TECHNICAL REPORT
IGE-380

The Brisingr theory and user guide

A. Hébert

Institut de génie nucléaire
Département de génie mécanique
École Polytechnique de Montréal
August 5, 2024

Contents

Contents . . . ii
1 Introduction . . . 1
2 The nodal expansion method . . . 2
2.1 Solution of the nodal expansion equations over a 1D traverse . . . 5
2.2 The quadratic transverse leakage approximation with the NEM . . . 9
2.3 The linear transformation technique . . . 10
2.4 The analytic nodal method in 2–D Cartesian geometry . . . 12
2.5 The nodal update procedure . . . 18
2.5.1 NEM update procedure . . . 18
2.5.2 ANM update procedure . . . 21
2.5.3 Update of the drift coefficients . . . 22
2.6 Intranodal flux reconstruction . . . 22
3 Input data specifications . . . 25
3.1 Syntactic rules for input data specifications . . . 25
3.2 The global input structure . . . 25
3.3 The `BRIF:` module . . . 26
3.3.1 Data input for module `BRIF:` . . . 27
3.4 The `NODSPH:` module . . . 29
3.4.1 Data input for module `NODSPH:` . . . 31
4 Examples of input data files . . . 33
4.1 IAEA-3D benchmark . . . 33
5 The Brisingr Package . . . 38
References . . . 41
Index . . . 41

1 Introduction

This technical report describes the theory and user guide of Brisingr[1], a solution technique of the neutron diffusion equation integrated in the Version5 code system.[2] Brisingr is an implementation of the *nodal expansion method*[3] (NEM) and of the *analytic nodal method*[4] (ANM), used to integrate the coupling equations between the homogeneous nodes of the finite difference diffusion operator with centered meshes and to solve the flux equations. This presentation is limited to a regular 1D/2D/3D Cartesian domain made of parallelepipeds.

2 The nodal expansion method

The principle of this method is to consider the shape of the neutron flux in each node as a superposition of basic shapes and not as a linear function such as those appearing in the classic finite difference approach. In order to avoid solving a too complex problem in which the coefficients of the basic shapes would be the unknowns, the problem is split into two:

- A finite difference problem in which the coupling coefficients are modified so as to produce currents with surfaces identical to those which would have been obtained if the flux had a nonlinear shape in the node;
- A nodal problem proper in which the coefficients of the basic shapes are determined.

The solution of the first problem, carried out globally on the whole domain, gives the average flux per node (data of the second problem). The solution of the second problem, carried out independently by pairs of nodes, gives the corrections to the coupling coefficients (data of the first problem). During the convergence process, the currents on each of the interfaces of the core nodes are re-evaluated and used to make a correction to the "finite difference" type diffusion operator. When the current is stabilized and the finite difference iteration is converged, the process is stopped.

We first consider a Cartesian one-dimensional (1D) traverse made from the assembly of many infinite slabs with parabolic transverse leakage. The 1D heterogeneous configuration corresponds to the case where the neutron flux is a function of a unique spatial variable. These cases can be solved analytically, whatever the type of conditions imposed at boundaries. Introduction of a transverse leakage term $L_g^y(x)+L_g^z(x)$ (unit: n/(cm^3·s)) is required to generalize the NEM to 2D and 3D cases.

Introducing *transverse integrated leakage terms*, Eq. (5.12) of Ref. 7 simplifies to a 1D equation where the nuclear properties of the reactor are only a function of the independent variable x. This transverse integrated equation is written

$$\frac{d}{dx}J_g(x)+\Sigma_{\mathrm{r},g}(x)\,\phi_g(x)+L_g^y(x)+L_g^z(x)\;=\;Q_g^\diamond(x) \tag{2.1}$$

with the current defined as

$$J_g(x)=-D_g(x)\frac{d\phi_g}{dx} \tag{2.2}$$

and the source defined as

$$Q_g^\diamond(x)=\sum_{\substack{h=1\\h\neq g}}^{G}\Sigma_{\mathrm{s},g\leftarrow h}(x)\,\phi_h(x)+\frac{\chi_g(x)}{K_{\mathrm{eff}}}\sum_{h=1}^{G}\nu\Sigma_{\mathrm{f},h}(x)\,\phi_h(x). \tag{2.3}$$

The boundary conditions are either a zero-flux condition ($\phi_g(x)=0$) or an albedo condition written

$$\mp D_g(x)\,\frac{d\phi_g}{dx}+\frac{1}{2}\,\frac{1-\beta_g(x)}{1+\beta_g(x)}\,\phi_g(x)=0 \tag{2.4}$$

where the sign "$-$" or "$+$" is used for a left ($x=x_{1/2}$) or a right boundary ($x=x_{I+1/2}$), respectively.

Each slab is assumed to be homogeneous, so that the corresponding nuclear properties $D_g(x)$, $\Sigma_{\mathrm{r},g}(x)$, $\Sigma_{\mathrm{s},g\leftarrow h}(x)$, $\chi_g(x)$ and $\nu\Sigma_{\mathrm{f},h}(x)$ are piecewise continuous. As shown in Fig. 1, the reactor domain is divided into I nodes of indices $1\leq i\leq I$, in such a way that the nuclear properties in node i are constant and equal to $D_{i,g}$, $\Sigma_{\mathrm{r},i,g}$, $\Sigma_{\mathrm{s},i,g\leftarrow h}$, $\chi_{i,g}$ and $\nu\Sigma_{\mathrm{f},i,h}$.

Using this geometric representation, the *transverse integrated leakage terms* are written

$$\begin{aligned}
L_g^y(x) &= -\frac{1}{\Delta y_j\Delta z_k}\int_{y_{j-1/2}}^{y_{j+1/2}}dy\int_{z_{k-1/2}}^{z_{k+1/2}}dz\,D_g(x,y,z)\frac{\partial^2}{\partial y^2}\phi_g(x,y,z)\\
&= -\frac{D_{i,j,k,g}}{\Delta y_j}\left[\frac{\partial}{\partial y}\phi_g(x_i,y_{j+1/2},z_k)-\frac{\partial}{\partial y}\phi_g(x_i,y_{j-1/2},z_k)\right]\\
&= \frac{1}{\Delta y_j}\left[J_g(x_i,y_{j+1/2},z_k)-J_g(x_i,y_{j-1/2},z_k)\right]
\end{aligned} \tag{2.5}$$

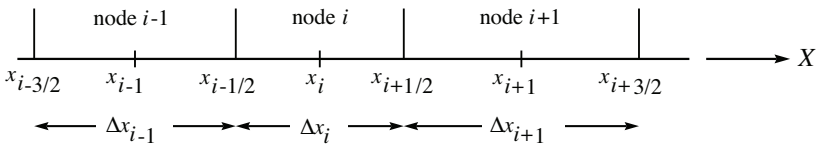

Figure 1: Spatial discretization.

and

$$\begin{aligned} L_g^z(x) &= -\frac{1}{\Delta y_j \Delta z_k}\int_{y_{j-1/2}}^{y_{j+1/2}} dy \int_{z_{k-1/2}}^{z_{k+1/2}} dz\, D_g(x,y,z)\frac{\partial^2}{\partial z^2}\phi_g(x,y,z) \\ &= -\frac{D_{i,j,k,g}}{\Delta z_k}\left[\frac{\partial}{\partial z}\phi_g(x_i,y_j,z_{k+1/2}) - \frac{\partial}{\partial y}\phi_g(x_i,y_j,z_{k-1/2})\right] \\ &= \frac{1}{\Delta z_k}\left[J_g(x_i,y_j,z_{k+1/2}) - J_g(x_i,y_j,z_{k-1/2})\right] \ . \end{aligned} \tag{2.6}$$

Smith introduced a *quadratic leakage approximation* in the nodal schemes leading to the version that is now currently used in legacy codes[4]. Smith proposed to use polynomial coefficients preserving the average leakage terms over three adjacent nodes. Such an approximation can be constructed for node (i,j), in the X–direction, using $\bar{L}^y_{i-1,g}$, $\bar{L}^y_{i,g}$ and $\bar{L}^y_{i+1,g}$. First consider the transverse leakage term $L^y_g(x)$ along the X-axis for the leakage in the Y direction. Assuming C_2 continuity of $L^y_g(x)$ over three consecutive nodes, this approximation is written

$$L^y_g(x) = \sum_{i=0}^{2} b_i\, x^i \tag{2.7}$$

where coefficients b_i are obtained so that

$$\frac{1}{\Delta x_{i-1}}\int_{x_{i-3/2}}^{x_{i-1/2}} dx\, L^y_g(x) = \bar{L}^y_{i-1,g} \quad , \quad \frac{1}{\Delta x_i}\int_{x_{i-1/2}}^{x_{i+1/2}} dx\, L^y_g(x) = \bar{L}^y_{i,g}$$

and

$$\frac{1}{\Delta x_{i+1}}\int_{x_{i+1/2}}^{x_{i+3/2}} dx\, L^y_g(x) = \bar{L}^y_{i+1,g} \quad , \tag{2.8}$$

and where $\bar{L}^y_{i,g}$ is the nodal averaged transverse leakage recovered from the previous nodal update iteration. The detailed processing of transverse leakage terms will be described in Sect. 2.2.

Equation (2.1) can be written in such a way as to be valid in node i:

$$\frac{d}{dx}J_g(x) + \Sigma_{\mathrm{r},i,g}\,\phi_g(x) + L^y_g(x) + L^z_g(x) \ = \ Q^\diamond_g(x) \ = \ \sum_{\substack{h=1\\h\neq g}}^{G}\Sigma_{\mathrm{s},i,g\leftarrow h}\,\phi_h(x) + \frac{\chi_{i,g}}{K_{\mathrm{eff}}}\sum_{h=1}^{G}\nu\Sigma_{\mathrm{f},i,h}\,\phi_h(x)$$
$$\text{if } x_{i-1/2} < x < x_{i+1/2} \tag{2.9}$$

with

$$J_g(x) = -D_{i,g}\frac{d}{dx}\phi_g(x). \tag{2.10}$$

Integrating Eq. (2.9) over node i, we obtain

$$\frac{J_{i+1/2,g} - J_{i-1/2,g}}{\Delta x_i} + \Sigma_{\mathrm{r},i,g}\,\bar{\phi}_{i,g} + \bar{L}^{yz}_{i,g} = \bar{Q}^\diamond_{i,g} \tag{2.11}$$

where $J_{i\pm1/2,g} \equiv J_g(x_{i\pm1/2})$ and

$$\bar{L}^{yz}_{i,g} = \frac{1}{\Delta x_i}\int_{x_{i-1/2}}^{x_{i+1/2}} dx\left[L^y_g(x) + L^z_g(x)\right]. \tag{2.12}$$

Neutron currents can be broken down into 2 parts by showing the usual finite difference component and the difference between the real current and its finite difference approximation. For the left right boundary, we write

$$J_{i+1/2,g} = \begin{cases} -D^+_{i,g}\left(\bar{\phi}_{i+1,g} - \bar{\phi}_{i,g}\right) - \tilde{D}^+_{i,g}\left(\bar{\phi}_{i+1,g} + \bar{\phi}_{i,g}\right) & \text{if } i < I \\ D^+_{i,g}\bar{\phi}_{i,g} - \tilde{D}^+_{i,g}\bar{\phi}_{i,g} & \text{if } i = I \end{cases} \tag{2.13}$$

and for the left node boundary:

$$J_{i-1/2,g} = \begin{cases} -D^-_{i,g}\left(\bar{\phi}_{i,g} - \bar{\phi}_{i-1,g}\right) - \tilde{D}^-_{i,g}\left(\bar{\phi}_{i,g} + \bar{\phi}_{i-1,g}\right) & \text{if } i > 1 \\ -D^-_{i,g}\bar{\phi}_{i,g} - \tilde{D}^-_{i,g}\bar{\phi}_{i,g} & \text{if } i = 1 \end{cases} \tag{2.14}$$

where $\tilde{D}^{\pm}_{i,g}$ is a *drift coefficient* that will be obtained from the *nodal update procedure* of Sect. 2.5 and where

$$\begin{aligned} D^+_{i,g} &= \frac{2D_{i,g}D_{i+1,g}}{D_{i,g}\Delta x_{i+1} + D_{i+1,g}\Delta x_i} \\ D^-_{i,g} &= \frac{2D_{i-1,g}D_{i,g}}{D_{i-1,g}\Delta x_i + D_{i,g}\Delta x_{i-1}} \,. \end{aligned} \tag{2.15}$$

Both coefficients $D^{\pm}_{i,g}$ and $\tilde{D}^{\pm}_{i,g}$ are non-dimensional quantities.

In case where abscissa $x_{i\pm1/2}$ is located on the boundary of the domain, we write

$$D^{\pm}_{i,g} = \begin{cases} \dfrac{2D_{i,g}}{\Delta x_i} & \text{for a zero flux BC} \\ \dfrac{2D_{i,g}\Lambda_{i,g}}{2D_{i,g} + \Lambda_{i,g}\Delta x_i} & \text{for an albedo BC} \end{cases} \tag{2.16}$$

where $\Lambda_{i,g}$ is the albedo function defined as

$$\Lambda = \frac{1}{2}\frac{1-\beta}{1+\beta}. \tag{2.17}$$

Eqs. (2.15) and (2.16) are identical to *mesh centered finite difference* (MCFD) coefficients of Sect. 5.2.2 in Ref. 7. Values $D^{\pm}_{i,g}$ and $\tilde{D}^{\pm}_{i,g}$ are stored in variables `FDMNCC` and `CNCC` of Brisingr, respectively. Values $\bar{\phi}_{i,g}$ are stored in variables `CMFD_sol`.

The system of equations thus obtained depends only on the average flux in the node. Its resolution can be carried out using the method with two iteration levels (internal and external) conventionally used in finite difference calculation codes. The internal iteration calculates the flux in equilibrium with a given source, while the external iteration determines the source created by the flux.

The re-evaluation of the neutron currents is made by solving, for each interface separating 2 adjacent nodes, a one-dimensional diffusion equation. This equation is obtained by integrating the three-dimensional equation on the directions transverse to the direction of reevaluation of the currents, assuming that the currents obey Fick's law. For example, for the case of an interface between nodes i and $i+1$, and for each of the nodes, these equations are

For the left node ($x_{i-1/2} < x < x_{i+1/2}$):

$$-D_{i,g}\frac{d^2}{dx^2}\phi_{i,g}(x) + \Sigma_{\mathrm{r},g}(x)\,\phi_{i,g}(x) + L^y_{i,g}(x) + L^z_{i,g}(x) = Q^{\diamond}_{i,g}(x), \tag{2.18}$$

For the right node ($x_{i+1/2} < x < x_{i+3/2}$):

$$-D_{i+1,g}\frac{d^2}{dx^2}\phi_{i+1,g}(x) + \Sigma_{\mathrm{r},g}(x)\,\phi_{i+1,g}(x) + L^y_{i+1,g}(x) + L^z_{i+1,g}(x) = Q^{\diamond}_{i+1,g}(x). \tag{2.19}$$

The flux continuity relation is written

$$f^+_{i,g}\,\phi_{i,g}(x_{i+1/2}) = f^-_{i+1,g}\,\phi_{i+1,g}(x_{i+1/2}) \tag{2.20}$$

where $f^+_{i,g}$ and $f^-_{i+1,g}$ are the flux discontinuity factors for nodes i and $i+1$ at $x_{i+1/2}$.

The current continuity relation

$$J(x_{i+1/2}) = -D_{i,g}\frac{d}{dx}\phi_{i,g}(x_{i+1/2}) = -D_{i+1,g}\frac{d}{dx}\phi_{i+1,g}(x_{i+1/2}) \tag{2.21}$$

complete the system of equations.

2.1 Solution of the nodal expansion equations over a 1D traverse

The nodal expansion method is based on an expansion of $\phi_{i,g}(x)$ of the form

$$\phi_{i,g}(u) = \sum_{\ell=0}^{4} a_{\ell,i,g}\,p_\ell(u) \tag{2.22}$$

and an expansion of the transverse leakage terms $L^{yz}_{i,g}(u)$ of the form

$$L^{yz}_{i,g}(u) = \bar{L}^{yz}_{i,g} + \rho^{yz}_{i,g,1}\,p_1(u) + \rho^{yz}_{i,g,2}\,p_2(u) \tag{2.23}$$

where the trial function are defined as[5]

$$\begin{aligned}
p_0(u) &= 1\\
p_1(u) &= u\\
p_2(u) &= 3u^2 - \frac{1}{4}\\
p_3(u) &= \left(u^2 - \frac{1}{4}\right)u\\
p_4(u) &= \left(u^2 - \frac{1}{4}\right)\left(u^2 - \frac{1}{20}\right)
\end{aligned} \tag{2.24}$$

and where u is the reduced coordinate defined over the support $-1/2 < u < 1/2$ as

$$u = \frac{x}{\Delta x_i} - \frac{1}{2}. \tag{2.25}$$

In the case where the discrepancy between the spectral indices of two nodes are higher than a fixed threshold, the order 3 and 4 trial functions are replaced by

$$\begin{aligned}
p_3(u) &= \sinh(\eta u)\\
p_4(u) &= \cosh(\eta u) - \int_{-1/2}^{1/2} du\,\cosh(\eta u) \;=\; \cosh(\eta u) - \frac{2}{\eta}\sinh(\eta/2)
\end{aligned} \tag{2.26}$$

with

$$\eta_{i,g} = \Delta x_i\sqrt{\frac{\Sigma_{\mathrm{r},i,g}}{D_{i,g}}} \tag{2.27}$$

where we assume that $\Sigma_{\mathrm{r},i,g} \geq 0$.

All expansion functions with $\ell > 0$ have zero averages. The coefficient $a_{0,i,g}$ is therefore the averaged flux $\bar{\phi}_{i,g}$.

Equation (2.18) is written in terms of reduced variable $-1/2 < u < 1/2$ as

$$R_{i,g}(u) = -\frac{D_{i,g}}{\Delta x_i^2}\frac{d^2}{du^2}\phi_{i,g}(u) + \Sigma_{\mathrm{r},i,g}\,\phi_{i,g}(u) + L^{yz}_{i,g}(u) - Q^{\diamond}_{i,g}(u) = 0. \tag{2.28}$$

The expansion in Eq. (2.22) being an approximation of the real solution, the application of the operator of diffusion to this function, leads to a function of u called remainder or residue defined as

$$\int_{-1/2}^{1/2} du\, p_\ell(u) R_{i,g}(u) = 0 \tag{2.29}$$

where $\ell = 0,\ 1,$ or 2 and $1 \leq g \leq G$.

