June 1991

LABAN-PEL: A TWO-DIMENSIONAL, MULTIGROUP DIFFUSION, HIGH-ORDER RESPONSE MATRIX CODE

by

E.Z. Müller

ATOMIC ENERGY CORPORATION OF SOUTH AFRICA LIMITED
P O BOX 582, PRETORIA, 0001

ISBN 0 86960 888 6

PEL 309

June 1991

LABAN-PEL: A TWO-DIMENSIONAL, MULTIGROUP DIFFUSION, HIGH-ORDER RESPONSE MATRIX CODE

E.Z. Müller

ATOMIC ENERGY CORPORATION OF SOUTH AFRICA LIMITED

PRETORIA

ABSTRACT

This report describes the capabilites of LABAN–PEL, which is a modified version of the two–dimensional, high–order response matrix code, LABAN, written by Lindahl. The new version extends the capabilities of the original code with regard to the treatment of neutron migration by including an option to utilize full group–to–group diffusion coefficient matrices. In addition, the code has been converted from single to double precision and the necessary routines added to activate its multigroup capability. The coding has also been converted to standard FORTRAN–77 to enhance the portability of the code. Details regarding the input data requirements and calculational options of LABAN–PEL are provided.

CONTENTS

1 INTRODUCTION

The high–order response matrix (RM) method developed by Weiss and Lindahl[1–4] is one of the most accurate coarse–mesh diffusion–theory methods developed to date. The method is based on a partial–current RM formalism in which the spatial shapes of the partial currents on node faces (a node being a spatially homogeneous subdomain defined by a coarse mesh) are approximated by expansions in terms of local sets of base functions. Weiss and Lindahl have developed the method practically in two–dimensional (2–D) Cartesian geometry by using Legendre polynomials for the local base coupled with a semi–analytic Fourier series method for the generation of the local (nodal) RMs. In essence, their method allows solutions of arbitrary accuracy to coarse–mesh problems, the degree of accuracy being determined by the order of the Legendre expansion used for partial currents. Two approaches[2,4] have been used for the implementation of the method, namely the RM source approach and the RM eigenvalue (RME) approach, the latter having been adopted in the LABAN code.[2]

The purpose of this report is to describe a modified version of LABAN which has recently been used to demonstrate[5] the utility of the RME method for determining benchmark quality solutions to 2–D reactor problems. This version, called LABAN–PEL (for the Pelindaba version), incorporates changes which have enhanced the versatility as well as the numerical accuracy of the original code. These changes are:

(i) Addition of routines for the computation of the eigenvalues and eigenvectors of nonsymmetric real matrices in order to make full use of the multigroup capabilities of LABAN.

(ii) Extensions to include nondiagonal diffusion coefficient (D) matrices in order to obtain a method which could be used to solve the multigroup P_1 equations and to study the impact of certain RM–based homogenization procedures[6,7] which may yield full diffusion coefficient matrices.

(iii) Conversion from single to double precision arithmetic to enhance the accuracy of the method (particularly in multigroup applications) as well as the convergence of the eigenvalue search algorithm.

(iv) Conversion to standard FORTRAN–77 for increased code portability.

The first two modifications affect only the local RM calculational part of the code, while the third change has an impact on both the RM generation and the RME solution parts. Without going into too much detail, for which the reader is referred to the original documentation (Ref. 2), these two parts are briefly discussed in Sections 2 and 3. The input data requirements of LABAN–PEL are considered in Section 4 and some remarks with respect to the portability and use of the code are made in Section 5.

2 THE HIGH–ORDER RESPONSE MATRIX EIGENVALUE EQUATIONS

The multigroup diffusion–theory RM method has as its basis the following response functional relations (where the energy variable has been discretized into G energy groups):

$$j^{+}_{ng}(s) = \oint_{S_n} ds' \{ \sum_{g'=1}^{G} G_{ngg'}(s' \rightarrow s) j^{-}_{ng'}(s') \} + j^{ext}_{ng}(s) \quad ; \quad (g=1,2,3,\ldots,G) \qquad (1)$$

In this expression, $j^{+}_{ng}(s)$ is the partial current in energy group g leaving node n at position s on surface S_n, $j^{-}_{ng'}(s')$ is the partial current in group g′ entering the node at position s′ on S_n, $G_{ngg'}(s' \to s)$ is an integral kernel (in the spatial variable) known as the response kernel, and $j^{ext}_{ng}(s)$ is the group g outgoing partial current at position s due to some external source.

Assuming the response kernel $G_{ngg'}(s' \to s)$ to be known, the first step in developing the high–order RM method is to partition the surface S_n into several simpler subsurfaces S_{nk} and then to project the partial currents on a local set of base functions defined on each subsurface. In 2–D Cartesian geometry, Weiss and Lindahl[1] chose the orthogonal Legendre polynomials $\{P_\ell(\xi) |\ 0 \le \ell \le \infty;\ -1 \le \xi \le +1\}$ as the base functions defined at nodal interfaces (four faces per node). Using local coordinates $s \in [-S_{nk}/2, S_{nk}/2]$ on each subsurface S_{nk} and a finite Legendre expansion up to order L (called the order of approximation), this leads to the following expressions:[2]

$$j^{+}_{nk\ell g} \simeq \sum_{k'=1}^{4} \sum_{\ell'=0}^{L} \sum_{g'=1}^{G} G_{nk\ell gk'\ell'g'} j^{-}_{nk'\ell'g'} + j^{ext}_{nk\ell g} \quad , \tag{2}$$

$$j^{\pm}_{nkg}(s) \simeq \sum_{\ell=0}^{L} j^{\pm}_{nk\ell g} v_{nk\ell}(s) \quad , \tag{3}$$

where surface S_{nk} is simply a line segment in 2–D and (with $\delta_{\ell\ell'}$ being the Kronecker delta)

$$v_{nk\ell}(s) = \frac{\sqrt{2\ell+1}}{S_{nk}} P_\ell(\xi) \quad ; \qquad (\xi = \frac{s}{S_{nk}}) \tag{4}$$

$$\int_{S_{nk}} ds\ v_{nk\ell}(s)v_{nk\ell'}(s) = \frac{1}{S_{nk}}\delta_{\ell\ell'} \quad , \tag{5}$$

$$j^{\pm}_{nk\ell g} = S_{nk}\int_{S_{nk}} ds v_{nk\ell}(s)j^{\pm}_{nkg}(s) \quad , \tag{6}$$

$$G_{nk\ell gk'\ell' g'} = S_{nk}\int_{S_{nk}} ds \int_{S_{nk'}} ds'\ v_{nk\ell}(s)G_{ngg'}(s'\to s)v_{nk'\ell'}(s') \quad . \tag{7}$$

