

ORNL/SUB/99-19XSY062V-1  
ANRCP-1999-9

# ornl

**OAK RIDGE  
NATIONAL  
LABORATORY**

*LOCKHEED MARTIN*

## **Studies of Flexible MOX/LEU Fuel Cycles**

**Gustavo Alonso-Vargas  
Marvin L. Adams**

MANAGED AND OPERATED BY  
LOCKHEED MARTIN ENERGY RESEARCH  
CORPORATION  
FOR THE UNITED STATES  
DEPARTMENT OF ENERGY

ORNL-27 (3-96)

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 any of their employees, makes any warranty, expressed or implied, assumes any legal liability or responsibility for the accuracy, completeness, or usefulness or 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, recommendations, or favoring by the United States Government or any agency thereof. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof.

**STUDIES OF FLEXIBLE MOX/LEU FUEL CYCLES**

Gustavo Alonso-Vargas  
Marvin L. Adams

Date Published: March 1999

Report Prepared by  
LOCKHEED MARTIN ENERGY RESEARCH CORP.  
P.O. Box 2008  
Oak Ridge, Tennessee 37831-6363  
Under  
Subcontract Number 85B99398V

Funded by  
Office of Fissile Materials Disposition  
United States Department of Energy

Prepared for

Computation Physics Engineering Division  
Oak Ridge National Laboratory  
Oak Ridge, Tennessee 37931  
managed by  
LOCKHEED MARTIN ENERGY RESEARCH CORP.  
for the  
U.S. DEPARTMENT OF ENERGY  
under contract DE-96-OR22464



# **FINAL REPORT**

## **STUDIES OF FLEXIBLE MOX/LEU FUEL CYCLES**

*Gustavo Alonso-Vargas and Marvin L. Adams*

**Department of Nuclear Engineering  
Texas A&M University  
Zachry 129  
College Station, Texas 77843**

**March 1999**



# Studies of Flexible MOX/LEU Fuel Cycles

## Final Report, Texas A&M University / Amarillo National Resource Center

Gustavo Alonso-Vargas and Marvin L. Adams, Texas A&M University

### I. Introduction

This project was a collaborative effort involving researchers from Oak Ridge National Laboratory and North Carolina State University as well as Texas A&M University. The background, briefly, is that the United States is planning to use some of its excess weapons Plutonium (Pu) to make mixed-oxide (MOX) fuel for existing light-water reactors (LWRs). Considerable effort has already gone into designing fuel assemblies and core loading patterns for the transition from full-uranium cores to partial-MOX and full-MOX cores. However, these designs have assumed that any time a reactor needs MOX assemblies, these assemblies will be supplied. In reality there are many possible scenarios under which this supply could be disrupted. It therefore seems prudent to verify that a reactor-based Pu-disposition program could tolerate such interruptions in an acceptable manner. Such verification was the overall aim of this project.

The task assigned to the Texas A&M team was to use the HELIOS code<sup>1,2</sup> to develop libraries of two-group homogenized cross sections for the various assembly designs that might be used in a Westinghouse Pressurized Water Reactor (PWR) that is burning weapons-grade MOX fuel. The NCSU team used these cross sections to develop optimized loading patterns under several assumed scenarios. Their results are documented in a companion report.<sup>3</sup>

### II. Methodology

Our starting point in developing the library was to test the HELIOS models of UO<sub>2</sub> and MOX assemblies. We compared the results against those from the CASMO-3 code.<sup>4</sup> This served as useful check on our HELIOS models and in addition it is desirable for a variety of reasons to understand the relative difference between the two codes.

Our HELIOS calculations used the “hy961a” cross-section library, which is an update of the hy941a library.<sup>5</sup> This library is based upon the evaluated nuclear data file ENDF/B-VI.<sup>6</sup> Our final results were generated using a 34-group cross-section set for LEU assemblies and an 89-group set for MOX assemblies, as described later. Our CASMO-3 calculations used the CASMO “K” libraries unless otherwise noted below; the exception was some calculations made to compare results from the “J” and “K” libraries. The main difference between the “K” and “J” libraries is an improvement in the cross sections for the higher Pu isotopes.<sup>7</sup>

When there were discrepancies between the results given by HELIOS and CASMO we investigated the cause. This led us to correct several input and code-usage mistakes, but in some cases it led us to conclude that the discrepancies resulted from certain differences in underlying cross-section libraries or code methodologies. Only when we were sure that we were correctly modeling our assemblies did we proceed to generate the different HELIOS models for the different assemblies.

From the HELIOS output files we generated the library of two-group homogenized cross sections using the ZENITH code. ZENITH input files were generated according the FORMOSA input data required. North Carolina State University provided us with the input data structure of FORMOSA.

We modeled both UO<sub>2</sub> and MOX assemblies. Figure 1 shows the layout of one eighth of an assembly; this picture is the same for all assemblies considered. Table I provides some details about the dimensions and materials of each assembly. These details are the same as used by previous studies of the ability of Westinghouse reactors to be used for disposition of weapons Pu.<sup>8,9</sup>

In Table II.a we list the different assemblies that were modeled, with their respective enrichments of <sup>235</sup>U or fissile plutonium and the numbers and types of burnable absorbers used. For each of these assemblies there is a base case and five different branch cases such that each HELIOS file produces six different ZENITH output files. Table II.b



shows the conditions under which these cases were modeled. For the baffle model there are only four branch cases. For the full MOX cores the soluble boron is enriched to 40% B-10; the partial MOX cores use natural boron (approximately 20% B-10). We created libraries for both the natural and enriched scenarios, but only the enriched libraries were passed along to NCSU. The ZENITH output files constituted the two-group libraries used by NCSU for further analysis.

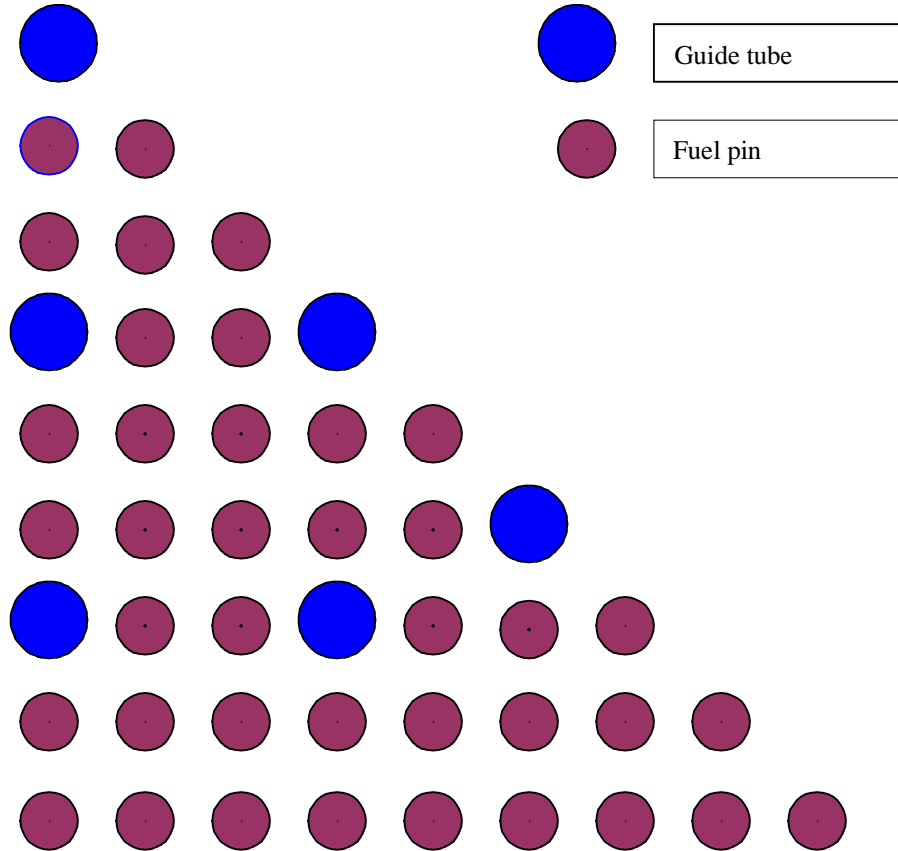


Figure 1. One eighth of an assembly.

Table II.a. MOX and UO<sub>2</sub> assemblies that were modeled.

Fuel lattice	17x17
Fuel rods per assembly	264
Fuel rod OD (in)	0.360
Fuel pellet OD (in)	0.3088
Cladding material	Zircaloy-4
Active Fuel Height	144
Clad Thickness (in)	0.0225
Pitch (in)	0.496
Assembly Pitch (in)	8.466
Assembly Gap (in)	0.034



Table II.a. MOX and UO<sub>2</sub> assemblies modeled.

Type	HELIOS file	Enrichment w/o	# IFBA	#WABA	Boron-10 per rod
MOX	hmhwh89h.inp	4.0	0	0	0.006 g/cm
MOX	Hmhw189g.inp	4.0	0	12	0.006 g/cm
MOX	hmhwn89g.inp	4.0	0	24	0.006 g/cm
MOX	hmlwh89g.inp	4.5	0	0	0.006 g/cm
MOX	hmlwl89g.inp	4.5	0	12	0.006 g/cm
MOX	hmlwn89g.inp	4.5	0	24	0.006 g/cm
UO <sub>2</sub>	hu35ni.inp	3.5	0	0	1.5 mg/in
UO <sub>2</sub>	hu3516i.inp	3.5	16	0	1.5 mg/in
UO <sub>2</sub>	hu3548i.inp	3.5	48	0	1.5 mg/in
UO <sub>2</sub>	hu3564i.inp	3.5	64	0	1.5 mg/in
UO <sub>2</sub>	hu3580i.inp	3.5	80	0	1.5 mg/in
UO <sub>2</sub>	hu35104i.inp	3.5	104	0	1.5 mg/in
UO <sub>2</sub>	hu35128i.inp	3.5	128	0	1.5 mg/in
UO <sub>2</sub>	hulni.inp	4.0	0	0	1.5 mg/in
UO <sub>2</sub>	hul16i.inp	4.0	16	0	1.5 mg/in
UO <sub>2</sub>	hul48i.inp	4.0	48	0	1.5 mg/in
UO <sub>2</sub>	hul64i.inp	4.0	64	0	1.5 mg/in
UO <sub>2</sub>	hul80i.inp	4.0	80	0	1.5 mg/in
UO <sub>2</sub>	hul104i.inp	4.0	104	0	1.5 mg/in
UO <sub>2</sub>	hul128i.inp	4.0	128	0	1.5 mg/in
UO <sub>2</sub>	huhni.inp	4.5	0	0	1.5 mg/in
UO <sub>2</sub>	huh16i.inp	4.5	16	0	1.5 mg/in
UO <sub>2</sub>	huh48i.inp	4.5	48	0	1.5 mg/in
UO <sub>2</sub>	huh64i.inp	4.5	64	0	1.5 mg/in
UO <sub>2</sub>	huh80i.inp	4.5	80	0	1.5 mg/in
UO <sub>2</sub>	huh104i.inp	4.5	104	0	1.5 mg/in
UO <sub>2</sub>	huh128i.inp	4.5	128	0	1.5 mg/in
BAFFLE	hmhwhref34.inp	4.5	0	24	0.006 g/cm

Table II.b. Base and branch conditions used in the HELIOS model.

Case	Fuel Temperature (K)	Moderator Temperature (K)	Boron concentration (PPM)
Base	963	582.95	1000
Fuel Temperature Low	582.95	582.95	1000
Moderator Temperature Low	963	564.59	1000
Moderator Temperature High	963	599.32	1000
Boron concentration Low	963	582.95	0
Boron concentration High	963	582.95	2000

### III. Results

#### III.A. Verification of correct use and understanding of HELIOS/ZENITH

We began our work by building a HELIOS model of a UO<sub>2</sub> assembly and comparing results against those from CASMO-3. The assembly type tested had 4.203 w/o of <sup>235</sup>U enrichment with integral fuel burnable absorber (IFBA) coatings incorporated onto 128 fuel pins. The infinite multiplication factor and absolute difference between results given by CASMO-3 and HELIOS are shown in Table III (along with results from a MOX assembly described later).

Table III. Multiplication factor for uranium and MOX assemblies.

	4.203 %w of <sup>235</sup> U			3.992 %w of fissile Pu		
BU	HELIOS	CASMO 70K	Difference	HELIOS	CASMO 70K	difference
0	1.11832	1.11716	0.00116	1.14394	1.15140	0.00746
500	1.09231	1.09137	0.00094	1.11523	1.12168	0.00645
1500	1.10314	1.10225	0.00089	1.10461	1.11101	0.00640
2500	1.11226	1.11139	0.00087	1.09712	1.10348	0.00636
5000	1.12340	1.12257	0.00083	1.08311	1.08953	0.00642
7500	1.12371	1.12319	0.00052	1.07169	1.07803	0.00634
10000	1.11702	1.11676	0.00026	1.06189	1.06809	0.00620
12500	1.10583	1.10575	0.00008	1.05303	1.05913	0.00610
15000	1.09180	1.09193	0.00013	1.04462	1.05065	0.00603
17500	1.07600	1.07618	0.00018	1.03620	1.04221	0.00601
20000	1.05915	1.05923	0.00008	1.02744	1.03352	0.00608
22500	1.04174	1.04186	0.00012	1.01809	1.02426	0.00617
25000	1.02407	1.02404	0.00003	1.00806	1.01429	0.00623
27500	1.00635	1.00611	0.00024	0.99733	1.00356	0.00623
30000	0.98872	0.98819	0.00053	0.98599	0.99207	0.00608
32500	0.97127	0.97031	0.00096	0.97416	0.97994	0.00578
35000	0.95396	0.95270	0.00126	0.96203	0.96735	0.00532
37500	0.93715	0.93518	0.00197	0.94973	0.95449	0.00476
40000	0.92060	0.91803	0.00257	0.93739	0.94150	0.00411
42500	0.90443	0.90123	0.00320	0.92512	0.92848	0.00336
45000	0.88868	0.88483	0.00385	0.91301	0.91557	0.00256
47500	0.87341	0.86888	0.00453	0.90109	0.90284	0.00175
50000	0.85862	0.85341	0.00521	0.88944	0.89033	0.00089
52500	0.84439	0.83850	0.00589	0.87808	0.87811	0.00003
55000	0.83075	0.82420	0.00655	0.86701	0.86619	-0.00082
57500	0.81772	0.81055	0.00717	0.85627	0.85461	-0.00166
60000	0.80533	0.79760	0.00773	0.84586	0.84339	-0.00247

Figure 2.a shows the behavior of the multiplication factor from HELIOS and CASMO-3, with several different cross-section libraries employed in the CASMO-3 runs. These were the 40-group and 70-group “J” libraries (denoted CASMO-40 and CASMO-70 in the figure) and the 70-group “K” library (denoted CASMO-70K). The “K” library was designed by Studsvik (the company that maintains and sells CASMO) to contain improved cross section data for Pu isotopes. We see that the different CASMO libraries cause smaller differences than the difference between CASMO and HELIOS. An important note is that we used only the 34-group library in the HELIOS runs.

Several runs were made in order to see if these discrepancies were related to the burnup step. When the number of burnup steps in CASMO was increased, the difference was smaller, however if we increase the number of burnup steps in HELIOS in order to have the same burnup steps as in CASMO, the difference returned to approximately its original value. Thus, we do not believe that burnup step is the source of a significant part of the difference; we believe instead that this difference stems from the different libraries and numerical approximations used in each code, and not to any parameters that are under the user’s control.

We turned next to a MOX assembly with an average of 3.992 w/o of fissile Pu, composed of four different pin enrichments and containing 24 wet annular burnable absorber (WABA) rods. In this case CASMO-3 and HELIOS differ at BOC, with CASMO higher. As was true for the uranium assembly, CASMO-3 depletes faster than HELIOS, which in the MOX case means the two codes approach the same k-infinity late in the cycle. The infinite

multiplication factor and absolute difference between results given by CASMO-3 and HELIOS are shown in Table III and Figure 2.b.

In the case of the MOX assembly we hypothesized that the differences were caused by differences in cross sections libraries. To test this hypothesis we replaced the weapons-grade plutonium in the assembly first with pure  $^{239}\text{Pu}$  and then with pure  $^{241}\text{Pu}$ . This, of course, isolates the differences in the two cross section libraries for these two fissile isotopes.

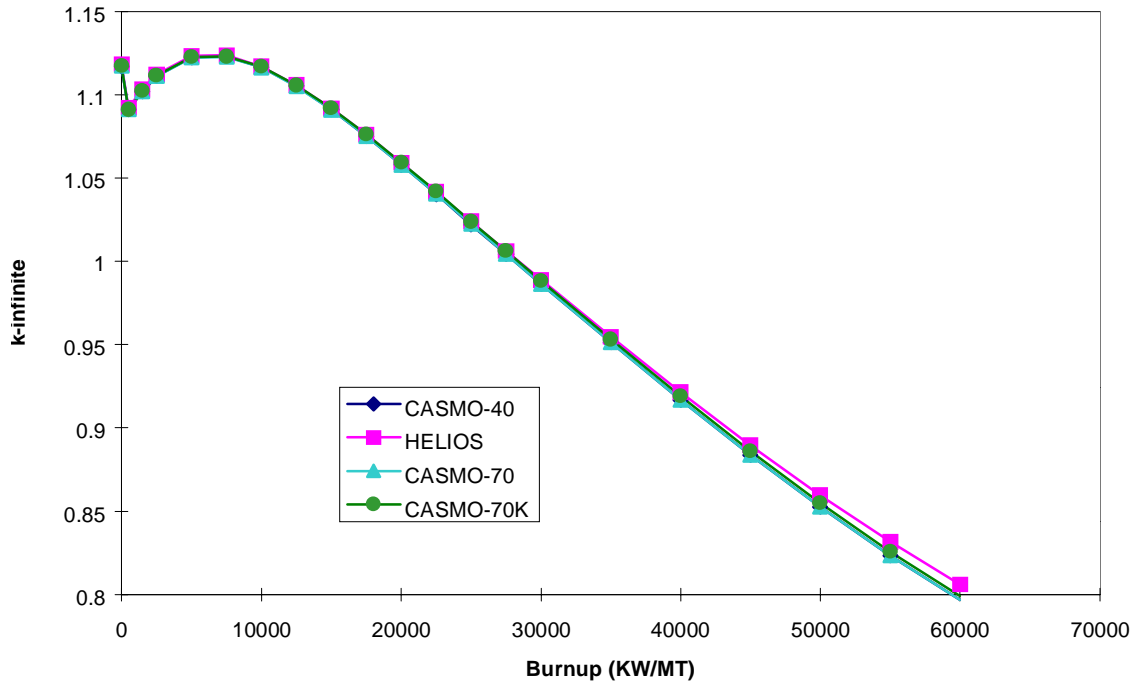


Figure 2.a. Behavior of multiplication factor for the UO<sub>2</sub> assembly.

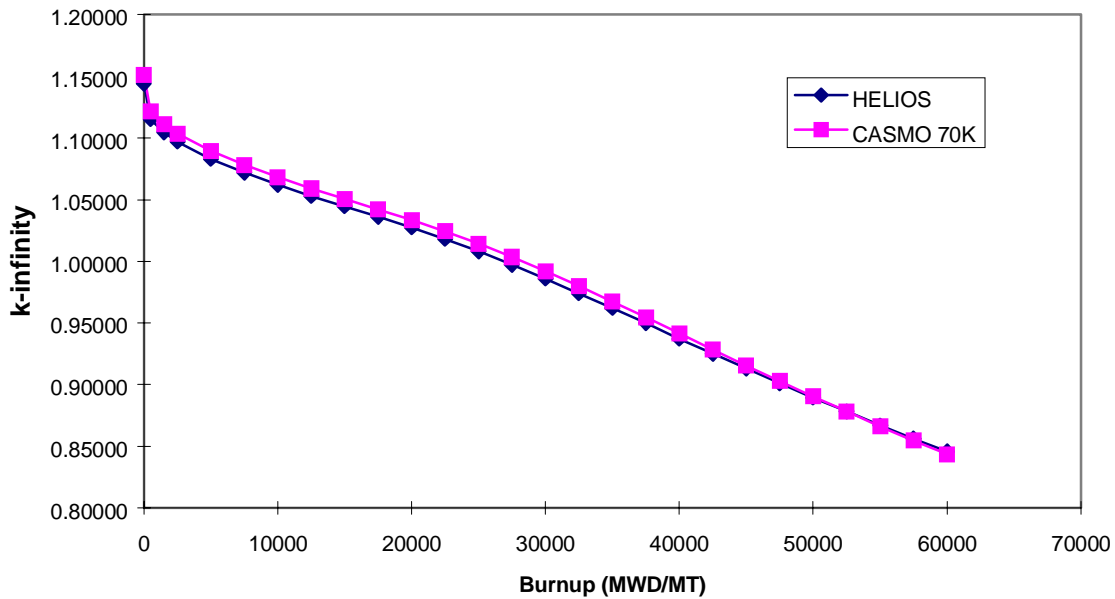


Figure 2.b. Behavior of multiplication factor for the MOX assembly.