We substitute Eq. (2.22) in Eq. (2.28). We multiply this expression by weight functions $p_0(u)$ to $p_2(u)$ and integrate over node i using Eq. (2.29). We obtain three relations on each node by forcing the residuals $R_{i,g}(u)$ to be projected on the trial functions defined in Eqs. (2.24) or (2.26). The operation leads to a system whose unknowns are the expansion coefficients. The three weights residual relations corresponding to a pure polynomial expansion base are[1]:

$$-\frac{2D_{i,g}}{\Delta x_i^2}\left(3a_{2,i,g}+\frac{a_{4,i,g}}{5}\right)+\Sigma_{\mathrm{r},i,g}\,a_{0,i,g}+\bar{L}^{yz}_{i,g}=\sum_{\substack{h=1\\h\neq g}}^{G}\Sigma_{\mathrm{s},i,g\leftarrow h}\,a_{0,i,h}+\frac{\chi_{i,g}}{K_{\mathrm{eff}}}\sum_{h=1}^{G}\nu\Sigma_{\mathrm{f},i,h}\,a_{0,i,h}, \tag{2.30}$$

$$\begin{aligned}-\frac{D_{i,g}}{2\Delta x_i^2}a_{3,i,g}+\frac{\Sigma_{\mathrm{r},i,g}}{12}\left(a_{1,i,g}-\frac{a_{3,i,g}}{10}\right)+\frac{\rho^{yz}_{i,g,1}}{12} &= \sum_{\substack{h=1\\h\neq g}}^{G}\frac{\Sigma_{\mathrm{s},i,g\leftarrow h}}{12}\left(a_{1,i,h}-\frac{a_{3,i,h}}{10}\right)\\ &+\frac{\chi_{i,g}}{K_{\mathrm{eff}}}\sum_{h=1}^{G}\frac{\nu\Sigma_{\mathrm{f},i,h}}{12}\left(a_{1,i,h}-\frac{a_{3,i,h}}{10}\right),\end{aligned} \tag{2.31}$$

and

$$\begin{aligned}-\frac{D_{i,g}}{5\Delta x_i^2}a_{4,i,g}+\frac{\Sigma_{\mathrm{r},i,g}}{20}\left(a_{2,i,g}-\frac{a_{4,i,g}}{35}\right)+\frac{\rho^{yz}_{i,g,2}}{20} &= \sum_{\substack{h=1\\h\neq g}}^{G}\frac{\Sigma_{\mathrm{s},i,g\leftarrow h}}{20}\left(a_{2,i,h}-\frac{a_{4,i,h}}{35}\right)\\ &+\frac{\chi_{i,g}}{K_{\mathrm{eff}}}\sum_{h=1}^{G}\frac{\nu\Sigma_{\mathrm{f},i,h}}{20}\left(a_{2,i,h}-\frac{a_{4,i,h}}{35}\right).\end{aligned} \tag{2.32}$$

The corresponding weight residual relations with hyperbolic basis functions from Eqs. (2.26) are

$$\begin{aligned}-\frac{D_{i,g}}{\Delta x_i^2}\left[6a_{2,i,g}+\alpha_0(\eta_{i,g})a_{4,i,g}\right]+\Sigma_{\mathrm{r},i,g}\,a_{0,i,g} &+ \bar{L}^{yz}_{i,g}=\sum_{\substack{h=1\\h\neq g}}^{G}\Sigma_{\mathrm{s},i,g\leftarrow h}\,a_{0,i,h}\\ &+\frac{\chi_{i,g}}{K_{\mathrm{eff}}}\sum_{h=1}^{G}\nu\Sigma_{\mathrm{f},i,h}\,a_{0,i,h},\end{aligned} \tag{2.33}$$

$$\begin{aligned}\frac{\Sigma_{\mathrm{r},i,g}}{12}\,a_{1,i,g} &+ \frac{\rho^{yz}_{i,g,1}}{12}=\sum_{\substack{h=1\\h\neq g}}^{G}\Sigma_{\mathrm{s},i,g\leftarrow h}\left[\frac{a_{1,i,h}}{12}+\frac{\alpha_1(\eta_{i,h})}{\eta_{i,h}^2}a_{3,i,h}\right]\\ &+\frac{\chi_{i,g}}{K_{\mathrm{eff}}}\sum_{h=1}^{G}\nu\Sigma_{\mathrm{f},i,h}\left[\frac{a_{1,i,h}}{12}+\frac{\alpha_1(\eta_{i,h})}{\eta_{i,h}^2}a_{3,i,h}\right]\end{aligned} \tag{2.34}$$

and

$$\begin{aligned}\frac{\Sigma_{\mathrm{r},i,g}}{20}\,a_{2,i,g} &+ \frac{\rho^{yz}_{i,g,2}}{20}=\sum_{\substack{h=1\\h\neq g}}^{G}\Sigma_{\mathrm{s},i,g\leftarrow h}\left[\frac{a_{2,i,h}}{20}+\frac{\alpha_2(\eta_{i,h})}{\eta_{i,h}^2}a_{4,i,h}\right]\\ &+\frac{\chi_{i,g}}{K_{\mathrm{eff}}}\sum_{h=1}^{G}\nu\Sigma_{\mathrm{f},i,h}\left[\frac{a_{2,i,h}}{20}+\frac{\alpha_2(\eta_{i,h})}{\eta_{i,h}^2}a_{4,i,h}\right]\end{aligned} \tag{2.35}$$

[1]The code Brisingr defines coefficients `quad%o11` $= 1/12$, `quad%o13` $= -1/120$, `quad%o22` $= 1/20$ and `quad%o24` $= -1/700$

where

$$\begin{aligned}\alpha_0(\eta) &= 2\eta\sinh(\eta/2)\\ \alpha_1(\eta) &= \eta\cosh(\eta/2) - 2\sinh(\eta/2)\\ \alpha_2(\eta) &= \left(\frac{12}{\eta}+\eta\right)\sinh(\eta/2) - 6\cosh(\eta/2)\end{aligned} \tag{2.36}$$

and where we used a simplification from equation

$$\frac{D_{i,g}}{\Delta x_i^2} = \frac{\Sigma_{\mathrm{r},i,g}}{\eta_{i,g}^2}. \tag{2.37}$$

In the case of a pure polynomial expansion base, interface fluxes in node i can be obtained from Eqs. (2.47) and (2.49). They are written

$$\phi^+_{i-1/2,g} = a_{0,i,g} - \frac{a_{1,i,g}}{2} + \frac{a_{2,i,g}}{2} \tag{2.38}$$

and

$$\phi^-_{i+1/2,g} = a_{0,i,g} + \frac{a_{1,i,g}}{2} + \frac{a_{2,i,g}}{2} \tag{2.39}$$

where $a_{0,i,g} = \bar{\phi}_{i,g}$.

The corresponding relations with hyperbolic basis functions from Eqs. (2.26) are

$$\phi^+_{i-1/2,g} = a_{0,i,g} - \frac{a_{1,i,g}}{2} + \frac{a_{2,i,g}}{2} - \sinh(\eta_{i,g}/2)a_{3,i,g} + \frac{\alpha_1(\eta_{i,g})}{\eta_{i,g}}a_{4,i,g} \tag{2.40}$$

and

$$\phi^-_{i+1/2,g} = a_{0,i,g} + \frac{a_{1,i,g}}{2} + \frac{a_{2,i,g}}{2} + \sinh(\eta_{i,g}/2)a_{3,i,g} + \frac{\alpha_1(\eta_{i,g})}{\eta_{i,g}}a_{4,i,g} \tag{2.41}$$

Interface net currents in node i can be obtained from Eqs. (2.47) and (2.49). In the case of a pure polynomial expansion base, they are written

$$J_{i-1/2,g} = -\frac{D_{i,g}}{\Delta x_i}\left(a_{1,i,g} - 3a_{2,i,g} + \frac{a_{3,i,g}}{2} - \frac{a_{4,i,g}}{5}\right) \tag{2.42}$$

and

$$J_{i+1/2,g} = -\frac{D_{i,g}}{\Delta x_i}\left(a_{1,i,g} + 3a_{2,i,g} + \frac{a_{3,i,g}}{2} + \frac{a_{4,i,g}}{5}\right). \tag{2.43}$$

The corresponding relations with hyperbolic basis functions from Eqs. (2.26) are

$$J_{i-1/2,g} = -\frac{D_{i,g}}{\Delta x_i}\left[a_{1,i,g} - 3a_{2,i,g} + \eta_{i,g}\cosh(\eta_{i,g}/2)a_{3,i,g} - \eta_{i,g}\sinh(\eta_{i,g}/2)a_{4,i,g}\right] \tag{2.44}$$

and

$$J_{i+1/2,g} = -\frac{D_{i,g}}{\Delta x_i}\left[a_{1,i,g} + 3a_{2,i,g} + \eta_{i,g}\cosh(\eta_{i,g}/2)a_{3,i,g} + \eta_{i,g}\sinh(\eta_{i,g}/2)a_{4,i,g}\right] \tag{2.45}$$

The flux and current continuity relations at surface $x_{i+1/2}$ corresponding to a pure polynomial expansion base are obtained from Eqs. (2.20) and (2.21) as

$$f^{x+}_{i,g}\left(a_{0,i,g} + \frac{a_{1,i,g}}{2} + \frac{a_{2,i,g}}{2}\right) = f^{x-}_{i+1,g}\left(a_{0,i+1,g} - \frac{a_{1,i+1,g}}{2} + \frac{a_{2,i+1,g}}{2}\right) \tag{2.46}$$

and

$$\frac{D_{i,g}}{\Delta x_i}\left(a_{1,i,g} + 3a_{2,i,g} + \frac{a_{3,i,g}}{2} + \frac{a_{4,i,g}}{5}\right) = \frac{D_{i+1,g}}{\Delta x_{i+1}}\left(a_{1,i+1,g} - 3a_{2,i+1,g} + \frac{a_{3,i+1,g}}{2} - \frac{a_{4,i+1,g}}{5}\right) \tag{2.47}$$

where $f^{x+}_{i,g}$ and $f^{x-}_{i+1,g}$ are the *flux discontinuity factors* imposed on surface $x_{i+1/2}$.

The corresponding weight residual relations with hyperbolic basis functions from Eqs. (2.26) are

$$f_{i,g}^{x+}\left[a_{0,i,g} + \frac{a_{1,i,g}}{2} + \frac{a_{2,i,g}}{2} + \sinh(\eta_{i,g}/2)a_{3,i,g} + \frac{\alpha_1(\eta_{i,g})}{\eta_{i,g}}a_{4,i,g}\right] = f_{i+1,g}^{x-}\left[a_{0,i+1,g} - \frac{a_{1,i+1,g}}{2} + \frac{a_{2,i+1,g}}{2} - \sinh(\eta_{i+1,g}/2)a_{3,i+1,g} + \frac{\alpha_1(\eta_{i+1,g})}{\eta_{i+1,g}}a_{4,i+1,g}\right] \tag{2.48}$$

and

$$\frac{D_{i,g}}{\Delta x_i}\left[a_{1,i,g} + 3a_{2,i,g} + \eta_{i,g}\cosh(\eta_{i,g}/2)a_{3,i,g} + \eta_{i,g}\sinh(\eta_{i,g}/2)a_{4,i,g}\right] = \frac{D_{i+1,g}}{\Delta x_{i+1}}\left[a_{1,i+1,g} - 3a_{2,i+1,g} + \eta_{i+1,g}\cosh(\eta_{i+1,g}/2)a_{3,i+1,g} - \eta_{i+1,g}\sinh(\eta_{i+1,g}/2)a_{4,i+1,g}\right] \tag{2.49}$$

The zero flux boundary conditions at surfaces $x_{-1/2}$ and $x_{I+1/2}$ corresponding to a pure polynomial expansion base are:

$$a_{0,1,g} - \frac{a_{1,1,g}}{2} + \frac{a_{2,1,g}}{2} = 0 \tag{2.50}$$

and

$$a_{0,I,g} + \frac{a_{1,I,g}}{2} + \frac{a_{2,I,g}}{2} = 0. \tag{2.51}$$

The corresponding weight residual relations with hyperbolic basis functions from Eqs. (2.26) are

$$a_{0,1,g} - \frac{a_{1,1,g}}{2} + \frac{a_{2,1,g}}{2} - \sinh(\eta_{1,g}/2)a_{3,1,g} + \frac{\alpha_1(\eta_{1,g})}{\eta_{1,g}}a_{4,1,g} = 0 \tag{2.52}$$

and

$$a_{0,I,g} + \frac{a_{1,I,g}}{2} + \frac{a_{2,I,g}}{2} + \sinh(\eta_{I,g}/2)a_{3,I,g} + \frac{\alpha_1(\eta_{I,g})}{\eta_{I,g}}a_{4,I,g} = 0 \tag{2.53}$$

The albedo boundary conditions at surfaces $x_{-1/2}$ and $x_{I+1/2}$ corresponding to a pure polynomial expansion base are:

$$(\Lambda_g^{x-})a_{0,1,g} - \left(\frac{\Lambda_g^{x-}}{2} + \frac{D_{1,g}}{\Delta x_1}\right)a_{1,1,g} + \left(\frac{\Lambda_g^{x-}}{2} + \frac{3D_{1,g}}{\Delta x_1}\right)a_{2,1,g} - \frac{D_{1,g}}{2\Delta x_1}a_{3,1,g} + \frac{D_{1,g}}{5\Delta x_1}a_{4,1,g} = 0 \tag{2.54}$$

and

$$(\Lambda_g^{x+})a_{0,I,g} + \left(\frac{\Lambda_g^{x+}}{2} + \frac{D_{I,g}}{\Delta x_I}\right)a_{1,I,g} + \left(\frac{\Lambda_g^{x+}}{2} + \frac{3D_{I,g}}{\Delta x_I}\right)a_{2,I,g} + \frac{D_{I,g}}{2\Delta x_I}a_{3,I,g} + \frac{D_{I,g}}{5\Delta x_I}a_{4,I,g} = 0 \tag{2.55}$$

where the albedo function Λ is expressed as a function of the boundary albedo β using Eq. (2.17).

If a reflector model is being used, the information available at boundary $x_{I+1/2}$ are the reference heterogeneous boundary flux $\phi^*_{I+1/2,g}$, the reference net boundary current $J^*_{I+1/2,g}$ and the boundary discontinuity factor $f^+_{I,g}$. In this case, the right albedo factor is written

$$\Lambda_g^{x+} = \frac{J^*_{I+1/2,g} f^{x+}_{I,g}}{\phi^*_{I+1/2,g}}. \tag{2.56}$$

The corresponding weight residual relations with hyperbolic basis functions from Eqs. (2.26) are

$$\begin{aligned}(\Lambda_g^{x-})a_{0,1,g} &- \left(\frac{\Lambda_g^{x-}}{2}+\frac{D_{1,g}}{\Delta x_1}\right)a_{1,1,g}+\left(\frac{\Lambda_g^{x-}}{2}+\frac{3D_{1,g}}{\Delta x_1}\right)a_{2,1,g}\\ &- \left(\Lambda_g^{x-}\sinh(\eta_{1,g}/2)+\frac{D_{1,g}}{\Delta x_1}\eta_{1,g}\cosh(\eta_{1,g}/2)\right)a_{3,1,g}\\ &+ \left(\Lambda_g^{x-}\frac{\alpha_1(\eta_{1,g})}{\eta_{1,g}}+\frac{D_{1,g}}{\Delta x_1}\eta_{1,g}\sinh(\eta_{1,g}/2)\right)a_{4,1,g}=0 \end{aligned} \tag{2.57}$$

and

$$\begin{aligned}(\Lambda_g^{x+})a_{0,I,g} &+ \left(\frac{\Lambda_g^{x+}}{2}+\frac{D_{I,g}}{\Delta x_I}\right)a_{1,I,g}+\left(\frac{\Lambda_g^{x+}}{2}+\frac{3D_{I,g}}{\Delta x_I}\right)a_{2,I,g}\\ &+ \left(\Lambda_g^{x+}\sinh(\eta_{I,g}/2)+\frac{D_{I,g}}{\Delta x_I}\eta_{I,g}\cosh(\eta_{I,g}/2)\right)a_{3,I,g}\\ &+ \left(\Lambda_g^{x+}\frac{\alpha_1(\eta_{I,g})}{\eta_{I,g}}+\frac{D_{I,g}}{\Delta x_I}\eta_{I,g}\sinh(\eta_{I,g}/2)\right)a_{4,I,g}=0. \end{aligned} \tag{2.58}$$

The $X-$directed *node leakage* $\bar{L}_{i,g}^x$ is finally obtained as

$$\bar{L}_{i,g}^x = J_{i+1/2,g}-J_{i-1/2,g} = \begin{cases} -\frac{2D_{i,g}}{\Delta x_i}\left(3a_{2,i,g}+a_{4,i,g}/5\right)\ , & \text{if pure polynomials;}\\ -\frac{2D_{i,g}}{\Delta x_i}\left[3a_{2,i,g}+\eta_{i,g}\sinh(\eta_{i,g}/2)a_{4,i,g}\right], & \text{otherwise .}\end{cases} \tag{2.59}$$

2.2 The quadratic transverse leakage approximation with the NEM

One important point which needs to be addressed before Eq. (2.9) can be solved are the transverse leakage terms $L_g^y(x)$ and $L_g^z(x)$. Their spatial dependency is unknown, so their shape must be approximated. The most popular approximation in nodal schemes is the quadratic transverse leakage approximation. For example, the X-direction spatial dependence of the transverse leakage is approximated by Eq. (2.23). The adjustment of coefficients $\rho_{i,g,1}^{yz}$ and $\rho_{i,g,2}^{yz}$ is made so as to preserve the node average leakages of nodes $i-1$, i and $i+1$.

The first three basis functions $p_\ell(u)$ are orthogonal on interval $-1/2\le u\le 1/2$, so that the quadratic leakage coefficients $\rho_{i,g,1}^{yz}$ and $\rho_{i,g,2}^{yz}$ of Eq. (2.23) are obtained as

$$\begin{aligned}\left\langle L_{i,g}^{yz}(u)\,p_1(u)\right\rangle &= \rho_{i,g,1}^{yz}\left\langle p_1(u)\,p_1(u)\right\rangle = \frac{\rho_{i,g,1}^{yz}}{12}\\ \left\langle L_{i,g}^{yz}(u)\,p_2(u)\right\rangle &= \rho_{i,g,2}^{yz}\left\langle p_2(u)\,p_2(u)\right\rangle = \frac{\rho_{i,g,2}^{yz}}{20}. \end{aligned} \tag{2.60}$$

These coefficients can be obtained with the help of the following symbolic Matlab script:

```
syms u
A=[1, -dxm/2, dxm^2/3 ; 1, dx/2, dx^2/3 ;
   1, (2*dx+dxp)/2, (3*dx^2+3*dx*dxp+dxp^2)/3];
coef=A\[Lxm ; Lx ; Lxp];
polyu=symfun(coef(1)+coef(2)*(u+1/2)*dx+coef(3)*((u+1/2)*dx)^2,u);
p0=symfun(1,u);
p1=symfun(u,u);
p2=symfun(3*u^2-1/4,u);
dummy=simplify(int(p0*polyu,-1/2,1/2))
rho1=12*simplify(int(p1*polyu,-1/2,1/2))
rho2=20*simplify(int(p2*polyu,-1/2,1/2))
```

where the variable `coef` contains the coefficients b_i of Eq. (2.7) and the variable `polyu` contains the function $L_{i,g}^{yz}(u)$ in Eq. (2.60). The result of this script is written

$$\begin{aligned}\rho_{i,g,1} &= \frac{\Delta x_i}{g_i}\Big[\left(\bar{L}_{i+1,g}-\bar{L}_{i,g}\right)\left(\Delta x_i+2\Delta x_{i-1}\right)\left(\Delta x_i+\Delta x_{i-1}\right)\\ &\qquad +\left(\bar{L}_{i,g}-\bar{L}_{i-1,g}\right)\left(\Delta x_i+2\Delta x_{i+1}\right)\left(\Delta x_i+\Delta x_{i+1}\right)\Big] \end{aligned} \tag{2.61}$$

$$\rho_{i,g,2} = \frac{\Delta x_i^2}{g_i}\Big[\left(\bar{L}_{i+1,g} - \bar{L}_{i,g}\right)\left(\Delta x_i + \Delta x_{i-1}\right) + \left(\bar{L}_{i-1,g} - \bar{L}_{i,g}\right)\left(\Delta x_i + \Delta x_{i+1}\right)\Big] \tag{2.62}$$

with

$$g_i = \left(\Delta x_i + \Delta x_{i+1}\right)\left(\Delta x_i + \Delta x_{i-1}\right)\left(\Delta x_{i-1} + \Delta x_i + \Delta x_{i+1}\right) \tag{2.63}$$

where $1 < i < I$. In the particular case where $i = 1$ or $i = I$, one can use a linear leakage approximation on two adjacent nodes, or stay with a quadratic leakage approximation, using nodes $1 \leq i \leq 3$ or $I - 2 \leq i \leq I$.

2.3 The linear transformation technique

A *linear transformation technique* is required with the *analytic nodal method* (ANM). Our presentation is limited to 2D Cartesian geometries, in order to simplify equations.[6] The multigroup formulation of the steady-state 2D neutron diffusion equation is written

$$\begin{aligned} -\boldsymbol{\nabla}\cdot D_g(\boldsymbol{r})\boldsymbol{\nabla}\phi_g(\boldsymbol{r}) \;+\; & \Sigma_{\mathrm{r},g}(\boldsymbol{r})\phi_g(\boldsymbol{r}) \\ = \;& \sum_{\substack{h=1\\h\neq g}}^{G} \Sigma_{g\leftarrow h}(\boldsymbol{r})\,\phi_h(\boldsymbol{r}) + \frac{\chi_g(\boldsymbol{r})}{K_{\mathrm{eff}}}\sum_{h=1}^{G}\nu\Sigma_{\mathrm{f}h}(\boldsymbol{r})\,\phi_h(\boldsymbol{r}) \end{aligned} \tag{2.64}$$

where

G	=	total number of energy groups
K_{eff}	=	effective multiplication factor
$\phi_g(\boldsymbol{r})$	=	neutron flux in group g
$D_g(\boldsymbol{r})$	=	diffusion coefficient in group g
$\Sigma_{\mathrm{r},g}(\boldsymbol{r})$	=	macroscopic removal cross section in group g
$\Sigma_{g\leftarrow h}(\boldsymbol{r})$	=	macroscopic scattering cross section from group h toward group g
$\chi_g(\boldsymbol{r})$	=	fission spectrum in group g
$\nu\Sigma_{\mathrm{f}h}(\boldsymbol{r})$	=	product of the macroscopic fission cross section by the average number of neutrons emitted per fission in group h.

The boundary condition is either a *zero flux* ($\phi_g(\boldsymbol{r}) = 0$ if $\boldsymbol{r} \in \partial W_i$) or an *albedo boundary condition* written as

$$D_g(\boldsymbol{r})\boldsymbol{\nabla}\phi_g(\boldsymbol{r})\cdot\boldsymbol{N}(\boldsymbol{r}) + \frac{1}{2}\frac{1-\beta(\boldsymbol{r})}{1+\beta(\boldsymbol{r})}\,\phi_g(\boldsymbol{r}) = 0 \quad \text{if } \boldsymbol{r}\in\partial W_i \tag{2.65}$$

where $\beta(\boldsymbol{r})$ is the albedo at surfacic point $\boldsymbol{r}$ and ∂W_i are the fraction of ∂V where the zero-flux or albedo boundary condition is applied.

