1999

Application of linear superposition methods to within-lattice loading design optimization of light water reactor nuclear fuel assemblies

Jie Zheng
Iowa State University

INFORMATION TO USERS

This manuscript has been reproduced from the microfilm master. UMI films the text directly from the original or copy submitted. Thus, some thesis and dissertation copies are in typewriter face, while others may be from any type of computer printer.

The quality of this reproduction is dependent upon the quality of the copy submitted. Broken or indistinct print, colored or poor quality illustrations and photographs, print bleedthrough, substandard margins, and improper alignment can adversely affect reproduction.

In the unlikely event that the author did not send UMI a complete manuscript and there are missing pages, these will be noted. Also, if unauthorized copyright material had to be removed, a note will indicate the deletion.

Oversize materials (e.g., maps, drawings, charts) are reproduced by sectioning the original, beginning at the upper left-hand corner and continuing from left to right in equal sections with small overlaps.

Photographs included in the original manuscript have been reproduced xerographically in this copy. Higher quality 6" x 9" black and white photographic prints are available for any photographs or illustrations appearing in this copy for an additional charge. Contact UMI directly to order.

Bell & Howell Information and Learning
300 North Zeeb Road, Ann Arbor, MI 48106-1346 USA

UMI®
800-521-0600

Application of linear superposition methods to within-lattice loading design optimization of light water reactor nuclear fuel assemblies

by

Jie Zheng

A dissertation submitted to the graduate faculty

in partial fulfillment of the requirements for the degree of

DOCTOR OF PHILOSOPHY

Major: Mechanical Engineering

Major Professor: G. I. Maldonado

Iowa State University

Ames, Iowa

1999

Bell & Howell Information and Learning Company
300 North Zeeb Road
P.O. Box 1346
Ann Arbor, MI 48106-1346

Graduate College
Iowa State University

This is to certify that the Doctoral dissertation of

Jie Zheng

has met the dissertation requirements of Iowa State University

Signature was redacted for privacy.

Committee Member

Signature was redacted for privacy.

Committee Member

Signature was redacted for privacy.

Committee Member

Signature was redacted for privacy.

Committee Member

Signature was redacted for privacy.

Major Professor

Signature was redacted for privacy.

For the Major Program

Signature was redacted for privacy.

For the Graduate College

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

1 INTRODUCTION

1.1 Overview of Within-lattice Loading Design Optimization of LWR Nuclear Fuel Assemblies

Minimizing nuclear fuel cycle costs is likely among the most important goals in the successful –and economically competitive– operation of a light water reactor (LWR). In fact, the strategy implemented by a nuclear power generating station to achieve this goal is generally referred to as "nuclear fuel management". However, with the ongoing advent of more powerful computers and emerging advances in combinatorial optimization. the above-noted strategy is now more commonly referred to as "nuclear fuel management optimization".

Strictly speaking, nuclear fuel management extends from mining the ore through the disposal (or re-processing) of the waste products. However, the focuses of optimization applications have been, so far, restricted to two main activities: "out-of-core" and "in-core" fuel management. In out-of-core activities, the cycle energy requirements and cycle length are established, subsequently average enrichment and batch sizes of fresh and burnt fuel are determined. In-core fuel management, on the other hand, a fixed set of nuclear fuel assemblies are judiciously arranged within the core to meet design objectives while satisfying operational and safety constraints.

Different from the core-wide fuel management, within-lattice fuel design deals with the characteristics of a single assembly. The decision variables in the optimization process are the pin distribution, individual pin enrichment and burnable poison (BP) con-

centrations, which constitute the lattice loading pattern (LLP). The LLP is described mathematically by the binary values of $p_{l,m}$, $e_{m,n}$ and $b_{m,k}$, respectively, where

$$p_{l,m} = \begin{cases} 1 & \text{fuel pin type m in location } l \\ 0 & \text{otherwise} \end{cases} \tag{1.1}$$

$$e_{m,n} = \begin{cases} 1 & \text{fuel enrichment candidate } n \text{ for pin type } m \\ 0 & \text{otherwise} \end{cases} \tag{1.2}$$

$$b_{m,k} = \begin{cases} 1 & \text{BP concentration candidate } k \text{ for pin type } m \\ 0 & \text{otherwise} \end{cases} \tag{1.3}$$

The pin type m refers to a group of pins with the same enrichment value and the same burnable poison concentration. For each pin type, the fuel (U^{235}) enrichment and BP concentration can correspond to a "palette" of options available at the manufacturing level.

Constraints include those on maximum power peaking at each burnup steps,

$$P(i) \leq P_{max}(i) \quad \text{for all burnup step } i \tag{1.4}$$

and minimum and maximum assembly-averaged multiplication factors at each burnup steps,

$$k_{\infty}^{min}(i) \leq k_{\infty}(i) \leq k_{\infty}^{max}(i) \quad \text{for all burnup step } i \tag{1.5}$$

and maximum assembly-averaged fuel (U^{235})enrichment,

$$E \leq E_{max} \tag{1.6}$$

Besides, the decision variables $p_{l,m}$,$e_{m,n}$ and $b_{m,k}$ must also satisfy:

- One pin per location $\Rightarrow \Sigma_m^{all} p_{l,m} = 1$ for all l
- Limit on the number of fuel pins of each pin type available for loading $\Rightarrow \Sigma_l^{all} p_{l,m} = N_m$ for all m. If the total number of pin types and the number of pins for each pin

type are fixed, it is required that $\Sigma_m^{all} N_m = N$, where N is the total number of pins in an assembly.

- At most one unique fuel enrichment value for each pin type at one time $\Rightarrow$ $\Sigma_n^{all} e_{m,n} = 1$ for fuel pin types and $\Sigma_n^{all} e_{m,n} = 0$ for non-fuel pin types, such as a water hole.

- At most one unique BP concentration value for each pin type at one time $\Rightarrow$ $\Sigma_k^{all} b_{m,k} = 1$ for the pin types containing BP content and $\Sigma_k^{all} e_{m,k} = 0$ for the pin types without BP content.

- Limit on the total number of BP pins allowed for loading $\Rightarrow$ $\Sigma_{l,\hat{m}}^{all} p_{l,\hat{m}} \leq N_b^{max}$. where, the pin type $\hat{m}$ is any pin type containing BP content.

- Exclusion one specific pin type from one certain location $\Rightarrow$ $p_{\bar{l},\bar{m}} = 0$ for exclusion pin type $\bar{m}$ from location $\bar{l}$

- Freezing one specific pin type on one certain location $\Rightarrow$ $p_{\bar{l},\bar{m}} = 1$ for freezing pin type $\bar{m}$ at location $\bar{l}$

The objective functions can be the minimization of power peak throughout the cycle, minimizing the assembly-averaged enrichment or the total uranium cost, maximizing EOL (end of life) reactivity, or the combination of minimizing power peaking and minimizing the assembly-averaged enrichment. In the problem studied, both objective functions and constraints are functions of the lattice loading pattern.

The statement of the optimization problem is the determination of all decision variables $p_{l,m}$, $e_{m,n}$ and $b_{m,k}$ such that the objective function is optimized subject to the constraints of the above equations. The problem is classified as a large-scale combinatorial optimization problem with non-linear objective function and constraints, that are computationally intensive to evaluate. The multi-modality and the lack of direct derivative information add to the complexity.

1.2 Overview of Simulated Annealing Algorithm

Stochastic optimization methods have proved to be capable of finding good approximations to the exact solutions of combinatorial problems far more efficiently than other conventional methods. The algorithm employed in this work has been the Simulated Annealing algorithm, a statistical mechanics tool based on the work of Metropolis *et al.* which was first introduced as a means of finding the equilibrium configuration of a collection of particles at a given temperature. Its major advantage over other methods is an ability to avoid becoming trapped at meta-stable states (*e.g.* local minima). The algorithm employs a random search procedure in which individual particle positions within the system configuration are perturbed, thereby producing a corresponding change in the energy state of the system. The algorithm not only accepts changes that decrease the system energy but also some changes that increase it. In the original problem, the latter were accepted with a probability given by the Boltzmann factor:

$$P = \exp\left(-\frac{\delta E}{k\bar{T}}\right) \tag{1.7}$$

where δE is the change in the system energy, k is the Boltzmann constant and $\bar{T}$ is the absolute system temperature. Each configuration accepted using the above criteria becomes the current "best" solution to which further system perturbations are applied. Over many random histories the computer simulation of the system approaches toward thermal equilibrium with its macroscopic parameters fluctuating about their mean values with a Boltzmann distribution appropriate to the temperature.

The connection between the Metropolis algorithm and minimization was first noted by Pincus but it was Kirkpatrick *et al.* who proposed that it form the basis of an optimization technique for combinatorial problems, in which a set of candidate solutions to minimize objective function F_{obj} is generated by random moves. Moves that increase

F_{obj} by δF_{obj} are accepted with probability:

$$P = \exp\left(-\frac{\delta F_{obj}}{T}\right) \tag{1.8}$$

where T is a control parameter, which by analogy is known as the system "temperature" irrespective of the objective function involved. As T is usually varied during the optimization search according to an "annealing schedule", which alludes to why this technique is commonly known as optimization by simulated annealing.

When applied to address the realistic LWR assembly within-lattice loading pattern optimization problem, the simulated annealing algorithm has a number of attractive features:

- The lack of a requirement for functional derivative information. Simulated annealing algorithm is thus unaffected by the type of nonlinearities that cause problems to more conventional optimization techniques.
- The ability to escape from local minima in multi-modal problems.
- The ability to identify not only a single solution but "families" of solutions in the vicinity of the global optimum.
- The versatility to select from a variety of objective function formulations.

1.3 Literature Review

Documentation on within-lattice optimization problem is fairly scarce, but a considerable amount of literature exists in the area of in-core fuel management and optimization. Because the two problems have many common features, a review of the studies of the latter problem can provide insights applicable to the research of the former. Thus, a detailed review of the few available papers on within-lattice optimization is provided

below, and it is followed by a brief review of some of the key literature in the area of in-core fuel management optimization.

Lim and Leonard formulated an optimization problem of which the objective was to best approximate a prescribed power distribution in a two-dimensional 8×8 BWR fuel assembly by selecting an optimal fuel enrichment distribution. One of the major constraints was to keep the assembly-averaged enrichment constant, which was determined typically by a fuel cycle analysis and to ensure that a suitable value of reactivity is present in the assembly. The lattice-physics parameters were calculated using a fast, self-developed code in which two-dimensional response matrix techniques were employed. A projected gradient method was adapted to the special case with the assumption that each pin has a unique enrichment. For a general case in which only a few number of enrichment types are allowed, a nonlinear mixed integer programming problem was solved using a pattern search method. The weakness of this study is that it is limited to BOL (beginning of life) only and burnable poison pins were not considered. Also, the accuracy of the self-developed code would become an issue due to the stringent licensing requirements of the nuclear industry.

Hirano *et al.* divided the within-lattice optimization problem into two sub-problems, one of which groups fuel rods into a set number of rod groups and at the same time determines the pin distribution, and the other is to optimize the enrichment of each pin group based on the resultant pin grouping pattern. The objective is to minimize a compound function which considers both the power peaking and power within gadolinium (Gd_2O_3) pins. The constraints include constant assembly-averaged enrichment, upper limits of both power peaking and gadolinium pin power. The lattice-physics computer code used is TGBLA. The algorithm for solving the grouping problem started with finding the optimal enrichment for each pin without any grouping, and then determining the pin group and pin distribution by exhaustive enumeration based on the resulting fuel enrichment ordering. The optimal enrichment for each pin type was obtained using

a method of approximation programming. An 8×8 BWR fuel bundle with MOX and Gd_2O_3 pins was tested for several depletion steps.

Adielson developed a GUI for CASMO data handling. It is also used for fuel assembly uranium enrichment and gadolinium concentration design and optimization work. The CASMO-3 or CASMO-4 code is adapted and perturbation models have been implemented for fast calculation of pin powers and k_∞. The optimization objective is to maximize dry-out performance indicated by the BTF distribution. The constraints can be given as constant assembly-averaged uranium enrichment, maximum permitted power peaking factor as a function of burnup, and so on. Given the number of different fuel rod types and an initial assembly pattern, the whole optimization method was divided into two sub-problems. The first sub-problem determines the number of pins for each pin type and the pin position distribution, and the other pursues the optimal enrichment levels for each pin type based on the resulting pin distribution from the first one. The two sub-problems are repeated a number of times depending on the given number of different fuel rod types. Though the optimization methods are described in detail, the specifics of the perturbation model are not divulged. A 9×9 BWR fuel bundle was studied. In this case, the change of Gd_2O_3 content in a fuel rod is handled automatically, as well as introducing and removal of burnable absorbers can be supported.

Different optimization algorithms have been studied for the in-core nuclear fuel optimization problem, including integer programming, genetic algorithms, artificial intelligence techniques (expert systems, artificial neural networks *etc.*), simulated annealing and other hybrid methods. Simplifications of the neutron physics calculations required have included backward diffusion, neural networks, and generalized perturbation theory (GPT). Brief descriptions of some of the key studies follow.

Quist, van Geemert, *et al.* (1999) applied mixed-integer nonlinear programming (MINLOP) in reactor core fuel reloading optimization, proving that MINLOP combined with local search heuristics is a promising approach.

Hida and Yoshioka (1992) studied the optimization of axial enrichment and Gadolinia distribution for BWR fuel under control rod programming. The successive linear programming method was used. Mahlers, Yu. P. (1991) developed an optimization algorithm using backward diffusion calculation.

Maldonado, Turinsky, *et al.* (1995) extended the FORMOSA-P code to utilize nodal-based (nodal expansion method) GPT and to perform feed enrichment minimization.

Yamamota, Noda, *et al.* (1997) developed an integrated scoping analysis tool for in-core fuel management of PWR by incorporating a loading pattern (LP) optimization module (GALLOP), an interactive LP design module (PATMAKER) and other utilities. The algorithm behind GALLOP is a hybrid genetic algorithm.

Axmann (1997) developed the RELOPAT optimization program by combining traditional evolution strategies (genetic algorithm) with heuristics from expert knowledge. Furthermore, parallel computing was introduced successfully based on the widely distributed parallel virtual machine (PVM). Generally three to five workstations were used.

Kim, Chang, *et al.* (1993) connected a rule-based system, fuzzy logic and an artificial neural network (ANN) with each other to achieve optimal design of PWR fuel loading patterns. The rule-based system classified the loading pattern into two types, while the fuzzy rule is helpful to get a more effective and faster search. ANN predicts core parameters for each pattern. The results showed that the ANN and fuzzy logic can be used to improve the capabilities of existing algorithms.

Siegelmann, Nissan, *et al.* (1997) developed an expert system, FUELCON, for optimized refueling design in nuclear engineering. A rule set revision part has been added to the code recently to update the rule set manually by an expert or automatically by a neural-network-based learning routine.

Parks and Knight (1990) integrated PANTHER with an automatic optimization procedure based on simulated annealing algorithm to search for optimal PWR reload core.

Kropaczek and Turinsky (1991) developed an in-core fuel management code for pressurized water reactor reload design that combined the stochastic optimization technique of simulated annealing with a computationally efficient core physics model based on second-order accurate generalized perturbation theory. This work proved that the simulated annealing algorithm is a powerful tool to solve practical reload problems. Subsequently, Maldonado and Turinsky added a nodal-based (Nodal Expansion Method) GPT model to the code which emerged from that research, namely, FORMOSA-P.

Parks (1990) made some innovations to enhance simulated annealing both through artificial intelligence learning features and by applying some design heuristics. It was proven that it is significantly faster and in several ways better than the standard approach of simulated annealing.

Smuc, Pevec and Petrovic (1994) improved upon the traditional simulated annealing algorithm via an adaptive trial loading pattern generator, in which the ENS (exact neighborhood structure) and BNS (binary exchange neighborhood structure) mode are alternately applied in one cooling cycle.

Stevens, Smith, *et al.* (1995) implemented design heuristics within simulated annealing, and the results showed that utilizing the designer's judgment during automated pattern generation can be effective.

From the above literature review, it should be noted that published works in the within-bundle loading optimization area are fairly limited in comparison to the abundant published documentation found in in-core reload optimization. Nevertheless, the analogies that can be drawn from in-core optimization studies can certainly benefit this research in several ways. For example, two important trends are observed within the realm of simulated annealing applications to in-core nuclear fuel optimization. First, the bare-bones simulated annealing strategy can, in fact, be adaptively improved to a specific application via expert-based systems or heuristically-driven artificial intelligence techniques. The second trend is that of speeding up the design-related parameter

evaluations via faster, yet still accurate models. Finally, another not-so-prevalent trend is that of employing parallel computing by using clusters of engineering workstations, which shows great promise for speeding up the linear superposition models (LSM) even further.

1.4 Objective of Research

The global objective of this research fits within out-of-core and in-core nuclear fuel management and concerns itself with developing a practical method to automatically determine an optimal within-assembly "pin-by-pin" loading pattern for fresh LWR fuel assemblies. In other words, this optimization application is intended to "fine-tune" each fresh fuel assembly representing a feed (fresh) out-of-core batch before it enters the in-core decision process.

This line of research originally led to the development of the FORMOSA-L code, which under sponsorship by the Electric Power Research Center of North Carolina State University is intended to ultimately couple to the suite of in-core nuclear fuel management optimization codes FORMOSA-P and FORMOSA-B for Pressurized and Boiling water reactors, respectively. The initial version of FORMOSA-L was developed at Iowa State University by coupling a simulated annealing optimization engine to one of EPRI's vintage lattice-physics code, CPM-2, for parameter evaluation. More recently, however, the option to employ a much more modern and sophisticated code, CASMO-3, has been implemented.

One of the major drawbacks of the earlier versions of the FORMOSA-L code was that its computational requirements for standard nuclear design calculations were impractical. This was true because each history of the optimization would launch a large-scale lattice-physics calculation. For example, when employing the CPM-2 code, it may require at least 5 CPU-seconds per history and per burnup step on a -DEC Alpha station- 300MHz

engineering workstation. Therefore, a 10,000 history FORMOSA-L optimization with 25 depletion steps in the lattice calculation could require as much as 2 CPU-weeks for execution.

To reduce the above-noted exorbitant computational requirements, a couple of obvious alternatives may include using a simplified –yet accurate– less time-consuming lattice-physics evaluator, and/or possibly performing some of the calculations in parallel. This research has implemented both of the above-noted options into FORMOSA-L. A first-order approximation technique was developed based upon the linear superposition of pre-evaluated single-pin material loading and/or spatial perturbations stored into a library. In this manner, during an optimization, lattice-physics parameters such as relative pin power and assembly-average k_∞ can be estimated to first order accuracy for multiple simultaneous perturbations occurring within a bundle. The linear superposition models (LSMs) developed have reduced the run time requirements of FORMOSA-L by at least an order of magnitude, while the degradation in accuracy of the evaluated parameters has been kept at a minimal by limiting the size of perturbations. Further accuracy improvements to the LSM, such as interpolations and second-order cross terms compensation, have been developed. Similarly, a parallel approach based on remote procedure call (RPC) has been studied for further speedups. Both synchronous implementation and asynchronous implementation have been developed for parallel computing of LSM library creation via RPC technique. In general, the emphasis of this research revolves around developing a technique that will make the FORMOSA-L code computationally practical by employing LSM, first, and then also by taking advantage of parallel processing.

This dissertation is organized as follows. The methodologies developed, Linear Superposition Models, are addressed in Chapter 2. Chapter 3 shows the results of error analysis of Linear Superposition Models. Further speed-up is provided by parallel computing of LSM library via the RPC technique, which is presented in Chapter 4. Chapter

5 explores the applications of LSM into the FORMOSA-L code, and the results from fidelity studies and time saving studies of LSMs are presented. A complete conclusion is given in Chapter 6.

2 LINEAR SUPERPOSITION MODELS

2.1 Background behind the LSM

A number of linear approximation techniques have been proven to be extremely useful in nuclear fuel management calculations when properly implemented. Generally, the aim has been to reduce the large CPU-time requirements of direct calculations by implementing simpler linear models that can adequately predict nuclear fuel behavior. In the case of the FORMOSA-L code, linear superposition models (LSMs) have been developed, tested, and implemented into the latest version of the code [Maldonado, 1998].

A lattice-physics calculation can be thought of as an arbitrary function whose independent variables are the design parameters which describe the fuel assembly (*i. e.* code input), and whose dependent variables include attributes such as the assembly-average k_∞ profile and/or relative pin power distribution versus burnup (*i. e.* code output). Accordingly, the basic principle behind the LSM is based upon casting the dependent variables into Taylor's series expansions in terms of the independent variables expanded about a selected reference assembly, where the truncation of second and higher-order terms makes this a first-order accurate model.

Assume an LWR fuel assembly contains N pins and M types of pins, where type refers to pure fuel, fuel with integral burnable poison, discrete burnable poison pins, water hole, etc.. Then the following vectors are defined:

$\vec{P}$: Pin power distribution, where P_i denotes the relative pin power at assembly spatial position i (N-vector)

$\vec{L}$: Pin type spatial distribution, where L_i denotes the pin type at assembly spatial position i (N-vector)

$\vec{E}$: Pin material distribution, where E_j denotes the material composition (*e.g.* fuel and/or burnable poison concentration) for the j^{th} pin type ($2M$-vector)

Then, we define two set of functions, $\vec{f}$ and g, to represent the relationship between the dependent variables ($\vec{P}$ and k_∞) and the independent variables ($\vec{L}$, $\vec{E}$, *etc.*), as follows:

$$\vec{P} = \vec{f}(\vec{L}, \vec{E}, etc.) \tag{2.1}$$

$$k_\infty = g(\vec{L}, \vec{E}, etc.) \tag{2.2}$$

Accordingly, the following Taylor's Series functionality for the assembly's power distribution and average k_∞ are assumed:

$$\vec{P} = \vec{f}(\vec{L}, \vec{E}, etc.) = \vec{P}_0 + \left[\frac{\partial \vec{f}}{\partial \vec{L}}\right]_0 \cdot \triangle \vec{L} + \left[\frac{\partial \vec{f}}{\partial \vec{E}}\right]_0 \cdot \triangle \vec{E} + O(\triangle^2) \tag{2.3}$$

$$k_\infty = g(\vec{L}, \vec{E}, etc.) = k_{\infty,0} + \left[\frac{\partial g}{\partial \vec{L}}\right]_0 \cdot \triangle \vec{L} + \left[\frac{\partial g}{\partial \vec{E}}\right]_0 \cdot \triangle \vec{E} + O(\triangle^2) \tag{2.4}$$

where the subscript "0" denotes a reference (unperturbed) condition, and the first-order accuracy approximation manifests itself when the second and higher-order terms are neglected, as shown below where the superscript "*" denotes the estimated quantities.

$$\vec{P} \approx \vec{P}^* = \vec{P}_0 + \left[\frac{\partial \vec{f}}{\partial \vec{L}}\right]_0 \cdot \triangle \vec{L} + \left[\frac{\partial \vec{f}}{\partial \vec{E}}\right]_0 \cdot \triangle \vec{E} \tag{2.5}$$

$$k_\infty \approx k_\infty^* = k_{\infty,0} + \left[\frac{\partial g}{\partial \vec{L}}\right]_0 \cdot \triangle \vec{L} + \left[\frac{\partial g}{\partial \vec{E}}\right]_0 \cdot \triangle \vec{E} \tag{2.6}$$

Note that Eq.(2.3) and Eq.(2.5) involve matrix-vector multiplication; whereas, Eq.(2.4) and Eq.(2.6) involve vector-vector multiplication.

From the view point of neutron physics, the LSM has a theoretical basis. All nuclear features of a nuclear assembly are determined by the neutron flux within the assemblies' geometrical space. In turn, the neutron behavior is governed by the neutron transport equation, which is a linear integro-differential equation. The linearity property of the neutron transport equation supports the LSM. When some combined perturbations are made to a reference assembly, the changes in relative power distribution and the multiplication factor can be approximately estimated by summing up the changes due to all involved single basic perturbations.

