SKETCH-N: A NODAL NEUTRON DIFFUSION CODE FOR SOLVING STEADY-STATE AND KINETICS PROBLEMS

VERSION 1.1

VOL. I. MODEL DESCRIPTION

DRAFT

Vyacheslav G. Zimin

Experimental Scientific Research and
Development Association "Training Simulator Systems"
ENIKO TSO, Kashirskoe shosse, 31, Moscow, 115409, Russia

August 2003

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ABSTRACT

1. Program Name and Title: SKETCH-N v.1.1: A nodal code for solving neutron diffusion equations of steady-state and kinetics problems.

2. Computers for which Program is Assigned and Other Machine Versions Available: Workstations under UNIX or Windows NT.

3. Problem Solved: The SKETCH-N code solves neutron diffusion equations in x-y-z geometry for steady-state and neutron kinetics problems. The code can treat an arbitrary number of neutron energy groups and delayed neutron precursors.

4. Method of the Solution: The polynomial, semi-analytic and analytic nodal methods based on the nonlinear iteration procedure can be used for spatial discretization of diffusion equations. The time integration of the neutron kinetics problem is performed by a fully implicit scheme with an analytical treatment of the delayed neutron precursors. The steady-state eigenvalue problems are solved by inverse iterations with Wielandt shift, the Chebyshev adaptive acceleration procedure is used for the neutron kinetics problems. The block symmetric Gauss-Seidel preconditioner is applied in the both iterative methods. The flux-weighting homogenization procedure is used for partially-rodded nodes to minimize a rod cusping effect. Simple one-phase model of the thermal-hydraulics of fuel assembly is included in the code. The code also has an interface module for a coupling with transient analysis codes , such as TRAC. The interface module performs a data exchange between the codes, synchronizes a time stepping and maps the neutronics data onto thermal-hydraulics spatial mesh and vice versa. The interface module is based on the message passing library PVM (Parallel Virtual Machine).

5. Restriction on the Complexity of the Problem: The code can treat the neutron diffusion problems in Cartesian geometry. Few-group macro cross sections and their dependencies are provided by a code user. The code does not have fuel burn-up modelling capabilities. An external thermal-hydraulics code is generally required for the calculation of the "real-life" problems.

6. Typical Running Time: The running time of the case A1 of the PWR NEACRP rod ejection benchmark (2 neutron energy groups, 6 groups of the delayed neutron precursors, 221x18 neutronics nodes, 876 time steps) is 309 s on PC Pentium III 600 MHz with an internal thermal hydraulics model.

7. Unusual Features of the Program: Dimensions of a problem are specified as parameters in the include files, the code should be recompiled when the problem dimensions are changed. The code has PVM-based interface module developed for a coupling with transient thermal-hydraulics codes. The interface model has been used for a coupling of the SKETCH-N code with the J-TRAC (TRAC-PF1) and TRAC-BF1 codes.

8. Related and Auxiliary Programs: PVM library is used for the interface module of the code.

9. Status: The SKETCH-N code has been verified by solving the steady-state and neutron kinetics benchmark problems. The coupled J-TRAC/SKETCH-N code system has been verified against NEACRP PWR rod ejection and rod withdrawal benchmarks. NEACRP BWR cold water injection benchmark has been used for verification of the TRAC-BF1/SKETCH-N system.

10. References:

Zimin V. G. "Nodal Neutron Kinetics Models Based on Nonlinear Iteration Procedure for LWR Analysis", PhD Thesis, Research Laboratory for Nuclear Reactors, Tokyo Institute of Technology, August 1997.

Zimin, V.G., and H. Ninokata, "Nodal Neutron Kinetics Model Based on Nonlinear Iteration Procedure for LWR Analysis," Ann. Nucl. Energy, 25, 507-528, 1998

Zimin, V.G., H. Ninokata, and L. R. Pogosbekyan "Polynomial and Semi-Analytic Nodal Methods for Nonlinear Iteration Procedure," *Proc. of the Int. Conf. on the Physics of Nuclear Science and Technology*, October 5-8, 1998, Long Island, New York, American Nuclear Society, vol. 2, pp. 994-1002, 1998

Zimin V. G., H. Asaka, Y. Anoda, and M. Enomoto, "Verification of the J-TRAC code with 3D Neutron Kinetics Model SKETCH-N for PWR Rod Ejection Analysis", *Proc. of the 9th International Topical Meeting on Nuclear Reactor Thermal Hydraulics (NURETH 9)*, San Francisco, California, October 3-8, CD-ROM, 1999.

Asaka, H., V. G. Zimin, T. Iguchi, and Y. Anoda, "Coupling of the Thermal-Hydraulic TRAC Codes with 3D Neutron Kinetics Code SKETCH-N", *Proc. of OECD/CSNI Workshop on Advanced Thermal-Hydraulic and Neutronics Codes: Current and Future Applications*, Barcelona, Spain, 10-13 April, 2000

Zimin, V. G., H. Asaka, Y. Anoda, E. Kaloinen and R. Kyrki-Rajamaki, "Analysis of NEACRP 3D BWR Core Transient Benchmark", *Proc. of the 4th Intl. Conf. on Supercomputing in Nuclear Application SNA 2000*, September 4-7, 2000, Tokyo, Japan.

V. G. Zimin and D. M. Baturin "Analysis of the VVER-440 AER2 Rod Ejection Benchmark by the SKETCH-N Code", *Proc. of 11th Symposium of AER on VVER Reactor Physics and Reactor Safety,* September 24 - 28, 2001, Csopak, Hungary, CD-ROM.

Zimin V. G. and D. M. Baturin "Polynomial Nodal Method for Solving Neutron Diffusion Equations in Hexagonal-Z Geometry ", *Ann. Nucl. Energy*, **29**, 2002, 1105-1117

11. Machine Requirements: A workstation under UNIX, Windows NT or Windows.

12. Program Language Used: Fortran 90

13. Operating System under which Program is Executed: UNIX, Windows NT.

14. Other Programming or Operating Information or Restrictions: The interface module requires PVM installed on a computer. The PVM is a public domain software available from NETLIB [http://www.epm.ornl.gov/pvm/pvm_home.html].

15. Name and Establishment of Authors:

Vyacheslav G. Zimin
Experimental Scientific Research and
Development Association "Training Simulator Systems"
ENIKO TSO, Kashirskoe shosse, 31, Moscow, 115409, Russia
Tel (Fax): +007-095-323-95-99
E-mail: slava@ets.mephi.ru

16. Material Available:
    Source Code
    Sample LWR Benchmark Problems Input and Output Files
    SKETCH-N Manual, vol. I. Model Description
    SKETCH-N Manual, vol. II User's Guide

17. Category: F
18. Keywords: kinetics, three-dimensional, neutron diffusion, nodal methods, nonlinear iteration procedure, reactor transient analysis.
19. Sponsor:

1. INTRODUCTION

SKETCH-N is a FORTRAN 90 computer code that solves few-group neutron diffusion equations in three-dimensional Cartesian geometry. The code can treat the steady-state eigenvalue problem and neutron kinetics problems. A criticality search of the soluble boron concentration can be performed in the case of the eigenvalue problem.

Arbitrary number of neutron energy groups and delayed neutron precursors can be used. The code can treat upscattering. Reactor model should be described in three-dimensional Cartesian geometry, arbitrary mesh size is allowed in all directions. Two and one-dimensional models can be also utilised. Reflective boundary conditions can be use to model a quarter- and half-symmetric cores. The boundary conditions include zero flux and current conditions and albedo type conditions.

Spatial discretization of the diffusion equations can be done using several nodal methods. All of them are based on the transverse integration procedure and quadratic leakage approximation. Polynomial nodal method (PNM) is based on the fourth order expansion of the transverse-integrated neutron flux into Legendre polynomials. In the semi-analytical nodal method (SANM), the third and fourth Legendre polynomials are replaced by hyperbolic sine and cosine. The sine and cosine are modified in the way to preserve an orthogonal system of the basic functions, similar to Legendre polynomials. An exact analytic solution of the transverse-integrated nodal equations is applied in the analytic nodal method (ANM). The nodal equations are solved using the nonlinear iteration procedure. In this approach, the coarse-mesh finite-difference (CMFD) method is forced to match the results of the nodal method. The nodal equations are used only to compute the nodal coupling coefficients, which are introduced into the CMFD equations. The resulting solution procedure is decoupled into two processes:

- iterative solution of the CMFD equations over the reactor domain with respect to the node-averaged neutron fluxes and eigenvalue;
- local solutions of the two-node problems for each face of the nodes to compute the face-averaged neutron currents and update the nodal coupling coefficients.

If the nodal coupling coefficients are set to zero we obtain the mesh-centered finite-difference method, which also can be used in the code. An application of the nonlinear iteration procedure results in an efficient algorithm because we need to solve relatively time-consuming nodal equations only a few times during the steady-state calculation and even a once per time step for the transient problems. An orthogonal system of the basic functions of the both PNM and SANM results in a very small and simple system of the nodal equations. In the both methods, for each two-node problem we need to solve only one $G \times G$ and one $2G \times 2G$ system of equations, where G is a number of the neutron energy groups. ANM is implemented using even more efficient approach using the matrix function theory. In the case of two group calculations, when the eigenvalues of $2x2$ buckling matrix are easy to compute, the nodal equation of ANM are reduced to two equations for the face-averaged neutron fluxes. As a result computing time of ANM can be practically the same as that of PNM in the case of two-group calculations. A code user can select the nodal method depending on the problem at hand. PNM provides maximum efficiency, while accuracy is usually better with SANM and ANM. In the case of the multigroup calculations, efficiency of ANM drastically decreases in a comparison with that of PNM and SANM. As default nodal method we recommend to use SANM because it is usually the most accurate method and it is the most stable to a choice of a mesh size. The steady-state neutron diffusion equations, nonlinear iteration procedure and the nodal methods are described in Chapter 2.

Time discretization of the neutron kinetics equations is based on the fully-implicit scheme with an analytic integration of the delayed neutron precursors. The nodal equations of the neutron kinetics problem are reduced to the nodal equations of the steady-state eigenvalue problem. An automatic time step size control based on the time-step doubling technique is used in the code. As an option the SKETCH-N code has point kinetics model. Parameters of the point kinetics equations are computed from the macro cross sections at each time step. The point kinetics model is very useful to show the effect of the change of the neutron flux shape on the transient results. Neutron kinetics equations and the time-discretization techniques are given in Chapter 3.

A solution of the CMFD equations is performed by iterative methods. Three iterative levels are used in the case of the steady-state eigenvalue problem. At the top iterative level, called nonlinear iterations, macro cross sections, nodal coupling coefficients and matrices are recomputed depending on the values of thermal-hydraulics parameters, node-average neutron fluxes and eigenvalue. At the next level, called outer iterations, the linear generalized eigenvalue problem is solved by inverse iterations with Wieland shift accelerated by Chebyshev polynomials. A solution of the linear system of equations at each outer iteration is performed by inner iterations. The block symmetric Gauss-Seidel method is used at the inner iteration level, with a size of the block equal to a number of neutron energy groups. The iterative solution techniques applied in the steady-state problems are described in Chapter 4.

In the case of the neutron kinetics problem, the CMFD equations at each time step are linearized taking the values of thermal-hydraulics parameters and nodal coupling coefficients from the previous time step. The resulting linear system of equations is solved by the Chebyshev semi-iterative method. The Chebyshev iterative parameters are computed using the adaptive procedure based on a comparison of the theoretical and actual convergence rate. The block symmetric Gauss-Seidel method is used as a preconditioner. The iterative methods used in the neutron kinetics problems are discussed in Chapter 5.

Based on the value of the thermal-hydraulics parameters, the SKETCH-N code updates the values of the macro cross sections for each reactor node. Three macro cross section models are available in the code. The first one is developed for the calculations of the NEACRP LWR core transient benchmark (Finneman and Galati, 1992) and based on the polynomial representation of the macro cross section as functions of boron concentration, Doppler fuel temperature, temperature and density of coolant. The reactor is defined at the beginner of cycle 1, burnup dependencies and Xenon poisoning are not modelled. The second more general form of the macro cross sections is developed based on the data provided in NEA/NSC Ringhals-1 Boiling Water Reactor (BWR) stability benchmark (Lefvert, 1994). In this problem, the cross section file contains

- macro cross sections given as three-dimensional tables with entries for burnup, void and conversion history (conversion history stands as a collective label for void history and control rod history);
- data for the Doppler feedback, which are given as a change of the infinite multiplication factor depending on square root of the Doppler fuel temperature, void fraction and burnup;
- microscopic xenon thermal absorption cross section is specified as a polynomial function of the void and burnup.

A basic set of the cross sections is computed in the code by a tensor-product linear interpolation in the given three-dimensional table. The Doppler effect is incorporated into the fast-group absorption cross section estimated from the condition to preserve the change of the infinite multiplication factor. Xenon poisoning is simulated using the given form of the micro absorption cross section. The both macro cross section models are developed for the given macro cross section data and they are not general enough to cover all Light Water Reactor (LWR) transients.

Thus, the third macro cross section model is recently developed for the "real-life" LWR problems. A few-group macro cross section is represented as a function of burnup and thermal-hydraulics parameters. The cross section is written as a sum of two terms: the base cross section, which depends only on burnup and computed under nominal fuel assembly conditions and the deviation, which depends on burnup and thermal-hydraulics variables of the fuel assembly. The one-dimensional dependence of the base cross section is interpolated by a cubic spline. The multi-dimensional dependence of the deviation is approximated by a multi-dimensional polynomial. Generation of the approximating polynomial is performed by an external module using the best-fitting selection of the polynomial terms by a stepwise regression algorithm. A number of terms is minimal to satisfy the user given accuracy of the approximation. For example, the five-dimensional polynomial approximating two-group macro cross sections of VVER fuel assembly contains about 20 terms for the fast group cross sections and about 50 terms for thermal group cross sections. The accuracy of this polynomial approximation is 0.05 %.

The other problem related to the calculation of the macro cross sections is homogenisation of the cross sections for the nodes with a partially inserted control rod. The problem is known as a rod cusping effect. The choice of the homogenisation method has strong impact on accuracy of the rod ejection and rod withdrawal transients in Pressurized Water Reactor (PWR). In the SKETCH-N code, flux-weighting homogenisation procedure is used for partially rodded nodes. The calculation of the macro cross sections is discussed in Chapter 6.

An internal thermal-hydraulics model can treat single-phase coolant flow under a constant pressure condition and developed mostly for the code benchmarking. Heat conduction equations are discretized using the finite-difference method. Only radial heat conduction is considered. Arbitrary mesh size is allowed. The fuel material properties are defined based on the NEACRP PWR rod ejection benchmark specification (Finneman and Galati, 1992). The mass continuity and energy conservation equations for coolant are solved using the finite-difference method. A code user should define the mass flow rate for each channel used in the calculations. The thermal-hydraulics module of the code is presented in Chapter 7.

For analysis of the reactor transients with two-phase coolant flow SKETCH-N code should be coupled with an external thermal-hydraulics model. To simplify the code coupling procedure the SKETCH-N code has an interface module based on the message-passing library PVM. The interface module has been used for the coupling of the SKETCH-N code with the transient analysis codes J-TRAC (TRAC-PF1) for PWR applications and TRAC-BF1 for BWR analysis. The codes are treated as separate processes and they communicate sending and receiving messages. The interface module is responsible for the data exchange between the codes, mapping of the data on the code spatial meshes and synchronization of the time stepping. SKETCH-N sends the messages with computed power distribution and time step size and receives the messages with feedbacks and proposal for a new time step size. Feedbacks are Doppler fuel temperature, coolant temperature and coolant density or void. Data mapping technique is based on the mapping matrix approach providing flexibility for a code user in the selection of a number of fuel bundles per a thermal-hydraulics channel and axial mesh size. Time stepping of the coupled code system is based on the leap-frog semi-implicit scheme, SKETCH-N code performs the time step calculation first. Time step size of the code system is equal to a minimum of the time step sizes proposed by the codes. Our experience have shown that a coupling using the interface module can be done easy even with large and complicated codes, such as TRAC. The interface module is described in the Chapter 8.

Chapter 9 concludes the report presenting a summary of the code features and references to the results of the code verification.

2. NODAL EQUATIONS FOR STEADY-STATE PROBLEMS

Equation Section 2

2.1 Neutron Diffusion Equations

In Cartesian geometry, which will be only considered thereafter, few-group, steady-state neutron diffusion equations in P_1 form are written as

$$\sum_{u=x,y,z} \frac{\partial}{\partial u} j_{gx}(x,y,z) + \Sigma_{rg}(x,y,z)\,\varphi_g(x,y,z) = \sum_{\substack{g'=1 \\ g'\neq g}}^{G} \Sigma_{sgg'}(x,y,z)\,\varphi_{g'}(x,y,z)$$

$$+\frac{1}{k_{eff}}\chi_g \sum_{g'=1}^{G} \nu\Sigma_{fg'}(x,y,z)\,\varphi_{g'}(x,y,z), \qquad (2.1)$$

$$j_{gu}(x,y,z) = -D_g(x,y,z)\frac{\partial}{\partial u}\varphi_g(x,y,z), \quad u = x,y,z; \quad g = 1,\ldots,G;, \qquad (2.2)$$

where

g - neutron energy group index;

- total number of neutron energy groups;

$j_{gu}(x,y,z)$ - u – component of the net neutron current $[cm^{-2}s^{-1}]$;

$\varphi_g(r)$ - neutron flux $[cm^{-2}s^{-1}]$;

$\Sigma_{rg}(r)$ - macroscopic removal cross section $[cm^{-1}]$;

$\Sigma_{sgg'}(r)$ - macroscopic scattering cross section from group g' to g $[cm^{-1}]$;

k_{eff} - reactor multiplication factor;

χ_g - fission spectrum;

$\nu\Sigma_{fg'}(r)$ - number of neutrons emitted per fission times macroscopic fission cross section $[cm^{-1}]$;

$D_g(r)$ - diffusion coefficient $[cm]$.

In the steady-state reactor calculations, the system of equations (2.1)-(2.2) is a generalized eigenvalue problem and we are looking for the first eigenvalue k and the first eigenvector $\varphi_g(x,y,z)$, $g = 1,\ldots,G$. It is convenient to convert the system (2.1)-(2.2) into the matrix form as

$$\sum_{u=x,y,z} \frac{\partial}{\partial u}\mathbf{j}_u(x,y,z) + \Sigma_r(x,y,z)\,\varphi(x,y,z) = \Sigma_s(x,y,z)\,\varphi(x,y,z) +$$

$$\frac{1}{k_{eff}}\chi\,\nu\Sigma_f^T(x,y,z)\,\varphi(x,y,z) \qquad (2.3)$$

$$\mathbf{j}_u(x,y,z) = -\mathbf{D}(x,y,z)\frac{\partial}{\partial u}\varphi(x,y,z),\quad u = x,y,z; \tag{2.4}$$

where

$\mathbf{j}_u(x,y,z) = col\{j_{1u}(x,y,z),\ldots,j_{Gu}(x,y,z)\},\quad u = x,y,z;$

$\varphi(x,y,z) = col\{\varphi_1(x,y,z),\ldots,\varphi_G(x,y,z)\};$

$\Sigma_r(x,y,z) = diag\{\Sigma_{r1}(x,y,z),\ldots,\Sigma_{rG}(x,y,z)\};$

$\chi = col\{\chi_1,\ldots,\chi_G\};$

$\nu\Sigma_f(x,y,z) = col\{\nu\Sigma_{f1}(x,y,z),\ldots,\nu\Sigma_{fG}(x,y,z)\};$

$\mathbf{D}(x,y,z) = diag\{D_1(x,y,z),\ldots,D_G(x,y,z)\};$

$$\Sigma_s(r) = \begin{bmatrix} 0 & \Sigma_{s12}(x,y,z) & \ldots & \Sigma_{s1G}(x,y,z) \\ \Sigma_{s21}(x,y,z) & 0 & \ldots & \Sigma_{s2G}(x,y,z) \\ \vdots & \vdots & \vdots & \vdots \\ \Sigma_{sG1}(x,y,z) & \Sigma_{sG1}(x,y,z) & \ldots & 0 \end{bmatrix},$$

index T means a transpose operation.

The system of equations (2.3)-(2.4) should be completed by boundary conditions. We assume that the reactor domain is enclosed by the boundary normal to either x, y or z directions as illustrated in Fig. 2.1 for two-dimensional case.

The diffusion equations are completed by the suitable boundary conditions as

$$f_g\Phi_g^k(\mathbf{x})\Big|_{x\in\partial V} \pm c_g\mathbf{D}_g^k\frac{\mathrm{d}}{\mathrm{dx}}\Phi_g^k(\mathbf{x})\Big|_{x\in\partial V} = 0, \tag{2.5}$$

where $\pm$ stands for the right (+) and the left (-) external interface; $c_g + f_g > 0$, $c_g \geq 0$, $f_g \geq 0$.

Using parameters c_g and f_g, the various boundary conditions can be obtained: zero flux at the reactor interface if $c_g = 0$ and $f_g = 1$; zero current when $c_g = 1$ and $f_g = 0$..., etc.We use the following form of the boundary conditions.

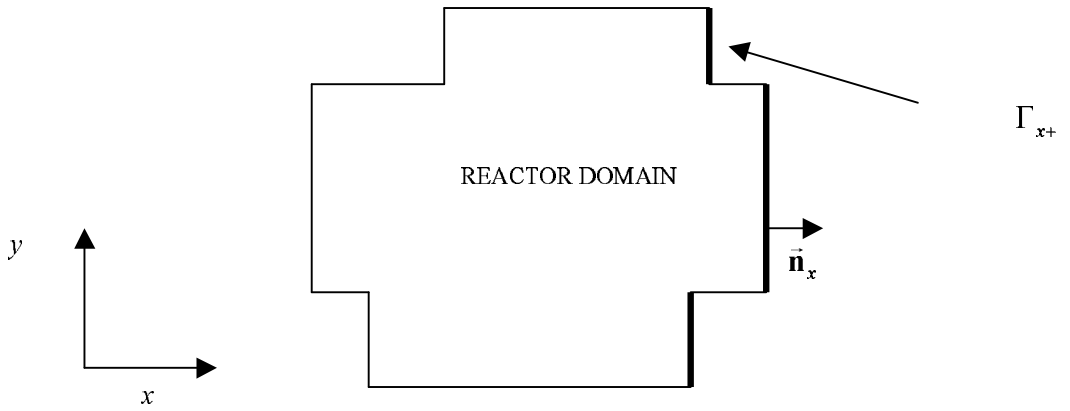

Figure 2.1. Example of the reactor domain and boundary in two-dimensional case.

2.2 Mesh-Centered Finite-Difference Method

To solve the neutron diffusion equations on computer, the continuous form (2.3)-(2.4) should be converted into a discrete representation. The reactor domain is subdivided into parallelepipeds called nodes with the grid indices x_l, y_m and z_n. A node (l,m,n) is defined as $x \in [x_l, x_{l+1}]$, $y \in [y_m, y_{m+1}]$, $z \in [z_n, z_{n+1}]$. It is convenient to move an origin of the local coordinates into the center of the node and define this node as a node k. The node widths are

$$\Delta x_k = x_{l+1} - x_l, \quad \Delta y_k = y_{m+1} - y_m, \quad \Delta z_k = z_{n+1} - z_n$$

and, in the local coordinates, the k node is

$$(x,y,z): \quad x \in [-\Delta x_k/2, \Delta x_k/2],\ y \in [-\Delta y_k/2, \Delta y_k/2],\ z \in [-\Delta z_k/2, \Delta z_k/2]$$

A node volume is defined as a product of the mesh spacing

$$V^k \equiv \Delta x_k\ \Delta y_k\ \Delta z_k .$$

In the following development, we omit a node index on the mesh spacing. We use $x+$ and $x-$ indices to denote the plus-x-directed (right) face of the node and the minus-x-directed (left) face of the node. The similar notations are applied for y- and z-directed faces. Material properties are assumed constant over the node and continuity of the flux and normal component of the current are imposed at internal faces of the node.
Integrating Eq. (2.3) over the node volume we obtain the nodal balance equation

$$\sum_{u=x,y,z} F_u^k [\mathbf{J}_{u+}^k - \mathbf{J}_{u-}^k] + \Sigma_r^k\ \bar{\Phi}^k V^k = \Sigma_s^k\ \bar{\Phi}^k V^k + \frac{1}{k_{eff}} \chi\, (\nu\Sigma_f^k)^T\ \bar{\Phi}^k V^k , \tag{2.6}$$

where

$\bar{\Phi}^k \equiv \dfrac{1}{V^k} \int_{-\Delta z/2}^{\Delta z/2} dz \int_{-\Delta y/2}^{\Delta y/2} dy \int_{-\Delta x/2}^{\Delta x/2} dx\, \varphi(x,y,z)$ - node-averaged neutron flux;

$\mathbf{J}_{x\pm}^k$ are the x-component of the net current averaged over the x-directed face of the node as

$$\mathbf{J}_{x\pm}^k \equiv \frac{1}{F_x^k} \int_{-\Delta z/2}^{\Delta z/2} dz \int_{-\Delta y/2}^{\Delta y/2} dy\, \mathbf{j}_x(x,y,z)\Big|_{x \to \pm \Delta x/2}$$

$F_x^k \equiv \Delta y\ \Delta z$ is the area of the x-directed face of the node.

Integrating the Fick's law (2.4) over the x-directed face of the node we get

$$\mathbf{J}_{x\pm}^k = -\mathbf{D}^k \frac{d}{dx} \Phi_x^k(x)\Big|_{x \to \pm \Delta x/2}, \tag{2.7}$$

where

$\Phi_x^k(x) \equiv \dfrac{1}{F_x^k} \int_{-\Delta z/2}^{\Delta z/2} dz \int_{-\Delta y/2}^{\Delta y/2} dy\, \varphi(x,y,z)$ is the face-averaged (transverse-integrated) neutron flux.

Neutron balance equations (2.6)-(2.7) are formally exact, but to solve them we have to add the relations between the face-averaged and the node-averaged fluxes. Let us consider, for example, the plus-x-directed (right) face of the node $x+$. In the finite-difference method, the derivative in Eq. (2.7) is replaced by the simple finite-difference as

$$\mathbf{J}^k_{x+} = -\hat{\mathbf{D}}^k_x \left[\Phi^k_{x+} - \bar{\Phi}^k \right], \tag{2.8}$$

where $\hat{\mathbf{D}}^k_x = 2\,\mathbf{D}^k / \Delta x$ is the nondimensional diffusion coefficient.

