M&C 2001 Salt Lake City, Utah, USA, September 2001.

LATTICE PARAMETERS GENERATION USING BI-CUBIC INTERPOLATION

José A. Castillo, Gustavo Alonso[‡]
Instituto Nacional de Investigaciones Nucleares
Km. 36.5 Autopista México-Toluca,
C.P. 52045, Salazar, Estado de México, MÉXICO
e-mail: jacm@nuclear.inin.mx, galonso@nuclear.inin.mx

Edmundo del Valle
[‡]Instituto Politécnico Nacional
Escuela Superior de Física y Matemáticas
Departamento de Ingeniería Nuclear
Unidad Profesional Adolfo López Mateos, Edificio 9, Col Lindavista,
C.P. 07738, México, D. F., MÉXICO
e-mail: edmundo@esfm.ipn.mx

Keywords: Lattice parameters, fuel assembly cross sections, lattice codes

ABSTRACT

Lattice parameters generation for reactor core analysis is a very time consuming task. In particular, design of new fuel reload requires several different lattice parameters data banks to get an adequate fuel reload. This achieves the economic and safety goals of reactor operation. In this work we propose a method to generate fuel lattice parameters by using a bi-cubic interpolation, instead of generating the lattice parameters by solving the neutron transport equation. This method reduces the time invested in generating lattice parameters for new fuel assemblies making possible to consider many more different reload scenarios. The interpolation variables per element are uranium and gadolinium enrichments, where each of these has 16 nodes. The banks generated by this method show good agreement with those generated by the RECORD computational code, which solves the transport equation. The relative differences between results are in all the cases less than 10^{-2}.

1.- INTRODUCTION

The process of getting an adequate fuel reload pattern requires of several trial and error tests. The core configuration must satisfy the constraints imposed by fuel cycle economic goals, safety margins, and operational flexibility. The fuel cycle economics is primarily a function of fuel enrichment and discharge burn-up, which are determined by cycle length (cycle energy), and the number of feed assemblies. On the other hand, safety margins and operational flexibility are highly dependent on the core power distributions, peaking factors, and reactivity feedback.

Thus, in this process we need to test several different enrichments, which will be given during the process. This leads to the necessity of creating new lattice parameters data banks for these new assemblies. The process requires the combination of fuel enrichment and burnable poison (gadolinium) content to meet the specific economic and operational goals of the fuel cycle. In Mexico there are two 640 MWe boiling water reactors which operate in 18-month cycles. It is of great interest for the utility to perform independent analysis from vendor's one. For this purpose it is used the Fuel Management System (FMS) by ScandPower (Shardhamar, 1989).

The lattice data bank generated by the RECORD code (Shardhamar, 1989), which is part of the FMS, is structured in two blocks. The first block is divided by "cases", which gives information for the second one. Each "case" is divided in two data groups (Torres, 1993) and (Hernández, 1995). The first group contains the conditions used to calculate the lattice parameters and the second group contains the isotopic concentrations inside the lattice, the macroscopic nuclear parameters and the dynamics nuclear parameters for each burnup step. The lattice parameter bank contains information for 42 different parameters. This gives an idea of the calculations required to generate a new lattice data bank. Therefore, a method to generate lattice data banks was developed, starting from a set of lattice data banks previously generated by RECORD (Castillo, 1999) and (López, 1994).

2.- METHODOLOGY

Obtaining as many polynomial as parameters exist in the data bank solves the problem of generating a lattice parameters data bank by using interpolation. The results will be closer to those yields by solving the transport equation, if the interpolation polynomial shows a closer behavior as the parameter to be interpolated.

2.1.- CANONICAL BASIS

Typically, to get the interpolation constants for the bi-cubic Lagrange interpolant one uses the canonical base and LU partial pivoting to solve the corresponding algebraic system. It is well known that for a bi-cubic Lagrange interpolant there will be 16 nodes because the dimension of the polynomial space that is going to be used is also 16. The tensor product

$$\begin{bmatrix} 1 \\ y \\ y^2 \\ y^3 \end{bmatrix} \begin{bmatrix} 1 & x & x^2 & x^3 \end{bmatrix} \tag{1}$$

leads to the matrix

$$\begin{bmatrix} 1 & x & x^2 & x^3 \\ y & xy & x^2y & x^3y \\ y^2 & xy^2 & x^2y^2 & x^3y^2 \\ y^3 & xy^3 & x^2y^3 & x^3y^3 \end{bmatrix} \tag{2}$$