Note that $j^{ext}_{nk\ell g}$ is defined in the same manner as $j^{\pm}_{nk\ell g}$ and that the normalization of the base functions was chosen such that expansion coefficient $j^{\pm}_{nk0g}$ is equal to the integrated partial current on face S_{nk}.

The response relation given by Eq. (2) is for individual moments of the outgoing partial current in a specific energy group and could be combined into a matrix expression for all moments and all groups:

$$\vec{j}^{+}_{n} \simeq \hat{G}_n \vec{j}^{-}_{n} + \vec{j}^{ext}_{n} \quad . \tag{8}$$

This expression relates to the local solution for node n. The global solution for N interacting nodes in a compact system can also be written in matrix notation:

$$\underline{\vec{J}}^{+} \simeq \underline{\hat{G}}\ \underline{\vec{J}}^{-} + \underline{\vec{J}}^{ext} \quad . \tag{9}$$

The global response matrix $\underline{\hat{G}}$ is here defined as an N*N block–diagonal matrix with each block representing the local response matrix of a node, i.e. $\underline{\hat{G}}=\mathrm{diag}(\hat{G}_1,\hat{G}_2,\hat{G}_3,...,\hat{G}_N)$, where N is the total number of nodes in the system.

In a compact system each particle emerging from a node n enters a neighbouring node m, or escapes through the outer boundary of the system. This can be written symbolically as

$$\vec{\underline{J}}^{-} \simeq \hat{\underline{H}}\,\vec{\underline{J}}^{+} \quad , \tag{10}$$

where the boundary conditions on the outer boundary of the system are also included in $\hat{\underline{H}}$. (The matrix can be decomposed as $\hat{\underline{H}} = \hat{\underline{H}}_p + \hat{\underline{H}}_a$, where $\hat{\underline{H}}_p$ is purely geometrical and $\hat{\underline{H}}_a$ describes the effects of the boundary conditions.) Then Eq. (9) can be written in the form,

$$\vec{\underline{J}}^{-} = (\hat{\underline{H}}\,\hat{\underline{G}})\vec{\underline{J}}^{-} + \hat{\underline{H}}\,\vec{\underline{J}}^{ext} \quad . \tag{11}$$

In the RME method as implemented in the code LABAN, the local response matrix $\hat{G}_n$ of a node n includes both scattering and fission processes and the external source term dissappears from Eq. (11) to yield the eigenvalue equation,

$$\hat{\underline{R}}\,\vec{\underline{J}}^{-} = \lambda\,\vec{\underline{J}}^{-} \quad , \tag{12}$$

where the eigenvalue λ is required to obtain a stationary solution and $\hat{\underline{R}} = \hat{\underline{H}}\,\hat{\underline{G}}$.

The solution to Eq. (12) is obtained by finding the largest eigenvalue λ_1 and its associated eigenvector $\vec{\underline{J}}_1^{-}$. This largest eigenvalue, which is the spectral radius of $\hat{\underline{R}}$, is known as the "explicit eigenvalue" since any other parameter which may be hidden in the response matrices could be considered as the "implicit eigenvalue" of the problem. Usually the implicit eigenvalue is the "reactivity eigenvalue" which is defined by the "criticality" condition,

$$\lambda_1(k_{eff}) = 1 \quad . \tag{13}$$

Once the eigenvalue problem posed by Eq. (12) (or Eq. (13)) has been solved, one has the option of calculating node–averaged fluxes or even node internal flux distributions using, a current–to–flux local RM $\hat{F}_n$ which is simply computed as a by–product during the evaluation of the current–to–current local RM $\hat{G}_n$, which itself can be partitioned into submatrices representing reflection, transmission and side–transmission. Formally, this does not influence the solution but is considered only as part of the output. However, in practice, the local RM $\hat{F}_n$ is used to check for neutron balance. Any deviation in this balance is then used to adjust the local RM $\hat{G}_n$ to compensate for truncation or round–off errors committed during the course of the calculation of RMs.

2.1 Solution of the RME Equations

In LABAN, Eq. (12) is solved by means of the power iteration method[8,9] to find the eigenvalue of largest modulus, which is called the dominant eigenvalue λ_1. However, since the global response matrix $\hat{\underline{R}}$ is consistently ordered[9] using a red–black checkerboard sweeping of the nodes, it is two–cyclic so that both λ_1 and $-\lambda_1$ are eigenvalues. Hence, the iterative algorithm used is as follows (with n the iteration counter):[2]

$$\begin{aligned}
\vec{\underline{J}}_r^-(n+1) &= \hat{\underline{R}}_b \, \vec{\underline{J}}_b(n) \\
\vec{\underline{Z}}_b^-(n+2) &= \hat{\underline{R}}_r \, \vec{\underline{J}}_r^-(n+1) \qquad (n=0,2,4,\ldots) \\
\vec{\underline{J}}_b^-(n+2) &= \frac{1}{\alpha_{n+2}} \, \hat{\underline{Z}}_b^-(n+2)
\end{aligned} \tag{14}$$

where the subscripts b and r refer to black and red nodes, and $\vec{\underline{J}}_b(0)$ is arbitrarily chosen such that all its zeroth moment components are unity. The normalization constant α_{n+2} is defined such that the Euclidian norm of the vector $\vec{\underline{J}}_b(n+2)$ is equal to unity:

$$\alpha_{n+2} = \|\vec{\underline{Z}}_b(n+2)\|_2 \quad . \tag{15}$$

Then, as $n\to\infty$, $\alpha_{n+2}\to\lambda_1^2$ and $\vec{\underline{J}}_b(n+2)\to\vec{\underline{J}}_{b1}$.

