The 216 configuration has not been considered in this section because measurements have not been performed with this configuration. Only the EC1, EC2, and EC3 experimental channels of the fuel zone have been used in the experiments. In these calculations and experiments, the beam tube has been removed from the assembly and replaced by air. Generally, the comparison between the deterministic, the Monte Carlo, and the experimental results shows a good agreement. At

the center of the EC1 experimental channel, which is close to the neutron source, the small difference between MCNP/MCNPX and PARTISN is due to the different modeling of the neutron source (see Section 2.C.1) and the material homogenizations. When both computer programs distribute the source over the same volume, the agreement improves, as shown in Figure 144. In these calculations, the irradiation sample has not been modeled and therefore the self-shielding effect has not been considered.

The MCNP/MCNPX and the PARTISN calculation methodologies for the neutron flux are explained in Section 8.B. The MCNP/MCNPX ^{3}He(n,p) calculation methodology followed the same approach used for the criticality calculation used in Section 10.A. The PARTISN ^{3}He(n,p) reaction rate has been obtained by multiplying the PARTISN neutron flux of section 8.B by the MCNP/MCNPX ^{3}He(n,p) microscopic cross section of Section 10.A.

11. Time Dependent Reaction Rates from D-D and D-T Pulsed Neutron Sources

11.A MCNP/MCNPX Results for D-D and D-T Pulsed Neutron Sources Calculations

Figures 156 through 158 plot the ^{3}He(n,p) and ^{235}U(n,f) reaction rates in the EC1, EC2, and EC5 experimental channels for the D-D and D-T pulsed neutron sources for the three fuel configurations of YALINA thermal. In these plots, the reaction rate has been normalized to the maximum value. The neutron source duration is 5 μs and the period is 20 ms. The reaction rate has been averaged over a tally volume with 0.45 cm radius and 25 cm length placed at the middle of the active fuel length. The MCNP/MCNPX results have been calculated using the pulse superimposition method developed for the YALINA booster assembly.[45,46] Appendix G lists the source code of the C program used for superimposing 25000 times the MCNP/MCNPX reaction rate obtained from a single pulse calculation; the program simultaneously uses 25 processors with the MPI software. In the MCNP/MCNPX numerical simulations for a single pulse, the neutron detector was not modeled.

The time dependent reaction rates show that the maximum value delay increases as the distance between the neutron source and the experimental channel increases. This delay is corresponding to the required time for the source neutrons to travel from the birth location to the detector position.

As the distance between the neutron source and the experimental channel increases, the delayed neutron background level increases relative to the prompt part. The neutron source contribution to the prompt part of the reaction rate decreases as this distance increases. This effect is stronger for the YALINA configuration with 216 fuel rods, relative to the other configurations since source neutrons contribute more to the neutron population because of the lower neutron multiplication.

The reaction rates from the delayed neutrons increase as the number of the fuel rods increases. The increase of the fuel rods number increases the neutron multiplication, which diminishes the neutron source contribution to the reaction rate. Consequently, the delayed neutron contribution increases relative to the contribution from the source neutrons.

The use of D-T neutron source results in a higher delayed neutrons background level in the reaction rate, relative to the use of D-D neutron source. The higher leakage of D-T source neutrons reduces the source contribution to the prompt part of the reaction rate.

11.B Comparison between MCNP/MCNPX and Experimental Results for D-D Pulsed Neutron Source

The comparisons between MCNP/MCNPX calculations and the experimental results for the ^{3}He(n,p) reaction rate as a function of time for the D-D pulsed neutron source are shown in Figures 159 through 162 for the different YALINA thermal configurations. In these plots, the reaction rate has been normalized to the maximum value. In the numerical simulations, the neutron detector has 0.45 cm radius and 25 cm length and it has a gas mixture with 8.14 mg/cm^3 density consisting of 80% ^{3}He and 20% atomic fraction natural krypton. The detector is explicitly modeled in the calculations to include the self-shielding effect. The comparison between MCNP/MCNPX and experimental results shows a good agreement.

11.C Comparison between MCNP/MCNPX and PARTISN Results for D-D Pulsed Neutron Source

Figures 163 through 166 show the MCNP/MCNPX and PARTISN reaction rate results due to the D-D pulsed neutron source for the 216 and 280 YALINA thermal configurations. Only the first 4 milliseconds of the pulse period have been calculated since the PARTISN computer program does not model delayed neutrons. The neutron source pulse duration is 5 μs. In these simulations the detector has not been modeled. The comparison between the Monte Carlo and the deterministic results is quite good; the match has been improved by using similar time steps in the Monte Carlo and the deterministic calculations. In MCNP/MCNPX, short time steps increase the statistical error while tracking the same number of source neutrons. In PARTISN, the time step is automatically adjusted according to the flux change between two consecutive time steps; however, it is possible to set an arbitrary number of time steps during any time interval. The PARTISN computer program has been modified to suppress the writing of the restart file. The original version of PARTISN writes a restart file of 20 Gbyte at each time step.

The MCNP/MCNPX reaction rate calculation methodology followed the one used in Section 11.A without the pulses superimposition. The PARTISN neutron flux calculation methodology followed the one illustrated in Section 8.B. The PARTISN neutron flux has been averaged over a volume with 20 cm length in the axial direction while the MCNP/MCNPX neutron flux has been averaged over a volume with 25 cm length in the axial direction. The PARTISN reaction rate has been calculated by multiplying the PARTISN 69 energy groups neutron flux by the 69 energy groups ^{3}He(n,p) microscopic cross section obtained by MCNP/MCNPX calculations. The MCNP/MCNPX tally of the microscopic cross section has been averaged over a volume with 0.45 cm radius and 20 cm length located at the center of the experimental channel. Figure 167 plots the 69 groups ^{3}He(n,p) microscopic cross section calculated by MCNP/MCNPX for the 216 and the 280 YALINA thermal configurations in the EC2 and the EC5 experimental channels when the assembly is driven by the D-D neutron source. The MCNP/MCNPX 69 groups ^{3}He(n,p) microscopic cross section does not change with the experimental channel or the number of fuel rods in the assembly. The

MCNP/MCNPX cross section matches the original ENDFB-VI.8 data. In all plots of this section, the reaction rate unit is barn per millisecond and square centimeter (b/cm^2·ms).

11.D Bell and Glasstone Correction Factors for Reactivity Measurements

In a subcritical assembly, the measured reactivity depends on the detector location since different experimental channels provide different reactivity values. In order to mitigate this spatial dependency, Bell and Glasstone[88] introduced a correction factor (c). When the area method[81] is applied for pulsed neutron source experiments, the correction factor is given by Equation 21.

$$c \equiv \frac{\rho_{cri}}{\rho_{src}} = -\frac{A_d}{A_p \cdot \beta_{eff}} \frac{k_{eff} - 1}{k_{eff}} \qquad (21)$$

In Equation 21, ρ_{cri} is the reactivity calculated by computer programs in criticality mode without external neutron source and ρ_{src} is the reactivity calculated by computer programs with the external neutron source. Using the area method, which is based on point kinetics,[81] ρ_{src} (expressed in dollars) is the ratio between the prompt (A_p) and the delayed (A_d) areas. These are the areas underneath the curve of the detector reaction rate measurements as a function of time. Figure 168 shows these areas for the 245 configuration using the EC2 experimental channel from the D-D neutron source with 5 μs neutron pulse duration and 20 ms period. The results shown in Figure 168 have been obtained by MCNPX simulations. In order to estimate ρ_{src}, the prompt and the delayed areas can be calculated by computer programs either using *kinetic* or *static* simulations. In the kinetic simulations, the reaction rate is tallied as function of time for one single neutron pulse and the pulse superimposition method[45,46] is applied. The total and delayed areas are calculated with one neutron transport simulation by integrating the reaction rate over the single period after performing the pulses superimposition. In the kinetic method, the prompt area is obtained as the difference between the total and delayed areas. The kinetic method can only be applied by MCNP/MCNPX computer code since it has the explicit representation of the delayed neutrons. In the static method,[18] the integrals over time of the prompt and the total reaction rates are tallied by two separate simulations, which model the neutron source as time independent. The static method can be applied by either MCNP/MCNPX or PARTISN codes. In the case of MCNP/MCNPX, the normal simulation estimating the total area A_t is repeated after suppression of the delayed neutrons to calculate the prompt area. In the static method, the statistical error associated to the areas ratio is given by Equation 10 with $x = A_t - A_p$ and $y = A_p$. The statistical error of $A_t - A_p$ is calculated by Equation 22.

$$\sigma_{Ax+By} = \sqrt{A^2\sigma_x^2 + B^2\sigma_y^2 + 2AB\,\mathrm{cov}(x,y)^2} \qquad (22)$$

In Equation 22, x and y are random variables, $E(x)$ and $E(y)$ are the expected values of the random variables, A and B arbitrary constants, and cov(x,y) is the covariance of x and y (which has been set to zero). In the case of PARTISN, the static method uses two different nuclear data libraries: with and without delayed neutrons, the same libraries used for the calculation of the effective delayed neutron fraction in Section 5.A.

Table VIII summarizes the correction factors obtained by MCNP/MCNPX for the 245 and the 280 configurations using the EC2 and the EC5 experimental channels with the ^{3}He detector and the D-D neutron source. Both the kinetic and the static methods have been used. In the kinetic case, the pulsed neutron source has 5μs duration and 20 ms period and the calculation has been performed without and with the detector model in the experimental channel, first and second rows of Table VIII, respectively. In the latter case, two detectors have been placed in the assembly to simulate the experiment. For the EC2 experimental channel, the two detectors are located in EC2 and EC3. For the EC5 experimental channel, the two detectors are located in EC5 and EC6. All the detectors have 0.45 cm radius and 25 cm length. A good agreement between the correction factors calculated using different methods was obtained. The correction factors in the EC2 and the EC5 experimental channels are smaller and higher than unity, respectively. Near the source zone, the prompt area is enhanced by the source neutrons through direct contribution to the detector reaction rate and indirect contribution from the prompt fission neutrons. In addition, a fraction of the delayed neutrons is absorbed by the beam tube (the delayed neutrons have a softer neutron spectrum relative to the prompt neutrons and they are more likely to be absorbed by the beam tube), which reduces the delayed area. According to Equation 21, these effects result in a correction factor of less than one. In the reflector zone, away from the neutron source these effects are reversed, which enhances the delayed neutron area and reduce the prompt area, which result in a correction factor higher than unity.

In the static method, the calculation of the statistical error of the correction factor used the statistical errors of the prompt and the total areas, the delayed neutron fraction based on Meulenkamp and van der Marck method,[89] and the multiplication factor with Equations 10, 22, and 23. The symbols of Equation 23 are the same as of Equation 10.

$$\sigma_{x\cdot y} = \sigma_{x\cdot y} = E(x)\cdot E(y)\sqrt{\frac{\sigma_x^2}{E^2(x)} + \frac{\sigma_y^2}{E^2(y)} + 2\frac{\mathrm{cov}(x,y)}{E(x)E(y)}} \qquad (23)$$

In the kinetic method calculations, a lower number of source neutrons were used as indicated in Table VIII. Therefore, the correction factors have higher statistical errors relative to the results obtained by the static method.

Table IX lists the correction factors obtained by PARTISN static calculations for the 245 and 280 configurations with the D-D neutron source without explicit detector modeling. The calculation of the prompt and the total ^{3}He(n,p) reaction rates followed the approach discussed in Section 10.B; the reaction rate has been averaged over 24.55 cm axial length. The deterministic correction factors results are in good agreement with the Monte Carlo results without detector modeling as shown in the first row of Table VIII.

Figures 169 and 170 plot the correction factor calculated by PARTISN as a function of the position along the z axis parallel to the fuel rods for the 245 and the 280 configurations, respectively, with the D-D neutron source for the different experimental channels. Only EC1 and EC2 experimental channels have a correction factor smaller than unity because of the close proximity of these two channels to the neutron source.

Table X summarizes the multiplication factors calculated in criticality mode by the Monte Carlo codes MCNP/MCNPX and EA-MC.[109] The EA-MC calculations have been performed at

Uppsala University with JENDL-3.2 nuclear data Files.[109] The MCNP/MCNPX calculations have been performed for the same configuration with different representation for the measuring detector: a) without detector model, b)with one detector modeled in EC2 or EC5, and with two detector modeled in two different channels: EC2 and EC3 or EC5 and EC6. All detectors have 0.45 cm radius and 25 cm length. When the detector is placed in the fuel zone (EC2), the multiplication factor diminishes by 300 pcm, relative to the values obtained without modeling the detector. The addition of the second detector in the fuel zone (EC2 and EC3) lowers the multiplication factor by 150-180 pcm, relative to the values obtained with one detector model. The insertion of the detectors in the reflector zone does not change much the neutron multiplication factor. For the 245 and the 280 configurations, the EA-MC k_{eff} values differ less than 290 pcm from the MCNP/MCNPX values as shown in Table X. For the 216 configuration, the MCNP/MCNPX k_{eff} value is 0.87902±9 and the EA-MC k_{eff} value is 0.88327±4; the 420 pcm difference between the two values is also caused by different nuclear data files, ENDFB-VI.6 for MCNP/MCNPX and JENDL-3.2 for EA-MC.

Following the Bell and Glasstone approach, the experimental reactivity (ρ_{exp}) is corrected by the c correction factor according to Equation 24, where ρ_c represents the corrected experimental reactivity.

$$\rho_c = c \cdot \rho_{\exp} \qquad (24)$$

Table XI reports the uncorrected experimental k_{eff}, and the corrected experimental k_{eff} for the 245 and the 280 configurations. The experimental reactivity has been measured by the area method using the ^{3}He(n,p) detector for the D-D pulsed neutron source with 5μs pulse duration and 20 ms period. The experimental measurements have been carried out using two detectors. For EC2 experimental channel, the other detector is located in EC3. For EC5 experimental channel, the other detector is located in EC6. The detector acquiring the reaction rate signal has 0.45 cm radius and 25 cm length. The correction factor has been applied using both the kinetic and the static methods. The first case has been investigated with and without modeling the detector. The results of Table XI show that the difference between the corrected experimental k_{eff} and the MCNP/MCNPX k_{eff} with one detector model (second row of Table X) is between 400 and 1200 pcm. The statistical error of the corrected k_{eff} is between 180 and 400 pcm, which has been calculated with Equations 10, 22, and 23.

The present numerical results have been obtained by MCNPX using the fuel composition of the IAEA benchmark specifications,[40,42] which is reported in Table XII. This composition neglects the presence of ^{234}U and the fuel impurities. Experimental analyses based on the gamma spectroscopy of 280 fuel rods estimated an average uranium mass of 78.388 g per fuel rod. The composition given in the middle column of Table XII takes into account ^{234}U, fuel impurities, and the average uranium mass per rod. For the 280 configuration, this composition (middle column of Table XII) decreases the criticality multiplication factor down to 0.95487±4. The right column in Table XII gives the fuel composition with the 2.1% reduction in the average uranium mass per rod; this adjustment results in k_{eff} of 0.94961±4, which matches better the experimental results.

12. Conclusions

This study utilized the state of the art nuclear data files, neutron transport computer programs, and calculation methodologies to analyze the YALINA thermal assembly. The computing programs includes: NJOY, MCNP/MCNPX, MONK, DRAGON, PARTISN, and TORT. The analyses covered: k_{eff}, source (Cf, D-D, and D-T) multiplication factor, delayed neutron fraction, prompt neutron lifetime and generation time, neutron flux profiles, neutron spectra, static reaction rates (^{3}He(n,p), ^{235}U(n,f), ^{197}Au(n,γ), ^{115}In(n,γ) and ^{55}Mn(n,γ)), and time dependent ^{3}He(n,p) reaction rates. Monte Carlo and deterministic results show a very good agreement.

For source driven problems, the agreement between MCNP/MCNPX and PARTISN improves when both codes use the same source model for space and energy. The PARTISN and TORT codes have not been used to model the D-T neutron source because the WIMSD 69 energy groups library has an upper energy of 10 MeV.

The WIMSD nuclear data library has been enlarged with additional isotopes for describing the polyethylene material, alloy impurities, and uranium isotopes without delayed fission neutrons. The enlarged library has been created with NJOY program by processing ENDF/B nuclear data files. The uranium isotopes without delayed neutrons data have been used for the calculation of the effective delayed neutron fraction by the k-ratio method and the Bell and Glasstone correction factor.

The k-ratio method for calculating the effective delayed neutron fraction has been applied not only to the MCNP/MCNPX simulations but also to the PARTISN analyses by preparing two WIMSD nuclear data libraries: one library including all the produced fission neutrons and the other including only the prompt neutrons.

In order to facilitate the deterministic modeling of the assembly, an interface computer program DRAGON2PARTISN has been written in C programming language. Once the DRAGON simulations have been carried out, the interface software generates the PARTISN macroscopic cross sections. The DRAGON2PARTISN source code has been modified (and changed into DRAGON2TORT) to generate also the cross sections for the TORT analyses.

The time dependent PARTISN ^{3}He(n,p) reaction rate for a single D-D neutron pulse matches the MCNP/MCNPX results. For these analyses, the original version of the PARTISN code has been modified to avoid the writing of restart files every time step, which saves significant computer resources.

The comparison between the MCNP/MCNPX and the experimental results for the multiplication factor and the ^{3}He(n,p) reaction rate, set by a californium neutron source or a D-D pulsed neutron source show a good agreement. This confirms the accuracy of the present calculation models and calculational methodologies.

Acknowledgements

This project is supported by the U.S. Department of Energy, National Nuclear Security Administration, Office of Global Nuclear Material Threat Reduction (NA213).

The authors acknowledge and thank Drs Jeffrey Favorite and Randy Baker (Los Alamos National Laboratory) for their suggestions on PARTISN calculations, Prof. Guy Marleau (Polytechnique de Montréal) for his suggestions on DRAGON calculations, Dr. Malcolm Armishaw (ANSWERS Software) for his suggestions on MONK calculations, and Dr. Sai-Chi Mo (Argonne National Laboratory) for his suggestions on NJOY calculations.

References

[1] C. D. BOWMAN et al., "*Nuclear Energy Generation and Waste Transmutation Using an Accelerator-Driven Intense Thermal Neutron Source*," Nuclear Instruments and Methods in Physics Research A 320, pp. 336, 1992.

[2] C. RUBBIA et al., "*Experimental Verification of the Concept of Energy Amplification by High Energy Induced Cascade*," CERN/ISC, CERN 93-31, 1993.

[3] V. TUYLE et al., "*Accelerator-Driven Subcritical Target Concept for Transmutation of Nuclear Wastes*," Nuclear Technology 101/1, pp. 1-17, 1993.

[4] H. TAKAHASHI, "*Transmutation of High-Level Radioactive Waste by a Charged-Particle Accelerator*," Nuclear Technology 111/1, pp. 149-162, 1995.

[5] M. SEGEV et al., "*Transmutation of Neptunium, Americium, Technetium, and Iodine in Fast Spectrum Cores Driven by Accelerated Protons*," Nuclear Science and Engineering 122/1, pp. 105-120, 1996.

[6] C.D. BOWMAN, "*Accelerator-driven Systems for Nuclear Waste Transmutation*," Annual Review of Nuclear and Particle Science 48/1, pp. 505-556, 1998.

[7] W.S. PARK, "*TRU Incineration Characteristics of Thermal and Fast Subcritical Reactors*," Annals of Nuclear Energy 26/16, pp. 1497-1508, 1999.

[8] H. TAKANO et al., "*Transmutation of Long-Lived Radioactive Waste Based on Double-Strata Concept*," Progress in Nuclear Energy 37/1-4, pp. 371-376, 2000.

[9] T. MUKAIYAMA et al., "*Review of Research and Development of Accelerator-Driven System in Japan for Transmutation of Long-Lived Nuclides*," Progress in Nuclear Energy 38/1-2, pp. 107-134, 2001.

[10] T. MUKAIYAMA et al., "*Partition and Transmutation Studies at JAERI both under OMEGA Program and High Intensity Proton Accelerator Project*," Progress in Nuclear Energy 40/3-4, pp. 403-413, 2002.

[11] T. SUGAWARA et al., "*Recent Activities for Accelerator Driven Systems in JAEA*," GLOBAL 2009, 2009.

[12] Y.Q. SHI et al., "*The Design, Construction, and Preliminary Experiment of China's ADS Subcritical Experimental Assembly VENUS-1*," Nuclear Technology 159/1, pp. 106-118, 2007.

[13] S. SOULE et al., "*Neutronic Studies in Support of Accelerator-Driven Systems: the MUSE Experiments in the MASURCA Facility*," Nuclear Science and Engineering 148, pp. 124-152, 2004.

[14] P. BAETEN and A. ABDERRAHIM, "*Measurement of Kinetic Parameters in the Fast Subcritical Core MASURCA*," Nuclear Engineering and Design 230/1-3, pp. 223-232, 2004.

[15] M. PLACSHY et al., "*Importance of the MUSE Experiments for Emerging ADS Concepts from the Nuclear Data Viewpoint*," Annals of Nuclear Energy 32, pp. 843-856, 2005.

[16] P. BAETEN et al., "*Analytical Investigation and Experimental Application of the Source Modulation Technique to Measure* ρ/β_{eff}," Progress in Nuclear Energy 48/6, pp. 550-558, 2006.

[17] A. BILLEBAUD, "*Prompt Multiplication Factor Measurements in Subcritical Systems: from MUSE Experiment to a Demonstration ADS*," Progress in Nuclear Energy 49/2, pp. 142-160, 2007.

[18] F. GABRIELLI et al., "*Inferring the Reactivity in Accelerator-Driven Systems: Corrective Spatial Factors for Source-Jerk and Area Methods*," Progress in Nuclear Energy 50, pp. 370-376, 2008.

[19] J.F. LEBRAT et al., "*Global Results from Deterministic and Stochastic Analysis of the MUSE-4 Experiments on the Neutronics of Accelerator-Driven Systems*," Nuclear Science and Engineering 158/1, pp. 49-67, 2008.
[20] C. RUBBIA et al., "*Neutronic Analyses of the TRADE Demonstration Facility*," Nuclear Science and Engineering 148, pp. 103-123, 2004.
[21] C. JAMMES et al., "*Comparison of Reactivity Estimations Obtained from Rod-Drop and Pulsed Neutron Source Experiments*," Annals of Nuclear Energy 32, pp. 1131-1145, 2005.
[22] X.CHENG et al., "*Thermal-Hydraulic Analysis of the TRADE Spallation Target*," Nuclear Instruments and Methods in Physics Research A 562/2, pp. 855-858, 2006.
[23] G. BANDINI et al., "*Transient Analyses to Support the Definition of TRADE Experiment Start-Up Procedures*," Nuclear Instruments and Methods in Physics Research A 562/2, pp. 906-909, 2006.
[24] F. GABRIELLI and A.D'ANGELO, "*Analyses of Operational Transients in the TRADE Experimental Program*," Nuclear Engineering and Design 239/1, pp. 2349-2356, 2009.
[25] General Atomics, "*TRIGA MARK II Reactor General Specifications and Description*," 1964.
[26] C.C. JAMMES et al., "*Absolute Reactivity Calibration of Accelerator-Driven Systems after RACE-T Experiments*," PHYSOR06, 2006.
[27] B. Carluec, "*The European Project PDS-XADS, Preliminary Design Studies of an Experimental Accelerator-Driven System*," Proc. Int. Workshop on P&T and ADS Development, Mol, Belgium, 2003.
[28] A. SORDO et al., "*Material Selection for Spallation Neutron Source Windows Application to PDS-XADS and XT-ADS Prototypes*," Nuclear Engineering and Design 239/11, pp. 2573-2580, 2009.
[29] A. ABDERRAHIM and P. D'HONDT, "*MYRRHA: A European Experimental ADS for R&D Applications Status at Mid-2005 and Prospective Towards Implementation*," Journal of Nuclear Science and Technology 44/3, pp. 491-498, 2007.
[30] D. de BRUYN, "*From MYRRHA to XT-ADS: the Design Evolution of an Experimental ADS System*," ACCAPP07, Pocatello, USA, 2007.
[31] F.ROELOFS et al., "*Development of an Euler-Euler Two-Phase Model for Application in the Windowless XT-ADS spallation target design*," ICAPP07, pp. 536-541, 2008.
[32] G. DOMANSKA et al., "*Investigations of the SAD Design Parameters for Optimum Experimental Performance*," Annals of Nuclear Energy 31/18, pp. 2127-2138, 2004.
[33] V. SHVETSOV et al., "*The Subcritical Assembly in Dubna (SAD)-Part I: Coupling All Major Components of an Accelerator Driven System (ADS)*," Nuclear Instruments and Methods in Physics Research A 562/2, pp. 883-886, 2006.
[34] W. GUDOWSKI et al., "*The Subcritical Assembly in Dubna (SAD)-Part II: Research Program for ADS-Demo Experiment*," Nuclear Instruments and Methods in Physics Research A 562/22, pp. 887-891, 2006.
[35] D.L. MOSES et al., "*Plutonium Disposition in the BN-600 Fast Neutron Reactor at the Beloyarsk Nuclear Power Plant*," Nuclear Instruments and Methods in Physics Research A 414, pp. 28-35, 1998.
[36] D. E. BELLER et al., "*The U.S. Accelerator Transmutation of Waste Program*," Nuclear Instruments and Methods for Physics Research A 463/3, pp. 468-486, 2001.
[37] V.V. KULIK, J.C. LEE and D. E. BELLER, "*Dynamic Analysis and Space-Time Effects in the ISU RACE Configuration*," Nuclear Instruments and Methods in Physics Research A 562/2, pp. 838-840, 2006.

[38] D.E. BELLER, "*Update on the Reactor-Accelerator Coupling Experiments (RACE) Project*," Transactions of the American Nuclear Society 95, pp. 943-944, 2006.
[39] D.E. BELLER et al., "*Overview of the U.S. Reactor-Accelerator Coupling Experiments (RACE) Project*," ACCAPP07, pp. 829-835, 2007.
[40] V. BOURNOS et al., "*YALINA-Thermal Benchmark Specifications for the IAEA Coordinated Research Projects on Analytical and Experimental Benchmark Analysis on Accelerator Driven Systems, and Low Enriched Uranium Fuel Utilization in Accelerator Driven Sub-Critical Assembly Systems*," IAEA, 2007.
[41] V. BOURNOS et al., "*YALINA-Booster Benchmark Specifications for the IAEA Coordinated Research Projects on Analytical and Experimental Benchmark Analysis on Accelerator Driven Systems, and Low Enriched Uranium Fuel Utilization in Accelerator Driven Sub-Critical Assembly Systems*," IAEA, 2007.
[42] Y. GOHAR and D.L. SMITH, "*YALINA, Facility, A Sub-Critical Accelerator-Driven System (ADS) for Nuclear Energy Research. Facility Description and Overview of the Research Program*," Argonne National Laboratory, ANL-10/05, 2010.
[43] G. ALIBERTI and Y. GOHAR, "*YALINA Analytical Benchmark Analyses Using the Deterministic ERANOS Code System*," Argonne National Laboratory, ANL-09/23, 2009.
[44] A. TALAMO and Y. GOHAR, "*Monte Carlo Modeling and Analyses of YALINA-Booster Subcritical Assembly, Part I: Analytical Model and Main Neutronics Parameters*," Argonne National Laboratory, ANL-NE-08/13, 2008.
[45] A. TALAMO and Y. GOHAR, "*Monte Carlo Modeling and Analyses of YALINA-Booster Subcritical Assembly, Part II: Pulsed Neutron Source Analyses*," Argonne National Laboratory, ANL-NE-08/33, 2008.
[46] A. TALAMO et al., "*Pulse Superimposition Calculational Methodology for Estimating the Subcriticality Level of Nuclear Fuel Assemblies,*" Nuclear Instruments and Methods in Physics Research A 606 pp. 661-668 2009.
[47] C.E. LEE, "*The Discrete SN Approximation to Transport Theory*," LA-2595, 1962.
[48] G. MARLEAU, A. HEBERT and R. ROY, "*A User Guide For DRAGON 3.05F*," Ecole Polytechnique de Montreal, IGE-174 Revision 6F, 2008.
[49] R.E. ALCOUFFE, R.S. BAKER, F.W. BRINKLEY, D.R. MARR, R.D. O'DELL, and W.F. WALTERS, "*DANTSYS: A Diffusion Accelerated Neutral Particle Code System*," LA-12969-M, 1995.
[50] R.E. ALCOUFFE, R.S.BAKER, J.A. DAHL, S.A.TURNER, and R.C. WARD, "*PARTISN: A Time Dependent, Parallel Neutral Particle Transport Code System*," LA-UR-08-07258, 2008.
[51] J. FAVORITE, "*A Brief User's Guide for PARTISN*," LA-UR-08-6339, 2008.
[52] J.A. DAHL, "*PARTISN results for the OECD/NEA 3-D extension C5G7 MOX benchmark*," Progress in Nuclear Energy 48/5, pp. 401-409, 2006.
[53] R.C. WARD et al., "*A Diffusion Synthetic Acceleration Method for Block Adaptive Mesh Refinement*," Nuclear Science and Engineering 152/2, pp. 164-179, 2006.
[54] W.A. RHOADES and D.B. SIMPSON, "*The TORT Three-Dimensional Discrete Ordinates Neutron/Photon Transport Code*," Oak Ridge National Laboratory, ORNL/TM-13221, 1997.
[55] http://www-nds.iaea.org/wimsd/
[56] N. METROPOLIS and S. ULAM, "*The Monte Carlo Method*," Journal of the American Statistical Association 44/247, pp. 335-341, 1949.
[57] MCNP Team, "*MCNP5.1.40*," LA-UR-05-8617, 2005.
[58] J.S. HENDRICKS et al., "*MCNPX, VERSION 2.6.B*," LA-UR-06-3248, 2006.
[59] ANSWERS SOFTWARE, "*MONK, User Guide for Version 9*," 2006.

[60] E.M. GELBARD, "*Implementations of the Superhistory Method*," Argonne National Laboratory, ANL/RA/CP-102252, 2000.
[61] N. R. SMITH, "*The MONK Superhistory Power Algorithm*," AEA Technology plc, 1996.
[62] M. HERMAN, "*ENDF-VI Formats Manual*," Brookhaven National Laboratory, BNL-NCS-44945—05-Rev, 2005.
[63] http://t2.lanl.gov/data/data.html
[64] B.G. CARLSON and C.E. LEE, "*Mechanical Quadrature and the Transport Equation*," LAMS-2573, 1961.
[65] B.G. CARLSON, "*Tables of Equal Weight Quadrature EQ_n Over the Unit Sphere*," LA-4734, 1971.
[66] R.E. ALCOUFFE, "*PARTISN Calculations of 3D Radiation Transport Benchmarks for Simple Geometries with Void Regions*," Progress in Nuclear Energy 39/2, pp. 181-190, 2001.
[67] R.E. MacFARLANE and D.W. MUIR, "*The NJOY Nuclear Data Processing System, Version 91*," Los Alamos National Laboratory, LA-12740-M, 1994.
[68] http://t2.lanl.gov/codes/njoy99/index.html
[69] http://www-nds.iaea.org/wimsd/downloads.htm
[70] http://www.nea.fr/abs/html/uscd1237.html
[71] P.G. AITKEN and B. JONES, "*Teach Yourself C in 21 Days*," Sams Publishing, 1992.
[72] W.A. RHOADES, "*The GIP Program for Preparation of Group Organized Cross Sections*," Informal Notes, 1978.
[73] J.P. JENAL et al., "*The Generation of a Computer Library for Discrete Ordinates Quadrature Sets*," Oak Ridge National Laboratory, ORNL/TM-6023, 1977.
[74] http://homepage.univie.ac.at/manfred.drosg/drosg2000.htm
[75] M. KARTHIKEYAN and A. HEBERT, "*Performance of Advanced Self-Shielding Models in DRAGON for the Estimation of CANDU-6 Safety Parameters*," Nuclear Technology 157, pp. 299-316, 2007.
[76] M. M. BRETSCHER, "*Perturbation Independent Methods for Calculating Research Reactor Kinetic Parameters*," ANL/RERTR/TM30, 1997.
[77] G. CHIBA, "*Calculation of Effective Delayed Neutron Fraction Using a Modified K-Ratio Method*," Journal of Nuclear Science and Technology 46/5, pp. 399-402, 2009.
[78] 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. 2931-3118, 2006.
[79] http://www.nea.fr/html/dbdata/JEFF/JEFF31/index-JEFF-N_1.html
[80] E.E. LEWIS and W.F. MILLER, "*Computational Methods in Neutron Transport*," American Nuclear Society, 1993.
[81] N. SJÖSTRAND, "*Measurement on a Subcritical Reactor Using a Pulsed Neutron Source*," Arkiv för Fysik 11, 1956.
[82] B.E. SIMMONS and J.S. KING, "*A Pulsed Technique for Reactivity Determination*," Nuclear Science and Engineering 3, pp. 595-608, 1958.
[83] T. GOZANI, "*A Modified Procedure for the Evaluation of Pulsed Source Experiments in Subcritical Reactors*," Nucleonik 4, 1962.
[84] E. GARELIS and L.J. RUSSEL, "*Theory of Pulsed Neutron Source Measurements*," Nuclear Science and Engineering 16, 1963.
[85] D. FIENO et al., "*Determination of Reactivities Using Pulsed Neutron Source Techniques for Highly Subcritical Solutions Reactors*," American Nuclear Society Winter Meeting, Chicago, USA, 1967.

[86] Y. CAO, "*Space-Time Kinetics and Time-Eigenfunctions*," University of Michigan, Ph.D. dissertation, 2008.
[87] J. CETNAR and W. GUDOWSKI, "*Neutron Multiplication in Stationary ADS by Using Monte Carlo Techniques*," AGH contribution to EURATOM project: PDS-XADS, pp. 83-88, 2005.
[88] G.I. BELL and S. GLASSTONE, "*Nuclear Reactor Theory*," Van Nostrand Reinhold Company, 1970.
[89] R.K. MEULEKAMP and S. C. van der MARCK, "*Calculating the Effective Delayed Neutron Fraction with Monte Carlo*," Nuclear Science and Engineering 152, pp. 142-148, 2006.
[90] V.V. KULIK and J.C. LEE, "*Space-Time Correction for Reactivity Determination in Source Driven Subcritical Systems*," Nuclear Science and Engineering 153, pp. 69-89, 2006.
[91] L.N. USSACHOFF, "*Equation for the Importance of Neutrons, Reactor Kinetics and the Theory of Perturbation*," Proceedings of the First U.N. Conference on Peaceful Uses of Atomic Energy, Geneva, Switzerland, 1955.
[92] J. LEWINS, "*Importance, the Adjoint Function*," Pergamon Press, 1965.
[93] KEEPIN et al., "*Delayed Neutrons from Fissionable Isotopes of Uranium, Plutonium and Thorium*," Physical Review 107/4, pp. 1044-1049, 1957.
[94] J.J. DUDERSTADT and L.J. HAMILTON, "*Nuclear Reactor Analysis*," John Wiley & Sons, pp. 216-218, 1976.
[95] K.M. CASE, F de HOFFMAN, and G. PLACZEK, "*Introduction to the Theory of Neutron Diffusion*," Los Alamos, 1953.
[96] A. GANDINI, "*HGPT Based Sensitivity Time-Dependent Methods for the Analysis of Subcritical Systems*," Annals of Nuclear Energy 28, pp. 1193-1217, 2001.
[97] G.G.M. COPPA et al., "*Three-Dimensional Neutron Analysis of Accelerator Driven Systems*," International Conference on Mathematics and Computation, Reactor Physics, and Environmental Analysis in Nuclear Applications, Madrid, Spain, 1999.
[98] M.G. KENDALL and A. STUART, "*The Advanced Theory of Statistics*," Vol. I. Second Edition. Charles Griffin & Company, p. 232, 1963.
[99] J. F. BRIESMEISTER, "*General Monte Carlo N-Particle Transport Code*," Version 4c, LANL, LA-13709-M, 2002.
[100] http://t2.lanl.gov/cgi-bin/endf?0,0,/inet/WWW/data/data/ENDFB-VII-neutron/U/235
[101] J. FAVORITE, "*A Four-Group Analytical Rod Transport Problem for (n,γ) and (n,n'γ) Verification*," Progress in Nuclear Energy 48, pp. 103-134, 2006.
[102] N. A. HANAN, S.C. MO, and J. E. MATOS, "*Transient Analyses for HEU and LEU Designs of the FRM-II*," International Meeting on Reduced Enrichment for Research and Test Reactors, São Paulo, Brazil, 1998.
[103] G.D. SPRIGGS et al., "*On the Definition of Neutron Lifetimes in Multiplying and Non-Multiplying Systems*," Los Alamos National Laboratory, LA-UR-97-1073, 1997.
[104] L.J. TEMPLIN, "*Reactor Physics Constants*," Second Edition. Argonne National Laboratory, ANL-5800, p. 444, 1963.
[105] M. IQBAL et al., "*Flow of Kinetic Parameters in a Typical Swimming Pool Type Research Reactor*," Annals of Nuclear Energy 35, pp. 518-524, 2008.
[106] J. FAVORITE, "*Flux Moments and Inner Products*," Los Alamos National Laboratory, X-1-TA, MS P365, 2008.
[107] R.D. BUSH et al., "*Definition of Neutron Lifespan and Lifetime in MCNP4b*," Los Alamos National Laboratory, LA-UR-97-222, 1997.

[108] A. TALAMO et al., "*Deterministic and Monte Carlo Analyses of YALINA Subcritical Assemblies*," IAEA Technical Meeting on LEU Fuel Utilization in ADS, Vienna, Austria, 2009.
[109] M. DALHFORS and Y. KADI. EA-MC Neutronic Calculations on IAEA ADS Benchmark 3.2. Uppsala University, TSL/ISV-2006-0296, 2006.
[110] S. DULLA et al., "*Kinetic Parameters for Source Driven Systems*," PHYSOR Proceedings, Vancouver, Canada, 2006.
[111] S.M. LEE and J.R. MAIORINO, "*Study of the Kinetics Parameters for Subcritical Media Driven by Source*," INAC Proceedings, Rio de Janeiro, Brazil, 2009.

Table I: Neutron multiplication factor (k_{eff}); the standard deviation is reported in pcm units

COMPUTER PROGRAM AND NUCLEAR DATA FILES		CONFIGURATION		
		216	**245**	**280**
MCNPX2.6f	ENDFB-VII	0.87933± 9	0.91940± 9	0.95869± 8
MCNPX2.7b	ENDFB-VI.6	0.87914± 2	0.91916± 2	0.95836± 3
MCNPX2.6b	ENDFB-VI.6	0.87902± 9	0.91911± 4	0.95833± 7
MCNPX2.6f	ENDFB-VI.2 (C_{free})	0.87632± 9	0.91744± 9	0.95669± 9
MONK9a1	ENDFB-VI.2; DICE (C_{free})	0.87530±20	0.91520±20	0.95450±20
MCNPX2.7a	ENDFB-VI.6; EC5-7 with graphite	0.87929± 8	-	-
PARTISN5.86	ENDFB-VI.8	0.88004	0.91905	0.95830
PARTISN5.86	ENDFB-VI.8; critical adjoint	0.87943	0.91841	0.95774
TORT3.1	ENDFB-VI.8	0.87925	0.91821	0.95727
MCNPX2.6f	JEFF-3.1	0.87987±10	0.91991± 8	0.95900± 8
MONK9a1	JEFF-3.1; BINGO	0.87950±20	0.91960±20	0.95860±20
EXPERIMENT DATE AND EXPERIMENTAL CHANNEL		**216**	**245**	**280**
July 2007	EC2 – Area Method	-	0.92280	0.95040
July 2007	EC5 – Area Method	-	0.91550	0.94790

Table II: Neutron source multiplication factor for the Californium neutron source with the beam tube; the standard deviation is reported in pcm units

COMPUTER PROGRAM AND NUCLEAR DATA FILES		CONFIGURATION		
		216	**245**	**280**
MCNPX2.6b	ENDFB-VI.6 – Equation 2	0.90886	0.94009	0.96947
MCNPX2.6b	ENDFB-VI.6 – Equation 1	0.90884	0.94008	0.96946
MONK9a1	ENDFB-VI.2; DICE (C_{free}) – Code Output	0.90820±30	0.93810±30	0.96700±20
PARTISN5.86	ENDFB-VI.8 – Equation 1	0.91103	0.94091	0.96987
TORT3.1	ENDFB-VI.8 – Equation 1	0.91120	0.94105	0.96997

Table III: Neutron source multiplication factor for the Californium neutron source without the beam tube; the standard deviation is reported in pcm units

COMPUTER PROGRAM AND NUCLEAR DATA FILES		CONFIGURATION		
		216	**245**	**280**
MCNPX2.6b	ENDFB-VI.6 – Equation 2	0.91649	0.94773	0.97701
MCNPX2.6f	ENDFB-VI.6 – Equation 1	0.91647	0.94772	0.97700
MONK9a1	ENDFB-VI.2; DICE (C_{free}) – Code Output	0.91460±30	0.94590±20	0.97460±20
PARTISN5.86	ENDFB-VI.8 – Equation 1	-	0.94843	0.97707

Table IV: Neutron source multiplication factor for the D-D neutron source; $(E, \vec{\Omega})$ indicates that the energy-angle distribution (Figure 37) of the neutron source is modeled; the standard deviation is reported in pcm units

COMPUTER PROGRAM AND NUCLEAR DATA FILES		CONFIGURATION		
		216	**245**	**280**
MCNPX2.6b	ENDFB-VI.6 – Equation 2	0.90488	0.93779	0.96848
MCNPX2.6b	ENDFB-VI.6 – Equation 2. $(E, \vec{\Omega})$ Source	0.90422	-	-
MCNPX2.6b	ENDFB-VI.6 – Equation 1	0.90486	0.93777	0.96847
MCNPX2.6b	ENDFB-VI.6 – Equation 1. $(E, \vec{\Omega})$ Source	0.90419	-	-
MONK9a1	ENDFB-VI.2; DICE (C_{free}) – Code Output	0.90240±30	0.93490±20	0.96560±20
PARTISN5.86	ENDFB-VI.8 – Equation 1	0.90246	0.93566	0.96742
TORT3.1	ENDFB-VI.8 – Equation 1	0.90263	0.93579	0.96752

Table V: Neutron source multiplication factor for the D-T neutron source; $(E, \vec{\Omega})$ indicates that the energy-angle distribution (Figure 37) of the neutron source is modeled; the standard deviation is reported in pcm units

COMPUTER PROGRAM AND NUCLEAR DATA FILES		CONFIGURATION		
		216	**245**	**280**
MCNPX2.6b	ENDFB-VI.6 – Equation 2	0.86608	0.91129	0.95505
MCNPX2.6b	ENDFB-VI.6 – Equation 2. $(E, \vec{\Omega})$ Source	0.86652	-	-
MCNPX2.6b	ENDFB-VI.6 – Equation 1	0.86209	0.90954	0.95459
MCNPX2.6b	ENDFB-VI.6 – Equation 1. $(E, \vec{\Omega})$ Source	0.86337	-	-
MONK9a1	ENDFB-VI.2; DICE (C_{free}) – Code Output	0.85990±30	0.90680±20	0.95030±20

Table VI: Delayed neutron fraction [pcm] as a function of the number of the fuel rods calculated with MCNP - ENDF/B-VI.6 and PARTISN – ENDF/B-VI.8

β ADJOINT-WEIGHTED (EFFECTIVE)		**CONFIGURATION**		
COMPUTER PROGRAM AND NUCLEAR DATA FILES		**216**	**245**	**280**
MCNPX2.7b	K-Ratio Equation 9	792±3.3	784±3.1	770±4.5
MCNP4c3	Meulekamp and van der Marck Equation 5	785±1.6	774±1.5	767±1.5
PARTISN5.86	Fission Source Adjoint Equation 11	783	774	764
PARTISN5.86	Cf Source Adjoint Equation 11	782	773	764
PARTISN5.86	DD Source Adjoint Equation 11	782	773	764
PARTISN5.86	K-Ratio Equation 9	808	786	775
β ADJOINT-UNWEIGHTED (AVERAGE)		**CONFIGURATION**		
COMPUTER PROGRAM AND NUCLEAR DATA FILES		**216**	**245**	**280**
MCNPX2.7a	Fission[AB] Tally Equation 12	693	687	680
MCNPX2.7a	K-Ratio[A] Equation 9	691±3.2	694±3.1	693±4.4
PARTISN5.86	Adjoint = 1 Equation 12	673	673	673
PARTISN5.86	K-Ratio[A] Equation 9	685	686	686

[A] Delayed neutrons have the same energy spectrum as prompt neutrons
[B] The statistical error is negligible

Table VII: Prompt neutron lifetime [μs] as a function of the number of the fuel rods

ℓ_p ADJOINT-WEIGHTED (EFFECTIVE)		**CONFIGURATION**		
COMPUTER PROGRAM AND NUCLEAR DATA FILES		**216**	**245**	**280**
MCNPX2.6b	ENDF/B-VI.6 – Equation 16	74.3±1.6	78.8±1.2	81.5±1.3
PARTISN5.86	ENDF/B-VI.8 – Fission Adjoint Equation 17	73.4	75.3	80.3
PARTISN5.86	ENDF/B-VI.8 – Cf Source Adjoint Equation 17	72.4	74.5	79.8
PARTISN5.86	ENDF/B-VI.8 – DD Source Adjoint Equation 17	73.0	74.9	80.1
ℓ_p ADJOINT-UNWEIGHTED (AVERAGE)		**CONFIGURATION**		
COMPUTER PROGRAM AND NUCLEAR DATA FILES		**216**	**245**	**280**
MCNPX2.6b	ENDFB-VI.6 – Equation 18 Adjoint=1	259.8	255.1	242.2
PARTISN5.86	ENDFB-VI.8 – Equation 18 Adjoint=1	240.5	237.1	227.8

Table VIII: Correction factors calculated by MCNP/MCNPX for the 245 and 280 configurations for ^{3}He detector, and D-D neutron source. All the modeled detectors have 0.45 cm radius and 25 cm length. For the two detectors case, the detectors are located in EC2 and EC3 or at EC5 and EC6 experimental channels

METHOD AND NUMBER OF MODELED DETECTORS		CONFIGURATION			
		245		280	
		EC2	EC5	EC2	EC5
Kinetic[A]	0	0.991	1.051	0.991	1.054
Kinetic[B]	1	0.993	1.065	0.986	1.046
Static[C]	2	0.973±0.007	1.059±0.009	0.967±0.005	1.039±0.006
A = 10 million neutron histories; B = 30 million neutron histories; C = 100 million neutron histories					

Table IX: Correction factor calculated by PARTISN for the 245 and 280 configurations for ^{3}He detector, and the D-D neutron source with static (source mode) simulations without detector modeling

245 CONFIGURATION				
EC1	EC2	EC3	EC5	EC6
0.940	0.992	1.046	1.055	1.055
280 CONFIGURATION				
EC1	EC2	EC3	EC5	EC6
0.960	0.996	1.038	1.046	1.048

Table X: Neutron multiplication factor (k_{eff}) obtained by MCNP/MCNPX (with ENDF/B-VI.6 nuclear data files) and EA-MC[109] (with JENDL-3.2 nuclear data library) for the 245 and 280 configurations. All the modeled detectors have 0.45 cm radius and 25 cm length. In the case two detectors are used the detectors are located at EC2 and EC3 or at EC5 and EC6 experimental channels. The standard deviation is reported in pcm

CODE AND NUMBER OF MODELED DETECTORS		CONFIGURATION			
		245		280	
		EC2	EC5	EC2	EC5
MCNPX	0	0.91911±4	0.91911±4	0.95833±7	0.95833±7
MCNPX	1	0.91578±7	0.91881±8	0.95524±8	0.95801±8
MCNPX	2	0.91435±8	0.91874±9	0.95341±7	0.95767±8
EA-MC	0	0.92184±2	0.92184±2	0.95797±2	0.95797±2

Table XI: Neutron multiplication factor (k_{eff}) obtained by pulsed neutron source experiments using the area method and two detectors. The detectors are located in the experimental channels EC2 and EC3 or EC5 and EC6. The detector acquiring the signal (EC2 or EC5) has 0.45 cm radius and 25 cm height. The error is reported in pcm units

REACTIVITY AND NUMBER OF MODELED DETECTORS		CONFIGURATION			
		245		280	
		EC2	EC5	EC2	EC5
ρ_{exp}	No correction	0.92402±410	0.91647±340	0.95072±180	0.94857±230
$c \cdot \rho_{exp}$	Kinetic 0	0.92464	0.91257	0.95112	0.94596
$c \cdot \rho_{exp}$	Kinetic 1	0.92449	0.91152	0.95136	0.94635
$c \cdot \rho_{exp}$	Static 2	0.92593±404	0.91198±363	0.95228±176	0.94667±239

Table XII: EK10 fuel composition (weight fractions)

Isotope	Without ^{234}U IAEA Benchmark Specifications	With ^{234}U Mass	With ^{234}U and 2.1% Correction
^{234}U	0.000000	6.993270E-04	6.846411E-04
^{235}U	0.079691	7.917936E-02	7.751659E-02
^{238}U	0.728557	7.280830E-01	7.127933E-01
^{16}O	0.142022	1.086529E-01	1.063712E-01
^{12}Mg	0.049730	8.311200E-02	1.023666E-01
^{10}B	0.000000	3.209660E-08	3.142257E-08
^{11}B	0.000000	1.421121E-07	1.391278E-07
C	0.000000	9.168880E-05	8.976334E-05
^{14}N	0.000000	9.132934E-06	8.941143E-06
^{15}N	0.000000	3.594571E-08	3.519085E-08
^{28}Si	0.000000	5.896629E-05	5.772800E-05
^{29}Si	0.000000	3.092349E-06	3.027409E-06
^{30}Si	0.000000	2.123521E-06	2.078927E-06
^{55}Mn	0.000000	4.309374E-06	4.218877E-06
^{54}Fe	0.000000	5.290215E-06	5.179121E-06
^{56}Fe	0.000000	8.675683E-05	8.493493E-05
^{57}Fe	0.000000	2.118113E-06	2.073633E-06
^{58}Fe	0.000000	2.743075E-07	2.685470E-07
^{58}Ni	0.000000	3.881562E-06	3.800049E-06
^{60}Ni	0.000000	1.546720E-06	1.514239E-06
^{61}Ni	0.000000	6.836170E-08	6.692610E-08
^{62}Ni	0.000000	2.214903E-07	2.168390E-07
^{64}Ni	0.000000	5.825982E-08	5.703636E-08
^{63}Cu	0.000000	2.511515E-06	2.458773E-06
^{65}Cu	0.000000	1.156037E-06	1.131760E-06

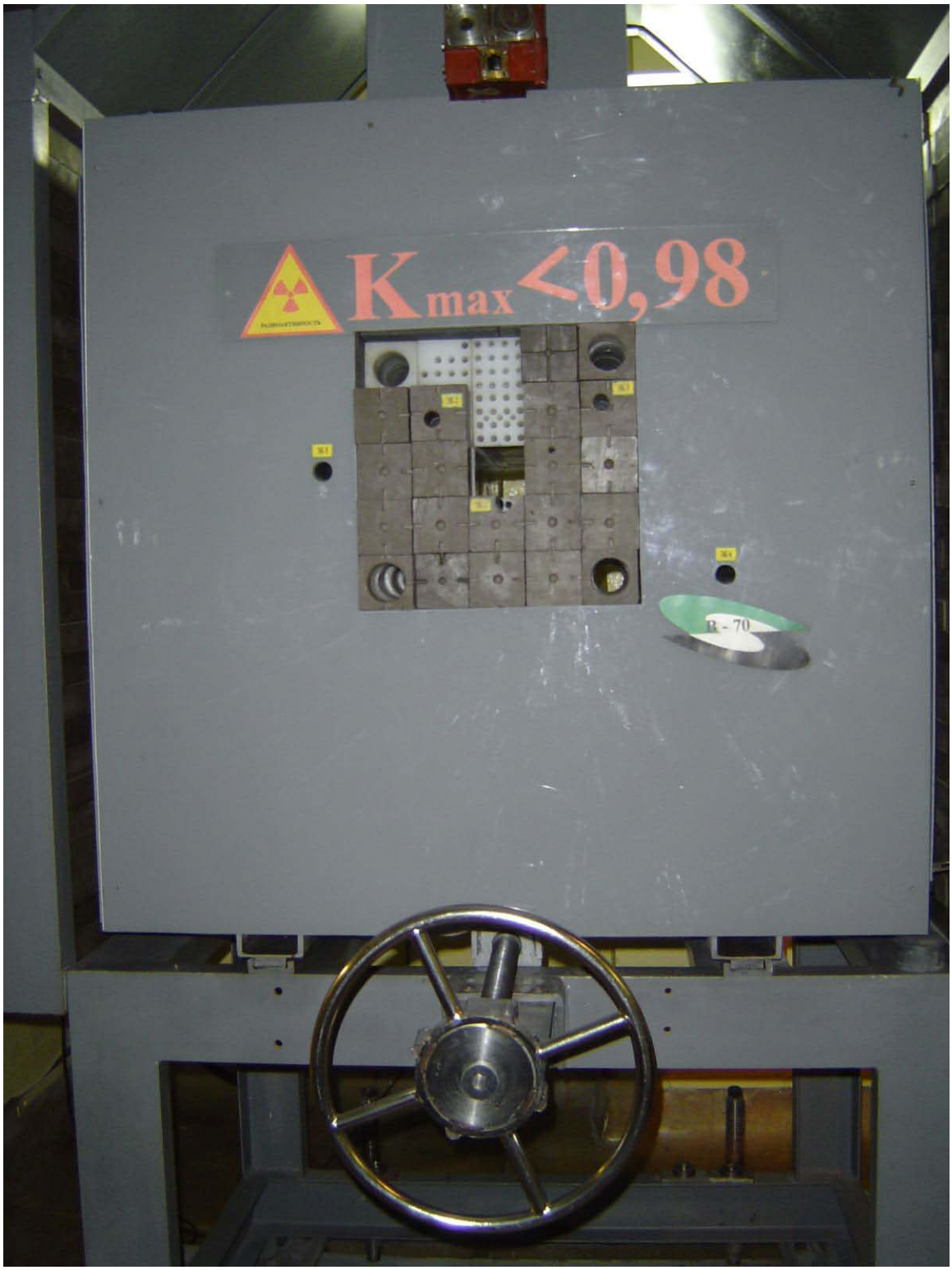

Figure 1. YALINA thermal facility

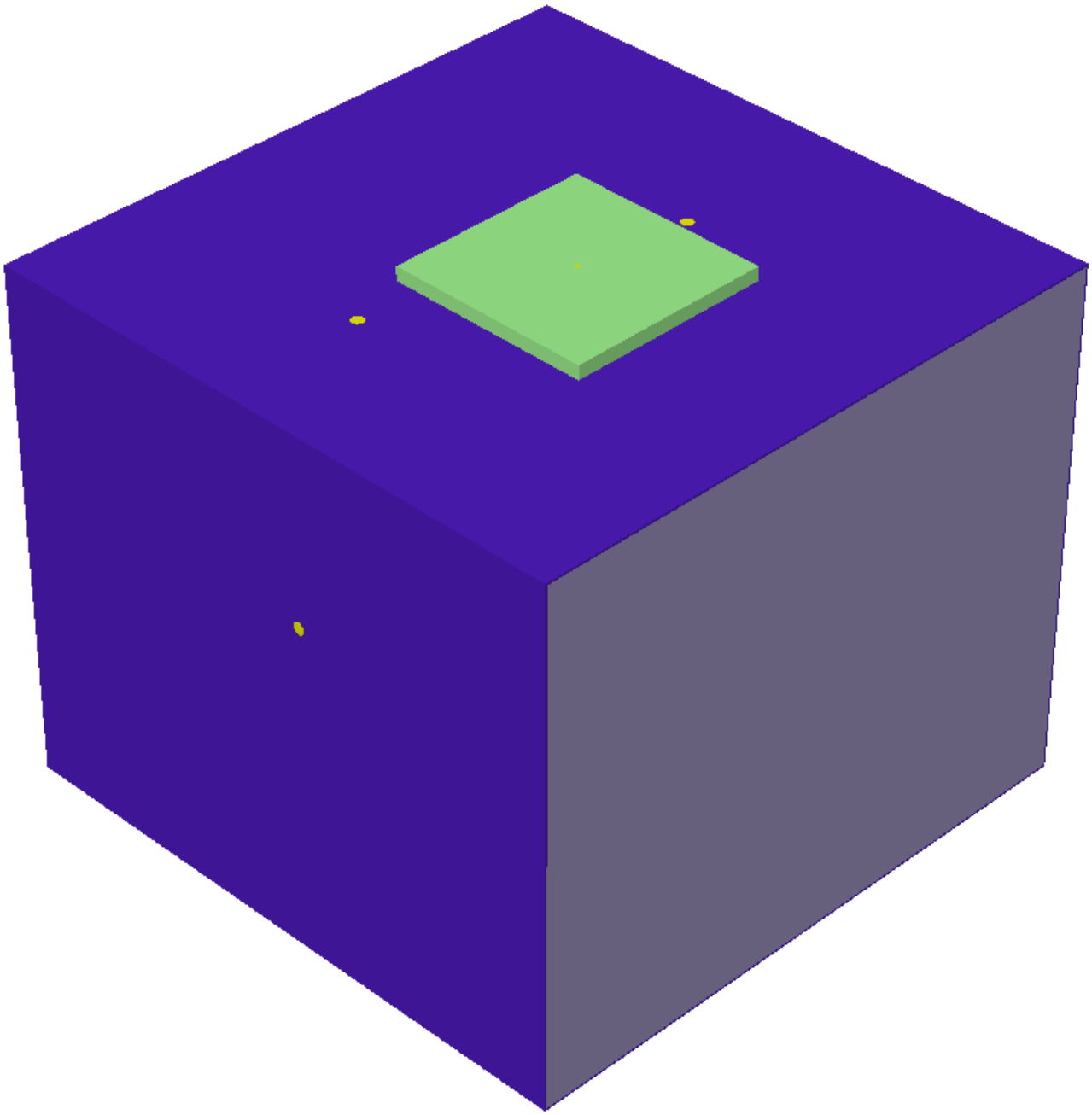

Figure 2. External view of the YALINA thermal subcritical assembly model

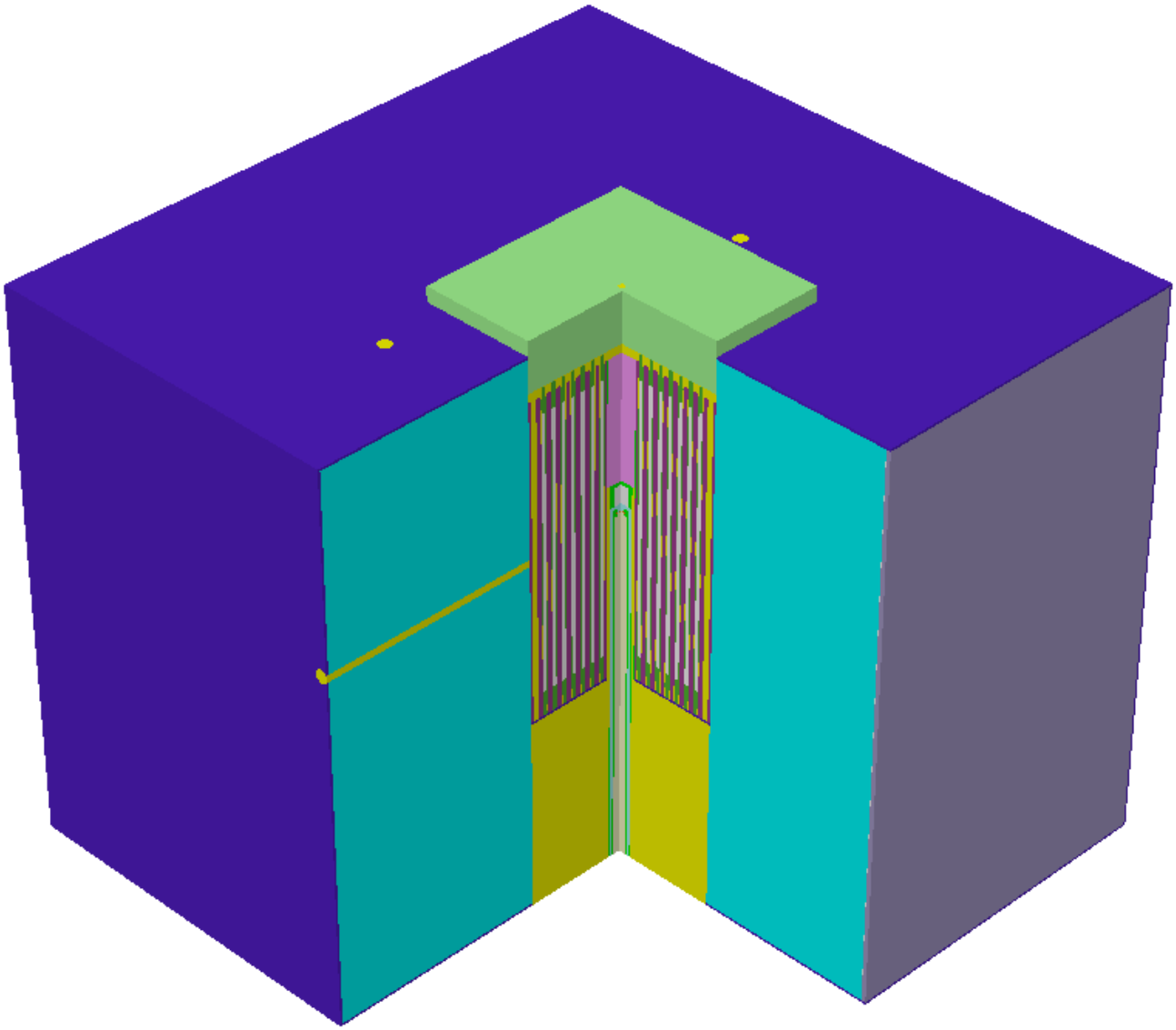

Figure 3. External view of the YALINA thermal subcritical assembly model with corner cut

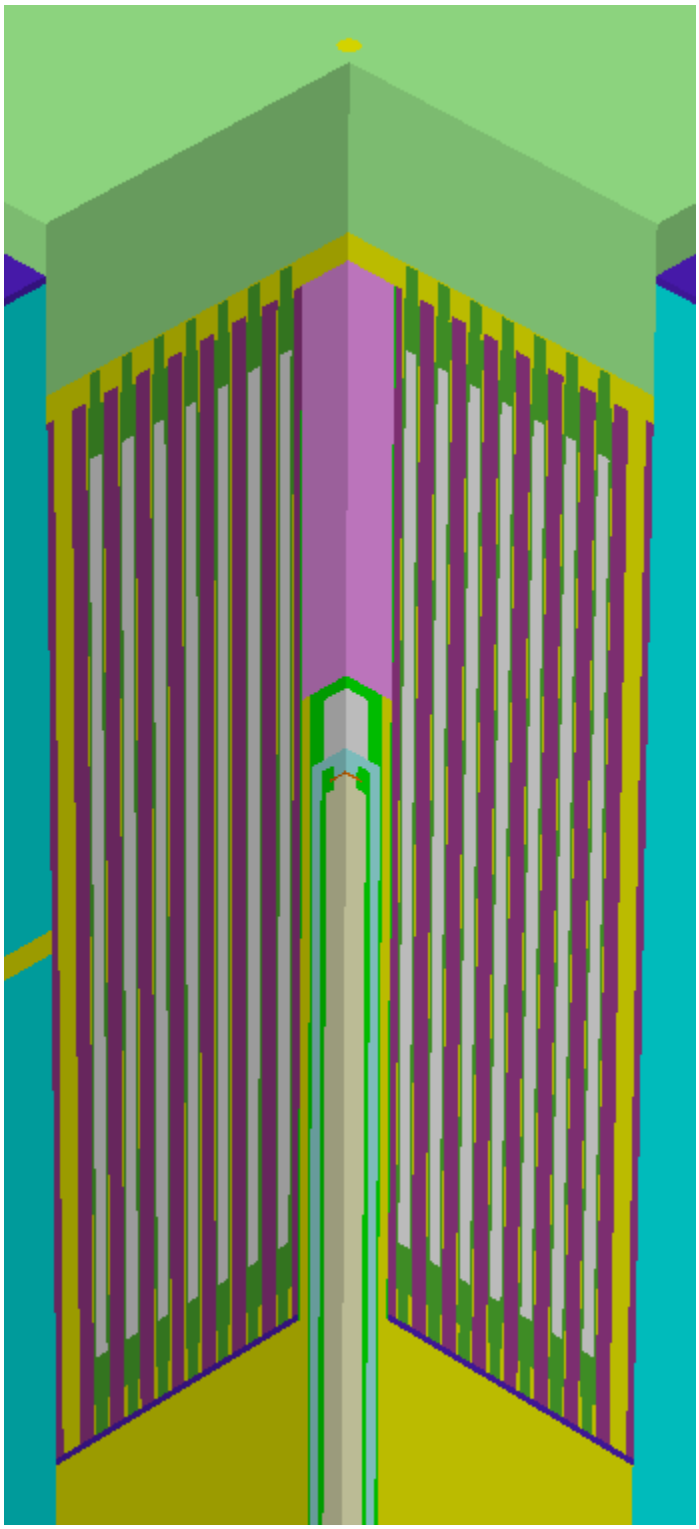

Figure 4. Zoom of the external view of the YALINA thermal subcritical assembly model with corner cut

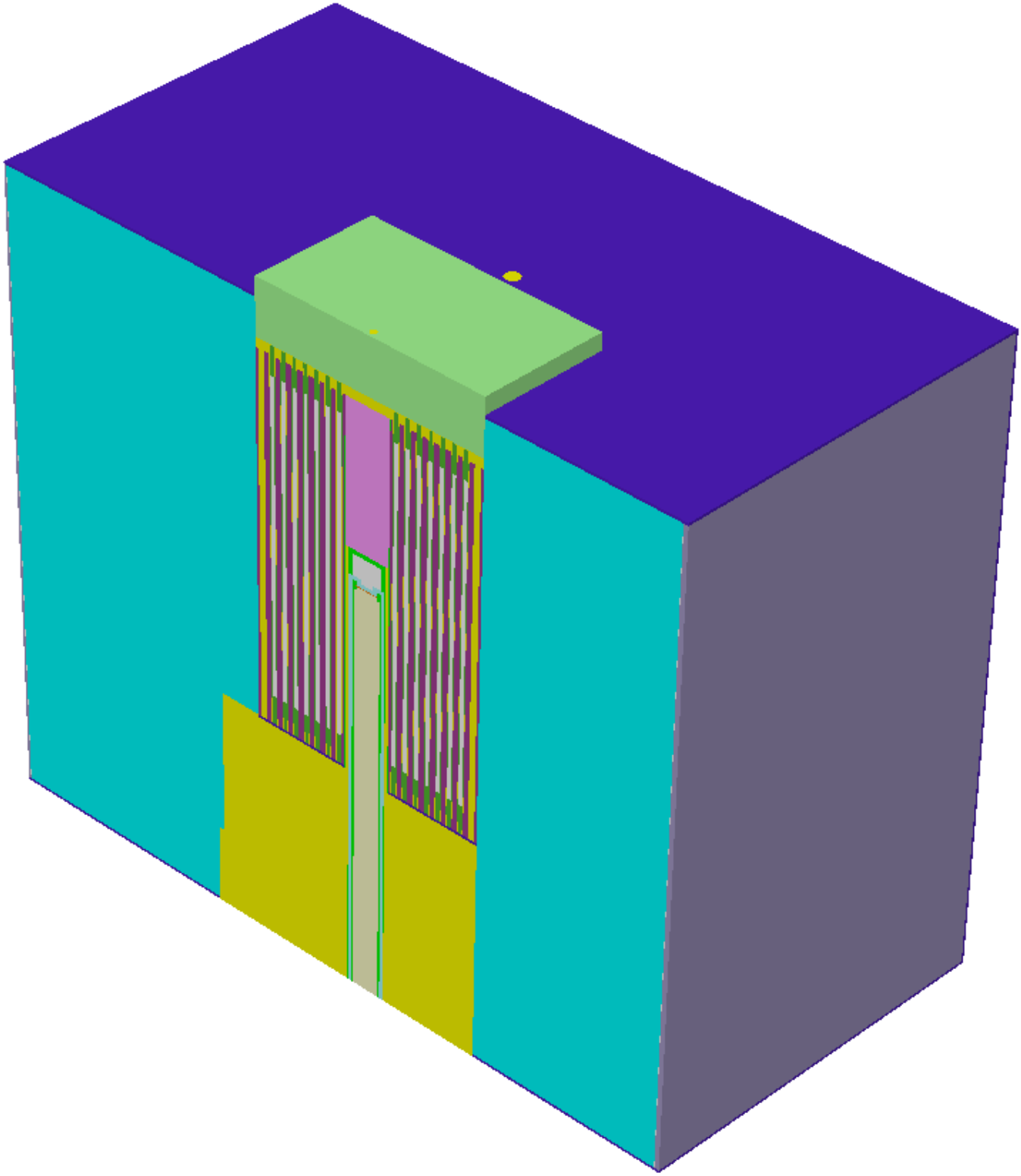

Figure 5. External view of the YALINA thermal subcritical assembly model with left half-side cut

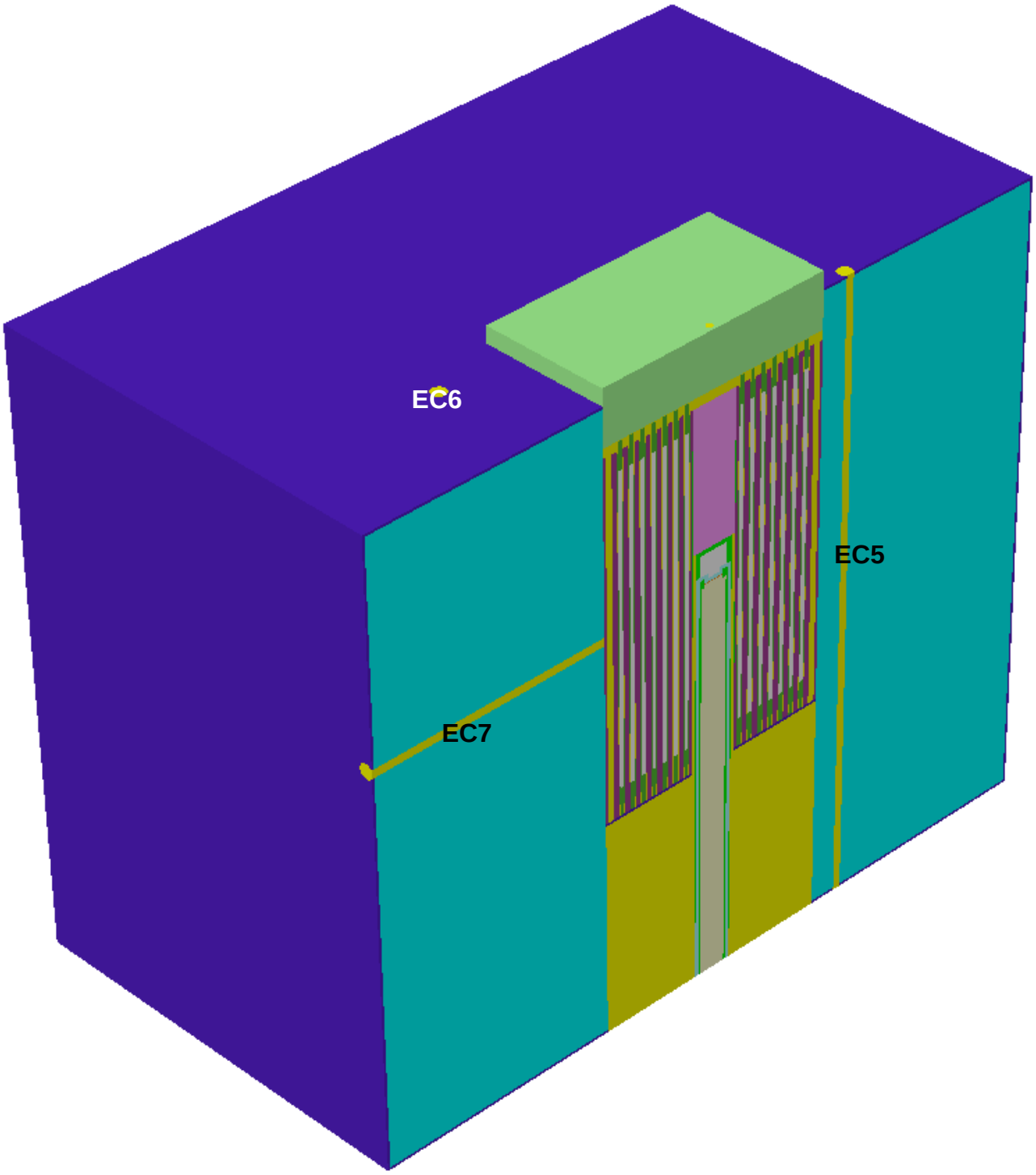

Figure 6. External view of the YALINA thermal subcritical assembly model with right half-side cut

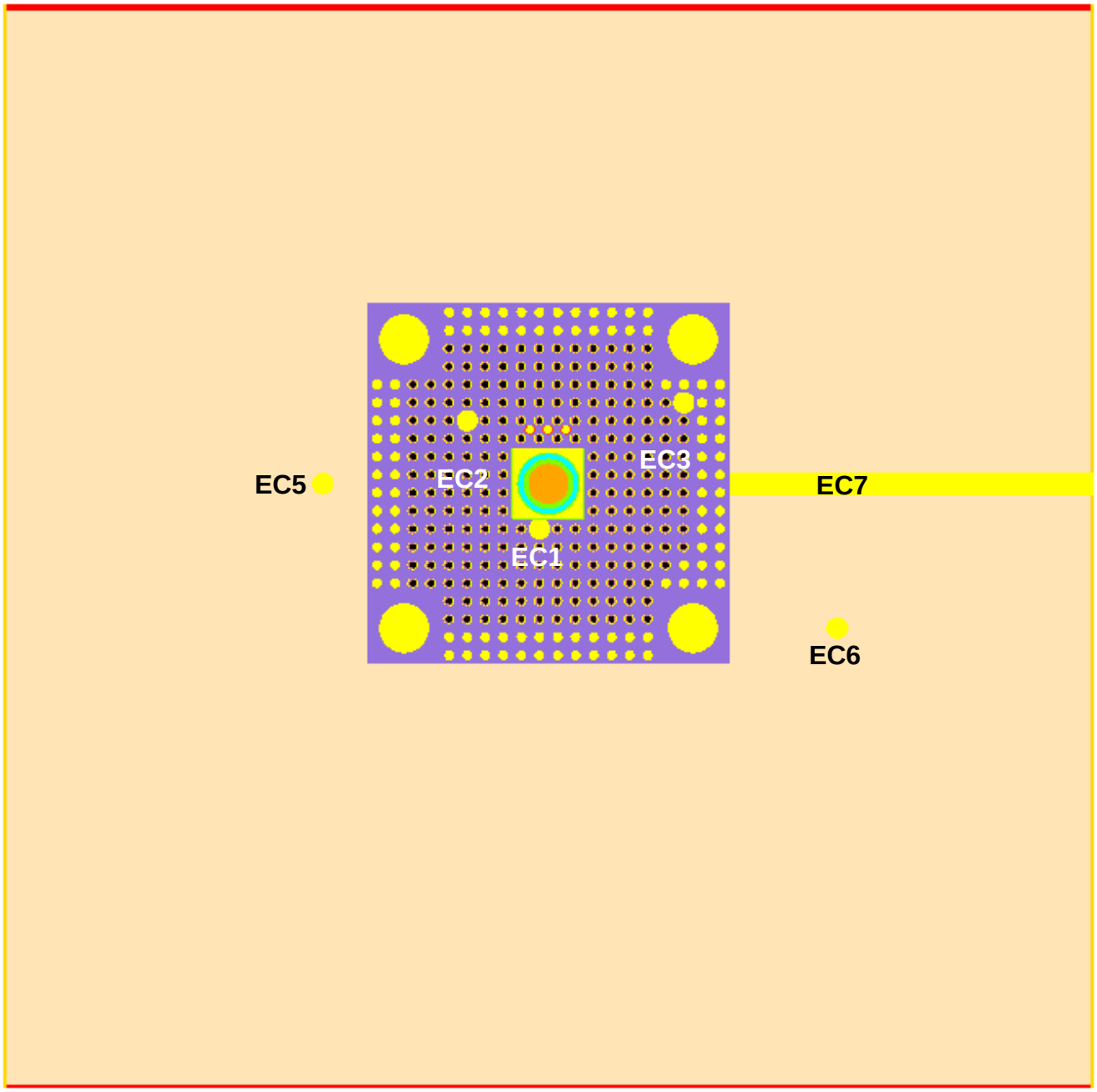

Figure 7. Horizontal section of the YALINA thermal subcritical assembly model at the middle of the active fuel height

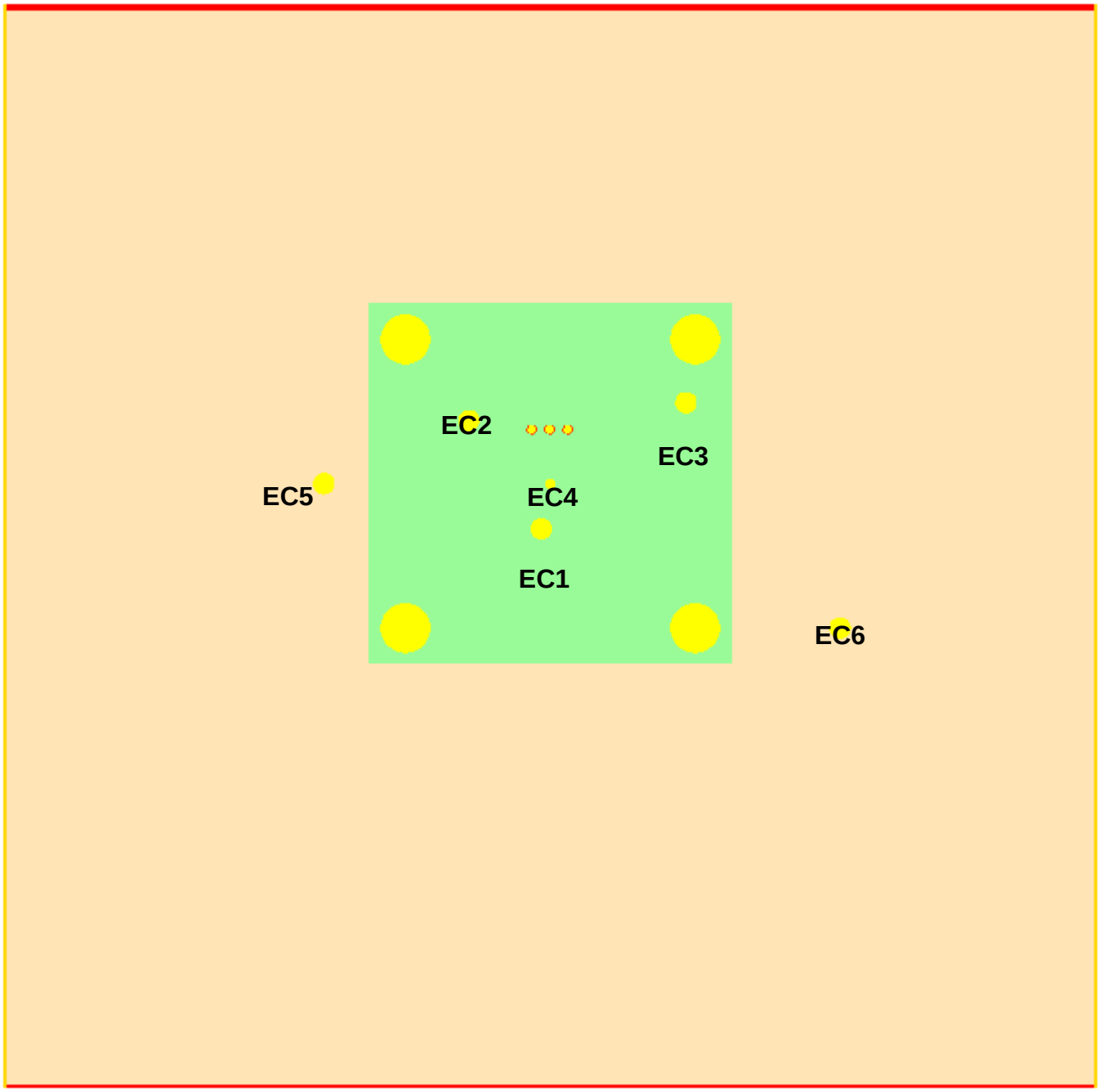

Figure 8. Horizontal section of the YALINA thermal subcritical assembly model at the height of the borated polyethylene axial reflector

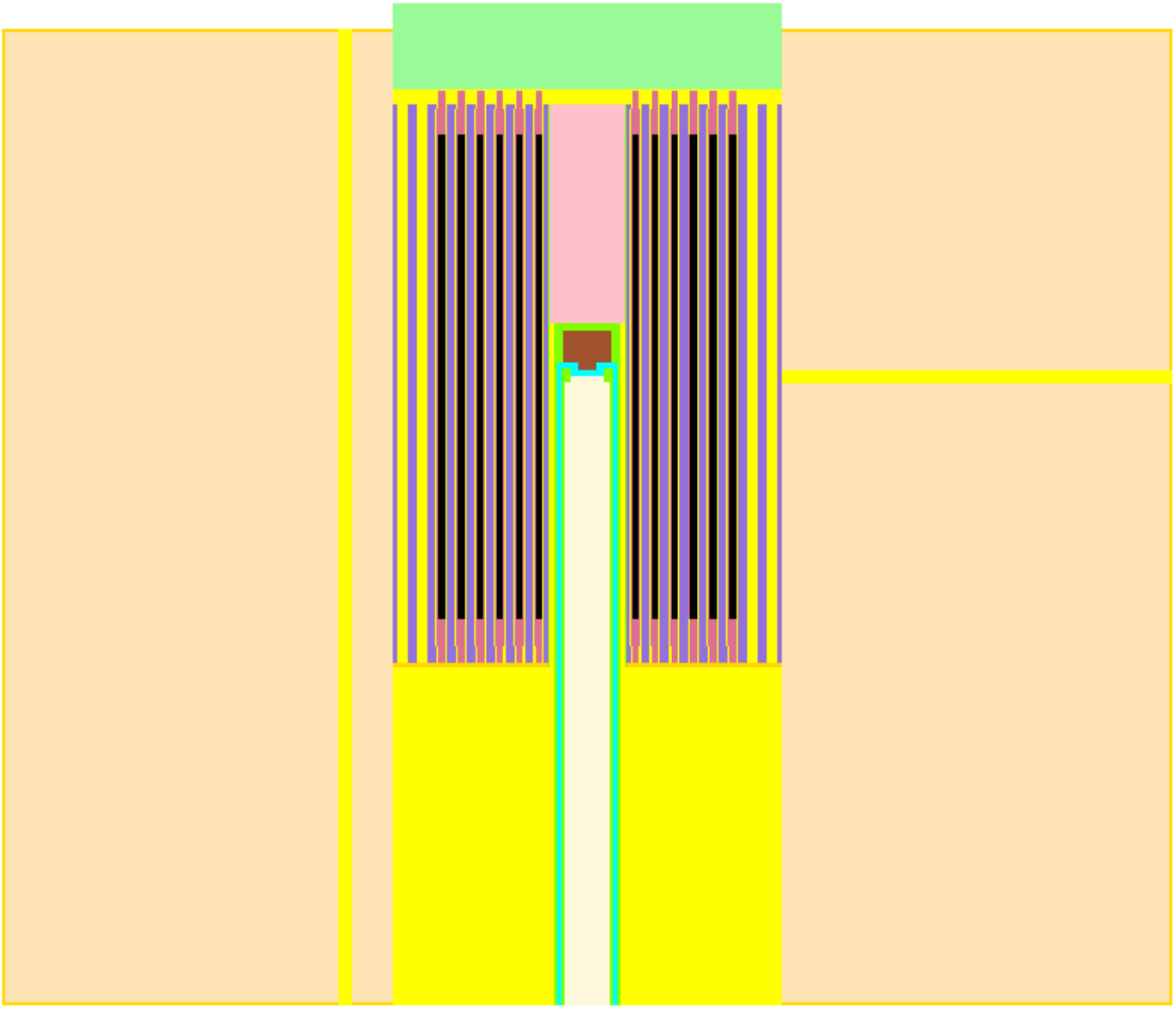

Figure 9. Vertical section of the YALINA thermal subcritical assembly model

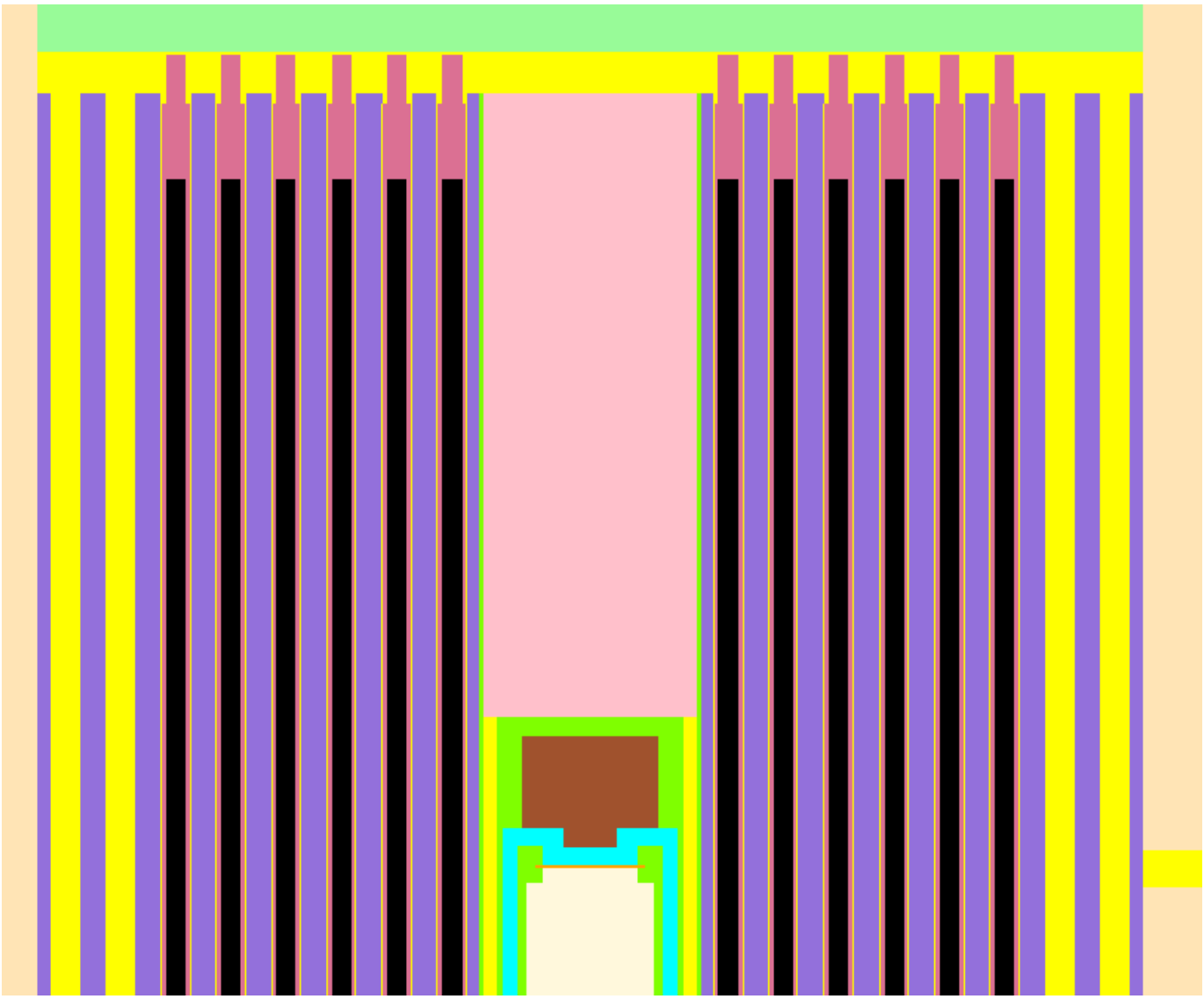

Figure 10. Zoom of the vertical section of the YALINA thermal subcritical assembly model

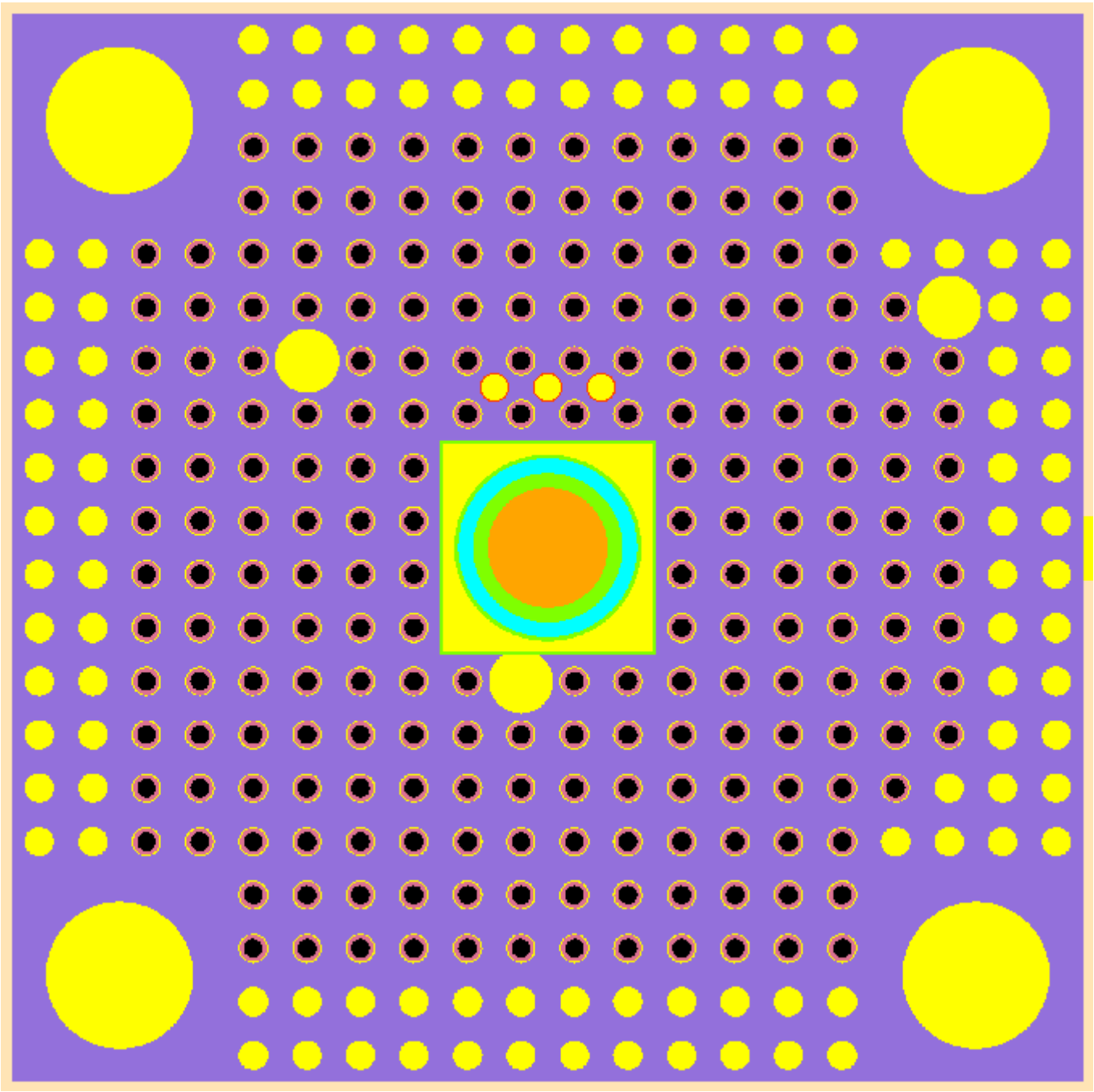

Figure 11. Horizontal section of the YALINA thermal subcritical assembly model with 216 fuel rods

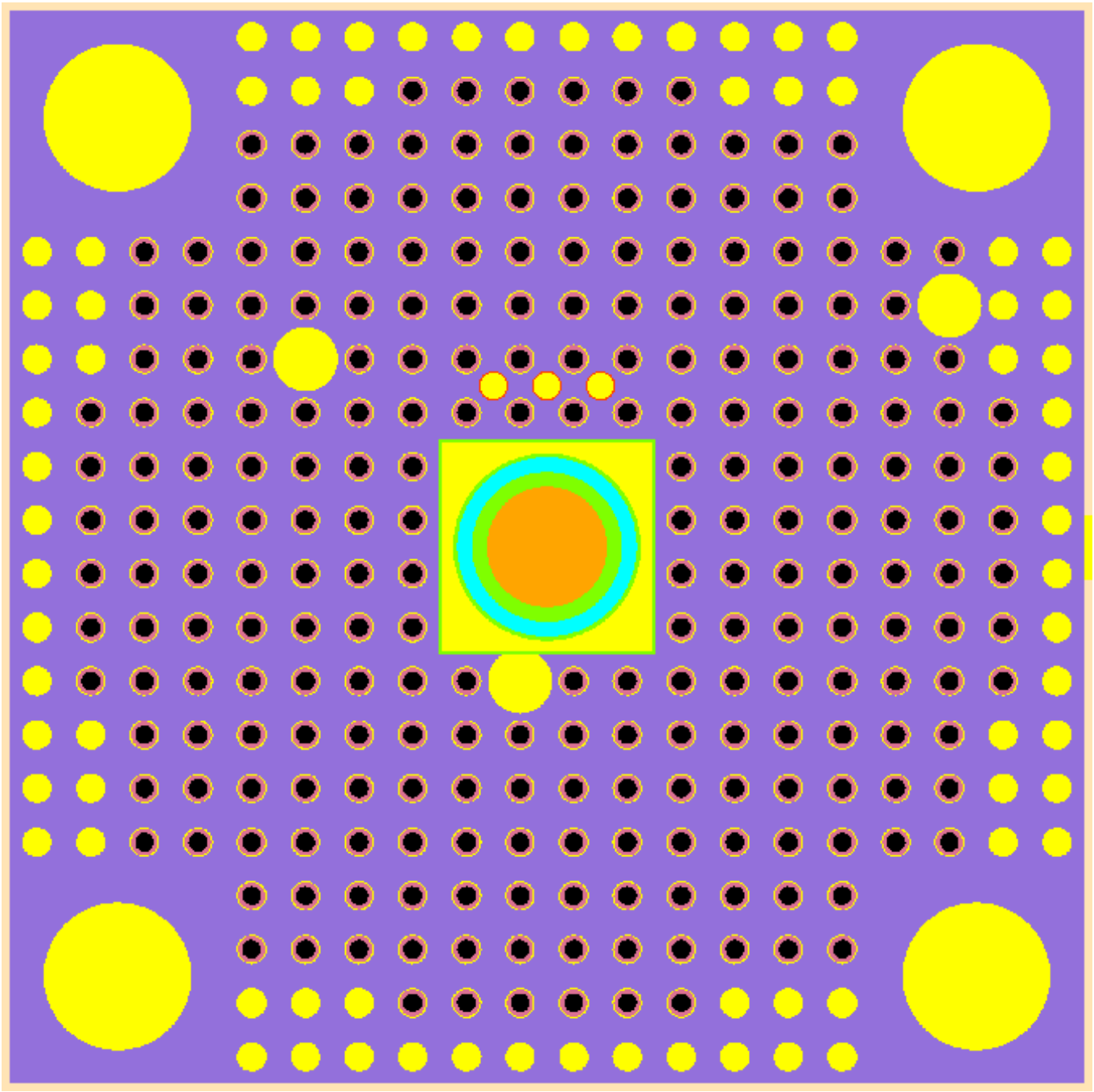

Figure 12. Horizontal section of the YALINA thermal subcritical assembly model with 245 fuel rods

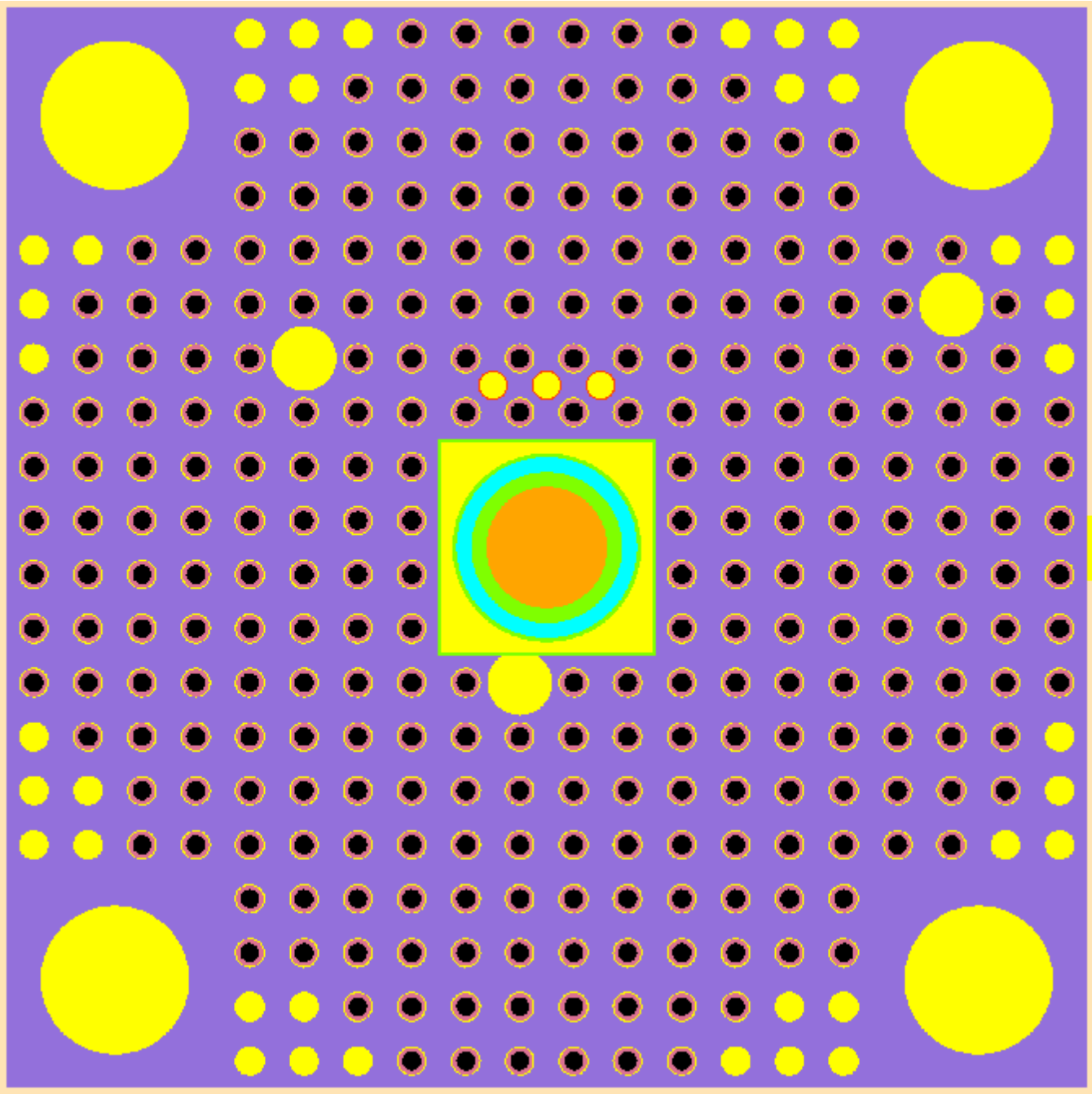

Figure 13. Horizontal section of the YALINA thermal subcritical assembly model with 280 fuel rods

