THREE DIMENSIONAL HETEROGENEOUS FINITE ELEMENT METHOD FOR STATIC MULTI-GROUP NEUTRON DIFFUSION

by

Elif Can Aydogdu

A Thesis Submitted in Partial Fulfillment

of the Requirements for the Degree of

Master of Applied Science

in

Faculty of Energy and Nuclear Science
Nuclear Engineering Program

University of Ontario Institute of Technology

August 2010

ABSTRACT

Because current full-core neutronic-calculations use two-group neutron diffusion and rely on homogenizing fuel assemblies, reconstructing pin powers from such a calculation is an elaborate and not very accurate process; one which becomes more difficult with increased core heterogeneity. A three-dimensional Heterogeneous Finite Element Method (HFEM) is developed to address the limitations of current methods by offering fine-group energy representation and fuel-pin-level spatial detail at modest computational cost. The calculational cost of the method is roughly equal to the calculational cost of the Finite Differences Method (FDM) using one mesh box per fuel assembly and a comparable number of energy groups. Pin-level fluxes are directly obtained from the method’s results without the need for reconstruction schemes.

Keywords: Heterogeneous Finite Element Method, Static Neutron Diffusion Equation, Finite Difference Equations, Weighted Residuals Method.

ACKNOWLEDGEMENTS

I would like to offer my sincerest gratitude to my supervisor and mentor, Dr. Eleodor Nichita, who has supported my efforts with patience and wisdom. I would like to express my appreciation for his friendly and helpful encouragements which gave me the confidence to pursue this research topic.

I offer many thanks to my thesis committee Mr. Peter Schwanke, Dr. Rachid Machrafi, Dr. Glenn Harvel, and Dr. Igor Pioro, for their time and valuable suggestions for improving this thesis.

I would also like to thank my friends Cristina Naidin, Eric Haghverdian, and Pouya Saneipour for their friendship and for making me feel at home. Infinite thanks to Mert Genc who guided me through numerous programming challenges; unreservedly and regardless of the time of day. My warmest gratitude goes to my friends Tan, Ulku, Aras, Maria, Alp, Anil, Turan, Omer, Emre, Alper, Gizem, Ozan, Can, Istek, Mert, Payam, Umut, Barkin and Sera for making me feel that I am not forgotten.

Warmest thanks to my best friend, my sister, my pink angel, Gamze, who keeps me awake when I am exhausted by reminding me of my responsibility to continue to breathe on her behalf as well. Thank you for your magic friendship. I feel your soul always with me, as I promised.

Probably the most difficult role during my education belonged to my dear friend, my love, Murat. He always supported me and believed in me at every step in my life and

encouraged me when I felt exhausted. Without him, I would have been lost a long time ago. Thanks for being there whenever I needed you and for following along in my pursuits. Thank you for your continuous and unlimited patience and for your great love and friendship.

My deepest gratitude goes to my entire family for their unflagging love and support throughout my life. They always encouraged me to pursue my dreams, even when those dreams knew no boundaries. I thank them for listening to my complaints and frustrations, and for believing in me. I have been extremely fortunate to have their support. Without their encouragement and understanding it would have been impossible for me to finish this work.

I am forever indebted to my mother, Gulhan, who made me believe that “nothing is impossible” and gave me the patience and energy that I needed. Although she is no longer with me, her soul is always with me and that makes me feel stronger. I am sure she shares my joy and happiness in Heaven.

Lastly, I offer my regards and blessings to all those who supported me in any way during this project.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

NOMENCLATURE

G	:coarse group index
g	:fine group index
e	:finite element index
n	:node (sampling point) index
r	:fine region index
R	:coarse region index
V	: volume (cm^3)
d^3r	: volume element (cm^3)
dE	: differential energy interval (eV)
v(E)	: neutron speed (cm/sec)
$d\hat{\Omega}$	: differential angle (steradian)
$d\vec{S}$	: differential surface element (cm^2)
$\vec{r}$	:position vector
$\hat{n}$	:outward pointing unit normal (to a given surface)
$n(\vec{r},E,\hat{\Omega})$	:phase space neutron density (neutrons/cm^3/eV/steradian)
$\varphi(\vec{r},E,\hat{\Omega})$	: angular neutron flux (neutrons/cm^2sec/eV/steradian)
$\chi(E)$	: normalized fission-neutron spectrum (fraction)
$\Sigma_t(\vec{r},E)$	: total macroscopic cross section for the neutrons with energy E and position $\vec{r}$ (cm^{-1}/eV)

$\Sigma_s(\vec{r},E' \rightarrow E,\hat{\Omega}' \rightarrow \hat{\Omega})$: scattering macroscopic cross section for neutrons with energy an initial energy E' and final energy E, initial direction of travel $\hat{\Omega}'$ and final direction of travel $\hat{\Omega}$ at position $\vec{r}$ (cm^{-1}/eV/steradian)

$\nu(E)$: average number of neutrons born per fission due to incident neutrons of energy E'

$\Sigma_f(\vec{r},E')$: macroscopic fission cross section for neutrons with initial position $\vec{r}$ and energy E' (cm^{-1}/eV)

$\nu\Sigma_f(\vec{r},E')$: macroscopic production (yield) cross section for neutrons with position $\vec{r}$ and energy E' (cm^{-1}/eV)

$\vec{J}(\vec{r},E,\hat{\Omega})$:neutron current vector (neutrons/cm^{2}sec/eV/steradian)

$\varphi(\vec{r},E)$:integral / total neutron flux (neutrons/cm^{2}sec/eV)

$D(\vec{r},E)$:diffusion coefficient (cm/eV)

E_g :upper bound of g energy group-g (eV)

$D_g(\vec{r})$: diffusion coefficient for group g at time t (cm)

$\Sigma_{sg'\rightarrow g}(\vec{r})$: scattering term from group g' to g at time t and location $\vec{r}$ (cm^{-1})

χ_g :Chi fraction for group g neutrons (fraction)

k_{eff} :effective multiplication constant

$\varphi_g(\vec{r})$: neutron flux in group g neutrons (neutrons/cm^{2}sec)

$\Sigma_{tg}(\vec{r})$: total macroscopic cross section for group g neutrons (cm^{-1})

$D_g(\vec{r})$: diffusion coefficient for group g neutrons (cm)

$\Sigma_{sg'\rightarrow g}(\vec{r})$: scattering macroscopic cross section from group g’ to g (cm^{-1})

χ_g : Fraction of fission neutrons with energy in group g (Chi fraction)

$\nu\Sigma_{fg'}(\vec{r})$:macroscopic production cross section of group g’ neutrons (cm^{-1})

u_{gn} :nth basis function for group g in the Finite Element Method

a_{gn} : co coefficient multiplying function u_{gn} in the in Finite Element system

$w_m(\vec{r})$: weight function of type m in the Finite Element Method

M_{ng}^{m} : loss matrix in the Finite Element Method

flux for group g neutrons and mth type weight function

F_{ng}^{m} : production matrix in the Finite Element Method

φ_{ne} : type e elementary basis function for node n

f_{sg}^{i} : discontinuity factor for group g homogeneous flux averaged over the surface s of node i

$\overline{\Phi}_{sg}$: face-average of group g homogeneous flux (face s)

$\overline{\Psi}_{sg}$: face-average of group g heterogeneous flux (face s)

${}^{0}\Psi_g(\vec{r})$:zero-current heterogeneous assembly flux shape of group g neutrons

$\Phi_g(\vec{r})$:nodal shape for group g neutrons

$\Phi = [\varphi_{g,i,j,k}]$:flux vector whose elements consist of the flux for fine-group g and fine-mesh box (i, j, k)

$\psi_{n,G}(g, i, j, k)$:discrete basis function corresponding to node n and coarse-energy-group G (HEM)

$w_{m,G}(g,i,j,k)$:discrete weight function for the node m and coarse energy-group G (HFEM)

$\varphi_{G,n,e}(g,i,j,k)$:discrete elementary basis function of type e corresponding to the node n and coarse energy-group G (HFEM)

OPERATORS AND SYMBOLS

∇	:Del operator
Δ	: Laplacian operator
M	:Loss operator
F	:Production operator
L	:Leakage operator
S	: Scattering operator
T	:Total reaction operator
$\underline{\underline{A}}$	:Notation used for matrix A
$\underline{c}$	:Notation used for column vector c

ACRONYMS

FD	:Finite Differences
FEM	:Finite Element Method
DF	:Discontinuity Factor
ADF	:Assembly Discontinuity Factor
GET	:Generalized Equivalence Theory
FFD	:Forward Finite Differences
BFD	:Backward Finite Differences
CFD	:Central Finite Differences
FDM	:Finite Difference Method
FGFMFDM	:Fine-group fine-mesh Finite Difference Method
FGCMFEM	:Fine-group coarse-mesh Finite Element Method
CGHFEM	:Coarse-group Heterogeneous Finite Element Method
RMS	:Root Mean Square

CHAPTER 1

INTRODUCTION

1.1 PREAMBLE

Finding the power-generation level at every point in a nuclear reactor is one of the major tasks of reactor physics. To find the fission-power generation rate, one needs to find the neutron flux at each point. The neutron flux is known to satisfy the neutron transport (Boltzmann) equation, which is an integro-differential equation in three spatial dimensions, an energy dimension and two angular dimensions, for a total of six dimensions in the static case and seven in the time-dependent case. Solving such an equation is a difficult task and numerical methods, as well as approximations, are heavily relied upon. In particular, if linear angular dependence of the neutron flux is assumed, the transport equation reduces to the diffusion equation which is an integro-differential equation in energy and space only, with angular dependence being dropped. The number of dimensions is thus reduced from six to four in the static case and from seven to five, in the time dependent case. There are different ways to discretize the diffusion equation in energy and space for subsequent numerical solution. Energy is usually discretized using a multigroup approach, whereas space can be discretized using a finite difference, finite element or nodal approach.

Because reactor cores are heterogeneous and neighbouring fuel pins can have considerably different properties, a large number (100-200) of energy groups and spatial regions (one homogenized region for each pin cell consisting of a fuel pin and surrounding coolant) would normally be required for the solution of the neutron diffusion equation. Such an approach is known as a many-group (or fine-group) fine-mesh approach and is very time consuming. To reduce the computational time, entire fuel assemblies are homogenized before proceeding to the solution of the diffusion equation. Additionally, the number of energy groups is condensed from 100-200 to a small number, usually two. Using fewer energy groups and less spatial detail is known as the few-group (or coarse-group) coarse-mesh approach and is far less time consuming. However, the use of homogenized assemblies leads to some loss of information, and hence of accuracy, in the flux solution (pin-level power information is lost). If individual pin powers are desired, one needs to use “pin power reconstruction” or “de-homogenization” schemes which rely on simplifying assumptions and do not offer very accurate results.

In designing next generation (Gen. III and IV) nuclear reactors, the tendency has been to go to even more heterogeneous cores than the ones of current-generation (Gen. II) reactors. Additionally, tighter regulatory requirements ask for accurate knowledge of fuel-pin power rather than for assembly-level power. Such increased accuracy for higher-heterogeneity cores can be achieved with current methods only by using pin-cell-level homogenization and a higher number (4-10) energy groups, that is by using a fine-group, fine-mesh approach, which is time consuming and hence impractical for routine design and analysis calculations. The purpose of this work was to develop a method that would provide fine-group neutron-flux detail and accurate pin-level flux and power distributions at modest computational cost. The result of the work is the development of a three-dimensional Heterogeneous Finite Element Method.

1.2 THESIS ORGANIZATION

Chapter two presents a theoretical background consisting of a brief overview of discretization and homogenization methods currently in use for full-core neutronic calculations analyzed from the perspective of computational cost. Chapter three presents the current progress, full-core neutronic calculations, namely attempts to date to treat increased heterogeneity and obtain better spatial detail without large increases in computational costs. Chapter four presents the developed three-dimensional Heterogeneous Finite Element Method and its implementation in a FORTRAN code. Chapter four presents the results obtained for a test case, which prove the functionality of the developed code. Chapter five presents conclusions stemming from the work performed and some proposed directions for future development.

CHAPTER 2

THEORETICAL BACKGROUND

2.1 STATIC NEUTRON TRANSPORT EQUATION

In a nuclear reactor the following processes take place (Henry,1975):

- Production of neutrons by induced fission
- Loss of neutrons by absorption
- Loss of neutrons by leakage
- Scattering of neutrons to other energies

The neutron transport equation expresses the neutron balance in an elementary volume. To derive its expression some notations will be first introduced. The derivation follows the one in Ott and Bezella (1983) ;Duderstadt and Hamilton (1976).

V : volume

d^3r : volume element

$n(\vec{r},E,\hat{\Omega})d^3rdEd\hat{\Omega}$: density of neutrons with energy E in dE and the direction of travel $\hat{\Omega}$ in solid angle $d\hat{\Omega}$

$\int_V n(\vec{r},E,\hat{\Omega})dEd\hat{\Omega}d^3r$: number of neutrons in the volume V with energy E in dE and the direction of travel $\hat{\Omega}$ in $d\hat{\Omega}$

$\mathrm{v(E)}$: neutron speed

$\varphi(\vec{r},E,\hat{\Omega}) \equiv n(\vec{r},E,\hat{\Omega})v(E)$: angular neutron flux

$\chi(E)$: normalized fission-neutron spectrum

$\Sigma_t(\vec{r},E)$: total macroscopic cross section for the neutrons with energy E a position $\vec{r}$

$\Sigma_s(\vec{r},E' \to E,\hat{\Omega}' \to \hat{\Omega})$: scattering macroscopic cross section for the neutrons with energy an initial energy E' and final energy E, initial direction of travel $\hat{\Omega}'$ and final direction of travel $\hat{\Omega}$ at position $\vec{r}$

$\nu(E)$: average number of neutrons born per fission due to incident neutrons of energy E'.

$\Sigma_f(\vec{r},E')$: macroscopic fission cross section for neutrons with initial position $\vec{r}$ and energy E'

$\nu\Sigma_f(\vec{r},E') \equiv \nu(E')\Sigma_f(\vec{r},E')$: macroscopic production cross section for neutrons with position $\vec{r}$ and energy E'

$\left[\int_V \Sigma_t(\vec{r},E,\hat{\Omega})\varphi(\vec{r},E,\hat{\Omega})d^3r\right]dEd\hat{\Omega}$: rate of neutron loss due to collisions of the neutrons in volume V with energy E in dE and the direction of travel $\hat{\Omega}$ in $d\hat{\Omega}$

$\left[\int\limits_V d^3r \int\limits_{4\pi} d\hat{\Omega}' \int\limits_o^{\infty} dE' \left(\Sigma_s(\vec{r},E' \to E,\hat{\Omega}' \to \hat{\Omega})\,\varphi(\vec{r},E',\hat{\Omega}')\right)\right] dE d\hat{\Omega}$: rate of neutron gain due to scattering of the neutrons in the volume V with final energy E in dE and the final direction of travel $\hat{\Omega}$ in $d\hat{\Omega}$

$\chi(E)\left[\int\limits_V d^3r \int\limits_{4\pi} d\hat{\Omega}' \int\limits_o^{\infty} dE' \nu\Sigma_f(\vec{r},E',\hat{\Omega}')\varphi(\vec{r},E',\hat{\Omega}')\right] dE d\hat{\Omega}$: rate of neutron gain due to fissioning of nuclei in the volume V with final energy E in dE and the final direction of travel $\hat{\Omega}$ in $d\hat{\Omega}$

The neutron leakage term includes both neutron leakage into and out of the volume V. The net leakage through the surface S enclosing volume V of neutrons with energy E in dE and direction of travel $\hat{\Omega}$ in $d\hat{\Omega}$ is found as

$\int\limits_S \vec{J}(\vec{r},E,\hat{\Omega})d\vec{S} = \int\limits_S \hat{\Omega}\varphi(\vec{r},E,\hat{\Omega})d\vec{S}$: neutron current through surface element dS for neutrons with energy E in dE and the direction of travel $\hat{\Omega}$ in $d\hat{\Omega}$

$\left[\int\limits_S \hat{\Omega}\varphi(\vec{r},E,\hat{\Omega})d\vec{S}\right] dE d\hat{\Omega}$: net rate of neutron leakage through the entire surface S for neutrons with energy E in dE and the direction of travel $\hat{\Omega}$ in $d\hat{\Omega}$.

Gauss Theorem (for a closed surface S) (Chow,2000):

$$\int\limits_S \vec{A}(\vec{r})d\vec{S} = \int\limits_V d^3r\nabla \cdot\vec{A}(\vec{r}) \qquad \textbf{(2.1)}$$

Where, $\vec{A}(\vec{r})$ is a vector depending on $\vec{r}$.

By using Eq. (2.1) Gauss' Theorem for a closed surface S:

$$\int_S \hat{\Omega}\varphi(\vec{r},E,\hat{\Omega})d\vec{S}dEd\hat{\Omega} = \left[\int_V d^3r\nabla\cdot\hat{\Omega}\varphi(\vec{r},E,\hat{\Omega})\right]dEd\hat{\Omega} \tag{2.2}$$

Since $\hat{\Omega}$ does not depend on r;

$$\nabla\cdot\hat{\Omega}\varphi(\vec{r},E,\hat{\Omega}) = \hat{\Omega}\nabla\varphi(\vec{r},E,\hat{\Omega}) \tag{2.3}$$

so,

$$\int_S \hat{\Omega}\varphi(\vec{r},E,\hat{\Omega}d\vec{S})dEd\hat{\Omega} = \hat{\Omega}\left[\int_V d^3r\nabla\varphi(\vec{r},E,\hat{\Omega})\right]dEd\hat{\Omega} \tag{2.4}$$

Combining all the reaction rates with their correct signs gives the steady-state Boltzmann Transport Equation for neutrons:

$$\begin{aligned}
&\int_V d^3r\ \hat{\Omega}\nabla\varphi(\vec{r},E,\hat{\Omega})dEd\hat{\Omega} + \Sigma_t(\vec{r},E,\hat{\Omega})\varphi(\vec{r},E,\hat{\Omega})dEd\hat{\Omega} \\
&-\int_{4\pi} d\hat{\Omega}'\int_0^\infty dE'\left(\Sigma_s(\vec{r},E'\rightarrow E,\hat{\Omega}'\rightarrow\hat{\Omega})\varphi(\vec{r},E',\hat{\Omega}')\right)dEd\hat{\Omega} = \\
&\frac{1}{k}\frac{1}{4\pi}\chi(E)\int_{4\pi} d\hat{\Omega}'\int_0^\infty dE'\nu\Sigma_f(\vec{r},E',\hat{\Omega}')\varphi(\vec{r},E',\hat{\Omega}')dEd\hat{\Omega}
\end{aligned} \tag{2.5}$$

The left hand side (LHS) of the Eq. (2.5) represents the net loss of neutrons whereas the right hand side (RHS) represents the neutron production by fission. The $\frac{1}{k}$ "scaling" factor multiplying the fission production constitutes the eigenvalue of the problem and ensures that the problem always has a solution. k is called the neutron multiplication factor.

Since the equation holds true for any volume, no matter how small and for any energy and angle, the integrand needs to vanish, resulting in:

$$\hat{\Omega}\nabla\varphi(\vec{r},E,\hat{\Omega})+\Sigma_t(\vec{r},E,\hat{\Omega})\varphi(\vec{r},E,\hat{\Omega})$$
$$-\int_{4\pi}d\hat{\Omega}'\int_{o}^{\infty}dE'\big(\Sigma_s(\vec{r},E'\rightarrow E,\hat{\Omega}'\rightarrow\hat{\Omega})\varphi(\vec{r},E',\hat{\Omega}')\big)=$$
$$\frac{1}{k}\frac{1}{4\pi}\chi(E)\int_{4\pi}d\hat{\Omega}'\int_{o}^{\infty}dE'\nu\Sigma_f(\vec{r},E',\hat{\Omega}')\varphi(\vec{r},E',\hat{\Omega}') \tag{2.6}$$

The transport equation ,Eq. (2.6), describes the exact behaviour of the neutrons but it is difficult to solve. To reduce its complexity, assumptions are often made about the angular dependence. If the angular dependence is assumed to be linear, the diffusion equation is obtained.

2.2 NEUTRON DIFFUSION EQUATION

The first important step in deriving the diffusion equation from the Boltzmann equation ,Eq. (2.6), is the use of the integral flux $\varphi(\vec{r},E)=\int_{4\pi}\varphi(\vec{r},\hat{\Omega},E)d\Omega$ instead of the angular flux $\varphi(\vec{r},\hat{\Omega},E)$.(Ott and Bezella,1983) The integro-differential form of the Boltzmann Equation is integrated over the angle $\hat{\Omega}$.

$$\int_{4\pi} d\hat{\Omega}\left(\hat{\Omega}\nabla\varphi(\vec{r},E,\hat{\Omega})\right)+\int_{4\pi} d\hat{\Omega}\left(\Sigma_t(\vec{r},E,\hat{\Omega})\,\varphi(\vec{r},E,\hat{\Omega})\right)-$$
$$\int_{4\pi} d\hat{\Omega}\left(\int_{4\pi} d\hat{\Omega}'\int_{o}^{\infty} dE'\left(\Sigma_s(\vec{r},E'\to E,\hat{\Omega}'\to\hat{\Omega})\,\varphi(\vec{r},E',\hat{\Omega}')\right)\right)= \tag{2.7}$$
$$\frac{1}{k}\int_{4\pi} d\hat{\Omega}\left(\frac{1}{4\pi}\chi(E)\int_{4\pi} d\hat{\Omega}'\int_{o}^{\infty} dE'\nu\Sigma_f(\vec{r},E',\hat{\Omega}')\,\varphi(\vec{r},E',\hat{\Omega}')\right)$$

Based on the definition of the integral flux ($\varphi(\vec{r},E)=\int_{4\pi}\varphi(\vec{r},\hat{\Omega},E)d\Omega$)the following hold true:

$$\int_{4\pi} d\hat{\Omega}\left(\hat{\Omega}\nabla\varphi(\vec{r},E,\hat{\Omega})\right)=\nabla\int_{4\pi} d\hat{\Omega}\left(\hat{\Omega}\varphi(\vec{r},E,\hat{\Omega})\right)=\nabla\cdot\vec{J}(\vec{r},E) \tag{2.8a}$$

$$\int_{4\pi} d\hat{\Omega}\left(\Sigma_t(\vec{r},E,\hat{\Omega})\,\varphi(\vec{r},E,\hat{\Omega})\right)=\Sigma_t(\vec{r},E)\,\varphi(\vec{r},E) \tag{2.8b}$$

$$\int_{4\pi} d\hat{\Omega}\left(\int_{4\pi} d\hat{\Omega}'\int_{o}^{\infty} dE'\left(\Sigma_s(\vec{r},E'\to E,\hat{\Omega}'\to\hat{\Omega})\,\varphi(\vec{r},E',\hat{\Omega}')\right)\right)$$
$$=\int_{o}^{\infty} dE'\left(\Sigma_s(\vec{r},E'\to E)\,\varphi(\vec{r},E')\right) \tag{2.8c}$$

$$\int_{4\pi} d\hat{\Omega}\left(\frac{1}{4\pi}\chi(E)\int_{4\pi} d\hat{\Omega}'\int_{o}^{\infty} dE'\nu\Sigma_f(\vec{r},E',\hat{\Omega}')\,\varphi(\vec{r},E',\hat{\Omega}')\right)$$
$$=\chi(E)\int_{o}^{\infty} dE'\nu\Sigma_f(\vec{r},E')\,\varphi(\vec{r},E') \tag{2.8d}$$

The angle-integrated balance equation can therefore be written as

$$-\nabla\cdot\vec{J}(\vec{r},E)+\Sigma_t(\vec{r},E)\varphi(\vec{r},E)+\int_o^\infty dE'\Sigma_s(\vec{r},E'\to E)\varphi(\vec{r},E')=$$
$$\frac{1}{k}\chi(E)\int_o^\infty dE'\big(\nu\Sigma_f(\vec{r},E')\varphi(\vec{r},E')\big) \tag{2.9}$$

Eq. (2.9) has two unknown functions: the integral flux and the current.

To reduce the function to one with a single unknown, a relationship between the integral flux and the current is postulated in the form of Fick’s Law:

$$\vec{J}(\vec{r},E)=-D(\vec{r},E)\nabla\varphi(\vec{r},E) \tag{2.10}$$

Fick’s Law is not exact, but only an approximation (Stacey,2002).

Fick’s Law is not valid for:

- strongly absorbing media (control rods, fuel)
- strongly anisotropic scattering medium
- close (several - ~ ± 3 - mean free paths) to a neutron source or interfaces or system boundaries

Nonetheless, except for cases with extremely large absorption or with strongly anisotropic scattering the overall error introduced by using Fick’s Law is considered acceptable.

When Fick's Law is used the diffusion equation can be written as

$$-\nabla\cdot\left[D(\vec{r},E)\nabla\varphi(\vec{r},E)\right]+\Sigma_t(\vec{r},E)\varphi(\vec{r},E)$$
$$+\int_0^\infty dE'\Sigma_s(\vec{r},E'\rightarrow E)\varphi(\vec{r},E')=\frac{1}{k}\chi(E)\int_0^\infty dE'\left(\nu\Sigma_f(\vec{r},E')\varphi(\vec{r},E')\right)$$

(2.11)

Eq. (2.11), is referred to as the steady-state, continuous-energy neutron diffusion equation.

For the problem to be well posed, it needs to be supplemented by homogeneous boundary conditions of the type:

$$\left.\frac{\partial\varphi(\vec{r},E)}{\partial n}-\alpha\varphi(\vec{r},E)\right|_S=0$$

(2.12)

where $\vec{n}$ is the unit normal vector pointing out of the surface S.

Each side of the diffusion equation can be represented as a linear operator acting on the flux, leading to the operator form of the diffusion equation:

$$M\varphi(\vec{r},E)=\frac{1}{k}F\varphi(\vec{r},E)$$

(2.13)

M represents the loss operator, whereas F represents the production operator. The operator form makes it obvious that steady-state diffusion is a generalized eigenvalue-eigenvector problem.

2.3 DISCRETIZATION OF THE DIFFUSION EQUATION

Analytical or semi-analytical techniques can be applied to the neutron diffusion problem but only for very simple cases. For real cases, the standard practice is to use numerical techniques to solve the neutron diffusion equation. To do that, the equation needs to be discretized.

2.3.1 Energy Discretization: Multigroup Approach

Neutron energy dependence is usually discretized using a multigroup approach. The following derivation follows the one in Duderstadt (1976). To develop a multigroup calculational method for energy distribution, the neutron energy interval of interest is divided into "G" intervals which are called energy groups as shown in Fig. (2.1)

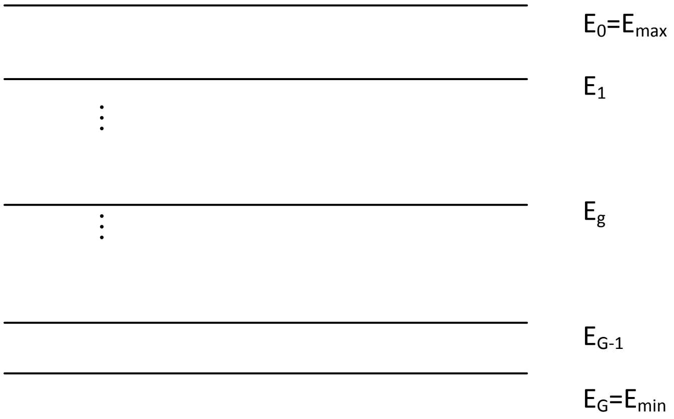


Figure 2.1: Multi-group energy structure

The diffusion equation can be integrated over the energy interval $E_g \leq E \leq E_{g-1}$

$$-\int_{Eg}^{Eg-1}\nabla\cdot\left[D(\vec{r},E)\nabla\varphi(\vec{r},E)\right]dE+\int_{Eg}^{Eg-1}\left(\Sigma_t(\vec{r},E)\,\varphi(\vec{r},E)\right)dE$$

$$+\int_{Eg}^{Eg-1}\int_{o}^{\infty}dE'\left(\Sigma_s(\vec{r},E'\to E)\,\varphi(\vec{r},E')\right)dE=\frac{1}{k}\int_{Eg}^{Eg-1}\chi(E)\int_{o}^{\infty}dE'\left(\nu\Sigma_f(\vec{r},E')\,\varphi(\vec{r},E')\right)dE \qquad \textbf{(2.14)}$$

The following definitions are introduced:

$\varphi_g(\vec{r})\equiv\int_{Eg}^{Eg-1}dE\,\varphi(\vec{r},E)$: Neutron flux in group g **(2.15a)**

$\Sigma_{tg}(\vec{r})=\dfrac{1}{\varphi_g(\vec{r},t)}\int_{Eg}^{Eg-1}dE\,\Sigma_t(\vec{r},E)\,\varphi(\vec{r},E)$:Total macroscopic cross section for group g **(2.15b)**

$D_g(\vec{r})=\dfrac{\int_{Eg}^{Eg-1}dE D(\vec{r},E)\nabla\varphi(\vec{r},E)}{\int_{Eg}^{Eg-1}dE\nabla\varphi(\vec{r},E)}$: Diffusion coefficient for group g **(2.15c)**

$$\int_{Eg}^{Eg-1}dE\int_{o}^{\infty}dE'\left(\Sigma_s(\vec{r},E'\to E)\,\varphi(\vec{r},E')\right)=\sum_{g'=1}^{G}\int_{Eg}^{Eg-1}dE\int_{Eg'}^{Eg'-1}dE'\left(\Sigma_s(\vec{r},E'\to E)\,\varphi(\vec{r},E')\right)$$

then; **(2.15d)**

$\Sigma_{sg'\to g}(\vec{r})=\dfrac{1}{\varphi_{g'}(\vec{r})}\int_{Eg}^{Eg-1}dE\int_{Eg'}^{Eg'-1}dE'\left(\Sigma_s(\vec{r},E'\to E)\,\varphi(\vec{r},E')\right)$: Scattering term from group g’ to g

(2.15e)

$\chi_g=\int_{Eg}^{Eg-1}dE\chi(E)$: Chi fraction of group g **(2.15f-1)**

$$\int_{Eg}^{Eg-1} \chi(E)dE\int_{o}^{\infty} dE'\left(\nu\Sigma_f(\vec{r},E')\varphi(\vec{r},E')\right) =$$
$$\chi_g\left[\sum_{g'=1}^{G} \int_{Eg'}^{Eg'-1} dE'\left(\nu\Sigma_f(\vec{r},E')\varphi(\vec{r},E')\right)\right] \quad \textbf{(2.15f-2)}$$

$$\nu\Sigma_{fg'}(\vec{r}) = \frac{1}{\varphi_{g'}(\vec{r})}\int_{Eg'}^{Eg'-1} dE'\left(\nu\Sigma_f(\vec{r},E')\varphi(\vec{r},E')\right)$$: Macroscopic production cross section. **(2.15g)**

By using the definitions given in Eqs. (15) the multigroup diffusion equation is obtained:

$$-\nabla\cdot D_g(\vec{r})\nabla\varphi_g(\vec{r}) + \Sigma_{tg}(\vec{r})\varphi_g(\vec{r}) - \sum_{g'=1}^{G}\Sigma_{sg'\to g}(\vec{r})\varphi_{g'}(\vec{r})$$
$$= \frac{1}{k}\chi_g\sum_{g'=1}^{G}\nu\Sigma_{fg'}(\vec{r})\varphi_{g'}(\vec{r})$$
$$\text{where } g = 1,2,3...G \quad \textbf{(2.16)}$$

2.3.2 Space Discretization

While a range of possible space discretization methods are outlined in the literature, the two most prominent methods, the Finite Difference Method (FDM) and the Finite Element Method (FEM), are used in this study.

2.3.2.1 Finite Differences Method

Finite Differences (FD) is a method which is used to solve ordinary and partial differential equations with given boundary conditions (Leveque, 2006). The solution function is approximated by its values at a discrete set of points and the derivatives or partial derivatives involved in the equation are approximated by, as the name says, finite differences. After applying the finite differences discretization ordinary or partial differential equations are reduced to a set of linear algebraic equations which can be solved numerically by computers.