Let us assume that the neighbouring node $(l+1, m, n)$ has the index $k+1$. Using the similar expression for the current at the left face of the node $k+1$ and continuity of the face-averaged flux and current we express the face-averaged flux in terms of the node-averaged fluxes as

$$\Phi^k_{x+} = \Phi^k_{x-} = \left[\hat{\mathbf{D}}^k_x + \hat{\mathbf{D}}^{k+1}_x \right]^{-1} \left(\hat{\mathbf{D}}^{k+1}_x \bar{\Phi}^{k+1} + \hat{\mathbf{D}}^k_x \bar{\Phi}^k \right),$$

Substituting the expression for the face-averaged flux into Eq. (2.8) we express the face-averaged current as

$$\mathbf{J}^k_{x+} = \mathbf{J}^{k+1}_{x-} = -\hat{\mathbf{D}}^{k+1}_x \left[\hat{\mathbf{D}}^{k+1}_x + \hat{\mathbf{D}}^k_x \right]^{-1} \hat{\mathbf{D}}^k_x \left(\bar{\Phi}^{k+1} - \bar{\Phi}^k \right)$$

In the case of node face on the boundary we use the boundary condition (2.5) instead of the flux and current continuity. The face-averaged neutron flux in the case of the boundary face is given as

$$\Phi^k_{x\pm} = \left[f^k_{x\pm} + \mathbf{B}^k_{x\pm} \hat{\mathbf{D}}^k_x \right]^{-1} \mathbf{B}^k_{x\pm} \hat{\mathbf{D}}^k_x \bar{\Phi}^k,$$

where $f^k_{x\pm}$ and $\mathbf{B}^k_{x\pm}$ are the constants of the boundary conditions at the $x\pm$ boundary face of the node. Then, the face-averaged current at the $x\pm$ boundary face of the node is expressed as

$$\mathbf{J}^k_{x\pm} = \pm \hat{\mathbf{D}}^k_x \left[f^k_{x\pm} + \mathbf{B}^k_{x\pm} \hat{\mathbf{D}}^k_x \right]^{-1} f^k_{x\pm} \bar{\Phi}^k.$$

Substituting the above expressions for the face-averaged current into the neutron balance equation (2.6) we obtain the mesh-centered finite-difference form of the neutron diffusion equation as

$$\sum_{u=x,y,z} F^k_u \left\{ -\tilde{\mathbf{D}}^k_u \bar{\Phi}^{k+1} + [\tilde{\mathbf{D}}^k_u + \tilde{\mathbf{D}}^{k-1}_u] \bar{\Phi}^k - \tilde{\mathbf{D}}^{k-1}_u \bar{\Phi}^{k-1} \right\} + \Sigma^k_r \bar{\Phi}^k V^k =$$

$$\Sigma^k_s \bar{\Phi}^k V^k + \frac{1}{k_{eff}} \chi (\nu\Sigma^k_f)^T \bar{\Phi}^k V^k \tag{2.9}$$

where

$k+1$ ($k-1$) is the index of the neighboring node in the positive (negative) u – direction;

$\tilde{\mathbf{D}}^k_u = \hat{\mathbf{D}}^{k+1}_u \left[\hat{\mathbf{D}}^{k+1}_u + \hat{\mathbf{D}}^k_u \right]^{-1} \hat{\mathbf{D}}^k_u$ is finite-difference coupling coefficient in the case of internal interface;

$\tilde{\mathbf{D}}^k_u = \hat{\mathbf{D}}^k_u \left[f^k_{x\pm} + \mathbf{B}^k_{x\pm} \hat{\mathbf{D}}^k_x \right]^{-1} f^k_{x\pm}$ is finite-difference coupling coefficient in the case of the interface at the boundary.

The finite-difference Eq. (2.9) is ready for solving on a computer. There is however a problem of accuracy. Accurate finite-difference calculation requires that the mesh spacing should be an order of diffusion length, which is on the order of a centimetre in the case of light water reactor. This requirement results in hundreds of thousands unknowns making the calculations too time-consuming. Next sections will show a way to improve an accuracy of the finite-difference method, without significant modifications of the Eq. (2.9).

2.3 Nonlinear Iteration Procedure

Kord Smith proposed a simple way to improve accuracy of the finite-difference method without significant modifications (Smith, 1984). In his approach, called "nonlinear iteration procedure", the coarse-mesh finite-difference method is forced to match the results of the other discretization scheme, for example a nodal method. The approach lacking a theoretical basis but proved to be very efficient. It is used in several neutron kinetics codes with different nodal methods: in the NESTLE (Al-Chalabi, *et al.*, 1993), SPANDEX (Aviles, 1993) and PARCS (Joo, *et al.*, 1996) codes with nodal expansion method (NEM) and in the SIMULATE-3K (Borkowski *et al.*, 1996) code with the semi-analytical nodal method. There are several variants different in the way of correction of the finite-difference equations. We adopted an approach used in the NESTLE code (Turinsky, 1994). The surface-averaged neutron current is expressed as

$$\mathbf{J}_{u+}^{k} = -\tilde{\mathbf{D}}_{u}^{k}\left(\bar{\Phi}^{k+1} - \bar{\Phi}^{k}\right) - \tilde{\mathbf{D}}_{u}^{k}(nod)\left(\bar{\Phi}^{k+1} + \bar{\Phi}^{k}\right), \tag{2.10}$$

where

k 1 is the index of the neighboring node in the positive $-$ direction;

$\tilde{\mathbf{D}}_{u}^{k}$ is matrix of the finite-difference coupling coefficients derived in the previous section;

$\tilde{\mathbf{D}}_{u}^{k}(nod)$ is a matrix of the nodal coupling coefficients, which will be defined below.

Substituting this expression for the neutron flux into the neutron balance equation (2.6) we obtain a new coarse-mesh finite-difference form of the neutron diffusion equations as

$$\sum_{u=x,y,z} F_{u}^{k}\left\{-[\tilde{\mathbf{D}}_{u}^{k} + \tilde{\mathbf{D}}_{u}^{k}(nod)]\bar{\Phi}^{k+1} + [\tilde{\mathbf{D}}_{u}^{k} - \tilde{\mathbf{D}}_{u}^{k}(nod) + \tilde{\mathbf{D}}_{u}^{k-1} + \tilde{\mathbf{D}}_{u}^{k-1}(nod)]\bar{\Phi}^{k} -\right.$$

$$\left. -[\tilde{\mathbf{D}}_{u}^{k-1} - \tilde{\mathbf{D}}_{u}^{k-1}(nod)]\bar{\Phi}^{k-1} + \Sigma_{r}^{k}\,\bar{\Phi}^{k}V^{k} = \Sigma_{s}^{k}\,\bar{\Phi}^{k}V^{k} + \frac{1}{k_{eff}}\chi(\nu\Sigma_{f}^{k})^{T}\bar{\Phi}^{k}V^{k} \right. \tag{2.11}$$

Eq. (2.11) is called the coarse-mesh finite-difference (CMFD) equation to distinguish it from the standard form of the finite-difference equation (2.9). The nonlinear iterations start with the nodal coupling coefficients $\tilde{\mathbf{D}}_{u}^{k}(nod)$ set up to 0. After several iterations solving CMFD equations (2.11) we get the new approximation for the node-averaged neutron fluxes and the eigenvalue. Using the equations of the nodal method we compute the values of the surface-average neutron current. In the CMFD method, the surface-averaged neutron current is determined by Eq. (2.10). Requiring that the CMFD method reproduce the same values of the surface-averaged neutron current we compute the nodal coupling coefficients as

$$\left(\tilde{D}_{u}^{k}(nod)\right)_{gg} = \frac{-J_{gu+}^{k} - \tilde{D}_{gu}^{k}\left(\bar{\Phi}_{g}^{k+1} - \bar{\Phi}_{g}^{k}\right)}{\bar{\Phi}_{g}^{k+1} + \bar{\Phi}_{g}^{k}}; \quad \left(\tilde{D}_{u}^{k}(nod)\right)_{gg'} = 0 \; if \; g' \neq g \tag{2.12}$$

The iterations are performed till convergence, when the nodal coupling coefficients do not change anymore. The total solution procedure is decoupled into two processes: iterative solution of the CMFD equations (2.11), where we compute the node-averaged neutron fluxes and eigenvalue; and solution of the nodal equations to compute the face-averaged neutron currents and nodal coupling coefficients by Eq. (2.12).

Derivation of the nodal equations used to compute the face-averaged neutron current is given in the following sections.

2.4 Transverse-Integration Procedure

Nodal methods used in the SKETCH code are based on the transverse-integration procedure proposed by H. Finnemann (1975). In this approach, three-dimensional neutron diffusion equations (2.3)-(2.4) are integrated in two directions transverse to the direction of the interest. In the case of x direction, the transverse integration operator is defined as

$$\frac{1}{F_x^k}\int_{-\Delta y/2}^{\Delta y/2} dy \int_{-\Delta z/2}^{\Delta z/2} dz\,.$$

As a result we have

$$\frac{d}{dx}\mathbf{J}_x^k(x)+\Sigma_r^k\,\mathbf{\Phi}_x^k(x)=\Sigma_s^x\,\mathbf{\Phi}_x^k(x)+\frac{1}{k_{eff}}\boldsymbol{\chi}\,(\boldsymbol{\nu}\Sigma_f^k)^T\,\mathbf{\Phi}_x^k(x)-\mathbf{S}_x^k(x) \qquad (2.13)$$

$$\mathbf{J}_x^k(x)=-\mathbf{D}^k\,\frac{d}{dx}\mathbf{\Phi}_x^k(x),;\quad x\in[x_l,x_{l+1}] \qquad (2.14)$$

where

$\mathbf{\Phi}_x^k(x)\equiv\dfrac{1}{F_x^k}\displaystyle\int_{-\Delta y/2}^{\Delta y/2} dy \int_{-\Delta z/2}^{\Delta z/2} dz\,\varphi(x,y,z)$ is the transverse-integrated neutron flux;

$\mathbf{J}_x^k(x)\equiv\dfrac{1}{F_x^k}\displaystyle\int_{-\Delta y/2}^{\Delta y/2} dy \int_{-\Delta z/2}^{\Delta z/2} dz\,\mathbf{j}_x(x,y,z)$ is the transverse-integrated neutron current;

$\mathbf{S}_x^k(x)\equiv\dfrac{1}{\Delta y}\mathbf{L}_{xy}^k(x)+\dfrac{1}{\Delta z}\mathbf{L}_{xz}^k(x)$ is the leakage transverse to x-direction;

$$\mathbf{L}_{xy}^k(x)\equiv\frac{1}{\Delta z}\int_{-\Delta z/2}^{\Delta z/2} dz\,[\mathbf{j}_y(x,\Delta y/2,z)-\mathbf{j}_y(x,-\Delta y/2,z)];$$

$$\mathbf{L}_{xz}^k(x)\equiv\frac{1}{\Delta y}\int_{-\Delta y/2}^{\Delta y/2} dy\,[\mathbf{j}_z(x,y,\Delta z/2)-\mathbf{j}_z(x,y,-\Delta z/2)].$$

Applying the same procedure for the y and z directions we get a system of three quasi-one-dimensional equations coupled by their right hand sides through the transverse leakages.

Before proceeding further it is convenient to map the interval $[-\Delta x/2,\Delta x/2]$ onto $[-1,+1]$ and rewrite equations (2.13)-(2.14) as

$$-\frac{d^2}{d\xi^2}\Phi_x^k(\xi)+\left(\mathbf{B}^2\right)_x^k\,\Phi_x^k(\xi)=-\hat{\mathbf{S}}_x^k(\xi), \qquad (2.15)$$

$$\mathbf{J}_x^k(\xi) = -\hat{\mathbf{D}}_x^k \frac{d}{d\xi}\Phi_x^k(\xi), \quad \xi \in [-1,1], \tag{2.16}$$

where

$\xi = \dfrac{2x}{\Delta x_k}$;

$\left(\mathbf{B}^2\right)_x^k = \dfrac{(\Delta x)^2}{4}\left[\mathbf{D}^k\right]^{-1}\left\{ \Sigma_r^k - \Sigma_s^k - \dfrac{1}{k_{eff}}\chi(\nu\Sigma_f^k)^T \right\}$ nondimensional buckling matrix ;

$\hat{\mathbf{S}}_x^k(\xi) = \dfrac{(\Delta x)^2}{4}\left[\mathbf{D}^k\right]^{-1}\mathbf{S}_x^k(x)$.

Assuming that the transverse leakage $\hat{\mathbf{S}}_x^k(\xi)$ is known, Eq. (2.15)-(2.16) can be easy solved by various ways.

2.5 Transverse Leakage Approximations

To solve the transverse-integrated equations (2.15)-(2.16) we need additional information on the ransverse-leakage. The first approximation can be derived assuming that the neutron flux inside the node is separable as

$$\varphi^k(x,y,z) = \Phi_x^k(x) + \Phi_y^k(y) + \Phi_z^k(z) - 2\bar{\Phi}^k$$

Than, using the definition of the transverse leakage we have that the transverse leakage is constant inside the node

$$\mathbf{S}_x^k(x) = \bar{\mathbf{S}}_x^k = \frac{1}{\Delta y}\bar{\mathbf{L}}_{xy}^k + \frac{1}{\Delta z}\bar{\mathbf{L}}_{xz}^k = \frac{1}{\Delta y}\left[\mathbf{J}_{y+}^k - \mathbf{J}_{y-}^k\right] + \frac{1}{\Delta z}\left[\mathbf{J}_{z+}^k - \mathbf{J}_{z-}^k\right]$$

This approach is called a *flat leakage approximation* (Shober, 1977). The flat leakage approximation leads to relatively large errors in the case of realistic LWR problems and an improved variant, called *quadratic leakage approximation* (Finneman, 1975) has been proposed. The transverse leakage inside the node is approximated by a quadratic polynomial as

$$\mathbf{S}_x^k(x) = \bar{\mathbf{S}}_x^k + \sum_{i=1}^{2}\mathbf{s}_{xi}^k P_i(2x/\Delta x),$$

where $P_i(\xi)$ are some polynomials, which in our case, are Legendre polynomials, i.e. $P_1(\xi) = \xi$ and

$P_2(\xi) = \dfrac{1}{2}(3\xi^2 - 1)$, $\xi = 2x/\Delta x$.

There are several ways to compute the 1st and 2nd expansion coefficients (Lawrence, 1986). We use the traditional approach extending the interpolating polynomial over the node k and its two neighbouring nodes $k-1$ and $k+1$. Then we require that the polynomial reproduces the same average transverse leakages also for the nodes k 1 and $k+1$. Thus, we have two additional equations to compute the 1st and 2nd expansion coefficients. The explicit expressions are given as

$$\mathbf{s}_{x1}^k = \left[g_x^k\right]^{-1}\left[\left(\bar{\mathbf{S}}_x^{k+1} - \bar{\mathbf{S}}_x^k\right)\left(1+\delta_x^{k-1}\right)\left(1+2\delta_x^{k-1}\right) + \left(\bar{\mathbf{S}}_x^k - \bar{\mathbf{S}}_x^{k-1}\right)\left(1+\delta_x^{k+1}\right)\left(1+2\delta_x^{k+1}\right)\right];$$

$$\mathbf{s}_{x2}^{k} = \left[g_x^k \right]^{-1} \left[\left(\bar{\mathbf{S}}_x^{k+1} - \bar{\mathbf{S}}_x^{k} \right) \left(1 + \delta_x^{k-1} \right) + \left(\bar{\mathbf{S}}_x^{k-1} - \bar{\mathbf{S}}_x^{k} \right) \left(1 + \delta_x^{k+1} \right) \right];$$

$$g_x^k = 2\left(1+\delta_x^{k-1}\right)\left(1+\delta_x^{k+1}\right)\left(1+\delta_x^{k+1}+\delta_x^{k-1}\right);\ \ \delta_x^{k-1} = h_x^{k-1}/h_x^k;\ \ \delta_x^{k+1} = h_x^{k+1}/h_x^k.$$

In the case, if the node does not have a neighbour, we use the boundary condition (2.5) instead of the requirement to reproduce the average over the node transverse leakage.

2.6 Polynomial Nodal Method (PNM)

In this section, we give the polynomial nodal method (PNM) to solve the Eq. (2.15)-(2.16). The transverse-integrated neutron flux is expanded into Legendre polynomials up to the 4th order as

$$\Phi_x^k(\xi) = \bar{\Phi}^k + \sum_{i=1}^{4} \mathbf{a}_{xi}^k P_i\left(\xi\right), \tag{2.17}$$

where

$\bar{\Phi}^k$ - node-averaged neutron flux;

$\mathbf{a}_{xi}^k = col\{a_{1xi}^k, \ldots, a_{Gxi}^k\}$ is a vector of the unknown flux expansion coefficients;

$P_i\left(\xi\right)$ - Legendre polynomials, which are defined as

$P_0(\xi) = 1;\ P_1(\xi) = \xi;\ P_2(\xi) = \frac{1}{2}(3\xi^2 - 1);$

$P_3(\xi) = \frac{1}{2}(5\xi^3 - 3\xi);\ P_4(\xi) = \frac{1}{8}(35\xi^4 - 30\xi^2 + 3).$

Legendre polynomials are orthogonal at the interval $[-1,1]$, i.e.

$$\int_{-1}^{+1} P_i(\xi) P_j(\xi) d\xi = 0 \quad if\ i \neq j \ \text{ and } \int_{-1}^{+1} P_i^2(\xi) d\xi = N_i = \frac{2}{2i+1}.$$

Orthogonality of the Legendre polynomials results in a simple set of the final nodal equations. In the nonlinear iteration procedure, the node-averaged neutron flux and eigenvalue are assumed known from the results of the CMFD iterations. We need to compute the face-averaged currents for each node interface. Using the flux expansion (2.17) we expressed the face-averaged current in terms of the flux expansion coefficients as

$$\mathbf{J}_{x+}^k = \mathbf{J}_x^k(+1) = -\hat{\mathbf{D}}_x^k \ \left\{ \mathbf{a}_{x1}^k + 3\ \mathbf{a}_{x2}^k + 6\ \mathbf{a}_{x3}^k + 10\ \mathbf{a}_{x4}^k \right\} \tag{2.18}$$

The unknown expansion coefficients are computed considering two-node problem for each node interface as shown in Fig. 2.2.

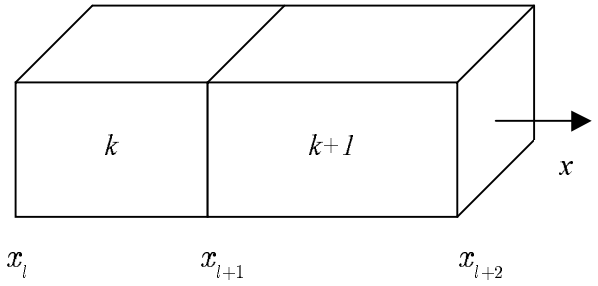

Figure 2.2. Two-node problem $(k, k+1)$.

For this two-node problem we have four unknown expansion coefficients per node per energy group or eight expansion coefficients per group per two-node problem. Thus, eight equations are required to compute them. The equations are

1. Nodal balance equations for the nodes k and $k+1$ ($2G$ equations)
2. Two moment-weighting equations for the nodes and ($4G$ equations)
3. Continuity of the face-averaged flux at an internal face of the two-node problem (G equations)
4. Continuity of the face-averaged current at an internal face of the two-node problem (G equations).

In the case of the face at the boundary, we define a one-node problem for the boundary node. In this case, we have $4G$ unknowns. The flux and current continuity equations are replaced by the boundary condition (2.5) and we have $4G$ nodal equations for $4G$ unknowns.

Substituting expansion (2.17) into Eq. (2.15) and integrating over the interval $[-1,+1]$ yields the *nodal balance equations:*

$$-\left\{ 3\,\mathbf{a}_{x2}^{k} + 10\,\mathbf{a}_{x4}^{k} \right\} + \left(\mathbf{B}^2\right)_x^k \bar{\mathbf{\Phi}}^k = -\hat{\mathbf{s}}_{x0}^{k}, \tag{2.19}$$

where

$$\hat{\mathbf{s}}_{x0}^{k} = \frac{1}{N_0}\int_{-1}^{+1} \hat{\mathbf{S}}_x^k(\xi)\,d\xi = \frac{(\Delta x)^2}{4}\left[\mathbf{D}^k\right]^{-1} \bar{\mathbf{S}}_x^k$$

Moment-weighting equations are obtained by moment-weighting with $P_1(\xi)$ and $P_2(\xi)$ yielding

1st-order moment-weighting equations (for nodes k and k+1):

$$-15\,\mathbf{a}_{x3}^{k} + \left(\mathbf{B}^2\right)_x^k \mathbf{a}_{x1}^{k} = -\hat{\mathbf{s}}_{x1}^{k}, \tag{2.20}$$

where

$$\hat{\mathbf{s}}_{x1}^{k} = \frac{1}{N_1}\int_{-1}^{+1} P_1(\xi)\,\hat{\mathbf{S}}_x^k(\xi)\,d\xi = \frac{(\Delta x)^2}{4}\left[\mathbf{D}^k\right]^{-1} \mathbf{s}_{x1}^{k}$$

2nd-order moment-weighting equations (for nodes k and k+1):

$$-35\,\mathbf{a}_{x4}^{k} + \left(\mathbf{B}^2\right)_x^k \mathbf{a}_{x2}^{k} = -\hat{\mathbf{s}}_{x2}^{k}, \tag{2.21}$$

where

$$\hat{\mathbf{s}}_{x2}^{k} = \frac{1}{N_2}\int_{-1}^{+1} P_2(\xi)\,\hat{\mathbf{S}}_x^k(\xi)\,d\xi = \frac{(\Delta x)^2}{4}\left[\mathbf{D}^k\right]^{-1} \mathbf{s}_{x2}^{k}$$

Continuity of the transverse-integrated neutron flux is written as

$$\bar{\Phi}^{k} + \mathbf{a}_{x1}^{k} + \mathbf{a}_{x2}^{k} + \mathbf{a}_{x3}^{k} + \mathbf{a}_{x4}^{k} = \bar{\Phi}^{k+1} - \mathbf{a}_{x1}^{k+1} + \mathbf{a}_{x2}^{k+1} - \mathbf{a}_{x3}^{k+1} + \mathbf{a}_{x4}^{k+1} \tag{2.22}$$

Continuity of the transverse-integrated neutron current is written as

$$-\hat{\mathbf{D}}_{x}^{k}\left\{ \mathbf{a}_{x1}^{k} + 3\,\mathbf{a}_{x2}^{k} + 6\,\mathbf{a}_{x3}^{k} + 10\,\mathbf{a}_{x4}^{k} \right\} = -\hat{\mathbf{D}}_{x}^{k+1}\left\{ \mathbf{a}_{x1}^{k+1} - 3\,\mathbf{a}_{x2}^{k+1} + 6\,\mathbf{a}_{x3}^{k+1} - 10\,\mathbf{a}_{x4}^{k+1} \right\} \tag{2.23}$$

The obtained equations are decoupled. The even flux expansion coefficients of the node k do not depend on the odd flux expansion coefficients of the node k and any expansion coefficients of the node $k+1$. Furthermore, using 1st-order moment weighting equation (2.20), the 3rd expansion coefficients are expressed in terms of the 1st expansion coefficients as

$$\mathbf{a}_{x3}^{k} = \frac{1}{15}\left\{ \left(\mathbf{B}^{2}\right)_{x}^{k} \mathbf{a}_{x1}^{k} + \hat{\mathbf{s}}_{x1}^{k} \right\} \tag{2.24}$$

Using 2nd-order moment weighting equation (2.21), the 4th expansion coefficients are expressed in terms of the 2nd expansion coefficients as

$$\mathbf{a}_{x4}^{k} = \frac{1}{35}\left\{ \left(\mathbf{B}^{2}\right)_{x}^{k} \mathbf{a}_{x2}^{k} + \hat{\mathbf{s}}_{x2}^{k} \right\} \tag{2.25}$$

Substituting Eq. (2.25) into the neutron balance equation (2.19) we obtain a system of G equations for the 2nd expansion coefficients

$$\left\{ 3\,\mathbf{I} + \frac{2}{7}\left(\mathbf{B}^{2}\right)_{x}^{k} \right\} \mathbf{a}_{x2}^{k} = \left(\mathbf{B}^{2}\right)_{x}^{k} \bar{\Phi}^{k} + \left\{ \hat{\mathbf{s}}_{x0}^{k} - \frac{2}{7}\hat{\mathbf{s}}_{x2}^{k} \right\}, \tag{2.26}$$

wher $\mathbf{I}$ is the identity matrix.

Substituting Eq. (2.24) into the neutron flux and neuron current continuity equations (2.22)-(2.23) we obtain $2G$ equations for the 1 expansion coefficients:

Flux continuity:

$$\left\{ \mathbf{I} + \frac{1}{15}\left(\mathbf{B}^{2}\right)_{x}^{k+1} \right\} \mathbf{a}_{x1}^{k+1} + \left\{ \mathbf{I} + \frac{1}{15}\left(\mathbf{B}^{2}\right)_{x}^{k} \right\} \mathbf{a}_{x1}^{k} = \left\{ \bar{\Phi}^{k+1} + \mathbf{a}_{x2}^{k+1} + \alpha_{x4}^{k+1} \right\} -$$

$$\left\{ \bar{\Phi}^{k} + \mathbf{a}_{x2}^{k} + \mathbf{a}_{x4}^{k} \right\} - \frac{1}{15}\left\{ \hat{\mathbf{s}}_{x1}^{k} + \hat{\mathbf{s}}_{x1}^{k+1} \right\} \tag{2.27}$$

Current continuity:

$$\hat{\mathbf{D}}_{x}^{k+1}\left\{ \mathbf{I} + \frac{2}{5}\left(\mathbf{B}^{2}\right)_{x}^{k+1} \right\} \mathbf{a}_{x1}^{k+1} - \hat{\mathbf{D}}_{x}^{k}\left\{ \mathbf{I} + \frac{2}{5}\left(\mathbf{B}^{2}\right)_{x}^{k} \right\} \mathbf{a}_{x1}^{k} =$$

$$\hat{\mathbf{D}}_{x}^{k+1}\left\{ 3\,\mathbf{a}_{x2}^{k+1} + 10\,\mathbf{a}_{x4}^{k+1} - \frac{2}{5}\hat{\mathbf{s}}_{x1}^{k+1} \right\} + \hat{\mathbf{D}}_{x}^{k}\left\{ 3\,\mathbf{a}_{x2}^{k} + 10\,\mathbf{a}_{x4}^{k} + \frac{2}{5}\hat{\mathbf{s}}_{x1}^{k} \right\} \tag{2.28}$$

As a result, the initial system of $8G$ nodal equations is reduced to G equations (2.26) with respect to the 2nd expansion coefficients of the node $k+1$ and $2G$ equations (2.27)-(2.28) with respect to the 1st expansion coefficients of the nodes and k. The even expansion coefficients of the node k are known from the solution of the previous two-node problem $(k-1,k)$. When the neutron flux expansion coefficients are computed, the face-averaged nodal current $\mathbf{J}^k_{x+}$ is calculated using Eq. (2.18).

It is worth to note, that Legendre polynomials $P_i(u)$ and the NEM basic functions (Finnemann, 1977) adjusted to the interval $[-1,+1]$ satisfy the following relation

$$\begin{pmatrix} P_0^{NEM}(u) \\ P_1^{NEM}(u) \\ P_2^{NEM}(u) \\ P_3^{NEM}(u) \\ P_4^{NEM}(u) \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1/2 & 0 & 0 & 0 \\ 0 & 0 & 1/2 & 0 & 0 \\ 0 & -1/20 & 0 & 1/20 & 0 \\ 0 & 0 & -1/70 & 0 & 1/70 \end{pmatrix} \times \begin{pmatrix} P_0(u) \\ P_1(u) \\ P_2(u) \\ P_3(u) \\ P_4(u) \end{pmatrix}$$

NEM basic functions span the same subspace as Legendre polynomials. If the moment weighting procedure is used in NEM, PNM and NEM become equivalent. In our opinion, the nodal equations of the PNM nodal method are simpler that that of NEM.