The algorithm can be accelerated as follows:

$$\vec{\underline{J}}_b(n+2) = \vec{\underline{J}}_b(n) + \omega_{n+2}\{\frac{1}{\alpha_{n+2}}\hat{\underline{Z}}_b(n+2) - \vec{\underline{J}}_b(n)\} \quad . \tag{16}$$

In order to use the standard successive overrelaxation (SOR) method,[9] a constant $\omega_{n+2}=\omega$ with $0\leq\omega\leq2$ should be supplied by the user (for $\omega=1$, the unaccelerated Gauss-Seidel algorithm[9] is used). An alternative method known as Aitken's δ^2 method[8,9] can also be used, and is recommended since it does not require an input relaxation parameter. Actually, a modified version of Aitken's δ^2 method is used by which ω_{n+2} is estimated from

$$\omega_{n+2} = \frac{\alpha_n - \alpha_{n-2}}{2\alpha_n - \alpha_{n+2} - \alpha_{n-2}} \quad , \tag{17}$$

and used to extrapolate α_{n+2} as well as $\vec{\underline{J}}_b(n+2)$. That is, in addition to Eq. (16), one also has

$$\alpha_{n+2} = \alpha_n + \omega_{n+2}\{\alpha_{n+2} - \alpha_n\} \quad . \tag{18}$$

This modified method predicts the asymptotic limit to which the predictions of each variable (each component of $\vec{J}_b(n+2)$ as well as α_{n+2}) is tending once successive errors start exhibiting an exponential decay. Thus, after each iteration ω_{n+2} is computed and used only when the solution is in its exponential mode, else $\omega=1$ is used. This decision relies on the following check:

$$|1 - \omega_{n+2}/\omega_n| < \epsilon_\omega \quad . \tag{19}$$

After an extrapolation has been made, $\omega=1$ is used until ω_{n+2} stabilizes again. It has been found that a too coarse criterion ϵ_ω might result in Aitken's acceleration being applied too soon, with the consequence that a converged solution is not obtained. In LABAN-PEL, $\epsilon_\omega=0.02$ has been selected since it seems to work well for a variety of problems. In those cases where this method fails, it is suggested that the standard unaccelerated Gauss-Seidel or an underrelaxation (SOR with $\omega<1$) method be used since the SOR method has also been shown[2] to sometimes diverge for $\omega>1$.

The power iterations, which are called the inner iterations to distinguish them from the k_{eff} iterations known as outer iterations, are terminated when a user-specified maximum number of inner iterations per outer have been exceeded, or upon convergence determined by the criterion

$$|1 - \alpha_{n+2}^{min}/\alpha_{n+2}^{max}| < \epsilon_\lambda \quad , \tag{20}$$

where (with J_{bi} indicating the ith component of $\vec{J}_b$)

$$\alpha_{n+2}^{min} = \min_i |J_{bi}^-(n+2)/J_{bi}^-(n)| \quad , \tag{21}$$

$$\alpha_{n+2}^{max} = \max_i |J_{bi}^-(n+2)/J_{bi}^-(n)| \quad . \tag{22}$$

Upon termination of the inner iterations, the eigenvalue $\lambda_1=\sqrt{\alpha_{n+2}}$, is computed.

2.2 Solution of the Criticality Equation

In order to solve the non-linear problem posed by Eq. (13), λ_1 has to be evaluated as a function of the assumed value of k_{eff}, so that an estimate of k_{eff} which would yield $\lambda_1(k_{eff})=1$ can be made and the process continued until this condition is met. In LABAN, the search is for $\bar{\lambda}(\bar{k})=1$, where $\bar{\lambda}=1/\lambda_1$ and $\bar{k}=1/k_{eff}$, since it has been found[2] that $\bar{\lambda}$ and $\bar{k}$ exhibit a closer linear relationship than λ_1 and k_{eff}. Once at least two assumed values of $\bar{k}$ have been used and the $\bar{\lambda}$ corresponding to these values computed, the method of false positioning (Regula Falsi)[8] can be used to make the third and subsequent estimates of $\bar{k}$ which would give $\bar{\lambda}(\bar{k})=1$:

$$\bar{k}^{(m+1)} = \bar{k}^{(m)} + \frac{\bar{\lambda}^{(m)} - 1}{\bar{\lambda}^{(m)} - \bar{\lambda}^{(m-1)}}[\bar{k}^{(m-1)} - \bar{k}^{(m)}] \qquad (m=1,2,3,\ldots) \tag{23}$$

While the first estimate, $\bar{k}^{(0)}$, is arbitrary and user-specified, the second estimate must be quite good since then the Regula Falsi method would generally converge.[8] Therefore, $\bar{k}^{(1)}$ is computed from neutron balance. The third estimate of $\bar{k}$, i.e. $\bar{k}^{(2)}$, is then obtained by Eq. (23). Instead of simply continuing with this method for further estimates of $\bar{k}^{(m+1)}$, a method based on the fitting of a parabola through the three latest points in the $\bar{k}$-$\bar{\lambda}$ space is used

to find $\lambda(k)=1$ (Regula Falsi fits only a straight line through two points):

$$k^{(m+2)} = k^{(m+1)} - \frac{\lambda^{(m+1)} - 1}{\lambda^{(m+1)} - \lambda^{(m-1)}}\left\{\frac{1 + \lambda^{(m+1)} - \lambda^{(m)} - \lambda^{(m-1)}}{\lambda^{(m+1)} - \lambda^{(m)}} * \left[k^{(m+1)} - k^{(m)}\right] + \frac{\lambda^{(m)} - 1}{\lambda^{(m)} - \lambda^{(m-1)}}\left[k^{(m)} - k^{(m-1)}\right]\right\}$$

$$(m=1,2,3,\ldots) \qquad (24)$$

The advantage of this method is that the convergence of algorithm (24) is nearly quadratic while that of Regula Falsi is[8] of order 1.6. The algorithm is considered converged when

$$|1 - k_{eff}^{(m+2)}/k_{eff}^{(m)}| < \epsilon_k \quad , \qquad (25)$$

$$|1 - \lambda_1^{(m)}| < \epsilon_k \quad , \qquad (26)$$

and fewer than 10 inner iterations were required to converge the inners.