While relatively elaborate finite-difference schemes can be devised for special applications (Nichita, Zabienski and Gravel, 2007), for a constant mesh size and uniform diffusion coefficient the FD approximation of the diffusion equation takes the very simple form below:

$$\begin{aligned}
&-D_g \frac{\varphi_g(i+1,j,k)+\varphi_g(i-1,j,k)-2\varphi_g(i,j,k)}{h_x^2} \\
&-D_g \frac{\varphi_g(i,j+1,k)+\varphi_g(i,j\text{-}1,k)-2\varphi_g(i,j,k)}{h_y^2} \\
&\text{-}D_g \frac{\varphi_g(i,j,k+1)+\varphi_g(i,j,k\text{-}1)-2\varphi_g(i,j,k)}{h_z^2} \\
&+\Sigma_{tg}(i,j,k)\varphi_g(i,j,k)-\sum_{g'=1}^{G}\Sigma_{sg'\to g}(i,j,k)\varphi_{g'}(i,j,k) \\
&=\frac{1}{k}\chi_g(i,j,k)\sum_{g'=1}^{G}\nu\Sigma_{fg'}(i,j,k)\varphi_{g'}(i,j,k)
\end{aligned} \tag{2.17}$$

where, (i,j,k) represent discrete grid points.

It can easily be seen that the above set of equations defines a linear homogeneous system corresponding to a generalized eigenvalue-eigenvector problem:

$$M\Phi=\frac{1}{k}F\Phi \tag{2.18}$$

where $\Phi = [\varphi_{g,i,j,k}]$ represents the eigenvector. Quantities $\varphi_{g,i,j,k}$ are the system unknowns, and represent the flux in group g at position (i,j,k). A detailed and general derivation of the FD equations, including boundary conditions is shown in Appendix A.

2.3.2.2 Finite Element Method

The Finite Element Method (FEM) is another way of discretizing the spatial dependence of the neutron flux which uses a totally different approach from finite differences space discretization (Ise, Yamazaki and Nauwra ,1978).

The starting point is the usual form of the multigroup diffusion equation, which can be written as:

$$
\begin{aligned}
&-\nabla \cdot \left(D_g(\vec{r})\nabla\varphi_g(\vec{r})\right) + \Sigma_{tg}(\vec{r})\varphi_g(\vec{r}) - \sum_{g'=1}^{G}\Sigma_{sg'\to g}(\vec{r})\varphi_{g'}(\vec{r}) \\
&= \frac{1}{k}\chi_g(\vec{r})\sum_{g'=1}^{G}\nu\Sigma_{fg'}(\vec{r})\varphi_{g'}(\vec{r}) = 0
\end{aligned}
\tag{2.19}
$$

The finite element method approximates the solution by a linear combination of predetermined functions (usually piecewise polynomials) called basis functions (Strang and Fix , 1975). The solution is sought as a linear combination of such basis functions.

$$
\varphi_g(\vec{r}) \approx \hat{\varphi}_g(\vec{r}) = \sum_{n=1}^{N} a_{ng}u_n(\vec{r})
\tag{2.20}
$$

In Eq. (2.20) ,which shows the flux approximation, u_{gn} is the nth basis function for group g and a_{gn} is its corresponding coefficient in the linear combination. N is the total number of basis functions used to approximate the space dependence of the neutron flux in each group g.

Substituting the sought-for linear combination in the diffusion equation, the following is obtained:

$$-\sum_{n=1}^{N} a_{ng}\nabla\cdot\left(D_g(\vec{r})\nabla u_n(\vec{r})\right)+\sum_{n=1}^{N} a_{ng}\Sigma_{tg}(\vec{r})u_n(\vec{r})-\sum_{n=1}^{N} a_{ng}\sum_{g'=1}^{G}\Sigma_{sg'\to g}(\vec{r})u_n(\vec{r})$$
$$=\frac{1}{k}\chi_g(\vec{r})\sum_{n=1}^{N} a_{ng}\sum_{g'=1}^{G}\nu\Sigma_{fg'}(\vec{r})u_n(\vec{r}) \qquad \textbf{(2.21)}$$

Defining the inner product of two functions u and v as (Ott and Neuhold ,1985):

$$\langle u;v\rangle \equiv \int_V u(\vec{r})\vec{v}(\vec{r})d^3r \qquad \textbf{(2.22)}$$

and taking the inner product of the diffusion equation with a set of trial functions w_m (m=1...N) for each group g, the following set of equations is obtained:

$$-\sum_{n=1}^{N} a_{ng}\int_V w_m(\vec{r})\nabla\cdot\left(D_g(\vec{r})\nabla u_n(\vec{r})\right)d^3\vec{r}+\sum_{n=1}^{N} a_{ng}\int_V w_m(\vec{r})\Sigma_{tg}(\vec{r})u_n(\vec{r})d^3\vec{r}$$
$$-\sum_{n=1}^{N} a_{ng}\sum_{g'=1}^{G}\int_V w_m(\vec{r})\Sigma_{sg'\to g}(\vec{r})u_n(\vec{r})d^3\vec{r}=\frac{1}{k}\sum_{n=1}^{N} a_{ng}\int_V\sum_{g'=1}^{G} w_m(\vec{r})\chi_g(\vec{r})\nu\Sigma_{fg'}(\vec{r})u_n(\vec{r})d^3\vec{r} \qquad \textbf{(2.23)}$$

Calculating the integral involving the divergence can become problematic if the basis functions are not twice differentiable.

To avoid that difficulty one can apply Gauss' theorem to the divergence term and, assuming for simplicity, that zero fluxes are used as boundary conditions, one can write:

$$\begin{aligned}&-\sum_{n=1}^{Nf} a_{ng}\int_V \nabla w_m(\vec{r})\cdot\left(D_g(\vec{r})\nabla u_n(\vec{r})\right)d^3\vec{r} + \sum_{n=1}^{Nf} a_{ng}\int_V w_m(\vec{r})\Sigma_{tg}(\vec{r})u_n(\vec{r})d^3\vec{r}\\&-\sum_{n=1}^{Nf} a_{ng}\sum_{g'=1}^{G}\int_V w_m(\vec{r})\Sigma_{sg'\to g}(\vec{r})u_n(\vec{r})d^3\vec{r} = \frac{1}{k}\sum_{n=1}^{Nf} a_{ng}\int_V\sum_{g'=1}^{G} w_m(\vec{r})\chi_g(\vec{r})\,\nu\Sigma_{fg'}(\vec{r})u_n(\vec{r})d^3\vec{r}\end{aligned} \tag{2.24}$$

The above only requires the basis functions to have first derivatives.

The simplest choice of trial functions is to choose them to be the same as the basis functions. In that case the FEM equations become:

$$\begin{aligned}&-\sum_{n=1}^{N} a_{ng}\int_V \nabla u_m(\vec{r})\cdot\left(D_g(\vec{r})\nabla u_n(\vec{r})\right)d^3\vec{r} + \sum_{n=1}^{N} a_{ng}\int_V u_{mg}(\vec{r})\Sigma_{tg}(\vec{r})u_n(\vec{r})d^3\vec{r}\\&-\sum_{n=1}^{N} a_{ng}\sum_{g'=1}^{G}\int_V u_m(\vec{r})\Sigma_{sg'\to g}(\vec{r})u_n(\vec{r})d^3\vec{r} = \frac{1}{k}\sum_{n=1}^{N} a_{ng}\int_V\sum_{g'=1}^{G} u_m(\vec{r})\chi_g(\vec{r})\,\nu\Sigma_{fg'}(\vec{r})u_n(\vec{r})d^3\vec{r}\end{aligned} \tag{2.25}$$

With obvious notations, the above equations can be written as:

$$\sum_{n=1}^{N} M_{ng}^{m} a_{ng} = \frac{1}{k}\sum_{n=1}^{Nf} F_{ng}^{m} a_{ng} \tag{2.26}$$

Eq. (2.26) represents a linear eigenvalue-eigenvector problem with the unknown eigenvector being defined by coefficients a_{ng}.

What makes the Finite Element Method unique is the way in which the basis functions are chosen. In the FEM, the domain is divided into smaller sub-domains called elements. In a Cartesian three-dimensional representation the elements are simply parallelepipeds. The corners of the elements are called nodes. There is one basis function for each node, and it is only nonzero inside the neighbouring elements of the node and zero everywhere else. It takes a value of 1 at the node and it decreases progressively away from the node until it vanishes on the far borders of the neighbouring elements. It follows that the number of basis functions used in the solution is actually equal to the number of nodes in the geometry. A simple representation of the basis functions for a one-dimensional case is shown in Fig. (2.2). Looking at Fig. (2.2) it can be seen that basis function n, for example, takes a value of 1 at node n and decreases to zero by the time it reaches the far boundaries of the two neighbouring elements of node n, which, in this case, coincide with nodes n-1 and n+1 respectively.

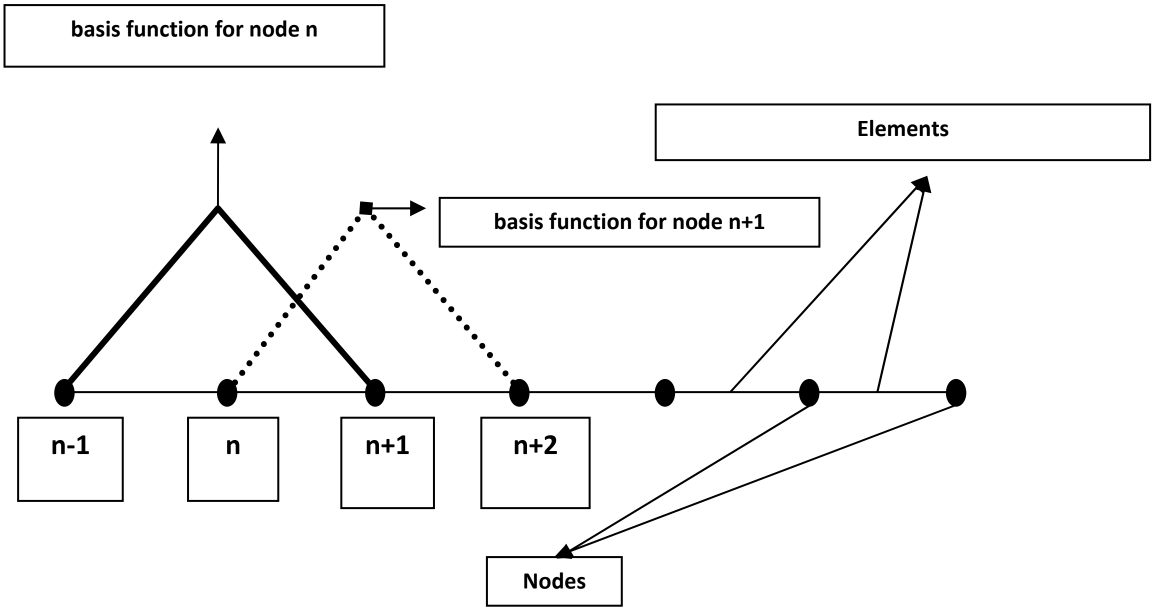


Figure 2.2: Schematic representation of elements, nodes and linear node basis functions in one-dimensional geometry.

Each basis function corresponding to a node n can be thought of as a sum of elementary basis functions each of which is nonzero only in one of the elements surrounding the node. Each restriction of the basis function to one of its neighbouring elements is called an elementary basis function. The sum is written:

$$u_n = \sum_e \varphi_{ne} \tag{2.27}$$

In Eq. (2.27) φ_{ne} is the elementary basis function corresponding to node n and (neighbouring) element e.

From the definition of the basis functions it follows that each elementary basis function φ_{ne} takes a value of 1 at node n and vanishes at all the other nodes in element e. With this choice of (elementary) basis functions, each unknown coefficient a_{ng} represents the amplitude of the group-g flux at node n.

Figure (2.3) shows schematically a basis functions and its two elementary basis functions for a one-dimensional geometry and linear elementary basis functions.

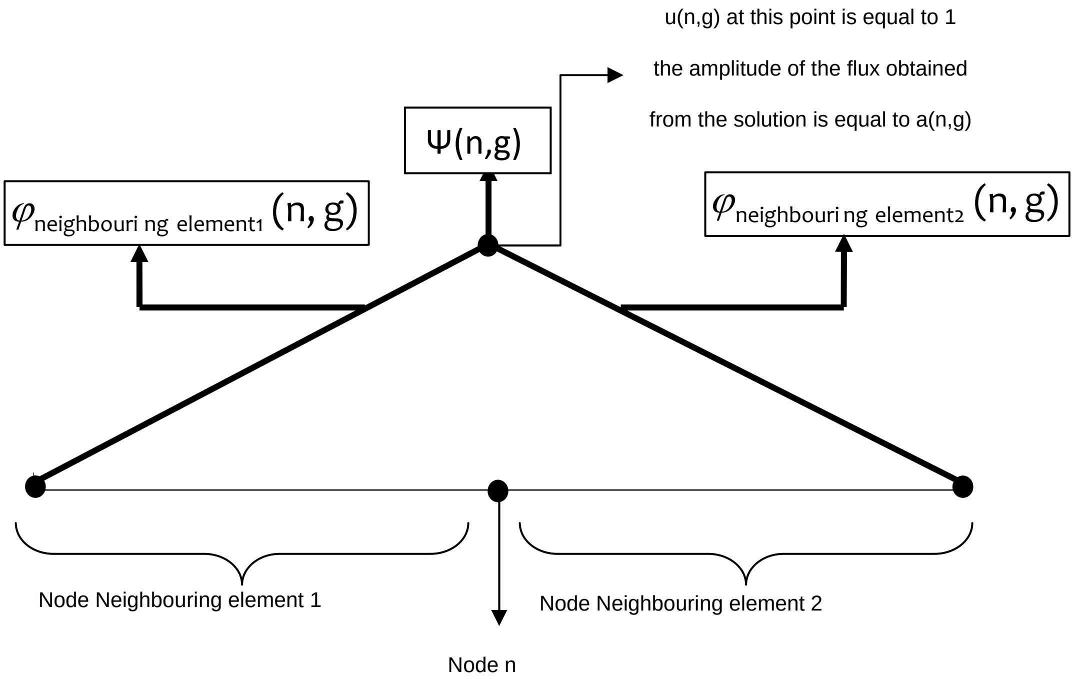

u(n,g) at this point is equal to 1
the amplitude of the flux obtained
from the solution is equal to a(n,g)
Ψ(n,g)
$\varphi_{\text{neighbouring element1}}(n,g)$
$\varphi_{\text{neighbouring element2}}(n,g)$
Node Neighbouring element 1
Node Neighbouring element 2
Node n

Figure 2.3: One dimensional representation of elementary basis functions as "branches" of a linear basis function

2.4 HOMOGENIZATION AND GROUP CONDENSATION – LATTICE AND CORE CALCULATIONS

Nuclear reactor cores are composed of a large number of fuel assemblies, each containing a large number of discrete fuel elements of different compositions, as well as cladding, coolant, structural elements, burnable poisons, water channels, control rods and so on, which leads to the existence of thousands of regions and high heterogeneity of the reactor core. Since the system is highly heterogeneous, ideally we should divide the core into many small pieces and solve the multi-group transport equation for the entire core to find the detailed and accurate flux distribution.

Fine-spatial discretization, accompanied by fine-energy discretization, using 100 or more fine-energy groups, would be ideal for the discretization of the full-core transport problem. To estimate the number of unknowns to be evaluated in such a detailed model, we can ignore the flux angular dependence and assume that each fuel pin is homogeneous along with the coolant and moderator regions in each fuel bundle. If this discretization were to be used, the number of unknowns for a CANDU core would be

$$380\frac{\text{channels}}{\text{core}}\text{x}12\frac{\text{bundles}}{\text{channels}}\text{x}39\frac{\text{regions(37pin}+\text{1coolant region}+\text{1moderator region)}}{\text{bundle}}$$

$$\text{x}100\frac{\text{energy group}}{\text{region}}=17{,}784{,}000$$

Solving a linear system with that many unknowns is not practical, so homogenization and group condensation is used. Additionally, to simplify the mathematical problem, the neutron transport equation is replaced by the diffusion equation, which is generally easier to discretize and solve numerically than the transport equation.

Neutronic calculations therefore proceed in two steps: a lattice step, and a core step. In the lattice step, a transport calculation using a fine-geometry and 100 or more fine groups is performed for each reactor “node” encompassing a fuel assembly and its corresponding moderator/coolant. The boundary conditions for lattice calculations are usually reflective. Subsequently, few-group flux- and volume-averaged macroscopic cross sections and diffusion coefficients are calculated for each node (Stacey,2002) as follows.

Group Condensation

$$\Sigma_G = \frac{\sum_{g \in G} \Sigma_g \varphi_g}{\sum_{g \in G} \varphi_g} \text{(coarse group G macroscopic cross section)} \quad \textbf{(2.28a)}$$

$$\Phi_G = \sum_g \varphi_g \text{ (coarse group G condensed flux)} \quad \textbf{(2.28b)}$$

Simple Homogenization

$$V_R = \sum_r V_r \text{ (volume of node R)} \quad \textbf{(2.29a)}$$

$$\hat{\Phi}_{RG} = \frac{\sum_r V_r \Phi_{rG}}{V_R} \text{(node R homogenized flux)} \quad \textbf{(2.29b)}$$

$$\hat{\Sigma}_{RG} = \frac{\sum_r \Sigma_{rG} V_r \Phi_{rG}}{\hat{\Phi}_{RG} V_r} \text{(node R homogenized macroscopic cross - section)} \quad \textbf{(2.29c)}$$

Simultaneous Homogenization and Group Condensation

$$\hat{\Phi}_{RG} = \frac{\sum_r V_r \sum_{g \in G} \varphi_{rg}}{\sum_r V_r} \text{(region R coarse group G flux)} \quad \textbf{(2.30a)}$$

$$\hat{\Sigma}_{RG} = \frac{\sum_r \sum_{g \in G} \Sigma_{rg} \varphi_{rg} V_r}{\sum_r V_r \sum_{g \in G} \varphi_{rg}} \text{(region R coarse group G macroscopic cross} - \text{section)} \quad \textbf{(2.30b)}$$

Finally, the core calculation is performed in few (usually two) energy groups assuming each node to be homogeneous with cross-sections generated during the lattice-calculation step.

The use of assembly-average cross sections leads to some loss of information, and hence of accuracy, in the flux solution (pin-level power information is lost). Moreover, single-assembly lattice calculations need to rely on approximate assembly-boundary conditions, namely reflective assembly-boundary conditions, which are different from the real (but unknown) assembly-boundary conditions present when the assembly is surrounded by other assemblies in the reactor. The use of approximate assembly-boundary conditions leads to some additional errors in evaluating the assembly-averaged cross sections. Nonetheless, this approach offers reasonable accuracy in calculating fuel assembly powers as long as the core is not too heterogeneous (i.e. the assembly-boundary conditions are not too far from reflective) and some more advanced homogenization methods are used. With the methodology currently in use, if individual pin powers are desired, one needs to rely on “pin power reconstruction” or “de-homogenization” methods because the assembly-homogenized-core calculation do not directly provide pin powers (that information having been lost in the homogenization process). Such methods are usually approximate and do not offer very accurate results. If accurate pin powers are desired, an intermediate homogenization step is used, whereby each fuel cell consisting of a fuel pin and surrounding coolant/moderator is homogenized separately and energy is only condensed to 5-10 groups. This is known as pin-cell-level homogenization (as opposed to assembly-level homogenization). The core calculation is then performed using a pin-only homogenized model in diffusion theory and 5-10 energy groups. Such calculations, while feasible, are still computationally challenging and impractical to use for routine design calculations. They are usually reserved for the analysis of safety-related scenarios of particular importance.

CHAPTER 3

CURRENT PROGRESS IN FULL-CORE NEUTRONIC CALCULATIONS

Current full-core flux solution techniques used for routine core analysis employ assembly-level homogenization and group condensation in order to achieve low computation time. Such an approach has limited accuracy in two respects:

1. A node-homogenized model is only an approximate representation of the true heterogeneous structure of the reactor and hence total assembly power results obtained using such a model will differ from those obtained if a heterogeneous model was used.

2. Because the detailed assembly structure information is lost through homogenization, only assembly-integrated powers can be calculated directly. Individual pin powers can only be reconstructed "a posteriori" using de-homogenization schemes, which rely on simplifying assumptions and are not very accurate.

To achieve accurate pin-power distributions, detailed pin-level (instead of assembly-level) homogenization needs to be performed, followed by a full-core fine-group, fine-mesh calculation. Pin-by-pin homogenization can follow a direct flux and volume average or it can use more elaborate techniques, such as the "Superhomogenization" introduced by Hebert (1993). Regardless of the actual technique used for pin-cell homogenization, the corresponding core calculation is time consuming because of the large number of regions.

3.1 HOMOGENIZATION METHODS

3.1.1 Generalized Equivalence Theory

Standard homogenization, while widely used for CANDU reactors, does not guarantee that using averaged cross sections in a full-core node-homogenized model will produce the same node-integrated results as if the true, heterogeneous, model was used. To improve on standard homogenization, Smith (1980) introduced Generalized Equivalence Theory (GET), which offers a way to calculate node-homogenized cross sections which, when used in a full-core node-homogenized calculation, produce the same node-integrated quantities (neutron flux and reaction rates) as the heterogeneous model. According to GET, the homogenized cross sections are calculated, similarly to standard homogenization, as a volume and flux average of heterogeneous cross sections:

$$\hat{\Sigma}_g = \frac{\int_V \Sigma_g(\vec{r})\Psi_g(\vec{r})d^3r}{\int_V \Psi_g(\vec{r})d^3r} \tag{3.1}$$

where Ψ_g is the detailed heterogeneous flux inside the fuel assembly obtained from a lattice calculation using exact assembly boundary conditions. Smith showed that, if homogeneous-core results are to match heterogeneous-core results, the flux for the homogeneous core has to be discontinuous at node boundaries.

Hence, GET replaces the continuity condition by a discontinuity condition:

$$\overline{\Phi}^{i}_{sg} f^{i}_{sg} = \overline{\Phi}^{j}_{sg} f^{j}_{sg} \tag{3.2}$$

where $\overline{\Phi}^{i}_{sg}$ and $\overline{\Phi}^{j}_{sg}$ are the homogeneous fluxes in nodes i and j respectively, averaged over the common face, s. Quantities f^{i}_{sg} and f^{j}_{sg} are called *discontinuity factors (DF)*, and are defined as the ratio of the face-averaged heterogeneous flux $\overline{\Psi}_{sg}$ to the face-averaged homogeneous flux $\overline{\Phi}_{sg}$:

$$f_{sg} \equiv \frac{\overline{\Psi}_{sg}}{\overline{\Phi}_{sg}} \tag{3.3}$$

It will be noted that GET introduces the discontinuity factors in an ad-hoc fashion. A systematic homogenization based on asymptotic expansion has been developed by Zhang, Rizwan-uddin and Dorning (1995). Their method defines the discontinuity factors and homogenized cross-sections by making use of the adjoint function.

In its “pure” form, GET requires prior knowledge of assembly boundary conditions obtained from a reference, heterogeneous, core calculation. However, those boundary conditions are not known beforehand and hence approximations need to be made, such as assuming the assembly boundary conditions to be reflective (zero current). If a zero-current boundary condition is used for the assembly calculation, the discontinuity factors become Assembly Discontinuity Factors, or ADFs (Smith, 1986), expressed as the ratio of the face-averaged heterogeneous flux (obtained using zero-current boundary conditions) to the node-averaged heterogeneous flux:

$$f_{sg}^{0} = \frac{\overline{\Psi}_{Sg}^{0}}{\overline{\Psi}_{Vg}^{0}} \tag{3.4}$$

The errors introduced by the zero-current approximation are known as the “leakage effect”.

In what follows, we present the several methods that have proven to be the most successful to date at treating the leakage effect.

3.1.2 Global-Local Iterations

The global local iteration method, also known as the explicit leakage iteration method was originally proposed by Henry and Hoxie (1981) and further developed by Aragones and Ahnert (1986). It consists of iterating between assembly (fine mesh) and core calculations. At each assembly calculation step, boundary conditions determined from the previous nodal calculation are used.

The method starts by generating the zero-current assembly homogenized parameters. A core calculation is then performed that provides not only the average nodal flux and reaction rates, but also interface average flux and current (or, in other words, leakage). These can subsequently be used to find an average current-to-flux ratio on each node face. The newly determined homogeneous conditions on the node boundaries are then used in fine mesh assembly calculations for each mesh. A new set of homogenized constants is thus generated. The cycle is repeated until convergence is achieved.

The method yields a significant increase in accuracy but is extremely time consuming since, at each leakage-iteration step, fine mesh calculations for each node (assembly) have to be performed.

A variation of the global-local iteration technique was introduced by Rahnema and Nichita (1995, 1997) whose method relies on pre-computing and tabulating cross sections and discontinuity factors for different boundary conditions. At each iteration step, assembly calculations are replaced by table interpolations and the computation time is correspondingly reduced by the time it would have taken to perform a full assembly calculation at each iteration.

3.1.3 Rehomogenization

A simple way of correcting for the leakage effect was proposed by Smith (1994) under the name of "rehomogenization". Smith assumes the intranodal flux to be separable into the product of a nodal and a zero-current assembly shape:

$$\Psi_g(\vec{r}) = \Phi_g(\vec{r})\,{}^0\Psi_g(\vec{r}) \tag{3.5}$$

where ${}^0\Psi_g(\vec{r})$ is the zero-current assembly shape and $\Phi_g(\vec{r})$ is the nodal shape. The assembly shape is normalized to an average value of 1 according to the following Eq. (3.6).

$$\frac{\int_V {}^0\Psi_g(\vec{r})d^3r}{V} = 1 \tag{3.6}$$

As a consequence of the separability assumption, the homogenized cross-sections can be written as:

$$\hat{\Sigma}_{xg} = \frac{\int_V \Sigma_{xg}(\vec{r})\Phi_g(\vec{r})^0\Psi_g(\vec{r})d^3r}{\int_V \Phi_g(\vec{r})^0\Psi_g(\vec{r})d^3r} \tag{3.7}$$

Eq. (3.7) provides the basis for iteration on the nodal shape $\Phi_g(\vec{r})$. At each iteration step, new homogenized cross-sections are generated using it. These cross sections are used in a nodal calculation that provides a new nodal shape $\Phi_g(\vec{r})$, and the cycle is repeated until convergence is achieved.

The method is well suited for use with polynomial nodal methods. In that case the intranodal flux in group g is expressed by:

$$\Phi_g(\vec{r}) = \sum_l c_{lg} P_l(\vec{r}) \tag{3.8}$$

where $P_l(\vec{r})$ are (multivariate) polynomials and c_{lg} are the coefficients of the multivariate polynomial of type "l" for group "g". The expression for the homogenized cross-sections becomes:

$$\hat{\Sigma}_{xg} \equiv \frac{\int\limits_V \Sigma_{xg}(\vec{r})\sum\limits_l c_l P_{lg}(\vec{r})^0\Psi_g(\vec{r})d^3r}{\int\limits_V \sum\limits_l c_l P_{lg}(\vec{r})^0\Psi_g(\vec{r})d^3r} = \frac{\sum\limits_l c_l \int\limits_V \Sigma_{xg}(\vec{r})P_{lg}(\vec{r})^0\Psi_g(\vec{r})d^3r}{\sum\limits_l c_l \int\limits_V P_{gl}(\vec{r})^0\Psi_g(\vec{r})d^3r} \equiv \frac{\sum\limits_l c_l a_l}{\sum\limits_l c_l b_l} \tag{3.9}$$

Quantities a_l and b_l can be pre-computed during the assembly calculation. The cross-section update becomes thus a simple matter of evaluating several products and a ratio.

The procedure is simple to implement and it doesn't require any supplementary assembly calculations. However, it cannot be used to correct the discontinuity factors as will be seen from a simple example given below:

Consider a one-dimensional, one-group problem and assume we want to compute the rehomogenized value of the discontinuity factor on boundary *k* of a particular node. Then, according to the definition of the discontinuity factor and to the separability condition we can express f^k as:

$$f^k = \frac{\Psi^k}{\Phi^k} = \frac{\Phi^{k\,0}\Psi^k}{\Phi^k} = {}^0\Psi^k = \frac{{}^0\Psi^k}{\frac{1}{V}\int\limits_V {}^0\Psi(x)dx} = {}^0f^k \tag{3.10}$$

where the left "0" superscript refers to zero-boundary condition quantities and the overbar has been omitted since in one dimension no face-averaging takes place.

Eq (3.10) shows that applying rehomogenization to the discontinuity factors doesn't yield any correction but rather reproduces the zero-current values.

3.1.4 Multi-Cell Calculations

Another possible approach to reducing the leakage effect is to simulate the true assembly boundary conditions by extending the lattice calculation to a larger model, one including 3 x 3 assemblies. Cross sections are only homogenized for the central assembly, but by changing the local conditions in the neighbouring ones, boundary conditions on the central assembly are implicitly changed. The values of the homogenized cross sections for the central assembly are tabulated as a function of the local conditions in the peripheral assemblies and hence assembly calculations are replaced by table interpolations, which are performed whenever cross-sections for the core calculation need to be determined. An implementation of this method for use with CANDU cores was recently reported by Shen (2006). As implemented, this method does not allow for the correction of discontinuity factors but, as shown by Nichita (2009, 2010) discontinuity factors are not usually needed for CANDU applications.

3.1.5 Dehomogenization / Pin Power Reconstruction

Once the neutron flux for the node-homogenized full core is known, detailed pin powers can be reconstructed if certain simplifying assumptions are made. A possibility is to assume that the flux inside the assembly can approximated by the product between the smooth core shape and the detailed intra-assembly shape, the latter being obtained from a zero-current assembly calculation. With this assumption, the flux and power at each pin position can be calculated, albeit approximately. Another approach to dehomogenization is to solve the diffusion equation in an assembly using approximate boundary conditions determined from the core calculation (e.g. Singh, 1995). Other approaches involve using the (smooth) shape of the neutron source found in the core calculation to solve for the detailed neutron flux in a heterogeneous assembly (Joo, 2009). Yet another technique employs finding the detailed flux in the assembly by using boundary conditions found in the core calculation and Monte Carlo assembly calculations. This is useful when strong heterogeneities are present not only in the x and y directions, but also in the radial direction (Tohjoh, 2006).

3.2 FINITE ELEMENT METHOD FOR FULL CORE NEUTRONIC CALCULATIONS

While not in common use for a majority of industrial codes, finite element methods have been applied to full-core calculations for a long time. For a good introduction to the application of the finite element method to reactor physics the reader is referred to the book by Strang and Fix (1973) or to the excellent paper by Hansen and Kang (1975). More recent developments include the use of discontinuous functions (Akroyd, 1996) or non-polynomial functions (Nichita and Rahnema, 1998) in an attempt to increase the overall accuracy.

3.3 FULL-CORE NEUTRONIC CALCULATIONS NOT REQUIRING ASSEMBLY HOMOGENIZATION

While virtually all full-core calculations rely on first performing assembly-level homogenization, in the last several years there have been attempts to avoid the lattice calculations altogether. One such attempt is based on the response matrix technique. Traditionally, the response matrix technique is used in conjunction with a node-homogenized core model. However, in a new development, response matrices are calculated for heterogeneous nodes using a Monte Carlo approach (Forget and Rahnema, 2006). Because response matrices are calculated for a heterogeneous node, the pin powers can be directly obtained once the neutron-current iterations, on which the response matrix technique is based, converge. A second such attempt was to reduce the number of unknowns in a full-core Collision Probability transport calculation by using a weighted residual method (Nichita, 2009). While the approach shows some promise, to date it has only been implemented into a one-dimensional code.