2.7 Semi-Analytic Nodal Method (SANM)

In the semi-analytic nodal method (SANM), the 3rd and 4th Legendre polynomials are replaced by hyperbolic sine and cosine, which are general solutions of the following homogeneous problem

$$-\frac{d^2}{d\xi^2}\Phi^k_{gx}(\xi) + \left(\alpha^k_{gx}\right)^2 \Phi^k_g(\xi) = 0,$$

where $\alpha^k_{gx} = \frac{\Delta x}{2}\sqrt{\left[D^k_g\right]^{-1}\Sigma^k_R}$

The sine and cosine are modified to preserve an orthogonality of the basic functions and normalized to unity at the right end of the interval $[-1,+1]$ in the same way as they are used in the mathematical analytic nodal method (MANM) developed by J. P. Hennart (1988). The expressions for the 3rd and 4th basic functions are as follows:

$$P^{SANM}_{g3}(\xi) = \frac{Sinh(\alpha^k_{gx}\xi) - m^k_{gx1}(sinh)\,P_1(\xi)}{Sinh(\alpha^k_{gx}) - m^k_{gx1}(sinh)};$$

$$P^{SANM}_{g4}(\xi) = \frac{Cosh(\alpha^k_{gx}\xi) - m^k_{gx0}(cosh)\,P_0(\xi) - m^k_{gx2}(cosh)\,P_2(\xi)}{Cosh(\alpha^k_{gx}) - m^k_{gx0}(cosh) - m^k_{gx2}(cosh)}$$

where

$$m^k_{gx1}(sinh) = \frac{1}{N_1}\int_{-1}^{1} \sinh(\alpha^k_{gx}\xi)\,P_1(\xi)\,d\xi;$$

$$m_{gxi}^{k}(cosh) = \frac{1}{N_i}\int_{-1}^{1} \cosh(\alpha_{gx}^{k}\xi)\, P_i(\xi)\, d\xi\,, \text{ for } i = 0,2\,;\;\; N_i = 2/(2i+1)\,,\; i = 0,1,2\,.$$

As a result the neutron flux is represented as

$$\Phi_x^k(\xi) = \bar{\Phi}^k + \sum_{i=1}^{4} \mathbf{P}_i^{SANM}(\xi)\,\mathbf{a}_{xi}^{k}\,,$$

where

$\mathbf{a}_{xi}^{k} = col\{a_{1xi}^{k},\ldots,a_{Gxi}^{k}\}$ is a vector of the unknown flux expansion coefficients;

$\mathbf{P}_i^{SANM}$ is the diagonal GxG matrix defined as

$$\mathbf{P}_i^{SANM} = P_i\,\mathbf{I}, \quad for \quad i = 1,2\,;$$

$$\left(\mathbf{P}_i^{SANM}\right)_{gg} = P_{gi}^{SANM}(\xi), \quad \left(\mathbf{P}_i^{SANM}\right)_{gg'} = 0, if\; g' \neq g \quad for \quad i = 3,4$$

The SANM equations are obtained in the same way as described in the previous section 2.6 for PNM. The 3rd flux expansion coefficient is expressed in terms of the 1st coefficient as:

$$\mathbf{a}_{x3}^{k} = \mathbf{A}_x^{k}\left\{\left(\mathbf{B}^2\right)_x^{k}\mathbf{a}_{x1}^{k} + \hat{\mathbf{s}}_{x1}^{k}\right\}, \tag{2.29}$$

where

$$\left(A_x^k\right)_{gg} = \left(\frac{\sinh\left(\alpha_{gx}^{k}\right) - m_{gx1}^{k}(sinh)}{\left(\alpha_{gx}^{k}\right)^2\, m_{gx1}^{k}(sinh)}\right),\;\; \left(A_x^k\right)_{gg'} = 0, if\; g' \neq g\,.$$

The 4 h expansion coefficient is expressed in terms of the 2n coefficient:

$$\mathbf{a}_{x4}^{k} = \mathbf{T}_x^{k}\left\{\left(\mathbf{B}^2\right)_x^{k}\mathbf{a}_{x2}^{k} + \hat{\mathbf{s}}_{x2}^{k}\right\}, \tag{2.30}$$

where

$$\left(T_x^k\right)_{gg} = \frac{\cosh(\alpha_{gx}^{k}) - m_{gx0}^{k}(cosh) - m_{gx2}^{k}(cosh)}{\left(\alpha_{gx}^{k}\right)^2\, m_{gx2}^{k}(cosh)},\;\; \left(T_x^k\right)_{gg'} = 0, if\;\; g' \neq g\,.$$

Substituting Eq. (2.30) into the neutron balance equation and Eq. (2.29) into the flux and current continuity equations, we again obtain G and 2G equations, which have to be solved for the two-node problem. They are as follows:

Neutron balance equation:

$$\left\{3\,\mathbf{I} + \mathbf{E}_x^{k}\left(\mathbf{B}^2\right)_x^{k}\right\}\mathbf{a}_{x2}^{k} = \left(\mathbf{B}^2\right)_x^{k}\bar{\Phi}^k + \left\{\hat{\mathbf{s}}_{x0}^{k} - \mathbf{E}_x^{k}\,\hat{\mathbf{s}}_{x2}^{k}\right\}$$

where

$$\left(E_x^k\right)_{gg} = \left(\frac{m_{gx0}^k(cosh)}{m_{gx2}^k(cosh)} - \frac{3}{\left(\alpha_{gx}^k\right)^2} \right), \quad \left(E_x^k\right)_{gg'} = 0, if \ g' \neq g$$

Flux continuity equation:

$$\left\{\mathbf{I} + \mathbf{A}_x^{k+1}\left(\mathbf{B}^2\right)_x^{k+1}\right\}\mathbf{a}_{x1}^{k+1} + \left\{\mathbf{I} + \mathbf{A}_x^{k}\left(\mathbf{B}^2\right)_x^{k}\right\}\mathbf{a}_{x1}^{k} = \left\{\bar{\Phi}^{k+1} + \mathbf{a}_{x2}^{k+1} + a_{x4}^{k+1}\right\} -$$

$$\left\{\bar{\Phi}^{k} + \mathbf{a}_{x2}^{k} + \mathbf{a}_{x4}^{k}\right\} - \left\{\mathbf{A}_x^k\ \hat{\mathbf{s}}_{x1}^k + \mathbf{A}_x^{k+1}\ \hat{\mathbf{s}}_{x1}^{k+1}\right\}$$

Current continuity equation:

$$\hat{\mathbf{D}}_x^{k+1}\left\{\mathbf{I} + \mathbf{F}_x^{k+1}\left(\mathbf{B}^2\right)_x^{k+1}\right\}\mathbf{a}_{x1}^{k+1} - \hat{\mathbf{D}}_x^{k}\left\{\mathbf{I} + \mathbf{F}_x^{k}\left(\mathbf{B}^2\right)_x^{k}\right\}\mathbf{a}_{x1}^{k} =$$

$$\hat{\mathbf{D}}_x^{k+1}\left\{3\,\mathbf{a}_{x2}^{k+1} + \mathbf{G}_x^{k+1}\,\mathbf{a}_{x4}^{k+1} - \mathbf{F}_x^{k+1}\,\hat{\mathbf{s}}_{x1}^{k+1}\right\} + \hat{\mathbf{D}}_x^{k}\left\{3\,\mathbf{a}_{x2}^{k} + \mathbf{G}_x^{k}\ \mathbf{a}_{x4}^{k} + \mathbf{F}_x^{k}\,\hat{\mathbf{s}}_{x1}^{k}\right\}$$

where

$$\left(F_x^k\right)_{gg} = \frac{\alpha_{gx}^k \cosh(\alpha_{gx}^k) - m_{gx1}^k(sinh)}{\left(\alpha_{gx}^k\right)^2\ m_{gx1}^k(sinh)}, \quad \left(F_x^k\right)_{gg'} = 0, if \ g' \neq g\ ;$$

$$\left(G_x^k\right)_{gg} = \frac{\alpha_{gx}^k\ \sinh(\alpha_{gx}^k)\ -\ 3\,m_{gx2}^k(cosh)}{\cosh(\alpha_{gx}^k) - m_{gx0}^k(cosh) - m_{gx2}^k(cosh)}, \quad \left(G_x^k\right)_{gg'} = 0, if \ g' \neq g\ .$$

When the nodal equations are solved, the face-averaged nodal current $\mathbf{J}_{x+}^k$ is computed as

$$\mathbf{J}_{x+}^k = \mathbf{J}_x^k(+1) = -\hat{\mathbf{D}}_x^k\ \left\{\ \mathbf{a}_{x1}^k + 3\,\mathbf{a}_{x2}^k + \mathbf{H}_x^k\,\mathbf{a}_{x3}^k + \mathbf{G}_x^k\,\mathbf{a}_{x4}^k\ \right\},$$

where

$$\left(H_x^k\right)_{gg} = \left[\frac{\alpha_{gx}^k \cosh(\alpha_{gx}^k) - m_{gx1}^k(sinh)}{\sinh(\alpha_{gx}^k) - m_{gx1}^k(sinh)} \right], \quad \left(H_x^k\right)_{gg'} = 0, if \ g' \neq g$$

2.8 Analytic Nodal Method (ANM)

In the analytic nodal method (ANM) (K. Smith, 1979), the transverse-integrated nodal equations (2.15) are solved analytically. An original formulation given in the thesis of K. Smith (1979) is bulky, we will present an alternative algorithm proposed by L. Pogosbekyan and recently incorporated into the SKETCH-N code (Zimin et. al., 1999). The algorithm is based on the matrix function theory. A brief

introduction to the matrix function theory is given in (Golub and Van Loan, 1996), more details can be found in (Gantmacher, 1959), (Pease, 1965). Several applications of the matrix functions to neutron diffusion and transport theory are described in textbook by Shikhov and Troyanskii (1981).
The general solution of the Eq. (2.15) is given as

$$\Phi_x^k(\xi) = \mathbf{sn}\left[\sqrt{\left(\mathbf{B}^2\right)_x^k}\xi\right]\mathbf{c1}_x^k + \mathbf{cs}\left[\sqrt{\left(\mathbf{B}^2\right)_x^k}\xi\right]\mathbf{c2}_x^k + \mathbf{R}_x^k(\xi), \tag{2.31}$$

where

constant vectors $\mathbf{c1}_x^k$ and $\mathbf{c2}_x^k$ are determined by the boundary conditions;

$\mathbf{R}_x^k(\xi)$ is a particular solution of Eq. (2.15) depending on the trasnverse leakage shape;

$\mathbf{sn}\left[\sqrt{\left(\mathbf{B}^2\right)_x^k}\xi\right]$, $\mathbf{cs}\left[\sqrt{\left(\mathbf{B}^2\right)_x^k}\xi\right]$ are the matrix functions determined by the buckling matrix $\left(\mathbf{B}^2\right)_x^k$.

The matrix functions $\mathbf{s}\left[\sqrt{\left(\mathbf{B}^2\right)_x^k}\xi\right]$ and $\mathbf{cs}\left[\sqrt{\left(\mathbf{B}^2\right)_x^k}\xi\right]$ are defined using their scalar analogs as

$$\mathrm{sn}\left[\sqrt{B^2}\xi\right] = \begin{cases}\sinh\left[B\xi\right], & if\ B \geq 0\\ \sin\left[|B|\xi\right], & if\ B < 0\end{cases};\quad \mathrm{cs}\left[\sqrt{B^2}\xi\right] = \begin{cases}\cosh\left[B\xi\right], & if\ B \geq 0\\ \cos\left[|B|\xi\right], & if\ B < 0\end{cases}.$$

If the quadratic leakage approximation is used, the particular solution $\mathbf{R}_x^k(\xi)$ is expressed in terms of the transverse leakage expansion coefficients as

$$\mathbf{R}_x^k(\xi) = \sum_{i=0}^{2} P_i(\xi)\,\mathbf{c}_{xi}^k,$$

where

$$\mathbf{c}_{x0}^k = -\left[\left(\mathbf{B}^2\right)_x^k\right]^{-1}\hat{\mathbf{s}}_{x0}^k - 3\left[\left(\mathbf{B}^2\right)_x^k\right]^{-2}\hat{\mathbf{s}}_{x2}^k;\quad \mathbf{c}_{x1}^k = -\left[\left(\mathbf{B}^2\right)_x^k\right]^{-1}\hat{\mathbf{s}}_{x1}^k;\quad \mathbf{c}_{x2}^k = -\left[\left(\mathbf{B}^2\right)_x^k\right]^{-2}\hat{\mathbf{s}}_{x2}^k.$$

To find the constants of the general solution of the homogeneous equation we need to define the boundary conditions. In the case of a two-node problem, the boundary conditions are given in terms of the node-averaged face-averaged fluxes at the internal face. Considering the node k they are given as

$$\frac{1}{2}\int_{-1}^{1}\Phi_x^k(\xi)d\xi = \bar{\Phi}^k \text{ and } \Phi_x^k(+1) = \Phi_{x+}^k.$$

Using these two constraints, the constants $\mathbf{c1}_x^k$ and $\mathbf{c2}_x^k$ are expressed in terms of the node-averaged flux $\bar{\Phi}^k$ and the face-averaged flux Φ_{x+}^k. Substituting the analytic solution (2.31) into the Fick's law (2.16) we can compute the face-averaged neutron current at the internal interface of the two-node problem. After some trigonometric manipulations, performed using the symbolic manipulator *Mathematica* (Wolfram S., 1996), we obtain the following expression for the face-averaged current

$$\mathbf{J}_{x+}^{k} = \mathbf{J}_{x}^{k}(+1) = -\hat{\mathbf{D}}_{x}^{k} \left\{ \mathbf{F}_1\left[\left(\mathbf{B}^2\right)_x^k\right]\bar{\mathbf{\Phi}}^{k} + \mathbf{F}_2\left[\left(\mathbf{B}^2\right)_x^k\right]\mathbf{\Phi}_{x+}^{k} + \sum_{i=0}^{2}\mathbf{F}_{i+3}\left[\left(\mathbf{B}^2\right)_x^k\right]\hat{\mathbf{s}}_{xi}^{k} \right\}, \quad (2.32)$$

where $\mathbf{F}_1\left[\left(\mathbf{B}^2\right)_x^k\right], \ldots, \mathbf{F}_5\left[\left(\mathbf{B}^2\right)_x^k\right]$ are the matrix functions.

The matrix functions $\mathbf{F}_1\left[\left(\mathbf{B}^2\right)_x^k\right], \ldots, \mathbf{F}_5\left[\left(\mathbf{B}^2\right)_x^k\right]$ are defined by the following scalar functions:

$$\{ f_1^{ANM}(x) , \ldots , f_5^{ANM}(x) \} = \left\{ x - \frac{1}{\Gamma^2}, \frac{1}{\Gamma}, 1 - \frac{1-\Gamma}{\Gamma^2 x}, \frac{1-\Gamma}{\Gamma x}, \frac{\Gamma(3+x)-3}{\Gamma^2 x^2} \right\}, \quad (2.33)$$

where

$$\Gamma = \frac{Tanh(\sqrt{x})}{\sqrt{x}} = \begin{cases} Tanh(\sqrt{|x|})/\sqrt{|x|}, & if \quad x \geq 0 \\ Tan(\sqrt{|x|})/\sqrt{|x|}, & if \quad x < 0 \end{cases}.$$

The only unknown in the expression (2.32) is the face-averaged neutron flux $\mathbf{\Phi}_{x+}^{k}$. To compute it, we consider the node $k+1$ of the two-node problem and obtain the analytical solution for this node. The procedure is almost the same as described above for the node k, the only difference we have in the boundary conditions. For the node k , the boundary conditions are given as

$$\frac{1}{2}\int_{-1}^{1} \Phi_x^{k+1}(\xi)d\xi = \bar{\Phi}^{k+1} \text{ and } \Phi_x^{k+1}(-1) = \Phi_{x-}^{k+1}.$$

As a result, the face-averaged current $\mathbf{J}_{x-}^{k+1}$ is expressed as

$$\mathbf{J}_{x-}^{k+1} = \mathbf{J}_{x}^{k+1}(-1) = \hat{\mathbf{D}}_{x}^{k+1} \left\{ \mathbf{F}_1\left[\left(\mathbf{B}^2\right)_x^{k+1}\right]\bar{\mathbf{\Phi}}^{k+1} + \mathbf{F}_2\left[\left(\mathbf{B}^2\right)_x^{k+1}\right]\mathbf{\Phi}_{x-}^{k+1} + \sum_{i=0}^{2}(-1)^i\mathbf{F}_{i+3}\left[\left(\mathbf{B}^2\right)_x^{k+1}\right]\hat{\mathbf{s}}_{xi}^{k+1} \right\}$$

Using the continuity conditions for the face-averaged flux

$$\mathbf{\Phi}_{x+}^{k} = \mathbf{\Phi}_{x-}^{k+1}$$

and the face-averaged current

$$\mathbf{J}_{x+}^{k} = \mathbf{J}_{x-}^{k+1}$$

we obtain a system of GxG equations to compute the face-averaged flux

$$\left\{ \hat{\mathbf{D}}_{x}^{k}\,\mathbf{F}_2\left[\left(\mathbf{B}^2\right)_x^k\right] + \hat{\mathbf{D}}_{x}^{k+1}\,\mathbf{F}_2\left[\left(\mathbf{B}^2\right)_x^{k+1}\right] \right\}\mathbf{\Phi}_{x+}^{k} = -\hat{\mathbf{D}}_{x}^{k} \left\{ \mathbf{F}_1\left[\left(\mathbf{B}^2\right)_x^k\right]\bar{\mathbf{\Phi}}^{k} + \sum_{i=0}^{2}\mathbf{F}_{i+3}\left[\left(\mathbf{B}^2\right)_x^k\right]\hat{\mathbf{s}}_{xi}^{k} \right\} -$$

$$\hat{\mathbf{D}}_{x}^{k+1} \left\{ \mathbf{F}_1\left[\left(\mathbf{B}^2\right)_x^{k+1}\right]\bar{\mathbf{\Phi}}^{k+1} + \sum_{i=0}^{2}(-1)^i\mathbf{F}_{i+3}\left[\left(\mathbf{B}^2\right)_x^{k+1}\right]\hat{\mathbf{s}}_{xi}^{k+1} \right\}. \quad (2.34)$$

After solving the Eq. (2.34), we can compute the face-averaged current using the expression (2.32).

During our derivation we ignore a problem of computing the matrix functions. There are several ways to make it. A comprehensive review of the methods to compute the matrix exponents gives nineteen dubious ways to make it (Moler and Van Loan, 1978). The authors use the word "dubious" to outline a loss of accuracy in the application of many methods to matrix exponents. In the application to our buckling matrices, which are well-conditioned, none of these methods looks dubious. A need to compute five matrix functions for each reactor node makes a selection of the most efficient method more important. In the case of two neutron energy groups, our choice of the optimal algorithm is the Lagrange-Sylvester polynomial representation of the matrix functions. In this case, eigenvalues of the buckling matrix are real and eigenvectors form a basis of G-dimensional space. As a result, the matrix function can be computed as (see, for example, Pease, 1965)

$$\mathbf{F}[\mathbf{B}] = \sum_{g=1}^{G} f(\lambda_g)\, \mathbf{E}_g[\mathbf{B}, \lambda_g], \text{ and } \mathbf{E}_g[\mathbf{B}, \lambda_g] = \prod_{j=1,\ j\neq g}^{G} \left(\mathbf{B} - \lambda_j \mathbf{I}\right) \Big/ \left(\lambda_g - \lambda_j\right),$$

where

$\mathbf{I}$ is an identity matrix;

$\mathbf{E}_g[\mathbf{B}, \lambda_g]$ is a projector into the eigensubspace of $\mathbf{B}$ corresponding to λ_g.

In the recent paper (Zimin et. al., 1998), we have shown that this way of the matrix function calculations results in the ANM algorithm, which is even faster than the conventional PNM algorithm described in Section 2.6. In the case, when a number of neutron energy groups more than two, eigenvalues of the buckling matrix are complex and computing time of the matrix functions drastically increases. The different algorithm should be considered in this case, for example, Pade approximation of the matrix functions or methods based on Schur decomposition. Because a lack of time we did not implement any these algorithms into the SKETCH-N code. So, if computing time is important, we recommend using SANM instead of ANM for calculations with more than 2 neutron energy groups.

It is worth to note that the matrix function approach can be applied to other nodal methods. Without going into details we only remark that using the scalar functions given below instead of the ANM functions defined by Eq. (2.33), we transform ANM into PNM.

$$\{\, f_1^{PNM}(x)\,,\ \ldots\,,\ f_5^{PNM}(x)\,\} = \left\{ \frac{1}{5}\left(-66 + 2x + \frac{2875}{15+x} - \frac{2744}{21+2x}\right),\right.$$
$$\left. 6 - \frac{75}{15+x}, \frac{525 + x(15+x)}{(15+x)\,(10.5+x)}, \frac{5}{15+x}, \frac{21(5+2x)}{5(15+x)(21+2)} \right\} \quad (2.35)$$

As a result we have an alternative formulation of PNM to that given in Section 2.6. Numerical experiments reported in (Zimin et. al., 1998) have shown that the matrix function formulation results in 25% saving in computing time in a comparison with the conventional algorithm described in Section 2.6.

Because $f_i^{ANM}(x) \to f_i^{PNM}(x),\ if\ x \to 0, i = 1,\ldots 5;$ the PNM functions (2.35) can be also used to remove singularities of the ANM functions (2.33) around zero. This way is used in (Downar et al., 1997). The approach is somehow awkward, because there is a need to remove singularities only for the eigenvalue, which is close to zero, while in this way it is used for the both eigenvalues. An alternative and more straightforward variant is an application of the Taylor or Pade expansion of the ANM matrix functions as described in (Noh et al., 1997). None of these methods is implemented into the SKETCH-N code. In our limited experience of using the ANM method, we have not found any instabilities of the ANM induced by the singular basic functions.

3. NEUTRON KINETICS MODEL

Equation Section 3

3.1 Neutron Kinetics Equations

In Cartesian geometry, few-group neutron kinetics equations in diffusion approximation are written as

$$\frac{1}{\mathrm{v}_g}\frac{\partial \varphi_g(r,t)}{\partial t} = \nabla D_g(r,t)\nabla \varphi_g(r,t) - \Sigma_{rg}(r,t)\varphi_g(r,t) + \sum_{\substack{g'=1 \\ g'\neq g}}^{G} \Sigma_{sgg'}(r,t)\varphi_{g'}(r,t) +$$

$$(1-\beta)\chi_g^p \sum_{g'=1}^{G} \nu\Sigma_{fg'}(r,t)\varphi_{g'}(r,t) + \chi_g^d \sum_{j=1}^{J} \lambda_j C_j(r,t)\,, \qquad (3.1)$$

$$\frac{\partial C_j(r,t)}{\partial t} = \beta_j \sum_{g'=1}^{G} \nu\Sigma_{fg'}(r,t)\varphi_{g'}(r,t) - \lambda_j C_j(r,t)\,, \quad g = 1,...,G; \quad j = 1,...,J; \qquad (3.2)$$

where to shorter notations we define x, y, z coordinates as r;

j is the delayed neutron precursor group index;

J is the total number of delayed precursor groups;

χ_g^p is the prompt neutron fission spectrum in group g;

χ_g^d is the delayed neutron fission spectrum in group g;

β_j is the delayed neutron yield fraction, group j;

$\beta = \sum_{j=1}^{J} \beta_j$ is the total yield of delayed neutrons per fission;

λ_j is the delayed neutron decay constant, group j;

C_j is the delayed neutron precursor group concentration.

the other notations are given in Section 2.1.

We convert the equations (3.1)-(3.2) into a matrix form as

$$\mathbf{v}^{-1}\frac{\partial \boldsymbol{\varphi}(r,t)}{\partial t} = \nabla \mathbf{D}(r,t)\nabla \boldsymbol{\varphi}(r,t) - \boldsymbol{\Sigma}_r(r,t)\boldsymbol{\varphi}(r,t) + \Sigma_s(r,t)\boldsymbol{\varphi}(r,t) +$$

$$(1-\beta)\boldsymbol{\chi}^p \nu\Sigma_f^T(r,t)\boldsymbol{\varphi}(r,t) + \boldsymbol{\chi}^d \sum_{j=1}^{J} \lambda_j C_j(r,t)\,, \qquad (3.3)$$

$$\frac{\partial C_j(r,t)}{\partial t} = \beta_j\, \nu\Sigma_f^T(r,t)\,\varphi\,(r,t) - \lambda_j C_j(r,t)\,, \quad j = 1,...,J;\,, \qquad (3.4)$$

where

$\mathbf{v}^{-1} = diag\{\mathrm{v}_1^{-1},\ldots,\mathrm{v}_G^{-1}\}$;

$\boldsymbol{\chi}^p = col\{\chi_1^p,\ldots,\chi_G^p\}$;

$\boldsymbol{\chi}^d = col\{\chi_1^d, \ldots, \chi_G^d\}$;

the other notations are given in Section 2.1.