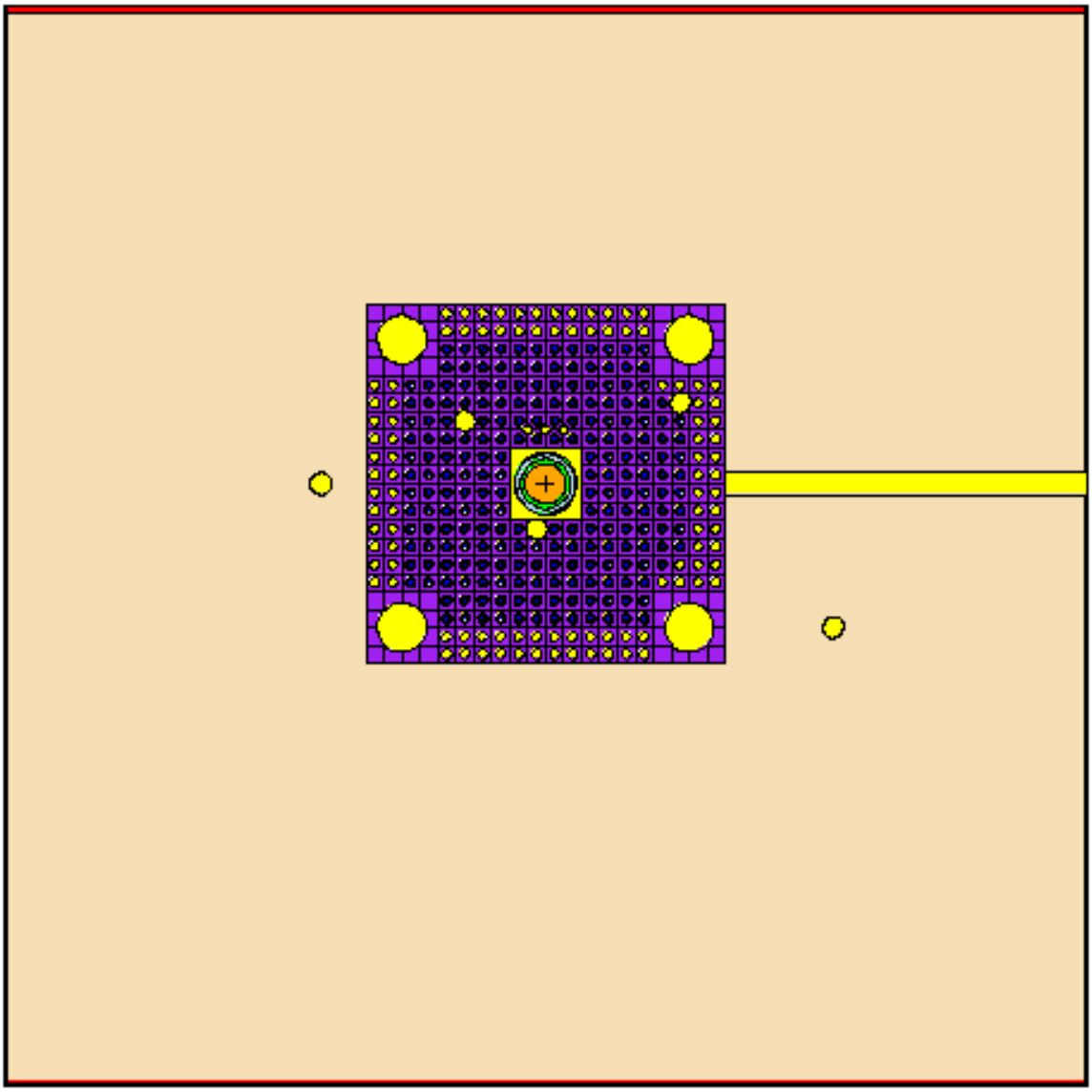

Figure 14. Horizontal section of the YALINA thermal subcritical assembly MCNP/MCNPX model

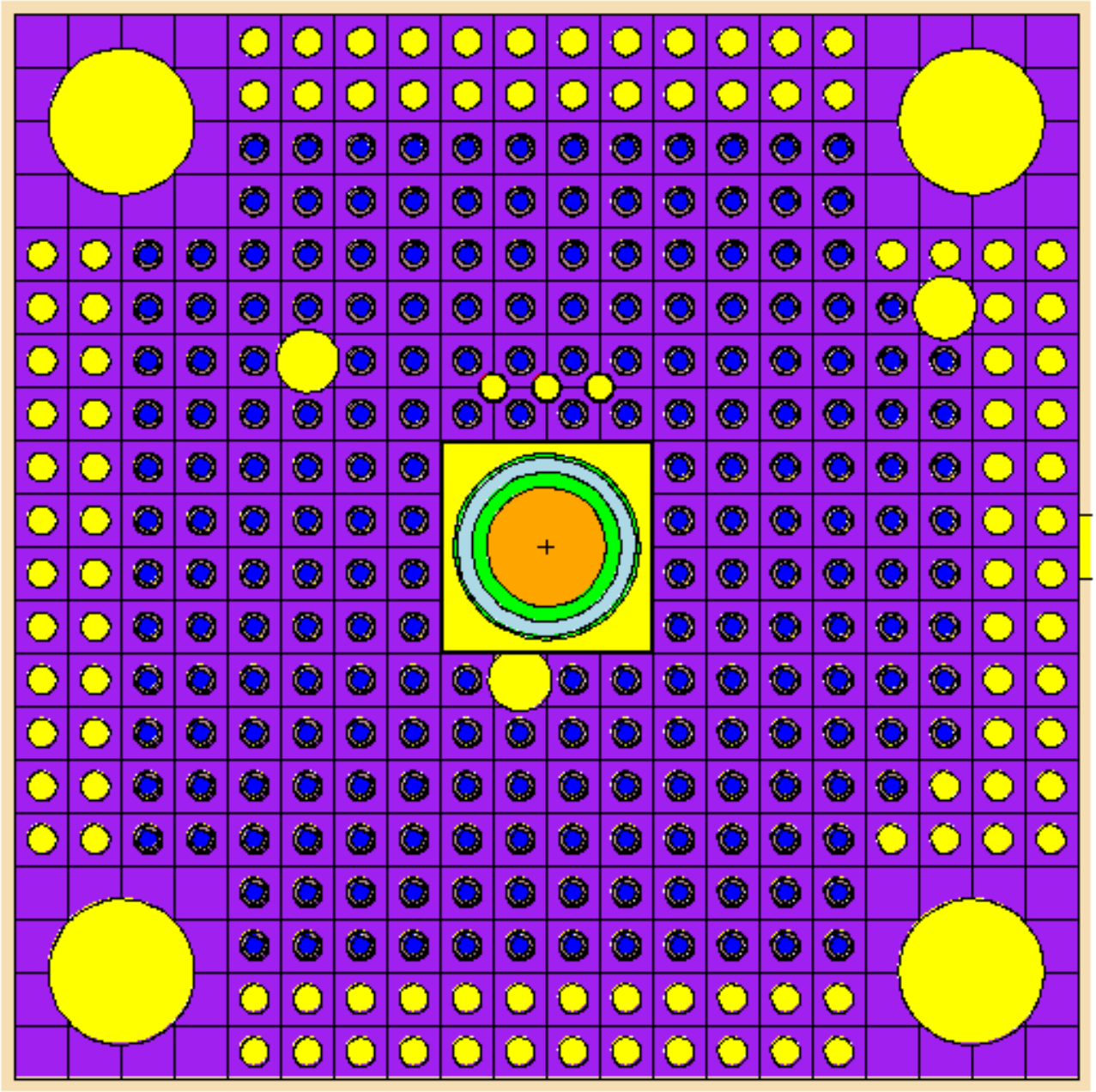

Figure 15. Zoom of the horizontal section of the YALINA thermal subcritical assembly MCNP/MCNPX model

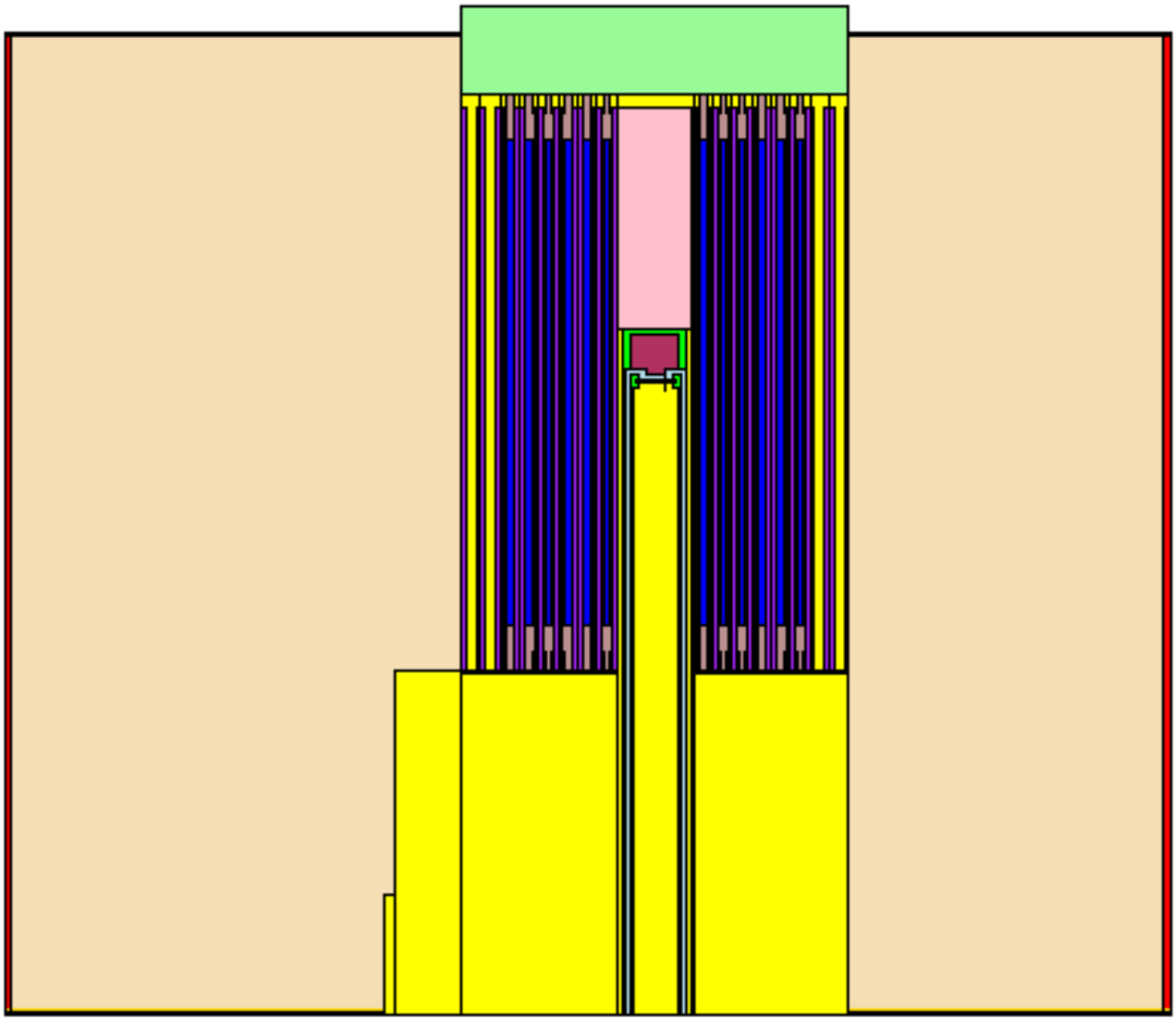

Figure 16. Vertical section of the YALINA thermal subcritical assembly MCNP/MCNPX model

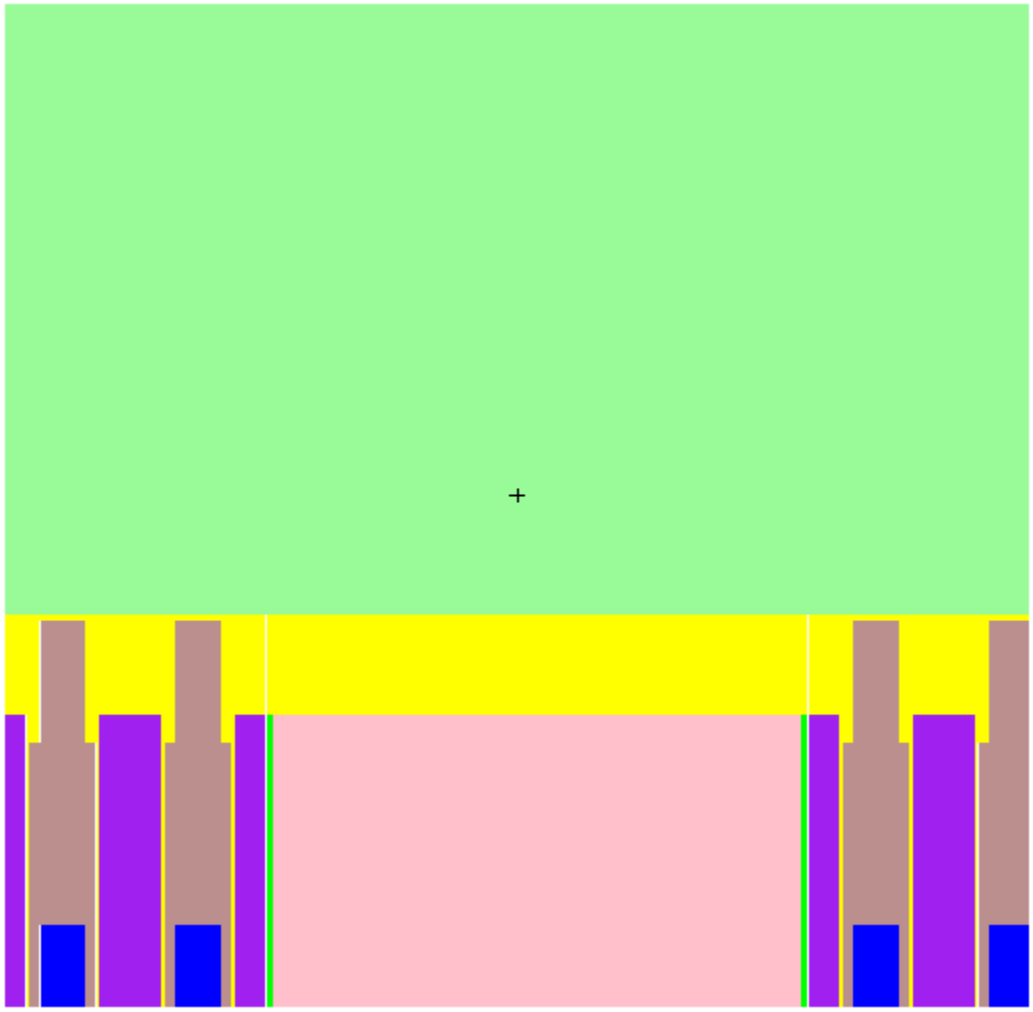

Figure 17. Zoom of the vertical section of the YALINA thermal subcritical assembly MCNP/MCNPX model

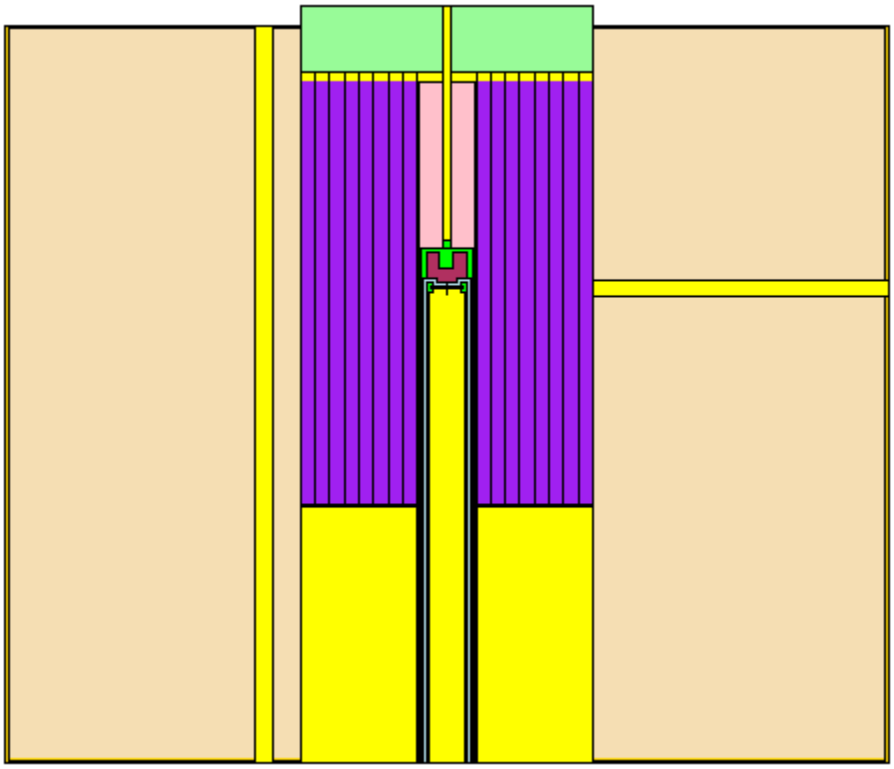

Figure 18. Vertical section of the YALINA thermal subcritical assembly MCNP/MCNPX model at x=0

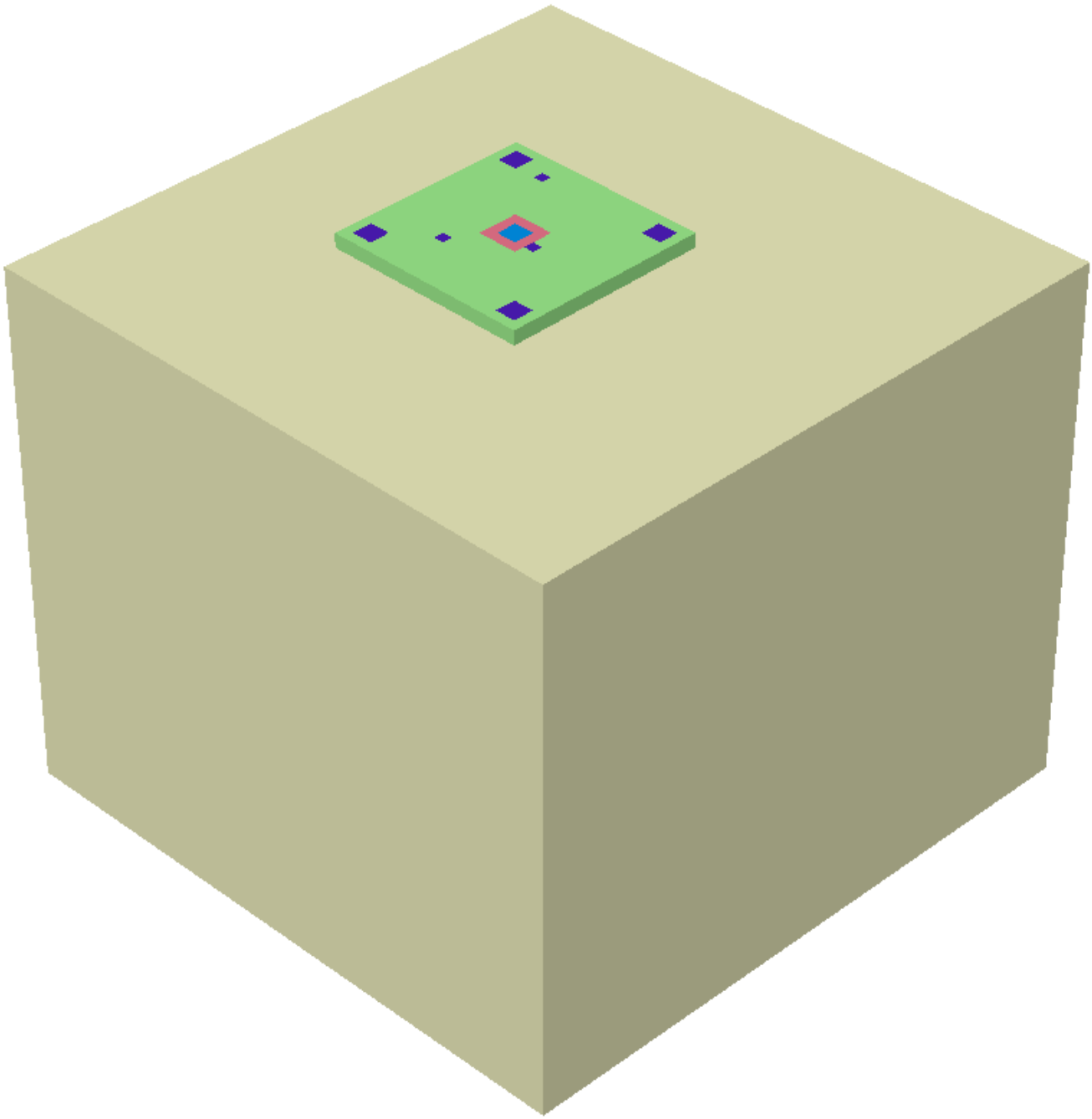

Figure 19. PARTISN model of YALINA Thermal

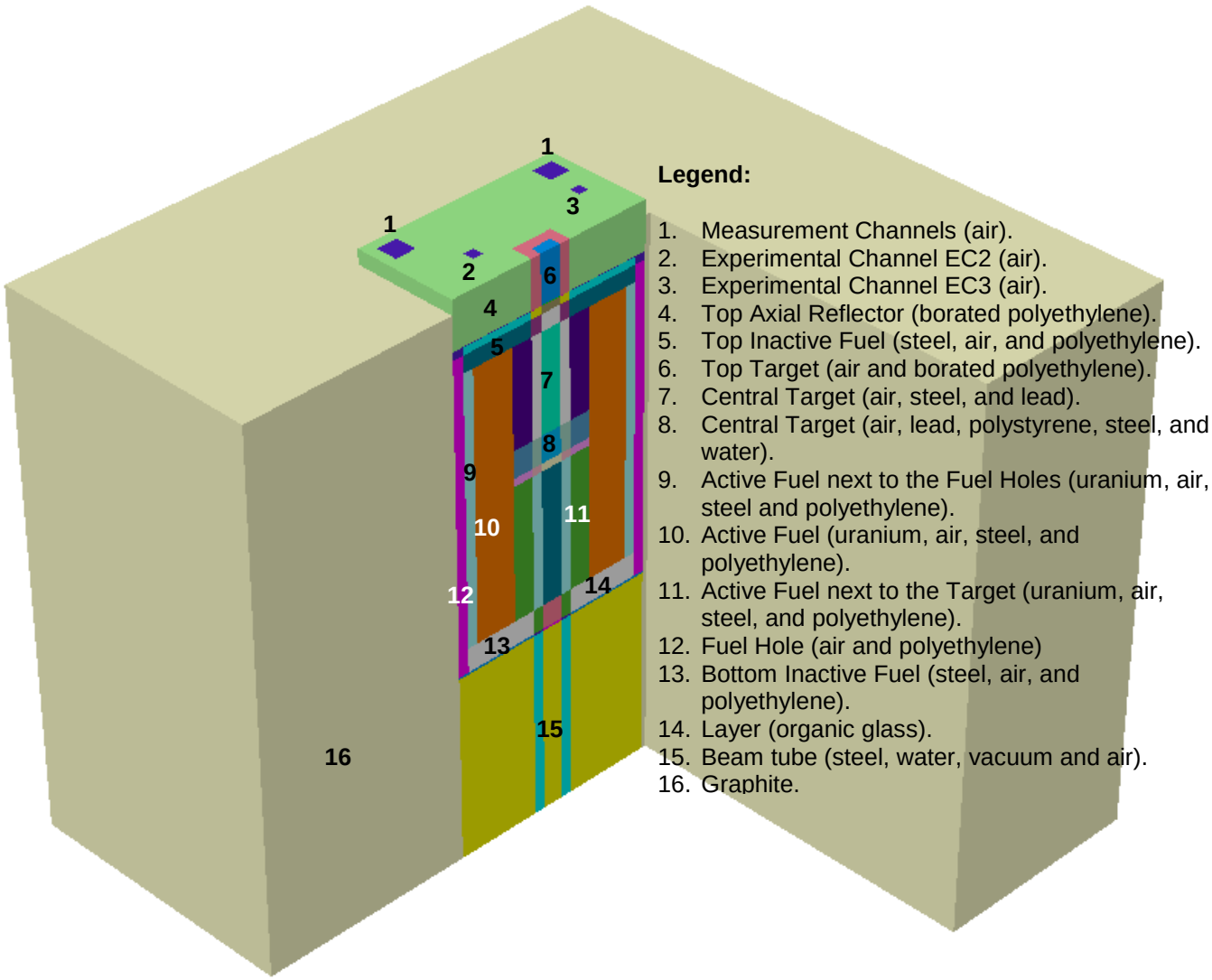

Figure 20. PARTISN model of YALINA Thermal with right side corner cut

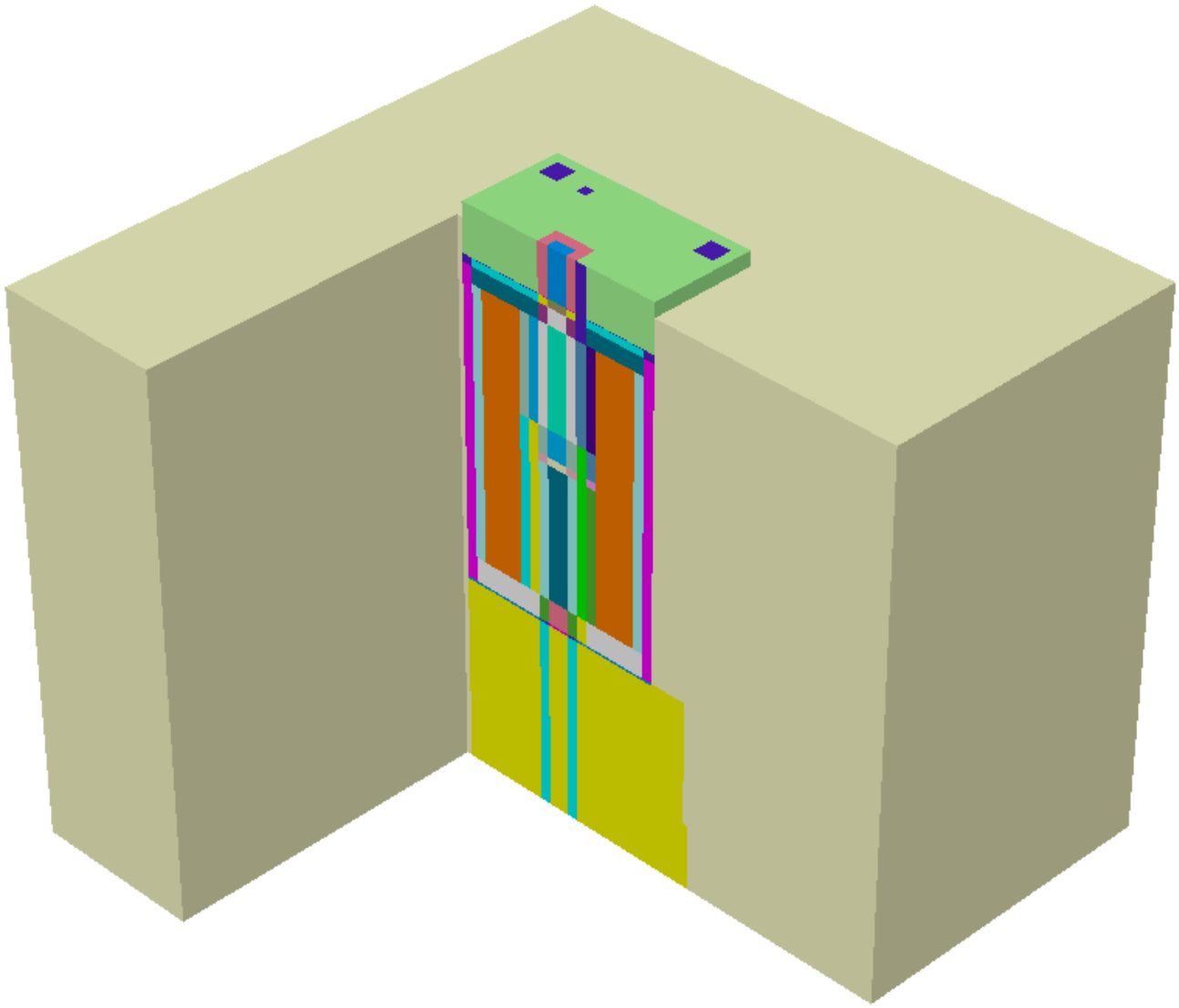

Figure 21. PARTISN model of YALINA Thermal with left side corner cut

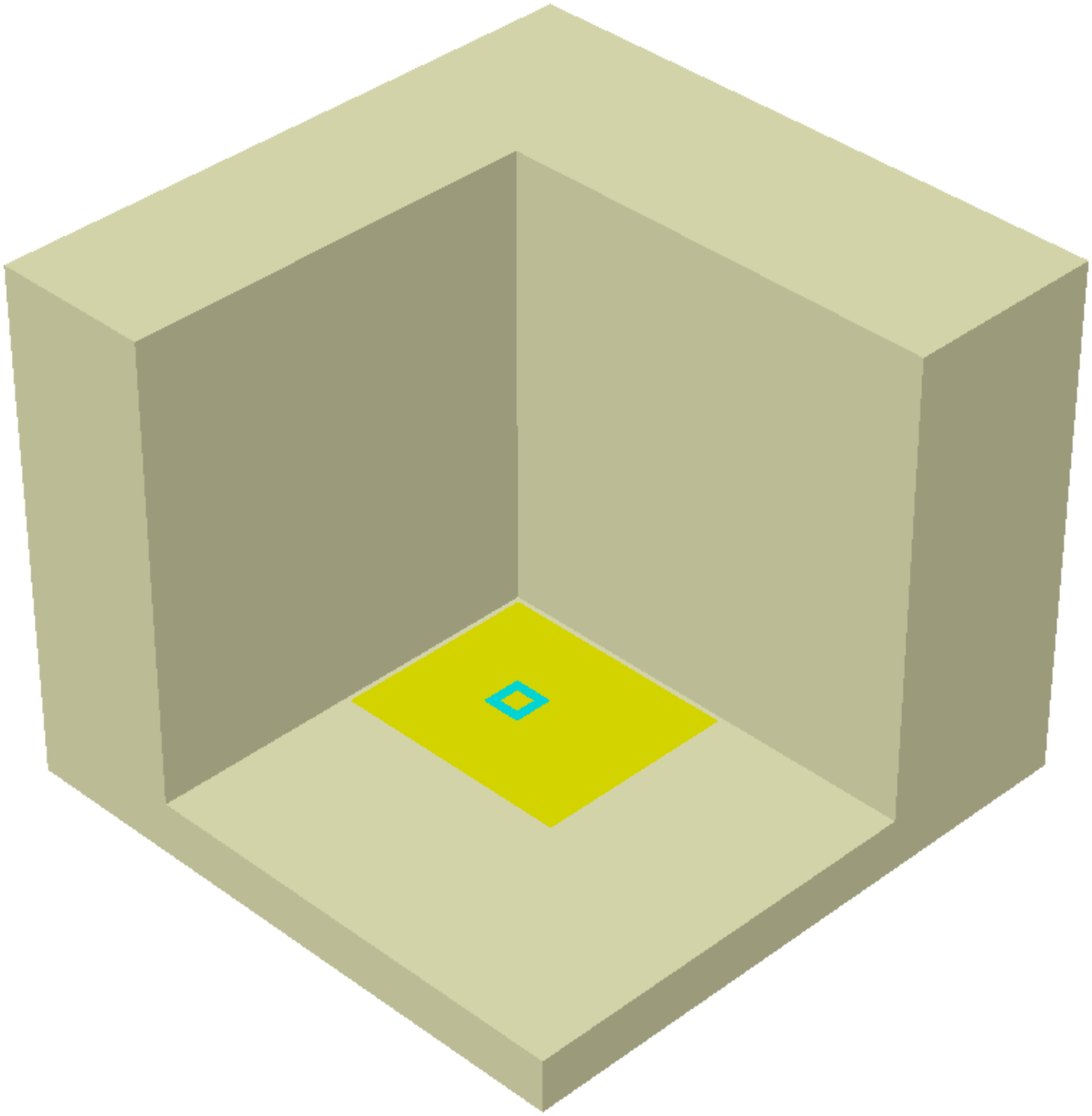

Figure 22. PARTISN model of YALINA Thermal with corner cut between z=0 and z=12.4

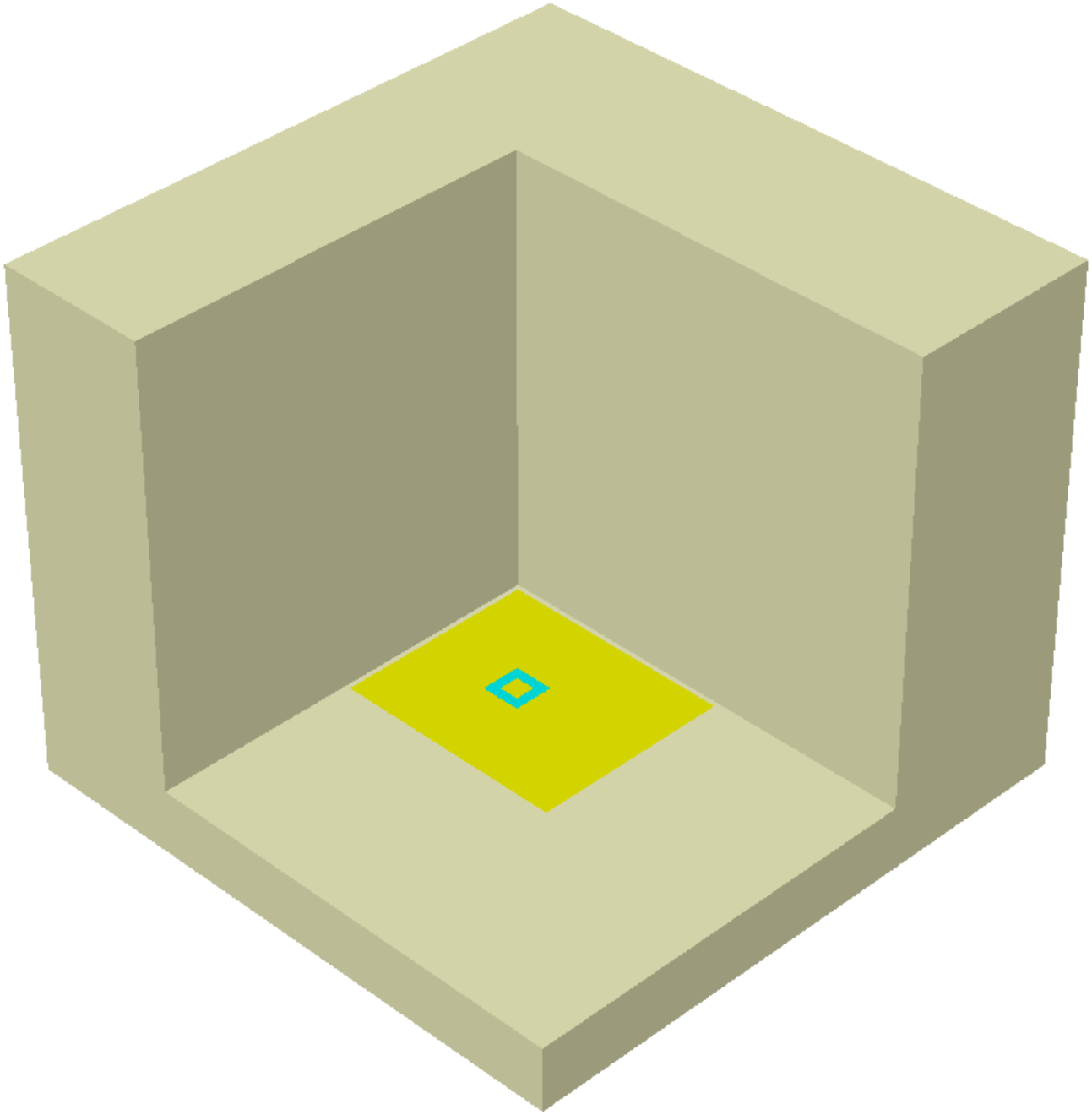

Figure 23. PARTISN model of YALINA Thermal with corner cut between z=12.4 and z=35

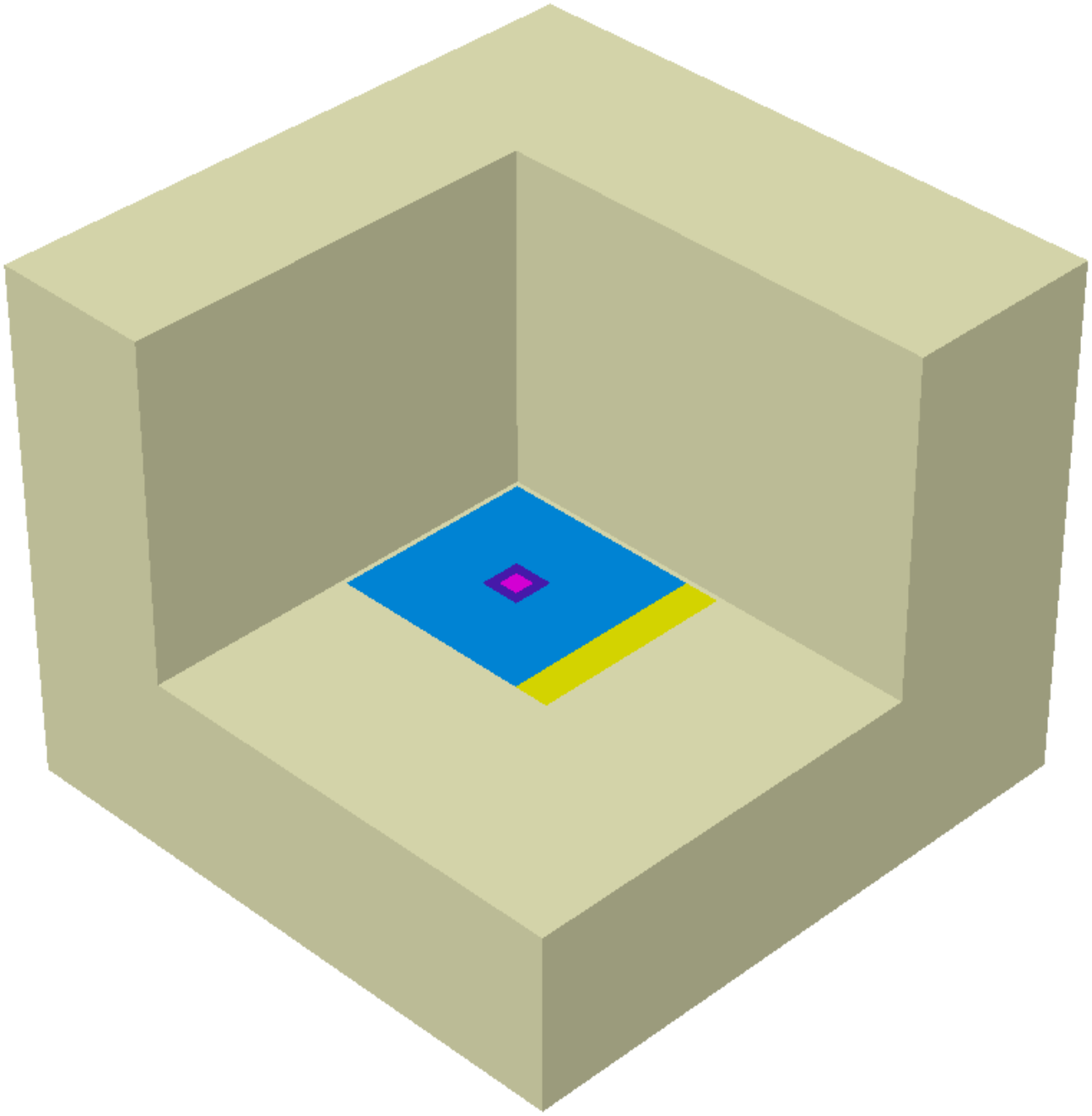

Figure 24. PARTISN model of YALINA Thermal with corner cut between z=35 and z=35.4

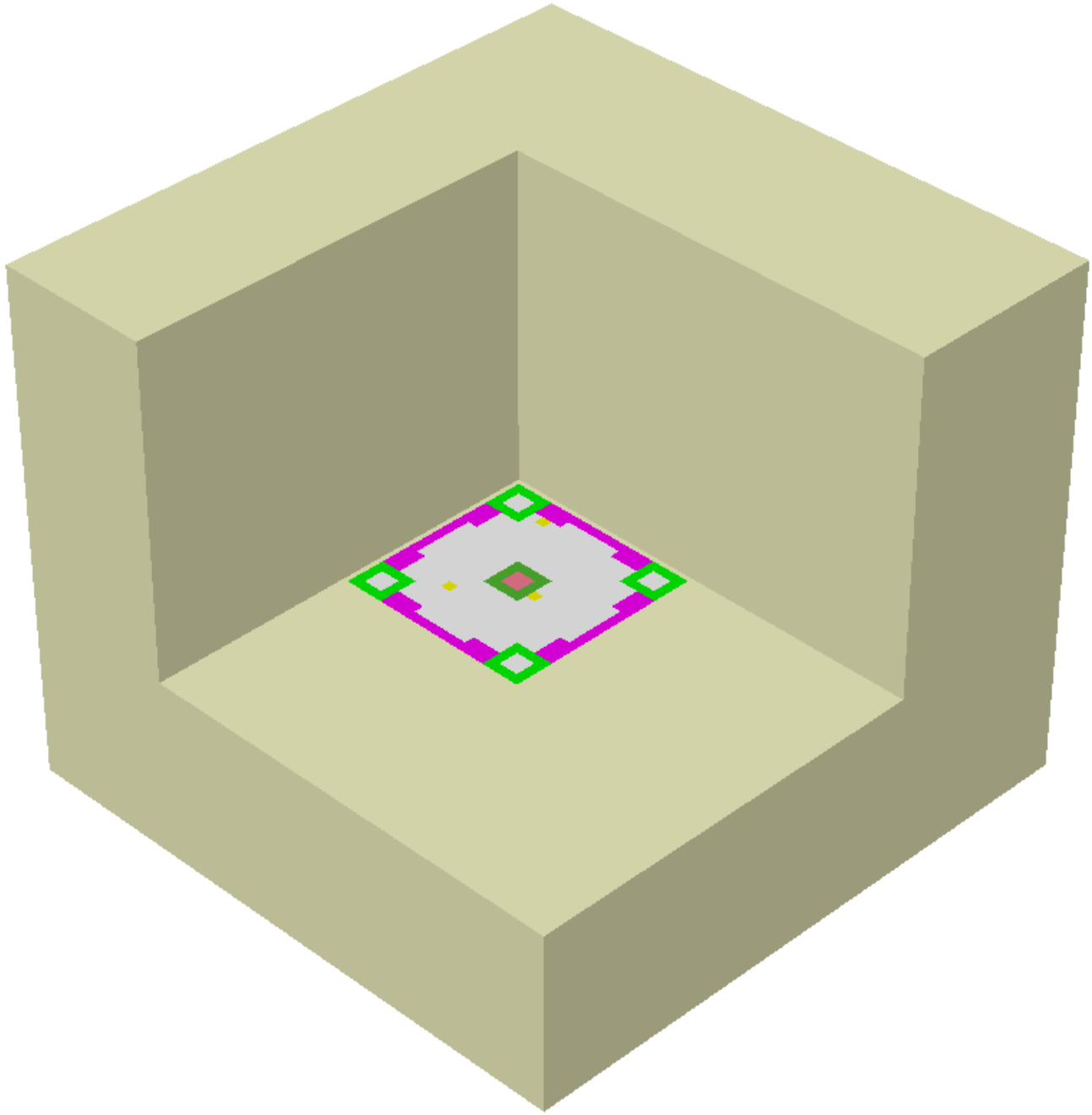

Figure 25. PARTISN model of YALINA Thermal with corner cut between z=35.4 and z=39.9

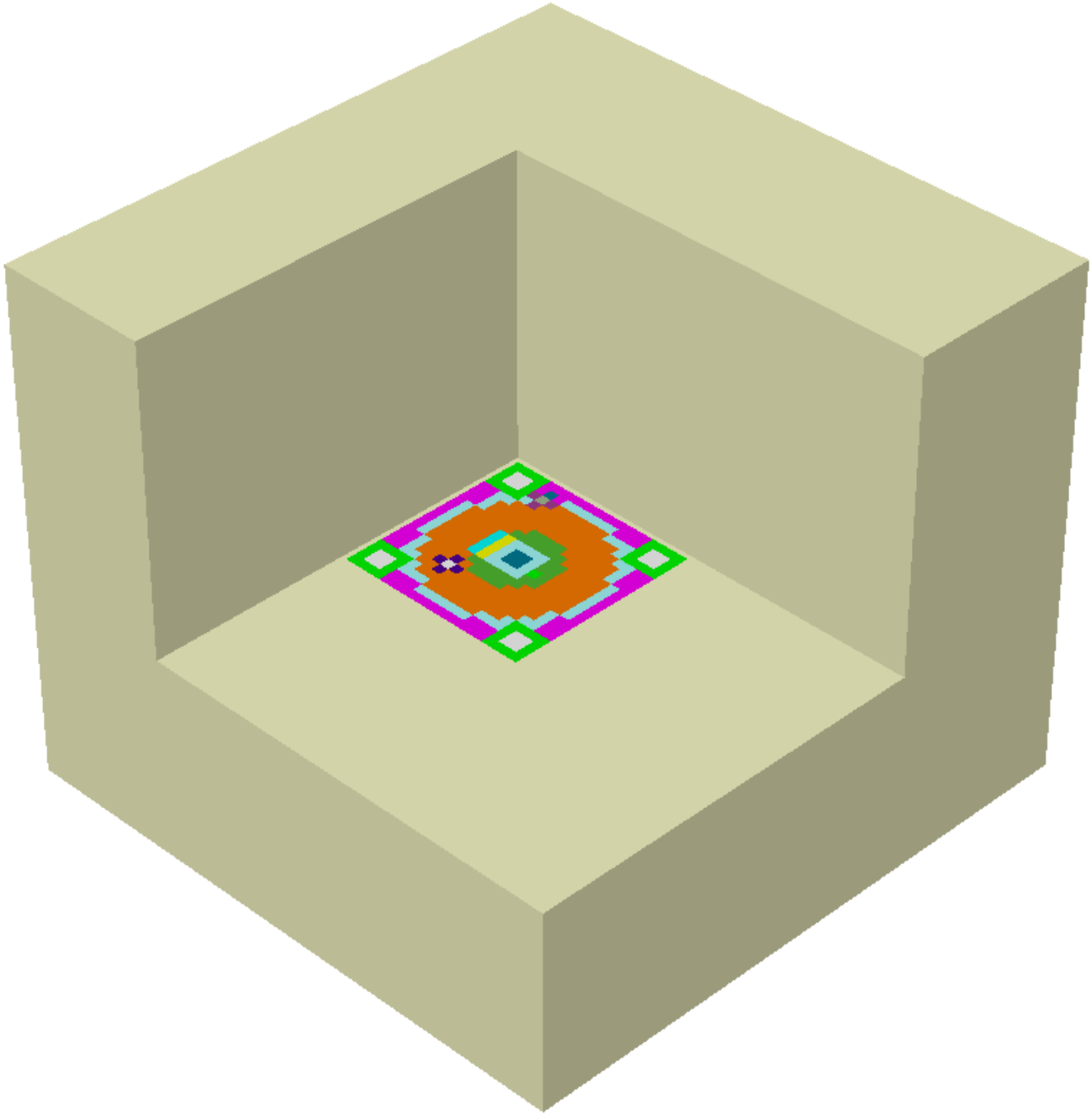

Figure 26. PARTISN model of YALINA Thermal with corner cut between z=39.9 and z=64.9

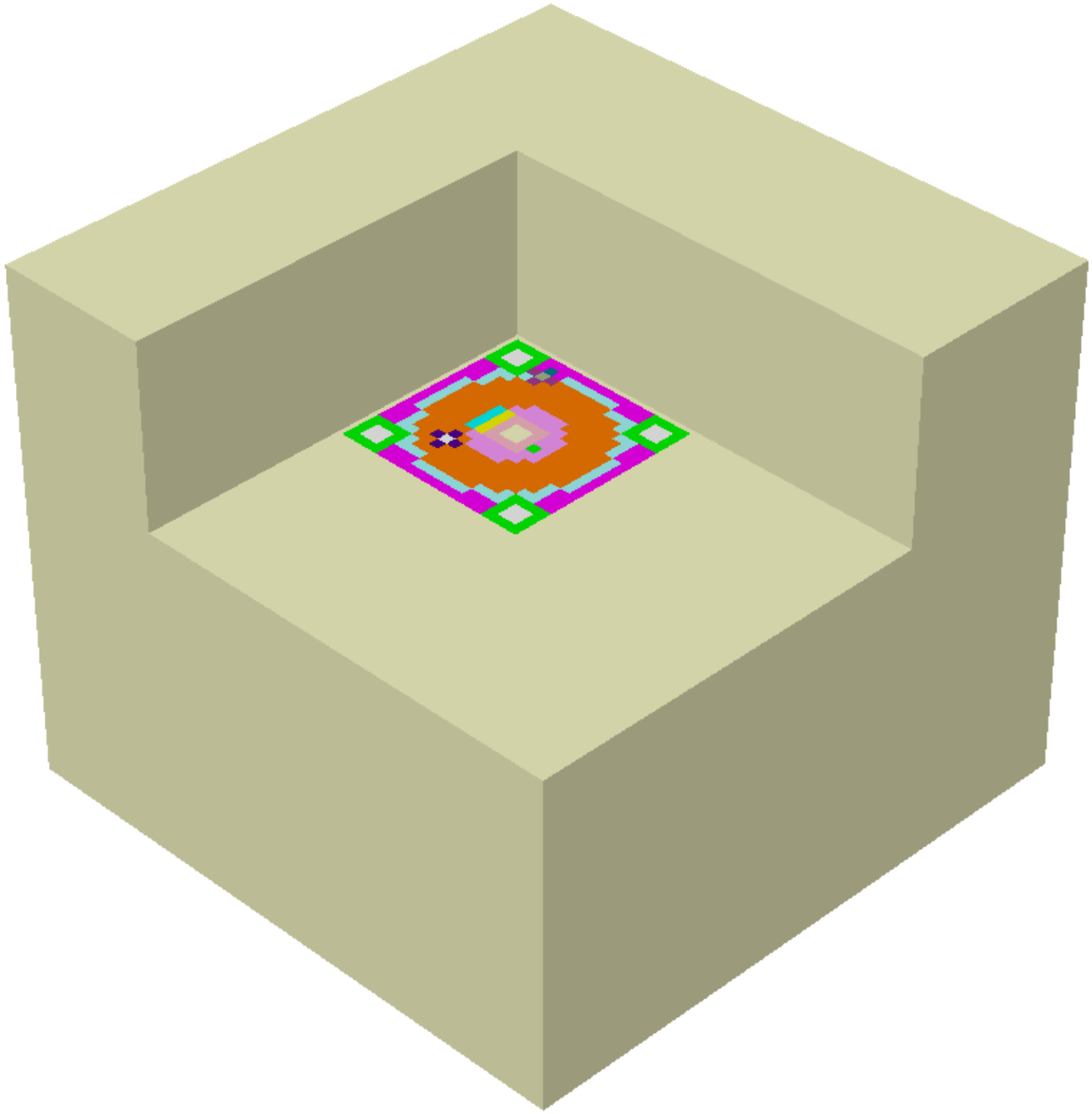

Figure 27. PARTISN model of YALINA Thermal with corner cut between z=64.9 and z=66.35

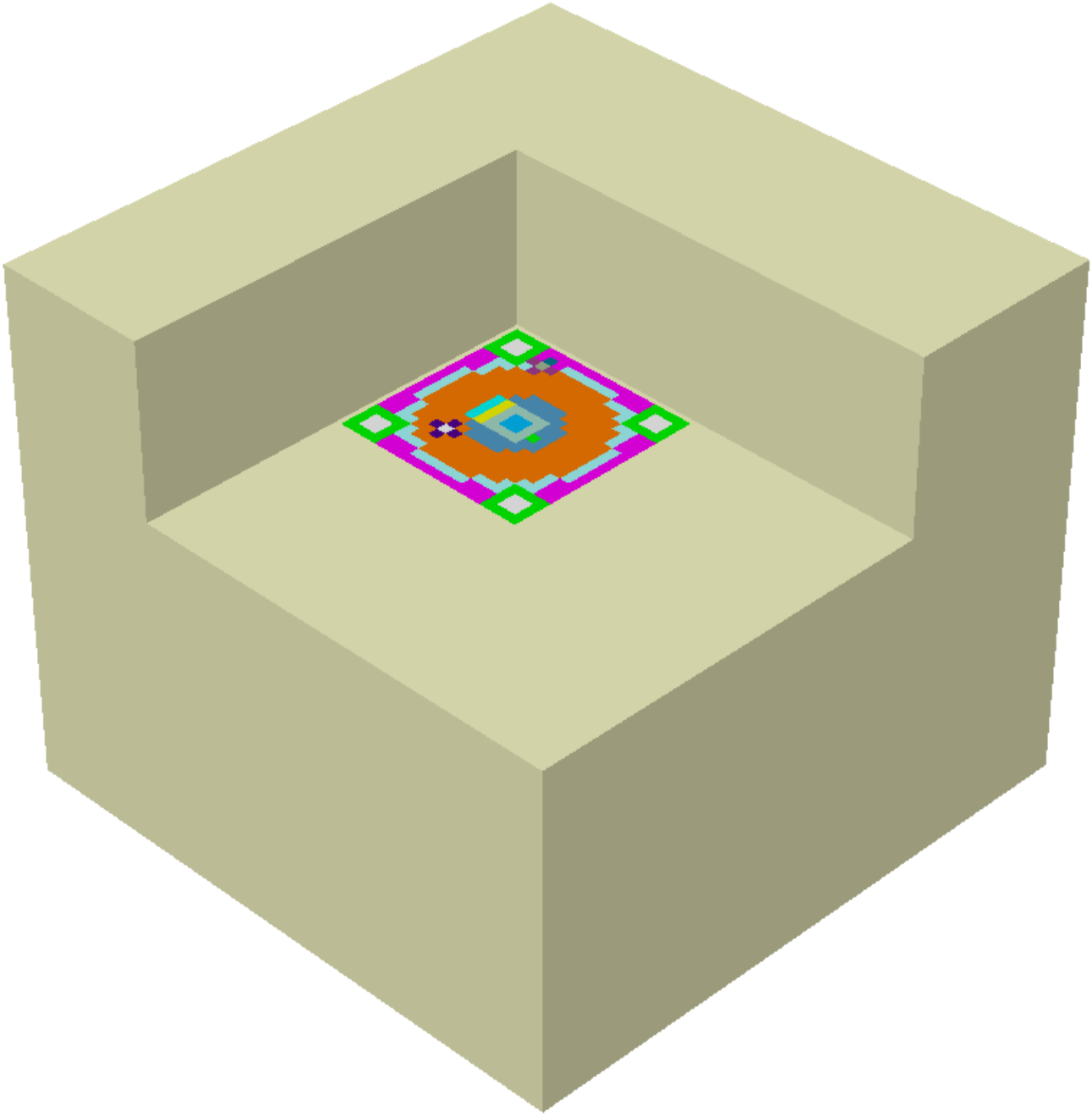

Figure 28. PARTISN model of YALINA Thermal with corner cut between z=66.35 and z=71.4

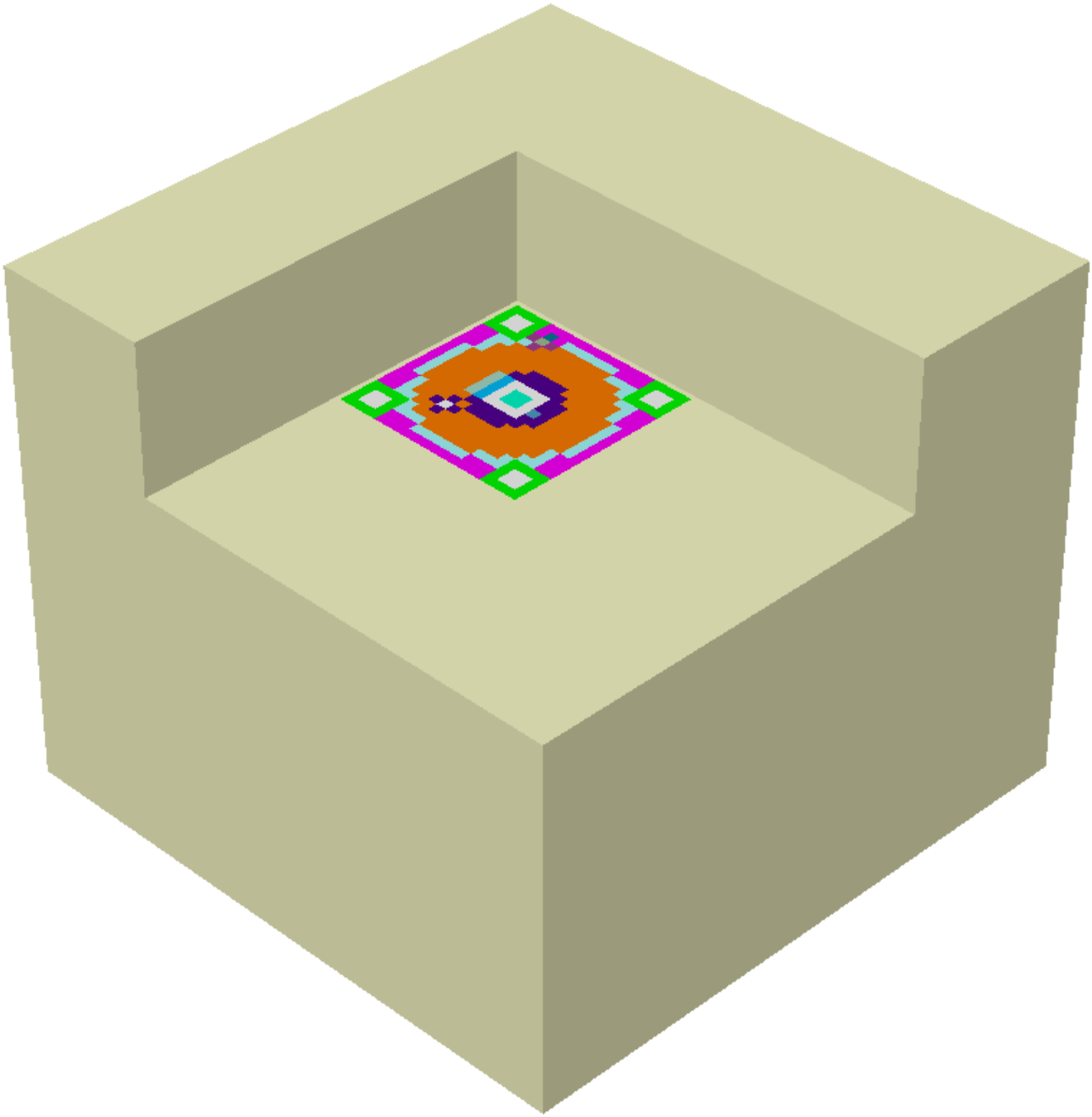

Figure 29. PARTISN model of YALINA Thermal with corner cut between z=71.4 and z=89.9

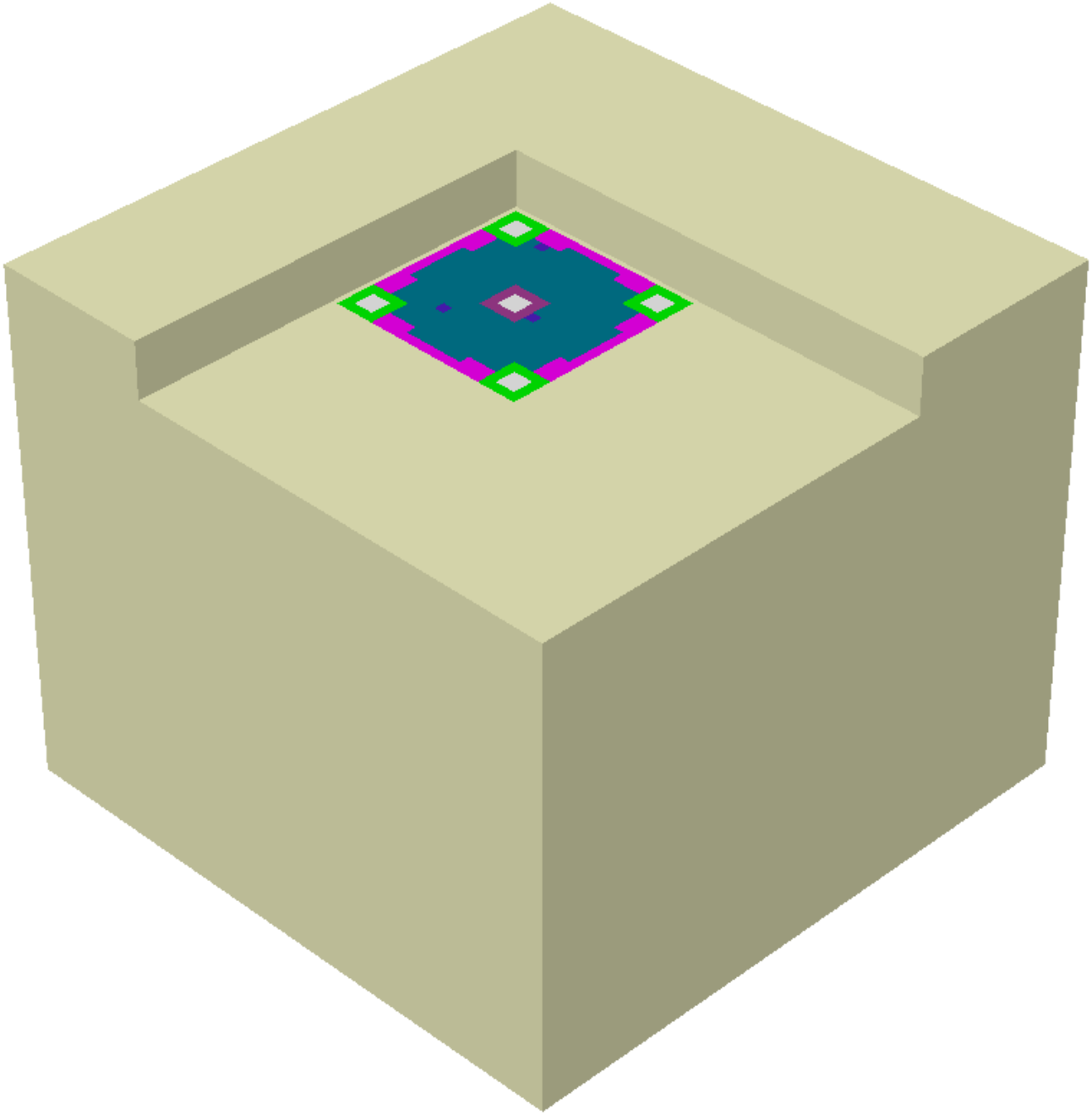

Figure 30. PARTISN model of YALINA Thermal with corner cut between z=89.9 and z=93

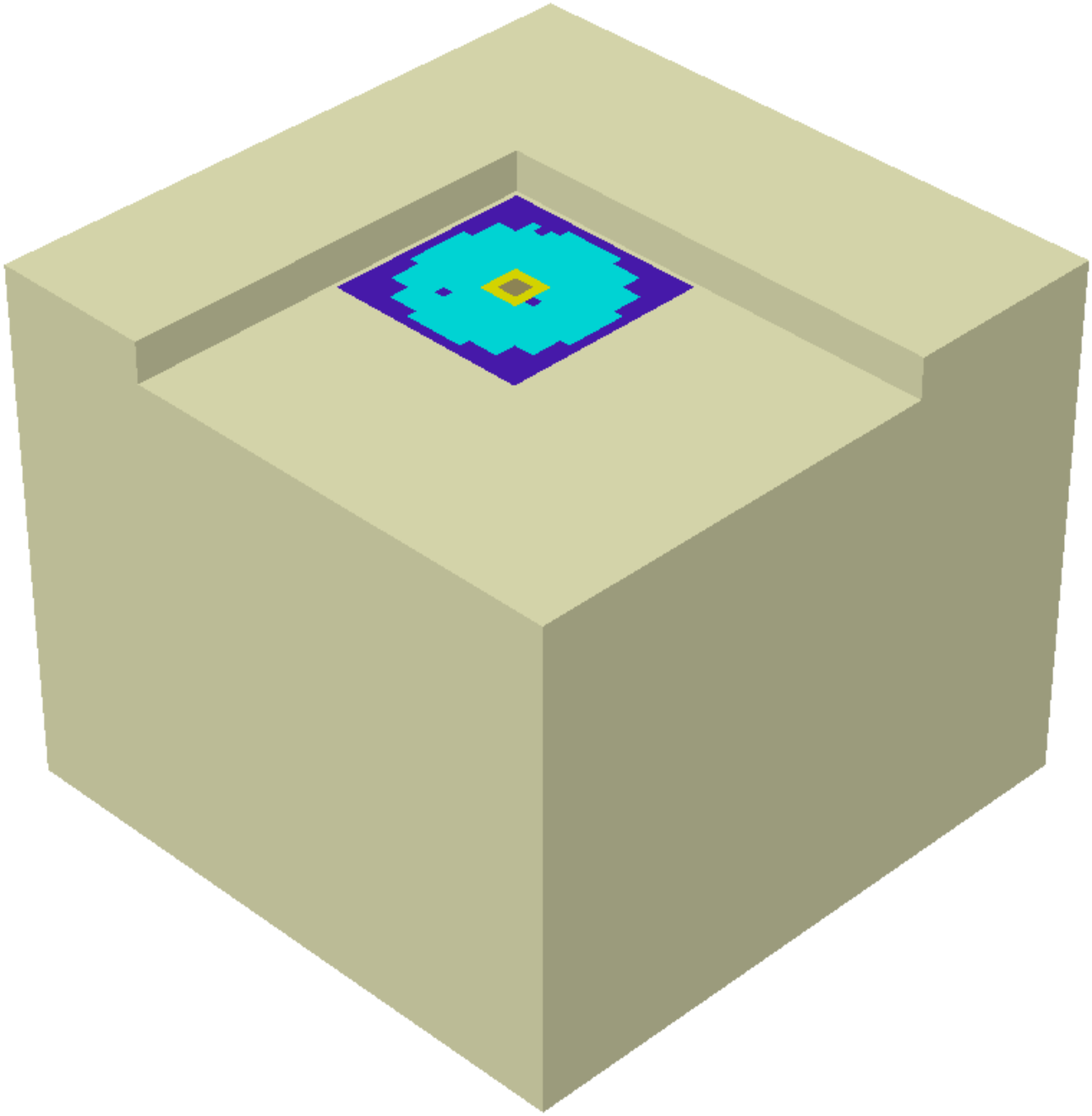

Figure 31. PARTISN model of YALINA Thermal with corner cut between z=93 and z=94.4

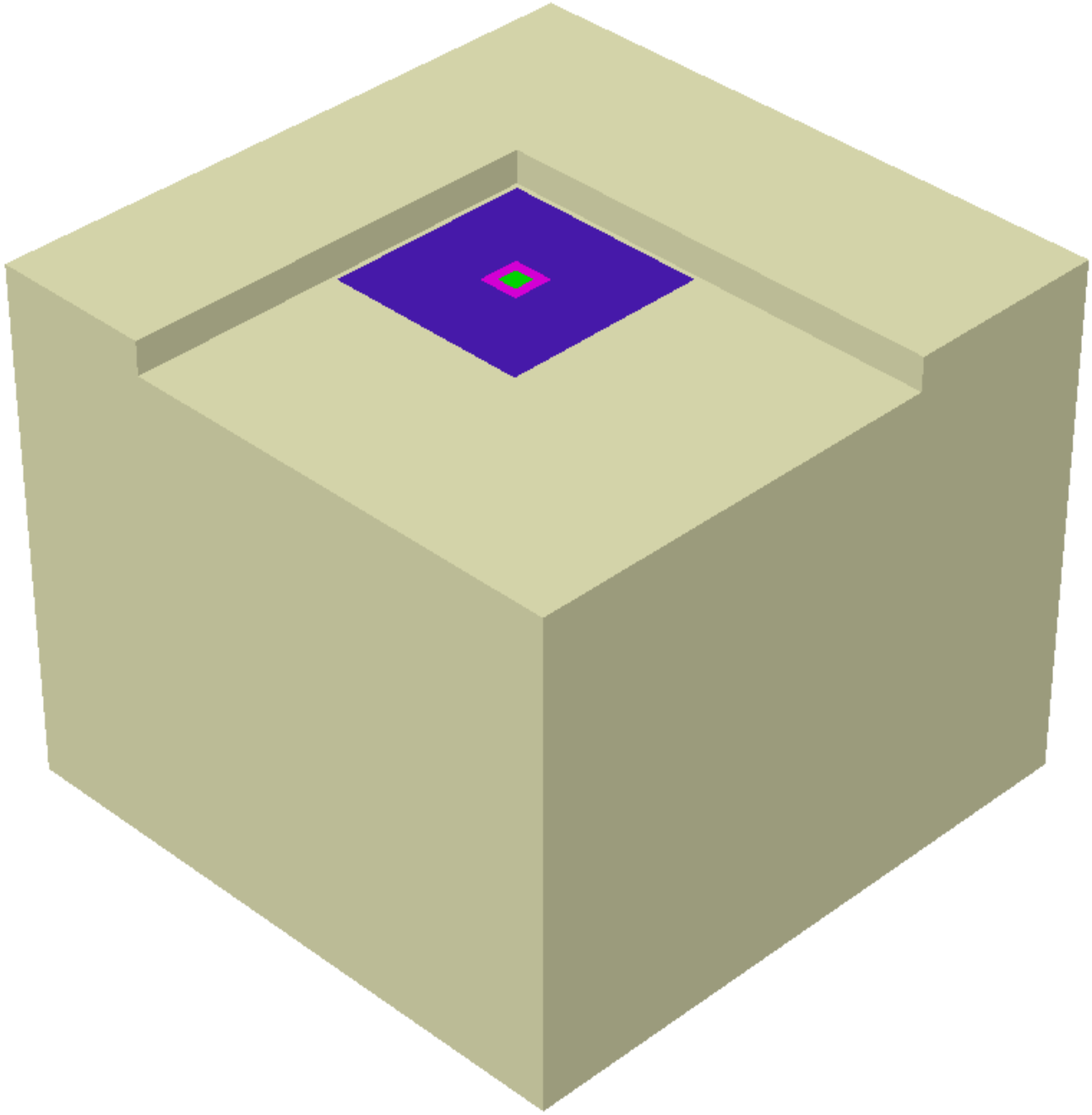

Figure 32. PARTISN model of YALINA Thermal with corner cut between z=94.4 and z=94.5

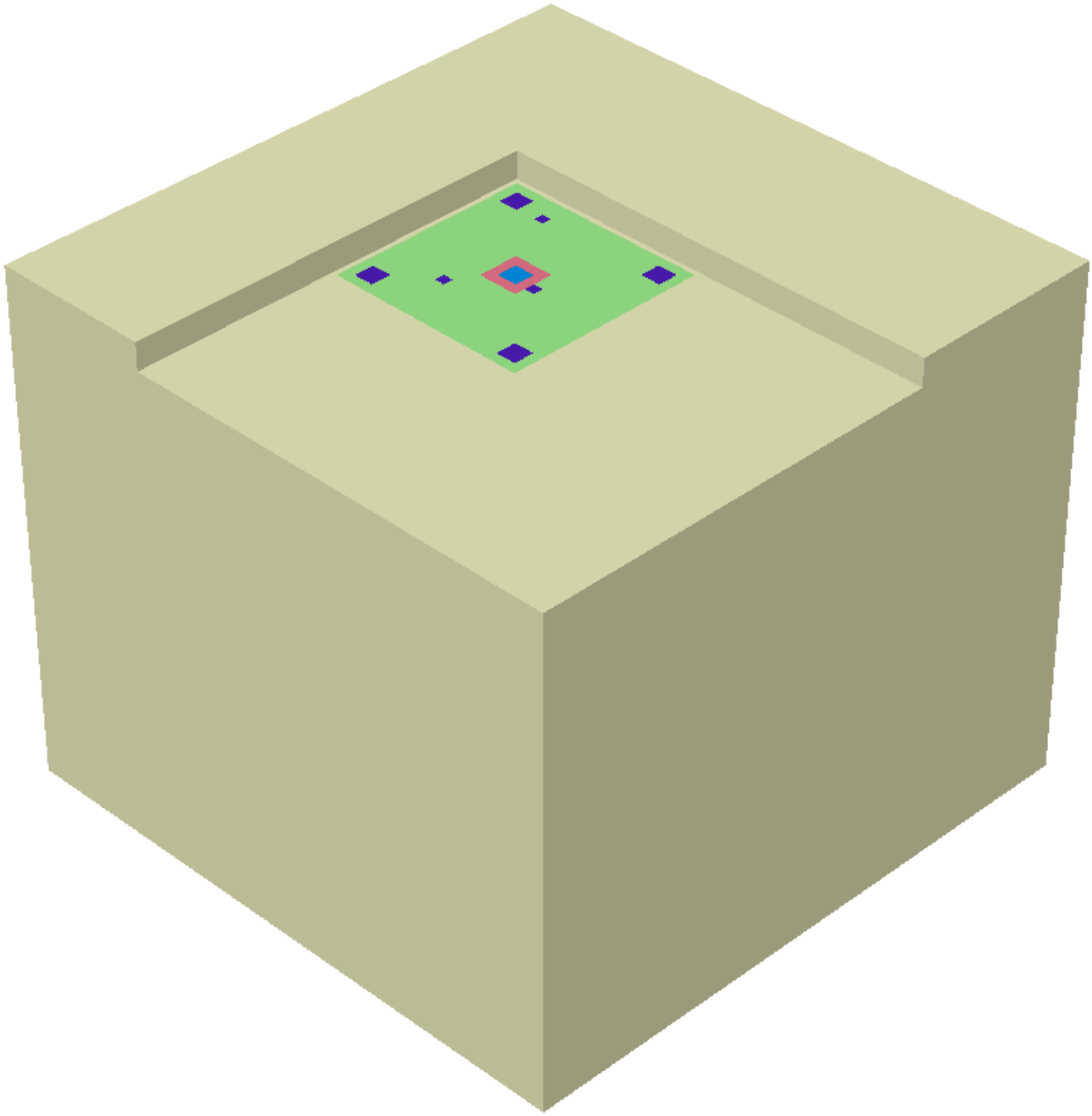

Figure 33. PARTISN model of YALINA Thermal with corner cut between z=94.5 and z=103.5

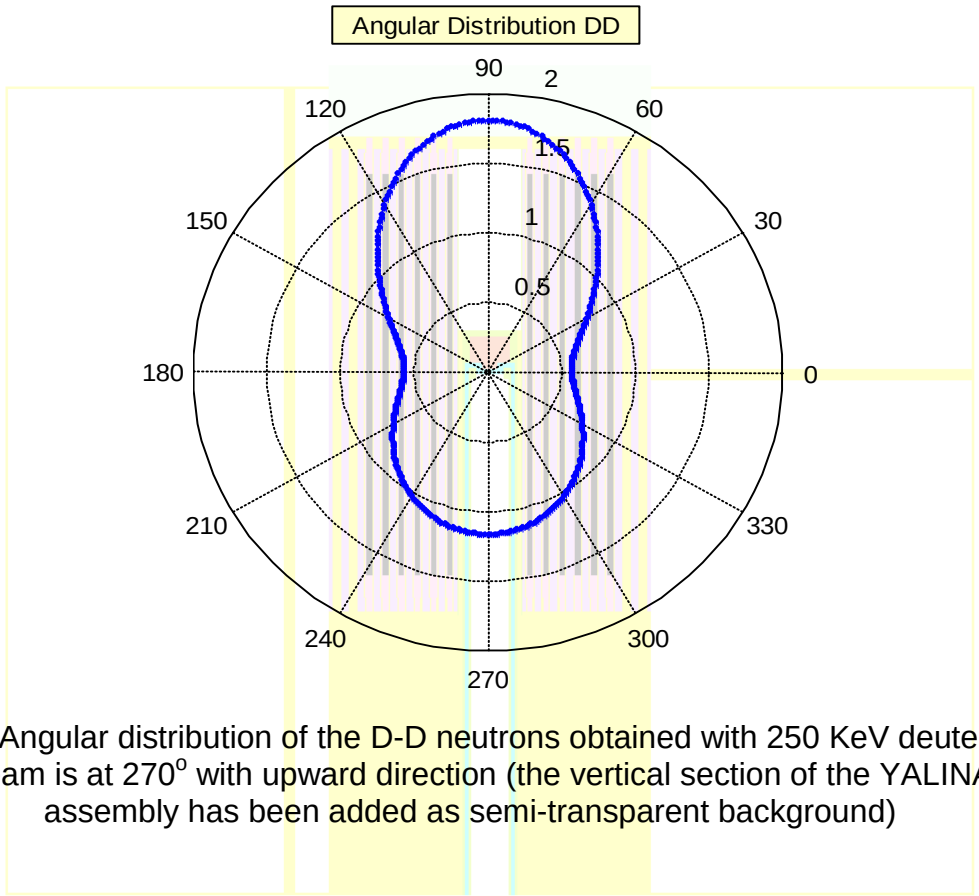

Figure 34. Angular distribution of the D-D neutrons obtained with 250 KeV deuterons, the deuteron beam is at 270° with upward direction (the vertical section of the YALINA thermal assembly has been added as semi-transparent background)

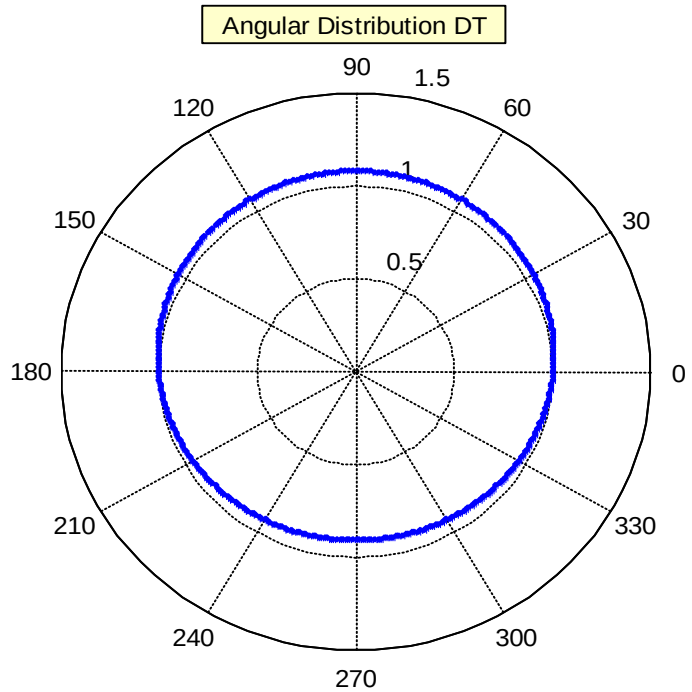

Figure 35. Angular distribution of D-T neutrons obtained with 250 KeV deuterons, the deuteron beam is at 270° with upward direction

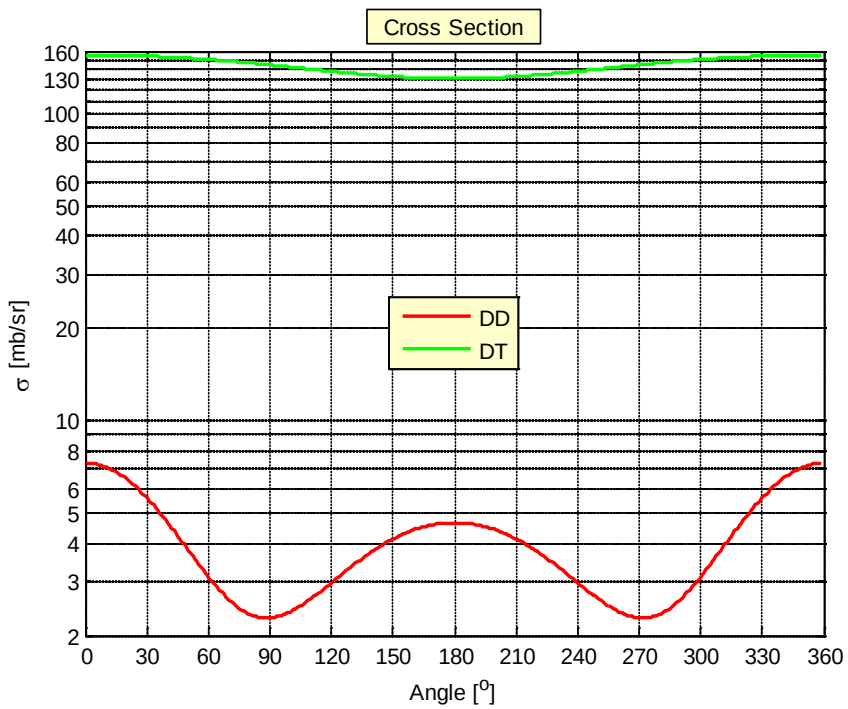

Figure 36. Microscopic differential cross section of D-D and D-T neutrons obtained with 250 KeV deuterons, the angle is between the deuteron beam and the neutron flight path (0° in the current figure corresponds to 90° in Figures 34 and 35 and it means the neutron is emitted in the forward direction relative to the deuteron beam direction)

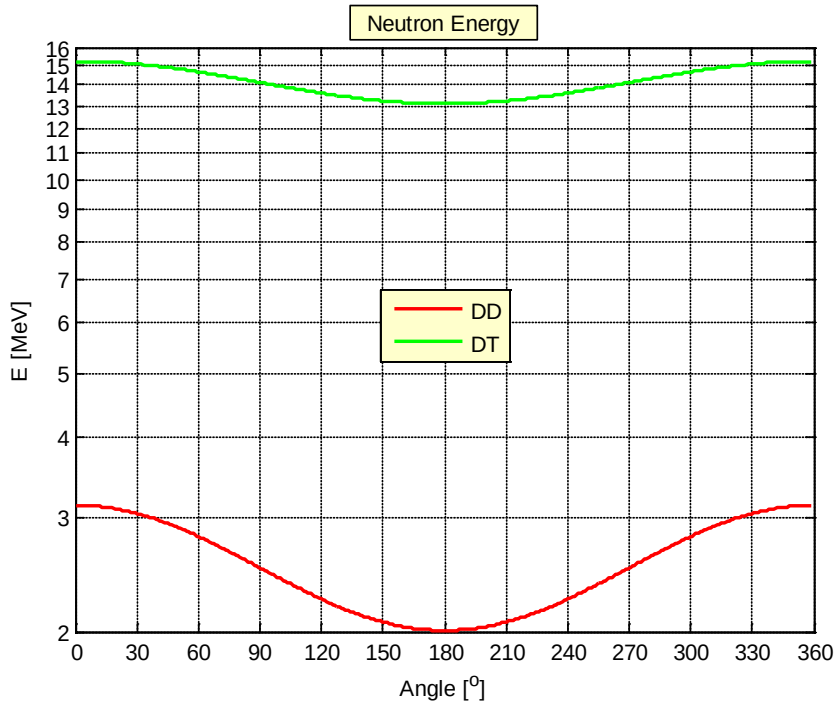

Figure 37. Energy of D-D and D-T neutrons obtained with 250 KeV deuterons, the angle is between the deuteron beam and the neutron flight path (0° in the current Figure corresponds to 90° in Figures 34 and 35 and it means the neutron is emitted in the forward direction relative to the deuteron beam direction)

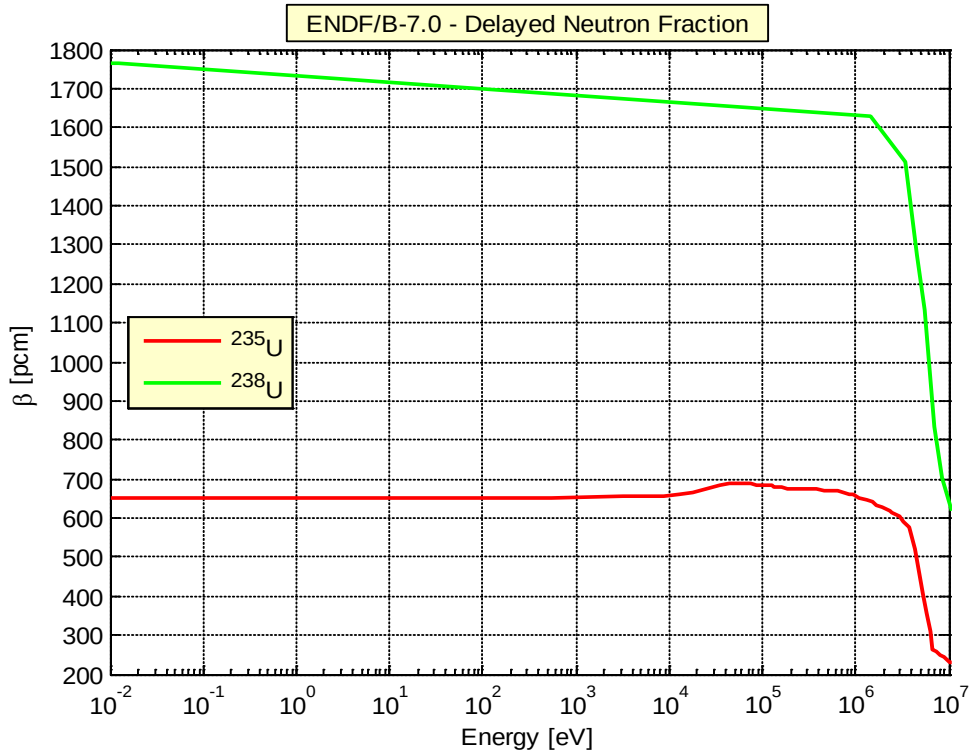

Figure 38. Delayed neutron fraction as a function of incident neutron energy

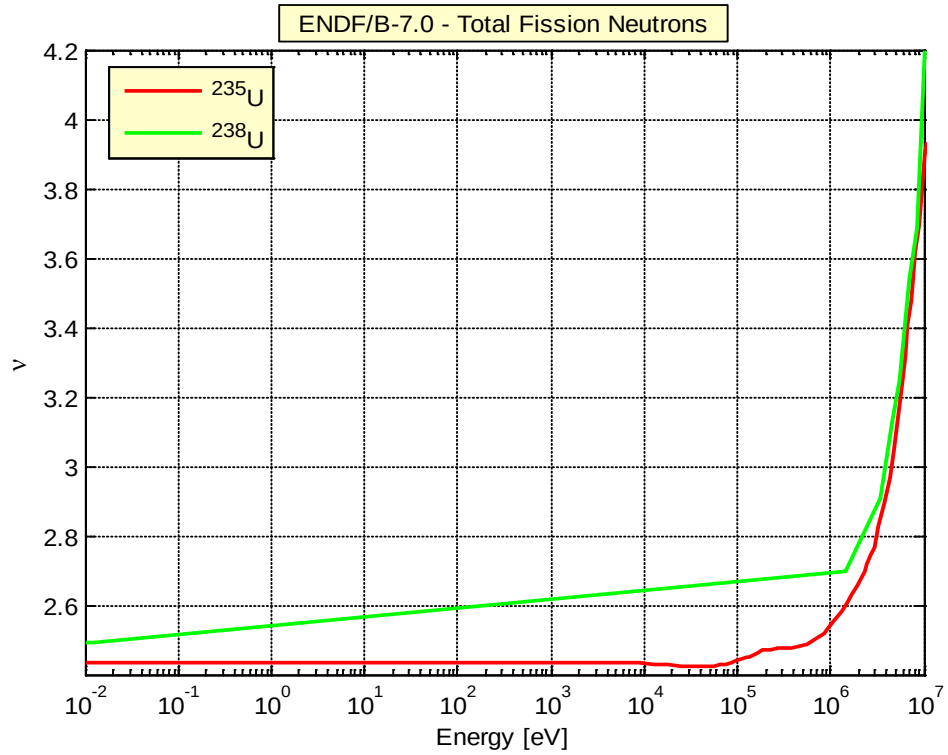

Figure 39. Total number of fission neutrons as a function of incident neutron energy

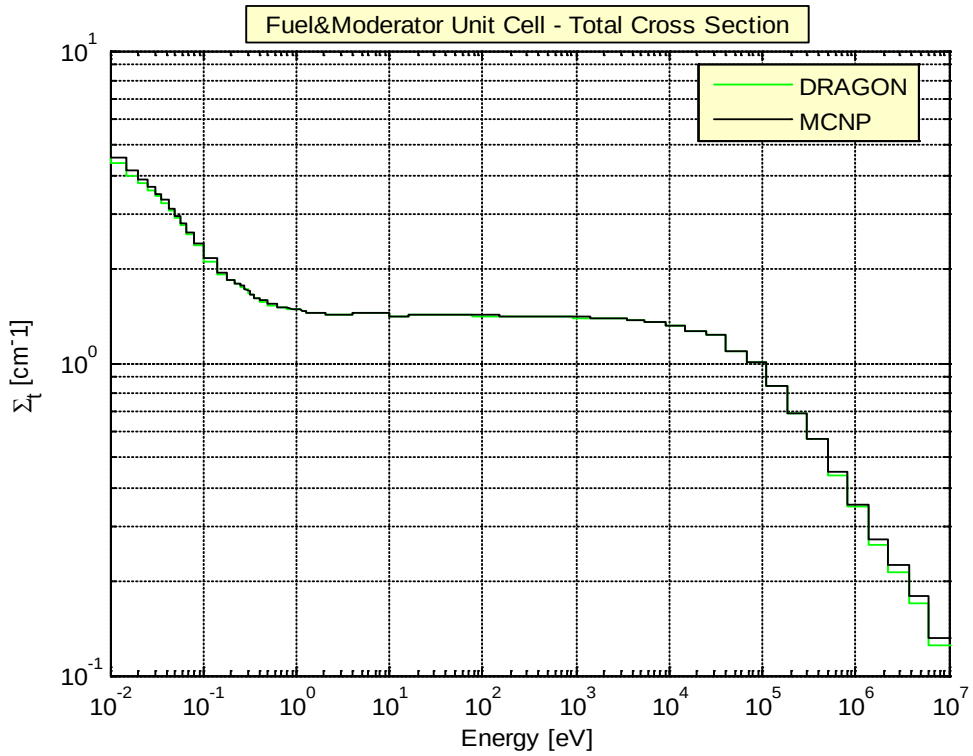

Figure 40. Total macroscopic cross section of the fuel and moderator unit cell

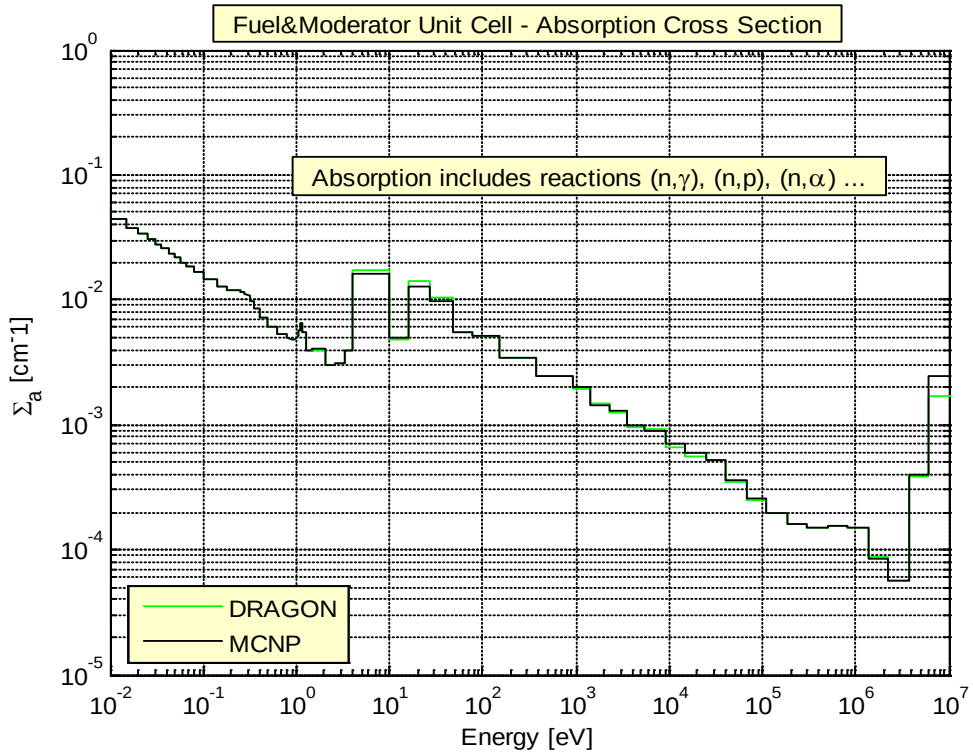

Figure 41. Absorption macroscopic cross section of the fuel and moderator unit cell

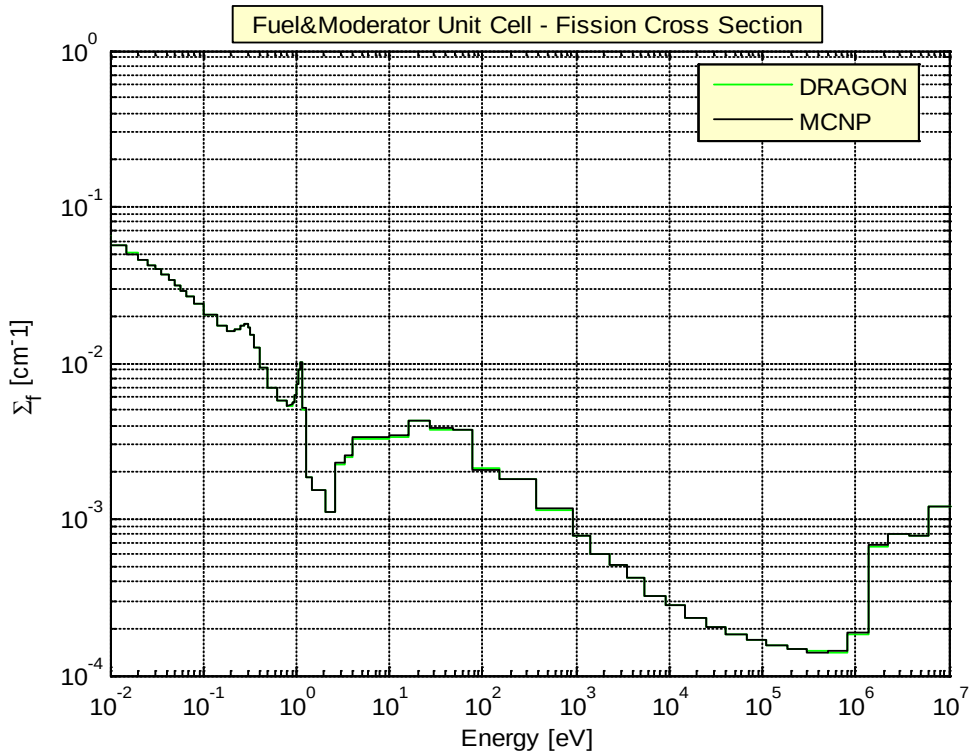

Figure 42. Fission macroscopic cross section of the fuel and moderator unit cell

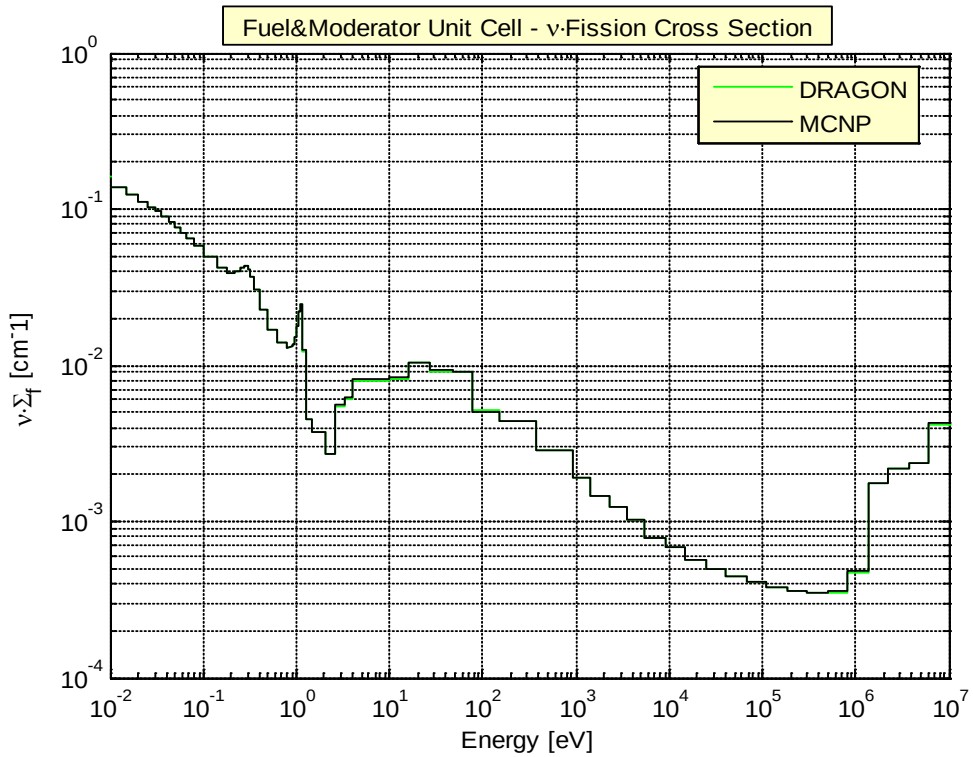

Figure 43. Fission neutrons multiplied by the fission macroscopic cross section of the fuel and moderator unit cell

