JNC TN9400 2000-081

Analysis Results of Samples Reactivity Measurements at BFS-58-1i1 Assembly on the Basis of JNC Analytical System and BFS Traditional Approach

May 2000

O-ARAI ENGINEERING CENTER

JAPAN NUCLEAR CYCLE DEVELOPMENT INSTITUTE

〒319-1184　茨城県那珂郡東海村村松４番地49
核燃料サイクル開発機構
技術展開部　技術協力課

JNC TN9400 2000 – 081
May, 2000

Analysis Results of Samples Reactivity Measurements at BFS-58-1i1 Assembly on the Basis of JNC Analytical System and BFS Traditional Approach

Sergey Bednyakov *)

Abstract

BFS-58-1i1 assembly is a critical configuration constructed at BFS-2 fast critical facility, IPPE/ Obninsk/ Russia, containing uranium-free plutonium fuel in its central zone.

Sodium void reactivity effect, spectral averaged cross-sections, Doppler reactivity effect, material samples central reactivity worth (CRW) have been measured there. The results obtained by means of the last experimental technique is a subject of the current report.

In order to make adequate a comparison conditions for calculation and experiment a code for heterogeneous and bilinear corrections for central reactivity worth ratios based on IPPE developments has been written and applied.

Some extrapolated experimental data was also examined from the point of view of their consistency and reliability and a conclusion about it has been made.

The previous analysis results made by different laboratories have been included in the current report and the discrepancies have been discussed.

*) JNC International Fellow (11/01/2000 - 10/01/2001)
Reactor Physics Research Group, System Engineering Technology Division, O-arai Engineering Center, JNC, Japan

JNC TN9400 2000－081
２０００年　　５月

サイクル機構の解析システムとBFS固有の手法に基づく BFS-58-1i1炉心サンプル反応度解析

（研究報告書）

セルゲイ　ベドニアコフ*

要旨

BFS-58-1i1 炉心はロシアのオブニンスクにある物理エネルギー研究所（IPPE）の高速臨界実験装置 BFS-2 を用いて構成された臨界体系であり、体系中央にウラン無しプルトニウム燃料領域の存在することが特徴である。

BFS-58-1i1 炉心を用いて、ナトリウムボイド反応度、反応率比、ドップラー反応度、炉心中央における物質反応度が測定されたが、本報告書ではそのうちの炉心中央物質反応度実験値のデータ処理について取り上げる。

反応度実測値を解析値と比較しうる実験値に変換するために、IPPE において導出された非均質補正と直接・随伴中性子束を用いた Bilinear 補正を行うための手法を適用し、得られた実験値を整合性と信頼性の観点から検証した。

更に、以前に他の研究機関が示した解析結果についても言及し、今回得られた最新の結果との比較・検討を行った。

*: 大洗工学センター　システム技術開発部　炉心技術開発グループ
国際特別研究員（2000年1月11日～2001年1月10日）

CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1.
INTRODUCTION.

1. Introduction

BFS-58-1i1 assembly is a critical configuration constructed at BFS-2 fast critical facility, IPPE/ Obninsk/ Russia, containing uranium-free plutonium fuel in its central zone. A set of experiments has been carried out at this core to investigate its physical characteristics. It has included sodium void reactivity effect, spectral averaged cross-sections, Doppler reactivity effect, material samples central reactivity worth (CRW). The description of the configuration and the experimental results were given in /1,2/. The results obtained by means of the last experimental technique is a subject of the current report.

Usually the CRW ratios of the investigated materials are used for integral tests of nuclear data libraries especially in the cases when cross-sections of the materials can't be measured from an experiment directly. Besides, they are very useful as confirmation data for other methods like fission chamber or foil irradiation measurements, for instance. The reactivity worth measurements for materials with well-known cross-sections are also used for testing of computational methods and codes.

Rather simple experimental procedures on obtaining of CRW of the materials are accompanied by a complicated analysis of the experimental values. To provide a reasonable statistical uncertainties the dimensions and masses of the used samples have to exceed some minimal meanings which results in a definite self-shielding in the samples. To take into consideration this effect different size samples are used in measurements and extrapolation of the CRW to a zero size sample condition is fulfilled. The real experimental condition for the samples is a position in the inter-tube gaps between BFS tubes which contain at the same time pellets of different materials. So, correct heterogeneity model of the surrounding has to be constructed to carry out the adequate calculations of the CRW values. Besides, it has to be taken into account that ordinary reactor codes preparing effective group cross-sections for calculations treat only direct flux resonance structure neglecting by one of the adjoint flux which is correct for direct flux functional only. Special procedures to correct such calculations for CRWs which are both direct and adjoint flux functional based on the first order perturbation theory have been worked out by V. Doulin at BFS facility (so-called resonance bilinear corrections) /3,4/. Later the algorithm of heterogeneous and resonance bilinear corrections calculation as well as CRW size dependency calculation have been also described in /5/ also based on V.Doulin developments.

No adequate analysis has been carried out for these measurements on the basis of JNC analytical system so far because of its complexity and probably a lack of experience to carry out of such type of analysis.

The BFS traditional analytical approach used routinely for such measurements at IPPE is now applied on the basis of JNC nuclear data library and fast reactor analytical system.

CHAPTER 2.
PROCEDURE OF BFS CRW EXPERIMENTS ASSESSMENT.

2. Procedure of BFS CRW experiments assessment

To compare correctly calculation (R^{cal}) and experiment (R^{exp}) data for CRW ratios from the measurements based on small periodical core reactivity perturbation the adequate conditions for both sides (R^{cal} and R^{exp}) of such comparison have to be provided. It can be achieved by inserting appropriate corrections both to calculation and experiment. Usually this comparison is fulfilled for the condition of heterogeneous medium and infinitely thin sample. The experimental CRW ratios (R^{exp}) extrapolated to a zero sample size for heterogeneous medium have already been obtained in IPPE on the basis of /5/ algorithm and presented in /1/. U-235 has been chosen in this work for a normalization as a very well studied nuclide for reactor application.

To obtain the adequate calculation meanings (R^{cal}) the following procedures had to be done:

a) calculation of CRW ratios for homogeneous medium (R^{cal_hom});

b) calculation of heterogeneous corrections for CRW ratios (dR^{het}) which brings the homogeneous calculation to the heterogeneous condition of the experiment;

c) calculation of bilinear resonance corrections for CRW ratios (dR^{bil}) which takes into consideration the fact that in traditional reactor calculations only flux averaged group neutron data are applied as far as for correct analysis of such experiments the bilinear cross-sections averaging (with direct and adjoint flux) is required, so $R^{cal} = R^{cal\_hom} + dR^{het} + dR^{bil}$.

First, the CRW ratios (R^{cal_hom}, R^{cal_het}) and corrections ($dR^{het} = R^{cal\_het} - R^{cal\_hom}$, dR^{bil}) have been calculated in cell calculation. CASUP 1-D cell code was used for this. The plate-stretch cell model for CASUP input is presented in the Attachment 1.

Then CRW ratios (R^{cal_hom}) calculation has been repeated using direct and adjoint fluxes from CITATION 2-D diffusion reactor code.

The eventual CRW ratios from calculation used for comparison with the experiment were obtained as:

$$R^{cal} = R^{cal\_hom}(\text{reactor}) + dR^{het}(\text{cell}) + dR^{bil}(\text{cell}).$$

Calculation of CRW ratios for the nuclides was based on the first order perturbation theory. The main interrelations from this theory for homogeneous and heterogeneous plate-stretch cell model as well as formulas for calculation of resonance bilinear correction has been adopted from /5/.

JENDL 3.2 library was used for calculations. A lack of fission spectra data for fissile nuclides in JENDL 3.2 excluding U-235, U-238 and Pu-239 has caused a use of ABBN-78 library /6/ data. The 26-group ABBN-78 data were simply proportioned in 70-group JENDL 3.2 data according to the energy groups width.

A text of the code for the described algorithm has been written by the author using MS Visual Basic 6.0 and is attached as it was at the current moment in the Attachment 2 without optimization and removing of auxiliary operators necessary for the purpose of debugging.

CHAPTER 3.
COMPUTER DATABASE FOR BFS FACILITY AND EXPORT OF INPUT DATA.

3. Computer database for BFS facility and export of input data

A computer database for BFS experimental data utilization has been established and initial description of BFS-58-1i1 assembly has been done. The database itself is a topic of different report however it was used in the assessments carried out in the current work. To make it a common source of input data for different codes special interfaces was decided to develop from it. The interface for CASUP cell code, one of the first, has been created.

The BFS unit cells are described by a 1-D slab model with material layers infinitely stretched and parallel to each other. According to the export option the slab model can consist of N+ (1,...,N) layers where N is the number of the pellets in the cell and up to N fictitious layers are added to contain wrapper materials like wrapper tube, pellets covers, inter-tube bars materials as well as air in the inter-tube gaps. Each fictitious layer is placed between main materials and can take an arbitrary place in the cell. In the current analysis two fictitious layers have been arranged and each of them was placed in the middle between two Pu layers in the cell.

The database has generated the input file (see Attachment 1) according to main principles held in the cell model arrangement: conservation of the main materials density (excluding wrapper layers), all materials volumes and the total cell height.

CHAPTER 4.
DISCUSSION OF THE RESULTS.

4. Discussion of the results

The described above scheme for the first order perturbation calculations of CRW ratios for homogeneous and plate-stretch geometry as well as assessment of resonance bilinear corrections has been applied in this work and the results are presented in Table 4.1.

Table 4.1. Calculated CRW ratios for heterogeneous medium structure and bilinear type of cross-sections weightning for zero size sample according to the described scheme (normalization to U-235)

Material	Experiment /1/ (extrapolated to l=0)	Calculation (CASUP/CITATION/Attachment 2)		
		homogeneous without corrections	heterogeneous corrections inserted	heterogeneous and bilinear corrections inserted
H	0.221 ± 0.002	0.322	0.322	0.332
Na	0.0079 ± 0.0003	0.00847	0.00929	0.01016
C	0.0126 ± 0.0001	0.0154	0.0167	0.0169
B-10	-4.14 ± 0.08	-3.21	-3.996	-3.993
Pu-239	1.16 ± 0.04	1.21	1.17	1.19
Pu (68%Pu239)	0.622 ± 0.020	0.807	0.675	0.726
Pu (78%Pu239)	0.76 ± 0.01	0.910	0.797	0.842
Pu (89%Pu240)	-0.89 ± 0.05	-0.326	-0.749	-0.623
Pu (59%Pu241)	1.18 ± 0.02	1.20	1.17	1.23
U-238	-0.85 ± 0.06	-0.479	-0.777	-0.762
Np-237	-2.32 ± 0.03	-1.73	-2.22	-2.18
Am-241	-2.42 ± 0.03	-1.78	-2.25	-2.19

The table is giving an idea about meanings of heterogeneous and bilinear corrections for the main reactor materials. The first notice which can be made is big calculation corrections for majority of materials. Especially it concerns to B-10 and threshold fissile materials. But for the other elements they are also not negligible. For pure Pu-239 and 59%-Pu241 the values of the corrections are on the level of the experimental uncertainty. The second notice is that for majority of materials the heterogeneous corrections are much bigger then bilinear ones. One have to take into consideration that for typical fast reactor spectra the uncertainties of such corrections have been studied /7/ and estimated as about 20-25% from the meaning of the correction itself. That was the reason to show in the last coloumn of the Table 4.2 (in parenthesis) a full uncertainties including assessment of corrections.

The Table 4.2 gives a comparison of calculation and experiment of CRW ratios obtained in the current work as well as the results of such analysis made by other laboratories.

It is seen that for B-10, Np-237, U-238, Pu-239 CRW ratios the calculation is equal to experiment in frames of the cited uncertainties. The results for them (excluding U-238 in CEA evaluation) are corresponding to each other in different laboratories.

Am-241 data have been underestimated by all teams and they are also corresponding to each other.

The experiments with plutonium isotope mixtures have been mainly overestimated by JNC (89%-Pu240 is an exception). Never the less they are agreed with IPPE ABBN93 based results for 68%-Pu239 and 78%-Pu239 mixtures.

Table 4.2. Calculation and experiment comparison for CRW ratios (normalization to U-235)

Material	Experiment /1/ (extrapolated to l=0)	C/E			
		CEA /1/		IPPE /1/	JNC
		JEF2	ERALIB1	ABBN93	JENDL3.2
H	0.221 ± 0.002	-	-	0.94	1.50 ± 0.9 (1.5) %
Na	0.0079 ± 0.0003	1.23	1.22	0.72	1.29 ± 3.8 (6) %
C	0.0126 ± 0.0001	1.98	2.14	0.95	1.34 ± 0.8 (3) %
B-10	-4.14 ± 0.08	0.93	0.98	0.97	0.96 ± 1.9 (5) %
Pu-239	1.16 ± 0.04	0.99	1.05	1.01	1.03 ± 3.4 (4) %
Pu (68%Pu239)	0.622 ± 0.020	0.86	0.71	1.14	1.17 ± 3.2 (4.5) %
Pu (78%Pu239)	0.76 ± 0.01	0.89	0.98	1.07	1.11 ± 1.3 (2.5) %
Pu (89%Pu240)	-0.89 ± 0.05	1.52	1.50	1.04	0.70 ± 5.6 (10) %
Pu (59%Pu241)	1.18 ± 0.02	0.93	0.99	0.93	1.04 ± 1.7 (1.8) %
U-238	-0.85 ± 0.06	1.23	1.28	1.03	0.90 ± 7.1 (11) %
Np-237	-2.32 ± 0.03	0.88	0.92	0.96	0.94 ± 1.3 (5) %
Am-241	-2.42 ± 0.03	0.86	0.90	0.89	0.91 ± 1.3 (4.5) %

The CRW ratios for all typical scatterers (H, C, Na) have been greatly overestimated in this work. Only IPPE calculations based on ABBN93 data library concerning H and C looks rather satisfactory. The CEA results looks more preferable for Na though they are also too overestimated. Taking into consideration the values of the discrepancies and uncertainties it is possible to say that JNC assessment of Na CRW ratio is almost agree with the CEA data.

In the current consideration the following moments should be taken into account. The spectrum of the central part of the core is untypical for usual fast reactors, it is rather soft and covers a more wide resonance part of neutron cross-sections for materials. It causes a large self-shielding effects for the materials in the core and is a source of additional uncertainty at the analysis. To demonstrate these effects the experimental dependencies of CRW ratios on the sample size for different elements as well as CRW ratios extrapolated to a zero size sample (N – nuclear density , l – mean chord length) are presented on Figs. 4.1-4.13. The numbers near experimental points on the figures give their ratios to the extrapolated value. The dotted lines connecting the experimental data are given only to indicate the main tendency of the data set. The real calculational curves have been obtained on the basis of Wigner approximation /8/ and algorithm /9/ developed at BFS in IPPE and not presented in this work. They were used for extrapolation of the real experiments to zero size sample condition. The large corrections to zero size sample condition are very well visible. The bright examples of the very large self-shileding corrections are U-238 (Fig. 4.2) and 89% Pu-240 (Fig. 4.12). So, for U-238 the extrapolated CRW ratio for the single sample is ~5 times more then the real experimental value. For 89% Pu-240 the CRW ratios for the most thin and the most thick samples are corrected to ~4 times and ~8 times larger values. Such a big extrapolation corrections may cause even bigger uncertainties then it is cited for the extrapolated data.

The other thing should be taken into account to explain some unclear C/E meanings and in case of a lack of information. It can be demonstrated on the basis of pure Pu-239 (Fig. 4.9) and 78% Pu-239 (Fig. 4.10) experiments. The experimental dependency on size sample of the CRW ratios for the real samples is not plain and this is the case when an interpreter has to decide how to treat such data. Surely, such inconsistency is a result of experiment errors. Neglecting the falling out data (the most thick sample of pure Pu-239 and the intermediate

sample of 78% Pu-239) it is easy to obtain C/E values for these materials which equal to ~1.00 for pure Pu-239 and ~1.05 for 78% Pu-239 for JNC analysis. And only CEA analysis for 78% Pu-239 is suffering from such a procedure.

There are C/E values for plutonium CRW ratios of different isotopic content on Fig. 4.14. The only conclusion which is easily resulting from it is that for mainly fissile materials all the data are overestimated in different extent. For the threshold fissile material (89% Pu-240) the CRW ratio is underestimated. As far as the CRW ratio value is negative for this sample the conclusion that for all plutonium CRW ratios their calculations are more positive then experiments can be made.

It is of interest to compare C/E discrepancies obtained in this work with the ones from SEG facility. The data from SEG experiments analysis have been taken from /10/ and are presented in Table 4.3. The uncertainties of the cited SEG data according to /10/ roughly can be accepted around 10% (relative) with unessential variation.

A parameter characterizing the spectra softness was adopted as their neutron part in the energy region of elastic slowing down. Approximately, it is a region below 9.1188 keV (upper boundary of 29 group in 70-group energy scale).

Table 4.3. C/E values for some CRW ratios obtained at BFS-2 and SEG facilities (normalization to U-235)

Assembly	Part of neutrons below 9.1188 keV, %	C/E, JENDL 3.2			
		R10/R235	R238/R235	R12/R235	R1/R235
BFS-58-1i1	~21	0.97± 1.9 (5) %	0.90± 7.1 (11) %	1.36± 0.8 (3) %	1.53± 0.9 (1.5) %
SEG-4	~21	0.75	0.74	-	-
SEG-5	~19	0.76	-	-	-
SEG-7A	~23	0.83	-	1.11	-
SEG-6/EK45	~1.6	0.95	0.98	0.98	1.11

The comparison of the discrepancies mainly shows a contradiction of the BFS and SEG results excepting SEG-6/EK45 assembly where CRW for B-10 and U-238 relatively to U-235 are agreed with BFS-58-1i1 ones. Though exactly these assemblies are very different by the neutron fraction in the slowing down region. Probably in this case the spectra form below 9.1188 keV also has an importance.