2.2 Separation of Material and Spatial Libraries

Extensive computational experiments have been conducted with the LSM. Accordingly, it has been concluded by observation that to maintain an acceptable level of accuracy, the material perturbations (*i.e.* enrichment or burnable absorber concentration changes) and the spatial re-arrangements (*i.e.* pin shuffles) should be best performed independently of each other during an optimization. Fortunately, the simulated annealing strategy is well-suited to handle this, so that material and spatial changes can be done in alternating cooling cycles. In other words, when evaluating changes due to spatial perturbations, the material properties are not perturbed ($\triangle \vec{E} = 0$). Likewise, when evaluating changes due to material perturbations, the spatial arrangement of the pins remains unperturbed ($\triangle \vec{L} = 0$). This separability assumption during optimizations was deemed appropriate to employ because of the considerable improvements in error performance it produced relative to treating both types of perturbations simultaneously. Although the concern of this approach is that trapping in local minima may occur, so far, no measurable degradation in optimization fidelity has been observed as a consequence

of this assumption.

The first key aspect of the LSM methodology is the construction of the appropriate linear superposition libraries, which are developed with respect to a reference (unperturbed) lattice physics calculation. Based upon the above-noted separability assumption, one of the two types of linear superposition libraries can be created during a simulated annealing cooling cycle; namely, a library involving only spatial perturbations, and a library involving only material perturbations. Each is described in the section which follows [Maldonado, G. I., (1999)].

2.2.1 Spatial Perturbation Library

The following sequence applies to the creation of a spatial perturbation library within the LSM and the subsequent calculation of first-order accurate estimates employing that library.

1. Select a reference assembly ($\vec{L}_0$, $\vec{E}_0$) and calculate $\vec{P}_0$ and $k_{\infty,0}$ via a direct lattice-physics evaluation (CPM-2/CASMO-3).

2. Hold $\vec{E}_0$ constant (*i.e.*, $\triangle\vec{E} = 0$) and determine the basic spatial perturbations (BSPs):

$$\triangle\vec{L}_{ij}; i \in \mathcal{N}, j \in \left(\mathcal{M} - \left\{\vec{L}_0(i)\right\}\right) \tag{2.7}$$

   where $\triangle\vec{L}_{ij}$ represents a perturbation occurring only at the ith location of the reference assembly, where the pin is changed from type $\vec{L}_0(i)$ to type j. $\mathcal{N}$ is the set of natural integer numbers from 1 to N and $\mathcal{M}$ is the set of natural integer numbers from 1 to M.

3. For each BSP, $\triangle\vec{L}_{ij}$, calculate the change of $\vec{P}$ and k_∞ with respect to the reference $\vec{P}_0$ and $k_{\infty,0}$, say, $\triangle\vec{P}_{ij}$ and $\triangle k_{\infty,ij}$. Then, the Spatial Perturbation Library $\mathcal{SP}$ is

now defined by:

$$\mathcal{SP} = \left\{\left(\vec{L}_0, \vec{P}_0, k_{\infty,0}\right), \left(\Delta\vec{L}_{ij}, \Delta\vec{P}_{ij}, \Delta k_{\infty,ij}\right)\right\}; i \in \mathcal{N}, j \in \left(\mathcal{M} - \left\{\vec{L}_0(i)\right\}\right) \quad (2.8)$$

4. During an optimization, a given candidate assembly's spatial distribution $\vec{L}$ needs to be decomposed into its particular BSPs with respect to the reference assembly spatial distribution $\vec{L}_0$:

$$\vec{L} = \vec{L}_0 + \sum_{(k,l)\in\mathcal{P}} \Delta\vec{L}_{kl}; \mathcal{P} \subset \mathcal{N} \times \mathcal{M} \quad (2.9)$$

5. Finally, the attributes $\vec{P}$ and k_∞ can be quickly estimated by the following summations:

$$\vec{P} \approx \vec{P}_0 + \sum_{(k,l)\in\mathcal{P}} \Delta\vec{P}_{kl} \quad (2.10)$$

$$k_\infty \approx k_{\infty,0} + \sum_{(k,l)\in\mathcal{P}} \Delta k_{\infty,kl} \quad (2.11)$$

2.2.2 Material Perturbation Library

Analogously to the previous section, The following sequence applies to the creation of a material perturbation library within the LSM and the subsequent calculation of first-order accurate estimates employing that library. It is noted that several material perturbation levels for each pin type's enrichment and/or BA concentration are adapted in the material library, which is called multi-level LSM and is used to reduce the round-off error by limiting of the number of summation terms in step 5.

1. Select a reference assembly ($\vec{L}_0$, $\vec{E}_0$) and calculate $\vec{P}_0$ and $k_{\infty,0}$ via a direct lattice-physics evaluation (CPM-2/CASMO-3).

2. Hold $\vec{L}_0$ constant (*i.e.*, $\Delta\vec{L} = 0$) and determine the basic material perturbations (BMPs):

$$\Delta\vec{E}_{jm}; j \in \mathcal{M}, m \in \mathcal{NE}_j \quad (2.12)$$

where $\triangle\vec{E}_{jm}$ represents a discrete material perturbation involving only the jth type pins of the reference assembly, where the fuel enrichment (or burnable absorber composition) is changed to a new level m. $\mathcal{NE}_j$ is the set of natural integer number from 1 to NE_j, where NE_j is the total number of pre-defined (via code input) fuel enrichment (or burnable absorber composition) levels for pin type j.

3. for each BMP, $\triangle\vec{E}_{jm}$, calculate the change of $\vec{P}$ and k_∞ with respect to the reference $\vec{P}_0$ and $k_{\infty,0}$, say, $\triangle\vec{P}_{jm}$ and $\triangle k_{\infty,jm}$. Then, the Material Perturbation Library $\mathcal{MP}$ is now defined by:

$$\mathcal{MP} = \left\{\left(\vec{E}_0, \vec{P}_0, k_{\infty,0}\right), \left(\triangle\vec{E}_{jm}, \triangle\vec{P}_{jm}, \triangle k_{\infty,jm}\right)\right\}; j \in \mathcal{M}, m \in \mathcal{NE}_j \tag{2.13}$$

4. During an optimization, a given candidate assembly's material distribution $\vec{E}$ needs to be decomposed into its particular BMPs with respect to the reference assembly's material distribution $\vec{E}_0$:

$$\vec{E} = \vec{E}_0 + \sum_{(k,l)\in\mathcal{Q}} \triangle\vec{E}_{kl}; \mathcal{Q} \subset \mathcal{M} \times \mathcal{NE}_j \tag{2.14}$$

5. Finally, the attributes $\vec{P}$ and k_∞ can be quickly estimated by the following summations:

$$\vec{P} \approx \vec{P}_0 + \sum_{(k,l)\in\mathcal{Q}} \triangle\vec{P}_{kl} \tag{2.15}$$

$$k_\infty \approx k_{\infty,0} + \sum_{(k,l)\in\mathcal{Q}} \triangle k_{\infty,kl} \tag{2.16}$$

2.2.3 Examples

For simplicity and clarity, consider a hypothetical 5 × 5 LWR assembly with octant symmetry and three pin types, enumerated by 1, 2, and 3, respectively. It is assumed that all the three pin types contain fuel (U^{235}) composition and no burnable poison. The fuel enrichment of the three pin types are denoted by E1, E2 and E3, respectively.

Figure 2.1 illustrates the spatial perturbation library structure and lattice-physics parameters estimation of the spatial perturbation assembly. Similarly, those of material perturbation library are shown in Figure 2.2. In both figures, all the information of an assembly are represented by a shaded square, including not only the pin distribution and pin enrichment but also its lattice-physics parameters, such as relative pin power distribution and k_∞. The non-shaded square denotes a BMP or BSP and the difference between its lattice-physics parameters and those of the reference assembly. Inside any one of these squares, either the triangle-shaped one-eighth pin distribution map (in Figure 2.1) or the fuel enrichment (in Figure 2.2) are presented.

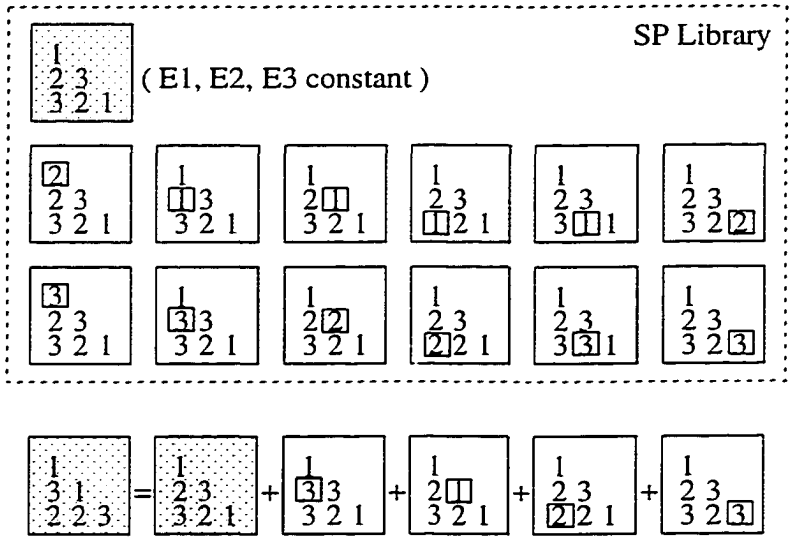


Figure 2.1 Example of spatial perturbation library

2.3 LSM with Combined Library

Due to the widely diverse nature of modern LWR fuel assemblies and under fairly specific circumstances, the separation of the material and spatial perturbation libraries could potentially lead to measurable misdirecting the optimizations due to local minima trapping. An additional not-so-obvious drawback is that at least twice as many SA

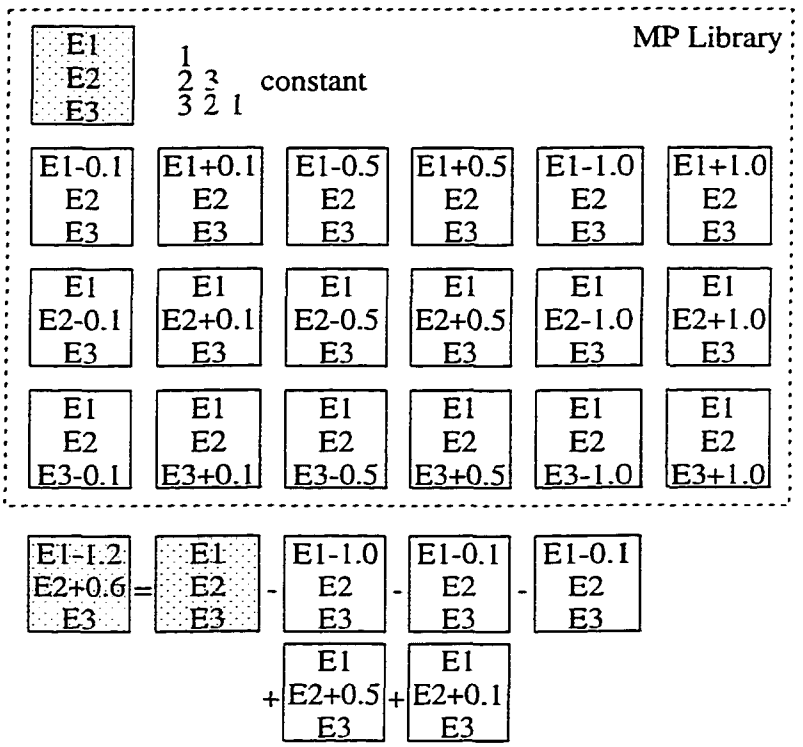


Figure 2.2 Example of material perturbation library

cooling cycles are generally required during the optimization process if material and spatial perturbations are performed independently of each other. Furthermore, it should be obvious that carrying out material and spatial perturbations simultaneously ought to lead to a more thorough global sweeping of the search space. In fact, a better scoping of the vast search space at hand could outweigh the impact of the larger prediction errors expected from the combined library treatment. Consequently, despite the reasonable results obtained for the separated library method, a unified or "combined" perturbation library has studied in order to try to answer the above questions.

2.3.1 Generation of a Combined Library

The following sequence applies to the creation of a combined library within the LSM and the subsequent calculation of the first-order accurate estimates utilizing that library. Most of the notation is similar to that used in the previous sections to describe

the creation of the material and the spatial perturbation libraries.

1. Select a reference assembly ($\vec{L}_0$, $\vec{E}_0$) and calculate $\vec{P}_0$ and $k_{\infty,0}$ via a direct lattice-physics evaluation (CPM-2/CASMO-3).

2. Hold $\vec{E}_0$ constant (*i.e.*, $\triangle\vec{E} = 0$) and determine the basic spatial perturbations (BSPs). For each BSP, $\triangle\vec{L}_{ij}$, calculate the change of $\vec{P}$ and k_∞ with respect to the reference $\vec{P}_0$ and $k_{\infty,0}$, say, $\triangle\vec{P}^S_{ij}$ and $\triangle k^S_{\infty,ij}$. Then, the Spatial Perturbation Library $\mathcal{SP}$ is now defined by:

$$\mathcal{SP} = \left\{\left(\vec{L}_0, \vec{P}_0, k_{\infty,0}\right), \left(\triangle\vec{L}_{ij}, \triangle\vec{P}^S_{ij}, \triangle k^S_{\infty,ij}\right)\right\}; i \in \mathcal{N}, j \in \left(\mathcal{M} - \left\{\vec{L}_0(i)\right\}\right) \quad (2.17)$$

where $\triangle\vec{L}_{ij}$ represents a perturbation occurring only at the ith location of the reference assembly, where the pin is changed from type $\vec{L}_0(i)$ to type j. $\mathcal{N}$ is the set of natural integer numbers from 1 to N, and $\mathcal{M}$ is the set of natural integer numbers from 1 to M.

3. Hold $\vec{L}_0$ constant (*i.e.*, $\triangle\vec{L} = 0$) and determine the basic material perturbations (BMPs) based on the reference assembly. For each BMP, $\triangle\vec{E}_{jm}$, calculate the change in $\vec{P}$ and k_∞ with respect to the reference $\vec{P}_0$ and $k_{\infty,0}$, say, $\triangle\vec{P}^{M0}_{jm}$ and $\triangle k^{M0}_{\infty,jm}$. Then, the Material Perturbation Library $\mathcal{MP}_0$ is now defined by:

$$\mathcal{MP}_0 = \left\{\left(\triangle\vec{E}_{jm}, \triangle\vec{P}^{M0}_{jm}, \triangle k^{M0}_{\infty,jm}\right)\right\}; j \in \mathcal{M}, m \in \mathcal{NE}_j \quad (2.18)$$

where $\triangle\vec{E}_{jm}$ represents a discrete material perturbation involving only the jth type pins of the reference assembly, where the fuel enrichment or burnable absorber concentration are changed to a new level m. $\mathcal{NE}_j$ is the set of natural integer number from 1 to NE_j, where NE_j is the total number of input-defined fuel enrichment or burnable absorber concentration levels allowed for pin type j.

4. For each perturbed assembly ($\vec{L}_0 + \triangle\vec{L}_{ij}$, $\vec{E}_0$) from Step 2, repeat what was done on the previous step and obtain the Material Perturbation Library $\mathcal{MP}_i$ with respect

to that perturbation, as follows:

$$\mathcal{MP}_i = \left\{\left(\Delta\vec{E}_{jm}, \Delta\vec{P}^{Mi}_{jm}, \Delta k^{Mi}_{\infty,jm}\right)\right\}; j \in \mathcal{M}, m \in \mathcal{NE}_j \tag{2.19}$$

where i belongs to the set $\mathcal{NS}$, which is the set of integer numbers from 1 to NS. the total number of BSPs from Step 2.

5. The combined library $\mathcal{CP}$ then is simply the union of $\mathcal{SP}$, $\mathcal{MP}_0$ and $\mathcal{MP}_i$.

$$\mathcal{CP} = \mathcal{SP} \cup \left(\bigcup_{i=0}^{NS} \mathcal{MP}_i\right) \tag{2.20}$$

6. During an optimization, a given candidate assembly ($\vec{L}$, $\vec{E}$) needs to be decomposed into its particular BSPs and BMPs with respect to the reference assembly ($\vec{L}_0$, $\vec{E}_0$):

$$\vec{L} = \vec{L}_0 + \sum_{(k,l)\in\mathcal{P}} \Delta\vec{L}_{kl}; \mathcal{P} \subset \mathcal{N} \times \mathcal{M} \tag{2.21}$$

$$\vec{E} = \vec{E}_0 + \sum_{(m,n)\in\mathcal{Q}} \Delta\vec{E}_{mn}; \mathcal{Q} \subset \mathcal{M} \times \mathcal{NE}_j \tag{2.22}$$

7. Finally, the lattice physics parameters $\vec{P}$ and k_∞ can be quickly estimated by the following summations:

$$\vec{P} \approx \left(\vec{P}_0 + \sum_{(m,n)\in\mathcal{Q}} \Delta\vec{P}^{M0}_{mn}\right) + \sum_{(k,l)\in\mathcal{P}} \left(\Delta\vec{P}^{S}_{kl} + \sum_{(m,n)\in\mathcal{Q}} \Delta\vec{P}^{Mi}_{mn} - \sum_{(m,n)\in\mathcal{Q}} \Delta\vec{P}^{M0}_{mn}\right) \tag{2.23}$$

$$k_\infty \approx \left(k_{\infty,0} + \sum_{(m,n)\in\mathcal{Q}} \Delta k^{M0}_{\infty,mn}\right) + \sum_{(k,l)\in\mathcal{P}} \left(\Delta k^{S}_{\infty,kl} + \sum_{(m,n)\in\mathcal{Q}} \Delta k^{Mi}_{\infty,kl} - \sum_{(m,n)\in\mathcal{Q}} \Delta k^{M0}_{\infty,kl}\right) \tag{2.24}$$

2.3.2 Combined Simplified Library

The combined library developed in the previous section can be simplified by assuming that the differences between the corresponding entities in $\mathcal{MP}_0$ and $\mathcal{MP}_i$ are small enough to be negligible. Under that assumption, all terms $\mathcal{MP}_i$ for i=1,NS, are not required, which reduces the overall library to:

$$\mathcal{CSP} = \mathcal{SP} \cup \mathcal{MP}_0 \tag{2.25}$$

Likewise, the summations required to estimate the attributes $\vec{P}$ and k_∞ inherit a simpler (though slightly less accurate) form:

$$\vec{P} \approx \vec{P}_0 + \sum_{(m,n)\in\mathcal{Q}} \triangle\vec{P}^{M0}_{mn} + \sum_{(k,l)\in\mathcal{P}} \triangle\vec{P}^{S}_{kl} \tag{2.26}$$

$$k_\infty \approx k_{\infty,0} + \sum_{(m,n)\in\mathcal{Q}} \triangle k^{M0}_{\infty,mn} + \sum_{(k,l)\in\mathcal{P}} \triangle k^{S}_{\infty,kl} \tag{2.27}$$

2.3.3 Examples

Consider the same hypothetical assembly in section 2.2.3. Figure 2.3 illustrates the combined library structure and the physics parameters estimation of the perturbed assembly, and combined simplified library is shown in Figure 2.4. In both figure, SP denotes the spatial perturbation library in Figure 2.1 and $MP0$ denotes the material perturbation library in Figure 2.2. MPi is the material perturbation library with respect to the each BSP in SP library. MPi shares the same structure with $MP0$ and they differ in pin distribution only.

2.4 Improvements of LSM

To improve the accuracy of LSM, the higher order terms in Taylor's series expansion should be considered. The number of second-order terms of material perturbations is far less than that of the spatial perturbations. For example, consider a typical one eighth assembly with 36 pin spatial positions and 4 fuel pin types without BP composition. For material perturbations of multi-level LSM, adding second-order terms into the library will increase the library size by as small as 10 CPM-2 or CASMO-3 calculations, or as few as 20 if two levels were adapted for each second-order term during interpolation because extrapolation will lead to a large error for a quadratic fit. For spatial perturbations, unfortunately, the large number of BSPs (108 in this case) makes the number of second-order terms as high as 5778. Thus, obviously, it is too computationally expensive to add these second-order terms into the spatial library. Even multiple shuffling

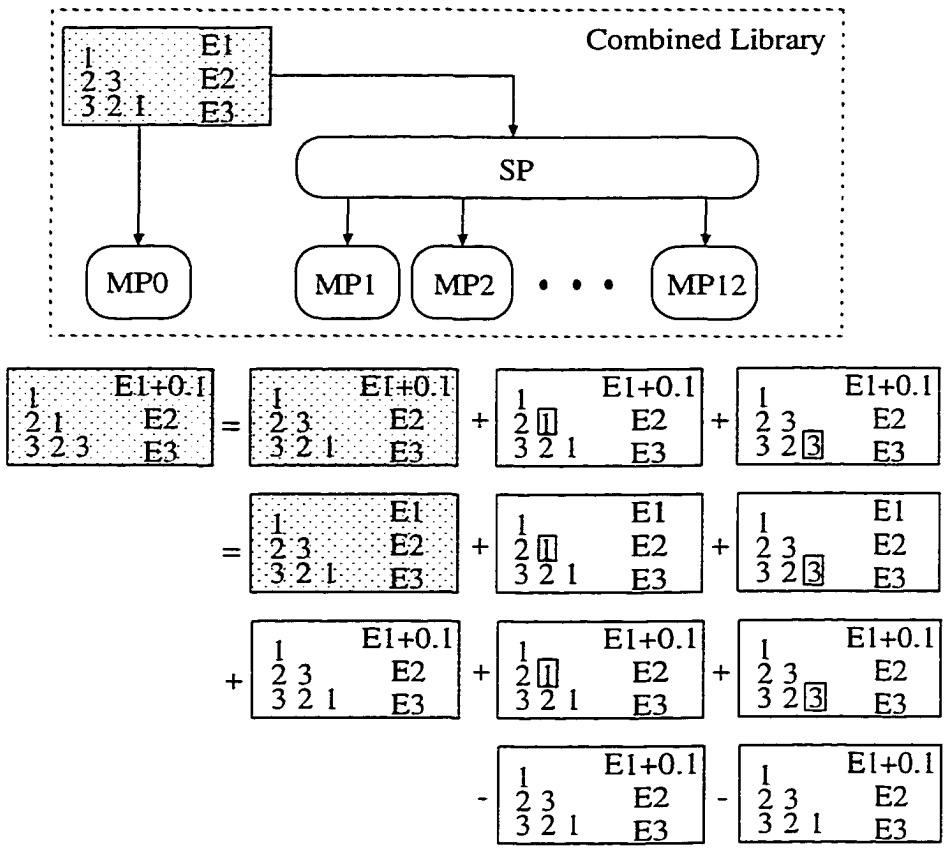


Figure 2.3 Example of combined library

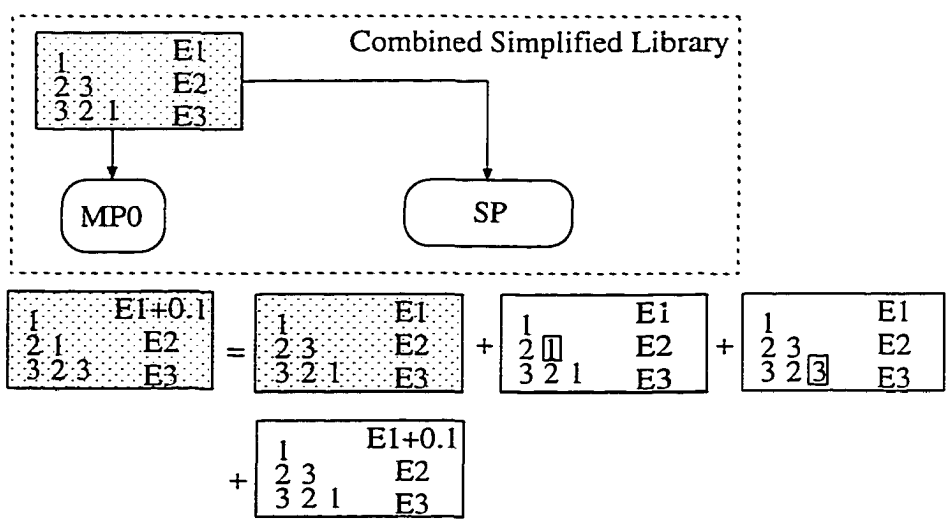


Figure 2.4 Example of combined simplified library

concerned, however, the perturbation size of spatial perturbation is usually smaller than that of material perturbation, because even single material perturbation will change the fuel enrichment of all pins of the involved pin type. Therefore, the errors of spatial perturbations are not as big as those of material perturbations in general. Consequently, the higher order models are studied only for material perturbations in this research.