The results from the pure  $^{239}\text{Pu}$  case were as expected, showing the same trends as in the original weapons-grade MOX runs. These results are given in the Table IV.

Table IV. k-infinity results with pure  $^{239}\text{Pu}$ .

Burnup (GWD/MT)	HELIOS	CASMO	Difference
0	1.14254	1.14992	0.00738
40	0.93781	0.94202	0.00421
50	0.88993	0.89097	0.00104
60	0.84635	0.84406	-0.00229

If we compare the MOX results from Table III against Table IV, we see that the differences between the two codes have the same sign but smaller magnitude. On the other hand, there is a very different trend for the  $^{241}\text{Pu}$  case. This is shown in Table V. At BOC there is a very large difference (2000 pcm) between the two codes, with a different sign than in the pure  $^{239}\text{Pu}$  case. That is, the CASMO library appears to have slightly more reactive  $^{239}\text{Pu}$  and considerably less reactive  $^{241}\text{Pu}$  than the HELIOS library.

Table V. K-infinity results with pure  $^{241}\text{Pu}$ .

Bu (GWD/MT)	HELIOS	CASMO	Difference
0	1.44485	1.42466	-0.02019
40	0.88829	0.87289	-0.01426
50	0.79332	0.77625	-0.01707
60	0.74200	0.72660	-0.01520

The results shown in Table V help to explain why the difference between the CASMO and HELIOS results changes sign in the other two cases. When almost pure  $^{239}\text{Pu}$  is present at BOC, then  $^{240}\text{Pu}$  and  $^{241}\text{Pu}$  will build up over time. At high burnup, much of the reactivity is due to the  $^{241}\text{Pu}$ , which explains why the CASMO k-infinity drops below that of HELIOS.

Moving forward in the comparison of results coming from these two codes, we encountered an unforeseen difficulty, namely a considerable disagreement between the Moderator Temperature Coefficient (MTC) predicted by HELIOS and the MTC predicted by CASMO-3, for a simple Low-Enriched Uranium (LEU) assembly.

We ran HELIOS for a base case and five branch cases for a Westinghouse LEU assembly with no control rods and no burnable absorbers. The key parameters for the base and branch cases are shown in Table VI below.

Table VI. Parameters of Base and Branch Cases

Case	Base	Branch 1	Branch 2	Branch 3	Branch 4	Branch 5
Fuel Temp (K)	<b>963</b>	<b>582.95</b>	963	963	963	963
Mod. Temp (K)	<b>582.95</b>	582.95	<b>599.32</b>	<b>564.59</b>	582.95	582.95
Boron Conc. (ppm)	<b>1000</b>	1000	1000	1000	<b>2000</b>	<b>0</b>

In Table VII below we present, for each branch at each depletion step, the ratio of the HELIOS  $k_{base}-k_{branch}$  to the CASMO-3 prediction of the same  $\Delta k$ . These  $\Delta k$ 's are approximately proportional to various reactivity coefficients. The base cases agree reasonably well between the two codes. We see that the fuel temperature coefficients from the two codes agree within a few percent except for a 9% outlier at 5 GWD/MT. The boron worth, despite being computed over fairly large changes in boron concentration, is in similarly good agreement.

The two codes seem to disagree significantly, however, on Moderator Temperature Coefficients (MTC). To determine the source of this disagreement we looked beyond k-infinity and studied the two-group assembly-averaged cross sections. There are two reasons for this. First, there is a tight connection between k-infinity and the cross sections; studying the latter might help us understand the differences in the former. Second, these cross

sections—not k-infinity—are the quantities that would be used later in the full-core analysis. Results are shown in Tables VIII and IX. (The ratio of changes in downscattering cross section is not shown; the ratios were in all cases between 0.99 and 1.00.)

Table VII. Ratio of branch  $\Delta k$ 's:  $\Delta k_{\text{HELIOS}} / \Delta k_{\text{CASMO}}$ .

Exposure (GWD/MT)	Branch 1 (High $T_F$ )	Branch 2 (High $T_M$ )	Branch 3 (Low $T_M$ )	Branch 4 (High B)	Branch 5 (Low B)
0	1.0297	1.5935	1.9782	0.9906	0.9611
5	1.0903	1.2353	5.2727	0.9825	0.9877
10	1.0376	1.3938	1.6163	0.9849	0.9819
20	1.0123	1.3597	1.2647	0.9843	0.9754
30	1.0097	1.4469	1.2296	0.9849	0.9718
40	0.9686	1.7746	0.8415	0.9928	0.9647
50	1.0057	1.9372	1.7917	0.9910	0.9669
60	1.0758	2.1935	-1.7037	0.9862	0.9723

Table VIII. Ratio of branch  $\Delta\Sigma$ 's and  $\Delta k$  for High Moderator Temperature Branch:  $\Delta x_{\text{HELIOS}} / \Delta x_{\text{CASMO}}$ .

Bu	Abs-1	Abs-2	nufiss-1	nufiss-2	k-inf
0	0.97	0.98	0.94	0.98	1.47
5	0.98	0.98	0.98	0.95	1.04
10	0.99	0.98	1.00	0.96	1.23
20	0.98	0.97	1.00	0.93	1.28
30	1.01	0.96	0.98	0.89	1.30
40	1.04	0.95	0.99	0.74	1.61
50	1.04	0.95	0.99	0.50	1.77
60	1.03	0.98	0.99	0.82	2.03

Table IX. Ratio of branch  $\Delta\Sigma$ 's and  $\Delta k$  for Low Moderator Temperature Branch:  $\Delta x_{\text{HELIOS}} / \Delta x_{\text{CASMO}}$ .

Bu	Abs-1	Abs-2	nufiss-1	nufiss-2	k-inf
0	0.97	0.99	0.91	0.96	1.9574
5	0.97	0.99	0.95	0.96	4.9200
10	0.98	0.98	1.00	0.94	1.6477
20	0.96	0.98	0.97	0.92	1.3043
30	0.95	0.97	0.95	0.88	1.3488
40	0.94	0.97	0.96	0.89	1.0494
50	0.94	0.96	0.96	0.20	3.0526
60	0.95	0.94	0.96	1.60	-1.6667

Two things stand out in Tables VIII and IX. First, the two codes agree *far* more closely on the change in each cross section than they do on the change in k-infinity. For example, in some cases the codes agree within 6% on the change in each cross section, but differ by >60% on the change in k-infinity. This seems somewhat contradictory at first, and led us to investigate further the relationship between changes in the cross sections and changes in k-infinity, as we shall discuss below. Second, there appears to be significant disagreement between the codes on the change in the group-2  $\nu\Sigma_2$  at high burnups. We shall discuss this disagreement first.

Table X shows the behavior of the relative change in fission cross sections ( $\Delta\Sigma/\Sigma$ ) for both codes and for high and low moderator temperature branches. A careful examination shows that at high burnups, where Tables VIII and IX showed that the ratio of the  $\Delta\nu\Sigma_2$ 's from the two codes is large, this ratio is approximately 0/0, because the change

in the cross section is very small (and in fact changes sign, passing through zero). Thus, we conclude that the two codes are actually calculating the changes in the two-group homogenized cross sections, as functions of changes in moderator temperature, remarkably consistently.

Table X. Cross sections changes for the Moderator-Temperature Branch cases.

Bu	Moderator (base-branch)/base				moderator (base-branch)/base			
	nufiss2				nufiss1			
	CASMO		HELIOS		CASMO		HELIOS	
	MH	ML	MH	ML	MH	ML	MH	ML
0	0.013809	-0.014267	0.013298	-0.013556	0.007617	-0.006525	0.007252	-0.006017
5	0.010387	-0.010700	0.009752	-0.010185	0.007398	-0.006319	0.007294	-0.006055
10	0.008288	-0.008721	0.007805	-0.008049	0.007273	-0.006090	0.007331	-0.006085
20	0.005654	-0.005847	0.005183	-0.005309	0.007514	-0.006424	0.007491	-0.006223
30	0.003746	-0.003885	0.003263	-0.003331	0.007949	-0.006813	0.007730	-0.006436
40	0.002056	-0.002056	0.001487	-0.001784	0.008156	-0.007042	0.008065	-0.006738
50	0.000506	-0.000422	0.000245	-0.000082	0.008622	-0.007422	0.008460	-0.007093
60	-0.001023	0.000930	-0.000804	0.001430	0.009084	-0.007783	0.008888	-0.007411

The next question is why the ratios of the  $\Delta k$ 's are so large when the changes in two-group cross sections are so small. Table X, which shows the actual relative cross-section changes ( $\Delta\Sigma/\Sigma$ ) for each code, offers a possible explanation for the discrepancies at late burnups. Note that the change in the group-2 fission cross section is very small at late burnups. Note further that each k calculation is iterative and that each code converges this iteration only to some pre-set tolerance. This iteration is on the k-eigenvalue itself as well as the scalar flux, and typically the fluxes are not converged as tightly as k. Because the cross sections come from the scalar fluxes that are generated during this iterative process, it follows that if the iteration is not converged tightly, the cross sections will not have many digits of precision. A small difference of two such numbers will of course be imprecise. This appears to be a likely cause for the large discrepancies in the two codes'  $\Delta k$ 's at late burnups.

However, because the changes in  $\Delta\nu\Sigma_f$  at earlier burnups are large enough to have some significant digits, the above argument does *not* seem to explain the large  $\Delta k_\infty$  ratio at those small burnups. To explore the reasons for this, we explored the relationship between k and the two-group cross sections. We found that if power iteration is used for the fine-group assembly-level k-eigenvalue calculation, and standard homogenization techniques are used for the two-group assembly-averaged cross sections, then there is a simple and direct relationship between the k-eigenvalue and the two-group cross sections, *regardless of how tightly the eigenvalue calculation is converged*. Details are in Appendix A; Eq. (8) from that appendix shows the relationship between the k-infinity *at the end of any iteration* and the two-group homogenized cross sections:

$$k^{(n+1)} = \frac{\overline{\nu\Sigma}_f \Phi_1 + \overline{\nu\Sigma}_f \frac{\overline{\Sigma}_{1 \rightarrow 2}^{net}}{\overline{\Sigma}_{a2}} \Phi_1}{\left[ \overline{\Sigma}_{a1} + \left( \overline{\Sigma}_{1 \rightarrow 2} - \overline{\Sigma}_{2 \rightarrow 1} \frac{\Phi_2}{\Phi_1} \right) \right] \Phi_1} = \frac{\overline{\nu\Sigma}_f}{\overline{\Sigma}_{a1} + \overline{\Sigma}_{1 \rightarrow 2}^{net}} + \frac{\overline{\Sigma}_{1 \rightarrow 2}^{net}}{\overline{\Sigma}_{a1} + \overline{\Sigma}_{1 \rightarrow 2}^{net}} \frac{\overline{\nu\Sigma}_f}{\overline{\Sigma}_{a2}} = \frac{\overline{\nu\Sigma}_f}{\overline{\Sigma}_{r1}} + \frac{\overline{\Sigma}_{1 \rightarrow 2}^{net}}{\overline{\Sigma}_{r1}} \frac{\overline{\nu\Sigma}_f}{\overline{\Sigma}_{a2}}. \quad (8)$$

We recognize that this result rests on some assumptions that are somewhat questionable. First, it assumes that power iteration is used as shown in Appendix A; in truth, CASMO and HELIOS do deviate somewhat from this "pure" power iteration. Second, both codes actually compute their k-infinities from a calculation that is performed after the two-dimensional transport k-eigenvalue calculation; this computation uses the two-group constants as input. Nevertheless, Appendix A is useful in that it shows the relationship in question in the limit that the codes' results approach the result of power iteration on the two-dimensional transport equation. To test whether the  $k_\infty$ 's reported by the two codes are in fact close to this limit, we employed Eq. (8) to calculate "two-group k" values and compared them against the reported k-infinity values. The results, taken from a typical LEU assembly, are shown in Table XI. Note that the CASMO results are reasonably self-consistent but the HELIOS ones are not.

Table XI. Reported  $k$ 's, 2-Group  $k$ 's, and their differences.

Burnup	HELIOS			CASMO		
	reported $k$	2-Group $k$	2G – report	Reported $k$	2-Group $k$	2G – report
0	1.13803	1.14054	<b>-0.00251</b>	1.13884	1.13842	<b>0.00042</b>
5	1.13813	1.14140	<b>-0.00327</b>	1.13827	1.13785	<b>0.00042</b>
10	1.13219	1.13546	<b>-0.00327</b>	1.13213	1.13171	<b>0.00042</b>
20	1.07706	1.08051	<b>-0.00345</b>	1.07705	1.07688	<b>0.00017</b>
30	1.00787	1.01142	<b>-0.00355</b>	1.00782	1.00781	<b>0.00001</b>
40	0.93993	0.94285	<b>-0.00292</b>	0.93938	0.93937	<b>0.00001</b>
50	0.87600	0.87880	<b>-0.00279</b>	0.87432	0.87416	<b>0.00016</b>
60	0.81893	0.82184	<b>-0.00291</b>	0.81618	0.81575	<b>0.00043</b>

After some investigation we found that the source of the HELIOS inconsistency is that the downscattering cross section it reports is *not* corrected [as in Eq. (7) of Appendix A] to be a *net* quantity. [The CASMO downscattering cross section is corrected as in Eq. (7).] It is possible to give ZENITH instructions to correct this cross section; the TAMU team developed and tested an input file that does this. After this correction, the HELIOS “two-group  $k$ ” is brought into reasonable agreement with the reported  $k$ , as shown in Table XII.

Table XII. Reported  $k$ 's, 2-Group  $k$ 's, and their differences, with corrected *net* downscattering.

Burnup	HELIOS			CASMO		
	reported $k$	2-Group $k$	2G – report	Reported $k$	2-Group $k$	2G – report
0	1.13803	1.13766	<b>0.00037</b>	1.13884	1.13842	<b>0.00042</b>
5	1.13813	1.13832	<b>-0.00019</b>	1.13827	1.13785	<b>0.00042</b>
10	1.13219	1.13206	<b>0.00013</b>	1.13213	1.13171	<b>0.00042</b>
20	1.07706	1.07682	<b>0.00024</b>	1.07705	1.07688	<b>0.00017</b>
30	1.00787	1.00765	<b>0.00022</b>	1.00782	1.00781	<b>0.00001</b>
40	0.93993	0.93909	<b>0.00084</b>	0.93938	0.93937	<b>0.00001</b>
50	0.87600	0.87508	<b>0.00092</b>	0.87432	0.87416	<b>0.00016</b>
60	0.81893	0.81821	<b>0.00072</b>	0.81618	0.81575	<b>0.00043</b>

Thus, this exercise has shown that **HELIOS users must take care to instruct ZENITH to output a *net* downscattering cross section**, unless the global two-group diffusion code will itself explicitly calculate upscattering. Without this correction, the reactivity of the assemblies in the global calculation would be artificially high by a few hundred pcm, as shown in the fourth column of Table XI, which could lead to **significant systematic errors** in the prediction of cycle length.

Another significant point is that **if HELIOS and CASMO disagree on MTC, then global calculations using the two codes’ cross section tables will also disagree on MTC**. This follows from the consistency between the two-group cross sections (which the global code uses to calculate  $k$ 's) and  $k$ 's reported by HELIOS and CASMO, from which their MTC estimates are computed.

The next task was to consider a branch case, in which there is a base  $k$  calculation and then another  $k$  calculation for a different (“branch”) state of the assembly. Our first aim was to determine the relationship between the *change in  $k$*  and the *change in the two-group cross sections*. Equation (8) of Appendix A makes this a relatively simple task, although the algebraic details are somewhat tedious. These algebraic details are shown in the second section of Appendix A. The final result is that for a given branch, the ratio of  $\Delta k$ 's from the two codes depends upon ratios of *combinations* of relative changes and also upon ratios of base-case quantities. Some of these combinations are actually *differences of differences* in cross sections. This leaves open the interesting possibility that the ratio of MTCs computed by the two codes can be *much* different than the ratio of the change in any cross section, even though (as we showed above) each  $k$  depends in a simple and direct way upon the two-group cross sections. In

Appendix A we use our derived expression to show that in fact this is what is happening in our comparisons. In other words, we now understand the mathematical reason that the two codes can agree closely on changes in the two-group constants while disagreeing on changes in  $k$ . The origin of this apparent contradiction is *not* related to iterative convergence, to errors in the codes, or to errors in input.

In summary, once the work described in this section (and Appendix A) was completed, we were confident that we understood how to properly use HELIOS to generate the desired cross section tables, and we were confident that the results would be reasonable.

On final check that we made compared the HELIOS 34-group and 89-group libraries. (ORNL researchers had discovered significant discrepancies between MOX results with the two libraries.) In Figure 3 and Table XIII we show  $k$ -infinities from four HELIOS runs: an LEU assembly and a MOX assembly, using the two different group structures for each. We see that the differences are small for the LEU assembly, but uncomfortably large for the MOX. We therefore decided that all MOX cross section tables to be given to NCSU would be generated using the 89-group library, and all LEU tables using the 34-group library.

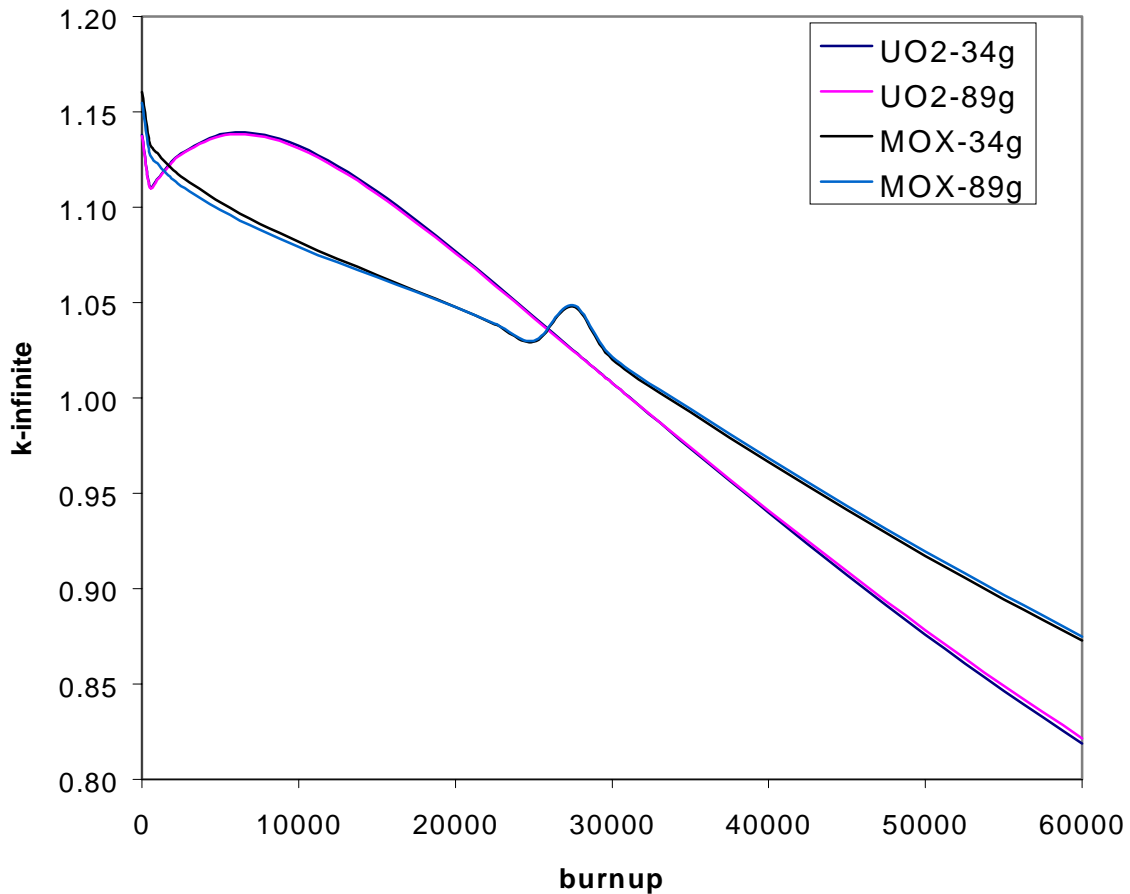


Figure 3. MOX and LEU results from HELIOS with different numbers of energy groups.



Table XIII. Comparison k-infinity given by HELIOS with different group libraries.