It is clear that no far-reaching conclusions should be made now on the basis of such comparison because of many effects and factors were treated and processed by a different way (for instance, 18- and 70-group energy scale, etc.). The assemblies have also specific configurations and content. SEG has a graphite coloumn in the core center surrounded with uranium fuel where graphite eliminates a resonance features from the neutrons coming from the fuel region as well as BFS-58-1i1 contains a uranium-free plutonium fuel without any irregularity inserted in the place of measurement. There is a very representative example of C/E data lamination at the BFS and KOBR spectra softening /11/ which was explained in later works. The C/E (calculations were based on ABBN-78 library) for CRW ratios for Pu-239, B-10 and H depending on the spectra softness parameter for assemblies with U-Pu or U fuel have been changing strongly and are presented in Table 4.4. There are some softness spectrum parameters typical for some types of reactors under the table to get an impression about investigated spectral region in the whole.

Table 4.4. Average levels of discrepancies between calculation and experiment for the array of BFS data for CRW ratios of main reaction standards (normalization to U-235) according to reference /11/.

Specrum softness parameter,*) %	Fuel	H E-C	B-10 (E-C)/E, %	Pu-239 (E-C)/E, %
0 - 7	U	~ 0.01	~ 7	~ 0
	U-Pu	~ 0.06	~ 7	~ 0
7 - 15	U	~ 0.01	~ 0 to 15 (including KOBR cores)	~ -2
	U-Pu	~ 0.06	~ 15	~ 4

*) for metal fuel breeder spectrum the softness parameter is less then 1%, for BN-600 it is ~ 5%, for BN-1600 it is ~ 8%.

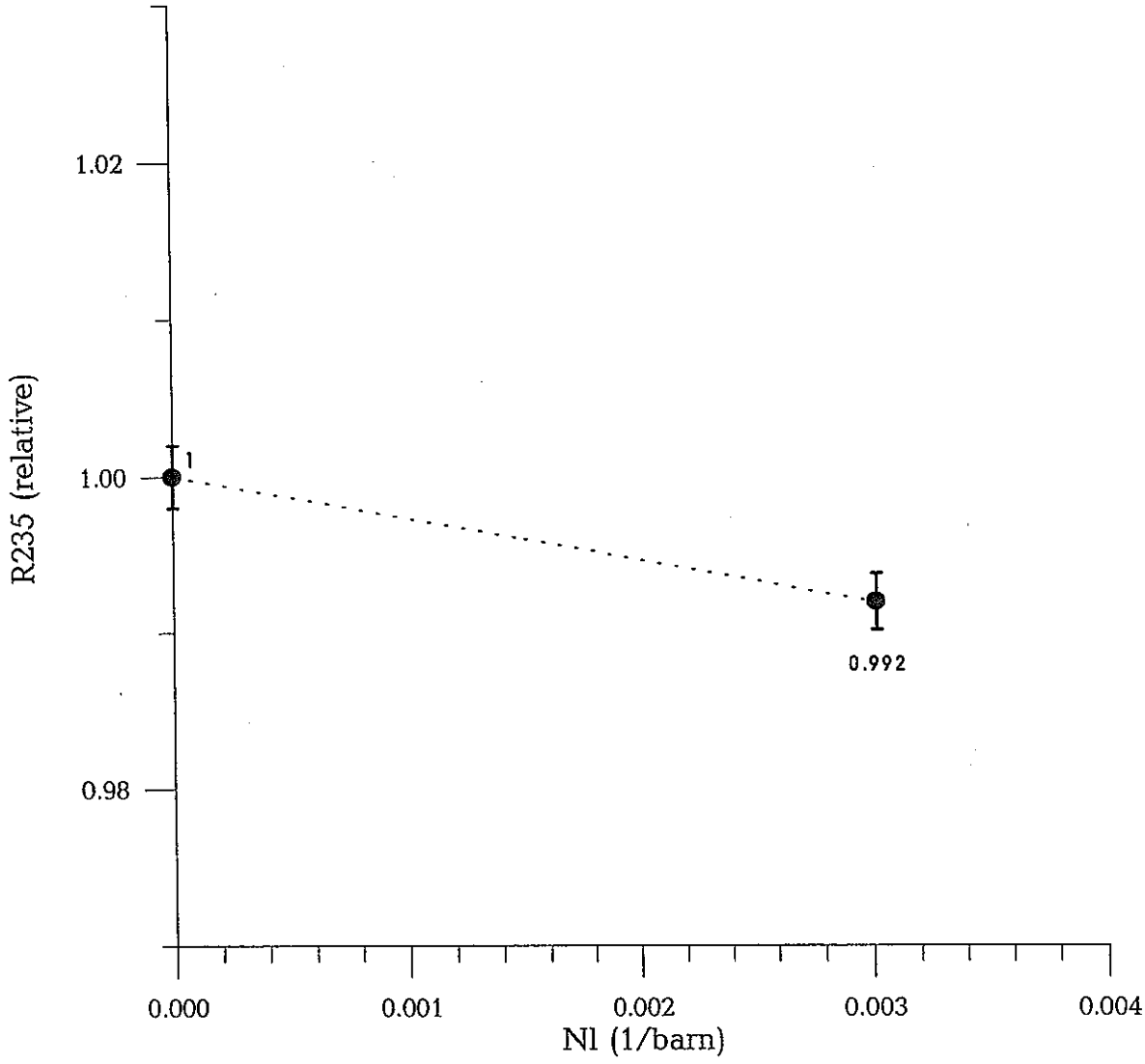

Fig. 4.1. Experimental relative U-235 CRW ratios.

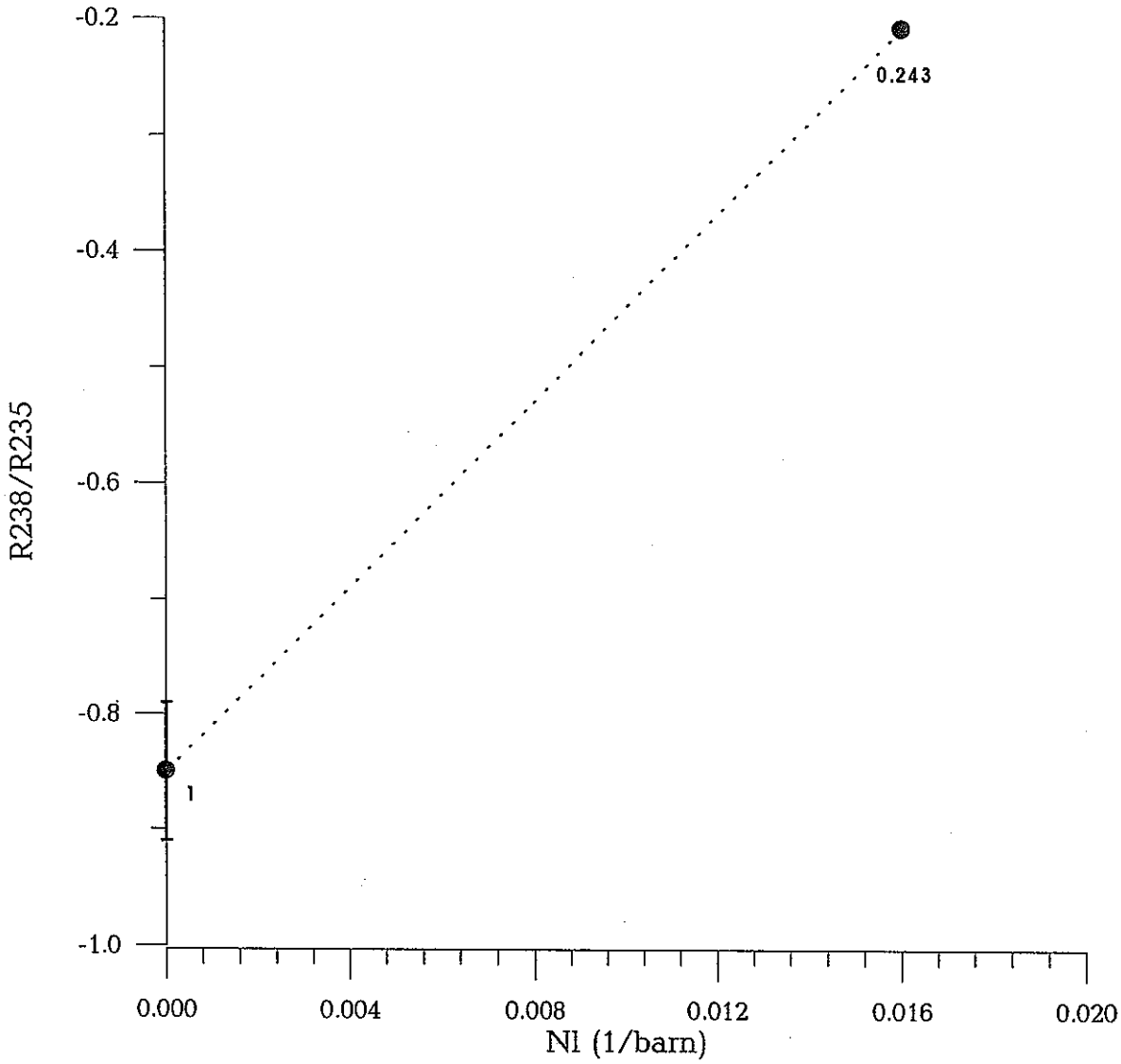

Fig. 4.2. Experimental U-238 to U-235 CRW ratios.

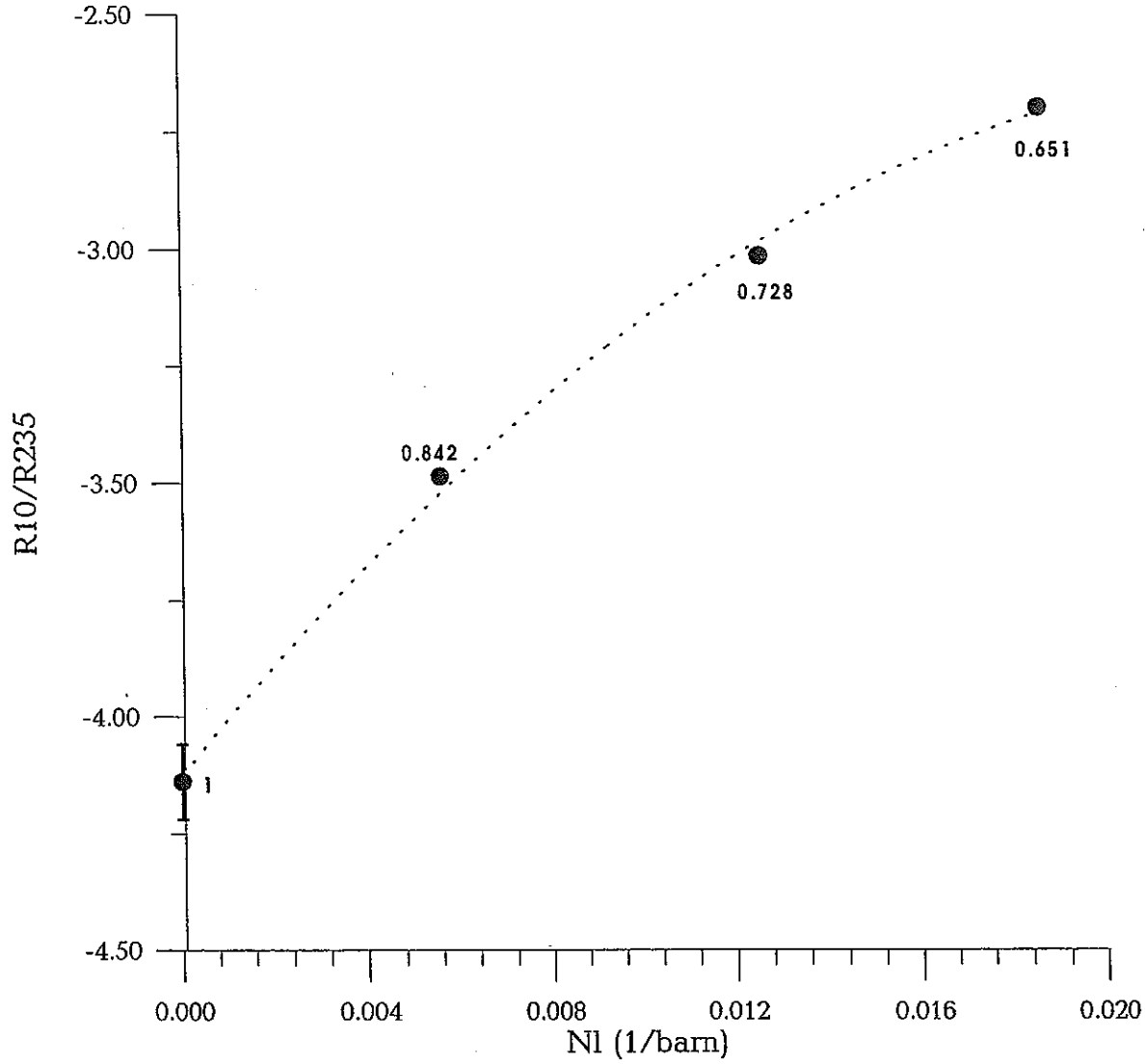

Fig. 4.3. Experimental B-10 to U-235 CRW ratios.

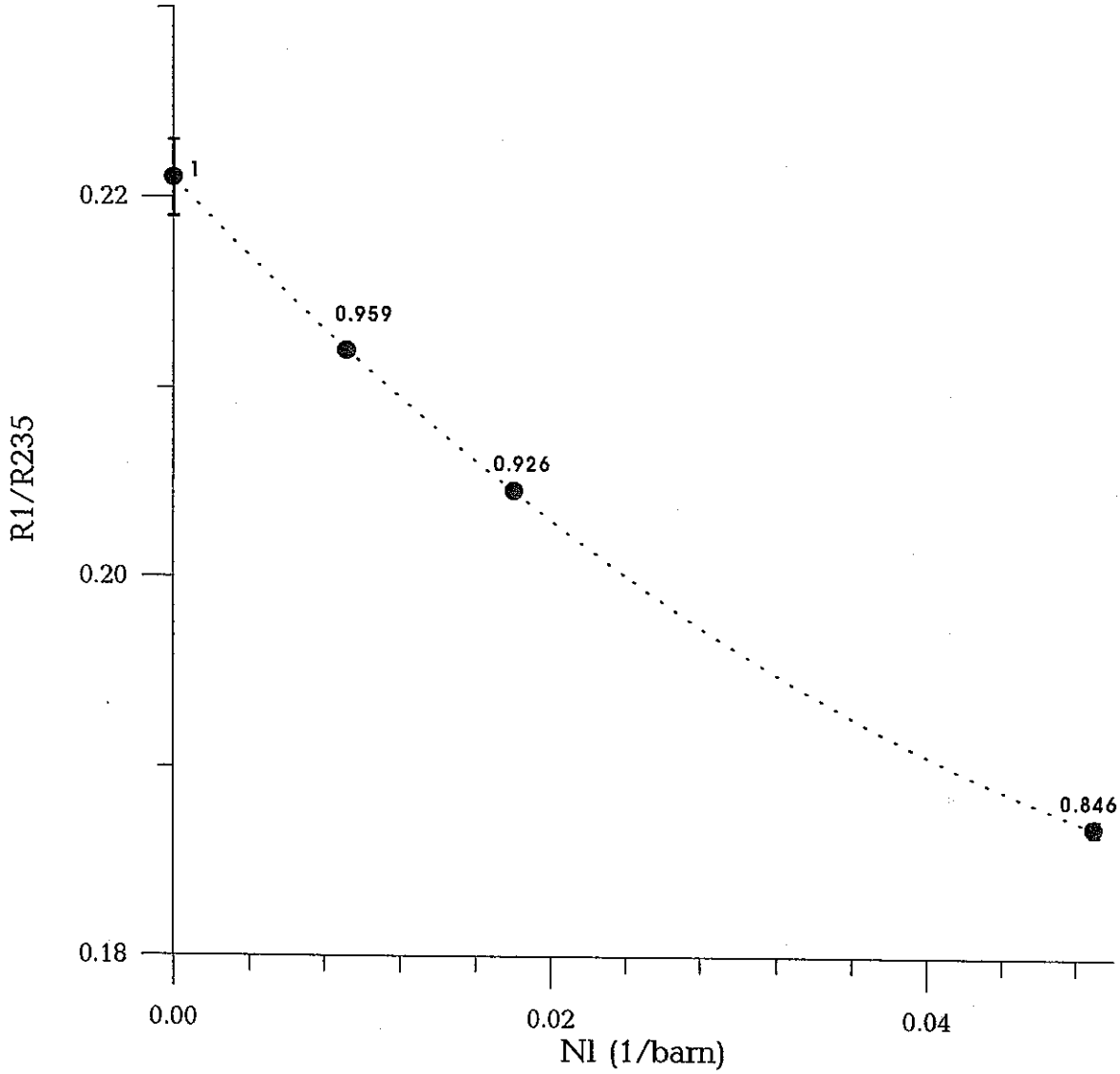

Fig. 4.4. Experimental H to U-235 CRW ratios.