A linear combination of all the elements of matrix 2 produces the following bi-cubic polynomial

$$\begin{aligned} g_h(x,y) = a_1 + a_2x + a_3x^2 + a_4x^3 + a_5y + a_6xy + a_7x^2y + a_8x^3y + a_9y^2 + a_{10}xy^2 + \ldots \\ + a_{11}x^2y^2 + a_{12}x^3y^2 + a_{13}y^3 + a_{14}xy^3 + a_{15}x^2y^3 + a_{16}x^3y^3 \end{aligned} \tag{3}$$

where constants $a_1,\ldots,a_{16}$ can be determined with the following algebraic system

$$\begin{bmatrix} 1 & x_1 & x_1^2 & \ldots & x_1y_1 & x_1^2y_1 & \ldots & x_1^3y_1^3 \\ 1 & x_2 & x_2^2 & & x_2y_1 & x_2^2y_1 & \ldots & x_2^3y_1^3 \\ & \cdot & & & \cdot & & & \cdot \\ & \cdot & & & \cdot & & & \cdot \\ 1 & x_3 & x_3^2 & \ldots & x_3y_2 & x_3^2y_2 & \ldots & x_3^3y_2^3 \\ 1 & x_4 & x_4^2 & \ldots & x_4y_2 & x_4^2y_2 & \ldots & x_4^3y_2^3 \\ & \cdot & & & \cdot & & & \cdot \\ & \cdot & & & \cdot & & & \cdot \\ 1 & x_4 & x_4^2 & \ldots & x_4y_4 & x_4^2y_4 & \ldots & x_4^3y_4^3 \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \\ \cdot \\ \cdot \\ a_7 \\ a_8 \\ \cdot \\ \cdot \\ a_{16} \end{bmatrix} = \begin{bmatrix} b_{11} \\ b_{21} \\ \cdot \\ \cdot \\ b_{32} \\ b_{42} \\ \cdot \\ \cdot \\ b_{44} \end{bmatrix}. \tag{4}$$

This problem can be represented as follows

$$Ax = b, \tag{5}$$

where A is a 16x16 matrix corresponding to the evaluations of the bi-cubic polynomial on each of the points where the interpolation is being evaluated; x is a vector whose elements correspond to the 16 unknowns that are needed to compute, and b is the right hand side vector corresponding to the value of each parameter of the original data bank.

Summarizing, 16 input data banks of the same type are needed to solve a bi-cubic interpolation so that once that the interpolation polynomial is generated then it is evaluated at the values of x and y (Gd_2O_3 and ^{235}U,, respectively) given by the user to build up a new data bank.

It is important to point out that when the process to generate a new data bank is initiated a set of 16 data are needed and obtained by the conventional procedure. This is to generate a mesh in X and Y, namely in the variables here above mentioned. During the whole process the mesh remains unchanged and due to the way the set of linear systems is solved the matrix is the same for all the interpolated parameters. This means, that what changes is the right hand side which is precisely the values of each one of the parameters of the data bank on each of the mesh points.

The main disadvantage of this technique is that the condition number of matrix A is high leading to changes for small perturbations in the interpolation parameters (Forsythe, 1981) and (Golub, 1990). In Section 3 this technique will be referred as INTPOLBI-v1.

2.2.- NEWTON BASIS

If u_i's and g_i's correspond to the values of uranium and gadolinium then using the Newton basis we propose the following tensor product

$$\begin{bmatrix} 1 \\ y-u_1 \\ (y-u_1)(y-u_2) \\ (y-u_1)(y-u_2)(y-u_3) \end{bmatrix} \begin{bmatrix} 1 & x-g_1 & (x-g_1)(x-g_2) & (x-g_1)(x-g_2)(x-g_3) \end{bmatrix} \tag{6}$$

from where we get the following matrix

$$A=\begin{bmatrix} 1 & x-g_1 & (x-g_1)(x-g_2) & (x-g_1)(x-g_2)(x-g_3) \\ y-u_1 & (y-u_1)(x-g_1) & (y-u_1)(x-g_1)(x-g_2) & (y-u_1)(x-g_1)(x-g_2)(x-g_3) \\ (y-u_1)(y-u_2) & (y-u_1)(y-u_2)(x-g_1) & (y-u_1)(y-u_2)(x-g_1)(x-g_2) & (y-u_1)(y-u_2)(x-g_1)(x-g_2)(x-g_3) \\ (y-u_1)(y-u_2)(y-u_3) & (y-u_1)(y-u_2)(y-u_3)(x-g_1) & (y-u_1)(y-u_2)(y-u_3)(x-g_1)(x-g_2) & (y-u_1)(y-u_2)(y-u_3)(x-g_1)(x-g_2)(x-g_3) \end{bmatrix} \tag{7}$$