3 EVALUATION OF LOCAL RESPONSE MATRICES

In the high-order RME method as implemented in LABAN, multigroup diffusion-theory RMs for homogeneous rectangular nodes (i.e. the $\hat{G}_n$ in Eq. (8)) are computed semi-analytically, using an adaptation of a technique proposed by Aoki and Shimizu.[10] Essentially, the method determines a rigorous Green's function solution $\phi_{gg'}(x,y)$ to the matrix equation

$$-\hat{D}\nabla^2\hat{\phi}(x,y) + \hat{\Sigma}_R\hat{\phi}(x,y) = \hat{0} \quad , \qquad (27)$$

subject to the boundary conditions

$$\frac{1}{4}\hat{\phi}(x,y) \pm \frac{1}{2}\hat{D}\nabla\hat{\phi}(x,y) = \hat{0} \qquad \text{for } x=\pm\frac{a}{2} \qquad (28)$$

$$\frac{1}{4}\hat{\phi}(x,y) + \frac{1}{2}\hat{D}\nabla\hat{\phi}(x,y) = \hat{0} \qquad \text{for } y=\frac{+b}{2} \qquad (29)$$

$$\frac{1}{4}\hat{\phi}(x,y) - \frac{1}{2}\hat{D}\nabla\hat{\phi}(x,y) = \hat{I}(x) \qquad \text{for } y=\frac{-b}{2} \qquad (30)$$

where $\frac{-a}{2}\leq x\leq\frac{+a}{2}$ and $\frac{-b}{2}\leq y\leq\frac{+b}{2}$ defines the spatial domain of a given rectangular node and $\hat{I}(x)$ is a G*G diagonal matrix with each non-zero entry $I_{gg}(x)$ representing an arbitrarily distributed partial current in a specific energy group g incident through the surface $y=\frac{-b}{2}$ of the node. In other words, with $\hat{D}$ being a matrix of diffusion coefficients (non-diagonal for generality) and $\hat{\Sigma}_R$ the "removal" matrix defined by

$$\hat{\Sigma}_R = \hat{\Sigma}_t - \hat{\Sigma}_s - \frac{1}{k_{eff}}\hat{\Sigma}_f \quad , \qquad (31)$$

where $\hat{\Sigma}_t$, $\hat{\Sigma}_s$, and $\hat{\Sigma}_f$ are the total cross section, scattering, and fission source matrices, $\phi_{gg'}(x,y)$ is the group g flux distribution produced by the partial current $I_{g'g'}(x)$.

The objective is to find an analytic expression for $\phi_{gg'}(x,y)$ from which the outgoing partial current distributions and hence the local response functions (reflection, transmission, side-transmission, etc.) can be obtained. Such an expression (albeit semi-analytic) can be obtained by means of a Fourier series in which $\phi_{gg'}(x,y)$ is sought (using local coordinates $\xi=\frac{2x}{a}$ and $\eta=\frac{2y}{b}$) in the

form[2,10,11]

$$\phi_{gg'}(\xi,\eta) = \sum_{\nu} X_{\nu}(\xi) Y_{gg',\nu}(\eta) C_{g',\nu} \quad . \tag{32}$$

Unfortunately, such a simple form can be made to satisfy Eqs. (27) through (30) only if $\hat{D}$ is diagonal and energy group invariant, specifically with regard to Eq. (28). However, this difficulty can be overcome by replacing the exact boundary conditions of Eq. (28) (i.e. on the node faces transverse to the incident face) by approximate boundary conditions in which $\hat{D}$ is replaced by some arbitrary constant D_0, to obtain an approximate Green's function $\hat{\phi}_a(\xi,\eta)$ which has the form of Eq. (32). This approximate Green's function, called the provisional flux,[2] is then used to determine outgoing partial currents and provisional RMs. The provisional RMs can be utilized to compute the true (exact) RMs by means of the method of error currents[10,11] which has its basis in the fact that the fundamental equation for neutron transport is linear.

Details regarding the method as sketched above and as it has been implemented in the LABAN code can be found in Lindahl's thesis.[2] However, several of the expressions given by Lindahl have been adapted to full diffusion coefficient matrices, and the neccessary modifications implemented in LABAN-PEL. In the following only those expressions which were actually modified in the code are given.