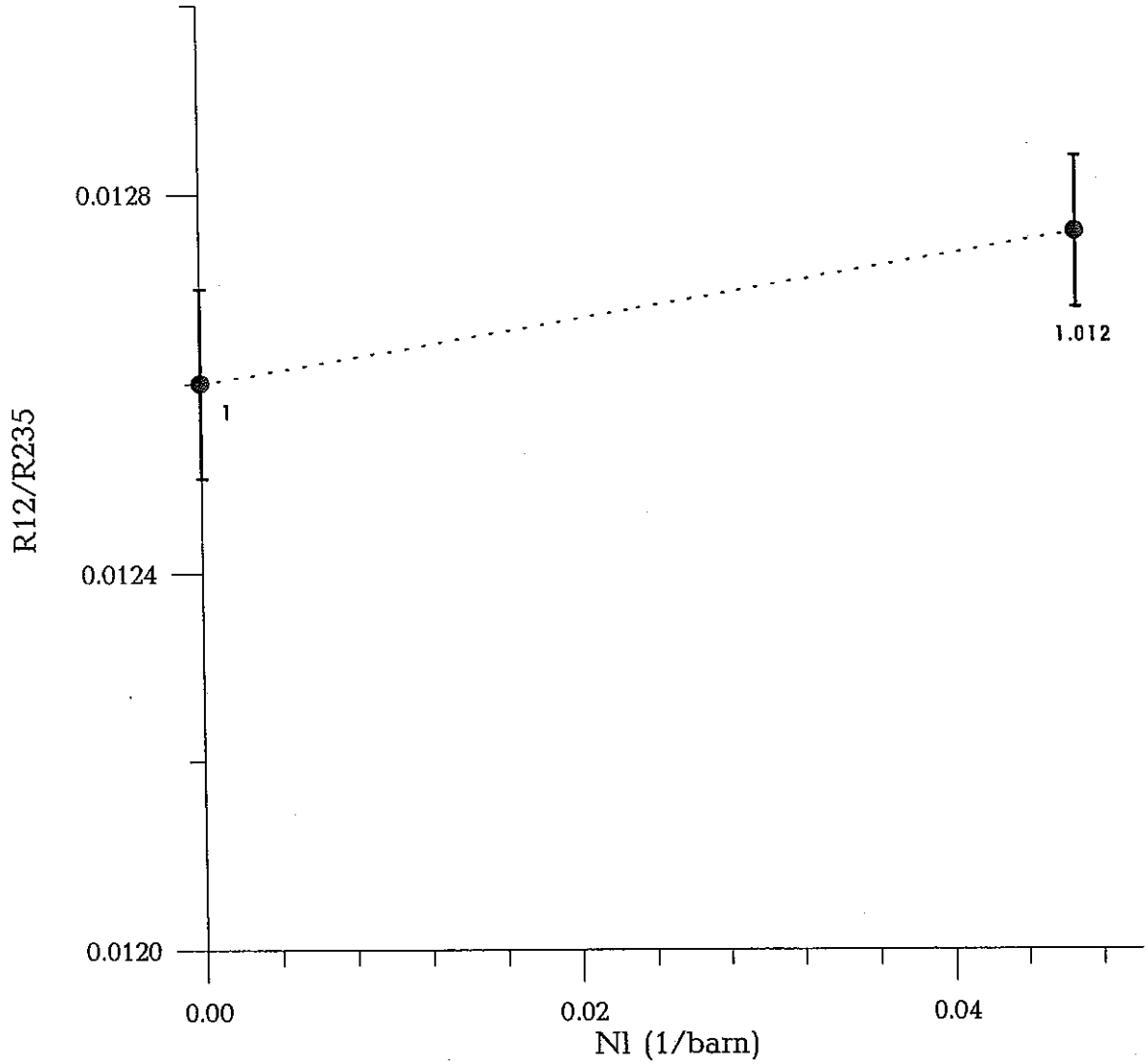

Fig. 4.5. Experimental C-12 to U-235 CRW ratios.

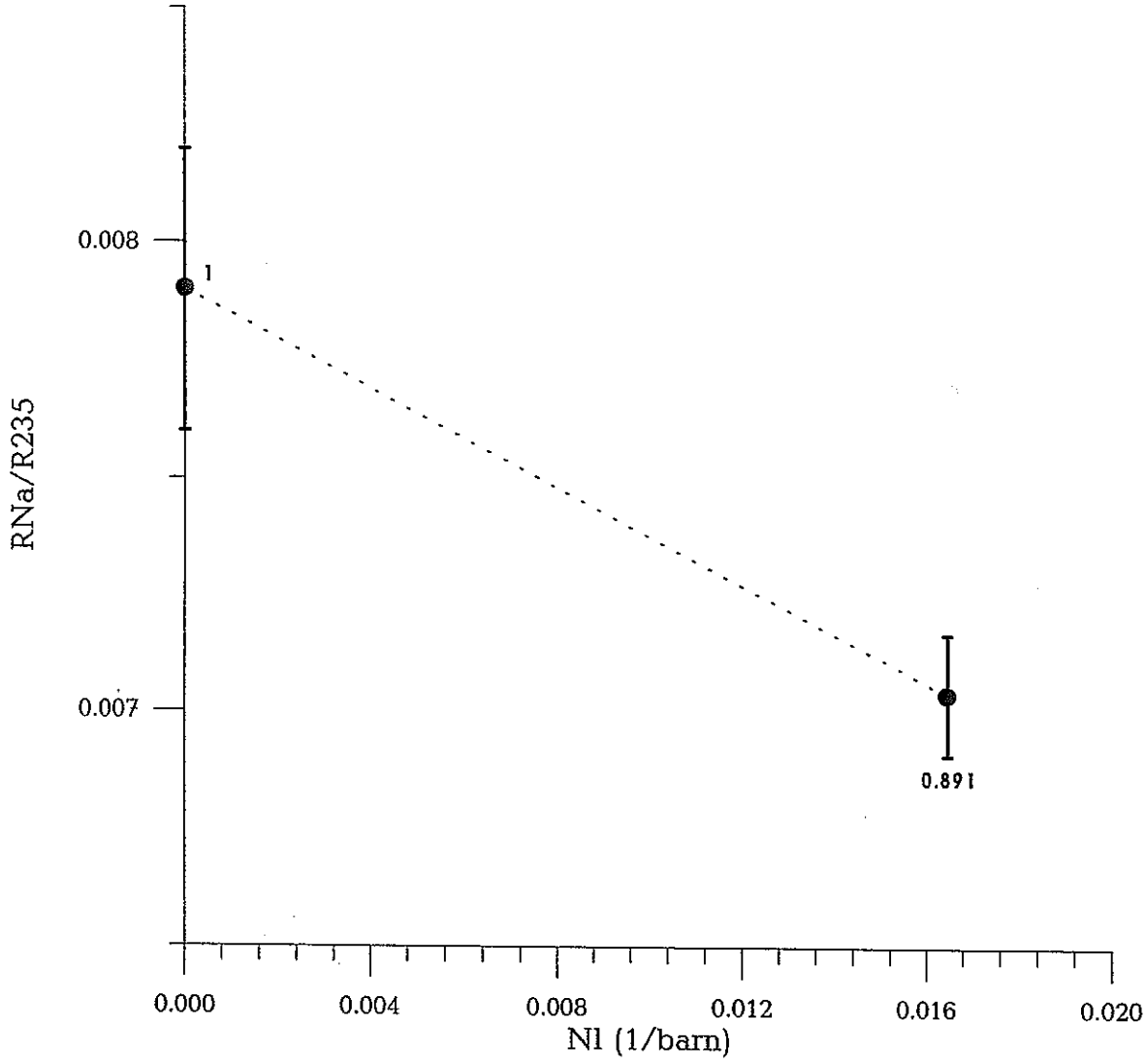

Fig. 4.6. Experimental Na to U-235 CRW ratios.

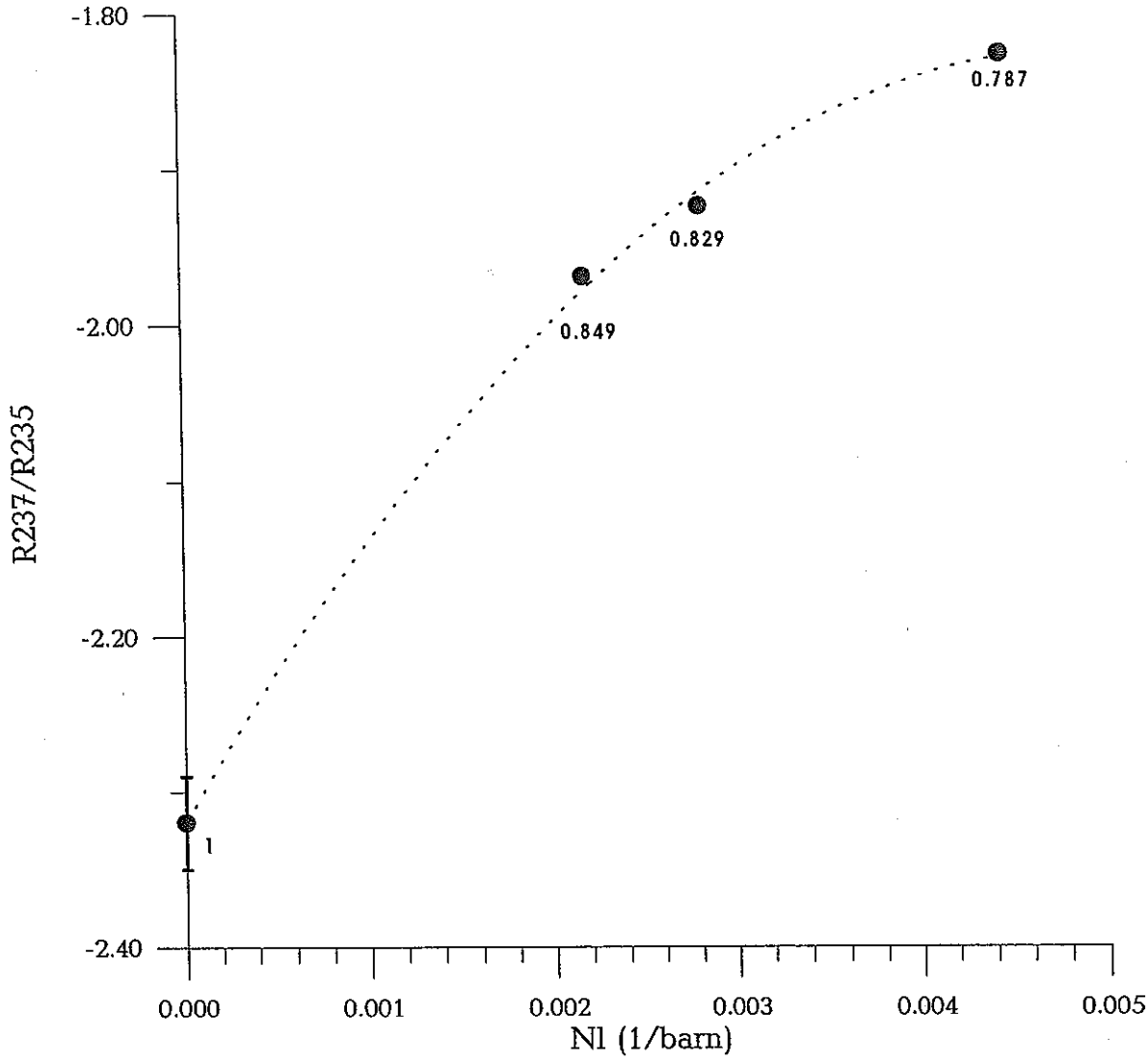

Fig. 4.7. Experimental Np-237 to U-235 CRW ratios.

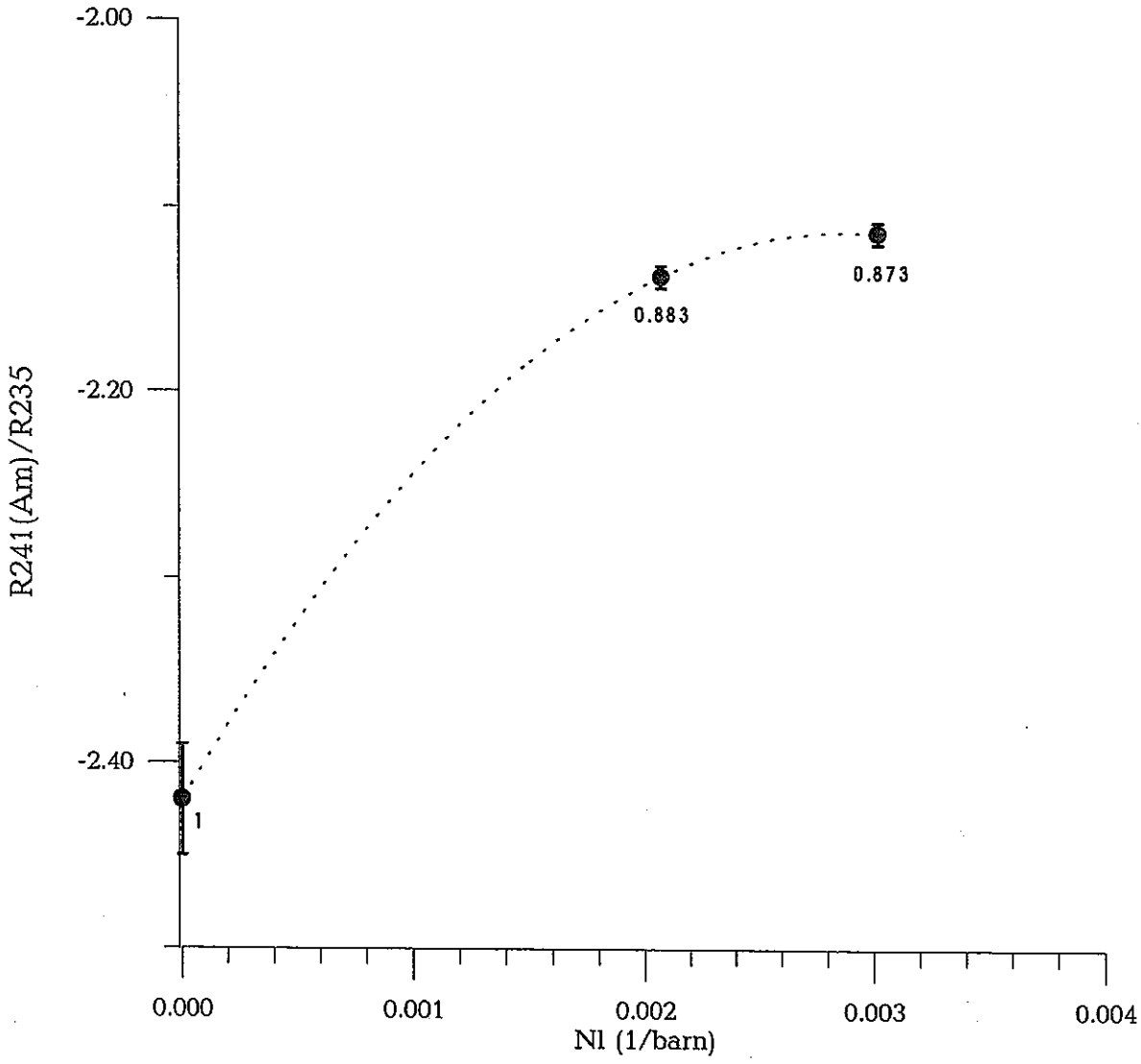

Fig. 4.8. Experimental Am-241 to U-235 CRW ratios.

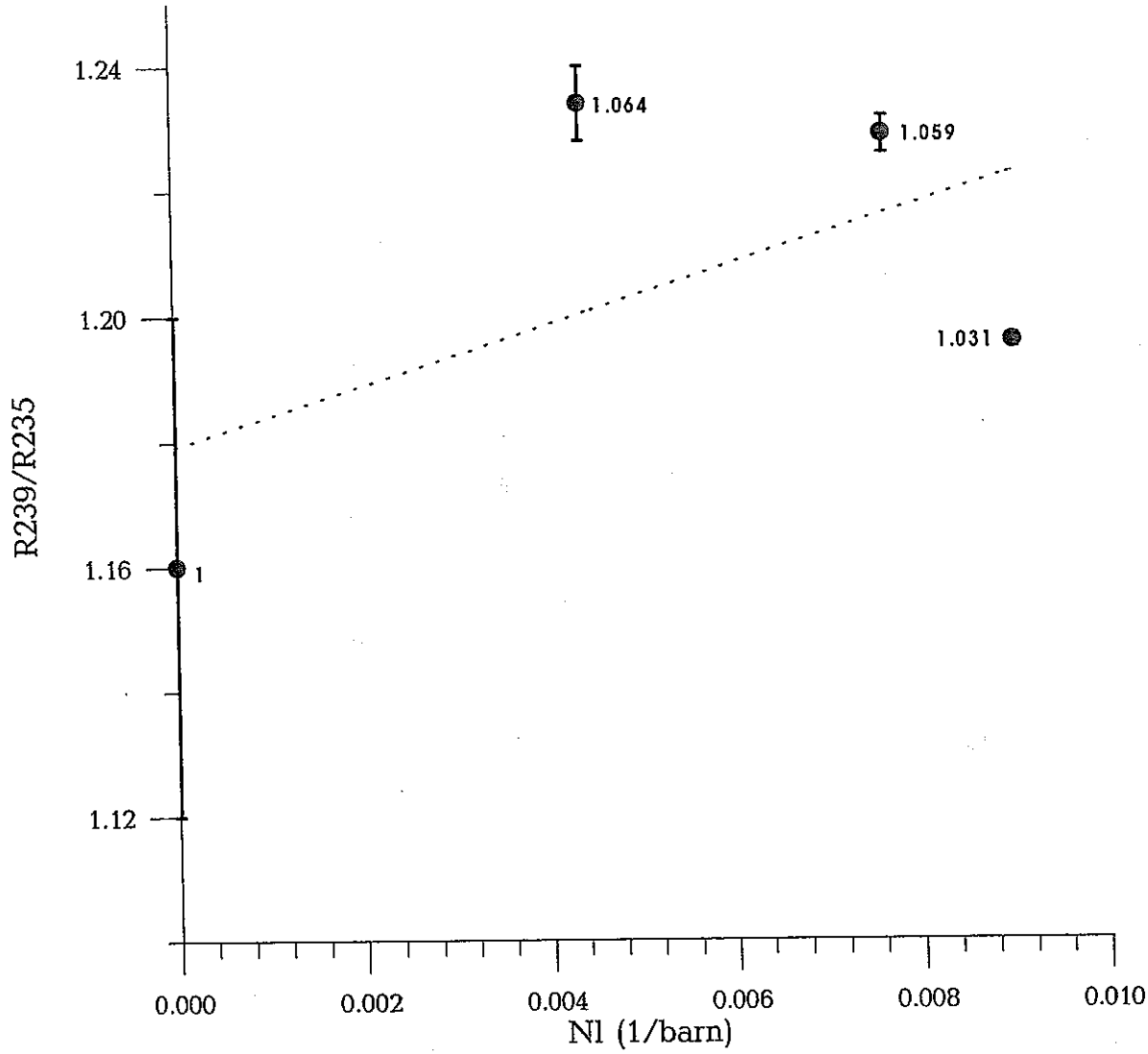

Fig. 4.9. Experimental Pu-239 to U-235 CRW ratios.