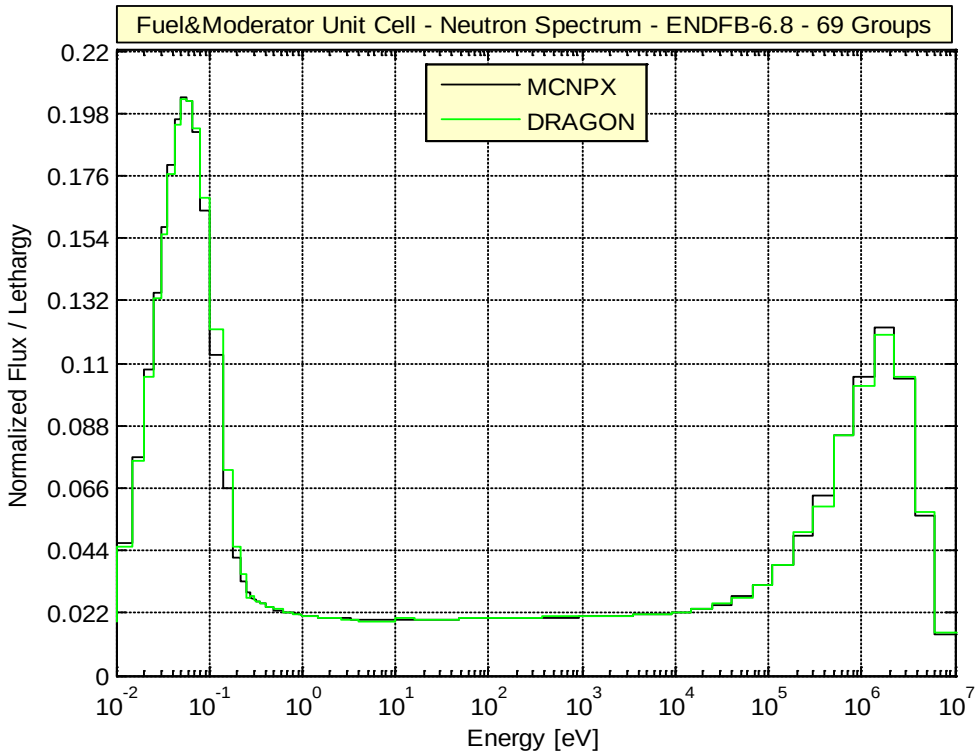

Figure 44. Neutron spectra of the fuel and moderator unit cell

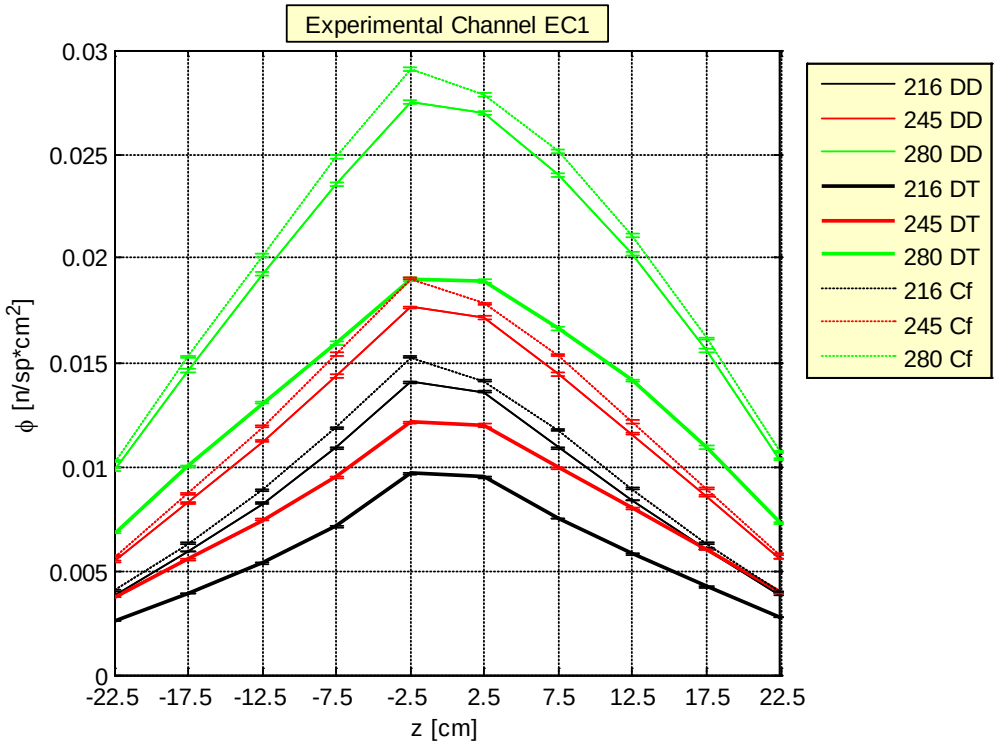

Figure 45. Neutron flux in the EC1 experimental channel calculated by MCNPX

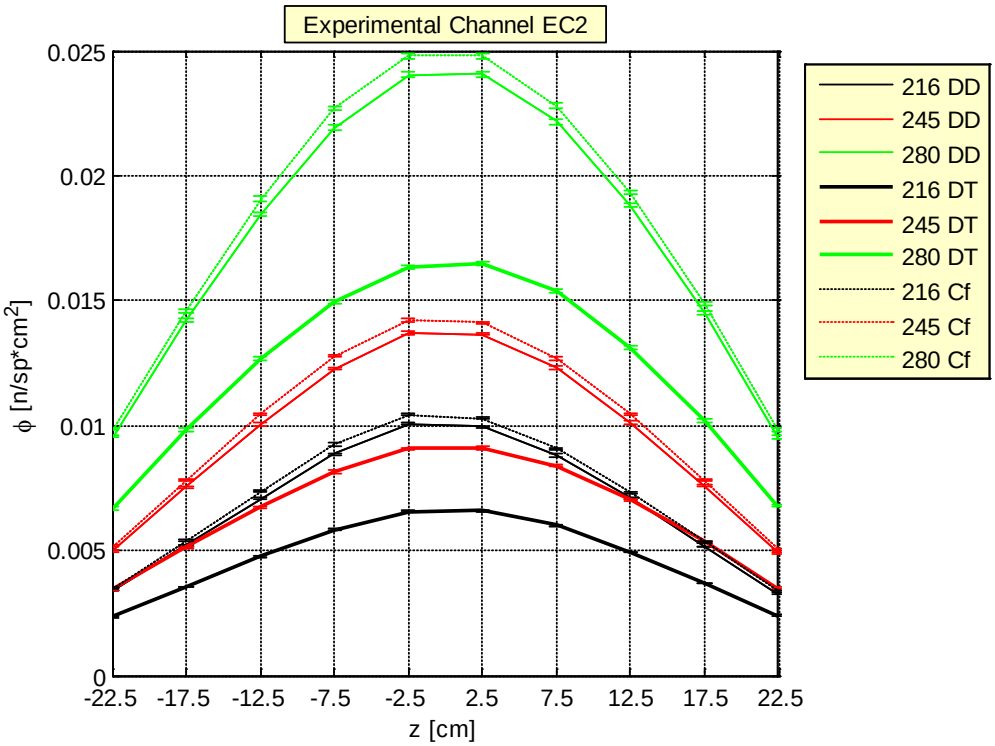

Figure 46. Neutron flux in the EC2 experimental channel calculated by MCNPX

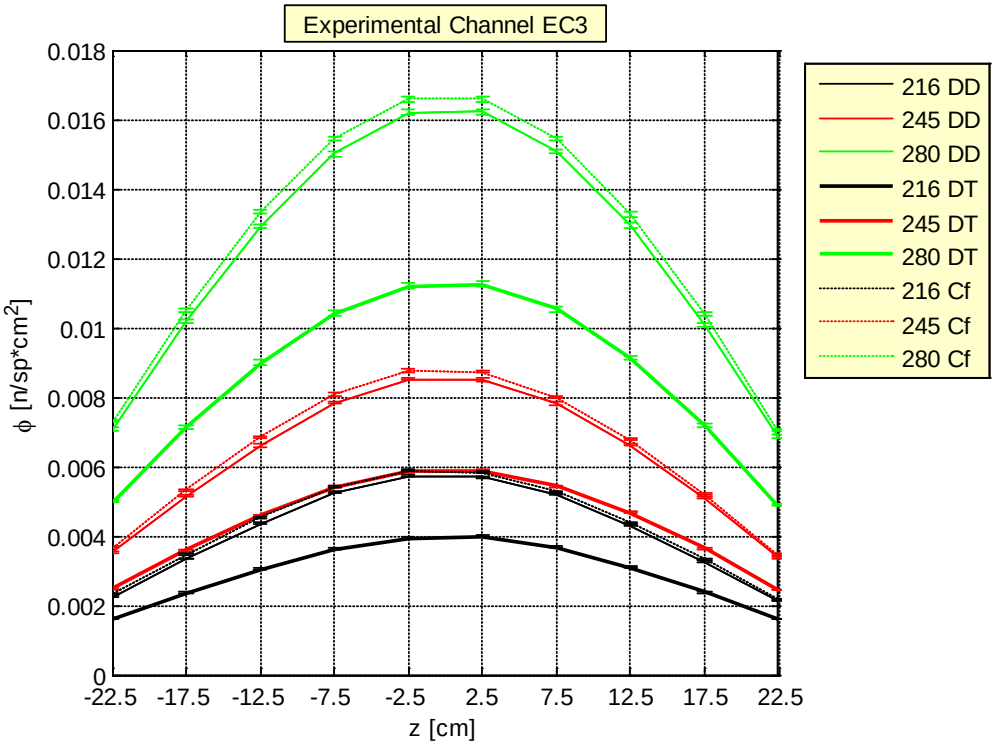

Figure 47. Neutron flux in the EC3 experimental channel calculated by MCNPX

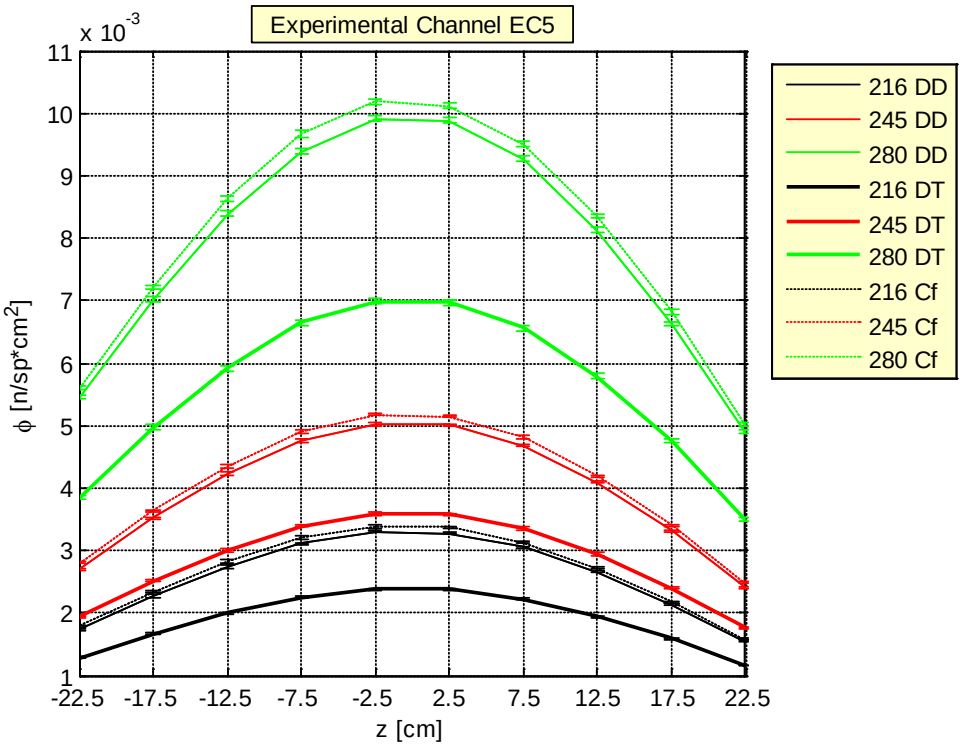

Figure 48. Neutron flux in the EC5 experimental channel calculated by MCNPX

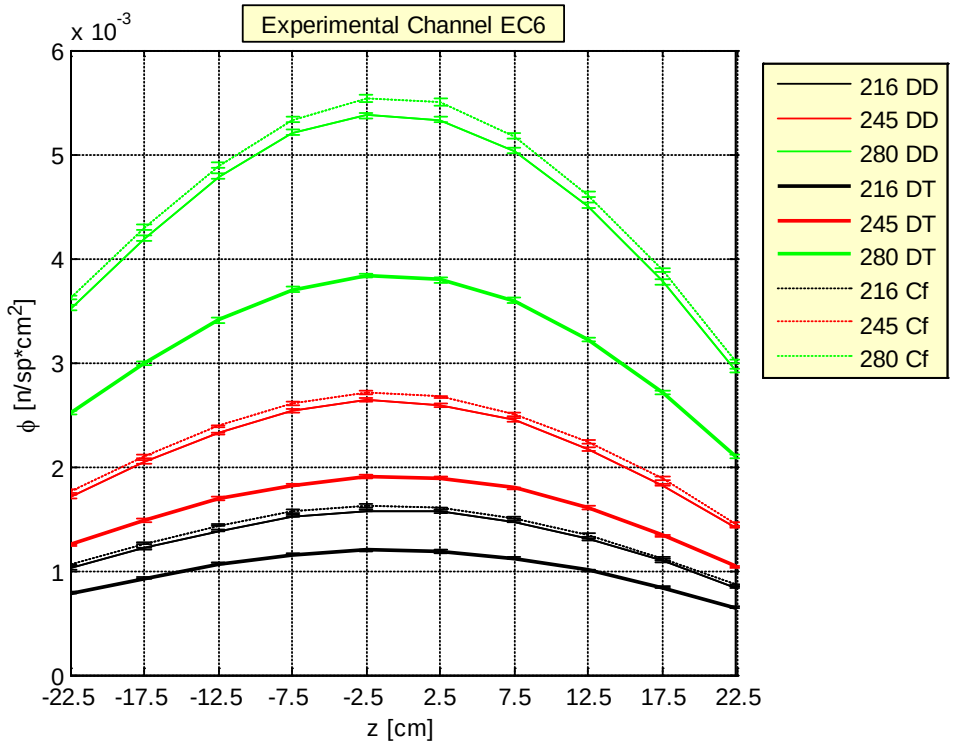

Figure 49. Neutron flux in the EC6 experimental channel calculated by MCNPX

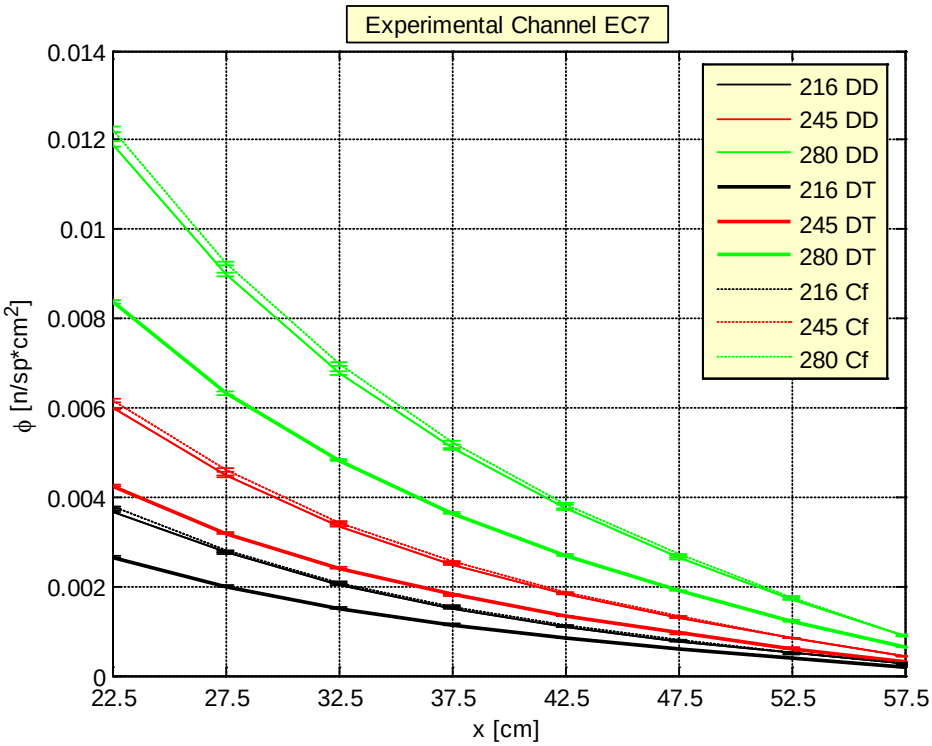

Figure 50. Neutron flux in the EC7 experimental channel calculated by MCNPX

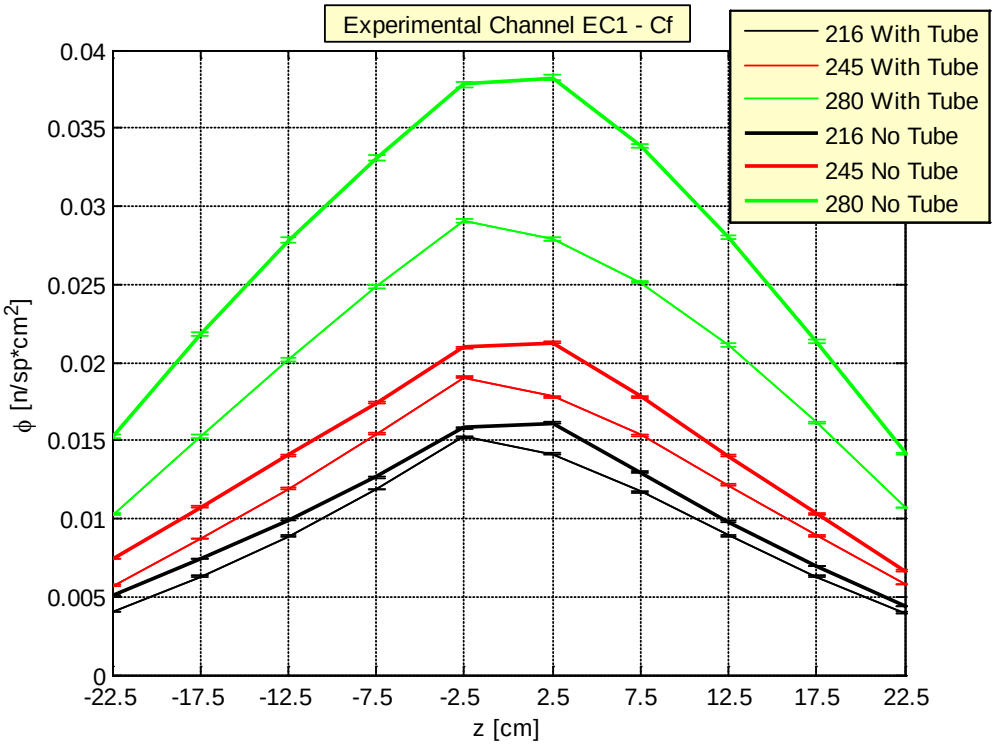

Figure 51. Neutron flux in the EC1 experimental channel calculated by MCNPX with Cf neutron source

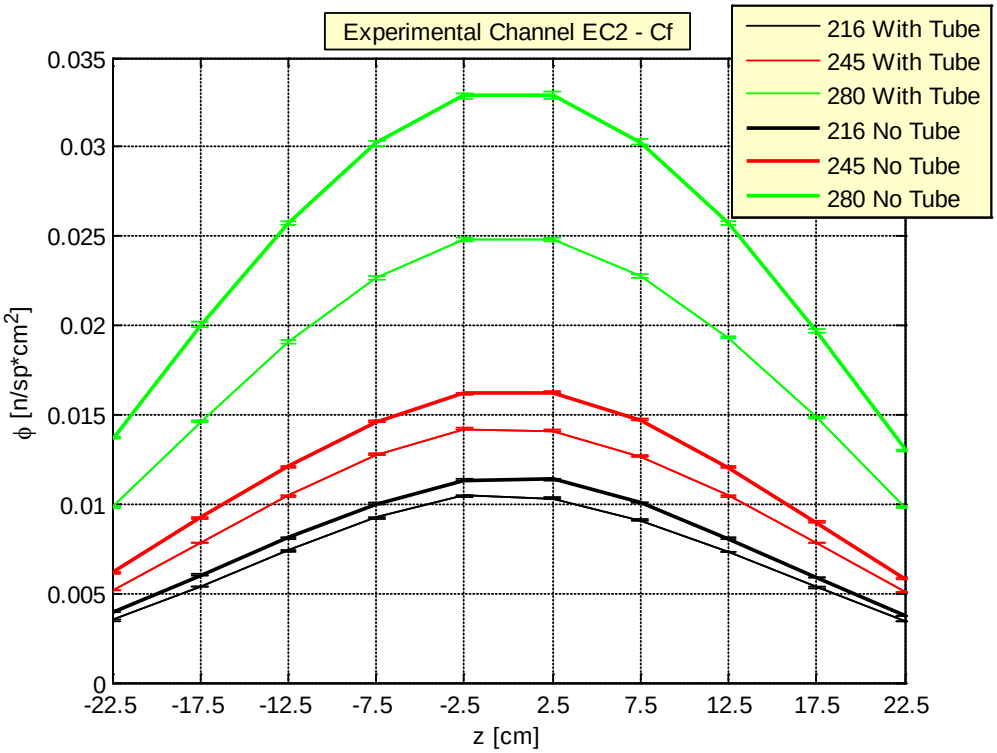

Figure 52. Neutron flux in the EC2 experimental channel calculated by MCNPX with Cf neutron source

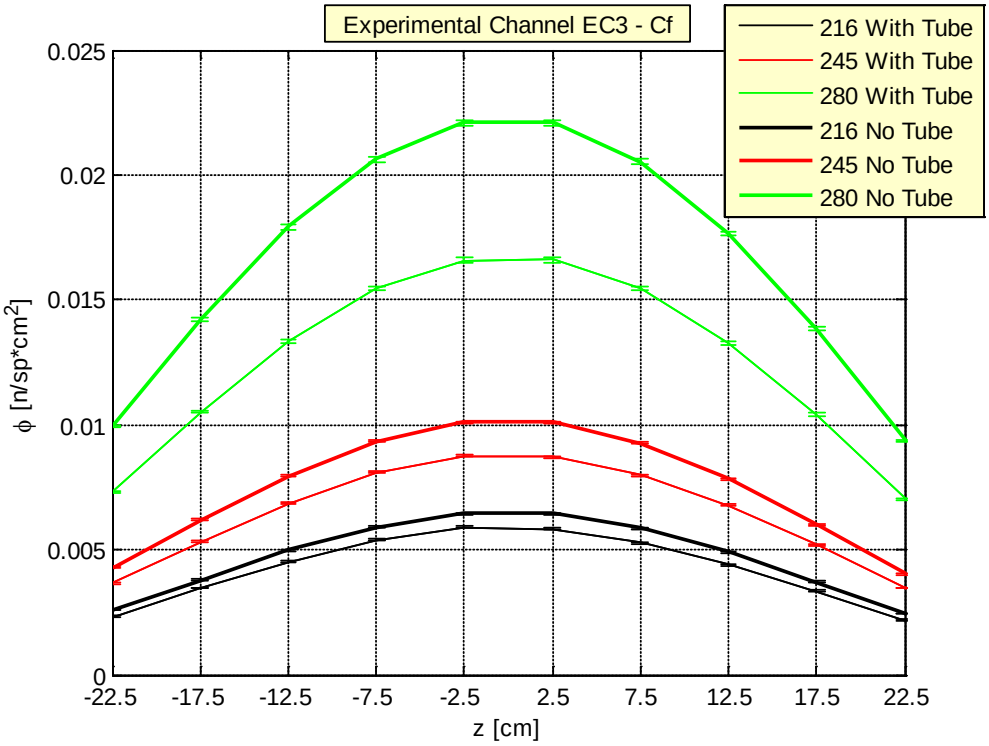

Figure 53. Neutron flux in the EC3 experimental channel calculated by MCNPX with Cf neutron source

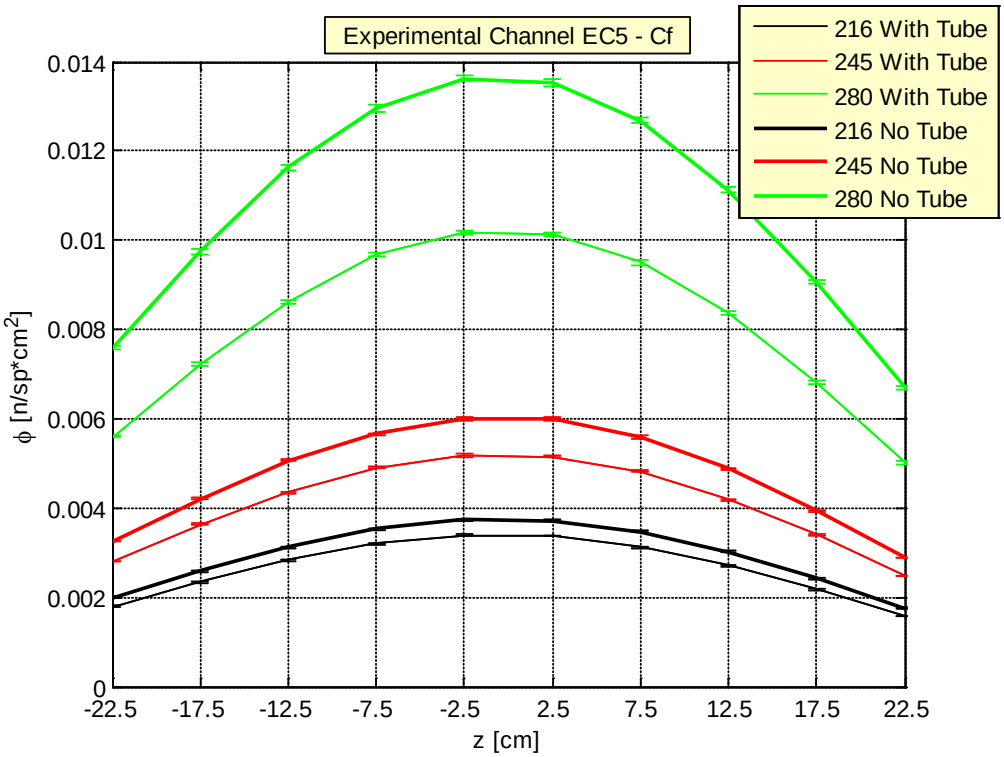

Figure 54. Neutron flux in the EC5 experimental channel calculated by MCNPX with Cf neutron source

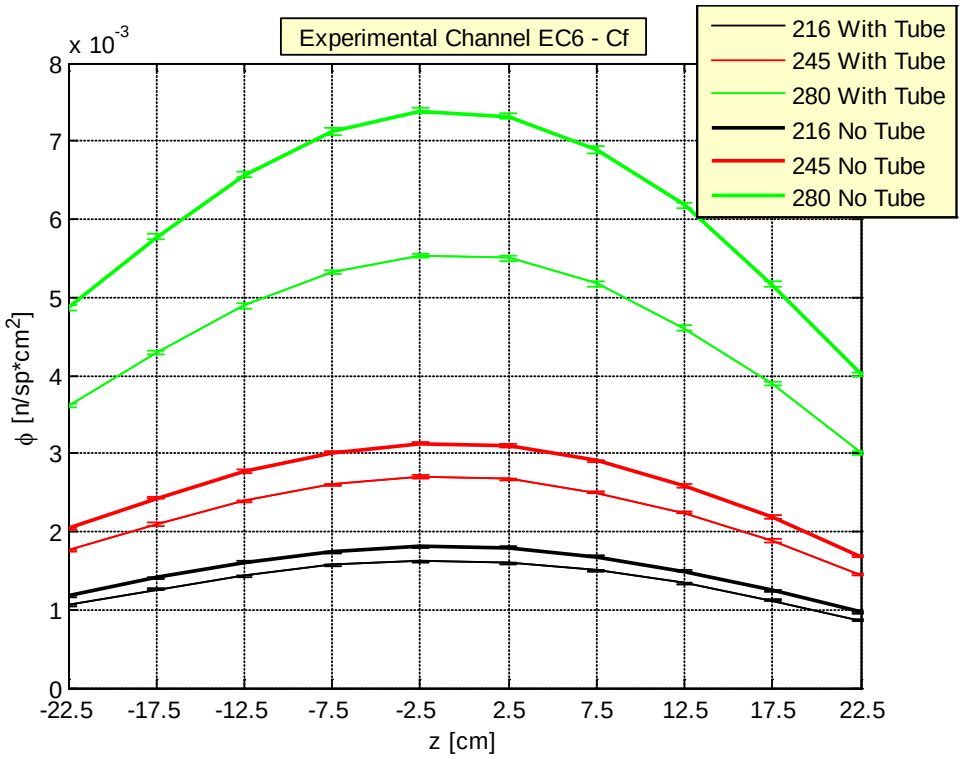

Figure 55. Neutron flux in the EC6 experimental channel calculated by MCNPX with Cf neutron source

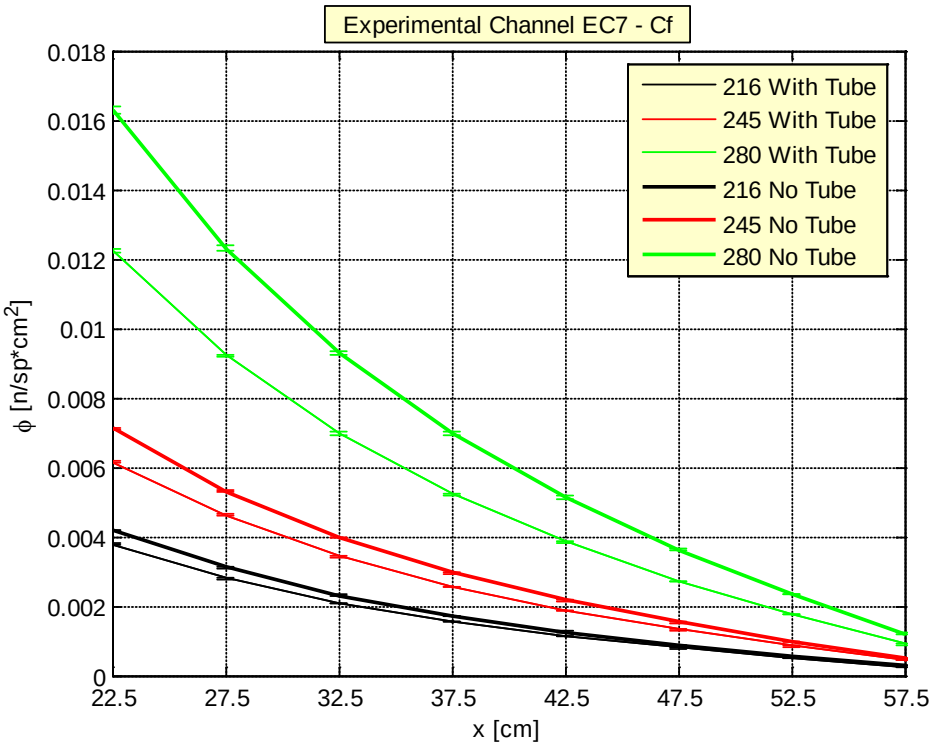

Figure 56. Neutron flux in the EC7 experimental channel calculated by MCNPX with Cf neutron source

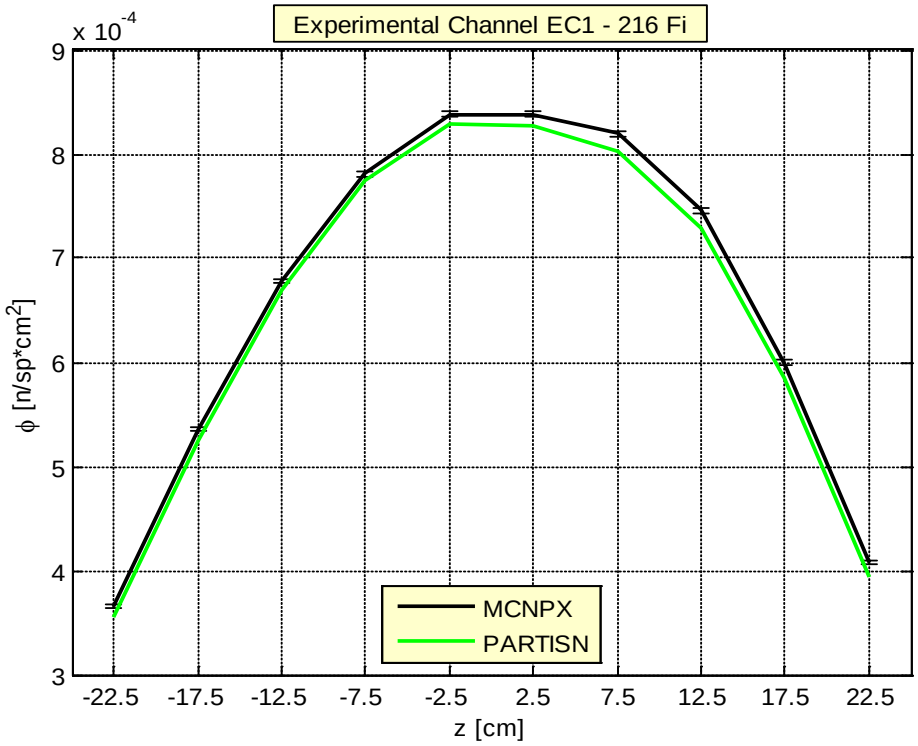

Figure 57. Neutron flux in the EC1 experimental channel from MCNPX and PARTISN criticality calculation for the YALINA thermal with 216 fuel rods

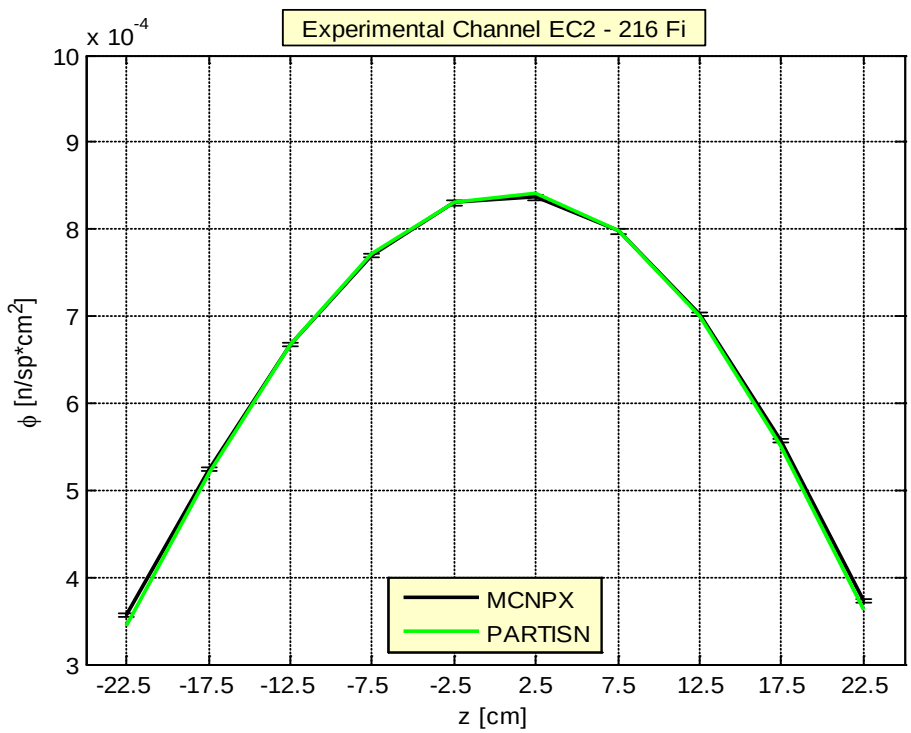

Figure 58. Neutron flux in the EC2 experimental channel from MCNPX and PARTISN criticality calculation for the YALINA thermal with 216 fuel rods

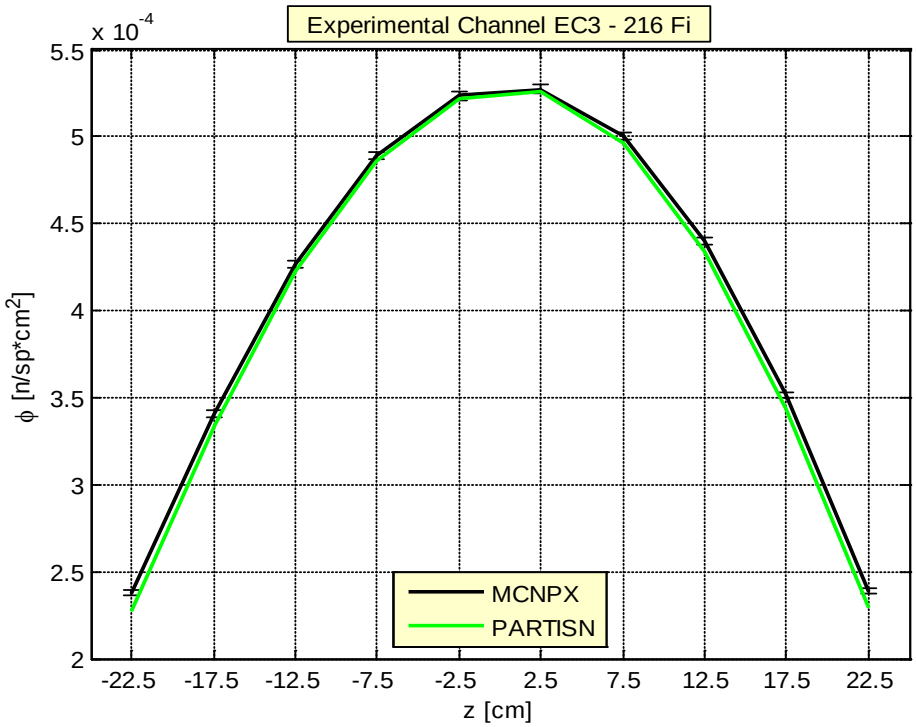

Figure 59. Neutron flux in the EC3 experimental channel from MCNPX and PARTISN criticality calculation for the YALINA thermal with 216 fuel rods

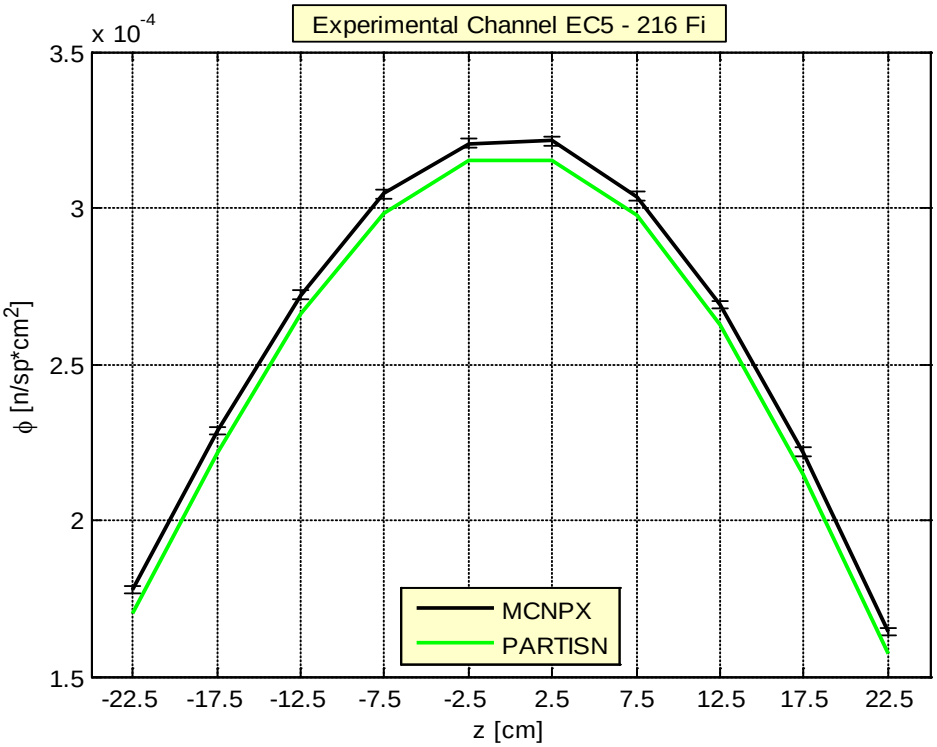

Figure 60. Neutron flux in the EC5 experimental channel from MCNPX and PARTISN criticality calculation for the YALINA thermal with 216 fuel rods

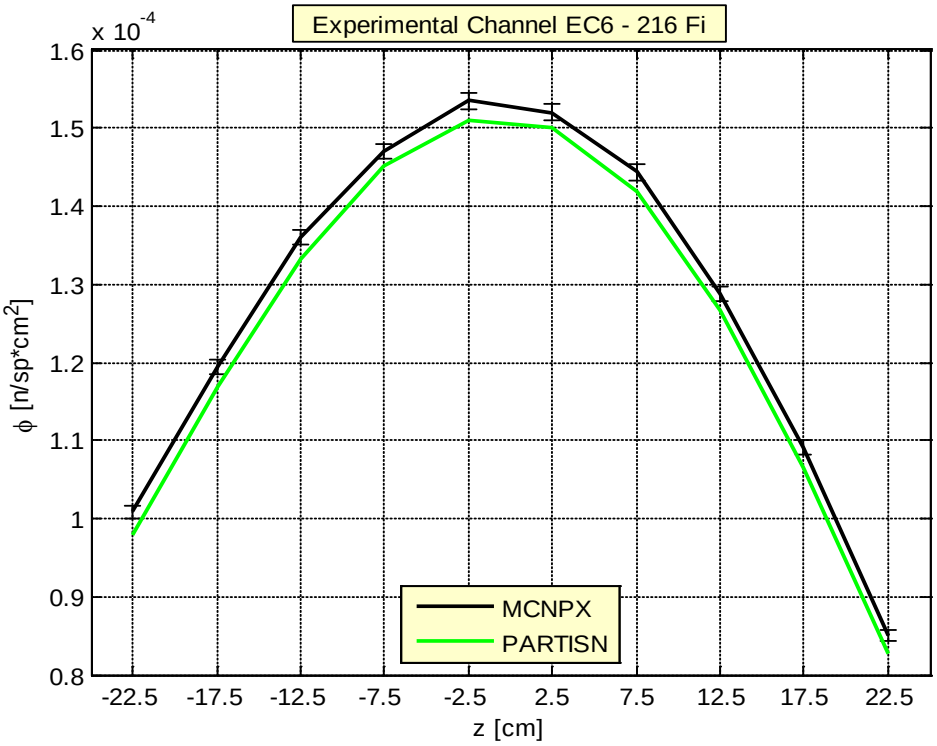

Figure 61. Neutron flux in the EC6 experimental channel from MCNPX and PARTISN criticality calculation for the YALINA thermal with 216 fuel rods

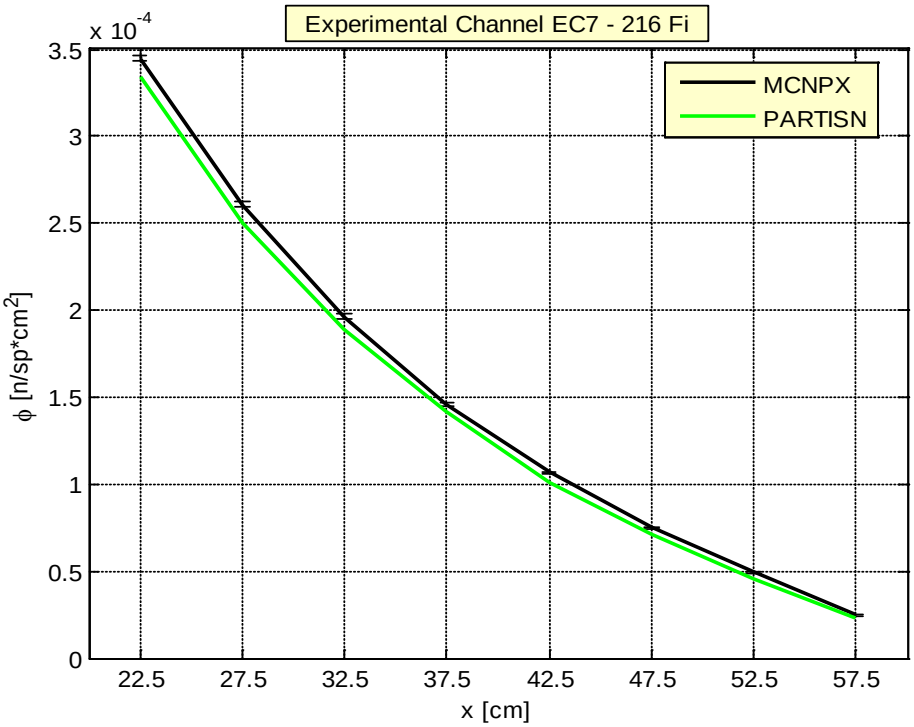

Figure 62. Neutron flux in the EC7 experimental channel from MCNPX and PARTISN criticality calculation for the YALINA thermal with 216 fuel rods

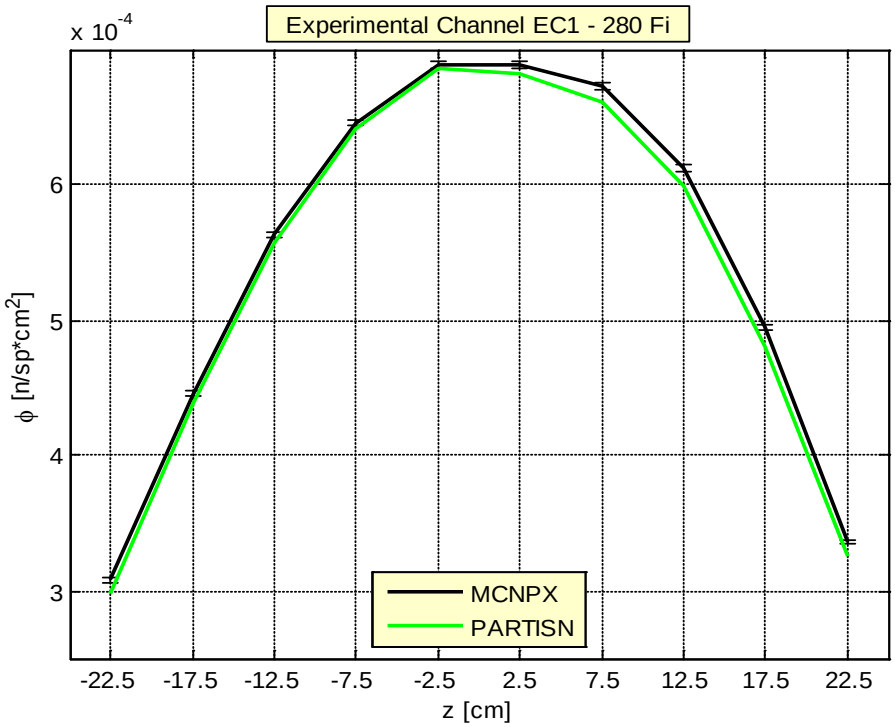

Figure 63. Neutron flux in the EC1 experimental channel from MCNPX and PARTISN criticality calculation for the YALINA thermal with 280 fuel rods

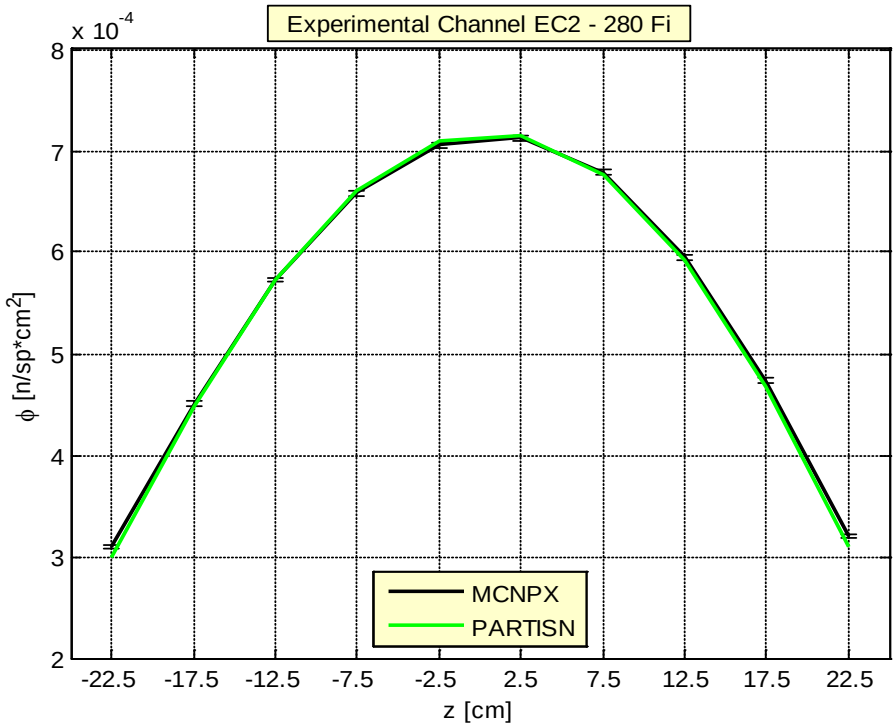

Figure 64. Neutron flux in the EC2 experimental channel from MCNPX and PARTISN criticality calculation for the YALINA thermal with 280 fuel rods

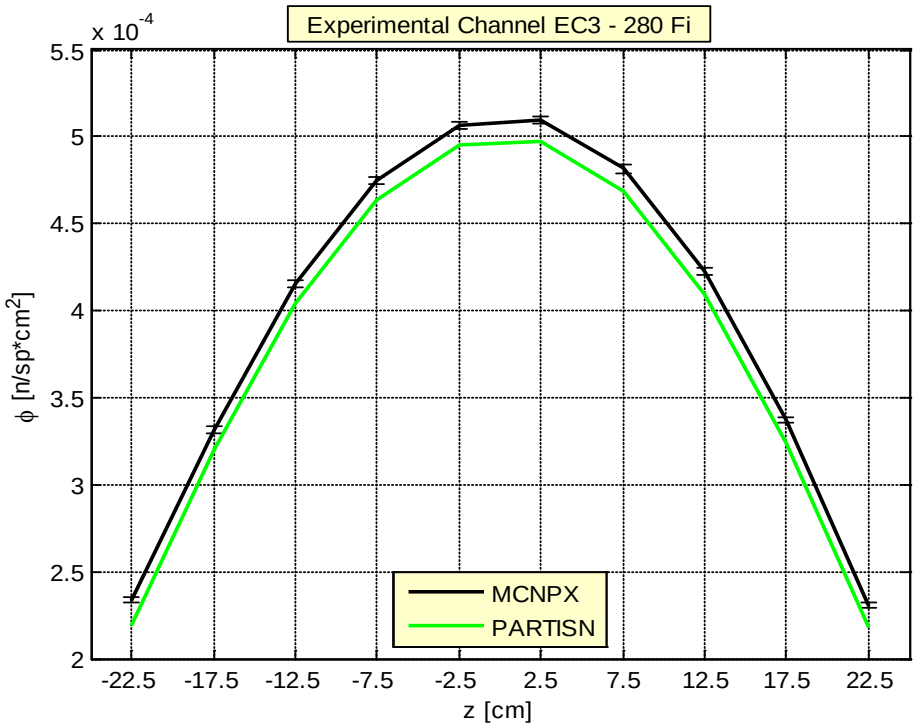

Figure 65. Neutron flux in the EC3 experimental channel from MCNPX and PARTISN criticality calculation for the YALINA thermal with 280 fuel rods

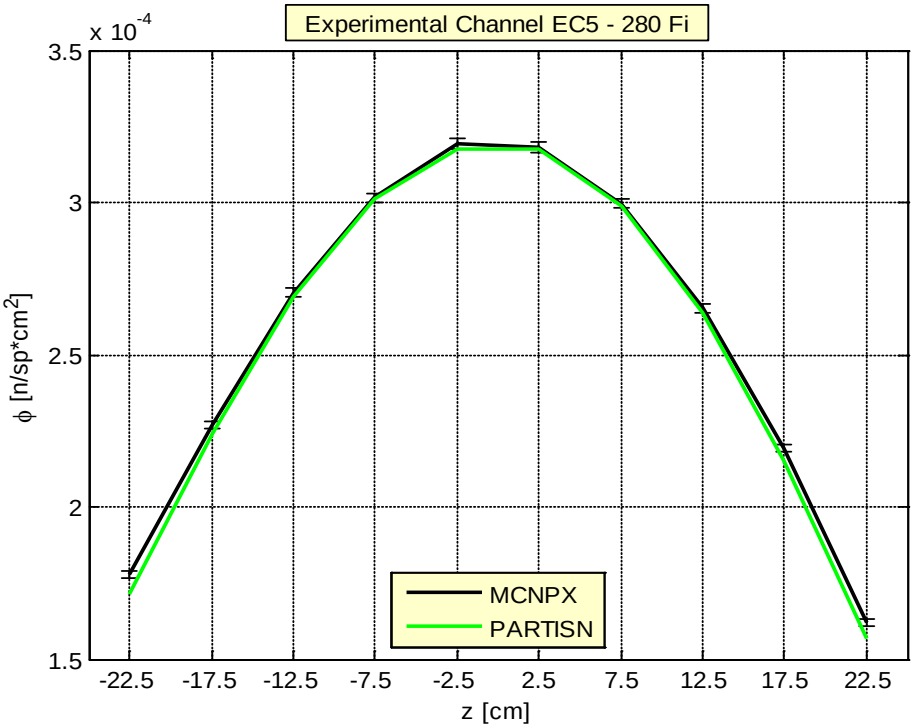

Figure 66. Neutron flux in the EC5 experimental channel from MCNPX and PARTISN criticality calculation for the YALINA thermal with 280 fuel rods

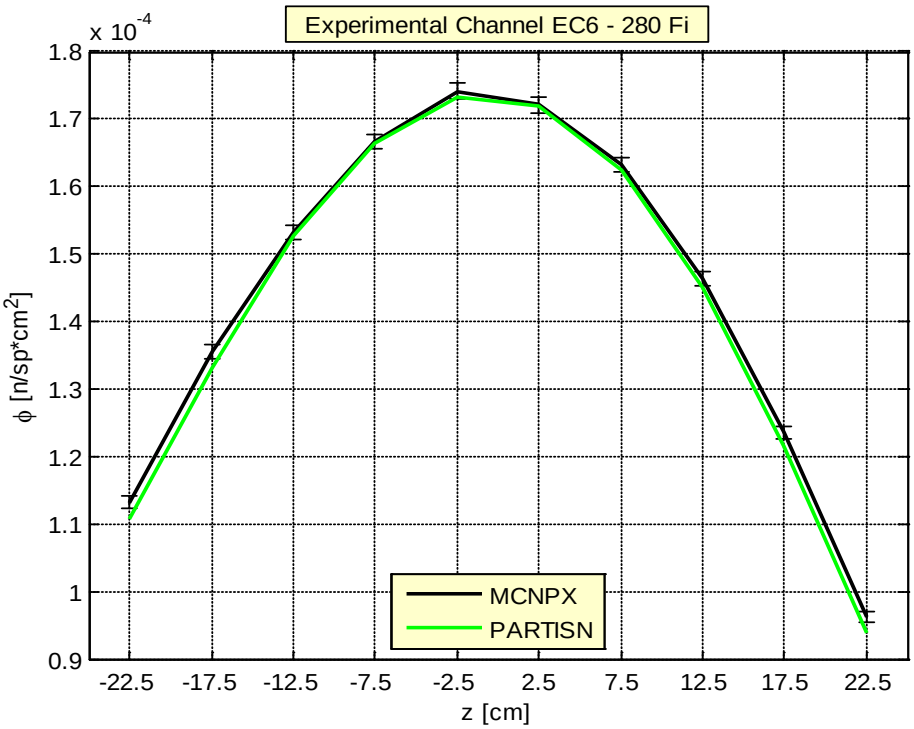

Figure 67. Neutron flux in the EC6 experimental channel from MCNPX and PARTISN criticality calculation for the YALINA thermal with 280 fuel rods

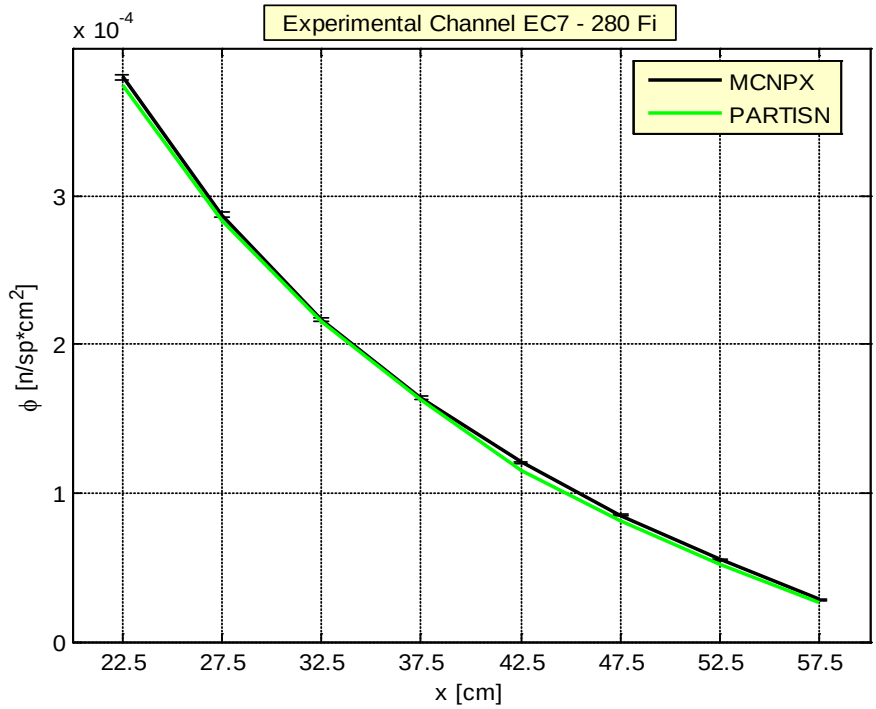

Figure 68. Neutron flux in the EC7 experimental channel from MCNPX and PARTISN criticality calculation for the YALINA thermal with 280 fuel rods

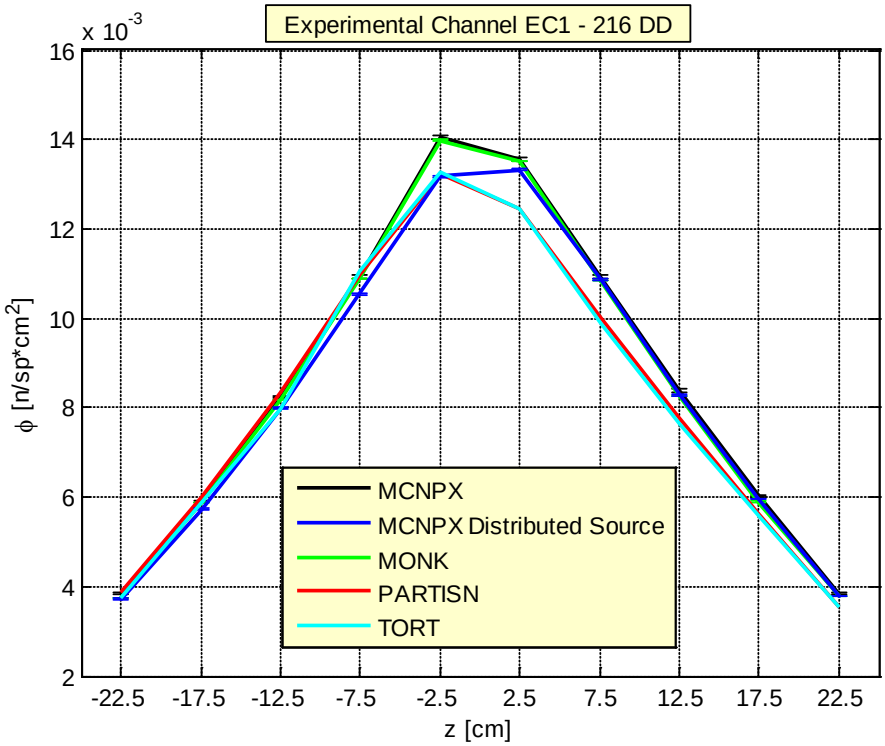

Figure 69. Neutron flux in the EC1 experimental channel calculated by MCNPX, MONK, PARTISN, and TORT for the YALINA thermal with 216 fuel rods and D-D neutron source

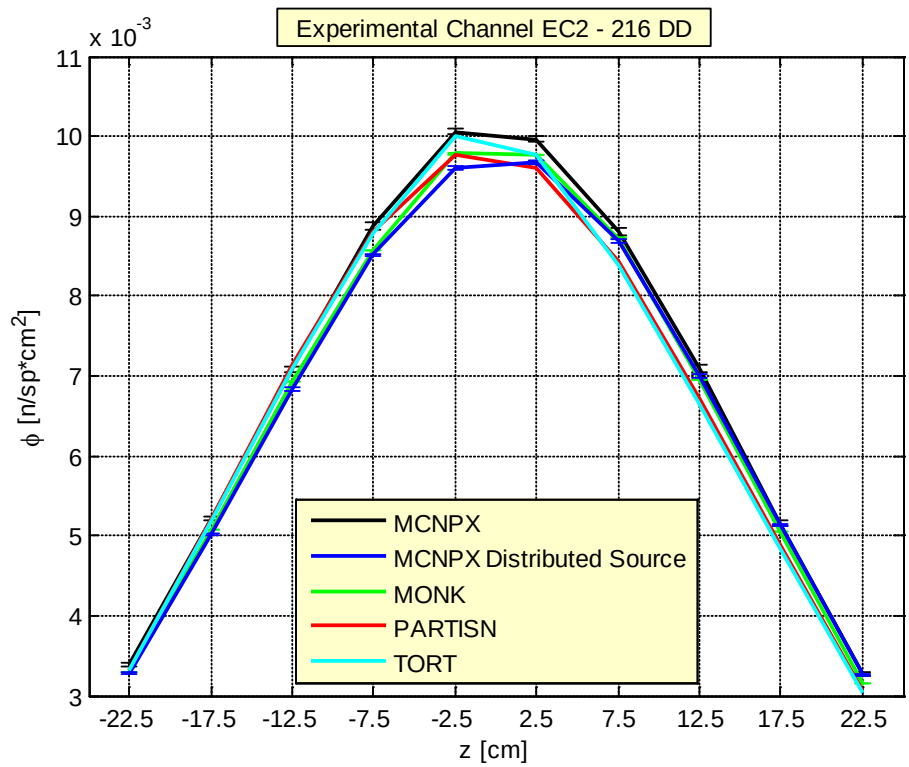

Figure 70. Neutron flux in the EC2 experimental channel calculated by MCNPX, MONK, PARTISN, and TORT for the YALINA thermal with 216 fuel rods and D-D neutron source

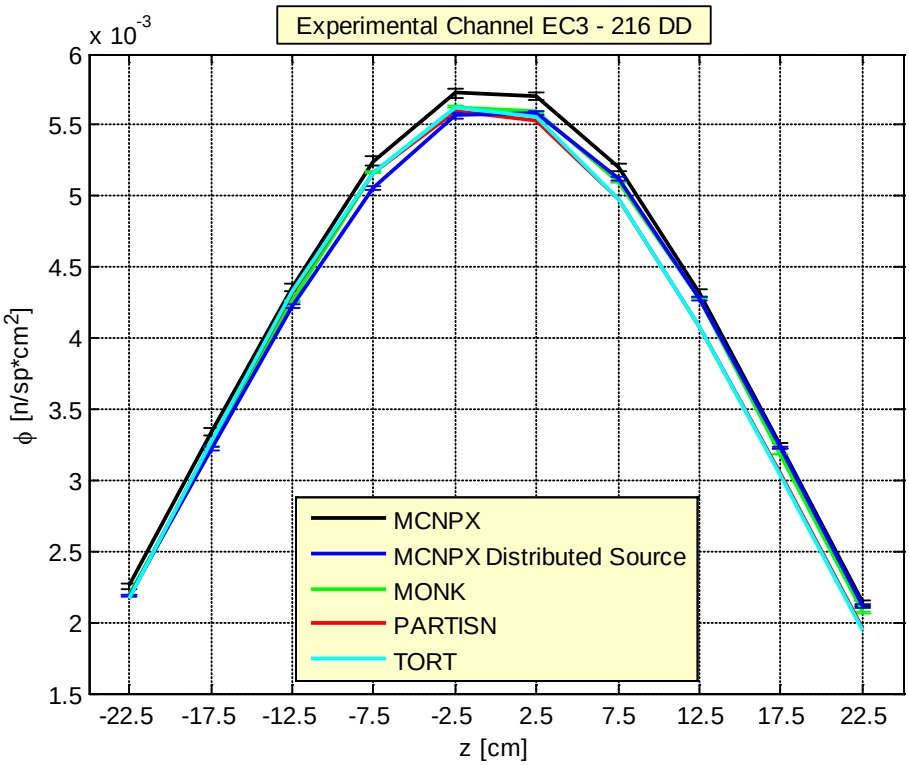

Figure 71. Neutron flux in the EC3 experimental channel calculated by MCNPX, MONK, PARTISN, and TORT for the YALINA thermal with 216 fuel rods and D-D neutron source

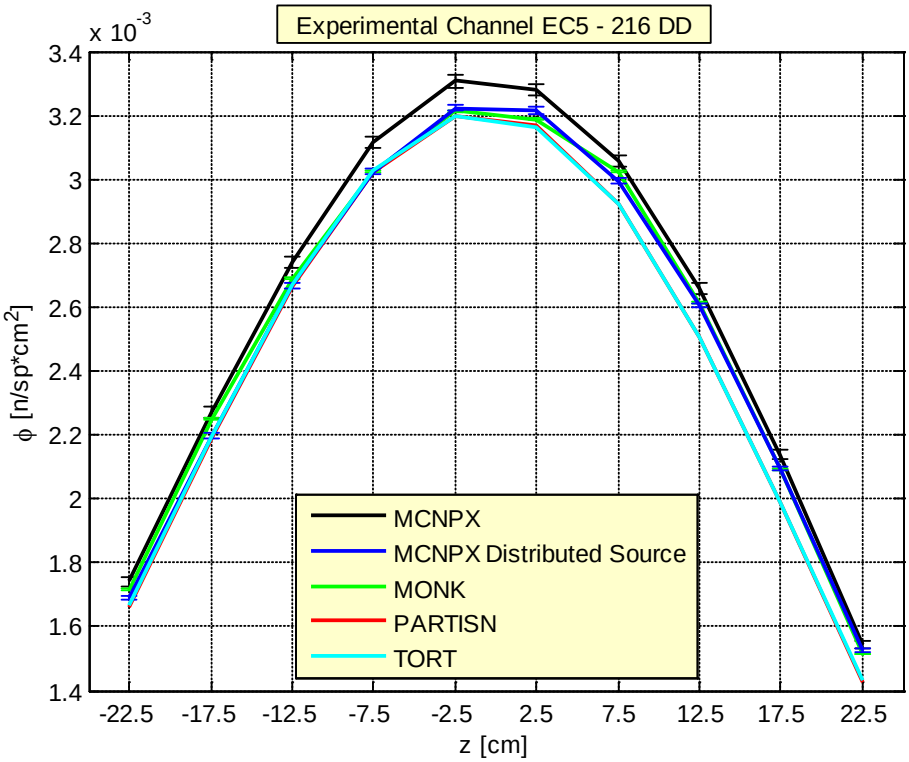

Figure 72. Neutron flux in the EC5 experimental channel calculated by MCNPX, MONK, PARTISN, and TORT for the YALINA thermal with 216 fuel rods and D-D neutron source

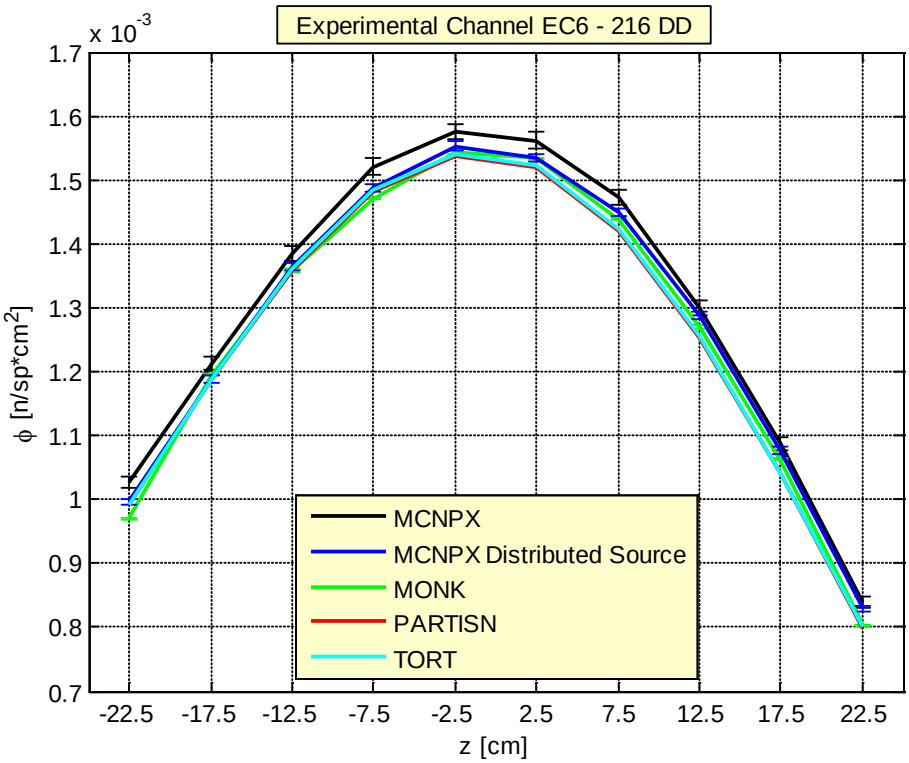

Figure 73. Neutron flux in the EC6 experimental channel calculated by MCNPX, MONK, PARTISN, and TORT for the YALINA thermal with 216 fuel rods and D-D neutron source

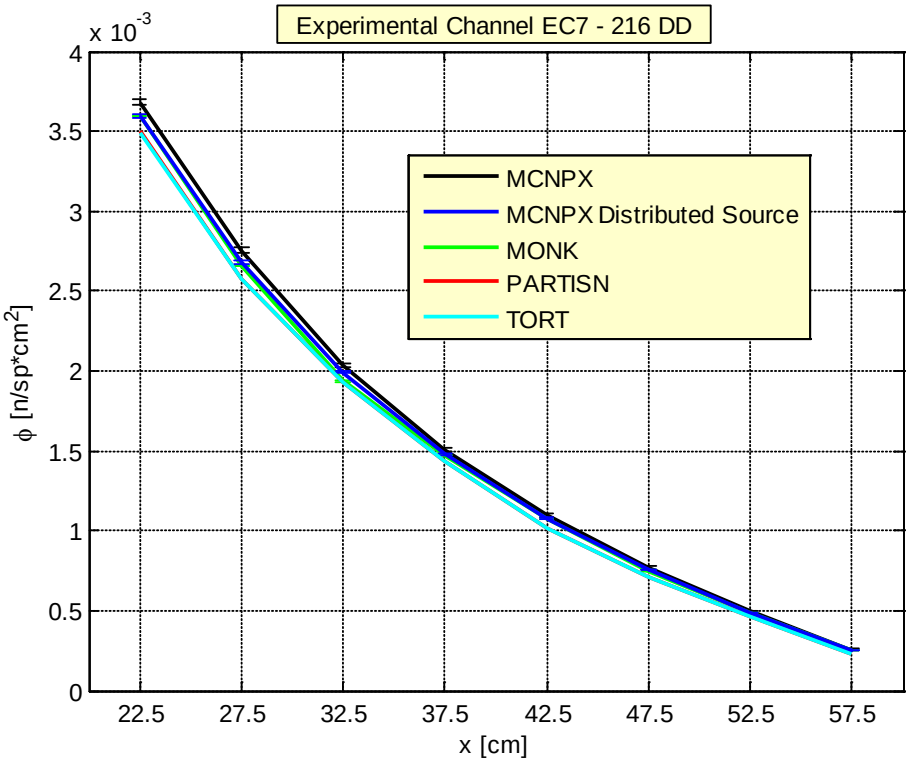

Figure 74. Neutron flux in the EC7 experimental channel calculated by MCNPX, MONK, PARTISN, and TORT for the YALINA thermal with 216 fuel rods and D-D neutron source

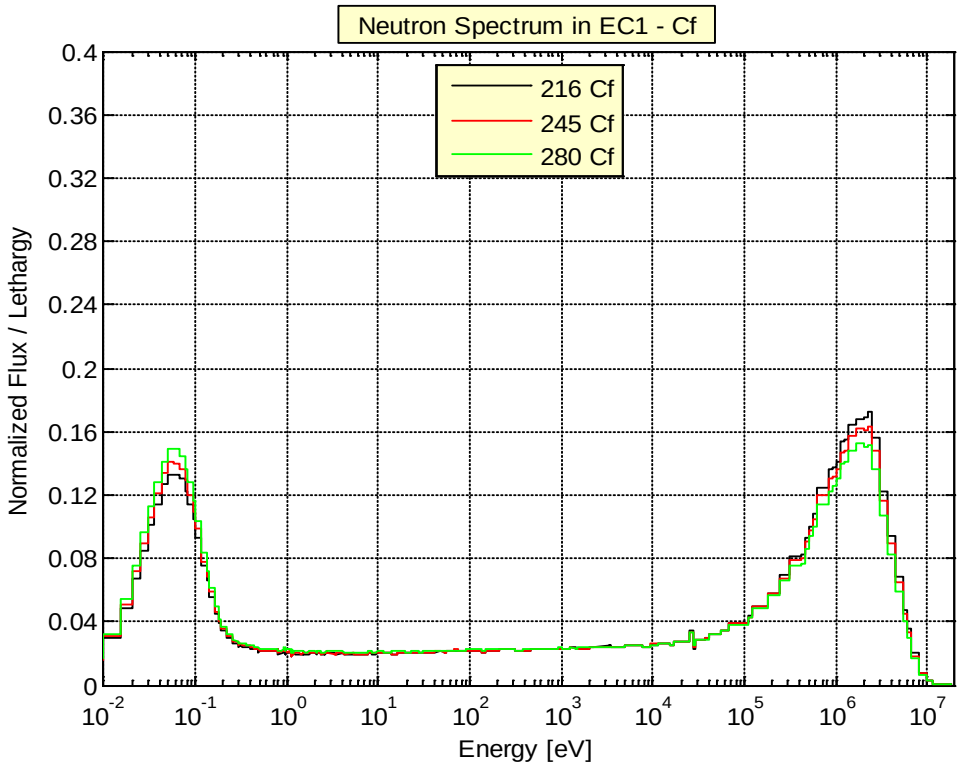

Figure 75. Neutron spectra in the EC1 experimental channel calculated by MCNPX with Cf neutron source

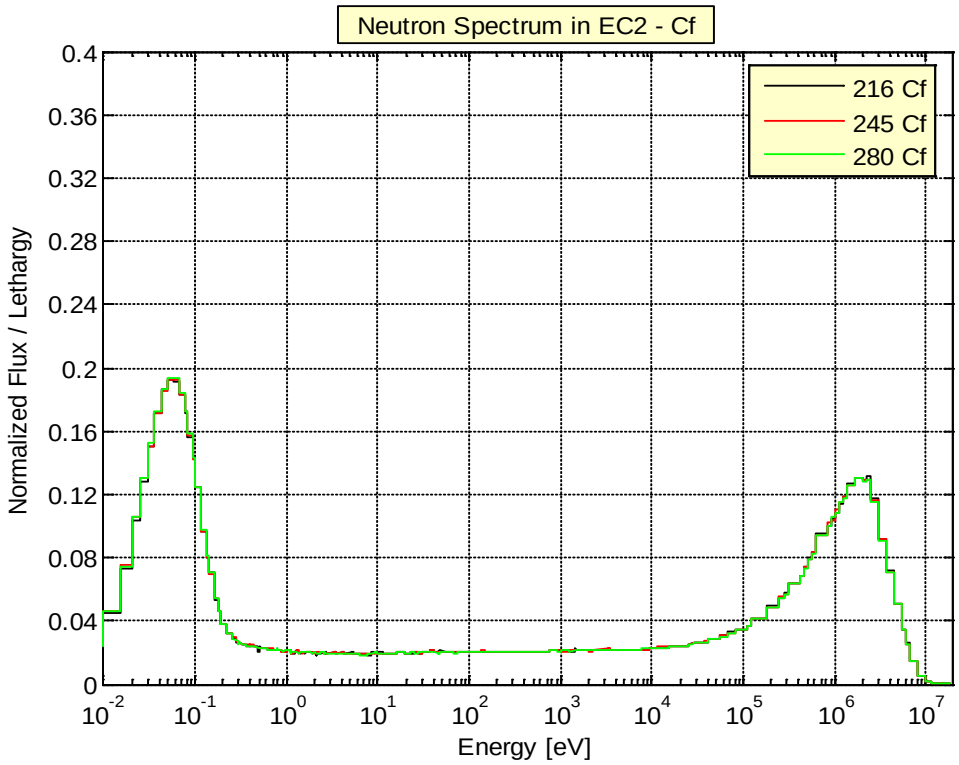

Figure 76. Neutron spectra in the EC2 experimental channel calculated by MCNPX with Cf neutron source

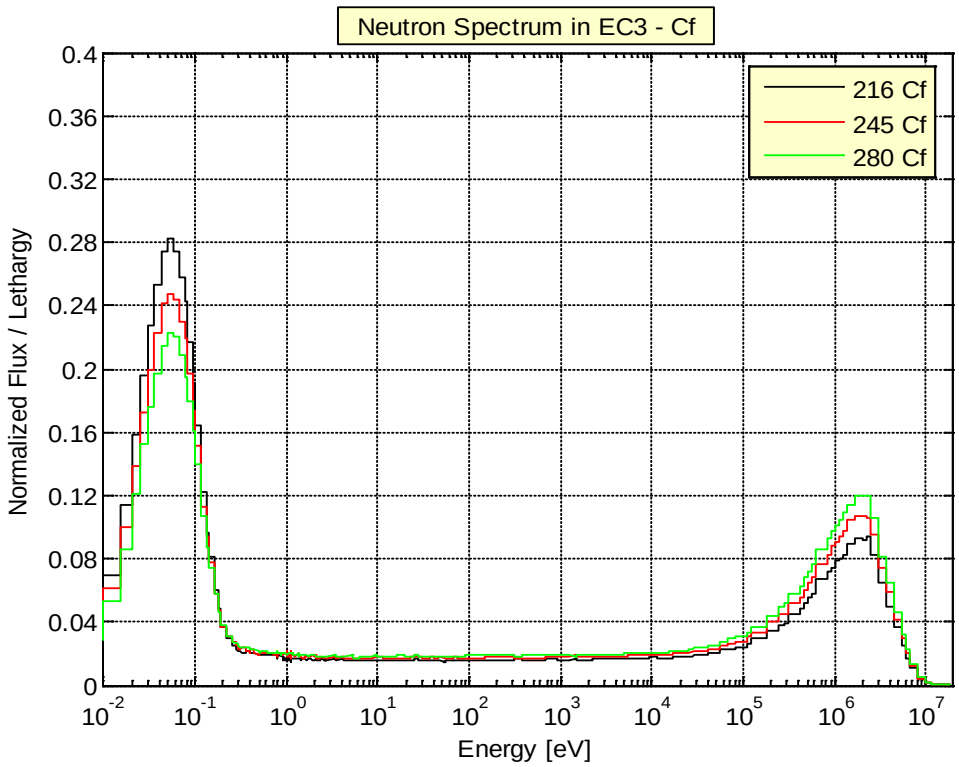

Figure 77. Neutron spectra in the EC3 experimental channel calculated by MCNPX with Cf neutron source

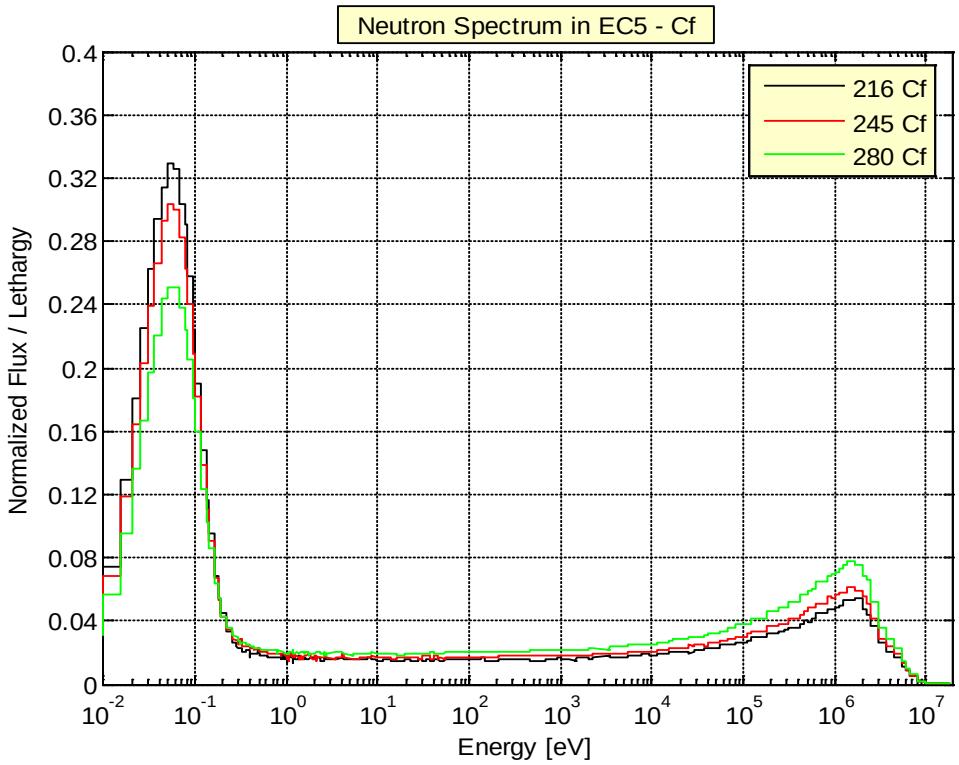

Figure 78. Neutron spectra in the EC5 experimental channel calculated by MCNPX with Cf neutron source

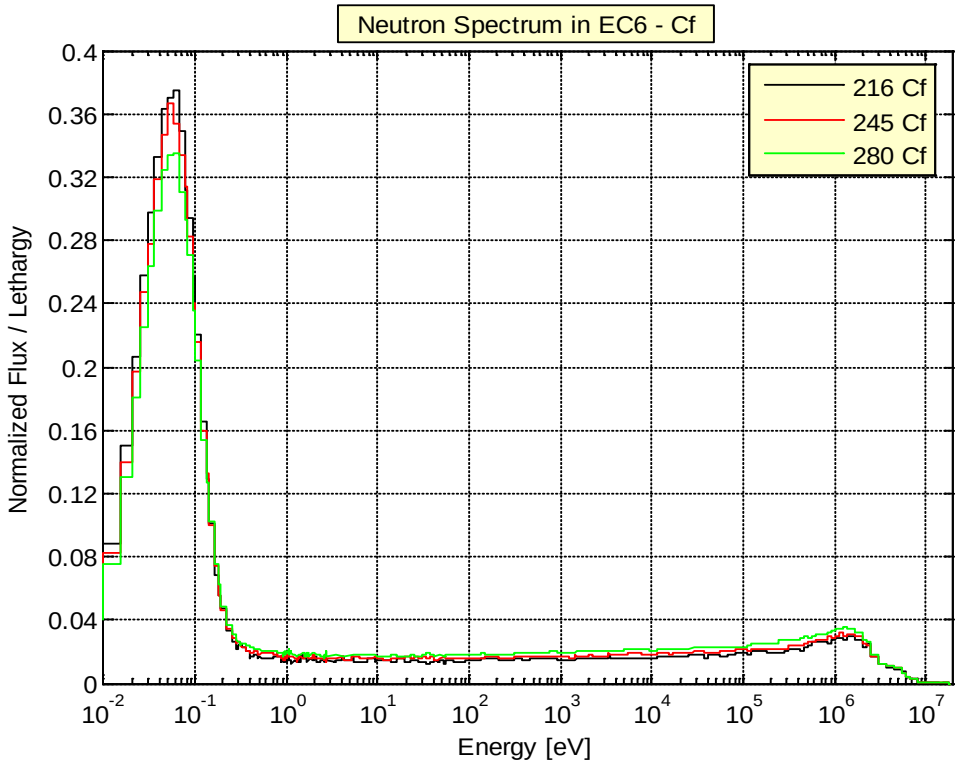

Figure 79. Neutron spectra in the EC6 experimental channel calculated by MCNPX with Cf neutron source

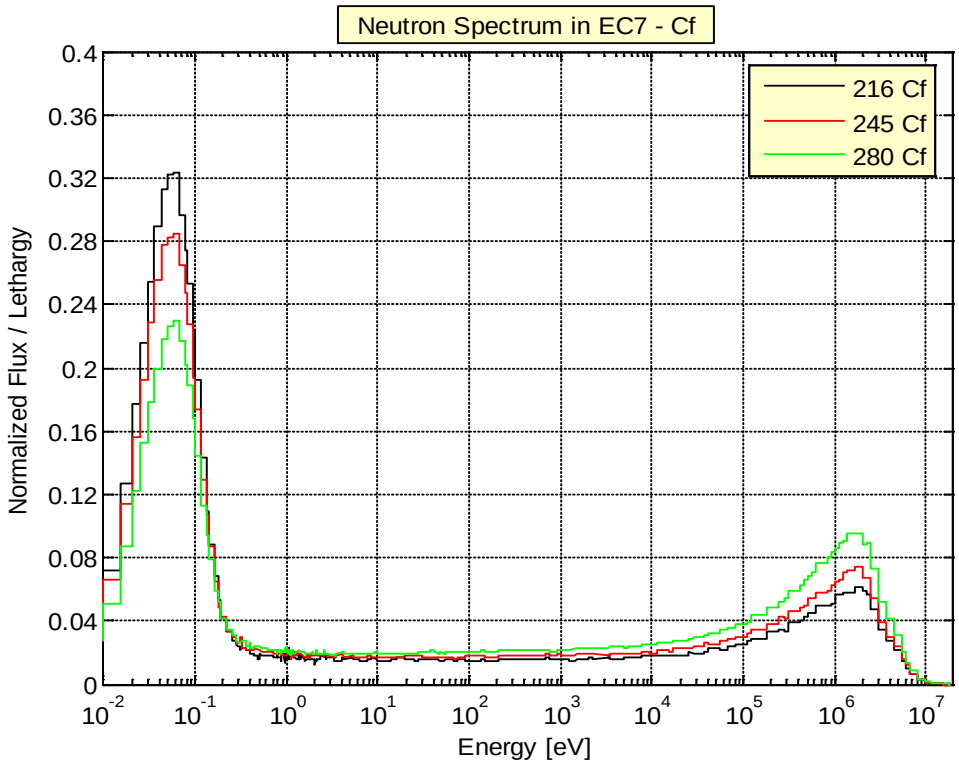

Figure 80. Neutron spectra in the EC7 experimental channel calculated by MCNPX with Cf neutron source

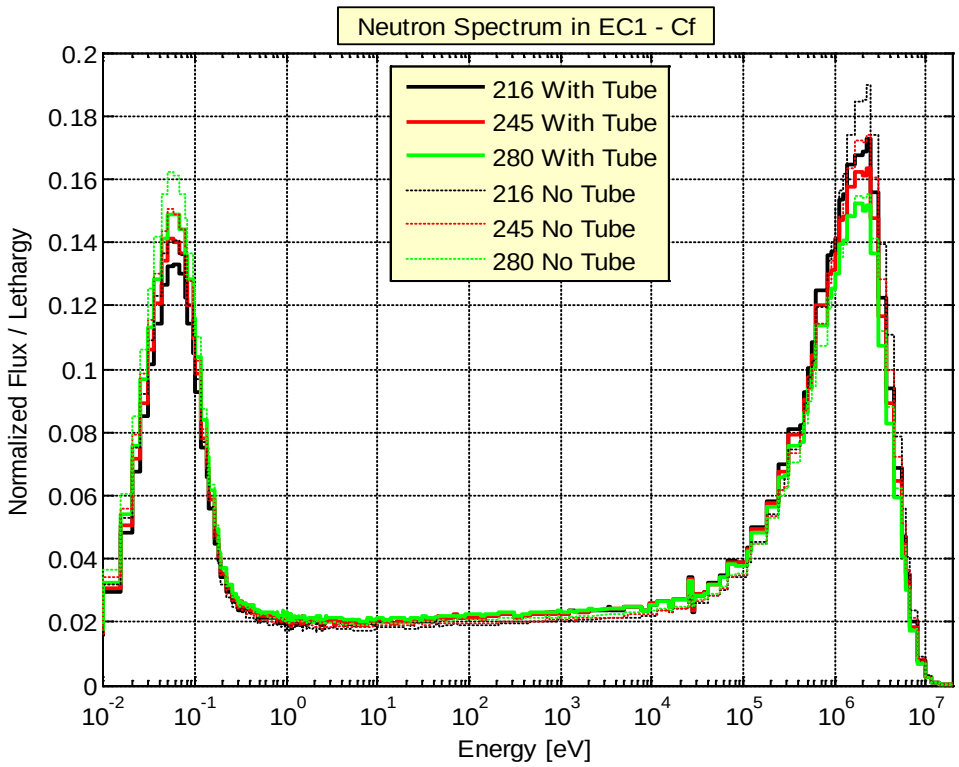

Figure 81. Neutron spectra in the EC1 experimental channel calculated by MCNPX with Cf neutron source and with and without the beam tube

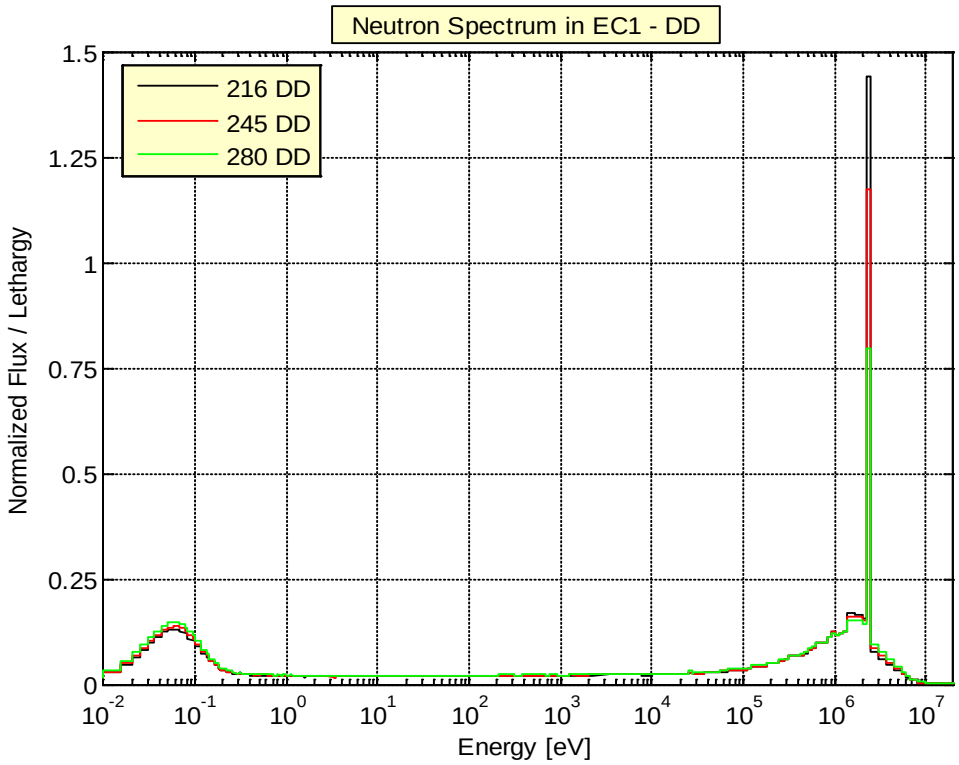

Figure 82. Neutron spectra in the EC1 experimental channel calculated by MCNPX with D-D neutron source

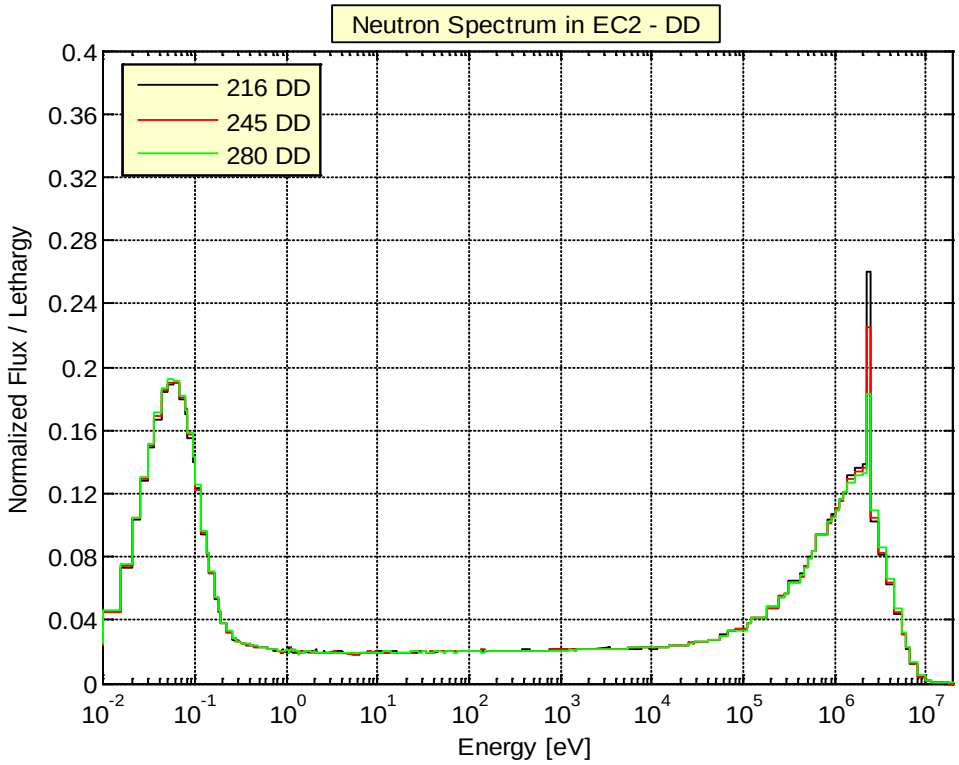

Figure 83. Neutron spectra in the EC2 experimental channel calculated by MCNPX with D-D neutron source

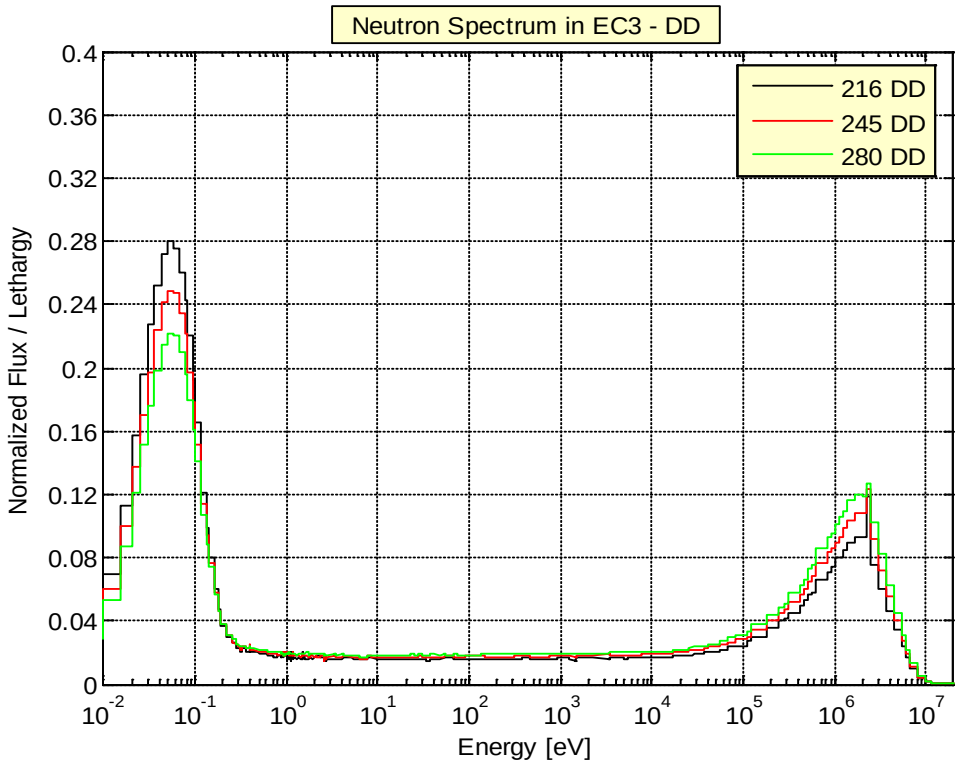

Figure 84. Neutron spectra in the EC3 experimental channel calculated by MCNPX with D-D neutron source

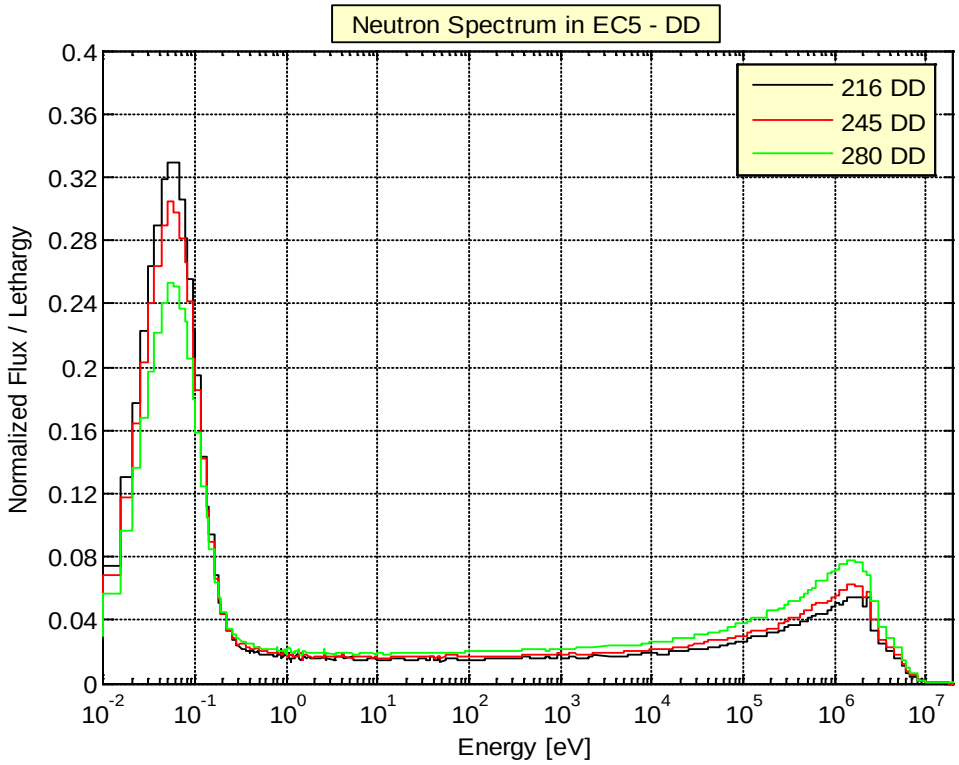

Figure 85. Neutron spectra in the EC5 experimental channel calculated by MCNPX with D-D neutron source

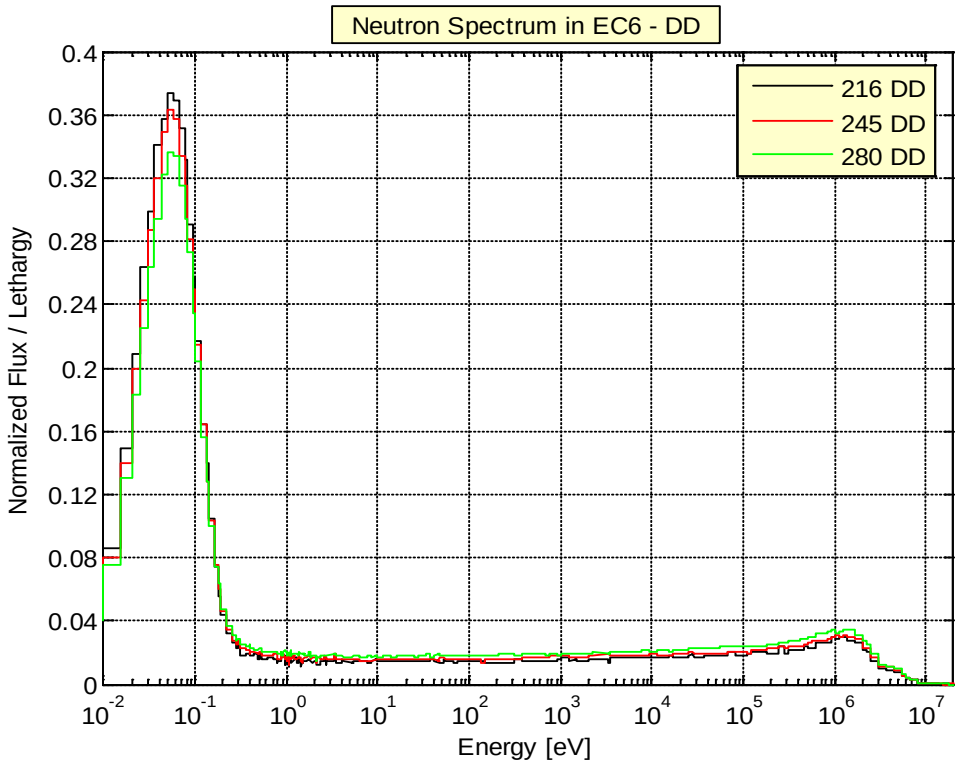

Figure 86. Neutron spectra in the EC6 experimental channel calculated by MCNPX with D-D neutron source