The system of equations (3.3)-(3.4) is completed by the boundary conditions (2.5) and initial conditions

$$\varphi(r,0) = \varphi_0(r)\,; \;\; C_j(r,0) = C_{j0}(r)\,.$$

If we start from the results of the steady-state calculations, $\varphi_0(r)$ is solution of the steady-state eigenvalue problem (2.3)-(2.4) and the steady-steady state concentration of the delayed neutron precursors is computed as

$$C_{j0}(r) = \frac{\beta_j}{\lambda_j}\,\boldsymbol{\nu}\Sigma_f^T(r,t)\,\varphi_0(r) \tag{3.5}$$

In the transient calculations, we also normalise the multiplication cross section on the value of the steady-state multiplication factor as

$$\boldsymbol{\nu}\Sigma_f(r,t) = \boldsymbol{\nu}\Sigma_f(r,t)/k_{eff}\,.$$

To obtain a numerical solution of the neutron kinetics problem we need to perform time and space discretization. Let us start with discretization in time. First, we analytically integrate the delayed neutron precursors equations (3.4) assuming a linear dependence of the fission source $\boldsymbol{\nu}\Sigma_f^T(r,t)\,\varphi(r,t)$ at the time interval (Stacey, 1969). As a result we have

$$C_j(r,t+\Delta t) = C_j(r,t)\exp(-\lambda_j\,\Delta t) + \frac{\beta_j}{\lambda_j}\left\{\boldsymbol{\nu}\Sigma_f^T(r,t)\,\varphi(r,t)\left[\frac{1-\exp(-\lambda_j\,\Delta t)}{\lambda_j\,\Delta t} - \exp(-\lambda_j\,\Delta t)\right] + \right.$$

$$\left. \boldsymbol{\nu}\Sigma_f^T(r,t+\Delta t)\,\varphi(r,t+\Delta t)\left[1 - \frac{1-\exp(-\lambda_j\,\Delta t)}{\lambda_j\,\Delta t}\right]\right\} \tag{3.6}$$

Using the fully-implicit scheme for neutron flux equation (3.3) and the above expression (3.6) for the delayed neutron precursors we obtain

$$\left\{-\nabla\,\mathbf{D}(r,t+\Delta t)\,\nabla + \boldsymbol{\Sigma}_r(r,t+\Delta t) + \mathbf{v}^{-1}\frac{1}{\Delta t} - \Sigma_s(r,t+\Delta t) - \boldsymbol{\chi}^{NEW}\boldsymbol{\nu}\Sigma_f^T(r,t+\Delta t)\right\}\varphi(r,t+\Delta t) =$$

$$\left(\mathbf{v}^{-1}\frac{1}{\Delta t} + \boldsymbol{\chi}^{OLD}\boldsymbol{\nu}\Sigma_f^T(r,t)\right)\varphi(r,t) + \boldsymbol{\chi}^d\sum_{j=1}^{J}\lambda_j C_j(r,t)\exp(-\lambda_j\,\Delta t)\,, \tag{3.7}$$

where

$$\boldsymbol{\chi}^{NEW} = (1-\beta)\boldsymbol{\chi}^p + \sum_{j=1}^{J}\beta_j\,\boldsymbol{\chi}^d\left(1 - \frac{1-\exp(-\lambda_j\,\Delta t)}{\lambda_j\,\Delta t}\right);$$

$$\chi^{OLD} = \sum_{j=1}^{J} \beta_j \chi^d \left(\frac{1-\exp(-\lambda_j \Delta t)}{\lambda_j \Delta t} - \exp(-\lambda_j \Delta t) \right).$$

Please, note that Eq. (3.7)-(3.6) are decoupled, at each time step we first solve the neutron flux equation (3.7), which use only delayed neutron precursors from the previous time step and then update the delayed neutron precursors using Eq. (3.6).

The time-discretized equations (3.7)-(3.6) are discretized in space using the nonlinear iteration procedure described in Section 2.3. The coarse-mesh finite-difference method applied to Eq. (3.7)-(3.6) yields

$$\sum_{u=x,y,z} F_u^k \left\{ -[\tilde{\mathbf{D}}_u^k + \tilde{\mathbf{D}}_u^k(nod)]\bar{\Phi}^{k+1}(t+\Delta t) + [\tilde{\mathbf{D}}_u^k - \tilde{\mathbf{D}}_u^k(nod) + \tilde{\mathbf{D}}_u^{k-1} + \tilde{\mathbf{D}}_u^{k-1}(nod)]\bar{\Phi}^k(t+\Delta t) - \right.$$

$$-[\tilde{\mathbf{D}}_u^{k-1} - \tilde{\mathbf{D}}_u^{k-1}(nod)]\bar{\Phi}^{k-1}(t \quad \Delta t) \quad \left(\begin{matrix} k \\ r \end{matrix} \quad \mathbf{v}^{-1} \frac{1}{\Delta t} \quad \begin{matrix} k \\ s \end{matrix} \bar{\Phi}^k V^k \quad \chi^{NEW} \left(\begin{matrix} k \\ f \end{matrix} \right)^T \right) \bar{\Phi}^k (t \quad \Delta t) V^k =$$

$$\left. \left(\mathbf{v}^{-1} \frac{1}{\Delta t} + \chi^{OLD} (\nu\Sigma_f^k)^T(t) \right) \bar{\Phi}^k(t) V^k + \chi^d V^k \sum_{j=1}^{J} \lambda_j \bar{C}_j^k(t) \exp(-\lambda_j \Delta t) \right. \quad (3.8)$$

$$\bar{C}_j^k(t+\Delta t) = \bar{C}_j^k(t) \exp(-\lambda_j \Delta t) + \frac{\beta_j}{\lambda_j} \left\{ (\nu\Sigma_f^k)^T(t) \bar{\Phi}^k(t) \left[\frac{1-\exp(-\lambda_j \Delta t)}{\lambda_j \Delta t} - \exp(-\lambda_j \Delta t) \right] + \right.$$

$$\left. (\nu\Sigma_f^k)^T(t+\Delta t) \bar{\Phi}^k(t+\Delta t) \left[1 - \frac{1-\exp(-\lambda_j \Delta t)}{\lambda_j \Delta t} \right] \right\} \quad (3.9)$$

where

all cross sections at the left-hand side are evaluated at the time $(t+\Delta t)$,

$\bar{C}_j^k \equiv \frac{1}{V^k} \int_{-\Delta z/2}^{\Delta z/2} dz \int_{-\Delta y/2}^{\Delta y/2} dy \int_{-\Delta x/2}^{\Delta x/2} dx \, \bar{C}_j^k(x,y,z)$ is node-averaged concentration of the delayed neutron precursors;

the other notations are given in Sections 2.2 and 2.3.

To obtain the nodal equations, we apply at first the transverse-integration procedure to the neutron flux equations (3.7). The result follows

$$\left\{ -\mathbf{D}^k \frac{d^2}{dx^2} + \Sigma_r^k + \mathbf{v}^{-1} \frac{1}{\Delta t} - \Sigma_s^k - \chi^{NEW} (\nu\Sigma_f^k)^T \right\} \Phi_x^k(x,t+\Delta t) = -\mathbf{S}_x^k(x,t+\Delta t) + \mathbf{Q}_x^k(x,t), \quad (3.10)$$

where we again dropped time index $(t+\Delta t)$ of the cross sections at the left hand side and

$$\mathbf{Q}_x^k(x,t) = \left(\mathbf{v}^{-1} \frac{1}{\Delta t} + \chi^{OLD} (\nu\Sigma_f^k)^T(t) \right) \Phi_x^k(x,t) + \chi^d \sum_{j=1}^{J} \lambda_j C_{jx}^k(x,t) \exp(-\lambda_j \Delta t);$$

$C_{jx}^k(x) \equiv \frac{1}{F_x^k} \int_{-\Delta y/2}^{\Delta y/2} dy \int_{-\Delta z/2}^{\Delta z/2} dz\, C_j^k(x,y,z)$ is the transverse-integrated concentration of the delayed neutron precursors, group j.

Straightforward application of the nodal methods to Eq. (3.10) requires knowledge of the higher-order expansion coefficients of the flux and precursor concentration from the previous time step. To reduce the computer memory requirements we follow a trick proposed by Engrand et al. (1992). The neutron flux equation (3.10) is led to the form of the steady-state equation (2.13) grouping all additional terms into the transverse-leakage term $\tilde{\mathbf{S}}_x^k(x,t+\Delta t)$. As a results we have

$$\left\{-\mathbf{D}^k \frac{d^2}{dx^2} + \Sigma_r^k\right\} \Phi_x^k(x,t+\Delta t) = \left\{\Sigma_s^k + \chi^p (\nu\Sigma_f^k)^T\right\} \Phi_x^k(x,t+\Delta t) - \tilde{\mathbf{S}}_x^k(x,t+\Delta t), \quad (3.11)$$

where

$$\tilde{\mathbf{S}}_x^k(x,t+\Delta t) = \mathbf{S}_x^k(x,t+\Delta t) + \mathbf{v}^{-1} \frac{1}{\Delta t}\left\{\Phi_x^k(x,t+\Delta t) - \Phi_x^k(x,t)\right\} - \chi^{OLD} (\nu\Sigma_f^k)^T(t)\, \Phi_x^k(x,t)$$

$$-\left\{\chi^{NEW} - \chi^p\right\} (\nu\Sigma_f^k)^T(t+\Delta t)\, \Phi_x^k(x,t+\Delta t) - \chi^d \sum_{j=1}^{J} \lambda_j C_{jx}^k(x,t) \exp(-\lambda_j \Delta t).$$

Equation (3.11) is practically identical to the steady-state equation (2.13). As a result all the nodal methods described in the Chapter 2 are applied in the same way to the neutron kinetics problems. The only difference is the calculation of the transverse-leakage term $\tilde{\mathbf{S}}_x^k(x,t+\Delta t)$.

3.2 Automatic Time Step Size Control

Most of the modern neutron kinetics codes use an automatic time step control procedure in order to obtain a solution with a given tolerance (Aviles, 1993; Taiwo et al., 1993; Crouzet and Turinsky, 1996; Joo et al., 1996). Moreover, a variable time step size usually decreases a total amount of time steps and respectively computing time. We use the standard time-step doubling strategy (Hairer et al., 1987) in order to estimate a temporal truncation error and to predict a time step size. The procedure is mathematically well-founded and described in details elsewhere (Hairer et al., 1987; Crouzet and Turinsky, 1996), we only shortly outline the main features.

Time integration from the time moment t_j is performed on the two temporal meshes: on the fine mesh with the time step size Δt_j and on the coarse mesh with the time step size $2\Delta t_j$ as shown in Fig. 3.1.

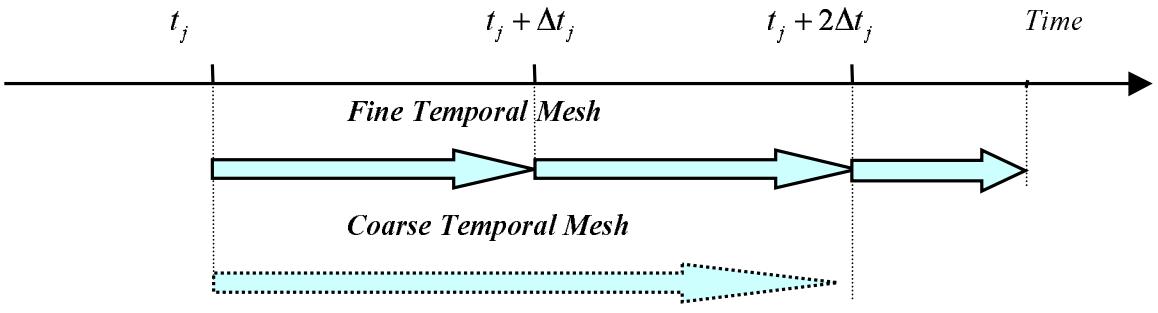

Figure 3.2. Time stepping in the time step doubling technique

As a result at the time moment $t_j + 2\Delta t_j$ we have the two solutions: $\Phi_{\Delta t_j}$ and $\Phi_{2\Delta t_j}$, respectively. As a solution we take the node-averaged power distribution because generally we want to control accuracy of the computed power. The local error of the fully-implicit scheme can be expressed as follows

$$Error = Const\ \Delta t^2 + O(\Delta t^3), \tag{3.12}$$

where constant $Const$ is proportional to the 2nd derivative of the neutron flux.

Applying the Richardson extrapolation for the two obtained solutions and neglecting the higher order terms we can obtain the temporal truncation error of the solution $\Phi_{\Delta t_j}$ as

$$Error = \left\| \Phi_{\Delta t_j} - \Phi_{2\Delta t_j} \right\|. \tag{3.13}$$

A most appropriate norm in the expression (3.13) is L_∞-norm if we wants to held the local errors of the power distribution within a given tolerance. However, an application of L_2-norm demonstrates more robust qualities for the time step size selection (Crouzet and Turinsky, 1996). In addition, the error, which has to be controlled, is a relative one and the expression (3.13) is replaced by the following one

$$Error = \left\| \Phi_{\Delta t_j} - \Phi_{2\Delta t_j} \right\|_2 \Big/ \left\| \Phi_{\Delta t_j} \right\|_2.$$

Using the expression (3.12) and requiring that the temporal truncation error is equal to the prescribed tolerance we get a prediction of the new time step size as

$$\Delta t^{NEW} = \sqrt{\frac{Tolerance}{Errror}}\ \Delta t_j.$$

A time step acceptance criterion can be written in the form

$$Errror \le Tolerance.$$

If the estimated error satisfies this criterion, the j-th time step is accepted and $\Delta t_{j+1} = \Delta t^{NEW}$ is used as a next time step size. Otherwise, the current time step is rejected and repeated with $\Delta t_j = \Delta t^{NEW}$.
In order to avoid repeated time step size rejections, safety parameters are introduced. The new time step size is calculated as

$$\Delta t^{NEW} = \Delta t_j \min\left[facmax, \max\left[facmin, safety \frac{\Delta t^{NEW}}{\Delta t_j} \right] \right],$$

where $facmax$ and $facmin$ are equal to 2 and 0.5; the typical value of the parameter $safety$ is in a range 0.8-0.9.

Performing calculations we often want to obtain a solution at the prescribed time moments. In order to handle this problem without abrupt change in the time step size, an additional modification of the new time step size Δt^{NEW} is done as follows

$$\Delta t^{NEW} = \frac{Time - t_j}{\left\lceil \frac{Time - t_j}{\Delta t^{NEW}} - \varepsilon \right\rceil},$$

where

Time is the prescribed time moment,

ε is a small multiple of the machine roundoff error;

$\lceil \circ \rceil$ denotes the integer ceiling function.

3.3 Point Kinetics Model

Three-dimensional neutron kinetics calculations are time-consuming and a code user usually wants to know an influence of space-dependent effects on transient results. To provide such capabilities, the SKETCH-N code has a point kinetics model, where neutron flux shape is assumed to be a constant during a transient. Thus, a comparison of the results computed by the three-dimensional model and the point kinetics model can be done using the same code.

The basic assumption of the point kinetics model is that the neutron flux vector can be factored into the product of the neutron flux shape, which does not vary in time, and amplitude function as

$$\varphi(r,t) = T(t)\,\psi(r), \tag{3.14}$$

where the shape function is normalised such that $\left\langle \mathbf{w}(r),\, \mathbf{v}^{-1}\,\psi(r) \right\rangle = 1$; $\mathbf{w}(r)$ is the weight function and the scalar product operation $\left\langle \circ, \circ \right\rangle$ is the integration over the reactor domain and neutron energy groups.

Weight function $\mathbf{w}(r)$ is usually chosen as a solution of the steady-state adjoint eigenvalue problem $\varphi_0^+(r)$. This choice eliminates the first-order errors in the reactivity due to the change of the neutron flux shape (Henry, 1975). A solution of the adjoint eigenvalue problem is performed in the SKETCH-N code before the transient calculations. Substituting the factorization (3.14) into Eq. (3.3)-(3.4), multiplying by the weight function $\varphi_0^+(r)$ and integrating over the reactor volume yields the point kinetics equations

$$\frac{dT(t)}{dt} = \frac{[\rho - \beta]}{\Lambda}\; T(t) + \sum_{j=1}^{J} \lambda_j c_j(t), \tag{3.15}$$

$$\frac{dc_j(t)}{dt} = \frac{\beta_j}{\Lambda}\; T(t) - \lambda_j c_j(t), \quad j = 1, \ldots, J; \tag{3.16}$$

where the point kinetics parameters are defined as

$$\rho = \rho(t) = \left\langle \varphi_0^+(r), \left[\nabla\, \mathbf{D}(r,t)\, \nabla - \Sigma_r(r,t) + \Sigma_s(r,t) + \left((1-\beta)\chi^p + \beta\, \chi^d \right) \nu\Sigma_f^T(r,t) \right] \psi(r) \right\rangle / F;$$

$$\beta_j = \beta_j(t) = \left\langle \varphi_0^+(r),\, \beta_j\, \chi^d\, \nu\Sigma_f^T(r,t)\, \psi(r) \right\rangle / F;$$

$$\beta = \sum_{j=1}^{J} \beta_j \ ;$$

$$\Lambda = \Lambda(t) = \left\langle \varphi_0^+(r),\, \mathbf{v}^{-1}\psi(r) \right\rangle / F = 1/F \ ;$$

$$F = F(t) = \left\langle \varphi_0^+(r),\, \left((1-\beta)\chi^p + \beta\,\chi^d\right) \nu\Sigma_f^T(r,t)\ \psi(r) \right\rangle ;$$

$$c_j(t) = \left\langle \varphi_0^+(r),\, \chi^d\ C_j(r,t) \right\rangle / F \ .$$

The system of equations (3.15)-(3.16) is completed by the initial conditions as

$$T(0) = \left\langle\ \varphi_0^+(r),\, \mathbf{v}^{-1}\ \varphi_0(r)\ \right\rangle ;$$

$$c_j(t) = \left\langle \varphi_0^+(r),\, \chi^d\ C_j(r,0) \right\rangle / F \ ;$$

where the initial steady-state concentration of the delayed neutron precursors is determined by Eq. (3.5). Time discretization of the point kinetics equations is performed by the same fully-implicit scheme with analytical integration of the delayed neutron precursors as in the case of the three-dimensional model (Stacey, 1969). The time-discretized point kinetics equations are

$$T(t+\Delta t) = \frac{T(t)\left\{\dfrac{1}{\Delta t} + \dfrac{1}{\Lambda}\displaystyle\sum_{j=1}^{J}\beta_j \left[\dfrac{1-\exp(-\lambda_j\Delta t)}{\lambda_j\Delta t} - \exp(-\lambda_j\Delta t)\right]\right\} + \displaystyle\sum_{j=1}^{J}\lambda_j c_j(t)\exp(-\lambda_j\Delta t)}{\left\{\dfrac{1}{\Delta t} - \dfrac{\rho(t)}{\Lambda} + \dfrac{1}{\Lambda}\displaystyle\sum_{j=1}^{J}\beta_j\left[\dfrac{1-\exp(-\lambda_j\Delta t)}{\lambda_j\Delta t}\right]\right\}}$$

$$c_j(t+\Delta t) = c_j(t)\exp(-\lambda_j\Delta t) + \frac{\beta_j}{\lambda_j\ \Lambda}\left\{\ T(t)\left[\frac{1-\exp(-\lambda_j\Delta t)}{\lambda_j\Delta t} - \exp(-\lambda_j\Delta t)\right] + \right.$$

$$\left. T(t+\Delta t)\left[1 - \frac{1-\exp(-\lambda_j\Delta t)}{\lambda_j\Delta t}\right]\right\}$$

Automatic time step control procedure described in the previous Section 3.2 is also applied for the point kinetics equations.

Before leaving this section we would like to note that the point kinetics parameters in the SKETCH-N code are generated automatically at each time step from the current values of the macro cross sections. Macro cross sections are recomputed at each time step depending on the control rod position and values of thermal-hydraulics feedbacks. Thus, a code user is liberated from the difficult and time-consuming task of preparation of the point kinetics parameters, which he has to do if he is using the transient analysis code with a separate point kinetics model.

4. ITERATIVE SOLUTION OF THE STEADY-STATE EIGENVALUE PROBLEMS

Equation Section 4

The nonlinear iteration procedure introduced in Chapter 2 results that a solution of the neutron diffusion equations (2.3)-(2.4) is decoupled into a global solution of the coarse-mesh finite-difference equations (2.11) and local solutions of the two-node nodal equations. The solution of the nodal equations of the polynomial, semi-analytic and analytic methods is discussed in the Chapter 2, while the iterative solution techniques applied to the equation (2.11) are considered in this Chapter.

4.1 Steady-State Eigenvalue Problem

The coarse-mesh finite-difference equation (2.11) can be written in the matrix form as

$$\mathbf{M}[\Phi,\lambda,\mathbf{T}]\,\Phi=\frac{1}{\lambda}\mathbf{F}[\mathbf{T}]\,\Phi, \tag{4.1}$$

where

λ is the eigenvalue, the reactor multiplication factor k_{eff};

$\Phi=col\{\bar{\Phi}^1,\ldots,\bar{\Phi}^K\}$ is the vector of the node-averaged neutron flux of size KG, K is the total number of the nodes, G is the number of neutron energy groups;

$\mathbf{M}[\Phi,\lambda,\mathbf{T}]$ is the $KG\times KG$ matrix, containing the finite-difference entries of the diffusion, removal and scattering operators; the dependency of the matrix $\mathbf{M}$ on the neutron flux Φ, eigenvalue λ and thermal-hydraulics parameters $\mathbf{T}$ is explicitly stated;

$\mathbf{F}[\mathbf{T}]$ is the $KG\times KG$ matrix, containing the finite-difference entries of the multiplication operator $\chi(\nu\Sigma_f^k)^T$, the matrix depends on the thermal-hydraulics feedbacks $\mathbf{T}$.

A solution of the steady-state eigenvalue problem (4.1) involves several iteration levels as shown in Fig. 4.1. A top level is the thermal-hydraulics iterations, where we recompute the macro cross sections and the finite-difference coefficients of the matrices and $\mathbf{F}$ depending on the values of the thermal hydraulics feedbacks (fuel temperature, coolant density etc.). The thermal-hydraulics feedbacks depend on the power distribution and respectively on the neutron flux shape. The calculation of the thermal-hydraulics parameters using an internal thermal-hydraulics module is discussed in Chapters 8; an external thermal-hydraulics model can be coupled with the SKETCH-N code using an interface module described in Chapter 9. A next iteration level is the nonlinear iterations, where the nodal coupling coefficients of the matrix are updated depending on the node-averaged neutron fluxes and the eigenvalue. The solution of the two-node nodal equations and recalculation of the nodal coupling coefficients are given in Chapter 2. With fixed thermal-hydraulics feedbacks and the nodal coupling coefficients the problem (4.1) is reduced to a linear eigenvalue problem as

$$\mathbf{M}\,\Phi=\frac{1}{\lambda}\mathbf{F}\,\Phi, \tag{4.2}$$

Sections 4.2 and 4.3 present the outer-inner iteration techniques used for solving Eq. (4.2).

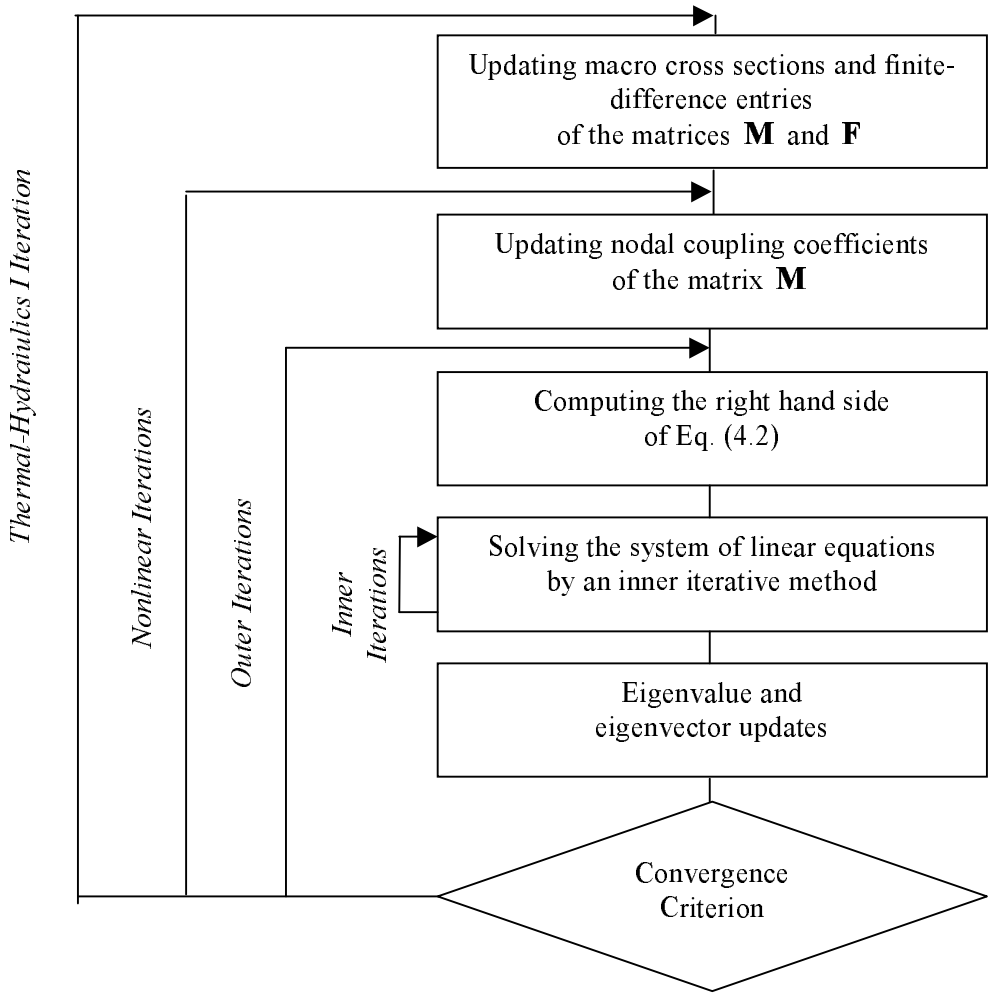

Figure 4.3. Iterative levels of the steady-state eigenvalue problem

In the code, we combine the thermal-hydraulics and nonlinear iterations into the one level called nonlinear iterations. The nonlinear iterations terminate when the outer iterations solving the linearized eigenvalue problem (4.2) converge in one iteration. A fixed number of outer iterations per nonlinear iteration is used if the convergence of the outer iterations is not reached. The number depends on the problem at hand, an optimal convergence is observed with 8-15 outer iterations per nonlinear iteration for performed LWR calculations. An algorithm of the global iterations is given in Fig. 4.2.

Let us make some theoretical remarks on the solution of the linear eigenvalue problem (4.2). In the limit to zero of the spatial mesh size, the nodal coupling coefficients of the matrix also approach zero and the matrix has the following properties:

- renumbering the vector Φ first by mesh points, second by neutron groups results in the matrix $\mathbf{A}$ with G blocks $\mathbf{M}_g$ size of $K \times K$, which are symmetric;
- diagonal elements of $\mathbf{M}$ are positive;
- off-diagonal elements of $\mathbf{M}$ are nonpositive (negative or zero);
- $\mathbf{M}$ is diagonally dominant;
- $\mathbf{M}$ is irreducible.

Under these conditions, theoretical analysis has been performed using Perron-Frobenious theory of nonnegative matrices, see Section 3.3 in (Wachspress, 1966). It is proved that the problem has simple real eigenvalue λ_1, such that $\lambda_1 > |\lambda_i|, i = 2, \ldots$ and the corresponding eigenvector $\Phi_1 > 0$.

choose initial guess $\Phi^{(0)} > 0$ *and* $\lambda^{(0)} > 0$
choose number of outer iterations
per nonlinear N_Outer
DO $m = 1, 2, \ldots$
compute thermal – hydraulics feedbacks
compute macro cross sections
compute matrices $\mathbf{M}$ *and* $\mathbf{F}$
initialise outer and inner iterations
DO $n = 1, 2, \ldots, N\_Outer$
perform outer iteration
IF *convergence* EXIT
END DO
IF $n == 1$ EXIT
END DO

Figure 4.2. Algorithm of the nonlinear iterations of the steady-state eigenvalue problem

In general case, when the nodal coupling coefficients are not zero, the above conditions are not guaranteed, for example, the diagonal dominance can be violated for the nodes with high gradient of the neutron flux. No theoretical analysis has been done in this case. However, for practical LWR calculations with a spatial mesh size equals to a size of a fuel assembly or smaller, the above conditions, except the first one, are valid at least for the most of the reactor nodes. As a result, iterative methods developed for traditional finite-difference codes can be also applied for solving the Eq. (4.2). The outer-inner iteration techniques used in the code are given in the following Sections 4.2 and 4.3.