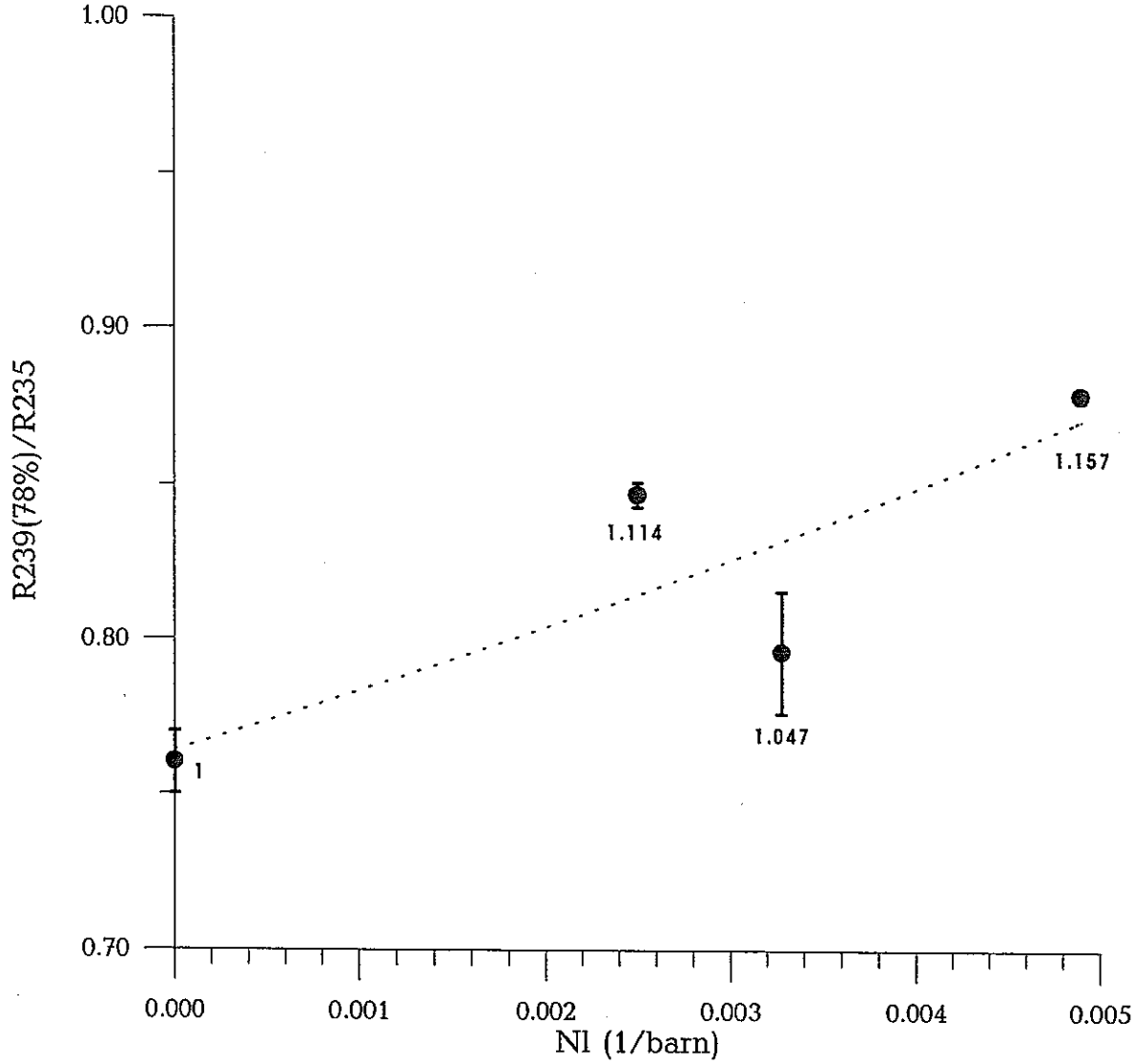

Fig. 4.10. Experimental Pu (78% Pu-239) to U-235 CRW ratios.

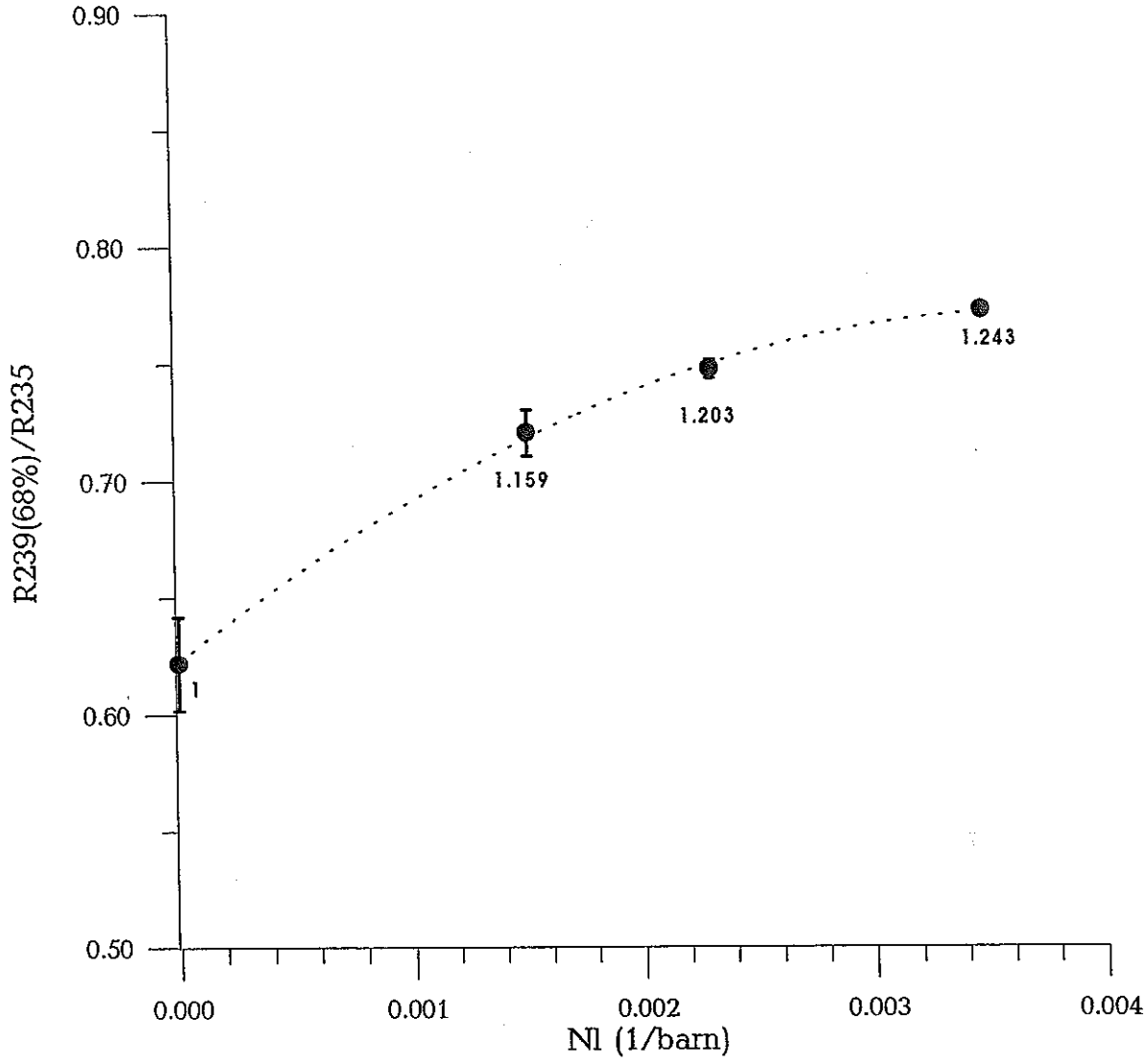

Fig. 4.11. Experimental Pu (68% Pu-239) to U-235 CRW ratios.

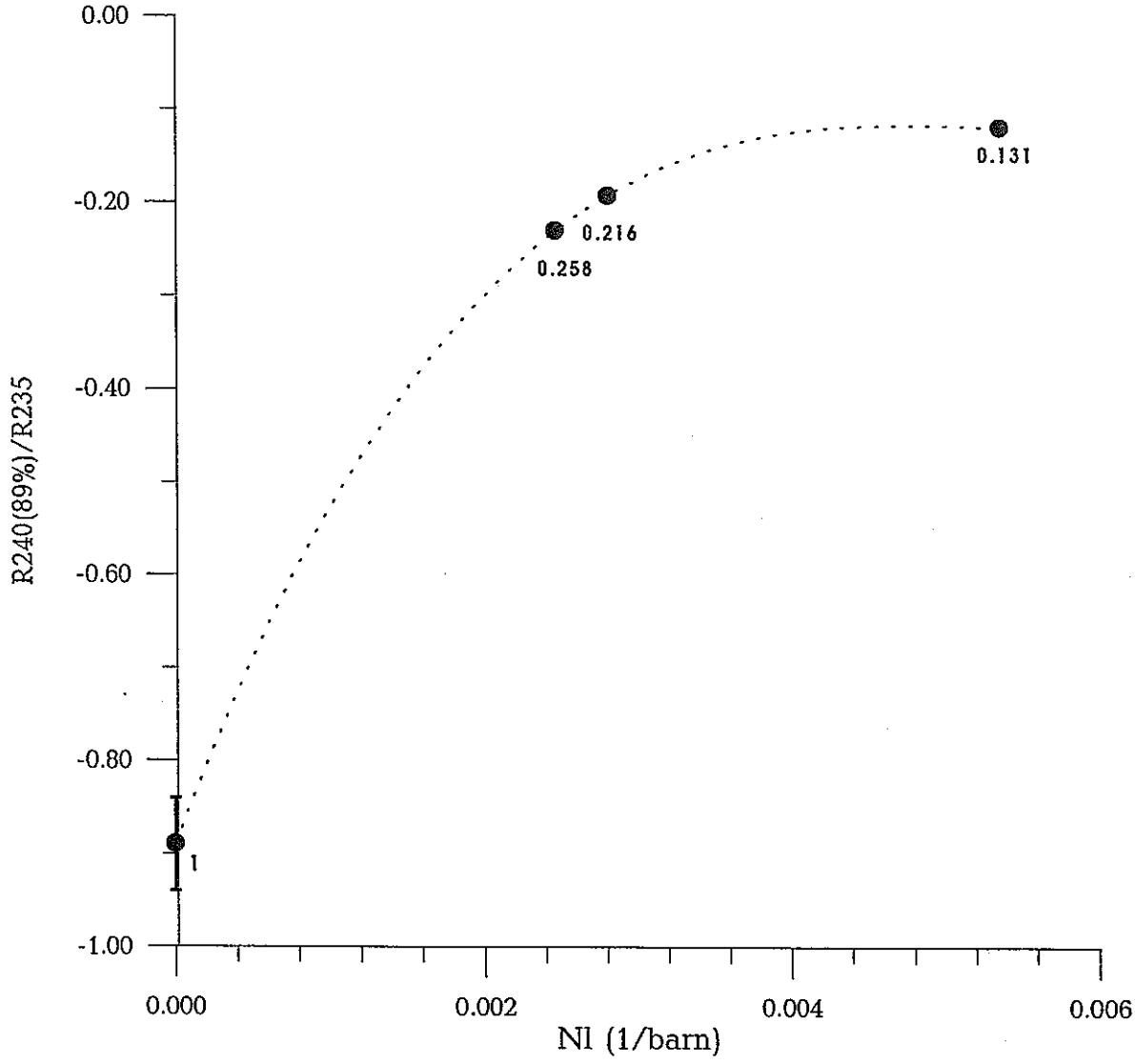

Fig. 4.12. Experimental Pu (89% Pu-240) to U-235 CRW ratios.

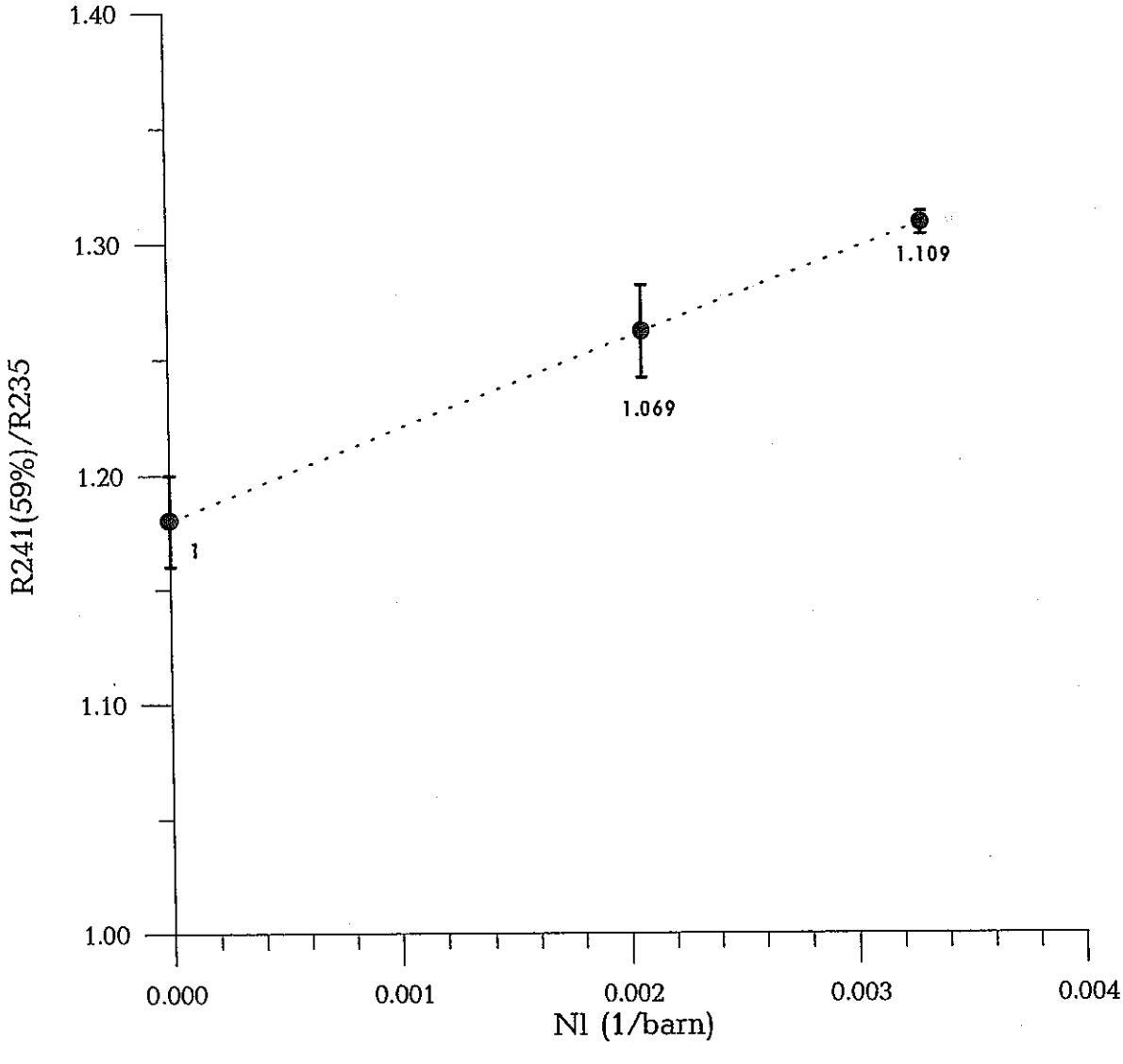

Fig. 4.13. Experimental Pu (59% Pu-241) to U-235 CRW ratios.