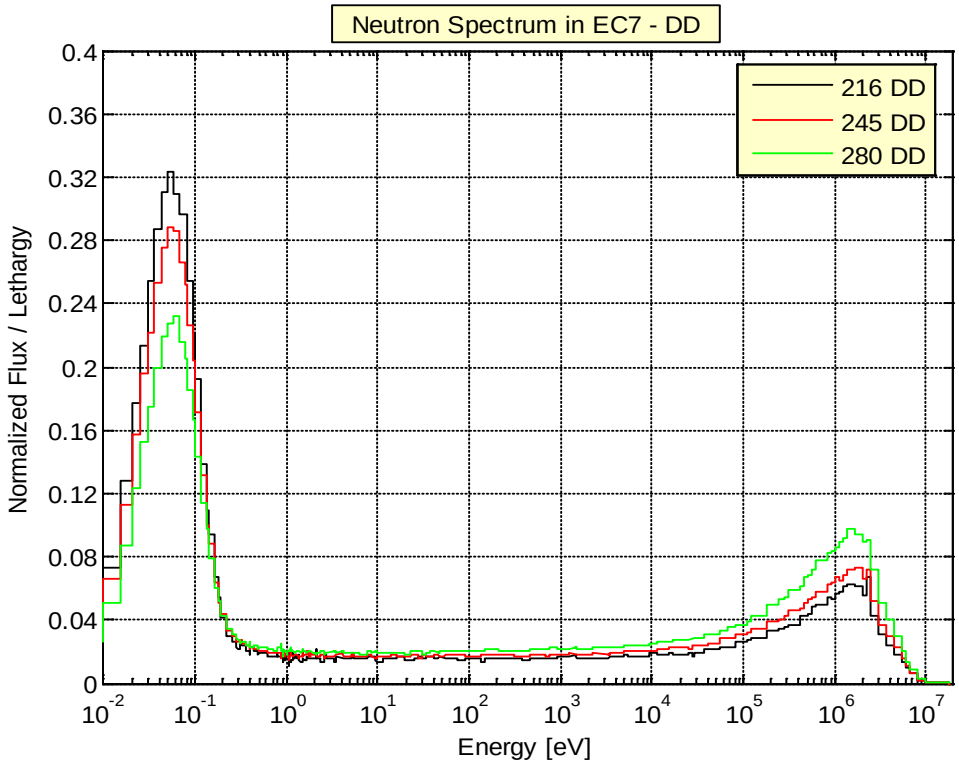

Figure 87. Neutron spectra in the EC7 experimental channel calculated by MCNPX with D-D neutron source

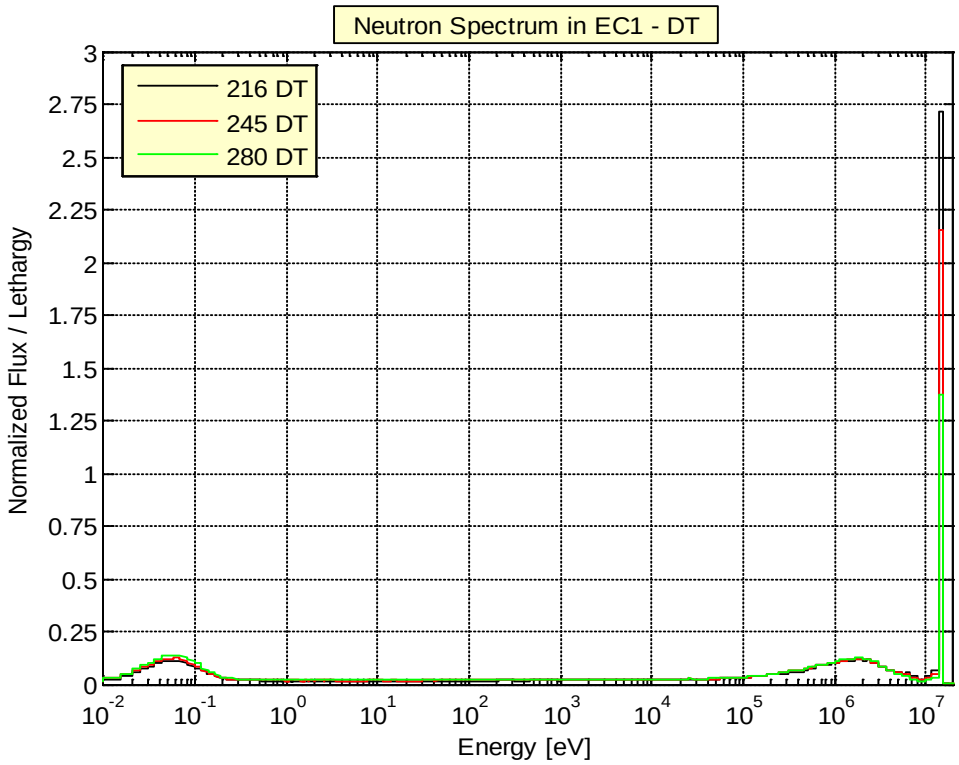

Figure 88. Neutron spectra in the EC1 experimental channel calculated by MCNPX with D-T neutron source

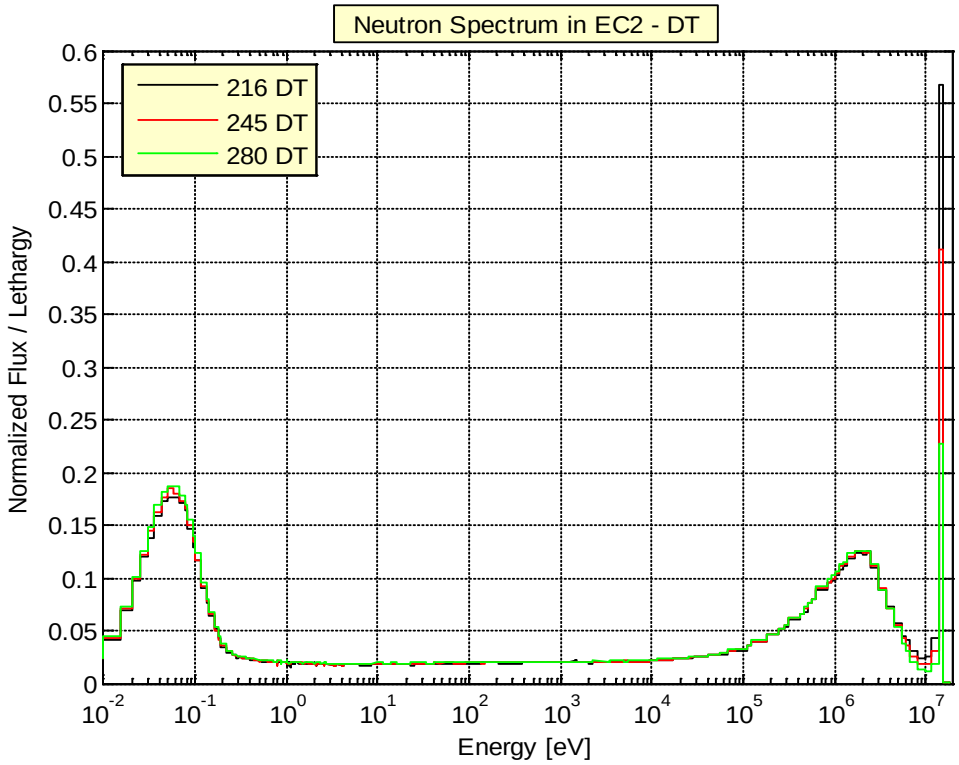

Figure 89. Neutron spectra in the EC2 experimental channel calculated by MCNPX with D-T neutron source

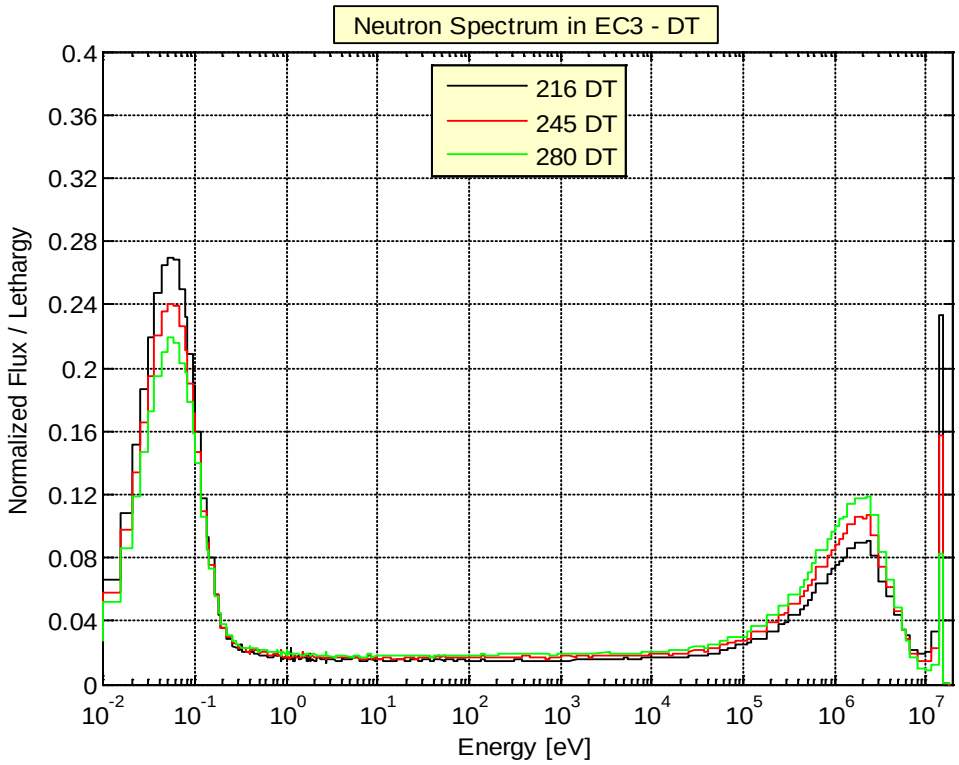

Figure 90. Neutron spectra in the EC3 experimental channel calculated by MCNPX with D-T neutron source

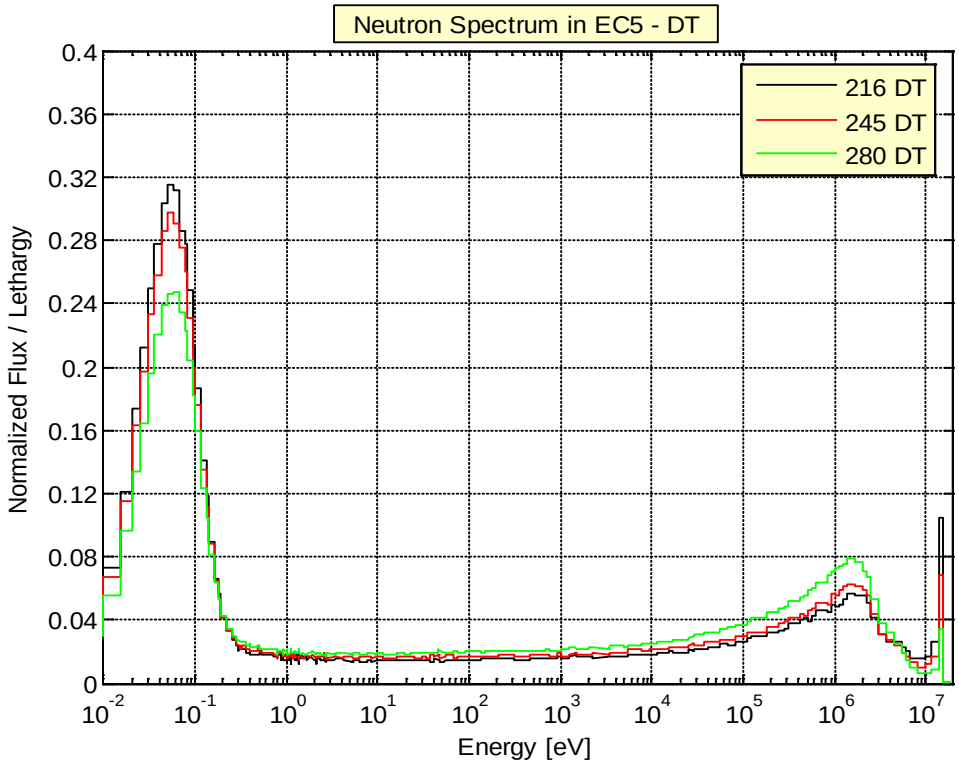

Figure 91. Neutron spectra in the EC5 experimental channel calculated by MCNPX with D-T neutron source

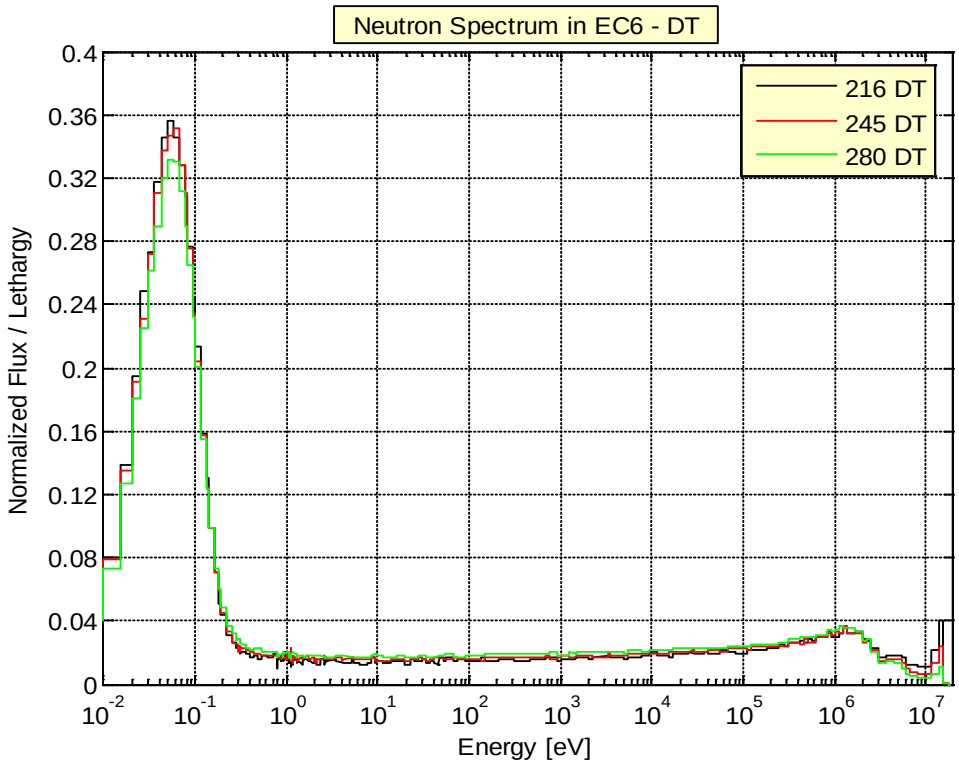

Figure 92. Neutron spectra in the EC6 experimental channel calculated by MCNPX with D-T neutron source

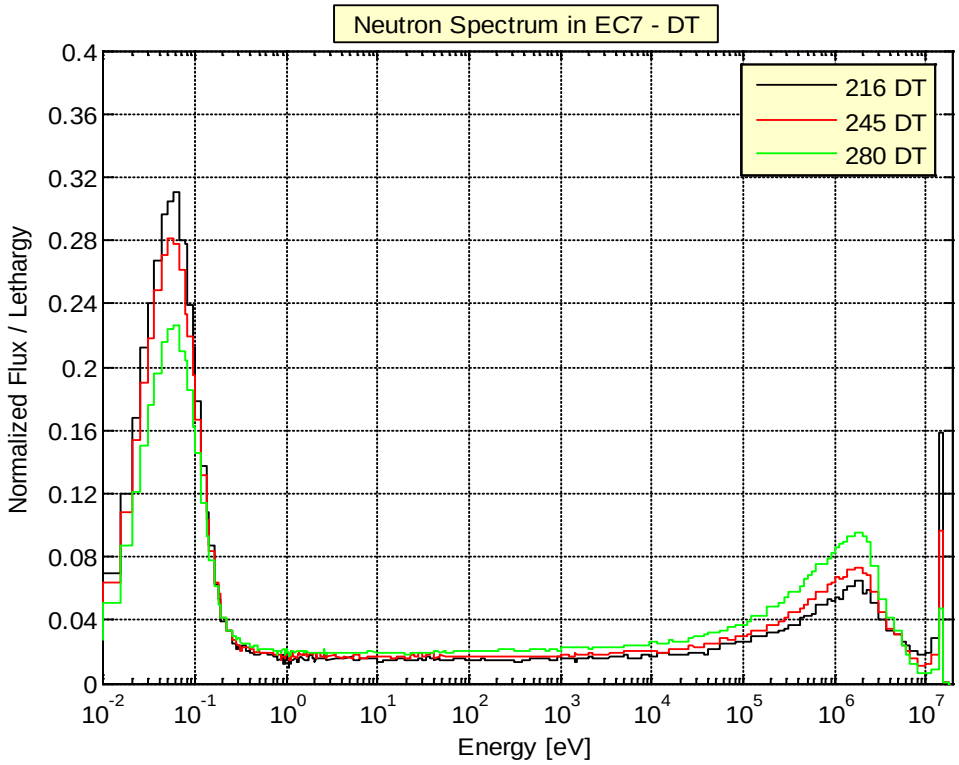

Figure 93. Neutron spectra in the EC7 experimental channel calculated by MCNPX with D-T neutron source

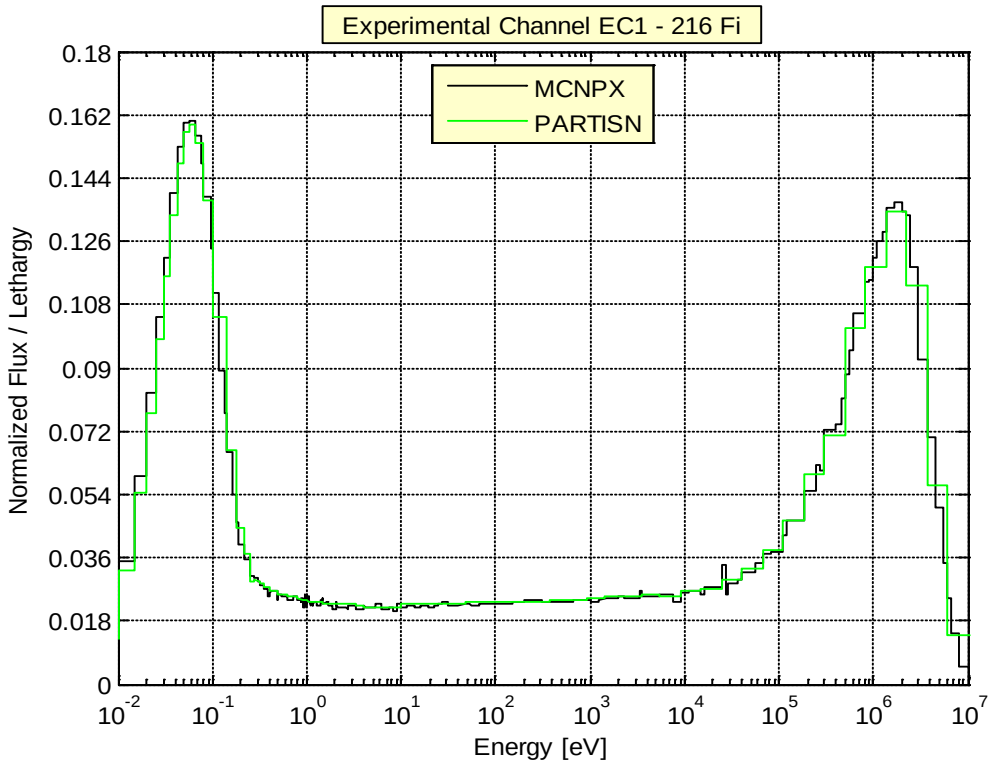

Figure 94. Neutron spectra in the EC1 experimental channel of the YALINA thermal with 216 fuel rods from criticality calculation by MCNPX and PARTISN

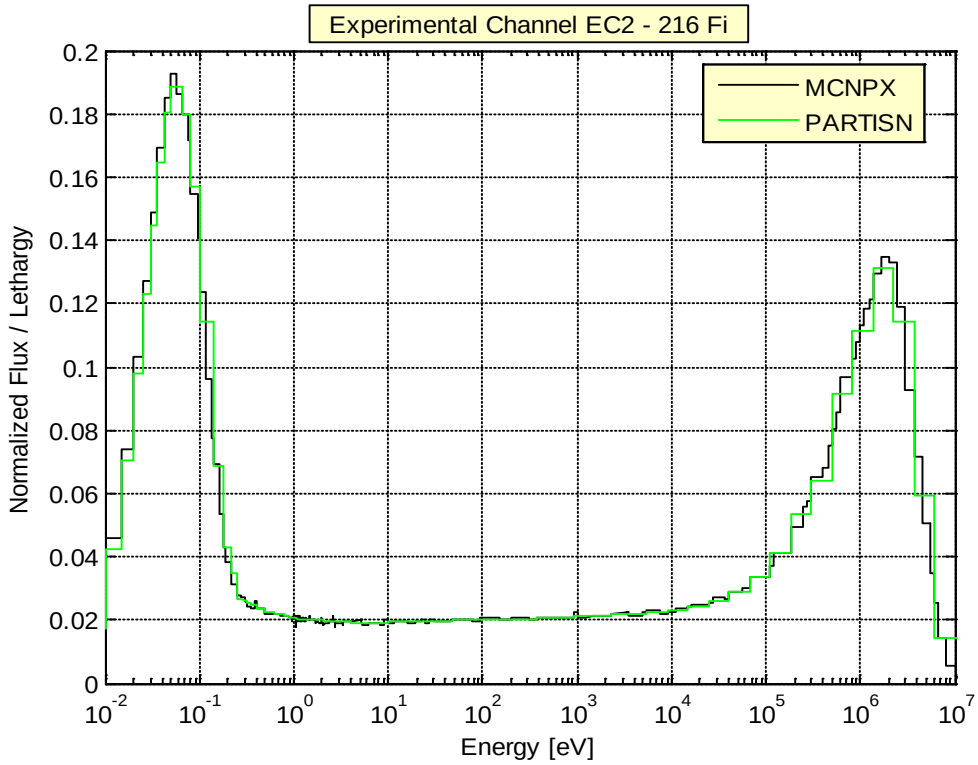

Figure 95. Neutron spectra in the EC2 experimental channel of the YALINA thermal with 216 fuel rods from criticality calculation by MCNPX and PARTISN

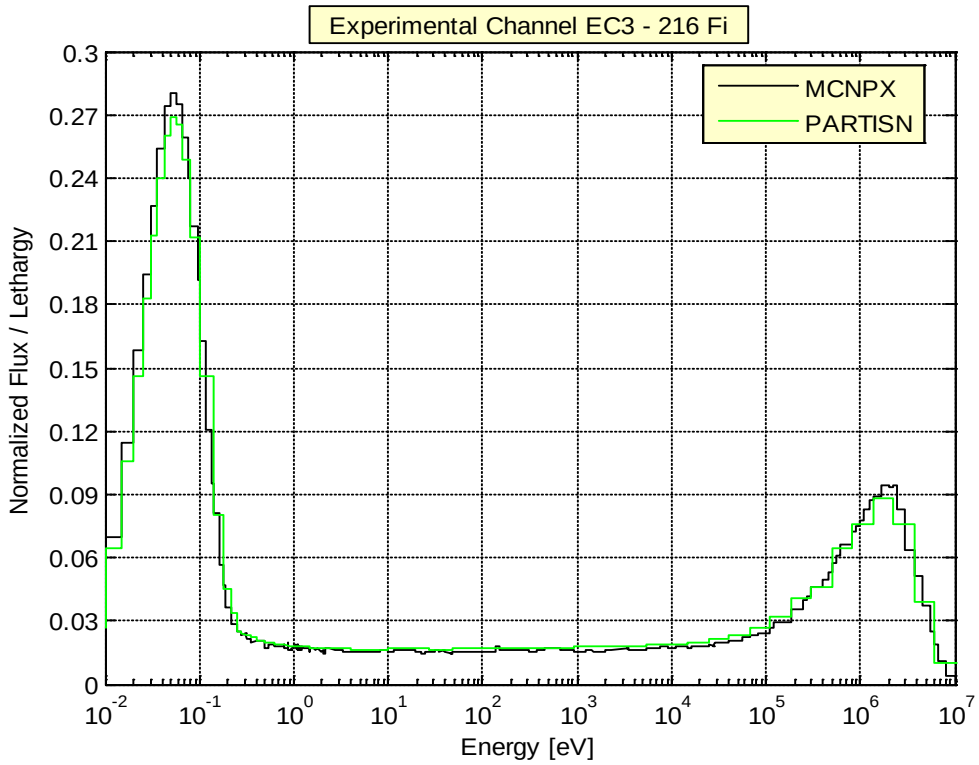

Figure 96. Neutron spectra in the EC3 experimental channel of the YALINA thermal with 216 fuel rods from criticality calculation by MCNPX and PARTISN

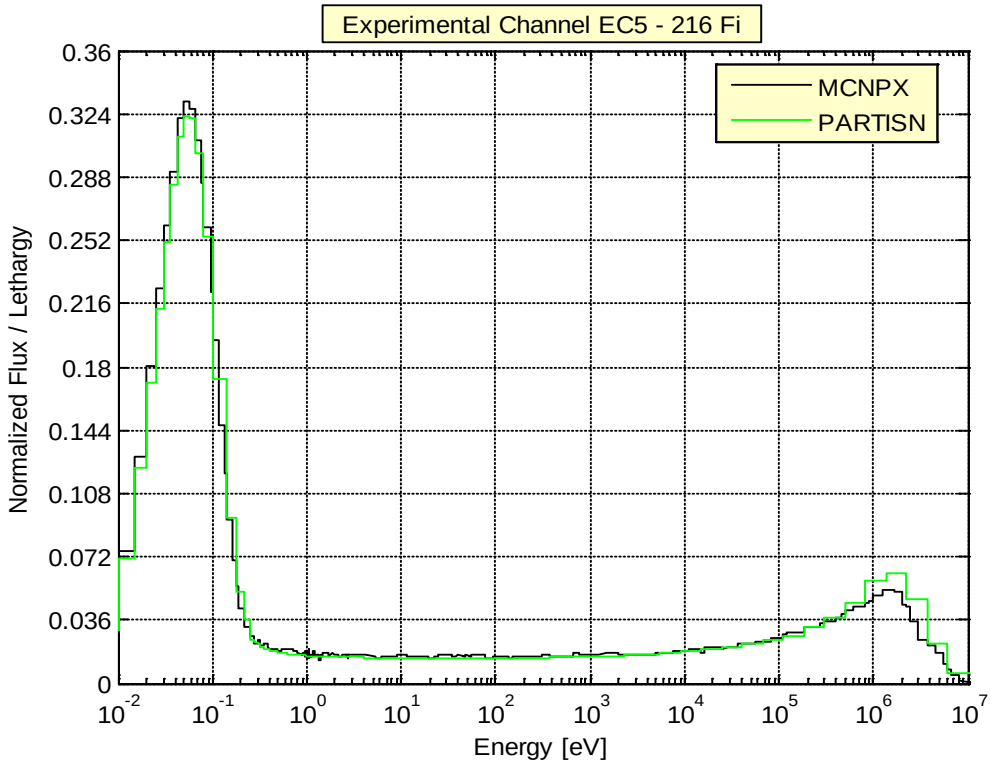

Figure 97. Neutron spectra in the EC5 experimental channel of the YALINA thermal with 216 fuel rods from criticality calculation by MCNPX and PARTISN

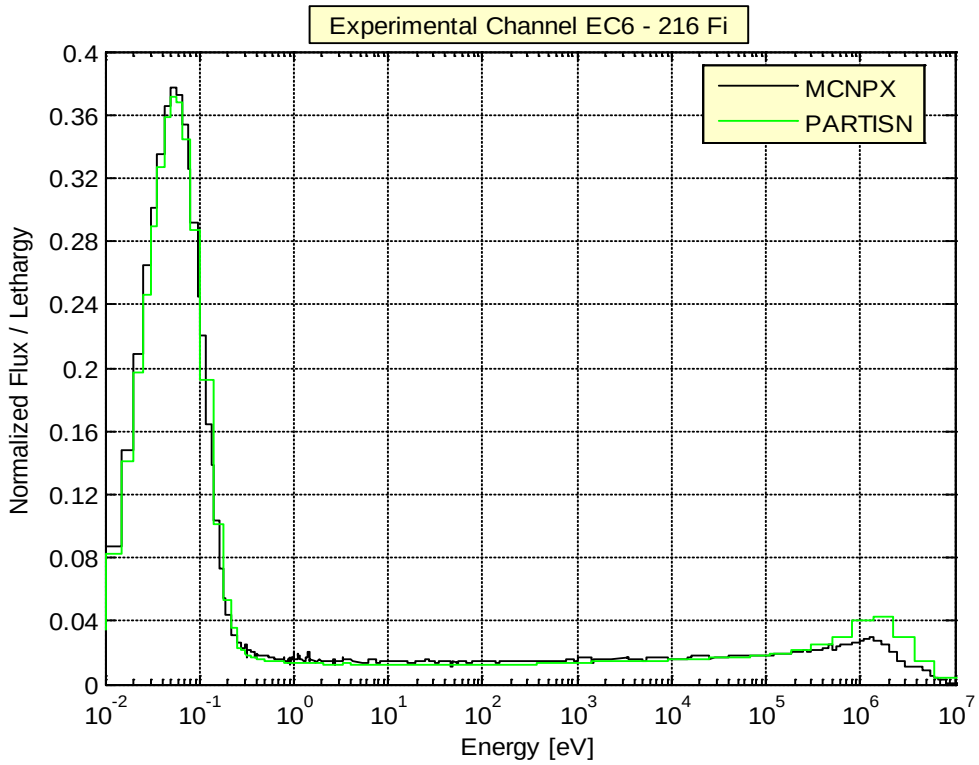

Figure 98. Neutron spectra in the EC6 experimental channel of the YALINA thermal with 216 fuel rods from criticality calculation by MCNPX and PARTISN

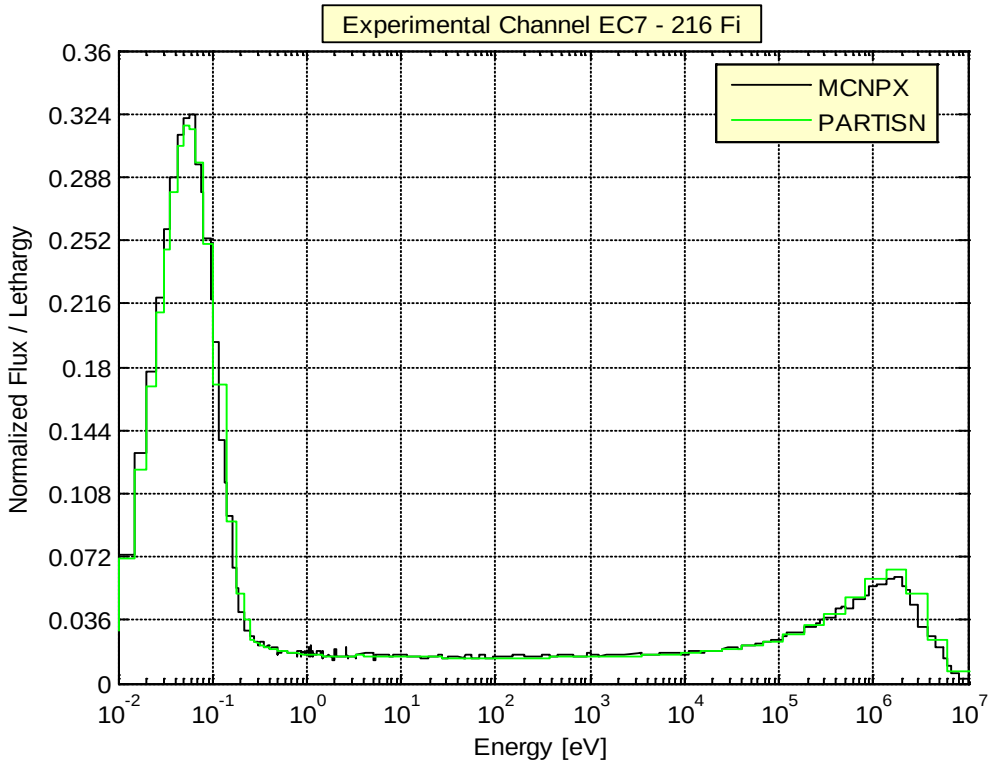

Figure 99. Neutron spectra in the EC7 experimental channel of the YALINA thermal with 216 fuel rods from criticality calculation by MCNPX and PARTISN

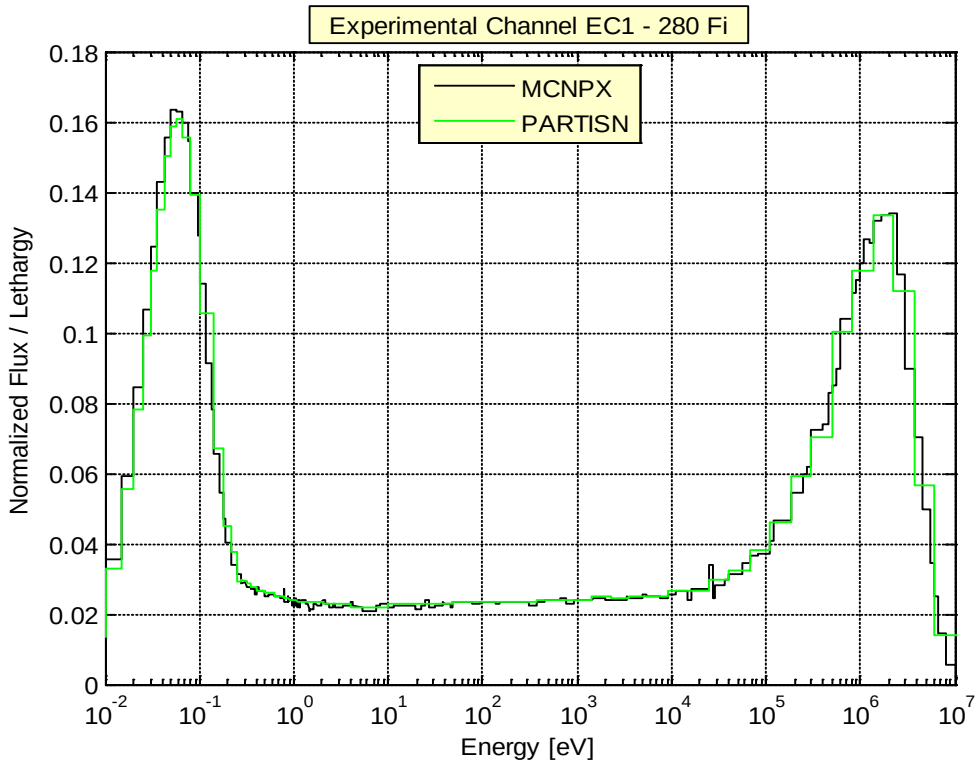

Figure 100. Neutron spectra in the EC1 experimental channel of the YALINA thermal with 280 fuel rods from criticality calculation by MCNPX and PARTISN

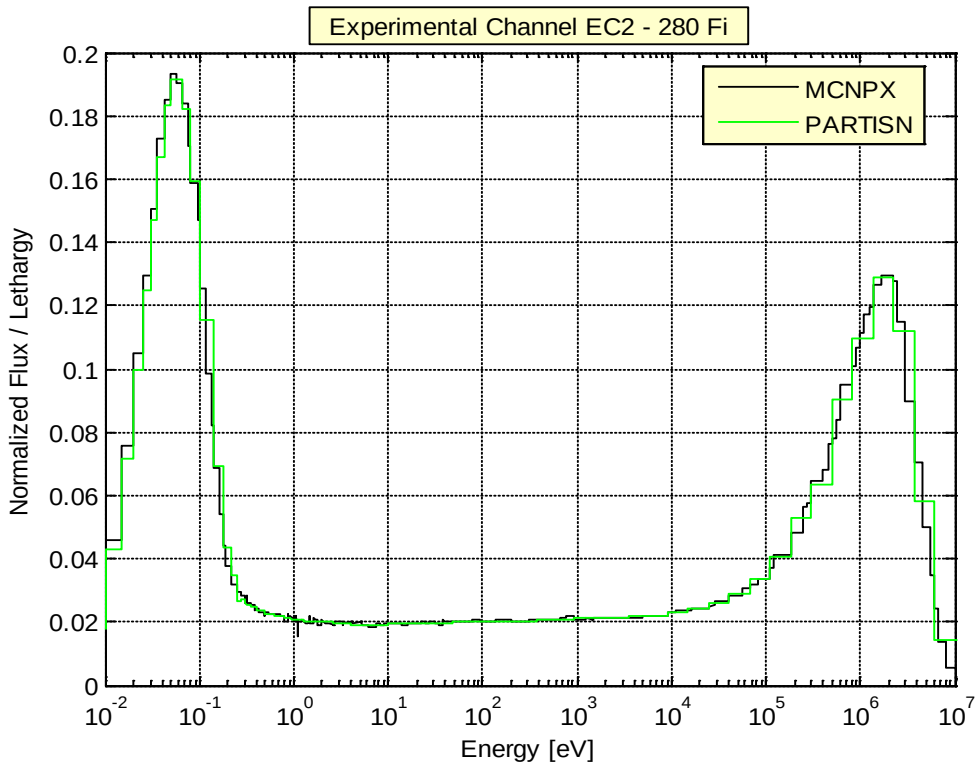

Figure 101. Neutron spectra in the EC2 experimental channel of the YALINA thermal with 280 fuel rods from criticality calculation by MCNPX and PARTISN

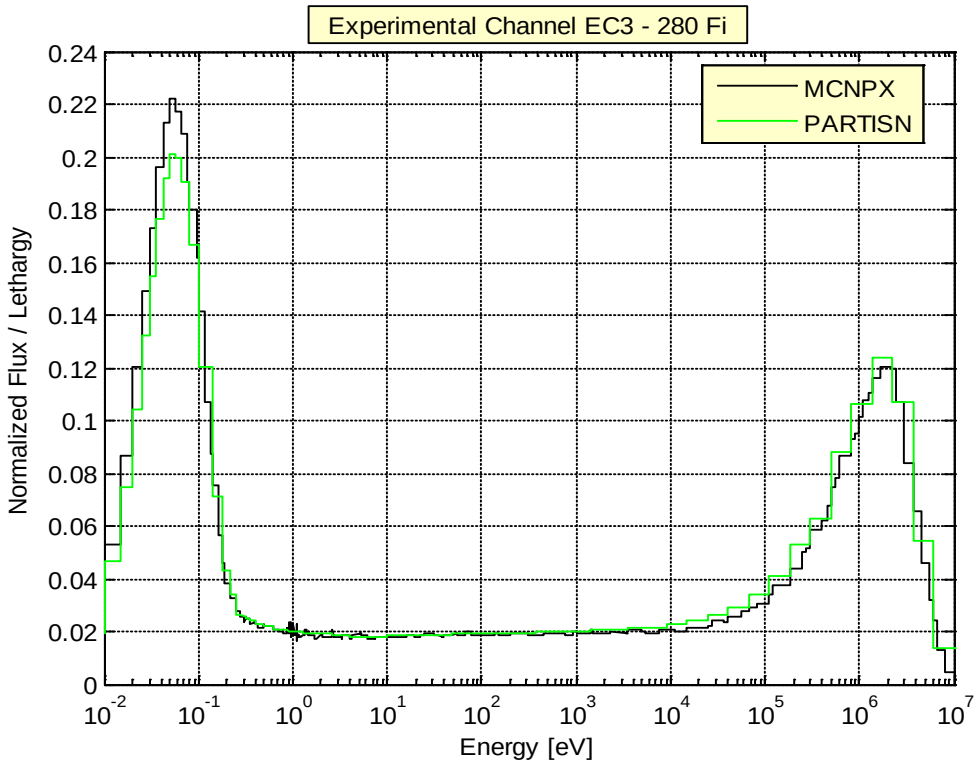

Figure 102. Neutron spectra in the EC3 experimental channel of the YALINA thermal with 280 fuel rods from criticality calculation by MCNPX and PARTISN

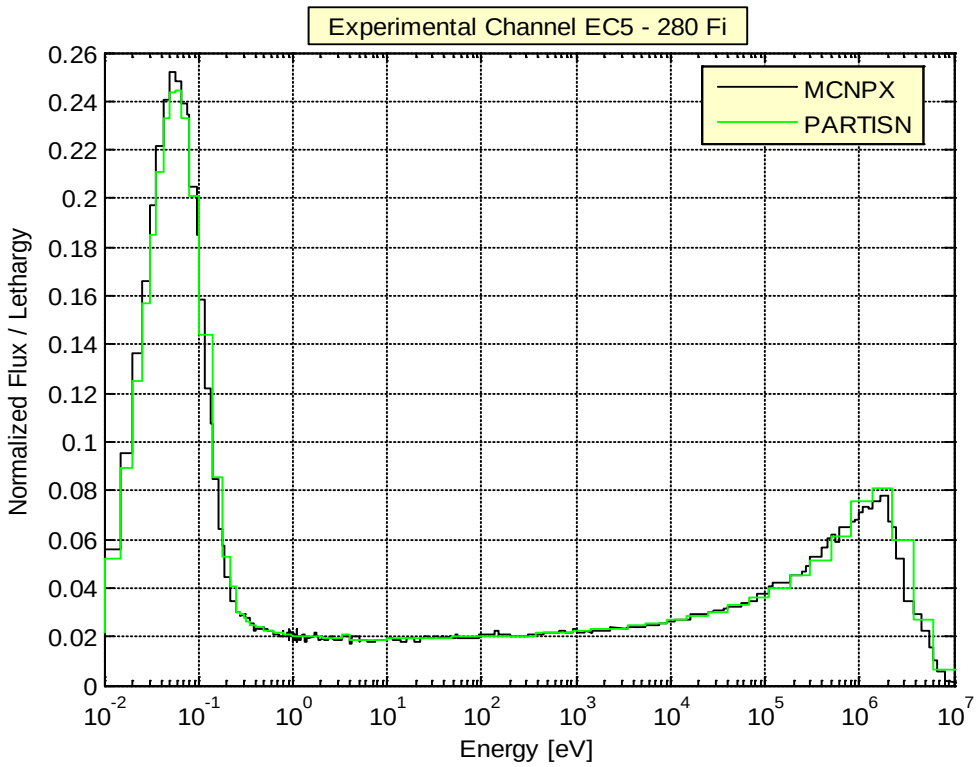

Figure 103. Neutron spectra in the EC5 experimental channel of the YALINA thermal with 280 fuel rods from criticality calculation by MCNPX and PARTISN

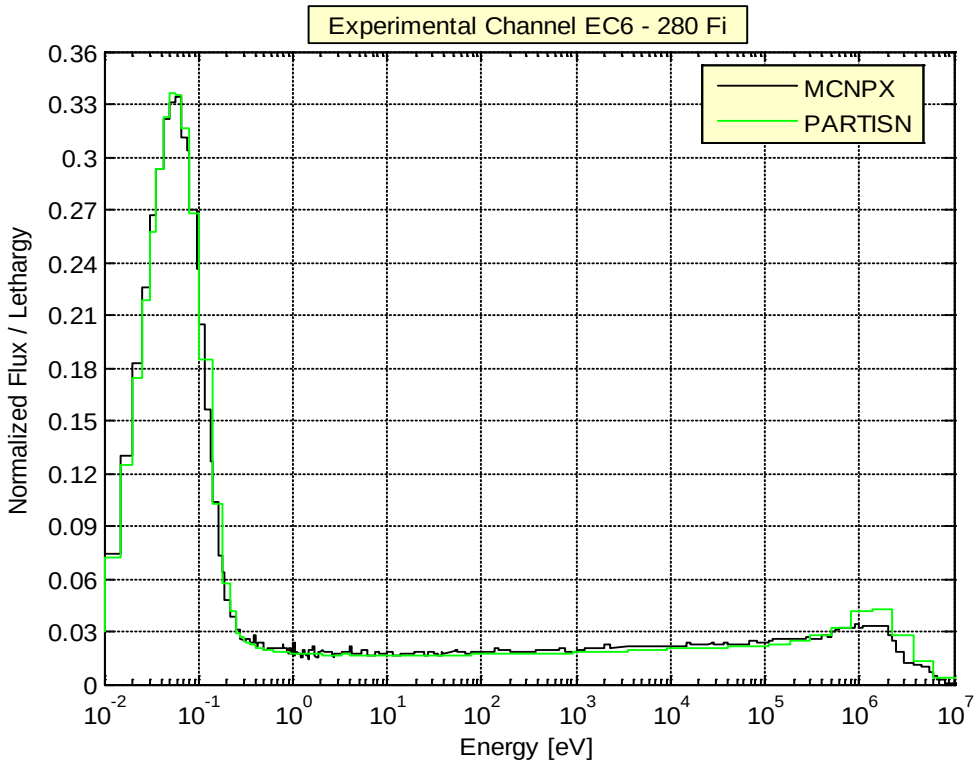

Figure 104. Neutron spectra in the EC6 experimental channel of the YALINA thermal with 280 fuel rods from criticality calculation by MCNPX and PARTISN

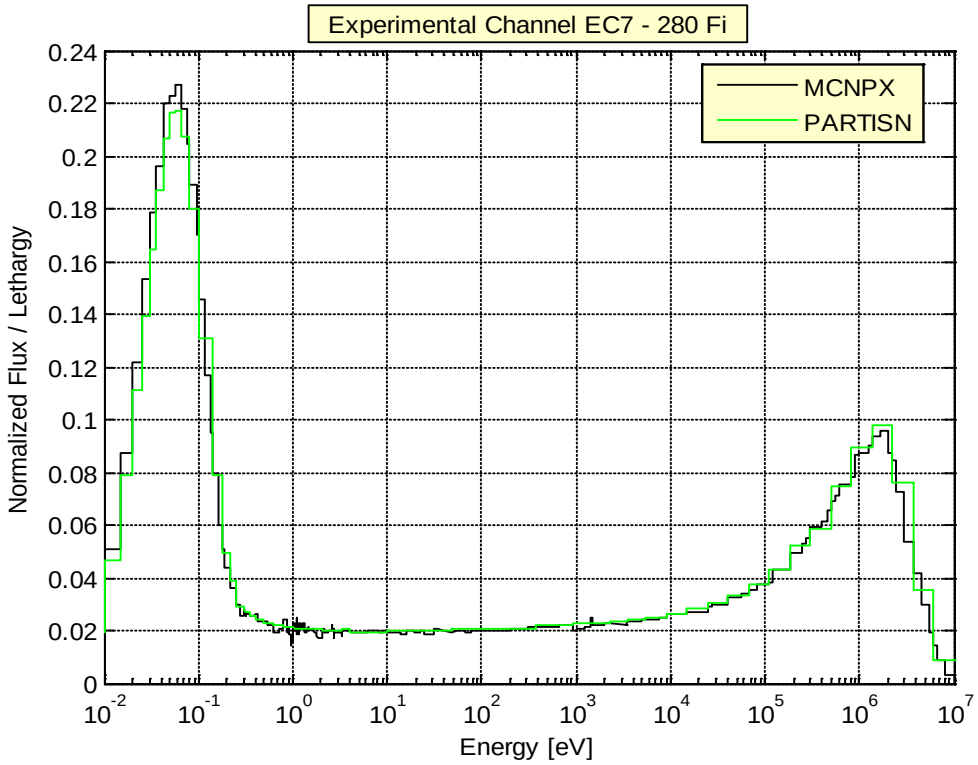

Figure 105. Neutron spectra in the EC7 experimental channel of the YALINA thermal with 280 fuel rods from criticality calculation by MCNPX and PARTISN

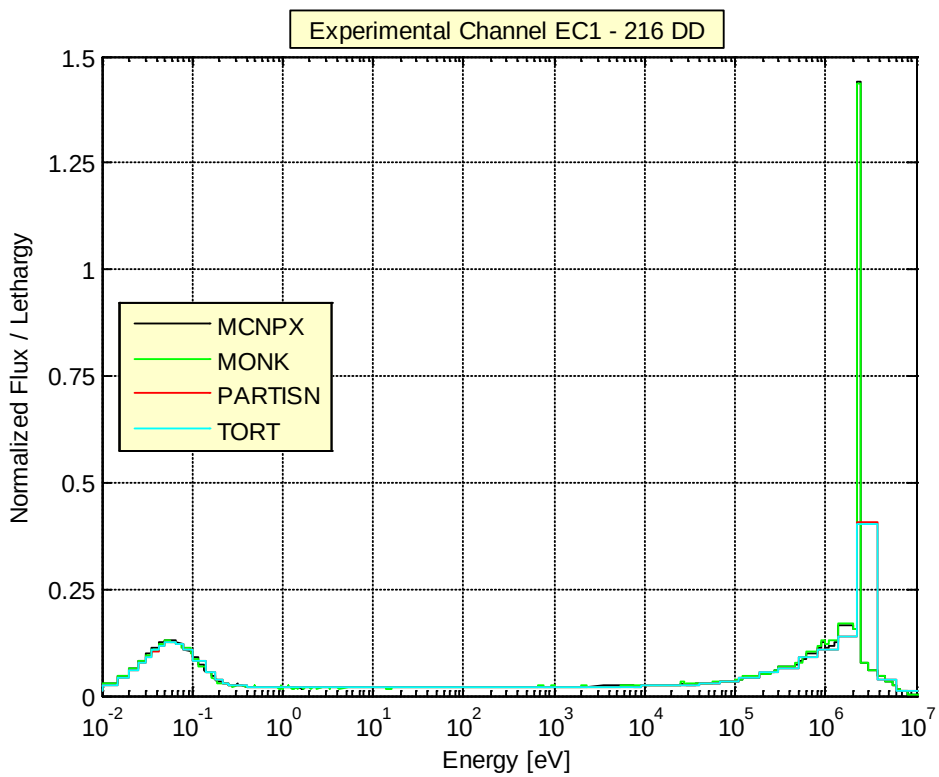

Figure 106. Neutron spectra in the EC1 experimental channel of the YALINA thermal with 216 fuel rods with D-D neutron source calculated by MCNPX, MONK, PARTISN, and TORT

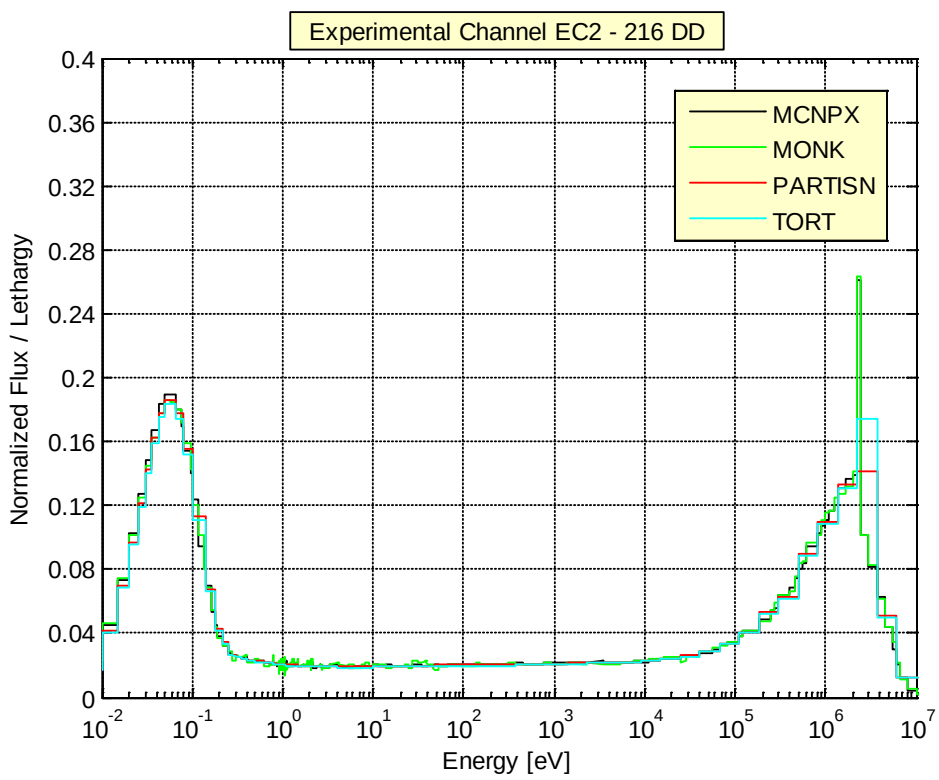

Figure 107. Neutron spectra in the EC2 experimental channel of the YALINA thermal with 216 fuel rods with D-D neutron source calculated by MCNPX, MONK, PARTISN, and TORT

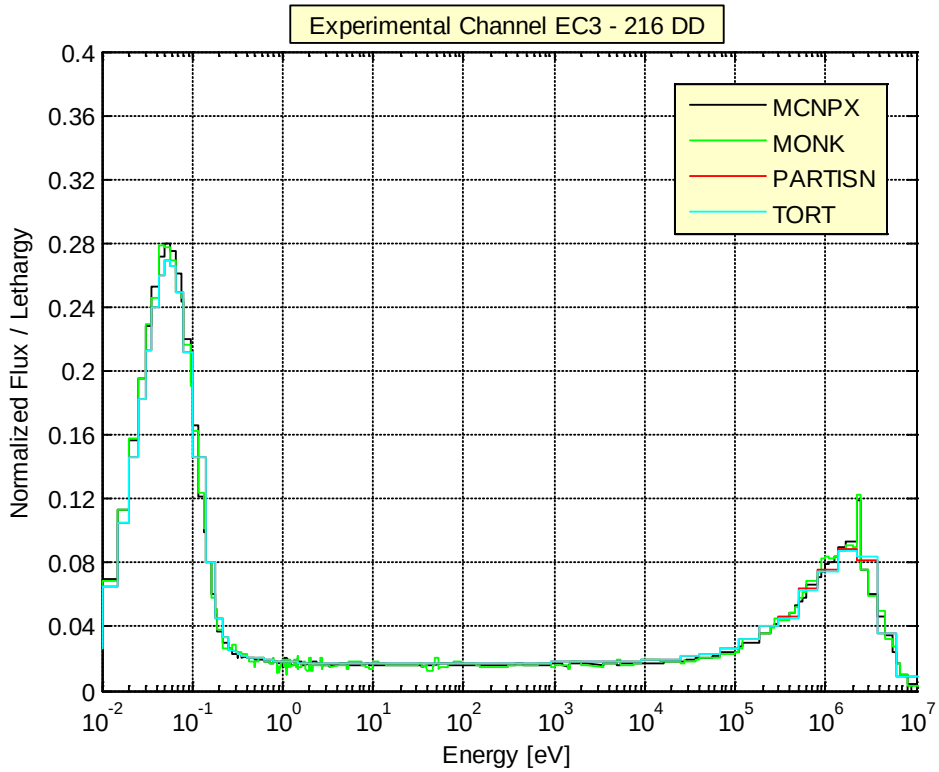

Figure 108. Neutron spectra in the EC3 experimental channel of the YALINA thermal with 216 fuel rods with D-D neutron source calculated by MCNPX, MONK, PARTISN, and TORT

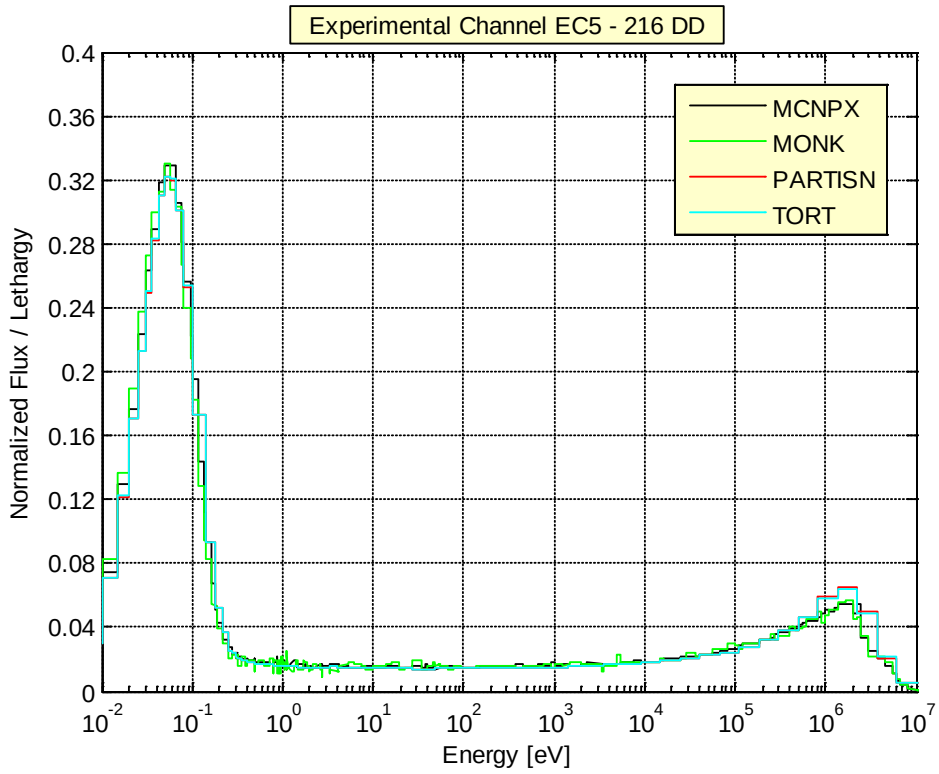

Figure 109. Neutron spectra in the EC5 experimental channel of the YALINA thermal with 216 fuel rods with D-D neutron source calculated by MCNPX, MONK, PARTISN, and TORT

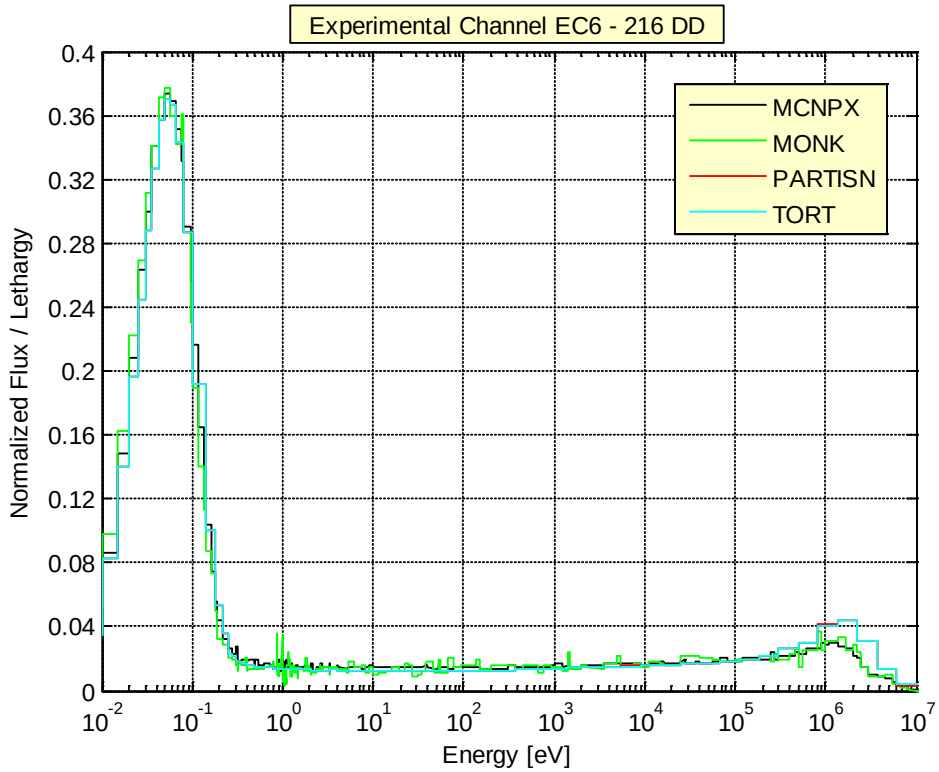

Figure 110. Neutron spectra in the EC6 experimental channel of the YALINA thermal with 216 fuel rods with D-D neutron source calculated by MCNPX, MONK, PARTISN, and TORT

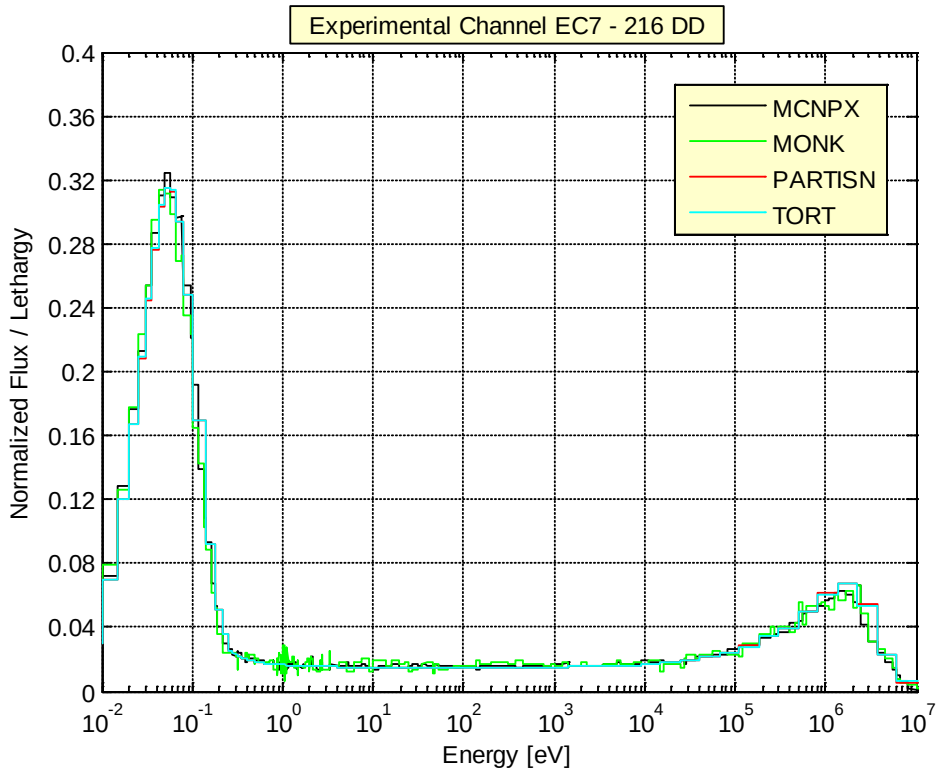

Figure 111. Neutron spectra in the EC7 experimental channel of the YALINA thermal with 216 fuel rods with D-D neutron source calculated by MCNPX, MONK, PARTISN, and TORT

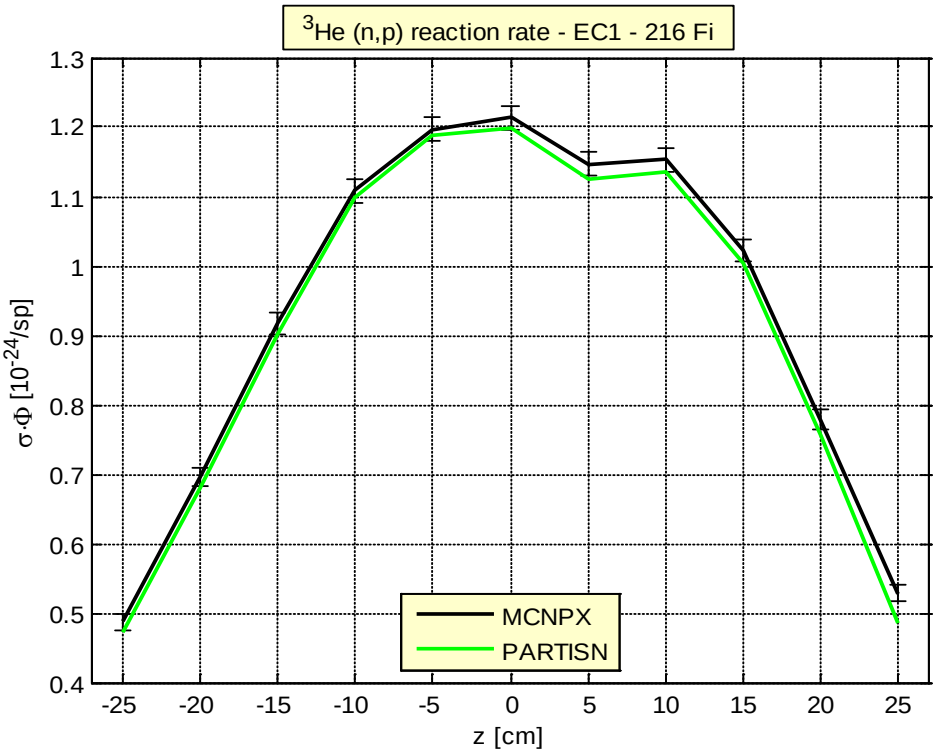

Figure 112. ^{3}He reaction rate in the EC1 experimental channel of the YALINA thermal with 216 fuel rods from criticality calculation by MCNPX and PARTISN

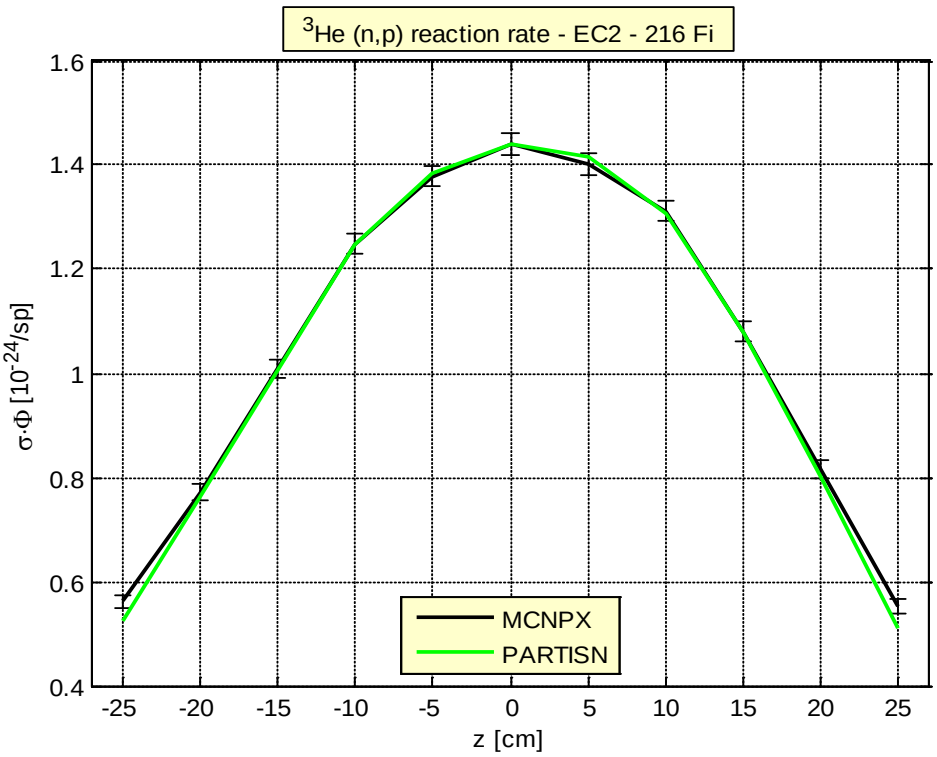

Figure 113. ^{3}He reaction rate in the EC2 experimental channel of the YALINA thermal with 216 fuel rods from criticality calculation by MCNPX and PARTISN

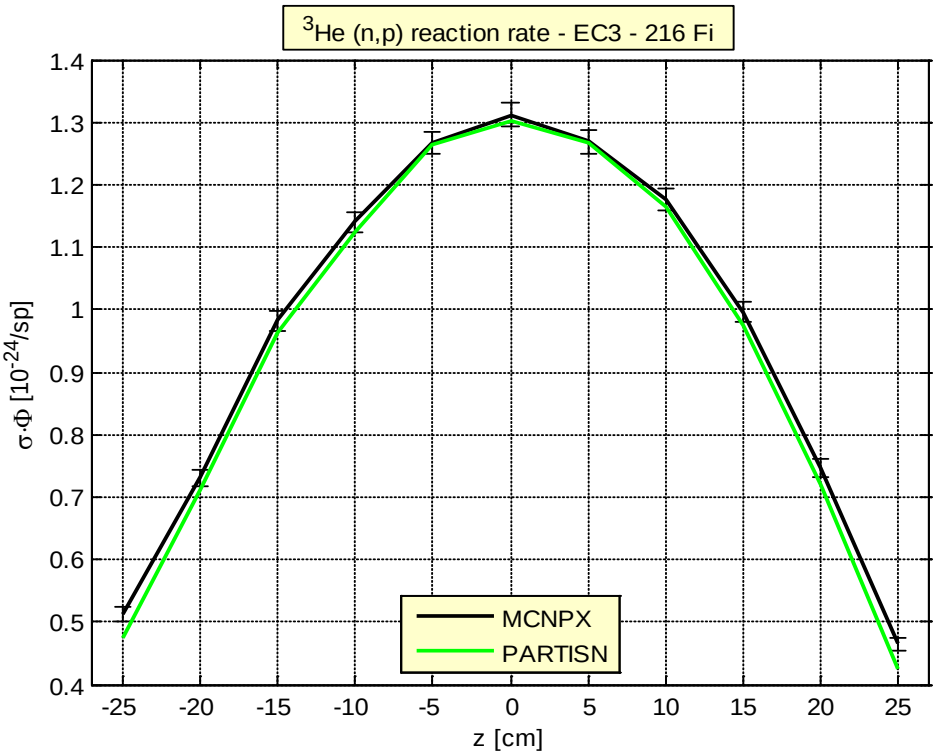

Figure 114. ^{3}He reaction rate in the EC3 experimental channel of the YALINA thermal with 216 fuel rods from criticality calculation by MCNPX and PARTISN

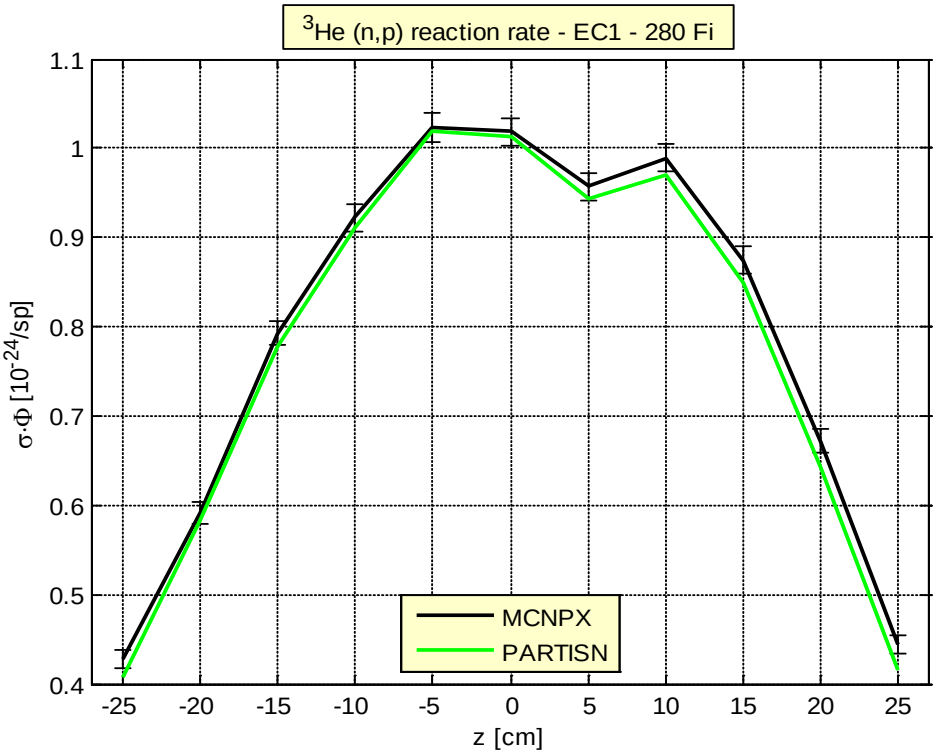

Figure 115. ^{3}He reaction rate in the EC1 experimental channel of the YALINA thermal with 280 fuel rods from criticality calculation by MCNPX and PARTISN

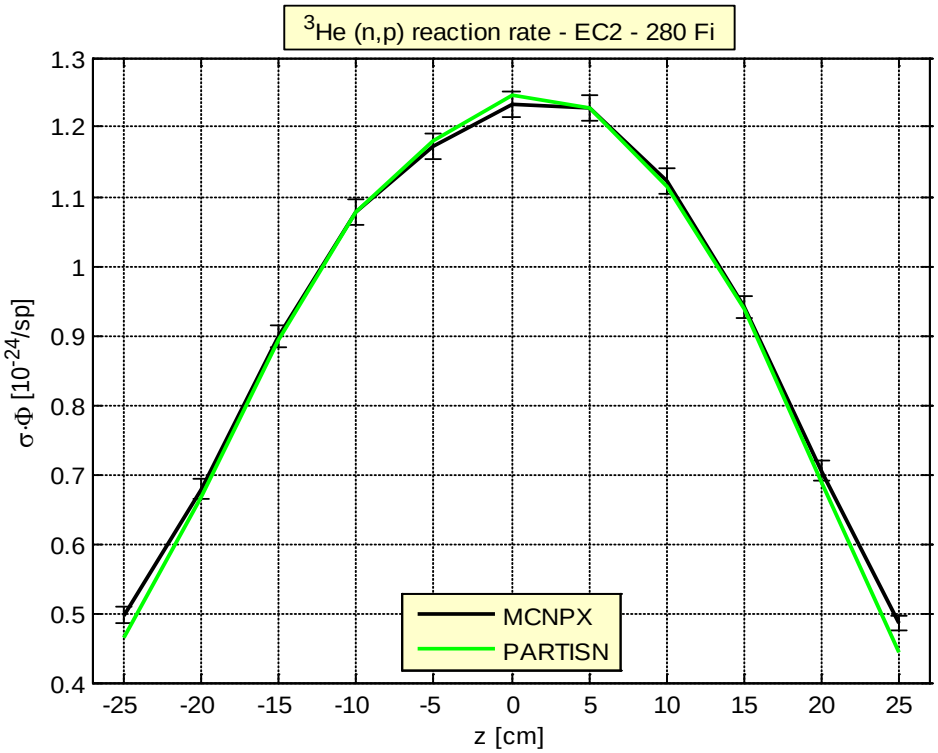

Figure 116. ^{3}He reaction rate in the EC2 experimental channel of the YALINA thermal with 280 fuel rods from criticality calculation by MCNPX and PARTISN

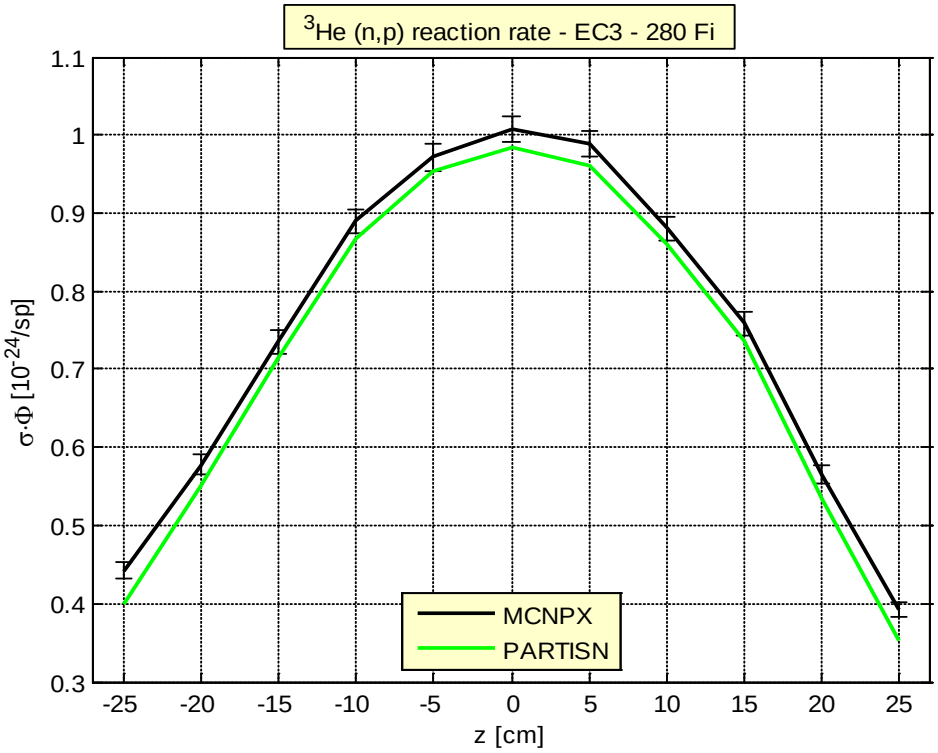

Figure 117. ^{3}He reaction rate in the EC3 experimental channel of the YALINA thermal with 280 fuel rods from criticality calculation by MCNPX and PARTISN

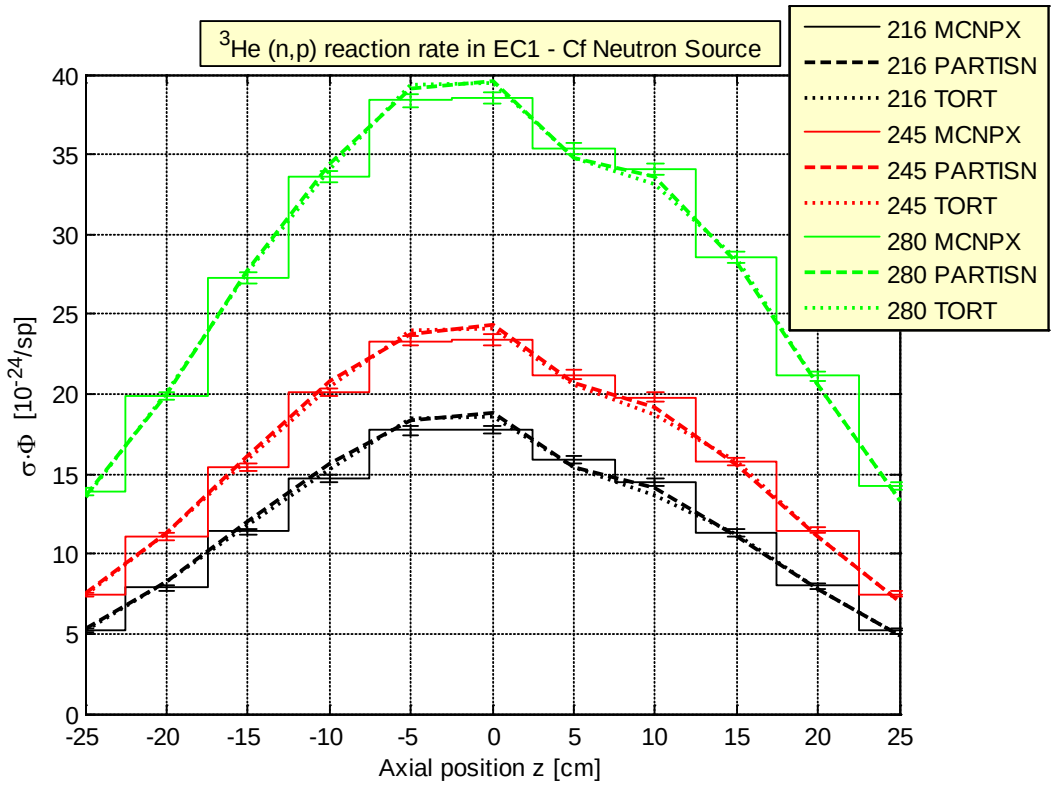

Figure 118. ^{3}He(n,p) reaction rate per source particle in the EC1 experimental channel with Cf neutron source calculated by MCNPX, PARTISN, and TORT

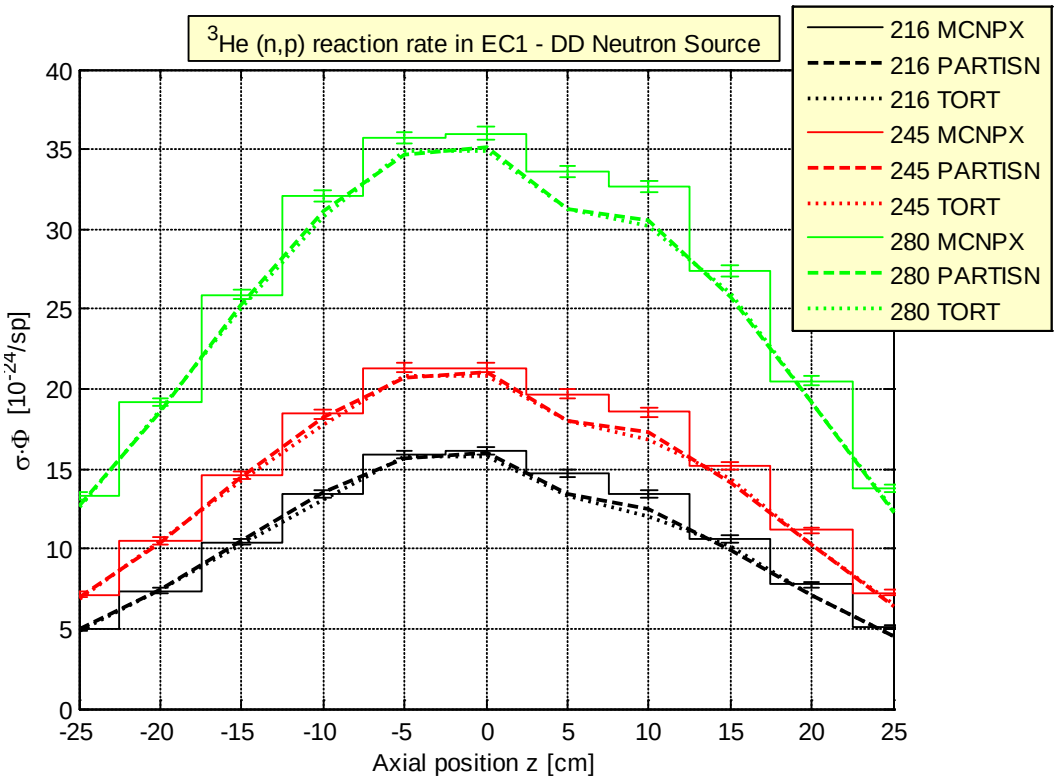

Figure 119. ^{3}He(n,p) reaction rate per source particle in the EC1 experimental channel with D-D neutron source calculated by MCNPX, PARTISN, and TORT

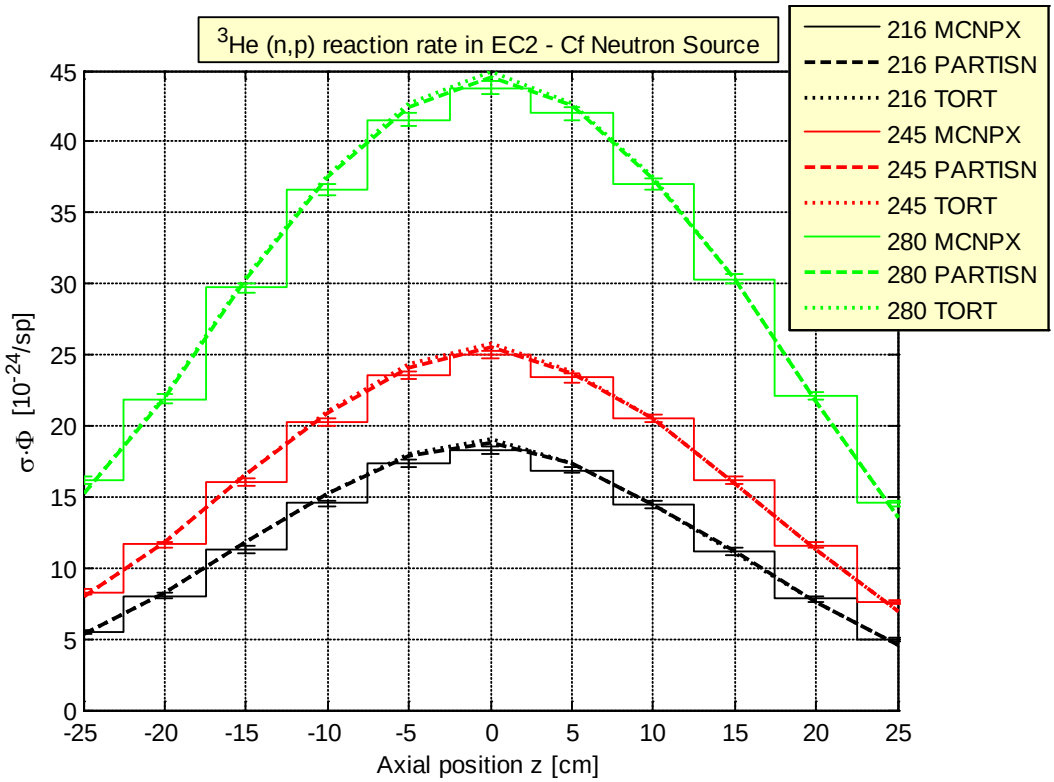

Figure 120. ^{3}He(n,p) reaction rate per source particle in the EC2 experimental channel with Cf neutron source calculated by MCNPX, PARTISN, and TORT

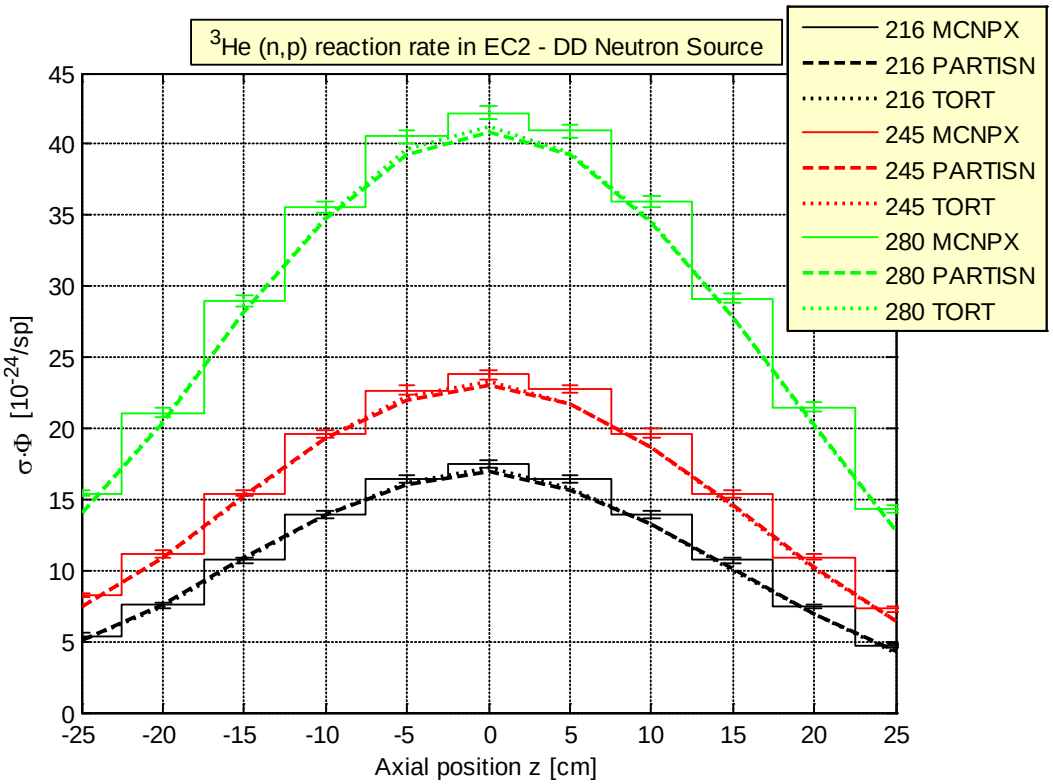

Figure 121. ^{3}He(n,p) reaction rate per source particle in the EC2 experimental channel with D-D neutron source calculated by MCNPX, PARTISN, and TORT

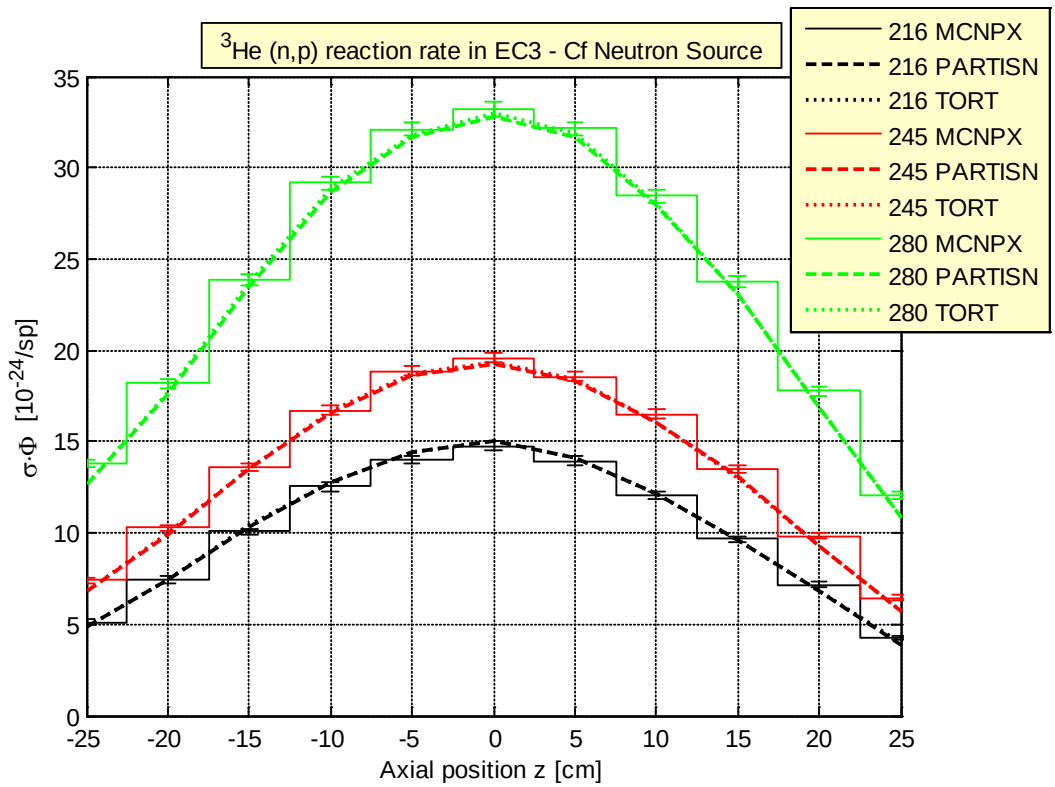

Figure 122. ^{3}He(n,p) reaction rate per source particle in the EC3 experimental channel with Cf neutron source calculated by MCNPX, PARTISN, and TORT

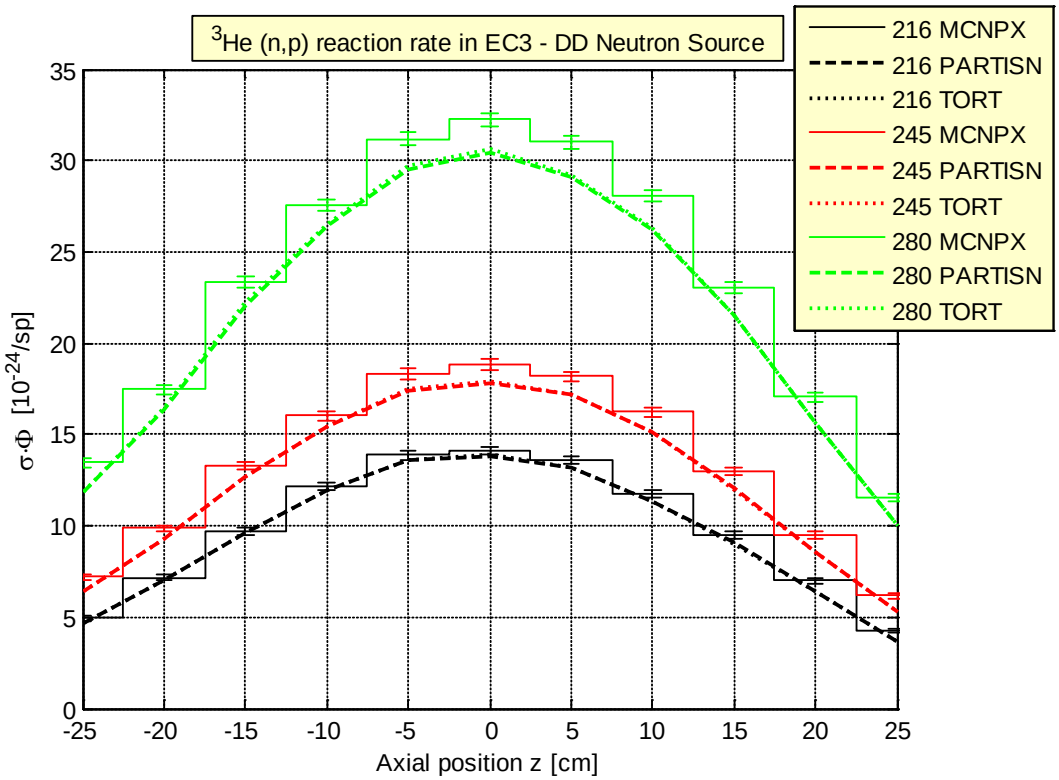

Figure 123. ^{3}He(n,p) reaction rate per source particle in the EC3 experimental channel with D-D neutron source calculated by MCNPX, PARTISN, and TORT

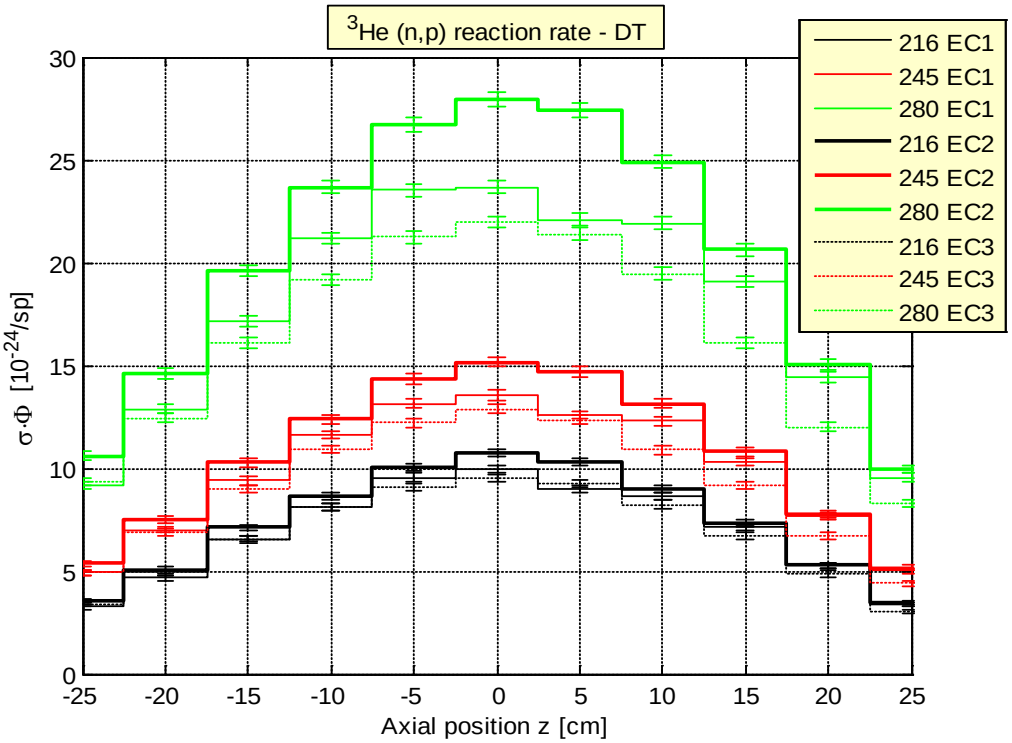

Figure 124. ^{3}He(n,p) reaction rate per source particle with D-T neutron source calculated by MCNPX

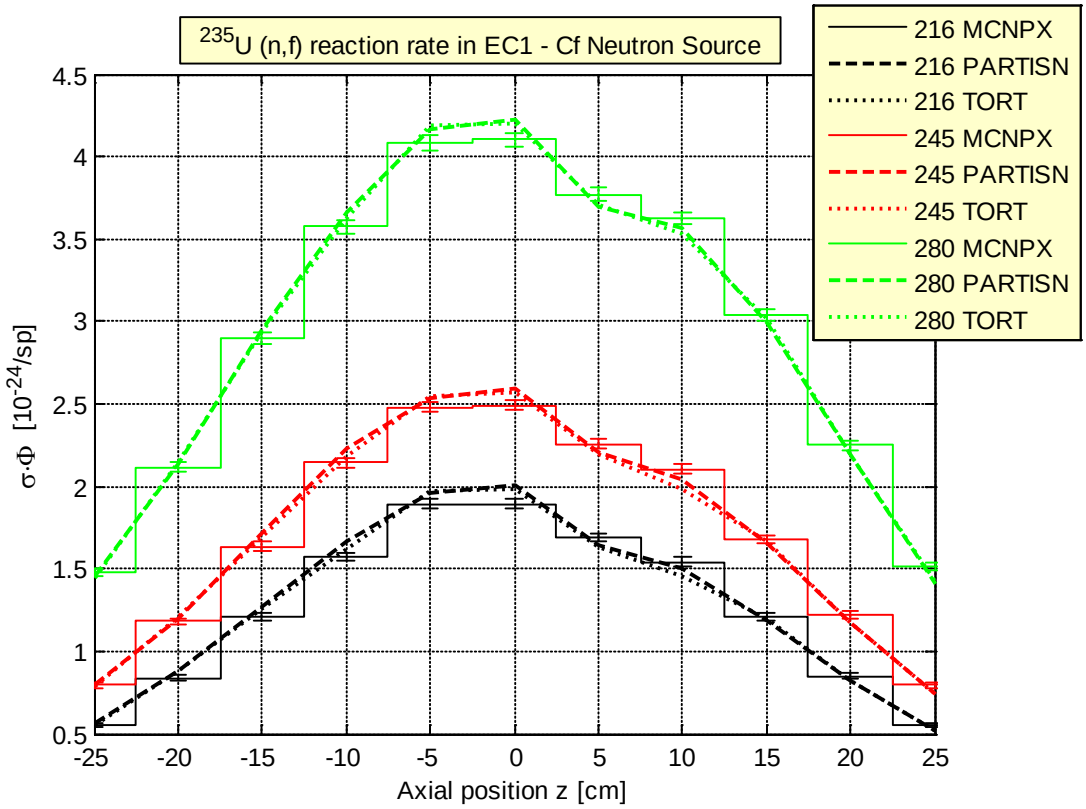

Figure 125. ^{235}U (n,f) reaction rate per source particle in the EC1 experimental channel with Cf neutron source calculated by MCNPX, PARTISN, and TORT