Burnup	UO <sub>2</sub> 4.5 w/o with 128 IFBA			MOX 4.5 w/o with 24 WABA		
	34groups	89 groups	% of difference	34groups	89groups	% of difference
0	1.13803	1.13734	0.060668	1.16063	1.15482	0.503109
500	1.11116	1.11060	0.050423	1.13392	1.12830	0.498094
1000	1.11524	1.11478	0.041264	1.12817	1.12272	0.485428
1500	1.12006	1.11964	0.037512	1.12359	1.11830	0.473039
2500	1.12813	1.12770	0.038131	1.11630	1.11134	0.446308
5000	1.13813	1.13740	0.064181	1.10266	1.09860	0.369561
7500	1.13840	1.13738	0.089680	1.09146	1.08825	0.294969
10000	1.13219	1.13098	0.106987	1.08175	1.07929	0.227928
12500	1.12171	1.12041	0.116029	1.07292	1.07116	0.164308
15000	1.10843	1.10713	0.117421	1.06456	1.06340	0.109084
17500	1.09332	1.09211	0.110795	1.05629	1.05568	0.057783
20000	1.07706	1.07599	0.099443	1.04779	1.04765	0.013363
22500	1.06013	1.05926	0.082133	1.03887	1.03913	-0.025021
25000	1.04283	1.04219	0.061409	1.02941	1.03002	-0.059222
27500	1.02536	1.02498	0.037074	1.04821	1.04877	-0.053396
30000	1.00787	1.00776	0.010915	1.02028	1.02151	-0.120410
35000	0.97362	0.97412	-0.051328	0.99266	0.99433	-0.167952
40000	0.93993	0.94105	-0.119016	0.96636	0.96833	-0.203443
45000	0.90728	0.90894	-0.182630	0.94129	0.94343	-0.226832
50000	0.87600	0.87815	-0.244833	0.91737	0.91962	-0.244666
55000	0.84644	0.84900	-0.301531	0.89453	0.89682	-0.255347
60000	0.81893	0.82177	-0.345595	0.87283	0.87512	-0.261678

### III.B. HELIOS/ZENITH Results

After we were confident in our proper use of the code and in its ability to deliver reasonable cross sections, we applied it to the assemblies listed in Table I to generate a large collection of cross-section files. These files were passed along to NCSU for use in full-core modeling.

All non-proprietary details of our geometric models are documented in our HELIOS input files, which are available electronically. We present some sample files in Appendices B, C, and D. Each assembly was a standard 17x17 array of pins of standard Westinghouse diameter and pitch. We employed both WABA and IFBA burnable absorbers, as shown in Table I. We used boron enriched to 40% in the coolant, but natural enrichment (19.6%) in the burnable absorbers. Central instrumentation tubes were filled with water.

HELIOS does not generate cross section files directly; its output is post-processed by the ZENITH code, whose output files constitute the two-group cross section libraries. We used the following naming convention for MOX files:

**zmXwY89gZ.out** ,

where:

X = enrichment = h or l for high (4.5 w/o fissile) or low (4.0 w/o fissile),  
 Y = number of WABA rods = h or l or n for 24 or 12 or none,  
 Z = case (branch) number = 1, 2, ..., 6

We use the following convention for uranium files:

**ZuXYiZ.out** ,

where:

X = enrichment = h or l or 35 for high (4.5 w/o) or low (4.0 w/o) or lower (3.5 w/o)  
 Y = number of IFBA rods = 16, 48, 64, 80, 104, 128 or n (none),  
 Z = case (branch) number = 1, 2, ..., 6

We use the following convention for baffle/reflector files:

**zmhwhrefZ.out**,

where:

Z = case (branch) number = 1, 2, ..., 5

Appendix E shows a ZENITH input file for a fuel assembly, and Appendix F contains a ZENITH file that produces cross sections for the reflector + baffle region.

## References

1. J. J. Cassal et al, "HELIOS: Geometric Capability of a New Fuel-Assembly Program," *Proc. Int. Topl. Mtg. Advances in Mathematics, Computations, and Reactor Physics*, Pittsburgh, Pennsylvania, April 28-May 2, 1991, p. 10.21-1 (1991).
2. E. A. Villarino, R. J. J. Stamm'ler, A. A. Ferri and J.J. Casal, "HELIOS: Angularly Dependent Collision Probabilities," *Nucl. Sci. Eng.* **112**, 16-31 (1992).
3. I. Ariani and P. J. Turinsky, "Evaluation of Mixed LEU-MOX Loading Patterns for LWRs Originating from the Inability to deliver MOX Assemblies (Westinghouse Reload Core Study)," ORNL/SUB/99-19XSY063V-2 (1999).
4. M. Edenius, K. S. Smith, and D. Ver Plank, "New Data and Methods for CASMO and SIMULATE," *Proc. Topl. Mtg. Advances in Reactor Physics and Safety*, Saratoga Springs, New York, September 17-19, Vol. 2, p. 1115 (1986). Published by U.S. Nuclear Regulatory Commission.
5. A. Ferri, "Generation and Contents of the ENDF/B-VI based HELIOS Library hy941a," Scandpower Technical Note TN2/44.70.13, 22 Nov. (1994).
6. P. F. Rose, "ENDF-201 ENDF/B-VI Summary Documentation," BNL-NCS-17541 (1992).
7. K. S. Smith, private communication. (1997).
8. A. A. Alsaed, "Disposition of Weapons-Grade Plutonium in Westinghouse Reactors," MS Thesis, Department of Nuclear Engineering, Texas A&M University (1996). See also A. A. Alsaed and M. L. Adams, "Disposition of Weapons-Grade Plutonium in Westinghouse Reactors," Amarillo National Resource Center Technical Report ANRCP-1998-1 (1998).
9. "Implementation of Weapons Grade MOX Fuel in Existing Pressurized Water Reactors," DOE/SF/19683-7 (1996).

## Appendix A: Relationship between $k$ and homogenized 2-group constants.

We begin with the equations that are used in an assembly-level transport calculation. If power iteration is used for the fine-group eigenvalue calculation, then:

$$\int_{\partial A} \hat{n} \cdot \vec{J}_g^{(n+1)}(\vec{r}) d^2 r + \int_A \left[ \Sigma_{a,g}(\vec{r}) + \sum_{g' \neq g} \Sigma_{s,g \rightarrow g'}(\vec{r}) \right] \phi_g^{(n+1)}(\vec{r}) d^3 r$$

$$= \int_A \left\{ \sum_{g' \neq g} \Sigma_{s,g' \rightarrow g}(\vec{r}) \phi_{g'}^{(n+1)}(\vec{r}) + \frac{1}{k^{(n)}} \chi_g \sum_{g'} v \Sigma_{f,g'}(\vec{r}) \phi_{g'}^{(n)}(\vec{r}) \right\} d^3 r, \quad g = 1, \dots, G, \quad (1a)$$

$$k^{(n+1)} = k^{(n)} \frac{\int_A \left\{ \sum_g v \Sigma_{f,g}(\vec{r}) \phi_g^{(n+1)}(\vec{r}) \right\} d^3 r}{\int_A \left\{ \sum_g v \Sigma_{f,g}(\vec{r}) \phi_g^{(n)}(\vec{r}) \right\} d^3 r} = \frac{\int_A \left\{ \sum_g v \Sigma_{f,g}(\vec{r}) \phi_g^{(n+1)}(\vec{r}) \right\} d^3 r}{\frac{1}{k^{(n)}} \int_A \left\{ \sum_g v \Sigma_{f,g}(\vec{r}) \phi_g^{(n)}(\vec{r}) \right\} d^3 r}. \quad (1b)$$

Here the integrals over  $A$  are over the assembly volume, and the integral over  $\partial A$  is over the assembly surface. The parenthetical superscript represents iteration index. Each surface is treated as a reflecting boundary; in many methods this reflection is enforced for each group at each iteration, making the surface term in Eq. (1a) zero. (This will often depend on how tightly the code converges its within-group iteration.) Equations (1) contain integrals and a continuous spatial variable  $r$ , but this is simply a notational convenience. In the actual calculation spatial quantities are discretized; nevertheless, they satisfy a form of Eqs. (1) in which spatial integrals are represented as sums of products of spatial-cell volumes times cell-averaged quantities.

If Eq. (1a) is summed over all “fast” groups the result is:

$$\int_A \sum_{fast\ g} \left[ \Sigma_{a,g}(\vec{r}) + \sum_{g' \neq g} \Sigma_{s,g \rightarrow g'}(\vec{r}) \right] \phi_g^{(n+1)}(\vec{r}) d^3 r$$

$$= \int_A \left\{ \sum_{fast\ g} \sum_{g' \neq g} \Sigma_{s,g' \rightarrow g}(\vec{r}) \phi_{g'}^{(n+1)}(\vec{r}) + \frac{1}{k^{(n)}} \sum_{fast\ g} \chi_g \sum_{g'} v \Sigma_{f,g'}(\vec{r}) \phi_{g'}^{(n)}(\vec{r}) \right\} d^3 r. \quad (2a)$$

Many of the scattering terms now cancel from the right and left sides of the equation; the only ones remaining are those that involve scattering between “fast” and “thermal” groups:

$$\int_A \sum_{fast\ g} \left[ \Sigma_{a,g}(\vec{r}) + \sum_{thermal\ g'} \Sigma_{s,g \rightarrow g'}(\vec{r}) \right] \phi_g^{(n+1)}(\vec{r}) d^3 r$$

$$= \int_A \left\{ \sum_{fast\ g} \sum_{thermal\ g'} \Sigma_{s,g' \rightarrow g}(\vec{r}) \phi_{g'}^{(n+1)}(\vec{r}) + \frac{1}{k^{(n)}} \sum_{fast\ g} \chi_g \sum_{g'} v \Sigma_{f,g'}(\vec{r}) \phi_{g'}^{(n)}(\vec{r}) \right\} d^3 r. \quad (2b)$$

Now we shall assume that  $\chi_g$  is zero for thermal groups, and thus sums to 1 over the fast groups. Then the “fast” equation becomes:

$$\begin{aligned} & \int_A \sum_{fast\ g} \left[ \Sigma_{a,g}(\bar{r}) + \sum_{thermal\ g'} \Sigma_{s,g \rightarrow g'}(\bar{r}) \right] \phi_g^{(n+1)}(\bar{r}) d^3 r \\ & = \int_A \left\{ \sum_{fast\ g} \sum_{thermal\ g'} \Sigma_{s,g' \rightarrow g}(\bar{r}) \phi_{g'}^{(n+1)}(\bar{r}) + \frac{1}{k^{(n)}} \sum_{g'} v \Sigma_{f,g'}(\bar{r}) \phi_{g'}^{(n)}(\bar{r}) \right\} d^3 r. \end{aligned} \quad (3a)$$

Under the same assumptions, summing Eq. (1a) over all thermal groups yields:

$$\int_A \sum_{thermal\ g} \left[ \Sigma_{a,g}(\bar{r}) + \sum_{fast\ g'} \Sigma_{s,g \rightarrow g'}(\bar{r}) \right] \phi_g^{(n+1)}(\bar{r}) d^3 r = \int_A \sum_{thermal\ g} \sum_{fast\ g'} \Sigma_{s,g' \rightarrow g}(\bar{r}) \phi_{g'}^{(n+1)}(\bar{r}) d^3 r. \quad (3b)$$

The second term on the right-hand side of Eq. (3a) is the denominator of the right-hand side of Eq. (1b). Substitute this into Eq. (1b):

$$k^{(n+1)} = \frac{\int_A \left\{ \sum_g v \Sigma_{f,g}(\bar{r}) \phi_g^{(n+1)}(\bar{r}) \right\} d^3 r}{\int_A \left\{ \sum_{fast\ g} \left[ \Sigma_{a,g}(\bar{r}) + \sum_{thermal\ g'} \Sigma_{s,g \rightarrow g'}(\bar{r}) \right] \phi_g^{(n+1)}(\bar{r}) - \sum_{thermal\ g} \sum_{fast\ g'} \Sigma_{s,g' \rightarrow g}(\bar{r}) \phi_{g'}^{(n+1)}(\bar{r}) \right\} d^3 r}. \quad (4)$$

It is important to note that **this equation is satisfied at the end of each iteration, regardless of whether or not the iteration has converged.**

Now we recall the standard definitions of assembly-averaged two-group scalar fluxes and cross sections:

$$\Phi_1 \equiv \frac{\int_A \sum_{fast\ g} \phi_g^{(n+1)}(\bar{r}) d^3 r}{\int_A d^3 r} = \frac{1}{A} \int_A \sum_{fast\ g} \phi_g^{(n+1)}(\bar{r}) d^3 r, \quad (5a)$$

where  $A$  is the assembly volume,

$$\Phi_2 \equiv \frac{1}{A} \int_A \sum_{thermal\ g} \phi_g^{(n+1)}(\bar{r}) d^3 r, \quad (5b)$$

$$\bar{\Sigma}_{a1} \equiv \frac{\int_A \sum_{fast\ g} \Sigma_{a,g}(\bar{r}) \phi_g^{(n+1)}(\bar{r}) d^3 r}{\int_A \sum_{fast\ g} \phi_g^{(n+1)}(\bar{r}) d^3 r} = \frac{1}{\Phi_1 A} \int_A \sum_{fast\ g} \Sigma_{a,g}(\bar{r}) \phi_g^{(n+1)}(\bar{r}) d^3 r, \quad (5c)$$

$$\bar{\Sigma}_{a2} \equiv \frac{\int_A \sum_{thermal\ g} \Sigma_{a,g}(\bar{r}) \phi_g^{(n+1)}(\bar{r}) d^3 r}{\int_A \sum_{thermal\ g} \phi_g^{(n+1)}(\bar{r}) d^3 r} = \frac{1}{\Phi_2 A} \int_A \sum_{thermal\ g} \Sigma_{a,g}(\bar{r}) \phi_g^{(n+1)}(\bar{r}) d^3 r, \quad (5d)$$

with very similar expressions for the average  $v\Sigma_{f1}$  and  $v\Sigma_{f2}$ , and finally

$$\bar{\Sigma}_{1 \rightarrow 2} \equiv \frac{\int_A \sum_{fast\ g} \sum_{thermal\ g'} \Sigma_{s,g \rightarrow g'}(\bar{r}) \phi_g^{(n+1)}(\bar{r}) d^3 r}{\int_A \sum_{fast\ g} \phi_g^{(n+1)}(\bar{r}) d^3 r} = \frac{1}{\Phi_1 A} \int_A \sum_{fast\ g} \sum_{thermal\ g'} \Sigma_{s,g \rightarrow g'}(\bar{r}) \phi_g^{(n+1)}(\bar{r}) d^3 r, \quad (5e)$$

$$\bar{\Sigma}_{2 \rightarrow 1} \equiv \frac{\int_A \sum_{thermal\ g} \sum_{fast\ g'} \Sigma_{s,g \rightarrow g'}(\bar{r}) \phi_g^{(n+1)}(\bar{r}) d^3 r}{\int_A \sum_{thermal\ g} \phi_g^{(n+1)}(\bar{r}) d^3 r} = \frac{1}{\Phi_2 A} \int_A \sum_{thermal\ g} \sum_{fast\ g'} \Sigma_{s,g \rightarrow g'}(\bar{r}) \phi_g^{(n+1)}(\bar{r}) d^3 r. \quad (5f)$$

Now if we use all of this in Eq. (4) we find:

$$\begin{aligned} k^{(n+1)} &= \frac{\int_A \left\{ \sum_g v\Sigma_{f,g}(\bar{r}) \phi_g^{(n+1)}(\bar{r}) \right\} d^3 r}{\int_A \left\{ \sum_{fast\ g} \left[ \Sigma_{a,g}(\bar{r}) + \sum_{thermal\ g'} \Sigma_{s,g \rightarrow g'}(\bar{r}) \right] \phi_g^{(n+1)}(\bar{r}) - \sum_{thermal\ g} \sum_{fast\ g'} \Sigma_{s,g \rightarrow g'}(\bar{r}) \phi_g^{(n+1)}(\bar{r}) \right\} d^3 r}. \\ &= \frac{A v \bar{\Sigma}_{f1} \Phi_1 + A v \bar{\Sigma}_{f2} \Phi_2}{A [\bar{\Sigma}_{a1} + \bar{\Sigma}_{1 \rightarrow 2}] \Phi_1 - A \bar{\Sigma}_{2 \rightarrow 1} \Phi_2} \end{aligned} \quad (6)$$

In light of the definitions of assembly-averaged fluxes and cross sections, note now that Eq. (3b) becomes:

$$A(\bar{\Sigma}_{a2} + \bar{\Sigma}_{2 \rightarrow 1})\Phi_2 = A\bar{\Sigma}_{1 \rightarrow 2}\Phi_1 \Rightarrow \bar{\Sigma}_{a2}\Phi_2 = \left( \bar{\Sigma}_{1 \rightarrow 2} - \bar{\Sigma}_{2 \rightarrow 1} \frac{\Phi_2}{\Phi_1} \right) \Phi_1 \Rightarrow \Phi_2 = \frac{\bar{\Sigma}_{1 \rightarrow 2}^{net}}{\bar{\Sigma}_{a2}} \Phi_1,$$

where we have defined a ‘‘net’’ downscattering cross section:

$$\bar{\Sigma}_{1 \rightarrow 2}^{net} \equiv \left( \bar{\Sigma}_{1 \rightarrow 2} - \bar{\Sigma}_{2 \rightarrow 1} \frac{\Phi_2}{\Phi_1} \right). \quad (7)$$

Now Eq. (6) can be rewritten:

$$\begin{aligned}
k^{(n+1)} &= \frac{\overline{v\Sigma} f_1 \Phi_1 + \overline{v\Sigma} f_2 \frac{\overline{\Sigma}_{1 \rightarrow 2}^{net}}{\overline{\Sigma}_{a2}} \Phi_1}{\left[ \overline{\Sigma}_{a1} + \left( \overline{\Sigma}_{1 \rightarrow 2} - \overline{\Sigma}_{2 \rightarrow 1} \frac{\Phi_2}{\Phi_1} \right) \right] \Phi_1} = \frac{\overline{v\Sigma} f_1}{\overline{\Sigma}_{a1} + \overline{\Sigma}_{1 \rightarrow 2}^{net}} + \frac{\overline{\Sigma}_{1 \rightarrow 2}^{net}}{\overline{\Sigma}_{a1} + \overline{\Sigma}_{1 \rightarrow 2}^{net}} \frac{\overline{v\Sigma} f_2}{\overline{\Sigma}_{a2}} \\
&= \frac{\overline{v\Sigma} f_1}{\overline{\Sigma}_{r1}} + \frac{\overline{\Sigma}_{1 \rightarrow 2}^{net}}{\overline{\Sigma}_{r1}} \frac{\overline{v\Sigma} f_2}{\overline{\Sigma}_{a2}}.
\end{aligned} \tag{8}$$

[Here we have defined the fast-group *removal* cross section,  $\Sigma_{r1}$ , to be the sum of the absorption plus net downscattering cross sections.] Equation (8) is our first main result:

**Regardless of how tightly the k-iteration is converged, Eq. (8) holds as a relationship between  $k$  and the two-group constants.**

This is true as long as power iteration is employed in the  $k$  iteration, with Eq. (1b) used as the update equation for  $k$ .