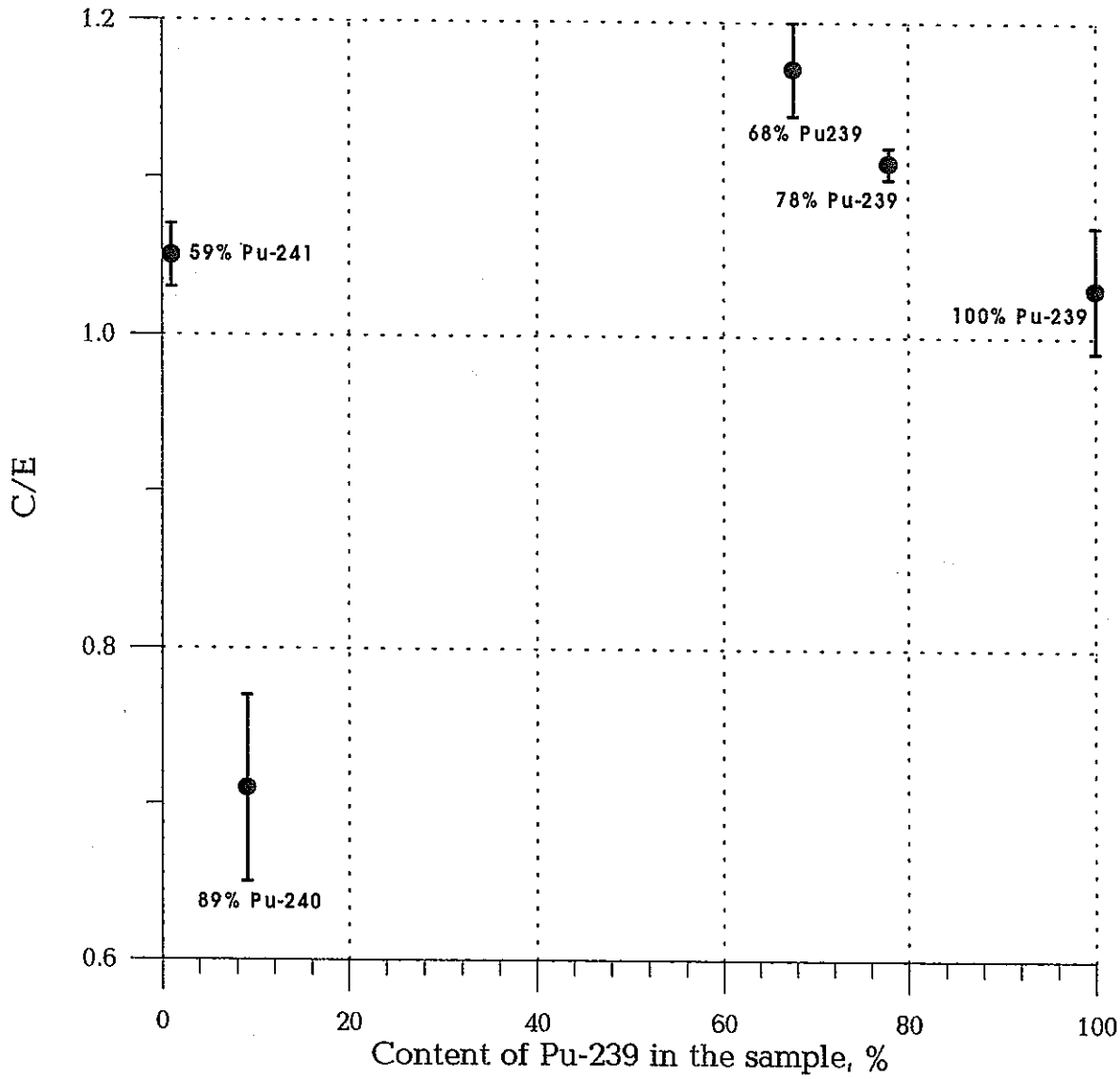

Fig. 4.14. C/E values for Pu CRW ratios for zero sample size. Normalization to U-235.

CHAPTER 5.
CONCLUSION.

5. Conclusion

1. First order perturbation calculations of the materials CRW ratios for BFS-58-1i1 assembly including assessment of heterogeneous and resonance bilinear corrections have been carried out. JNC tools were used for calculation of micro- and macro- cross-sections as well as direct and adjoint fluxes (CASUP, CITATION) and for neutron data library (JENDL3.2) access. BFS approach was used for assessment of heterogeneous and resonance bilinear corrections.

2. A computer database for BFS critical facilities has been established as a common source for the experiments analysis by different computer codes. BFS-58-1i1 cell was described in it and cell plate-stretch model was generated from it to CASUP cell code.

3. A comparison of calculation and experiment CRW ratios has been carried out and intercomparison with other laboratories analysis has been fulfilled.

For B-10, Np-237, U-238, Pu-239 CRW ratios the calculation is equal to experiment in frames of the cited uncertainties. The results for them (excluding U-238 in CEA evaluation) are corresponding to each other in different laboratories.

Am-241 CRW ratio has been underestimated by all teams and they are also corresponding to each other.

The CRW ratios for all typical scatterers (H, C, Na) have been greatly overestimated in this work.

The experiments with plutonium isotope mixtures have been mainly overestimated by JNC (89%-Pu240 is an exception). Never the less they are agreed with IPPE ABBN93 based results for 68%-Pu239 and 78%-Pu239 mixtures.

4. One has to be careful to use the data with the very big corrections (sample self-shielding, heterogeneous and bilinear effects). The BFS-58-1i1 mustn't be the only assembly to make conclusions for this type of spectrum and core content.

More detail experimental information should be taken into account in case of analysis of experiments in such complicated conditions (see examples on Figs. 9, 10) in order to convince in the consistency of the analyzed data.

More experimental data for different spectra (assemblies) are necessary to have a reliable systematic array in order to justify used neutron data and codes..

5. The weak places of the preliminary analysis can be eliminated in future and involve the following items:

- first order perturbation calculations for different materials were made on the first stage using direct and adjoint flux from CITATION by the algorithm presented in Attachment 2 but in future it is desirable to apply a reactor code (PERKY) included in JNC analytical system for such calculations;
- the code for assessment of heterogeneous and bilinear corrections has a very small statistical ground for the moment to justify the assessment results.

In spite of these deficiencies the conclusions mustn't have dramatic changes and only some details can be corrected.

The author is conserving a place for the further analysis in case of any serious arguments will be found in its favour and also for future nuclear data library adjustment based on the fulfilled analysis.

REFERENCES.

References

[1] S.Hunter. Analysis of measurements for a uranium-free core experiment at the BFS-2 critical assembly. - JNC TN9400 99-049, April, 1999.

[2] I.Krivitsky et al. Calculational analysis of the critical assembly BFS-58-1-i1 simulating a fast reactor core with uranium-free fuel. - ENC'98, Nice, 1998.

[3] V.Doulin. Accuracy assessment of group calculations of reactors criticality perturbations. – Voprosy atomnoi nauki i tekhniki, (Yadernye konstanty), 1983, issue 1 (in Russian).

[4] V.Doulin. About assessment of reactors criticality perturbation experiments. – Voprosy atomnoi nauki i tekhniki, (Yadernye konstanty), 1988, issue 2 (in Russian).

[5] S.Bednyakov. Realization of integral experiments assessment method for BFS facility conditions. – Preprint IPPE – 2114, 1990 (in Russian).

[6] L.Abagyan et al. Group constants for reactor and shield calculations. – Moscow, Energoizdat, 1981 (in Russian).

[7] S.Bednyakov et al. Experimental validation of methods of assessment of fast assemblies criticality perturbations by small samples. – Atomnaya energiya, vol. 65, issue 6, 1988 (in Russian).

[8] G. Bell, S. Glasstone. Nuclear reactor theory. – VNR Company, 1970.

[9] V. Doulin. Reactor criticality perturbation and group constants correction. – Moscow, Atomizdat, 1979.

[10] K. Dietze. Analysis of the Rossendorf SEG experiments using the JNC route for reactor calculation. – JNC TN9400 99-089, November, 1999.

[11] S. Bednyakov et al. Testing of some reactor functionals in experiments on fast critical assemblies. . – Atomnaya energiya, vol. 69, issue 1, 1990 (in Russian).

ACKNOWLEDGEMENTS.

Acknowledgements

The author is grateful to Kazuteru Sugino (research staff of Reactor Physics Group), Wakaei Sato and Takehiko Iwai (Nuclear Energy System Incorporation) for preparing of some necessary input data.

ATTACHMENTS.

Attachment 1. Plate-stretch cell model for CASUP input.

```
bfs_581i/zle_581i/pu581i__/TubeSS_2/no bars/2 wrappers
 1 1 1 0 1 0 0 0 1 0 0 1 0 1 10
21 21 1 21 1 70 0
 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20
21
 2.5879  0.7379  0.6656  0.7557  0.6656  0.1732  0.7379  0.6656
 0.7379  0.6656  2.5879  0.7379  0.6656  0.7379  0.6656  0.7379
 0.1732  0.6656  0.7557  0.6656  0.7379
0.0000E+0
300.00000
 7 4 1 7 1 5 4 1 4 1 7 4 1 4 1 4 5 1 7 1
 4
Wrapper     7
  6 1.49783E-04  13 5.66527E-04  22 2.32838E-04  24 6.22574E-03
  25 4.90961E-04  26 2.23581E-02  28 2.91098E-03
Al2O3___    4
  8 6.96316E-02  13 4.64039E-02  22 4.95131E-05  26 4.24742E-05
Na_pure_    1
  11 2.44076E-02
SS__9.93    7
  6 3.89583E-04  13 1.47404E-03  22 6.05875E-04  24 1.61980E-02
  25 1.27754E-03  26 5.81708E-02  28 7.57384E-03
Na_pure_    1
  11 2.44076E-02
Pu_95%__    5
  31 1.94628E-03 949 3.21890E-02 940 1.55907E-03 941 2.29537E-05
 951 6.23800E-05
Al2O3___    4
  8 6.96316E-02  13 4.64039E-02  22 4.95131E-05  26 4.24742E-05
Na_pure_    1
  11 2.44076E-02
Al2O3___    4
  8 6.96316E-02  13 4.64039E-02  22 4.95131E-05  26 4.24742E-05
Na_pure_    1
  11 2.44076E-02
Wrapper     7
  6 1.49783E-04  13 5.66527E-04  22 2.32838E-04  24 6.22574E-03
  25 4.90961E-04  26 2.23581E-02  28 2.91098E-03
Al2O3___    4
  8 6.96316E-02  13 4.64039E-02  22 4.95131E-05  26 4.24742E-05
Na_pure_    1
  11 2.44076E-02
Al2O3___    4
  8 6.96316E-02  13 4.64039E-02  22 4.95131E-05  26 4.24742E-05
Na_pure_    1
  11 2.44076E-02
Al2O3___    4
  8 6.96316E-02  13 4.64039E-02  22 4.95131E-05  26 4.24742E-05
Pu_95%__    5
  31 1.94628E-03 949 3.21890E-02 940 1.55907E-03 941 2.29537E-05
 951 6.23800E-05
Na_pure_    1
  11 2.44076E-02
SS__9.93    7
  6 3.89583E-04  13 1.47404E-03  22 6.05875E-04  24 1.61980E-02
  25 1.27754E-03  26 5.81708E-02  28 7.57384E-03
Na_pure_    1
  11 2.44076E-02
Al2O3___    4
  8 6.96316E-02  13 4.64039E-02  22 4.95131E-05  26 4.24742E-05
```

```
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 1 1 1 0 1 0 0 0
 1
 1
1.0000E-041.0000E-05      10     600
C10E01
```