3.1 Modified Expressions Used in the Evaluation of the RMs

The first step in the calculation of the local RMs is the computation of the

"buckling matrix" for each rectangular node corresponding to the latest value of k_{eff}

$$\hat{B} = \hat{D}^{-1}\hat{\Sigma}_R \quad . \tag{33}$$

This matrix is used in the determination of the $Y_{gg',\nu}(y)$ functions in Eq. (32). These functions are evaluated in the form of (dropping the index ν)

$$Y_{gg'}(\eta) = \sum_{j=1}^{2G} S_{gj}(\eta) c_{jg'} \quad , \tag{34}$$

where the integration constants $c_{jg'}$ are determined from G uncoupled systems (one for each g') of linear equations which are also affected by the adoption of a full D matrix treatment (Eq. (9.35) in Ref. 2):

$$\sum_{j=1}^{2G} \left[S_{gj}(\eta) + \frac{4}{b} \sum_{i=1}^{G} D_{gi} \left(\frac{\partial}{\partial \eta} S_{ij}(\eta) \right) \right]_{\eta=+1} c_{jg'} = 0$$

$$(g=1,2,3,\ldots,G) \tag{35}$$

$$\sum_{j=1}^{2G} \left[S_{gj}(\eta) - \frac{4}{b} \sum_{i=1}^{G} D_{gi} \left(\frac{\partial}{\partial \eta} S_{ij}(\eta) \right) \right]_{\eta=-1} c_{jg'} = 4\delta_{gg'}$$

The next step is to compute the provisional RMs, and in this case the provisional side-transmission RM is directly affected by the full D matrix treatment (Eq. (9.43) in Ref. 2):

$$\tilde{B}^{\pm}_{y,\ell g \ell' g'} = \frac{b}{8a}\sqrt{2\ell+1}\,\sqrt{2\ell'+1}\sum_{\nu} t_\nu X_\nu(\pm 1)\langle P_{\ell'}|X_\nu\rangle$$
$$* \left\{ \sum_{j=1}^{G} \left[\delta_{gj} + \frac{1}{D_0} D_{gj} \right] \langle P_\ell | Y_{jg'\nu} \rangle \right\} \quad , \tag{36}$$

where $\langle P_\ell|X_\nu\rangle$ is the ℓ'th Legendre moment of $X(\xi)$ and t_ν is a normalization factor.

The final step involves the computation of the true RMs through the use of the error current RMs, the latter of being affected by the D matrix and evaluated according to (Eq. (9.48) in Ref. 2)

$$\tilde{W}^{\pm}_{y,\ell g \ell' g'} = \sum_{i=1}^{G}\sum_{j=1}^{G} \left\{ P_{gj} Q_{ji} \right\} \tilde{B}^{\pm}_{y,i\ell' g'} \quad , \tag{37}$$

where, using matrix notation,

$$\hat{P} = [D_0 \hat{D}^{-1} - \hat{E}] \quad , \tag{38}$$

$$\hat{Q} = [D_0 \hat{D}^{-1} + \hat{E}]^{-1} \quad , \tag{39}$$

with $\hat{E}$ being the G*G unity matrix.

There is, however, an intermediate step prior to the computation of the error current RMs, which also required modification, namely that of checking for neutron balance of the provisional RMs. The balance to be satisfied is, for each coordinate direction u and each energy group g,

$$\bar{A}_{u,0g\ell'g'} + \bar{C}_{u,0g\ell'g'} + 2\{\bar{B}^{+}_{u,0g\ell'g'} + \bar{W}^{+}_{u,0g\ell'g'}\}$$

$$+ \sum_{g''=1}^{G} \Sigma_{Rgg''} \bar{F}_{u,00g''\ell'g'} = \delta_{gg'}\delta_{0\ell'} \quad . \qquad (40)$$

The first two terms refer to the reflection and transmission contributions to the out–leakage and the third term to the side–transmission leakage (which includes the error currents caused by the violation of the boundary conditions on the transverse nodal faces). Upon substitution of Eq. (37), this condition reduces to

$$\bar{A}_{u,0g\ell'g'} + \bar{C}_{u,0g\ell'g'} + \sum_{g''=1}^{G} 4Q_{gg''} \bar{B}^{+}_{u,0g''\ell'g'}$$

$$+ \sum_{g''=1}^{G} \Sigma_{Rgg''} \bar{F}_{u,00g''\ell'g'} = \delta_{gg'}\delta_{0\ell'} \quad , \qquad (41)$$

where $\hat{Q}$ is defined by Eq. (39). If this balance condition is not satisfied, the residual,

$$\hat{\Delta}_{u,0g\ell'g'} = \hat{A}_{u,0g\ell'g'} + \hat{C}_{u,0g\ell'g'} + \sum_{g''=1}^{G} 4Q_{gg''} \bar{B}^{+}_{u,0g''\ell'g'}$$

$$+ \sum_{g''=1}^{G} \Sigma_{Rgg''} \bar{F}_{u,00g''\ell'g'} - \delta_{gg'}\delta_{0\ell'} \quad , \qquad (42)$$

is used to adjust the provisional side–transmission RM:

$$\bar{B}^{+}_{u,0g\ell'g'} = \bar{B}^{+}_{u,0g\ell'g'} - \sum_{g''=1}^{G} K_{gg''}\bar{\Delta}_{u,0g''\ell'g'} \quad , \tag{43}$$

where $\hat{K} = (4\hat{Q})^{-1}$. The last two expressions are actually the ones which are coded and thus affected by the adoption of full D matrices.

4 LABAN–PEL INPUT DATA DESCRIPTION

The user–specified input requirements for a normal LABAN–PEL run consist of 16 basic input records (see Section 4.1). In the case of continuation cases, where only selected input data parameters are changed, only those records that are affected need to be respecified (see Section 4.2). The facility also exists for reading cross sections from binary files instead of from the standard card input deck (see Section 4.3). Three compulsory logical I/O units are required, namely units 5, 6 and 10 for input data, output (printed) data and a scratch file, respectively. Two additional units may be required depending on the input options (i.e., JFIL and IFLXS). For the Personal Computer (80836 AT) version of the code, the user must also supply a so–called "install" file containing the file and path names of the files connected to the three compulsory logical units (given in the order of units 6,5 and JFIL). This install file has the fixed name of "LABANPEL.INS" and can also be used for the input data on logical unit 5 by simply specifying the file name connected to unit 5 also as LABANPEL.INS. The user is required to activate the statement IVERS=1 between the CPC comments in the main program for the PC version of the code prior to compilation (IVERS=0 should be used for the mainframe version).

4.1 Base Case

The input records (cards) for a normal run are listed below. It should be noted that unless otherwise indicated, FREE FORMAT is assumed.

