2003 추계학술발표회 논문집
한국원자력학회

성분 분해법으로 유도된 개량 AFEN을 이용한 객체 지향 노달 코드 개발

Development of an Object Oriented Nodal Code with the Refined AFEN Derived by the Method of Component Decomposition

노재만, 유재운, 주형국

한국원자력연구소
대전광역시 유성구 덕진동 150

요 약

본 연구에서는 성분 분해법을 개발하여 개량 해석 함수 전개 노달 방법의 노드간 연관 방정식을 체계화하였고 이들을 푸는 객체 지향 노달 코드를 개발하였다. 성분 분해법은 노달 방법에서의 노드내 중성자속 전개 함수를 각 방향별 우함수 혹은 기함수의 성분들로 분해하여 한 개의 큰 선형 방정식 시스템을 여러 개의 단일 방정식으로 바꾼다. 이 때 유도된 연관 방정식 자체가 다중 소격격자 가속 방정식이 되어서 별도의 가속 기법의 개발이 필요 없다. 또 객체 지향 노달 코드는 추상화, 은닉, 상속과 다중성, 동적 기억 장소 할당, 연산자 중복 정의 등의 객체 지향 프로그램 개념을 충분히 활용하여 입출력이 편리하고 기억 장소 관리가 유연하며 유지 보수가 매우 편리하다.

Abstract

In this study, the method of component decomposition was invented to derive the systematic internode coupled equations of the refined AFEN method and an object oriented nodal code was developed to solve the derived internodal coupled equations. The method of component decomposition decomposes the intranodal flux expansion of a nodal method into the even and odd components in three dimensions to reduce a large size coupled linear system equation into several small size single equations. This method requires no additional technique to accelerate the iteration process to solve the internodal coupled equations, since the derived equations can automatically act as the coarse mesh rebalance equations. Utilizing fully the object oriented programming concepts such as abstraction, encapsulation, inheritance and polymorphism, dynamic memory allocation, and operator overloading, the object oriented nodal code developed here can facilitate the input/output and the dynamic control of memories, and can make the maintenance easy.

1. 서 론

본 연구에서 개발한 객체 지향 노달 코드인 NODOO (Object Oriented Nodal)는 노심의 중성자속 계산의 정확도를 높이기 위하여 아주 최근에 개발된 개량 해석함수 전개 노달 방법(Refined AFEN)[1]을 사용하였다. 이 방법은 노드내의 중성자속 전개에 횡방향 경사 기저 함수를 추가하여 기존의 해석함수 전개 노달 방법(AFEN)을 개량한 것이다. 개량 AFEN 방법에 추가된 기저 함수는 원래 AFEN의 한 방향으로의 해에다 그 방향과 수직인 횡방향 일차함수를 곱한 것들, 예를 들면 y sin(kx), z sin(kx)과 yz sin(kx)이다. 이 새로 추가된 함수들도 원래 AFEN 전개 함수와 마찬가지로 중성자 확산 방정식을 만족한다. 새 전개 함수에 대응하여 추가된 노달 변수는 경계면 가중 중성자속이다. 본 연구에서는 가중 함수로 물리적 의미가 명확한 계단 함수를 사용하였다. 또 가중 중성자속를 도입함으로써 충분한 정확도를 유지할 수 있기 때문에 기존 AFEN의 노달 미지수인 격자점 중성자속을 사용하지 않았다. 또 AFEN 방법이나 해석적 노달방법(ANM: Aalytic Nodal Method)에서 특정 노드의 순 누출이 없을 때 생길 수 있는 수치적 특이성을 제거하기 위하여 최근에 개발된 Continued Factoring 방법)[2]을 도입하였다.

본 연구에서는 각종 노달 방법에 관련되는 복잡한 식을 쉽게 취급할 수 있도록 성분 분해법(MCD법: Method of Component Decomposition)을 개발하였다. 이 방법은 노드내 3차원 중성자속 분포를 x, y와 z에 대해 우함수와 기함수인가에 따라 8개의 성분들로 분해한다. 이렇게 하면 서로 연관된 많은 선형 방정식으로 구성된 시스템을 여러 개의 개별 방정식으로 나눌 수 있어서 그 방정식을 컴퓨터로 풀 때 요구되는 노력을 경감할 수 있다. 또 유도되는 성분별 중성자속 균형 방정식 자체가 소격격자 재균형식이 되므로 별도의 다른 가속 기법을 도입할 필요가 없다.