Attachment 2. Code text for CRW ratios assessment.

```
Private Sub btnEnd_Click()
End
End Sub

Private Sub btnPrint_Click()
Results = "Results"
  Path = "D:\Projects\HeBiCor\"
  If Option1.Value = True Then s = "(CASUP)"
  If Option1.Value = False Then s = "(CASUP/CITATION)"

  Open Path + Results For Output As 1

  Print #1, "HOMOGENEOUS CALCULATION " + s
  Print #1,
  For l = 0 To lstResHom.ListCount - 1
  Print #1, lstResHom.List(l)
  Next l

  If Option1.Value = True Then
  Print #1,
  Print #1, "HETEROGENEOUS CALCULATION " + s
  Print #1,
  For l = 0 To lstResHet.ListCount - 1
  Print #1, lstResHet.List(l)
  Next l

  Print #1,
  Print #1, "CALCULATION RESULTS " + s
  Print #1,
  For l = 0 To lstResCor.ListCount - 1
  Print #1, lstResCor.List(l)
  Next l
  End If

  Close #1

End Sub

Private Sub btnStart_Click()
  Dim i, j, k As Long

  elm = "951"

  'READING FILE FROM LIBRARY FOR NON-SELFSHIELDED DATA
  Dim Blanks() As Integer

  Path = "D:\Projects\HeBiCor\"
  DataFromLibrary = "JFS3J3_EXCEPT_FTAB.N33.sp"
  Face.MousePointer = 11


  'Reading of a first string
  Open Path + DataFromLibrary For Input As 1
  Line Input #1, stroka
  stroka = Trim(stroka): len_s = Len(stroka)

  blank_count = 0

    For i = 1 To len_s
```

```
    If InStr(Right(Left(stroka, i), 1), " ") > 0 Then
    List2.AddItem (i)

    blank_count = blank_count + 1
    ReDim Preserve Blanks(blank_count)
    Blanks(blank_count) = i
    End If
    Next i

  Dim FirstLineData() As String
  ReDim FirstLineData(blank_count + 1)

    For i = 1 To blank_count
    bla = 0
    If i > 1 Then bla = Blanks(i - 1)
    FirstLineData(i) = Right(Left(stroka, Blanks(i) - 1), Blanks(i) - bla - 1)

    List3.AddItem (FirstLineData(i))
    Next i
  FirstLineData(blank_count + 1) = Right(stroka, len_s - Blanks(blank_count))
  FirLiDa = blank_count + 1
  List3.AddItem (FirstLineData(i))

  'Reading the remaining  strings
  Dim LineData() As String
  Dim FullArray() As Single   'All data array from the file (excluding 1 line)
  Dim CurLine As Integer
  Dim arr_count As Long       'All data counter from the file (excluding 1 line)

  arr_count = 0
  Do Until EOF(1)
  Line Input #1, stroka
  stroka = Trim(stroka): len_s = Len(stroka): blank_count = 0

    For i = 1 To len_s
    If InStr(Right(Left(stroka, i), 1), " ") > 0 Then
    blank_count = blank_count + 1
    ReDim Preserve Blanks(blank_count)
    Blanks(blank_count) = i
    End If
    Next i

  CurLine = blank_count + 1
  ReDim LineData(CurLine)

    For i = 1 To blank_count
    bla = 0
    If i > 1 Then bla = Blanks(i - 1)
    LineData(i) = Right(Left(stroka, Blanks(i) - 1), Blanks(i) - bla - 1)
    arr_count = arr_count + 1
    ReDim Preserve FullArray(arr_count)
    FullArray(arr_count) = Val(LineData(i))
    Next i

  LineData(blank_count + 1) = Right(stroka, len_s - Blanks(blank_count))
  arr_count = arr_count + 1
  ReDim Preserve FullArray(arr_count)
  FullArray(arr_count) = Val(LineData(i))

  Loop
  Close 1
```

```
' THE DATA HAVE BEEN READ INTO "FullArray" ARRAY

    lnmax = Val(FirstLineData(1)): up_1 = lnmax
     imax = Val(FirstLineData(2)): up_2 = up_1 + imax + 1
    mxchi = Val(FirstLineData(3)): up_3 = up_2 + imax * mxchi
    mxr1d = Val(FirstLineData(4)): up_4 = up_3 + mxchi
   mtxr23 = Val(FirstLineData(5)): up_5 = up_4 + lnmax
   mxdnse = Val(FirstLineData(6)): up_6 = up_5 + 3 * lnmax
   mxdnsi = Val(FirstLineData(7))
   mxdns2 = Val(FirstLineData(8))
   mxdwne = Val(FirstLineData(9))
   mxdwni = Val(FirstLineData(10))
   mxdwn2 = Val(FirstLineData(11))
   mxreac = Val(FirstLineData(12))
   mxsigo = Val(FirstLineData(13))
                up_7 = up_6 + 3 * lnmax + mxreac * lnmax + mxsigo * lnmax
   'mxtemp = Val(FirstLineData(14)
   '   mxr = Val(FirstLineData(15)
    iswh = Val(FirstLineData(16))

'Auxiliary arrays
Dim EnBoun() As Single: ReDim Preserve EnBoun(imax + 1)  'Energy group bounds
Dim FisMat() As String: ReDim Preserve FisMat(mxchi)     'Fissile elements (isotopes)
Dim Ld() As Integer: ReDim Preserve Ld(3, lnmax)  'Max. sink group number (el,in,n2n)
Dim La() As Integer
ReDim Preserve La(3, lnmax)     'Lowest en. group that cr.-sec. are given (el,in,n2n)


'Reading data for some arrays borders
gn = imax                       'Energy groups number
en = lnmax                      'Elements (isotopes) number

'Reservation of a memory for non-diluted parameters
Dim ScInf() As Single: ReDim Preserve ScInf(en, gn)  'Non-diluted capture (micro)
Dim SfInf() As Single: ReDim Preserve SfInf(en, gn)  'Non-diluted fission (micro)
Dim SeInf() As Single: ReDim Preserve SeInf(en, gn)  'Non-diluted elastic (micro)
Dim SmInf() As Single: ReDim Preserve SmInf(en, gn)  'Non-diluted elastic moderation (micro)
Dim SiInf() As Single: ReDim Preserve SiInf(en, gn)  'Non-diluted inelastic (micro)
Dim SnInf() As Single: ReDim Preserve SnInf(en, gn)  'Non-diluted n,2n (micro)
Dim Mu() As Single: ReDim Preserve Mu(en, gn)        'Cosinus of scattering angle
Dim Nu() As Single: ReDim Preserve Nu(en, gn)        'Neutron yield/fission
Dim Xi() As Single: ReDim Preserve Xi(en, gn)        'Fission spectrum
Dim ElN() As String: ReDim Preserve ElN(en)          'Elements (isotopes) names in the cell
Dim SeMtx() As Single: ReDim Preserve SeMtx(en, gn, gn) 'Elastic scattering matrix (micro)
Dim SiMtx() As Single: ReDim Preserve SiMtx(en, gn, gn) 'Inelastic scattering matrix (micro)
Dim SnMtx() As Single: ReDim Preserve SnMtx(en, gn, gn) 'N,2n reaction matrix (micro)

'Reading data for non-diluted parameters
For i = 1 To up_1
ElN(i) = Trim(Str(FullArray(i)))          'Elements (isotopes) names
List4.AddItem (ElN(i))
Next i

ii = 0
For i = up_1 + 1 To up_2
ii = ii + 1: EnBoun(ii) = FullArray(i)     'Energy group boundaries
EnerBo.AddItem (Str(EnBoun(ii)))
Next i

ii = 0
For i = up_3 + 1 To up_4
ii = ii + 1: FisMat(ii) = Trim(Str(FullArray(i)))  'Fissile elements (isotopes)
```

```
FisEl.AddItem (FisMat(ii))
Next i

For k = 1 To lnmax
  For j = 1 To imax
  Xi(k, j) = 0
  Next j
Next k

ii = 0
For l = 1 To mxchi
  For k = 1 To lnmax
  If ElN(k) = FisMat(l) Then
  ii = ii + 1
    For i = up_2 + 1 + imax * (ii - 1) To up_2 + imax * ii
    Xi(k, i - up_2 - imax * (ii - 1)) = FullArray(i)          'Fission spectra
    If ii = 1 Then List7.AddItem (Str(Xi(k, i - up_2 - imax * (ii - 1))))
    If ii = 2 Then List8.AddItem (Str(Xi(k, i - up_2 - imax * (ii - 1))))
    If ii = 3 Then List9.AddItem (Str(Xi(k, i - up_2 - imax * (ii - 1))))
    Next i
  End If
Next k
Next l

For k = 1 To lnmax
  For l = 1 To 3
  Ld(l, k) = FullArray(up_5 + l + 3 * (k - 1)) 'Max. sink group number (el,in,n2n)
  Next l
  List10.AddItem (Str(Ld(1, k)) + Str(Ld(2, k)) + Str(Ld(3, k)))
Next k

For k = 1 To lnmax
  For l = 1 To 3
  La(l, k) = FullArray(up_6 + l + 3 * (k - 1)) 'Lowest en. group that cr.-sec. are given (el,in,n2n)
  Next l
  List11.AddItem (Str(La(1, k)) + Str(La(2, k)) + Str(La(3, k)))
Next k

k = 0
For i = 1 To imax
  For j = 1 To mxrld
    For m = 1 To lnmax
    If j = 1 Then k = k + 1: SfInf(m, i) = FullArray(up_7 + k)
    If j = 2 Then k = k + 1: Nu(m, i) = FullArray(up_7 + k)
    If j = 3 Then k = k + 1: ScInf(m, i) = FullArray(up_7 + k)
    If j = 4 Then k = k + 1: SiInf(m, i) = FullArray(up_7 + k)
    If j = 5 Then k = k + 1: SeInf(m, i) = FullArray(up_7 + k)
    If j = 6 Then k = k + 1: Mu(m, i) = FullArray(up_7 + k)
    If j = 7 Then k = k + 1: SmInf(m, i) = FullArray(up_7 + k)
    If j = 8 Then k = k + 1: SnInf(m, i) = FullArray(up_7 + k)

    If j = 1 And ElN(m) = elm Then
    FisInf.AddItem (Str(SfInf(m, i)))
    End If

    If j = 2 And ElN(m) = elm Then
    NuList.AddItem (Str(Nu(m, i)))
    End If

    If j = 3 And ElN(m) = elm Then
    CapInf.AddItem (Str(ScInf(m, i)))
    End If
```

```
    If j = 4 And ElN(m) = elm Then
    List18.AddItem (Str(SiInf(m, i)))
    End If

    If j = 5 And ElN(m) = elm Then
    ElGroup.AddItem (Str(SeInf(m, i)))
    End If

    If j = 6 And ElN(m) = elm Then
    AvCos.AddItem (Str(Mu(m, i)))
    End If

    If j = 7 And ElN(m) = elm Then
    ElMod.AddItem (Str(SmInf(m, i)))
    End If

    If j = 8 And ElN(m) = elm Then
    List19.AddItem (Str(SnInf(m, i)))
    End If

    Next m
  Next j

If i <= mxdwne Then
  For m = 1 To lnmax
    For j = 1 To mxdnse
    k = k + 1

    l = i + j - 1
    If l <= imax Then
    SeMtx(m, i, l) = FullArray(up_7 + k) 'Elastic non-diluted matrix
      Else
      SeMtx(m, i, imax) = SeMtx(m, i, imax) + FullArray(up_7 + k) 'Elastic non-diluted matrix
    End If

    Next j
  Next m
End If

If iswh <> 0 Then
  For j = 1 To imax
    k = k + 1

    l = i + j - 1
    If l <= imax Then
    SeMtx(iswh, i, l) = FullArray(up_7 + k) 'Elastic hydrogen matrix
      Else
      SeMtx(iswh, i, imax) = SeMtx(iswh, i, imax) + FullArray(up_7 + k) 'Elastic hydrogen matrix
    End If

  Next j
End If


If i <= mxdwni Then
  For m = 1 To lnmax
    For j = 1 To mxdnsi
    k = k + 1

    l = i + j - 1
    If l <= imax Then
```

```
      SiMtx(m, i, l) = FullArray(up_7 + k) 'Inelastic matrix
        Else
        SiMtx(m, i, imax) = SiMtx(m, i, imax) + FullArray(up_7 + k) 'Inelastic matrix
      End If

      Next j
    Next m
  End If


  If i <= mxdwn2 Then
    For m = 1 To lnmax
      For j = 1 To mxdns2
      k = k + 1

      l = i + j - 1
      If l <= imax Then
      SnMtx(m, i, l) = FullArray(up_7 + k) 'N,2n matrix
        Else
        SnMtx(m, i, imax) = SnMtx(m, i, imax) + FullArray(up_7 + k) 'N,2n matrix
      End If

      Next j
    Next m
  End If
  Next i

  'Incorporation of group elastic remove cross-sections into zero value elastic matrix
  For m = 1 To lnmax
    For i = 1 To imax - 1: el = 0
      For j = i To imax
       el = el + SeMtx(m, i, j)
       If j = imax And el = 0 And SeInf(m, i) > 0 Then
       SeMtx(m, i, i) = SeInf(m, i) - SmInf(m, i)
       SeMtx(m, i, i + 1) = SmInf(m, i)
       End If
      Next j
    Next i
  Next m

       For i = 1 To imax
       el = 0: inel = 0: nn = 0
       For j = i To imax
       For m = 1 To lnmax
       If ElN(m) = elm Then
       ElMatrix.AddItem (Str(i) + Str(j) + Str(SeMtx(m, i, j)))
       InMatrix.AddItem (Str(i) + Str(j) + Str(SiMtx(m, i, j)))
       NnMatrix.AddItem (Str(i) + Str(j) + Str(SnMtx(m, i, j)))
       el = el + SeMtx(m, i, j)
       inel = inel + SiMtx(m, i, j)
       nn = nn + 0.5 * SnMtx(m, i, j)
       End If
       Next m
       Next j
       ElMatSum.AddItem (Str(el))
       InMatSum.AddItem (Str(inel))
       NnMatSum.AddItem (Str(Round(nn, 8)))
       Next I
```

```
'READING DATA FROM INPUT FILE FOR <CASUP> CELL CALCULATION

HetCASUPInp = "CasInp.csp"

'Layer parameters
Dim t() As Single              'Layers thickness in the cell model
Dim mmk() As Integer           'The number of isotopes in each layer
Dim mmkiso() As String         'Names of nuclides from mmk() array

'Reading of the HetCASUPInp file.
Open Path + HetCASUPInp For Input As 1
i = 0
For l = 1 To 3
Line Input #1, stroka: i = i + 1  'Skipping of needless strings
Next l

ln = Val(Left(stroka, 3))      'Number of layers in cell model (99-CASUP max)
ReDim Preserve t(ln), mmk(ln), mmkiso(ln, en)
aa = Int(ln / 20) + IIf(ln / 20 - Int(ln / 20) = 0, 0, 1)
              'aa - number of strings to be read as material map (needless)
              'and later as isotopes number in every layer for CASUP
For l = 1 To aa
Line Input #1, stroka: i = i + 1 'Material map for CASUP (needless)
Next l

a = Int(ln / 9) + IIf(ln / 9 - Int(ln / 9) = 0, 0, 1)
              'a - number of strings to be read for thicknesses data
            ' '9 - is CASUP format
n = 0
For l = 1 To a
Line Input #1, stroka: i = i + 1
If l < a Then

For m = 1 To 8                          '8 - is CASUP format
n = n + 1
t(n) = Val(Left(Right(stroka, (8 - m + 1) * 9), 9))     '8 & 9 - is CASUP format
Next m

  Else
  ls = Len(stroka) / 9                  '9 - is CASUP format

  For m = 1 To ln - 8 * (a - 1)             '8 - is CASUP format
  n = n + 1
  t(n) = Val(Left(Right(stroka, (ls - m + 1) * 9), 9)) '9 - is CASUP format
  Next m

End If
Next l

For l = 1 To 2
Line Input #1, stroka: i = i + 1             'Skipping of needless strings
Next l

n = 0
For l = 1 To aa
Line Input #1, stroka: i = i + 1
If l < aa Then

For m = 1 To 20
n = n + 1
mmk(n) = Val(Left(Right(stroka, (20 - m + 1) * 3), 3)) '20 & 3 - is CASUP format
Next m
```

```
  Else
  ls = Len(stroka) / 3                    '3 - is CASUP format

  For m = 1 To ln - 20 * (aa - 1)                '20 - is CASUP format
  n = n + 1
  mmk(n) = Val(Left(Right(stroka, (ls - m + 1) * 3), 3)) '3 - is CASUP format
  Next m

 End If
 Next l

 For l = 1 To ln
 Line Input #1, stroka: i = i + 1   'Name of layer and number of isotopes in it
                        '(needless)

 a = Int(mmk(l) / 4) + IIf(mmk(l) / 4 - Int(mmk(l) / 4) = 0, 0, 1)
                        'a - number of strings to be read nuclides data
                        '4 & 12 - is CASUP format

   n = 0
   For mm = 1 To a
   Line Input #1, stroka: i = i + 1
   If mm < a Then

    For m = 1 To 4                  '4 - is CASUP format
    n = n + 1
    mmkiso(l, n) = Trim(Left(Right(stroka, (4 - m + 1) * (4 + 12)), 4))
    Next m                          '4 & 12 - is CASUP format

      Else
      ls = Len(stroka) / (4 + 12)       '4 & 12 - is CASUP format

      For m = 1 To ls
      n = n + 1
      mmkiso(l, n) = Trim(Left(Right(stroka, (ls - m + 1) * (4 + 12)), 4))
      Next m                          '4 & 12 - is CASUP format

   End If
   Next mm

 Next l

 Close 1

Dim tmpiso() As Integer: ReDim Preserve tmpiso(en)
For l = 1 To en: tmpiso(l) = 0: Next l
For l = 1 To en
  For m = 1 To ln
    For n = 1 To en
    If mmkiso(m, n) = ElN(l) Then tmpiso(l) = 1
    Next n
  Next m
Next l
For l = 1 To en: iq = iq + tmpiso(l): Next l  'iq - the number of nuclides in the cell
Dim mmkhoiso() As String: ReDim Preserve mmkhoiso(iq)  'List of nuclides in hom. cell
nn = 0
For l = 1 To en
If tmpiso(l) = 1 Then nn = nn + 1: mmkhoiso(nn) = ElN(l)
Next l
```

```
'READING DATA FROM OUTPUT FILES OF <CASUP> CELL CALCULATION
                DilCASUPf61 = "Befft61"
                HetCASUPf62 = "Befft62"
  HomCASUPf77 = "Bofft77": HetCASUPf77 = "Befft77"
  HomCASUPa77 = "Boaft77": HetCASUPa77 = "Beaft77"
  HomCASUPf60 = "Bofft60": HetCASUPf60 = "Befft60"

ids = 29 'CASUP format for down-scattering groups

Dim SdSmp As Single                    'Dilution of a sample

'Layer calculated parameters
Dim Khet As Single                     'Heterogeneous K-effective
Dim Fa() As Single: ReDim Preserve Fa(ln, gn)       'Adjoint flux in layers
Dim Fd() As Single: ReDim Preserve Fd(ln, gn)       'Direct flux in layers
Dim Sc() As Single: ReDim Preserve Sc(ln, en, gn)   'Capture in layers (micro)
Dim Sf() As Single: ReDim Preserve Sf(ln, en, gn)   'Fission in layers (micro)
Dim St() As Single: ReDim Preserve St(ln, en, gn)   'Total in layers (micro)
Dim StrMac() As Single: ReDim Preserve StrMac(ln, gn) 'Transport in layers (macro)
Dim NuSfMac() As Single: ReDim Preserve NuSfMac(ln, gn)  'Nu*Fission in layers (macro)
Dim So() As Single: ReDim Preserve So(ln, en, gn)       'Dilution in layers (micro)
Dim SmLay() As Single: ReDim Preserve SmLay(en, gn, gn)  'Total moderation in layer (micro)

'Homogeneous parameters
Dim Khom As Single                     'Homogeneous K-effective
Dim FaHom() As Single: ReDim Preserve FaHom(gn)     'Homogeneous adjoint flux