2.4.1 LSM with Linear Interpolation

The multi-level LSM has its inherent weakness. Consider only one pin enrichment perturbed, the nature of multi-levels LSM is to apply first-order derivative at reference state to another state. For a quadratic fit, it is better than the single-level method, but the interpolation method will yield much better accuracy. In fact, the approach researched was to adjust the number of levels and the real value of each level to cover all the possible enrichment perturbation space of each pin type to avoid extrapolation. In this section, LSM with linear interpolation is developed. The procedures to create this library are as same as steps 1-3 in section 2.2.2, and the differences arise in the method employed to estimate the physics parameters of the perturbed assembly.

1. The material perturbation library $\mathcal{MP}$ is:

$$\mathcal{MP} = \left\{ \left(\vec{E}_0, \vec{P}_0, k_{\infty,0} \right), \left(\triangle \vec{E}_{jm}, \triangle \vec{P}_{jm}, \triangle k_{\infty,jm} \right) \right\}; j \in \mathcal{M}, m \in \mathcal{NE}_j \qquad (2.28)$$

   where the total number of levels, NE, and the enrichment (BA concentration) at each level, $\triangle \vec{E}_{jm}$ are adjusted to meet the requirements of interpolation.

2. A perturbed assembly's material distribution can be decomposed into:

$$\vec{E} = \vec{E}_0 + \sum_{j \in \mathcal{M}} \triangle \vec{E}_j, \mathcal{M} = \{1, 2, \ldots, M\} \qquad (2.29)$$

3. $\vec{P}$ and k_∞ of the perturbed assembly can be estimated by the following summations:

$$\vec{P} \approx \vec{P}_0 + \sum_{j \in \mathcal{M}} \triangle \vec{P}_j \qquad (2.30)$$

$$k_\infty \approx k_{\infty,0} + \sum_{j \in \mathcal{M}} \triangle k_{\infty,j} \tag{2.31}$$

where,

$$\triangle \vec{P}_j = \triangle \vec{P}_{j(l-1)} + \frac{\left|\triangle \vec{E}_j - \triangle \vec{E}_{j(l-1)}\right|}{\left|\triangle \vec{E}_{jl} - \triangle \vec{E}_{j(l-1)}\right|} \left(\triangle \vec{P}_{jl} - \triangle \vec{P}_{j(l-1)}\right) \tag{2.32}$$

$$\triangle k_{\infty,j} = \triangle k_{\infty,j(l-1)} + \frac{\left|\triangle \vec{E}_j - \triangle \vec{E}_{j(l-1)}\right|}{\left|\triangle \vec{E}_{jl} - \triangle \vec{E}_{j(l-1)}\right|} \left(\triangle k_{\infty,jl} - \triangle k_{\infty,j(l-1)}\right) \tag{2.33}$$

if $\vec{E}_j$ falls between $\vec{E}_{j(l-1)}$ and $\vec{E}_{jl}$.

2.4.2 LSM with Second-order Interpolation

For further accuracy improvement, second-order interpolation can be exployed instead of the first-order interpolation in the previous section. To implemented LSM with second-order interpolation, replace Eq. 2.32 and Eq. 2.33 with the following two equations:

$$\triangle \vec{P}_j = \left|\triangle \vec{E}_j\right|^2 \vec{a}_{p,j} + \left|\triangle \vec{E}_j\right| \vec{b}_{p,j} + \vec{c}_{p,j} \tag{2.34}$$

$$\triangle k_{\infty,j} = \left|\triangle \vec{E}_j\right|^2 a_{k_\infty,j} + \left|\triangle \vec{E}_j\right| b_{k_\infty,j} + c_{k_\infty,j} \tag{2.35}$$

where the coefficient vectors $\vec{a}_{p,j}$, $\vec{b}_{p,j}$, $\vec{c}_{p,j}$ and $a_{k_\infty,j}$, $b_{k_\infty,j}$, $c_{k_\infty,j}$ are determined by:

$$\begin{bmatrix} \vec{a}'_{p,j} \\ \vec{b}'_{p,j} \\ \vec{c}'_{p,j} \end{bmatrix} = \begin{bmatrix} \left|\triangle \vec{E}_{j(l-1)}\right|^2 & \left|\triangle \vec{E}_{j(l-1)}\right| & 1 \\ \left|\triangle \vec{E}_{jl}\right|^2 & \left|\triangle \vec{E}_{jl}\right| & 1 \\ \left|\triangle \vec{E}_{j(l+1)}\right|^2 & \left|\triangle \vec{E}_{j(l+1)}\right| & 1 \end{bmatrix}^{-1} \begin{bmatrix} \triangle \vec{P}'_{j(l-1)} \\ \triangle \vec{P}'_{jl} \\ \triangle \vec{P}'_{j(l+1)} \end{bmatrix} \tag{2.36}$$

$$\begin{bmatrix} a_{k_\infty,j} \\ b_{k_\infty,j} \\ c_{k_\infty,j} \end{bmatrix} = \begin{bmatrix} \left|\triangle \vec{E}_{j(l-1)}\right|^2 & \left|\triangle \vec{E}_{j(l-1)}\right| & 1 \\ \left|\triangle \vec{E}_{jl}\right|^2 & \left|\triangle \vec{E}_{jl}\right| & 1 \\ \left|\triangle \vec{E}_{j(l+1)}\right|^2 & \left|\triangle \vec{E}_{j(l+1)}\right| & 1 \end{bmatrix}^{-1} \begin{bmatrix} \triangle k_{\infty,j(l-1)} \\ \triangle k_{\infty,jl} \\ \triangle k_{\infty,j(l+1)} \end{bmatrix} \tag{2.37}$$

when $\triangle \vec{E}_j$ falls between $\frac{1}{2}\left(\triangle \vec{E}_{j(l-1)} + \triangle \vec{E}_{jl}\right)$ and $\frac{1}{2}\left(\triangle \vec{E}_{jl} + \triangle \vec{E}_{j(l+1)}\right)$.

2.4.3 Second-order Cross-term Compensation

All the models, multi-level LSM, LSM with linear or second-order interpolation, improve their accuracy by refining the algorithm for material perturbations involved with only one pin type. Consider the Taylor's series expansion after truncating off the third and higher terms:

$$\vec{P} \approx \vec{P}^* = \vec{P}_0 + \left[\frac{\partial \vec{f}}{\partial \vec{E}}\right]_0 \cdot \triangle \vec{E} + \frac{1}{2} \begin{bmatrix} \triangle \vec{E}' \cdot \left[\frac{\partial^2 f_1}{\partial \vec{E}^2}\right]_0 \cdot \triangle \vec{E} \\ \cdot \\ \cdot \\ \cdot \\ \triangle \vec{E}' \cdot \left[\frac{\partial^2 f_N}{\partial \vec{E}^2}\right]_0 \cdot \triangle \vec{E} \end{bmatrix} \tag{2.38}$$

$$k_\infty \approx k_\infty^* = k_{\infty,0} + \left[\frac{\partial g}{\partial \vec{E}}\right]_0 \cdot \triangle \vec{E} + \frac{1}{2} \triangle \vec{E}' \cdot \left[\frac{\partial^2 g}{\partial \vec{E}^2}\right]_0 \cdot \triangle \vec{E} \tag{2.39}$$

in which the second-order terms include not only the second-order derivatives of each pin type but also the cross terms presenting the interactive effect between any two pin types. The former terms have been considered, to different extent, by the multi-level LSM, and LSM with interpolation. Thus, the errors, not including the errors introduced in computation (round-off error), are mainly due to the second-order cross terms. When more than two pin types are perturbed simultaneously, the second-order cross terms should be considered to improve the accuracy.

Based on the procedures used to create the material library of LSM in Section 2.2.2. The following steps are needed to for the second-order cross-term compensation.

1. Add the following dual material perturbations into the basic material perturbations (BMPs):

$$\triangle \left(\vec{E}_{jn}, \vec{E}_{kn}\right); j, k \in \mathcal{M}, j \neq k, n \in \mathcal{NE}2 \tag{2.40}$$

where $\left(\vec{E}_{jn}, \vec{E}_{kn}\right)$ represents a dual material perturbation involving the jth and kth pin types, where the fuel enrichment (or burnable absorber compositions) of

pin type j and k are changed to a new level n at the same time. $\mathcal{NE}2$ is the set of natural integer number from 1 to $NE2$, where $NE2$ is the total number of pre-defined (via code input) fuel enrichment (or BA concentration) levels for dual perturbations.

2. For each new dual BMP, $\left(\vec{E}_{jn}, \vec{E}_{kn}\right)$, calculate second-order cross derivative. say. $\triangle\vec{P}_{jkn}$ and $\triangle k_{\infty,jkn}$.

$$\triangle\vec{P}_{jkn} = \frac{\vec{P}_{jkn} - \vec{P}_{jn} - \vec{P}_{kn} + \vec{P}_0}{\triangle\vec{E}_{jn}\triangle\vec{E}_{kn}} \tag{2.41}$$

$$\triangle k_{\infty,jkn} = \frac{k_{\infty,jkn} - k_{\infty,jn} - k_{\infty,kn} + k_{\infty,0}}{\triangle\vec{E}_{jn}\triangle\vec{E}_{kn}} \tag{2.42}$$

where, $\vec{P}_{jkn}$ and $k_{\infty,jkn}$ are the lattice parameters of the dual perturbation, while $\vec{P}_{jn}$ and $k_{\infty,jn}$, $\vec{P}_{kn}$ and $k_{\infty,kn}$ are those of the single perturbations. and $\vec{P}_0$ and $k_{\infty,0}$ are those of the reference assembly, as noted earlier.

3. Then, add the following entries into the Material Perturbation Library $\mathcal{MP}$:

$$\left(\triangle\vec{E}_{jkn}, \triangle\vec{P}_{jkn}, \triangle k_{\infty,jkn}\right), j,k \in \mathcal{M}, n \in \mathcal{NE}2 \tag{2.43}$$

4. The second-order cross compensation terms, $\triangle\vec{P}_{2c}$ and $\triangle k_{\infty,2c}$ are:

$$\triangle\vec{P}_{2c} = \sum_{(i,j)} \triangle\vec{P}_{ij\hat{n}}\triangle\vec{E}_i\triangle\vec{E}_j \tag{2.44}$$

$$\triangle k_{\infty,2c} = \sum_{(i,j)} \triangle k_{\infty,ij\hat{n}}\triangle\vec{E}_i\triangle\vec{E}_j \tag{2.45}$$

where $i,j \in$ {all the perturbed pin type numbers}, $i \neq j$ and $\hat{n}$ denotes that $\triangle\vec{P}_{ij\hat{n}}$ is linearly interpolated from $\triangle\vec{P}_{ij1}$ to $\triangle\vec{P}_{ijNE2}$.

3 ERROR ANALYSIS OF LSM

3.1 Purpose of Error Analysis

In general, the errors generated by the LSM relative to direct (exact) evaluations using CPM-2 or CASMO-3 should be small enough to not misdirect the optimization process (i.e., retain optimization fidelity). It is obvious that the smaller the errors, the better the optimization fidelity. This implies that extensive investigation of errors under a wide range of circumstances (fuel assembly types and loadings) must be performed. In fact, if necessary, this could lead to developing any additional necessary steps required to control any undesirable levels of error. So far, however, the results show that the current levels of error appear to be sufficiently adequate.

The ideas behind a thorough purposes of error analysis of the LSM are:

- To identify the error sources. The LSM errors are composed of two distinct parts, a *truncation* error and *round-off* error. The fraction of each error type relative to the total error will guide the direction of LSM accuracy improvements;
- To improve the LSM accuracy. In fact, the multi-level LSM, LSM with interpolation and second-order cross-term compensation are the outcome of the error analysis.
- To evaluate different libraries. The separated libraries (*i.e.* spatial and material library), combined library and combined simplified library will be contrasted. The comparison will benefit their proper application to the FORMOSA-L code.

3.2 Approach to Error Analysis

The errors of the LSM are defined as the differences between pin power and k_∞ obtained via LSM and those values obtained using CPM-2 or CASMO-3 for a specified fuel assembly loading and pin arrangement. Thus, the error can be expressed as:

$$\vec{PE} = \vec{P}^{LSM} - \vec{P}^{Exact} \tag{3.1}$$

$$KE = k_\infty^{LSM} - k_\infty^{Exact} \tag{3.2}$$

where $\vec{PE}$ and KE define the errors in pin power and k_∞, respectively. $\vec{P}$ and k_∞ are the pin power and infinite multiplication factor. The superscripts "LSM" and "Exact" indicate the values obtained by LSM or by direct (exact) evaluations. Also other types of error functions can be defined based on the above error definition, such as maximum pin power error (MAXPE), relative mean square (RMS) of pin power error (RMSPE), maximum k_∞ error (MAXKE) and RMS of k_∞ error (RMSKE).

$$MAXPE = \max(|PE_{ij}|) \tag{3.3}$$

$$RMSPE = \sqrt{\frac{\sum_{ij} PE_{ij}^2}{N \times M}}, \text{ for all positions "i" and all burnup steps "j"} \tag{3.4}$$

$$MAXKE = \max(|KE_j|) \tag{3.5}$$

$$RMSKE = \sqrt{\frac{\sum_j KE_j^2}{M}}, \text{ for all burnup steps "j"} \tag{3.6}$$

In addition, because of the great importance of power peaking within the optimization process in the FORMOSA-L code, the difference between the power peaking obtained by LSM versus that obtained with CPM-2 or CASMO-3 is carefully analyzed. The study provides statistical evidence that the power peaking error is always less than or equal to MAXPE, even though the pin power error can mis-predict the power peaking to a different burnup step and/or to a different position.

The LSM errors are composed of two distinct parts, a *truncation* error and *round-off* error. The truncation error arises from eliminating the higher order terms of the

Taylor's series used to derive the LSM method. On the other hand, the round-off error is due to the fact that the pin power and k_∞ values available within the standard CPM-2 output file have only 3 and 6 significant digits, respectively. The corresponding significant digits for CASMO-3 output are 3 and 5, respectively. It should be noted that the round-off error is a byproduct of the non-intrusive coupling between FORMOSA-L and the lattice physics codes which is done purely via input/output interfacing thus, the coupling to FORMOSA-L is done purely via input/output files, not directly via data arrays or variables, thus the noted limitation. This round-off error is introduced and accumulated when LSM is used in each SA history to estimate the lattice parameters. As mentioned in Section 3.1, error analyses are needed to determine the fraction of each type of error relative to the total error, the effect of perturbation size on accuracy, and accuracy comparison among different LSM libraries. Any LSM enhancements needed (*e.g.* interpolation and second-order cross-term compensation) are clearly driven by the outcome of error analyses.

To identify the error sources, an approach is needed to separate the different errors. In the case of CPM-2, due to the availability of its source code, a modified version called CPM-2M was modified to output 6 and 8 decimal digits for pin power and k_∞, respectively. This facilitated the estimation of the fraction of each type of error, as the round-off error is basically eliminated with CPM-2M due to adding more digits. The errors obtained by CPM-2M are the truncation error, and those of the original CPM-2 include both truncation error and round-off error. In this manner, the different types of errors are separated and identified. It is worthy to note that the iteration stopping criteria of the CPM-2 code was not changed, so the time requirement of each running of the CPM-2M code was not affected. Also, what affected on the round-off error are the significant digits of pin power and k_∞ values in the output file, not the convergence criteria. Figure 3.1 illustrates the proposed error fraction analysis for FORMOSA-L.

Currently, the $\vec{PE}$ and KE of the optimal assembly obtained by FORMOSA-L

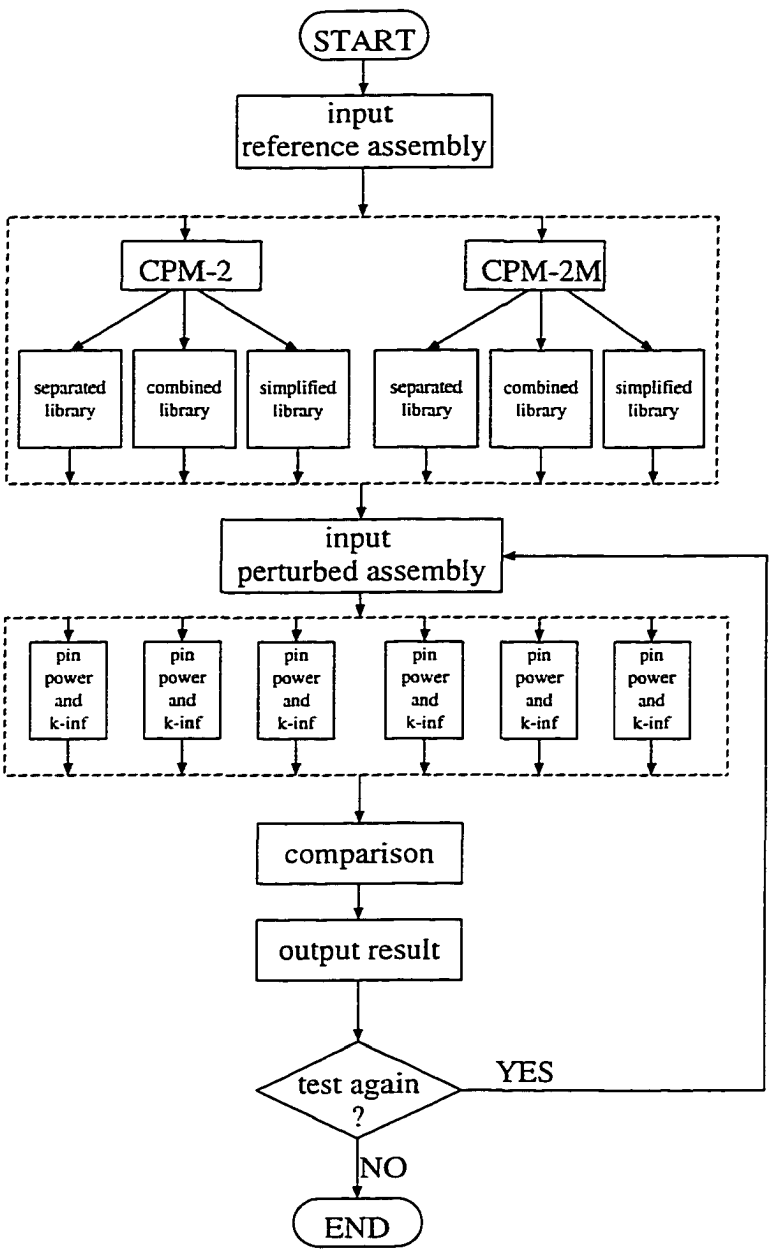


Figure 3.1 Error analysis mode for FORMOSA-L

are calculated and available as code output at the end of each cooling cycle. An option to compute the error functions MAXPE, RMSPE, MAXKE, and RMSKE within FORMOSA-L is a very important option that facilitates the study of the LSM performance, even though this option requires extremely large computational times due to the fact that the "exact" pin power and k_∞ needs to be obtained via direct CPM-2/CASMO-3 evaluations each SA history. Another important aspect of the error analysis was to create a version of the LSM whereby only one type of LSM library is required for specific perturbations the user requests, then the user can choose an arbitrary perturbed assembly as code input to do error analysis. This facilitates the generation of LSM results fairly quickly for manual (and arbitrary) perturbations of assembly loadings to examine the performance of the LSM. Random sampling with a certain sample size is provided as an alternative to manual input of perturbations.

3.3 Identification of Error Sources

Consider an octant-symmetric 15×15 PWR assembly with 5 types of pins. Pin types 1 through 3 are UO_2 rods, and pin type 4 is a water hole, while pin type 7 corresponds to B_4C discrete burnable absorbers (BAs). It is shown in Figure 3.2.

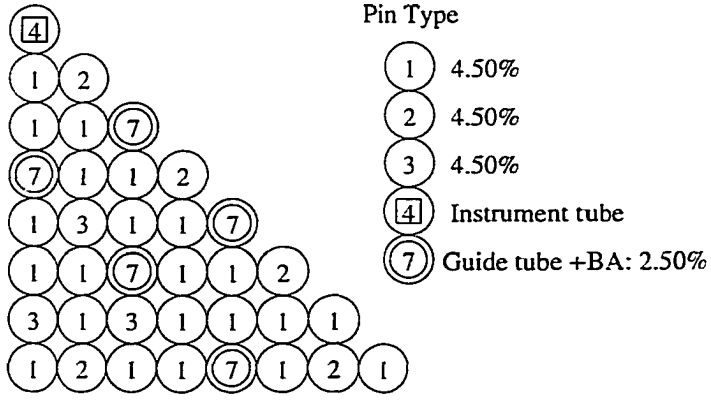


Figure 3.2 PWR reference assembly

The perturbed assembly is generated by freezing the pin distribution, fuel enrichment of pin types 2 and 3, BA composition of pin type 7 and only changing the enrichment of pin type 1 from 4.5wt% to 3.005Wt%, which is clearly a very large perturbation. The physics parameters, *i.e.* pin power and infinite multiplication factor, are calculated only at BOL by using multi-level LSM with different levels in the creation of the material library, respectively. Different levels assure that the number of summation terms in Equation 2.10 (or 2.11) are different and so the round-off error can be identified. The errors yielded by CPM-2 and those by CPM-2M are compared. Figure 3.3 shows the results of MAXPE and MAXKE.

From Figure 3.3, it is observed that the difference between MAXPE by CPM-2 and MAXPE by CPM-2M is closely proportional to the number of summation terms. It is identified as round-off error, because its proportionality to the number of summation terms is the main feature of round-off error. Also it is found that the round-off error is significant for pin power although negligible for k_∞. This is true because k_∞ values in the CPM-2 code output have more decimal digits than power values and thus have a smaller round-off error. Therefore, to improve accuracy of LSM, both truncation error and round-off error need to be addressed. In fact, the truncation error can be reduced by employing new methods including interpolation and second-order cross-term compensation, and the round-off error via limiting the number of summation terms.

3.4 Error Analysis: Material Library Case

3.4.1 Single Pin Perturbations

Consider the same reference assembly as that used in Section 3.3. In this case, however, only the enrichment of pin type 1 can be perturbed while the pin distribution and material composition of pin types 2, 3 and 7 are kept fixed. This kind of perturbations are called single pin perturbations. The perturbed assemblies are generated by decreas-

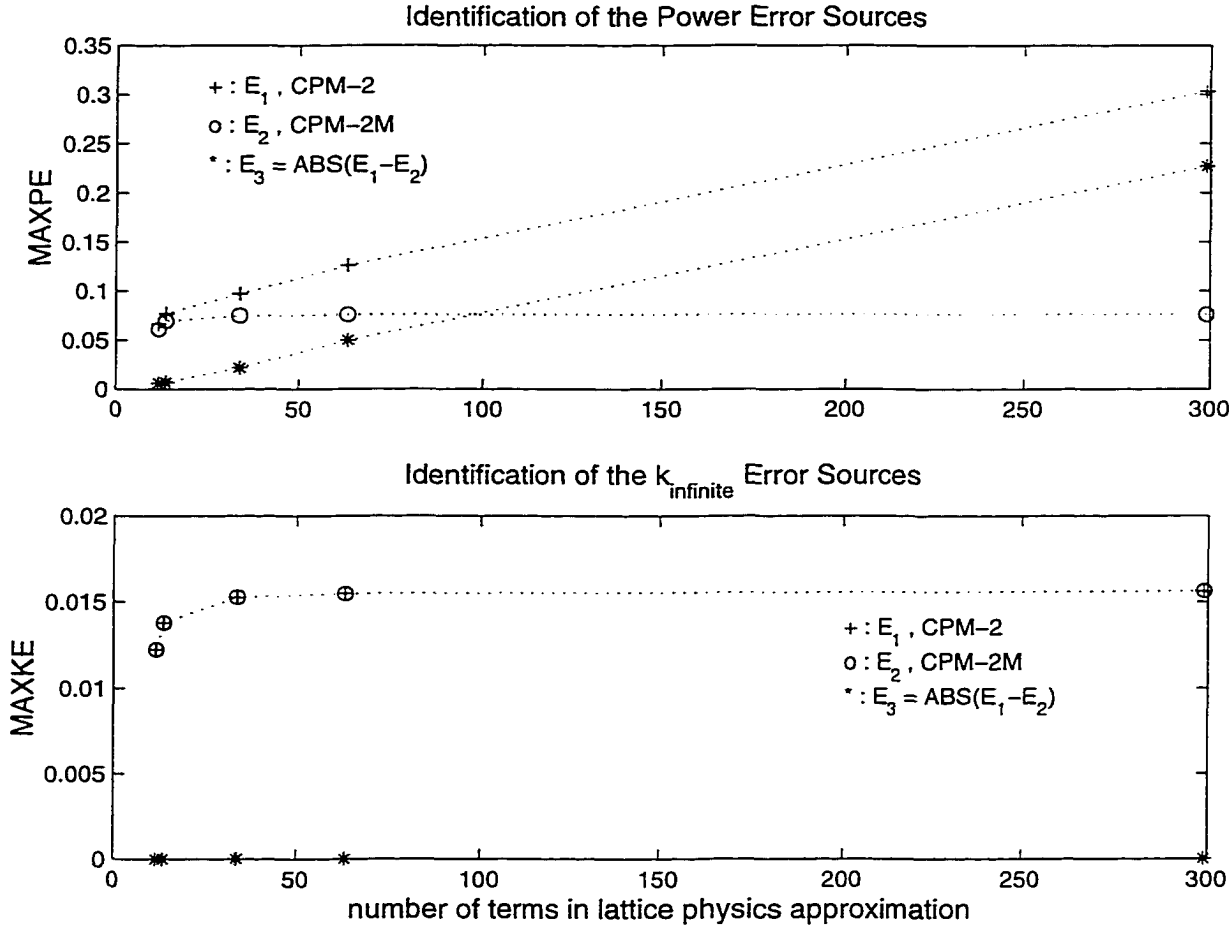


Figure 3.3 Identification of pin power and k_∞ error sources

ing the enrichment of pin type 1 from 4.5wt% (E_0) to 3.0wt% at a constant negative step size of -0.05wt% (DE=0.05wt%). In the material library, seven levels are used, *i.e.* 4.45wt%, 4.25wt%, 4.00wt%, 3.75wt%, 3.50wt%, 3.25wt% and 3.00wt%, which correspond to the data points 1, 5, 10, 15, 20, 25 and 30 on the X-axis of Figure 3.4. The multi-level LSM, LSM with linear interpolation and LSM with second interpolation are compared and the results are shown in Figure 3.4

The upper figure in Figure 3.4 shows that the power truncation error of multi-level LSM is the largest and that of LSM with linear interpolation is significantly better, while the LSM with second-order interpolation reduces the power truncation error a little further. The lower figure in Figure 3.4 shows the total power error behavior.