The Cartesian domain is partitioned into rectangular nodes (i, j), as depicted in Fig. 2. The nuclear properties are assumed uniform over each node. Equation (2.64) can be written in such a way to be valid on node (i, j) as

$$\begin{aligned} -D_{i,j,g}\frac{\partial^2\phi_g}{\partial x^2} \;-\; & D_{i,j,g}\frac{\partial^2\phi_g}{\partial y^2} \;+\; \Sigma_{\mathrm{r},i,j,g}\,\phi_g(x,y) \\ = \;& \sum_{\substack{h=1\\h\neq g}}^{G}\Sigma_{i,j,g\leftarrow h}\,\phi_h(x,y) + \frac{\chi_{i,j,g}}{K_{\mathrm{eff}}}\sum_{h=1}^{G}\nu\Sigma_{\mathrm{f},i,j,h}\,\phi_h(x,y) \end{aligned} \tag{2.66}$$

where $x_{i-1/2} < x < x_{i+1/2}$ and $y_{j-1/2} < y < y_{j+1/2}$.

At this point, we introduce the *linear transformation technique*, in order to uncouple the energy groups. Equation (2.66) is first rewritten in matrix form as

$$\frac{\partial^2}{\partial x^2}\boldsymbol{\Phi}(x,y) + \frac{\partial^2}{\partial y^2}\boldsymbol{\Phi}(x,y) + \mathbb{F}_{i,j}\boldsymbol{\Phi}(x,y) = \boldsymbol{0} \tag{2.67}$$

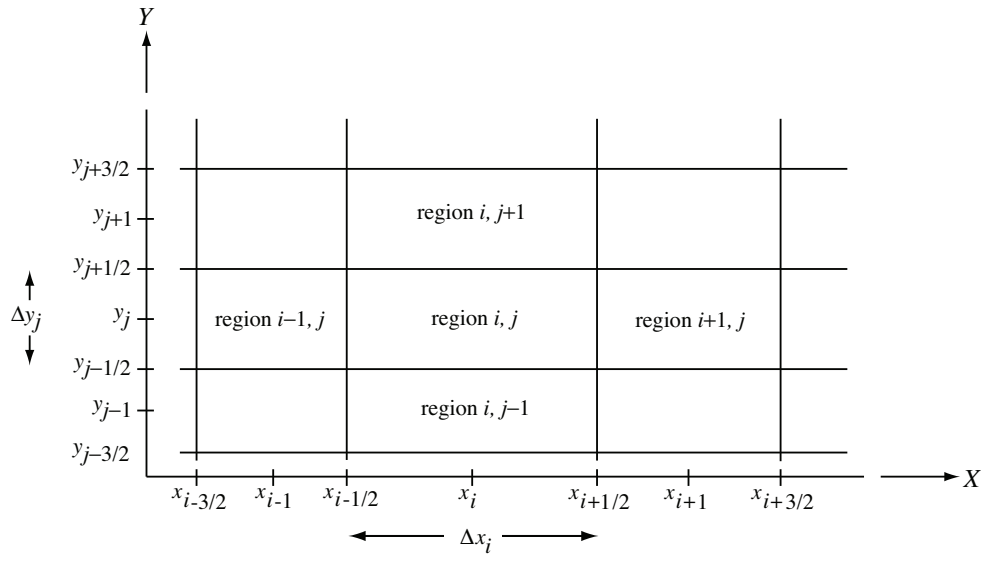

Figure 2: Definition of the regions in 2–D Cartesian geometry

where $x_{i-1/2} < x < x_{i+1/2}$ and $y_{j-1/2} < y < y_{j+1/2}$, with

$$\mathbf{\Phi}(x,y) = \begin{pmatrix} \phi_1(x,y) \\ \vdots \\ \phi_G(x,y) \end{pmatrix} \tag{2.68}$$

and

$$\mathbb{F}_{i,j} = \begin{bmatrix} f_{i,j,11} & f_{i,j,12} & \dots & f_{i,j,1G} \\ f_{i,j,21} & f_{i,j,22} & \dots & f_{i,j,2G} \\ \vdots & \vdots & \ddots & \vdots \\ f_{i,j,G1} & f_{i,j,G2} & \dots & f_{i,j,GG} \end{bmatrix} \tag{2.69}$$

where the components $f_{gi,j,h}$ of this matrix are written as

$$f_{gi,j,h} = \frac{1}{D_{i,j,g}} \left[-\Sigma_{\mathrm{r},i,j,g}\delta_{gh} + \Sigma_{i,j,g\leftarrow h}(1-\delta_{gh}) + \frac{\chi_{i,j,g}}{K_{\mathrm{eff}}}\,\nu\Sigma_{\mathrm{f},i,j,h} \right] \quad . \tag{2.70}$$

The next step consists to find all eigenvectors $\boldsymbol{t}_{i,j,\ell}$ of matrix $\mathbb{F}_{i,j}$ with the associated eigenvalues $\lambda_{i,j,\ell}$. We build a matrix $\mathbb{T}_{i,j}$ whose columns are the eigenvectors of $\mathbb{F}_{i,j}$. This matrix is written

$$\mathbb{T}_{i,j} = \begin{pmatrix} \boldsymbol{t}_{i,j,1} \\ \boldsymbol{t}_{i,j,2} \\ \dots \\ \boldsymbol{t}_{i,j,g} \end{pmatrix} \tag{2.71}$$

so that

$$\mathbb{F}_{i,j}\mathbb{T}_{i,j} = \mathbb{T}_{i,j}\mathrm{diag}(\lambda_{i,j,\ell}) \quad . \tag{2.72}$$

The linear transformation technique used to solve Eq. (2.66) is based on the introduction of an unknown vector $\mathbf{\Psi}(x,y)$ defined as

$$\mathbf{\Phi}(x,y) = \mathbb{T}_{i,j}\mathbf{\Psi}(x,y) = \begin{bmatrix} t_{i,j,11} & t_{i,j,12} & \dots & t_{i,j,1G} \\ t_{i,21} & t_{i,j,22} & \dots & t_{i,j,2G} \\ \vdots & \vdots & \ddots & \vdots \\ t_{i,j,G1} & t_{i,G2} & \dots & t_{i,j,GG} \end{bmatrix} \begin{pmatrix} \psi_1(x,y) \\ \psi_2(x,y) \\ \vdots \\ \psi_G(x,y) \end{pmatrix} \tag{2.73}$$

and to its substitution in Eq. (2.67). We obtain

$$\frac{\partial^2}{\partial x^2}\,\mathbb{T}_{i,j}\,\mathbf{\Psi}(x,y)+\frac{\partial^2}{\partial y^2}\,\mathbb{T}_{i,j}\,\mathbf{\Psi}(x,y)+\mathbb{F}_{i,j}\,\mathbb{T}_{i,j}\,\mathbf{\Psi}(x,y)=\mathbf{0}\quad . \tag{2.74}$$

We next left–multiply each side of Eq. (2.74) by $[\mathbb{T}_i]^{-1}$ and use Eq. (2.72) to obtain

$$\frac{\partial^2}{\partial x^2}\mathbf{\Psi}(x,y)+\frac{\partial^2}{\partial y^2}\mathbf{\Psi}(x,y)+\mathrm{diag}(\lambda_{i,j,\ell})\mathbf{\Psi}(x,y)=\mathbf{0} \tag{2.75}$$

where $x_{i-1/2}<x<x_{i+1/2}$ and $y_{j-1/2}<y<y_{j+1/2}$.

Equation (2.75) is similar to Eq. (2.67) with the difference that all the energy groups are uncoupled. Its resolution will therefore reduces to the solution of G one-speed problems.

2.4 The analytic nodal method in 2–D Cartesian geometry

The two-dimensional (2–D) Cartesian heterogeneous reactor configurations correspond to the case where the neutron flux is function of two spatial variables. These cases cannot be solved analytically and the *analytic nodal method* (ANM) is an attempt to find a solution with the *smallest possible approximation*. Here, we are limiting our investigations to a 2–D Cartesian domain made from the assembly of many x–y rectangular nodes which are infinite in the z direction.

In this case, the neutron flux and the nuclear properties of the reactor are only function of the independent variables x and y. Equation (2.64) simplifies to Eq. (2.66) where each node is assumed to be homogeneous, so that the corresponding nuclear properties are piecewise continuous. As shown in Fig. 2, the reactor domain is divided into $I\times J$ regions of indices $1\leq i\leq I$ and $1\leq j\leq J$, in such a way that the nuclear properties in region i,j are constant and equal to $D_{i,j,g}$, $\Sigma_{\mathrm{r},i,j,g}$, $\Sigma_{i,j,g\leftarrow h}$, $\chi_{i,j,g}$ and $\nu\Sigma_{\mathrm{f},i,j,h}$.

The *linear transformation technique* of Sect. 2.3 is applied on each node, leading to the linear transformation $G\times G$ matrix $\mathbb{T}_{i,j}$ and to a set of G eigenvalues $\lambda_{i,j,\ell}$. The transformation process is repeated for each node, leading to $I\times J$ matrix equations written as

$$\frac{\partial^2}{\partial x^2}\mathbf{\Psi}(x,y)+\frac{\partial^2}{\partial y^2}\mathbf{\Psi}(x,y)+\mathrm{diag}(\lambda_{i,j,\ell})\,\mathbf{\Psi}(x,y)=\mathbf{0} \tag{2.76}$$

if $x_{i-1/2}<x<x_{i+1/2}$ and $y_{j-1/2}<y<y_{j+1/2}$. Each equation is uncoupled in energy, and can be written as G differential equations of the form

$$\frac{\partial^2}{\partial x^2}\psi_g(x,y)+\frac{\partial^2}{\partial y^2}\psi_g(x,y)+\lambda_{i,j,g}\,\psi_g(x,y)=0\;;\quad g=1,G\quad . \tag{2.77}$$

Unfortunately, it is impossible to find the analytical solution of Eq. (2.77) because its dependent variable $\psi_g(x,y)$ is generally not separable. The ANM is based on *transverse integration* of Eq. (2.77), leading to

$$\int_{y_{j-1/2}}^{y_{j+1/2}}dy\,\frac{\partial^2}{\partial x^2}\psi_g(x,y)+\int_{y_{j-1/2}}^{y_{j+1/2}}dy\,\frac{\partial^2}{\partial y^2}\psi_g(x,y)+\lambda_{i,j,g}\int_{y_{j-1/2}}^{y_{j+1/2}}dy\,\psi_g(x,y)=0$$

which can be transverse integrated along the Y axis as

$$\frac{\partial^2}{\partial x^2}\psi^y_{j,g}(x)+\lambda_{i,j,g}\,\psi^y_{j,g}(x)=\mathcal{L}^y_{j,g}(x)\;, \tag{2.78}$$

$$\psi^y_{j,g}(x)=\frac{1}{\Delta y_j}\int_{y_{j-1/2}}^{y_{j+1/2}}dy\,\psi_g(x,y) \tag{2.79}$$

where $\Delta y_j=y_{j+1/2}-y_{j-1/2}$ and where we introduced the X–directed *transverse leakage* term as

$$\mathcal{L}^y_{j,g}(x)=-\frac{1}{\Delta y_j}\int_{y_{j-1/2}}^{y_{j+1/2}}dy\,\frac{\partial^2}{\partial y^2}\psi_g(x,y)=-\frac{1}{\Delta y_j}\,\frac{\partial}{\partial y}\psi_g(x,y)\Big|_{y_{j-1/2}}^{y_{j+1/2}}\quad . \tag{2.80}$$

Similarly, the transverse integration along the X axis leads to

$$\frac{\partial^2}{\partial y^2}\psi^x_{i,g}(y) + \lambda_{i,j,g}\,\psi^x_{i,g}(y) = \mathcal{L}^x_{i,g}(y)\ , \tag{2.81}$$

$$\psi^x_{i,g}(y) = \frac{1}{\Delta x_i}\int_{x_{i-1/2}}^{x_{i+1/2}} dx\,\psi_g(x,y) \tag{2.82}$$

where $\Delta x_i = x_{i+1/2} - x_{i-1/2}$ and where we introduced the Y–directed *transverse leakage* term as

$$\mathcal{L}^x_{i,g}(y) = -\frac{1}{\Delta x_i}\int_{x_{i-1/2}}^{x_{i+1/2}} dx\,\frac{\partial^2}{\partial x^2}\psi_g(x,y) = -\frac{1}{\Delta x_i}\,\frac{\partial}{\partial x}\psi_g(x,y)\Big|_{x_{i-1/2}}^{x_{i+1/2}}\ . \tag{2.83}$$

Equations (2.78) and (2.81) can be solved analytically, provided that the x and y variation of the transverse leakage terms $\mathcal{L}^y_{j,g}(x)$ and $\mathcal{L}^x_{i,g}(y)$ are known. This is where we introduce the *quadratic leakage approximation* of Eq.(2.7) as the *unique* approximation of the ANM. Such an approximation can be constructed for node (i,j), in the X–direction, using $\bar{L}^y_{i-1,j,g}$, $\bar{L}^y_{i,j,g}$ and $\bar{L}^y_{i+1,j,g}$, the transverse leakage terms *without* linear transformation. In his thesis, Smith developed the ANM with a quadratic leakage approximation in two–group, 3D Cartesian geometry. Here, we are presenting the ANM with the quadratic transverse leakage approximation in G–group and 2–D Cartesian geometry.

We now present the relations between node averaged fluxes $\bar{\phi}_{i,j,g}$ and $X-$directed boundary fluxes $\phi^y_{j,g}(x_{i\pm1/2})$ and boundary net currents $J^y_{j,g}(x_{i\pm1/2})$. Under these conditions, the right-hand term of Eq. (2.78) is assumed to exhibit a quadratic variation in x, written as

$$L^y_{j,g}(x) = b_0 + b_1\,x + b_2\,x^2$$

where $x_{i-1/2} \le x \le x_{i+1/2}$. The polynomial coefficients in Eq. (2.84) are related to $\bar{L}^y_{i-1,j,g}$, $\bar{L}^y_{i,j,g}$ and $\bar{L}^y_{i+1,j,g}$, the transverse leakage terms *without linear transformation*, using

$$\frac{1}{\Delta x_{i-1}}\int_{x_{i-3/2}}^{x_{i-1/2}} dx\,L^y_{j,g}(x) = \bar{L}^y_{i-1,j,g}\quad,\quad \frac{1}{\Delta x_i}\int_{x_{i-1/2}}^{x_{i+1/2}} dx\,L^y_{j,g}(x) = \bar{L}^y_{i,j,g}$$

and

$$\frac{1}{\Delta x_{i+1}}\int_{x_{i+1/2}}^{x_{i+3/2}} dx\,L^y_{j,g}(x) = \bar{L}^y_{i+1,j,g}\quad, \tag{2.84}$$

so that

$$\begin{bmatrix} b_0 \\ b_1 \\ b_2 \end{bmatrix} = \begin{bmatrix} 1 & \frac{x_{i-3/2}+x_{i-1/2}}{2} & \frac{{x_{i-3/2}}^2+x_{i-3/2}x_{i-1/2}+{x_{i-1/2}}^2}{3} \\ 1 & \frac{x_{i-1/2}+x_{i+1/2}}{2} & \frac{{x_{i-1/2}}^2+x_{i-1/2}x_{i+1/2}+{x_{i+1/2}}^2}{3} \\ 1 & \frac{x_{i+1/2}+x_{i+3/2}}{2} & \frac{{x_{i+1/2}}^2+x_{i+1/2}x_{i+3/2}+{x_{i+3/2}}^2}{3} \end{bmatrix}^{-1} \begin{bmatrix} \bar{L}^y_{i-1,j,g} \\ \bar{L}^y_{i,j,g} \\ \bar{L}^y_{i+1,j,g} \end{bmatrix}\ . \tag{2.85}$$

If a node i is located on a zero-flux left boundary, Eqs. (2.84) are still used, taking care to set $\bar{L}^y_{i-1,j,g} = 0$. Similarly, if a node i is located on a zero-flux right boundary, we use $\bar{L}^y_{i+1,j,g} = 0$. This approach is selected in a try to keep the compatibility with the recipe used in QUANDRY, the original ANM implementation by Smith.[4]

The integration of Eq. (2.78) over node (i,j) leads to the transformed *nodal balance equation*, written as

$$\bar{\psi}_{i,j,g} \;=\; \frac{1}{\lambda_{i,j,g}}\left(\bar{\mathcal{L}}^x_{i,j,g} + \bar{\mathcal{L}}^y_{i,j,g}\right)\quad .$$

Let us first consider the case where $\lambda_{i,j,g} \ge 0$. In energy group g and in node (i,j), Eq. (2.78) has an analytical solution of the form

$$\psi^y_{j,g}(x) = \alpha + \beta\,x + \gamma\,x^2 + A_{i,j,g}\cos(\sqrt{\lambda_{i,j,g}}\;x) + B_{i,j,g}\sin(\sqrt{\lambda_{i,j,g}}\;x) \tag{2.86}$$

if $x_{i-1/2} < x < x_{i+1/2}$. The particular solution is a quadratic polynomial in x with coefficients

$$\alpha = \frac{b_0}{\lambda_{i,j,g}} - \frac{2b_2}{\lambda^2_{i,j,g}}\ ,\quad \beta = \frac{b_1}{\lambda_{i,j,g}}\quad \text{and}\quad \gamma = \frac{b_2}{\lambda_{i,j,g}} \tag{2.87}$$

where b_0, b_1 and b_2 are the polynomial coefficients of the transverse leakage term $L^y_{j,g}(x)$, as given by Eq. (2.84).

Integrating Eq. (2.86) over the node leads to

$$\begin{aligned}\bar{\psi}_{i,j,g} &= \alpha + \beta\,\frac{x_{i-i/2}+x_{i+i/2}}{2} + \gamma\,\frac{x^2_{i-i/2}+x_{i-i/2}x_{i+i/2}+x^2_{i+i/2}}{3} \\ &+ \frac{A_{i,j,g}}{\Delta x_i\sqrt{\lambda_{i,j,g}}}\sin(\sqrt{\lambda_{i,j,g}}\;x)\Big|^{x_{i+i/2}}_{x_{i-i/2}} - \frac{B_{i,j,g}}{\Delta x_i\sqrt{\lambda_{i,j,g}}}\cos(\sqrt{\lambda_{i,j,g}}\;x)\Big|^{x_{i+i/2}}_{x_{i-i/2}} \quad .\end{aligned} \tag{2.88}$$

Differentiating Eq. (2.86) over the node leads to

$$\begin{aligned}\mathcal{J}^y_{j,g}(x) &= -\beta - 2\gamma\,x + A_{i,j,g}\sqrt{\lambda_{i,j,g}}\sin(\sqrt{\lambda_{i,j,g}}\;x) \\ &- B_{i,j,g}\sqrt{\lambda_{i,j,g}}\cos(\sqrt{\lambda_{i,j,g}}\;x) \quad .\end{aligned} \tag{2.89}$$

Equations (2.86) to (2.89) can be rewritten, if $\lambda_{i,j,g} \leq 0$, as

$$\begin{aligned}\psi^y_{j,g}(x) &= \alpha + \beta\,x + \gamma\,x^2 + C_{i,j,g}\cosh(\sqrt{-\lambda_{i,j,g}}\;x) \\ &+ E_{i,j,g}\sinh(\sqrt{-\lambda_{i,j,g}}\;x) \quad ,\end{aligned} \tag{2.90}$$

$$\begin{aligned}\bar{\psi}_{i,j,g} &= \alpha + \beta\,\frac{x_{i-i/2}+x_{i+i/2}}{2} + \gamma\,\frac{x^2_{i-i/2}+x_{i-i/2}x_{i+i/2}+x^2_{i+i/2}}{3} \\ &+ \frac{C_{i,j,g}}{\Delta x_i\sqrt{-\lambda_{i,j,g}}}\sinh(\sqrt{-\lambda_{i,j,g}}\;x)\Big|^{x_{i+i/2}}_{x_{i-i/2}} \\ &+ \frac{E_{i,j,g}}{\Delta x_i\sqrt{-\lambda_{i,j,g}}}\cosh(\sqrt{-\lambda_{i,j,g}}\;x)\Big|^{x_{i+i/2}}_{x_{i-i/2}}\end{aligned} \tag{2.91}$$

and

$$\begin{aligned}\mathcal{J}^y_{j,g}(x) &= -\beta - 2\gamma\,x - C_{i,j,g}\sqrt{-\lambda_{i,j,g}}\sinh(\sqrt{-\lambda_{i,j,g}}\;x) \\ &- E_{i,j,g}\sqrt{-\lambda_{i,j,g}}\cosh(\sqrt{-\lambda_{i,j,g}}\;x) \quad .\end{aligned} \tag{2.92}$$