The next task was to consider a branch case, in which there is a base  $k$  calculation and then another  $k$  calculation for a different (“branch”) state of the assembly. Here we try to determine the relationship between the *change in  $k$*  and the *change in the two-group cross sections*. Equation (8) makes this a relatively simple task, although the algebraic details are somewhat tedious. The change in  $k$  on a branch is:

$$k_b - k = \left[ \frac{\overline{v\Sigma} f_1}{\overline{\Sigma}_{r1}} + \frac{\overline{\Sigma}_{1 \rightarrow 2}^{net}}{\overline{\Sigma}_{r1}} \frac{\overline{v\Sigma} f_2}{\overline{\Sigma}_{a2}} \right]_b - \frac{\overline{v\Sigma} f_1}{\overline{\Sigma}_{r1}} - \frac{\overline{\Sigma}_{1 \rightarrow 2}^{net}}{\overline{\Sigma}_{r1}} \frac{\overline{v\Sigma} f_2}{\overline{\Sigma}_{a2}}, \tag{9}$$

where the subscript  $b$  represents the “branch” state. Now let us define the change in each group cross section:

$$\delta_{f1} \equiv \left[ \overline{v\Sigma} f_1 \right]_b - \overline{v\Sigma} f_1, \quad \delta_{f2} \equiv \left[ \overline{v\Sigma} f_2 \right]_b - \overline{v\Sigma} f_2, \tag{10a}$$

$$\delta_{a1} \equiv \left[ \overline{\Sigma}_{a1} \right]_b - \overline{\Sigma}_{a1}, \quad \delta_{a2} \equiv \left[ \overline{\Sigma}_{a2} \right]_b - \overline{\Sigma}_{a2}, \tag{10b}$$

$$\delta_{12} \equiv \left[ \overline{\Sigma}_{1 \rightarrow 2}^{net} \right]_b - \overline{\Sigma}_{1 \rightarrow 2}^{net}, \quad \delta_{r1} \equiv \left[ \overline{\Sigma}_{r1} \right]_b - \overline{\Sigma}_{r1}. \tag{10c}$$

Similarly, we define **relative changes** in the cross sections:

$$\varepsilon_{f1} \equiv \delta_{f1} / \overline{v\Sigma} f_1, \quad \varepsilon_{f2} \equiv \delta_{f2} / \overline{v\Sigma} f_2, \quad \varepsilon_{12} \equiv \delta_{12} / \overline{\Sigma}_{1 \rightarrow 2}^{net}, \tag{11a}$$

$$\varepsilon_x \equiv \delta_x / \overline{\Sigma}_x, \quad x = a1, a2, \text{ or } r1. \tag{11b}$$

Now we can manipulate Eq. (9) as follows.

$$\begin{aligned}
k_b - k &= \frac{1}{\overline{\Sigma}_{r1}} \left[ \frac{\overline{v\Sigma} f_1 (1 + \varepsilon_{f1})}{1 + \varepsilon_{r1}} - \overline{v\Sigma} f_1 + \frac{\overline{\Sigma}_{1 \rightarrow 2}^{net} (1 + \varepsilon_{12})}{1 + \varepsilon_{r1}} \frac{\overline{v\Sigma} f_2 (1 + \varepsilon_{f2})}{\overline{\Sigma}_{a2} (1 + \varepsilon_{a2})} - \overline{\Sigma}_{1 \rightarrow 2}^{net} \frac{\overline{v\Sigma} f_2}{\overline{\Sigma}_{a2}} \right] \\
&= \frac{1}{\overline{\Sigma}_{r1}} \left[ \overline{v\Sigma} f_1 \left( \frac{(1 + \varepsilon_{f1})}{1 + \varepsilon_{r1}} - 1 \right) + \overline{\Sigma}_{1 \rightarrow 2}^{net} \frac{\overline{v\Sigma} f_2}{\overline{\Sigma}_{a2}} \left( \frac{(1 + \varepsilon_{12}) (1 + \varepsilon_{f2})}{(1 + \varepsilon_{r1}) (1 + \varepsilon_{a2})} - 1 \right) \right] \\
&= \frac{\overline{v\Sigma} f_1}{\overline{\Sigma}_{r1}} \left( \frac{\varepsilon_{f1} - \varepsilon_{r1}}{1 + \varepsilon_{r1}} \right) + \frac{\overline{\Sigma}_{1 \rightarrow 2}^{net}}{\overline{\Sigma}_{r1}} \frac{\overline{v\Sigma} f_2}{\overline{\Sigma}_{a2}} \left( \frac{\varepsilon_{12} + \varepsilon_{f2} + \varepsilon_{f2} \varepsilon_{12} - \varepsilon_{r1} - \varepsilon_{a2} - \varepsilon_{r1} \varepsilon_{a2}}{(1 + \varepsilon_{r1})(1 + \varepsilon_{a2})} \right)
\end{aligned} \tag{12}$$

Let us define  $\alpha$ , a number between 0 and 1, as follows:

$$\alpha k = \frac{\overline{\nu \Sigma_{f1}}}{\overline{\Sigma_{r1}}} \text{ and } (1-\alpha)k = \frac{\overline{\Sigma_{1 \rightarrow 2}^{net}}}{\overline{\Sigma_{r1}}} \frac{\overline{\nu \Sigma_{f2}}}{\overline{\Sigma_{a2}}}. \quad (13)$$

Then Eq. (12) becomes

$$k_b - k = \alpha k \left( \frac{\varepsilon_{f1} - \varepsilon_{r1}}{1 + \varepsilon_{r1}} \right) + (1-\alpha)k \left( \frac{\varepsilon_{12} + \varepsilon_{f2} + \varepsilon_{f2}\varepsilon_{12} - \varepsilon_{r1} - \varepsilon_{a2} - \varepsilon_{r1}\varepsilon_{a2}}{(1 + \varepsilon_{r1})(1 + \varepsilon_{a2})} \right), \quad (14)$$

or

$$(k_b - k) = k \frac{\left[ \alpha (\varepsilon_{f1} - \varepsilon_{r1})(1 + \varepsilon_{a2}) + (1-\alpha) (\varepsilon_{12} + \varepsilon_{f2} + \varepsilon_{f2}\varepsilon_{12} - \varepsilon_{r1} - \varepsilon_{a2} - \varepsilon_{r1}\varepsilon_{a2}) \right]}{(1 + \varepsilon_{r1})(1 + \varepsilon_{a2})}. \quad (15)$$

It follows that the ratio of the HELIOS branch  $\Delta k$  to the CASMO branch  $\Delta k$  is:

$$\frac{(k_b - k)_H}{(k_b - k)_C} = \frac{k_H}{k_C} \frac{(1 + \varepsilon_{r1})_C (1 + \varepsilon_{a2})_C}{(1 + \varepsilon_{r1})_H (1 + \varepsilon_{a2})_H} \left[ \frac{\left( \begin{array}{c} \alpha_H (\varepsilon_{f1} - \varepsilon_{r1})_H (1 + \varepsilon_{a2})_H \\ + (1-\alpha)_H (\varepsilon_{12} + \varepsilon_{f2} + \varepsilon_{f2}\varepsilon_{12} - \varepsilon_{r1} - \varepsilon_{a2} - \varepsilon_{r1}\varepsilon_{a2})_H \end{array} \right)}{\left( \begin{array}{c} \alpha_C (\varepsilon_{f1} - \varepsilon_{r1})_C (1 + \varepsilon_{a2})_C \\ + (1-\alpha)_C (\varepsilon_{12} + \varepsilon_{f2} + \varepsilon_{f2}\varepsilon_{12} - \varepsilon_{r1} - \varepsilon_{a2} - \varepsilon_{r1}\varepsilon_{a2})_C \end{array} \right)} \right]. \quad (16)$$

Intuitively, one might have expected that the ratio  $(k_b - k)_H / (k_b - k)_C$  on the left side of the equation would depend fundamentally upon ratios of changes in cross sections, such as  $(\delta_{a2})_H / (\delta_{a2})_C$ , or perhaps upon ratios of relative changes in cross sections, such as  $(\varepsilon_{a2})_H / (\varepsilon_{a2})_C$ . However, an examination of the right side of equation (16) suggests that this is not the case — instead, the  $\Delta k$  ratio depends upon ratios of **combinations** of relative changes and also upon ratios of base-case quantities. Some of these combinations are actually **differences** of relative changes in cross sections. This leaves open the interesting possibility that the ratio of MTCs computed by the two codes can be *much* different than the ratio of the change (between branch and base cases) in any cross section, even though (as we showed above) each  $k$  depends in a simple and direct way upon the two-group cross sections. This phenomenon is observed in Table II above. For example, at zero burnup this table shows that the ratio  $(k_b - k)_H / (k_b - k)_C$  is 1.95, even though all of the ratios of cross section changes,  $(\delta_x)_H / (\delta_x)_C$ , are between 0.91 and 1.0.

To verify that the MTC ratios reported in Table XII are in fact consistent with the two-group cross sections that the codes are outputting, we have selected a representative LEU assembly and a representative burnup step (40 GWD/MTU) and carefully examined the two-group cross sections from the two codes. For the chosen case we find the following ratios of changes in two-group cross sections:

$$(\delta_{f1})_H / (\delta_{f1})_C = 0.9642 \quad (17a)$$

$$(\delta_{f2})_H / (\delta_{f2})_C = 0.9167 \quad (17b)$$

$$(\delta_{a1})_H / (\delta_{a1})_C = 0.9365 \quad (17c)$$

$$(\delta_{a2})_H / (\delta_{a2})_C = 0.9797 \quad (17d)$$

$$(\delta_{r1})_H / (\delta_{r1})_C = 0.9924 \quad (17e)$$

$$(\delta_{12})_H / (\delta_{12})_C = 0.9991 \quad (17f)$$

Note that the two codes agree on most of the cross section changes to within 4%, and on all of them to within 9%. We now calculate the terms that appear in Eq. (16):

$$\frac{k_H}{k_C} = 1.0006 \quad (18a)$$

$$\alpha_H (\varepsilon_{f1} - \varepsilon_{r1})_H (1 + \varepsilon_{a2})_H = -0.00786 \quad (18b)$$

$$\alpha_C (\varepsilon_{f1} - \varepsilon_{r1})_C (1 + \varepsilon_{a2})_C = -0.00770 \quad (18c)$$

$$(1 - \alpha)_H (\varepsilon_{12} + \varepsilon_{f2} + \varepsilon_{f2} \varepsilon_{12} - \varepsilon_{r1} - \varepsilon_{a2} - \varepsilon_{r1} \varepsilon_{a2})_H = 0.01033 \quad (18d)$$

$$(1 - \alpha)_C (\varepsilon_{12} + \varepsilon_{f2} + \varepsilon_{f2} \varepsilon_{12} - \varepsilon_{r1} - \varepsilon_{a2} - \varepsilon_{r1} \varepsilon_{a2})_C = 0.00923 \quad (18e)$$

$$\frac{\alpha_H (\varepsilon_{f1} - \varepsilon_{r1})_H (1 + \varepsilon_{a2})_H + (1 - \alpha)_H (\varepsilon_{12} + \varepsilon_{f2} + \varepsilon_{f2} \varepsilon_{12} - \varepsilon_{r1} - \varepsilon_{a2} - \varepsilon_{r1} \varepsilon_{a2})_H}{\alpha_C (\varepsilon_{f1} - \varepsilon_{r1})_C (1 + \varepsilon_{a2})_C + (1 - \alpha)_C (\varepsilon_{12} + \varepsilon_{f2} + \varepsilon_{f2} \varepsilon_{12} - \varepsilon_{r1} - \varepsilon_{a2} - \varepsilon_{r1} \varepsilon_{a2})_C} = \frac{0.002462}{0.001530} = 1.61. \quad (18f)$$

$$\frac{(1 + \varepsilon_{r1})_C (1 + \varepsilon_{a2})_C}{(1 + \varepsilon_{r1})_H (1 + \varepsilon_{a2})_H} = 1.0004 \quad (18g)$$

If we insert these numbers into Eq. (16) we obtain:

$$\frac{(k_b - k)_H}{(k_b - k)_C} = (1.0006)(1.0004)(1.61) = 1.61. \quad (19)$$

There are several things worth noting. First, the two codes agree to within a few percent on the change in each cross section as a function of the change in temperature. Second, even the *differences* of the relative changes in cross section are in reasonable agreement – Eqs. (18b) and (18c) agree within 2% and Eqs. (18d) and (18e) agree within 11%. Third, the values of (18b) and (18c) are negative whereas those of (18d) and (18e) are positive; thus, when the terms are combined as in (18f), we basically have a *difference of differences of relative changes*, and this is where the codes do not agree. This results in MTCs that differ by more than 60%.

## Appendix B. HELIOS Input File: MOX, high fissile, 24 WABA, 89 groups (hmhwh89h.inp)

```

+THEL
Vmhwh89h = CASE('hy8918-961a.dat' / 'Vmhwh89h.hrf' / 'MOX 4.5 24 waba')

! Data including thermal expansion      !
!fuel!
$rppe = PAR("0.3950905")
$rrair = PAR("0.4010399")
$rrcan = PAR("0.4582740")
$rrifba = PROPIETARY 1
!the radius was calculated preserving the ifba area!
$rrcool = PAR("0.5624394")
$rrbox = PAR("0.6032221")
$rrcri = PAR("0.4331")
$rrcrg = PAR("0.4369")
$rrcro = PAR("0.4848823")
!thermal expansion included for waba rod!
$rrwco = PROPIETARY 2
$rrwbox = PROPIETARY 3
$rrwair = PROPIETARY 4
$rrwaba = PROPIETARY 5
$rrwabair = PROPIETARY 6
$rrgap = PROPIETARY 7

$rpitch = PAR("1.2598")
$rp2 = PAR("$rpitch/2")
$rp2sq = PAR("$rrcan/2**0.5")
$rp2sqh = PAR("$rrbox/2**0.5")
$rp2gap = PAR("$rrgap/2")
$rcoup = PAR(3)

! Materials !
'H2O-1' = MAT(NB/.705/1001,11.19;8016,88.81;5010,0.04;5011,0.06)
'H2O-2' = MAT(NB/.705/1001,11.19;8016,88.81;5010,0.0;5011,0.0)
'H2O-3' = MAT(NB/.705/1001,11.19;8016,88.81;5010,0.08;5011,0.12)
'H2O-4' = MAT(NB/.743/1001,11.19;8016,88.81;5010,0.04;5011,0.06)
'H2O-5' = MAT(NB/.662/1001,11.19;8016,88.81;5010,0.04;5011,0.06)

Zy2 = MAT(NB/6.504/40002,100.0)

'PU-1' = MAT(10.41/92235,1.763E-1;94238,0.0; 94239,3.94991;94240,
0.248936;94241,1.68366E-02;94242,4.23120E-03;
92234,1.41040E-03;92238,83.7524;8016,11.85)
'PU-2' = MAT(10.41/92235,1.763E-1;94238,0.0; 94239,3.94991;94240,
0.248936;94241,1.68366E-02;94242,4.23120E-03;
92234,1.41040E-03;92238,83.7524;8016,11.85)
'boron1' = MAT(3.563548/8016,42.5262;6000,2.1004;13027,47.8109;
5010,1.3938;5011,6.1687)
'boron2' = MAT(3.563548/8016,42.5262;6000,2.1004;13027,47.8109;

```

```

                    5010,1.3938;5011,6.1687)
'air'    = MAT(NB/.001/8016,100)

! Geometry !
Pin      = CCS($rpe,$rair,$rcan//fuel,gap,clad)
Whole    = CCS($rcool,$rbox//water,clad)
Ipin     = CCS($rpe,$rifba,$rair,$rcan//fuel,ifba,gap,clad)
Bpin     = CCS($rwo,$rwo,$rwair,$rwaba,$rwabair,$rcro,
              $rcool,$rbox//water,clad,gap,crod,gap,clad,water,clad)

$cell    = PAR(("-$p2", "-$p2") ("-$p2", $p2) ($p2, $p2) ($p2, "-$p2") !nodes 1-4!
(0, "-$p2") ("-$p2", 0) (0, $p2) ($p2, 0) ($rcan, 0) (0, "-$rcan") ("-$rcan", 0)
(0, $rcan) ($psq2, $psq2) ($psq2, "-$psq2") ("-$psq2", "-$psq2") ("-$psq2", $psq2)
/4, cool/Pin(0, 0)/ 1, 6, 11, 15, cool; 6, 2, 16, 11, cool; 2, 7, 12, 16, cool;
7, 3, 13, 12, cool; 3, 8, 9, 13, cool; 8, 4, 14, 9, cool; 4, 5, 10, 14, cool)

$icell   = PAR(("-$p2", "-$p2") ("-$p2", $p2) ($p2, $p2) ($p2, "-$p2") !nodes 1-4!
(0, "-$p2") ("-$p2", 0) (0, $p2) ($p2, 0) ($rcan, 0) (0, "-$rcan") ("-$rcan", 0)
(0, $rcan) ($psq2, $psq2) ($psq2, "-$psq2") ("-$psq2", "-$psq2") ("-$psq2", $psq2)
/4, cool/Ipin(0, 0)/ 1, 6, 11, 15, cool; 6, 2, 16, 11, cool;
2, 7, 12, 16, cool; 7, 3, 13, 12, cool; 3, 8, 9, 13, cool; 8, 4, 14, 9, cool; 4, 5, 10, 14, cool)

$wcell   = PAR(("-$p2", "-$p2") ("-$p2", $p2) ($p2, $p2) ($p2, "-$p2") !nodes 1-4!
(0, "-$p2") ("-$p2", 0) (0, $p2) ($p2, 0) ($rbox, 0) (0, "-$rbox") ("-$rbox", 0)
(0, $rbox) ($psqh2, $psqh2) ($psqh2, "-$psqh2") ("-$psqh2", "-$psqh2")
("-$psqh2", $psqh2)/4, cool/Whole(0, 0)/ 1, 6, 11, 15, cool; 6, 2, 16, 11, cool;
2, 7, 12, 16, cool; 7, 3, 13, 12, cool; 3, 8, 9, 13, cool; 8, 4, 14, 9, cool; 4, 5, 10, 14, cool)

$bcell   = PAR(("-$p2", "-$p2") ("-$p2", $p2) ($p2, $p2) ($p2, "-$p2") !nodes 1-4!
(0, "-$p2") ("-$p2", 0) (0, $p2) ($p2, 0) ($rbox, 0) (0, "-$rbox") ("-$rbox", 0)
(0, $rbox) ($psqh2, $psqh2) ($psqh2, "-$psqh2") ("-$psqh2", "-$psqh2")
("-$psqh2", $psqh2)/4, cool/Bpin(0, 0)/ 1, 6, 11, 15, cool; 6, 2, 16, 11, cool;
2, 7, 12, 16, cool; 7, 3, 13, 12, cool; 3, 8, 9, 13, cool; 8, 4, 14, 9, cool; 4, 5, 10, 14, cool)

$wgap    = PAR(("-$pgap2", "-$p2") ("-$pgap2", $p2) ($pgap2, $p2) ($pgap2, "-$p2")
/4, cool//)

$cell2   = PAR(("-$p2", "-$p2") ("-$p2", 0) ($p2, 0) ($p2, "-$p2")
(0, "-$p2") ($rcan, 0) (0, "-$rcan") ("-$rcan", 0) ($psq2, "-$psq2")
("-$psq2", "-$psq2") /4, cool/Pin(0, 0) 3/ 3, 4, 9, 6, cool; 4, 5, 7, 9, cool;
5, 1, 10, 7, cool)

$icell2  = PAR(("-$p2", "-$p2") ("-$p2", 0) ($p2, 0) ($p2, "-$p2")
(0, "-$p2") ($rcan, 0) (0, "-$rcan") ("-$rcan", 0) ($psq2, "-$psq2")
("-$psq2", "-$psq2") /4, cool/Ipin(0, 0) 4/ 3, 4, 9, 6, cool; 4, 5, 7, 9, cool;
5, 1, 10, 7, cool)

$wcell2  = PAR(("-$p2", "-$p2") ("-$p2", 0) ($p2, 0) ($p2, "-$p2")
(0, "-$p2") ($rbox, 0) (0, "-$rbox") ("-$rbox", 0) ($psqh2, "-$psqh2")
("-$psqh2", "-$psqh2") /4, cool/Whole(0, 0) 2/3, 4, 9, 6, cool; 4, 5, 7, 9, cool;
5, 1, 10, 7, cool)

$bcell2  = PAR(("-$p2", "-$p2") ("-$p2", 0) ($p2, 0) ($p2, "-$p2")
(0, "-$p2") ($rbox, 0) (0, "-$rbox") ("-$rbox", 0) ($psqh2, "-$psqh2")
("-$psqh2", "-$psqh2") /4, cool/Bpin(0, 0) 8/3, 4, 9, 6, cool; 4, 5, 7, 9, cool;
5, 1, 10, 7, cool)