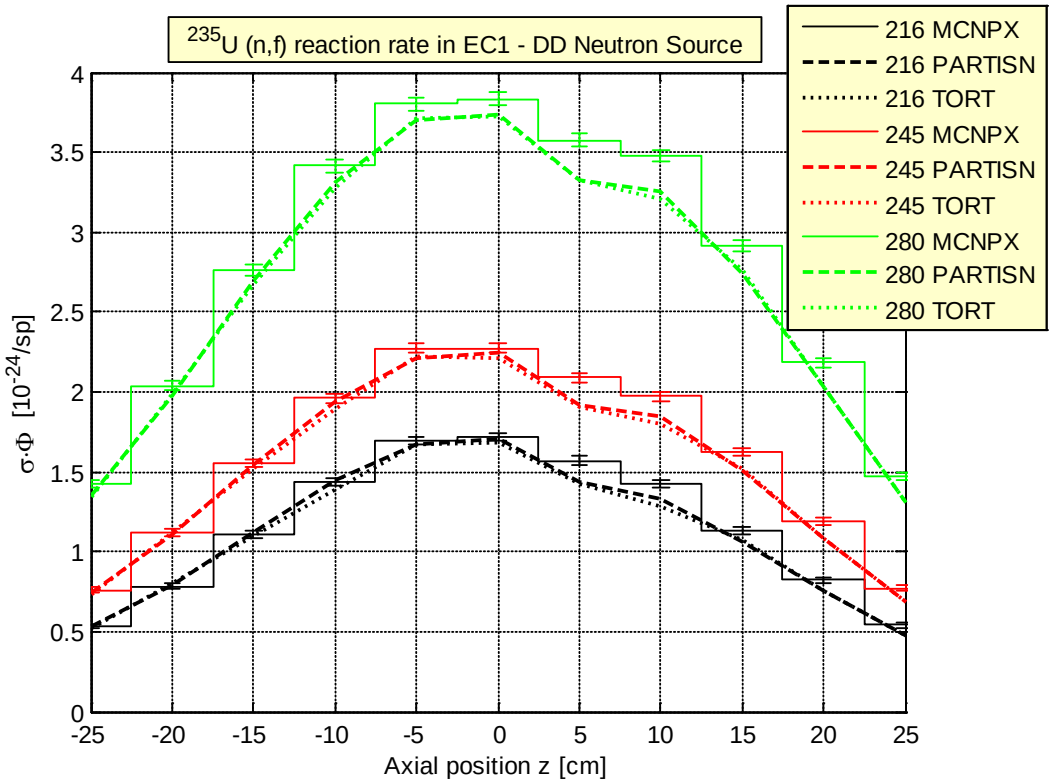

Figure 126. ^{235}U (n,f) reaction rate per source particle in the EC1 experimental channel with D-D neutron source calculated by MCNPX, PARTISN, and TORT

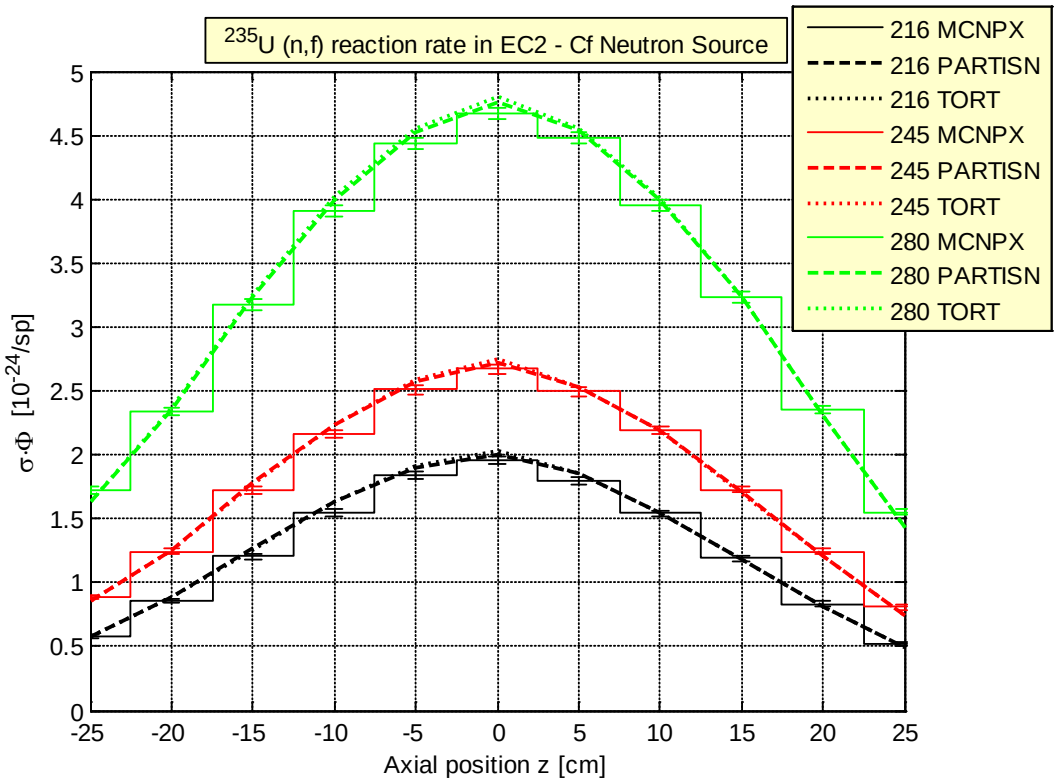

Figure 127. ^{235}U (n,f) reaction rate per source particle in the EC2 experimental channel with Cf neutron source calculated by MCNPX, PARTISN, and TORT

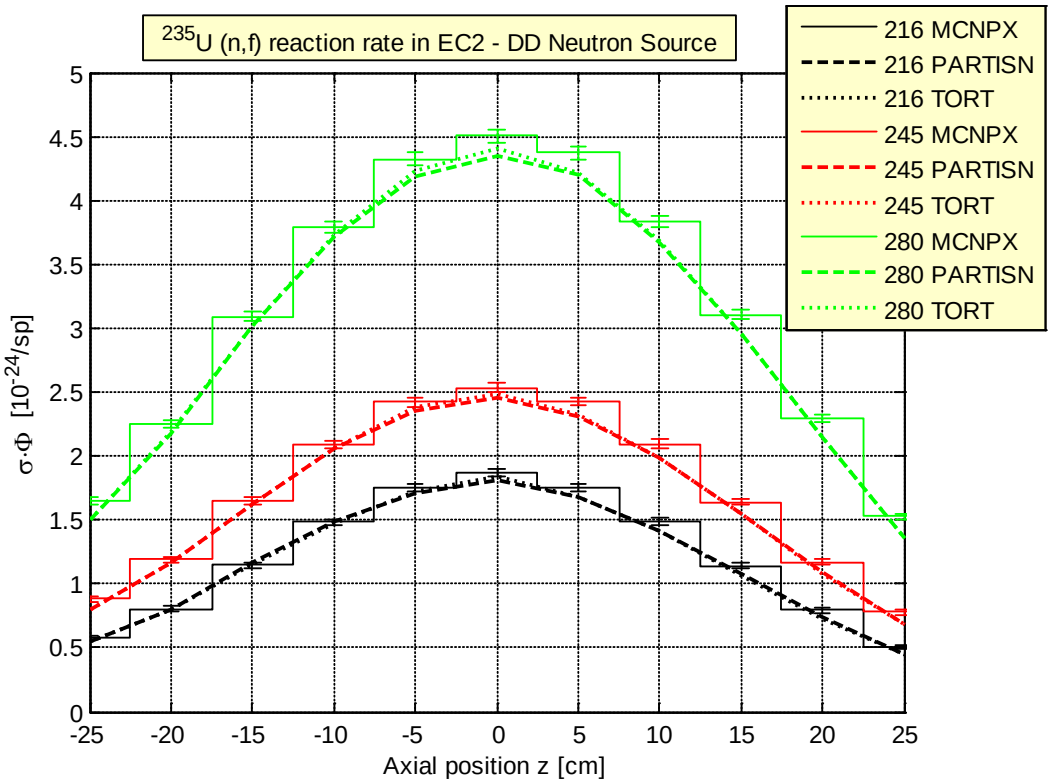

Figure 128. ^{235}U (n,f) reaction rate per source particle in the EC2 experimental channel with D-D neutron source calculated by MCNPX, PARTISN, and TORT

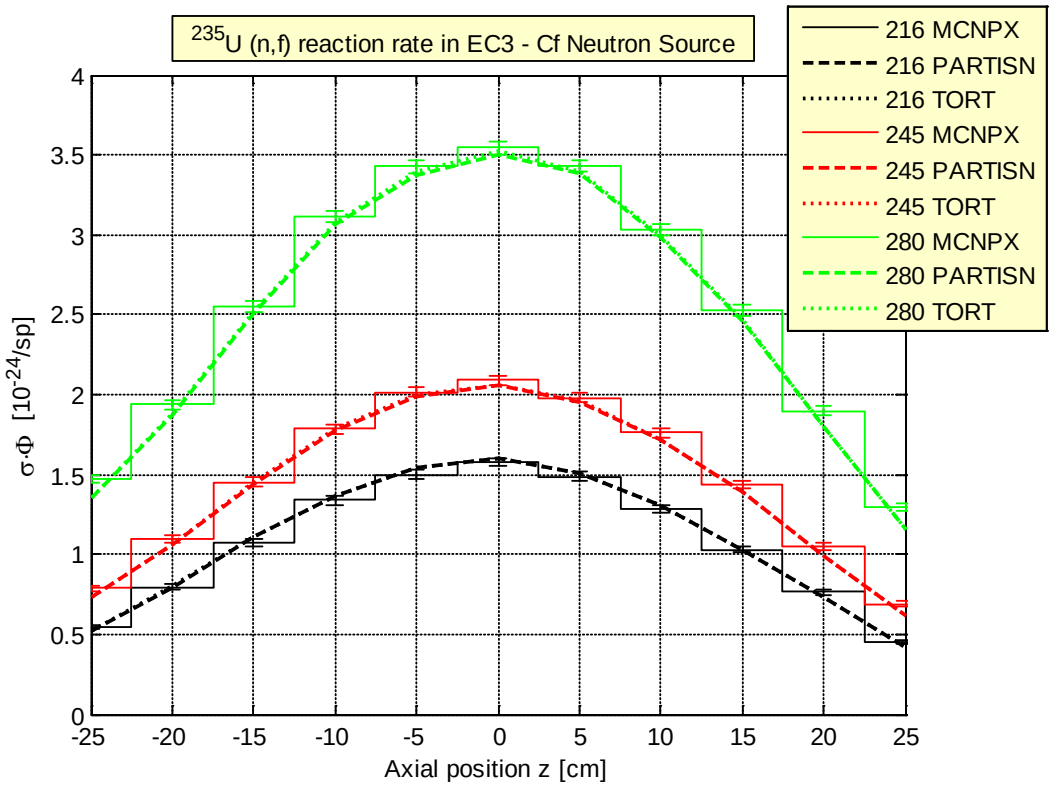

Figure 129. ^{235}U (n,f) reaction rate per source particle in the EC3 experimental channel with Cf neutron source calculated by MCNPX, PARTISN, and TORT

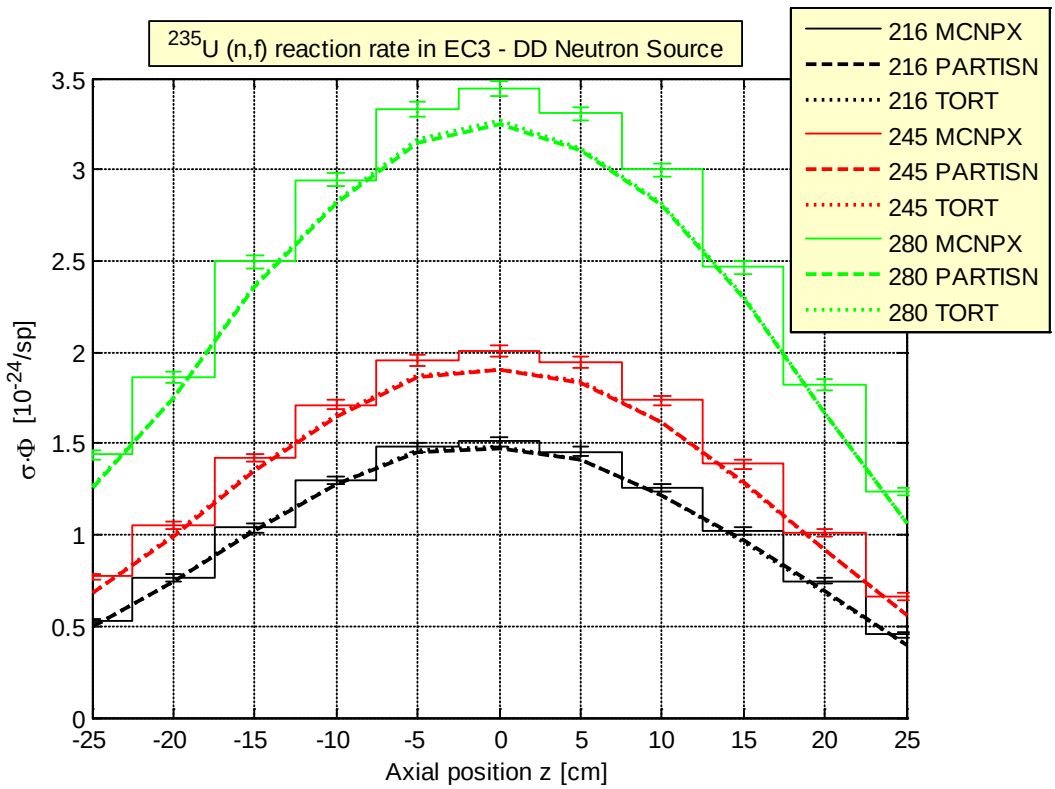

Figure 130. ^{235}U (n,f) reaction rate per source particle in the EC3 experimental channel with D-D neutron source calculated by MCNPX, PARTISN, and TORT

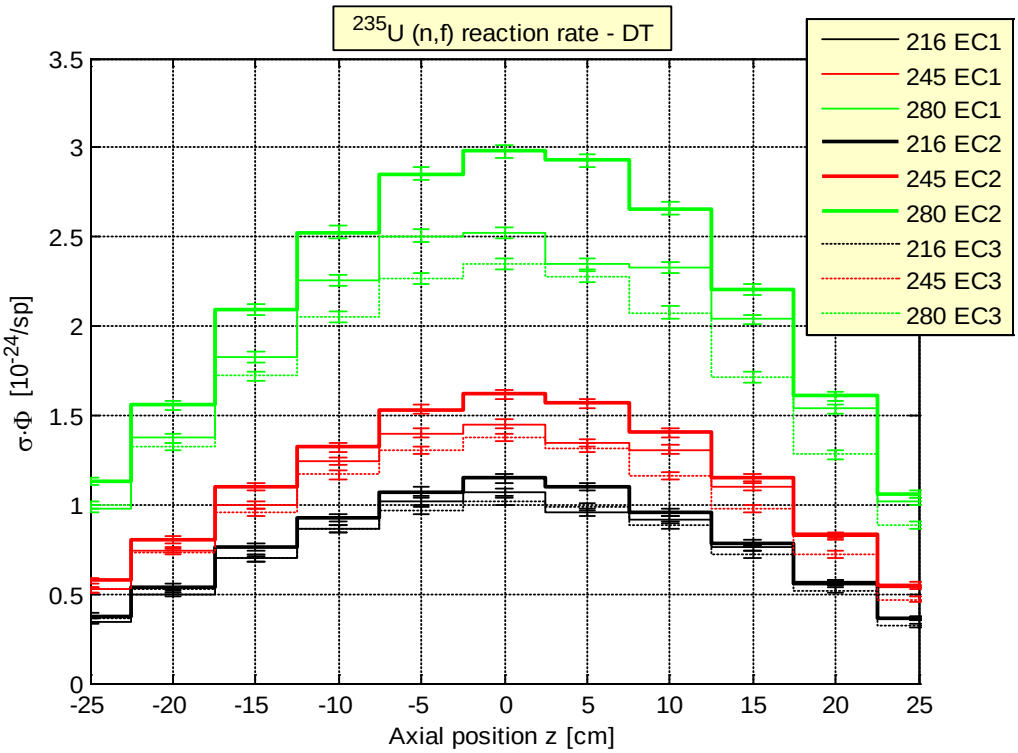

Figure 131. ^{235}U (n,f) reaction rate per source particle with D-T neutron source calculated by MCNPX

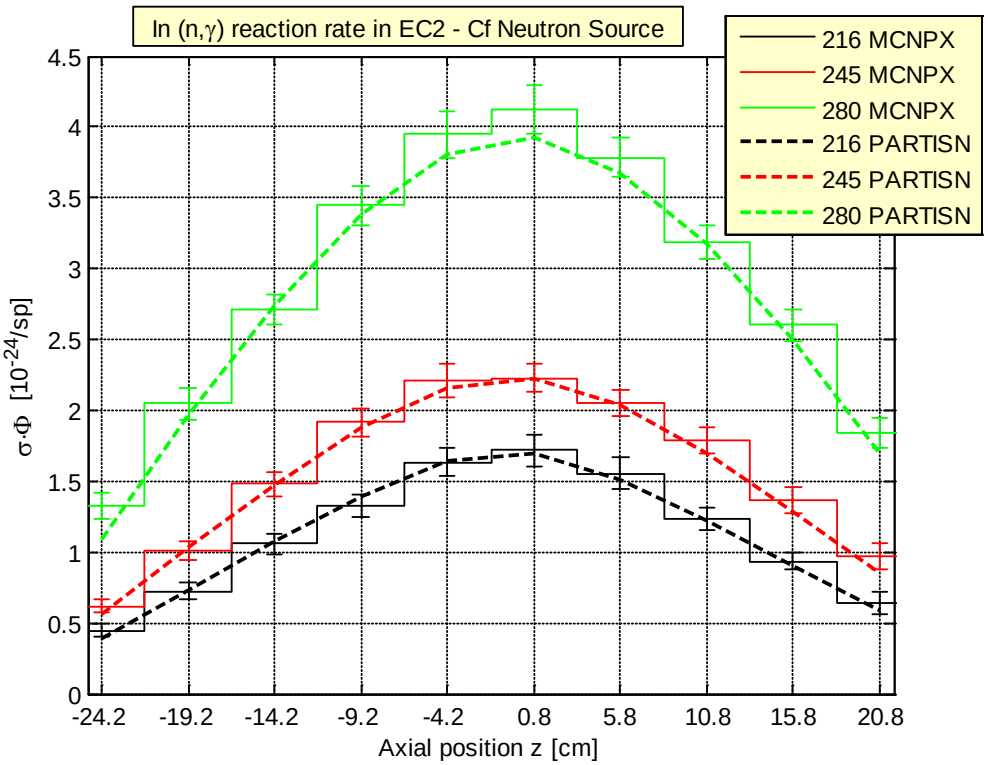

Figure 132. In (n,γ) reaction rate per source particle in EC2 experimental channel with Cf neutron source calculated by MCNPX and PARTISN

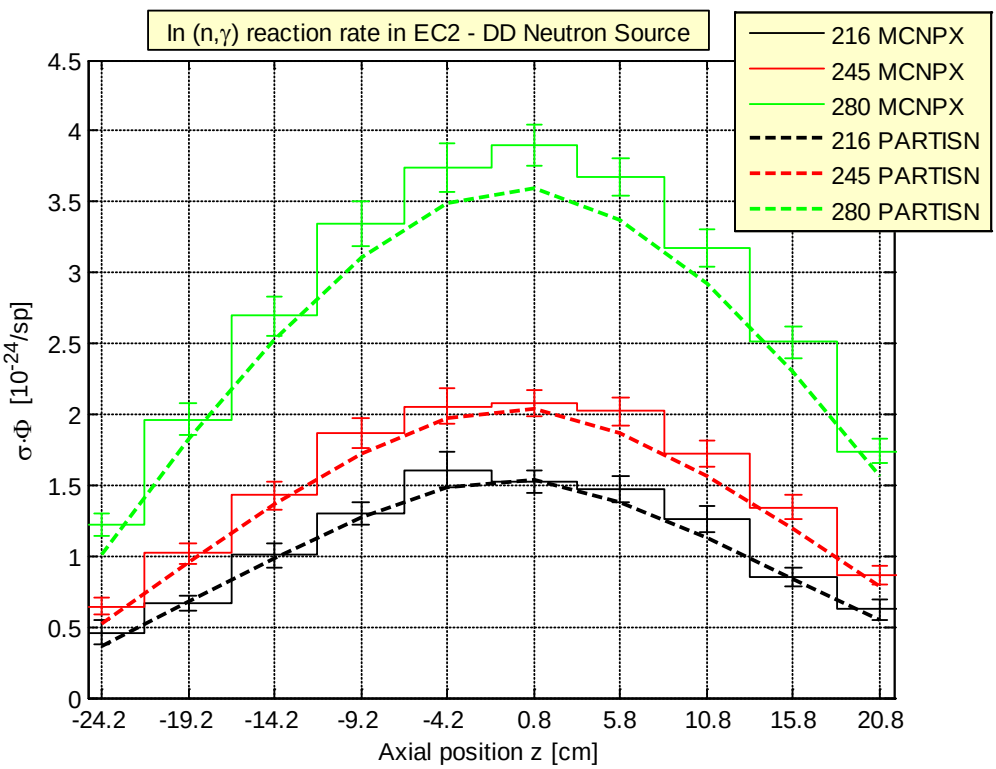

Figure 133. In (n,γ) reaction rate per source particle in EC2 experimental channel with D-D neutron source calculated by MCNPX and PARTISN

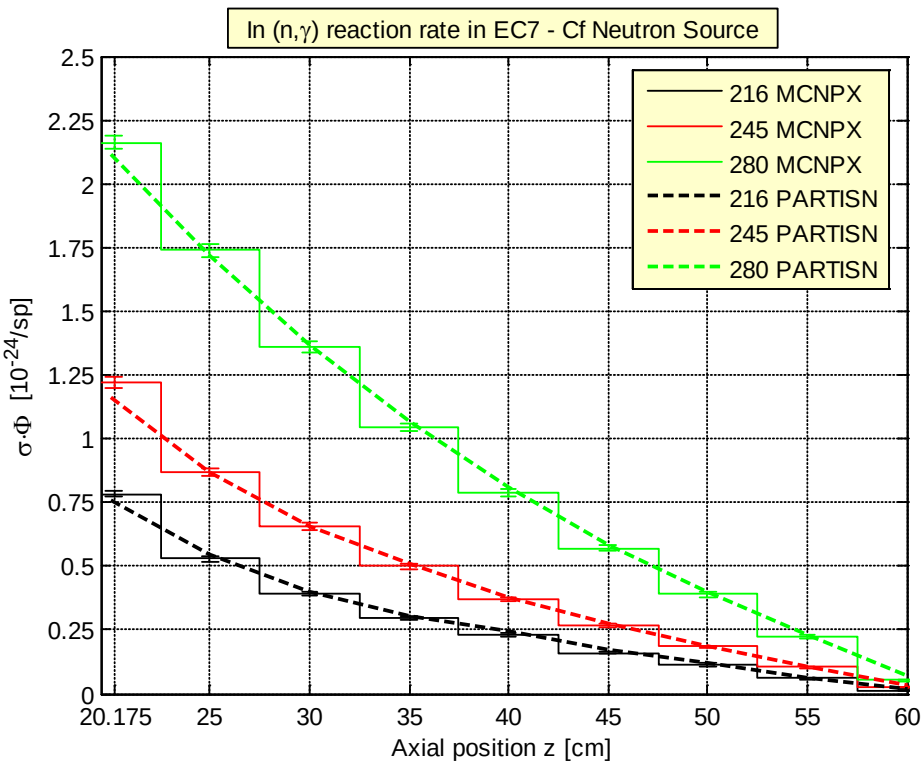

Figure 134. In (n,γ) reaction rate per source particle in EC7 experimental channel with Cf neutron source calculated by MCNPX and PARTISN

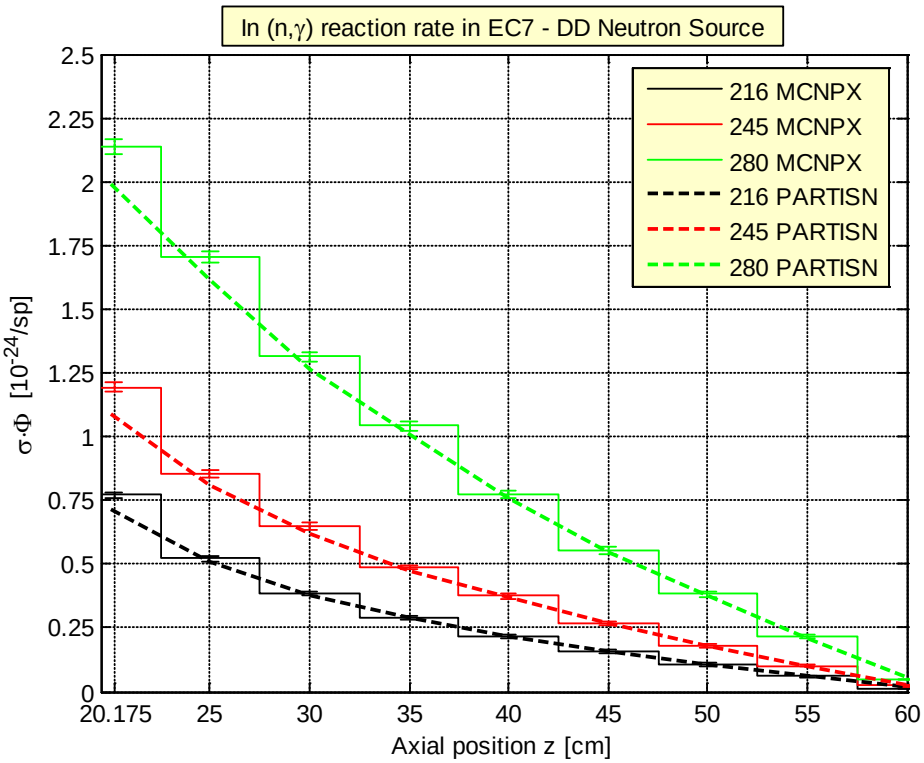

Figure 135. In (n,γ) reaction rate per source particle in EC7 experimental channel with D-D neutron source calculated by MCNPX and PARTISN

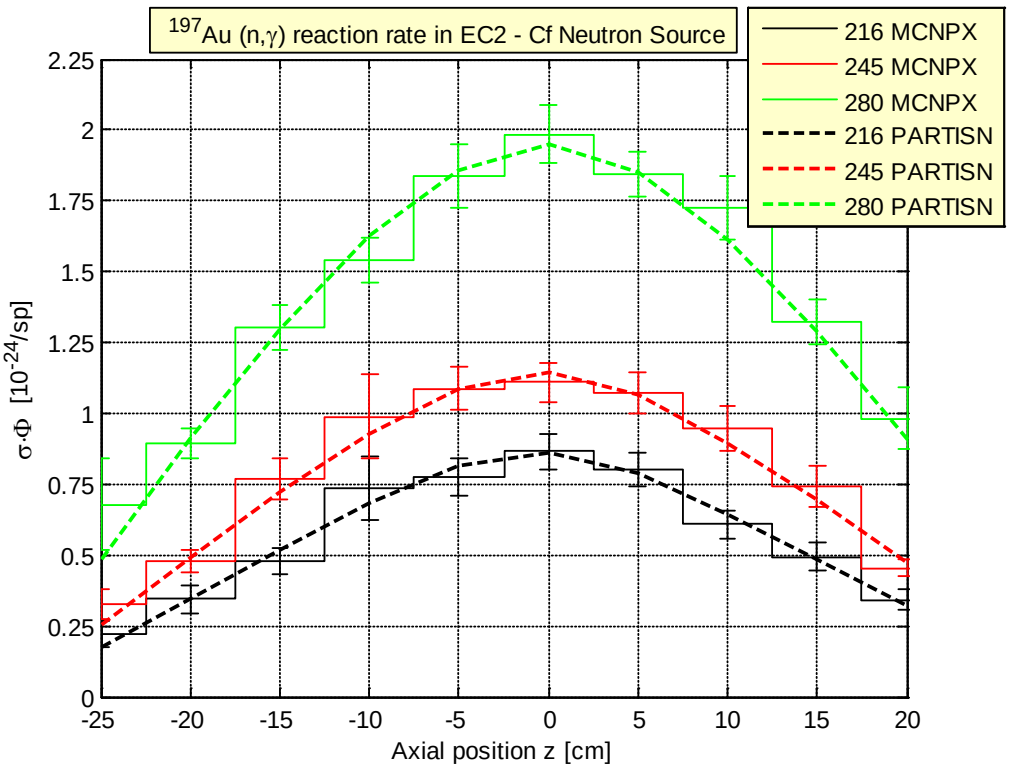

Figure 136. ^{197}Au (n,γ) reaction rate per source particle in the EC2 experimental channel with Cf neutron source calculated by MCNPX and PARTISN

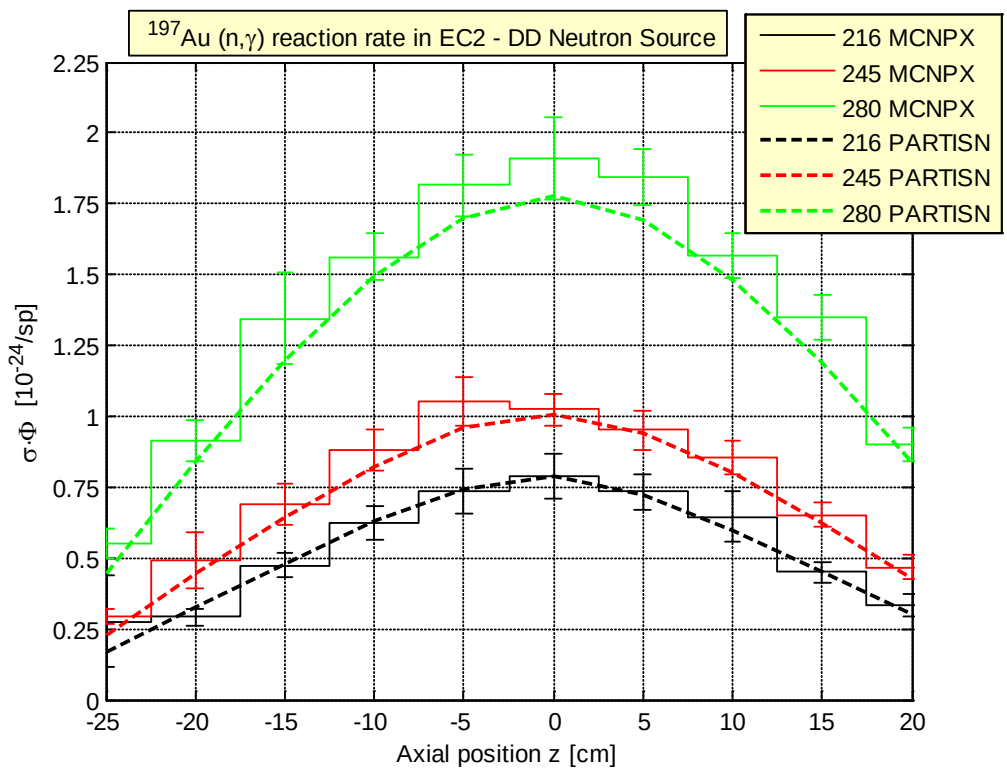

Figure 137. ^{197}Au (n,γ) reaction rate per source particle in the EC2 experimental channel with D-D neutron source calculated by MCNPX and PARTISN

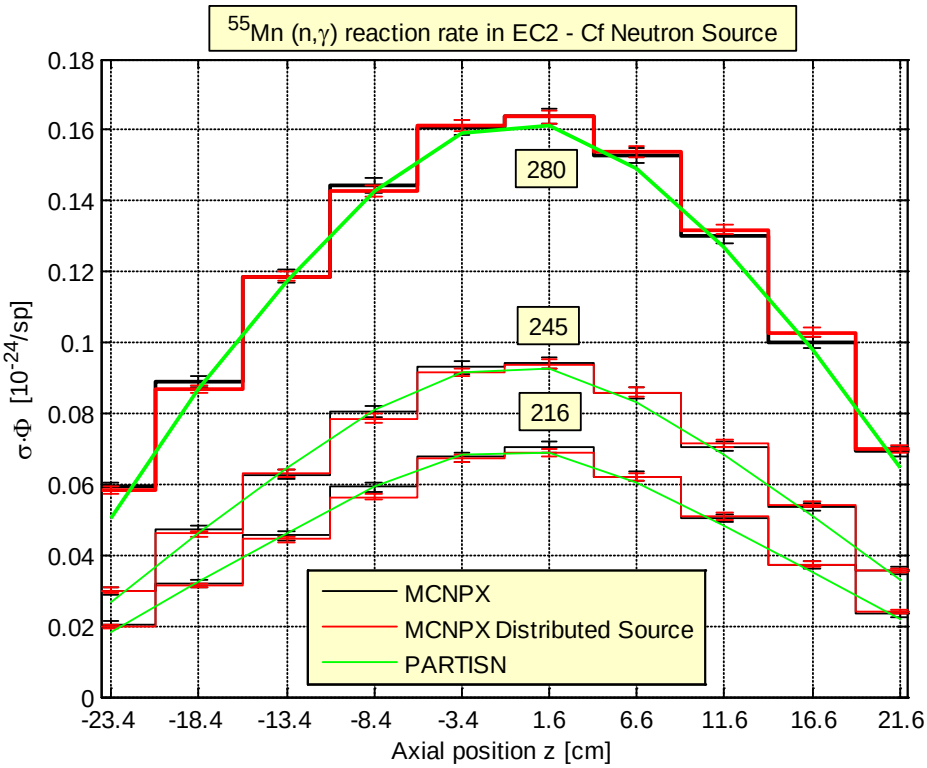

Figure 138. ^{55}Mn (n,γ) reaction rate per source particle in the EC2 experimental channel with Cf neutron source calculated by MCNPX and PARTISN

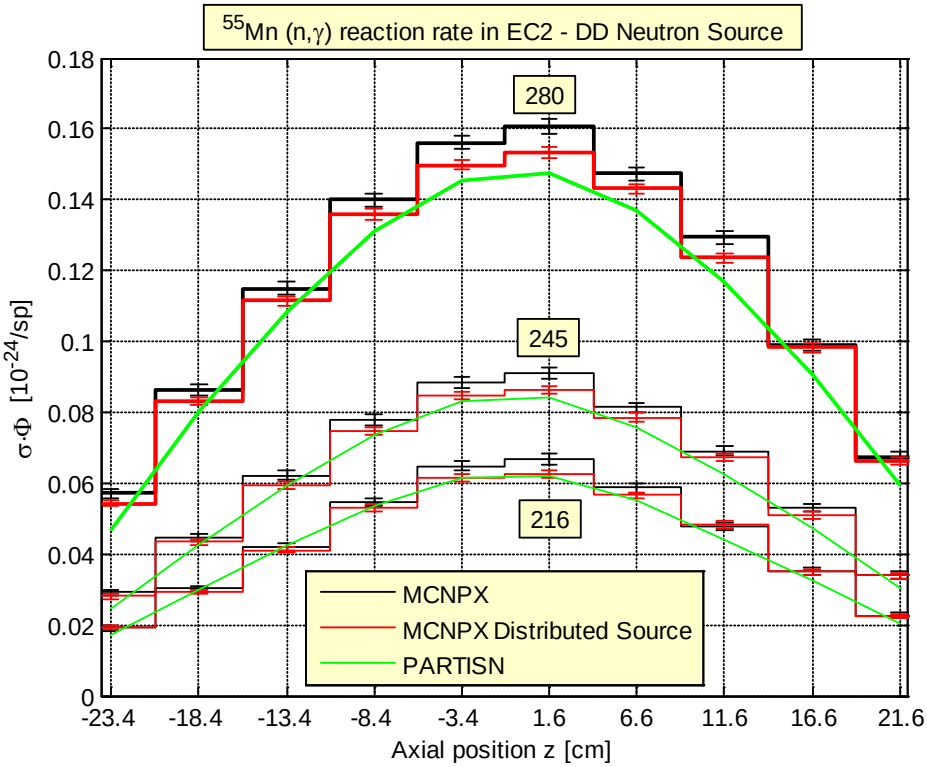

Figure 139. ^{55}Mn (n,γ) reaction rate per source particle in the EC2 experimental channel with D-D neutron source calculated by MCNPX and PARTISN

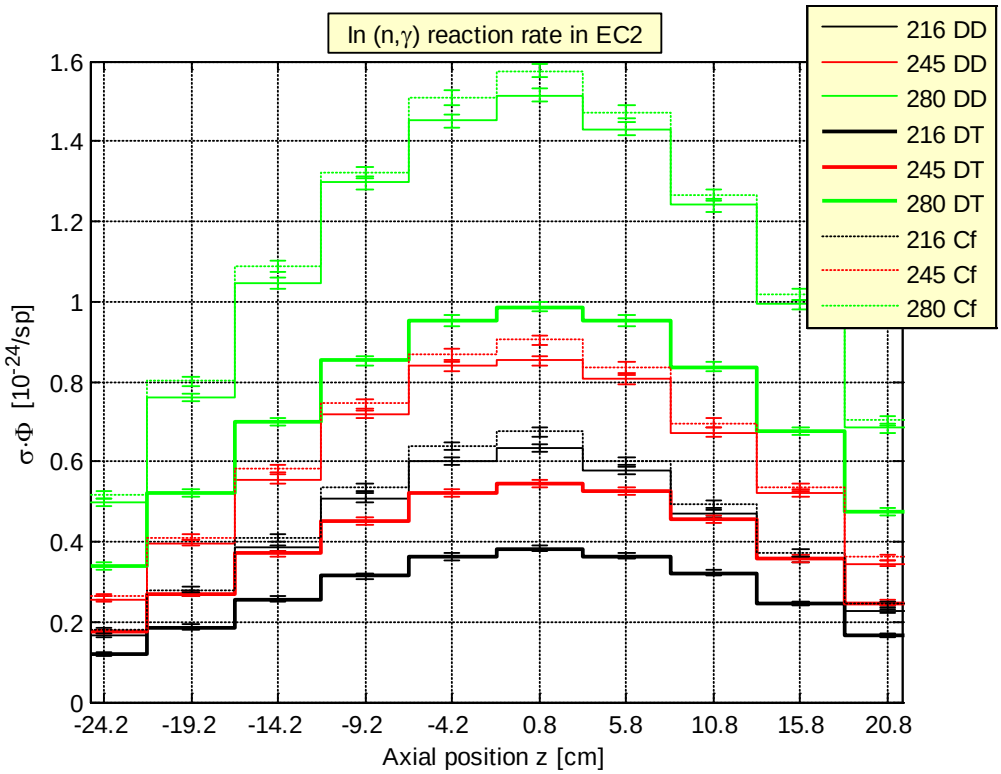

Figure 140. In (n,γ) reaction rate per source particle in the EC2 experimental channel with explicit modeling of the irradiation samples calculated by MCNPX

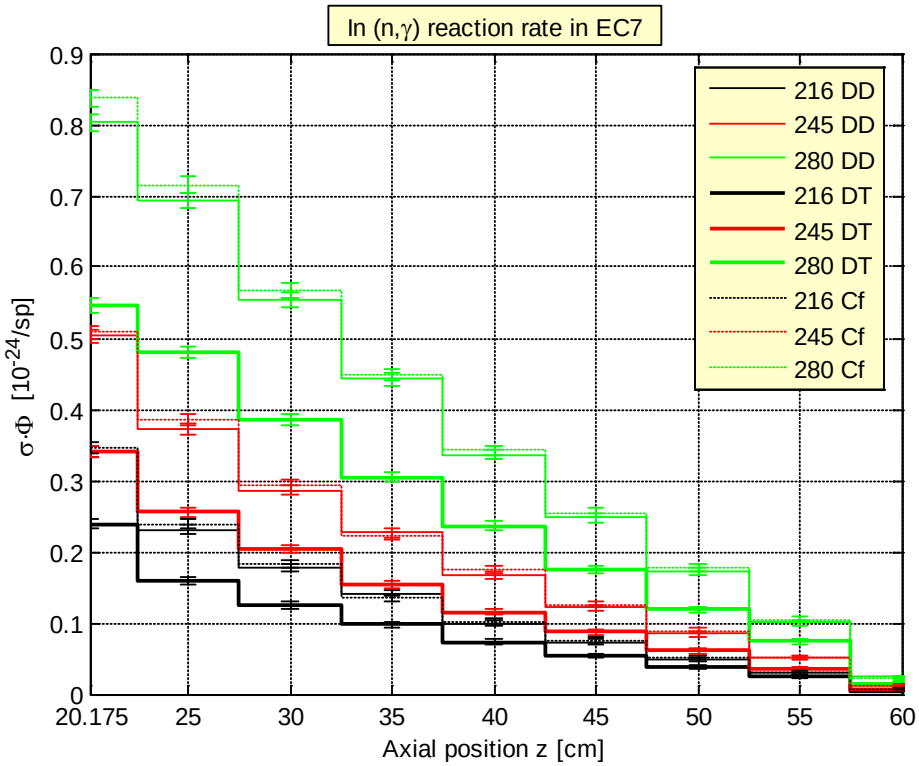

Figure 141. In (n,γ) reaction rate per source particle in the EC7 experimental channel with explicit modeling of the irradiation samples calculated by MCNPX

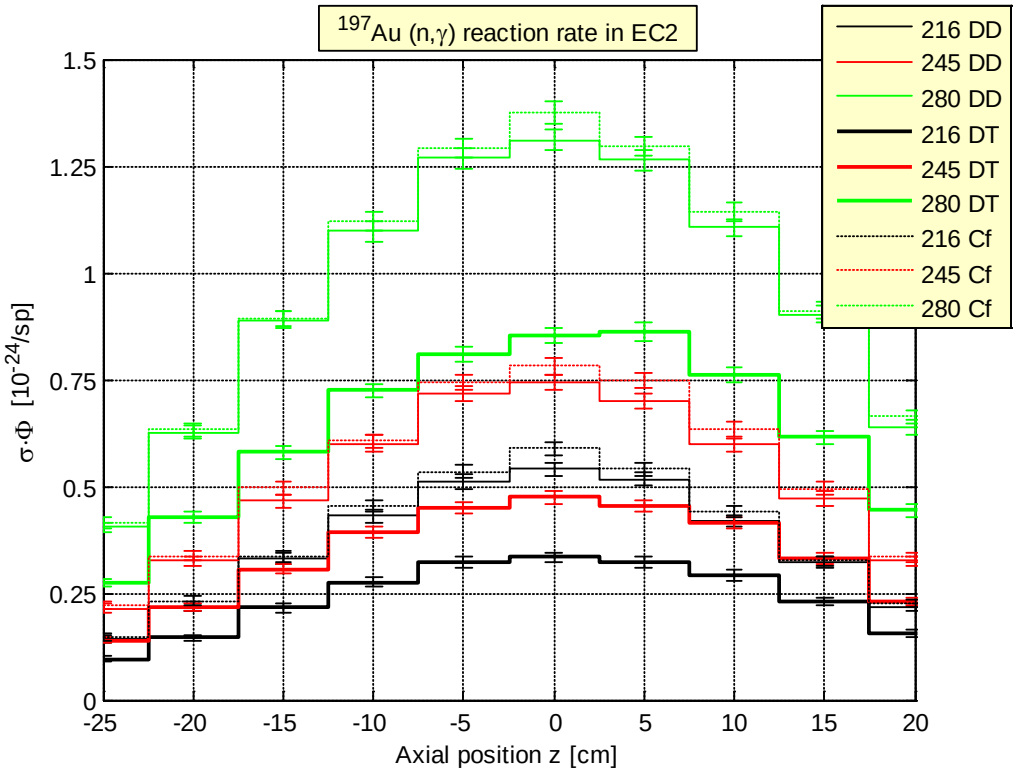

Figure 142. ^{197}Au (n,γ) reaction rate per source particle in the EC2 experimental channel with explicit modeling of the irradiation samples calculated by MCNPX

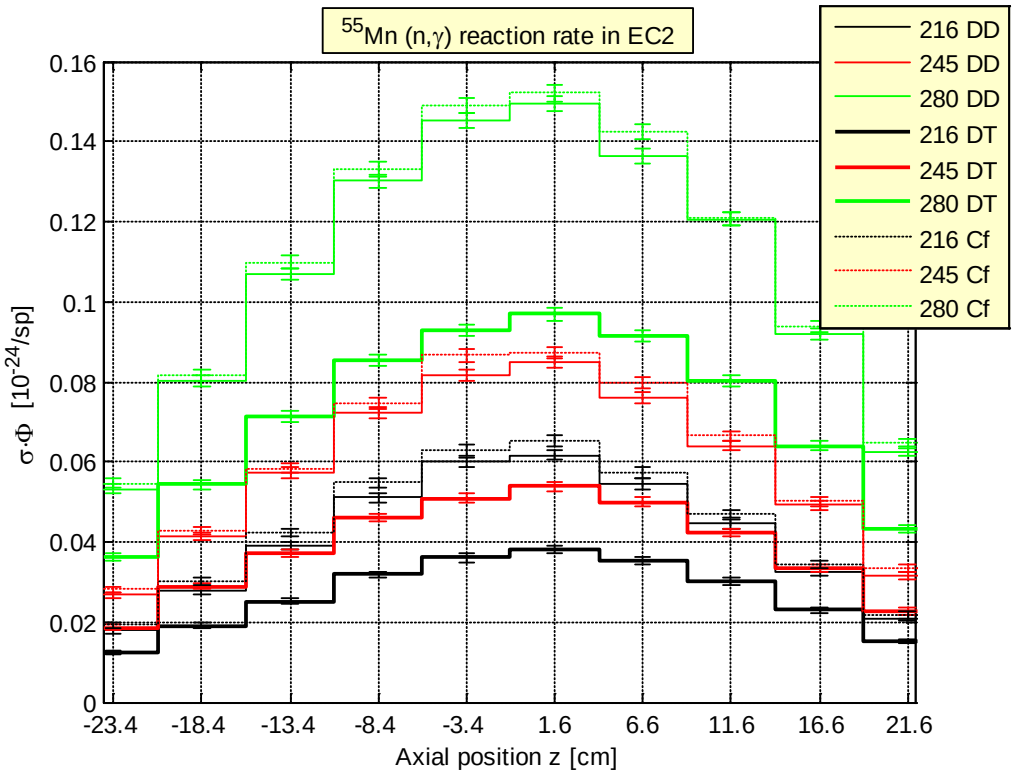

Figure 143. ^{55}Mn (n,γ) reaction rate per source particle in the EC2 experimental channel with explicit modeling of the irradiation samples calculated by MCNPX

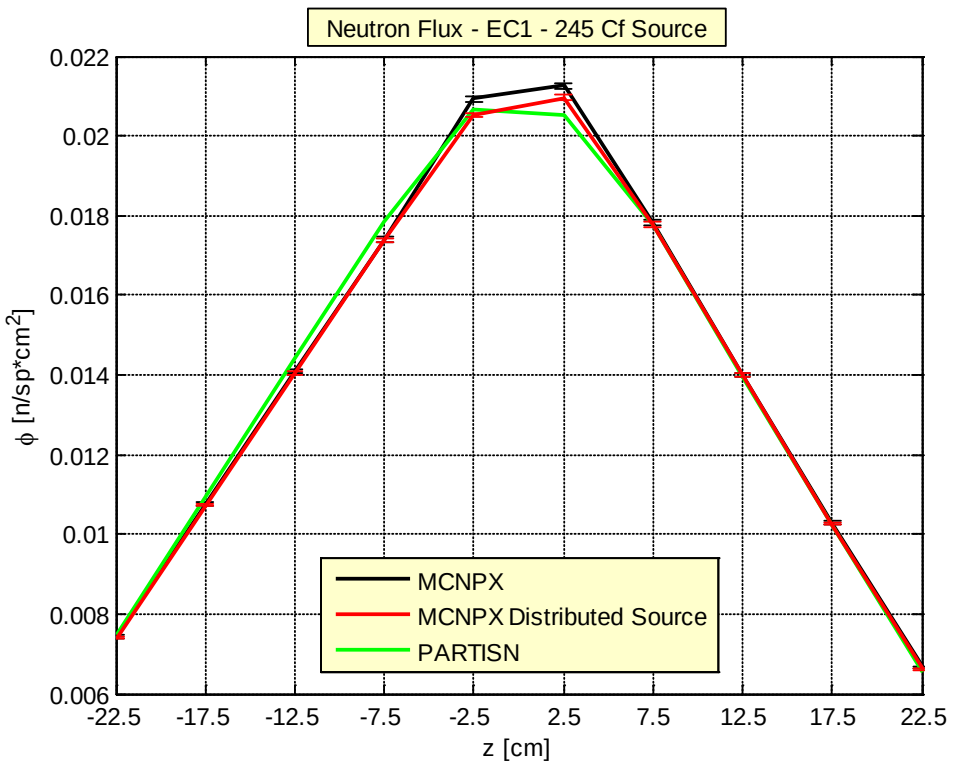

Figure 144. Comparison of MCNP and PARTISN neutron flux in the EC1 experimental channel of YALINA thermal with 245 fuel rods from Cf neutron source

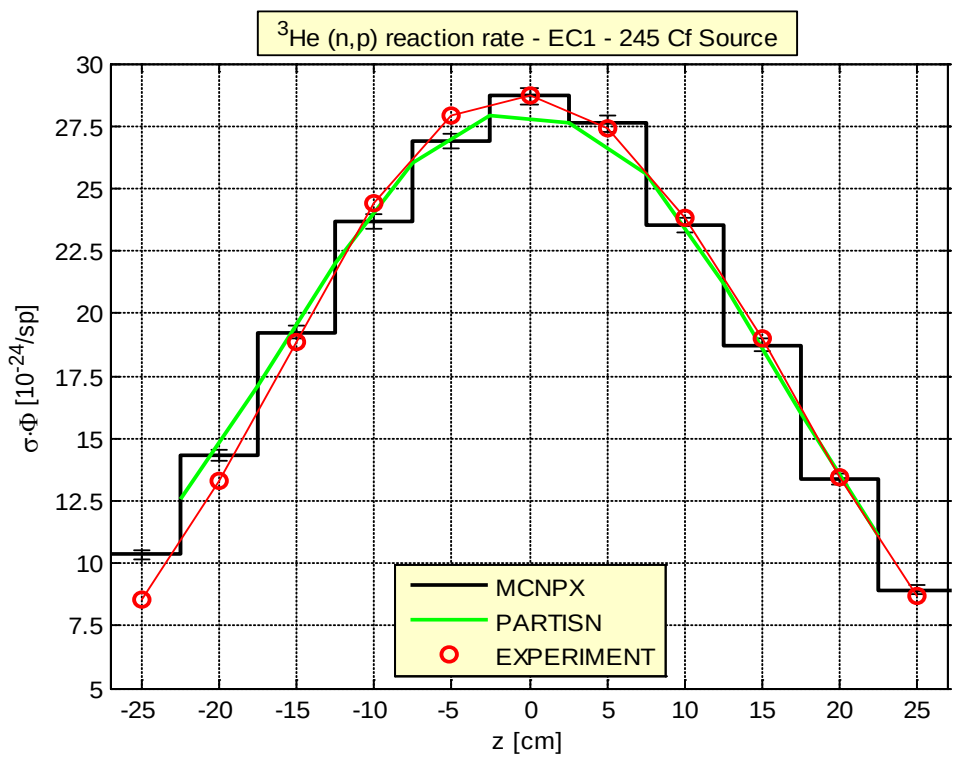

Figure 145. Comparison of MCNP, PARTISN and experimental results for the ^{3}He reaction rate in the EC1 experimental channel of YALINA thermal with 245 fuel rods from Cf neutron source

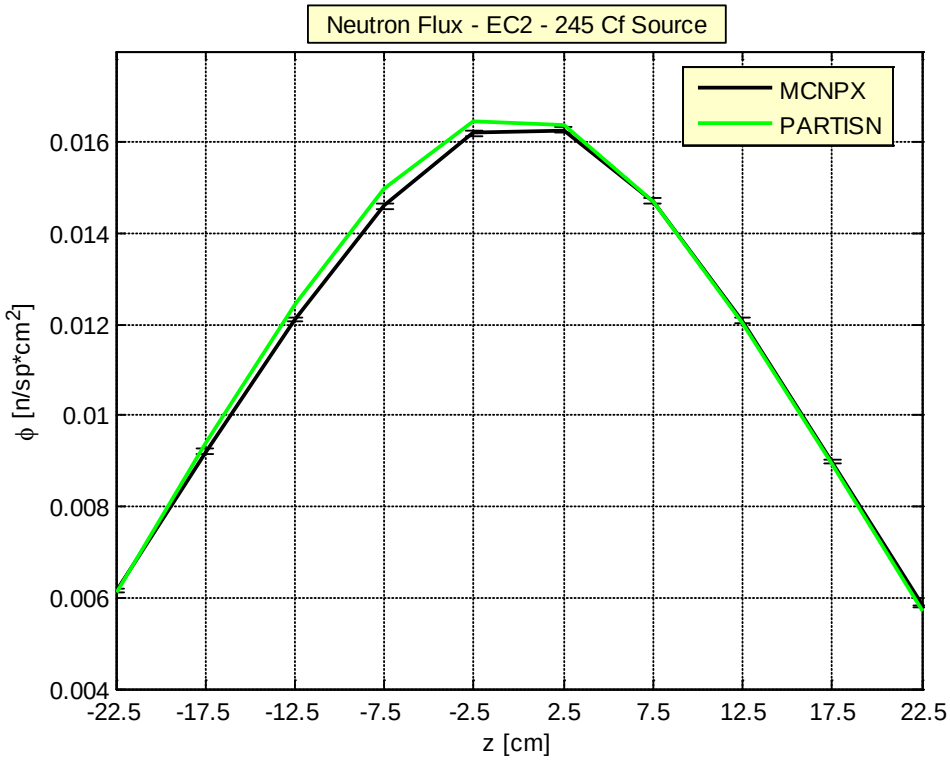

Figure 146. Comparison of MCNP and PARTISN neutron flux in the EC2 experimental channel of the YALINA thermal with 245 fuel rods from Cf neutron source

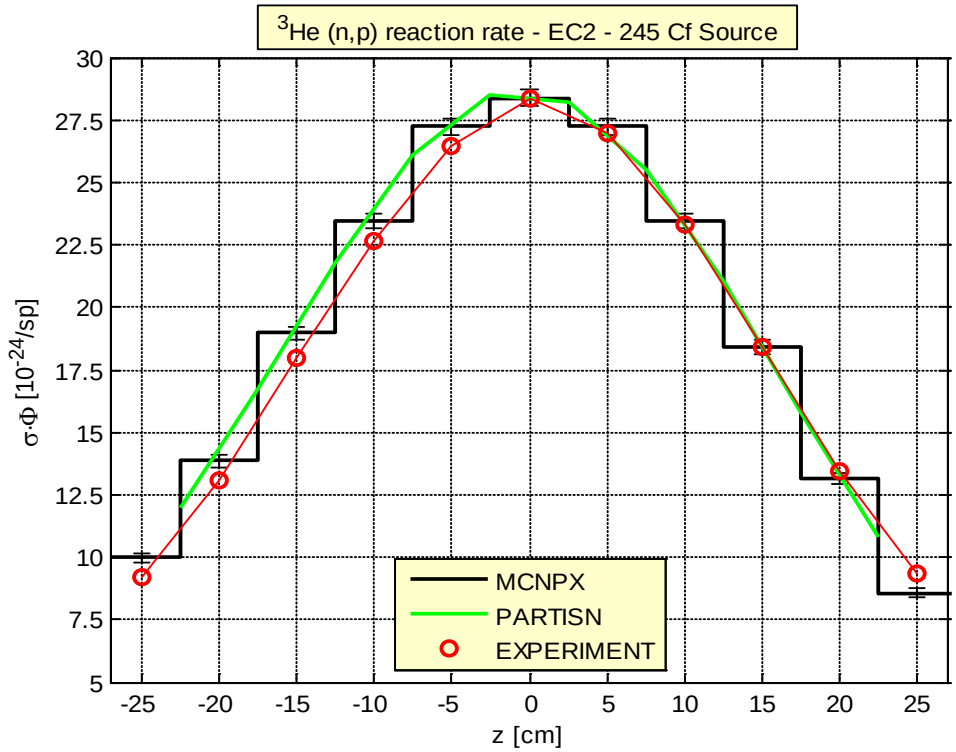

Figure 147. Comparison of MCNP, PARTISN and experimental results for the ^{3}He reaction rate in the EC2 experimental channel of the YALINA thermal with 245 fuel rods from Cf neutron source

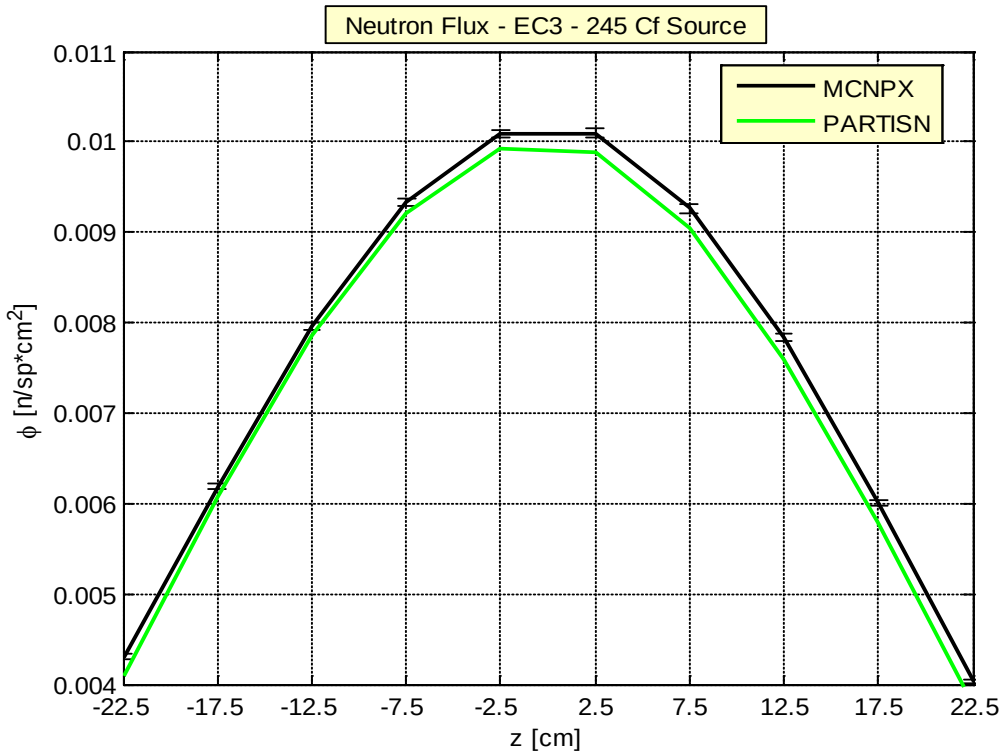

Figure 148. Comparison of MCNP and PARTISN neutron flux in the EC3 experimental channel of the YALINA thermal with 245 fuel rods from Cf neutron source

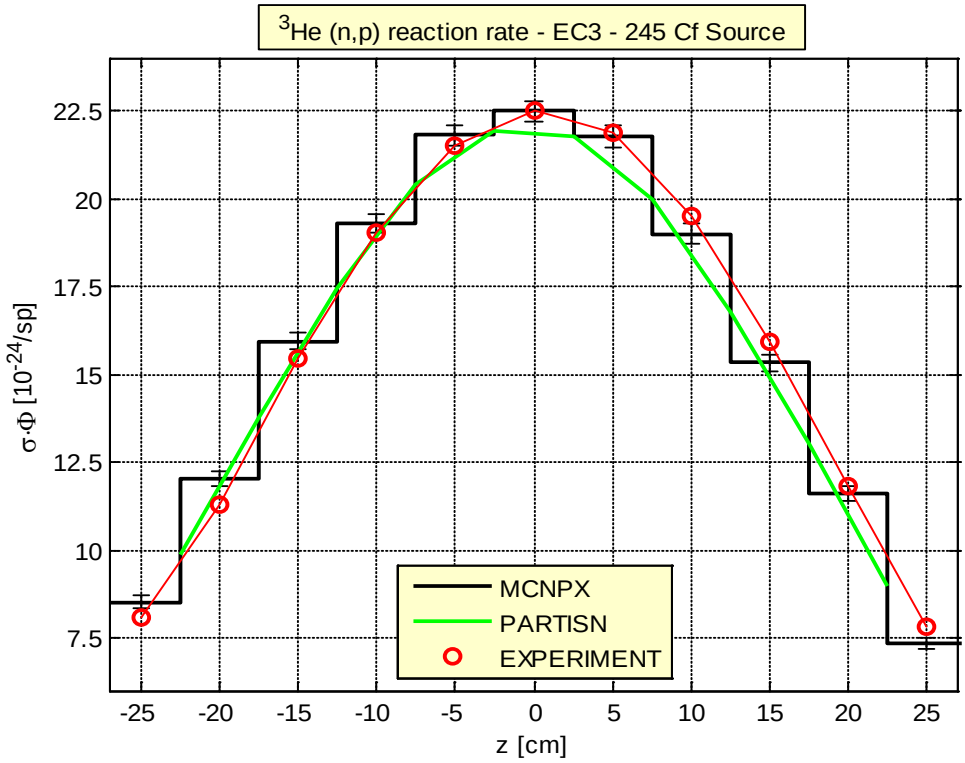

Figure 149. Comparison of MCNP, PARTISN and experimental results for the ^{3}He reaction rate in the EC3 experimental channel of the YALINA thermal with 245 fuel rods from Cf neutron source

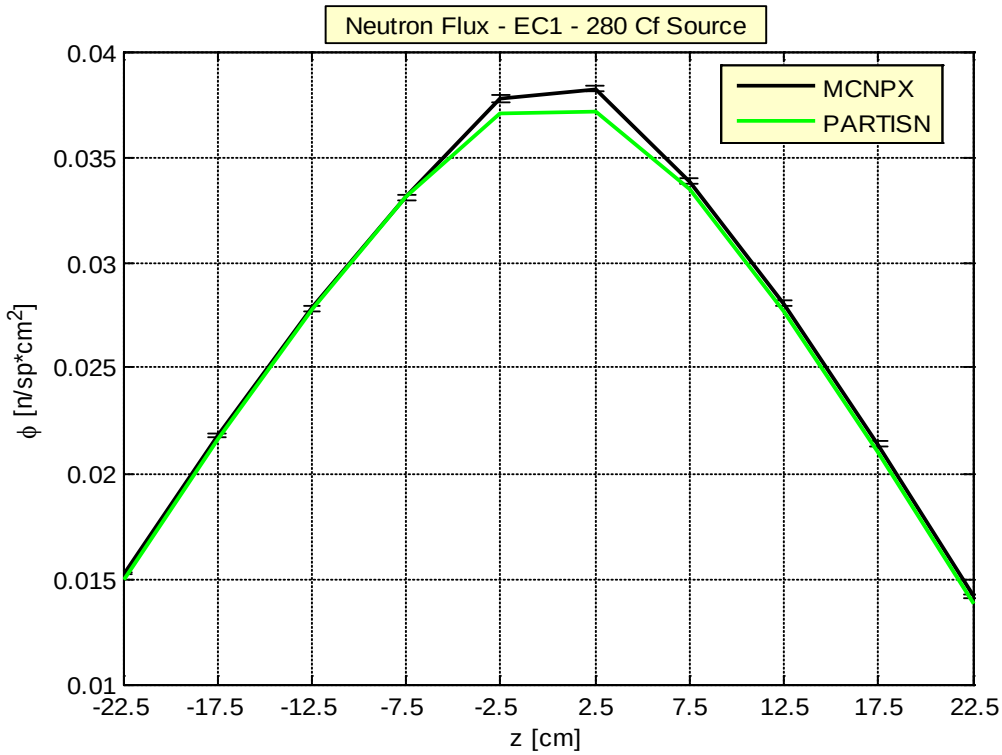

Figure 150. Comparison of MCNP and PARTISN neutron flux in the EC1 experimental channel of the YALINA thermal with 280 fuel rods from Cf neutron source

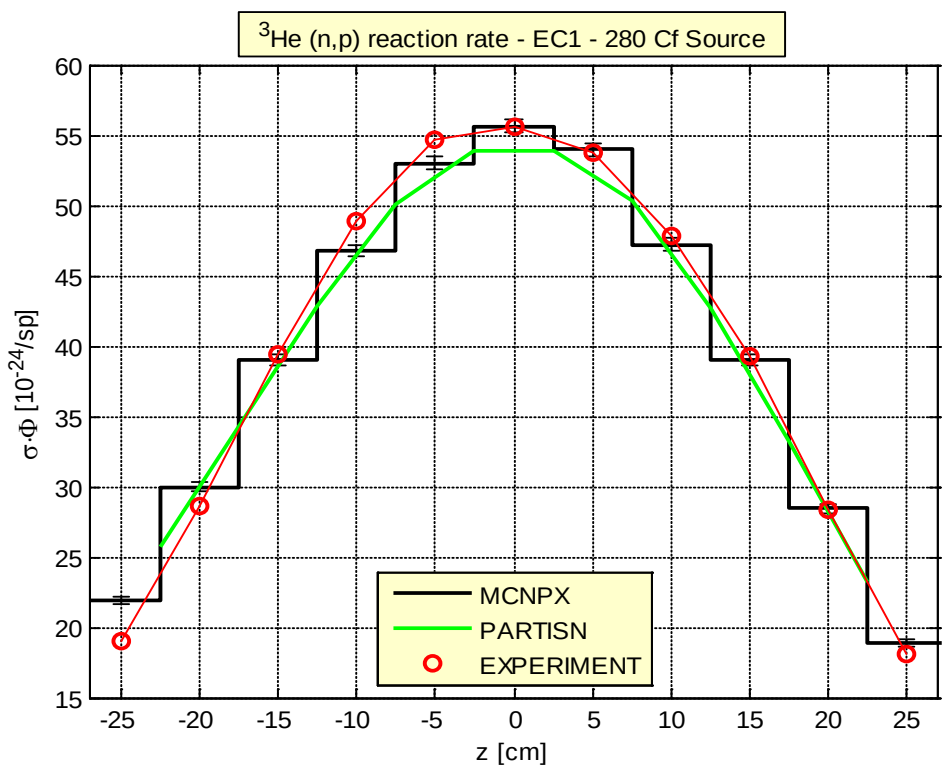

Figure 151. Comparison of MCNP, PARTISN and experimental results for the ^{3}He reaction rate in the EC1 experimental channel of the YALINA thermal with 280 fuel rods from Cf neutron source

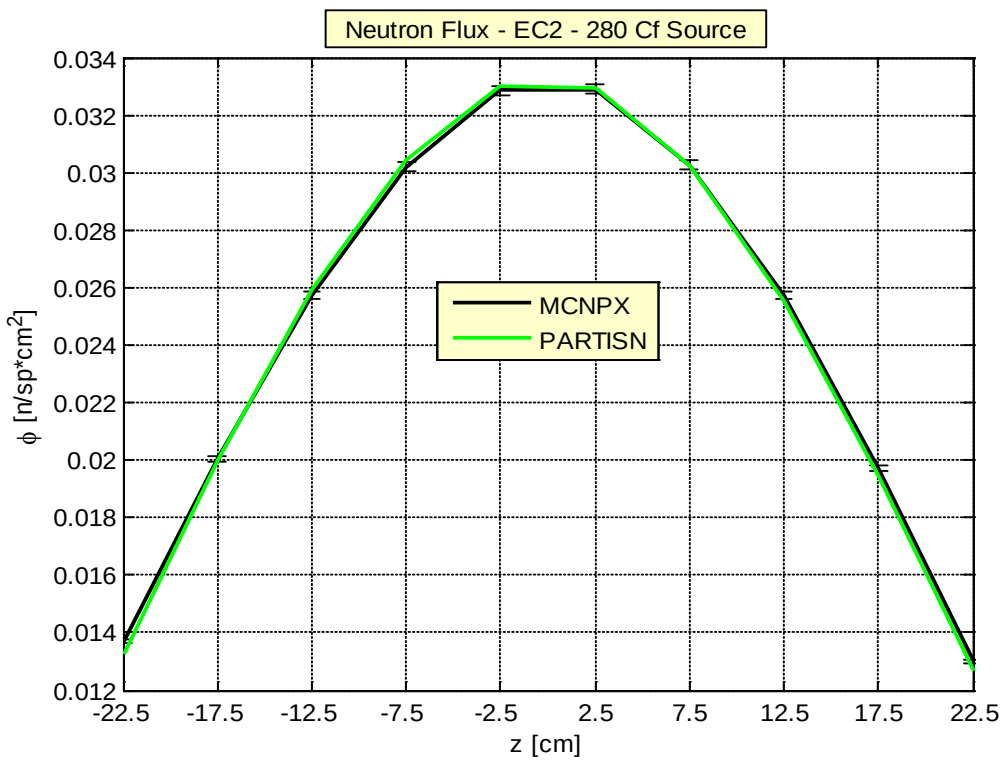

Figure 152. Comparison of MCNP and PARTISN neutron flux in the EC2 experimental channel of the YALINA thermal with 280 fuel rods from Cf neutron source

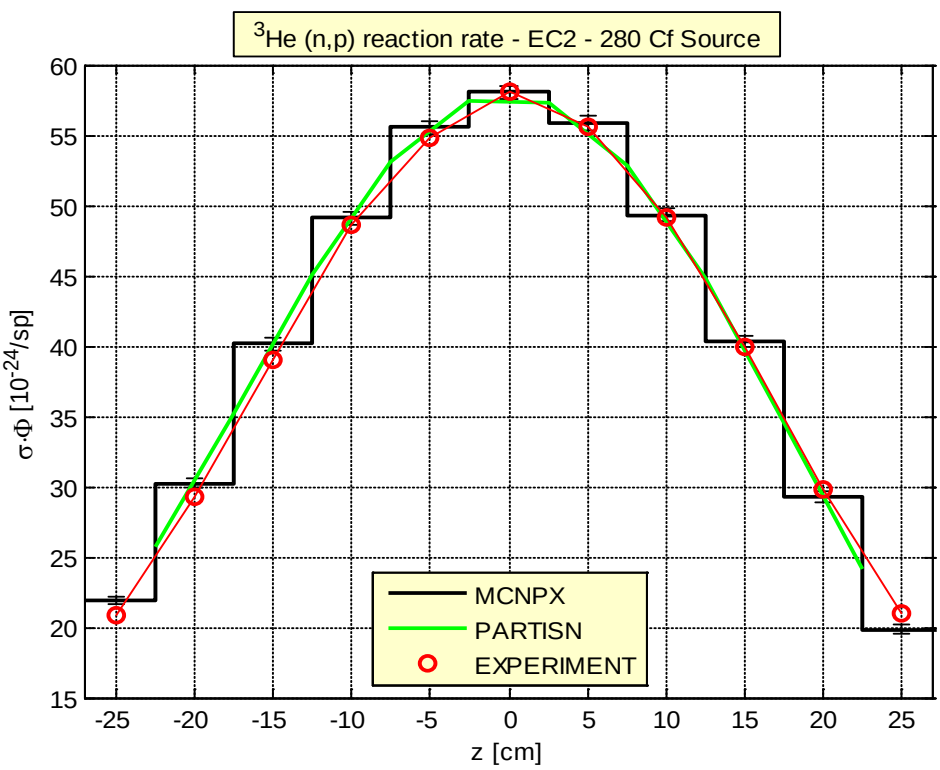

Figure 153. Comparison of MCNP, PARTISN and experimental results for the ^{3}He reaction rate in the EC2 experimental channel of the YALINA thermal with 280 fuel rods from Cf neutron source

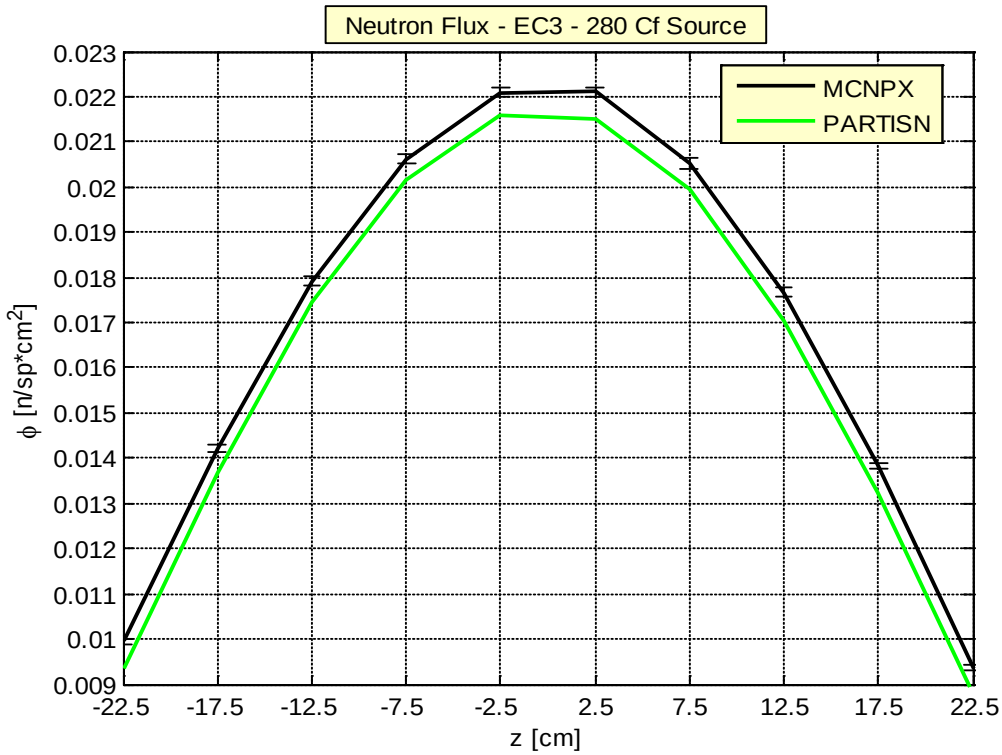

Figure 154. Comparison of MCNP and PARTISN neutron flux in the EC3 experimental channel of the YALINA thermal with 280 fuel rods from Cf neutron source

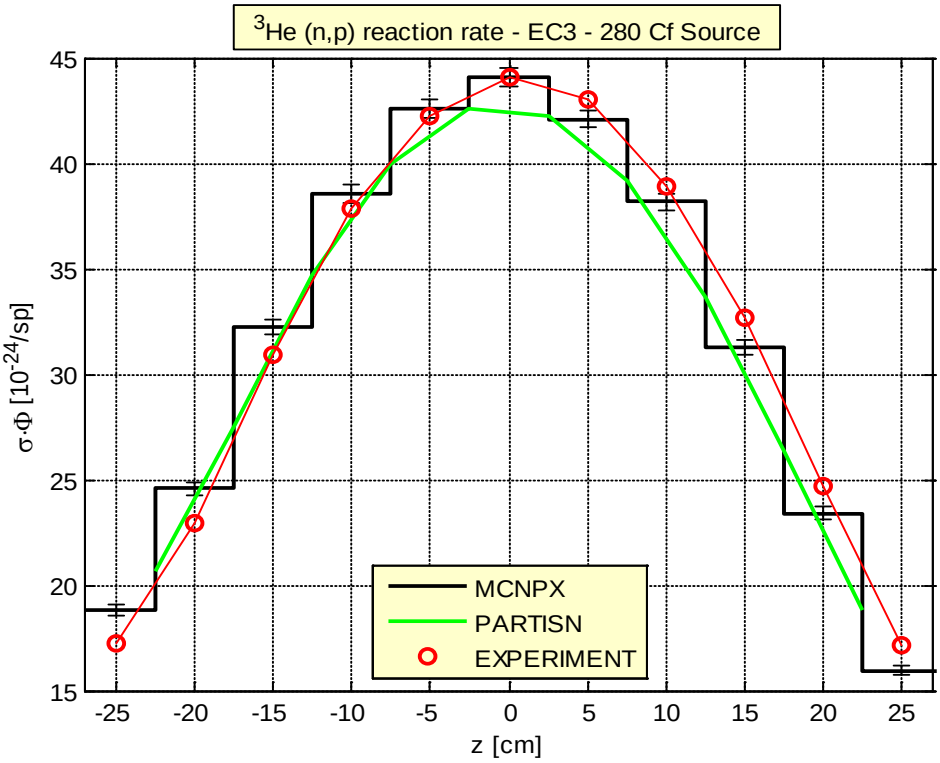

Figure 155. Comparison of MCNP, PARTISN and experimental results for the ^{3}He reaction rate in the EC3 experimental channel of the YALINA thermal with 280 fuel rods from Cf neutron source

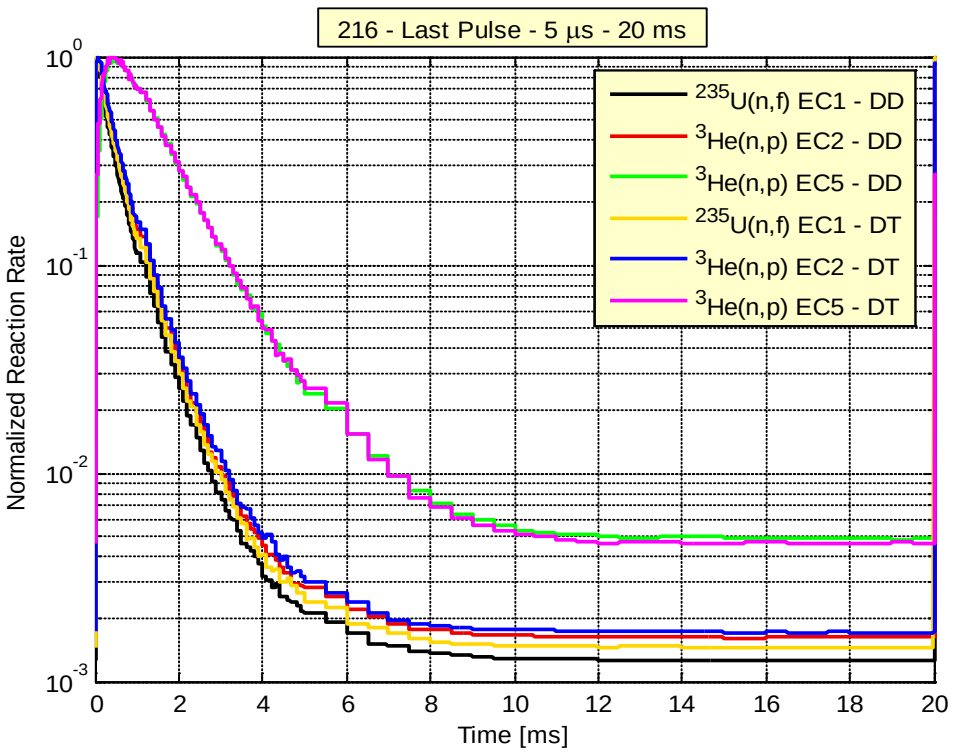

Figure 156. ^{3}He(n,p) and ^{235}U(n,f) reaction rates in the YALINA thermal with 216 fuel rods from D-D and D-T pulsed neutron sources

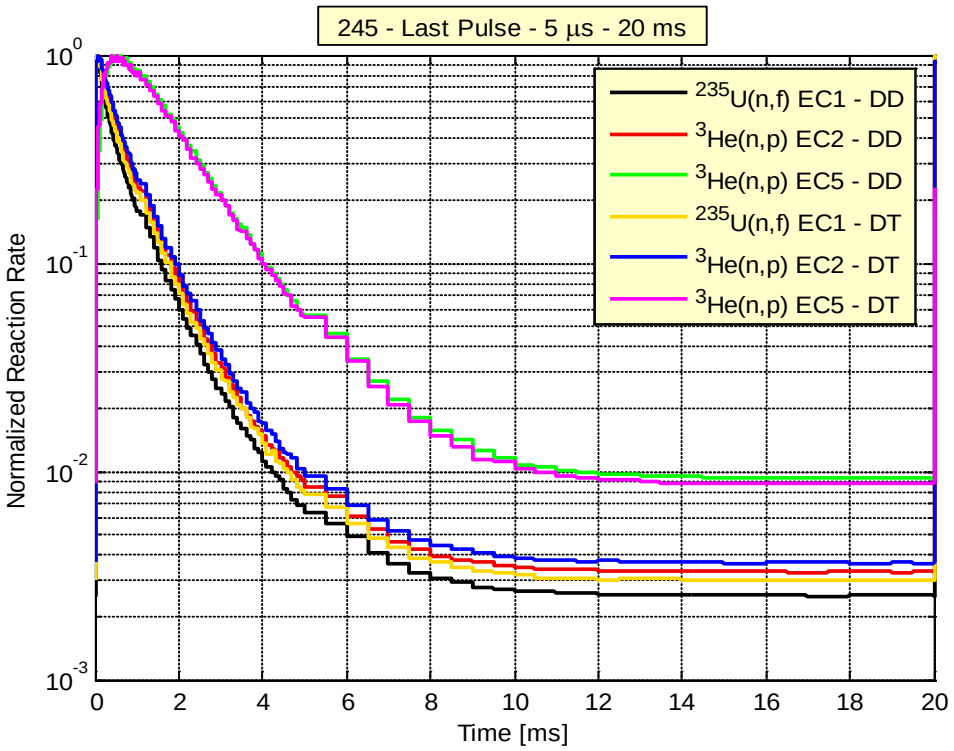

Figure 157. ^{3}He(n,p) and ^{235}U(n,f) reaction rates in the YALINA thermal with 245 fuel rods from D-D and D-T pulsed neutron sources

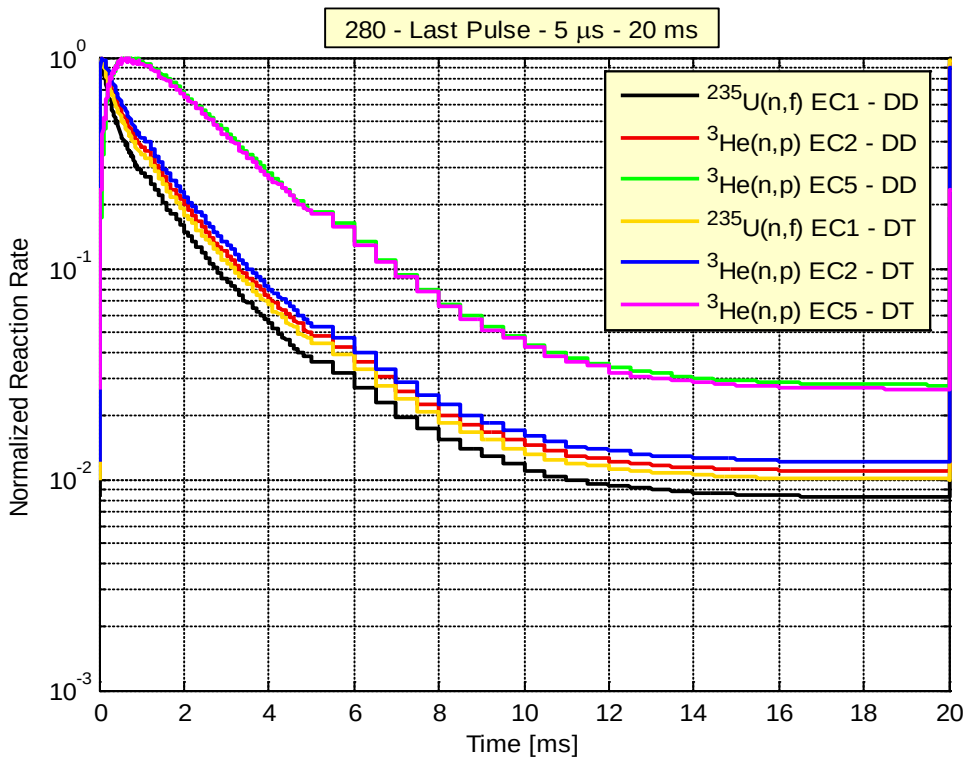

Figure 158. ^{3}He(n,p) and ^{235}U(n,f) reaction rates in the YALINA thermal with 280 fuel rods from D-D and D-T pulsed neutron sources

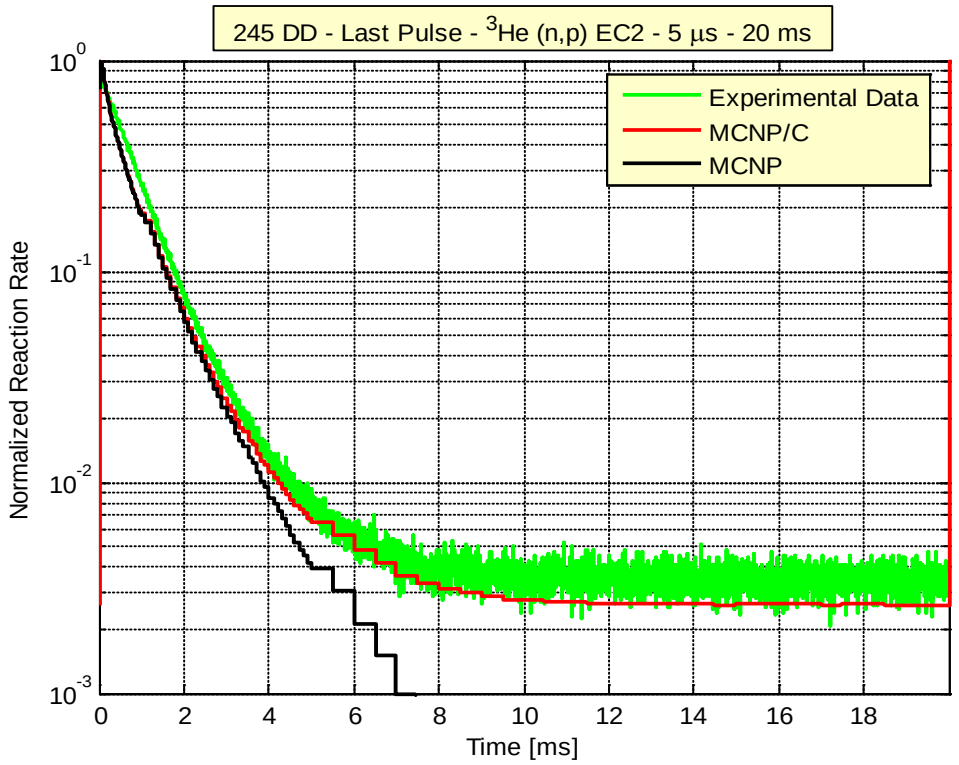

Figure 159. ^{3}He(n,p) reaction rate in the EC2 experimental channel of YALINA thermal with 245 fuel rods from D-D pulsed neutron source

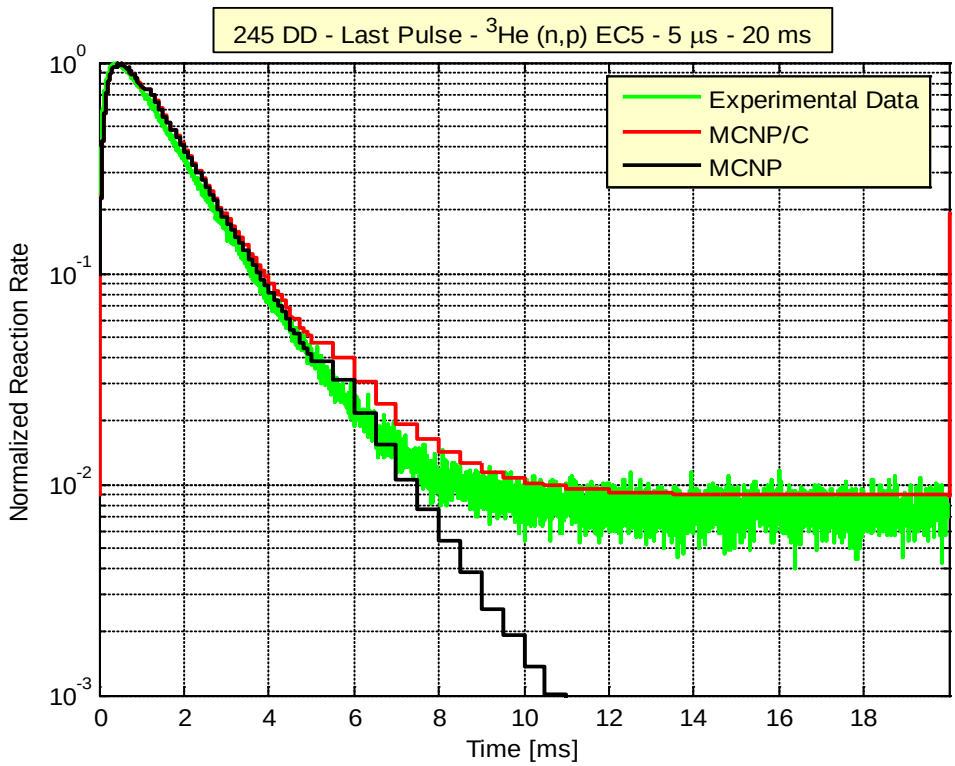

Figure 160. ^{3}He(n,p) reaction rate in the EC5 experimental channel of YALINA thermal with 245 fuel rods from D-D pulsed neutron source

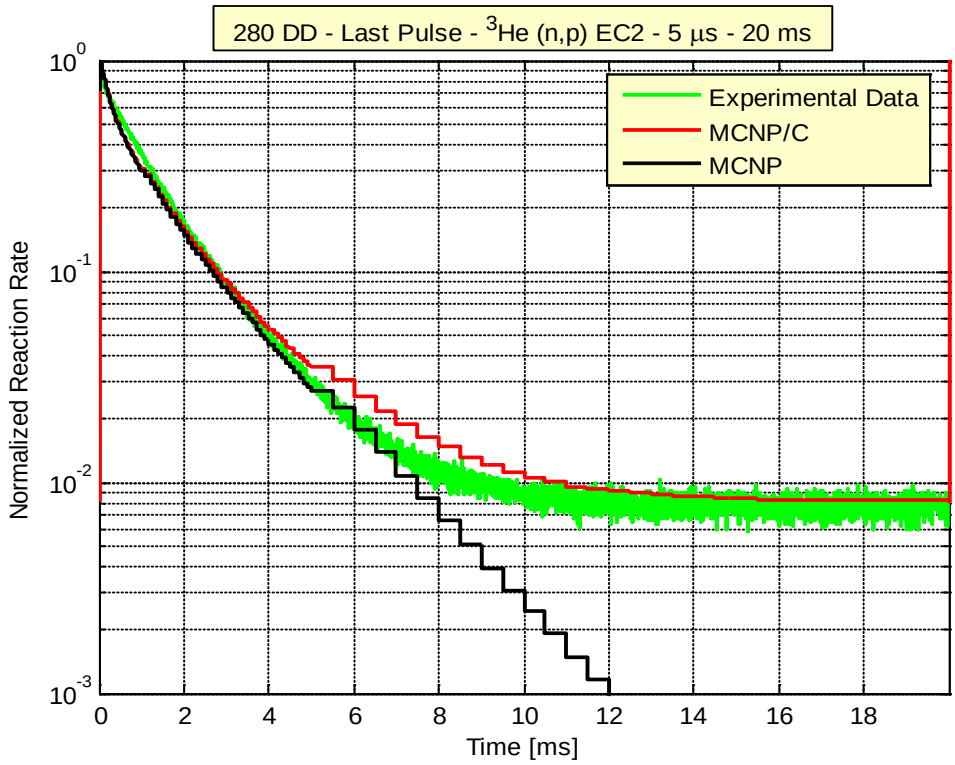

Figure 161. ^{3}He(n,p) reaction rate in the EC2 experimental channel of the YALINA thermal with 280 fuel rods from D-D pulsed neutron source

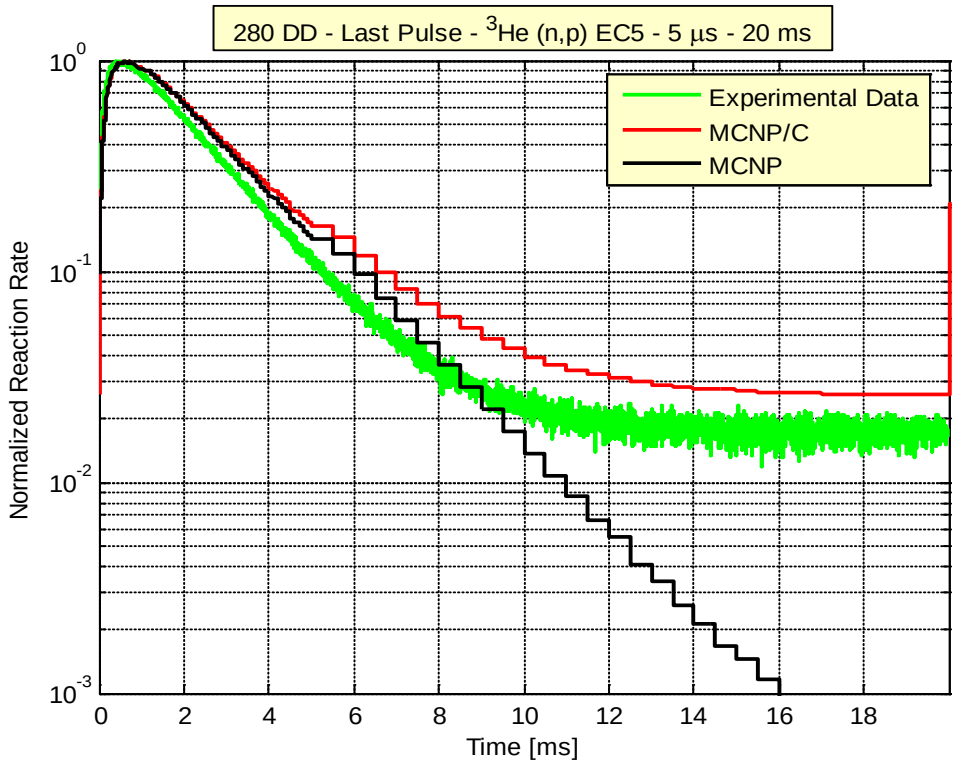

Figure 162. ^{3}He(n,p) reaction rate in the EC5 experimental channel of YALINA thermal with 280 fuel rods from D-D pulsed neutron source

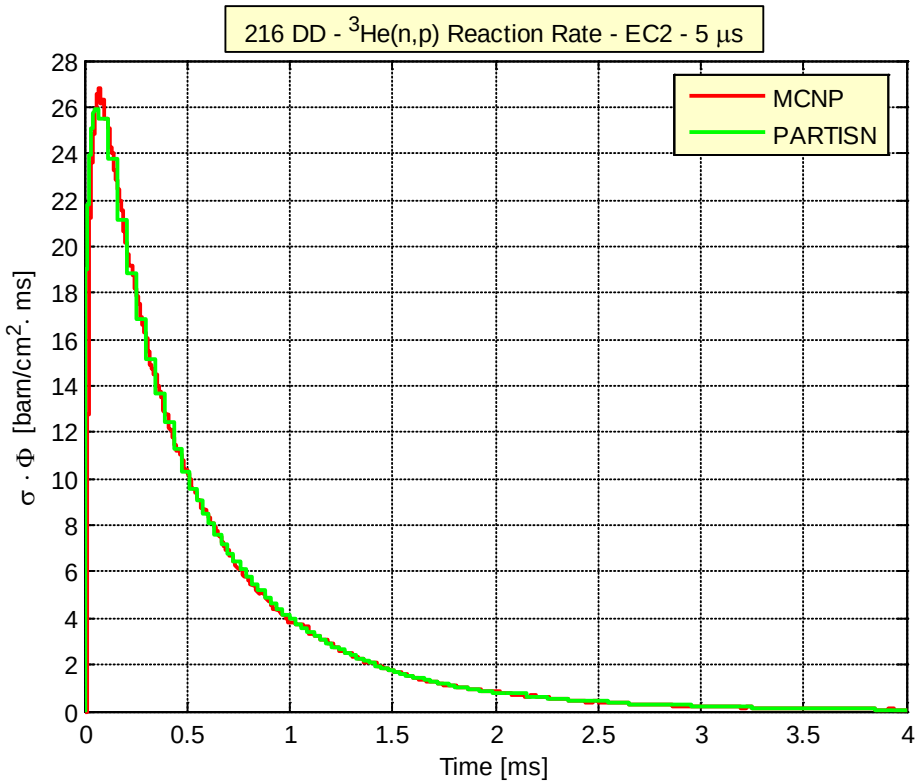

Figure 163. Comparison of MCNP/MCNPX and PARTISN results for ^{3}He(n,p) reaction rate in the EC2 experimental channel of YALINA thermal with 216 fuel rods from D-D single pulse

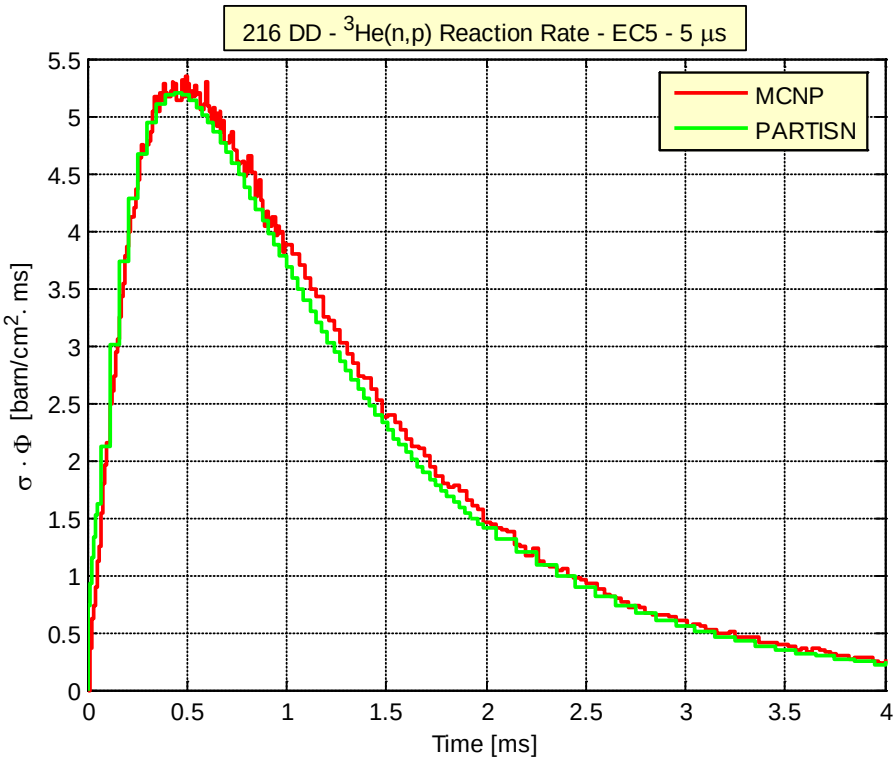

Figure 164. Comparison of MCNP/MCNPX and PARTISN results for ^{3}He(n,p) reaction rate in the EC5 experimental channel of YALINA thermal with 216 fuel rods from D-D single pulse

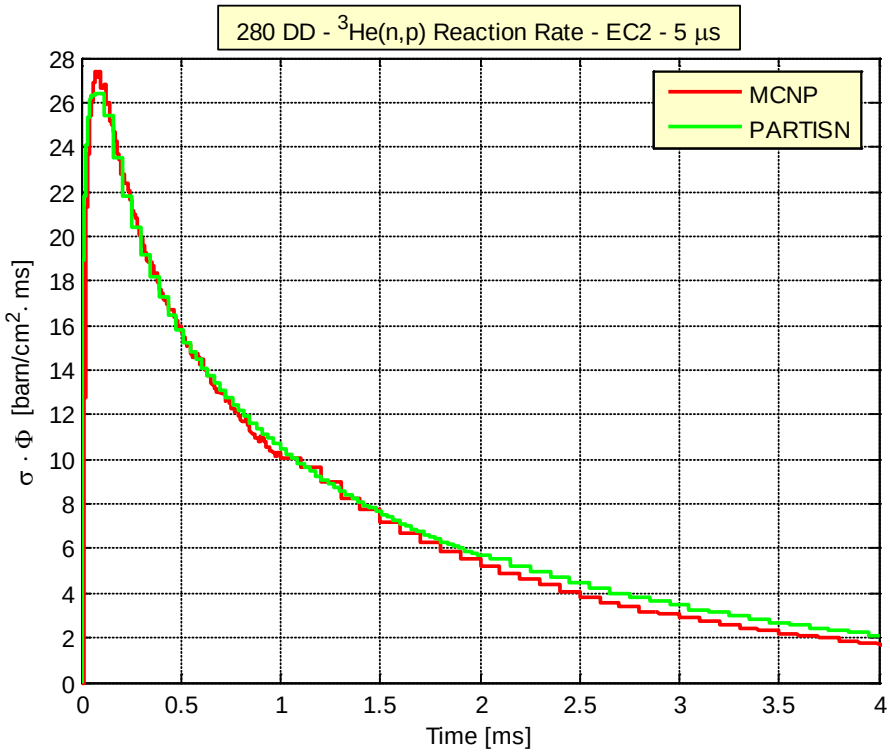

Figure 165. Comparison of MCNP/MCNPX and PARTISN results for ^{3}He(n,p) reaction rate in the EC2 experimental channel of YALINA thermal with 280 fuel rods from D-D single pulse

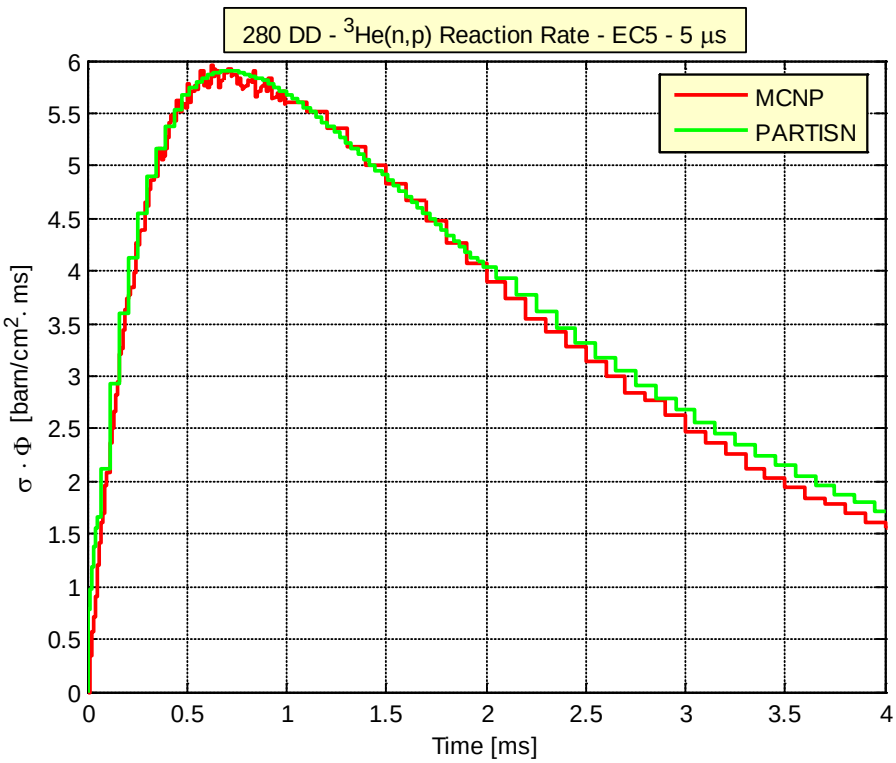

Figure 166. Comparison of MCNP/MCNPX and PARTISN results for ^{3}He(n,p) reaction rate in the EC5 experimental channel of YALINA thermal with 280 fuel rods from D-D single pulse

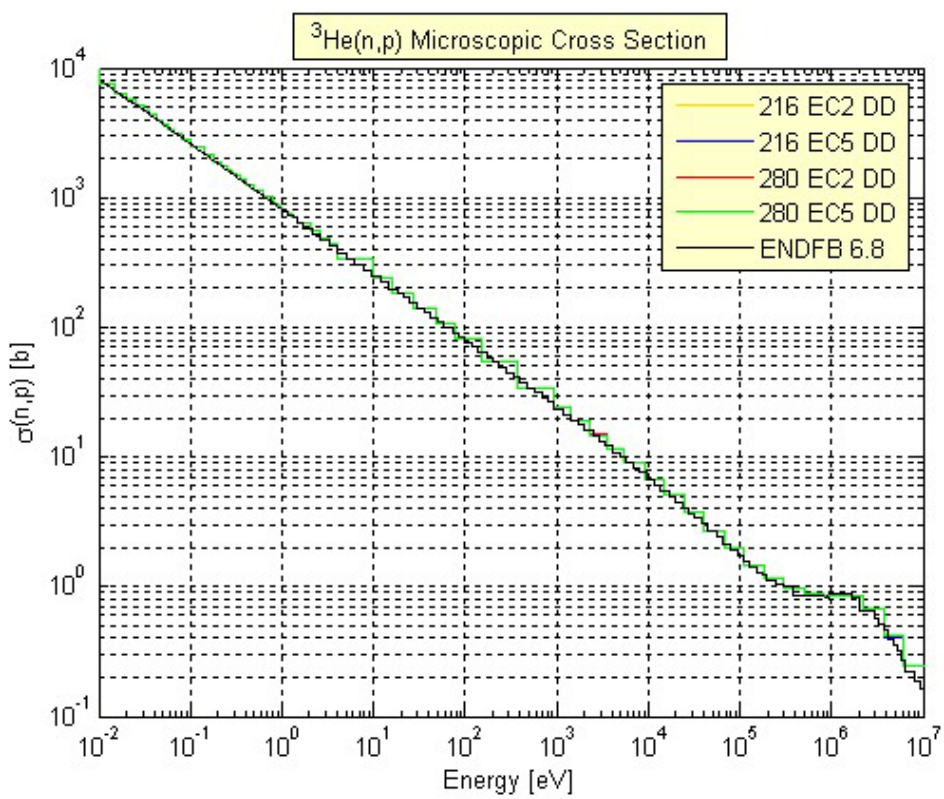

Figure 167. ^{3}He(n,p) microscopic cross section from MCNPX calculations and from ENDF/B-VI.8 nuclear data files

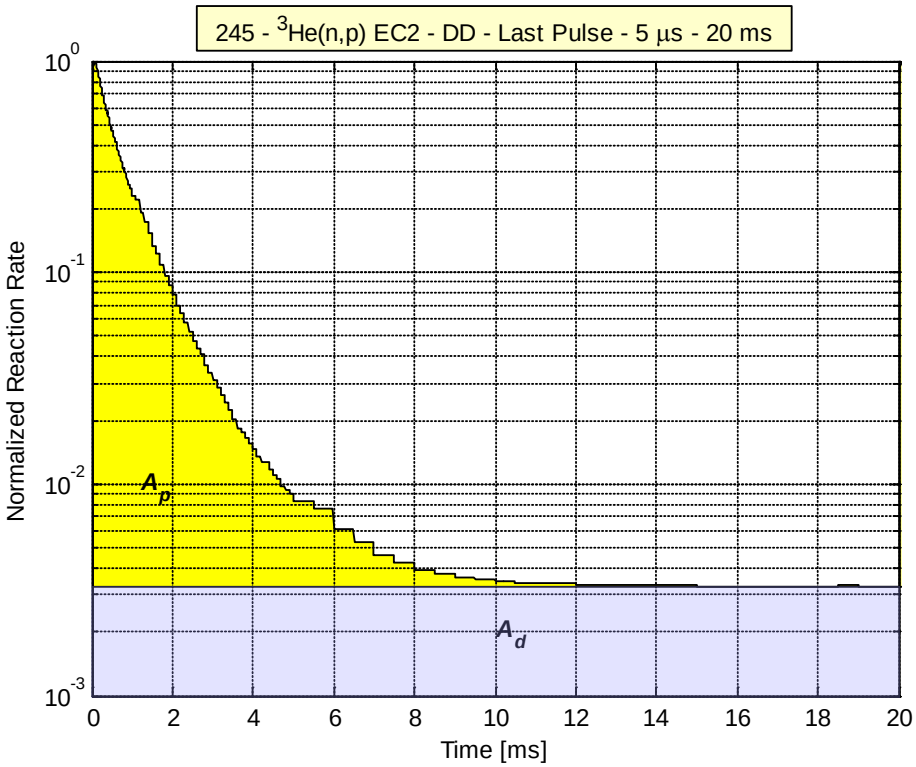

Figure 168. Illustration of A_p and A_d of the area method

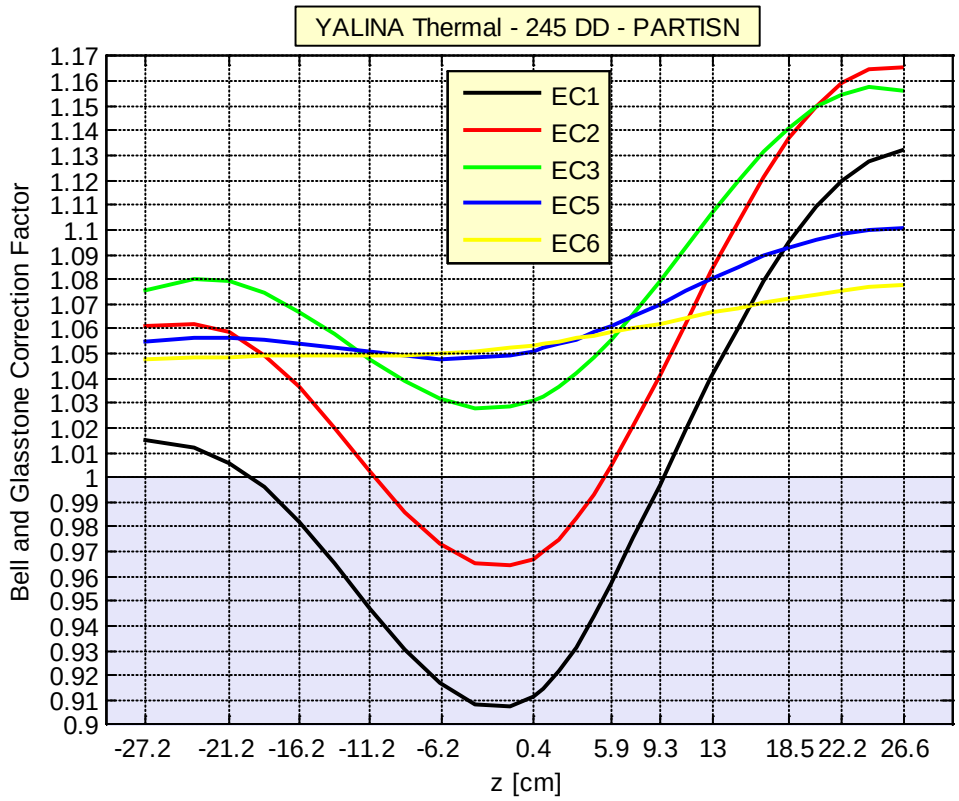

Figure 169. Correction factor as a function of the axial position calculated by PARTISN for the 245 configuration and the D-D neutron source

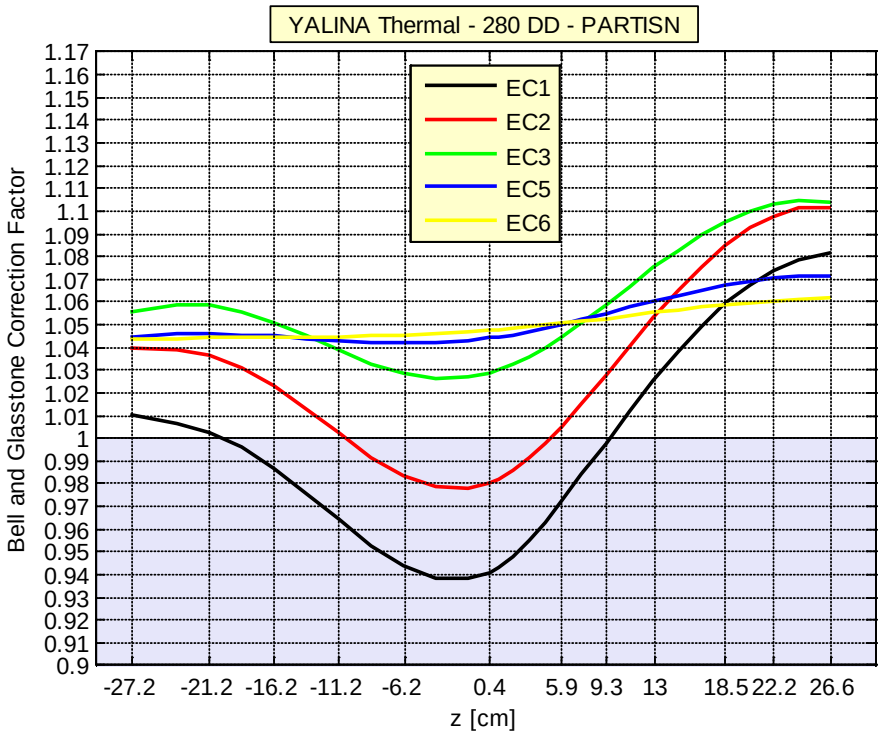

Figure 170. Correction factor as a function of the axial position calculated by PARTISN for the 280 configuration and the D-D neutron source

Appendix A: DRAGON Modeling

Three types of DRAGON models have been used to generate the macroscopic cross sections. The first model consists of a single EK10 fuel rod inside a 2 x 2 cm polyethylene square box using a two-dimensional geometry. This input generates the homogenized cross section for the fuel and moderator unit cell. Figure A1 illustrates the geometry used in DRAGON for this input. The fuel and the moderator zones have been split into multiple regions to better match the MCNP results. The fuel zone has been divided into 3 regions and the moderator zone into more than 20 regions. Reflected boundary conditions have been applied to all side boundaries of Figure A1.