To proceed further, we need to rewrite Eqs. (2.86) to (2.92) in matrix algebra. We define

$$\begin{aligned}\boldsymbol{\Psi}^y_j(x) &= \{\psi^y_{j,g}(x)\;;\;\; g=1,G\} \\ \bar{\boldsymbol{\Psi}}_{i,j} &= \{\bar{\psi}_{i,j,g}\;;\;\; g=1,G\} \\ \boldsymbol{\mathcal{J}}^y_j(x) &= \{\mathcal{J}^y_{j,g}(x)\;;\;\; g=1,G\} \\ \boldsymbol{\mathcal{J}}^x_i(y) &= \{\mathcal{J}^x_{i,g}(y)\;;\;\; g=1,G\} \\ \boldsymbol{A}_{i,j} &= \{A_{i,j,g}\;;\;\; g=1,G\} \\ \boldsymbol{B}_{i,j} &= \{B_{i,j,g}\;;\;\; g=1,G\} \\ \boldsymbol{J}^x_i(y) &= \Big\{-D_{i,j,g}\frac{d}{dy}\phi^x_{i,g}(y)\;;\;\; g=1,G\Big\} = \mathrm{diag}(D_{i,j,g})\,\mathbb{T}_{i,j}\,\boldsymbol{\mathcal{J}}^x_i(y) \\ \mathbb{S}_{i,j} &= \mathbb{T}^{-1}_{i,j}\mathrm{diag}(D_{i,j,g})^{-1} \quad ,\end{aligned}$$

so that the above equations can be cast into

$$\begin{aligned}\begin{pmatrix}\bar{\boldsymbol{\Psi}}_{i,j}\\ \boldsymbol{\mathcal{J}}^y_j(x_{i-1/2})\end{pmatrix} &= \mathbb{M}^-_{i,j}\begin{pmatrix}\boldsymbol{A}_{i,j}\\ \boldsymbol{B}_{i,j}\end{pmatrix} + \begin{bmatrix}-\mathbb{V}^-_{i,j} & \mathbb{V}^-_{i,j}\end{bmatrix} \\ &\times \begin{bmatrix}\mathbb{S}_{i,j} & \mathbb{O} & \mathbb{O} & \mathbb{O} & \mathbb{O} & \mathbb{O}\\ \mathbb{O} & \mathbb{S}_{i,j} & \mathbb{O} & \mathbb{O} & \mathbb{O} & \mathbb{O}\\ \mathbb{O} & \mathbb{O} & \mathbb{S}_{i,j} & \mathbb{O} & \mathbb{O} & \mathbb{O}\\ \mathbb{O} & \mathbb{O} & \mathbb{O} & \mathbb{S}_{i,j} & \mathbb{O} & \mathbb{O}\\ \mathbb{O} & \mathbb{O} & \mathbb{O} & \mathbb{O} & \mathbb{S}_{i,j} & \mathbb{O}\\ \mathbb{O} & \mathbb{O} & \mathbb{O} & \mathbb{O} & \mathbb{O} & \mathbb{S}_{i,j}\end{bmatrix}\begin{pmatrix}\boldsymbol{J}^x_{i-1}(y_{j-1/2})\\ \boldsymbol{J}^x_i(y_{j-1/2})\\ \boldsymbol{J}^x_{i+1}(y_{j-1/2})\\ \boldsymbol{J}^x_{i-1}(y_{j+1/2})\\ \boldsymbol{J}^x_i(y_{j+1/2})\\ \boldsymbol{J}^x_{i+1}(y_{j+1/2})\end{pmatrix} \quad ,\end{aligned} \tag{2.93}$$

$$\begin{pmatrix} \bar{\boldsymbol{\Psi}}_{i,j} \\ \boldsymbol{\mathcal{J}}^y_j(x_{i+1/2}) \end{pmatrix} = \mathbb{M}^+_{i,j} \begin{pmatrix} \boldsymbol{A}_{i,j} \\ \boldsymbol{B}_{i,j} \end{pmatrix} + \begin{bmatrix} -\mathbb{V}^+_{i,j} & \mathbb{V}^+_{i,j} \end{bmatrix} \times \begin{bmatrix} \mathbb{S}_{i,j} & \mathbb{0} & \mathbb{0} & \mathbb{0} & \mathbb{0} & \mathbb{0} \\ \mathbb{0} & \mathbb{S}_{i,j} & \mathbb{0} & \mathbb{0} & \mathbb{0} & \mathbb{0} \\ \mathbb{0} & \mathbb{0} & \mathbb{S}_{i,j} & \mathbb{0} & \mathbb{0} & \mathbb{0} \\ \mathbb{0} & \mathbb{0} & \mathbb{0} & \mathbb{S}_{i,j} & \mathbb{0} & \mathbb{0} \\ \mathbb{0} & \mathbb{0} & \mathbb{0} & \mathbb{0} & \mathbb{S}_{i,j} & \mathbb{0} \\ \mathbb{0} & \mathbb{0} & \mathbb{0} & \mathbb{0} & \mathbb{0} & \mathbb{S}_{i,j} \end{bmatrix} \begin{pmatrix} \boldsymbol{J}^x_{i-1}(y_{j-1/2}) \\ \boldsymbol{J}^x_i(y_{j-1/2}) \\ \boldsymbol{J}^x_{i+1}(y_{j-1/2}) \\ \boldsymbol{J}^x_{i-1}(y_{j+1/2}) \\ \boldsymbol{J}^x_i(y_{j+1/2}) \\ \boldsymbol{J}^x_{i+1}(y_{j+1/2}) \end{pmatrix}, \tag{2.94}$$

$$\boldsymbol{\Psi}^y_j(x_{i-1/2}) = \mathbb{N}^-_{i,j} \begin{pmatrix} \boldsymbol{A}_{i,j} \\ \boldsymbol{B}_{i,j} \end{pmatrix} + \begin{bmatrix} -\mathbb{U}^-_{i,j} & \mathbb{U}^-_{i,j} \end{bmatrix} \times \begin{bmatrix} \mathbb{S}_{i,j} & \mathbb{0} & \mathbb{0} & \mathbb{0} & \mathbb{0} & \mathbb{0} \\ \mathbb{0} & \mathbb{S}_{i,j} & \mathbb{0} & \mathbb{0} & \mathbb{0} & \mathbb{0} \\ \mathbb{0} & \mathbb{0} & \mathbb{S}_{i,j} & \mathbb{0} & \mathbb{0} & \mathbb{0} \\ \mathbb{0} & \mathbb{0} & \mathbb{0} & \mathbb{S}_{i,j} & \mathbb{0} & \mathbb{0} \\ \mathbb{0} & \mathbb{0} & \mathbb{0} & \mathbb{0} & \mathbb{S}_{i,j} & \mathbb{0} \\ \mathbb{0} & \mathbb{0} & \mathbb{0} & \mathbb{0} & \mathbb{0} & \mathbb{S}_{i,j} \end{bmatrix} \begin{pmatrix} \boldsymbol{J}^x_{i-1}(y_{j-1/2}) \\ \boldsymbol{J}^x_i(y_{j-1/2}) \\ \boldsymbol{J}^x_{i+1}(y_{j-1/2}) \\ \boldsymbol{J}^x_{i-1}(y_{j+1/2}) \\ \boldsymbol{J}^x_i(y_{j+1/2}) \\ \boldsymbol{J}^x_{i+1}(y_{j+1/2}) \end{pmatrix} \tag{2.95}$$

and

$$\boldsymbol{\Psi}^y_j(x_{i+1/2}) = \mathbb{N}^+_{i,j} \begin{pmatrix} \boldsymbol{A}_{i,j} \\ \boldsymbol{B}_{i,j} \end{pmatrix} + \begin{bmatrix} -\mathbb{U}^+_{i,j} & \mathbb{U}^+_{i,j} \end{bmatrix} \times \begin{bmatrix} \mathbb{S}_{i,j} & \mathbb{0} & \mathbb{0} & \mathbb{0} & \mathbb{0} & \mathbb{0} \\ \mathbb{0} & \mathbb{S}_{i,j} & \mathbb{0} & \mathbb{0} & \mathbb{0} & \mathbb{0} \\ \mathbb{0} & \mathbb{0} & \mathbb{S}_{i,j} & \mathbb{0} & \mathbb{0} & \mathbb{0} \\ \mathbb{0} & \mathbb{0} & \mathbb{0} & \mathbb{S}_{i,j} & \mathbb{0} & \mathbb{0} \\ \mathbb{0} & \mathbb{0} & \mathbb{0} & \mathbb{0} & \mathbb{S}_{i,j} & \mathbb{0} \\ \mathbb{0} & \mathbb{0} & \mathbb{0} & \mathbb{0} & \mathbb{0} & \mathbb{S}_{i,j} \end{bmatrix} \begin{pmatrix} \boldsymbol{J}^x_{i-1}(y_{j-1/2}) \\ \boldsymbol{J}^x_i(y_{j-1/2}) \\ \boldsymbol{J}^x_{i+1}(y_{j-1/2}) \\ \boldsymbol{J}^x_{i-1}(y_{j+1/2}) \\ \boldsymbol{J}^x_i(y_{j+1/2}) \\ \boldsymbol{J}^x_{i+1}(y_{j+1/2}) \end{pmatrix} \tag{2.96}$$

where $\mathbb{M}^\pm_{i,j}$ are two $2G \times 2G$ matrices. $\mathbb{N}^\pm_{i,j}$ are two $G \times 2G$ matrices. Finally, $\mathbb{U}^\pm_{i,j}$ are $G \times 3G$ matrices and $\mathbb{V}^\pm_{i,j}$ are $2G \times 3G$ matrices. The coefficients of these matrices are recovered from Eqs. (2.86) to (2.92). Coefficients $\boldsymbol{A}_{i,j}$ and $\boldsymbol{B}_{i,j}$ from Eq. (2.95) can be eliminated using Eq. (2.93) and coefficients $\boldsymbol{A}_{i,j}$ and $\boldsymbol{B}_{i,j}$ from Eq. (2.96) can be eliminated using Eq. (2.94). The resulting equations can be cast into

$$\boldsymbol{\Psi}^y_j(x_{i-1/2}) = \mathbb{P}^-_{i,j} \bar{\boldsymbol{\Psi}}_{i,j} + \mathbb{P}^+_{i,j} \boldsymbol{\mathcal{J}}^y_j(x_{i-1/2}) + \mathbb{P}^{\text{sour}}_{i,j} \begin{pmatrix} \boldsymbol{J}^x_{i-1}(y_{j-1/2}) \\ \boldsymbol{J}^x_i(y_{j-1/2}) \\ \boldsymbol{J}^x_{i+1}(y_{j-1/2}) \\ \boldsymbol{J}^x_{i-1}(y_{j+1/2}) \\ \boldsymbol{J}^x_i(y_{j+1/2}) \\ \boldsymbol{J}^x_{i+1}(y_{j+1/2}) \end{pmatrix} \tag{2.97}$$

where $\mathbb{P}^\pm_{i,j}$ are $G \times G$ matrices and $\mathbb{P}^{\text{sour}}_{i,j}$ is a $G \times 6G$ matrix;

$$\boldsymbol{\Psi}^y_j(x_{i+1/2}) = \mathbb{Q}^-_{i,j} \bar{\boldsymbol{\Psi}}_{i,j} + \mathbb{Q}^+_{i,j} \boldsymbol{\mathcal{J}}^y_j(x_{i+1/2}) + \mathbb{Q}^{\text{sour}}_{i,j} \begin{pmatrix} \boldsymbol{J}^x_{i-1}(y_{j-1/2}) \\ \boldsymbol{J}^x_i(y_{j-1/2}) \\ \boldsymbol{J}^x_{i+1}(y_{j-1/2}) \\ \boldsymbol{J}^x_{i-1}(y_{j+1/2}) \\ \boldsymbol{J}^x_i(y_{j+1/2}) \\ \boldsymbol{J}^x_{i+1}(y_{j+1/2}) \end{pmatrix} \tag{2.98}$$

where $\mathbb{Q}^\pm_{i,j}$ are $G \times G$ matrices and $\mathbb{Q}^{\text{sour}}_{i,j}$ is a $G \times 6G$ matrix.

The transformed nodal balance equation (2.86) can be written in matrix form as

$$\bar{\boldsymbol{\Psi}}_{i,j} = \begin{bmatrix} -\mathbb{X}_{i,j} & \mathbb{X}_{i,j} \end{bmatrix} \begin{pmatrix} \boldsymbol{\mathcal{J}}^y_j(x_{i-1/2}) \\ \boldsymbol{\mathcal{J}}^y_j(x_{i+1/2}) \end{pmatrix} + \begin{bmatrix} -\mathbb{Y}_{i,j} & \mathbb{Y}_{i,j} \end{bmatrix} \begin{pmatrix} \boldsymbol{\mathcal{J}}^x_i(y_{j-1/2}) \\ \boldsymbol{\mathcal{J}}^x_i(y_{j+1/2}) \end{pmatrix}. \tag{2.99}$$

Finally, the linear transformation can be inverted, with the help of the following definitions:

$$\begin{aligned}
\boldsymbol{\Phi}_j^y(x) &= \{\phi_{j,g}^y(x)\ ;\ \ g=1,G\} = \mathbb{T}_{i,j}\,\boldsymbol{\Psi}_j^y(x)\\
\bar{\boldsymbol{\Phi}}_{i,j} &= \{\bar{\phi}_{i,j,g}\ ;\ \ g=1,G\} = \mathbb{T}_{i,j}\,\bar{\boldsymbol{\Psi}}_{i,j}\\
\boldsymbol{J}_j^y(x) &= \Big\{-D_{i,j,g}\frac{d}{dx}\phi_{j,g}^y(x)\ ;\ \ g=1,G\Big\} = \mathrm{diag}(D_{i,j,g})\,\mathbb{T}_{i,j}\,\boldsymbol{\mathcal{J}}_j^y(x)\ \ .
\end{aligned}$$

We can show that

$$\boldsymbol{\Phi}_j^y(x_{i-1/2}) = \mathbb{L}_{i,j}^{x-}\bar{\boldsymbol{\Phi}}_{i,j} + \mathbb{L}_{i,j}^{x+}\boldsymbol{J}_j^y(x_{i-1/2}) + \mathbb{L}_{i,j}^{x,\mathrm{sour}}\begin{pmatrix}\boldsymbol{J}_{i-1}^x(y_{j-1/2})\\ \boldsymbol{J}_i^x(y_{j-1/2})\\ \boldsymbol{J}_{i+1}^x(y_{j-1/2})\\ \boldsymbol{J}_{i-1}^x(y_{j+1/2})\\ \boldsymbol{J}_i^x(y_{j+1/2})\\ \boldsymbol{J}_{i+1}^x(y_{j+1/2})\end{pmatrix} \tag{2.100}$$

and

$$\boldsymbol{\Phi}_j^y(x_{i+1/2}) = \mathbb{R}_{i,j}^{x-}\bar{\boldsymbol{\Phi}}_{i,j} + \mathbb{R}_{i,j}^{x+}\boldsymbol{J}_j^y(x_{i+1/2}) + \mathbb{R}_{i,j}^{x,\mathrm{sour}}\begin{pmatrix}\boldsymbol{J}_{i-1}^x(y_{j-1/2})\\ \boldsymbol{J}_i^x(y_{j-1/2})\\ \boldsymbol{J}_{i+1}^x(y_{j-1/2})\\ \boldsymbol{J}_{i-1}^x(y_{j+1/2})\\ \boldsymbol{J}_i^x(y_{j+1/2})\\ \boldsymbol{J}_{i+1}^x(y_{j+1/2})\end{pmatrix} \tag{2.101}$$

where

$$\begin{aligned}
\mathbb{L}_{i,j}^{x-} &= \mathbb{T}_{i,j}\,\mathbb{P}_{i,j}^{-}\,\mathbb{T}_{i,j}^{-1}\ , \quad \mathbb{L}_{i,j}^{x+} = \mathbb{T}_{i,j}\,\mathbb{P}_{i,j}^{+}\,\mathbb{S}_{i,j}\ ,\\
\mathbb{L}_{i,j}^{x,\mathrm{sour}} &= \mathbb{T}_{i,j}\,\mathbb{P}_{i,j}^{\mathrm{sour}}\begin{bmatrix}\mathbb{S}_{i,j} & \mathbb{0} & \mathbb{0} & \mathbb{0} & \mathbb{0} & \mathbb{0}\\ \mathbb{0} & \mathbb{S}_{i,j} & \mathbb{0} & \mathbb{0} & \mathbb{0} & \mathbb{0}\\ \mathbb{0} & \mathbb{0} & \mathbb{S}_{i,j} & \mathbb{0} & \mathbb{0} & \mathbb{0}\\ \mathbb{0} & \mathbb{0} & \mathbb{0} & \mathbb{S}_{i,j} & \mathbb{0} & \mathbb{0}\\ \mathbb{0} & \mathbb{0} & \mathbb{0} & \mathbb{0} & \mathbb{S}_{i,j} & \mathbb{0}\\ \mathbb{0} & \mathbb{0} & \mathbb{0} & \mathbb{0} & \mathbb{0} & \mathbb{S}_{i,j}\end{bmatrix}
\end{aligned} \tag{2.102}$$

and

$$\begin{aligned}
\mathbb{R}_{i,j}^{x-} &= \mathbb{T}_{i,j}\,\mathbb{Q}_{i,j}^{-}\,\mathbb{T}_{i,j}^{-1}\ , \quad \mathbb{R}_{i,j}^{x+} = \mathbb{T}_{i,j}\,\mathbb{Q}_{i,j}^{+}\,\mathbb{S}_{i,j}\ ,\\
\mathbb{R}_{i,j}^{x,\mathrm{sour}} &= \mathbb{T}_{i,j}\,\mathbb{Q}_{i,j}^{\mathrm{sour}}\begin{bmatrix}\mathbb{S}_{i,j} & \mathbb{0} & \mathbb{0} & \mathbb{0} & \mathbb{0} & \mathbb{0}\\ \mathbb{0} & \mathbb{S}_{i,j} & \mathbb{0} & \mathbb{0} & \mathbb{0} & \mathbb{0}\\ \mathbb{0} & \mathbb{0} & \mathbb{S}_{i,j} & \mathbb{0} & \mathbb{0} & \mathbb{0}\\ \mathbb{0} & \mathbb{0} & \mathbb{0} & \mathbb{S}_{i,j} & \mathbb{0} & \mathbb{0}\\ \mathbb{0} & \mathbb{0} & \mathbb{0} & \mathbb{0} & \mathbb{S}_{i,j} & \mathbb{0}\\ \mathbb{0} & \mathbb{0} & \mathbb{0} & \mathbb{0} & \mathbb{0} & \mathbb{S}_{i,j}\end{bmatrix}.
\end{aligned} \tag{2.103}$$

The X–directed ANM coupling relations are therefore written by imposing the flux continuity on $x_{i-1/2}$, as

$$\begin{aligned}
\big[\boldsymbol{f}_{i-1,j}^{x+}\big]&\left[\mathbb{R}_{i-1,j}^{x-}\bar{\boldsymbol{\Phi}}_{i-1,j} + \mathbb{R}_{i-1,j}^{x+}\boldsymbol{J}_j^y(x_{i-1/2}) + \mathbb{R}_{i-1,j}^{x,\mathrm{sour}}\begin{pmatrix}\boldsymbol{J}_{i-2}^x(y_{j-1/2})\\ \boldsymbol{J}_{i-1}^x(y_{j-1/2})\\ \boldsymbol{J}_i^x(y_{j-1/2})\\ \boldsymbol{J}_{i-2}^x(y_{j+1/2})\\ \boldsymbol{J}_{i-1}^x(y_{j+1/2})\\ \boldsymbol{J}_i^x(y_{j+1/2})\end{pmatrix}\right]\\
&= \big[\boldsymbol{f}_{i,j}^{x-}\big]\left[\mathbb{L}_{i,j}^{x-}\bar{\boldsymbol{\Phi}}_{i,j} + \mathbb{L}_{i,j}^{x+}\boldsymbol{J}_j^y(x_{i-1/2}) + \mathbb{L}_{i,j}^{x,\mathrm{sour}}\begin{pmatrix}\boldsymbol{J}_{i-1}^x(y_{j-1/2})\\ \boldsymbol{J}_i^x(y_{j-1/2})\\ \boldsymbol{J}_{i+1}^x(y_{j-1/2})\\ \boldsymbol{J}_{i-1}^x(y_{j+1/2})\\ \boldsymbol{J}_i^x(y_{j+1/2})\\ \boldsymbol{J}_{i+1}^x(y_{j+1/2})\end{pmatrix}\right]
\end{aligned} \tag{2.104}$$

where $\left[\boldsymbol{f}_{i-1,j}^{x+}\right]$ and $\left[\boldsymbol{f}_{i,j}^{x-}\right]$ are $G \times G$ diagonal matrices containing multigroup discontinuity functions.

The nodal balance equation can be written in term of non-transformed variables as

$$\bar{\boldsymbol{\Phi}}_{i,j} = \mathbb{B}_{i,j}^{x} \begin{pmatrix} \boldsymbol{J}_j^y(x_{i-1/2}) \\ \boldsymbol{J}_j^y(x_{i+1/2}) \end{pmatrix} + \mathbb{B}_{i,j}^{y} \begin{pmatrix} \boldsymbol{J}_i^x(y_{j-1/2}) \\ \boldsymbol{J}_i^x(y_{j+1/2}) \end{pmatrix} \tag{2.105}$$

where

$$\mathbb{B}_{i,j}^{x} = \mathbb{T}_{i,j} \begin{bmatrix} -\mathbb{X}_{i,j} & \mathbb{X}_{i,j} \end{bmatrix} \begin{bmatrix} \mathbb{S}_{i,j} & \mathbb{O} \\ \mathbb{O} & \mathbb{S}_{i,j} \end{bmatrix} \tag{2.106}$$

and

$$\mathbb{B}_{i,j}^{y} = \mathbb{T}_{i,j} \begin{bmatrix} -\mathbb{Y}_{i,j} & \mathbb{Y}_{i,j} \end{bmatrix} \begin{bmatrix} \mathbb{S}_{i,j} & \mathbb{O} \\ \mathbb{O} & \mathbb{S}_{i,j} \end{bmatrix} \quad . \tag{2.107}$$

Relations (2.104) are used together with the Y–directed ANM coupling relations and with the nodal balance equation (2.105) to build the global matrix system.