본 연구에서 개발한 노달 코드 NODOO는 순차적 프로그램(Sequential Program) 기법을 이용하는 기존의 노달 코드와는 달리 추상화, 자료의 은익, 상속과 다중성, 동적 기억 장소 할당, 연산자 중복 정의 등의 객체 지향 프로그램 고유 개념을 충분히 활용하는 객체 지향 프로그램이다. 노달 코드가 하는 일을 기능적으로 구분하여 될 수 있는 데로 독립된 입출력, 에너지군 구조 처리, 단면적 처리, 노심 구조 처리 모듈로 나누고 모듈간의 연계는 최소화하며 그 마저도

자료 자체가 아니라 연계 함수(Interface Function)로만 이루어진다. 예를 들면 에너지 군에 대한 반복 루프는 에너지군 처리 모듈에만 나타나고 다른 모듈에는 나타나지 않는다. 입력은 입력 변수에 대한 설명까지 덧붙일 수 있도록 신문처럼 많은 산문 속에 순차적으로 나타나는 수치를 읽을 수 있게 하였다. 출력은 이용자가 기존의 노달 코드의 출력 양식에 익숙해 있는 것을 감안하여 기존 양식을 유지하려고 하였다. 기억 장소는 동적으로만 할당하며 존재하는 물리 변수의 중복 정의를 않으며 그 것에 접근하는 방식만 다양화 한다. 기존 프로그램들이 대부분 비슷한 수식들이 수많은 if문과 함께 프로그램의 여러 부분에 산개해 있는 것과는 달리 원래 한 개만 있는 수식은 프로그램의 단 한 군데만 나타나게 하여 유지 보수를 매우 쉽게 하였다. 실제 노심 구성 원리대로 연료봉, 핵연료, 노심을 구성하고 노심의 확장, 축소, 복제, 1차원 투영, 2차원 투영 등이 자유롭게 하였다.

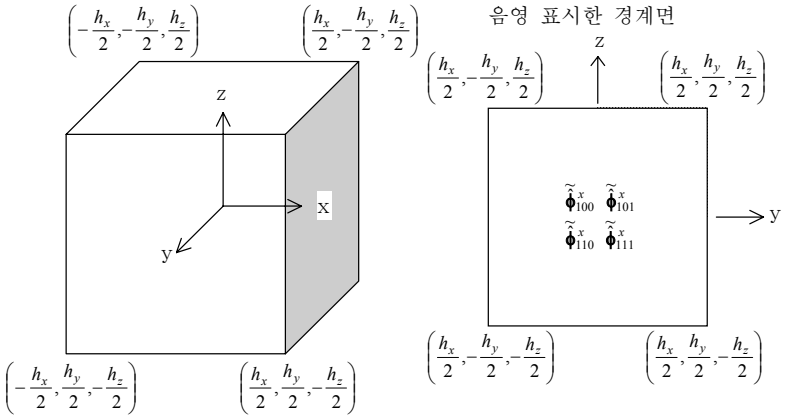
2. 본 론

2.1 성분 분해법

성분 분해법을 이용한 노달 방법 전개는 개량 AFEN 뿐만 아니라 다른 노달 방법에도 적용하는 것이 가능하다. 만일 NEM에 적용하면 현재 다른 노달 코드들에서 사용되는 형태보다 훨씬 간단한 형태의 연관 방정식을 얻을 수 있다.

먼저 그림 1과 같은 3차원 노드의 다에너지군 확산 방정식을 생각하자.

$$-\nabla^2\hat{\boldsymbol{\phi}}(\mathbf{r})+\Lambda\hat{\boldsymbol{\phi}}(\mathbf{r})=0 \qquad (1)$$



음영으로 표시한 경계면의 가중 중성자속

$$\tilde{\hat{\boldsymbol{\phi}}}^{x}_{1jk}=\frac{1}{h_y h_z}\int_{-\frac{h_y}{2}}^{\frac{h_y}{2}}\int_{-\frac{h_z}{2}}^{\frac{h_z}{2}} \mathrm{w}_j(y)\mathrm{w}_k(z)\,\hat{\boldsymbol{\phi}}\left(\frac{h_x}{2},y,z\right)dzdy$$

그림 1. 3차원 노드 구조

(1)식의 우변에 노드내의 단면적 분포에 의한 고정 선원 분포나 시간 천이 선원 분포 등이 있을 때도 지금부터의 논의는 그대로 적용된다.