CHAPTER 4

HETEROGENEOUS FINITE ELEMENT METHOD

4.1 GENERAL METHOD DESCRIPTION

The Heterogeneous Finite Element Method (HFEM) presented here follows the major lines of the usual Finite Element Method, but starts from the fine-group fine-mesh finite-difference-discretized form of the diffusion equation, written in operator form as:

$$M\Phi = \frac{1}{k}F\Phi \tag{4.1}$$

In Eq. (4.1) $\Phi = [\varphi_{g,i,j,k}]$ represents a vector whose elements consist of the flux for fine-group g and fine-mesh box (i, j, k). M and F represent the fine-group fine-mesh discrete forms of the loss operator, M, and production operator, F, defined respectively as:

$$[M\Phi]_{g,i,j,k} = -[L\Phi] + \Sigma_{tg,i,j,k}\varphi_{g,i,j,k} - \sum_{g'=1}^{N_g}\Sigma_{sg'\to g,i,j,k}\varphi_{g',i,j,k} \tag{4.2a}$$

$$[F\Phi]_{g,i,j,k} = \chi_{g,i,j,k}\sum_{g'=1}^{N_g}\nu\Sigma_{fg',i,j,k}\varphi_{g',i,j,k} \tag{4.2b}$$

The leakage operator, L, is defined as the sum of the leakages in each direction:

$$[L\Phi]_{g,i,j,k} = [L_x\Phi]_{g,i,j,k} + [L_y\Phi]_{g,i,j,k} + [L_y\Phi]_{g,i,j,k} \tag{4.3}$$

The directional Lx, Ly and Lz operators are defined as (see Appendix A):

$$\left[L_x\Phi\right]_{g,i,j,k} \equiv \frac{D_{g,i,j,k}}{h_{xi,j,k}}\left[\frac{2D_{g,i+1,j,k}(\varphi_{i+1,j,k}-\varphi_{g,i,j,k})}{h_{xi+1,j,k}D_{g,i,j,k}+h_{xi,j,k}D_{g,i+1,j,k}}-\frac{2D_{g,i-1,j,k}\left(\varphi_{g,i,j,k}-\varphi_{g,i-1,j,k}\right)}{h_{xi,j,k}D_{g,i-1,j,k}+h_{xi-1,j,k}D_{g,i,j,k}}\right] \tag{4.4a}$$

$$\left[L_y\Phi\right]_{g,i,j,k} \equiv \frac{D_{g,i,j,k}}{h_{yi,j,k}}\left[\frac{2D_{g,i,j+1,k}(\varphi_{i,j+1,k}-\varphi_{g,i,j,k})}{h_{xi,j+1,k}D_{g,i,j,k}+h_{yi,j,k}D_{g,i,j+1,k}}-\frac{2D_{g,i,j-1,k}\left(\varphi_{g,i,j,k}-\varphi_{g,i,j-1,k}\right)}{h_{yi,j,k}D_{g,i,j-1,k}+h_{yi,j-1,k}D_{g,i,j,k}}\right] \tag{4.4b}$$

$$\left[L_z\Phi\right]_{g,i,j,k} \equiv \frac{D_{g,i,j,k}}{h_{zi,j,k}}\left[\frac{2D_{g,i,j,k+1}(\varphi_{i,j,k+1}-\varphi_{g,i,j,k})}{h_{zi,j,k+1}D_{g,i,j,k}+h_{zi,j,k}D_{g,i,j,k+1}}-\frac{2D_{g,i,j,k-1}\left(\varphi_{g,i,j,k}-\varphi_{g,i,j,k-1}\right)}{h_{zi,j,k}D_{g,i,j,k-1}+h_{zi,j,k-1}D_{g,i,j,k}}\right] \tag{4.4c}$$

The HFEM method proceeds by dividing the volume of the reactor into large parallelepiped-shaped elements. The corners of the elements represent the nodes. Each element is, in turn, subdivided into subregions, each subregion corresponding to one fine-mesh box of the finite-difference grid. Normally, an element would encompass a fuel assembly in the x-y plane and extend approximately 0.5m in the Z direction. A subregion would correspond to a homogenized fuel-pin cell (pin plus coolant) and would only extend a few cm in the Z direction. The actual choice of elements and subregions is flexible and depends on the configuration being analyzed. The energy is divided into coarse groups, indexed by G. Coarse groups are, in turn, subdivided into fine groups indexed by g. The fine-group structure is the same as in the finite-difference discretization. A simple representation of the core division in elements and subregions is shown in Fig. (4.1).

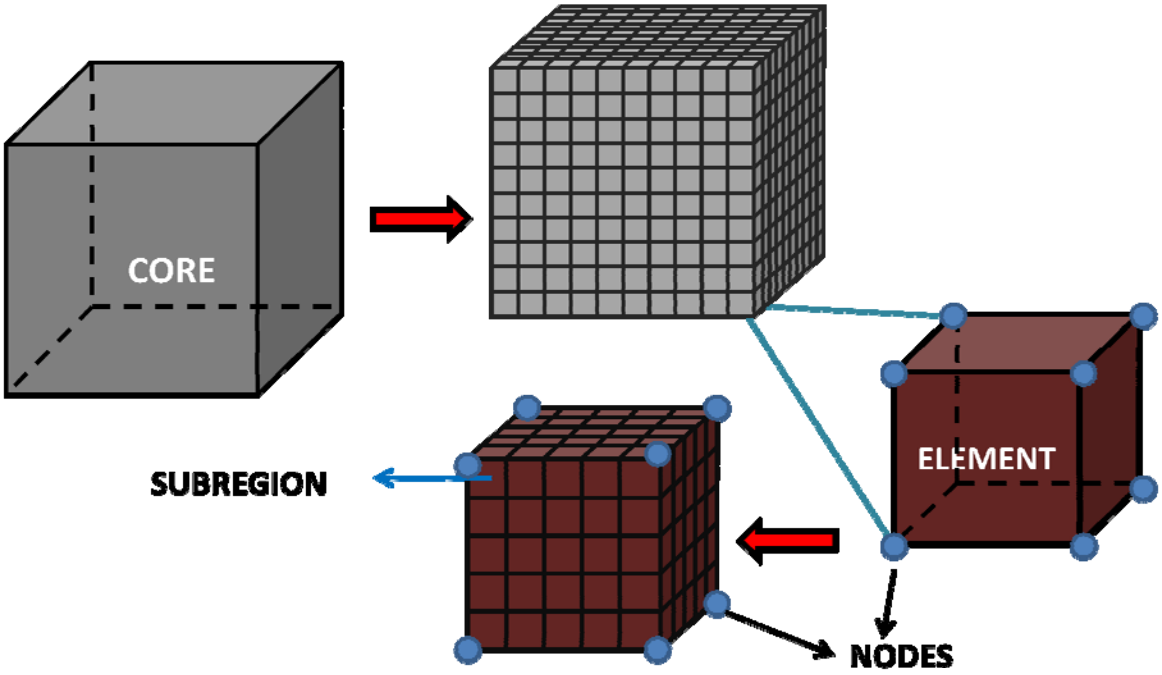


Figure 4.1: Elements, nodes and subregions for a three-dimensional geometry

The solution of the discretized multi-group diffusion equations is sought as a linear combination of discrete basis (trial) functions of space and energy:

$$\Phi(g,i,j,k)=\sum_{G=1}^{N_G}\sum_{n=1}^{Nn}a_{n,G}\psi_{n,G}(g,i,j,k) \qquad \textbf{(4.5)}$$

Where $\psi_{n,G}(g,i,j,k)$ is the (discrete) basis function corresponding to node n for coarse energy-group G and $a_{n,G}$ are the corresponding expansion coefficients which are to be determined.

Substituting this type of solution into the initial finite-difference-discretized diffusion equation in operator form, we obtain:

$$\sum_{G=1}^{N_G}\sum_{n=1}^{Nn} a_{n,G} M\psi_{n,G} = \frac{1}{k}\sum_{G=1}^{N_G}\sum_{n=1}^{Nn} a_{n,G} F\psi_{n,G} \qquad \textbf{(4.6)}$$

Next, just as for the regular finite element method, we take the inner product of Eq. (4.6) with weight functions $w_{m,G}(g,i,j,k)$ which, for the development of the current method, are selected to be the same as the basis functions $w_{m,H}(g,i,j,k) \equiv \psi_{m,H}(g,i,j,k)$.

The resulting linear system is:

$$\sum_{G=1}^{N_G}\sum_{n=1}^{Nn} a_{n,G} \left\langle \psi_{m,H}; M\psi_{n,G} \right\rangle = \frac{1}{k}\sum_{G=1}^{N_G}\sum_{n=1}^{Nn} a_{n,G} \left\langle \psi_{m,H}; F\psi_{n,G} \right\rangle \qquad \textbf{(4.7)}$$

where $\psi_{n,G}$ is the basis function corresponding to node n and coarse group G and $\psi_{m,H}$ is the weight function (same as the basis function) corresponding node m and coarse group H.

It can be seen that Eq. (4.7) represents a homogeneous linear system for which the number of unknowns, $a_{n,G}$, equals the number of equations and equals N_GxN_n. It represents a generalized eigenvalue eigenvector problem with eigenvalue $\frac{1}{k_{eff}}$ and an eigenvector defined by its components $a_{n,G}$. Because the system is homogeneous, its solution can only be determined up to a multiplicative constant.

For each node n and coarse group G, the basis function is defined as a sum of elementary basis functions $\varphi_{G,n,e}$, each of which is nonzero only in one of the elements surrounding the node.

$$\psi_{n,G}(g,i,j,k) = \sum_{e} \varphi_{G,n,e}(g,i,j,k) \tag{4.8}$$

Each elementary basis function $\varphi_{G,n,e}$ represents a "branch" of the basis function $\psi_{n,G}$ which is non-zero only over one element "e".

Just as for the regular FEM, each elementary basis function $\varphi_{G,n,e}$ is normalized such that it takes a value of one at its corresponding node and it vanishes at all the other nodes in element e. With this choice, the coefficients $a_{n,G}$ that are to be determined by solving the problem represent the amplitude of the flux in coarse group G, at each of the nodes n.

In general, the choice of basis (trial) functions depends on the problem that needs to be solved. The heterogeneous finite element method allows the basis functions to be chosen such that they capture the fine-mesh detail of the flux distribution. However, the number of unknowns is of the same order of magnitude as the number of nodes which are the corners of the coarse regions times the number of coarse groups, therefore much smaller than if a full-core fine-mesh fine-group solution was performed.

The heterogeneous finite element method thus offers fine-mesh and fine group detail at coarse-mesh coarse-group computational cost.

4.2 CHOICE OF BASIS FUNCTIONS

The sampling points for which the solution is obtained are called the nodes of the geometry which are the corners of the volumetric finite elements. There is one basis function per system node per coarse energy group and hence basis functions will be called “node basis functions”. Each node basis function is non-zero only over the (eight) elements neighbouring the corresponding node and vanishes on the outer boundary of these elements. Since a volumetric element has eight corners, the flux behaviour inside the element is represented by a linear combination of its eight corner nodes’ basis functions or, more precisely, the corresponding elementary basis functions. Each elementary basis function represents the restriction to one element of the (wider-domain) basis function. (In three dimensions, each basis function consists of eight elementary basis functions “glued” together at the node.)

In general, choosing the node basis functions that describe the system behaviour depends on (Hansen and Kang ,1975):

- The desired mathematical form of the function
- Continuity conditions
- The nature of the physical problem to be approximated
- Boundary and mesh interface conditions

The HFEM uses piecewise defined node basis functions that have the general behaviour of the sought-for solution. A depiction of a one-dimensional discrete basis function made up of two elementary basis functions (or "branches") is shown in Fig. (4.2).

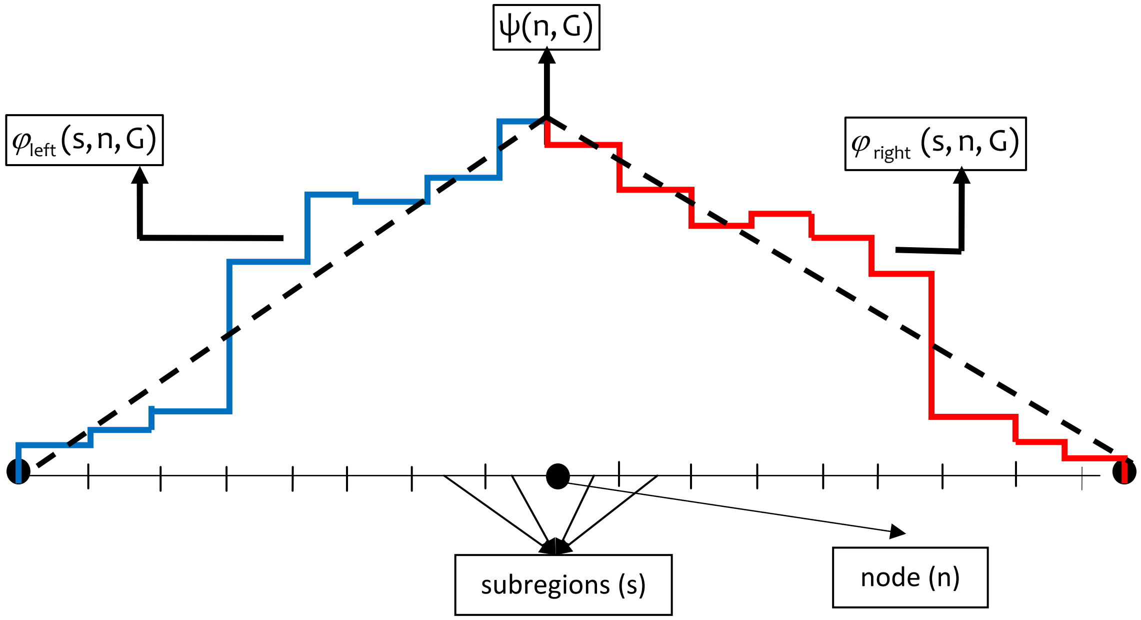


Figure 4.2 :Example of one-dimensional discrete basis function

In Fig. (4.2) the dashed line represents the basis function for the regular (homogeneous FEM) while the solid line represents the (discrete) basis function for the heterogeneous FEM. It can be seen that the basis function for the heterogeneous finite element method can be chosen to follow the intra-assembly flux shape.

The advantages of using piecewise-defined trial functions:

- Because of their fine-mesh detail, they provide a good local approximation.
- It is very easy to satisfy interface conditions.
- Because they vanish outside of elements that do not include the node, they yield a sparse algebraic system which is easier to solve compared to a full-matrix system.
- They can be chosen with great flexibility and adapted to best suit the accuracy and desired computational cost of each problem.

The linear system arising from using piecewise expansion functions in Eq. (4.7) is sparse and, in fact, similar in structure to the finite difference system of equations (only with fewer unknowns).

Because the basis functions take a value of 1.00 at one node and vanish on all outer borders of their eight-node domain, flux continuity is implicitly satisfied throughout the system.

Due to the discrete nature of the flux approximation (The basis functions are assumed to be constant in each subregion.) all integrations turn to summations. In particular, the inner product of any two discrete functions u and v depending on both energy and position is now defined as:

$$\langle u; v \rangle \equiv \sum_{g} \sum_{i,j,k} u(g,i,j,k)v(g,i,j,k)h_{xi,j,k}h_{yi,j,k}h_{zi,j,k} \tag{4.9}$$

where $h_{xi,j,k}, h_{yi,j,k}, h_{zi,j,k}$ are the dimensions of the fine-mesh box (i,j,k).

It will be noted that the inner product used for the HFEM encompasses energy. This is different from the regular FEM where the inner product is only defined over spatial variables.

Since the node basis functions carry the pin-level detail, assembly homogenization is not necessary for the HFEM. Pin-cell homogenization is sufficient.

As discussed previously, different forms of the elementary basis functions can be employed (to suit the problem at hand) because the method is flexible-enough to allow arbitrary shapes. For a first implementation of the method, a discrete tri-linear form was chosen. While this does not make full use of the method's ability to reproduce rapidly-varying flux shapes, it has the advantage of allowing easy comparison with continuous, homogeneous, finite element methods for the purpose of verifying the correctness of the algorithm. Figs. (4.3) and (4.4) show the discrete linear basis function in one-dimensional geometry and the bi-linear basis function in a two-dimensional geometry respectively.

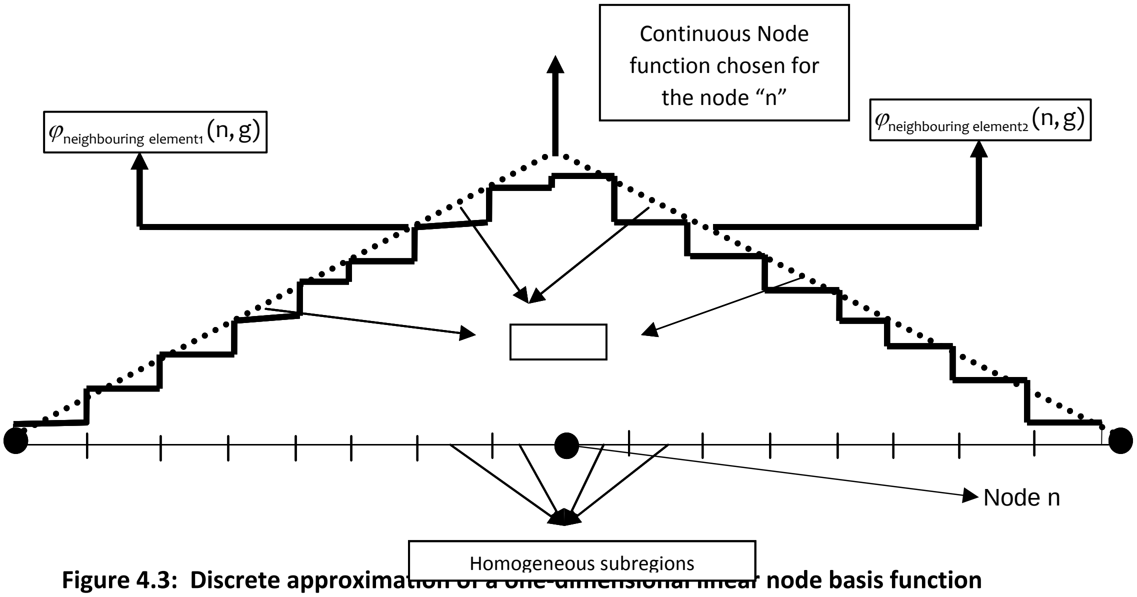

Continuous Node function chosen for the node “n”
$\varphi_{\text{neighbouring element1}}(n, g)$
$\varphi_{\text{neighbouring element2}}(n, g)$
Node n
Homogeneous subregions

Figure 4.3: Discrete approximation of a one-dimensional linear node basis function

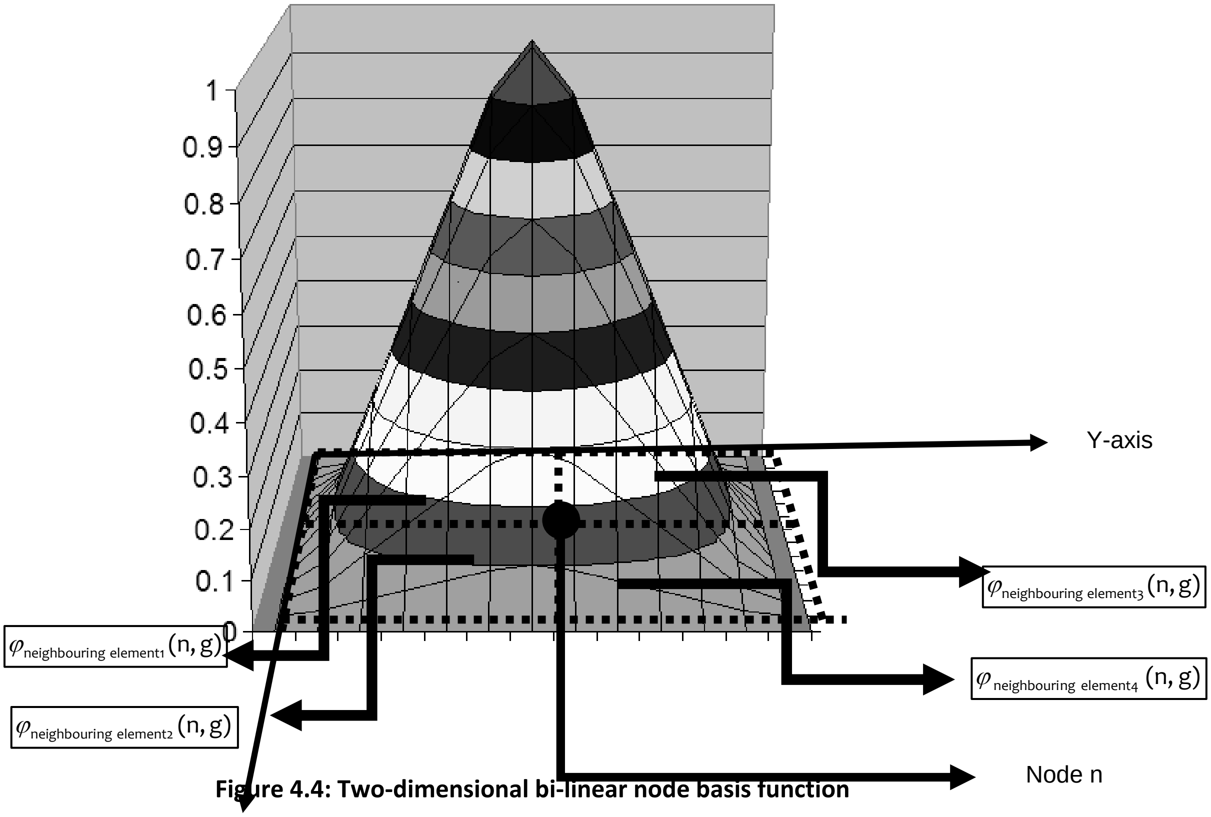


Figure 4.4: Two-dimensional bi-linear node basis function

4.3 DEGREES OF FREEDOM OF THE HFEM SYSTEM

In finite element analysis, the number of degrees of freedom (DOF) represent the number of unknowns in the system. According to Eq. (4.7) the number of unknowns, $a_{n,G}$, in the HFEM system is equal to the product between the number of nodes and the number of coarse groups, that is $N_G x N_n$. Similarly, Eq. (4.1) shows that the number of unknowns $\varphi_{g,i,j,k}$ for the fine-mesh fine-group FD system is equal to the product between the number of subregions and the number of fine groups $N_g x N_s$. For large systems, the number of nodes approximately equals the number of elements. That is because, with the exception of boundaries, there are eight nodes for each element and each node has eight neighbouring elements. If we also assume, for simplicity, a constant number of S subregions per element, and a number R of fine energy groups per coarse energy group, the number of degrees of freedom for the HFEM method is written as:

$$DOF_{HFEM} = N_G \times N_n = \frac{N_g}{R}\frac{N_s}{S} = \frac{1}{RS} DOF_{FD} \tag{4.10}$$

It follows that the number of unknowns is by a factor of RS (number of fine groups per coarse group times number of subregions per element) when going from FD to HFEM.

4.3 DETAILED EXPRESSIONS OF THE HFEM SYSTEM COEFFICIENTS

The HFEM equations are written as:

$$\sum_{G=1}^{N_G}\sum_{n=1}^{Nn} a_{n,G}\left\langle \psi_{m,H};\mathrm{M}\psi_{n,G}\right\rangle = \frac{1}{k}\sum_{G=1}^{N_G}\sum_{n=1}^{Nn} a_{n,G}\left\langle \psi_{m,H};\mathrm{F}\psi_{n,G}\right\rangle \tag{4.11}$$

which can be abbreviated as:

$$\sum_{G=1}^{N_G}\sum_{n=1}^{Nn} a_{n,G}\mathrm{M}_{m,n}^{H,G} = \frac{1}{k}\sum_{G=1}^{N_G}\sum_{n=1}^{Nn} a_{n,G}\mathrm{F}_{m,n}^{H,G} \tag{4.12}$$

The detailed expressions for operators M and F are:

$$\mathrm{F}_{m,n}^{H,G} = \left\langle \psi_{m,H}(g,i,j,k);_{g,i,j,k} \sum_{g'=1}^{N_g} \nu\Sigma_{fg',i,j,k}\psi_{n,G}(g',i,j,k) \right\rangle \tag{4.13}$$

$$\mathrm{M}_{m,n}^{H,G} = L_{m,n}^{H,G} + T_{m,n}^{H,G} + S_{m,n}^{H,G} \tag{4.14}$$

Where L T and S are defined as:

$$\begin{aligned}
&\mathrm{L}_{m,n}^{H,G} = \left\langle \psi_{m,H}(g,i,j,k); -\left[L_x \psi_{n,G}\right](g,i,j,k) \right\rangle + \\
&\left\langle \psi_{m,H}(g,i,j,k); -\left[L_y \psi_{n,G}\right](g,i,j,k) \right\rangle + \\
&\left\langle \psi_{m,H}(g,i,j,k); -\left[L_z \psi_{n,G}\right](g,i,j,k) \right\rangle = \\
&\left\langle \psi_{m,H}(g,i,j,k); \frac{D_{g,i,j,k}}{h_{xi,j,k}} \left[\frac{2D_{g,i+1,j,k}\left(\psi_{n,G}(g,i+1,j,k) - \psi_{n,G}(g,i,j,k)\right)}{h_{xi+1,j,k} D_{g,i,j,k} + h_{xi,j,k} D_{g,i+1,j,k}} - \frac{2D_{g,i-1,j,k}\left(\psi_{n,G}(g,i,j,k) - \psi_{n,G}(g,i-1,j,k)\right)}{h_{xi,j,k} D_{g,i-1,j,k} + h_{xi-1,j,k} D_{g,i,j,k}} \right] \right\rangle + \\
&\left\langle \psi_{m,H}(g,i,j,k); \frac{D_{g,i,j,k}}{h_{yi,j,k}} \left[\frac{2D_{g,i,j+1,k}\left(\psi_{n,G}(g,i,j+1,k) - \psi_{n,G}(g,i,j,k)\right)}{h_{yi,j+1,k} D_{g,i,j,k} + h_{yi,j,k} D_{g,i,j+1,k}} - \frac{2D_{g,i,j-1,k}\left(\psi_{n,G}(g,i,j,k) - \psi_{n,G}(g,i,j-1,k)\right)}{h_{yi,j,k} D_{g,i,j-1,k} + h_{yi,j-1,k} D_{g,i,j,k}} \right] \right\rangle + \\
&\left\langle \psi_{m,H}(g,i,j,k); \frac{D_{g,i,j,k}}{h_{zi,j,k}} \left[\frac{2D_{g,i,j,k+1}\left(\psi_{n,G}(g,i,j,k+1) - \psi_{n,G}(g,i,j,k)\right)}{h_{zi,j,k+1} D_{g,i,j,k} + h_{zi,j,k} D_{g,i,j,k+1}} - \frac{2D_{g,i,j,k-1}\left(\psi_{n,G}(g,i,j,k) - \psi_{n,G}(g,i,j,k-1)\right)}{h_{yi,j,k} D_{g,i,j,k-1} + h_{xi,j,k} D_{g,i,j,k-1}} \right] \right\rangle
\end{aligned} \tag{4.15}$$

$$\mathrm{S}_{m,n}^{H,G} = \left\langle \psi_{m,H}(g,i,j,k); -\sum_{g'=1}^{N_g} \Sigma_{sg'\rightarrow g,i,j,k} \psi_{n,G}(g',i,j,k) \right\rangle \tag{4.16}$$

$$\mathrm{T}_{m,n}^{H,G} = \left\langle \psi_{m,H}(g,i,j,k); +\Sigma_{tg,i,j,k} \psi_{n,G}(g,i,j,k) \right\rangle \tag{4.17}$$

Coefficients $c_{n,G}$ define a block column vector $\underline{c}$:

$$\underline{c} = \begin{bmatrix} c_{1,1} \\ \vdots \\ c_{N_n,1} \\ \vdots \\ c_{1,G} \\ \vdots \\ c_{N_n,N_G} \end{bmatrix} \tag{4.18}$$

where nodes are swept first and (coarse) groups are swept second, such that each block corresponds to a (coarse) group G.

Consequently, each operator in Eqs. (4.13) and (4.14) will have a corresponding block matrix of the general form:

$$\underline{\underline{A}} = \begin{bmatrix} a^{1,1}_{1,1} & \cdots & a^{1,1}_{1,N_n} & \cdots & a^{1,N_G}_{1,1} & \cdots & a^{1,N_G}_{1,N_n} \\ \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ a^{1,1}_{N_n,1} & \cdots & a^{1,1}_{N_n,N_n} & \cdots & a^{1,N_G}_{N_n,1} & \cdots & a^{1,N_G}_{N_n,N_n} \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ a^{N_G,1}_{1,1} & \cdots & a^{N_G,1}_{1,N_n} & \cdots & a^{N_G,N_G}_{1,1} & \cdots & a^{N_G,N_G}_{1,N_n} \\ \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ a^{N_G,1}_{N_n,1} & \cdots & a^{N_G,1}_{N_n,N_n} & \cdots & a^{N_G,N_G}_{N_n,1} & \cdots & a^{N_G,N_G}_{N_n,N_n} \end{bmatrix} \tag{4.19}$$

where each block corresponds to a pair of (coarse) energy groups (H,G).

In particular, we have:

$$\underline{\underline{L}} = \begin{bmatrix} L^{1,1}_{1,1} & \cdots & L^{1,1}_{1,N_n} & \cdots & 0 & \cdots & 0 \\ \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ L^{1,1}_{N_n,1} & \cdots & L^{1,1}_{N_n,N_n} & \cdots & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ 0 & \cdots & 0 & \cdots & L^{N_G,N_G}_{1,1} & \cdots & L^{N_G,N_G}_{1,N_n} \\ \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & \cdots & 0 & \cdots & L^{N_G,N_G}_{N_n,1} & \cdots & L^{N_G,N_G}_{N_n,N_n} \end{bmatrix} \tag{4.20}$$

$$\underline{\underline{S}} = \begin{bmatrix} S^{1,1}_{1,1} & \cdots & 0 & \cdots & S^{1,N_G}_{1,1} & \cdots & 0 \\ \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & \cdots & S^{1,1}_{N_n,N_n} & \cdots & 0 & \cdots & S^{1,N_G}_{N_n,N_n} \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ S^{N_G,1}_{1,1} & \cdots & 0 & \cdots & S^{N_G,N_G}_{1,1} & \cdots & 0 \\ \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & \cdots & S^{N_G,1}_{N_n,N_n} & \cdots & 0 & \cdots & S^{N_G,N_G}_{N_n,N_n} \end{bmatrix} \tag{4.21}$$

$$\underline{\underline{T}} = \begin{bmatrix} T^{1,1}_{1,1} & \cdots & 0 & \cdots & 0 & \cdots & 0 \\ \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & \cdots & T^{1,1}_{N_n,N_n} & \cdots & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ 0 & \cdots & 0 & \cdots & T^{N_G,N_G}_{1,1} & \cdots & 0 \\ \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & \cdots & 0 & \cdots & 0 & \cdots & T^{N_G,N_G}_{N_n,N_n} \end{bmatrix} \tag{4.22}$$

$$\underline{\underline{F}} = \begin{bmatrix} F^{1,1}_{1,1} & \cdots & 0 & \cdots & F^{1,N_G}_{1,1} & \cdots & 0 \\ \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & \cdots & F^{1,1}_{N_n,N_n} & \cdots & 0 & \cdots & F^{1,N_G}_{N_n,N_n} \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ F^{N_G,1}_{1,1} & \cdots & 0 & \cdots & a^{N_G,N_G}_{1,1} & \cdots & 0 \\ \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & \cdots & F^{N_G,1}_{N_n,N_n} & \cdots & 0 & \cdots & F^{N_G,N_G}_{N_n,N_n} \end{bmatrix} \tag{4.23}$$

The HFEM system can then be written in matrix form as:

$$\underline{\underline{M}}\underline{c} = \frac{1}{k}\underline{\underline{F}}\underline{c} \Leftrightarrow \underline{\underline{L}}\underline{c} + \underline{\underline{T}}\underline{c} - \underline{\underline{S}}\underline{c} = \frac{1}{k}\underline{\underline{F}}\underline{c} \tag{4.24}$$

For reasons that will become clearer later, it is convenient to introduce a new matrix P, as the sum of the leakage and total interaction matrices:

$$\underline{\underline{P}} = \underline{\underline{L}} + \underline{\underline{T}} \tag{4.25}$$

Because matrix T is diagonal, matrix P has the same block diagonal structure as matrix L, that is:

$$\underline{\underline{P}} = \begin{bmatrix} P^{1,1}_{1,1} & \cdots & P^{1,1}_{1,N_n} & \cdots & 0 & \cdots & 0 \\ \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ P^{1,1}_{N_n,1} & \cdots & P^{1,1}_{N_n,N_n} & \cdots & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ 0 & \cdots & 0 & \cdots & P^{N_G,N_G}_{1,1} & \cdots & P^{N_G,N_G}_{1,N_n} \\ \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & \cdots & 0 & \cdots & P^{N_G,N_G}_{N_n,1} & \cdots & P^{N_G,N_G}_{N_n,N_n} \end{bmatrix} \tag{4.26}$$

With the new notation, the system of equations can be written as:

$$\underline{\underline{P}}\,\underline{c} - \underline{\underline{S}}\,\underline{c} = \frac{1}{k}\underline{\underline{F}}\,\underline{c} \tag{4.27}$$

4.4 HFEM SYSTEM SOLUTION

The system is solved using a system of nested iterations (Lewis and Miller,1984). The eigenvalue problem is solved using inverse power iteration:

$$\left(\underline{\underline{P}} - \underline{\underline{S}}\right)\underline{c}^{n} = \frac{1}{k_{n-1}}\underline{\underline{F}}\,\underline{c}^{n-1} \tag{4.28}$$

Where k_{n-1} can be calculated as:

$$k_{n-1} = \frac{\left(\underline{c}^{n-1}\right)^{T}\underline{\underline{F}}\,\underline{c}^{n-1}}{\left(\underline{c}^{n-1}\right)^{T}\left(\underline{\underline{P}} - \underline{\underline{S}}\right)\underline{c}^{n-1}} \tag{4.29}$$

or, simply, as:

$$k_{n-1} = k_{i-2}\,\frac{(1)^{T}\,\underline{c}^{n-1}}{(1)^{T}\,\underline{c}^{n-2}} \tag{4.30}$$

where “1” is the unit column vector and the “T” superscript symbolizes the transpose. Solving the linear system expressed by Eq. (4.27) is difficult because of the large size of the matrix involved. To reduce the size of the matrix, it would be convenient to solve the system group by group. However, that is not possible because matrix $\underline{\underline{M}} = \left(\underline{\underline{P}} - \underline{\underline{S}}\right)$ is not block diagonal. To avoid this difficulty, Lewis and Miller (1984) were followed in introducing a level of iteration over the scattering source:

$$\underline{\underline{P}}\,\underline{c}_m^n = \frac{1}{k}\underline{\underline{F}}\,\underline{c}^{n-1} + \underline{\underline{S}}\,\underline{c}_{m-1}^n \tag{4.31}$$

The process is a Neumann iteration which is known to converge (Lewis and Miller,1984).