Then, taking a linear combination of the elements of Eq. (7) the following bi-cubic polynomial is produced

$$\begin{aligned} p(x,y) = {} & a_1 + a_2(x-g_1) + a_3(x-g_1)(x-g_2) + a_4(x-g_1)(x-g_2)(x-g_3) + \\ & a_5(y-u_1) + a_6(x-g_1)(y-u_1) + a_7(x-g_1)(x-g_2)(y-u_1) + \\ & a_8(x-g_1)(x-g_2)(x-g_3)(y-u_1) + a_9(y-u_1)(y-u_2) + \\ & a_{10}(x-g_1)(y-u_1)(y-u_2) + a_{11}(x-g_1)(x-g_2)(y-u_1)(y-u_2) + \\ & a_{12}(x-g_1)(x-g_2)(x-g_3)(y-u_1)(y-u_2) + \\ & a_{13}(y-u_1)(y-u_2)(y-u_3) + a_{14}(x-g_1)(y-u_1)(y-u_2)(y-u_3) + \\ & a_{15}(x-g_1)(x-g_2)(y-u_1)(y-u_2)(y-u_3) + \\ & a_{16}(x-g_1)(x-g_2)(x-g_3)(y-u_1)(y-u_2)(y-u_3) \end{aligned} \tag{8}$$

By evaluating *p(x,y)* at the points of the original mesh and in the order that it is shown in the following, one gets the algebraic system $Ax=b$ where matrix A is as follows

$$A=\begin{bmatrix} A_1 & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ A_1 & (u_2-u_1)A_1 & \mathbf{0} & \mathbf{0} \\ A_1 & (u_3-u_1)A_1 & (u_3-u_1)(u_3-u_2)A_1 & \mathbf{0} \\ A_1 & (u_4-u_1)A_1 & (u_4-u_1)(u_4-u_2)A_1 & (u_4-u_1)(u_4-u_2)(u_4-u_3)A_1 \end{bmatrix} \tag{9}$$

with

$$A_1 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 1 & (g_2 - g_1) & 0 & 0 \\ 1 & (g_3 - g_1) & (g_3 - g_1)(g_3 - g_2) & 0 \\ 1 & (g_4 - g_1) & (g_4 - g_1)(g_4 - g_2) & (g_4 - g_1)(g_4 - g_2)(g_4 - g_3) \end{bmatrix} \tag{10}$$

b being the right hand side vector and x the vector of constants $a_1,\ldots,a_{16}$.

Performing several elementary operations with the rows of matrix A one gets an equivalent algebraic system

$$\hat{A}x = \hat{b} \tag{11}$$

where

$$\hat{A} = \begin{bmatrix} A_1 & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & A_1 & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & A_1 & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & A_1 \end{bmatrix} \tag{12}$$

and

$$\hat{b} = \begin{bmatrix} b_1 \\ \dfrac{b_2 - b_1}{u_2 - u_1} \\ \dfrac{b_3 - b_1}{(u_3 - u_2)(u_3 - u_1)} - \dfrac{b_2 - b_1}{(u_3 - u_2)(u_2 - u_1)} \\ \dfrac{b_4 - b_1}{(u_4 - u_3)(u_4 - u_2)(u_4 - u_1)} - \dfrac{b_3 - b_1}{(u_4 - u_3)(u_3 - u_2)(u_3 - u_1)} + \dfrac{b_2 - b_1}{(u_4 - u_2)(u_3 - u_2)(u_2 - u_1)} \end{bmatrix} \tag{13}$$

where

$$b_1 = \begin{bmatrix} p_{k1} \\ p_{k2} \\ p_{k3} \\ p_{k4} \end{bmatrix} \qquad b_2 = \begin{bmatrix} p_{k5} \\ p_{k6} \\ p_{k7} \\ p_{k8} \end{bmatrix} \qquad b_3 = \begin{bmatrix} p_{k9} \\ p_{k10} \\ p_{k11} \\ p_{k12} \end{bmatrix} \qquad b_4 = \begin{bmatrix} p_{k13} \\ p_{k14} \\ p_{k15} \\ p_{k16} \end{bmatrix} \tag{14}$$

and p_{ki}, $i = 1,\ldots,16$ correspond to the values of the data bank.