개량 AFEN에서는 이 노드의 중성자속을 최대 72개의 기저함수로 전개가 가능하고 참고도서 [1]에도 48개의 기저함수로 전개하였지만 NODOO에서는 24개의 기저함수만을 사용하였다.

$$\begin{aligned}
&\hat{\phi}(x,y,z) = \\
&\cosh\left(\sqrt{\Lambda}x\right)\left(\mathbf{A}^x_{000}+2\frac{y}{h_y}\mathbf{A}^x_{010}+2\frac{z}{h_z}\mathbf{A}^x_{001}+4\frac{y}{h_y}\frac{z}{h_z}\mathbf{A}^x_{011}\right)+\sinh\left(\sqrt{\Lambda}x\right)\left(\mathbf{A}^x_{100}+2\frac{y}{h_y}\mathbf{A}^x_{110}+2\frac{z}{h_z}\mathbf{A}^x_{101}+4\frac{y}{h_y}\frac{z}{h_z}\mathbf{A}^x_{111}\right)\\
&+\cosh\left(\sqrt{\Lambda}y\right)\left(\mathbf{A}^y_{000}+2\frac{z}{h_z}\mathbf{A}^y_{001}+2\frac{x}{h_x}\mathbf{A}^y_{100}+4\frac{z}{h_z}\frac{x}{h_x}\mathbf{A}^y_{101}\right)+\sinh\left(\sqrt{\Lambda}y\right)\left(\mathbf{A}^y_{010}+2\frac{z}{h_z}\mathbf{A}^y_{011}+2\frac{x}{h_x}\mathbf{A}^y_{110}+4\frac{z}{h_z}\frac{x}{h_x}\mathbf{A}^y_{111}\right)\\
&+\cosh\left(\sqrt{\Lambda}z\right)\left(\mathbf{A}^z_{000}+2\frac{x}{h_x}\mathbf{A}^z_{100}+2\frac{y}{h_y}\mathbf{A}^z_{010}+4\frac{x}{h_x}\frac{y}{h_y}\mathbf{A}^z_{110}\right)+\sinh\left(\sqrt{\Lambda}z\right)\left(\mathbf{A}^z_{001}+2\frac{x}{h_x}\mathbf{A}^z_{101}+2\frac{y}{h_y}\mathbf{A}^z_{011}+4\frac{x}{h_x}\frac{y}{h_y}\mathbf{A}^z_{111}\right), \qquad (2)
\end{aligned}$$

이 식에 나오는 행렬함수들은 행렬함수 이론[3]을 이용하여 계산한다. 각 계수의 위 첨자는 그 기저함수가 갖는 hyperbolic 함수가 x, y, 와 z 중 어느 것에 의존하는가를 가리키고 아래 첨자는 계수에 붙는 기저 함수가 순서대로 x, y, 와 z에 대해 우함수인가 기함수인가를 나타낸다. 예를 들면 계수 $\mathbf{A}^z_{100}$에 대응하는 기저 함수는 z방향으로의 우함수인 hyperbolic 함수를 갖고 x에 대해서는 기함수, y에 대해서는 우함수인 기저 함수이다.

중성자속 전개인 식 (2)에 있는 24개의 계수는 6개의 경계면 중성자속과 18개의 경계면 가중 중성자속들로 표현한다. 예를 들어 그림 1에 보인 3차원 노드의 x축과 직각인 두 경계면 중 오른쪽 경계면에서 중성자속과 가중 중성자속은

$$\tilde{\hat{\phi}}^x_{1jk} = \frac{1}{h_y h_z}\int_{-\frac{h_y}{2}}^{\frac{h_y}{2}}\int_{-\frac{h_z}{2}}^{\frac{h_z}{2}} \mathrm{w_j}(y)\mathrm{w_k}(z)\,\hat{\phi}(\frac{h_x}{2},y,z)dzdy\,, \qquad (3)$$