CARD 1: FORMAT (A5,15A5)

'LABAN'	Keyword to initiate a new LABAN run (for continuation cases, see Section 4.2)
ITITLE	Case description

CARD 2:

NG	Number of energy groups
NNX	Maximum number of nodes in horizontal x–direction
NNY	Maximum number of nodes in vertical y–direction
MIXXSE	Number of cross section sets to be read, including those for mixtures with external/input RMs.
IDIAG	=0 -> diagonal D matrix to be read and used =1 -> full D^{-1} matrix to be read and used =2 -> full D matrix to be read and used
MIXEXT	Number of external response matrix sets
MIXALB	Number of albedo boundary condition sets
KEFF	Initial k_{eff} guess (default=1.0)

CARD 3:

L	Order of approximation for partial currents
LR	Order of approximation for response matrix calculation with LR $\geq$ L (default=L); used specifically to compute more

accurate error current RMs

LQ — Order of approximation for fluxes; used in detailed intranode flux distribution calculations (=0 sufficient if only node average flux is needed)

ICUR — >1 -> P_ℓ (ℓ=0,...,L) partial currents printed
=1 -> P_0 partial currents printed
=0 -> No partial currents printed

IFLUX — =1 -> Flux RM in group representation printed
=0 -> Not printed

IBUCK — =1 -> Flux RM in buckling representation printed
=0 -> Not printed

NGRID — >0 -> For each node the point fluxes at NGRID*NGRID points are printed if IFLUX = 1.
=0 -> Not printed

IADJNT — =0 -> Forward solution sought
=1 -> Adjoint solution sought

JFIL — >0 -> k_{eff}, number of nodes, nodal powers and nodal volumes saved on logical unit JFIL (JFIL may not equal 5, 6 or 10)
≤0 -> Not saved

CARD 4:

NNORM — =0 -> Total core power normalized to CPOW. If CPOW=0., core average power is normalized to 1.0
=1 -> Total absorptions in core normalized to 1.

CPOW — Core linear power density in watts/cm. This should correspond to the thermal power contribution of the core

volume being modeled.

EPF — Liberated energy in joules/fission (default=$3.2*10^{-11}$)

KACC — =0 -> SOR acceleration of power iterations
=1 -> Geometrical extrapolation of power iterations (Aitken's δ^2 method)

MAXIN — Maximum number of power iterations (default=150)

MAXOUT — Maximum number of outer k_{eff} iterations (default=5)

EPSR — Accuracy requirement for RM calculations; determines number of terms in Fourier expansion (default=10^{-5})

EPSI — Convergence criterion for power iterations (default=10^{-5})

EPSK — Convergence criterion for k_{eff} iterations (default=10^{-5})

OMEGA — SOR relaxation parameter (default=1.0)

CARD 5:

NDX(I) — Horizontal subdivision of nodes with I=1,NNX (from left to right)

CARD 6:

NDY(J) — Vertical subdivision of nodes with J=1,NNY (from top to bottom)

CARD 7:

HX(I) — Node widths in cm with I=1,NNX

CARD 8:

HY(J) — Node heights in cm with J=1,NNY

CARD 9: FORMAT (20I4)

MAP(I,J) Mixture allocation map with I=1,NNY+2 and J=1,NNX+2 (left to right and top to bottom, usually given as NNY+2 cards); mixture types range from 1 to MIXXSE. Albedo types range from -1 to -MIXALB. Zeros or blanks are used for fringe nodes.

Cards 10 and 11 are repeated MIXXSE times

CARD 10: FORMAT (3I5,2X,A32)

MIXTP Mixture number (ranges from 1 to MIXXSE; must be in chronological order)

IFLXS Logical unit number of cross section file (|IFLXS| may not equal 6, 10 or JFIL)

If IFLXS=5, then the cross sections are read from the same file as the other data (see Card 11)

If IFLXS≠5 and positive, then the cross sections are read from a STANDARD file (see Section 4.3.1 for format)

If IFLXS≠5 and negative, then the cross sections are read from a LABAN file (see Section 4.3.2 for format)

IDENT Cross section set identifier on unit IFLXS (must be the same as MIXTP if IFLXS=5)

LTITLE Title for this cross section set. For the PC version, this is the filename (including the directory path) of the file on unit IFLXS.

CARD 11: (Required only if IFLXS=5)

XSEC Cross sections to be given in the following order:

D_1, Σ_{a1}, $\nu\Sigma_{f1}$, χ_1, $\Sigma_{s1g'}$, Σ_{f1}, f_1, $M_{1g'}$,
D_2, Σ_{a2}, $\nu\Sigma_{f2}$, χ_2, $\Sigma_{s2g'}$, Σ_{f2}, f_2, $M_{2g'}$,

. . .

. . .

. . .

D_{NG}, Σ_{aNG}, $\nu\Sigma_{fNG}$, χ_{NG}, $\Sigma_{sNGg'}$, Σ_{fNG}, f_{NG}, $M_{NGg'}$,
with $g'=1,2,...,NG$. The f_g are so-called group heterogeneity factors as obtained from e.g. Koebke's[12] homogenization method and are used only in flux editing since they are assumed already included in the cross sections (usually they are simply specified as unity). The $M_{gg'}$ are the elements of the full D^{-1} or the full D matrix (depending on IDIAG).

Cards 12 and 13 are repeated MIXALB times (in the order corresponding to that given in the mixture allocation map)

CARD 12: FORMAT (A5)

IIBC ='BLACK' -> Black (vacuum) boundary condition
='WHITE' -> Reflective boundary condition
='ALBED' -> Albedo boundary condition to be read on Card 13

CARD 13: (Required only if IIBC='ALBED')

ALB Albedo matrix given rowwise:
A_{11}, A_{12}, ..., A_{1NG}

. . .

. . .

. . .