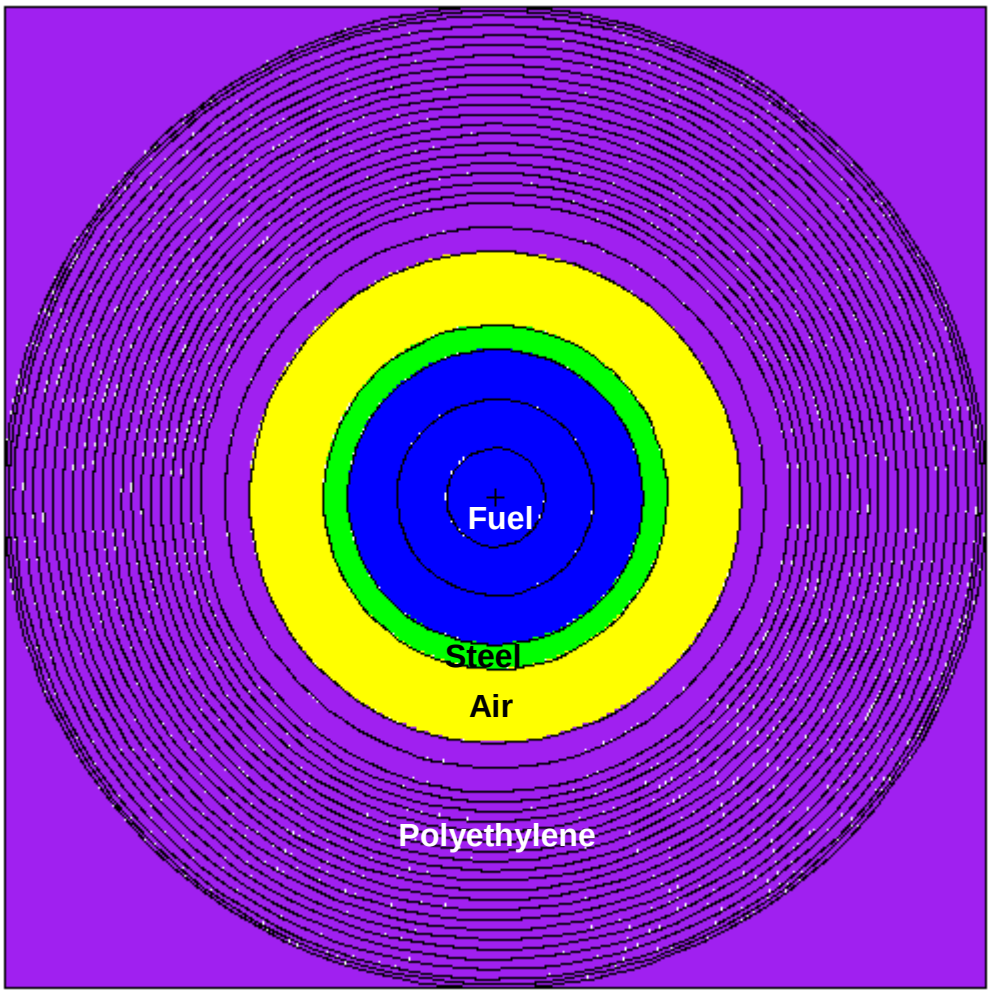

Figure A1. DRAGON geometry for the fuel and moderator unit cell

The second model represents one eight of the assembly including the target, fuel, and reflector zones using a two-dimensional geometry. This model has been modified several times to generate the homogenized cross sections for the fuel rods next a fuel hole, the fuel hole, the experimental and measurement channels, the fuel zone with the empty channels for the control rods, and the graphite reflector regions. This model has also been modified to

take into account the different number of fuel rods in the three fuel loading configurations (216, 245, and 280). Figure A2 shows the DRAGON geometry for this model when the target region is filled with lead. A reflected boundary condition has been applied to the right side; symmetric boundary conditions have been applied to the diagonal and bottom boundaries.

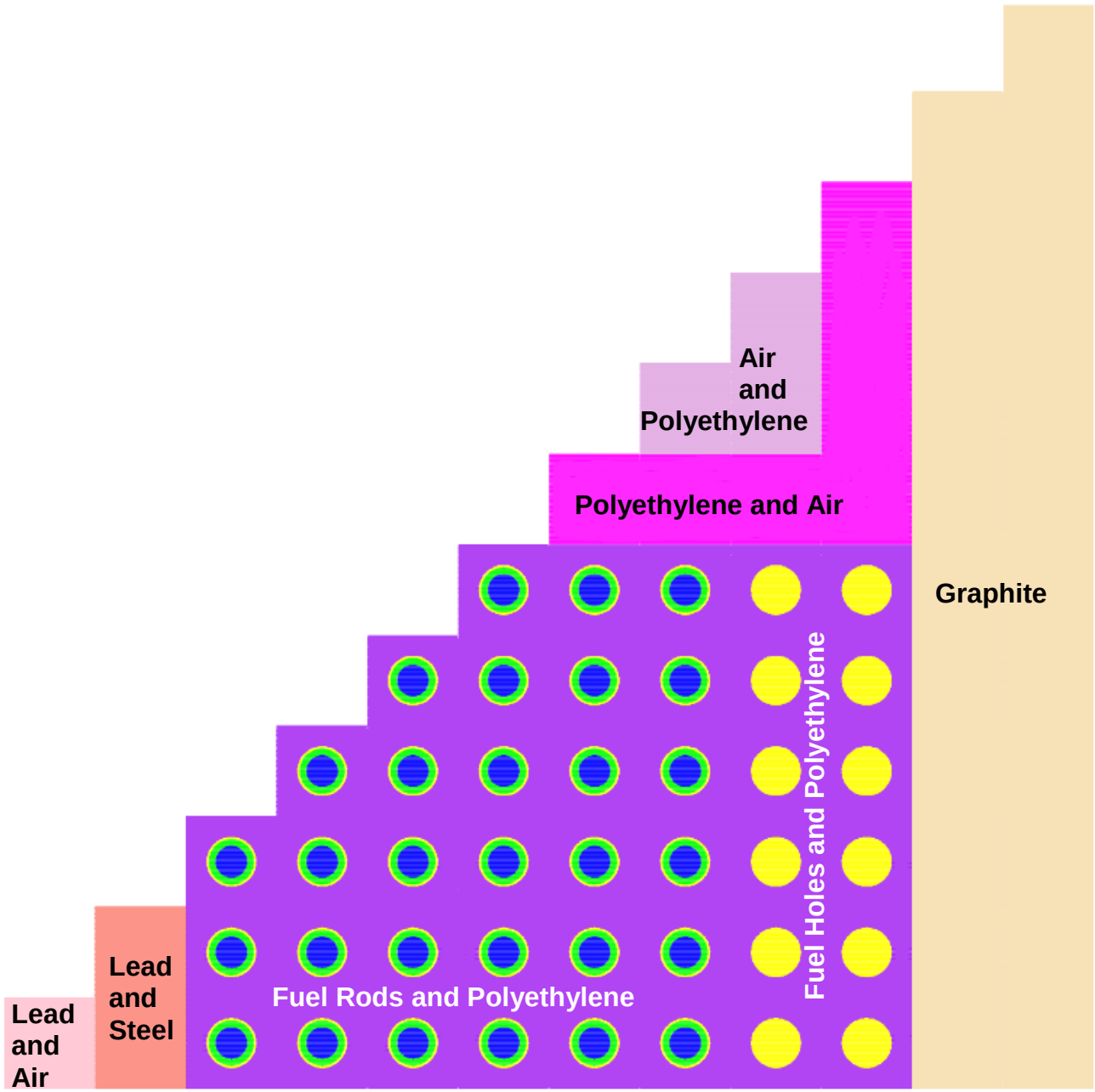

Figure A2. DRAGON geometry for one eight of the subcritical assembly

The third model represents a one quarter of the target zone surrounded by two concentric square rows of fuel pins using a three-dimensional geometry. This input has the total axial length of the facility and it was used for generating the homogenized cross sections of the target, the beam tube, the lead, the borated polyethylene, and the inactive fuel regions.

Figures A3 and A4 illustrate the horizontal and vertical sections, respectively, of the DRAGON model for the target region. The axial length of the borated polyethylene and the air zones at the top and the bottom has been halved in the DRAGON geometry to take advantage of the reflective boundary conditions. Reflected boundary conditions have been applied to the top and left boundary of the model in Figure A3 and to the top and bottom boundaries of the model in Figure A4. Symmetric boundary conditions have been applied to the right and bottom boundaries of the model in Figure A3.

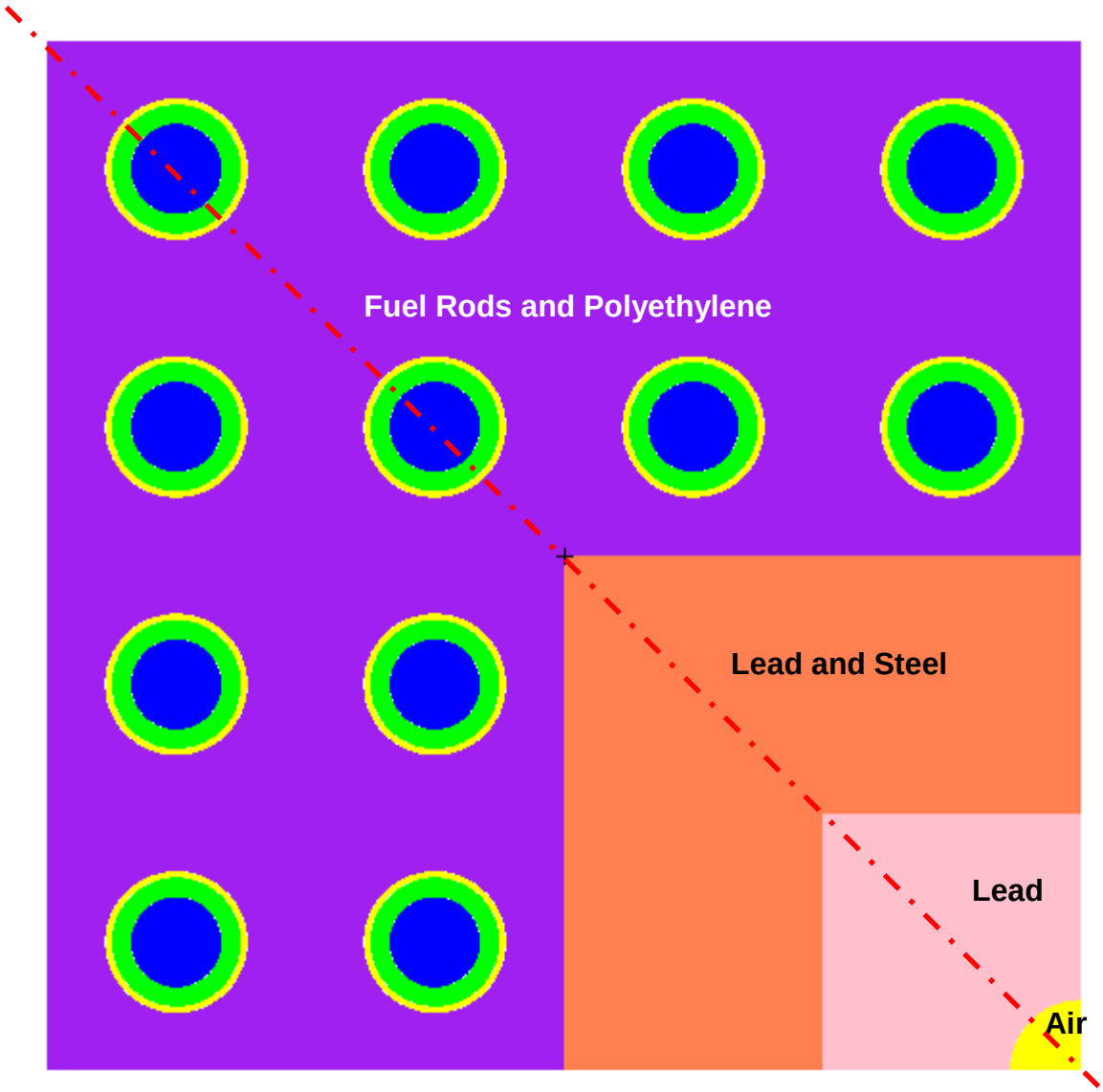

Figure A3. Horizontal section in the lead zone of the DRAGON geometry for the target region

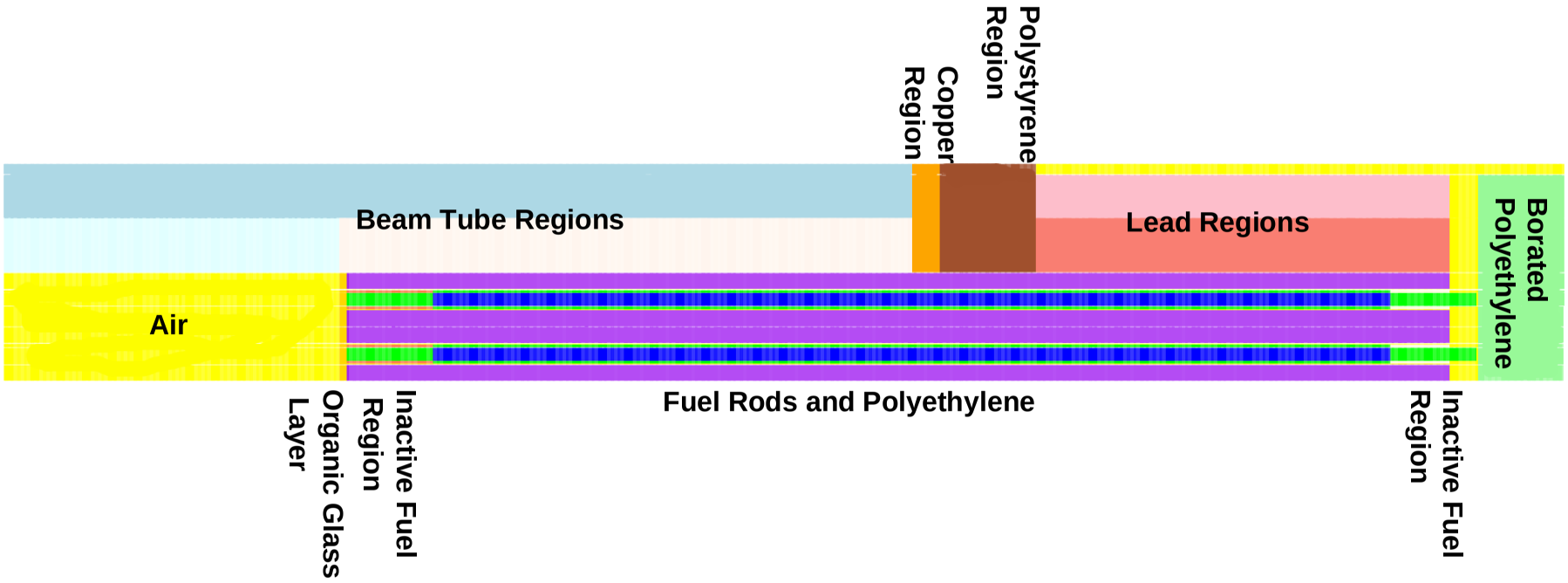

Figure A4. Vertical section along the plane indicated by the red line of Figure A3 for the target region

Appendix B: Alternative Definition of the Kinetics Parameters for Subcritical Systems

In this Appendix, the alternative definition of kinetics parameters for subcritical systems proposed by Dulla et al.[110] and applied by Lee and Maiorino.[111] is examined. In the new definition of the kinetic parameters, given in Equations $B1$-$B3$, neutrons produced from (n,2n) reactions and external source neutrons are counted as prompt neutrons. The original proposal of Dulla at al.[110] included only source neutrons as prompt neutrons. The definition of the denominator of the kinetic parameters (F factor) is arbitrary.[88] The symbols in Equations $B1$-$B3$ follow the notation used in Equations 6 and 15; $\Sigma_{n,2n}$ is the macroscopic cross section for the (n,2n) reaction. The alternative definition of the kinetic parameters represents different physical quantities relative to the classical definition. Consequently, the alternative kinetic parameters are referred to as *source* prompt neutron lifetime $\ell_{p,src}$ and *source* delayed neutron fraction β_{src}.

$$\ell_{p,src} \equiv \frac{k_{eff}}{F}\int_V d\vec{r}\int_0^{20MeV} dE\int_{4\pi} d\vec{\Omega}\varphi_s^+(\vec{r},\vec{\Omega},E)\frac{1}{\mathrm{v}(E)}\varphi_s(\vec{r},\vec{\Omega},E) \qquad (B1)$$

$$\beta_{src} \equiv \sum_{k=1}^{6}\frac{1}{F}\int_V d\vec{r}\int_0^{20MeV} dE\int_{4\pi} d\vec{\Omega}\int_0^{20MeV} dE'\int_{4\pi} d\vec{\Omega}'\varphi_s^+(\vec{r},\vec{\Omega}',E')\chi_{d,k}(\vec{r},E')\nu_{d,k}(\vec{r},E)\Sigma_f(\vec{r},E)\varphi_s(\vec{r},\vec{\Omega},E) \qquad (B2)$$

$$\begin{aligned} F \equiv & \int_V d\vec{r}\int_0^{20MeV} dE\int_{4\pi} d\vec{\Omega}\int_0^{20MeV} dE'\int_{4\pi} d\vec{\Omega}'\varphi_s^+(\vec{r},\vec{\Omega},E)\Big[\chi(E)\nu\Sigma_f(\vec{r},E') + \\ & + 2\Sigma_{n,2n}(\vec{r},\vec{\Omega}\cdot\vec{\Omega}',E'\to E)\Big]\varphi_s(\vec{r},\vec{\Omega}',E',t) + \int_V d\vec{r}\int_0^{20MeV} dE\int_{4\pi} d\vec{\Omega}\varphi_s^+(\vec{r},\vec{\Omega},E)s(\vec{r},\vec{\Omega},E) \end{aligned} \qquad (B3)$$

The obtained alternative kinetic parameters are smaller than the classical kinetic parameters, as can be shown by examining Tables A.I and A.II relative to Tables VI and VII. The alternative kinetic parameters increase as the number of fuel rods increases. Therefore, the alternative kinetic parameters depend on the subcriticality of the assembly.

For the classical formulation of the kinetic parameters, the difference between the three fuel loading configurations is driven by the neutron spectrum and this sets the highest value for the configuration with 216 fuel rods. This configuration has the softest neutron spectrum and that weights more the thermal component of the adjoint neutron flux. For the new formulation of the kinetic parameters, the difference between the three fuel loading configurations is driven by the subcriticality level and this sets the highest value for the configuration with 280 fuel rods.

Table B.I: Source delayed neutron fraction [pcm] as function of the fuel rods number calculated using PARTISN5.86, nuclear data library based on ENDF/B-VI.8 nuclear data files, and Equations B2 and B3

ADJOINT-WEIGHTED (EFFECTIVE) β	**CONFIGURATION**		
NEUTRON SOURCE	**216**	**245**	**280**
Cf Source	690	711	732
D-D Source	689	711	732

Table B.II: Source prompt neutron lifetime [μs] as function of the fuel rods number calculated using PARTISN5.86, nuclear data library based on ENDF/B-VI.8 nuclear data files, and Equations B1 and B3

ADJOINT-WEIGHTED (EFFECTIVE) ℓ_p	**CONFIGURATION**		
NEUTRON SOURCE	**216**	**245**	**280**
Cf Source Adjoint Equations 10 and 15	63.9	68.6	76.4
D-D Source Adjoint Equations 10 and 15	64.3	68.9	76.7

Appendix C: DRAGON2PARTISN Source Code

To code can be compiled on Linux systems by the command:

```
gcc -o dragon2partisn dragon2partisn.c
```

The following example generates the cross section of PARTISN and it stores them into the ASCII file xs.txt:

```
./dragon2partisn xspin/xs.txt            xs.txt      1  y y y
./dragon2partisn xslead/xs.txt           xs.txt      1  n n y
./dragon2partisn xslead/xs.txt           xs.txt      2  n n y
./dragon2partisn xslead/xs.txt           xs.txt      3  y n y
```

The xspin/xs.txt and xslead/xs.txt files are two different DRAGON 3.05E output files with ASCII macroscopic cross sections located in the directories xspin and xslead.

1, 1, 2 and 3 are the DRAGON materials to be processed which become materials 1 2 3 4 in PARTISN.

The first y/n flag indicates whether the material is fissile.

The second y/n flag indicates whether the code has to print the fission neutron spectrum.

The third y/n flag is reserved for future developments (always use y).

```
// Author: Alberto Talamo
// Email : alby@anl.gov
// Program written to generate PARTISN cross sections for Yalina Thermal

#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <math.h>

FILE         *fr,*fw;
char
buf[256],*head,*filer,*filew,found='n',mark='n',fissile='n',truefissile='n',spectrum='n',partisn='y',s1
='n';
int           i,j,k,ngrp,xmat,sumpos=0,sumpos1=0,*npos,*ipos,*npos1,*ipos1;
double       *tot,*tottr,*fis,*nfi,*chi,*cap,*sca,*sca1,*sinp,**sout,*sinp1,**sout1,holder;

main(int argc,char *argv[], int mat)
{
printf("\n");
if (argc<7)
   {
   printf("USAGE:    filter inputfile outputfile dragonmaterial y|n y|n y|n\n");
   printf("Please put input file, output file, material, fissile, spectrum-output partisn-output in the
execution line\n");
   printf("Use fissile           =y if material is fissile\n");
   printf("Use spectrum          =y for printing spectrum to separate file\n");
   printf("Use partisn           =y for printing data in partisn format\n");
   exit(0);
   }
filer=argv[1];
filew=argv[2];
mat=atoi(argv[3]);
fissile=argv[4][0];
spectrum=argv[5][0];
partisn=argv[6][0];
if ((fissile!='y')&&(fissile!='n'))
   {
   printf("Fissile value can only be y or n\n");
   exit(0);
   }
if ((spectrum!='y')&&(spectrum!='n'))
   {
   printf("Spectrum output value can only be y or n\n");
   exit(0);
   }
if ((partisn!='y')&&(partisn!='n'))
   {
   printf("Partisn value can only be y or n\n");
   exit(0);
   }
if ((fr=fopen(filer,"r"))==NULL)
   {
   printf("Input file not found\n");
   exit(0);
   }
while (!feof(fr))
   {
   fgets(buf,256,fr);
   if (strstr(buf,"COMPO")&&strstr(buf,"-1"))
      {
      head=strtok(buf," ");
      head=strtok(NULL," ");
      head=strtok(NULL," ");
      xmat=atoi(head);
      if (xmat==mat)
         {
         while(mark=='n')
            {
            fgets(buf,256,fr);
            if (strstr(buf,"CHI")&&(fissile=='n'))
               {
               printf("Material %d is fissile\n",mat);
               exit(0);
               }
```

```
            if (strstr(buf,"CHI")&&(fissile=='y'))
               {
               truefissile='y';
               }
            if ((strstr(buf,"COMPO"))||(feof(fr))) mark='y';
            }
         }
      }
   }
fclose(fr);
fr=fopen(filer,"r");
while (!feof(fr))
   {
   fgets(buf,256,fr);
   if (strstr(buf,"SCAT 1")) s1='y';
   }
fclose(fr);
if (s1=='n')
   {
   printf("Order 1 scattering data not found for material %d in the DRAGON input\n",mat);
   exit(0);
   }
mark='n';
fr=fopen(filer,"r");
while (!feof(fr))
   {
   fgets(buf,256,fr);
   if (strstr(buf,"COMPO")&&strstr(buf,"-1"))
      {
      head=strtok(buf," ");
      head=strtok(NULL," ");
      head=strtok(NULL," ");
      xmat=atoi(head);
      if (xmat==mat)
         {
         found='y';
         while(mark=='n')
            {
            fgets(buf,256,fr);
            if (strstr(buf,"TOTAL"))
               {
               head=strtok(buf," ");
               head=strtok(NULL," ");
               head=strtok(NULL," ");
               head=strtok(NULL," ");
               head=strtok(NULL," ");
               ngrp=atoi(head);
               tot  =malloc(ngrp*sizeof(double));
               tottr=malloc(ngrp*sizeof(double));
               sca  =malloc(ngrp*sizeof(double));
               sca1 =malloc(ngrp*sizeof(double));
               cap  =malloc(ngrp*sizeof(double));
               npos =malloc(ngrp*sizeof(int));
               ipos =malloc(ngrp*sizeof(int));
               npos1=malloc(ngrp*sizeof(int));
               ipos1=malloc(ngrp*sizeof(int));
               sout =malloc(ngrp*sizeof(double *));
               sout1=malloc(ngrp*sizeof(double *));
               fis  =malloc(ngrp*sizeof(double));
               nfi  =malloc(ngrp*sizeof(double));
               chi  =malloc(ngrp*sizeof(double));
               for (i=0;i<ngrp;i++)
                   {
                   tot  [i]=0;
                   tottr[i]=0;
                   sca  [i]=0;
                   sca1 [i]=0;
                   cap  [i]=0;
                   npos [i]=0;
                   ipos [i]=0;
                   npos1[i]=0;
                   ipos1[i]=0;
                   fis  [i]=0;
                   nfi  [i]=0;
```

```
                    chi  [i]=0;
                    sout [i]=malloc(ngrp*sizeof(double));
                    sout1[i]=malloc(ngrp*sizeof(double));
                    }
                for (i=0;i<ngrp;i++)
                    {
                    for (j=0;j<ngrp;j++)
                        {
                        sout [i][j]=0;
                        sout1[i][j]=0;
                        }
                    }
                k=0;
                for (i=0;i<=(ngrp/5);i++)
                    {
                    fgets(buf,256,fr);
                    head=strtok(buf," ");
                    for (j=0;j<5;j++)
                        {
                        if (k<ngrp)
                           {
                           tot[k]=atof(head);
                           k++;
                           head=strtok(NULL," ");
                           }
                        }
                    }
                mark='y';
                }
            }
        }
    }
}
fclose(fr);
mark='n';
fr=fopen(filer,"r");
while (!feof(fr))
   {
   fgets(buf,256,fr);
   if (strstr(buf,"COMPO")&&strstr(buf,"-1"))
      {
      head=strtok(buf," ");
      head=strtok(NULL," ");
      head=strtok(NULL," ");
      xmat=atoi(head);
      if (xmat==mat)
         {
         while(mark=='n')
            {
            fgets(buf,256,fr);
            if (strstr(buf,"STRD"))
               {
               k=0;
               for (i=0;i<=(ngrp/5);i++)
                   {
                   fgets(buf,256,fr);
                   head=strtok(buf," ");
                   for (j=0;j<5;j++)
                       {
                       if (k<ngrp)
                          {
                          tottr[k]=atof(head);
                          k++;
                          head=strtok(NULL," ");
                          }
                       }
                   }
               mark='y';
               }
            }
         }
      }
   }
fclose(fr);
```

```
mark='n';
fr=fopen(filer,"r");
while (!feof(fr))
   {
   fgets(buf,256,fr);
   if (strstr(buf,"COMPO")&&strstr(buf,"-1"))
      {
      head=strtok(buf," ");
      head=strtok(NULL," ");
      head=strtok(NULL," ");
      xmat=atoi(head);
      if (xmat==mat)
         {
         while(mark=='n')
            {
            fgets(buf,256,fr);
            if (strstr(buf,"SIGS 0"))
               {
               k=0;
               for (i=0;i<=(ngrp/5);i++)
                   {
                   fgets(buf,256,fr);
                   head=strtok(buf," ");
                   for (j=0;j<5;j++)
                       {
                       if (k<ngrp)
                          {
                          sca[k]=atof(head);
                          k++;
                          head=strtok(NULL," ");
                          }
                       }
                   }
               mark='y';
               }
            }
         }
      }
   }
fclose(fr);
mark='n';
fr=fopen(filer,"r");
while (!feof(fr))
   {
   fgets(buf,256,fr);
   if (strstr(buf,"COMPO")&&strstr(buf,"-1"))
      {
      head=strtok(buf," ");
      head=strtok(NULL," ");
      head=strtok(NULL," ");
      xmat=atoi(head);
      if (xmat==mat)
         {
         while(mark=='n')
            {
            fgets(buf,256,fr);
            if (strstr(buf,"SIGS 1"))
               {
               k=0;
               for (i=0;i<=(ngrp/5);i++)
                   {
                   fgets(buf,256,fr);
                   head=strtok(buf," ");
                   for (j=0;j<5;j++)
                       {
                       if (k<ngrp)
                          {
                          sca1[k]=atof(head);
                          k++;
                          head=strtok(NULL," ");
                          }
                       }
                   }
               mark='y';
```

```
                }
            }
        }
    }
}
fclose(fr);
mark='n';
fr=fopen(filer,"r");
while (!feof(fr))
   {
   fgets(buf,256,fr);
   if (strstr(buf,"COMPO")&&strstr(buf,"-1"))
      {
      head=strtok(buf," ");
      head=strtok(NULL," ");
      head=strtok(NULL," ");
      xmat=atoi(head);
      if (xmat==mat)
         {
         while(mark=='n')
            {
            fgets(buf,256,fr);
            if (strstr(buf,"NG"))
               {
               k=0;
               for (i=0;i<=(ngrp/5);i++)
                   {
                   fgets(buf,256,fr);
                   head=strtok(buf," ");
                   for (j=0;j<5;j++)
                       {
                       if (k<ngrp)
                          {
                          cap[k]=atof(head);
                          k++;
                          head=strtok(NULL," ");
                          }
                       }
                   }
               mark='y';
               }
            }
         }
      }
   }
fclose(fr);
if (truefissile=='y')
   {
   mark='n';
   fr=fopen(filer,"r");
   while (!feof(fr))
      {
      fgets(buf,256,fr);
      if (strstr(buf,"COMPO")&&strstr(buf,"-1"))
         {
         head=strtok(buf," ");
         head=strtok(NULL," ");
         head=strtok(NULL," ");
         xmat=atoi(head);
         if (xmat==mat)
            {
            while(mark=='n')
               {
               fgets(buf,256,fr);
               if (strstr(buf,"NFTOT"))
                  {
                  k=0;
                  for (i=0;i<=(ngrp/5);i++)
                      {
                      fgets(buf,256,fr);
                      head=strtok(buf," ");
                      for (j=0;j<5;j++)
                          {
                          if (k<ngrp)
```

```
                        {
                        fis[k]=atof(head);
                        k++;
                        head=strtok(NULL," ");
                        }
                    }
                }
            mark='y';
            }
        }
    }
  }
fclose(fr);
mark='n';
fr=fopen(filer,"r");
while (!feof(fr))
  {
  fgets(buf,256,fr);
  if (strstr(buf,"COMPO")&&strstr(buf,"-1"))
    {
    head=strtok(buf," ");
    head=strtok(NULL," ");
    head=strtok(NULL," ");
    xmat=atoi(head);
    if (xmat==mat)
      {
      while(mark=='n')
        {
        fgets(buf,256,fr);
        if (strstr(buf,"NUSIGF"))
          {
          k=0;
          for (i=0;i<=(ngrp/5);i++)
              {
              fgets(buf,256,fr);
              head=strtok(buf," ");
              for (j=0;j<5;j++)
                  {
                  if (k<ngrp)
                      {
                      nfi[k]=atof(head);
                      k++;
                      head=strtok(NULL," ");
                      }
                  }
              }
          mark='y';
          }
        }
      }
    }
  }
fclose(fr);
mark='n';
fr=fopen(filer,"r");
while (!feof(fr))
  {
  fgets(buf,256,fr);
  if (strstr(buf,"COMPO")&&strstr(buf,"-1"))
    {
    head=strtok(buf," ");
    head=strtok(NULL," ");
    head=strtok(NULL," ");
    xmat=atoi(head);
    if (xmat==mat)
      {
      while(mark=='n')
        {
        fgets(buf,256,fr);
        if (strstr(buf,"CHI"))
          {
          k=0;
          for (i=0;i<=(ngrp/5);i++)
```

```
                        {
                        fgets(buf,256,fr);
                        head=strtok(buf," ");
                        for (j=0;j<5;j++)
                            {
                            if (k<ngrp)
                               {
                               chi[k]=atof(head);
                               k++;
                               head=strtok(NULL," ");
                               }
                            }
                        }
                     mark='y';
                     }
                  }
               }
            }
         }
      fclose(fr);
      }
   sumpos=0;
   mark='n';
   fr=fopen(filer,"r");
   while (!feof(fr))
      {
      fgets(buf,256,fr);
      if (strstr(buf,"COMPO")&&strstr(buf,"-1"))
         {
         head=strtok(buf," ");
         head=strtok(NULL," ");
         head=strtok(NULL," ");
         xmat=atoi(head);
         if (xmat==mat)
            {
            while(mark=='n')
               {
               fgets(buf,256,fr);
               if (strstr(buf,"NJJ  0"))
                  {
                  k=0;
                  while (k<ngrp)
                      {
                      fgets(buf,256,fr);
                      head=strtok(buf," ");
                      for (j=0;j<8;j++)
                          {
                          if (k<ngrp)
                             {
                             npos[k]=atoi(head);
                             sumpos=sumpos+npos[k];
                             k++;
                             head=strtok(NULL," ");
                             }
                          }
                      }
                  mark='y';
                  }
               }
            }
         }
      }
   fclose(fr);
   mark='n';
   fr=fopen(filer,"r");
   while (!feof(fr))
      {
      fgets(buf,256,fr);
      if (strstr(buf,"COMPO")&&strstr(buf,"-1"))
         {
         head=strtok(buf," ");
         head=strtok(NULL," ");
         head=strtok(NULL," ");
         xmat=atoi(head);
```

```
        if (xmat==mat)
           {
           while(mark=='n')
              {
              fgets(buf,256,fr);
              if (strstr(buf,"IJJ  0"))
                 {
                 k=0;
                 while (k<ngrp)
                     {
                     fgets(buf,256,fr);
                     head=strtok(buf," ");
                     for (j=0;j<8;j++)
                         {
                         if (k<ngrp)
                            {
                            ipos[k]=atoi(head);
                            k++;
                            head=strtok(NULL," ");
                            }
                         }
                     }
                 mark='y';
                 }
              }
           }
        }
    }
fclose(fr);
if (s1=='y')
    {
    sumpos1=0;
    mark='n';
    fr=fopen(filer,"r");
    while (!feof(fr))
        {
        fgets(buf,256,fr);
        if (strstr(buf,"COMPO")&&strstr(buf,"-1"))
           {
           head=strtok(buf," ");
           head=strtok(NULL," ");
           head=strtok(NULL," ");
           xmat=atoi(head);
           if (xmat==mat)
              {
              while(mark=='n')
                 {
                 fgets(buf,256,fr);
                 if (strstr(buf,"NJJ  1"))
                    {
                    k=0;
                    while (k<ngrp)
                        {
                        fgets(buf,256,fr);
                        head=strtok(buf," ");
                        for (j=0;j<8;j++)
                            {
                            if (k<ngrp)
                               {
                               npos1[k]=atoi(head);
                               sumpos1=sumpos1+npos1[k];
                               k++;
                               head=strtok(NULL," ");
                               }
                            }
                        }
                    mark='y';
                    }
                 }
              }
           }
        }
    fclose(fr);
    mark='n';
```

```
   fr=fopen(filer,"r");
   while (!feof(fr))
      {
      fgets(buf,256,fr);
      if (strstr(buf,"COMPO")&&strstr(buf,"-1"))
         {
         head=strtok(buf," ");
         head=strtok(NULL," ");
         head=strtok(NULL," ");
         xmat=atoi(head);
         if (xmat==mat)
            {
            while(mark=='n')
               {
               fgets(buf,256,fr);
               if (strstr(buf,"IJJ  1"))
                  {
                  k=0;
                  while (k<ngrp)
                      {
                      fgets(buf,256,fr);
                      head=strtok(buf," ");
                      for (j=0;j<8;j++)
                          {
                          if (k<ngrp)
                             {
                             ipos1[k]=atoi(head);
                             k++;
                             head=strtok(NULL," ");
                             }
                          }
                      }
                  mark='y';
                  }
               }
            }
         }
      }
   fclose(fr);
   }
sinp =malloc(sumpos *sizeof(double));
sinp1=malloc(sumpos1*sizeof(double));
for (i=0;i<sumpos ;i++) sinp [i]=0;
for (i=0;i<sumpos1;i++) sinp1[i]=0;
mark='n';
fr=fopen(filer,"r");
while (!feof(fr))
   {
   fgets(buf,256,fr);
   if (strstr(buf,"COMPO")&&strstr(buf,"-1"))
      {
      head=strtok(buf," ");
      head=strtok(NULL," ");
      head=strtok(NULL," ");
      xmat=atoi(head);
      if (xmat==mat)
         {
         while(mark=='n')
            {
            fgets(buf,256,fr);
            if (strstr(buf,"SCAT 0"))
               {
               k=0;
               while (k<sumpos)
                   {
                   fgets(buf,256,fr);
                   head=strtok(buf," ");
                   for (j=0;j<5;j++)
                       {
                       if (k<sumpos)
                          {
                          sinp[k]=atof(head);
                          k++;
                          head=strtok(NULL," ");
```

```
                        }
                    }
                }
            mark='y';
            }
        }
    }
  }
}
fclose(fr);
mark='n';
fr=fopen(filer,"r");
while (!feof(fr))
  {
  fgets(buf,256,fr);
  if (strstr(buf,"COMPO")&&strstr(buf,"-1"))
    {
    head=strtok(buf," ");
    head=strtok(NULL," ");
    head=strtok(NULL," ");
    xmat=atoi(head);
    if (xmat==mat)
      {
      while(mark=='n')
        {
        fgets(buf,256,fr);
        if (strstr(buf,"SCAT 1"))
          {
          k=0;
          while (k<sumpos1)
              {
              fgets(buf,256,fr);
              head=strtok(buf," ");
              for (j=0;j<5;j++)
                  {
                  if (k<sumpos1)
                      {
                      sinp1[k]=atof(head);
                      k++;
                      head=strtok(NULL," ");
                      }
                  }
              }
          mark='y';
          }
        }
      }
    }
  }
fclose(fr);
if (found=='n')
  {
  printf("Material %d not found\n",mat);
  exit(0);
  }
if ((fissile=='y')&&(truefissile=='n')&&(found=='y'))
  {
  printf("Material %d is not fissile\n",mat);
  exit(0);
  }
if (found=='y')
  {
  printf("The total cross section for material %d is:\n",mat);
  for (i=0;i<ngrp;i++)
      {
      printf("tot[%2d]=%12.8e  ",i+1,tot[i]);
      if ((i%5)==4) printf ("\n");
      }
  printf ("\n\n");
  printf("The transport corrected cross section for material %d is:\n",mat);
  for (i=0;i<ngrp;i++)
      {
      printf("tot[%2d]=%12.8e  ",i+1,tottr[i]);
      if ((i%5)==4) printf ("\n");
```

```
    }
printf ("\n\n");
if (truefissile=='y')
   {
   printf("The fission cross section for material %d is:\n",mat);
   for (i=0;i<ngrp;i++)
       {
       printf("fis[%2d]=%12.8e  ",i+1,fis[i]);
       if ((i%5)==4) printf ("\n");
       }
   printf ("\n\n");
   printf("The nu*fission cross section for material %d is:\n",mat);
   for (i=0;i<ngrp;i++)
       {
       printf("nfi[%2d]=%12.8e  ",i+1,nfi[i]);
       if ((i%5)==4) printf ("\n");
       }
   printf ("\n\n");
   printf("The fission spectrum for material %d is:\n",mat);
   for (i=0;i<ngrp;i++)
       {
       printf("chi[%2d]=%12.8e  ",i+1,chi[i]);
       if ((i%5)==4) printf ("\n");
       }
   printf ("\n\n");
   }
printf("The capture cross section for material %d is:\n",mat);
for (i=0;i<ngrp;i++)
    {
    printf("cap[%2d]=%12.8e  ",i+1,cap[i]);
    if ((i%5)==4) printf ("\n");
    }
printf ("\n\n");
printf("The total 0 scattering cross section for material %d is:\n",mat);
for (i=0;i<ngrp;i++)
    {
    printf("sca[%2d]=%+12.8e  ",i+1,sca[i]);
    if ((i%5)==4) printf ("\n");
    }
printf ("\n\n");
printf("The number of positions in the 0 scattering matrix for material %d is:\n",mat);
for (i=0;i<ngrp;i++)
    {
    printf("npos0[%2d]=%2d  ",i+1,npos[i]);
    if ((i%5)==4) printf ("\n");
    }
printf ("\n\n");
printf("The first  of position  in the 0 scattering matrix for material %d is:\n",mat);
for (i=0;i<ngrp;i++)
    {
    printf("ipos0[%2d]=%2d  ",i+1,ipos[i]);
    if ((i%5)==4) printf ("\n");
    }
printf ("\n\n");
printf("The sum of the position of the 0 scattering matrix for material %d is %d\n\n",mat,sumpos);
printf("The input 0 scattering matrix for material %d is:\n",mat);
for (i=0;i<sumpos;i++)
    {
    printf("sca0[%4d]=%+12.8f  ",i+1,sinp[i]);
    if ((i%5)==4) printf ("\n");
    }
printf ("\n\n");
printf("The total 1 scattering cross section for material %d is:\n",mat);
for (i=0;i<ngrp;i++)
    {
    printf("sca0[%2d]=%+12.8e  ",i+1,sca1[i]);
    if ((i%5)==4) printf ("\n");
    }
printf ("\n\n");
printf("The number of positions in the 1 scattering matrix for material %d is:\n",mat);
for (i=0;i<ngrp;i++)
    {
    printf("npos1[%2d]=%2d  ",i+1,npos1[i]);
    if ((i%5)==4) printf ("\n");
```

```
    }
printf ("\n\n");
printf("The first  of position  in the 1 scattering matrix for material %d is:\n",mat);
for (i=0;i<ngrp;i++)
    {
    printf("ipos1[%2d]=%2d  ",i+1,ipos1[i]);
    if ((i%5)==4) printf ("\n");
    }
printf ("\n\n");
printf("The sum of the position of the 1 scattering matrix for material %d is %d\n\n",mat,sumpos1);
printf("The input 1 scattering matrix for material %d is:\n",mat);
for (i=0;i<sumpos1;i++)
    {
       printf("sca1[%4d]=%+12.8e  ",i+1,sinp1[i]);
       if ((i%5)==4) printf ("\n");
    }
printf ("\n\nBuilding the full scattering matrices\n\n");
k=0;
for (i=0;i<ngrp;i++)
    {
    k=k+npos[i];
    for (j=0;j<npos[i];j++)
        {
        sout[j+ipos[i]-npos[i]][i]=sinp[k-j-1];
        }
    }
k=0;
for (i=0;i<ngrp;i++)
    {
    k=k+npos1[i];
    for (j=0;j<npos1[i];j++)
        {
        sout1[j+ipos1[i]-npos1[i]][i]=sinp1[k-j-1];
        }
    }
printf("Checking 0 scattering matrix\n");
for (i=0;i<ngrp;i++)
    {
    holder=0;
    for (j=0;j<ngrp;j++)
        {
        holder=holder+sout[i][j];
        }
    if (abs(holder-sca[i])>1e-6)
       {
       printf ("Check on 0 scattering matrix not passed for group %d\n",i+1);
       exit(0);
       }
    }
printf("Checking 1 scattering matrix\n");
for (i=0;i<ngrp;i++)
    {
    holder=0;
    for (j=0;j<ngrp;j++)
        {
        holder=holder+sout1[i][j];
        }
    if (abs(holder-sca1[i])>1e-6)
       {
       printf ("Check on 1 scattering matrix not passed for group %d\n",i+1);
       exit(0);
       }
    }
printf("\n\nThe full 0 scattering matrix is:\n");
for (i=0;i<ngrp;i++)
    {
    for (j=0;j<ngrp;j++)
        {
        printf("sca0[%4d][%4d]=%12.8e ",i+1,j+1,sout[i][j]);
        if ((j%5)==4) printf ("\n");
        }
    printf("\n");
    }
printf("\n\nThe full 1 scattering matrix is:\n");
```

```
for (i=0;i<ngrp;i++)
    {
    for (j=0;j<ngrp;j++)
        {
        printf("sca1[%4d][%4d]=%12.8e ",i+1,j+1,sout1[i][j]);
        if ((j%5)==4) printf ("\n");
        }
    printf("\n");
    }
if ((fw=fopen(filew,"a+"))==NULL)
   {
   printf("Cannot write file %s\n",filew);
   exit(0);
   }
if (partisn=='y')
   {
   printf("\n\nWriting PARTISN cross sections\n\n");
   printf("The PARTISN number of cross section data is %d\n",2*ngrp-1+4);
   printf("The PARTISN position of the total cross section is 4\n");
   printf("The PARTISN position of the self-scattering cross section is %d\n",ngrp+4);
   printf("\n\nCreating adjusted p1 scattering matrix\n\n");
   for (i=0;i<ngrp;i++)
       {
       for (j=0;j<ngrp;j++)
           {
           if (j==i)
              {
              sout[i][j]=sout[i][j];
              }
           }
       }
   fprintf(fw,"DRAGON material %d from file %s - FILTER to file %s\n",mat,filer,filew);
   for (i=0;i<ngrp;i++)
       {
       fprintf(fw,"1. %12.8e %12.8e %12.8e\n",cap[i]+fis[i],nfi[i],tot[i]);
       k=0;j=ngrp-1+i;
       while (k<(2*ngrp-1))
           {
           if ((j>=0)&&(j<=(ngrp-1)))
              {
              fprintf(fw,"%12.8e ",sout[j][i]);
              }
           else
              {
              fprintf(fw,"%12.8e ",0.0);
              }
           if (k==(2*ngrp-2))
              {
              fprintf(fw,"%s","/");
              }
           if ((k%5)==4)
              {
              fprintf(fw,"\n");
              }
           k++;
           j--;
           }
       if (!(((k-1)%5)==4)) fprintf(fw,"\n");
       }
   fprintf(fw,"%s\n","T");
   // this piece of code is left for further developments
   // and it has given a worse match with MCNP results
   fprintf(fw,"ORDER 1 SCATTERING\n");
   for (i=0;i<ngrp;i++)
       {
       fprintf(fw,"1. %12.8e %12.8e %12.8e\n",cap[i]+fis[i],nfi[i],tottr[i]);
       k=0;j=ngrp-1+i;
       while (k<(2*ngrp-1))
           {
           if ((j>=0)&&(j<=(ngrp-1)))
              {
              fprintf(fw,"%12.8e ",sout1[j][i]);
              }
           else
```

```
                {
                fprintf(fw,"%12.8e ",0.0);
                }
            if (k==(2*ngrp-2))
                {
                fprintf(fw,"%s","/");
                }
            if ((k%5)==4)
                {
                fprintf(fw,"\n");
                }
            k++;
            j--;
            }
        if (!(((k-1)%5)==4)) fprintf(fw,"\n");
        }
    fprintf(fw,"%s\n","T");
    fclose(fw);
    printf("\nPARTISN cross section successfully written for material %d\n\n",mat);
    if (spectrum=='y')
        {
        if ((fw=fopen("spectrum.txt","a+"))==NULL)
            {
            printf("Cannot write file %s\n",filew);
            exit(0);
            }
        fprintf(fw,"The spectrum for DRAGON material %d is:\n",mat);
        for (i=0;i<ngrp;i++)
            {
            fprintf(fw,"%12.8e ",chi[i]);
            if ((i%5)==4)
                {
                fprintf(fw,"\n");
                }
            }
        fclose(fw);
        printf("\nFission spectrum successfully written for material %d\n\n",mat);
        }
    }
if (partisn=='n')
    {
    printf("\n\nCreating adjusted p1 scattering matrix\n\n");
    for (i=0;i<ngrp;i++)
        {
        for (j=0;j<ngrp;j++)
            {
            if (j==i)
                {
                sout[i][j]=sout[i][j];
                }
            }
        }
    printf("DRAGON material %d from file %s - FILTER to file %s\n",mat,filer,filew);
    fprintf(fw,"chit"); for (i=0;i<ngrp;i++) fprintf(fw,"  %12.8e",chi[i]); fprintf(fw,"\n");
    fprintf(fw,"chid"); for (i=0;i<ngrp;i++) fprintf(fw,"  %12.8e",chi[i]); fprintf(fw,"\n");
    fprintf(fw,"xsto"); for (i=0;i<ngrp;i++) fprintf(fw,"  %12.8e",tot[i]); fprintf(fw,"\n");
    fprintf(fw,"xsnf"); for (i=0;i<ngrp;i++) fprintf(fw,"  %12.8e",nfi[i]); fprintf(fw,"\n");
    fprintf(fw,"xsfi"); for (i=0;i<ngrp;i++) fprintf(fw,"  %12.8e",fis[i]); fprintf(fw,"\n");
    for (j=0;j<ngrp;j++)
        {
        fprintf(fw,"xssc"); for (i=0;i<ngrp;i++) fprintf(fw,"  %12.8e",sout[j][i]); fprintf(fw,"\n");
        }
    }
}
}
```

Appendix D: DRAGON2TORT Source Code

To code can be compiled on Linux systems by the command:

```
gcc -o dragon2tort dragon2tort.c
```

The following example generates the cross section of TORT and it stores them into the ASCII file xs.txt:

```
./dragon2tort xspin/xs.txt            xs.txt      1  y y y
./dragon2tort xslead/xs.txt           xs.txt      1  n n y
./dragon2tort xslead/xs.txt           xs.txt      2  n n y
./dragon2tort xslead/xs.txt           xs.txt      3  y n y
```

The xspin/xs.txt and xslead/xs.txt files are two different DRAGON 3.05E output files with ASCII macroscopic cross sections located in the directories xspin and xslead.

1, 1, 2 and 3 are the DRAGON materials to be processed which become materials 1 2 3 4 in TORT.

The first y/n flag indicates whether the material is fissile.

The second y/n flag indicates whether the code has to print the fission neutron spectrum.

The third y/n flag is reserved for future developments (always use y)

```
// Author: Alberto Talamo
// Email : alby@anl.gov
// Program written to generate TORT cross sections for Yalina Thermal

#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <math.h>

FILE         *fr,*fw;
char
buf[256],*head,*filer,*filew,found='n',mark='n',fissile='n',truefissile='n',spectrum='n',tort='y',s1='n
';
int           i,j,k,ngrp,xmat,sumpos=0,sumpos1=0,*npos,*ipos,*npos1,*ipos1;
double       *tot,*tottr,*fis,*nfi,*chi,*cap,*sca,*sca1,*sinp,**sout,*sinp1,**sout1,holder;

main(int argc,char *argv[], int mat)
{
printf("\n\n");
printf("            COPYRIGHT 2008, ALBERTO TALAMO\n\n");
printf("            alby@anl.gov\n\n\n\n");
printf("            Released 28 May 2008\n\n\n\n");
printf("            DRAGON2TORT v 1.0 is free software: you can redistribute it and/or modify\n");
printf("            it under the terms of the GNU General Public License as published by\n");
printf("            the Free Software Foundation version 3 of the License.\n\n");
printf("            DRAGON2TORT is distributed in the hope that it will be useful,\n");
printf("            but WITHOUT ANY WARRANTY; without even the implied warranty of\n");
printf("            MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the\n");
printf("            GNU General Public License for more details.\n\n");
printf("            You should have received a copy of the GNU General Public License\n");
printf("            along with DRAGON2TORT.  If not, see <http://www.gnu.org/licenses/>.\n\n");
printf("            Argonne National Laboratory makes no warranty\n");
printf("            express or implied, and assumes no liability or\n");
printf("            responsibility for the use of DRAGON2TORT.\n\n\n\n");
if (argc<7)
   {
   printf("USAGE:    filter inputfile outputfile dragonmaterial y|n y|n y|n\n");
   printf("Please put input file, output file, material, fissile, spectrum-output tort-output in the
execution line\n");
   printf("Use fissile           =y if material is fissile\n");
   printf("Use spectrum          =y for printing spectrum to separate file\n");
   printf("Use tort           =y for printing data in tort format\n");
   exit(0);
   }
filer=argv[1];
filew=argv[2];
mat=atoi(argv[3]);
fissile=argv[4][0];
spectrum=argv[5][0];
tort=argv[6][0];
if ((fissile!='y')&&(fissile!='n'))
   {
   printf("Fissile value can only be y or n\n");
   exit(0);
   }
if ((spectrum!='y')&&(spectrum!='n'))
   {
   printf("Spectrum output value can only be y or n\n");
   exit(0);
   }
if ((tort!='y')&&(tort!='n'))
   {
   printf("Tort value can only be y or n\n");
   exit(0);
   }
if ((fr=fopen(filer,"r"))==NULL)
   {
   printf("Input file not found\n");
   exit(0);
   }
while (!feof(fr))
   {
   fgets(buf,256,fr);
   if (strstr(buf,"COMPO")&&strstr(buf,"-1"))
```

```
      {
      head=strtok(buf," ");
      head=strtok(NULL," ");
      head=strtok(NULL," ");
      xmat=atoi(head);
      if (xmat==mat)
         {
         while(mark=='n')
            {
            fgets(buf,256,fr);
            if (strstr(buf,"CHI")&&(fissile=='n'))
               {
               printf("Material %d is fissile\n",mat);
               exit(0);
               }
            if (strstr(buf,"CHI")&&(fissile=='y'))
               {
               truefissile='y';
               }
            if ((strstr(buf,"COMPO"))||(feof(fr))) mark='y';
            }
         }
      }
   }
fclose(fr);
fr=fopen(filer,"r");
while (!feof(fr))
   {
   fgets(buf,256,fr);
   if (strstr(buf,"SCAT 1")) s1='y';
   }
fclose(fr);
if (s1=='n')
   {
   printf("Order 1 scattering data not found for material %d in the DRAGON input\n",mat);
   exit(0);
   }
mark='n';
fr=fopen(filer,"r");
while (!feof(fr))
   {
   fgets(buf,256,fr);
   if (strstr(buf,"COMPO")&&strstr(buf,"-1"))
      {
      head=strtok(buf," ");
      head=strtok(NULL," ");
      head=strtok(NULL," ");
      xmat=atoi(head);
      if (xmat==mat)
         {
         found='y';
         while(mark=='n')
            {
            fgets(buf,256,fr);
            if (strstr(buf,"TOTAL"))
               {
               head=strtok(buf," ");
               head=strtok(NULL," ");
               head=strtok(NULL," ");
               head=strtok(NULL," ");
               head=strtok(NULL," ");
               ngrp=atoi(head);
               tot  =malloc(ngrp*sizeof(double));
               tottr=malloc(ngrp*sizeof(double));
               sca  =malloc(ngrp*sizeof(double));
               sca1 =malloc(ngrp*sizeof(double));
               cap  =malloc(ngrp*sizeof(double));
               npos =malloc(ngrp*sizeof(int));
               ipos =malloc(ngrp*sizeof(int));
               npos1=malloc(ngrp*sizeof(int));
               ipos1=malloc(ngrp*sizeof(int));
               sout =malloc(ngrp*sizeof(double *));
               sout1=malloc(ngrp*sizeof(double *));
               fis  =malloc(ngrp*sizeof(double));
```

```
                nfi  =malloc(ngrp*sizeof(double));
                chi  =malloc(ngrp*sizeof(double));
                for (i=0;i<ngrp;i++)
                    {
                    tot  [i]=0;
                    tottr[i]=0;
                    sca  [i]=0;
                    sca1 [i]=0;
                    cap  [i]=0;
                    npos [i]=0;
                    ipos [i]=0;
                    npos1[i]=0;
                    ipos1[i]=0;
                    fis  [i]=0;
                    nfi  [i]=0;
                    chi  [i]=0;
                    sout [i]=malloc(ngrp*sizeof(double));
                    sout1[i]=malloc(ngrp*sizeof(double));
                    }
                for (i=0;i<ngrp;i++)
                    {
                    for (j=0;j<ngrp;j++)
                        {
                        sout [i][j]=0;
                        sout1[i][j]=0;
                        }
                    }
                k=0;
                for (i=0;i<=(ngrp/5);i++)
                    {
                    fgets(buf,256,fr);
                    head=strtok(buf," ");
                    for (j=0;j<5;j++)
                        {
                        if (k<ngrp)
                           {
                           tot[k]=atof(head);
                           k++;
                           head=strtok(NULL," ");
                           }
                        }
                    }
                mark='y';
                }
            }
        }
    }
}
fclose(fr);
mark='n';
fr=fopen(filer,"r");
while (!feof(fr))
   {
   fgets(buf,256,fr);
   if (strstr(buf,"COMPO")&&strstr(buf,"-1"))
      {
      head=strtok(buf," ");
      head=strtok(NULL," ");
      head=strtok(NULL," ");
      xmat=atoi(head);
      if (xmat==mat)
         {
         while(mark=='n')
            {
            fgets(buf,256,fr);
            if (strstr(buf,"STRD"))
               {
               k=0;
               for (i=0;i<=(ngrp/5);i++)
                   {
                   fgets(buf,256,fr);
                   head=strtok(buf," ");
                   for (j=0;j<5;j++)
                       {
```

```
                        if (k<ngrp)
                           {
                           tottr[k]=atof(head);
                           k++;
                           head=strtok(NULL," ");
                           }
                        }
                     }
                  mark='y';
                  }
               }
            }
         }
      }
   }
fclose(fr);
mark='n';
fr=fopen(filer,"r");
while (!feof(fr))
   {
   fgets(buf,256,fr);
   if (strstr(buf,"COMPO")&&strstr(buf,"-1"))
      {
      head=strtok(buf," ");
      head=strtok(NULL," ");
      head=strtok(NULL," ");
      xmat=atoi(head);
      if (xmat==mat)
         {
         while(mark=='n')
            {
            fgets(buf,256,fr);
            if (strstr(buf,"SIGS 0"))
               {
               k=0;
               for (i=0;i<=(ngrp/5);i++)
                   {
                   fgets(buf,256,fr);
                   head=strtok(buf," ");
                   for (j=0;j<5;j++)
                       {
                       if (k<ngrp)
                          {
                          sca[k]=atof(head);
                          k++;
                          head=strtok(NULL," ");
                          }
                       }
                   }
               mark='y';
               }
            }
         }
      }
   }
fclose(fr);
mark='n';
fr=fopen(filer,"r");
while (!feof(fr))
   {
   fgets(buf,256,fr);
   if (strstr(buf,"COMPO")&&strstr(buf,"-1"))
      {
      head=strtok(buf," ");
      head=strtok(NULL," ");
      head=strtok(NULL," ");
      xmat=atoi(head);
      if (xmat==mat)
         {
         while(mark=='n')
            {
            fgets(buf,256,fr);
            if (strstr(buf,"SIGS 1"))
               {
               k=0;
```

```
                for (i=0;i<=(ngrp/5);i++)
                    {
                    fgets(buf,256,fr);
                    head=strtok(buf," ");
                    for (j=0;j<5;j++)
                        {
                        if (k<ngrp)
                           {
                           sca1[k]=atof(head);
                           k++;
                           head=strtok(NULL," ");
                           }
                        }
                    }
                mark='y';
                }
            }
        }
    }
}
fclose(fr);
mark='n';
fr=fopen(filer,"r");
while (!feof(fr))
   {
   fgets(buf,256,fr);
   if (strstr(buf,"COMPO")&&strstr(buf,"-1"))
      {
      head=strtok(buf," ");
      head=strtok(NULL," ");
      head=strtok(NULL," ");
      xmat=atoi(head);
      if (xmat==mat)
         {
         while(mark=='n')
            {
            fgets(buf,256,fr);
            if (strstr(buf,"NG"))
               {
               k=0;
               for (i=0;i<=(ngrp/5);i++)
                   {
                   fgets(buf,256,fr);
                   head=strtok(buf," ");
                   for (j=0;j<5;j++)
                       {
                       if (k<ngrp)
                          {
                          cap[k]=atof(head);
                          k++;
                          head=strtok(NULL," ");
                          }
                       }
                   }
               mark='y';
               }
            }
         }
      }
   }
fclose(fr);
if (truefissile=='y')
   {
   mark='n';
   fr=fopen(filer,"r");
   while (!feof(fr))
      {
      fgets(buf,256,fr);
      if (strstr(buf,"COMPO")&&strstr(buf,"-1"))
         {
         head=strtok(buf," ");
         head=strtok(NULL," ");
         head=strtok(NULL," ");
         xmat=atoi(head);
```

```
      if (xmat==mat)
         {
         while(mark=='n')
            {
            fgets(buf,256,fr);
            if (strstr(buf,"NFTOT"))
               {
               k=0;
               for (i=0;i<=(ngrp/5);i++)
                   {
                   fgets(buf,256,fr);
                   head=strtok(buf," ");
                   for (j=0;j<5;j++)
                       {
                       if (k<ngrp)
                          {
                          fis[k]=atof(head);
                          k++;
                          head=strtok(NULL," ");
                          }
                       }
                   }
               mark='y';
               }
            }
         }
      }
   }
fclose(fr);
mark='n';
fr=fopen(filer,"r");
while (!feof(fr))
   {
   fgets(buf,256,fr);
   if (strstr(buf,"COMPO")&&strstr(buf,"-1"))
      {
      head=strtok(buf," ");
      head=strtok(NULL," ");
      head=strtok(NULL," ");
      xmat=atoi(head);
      if (xmat==mat)
         {
         while(mark=='n')
            {
            fgets(buf,256,fr);
            if (strstr(buf,"NUSIGF"))
               {
               k=0;
               for (i=0;i<=(ngrp/5);i++)
                   {
                   fgets(buf,256,fr);
                   head=strtok(buf," ");
                   for (j=0;j<5;j++)
                       {
                       if (k<ngrp)
                          {
                          nfi[k]=atof(head);
                          k++;
                          head=strtok(NULL," ");
                          }
                       }
                   }
               mark='y';
               }
            }
         }
      }
   }
fclose(fr);
mark='n';
fr=fopen(filer,"r");
while (!feof(fr))
   {
   fgets(buf,256,fr);
```

```
      if (strstr(buf,"COMPO")&&strstr(buf,"-1"))
         {
         head=strtok(buf," ");
         head=strtok(NULL," ");
         head=strtok(NULL," ");
         xmat=atoi(head);
         if (xmat==mat)
            {
            while(mark=='n')
               {
               fgets(buf,256,fr);
               if (strstr(buf,"CHI"))
                  {
                  k=0;
                  for (i=0;i<=(ngrp/5);i++)
                      {
                      fgets(buf,256,fr);
                      head=strtok(buf," ");
                      for (j=0;j<5;j++)
                          {
                          if (k<ngrp)
                             {
                             chi[k]=atof(head);
                             k++;
                             head=strtok(NULL," ");
                             }
                          }
                      }
                  mark='y';
                  }
               }
            }
         }
      }
   fclose(fr);
   }
sumpos=0;
mark='n';
fr=fopen(filer,"r");
while (!feof(fr))
   {
   fgets(buf,256,fr);
   if (strstr(buf,"COMPO")&&strstr(buf,"-1"))
      {
      head=strtok(buf," ");
      head=strtok(NULL," ");
      head=strtok(NULL," ");
      xmat=atoi(head);
      if (xmat==mat)
         {
         while(mark=='n')
            {
            fgets(buf,256,fr);
            if (strstr(buf,"NJJ  0"))
               {
               k=0;
               while (k<ngrp)
                   {
                   fgets(buf,256,fr);
                   head=strtok(buf," ");
                   for (j=0;j<8;j++)
                       {
                       if (k<ngrp)
                          {
                          npos[k]=atoi(head);
                          sumpos=sumpos+npos[k];
                          k++;
                          head=strtok(NULL," ");
                          }
                       }
                   }
               mark='y';
               }
            }
```

```
            }
         }
      }
   fclose(fr);
   mark='n';
   fr=fopen(filer,"r");
   while (!feof(fr))
      {
      fgets(buf,256,fr);
      if (strstr(buf,"COMPO")&&strstr(buf,"-1"))
         {
         head=strtok(buf," ");
         head=strtok(NULL," ");
         head=strtok(NULL," ");
         xmat=atoi(head);
         if (xmat==mat)
            {
            while(mark=='n')
               {
               fgets(buf,256,fr);
               if (strstr(buf,"IJJ  0"))
                  {
                  k=0;
                  while (k<ngrp)
                      {
                      fgets(buf,256,fr);
                      head=strtok(buf," ");
                      for (j=0;j<8;j++)
                          {
                          if (k<ngrp)
                             {
                             ipos[k]=atoi(head);
                             k++;
                             head=strtok(NULL," ");
                             }
                          }
                      }
                  mark='y';
                  }
               }
            }
         }
      }
   fclose(fr);
   if (s1=='y')
      {
      sumpos1=0;
      mark='n';
      fr=fopen(filer,"r");
      while (!feof(fr))
         {
         fgets(buf,256,fr);
         if (strstr(buf,"COMPO")&&strstr(buf,"-1"))
            {
            head=strtok(buf," ");
            head=strtok(NULL," ");
            head=strtok(NULL," ");
            xmat=atoi(head);
            if (xmat==mat)
               {
               while(mark=='n')
                  {
                  fgets(buf,256,fr);
                  if (strstr(buf,"NJJ  1"))
                     {
                     k=0;
                     while (k<ngrp)
                         {
                         fgets(buf,256,fr);
                         head=strtok(buf," ");
                         for (j=0;j<8;j++)
                             {
                             if (k<ngrp)
                                {
```

```
                                npos1[k]=atoi(head);
                                sumpos1=sumpos1+npos1[k];
                                k++;
                                head=strtok(NULL," ");
                                }
                            }
                        }
                    mark='y';
                    }
                }
            }
        }
    }
    fclose(fr);
    mark='n';
    fr=fopen(filer,"r");
    while (!feof(fr))
        {
        fgets(buf,256,fr);
        if (strstr(buf,"COMPO")&&strstr(buf,"-1"))
            {
            head=strtok(buf," ");
            head=strtok(NULL," ");
            head=strtok(NULL," ");
            xmat=atoi(head);
            if (xmat==mat)
                {
                while(mark=='n')
                    {
                    fgets(buf,256,fr);
                    if (strstr(buf,"IJJ  1"))
                        {
                        k=0;
                        while (k<ngrp)
                            {
                            fgets(buf,256,fr);
                            head=strtok(buf," ");
                            for (j=0;j<8;j++)
                                {
                                if (k<ngrp)
                                    {
                                    ipos1[k]=atoi(head);
                                    k++;
                                    head=strtok(NULL," ");
                                    }
                                }
                            }
                        mark='y';
                        }
                    }
                }
            }
        }
    fclose(fr);
    }
sinp =malloc(sumpos *sizeof(double));
sinp1=malloc(sumpos1*sizeof(double));
for (i=0;i<sumpos ;i++) sinp [i]=0;
for (i=0;i<sumpos1;i++) sinp1[i]=0;
mark='n';
fr=fopen(filer,"r");
while (!feof(fr))
    {
    fgets(buf,256,fr);
    if (strstr(buf,"COMPO")&&strstr(buf,"-1"))
        {
        head=strtok(buf," ");
        head=strtok(NULL," ");
        head=strtok(NULL," ");
        xmat=atoi(head);
        if (xmat==mat)
            {
            while(mark=='n')
                {
```

```
            fgets(buf,256,fr);
            if (strstr(buf,"SCAT 0"))
               {
               k=0;
               while (k<sumpos)
                   {
                   fgets(buf,256,fr);
                   head=strtok(buf," ");
                   for (j=0;j<5;j++)
                       {
                       if (k<sumpos)
                          {
                          sinp[k]=atof(head);
                          k++;
                          head=strtok(NULL," ");
                          }
                       }
                   }
               mark='y';
               }
            }
         }
      }
   }
fclose(fr);
mark='n';
fr=fopen(filer,"r");
while (!feof(fr))
   {
   fgets(buf,256,fr);
   if (strstr(buf,"COMPO")&&strstr(buf,"-1"))
      {
      head=strtok(buf," ");
      head=strtok(NULL," ");
      head=strtok(NULL," ");
      xmat=atoi(head);
      if (xmat==mat)
         {
         while(mark=='n')
            {
            fgets(buf,256,fr);
            if (strstr(buf,"SCAT 1"))
               {
               k=0;
               while (k<sumpos1)
                   {
                   fgets(buf,256,fr);
                   head=strtok(buf," ");
                   for (j=0;j<5;j++)
                       {
                       if (k<sumpos1)
                          {
                          sinp1[k]=atof(head);
                          k++;
                          head=strtok(NULL," ");
                          }
                       }
                   }
               mark='y';
               }
            }
         }
      }
   }
fclose(fr);
if (found=='n')
   {
   printf("Material %d not found\n",mat);
   exit(0);
   }
if ((fissile=='y')&&(truefissile=='n')&&(found=='y'))
   {
   printf("Material %d is not fissile\n",mat);
   exit(0);
```

```
    }
if (found=='y')
    {
    printf("The total cross section for material %d is:\n",mat);
    for (i=0;i<ngrp;i++)
        {
        printf("tot[%2d]=%12.8e  ",i+1,tot[i]);
        if ((i%5)==4) printf ("\n");
        }
    printf ("\n\n");
    printf("The transport corrected cross section for material %d is:\n",mat);
    for (i=0;i<ngrp;i++)
        {
        printf("tot[%2d]=%12.8e  ",i+1,tottr[i]);
        if ((i%5)==4) printf ("\n");
        }
    printf ("\n\n");
    if (truefissile=='y')
       {
       printf("The fission cross section for material %d is:\n",mat);
       for (i=0;i<ngrp;i++)
           {
           printf("fis[%2d]=%12.8e  ",i+1,fis[i]);
           if ((i%5)==4) printf ("\n");
           }
       printf ("\n\n");
       printf("The nu*fission cross section for material %d is:\n",mat);
       for (i=0;i<ngrp;i++)
           {
           printf("nfi[%2d]=%12.8e  ",i+1,nfi[i]);
           if ((i%5)==4) printf ("\n");
           }
       printf ("\n\n");
       printf("The fission spectrum for material %d is:\n",mat);
       for (i=0;i<ngrp;i++)
           {
           printf("chi[%2d]=%12.8e  ",i+1,chi[i]);
           if ((i%5)==4) printf ("\n");
           }
       printf ("\n\n");
       }
    printf("The capture cross section for material %d is:\n",mat);
    for (i=0;i<ngrp;i++)
        {
        printf("cap[%2d]=%12.8e  ",i+1,cap[i]);
        if ((i%5)==4) printf ("\n");
        }
    printf ("\n\n");
    printf("The total 0 scattering cross section for material %d is:\n",mat);
    for (i=0;i<ngrp;i++)
        {
        printf("sca[%2d]=%+12.8e  ",i+1,sca[i]);
        if ((i%5)==4) printf ("\n");
        }
    printf ("\n\n");
    printf("The number of positions in the 0 scattering matrix for material %d is:\n",mat);
    for (i=0;i<ngrp;i++)
        {
        printf("npos0[%2d]=%2d  ",i+1,npos[i]);
        if ((i%5)==4) printf ("\n");
        }
    printf ("\n\n");
    printf("The first  of position  in the 0 scattering matrix for material %d is:\n",mat);
    for (i=0;i<ngrp;i++)
        {
        printf("ipos0[%2d]=%2d  ",i+1,ipos[i]);
        if ((i%5)==4) printf ("\n");
        }
    printf ("\n\n");
    printf("The sum of the position of the 0 scattering matrix for material %d is %d\n\n",mat,sumpos);
    printf("The input 0 scattering matrix for material %d is:\n",mat);
    for (i=0;i<sumpos;i++)
        {
        printf("sca0[%4d]=%+12.8f  ",i+1,sinp[i]);
```

```
    if ((i%5)==4) printf ("\n");
    }
printf ("\n\n");
printf("The total 1 scattering cross section for material %d is:\n",mat);
for (i=0;i<ngrp;i++)
    {
    printf("sca0[%2d]=%+12.8e  ",i+1,sca1[i]);
    if ((i%5)==4) printf ("\n");
    }
printf ("\n\n");
printf("The number of positions in the 1 scattering matrix for material %d is:\n",mat);
for (i=0;i<ngrp;i++)
    {
    printf("npos1[%2d]=%2d  ",i+1,npos1[i]);
    if ((i%5)==4) printf ("\n");
    }
printf ("\n\n");
printf("The first  of position  in the 1 scattering matrix for material %d is:\n",mat);
for (i=0;i<ngrp;i++)
    {
    printf("ipos1[%2d]=%2d  ",i+1,ipos1[i]);
    if ((i%5)==4) printf ("\n");
    }
printf ("\n\n");
printf("The sum of the position of the 1 scattering matrix for material %d is %d\n\n",mat,sumpos1);
printf("The input 1 scattering matrix for material %d is:\n",mat);
for (i=0;i<sumpos1;i++)
    {
       printf("sca1[%4d]=%+12.8e  ",i+1,sinp1[i]);
       if ((i%5)==4) printf ("\n");
    }
printf ("\n\nBuilding the full scattering matrices\n\n");
k=0;
for (i=0;i<ngrp;i++)
    {
    k=k+npos[i];
    for (j=0;j<npos[i];j++)
        {
        sout[j+ipos[i]-npos[i]][i]=sinp[k-j-1];
        }
    }
k=0;
for (i=0;i<ngrp;i++)
    {
    k=k+npos1[i];
    for (j=0;j<npos1[i];j++)
        {
        sout1[j+ipos1[i]-npos1[i]][i]=sinp1[k-j-1];
        }
    }
printf("Checking 0 scattering matrix\n");
for (i=0;i<ngrp;i++)
    {
    holder=0;
    for (j=0;j<ngrp;j++)
        {
        holder=holder+sout[i][j];
        }
    if (abs(holder-sca[i])>1e-6)
       {
       printf ("Check on 0 scattering matrix not passed for group %d\n",i+1);
       exit(0);
       }
    }
printf("Checking 1 scattering matrix\n");
for (i=0;i<ngrp;i++)
    {
    holder=0;
    for (j=0;j<ngrp;j++)
        {
        holder=holder+sout1[i][j];
        }
    if (abs(holder-sca1[i])>1e-6)
       {
```

```
        printf ("Check on 1 scattering matrix not passed for group %d\n",i+1);
        exit(0);
        }
    }
printf("\n\nThe full 0 scattering matrix is:\n");
for (i=0;i<ngrp;i++)
    {
    for (j=0;j<ngrp;j++)
        {
        printf("sca0[%4d][%4d]=%12.8e ",i+1,j+1,sout[i][j]);
        if ((j%5)==4) printf ("\n");
        }
    printf("\n");
    }
printf("\n\nThe full 1 scattering matrix is:\n");
for (i=0;i<ngrp;i++)
    {
    for (j=0;j<ngrp;j++)
        {
        printf("sca1[%4d][%4d]=%12.8e ",i+1,j+1,sout1[i][j]);
        if ((j%5)==4) printf ("\n");
        }
    printf("\n");
    }
if ((fw=fopen(filew,"a+"))==NULL)
   {
   printf("Cannot write file %s\n",filew);
   exit(0);
   }
if (tort=='y')
   {
   printf("\n\nWriting TORT cross sections\n\n");
   printf("The TORT number of cross section data is %d\n",2*ngrp-1+4);
   printf("The TORT position of the total cross section is 4\n");
   printf("The TORT position of the self-scattering cross section is %d\n",ngrp+4);
   printf("\n\nCreating adjusted p1 scattering matrix\n\n");
   for (i=0;i<ngrp;i++)
       {
       for (j=0;j<ngrp;j++)
           {
           if (j==i)
              {
              sout[i][j]=sout[i][j];
              }
           }
       }
   for (i=0;i<ngrp;i++)
       {
       fprintf(fw,"1. %+11.4e %+11.4e %+11.4e\n",cap[i]+fis[i],nfi[i],tot[i]);
       k=0;j=ngrp-1+i;
       while (k<(2*ngrp-1))
           {
           if ((j>=0)&&(j<=(ngrp-1)))
              {
              fprintf(fw,"%+11.4e ",sout[j][i]);
              }
           else
              {
              fprintf(fw,"%+11.4e ",0.0);
              }
           if ((k%5)==4)
              {
              fprintf(fw,"\n");
              }
           k++;
           j--;
           }
       if (!(((k-1)%5)==4)) fprintf(fw,"\n");
       }
   // this piece of code is left for further developments
   // and it has given a worse match with MCNP results
   for (i=0;i<ngrp;i++)
       {
       fprintf(fw,"1. %+11.4e %+11.4e %+11.4e\n",cap[i]+fis[i],nfi[i],tot[i]);
```

```
        k=0;j=ngrp-1+i;
        while (k<(2*ngrp-1))
            {
            if ((j>=0)&&(j<=(ngrp-1)))
               {
               fprintf(fw,"%+11.4e ",sout1[j][i]*3);
//             fprintf(fw,"%+11.4e ",sout1[j][i]);
               }
            else
               {
               fprintf(fw,"%+11.4e ",0.0);
               }
            if ((k%5)==4)
               {
               fprintf(fw,"\n");
               }
            k++;
            j--;
            }
        if (!(((k-1)%5)==4)) fprintf(fw,"\n");
        }
    fclose(fw);
    printf("\nTORT cross section successfully written for material %d\n\n",mat);
    if (spectrum=='y')
       {
       if ((fw=fopen("spectrum.txt","a+"))==NULL)
          {
          printf("Cannot write file %s\n",filew);
          exit(0);
          }
       fprintf(fw,"The spectrum for DRAGON material %d is:\n",mat);
       for (i=0;i<ngrp;i++)
           {
           fprintf(fw,"%12.8e ",chi[i]);
           if ((i%5)==4)
              {
              fprintf(fw,"\n");
              }
           }
       fclose(fw);
       printf("\nFission spectrum successfully written for material %d\n\n",mat);
       }
    }
 if (tort=='n')
    {
    printf("\n\nCreating adjusted p1 scattering matrix\n\n");
    for (i=0;i<ngrp;i++)
        {
        for (j=0;j<ngrp;j++)
            {
            if (j==i)
               {
               sout[i][j]=sout[i][j];
               }
            }
        }
    printf("DRAGON material %d from file %s - FILTER to file %s\n",mat,filer,filew);
    fprintf(fw,"chit"); for (i=0;i<ngrp;i++) fprintf(fw,"  %12.8e",chi[i]); fprintf(fw,"\n");
    fprintf(fw,"chid"); for (i=0;i<ngrp;i++) fprintf(fw,"  %12.8e",chi[i]); fprintf(fw,"\n");
    fprintf(fw,"xsto"); for (i=0;i<ngrp;i++) fprintf(fw,"  %12.8e",tot[i]); fprintf(fw,"\n");
    fprintf(fw,"xsnf"); for (i=0;i<ngrp;i++) fprintf(fw,"  %12.8e",nfi[i]); fprintf(fw,"\n");
    fprintf(fw,"xsfi"); for (i=0;i<ngrp;i++) fprintf(fw,"  %12.8e",fis[i]); fprintf(fw,"\n");
    for (j=0;j<ngrp;j++)
        {
        fprintf(fw,"xssc"); for (i=0;i<ngrp;i++) fprintf(fw,"  %12.8e",sout[j][i]); fprintf(fw,"\n");
        }
    }
 }
}
```

Appendix E: Source Code for the Calculation of the Effective Delayed Neutron Fraction

To code can be compiled on Linux systems by the command:

```
gcc –o betaadj betaadj.c
```

The code can be executed by the command:

```
./betaadj
```

The code expects two PARTISN ETDOUT files in the current directory:

1) `edtouta` containing the adjoint flux
2) `edtoutr` containing the fission rate integrated over volume