This means that the original problem has been decomposed in the sequential solution of four algebraic systems of order 4 where what changes is the right hand side. This can be expressed as follows

$$A_1 \, x^* = b^*, \tag{15}$$

where A_1 is a matrix of order 4 with the structure shown previously in Eq. (10), vector b^*

assumes four different values and it is calculated according to Eq. (13), and finally, x^* corresponds to a vector that contains the solution when all the four algebraic systems are solved.

Taking into account what was described in the previous lines we propose the following algorithm to find out the solution of the algebraic system (11) that will be the same as the original one. The algorithm is applied as many times as parameters do exist in the data bank. The algorithm is the following

Algorithm

for i = 1:4

$$p_{kni} = p_{ki}$$

for i = 5:8

$$p_{kni} = \frac{p_{ki} - p_{k(i-4)}}{u_2 - u_1}$$

for i = 9:12

$$p_{kni} = \frac{p_{ki} - p_{k(i-4)}}{(u_3 - u_2)(u_3 - u_1)} - \frac{p_{k(i-4)} - p_{k(i-8)}}{(u_3 - u_1)(u_2 - u_1)}$$

for i = 13:16

$$p_{kni} = \frac{p_{ki} - p_{k(i-4)}}{(u_4 - u_3)(u_4 - u_2)(u_4 - u_1)} - \frac{(p_{k(i-4)} - p_{k(i-8)})(u_4 + u_3 - u_2 - u_1)}{(u_4 - u_2)(u_4 - u_1)(u_3 - u_2)(u_3 - u_1)} + \frac{p_{k(i-8)} - p_{k(i-12)}}{(u_4 - u_1)(u_3 - u_1)(u_2 - u_1)}$$

for i = 1:4

$$x_{(i-1)4+1} = p_{kn(i-1)4+1}$$

$$x_{(i-1)4+2} = \frac{p_{kn(i-1)4+2} - p_{kn(i-1)4+1}}{g_2 - g_1}$$

$$x_{(i-1)4+3} = \frac{p_{kn(i-1)4+3} - p_{kn(i-1)4+1}}{(g_3 - g_2)(g_3 - g_1)} - \frac{p_{kn(i-1)4+2} - p_{kn(i-1)4+1}}{(g_3 - g_2)(g_2 - g_1)}$$

$$x_{(i-1)4+4} = \frac{p_{kn(i-1)4+4} - p_{kn(i-1)4+1}}{(g_4 - g_3)(g_4 - g_2)(g_4 - g_1)} - \frac{p_{kn(i-1)4+3} - p_{kn(i-1)4+1}}{(g_4 - g_3)(g_3 - g_2)(g_3 - g_1)} + \frac{p_{kn(i-1)4+2} - p_{kn(i-1)4+1}}{(g_4 - g_2)(g_3 - g_2)(g_2 - g_1)}$$

With this algorithm the sixteen coefficients of the bi-cubic polynomial are obtained to be able to generate new data banks. This time matrix A has a better condition number than in the previous technique and it will be referred as INTPOLBI-v2 in the next section (Castillo, 1999).

3.- RESULTS

To test the accuracy of the methods here described we built up a grid with 16 lattice data banks already generated by the RECORD code, these banks provides information to four different uranium enrichments (3 w/o, 3.3 w/o, 3.6 w/o, and 4 w/o) and four different gadolinium

enrichments (4 w/o, 12.8 w/o, 22.4 w/o, and 32 w/o). It is important to note that the gadolinium enrichment represents the weight percentage by fuel rod multiplied by the number of fuel rods containing gadolinium in the lattice.

Thus, from this grid we used the program INTPOLBI (versions v1 and v2) (Castillo, 1999) to generate 10 different lattice data banks using the two different versions of our code. Versions v1 and v2 use the canonic and Newton basis, respectively. Table 1 shows the weight percentage of U-235 and gadolinium that each bank contains; furthermore, "I" and "J" should substitute the asterisk of the lattice bank title, for the first and second version of our program INTPOLBI, respectively.