A_{NG1}, A_{NG2}, ..., A_{NGNG}

Cards 14 and 15 are repeated MIXEXT times

CARD 14: FORMAT (3I5,5X,A60)

MIXRES	Mixture number for external response matrix (ranges from MIXXSE-MIXEXT+1 to MIXXSE)
L0	Approximation order (may differ from L of Card 3)
NANT	=1 -> Only one set of A, T, B, F matrices to be given (i.e. HX=HY) =2 -> Two sets of A, T, B, F matrices to be given (i.e. HX≠HY)
NTITLE	Response matrix identification (any text)

CARD 15: (4 records repeated NANT times)

A(I,J)	Reflection matrix given rowwise with I,J=NG*(L0+1)
T(I,J)	Transmission matrix given rowwise with I,J=NG*(L0+1)
$B^+(I,J)$	Side-transmission matrix given rowwise with I,J=NG*(L0+1)
F(I,J)	Total flux matrix given rowwise with I,J=NG*(L0+1)

CARD 16: FORMAT (A5)

'END'	Keyword to terminate LABAN input (must be the very last card of any LABAN input deck)

4.2 Continuation Cases

Two types of continuation cases are allowed:

Case 1 – NEW L: Orders of approximation and iteration parameters can be varied from those specified for the base case. This case requires that only Cards 1, 3, 4, 14, and 15 be respecified, with the keyword 'NEWL' given on Card 1.

Case 2 – NEW X: Orders of approximation, iteration parameters as well as node subdivisions can be varied from those specified for the base case. This case requires that only Cards 1, 3, 4, 5, 6, 14, and 15 be respecified, with the keyword 'NEWX' given on Card 1.

4.3 Formats of Input Cross Section Files

Two types of cross section libraries can be read by LABAN-PEL, the formats of each of which is described below.

4.3.1 STANDARD File

This is a single-precision binary file consisting of three general records and 1+2*LG records per cross section set. The general records are:

RECORD 1:

LG	Number of energy groups

RECORD 2:

CHI	$\chi_1, \chi_2, \chi_3, \cdots, \chi_{LG}$

RECORD 3: Dummy record (not used at present).

The 1+2*LG records per cross section set are:

RECORD 1:

ID — Cross section set identifier

RECORD 2:

XSEC1 — Cross sections for group 1 given in the following order:

D_1, Σ_{a1}, $\nu\Sigma_{f1}$, Σ_{f1}, f_1

RECORD 3:

SCAT1 — Scattering to group 1:

Σ_{s11}, Σ_{s12}, ..., Σ_{s1LG}

RECORD 4:

XSEC2 — Cross sections for group 2 given in the following order:

D_2, Σ_{a2}, $\nu\Sigma_{f2}$, Σ_{f2}, f_2

RECORD 5:

SCAT2 — Scattering to group 2:

Σ_{s21}, Σ_{s22}, ..., Σ_{s2LG}

. . .

. . .

. . .

RECORD 2*LG:

XSECLG — Cross sections for group LG given in the following order:

D_{LG}, Σ_{aLG}, $\nu\Sigma_{fLG}$, Σ_{fLG}, f_{LG}

RECORD 1+2*LG:

SCATLG Scattering to group LG:

Σ_{sLG1}, Σ_{sLG2}, ..., Σ_{sLGLG}

where the f_g are the group heterogeneity factors used only in flux editing.

It should be noted that this type of cross section library can only be used in calculations with diagonal D matrices.

4.3.2 LABAN File

This is a single-precision binary file on logical unit IFLXS with the following two records per cross section set:

RECORD 1:

ID Cross section set identifier

LG Number of energy groups for this cross section set

JDIAG =0 -> Only diagonal D matrix available for this cross section set

=1 -> Diagonal D matrix as well as full D^{-1} matrix available for this cross section set

=2 -> Diagonal D matrix as well as full D matrix available for this cross section set

RECORD 2:

XSEC Cross sections given in the following order:

D_1, D_2, ..., D_{LG},

Σ_{a1}, Σ_{a2}, ..., Σ_{aLG}

$\nu\Sigma_{f1}$, $\nu\Sigma_{f2}$, ..., $\nu\Sigma_{fLG}$

χ_1, χ_2, ..., χ_{LG}

Σ_{s11}, Σ_{s21}, ..., Σ_{sLG1}

Σ_{s12}, Σ_{s22}, ..., Σ_{sLG2}

. . .

. . .

. . .

Σ_{s1LG}, Σ_{s2LG}, ..., Σ_{sLGLG}

Σ_{f1}, Σ_{f2}, ..., Σ_{fLG}

f_1, f_2, ..., f_{LG}

M_{11}, M_{21}, ..., M_{LG1}

M_{12}, M_{22}, ..., M_{LG2}

. . .

. . .

. . .

M_{1LG}, M_{2LG}, ..., M_{LGLG},

where the f_g are the group heterogeneity factors used only in flux editing and the $M_{gg'}$ are the elements of the full D^{-1} or the full D matrix (depending on JDIAG).

5. COMMENTS

Although LABAN-PEL performs several consistency checks of the input data, the user is still responsible for ensuring consistency of some of the data. For instance, if the cross sections are read from the same file as the other data (i.e.

IFLXS=5), the user should ascertain that the D matrix data is consistent with the IDIAG option specified. It is important for the user to bear in mind that the value of IDIAG applies to all cross section sets (all mixtures) and that erroneous results will be obtained if this convention is not strictly applied. (Note that the group diffusion coefficients can be assigned dummy values when IDIAG≠0 since LABAN-PEL computes these from the column sums of $\hat{D}^{-1}$. The D_g's are required for the estimation of the D_0 mentioned in Section 3.) Another common source of error relates to the specification of the mixture allocation map and/or the node sizes. If LABANPEL–PEL aborts with a message that a cross section set could not be found, the user should first check for compatibilty of the number of energy groups (NG) and/or the cross section set identifier (IDENT).