Dim FdHom() As Single: ReDim Preserve FdHom(gn)     'Homogeneous direct flux
Dim SfHom() As Single: ReDim Preserve SfHom(en, gn) 'Homogeneous fission (micro)
Dim ScHom() As Single: ReDim Preserve ScHom(en, gn) 'Homogeneous capture (micro)
Dim SmHom() As Single: ReDim Preserve SmHom(en, gn, gn) 'Homogeneous total moderation (micro)
Dim SoHom() As Single: ReDim Preserve SoHom(en, gn) 'Homogeneous dilution (micro)
Dim SeHom() As Single: ReDim Preserve SeHom(en, gn) 'Homogeneous elastic (micro)

'Filling some of homogeneous and layer arrays with non-selfshielded data.
'Some of these data will be replaced if the elements are in corresponding media.

For n = 1 To gn
   For m = 1 To en
   ScHom(m, n) = ScInf(m, n)
   SfHom(m, n) = SfInf(m, n)
   SeHom(m, n) = SeInf(m, n)
     For k = 1 To gn
     SmHom(m, n, k) = SeMtx(m, n, k) + SiMtx(m, n, k)
     Next k
   Next m
Next n

For l = 1 To ln
   For n = 1 To gn
     For m = 1 To en
     St(l, m, n) = ScInf(m, n) + SfInf(m, n) + SeInf(m, n) + SiInf(m, n)
     Sc(l, m, n) = ScInf(m, n)
     Sf(l, m, n) = SfInf(m, n)
     Next m
    Next n
Next l

   'Reading of the DilCASUPf61 file.
   Open Path + DilCASUPf61 For Input As 1
   arr_count = 0: i = 0
   Do Until EOF(1)
   Line Input #1, stroka: arr_count = arr_count + 1
```

```
a = Len(stroka) / 12                    '12 - is CASUP format
For m = 1 To a
i = i + 1: ReDim Preserve FullArray(i)
FullArray(i) = Val(Left(Right(stroka, (a - m + 1) * 12), 12))  '12 - is CASUP format
Next m
Loop
Close 1

For n = 1 To gn
  For m = 1 To en
  SoHom(m, n) = 10000000000#
  Next m

  For l = 1 To ln
    For m = 1 To en
    So(l, m, n) = 10000000000#
    Next m
  Next l
Next n

mk1 = 0
For k = 1 To ln
mk1 = mk1 + mmk(k)
Next k

For n = 1 To gn
  For m = 1 To iq
    For nn = 1 To en
    If mmkhoiso(m) = ElN(nn) Then Exit For
    Next nn
  SoHom(nn, n) = FullArray((n - 1) * iq + (n - 1) * mk1 + m)
  If ElN(nn) = elm Then DilHom.AddItem (Str(n) + "  " + Str(SoHom(nn, n)))
  Next m
mk2 = 0
  For l = 1 To ln
  mk2 = mk2 + mmk(l)
    For m = 1 To mmk(l)
      For nn = 1 To en
      If mmkiso(l, m) = ElN(nn) Then Exit For
      Next nn
      So(l, nn, n) = FullArray(n * iq + (n - 1) * mk1 + (mk2 - mmk(l)) + m)
      If ElN(nn) = elm Then DilHet.AddItem (Str(n) + "  " + Str(l) + "  " + Str(So(l, nn, n)))
    Next m
  Next l
Next n


'For l = 1 To ln
'For n = 1 To gn
'For i = 1 To en
'If l >= 6 And l <= 8 Then
'If ElN(i) = elm Then Debug.Print (Str(l) + "  " + Str(n) + "  " + Str(So(l, i, n)))
'End If
'Next i
'Next n
'Next l

  'Reading of the HetCASUPf62 file.
  Open Path + HetCASUPf62 For Input As 1
  arr_count = 0: i = 0
  Do Until EOF(1)
  Line Input #1, stroka: arr_count = arr_count + 1
```

```
a = Len(stroka) / 12                          '12 - is CASUP format
For m = 1 To a
i = i + 1: ReDim Preserve FullArray(i)
FullArray(i) = Val(Left(Right(stroka, (a - m + 1) * 12), 12))  '12 - is CASUP format
Next m
Loop
Close 1

For k = 1 To ln
  For i = 1 To gn
  StrMac(k, i) = FullArray(gn * (6 + (ids + 1)) * (k - 1) + i)
        StrMa.AddItem (Str(StrMac(k, i)) + Str(k) + Str(i))
  Next i
  For i = 1 To gn

  NuSfMac(k, i) = FullArray(gn * (6 + (ids + 1)) * (k - 1) + gn * 2 + i)
        NuSfMa.AddItem (Str(NuSfMac(k, i)) + Str(k) + Str(i))
 Next i
Next k

'Reading of the HomCASUPf77 file.
Open Path + HomCASUPf77 For Input As 1
arr_count = 0: i = 0
Do Until EOF(1)
Line Input #1, stroka: arr_count = arr_count + 1
a = Len(stroka) / 12                          '12 - is CASUP format
For m = 1 To a
i = i + 1: ReDim Preserve FullArray(i)
FullArray(i) = Val(Left(Right(stroka, (a - m + 1) * 12), 12))  '12 & 6 - is CASUP format
Next m
Loop
Close 1


epsilon = 0: fullsum = 0
For l = 1 To gn
FdHom(l) = FullArray(l)
                    FdHo.AddItem (Str(FdHom(l)) + Str(l))
If l >= 29 Then epsilon = epsilon + FdHom(l)
fullsum = fullsum + FdHom(l)
Next l
epsilon = Round(epsilon * 100 / fullsum, 1)      'Part of neutrons below 9.1188 keV
              Epsilo.AddItem (Str(epsilon) + "% n-ov below E=9.1188 keV")
Khom = FullArray(gn + 1)
                    FdHo.AddItem (Str(Khom))


'Reading of the HetCASUPf77 file.
Open Path + HetCASUPf77 For Input As 1
arr_count = 0: i = 0
Do Until EOF(1)
Line Input #1, stroka: arr_count = arr_count + 1
If EOF(1) = False Then
For m = 1 To 6                                '6 - format of CASUP format
i = i + 1: ReDim Preserve FullArray(i)
FullArray(i) = Val(Left(Right(stroka, (6 - m + 1) * 12), 12))  '12 & 6 - is CASUP format
Next m
  Else
  Khet = Val(Trim(stroka))
End If
Loop
Close 1
```

```
For l = 1 To gn
  For m = 1 To ln
  Fd(m, l) = FullArray((l - 1) * ln + m)
                    FdHet.AddItem (Str(Fd(m, l)) + Str(m) + Str(l))
  Next m
Next l
                    FdHet.AddItem (Str(Khet))

'Reading of the HomCASUPa77 file.
Open Path + HomCASUPa77 For Input As 1
arr_count = 0: i = 0
Do Until EOF(1)
Line Input #1, stroka: arr_count = arr_count + 1
a = Len(stroka) / 12                          '12 - is CASUP format
For m = 1 To a                                '6 - format of CASUP format
i = i + 1: ReDim Preserve FullArray(i)
FullArray(i) = Val(Left(Right(stroka, (a - m + 1) * 12), 12)) '12 & 6 - is CASUP format
Next m
Loop
Close 1

For l = 1 To gn
FaHom(l) = FullArray(l)
                    FaHo.AddItem (Str(FaHom(l)) + Str(l))
Next l

'Reading of the HetCASUPa77 file.
Open Path + HetCASUPa77 For Input As 1
arr_count = 0: i = 0
Do Until EOF(1)
Line Input #1, stroka: arr_count = arr_count + 1
If EOF(1) = False Then
For m = 1 To 6                                '6 - format of CASUP format
i = i + 1: ReDim Preserve FullArray(i)
FullArray(i) = Val(Left(Right(stroka, (6 - m + 1) * 12), 12)) '12 & 6 - is CASUP format
Next m
End If
Loop
Close 1

For l = 1 To gn
  For m = 1 To ln
  Fa(m, l) = FullArray((l - 1) * ln + m)
                    FaHet.AddItem (Str(Fa(m, l)) + Str(m) + Str(l))
  Next m
Next l

'Reading of the HomCASUPf60 file.
Open Path + HomCASUPf60 For Input As 1
arr_count = 0: i = 0
Do Until EOF(1)
Line Input #1, stroka: arr_count = arr_count + 1
a = Len(stroka) / 12                          '12 - is CASUP format
For m = 1 To a
i = i + 1: ReDim Preserve FullArray(i)
FullArray(i) = Val(Left(Right(stroka, (a - m + 1) * 12), 12)) '12 - is CASUP format
Next m
Loop
Close 1

For n = 1 To gn
```

```
      For m = 1 To iq
        For l = 1 To en
        If mmkhoiso(m) = ElN(l) Then Exit For
        If l = en And mmkhoiso(m) <> ElN(l) Then GoTo 5
        Next l

      SfHom(l, n) = FullArray((n - 1) * (14 + ids) * iq + (m - 1) * (14 + ids) + 3)
      ScHom(l, n) = FullArray((n - 1) * (14 + ids) * iq + (m - 1) * (14 + ids) + 5 + ids)

If mmkhoiso(m) = elm Then FisHo.AddItem (Str(n) + "  " + Str(SfHom(l, n)))
If mmkhoiso(m) = elm Then CapHo.AddItem (Str(n) + "  " + Str(ScHom(l, n)))

      SeHom(l, n) = 0
        For nn = 1 To ids
        If n + nn - 1 <= gn Then

          If nn = 1 Then
          cto = FullArray((n - 1) * (14 + ids) * iq + (m - 1) * (14 + ids) + 13 + ids)
          ctr = FullArray((n - 1) * (14 + ids) * iq + (m - 1) * (14 + ids) + 7 + ids)
          c = cto - ctr
            Else
            c = 0
          End If

        SmHom(l, n, n + nn - 1) = FullArray((n - 1) * (14 + ids) * iq + (m - 1) * (14 + _
                        ids) + 3 + nn) - SnMtx(l, n, n + nn - 1) + c

        SeHom(l, n) = SeHom(l, n) + (SmHom(l, n, n + nn - 1) - SiMtx(l, n, n + nn - 1))

          If mmkhoiso(m) = elm Then
          ScMatHom.AddItem (Str(n) + "  " + Str(n + nn - 1) + "  " + Str(SmHom(l, n, n + nn - 1)))
          End If

        End If
        Next nn

          If mmkhoiso(m) = elm Then
          ElHom.AddItem (Str(n) + "  " + Str(SeHom(l, n)))
          End If

5     Next m
    Next n

    'Reading of the HetCASUPf60 file.
    Open Path + HetCASUPf60 For Input As 1
    arr_count = 0: i = 0
    Do Until EOF(1)
    Line Input #1, stroka: arr_count = arr_count + 1
    a = Len(stroka) / 12                              '12 - is CASUP format
    For m = 1 To a
    i = i + 1: ReDim Preserve FullArray(i)
    FullArray(i) = Val(Left(Right(stroka, (a - m + 1) * 12), 12))  '12 - is CASUP format
    Next m
    Loop
    Close 1

    For n = 1 To gn
    mk2 = 0
      For k = 1 To ln
      mk2 = mk2 + mmk(k)
        For m = 1 To mmk(k)
          For l = 1 To en
```

```
        If mmkiso(k, m) = ElN(l) Then Exit For
        If l = en And mmkiso(k, m) <> ElN(l) Then GoTo 6
        Next l
        St(k, l, n) = FullArray((n - 1) * mk1 * (14 + ids) + _
                (mk2 - mmk(k)) * (14 + ids) + (m - 1) * (14 + ids) + 1)
        Sf(k, l, n) = FullArray((n - 1) * mk1 * (14 + ids) + _
                (mk2 - mmk(k)) * (14 + ids) + (m - 1) * (14 + ids) + 3)
        Sc(k, l, n) = FullArray((n - 1) * mk1 * (14 + ids) + _
              (mk2 - mmk(k)) * (14 + ids) + (m - 1) * (14 + ids) + 5 + ids)

If mmkiso(k, m) = elm And k = 6 Then TotLay.AddItem (Str(n) + "  " + Str(St(k, l, n)))
If mmkiso(k, m) = elm And k = 6 Then FisLay.AddItem (Str(n) + "  " + Str(Sf(k, l, n)))
If mmkiso(k, m) = elm And k = 6 Then CapLay.AddItem (Str(n) + "  " + Str(Sc(k, l, n)))

        If mmkiso(k, m) = elm And k = 6 Then
        For nn = 1 To ids
          If n + nn - 1 <= gn Then

            If n = n + nn - 1 Then
            cto = FullArray((n - 1) * mk1 * (14 + ids) + _
                    (mk2 - mmk(k)) * (14 + ids) + (m - 1) * (14 + ids) + 13 + ids)
            ctr = FullArray((n - 1) * mk1 * (14 + ids) + _
                    (mk2 - mmk(k)) * (14 + ids) + (m - 1) * (14 + ids) + 7 + ids)
            c = cto - ctr
              Else
              c = 0
            End If

          SmLay(l, n, n + nn - 1) = FullArray((n - 1) * mk1 * (14 + ids) + _
                  (mk2 - mmk(k)) * (14 + ids) + (m - 1) * (14 + ids) + 3 + nn) - SnMtx(l, n, n + nn - 1) + c
          ScMatHet.AddItem (Str(n) + "  " + Str(n + nn - 1) + "  " + Str(SmLay(l, n, n + nn - 1)))
          End If
        Next nn
        End If

6       Next m
      Next k
    Next n

'It is PROHIBITED TO CHANGE a content of FULLARRAY() array since this point because it is
'used for calculations as a scattering matrix as a 4D array.

If Option1.Value = False And Option2.Value = True Then
lstResHet.Clear

   'Reading of the HomReacFdFa file.
   HomReacFdFa = "FluxAdj.oin"
   Open Path + HomReacFdFa For Input As 1
   For l = 1 To gn
   Line Input #1, stroka
   FdHom(l) = Val(Left(stroka, 11))
   FaHom(l) = Val(Right(stroka, 11))
   Next l
   Close 1
End If

' Homogeneous parameters
'Dim SmMac() As Single: ReDim Preserve SmMac(gn)     'Homogeneous elastic moderation (macro)

'Layer paramaters
'Dim Sm() As Single: ReDim Preserve Sm(ln, en, gn)    'Elastic moderation in layers (micro)
```

```
'RESERVATION OF A MEMORY FOR ARRAYS. CALCULATED DATA.

Dim Rt() As Single: ReDim Preserve Rt(en)          'Homogeneous total reactivity
Dim Ra() As Single: ReDim Preserve Ra(en)          'Homogeneous absorbtion reactivity
Dim Rm() As Single: ReDim Preserve Rm(en)          'Homogeneous multiplication reactivity
Dim Rs() As Single: ReDim Preserve Rs(en)          'Homogeneous moderation reactivity
Dim Rt_slb() As Single: ReDim Preserve Rt_slb(en)  'Heterogeneous total reactivity
Dim Ra_slb() As Single: ReDim Preserve Ra_slb(en)  'Hetergeneous absorbtion reactivity
Dim Rm_slb() As Single: ReDim Preserve Rm_slb(en)  'Heterogeneous multiplication reactivity
Dim Rs_slb() As Single: ReDim Preserve Rs_slb(en)  'Heterogeneous moderation reactivity
Dim Rbsslb() As Single: ReDim Preserve Rbsslb(en)  'Heterogeneous bilinear (self) reactivity
Dim Rboslb() As Single: ReDim Preserve Rboslb(en)  'Heterogeneous bilinear (others) reactivity
Dim SpInf() As Single: ReDim Preserve SpInf(en, gn) 'Non-diluted potential (micro)

Dim kl() As Single: ReDim Preserve kl(ln)          '1) K-eff(ln); 2) 0&K for M&R
Dim SumZ() As Single: ReDim Preserve SumZ(gn)      'Auxiliary array for Rs_slb() formula
Dim FdR() As Single: ReDim Preserve FdR(gn)        'Flux in <R>-type layer
Dim FdM() As Single: ReDim Preserve FdM(gn)        'Flux in <M>-type layer
Dim FaM() As Single: ReDim Preserve FaM(gn)        'Adjoint flux in <M>-type layer
Dim FaR() As Single: ReDim Preserve FaR(gn)        'Adjoint flux in <R>-type layer
Dim SoR() As Single: ReDim Preserve SoR(en, gn)    'Dilution in <R>-layer (micro)
Dim ScR() As Single: ReDim Preserve ScR(en, gn)    'Capture in <R>-layer (micro)
Dim SfR() As Single: ReDim Preserve SfR(en, gn)    'Fission in <R>-layer (micro)
Dim StRs() As Single: ReDim Preserve StRs(en, gn)  'Total in <R>-layer (micro)
Dim StM() As Single: ReDim Preserve StM(en, gn)    'Total in <M>-layer (micro)
Dim tmparr(), tmparr1() As Single                  'Auxiliary arrays


SdSmp = 10000000000#
normalizer = "925"

'CALCULATIONS

'-------This fragment is needed only because of lack of Xi-fission in JFS3 -----
Dim NuAv() As Single: ReDim Preserve NuAv(en)
norfi = 0

For j = 1 To gn
norfi = norfi + FdHom(j)
Next j

For i = 1 To en
  For j = 1 To gn
  NuAv(i) = NuAv(i) + Nu(i, j) * FdHom(j) / norfi
  Next j
NuAver.AddItem (ElN(i) + ":   " + Trim(Str(NuAv(i))))
Next i

gnabbn78 = 26: nuabbn78 = 10: fsplim78 = 13
Dim EnBo78(): ReDim Preserve EnBo78(gnabbn78 + 1)  'Energy group borders for ABBN78
Dim NuAvSet(): ReDim Preserve NuAvSet(nuabbn78)    'Averaged neutron yields for a table of fission spectra
Dim FSpAbbn(): ReDim Preserve FSpAbbn(nuabbn78 * fsplim78) 'ABBN78 table for fission spectra
Dim EnBoMix(): ReDim Preserve EnBoMix(gn + gnabbn78 + 2)  'Enery borders of JFS and ABBN78
Dim XiMix(): ReDim Preserve XiMix(gn + gnabbn78) 'Fission spectra for mixed array of borders

'Reading of ABBN-78 data for fission spectra
FisABBN78 = "Abbn78"
Open Path + FisABBN78 For Input As 1
For l = 1 To gnabbn78 + 1
Line Input #1, stroka
EnBo78(l) = Val(stroka)
Next l
```

```
For l = 1 To nuabbn78
Line Input #1, stroka
NuAvSet(l) = Val(stroka)
Next l

For l = 1 To nuabbn78 * fsplim78
Line Input #1, stroka
FSpAbbn(l) = Val(stroka)
Next l
Close 1

'Transforming ABBN78 26-group set to JFS 70-group set for fission spectra
ReDim tmparr1(gn + gnabbn78 + 2)
For l = 1 To gn + gnabbn78 + 2
If l <= gn + 1 Then tmparr1(l) = EnBoun(l)
If l > gn + 1 Then tmparr1(l) = EnBo78(l - (gn + 1))
Next l

  ReDim tmparr(gn + gnabbn78 + 2)   'Sorting descending: start
  For l = 1 To gn + gnabbn78 + 2
  nn = 1
    For m = 1 To gn + gnabbn78 + 2
    If Val(tmparr1(l)) < Val(tmparr1(m)) Then nn = nn + 1
    Next m
  tmparr(l) = nn
  Next l
  For l = 1 To gn + gnabbn78 + 2
  EnBoMix(tmparr(l)) = tmparr1(l)
  Next l                    'Sorting descending: finish

For n = 1 To en

  For nnn = 1 To mxchi
  If ElN(n) = FisMat(nnn) Then GoTo 7
  Next nnn

 mindif = NuAvSet(nuabbn78)

  For l = 1 To gn + gnabbn78
  XiMix(l) = 0
  Next l

  If NuAv(n) > 0 Then
  For l = 1 To nuabbn78
  If Abs(NuAvSet(l) - NuAv(n)) <= mindif Then mindif = Abs(NuAvSet(l) - NuAv(n)): nusel = l
  Next l

  For l = 1 To gn + gnabbn78
    For ll = 1 To fsplim78
    If EnBoMix(l) = EnBo78(ll) Then XiMix(l) = FSpAbbn((nusel - 1) * fsplim78 + ll)
    Next ll

  If EnBoMix(l) = EnBo78(fsplim78 + 1) Then XiMix(l) = 0.000000000001

  Next l
  End If

  b = 0: b3 = 0
  For l = 1 To gn + gnabbn78

  If XiMix(l) > 0 Then
```

```
b = b + 1: b3 = b3 + 1
If b3 < fsplim78 + 1 And b = 1 Then b1 = l
End If

  If b = 2 Then
  b2 = l: b = 0: l = l - 1: b3 = b3 - 1
  ep = XiMix(b1) / (EnBoMix(b1) - EnBoMix(b2))