Table 1. Lattice data bank generated using INTPOLBI

Lattice bank title	Gadolinium w/0	U-235 w/0
L1REFC*1.REC	22.4	3.5
L1REFC*2.REC	12.8	3.2
L1REFC*3.REC	32.0	3.8
L1REFC*4.REC	4.0	3.9
L1REFC*5.REC	32.0	3.1
L1REFC*6.REC	12.8	3.7
L1REFC*7.REC	4.0	3.4
L1REFC*8.REC	32.0	3.4
L1REFC*9.REC	4.0	3.1
L1REFC*10.REC	22.4	3.9

We also generated the lattice data banks of Table 1 using the RECORD code to make comparisons of the accuracy produced by the program INTPOLBI.

Each lattice data bank contains all the parameters required to make reactor core simulations under different reactor operation conditions as a function of the lattice burnup. We compared all these parameters produced by INTPOLBI and RECORD code. Table 2 shows the maximum relative errors for each lattice data bank. Furthermore, we display the CPU time invested by INTPOLBI in each run. It is important to notice that each lattice data bank generated by RECORD code takes about 24 hours in our system inasmuch it takes only seconds to generate a new data bank using INTPOLBI as it is shown in Table 2.

Taking particular interest in the multiplication factor one realizes from Figures 1 to 4 that the relative error remains under the 1000 pcm, which is a good indication of the predictability of INTPOLBI for the lattice data bank generation.

As it can be seen the relative error grows as the gadolinium content increases. This is due

to the fact that one is considering this as a unit variable when in reality it corresponds to two variables, number of gadolinium pins and the average of gadolinium content by pin. However, the results show relative errors with a consistent behavior. As it can be seen from Figures 1 to 4 the relative error is greater when the gadolinium is present, it disappears around 10 GWd/tonnU showing a more monotonic behavior of the relative error.

Table 2. Relative errors for the lattice parameters

Program	**Lattice Data Bank**	**Minimum Relative error**	**Maximum relative error**	**CPU time (seconds)**
INTPOLBI v1	L1REFCI1	0.0E0	6.2298E-3	6.0585
INTPOLBI v2	L1REFCJ1	8.5025E-8	6.2913E-3	6.1796
INTPOLBI v1	L1REFCI2	0.0E0	9.033E-3	6.4218
INTPOLBI v2	L1REFCJ2	0.0E0	9.0631E-3	6.1992
INTPOLBI v1	L1REFCI3	0.0E0	1.4715E-2	6.0195
INTPOLBI v2	L1REFCJ3	0.0E0	1.4784E-2	6.1484
INTPOLBI v1	L1REFCI4	0.0E0	1.1343E-2	6.0312
INTPOLBI v2	L1REFCJ4	0.0E0	1.1345E-2	6.4023
INTPOLBI v1	L1REFCI5	6.1056E-8	1.7589E-2	6.8281
INTPOLBI v2	L1REFCJ5	0.0E0	1.768E-2	6.2187
INTPOLBI v1	L1REFCI6	0.0E0	3.0313E-2	6.8984
INTPOLBI v2	L1REFCJ6	0.0E0	3.0333E-2	6.2304
INTPOLBI v1	L1REFCI7	0.0E0	4.9653E-3	6.4609
INTPOLBI v2	L1REFCJ7	0.0E0	4.9647E-3	6.2695
INTPOLBI v1	L1REFCI8	0.0E0	2.9497E-2	6.6914
INTPOLBI v2	L1REFCJ8	0.0E0	2.9597E-2	6.289
INTPOLBI v1	L1REFCI9	0.0E0	1.7986E-2	5.9804
INTPOLBI v2	L1REFCJ9	0.0E0	1.7986E-2	6.5
INTPOLBI v1	L1REFCI10	0.0E0	8.5291E-3	6.2421
INTPOLBI v2	L1REFCJ10	0.0E0	8.5537E-3	6.6601