At each scattering-source iteration, we are still left with the need to solve a linear system of the type:

$$\underline{\underline{P}}\,\underline{c}_m^n = \underline{RHS} \tag{4.32}$$

where the right-hand side is:

$$\underline{RHS} = \frac{1}{k}\underline{\underline{F}}\,\underline{c}^{n-1} + \underline{\underline{S}}\,\underline{c}_{m-1}^n \tag{4.33}$$

However, given the fact that matrix P is block diagonal, we can solve the Eq. (4.33) for each group separately, namely solving, in sequence:

$$\begin{bmatrix} P_{1,1}^{G,G} & \cdots & P_{1,N_n}^{G,G} \\ \vdots & \ddots & \vdots \\ P_{N_n,1}^{G,G} & \cdots & P_{N_n,N_n}^{G,G} \end{bmatrix} \begin{bmatrix} c_{1,G} \\ \vdots \\ c_{N_n,G} \end{bmatrix} = \begin{bmatrix} RHS_{1,G} \\ \vdots \\ RHS_{N_n,G} \end{bmatrix} \tag{4.34}$$

for coarse groups 1 to N_G. In this case the number of unknowns is reduced to N_n for each group G.

It will be noted that because of the properties of the basis functions, the spatial matrix P^{GG} is sparse, which makes iterative methods the preferred approach for solving the system.

Two separate iterative methods were implemented for the solution of system (Eq. 4.34): The Gauss-Seidel method (Saad, 2000; Barrett, Berry, Chan, Demmel, Donato, Dongarra, Eijkhout, Pozo, Romine, Vorst,2008) and the Orthomin method (Greenbaum,1997 ; Abe and Zhang,2006). Gauss Seidel has been historically used in reactor-physics applications and has proven to be robust. However, lately (Modak, 2006) has found the Orthomin to also work very well in eigenvalue problems applied to reactor physics. The choice of which method to use is thus left to the user and depends of the particulars of the problem being studied. The Gauss-Seidel method is known to always (although not only) converge whenever the matrix is diagonal dominant. For the HFEM, the Gauss-Seidel method has been found to converge, but the matrix has not been proven theoretically to always be diagonal dominant so it was considered prudent to provide the user with a second solution method, i.e. Orthomin. Between the two, it is expected that most realistic problems can be solved.

The iteration process starts from a "guess" vector $\underline{c}_0$, which is usually taken to be the column unit vector, $\underline{c}_0 = \underline{1}$. The converged solution does not depend on the choice of initial vector (Lewis and Miller, 1984).

The k-eff value is kept constant during the scattering-source iterations. Convergence is reached when the change in the coefficient vector c from one iteration to the next becomes smaller than a user-defined convergence criterion (tolerance). Separate convergence criteria have to be specified for the system solution, the scattering iterations and the inverse power iterations. Usually a value of 1x10^{-5} relative change in solution from one iteration to the next is considered acceptable .

The flowchart of the three levels of iteration is shown in Fig. (4.5). Additional implementation details are given in Appendix B.

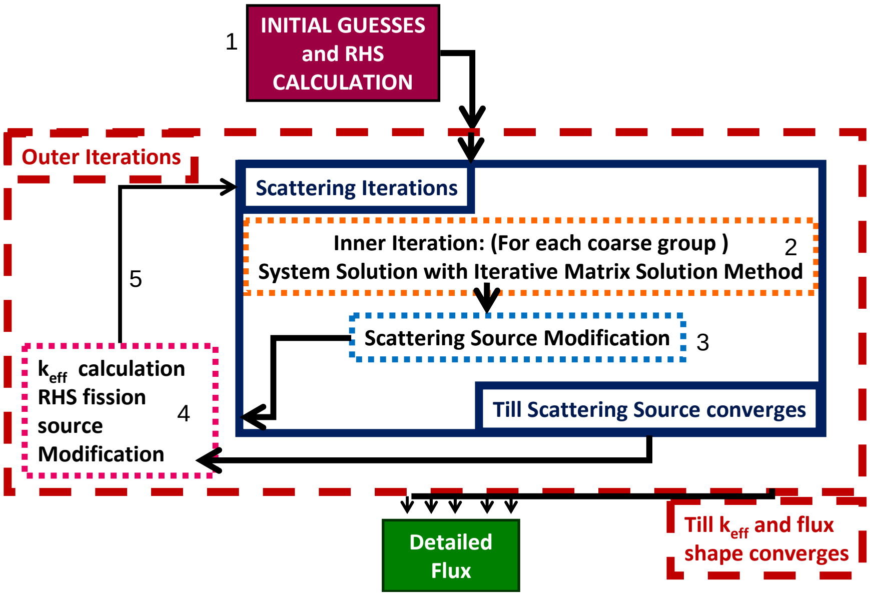

1
INITIAL GUESSES and RHS CALCULATION
Outer Iterations
Scattering Iterations
Inner Iteration: (For each coarse group)
System Solution with Iterative Matrix Solution Method
2
5
Scattering Source Modification
3
k_{eff} calculation
RHS fission source Modification
4
Till Scattering Source converges
Detailed Flux
Till k_{eff} and flux shape converges

Figure 4.5:Flowchart of the iterative calculation procedure

CHAPTER 5

CALCULATIONS AND RESULTS

5.1 TEST SYSTEM DESCRIPTION

5.1.1 System Geometry

A three-dimensional heterogeneous core model was used to test the method that has been developed. The system consists of 10 x 10 x 10 heterogeneous assemblies (or "elements"). Each assembly consists, in turn, of 5 x 5 x 5 homogeneous subregions, or unit cells. The dimensions of each assembly are 40 cm x 50 cm x 60 cm and those of each subregion are, consequently, 8 cm x 10 cm x 12 cm. The overall size of the model is close to that of a CANDU core. The system geometry is shown in Fig. (5.1).

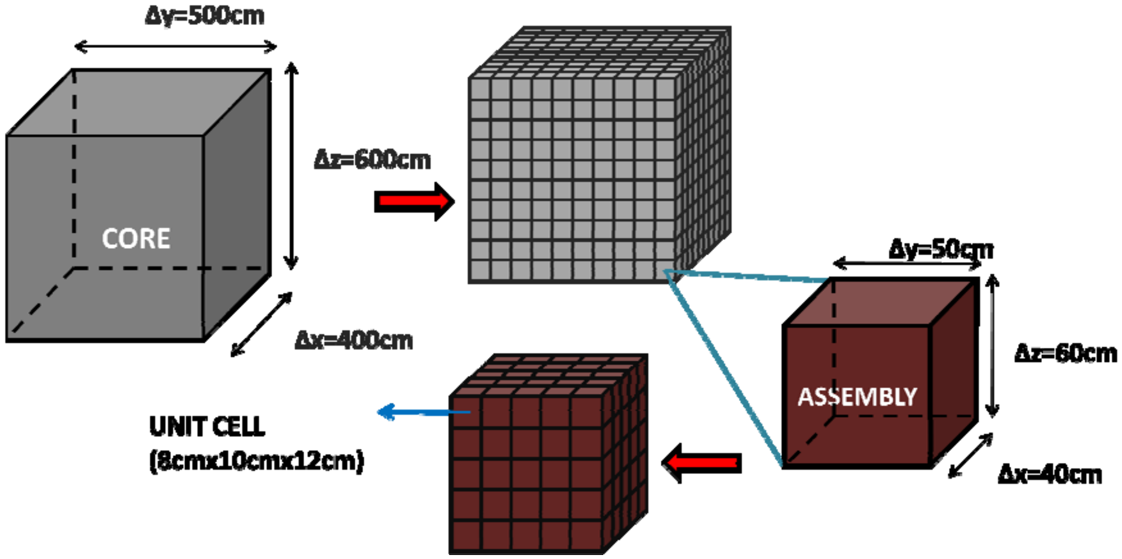


Figure 5.1: Test system geometry

5.1.2 Material Map

Three types of homogenized materials were used in the test model, corresponding to a fresh-fuel CANDU cell, a mid-burnup-fuel CANDU cell and discharge-burnup-fuel CANDU cell. In this simple test model, each assembly consists of a single material of either type 1 (fresh), type 2 (mid-burnup) or type 3 (discharge-burnup). The material distribution is depicted in Fig. (5.2) by means of an X-Y cross-section and a Y-Z cross section, both passing through the center of the system.

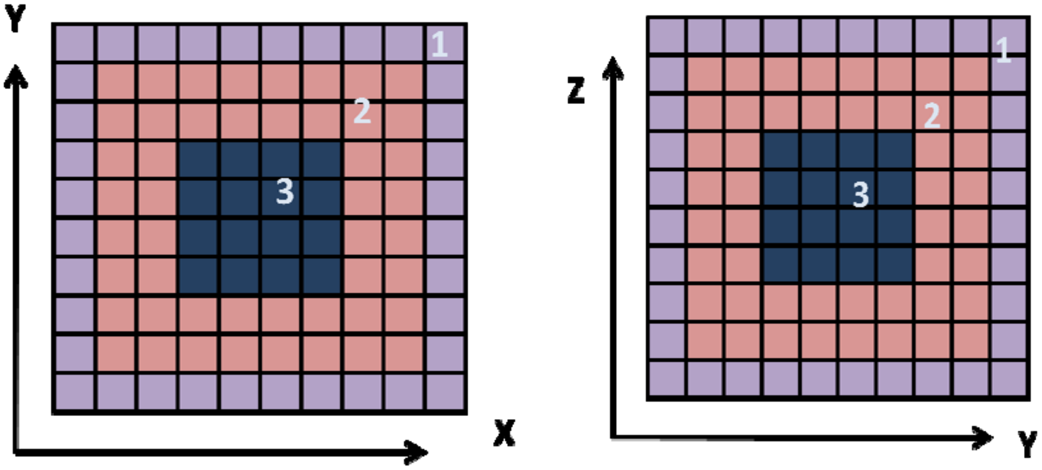


Figure 5.2: Material distribution in the test model

5.1.3 Material Properties

Three-energy group material properties were used, as shown in Table (5.1). The group boundaries are: E_0=1.0E7 eV > E_1 =755.0 eV, > E_2=0.625 eV > E_3=1.0E-6 eV.

Table 5.1: Three-group material properties

	Material Type 1 (Fresh Fuel)	Material Type 2 (Mid-burnup Fuel)	Material Type 3 (Discharge-Burnup Fuel)
D_1 (cm)	1.3410	1.34063	1.34036
D_2(cm)	1.2838	1.28077	1.27937
D_3(cm)	0.8454	0.84360	0.84334
$\Sigma_t^1(cm^{-1})$	0.3060	0.30607	0.30611
$\Sigma_t^2(cm^{-1})$	0.33635	0.33724	0.33763
$\Sigma_t^3(cm^{-1})$	0.43899	0.44014	0.4403
$\Sigma_s^{1\to1}(cm^{-1})$	0.2920E	2.92E-01	2.92E-01
$\Sigma_s^{1\to2}(cm^{-1})$	0.01280	1.28E-02	1.27E-02
$\Sigma_s^{1\to3}(cm^{-1})$	7.54E-07	7.54E-07	7.36E-07
$\Sigma_s^{2\to1}(cm^{-1})$	0.00000	1.55E-14	1.78E-14
$\Sigma_s^{2\to2}(cm^{-1})$	0.30200	3.03E-01	3.03E-01
$\Sigma_s^{2\to3}(cm^{-1})$	0.03150	0.03120	0.03110
$\Sigma_s^{3\to1}(cm^{-1})$	0.00000	0.00000	0.00000
$\Sigma_s^{3\to2}(cm^{-1})$	6.85E-05	7.46E-05	7.47E-05
$\Sigma_s^{3\to3}(cm^{-1})$	0.43500	0.43600	0.43600
$\nu\Sigma_f^1(cm^{-1})$	9.76E-04	9.57E-04	9.45E-04
$\nu\Sigma_f^2(cm^{-1})$	6.68E-04	5.59E-04	5.06E-04
$\nu\Sigma_f^3(cm^{-1})$	4.50E-03	4.77E-03	4.62E-03
χ_1	0.999999583	0.999999583	0.999999583
χ_2	4.03E-07	4.03E-07	4.03E-07
χ_3	0.00000	0.00000	0.00000

5.2 CALCULATIONS

To test the accuracy of the new HFEM, three sets of calculations were performed.

SET 1: Finite-difference method (FDM). This calculation was performed using finite differences for three energy groups and fine meshes, with each fine mesh corresponding to an 8 cm x 10 cm x 12 cm subregion. This is taken as the reference case.

SET 2: Finite element method (FEM). This calculation was performed using the usual finite element method in three energy groups and treating each assembly (40 cm x 50 cm x 60 cm) as an element. For the current choice of (discrete tri-linear) basis functions in the HFEM, FEM results should be similar to HFEM (SET 3) results. Trilinear functions are described in detail in Appendix A.

SET 3 Heterogeneous finite element method (HFEM). This calculation was performed using the newly-implemented HFEM using two "coarse" energy groups (where coarse-energy group 1 spans the range of fine-energy groups 1 and 2) and treating each assembly (40 cm x 50 cm x 60 cm) as an element. Even though this method uses two coarse-energy groups it still calculates the three energy group flux.

At this initial stage of testing, it was considered important to assess the accuracy of the HFEM method compared to the fine-mesh FD method, as well as the ***relative*** reduction in computation time achieved by HFEM relative to the fine mesh FD method. To achieve both a fair comparison and an exact estimate of the accuracy, the same, very tight, convergence criterion was used for both methods, namely 1.0E-6. For the HFEM method, the same convergence criterion was used for all three levels of iteration. Moreover, to decrease the effect of round-off errors, double precision arithmetic was used in all calculations. While this conservative approach was expected to lead to fairly large computation times, it was considered appropriate for this stage of testing.

All calculations were performed using a 2.2 GHz AMD Turion Dual-Core processor.

To facilitate the comparison of results from the different sets of calculations, cumulative-flux axial profiles for each direction were calculated, together with their percent errors defined as:

$$\mathrm{PE}(\%) = \frac{\text{Method Calculated Flux - Reference Flux}}{\text{Reference Flux}} \mathrm{x}100 \tag{5.1}$$

The cumulative flux for direction x was defined as

$$\hat{\varphi}_g(\mathrm{i}) = \sum_{\mathrm{j,k}} \varphi_g(\mathrm{i,j,k}) h_{\mathrm{y,i,j,k}} h_{\mathrm{z,i,j,k}} \tag{5.2}$$

and cumulative fluxes for the y and z direction were defined analogously.

The root mean square percent error was calculated for each method as:

$$\mathrm{RMSE}(\%) = \frac{\sqrt{\frac{1}{\text{number of sampling points}} \sum_{\text{sampling points}} (\text{Mehod Calculated Flux - Reference Flux})^2}}{\text{maximum reference flux}} \times 100 \tag{5.3}$$

5.3 RESULTS

Results for the effective multiplication constant and the computation times are shown in Table (5.2).

Table 5.2: Effective multiplication constant results

METHOD	k-eff	difference in mk	Time (min)
FDM	0.99012	-	4300
FEM	0.98957	-0.56134	23
HFEM	0.98934	-0.79627	16

Results for the RMS percent error are shown in Table (5.3).

Table 5.3: RMS percent error results

	FEM RMSE%	HFEM RMSE%
Group 1-x direction	0.8100	0.3781
Group 1-y direction	1.2503	0.5085
Group 1-z direction	2.2893	0.9312
Group 2-x direction	0.8123	0.3665
Group 2-y direction	0.8443	0.4991
Group 2-z direction	2.2883	0.9211
Group 3-x direction	0.8781	0.4305
Group 3-y direction	1.3778	0.6298
Group 3-z direction	2.4931	1.1418
Overall	1.4493	0.6452

Axial-profile plots for the cumulative fluxes for each direction and for each fine group, together with their percent errors are shown in Figs. (5.3) to (5.20). Additional comparisons with FEM and coarse-mesh FD are included in Appendix C.

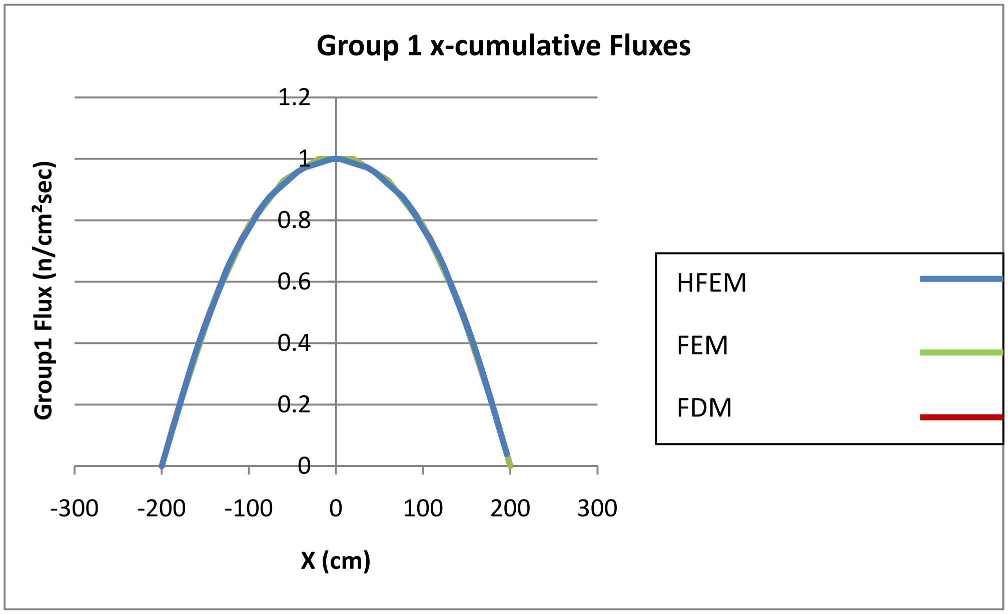


Figure 5.3: Group 1 Flux axial profile (X direction)

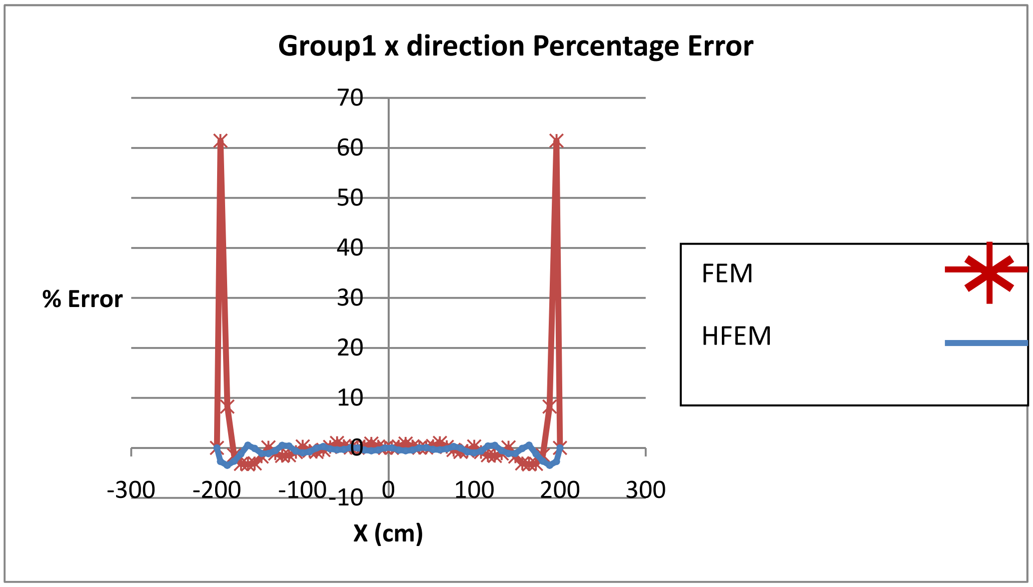


Figure 5.4: Group 1 Flux percent error axial profile (X direction)

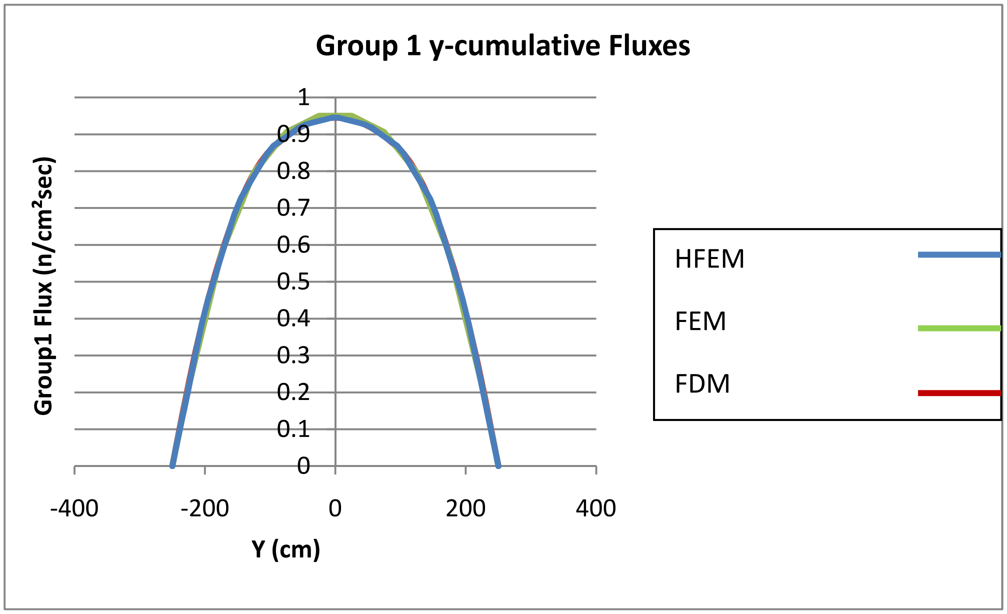


Figure 5.5: Group 1 Flux axial profile (Y direction)

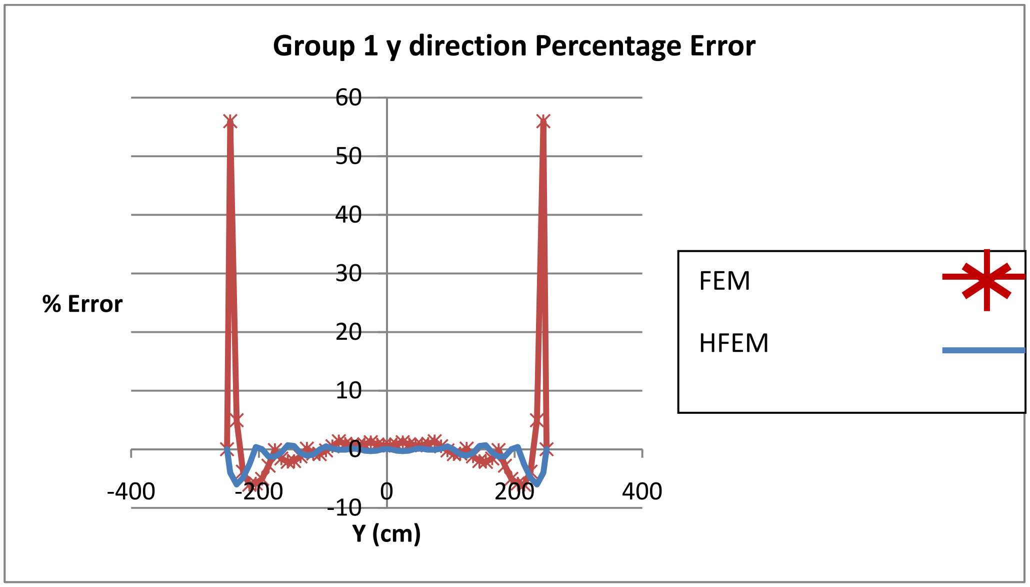


Figure 5.6: Group 1 Flux percent error axial profile (Y direction)

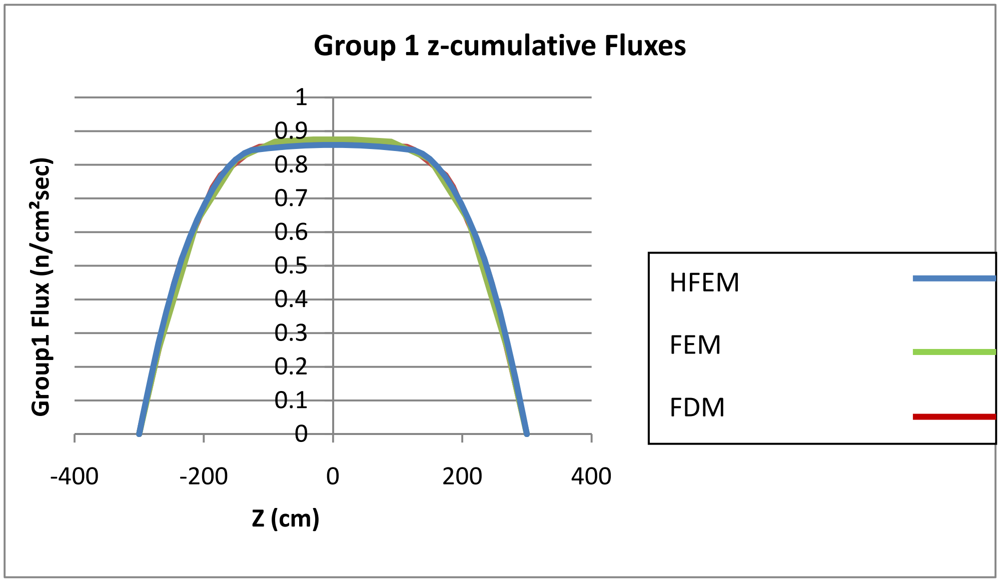


Figure 5.7: Group 1 Flux axial profile (Z direction)

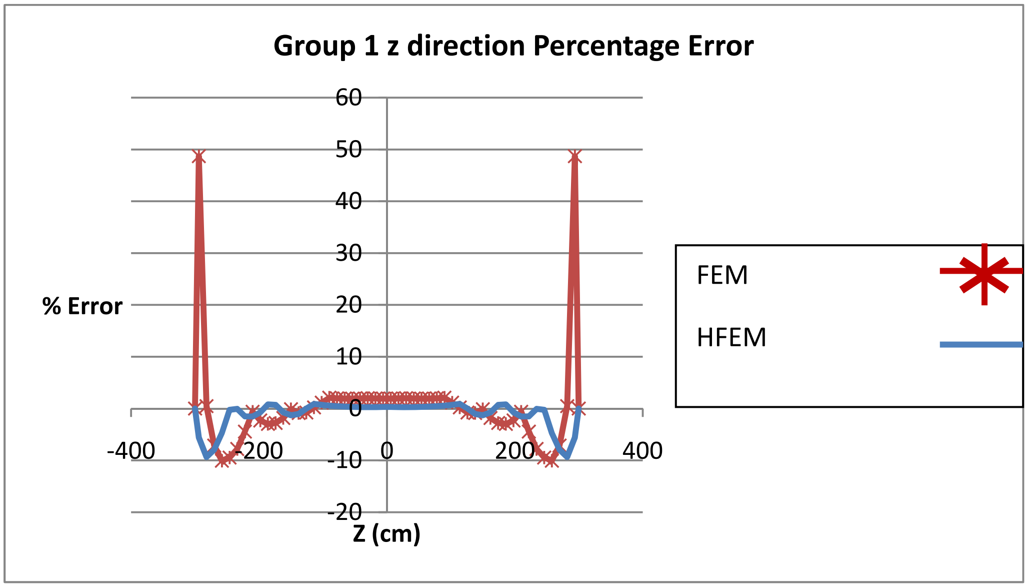


Figure 5.8: Group 1 Flux percent error axial profile (Z direction)

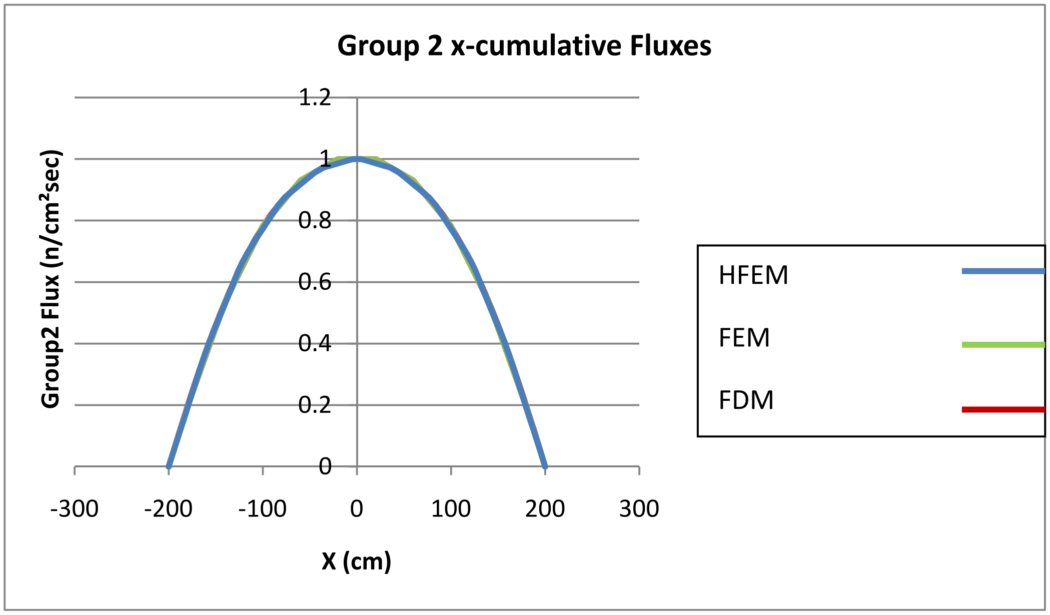


Figure 5.9: Group 2 Flux axial profile (X direction)

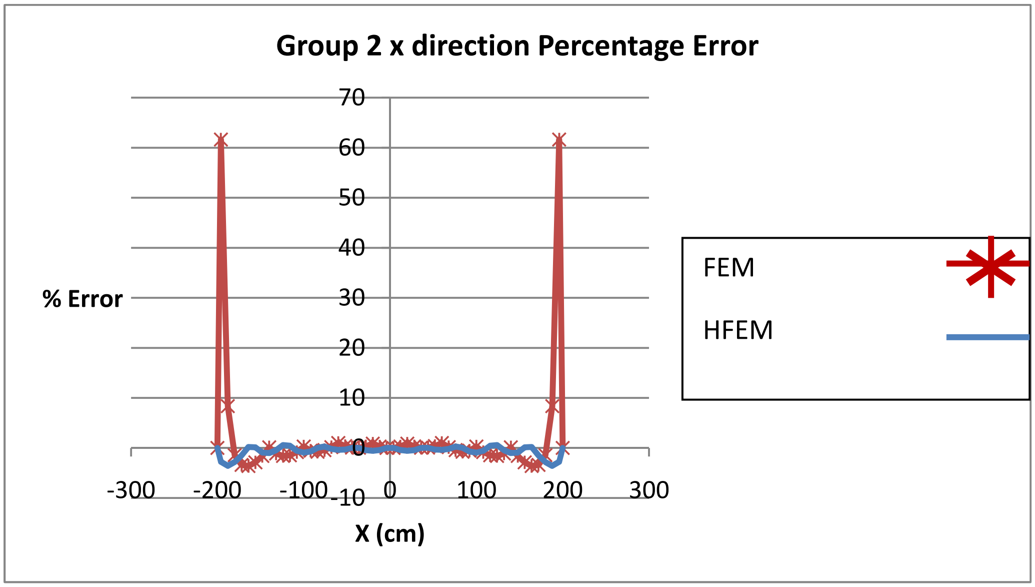


Figure 5.10: Group 2 Flux percent error axial profile (X direction)