Calculation of matrices $\mathbb{L}_{i,j}$, $\mathbb{R}_{i,j}$, $\mathbb{B}_{i,j}^{x}$ and $\mathbb{B}_{i,j}^{y}$ for each node represents the core of the ANM. These matrices are function of the G–group cross sections and diffusion coefficients, of the node size *and* of the effective multiplication factor K_{eff}. They must be updated during the power iteration, as K_{eff} change.

Assuming a rectangular domain, the zero flux boundary conditions at surfaces $x_{-1/2}$ and $x_{I+1/2}$ for all values of y corresponding to the ANM are:

$$\mathbb{L}_{1,j}^{x-} \bar{\boldsymbol{\Phi}}_{1,j} + \mathbb{L}_{1,j}^{x+} \boldsymbol{J}_j^y(x_{-1/2}) + \mathbb{L}_{1,j}^{x,\text{sour}} \begin{pmatrix} \boldsymbol{J}_1^x(y_{j-1/2}) \\ \boldsymbol{J}_2^x(y_{j-1/2}) \\ \boldsymbol{J}_3^x(y_{j-1/2}) \\ \boldsymbol{J}_1^x(y_{j+1/2}) \\ \boldsymbol{J}_2^x(y_{j+1/2}) \\ \boldsymbol{J}_3^x(y_{j+1/2}) \end{pmatrix} = 0 \ , \quad \forall j \tag{2.108}$$

and

$$\mathbb{R}_{I,j}^{x-} \bar{\boldsymbol{\Phi}}_{I,j} + \mathbb{R}_{I,j}^{x+} \boldsymbol{J}_j^y(x_{I-1/2}) + \mathbb{R}_{I,j}^{x,\text{sour}} \begin{pmatrix} \boldsymbol{J}_{I-2}^x(y_{j-1/2}) \\ \boldsymbol{J}_{I-1}^x(y_{j-1/2}) \\ \boldsymbol{J}_I^x(y_{j-1/2}) \\ \boldsymbol{J}_{I-2}^x(y_{j+1/2}) \\ \boldsymbol{J}_{I-1}^x(y_{j+1/2}) \\ \boldsymbol{J}_I^x(y_{j+1/2}) \end{pmatrix} = 0 \ , \quad \forall j. \tag{2.109}$$

Assuming a rectangular domain, the albedo boundary conditions at surfaces $x_{-1/2}$ and $x_{I+1/2}$ for all values of y corresponding to the ANM are:

$$\left\{\mathbb{I} + \left[\boldsymbol{\Lambda}_j^{x-}\right]\left[\mathbb{L}_{1,j}^{x+}\right]\right\} \boldsymbol{J}_j^y(x_{-1/2}) = -\left[\boldsymbol{\Lambda}_j^{x-}\right] \left[\mathbb{L}_{1,j}^{x-} \bar{\boldsymbol{\Phi}}_{1,j} + \mathbb{L}_{1,j}^{x,\text{sour}} \begin{pmatrix} \boldsymbol{J}_1^x(y_{j-1/2}) \\ \boldsymbol{J}_2^x(y_{j-1/2}) \\ \boldsymbol{J}_3^x(y_{j-1/2}) \\ \boldsymbol{J}_1^x(y_{j+1/2}) \\ \boldsymbol{J}_2^x(y_{j+1/2}) \\ \boldsymbol{J}_3^x(y_{j+1/2}) \end{pmatrix} \right] \ , \quad \forall j \tag{2.110}$$

and

$$\left\{-\mathbb{I} + \left[\boldsymbol{\Lambda}_j^{x+}\right]\left[\mathbb{R}_{I,j}^{x+}\right]\right\} \boldsymbol{J}_j^y(x_{I-1/2}) = -\left[\boldsymbol{\Lambda}_j^{x+}\right] \left[\mathbb{R}_{I,j}^{x-} \bar{\boldsymbol{\Phi}}_{I,j} + \mathbb{R}_{I,j}^{x,\text{sour}} \begin{pmatrix} \boldsymbol{J}_{I-2}^x(y_{j-1/2}) \\ \boldsymbol{J}_{I-1}^x(y_{j-1/2}) \\ \boldsymbol{J}_I^x(y_{j-1/2}) \\ \boldsymbol{J}_{I-2}^x(y_{j+1/2}) \\ \boldsymbol{J}_{I-1}^x(y_{j+1/2}) \\ \boldsymbol{J}_I^x(y_{j+1/2}) \end{pmatrix} \right] \ , \quad \forall j \tag{2.111}$$

where $\mathbb{I}$ is the $G \times G$ identity matrix and where $\left[\boldsymbol{\Lambda}_j^{x-}\right]$ and $\left[\boldsymbol{\Lambda}_j^{x+}\right]$ are $G \times G$ diagonal matrices containing multigroup albedo functions as defined by Eq. (2.17).

2.5 The nodal update procedure

The nodal update procedure performs a sequence of one- and two-nodes one-dimensional fixed-source calculations to obtain the NEM or ANM nodal coefficients as a function of node-averaged fluxes $\bar{\phi}_{i,g}$, sources $\bar{Q}_{i,g}$ and discontinuity factors $f_{i,g}^{\pm}$. These nodal coefficients are then used to re-evaluate the discontinuous neutron flux $\phi_{i\pm1/2,g}^{\mp}$ and continuous current $J_{i\pm1/2,g}$ at node interfaces and to update the drift coefficients $\tilde{D}_{i,g}^{\pm}$. This algorithm is depicted in Fig. 3.

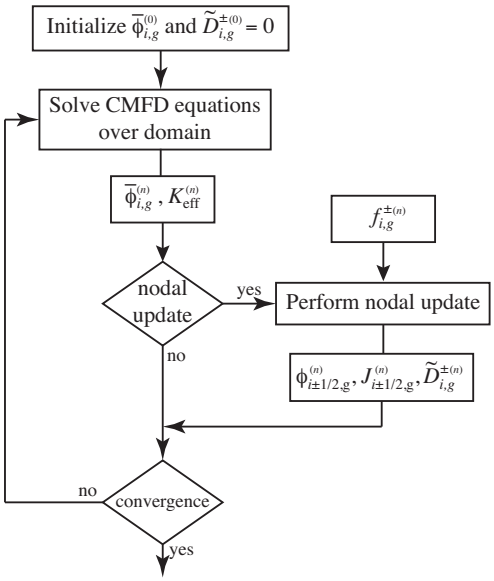

Figure 3: The nodal update procedure.

The two-nodes matrix equations are used to evaluate the neutron flux and current between two internal nodes. We first present the calculation of $\phi_{i\pm1/2,g}^{\mp}$ and $J_{i\pm1/2,g}$ for two adjacent nodes along the $X-$axis.

2.5.1 NEM update procedure

If the nodal expansion method is used, the order of the *two-node* matrix system is equal to $8G$, where G is the number of energy groups. It is written

$$\begin{bmatrix} \mathbb{A}_{1,1} & \mathbb{O} & \mathbb{O} & \mathbb{O} \\ \mathbb{O} & \mathbb{A}_{2,2} & \mathbb{O} & \mathbb{O} \\ \mathbb{O} & \mathbb{O} & \mathbb{A}_{3,3} & \mathbb{A}_{3,4} \\ \mathbb{A}_{4,1} & \mathbb{A}_{4,2} & \mathbb{A}_{4,3} & \mathbb{A}_{4,4} \end{bmatrix} \begin{bmatrix} \boldsymbol{a}_{i,g}^{\text{even}} \\ \boldsymbol{a}_{i+1,g}^{\text{even}} \\ \boldsymbol{a}_{i,g}^{\text{odd}} \\ \boldsymbol{a}_{i+1,g}^{\text{odd}} \end{bmatrix} = \begin{bmatrix} \boldsymbol{S}_{i,g}^{\text{even}} \\ \boldsymbol{S}_{i+1,g}^{\text{even}} \\ \boldsymbol{S}_{i,i+1,g}^{\text{odd}} \\ \boldsymbol{S}_{i,i+1,g}^{\text{cont}} \end{bmatrix} \tag{2.112}$$

where $\mathbb{O}$ are $2G \times 2G$ null matrices and where the unknowns are defined as $\boldsymbol{a}_{i,g}^{\text{even}} = \text{col}\{a_{2,i,g}, a_{4,i,g}\}$ and $\boldsymbol{a}_{i,g}^{\text{odd}} = \text{col}\{a_{1,i,g}, a_{3,i,g}\}$. In the case of a pure polynomial expansion with $G = 2$, the sub-matrix components in Eq. (2.112) are recovered from Eqs. (2.30) to (2.47) as

$$\mathbb{A}_{1,1} = \begin{bmatrix} -\frac{6D_{i,1}}{2\Delta x_i^2} & -\frac{2D_{i,1}}{5\Delta x_i^2} & 0 & 0 \\ 0 & 0 & -\frac{6D_{i,2}}{2\Delta x_i^2} & -\frac{2D_{i,2}}{5\Delta x_i^2} \\ \frac{A_{i,1}}{20} & -\frac{D_{i,1}}{5\Delta x_i^2} - \frac{A_{i,1}}{700} & -\frac{Q_{i,1\leftarrow2}}{20} & \frac{Q_{i,1\leftarrow2}}{700} \\ -\frac{Q_{i,2\leftarrow1}}{20} & \frac{Q_{i,2\leftarrow1}}{700} & \frac{A_{i,2}}{20} & -\frac{D_{i,2}}{5\Delta x_i^2} - \frac{A_{i,2}}{700} \end{bmatrix}, \tag{2.113}$$

$$\mathbb{A}_{2,2} = \begin{bmatrix} -\frac{6D_{i+1,1}}{2\Delta x_{i+1}^2} & -\frac{2D_{i+1,1}}{5\Delta x_{i+1}^2} & 0 & 0 \\ 0 & 0 & -\frac{6D_{i+1,2}}{2\Delta x_{i+1}^2} & -\frac{2D_{i+1,2}}{5\Delta x_{i+1}^2} \\ \frac{A_{i+1,1}}{20} & -\frac{D_{i+1,1}}{5\Delta x_{i+1}^2} - \frac{A_{i+1,1}}{700} & -\frac{Q_{i+1,1\leftarrow 2}}{20} & \frac{Q_{i+1,1\leftarrow 2}}{700} \\ -\frac{Q_{i+1,2\leftarrow 1}}{20} & \frac{Q_{i+1,2\leftarrow 1}}{700} & \frac{A_{i+1,2}}{20} & -\frac{D_{i+1,2}}{5\Delta x_{i+1}^2} - \frac{A_{i+1,2}}{700} \end{bmatrix}, \tag{2.114}$$

$$\mathbb{A}_{3,3} = \begin{bmatrix} \frac{A_{i,1}}{12} & -\frac{D_{i,1}}{2\Delta x_i^2} + \frac{A_{i,1}}{120} & -\frac{Q_{i,1\leftarrow 2}}{12} & \frac{Q_{i,1\leftarrow 2}}{120} \\ -\frac{Q_{i,2\leftarrow 1}}{12} & \frac{Q_{i,2\leftarrow 1}}{120} & \frac{A_{i,2}}{12} & -\frac{D_{i,2}}{2\Delta x_i^2} + \frac{A_{i,2}}{120} \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}, \tag{2.115}$$

$$\mathbb{A}_{3,4} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ \frac{A_{i+1,1}}{12} & -\frac{D_{i+1,1}}{2\Delta x_{i+1}^2} + \frac{A_{i+1,1}}{120} & -\frac{Q_{i+1,1\leftarrow 2}}{12} & \frac{Q_{i+1,1\leftarrow 2}}{120} \\ -\frac{Q_{i+1,2\leftarrow 1}}{12} & \frac{Q_{i+1,2\leftarrow 1}}{120} & \frac{A_{i+1,2}}{12} & -\frac{D_{i+1,2}}{2\Delta x_{i+1}^2} + \frac{A_{i+1,2}}{120} \end{bmatrix}, \tag{2.116}$$

$$\mathbb{A}_{4,1} = \begin{bmatrix} \frac{3D_{i,1}}{\Delta x_i} & \frac{D_{i,1}}{5\Delta x_i} & 0 & 0 \\ 0 & 0 & \frac{3D_{i,2}}{\Delta x_i} & \frac{D_{i,2}}{5\Delta x_i} \\ -\frac{f_{i,1}^+}{2} & 0 & 0 & 0 \\ 0 & 0 & -\frac{f_{i,2}^+}{2} & 0 \end{bmatrix}, \tag{2.117}$$

$$\mathbb{A}_{4,2} = \begin{bmatrix} \frac{3D_{i+1,1}}{\Delta x_{i+1}} & \frac{D_{i+1,1}}{5\Delta x_{i+1}} & 0 & 0 \\ 0 & 0 & \frac{3D_{i+1,2}}{\Delta x_{i+1}} & \frac{D_{i+1,2}}{5\Delta x_{i+1}} \\ \frac{f_{i+1,1}^-}{2} & 0 & 0 & 0 \\ 0 & 0 & \frac{f_{i+1,2}^-}{2} & 0 \end{bmatrix}, \tag{2.118}$$

$$\mathbb{A}_{4,3} = \begin{bmatrix} -\frac{D_{i,1}}{\Delta x_i} & -\frac{D_{i,1}}{2\Delta x_i} & 0 & 0 \\ 0 & 0 & -\frac{D_{i,2}}{\Delta x_i} & -\frac{D_{i,2}}{2\Delta x_i} \\ \frac{f_{i,1}^+}{2} & 0 & 0 & 0 \\ 0 & 0 & \frac{f_{i,2}^+}{2} & 0 \end{bmatrix} \tag{2.119}$$

and

$$\mathbb{A}_{4,4} = \begin{bmatrix} \frac{D_{i+1,1}}{\Delta x_{i+1}} & \frac{D_{i+1,1}}{2\Delta x_{i+1}} & 0 & 0 \\ 0 & 0 & \frac{D_{i+1,2}}{\Delta x_{i+1}} & \frac{D_{i+1,2}}{2\Delta x_{i+1}} \\ \frac{f_{i+1,1}^-}{2} & 0 & 0 & 0 \\ 0 & 0 & \frac{f_{i+1,2}^-}{2} & 0 \end{bmatrix} \tag{2.120}$$

where

$$A_{i,g} = \Sigma_{\mathrm{r},i,g} - \frac{\chi_{i,g}}{K_{\mathrm{eff}}} \nu\Sigma_{\mathrm{f},i,g}, \tag{2.121}$$

$$Q_{i,g\leftarrow h} = \Sigma_{\mathrm{s},i,g\leftarrow h} + \frac{\chi_{i,g}}{K_{\mathrm{eff}}} \nu\Sigma_{\mathrm{f},i,h}, \tag{2.122}$$

$$\boldsymbol{S}_{i,g}^{\mathrm{even}} = \begin{bmatrix} -\Sigma_{\mathrm{r},i,1}\,\bar{\phi}_{i,1} - \bar{L}_{i,1}^{yz} + \Sigma_{\mathrm{s},i,1\leftarrow 2}\,\bar{\phi}_{i,2} + \frac{\chi_{i,1}}{K_{\mathrm{eff}}} \sum_{h=1}^{G} \nu\Sigma_{\mathrm{f},i,h}\,\bar{\phi}_{i,h} \\ -\Sigma_{\mathrm{r},i,2}\,\bar{\phi}_{i,2} - \bar{L}_{i,2}^{yz} + \Sigma_{\mathrm{s},i,2\leftarrow 1}\,\bar{\phi}_{i,1} + \frac{\chi_{i,2}}{K_{\mathrm{eff}}} \sum_{h=1}^{G} \nu\Sigma_{\mathrm{f},i,h}\,\bar{\phi}_{i,h} \\ -\frac{\bar{L}_{i,1}^{yz}}{20} \\ -\frac{\bar{L}_{i,2}^{yz}}{20} \end{bmatrix}, \tag{2.123}$$

$$\boldsymbol{S}^{\text{odd}}_{i,i+1g} = \begin{bmatrix} -\frac{\bar{L}^{yz}_{i,1}}{12} \\ -\frac{\bar{L}^{yz}_{i,2}}{12} \\ -\frac{\bar{L}^{yz}_{i+1,1}}{12} \\ -\frac{\bar{L}^{yz}_{i+1,2}}{12} \end{bmatrix}, \tag{2.124}$$

and

$$\boldsymbol{S}^{\text{cont}}_{i,i+1,g} = \begin{bmatrix} 0 \\ 0 \\ -f^{+}_{i,1}\,\bar{\phi}_{i,1} + f^{-}_{i+1,1}\,\bar{\phi}_{i+1,1} \\ -f^{+}_{i,2}\,\bar{\phi}_{i,2} + f^{-}_{i+1,2}\,\bar{\phi}_{i+1,2} \end{bmatrix}. \tag{2.125}$$

The optimal sequence for solving the linear system of Eq. (2.112) is

$$\begin{aligned} \boldsymbol{a}^{\text{even}}_{i,g} &= \mathbb{A}^{-1}_{1,1}\,\boldsymbol{S}^{\text{even}}_{i,g} \\ \boldsymbol{a}^{\text{even}}_{i+1,g} &= \mathbb{A}^{-1}_{2,2}\,\boldsymbol{S}^{\text{even}}_{i+1,g} \\ \begin{bmatrix} \boldsymbol{a}^{\text{odd}}_{i,g} \\ \boldsymbol{a}^{\text{odd}}_{i+1,g} \end{bmatrix} &= \begin{bmatrix} \mathbb{A}_{3,3} & \mathbb{A}_{3,4} \\ \mathbb{A}_{4,3} & \mathbb{A}_{4,4} \end{bmatrix}^{-1} \begin{bmatrix} \boldsymbol{S}^{\text{odd}}_{i,i+1,g} \\ \boldsymbol{S}^{\text{cont}}_{i,i+1,g} - \mathbb{A}_{4,1}\,\boldsymbol{a}^{\text{even}}_{i,g} - \mathbb{A}_{4,2}\,\boldsymbol{a}^{\text{even}}_{i+1,g} \end{bmatrix}. \end{aligned} \tag{2.126}$$

The *one-node* matrix equations are used to evaluate the neutron flux and current on a domain boundary. We next present the calculation of $\phi_{i\pm1/2,g}$ and $J_{i\pm1/2,g}$ for a boundary surface $i \pm 1/2$ along the $X-$axis. The order of the matrix system is equal to $4G$, where G is the number of energy groups. It is written

$$\begin{bmatrix} \mathbb{A}_{1,1} & \mathbb{O} \\ \mathbb{O} & \mathbb{B}_{2,2} \\ \mathbb{B}_{3,1} & \mathbb{B}_{3,2} \end{bmatrix} \begin{bmatrix} \boldsymbol{a}^{\text{even}}_{i,g} \\ \boldsymbol{a}^{\text{odd}}_{i,g} \end{bmatrix} = \begin{bmatrix} \boldsymbol{S}^{\text{even}}_{i,g} \\ \boldsymbol{S}^{\text{odd}}_{i,g} \\ \boldsymbol{S}^{\text{bc}}_{i,g} \end{bmatrix} \tag{2.127}$$

In the case of a pure polynomial expansion with $G = 2$, the sub-matrix components in Eq. (2.127) are recovered from Eqs. (2.30) to (2.17) as

$$\mathbb{B}_{2,2} = \begin{bmatrix} \frac{A_{i,1}}{12} & -\frac{D_{i,1}}{2\Delta x_i^2} + \frac{A_{i,1}}{120} & -\frac{Q_{i,1\leftarrow 2}}{12} & \frac{Q_{i,1\leftarrow 2}}{120} \\ -\frac{Q_{i,2\leftarrow 1}}{12} & \frac{Q_{i,2\leftarrow 1}}{120} & \frac{A_{i,2}}{12} & -\frac{D_{i,2}}{2\Delta x_i^2} + \frac{A_{i,2}}{120} \end{bmatrix} \tag{2.128}$$

with

$$\boldsymbol{S}^{\text{odd}}_{i,g} = \begin{bmatrix} -\frac{\bar{L}^{yz}_{i,1}}{12} \\ -\frac{\bar{L}^{yz}_{i,2}}{12} \end{bmatrix}. \tag{2.129}$$

Three different types of boundary conditions are now considered:

zero net current

$$\mathbb{B}_{3,1} = \begin{bmatrix} 3 & \frac{1}{5} & 0 & 0 \\ 0 & 0 & 3 & \frac{1}{5} \end{bmatrix}, \tag{2.130}$$

$$\mathbb{B}_{3,2} = \begin{bmatrix} \mp 1 & \mp\frac{1}{2} & 0 & 0 \\ 0 & 0 & \mp 1 & \mp\frac{1}{2} \end{bmatrix} \tag{2.131}$$

and

$$\boldsymbol{S}^{\text{bc}}_{i,g} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \tag{2.132}$$

where the minus sign in Eq. (2.131) is used for the left boundary condition.

albedo condition

$$\mathbb{B}_{3,1} = \begin{bmatrix} \left(\frac{\Lambda_1^{x\mp}}{2} + \frac{3D_{i,1}}{\Delta x_i}\right) & \frac{D_{i,1}}{5\Delta x_i} & 0 & 0 \\ 0 & 0 & \left(\frac{\Lambda_2^{x\mp}}{2} + \frac{3D_{i,2}}{\Delta x_i}\right) & \frac{D_{i,2}}{5\Delta x_i} \end{bmatrix}, \tag{2.133}$$

$$\mathbb{B}_{3,2} = \begin{bmatrix} \mp\left(\frac{\Lambda_1^{x\mp}}{2} + \frac{D_{i,1}}{2\Delta x_i}\right) & \mp\frac{D_{i,1}}{2\Delta x_i} & 0 & 0 \\ 0 & 0 & \mp\left(\frac{\Lambda_2^{x\mp}}{2} + \frac{D_{i,2}}{\Delta x_i}\right) & \mp\frac{D_{i,2}}{2\Delta x_i} \end{bmatrix} \tag{2.134}$$

and

$$\boldsymbol{S}_{i,g}^{\text{bc}} = \begin{bmatrix} -(\Lambda_1^{x\mp})\bar{\phi}_{i,1} \\ -(\Lambda_2^{x\mp})\bar{\phi}_{i,2} \end{bmatrix} \tag{2.135}$$

where the minus sign in Eq. (2.134) is used for the left boundary condition.

zero flux

$$\mathbb{B}_{3,1} = \begin{bmatrix} \frac{1}{2} & 0 & 0 & 0 \\ 0 & 0 & \frac{1}{2} & 0 \end{bmatrix}, \tag{2.136}$$

$$\mathbb{B}_{3,2} = \begin{bmatrix} \mp\frac{1}{2} & 0 & 0 & 0 \\ 0 & 0 & \mp\frac{1}{2} & 0 \end{bmatrix} \tag{2.137}$$

and

$$\boldsymbol{S}_{i,g}^{\text{bc}} = \begin{bmatrix} -\bar{\phi}_{i,1} \\ -\bar{\phi}_{i,2} \end{bmatrix} \tag{2.138}$$

where the minus sign in Eq. (2.137) is used for the left boundary condition.