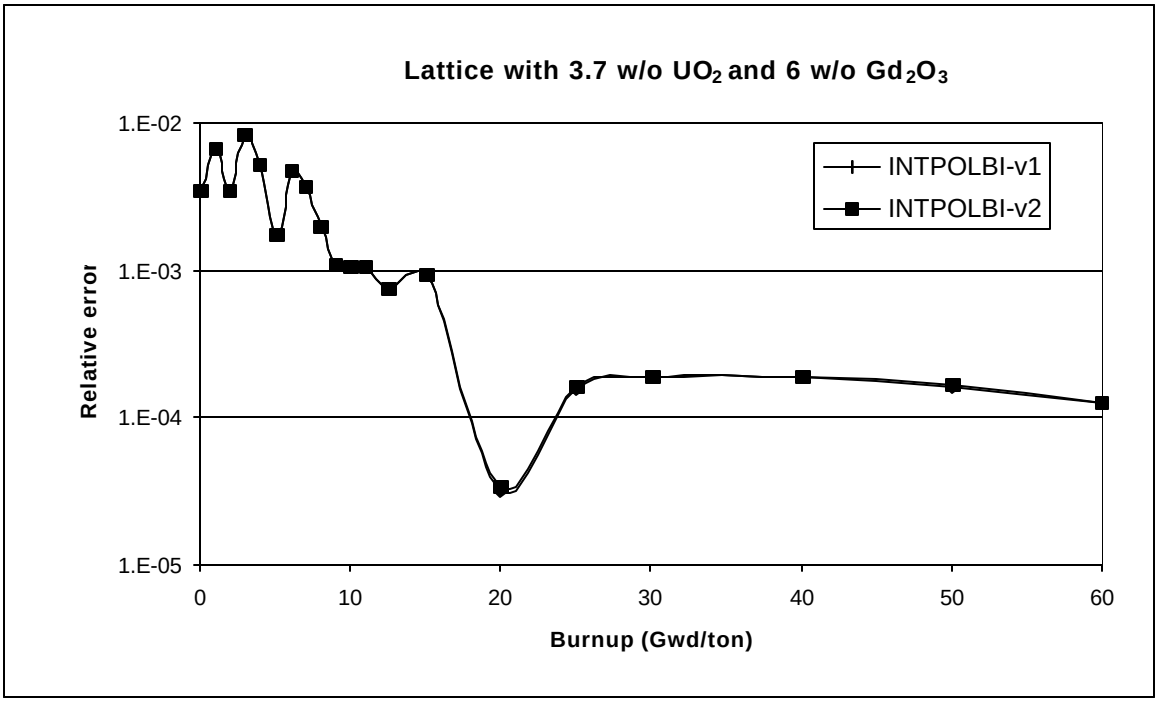

Figure 1. Infinite multiplication factor behavior with a 6w/o gadolinium content.

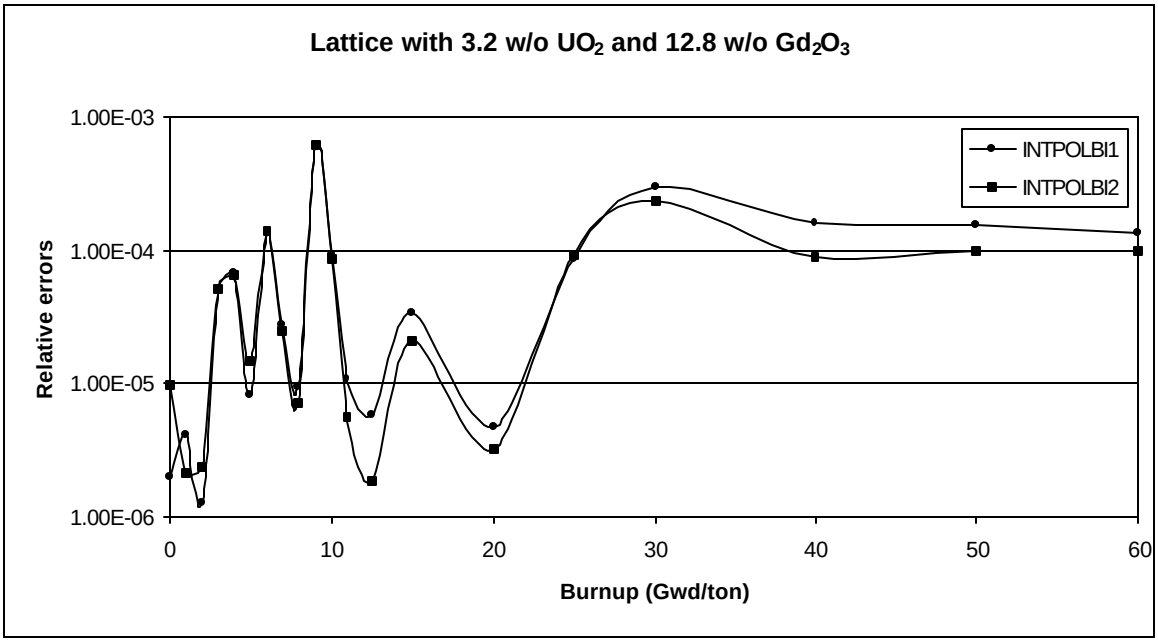

Figure 2. Infinite multiplication factor behavior with a 12.8w/o gadolinium content.