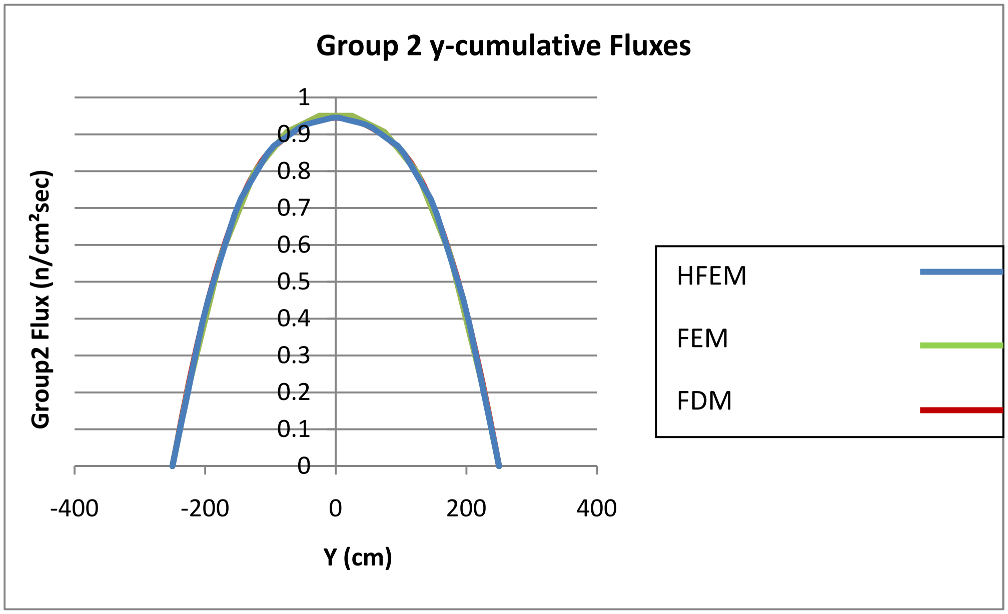


Figure 5.11: Group 2 Flux axial profile (Y direction)

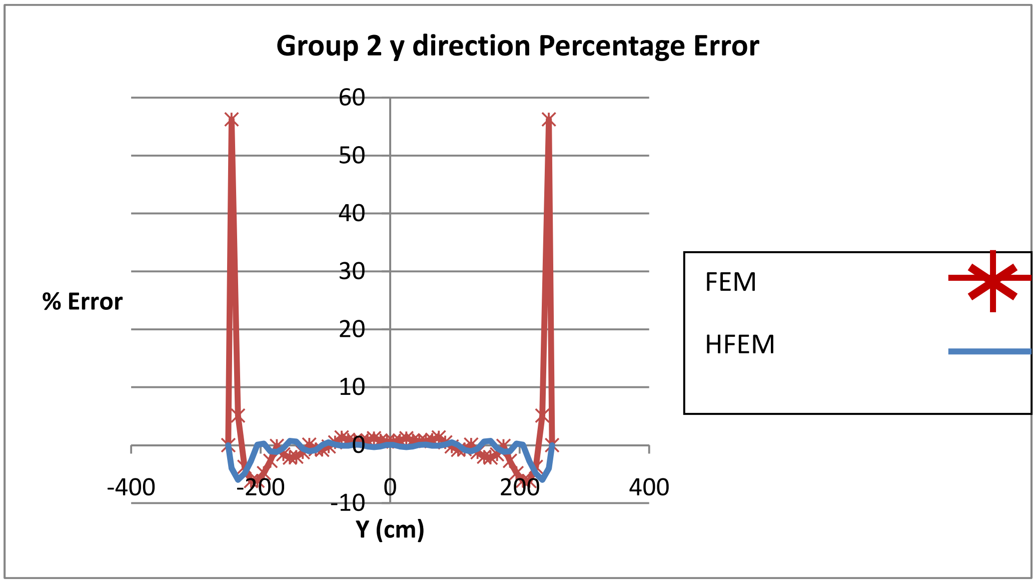


Figure 5.12: Group 2 Flux percent error axial profile (Y direction)

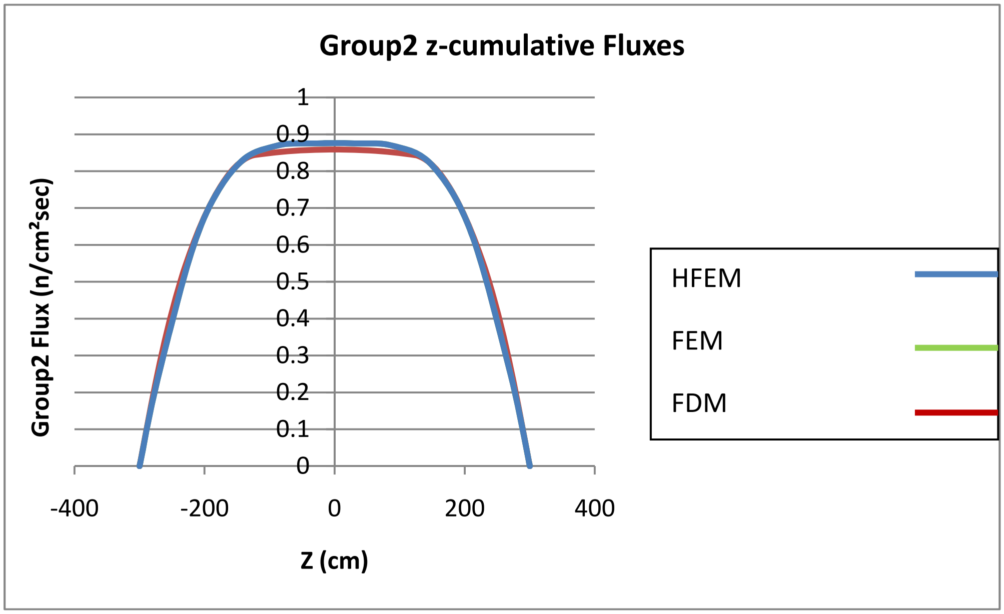


Figure 5.13: Group 2 Flux axial profile (Z direction)

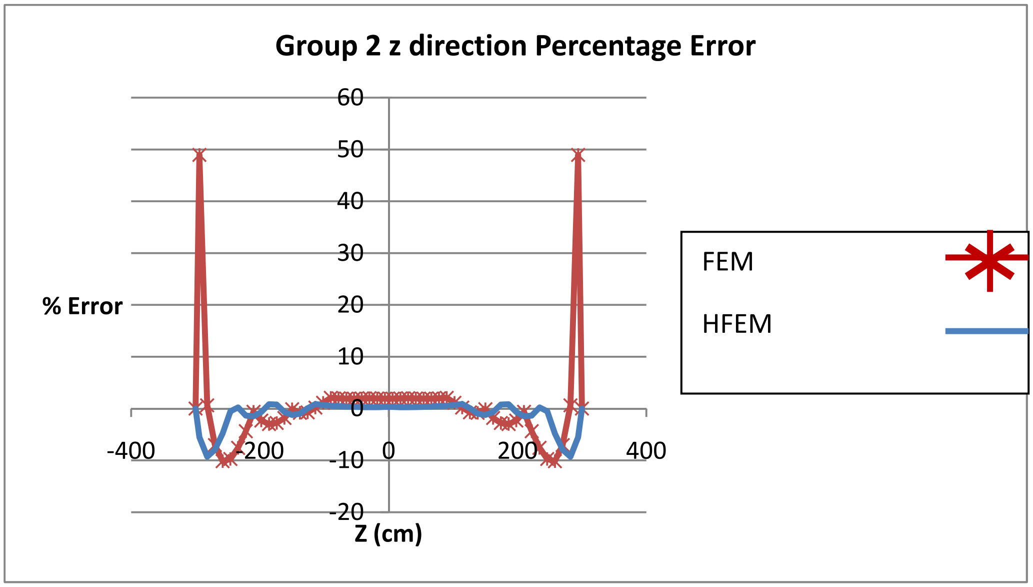


Figure 5.14: Group 2 Flux percent error axial profile (Z direction)

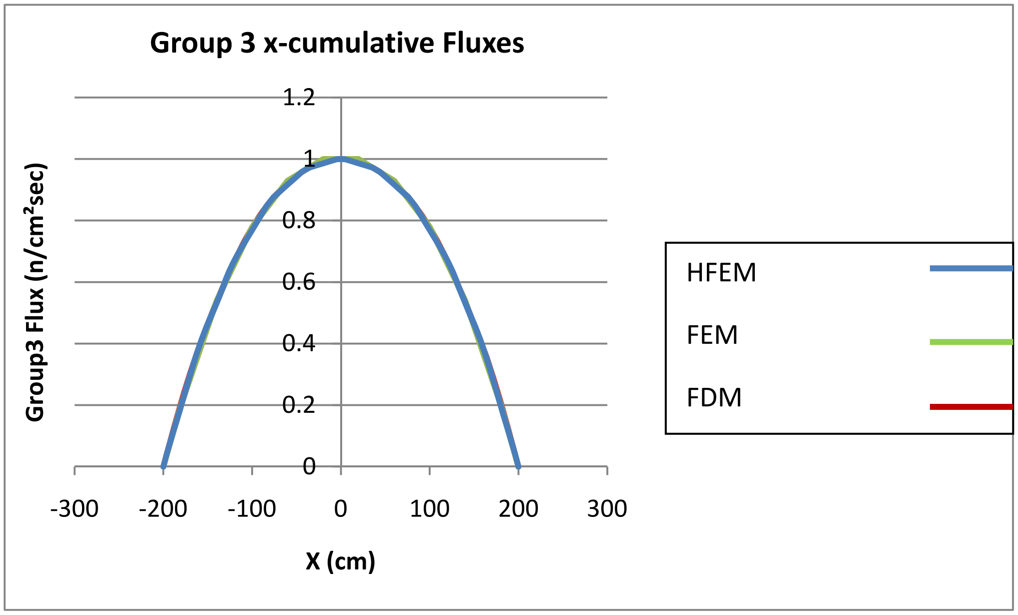


Figure 5.15: Group 3 Flux axial profile (X direction)

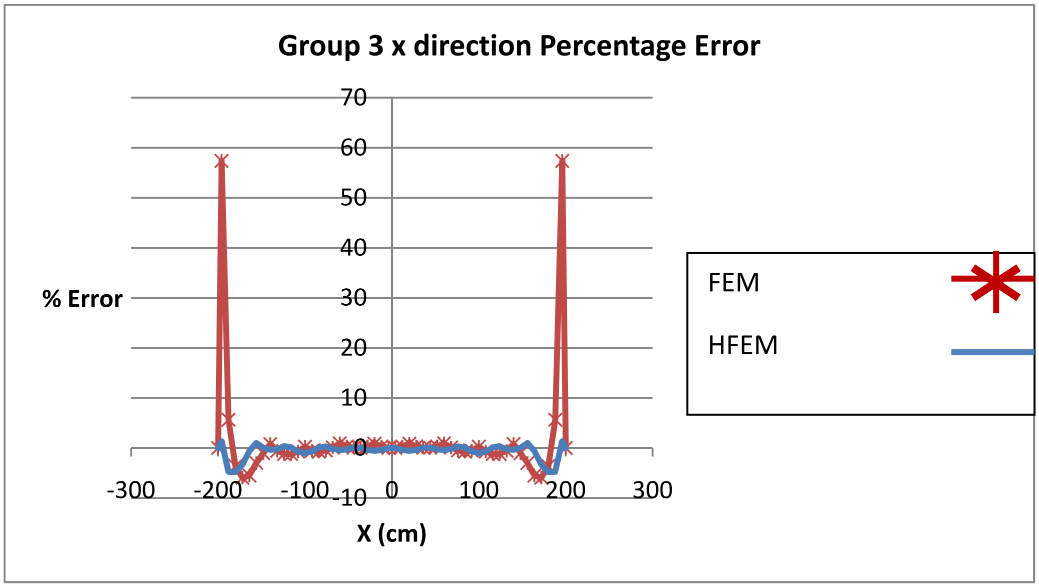


Figure 5.16: Group 3 Flux percent error axial profile (X direction)

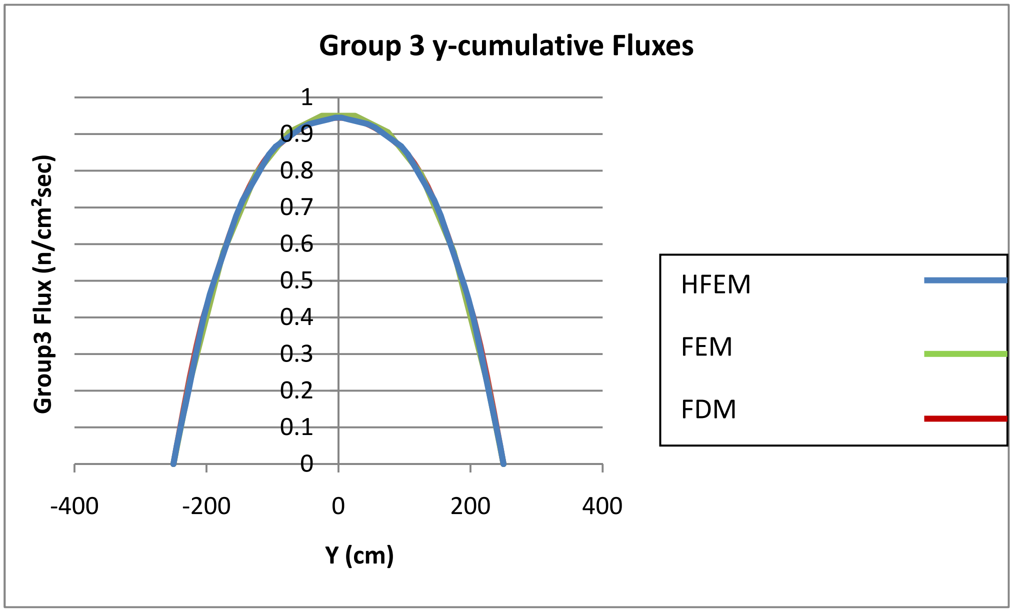


Figure 5.17: Group 3 Flux axial profile (Y direction)

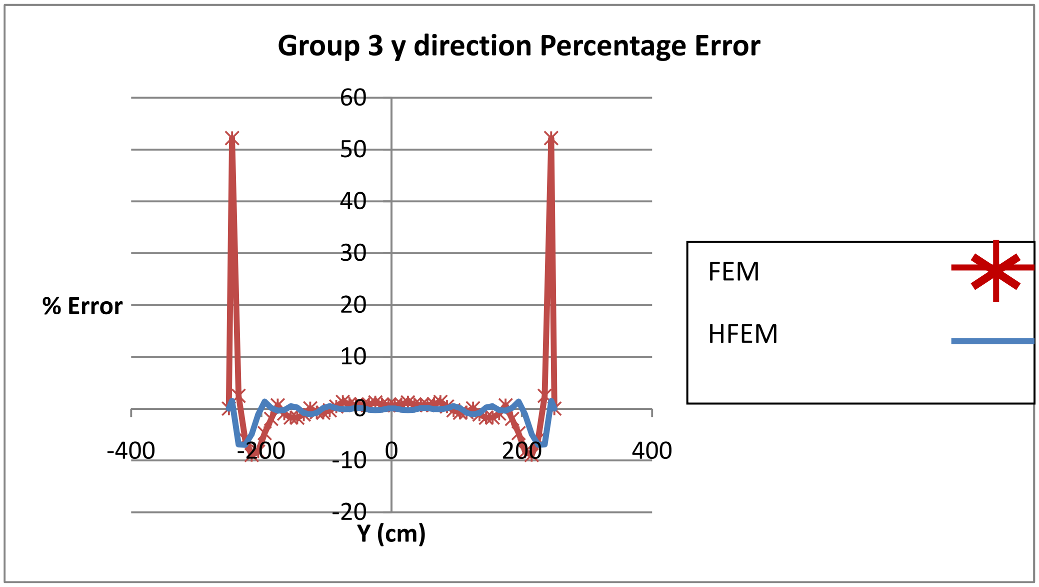


Figure 5.18: Group 3 Flux percent error axial profile (Y direction)

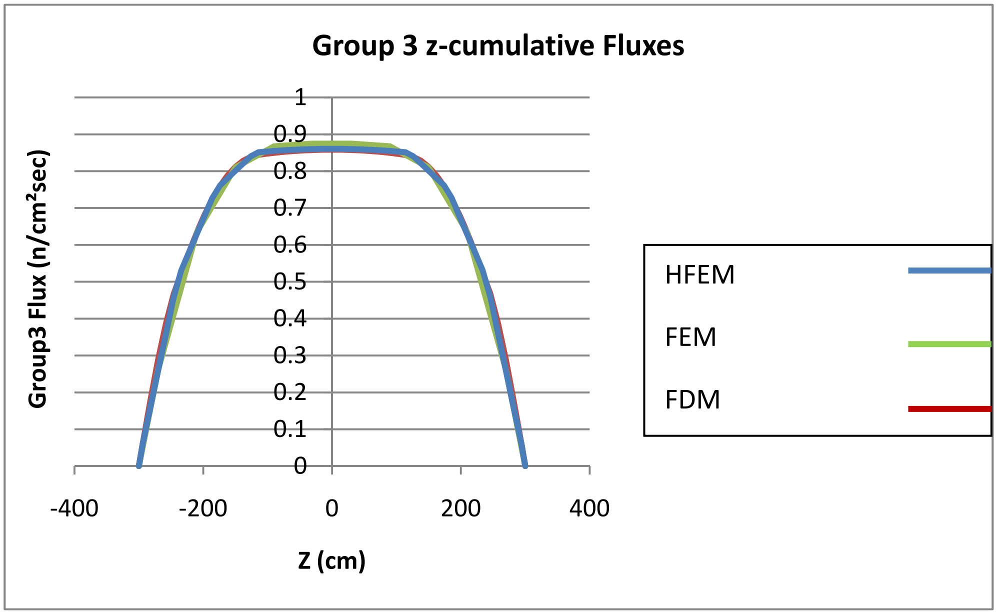


Figure 5.19: Group 3 Flux axial profile (Z direction)

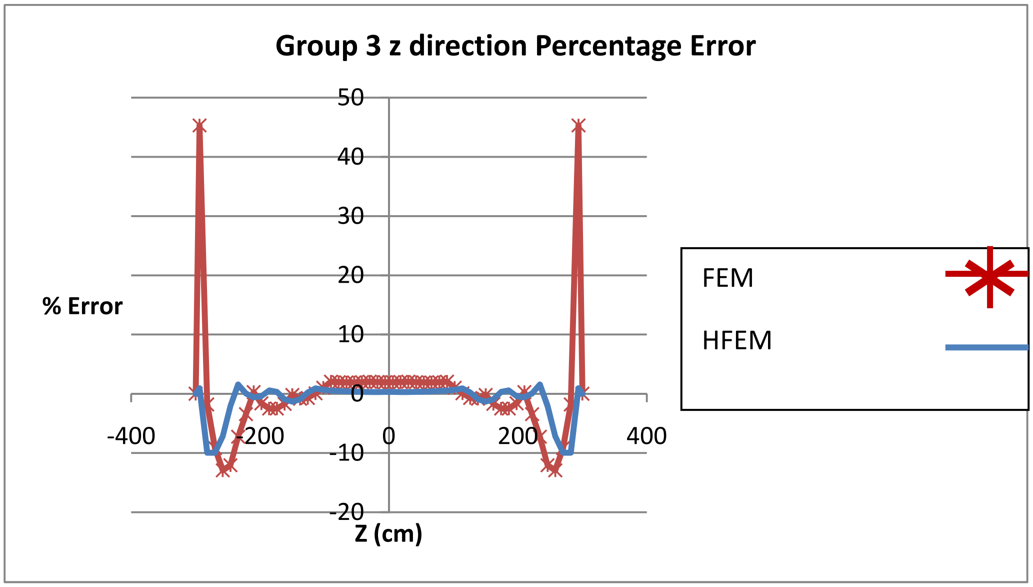


Figure 5.20: Group 3 Flux percent error axial profile (Z direction)

5.4 DISCUSSION

The purpose of the test calculations was to verify the functionality of the HFEM implementation in the FORTRAN code and to assess its potential for reducing computational time. To verify that a new code produces correct results, it is usually compared against a well-documented code that uses the exact same method. In this case, however, such a code was not available. To be able to still perform a good verification, a regular FEM code had to be used. In doing so, however, two requirements had to me met: The elements had to be homogeneous and the basis functions had to be close approximations of the ones used by the regular FEM code. The first requirement was met by using a model with homogeneous elements, while the second requirement was met by using discrete approximations of the trilinear functions used by the regular FEM. While such a choice of model and functions does not make full use of the FEM's flexibility, it is a necessary step in verifying the code's functionality and in creating a basis for future development of special basis functions suitable for specific applications. One generalization was still allowed in the model, by using only a two-group representation in the HFEM code, compared to a three-group representation in the FEM code. The comparison against fine-mesh finite-difference results was made in order to prove the code's ability to generate accurate fine-mesh results and to assess the relative reduction in computational time.

Table (5.2) shows that the value of the multiplication constant produced by HFEM is close to the value calculated using FDM, the error being less than 1mk, which is usually an accepted limit for k-eff errors. The error is also of approximately the same order of magnitude as that obtained using regular FEM, which is consistent with using similar-shape basis functions as the regular FEM.

The computation time is reduced by approximately 250 times compared to the FDM which is consistent with a 125x3/2 = 187 reduction in the number of unknowns used by the HFEM. The additional reduction in computational time is presumed to be caused by better convergence properties of the HFEM matrix compared to the FDM matrix. The computational time of the HFEM is approximately 2/3 of the regular FEM computational time which is consistent with using 2 groups instead of 3 groups. Since, for the current choice of basis functions, the regular FEM and HFEM matrices are quasi-identical, they have similar convergence properties and hence no additional difference in computational time is to be expected and none is, indeed, observed.

It is to be noted that, given the particular way in which the test was performed, absolute values of computational times for the three methods are not representative of production runs. What is important and relevant are the ratios of computational times for different methods since they all used the same iteration parameters.

The RMS percent errors shown in Table (5.3) show the HFEM to compare favourably with FDM, the errors being of the order of 1%. The fact that HFEM errors tend to be slightly smaller than the FEM errors is considered to be fortuitous and not significant as, overall, errors are very small for both methods.

The axial flux profiles and percent errors show good agreement between the HFEM and FDM method as well as between the HFEM and FEM. The comparatively large errors seen in the FEM results for points immediately near the boundary are interpreted as being due to the very small (near zero) value of the flux at that point which appears in Eq. (5.1).

Overall, the test has verified the implementation of the HFEM method to be correct and to yield relative reductions in computational time consistent with the expectations.

CHAPTER 6

CONCLUSIONS AND FUTURE WORK

6.1 CONCLUSION

A Heterogeneous Finite Element Method (HFEM) for three dimensional, multi-group diffusion theory has been developed and implemented in a FORTRAN code. The method allows for the use of general homogeneous boundary conditions in the form of a current-to-flux ratio, which, depending on the value of the ratio, can simulate reflective boundary conditions, vacuum boundary conditions and zero-flux boundary conditions. The HFEM FORTRAN program accepts an arbitrary number of nodes or and energy groups. The HFEM FORTRAN program has been tested for a 400 cm x 500 cm x 600 cm model. It was found that using two groups and 10 x 10 x 10 elements for the HFEM gives very similar results to using three groups and 50 x 50 x 50 meshes for FDM. The difference in k-eff was below 1 mk and the RMS percent difference was below 1%. The computation time for HFEM was found to be reduced by a factor of 250 compared to FD, which is in line with the reduction in the number of unknowns.

The HFEM offers the following advantages compared to traditional assembly-homogenization followed by coarse-mesh full-core calculation methods:

- Eliminates the need of assembly level homogenization

- Offers fuel-pin level detail, thus eliminating the need for “a posteriori” pin-power reconstruction.
- Allows the use of arbitrary basis functions thus allowing them to be tailored to specific applications.
- Offers accuracy comparable to fine-mesh calculations at a low computational cost, comparable to that of coarse-mesh methods.

If fine-mesh calculations are taken as reference, the HFEM offers comparable accuracy at a drastically reduced computational cost and memory requirements (usually a factor of 100 or better).

It can be concluded that the Heterogeneous Finite Element Method achieves high accuracy with very low computational cost which makes it suitable for fast analysis.

6.2 FUTURE WORK

More realistic geometries, including strongly-heterogeneous assemblies, will have to be studied to confirm HFEM’s ability to consistently provide accurate results with modest computational cost. Suitable basis functions will have to be developed for specific applications. For CANDU applications, a pin-by-pin homogenization method will have to be developed, which is not a trivial task, given the non-Cartesian fuel-pin distribution in CANDU bundles.

To improve performance, it is desirable to optimize the convergence criteria and number of iterations for the three nested iteration loops so as to achieve good accuracy without an excessive number of iterations. It is also desirable to investigate rebalancing and other convergence acceleration techniques, none of which were used in the current implementation.

REFERENCES

[1] K. ABE and S. LIANG ZHANG ,(2006), "A Variant of the Orthomin(m) Method for Solving Linear Systems", Nagoya Univ., Graduate School of Engineering, Japan

[2] R. T. ACKROYD, A.M. GASHUT, O. A. ABUZID (1996) "Discontinuous Variational Solutions for the Neutron Diffusion Equation", Ann. Nucl. Energy, 23, 1289-1300

[3] J. M. ARAGONES, C. AHNERT (1986) " A Linear Discontinuous Finite Difference Formulation for Synthetic Coarse-Mesh Few-Group Diffusion Calculations", Nucl. Sci. Eng. 94, 309

[4] R. BARRETT et. Al. (2008), "Templates for the Solution of Linear Systems: Building Blocks for Iterative Methods", SIAM.

[5] T. L. CHOW , (2000), "Mathematical Methods for Physicists", Cambridge University, United Kingdom

[6] I. CRAW, (2000), "Advanced Calculus and Analysis", Department of Mathematical Sciences University of Aberdeen, United Kingdom

[7] J. J. DUDERSTADT and L. J. HAMILTON, (1976), "Nuclear Reactor Analysis", John Wiley & Sons, New York.

[8] B. FORGET and F. RAHNEMA (2006) "Inexpensive Detailed Full-Core Neutron Flux Solution Method for CANDU Reactors", Trans. Am. Nucl. Soc. 96, 709-712

[9] ANNE GREENBAUM, (1997), "Iterative Methods for Solving Linear Systems", SIAM.

[10] K.F HANSEN and C.M, KANG (1975) "Finite Element Methods in Reactor Physics Analysis", Advances in Nuclear Science and Technology. Vol.8, Academic Press.

[11] A. HEBERT (1993) "A Consistent Technique for the Pin-by-Pin Homogenization of a Pressurized Water Reactor Assembly", Nucl. Sci. Eng. 113, 227-238

[12] A.F HENRY, (1975), “Nuclear Reactor Analysis" ,MIT.

[13] A. F. HENRY, C. L. HOXIE, (1981) ,“Reconstruction of Heterogeneous PWR Flux Shapes from Nodal Calculations”, Trans. Am. Nucl. Soc. 39, 905.

[14] T. J.R HUGHES, (1987), "The Finite Element Method", Prentice-Hall Englewood Cliffs NJ.

[15] T. ISE, T. YAMAZAKI and Y. NAUWRA , (1978), "Application of the Finite Element Method to the Three-Dimensional Neutron Diffusion Equation", NEACRP 21th Meeting Tokai.

[16] H. G. JOO, I. J. YOON and J. S. BAEK (2009) "Multigroup pin power reconstruction with two-dimensional source expansion and corner flux discontinuity", Ann. Nucl. Energy 36(1), 85-97

[17] R. J. LEVEQUE, (2006), "Finite Difference Methods for Differential Equations" University of Washington.

[18] R. J. LEVEQUE , (2007) , "Finite difference methods for ordinary and partial differential equations", Society for Industrial and Applied Mathematics, Philadelphia, USA

[19] E. E LEWIS and W.F. MILLER, Jr., (1984), "Computational Methods in Neutron Transport" John Wiley & Sons, New York.

[20] R. S. MODAK and A. GUPTA (2006) "New applications of Orthomin(1) algorithm for K-eigenvalue problem in Reactor Physics ", Ann. Nucl. Energy., 33, 538-543

[21] E. NICHITA (2009) "Evaluating Accuracy of Standard Homogenization and Need for Generalized Equivalence Theory for ACR-lattice Checkerboard Configurations", Ann. Nucl. Energy, 36, 760

[22] E. NICHITA (2009) "Evaluating Accuracy of Standard Homogenization and Need for Generalized Equivalence Theory for ACR-lattice Checkerboard Configurations", Ann. Nucl. Energy, 36, 760-772

[23] E. NICHITA (2009) "Inexpensive Detailed Full-Core Neutron Flux Solution Method for CANDU Reactors", Trans. Am. Nucl. Soc. 100, 631-633

[24] E. NICHITA (2010) "Generalized Equivalence Theory for Checkerboard Configurations in Natural-Uranium CANDU Lattices", Proc. PHYSOR 2010, International Conference on Reactor Physics, Pittsburgh, PA, May 9-14,

[25] E. NICHITA, and F. RAHNEMA (1998) "A Finite Element Method for Boiling Water Reactors", Proc. Int’l Conf. on Phys. of Nucl. Sci. and Tech., Long Island – NY, Oct. 5-8

[26] E. NICHITA, K. ZABIENSKI and M. GRAVEL, (2007) "Three-Dimensional Two-Energy Group Finite-Difference Neutron-Diffusion Code with Discontinuity Factors" , Proc. 28th Annual Conference of the Canadian Nuclear Society, Saint John NB June 3-6.

[27] K. O. OTT and ROBERT J NEUHOLD, (1983), "Introductory Nuclear Reactor Dynamics" American Nuclear Society, Illinois.

[28] K. O. OTT and WINFRED A. BEZELLA, (1983), "Introductory Nuclear Reactor Statics" American Nuclear Society, Illinois.

[29] F. RAHNEMA and E. NICHITA (1995) "Cross Section Homogenization Using Precomputed Surface Flux Response Matrices", Trans. Am. Nucl. Soc. 72, 358-360

[30] F. RAHNEMA and E. M. NICHITA (1997) "Leakage Corrected Spatial (Assembly) Homogenization Technique", Ann. Nucl. Energy., 24 477-488

[31] O. RÜBENKÖNIG, (2006), ‘"The Finite Difference Method (FDM) - An introduction’", Albert Ludwigs University of Freiburg, Germany

[32] Y. SAAD, (2000), "Iterative Methods for Sparse Linear Systems"

[33] W. SHEN (2006) "Development of a Multicell Methodology to Account for Heterogeneous Core Effects in the Core-Analysis Diffusion Code", PHYSOR 2006, ANS Topical Meeting on Reactor Physics, Vancouver, BC, Canada

[34] K. SINGH, M. WATANABE and V. KUMAR (1995) "Reconstruction of pin power in fuel assemblies from nodal calculations in hexagonal geometry", Ann. Nucl. Energy,22, 629-647

[35] K. S. SMITH (1994) "A Practical and Efficient Iterative Method for LWR Fuel Assembly Homogenization", Trans. Am. Nucl. Soc. 71, 238-239

[36] SMITH, K.S. (1980), "Spatial Homogenization Methods for Light Water Reactor Analysis," Ph.D. Thesis, Massachusetts Institute of Technology.

[37] SMITH, K.S. (1986), "Assembly Homogenization Techniques for Light Water Reactor Analysis, " Progress in Nuclear Energy, 17, 303-335.

[38] W. M. STACEY, (2002), "Nuclear Reactor Physics", John Wiley & Sons, New York.

[39] G. STRANG and G. J. FIX (1973) "An Analysis of The Finite Element Method", Prentice-Hall Englewood Cliffs NJ.