4.2 Outer Iterations

A power method (Nakamura, 1977) can be used for solving the generalized eigenvalue problem (4.2). An algorithm of the method is given in Fig. 4.3, where $\mathbf{e} = col\{1, 1, \ldots, 1\}$ is the unity vector; $\langle \circ, \circ \rangle$ denotes the dot product.

choose initial guess $\Phi^{(0)} > 0$ *and* $\lambda^{(0)} > 0$
choose number of outer iterations
per nonlinear N_Outer
DO $n = 1, 2, \ldots, N\_Outer$
solve: $\mathbf{M}\,\Phi^{(n)} = \frac{1}{\lambda^{(n-1)}}\,\mathbf{F}\,\Phi^{(n-1)}$
$\lambda^{(n)} = \lambda^{(n-1)} \frac{\langle \mathbf{e}, \mathbf{F}\,\Phi^{(n)} \rangle}{\langle \mathbf{e}, \mathbf{F}\,\Phi^{(n-1)} \rangle}$
IF *convergence* EXIT
END DO

Figure 4.3. Algorithm of the power method applied to the generalized eigenvalue problem

The iterations of the power method are called outer iterations, because a system of linear equations

$$\mathbf{M}\,\mathbf{\Phi}^{(n)} = \frac{1}{\lambda^{(n-1)}}\,\mathbf{F}\,\mathbf{\Phi}^{(n-1)} \tag{4.3}$$

is usually also solved by an iterative method forming inner iterations. A convergence rate of the power method is determined by the dominance ratio $d = \frac{\lambda_2}{\lambda_1}$, where λ_2 is the second largest eigenvalue of the generalized eigenvalue problem (4.2). Decreasing the dominance ratio d increases the convergence. In practical LWR models, the dominance ratio is close to unity and the convergence may be very slow.

The simplest and the most popular method to improve the convergence of the power method is to move a part of the right hand side of Eq. (4.2) into the left hand side as

$$\left[\mathbf{M} - \frac{1}{\lambda'}\mathbf{F}\right]\mathbf{\Phi} = \frac{1}{\Lambda}\mathbf{\Phi} \tag{4.4}$$

$$\frac{1}{\Lambda} = \frac{1}{\lambda} - \frac{1}{\lambda'},$$

where λ' is the chosen eigenvalue shift.

The dominance ratio of the shifted eigenvalue problem (4.4) is given as

$$d' = \frac{\frac{1}{\lambda_1} - \frac{1}{\lambda'}}{\frac{1}{\lambda_2} - \frac{1}{\lambda'}}$$

If λ' is larger than λ_1, the dominance ratio is less than unity and less than the dominance ratio d of the original eigenvalue problem (4.2). If λ' is equal to the eigenvalue λ_1, the dominance ratio is zero, but we may encounter a problem of solving the linear system by inner iterations, because the matrix $\left[\mathbf{M} - \frac{1}{\lambda'}\mathbf{F}\right]$ is singular. In practice, the eigenvalue shift λ' is computed during the iterations as follows

$$\lambda' = \lambda^{(n)} + \delta\lambda,$$

where $\lambda^{(n)}$ is the current estimate of the eigenvalue and $\delta\lambda$ is the chosen shift parameter.
An optimal choice of the shift parameter is a problem dependent. Decreasing the shift parameter increases the convergence rate of the power iterations, but decreases the convergence rate of the inner iterations. In the performed LWR calculations, an optimal choice of the shift parameter was in the interval $[0.02, 0.1]$. Applying the power method to the shifted problem (4.4) we get the Wieland method (Nakamura, 1977), also called the method of fractional iterations (Wachspress, 1966). An algorithm of the Wieland method is given in Fig. 4.4.

choose initial guess $\Phi^{(0)} > 0$ *and* $\lambda^{(0)} > 0$

choose shift parameter $\delta\lambda$

choose number of outer iterations per nonlinear N_Outer

DO $n = 1, 2, \ldots, N\_Outer$

$$\lambda' = \lambda^{(n-1)} + \delta\lambda$$

$$solve: \left[\mathbf{M} - \frac{1}{\lambda'}\mathbf{F} \right] \Phi^{(n)} = \frac{1}{\Lambda^{(n-1)}} \mathbf{F}\, \Phi^{(n-1)}$$

$$\Lambda^{(n)} = \Lambda^{(n-1)} \frac{\left\langle \mathbf{e}, \mathbf{F}\, \Phi^{(n)} \right\rangle}{\left\langle \mathbf{e}, \mathbf{F}\, \Phi^{(n-1)} \right\rangle}$$

$$\lambda^{(n)} = \frac{\Lambda^{(n)} \lambda'}{\Lambda^{(n)} + \lambda'}$$

IF *convergence* EXIT

END DO

Figure 4.4. Algorithm of the Wieland method for generalized eigenvalue problem

The convergence of the Wieland method can be further accelerated by applying Chebyshev polynomials. Application of Chebyshev polynomials for acceleration of the power method is discussed in Section 11.2 of (Hageman and Young, 1981). An extension of the method for the shifted eigenvalue problem (4.4) is straightforward. Let us order the eigenvalues of the matrix $\left[\mathbf{M} - \frac{1}{\lambda'}\mathbf{F} \right]^{-1} \mathbf{F}$ as

$$\Lambda_1 > |\Lambda_2| \geq |\Lambda_3| \geq \cdots \geq |\Lambda_{KG}|.$$

To apply the Chebyshev acceleration we use the following assumptions

1) the eigenvalues Λ_i are real;
2) the estimate of the dominance ratio $d' = \frac{\Lambda_2}{\Lambda_1}$ and the ratio of the minimum to maximum eigenvalues $b = \frac{\Lambda_{KG}}{\Lambda_1}$ are available.

In the practical LWR calculations, d' may be chosen from the interval $[0.9, 0.98]$, while $b = 0$. An adaptive procedure to estimate the dominance ratio has been developed based on a comparison of the observed and theoretical convergence rate (Hageman and Young, 1981). However, the adaptive procedure is not reliable in the steady-state calculations, because the both matrices $\mathbf{M}$ and $\mathbf{F}$ are changed at the nonlinear iterations. Algorithm of the Chebyshev acceleration for the power method is taken from the Section 11.2 of (Hageman and Young, 1981). We introduce only two modifications:

1) Chebyshev iterations are started when several power iterations are performed. It increases stability of the algorithm when the matrices and $\mathbf{F}$ are changed significantly in the top nonlinear iteration level.
2) Recalculation of the eigenvalue shift λ is moved into the top nonlinear iteration level. As a result the matrix $\left[\mathbf{M} - \frac{1}{\lambda'}\mathbf{F} \right]$ does not change during the outer iterations and the Chebyshev acceleration procedure works efficiently.

An algorithm of the Chebyshev acceleration procedure is given in Fig. 4.5.

choose initial guess $\Phi^{(0)} > 0$ *and* $\lambda^{(0)} > 0$

choose shift parameter $\delta\lambda$

choose dominace ratio d' *and ratio* b

chose iteration number to start Chebyshev
iterations N_Cheb_Start

choose number of outer iterations
per nonlinear N_Outer

$$p = 0; \quad \gamma = \frac{2}{(2 - d' - b)}; \quad \sigma = \frac{d' - b}{2 - d' - b}$$

$$\lambda' = \lambda^{(0)} + \delta\lambda$$

DO $n = 1, 2, \ldots, N\_Outer$

$$solve: \left[\mathbf{M} - \frac{1}{\lambda'} \mathbf{F} \right] \Phi^{(n)} = \frac{1}{\Lambda^{(n-1)}} \mathbf{F}\, \Phi^{(n-1)}$$

IF $n \geq N\_Cheb\_Start$ THEN

IF $p == 0$ THEN

$$\rho = 1.0$$

ELSE IF $p == 1$ THEN

$$\rho = 1 \Big/ \left(1 - \frac{1}{2} \sigma^2 \right)$$

ELSE

$$\rho = 1 \Big/ \left(1 - \frac{1}{4} \sigma^2 \rho \right)$$

END IF

$$p = p + 1$$

$$\Phi^{(n)} = \rho \left(\gamma\, \Phi^{(n)} + (1 - \gamma)\, \Phi^{(n-1)} \right) + (1 - \rho)\, \Phi^{(n-2)}$$

END IF

$$\Lambda^{(n)} = \Lambda^{(n-1)} \frac{\left\langle \mathbf{e}, \mathbf{F}\, \Phi^{(n)} \right\rangle}{\left\langle \mathbf{e}, \mathbf{F}\, \Phi^{(n-1)} \right\rangle}$$

$$\lambda^{(n)} = \frac{\Lambda^{(n)} \lambda'}{\Lambda^{(n)} + \lambda'}$$

IF *convergence* EXIT

END DO

Figure 4.5. Algorithm of the Chebyshev acceleration procedure applied to the Wieland method for generalized eigenvalue problem

The algorithm 4.5 is the final algorithm used in the code. Setting $N\_Cheb\_Start > N\_Outer$ we get the Wieland method, because the Chebyshev iterations are not started; if additionally $1/\lambda' = 0$ we obtain the simple iterations of the power method.

The following convergence tests are applied

$$convergence: \ \frac{\max\limits_k \left|\Phi_k^{(n)} - \Phi_k^{(n-1)}\right|}{\max\limits_k \left|\Phi_k^{(n)}\right|} \le \varepsilon_\Phi \ \text{and} \ \frac{\left|k_{eff}^{\max} - k_{eff}^{\min}\right|}{2} \le \varepsilon_\lambda$$

where ε_Φ is the tolerance of the neutron flux; ε_λ is the tolerance of the eigenvalue;

$$k_{eff}^{\max} = \max_k \frac{\left|(\mathbf{F}\,\Phi)_k^{(n)}\right|}{\left|(\mathbf{F}\,\Phi)_k^{(n-1)}\right|}; \quad k_{eff}^{\min} = \min_k \frac{\left|(\mathbf{F}\,\Phi)_k^{(n)}\right|}{\left|(\mathbf{F}\,\Phi)_k^{(n-1)}\right|}.$$

The default values of the both tolerances ε_Φ and ε_λ are 10^{-5}.

4.3 Inner Iterations

At each outer iterations we should solve the linear system

$$\mathbf{A}\,\mathbf{x} = \mathbf{b}, \tag{4.5}$$

$$\text{where } \mathbf{A} \equiv \left[\mathbf{M} - \frac{1}{\lambda'}\mathbf{F}\right]; \ \mathbf{x} \equiv \Phi^{(n)}; \ \mathbf{b} = \frac{1}{\Lambda^{(n-1)}}\mathbf{F}\,\Phi^{(n-1)}.$$

The matrix $\mathbf{A}$ is large and sparse, iterative methods are used for solving Eq. (4.5) forming inner iterations. An application of the iterative method is further motivated by the fact that we do not need to solve Eq. (4.5) exactly, because the right hand side is known only approximately in the beginner of outer iterations. Several iteration strategies are possible to minimise total computing time of the outer and inner iterations, for detail discussion see (Hageman and Young, 1981; Marchuk and Lebedev, 1981; Fergusson and Derstine, 1977). In the SKETCH-N code, we apply a fixed number of inner iterations per outer. In the performed calculations, two inner iterations per outer iteration result in the minimum computing time.

The matrix $\mathbf{A}$ has a block structure with K blocks of the size $G \times G$, where G is a number of neutron energy groups; K is a total number of nodes. The number of energy groups is small in LWR calculations and the matrix blocks are easy to invert and multiply. Thus, the block iterative methods may be efficient for solving Eq. (4.5). In the code, we apply the block symmetric Gauss-Seidel method (Hageman and Young, 1981; Varga, 1962). Let us start with the block Gauss-Seidel method, the i-th iteration of the method can be written as

$$\begin{aligned} &\text{DO } k = 1,2,\ldots,K \\ &\quad \mathbf{x}_k^{(i)} = \mathbf{A}_{kk}^{-1}\left(\mathbf{b}_k - \sum_{j<k}\mathbf{A}_{kj}\mathbf{x}_j^{(i)} - \sum_{j>k}\mathbf{A}_{kj}\mathbf{x}_j^{(i-1)}\right) \\ &\text{END DO} \end{aligned} \tag{4.6}$$

where k is the node index, $\mathbf{x}_k = col\{x_k^1, x_k^2, \ldots, x_k^G\}$; $\mathbf{b}_k = col\{b_k^1, b_k^2, \ldots, b_k^G\}$; $\mathbf{A}_{kk}$ is the diagonal block of the matrix $\mathbf{A}$; $\mathbf{A}_{kj}$ is the off-diagonal block located below the block diagonal if $j < k$ and above the block diagonal if $j > k$.

The block symmetric Gauss-Seidel method combines the forward sweep of the Gauss-Seidel method (4.6) and a backward sweep updating the unknowns in reverse order as

$$\begin{aligned}
&\text{DO } k = K, K-1, \ldots, 1 \\
&\quad \mathbf{x}_k = \mathbf{A}_{kk}^{-1}\left(\mathbf{b}_k - \sum_{<k} \mathbf{A}_k \mathbf{x}^{-1} - \sum_{>k} \mathbf{A}_k \mathbf{x} \right) \\
&\text{END DO}
\end{aligned}$$

An algorithm of the symmetric Gauss-Seidel method is given in Fig. 4.6.

choose initial guess $\mathbf{x}^{(0)}$
choose number of inner iterations N_Inner
per outer iteration
$\mathbf{x}^{(1/2)} = \mathbf{x}^{(0)}$
compute $\mathbf{A}_{kk}^{-1}$
DO $i = 1, 2, \ldots, N\_Inner$
  DO $k = 1, 2, \ldots, K$
    $\mathbf{r} = 0$
    DO $j < k$
      $\mathbf{r} = \mathbf{r} + \mathbf{A}_{kj}\mathbf{x}_j^{(i-1/2)}$
    END DO
    DO $j > k$
      $\mathbf{r} = \mathbf{r} + \mathbf{A}_{kj}\mathbf{x}_j^{(i-1)}$
    END DO
    $\mathbf{x}_k^{(i-1/2)} = \mathbf{A}_{kk}^{-1}\left(\mathbf{b}_k - \mathbf{r}\right)$
  END DO
  DO $k = K, K-1, \ldots, 1$
    $\mathbf{r} = 0$
    DO $j < k$
      $\mathbf{r} = \mathbf{r} + \mathbf{A}_{kj}\mathbf{x}_j^{(i-1/2)}$
    END DO
    DO $j > k$
      $\mathbf{r} = \mathbf{r} + \mathbf{A}_{kj}\mathbf{x}_j^{(i)}$
    END DO
    $\mathbf{x}_k^{(i)} = \mathbf{A}_{kk}^{-1}\left(\mathbf{b}_k - \mathbf{r}\right)$
  END DO
END DO

Figure 4.6. Algorithm of the block symmetric Gauss-Seidel method used for inner iterations

The flux vector $\mathbf{\Phi}^{(n-1)}$ available from the last outer iteration is used as an initial guess for the inner iterations. Because the block diagonal matrix $\mathbf{A}_{kk} \equiv \left[\mathbf{M} - \frac{1}{\lambda'} \mathbf{F} \right]_{kk}$ does not change during outer iterations, the calculation of the block inverse $\mathbf{A}_{kk}^{-1}$ is done at the top nonlinear iteration level.

4.4 Critical Boron Search

Critical boron search is a variant of the eigenvalue problem typically encountered in PWR applications. We need to find a value of the boron concentration, which results in the reactor multiplication factor (eigenvalue) equal to unity. A problem can be written as the nonlinear equation

$$\lambda(c) - 1 = 0 \tag{4.7}$$

where c is the boron concentration; λ is an eigenvalue of the generalized eigenvalue problem (4.1).

The problem is simplified by assumption that the boron concentration is constant over the reactor domain. The eigenvalue λ is practically linear function of the boron concentration and Eq. (4.7) can be easy solved by the Newton method (Ueberhuber, 1997). Eq. (4.7) is converted into standard form of the nonlinear equation as

$$f(x) = 0$$

where $x \equiv c$; $f(x) \equiv \lambda(x) - 1$.

An iteration of the Newton method is written as

$$x^{(m+1)} = x^{(m+1)} - f(x^{(m)}) / f'(x^{(m)}), \quad m = 1, 2, \ldots$$

where m is an nonlinear iteration index; $f'(x)$ is the first derivative of the function $f(x)$.

Because the boron concentration can not be negative, we introduce an additional constraint as

$$x^{(m+1)} = \max\left[x^{(m+1)}, 0 \right]$$

The derivative $f'(x^{(m+1)})$ is computed numerically as

$$f'(x^{(m)}) = \frac{f(x^{(m)}) - f(x^{(m-1)})}{x^{(m)} - x^{(m-1)}}, \quad m = 1, 2, \ldots$$

The first step of the method is different because we do not know the function derivative in advance. At the first step the value of the boron concentration is computed as

$$x^{(1)} = \begin{cases} x^{(0)} + \Delta x^{(0)} & if\ f(x^{(0)}) > 0 \\ \max[x^{(0)} - \Delta x^{(0)}, 0] & if\ f(x^{(0)}) < 0 \end{cases}$$

where $\Delta x^{(0)}$ is the initial change of the boron concentration, the value 200 ppm is used in the code.

The critical boron search is incorporated into the top level of the nonlinear iterations. We start the search if the eigenvalue is computed within a tolerance 0.01. The boron search terminates if the eigenvalue is equal to unity within a given tolerance, the value $10^{-5} = 1\ pcm$ is the default value in the code. Due to a linear dependence of the eigenvalue on the boron concentration, only a few iterations of the Newton method are usually needed.

5. ITERATIVE SOLUTION OF NEUTRON KINETICS PROBLEMS

Equation Section 5

In the Chapter 3, we derived the coarse-mesh finite-difference form of the neutron kinetics equations. Iterative techniques to solve these equations are presented in this Chapter.

5.1 Global Iteration Strategy

The coarse-mesh finite-difference equations (3.7) are written in the matrix form as

$$\mathbf{A}[\Phi(t+\Delta t), \mathbf{T}(t+\Delta t)]\, \Phi(t+\Delta t) = \mathbf{b}, \tag{5.1}$$

where

$\Phi(t+\Delta t) = col\{\, \bar{\Phi}^1(t+\Delta t), \ldots, \bar{\Phi}^K(t+\Delta t)\,\}$ is the vector of the node-averaged neutron flux at the time moment $t+\Delta t$; size of the vector is KG, K is the total number of the nodes, G is the number of neutron energy groups;

$\mathbf{A}[\Phi(t+\Delta t), \mathbf{T}(t+\Delta t)]$ is the $KG \times KG$ matrix, containing the finite-difference entries of the diffusion, removal, scattering and fission operators; the dependency of the matrix $\mathbf{A}$ on the neutron flux Φ and the thermal-hydraulics parameters $\mathbf{T}$ is explicitly stated.

$\mathbf{b}$ is the right hand side vector of size KG.

In the transient calculations, the neutron flux and thermal-hydraulics parameters do not change significantly at the time step. Thus, we linearized Eq. (5.1) using the neutron flux and the thermal-hydraulics parameters from the previous time step to compute the matrix . As a result we do not perform nonlinear iterations during transient calculations, we need only to solve the linear system

$$\mathbf{A}\, \Phi(t+\Delta t) = \mathbf{b}, \tag{5.2}$$

where $\mathbf{A} \equiv \mathbf{A}[\Phi(t), \mathbf{T}(t)]$.

5.2 Basic Iterative Method

The block symmetric Gauss-Seidel (BSGS) method described in the Section 4.3 can be used for solving Eq. (5.2). The i-th iteration of the BSGS method can be written as

$$\mathbf{M}\, \Phi^{(i)} = \left(\mathbf{M} - \mathbf{A}\right) \Phi^{(i-1)} + \mathbf{b}, \quad i = 1, 2 \ldots, \tag{5.3}$$

where $\mathbf{M}$ is called the preconditioner matrix.
In the BSGS method, the matrix $\mathbf{M}$ is given as

$$\mathbf{M} = \left(\mathbf{D}_A - \mathbf{L}_A\right) \mathbf{D}_A^{-1} \left(\mathbf{D}_A - \mathbf{U}_A\right),$$

where

$\mathbf{D}_A$ is the block diagonal part of the matrix $\mathbf{A}$;

$\mathbf{L}_A$ is the block lower triangular part of the matrix $\mathbf{A}$;

$\mathbf{U}_A$ is the block upper triangular part of $\mathbf{A}$.

The iteration (5.3) can be written in the form

$$\boldsymbol{\Phi}^{(i)} = \mathbf{G}\,\boldsymbol{\Phi}^{(i-1)} + \mathbf{f}, \quad i = 1, 2\ldots, \tag{5.4}$$

where

$\mathbf{G} \equiv \mathbf{M}^{-1}\left(\mathbf{M} - \mathbf{A}\right) = \mathbf{I} - \mathbf{M}^{-1}\mathbf{A}$ is called the iteration matrix;

$\mathbf{f} \equiv \mathbf{M}^{-1}\mathbf{b}$.

If a residual of the iterative process (5.4) is defined as

$$\boldsymbol{\delta}^{(i)} = \mathbf{G}\,\boldsymbol{\Phi}^{(i-1)} + \mathbf{f} - \boldsymbol{\Phi}^{(i-1)}, \quad i = 1, 2\ldots$$

the iterations of the BSGS method are given by the algorithm in the Fig. 5.1.

choose initial guess $\boldsymbol{\Phi}^{(0)}$
DO $i = 1, 2, \ldots,$
$\boldsymbol{\delta}^{(i)} = \mathbf{G}\,\boldsymbol{\Phi}^{(i-1)} + \mathbf{f} - \boldsymbol{\Phi}^{(i-1)}$
$\boldsymbol{\Phi}^{(i)} = \boldsymbol{\Phi}^{(i-1)} + \boldsymbol{\delta}^{(i)}$
IF *convergence* EXIT
END DO

Figure 5.1. Algorithm of the basic iterative method

Please, note that the matrix $\mathbf{G}$ is not actually used in the calculations, we simply perform several iterations of the BSGS method using the algorithm given in Fig. 4.6 to compute the approximation $\mathbf{G}\,\boldsymbol{\Phi}^{(i-1)} + \mathbf{f}$.

A convergence rate of the method is determined by the spectral radius (maximum eigenvalue) of the iteration matrix $\mathbf{G}$. In practical LWR calculations, the spectral radius is close to unity and convergence of the BSGS method is slow. The following section describes the acceleration of the basic iterative method using Chebyshev polynomials.

5.3 Adaptive Chebyshev Acceleration Procedure

The basic iterative method given in Fig. 5.1 can be accelerated using the Chebyshev semi-iterative method (Hageman and Young, 1981, Varga, 1962). We make an assumption that the eigenvalues λ_i of the matrix $\mathbf{G}$ are real and can be ordered as

$$m(\mathbf{G}) \equiv \lambda_{KG} \le \cdots \le \lambda_1 \equiv M(\mathbf{G})$$

To apply the Chebyshev polynomials we need the estimates m_E and M_E of the minimum and maximum eigenvalues $m(\mathbf{G})$ and $M(\mathbf{G})$. In the case of the BSGS method, the estimate of the minimum eigenvalue can be taken as zero, the estimate of the maximum eigenvalue is computed using the adaptive procedure comparing the theoretical and observed convergence rates. In the following, we skip theoretical details of the both Chebyshev semi-iterative method and the adaptive procedure; they are given in the book of Hageman and Young (1981). The algorithm taken from the same book is presented in Fig. 5.2; the calculation of the Chebyshev iterative parameters is shown in Fig. 5.3, calculation of the estimate of the maximum eigenvalue is given in Fig. 5.4.

Input :

$\boldsymbol{\Phi}^{(0)}, tol, M_E, m_E, \tau, F, N\_Cheb\_Start, P\_Estimate\_Change$

Initialize :

$n = 0;\ p = 0;\ s = 0;\ M'_E = M_E;\ r\_norm = 1.0;$

DO $n = 1, 2, \ldots,$

$r\_norm\_p = r\_norm;$

IF $n \geq N\_Cheb\_Start$ *THEN*

compute Chebyshev parameters ρ *and* γ :

see Fig. 5.3

$p = p + 1$

ELSE

$\rho = 1.0; \quad \gamma = 1.0$

END IF

calculate new iterate :

$$\boldsymbol{\delta}^{(n)} = \mathbf{G}\,\boldsymbol{\Phi}^{(n-1)} + \mathbf{f} - \boldsymbol{\Phi}^{(n-1)}; \quad r\_norm = \left\| \boldsymbol{\delta}^{(n)} \right\|_2$$

$$\boldsymbol{\Phi}^{(n)} = \rho\left(\gamma\, \delta^{(n-1)} + \boldsymbol{\Phi}^{(n-1)} \right) + (1-\rho)\,\boldsymbol{\Phi}^{(n-2)}; \; f\_norm = \left\| \boldsymbol{\Phi}^{(n)} \right\|_2$$

calculate new estimate M'_E :

see Fig. 5.4

parameter change test :

IF $p \geq P\_Estimate\_Change\ .AND.\ B > (Q)^F$ *THEN*

$p = 0$

END IF

converergence test :

$$\text{IF} \left(\frac{1}{1 - M'_E} \right) \frac{r\_norm}{f\_norm} \leq tol \text{ EXIT}$$

END DO

Figure 5.2 Algorithm of the adaptive Chebyshev acceleration procedure applied to the basic iterative method.

The input quantities used in the Chebyshev acceleration procedure are:

$\boldsymbol{\Phi}^{(0)}$ is the initial guess vector;

tol is tolerance criteria;

$_E$ is the lower bound for $(\mathbf{G})$, the minimum eigenvalue of the iteration matrix $\mathbf{G}$; $m_E = 0$ in the case of the BSGS method;

M_E is the initial estimate for $M(\mathbf{G})$, the maximum eigenvalue of the iteration matrix $\mathbf{G}$; M_E must satisfy $M_E < M(\mathbf{G}) < 1$; M_E 0.8 is the default value in the code;

τ is the upper bound for the maximum eigenvalue estimate; if $M'_E > \tau$ then $M'_E = \tau$; $\tau = 0.995$ is the default value;

F is the damping factor used in the parameter change test; if $F = 0$ the estimate M_E is not changed

and we have nonadaptive procedure; F may be chosen from the interval $[0.65, 0.85]$; $F = 0.8$ is the default value;

N_Cheb_Start is the iteration number when the Chebyshev iterations start; $N\_Cheb\_Start = 5$ is the default value;

$P\_Estimate\_Change$ is the minimum degree of the Chebyshev polynomial when the estimate M_E can be changed; $P\_Estimate\_Change = 5$ is the default value.