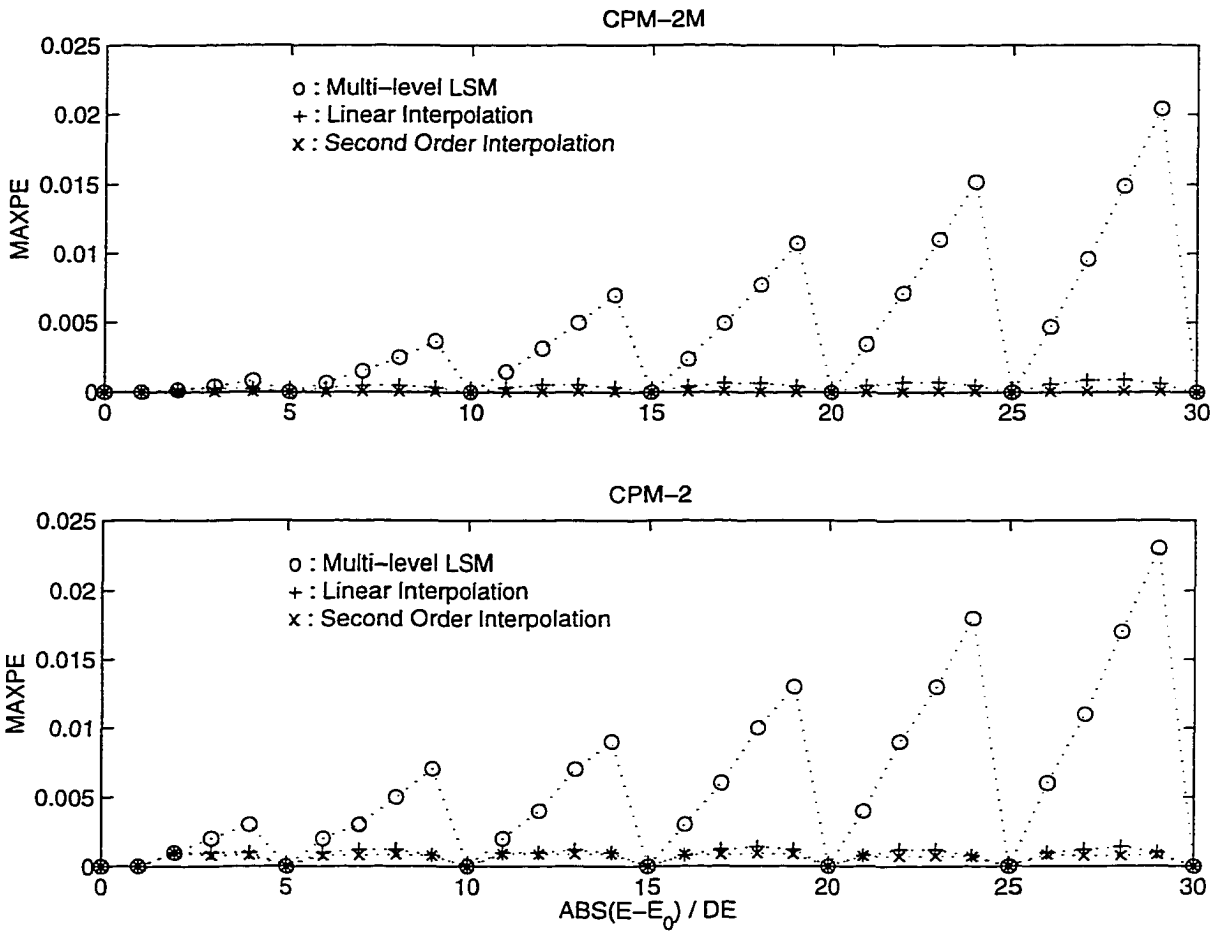


Figure 3.4 Power error behavior comparison of different methods

Comparing these two figures, it is found that the power round-off error is kept small because interpolation methods limit the number of summation terms to a small value. The total k_∞ errors shown in Figure 3.5 follow the same behavior as the power truncation error, because for k_∞ the fraction of round-off error is very small. Considering the different round-off error fractions of power and k_∞, the LSM with linear interpolation is quite adequate for power approximation, while the LSM with second-order interpolation is better for the k_∞ approximation.

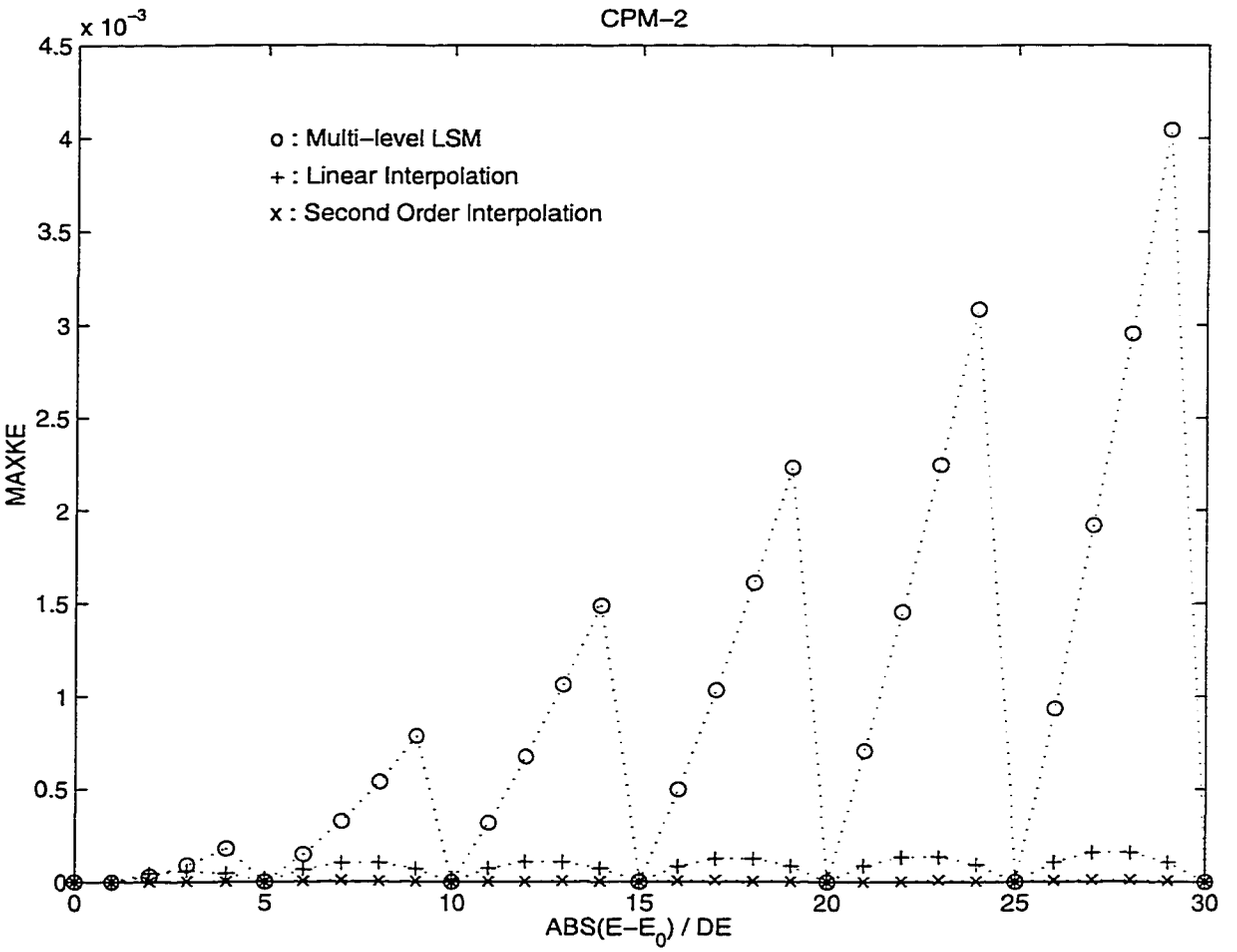


Figure 3.5 k_∞ error behavior comparison of different methods

3.4.2 Binary Pin Perturbations

Binary pin perturbations imply that the perturbation produced with simultaneous enrichment changes of any two pin types. Consider the PWR assembly in Section 3.3 as the reference assembly and allow the enrichment of pin types 1 and 2 to change simultaneously. Both the enrichment perturbation ranges of these two pin types are from 5.0wt% to 3.0wt% with a step size of DE=0.05wt% based on the reference enrichment of 4.5wt%. The library contains ±1.0, ±5.0, ±10.0, -15.0, -20.0, -25.0 and -30.0 levels (×DE) from 4.5wt%. In this test case, we keep all the possible enrichment perturbations of each pin type (1 or 2) at these points, whose error due to single perturbation are always zero. In this way, the total errors of binary perturbations only contain the interactive effect of the two single perturbations, which is our most concerned in this section. The second-order cross terms compensation is used to reduce the error. The second-order cross derivatives are calculated only in two levels, and the CPM-2M code is used in this test. Figure 3.6 shows that the errors of binary perturbations are reduced via second-order cross-term compensation.

3.4.3 Multiple Pin Perturbations

Consider the PWR assembly in Section 3.3 as the reference assembly. All the enrichment of pin type 1, 2 and 3 are allowed to be perturbed within the range from 5.0wt% to 3.0wt% with a step size of 0.05wt%, while the B_4C composition of pin type 7 also can be changed within 3.5wt% to 1.5wt% with a larger step size of 0.1wt%. The enrichment levels in library for pin type 1, 2 and 3 are the same as those in the previous section. The BA composition levels are ±1.0, ±5.0 and ±10.0. The perturbed assemblies are randomly generated by calling the material perturbation routines in FORMOSA-L code. In this case we randomly sample 1000 assemblies and use CPM-2 code. The LSM methods are LSM with linear interpolation for pin power approximations and LSM with second-order

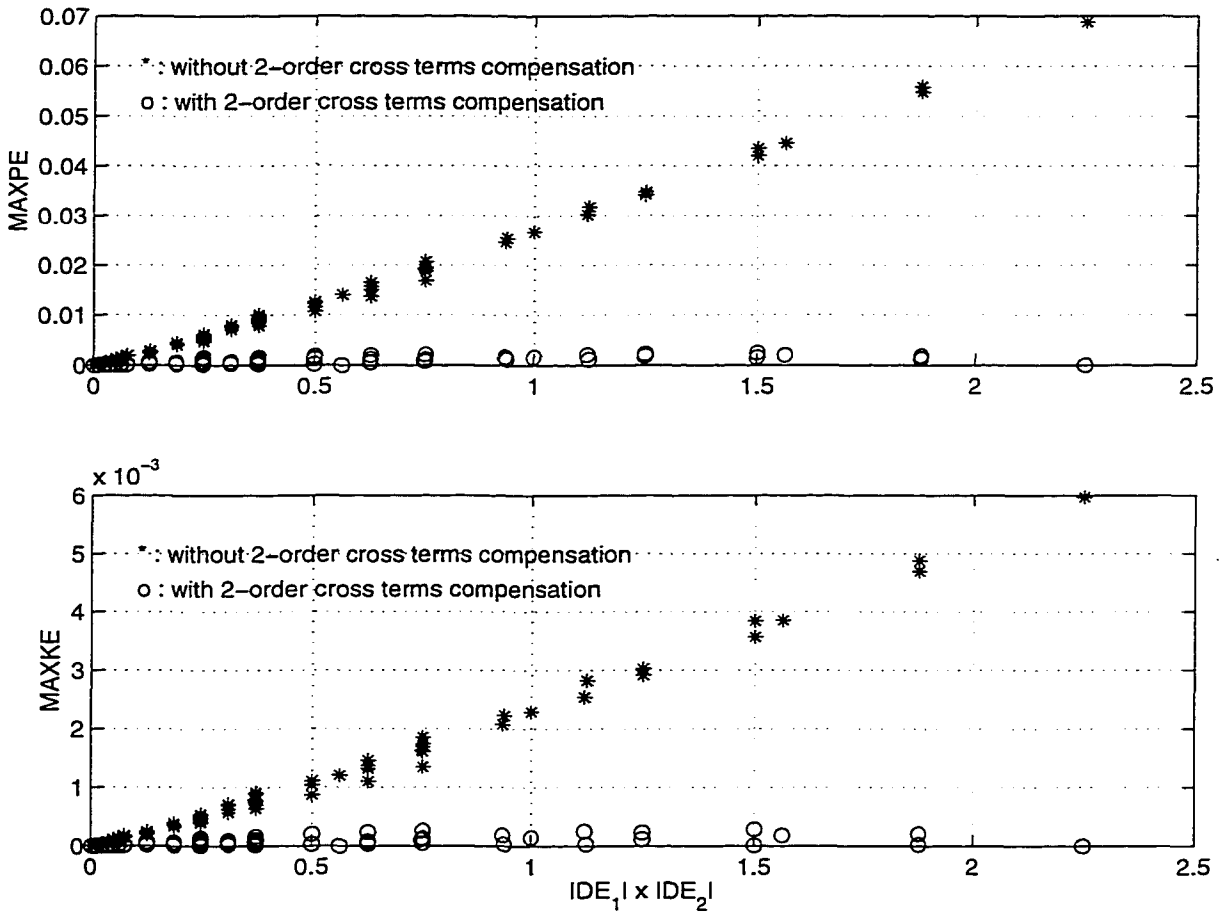


Figure 3.6 Error reduction via second-order cross-term compensation

interpolation for k_∞ approximations, and second-order cross terms are compensated for both pin power and k_∞.

The results are shown in Figure 3.7. The maximum power errors (MAXPE) are less than 0.004 and the maximum k_∞ errors (MAXKE) are less than 0.0002 (*i.e.* 20 pcm). Also, the errors of power peaking are under 0.004, while most of them are under 0.002. The accuracy of LSM is improved by the interpolation approach and second-order cross terms compensation.

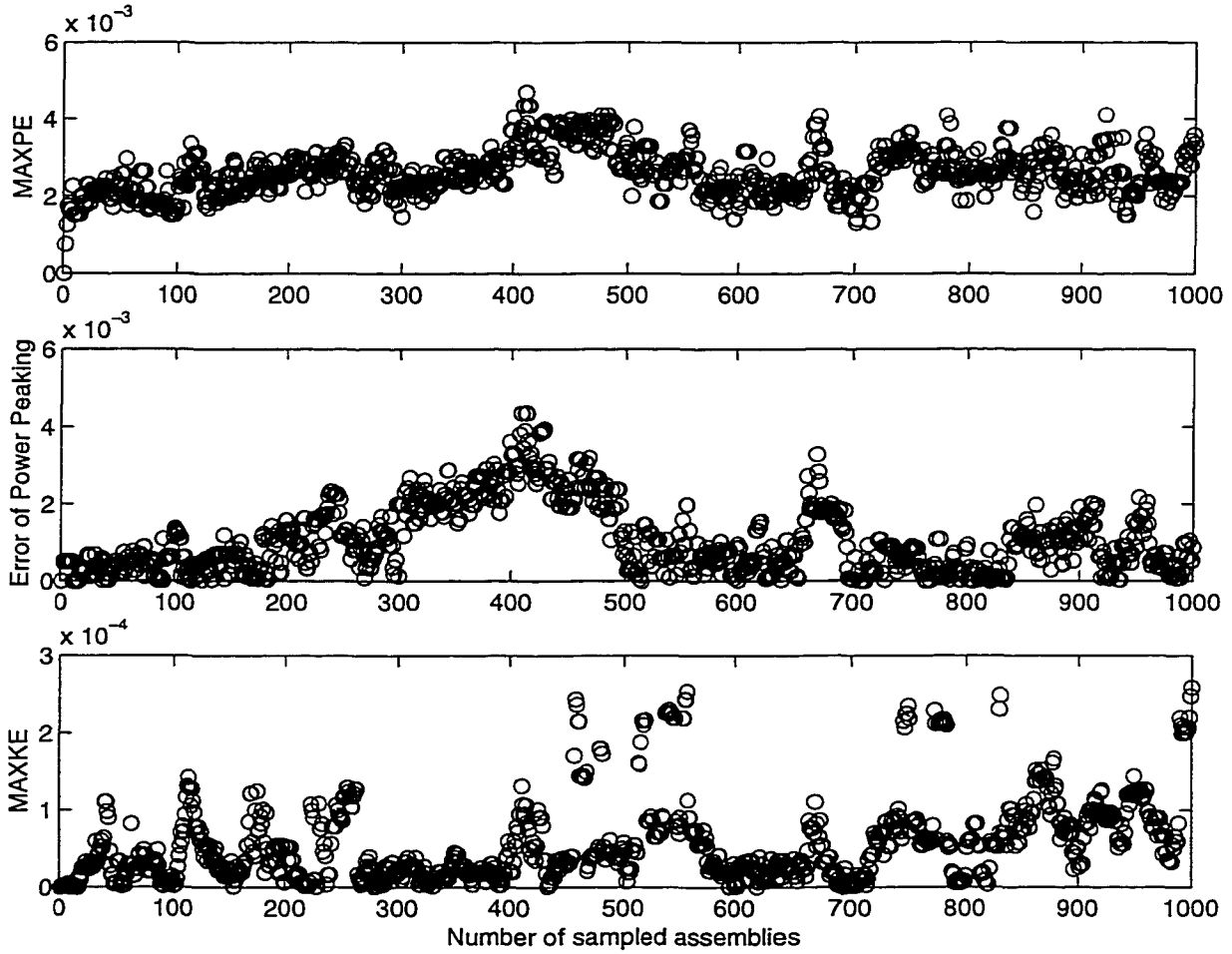


Figure 3.7 Error of multiple material perturbations using CPM-2

3.5 Error Analysis: Spatial Library Case

3.5.1 Binary Pin Perturbations

As mentioned in Section 2.3, the accuracy improvement for spatial perturbation is computationally expensive and the error level is lower than that of material perturbations (if no accuracy improvement is considered). In this section, the error behavior of spatial perturbations is studied. Consider the same PWR assembly and the enrichment of pin type 1, 2 and 3 are 5.0wt%, 4.0wt% and 3.0wt%, while the BA composition of pin type 7 is 2.5wt%. All the material compositions are fixed and only spatial shuffling are allowed. The water hole (pin type 4) and the discrete BA pins (pin type 7) are frozen into their initial positions. Binary spatial perturbations mean only one shuffling involved two different pins. In this test, we only shuffle one pin of pin type 1 with another pin of pin type 2, or one pin 1 and one pin 3.

In Figure 3.8, the distance between the two shuffled pins (D) is measured with the unit of pin cell lattice pitch. The errors of binary spatial perturbations decrease with the increase of the distance between the two shuffled pins, because the bigger the distance, the more independent they are and the smaller the cross error. Also, the errors increase with the the enrichment difference between the two shuffled pins.

3.5.2 Multiple Pin Perturbations

Multiple spatial perturbations are generated by randomly shuffling the assembly through calling the associated subroutines in FORMOSA-L code. One thousand perturbed assemblies are sampled and the reference assembly is as same as that described in the previous section.