[40] M. TOHJOH, M. WATANABE and A. YAMAMOTO (2006) "Three-dimensional pin power reconstruction for the axially heterogeneous region in BWR", Ann. Nucl. Energy, 33(3), 242-251

[41] E. Varin, R. Roy, G. Hotte and R. Baril (2004). “ CANDU-6 operation post-simulations using the reactor physics codes DRAGON/DONJON”, Ann. Nucl. Energy, 31/18, pp 2139-2155

[42] E. Varin, A. Hebert, R. Roy and J. Koclas (2005). “ A user Guide for DONJON”, IGN, Ecole Polytechnique de Montr´eal

[43] H. ZHANG, RIZWAN-UDDIN and J.J. DORNING (1995) "Systematic Homogenization and Self-Consistent Flux and Pin Power Reconstruction for Nodal Diffusion Methods in Diffusion Equation-Based Theory", Nucl. Sci. Eng. 121, 226-244

APPENDIX A

DETAILED DERIVATION OF FINITE DIFFERENCE EQUATIONS

The finite difference approximation of derivatives is based on the standard limit definition of the derivative which is given as (Craw, 2000):

$$\left.\frac{du(x)}{dx}\right|_{x=a} = \mathrm{Lim}_{h\to 0}\frac{u(a+h)-u(a)}{h} = \mathrm{Lim}_{h\to 0}\frac{u(a)-u(a-h)}{h} \tag{A.1}$$

where, a is the point where the derivative needs to be approximated and h is the infinitesimal increase in argument

According to Eq. (A.1) , if h is not small enough, the definition turns out to be an approximation (Randall J. LeVeque, 2007) such that;

$$\left.\frac{du(x)}{dx}\right|_{x=a} \cong \frac{u(a)-u(a-h)}{h} \qquad \longrightarrow \qquad \text{backward approximation} \tag{A.2a}$$

$$\left.\frac{du(x)}{dx}\right|_{x=a} \cong \frac{u(a+h)-u(a)}{h} \qquad \longrightarrow \qquad \text{forward approximation} \tag{A.2b}$$

$$\left.\frac{du(x)}{dx}\right|_{x=a} \cong \frac{u(a+h)-u(a-h)}{2h} \qquad \longrightarrow \qquad \text{central approximation} \tag{A.2c}$$

Equation (A.2c) is actually equal to the arithmetic mean of the forward and backward approximations.

The geometrical representation of Eqs. (A.1) ans (A.2) can be seen in Fig. (A.1) where finite differences are used to approximate the derivative of the function u(x) at x=a.

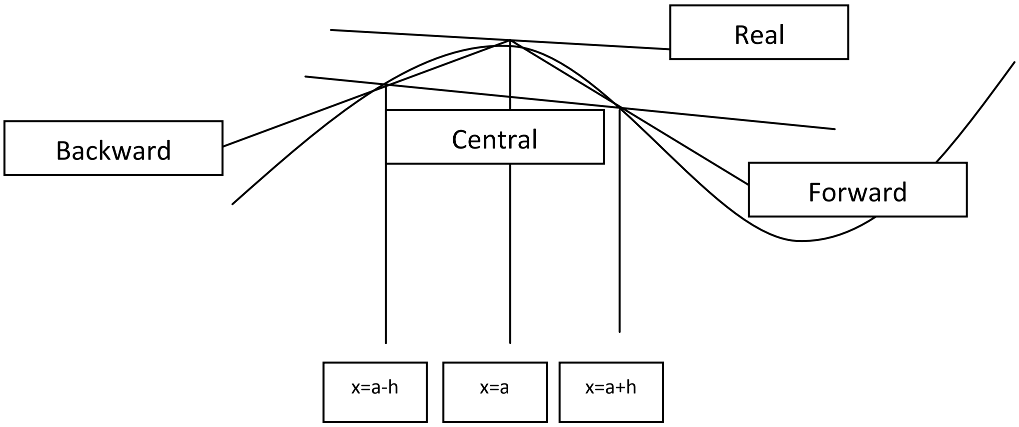


Figure A.1 Forward Backward and Central Finite Difference Approximations

If the geometry simple the mesh size can be chosen to be constant.

Finite difference equations can also be applied to partial differential equations

Finite Difference equations can be used for higher order partial differential equations and for higher order ordinary differential equations.

The second order derivative approximation in terms of first order derivatives are given in the set of Eqs. (A.3):

$$\left.\frac{d^2u(x)}{dx^2}\right|_{x=a(FDD)} \cong \frac{\left.\frac{du(x+h)}{dx}\right|_{x=a} - \left.\frac{du(x)}{dx}\right|_{x=a}}{h}$$

(A.3a)

$$\left.\frac{d^2u(x)}{dx^2}\right|_{x=a(BDD)} \cong \frac{\left.\frac{du(x)}{dx}\right|_{x=a} - \left.\frac{du(x+h)}{dx}\right|_{x=a}}{h}$$

(A.3b)

More generally, n_{th} order forward, backward and central finite differences are given below:

$$\left.\frac{d^{(n)}u(x)}{dx^{(n)}}\right|_{FDD} = \sum_{i=o}^{n}(-1)^i\binom{n}{i}u(x+(n-i)h)$$

(A.4a)

$$\left.\frac{d^{(n)}u(x)}{dx^{(n)}}\right|_{BDD} = \sum_{i=o}^{n}(-1)^i\binom{n}{i}u(x-ih)$$

(A.4b)

$$\left.\frac{d^{(n)}u(x)}{dx^{(n)}}\right|_{CDD} = \sum_{i=o}^{n}(-1)^i\binom{n}{i}u(x-\left(\frac{n}{2}-i\right)h)$$

(A.4c)

In what follows the application of finite differences to the diffusion equation is explained. In three dimensions, to define the location of the region used in FDM a three-index notation is used to define the x, y and z coordinates of the region. The flux inside each region is approximated by the flux at the center of the region (mesh box):

$$\varphi_{\mathrm{FDregion}} = \varphi_{\mathrm{i,j,k}} = \varphi\left(\frac{x_F + x_B}{2}, \frac{y_R + y_L}{2}, \frac{z_T + z_B}{2}\right)$$

(A.5)

where the space dependent flux is $\varphi(x,y,z)$ and x, y and z are the indexes that define the location in the geometry.

The indexing of the volumetric region is done according to the indexing of the flux at the center of the region as depicted in the Fig. (A.2).

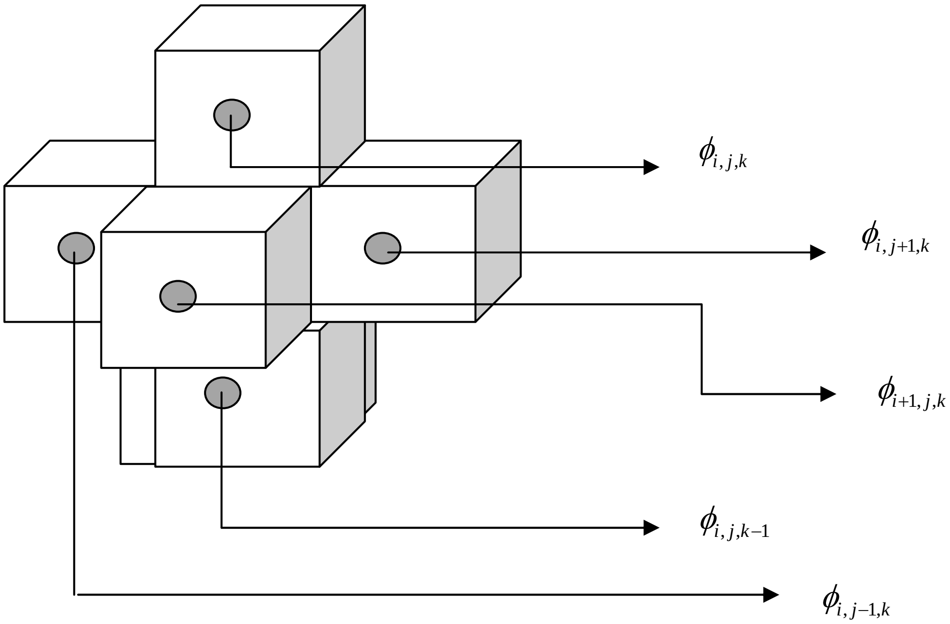
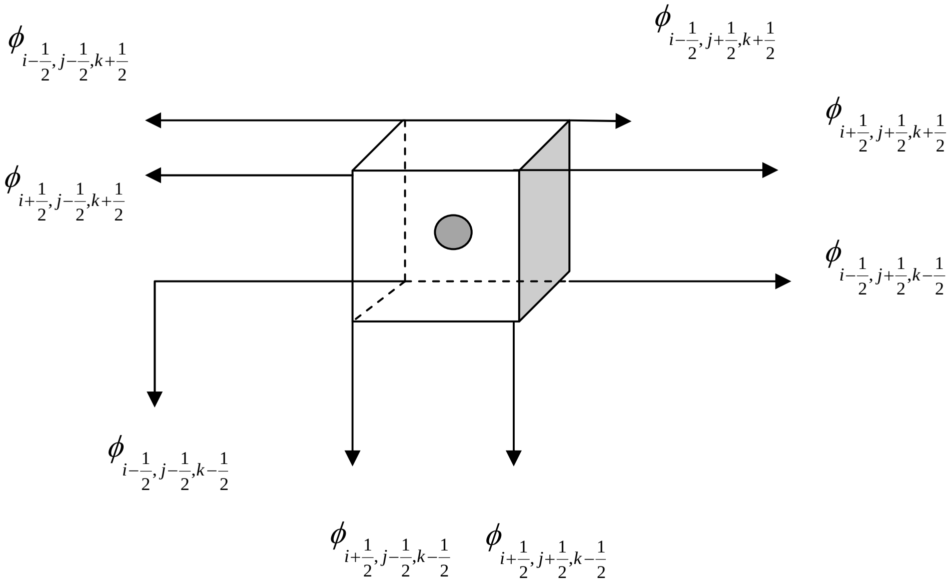


Figure A.2 Schematic representation of the indexing of central fluxes of volumetric regions

The flux indexing at the corners of the volumetric elements are depicted in the Fig. (A.3)



$\phi_{i-\frac{1}{2},j-\frac{1}{2},k+\frac{1}{2}}$
$\phi_{i-\frac{1}{2},j+\frac{1}{2},k+\frac{1}{2}}$
$\phi_{i+\frac{1}{2},j+\frac{1}{2},k+\frac{1}{2}}$
$\phi_{i+\frac{1}{2},j-\frac{1}{2},k+\frac{1}{2}}$
$\phi_{i-\frac{1}{2},j+\frac{1}{2},k-\frac{1}{2}}$
$\phi_{i-\frac{1}{2},j-\frac{1}{2},k-\frac{1}{2}}$
$\phi_{i+\frac{1}{2},j-\frac{1}{2},k-\frac{1}{2}}$
$\phi_{i+\frac{1}{2},j+\frac{1}{2},k-\frac{1}{2}}$

Figure A.3 Schematic representation of the indexing of corner fluxes of a volumetric region

Average neutron fluxes at mesh boundaries are given as;

$$\varphi_{i+\frac{1}{2}.j.k} = \varphi\left(x_F, \frac{y_R + y_L}{2}, \frac{z_T + z_B}{2} \right) \tag{A.6a}$$

$$\varphi_{i-\frac{1}{2}.j.k} = \varphi\left(x_B, \frac{y_R + y_L}{2}, \frac{z_T + z_B}{2} \right) \tag{A.6b}$$

$$\varphi_{i.j+\frac{1}{2}.k} = \varphi\left(\frac{x_F + x_B}{2}, y_R, \frac{z_T + z_B}{2} \right) \tag{A.6c}$$

$$\varphi_{i.j-\frac{1}{2}.k} = \varphi\left(\frac{x_F + x_B}{2}, y_L, \frac{z_T + z_B}{2} \right) \tag{A.6d}$$

$$\varphi_{\text{i.j.k}+\frac{1}{2}} = \varphi\left(\frac{x_F + x_B}{2}, \frac{y_R + y_L}{2}, z_T\right) \tag{A.6e}$$

$$\varphi_{\text{i.j.k}-\frac{1}{2}} = \varphi\left(\frac{x_F + x_B}{2}, \frac{y_R + y_L}{2}, z_B\right) \tag{A.6f}$$

The indexing of fluxes in boxes neighboring the current box (i,j,k) is shown in Fig. (A.4) Fig. A 5 shows the dimensions of box (i,j,k).

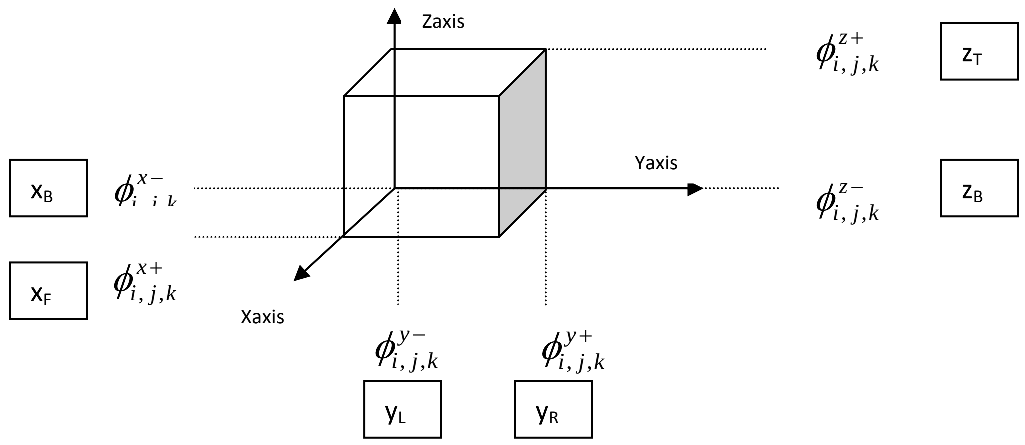


Figure A.4 Schematic representation of forward and backward fluxes for a volumetric region

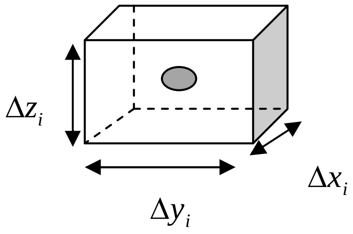


Figure A.5 Dimensions of a volumetric region

According to the finite difference approximation, the flux derivative is approximated with finite differences at mesh boundaries as depicted in Fig (A.6) for a one dimensional region:

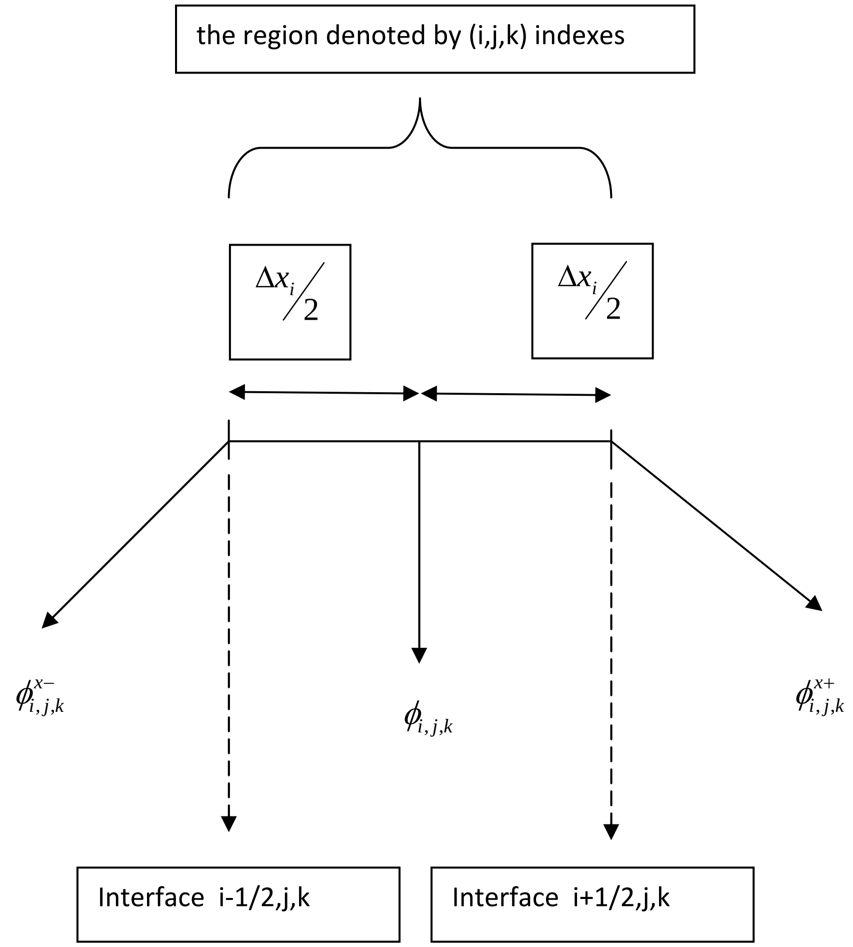


Figure A.6 Schematic representation of 1 dimensional region related fluxes

The flux derivatives at mesh boundaries can be written as:

$$\left(\frac{\partial\varphi(x,y,z)}{\partial x}\right)^{x-}_{i+\frac{1}{2},j,k} \approx \frac{\varphi^{x+}_{i,j,k}-\varphi_{i,j,k}}{\Delta x_i/2} = \frac{\varphi_{i+\frac{1}{2},j,k}-\varphi_{i,j,k}}{\Delta x_i/2} \quad \text{backward differences} \qquad \textbf{(A.8a)}$$

$$\left(\frac{\partial \varphi(x,y,z)}{\partial x}\right)^{x+}_{i+\frac{1}{2},j,k} \approx \frac{\varphi_{i+1,j,k}-\varphi^{x+}_{i,j,k}}{\Delta x_{i+1}/2} = \frac{\varphi_{i+1,j,k}-\varphi_{i+\frac{1}{2},j,k}}{\Delta x_{i+1}/2} \quad \text{forward differences} \qquad \textbf{(A.8b)}$$

$$\left(\frac{\partial \varphi(x,y,z)}{\partial x}\right)^{x+}_{i-\frac{1}{2},j,k} \approx \frac{\varphi_{i,j,k}-\varphi^{x-}_{i,j,k}}{\Delta x_{i}/2} = \frac{\varphi_{i,j,k}-\varphi_{i-\frac{1}{2},j,k}}{\Delta x_{i}/2} \quad \text{forward differences} \qquad \textbf{(A.8c)}$$

$$\left(\frac{\partial \varphi(x,y,z)}{\partial x}\right)^{x-}_{i-\frac{1}{2},j,k} \approx \frac{\varphi^{x-}_{i,j,k}-\varphi_{i-1,j,k}}{\Delta x_{i-1}/2} = \frac{\varphi_{i-\frac{1}{2},j,k}-\varphi_{i-1,j,k}}{\Delta x_{i-1}/2} \quad \text{backward differences} \qquad \textbf{(A.8d)}$$

$$\left(\frac{\partial \varphi(x,y,z)}{\partial y}\right)^{y-}_{i,j+\frac{1}{2},k} \approx \frac{\varphi^{y+}_{i,j,k}-\varphi_{i,j,k}}{\Delta y_{j}/2} = \frac{\varphi_{i,j+\frac{1}{2},k}-\varphi_{i,j,k}}{\Delta y_{j}/2} \quad \text{with backward differences} \qquad \textbf{(A.9a)}$$

$$\left(\frac{\partial \varphi(x,y,z)}{\partial y}\right)^{y+}_{i,j+\frac{1}{2},k} \approx \frac{\varphi_{i,j+1,k}-\varphi^{y+}_{i,j,k}}{\Delta y_{j+1}/2} = \frac{\varphi_{i,j+1,k}-\varphi_{i,j+\frac{1}{2},k}}{\Delta y_{j+1}/2} \quad \text{with forward differences} \qquad \textbf{(A.9b)}$$

$$\left(\frac{\partial \varphi(x,y,z)}{\partial y}\right)^{y+}_{i,j-\frac{1}{2},k} \approx \frac{\varphi_{i,j,k}-\varphi^{y-}_{i,j,k}}{\Delta y_{j}/2} = \frac{\varphi_{i,j,k}-\varphi_{i,j-\frac{1}{2},k}}{\Delta y_{j}/2} \quad \text{with forward differences}$$

(A.9c)

$$\left(\frac{\partial \varphi(x,y,z)}{\partial y}\right)^{y-}_{i,j-\frac{1}{2},k} \approx \frac{\varphi^{y-}_{i,j,k}-\varphi_{i,j-1,k}}{\Delta y_{j-1}/2} = \frac{\varphi_{i,j-\frac{1}{2},k}-\varphi_{i,j-1,k}}{\Delta y_{j-1}/2} \quad \text{backward differences} \qquad \textbf{(A.9d)}$$

$$\left(\frac{\partial \varphi(x,y,z)}{\partial z}\right)^{z-}_{i,j,k+\frac{1}{2}} \approx \frac{\varphi^{z+}_{i,j,k}-\varphi_{i,j,k}}{\Delta z_{k}/2} = \frac{\varphi_{i,j,k+\frac{1}{2}}-\varphi_{i,j,k}}{\Delta z_{k}/2} \quad \text{backward differences} \qquad \textbf{(A.10a)}$$

$$\left(\frac{\partial\varphi(x,y,z)}{\partial z}\right)^{z+}_{i,j,k+\frac{1}{2}} \approx \frac{\varphi_{i,j,k+1}-\varphi^{z+}_{i,j,k}}{\Delta z_{k+1}\big/2} = \frac{\varphi_{i,j,k+1}-\varphi_{i,j,k+\frac{1}{2}}}{\Delta z_{k+1}\big/2} \quad \text{forward differences} \qquad \textbf{(A.10b)}$$

$$\left(\frac{\partial\varphi(x,y,z)}{\partial z}\right)^{z+}_{i,j,k-\frac{1}{2}} \approx \frac{\varphi_{i,j,k}-\varphi^{z-}_{i,j,k}}{\Delta z_{k}\big/2} = \frac{\varphi_{i,j,k}-\varphi_{i,j,k-\frac{1}{2}}}{\Delta z_{k}\big/2} \quad \text{forward differences} \qquad \textbf{(A.10c)}$$

$$\left(\frac{\partial\varphi(x,y,z)}{\partial z}\right)^{z-}_{i,j,k-\frac{1}{2}} \approx \frac{\varphi^{z-}_{i,j,k}-\varphi_{i,j,k-1}}{\Delta z_{k-1}\big/2} = \frac{\varphi_{i,j,k-\frac{1}{2}}-\varphi_{i,j,k-1}}{\Delta z_{k-1}\big/2} \quad \text{backward differences} \qquad \textbf{(A.10d)}$$

where $(i\mp\frac{1}{2},j,k),(i,j\mp\frac{1}{2},k),(i,j,k\mp\frac{1}{2})$ represent interfaces at which the derivative is to be calculated and $x^{\mp},y^{\mp},z^{\mp}$ denotes which finite difference approximation is used with "+" designating forward and "-" designating backwards.

Currents are calculated as:

$$J^{x-}_{i+\frac{1}{2},j,k} = -D_{i,j,k}\left(\frac{\partial\varphi(x,y,z)}{\partial x}\right)^{x-}_{i+\frac{1}{2},j,k} \approx -D_{i,j,k}\,\frac{\varphi^{x+}_{i,j,k}-\varphi_{i,j,k}}{\Delta x_{i}\big/2} = -D_{i,j,k}\,\frac{\varphi_{i+\frac{1}{2},j,k}-\varphi_{i,j,k}}{\Delta x_{i}\big/2} \qquad \textbf{(A.11a)}$$

$$J^{x+}_{i+\frac{1}{2},j,k} = -D_{i+1,j,k}\left(\frac{\partial\varphi(x,y,z)}{\partial x}\right)^{x+}_{i+\frac{1}{2},j,k} \approx -D_{i+1,j,k}\,\frac{\varphi_{i+1,j,k}-\varphi^{x+}_{i,j,k}}{\Delta x_{i+1}\big/2} = -D_{i+1,j,k}\,\frac{\varphi_{i+1,j,k}-\varphi_{i+\frac{1}{2},j,k}}{\Delta x_{i+1}\big/2} \qquad \textbf{(A.11b)}$$

$$J^{x+}_{i-\frac{1}{2},j,k} = -D_{i,j,k}\left(\frac{\partial\varphi(x,y,z)}{\partial x}\right)^{x+}_{i-\frac{1}{2},j,k} \approx -D_{i,j,k}\,\frac{\varphi_{i,j,k}-\varphi^{x-}_{i,j,k}}{\Delta x_{i}\big/2} = -D_{i,j,k}\,\frac{\varphi_{i,j,k}-\varphi_{i-\frac{1}{2},j,k}}{\Delta x_{i}\big/2} \qquad \textbf{(A.11c)}$$

$$J^{x-}_{i-\frac{1}{2},j,k} = -D_{i-1,j,k}\left(\frac{\partial\varphi(x,y,z)}{\partial x}\right)^{x-}_{i-\frac{1}{2},j,k} \approx -D_{i-1,j,k}\,\frac{\varphi^{x-}_{i,j,k}-\varphi_{i-1,j,k}}{\Delta x_{i-1}\big/2} = -D_{i-1,j,k}\,\frac{\varphi_{i-\frac{1}{2},j,k}-\varphi_{i-1,j,k}}{\Delta x_{i-1}\big/2} \tag{A.11d}$$

$$J^{y-}_{i,j+\frac{1}{2},k} = -D_{i,j,k}\left(\frac{\partial\varphi(x,y,z)}{\partial y}\right)^{y-}_{i,j+\frac{1}{2},k} \approx -D_{i,j,k}\,\frac{\varphi^{y+}_{i,j,k}-\varphi_{i,j,k}}{\Delta y_{j}\big/2} = -D_{i,j,k}\,\frac{\varphi_{i,j+\frac{1}{2},k}-\varphi_{i,j,k}}{\Delta y_{j}\big/2} \tag{A.12a}$$

$$J^{y+}_{i,j+\frac{1}{2},k} = -D_{i,j+1,k}\left(\frac{\partial\varphi(x,y,z)}{\partial y}\right)^{y+}_{i,j+\frac{1}{2},k} \approx -D_{i,j+1,k}\,\frac{\varphi_{i,j+1,k}-\varphi^{y+}_{i,j,k}}{\Delta y_{j+1}\big/2} = -D_{i,j+1,k}\,\frac{\varphi_{i,j+1,k}-\varphi_{i,j+\frac{1}{2},k}}{\Delta y_{j+1}\big/2} \tag{A.12b}$$

$$J^{y+}_{i,j-\frac{1}{2},k} = -D_{i,j,k}\left(\frac{\partial\varphi(x,y,z)}{\partial y}\right)^{y+}_{i,j-\frac{1}{2},k} \approx -D_{i,j,k}\,\frac{\varphi_{i,j,k}-\varphi^{y-}_{i,j,k}}{\Delta y_{j}\big/2} = -D_{i,j,k}\,\frac{\varphi_{i,j,k}-\varphi_{i,j-\frac{1}{2},k}}{\Delta y_{j}\big/2} \tag{A.12c}$$

$$J^{y-}_{i,j-\frac{1}{2},k} = -D_{i,j-1,k}\left(\frac{\partial\varphi(x,y,z)}{\partial y}\right)^{y-}_{i,j-\frac{1}{2},k} \approx -D_{i,j-1,k}\,\frac{\varphi^{y-}_{i,j,k}-\varphi_{i,j-1,k}}{\Delta y_{j-1}\big/2} = -D_{i,j-1,k}\,\frac{\varphi_{i,j-\frac{1}{2},k}-\varphi_{i,j-1,k}}{\Delta y_{j-1}\big/2} \tag{A.12d}$$

$$J^{z-}_{i,j,k+\frac{1}{2}} = -D_{i,j,k}\left(\frac{\partial\varphi(x,y,z)}{\partial z}\right)^{z-}_{i,j,k+\frac{1}{2}} \approx -D_{i,j,k}\,\frac{\varphi^{z+}_{i,j,k}-\varphi_{i,j,k}}{\Delta z_{k}\big/2} = -D_{i,j,k}\,\frac{\varphi_{i,j,k+\frac{1}{2}}-\varphi_{i,j,k}}{\Delta z_{k}\big/2} \tag{A.13a}$$

$$J^{z+}_{i,j,k+\frac{1}{2}} = -D_{i,j,k+1}\left(\frac{\partial\varphi(x,y,z)}{\partial z}\right)^{z+}_{i,j,k+\frac{1}{2}} \approx -D_{i,j,k+1}\,\frac{\varphi_{i,j,k+1}-\varphi^{z+}_{i,j,k}}{\Delta z_{k+1}\big/2} = -D_{i,j,k+1}\,\frac{\varphi_{i,j,k+1}-\varphi_{i,j,k+\frac{1}{2}}}{\Delta z_{k+1}\big/2} \tag{A.13b}$$

$$J^{z+}_{i,j,k-\frac{1}{2}} = -D_{i,j,k}\left(\frac{\partial\varphi(x,y,z)}{\partial z}\right)^{z+}_{i,j,k-\frac{1}{2}} \approx -D_{i,j,k}\,\frac{\varphi_{i,j,k}-\varphi^{z-}_{i,j,k}}{\Delta z_{k}\big/2} = -D_{i,j,k}\,\frac{\varphi_{i,j,k}-\varphi_{i,j,k-\frac{1}{2}}}{\Delta z_{k}\big/2} \tag{A.13c}$$

$$J^{z-}_{i,j,k-\frac{1}{2}} = -D_{i,j,k-1}\left(\frac{\partial\varphi(x,y,z)}{\partial z}\right)^{z-}_{i,j,k-\frac{1}{2}} \approx -D_{i,j,k-1}\frac{\varphi^{z-}_{i,j,k} - \varphi_{i,j,k-1}}{\Delta z_{k-1}/2} = -D_{i,j,k-1}\frac{\varphi_{i,j,k-\frac{1}{2}} - \varphi_{i,j,k-1}}{\Delta z_{k-1}/2} \qquad \textbf{(A.13d)}$$

The continuity of currents across interfaces results in:

$$J^{x-}_{i+\frac{1}{2},j,k} = J^{x+}_{i+\frac{1}{2},j,k} \Rightarrow -D_{i,j,k}\frac{\varphi_{i+\frac{1}{2},j,k} - \varphi_{i,j,k}}{\Delta x_i/2} = -D_{i+1,j,k}\frac{\varphi_{i+1,j,k} - \varphi_{i+\frac{1}{2},j,k}}{\Delta x_{i+1}/2} \quad \text{then,}$$

$$\varphi_{i+\frac{1}{2},j,k} = \frac{\Delta x_{i+1}D_{i,j,k}\varphi_{i,j,k} + \Delta x_i D_{i+1,j,k}\varphi_{i+1,j,k}}{\Delta x_{i+1}D_{i,j,k} + \Delta x\ D_{i+1,jk}}$$

$$J^{x-}_{i-\frac{1}{2},j,k} = J^{x+}_{i-\frac{1}{2},j,k} \Rightarrow -D_{i\text{-}1,j,k}\frac{\varphi_{i-\frac{1}{2},j,k} - \varphi_{i\text{-}1,j,k}}{\Delta x_{i-1}/2} = -D_{i,j,k}\frac{\varphi_{i,j,k} - \varphi_{i-\frac{1}{2},j,k}}{\Delta x_i/2} \quad \text{then,}$$

$$\varphi_{i-\frac{1}{2},j,k} = \frac{\Delta x_{i-1}D_{i,j,k}\varphi_{i,j,k} + \Delta x_i D_{i\text{-}1,j,k}\varphi_{i\text{-}1,j,k}}{\Delta x_{i-1}D_{i,j,k} + \Delta x_i D_{i\text{-}1,j,k}}$$

(A.14a)

$$J^{y-}_{i,j+\frac{1}{2},k} = J^{y+}_{i,j+\frac{1}{2},k} \Rightarrow -D_{i,j,k}\frac{\varphi_{i,j+\frac{1}{2},k} - \varphi_{i,j,k}}{\Delta y_j/2} = -D_{i,j+1,k}\frac{\varphi_{i,j+1,k} - \varphi_{i,j+\frac{1}{2},k}}{\Delta y_{j+1}/2} \quad \text{then,}$$

$$\varphi_{i,j+\frac{1}{2},k} = \frac{\Delta y_{j+1}D_{i,j,k}\varphi_{i,j,k} + \Delta y_i D_{i,j+1,k}\varphi_{i,j+1,k}}{\Delta y_{j+1}D_{i,j,k} + \Delta y_j D_{i,j+1,k}}$$

$$J^{y-}_{i,j-\frac{1}{2},k} = J^{y+}_{i,j-\frac{1}{2},k} \Rightarrow -D_{i,j-1,k}\frac{\varphi_{i,j-\frac{1}{2},k} - \varphi_{i,j-1,k}}{\Delta y_{j-1}/2} = -D_{i,j,k}\frac{\varphi_{i,j,k} - \varphi_{i,j-\frac{1}{2},k}}{\Delta y_j/2} \quad \text{then,}$$

$$\varphi_{i,j-\frac{1}{2},k} = \frac{\Delta y_{j-1}D_{i,j,k}\varphi_{i,j,k} + \Delta y_i D_{i,j-1,k}\varphi_{i,j-1,k}}{\Delta y_{j-1}D_{i,j,k} + \Delta y_j D_{i,j-1,k}}$$