The optimal sequence for solving the linear system of Eq. (2.127) is

$$\begin{aligned} \boldsymbol{a}_{i,g}^{\text{even}} &= \mathbb{A}_{1,1}^{-1}\, \boldsymbol{S}_{i,g}^{\text{even}} \\ \boldsymbol{a}_{i,g}^{\text{odd}} &= \begin{bmatrix} \mathbb{B}_{2,2} \\ \mathbb{B}_{3,2} \end{bmatrix}^{-1} \begin{bmatrix} \boldsymbol{S}_{i,g}^{\text{odd}} \\ \boldsymbol{S}_{i,g}^{\text{bc}} - \mathbb{B}_{3,1}\, \boldsymbol{a}_{i,g}^{\text{even}} \end{bmatrix}. \end{aligned} \tag{2.139}$$

Similar matrix equations can be obtained for hyperbolic basis functions. Once the solution of the linear system is obtained for a 1D traverse, interface fluxes and currents can be obtained from Eqs. (2.38) to (2.45).

2.5.2 ANM update procedure

The ANM update procedure is presented for the X-axis. The two-nodes relations corresponding to the analytic nodal method are obtained from Eq. (2.104). The X-directed net current $\boldsymbol{J}_j^y(x_{i-1/2})$ at node interface $x_{i-1/2}$ is obtained as a function of the CMFD neighbour fluxes $\bar{\boldsymbol{\Phi}}_{i-1,j}$ and $\bar{\boldsymbol{\Phi}}_{i,j}$, and of the Y–directed net currents $\boldsymbol{J}_i^x(y_{j\pm1/2})$ taken from the previous nodal correction iteration. Nodal balance equations (2.105) are not used. We write

$$\begin{aligned} \left\{\left[\boldsymbol{f}_{i-1}^{x+}\right]\mathbb{R}_{i-1,j}^{x+} - \left[\boldsymbol{f}_i^{x-}\right]\mathbb{L}_{i,j}^{x+}\right\}\boldsymbol{J}_j^y(x_{i-1/2}) \;=\; &-\left[\boldsymbol{f}_{i-1}^{x+}\right]\left[\mathbb{R}_{i-1,j}^{x-}\bar{\boldsymbol{\Phi}}_{i-1,j} + \mathbb{R}_{i-1,j}^{x,\text{sour}}\begin{pmatrix} \boldsymbol{J}_{i-2}^x(y_{j-1/2}) \\ \boldsymbol{J}_{i-1}^x(y_{j-1/2}) \\ \boldsymbol{J}_i^x(y_{j-1/2}) \\ \boldsymbol{J}_{i-2}^x(y_{j+1/2}) \\ \boldsymbol{J}_{i-1}^x(y_{j+1/2}) \\ \boldsymbol{J}_i^x(y_{j+1/2}) \end{pmatrix}\right] \\ &+ \left[\boldsymbol{f}_i^{x-}\right]\left[\mathbb{L}_{i,j}^{x-}\bar{\boldsymbol{\Phi}}_{i,j} + \mathbb{L}_{i,j}^{x,\text{sour}}\begin{pmatrix} \boldsymbol{J}_{i-1}^x(y_{j-1/2}) \\ \boldsymbol{J}_i^x(y_{j-1/2}) \\ \boldsymbol{J}_{i+1}^x(y_{j-1/2}) \\ \boldsymbol{J}_{i-1}^x(y_{j+1/2}) \\ \boldsymbol{J}_i^x(y_{j+1/2}) \\ \boldsymbol{J}_{i+1}^x(y_{j+1/2}) \end{pmatrix}\right]. \end{aligned} \tag{2.140}$$

If a zero-flux boundary condition is set, boundary currents $\boldsymbol{J}_j^y(x_{-1/2})$ and $\boldsymbol{J}_j^y(x_{I-1/2})$ are given by Eqs. (2.108) and (2.109), respectively. If an albedo boundary condition is set, boundary currents $\boldsymbol{J}_j^y(x_{-1/2})$ and $\boldsymbol{J}_j^y(x_{I-1/2})$ are given by Eqs. (2.110) and (2.111), respectively.

After the evaluation of X-directed net currents $\boldsymbol{J}_j^y(x_{i\pm1/2})$ is completed, boundary fluxes are obtained using Eqs. (2.100) and (2.101).

2.5.3 Update of the drift coefficients

The updated drift coefficients $\tilde{D}_{i,g}^{\pm}$ between two internal nodes are finally obtained from Eqs. (2.13) and (2.14) as

$$\tilde{D}_{i,g}^{+} = \begin{cases} -\dfrac{J_{i+1/2,g} + D_{i,g}^{+}\left(\bar{\phi}_{i+1,g} - \bar{\phi}_{i,g}\right)}{\phi_{i+1,g} + \phi_{i,g}} & \text{if } i < I \\ -\dfrac{J_{i+1/2,g} - D_{i,g}^{+}\bar{\phi}_{i,g}}{\phi_{i,g}} & \text{if } i = I \end{cases} \tag{2.141}$$

and

$$\tilde{D}_{i,g}^{-} = \begin{cases} -\dfrac{J_{i-1/2,g} + D_{i,g}^{-}\left(\bar{\phi}_{i,g} - \bar{\phi}_{i-1,g}\right)}{\bar{\phi}_{i,g} + \bar{\phi}_{i-1,g}} & \text{if } i > 1 \\ -\dfrac{J_{i-1/2,g} + D_{i,g}^{-}\bar{\phi}_{i,g}}{\bar{\phi}_{i,g}} & \text{if } i = 1\ . \end{cases} \tag{2.142}$$

where $D_{i,g}^{\pm}$ is defined by Eqs. (2.15) and (2.16).

2.6 Intranodal flux reconstruction

The nodal expansion for the solution of the diffusion equation is made in one dimension for each of the directions x, y and z. The crossed terms are not considered but are taken into account implicitly in the processing of the transverse leakage terms, according to the formulation made in Sects. 2.2 and 2.4. Determination of the homogeneous intranodal flux is required for *fine power reconstruction*. Intranodal flux reconstruction is performed after convergence of the NEM or ANM. This capability is available in module `VAL:` of Trivac5 and in module `IDET:` of Donjon5. In 2D and 3D cases, we perform a polynomial flux reconstruction using the following polynomials along each direction:

$$\begin{aligned} p_1(u) &= u \\ p_2(u) &= 3u^2 - \frac{1}{4} \\ p_3(u) &= \left(u^2 - \frac{1}{4}\right) u \\ p_4(u) &= \left(u^2 - \frac{1}{4}\right)\left(u^2 - \frac{1}{20}\right) \end{aligned} \tag{2.143}$$

where $-1/2 \leq u \leq 1/2$.

The procedure consists of the following steps:

1. Perfom an additive polynomial reconstruction preserving all information contained in the NEM or ANM solution in each energy group: averaged fluxes $\bar{\phi}_{i,j,k}$, surface fluxes $\phi_{i\pm1/2,j,k}^{\mp}$, $\phi_{i,j\pm1/2,k}^{\mp}$, $\phi_{i,j,k\pm1/2}^{\mp}$ and surface currents $J_{i\pm1/2,j,k}^{x}$, $J_{i,j\pm1/2,k}^{y}$, $J_{i,j,k\pm1/2}^{z}$.
2. Calculate the averaged corner point flux values.
3. Perform a polynomial *corner flux correction*.

The polynomial expansion of step 1 is based on the following 3D expansion:

$$\phi^{\text{step1}}(x,y,z) = \bar{\phi}_{i,j,k} + \sum_{\ell=1}^{4} a_{\ell,i,j,k}^{x}\, p_\ell(u) + \sum_{\ell=1}^{4} a_{\ell,i,j,k}^{y}\, p_\ell(v) + \sum_{\ell=1}^{4} a_{\ell,i,j,k}^{z}\, p_\ell(w) \tag{2.144}$$

where $x_{i-1/2} < x < x_{i+1/2}$, $y_{j-1/2} < y < y_{j+1/2}$ and $z_{k-1/2} < z < z_{k+1/2}$
and where u, v and w are the reduced coordinates defined over the support $(-1/2,\, 1/2)$ as

$$u = \frac{x}{\Delta x_i} - \frac{1}{2},\ v = \frac{y}{\Delta y_j} - \frac{1}{2} \text{ and } w = \frac{z}{\Delta z_k} - \frac{1}{2}. \tag{2.145}$$

Coefficients $a^x_{\ell,i,j,k}$, $a^y_{\ell,i,j,k}$ and $a^z_{\ell,i,j,k}$ are the solution of three order–4 linear systems using information of the NEM or ANM solution in each node.

The reconstructed flux $\phi^{\text{step1}}(x,y,z)$ presents important discontinuities along the node boundaries, with maximum discrepancies at node corners. An internal corner point is surrounded by 8 nodes, similar to those depicted in Fig. 4. An averaged corner point flux value at node corner 1 can be obtain as

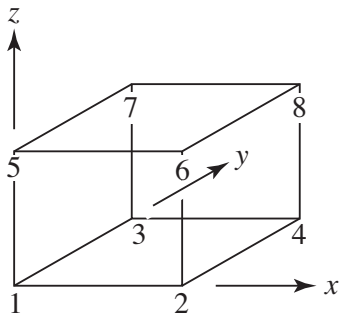

Figure 4: 3D node corner numerotation.

$$\bar{\phi}^{\text{corn1}}_{i,j,k} = \frac{1}{8}\left(\phi^{\text{corn1}}_{i,j,k} + \phi^{\text{corn2}}_{i-1,j,k} + \phi^{\text{corn3}}_{i,j-1,k} + \phi^{\text{corn4}}_{i-1,j-1,k} + \phi^{\text{corn5}}_{i,j,k-1} + \phi^{\text{corn6}}_{i-1,j,k-1} + \phi^{\text{corn7}}_{i,j-1,k-1} + \phi^{\text{corn8}}_{i-1,j-1,k-1}\right). \tag{2.146}$$

Similar expressions can be written for each node corner, taking care of the fact that boundary nodes have a number of neighbours smaller than 8.

The third step consists to correct the polynomial expansion of Eq. (2.144) so as to fit averaged corner point flux value obtained in second step. The corrected expansion is written

$$\phi^{\text{step3}}(x,y,z) = \phi^{\text{step1}}(x,y,z) + \sum_{n=1}^{2}\sum_{m=1}^{2}\sum_{\ell=1}^{2} a^{\text{c}}_{\ell,m,n,i,j,k}\, p_\ell(u)\, p_m(v)\, p_n(w) \tag{2.147}$$

where coefficients $a^{\text{c}}_{\ell,m,n,i,j,k}$ are the solution of three order–8 linear system in each node. The solution of this linear system is

$$\begin{bmatrix} a^{\text{c}}_{1,1,1,i,j,k} \\ a^{\text{c}}_{2,1,1,i,j,k} \\ a^{\text{c}}_{1,2,1,i,j,k} \\ a^{\text{c}}_{2,2,1,i,j,k} \\ a^{\text{c}}_{1,1,2,i,j,k} \\ a^{\text{c}}_{2,1,2,i,j,k} \\ a^{\text{c}}_{1,2,2,i,j,k} \\ a^{\text{c}}_{2,2,2,i,j,k} \end{bmatrix} = \begin{bmatrix} -1 & 1 & 1 & -1 & 1 & -1 & -1 & 1 \\ 1 & 1 & -1 & -1 & -1 & -1 & 1 & 1 \\ 1 & -1 & 1 & -1 & -1 & 1 & -1 & 1 \\ -1 & -1 & -1 & -1 & 1 & 1 & 1 & 1 \\ 1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 \\ -1 & -1 & 1 & 1 & -1 & -1 & 1 & 1 \\ -1 & 1 & -1 & 1 & -1 & 1 & -1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} \bar{\phi}^{\text{corn1}}_{i,j,k} - \phi^{\text{corn1}}_{i,j,k} \\ \bar{\phi}^{\text{corn2}}_{i,j,k} - \phi^{\text{corn2}}_{i,j,k} \\ \bar{\phi}^{\text{corn3}}_{i,j,k} - \phi^{\text{corn3}}_{i,j,k} \\ \bar{\phi}^{\text{corn4}}_{i,j,k} - \phi^{\text{corn4}}_{i,j,k} \\ \bar{\phi}^{\text{corn5}}_{i,j,k} - \phi^{\text{corn5}}_{i,j,k} \\ \bar{\phi}^{\text{corn6}}_{i,j,k} - \phi^{\text{corn6}}_{i,j,k} \\ \bar{\phi}^{\text{corn7}}_{i,j,k} - \phi^{\text{corn7}}_{i,j,k} \\ \bar{\phi}^{\text{corn8}}_{i,j,k} - \phi^{\text{corn8}}_{i,j,k} \end{bmatrix}. \tag{2.148}$$

The effect of corner flux correction is depicted in Fig. 5 for a 2D case. The result of polynomial reconstruction based on Eqs. (2.144) (on left) and (2.147) (on right) are compared.

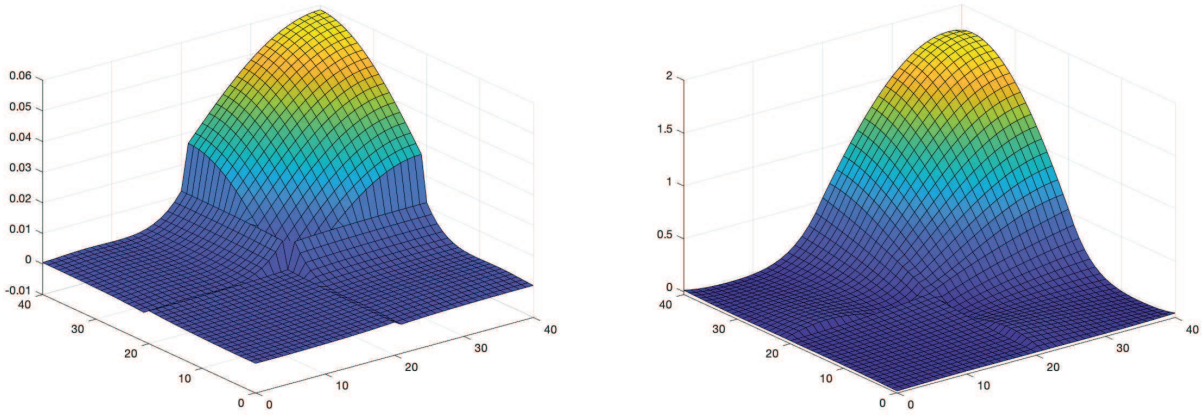

Figure 5: Polynomial reconstruction in 2D.

3 Input data specifications

3.1 Syntactic rules for input data specifications

The input data to any module is read in free format using the subroutine `REDGET`. The rules for specifying the input data are therefore given in this section. The users guide was written using the following conventions:

- the parameters surrounded by single square brackets '[ ]' denote an optional input;
- the parameters surrounded by double square brackets '[[ ]]' denote an optional input which may be repeated as many times as desired;
- the parameters in braces separated by vertical bars '{ | | }' denote a choice of input where (one and *only* one is mandatory);
- the parameters in **bold face** and in brackets '( )' denote an input structure;
- the parameters in italics and in brackets with an index '(*data*(i), i=1,n)' denote a set of n inputs;
- the words using the typewriter font are character constants `keywordS` used as keywords;
- the words in italics are user defined variables, they should be lower case and are of type integer (starting with *i* to *n*) and real (starting with *a* to *h* or *o* to *z*) or of type character in uppercase *CHARACTER*.

3.2 The global input structure

Brisingr is built around the GAN generalized driver.[2] Input data must therefore follow the calling specifications given below:

Table 1: Structure **(Brisingr)**

[`LINKED_LIST` [[*NAME1*]] `;`]
[`XSM_FILE` [[*NAME2*]] `;`]
[`SEQ_BINARY` [[*NAME3*]] `;`]
[`SEQ_ASCII` [[*NAME4*]] `;`]
[`HDF5_FILE` [[*NAME5*]] `;`]
[`MODULE` [[*NAME6*]] `;`]
[[**(specif)**]]
`END: ;`

where

NAME1 `Character*12` name of a LCM object.

NAME2 `Character*12` name of an XSM file.

NAME3 `Character*12` name of a sequential binary file.

NAME4 `Character*12` name of a sequential ASCII file.

NAME5 `Character*12` name of a HDF5 file.

NAME6 `Character*12` name of a module.

(specif) Input specifications for a single module. Specifications for Brisingr modules will be given in the following sections.

The input data always begins with the declaration of each LCM object, XSM file, sequential (binary or ASCII) file or HDF5 file that will be required by the following modules. This is followed by the declaration of the modules actually used in the input data deck. The following data describes a sequence of module calls, in the format of the GAN generalized driver. As indicated in Fig. 6, the modules communicate with each other through LCM objects or XSM files whose specifications are given in section 2. The Brisingr user generally has the choice to declare its data structures as `LINKED_LIST` to reduce CPU time resources or as `XSM_FILE` to reduce CPU memory resources.

The input data always ends with a call to the `END:` module.

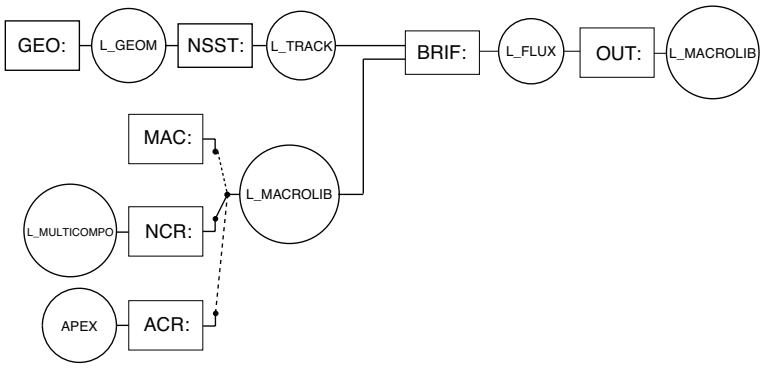

Figure 6: The Brisingr modular approach.

3.3 The `BRIF:` module

The `BRIF:` module is used to compute the solution to an eigenvalue problem corresponding to a nodal expansion method (NEM) discretization. The actual implementation is limited to 1D/2D/3D Cartesian geometries. The calling specifications are:

Table 2: Structure **(BRIF:)**

FLUX `:=` `BRIF:` *TRACK MACRO* `::` **(BRIF_data)**

where

FLUX `character*12` name of the LCM object (type `L_FLUX`) containing the solution.

TRACK `character*12` name of the LCM object (type `L_TRACK`) containing the TRACKING.

MACRO `character*12` name of the LCM object (type `L_MACROLIB`) containing the cross sections, diffusion coefficients and discontinuity factors.

(BRIF_data) structure containing the data to module `BRIF:` (see Sect. 3.3.1).

3.3.1 Data input for module `BRIF:`

Table 3: Structure **(BRIF_data)**

```
[ EDIT iprint ]
[ NUPD max_no_nodal_iter ]
[ EXTE [ max_no_outer_iter ] [ outer_tol ] ]
[ INNE [ max_no_inner_iter ] [ inner_tol ] ]
[ THER [ max_no_group_iter ] [ group_tol ] ]
[ NODA { NEM | ANM | OFF } ]
[ GEOM { colorset | general } ]
[ ADJ ] [ { NODF | SELE } ]
[ CMFD [ { all_groups | one_group } ] [ Wielandt { ON | OFF } ]
      [ Krylov { bicgstab | cg | jacobi | ldu } ] ]
[ VOID { Mark | Marshak } ]
[ LEAK { flat | linear | quadratic } ]
[ FREQ ev_nodal_freq ]
[ BUCK valb2 ]
;
```

where

`EDIT` keyword used to set *iprint*.

iprint index used to control the printing in module `BRIF:`. =0 for no print; =1 for minimum printing (default value).