```

```
$wgap2 = PAR(("-$pgap2", "-$p2") ("-$pgap2", 0) ($pgap2, 0) ($pgap2, "-$p2")
/4, cool//)
```

```
$wcell4 = PAR((0, "-$p2") (0, 0) ($p2, 0) ($p2, "-$p2")
($rbox, 0) (0, "-$rbox") ($psqh2, "-$psqh2")
/4, cool/Whole(0, 0) 2/4, 1, 6, 7, cool)
```

```
$wgap4 = PAR(("-$pgap2", "-$pgap2") ("-$pgap2", $pgap2) ($pgap2, $pgap2 )
($pgap2, "-$pgap2") /4, cool//)
```

```
'1.0' = STR($cell)
'2.0' = STR($icell)
'3.0' = STR($wcell)
'4.0' = STR($bcell)
'5.0' = STR($cell2)
'6.0' = STR($icell2)
'7.0' = STR($wcell2)
'8.0' = STR($bcell2)
'9.0' = STR($wcell4)
'10.0' = STR($wgap)
'11.0' = STR($wgap2)
'12.0' = STR($wgap4)
```

```
White = ALB(1/1/1)
```

```
$scorder = PAR((1, 3, 4)$coup(2, 2, 1)/
(2, 3, 4)$coup(3, 2, 1)/(3, 3, 4)$coup(4, 2, 1)/
(4, 3, 4)$coup(5, 2, 1)/(5, 3, 4)$coup(6, 2, 1)/
(6, 3, 4)$coup(7, 2, 1)/(7, 3, 4)$coup(8, 2, 1)/
(8, 3, 4)$coup(9, 2, 1)/(9, 3, 4)$coup(10, 2, 1))
```

```
$scorder1 = PAR((1, 4, 1)$coup(2, 2, 1)/
(2, 3, 4)$coup(3, 2, 1)/(3, 3, 4)$coup(4, 2, 1)/
(4, 3, 4)$coup(5, 2, 1)/(5, 3, 4)$coup(6, 2, 1)/
(6, 3, 4)$coup(7, 2, 1)/(7, 3, 4)$coup(8, 2, 1)/
(8, 3, 4)$coup(9, 2, 1)/(9, 3, 4)$coup(10, 2, 1))
```

```
$scorder2 = PAR((1, 4, 1)$coup(2, 3, 2)/
(2, 4, 1)$coup(3, 3, 2)/(3, 4, 1)$coup(4, 3, 2)/
(4, 4, 1)$coup(5, 3, 2)/(5, 4, 1)$coup(6, 3, 2)/
(6, 4, 1)$coup(7, 3, 2)/(7, 4, 1)$coup(8, 3, 2)/
(8, 4, 1)$coup(9, 3, 2)/(9, 4, 1)$coup(10, 3, 2))
```

```
Row1=CNX('9.0', '5.0', '5.0', '8.0', '5.0', '5.0', '8.0', '5.0', '5.0', '11.0'/$scorder)
```

```
Row2=CNX('5.0', '1.0', '1.0', '1.0', '1.0', '1.0', '1.0', '1.0', '1.0', '10.0'/$scorder1)
```

```
Row3=CNX('5.0', '1.0', '1.0', '1.0', '1.0', '1.0', '1.0', '1.0', '1.0', '10.0'/$scorder1)
```

```
Row4=CNX('8.0', '1.0', '1.0', '4.0', '1.0', '1.0', '4.0', '1.0', '1.0', '10.0'/$scorder1)
```

```
Row5=CNX('5.0', '1.0', '1.0', '1.0', '1.0', '1.0', '1.0', '1.0', '1.0', '10.0'/$scorder1)
```

```

Row6=CNX('5.0','1.0','1.0','1.0','1.0','4.0','1.0','1.0','1.0','1.0','10.0'/$corder1)
Row7=CNX('8.0','1.0','1.0','4.0','1.0','1.0','1.0','1.0','1.0','1.0','10.0'/$corder1)
Row8=CNX('5.0','1.0','1.0','1.0','1.0','1.0','1.0','1.0','1.0','1.0','10.0'/$corder1)
Row9=CNX('5.0','1.0','1.0','1.0','1.0','1.0','1.0','1.0','1.0','1.0','10.0'/$corder1)
Row10=CNX('11.0','10.0','10.0','10.0','10.0','10.0','10.0','10.0','10.0','10.0',
'12.0'/$corder2)

System =CNX(Row1,Row2,Row3,Row4,Row5,Row6,Row7,Row8,Row9,Row10/
(1-2,4,1)$coup(2-2,3,2)/
(2-2,4,1)$coup(3-2,3,2)/(3-2,4,1)$coup(4-2,3,2)/
(4-2,4,1)$coup(5-2,3,2)/(5-2,4,1)$coup(6-2,3,2)/
(6-2,4,1)$coup(7-2,3,2)/(7-2,4,1)$coup(8-2,3,2)/
(8-2,4,1)$coup(9-2,3,2)/(9-2,4,1)$coup(10-2,4,3))

System = BDRY((1-10,3)(10-1,2)$coup(0)/(1-1,2)(1-10,3)$coup(1-1,2)(10-1,2))

! Overlays !
AllMat = OVLN('H2O-1' /*-*** / Zy2 / *--*-clad /
'air' /*-***-gap/

'PU-1' /*-'1.0'--fuel /
'PU-2' /*-'5.0'--fuel /
'boron1' /*-'4.0'--crod /
'boron2' /*-'8.0'--crod)

!Base case!
All1 = OVLD(1.0 /*-***)
AllT = OVLT(582.95 /*-*** / 963 /*-***-fuel / 628.556 /*-***-clad)
mos = OVSM(AllMat)
dos = OVSD(All1)
tos = OVST(AllT)
!removal of waba rods at 25 GWD/MT!
CROMat = OVLN('H2O-1' /*-'4.0'--crod, /*-'8.0'--crod)
moscr = OVSM(mos/CROMat)

!Low fuel temperature !
LofT = OVLT(582.95 /*-***-fuel / 582.95 /*-***-clad)
tLfos = OVST(tos/LofT)

!Low and High moderator temperature!
mLoMat = OVLN('H2O-4' /*-***-water, *--0-cool)
mHiMat = OVLN('H2O-5' /*-***-water, *--0-cool)
mLom = OVSM(mos/mLoMat)
mHim = OVSM(mos/mHiMat)

LomT = OVLT(564.59 /*-***-water, *--0-cool / 612.3992 /*-***-clad)
HimT = OVLT(599.32 /*-***-water, *--0-cool / 642.9616 /*-***-clad)
tLmT = OVST(tos/LomT)
tHmT = OVST(tos/HimT)

!Low and High boron concentration(0 and 2000ppm)!

```

```

LobMat  = OVLM('H2O-2' /*-**-water, *-**-0-cool)
HibMat  = OVLM('H2O-3' /*-**-water, *-**-0-cool)
mLob    = OVSM(mos/LobMat)
mHib    = OVSM(mos/HibMat)

!Burnup      !
st        = STAT(mos,dos,tos,38.02)
stccr     = STAT(moscr,dos,tos,38.02)
stloF     = STAT(mos,dos,tLfos,38.02)
sthiM     = STAT(mHim,dos,tHmT,38.02)
stloM     = STAT(mLom,dos,tLmT,38.02)
sthiB     = STAT(mHib,dos,tos,38.02)
stloB     = STAT(mLob,dos,tos,38.02)
Path      = PATH(/P,(st),1500/3,2500,25000/9,60000/7)
LFPath    = TREE(Path/P,(stloF)/0,500,1500,5000,20000/3,60000/4)
HMPPath   = TREE(Path/P,(sthiM)/0,500,1500,5000,20000/3,60000/4)
LMPPath   = TREE(Path/P,(stloM)/0,500,1500,5000,20000/3,60000/4)
HBPath    = TREE(Path/P,(sthiB)/0,500,1500,5000,20000/3,60000/4)
LBPath    = TREE(Path/P,(stloB)/0,500,1500,5000,20000/3,60000/4)
                                ! Output      !

G1        = GROUP(N/ 0)
G2        = GROUP(N/ 0.625, 0)
Apins     = AREA(*-**-<water>,*-**-<fuel>)
Aall      = AREA(<*-**-**>)

Asol      = AREA(<*-**-water,*-**-0-cool>)
Abur      = AREA(<*-**-crod>)

micAsol   = MICRO(G2,Asol//tr,ab,fi,nf,kf)
micAbur   = MICRO(G2,Abur//tr,ab,fi,nf,kf)
macAsol   = MACRO(G2,Asol/tr,ab,fi,nf,kf,p0)
macAbur   = MACRO(G2,Abur/tr,ab,fi,nf,kf,p0)

micAll    = MICRO(G1,Aall//tr,ab,fi)

micAll2   = MICRO(G2,Aall//tr,ab,fi,nf,kf)
macAll2   = MACRO(G2,Aall/tr,ab,fi,nf,kf,p0)

MacPin    = MACRO(G1, Apins/tr,fi,kf)

!corners !
NE        = FACE((1-10,2,4))
SE        = FACE((10-10,4,2))
SW        = FACE((10-1,1,3))
NW        = FACE((1-1,1,3))
!interior face!
Interior  = FACE((1-1,2,1) (2-1,3,2) (3-1,3,2) (4-1,3,2)
                (5-1,3,2) (6-1,3,2) (7-1,3,2) (8-1,3,2)
                (9-1,3,2) (10-1,3,2))

!exterior face!
Exterior  = FACE((10-1,2,1) (10-2,2,1) (10-3,2,1) (10-4,2,1)
                (10-5,2,1) (10-6,2,1) (10-7,2,1) (10-8,2,1))

```

(10-9,2,1) (10-10,2,1))

CurCorner = CUR(G2,NW,SW,SE,NE)

CurSide = CUR(G2,Interior,Exterior)

Vmhwh89h = RUN()

## Appendix C. HELIOS Input File: MOX, LEU, 3.5%, 104 IFBA (hu35104i.inp)

```

+THEL
Vu35104i  = CASE('library.bin'/ 'Vu35104i.hrf'/ 'UO-2 4.5 104 ifba')

! Data including thermal expansion      !
!fuel!
$rppe    = PAR("0.3950905")
$rair    = PAR("0.4010399")
$rcan    = PAR("0.4582740")
$rifba   = PAR("0.3990615")  !the radius was calculated preserving the ifba area!
$rcool   = PAR("0.5624394")
$rbox    = PAR("0.6032221")
$rcri    = PAR("0.4331")
$rcrg    = PAR("0.4369")
$rcro    = PAR("0.4839")
$rwco    = PAR("0.2858")
$rwbox   = PAR("0.3391")
$rwair   = PAR("0.3531")
$rwaba   = PAR("0.4039")
$rwabair = PAR("0.4179")
$rgap    = PAR("0.0435")

$pitch   = PAR("1.2598")
$sp2     = PAR("$pitch/2")
$psq2    = PAR("$rcan/2**0.5")
$psqh2   = PAR("$rbox/2**0.5")
$pgap2   = PAR("$rgap/2")
$coup    = PAR(3)

! Materials !
'H2O-1'  = MAT(NB/.705/1001,11.19;8016,88.81;5010,0.04;5011,0.06)
'H2O-2'  = MAT(NB/.705/1001,11.19;8016,88.81;5010,0.0;5011,0.0)
'H2O-3'  = MAT(NB/.705/1001,11.19;8016,88.81;5010,0.08;5011,0.12)
'H2O-4'  = MAT(NB/.743/1001,11.19;8016,88.81;5010,0.04;5011,0.06)
'H2O-5'  = MAT(NB/.662/1001,11.19;8016,88.81;5010,0.04;5011,0.06)

Zy2      = MAT(NB/6.504/40002,100.0)
'UO2-1'  = MAT(10.242/92234,2.4682E-2;92235,3.08525;92238,85.0401;
            8016,11.85)
'UO2-2'  = MAT(10.242/92234,2.4682E-2;92235,3.08525;92238,85.0401;
            8016,11.85)
'UO2-3'  = MAT(10.242/92234,2.4682E-2;92235,3.08525;92238,85.0401;
            8016,11.85)
'UO2-4'  = MAT(10.242/92234,2.4682E-2;92235,3.08525;92238,85.0401;
            8016,11.85)

'Uifba'  = MAT(1.69/40002,80.85;5010,3.53;5011,15.62)
'boron'  = MAT(0.71/5010,19.6;5011,80.4)

```

```

'air'    = MAT(NB/.001/8016,100)

! Geometry !
Pin      = CCS($rpe,$rair,$rcan//fuel,gap,clad)
Whole    = CCS($rcool,$rbox//water,clad)
Ipin     = CCS($rpe,$rifba,$rair,$rcan//fuel,ifba,gap,clad)
Bpin     = CCS($rcri,$rcrg,$rcro,$rcool,$rbox//water,gap,clad,water,clad)

$cell    = PAR(("-$p2", "-$p2") ("-$p2", $p2) ($p2, $p2) ($p2, "-$p2") !nodes 1-4!
(0, "-$p2") ("-$p2", 0) (0, $p2) ($p2, 0) ($rcan, 0) (0, "-$rcan") ("-$rcan", 0)
(0, $rcan) ($psq2, $psq2) ($psq2, "-$psq2") ("-$psq2", "-$psq2") ("-$psq2", $psq2)
/4, cool/Pin(0, 0)/ 1, 6, 11, 15, cool; 6, 2, 16, 11, cool; 2, 7, 12, 16, cool;
7, 3, 13, 12, cool; 3, 8, 9, 13, cool; 8, 4, 14, 9, cool; 4, 5, 10, 14, cool)

$icell   = PAR(("-$p2", "-$p2") ("-$p2", $p2) ($p2, $p2) ($p2, "-$p2") !nodes 1-4!
(0, "-$p2") ("-$p2", 0) (0, $p2) ($p2, 0) ($rcan, 0) (0, "-$rcan") ("-$rcan", 0)
(0, $rcan) ($psq2, $psq2) ($psq2, "-$psq2") ("-$psq2", "-$psq2") ("-$psq2", $psq2)
/4, cool/Ipin(0, 0)/ 1, 6, 11, 15, cool; 6, 2, 16, 11, cool;
2, 7, 12, 16, cool; 7, 3, 13, 12, cool; 3, 8, 9, 13, cool; 8, 4, 14, 9, cool; 4, 5, 10, 14, cool)

$wcell   = PAR(("-$p2", "-$p2") ("-$p2", $p2) ($p2, $p2) ($p2, "-$p2") !nodes 1-4!
(0, "-$p2") ("-$p2", 0) (0, $p2) ($p2, 0) ($rbox, 0) (0, "-$rbox") ("-$rbox", 0)
(0, $rbox) ($psqh2, $psqh2) ($psqh2, "-$psqh2") ("-$psqh2", "-$psqh2")
("-$psqh2", $psqh2)/4, cool/Whole(0, 0)/ 1, 6, 11, 15, cool; 6, 2, 16, 11, cool;
2, 7, 12, 16, cool; 7, 3, 13, 12, cool; 3, 8, 9, 13, cool; 8, 4, 14, 9, cool; 4, 5, 10, 14, cool)

$bcell   = PAR(("-$p2", "-$p2") ("-$p2", $p2) ($p2, $p2) ($p2, "-$p2") !nodes 1-4!
(0, "-$p2") ("-$p2", 0) (0, $p2) ($p2, 0) ($rbox, 0) (0, "-$rbox") ("-$rbox", 0)
(0, $rbox) ($psqh2, $psqh2) ($psqh2, "-$psqh2") ("-$psqh2", "-$psqh2")
("-$psqh2", $psqh2)/4, cool/Whole(0, 0)/ 1, 6, 11, 15, cool; 6, 2, 16, 11, cool;
2, 7, 12, 16, cool; 7, 3, 13, 12, cool; 3, 8, 9, 13, cool; 8, 4, 14, 9, cool; 4, 5, 10, 14, cool)

$wgap    = PAR(("-$pgap2", "-$p2") ("-$pgap2", $p2) ($pgap2, $p2) ($pgap2, "-$p2")
/4, cool//)

$cell2   = PAR(("-$p2", "-$p2") ("-$p2", 0) ($p2, 0) ($p2, "-$p2")
(0, "-$p2") ($rcan, 0) (0, "-$rcan") ("-$rcan", 0) ($psq2, "-$psq2")
("-$psq2", "-$psq2") /4, cool/Pin(0, 0) 3/ 3, 4, 9, 6, cool; 4, 5, 7, 9, cool;
5, 1, 10, 7, cool)

$icell2  = PAR(("-$p2", "-$p2") ("-$p2", 0) ($p2, 0) ($p2, "-$p2")
(0, "-$p2") ($rcan, 0) (0, "-$rcan") ("-$rcan", 0) ($psq2, "-$psq2")
("-$psq2", "-$psq2") /4, cool/Ipin(0, 0) 4/ 3, 4, 9, 6, cool; 4, 5, 7, 9, cool;
5, 1, 10, 7, cool)

$wcell2  = PAR(("-$p2", "-$p2") ("-$p2", 0) ($p2, 0) ($p2, "-$p2")
(0, "-$p2") ($rbox, 0) (0, "-$rbox") ("-$rbox", 0) ($psqh2, "-$psqh2")
("-$psqh2", "-$psqh2") /4, cool/Whole(0, 0) 2/3, 4, 9, 6, cool; 4, 5, 7, 9, cool;
5, 1, 10, 7, cool)

$bcell2  = PAR(("-$p2", "-$p2") ("-$p2", 0) ($p2, 0) ($p2, "-$p2")
(0, "-$p2") ($rbox, 0) (0, "-$rbox") ("-$rbox", 0) ($psqh2, "-$psqh2")
("-$psqh2", "-$psqh2") /4, cool/Whole(0, 0) 2/3, 4, 9, 6, cool; 4, 5, 7, 9, cool;
5, 1, 10, 7, cool)

$wgap2   = PAR(("-$pgap2", "-$p2") ("-$pgap2", 0) ($pgap2, 0) ($pgap2, "-$p2")
/4, cool//)

```

```

$wcell4 = PAR((0,"-$p2") (0,0) ($p2,0) ($p2,"-$p2")
($rbox,0) (0,"-$rbox") ($psqh2,"-$psqh2")
/4,cool/Whole(0,0) 2/4,1,6,7,cool)

$wgap4 = PAR(("-$pgap2","-$pgap2") ("-$pgap2",$pgap2) ($pgap2,$pgap2 )
($pgap2,"-$pgap2") /4,cool//)

'1.0' = STR($cell)
'2.0' = STR($icell)
'3.0' = STR($wcell)
'4.0' = STR($bcell)
'5.0' = STR($cell2)
'6.0' = STR($icell2)
'7.0' = STR($wcell2)
'8.0' = STR($bcell2)
'9.0' = STR($wcell4)
'10.0' = STR($wgap)
'11.0' = STR($wgap2)
'12.0' = STR($wgap4)

White = ALB(1/1/1)

$corder =PAR((1,3,4)$coup(2,2,1)/
(2,3,4)$coup(3,2,1)/(3,3,4)$coup(4,2,1)/
(4,3,4)$coup(5,2,1)/(5,3,4)$coup(6,2,1)/
(6,3,4)$coup(7,2,1)/(7,3,4)$coup(8,2,1)/
(8,3,4)$coup(9,2,1)/(9,3,4)$coup(10,2,1))

$corder1 =PAR((1,4,1)$coup(2,2,1)/
(2,3,4)$coup(3,2,1)/(3,3,4)$coup(4,2,1)/
(4,3,4)$coup(5,2,1)/(5,3,4)$coup(6,2,1)/
(6,3,4)$coup(7,2,1)/(7,3,4)$coup(8,2,1)/
(8,3,4)$coup(9,2,1)/(9,3,4)$coup(10,2,1))

$corder2 =PAR((1,4,1)$coup(2,3,2)/
(2,4,1)$coup(3,3,2)/(3,4,1)$coup(4,3,2)/
(4,4,1)$coup(5,3,2)/(5,4,1)$coup(6,3,2)/
(6,4,1)$coup(7,3,2)/(7,4,1)$coup(8,3,2)/
(8,4,1)$coup(9,3,2)/(9,4,1)$coup(10,3,2))

Row1=CNX('9.0','6.0','6.0','7.0','6.0','6.0','7.0','6.0','5.0','11.0'/$corder)
Row2=CNX('6.0','1.0','1.0','2.0','1.0','1.0','2.0','1.0','1.0','10.0'/$corder1)
Row3=CNX('6.0','1.0','1.0','2.0','1.0','1.0','2.0','1.0','1.0','10.0'/$corder1)
Row4=CNX('7.0','2.0','2.0','3.0','2.0','2.0','3.0','2.0','1.0','10.0'/$corder1)
Row5=CNX('6.0','1.0','1.0','2.0','1.0','2.0','2.0','1.0','1.0','10.0'/$corder1)
Row6=CNX('6.0','1.0','1.0','2.0','2.0','3.0','2.0','1.0','1.0','10.0'/$corder1)

```

```

Row7=CNX('7.0','2.0','2.0','3.0','2.0','2.0','1.0','1.0','1.0','10.0'/$corder1)
Row8=CNX('6.0','1.0','1.0','2.0','1.0','1.0','1.0','1.0','1.0','10.0'/$corder1)
Row9=CNX('5.0','1.0','1.0','1.0','1.0','1.0','1.0','1.0','2.0','10.0'/$corder1)
Row10=CNX('11.0','10.0','10.0','10.0','10.0','10.0','10.0','10.0','10.0','10.0',
'12.0'/$corder2)

System =CNX(Row1,Row2,Row3,Row4,Row5,Row6,Row7,Row8,Row9,Row10/
(1-2,4,1)$coup(2-2,3,2)/
(2-2,4,1)$coup(3-2,3,2)/(3-2,4,1)$coup(4-2,3,2)/
(4-2,4,1)$coup(5-2,3,2)/(5-2,4,1)$coup(6-2,3,2)/
(6-2,4,1)$coup(7-2,3,2)/(7-2,4,1)$coup(8-2,3,2)/
(8-2,4,1)$coup(9-2,3,2)/(9-2,4,1)$coup(10-2,4,3))

System = BDRY((1-10,3)(10-1,2)$coup(0)/(1-1,2)(1-10,3)$coup(1-1,2)(10-1,2))

! Overlays !
AllMat = OVLM('H2O-1' /*-*** / Zy2 / *-***-clad /
'air' /*-***-gap/
'Uifba' /*-***-ifba/
'UO2-1' /*-'1.0'-*-fuel /
'UO2-2' /*-'2.0'-*-fuel /
'UO2-3' /*-'5.0'-*-fuel /
'UO2-4' /*-'6.0'-*-fuel)

!Base case!
All1 = OVLD(1.0 /*-***)
AllT = OVLT(582.95 /*-*** / 963 /*-***-fuel / 628.556 /*-***-clad)
mos = OVSM(AllMat)
dos = OVSD(All1)
tos = OVST(AllT)

!Low fuel temperature !
Loft = OVLT(582.95 /*-***-fuel / 582.95 /*-***-clad)
tLfes = OVST(tos/Loft)

!Low and High moderator temperature!
mLoMat = OVLM('H2O-4' /*-***-water, *--0-cool)
mHiMat = OVLM('H2O-5' /*-***-water, *--0-cool)
mLom = OVSM(mos/mLoMat)
mHim = OVSM(mos/mHiMat)

LomT = OVLT(564.59 /*-***-water, *--0-cool / 612.3992 /*-***-clad)
HimT = OVLT(599.32 /*-***-water, *--0-cool / 642.9616 /*-***-clad)
tLmT = OVST(tos/LomT)
tHmT = OVST(tos/HimT)

!Low and High boron concentration(0 and 2000ppm)!
LobMat = OVLM('H2O-2' /*-***-water, *--0-cool)
HibMat = OVLM('H2O-3' /*-***-water, *--0-cool)
mLob = OVSM(mos/LobMat)
mHib = OVSM(mos/HibMat)