(A.14b)

$$J^{z-}_{i,j,k+\frac{1}{2}} = J^{z+}_{i,j,k+\frac{1}{2}} \Rightarrow -D_{i,j,k}\frac{\varphi_{i,j,k+\frac{1}{2}} - \varphi_{i,j,k}}{\Delta z_k/2} = -D_{i,j,k+1}\frac{\varphi_{i,j,k+1} - \varphi_{i,j,k+\frac{1}{2}}}{\Delta z_{k+1}/2} \quad \text{then,}$$

$$\varphi_{i,j,k+\frac{1}{2}} = \frac{\Delta z_{k+1} D_{i,j,k}\varphi_{i,j,k} + \Delta z_k D_{i,j,k+1}\varphi_{i,j,k+1}}{\Delta x_{k+1} D_{i,j,k} + \Delta x_k D_{i,j,k+1}}$$

$$J^{z-}_{i,j,k-\frac{1}{2}} = J^{z+}_{i,j,k-\frac{1}{2}} \Rightarrow -D_{i,j,k-1}\frac{\varphi_{i,j,k-\frac{1}{2}} - \varphi_{i,j,k-1}}{\Delta z_{k-1}/2} = -D_{i,j,k}\frac{\varphi_{i,j,k} - \varphi_{i,j,k-\frac{1}{2}}}{\Delta z_k/2} \quad \text{then,}$$

$$\varphi_{i,j,k-\frac{1}{2}} = \frac{\Delta z_{k-1} D_{i,j,k}\varphi_{i,j,k} + \Delta z_k D_{i,j,k-1}\varphi_{i,j,k-1}}{\Delta x_{k-1} D_{i,j,k} + \Delta x_k D_{i,j,k-1}} \qquad \textbf{(A.14c)}$$

Inserting the expressions for $\varphi_{i,j,k-\frac{1}{2}}, \varphi_{i,j,k+\frac{1}{2}}, \varphi_{i,j-\frac{1}{2},k}, \varphi_{i,j+\frac{1}{2},k}, \varphi_{i-\frac{1}{2},j,k}, \varphi_{i+\frac{1}{2},j,k}$ into Eqns. (A.8,9,10)) we obtain:

$$\left(\frac{\partial\varphi(x,y,z)}{\partial x}\right)^{x-}_{i+\frac{1}{2},j,k} \approx \frac{2D_{i+1,j,k}(\varphi_{i+1,j,k} - \varphi_{i,j,k})}{\Delta x_{i+1} D_{i,j,k} + \Delta x_i D_{i+1,j,k}} \qquad \textbf{(A.15a)}$$

$$\left(\frac{\partial\varphi(x,y,z)}{\partial x}\right)^{x+}_{i-\frac{1}{2},j,k} \approx \frac{-2D_{i-1,j,k}(\varphi_{i-1,j,k} - \varphi_{i,j,k})}{\Delta x_{i-1} D_{i,j,k} + \Delta x_i D_{i-1,j,k}} \qquad \textbf{(A.15b)}$$

$$\left(\frac{\partial\varphi(x,y,z)}{\partial y}\right)^{y-}_{i,j+\frac{1}{2},k} \approx \frac{2D_{i,j+1,k}(\varphi_{i,j+1,k} - \varphi_{i,j,k})}{\Delta y_{j+1} D_{i,j,k} + \Delta y_j D_{i,j+1,k}} \qquad \textbf{(A.15c)}$$

$$\left(\frac{\partial\varphi(x,y,z)}{\partial y}\right)^{y+}_{i,j-\frac{1}{2},k} \approx \frac{-2D_{i,j-1,k}(\varphi_{i,j-1,k} - \varphi_{i,j,k})}{\Delta y_{j-1} D_{i,j,k} + \Delta y_j D_{i,j-1,k}} \qquad \textbf{(A.15d)}$$

$$\left(\frac{\partial\varphi(x,y,z)}{\partial z}\right)^{z-}_{i,j,k+\frac{1}{2}} \approx \frac{2D_{i,j,k+1}(\varphi_{i,j,k+1}-\varphi_{i,j,k})}{\Delta z_{k+1}D_{i,j,k}+\Delta z_{k}D_{i,j,k+1}} \tag{A.15e}$$

$$\left(\frac{\partial\varphi(x,y,z)}{\partial z}\right)^{z+}_{i,j,k-\frac{1}{2}} \approx \frac{-2D_{i,j,k-1}(\varphi_{i,j,k-1}-\varphi_{i,j,k})}{\Delta z_{k-1}D_{i,j,k}+\Delta z_{k}D_{i,j,k-1}} \tag{A.15f}$$

First order derivatives in Eqs. (A.15) are used to derive second order derivatives. For inner regions where all neighbours exist, the central finite difference approximation for the second derivative is given by:

$$\left(\frac{\partial^2\varphi(x,y,z)}{\partial x^2}\right)_{i,j,k} \approx \frac{1}{\Delta x_i}\left(\left(\frac{\partial\varphi(x,y,z)}{\partial x}\right)^{x-}_{i+\frac{1}{2},j,k} - \left(\frac{\partial\varphi(x,y,z)}{\partial x}\right)^{x+}_{i-\frac{1}{2},j,k}\right) \tag{A.16a}$$

$$\left(\frac{\partial^2\varphi(x,y,z)}{\partial y^2}\right)_{i,j,k} \approx \frac{1}{\Delta y_j}\left(\left(\frac{\partial\varphi(x,y,z)}{\partial y}\right)^{y-}_{i,j+\frac{1}{2},k} - \left(\frac{\partial\varphi(x,y,z)}{\partial y}\right)^{y+}_{i,j-\frac{1}{2},k}\right) \tag{A.16b}$$

$$\left(\frac{\partial^2\varphi(x,y,z)}{\partial z^2}\right)_{i,j,k} \approx \frac{1}{\Delta z_k}\left(\left(\frac{\partial\varphi(x,y,z)}{\partial z}\right)^{z-}_{i,j,k+\frac{1}{2}} - \left(\frac{\partial\varphi(x,y,z)}{\partial z}\right)^{z+}_{i,j,k-\frac{1}{2}}\right) \tag{A.16c}$$

Inserting the first order partial derivative approximation (set of Eqns. (A.15)) into the second order partial derivative definition (set of Eqns. (A.16)) yields;

$$\left(\frac{\partial^2\varphi(x,y,z)}{\partial x^2}\right)_{i,j,k} \approx -\frac{1}{D_{ijk}}\left(a^{x+}_{i,j,k}\varphi_{i+1,j,k} - (a^{cx+}_{i,j,k}+a^{cx-}_{i,j,k})\varphi_{i,j,k} + a^{x-}_{i,j,k}\varphi_{i-1,j,k}\right) \tag{A.17a}$$

$$\left(\frac{\partial^2\varphi(x,y,z)}{\partial y^2}\right)_{i,j,k} \approx -\frac{1}{D_{ijk}}\left(a^{y+}_{i,j,k}\varphi_{i,j+1,k} - (a^{cy+}_{i,j,k}+a^{cy-}_{i,j,k})\varphi_{i,j,k} + a^{y-}_{i,j,k}\varphi_{i,j-1,k}\right) \tag{A.17b}$$

$$\left(\frac{\partial^2 \varphi(x,y,z)}{\partial z^2}\right)_{i,j,k} \approx -\frac{1}{D_{ijk}}\left(a^{z+}_{i,j,k}\varphi_{i,j,k+1} - (a^{cx+}_{i,j,k} + a^{cy-}_{i,j,k})\varphi_{i,j,k} + a^{z-}_{i,j,k}\varphi_{i,j,k+1}\right) \tag{A.17c}$$

where;

$$\begin{aligned}
a^{x+}_{i,j,k} &= a^{cx+}_{i,j,k} = -\frac{D_{i,j,k}}{\Delta x_i}\frac{2D_{i+1,j,k}}{\Delta x_{i+1}D_{i,j,k} + \Delta x_i D_{i+1,j,k}} \\
a^{x-}_{i,j,k} &= a^{cx-}_{i,j,k} = -\frac{D_{i,j,k}}{\Delta x_i}\frac{2D_{i-1,j,k}}{\Delta x_{i-1}D_{i,j,k} + \Delta x_i D_{i-1,j,k}} \\
a^{y+}_{i,j,k} &= a^{cy+}_{i,j,k} = -\frac{D_{i,j,k}}{\Delta y_j}\frac{2D_{i,j+1,k}}{\Delta y_{j+1}D_{i,j,k} + \Delta y_j D_{i,j+1,k}} \\
a^{y-}_{i,j,k} &= a^{cy-}_{i,j,k} = -\frac{D_{i,j,k}}{\Delta y_j}\frac{2D_{i,j-1,k}}{\Delta y_{j-1}D_{i,j,k} + \Delta y_j D_{i,j-1,k}} \\
a^{z+}_{i,j,k} &= a^{cz+}_{i,j,k} = -\frac{D_{i,j,k}}{\Delta z_k}\frac{2D_{i,j,k+1}}{\Delta z_{k+1}D_{i,j,k} + \Delta z_k D_{i,j,k+1}} \\
a^{z-}_{i,j,k} &= a^{cz-}_{i,j,k} = -\frac{D_{i,j,k}}{\Delta z_k}\frac{2D_{i,j,k-1}}{\Delta z_{k-1}D_{i,j,k} + \Delta z_k D_{i,j,k-1}}
\end{aligned} \tag{A.18}$$

The “c” superscripts designate the central region coefficient.

For external boundary meshes it is not possible to apply the central differences and thus one sided backward or forward difference are applied. If the boundary subregion is a boundary on the positive side then backward differences are applied, otherwise forward differences are applied (Oliver Rübenkönig, 2006).

To allow the boundary conditions to be as general as possible during the simulation, mixed heterogeneous (boundary conditions are implemented by imposing an outward current-to-flux ratio on each boundary

$$\gamma_{\text{boundary}} = \left(\frac{\vec{J} \cdot \hat{n}}{\varphi} \right)_{\text{boundary}} \tag{A.19}$$

where $\hat{n}$ is the outward-pointing unit vector normal to the external boundary.

According to the choice of γ_{boundary} reflective, zero flux and vacuum boundary conditions can be simulated. Considering the x direction, u is defined such u = 1 for the positive

$$\gamma^{xu} = \frac{uJ^{x}_{(b+\frac{u}{2},j,k)}}{\varphi_{(b+\frac{u}{2},j,k)}} = \frac{uJ^{xu}_{(b,j,k)}}{\varphi_{(b+\frac{u}{2},j,k)}} = \frac{uJ^{x(-u)}_{(b+\frac{u}{2},j,k)}}{\varphi_{(b+\frac{u}{2},j,k)}} \tag{A.20a}$$

$$\gamma^{xu} = \left(\frac{-D_{b,j,k}(\varphi^{xu}_{(b,j,k)} - \varphi_{(b,j,k)})}{(\Delta x_b/2)\varphi^{xu}_{(b,j,k)}} \right) \tag{A.20b}$$

Solving the Eq. (A.20b) for $\varphi_{(b,j,k)}$ we obtain:

$$\varphi^{xu}_{(b,j,k)} = \frac{-D_{b,j,k}\varphi_{(b,j,k)}}{\Delta x_b \gamma^{xu} + 2D_{b,j,k}} \tag{A.20c}$$

Similar equations are written for the y direction:

$$\gamma^{yu} = \frac{uJ^{y}_{(i,b+\frac{u}{2},k)}}{\varphi_{(i,b+\frac{u}{2},k)}} = \frac{uJ^{xu}_{(i,b,k)}}{\varphi_{(i,b+\frac{u}{2},k)}} = \frac{uJ^{y(-u)}_{(i,b+\frac{u}{2},k)}}{\varphi_{(i,b+\frac{u}{2},k)}} \tag{A.21a}$$

$$\gamma^{yu} = \left(\frac{-D_{i,b,k}(\varphi^{yu}_{(i,b,k)} - \varphi_{(i,b,k)})}{(\Delta y_b/2)\varphi^{yu}_{(i,b,k)}} \right) \tag{A.21b}$$

Solving Eq. (A.21b) for $\varphi_{(i,b,k)}$ gives:

$$\varphi^{yu}_{(i,b,k)} = \frac{-D_{i,b,k}\varphi_{(i,b,k)}}{\Delta y_b \gamma^{yu} + 2D_{i,b,k}} \tag{A.21c}$$

For the z direction, we have:

$$\gamma^{zu} = \frac{uJ^{z}_{(i,j,b+\frac{u}{2})}}{\varphi_{(i,j,b+\frac{u}{2})}} = \frac{uJ^{zu}_{(i,j,b)}}{\varphi_{(i,j,b+\frac{u}{2})}} = \frac{uJ^{z(-u)}_{(i,j,b+\frac{u}{2})}}{\varphi_{(i,j,b+\frac{u}{2})}} \tag{A.22a}$$

$$\gamma^{zu} = \left(\frac{-D_{(i,j,b)}(\varphi^{zu}_{(i,j,b)} - \varphi_{(i,j,b)})}{(\Delta z_b/2)\varphi^{zu}_{(i,j,b)}} \right) \tag{A.22b}$$

Solving Eq. (A.22b) for $\varphi_{(b,j,k)}$ gives:

$$\varphi^{zu}_{(i,j,b)} = \frac{-D_{(i,j,b)}\varphi_{(i,j,b)}}{\Delta z_b \gamma^{yu} + 2D_{(i,j,b)}} \tag{A.22c}$$

Inserting the flux expressions A.20.c, A.21.c and A.22.c into the expressions of the derivatives, we obtain:

$$\gamma : \left(\frac{\partial\varphi(x,y,z)}{\partial x}\right)^{x(-u)}_{b+\frac{u}{2},j,k} = \frac{-\Delta x_b}{D_{b,j,k}} a^{cxu}_{b,j,k}\varphi_{b,j,k}$$

where; $a^{cxu}_{b,j,k} = \dfrac{1}{\Delta x_b\left(\dfrac{1}{\gamma^{xu}} + \dfrac{\Delta x_b}{2D_{b,j,k}}\right)}$

itfollows that

$$\left(\frac{\partial^2\varphi(x,y,z)}{\partial x^2}\right)_{b,j,k} = \frac{1}{D_{b,j,k}}\left(a^{x(-u)}_{b-u,j,k}\varphi_{b-u,j,k} - (a^{cx(-u)}_{b,j,k} + a^{cxu}_{b,j,k})\varphi_{b,j,k}\right)$$

(A.23)

$$\left(\frac{\partial\varphi(x,y,z)}{\partial y}\right)^{y(-u)}_{b,j+\frac{u}{2},k} = \frac{-\Delta y_b}{D_{i,b,k}} a^{cyu}_{i,b,k}\varphi_{i,b,k}$$

where; $a^{cyu}_{i,b,k} = \dfrac{1}{\Delta y_b\left(\dfrac{1}{\gamma^{yu}} + \dfrac{\Delta y_b}{2D_{i,b,k}}\right)}$

itfollowst hat

$$\left(\frac{\partial^2\varphi(x,y,z)}{\partial y^2}\right)_{i,b,k} = \frac{1}{D_{i,b,k}}\left(a^{y(-u)}_{i,b-u,k}\varphi_{i,b-u,k} - (a^{cy(-u)}_{i,b,k} + a^{cyu}_{i,b,k})\varphi_{i,b,k}\right).$$

(A.24)

$$\left(\frac{\partial\varphi(x,y,z)}{\partial y}\right)_{i,j,b+\frac{u}{2}}^{z(-u)} = \frac{-\Delta z_b}{D_{i,j,b}} a_{i,j,b}^{czu}\varphi_{i,j,b}$$

where; $a_{i,j,b}^{czu} = \dfrac{1}{\Delta z_b\left(\dfrac{1}{\gamma^{zu}} + \dfrac{\Delta z_b}{2D_{i,j,b}}\right)}$

itfollowst hat

$$\left(\frac{\partial^2\varphi(x,y,z)}{\partial z^2}\right)_{i,j,b} = \frac{1}{D_{i,j,b}}\left(a_{i,j,b-u}^{z(-u)}\varphi_{i,j,b-u} - (a_{i,j,b}^{cz(-u)} + a_{i,j,b}^{czu})\varphi_{i,j,b}\right)$$

(A.25)

The diffusion equation is written in operator form as: $M\Phi = \frac{1}{k}F\Phi$ where $\Phi = \lfloor\varphi_{g,i,j,k}\rfloor$ represents a vector whose elements are the flux in group g, at position (i, j, k). M and F represent the multi-group discrete forms of the loss operator, M, and production operator, F, respectively:

$$[M\Phi]_{g,i,j,k} = -[L\Phi] + \Sigma_{tg,i,j,k}\varphi_{g,i,j,k} - \sum_{g'=1}^{G}\Sigma_{sg'\rightarrow g,i,j,k}\varphi_{g'} \quad \textbf{(A.26)}$$

$$[F\Phi]_{g,i,j,k} = \chi_{g,i,j,k}\sum_{g'=1}^{G}\nu\Sigma_{fg',i,j,k}\varphi_{g',i,j,k} \quad \textbf{(A.27)}$$

The leakage operator, L, is defined as the sum of the leakages in each direction:

$$[L\Phi]_{g,i,j,k} = -[L_x\Phi]_{g,i,j,k} - [L_y\Phi]_{g,i,j,k} - [L_y\Phi]_{g,i,j,k} \qquad \textbf{(A.28)}$$

The directional Lx, Ly and Lz operators are defined as:

$$[L_x\Phi]_{g,i,j,k} \equiv \frac{D_{g,i,j,k}}{h_{xi,j,k}}\left[\frac{2D_{g,i+1,j,k}(\varphi_{i+1,j,k} - \varphi_{g,i,j,k})}{h_{xi+1,j,k}D_{g,i,j,k} + h_{xi,j,k}D_{g,i+1,j,k}} - \frac{2D_{g,i-1,j,k}(\varphi_{g,i,j,k} - \varphi_{g,i-1,j,k})}{h_{xi,j,k}D_{g,i-1,j,k} + h_{xi-1,j,k}D_{g,i,j,k}}\right] \qquad \textbf{(A.29a)}$$

$$[L_y\Phi]_{g,i,j,k} \equiv \frac{D_{g,i,j,k}}{h_{yi,j,k}}\left[\frac{2D_{g,i,j+1,k}(\varphi_{i,j+1,k} - \varphi_{g,i,j,k})}{h_{xi,j+1,k}D_{g,i,j,k} + h_{yi,j,k}D_{g,i,j+1,k}} - \frac{2D_{g,i,j-1,k}(\varphi_{g,i,j,k} - \varphi_{g,i,j-1,k})}{h_{yi,j,k}D_{g,i,j-1,k} + h_{yi,j-1,k}D_{g,i,j,k}}\right] \qquad \textbf{(A.29b)}$$

$$[L_z\Phi]_{g,i,j,k} \equiv \frac{D_{g,i,j,k}}{h_{zi,j,k}}\left[\frac{2D_{g,i,j,k+1}(\varphi_{i,j,k+1} - \varphi_{g,i,j,k})}{h_{zi,j,k+1}D_{g,i,j,k} + h_{zi,j,k}D_{g,i,j,k+1}} - \frac{2D_{g,i,j,k-1}(\varphi_{g,i,j,k} - \varphi_{g,i,j,k-1})}{h_{zi,j,k}D_{g,i,j,k-1} + h_{zi,j,k-1}D_{g,i,j,k}}\right] \qquad \textbf{(A.29c)}$$

In the set of Eqns. (A.29) $h_{xi,j,k}$, $h_{yi,j,k}$ and $h_{zi,j,k}$ are mesh sizes of the region indexed by (i,j,k) in x y and z directions respectively; $\Sigma_{tg,i,j,k}$ is the total g-group macroscopic cross section, $\Sigma_{sg'\rightarrow g,i,j,k}$ is the scattering macroscopic cross section from group g' to group g, $\chi_{g,i,j,k}$ is the Chi-fraction of g group neutrons and $\nu\Sigma_{fg',i,j,k}$ is the macroscopic yield cross section of group g neutrons in the region (i,j,k).

For a constant size and uniform diffusion coefficient the multi-group diffusion equation written for the region indexed with (i,j,k) simplifies to:

$$
\begin{aligned}
&-D_g \frac{\varphi_g(i+1,j,k)+\varphi_g(i-1,j,k)-2\varphi_g(i,j,k)}{h_x^2} \\
&-D_g \frac{\varphi_g(i,j+1,k)+\varphi_g(i,j-1,k)-2\varphi_g(i,j,k)}{h_y^2} \\
&-D_g \frac{\varphi_g(i,j,k+1)+\varphi_g(i,j,k-1)-2\varphi_g(i,j,k)}{h_z^2} \\
&+\Sigma_{tg}(i,j,k)\varphi_g(i,j,k)-\sum_{g'=1}^{G}\Sigma_{sg'\to g}(i,j,k)\varphi_{g'}(i,j,k) \\
&=\frac{1}{k}\chi_g(i,j,k)\sum_{g'=1}^{G}\nu\Sigma_{fg'}(i,j,k)\varphi_{g'}(i,j,k)
\end{aligned}
\qquad \textbf{(A.30)}
$$

APPENDIX B

HFEM IMPLEMENTATION DETAILS

B.1 SIMPLE SYSTEM GEOMETRY

To illustrate the indexing details for the three-dimensional HFEM, this section uses a simple model consisting of a three dimensional core, divided into 3x3x3 elements in the x, y and z directions respectively as shown in Fig. (B.1)

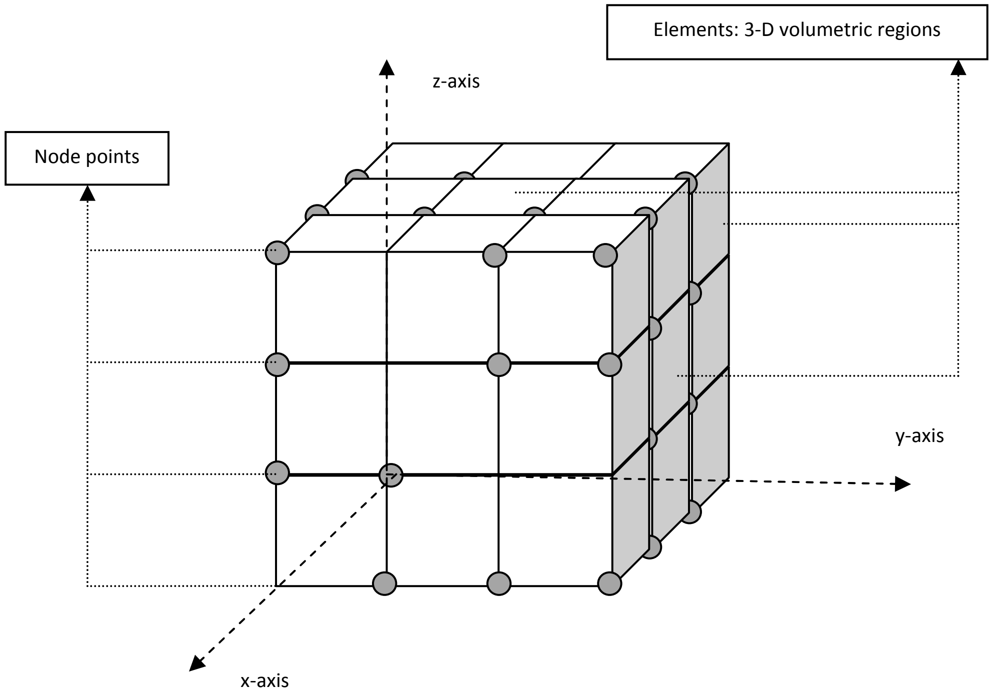


Figure B.1 Three dimensional system with (3x3x3) assemblies

Each element is divided into 3x3x3 subregions, in turn, in x, y and z directions as shown in Fig. (B.2)

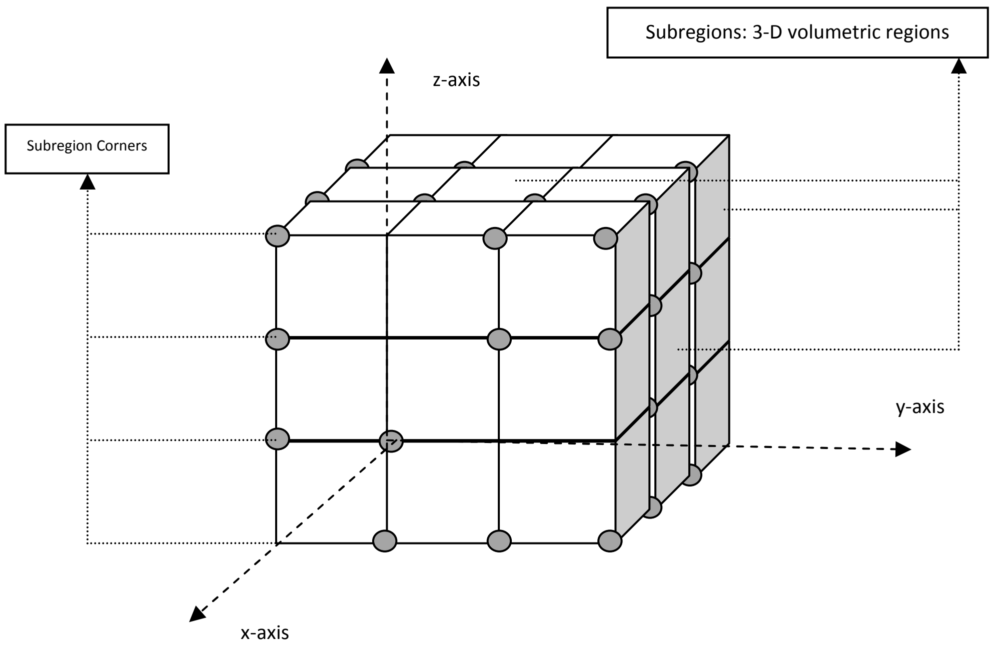


Figure B.2 Three dimensional assembly with (3x3x3) unit cells

B.2 ELEMENT, NODE AND SUBREGION INDEXING

For any geometry, element, subregion and node indexing is required for coding purposes for the defined regions to be referenced. This part explains the indexing used in the HFEM FORTRAN code.

B.2.1 Three-Index Notation

In this notation, elements and nodes are represented by their location with respect to the global origin of the coordinate system of the geometry. The element indexing starts with “1” but the node indexing starts with “0” in each of the three dimensions.

B.2.1 One-Index Notation

In this notation elements are represented by a single index “e” and nodes are represented by a single index “n”. This indexing starts from the origin. First of all the y-z plane element/node layers are indexed for each x level. The y-z plane element/node layers are indexed for each y-level. In each y-layer the elements/nodes in the z direction are indexed first and then the same procedure is repeated for all y-levels from left to right. After one y-z layer is indexed the indexing of a new y-z layer is started by indexing the new x-level from back to front. The only difference between the node and element indexing is that the node indexing starts at “0”, while the element indexing starts at “1”.

An example of three- and one-index notations for the x- layer of elements located at the very back of the geometry is shown in Fig. (B.3). The one-index notation for the nodes at the x-level-1 is depicted in the Fig. (B.4a) and for x-level-2 and x-level 3 the indexing is shown in Fig. (B.4b).

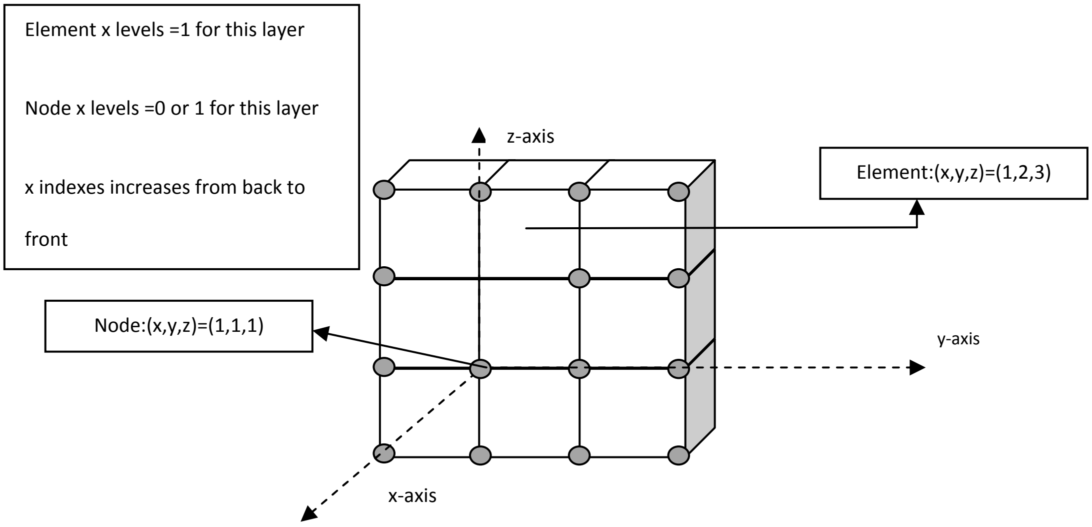

Element x levels =1 for this layer
Node x levels =0 or 1 for this layer
x indexes increases from back to front
z-axis
Element:(x,y,z)=(1,2,3)
Node:(x,y,z)=(1,1,1)
y-axis
x-axis

Figure B.3 Three-index and one-index notation examples for nodes and elements

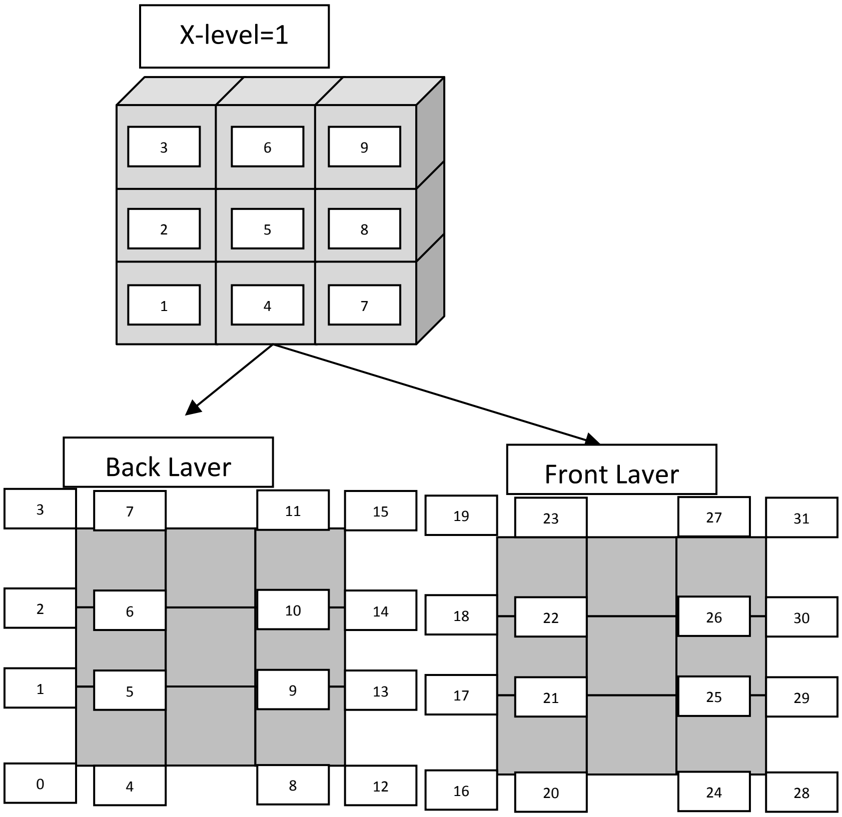


Figure B.4a One-index node numbering for X=1 layer

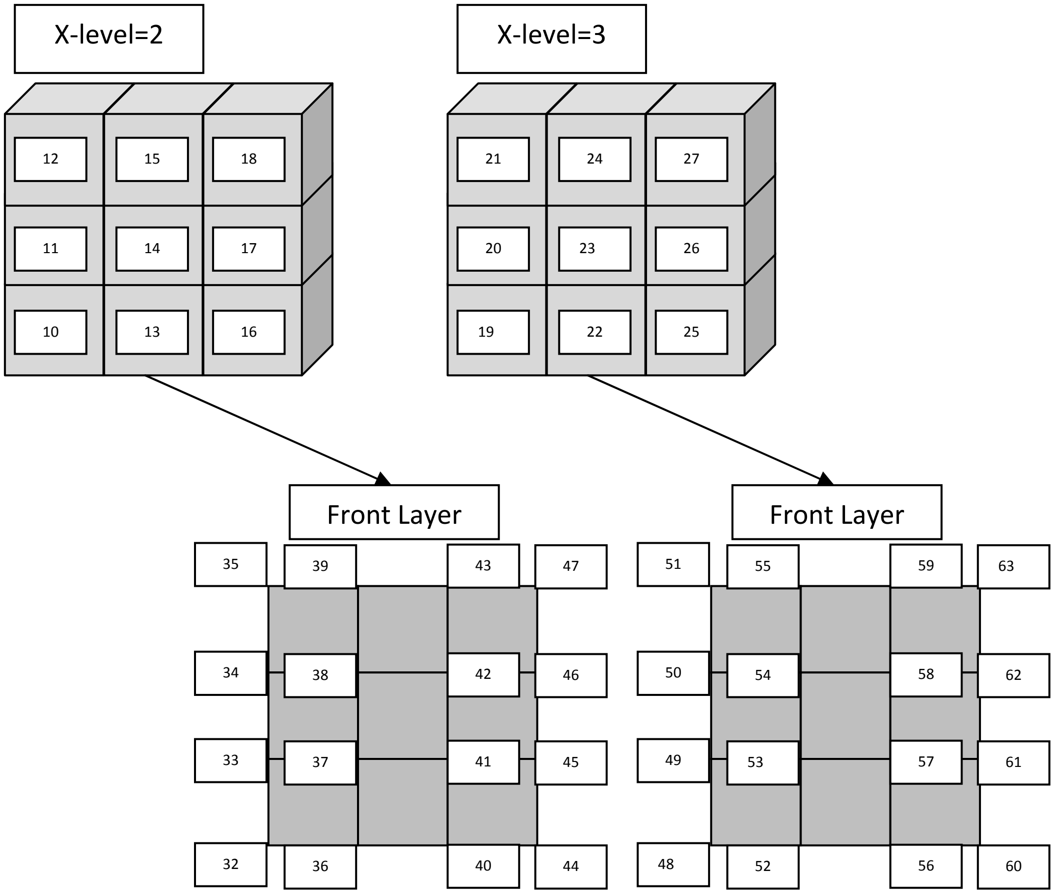


Figure B.4b One-index node numbering for X=2 and X=3 layers

Global subregion three-index notation and one-index notation is done in the same manner such that the global system will include 9 subregions in each of the three directions x, y and z. During the computation local indexing of the subregions inside the elements is defined according to the local element geometry. Therefore, another three-index notation is used, one that indexes the subregions according to the local element coordinates. Local coordinates does not represent the subregion in global sense so to be able to recognize the related subregion the element that the subregion belong is also required. This notation is four indexed notation of subregions.