The results in Figure 3.9 show that the round-off errors are not significant and the truncation error is dominant for the spatial perturbations. The MAXPE of most assemblies are under 0.02 and a few of them are as high as 0.03. The errors of power

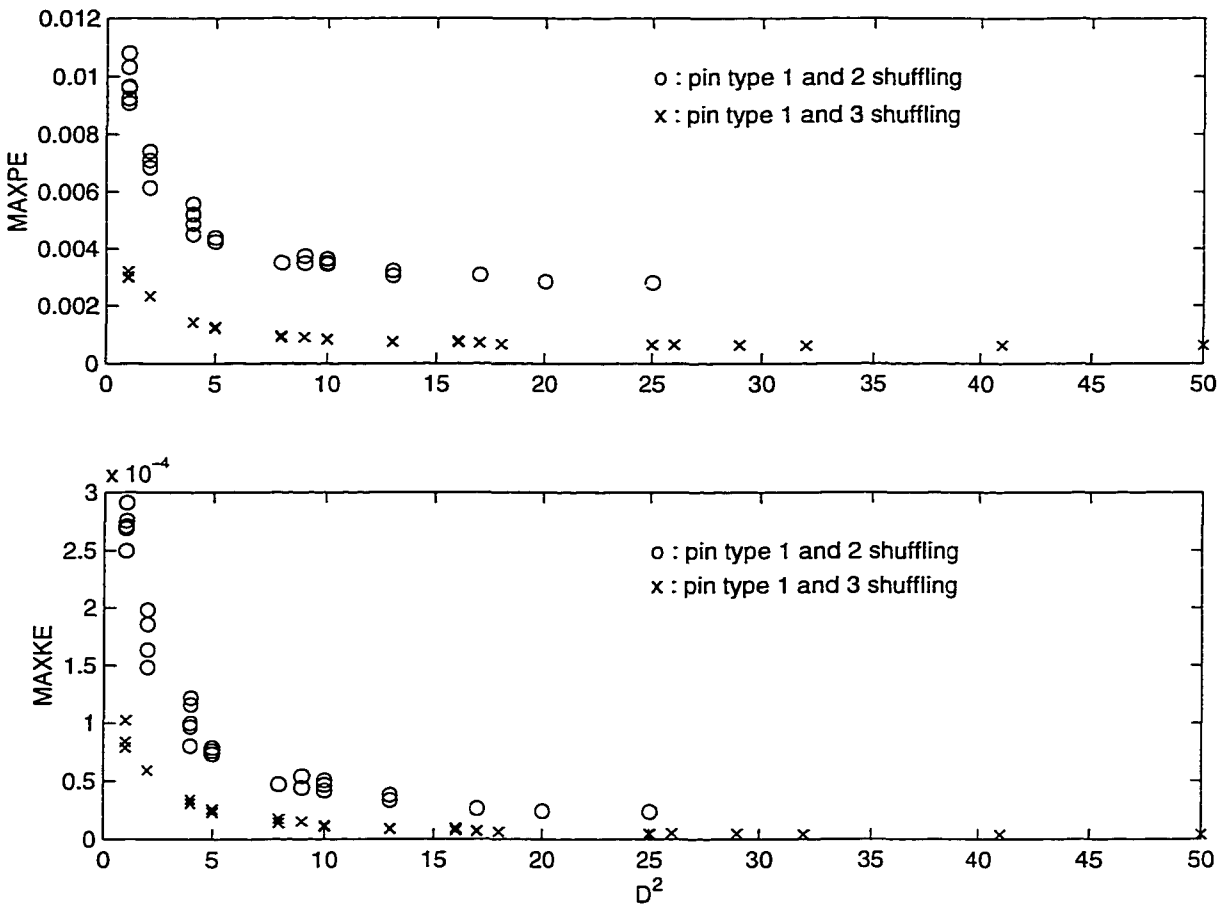


Figure 3.8 Error of binary pin spatial perturbations using CPM-2

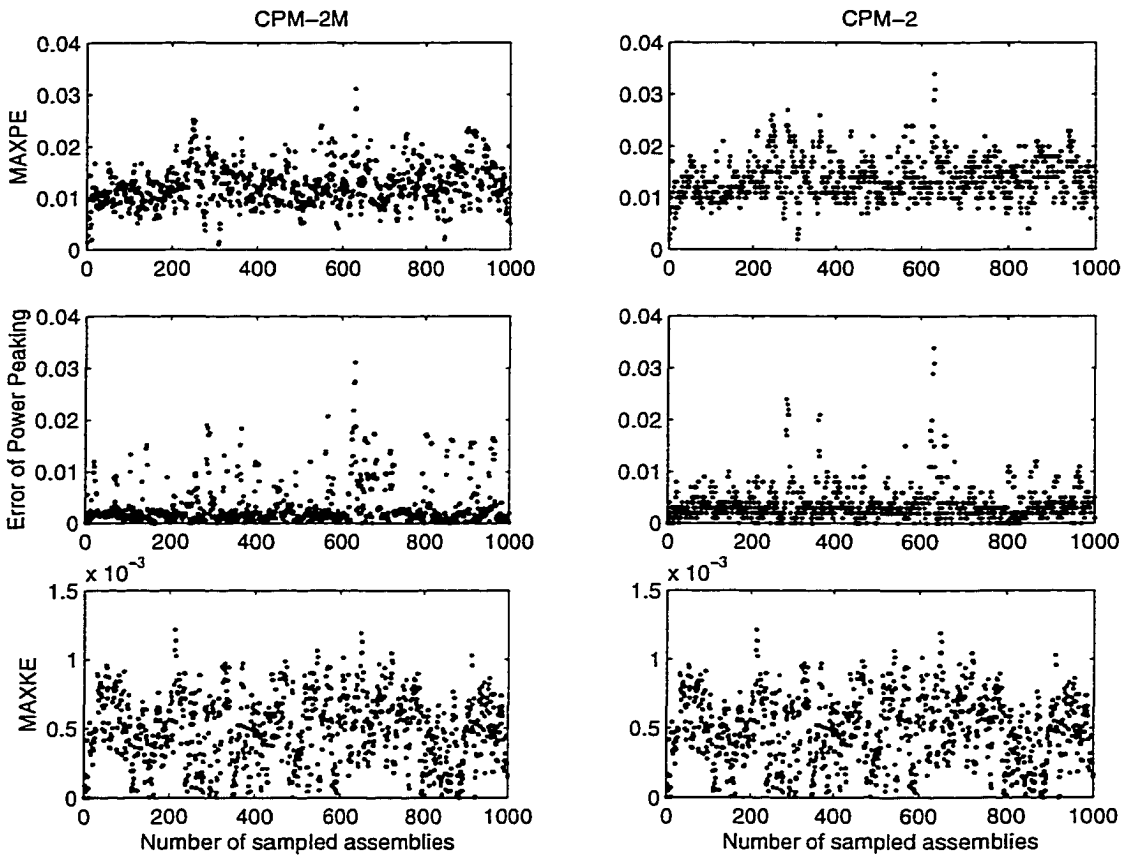


Figure 3.9 Error of multiple pin spatial perturbations using CPM-2

peaking (PPE) of most assemblies are under 0.01 and MAXKE are under 0.001 (*i.e.* 100 pcm).

3.6 Comparison of Different Libraries

The error behavior of the combined library and the combined simplified library are studied in this section. The reference assembly is similar to that used before. The initial fuel enrichment of pin type 1, 2 and 3 are 4.5wt%, 4.0wt% and 3.5wt%, respectively, and the allowed perturbation ranges are [4.0, 5.0], [3.5, 4.5] and [3.0, 4.0], respectively, with the same step size of 0.05wt%. Similarly, the BA composition of pin type 7 is 2.5wt% and it ranged within [2.0, 3.0] with 0.05wt% minimum allowed perturbation size. The

LSM with linear interpolation is employed for power estimations while the second-order interpolation for k_∞ estimations. The second-order cross term compensation is used only once (in MP0, not in MPi). For each example, one thousand candidate assemblies were randomly generated.

The results are shown in Table 3.1. In each entry of this table, the figure on top is the maximum error among the one thousand assemblies, and the figure inside the parentheses corresponds to the error level under which 90% of the assemblies are. First, through the comparisons of the results obtained by CPM-2 and CPM-2M, it can be observed that the round-off errors are significant for power, but not for k_∞. Second, the combined library leads to higher accuracy while the combined simplified library results in the highest error level, much too high to be acceptable.

Table 3.1 Error comparisons of $\mathcal{CP}$ library and $\mathcal{CSP}$ library

	combined library		combined simplified library	
	CPM-2M	CPM-2	CPM-2M	CPM-2
MAXPE	0.023 (0.010)	0.051 (0.033)	0.247 (0.172)	0.219 (0.169)
RMSPE	0.005 (0.003)	0.017 (0.013)	0.122 (0.074)	0.118 (0.073)
MAXKE	0.00112 (0.00035)	0.00085 (0.00042)	0.00148 (0.00074)	0.00134 (0.00085)
PPE	0.011 (0.003)	0.034 (0.020)	0.184 (0.084)	0.149 (0.081)

3.7 Error Analysis of LSM Using CASMO-3

The FORMOSA-L code provides two options for selecting the lattice-physics evaluators, either CPM-2 or CASMO-3. The former sections of this chapter only deal with the error analysis of LSM using CPM-2. The error analysis of LSM using CASMO-3 are studied in this section. As far as LSM is concerned, the CASMO-3 code differs with CPM-2 code in two ways. As mentioned before, the first obvious difference is the format

of their output files, For instance, in CASMO-3 only five significant figures are provided for k_∞ rather than the six figures in CPM-2. This purely I/O related difference leads to larger round-off errors in k_∞ estimations. The second difference is related to the underlying methodology used to solve the physical problem, which yields two different functionalities. These noted differences result in the different truncation error behavior with in LSM. Also, the BWR assembly and PWR assembly should have different LSM error behavior, because they are so different, such as different geometry size, different power density and system pressure, and different control rod system. The last concern is the existence of gadolinia, one kind of black burnable poisons, in some assemblies will lead to more non-linearity and thus higher error level. One BWR and one PWR assemblies, both of with have gadolinia pins, will be studied in this section using CASMO-3 to answer the above mentioned questions.

3.7.1 BWR Case

Consider the 8×8 BWR assembly with half symmetry shown in Figure 3.10. Pin types 1,2, and 3 are fuel rods with different U^{235} enrichment, and pin type 4 consists of fuel and gadolinia (BA). Pin type 5 is a water hole. The water hole is "frozen" in excluded position, and sometimes pin type 4 could be either frozen in place or just excluded only

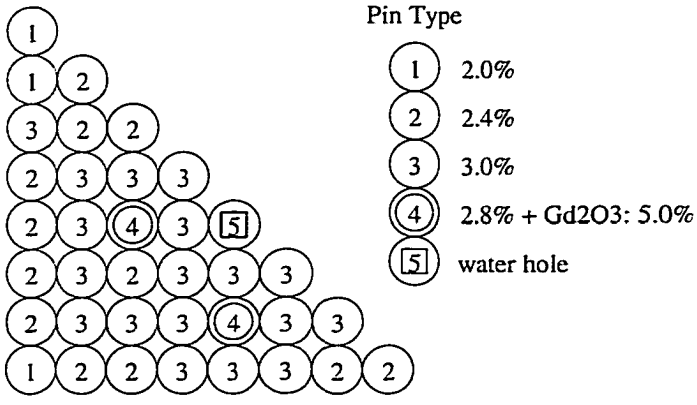


Figure 3.10 BWR reference assembly

from peripheral positions in the assembly. The allowed fuel enrichment perturbation ranges are (1.5wt%, 2.5wt%) for pin 1, (2.0wt%, 3.0wt%) for pin 2, (2.5wt%, 4.0wt%) for pin 3 and (2.0wt%, 3.5wt%) for pin 4, while the BA perturbation range for Gd_2O_3 is (4.0wt%-6.0wt%). The perturbation step size is 0.05wt% for all material compositions. In the material library, the calculated points are at every 0.1wt% from the reference values.

Figure 3.11 shows the errors of one thousand randomly material perturbed assemblies. The maximum MAXPE is under 0.011, while 90% of the assemblies are under 0.006. The maximum PPE is under 0.011, while 90% of them are less than 0.003. The maximum MAXKE is 0.00073, while 90% of them are less than 0.00025. Compared with the PWR case using CPM-2, the errors level are a little higher. These difference could be attributed to the presence of gadolinia and/or to the higher levels of nonlinear behavior in BWRs.

Figure 3.12 shows the errors of one thousand randomly spatial perturbed assemblies. The left figure show the results of spatial perturbations with pin type 4 shuffling, while the right figures without pin type 4 shuffling (*i.e.* , the pins of type 4 are frozen at their initial places as in the reference assembly). The error levels without Gd_2O_3 shuffling are similar with those of PWR shown in previous sections using CPM-2, but those with Gd_2O_3 shuffling are much worse. The maximum power peak error is 6.6% (4.6% for 90% of the assemblies), and the MAXKE is 5.28% (4.48% for 90% of the assemblies). As mentioned earlier, the higher error level is possible due to the strong "black" burnable absorber of gadolinia and to the nonlinear behavior of BWR.

Figure 3.13 and Figure 3.14 show the errors of the combined library and the combined simplified library using CASMO-3. For the combined library with Gd pin shuffling, the maximum power peak error is as high as 14% (8.3% for 90% of the assemblies), and MAXKE is about 5.2% (4.3% for 90% of the assemblies). Freezing the Gd pins improves the maximum power peak error a little bit to 13% (6.4% for 90% of the assemblies),

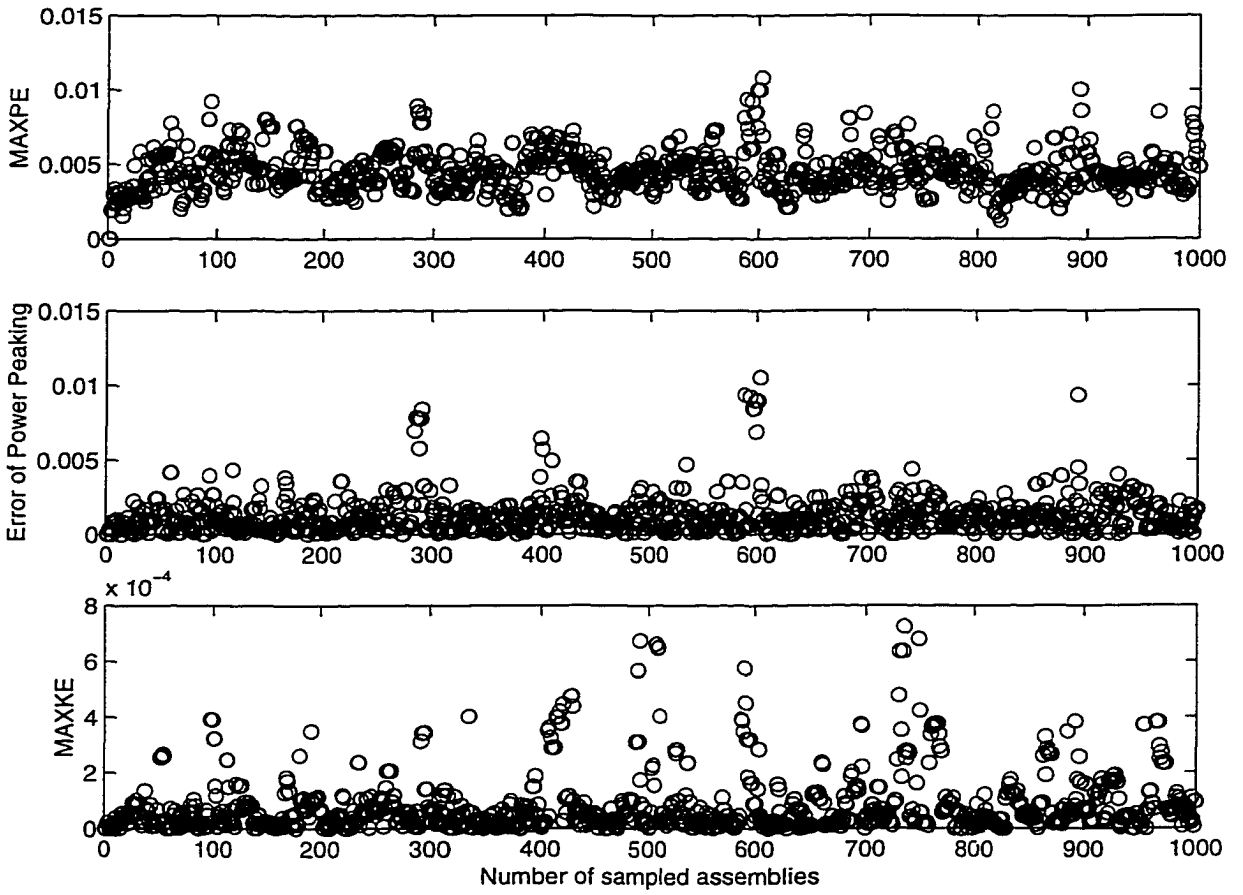


Figure 3.11 Error behavior of LSM material perturbations using CASMO-3

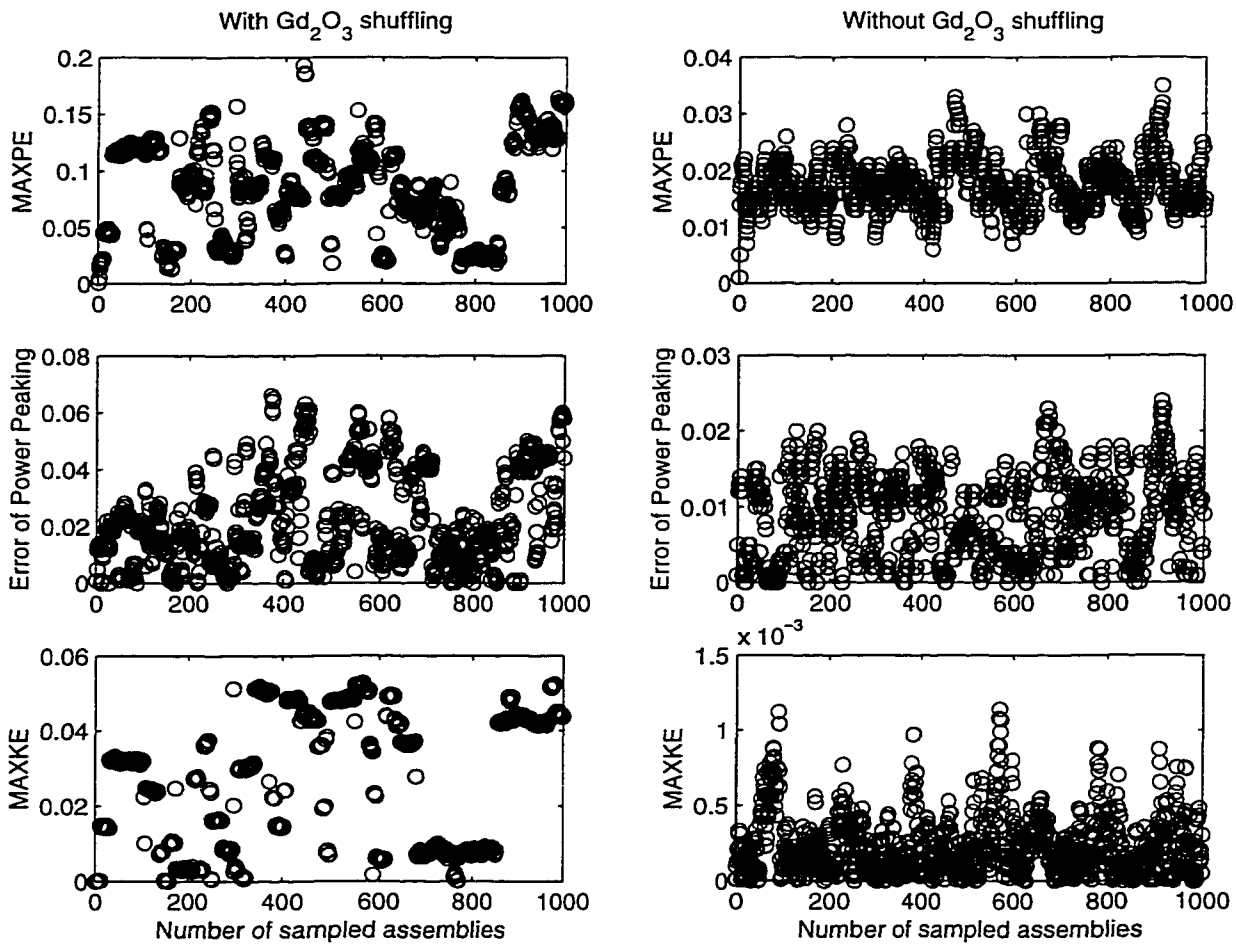


Figure 3.12 Error behavior of LSM spatial perturbations using CASMO-3

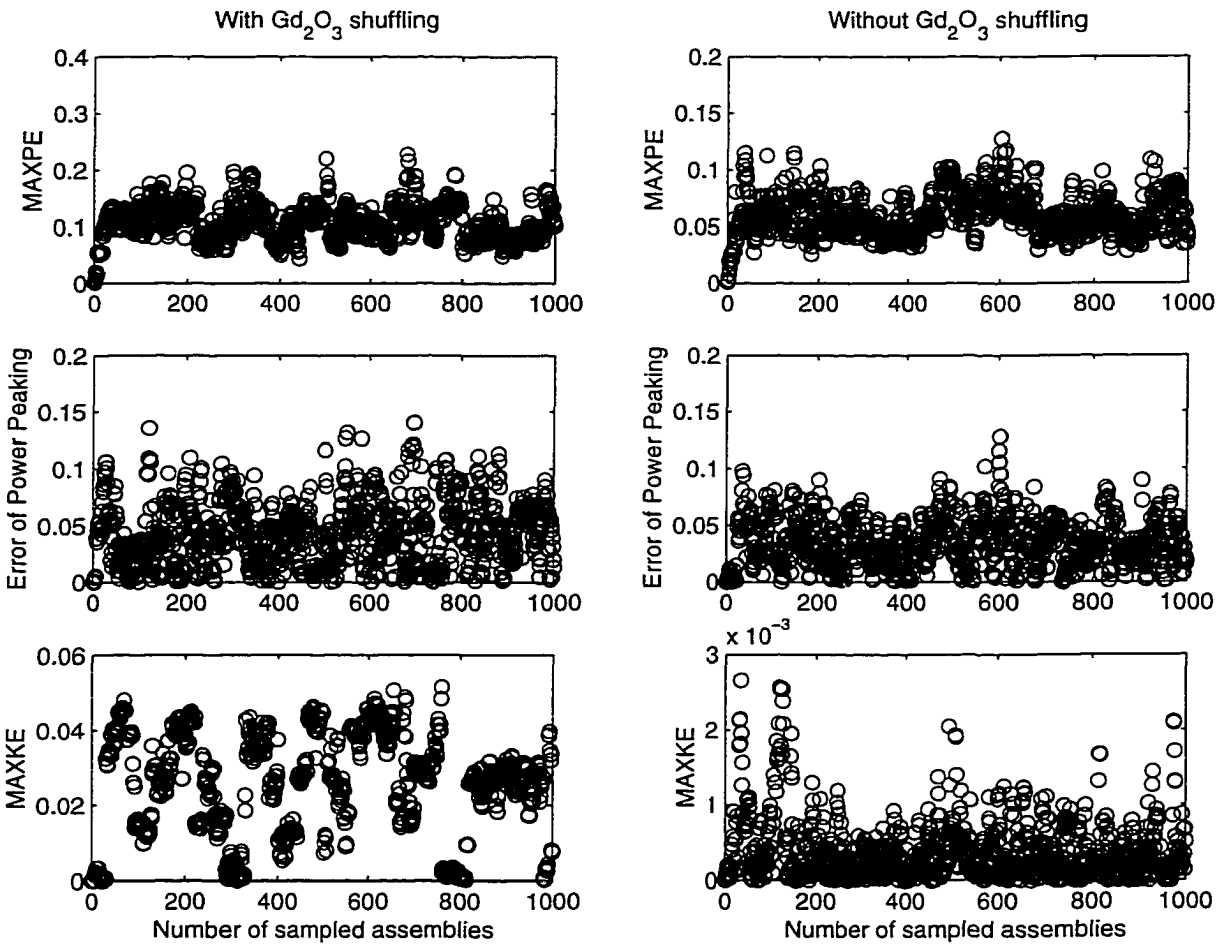


Figure 3.13 Error behavior of LSM combined library using CASMO-3

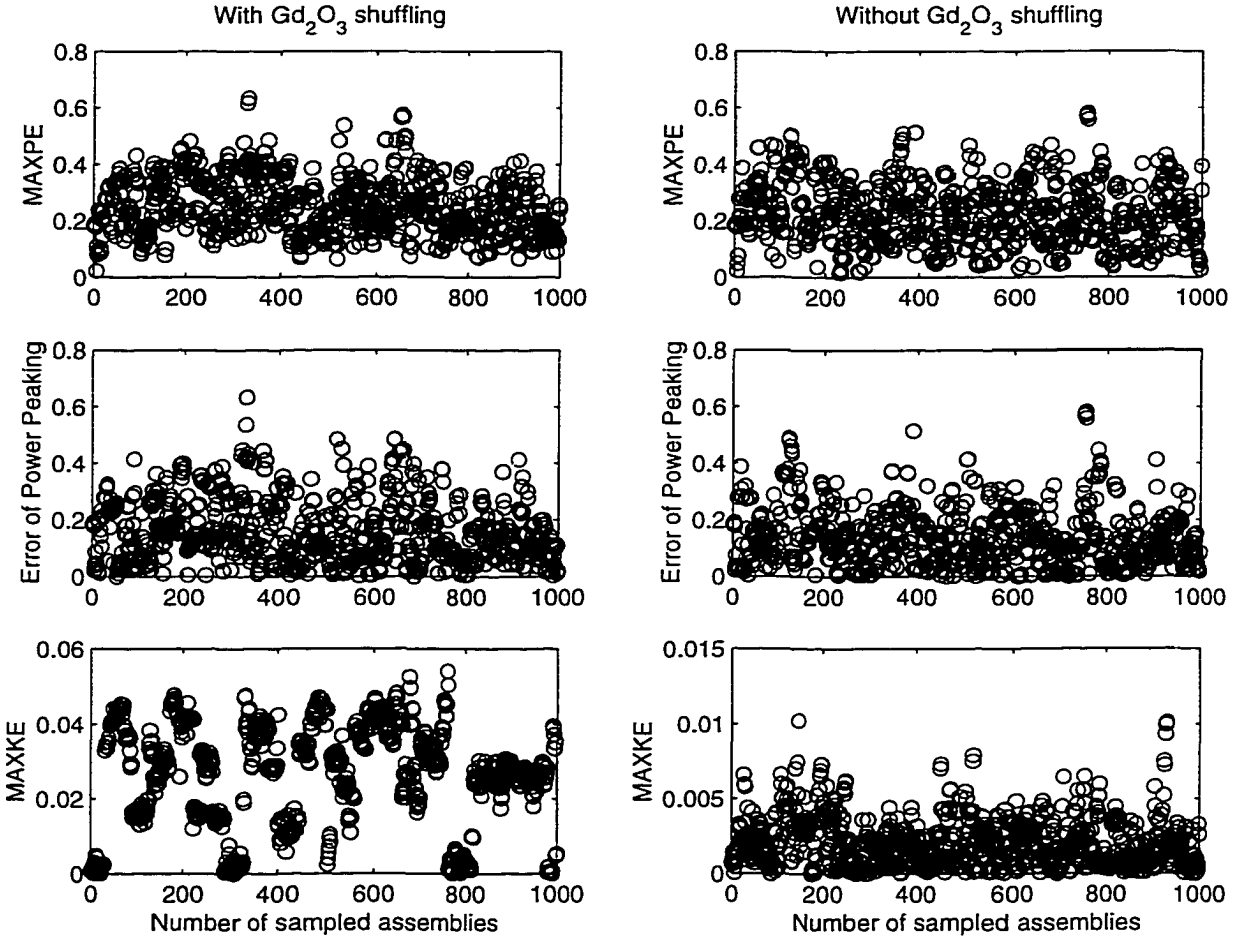


Figure 3.14 Error behavior of LSM combined simplified library using CASMO-3

while MAXKE was improved very much to a level as low as 0.27% (0.09% for 90% of the assemblies). It can be concluded that the spatial perturbation error play an important role in the total pin power error, so the future work is to find a way to reduce the spatial perturbation error without a big increase of the library size, if necessary. Also the higher number of total summation terms leads to more round-off error in the lattice-physics parameters estimation, especially in relative pin power error, which can be solved if the source codes of CASMO-3 are available. Obviously, the error levels of the combined simplified library are too high to be acceptable. The high level error indicates no only that the combined simplified library can not be useful in the lattice-physics estimations, but also that the separation of material perturbations and spatial perturbations is necessary, because the combined simplified library happens to be the case where the material perturbations and spatial perturbations are not separated with the dual libraries (MP and SP). The assumption of separation keeps the errors in a low level.