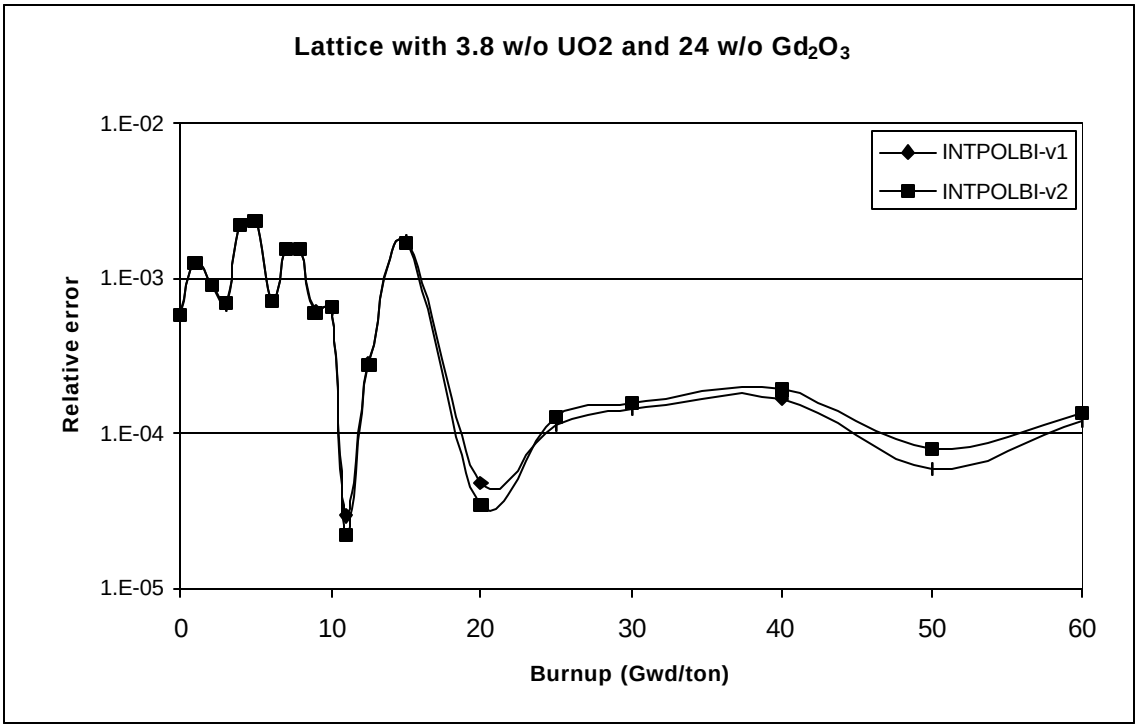

Figure 3. Infinite multiplication factor behavior with a 24w/o gadolinium content.