B.3 MAIN SYSTEM GEOMETRY INPUT

According to the figures showing the system geometry (Figs. (B.1) and (B.2)), the reactor volume is divided into three elements in the x, y and z directions and each of the elements are divided into 3 subregions in the x, y and z directions. So in total there are 27 elements and 729 subregions in the system. According to the given geometry there are a total of 64 nodes that are indexed from 0 to 63. The node index is n. The number of coarse-energy groups is equal to N_G and each coarse-energy group is divided into Ng_G fine-energy groups so that there are Ng fine-energy groups for which the system properties are defined by Eqn. (B.1) in terms of the number of fine groups inside each coarse energy group.

$$Ng = \sum_{G=1}^{NG} Ng_G \tag{B.1}$$

Note that index G is used for the coarse-energy group, while index g is used for the global fine energy group and index g_G is used for the fine groups within each coarse energy group.

B. 4 FLUX EXPANSION:

The continuous space and energy dependence of the flux is approximated by discrete fine-energy groups and piecewise-defined constant fluxes inside each subregion s.

$$\varphi(x,y,z,E) \approx \sum_{g} \left(\underline{\hat{\varphi}}\right)_g \qquad \textbf{(B.2)}$$

In Eq. (B.2) the flux vector $\left(\underline{\hat{\varphi}}\right)_g$ is:

$$\left(\underline{\hat{\varphi}}\right)_g = \begin{bmatrix} \hat{\varphi}_{s=1} \\ \hat{\varphi}_{s=2} \\ \hat{\varphi}_{s=3} \\ \vdots \\ \hat{\varphi}_{s=729} \end{bmatrix}_g = \begin{bmatrix} \hat{\varphi}_{s=1,g} \\ \hat{\varphi}_{s=2,g} \\ \hat{\varphi}_{s=3,g} \\ \vdots \\ \hat{\varphi}_{s=729,g} \end{bmatrix} \qquad \textbf{(B.3)}$$

According to HFEM, the flux expansion as a linear combination of basis functions $\underline{\psi}_n$ is:

$$\left(\underline{\hat{\varphi}}\right)_g = \left(\sum_{n=0}^{63} a_n \underline{\psi}_n\right)_g \qquad \textbf{(B.4)}$$

The node basis function appearing in Eq. (B.4) are defined as:

$$\left(\underline{\psi}_n\right)_g = \begin{bmatrix} \psi_{n,s=1} \\ \psi_{n,s=2} \\ \psi_{n,s=3} \\ \vdots \\ \psi_{n,s=729} \end{bmatrix}_g = \begin{bmatrix} \psi_{n,s=1,g} \\ \psi_{n,s=2,g} \\ \psi_{n,s=3,g} \\ \vdots \\ \psi_{n,s=729,g} \end{bmatrix} \qquad \textbf{(B.5)}$$

Each node basis function $\underline{\psi}_n$ is defined over the neighbouring elements of the node n so that the vector entries of the node function for node n are non-zero only for the subregions that belong to the neighbouring elements of the node so that the vector $\underline{\psi}_n$ is sparse. According to the defined system geometry, for an inner node there are eight element neighbours which include 27 subregions so that the number of non-zero entries for an inner node is 125. If the node is not an inner node such that some of the element neighbouring positions do not include reactor material, the number of non-zero entries is less than 125. For example, node 2 in Fig. (B.4a) has only 4 neighbouring elements so the number of non-zero entries for this node's basis function is less than 125.

The node basis function is a linear combination of eight elementary basis functions defined in each of the neighbouring elements of the node. Elementary basis functions inside the elements are defined to be tri-linear functions having the property of being equal to 1 only at one corner of the element and equal to zero at the other corners. Figure (B.5) defines the corners according to the non-zero elementary basis functions.

For simplicity the numbering of the corner nodes of an element is done according to the number of the elementary basis function which is equal to 1 at the given corner. The corner node numbering is shown in Fig. (B.5).

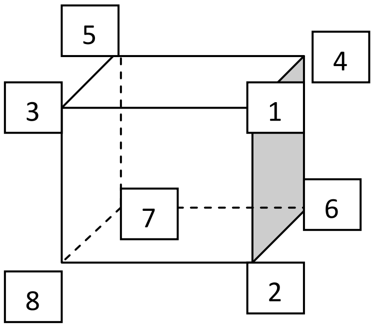


Figure B.5 Node local numbering for a three dimensional element

Neighbouring elements of the nodes are also numbered. Figure B.6 shows the element numbering for an inner node.

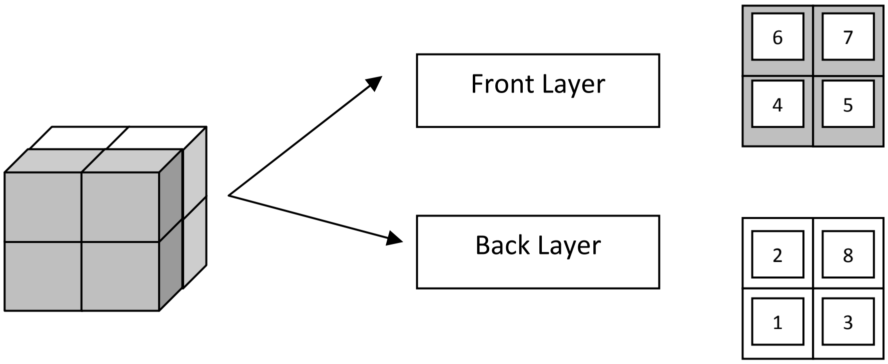


Figure B.6 Numbering scheme for the eight the neighbouring elements in a three dimensional system

The contribution to the inner node basis function is comprised of the non-zero elementary basis functions of the neighbouring elements at the inner-node position. According to this numbering, the non-zero elementary basis-function inside the neighbouring element has the same index number as the neighbouring node:

$$\left(\underline{\psi}_n\right)_g = \begin{pmatrix} \bar{\varphi}_{1\text{ of neighbour }1} + \bar{\varphi}_{2\text{ of neighbour }2} + \bar{\varphi}_{3\text{ of neighbour }3} + \bar{\varphi}_{4\text{ of neighbour }4} + \bar{\varphi}_{5\text{ of neighbour }5} + \bar{\varphi}_{6\text{ of neighbour }6} \\ + \bar{\varphi}_{7\text{ of neighbour }7} + \bar{\varphi}_{8\text{ of neighbour }8} \end{pmatrix}_g \tag{B.6}$$

Or,

$$\left(\underline{\psi}_{n}\right)_{g} = \left(\sum_{i=1}^{8} \underline{\varphi}_{\text{i of neighbour i}}\right)_{g} \qquad \textbf{(B.7)}$$

In Eqn. (B.7) $\underline{\varphi}_{\text{i of neighbour i}}$ is:

$$\left(\underline{\varphi}_{\text{i of neighbour i}}\right)_{g} = \begin{bmatrix} \varphi_{\text{i of neighbour i,s=1}} \\ \varphi_{\text{i of neighbour i,s=2}} \\ \varphi_{\text{i of neighbour i,s=3}} \\ \vdots \\ \varphi_{\text{i of neighbour i,s=729}} \end{bmatrix}_{g} \qquad \textbf{(B.8)}$$

B.5 ELEMENTARY BASIS FUNCTION CALCULATION FOR EACH SUBREGION

There are two types of elementary basis functions used in the solution.

Type 1 elementary basis functions are tri-linear functions which are functions of (x,y,z) defined in each element according to the local coordinate system of the element. Each of these functions is equal to “1” at only one of the local corner node position and equal to “0” at other corners which is illustrated in Fig. (B.8) for a one dimensional geometry with linear elementary basis functions. This local x,y,z dependency of the elementary basis functions is approximated by piecewise constant functions inside each subregion. Subregion constant functions are three dimensional and are equal to the value of the elementary basis function at the center of the subregion.

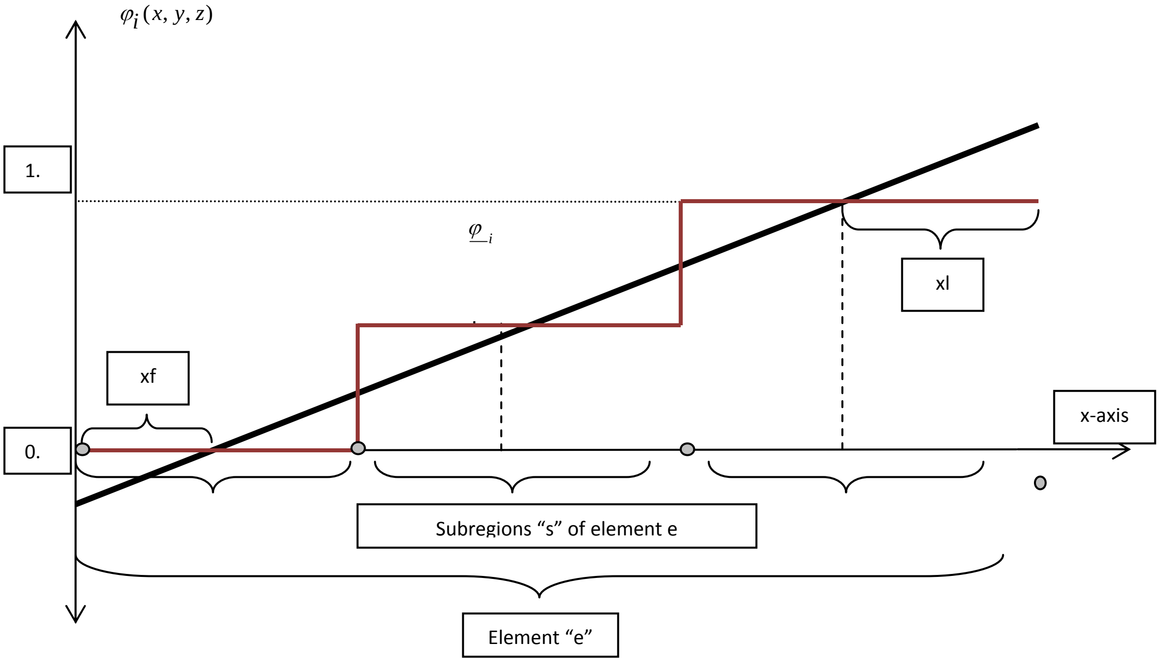


Figure B.8 Type 1 linear elementary basis function

<u>Type 2</u> elementary basis functions which are called “0-corrected elementary basis functions” are again tri-linear functions defined according to the local coordinate system of the element and these tri-linear functions are arranged such that each elementary basis function is equal to “1” at the center of only one corner subregion and “0” at the other corner subregions which can be seen in Fig. (B.9).

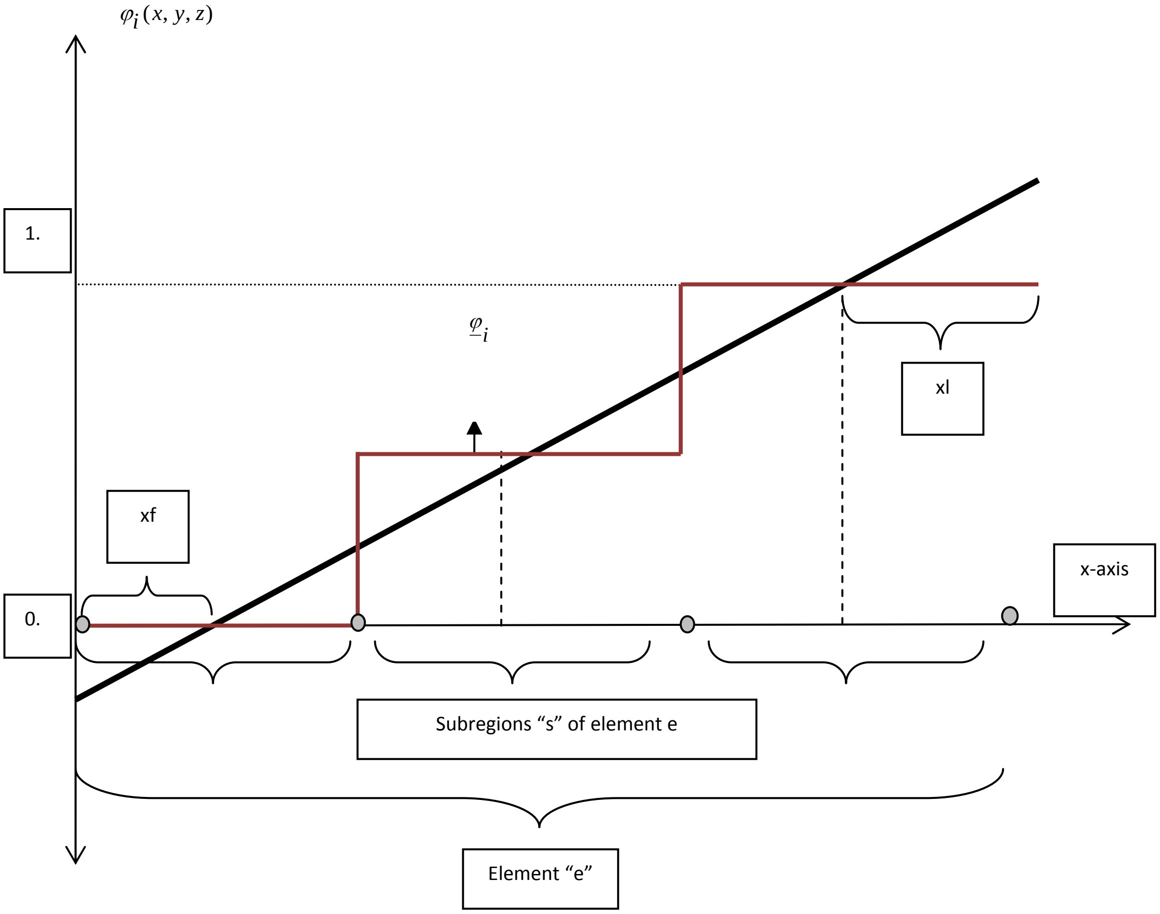


Figure B.9 Type 2 linear elementary basis function

B.5.1 Tri-Linear Elementary Basis Functions Used In The Solution:

B.5.1.1 Type 1 Tri-Linear Elementary Basis Functions:

For any fine energy group ''g'' and element ''e''

$$\left(\varphi_1(x,y,z)\right)_g = \left(\frac{x}{\Delta_{x.element}}\right)\left(\frac{y}{\Delta_{y.element}}\right)\left(\frac{z}{\Delta_{z.element}}\right)$$

$$\left(\varphi_2(x,y,z)\right)_g = \left(\frac{x}{\Delta_{x.element}}\right)\left(\frac{y}{\Delta_{y.element}}\right)\left(1-\frac{z}{\Delta_{z.element}}\right)$$

$$\left(\varphi_3(x,y,z)\right)_g = \left(\frac{x}{\Delta_{x.element}}\right)\left(1.0-\frac{y}{\Delta_{y.element}}\right)\left(\frac{z}{\Delta_{z.element}}\right)$$

$$\left(\varphi_4(x,y,z)\right)_g = \left(1.0-\frac{x}{\Delta_{x.element}}\right)\left(\frac{y}{\Delta_{y.element}}\right)\left(\frac{z}{\Delta_{z.element}}\right)$$

$$\left(\varphi_5(x,y,z)\right)_g = \left(1.0-\frac{x}{\Delta_{x.element}}\right)\left(1.0-\frac{y}{\Delta_{y.element}}\right)\left(\frac{z}{\Delta_{z.element}}\right)$$

$$\left(\varphi_6(x,y,z)\right)_g = \left(1.0-\frac{x}{\Delta_{x.element}}\right)\left(\frac{y}{\Delta_{y.element}}\right)\left(1.0-\frac{z}{\Delta_{z.element}}\right)$$

$$\left(\varphi_7(x,y,z)\right)_g = \left(1.0-\frac{x}{\Delta_{x.element}}\right)\left(1.0-\frac{y}{\Delta_{y.element}}\right)\left(1.0-\frac{z}{\Delta_{z.element}}\right)$$

$$\left(\varphi_8(x,y,z)\right)_g = \left(\frac{x}{\Delta_{x.element}}\right)\left(1.0-\frac{y}{\Delta_{y.element}}\right)\left(1.0-\frac{z}{\Delta_{z.element}}\right)$$

(B.9)

Where (x,y,z) are defined according to the local coordinates of the element ''e''

B.5.1.2 Type 2 Tri-Linear Elementary Basis Functions

For any fine energy group ''g'' and element ''e''

$$
\begin{aligned}
(\varphi_1(x,y,z))_g &= \left(\frac{x-xf}{\Delta_{x.element}-xf-xl}\right)\left(\frac{y-yf}{\Delta_{y.element}-yf-yl}\right)\left(\frac{z-zf}{\Delta_{z.element}-zf-zl}\right)\\
(\varphi_2(x,y,z))_g &= \left(\frac{x-xf}{\Delta_{x.element}-xf-xl}\right)\left(\frac{y-yf}{\Delta_{y.element}-yf-yl}\right)\left(1-\frac{z-zf}{\Delta_{z.element}-zf-zl}\right)\\
(\varphi_3(x,y,z))_g &= \left(\frac{x-xf}{\Delta_{x.element}-xf-xl}\right)\left(1.0-\frac{y-yf}{\Delta_{y.element}-yf-yl}\right)\left(\frac{z-zf}{\Delta_{z.element}-zf-zl}\right)\\
(\varphi_4(x,y,z))_g &= \left(1.0-\frac{x-xf}{\Delta_{x.element}-xf-xl}\right)\left(\frac{y-yf}{\Delta_{y.element}-yf-yl}\right)\left(\frac{z-zf}{\Delta_{z.element}-zf-zl}\right)\\
(\varphi_5(x,y,z))_g &= \left(1.0-\frac{x-xf}{\Delta_{x.element}-xf-xl}\right)\left(1.0-\frac{y-yf}{\Delta_{y.element}-yf-yl}\right)\left(\frac{z-zf}{\Delta_{z.element}-zf-zl}\right)\\
(\varphi_6(x,y,z))_g &= \left(1.0-\frac{x-xf}{\Delta_{x.element}-xf-xl}\right)\left(\frac{y-yf}{\Delta_{y.element}-yf-yl}\right)\left(1.0-\frac{z-zf}{\Delta_{z.element}-zf-zl}\right)\\
(\varphi_7(x,y,z))_g &= \left(1.0-\frac{x-xf}{\Delta_{x.element}-xf-xl}\right)\left(1.0-\frac{y-yf}{\Delta_{y.element}-yf-yl}\right)\left(1.0-\frac{z-zf}{\Delta_{z.element}-zf-zl}\right)\\
(\varphi_8(x,y,z))_g &= \left(\frac{x-xf}{\Delta_{x.element}-xf-xl}\right)\left(1.0-\frac{y-yf}{\Delta_{y.element}-yf-yl}\right)\left(1.0-\frac{z-zf}{\Delta_{z.element}-zf-zl}\right)
\end{aligned}
\tag{B.10}
$$

Where (x,y,z) are defined according to the local coordinates of the element ''e'' and xf, xl, yf, yl, zf, zl are the shift distances of type-2 functions.

HFEM leakage matrix requires the pre-calculation of $\underline{\underline{L}}\underline{\psi}_{g,n'=0,\ldots 63}$ which create the geometric coupling in the system. The leakage matrix is not diagonal because the derivative of the flux at the subregions is approximated by the flux values at the subregion itself and at neighbouring subregions which results in space coupling. The matrix $\underline{\underline{L}}$ is a sparse matrix with a maximum of "7" non-zero entries in each row. The

$\underline{\underline{L}}\underline{\psi}_{g,n'=0,\ldots 63}$ vector is non-zero for the node neighbouring elements subregions and also non-zero for the boundary subregion of the non-neighbouring elements.

This behaviour is described in the set of Figs. (B.10) for a one dimensional geometry:

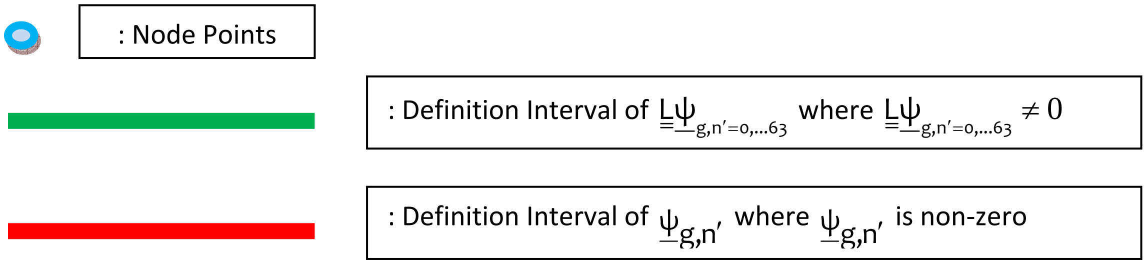

: Node Points
: Definition Interval of $\underline{\underline{L}}\underline{\psi}_{g,n'=0,\ldots 63}$ where $\underline{\underline{L}}\underline{\psi}_{g,n'=0,\ldots 63} \neq 0$
: Definition Interval of $\underline{\psi}_{g,n'}$ where $\underline{\psi}_{g,n'}$ is non-zero

Figure B.10a Explanation of symbols for Fig. B.10b

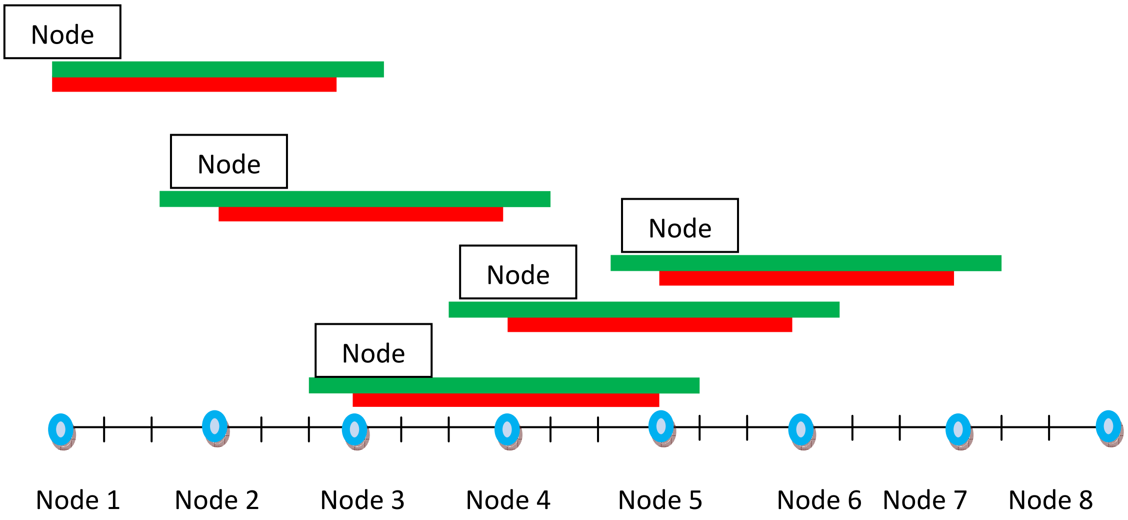


Figure B.10b Space Coupling Behaviour in 1 dimensional geometry for type 1 basis functions

According to the given definitions for the intervals in the Fig. (B.10b), node 4 couples with node 5 and node 3 which have common neighbouring elements so that these nodes are coupled to the subregions of the common neighbouring elements and they are the first neighbours of node 4. On the other hand, node 4 is coupled with node 2 and node 6 which are the second neighbouring nodes of node 4 which do not have common neighbouring elements but the nodes are coupled because of the neighbouring subregions of the boundary subregions of node 4. And of course node 4 is coupled with itself.

By considering the coupling behaviour of each node with the first and second neighbouring nodes then it can be concluded that the matrix of HFEM system includes a maximum 125 non-zero entries for each row which are the total number of first and second neighbouring nodes and of the node itself.

If type-2 elementary basis functions are used, the definition of the node basis functions will be non-zero through the inner subregions of the node neighbouring elements. So, when the dependency of the subregion on the neighbouring subregions for the subregion leakage is introduced, the last subregion that has non-zero node basis function will have a leakage written in terms of its neighbouring subregions. Since the last subregion that has non-zero node basis functions is not a boundary subregion of the node then it`s neighbours will be inside the node neighbouring elements subregions. Therefore that the leakage by using type 2 elementary basis functions will be coupled over only 27 nodes which is the total number of first neighbouring nodes and the node itself. Because of this behaviour, in HFEM leakage matrix there are maximum 27 non-zero terms per each row which actually means each node is coupled with only 27 nodes.

(whose $\underline{\underline{L}}\underline{\psi}_{g,n'=0,\ldots,63}$ vector gives non-zero inner product when multiplied with the node basis function $\underline{\psi}_{g,n}$).

This behaviour is described in the set of Figs. (B11) for a one dimensional geometry:

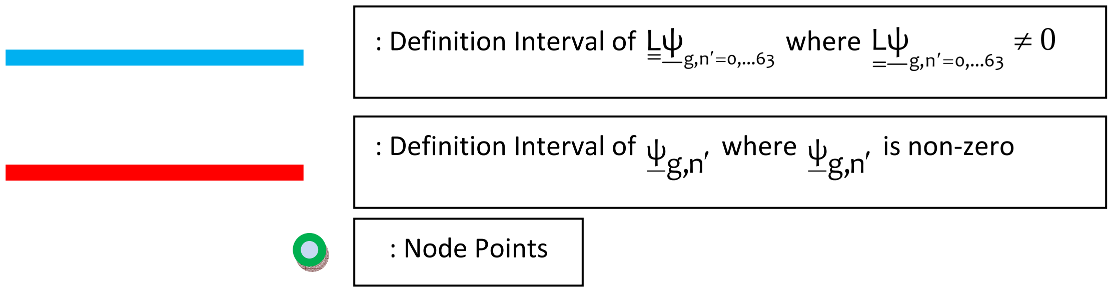

: Definition Interval of $\underline{\underline{L}}\underline{\psi}_{g,n'=0,\ldots,63}$ where $\underline{\underline{L}}\underline{\psi}_{g,n'=0,\ldots,63} \neq 0$
: Definition Interval of $\underline{\psi}_{g,n'}$ where $\underline{\psi}_{g,n'}$ is non-zero
: Node Points

Figure B.11a Explanation of symbols for Fig. B11.b

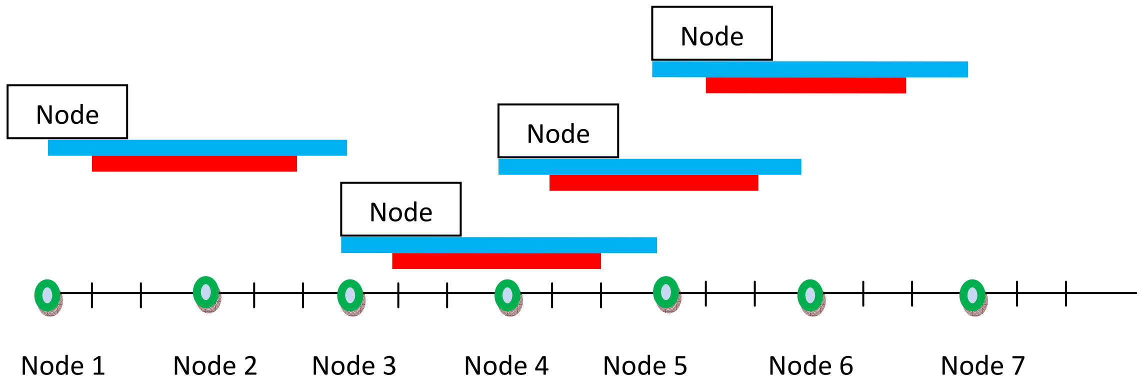


Figure B.11b Space Coupling Behaviour in 1 dimensional geometry for type 1 basis functions

APPENDIX C

ADDITIONAL RESULTS FOR THE TEST SYSTEM

For a more extensive results comparison, two additional sets of calculations were performed, both using the code DONJON (Varin, 2004, 2005).

SET 4: DONJON Fine-group, coarse-mesh homogeneous finite element method (DONJON-FEM) solution. This solution is performed by DONJON using the usual finite element method with tri-linear basis functions in three energy groups.

SET 5: DONJON Fine-group, coarse-mesh finite-difference method (DONJON-CMFDM). This solution is performed by DONJON using finite differences for three groups by using coarse meshes which are the system assemblies.

K-eff results are shown in Table C.1, together with the computation time.

Table C.1: Comparison of k-eff and execution times for different methods

METHOD	keff	difference in mk	Time (min)
FDM	0.99012	-	4300
HFEM	0.98935	-0.78605	16
DONJON FEM	0.99311	3.04079	1
DONJON CMFDM	0.99021	0.09180	1

Table C.2 shows the errors obtained using each of the methods.

Table C.2: Comparison of normalized root mean square percent errors

FLUX	DONJON FEM (RMSE%)	DONJON FDM (RMSE%)	HFEM (RMSE%)
Group 1-x direction	3.5389	3.7441	0.3781
Group 1-y direction	3.5869	3.6005	0.5085
Group 1-z direction	1.2860	1.9899	0.9312
Group 2-x direction	3.5400	3.7391	0.3665
Group 2-y direction	3.5811	3.5974	0.4991
Group 2-z direction	1.2934	1.9863	0.9211
Group 3-x direction	3.7628	3.7628	0.4305
Group 3-y direction	3.6762	3.6762	0.6298
Group 3-z direction	2.0880	2.0880	1.1418
Overall	2.9282	3.1316	0.6452

Results in Tables C.1 and C.2 show the current implementation of HFEM to perform better than the regular FEM and the coarse-mesh FD in terms of accuracy while at the same time, taking a substantially longer time (16 times longer). This reinforces the need to optimize the convergence criteria and to attempt to implement acceleration techniques.