The iteration counters used in the algorithm are

n is the current iteration number;

p is the degree of the Chebyshev polynomial;

s is the number of the Chebyshev polynomial sequence.

$$
\begin{aligned}
&IF\ p == 0\ THEN \\
&\quad s = s + 1 \quad < new\ Chebyshev\ polynomial\ sequaence > \\
&\quad IF\ M_E' > \tau\ THEN \\
&\qquad M_E' = \tau \\
&\quad END\ IF \\
&\quad M_E = M_E' \\
&\quad \gamma = \frac{2}{2 - M_E - m_E}; \quad \sigma_E = \frac{M_E - m_E}{2 - M_E - m_E}; \\
&\quad \rho = 1.0; \quad r = \frac{1 - \sqrt{1 - \sigma_E^2}}{1 + \sqrt{1 - \sigma_E^2}} \\
&ELSE\ IF\ p == 1\ THEN \\
&\quad \rho = 1 \Big/ \left(1 - \frac{1}{2} \sigma_E^2 \right) \\
&\text{ELSE} \\
&\quad \rho = 1 \Big/ \left(1 - \frac{1}{4} \sigma_E^2 \rho \right) \\
&\text{END IF}
\end{aligned}
$$

Figure 5.3. Algorithm to compute the Chebyshev iterative parameters ρ and γ

$$
\begin{aligned}
&IF\ \ p == 0\ THEN \\
&\quad r\_norm\_s = r\_norm \\
&END\ IF \\
&IF\ \ \ p > 2\ \ THEN \\
&\quad Q = \frac{2r^{p/2}}{1+r^{p}} \\
&\quad B = r\_norm / r\_norm\_s \\
&\quad IF\ B < 1\ .AND.\ B > Q\ THEN \\
&\qquad X = \left[\tfrac{1}{2}\left(1+r^{p}\right)\left(B+\sqrt{B^{2}-Q^{2}}\right)\right]^{1/p} \\
&\qquad M'_{E} = \frac{1}{2}\left[M_{E} + m_{E} + \left(\frac{2-M_{E}-m_{E}}{1+r}\right)\left(\frac{X^{2}+r}{X}\right)\right] \\
&\quad ELSE \\
&\qquad M'_{E} = M_{E} \\
&\quad END\ IF \\
&END\ IF
\end{aligned}
$$

Figure 5.4 Algorithm to compute the estimation M'_E of the maximum eigenvalue of the iteration matrix

6. MACRO CROSS SECTION MODELS

Equation Section 6

In the neutron diffusion calculations, we assume that the reactor domain is subdivided into the nodes and the macro cross sections are constant over the node. One or four nodes per fuel assembly are typically used in LWR calculations. Based on the material composition the nodes are grouped into the node types and for each type the macro cross sections are computed in advance using the transport theory codes like WIMS, CASMO, HELIOS, etc. The macro cross sections depend on the feedback variables such as burnup, Doppler fuel temperature, coolant void fraction, etc., and the transport calculations are performed to cover a range of the change of the feedback variables. As a result we have a multidimensional table of the macro cross sections depending on the feedback variables. In the neutron diffusion code, the macro cross sections can be computed by interpolation in this table, however due to computer memory limitation, the table dimension is usually reduced approximating the macro cross section by some functions. Macro cross section model of the neutron diffusion code assumes a certain form of the dependencies of the macro cross sections on the feedback variables.

Two macro cross section models are available in the SKETCH-N code. The first model, developed for the calculations of the NEACRP LWR core transient benchmark (Finneman and Galati, 1992) is based on the representation of the macro cross sections as polynomial functions of boron concentration, Doppler fuel temperature, temperature and density of coolant. The model is described in Section 6.1. The second model is developed for the data provided in NEA/NSC Ringhals-1 BWR stability benchmark (Lefvert, 1994). In this model, the macro cross sections are given as a combination of a three-dimensional table with entries for burnup, void and conversion history and additional polynomial dependencies for the Doppler feedback and Xenon poisoning. The model is presented in Section 6.2. The both macro cross section models were developed for the given macro cross section data and they are not general enough to cover all LWR transients. An application of the code to other transients or other reactor types will require a development of a new model.

The macro cross sections depend also on the position of a control rod. The control rods are moving during reactor transients and we have a problem to compute the macro cross sections of the nodes with a partially inserted control rod. Traditional volume-weighting homogenization procedure gives significant errors, known as a rod cusping effect. The effect is especially important in the case of PWR rod ejection and rod withdrawal transients. Flux-weighting homogenization procedure, which significantly reduces these errors, is described in the Section 6.3

6.1 Linear Polynomial Model

Two neutron energy groups are used in the LWR core transient benchmark (Finneman and Galati, 1992). Neutron velocities and energy release per fission are considered to be independent in time and space. Six groups of delayed neutron precursors are applied. Time constants and fractions of the delayed neutrons are constants over the reactor domain.

Within the reactor domain different material compositions and corresponding sets of the macro cross sections are defined. For each material composition the following macro cross sections are given

Σ_{tr1}, Σ_{tr2} are the transport macro cross sections;

Σ_{a1}, Σ_{a2} are the absorption macro cross sections;

Σ_{s21} is the scattering macro cross section from the group 1 into group 2;

Σ_{f1}, Σ_{f2} are the fission macro cross sections;

$\nu\Sigma_{f1}$, $\nu\Sigma_{f2}$ are the multiplication macro cross sections.

The diffusion coefficient is computed in terms of the transport macro cross section as

$$D_g = 1/\left(3\Sigma_{trg}\right), \quad g = 1,2$$

The removal macro cross sections are computed as

$$\Sigma_{r1} = \Sigma_{a1} + \Sigma_{s21}, \quad \Sigma_{r2} = \Sigma_{a2}.$$

The macro cross sections depend on the following feedback variables

c is the borom concentration $[ppm]$;

ρ is the coolant density $[g/cm^3]$;

T_M is the coolant temperature $[°C]$;

T_D is the Doppler fuel temperature $[°K]$.

The Doppler fuel temperature is computed interpolating the fuel rod centerline temperature $T_{F,C}$ and the fuel rod surface temperature $T_{F,S}$ as

$$T_D = (1-\alpha)\,T_{F,C} + \alpha\,T_{F,S},$$

where α is taken equal 0.7.

The macro cross sections are expressed in terms of the feedback variables as

$$\Sigma = \Sigma_0 + \left(\partial\Sigma/\partial c\right)_0 (c - c_0) + \left(\partial\Sigma/\partial\rho\right)_0 (\rho - \rho_0) + \left(\partial\Sigma/\partial T_M\right)_0 (T_M - T_{M0}) +$$

$$\left(\partial\Sigma/\partial\sqrt{T_D}\right)_0 (\sqrt{T_D} - \sqrt{T_{D0}}),$$

where index 0 marks the reference values

The reference macro cross sections, their derivatives and the feedback reference values are given for each material composition.

The macro cross section of the node with a control rod is determined by adding the differential cross section $\Delta\Sigma^{CR}$ to the cross section without control rod as

$$\Sigma^{\text{with CR}} = \Sigma^{\text{without CR}} + \Delta\Sigma^{CR}.$$

The differential cross section $\Delta\Sigma^{CR}$ does not depend on the feedbacks. The contribution of the control rod driver is treated in the same way. The treatment of the node with partially inserted control rod will be given in Section 6.3

6.2 Model for Ringhals-1 BWR Stability Benchmark

Ringhals-1 BWR stability benchmark has been recently performed by Nuclear Science Committee of OECD Nuclear Energy Agency for validation of computer codes used in BWR stability analysis. The data come from measurements performed in Swedish reactor Ringhals-1 designed by ABB Atom and owned and operated by Vattenfall AB. An extensive set of the data describing Ringhals-1 reactor is given in the benchmark specification report (Lefvert, 1994), Ringhals report (Johansson, 1994) and the files on CD-ROM. Two neutron energy groups and six groups of delayed neutrons are used. Within reactor domain different material compositions and the corresponding sets of the macro cross sections are defined. For each material composition the following macro cross sections are given

D_1, D_2 are the diffusion coefficients;

Σ_{a1}, Σ_{a2} are the absorption macro cross sections;

Σ_{s21} is the scattering macro cross section from the group 1 into group 2;

$\nu\Sigma_{f1}$, $\nu\Sigma_{f2}$ are multiplication macro cross sections;

ν is the neutron yield per fission.

For each material composition the macro cross sections are given as three-dimensional tables with entries for burnup, void and conversion history. Conversion history is a collective variable taking into account void history and control rod history. To compute the macro cross section for intermediate values of the void, burnup and conversion history, a three-dimensional tensor-product linear interpolation (Press et. al., 1992; Ueberhuber, 1995) is applied. Let us mathematically define the problem. Feedback variables form a three-dimensional Cartesian grid as $x_l,\ l=1,\ldots,L;\quad y_m,\ m=1,\ldots,M;\quad z_n,\ n=1,\ldots,N$; and the macro cross sections are the functions given at the grid points as $f_{lmn}=f(x_l,y_m,z_n)$. We need to find a value of the function f at the point (x,y,z), where $x\in[x_l,x_{l+1}];\ y\in[y_m,y_{m+1}]$ and $z\in[z_n,z_{n+1}]$. The value $f(x,y,z)$ is computed using the tensor-product linear interpolation (Ueberhuber, 1995), which is defined recursively as

$$f^{1D}(\xi,f_l,f_{l+1})=(1-\xi)f_l+\xi f_{l+1};;$$

$$f^{2D}[\xi,\eta,f_{l,m},f_{l+1,m},f_{l,m+1},f_{l+1,m+1}]=(1-\eta)f^{1D}[\xi,f_{l,m},f_{l+1,m}]+\eta f^{1D}[\xi,f_{l,m+1},f_{l+1,m+1}];$$

$$f^{3D}[\xi,\eta,\zeta,f_{lmn},f_{l+1,m,n},f_{l,m+1,n},f_{l+1,m+1,n},f_{l,m,n+1},f_{l+1,m,n+1},f_{l,m+1,n+1},f_{l+1,m+1,n+1}]=$$

$$(1-\zeta)f^{2D}[\xi,\eta,f_{l,m,n},f_{l+1,m,n},f_{l,m+1,n},f_{l+1,m+1,n}]+$$

$$\zeta f^{2D}[\xi,\eta,f_{l,m,n+1},f_{l+1,m,n+1},f_{l,m+1,n+1},f_{l+1,m+1,n+1}];\qquad(6.1)$$

where

$$\xi=\frac{x-x_l}{x_{l+1}-x_l};\qquad \eta=\frac{y-y_m}{y_{m+1}-y_m};\qquad \zeta=\frac{z-z_n}{z_{n+1}-z_n};$$

and f^{1D},f^{2D},f^{3D} are one-, two-, and three-dimensional linear interpolation, respectively.

The macro cross section of the node with a control rod is determined by adding the differential cross section $\Delta\Sigma^{\mathrm{CR}}$ contributed by a control rod to the cross section without control rod as

$$\Sigma^{\text{with CR}}=\Sigma^{\text{without CR}}+\Delta\Sigma^{\mathrm{CR}}$$

The differential cross sections for absorption cross section $\Delta\Sigma_{a2}^{\mathrm{CR}}$ and for multiplication cross section $\Delta\nu\Sigma_{f2}^{\mathrm{CR}}$ are given also as three-dimensional table with entries for burnup, void and conversion history for each material composition. Their values are computed by the three-dimensional linear interpolation (6.1). The other differential cross sections are constant for each material composition.

The data for the Doppler feedback are given as a change of the infinite multiplication factor as

$$\Delta k_{\infty}^{Doppl}=D(E,\ v)\ \left(\ \sqrt{T_{fuel}}-\sqrt{T_0}\right),\qquad(6.2)$$

$$D(E,\ \mathrm{v}) = (\ 1 + d_1 \times E + d_2 \times E^2 + d_3 \times E^3\ ) \times (\ d_4 + d_5 \times \mathrm{v}\ ),$$

where

Δk_∞^{Doppl} is the change of the infinite multiplication factor due to the Doppler effect;

T_{fuel} is the Doppler fuel temperature;

T_0 is the nominal temperature at which the cross sections have been generated (791 K);

E is the burnup,

v is the instantaneous void,

$d_1 \div d_5$ are the coefficients given for each fuel type.

In the case of two neutron energy groups, the infinite multiplication factor k_∞^0 is computed as

$$k_\infty^0 = \frac{\nu\Sigma_{f1}^0 + \frac{\Sigma_r^0}{\Sigma_{a2}^0}\nu\Sigma_{f2}^0}{\Sigma_r^0 + \Sigma_{a1}^0},$$

where index 0 marks the values computed at the reference Doppler fuel temperature 791 K.
The Doppler effect is incorporated into the fast group absorption macro cross section estimated from the condition to preserve the change of the infinite multiplication factor given by Eq. (6.2). The fast group absorption macro cross section is computed as

$$\Sigma_{a1}^{Doppl} = \frac{\nu\Sigma_{f1}^0 + \frac{\Sigma_r^0}{\Sigma_{a2}^0}\nu\Sigma_{f2}^0}{\Delta k_\infty^{Doppl} + k_\infty^0} - \Sigma_r^0 .$$

In the benchmark, the microscopic xenon thermal absorption cross section σ_{a2}^{Xe} is specified as

$$\sigma_{a2}^{Xe} = (\ s_1 + s_2 \times E\ ) \times (\ s_3 + s_4 \times \mathrm{v} + \ s_4 \times \mathrm{v}^2\ ),$$

where $s_1 \div s_5$ are the coefficient given for each material composition.
In the code, Xenon poisoning is taking into account as

$$\Sigma_{a2}^{Xe} = \Sigma_{a2}^0 + \sigma_{a2}^{Xe} \times (\ Xe - Xe^0\ ),$$

where index 0 marks macro cross sections computed with the nominal xenon concentration Xe^0.
The reference xenon concentration Xe^0 is not specified in benchmark data. The only reference about its value is given in (Johansson, 1994), where is written *"Absorption cross section for thermal neutrons SIGA2 include xenon corrections (i.e. they are calculated assuming equilibrium xenon at full power)."*
We compute approximately an equilibrium xenon concentration as follows

$$Xe^0 = \frac{(\gamma_I + \gamma_{Xe})\ \ \Sigma_f\ \ \Phi_2^{\mathrm{Re}f}}{\lambda_{Xe} + \sigma_{Xe}\ \ \Phi_2^{\mathrm{Re}f}},$$

where

γ_I is the effective yield of iodine (0.064 atoms/fission);

γ_{Xe} is the effective yield of xenon (0.003 atoms/fission);

λ_{Xe} is the xenon decay constant (2.09E-05 s^{-1});

σ_{Xe} is the microscopic xenon absorption cross section;

Φ_2^{Ref} is the reference thermal flux;

Σ_f is the effective fission cross section computed as $\Sigma_f = \Sigma_{f2} + \Sigma_{f1}\Phi_1 / \Phi_2 \approx \Sigma_{f2} + \Sigma_{f1}\Sigma_{a2} / \Sigma_R$.

Xenon constants are taken from (Cameron, 1982). The reference thermal flux for the specified reactor power is computed as

$$\Phi_2^{Ref} = \frac{P_{NOM}}{V_{core} \ \varepsilon \ \Sigma_f},$$

where

P_{NOM} is the nominal reactor power (2270 MWt);

V_{core} is the volume of the reactor core (5.5640E+7 cm^3);

ε is the conversion factor (3.237E-11 W/fission).

6.3 General Polynomial Model

In this section, we present the general macro cross section model, which can be used for the both PWR and BWR. Theory of this approach is described in detail in the separate report (Zimin and Semenov, 2002), here we only outline the basic ideas.

Let us consider as an example fuel assembly macro cross sections, which depend on burnup B, coolant and fuel temperature T_c, T_f, coolant density ρ_c and boron concentration c_{bor}. As a result of the fuel assembly lattice calculations we have two sets of the cross sections:

- the base set $\Sigma^{base}(B_n)$, $n = 1,\ldots,N_{burnup}$, which is computed under nominal fuel assembly conditions and depends only on burnup;
- the branch set $\Sigma^{dev}(B_n, T_f^{nj}, \rho_c^{nj}, T_c^{nj}, c_{bor}^{nj})$, $n = 1,\ldots,N_{burnup}$; $j = 1,\ldots,J_{dev}$, which is generated in the branch calculations and depends on burnup and the thermal-hydraulics fuel assembly parameters.

A one-dimensional dependence of the base cross sections on burnup is interpolated using the cubic spline. The error of this interpolation procedure is usually very small. In the test VVER calculations, the error does not exceed 0.01 % (Zimin and Semenov, 2002).

To decrease the dependence of the branch cross sections on burnup, we form the set of the deviations

$$\Delta\Sigma^{dev} = \begin{cases} \Sigma^{dev}(B_n, T_f^{nj}, \rho_c^{nj}, T_c^{nj}, c_{bor}^{nj}) - \Sigma^{base}(B_n), \\ \qquad n = 1,\ldots,N_{burnup},\ j = 1,\ldots,J_{dev}; \\ 0, \quad n = 1,\ldots,N_{burnup}; \end{cases}$$

The deviations $\Delta\Sigma^{dev}$ are approximated using multi-dimensional polynomial by the least-square method. Construction of a form of the approximating polynomial is performed automatically using the best-fitting selection of the polynomial terms by the stepwise selection algorithm. A number of the polynomial terms is minimal to satisfy a user given accuracy of the approximation.

The selection of the best-fitting approximating polynomial and the spline interpolation of the initial cross section set is performed by the external module. The prepared neutron cross section library is written into the file, which is the input file of the SKETCH-N code. As a result, in the SKETCH-N code the cross sections are computed as

$$\Sigma(B, T_f, \rho_c, T_c, c_{bor}) = \Sigma^{spline}(B) + \Delta\Sigma^{polynom}(B, T_f, \rho_c, T_c, c_{bor}),$$

where $\Sigma^{spline}(B)$ is the cubic spline and $\Delta\Sigma^{polynom}$ is the selected multi-dimensional polynom.

The described neutron cross section model may be used for different reactor types. The main advantage of this model that the constructed polinomial provides the user given accuracy of the approximation. The model is also fast running in a comparison with the traditional table interopolation. For example, for performed VVER calculations, to approximate the two-group neutron cross sections within 0.05 % five-dimensional polynomial contains only about 20 terms for the fast group cross sections and about 50 terms for the cross sections of the thermal group.

6.4 Treatment of the Node with Partially Inserted Control Rod

In the nodal calculations, we assume that the macro cross sections are constant over the node and two sets of the data with and without control rods are usually given. In the actual calculations, the control rod can be in an intermediate position inside a node, thus introducing a problem of the macro cross section homogenization. The simplest method is a volume-weighting homogenization procedure given by the formula

$$\Sigma^{l,m,n} = (1-\chi)\,\Sigma^{\text{without CR}} + \chi\,\Sigma^{\text{with CR}}, \tag{6.3}$$

where

χ is the weighting factor computed using the volume-weighting as $\chi = \dfrac{h_n^{\text{with CR}}}{h_n^{\text{with CR}} + h_n^{\text{without CR}}}$;

$h_n^{\text{with CR}}$ is the height of the rodded part of the node;

$h_n^{\text{without CR}}$ is the height of the unrodded part of the node.

The neutron flux changes sharply at the interface of the rodded and unrodded parts as illustrated in Fig. 6.1.

As a result, the volume-weighting homogenization procedure (6.3) introduces significant errors in PWR calculations, known as a rod cusping effect. The term “rod cusping” is due to the cusp-like behavior of the node-averaged flux versus time when the control rod moves through the node. These errors can be reduced applying the flux-volume homogenization. The macro cross sections are computed using the same formula (6.3), but the weighting factor χ is calculated using flux-weighting i.e.

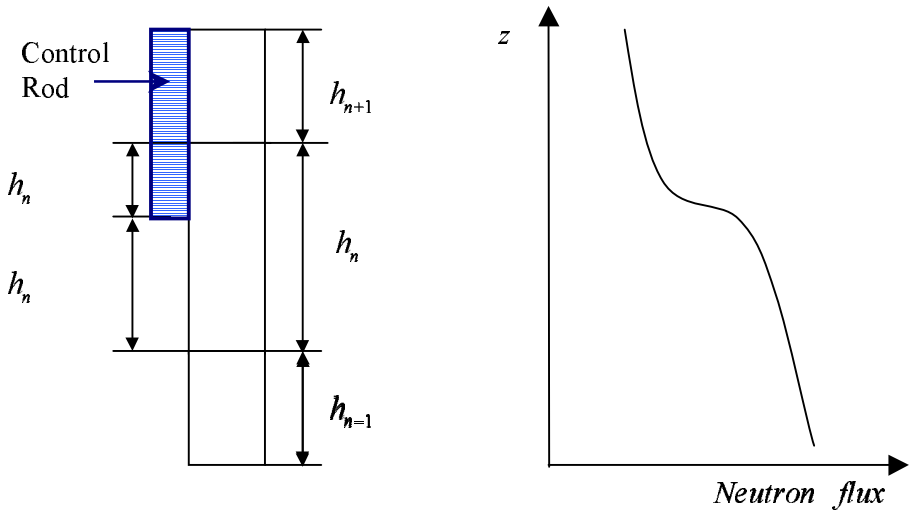

Fig. 6.1. Illustration of the rod cusping effect.

$$\chi_g = \frac{\Phi_g^{\text{with CR}}\, h_n^{\text{with CR}}}{\Phi_g^{\text{with CR}} h_n^{\text{with CR}} + \Phi_g^{\text{without CR}} h_n^{\text{without CR}}}$$

where

g is an index of neutron energy group;

$\Phi_g^{\text{with CR}}$ is the average flux of the rodded part of the node;

$\Phi_g^{\text{without CR}}$ is the average flux of the unrodded part of the node.

These two values $\Phi_g^{\text{with CR}}$ and $\Phi_g^{\text{with t CR}}$ are not known from the results of the nodal calculations. In the case of a strongly absorbing control rod, which is the case when the rod cusping effect is significant, the neutron flux changes sharply at the rod tip and varies slowly away from the tip as is shown in Fig. 6.1. Based on this observation a simple approximating procedure to compute $\Phi_g^{\text{with CR}}$ and $\Phi_g^{\text{with t CR}}$ has been proposed (Gehin, 1992). The flux in the unrodded part of the node is calculated by averaging the fluxes of the partially rodded node and its lower neighbor as

$$\Phi_g^{\text{without CR}} = \xi\, \Phi_g^{l,m,n} + (1-\xi)\, \Phi_g^{l,m,n-1},$$

where $\xi = \dfrac{h_n^{\text{without CR}}}{h_n^{\text{without CR}} + h_{n-1}}$.

Likewise, the average flux in the rodded part of the node is approximated by averaging the fluxes in partially rodded node and its upper neighbor as

$$\Phi_g^{\text{with CR}} = \eta\, \Phi_g^{l,m,n} + (1-\eta)\, \Phi_g^{l,m,n+1},$$

where $\eta = \dfrac{h_n^{\text{with CR}}}{h_n^{\text{with CR}} + h_{n+1}}$.

This approach works well in the performed PWR rod ejection calculations, the rod cusping errors are small even in the case of the coarse axial mesh size of 30 cm (Zimin et. al., 1999). However, in the case of the PWR rod withdrawal transients, when several control rods are moving, a fine axial mesh is needed to get accurate results (Asaka et. al., 2000). More elaborate approach should be developed if a code user wants to perform general PWR calculations with a coarse axial mesh.

7. INTERNAL THERMAL-HYDRAULICS MODEL

Equation Section 7

Chapter 7 describes an internal thermal-hydraulics model of the SKETCH-N code. The model is developed for PWR calculations with single-phase coolant flow. Heat conduction equations in fuel rods are discretized in time and space using the finite-difference method. Resulting three-diagonal system of equations is solved using LU decomposition. Heat conduction is considered only in radial direction. Material properties are taken from the NEACRP LWR core transient benchmark specification (Finnemann and Galati, 1992). Fluid dynamics is modeled under single-phase flow conditions. The mass flow rate in each coolant channel is assumed to be known and specified by a code user. As a result we need to solve only mass continuity and energy conservation equations. They are discretized in space using the finite-difference method and in time by the semi-implicit scheme. The heat conduction model has been verified by a comparison with an analytical solution available for steady-state conditions. The thermal-hydraulics model has been applied for the NEACRP PWR rod ejection benchmark (Finneman and Galati, 1992) and demonstrated accurate results for fuel and coolant temperatures.

7.1 Heat Conduction in the Fuel Rod

The heat conduction equation in the fuel rod is written as

$$\rho(r,t)\ C(r,t)\frac{\partial T(r,t)}{\partial t}=\frac{1}{r}\frac{\partial}{\partial r}r\ k(r,t)\ \frac{\partial}{\partial r}\ T(r,t)\ +\ q(r,t) \tag{7.1}$$

where

$T(r,t)$ is the temperature at the point r along the fuel pin radius [°K];

$k(r,t)$ is the thermal conductivity [watts/m °K];

$q(r,t)$ is the volumetric power generation rate [watts/m^3];

$C(r,t)$ is the specific heat of the fuel pin material [J/kg °K];

$\rho(r,t)$ is the fuel pin density [kg/m^3].

Eq. (7.1) is completed by the boundary conditions:

$$-k(r,t)\frac{\partial}{\partial r}\ T(r,t)\Big|_{r=0}=0 \tag{7.2}$$

$$-k(r,t)\ \frac{\partial}{\partial r}T(r,t)\Big|_{r=R_F}=q''(R_F,t)=\frac{R_{CL}}{R_F}q''(R_{CL}^{in},t) \tag{7.3}$$

$$-k(r,t)\frac{\partial}{\partial r}\ T(r,t)\Big|_{r=R_{CL}^{in}}=q''(R_{CL}^{in},t)=H_g\ \left[\ T(R_F,t)-T(R_{CL}^{in},t)\ \right] \tag{7.4}$$

$$-k(r,t)\frac{\partial}{\partial r}\ T(r,t)\Big|_{r=R_{CL}^{out}}=q''(R_{CL}^{out},t)=h\ \left[\ T(R_{cl}^{out},t)-T_L(t)\ \right] \tag{7.5}$$

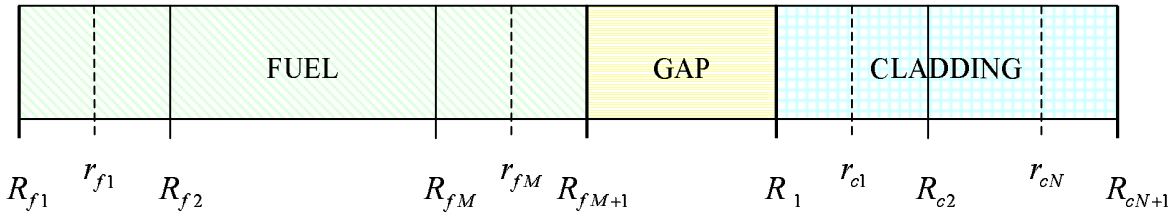

Fig. 7.1. Radial spatial mesh in fuel rods

where

R_F is the radius of the fuel pellet [m];

R_{CL}^{in} is the inner radius of the cladding [m];

R_{CL}^{out} is the outer radius of the cladding [m];

H_g is the gap conductance [W/m^2 °K];

$q''(R_{CL}^{in},t)$ is the heat flux at the inner surface of the cladding [W/m^2];

$q''(R_F,t)$ is the heat flux at the outer surface of the fuel pellet [W/m^2];

$T_L(t)$ - is the bulk coolant temperature [°K].