```

```

!Burnup      !
st           = STAT(mos,dos,tos,37.79)
stloF       = STAT(mos,dos,tLfos,37.79)
sthiM       = STAT(mHim,dos,tHmT,37.79)
stloM       = STAT(mLom,dos,tLmT,37.79)
sthiB       = STAT(mHib,dos,tos,37.79)
stloB       = STAT(mLob,dos,tos,37.79)
Path        = PATH(/ (st),1500/3,2500,30000/11, 60000/6)
LFPath      = TREE(Path/(stloF)/0,500,1500,5000,20000/3,60000/4)
HMPPath     = TREE(Path/(sthiM)/0,500,1500,5000,20000/3,60000/4)
LMPPath     = TREE(Path/(stloM)/0,500,1500,5000,20000/3,60000/4)
HBPath      = TREE(Path/(sthiB)/0,500,1500,5000,20000/3,60000/4)
LBPath      = TREE(Path/(stloB)/0,500,1500,5000,20000/3,60000/4)
              ! Output      !

G1          = GROUP(N/ 0)
G2          = GROUP(N/ 0.625, 0)
Apins       = AREA(*-***-<water>,*-***-<fuel>)
Aall        = AREA(<*-***>)
Asol        = AREA(<*-***-water,*-**-0-cool>)
Abur        = AREA(<*-***-ifba>)

micAsol     = MICRO(G2,Asol//tr,ab,fi,nf,kf)
micAbur     = MICRO(G2,Abur//tr,ab,fi,nf,kf)
macAsol     = MACRO(G2,Asol/tr,ab,fi,nf,kf,p0)
macAbur     = MACRO(G2,Abur/tr,ab,fi,nf,kf,p0)

micAll      = MICRO(G1,Aall//tr,ab,fi)

micAll2     = MICRO(G2,Aall//tr,ab,fi,nf,kf)
macAll2     = MACRO(G2,Aall/tr,ab,fi,nf,kf,p0)

MacPin      = MACRO(G1, Apins/tr,fi,kf)

!corners !
NE  = FACE((1-10,2,4))
SE  = FACE((10-10,4,2))
SW  = FACE((10-1,1,3))
NW  = FACE((1-1,1,3))

!interior face!
Interior = FACE((1-1,2,1) (2-1,3,2) (3-1,3,2) (4-1,3,2)
              (5-1,3,2) (6-1,3,2) (7-1,3,2) (8-1,3,2)
              (9-1,3,2) (10-1,3,2))

!exterior face!
Exterior = FACE((10-1,2,1) (10-2,2,1) (10-3,2,1) (10-4,2,1)
              (10-5,2,1) (10-6,2,1) (10-7,2,1) (10-8,2,1)
              (10-9,2,1) (10-10,2,1))

CurCorner  = CUR(G2,NW,SW,SE,NE)
CurSide    = CUR(G2,Interior,Exterior)

Vu35104i   = RUN()

```



## Appendix D. HELIOS Input File: Reflector (hnhwhref34.inp)

```

+THEL
Vmhwhref = CASE('library.bin'/ 'Vmhwhref.hrf'/ 'MOX 4.5 24 waba')

! Data including thermal expansion      !
!fuel!
$rppe = PAR("0.3950905")
$rrair = PAR("0.4010399")
$rrcan = PAR("0.4582740")
$rrifba = PROPIETARY 1
!the radius was calculated preserving the ifba area!
$rrcool = PAR("0.5624394")
$rrbox = PAR("0.6032221")
$rrcri = PAR("0.4331")
$rrcrg = PAR("0.4369")
$rrcro = PAR("0.4848823")
!thermal expansion included for waba rod!
$rrwco = PROPIETARY 2
$rrwbox = PROPIETARY 3
$rrwair = PROPIETARY 4
$rrwaba = PROPIETARY 5
$rrwabair = PROPIETARY 6
$rrgap = PROPIETARY 7

$rpitch = PAR("1.2598")
$rp2 = PAR("$rpitch/2")
$rp2sq = PAR("$rrcan/2**0.5")
$rp2sqh = PAR("$rrbox/2**0.5")
$rp2gap = PAR("$rrgap/2")
$rcoup = PAR(3)

! Materials !
'H2O-1' = MAT(NB/.705/1001,11.19;8016,88.81;5010,0.04;5011,0.06)
'H2O-2' = MAT(NB/.705/1001,11.19;8016,88.81;5010,0.0;5011,0.0)
'H2O-3' = MAT(NB/.705/1001,11.19;8016,88.81;5010,0.08;5011,0.12)
'H2O-4' = MAT(NB/.743/1001,11.19;8016,88.81;5010,0.04;5011,0.06)
'H2O-5' = MAT(NB/.662/1001,11.19;8016,88.81;5010,0.04;5011,0.06)

Zy2 = MAT(NB/6.504/40002,100.0)

'PU-1' = MAT(10.41/92235,1.763E-1;94238,0.0; 94239,3.94991;94240,
0.248936;94241,1.68366E-02;94242,4.23120E-03;
92234,1.41040E-03;92238,83.7524;8016,11.85)
'PU-2' = MAT(10.41/92235,1.763E-1;94238,0.0; 94239,3.94991;94240,
0.248936;94241,1.68366E-02;94242,4.23120E-03;
92234,1.41040E-03;92238,83.7524;8016,11.85)
'boron1' = MAT(3.563548/8016,42.5262;6000,2.1004;13027,47.8109;
5010,1.3938;5011,6.1687)
'boron2' = MAT(3.563548/8016,42.5262;6000,2.1004;13027,47.8109;
5010,1.3938;5011,6.1687)

```

```
'air' = MAT(NB/.001/8016,100)
'CRS' = MAT(NB/7.92/14000,0.51;24000,17.40;25055,1.99;26000,
           68.35;28000,11.70)
```

```
! Geometry !
```

```
Pin      = CCS($rpe,$rair,$rcan//fuel,gap,clad)
Whole    = CCS($rcool,$rbox//water,clad)
Ipin     = CCS($rpe,$rifba,$rair,$rcan//fuel,ifba,gap,clad)
Bpin     = CCS($rwo,$rwo,$rwair,$rwaba,$rwabair,$rcro,
              $rcool,$rbox//water,clad,gap,crod,gap,clad,water,clad)

$cell    = PAR(("-$p2","-$p2") ("-$p2",$p2) ($p2,$p2) ($p2,"-$p2") !nodes 1-4!
(0,"-$p2") ("-$p2",0) (0,$p2) ($p2,0) ($rcan,0) (0,"-$rcan") ("-$rcan",0)
(0,$rcan) ($psq2,$psq2) ($psq2,"-$psq2") ("-$psq2","-$psq2") ("-$psq2",$psq2)
/4,cool/Pin(0,0)/ 1,6,11,15,cool; 6,2,16,11,cool; 2,7,12,16,cool;
7,3,13,12,cool; 3,8,9,13,cool; 8,4,14,9,cool; 4,5,10,14,cool)

$icell   = PAR(("-$p2","-$p2") ("-$p2",$p2) ($p2,$p2) ($p2,"-$p2") !nodes 1-4!
(0,"-$p2") ("-$p2",0) (0,$p2) ($p2,0) ($rcan,0) (0,"-$rcan") ("-$rcan",0)
(0,$rcan) ($psq2,$psq2) ($psq2,"-$psq2") ("-$psq2","-$psq2") ("-$psq2",$psq2)
/4,cool/Ipin(0,0)/ 1,6,11,15,cool; 6,2,16,11,cool;
2,7,12,16,cool;7,3,13,12,cool; 3,8,9,13,cool; 8,4,14,9,cool; 4,5,10,14,cool)

$wcell   = PAR(("-$p2","-$p2") ("-$p2",$p2) ($p2,$p2) ($p2,"-$p2") !nodes 1-4!
(0,"-$p2") ("-$p2",0) (0,$p2) ($p2,0) ($rbox,0) (0,"-$rbox") ("-$rbox",0)
(0,$rbox) ($psqh2,$psqh2) ($psqh2,"-$psqh2") ("-$psqh2","-$psqh2")
("$psqh2",$psqh2)/4,cool/Whole(0,0)/ 1,6,11,15,cool; 6,2,16,11,cool;
2,7,12,16,cool;7,3,13,12,cool; 3,8,9,13,cool; 8,4,14,9,cool; 4,5,10,14,cool)

$bcell   = PAR(("-$p2","-$p2") ("-$p2",$p2) ($p2,$p2) ($p2,"-$p2") !nodes 1-4!
(0,"-$p2") ("-$p2",0) (0,$p2) ($p2,0) ($rbox,0) (0,"-$rbox") ("-$rbox",0)
(0,$rbox) ($psqh2,$psqh2) ($psqh2,"-$psqh2") ("-$psqh2","-$psqh2")
("$psqh2",$psqh2)/4,cool/Bpin(0,0)/ 1,6,11,15,cool; 6,2,16,11,cool;
2,7,12,16,cool;7,3,13,12,cool; 3,8,9,13,cool; 8,4,14,9,cool; 4,5,10,14,cool)

$wgap    = PAR(("-$pgap2","-$p2") ("-$pgap2",$p2) ($pgap2,$p2) ($pgap2,"-$p2")
/4,cool//)

$cell12  = PAR(("-$p2","-$p2") ("-$p2",0) ($p2,0) ($p2,"-$p2")
(0,"-$p2") ($rcan,0) (0,"-$rcan") ("-$rcan",0) ($psq2,"-$psq2")
("$psq2","-$psq2") /4,cool/Pin(0,0) 3/ 3,4,9,6,cool; 4,5,7,9,cool;
5,1,10,7,cool)

$icell12 = PAR(("-$p2","-$p2") ("-$p2",0) ($p2,0) ($p2,"-$p2")
(0,"-$p2") ($rcan,0) (0,"-$rcan") ("-$rcan",0) ($psq2,"-$psq2")
("$psq2","-$psq2")/4,cool/Ipin(0,0) 4/ 3,4,9,6,cool; 4,5,7,9,cool;
5,1,10,7,cool)

$wcell12 = PAR(("-$p2","-$p2") ("-$p2",0) ($p2,0) ($p2,"-$p2")
(0,"-$p2") ($rbox,0) (0,"-$rbox") ("-$rbox",0) ($psqh2,"-$psqh2")
("$psqh2","-$psqh2")/4,cool/Whole(0,0) 2/3,4,9,6,cool; 4,5,7,9,cool;
5,1,10,7,cool)

$bcell12 = PAR(("-$p2","-$p2") ("-$p2",0) ($p2,0) ($p2,"-$p2")
(0,"-$p2") ($rbox,0) (0,"-$rbox") ("-$rbox",0) ($psqh2,"-$psqh2")
("$psqh2","-$psqh2")/4,cool/Bpin(0,0) 8/3,4,9,6,cool; 4,5,7,9,cool;
```

```

5,1,10,7,cool)

$wgap2 = PAR(("-$pgap2", "-$p2") ("-$pgap2",0) ($pgap2,0) ($pgap2, "-$p2")
/4,cool//)

$wcell4 = PAR((0, "-$p2") (0,0) ($p2,0) ($p2, "-$p2")
($rbox,0) (0, "-$rbox") ($psqh2, "-$psqh2")
/4,cool/Whole(0,0) 2/4,1,6,7,cool)

$wgap4 = PAR(("-$pgap2", "-$pgap2") ("-$pgap2", $pgap2) ($pgap2, $pgap2 )
($pgap2, "-$pgap2") /4,cool//)

$baffle = PAR((0,0) (0, "$rgap+17*$p2") ("2.8575", "$rgap+17*$p2")
("2.8575",0) ("2.8575/2",0) ("2.8575/2", "$rgap+17*$p2") /4,baf//
1,2,6,5,baf)

$refl = PAR((0,0) (0, "$rgap+17*$p2") (18.64614, "$rgap+17*$p2")
(18.64614,0) (1.578864,0) (1.578864, "$rgap+17*$p2")
(3.157728,0) (3.157728, "$rgap+17*$p2") (4.739562,0)
(4.739562, "$rgap+17*$p2") (6.315456,0) (6.315456, "$rgap+17*$p2")
(7.89432,0) (7.89432, "$rgap+17*$p2") (10.044684,0)
(10.044684, "$rgap+17*$p2") (12.195048,0) (12.195048, "$rgap+17*$p2")
(14.345412,0) (14.345412, "$rgap+17*$p2") (16.495776,0)
(16.495776, "$rgap+17*$p2") / 4,ref//1,2,6,5,ref1; 5,6,8,7,ref1;
7,8,10,9,ref1; 9,10,12,11,ref1; 11,12,14,13,ref1; 13,14,16,15,ref;
15,16,18,17,ref; 17,18,20,19,ref; 19,20,22,21,ref)

'1.0' = STR($cell)
'2.0' = STR($icell)
'3.0' = STR($wcell)
'4.0' = STR($bcell)
'5.0' = STR($cell2)
'6.0' = STR($icell2)
'7.0' = STR($wcell2)
'8.0' = STR($bcell2)
'9.0' = STR($wcell4)
'10.0' = STR($wgap)
'11.0' = STR($wgap2)
'12.0' = STR($wgap4)
'13.0' = STR($baffle)
'14.0' = STR($refl)

White = ALB(1/1/1)
Black = ALB(1/0/0)

$scorder =PAR((1,3,4)$coup(2,2,1)/
(2,3,4)$coup(3,2,1)/(3,3,4)$coup(4,2,1)/
(4,3,4)$coup(5,2,1)/(5,3,4)$coup(6,2,1)/
(6,3,4)$coup(7,2,1)/(7,3,4)$coup(8,2,1)/
(8,3,4)$coup(9,2,1)/(9,3,4)$coup(10,2,1))

$scorder1 =PAR((1,4,1)$coup(2,2,1)/
(2,3,4)$coup(3,2,1)/(3,3,4)$coup(4,2,1)/
(4,3,4)$coup(5,2,1)/(5,3,4)$coup(6,2,1)/
(6,3,4)$coup(7,2,1)/(7,3,4)$coup(8,2,1)/
(8,3,4)$coup(9,2,1)/(9,3,4)$coup(10,2,1))

```

```

$corder2 =PAR((1,4,1)$coup(2,3,2)/
(2,4,1)$coup(3,3,2)/(3,4,1)$coup(4,3,2)/
(4,4,1)$coup(5,3,2)/(5,4,1)$coup(6,3,2)/
(6,4,1)$coup(7,3,2)/(7,4,1)$coup(8,3,2)/
(8,4,1)$coup(9,3,2)/(9,4,1)$coup(10,3,2))

Row1=CNX('9.0','5.0','5.0','8.0','5.0','5.0','8.0','5.0','5.0','11.0'/$corder)
Row2=CNX('5.0','1.0','1.0','1.0','1.0','1.0','1.0','1.0','1.0','10.0'/$corder1)
Row3=CNX('5.0','1.0','1.0','1.0','1.0','1.0','1.0','1.0','1.0','10.0'/$corder1)
Row4=CNX('8.0','1.0','1.0','4.0','1.0','1.0','4.0','1.0','1.0','10.0'/$corder1)
Row5=CNX('5.0','1.0','1.0','1.0','1.0','1.0','1.0','1.0','1.0','10.0'/$corder1)
Row6=CNX('5.0','1.0','1.0','1.0','1.0','4.0','1.0','1.0','1.0','10.0'/$corder1)
Row7=CNX('8.0','1.0','1.0','4.0','1.0','1.0','1.0','1.0','1.0','10.0'/$corder1)
Row8=CNX('5.0','1.0','1.0','1.0','1.0','1.0','1.0','1.0','1.0','10.0'/$corder1)
Row9=CNX('5.0','1.0','1.0','1.0','1.0','1.0','1.0','1.0','1.0','10.0'/$corder1)
Row10=CNX('11.0','10.0','10.0','10.0','10.0','10.0','10.0','10.0','10.0','10.0',
'12.0'/$corder2)

System =CNX(Row1,Row2,Row3,Row4,Row5,Row6,Row7,Row8,Row9,Row10,'13.0','14.0'/
(1-2,4,1)$coup(2-2,3,2)/
(2-2,4,1)$coup(3-2,3,2)/(3-2,4,1)$coup(4-2,3,2)/
(4-2,4,1)$coup(5-2,3,2)/(5-2,4,1)$coup(6-2,3,2)/
(6-2,4,1)$coup(7-2,3,2)/(7-2,4,1)$coup(8-2,3,2)/
(8-2,4,1)$coup(9-2,3,2)/(9-2,4,1)$coup(10-2,4,3)/
(10-10,1,4)$coup(11,1,(0,"$rgap"))/(11,3,4)$coup(12,2,1))

System = BDRY((12,4,3)$coup(White)/(12,3,4)$coup(Black))

! Overlays !
AllMat = OVLM('H2O-1' /*-*** / Zy2 / *-**-clad /
'air' /*-**-gap/

'PU-1' /*-'1.0'-*-fuel /
'PU-2' /*-'5.0'-*-fuel /
'boron1' /*-'4.0'-*-crod /
'boron2' /*-'8.0'-*-crod /
'CRS' /*-**-0-baf)

!Base case!
All1 = OVLD(1.0 /*-***)
AllT = OVLT(582.95 /*-*** / 963 /*-**-fuel / 628.556 /*-**-clad)
mos = OVSM(AllMat)

```

```

dos      = OVSD(All1)
tos      = OVST(AllT)
!removal of waba rods at 25 GWD/MT!
CROMat  = OVLM('H2O-1' /*--4.0'--crod,*--8.0'--crod)
moscr   = OVSM(mos/CROMat)

!Low fuel temperature !
LofT    = OVLT(582.95 /*--582.95 / 582.95 /*--clad)
tLfoss  = OVST(tos/LofT)

!Low and High moderator temperature!
mLoMat  = OVLM('H2O-4' /*--water, *--0-cool)
mHiMat  = OVLM('H2O-5' /*--water, *--0-cool)
mLom    = OVSM(mos/mLoMat)
mHim    = OVSM(mos/mHiMat)

LomT    = OVLT(564.59 /*--water, *--0-cool / 612.3992 /*--clad)
HimT    = OVLT(599.32 /*--water, *--0-cool / 642.9616 /*--clad)
tLmT    = OVST(tos/LomT)
tHmT    = OVST(tos/HimT)

!Low and High boron concentration(0 and 2000ppm)!
LobMat  = OVLM('H2O-2' /*--water, *--0-cool)
HibMat  = OVLM('H2O-3' /*--water, *--0-cool)
mLob    = OVSM(mos/LobMat)
mHib    = OVSM(mos/HibMat)

!Burnup      !
st       = STAT(mos,dos,tos,38.02)
stccr   = STAT(moscr,dos,tos,38.02)
!stloF    = STAT(mos,dos,tLfoss,38.02)!
sthiM   = STAT(mHim,dos,tHmT,38.02)
stloM   = STAT(mLom,dos,tLmT,38.02)
sthiB   = STAT(mHib,dos,tos,38.02)
stloB   = STAT(mLob,dos,tos,38.02)
Path     = PATH(/P,(st),1500/3,2500,25000/9,60000/7)
!LFPath  = TREE(Path/P,(stloF)/0,500,1500,5000,20000/3,60000/4)!
HMPath  = TREE(Path/P,(sthiM)/0,500,1500,5000,20000/3,60000/4)
LMPath  = TREE(Path/P,(stloM)/0,500,1500,5000,20000/3,60000/4)
HBPath  = TREE(Path/P,(sthiB)/0,500,1500,5000,20000/3,60000/4)
LBPath  = TREE(Path/P,(stloB)/0,500,1500,5000,20000/3,60000/4)
! Output      !

G1      = GROUP(N/ 0)
G2      = GROUP(N/ 0.625, 0)
ABARE   = AREA(<11-**,12-0-ref1>)
ABA     = AREA(<11-**>)
ARE     = AREA(<12-0-ref>)

micBARE = MICRO(G2,ABARE//ab)
macBARE = MACRO(G2,ABARE/tr,ab,fi,nf,kf,p0)

MicBA   = MICRO(G2, ABA//ab)
MacBA   = MACRO(G2, ABA/tr,ab,fi,nf,kf,p0)

MicRE   = MICRO(G2, ARE//ab)
MacRE   = MACRO(G2, ARE/tr,ab,fi,nf,kf,p0)