이다. 여기서 아래첨자 1jk의 1은 x축상의 오른쪽 경계면을 나타내고 (왼쪽 경계면은 0이다.) j와 k는 각각 우항인지 기항인지를 구별하기 위하여 0과 1이 될 수 있는 수이다. 본 연구에서는 가중 중성자속을 정의할 때 계단 함수를 가중함수로 사용하므로 $\mathrm{w_i}(u)$는

$$\mathrm{w}_0(u)=1, \qquad \mathrm{w}_1(u)=\begin{cases} 1 & \mathrm{u}>0\text{일때} \\ 0 & \mathrm{u}=0\text{일때} \\ -1 & \mathrm{u}<0\text{일때} \end{cases} \qquad (4)$$

로 정의된다. j와 k가 모두 0이 되면 식 (3)은 경계면 평균 중성자속이 되고 어느 하나라도 1이 되면 경계면 가중 중성자속이 된다. 지금부터는 기존의 경계면 평

균 중성자속도 단위 함수를 가중함수로 사용한 경계면 가중 중성자속으로 생각할 수 있으므로 서로 구분하지 않고 이 둘 모두를 가중 중성자속이라고 부른다.

같은 방법으로 위 $\tilde{\hat{\boldsymbol{\phi}}}^{y}_{100}$은 y축에 수직인 두 개의 경계면 중에서 왼쪽 경계면에서 $w_1(x)w_0(z)$로 가중한 평균 중성자속을 의미한다.

식 (3)에 대응하는 경계면에서 노드로 들어오는 가중 중성자류는

$$\tilde{\mathbf{J}}^{x}_{1jk} = \frac{\mathbf{D}}{h_y h_z}\int_{-\frac{h_y}{2}}^{\frac{h_y}{2}}\int_{-\frac{h_z}{2}}^{\frac{h_z}{2}} w_j(y)w_k(z)\frac{\partial}{\partial x}\hat{\boldsymbol{\phi}}(x,y,z)dzdy\Big|_{x=\frac{h_x}{2}} \tag{5}$$

로 정의된다.

식 (3)와 식 (5)과 같이 경계면에서 횡방향 계단함수로 가중한 물리량을 정의하면 이 경계면을 y축과 z축으로 분할할 때 만들어지는 네 사분면에서 단순 평균한 물리량을 이들 가중 물리량들의 선형 결합으로 표시할 수 있다. 예를 들면 y>0 및 z>0인 사분면(그림 1에서 음영으로 표시한 부분)의 중성자속 평균은

$$\frac{\mathbf{1}}{\mathbf{4}}\left(\tilde{\hat{\boldsymbol{\phi}}}^{x}_{100}+\tilde{\hat{\boldsymbol{\phi}}}^{x}_{101}+\tilde{\hat{\boldsymbol{\phi}}}^{x}_{110}+\tilde{\hat{\boldsymbol{\phi}}}^{x}_{111}\right) \tag{6}$$

로 표시된다. 따라서 경계면 가중 물리량들은 결국 네 사분면 각각에서의 물리량과 수학적으로 동치이다.

성분 분해법(MCD법: Method of Component Decomposition)은 중성자속 전개 (2)를 x, y, 와 z에 대하여 우함수 혹은 기함수인가에 따라 크게 8개의 성분으로 분해한다. 예를 들어 x에 대해 기함수이고 y와 z에 대해서는 우함수인 성분, 즉 100성분은

$$\begin{aligned}\hat{\boldsymbol{\phi}}_{100}(x,y,z) &= \frac{1}{8}\{\hat{\boldsymbol{\phi}}(x,y,z)-\hat{\boldsymbol{\phi}}(-x,y,z)+\hat{\boldsymbol{\phi}}(x,-y,z)+\hat{\boldsymbol{\phi}}(x,y,-z)\\ &\quad -\hat{\boldsymbol{\phi}}(-x,-y,z)+\hat{\boldsymbol{\phi}}(x,-y,-z)-\hat{\boldsymbol{\phi}}(-x,y,-z)-\hat{\boldsymbol{\phi}}(-x,-y,-z)\}\\ &= \sinh\left(\sqrt{\boldsymbol{\Lambda}}x\right)\mathbf{A}^{x}_{100}+2\frac{x}{h_x}\cosh\left(\sqrt{\boldsymbol{\Lambda}}y\right)\mathbf{A}^{y}_{100}+2\frac{x}{h_x}\cosh\left(\sqrt{\boldsymbol{\Lambda}}z\right)\mathbf{A}^{z}_{100}\end{aligned} \tag{7}$$

이다. 이렇게 나누는 이유는 24×24인 선형계 (24개 미지수를 가진 24개 선형방정식을 갖는 계)를 몇 개의 중간 단계를 거쳐 최종적으로 24개의 개별 방정식과 8개의 개별 보조 방정식으로 나누기 위해서이다.