3.7.2 PWR Case

It is observed that the LSM error level of a BWR assembly using CASMO-3 are a little higher than the LSM error level of a PWR assembly using CPM-2. The difference of error level is possibly due to the difference of CPM-2 and CASMO-3, or the difference of PWR and BWR. To better identify the reason of the different error levels, one more test has been studied on an octant-symmetric 17×17 PWR assemblies, shown in Figure 3.15. Pin types 1, 2, and 3 are fuel pins while pin type 3 contains gadolinium. Pin type 4 is instrument tube and Pin type 5 and 6 are guide tubes, where pin type 6 contains burnable absorber B_4C. Similarly to the previous section, the PWR assembly were studied at one BOL burnup step.

Table 3.2 shows the results of the error analysis of this PWR assembly. It is found that the error behavior is similar with that of the PWR case using CPM-2 code, when

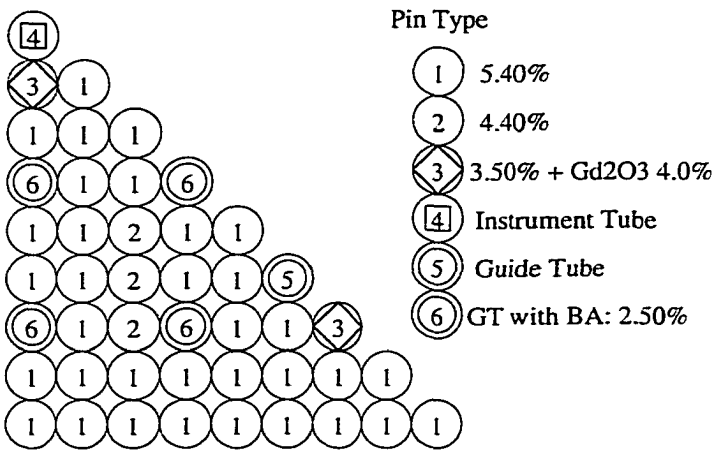


Figure 3.15 17x17 PWR assembly

the gadolinia pins are not allowed in shuffling. And the error level is a little higher when the shuffling of gadolinia pins are allowed. It can be concluded that the reason of higher level of BWR case is not the difference of the two codes, but the difference of BWR and PWR assembly. Also, the existence of gadolinia leads higher error for both PWR and BWR case in the spatial perturbation library and the combined library.

3.8 Discussions

The total LSM error consists truncation error and round-off error, both of which play significant role for pin power and only the former of which dominates the k_∞ error. The error level of all LSM models can be reduced by eliminating the round-off error if the source codes of CPM-2 or CASMO-3 are available to the users, like we did in CPM-2M by outputting more significance digits of relative pin power and k_∞.

Through these error analysis in this chapter, we found that the pin power error and k_∞ error of the material perturbation library has been reduced to an acceptable level via LSM of interpolation and second-order cross-term compensation. No matter of BWR or PWR, and no matter with or without gadolinia, the material perturbation library has a good error behavior. For spatial perturbation library, the error of BWR assembly is

Table 3.2 Error results of PWR case using CASMO-3

	Material Library	Spatial Library		Combined Library		Combined Simplified Library	
		with Gd shff.	without Gd shff.	with Gd shff.	without Gd shff.	with Gd shff.	without Gd shff.
MAXPE	0.029	0.235	0.006	0.253	0.116	0.596	0.609
	(0.011)	(0.114)	(0.005)	(0.105)	(0.053)	(0.226)	(0.233)
RMSPE	0.008	0.042	0.003	0.050	0.039	0.144	0.127
	(0.002)	(0.024)	(0.002)	(0.027)	(0.016)	(0.419)	(0.365)
MAXKE	0.00201	0.01602	0.00018	0.02093	0.00291	0.01567	0.00237
	(0.00062)	(0.00827)	(0.00011)	(0.01120)	(0.00134)	(0.00942)	(0.00096)
PPE	0.011	0.042	0.006	0.105	0.059	0.174	0.140
	(0.005)	(0.011)	(0.002)	(0.035)	(0.018)	(0.091)	(0.079)

Figures in parentheses are under which the error of 90% assemblies are.

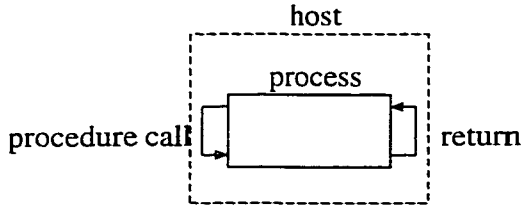
higher than that of PWR, and the existence of gadolinia will increase the error. The error of the combined library is in a quite low level when the gadolinia pins are frozen. The high error level of combined simplified library makes it a bad model and it can not be applied in FORMOSA-L. The failure of this model proves that the assumption of separation of material and spatial perturbations are necessary.

4 PARALLEL COMPUTING CREATION OF LSM LIBRARY VIA RPC

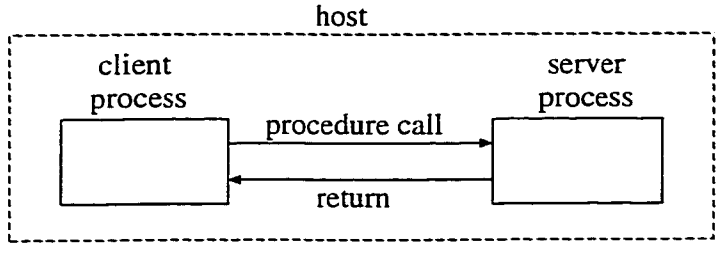
4.1 Brief Introduction to Remote Procedure Call (RPC)

As previously mentioned, performing computations in parallel is an attractive choice for reducing the large computation times typically encountered with these large-scale non-linear combinatorial optimization problems. A single Markov chain is often followed in traditional SA, thus, limited parallelism can be exploited. However, Lee, K. G. (1992) has developed multiple Markov chain parallel schemes, with both synchronous and asynchronous implementations being studied. In LSM, the primary expenditure of CPU time occurs while constructing the superposition library (not during the optimization), a process which could be done completely in parallel. It should be noted that, instead of pursuing computer platforms with parallel processors, this study is more interested in pursuing clusters of networked workstations.

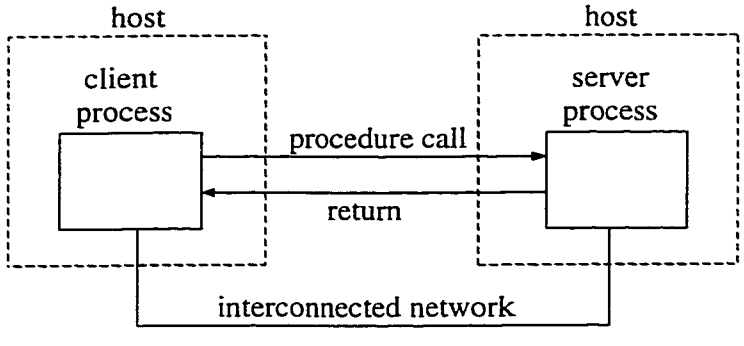
A *local procedure call* is analogous to its counterpart in standard serial programming. The procedure being called (callee) and the calling procedure (caller) are in the same process. A *remote procedure call* (RPC) occurs when the callee and caller are in different processes (client and server processes). Further, RPC in general allows a client on one host to call a server procedure on another host, as long as the two hosts are connected by some form of network [Stevens, W. R. (1999)]. These three types of procedure calls are illustrated in Figure 4.1. Thus, rather than building a large program that performs everything, the application can be divided among multiple processes, and some of the



Local Procedure Call



Remote Procedure Call (RPC) on a single host



Remote Procedure Call (RPC) between hosts

Figure 4.1 Three types of procedure calls

processes can run on different hosts which are interconnected by a network.

To implement network communications among various pieces of the application, most popular applications (Web browsers, Web servers, Telnet clients, Telnet servers, etc.) are written using *explicit network programming*. For example, using the socket API, clients call *socket*, *connect*, *read*, and *write*, whereas servers call *socket*, *bind*, *listen*, *accept*, *read* and *write*. RPC provides an alternative way by using *implicit network programming*. This enables the development of parallel applications by using the familiar procedure call, but the client and server can be executing on different hosts. The major advantage being that RPC controls the network traffic and load balancing such that it does not need be a concern of the programmer.

4.2 Applications of RPC to LSM Library Creation

The library creation typically requires tens between hundreds of lattice physics calculations, either with CPM-2 or CASMO-3, which are both computationally intensive. Therefore, the library creation is truly the bottleneck of the FORMOSA-L code. The solution to this problem was to create the library in parallel. The envisioned application of RPC for the LSM is shown in Figure 4.2. The workload of library creation is distributed to several workstations and the total time used in library constructing is greatly reduced. This reduction speeds up the FORMOSA-L code. Though it is true that the more the hosts the faster the code, up to 10, usually 3 to 5, hosts are suggested to be used in practice, this is because the efficiency will be decreased with the increase of the number of hosts.

Figure 4.3 and Figure 4.4 illustrate the two different implementations, *i.e.* synchronous implementation and asynchronous implementation, respectively. In the synchronous implementation, n runs are set up and then allocated to n workstations. n children processes deal with these runs by calling the remote procedure on each host

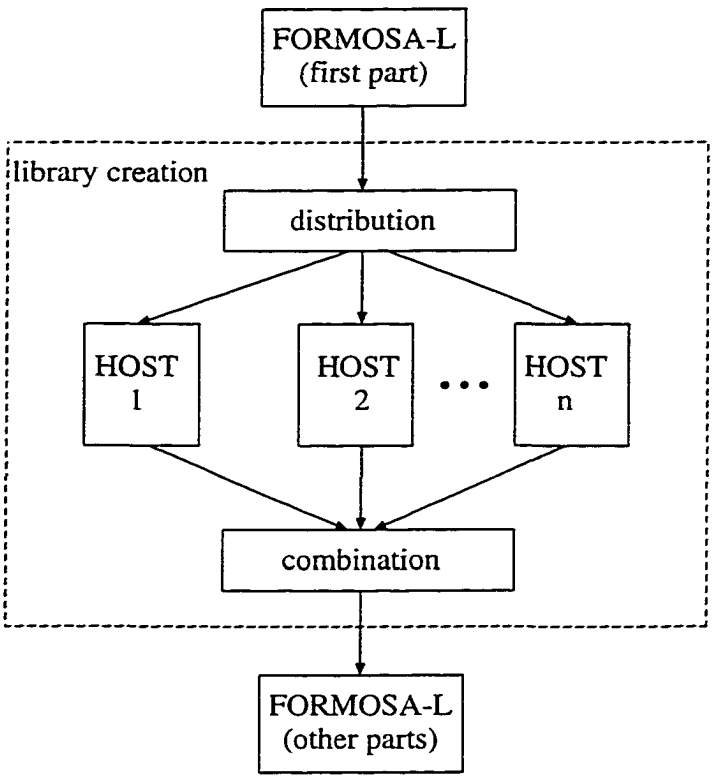


Figure 4.2 Application of RPC in LSM

and executing in parallel. The code then waits for the conclusion of all the children processes. The outputs of these runs are processed one at a time. Another n runs are setup and executed after the output readings. The above steps are continued until the library creation is finished. In practice, the synchronous implementation is easy to code and the performance is good if the speeds of all the used hosts are similar to each other. When the speeds of all the hosts differ a lot, the disadvantage of synchronization comes about because the fast machines must wait for the slowest machine and the CPU resources are wasted somehow. The worst scenario occurs when the number of total hosts is small and the differences of their speeds are large.

To overcome the drawback of synchronous implementation, asynchronous implementation has been developed. Two indicators are introduced for each host, one of witch indicates whether the host is free or occupied and the other of which indicates whether

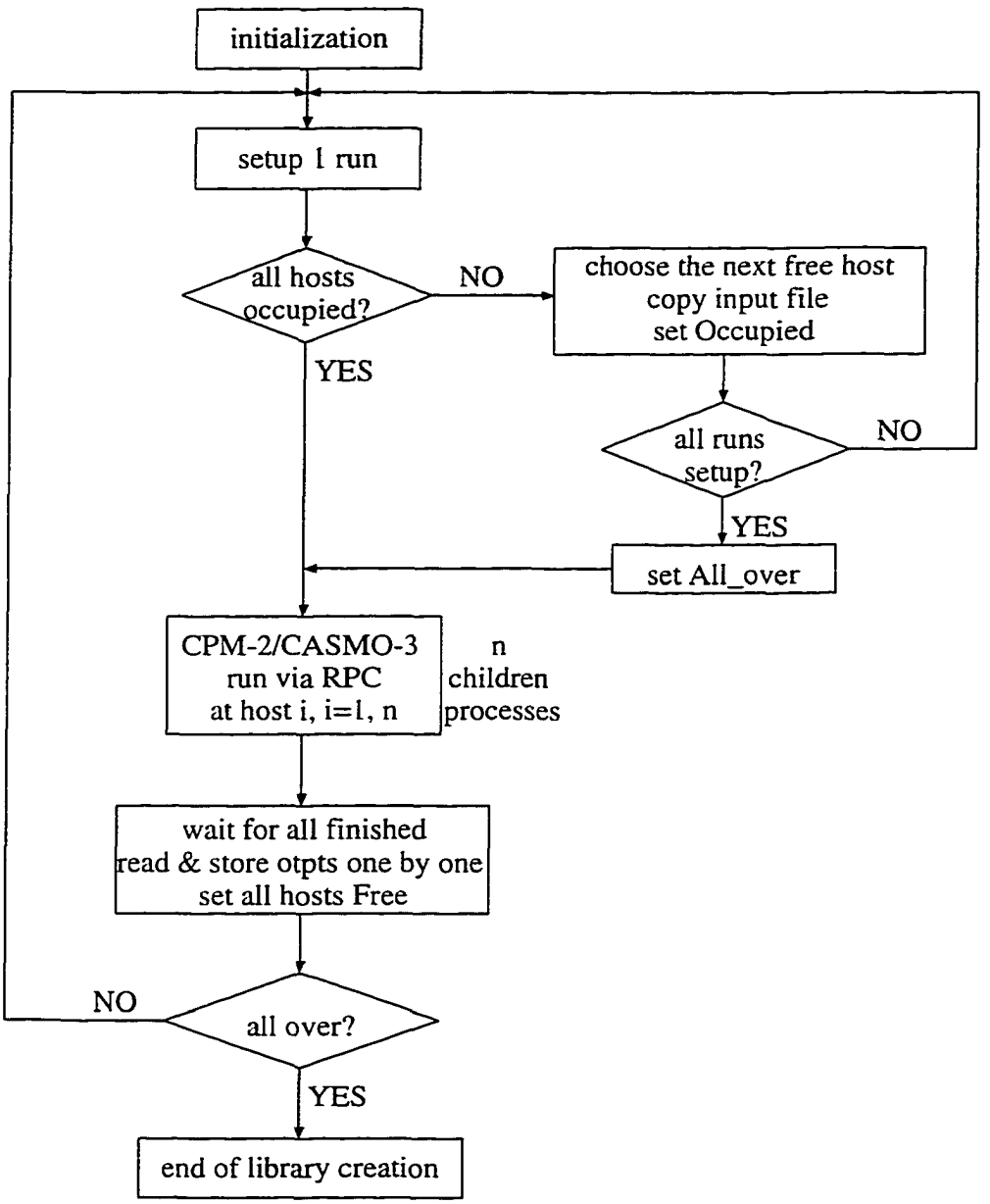


Figure 4.3 Application of RPC in LSM: synchronous implementation

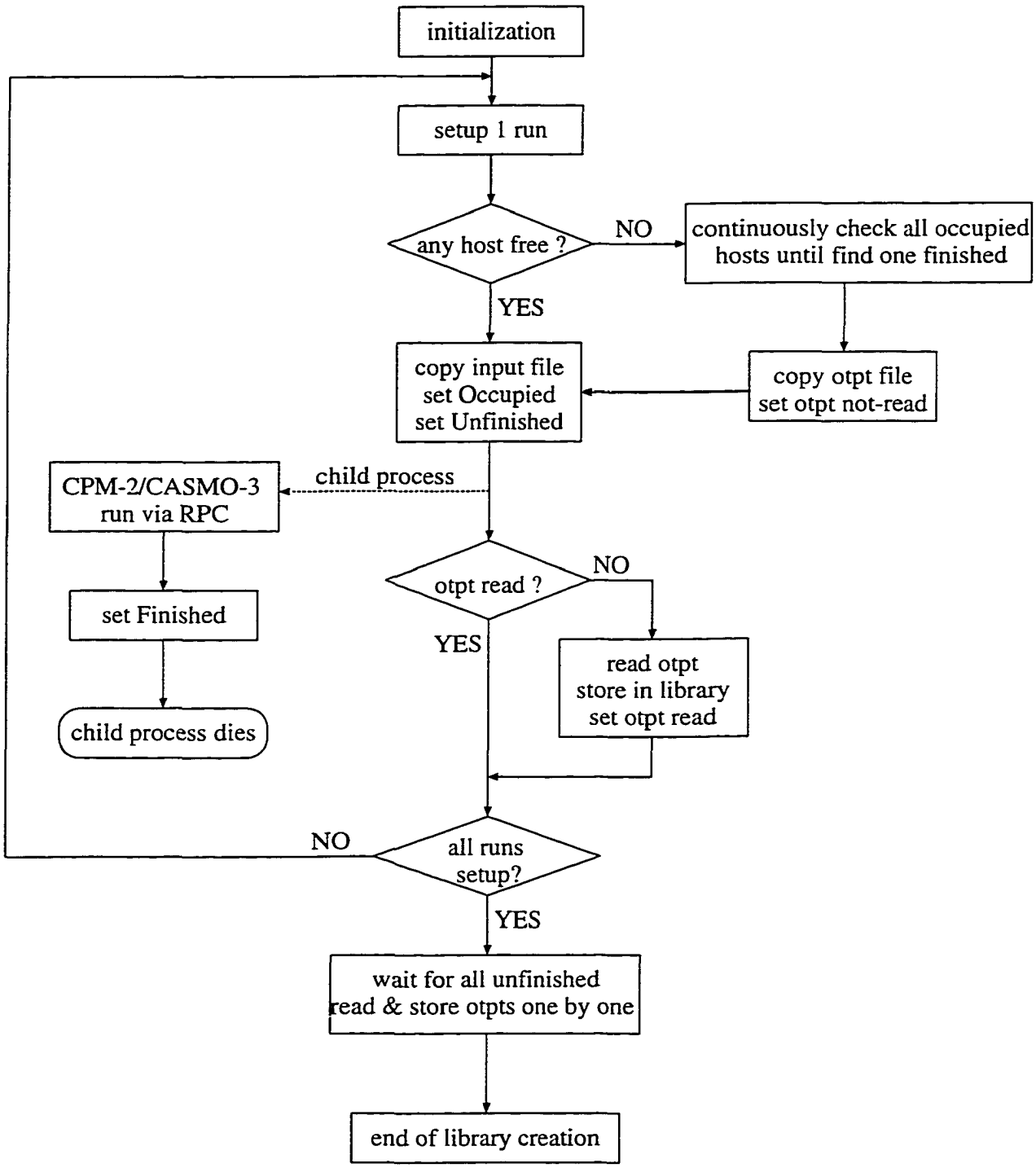


Figure 4.4 Application of RPC in LSM: asynchronous implementation

the run on the occupied host is finished or not. Three possible states exist for each host, *i.e.* free, occupied but running finished, occupied and running not finished. After one running has been set up, the code looks for any free host to execute it. If all hosts have been occupied, the code checks the "finished_or_not" indicators. If no host finishes its run, the code waits until any one of them finishes. Then the code copies the output file and allocates the newly set-up run to the free host. The results are read from the output file and stored in the library. Then the next run is set up until all runs are done. Clearly, the most efficient asynchronous implementation occurs when the workstations are used in the most efficient way. The faster machines don't need to wait for the slower machines and this benefits the asynchronous implementation with better speed-up performance. The faster machines simply continue to execute more runs than the slower ones and no CPU resource is wasted because the machine is in use again once it is set free. Also, the number of hosts and the differences of the machines' speeds do not matter in the asynchronous implementation.

4.3 Results

Consider the 15×15 PWR assembly in Figure 3.2 as the reference assembly. Twenty burnup steps (through the fuel cycle) are calculated for each run. Only one spatial perturbation library is created. The water hole and discrete BA pins are frozen at their positions, so 61 CPM-2 runs are required for this case. Two workstations were employed for this test: host 1 is a 500MHz DEC alpha workstation and host 2 is a 300MHz DEC alpha workstation.

The results are shown in Table 4.1. It is found that synchronous implementation reduces the run time by about 50% with respect to the slower workstation, but there is no speed-up but rather a slow-down compared with the faster host, as predicted in the previous section. The asynchronous implementation however displayed important

speedup s relative to both hosts individually, almost 70% relative to the slower host and 25% relative to the faster host.

Limited by the total number of available workstation in local facility, results using more platforms are not available. However, it can be safely predicted that with more available machines, the greater the speed-up. If the machines' speeds are comparable to each other, the LSM and FORMOSA-L should achieve speedups in proportion to the number of workstations available at any given time.

Table 4.1 Time saving of parallel computing via RPC

computing type	Number of host(s)	Host NO.	Time used (sec)	percent	percent
serial	1	1	3455.2	N/A	100%
serial	1	2	8262.6	100%	N/A
parallel: syn.	2	1&2	4228.7	51.2%	122.4%
parallel: asyn.	2	1&2	2617.4	31.7%	75.8%

5 APPLICATION OF LSM TO FORMOSA-L

5.1 Introduction of FORMOSA-L

The primary goal of lattice-level loading design and optimization of LWR fuel assembly is to optimize an assembly's spatial distribution of fuel enrichment and BAs on the "pin-by-pin" basis. Experienced fuel designers must repeatedly cycle through various sequences of interconnected nuclear design calculation until a pre-established criteria related to performance and nuclear safety are satisfied. FORMOSA-L is an optimization engine applicable to light water reactor fuel assembly loading design to delegate the manpower-intensive task by automatically considering all the appropriate objectives and constraints. The optimization technique employed is the simulated annealing algorithm, while the evaluation of the design objectives and constraints is handled by coupling to the lattice-physics code CPM-2 or CASMO-3 in a non-intrusive manner (via I/O files).

In FORMOSA-L, the primary optimization control variables include the "pin-by-pin" placement and/or loading of nuclear fuel, burnable absorbers, or other material regions within a fuel assembly. The pin type number and the number of pins of each type are keep constant. By starting from a reference lattice loading pattern, new pattern candidates are stochastically generated by randomly perturbing the latest accepted pattern. The perturbations consist of a binary shuffling of two pins of different pin types, *i.e.* spatial perturbation, the changes of fuel enrichment and/or BA concentration within a specific pin type, *i.e.* material perturbation.

The objective functions available in the FORMOSA-L code include:

- Minimization of power peaking;
- Minimization of assembly-averaged enrichment;
- Minimization of total uranium cost;
- Minimization of a combined objective function of power peaking and average enrichment;
- Maximization of the EOL k_∞.