The heat conduction Eq. (7.1) is discretized at the spatial mesh given in Fig. 7.1, where the solid lines show locations of the mesh points and the dashed lines indicate control volumes. The temperatures are computed at the points R_{fm}, $m=1,\ldots,M+1$ and R_{cn}, $m=1,\ldots,M+1$. The control volumes are taken as the intervals $[r_{fm-1},r_{fm}]$ and $[r_{cm-1},r_{cm}]$ for fuel and cladding, respectively. The derivation of the heat conduction equations is performed in the way as described in (Patankar, 1980). Let as consider the inner spatial region $r_{m-1} \le r \le r_m$. Eq. (7.1) is multiplied by r and integrated over the interval $[r_{m-1},\ r_m]$. The result is as follows

$$\left[\frac{r_m^2-r_{m-1}^2}{2}\right]\ \rho_m(t)\ C_m(t)\frac{\partial T_m(t)}{\partial t} = r_m k(r_m,t)\frac{\partial}{\partial r}\ T(r,t)\Big|_{r=r_m} -$$

$$r_{m-1}k(r_{m-1},t)\frac{\partial}{\partial r}\ T(r,t)\Big|_{r=r_{m-1}} +\ q_m(t)\ \left[\frac{r_m^2-r_{m-1}^2}{2}\right]$$

where

$q_m(t)$ is the average volumetric power generation rate at the interval $[r_{m-1},r_m]$,

$T_i(t)$ is the average temperature at the interval $[r_{m-1},r_m]$.

The heat flux at the interface between the nodes m and $m+1$ is expressed as follows

$$k(r_m,t)\ \frac{\partial}{\partial r}\ T(r,t)\Big|_{r=r_m} = \frac{2k_m(t)\ k_{m+1}(t)}{k_m(t)+k_{m+1}(t)}\ \frac{\left(T_{m+1}(t)-T_m(t)\right)}{\Delta R_m}$$

where $\Delta R_m = R_{m+1} - R_m$.
The resulting equation for the internal node is

$$\left[\frac{r_m^2 - r_{m-1}^2}{2}\right]\ \rho_m(t)\ C_m(t)\frac{\partial T_m(t)}{\partial t} = r_m\ \frac{2k_m(t)\ k_{m+1}(t)}{\Delta R_m\left(k_m(t)+k_{m+1}(t)\right)}\left(T_{m+1}(t)-T_m(t)\right)+$$

$$r_{m-1}\ \frac{2k_m(t)\ k_{m-1}(t)}{\Delta R_{m-1}\left(k_m(t)+k_{m-1}(t)\right)}\left(T_{m-1}(t)-T_m(t)\right)+\left[\frac{r_m^2 - r_{m-1}^2}{2}\right]\times q_m(t) \qquad (7.6)$$

Using the boundary conditions (7.2)-(7.5) for the heat flux, the equations are also discretized for the boundary mesh points. The results are as follows:

centerline fuel temperature:

$$\left[\frac{r_{f1}^2}{2}\right]\ \rho_{f1}(t)\ C_{f1}(t)\frac{\partial T_{f1}(t)}{\partial t} = r_{f1}\ \frac{2k_{f2}(t)\ k_{f1}(t)}{\Delta R_{f1}\left(k_{f1}(t)+k_{f2}(t)\right)}\left(T_{f1}(t)-T_{f1}(t)\right)+\left[\frac{r_{f1}^2}{2}\right]\times q_1(t) \qquad (7.7)$$

surface fuel temperature:

$$\left[\frac{R_{f\,M+1}^2 - r_{f\,M}^2}{2}\right]\ \rho_{f\,M+1}(t)\ C_{f\,M+1}(t)\frac{\partial T_{f\,M+1}(t)}{\partial t} = R_{c1}\ H_g\left(\ T_{c1}(t)-T_{f\,M+1}(t)\ \right)+$$

$$r_{f\,M}\ \frac{2k_{f\,M}(t)\ k_{f\,M+1}(t)}{\Delta R_{f\,M}\left(k_{f\,M}(t)+k_{f\,M+1}(t)\right)}\left(T_{f\,M}(t)-T_{f\,M+1}(t)\right)+\left[\frac{R_{f\,M+1}^2 - r_{f\,M}^2}{2}\right]\times q_{M+1}(t) \qquad (7.8)$$

inner surface cladding temperature:

$$\left[\frac{r_{c2}^2 - R_{c1}^2}{2}\right]\ \rho_{c1}(t)\ C_{c1}(t)\frac{\partial T_{c1}(t)}{\partial t} = r_{c1}\ \frac{2k_{c1}(t)\ k_{c2}(t)}{\Delta R_{c1}\left(k_{c1}(t)+k_{c2}(t)\right)}\left(T_{c2}(t)-T_{c1}(t)\right)+$$

$$R_{c1}\ H_g\left(\ T_{f\,M+1}(t)-T_{c1}(t)\right) \qquad (7.9)$$

outer surface cladding temperature:

$$\left[\frac{R_{c\,N+1}^2 - r_{c\,N}^2}{2}\right]\ \rho_{c\,N+1}(t)\ C_{c\,N+1}(t)\frac{\partial T_{c\,N+1}(t)}{\partial t} = R_{c\,N+1}\ h\ \left(T_{bulk}(t)-T_{cN+1}(t)\right)+$$

$$r_{cN} \quad \frac{2k_{cN+1}(t)\ \ k_{cN}(t)}{\Delta R_{cN}\left(k_{cN+1}(t)+k_{cN}(t)\right)}\left(T_{cN}(t)-T_{cN+1}(t)\right) \tag{7.10}$$

The equations (7.6)-(7.10) are discretized in time using a fully-implicit scheme. The result for the internal mesh point are written as

$$a_m T_{m-1}(t+\Delta t)+b_m T_m(t+\Delta t)+c_m T_{m+1}(t+\Delta t)=s_m(t+\Delta t), \tag{7.11}$$

where

$$a_m=-r_{m-1}\ \frac{2k_m(t+\Delta t)\ \ k_{m-1}(t+\Delta t)}{\Delta R_{m-1}\left(k_m(t+\Delta t)+k_{m-1}(t+\Delta t)\right)};$$

$$c_m=-r_m\ \frac{2k_m(t+\Delta t)\ \ k_{m+1}(t+\Delta t)}{\Delta R_m\left(k_m(t+\Delta t)+k_{m+1}(t+\Delta t)\right)};$$

$$b_m=-a_m-b_m+\left[\frac{r_m^2-r_{m-1}^2}{2}\right]\ \frac{\rho_m(t+\Delta t)\ \ C_m(t+\Delta t)}{\Delta t};$$

$$s_m(t+\Delta t)=\left[\frac{r_m^2-r_{m-1}^2}{2}\right]\left[\frac{\rho_m(t+\Delta t)\ \ C_m(t+\Delta t)}{\Delta t}\ \ T_m(t)+q_m(t+\Delta t)\right].$$

The time-discretized equations for the boundary nodes are written in the same way. The nonlinear equations (7.11) are linearized taking the values of thermal conductivity k, density ρ and specific heat C from the previous time step. The resulting three-diagonal system of linear equations is solved using LU decomposition (Ueberhuber, 1995). Setting the term $1/\Delta t$ to zero, the steady-state form of the heat conduction equations is obtained. The same solution procedure is used in this case. To verify the computer code, the steady-state calculations were compared with the analytical solution, which can be derived in the case of constant material properties and flat power distribution (Lahey and Moody, 1977).

7.2 Fluid Dynamics Equations of the Coolant

The equations for the coolant temperature can be written in the following form (Shober et. al., 1977):

$$\rho_L(z,t)\ \ C_L(z,t)\ \frac{\partial T_L(z,t)}{\partial t}+G_L(z,t)\ \ C_L(z,t)\ \frac{\partial T_L(z,t)}{\partial z}=\gamma\ \ q_L''(z,t)+q_L'''(z,t), \tag{7.12}$$

where

$T_L(z,t)$ is the bulk coolant temperature [°K];

$G_L(z,t)$ is the coolant mass flow rate [kg/m^2 s];

$q_L'''(z,t)$ is the volumetric power generation rate at the coolant [watts/m^3];

$C_L(z,t)$ is the coolant specific heat [J/kg °K];

$\rho_L(z,t)$ is the coolant density [kg/m^3];

γ is the ratio of the surface area of the cladding to the volume of the coolant [m^{-1}];

$q''_L(z,t)$ is the heat flux at the surface of the cladding [watts/m^2].

The equations are discretized at the axial mesh, which coincides with the neutronics mesh. The z-derivative is approximated as

$$\frac{\partial T_L(z,t)}{\partial z} = \frac{2\left[\ T_{Li}(t) - T_{Li}^{in}(t)\ \right]}{\Delta z_i},$$

where

$T_{Li}(t)$ is the average coolant temperature of the axial node i;

$T_{Li}^{in}(t)$ is the inlet coolant temperature of the node i;

Δz_i is the axial height of the node i.

The inlet temperature for the node i is computed using the linear interpolation from the previous node $i-1$ as

$$T_{Li}^{in}(t) = 2T_{Li-1}(t) - T_{Li-1}^{in}(t) \tag{7.13}$$

The resulting equation for the coolant temperature is:

$$\rho_{Li}(t)\ C_{Li}(t)\ \frac{\partial T_{Li}(t)}{\partial t} + G_{Li}(t)\ C_{Li}(t)\ \frac{2\left[\ T_{Li}(t) - T_{Li}^{in}(t)\ \right]}{\Delta z_i} = \gamma\ \psi_i(t) + q_{Li}(t),$$

Using the fully-implicit scheme, the equation (7.13) is written as

$$T_{Li}(t+\Delta t)\left\{\ \rho_{Li}(t+\Delta t)\ C_{Li}(t+\Delta t)\ \frac{1}{\Delta t} + G_{Li}(t+\Delta t)\ C_{Li}(t+\Delta t)\ \frac{2}{\Delta z_i}\right\} =$$

$$\gamma\ \psi_i(t+\Delta t) + q_{Li}(t+\Delta t) + T_{Li}(t)\left\{\ \rho_{Li}(t+\Delta t)\ C_{Li}(t+\Delta t)\ \frac{1}{\Delta t}\right\} +$$

$$T_{Li}^{in}(t+\Delta t)\ G_{Li}(t)\ C_{Li}(t)\ \frac{2}{\Delta z_i} \tag{7.14}$$

The nonlinear equation (7.14) is linearized using the values of the coolant density ρ_{Li} and the coolant specific heat C_{Li} from the previous time step. The mass flow rate C_{Li} is constant for all axial nodes. The heat flux at the cladding surface is also approximated using the coolant and cladding temperatures from the previous time step

$$q''_{Li}(t+\Delta t) \approx h_{Li}(t)\ \left[\ T(R_{cl}^{out},t) - T_{Li}(t)\ \right], \tag{7.15}$$

where the heat transfer coefficient $h_i(t)$ is computed using the inlet coolant temperature $T_{Li}^{in}(t+\Delta t)$.

To compute the heat transfer coefficient the Dittus-Boelter correlation (Wakil, 1993) is applied:

$$h_{L\,i} = 0.023\ \ \mathrm{Re}^{0.8}\ \ \mathrm{Pr}^{0.4}\,\frac{k_L}{D_e} \tag{7.16}$$

where

k_L is the thermal conductivity of the coolant [watts/m °K];

$\mathbf{Re}$ is the Reynolds number;

$\mathbf{Pr}$ is the Prandtl number;

D_e is the equivalent hydraulic diameter.

The Prandtl and Reynolds number are defined as

$$\mathrm{Re} = \frac{D_e G}{\mu};\quad \mathrm{Pr} = \frac{C_p \mu}{k_L};$$

where

G is the mass flow rate of the coolant [kg/m^2 s];

μ is the viscosity of the coolant [kg/m s].

Final algorithm of the solution of the both heat conduction and fluid dynamics equations is summarized in the Fig. 7.2, where $N\_Channel$ is a number of the coolant channels and NZ is a number of axial layers. Two water packages are available in the SKETCH-N code. The one is adapted from the J-TRAC code (Akimoto, 1989), and the other is taken from the JAERI subroutine library JSSL (Inoue, 1982). The steady-state calculations are performed in the same way, setting the term $1/\Delta t$ to zero.

```
DO  k = 1,..., N_Channel
  DO  n = 1,..., NZ
    compute inlet coolant temperature of the node by Eq. (7.13)
    compute heat transfer coefficient by Eq. (7.16)
    compute the heat flux by Eq. (7.15)
      using temperatures from the previous time step
    solve Eq. (7.14) for the average coolant temperature
    compute coolant properties using water property package
    recalculate the heat transfer coefficient by Eq. (7.16)
    recalculate the heat flux by Eq. (7.15)
    solve heat conduction Eq. (7.11) in the fuel rod
  END DO
END DO
```

Fig. 7.2. Algorithm of the time step calculation by the thermal-hydraulics model.

8. COUPLING INTERFACE MODULE

Equation Section 8

The internal thermal-hydraulics model described in the precious chapter 7 can treat only single-phase coolant flow, thus limiting applications of the code to operational transients in PWR. To use the SKETCH-N code in LWR safety analysis, a coupling with an external thermal-hydraulics code is needed. To simplify the coupling procedure, an interface module has been developed. The module provides data exchange between the coupled codes, data mapping between spatial meshes of the codes and synchronization of time stepping. The module has been applied for a coupling of the SKETCH-N code with the transient analysis codes J-TRAC and TRAC-BF1 (Zimin et.al., 1999; Asaka et. al., 2000; Zimin et. al., 2000). The description of the interface module is given in this chapter.

8.1 Time Stepping and Data Exchange

The coupling and data transfer between the codes is organized using the message-passing library Parallel Virtual Machine (PVM) (Al Geist et al., 1994). In the following, we assume that an external thermal-hydraulics model is the transient analysis code TRAC, because a coupling with the J-TRAC code (Akimoto, 1989), which is a JAERI version of the TRAC-PF1/MOD1, and with the TRAC-BF1 code (Borkowski et al., 1992) has been already performed (Zimin et.al., 1999; Asaka et. al., 2000; Zimin et. al., 2000). The codes are treated as separate processes and the subroutines based on PVM are responsible for data exchange between the codes and synchronization of the time stepping. A flow chart of the coupled TRAC/SKETCH-N calculations is illustrated in Fig. 8.1.

After performing the input, the TRAC code enrolls into PVM and spawns the child process - SKETCH-N. When SKETCH-N is started, it gets an identification number of the parent process and the codes can communicate to each other sending/receiving messages. At the beginner of a time step, TRAC sends a message to SKETCH-N with thermal-hydraulics reactor data (fuel temperature, coolant density and coolant temperature) and an estimation of the next time step size. SKETCH-N receives the message, selects a new time step size and performs the neutronics calculation. Then, SKETCH-N sends a message to TRAC with the computed power distribution and the used time step size. TRAC receives the message and performs the thermal-hydraulics calculation. The procedure is repeated till the end of the transient.

Actually a time stepping procedure is slightly more complicated as illustrated in Fig 8.2 due to an automatic time step control based on the time step doubling technique. SKETCH performs the calculations using two temporal meshes: the fine temporal mesh, which consists of pairs of the time steps of equal length, and the coarse temporal mesh with a double time step size. A comparison of the power distribution computed on these meshes gives an estimate of the SKETCH time step size. Finally, a time step size of the TRAC/SKETCH code system is selected as a minimum of the time step sizes proposed by the codes.

The list of the variables, which transfer between the meshes, is given as follows

from the neutronics mesh into the heat conduction mesh:

heat generation rate of the heat conduction nodes [W];

from the neutronics mesh into the fluid dynamics mesh:

heat generation rate of the fluid dynamics nodes due to the direct coolant heating [W];

from the heat conduction mesh into the neutronics mesh:

average Doppler fuel temperature of the neutronics nodes [°K];

from the fluid dynamics mesh into the neutronics mesh:

average coolant temperature [°K] and coolant density [kg/m^3] of the neutronics nodes.

Variables transferring from the TRAC code depend on the macro cross section model of the SKETCH and can be changed for a new reactor type or a new problem.

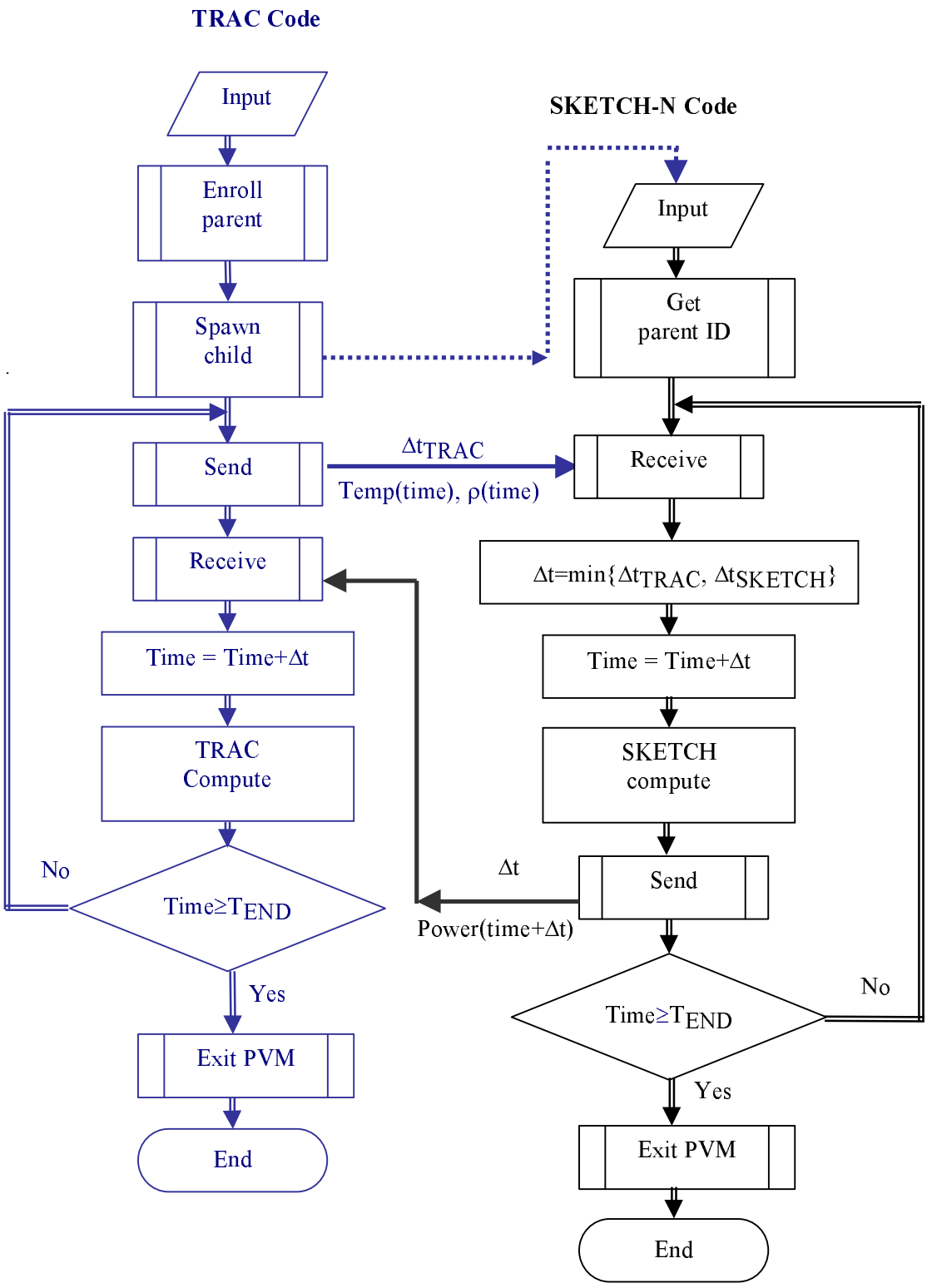

Figure 8.1. Simplified flow chart of the TRAC/SKETCH-N calculations

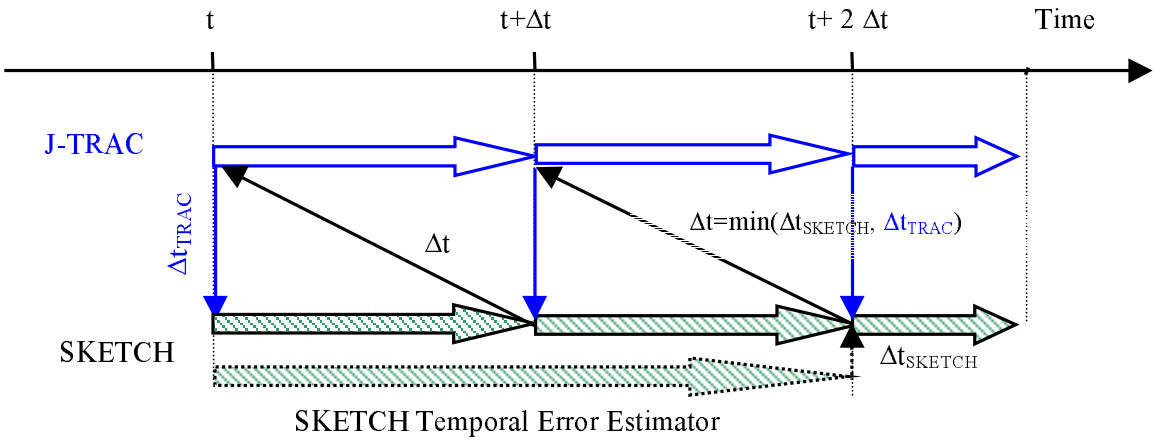

Figure 8.2. Time Stepping of the Coupled TRAC/SKETCH code system

8.2 Mapping of the Data between Spatial Meshes of the Codes

TRAC and SKETCH-N codes usually perform calculations using different spatial meshes. Additionally TRAC code applies axially staggered meshes for the fluid dynamics and heat conduction models. An example of the axial meshes used in the codes is given in Fig. 8.3. The coupled TRAC/SKETCH-N calculations require a mapping of the data between the neutronics, heat conduction and fluid dynamics meshes.

The mapping procedure between the meshes is based on the mapping matrix approach developed for the PARCS code (Downar, Barber et. al, 1997). Let us consider the mapping of the heat generation rate computed by the SKETCH code on the neutronics spatial mesh into the heat conduction mesh and into the fluid dynamics mesh. The mapping is performed using the two mapping matrices: S_{NEUT}^{HC} and S_{NEUT}^{FD}.

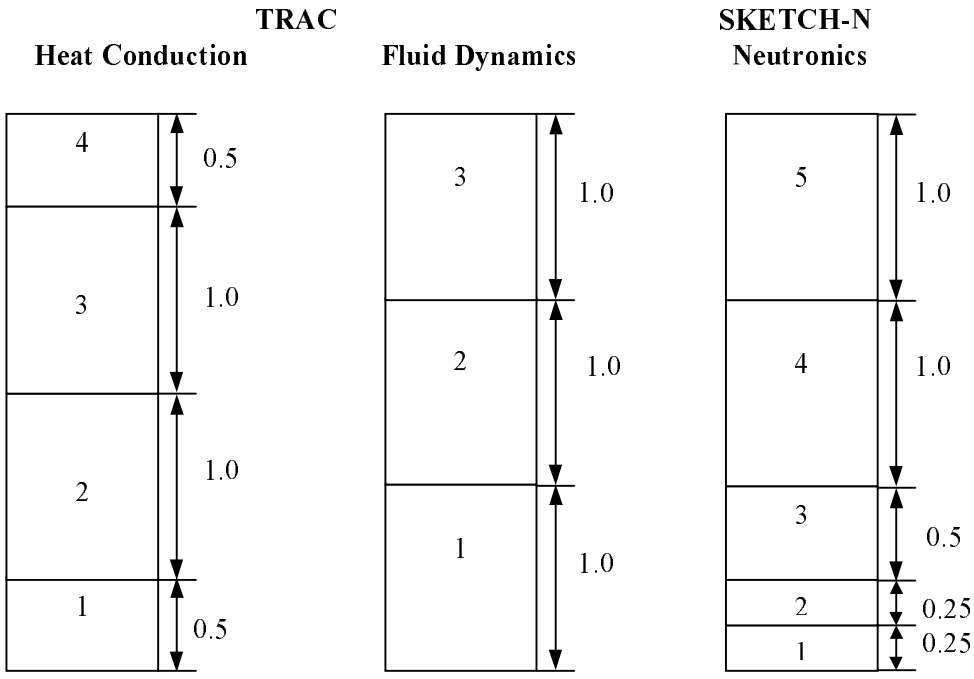

Figure 8.3. An example of the axial meshes in the TRAC and SKETCH-N codes

Even the spatial meshes of SKETCH-N and TRAC codes can be very different, they share a one common feature: the 3D spatial mesh of the both codes is a tensor product of the 2D spatial mesh in radial plane and the 1D axial mesh. As a result the 3D mapping matrix can be also expressed as a tensor product of the 2D and 1D mapping matrices:

$$S = S2\mathrm{D} \otimes S1\mathrm{D} \tag{8.1}$$

where

$\otimes$ is a direct (Kronecker) matrix product;

$S2\mathrm{D}$ is the mapping matrix describing a correspondence of 2D radial meshes;

$S1\mathrm{D}$ is the mapping matrix for axial meshes.

These mapping matrices S D and $S1\mathrm{D}$ are defined by a code user for two pairs of the spatial meshes: neutronics mesh $\Rightarrow$ heat conduction mesh and neutronics mesh $\Rightarrow$ fluid dynamics mesh. Let us consider the calculation of the mapping matrix for two 1D axial meshes: the neutronics mesh z with J nodes, j-th node is the interval $[z_j, z_{j+1}]$ and the heat conduction mesh Z with I nodes, i-th node is the interval $[Z_i, Z_{i+1}]$. The mapping matrix 1D for these two meshes is computed as

$$S1\mathrm{D}_{ij} = \frac{\left| [\ z_j\ , z_{j+1}\ ] \cap [\ Z_i\ , Z_{i+1}\ ] \right|}{\left| [\ z_j\ , z_{j+1}\ ] \right|}, \tag{8.2}$$

where

$[\ z_j\ , z_{j+1}\ ] \cap [\ Z_i\ , Z_{i+1}\ ]$ is an intersection of the intervals $[\ z_j\ , z_{j+1}\ ]$ and $[\ Z_i\ , Z_{i+1}\ ]$;

$|\circ|$ is the function equals to the length of the interval.

For example, the mapping matrices for the axial meshes defined in Fig. 2 are given as

$$S1\mathrm{D}^{HC}_{NEUT} = \begin{pmatrix} 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0.5 & 0 \\ 0 & 0 & 0 & 0.5 & 0.5 \\ 0 & 0 & 0 & 0 & 0.5 \end{pmatrix}; \quad S1\mathrm{D}^{FD}_{NEUT} = \begin{pmatrix} 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix},$$

where

$S1\mathrm{D}^{HC}_{NEUT}$ is the mapping matrix from the neutronics mesh into the heat conduction mesh;

$S1\mathrm{D}^{FD}_{NEUT}$ is the mapping matrix from the neutronics mesh into the fluid dynamics mesh.