The description of the input data/options given in Section 4 should be adequate, except perhaps for two of those. The first is the IBUCK=1 option which prints the nodal fluxes in the so–called buckling representation. The buckling representation for a given node is defined as that representation (basis) in which the buckling matrix $\hat{B}$ for the node (see Eq. (33)) is diagonalized. The fluxes for a node can be written in this representation by means of the transformation,[13]

$$\langle \beta_i | \phi \rangle = \sum_{g=1}^{G} \langle \beta_i | g \rangle \langle g | \phi \rangle \qquad (i=1,2,\ldots,G) \qquad (44)$$

where $\langle g|\phi\rangle$ is the group g flux and the $\langle \beta_i | g\rangle$ are the elements of the matrix $\hat{U}^{-1}$, which is the inverse of the modal matrix $\hat{U}$ containing the eigenvectors of the matrix $\hat{B}$ as its columns. Conversely, the group fluxes can then be written as (with $\langle g|\beta_i\rangle$ being the elements of $\hat{U}$)

$$\langle g|\phi\rangle = \sum_{i=1}^{G} \langle g|\beta_i\rangle\langle\bar{\beta}_i|\phi\rangle \quad , \qquad (g=1,2,\ldots,G) \qquad (45)$$

which expresses the flux in a node in terms of its modal components. Hence, the flux in the buckling representation, $\langle\bar{\beta}_i|\phi\rangle$, is often called the i'th modal flux component. Thus, the option IBUCK=1 allows one to study the magnitudes of the various modal flux components. At present, this option is operational only in 2-group cases.

The second aspect which may require clarification is that of the meaning of the group heterogeneity factors (f_g's) which are required as input cross section data. These factors are similar to Koebke's[12] "simplified equivalence theory" heterogeneity factors or the so-called[7] "normalized generalized equivalence theory" discontinuity factors which are divided into conventional flux weighted cross sections to determine equivalent diffusion theory parameters. In LABAN-PEL, it is assumed that the cross section data already contain this correction and the heterogeneity factors are simply used to obtain the true (physical) flux values from the computed (equivalent) flux values through division of the latter by the corresponding heterogeneity factors. For most applications, these factors can be assigned unity values since they do not affect the computed values of the k_{eff} or the nodal powers.

A comment with regard to the methods available for the acceleration of the inner (power) iterations is in order here. It has been found that, for some problems, the power iterations can diverge with both the SOR method and Aitken's δ^2 method. In such cases the user is advised to use the SOR method with the relaxation parameter (OMEGA) given a value of unity or less (i.e.

underrelaxation). Also, it is recommended that sufficient inner (power) iterations be used per outer in order to determine the explicit eigenvalue accurately, since this could reduce the number of outer iterations required for determining the k_{eff}. This is particularly important when several different mixtures are specified for a problem since the evaluation of the local response matrices per outer iteration may then be more costly than the inner iterations.

As far as the portability of LABAN–PEL is concerned, it should be noted that the code has been written in standard FORTRAN 77, and that it has been implemented on both a mainframe computer (IBM 370 series) and on an 80386 AT personal computer. A single version of the code is used on both types of computer platforms, with only the single parameter IVERS in the program main (indicated between CPC comments) that needs to be changed. The mainframe version works with IVERS=0 while the PC version requires IVERS=1. In the case of the PC version, the user must also specify a so–called install file with the generic name LABANPEL.INS, on which the the file names of the printed output (logical unit 6), the input data file (logical unit 5) and the power distribution output file (logical unit JFIL) are given. These file names should be given as CHARACTER*32 names on consecutive records (in the exact order noted above). The user is further cautioned that logical unit 10 is reserved for a scratch file which contains all the data necessary to restart a LABANPEL run (for such purposes the status of logical unit 10 should be changed accordingly).

Finally, it might be informative to mention some of the basic performance characteristics of the high–order RME method such as that the number of power iterations necessary to find the explicit eigenvalue is quite insensitive to the Legendre order but increases roughly linearly with the number of nodes.[2] The

computational time (CPU time) increases faster than linear but somewhat slower than quadratic with increasing Legendre order.[2] However, the convergence of the method with increasing Legendre order is faster than that with increasing number of nodes, and it is thus better suited to coarse–mesh applications.[2]

REFERENCES

1. Z. WEISS and S–Ö. LINDAHL, Nucl. Sci. Eng., 58, 166 (1975).

2. G.S–Ö. LINDAHL, "Multi–Dimensional Response Matrix Method," PhD Thesis, Kansas State University (1976).

3. Z. WEISS, Nucl. Sci. Eng., 63, 457 (1977).

4. S–Ö. LINDAHL and Z. WEISS, Adv. Nucl. Sci. Techn., 13, 736 (1981).

5. E.Z. MÜLLER and Z. WEISS, "Benchmarking with the High–Order Response Matrix Method," Ann. Nucl. Energy, 18, 535, 1991.

6. R.A. BONALUMI, Nucl. Sci. Eng., 77, 219 (1981).

7. E.Z. MÜLLER, Nucl.Sci. Eng., 103, 359 (1989).

8. E. ISAACSON and H.B. KELLER, "Analysis of Numerical Methods," John Wiley & Sons, Inc., 1966.

9. A. JENNINGS, "Matrix Computation for Engineers and Scientists," John Wiley & Sons, Ltd., 1977.

10. K. AOKI and A. SHIMIZU, J. Nucl. Sci. Techn., 2. 149 (1965).

11. A. SHIMIZU and K. AOKI, "Application of Invariant Embedding to Reactor Physics," Academic Press, 1972.

12. K.KOEBKE, "Advances in Homogenization and Dehomogenization," Proc. Int. Topl. Mtg. Advances in Mathematical Methods for the Solution of Nuclear Engineering Problems, Munich, FRG, 27-29 April, 1981.

13. E.Z. MÜLLER, "A Brief Introduction to the Dirac Bra-Ket Notation for use in Nodal Methods," Internal Report RTE01-2/2-056, Reactor Theory Division, Atomic Energy Corporation of South Africa (1989).

ISBN 0 86960 888 6