The typical constraints include burnup-dependent upper and lower bounds on assembly-averaged k_∞, burnup-dependent maximum pin power peaking limit, maximum assembly-averaged enrichment. The above constraints are applied as hard constraints or penalty constraints. Other minor constraints can be posed, such as exclusion and/or freezing a certain pin at some positions, limiting the perturbation space of fuel enrichment and/or BA concentration in a predefined scope or a sample pool.

The SA annealing schedule consists of four key aspects, initial temperature, final temperature, Markov chain length, and the temperature reduction coefficient. All of them can be either user defined or automatically determined by built-in procedures in FORMOSA-L code. Also, the code will end after running a certain number of cycles as user input, or after the convergence criterion being satisfied.

5.2 Applications of LSM to FORMOSA-L

As a fast, yet accurate, model of approximation of lattice-physics parameters, LSM can replace the direct lattice-physics evaluation code through the optimization process. One type of LSM library is needed to create firstly before the SA algorithm begins its annealing precess. The reference assembly which the library is based on is from the input file. If more cooling cycles are needed, the library will be re-created at the beginning of each cycle based on the optimal assembly got in the previous cycle.

When separate libraries are employed, material and spatial perturbations are performed in alternating cooling cycles. This separation maintains the LSM accuracy within acceptable levels. As mentioned in Chapter 3, the power peaking error and k_∞ error are much lower than those of combined libraries. The low level of errors will assure not to mislead the optimization. The drawback of this separation of libraries is that the optimization problem is divided into two sub-problems, which limits the extent of the search space. Nevertheless, this can be addressed by running additional cooling cycles, because smaller size of the libraries makes this practical. Of course, further speed-ups construction can be obtained by parallel library.

if the combined library is used in the FORMOSA-L, the material perturbations and spatial re-arrangements can be performed simultaneously. However this causes a new problem whereby the higher error level will lead new fidelity problem. In the actual optimization problem, the error level can be reduced if the reference assembly is quite good to begin with. Also, multiple cooling cycles can help because the optimal assembly obtained from the previous cooling cycle should be good enough to limit the perturbation levels to be small. Fidelity issues will be studied in one of the latter sections.

Regardless of which type of available libraries is employed in the FORMOSA-L, two options can be provided. The first is that the LSM approximations of the lattice-physics parameters are used through the cooling schedule in a cycle after the library is prepared. In this option, the several near-optima assemblies obtained at the end of the cooling cycle will have some errors and their real lattice-physics parameters can be obtained by direct evaluation using CPM-2 or CASMO-3 and the error level will be checked if it is low enough to be an acceptable level. If not, more cooling cycles are needed. The second possible application option of the LSM into the FORMOSA-L involves the LSM approximation of lattice-physics parameters being used only at the first part of each cooling cycle when the temperature is high. When the temperature approaches the final (lowest) temperature, the direct evaluation, CPM-2 or CASMO-3,

is toggled back in and the material and spatial perturbations can then be made at the same time. Though the run time increases compared with the first option, returning to direct evaluation benefits the optimization process by two ways. One is that the search space is broadened by simultaneous perturbations in separated library case. The other is that the errors are eliminated. In this option, the LSM is used as a fast scoping tool.

5.3 Fidelity Study of LSM

5.3.1 Robustness of Separability Assumption (BWR study)

The robustness of the separability assumption has been thoroughly tested by performing several power peaking minimizations on a BWR fuel assembly with a beginning-of-life (BOL) burnup step [Maldonado, G. I., (1999)]. The specific lattice selected was chosen because of its relatively large number of pin types (11) and because of the inherently challenging aspects of combining UO_2, MOX (mixed oxide), and gadolinia loadings within the same assembly. The sample cases studied are described below:

C1 BWR assembly power peaking minimization; one cooling cycle; one burnup step (BOL); all histories evaluated directly with CPM-2.

C2 BWR assembly power peaking minimization; four cooling cycles; one burnup step (BOL); all histories evaluated with LSM except those near the end of each cooling cycle where direct CPM-2 evaluations were toggled back on.

C3 BWR assembly power peaking minimization; four cooling cycles; one burnup step (BOL); all histories evaluated with LSM. For statistical considerations, sub-cases 3a through 3d were carried out by starting from different initial assembly designs and/or using a different random number seed.

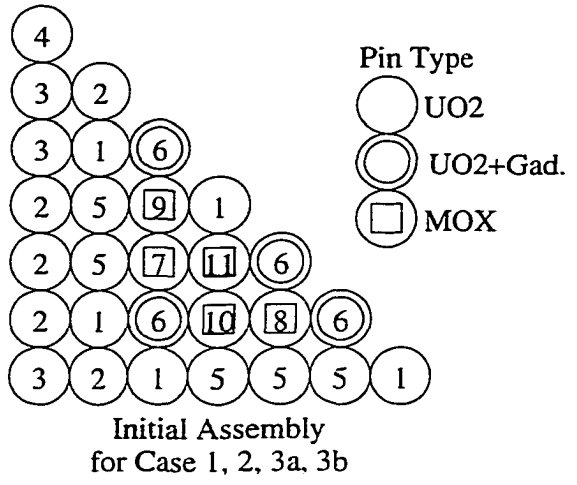
In all cases above, the assembly-average k_∞ profiles versus burnup were constrained to within $\pm 1.0\%$ of the original (reference) assembly's k_∞. The assembly tested is an

half-symmetric 7×7 MOX and gadolinia-loaded BWR assembly lattice with 11 different types of fuel pins, as illustrated by Figure 5.1. Pin types 1 through 5 contained UO_2, pin type 6 contained UO_2 and gadolinium, while pin types 7 through 11 contained MOX pins. For these test cases, the control variables were the location of each pin, the ^{235}U enrichment of pin type 1 through 6, and the weight percent of Pu (over total heavy metal including U and Pu) of pin type 7 through 11. while the concentration of gadolinium was kept constant and the MOX pins contain natural uranium (fixed) and plutonium contents (variable).

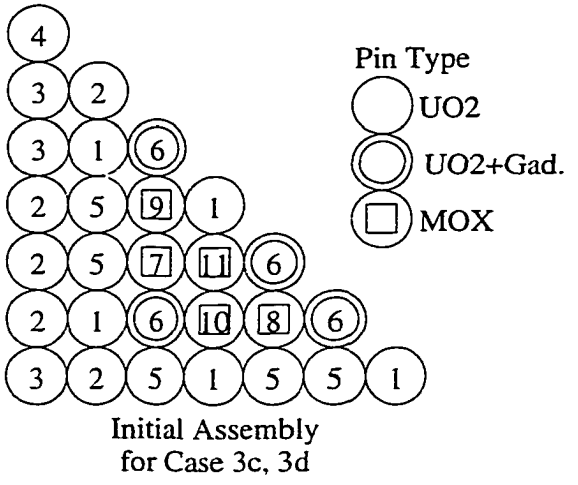
Table 5.1 describes the initial and resulting U_{235} enrichment contents and Pu weight percent from each above-noted optimization, while Figure 5.1 shows the corresponding before-and-after pin arrangements. Table 5.1 also provides the corresponding power peaking (P_{max}) for all the cases considered. The reference assembly's relative power peak was initially 1.258, while the minimized power peaking resulting from the "all-CPM" Case C1 was 1.169. Most notably, the power peaks for all five cases involving the LSM (cases C2 through C3) all yielded minimum values within 1.146 and 1.157. The progression of the optimizations corresponding to cases 1 through 3d are shown in Figure 5.2, which reveal typical simulated annealing behavioral trends for each of the four-cooling-cycle cases.

It could be stated that within a margin of error of about 2.0%, and as far as the final objective function value is concerned, all of the optimization results herein presented are fairly equivalent to each other and likely belong to the same family of near-optima. Consequently, based on these results, it appears that the usage of LSM and of the separability assumption during the optimization do not adversely misdirect the minimization process in a significant manner.

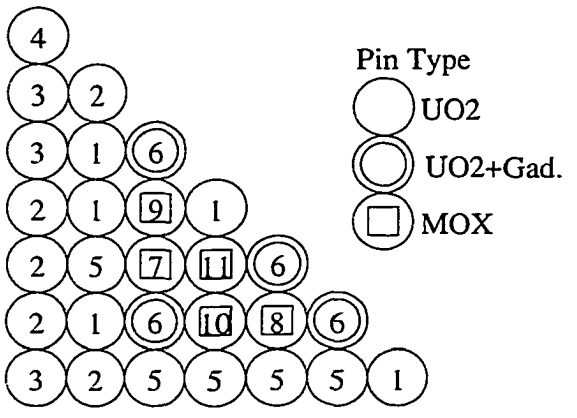
It should also be noted that Cases C3 (a through d) demonstrated excellent prediction performance by the LSM, with the largest absolute errors occurring during the first cooling cycle; namely, MAXPE of 0.007 and no error in power peaking , and errors in k_∞



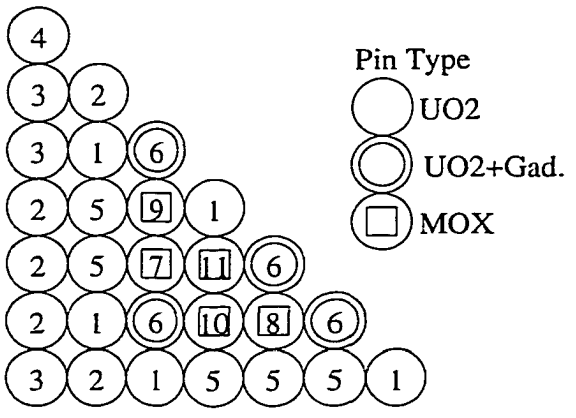
Initial Assembly
for Case 1, 2, 3a, 3b



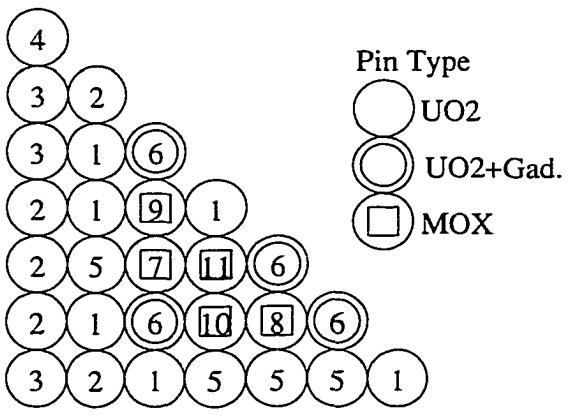
Initial Assembly
for Case 3c, 3d



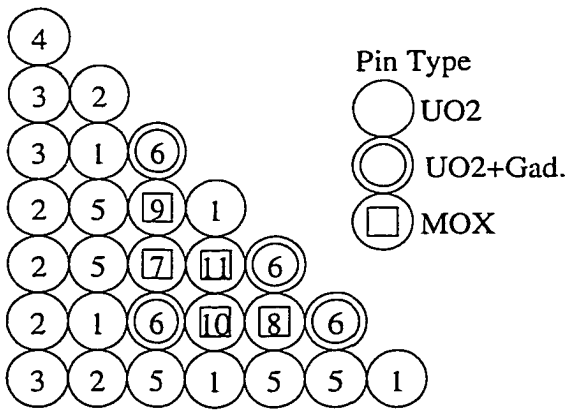
Case1: Optimized Loading (CPM-2)



Case2: Optimized Loading (LSM+CPM-2)



Case3a, b: Optimized Loading (LSM)



Case3c, d: Optimized Loading (LSM)

Figure 5.1 Pin layout of initial and optimized BWR sample assemblies

Table 5.1 Pin enrichment and power peaks for Case C1 through C3

		Pin 1 (U*)	Pin 2 (U)	Pin 3 (U)	Pin 4 (U)	Pin 5 (U)	Pin 6 (U)	Pin 7 (Pu**)	Pin 8 (Pu)	Pin 9 (Pu)	Pin 10 (Pu)	Pin 11 (Pu)	Pmax
C1	Init.	2.55	1.95	1.70	1.35	3.30	2.55	2.65	3.90	2.90	4.00	3.95	1.258
	Opt.	2.90	2.05	1.70	1.40	3.40	2.90	3.10	3.80	2.65	4.10	4.5	1.169
C2	Init.	2.55	1.95	1.70	1.35	3.30	2.55	2.65	3.90	2.90	4.00	3.95	1.258
	Opt.	2.80	1.95	1.60	1.30	3.35	3.50	3.0	3.75	2.90	3.50	4.3	1.148
C3a	Init.	2.55	1.95	1.70	1.35	3.30	2.55	2.65	3.90	2.90	4.00	3.95	1.258
	Opt.	2.80	1.95	1.60	1.35	3.35	3.50	3.05	3.75	2.80	3.40	4.35	1.150
C3b	Init.	2.55	1.95	1.70	1.35	3.30	2.55	2.65	3.90	2.90	4.00	3.95	1.258
	Opt.	2.65	1.90	1.55	1.30	3.20	3.50	2.80	3.45	2.65	3.30	4.30	1.146
C3c	Init.	2.40	1.95	1.70	1.35	3.30	2.55	2.65	3.90	2.90	4.00	3.95	1.273
	Opt.	2.85	2.00	1.65	1.35	3.30	3.50	3.10	3.70	2.95	3.45	3.60	1.157
C3d	Init.	2.55	1.95	1.70	1.35	3.30	2.55	2.65	3.90	2.90	4.00	3.95	1.263
	Opt.	2.80	2.00	1.65	1.35	3.30	3.45	3.10	3.75	2.85	3.35	4.45	1.153

(*) U-235 enrichment, wt%, over total Uranium

(**) Pu weight percent, wt%, over total heavy metal (natural Uranium and Plutonium)

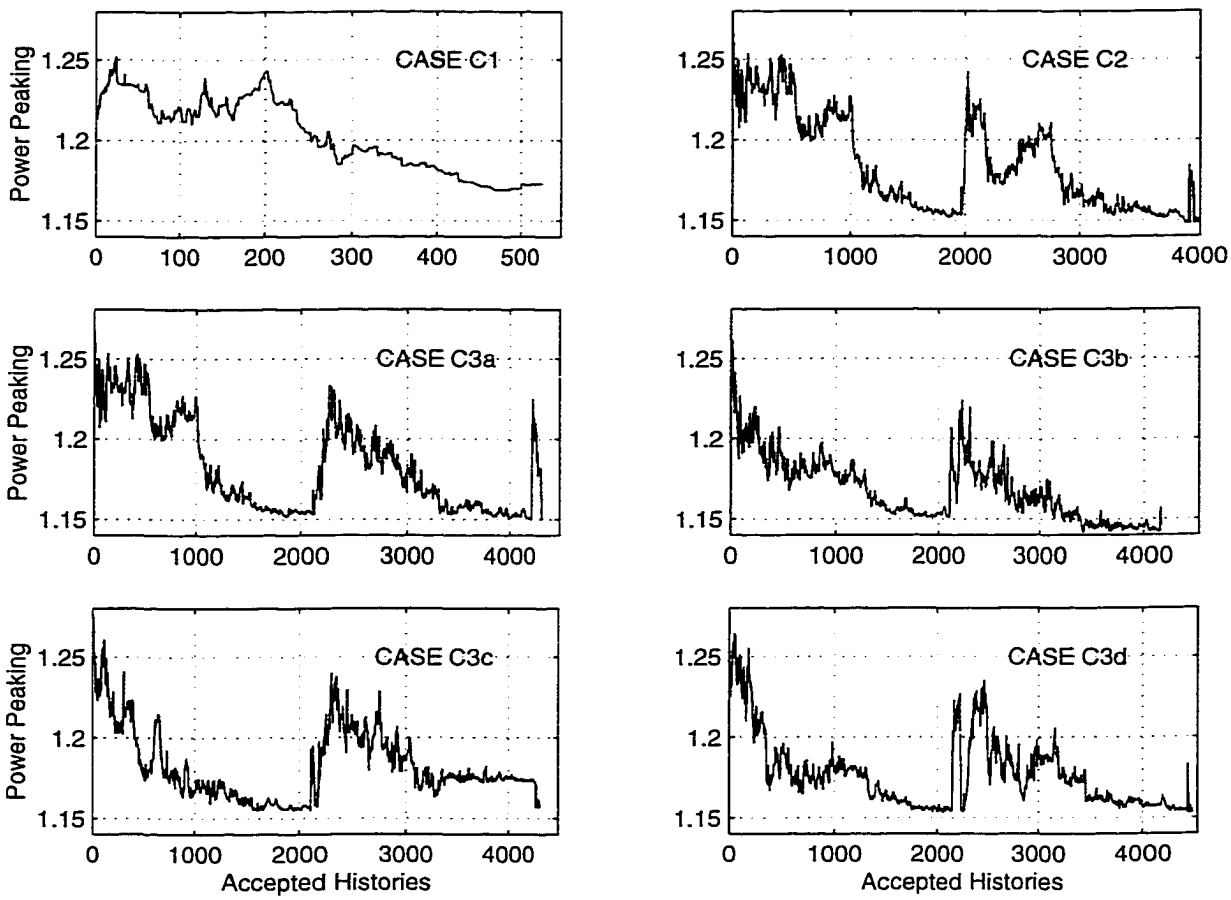


Figure 5.2 BWR power peak minimization Case C1 through 3d

of 0.00002. Additionally, it should be emphasized that optimization case C1 performed material and spatial perturbations simultaneously and not in alternating cooling cycles. Therefore, its comparable optimization outcome relative to cases C2 and C3 (in terms of objective function value) suggest that, so far, no major degradation in optimization fidelity has been detected as a consequence of the separability assumption.

For case C3, which employed LSM in its entirety, Figure 5.3 displays the pin-by-pin power errors versus a direct CPM-2 evaluation at the end of the first cycle. This test shows very good power prediction performance with the maximum error of almost 0.7% and the error of power peaking is zero. The error in k_∞ was less than 0.005%. In general, the overall performance is very encouraging.

Pin Power						
1.149						
*1.150						
0.001						
1.127	1.028					
1.125	1.024					
-0.002	-0.004					
0.992	1.108	0.357	⇐ CPM-2			
0.990	1.106	0.359	⇐ LSM			
-0.002	-0.002	0.002	⇐ ERROR=LSM - CPM-2			
1.100	1.149	1.023	0.545			
1.100	1.149	1.027	0.542			
0.000	0.000	0.004	-0.003			
1.084	1.130	1.048	1.096	0.299		
1.082	1.127	1.049	1.096	0.301		
-0.002	-0.003	0.001	0.000	0.002		
1.140	1.053	0.352	1.114	*1.150	0.339	
1.146	1.050	0.347	1.117	1.147	0.342	
0.006	-0.003	-0.005	0.003	-0.003	0.003	
1.064	0.973	1.020	1.035	1.005	1.076	1.145
1.064	0.974	1.017	1.042	1.003	1.075	1.144
0.000	0.001	-0.003	*0.007	-0.002	0.002	-0.001

k-infinity	
1.088814	⇐ CPM-2
1.088834	⇐ LSM
0.000020	⇐ ERROR=LSM - CPM-2

Figure 5.3 LSM errors for Case C3a optimized assembly at the end of the first cycle

5.3.2 Robustness of Separability Assumption (PWR study)

To better evaluate the robustness of the separability assumption, more tests have been implemented on an octant-symmetric 17×17 PWR assemblies, shown in Figure 3.15 in Chapter 3. Similarly to the previous section, six cases were studied. Note that three burnup steps are calculated for the assembly at 0.0, 15.0, 30.0 GWD/MTU.

C4 PWR assembly power peaking minimization; one cooling cycle; three burnup steps; all histories evaluated directly with CASMO-3.

C5 PWR assembly power peaking minimization; four cooling cycles; three burnup steps; all histories evaluated with LSM except those near the end of each cooling cycle where direct CASMO-3 evaluations were toggled back on.

C6 PWR assembly power peaking minimization; four cooling cycles; three burnup steps; all histories evaluated with LSM. For statistical considerations, sub-cases 6a through 6d were carried out by starting from different initial assembly designs and/or using a different random number seed.

Table 5.2 shows the power peaking of the "best" assemblies obtained in Case C4 through Case C6 cycle by cycle. The optimal power peaking of Case C5 through Case C6d are within the range from 1.087 to 1.096, which are very close but a little worse than the result of "all-CASMO" case C4, 1.067. The errors of power peaking is as low as 0.3% or less in the material perturbations cycle, *i.e.* cycle 1 and 3, while those in the spatial perturbations cycle, *i.e.* cycle 2 and 4 are less than 2.5% or less. Also the errors in later cycles are obviously less than those in the earlier cycles. The k_∞ errors, which are not shown in the table, are 10 pcm or less in the material perturbations cycle and 100 pcm or less in the spatial perturbations cycle, based on the observation of the "best" assemblies at the end of all the cycles in all the Case C6a through Case C6d.

5.4 Comparison of Different libraries

5.4.1 Combined Library Results (PWR Study)

Both types of combined libraries were tested on the same 15 × 15 PWR assembly used before in Figure 3.2. The following three optimization cases were performed:

C7 PWR assembly power peaking minimization; two cooling cycles; 10 burnup steps; all histories evaluated with LSM with separate material and spatial perturbation libraries.

Table 5.2 Power peaks for Case C4 through C6

	init.	cycle 1	cycle 2	cycle 3	cycle 4	opt.
C4	1.133	1.067	N/A	N/A	N/A	1.067
C5	1.133	1.095	1.094	1.094	1.093	1.093
C6a	1.133	1.103 (1.102)	1.101 (1.088)	1.096 (1.097)	1.095 (1.092)	1.095
C6b	1.133	1.102 (1.101)	1.097 (1.091)	1.097 (1.097)	1.096 (1.096)	1.096
C6c	1.135	1.108 (1.107)	1.105 (1.089)	1.097 (1.096)	1.087 (1.087)	1.087
C6d	1.134	1.108 (1.105)	1.115 (1.093)	1.107 (1.107)	1.093 (1.091)	1.093

Figures in parentheses were obtained by LSM

C8 PWR assembly power peaking minimization; one cooling cycles; 10 burnup steps; all histories evaluated with LSM using the standard combined library.

C9 PWR assembly power peaking minimization; one cooling cycles; 10 burnup steps; all histories evaluated with LSM using the simplified combined library.

Table 5.3 highlights the most important aspects of the results. Namely, the results from employing the standard combined library, Case C8, showed an improvement in the objective function at the expense of a much higher computation requirement relative

Table 5.3 CPU cost, power peaks and error for Case C7 through C9

	CPU cost(*)	initial PP	cycle 1		cycle 2		optimized PP
			PP	MAXKE	PP	MAXKE	
C7	144	1.098	1.083 (1.084)	0.000042	1.075 (1.077)	0.000006	1.075
C8	3613	1.098	1.049 (1.047)	0.000215	N/A	N/A	1.049
C9	143	1.098	1.083 (1.084)	0.000020	N/A	N/A	1.083

(*) CPU Cost Normalized to Units of CPM-2 Executions;
Figures in parentheses were obtained by LSM;
PP denotes power peaking.

to Case C7 (separate libraries). The improvement is the benefit of more search space coverage because of the simultaneous perturbations, which outweigh the impact of higher error level. Finally, the simplified combined library result, Case C9, seems to indicate that the larger prediction error may be causing increased optimization misdirection, thus, the outcome is comparable to that of the separate LSM case but worse than that of standard combined library. In spite of simultaneously optimizing spatial and material properties during the SA search, it seemed that all perturbations involving spatial perturbations were not accepted and the combined simplified library was demoted to a material library. The initial and final assembly designs, as well as the progressions of these optimizations are presented in Figures 5.4 and 5.5.