`NUPD` keyword to specify the maximum number of nodal update iterations.

max_no_nodal_iter maximum number of nodal update iterations iterations. The fixed default value is *max_no_outer_iter* = 300.

`EXTE` keyword to specify that the control parameters for the K_{eff} iteration are to be modified.

max_no_outer_iter maximum number of K_{eff} iterations. The fixed default value is *max_no_outer_iter* = 300.

outer_tol convergence criterion for the K_{eff} iterations. The fixed default value is *outer_tol* = 1.0×10^{-8}.

`INNER` keyword to specify that the control parameters for the inner iteration are to be modified. This information is used by the Krylov solver.

max_no_inner_iter maximum number of inner iterations. The fixed default value is *max_no_inner_iter* = 300.

inner_tol convergence criterion for the inner iterations. The fixed default value is *inner_tol* = 1.0×10^{-8}.

`THER` keyword to specify that the control parameters for the thermal upscattering iteration are to be modified.

max_no_group_iter maximum number of thermal upscattering iterations. The fixed default value is *max_no_group_iter* = 300.

group_tol convergence criterion for the thermal upscattering iterations. The fixed default value is *group_tol* $= 1.0 \times 10^{-6}$.

`NODA` keyword used to set the type of nodal update.

`NEM` nodal expansion method (default value).

`ANM` analytic nodal method.

`OFF` nNo nodal updatre performed. A solution of a pure coarse-mesh finite difference (CMFD) equation is obtained.

`GEOM` keyword used to set how discontinuity factors are used.

`colorset` apply discontinuity factors for colorset calculations.

`general` general rule of application (default value).

`ADJ` keyword used to fperform an adjoint calculation. By default, a direct solution is obtained.

`NODF` keyword used to force discontinuity factors to one.

`SELE` keyword used to replace discontinuity factors with SPH factors. Discontinuity factors are set to one and macroscopic cross sections, diffusion coefficients and physical albedos are SPH-corrected. This option is only available in cases where the nodes have equal discontinuity factors on their sides.

`CMFD` keyword to specify various options of the coarse-mesh finite difference (CMFD) calculation

`all_groups` keyword to specify that the inner problem is solved simultanously for all energy groups (default option).

`one_groups` keyword to specify that the inner problem is solved one group a time.

`Wielandt` keyword to activate/desactivate (`ON`/`OFF`) the acceleration of K_{eff} iterations using the Wielandt method (`ON` is set by default).

`Krylov` keyword to set the type of preconditionning used bt the internal Krylov solver.

`bicgstab` Biconjugate gradient stabilized method (default option).

`cg` Conjugate gradient method.

`jacobi` Jacobi method.

`ldu` Full or incomplete factorization of the sparse matricces.

`VOID` keyword to specify the type of albedo or vacuum boundary condition.

`Marshak` Marshak boundary condition (default option).

`Mark` Mark boundary condition.

`LEAK` keyword to specify the type of transverse leakage approximation.

`flat` flat leakage approximation.

`linear` quadratic leakage approximation in the internal nodes and linear leakage approximation in the boundary nodes.

`quadratic` quadratic leakage approximation in all the nodes, including boundary nodes (default option).

`FREQ` keyword to specify the frequency of nodal updates.

ev_nodal_freq number of external iterations before performing a nodal update. The fixed default value is *ev_nodal_freq* = 0.

`BUCK` keyword used to specify the fixed buckling. By default, *valb2* = 0 cm^{-2}

valb2 value of the fixed total buckling in cm^{-2}.

3.4 The `NODSPH:` module

This module compute a MACROLIB for a 2D *equivalent macro-geometry* based on *discontinuity factor equivalence theory*. A fine-mesh reference calculation (using a fine-group transport calculation) is first performed so as to produce a coarse-group and coarse-mesh Macrolib stored within an output EDITION object (*EDIT_REF*), compatible with the selected reflector model. Module `NODSPH:` recovers the reference GEOMETRY, depicted in Figs. 7 to 9, from object *GEOM_GAP*. The S_n GEOMETRY must have a reflective (`REFL` or `ALBE 1.0`) boundary condition on its left (`X-`) and lower (`Y-`) boundaries. Module `NODSPH:` recovers the following information from each *EDIT_REF* object:

- Coarse group surfacic fluxes between the nodes using averaged flux values recovered into *gap* volumes, corresponding to water blades in fuel assemblies or to tiny meshes in reflector zones. There is no need to define a gap over a symmetry surface, except if this surface is a boundary where an *assembly discontinuity factor* (ADF) is required. However, a gap should be set over a void or albedo $\neq 1$ boundary.
- Coarse group net currents on the heavy line segments between the nodes in Figs. 7 to 9. These values are obtained from a balance relation, assuming reflection on the left and lower boundaries.
- Averaged macroscopic cross sections and diffusion coefficients within the *no-gap* homogenized nodes.

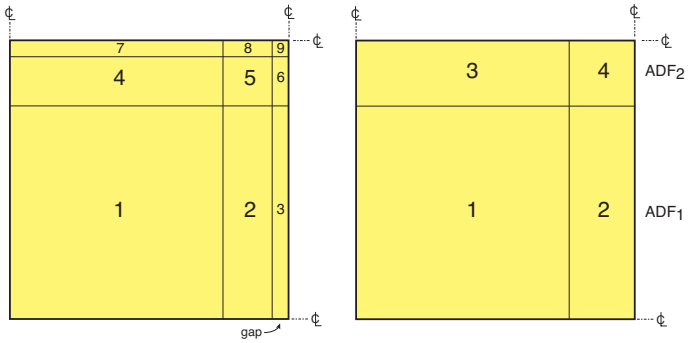

Figure 7: Definition of quarter-assembly macro-geometries (with and without gaps) used by the `NODSPH:` module.

At output, a MACROLIB object is produced with equivalent macroscopic cross sections, diffusion coefficients, discontinuity factors and albedos. A verification calculation is performed over the `NODSPH:` no-gap geometry as depicted in Figs. 7 to 9.

Nodal expansion base functions are used to represent the flux nodal expansion methods. By default, polynomials defined over $(-0.5, 0.5)$ are used as base functions:[5]

$$\begin{aligned} P_0(u) &= 1 \\ P_1(u) &= u \\ P_2(u) &= 3u^2 - \frac{1}{4} \\ P_3(u) &= \left(u^2 - \frac{1}{4}\right) u \\ P_4(u) &= \left(u^2 - \frac{1}{4}\right)\left(u^2 - \frac{1}{20}\right) \end{aligned} \tag{3.1}$$

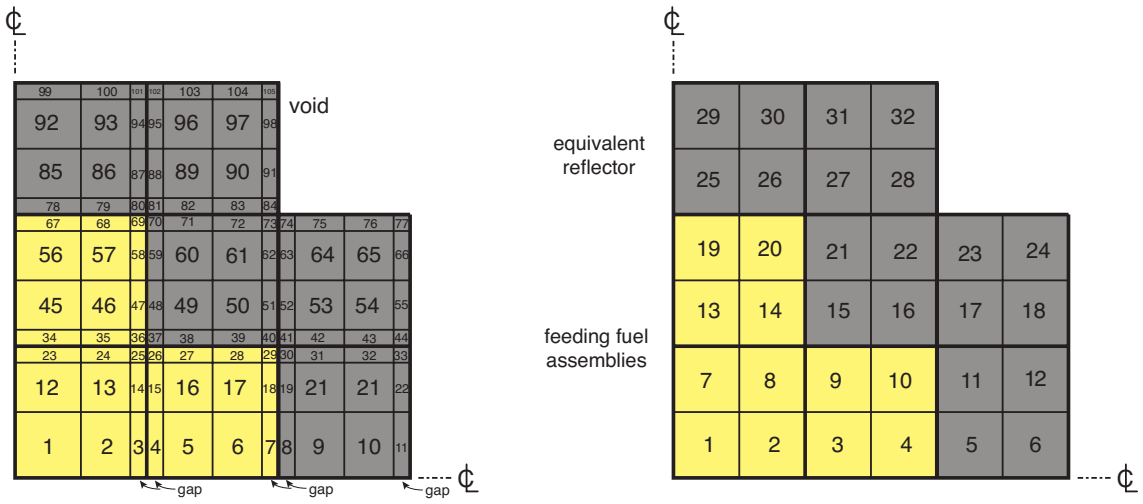

Figure 8: Definition of fuel–reflector macro-geometries (with and without gaps) used by the `NODSPH:` module.

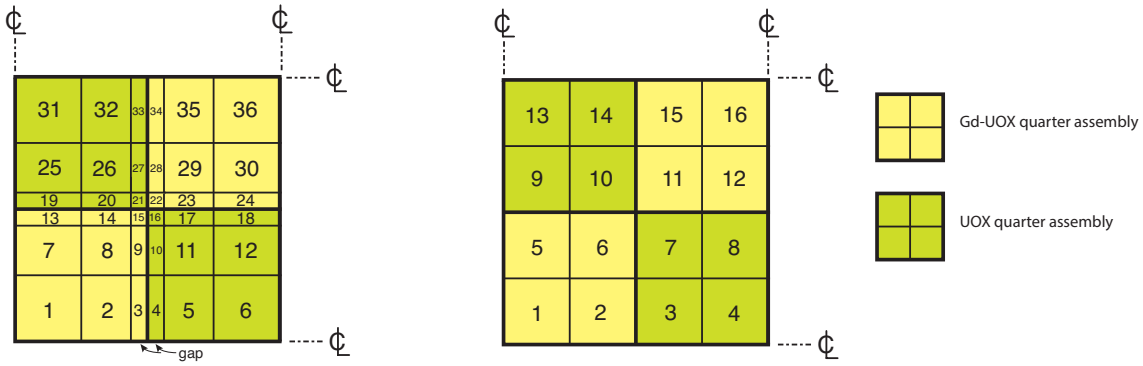

Figure 9: Definition of colorset macro-geometries (with and without gaps) used by the `NODSPH:` module.

There is the option of using hyperbolic functions in some energy groups:

$$\begin{aligned} P_3(u) &= \sinh(\zeta_g u) \\ P_4(u) &= \cosh(\zeta_g u) - \frac{2}{\zeta}\sinh(\zeta_g/2) \end{aligned} \tag{3.2}$$

where

$$\zeta_g = \Delta x \sqrt{\frac{\Sigma_{\mathrm{r},g}}{D_g}} \tag{3.3}$$

where Δx, $\Sigma_{\mathrm{r},g}$ and D_g are the node width (cm), the macroscopic removal cross section (cm^{-1}) and the diffusion coefficient (cm) in group g, respectively.

The calling specifications are:

Table 4: Structure **(NODSPH:)**

GEOM MACRO `:=` `NODSPH:` *GEOM_GAP* [[*EDIT_REF*]] `::` **(NODSPH_data)**

where

GEOM `character*12` name of the nodal GEOMETRY (type `L_GEOM`) object open creation mode. This geometry can be used for performing a verification calculation over the 1D nodal geometry.

MACRO `character*12` name of the nodal MACROLIB (type `L_MACROLIB`) object open in creation mode.

GEOM_GAP `character*12` name of the MACRO-GEOMETRY with gaps (type `L_GEOM`) object open read-only mode. This object describes geometries depicted on left sides of Figs. 7 to 9.

EDIT_REF `character*12` name of a reference EDITION (type `L_EDIT`) object, containing a coarse-group and coarse-mesh MACROLIB for the MACRO-GEOMETRY with gaps.

NODSPH_data input data structure containing specific data (see Section 3.4.1).

3.4.1 Data input for module `NODSPH:`

Table 5: Structure **(NODSPH_data)**

[`EDIT` *iprint*]
[`HYPE` *igmax*]
`GRID` ((*igrid*(*i*,*j*),*j*=1,*LY*),*i*=1,*LX*)
[{ `ALBE` | `NOAL` }] [{ `LEFT` | `RIGH` }]
[`NGET` [(*adf*(*g*), *g*=1,N_g)]]
`;`

where

`EDIT` keyword used to set *iprint*.

iprint index used to control the printing in module `NODSPH:`. =0 for no print; =1 for minimum printing (default value).

`HYPE` keyword used to specify the type of nodal expansion base functions. By default, polynomial base functions are used in all energy groups. This keyword has no effect with the analytic nodal method.

igmax hyperbolic base functions are used for coarse energy groups with indices $\geq$ *igmax*.

`GRID` keyword used to set the nodal mixture indices *igrid*. A $LX \times LY$ matrix follows, where LX and LY are the number of lines and columns of the *MIX* array in the MACROGEOMETRY *GEOM_GAP*.

igrid(i,j) Nodal mixtures indices in non-gap locations. Mixture indices should not be repeated to keep consistency about discontinuity factor orientations. In gap locations, symbols "`|`", "`-`" and "`.`" are used.

The `GRID` data structures corresponding to Figs. 7 to 9 are set as

```
GRID (* quarter-assembly *)
 1 2 |
 3 4 |
 - - .
```

```
GRID (* fuel-reflector *)
  1  2 | |  3  4 | |  5  6 |
  7  8 | |  9 10 | | 11 12 |
  -  - . .  -  - . .  -  - .
 13 14 | | 15 16 | | 17 18 |
 19 20 | | 21 22 | | 23 24 |
  -  - . .  -  - . .  -  - .
 25 26 | | 27 28 | |  0  0 |
 29 30 | | 31 32 | |  0  0 |
  -  - . .  -  - . .  -  - .
```

and

```
GRID (* colorset *)
  1  2 | |  3  4
  5  6 | |  7  8
  -  - . .  -  -
  -  - . .  -  -
  9 10 | | 11 12
 13 14 | | 15 16
```

`ALBE` keyword used to compute an equivalent albedo in each coarse energy group (default option).

`NOAL` keyword used to desactivate equivalent albedo calculation.

`LEFT` keyword used to merge single gaps with the left nodes.

`RIGH` keyword used to merge single gaps with the right nodes (default option).

`NGET` keyword used to force the value of the fuel assembly discontinuity factor at the fuel-reflector interface, as used by the NGET normalization. By default, this value is not modified by NGET normalization.

adf value of the assembly discontinuity factor (ADF) on the fuel-reflector interface in group $g \leq N_g$. If keyword `NGET` is set and *adf* values are not given, the ADF values are recovered from *EDIT_SN*.

4 Examples of input data files

4.1 IAEA-3D benchmark

The IAEA-3D benchmark is defined in Ref. 9 and its geometry is represented in Fig. 10. Here, it is solved using the nodal expansion method with axial mesh set at 0, 20, 63.33, 106.67, 150, 193.33, 236.67, 280, 320, 360 and 380 cm.