이제 식 (7)의 계수를 결정하자. 먼저 각 성분에 해당하는 가중 평균 중성자속을 다음과 같이 정의하자.

$$\bar{\boldsymbol{\phi}}_{ijk} = \frac{1}{h_x h_y h_z}\int_{-\frac{h_x}{2}}^{\frac{h_x}{2}}\int_{-\frac{h_y}{2}}^{\frac{h_y}{2}}\int_{-\frac{h_z}{2}}^{\frac{h_z}{2}} \mathrm{w}_{\mathrm{i}}(x)\mathrm{w}_{\mathrm{j}}(y)\mathrm{w}_{\mathrm{k}}(z)\,\hat{\boldsymbol{\phi}}(x,y,z)dzdydx\,, \tag{8}$$

그러면 기존의 노드 평균 중성자속은 000 성분의 가중 평균 중성자속임을 알 수 있다.

또 앞서 예를 든 100성분에 해당하는 가중 평균 중성자속은

$$\begin{aligned}\bar{\boldsymbol{\phi}}_{100} &= \frac{1}{h_x h_y h_z}\int_{-\frac{h_x}{2}}^{\frac{h_x}{2}}\int_{-\frac{h_y}{2}}^{\frac{h_y}{2}}\int_{-\frac{h_z}{2}}^{\frac{h_z}{2}} \mathrm{w}_1(x)\mathrm{w}_0(y)\mathrm{w}_0(z)\,\hat{\boldsymbol{\phi}}(x,y,z)dzdydx \\ &= \frac{\sinh^2(\frac{\sqrt{\boldsymbol{\Lambda}}h_x}{4})}{\frac{\sqrt{\boldsymbol{\Lambda}}h_x}{4}}\mathbf{A}^x_{100} + \frac{1}{2}\left\{\frac{\sinh(\frac{\sqrt{\boldsymbol{\Lambda}}h_y}{2})}{\frac{\sqrt{\boldsymbol{\Lambda}}h_y}{2}}\mathbf{A}^y_{100} + \frac{\sinh(\frac{\sqrt{\boldsymbol{\Lambda}}h_z}{2})}{\frac{\sqrt{\boldsymbol{\Lambda}}h_z}{2}}\mathbf{A}^z_{100}\right\}\end{aligned} \tag{9}$$

로 나타난다. 앞서 경계면에 대한 가중 중성자속 경우에서와 같은 논의를 거치면 이 8개의 성분의 가중 평균 중성자속들은 x축, y축과 z축으로 분할되는 8개의 8분 육면체에 대한 중성자속 평균들과 수학적으로 동치이다.

다음에는 경계면 가중 중성자속 (3) 와 성분별 가중 평균 중성자속 (8)으로부터 성분별 방향별 중성자속들을 정의한다. 예를 들면 100성분의 방향별 중성자속들은 다음 식들에서 보이는 형태로 정의된다.

$$\hat{\boldsymbol{\omega}}^x_{100} = \frac{1}{2}\left(\tilde{\hat{\boldsymbol{\phi}}}^x_{100} - \tilde{\hat{\boldsymbol{\phi}}}^x_{000}\right) - 2\bar{\boldsymbol{\phi}}_{100} = \left\{\sinh(\frac{\sqrt{\boldsymbol{\Lambda}}h_x}{2}) - \frac{2\sinh^2(\frac{\sqrt{\boldsymbol{\Lambda}}h_x}{4})}{\frac{\sqrt{\boldsymbol{\Lambda}}h_x}{4}}\right\}\mathbf{A}^x_{100}, \tag{10}$$

$$\hat{\boldsymbol{\omega}}^y_{100} = \frac{1}{2}\left(\tilde{\hat{\boldsymbol{\phi}}}^y_{110} + \tilde{\hat{\boldsymbol{\phi}}}^y_{100}\right) - \bar{\boldsymbol{\phi}}_{100} = \frac{1}{2}\left\{\cosh(\frac{\sqrt{\boldsymbol{\Lambda}}h_y}{2}) - \frac{\sinh(\frac{\sqrt{\boldsymbol{\Lambda}}h_y}{2})}{\frac{\sqrt{\boldsymbol{\Lambda}}h_y}{2}}\right\}\mathbf{A}^y_{100}, \tag{11}$$