    For m = b1 To b2 - 1
    XiMix(m) = ep * (EnBoMix(m) - EnBoMix(m + 1))
    Next m
  End If

If b3 = fsplim78 + 1 Then
XiMix(l - 1) = XiMix(l - 1) + XiMix(l)
XiMix(l) = 0
Exit For
End If

Next l

b = 1
Xi(n, 1) = XiMix(1) + XiMix(2)

For l = 2 To gn
  For m = 3 To gn + gnabbn78
  If EnBoun(l) = EnBoMix(m) Then
  Xi(n, l) = XiMix(m)
  If b = fsplim78 Then Exit For
  If EnBoun(l + 1) = EnBoMix(m + 2) Then Xi(n, l) = Xi(n, l) + XiMix(m + 1): b = b + 1
  Exit For
  End If
  Next m
If b = fsplim78 + 1 Then Exit For
Next l

7 Next n

  'Cell height calculation
  tcell = 0
  For l = 1 To ln
  tcell = tcell + t(l)
  Next l

  'Potential cross-section calculation
  For i = 1 To en

    For j = 1 To gn
    If SeHom(i, j) / SeInf(i, j) < 0.999 Then b1 = j - 1: Exit For
    Next j

    For j = gn To 1 Step -1
    If SeHom(i, j) / SeInf(i, j) < 0.999 Then b2 = j + 1: Exit For
    Next j

  If b1 < 1 Then b1 = 1
  If b2 > gn Then b2 = gn

  dh = (SeInf(i, b2) - SeInf(i, b1)) / (b2 - b1 - 1)
  k = 0

    For j = 1 To gn
    If j <= b1 And j >= b2 Then
```

```
      SpInf(i, j) = SeInf(i, j)
        Else
        k = k + 1
        SpInf(i, j) = SeInf(i, b1) + dh * (k - 0.5)
        If SpInf(i, j) > SeInf(i, j) Then SpInf(i, j) = SeInf(i, j)
      End If
      Next j

  'If ElN(i) = elm Then
  'For j = 1 To gn
  'Tmp.AddItem (Str(j) + " " + Str(SpInf(i, j)) + "  " + Str(SeInf(i, j)))
  'Next j
  'End If

    Next i

     'Homogeneous calculations of reactivity numerators
    lstResHom.Clear
    For i = 1 To en
    Rt(i) = 0: Ra(i) = 0: Rm(i) = 0: Rs(i) = 0: xfc = 0

      For j = 1 To gn
      xfc = xfc + FaHom(j) * Xi(i, j)
      Next j

      For j = 1 To gn
      Ra(i) = Ra(i) - FdHom(j) * FaHom(j) * (ScHom(i, j) + SfHom(i, j))
        For k = j + 1 To gn
        Rs(i) = Rs(i) + FdHom(j) * (FaHom(k) - FaHom(j)) * SmHom(i, j, k)
        Next k
      Rm(i) = Rm(i) + Nu(i, j) * SfHom(i, j) * FdHom(j) * xfc / Khom
      Next j

    Rt(i) = Ra(i) + Rm(i) + Rs(i)
    Next i


    For i = 1 To en          'Search of a normalizer in the array of elements
    If ElN(i) = normalizer Then denom = Rt(i): Exit For
    Next i

    For i = 1 To en          'Normalization
    Ra(i) = Ra(i) / denom: tmparr(1) = Ra(i)
    Rm(i) = Rm(i) / denom: tmparr(2) = Rm(i)
    Rs(i) = Rs(i) / denom: tmparr(3) = Rs(i)
    Rt(i) = Rt(i) / denom: tmparr(4) = Rt(i)


    ro = Space(3 - Len(ElN(i))) + ElN(i) + ":"


      For k = 1 To 4              'Transformation of the reactivities to "###.#####" format
      If Abs(tmparr(k)) < 0.000005 Then
      tmpa = "  0.00000"
        ElseIf tmparr(k) = 1 Then
        tmpa = "  1.00000"

        Else
        sl = Len(Trim(Str(Round(tmparr(k), 5))))                  'String length
        dpp = InStr(Trim(Str(Round(tmparr(k), 5))), ".")      'Decimal point position
        rpl = sl - dpp                    'Number of figures after decimal point
```

```
    nz = ""
    For ll = 1 To 5 - rpl
    nz = nz + "0"                    'Number of zeros added from the right
    Next ll

    If dpp = 1 Then tmpa = "  0." + Right(Trim(Str(Round(tmparr(k), 5))) + nz, 5)
    If dpp = 2 And tmparr(k) < 0 Then tmpa = " -0." + _
                          Right(Trim(Str(Round(tmparr(k), 5))) + nz, 5)
    If dpp = 2 And tmparr(k) > 0 Then tmpa = "  " + _
                          Trim(Str(Round(tmparr(k), 5))) + nz
    If dpp = 3 And tmparr(k) < 0 Then tmpa = " " + _
                          Trim(Str(Round(tmparr(k), 5))) + nz
    If dpp = 3 And tmparr(k) > 0 Then tmpa = " " + _
                          Trim(Str(Round(tmparr(k), 5))) + nz
  End If
  ro = ro + tmpa
  Next k

 lstResHom.AddItem (ro)
 Next i

If Option1.Value = False Then GoTo 1

 'Heterogeneous calculations of reactivity numerators
 lstResHet.Clear

 For i = 1 To en
 Rt_slb(i) = 0: Ra_slb(i) = 0: Rm_slb(i) = 0: Rs_slb(i) = 0

 mk2 = 0
  For lay = 1 To ln
  mk2 = mk2 + mmk(lay)
    For ll = 1 To mmk(lay)
    If mmkiso(lay, ll) = ElN(i) Then Exit For
    If ll = mmk(lay) And mmkiso(lay, ll) <> ElN(i) Then
      For n = 1 To gn
        For nn = 1 To gn
        SmLay(i, n, nn) = SeMtx(i, n, nn) + SiMtx(i, n, nn)
        Next nn
      Next n
    GoTo 4
    End If
    Next ll

    For n = 1 To gn
      For nn = 1 To gn
      SmLay(i, n, nn) = 0
      Next nn
    Next n

    For n = 1 To gn
      For nn = 1 To ids
        If n + nn - 1 <= gn Then

          If n = n + nn - 1 Then
          cto = FullArray((n - 1) * mk1 * (14 + ids) + _
              (mk2 - mmk(lay)) * (14 + ids) + (ll - 1) * (14 + ids) + 13 + ids)
          ctr = FullArray((n - 1) * mk1 * (14 + ids) + _
              (mk2 - mmk(lay)) * (14 + ids) + (ll - 1) * (14 + ids) + 7 + ids)
          c = cto - ctr
            Else
            c = 0
```

```
        End If

      SmLay(i, n, n + nn - 1) = FullArray((n - 1) * mk1 * (14 + ids) + _
            (mk2 - mmk(lay)) * (14 + ids) + (ll - 1) * (14 + ids) + 3 + nn) - SnMtx(i, n, n + nn - 1) + c
    If ElN(i) = elm And lay = 6 Then
    Tmp.AddItem (Str(n) + " " + Str(n + nn - 1) + " " + Str(SmLay(i, n, n + nn - 1)))
    End If

      End If

    Next nn

  Next n

4   xfc = 0
    For j = 1 To gn
    xfc = xfc + Fa(lay, j) * Xi(i, j)
    Next j

    For j = 1 To gn
    Ra_slb(i) = Ra_slb(i) - Fd(lay, j) * Fa(lay, j) * _
                (Sc(lay, i, j) + Sf(lay, i, j)) * (t(lay) / tcell)

      For k = j + 1 To gn
      Rs_slb(i) = Rs_slb(i) + Fd(lay, j) * (Fa(lay, k) - Fa(lay, j)) * SmLay(i, j, k) * _
                                          (t(lay) / tcell)
      Next k

    Rm_slb(i) = Rm_slb(i) + Nu(i, j) * Sf(lay, i, j) * Fd(lay, j) * xfc * (t(lay) / tcell) _
                / Khet
    Next j

  Next lay

Next i

'Search for layers of R-type and M-type
For l = 1 To ln: kl(l) = 0: norfl = 0
    For j = 1 To gn
    kl(l) = kl(l) + NuSfMac(l, j) * Fd(l, j)
    norfl = norfl + Fd(l, j)
    Next j
kl(l) = kl(l) / norfl  'Layer K-eff
Next l

kmax = 0
For l = 1 To ln
If kl(l) > kmax Then kmax = kl(l): lmax = l 'Maximal layer K-eff and layer number
Next l

For l = 1 To ln
If kl(l) > 0 And (kmax - kl(l)) >= 0.05 Then kl(l) = 0  '<M> layers with K-eff>0
            '***** check it !!! (about 0.05), may be to correct? ******* check it !!! **
Next l

'Data preparation for binary cell

For j = 1 To gn
FdR(j) = 0: FdM(j) = 0: FaM(j) = 0: FaR(j) = 0
For i = 1 To en
StRs(i, j) = 0: StM(i, j) = 0: SfR(i, j) = 0: ScR(i, j) = 0: SoR(i, j) = 0
Next i
```

```
Next j

nrl = 0: nml = 0: trp = 0: tnp = 0
For l = 1 To ln
If kl(l) > 0 Then
nrl = nrl + 1                    'The number of <R>-type layers
trp = trp + t(l)                 'Thickness of the resonance part of the cell

For j = 1 To gn
FaR(j) = FaR(j) + Fa(l, j) * t(l)
FdR(j) = FdR(j) + Fd(l, j) * t(l)
Next j

  Else
  nml = nml + 1                  'The number of <M>-type layers
  tnp = tnp + t(l)               'Thickness of the non-resonance part of the cell

  For j = 1 To gn
  FdM(j) = FdM(j) + Fd(l, j) * t(l)
  FaM(j) = FaM(j) + Fa(l, j) * t(l)
  Next j

End If
Next l

For l = 1 To ln
  For i = 1 To en
    For ii = 1 To iq
    If mmkhoiso(ii) = ElN(i) Then Exit For
    If ii = iq And mmkhoiso(ii) <> ElN(i) Then GoTo 13
    Next ii

    'For nn = 1 To mmk(l)
    'If mmkiso(l, nn) = ElN(i) And kl(l) > 0 Then Exit For
    'If mmkiso(l, nn) = ElN(i) And kl(l) = 0 Then GoTo 10
    'If nn = mmk(l) And mmkiso(l, nn) <> ElN(i) Then GoTo 10
    'Next nn

    For j = 1 To gn
    If kl(l) = 0 Then
    StM(i, j) = StM(i, j) + _
            (ScHom(i, j) + SfHom(i, j) + SeHom(i, j) + SiInf(i, j)) * t(l)
    End If

    If kl(l) > 0 Then
    StRs(i, j) = StRs(i, j) + St(l, i, j) * t(l)
    SfR(i, j) = SfR(i, j) + Sf(l, i, j) * t(l)
    ScR(i, j) = ScR(i, j) + Sc(l, i, j) * t(l)
    SoR(i, j) = SoR(i, j) + So(l, i, j) * t(l)
    End If

    Next j: GoTo 8

'10   For j = 1 To gn
'    StRs(i, j) = StRs(i, j) + _
'            (ScInf(i, j) + SfInf(i, j) + SeInf(i, j) + SiInf(i, j)) * t(l)
'    StM(i, j) = StM(i, j) + _
'            (ScHom(i, j) + SfHom(i, j) + SeHom(i, j) + SiInf(i, j)) * t(l)
'    SfR(i, j) = SfR(i, j) + SfInf(i, j) * t(l)
'    ScR(i, j) = ScR(i, j) + ScInf(i, j) * t(l)
'    SoR(i, j) = SoR(i, j) + 10000000000# * t(l)
'    Next j: GoTo 8
```

```
13    For j = 1 To gn
      StRs(i, j) = ScInf(i, j) + SfInf(i, j) + SeInf(i, j) + SiInf(i, j)
      StM(i, j) = ScInf(i, j) + SfInf(i, j) + SeInf(i, j) + SiInf(i, j)
      SfR(i, j) = SfInf(i, j)
      ScR(i, j) = ScInf(i, j)
      SoR(i, j) = 10000000000#
      Next j

8     Next i
    Next l

    For j = 1 To gn       'Calculation of averaged parameters for <R>- and <M>-type layers
    FaR(j) = FaR(j) / trp
    FdR(j) = FdR(j) / trp
    FaM(j) = FaM(j) / tnp
    FdM(j) = FdM(j) / tnp

      For i = 1 To en
        For nn = 1 To iq
        If mmkhoiso(nn) = ElN(i) Then GoTo 12
        If nn = iq And mmkhoiso(nn) <> ElN(i) Then GoTo 11
        Next nn
12    StRs(i, j) = StRs(i, j) / trp
      SfR(i, j) = SfR(i, j) / trp
      ScR(i, j) = ScR(i, j) / trp
      SoR(i, j) = SoR(i, j) / trp
      StM(i, j) = StM(i, j) / tnp
11    Next i

    Next j

    'First term of bilinear correction (additional self-shielding)
    For j = 1 To gn: SumZ(j) = 0

      For l = 1 To ln
      If kl(l) > 0 Then      'Z calculation for <R>-type layer
      Z = 1
        Else                 'Z calculation for <M>-type layer
        xs = 0

        For n = 1 To ln
        If kl(n) = 0 Then
        xs = xs + t(n) * StrMac(n, j)
        End If
        Next n

        Z = 1 / (1 + 2 * xs) ^ 2
      End If
      SumZ(j) = SumZ(j) + Fd(l, j) * Fa(l, j) * t(l) * Z / tcell
      Next l

    Next j

    For i = 1 To en: Rbsslb(i) = 0
    xfc = 0

      For j = 1 To gn
      xfc = xfc + FaR(j) * Xi(i, j)
      Next j

      For j = 1 To gn
```

```
        paren1 = xfc * Nu(i, j) / FaR(j) - 1

        If SfInf(i, j) > 0 Then
        paren2 = 1 - (SfR(i, j) / SfInf(i, j)) ^ 2
          Else
          paren2 = 0
        End If

        If ScInf(i, j) > 0 Then
        paren3 = 1 - (ScR(i, j) / ScInf(i, j)) ^ 2
          Else
          paren3 = 0
        End If

        numerator = (paren1 * SfR(i, j) * paren2 - ScR(i, j) * paren3) * SumZ(j)
        denominator = 2 * (1 + SpInf(i, j) / SoR(i, j)) * (1 + StRs(i, j) / SdSmp)
        Rbsslb(i) = Rbsslb(i) - numerator / denominator
        Next j

      Next i

'For i = 1 To en
'If ElN(i) = "940" Then
'For j = 1 To gn
'Debug.Print j, StM(i, j), StRs(i, j)
'Next j
'End If
'Next i

      'Second term of bilinear correction (de-shielding of others)
      For i = 1 To en: Rboslb(i) = 0

        For j = 1 To gn

          For ii = 1 To en: If ii = i Then GoTo 2

            For l = 1 To ln: If kl(l) = 0 Then GoTo 3

            For nn = 1 To mmk(l)
            If mmkiso(l, nn) = ElN(ii) Then Exit For
            If nn = mmk(l) And mmkiso(l, nn) <> ElN(ii) Then GoTo 3
            Next nn

            xfc = 0

              For jj = 1 To gn
              xfc = xfc + Fa(l, jj) * Xi(ii, jj)
              Next jj

            paren1 = Nu(ii, j) * xfc / Fa(l, j) - 1

            If SfInf(ii, j) > 0 Then
            paren2 = 1 - (Sf(l, ii, j) / SfInf(ii, j)) ^ 2
              Else
              paren2 = 0
            End If

            If ScInf(ii, j) > 0 Then
            paren3 = 1 - (Sc(l, ii, j) / ScInf(ii, j)) ^ 2
              Else
              paren3 = 0
            End If
```

```
        paren4 = paren1 * paren2 * Sf(l, ii, j) - Sc(l, ii, j) * paren3
        paren5 = (1 - tnp / tcell) * (So(l, ii, j) / SoHom(ii, j)) ^ 2
        paren6 = Fd(l, j) * Fa(l, j) * t(l) / tcell + paren5 * FdM(j) * FaM(j)
        doublesum = doublesum + paren4 * paren6 / (SpInf(ii, j) + So(l, ii, j))
3       Next l

2     Next ii

    Rboslb(i) = Rboslb(i) + StM(i, j) * doublesum / (2 * (1 + StM(i, j) / SdSmp))
    Next j

  Next i

  For i = 1 To en
  Rt_slb(i) = Ra_slb(i) + Rm_slb(i) + Rs_slb(i) + Rbsslb(i) + Rboslb(i)
  Next i

  For i = 1 To en          'Search of a normalizer in the array of elements
  If ElN(i) = normalizer Then denom = Rt_slb(i): Exit For
  Next i

  For i = 1 To en                              'Normalization to U235 reactivity
  Ra_slb(i) = Ra_slb(i) / denom: tmparr(1) = Ra_slb(i)
  Rm_slb(i) = Rm_slb(i) / denom: tmparr(2) = Rm_slb(i)
  Rs_slb(i) = Rs_slb(i) / denom: tmparr(3) = Rs_slb(i)
  Rbsslb(i) = Rbsslb(i) / denom: tmparr(4) = Rbsslb(i)
  Rboslb(i) = Rboslb(i) / denom: tmparr(5) = Rboslb(i)
  Rt_slb(i) = Rt_slb(i) / denom: tmparr(7) = Rt_slb(i)
                    tmparr(6) = tmparr(1) + tmparr(2) + tmparr(3)
  ro = Space(3 - Len(ElN(i))) + ElN(i) + ":"

    For k = 1 To 7          'Transformation of the reactivities to "###.#####" format
    If Abs(tmparr(k)) < 0.000005 Then
    tmpa = "  0.00000"

      ElseIf tmparr(k) = 1 Then
      tmpa = "  1.00000"

      Else
      sl = Len(Trim(Str(Round(tmparr(k), 5))))              'String length
      dpp = InStr(Trim(Str(Round(tmparr(k), 5))), ".")      'Decimal point position
      rpl = sl - dpp                'Number of figures after decimal point

      nz = ""
      For ll = 1 To 5 - rpl
      nz = nz + "0"                 'Number of zeros added from the right
      Next ll

      If dpp = 1 Then tmpa = "  0." + Right(Trim(Str(Round(tmparr(k), 5))) + nz, 5)
      If dpp = 2 And tmparr(k) < 0 Then tmpa = " -0." + _
                          Right(Trim(Str(Round(tmparr(k), 5))) + nz, 5)
      If dpp = 2 And tmparr(k) > 0 Then tmpa = "  " + _
                          Trim(Str(Round(tmparr(k), 5))) + nz
      If dpp = 3 And tmparr(k) < 0 Then tmpa = " " + _
                          Trim(Str(Round(tmparr(k), 5))) + nz
      If dpp = 3 And tmparr(k) > 0 Then tmpa = " " + _
                          Trim(Str(Round(tmparr(k), 5))) + nz
    End If

    ro = ro + tmpa
    Next k
```

```
    lstResHet.AddItem (ro)

    Next i

  For i = 1 To en
  ro = Space(3 - Len(ElN(i))) + ElN(i) + ":"
  tmparr(1) = Rt(i)
  tmparr(2) = Rt_slb(i) - Rbsslb(i) - Rboslb(i)
  tmparr(3) = Rt_slb(i)
  tmparr(4) = tmparr(2) - tmparr(1)
  tmparr(5) = tmparr(3) - tmparr(1)
     For k = 1 To 5            'Transformation of the reactivities to "###.#####" format
    If Abs(tmparr(k)) < 0.000005 Then
    tmpa = "  0.00000"
      ElseIf tmparr(k) = 1 Then
      tmpa = "  1.00000"

      Else
      sl = Len(Trim(Str(Round(tmparr(k), 5))))              'String length
      dpp = InStr(Trim(Str(Round(tmparr(k), 5))), ".")      'Decimal point position
      rpl = sl - dpp                'Number of figures after decimal point

      nz = ""
      For ll = 1 To 5 - rpl
      nz = nz + "0"                 'Number of zeros added from the right
      Next ll

      If dpp = 1 Then tmpa = "  0." + Right(Trim(Str(Round(tmparr(k), 5))) + nz, 5)
      If dpp = 2 And tmparr(k) < 0 Then tmpa = " -0." + _
                          Right(Trim(Str(Round(tmparr(k), 5))) + nz, 5)
      If dpp = 2 And tmparr(k) > 0 Then tmpa = "  " + _
                          Trim(Str(Round(tmparr(k), 5))) + nz
      If dpp = 3 And tmparr(k) < 0 Then tmpa = " " + _
                          Trim(Str(Round(tmparr(k), 5))) + nz
      If dpp = 3 And tmparr(k) > 0 Then tmpa = " " + _
                          Trim(Str(Round(tmparr(k), 5))) + nz
     End If
     ro = ro + tmpa
     Next k

   lstResCor.AddItem (ro)
   Next i

1  Face.MousePointer = 1
End Sub

Private Sub Form_Load()
Option1.Value = True
Option2.Value = False
End Sub
```