The calculation of the mapping matrices for 2D spatial meshes is very similar, the only difference that the area of the nodes is taken instead of the length of the nodes. Because the mapping matrices are sparse, the compressed row storage (CRS) format is used for their input and storage (Barrett et al., 1994). According to the definition, see Eq. (8.2), the mapping matrices satisfy the following condition

$$\sum_{i=1}^{I} S_{ij} = 1$$

This condition providing the conservation law is checked in the code when an input of the mapping matrices is performed.

Using the defined mapping matrices the heat generation rate on the heat conduction mesh $P\ [HC]$ is computed as

$$P[HC] = (1-\alpha)\, S^{HC}_{NEUT}\ P[NEUT],$$

where

α is a part of the energy directly deposited into the coolant;

$P[NEUT]$ is the heat generation rate computed by the SKETCH-N code on the neutronics spatial mesh;

S_{NEUT}^{HC} is the 3D mapping matrix from the neutronics mesh into the heat conduction mesh computed using Eq. (8.1).

The heat generation rate on the fluid dynamics mesh $P[FD]$ is computed as

$$P[FD] = \alpha \; S_{NEUT}^{FD} \; P[NEUT],$$

where S_{NEUT}^{FD} is the 3D mapping matrix from the neutronics mesh into the fluid dynamics mesh computed using Eq. (8.1).

The data mapping from the TRAC spatial meshes into the neutronics mesh is performed in a similar fashion. The Doppler fuel temperature computed by the TRAC code on the heat conduction mesh $TF[HC]$ is mapped into the neutronics mesh as

$$TF[NEUT] = S_{HC}^{NEUT} \; TF[HC],$$

where

$TF[NEUT]$ is the Doppler fuel temperature on the neutronics mesh,

S_{HC}^{NEUT} is the mapping matrix from the heat conduction mesh into the neutronics mesh.

If geometry is specified consistently for the both codes, the mapping matrix S_{HC}^{NEUT} is simply a transpose of the mapping matrix S_{NEUT}^{HC} :

$$S_{HC}^{NEUT} = \left(S_{NEUT}^{HC} \right)^T$$

The treatment of the coolant temperature and coolant density is slightly more involved. Due to the donor-cell method used in the TRAC code, the computed coolant temperatures and coolant densities are the values at the interfaces of the fluid dynamics nodes. To get the node-average values of the coolant temperature on the fluid dynamics axial mesh the linear interpolation is applied as follows

$$TC^n[FD] = 0.5 \; \left(\; TC^n[FD\_TRAC] + TC^{nn}[FD\_TRAC] \; \right), \quad n = 1, \ldots, N;$$

where

n is an axial index of the node;

$TC[FD]$ is the node-average value of the coolant temperature;

$TC[FD\_TRAC]$ is the coolant temperature computed by the TRAC code;

$nn = \begin{cases} n-1, \; if \;\; V_Z \geq 0 \\ n+1, \; if \;\; V_Z < 0 \end{cases}$; where V_Z is the coolant velocity in axial direction.

The coolant density is interpolated in the same way. Then, the node-averaged values of the coolant density and the coolant temperature are mapped into the neutronics mesh in the similar way as described above for the fuel temperature.

9. CONCLUSIONS AND FUTURE PLANS

9.1 Summary of the SKETCH-N Code Features

The basic features of the nodal three-dimensional neutron kinetics code SKETCH-N are summarized in the following:

- diffusion approximation;
- 3D, 2D and 1D reactor models in Cartesian geometry with arbitrary mesh size in any direction;
- arbitrary number of neutron energy groups and delayed neutron precursors;
- transverse-integrated polynomial, semi-analytic and analytic nodal methods with quadratic leakage approximation for spatial discretization;
- nonlinear iteration procedure for a solution of the nodal equations;
- fully-implicit scheme with analytic integration of the delayed neutron precursors for time discretization;
- adaptive time step control based on the step doubling technique;
- inverse iterations with Wieland shift accelerated by Chebyshev polynomials for the steady-state eigenvalue problems;
- adaptive Chebyshev acceleration procedure for the neutron kinetics problems;
- block symmetric Gauss-Seidel method as a preconditioner;
- consistent point kinetics option, where the point kinetics parameters are computed at each time step using the values of macro cross sections;
- internal thermal-hydraulics model for PWR operational transients with single-phase coolant flow;
- interface module based on the message passing library PVM for the coupling with external thermal-hydraulics codes, such as TRAC.

9.2 Outline of the Code Verification

This report describes only the SKETCH-N models, verification results of the LWR benchmark problem calculations are given in the papers cited below. The computed steady-state LWR neutronics problems include:

- classical 2D & 3D International Atomic Energy Agency (IAEA) PWR problems;
- 2D Biblis PWR checker-board-loaded core;
- 2D Zion-1 PWR problem with explicit modeling of the baffle;
- 2D 4-group Koeberg PWR checker-board-loaded core with realistic cross sections including up-scattering.

The results are given in (Zimin, 1997; Zimin et al., 1998). The neutron kinetics module has been verified against 3D Langenbuch-Maurer-Werner (LMW) operational transient in a small PWR model; and 2D & 3D super-prompt-critical rod drop accident in BWR (Zimin and Ninokata, 1998). The SKETCH-N code has been coupled with the J-TRAC (TRAC-PF1) and TRAC-BF1 codes. The J-TRAC/SKETCH-N code system has been verified against NEACRP PWR rod ejection benchmark and NEA/NSC PWR benchmark on uncontrolled rod withdrawal at zero power (Zimin et al., 1999; Asaka et al., 2000). Assessment of the TRAC-BF1/SKETCH-N code system has been performed by NEACRP BWR cold water injection benchmark (Asaka et al., 2000; Zimin et al., 2000).

To outline the SKETCH-N accuracy we give some results of the steady-state LWR neutronics calculations in Table 9.1 taken from (Zimin et al., 1998).

Table 9.1. SKETCH-N numerical results of the steady-state LWR benchmark problems.

Problem, Mesh	Nodal Method	E_{max}/E_{av}, %	Position of E_{max} (X,Y)	k_{eff}	Reference k_{eff}	Δk_{eff}, pcm	No. of Iterations: Nonlinear/ Outers
IAEA-2D 9x9	NEM[1)]	-2.05/-	(6,6)	1.0295	1.0296	-	-/32
	PNM	-1.8/0.7	(6,6)	1.02950	1.029585[2)]	-8	9/28
	SANM	0.5/0.2	(6,5)	1.02956		-3	
	ANM	1.1/0.3	(5,7)	1.02962		+3	
BIBLIS-2D, 9x9	NEM[3)]	1.25/0.49	-	1.02528	1.02512	-	-
	PNM	-1.2/0.4	(1,8)	1.02521	1.025110[2)]	+10	10/30
	SANM	1.8/0.5	(6,6)	1.02526		+15	
	ANM	2.2/0.7	(6,6)	1.02532		+21	
BWR-2D 11x11	NEM[4)]	1.36/0.26	(1,1)	0.996329	0.99636[4)]	-	-/22
	PNM	1.9/0.6	(1,1)	0.99626		-10	8/28
	SANM	0.5/0.2	(1,1)	0.99635		-1	
	ANM	0.2/0.1	(9,6)	0.99641		+5	
IAEA-3D 9x9x19	NEM[1)]	-1.62/-	(6,6)	1.0290	1.0290[1)]	-	-/24
	PNM	-1.8/0.7	(6,6)	1.02899	1.029074[5)]	-8	8/38
	SANM	0.4/0.2	(7,5)	1.02905		-2	
	ANM	1.1/0.3	(7,5)	1.02912		+4	
BWR-3D 12x12x14	NEM[4)]	1.17/0.22	(1,1)	0.996361	0.996368[4)]	-	7/21
	PNM	1.4/0.4	(1,1)	0.99633		-4	7/37
	SANM	0.4/0.1	(1,1)	0.99637		0	
	ANM	0.5/0.1	(8,8)	0.99643		+6	
LMW 6x6x10	NEM[1)]	-1.2/-	(5,4)	0.99958	0.99966[1)]	-	-/19
	PNM	-1.3/0.4	(5,4)	0.99958		-8	7/26
	SANM	0.4/0.1	(4,4)	0.99971		+5	
	ANM	0.5/0.2	(4,4)	0.99977		+11	

[1)] the results are taken from (Bandini, 1990);
[2)] the reference results are taken from (Muller and Weiss, 1991);
[3)] the NEM results are taken from (Greenman et al., 1979);
[4)] the results are taken from (Gehin, 1992);
[5)] the reference results computed by the SKETCH-N code with SANM on spatial mesh 34x34x38 (5 cm in X-Y directions and 10 cm in Z direction).

The presented problems include

- 2D & 3D IAEA PWR problems (Argonne Code Center Benchmark Problem Book, 1977),
- 2D BIBLIS PWR checker-board-loaded core (Muller and Weiss, 1992);

and initial conditions of the neutron kinetics benchmarks:

- 3D LMW LWR problem (Langenbuch et al., 1977);
- 2D and 3D BWR LRA problems (Argonne Code Center Benchmark Problem Book, 1977).

The SKETCH-N results are given using the polynomial (PNM), semi-analytic (SANM) and analytic (ANM) nodal methods. The results of the nodal expansion method (NEM) taken from the literature are added for a comparison. All calculations are performed using one node per fuel assembly in x-y plane. The reference solutions for these problems are computed using a fine spatial mesh or high-order nodal methods. A comparison is performed for the multiplication factor k_{eff} and the assembly-averaged 2D power densities. The errors in k_{eff} are given in pcm (1 pcm=0.001 %), the maximum E_{max} and average E_{av} errors in 2D power distribution are given in percents. The results show that all the nodal methods provide acceptable accuracy for these problems. The maximum errors of the assembly-averaged power density are less than 2.5% and the errors in eigenvalue are about 10-15 pcm. The accuracy of the semi-analytic

and analytic methods is usually better than that of PNM, except for the BIBLIS-2D checker-board-loaded core. Table 9.1 also shows a number of nonlinear iterations and a total number of outer iterations. The iterative solution techniques are efficient, the number of the nonlinear iterations does not exceed 10 for all the given problems, and the total number of the outer iterations does not exceed 50.

Accuracy of the neutron kinetics module is illustrated by the NEACRP PWR rod ejection benchmark (Finneman and Galati, 1992). All six cases of the benchmark are computed by the J-TRAC/SKETCH-N code system (Zimin et al., 1999). The cases A1, B1 and C1 are defined at the hot zero power (HZP) initial conditions, while the cases A2, B2, and C2 are specified at the full power (FP). Table 9.2 and 9.3 show respectively the steady-state and transient results taken from (Zimin et al., 1999).

Table 9.2. J-TRAC/SKETCH-N steady-state results of the PWR NEACRP rod ejection benchmark (upper values), PANTHER reference solution (middle values) and the deviations from the PANTHER reference solution *(lower values)*.

Parameter	Hot Zero Power (HZP)			Full Power (FP)		
	Case A1	Case B1	Case C1	Case A2	Case B2	Case C2
Critical Boron Concentration, ppm	561.9 561.20 *+0.7 ppm*	1248.7 1247.98 *+0.7 ppm*	1129.2 1128.29 *+0.9 ppm*	1155.4 1156.63 *-1.2 ppm*	1182.2 1183.83 *-1.6 ppm*	1155.6 1156.63 *-1 ppm*
3D Nodal Power Peaking Factor	2.8770 2.8792 *-0.08 %*	1.9333 1.9330 *+0.02 %*	2.1873 2.1867 *+0.02 %*	2.2223 2.2073 *+0.7 %*	2.1107 2.0954 *+0.7 %*	2.2213 2.2073 *+0.6 %*
Control Rod Worth, Pcm	822.2 824.31 *-2 pcm*	826.4 826.18 *+0.2 pcm*	948.4 949.09 *-0.7 pcm*	85 91.58 *- 7 pcm*	92 99.45 *- 7 pcm*	74 79.23 *-5 pcm*

Table 9.3 J-TRAC/SKETCH-N transient results of the PWR NEACRP rod ejection benchmark (upper values), PANTHER reference solution (middle values) and deviations from the PANTHER reference solution *(lower values)*.

Parameter	Hot Zero Power (HZP)			Full Power (FP)		
	Case A1	Case B1	Case C1	Case A2	Case B2	Case C2
Time to the Power Peak, s	0.547 0.5375 *+0.01 s*	0.525 0.5225 *+0.003 s*	0.2732 0.2712 *+0.002* s	0.095 0.095 -	0.095 0.1 *-0.005 s*	0.095 0.095 -
Power at the Peak, %	123.0 126.78 *-3.0 %*	229.7 231.51 *-0.8 %*	437.0 441.12 *-1 %*	108.0 108.3 *-0.3 %*	106.4 106.4 -	107.25 107.34 *+0.1 %*
Power at 5 s, %	19.0 19.69 *-3.5 %*	31.0 31.97 *-3.0 %*	14.3 14.60 *-2 %*	103.3 103.62 *-0.3 %*	103.8 103.94 *-0.1 %*	102.9 103.14 *-0.2 %*
Average Doppler Temperature at 5 s, °C	324.1 324.89 *-0.8 ℃*	349.2 349.96 *-0.8 ℃*	315.8 315.91 *-0.1 ℃*	554.8 555.16 *-0.4 ℃*	552.8 552.39 *+0.4 ℃*	553.7 553.90 *-0.2 ℃*
Max Fuel Centerline Temperature at 5 s, °C	672 679.30 *-7℃*	556 559.66 *- 4 ℃*	675 674.20 *+0.8 ℃*	1688 1679.6 *+8 ℃*	1556 1576.10 *-20 ℃*	1724 1723.80 *+0.2 ℃*
Coolant Outlet Temperature at 5 s, °C	293.1 293.22 *-0.1 ℃*	297.6 297.72 *-0.1 ℃*	291.54 291.53 *0.01 ℃*	325.7 324.90 *+0.8 ℃*	325.9 324.98 *+0.9 ℃*	325.6 324.77 *+0.8℃*

The J-TRAC/SKETCH-N calculations are performed using the semi-analytic nodal method with 4 nodes per fuel assembly in x-y plane. Automatic time step control is used in the transient calculations. The results are compared with the reference solution computed by the PANTHER code using fine spatial and temporal meshes (Knight and Bryce, 1997). Both the steady-state and transient results show excellent agreement with the reference PANTHER results. Please, note that similar results can be also computed by the stand-alone SKETCH-N code with the internal thermal-hydraulics model.
The presented results and the other results given in the cited papers demonstrate acceptable accuracy and efficiency of the SKETCH-N code for LWR safety analysis and design.

REFERENCES

Akimoto, H., et al. (1989). 18th Report on the Best-Estimate Code Development. J-TRAC User Manual (Version 62.1), JAERI Report, JAERI-Memo-63-478, January 1989. [In Japanese].

Al-Chalabi, R. M., P. J. Turinsky, F. X. Faure, H. N. Sarsour and P. R. Engrand (1993). NESTLE: A Nodal Kinetics Code. *Trans. Am. Nucl. Soc.*, **68**, 432.

Argonne Code Center Benchmark Problem Book, *ANL-7416*, Suppl. 2 (1977).

Asaka H., V. Zimin, T. Iguchi, S. Ohmizu and Y. Anoda, (2000). Coupling of the Thermal-Hydraulic TRAC Codes with 3D Neutron Kinetics Code SKETCH-N. *Proc. of the OECD/CSNI Workshop on Advanced Thermal-Hydraulic and Neutronics Codes: Current and Future Applications,* Barcelona, Spain, 10-13 April, 2000.

Aviles, B. N. (1993). Development of a Variable Time-Step Transient NEM Code: SPANDEX. *Trans. Am. Nucl. Soc.,* **68**, 425.

Bandini, B. R., (1990). A Three-Dimensional Transient Neutronics Routine for the TRAC-PF1 Reactor Thermal Hydraulic Computer Code, Ph.D. Thesis, Pennsylvania State University, LA-SUB-95-21.

Borkowski, J., K. Smith, D. Hagrman, D. Kropaczek, J. Phodes, III and P. Esser (1996). SIMULATE-3K Simulation of the Ringhals 1 BWR Stability Measurements. *Proc. Int. Conf. on the Physics of Reactors (PHYSOR 96),* 16-20 September, Mito, Ibaraki, Japan, p. J-121.

Cameron, I. R. (1982), *Nuclear Fission Reactors*, Plenum Press, New York

Crouzet, N. and P. Turinsky (1996). A Second-Derivative-Based Adaptive Time-Step Method for Spatial Kinetics Calculations. *Nucl. Sci. Eng.*, **123**, 206.

Downar, T., Barber, D., Ebert, D., Mousseau, V., (1997). Three-dimensional spatial kinetics for coupled thermal-hydraulics systems analysis codes. Presentation on 25th Water Reactor Safety Meeting, 22 October 1997.

Downar, T. J. , H. G. Joo and G. Jiang, (1997). A Hybrid ANM/NEM Interface Current Technique for the Nonlinear Nodal Calculations, *Proc. Joint Int. Conf. On Mathematical Methods and Supercomputing for Nuclear Applications*, Saratoga Spring, New York, October 5-9, 1997, I-124, American Nuclear Society (1997)

Engrand, P. R., G. I. Maldonado, R. M. Al-Chalabi and P. J. Turinsky, (1992). Nonlinear Iterative Strategy for NEM Refinement and Extension. *Trans. Am. Nucl. Soc.*, **65**, 221.

Ferguson, D.R. and K. L. Derstine (1977). Optimized Iteration Strategies and Data Management Considerations for Fast Reactors Finite Difference Diffusion Theory Codes. *Nucl. Sci. Eng.,* **64**, 593.

Finnemann, H. (1975). A Consistent Nodal Method for the Analysis of Space-Time Effects in Large LWR's. *Proc. Joint NECRP/CSNI Spesialists' Mtg. New Developments in Three-Dimensional Neutron Kinetics and Review of Kinetics Benchmark Calculations,* Garching, FRG, January 22-24, 1975, p.13.

Finnemann, H., F. Bennewitz and M. R. Wagner, (1977). Interface Current Techniques for Multidimensional Reactor Calculations. *Atomkernenergie,* **30**, 123.

Finnemann, H., and A. Galati, (1992). NEACRP 3-D LWR core transient benchmark. Final specifications, NEACRP-L-335 (Revision 1), January 1992.

Gantmacher, F. R., (1959). *Theory of Matrices, Vol. I,* Chapter V, Chelsea Publishing Company, New York.

Gehin J. C., (1992). A Quasi-Static Polynomial Nodal Method for Nuclear Reactor Analysis, PhD Thesis, Nuclear Engineering, Massachusetts Institute of Technology, DOE/OR/0033-T557.

Golub, G. H. and C. H. Van Loan, (1996), *Matrix Computations,* Chapter 11, 3rd Edition, John Hopkins University Press, Baltimore and London.

Greenman G., K. Smith and A. F. Henry, (1979). Recent Advances in an Analytic Nodal Method for Static and Transient Reactor Analysis. *Proc. of Topl. Meet. On Computational Methods in Nuclear Engineering,* Williamsburg, Virginia, April 23-25, 1979, American Nuclear Society.

Hageman, L.A. and D. M. Young (1981). *Applied Iterative Methods*, Academic Press, New York.

Hairer, E., S. P. Norset and G. Wanner (1987). *Solving Ordinary Differential Equations I. Nonstiff Problems*. Springer-Verlag, Berlin.

Hairer, E., and G. Wanner (1996). *Solving Ordinary Differential Equations II. Stiff and Differentia-Algebraic Problems*. 2nd Edition, Springer-Verlag, Berlin.

Hennart, J. P. (1988). On Numerical Analysis of Analytical Nodal Methods. *Numer. Methods Partial Different. Equ.*, **4**, 233.

Henry, A. F. (1975), *Nuclear Reactor Analysis,* MIT Press, Cambridge, Massachusetts and London.

Inoue S., T. Fujimura, T. Tsutsui and T. Nishida, (1982). JSSL-JAERI Scientific Subroutine Library, 3rd Edition, JAERI-M-82-095, July 1982. [In Japanese].

Johansson, M. (1994). Stability benchmark. Questions and News, Note 94/L/1051, September 9, 1994.

Joo, H. G., T. J. Downar and D. A. Barber (1996). Methods and Performance of a Parallel Reactor Kinetics Code PARCS. *Proc. Int. Conf. on the Physics of Reactors PHYSOR 96,* 16-20 September, Mito, Ibaraki, Japan, p. J-42.

Knight, M.P., Bryce, P., (1997). Derivation of a Refined PANTHER Solution to the NEACRP PWR Rod Ejection Transients. *Proc. of the Joint Int. Conf. on Mathematical Methods and Supercomputing for Nuclear Applications*, Saratoga Springs, N.Y., October 5-9, 1997, American Nuclear Society, vol. 1, pp. 302-313.

Lahey, R. T.,Jr. and F. J. Moody (1977). *Thermal-Hydraulics of a Boiling Water Reactor,* American Nuclear Society,

Langenbuch, S., W. Maurer and W. Werner, (1977). Coarse-Mesh Flux-Expansion Method for the Analysis of Space-Time Effects in Large Light Water Reactor Cores. *Nucl. Sci. Eng.*, **63**, 437.

Lawrence, R. D. (1986). Progress in Nodal Methods for the Solution of the Neutron Diffusion and Transport Equations, *Progress in Nuclear Energy,* **17**, 271.

Lefvert, T, (1994). BWR Stability Benchmark. Final Specification. NEA/NSC/DOC(94)15, March 1994.

Marchuk, G.I. and V.I. Lebedev (1981). *Numerical Methods in the Theory of Neutron Transport,* Atomizdat, Moscow, [In Russian].

Moler, C. and C. Van Loan, (1978). Nineteen Dubious Ways to Compute the Exponential of a Matrix. *SIAM Review,* **20**, 801-836.

Muller, E. Z. and Z. J. Weiss, (1991). Benchmarking with the Multigroup Diffusion High-Order Response Matrix Method, *Ann. Nucl. Energy*, **9**, 535.

Noh, J.M., and N. Z Cho, (1994). A New Approach of Analytic Basic Function Expansion to Neutron Diffusion Nodal Calculation, *Nucl. Sci. Eng.*, **116**, 165-180.

Noh, J. M., L. Pogosbekyan, Y. J. Kim and H. K. Joo, (1997). A General Approach to Multigroup Extension of the Analytic Function Expansion Nodal Method Based on Matrix Function Theory, *Proc. Joint Int. Conf. On Mathematical Methods and Supercomputing for Nuclear Applications*, Saratoga Spring, New York, October 5-9, 1997, I-144, American Nuclear Society.

Patankar, S., (1980). *Numerical Heat Transfer and Fluid Flow,* Hemisphere Publishing Corporation.

Pease III, M. C., (1965). *Methods of Matrix Algebra,* Chapter VI, Academic Press, New York.

Press, W. H., Teukolsky, S. A., Vetterling, W. T., Flannery, B. P., (1992). *Numerical Recipes in Fortran: The Art of Scientific Computing, vol. I.,* 2nd Edition,. Cambridge University Press, Cambridge.

Saad, Y. (1994). SPARSKIT: A Basic Tool Kit for Sparse Matrix Computations, Version 2, University of Minnesota, Minneapolis, June 1994.

Saad, Y. (1996). *Iterative Methods for Sparse Linear Systems,* PWS Pub. Co., Boston.

Shikhov, S.B. and V. B. Troyanskii, (1983). *Nuclear Reactor Theory, vol. 2, Transport Theory,* Energoatomizdat, Moscow, [In Russian]

Shober, R. A., R. N. Sims, and A. F Henry (1977). Two Nodal Methods for Solving Time-Dependent Group Diffusion Equations. *Nucl. Sci. Eng.*, **64**, 528.

Smith, K. S. (1979), *An Analytic Nodal Method for Solving the Two-Group, Multidimensional, Static and Transient Neutron Diffusion Equations,* Thesis, Nuclear Engineering, Massachusetts Institute of Technology.

Smith, K. S. (1984). Nodal Method Storage Reduction by Nonlinear Iteration. *Trans. Am. Nucl. Soc.*, **44**, 265.

Stacey, W. M. (1969). *Space-Time Nuclear Reactor Kinetics*, Academic Press, New York.

Tomasevic, D. I., (1997). A Consistent Transverse Leakage Approximation. *Proc. of the Joint Int. Conf. on Mathematical Methods and Supercomputing for Nuclear Applications*, Saratoga Springs, N.Y., October 5-9, 1997, American Nuclear Society, vol. 1, pp. 134-143.

Ueberhuber, C. W. (1995) *Numerical Computation: Methods, Software and Analysis,* Vol. I and II, Springer-Verlag, Berlin, New York.

Varga, R. S. (1962). *Matrix Iterative Analysis*, Prentice-Hall, Englewood Cliffs, New Jersey.

Wachspress, E. L. (1966). *Iterative Solution of Elliptic Systems and Applications to the Neutron Diffusion Equations of Reactor Physics*. Prentice-Hall, Englewood Cliffs, New Jersey.

Wagner, C. (1999). Introduction to Algebraic Multigrid, Version 1.0, University of Heidelberg, January 1999.

Wakil M. M., (1993). *Nuclear Heat Transport.* American Nuclear Society.

Wolfram S., (1996). *The Mathematica Book,* 3rd Edition, Cambridge University Press, Cambridge.

Zimin V. G. (1997). Nodal Neutron Kinetics Models Based on Nonlinear Iteration Procedure for LWR Analysis, PhD Thesis, Research Laboratory for Nuclear Reactors, Tokyo Institute of Technology, August 1997.

Zimin, V.G., Ninokata, H. (1998). Nodal Neutron Kinetics Model Based on Nonlinear Iteration Procedure for LWR Analysis. *Ann. Nucl. Energy,* **25**, 507-528.

Zimin, V.G., Ninokata, H., Pogosbekyan L., (1998). Polynomial and Semi-Analytic Nodal Methods for Nonlinear Iteration Procedure. *Proc. of the Int. Conf. on the Physics of Nuclear Science and Technology (PHYSOR)*, October 5-8, 1998, Long Island, New York, American Nuclear Society, vol. 2, pp. 994-1002.

Zimin V. G., H. Asaka, Y. Anoda and M. Enomoto, (1999). Verification of the J-TRAC Code with 3D Neutron Kinetics Model SKETCH-N for PWR Rod Ejection Analysis. *Proc. of the 9 International Topical Meeting on Nuclear Reactor Thermal Hydraulics (NURETH 9),* San Francisco, California, October 3-8, 1999, CD-ROM, American Nuclear Society.

Zimin V. G., H. Asaka, Y. Anoda, E. Kaloinen, R. Kyrki-Rajamäki, (2000). Analysis of NEACRP 3D BWR Core Transient Benchmark. *Proc. of the 4th Intl. Conf. on Supercomputing in Nuclear Application* (*SNA 2000),* September 4-7, 2000, Tokyo, Japan, CD-ROM.

Zimin V. G. and A. A. Semenov (2002) "Methodology of the Building Dependencies of the Neutron Cross Sections for Few-Group Reactor Calculations based on Regression Analysis", Internal Report, MEPhI, Oct. 2002 [In Russian]