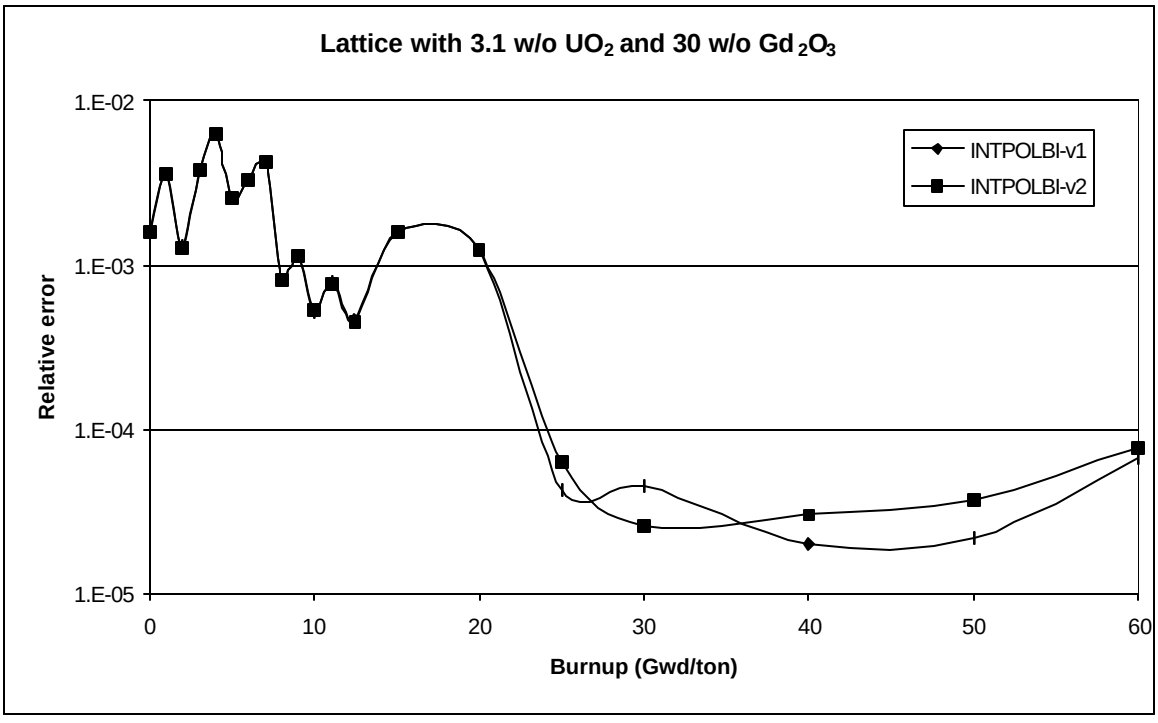

Figure 4. Infinite multiplication factor behavior with a 30 w/o gadolinium content.

4.- CONCLUSIONS

The results generated with ***INTPOLBI*** are in good agreement with those generated with RECORD code, with relative errors less than 10^{-2}. Thus, for the purpose of reload analysis this is a good method to produce lattice data banks in a much faster way, allowing to have more scenarios to assess.

Two interpolation variables were used, uranium enrichment and gadolinium content, however to be more precise and reduce more the error one needs to explore the possibility to create an interpolation that considers three variables. This means the gadolinium content can be expanded in two variables, number of gadolinium pins and the average of gadolinium content by pin.

Currently, the error on the infinite multiplication factor is under 1000 pcm. Therefore it is possible to say that the program generated is very useful and that it will produce acceptable results. On the other hand, once that the adequate uranium and gadolinium enrichment for the analyzed reload have been found, it will be adequate to generate the final assembly design using RECORD to have a more precise analysis in the reactor core analysis.

INTPOLBI is a useful tool that reduces the time invested in reactor core reload analysis, giving the chance to assess more different scenarios of energy generation.

ACKNOWLEDGEMENTS

The authors acknowledge the support of **CONACyT** to develop part of this work under the contract **33806-U.**

REFERENCES

Becker, E. B:, Carey, G. F., Oden, J. T., *Finite Elements, An Introduction* Vol. I, Englewood Cliffs, N. J., Prentice Hall, 1981

Castillo, A., *Generación de Bancos de Datos Nucleares Mediante Interpolación,* Master Thesis, ESFM-IPN, Mexico, 1999.

Forsythe, G; Malcolm, M.; Moler C., *Computer Methods for Mathematical Computations,* Prentice Hall, 1977.

Golub, G; Van Loan, Ch., *Matrix Computations,* Johns Hopkins University Press, 1990.

Hernández, H., Generación de una base de datos para el código FCSII del sistema FMS, Reporte IT.ETN/DSN-280, ININ, November, 1995.

López, H., Esquivel, J., Generación de Bancos de datos nucleares por interpolación, Reporte IIE/13/10046/I 04/P, IIE, October, 1994.

Shardhamar, T.; Halbritter, J. P., User Manual for RECORD, FMS Volume II FMS/2/RECORD/B-01, pp. 10-11, SCANDPOWER, December, 1989.

Torres, C.; Alonso, G., Guía para documentar un Banco Nuclear de Datos de la CNLV con los códigos FMS de SCANDPOWER, Reporte IT.SN/DFR-123, pp. 2-3. ININ, June , 1993.