```

```
fuba      = FACE((1-10,3,4) (2-10,3,4) (3-10,3,4) (4-10,3,4)
                (5-10,3,4) (6-10,3,4) (7-10,3,4) (8-10,3,4)
                (9-10,3,4)(10-10,4,1))
Cufuba    = CUR(G2,fuba)

bare      = FACE((12,13,14))
Cubare    = CUR(G2,bare)

refle     = FACE((12,4,3))
Curef     = CUR(G2,refle)

Vmhwref   = RUN()
```



```

PARAM          = LIS(;;6/'Parameters'/E/f5: gapt; ppitch; apitch; isym; num)

!----- Enter your case and file selection ----!
%'Case#'       = 1
%'file'        = 7

!--- For each file, enter the following info.: -----!
!----- Burnup steps for base case                : $BurnupBase   ---!
!----- Burnup steps for branch case              : $BurnupBranch ---!
!----- Helios case                               : $Hcase         ---!
!----- Helios output file                        : $Hfile         ---!
!----- U-235 enrichment                          : $enr          ---!
!----- Weight percent of BA                      : $wba          ---!
!----- Present number of BA rods                 : $ibap         ---!

%if ('file'.eq.1) then
$BurnupBase    = PAR(0, 500, 1000, 1500, 2500, 5000, 7500,10000,12500,
                    15000,17500,20000,22500,25000,27500,30000,35000,
                    40000,45000,50000,55000,60000)
$BurnupBranch  = PAR(0, 500, 1500, 5000,10000,15000,20000,
                    30000,40000,50000,60000)
$Hcase         = PAR('Vuhhi')
$Hfile         = PAR('Vuhhi.hrf')
C1             = IMP(HELIOS; $Hcase/$Hfile)

enr            = FOR(4.5)
wba            = FOR(0.000)
ibap           = FOR(128)          ! present number of BA rods !
FILEPARAM     = LIS('/Parameters'/E/f5: enr; f0: wba; ibap)
%endif

%if ('file'.eq.2) then
$BurnupBase    = PAR(0, 500, 1000, 1500, 2500, 5000, 7500,10000,12500,
                    15000,17500,20000,22500,25000,27500,30000,35000,
                    40000,45000,50000,55000,60000)
$BurnupBranch  = PAR(0, 500, 1500, 5000,10000,15000,20000,
                    30000,40000,50000,60000)
$Hcase         = PAR('Vuhmi')
$Hfile         = PAR('Vuhmi.hrf')
C1             = IMP(HELIOS; $Hcase/$Hfile)

enr            = FOR(4.5)
wba            = FOR(0.000)
ibap           = FOR(64)
FILEPARAM     = LIS('/Parameters'/E/f5: enr; f0: wba; ibap)
%endif

%if ('file'.eq.3) then
$BurnupBase    = PAR(0, 500, 1000, 1500, 2500, 5000, 7500,10000,12500,
                    15000,17500,20000,22500,25000,27500,30000,35000,
                    40000,45000,50000,55000,60000)
$BurnupBranch  = PAR(0, 500, 1500, 5000,10000,15000,20000,
                    30000,40000,50000,60000)
$Hcase         = PAR('Vuhni')

```

```

$Hfile      = PAR('Vuhni.hrf')
C1          = IMP(HELIOS; $Hcase/$Hfile)

enr         = FOR(4.5)
wba         = FOR(0.000)
ibap        = FOR(0)
FILEPARAM   = LIS('/Parameters'/E/f5: enr; f0: wba; ibap)
%endif

%if ('file'.eq.4) then
$BurnupBase = PAR(0, 500, 1000, 1500, 2500, 5000, 7500,10000,12500,
                  15000,17500,20000,22500,25000,27500,30000,35000,
                  40000,45000,50000,55000,60000)
$BurnupBranch = PAR(0, 500, 1500, 5000,10000,15000,20000,
                    30000,40000,50000,60000)
$Hcase       = PAR('Vulhi')
$Hfile       = PAR('Vulhi.hrf')
C1           = IMP(HELIOS; $Hcase/$Hfile)

enr         = FOR(4.0)
wba         = FOR(0.000)
ibap        = FOR(128)
FILEPARAM   = LIS('/Parameters'/E/f5: enr; f0: wba; ibap)
%endif

%if ('file'.eq.5) then
$BurnupBase = PAR(0, 500, 1000, 1500, 2500, 5000, 7500,10000,12500,
                  15000,17500,20000,22500,25000,27500,30000,35000,
                  40000,45000,50000,55000,60000)
$BurnupBranch = PAR(0, 500, 1500, 5000,10000,15000,20000,
                    30000,40000,50000,60000)
$Hcase       = PAR('Vulmi')
$Hfile       = PAR('Vulmi.hrf')
C1           = IMP(HELIOS; $Hcase/$Hfile)

enr         = FOR(4.0)
wba         = FOR(0.000)
ibap        = FOR(64)
FILEPARAM   = LIS('/Parameters'/E/f5: enr; f0: wba; ibap)
%endif

%if ('file'.eq.6) then
$BurnupBase = PAR(0, 500, 1000, 1500, 2500, 5000, 7500,10000,12500,
                  15000,17500,20000,22500,25000,27500,30000,35000,
                  40000,45000,50000,55000,60000)
$BurnupBranch = PAR(0, 500, 1500, 5000,10000,15000,20000,
                    30000,40000,50000,60000)
$Hcase       = PAR('Vulni')
$Hfile       = PAR('Vulni.hrf')
C1           = IMP(HELIOS; $Hcase/$Hfile)

enr         = FOR(4.0)
wba         = FOR(0.000)
ibap        = FOR(0)

```

```

FILEPARAM      = LIS('/Parameters'/E/f5: enr; f0: wba; ibap)
%endif

%if ('file'.eq.7) then
$BurnupBase    = PAR(0, 500, 1000, 1500, 2500, 5000, 7500,10000,12500,
                    15000,17500,20000,22500,25000,27500,30000,35000,
                    40000,45000,50000,55000,60000)
$BurnupBranch  = PAR(0, 500, 1500, 5000,10000,15000,20000,
                    30000,40000,50000,60000)
$Hcase         = PAR('Vmhwh89h')
$Hfile         = PAR('Vmhwh89h.hrf')
C1             = IMP(HELIOS; $Hcase/$Hfile)

enr            = FOR(4.5)
wba            = FOR(0.000)
ibap           = FOR(24)
FILEPARAM      = LIS('/Parameters'/E/f5: enr; f0: wba; ibap)
%endif

%if ('file'.eq.8) then
$BurnupBase    = PAR(0, 500, 1000, 1500, 2500, 5000, 7500,10000,12500,
                    15000,17500,20000,22500,25000,27500,30000,35000,
                    40000,45000,50000,55000,60000)
$BurnupBranch  = PAR(0, 500, 1500, 5000,10000,15000,20000,
                    30000,40000,50000,60000)
$Hcase         = PAR('Vmhw189g')
$Hfile         = PAR('Vmhw189g.hrf')
C1             = IMP(HELIOS; $Hcase/$Hfile)

enr            = FOR(4.0)
wba            = FOR(0.000)
ibap           = FOR(12)
FILEPARAM      = LIS('/Parameters'/E/f5: enr; f0: wba; ibap)
%endif

%if ('file'.eq.9) then
$BurnupBase    = PAR(0, 500, 1000, 1500, 2500, 5000, 7500,10000,12500,
                    15000,17500,20000,22500,25000,27500,30000,35000,
                    40000,45000,50000,55000,60000)
$BurnupBranch  = PAR(0, 500, 1500, 5000,10000,15000,20000,
                    30000,40000,50000,60000)
$Hcase         = PAR('Vmlwh89g')
$Hfile         = PAR('Vmlwh89g.hrf')
C1             = IMP(HELIOS; $Hcase/$Hfile)

enr            = FOR(4.0)
wba            = FOR(0.000)
ibap           = FOR(24)
FILEPARAM      = LIS('/Parameters'/E/f5: enr; f0: wba; ibap)
%endif

%if ('file'.eq.10) then
$BurnupBase    = PAR(0, 500, 1000, 1500, 2500, 5000, 7500,10000,12500,

```

```

                15000,17500,20000,22500,25000,27500,30000,35000,
                40000,45000,50000,55000,60000)
$BurnupBranch = PAR(0, 500, 1500, 5000,10000,15000,20000,
                30000,40000,50000,60000)
$Hcase       = PAR('Vmlwl89g')
$Hfile      = PAR('Vmlwl89g.hrf')
C1          = IMP(HELIOS; $Hcase/$Hfile)

enr         = FOR(4.0)
wba         = FOR(0.000)
ibap        = FOR(12)
FILEPARAM   = LIS('/Parameters'/E/f5: enr; f0: wba; ibap)
%endif

%if ('file'.eq.11) then
$BurnupBase = PAR(0, 500, 1000, 1500, 2500, 5000, 7500,10000,12500,
                15000,17500,20000,22500,25000,27500,30000,35000,
                40000,45000,50000,55000,60000)
$BurnupBranch = PAR(0, 500, 1500, 5000,10000,15000,20000,
                30000,40000,50000,60000)
$Hcase       = PAR('Vmhwn89g')
$Hfile      = PAR('Vmhwn89g.hrf')
C1          = IMP(HELIOS; $Hcase/$Hfile)

enr         = FOR(4.5)
wba         = FOR(0.000)
ibap        = FOR(0)
FILEPARAM   = LIS('/Parameters'/E/f5: enr; f0: wba; ibap)
%endif

%if ('file'.eq.12) then
$BurnupBase = PAR(0, 500, 1000, 1500, 2500, 5000, 7500,10000,12500,
                15000,17500,20000,22500,25000,27500,30000,35000,
                40000,45000,50000,55000,60000)
$BurnupBranch = PAR(0, 500, 1500, 5000,10000,15000,20000,
                30000,40000,50000,60000)
$Hcase       = PAR('Vmlwn89g')
$Hfile      = PAR('Vmlwn89g.hrf')
C1          = IMP(HELIOS; $Hcase/$Hfile)

enr         = FOR(4.0)
wba         = FOR(0.000)
ibap        = FOR(0)
FILEPARAM   = LIS('/Parameters'/E/f5: enr; f0: wba; ibap)
%endif

```

```

!--- For each case, enter the following info.: -----!
!----- Path : $Path ---!
!----- State : $State ---!
!----- Instantaneous fuel temperature : tfuchg ---!
!----- Reference fuel temperature : tfuref ---!
!----- Instantaneous moderator temperature : tmochg ---!
!----- Reference moderator temperature : tmoref ---!
!----- Instantaneous boron concentration : ppmchg ---!
!----- Reference boron concentration : ppmref ---!

```

```

%if ('Case#'.eq.1) then
  $Path      = PAR('Path')
  $State     = PAR('st')
  $Burnup   = PAR($BurnupBase)
  $scalp    = PAR($Path:($State)$Burnup)

  tfuchg    = FOR(963)
  tfuref    = FOR(963)
  tmochg    = FOR(582.95)
  tmoref    = FOR(582.95)
  ppmchg    = FOR(1000)
  ppmref    = FOR(1000)
  CASEPARAM = LIS(;;6/'Parameters'
                  /E/f2:tfuchg; tfuref; tmochg; tmoref;ppmchg; ppmref)
%endif

%if ('Case#'.eq.2) then
  $Path      = PAR('LFPath')
  $State     = PAR(stloF)
  $Burnup   = PAR($BurnupBranch)
  $scalp    = PAR($Path:($State)$Burnup)

  tfuchg    = FOR(582.95)
  tfuref    = FOR(963)
  tmochg    = FOR(582.95)
  tmoref    = FOR(582.95)
  ppmchg    = FOR(1000)
  ppmref    = FOR(1000)
  CASEPARAM = LIS(;;6/'Parameters'
                  /E/f2:tfuchg; tfuref; tmochg; tmoref;ppmchg; ppmref)
%endif

%if ('Case#'.eq.3) then
  $Path      = PAR('HMPath')
  $State     = PAR(sthiM)
  $Burnup   = PAR($BurnupBranch)
  $scalp    = PAR($Path:($State)$Burnup)

  tfuchg    = FOR(963)
  tfuref    = FOR(963)
  tmochg    = FOR(599.32)
  tmoref    = FOR(582.95)
  ppmchg    = FOR(1000)
  ppmref    = FOR(1000)
  CASEPARAM = LIS(;;6/'Parameters'
                  /E/f2:tfuchg; tfuref; tmochg; tmoref;ppmchg; ppmref)
%endif

%if ('Case#'.eq.4) then
  $Path      = PAR('LMPath')
  $State     = PAR(stloM)
  $Burnup   = PAR($BurnupBranch)
  $scalp    = PAR($Path:($State)$Burnup)

  tfuchg    = FOR(963)

```





```

'Yield(I)'      = FOR(1E24*'N(I)')*$lambdaI/avfxfi)
'Yield(Xe)'    = FOR(1E24*'N(Xe)')*(avfxaXe*1E-24+$lambdaXe)/avfxfi-'Yield(I)')
'Yield(Pm)'    = FOR(1E24*'N(Pm)')*$lambdaPm/avfxfi)

FPYIELDS      = LIS(/'Fission Product Yields for I-135, Xe-135, Pm-149'
                  /E/f0:bu; e5:'Yield(I)';'Yield(Xe)';'Yield(Pm)')

! (4) XESM !
micxe         = SEL(ab/MIC/C1;$micALL2/$scalp/54635)
ndxe          = SEL(nd/MIC/C1;$micALL2/$scalp/54635)
macxe         = FOR(micxe*ndxe)
micism        = SEL(ab/MIC/C1;$micALL2/$scalp/62649)
ndism         = SEL(nd/MIC/C1;$micALL2/$scalp/62649)
macism        = FOR(micism*ndism)

XESM          = LIS(/'Micro- & Macro- Absorption XS for Xe and Sm'
                  /E/f0:bu; e5: micxe;macxe;micism;macism)

! (5) SOLBO !
micabbo10     = SEL(ab/MIC/C1;$micAsol/$scalp/5010)
ndbo10x       = SEL(nd/MIC/C1;$micAsol/$scalp/5010)
vo            = SEL(vo/MIC/C1;$micALL/$scalp)
vox           = SEL(vo/MIC/C1;$micAsol/$scalp)
macabbo10     = FOR(micabbo10*ndbo10x*vox/vo)

micabbo11     = SEL(ab/MIC/C1;$micAsol/$scalp/5011)
ndbo11x       = SEL(nd/MIC/C1;$micAsol/$scalp/5011)
macabbo11     = FOR(micabbo11*ndbo11x*vox/vo)

micbo         = FOR((macabbo10+macabbo11)/((ndbo10x+ndbo11x)*vox/vo))
macbo         = FOR(macabbo10+macabbo11)

SOLBO         = LIS(/'Micro- & Macro- Absorption XS for Soluble Boron'
                  /E/f0:bu; e5: micbo; macbo)

! (6) KINF !
kinf          = SEL(kinf/MAC/C1;$macALL2/$scalp)

! (7) POWERRAT !
fx            = SEL(fx/MAC/C1;$macPin/$scalp)
kf            = SEL(kf/MAC/C1;$macPin/$scalp)
vo            = SEL(vo/MAC/C1;$macPin/$scalp)
vx            = SEL(vx/MAC/C1;$macPin/$scalp)
vy            = SEL(vy/MAC/C1;$macPin/$scalp)

Power         = FOR(fx*kf*vo)

!<<< might need to change def'n of average power. Avg. over FUEL pin only >>>!
%if ('file'.eq.1) then
AvePower      = FOR((@smR(Power)))/66)
%endif

%if ('file'.eq.2) then
AvePower      = FOR((@smR(Power)))/66)
%endif

%if ('file'.eq.3) then
AvePower      = FOR((@smR(Power)))/66)

```

```

%endif

%if ('file'.eq.4) then
AvePower      = FOR((@smR(Power))/66)
%endif

%if ('file'.eq.5) then
AvePower      = FOR((@smR(Power))/66)
%endif

%if ('file'.eq.6) then
AvePower      = FOR((@smR(Power))/66)
%endif

%if ('file'.eq.7) then
AvePower      = FOR((@smR(Power))/66)
%endif

%if ('file'.eq.8) then
AvePower      = FOR((@smR(Power))/66)
%endif

%if ('file'.eq.9) then
AvePower      = FOR((@smR(Power))/66)
%endif

%if ('file'.eq.10) then
AvePower      = FOR((@smR(Power))/66)
%endif

%if ('file'.eq.11) then
AvePower      = FOR((@smR(Power))/66)
%endif

%if ('file'.eq.12) then
AvePower      = FOR((@smR(Power))/66)
%endif

PP            = FOR(Power/AvePower)
PP2          = FOR(2*PP)
PPmax        = FOR(@mxR(PP))

! (8) FLUXRAT !
fx           = SEL(fx/MAC/C1;$macALL2/$scalp)
i1           = IND(fx/RAN:;;1;)
i2           = IND(fx/RAN:;;2;)
flxfast      = FOR(fx^i1)
flxthm      = FOR(fx^i2)
fxrat        = FOR((flxfast)/(flxthm))

POWER        = LIS(/'k_inf, Max Rel. Pin Power, flux ratio'
                  /E/f0:bu; f5:kinf; PPmax; fxrat)

! (9) FORMFACTOR !
irow1        = IND(PP2/RAN:1-9;;;)
irow2a       = IND(PP2/RAN:10 ;;;)
irow2b       = IND(PP/RAN:11-18;;;)

```

```

irow3a      = IND(PP/RAN:19  ;;;; )
irow3b      = IND(PP/RAN:20-27;;;;)
irow4a      = IND(PP/RAN:28  ;;;; )
irow4b      = IND(PP/RAN:29-36;;;;)
irow5a      = IND(PP/RAN:37  ;;;; )
irow5b      = IND(PP/RAN:38-45;;;;)
irow6a      = IND(PP/RAN:46  ;;;; )
irow6b      = IND(PP/RAN:47-54;;;;)
irow7a      = IND(PP/RAN:55  ;;;; )
irow7b      = IND(PP/RAN:56-63;;;;)
irow8a      = IND(PP/RAN:64  ;;;; )
irow8b      = IND(PP/RAN:65-72;;;;)
irow9a      = IND(PP/RAN:73  ;;;; )
irow9b      = IND(PP/RAN:74-81;;;;)

PProw1      = FOR(PP2^irow1  )
PProw2a     = FOR(PP2^irow2a )
PProw2b     = FOR(PP^irow2b  )
PProw3a     = FOR(PP2^irow3a )
PProw3b     = FOR(PP^irow3b  )
PProw4a     = FOR(PP2^irow4a )
PProw4b     = FOR(PP^irow4b  )
PProw5a     = FOR(PP2^irow5a )
PProw5b     = FOR(PP^irow5b  )
PProw6a     = FOR(PP2^irow6a )
PProw6b     = FOR(PP^irow6b  )
PProw7a     = FOR(PP2^irow7a )
PProw7b     = FOR(PP^irow7b  )
PProw8a     = FOR(PP2^irow8a )
PProw8b     = FOR(PP^irow8b  )
PProw9a     = FOR(PP2^irow9a )
PProw9b     = FOR(PP^irow9b  )

FORMFACTOR  = LIS(;;9/'Pin Power'/E/f5:PProw1;PProw2a;PProw2b;PProw3a;
                PProw3b;PProw4a;PProw4b;;PProw5a;PProw5b;
                PProw6a;PProw6b;PProw7a;PProw7b;PProw8a;PProw8b;
                PProw9a;PProw9b)

```

!<<< End of form factor user input

>>>!

```

! (10) MACRO !
dr          = SEL(dr/MAC/C1;$macALL2/$scalp)
tr          = SEL(tr/MAC/C1;$macALL2/$scalp)
ab          = SEL(ab/MAC/C1;$macALL2/$scalp)
nf          = SEL(nf/MAC/C1;$macALL2/$scalp)
kf          = SEL(kf/MAC/C1;$macALL2/$scalp)
p0          = SEL(p0/MAC/C1;$macALL2/$scalp)
fi          = SEL(fi/MAC/C1;$macALL2/$scalp)

i12         = IND(p0/RAN:;;;2;1)
i21         = IND(p0/RAN:;;;1;2)
i1          = IND(nf/RAN:;;;1;)
i2          = IND(nf/RAN:;;;2;)
diff        = FOR(dr/(3.0*tr))
d1          = FOR(diff^i1)
d2          = FOR(diff^i2)
s12         = FOR(p0^i12)