The code works for 69 energy groups calculations.

```
// Author: Alberto Talamo
// Email : alby@anl.gov
// Program written to calculate Yalina Thermal effective delayed neutron fraction

#define LINES 12
#define G     69
#define ODDG  3
#define SKIP  36

// LINES include both full lines and half-full lines
// ODDG  includes ONLY group entries NOT the total (last value)

// include standard and math C libraries
#include <stdio.h>
#include <math.h>
#include <stdlib.h>
#include <errno.h>
#include <string.h>

double calculate_chiig (unsigned long int,unsigned long int,double);
double calculate_betag(double);

double             e[70],v[69],chi[69];
double             eb[79],b[79],ebd[6],bd[6],wbdi[6];
double
eb1[186],b1[186],eb2[221],b2[221],eb3[302],b3[302],eb4[302],b4[302],eb5[302],b5[302],eb6[302],b6[302];

int main(int argc,char *argv[])
{
FILE              *fr=NULL,*fa=NULL;
char               buff[256],bufr[256],bufa[256];
unsigned long int  i,g;
double            *datar,*dataa,*datah,xr[6],xa[6],num=0,den=0,h=0,he=0,chiig=0,betag=0;
double             nor[6],hnum,hden;

e[ 0]=19.64E+6;
e[ 1]=6.065307000000000E+6;
e[ 2]=3.678794000000000E+6;
e[ 3]=2.231302000000000E+6;
e[ 4]=1.353353000000000E+6;
e[ 5]=8.208500000000000E+5;
e[ 6]=4.978707000000000E+5;
e[ 7]=3.019738000000000E+5;
e[ 8]=1.831564000000000E+5;
e[ 9]=1.110900000000000E+5;
e[10]=6.737947000000000E+4;
e[11]=4.086771000000000E+4;
e[12]=2.478752000000000E+4;
e[13]=1.503349000000000E+4;
e[14]=9.118820000000000E+3;
e[15]=5.530844000000000E+3;
e[16]=3.526622000000000E+3;
e[17]=2.248673000000000E+3;
e[18]=1.433817000000000E+3;
e[19]=9.142423000000000E+2;
e[20]=3.717032000000000E+2;
e[21]=1.486254000000000E+2;
e[22]=7.567357000000000E+1;
e[23]=4.825160000000000E+1;
e[24]=2.760773000000000E+1;
e[25]=1.592827000000000E+1;
e[26]=9.905554000000000E+0;
e[27]=4.000000000000000E+0;
e[28]=3.300000000000000E+0;
e[29]=2.600000000000000E+0;
e[30]=2.100000000000000E+0;
e[31]=1.500000000000000E+0;
e[32]=1.300000000000000E+0;
e[33]=1.150000000000000E+0;
e[34]=1.123000000000000E+0;
e[35]=1.097000000000000E+0;
e[36]=1.071000000000000E+0;
e[37]=1.045000000000000E+0;
e[38]=1.020000000000000E+0;
```

```
e[39]=9.960000000000001E-1;
e[40]=9.720000000000000E-1;
e[41]=9.500000000000000E-1;
e[42]=9.099999999999999E-1;
e[43]=8.500000000000000E-1;
e[44]=7.800000000000000E-1;
e[45]=6.250000000000000E-1;
e[46]=5.000000000000000E-1;
e[47]=4.000000000000000E-1;
e[48]=3.500000000000000E-1;
e[49]=3.200000000000000E-1;
e[50]=3.000000000000000E-1;
e[51]=2.800000000000000E-1;
e[52]=2.480000000000000E-1;
e[53]=2.200000000000000E-1;
e[54]=1.800000000000000E-1;
e[55]=1.400000000000000E-1;
e[56]=9.999999999999999E-2;
e[57]=8.000000000000000E-2;
e[58]=6.700000000000000E-2;
e[59]=5.800000000000000E-2;
e[60]=5.000000000000000E-2;
e[61]=4.200000000000000E-2;
e[62]=3.500000000000000E-2;
e[63]=3.000000000000000E-2;
e[64]=2.500000000000000E-2;
e[65]=2.000000000000000E-2;
e[66]=1.500000000000000E-2;
e[67]=1.000000000000000E-2;
e[68]=5.000000000000000E-3;
e[69]=0.0;

chi[ 0]=4.23614420e-02;
chi[ 1]=9.99016762e-02;
chi[ 2]=2.12527424e-01;
chi[ 3]=2.29415342e-01;
chi[ 4]=1.75834775e-01;
chi[ 5]=1.28027558e-01;
chi[ 6]=5.56633994e-02;
chi[ 7]=2.43730601e-02;
chi[ 8]=1.64825842e-02;
chi[ 9]=9.93314944e-03;
chi[10]=1.95214420e-03;
chi[11]=1.83465076e-03;
chi[12]=1.18020945e-03;
chi[13]=1.32842004e-04;
chi[14]=1.97383750e-04;
chi[15]=8.78311112e-05;
chi[16]=4.53476314e-05;
chi[17]=2.84571652e-05;
chi[18]=7.22563937e-06;
chi[19]=9.71575082e-06;
chi[20]=2.73465866e-06;
chi[21]=7.09480332e-07;
chi[22]=1.40179225e-07;
chi[23]=1.24003719e-07;
chi[24]=5.07342008e-08;
chi[25]=9.46713996e-09;
chi[26]=1.43810430e-08;
for (g=27;g<G;g++)
    {
    chi[g]=0;
    }

eb[ 0]=1.000000e-5;
eb[ 1]=2.530000e-2;
eb[ 2]=5.000000e-2;
eb[ 3]=1.000000e+1;
eb[ 4]=1.000000e+2;
eb[ 5]=1.000000e+3;
eb[ 6]=5.500000e+3;
eb[ 7]=7.750000e+3;
eb[ 8]=1.000000e+4;
eb[ 9]=1.500000e+4;
```

```
eb[10]=2.000000e+4;
eb[11]=3.000000e+4;
eb[12]=4.000000e+4;
eb[13]=5.000000e+4;
eb[14]=6.000000e+4;
eb[15]=7.000000e+4;
eb[16]=8.000000e+4;
eb[17]=9.000000e+4;
eb[18]=1.000000e+5;
eb[19]=1.200000e+5;
eb[20]=1.300000e+5;
eb[21]=1.400000e+5;
eb[22]=1.500000e+5;
eb[23]=1.700000e+5;
eb[24]=2.000000e+5;
eb[25]=2.500000e+5;
eb[26]=3.000000e+5;
eb[27]=3.500000e+5;
eb[28]=4.000000e+5;
eb[29]=5.000000e+5;
eb[30]=6.000000e+5;
eb[31]=7.000000e+5;
eb[32]=8.000000e+5;
eb[33]=9.000000e+5;
eb[34]=1.000000e+6;
eb[35]=1.200000e+6;
eb[36]=1.400000e+6;
eb[37]=1.600000e+6;
eb[38]=1.800000e+6;
eb[39]=2.000000e+6;
eb[40]=2.200000e+6;
eb[41]=2.400000e+6;
eb[42]=2.600000e+6;
eb[43]=2.800000e+6;
eb[44]=3.000000e+6;
eb[45]=3.500000e+6;
eb[46]=4.000000e+6;
eb[47]=4.500000e+6;
eb[48]=5.000000e+6;
eb[49]=5.500000e+6;
eb[50]=6.000000e+6;
eb[51]=6.500000e+6;
eb[52]=7.000000e+6;
eb[53]=7.500000e+6;
eb[54]=8.000000e+6;
eb[55]=8.500000e+6;
eb[56]=9.000000e+6;
eb[57]=9.500000e+6;
eb[58]=1.000000e+7;
eb[59]=1.050000e+7;
eb[60]=1.100000e+7;
eb[61]=1.150000e+7;
eb[62]=1.200000e+7;
eb[63]=1.250000e+7;
eb[64]=1.300000e+7;
eb[65]=1.350000e+7;
eb[66]=1.400000e+7;
eb[67]=1.450000e+7;
eb[68]=1.500000e+7;
eb[69]=1.550000e+7;
eb[70]=1.600000e+7;
eb[71]=1.650000e+7;
eb[72]=1.700000e+7;
eb[73]=1.750000e+7;
eb[74]=1.800000e+7;
eb[75]=1.850000e+7;
eb[76]=1.900000e+7;
eb[77]=1.950000e+7;
eb[78]=2.000000e+7;

b[ 0]=2.436700e+0;
b[ 1]=2.436700e+0;
b[ 2]=2.436700e+0;
b[ 3]=2.436700e+0;
```

```
b[ 4]=2.433800e+0;
b[ 5]=2.433800e+0;
b[ 6]=2.433800e+0;
b[ 7]=2.433800e+0;
b[ 8]=2.433800e+0;
b[ 9]=2.430568e+0;
b[10]=2.428822e+0;
b[11]=2.425701e+0;
b[12]=2.423662e+0;
b[13]=2.423474e+0;
b[14]=2.424763e+0;
b[15]=2.427546e+0;
b[16]=2.430875e+0;
b[17]=2.434348e+0;
b[18]=2.437763e+0;
b[19]=2.445148e+0;
b[20]=2.448498e+0;
b[21]=2.451609e+0;
b[22]=2.454480e+0;
b[23]=2.461376e+0;
b[24]=2.468672e+0;
b[25]=2.471926e+0;
b[26]=2.474470e+0;
b[27]=2.476584e+0;
b[28]=2.478467e+0;
b[29]=2.483665e+0;
b[30]=2.488804e+0;
b[31]=2.498085e+0;
b[32]=2.508477e+0;
b[33]=2.519675e+0;
b[34]=2.532706e+0;
b[35]=2.556938e+0;
b[36]=2.582102e+0;
b[37]=2.606924e+0;
b[38]=2.630685e+0;
b[39]=2.653354e+0;
b[40]=2.675612e+0;
b[41]=2.697988e+0;
b[42]=2.720509e+0;
b[43]=2.743971e+0;
b[44]=2.768200e+0;
b[45]=2.827824e+0;
b[46]=2.892750e+0;
b[47]=2.967097e+0;
b[48]=3.045345e+0;
b[49]=3.132001e+0;
b[50]=3.222551e+0;
b[51]=3.317200e+0;
b[52]=3.405401e+0;
b[53]=3.476412e+0;
b[54]=3.544635e+0;
b[55]=3.617747e+0;
b[56]=3.689833e+0;
b[57]=3.761163e+0;
b[58]=3.830917e+0;
b[59]=3.899283e+0;
b[60]=3.966945e+0;
b[61]=4.034130e+0;
b[62]=4.101022e+0;
b[63]=4.169664e+0;
b[64]=4.241148e+0;
b[65]=4.314359e+0;
b[66]=4.388344e+0;
b[67]=4.462294e+0;
b[68]=4.537460e+0;
b[69]=4.614076e+0;
b[70]=4.690103e+0;
b[71]=4.763749e+0;
b[72]=4.833436e+0;
b[73]=4.899934e+0;
b[74]=4.964851e+0;
b[75]=5.028246e+0;
b[76]=5.090174e+0;
b[77]=5.150692e+0;
```

```
b[78]=5.209845e+0;

ebd[0]=1.000000e-5;
ebd[1]=2.530000e-2;
ebd[2]=5.000000e+4;
ebd[3]=4.000000e+6;
ebd[4]=7.000000e+6;
ebd[5]=2.000000e+7;

bd[0]=1.585000e-2;
bd[1]=1.585000e-2;
bd[2]=1.670000e-2;
bd[3]=1.670000e-2;
bd[4]=9.000000e-3;
bd[5]=9.000000e-3;

wbdi[0]=3.197266e-2;
wbdi[1]=1.663712e-1;
wbdi[2]=1.613104e-1;
wbdi[3]=4.596467e-1;
wbdi[4]=1.334993e-1;
wbdi[5]=4.719978e-2;

eb1[  0]= 0.0000000e+000;
 b1[  0]= 2.3642900e-007;
eb1[  1]= 1.0000000e+004;
 b1[  1]= 2.1184730e-006;
eb1[  2]= 2.0000000e+004;
 b1[  2]= 2.3697070e-006;
eb1[  3]= 3.0000000e+004;
 b1[  3]= 8.2064170e-007;
eb1[  4]= 4.0000000e+004;
 b1[  4]= 1.5500980e-006;
eb1[  5]= 5.0000000e+004;
 b1[  5]= 2.3761880e-006;
eb1[  6]= 6.0000000e+004;
 b1[  6]= 1.6040290e-006;
eb1[  7]= 7.0000000e+004;
 b1[  7]= 2.0367980e-006;
eb1[  8]= 8.0000000e+004;
 b1[  8]= 1.6775230e-006;
eb1[  9]= 9.0000000e+004;
 b1[  9]= 9.5116420e-007;
eb1[ 10]= 1.0000000e+005;
 b1[ 10]= 7.6421610e-007;
eb1[ 11]= 1.1000000e+005;
 b1[ 11]= 1.1896910e-006;
eb1[ 12]= 1.2000000e+005;
 b1[ 12]= 1.7619240e-006;
eb1[ 13]= 1.3000000e+005;
 b1[ 13]= 1.9944970e-006;
eb1[ 14]= 1.4000000e+005;
 b1[ 14]= 2.0761490e-006;
eb1[ 15]= 1.5000000e+005;
 b1[ 15]= 1.6408930e-006;
eb1[ 16]= 1.6000000e+005;
 b1[ 16]= 1.3809110e-006;
eb1[ 17]= 1.7000000e+005;
 b1[ 17]= 1.8061250e-006;
eb1[ 18]= 1.8000000e+005;
 b1[ 18]= 2.0465980e-006;
eb1[ 19]= 1.9000000e+005;
 b1[ 19]= 1.4208980e-006;
eb1[ 20]= 2.0000000e+005;
 b1[ 20]= 1.3025290e-006;
eb1[ 21]= 2.1000000e+005;
 b1[ 21]= 1.3546730e-006;
eb1[ 22]= 2.2000000e+005;
 b1[ 22]= 1.2816820e-006;
eb1[ 23]= 2.3000000e+005;
 b1[ 23]= 1.4886790e-006;
eb1[ 24]= 2.4000000e+005;
 b1[ 24]= 2.4435270e-006;
eb1[ 25]= 2.5000000e+005;
```

```
 b1[ 25]= 2.9509680e-006;
eb1[ 26]= 2.6000000e+005;
 b1[ 26]= 2.2517920e-006;
eb1[ 27]= 2.7000000e+005;
 b1[ 27]= 1.6379560e-006;
eb1[ 28]= 2.8000000e+005;
 b1[ 28]= 1.1205940e-006;
eb1[ 29]= 2.9000000e+005;
 b1[ 29]= 8.1822770e-007;
eb1[ 30]= 3.0000000e+005;
 b1[ 30]= 8.5142850e-007;
eb1[ 31]= 3.1000000e+005;
 b1[ 31]= 1.0501910e-006;
eb1[ 32]= 3.2000000e+005;
 b1[ 32]= 1.0127600e-006;
eb1[ 33]= 3.3000000e+005;
 b1[ 33]= 8.3308890e-007;
eb1[ 34]= 3.4000000e+005;
 b1[ 34]= 8.2168590e-007;
eb1[ 35]= 3.5000000e+005;
 b1[ 35]= 8.0332600e-007;
eb1[ 36]= 3.6000000e+005;
 b1[ 36]= 1.2835390e-006;
eb1[ 37]= 3.7000000e+005;
 b1[ 37]= 1.7473810e-006;
eb1[ 38]= 3.8000000e+005;
 b1[ 38]= 1.4961920e-006;
eb1[ 39]= 3.9000000e+005;
 b1[ 39]= 1.2380730e-006;
eb1[ 40]= 4.0000000e+005;
 b1[ 40]= 1.1055540e-006;
eb1[ 41]= 4.1000000e+005;
 b1[ 41]= 1.0728630e-006;
eb1[ 42]= 4.2000000e+005;
 b1[ 42]= 9.7547050e-007;
eb1[ 43]= 4.3000000e+005;
 b1[ 43]= 1.0139280e-006;
eb1[ 44]= 4.4000000e+005;
 b1[ 44]= 1.0810930e-006;
eb1[ 45]= 4.5000000e+005;
 b1[ 45]= 8.5467670e-007;
eb1[ 46]= 4.6000000e+005;
 b1[ 46]= 9.9310960e-007;
eb1[ 47]= 4.7000000e+005;
 b1[ 47]= 1.1514630e-006;
eb1[ 48]= 4.8000000e+005;
 b1[ 48]= 1.0704800e-006;
eb1[ 49]= 4.9000000e+005;
 b1[ 49]= 1.0939650e-006;
eb1[ 50]= 5.0000000e+005;
 b1[ 50]= 1.1254100e-006;
eb1[ 51]= 5.1000000e+005;
 b1[ 51]= 9.1002160e-007;
eb1[ 52]= 5.2000000e+005;
 b1[ 52]= 7.3212090e-007;
eb1[ 53]= 5.3000000e+005;
 b1[ 53]= 5.7679500e-007;
eb1[ 54]= 5.4000000e+005;
 b1[ 54]= 5.5076940e-007;
eb1[ 55]= 5.5000000e+005;
 b1[ 55]= 7.2191920e-007;
eb1[ 56]= 5.6000000e+005;
 b1[ 56]= 7.9505850e-007;
eb1[ 57]= 5.7000000e+005;
 b1[ 57]= 9.0032350e-007;
eb1[ 58]= 5.8000000e+005;
 b1[ 58]= 9.2454210e-007;
eb1[ 59]= 5.9000000e+005;
 b1[ 59]= 7.2494060e-007;
eb1[ 60]= 6.0000000e+005;
 b1[ 60]= 6.1113220e-007;
eb1[ 61]= 6.1000000e+005;
 b1[ 61]= 5.4271030e-007;
eb1[ 62]= 6.2000000e+005;
```

```
 b1[ 62]= 3.7186750e-007;
eb1[ 63]= 6.3000000e+005;
 b1[ 63]= 3.7783810e-007;
eb1[ 64]= 6.4000000e+005;
 b1[ 64]= 3.4935150e-007;
eb1[ 65]= 6.5000000e+005;
 b1[ 65]= 3.3129270e-007;
eb1[ 66]= 6.6000000e+005;
 b1[ 66]= 2.7793070e-007;
eb1[ 67]= 6.7000000e+005;
 b1[ 67]= 3.3123970e-007;
eb1[ 68]= 6.8000000e+005;
 b1[ 68]= 3.8705320e-007;
eb1[ 69]= 6.9000000e+005;
 b1[ 69]= 3.1405330e-007;
eb1[ 70]= 7.0000000e+005;
 b1[ 70]= 3.0606000e-007;
eb1[ 71]= 7.1000000e+005;
 b1[ 71]= 3.4054080e-007;
eb1[ 72]= 7.2000000e+005;
 b1[ 72]= 3.9295560e-007;
eb1[ 73]= 7.3000000e+005;
 b1[ 73]= 4.8489230e-007;
eb1[ 74]= 7.4000000e+005;
 b1[ 74]= 6.0266530e-007;
eb1[ 75]= 7.5000000e+005;
 b1[ 75]= 5.1302230e-007;
eb1[ 76]= 7.6000000e+005;
 b1[ 76]= 4.7379900e-007;
eb1[ 77]= 7.7000000e+005;
 b1[ 77]= 4.1896790e-007;
eb1[ 78]= 7.8000000e+005;
 b1[ 78]= 3.7708300e-007;
eb1[ 79]= 7.9000000e+005;
 b1[ 79]= 2.7883990e-007;
eb1[ 80]= 8.0000000e+005;
 b1[ 80]= 2.8609210e-007;
eb1[ 81]= 8.1000000e+005;
 b1[ 81]= 2.9175340e-007;
eb1[ 82]= 8.2000000e+005;
 b1[ 82]= 3.1253650e-007;
eb1[ 83]= 8.3000000e+005;
 b1[ 83]= 4.0444640e-007;
eb1[ 84]= 8.4000000e+005;
 b1[ 84]= 6.9158160e-007;
eb1[ 85]= 8.5000000e+005;
 b1[ 85]= 7.4276060e-007;
eb1[ 86]= 8.6000000e+005;
 b1[ 86]= 5.0668460e-007;
eb1[ 87]= 8.7000000e+005;
 b1[ 87]= 3.3776040e-007;
eb1[ 88]= 8.8000000e+005;
 b1[ 88]= 2.8047450e-007;
eb1[ 89]= 8.9000000e+005;
 b1[ 89]= 2.2385060e-007;
eb1[ 90]= 9.0000000e+005;
 b1[ 90]= 2.2218850e-007;
eb1[ 91]= 9.1000000e+005;
 b1[ 91]= 2.4169570e-007;
eb1[ 92]= 9.2000000e+005;
 b1[ 92]= 2.5248070e-007;
eb1[ 93]= 9.3000000e+005;
 b1[ 93]= 3.3791620e-007;
eb1[ 94]= 9.4000000e+005;
 b1[ 94]= 4.3919400e-007;
eb1[ 95]= 9.5000000e+005;
 b1[ 95]= 4.1440740e-007;
eb1[ 96]= 9.6000000e+005;
 b1[ 96]= 3.3660040e-007;
eb1[ 97]= 9.7000000e+005;
 b1[ 97]= 2.9985030e-007;
eb1[ 98]= 9.8000000e+005;
 b1[ 98]= 2.6709260e-007;
eb1[ 99]= 9.9000000e+005;
```

```
 b1[ 99]= 2.6292460e-007;
eb1[100]= 1.0000000e+006;
 b1[100]= 2.2045220e-007;
eb1[101]= 1.0100000e+006;
 b1[101]= 1.8137610e-007;
eb1[102]= 1.0200000e+006;
 b1[102]= 1.5532050e-007;
eb1[103]= 1.0300000e+006;
 b1[103]= 1.9232130e-007;
eb1[104]= 1.0400000e+006;
 b1[104]= 2.5076590e-007;
eb1[105]= 1.0500000e+006;
 b1[105]= 2.7307480e-007;
eb1[106]= 1.0600000e+006;
 b1[106]= 2.0252510e-007;
eb1[107]= 1.0700000e+006;
 b1[107]= 1.5000270e-007;
eb1[108]= 1.0800000e+006;
 b1[108]= 1.4738900e-007;
eb1[109]= 1.0900000e+006;
 b1[109]= 1.6256010e-007;
eb1[110]= 1.1000000e+006;
 b1[110]= 1.3848800e-007;
eb1[111]= 1.1100000e+006;
 b1[111]= 2.0249590e-007;
eb1[112]= 1.1200000e+006;
 b1[112]= 2.7152580e-007;
eb1[113]= 1.1300000e+006;
 b1[113]= 3.3732520e-007;
eb1[114]= 1.1400000e+006;
 b1[114]= 3.8656040e-007;
eb1[115]= 1.1500000e+006;
 b1[115]= 3.0101520e-007;
eb1[116]= 1.1600000e+006;
 b1[116]= 2.1068140e-007;
eb1[117]= 1.1700000e+006;
 b1[117]= 2.3223020e-007;
eb1[118]= 1.1800000e+006;
 b1[118]= 2.2189040e-007;
eb1[119]= 1.1900000e+006;
 b1[119]= 2.1620800e-007;
eb1[120]= 1.2000000e+006;
 b1[120]= 1.9032010e-007;
eb1[121]= 1.2100000e+006;
 b1[121]= 1.6056110e-007;
eb1[122]= 1.2200000e+006;
 b1[122]= 9.1099230e-008;
eb1[123]= 1.2300000e+006;
 b1[123]= 8.8747780e-008;
eb1[124]= 1.2400000e+006;
 b1[124]= 7.7003790e-008;
eb1[125]= 1.2500000e+006;
 b1[125]= 9.5079580e-008;
eb1[126]= 1.2600000e+006;
 b1[126]= 7.0511210e-008;
eb1[127]= 1.2700000e+006;
 b1[127]= 5.4205650e-008;
eb1[128]= 1.2800000e+006;
 b1[128]= 5.7144840e-008;
eb1[129]= 1.2900000e+006;
 b1[129]= 5.1591350e-008;
eb1[130]= 1.3000000e+006;
 b1[130]= 7.3959710e-008;
eb1[131]= 1.3100000e+006;
 b1[131]= 7.1609230e-008;
eb1[132]= 1.3200000e+006;
 b1[132]= 8.0865020e-008;
eb1[133]= 1.3300000e+006;
 b1[133]= 6.4650970e-008;
eb1[134]= 1.3400000e+006;
 b1[134]= 8.3989660e-008;
eb1[135]= 1.3500000e+006;
 b1[135]= 7.4459860e-008;
eb1[136]= 1.3600000e+006;
```

```
 b1[136]= 9.4653770e-008;
eb1[137]= 1.3700000e+006;
 b1[137]= 9.7006400e-008;
eb1[138]= 1.3800000e+006;
 b1[138]= 7.8644500e-008;
eb1[139]= 1.3900000e+006;
 b1[139]= 6.6884840e-008;
eb1[140]= 1.4000000e+006;
 b1[140]= 3.9151230e-008;
eb1[141]= 1.4100000e+006;
 b1[141]= 4.1502680e-008;
eb1[142]= 1.4200000e+006;
 b1[142]= 4.5305550e-008;
eb1[143]= 1.4300000e+006;
 b1[143]= 4.3158550e-008;
eb1[144]= 1.4400000e+006;
 b1[144]= 7.6912280e-008;
eb1[145]= 1.4500000e+006;
 b1[145]= 3.8456140e-008;
eb1[146]= 1.4600000e+006;
 b1[146]= 4.2436490e-008;
eb1[147]= 1.4700000e+006;
 b1[147]= 2.8747290e-008;
eb1[148]= 1.4800000e+006;
 b1[148]= 2.1980230e-008;
eb1[149]= 1.4900000e+006;
 b1[149]= 1.3415160e-008;
eb1[150]= 1.5000000e+006;
 b1[150]= 2.1289390e-008;
eb1[151]= 1.5100000e+006;
 b1[151]= 1.5510560e-008;
eb1[152]= 1.5200000e+006;
 b1[152]= 4.2230610e-009;
eb1[153]= 1.5300000e+006;
 b1[153]= 2.1220760e-008;
eb1[154]= 1.5400000e+006;
 b1[154]= 4.3502500e-010;
eb1[155]= 1.5500000e+006;
 b1[155]= 3.6534590e-009;
eb1[156]= 1.5600000e+006;
 b1[156]= 2.0516190e-008;
eb1[157]= 1.5700000e+006;
 b1[157]= 2.2986760e-008;
eb1[158]= 1.5800000e+006;
 b1[158]= 8.2997200e-010;
eb1[159]= 1.5900000e+006;
 b1[159]= 8.0984140e-009;
eb1[160]= 1.6000000e+006;
 b1[160]= 9.4343500e-010;
eb1[161]= 1.6100000e+006;
 b1[161]= 1.0363100e-008;
eb1[162]= 1.6200000e+006;
 b1[162]= 8.6291270e-009;
eb1[163]= 1.6300000e+006;
 b1[163]= 4.5021890e-009;
eb1[164]= 1.6400000e+006;
 b1[164]= 1.9802020e-008;
eb1[165]= 1.6500000e+006;
 b1[165]= 2.5059290e-008;
eb1[166]= 1.6600000e+006;
 b1[166]= 5.4219710e-009;
eb1[167]= 1.6700000e+006;
 b1[167]= 1.1804300e-010;
eb1[168]= 1.6800000e+006;
 b1[168]= 3.3839390e-009;
eb1[169]= 1.6900000e+006;
 b1[169]= 8.1350150e-009;
eb1[170]= 1.7000000e+006;
 b1[170]= 0.0000000e+000;
eb1[171]= 1.7100000e+006;
 b1[171]= 3.4960300e-009;
eb1[172]= 1.7200000e+006;
 b1[172]= 1.1104270e-009;
eb1[173]= 1.7300000e+006;
```

```
 b1[173]= 0.0000000e+000;
eb1[174]= 1.7400000e+006;
 b1[174]= 2.8514000e-010;
eb1[175]= 1.7500000e+006;
 b1[175]= 1.8951000e-010;
eb1[176]= 1.7600000e+006;
 b1[176]= 1.1406200e-010;
eb1[177]= 1.7700000e+006;
 b1[177]= 5.8291600e-011;
eb1[178]= 1.7800000e+006;
 b1[178]= 2.1824700e-011;
eb1[179]= 1.7900000e+006;
 b1[179]= 4.1814300e-012;
eb1[180]= 1.8000000e+006;
 b1[180]= 5.5958300e-031;
eb1[181]= 1.8100000e+006;
 b1[181]= 0.0000000e+000;
eb1[182]= 1.8200000e+006;
 b1[182]= 0.0000000e+000;
eb1[183]= 1.8300000e+006;
 b1[183]= 0.0000000e+000;
eb1[184]= 1.8400000e+006;
 b1[184]= 0.0000000e+000;
eb1[185]= 20.000000e+006;
 b1[185]= 0.0000000e+000;
eb2[  0]= 0.0000000e+000;
 b2[  0]= 1.0515390e-006;
eb2[  1]= 1.0000000e+004;
 b2[  1]= 1.0763120e-006;
eb2[  2]= 2.0000000e+004;
 b2[  2]= 1.0800620e-006;
eb2[  3]= 3.0000000e+004;
 b2[  3]= 1.0700410e-006;
eb2[  4]= 4.0000000e+004;
 b2[  4]= 1.0606630e-006;
eb2[  5]= 5.0000000e+004;
 b2[  5]= 1.0577670e-006;
eb2[  6]= 6.0000000e+004;
 b2[  6]= 1.0529180e-006;
eb2[  7]= 7.0000000e+004;
 b2[  7]= 1.3563340e-006;
eb2[  8]= 8.0000000e+004;
 b2[  8]= 1.5663060e-006;
eb2[  9]= 9.0000000e+004;
 b2[  9]= 1.6479710e-006;
eb2[ 10]= 1.0000000e+005;
 b2[ 10]= 1.4255730e-006;
eb2[ 11]= 1.1000000e+005;
 b2[ 11]= 1.3123720e-006;
eb2[ 12]= 1.2000000e+005;
 b2[ 12]= 1.3301650e-006;
eb2[ 13]= 1.3000000e+005;
 b2[ 13]= 1.2481040e-006;
eb2[ 14]= 1.4000000e+005;
 b2[ 14]= 1.1568440e-006;
eb2[ 15]= 1.5000000e+005;
 b2[ 15]= 1.2024630e-006;
eb2[ 16]= 1.6000000e+005;
 b2[ 16]= 9.8173980e-007;
eb2[ 17]= 1.7000000e+005;
 b2[ 17]= 8.8840110e-007;
eb2[ 18]= 1.8000000e+005;
 b2[ 18]= 8.9690780e-007;
eb2[ 19]= 1.9000000e+005;
 b2[ 19]= 9.1398390e-007;
eb2[ 20]= 2.0000000e+005;
 b2[ 20]= 8.9212460e-007;
eb2[ 21]= 2.1000000e+005;
 b2[ 21]= 1.0164730e-006;
eb2[ 22]= 2.2000000e+005;
 b2[ 22]= 1.0664770e-006;
eb2[ 23]= 2.3000000e+005;
 b2[ 23]= 1.0604140e-006;
eb2[ 24]= 2.4000000e+005;
```

```
 b2[ 24]= 1.1601270e-006;
eb2[ 25]= 2.5000000e+005;
 b2[ 25]= 1.5522230e-006;
eb2[ 26]= 2.6000000e+005;
 b2[ 26]= 1.4687350e-006;
eb2[ 27]= 2.7000000e+005;
 b2[ 27]= 1.3094820e-006;
eb2[ 28]= 2.8000000e+005;
 b2[ 28]= 9.6738970e-007;
eb2[ 29]= 2.9000000e+005;
 b2[ 29]= 7.9123850e-007;
eb2[ 30]= 3.0000000e+005;
 b2[ 30]= 8.0281970e-007;
eb2[ 31]= 3.1000000e+005;
 b2[ 31]= 1.1269290e-006;
eb2[ 32]= 3.2000000e+005;
 b2[ 32]= 1.1958020e-006;
eb2[ 33]= 3.3000000e+005;
 b2[ 33]= 1.0565680e-006;
eb2[ 34]= 3.4000000e+005;
 b2[ 34]= 1.0157050e-006;
eb2[ 35]= 3.5000000e+005;
 b2[ 35]= 1.1354010e-006;
eb2[ 36]= 3.6000000e+005;
 b2[ 36]= 1.8018930e-006;
eb2[ 37]= 3.7000000e+005;
 b2[ 37]= 2.2836700e-006;
eb2[ 38]= 3.8000000e+005;
 b2[ 38]= 1.9389550e-006;
eb2[ 39]= 3.9000000e+005;
 b2[ 39]= 1.6371910e-006;
eb2[ 40]= 4.0000000e+005;
 b2[ 40]= 1.4063460e-006;
eb2[ 41]= 4.1000000e+005;
 b2[ 41]= 1.2295840e-006;
eb2[ 42]= 4.2000000e+005;
 b2[ 42]= 1.1660950e-006;
eb2[ 43]= 4.3000000e+005;
 b2[ 43]= 1.0576310e-006;
eb2[ 44]= 4.4000000e+005;
 b2[ 44]= 1.0162430e-006;
eb2[ 45]= 4.5000000e+005;
 b2[ 45]= 1.1475590e-006;
eb2[ 46]= 4.6000000e+005;
 b2[ 46]= 1.3503870e-006;
eb2[ 47]= 4.7000000e+005;
 b2[ 47]= 1.5077290e-006;
eb2[ 48]= 4.8000000e+005;
 b2[ 48]= 1.4157470e-006;
eb2[ 49]= 4.9000000e+005;
 b2[ 49]= 1.4290800e-006;
eb2[ 50]= 5.0000000e+005;
 b2[ 50]= 1.4165000e-006;
eb2[ 51]= 5.1000000e+005;
 b2[ 51]= 1.2973680e-006;
eb2[ 52]= 5.2000000e+005;
 b2[ 52]= 1.0836610e-006;
eb2[ 53]= 5.3000000e+005;
 b2[ 53]= 9.0744000e-007;
eb2[ 54]= 5.4000000e+005;
 b2[ 54]= 9.0452130e-007;
eb2[ 55]= 5.5000000e+005;
 b2[ 55]= 1.0754170e-006;
eb2[ 56]= 5.6000000e+005;
 b2[ 56]= 1.0525560e-006;
eb2[ 57]= 5.7000000e+005;
 b2[ 57]= 1.1721940e-006;
eb2[ 58]= 5.8000000e+005;
 b2[ 58]= 1.1390740e-006;
eb2[ 59]= 5.9000000e+005;
 b2[ 59]= 8.6442890e-007;
eb2[ 60]= 6.0000000e+005;
 b2[ 60]= 7.1309910e-007;
eb2[ 61]= 6.1000000e+005;
```

```
 b2[ 61]= 6.1369020e-007;
eb2[ 62]= 6.2000000e+005;
 b2[ 62]= 4.5017710e-007;
eb2[ 63]= 6.3000000e+005;
 b2[ 63]= 4.3206930e-007;
eb2[ 64]= 6.4000000e+005;
 b2[ 64]= 4.1481620e-007;
eb2[ 65]= 6.5000000e+005;
 b2[ 65]= 4.4103720e-007;
eb2[ 66]= 6.6000000e+005;
 b2[ 66]= 4.3674980e-007;
eb2[ 67]= 6.7000000e+005;
 b2[ 67]= 5.0574690e-007;
eb2[ 68]= 6.8000000e+005;
 b2[ 68]= 4.8788300e-007;
eb2[ 69]= 6.9000000e+005;
 b2[ 69]= 4.0697900e-007;
eb2[ 70]= 7.0000000e+005;
 b2[ 70]= 3.8472520e-007;
eb2[ 71]= 7.1000000e+005;
 b2[ 71]= 4.5362860e-007;
eb2[ 72]= 7.2000000e+005;
 b2[ 72]= 5.2691040e-007;
eb2[ 73]= 7.3000000e+005;
 b2[ 73]= 6.3565050e-007;
eb2[ 74]= 7.4000000e+005;
 b2[ 74]= 7.8211860e-007;
eb2[ 75]= 7.5000000e+005;
 b2[ 75]= 6.8678060e-007;
eb2[ 76]= 7.6000000e+005;
 b2[ 76]= 6.1950470e-007;
eb2[ 77]= 7.7000000e+005;
 b2[ 77]= 5.5272300e-007;
eb2[ 78]= 7.8000000e+005;
 b2[ 78]= 4.7472190e-007;
eb2[ 79]= 7.9000000e+005;
 b2[ 79]= 3.6189770e-007;
eb2[ 80]= 8.0000000e+005;
 b2[ 80]= 3.5143200e-007;
eb2[ 81]= 8.1000000e+005;
 b2[ 81]= 3.5429560e-007;
eb2[ 82]= 8.2000000e+005;
 b2[ 82]= 3.6220880e-007;
eb2[ 83]= 8.3000000e+005;
 b2[ 83]= 5.0202190e-007;
eb2[ 84]= 8.4000000e+005;
 b2[ 84]= 8.5932640e-007;
eb2[ 85]= 8.5000000e+005;
 b2[ 85]= 9.2307030e-007;
eb2[ 86]= 8.6000000e+005;
 b2[ 86]= 6.3006740e-007;
eb2[ 87]= 8.7000000e+005;
 b2[ 87]= 4.1223760e-007;
eb2[ 88]= 8.8000000e+005;
 b2[ 88]= 3.4966060e-007;
eb2[ 89]= 8.9000000e+005;
 b2[ 89]= 2.7640800e-007;
eb2[ 90]= 9.0000000e+005;
 b2[ 90]= 2.5489220e-007;
eb2[ 91]= 9.1000000e+005;
 b2[ 91]= 2.8426020e-007;
eb2[ 92]= 9.2000000e+005;
 b2[ 92]= 3.1388620e-007;
eb2[ 93]= 9.3000000e+005;
 b2[ 93]= 4.0217560e-007;
eb2[ 94]= 9.4000000e+005;
 b2[ 94]= 5.1585060e-007;
eb2[ 95]= 9.5000000e+005;
 b2[ 95]= 5.2159630e-007;
eb2[ 96]= 9.6000000e+005;
 b2[ 96]= 4.3845460e-007;
eb2[ 97]= 9.7000000e+005;
 b2[ 97]= 3.7443530e-007;
eb2[ 98]= 9.8000000e+005;
```

```
 b2[ 98]= 3.3213260e-007;
eb2[ 99]= 9.9000000e+005;
 b2[ 99]= 3.1779460e-007;
eb2[100]= 1.0000000e+006;
 b2[100]= 2.8078650e-007;
eb2[101]= 1.0100000e+006;
 b2[101]= 2.3296910e-007;
eb2[102]= 1.0200000e+006;
 b2[102]= 1.9869580e-007;
eb2[103]= 1.0300000e+006;
 b2[103]= 2.4186660e-007;
eb2[104]= 1.0400000e+006;
 b2[104]= 3.1960240e-007;
eb2[105]= 1.0500000e+006;
 b2[105]= 3.4905720e-007;
eb2[106]= 1.0600000e+006;
 b2[106]= 2.5738030e-007;
eb2[107]= 1.0700000e+006;
 b2[107]= 1.9173360e-007;
eb2[108]= 1.0800000e+006;
 b2[108]= 1.8687120e-007;
eb2[109]= 1.0900000e+006;
 b2[109]= 2.0580520e-007;
eb2[110]= 1.1000000e+006;
 b2[110]= 1.7305620e-007;
eb2[111]= 1.1100000e+006;
 b2[111]= 2.4886720e-007;
eb2[112]= 1.1200000e+006;
 b2[112]= 3.2940680e-007;
eb2[113]= 1.1300000e+006;
 b2[113]= 4.1989740e-007;
eb2[114]= 1.1400000e+006;
 b2[114]= 4.8439270e-007;
eb2[115]= 1.1500000e+006;
 b2[115]= 3.7810070e-007;
eb2[116]= 1.1600000e+006;
 b2[116]= 2.6058430e-007;
eb2[117]= 1.1700000e+006;
 b2[117]= 2.8758020e-007;
eb2[118]= 1.1800000e+006;
 b2[118]= 2.7495820e-007;
eb2[119]= 1.1900000e+006;
 b2[119]= 2.6093710e-007;
eb2[120]= 1.2000000e+006;
 b2[120]= 2.2967900e-007;
eb2[121]= 1.2100000e+006;
 b2[121]= 2.0162160e-007;
eb2[122]= 1.2200000e+006;
 b2[122]= 1.1951350e-007;
eb2[123]= 1.2300000e+006;
 b2[123]= 1.1935090e-007;
eb2[124]= 1.2400000e+006;
 b2[124]= 1.0614790e-007;
eb2[125]= 1.2500000e+006;
 b2[125]= 1.2388280e-007;
eb2[126]= 1.2600000e+006;
 b2[126]= 9.0542740e-008;
eb2[127]= 1.2700000e+006;
 b2[127]= 6.8245140e-008;
eb2[128]= 1.2800000e+006;
 b2[128]= 6.7737570e-008;
eb2[129]= 1.2900000e+006;
 b2[129]= 6.8700420e-008;
eb2[130]= 1.3000000e+006;
 b2[130]= 9.1914570e-008;
eb2[131]= 1.3100000e+006;
 b2[131]= 8.9385640e-008;
eb2[132]= 1.3200000e+006;
 b2[132]= 9.8965310e-008;
eb2[133]= 1.3300000e+006;
 b2[133]= 8.2207700e-008;
eb2[134]= 1.3400000e+006;
 b2[134]= 1.0491840e-007;
eb2[135]= 1.3500000e+006;
```

```
 b2[135]= 9.1026430e-008;
eb2[136]= 1.3600000e+006;
 b2[136]= 1.1281490e-007;
eb2[137]= 1.3700000e+006;
 b2[137]= 1.1196310e-007;
eb2[138]= 1.3800000e+006;
 b2[138]= 7.4063490e-008;
eb2[139]= 1.3900000e+006;
 b2[139]= 7.8681520e-008;
eb2[140]= 1.4000000e+006;
 b2[140]= 5.3087340e-008;
eb2[141]= 1.4100000e+006;
 b2[141]= 5.1822110e-008;
eb2[142]= 1.4200000e+006;
 b2[142]= 5.9963010e-008;
eb2[143]= 1.4300000e+006;
 b2[143]= 4.9983590e-008;
eb2[144]= 1.4400000e+006;
 b2[144]= 9.5447430e-008;
eb2[145]= 1.4500000e+006;
 b2[145]= 4.7694320e-008;
eb2[146]= 1.4600000e+006;
 b2[146]= 5.2409120e-008;
eb2[147]= 1.4700000e+006;
 b2[147]= 3.5730860e-008;
eb2[148]= 1.4800000e+006;
 b2[148]= 2.8549240e-008;
eb2[149]= 1.4900000e+006;
 b2[149]= 1.9136980e-008;
eb2[150]= 1.5000000e+006;
 b2[150]= 2.9072530e-008;
eb2[151]= 1.5100000e+006;
 b2[151]= 2.1314040e-008;
eb2[152]= 1.5200000e+006;
 b2[152]= 6.5461280e-009;
eb2[153]= 1.5300000e+006;
 b2[153]= 2.7899340e-008;
eb2[154]= 1.5400000e+006;
 b2[154]= 3.5495120e-009;
eb2[155]= 1.5500000e+006;
 b2[155]= 7.6967420e-009;
eb2[156]= 1.5600000e+006;
 b2[156]= 2.7193730e-008;
eb2[157]= 1.5700000e+006;
 b2[157]= 3.0639010e-008;
eb2[158]= 1.5800000e+006;
 b2[158]= 4.0798970e-009;
eb2[159]= 1.5900000e+006;
 b2[159]= 1.2985910e-008;
eb2[160]= 1.6000000e+006;
 b2[160]= 3.7167560e-009;
eb2[161]= 1.6100000e+006;
 b2[161]= 1.3615180e-008;
eb2[162]= 1.6200000e+006;
 b2[162]= 1.1433780e-008;
eb2[163]= 1.6300000e+006;
 b2[163]= 6.7905040e-009;
eb2[164]= 1.6400000e+006;
 b2[164]= 2.5500520e-008;
eb2[165]= 1.6500000e+006;
 b2[165]= 3.1904050e-008;
eb2[166]= 1.6600000e+006;
 b2[166]= 7.5120250e-009;
eb2[167]= 1.6700000e+006;
 b2[167]= 9.6503000e-010;
eb2[168]= 1.6800000e+006;
 b2[168]= 6.4633940e-009;
eb2[169]= 1.6900000e+006;
 b2[169]= 1.3569370e-008;
eb2[170]= 1.7000000e+006;
 b2[170]= 4.9448790e-009;
eb2[171]= 1.7100000e+006;
 b2[171]= 8.0145280e-009;
eb2[172]= 1.7200000e+006;
```

```
 b2[172]= 2.6987840e-009;
eb2[173]= 1.7300000e+006;
 b2[173]= 1.3672810e-009;
eb2[174]= 1.7400000e+006;
 b2[174]= 9.0183400e-010;
eb2[175]= 1.7500000e+006;
 b2[175]= 1.2194060e-009;
eb2[176]= 1.7600000e+006;
 b2[176]= 4.1593850e-009;
eb2[177]= 1.7700000e+006;
 b2[177]= 6.2791480e-009;
eb2[178]= 1.7800000e+006;
 b2[178]= 8.6756180e-009;
eb2[179]= 1.7900000e+006;
 b2[179]= 8.4832530e-009;
eb2[180]= 1.8000000e+006;
 b2[180]= 3.1930800e-011;
eb2[181]= 1.8100000e+006;
 b2[181]= 2.9311000e-011;
eb2[182]= 1.8200000e+006;
 b2[182]= 2.6869700e-011;
eb2[183]= 1.8300000e+006;
 b2[183]= 2.4595400e-011;
eb2[184]= 1.8400000e+006;
 b2[184]= 2.2479700e-011;
eb2[185]= 1.8500000e+006;
 b2[185]= 2.0493700e-011;
eb2[186]= 1.8600000e+006;
 b2[186]= 1.8603900e-011;
eb2[187]= 1.8700000e+006;
 b2[187]= 1.6843200e-011;
eb2[188]= 1.8800000e+006;
 b2[188]= 1.5184100e-011;
eb2[189]= 1.8900000e+006;
 b2[189]= 1.3632100e-011;
eb2[190]= 1.9000000e+006;
 b2[190]= 1.2192000e-011;
eb2[191]= 1.9100000e+006;
 b2[191]= 1.3571100e-011;
eb2[192]= 1.9200000e+006;
 b2[192]= 1.0703300e-011;
eb2[193]= 1.9300000e+006;
 b2[193]= 8.7486700e-012;
eb2[194]= 1.9400000e+006;
 b2[194]= 7.5378800e-012;
eb2[195]= 1.9500000e+006;
 b2[195]= 6.6313700e-012;
eb2[196]= 1.9600000e+006;
 b2[196]= 5.7908700e-012;
eb2[197]= 1.9700000e+006;
 b2[197]= 5.0219500e-012;
eb2[198]= 1.9800000e+006;
 b2[198]= 4.3325800e-012;
eb2[199]= 1.9900000e+006;
 b2[199]= 3.7026400e-012;
eb2[200]= 2.0000000e+006;
 b2[200]= 3.1273000e-012;
eb2[201]= 2.0100000e+006;
 b2[201]= 2.6156100e-012;
eb2[202]= 2.0200000e+006;
 b2[202]= 2.1670900e-012;
eb2[203]= 2.0300000e+006;
 b2[203]= 1.7844300e-012;
eb2[204]= 2.0400000e+006;
 b2[204]= 1.4654100e-012;
eb2[205]= 2.0500000e+006;
 b2[205]= 1.1919000e-012;
eb2[206]= 2.0600000e+006;
 b2[206]= 9.4735400e-013;
eb2[207]= 2.0700000e+006;
 b2[207]= 7.2952100e-013;
eb2[208]= 2.0800000e+006;
 b2[208]= 5.4000700e-013;
eb2[209]= 2.0900000e+006;
```

```
 b2[209]= 3.7791400e-013;
eb2[210]= 2.1000000e+006;
 b2[210]= 2.4427500e-013;
eb2[211]= 2.1100000e+006;
 b2[211]= 1.3953300e-013;
eb2[212]= 2.1200000e+006;
 b2[212]= 6.4065300e-014;
eb2[213]= 2.1300000e+006;
 b2[213]= 1.8496900e-014;
eb2[214]= 2.1400000e+006;
 b2[214]= 2.2570300e-015;
eb2[215]= 2.1500000e+006;
 b2[215]= 0.0000000e+000;
eb2[216]= 2.1600000e+006;
 b2[216]= 0.0000000e+000;
eb2[217]= 2.1700000e+006;
 b2[217]= 0.0000000e+000;
eb2[218]= 2.1800000e+006;
 b2[218]= 0.0000000e+000;
eb2[219]= 2.1900000e+006;
 b2[219]= 0.0000000e+000;
eb2[220]= 20.000000e+006;
 b2[220]= 0.0000000e+000;
eb3[  0]= 0.0000000e+000;
 b3[  0]= 7.3623090e-007;
eb3[  1]= 1.0000000e+004;
 b3[  1]= 1.0874920e-006;
eb3[  2]= 2.0000000e+004;
 b3[  2]= 1.1093480e-006;
eb3[  3]= 3.0000000e+004;
 b3[  3]= 1.1108600e-006;
eb3[  4]= 4.0000000e+004;
 b3[  4]= 1.0951400e-006;
eb3[  5]= 5.0000000e+004;
 b3[  5]= 1.1689930e-006;
eb3[  6]= 6.0000000e+004;
 b3[  6]= 1.1030840e-006;
eb3[  7]= 7.0000000e+004;
 b3[  7]= 1.0621310e-006;
eb3[  8]= 8.0000000e+004;
 b3[  8]= 1.1932370e-006;
eb3[  9]= 9.0000000e+004;
 b3[  9]= 1.3800830e-006;
eb3[ 10]= 1.0000000e+005;
 b3[ 10]= 1.4964580e-006;
eb3[ 11]= 1.1000000e+005;
 b3[ 11]= 1.6037340e-006;
eb3[ 12]= 1.2000000e+005;
 b3[ 12]= 1.6277940e-006;
eb3[ 13]= 1.3000000e+005;
 b3[ 13]= 1.5661720e-006;
eb3[ 14]= 1.4000000e+005;
 b3[ 14]= 1.4800900e-006;
eb3[ 15]= 1.5000000e+005;
 b3[ 15]= 1.3386210e-006;
eb3[ 16]= 1.6000000e+005;
 b3[ 16]= 1.3852410e-006;
eb3[ 17]= 1.7000000e+005;
 b3[ 17]= 1.4447050e-006;
eb3[ 18]= 1.8000000e+005;
 b3[ 18]= 1.3652470e-006;
eb3[ 19]= 1.9000000e+005;
 b3[ 19]= 1.3267560e-006;
eb3[ 20]= 2.0000000e+005;
 b3[ 20]= 1.2519220e-006;
eb3[ 21]= 2.1000000e+005;
 b3[ 21]= 1.3726970e-006;
eb3[ 22]= 2.2000000e+005;
 b3[ 22]= 1.3659820e-006;
eb3[ 23]= 2.3000000e+005;
 b3[ 23]= 1.4417140e-006;
eb3[ 24]= 2.4000000e+005;
 b3[ 24]= 1.4125840e-006;
eb3[ 25]= 2.5000000e+005;
```

```
 b3[ 25]= 1.2187050e-006;
eb3[ 26]= 2.6000000e+005;
 b3[ 26]= 1.3091140e-006;
eb3[ 27]= 2.7000000e+005;
 b3[ 27]= 1.4721220e-006;
eb3[ 28]= 2.8000000e+005;
 b3[ 28]= 1.4480080e-006;
eb3[ 29]= 2.9000000e+005;
 b3[ 29]= 1.4168730e-006;
eb3[ 30]= 3.0000000e+005;
 b3[ 30]= 1.3209120e-006;
eb3[ 31]= 3.1000000e+005;
 b3[ 31]= 1.2905760e-006;
eb3[ 32]= 3.2000000e+005;
 b3[ 32]= 1.2805610e-006;
eb3[ 33]= 3.3000000e+005;
 b3[ 33]= 1.2897820e-006;
eb3[ 34]= 3.4000000e+005;
 b3[ 34]= 1.2301970e-006;
eb3[ 35]= 3.5000000e+005;
 b3[ 35]= 1.2930720e-006;
eb3[ 36]= 3.6000000e+005;
 b3[ 36]= 1.3461340e-006;
eb3[ 37]= 3.7000000e+005;
 b3[ 37]= 1.4145260e-006;
eb3[ 38]= 3.8000000e+005;
 b3[ 38]= 1.4433500e-006;
eb3[ 39]= 3.9000000e+005;
 b3[ 39]= 1.3797690e-006;
eb3[ 40]= 4.0000000e+005;
 b3[ 40]= 1.2186210e-006;
eb3[ 41]= 4.1000000e+005;
 b3[ 41]= 1.0744450e-006;
eb3[ 42]= 4.2000000e+005;
 b3[ 42]= 9.8701650e-007;
eb3[ 43]= 4.3000000e+005;
 b3[ 43]= 1.0058210e-006;
eb3[ 44]= 4.4000000e+005;
 b3[ 44]= 1.0063590e-006;
eb3[ 45]= 4.5000000e+005;
 b3[ 45]= 9.9749270e-007;
eb3[ 46]= 4.6000000e+005;
 b3[ 46]= 9.7657910e-007;
eb3[ 47]= 4.7000000e+005;
 b3[ 47]= 9.7371860e-007;
eb3[ 48]= 4.8000000e+005;
 b3[ 48]= 9.1897430e-007;
eb3[ 49]= 4.9000000e+005;
 b3[ 49]= 9.1955920e-007;
eb3[ 50]= 5.0000000e+005;
 b3[ 50]= 9.4844360e-007;
eb3[ 51]= 5.1000000e+005;
 b3[ 51]= 8.9677710e-007;
eb3[ 52]= 5.2000000e+005;
 b3[ 52]= 8.6966560e-007;
eb3[ 53]= 5.3000000e+005;
 b3[ 53]= 8.8350110e-007;
eb3[ 54]= 5.4000000e+005;
 b3[ 54]= 8.3784590e-007;
eb3[ 55]= 5.5000000e+005;
 b3[ 55]= 8.3663860e-007;
eb3[ 56]= 5.6000000e+005;
 b3[ 56]= 8.1303170e-007;
eb3[ 57]= 5.7000000e+005;
 b3[ 57]= 8.1248990e-007;
eb3[ 58]= 5.8000000e+005;
 b3[ 58]= 9.0800630e-007;
eb3[ 59]= 5.9000000e+005;
 b3[ 59]= 1.0968810e-006;
eb3[ 60]= 6.0000000e+005;
 b3[ 60]= 1.1766510e-006;
eb3[ 61]= 6.1000000e+005;
 b3[ 61]= 1.0598120e-006;
eb3[ 62]= 6.2000000e+005;
```

```
 b3[ 62]= 1.0155490e-006;
eb3[ 63]= 6.3000000e+005;
 b3[ 63]= 8.7646950e-007;
eb3[ 64]= 6.4000000e+005;
 b3[ 64]= 8.8886850e-007;
eb3[ 65]= 6.5000000e+005;
 b3[ 65]= 9.5259790e-007;
eb3[ 66]= 6.6000000e+005;
 b3[ 66]= 9.3986340e-007;
eb3[ 67]= 6.7000000e+005;
 b3[ 67]= 8.8223520e-007;
eb3[ 68]= 6.8000000e+005;
 b3[ 68]= 8.0135650e-007;
eb3[ 69]= 6.9000000e+005;
 b3[ 69]= 7.7792370e-007;
eb3[ 70]= 7.0000000e+005;
 b3[ 70]= 7.5206190e-007;
eb3[ 71]= 7.1000000e+005;
 b3[ 71]= 7.6305100e-007;
eb3[ 72]= 7.2000000e+005;
 b3[ 72]= 7.6992290e-007;
eb3[ 73]= 7.3000000e+005;
 b3[ 73]= 8.1994740e-007;
eb3[ 74]= 7.4000000e+005;
 b3[ 74]= 7.5434740e-007;
eb3[ 75]= 7.5000000e+005;
 b3[ 75]= 6.5065510e-007;
eb3[ 76]= 7.6000000e+005;
 b3[ 76]= 7.7111860e-007;
eb3[ 77]= 7.7000000e+005;
 b3[ 77]= 8.0125930e-007;
eb3[ 78]= 7.8000000e+005;
 b3[ 78]= 6.3593030e-007;
eb3[ 79]= 7.9000000e+005;
 b3[ 79]= 6.5602120e-007;
eb3[ 80]= 8.0000000e+005;
 b3[ 80]= 6.0034980e-007;
eb3[ 81]= 8.1000000e+005;
 b3[ 81]= 5.2361530e-007;
eb3[ 82]= 8.2000000e+005;
 b3[ 82]= 4.2804020e-007;
eb3[ 83]= 8.3000000e+005;
 b3[ 83]= 3.7595530e-007;
eb3[ 84]= 8.4000000e+005;
 b3[ 84]= 3.2668910e-007;
eb3[ 85]= 8.5000000e+005;
 b3[ 85]= 3.2282300e-007;
eb3[ 86]= 8.6000000e+005;
 b3[ 86]= 3.0662700e-007;
eb3[ 87]= 8.7000000e+005;
 b3[ 87]= 3.2405740e-007;
eb3[ 88]= 8.8000000e+005;
 b3[ 88]= 3.2975530e-007;
eb3[ 89]= 8.9000000e+005;
 b3[ 89]= 3.4948070e-007;
eb3[ 90]= 9.0000000e+005;
 b3[ 90]= 3.5291910e-007;
eb3[ 91]= 9.1000000e+005;
 b3[ 91]= 3.1742680e-007;
eb3[ 92]= 9.2000000e+005;
 b3[ 92]= 2.4581370e-007;
eb3[ 93]= 9.3000000e+005;
 b3[ 93]= 2.2090590e-007;
eb3[ 94]= 9.4000000e+005;
 b3[ 94]= 2.0443730e-007;
eb3[ 95]= 9.5000000e+005;
 b3[ 95]= 2.1448330e-007;
eb3[ 96]= 9.6000000e+005;
 b3[ 96]= 2.1058670e-007;
eb3[ 97]= 9.7000000e+005;
 b3[ 97]= 2.1818130e-007;
eb3[ 98]= 9.8000000e+005;
 b3[ 98]= 2.2990550e-007;
eb3[ 99]= 9.9000000e+005;
```

```
 b3[ 99]= 2.1842780e-007;
eb3[100]= 1.0000000e+006;
 b3[100]= 1.7389300e-007;
eb3[101]= 1.0100000e+006;
 b3[101]= 1.8819670e-007;
eb3[102]= 1.0200000e+006;
 b3[102]= 1.6192420e-007;
eb3[103]= 1.0300000e+006;
 b3[103]= 1.4670100e-007;
eb3[104]= 1.0400000e+006;
 b3[104]= 1.4270080e-007;
eb3[105]= 1.0500000e+006;
 b3[105]= 1.1864080e-007;
eb3[106]= 1.0600000e+006;
 b3[106]= 1.4317600e-007;
eb3[107]= 1.0700000e+006;
 b3[107]= 1.2139040e-007;
eb3[108]= 1.0800000e+006;
 b3[108]= 1.2162330e-007;
eb3[109]= 1.0900000e+006;
 b3[109]= 1.1481070e-007;
eb3[110]= 1.1000000e+006;
 b3[110]= 1.0955470e-007;
eb3[111]= 1.1100000e+006;
 b3[111]= 1.1777020e-007;
eb3[112]= 1.1200000e+006;
 b3[112]= 1.3288110e-007;
eb3[113]= 1.1300000e+006;
 b3[113]= 1.0591900e-007;
eb3[114]= 1.1400000e+006;
 b3[114]= 1.1125670e-007;
eb3[115]= 1.1500000e+006;
 b3[115]= 1.0471610e-007;
eb3[116]= 1.1600000e+006;
 b3[116]= 1.1497730e-007;
eb3[117]= 1.1700000e+006;
 b3[117]= 1.0688110e-007;
eb3[118]= 1.1800000e+006;
 b3[118]= 1.1316310e-007;
eb3[119]= 1.1900000e+006;
 b3[119]= 8.9393870e-008;
eb3[120]= 1.2000000e+006;
 b3[120]= 9.6042810e-008;
eb3[121]= 1.2100000e+006;
 b3[121]= 7.5092760e-008;
eb3[122]= 1.2200000e+006;
 b3[122]= 1.1195640e-007;
eb3[123]= 1.2300000e+006;
 b3[123]= 8.2938050e-008;
eb3[124]= 1.2400000e+006;
 b3[124]= 7.4063090e-008;
eb3[125]= 1.2500000e+006;
 b3[125]= 6.5986000e-008;
eb3[126]= 1.2600000e+006;
 b3[126]= 7.3147490e-008;
eb3[127]= 1.2700000e+006;
 b3[127]= 6.4860040e-008;
eb3[128]= 1.2800000e+006;
 b3[128]= 5.2557550e-008;
eb3[129]= 1.2900000e+006;
 b3[129]= 7.3950900e-008;
eb3[130]= 1.3000000e+006;
 b3[130]= 6.5329870e-008;
eb3[131]= 1.3100000e+006;
 b3[131]= 4.3043050e-008;
eb3[132]= 1.3200000e+006;
 b3[132]= 5.1310950e-008;
eb3[133]= 1.3300000e+006;
 b3[133]= 6.1184630e-008;
eb3[134]= 1.3400000e+006;
 b3[134]= 6.2567130e-008;
eb3[135]= 1.3500000e+006;
 b3[135]= 5.6436720e-008;
eb3[136]= 1.3600000e+006;
```

```
 b3[136]= 5.3897730e-008;
eb3[137]= 1.3700000e+006;
 b3[137]= 4.2678960e-008;
eb3[138]= 1.3800000e+006;
 b3[138]= 6.5176290e-008;
eb3[139]= 1.3900000e+006;
 b3[139]= 3.2091550e-008;
eb3[140]= 1.4000000e+006;
 b3[140]= 4.2938540e-008;
eb3[141]= 1.4100000e+006;
 b3[141]= 4.6706390e-008;
eb3[142]= 1.4200000e+006;
 b3[142]= 4.5433130e-008;
eb3[143]= 1.4300000e+006;
 b3[143]= 3.7745200e-008;
eb3[144]= 1.4400000e+006;
 b3[144]= 4.4642660e-008;
eb3[145]= 1.4500000e+006;
 b3[145]= 2.7453880e-008;
eb3[146]= 1.4600000e+006;
 b3[146]= 2.0760400e-008;
eb3[147]= 1.4700000e+006;
 b3[147]= 2.5156790e-008;
eb3[148]= 1.4800000e+006;
 b3[148]= 2.5837770e-008;
eb3[149]= 1.4900000e+006;
 b3[149]= 4.1328170e-008;
eb3[150]= 1.5000000e+006;
 b3[150]= 2.2823690e-008;
eb3[151]= 1.5100000e+006;
 b3[151]= 2.7988070e-008;
eb3[152]= 1.5200000e+006;
 b3[152]= 3.7918410e-008;
eb3[153]= 1.5300000e+006;
 b3[153]= 4.2105200e-008;
eb3[154]= 1.5400000e+006;
 b3[154]= 3.1300660e-008;
eb3[155]= 1.5500000e+006;
 b3[155]= 3.6036850e-008;
eb3[156]= 1.5600000e+006;
 b3[156]= 3.4590730e-008;
eb3[157]= 1.5700000e+006;
 b3[157]= 3.0060780e-008;
eb3[158]= 1.5800000e+006;
 b3[158]= 3.1661950e-008;
eb3[159]= 1.5900000e+006;
 b3[159]= 3.0106900e-008;
eb3[160]= 1.6000000e+006;
 b3[160]= 2.4053860e-008;
eb3[161]= 1.6100000e+006;
 b3[161]= 2.5244600e-008;
eb3[162]= 1.6200000e+006;
 b3[162]= 1.4444910e-008;
eb3[163]= 1.6300000e+006;
 b3[163]= 3.1469660e-008;
eb3[164]= 1.6400000e+006;
 b3[164]= 3.3816410e-008;
eb3[165]= 1.6500000e+006;
 b3[165]= 2.9549730e-008;
eb3[166]= 1.6600000e+006;
 b3[166]= 1.8798960e-008;
eb3[167]= 1.6700000e+006;
 b3[167]= 2.0810700e-008;
eb3[168]= 1.6800000e+006;
 b3[168]= 1.6588210e-008;
eb3[169]= 1.6900000e+006;
 b3[169]= 2.9494610e-008;
eb3[170]= 1.7000000e+006;
 b3[170]= 4.1558190e-008;
eb3[171]= 1.7100000e+006;
 b3[171]= 2.5863500e-008;
eb3[172]= 1.7200000e+006;
 b3[172]= 2.8768690e-008;
eb3[173]= 1.7300000e+006;
```

```
 b3[173]= 3.1011290e-008;
eb3[174]= 1.7400000e+006;
 b3[174]= 1.2030730e-008;
eb3[175]= 1.7500000e+006;
 b3[175]= 1.4620750e-008;
eb3[176]= 1.7600000e+006;
 b3[176]= 2.4056830e-008;
eb3[177]= 1.7700000e+006;
 b3[177]= 2.0542030e-008;
eb3[178]= 1.7800000e+006;
 b3[178]= 1.9559930e-008;
eb3[179]= 1.7900000e+006;
 b3[179]= 1.6476670e-008;
eb3[180]= 1.8000000e+006;
 b3[180]= 1.1982890e-008;
eb3[181]= 1.8100000e+006;
 b3[181]= 1.3020750e-008;
eb3[182]= 1.8200000e+006;
 b3[182]= 7.2590950e-009;
eb3[183]= 1.8300000e+006;
 b3[183]= 4.3820460e-009;
eb3[184]= 1.8400000e+006;
 b3[184]= 9.2027840e-009;
eb3[185]= 1.8500000e+006;
 b3[185]= 4.7312400e-009;
eb3[186]= 1.8600000e+006;
 b3[186]= 5.9386460e-009;
eb3[187]= 1.8700000e+006;
 b3[187]= 7.5269350e-009;
eb3[188]= 1.8800000e+006;
 b3[188]= 6.9362210e-009;
eb3[189]= 1.8900000e+006;
 b3[189]= 4.4413090e-009;
eb3[190]= 1.9000000e+006;
 b3[190]= 1.4324580e-008;
eb3[191]= 1.9100000e+006;
 b3[191]= 8.7118120e-009;
eb3[192]= 1.9200000e+006;
 b3[192]= 1.2918120e-008;
eb3[193]= 1.9300000e+006;
 b3[193]= 3.7701100e-009;
eb3[194]= 1.9400000e+006;
 b3[194]= 7.0020910e-009;
eb3[195]= 1.9500000e+006;
 b3[195]= 2.6783700e-009;
eb3[196]= 1.9600000e+006;
 b3[196]= 5.2264300e-009;
eb3[197]= 1.9700000e+006;
 b3[197]= 7.2397680e-009;
eb3[198]= 1.9800000e+006;
 b3[198]= 1.1158030e-008;
eb3[199]= 1.9900000e+006;
 b3[199]= 4.1909290e-009;
eb3[200]= 2.0000000e+006;
 b3[200]= 3.4833680e-009;
eb3[201]= 2.0100000e+006;
 b3[201]= 7.0904660e-009;
eb3[202]= 2.0200000e+006;
 b3[202]= 5.8491740e-009;
eb3[203]= 2.0300000e+006;
 b3[203]= 2.3087450e-009;
eb3[204]= 2.0400000e+006;
 b3[204]= 4.8086760e-009;
eb3[205]= 2.0500000e+006;
 b3[205]= 2.4108260e-009;
eb3[206]= 2.0600000e+006;
 b3[206]= 5.9661920e-009;
eb3[207]= 2.0700000e+006;
 b3[207]= 3.2299340e-009;
eb3[208]= 2.0800000e+006;
 b3[208]= 5.2390530e-009;
eb3[209]= 2.0900000e+006;
 b3[209]= 5.1394840e-009;
eb3[210]= 2.1000000e+006;
```

```
 b3[210]= 4.8137990e-009;
eb3[211]= 2.1100000e+006;
 b3[211]= 4.5353130e-009;
eb3[212]= 2.1200000e+006;
 b3[212]= 2.9956290e-009;
eb3[213]= 2.1300000e+006;
 b3[213]= 3.4748500e-010;
eb3[214]= 2.1400000e+006;
 b3[214]= 2.1740800e-009;
eb3[215]= 2.1500000e+006;
 b3[215]= 2.1887520e-009;
eb3[216]= 2.1600000e+006;
 b3[216]= 7.9002100e-010;
eb3[217]= 2.1700000e+006;
 b3[217]= 2.3026410e-009;
eb3[218]= 2.1800000e+006;
 b3[218]= 5.0824620e-009;
eb3[219]= 2.1900000e+006;
 b3[219]= 2.6744110e-009;
eb3[220]= 2.2000000e+006;
 b3[220]= 2.2327820e-009;
eb3[221]= 2.2100000e+006;
 b3[221]= 5.0319400e-010;
eb3[222]= 2.2200000e+006;
 b3[222]= 2.6706400e-011;
eb3[223]= 2.2300000e+006;
 b3[223]= 2.5565800e-011;
eb3[224]= 2.2400000e+006;
 b3[224]= 4.9302400e-010;
eb3[225]= 2.2500000e+006;
 b3[225]= 8.8786700e-010;
eb3[226]= 2.2600000e+006;
 b3[226]= 7.5095200e-010;
eb3[227]= 2.2700000e+006;
 b3[227]= 7.5383500e-010;
eb3[228]= 2.2800000e+006;
 b3[228]= 3.9940040e-009;
eb3[229]= 2.2900000e+006;
 b3[229]= 9.5543700e-010;
eb3[230]= 2.3000000e+006;
 b3[230]= 3.6276130e-009;
eb3[231]= 2.3100000e+006;
 b3[231]= 1.7941400e-011;
eb3[232]= 2.3200000e+006;
 b3[232]= 4.4576900e-010;
eb3[233]= 2.3300000e+006;
 b3[233]= 4.4503000e-010;
eb3[234]= 2.3400000e+006;
 b3[234]= 1.5723700e-011;
eb3[235]= 2.3500000e+006;
 b3[235]= 1.1119100e-010;
eb3[236]= 2.3600000e+006;
 b3[236]= 7.7011600e-010;
eb3[237]= 2.3700000e+006;
 b3[237]= 2.3724060e-009;
eb3[238]= 2.3800000e+006;
 b3[238]= 5.3089200e-010;
eb3[239]= 2.3900000e+006;
 b3[239]= 8.5467900e-011;
eb3[240]= 2.4000000e+006;
 b3[240]= 3.1301890e-009;
eb3[241]= 2.4100000e+006;
 b3[241]= 5.6829000e-010;
eb3[242]= 2.4200000e+006;
 b3[242]= 1.2157300e-009;
eb3[243]= 2.4300000e+006;
 b3[243]= 1.2152450e-009;
eb3[244]= 2.4400000e+006;
 b3[244]= 9.9871000e-012;
eb3[245]= 2.4500000e+006;
 b3[245]= 2.8584100e-010;
eb3[246]= 2.4600000e+006;
 b3[246]= 2.7091450e-009;
eb3[247]= 2.4700000e+006;
```

```
 b3[247]= 2.7190010e-009;
eb3[248]= 2.4800000e+006;
 b3[248]= 2.5430800e-010;
eb3[249]= 2.4900000e+006;
 b3[249]= 2.4442000e-010;
eb3[250]= 2.5000000e+006;
 b3[250]= 2.3492000e-010;
eb3[251]= 2.5100000e+006;
 b3[251]= 2.2569000e-010;
eb3[252]= 2.5200000e+006;
 b3[252]= 2.1685300e-010;
eb3[253]= 2.5300000e+006;
 b3[253]= 2.0814200e-010;
eb3[254]= 2.5400000e+006;
 b3[254]= 1.9983100e-010;
eb3[255]= 2.5500000e+006;
 b3[255]= 1.9190000e-010;
eb3[256]= 2.5600000e+006;
 b3[256]= 1.8410300e-010;
eb3[257]= 2.5700000e+006;
 b3[257]= 1.7669000e-010;
eb3[258]= 2.5800000e+006;
 b3[258]= 4.8174400e-010;
eb3[259]= 2.5900000e+006;
 b3[259]= 4.5673900e-010;
eb3[260]= 2.6000000e+006;
 b3[260]= 4.3277100e-010;
eb3[261]= 2.6100000e+006;
 b3[261]= 4.0961200e-010;
eb3[262]= 2.6200000e+006;
 b3[262]= 3.8764900e-010;
eb3[263]= 2.6300000e+006;
 b3[263]= 3.6671700e-010;
eb3[264]= 2.6400000e+006;
 b3[264]= 3.4676500e-010;
eb3[265]= 2.6500000e+006;
 b3[265]= 3.2770000e-010;
eb3[266]= 2.6600000e+006;
 b3[266]= 3.0948500e-010;
eb3[267]= 2.6700000e+006;
 b3[267]= 2.9196600e-010;
eb3[268]= 2.6800000e+006;
 b3[268]= 2.7515800e-010;
eb3[269]= 2.6900000e+006;
 b3[269]= 2.5914300e-010;
eb3[270]= 2.7000000e+006;
 b3[270]= 2.4385700e-010;
eb3[271]= 2.7100000e+006;
 b3[271]= 2.2941900e-010;
eb3[272]= 2.7200000e+006;
 b3[272]= 2.1613700e-010;
eb3[273]= 2.7300000e+006;
 b3[273]= 2.0360800e-010;
eb3[274]= 2.7400000e+006;
 b3[274]= 1.9161500e-010;
eb3[275]= 2.7500000e+006;
 b3[275]= 1.8011900e-010;
eb3[276]= 2.7600000e+006;
 b3[276]= 1.6912500e-010;
eb3[277]= 2.7700000e+006;
 b3[277]= 1.5868200e-010;
eb3[278]= 2.7800000e+006;
 b3[278]= 1.4874200e-010;
eb3[279]= 2.7900000e+006;
 b3[279]= 1.3924400e-010;
eb3[280]= 2.8000000e+006;
 b3[280]= 1.3026900e-010;
eb3[281]= 2.8100000e+006;
 b3[281]= 1.2174900e-010;
eb3[282]= 2.8200000e+006;
 b3[282]= 1.1369100e-010;
eb3[283]= 2.8300000e+006;
 b3[283]= 1.0616300e-010;
eb3[284]= 2.8400000e+006;
```

```
 b3[284]= 9.8956800e-011;
eb3[285]= 2.8500000e+006;
 b3[285]= 9.2141800e-011;
eb3[286]= 2.8600000e+006;
 b3[286]= 8.5611600e-011;
eb3[287]= 2.8700000e+006;
 b3[287]= 7.9489200e-011;
eb3[288]= 2.8800000e+006;
 b3[288]= 7.3682700e-011;
eb3[289]= 2.8900000e+006;
 b3[289]= 6.8207400e-011;
eb3[290]= 2.9000000e+006;
 b3[290]= 6.3043900e-011;
eb3[291]= 2.9100000e+006;
 b3[291]= 5.8256300e-011;
eb3[292]= 2.9200000e+006;
 b3[292]= 5.3806300e-011;
eb3[293]= 2.9300000e+006;
 b3[293]= 4.9691000e-011;
eb3[294]= 2.9400000e+006;
 b3[294]= 4.5799700e-011;
eb3[295]= 2.9500000e+006;
 b3[295]= 4.2154800e-011;
eb3[296]= 2.9600000e+006;
 b3[296]= 3.8741300e-011;
eb3[297]= 2.9700000e+006;
 b3[297]= 3.5554900e-011;
eb3[298]= 2.9800000e+006;
 b3[298]= 3.2575000e-011;
eb3[299]= 2.9900000e+006;
 b3[299]= 2.9802300e-011;
eb3[300]= 3.0000000e+006;
 b3[300]= 0.0000000e+000;
eb3[301]= 20.000000e+006;
 b3[301]= 0.0000000e+000;
eb4[  0]= 0.0000000e+000;
 b4[  0]= 4.6906190e-007;
eb4[  1]= 1.0000000e+004;
 b4[  1]= 9.1690400e-007;
eb4[  2]= 2.0000000e+004;
 b4[  2]= 9.2202380e-007;
eb4[  3]= 3.0000000e+004;
 b4[  3]= 8.4312920e-007;
eb4[  4]= 4.0000000e+004;
 b4[  4]= 8.4146540e-007;
eb4[  5]= 5.0000000e+004;
 b4[  5]= 7.8908120e-007;
eb4[  6]= 6.0000000e+004;
 b4[  6]= 9.2624400e-007;
eb4[  7]= 7.0000000e+004;
 b4[  7]= 9.5769110e-007;
eb4[  8]= 8.0000000e+004;
 b4[  8]= 9.9526590e-007;
eb4[  9]= 9.0000000e+004;
 b4[  9]= 1.0267900e-006;
eb4[ 10]= 1.0000000e+005;
 b4[ 10]= 1.0036840e-006;
eb4[ 11]= 1.1000000e+005;
 b4[ 11]= 1.0929280e-006;
eb4[ 12]= 1.2000000e+005;
 b4[ 12]= 1.1730330e-006;
eb4[ 13]= 1.3000000e+005;
 b4[ 13]= 1.2049300e-006;
eb4[ 14]= 1.4000000e+005;
 b4[ 14]= 1.2204620e-006;
eb4[ 15]= 1.5000000e+005;
 b4[ 15]= 1.2950450e-006;
eb4[ 16]= 1.6000000e+005;
 b4[ 16]= 1.4007900e-006;
eb4[ 17]= 1.7000000e+005;
 b4[ 17]= 1.3370550e-006;
eb4[ 18]= 1.8000000e+005;
 b4[ 18]= 1.2597460e-006;
eb4[ 19]= 1.9000000e+005;
```

```
 b4[ 19]= 1.1271540e-006;
eb4[ 20]= 2.0000000e+005;
 b4[ 20]= 1.1845020e-006;
eb4[ 21]= 2.1000000e+005;
 b4[ 21]= 1.1842540e-006;
eb4[ 22]= 2.2000000e+005;
 b4[ 22]= 1.1784780e-006;
eb4[ 23]= 2.3000000e+005;
 b4[ 23]= 1.1947270e-006;
eb4[ 24]= 2.4000000e+005;
 b4[ 24]= 1.2048930e-006;
eb4[ 25]= 2.5000000e+005;
 b4[ 25]= 1.0763750e-006;
eb4[ 26]= 2.6000000e+005;
 b4[ 26]= 1.1547200e-006;
eb4[ 27]= 2.7000000e+005;
 b4[ 27]= 1.1941650e-006;
eb4[ 28]= 2.8000000e+005;
 b4[ 28]= 1.2249190e-006;
eb4[ 29]= 2.9000000e+005;
 b4[ 29]= 1.2694270e-006;
eb4[ 30]= 3.0000000e+005;
 b4[ 30]= 1.2226450e-006;
eb4[ 31]= 3.1000000e+005;
 b4[ 31]= 1.1884210e-006;
eb4[ 32]= 3.2000000e+005;
 b4[ 32]= 1.2014770e-006;
eb4[ 33]= 3.3000000e+005;
 b4[ 33]= 1.2496690e-006;
eb4[ 34]= 3.4000000e+005;
 b4[ 34]= 1.1600930e-006;
eb4[ 35]= 3.5000000e+005;
 b4[ 35]= 1.1127810e-006;
eb4[ 36]= 3.6000000e+005;
 b4[ 36]= 1.1101610e-006;
eb4[ 37]= 3.7000000e+005;
 b4[ 37]= 1.0775710e-006;
eb4[ 38]= 3.8000000e+005;
 b4[ 38]= 1.0965350e-006;
eb4[ 39]= 3.9000000e+005;
 b4[ 39]= 1.1024910e-006;
eb4[ 40]= 4.0000000e+005;
 b4[ 40]= 1.0273860e-006;
eb4[ 41]= 4.1000000e+005;
 b4[ 41]= 1.0407200e-006;
eb4[ 42]= 4.2000000e+005;
 b4[ 42]= 9.4107160e-007;
eb4[ 43]= 4.3000000e+005;
 b4[ 43]= 9.4084250e-007;
eb4[ 44]= 4.4000000e+005;
 b4[ 44]= 9.3049530e-007;
eb4[ 45]= 4.5000000e+005;
 b4[ 45]= 9.1472470e-007;
eb4[ 46]= 4.6000000e+005;
 b4[ 46]= 9.6386200e-007;
eb4[ 47]= 4.7000000e+005;
 b4[ 47]= 1.1220730e-006;
eb4[ 48]= 4.8000000e+005;
 b4[ 48]= 1.1888080e-006;
eb4[ 49]= 4.9000000e+005;
 b4[ 49]= 1.2304510e-006;
eb4[ 50]= 5.0000000e+005;
 b4[ 50]= 1.1346610e-006;
eb4[ 51]= 5.1000000e+005;
 b4[ 51]= 1.0268460e-006;
eb4[ 52]= 5.2000000e+005;
 b4[ 52]= 9.7816960e-007;
eb4[ 53]= 5.3000000e+005;
 b4[ 53]= 9.7297560e-007;
eb4[ 54]= 5.4000000e+005;
 b4[ 54]= 9.6257910e-007;
eb4[ 55]= 5.5000000e+005;
 b4[ 55]= 8.7675280e-007;
eb4[ 56]= 5.6000000e+005;
```

```
 b4[ 56]= 8.7142560e-007;
eb4[ 57]= 5.7000000e+005;
 b4[ 57]= 7.9150880e-007;
eb4[ 58]= 5.8000000e+005;
 b4[ 58]= 7.5014170e-007;
eb4[ 59]= 5.9000000e+005;
 b4[ 59]= 7.6704380e-007;
eb4[ 60]= 6.0000000e+005;
 b4[ 60]= 8.6106340e-007;
eb4[ 61]= 6.1000000e+005;
 b4[ 61]= 8.9644440e-007;
eb4[ 62]= 6.2000000e+005;
 b4[ 62]= 9.0566100e-007;
eb4[ 63]= 6.3000000e+005;
 b4[ 63]= 8.2101050e-007;
eb4[ 64]= 6.4000000e+005;
 b4[ 64]= 7.4079980e-007;
eb4[ 65]= 6.5000000e+005;
 b4[ 65]= 6.9615430e-007;
eb4[ 66]= 6.6000000e+005;
 b4[ 66]= 7.1242310e-007;
eb4[ 67]= 6.7000000e+005;
 b4[ 67]= 6.7350730e-007;
eb4[ 68]= 6.8000000e+005;
 b4[ 68]= 6.3685300e-007;
eb4[ 69]= 6.9000000e+005;
 b4[ 69]= 6.5351050e-007;
eb4[ 70]= 7.0000000e+005;
 b4[ 70]= 6.7645560e-007;
eb4[ 71]= 7.1000000e+005;
 b4[ 71]= 5.7633080e-007;
eb4[ 72]= 7.2000000e+005;
 b4[ 72]= 5.5188910e-007;
eb4[ 73]= 7.3000000e+005;
 b4[ 73]= 5.6467470e-007;
eb4[ 74]= 7.4000000e+005;
 b4[ 74]= 5.3809640e-007;
eb4[ 75]= 7.5000000e+005;
 b4[ 75]= 5.4557640e-007;
eb4[ 76]= 7.6000000e+005;
 b4[ 76]= 5.5866990e-007;
eb4[ 77]= 7.7000000e+005;
 b4[ 77]= 5.4015870e-007;
eb4[ 78]= 7.8000000e+005;
 b4[ 78]= 5.3491560e-007;
eb4[ 79]= 7.9000000e+005;
 b4[ 79]= 5.1110150e-007;
eb4[ 80]= 8.0000000e+005;
 b4[ 80]= 4.4419370e-007;
eb4[ 81]= 8.1000000e+005;
 b4[ 81]= 4.5047470e-007;
eb4[ 82]= 8.2000000e+005;
 b4[ 82]= 4.4329130e-007;
eb4[ 83]= 8.3000000e+005;
 b4[ 83]= 4.0697590e-007;
eb4[ 84]= 8.4000000e+005;
 b4[ 84]= 3.9933770e-007;
eb4[ 85]= 8.5000000e+005;
 b4[ 85]= 4.1249340e-007;
eb4[ 86]= 8.6000000e+005;
 b4[ 86]= 3.6936920e-007;
eb4[ 87]= 8.7000000e+005;
 b4[ 87]= 3.8383970e-007;
eb4[ 88]= 8.8000000e+005;
 b4[ 88]= 4.4588510e-007;
eb4[ 89]= 8.9000000e+005;
 b4[ 89]= 4.6836340e-007;
eb4[ 90]= 9.0000000e+005;
 b4[ 90]= 4.6853320e-007;
eb4[ 91]= 9.1000000e+005;
 b4[ 91]= 5.3733240e-007;
eb4[ 92]= 9.2000000e+005;
 b4[ 92]= 4.8535500e-007;
eb4[ 93]= 9.3000000e+005;
```

```
 b4[ 93]= 4.3167050e-007;
eb4[ 94]= 9.4000000e+005;
 b4[ 94]= 3.9581840e-007;
eb4[ 95]= 9.5000000e+005;
 b4[ 95]= 4.2608060e-007;
eb4[ 96]= 9.6000000e+005;
 b4[ 96]= 3.7131440e-007;
eb4[ 97]= 9.7000000e+005;
 b4[ 97]= 4.3668400e-007;
eb4[ 98]= 9.8000000e+005;
 b4[ 98]= 3.9026860e-007;
eb4[ 99]= 9.9000000e+005;
 b4[ 99]= 3.8035590e-007;
eb4[100]= 1.0000000e+006;
 b4[100]= 4.1303150e-007;
eb4[101]= 1.0100000e+006;
 b4[101]= 4.5603950e-007;
eb4[102]= 1.0200000e+006;
 b4[102]= 3.3549110e-007;
eb4[103]= 1.0300000e+006;
 b4[103]= 4.0142800e-007;
eb4[104]= 1.0400000e+006;
 b4[104]= 3.0623070e-007;
eb4[105]= 1.0500000e+006;
 b4[105]= 2.3864860e-007;
eb4[106]= 1.0600000e+006;
 b4[106]= 2.8427980e-007;
eb4[107]= 1.0700000e+006;
 b4[107]= 2.5852250e-007;
eb4[108]= 1.0800000e+006;
 b4[108]= 3.2475690e-007;
eb4[109]= 1.0900000e+006;
 b4[109]= 2.6955920e-007;
eb4[110]= 1.1000000e+006;
 b4[110]= 2.3018840e-007;
eb4[111]= 1.1100000e+006;
 b4[111]= 2.5846640e-007;
eb4[112]= 1.1200000e+006;
 b4[112]= 2.8294310e-007;
eb4[113]= 1.1300000e+006;
 b4[113]= 2.6849910e-007;
eb4[114]= 1.1400000e+006;
 b4[114]= 2.9249900e-007;
eb4[115]= 1.1500000e+006;
 b4[115]= 2.4491130e-007;
eb4[116]= 1.1600000e+006;
 b4[116]= 2.8414340e-007;
eb4[117]= 1.1700000e+006;
 b4[117]= 2.4642110e-007;
eb4[118]= 1.1800000e+006;
 b4[118]= 2.6646560e-007;
eb4[119]= 1.1900000e+006;
 b4[119]= 2.2207850e-007;
eb4[120]= 1.2000000e+006;
 b4[120]= 2.8257850e-007;
eb4[121]= 1.2100000e+006;
 b4[121]= 1.8155250e-007;
eb4[122]= 1.2200000e+006;
 b4[122]= 1.8727990e-007;
eb4[123]= 1.2300000e+006;
 b4[123]= 1.9041670e-007;
eb4[124]= 1.2400000e+006;
 b4[124]= 1.6500080e-007;
eb4[125]= 1.2500000e+006;
 b4[125]= 1.8777230e-007;
eb4[126]= 1.2600000e+006;
 b4[126]= 2.1000420e-007;
eb4[127]= 1.2700000e+006;
 b4[127]= 1.7996690e-007;
eb4[128]= 1.2800000e+006;
 b4[128]= 1.5064540e-007;
eb4[129]= 1.2900000e+006;
 b4[129]= 1.4625080e-007;
eb4[130]= 1.3000000e+006;
```

```
 b4[130]= 1.8702970e-007;
eb4[131]= 1.3100000e+006;
 b4[131]= 1.2859950e-007;
eb4[132]= 1.3200000e+006;
 b4[132]= 1.1313950e-007;
eb4[133]= 1.3300000e+006;
 b4[133]= 1.6168180e-007;
eb4[134]= 1.3400000e+006;
 b4[134]= 1.3892550e-007;
eb4[135]= 1.3500000e+006;
 b4[135]= 1.8082730e-007;
eb4[136]= 1.3600000e+006;
 b4[136]= 1.4202410e-007;
eb4[137]= 1.3700000e+006;
 b4[137]= 1.5087960e-007;
eb4[138]= 1.3800000e+006;
 b4[138]= 2.3314790e-007;
eb4[139]= 1.3900000e+006;
 b4[139]= 1.2185290e-007;
eb4[140]= 1.4000000e+006;
 b4[140]= 2.2121560e-007;
eb4[141]= 1.4100000e+006;
 b4[141]= 1.7179600e-007;
eb4[142]= 1.4200000e+006;
 b4[142]= 1.8584270e-007;
eb4[143]= 1.4300000e+006;
 b4[143]= 1.3771830e-007;
eb4[144]= 1.4400000e+006;
 b4[144]= 1.6274370e-007;
eb4[145]= 1.4500000e+006;
 b4[145]= 1.6173410e-007;
eb4[146]= 1.4600000e+006;
 b4[146]= 1.7349550e-007;
eb4[147]= 1.4700000e+006;
 b4[147]= 1.2561340e-007;
eb4[148]= 1.4800000e+006;
 b4[148]= 1.2048440e-007;
eb4[149]= 1.4900000e+006;
 b4[149]= 1.2404150e-007;
eb4[150]= 1.5000000e+006;
 b4[150]= 8.2179530e-008;
eb4[151]= 1.5100000e+006;
 b4[151]= 7.8252170e-008;
eb4[152]= 1.5200000e+006;
 b4[152]= 1.5413700e-007;
eb4[153]= 1.5300000e+006;
 b4[153]= 1.0116020e-007;
eb4[154]= 1.5400000e+006;
 b4[154]= 8.6974000e-008;
eb4[155]= 1.5500000e+006;
 b4[155]= 8.8651530e-008;
eb4[156]= 1.5600000e+006;
 b4[156]= 1.2311410e-007;
eb4[157]= 1.5700000e+006;
 b4[157]= 6.3318490e-008;
eb4[158]= 1.5800000e+006;
 b4[158]= 3.6518470e-008;
eb4[159]= 1.5900000e+006;
 b4[159]= 7.9768420e-008;
eb4[160]= 1.6000000e+006;
 b4[160]= 7.7095590e-008;
eb4[161]= 1.6100000e+006;
 b4[161]= 7.6666980e-008;
eb4[162]= 1.6200000e+006;
 b4[162]= 6.9206190e-008;
eb4[163]= 1.6300000e+006;
 b4[163]= 6.9749600e-008;
eb4[164]= 1.6400000e+006;
 b4[164]= 8.1186320e-008;
eb4[165]= 1.6500000e+006;
 b4[165]= 7.2932350e-008;
eb4[166]= 1.6600000e+006;
 b4[166]= 5.9335730e-008;
eb4[167]= 1.6700000e+006;
```

```
 b4[167]= 6.8188170e-008;
eb4[168]= 1.6800000e+006;
 b4[168]= 4.5271050e-008;
eb4[169]= 1.6900000e+006;
 b4[169]= 6.1807150e-008;
eb4[170]= 1.7000000e+006;
 b4[170]= 5.0963440e-008;
eb4[171]= 1.7100000e+006;
 b4[171]= 4.9935340e-008;
eb4[172]= 1.7200000e+006;
 b4[172]= 5.2135140e-008;
eb4[173]= 1.7300000e+006;
 b4[173]= 6.3206940e-008;
eb4[174]= 1.7400000e+006;
 b4[174]= 5.5703650e-008;
eb4[175]= 1.7500000e+006;
 b4[175]= 4.9096560e-008;
eb4[176]= 1.7600000e+006;
 b4[176]= 4.7044810e-008;
eb4[177]= 1.7700000e+006;
 b4[177]= 3.7483810e-008;
eb4[178]= 1.7800000e+006;
 b4[178]= 4.0583340e-008;
eb4[179]= 1.7900000e+006;
 b4[179]= 2.7977110e-008;
eb4[180]= 1.8000000e+006;
 b4[180]= 3.8498580e-008;
eb4[181]= 1.8100000e+006;
 b4[181]= 3.7535750e-008;
eb4[182]= 1.8200000e+006;
 b4[182]= 4.7161750e-008;
eb4[183]= 1.8300000e+006;
 b4[183]= 3.5191880e-008;
eb4[184]= 1.8400000e+006;
 b4[184]= 2.9624540e-008;
eb4[185]= 1.8500000e+006;
 b4[185]= 3.6047590e-008;
eb4[186]= 1.8600000e+006;
 b4[186]= 3.3438620e-008;
eb4[187]= 1.8700000e+006;
 b4[187]= 3.2801160e-008;
eb4[188]= 1.8800000e+006;
 b4[188]= 4.4801240e-008;
eb4[189]= 1.8900000e+006;
 b4[189]= 4.5334540e-008;
eb4[190]= 1.9000000e+006;
 b4[190]= 4.6847640e-008;
eb4[191]= 1.9100000e+006;
 b4[191]= 5.8712100e-008;
eb4[192]= 1.9200000e+006;
 b4[192]= 5.5434880e-008;
eb4[193]= 1.9300000e+006;
 b4[193]= 4.4988420e-008;
eb4[194]= 1.9400000e+006;
 b4[194]= 4.3537620e-008;
eb4[195]= 1.9500000e+006;
 b4[195]= 4.1351180e-008;
eb4[196]= 1.9600000e+006;
 b4[196]= 3.5421330e-008;
eb4[197]= 1.9700000e+006;
 b4[197]= 2.5920720e-008;
eb4[198]= 1.9800000e+006;
 b4[198]= 2.4207710e-008;
eb4[199]= 1.9900000e+006;
 b4[199]= 2.9941450e-008;
eb4[200]= 2.0000000e+006;
 b4[200]= 1.8957920e-008;
eb4[201]= 2.0100000e+006;
 b4[201]= 2.4517670e-008;
eb4[202]= 2.0200000e+006;
 b4[202]= 2.6801490e-008;
eb4[203]= 2.0300000e+006;
 b4[203]= 2.2859400e-008;
eb4[204]= 2.0400000e+006;
```

```
 b4[204]= 2.8754160e-008;
eb4[205]= 2.0500000e+006;
 b4[205]= 2.8265670e-008;
eb4[206]= 2.0600000e+006;
 b4[206]= 2.0169420e-008;
eb4[207]= 2.0700000e+006;
 b4[207]= 1.9408210e-008;
eb4[208]= 2.0800000e+006;
 b4[208]= 1.9380010e-008;
eb4[209]= 2.0900000e+006;
 b4[209]= 1.7360530e-008;
eb4[210]= 2.1000000e+006;
 b4[210]= 1.6537540e-008;
eb4[211]= 2.1100000e+006;
 b4[211]= 2.3267040e-008;
eb4[212]= 2.1200000e+006;
 b4[212]= 1.5785100e-008;
eb4[213]= 2.1300000e+006;
 b4[213]= 2.2465530e-008;
eb4[214]= 2.1400000e+006;
 b4[214]= 1.3508660e-008;
eb4[215]= 2.1500000e+006;
 b4[215]= 1.6143370e-008;
eb4[216]= 2.1600000e+006;
 b4[216]= 1.0598450e-008;
eb4[217]= 2.1700000e+006;
 b4[217]= 1.8157340e-008;
eb4[218]= 2.1800000e+006;
 b4[218]= 1.5971960e-008;
eb4[219]= 2.1900000e+006;
 b4[219]= 1.3830140e-008;
eb4[220]= 2.2000000e+006;
 b4[220]= 1.8480160e-008;
eb4[221]= 2.2100000e+006;
 b4[221]= 1.6167310e-008;
eb4[222]= 2.2200000e+006;
 b4[222]= 1.8005760e-008;
eb4[223]= 2.2300000e+006;
 b4[223]= 1.5230130e-008;
eb4[224]= 2.2400000e+006;
 b4[224]= 9.9166360e-009;
eb4[225]= 2.2500000e+006;
 b4[225]= 1.5690450e-008;
eb4[226]= 2.2600000e+006;
 b4[226]= 1.0098340e-008;
eb4[227]= 2.2700000e+006;
 b4[227]= 1.2467990e-008;
eb4[228]= 2.2800000e+006;
 b4[228]= 1.2391660e-008;
eb4[229]= 2.2900000e+006;
 b4[229]= 1.3023410e-008;
eb4[230]= 2.3000000e+006;
 b4[230]= 1.4150590e-008;
eb4[231]= 2.3100000e+006;
 b4[231]= 6.4036020e-009;
eb4[232]= 2.3200000e+006;
 b4[232]= 1.8894980e-008;
eb4[233]= 2.3300000e+006;
 b4[233]= 1.3764590e-008;
eb4[234]= 2.3400000e+006;
 b4[234]= 1.4179140e-008;
eb4[235]= 2.3500000e+006;
 b4[235]= 1.6993500e-008;
eb4[236]= 2.3600000e+006;
 b4[236]= 7.2969850e-009;
eb4[237]= 2.3700000e+006;
 b4[237]= 1.1874240e-008;
eb4[238]= 2.3800000e+006;
 b4[238]= 9.4897940e-009;
eb4[239]= 2.3900000e+006;
 b4[239]= 1.1374620e-008;
eb4[240]= 2.4000000e+006;
 b4[240]= 1.4388660e-008;
eb4[241]= 2.4100000e+006;
```

```
 b4[241]= 8.0999850e-009;
eb4[242]= 2.4200000e+006;
 b4[242]= 8.6883290e-009;
eb4[243]= 2.4300000e+006;
 b4[243]= 5.8601730e-009;
eb4[244]= 2.4400000e+006;
 b4[244]= 6.7445610e-009;
eb4[245]= 2.4500000e+006;
 b4[245]= 7.2162290e-009;
eb4[246]= 2.4600000e+006;
 b4[246]= 7.6119210e-009;
eb4[247]= 2.4700000e+006;
 b4[247]= 7.5771860e-009;
eb4[248]= 2.4800000e+006;
 b4[248]= 5.9101560e-009;
eb4[249]= 2.4900000e+006;
 b4[249]= 8.7453920e-009;
eb4[250]= 2.5000000e+006;
 b4[250]= 5.8128120e-009;
eb4[251]= 2.5100000e+006;
 b4[251]= 5.7657110e-009;
eb4[252]= 2.5200000e+006;
 b4[252]= 3.6190820e-009;
eb4[253]= 2.5300000e+006;
 b4[253]= 6.4925830e-009;
eb4[254]= 2.5400000e+006;
 b4[254]= 3.5305530e-009;
eb4[255]= 2.5500000e+006;
 b4[255]= 3.4885670e-009;
eb4[256]= 2.5600000e+006;
 b4[256]= 3.4469600e-009;
eb4[257]= 2.5700000e+006;
 b4[257]= 6.3707990e-009;
eb4[258]= 2.5800000e+006;
 b4[258]= 3.4284330e-009;
eb4[259]= 2.5900000e+006;
 b4[259]= 3.3898590e-009;
eb4[260]= 2.6000000e+006;
 b4[260]= 2.9311980e-009;
eb4[261]= 2.6100000e+006;
 b4[261]= 2.8936470e-009;
eb4[262]= 2.6200000e+006;
 b4[262]= 2.8577860e-009;
eb4[263]= 2.6300000e+006;
 b4[263]= 2.8224000e-009;
eb4[264]= 2.6400000e+006;
 b4[264]= 2.7880860e-009;
eb4[265]= 2.6500000e+006;
 b4[265]= 2.7559120e-009;
eb4[266]= 2.6600000e+006;
 b4[266]= 2.7242680e-009;
eb4[267]= 2.6700000e+006;
 b4[267]= 2.6939660e-009;
eb4[268]= 2.6800000e+006;
 b4[268]= 1.8230160e-009;
eb4[269]= 2.6900000e+006;
 b4[269]= 1.7941370e-009;
eb4[270]= 2.7000000e+006;
 b4[270]= 4.8200310e-009;
eb4[271]= 2.7100000e+006;
 b4[271]= 1.7385400e-009;
eb4[272]= 2.7200000e+006;
 b4[272]= 1.7120140e-009;
eb4[273]= 2.7300000e+006;
 b4[273]= 1.6863270e-009;
eb4[274]= 2.7400000e+006;
 b4[274]= 1.6610150e-009;
eb4[275]= 2.7500000e+006;
 b4[275]= 1.6365210e-009;
eb4[276]= 2.7600000e+006;
 b4[276]= 4.7156700e-009;
eb4[277]= 2.7700000e+006;
 b4[277]= 4.7126490e-009;
eb4[278]= 2.7800000e+006;
```

```
 b4[278]= 1.5661470e-009;
eb4[279]= 2.7900000e+006;
 b4[279]= 5.1547440e-009;
eb4[280]= 2.8000000e+006;
 b4[280]= 5.0746660e-009;
eb4[281]= 2.8100000e+006;
 b4[281]= 1.8957220e-009;
eb4[282]= 2.8200000e+006;
 b4[282]= 2.8355130e-009;
eb4[283]= 2.8300000e+006;
 b4[283]= 2.7611590e-009;
eb4[284]= 2.8400000e+006;
 b4[284]= 2.6900920e-009;
eb4[285]= 2.8500000e+006;
 b4[285]= 2.6214230e-009;
eb4[286]= 2.8600000e+006;
 b4[286]= 2.5539750e-009;
eb4[287]= 2.8700000e+006;
 b4[287]= 2.4884650e-009;
eb4[288]= 2.8800000e+006;
 b4[288]= 2.4247710e-009;
eb4[289]= 2.8900000e+006;
 b4[289]= 2.3621510e-009;
eb4[290]= 2.9000000e+006;
 b4[290]= 2.2999620e-009;
eb4[291]= 2.9100000e+006;
 b4[291]= 2.2393730e-009;
eb4[292]= 2.9200000e+006;
 b4[292]= 2.1806530e-009;
eb4[293]= 2.9300000e+006;
 b4[293]= 2.1222830e-009;
eb4[294]= 2.9400000e+006;
 b4[294]= 2.0650610e-009;
eb4[295]= 2.9500000e+006;
 b4[295]= 2.0093130e-009;
eb4[296]= 2.9600000e+006;
 b4[296]= 1.9546630e-009;
eb4[297]= 2.9700000e+006;
 b4[297]= 1.9007070e-009;
eb4[298]= 2.9800000e+006;
 b4[298]= 1.8475450e-009;
eb4[299]= 2.9900000e+006;
 b4[299]= 1.7954240e-009;
eb4[300]= 3.0000000e+006;
 b4[300]= 0.0000000e+000;
eb4[301]= 20.000000e+006;
 b4[301]= 0.0000000e+000;
eb5[  0]= 0.0000000e+000;
 b5[  0]= 3.1349080e-007;
eb5[  1]= 1.0000000e+004;
 b5[  1]= 1.5351750e-006;
eb5[  2]= 2.0000000e+004;
 b5[  2]= 9.6561200e-007;
eb5[  3]= 3.0000000e+004;
 b5[  3]= 8.2377380e-007;
eb5[  4]= 4.0000000e+004;
 b5[  4]= 9.4940160e-007;
eb5[  5]= 5.0000000e+004;
 b5[  5]= 1.0236760e-006;
eb5[  6]= 6.0000000e+004;
 b5[  6]= 1.1387530e-006;
eb5[  7]= 7.0000000e+004;
 b5[  7]= 1.2989320e-006;
eb5[  8]= 8.0000000e+004;
 b5[  8]= 1.2863340e-006;
eb5[  9]= 9.0000000e+004;
 b5[  9]= 1.2824980e-006;
eb5[ 10]= 1.0000000e+005;
 b5[ 10]= 1.3189020e-006;
eb5[ 11]= 1.1000000e+005;
 b5[ 11]= 1.3897260e-006;
eb5[ 12]= 1.2000000e+005;
 b5[ 12]= 1.3891230e-006;
eb5[ 13]= 1.3000000e+005;
```

```
 b5[ 13]= 1.3774270e-006;
eb5[ 14]= 1.4000000e+005;
 b5[ 14]= 1.3802780e-006;
eb5[ 15]= 1.5000000e+005;
 b5[ 15]= 1.3748280e-006;
eb5[ 16]= 1.6000000e+005;
 b5[ 16]= 1.4641450e-006;
eb5[ 17]= 1.7000000e+005;
 b5[ 17]= 1.5165120e-006;
eb5[ 18]= 1.8000000e+005;
 b5[ 18]= 1.5327420e-006;
eb5[ 19]= 1.9000000e+005;
 b5[ 19]= 1.3874450e-006;
eb5[ 20]= 2.0000000e+005;
 b5[ 20]= 1.4430570e-006;
eb5[ 21]= 2.1000000e+005;
 b5[ 21]= 1.4396080e-006;
eb5[ 22]= 2.2000000e+005;
 b5[ 22]= 1.4658500e-006;
eb5[ 23]= 2.3000000e+005;
 b5[ 23]= 1.3620080e-006;
eb5[ 24]= 2.4000000e+005;
 b5[ 24]= 1.3686070e-006;
eb5[ 25]= 2.5000000e+005;
 b5[ 25]= 1.3703950e-006;
eb5[ 26]= 2.6000000e+005;
 b5[ 26]= 1.3790210e-006;
eb5[ 27]= 2.7000000e+005;
 b5[ 27]= 1.3182030e-006;
eb5[ 28]= 2.8000000e+005;
 b5[ 28]= 1.2915760e-006;
eb5[ 29]= 2.9000000e+005;
 b5[ 29]= 1.2221690e-006;
eb5[ 30]= 3.0000000e+005;
 b5[ 30]= 1.2718980e-006;
eb5[ 31]= 3.1000000e+005;
 b5[ 31]= 1.2490890e-006;
eb5[ 32]= 3.2000000e+005;
 b5[ 32]= 1.2017000e-006;
eb5[ 33]= 3.3000000e+005;
 b5[ 33]= 1.2249520e-006;
eb5[ 34]= 3.4000000e+005;
 b5[ 34]= 1.2437640e-006;
eb5[ 35]= 3.5000000e+005;
 b5[ 35]= 1.2134490e-006;
eb5[ 36]= 3.6000000e+005;
 b5[ 36]= 1.1747710e-006;
eb5[ 37]= 3.7000000e+005;
 b5[ 37]= 1.1134380e-006;
eb5[ 38]= 3.8000000e+005;
 b5[ 38]= 1.1068930e-006;
eb5[ 39]= 3.9000000e+005;
 b5[ 39]= 1.0857480e-006;
eb5[ 40]= 4.0000000e+005;
 b5[ 40]= 1.0026840e-006;
eb5[ 41]= 4.1000000e+005;
 b5[ 41]= 1.0232830e-006;
eb5[ 42]= 4.2000000e+005;
 b5[ 42]= 1.0179440e-006;
eb5[ 43]= 4.3000000e+005;
 b5[ 43]= 1.0777770e-006;
eb5[ 44]= 4.4000000e+005;
 b5[ 44]= 9.6880970e-007;
eb5[ 45]= 4.5000000e+005;
 b5[ 45]= 1.0026300e-006;
eb5[ 46]= 4.6000000e+005;
 b5[ 46]= 9.7749980e-007;
eb5[ 47]= 4.7000000e+005;
 b5[ 47]= 1.0174580e-006;
eb5[ 48]= 4.8000000e+005;
 b5[ 48]= 1.0245330e-006;
eb5[ 49]= 4.9000000e+005;
 b5[ 49]= 1.0633960e-006;
eb5[ 50]= 5.0000000e+005;
```

```
 b5[ 50]= 1.0217550e-006;
eb5[ 51]= 5.1000000e+005;
 b5[ 51]= 9.5211800e-007;
eb5[ 52]= 5.2000000e+005;
 b5[ 52]= 9.2140230e-007;
eb5[ 53]= 5.3000000e+005;
 b5[ 53]= 9.4699690e-007;
eb5[ 54]= 5.4000000e+005;
 b5[ 54]= 9.1804420e-007;
eb5[ 55]= 5.5000000e+005;
 b5[ 55]= 8.5076930e-007;
eb5[ 56]= 5.6000000e+005;
 b5[ 56]= 8.5519510e-007;
eb5[ 57]= 5.7000000e+005;
 b5[ 57]= 8.1382240e-007;
eb5[ 58]= 5.8000000e+005;
 b5[ 58]= 8.5125670e-007;
eb5[ 59]= 5.9000000e+005;
 b5[ 59]= 7.3946660e-007;
eb5[ 60]= 6.0000000e+005;
 b5[ 60]= 7.7441190e-007;
eb5[ 61]= 6.1000000e+005;
 b5[ 61]= 7.4660100e-007;
eb5[ 62]= 6.2000000e+005;
 b5[ 62]= 6.9016580e-007;
eb5[ 63]= 6.3000000e+005;
 b5[ 63]= 6.9906410e-007;
eb5[ 64]= 6.4000000e+005;
 b5[ 64]= 5.9733640e-007;
eb5[ 65]= 6.5000000e+005;
 b5[ 65]= 6.4258690e-007;
eb5[ 66]= 6.6000000e+005;
 b5[ 66]= 6.0213720e-007;
eb5[ 67]= 6.7000000e+005;
 b5[ 67]= 5.9365220e-007;
eb5[ 68]= 6.8000000e+005;
 b5[ 68]= 6.2085260e-007;
eb5[ 69]= 6.9000000e+005;
 b5[ 69]= 6.0799980e-007;
eb5[ 70]= 7.0000000e+005;
 b5[ 70]= 5.5610690e-007;
eb5[ 71]= 7.1000000e+005;
 b5[ 71]= 5.0920170e-007;
eb5[ 72]= 7.2000000e+005;
 b5[ 72]= 5.3962200e-007;
eb5[ 73]= 7.3000000e+005;
 b5[ 73]= 5.2729340e-007;
eb5[ 74]= 7.4000000e+005;
 b5[ 74]= 5.2074720e-007;
eb5[ 75]= 7.5000000e+005;
 b5[ 75]= 4.8662970e-007;
eb5[ 76]= 7.6000000e+005;
 b5[ 76]= 4.6440430e-007;
eb5[ 77]= 7.7000000e+005;
 b5[ 77]= 4.5003720e-007;
eb5[ 78]= 7.8000000e+005;
 b5[ 78]= 4.4710110e-007;
eb5[ 79]= 7.9000000e+005;
 b5[ 79]= 4.3723010e-007;
eb5[ 80]= 8.0000000e+005;
 b5[ 80]= 3.8348310e-007;
eb5[ 81]= 8.1000000e+005;
 b5[ 81]= 3.7991800e-007;
eb5[ 82]= 8.2000000e+005;
 b5[ 82]= 3.7701210e-007;
eb5[ 83]= 8.3000000e+005;
 b5[ 83]= 3.6524170e-007;
eb5[ 84]= 8.4000000e+005;
 b5[ 84]= 3.4440670e-007;
eb5[ 85]= 8.5000000e+005;
 b5[ 85]= 3.5400550e-007;
eb5[ 86]= 8.6000000e+005;
 b5[ 86]= 3.0449840e-007;
eb5[ 87]= 8.7000000e+005;
```

```
 b5[ 87]= 3.0935720e-007;
eb5[ 88]= 8.8000000e+005;
 b5[ 88]= 3.3335480e-007;
eb5[ 89]= 8.9000000e+005;
 b5[ 89]= 3.4611990e-007;
eb5[ 90]= 9.0000000e+005;
 b5[ 90]= 3.1740920e-007;
eb5[ 91]= 9.1000000e+005;
 b5[ 91]= 3.5508430e-007;
eb5[ 92]= 9.2000000e+005;
 b5[ 92]= 3.2450070e-007;
eb5[ 93]= 9.3000000e+005;
 b5[ 93]= 3.0712670e-007;
eb5[ 94]= 9.4000000e+005;
 b5[ 94]= 3.1996430e-007;
eb5[ 95]= 9.5000000e+005;
 b5[ 95]= 3.0523250e-007;
eb5[ 96]= 9.6000000e+005;
 b5[ 96]= 2.9617440e-007;
eb5[ 97]= 9.7000000e+005;
 b5[ 97]= 3.1102350e-007;
eb5[ 98]= 9.8000000e+005;
 b5[ 98]= 2.8319730e-007;
eb5[ 99]= 9.9000000e+005;
 b5[ 99]= 2.7478190e-007;
eb5[100]= 1.0000000e+006;
 b5[100]= 3.0348540e-007;
eb5[101]= 1.0100000e+006;
 b5[101]= 2.9068510e-007;
eb5[102]= 1.0200000e+006;
 b5[102]= 2.6109140e-007;
eb5[103]= 1.0300000e+006;
 b5[103]= 2.9471180e-007;
eb5[104]= 1.0400000e+006;
 b5[104]= 2.4376290e-007;
eb5[105]= 1.0500000e+006;
 b5[105]= 1.9373490e-007;
eb5[106]= 1.0600000e+006;
 b5[106]= 2.2157590e-007;
eb5[107]= 1.0700000e+006;
 b5[107]= 2.2458080e-007;
eb5[108]= 1.0800000e+006;
 b5[108]= 2.1948540e-007;
eb5[109]= 1.0900000e+006;
 b5[109]= 1.9338290e-007;
eb5[110]= 1.1000000e+006;
 b5[110]= 1.7324900e-007;
eb5[111]= 1.1100000e+006;
 b5[111]= 1.9141560e-007;
eb5[112]= 1.1200000e+006;
 b5[112]= 1.9084880e-007;
eb5[113]= 1.1300000e+006;
 b5[113]= 1.7714940e-007;
eb5[114]= 1.1400000e+006;
 b5[114]= 1.9894090e-007;
eb5[115]= 1.1500000e+006;
 b5[115]= 1.9484530e-007;
eb5[116]= 1.1600000e+006;
 b5[116]= 1.9744090e-007;
eb5[117]= 1.1700000e+006;
 b5[117]= 1.7266380e-007;
eb5[118]= 1.1800000e+006;
 b5[118]= 1.9191740e-007;
eb5[119]= 1.1900000e+006;
 b5[119]= 1.7755650e-007;
eb5[120]= 1.2000000e+006;
 b5[120]= 1.9609060e-007;
eb5[121]= 1.2100000e+006;
 b5[121]= 1.7858860e-007;
eb5[122]= 1.2200000e+006;
 b5[122]= 1.4487980e-007;
eb5[123]= 1.2300000e+006;
 b5[123]= 1.6260160e-007;
eb5[124]= 1.2400000e+006;
```

```
 b5[124]= 1.6614110e-007;
eb5[125]= 1.2500000e+006;
 b5[125]= 1.5673920e-007;
eb5[126]= 1.2600000e+006;
 b5[126]= 1.4447470e-007;
eb5[127]= 1.2700000e+006;
 b5[127]= 1.4933960e-007;
eb5[128]= 1.2800000e+006;
 b5[128]= 1.3178980e-007;
eb5[129]= 1.2900000e+006;
 b5[129]= 1.3691660e-007;
eb5[130]= 1.3000000e+006;
 b5[130]= 1.3581030e-007;
eb5[131]= 1.3100000e+006;
 b5[131]= 1.2568750e-007;
eb5[132]= 1.3200000e+006;
 b5[132]= 1.2314830e-007;
eb5[133]= 1.3300000e+006;
 b5[133]= 1.3778800e-007;
eb5[134]= 1.3400000e+006;
 b5[134]= 1.1410150e-007;
eb5[135]= 1.3500000e+006;
 b5[135]= 1.4580640e-007;
eb5[136]= 1.3600000e+006;
 b5[136]= 1.1273840e-007;
eb5[137]= 1.3700000e+006;
 b5[137]= 1.1868950e-007;
eb5[138]= 1.3800000e+006;
 b5[138]= 1.4730060e-007;
eb5[139]= 1.3900000e+006;
 b5[139]= 1.1408250e-007;
eb5[140]= 1.4000000e+006;
 b5[140]= 1.2850720e-007;
eb5[141]= 1.4100000e+006;
 b5[141]= 1.1415640e-007;
eb5[142]= 1.4200000e+006;
 b5[142]= 1.2522290e-007;
eb5[143]= 1.4300000e+006;
 b5[143]= 1.0663670e-007;
eb5[144]= 1.4400000e+006;
 b5[144]= 1.0709420e-007;
eb5[145]= 1.4500000e+006;
 b5[145]= 1.0660530e-007;
eb5[146]= 1.4600000e+006;
 b5[146]= 1.1692490e-007;
eb5[147]= 1.4700000e+006;
 b5[147]= 1.0999860e-007;
eb5[148]= 1.4800000e+006;
 b5[148]= 8.3804320e-008;
eb5[149]= 1.4900000e+006;
 b5[149]= 8.7308190e-008;
eb5[150]= 1.5000000e+006;
 b5[150]= 7.4367650e-008;
eb5[151]= 1.5100000e+006;
 b5[151]= 8.5802410e-008;
eb5[152]= 1.5200000e+006;
 b5[152]= 9.8454500e-008;
eb5[153]= 1.5300000e+006;
 b5[153]= 9.0302420e-008;
eb5[154]= 1.5400000e+006;
 b5[154]= 7.5359700e-008;
eb5[155]= 1.5500000e+006;
 b5[155]= 9.0747320e-008;
eb5[156]= 1.5600000e+006;
 b5[156]= 8.1598530e-008;
eb5[157]= 1.5700000e+006;
 b5[157]= 7.4278760e-008;
eb5[158]= 1.5800000e+006;
 b5[158]= 6.0280260e-008;
eb5[159]= 1.5900000e+006;
 b5[159]= 8.0973300e-008;
eb5[160]= 1.6000000e+006;
 b5[160]= 7.4156500e-008;
eb5[161]= 1.6100000e+006;
```

```
 b5[161]= 6.3035420e-008;
eb5[162]= 1.6200000e+006;
 b5[162]= 7.1802860e-008;
eb5[163]= 1.6300000e+006;
 b5[163]= 6.7313780e-008;
eb5[164]= 1.6400000e+006;
 b5[164]= 8.8871680e-008;
eb5[165]= 1.6500000e+006;
 b5[165]= 6.7937370e-008;
eb5[166]= 1.6600000e+006;
 b5[166]= 6.9253290e-008;
eb5[167]= 1.6700000e+006;
 b5[167]= 6.8377160e-008;
eb5[168]= 1.6800000e+006;
 b5[168]= 6.1872150e-008;
eb5[169]= 1.6900000e+006;
 b5[169]= 6.6644880e-008;
eb5[170]= 1.7000000e+006;
 b5[170]= 5.0038890e-008;
eb5[171]= 1.7100000e+006;
 b5[171]= 6.2772550e-008;
eb5[172]= 1.7200000e+006;
 b5[172]= 4.6882020e-008;
eb5[173]= 1.7300000e+006;
 b5[173]= 4.9922200e-008;
eb5[174]= 1.7400000e+006;
 b5[174]= 5.7472040e-008;
eb5[175]= 1.7500000e+006;
 b5[175]= 4.7905710e-008;
eb5[176]= 1.7600000e+006;
 b5[176]= 5.0501260e-008;
eb5[177]= 1.7700000e+006;
 b5[177]= 4.3333640e-008;
eb5[178]= 1.7800000e+006;
 b5[178]= 4.3356870e-008;
eb5[179]= 1.7900000e+006;
 b5[179]= 3.8914650e-008;
eb5[180]= 1.8000000e+006;
 b5[180]= 3.8565980e-008;
eb5[181]= 1.8100000e+006;
 b5[181]= 4.8949030e-008;
eb5[182]= 1.8200000e+006;
 b5[182]= 5.3352150e-008;
eb5[183]= 1.8300000e+006;
 b5[183]= 3.7029280e-008;
eb5[184]= 1.8400000e+006;
 b5[184]= 3.9365220e-008;
eb5[185]= 1.8500000e+006;
 b5[185]= 4.4136230e-008;
eb5[186]= 1.8600000e+006;
 b5[186]= 3.4238950e-008;
eb5[187]= 1.8700000e+006;
 b5[187]= 4.1874120e-008;
eb5[188]= 1.8800000e+006;
 b5[188]= 4.2203960e-008;
eb5[189]= 1.8900000e+006;
 b5[189]= 4.4504490e-008;
eb5[190]= 1.9000000e+006;
 b5[190]= 4.1358790e-008;
eb5[191]= 1.9100000e+006;
 b5[191]= 4.1689670e-008;
eb5[192]= 1.9200000e+006;
 b5[192]= 3.7712220e-008;
eb5[193]= 1.9300000e+006;
 b5[193]= 4.4652790e-008;
eb5[194]= 1.9400000e+006;
 b5[194]= 4.1650230e-008;
eb5[195]= 1.9500000e+006;
 b5[195]= 4.1058190e-008;
eb5[196]= 1.9600000e+006;
 b5[196]= 3.9260750e-008;
eb5[197]= 1.9700000e+006;
 b5[197]= 3.8678540e-008;
eb5[198]= 1.9800000e+006;
```

```
 b5[198]= 2.8623840e-008;
eb5[199]= 1.9900000e+006;
 b5[199]= 3.2343680e-008;
eb5[200]= 2.0000000e+006;
 b5[200]= 2.5729160e-008;
eb5[201]= 2.0100000e+006;
 b5[201]= 3.3342010e-008;
eb5[202]= 2.0200000e+006;
 b5[202]= 3.0990530e-008;
eb5[203]= 2.0300000e+006;
 b5[203]= 2.7843580e-008;
eb5[204]= 2.0400000e+006;
 b5[204]= 4.4473340e-008;
eb5[205]= 2.0500000e+006;
 b5[205]= 4.4015210e-008;
eb5[206]= 2.0600000e+006;
 b5[206]= 3.3620680e-008;
eb5[207]= 2.0700000e+006;
 b5[207]= 2.6555810e-008;
eb5[208]= 2.0800000e+006;
 b5[208]= 2.1158260e-008;
eb5[209]= 2.0900000e+006;
 b5[209]= 2.4468840e-008;
eb5[210]= 2.1000000e+006;
 b5[210]= 2.4061140e-008;
eb5[211]= 2.1100000e+006;
 b5[211]= 2.8299850e-008;
eb5[212]= 2.1200000e+006;
 b5[212]= 2.8877150e-008;
eb5[213]= 2.1300000e+006;
 b5[213]= 2.8772820e-008;
eb5[214]= 2.1400000e+006;
 b5[214]= 3.3092760e-008;
eb5[215]= 2.1500000e+006;
 b5[215]= 1.9809240e-008;
eb5[216]= 2.1600000e+006;
 b5[216]= 2.0904350e-008;
eb5[217]= 2.1700000e+006;
 b5[217]= 1.8982600e-008;
eb5[218]= 2.1800000e+006;
 b5[218]= 2.8254490e-008;
eb5[219]= 2.1900000e+006;
 b5[219]= 1.4566240e-008;
eb5[220]= 2.2000000e+006;
 b5[220]= 3.5506490e-008;
eb5[221]= 2.2100000e+006;
 b5[221]= 1.7423690e-008;
eb5[222]= 2.2200000e+006;
 b5[222]= 2.1014780e-008;
eb5[223]= 2.2300000e+006;
 b5[223]= 1.9336000e-008;
eb5[224]= 2.2400000e+006;
 b5[224]= 2.3779730e-008;
eb5[225]= 2.2500000e+006;
 b5[225]= 1.9083670e-008;
eb5[226]= 2.2600000e+006;
 b5[226]= 1.8567670e-008;
eb5[227]= 2.2700000e+006;
 b5[227]= 2.0190970e-008;
eb5[228]= 2.2800000e+006;
 b5[228]= 1.9166320e-008;
eb5[229]= 2.2900000e+006;
 b5[229]= 1.9912190e-008;
eb5[230]= 2.3000000e+006;
 b5[230]= 1.5047430e-008;
eb5[231]= 2.3100000e+006;
 b5[231]= 2.4916350e-008;
eb5[232]= 2.3200000e+006;
 b5[232]= 2.2318680e-008;
eb5[233]= 2.3300000e+006;
 b5[233]= 1.9470940e-008;
eb5[234]= 2.3400000e+006;
 b5[234]= 1.6033790e-008;
eb5[235]= 2.3500000e+006;
```

```
 b5[235]= 1.6411720e-008;
eb5[236]= 2.3600000e+006;
 b5[236]= 1.6851690e-008;
eb5[237]= 2.3700000e+006;
 b5[237]= 1.3833750e-008;
eb5[238]= 2.3800000e+006;
 b5[238]= 1.9414790e-008;
eb5[239]= 2.3900000e+006;
 b5[239]= 1.5756980e-008;
eb5[240]= 2.4000000e+006;
 b5[240]= 1.4368480e-008;
eb5[241]= 2.4100000e+006;
 b5[241]= 1.8675950e-008;
eb5[242]= 2.4200000e+006;
 b5[242]= 1.3524970e-008;
eb5[243]= 2.4300000e+006;
 b5[243]= 9.4762730e-009;
eb5[244]= 2.4400000e+006;
 b5[244]= 1.1692090e-008;
eb5[245]= 2.4500000e+006;
 b5[245]= 1.6229660e-008;
eb5[246]= 2.4600000e+006;
 b5[246]= 1.4042520e-008;
eb5[247]= 2.4700000e+006;
 b5[247]= 1.2962180e-008;
eb5[248]= 2.4800000e+006;
 b5[248]= 1.2915680e-008;
eb5[249]= 2.4900000e+006;
 b5[249]= 2.5897080e-008;
eb5[250]= 2.5000000e+006;
 b5[250]= 9.8188000e-009;
eb5[251]= 2.5100000e+006;
 b5[251]= 8.7313180e-009;
eb5[252]= 2.5200000e+006;
 b5[252]= 8.3760750e-009;
eb5[253]= 2.5300000e+006;
 b5[253]= 9.8627510e-009;
eb5[254]= 2.5400000e+006;
 b5[254]= 1.0125910e-008;
eb5[255]= 2.5500000e+006;
 b5[255]= 9.5293200e-009;
eb5[256]= 2.5600000e+006;
 b5[256]= 1.0911350e-008;
eb5[257]= 2.5700000e+006;
 b5[257]= 1.4880580e-008;
eb5[258]= 2.5800000e+006;
 b5[258]= 7.7456830e-009;
eb5[259]= 2.5900000e+006;
 b5[259]= 8.1500530e-009;
eb5[260]= 2.6000000e+006;
 b5[260]= 6.7074480e-009;
eb5[261]= 2.6100000e+006;
 b5[261]= 7.1119410e-009;
eb5[262]= 2.6200000e+006;
 b5[262]= 1.2752870e-008;
eb5[263]= 2.6300000e+006;
 b5[263]= 7.1781580e-009;
eb5[264]= 2.6400000e+006;
 b5[264]= 6.0820330e-009;
eb5[265]= 2.6500000e+006;
 b5[265]= 1.2042210e-008;
eb5[266]= 2.6600000e+006;
 b5[266]= 7.6749270e-009;
eb5[267]= 2.6700000e+006;
 b5[267]= 7.0867990e-009;
eb5[268]= 2.6800000e+006;
 b5[268]= 6.5523430e-009;
eb5[269]= 2.6900000e+006;
 b5[269]= 1.3606940e-008;
eb5[270]= 2.7000000e+006;
 b5[270]= 9.3394970e-009;
eb5[271]= 2.7100000e+006;
 b5[271]= 7.0801020e-009;
eb5[272]= 2.7200000e+006;
```

```
 b5[272]= 1.3691210e-008;
eb5[273]= 2.7300000e+006;
 b5[273]= 1.3143790e-008;
eb5[274]= 2.7400000e+006;
 b5[274]= 6.0847830e-009;
eb5[275]= 2.7500000e+006;
 b5[275]= 6.0114140e-009;
eb5[276]= 2.7600000e+006;
 b5[276]= 7.6359310e-009;
eb5[277]= 2.7700000e+006;
 b5[277]= 7.5779690e-009;
eb5[278]= 2.7800000e+006;
 b5[278]= 5.2772190e-009;
eb5[279]= 2.7900000e+006;
 b5[279]= 5.9941670e-009;
eb5[280]= 2.8000000e+006;
 b5[280]= 1.3011050e-008;
eb5[281]= 2.8100000e+006;
 b5[281]= 4.9034810e-009;
eb5[282]= 2.8200000e+006;
 b5[282]= 4.8451110e-009;
eb5[283]= 2.8300000e+006;
 b5[283]= 1.1607030e-008;
eb5[284]= 2.8400000e+006;
 b5[284]= 1.1321090e-008;
eb5[285]= 2.8500000e+006;
 b5[285]= 4.8599500e-009;
eb5[286]= 2.8600000e+006;
 b5[286]= 6.3676030e-009;
eb5[287]= 2.8700000e+006;
 b5[287]= 5.2337890e-009;
eb5[288]= 2.8800000e+006;
 b5[288]= 5.1606240e-009;
eb5[289]= 2.8900000e+006;
 b5[289]= 6.1485020e-009;
eb5[290]= 2.9000000e+006;
 b5[290]= 3.6771450e-009;
eb5[291]= 2.9100000e+006;
 b5[291]= 4.6738650e-009;
eb5[292]= 2.9200000e+006;
 b5[292]= 8.6679020e-009;
eb5[293]= 2.9300000e+006;
 b5[293]= 6.8772130e-009;
eb5[294]= 2.9400000e+006;
 b5[294]= 5.7210460e-009;
eb5[295]= 2.9500000e+006;
 b5[295]= 5.6408020e-009;
eb5[296]= 2.9600000e+006;
 b5[296]= 4.7513360e-009;
eb5[297]= 2.9700000e+006;
 b5[297]= 4.6744760e-009;
eb5[298]= 2.9800000e+006;
 b5[298]= 5.1397440e-009;
eb5[299]= 2.9900000e+006;
 b5[299]= 6.1586340e-009;
eb5[300]= 3.0000000e+006;
 b5[300]= 0.0000000e+000;
eb5[301]= 20.000000e+006;
 b5[301]= 0.0000000e+000;
eb6[  0]= 0.0000000e+000;
 b6[  0]= 2.3780630e-007;
eb6[  1]= 1.0000000e+004;
 b6[  1]= 2.8496800e-006;
eb6[  2]= 2.0000000e+004;
 b6[  2]= 1.1830990e-006;
eb6[  3]= 3.0000000e+004;
 b6[  3]= 8.0506760e-007;
eb6[  4]= 4.0000000e+004;
 b6[  4]= 1.1268910e-006;
eb6[  5]= 5.0000000e+004;
 b6[  5]= 9.4672110e-007;
eb6[  6]= 6.0000000e+004;
 b6[  6]= 1.0417670e-006;
eb6[  7]= 7.0000000e+004;
```

```
 b6[  7]= 1.4147030e-006;
eb6[  8]= 8.0000000e+004;
 b6[  8]= 1.4289610e-006;
eb6[  9]= 9.0000000e+004;
 b6[  9]= 1.4099600e-006;
eb6[ 10]= 1.0000000e+005;
 b6[ 10]= 1.3007170e-006;
eb6[ 11]= 1.1000000e+005;
 b6[ 11]= 1.2217920e-006;
eb6[ 12]= 1.2000000e+005;
 b6[ 12]= 1.2520870e-006;
eb6[ 13]= 1.3000000e+005;
 b6[ 13]= 1.2476070e-006;
eb6[ 14]= 1.4000000e+005;
 b6[ 14]= 1.2704880e-006;
eb6[ 15]= 1.5000000e+005;
 b6[ 15]= 1.3828360e-006;
eb6[ 16]= 1.6000000e+005;
 b6[ 16]= 1.3348340e-006;
eb6[ 17]= 1.7000000e+005;
 b6[ 17]= 1.3458880e-006;
eb6[ 18]= 1.8000000e+005;
 b6[ 18]= 1.4019070e-006;
eb6[ 19]= 1.9000000e+005;
 b6[ 19]= 1.3140090e-006;
eb6[ 20]= 2.0000000e+005;
 b6[ 20]= 1.3560930e-006;
eb6[ 21]= 2.1000000e+005;
 b6[ 21]= 1.3717330e-006;
eb6[ 22]= 2.2000000e+005;
 b6[ 22]= 1.3663250e-006;
eb6[ 23]= 2.3000000e+005;
 b6[ 23]= 1.2719740e-006;
eb6[ 24]= 2.4000000e+005;
 b6[ 24]= 1.3024090e-006;
eb6[ 25]= 2.5000000e+005;
 b6[ 25]= 1.4203390e-006;
eb6[ 26]= 2.6000000e+005;
 b6[ 26]= 1.3311400e-006;
eb6[ 27]= 2.7000000e+005;
 b6[ 27]= 1.2131640e-006;
eb6[ 28]= 2.8000000e+005;
 b6[ 28]= 1.1827850e-006;
eb6[ 29]= 2.9000000e+005;
 b6[ 29]= 1.1569020e-006;
eb6[ 30]= 3.0000000e+005;
 b6[ 30]= 1.1881230e-006;
eb6[ 31]= 3.1000000e+005;
 b6[ 31]= 1.3384470e-006;
eb6[ 32]= 3.2000000e+005;
 b6[ 32]= 1.2184240e-006;
eb6[ 33]= 3.3000000e+005;
 b6[ 33]= 1.2115860e-006;
eb6[ 34]= 3.4000000e+005;
 b6[ 34]= 1.1955370e-006;
eb6[ 35]= 3.5000000e+005;
 b6[ 35]= 1.1761880e-006;
eb6[ 36]= 3.6000000e+005;
 b6[ 36]= 1.1474010e-006;
eb6[ 37]= 3.7000000e+005;
 b6[ 37]= 1.0370830e-006;
eb6[ 38]= 3.8000000e+005;
 b6[ 38]= 9.5877990e-007;
eb6[ 39]= 3.9000000e+005;
 b6[ 39]= 8.7827650e-007;
eb6[ 40]= 4.0000000e+005;
 b6[ 40]= 8.7317890e-007;
eb6[ 41]= 4.1000000e+005;
 b6[ 41]= 9.0066260e-007;
eb6[ 42]= 4.2000000e+005;
 b6[ 42]= 9.9777660e-007;
eb6[ 43]= 4.3000000e+005;
 b6[ 43]= 1.0420640e-006;
eb6[ 44]= 4.4000000e+005;
```

```
 b6[ 44]= 9.5381240e-007;
eb6[ 45]= 4.5000000e+005;
 b6[ 45]= 9.8224630e-007;
eb6[ 46]= 4.6000000e+005;
 b6[ 46]= 9.2318690e-007;
eb6[ 47]= 4.7000000e+005;
 b6[ 47]= 8.7727190e-007;
eb6[ 48]= 4.8000000e+005;
 b6[ 48]= 8.4320520e-007;
eb6[ 49]= 4.9000000e+005;
 b6[ 49]= 8.4104990e-007;
eb6[ 50]= 5.0000000e+005;
 b6[ 50]= 9.1713620e-007;
eb6[ 51]= 5.1000000e+005;
 b6[ 51]= 8.4525320e-007;
eb6[ 52]= 5.2000000e+005;
 b6[ 52]= 8.1776320e-007;
eb6[ 53]= 5.3000000e+005;
 b6[ 53]= 8.9627840e-007;
eb6[ 54]= 5.4000000e+005;
 b6[ 54]= 9.3087590e-007;
eb6[ 55]= 5.5000000e+005;
 b6[ 55]= 8.5629880e-007;
eb6[ 56]= 5.6000000e+005;
 b6[ 56]= 8.1588500e-007;
eb6[ 57]= 5.7000000e+005;
 b6[ 57]= 8.8232040e-007;
eb6[ 58]= 5.8000000e+005;
 b6[ 58]= 8.9538200e-007;
eb6[ 59]= 5.9000000e+005;
 b6[ 59]= 7.7069720e-007;
eb6[ 60]= 6.0000000e+005;
 b6[ 60]= 7.5737450e-007;
eb6[ 61]= 6.1000000e+005;
 b6[ 61]= 6.9971900e-007;
eb6[ 62]= 6.2000000e+005;
 b6[ 62]= 6.4640250e-007;
eb6[ 63]= 6.3000000e+005;
 b6[ 63]= 6.9684160e-007;
eb6[ 64]= 6.4000000e+005;
 b6[ 64]= 6.1481630e-007;
eb6[ 65]= 6.5000000e+005;
 b6[ 65]= 6.4712740e-007;
eb6[ 66]= 6.6000000e+005;
 b6[ 66]= 6.0735250e-007;
eb6[ 67]= 6.7000000e+005;
 b6[ 67]= 6.0247610e-007;
eb6[ 68]= 6.8000000e+005;
 b6[ 68]= 6.3550150e-007;
eb6[ 69]= 6.9000000e+005;
 b6[ 69]= 6.3365340e-007;
eb6[ 70]= 7.0000000e+005;
 b6[ 70]= 5.7075580e-007;
eb6[ 71]= 7.1000000e+005;
 b6[ 71]= 5.2998900e-007;
eb6[ 72]= 7.2000000e+005;
 b6[ 72]= 5.5644760e-007;
eb6[ 73]= 7.3000000e+005;
 b6[ 73]= 5.6515030e-007;
eb6[ 74]= 7.4000000e+005;
 b6[ 74]= 5.6156350e-007;
eb6[ 75]= 7.5000000e+005;
 b6[ 75]= 5.0446040e-007;
eb6[ 76]= 7.6000000e+005;
 b6[ 76]= 4.7424540e-007;
eb6[ 77]= 7.7000000e+005;
 b6[ 77]= 4.6065410e-007;
eb6[ 78]= 7.8000000e+005;
 b6[ 78]= 4.6049320e-007;
eb6[ 79]= 7.9000000e+005;
 b6[ 79]= 5.1369360e-007;
eb6[ 80]= 8.0000000e+005;
 b6[ 80]= 4.9721550e-007;
eb6[ 81]= 8.1000000e+005;
```

```
 b6[ 81]= 4.7423070e-007;
eb6[ 82]= 8.2000000e+005;
 b6[ 82]= 4.4868550e-007;
eb6[ 83]= 8.3000000e+005;
 b6[ 83]= 4.0629120e-007;
eb6[ 84]= 8.4000000e+005;
 b6[ 84]= 3.4950650e-007;
eb6[ 85]= 8.5000000e+005;
 b6[ 85]= 3.4239860e-007;
eb6[ 86]= 8.6000000e+005;
 b6[ 86]= 3.1210140e-007;
eb6[ 87]= 8.7000000e+005;
 b6[ 87]= 3.0787010e-007;
eb6[ 88]= 8.8000000e+005;
 b6[ 88]= 3.0938310e-007;
eb6[ 89]= 8.9000000e+005;
 b6[ 89]= 3.1503110e-007;
eb6[ 90]= 9.0000000e+005;
 b6[ 90]= 3.1543940e-007;
eb6[ 91]= 9.1000000e+005;
 b6[ 91]= 3.1145230e-007;
eb6[ 92]= 9.2000000e+005;
 b6[ 92]= 3.0580060e-007;
eb6[ 93]= 9.3000000e+005;
 b6[ 93]= 2.9135220e-007;
eb6[ 94]= 9.4000000e+005;
 b6[ 94]= 3.5321170e-007;
eb6[ 95]= 9.5000000e+005;
 b6[ 95]= 3.0578630e-007;
eb6[ 96]= 9.6000000e+005;
 b6[ 96]= 2.7837640e-007;
eb6[ 97]= 9.7000000e+005;
 b6[ 97]= 2.9890750e-007;
eb6[ 98]= 9.8000000e+005;
 b6[ 98]= 2.5262100e-007;
eb6[ 99]= 9.9000000e+005;
 b6[ 99]= 2.7075260e-007;
eb6[100]= 1.0000000e+006;
 b6[100]= 2.7912580e-007;
eb6[101]= 1.0100000e+006;
 b6[101]= 2.5769370e-007;
eb6[102]= 1.0200000e+006;
 b6[102]= 2.7068450e-007;
eb6[103]= 1.0300000e+006;
 b6[103]= 2.7213460e-007;
eb6[104]= 1.0400000e+006;
 b6[104]= 2.3691040e-007;
eb6[105]= 1.0500000e+006;
 b6[105]= 2.2456880e-007;
eb6[106]= 1.0600000e+006;
 b6[106]= 2.1965630e-007;
eb6[107]= 1.0700000e+006;
 b6[107]= 2.1546370e-007;
eb6[108]= 1.0800000e+006;
 b6[108]= 2.0322860e-007;
eb6[109]= 1.0900000e+006;
 b6[109]= 1.8394990e-007;
eb6[110]= 1.1000000e+006;
 b6[110]= 1.7744720e-007;
eb6[111]= 1.1100000e+006;
 b6[111]= 1.8962150e-007;
eb6[112]= 1.1200000e+006;
 b6[112]= 1.8543160e-007;
eb6[113]= 1.1300000e+006;
 b6[113]= 1.7597210e-007;
eb6[114]= 1.1400000e+006;
 b6[114]= 1.9073260e-007;
eb6[115]= 1.1500000e+006;
 b6[115]= 1.8916380e-007;
eb6[116]= 1.1600000e+006;
 b6[116]= 1.8888100e-007;
eb6[117]= 1.1700000e+006;
 b6[117]= 1.7157010e-007;
eb6[118]= 1.1800000e+006;
```

```
 b6[118]= 1.7435670e-007;
eb6[119]= 1.1900000e+006;
 b6[119]= 1.6263980e-007;
eb6[120]= 1.2000000e+006;
 b6[120]= 1.6227240e-007;
eb6[121]= 1.2100000e+006;
 b6[121]= 1.7150110e-007;
eb6[122]= 1.2200000e+006;
 b6[122]= 1.4854300e-007;
eb6[123]= 1.2300000e+006;
 b6[123]= 1.6716780e-007;
eb6[124]= 1.2400000e+006;
 b6[124]= 1.5772920e-007;
eb6[125]= 1.2500000e+006;
 b6[125]= 1.4949100e-007;
eb6[126]= 1.2600000e+006;
 b6[126]= 1.4375070e-007;
eb6[127]= 1.2700000e+006;
 b6[127]= 1.5301080e-007;
eb6[128]= 1.2800000e+006;
 b6[128]= 1.4529810e-007;
eb6[129]= 1.2900000e+006;
 b6[129]= 1.5861290e-007;
eb6[130]= 1.3000000e+006;
 b6[130]= 1.4389230e-007;
eb6[131]= 1.3100000e+006;
 b6[131]= 1.3800780e-007;
eb6[132]= 1.3200000e+006;
 b6[132]= 1.3123880e-007;
eb6[133]= 1.3300000e+006;
 b6[133]= 1.2538290e-007;
eb6[134]= 1.3400000e+006;
 b6[134]= 1.2296450e-007;
eb6[135]= 1.3500000e+006;
 b6[135]= 1.3234780e-007;
eb6[136]= 1.3600000e+006;
 b6[136]= 1.1697690e-007;
eb6[137]= 1.3700000e+006;
 b6[137]= 1.1389210e-007;
eb6[138]= 1.3800000e+006;
 b6[138]= 1.2009980e-007;
eb6[139]= 1.3900000e+006;
 b6[139]= 1.1906520e-007;
eb6[140]= 1.4000000e+006;
 b6[140]= 1.1944240e-007;
eb6[141]= 1.4100000e+006;
 b6[141]= 1.1550720e-007;
eb6[142]= 1.4200000e+006;
 b6[142]= 1.1596430e-007;
eb6[143]= 1.4300000e+006;
 b6[143]= 1.1463150e-007;
eb6[144]= 1.4400000e+006;
 b6[144]= 9.9606620e-008;
eb6[145]= 1.4500000e+006;
 b6[145]= 9.1859770e-008;
eb6[146]= 1.4600000e+006;
 b6[146]= 9.8858930e-008;
eb6[147]= 1.4700000e+006;
 b6[147]= 1.2381720e-007;
eb6[148]= 1.4800000e+006;
 b6[148]= 9.5616570e-008;
eb6[149]= 1.4900000e+006;
 b6[149]= 1.0034940e-007;
eb6[150]= 1.5000000e+006;
 b6[150]= 9.2810550e-008;
eb6[151]= 1.5100000e+006;
 b6[151]= 1.0408650e-007;
eb6[152]= 1.5200000e+006;
 b6[152]= 1.1778210e-007;
eb6[153]= 1.5300000e+006;
 b6[153]= 1.0077300e-007;
eb6[154]= 1.5400000e+006;
 b6[154]= 8.1171870e-008;
eb6[155]= 1.5500000e+006;
```

```
 b6[155]= 9.8279020e-008;
eb6[156]= 1.5600000e+006;
 b6[156]= 7.6449370e-008;
eb6[157]= 1.5700000e+006;
 b6[157]= 8.5988370e-008;
eb6[158]= 1.5800000e+006;
 b6[158]= 8.5273420e-008;
eb6[159]= 1.5900000e+006;
 b6[159]= 8.3842320e-008;
eb6[160]= 1.6000000e+006;
 b6[160]= 7.7267180e-008;
eb6[161]= 1.6100000e+006;
 b6[161]= 8.3362310e-008;
eb6[162]= 1.6200000e+006;
 b6[162]= 8.1097320e-008;
eb6[163]= 1.6300000e+006;
 b6[163]= 8.3207050e-008;
eb6[164]= 1.6400000e+006;
 b6[164]= 8.3781340e-008;
eb6[165]= 1.6500000e+006;
 b6[165]= 7.8360060e-008;
eb6[166]= 1.6600000e+006;
 b6[166]= 7.6895970e-008;
eb6[167]= 1.6700000e+006;
 b6[167]= 7.4219650e-008;
eb6[168]= 1.6800000e+006;
 b6[168]= 7.8700020e-008;
eb6[169]= 1.6900000e+006;
 b6[169]= 7.9529130e-008;
eb6[170]= 1.7000000e+006;
 b6[170]= 6.3851140e-008;
eb6[171]= 1.7100000e+006;
 b6[171]= 6.1177820e-008;
eb6[172]= 1.7200000e+006;
 b6[172]= 5.9755980e-008;
eb6[173]= 1.7300000e+006;
 b6[173]= 5.8977870e-008;
eb6[174]= 1.7400000e+006;
 b6[174]= 6.2864480e-008;
eb6[175]= 1.7500000e+006;
 b6[175]= 5.3695350e-008;
eb6[176]= 1.7600000e+006;
 b6[176]= 5.6164020e-008;
eb6[177]= 1.7700000e+006;
 b6[177]= 5.9618120e-008;
eb6[178]= 1.7800000e+006;
 b6[178]= 5.6373470e-008;
eb6[179]= 1.7900000e+006;
 b6[179]= 5.8108290e-008;
eb6[180]= 1.8000000e+006;
 b6[180]= 5.8961170e-008;
eb6[181]= 1.8100000e+006;
 b6[181]= 5.8589830e-008;
eb6[182]= 1.8200000e+006;
 b6[182]= 5.8723890e-008;
eb6[183]= 1.8300000e+006;
 b6[183]= 4.9240470e-008;
eb6[184]= 1.8400000e+006;
 b6[184]= 6.0291680e-008;
eb6[185]= 1.8500000e+006;
 b6[185]= 6.4181500e-008;
eb6[186]= 1.8600000e+006;
 b6[186]= 5.5478260e-008;
eb6[187]= 1.8700000e+006;
 b6[187]= 5.4407700e-008;
eb6[188]= 1.8800000e+006;
 b6[188]= 5.5919210e-008;
eb6[189]= 1.8900000e+006;
 b6[189]= 5.4479820e-008;
eb6[190]= 1.9000000e+006;
 b6[190]= 5.6089850e-008;
eb6[191]= 1.9100000e+006;
 b6[191]= 5.3554650e-008;
eb6[192]= 1.9200000e+006;
```

```
 b6[192]= 4.2152040e-008;
eb6[193]= 1.9300000e+006;
 b6[193]= 5.5706320e-008;
eb6[194]= 1.9400000e+006;
 b6[194]= 4.9790060e-008;
eb6[195]= 1.9500000e+006;
 b6[195]= 4.4804550e-008;
eb6[196]= 1.9600000e+006;
 b6[196]= 4.4788730e-008;
eb6[197]= 1.9700000e+006;
 b6[197]= 5.6858400e-008;
eb6[198]= 1.9800000e+006;
 b6[198]= 4.1539580e-008;
eb6[199]= 1.9900000e+006;
 b6[199]= 4.1134810e-008;
eb6[200]= 2.0000000e+006;
 b6[200]= 3.8012780e-008;
eb6[201]= 2.0100000e+006;
 b6[201]= 4.2472720e-008;
eb6[202]= 2.0200000e+006;
 b6[202]= 3.5638140e-008;
eb6[203]= 2.0300000e+006;
 b6[203]= 3.4932430e-008;
eb6[204]= 2.0400000e+006;
 b6[204]= 4.3382970e-008;
eb6[205]= 2.0500000e+006;
 b6[205]= 4.2838490e-008;
eb6[206]= 2.0600000e+006;
 b6[206]= 4.8832590e-008;
eb6[207]= 2.0700000e+006;
 b6[207]= 4.0855650e-008;
eb6[208]= 2.0800000e+006;
 b6[208]= 3.5829920e-008;
eb6[209]= 2.0900000e+006;
 b6[209]= 3.6067620e-008;
eb6[210]= 2.1000000e+006;
 b6[210]= 3.6252500e-008;
eb6[211]= 2.1100000e+006;
 b6[211]= 3.8767440e-008;
eb6[212]= 2.1200000e+006;
 b6[212]= 4.0318690e-008;
eb6[213]= 2.1300000e+006;
 b6[213]= 3.9840820e-008;
eb6[214]= 2.1400000e+006;
 b6[214]= 3.5242140e-008;
eb6[215]= 2.1500000e+006;
 b6[215]= 3.4319990e-008;
eb6[216]= 2.1600000e+006;
 b6[216]= 3.2023560e-008;
eb6[217]= 2.1700000e+006;
 b6[217]= 3.0917390e-008;
eb6[218]= 2.1800000e+006;
 b6[218]= 3.2056480e-008;
eb6[219]= 2.1900000e+006;
 b6[219]= 2.6609580e-008;
eb6[220]= 2.2000000e+006;
 b6[220]= 4.0096650e-008;
eb6[221]= 2.2100000e+006;
 b6[221]= 3.1332020e-008;
eb6[222]= 2.2200000e+006;
 b6[222]= 3.1387970e-008;
eb6[223]= 2.2300000e+006;
 b6[223]= 3.1244830e-008;
eb6[224]= 2.2400000e+006;
 b6[224]= 3.5702790e-008;
eb6[225]= 2.2500000e+006;
 b6[225]= 2.6444670e-008;
eb6[226]= 2.2600000e+006;
 b6[226]= 2.7185740e-008;
eb6[227]= 2.2700000e+006;
 b6[227]= 2.8332390e-008;
eb6[228]= 2.2800000e+006;
 b6[228]= 2.8013800e-008;
eb6[229]= 2.2900000e+006;
```

```
 b6[229]= 2.6331640e-008;
eb6[230]= 2.3000000e+006;
 b6[230]= 2.5096200e-008;
eb6[231]= 2.3100000e+006;
 b6[231]= 3.5171310e-008;
eb6[232]= 2.3200000e+006;
 b6[232]= 2.7547030e-008;
eb6[233]= 2.3300000e+006;
 b6[233]= 2.7308630e-008;
eb6[234]= 2.3400000e+006;
 b6[234]= 2.8944000e-008;
eb6[235]= 2.3500000e+006;
 b6[235]= 2.5887320e-008;
eb6[236]= 2.3600000e+006;
 b6[236]= 2.7283320e-008;
eb6[237]= 2.3700000e+006;
 b6[237]= 2.3173720e-008;
eb6[238]= 2.3800000e+006;
 b6[238]= 2.3809660e-008;
eb6[239]= 2.3900000e+006;
 b6[239]= 2.2476600e-008;
eb6[240]= 2.4000000e+006;
 b6[240]= 2.3116510e-008;
eb6[241]= 2.4100000e+006;
 b6[241]= 3.1081040e-008;
eb6[242]= 2.4200000e+006;
 b6[242]= 2.7616570e-008;
eb6[243]= 2.4300000e+006;
 b6[243]= 2.0298260e-008;
eb6[244]= 2.4400000e+006;
 b6[244]= 2.3710130e-008;
eb6[245]= 2.4500000e+006;
 b6[245]= 2.3148800e-008;
eb6[246]= 2.4600000e+006;
 b6[246]= 2.3821650e-008;
eb6[247]= 2.4700000e+006;
 b6[247]= 2.0725860e-008;
eb6[248]= 2.4800000e+006;
 b6[248]= 2.4128720e-008;
eb6[249]= 2.4900000e+006;
 b6[249]= 2.9082360e-008;
eb6[250]= 2.5000000e+006;
 b6[250]= 2.2424150e-008;
eb6[251]= 2.5100000e+006;
 b6[251]= 1.9312410e-008;
eb6[252]= 2.5200000e+006;
 b6[252]= 1.9836420e-008;
eb6[253]= 2.5300000e+006;
 b6[253]= 1.9631280e-008;
eb6[254]= 2.5400000e+006;
 b6[254]= 2.5295910e-008;
eb6[255]= 2.5500000e+006;
 b6[255]= 2.3637040e-008;
eb6[256]= 2.5600000e+006;
 b6[256]= 2.7870690e-008;
eb6[257]= 2.5700000e+006;
 b6[257]= 2.0017430e-008;
eb6[258]= 2.5800000e+006;
 b6[258]= 1.8654500e-008;
eb6[259]= 2.5900000e+006;
 b6[259]= 1.9963000e-008;
eb6[260]= 2.6000000e+006;
 b6[260]= 1.6058120e-008;
eb6[261]= 2.6100000e+006;
 b6[261]= 1.7362550e-008;
eb6[262]= 2.6200000e+006;
 b6[262]= 1.9118010e-008;
eb6[263]= 2.6300000e+006;
 b6[263]= 1.7753960e-008;
eb6[264]= 2.6400000e+006;
 b6[264]= 1.4580760e-008;
eb6[265]= 2.6500000e+006;
 b6[265]= 1.7113120e-008;
eb6[266]= 2.6600000e+006;
```

```
 b6[266]= 1.9504700e-008;
eb6[267]= 2.6700000e+006;
 b6[267]= 1.7836960e-008;
eb6[268]= 2.6800000e+006;
 b6[268]= 1.6910460e-008;
eb6[269]= 2.6900000e+006;
 b6[269]= 2.2530180e-008;
eb6[270]= 2.7000000e+006;
 b6[270]= 2.0394310e-008;
eb6[271]= 2.7100000e+006;
 b6[271]= 1.8721720e-008;
eb6[272]= 2.7200000e+006;
 b6[272]= 2.2870580e-008;
eb6[273]= 2.7300000e+006;
 b6[273]= 2.1213420e-008;
eb6[274]= 2.7400000e+006;
 b6[274]= 1.5977780e-008;
eb6[275]= 2.7500000e+006;
 b6[275]= 1.5830440e-008;
eb6[276]= 2.7600000e+006;
 b6[276]= 1.5678120e-008;
eb6[277]= 2.7700000e+006;
 b6[277]= 1.5540260e-008;
eb6[278]= 2.7800000e+006;
 b6[278]= 1.3844080e-008;
eb6[279]= 2.7900000e+006;
 b6[279]= 1.0591750e-008;
eb6[280]= 2.8000000e+006;
 b6[280]= 1.5616620e-008;
eb6[281]= 2.8100000e+006;
 b6[281]= 1.2643610e-008;
eb6[282]= 2.8200000e+006;
 b6[282]= 1.0930370e-008;
eb6[283]= 2.8300000e+006;
 b6[283]= 1.5187200e-008;
eb6[284]= 2.8400000e+006;
 b6[284]= 1.4300160e-008;
eb6[285]= 2.8500000e+006;
 b6[285]= 1.1312570e-008;
eb6[286]= 2.8600000e+006;
 b6[286]= 1.5910810e-008;
eb6[287]= 2.8700000e+006;
 b6[287]= 1.2635400e-008;
eb6[288]= 2.8800000e+006;
 b6[288]= 1.2518130e-008;
eb6[289]= 2.8900000e+006;
 b6[289]= 1.5559340e-008;
eb6[290]= 2.9000000e+006;
 b6[290]= 8.2936920e-009;
eb6[291]= 2.9100000e+006;
 b6[291]= 1.1357610e-008;
eb6[292]= 2.9200000e+006;
 b6[292]= 1.5097300e-008;
eb6[293]= 2.9300000e+006;
 b6[293]= 1.4213100e-008;
eb6[294]= 2.9400000e+006;
 b6[294]= 1.0887630e-008;
eb6[295]= 2.9500000e+006;
 b6[295]= 1.0765470e-008;
eb6[296]= 2.9600000e+006;
 b6[296]= 8.2275770e-009;
eb6[297]= 2.9700000e+006;
 b6[297]= 8.1081220e-009;
eb6[298]= 2.9800000e+006;
 b6[298]= 9.6054320e-009;
eb6[299]= 2.9900000e+006;
 b6[299]= 1.2748300e-008;
eb6[300]= 3.0000000e+006;
 b6[300]= 0.0000000e+000;
eb6[301]= 20.000000e+006;
 b6[301]= 0.0000000e+000;

for (g=0;g<G;g++)
    {
```

```
    v[g]=e[g]-((e[g]-e[g+1])/2);
    v[g]=2.2e5*sqrt(v[g]/0.025);
// printf("v[%d]=%7.2e %7.2e\n",g,v[g], sqrt(1));
    }
for (g=0;g<G;g++)
    {
    v[g]=1/v[g];
    }
datar=malloc(G*sizeof(double));
dataa=malloc(G*sizeof(double));
datah=malloc(G*sizeof(double));
if ((fr=fopen("edtoutr","r"))==NULL) exit(1);
if ((fa=fopen("edtouta","r"))==NULL) exit(1);
for (i=0;i<SKIP;i++)
    {
    fgets(bufr,256, fr);
    fgets(bufa,256, fa);
    }
while (!feof(fr))
   {
   for (i=0;i<LINES-1;i++)
       {
       fgets(bufr,256,fr);
       fgets(bufa,256,fa);
       sscanf(bufr,"%lf %lf %lf %lf %lf %lf",&xr[0],&xr[1],&xr[2],&xr[3],&xr[4],&xr[5]);
       sscanf(bufa,"%lf %lf %lf %lf %lf %lf",&xa[0],&xa[1],&xa[2],&xa[3],&xa[4],&xa[5]);
       for (g=0;g<6;g++)
           {
           datar[g+(i*6)]=xr[g];
           dataa[g+(i*6)]=xa[g];
           }
       }
   fgets(bufr,256, fr);
   fgets(bufa,256, fa);
   sscanf(bufr,"%lf %lf %lf %lf %lf %lf",&xr[0],&xr[1],&xr[2],&xr[3],&xr[4],&xr[5]);
   sscanf(bufa,"%lf %lf %lf %lf %lf %lf",&xa[0],&xa[1],&xa[2],&xa[3],&xa[4],&xa[5]);
   if (!feof(fr))
      {
      for (g=0;g<ODDG;g++)
          {
          datar[g+(i*6)]=xr[g];
          dataa[g+(i*6)]=xa[g];
          }
      for (g=0;g<G;g++)
          {
          datah[g]=dataa[g];
          }
      h   =0;
      hden=0;
      for (g=0;g<G;g++)
          {
// uncomment this line if partisn is version 2 (do not uncomment if version 5)
//        dataa[g]=datah[G-1-g];
          he      =0.5*(e[g]+e[g+1]);
          betag   =calculate_betag(he);
          h       =h+betag*datar[g];
          hden    =hden+datar[g];
          }
      for (i=0;i<6;i++)
          {
          nor[i]=0;
          hnum  =0;
          for (g=0;g<G;g++)
              {
              he    =0.5*(e[g]+e[g+1]);
              chiig =calculate_chiig(i+1,g,he);
// adjoint calculation
              hnum  =hnum+h*dataa[g]*chiig*wbdi[i];
// no adjoint calculation
//            hnum  =hnum+h*1.0*chiig*wbdi[i];
              nor[i]=nor[i]+chiig;
              }
          num=num+hnum/nor[i];
          }
```

```
      for (g=0;g<G;g++)
          {
// adjoint calculation
          den=den+hden*dataa[g]*chi[g];
// no adjoint calculation
//        den=den+hden*1.0*chi[g];
          }
      }
   }
fclose(fr);
fclose(fa);
free(datar);
free(dataa);
free(datah);
printf("The delayed neutron fraction time is %7.3e\n",num/den);
}

double calculate_chiig(unsigned long int i,unsigned long int g,double energy)
{
unsigned long int gb1=185,gb2=220,gb3=301,gb4=301,gb5=301,gb6=301;
double h;

if (i==1)
   {
   while (eb1[gb1]>energy)
         {
         gb1--;
         }
   h=b1[gb1]*(e[g]-e[g+1])/(eb1[gb1+1]-eb1[gb1]);
   }
if (i==2)
   {
   while (eb2[gb2]>energy)
         {
         gb2--;
         }
   h=b2[gb2]*(e[g]-e[g+1])/(eb2[gb2+1]-eb2[gb2]);
   }
if (i==3)
   {
   while (eb3[gb3]>energy)
         {
         gb3--;
         }
   h=b3[gb3]*(e[g]-e[g+1])/(eb3[gb3+1]-eb3[gb3]);
   }
if (i==4)
   {
   while (eb4[gb4]>energy)
         {
         gb4--;
         }
   h=b4[gb4]*(e[g]-e[g+1])/(eb4[gb4+1]-eb4[gb4]);
   }
if (i==5)
   {
   while (eb5[gb5]>energy)
         {
         gb5--;
         }
   h=b5[gb5]*(e[g]-e[g+1])/(eb5[gb5+1]-eb5[gb5]);
   }
if (i==6)
   {
   while (eb6[gb6]>energy)
         {
         gb6--;
         }
   h=b6[gb6]*(e[g]-e[g+1])/(eb6[gb6+1]-eb6[gb6]);
   }
// printf("i=%d energy=%f gb1=%d b1=%e\n",i,energy,gb1,b1[gb1]);
return(h);
}
```

```
double calculate_betag(double energy)
{
unsigned long int gb=78,gbd=5;

while (eb[gb]>energy)
      {
      gb--;
// printf("gb=%d energy=%f\n",gb,energy);
      }
while (ebd[gbd]>energy)
      {
      gbd--;
// printf("gbd=%d energy=%f\n",gbd,energy);
      }
// printf("energy=%f gb=%d b=%e gbd=%d bd=%e\n",energy,gb,b[gb+1],gbd,bd[gbd+1]);
return(bd[gbd+1]/b[gb+1]);
}
```