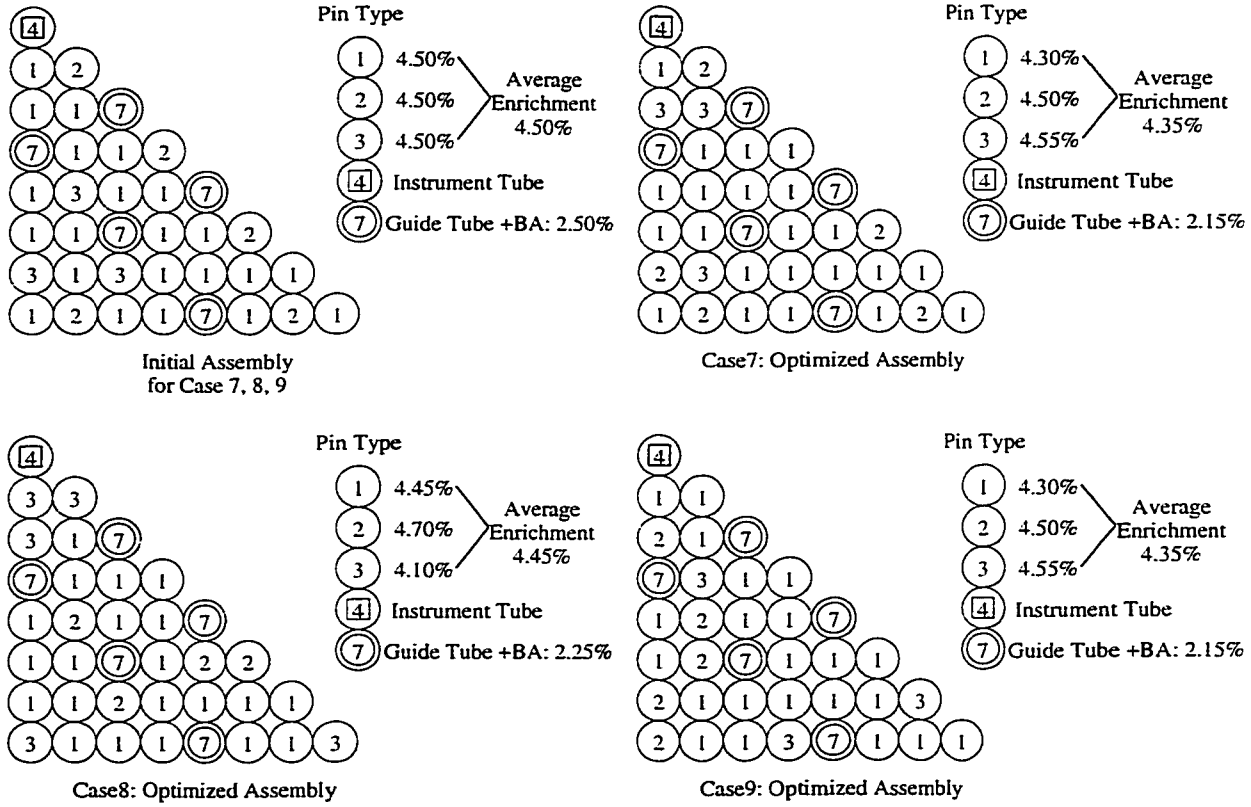


Figure 5.4 Initial and optimized PWR assemblies for Case C7 through C9

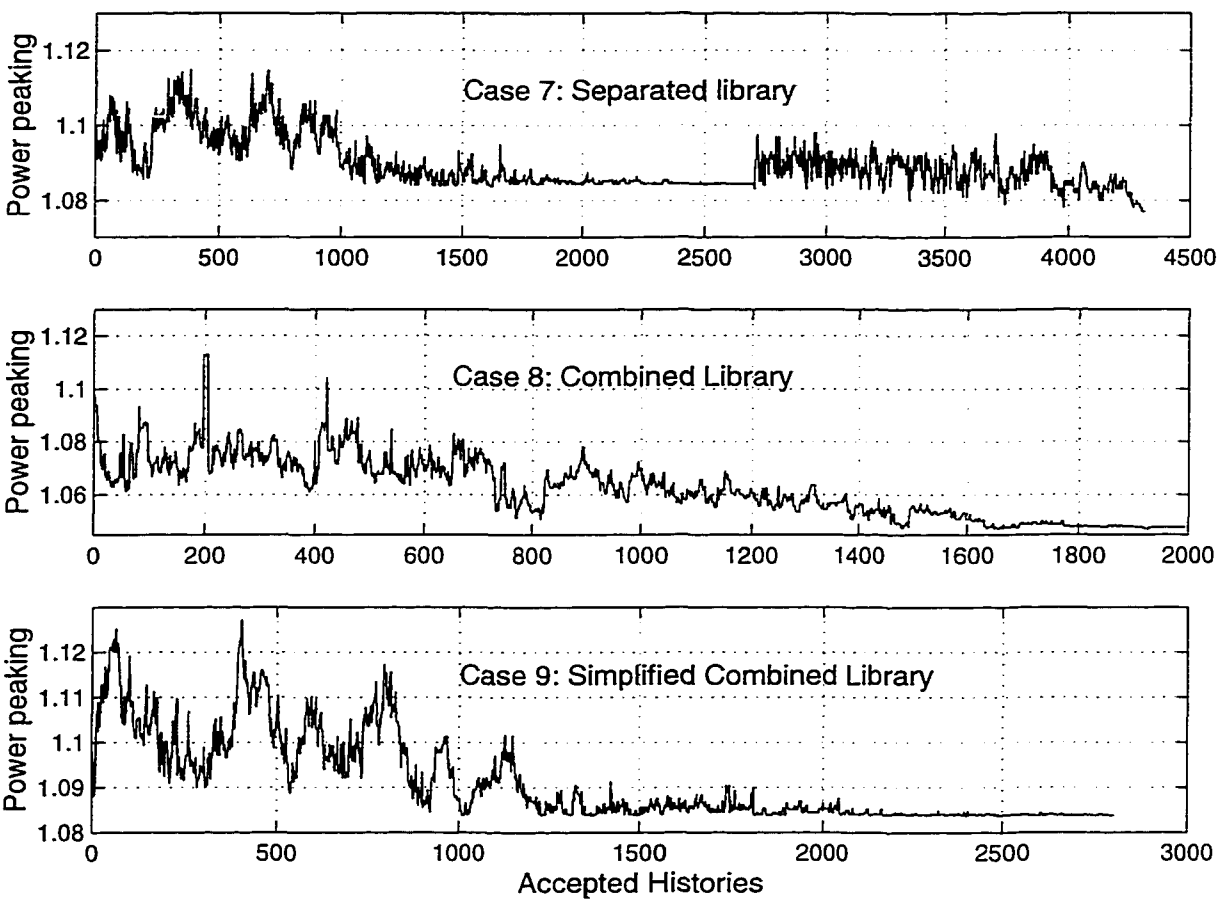


Figure 5.5 PWR power peak minimization: Case C7 through C9

5.4.2 Combined Library Results (BWR case)

Three additional BWR assembly power peaking minimizations, corresponding to the same fuel lattice described within Section 3.7, were carried out to investigate the impact of LSM prediction error upon the final result. The cases are:

C10 BWR assembly power peaking minimization; two cooling cycles: one burnup step (BOL); all histories evaluated with LSM using the separated libraries.

C11 BWR assembly power peaking minimization; one cooling cycle; one burnup step (BOL); all histories evaluated with LSM using the standard combined library.

C12 BWR assembly power peaking minimization; one cooling cycles; one burnup step (BOL); all histories evaluated with LSM using the simplified combined library.

Table 5.4 presents the results got from the BWR lattice employed, while Figure 5.6 illustrates the initial and optimized assemblies and Figure 5.7 shows the progression of the optimizations. Essentially, it is noted that optimization Case C11 led to a similar near-optimum region as Case C10.

Table 5.4 CPU cost, power peaks and error for Case C10 through C12

	CPU cost(*)	initial PP	cycle 1		cycle 2		optimized PP
			PP	MAXKE	PP	MAXKE	
C10	158	1.515	1.397 (1.395)	0.00012	1.396 (1.397)	0.00004	1.396
C11	5941	1.515	1.404 (1.362)	0.00017	N/A	N/A	1.404
C12	157	1.515	1.503 (1.251)	0.00425	N/A	N/A	1.503

(*) CPU Cost Normalized to Units of CASMO-3 Executions;
Figures in parentheses were obtained by LSM;
PP denotes power peaking.

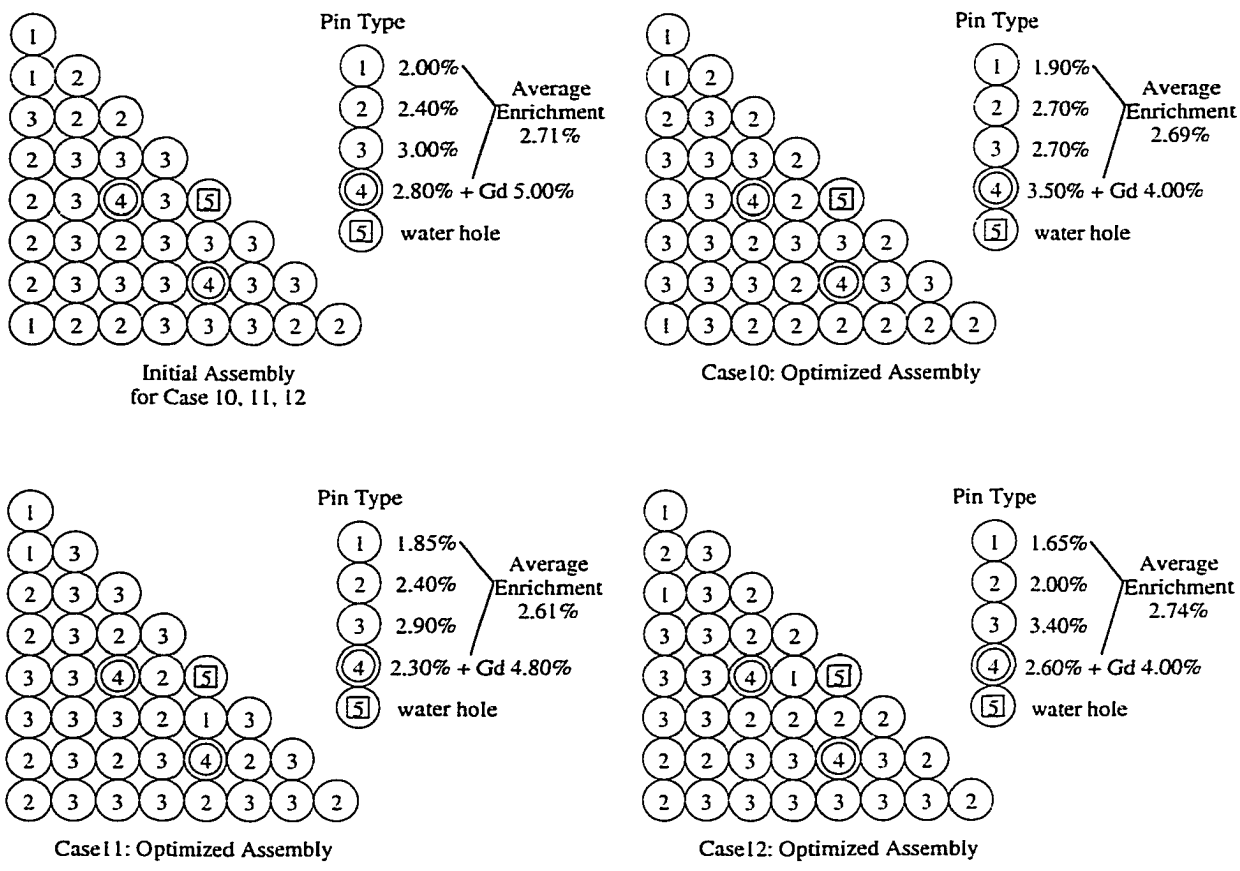


Figure 5.6 Initial and optimized PWR assemblies for Case C10 through C12

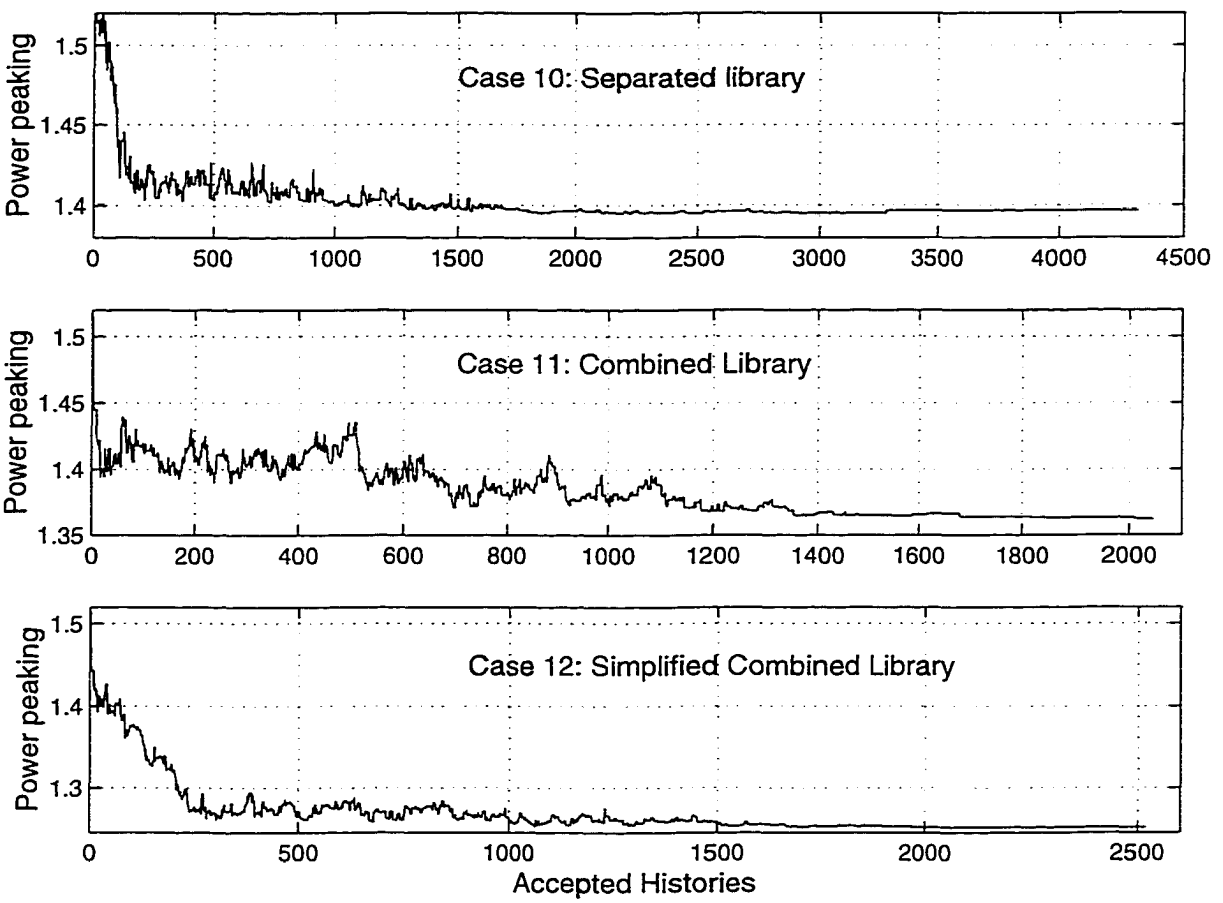


Figure 5.7 BWR power peak minimization Case C10 through C12

However, a more important observation is probably the large tradeoff between prediction error and computational overhead expenses between the two combined library techniques. In fact, the simplified combined library has a very large error in power peaking, while the standard combined library requires a considerable computational overhead to construct. Thus, given the fact that the pure-LSM Case C10 reached a similar near-optimum region without the above-noted errors or overhead costs, at least the BWR results continue to endorse the practical aspects of the separate library approach. Although the PWR combined library approach showed improvements in the previous section, this BWR combined library did not get significant improvements compared to separate libraries, because BWR assembly and the existence of gadolinia result in higher error level. In both BWR and PWR case, the combined simplified library has large and non-conservative errors that could, in fact, misdirect the optimization. This is not out of expectation because the misdirecting of combined simplified library prove the necessity of the separation of material and spatial perturbations.

5.5 Time Saving Study of LSM

The primary objective to develop LSM is to reduce the required running time of the previous version of FORMOSA-L code. The results of time saving due to LSM are shown in this section. The speed-up factor (SF) is based on the average time per history. SF is defined as the ratio of the average time per history of "all-CPM" or "all-CASMO" case (*e.g.* Case C1) to that of LSM case (*e.g.* Case C3a.

Table 5.5 presents the speedup results associated with Case C1, C2 and C3a. With regard to the average time per history the speedups provided by the LSM continue to be remarkable relative to direct evaluations, the hybrid case C2 is about 5 times faster than case C1, while the pure LSM cases C3 (a through d) are roughly 7 times faster. Note that these estimates do include the creation of the linear superposition

libraries within the cases that employ the LSM.

Table 5.6 shows the speedup results associated with Case C4, C5 and C6a. The speedup factor of the hybrid case is about 6.5, while that of the pure LSM case is remarkably as high as 39.4. The higher speedup factor compared with the BWR Case C1, C2 and C3a is because the lower number of pin types in this PWR assembly. Higher pin type numbers makes the material perturbation library size big, not only due to the single perturbations part but also due to the calculations of the second-order cross terms. In practise, the total number of pin types in an assembly is usually 3-5, so the speedup factor is usually high. Typically, the LSM can reduce the running time requirements by at least an order of magnitude.

Table 5.5 Time saving study of Case C1, C2 and C3a

	num. of cycles	total time	num. of attempts	time per history	SF
C1	1	27365.7	4321	6.333	1.0
C2	4	21327.8	15557	1.371	4.6
C3a	4	14731.0	15617	0.943	6.7

Time shown in seconds

Table 5.6 Time saving study of Case C4, C5 and C6a

	num. of cycles	total time	num. of attempts	time per history	SF
C4	1	66182.3	3389	19.529	1.0
C5	4	44582.9	14714	3.030	6.5
C6a	4	7216.0	14540	0.496	39.4

Time shown in seconds

6 CONCLUSIONS

Linear superposition models (LSM) have been developed and fully implemented within the FORMOSA-L computer code, software developed at Iowa State University under sponsorship of the Electric Power Research Center of North Carolina State University, to enable the speedy and accurate estimation of lattice-physics parameters during within-assembly lattice loading optimization. The procedures to create the separated libraries (material library and spatial library) and combined library are presented and several techniques to improve the accuracy have been studied. LSM with linear interpolation, LSM with second-order interpolation and second-order cross-term compensation have been implemented into the FORMOSA-L code and results in a more accurate estimator of the lattice-physics parameters of a perturbed assembly.

A comprehensive error analysis of the LSM has been completed to ultimately assess the limitations of the LSM and to understand the impact of LSM errors upon optimization fidelity. The error sources are identified as the truncation error and the round-off error. For the material perturbations, LSM with interpolation and second-order cross-term compensation approach are actually the outcomes of the error analysis to reduce the truncation error. The side effects of these methods is the decrease of the round-off error due to the decrease of the number of summation terms in the estimation equations. The error of material perturbations library is at a very low level, no matter for PWR or BWR assembly, and no matter with or without gadolinia contents. For the spatial perturbation library and the combined library, BWR assembly has higher error level than PWR, and freezing the gadolinia pins can keep the error at a lower level. In

general, the round-off error can be eliminated from the total error if the source codes of CPM-2 or CASMO-3 are available, like we did in CPM-2M. Combined simplified library has an unacceptably high error, which proves that the separation of material and spatial perturbations is necessary. The future work to improve the accuracy of LSM should focus on one of the two directions: elimination of round-off error and second-order term compensation in the spatial perturbation library.

The separate libraries and the combined library are good models when they are employed in the FORMOSA-L and no major misdirecting of optimization process is observed. The separate libraries approach has lower error level and less computing requirement, but the material and spatial perturbations must be separately performed in alternating cooling cycles, which limits the extent of the search space. The combined library approach overcome the weakness of the separate libraries approach because more search space coverage is obtained due to the simultaneous material and spatial perturbations, which can outweigh the impact of the high error level. In general, both approaches do not show significant loss of fidelity.

It has been shown that the LSM can be used as a fast scoping technique at the beginning of an optimization process, or it can completely replace direct lattice-physics evaluations throughout the entire optimization. The results presented show that the LSM can provide remarkable speedups relative to direct evaluations, which certainly facilitates the pursuit of "real life" assembly lattice loading design problems.

The parallel computing via RPC technique has been studied to further speed up the LSM library creation. Both synchronous and asynchronous implementations have added to the FORMOSA-L code. The test case using two workstations show the speedup has got by this technique. Also it provides a basis for the future work of possibly running multiple Markov chains within simulated annealing.

BIBLIOGRAPHY

Adielson, K. (1997). Hot-Bird: a tool for automated optimization of fuel enrichment design. *Proceedings of the American Nuclear Society topical meeting - Advances in nuclear fuel management II*, *2*, bf 17-31.

Axmann, J. K. (1997). Parallel adaptive evolutionary algorithms for pressurized water reactor reload pattern optimizations. *Nuclear technology*, *119*(3), 276-290.

Hida, K., & Yoshioka, R. (1992). Optimization of axial enrichment and Gadolinia distributions for BWR fuel under control rod programming, (II): Optimization of reload fuel. *Journal of nuclear science and technology*, *29*(4), 313-324.

Hirado, Y., Hida, K., Sakurada, K., & Yamamoto, M. (1997) Optimization of fuel rod enrichment distribution to minimize rod power peaking throughout life within BWR fuel assembly. *Journal of nuclear science and technology*, *34*(1), 5-12.

Kim, H. G., Chang, S. H., & Lee, B. H. (1993). Optimal fuel loading pattern design using an artificial neural network and a fuzzy rule-based system. *Nuclear science and engineering*, *115*(2), 152-163.

Kirkpatrick, S., Gelatt, C. D., & Vecchi, M. P.(1983). Optimization by Simulated Annealing. *Science*, *220*(4598), 671-680.

Kropaczek, D. J. & Turinsky, P. J. (1991). In-core nuclear fuel management optimization for pressurized water reactors utilizing simulated annealing. *Nuclear science and engineering*, *95*(1), 9-32.

Lee, K. G. (1992). Parallel simulated annealing using multiple Markov chains: synchronous and asynchronous implementations. Ph. D. dissertation, Cornell University.

Lim, E. Y. & Leonard A. (1977). Optimal pin enrichment distributions in boiling water reactor fuel bundles. *Nuclear science and engineering*, *64*(2), 694-708.

Mahlers, Yu. P. (1991). Algorithms for solving some problems of the nuclear reactor optimization. *Annals of nuclear energy, 18*(11), 661-664.

Maldonado, G. I., Turinsky, P. J., & Kropaczek, D. J. (1995). Employing nodal generalized perturbation theory for the minimization of feed enrichment during pressurized water reactor in-core nuclear fuel management optimization. *Nuclear science and engineering, 121*(2), 312-325.

Maldonado, G. I., & Zheng, J. (1998). Approximation of lattice-physics parameters via linear superposition. In *Trans. Am. Nucl. Soc., 79*, 317.

Maldonado, G. I., & Zheng, J. (1999). Separability of perturbation within a superposition-based lattice physics model. In *Trans. Am. Nucl. Soc., 80*, 234.

Metropolis, N., & Rosenbluth, A. W. (1953). Equation of state calculations by fast computing machines. *Journal of chemical physics, 21*, 1087.

Parks, G. T. (1990). An intelligent stochastic optimization routine for nuclear fuel cycle design. *Nuclear technology, 89*(2), 233-246.

Parks, G. T. & Knight, M. P. (1990). Pressurized water reactor fuel management using PANTHER. *Nuclear engineering, 37*(4), 104-109.

Pincus, M. (1970). A Monte Carlo method for the approximate solution of certain types of constrained optimization problems. *Operations research, 18*(6), 1225-1233.

Quist, A. J., van Geemert, R., & Hoogenboom, J. E. (1999). Application of nonlinear optimization to reactor core fuel reloading. *Annals of nuclear energy, 26*(5), 423-448.

Siegelmann, H.T., Nissan, & E., Galperin, A. (1997). Novel neural/symbolic hybrid approach to heuristically optimized fuel allocation and automated revision of heuristics in nuclear engineering. *Advances in engineering software, 28*(9), 581-592.

Smuc, T., Pevec, D., & Petrovic, B. (1994). Annealing strategies for loading pattern optimization. *Annals of nuclear energy, 21*(6), 325-336.

Stevens, J. G., Smith, K. S., Rempe, K. R., & Downar, T. J. (1995). Optimization of pressurized water reactor shuffling by simulated annealing with heuristics. *Nuclear science and engineering, 121*(1), 67-88.

Stevens, W. R. (1999). *Unix network programming, Volume 2: Interprocess communications. Second edition.* Upper Saddle River, New Jersey. Prentice Hall PTR.

Yamamoto, A., Noda, H., Ito, N., & Maruyama, T. (1997). INSIGHT: An integrated scoping analysis tool for in-core fuel management of PWR. *Journal of nuclear science and technology, 34*(8), 847-855.