```

```

s21          = FOR(p0^i21)
ab1          = FOR(ab^i1)
ab2          = FOR(ab^i2)
nsigf1      = FOR(nf^i1)
nsigf2      = FOR(nf^i2)
ksigf1      = FOR(kf^i1)
ksigf2      = FOR(kf^i2)
fi1         = FOR(fi^i1)
fi2         = FOR(fi^i2)
nu2         = FOR(nsigf2/fi2)
kappa2      = FOR(ksigf2/fi2)
srem        = FOR(s12-s21/fxrat)

'MACRO-FAST' = LIS(/'Macroscopic XS Fast'
                  /E/f0:bu; e5: d1;srem;ab1;nsigf1;ksigf1)

'MACRO-TERMAL' = LIS(/'Macroscopic XS Thermal'
                    /E/f0:bu; e5: d2;ab2;nsigf2;ksigf2;nu2;kappa2)

! (11) ISOTOPES !
id          = SEL(id/MIC/C1;$micALL)
ISOTOPES    = LIST(/'Isotopes present in the system'/I/f0: id)

!<<< Make list for isotopes present in the assembly >>>!

!idformsa=1!
nd          = SEL(nd/MIC/C1;$micALL2/$scalp/92234)
micab       = SEL(ab/MIC/C1;$micALL2/$scalp/92234)
micfi       = SEL(fi/MIC/C1;$micALL2/$scalp/92234)
mickf       = SEL(kf/MIC/C1;$micALL2/$scalp/92234)
micnf       = SEL(nf/MIC/C1;$micALL2/$scalp/92234)
i1          = IND(mickf/RAN:;;1;)
mickf1      = FOR(mickf^i1)
'92234'     = LIS(/'Micro. vs. burnup for Isotope U-234'
                  /E/f0:bu; e5: nd; micab; micfi; micnf; mickf1)

!idformsa=2!
nd          = SEL(nd/MIC/C1;$micALL2/$scalp/92235)
micab       = SEL(ab/MIC/C1;$micALL2/$scalp/92235)
micfi       = SEL(fi/MIC/C1;$micALL2/$scalp/92235)
mickf       = SEL(kf/MIC/C1;$micALL2/$scalp/92235)
micnf       = SEL(nf/MIC/C1;$micALL2/$scalp/92235)
i1          = IND(mickf/RAN:;;1;)
mickf1      = FOR(mickf^i1)
'92235'     = LIS(/'Micro. vs. burnup for Isotope U-235'
                  /E/f0:bu; e5: nd; micab; micfi; micnf; mickf1)

!idformsa=3!
nd          = SEL(nd/MIC/C1;$micALL2/$scalp/92236)
micab       = SEL(ab/MIC/C1;$micALL2/$scalp/92236)
micfi       = SEL(fi/MIC/C1;$micALL2/$scalp/92236)
mickf       = SEL(kf/MIC/C1;$micALL2/$scalp/92236)
micnf       = SEL(nf/MIC/C1;$micALL2/$scalp/92236)
i1          = IND(mickf/RAN:;;1;)
mickf1      = FOR(mickf^i1)
'92236'     = LIS(/'Micro. vs. burnup for Isotope U-236'
                  /E/f0:bu; e5: nd; micab; micfi; micnf; mickf1)

```

```

!idformsa=4!
nd          = SEL(nd/MIC/C1;$micALL2/$scalp/92238)
micab       = SEL(ab/MIC/C1;$micALL2/$scalp/92238)
micfi       = SEL(fi/MIC/C1;$micALL2/$scalp/92238)
mickf       = SEL(kf/MIC/C1;$micALL2/$scalp/92238)
micnf       = SEL(nf/MIC/C1;$micALL2/$scalp/92238)
il          = IND(mickf/RAN:;;1;)
mickf1      = FOR(mickf^il)
'92238'     = LIS(/'Micro. vs. burnup for Isotope U-238'
              /E/f0:bu;e5: nd; micab; micfi; micnf; mickf1)

!idformsa=5!
nd          = SEL(nd/MIC/C1;$micALL2/$scalp/93239)
micab       = SEL(ab/MIC/C1;$micALL2/$scalp/93239)
micfi       = SEL(fi/MIC/C1;$micALL2/$scalp/93239)
mickf       = SEL(kf/MIC/C1;$micALL2/$scalp/93239)
micnf       = SEL(nf/MIC/C1;$micALL2/$scalp/93239)
il          = IND(mickf/RAN:;;1;)
mickf1      = FOR(mickf^il)
'93239'     = LIS(/'Micro. vs. burnup for Isotope Np-239'
              /E/f0:bu; e5: nd; micab; micfi; micnf; mickf1)

!idformsa=6!
nd          = SEL(nd/MIC/C1;$micALL2/$scalp/94239)
micab       = SEL(ab/MIC/C1;$micALL2/$scalp/94239)
micfi       = SEL(fi/MIC/C1;$micALL2/$scalp/94239)
mickf       = SEL(kf/MIC/C1;$micALL2/$scalp/94239)
micnf       = SEL(nf/MIC/C1;$micALL2/$scalp/94239)
il          = IND(mickf/RAN:;;1;)
mickf1      = FOR(mickf^il)
'94239'     = LIS(/'Micro. vs. burnup for Isotope Pu-239'
              /E/f0:bu; e5: nd; micab; micfi; micnf; mickf1)

!idformsa=7!
nd          = SEL(nd/MIC/C1;$micALL2/$scalp/94240)
micab       = SEL(ab/MIC/C1;$micALL2/$scalp/94240)
micfi       = SEL(fi/MIC/C1;$micALL2/$scalp/94240)
mickf       = SEL(kf/MIC/C1;$micALL2/$scalp/94240)
micnf       = SEL(nf/MIC/C1;$micALL2/$scalp/94240)
il          = IND(mickf/RAN:;;1;)
mickf1      = FOR(mickf^il)
'94240'     = LIS(/'Micro. vs. burnup for Isotope Pu-240'
              /E/f0:bu; e5: nd; micab; micfi; micnf; mickf1)

!idformsa=8!
nd          = SEL(nd/MIC/C1;$micALL2/$scalp/94241)
micab       = SEL(ab/MIC/C1;$micALL2/$scalp/94241)
micfi       = SEL(fi/MIC/C1;$micALL2/$scalp/94241)
mickf       = SEL(kf/MIC/C1;$micALL2/$scalp/94241)
micnf       = SEL(nf/MIC/C1;$micALL2/$scalp/94241)
il          = IND(mickf/RAN:;;1;)
mickf1      = FOR(mickf^il)
'94241'     = LIS(/'Micro. vs. burnup for Isotope Pu-241'
              /E/f0:bu; e5: nd; micab; micfi; micnf; mickf1)

!idformsa=9!

```

```

nd          = SEL(nd/MIC/C1;$micALL2/$scalp/94242)
micab       = SEL(ab/MIC/C1;$micALL2/$scalp/94242)
micfi       = SEL(fi/MIC/C1;$micALL2/$scalp/94242)
mickf       = SEL(kf/MIC/C1;$micALL2/$scalp/94242)
micnf       = SEL(nf/MIC/C1;$micALL2/$scalp/94242)
il          = IND(mickf/RAN:;;;1;)
mickf1      = FOR(mickf^il)
'94242'     = LIS(/'Micro. vs. burnup for Isotope Pu-242'
              /E/f0:bu; e5: nd; micab; micfi; micnf; mickf1)

```

!idformsa=10!

```

nd          = SEL(nd/MIC/C1;$micALL2/$scalp/95241)
micab       = SEL(ab/MIC/C1;$micALL2/$scalp/95241)
micfi       = SEL(fi/MIC/C1;$micALL2/$scalp/95241)
mickf       = SEL(kf/MIC/C1;$micALL2/$scalp/95241)
micnf       = SEL(nf/MIC/C1;$micALL2/$scalp/95241)
il          = IND(mickf/RAN:;;;1;)
mickf1      = FOR(mickf^il)
'95241'     = LIS(/'Micro. vs. burnup for Isotope Am-241'/E/
              f0:bu; e5:nd; micab; micfi; micnf; mickf1)

```

!idformsa=11!

```

ndx         = SEL(nd/MIC/C1;$micAbur/$scalp/5010)
vo          = SEL(vo/MIC/C1;$micALL/$scalp)
vox         = SEL(vo/MIC/C1;$micAbur/$scalp)
nd          = FOR(ndx*vox/vo)

micab       = SEL(ab/MIC/C1;$micAbur/$scalp/5010)
micfi       = SEL(fi/MIC/C1;$micAbur/$scalp/5010)
mickf       = SEL(kf/MIC/C1;$micAbur/$scalp/5010)
micnf       = SEL(nf/MIC/C1;$micAbur/$scalp/5010)
il          = IND(mickf/RAN:;;;1;)
mickf1      = FOR(mickf^il)
'5010'     = LIS(/'Micro. vs. burnup for Boron 10'
              /E/f0:bu; e5: nd; micab; micfi; micnf; mickf1)

```

!idformosa=12!

```

ndx         = SEL(nd/MIC/C1;$micAbur/$scalp/5011)
vo          = SEL(vo/MIC/C1;$micALL/$scalp)
vox         = SEL(vo/MIC/C1;$micAbur/$scalp)
nd          = FOR(ndx*vox/vo)

micab       = SEL(ab/MIC/C1;$micAbur/$scalp/5011)
micfi       = SEL(fi/MIC/C1;$micAbur/$scalp/5011)
mickf       = SEL(kf/MIC/C1;$micAbur/$scalp/5011)
micnf       = SEL(nf/MIC/C1;$micAbur/$scalp/5011)
il          = IND(mickf/RAN:;;;1;)
mickf1      = FOR(mickf^il)
'5011'     = LIS(/'Micro. vs. burnup for Boron 11'
              /E/f0:bu; e5: nd; micab; micfi; micnf; mickf1)

```

!<<< End of isotope micro. xs user input

>>>!

END()

## Appendix F. Zenith File: Reflector (znhwhref1.inp)

```

BEGIN ('Output generator for FORCIP-P : Baffle- Reflector Region' / 0.4; 0.4)

!----- Gustavo Alonso -----!
!----- Texas A&M University -----!
!----- Westinghouse 17x17 -----!

!=====!
!Case #      :      Descriptions                                     !
!-----!
! 1          : Base Conditions [Average Conditions @HFP ]         !
! 2          : Moderator Temperature Increase                     !
! 3          : Moderator Temperature Decrease                     !
! 4          : Boron Concentration Increase                       !
! 5          : Boron Concentration Decrease                       !
!-----!
!--- Enter your case selection -----!
%'Case#'      = 1

!----- Burnup steps for base case           : $BurnupBase   ---!
!----- Burnup steps for branch case        : $BurnupBranch  ---!
!----- Helios case                          : $Hcase         ---!
!----- Helios output file                   : $Hfile         ---!

$BurnupBase   = PAR(0)
$BurnupBranch = PAR(0)
$Hcase        = PAR('Vmhwhref')
$Hfile        = PAR('Vmhwhref.hrf')
C1            = IMP(HELIOS; $Hcase/$Hfile)

!--- For each case, enter the following info.: -----!
!----- Path                                     : $Path   ---!
!----- State                                    : $State  ---!
!----- Instantaneous fuel temperature          : tfuchg  ---!
!----- Reference fuel temperature              : tfuref  ---!
!----- Instantaneous moderator temperature    : tmochg  ---!
!----- Reference moderator temperature         : tmoref  ---!
!----- Instantaneous boron concentration      : ppmchg  ---!
!----- Reference boron concentration          : ppmref  ---!

%if ('Case#'.eq.1) then
  $Path      = PAR('Path')
  $State     = PAR('st')
  $Burnup    = PAR($BurnupBase)
  $scalp     = PAR($Path:( $State)$Burnup)

  tfuchg     = FOR(963)
  tfuref     = FOR(963)
  tmochg     = FOR(582.95)
  tmoref     = FOR(582.95)
  ppmchg     = FOR(1000)
  ppmref     = FOR(1000)
  CASEPARAM  = LIS(;;6/'Parameters'
                  /E/f0:tfuchg; tfuref; tmochg; tmoref; ppmchg; ppmref)

```

```

%endif

%if ('Case#'.eq.2) then
  $Path      = PAR('HMPath')
  $State     = PAR(sthiM)
  $Burnup   = PAR($BurnupBranch)
  $scalp    = PAR($Path:($State)$Burnup)

  tfuchg    = FOR(963)
  tfuref    = FOR(963)
  tmochg    = FOR(599.32)
  tmoref    = FOR(582.95)
  ppmchg    = FOR(1000)
  ppmref    = FOR(1000)
  CASEPARAM = LIS(;;6/'Parameters'
                  /E/f0:tfuchg; tfuref; tmochg; tmoref;ppmchg; ppmref)
%endif

%if ('Case#'.eq.3) then
  $Path      = PAR('LMPath')
  $State     = PAR(stloM)
  $Burnup   = PAR($BurnupBranch)
  $scalp    = PAR($Path:($State)$Burnup)

  tfuchg    = FOR(963)
  tfuref    = FOR(963)
  tmochg    = FOR(564.59)
  tmoref    = FOR(582.95)
  ppmchg    = FOR(1000)
  ppmref    = FOR(1000)
  CASEPARAM = LIS(;;6/'Parameters'
                  /E/f0:tfuchg; tfuref; tmochg; tmoref;ppmchg; ppmref)
%endif

%if ('Case#'.eq.4) then
  $Path      = PAR('HBPath')
  $State     = PAR(sthiB)
  $Burnup   = PAR($BurnupBranch)
  $scalp    = PAR($Path:($State)$Burnup)

  tfuchg    = FOR(963)
  tfuref    = FOR(963)
  tmochg    = FOR(582.95)
  tmoref    = FOR(582.95)
  ppmchg    = FOR(2000)
  ppmref    = FOR(1000)
  CASEPARAM = LIS(;;6/'Parameters'
                  /E/f0:tfuchg; tfuref; tmochg; tmoref;ppmchg; ppmref)
%endif

%if ('Case#'.eq.5) then
  $Path      = PAR('LBPath')
  $State     = PAR(stloB)
  $Burnup   = PAR($BurnupBranch)
  $scalp    = PAR($Path:($State)$Burnup)

  tfuchg    = FOR(963)

```



```

xb          = FOR(2.8575+7.89432)
$macREG    = PAR('macBARE')
$micREG    = PAR('micBARE')
$Curleft  = PAR('Cufuba')
$Curight  = PAR('Curef')
!---- END of USER INPUT -----!

dr          = SEL(dr/MAC/C1;$macREG/$scalp)
tr          = SEL(tr/MAC/C1;$macREG/$scalp)
p0         = SEL(p0/MAC/C1;$macREG/$scalp)
abs        = SEL(ab/MAC/C1;$macREG/$scalp)

flx        = SEL(flx/MAC/C1;$macREG/$scalp)
i1         = IND(flx/RAN:;;1;)
i2         = IND(flx/RAN:;;2;)
flxfast    = FOR(flx^i1)
flxthm     = FOR(flx^i2)
fxrat      = FOR((flxfast)/(flxthm))

micabbo10  = SEL(ab/MIC/C1;$micREG/$scalp/5010)
ndbo10     = SEL(nd/MIC/C1;$micREG/$scalp/5010)
macabbo10  = FOR(micabbo10*ndbo10)

micabbo11  = SEL(ab/MIC/C1;$micREG/$scalp/5011)
ndbo11     = SEL(nd/MIC/C1;$micREG/$scalp/5011)
macabbo11  = FOR(micabbo11*ndbo11)

micabbo    = FOR((macabbo10+macabbo11)/(ndbo10+ndbo11))
macabbo    = FOR(macabbo10+macabbo11)

jpa        = SEL(jp/CUR/C1;$Curleft/$scalp)
jma        = SEL(jm/CUR/C1;$Curleft/$scalp)

jpb        = SEL(jp/CUR/C1;$Curright/$scalp)
jmb        = SEL(jm/CUR/C1;$Curright/$scalp)

i12        = IND(p0/RAN:;;2;1)
i21        = IND(p0/RAN:;;1;2)
i1         = IND(abs/RAN:;;1;)
i2         = IND(abs/RAN:;;2;)
Diff       = FOR(dr/(3.0*tr))
srem       = FOR(p0^i12)
s21        = FOR(p0^i21)
ab1        = FOR(abs^i1)
ab2        = FOR(abs^i2)
D1         = FOR(Diff^i1)
D2         = FOR(Diff^i2)
s12        = FOR(srem-s21/fxrat)

jpa1       = FOR(jpa^i1)
jpa2       = FOR(jpa^i2)
jma1       = FOR(jma^i1)
jma2       = FOR(jma^i2)

```



```

!
! Homogenize region is bounded by surface a and b
!-----!

!----- the followings are calculated HELIOS -----!
! jpa1 : outward current at fuel-baffle interface (fast)
! jpa2 : outward current at fuel-baffle interface (thermal)
! jma1 : inward current at fuel-baffle interface (fast)
! jma2 : inward current at fuel-baffle interface (thermal)
! jpb1 : outward current at 1/2 reflector interface (fast)
! jpb2 : outward current at 1/2 reflector interface (thermal)
! jmb1 : inward current at 1/2 reflector interface (fast)
! jmb2 : inward current at 1/2 reflector interface (thermal)
!
! ab1 : absorption xs (fast)
! ab2 : absorption xs (thermal)
! s12 : downscatter
! d1 : diffusion coefficient (fast)
! d2 : diffusion coefficient (thermal)
!-----!

!--- USER INPUT -----!
! xb : width of the homogenized region
! $macREG : 2-group, macro. XS homogenized over region between a and b
! $micREG : 2-group, micro. XS homogenized over region between a and b
! $Curleft : 2-group currents on surface a
! $Curright : 2-group currents on surface b
!-----!

xb = FOR(10.75182)
$macREG = PAR('MacRE')
$micREG = PAR('MicRE')
$Curleft = PAR('Cubare')
$Curright = PAR('Curef')
!--- END of USER INPUT -----!

dr = SEL(dr/MAC/C1;$macREG/$scalp)
tr = SEL(tr/MAC/C1;$macREG/$scalp)
p0 = SEL(p0/MAC/C1;$macREG/$scalp)
abs = SEL(ab/MAC/C1;$macREG/$scalp)

flx = SEL(fx/MAC/C1;$macREG/$scalp)
i1 = IND(flx/RAN;;;1)
i2 = IND(flx/RAN;;;2)
flxfast = FOR(flx^i1)
flxthm = FOR(flx^i2)
fxrat = FOR((flxfast)/(flxthm))

micabbo10 = SEL(ab/MIC/C1;$micREG/$scalp/5010)
ndbo10 = SEL(nd/MIC/C1;$micREG/$scalp/5010)
macabbo10 = FOR(micabbo10*ndbo10)

micabbo11 = SEL(ab/MIC/C1;$micREG/$scalp/5011)
ndbo11 = SEL(nd/MIC/C1;$micREG/$scalp/5011)
macabbo11 = FOR(micabbo11*ndbo11)

micabbo = FOR((macabbo10+macabbo11)/(ndbo10+ndbo11))

```

```

macabbo          = FOR(macabbo10+macabbo11)

jpa              = SEL(jp/cur/C1;$Curleft/$scalp)
jma              = SEL(jm/cur/C1;$Curleft/$scalp)

jpb              = SEL(jp/cur/C1;$Curright/$scalp)
jmb              = SEL(jm/cur/C1;$Curright/$scalp)

i12              = IND(p0/RAN:;;;2;1)
i21              = IND(p0/RAN:;;;1;2)
i1               = IND(abs/RAN:;;;1;)
i2               = IND(abs/RAN:;;;2;)
Diff             = FOR(dr/(3.0*tr))
srem             = FOR(p0^i12)
s21              = FOR(p0^i21)
ab1              = FOR(abs^i1)
ab2              = FOR(abs^i2)
D1               = FOR(Diff^i1)
D2               = FOR(Diff^i2)
s12              = FOR(srem-s21/fxrat)

jpa1             = FOR(jpa^i1)
jpa2             = FOR(jpa^i2)
jma1             = FOR(jma^i1)
jma2             = FOR(jma^i2)

jpb1             = FOR(jpb^i1)
jpb2             = FOR(jpb^i2)
jmb1             = FOR(jmb^i1)
jmb2             = FOR(jmb^i2)

L1               = FOR((D1/(ab1+s12))^*0.5)
L2               = FOR((D2/ab2)^*0.5)
L1L2             = FOR((1/(L1**2))-(1/(L2**2)))
D12              = FOR(D1*D2)

coshxbL1        = FOR(0.5*((2.71828**(xb/L1))+(2.71828**(-xb/L1))))
sinhxbL1        = FOR(0.5*((2.71828**(xb/L1))-(2.71828**(-xb/L1))))
coshxbL2        = FOR(0.5*((2.71828**(xb/L2))+(2.71828**(-xb/L2))))
sinhxbL2        = FOR(0.5*((2.71828**(xb/L2))-(2.71828**(-xb/L2))))

ja1              = FOR(jpa1-jma1)
ja2              = FOR(jpa2-jma2)
jb1              = FOR(jpb1-jmb1)
jb2              = FOR(jpb2-jmb2)

fxhom1           = FOR((ja1*coshxbL1-jb1)/((D1/L1)*sinhxbL1))

term1            = FOR((L2/D2)*(ja2+s12*ja1/(D1*L1L2))*coshxbL2/sinhxbL2)
term2            = FOR((s12*L2/(L1L2*D12*sinhxbL2))*(-jb1))
term3            = FOR(-jb2/(D2*sinhxbL2/L2))
term4            = FOR((-s12*L1/(D12*L1L2*sinhxbL1))*(ja1*coshxbL1-jb1))

fxhom2           = FOR(term1+term2+term3+term4)

fxhet            = FOR(2*(jpa+jma))

```

```
fxhet1      = FOR(fxhet^i1)
fxhet2      = FOR(fxhet^i2)

adf1        = FOR(fxhet1/fxhom1)
adf2        = FOR(fxhet2/fxhom2)

ADFRF2      = LIS(/'ADF Baffle+Reflector'/E/f0:bu;e5:adf1;adf2)
POLRF2      = LIS(/'Macro. Baffle+Refl' /E/f0:bu;e5:D1;D2;s12;ab1;ab2;
                    micabbo; macabbo)

END()
```