Appendix F: Source Code for the Calculation of the Prompt Neutron Generation Time

To code can be compiled on Linux systems by the command:

```
gcc –o pgtadj pgtadj.c
```

The code can be executed by the command:

```
./pgtadj
```

The code expects three PARTISN ETDOUT files in the current directory:

1) `edtouta` containing the adjoint flux
2) `edtoutr` containing the fission rate integrated over volume
3) `edtoutf` containing the forward flux integrated over volume

The code works for 69 energy groups calculations.

```
// Author: Alberto Talamo
// Email : alby@anl.gov
// Program written to calculate Yalina Thermal prompt neutron generation time

#define LINES 12
#define G     69
#define ODDG  3
#define SKIP  36

// LINES include both full lines and half-full lines
// ODDG  includes ONLY group entries NOT the total (last value)

// include standard and math C libraries
#include <stdio.h>
#include <math.h>
#include <stdlib.h>
#include <errno.h>
#include <string.h>

int main(int argc,char *argv[])
{
FILE              *ff=NULL,*fr=NULL,*fa=NULL;
char               buff[256],bufr[256],bufa[256];
unsigned long int  i,g;
double            *dataf,*datar,*dataa,*datah,xf[6],xr[6],xa[6],e[70],v[69],chi[69],num=0,den=0,h=0;

e[ 0]=10.00E+6;
e[ 1]=6.065307000000000E+6;
e[ 2]=3.678794000000000E+6;
e[ 3]=2.231302000000000E+6;
e[ 4]=1.353353000000000E+6;
e[ 5]=8.208500000000000E+5;
e[ 6]=4.978707000000000E+5;
e[ 7]=3.019738000000000E+5;
e[ 8]=1.831564000000000E+5;
e[ 9]=1.110900000000000E+5;
e[10]=6.737947000000000E+4;
e[11]=4.086771000000000E+4;
e[12]=2.478752000000000E+4;
e[13]=1.503349000000000E+4;
e[14]=9.118820000000000E+3;
e[15]=5.530844000000000E+3;
e[16]=3.526622000000000E+3;
e[17]=2.248673000000000E+3;
e[18]=1.433817000000000E+3;
e[19]=9.142423000000000E+2;
e[20]=3.717032000000000E+2;
e[21]=1.486254000000000E+2;
e[22]=7.567357000000000E+1;
e[23]=4.825160000000000E+1;
e[24]=2.760773000000000E+1;
e[25]=1.592827000000000E+1;
e[26]=9.905554000000000E+0;
e[27]=4.000000000000000E+0;
e[28]=3.300000000000000E+0;
e[29]=2.600000000000000E+0;
e[30]=2.100000000000000E+0;
e[31]=1.500000000000000E+0;
e[32]=1.300000000000000E+0;
e[33]=1.150000000000000E+0;
e[34]=1.123000000000000E+0;
e[35]=1.097000000000000E+0;
e[36]=1.071000000000000E+0;
e[37]=1.045000000000000E+0;
e[38]=1.020000000000000E+0;
e[39]=9.960000000000001E-1;
e[40]=9.720000000000000E-1;
e[41]=9.500000000000000E-1;
e[42]=9.099999999999999E-1;
e[43]=8.500000000000000E-1;
e[44]=7.800000000000000E-1;
e[45]=6.250000000000000E-1;
e[46]=5.000000000000000E-1;
e[47]=4.000000000000000E-1;
```

```
e[48]=3.500000000000000E-1;
e[49]=3.200000000000000E-1;
e[50]=3.000000000000000E-1;
e[51]=2.800000000000000E-1;
e[52]=2.480000000000000E-1;
e[53]=2.200000000000000E-1;
e[54]=1.800000000000000E-1;
e[55]=1.400000000000000E-1;
e[56]=9.999999999999999E-2;
e[57]=8.000000000000000E-2;
e[58]=6.700000000000000E-2;
e[59]=5.800000000000000E-2;
e[60]=5.000000000000000E-2;
e[61]=4.200000000000000E-2;
e[62]=3.500000000000000E-2;
e[63]=3.000000000000000E-2;
e[64]=2.500000000000000E-2;
e[65]=2.000000000000000E-2;
e[66]=1.500000000000000E-2;
e[67]=1.000000000000000E-2;
e[68]=5.000000000000000E-3;
e[69]=1.000000000000000E-5;

chi[ 0]=2.67091803e-02;
chi[ 1]=1.14537567e-01;
chi[ 2]=2.17341527e-01;
chi[ 3]=2.32795596e-01;
chi[ 4]=1.75022319e-01;
chi[ 5]=1.09199747e-01;
chi[ 6]=6.18954971e-02;
chi[ 7]=3.18517275e-02;
chi[ 8]=1.57796200e-02;
chi[ 9]=7.73822563e-03;
chi[10]=3.69500765e-03;
chi[11]=1.77082187e-03;
chi[12]=8.63423396e-04;
chi[13]=4.27901075e-04;
chi[14]=1.90922379e-04;
chi[15]=8.67884519e-05;
chi[16]=4.47510574e-05;
chi[17]=2.32361072e-05;
chi[18]=1.24022135e-05;
chi[19]=9.76445699e-06;
chi[20]=2.78398807e-06;
chi[21]=7.08925313e-07;
chi[22]=2.18123503e-07;
chi[23]=1.42040903e-07;
chi[24]=7.20953039e-08;
chi[25]=3.34741408e-08;
chi[26]=2.05054338e-08;
for (g=27;g<G;g++)
    {
    chi[g]=0;
    }
// Blotzmann constant 8.617343*1e-5 [eV/K]
for (g=0;g<G;g++)
    {
    v[g]=e[g]-((e[g]-e[g+1])*0.5);
    v[g]=2.2e5*sqrt(v[g]/(293.15*8.617343*1e-5));
// printf("v[%d]=%7.2e %7.2e\n",g,v[g], sqrt(1));
    }
for (g=0;g<G;g++)
    {
    v[g]=1/v[g];
    }
dataf=malloc(G*sizeof(double));
datar=malloc(G*sizeof(double));
dataa=malloc(G*sizeof(double));
datah=malloc(G*sizeof(double));
if ((ff=fopen("edtoutf","r"))==NULL) exit(1);
if ((fr=fopen("edtoutr","r"))==NULL) exit(1);
if ((fa=fopen("edtouta","r"))==NULL) exit(1);
for (i=0;i<SKIP;i++)
    {
```

```
    fgets(buff,256, ff);
    fgets(bufr,256, fr);
    fgets(bufa,256, fa);
    }
while (!feof(ff))
   {
   for (i=0;i<LINES-1;i++)
       {
       fgets(buff,256,ff);
       fgets(bufr,256,fr);
       fgets(bufa,256,fa);
       sscanf(buff,"%lf %lf %lf %lf %lf %lf",&xf[0],&xf[1],&xf[2],&xf[3],&xf[4],&xf[5]);
       sscanf(bufr,"%lf %lf %lf %lf %lf %lf",&xr[0],&xr[1],&xr[2],&xr[3],&xr[4],&xr[5]);
       sscanf(bufa,"%lf %lf %lf %lf %lf %lf",&xa[0],&xa[1],&xa[2],&xa[3],&xa[4],&xa[5]);
       for (g=0;g<6;g++)
           {
           dataf[g+(i*6)]=xf[g];
           datar[g+(i*6)]=xr[g];
           dataa[g+(i*6)]=xa[g];
           }
       }
   fgets(buff,256, ff);
   fgets(bufr,256, fr);
   fgets(bufa,256, fa);
   sscanf(buff,"%lf %lf %lf %lf %lf %lf",&xf[0],&xf[1],&xf[2],&xf[3],&xf[4],&xf[5]);
   sscanf(bufr,"%lf %lf %lf %lf %lf %lf",&xr[0],&xr[1],&xr[2],&xr[3],&xr[4],&xr[5]);
   sscanf(bufa,"%lf %lf %lf %lf %lf %lf",&xa[0],&xa[1],&xa[2],&xa[3],&xa[4],&xa[5]);
   if (!feof(ff))
      {
      for (g=0;g<ODDG;g++)
          {
          dataf[g+(i*6)]=xf[g];
          datar[g+(i*6)]=xr[g];
          dataa[g+(i*6)]=xa[g];
          }
      for (g=0;g<G;g++)
          {
          datah[g]=dataa[g];
          }
      h=0;
      for (g=0;g<G;g++)
          {
// uncomment this line if partisn is version 2 (do not uncomment if version 5)
//        dataa[g]=datah[G-1-g];
          h=h+datar[g];
          }
      for (g=0;g<G;g++)
          {
// adjoint calculation
          num=num+dataf[g]*dataa[g]*v[g];
          den=den+h*dataa[g]*chi[g];
// no adjoint calculation
//        num=num+dataf[g]*1.0*v[g];
//        den=den+h*1.0*chi[g];
          }
      }
   }
fclose(ff);
fclose(fr);
fclose(fa);
free(dataf);
free(datar);
free(dataa);
free(datah);
printf("The prompt neutron generation time is %15.10e\n",num/den);

}
```

Appendix G: Source Code for the Pulse Superimposition

To code can be compiled on Linux systems by the command:

```
gcc –o pulse pulse.c
```

The code can be executed by the command:

```
./pulse rate.txt time.txt
```

The code expects two files in the current directory:

1) `rate.txt` containing the reaction rate and its relative error calculated by MCNPX
2) `time.txt` containing the time bin (in shakes) for the reaction rate

The samples of these two files are given in Appendices H and I.

The code works for 25000 pulses, 20 ms period of the external neutron source, and 25 processors.

```
// Author: Alberto Talamo
// Email : alby@anl.gov
// Program written to calculate Yalina Thermal response to neutron pulses by MCNP
// The program superimposes 25000 pulses and it uses 25 MPI processors

#include <stdio.h>
#include <math.h>
#include <stdlib.h>
#include <string.h>
#include "/usr/local/mpich/intel/include/mpi.h"

#define NM    441     // number of MCNP data to process
#define N     50000000// maximum of a vector monitoring about 500 s with 10 micros step
#define NP    25000   // number of pulses in about 500 s occurring each 20 millis
#define NPI   2000    // number of time intervals in 20 millis with 10 micros step

static double                X[50000000],XP[50000000];  // take the memory from heap instead of stack
which is limited

int main(int argc,char *argv[])
{
FILE                 *fp=NULL,*f0,*f1,*f2,*f3,*f4,*f5,*f6,*f7,*f8,*f9,*f10,*f11,*f12,*f13;
char                  buf[256],*head;
int                   myid,numprocs,err;
unsigned long         i,j,k,m,n;
double                time;
double               *t,*dt,*x,*e;

MPI_Init(&argc, &argv);
MPI_Comm_size(MPI_COMM_WORLD,&numprocs);
MPI_Comm_rank(MPI_COMM_WORLD,&myid);

t =calloc(NM  ,sizeof(double));
dt=calloc(NM-1,sizeof(double));
x =calloc(NM  ,sizeof(double));
e =calloc(NM  ,sizeof(double));

if (argc<3)
   {
   printf("Usage: pulse filename_x filename_t\n");
   exit(1);
   }

if (myid==0) printf ("\n");
if (myid==0) printf ("Reading reaction rates from the MCNP output\n");
if (myid==0) printf ("\n");
if ((fp=fopen(argv[1],"r"))==NULL) {printf("file 1 not found\n");exit(1);}
k=0;
while (!feof(fp))
    {
    fgets(buf,256,fp);
    head=strtok(buf," ");
    x[k]=atof(head);
    head=strtok(NULL," ");
    e[k]=atof(head);
    if (myid==0) printf("reaction rate[%3lu]=%10e      error[%3lu]=%10e\n",k,x[k],k,e[k]);
    k++;
    if (k==NM) break;
    for (i=2;i<=5;i++)
        {
        head=strtok(NULL," ");
        x[k]=atof(head);
        head=strtok(NULL," ");
        e[k]=atof(head);
        if (myid==0) printf("reaction rate[%3lu]=%10e      error[%3lu]=%10e\n",k,x[k],k,e[k]);
        k++;
        }
    }
fclose (fp);

if (myid==0) printf ("\n");
if (myid==0) printf ("Reading time structure from the MCNP output\n");
if (myid==0) printf ("\n");
```

```
if ((fp=fopen(argv[2],"r"))==NULL) {printf("file 2 not found\n");exit(1);}
k=0;
while (!feof(fp))
    {
    fgets(buf,256,fp);
    head=strtok(buf," ");
    t[k]=atof(head)*1e-8;
    if (myid==0) printf("t[%3lu]=%7g\n",k,t[k]);
    k++;
    if (k==NM) break;
    for (i=2;i<=5;i++)
        {
        head=strtok(NULL," ");
        t[k]=atof(head)*1e-8;
        if (myid==0) printf("t[%3lu]=%7g\n",k,t[k]);
        k++;
        }
    }
fclose (fp);

if (myid==0) printf ("\n");
if (myid==0) printf ("Check for max or min arguments in the execution line\n");
if (myid==0) printf ("\n");
if (strstr(argv[3],"max"))
   {
   for (i=0;i<NM-1;i++) x[i]=x[i]+x[i]*e[i];
   if (myid==0) printf ("\n");
   if (myid==0) printf ("Statistical error added\n");
   if (myid==0) printf ("\n");
   }
if (strstr(argv[3],"min"))
   {
   for (i=0;i<NM-1;i++) x[i]=x[i]-x[i]*e[i];
   if (myid==0) printf ("\n");
   if (myid==0) printf ("Statistical error subtracted\n");
   if (myid==0) printf ("\n");
   }

if (myid==0) printf ("\n");
if (myid==0) printf ("Calculating time intervals\n");
if (myid==0) printf ("\n");
for (i=0;i<NM-1;i++) dt[i]=(t[i+1]-t[i]);
if (myid==0) {for (i=0;i<NM-1;i++) printf("between t[%3lu]=%7g \t and \t t[%3lu]=%7g \t
dt[%3lu]=%7g\n",i,t[i],i,t[i+1],i,dt[i]);}

if (myid==0) printf ("\n");
if (myid==0) printf ("Normalizing reaction rates\n");
if (myid==0) printf ("\n");
if (myid==0) printf("t[  0]=%7g \t x_old[  0]=%7g\t x_new[  0]=%7g\n",t[0],x[0],x[0]);
for (i=0;i<NM-1;i++)
    {
    if (myid==0) printf("t[%3lu]=%7g \t x_old[%3lu]=%7g\t",i+1,t[i+1],i+1,x[i+1]);
    x[i+1]=x[i+1]/dt[i];
    if (myid==0) printf("x_new[%3lu]=%7g\n",i+1,x[i+1]);
    }

if (myid==0) printf ("\n");
if (myid==0) printf ("Calculating response of 1 pulse over %d [s]\n",N-1);
if (myid==0) printf ("\n");
for (i=0;i<N;i++)
    {
    X[i] =0;
    XP[i]=0;
    }
i=0;
time=0;
for (j=0;j<N;j++)
    {
    X[j]=x[i]; //+((x[i+1]-x[i])*(time-t[i])/(t[i+1]-t[i]));
    time=time+dt[0];
    if ((X[j]>1e8)||(X[j]<0)) {printf("error on i=%lu\n",j); exit(1);}
    if (time>=0.9999999999*t[i+1]) i++;
    }
if (myid==0)
```

```
   {
   f0 =fopen("0.m","w");
   fprintf(f0 ,"clear all; close all;\n");
   fprintf(f0 ,"x=[\n");
   time=0;
   for (j=0;j<(10*NPI);j++)
       {
       fprintf(f0,"%10f\t%10f\n",time,X[j]);   // first 10 1 pulse
       time=time+dt[0];
       }
   }
err=MPI_Barrier(MPI_COMM_WORLD);
if (myid==0) printf ("\n");
if (myid==0) printf ("Calculating %d pulses superimposition\n",NP);
if (myid==0) printf ("\n");
for (m=0;m<25;m++)
    {
    if (myid==m)
       {
       for (j=m*1000;j<(m+1)*1000;j++)
           {
           i=j*NPI;
           for (k=i;k<N;k++) XP[k]=XP[k]+X[k-i];
           }
       }
    }
/* old code for 5 processors
if (myid==0)
   {
   for (j=0;j<3000;j++)
       {
       i=j*NPI;
       for (k=i;k<N;k++) XP[k]=XP[k]+X[k-i];
       }
   }
if (myid==1)
   {
   for (j=3000;j<7000;j++)
       {
       i=j*NPI;
       for (k=i;k<N;k++) XP[k]=XP[k]+X[k-i];
       }
   }
if (myid==2)
   {
   for (j=7000;j<12000;j++)
       {
       i=j*NPI;
       for (k=i;k<N;k++) XP[k]=XP[k]+X[k-i];
       }
   }
if (myid==3)
   {
   for (j=12000;j<18000;j++)
       {
       i=j*NPI;
       for (k=i;k<N;k++) XP[k]=XP[k]+X[k-i];
       }
   }
if (myid==4)
   {
   for (j=18000;j<NP;j++)
       {
       i=j*NPI;
       for (k=i;k<N;k++) XP[k]=XP[k]+X[k-i];
       }
   }
*/
for (i=0;i<N;i++) X[i]=0;
printf("rank %d finished \n",myid);
err=MPI_Barrier(MPI_COMM_WORLD);
err=MPI_Allreduce(&XP[0]       ,&X[0]       ,10000000,MPI_DOUBLE,MPI_SUM,MPI_COMM_WORLD);
err=MPI_Allreduce(&XP[10000000],&X[10000000],10000000,MPI_DOUBLE,MPI_SUM,MPI_COMM_WORLD);
err=MPI_Allreduce(&XP[20000000],&X[20000000],10000000,MPI_DOUBLE,MPI_SUM,MPI_COMM_WORLD);
```

```
err=MPI_Allreduce(&XP[30000000],&X[30000000],10000000,MPI_DOUBLE,MPI_SUM,MPI_COMM_WORLD);
err=MPI_Allreduce(&XP[40000000],&X[40000000],10000000,MPI_DOUBLE,MPI_SUM,MPI_COMM_WORLD);

if (myid==0)
   {
   printf ("\n");
   printf ("Writing Matlab files\n",NP);
   printf ("\n");
   f1 =fopen("1.m","w");
   f2 =fopen("2.m","w");
   f3 =fopen("3.m","w");
   f4 =fopen("4.m","w");
   f5 =fopen("5.m","w");
   f6 =fopen("6.m","w");
   f7 =fopen("7.m","w");
   f8 =fopen("8.m","w");
   f9 =fopen("9.m","w");
   f10=fopen("10.m","w");
   f11=fopen("11.m","w");
   f12=fopen("12.m","w");
   f13=fopen("13.m","w");
   fprintf(f1 ,"clear all; close all;\n");
   fprintf(f2 ,"clear all; close all;\n");
   fprintf(f3 ,"clear all; close all;\n");
   fprintf(f4 ,"clear all; close all;\n");
   fprintf(f5 ,"clear all; close all;\n");
   fprintf(f6 ,"clear all; close all;\n");
   fprintf(f7 ,"clear all; close all;\n");
   fprintf(f8 ,"clear all; close all;\n");
   fprintf(f9 ,"clear all; close all;\n");
   fprintf(f10,"clear all; close all;\n");
   fprintf(f11,"clear all; close all;\n");
   fprintf(f12,"clear all; close all;\n");
   fprintf(f13,"clear all; close all;\n");
   fprintf(f1 ,"x=[\n");
   fprintf(f2 ,"x=[\n");
   fprintf(f3 ,"x=[\n");
   fprintf(f4 ,"x=[\n");
   fprintf(f5 ,"x=[\n");
   fprintf(f6 ,"x=[\n");
   fprintf(f7 ,"x=[\n");
   fprintf(f8 ,"x=[\n");
   fprintf(f9 ,"x=[\n");
   fprintf(f10,"x=[\n");
   fprintf(f11,"x=[\n");
   fprintf(f12,"x=[\n");
   fprintf(f13,"x=[\n");
   time=0;
   for (j=0;j<N;j++)
       {
       if ((j>=(          0))&&(j<(   10*NPI)))   fprintf(f1 ,"%10f\t%10f\n",time,X[j]);
       if ((j>=(    10 *NPI))&&(j<(   20*NPI)))   fprintf(f2 ,"%10f\t%10f\n",time,X[j]);
       if ((j>=(    20 *NPI))&&(j<(   30*NPI)))   fprintf(f3 ,"%10f\t%10f\n",time,X[j]);
       if ((j>=(    30 *NPI))&&(j<(   40*NPI)))   fprintf(f4 ,"%10f\t%10f\n",time,X[j]);
       if ((j>=(    40 *NPI))&&(j<(   50*NPI)))   fprintf(f5 ,"%10f\t%10f\n",time,X[j]);
       if ((j>=(    90 *NPI))&&(j<(  100*NPI)))   fprintf(f6 ,"%10f\t%10f\n",time,X[j]);
       if ((j>=(   490 *NPI))&&(j<(  500*NPI)))   fprintf(f7 ,"%10f\t%10f\n",time,X[j]);
       if ((j>=(   990 *NPI))&&(j<( 1000*NPI)))   fprintf(f8 ,"%10f\t%10f\n",time,X[j]);
       if ((j>=(  4990 *NPI))&&(j<( 5000*NPI)))   fprintf(f9 ,"%10f\t%10f\n",time,X[j]);
       if ((j>=(  9990 *NPI))&&(j<(10000*NPI)))   fprintf(f10,"%10f\t%10f\n",time,X[j]);
       if ((j>=( 14990 *NPI))&&(j<(15000*NPI)))   fprintf(f11,"%10f\t%10f\n",time,X[j]);
       if ((j>=( 19990 *NPI))&&(j<(20000*NPI)))   fprintf(f12,"%10f\t%10f\n",time,X[j]);
       if ((j>=((NP-10)*NPI))&&(j<(   NP*NPI)))   fprintf(f13,"%10f\t%10f\n",time,X[j]);
       time=time+dt[0];
       }
   fprintf(f0 ,"];\n");
   fprintf(f1 ,"];\n");
   fprintf(f2 ,"];\n");
   fprintf(f3 ,"];\n");
   fprintf(f4 ,"];\n");
   fprintf(f5 ,"];\n");
   fprintf(f6 ,"];\n");
   fprintf(f7 ,"];\n");
   fprintf(f8 ,"];\n");
```

```
   fprintf(f9 ,"];\n");
   fprintf(f10,"];\n");
   fprintf(f11,"];\n");
   fprintf(f12,"];\n");
   fprintf(f13,"];\n");
   }
MPI_Finalize();
}
```

Appendix H: Sample for Rate.txt

0.00000E+00	0.0000	1.13953E-01	0.0044	1.56660E-01	0.0054	1.55760E-01	0.0057	1.55946E-01	0.0057
1.54897E-01	0.0056	1.51131E-01	0.0055	1.48514E-01	0.0054	1.44838E-01	0.0053	1.41631E-01	0.0052
1.35926E-01	0.0051	1.34038E-01	0.0050	1.29531E-01	0.0049	1.26246E-01	0.0048	1.22626E-01	0.0048
1.18163E-01	0.0047	1.15817E-01	0.0047	1.13572E-01	0.0047	1.12073E-01	0.0046	1.08705E-01	0.0047
1.04747E-01	0.0047	1.03021E-01	0.0048	1.00640E-01	0.0048	9.84268E-02	0.0049	9.55888E-02	0.0050
9.36875E-02	0.0050	9.13924E-02	0.0051	8.93243E-02	0.0051	8.90116E-02	0.0051	8.68522E-02	0.0052
8.52221E-02	0.0052	8.37964E-02	0.0053	8.18419E-02	0.0054	8.05008E-02	0.0054	7.95001E-02	0.0054
7.77117E-02	0.0055	7.57668E-02	0.0055	7.48047E-02	0.0056	7.27074E-02	0.0056	7.15114E-02	0.0057
7.09407E-02	0.0057	6.91205E-02	0.0058	6.76630E-02	0.0059	6.62798E-02	0.0059	6.50461E-02	0.0060
6.46022E-02	0.0060	6.31716E-02	0.0061	6.28360E-02	0.0061	6.16593E-02	0.0061	6.10662E-02	0.0062
5.99103E-02	0.0063	5.83741E-02	0.0063	5.86149E-02	0.0063	5.69606E-02	0.0064	5.58404E-02	0.0065
5.49339E-02	0.0065	5.43017E-02	0.0066	5.35027E-02	0.0066	5.25807E-02	0.0066	5.16728E-02	0.0067
5.10006E-02	0.0068	5.03954E-02	0.0068	4.96597E-02	0.0068	4.93111E-02	0.0069	4.80082E-02	0.0069
4.79212E-02	0.0070	4.67486E-02	0.0071	4.57162E-02	0.0071	4.53564E-02	0.0072	4.48214E-02	0.0072
4.38344E-02	0.0073	4.36790E-02	0.0073	4.28998E-02	0.0074	4.26506E-02	0.0075	4.17037E-02	0.0074
4.07680E-02	0.0076	4.02451E-02	0.0076	3.96806E-02	0.0077	3.93150E-02	0.0077	3.80349E-02	0.0078
3.76718E-02	0.0079	3.72325E-02	0.0079	3.66437E-02	0.0080	3.65179E-02	0.0079	3.61772E-02	0.0080
3.53905E-02	0.0081	3.51348E-02	0.0081	3.46768E-02	0.0082	3.40993E-02	0.0082	3.35354E-02	0.0083
3.28490E-02	0.0084	3.24995E-02	0.0085	3.21134E-02	0.0085	3.15211E-02	0.0086	3.13025E-02	0.0086
3.09288E-02	0.0087	2.98677E-02	0.0088	3.02280E-02	0.0088	2.97954E-02	0.0088	2.92877E-02	0.0090
2.89945E-02	0.0089	2.69470E-01	0.0035	2.36965E-01	0.0038	2.08502E-01	0.0040	1.82458E-01	0.0043
1.62623E-01	0.0046	1.44748E-01	0.0048	1.28111E-01	0.0051	1.13624E-01	0.0054	1.01103E-01	0.0058
9.06354E-02	0.0061	8.15077E-02	0.0064	7.24202E-02	0.0068	6.46975E-02	0.0072	5.83484E-02	0.0076
5.30132E-02	0.0079	4.84682E-02	0.0083	4.30296E-02	0.0088	4.00458E-02	0.0091	3.54141E-02	0.0096
3.23037E-02	0.0102	3.01773E-02	0.0105	2.69960E-02	0.0111	2.44441E-02	0.0116	2.30416E-02	0.0121
2.06530E-02	0.0128	1.94253E-02	0.0132	1.74553E-02	0.0139	1.58821E-02	0.0144	1.48419E-02	0.0151
1.33209E-02	0.0158	1.24353E-02	0.0163	1.13763E-02	0.0171	1.05739E-02	0.0179	9.78074E-03	0.0187
8.74149E-03	0.0193	8.11011E-03	0.0202	7.56096E-03	0.0209	7.12350E-03	0.0215	6.51862E-03	0.0228
6.13514E-03	0.0232	2.39544E-02	0.0131	1.68607E-02	0.0156	1.17828E-02	0.0191	7.73690E-03	0.0235
5.67600E-03	0.0274	4.10335E-03	0.0316	2.87740E-03	0.0383	2.05563E-03	0.0451	1.33442E-03	0.0547
1.06112E-03	0.0604	8.26713E-04	0.0666	6.30884E-04	0.0849	4.76810E-04	0.0930	3.24849E-04	0.0997
2.71932E-04	0.1122	2.35398E-04	0.1196	2.23960E-04	0.1310	2.44937E-04	0.1281	1.84731E-04	0.1521
2.22477E-04	0.1291	2.18682E-04	0.1382	1.86220E-04	0.1460	1.63188E-04	0.1552	1.57264E-04	0.1556
1.93255E-04	0.1344	1.98428E-04	0.1340	1.48398E-04	0.1602	1.34891E-04	0.1613	1.68635E-04	0.1639
1.52153E-04	0.1477	3.99563E-04	0.1285	3.48577E-04	0.1121	3.81642E-04	0.1108	3.02179E-04	0.1113
3.29266E-04	0.1091	3.24334E-04	0.1184	3.48061E-04	0.1265	2.94098E-04	0.1228	3.45040E-04	0.1097
3.54714E-04	0.1060	3.38655E-04	0.1187	2.52146E-04	0.1139	3.64444E-04	0.1065	3.05950E-04	0.1316
3.04728E-04	0.1171	3.59598E-04	0.1063	3.36195E-04	0.1170	3.12289E-04	0.1148	3.03802E-04	0.1101
2.79353E-04	0.1216	3.46046E-04	0.1210	3.69794E-04	0.1172	3.56293E-04	0.1225	3.19850E-04	0.1124
3.19862E-04	0.1091	3.49552E-04	0.1241	3.24110E-04	0.1182	2.66388E-04	0.1189	2.75031E-04	0.1144
2.80285E-04	0.1159	1.65059E-03	0.0582	1.59504E-03	0.0599	1.57943E-03	0.0535	1.63417E-03	0.0559
1.54008E-03	0.0571	1.48346E-03	0.0549	1.73376E-03	0.0565	1.50201E-03	0.0560	1.47157E-03	0.0571
1.39804E-03	0.0563	1.45452E-03	0.0580	1.59813E-03	0.0582	1.37611E-03	0.0578	1.46945E-03	0.0564
1.51134E-03	0.0561	1.44513E-03	0.0558	1.43053E-03	0.0563	1.41956E-03	0.0580	1.51650E-03	0.0594
1.39505E-03	0.0606	1.48387E-03	0.0582	1.44960E-03	0.0590	1.35735E-03	0.0581	1.35347E-03	0.0601
1.49075E-03	0.0607	1.33549E-03	0.0618	1.43172E-03	0.0571	1.26667E-03	0.0593	1.41121E-03	0.0600
1.32772E-03	0.0582	2.70244E-03	0.0451	2.67400E-03	0.0435	2.61244E-03	0.0434	2.56794E-03	0.0449
2.45889E-03	0.0465	2.37826E-03	0.0487	2.43131E-03	0.0443	2.38541E-03	0.0462	2.36875E-03	0.0436
2.25053E-03	0.0489	2.38931E-03	0.0466	2.40863E-03	0.0469	2.53644E-03	0.0469	2.40057E-03	0.0473
2.08470E-03	0.0490	2.31089E-03	0.0496	2.06541E-03	0.0477	2.03168E-03	0.0499	2.24679E-03	0.0495
2.15067E-03	0.0485	1.96334E-03	0.0485	1.98583E-03	0.0498	1.98651E-03	0.0508	1.77756E-03	0.0504

1.99253E-03	0.0515	2.16522E-03	0.0469	1.97136E-03	0.0485	1.99735E-03	0.0527	1.95890E-03	0.0498
1.86997E-03	0.0500	9.67258E-03	0.0239	8.88619E-03	0.0239	8.32593E-03	0.0247	8.34326E-03	0.0253
8.27398E-03	0.0254	7.93475E-03	0.0259	7.33274E-03	0.0268	7.38122E-03	0.0263	7.04349E-03	0.0277
6.68962E-03	0.0279	7.02470E-03	0.0277	6.80722E-03	0.0275	6.50051E-03	0.0287	6.09132E-03	0.0292
5.73938E-03	0.0300	5.73200E-03	0.0292	5.69906E-03	0.0301	5.79559E-03	0.0302	5.42505E-03	0.0307
5.59119E-03	0.0309	4.96872E-03	0.0314	5.07365E-03	0.0310	4.81610E-03	0.0331	4.70591E-03	0.0331
4.73632E-03	0.0330	4.57248E-03	0.0341	4.94824E-03	0.0337	4.54515E-03	0.0364	4.40218E-03	0.0342
4.32630E-03	0.0339	8.63094E-03	0.0246	8.18702E-03	0.0258	7.88695E-03	0.0259	7.69666E-03	0.0256
7.27504E-03	0.0267	7.36160E-03	0.0270	6.72561E-03	0.0282	6.41543E-03	0.0290	6.23203E-03	0.0295
6.23244E-03	0.0293	5.90902E-03	0.0294	5.97301E-03	0.0298	5.61064E-03	0.0306	5.29572E-03	0.0310
5.36760E-03	0.0323	5.02735E-03	0.0327	5.33547E-03	0.0323	5.06522E-03	0.0322	4.76250E-03	0.0333
4.65605E-03	0.0339	4.27969E-03	0.0346	4.38742E-03	0.0331	4.16895E-03	0.0346	4.43277E-03	0.0349
4.21516E-03	0.0356	3.90201E-03	0.0374	3.77091E-03	0.0369	3.87244E-03	0.0378	3.41445E-03	0.0379
3.78706E-03	0.0370	1.71562E-02	0.0178	1.50575E-02	0.0190	1.32887E-02	0.0204	1.25407E-02	0.0206
1.10767E-02	0.0222	1.01280E-02	0.0224	9.24799E-03	0.0236	8.31601E-03	0.0248	8.11823E-03	0.0269
7.25752E-03	0.0262	6.39352E-03	0.0273	5.79556E-03	0.0289	6.01232E-03	0.0301	5.46315E-03	0.0301
4.99945E-03	0.0324	4.89279E-03	0.0329	4.62607E-03	0.0338	4.04479E-03	0.0359	3.85734E-03	0.0370
3.68891E-03	0.0368	3.80823E-03	0.0398	3.38065E-03	0.0397	3.34334E-03	0.0420	2.92685E-03	0.0413
2.99919E-03	0.0418	2.85733E-03	0.0436	2.68329E-03	0.0431	2.68845E-03	0.0447	2.47609E-03	0.0460
2.45455E-03	0.0466	4.65322E-03	0.0338	4.40926E-03	0.0345	3.98132E-03	0.0347	3.58353E-03	0.0382
3.63164E-03	0.0384	3.47037E-03	0.0408	3.17182E-03	0.0426	3.14641E-03	0.0417	2.87304E-03	0.0429
2.70446E-03	0.0452	2.69467E-03	0.0441	2.62359E-03	0.0451	2.42885E-03	0.0449	2.28325E-03	0.0474
2.22831E-03	0.0506	1.92805E-03	0.0515	2.13871E-03	0.0496	2.02847E-03	0.0510	1.88068E-03	0.0534
1.86167E-03	0.0536	1.74200E-03	0.0559	1.80550E-03	0.0568	1.67727E-03	0.0543	1.65363E-03	0.0556
1.61088E-03	0.0585	1.54724E-03	0.0587	1.24987E-03	0.0602	1.31911E-03	0.0630	1.17349E-03	0.0649
1.25042E-03	0.0664	5.86865E-03	0.0300	4.69511E-03	0.0330	4.54096E-03	0.0355	3.55668E-03	0.0398
3.19440E-03	0.0409	2.65545E-03	0.0434	2.21265E-03	0.0492	1.91576E-03	0.0502	1.77105E-03	0.0548
1.66468E-03	0.0570	1.53200E-03	0.0586	1.29424E-03	0.0697	1.02859E-03	0.0699	9.96081E-04	0.0708
8.29715E-04	0.0834	8.52000E-04	0.0797	7.84000E-04	0.0852	6.28183E-04	0.0926	6.19252E-04	0.0979
5.24738E-04	0.1026	3.85752E-04	0.1073	4.37039E-04	0.1181	3.39794E-04	0.1265	2.76690E-04	0.1315
3.04890E-04	0.1249	3.40338E-04	0.1162	2.81562E-04	0.1430	1.75400E-04	0.1598	2.08145E-04	0.1684
2.22416E-04	0.1540	3.05308E-04	0.1314	2.63016E-04	0.1433	2.36799E-04	0.1397	1.61522E-04	0.1707
1.52322E-04	0.1742	9.30407E-05	0.1891	1.21051E-04	0.1938	7.99246E-05	0.2277	9.02298E-05	0.2503
8.88070E-05	0.2107	6.67547E-05	0.2309	8.55529E-05	0.2524	2.82177E-05	0.3368	6.27147E-05	0.2734
3.14178E-05	0.3882	3.74369E-05	0.2913	2.69919E-05	0.4891	2.40634E-05	0.4842	1.71803E-05	0.4054
1.70788E-05	0.5934	1.82402E-05	0.4215	3.86693E-05	0.3795	1.17329E-05	0.5519	7.33087E-06	0.7606
4.09368E-05	0.7269	1.11636E-05	0.5159	4.93543E-06	0.7828	4.11381E-06	0.6781	3.63346E-11	0.6806
5.48587E-07	0.9982								

Appendix I: Sample for Time.txt

0.0000E+00	1.0000E+03	2.0000E+03	3.0000E+03	4.0000E+03
5.0000E+03	6.0000E+03	7.0000E+03	8.0000E+03	9.0000E+03
1.0000E+04	1.1000E+04	1.2000E+04	1.3000E+04	1.4000E+04
1.5000E+04	1.6000E+04	1.7000E+04	1.8000E+04	1.9000E+04
2.0000E+04	2.1000E+04	2.2000E+04	2.3000E+04	2.4000E+04
2.5000E+04	2.6000E+04	2.7000E+04	2.8000E+04	2.9000E+04
3.0000E+04	3.1000E+04	3.2000E+04	3.3000E+04	3.4000E+04
3.5000E+04	3.6000E+04	3.7000E+04	3.8000E+04	3.9000E+04
4.0000E+04	4.1000E+04	4.2000E+04	4.3000E+04	4.4000E+04
4.5000E+04	4.6000E+04	4.7000E+04	4.8000E+04	4.9000E+04
5.0000E+04	5.1000E+04	5.2000E+04	5.3000E+04	5.4000E+04
5.5000E+04	5.6000E+04	5.7000E+04	5.8000E+04	5.9000E+04
6.0000E+04	6.1000E+04	6.2000E+04	6.3000E+04	6.4000E+04
6.5000E+04	6.6000E+04	6.7000E+04	6.8000E+04	6.9000E+04
7.0000E+04	7.1000E+04	7.2000E+04	7.3000E+04	7.4000E+04
7.5000E+04	7.6000E+04	7.7000E+04	7.8000E+04	7.9000E+04
8.0000E+04	8.1000E+04	8.2000E+04	8.3000E+04	8.4000E+04
8.5000E+04	8.6000E+04	8.7000E+04	8.8000E+04	8.9000E+04
9.0000E+04	9.1000E+04	9.2000E+04	9.3000E+04	9.4000E+04
9.5000E+04	9.6000E+04	9.7000E+04	9.8000E+04	9.9000E+04
1.0000E+05	1.1000E+05	1.2000E+05	1.3000E+05	1.4000E+05
1.5000E+05	1.6000E+05	1.7000E+05	1.8000E+05	1.9000E+05
2.0000E+05	2.1000E+05	2.2000E+05	2.3000E+05	2.4000E+05
2.5000E+05	2.6000E+05	2.7000E+05	2.8000E+05	2.9000E+05
3.0000E+05	3.1000E+05	3.2000E+05	3.3000E+05	3.4000E+05
3.5000E+05	3.6000E+05	3.7000E+05	3.8000E+05	3.9000E+05
4.0000E+05	4.1000E+05	4.2000E+05	4.3000E+05	4.4000E+05
4.5000E+05	4.6000E+05	4.7000E+05	4.8000E+05	4.9000E+05
5.0000E+05	5.5000E+05	6.0000E+05	6.5000E+05	7.0000E+05
7.5000E+05	8.0000E+05	8.5000E+05	9.0000E+05	9.5000E+05
1.0000E+06	1.0500E+06	1.1000E+06	1.1500E+06	1.2000E+06
1.2500E+06	1.3000E+06	1.3500E+06	1.4000E+06	1.4500E+06
1.5000E+06	1.5500E+06	1.6000E+06	1.6500E+06	1.7000E+06
1.7500E+06	1.8000E+06	1.8500E+06	1.9000E+06	1.9500E+06
2.0000E+06	2.1000E+06	2.2000E+06	2.3000E+06	2.4000E+06
2.5000E+06	2.6000E+06	2.7000E+06	2.8000E+06	2.9000E+06
3.0000E+06	3.1000E+06	3.2000E+06	3.3000E+06	3.4000E+06
3.5000E+06	3.6000E+06	3.7000E+06	3.8000E+06	3.9000E+06
4.0000E+06	4.1000E+06	4.2000E+06	4.3000E+06	4.4000E+06
4.5000E+06	4.6000E+06	4.7000E+06	4.8000E+06	4.9000E+06
5.0000E+06	5.5000E+06	6.0000E+06	6.5000E+06	7.0000E+06
7.5000E+06	8.0000E+06	8.5000E+06	9.0000E+06	9.5000E+06
1.0000E+07	1.0500E+07	1.1000E+07	1.1500E+07	1.2000E+07
1.2500E+07	1.3000E+07	1.3500E+07	1.4000E+07	1.4500E+07
1.5000E+07	1.5500E+07	1.6000E+07	1.6500E+07	1.7000E+07
1.7500E+07	1.8000E+07	1.8500E+07	1.9000E+07	1.9500E+07
2.0000E+07	2.1000E+07	2.2000E+07	2.3000E+07	2.4000E+07
2.5000E+07	2.6000E+07	2.7000E+07	2.8000E+07	2.9000E+07
3.0000E+07	3.1000E+07	3.2000E+07	3.3000E+07	3.4000E+07
3.5000E+07	3.6000E+07	3.7000E+07	3.8000E+07	3.9000E+07
4.0000E+07	4.1000E+07	4.2000E+07	4.3000E+07	4.4000E+07

4.5000E+07	4.6000E+07	4.7000E+07	4.8000E+07	4.9000E+07
5.0000E+07	5.5000E+07	6.0000E+07	6.5000E+07	7.0000E+07
7.5000E+07	8.0000E+07	8.5000E+07	9.0000E+07	9.5000E+07
1.0000E+08	1.0500E+08	1.1000E+08	1.1500E+08	1.2000E+08
1.2500E+08	1.3000E+08	1.3500E+08	1.4000E+08	1.4500E+08
1.5000E+08	1.5500E+08	1.6000E+08	1.6500E+08	1.7000E+08
1.7500E+08	1.8000E+08	1.8500E+08	1.9000E+08	1.9500E+08
2.0000E+08	2.1000E+08	2.2000E+08	2.3000E+08	2.4000E+08
2.5000E+08	2.6000E+08	2.7000E+08	2.8000E+08	2.9000E+08
3.0000E+08	3.1000E+08	3.2000E+08	3.3000E+08	3.4000E+08
3.5000E+08	3.6000E+08	3.7000E+08	3.8000E+08	3.9000E+08
4.0000E+08	4.1000E+08	4.2000E+08	4.3000E+08	4.4000E+08
4.5000E+08	4.6000E+08	4.7000E+08	4.8000E+08	4.9000E+08
5.0000E+08	5.5000E+08	6.0000E+08	6.5000E+08	7.0000E+08
7.5000E+08	8.0000E+08	8.5000E+08	9.0000E+08	9.5000E+08
1.0000E+09	1.0500E+09	1.1000E+09	1.1500E+09	1.2000E+09
1.2500E+09	1.3000E+09	1.3500E+09	1.4000E+09	1.4500E+09
1.5000E+09	1.5500E+09	1.6000E+09	1.6500E+09	1.7000E+09
1.7500E+09	1.8000E+09	1.8500E+09	1.9000E+09	1.9500E+09
2.0000E+09	2.1000E+09	2.2000E+09	2.3000E+09	2.4000E+09
2.5000E+09	2.6000E+09	2.7000E+09	2.8000E+09	2.9000E+09
3.0000E+09	3.1000E+09	3.2000E+09	3.3000E+09	3.4000E+09
3.5000E+09	3.6000E+09	3.7000E+09	3.8000E+09	3.9000E+09
4.0000E+09	4.1000E+09	4.2000E+09	4.3000E+09	4.4000E+09
4.5000E+09	4.6000E+09	4.7000E+09	4.8000E+09	4.9000E+09
5.0000E+09	5.5000E+09	6.0000E+09	6.5000E+09	7.0000E+09
7.5000E+09	8.0000E+09	8.5000E+09	9.0000E+09	9.5000E+09
1.0000E+10	1.0500E+10	1.1000E+10	1.1500E+10	1.2000E+10
1.2500E+10	1.3000E+10	1.3500E+10	1.4000E+10	1.4500E+10
1.5000E+10	1.5500E+10	1.6000E+10	1.6500E+10	1.7000E+10
1.7500E+10	1.8000E+10	1.8500E+10	1.9000E+10	1.9500E+10
2.0000E+10	2.1000E+10	2.2000E+10	2.3000E+10	2.4000E+10
2.5000E+10	2.6000E+10	2.7000E+10	2.8000E+10	2.9000E+10
3.0000E+10	3.1000E+10	3.2000E+10	3.3000E+10	3.4000E+10
3.5000E+10	3.6000E+10	3.7000E+10	3.8000E+10	3.9000E+10
4.0000E+10	4.1000E+10	4.2000E+10	4.3000E+10	4.4000E+10
4.5000E+10	4.6000E+10	4.7000E+10	4.8000E+10	4.9000E+10
5.0000E+10				

Nuclear Engineering Division
Argonne National Laboratory
9700 South Cass Avenue
Argonne, IL 60439

www.anl.gov

A U.S. Department of Energy laboratory
managed by UChicago Argonne, LLC