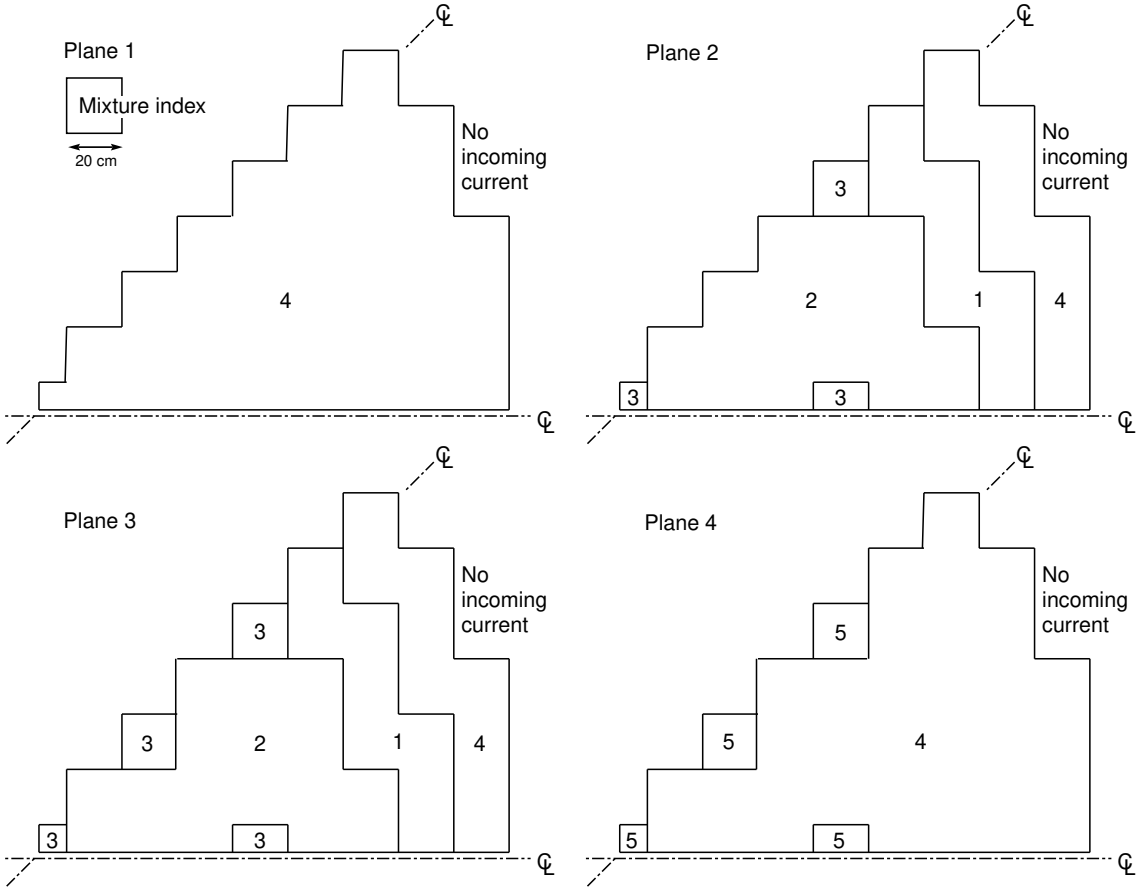

Figure 10: Description of the IAEA-3D benchmark.

```
*----
*  TEST CASE IAEA3D UNFOLDED
*  MACROLIB-DEFINED CROSS SECTIONS
*----
*  Define STRUCTURES and MODULES used
*----
LINKED_LIST IAEA3D TRACK MACRO FLUX REF EDIT ERROR ;
MODULE GEO: NSST: BRIF: MAC: OUT: ERROR: END: ;
SEQ_ASCII _iaea3d_ref :: FILE './_iaea3d_full_ref.txt' ;
PROCEDURE assertS ;
*
IAEA3D := GEO: :: CAR3D 17 17 4
          EDIT 2
          X- VOID  X+ VOID
          Y- VOID  Y+ VOID
```

```
Z- VOID  Z+ VOID
MESHX -160.0 -140.0 -120.0 -100.0 -80.0 -60.0 -40.0 -20.0
      0.0 20.0 40.0 60.0 80.0 100.0 120.0 140.0 160.0 180.0
MESHY -160.0 -140.0 -120.0 -100.0 -80.0 -60.0 -40.0 -20.0
      0.0 20.0 40.0 60.0 80.0 100.0 120.0 140.0 160.0 180.0
MESHZ 0.0 20.0 280.0 360.0 380.0
SPLITZ 1 6 2 1
MIX
    ! PLANE NB 1
    0 0 0 0 0 4 4 4  4  4 4 4 0 0 0 0 0
    0 0 0 4 4 4 4 4  4  4 4 4 4 4 0 0 0
    0 0 4 4 4 4 4 4  4  4 4 4 4 4 4 0 0
    0 4 4 4 4 4 4 4  4  4 4 4 4 4 4 4 0
    0 4 4 4 4 4 4 4  4  4 4 4 4 4 4 4 0
    4 4 4 4 4 4 4 4  4  4 4 4 4 4 4 4 4
    4 4 4 4 4 4 4 4  4  4 4 4 4 4 4 4 4
    4 4 4 4 4 4 4 4  4  4 4 4 4 4 4 4 4

    4 4 4 4 4 4 4 4  4  4 4 4 4 4 4 4 4

    4 4 4 4 4 4 4 4  4  4 4 4 4 4 4 4 4
    4 4 4 4 4 4 4 4  4  4 4 4 4 4 4 4 4
    4 4 4 4 4 4 4 4  4  4 4 4 4 4 4 4 4
    0 4 4 4 4 4 4 4  4  4 4 4 4 4 4 4 0
    0 4 4 4 4 4 4 4  4  4 4 4 4 4 4 4 0
    0 0 4 4 4 4 4 4  4  4 4 4 4 4 4 0 0
    0 0 0 4 4 4 4 4  4  4 4 4 4 4 0 0 0
    0 0 0 0 0 4 4 4  4  4 4 4 0 0 0 0 0

    ! PLANE NB 2
    0 0 0 0 0 4 4 4  4  4 4 4 0 0 0 0 0
    0 0 0 4 4 4 1 1  1  1 1 4 4 4 0 0 0
    0 0 4 4 1 1 1 2  2  2 1 1 1 4 4 0 0
    0 4 4 1 1 2 2 2  2  2 2 2 1 1 4 4 0
    0 4 1 1 3 2 2 2  3  2 2 2 3 1 1 4 0
    4 4 1 2 2 2 2 2  2  2 2 2 2 2 1 4 4
    4 1 1 2 2 2 2 2  2  2 2 2 2 2 1 1 4
    4 1 2 2 2 2 2 2  2  2 2 2 2 2 2 1 4

    4 1 2 2 3 2 2 2  3  2 2 2 3 2 2 1 4

    4 1 2 2 2 2 2 2  2  2 2 2 2 2 2 1 4
    4 1 1 2 2 2 2 2  2  2 2 2 2 2 1 1 4
    4 4 1 2 2 2 2 2  2  2 2 2 2 2 1 4 4
    0 4 1 1 3 2 2 2  3  2 2 2 3 1 1 4 0
    0 4 4 1 1 2 2 2  2  2 2 2 1 1 4 4 0
    0 0 4 4 1 1 1 2  2  2 1 1 1 4 4 0 0
    0 0 0 4 4 4 1 1  1  1 1 4 4 4 0 0 0
    0 0 0 0 0 4 4 4  4  4 4 4 0 0 0 0 0

    ! PLANE NB 3
    0 0 0 0 0 4 4 4  4  4 4 4 0 0 0 0 0
    0 0 0 4 4 4 1 1  1  1 1 4 4 4 0 0 0
    0 0 4 4 1 1 1 2  2  2 1 1 1 4 4 0 0
    0 4 4 1 1 2 2 2  2  2 2 2 1 1 4 4 0
```

```
            0 4 1 1 3 2 2 2  3  2 2 2 3 1 1 4 0
            4 4 1 2 2 2 2 2  2  2 2 2 2 2 1 4 4
            4 1 1 2 2 2 3 2  2  2 3 2 2 2 1 1 4
            4 1 2 2 2 2 2 2  2  2 2 2 2 2 2 1 4

            4 1 2 2 3 2 2 2  3  2 2 2 3 2 2 1 4

            4 1 2 2 2 2 2 2  2  2 2 2 2 2 2 1 4
            4 1 1 2 2 2 3 2  2  2 3 2 2 2 1 1 4
            4 4 1 2 2 2 2 2  2  2 2 2 2 2 1 4 4
            0 4 1 1 3 2 2 2  3  2 2 2 3 1 1 4 0
            0 4 4 1 1 2 2 2  2  2 2 2 1 1 4 4 0
            0 0 4 4 1 1 1 2  2  2 1 1 1 4 4 0 0
            0 0 0 4 4 4 1 1  1  1 1 4 4 4 0 0 0
            0 0 0 0 0 4 4 4  4  4 4 4 0 0 0 0 0

            ! PLANE NB 4
            0 0 0 0 0 4 4 4  4  4 4 4 0 0 0 0 0
            0 0 0 4 4 4 4 4  4  4 4 4 4 4 0 0 0
            0 0 4 4 4 4 4 4  4  4 4 4 4 4 4 0 0
            0 4 4 4 4 4 4 4  4  4 4 4 4 4 4 4 0
            0 4 4 4 5 4 4 4  5  4 4 4 5 4 4 4 0
            4 4 4 4 4 4 4 4  4  4 4 4 4 4 4 4 4
            4 4 4 4 4 4 5 4  4  4 5 4 4 4 4 4 4
            4 4 4 4 4 4 4 4  4  4 4 4 4 4 4 4 4

            4 4 4 4 5 4 4 4  5  4 4 4 5 4 4 4 4

            4 4 4 4 4 4 4 4  4  4 4 4 4 4 4 4 4
            4 4 4 4 4 4 5 4  4  4 5 4 4 4 4 4 4
            4 4 4 4 4 4 4 4  4  4 4 4 4 4 4 4 4
            0 4 4 4 5 4 4 4  5  4 4 4 5 4 4 4 0
            0 4 4 4 4 4 4 4  4  4 4 4 4 4 4 4 0
            0 0 4 4 4 4 4 4  4  4 4 4 4 4 4 0 0
            0 0 0 4 4 4 4 4  4  4 4 4 4 4 0 0 0
            0 0 0 0 0 4 4 4  4  4 4 4 0 0 0 0 0
          ;
TRACK := NSST: IAEA3D :: EDIT 2 MAXR 3000 HYPE 2 ;
************************************************************************
*  MACROLIB DEFINITION                                                 *
************************************************************************
MACRO := MAC: ::
 EDIT 2 NGRO 2 NMIX 5 NIFI 1
 READ INPUT
 MIX     1
      DIFF  1.500E+00  4.0000E-01
     TOTAL  3.000E-02  8.0000E-02
    NUSIGF  0.000E+00  1.3500E-01
       CHI  1.0        0.0
  H-FACTOR  0.000E+00  1.3500E-01
      SCAT  1 1 0.0 2 2 0.0 0.2E-01
 MIX     2
      DIFF  1.500E+00  4.0000E-01
     TOTAL  3.000E-02  8.5000E-02
    NUSIGF  0.000E+00  1.3500E-01
```

```
       CHI  1.0        0.0
  H-FACTOR  0.000E+00  1.3500E-01
      SCAT  1 1 0.0 2 2 0.0 0.2E-01
 MIX     3
      DIFF  1.500E+00  4.00000E-01
     TOTAL  3.000E-02  1.30000E-01
    NUSIGF  0.000E+00  1.35000E-01
       CHI  1.0        0.0
  H-FACTOR  0.000E+00  1.35000E-01
      SCAT  1 1 0.0 2 2 0.0 0.2E-01
 MIX     4
      DIFF  2.000E+00  3.0000E-01
     TOTAL  4.000E-02  1.0000E-02
      SCAT  1 1 0.0 2 2 0.0 0.4E-01
 MIX     5
      DIFF  2.000E+00  3.0000E-01
     TOTAL  4.000E-02  5.5000E-02
      SCAT  1 1 0.0 2 2 0.0 0.4E-01
 ;
************************************************************************
*  FLUX SOLUTION (NODAL EXPANSION METHOD)                              *
************************************************************************

FLUX := BRIF: TRACK MACRO ::
EDIT 1
NUPD  100       !max no. of nodal updates
EXTE  1.E-8 300 !max no. of iterations and tolerance for outer iterations for power method
INNE  1.E-8 300 !max no. of iterations and tolerance for Krylov iterations for power method
THER  1.E-6 300 !max no. of iterations and tolerance for thermal iterations for power method
CMFD all_groups Wielandt ON Krylov bicgstab
VOID Marshak
NODA NEM
LEAK quadratic
;
REF := _iaea3d_ref :: EDIT 1 ;
EDIT := OUT: FLUX TRACK MACRO IAEA3D ::
       EDIT 2 INTG IN ;
ERROR: REF EDIT ;
assertS FLUX :: 'K-EFFECTIVE' 1 1.029083 ;
END: ;
```

The corresponding numerical results are presented in Table 6. Statistics are given for k_{eff} and power distribution accuracies. The reference solution is a converged Raviart-Thomas solution with cubic polynomials, Gauss-Legendre integration and 2×2×2 mesh splitting. Reference powers P_i^* of assembly i were obtained with the following formula:

$$P_i^* = \frac{1}{V_i} \int_{V_i} d^3r \sum_h \nu\Sigma_{\text{f},h}(\boldsymbol{r})\phi_h(\boldsymbol{r}) \tag{4.1}$$

where V_i is the volume of assembly i.

These powers are then compared to data from less accurate calculations in order to obtain maximum and average errors:

$$\epsilon_{\max} = \max_i \left\{ \frac{|P_i - P_i^*|}{P_i^*} \right\} \tag{4.2}$$

and

$$\bar{\epsilon} = \frac{1}{V_{\text{core}}} \sum_i V_i \, \frac{|P_i - P_i^*|}{P_i^*}. \tag{4.3}$$

Table 6: IAEA3D benchmark calculations.[a]

Type of method	Transverse leakage	Mesh-splitting	N_{tot}	k_{eff}	Δk_{eff} (pcm)	$\epsilon_{\max}$ (%)	$\bar{\epsilon}$ (%)
Coarse mesh		No	24681	1.031891	282.1	25.6	9.4
finite differences		2×2×2	195124	1.028992	−7.8	26.4	9.6
Nodal expansion	flat	No	24681	1.029760	69.0	5.6	0.7
method	flat	2×2×2	195124	1.029258	18.8	1.6	0.3
	quadratic	No	24681	1.029083	1.3	1.2	0.4
	quadratic	2×2×2	195124	1.029085	1.5	0.2	0.1
Analytic nodal	flat	No	24681	1.029833	76.4	5.7	0.8
method	flat	2×2×2	195124	1.029442	37.3	3.8	0.5
	quadratic	No	24681	1.029131	6.2	1.2	0.3
	quadratic	2×2×2	195124	1.029111	4.2	0.5	0.1

(a) The reference effective multiplication factor is $k_{\text{eff}} = 1.029070$.

5 The Brisingr Package

The following archive is required to install Brisingr:

`Brisingr_Version5.0.8_evn.tgz`

Information is recovered from the archive using

`tar xvfz Brisingr_Version5.0.8_evn.tgz`

The `tar xvfz` operations will create a directory named `Brisingr_Version5_evn`. Another directory named `libraries` should also be created at the same level to hold cross section libraries. Directory `libraries` is generally a symbolic link to an existing location. The complete setup is depicted in Figure 11.

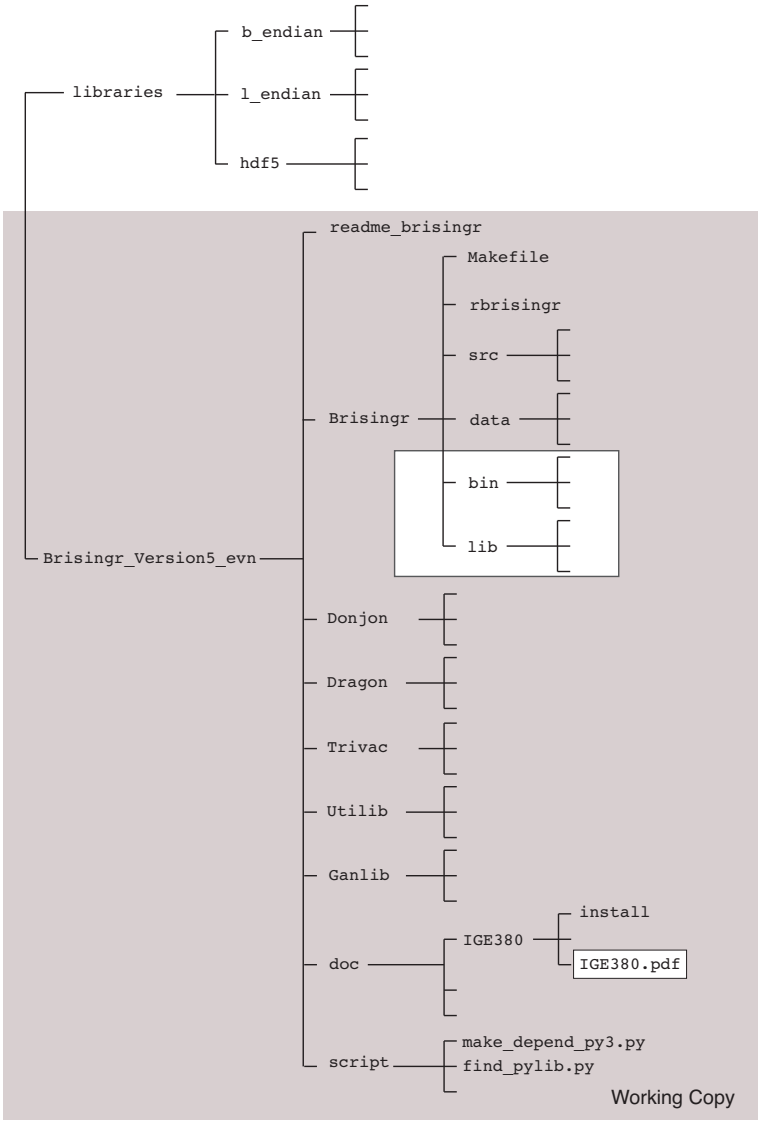

Figure 11: Distribution content.

Directory `Brisingr_Version5_evn` contains the information required to install and configure Brisingr. Inside this directory is a file named `readme_brisingr` that contain the information required to configure Brisingr on your system. This configuration process has the effect to add a few directories and binary files to the `Brisingr_Version5_evn` directory.

It is possible to perform a basic installation on a Unix-based system using makefiles. *Note:* On AIX and Solaris OS, you must replace `make` with `gmake` (the GNU variant of `make` utility). To install Brisingr, simply do

```
cd ~/Brisingr_Version5_evn/Brisingr/
make
make clean
```

To execute the Brisingr non-regression tests, do

```
make tests
```

Directory `libraries` contains cross section libraries that can be used to test your implementation. The `libraries` directory must me installed as shown in Figure 11 *before* following the instructions of the `readme_brisingr` file for executing any test requising a cross-section library. As an example, the `SmallCore_BaffRefl.access` script creates a symbolic link between the Apollob file named `CEA514_T2_V1_SHEM281_GV0.3.3_N.xsm` and local file `CEAT2` used in the `SmallCore_BaffRefl.x2m` non-regression test. The `CEA514_T2_V1_SHEM281_GV0.3.3_N.xsm` file is not included in the archive.

The content of the `readmeBrisingr` file follows:

File: `readmeBrisingr`

```
# To activate hdf5 and Python3 bindings in the make utility, you need to define
# environment variables on a UNIX system. Add the information about the HDF5_INC,
# HDF5_API and FORTRANPATH environment variables in the .profile or .bashrc script.
# These lines are OS-dependent.

# Availability of python3 utility and definition of the FORTRANPATH
# environment variable are prerequisite requirements for using both Makefiles
# and the PyGan bindings.
#
# On the recherche network at Polytechnique Montreal:
# Support for HDF5
export HDF5_INC="/usr/local/hdf5/include" # HDF5 include directory
if [ $MachineExtension = "-aix" ]
then
    export HDF5_INC="/usr/include" # HDF5 include directory
fi
export HDF5_API="$HDF5_INC/../lib" # HDF5 C API
export LD_LIBRARY_PATH="$LD_LIBRARY_PATH:$HDF5_API"
# Support for Python3 API
export FORTRANPATH="/usr/lib/gcc/x86_64-redhat-linux/4.8.5/" # contains libgfortran.so

#
# On the RedHat 8 operating system:
# Support for HDF5
export HDF5_INC="/usr/include" # HDF5 include directory
export HDF5_API="$HDF5_INC/../lib64" # HDF5 C API
export LD_LIBRARY_PATH="$LD_LIBRARY_PATH:$HDF5_API"
# Support for Python3 API
export FORTRANPATH="/usr/lib/gcc/x86_64-redhat-linux/8/" # contains libgfortran.so

#
# On the Ubuntu operating system:
# Support for HDF5
export HDF5_INC="/usr/include/hdf5/serial/" # HDF5 include directory
export HDF5_API="/usr/lib/x86_64-linux-gnu/hdf5/serial" # HDF5 C API
```

```
export LD_LIBRARY_PATH="$LD_LIBRARY_PATH:$HDF5_API"
# Support for Python3 API
export FORTRANPATH="/usr/lib/gcc/x86_64-linux/9/" # contains libgfortran.so

#
# On the Scibian 10 operating system:
# Support for HDF5
export HDF5_INC="/usr/include/hdf5/serial" # HDF5 include directory
export HDF5_API="/usr/lib/x86_64-linux-gnu/hdf5/serial" # HDF5 C API
export LD_LIBRARY_PATH="$LD_LIBRARY_PATH:$HDF5_API"
# Support for Python3 API
export FORTRANPATH="/usr/lib/gcc/x86_64-linux-gnu/8/" # contains libgfortran.so

#
# Instructions for configuring Brisingr_Version5_evn on UNIX systems
#
# To configure Brisingr_Version5_evn components with custom compiler using makefiles:
cd ~/Brisingr_Version5_evn/Brisingr/
make
make clean
#
# To configure Brisingr_Version5_evn components with Intel compiler using makefiles:
cd ~/Brisingr_Version5_evn/Brisingr/
make intel=1
make clean
#
# To build an OpenMP-enabled version, simply write
make openmp=1
#
# To execute the non-regression tests with custom compiler:
make tests
#
# To execute the non-regression tests with Intel compiler:
make tests intel=1
#
# On AIX and Solaris OS, you must use GNU Make:
cd ~/Brisingr_Version5_evn/Brisingr/
gmake
gmake clean
gmake tests

# To execute Brisingr with Intel compiler:
cd ~/Brisingr_Version5_evn/Brisingr/
./rbrisingr -c intel AFA_180_310_type1.x2m

# To execute Brisingr with custom compiler:
cd ~/Brisingr_Version5_evn/Brisingr/
./rbrisingr AFA_180_310_type1.x2m

# To configure the doc
cd ~/Brisingr_Version5_evn/doc/IGE380
./install

# To read the doc:
gv -antialias ~/Brisingr_Version5_evn/doc/IGE380/IGE380.pdf
```

References

[1] M. Gamarino, "Modal methods for rehomogenization of nodal cross sections in nuclear reactor core analysis," Ph. D. thesis, Delft University of Technology (2018).

[2] A. Hébert and R. Roy, "The GANLIB5 kernel guide (64–bit clean version)," Report IGE-332, Polytechnique Montréal (2022). See the home page at http://merlin.polymtl.ca/downloads/IGE332.pdf

[3] H. Finnemann, F. Bennewitz and M. R. Wagner "Interface current techniques for multidimensional reactor calculations," *Atomkernenergie*, Vol. 30, No. 2 pp. 123–128 (1977),

[4] K. S. Smith, "An Analytic Nodal Method for Solving the Two-Group, Multidimensional, Static and Transient Neutron Diffusion Equation," Nuclear Engineer's Thesis, Massachusetts Institute of Technology, Department of Nuclear Engineering, 1979.

[5] P. J. Turinsky, R. M. K. Al-Chalabi, P. Engrand, H. N. Sarsour, F. X. Faure and W. Guo, "NESTLE: Few-group neutron diffusion equation solver utilzing the nodal expansion method for eigenvalue, adjoint, fixed-source steady-state and transient problem," Electric Power Research Center, North Carolina State University, Raleigh, NC 27695-7909 (1994).

[6] A. Hébert, "A simplified presentation of the multigroup analytic nodal method in 2–D Cartesian geometry," *Ann. nucl. Energy*, **35**, 2142–2149 (2008).

[7] A. Hébert, *Applied Reactor Physics*, Third Edition, Presses Internationales Polytechnique, ISBN 978-2-553-01735-3, 410 p., Montréal, 2020.

[8] D. Shapero, "The Sigma GitHub website". See the home page at https://github.com/danshapero/sigma

[9] "Argonne Code Center: Benchmark Problem Book," ANL-7416, Supp. 2, ID11-A2, Argonne National Laboratory (1977).

Index

`::`, 26, 31
`:=`, 26, 31
`;`, 25

adf, 31, 32
`ADJ`, 27, 28
`ALBE`, 31, 32
`all_groups`, 27, 28
`ANM`, 27, 28

`bicgstab`, 27, 28
(BRIF:), 26
`BRIF:`, 26
(BRIF_data), 26, 27
(Brisingr), 25
`BUCK`, 27, 29

`cg`, 27, 28
`CMFD`, 27, 28
`colorset`, 27, 28

`EDIT`, 27, 31
EDIT_REF, 29, 31
EDIT_SN, 32
`END: ;`, 25
ev_nodal_freq, 27, 29
`EXTE`, 27

`flat`, 27, 28
FLUX, 26
`FREQ`, 27, 28

`general`, 27, 28
GEOM, 31
`GEOM`, 27, 28
GEOM_GAP, 29, 31, 32
`GRID`, 31, 32
group_tol, 27, 28

`HDF5_FILE`, 25
`HYPE`, 31

igmax, 31
igrid, 31, 32
`INNE`, 27
`INNER`, 27
inner_tol, 27
iprint, 27, 31

`jacobi`, 27, 28

`Krylov`, 27, 28

`ldu`, 27, 28
`LEAK`, 27, 28
`LEFT`, 31, 32
`linear`, 27, 28
`LINKED_LIST`, 25
LX, 31, 32
LY, 31, 32

MACRO, 26, 31
`Mark`, 27, 28
`Marshak`, 27, 28
max_no_group_iter, 27
max_no_inner_iter, 27
max_no_nodal_iter, 27
max_no_outer_iter, 27
MIX, 32
`MODULE`, 25

NAME1, 25
NAME2, 25
NAME3, 25
NAME4, 25
NAME5, 25
NAME6, 25, 26
`NEM`, 27, 28
`NGET`, 31, 32
`NOAL`, 31, 32
`NODA`, 27, 28
`NODF`, 27, 28
(NODSPH:), 31
`NODSPH:`, 31
(NODSPH_data), 31
NODSPH_data, 31
`NUPD`, 27

`OFF`, 27, 28
`ON`, 27, 28
`one_group`, 27
`one_groups`, 28
outer_tol, 27

`quadratic`, 27, 28

`RIGH`, 31, 32

`SELE`, 27, 28
`SEQ_ASCII`, 25
`SEQ_BINARY`, 25
(specif), 25, 26

`THER`, 27
TRACK, 26

valb2, 27, 29
`VOID`, 27, 28

`Wielandt`, 27, 28

`XSM_FILE`, 25