$$\hat{\boldsymbol{\omega}}^z_{100} = \frac{1}{2}\left(\tilde{\hat{\boldsymbol{\phi}}}^z_{101} + \tilde{\hat{\boldsymbol{\phi}}}^z_{100}\right) - \bar{\boldsymbol{\phi}}_{100} = \frac{1}{2}\left\{\cosh(\frac{\sqrt{\boldsymbol{\Lambda}}h_z}{2}) - \frac{\sinh(\frac{\sqrt{\boldsymbol{\Lambda}}h_z}{2})}{\frac{\sqrt{\boldsymbol{\Lambda}}h_z}{2}}\right\}\mathbf{A}^z_{100}\,. \tag{12}$$

이 식들 보면, 성분별 방향별 중성자속이 한 개의 계수만을 포함하고 있다. 즉, 처음 24×24 선형계가 24개의 개별 방정식들로 분해된 것이다.

다음으로 성분별 방향별 중성자류와 중성자속의 연관 방정식을 만들기 위해 성분별 중성자속과 마찬가지로 각 성분별 방향별 중성자류를 다음의 100 성분의 예와 같이 정의하자.

$$\boldsymbol{\eta}_{100}^{x}=\frac{1}{2}\left(\tilde{\hat{\mathbf{J}}}_{100}^{x}-\tilde{\hat{\mathbf{J}}}_{000}^{x}\right)+\frac{4\mathbf{D}}{h_x}\bar{\boldsymbol{\phi}}_{100}=-\frac{2\mathbf{D}}{h_x}\left\{\frac{\sqrt{\boldsymbol{\Lambda}}h_x}{2}\cosh(\frac{\sqrt{\boldsymbol{\Lambda}}h_x}{2})-\frac{2\sinh^2(\frac{\sqrt{\boldsymbol{\Lambda}}h_x}{4})}{\frac{\sqrt{\boldsymbol{\Lambda}}h_x}{4}}\right\}\mathbf{A}_{100}^{x}, \tag{13}$$

$$\boldsymbol{\eta}_{100}^{y}=\frac{1}{2}\left(\tilde{\hat{\mathbf{J}}}_{110}^{y}+\tilde{\hat{\mathbf{J}}}_{100}^{y}\right)=-\frac{\mathbf{D}}{h_y}\frac{\sqrt{\boldsymbol{\Lambda}}h_y}{2}\sinh(\frac{\sqrt{\boldsymbol{\Lambda}}h_x}{2})\mathbf{A}_{100}^{y}, \tag{14}$$

$$\boldsymbol{\eta}_{100}^{z}=\frac{1}{2}\left(\tilde{\hat{\mathbf{J}}}_{101}^{z}+\tilde{\hat{\mathbf{J}}}_{100}^{z}\right)=-\frac{\mathbf{D}}{h_z}\frac{\sqrt{\boldsymbol{\Lambda}}h_z}{2}\sinh(\frac{\sqrt{\boldsymbol{\Lambda}}h_z}{2})\mathbf{A}_{100}^{z}. \tag{15}$$

이제 계수를 소거하고 성분별 방향별 중성자류를 성분별 방향별 중성자속으로 표현하자.

$$\boldsymbol{\eta}_{100}^{x}=\boldsymbol{\tau}_{x1}\hat{\boldsymbol{\omega}}_{100}^{x}, \tag{16}$$

$$\boldsymbol{\eta}_{100}^{y}=\boldsymbol{\tau}_{y0}\hat{\boldsymbol{\omega}}_{100}^{y}, \tag{17}$$

$$\boldsymbol{\eta}_{100}^{z}=\boldsymbol{\tau}_{z0}\hat{\boldsymbol{\omega}}_{100}^{z}, \tag{18}$$

여기서 u=x, y, 혹은 z에 대하여

$$\boldsymbol{\tau}_{u0}=\frac{\mathbf{D}}{h_u}\frac{\boldsymbol{\Lambda}_u(\boldsymbol{\rho}_{u1}+1)}{2\boldsymbol{\rho}_{u1}}, \tag{19}$$

$$\boldsymbol{\tau}_{u1}=-\frac{2\mathbf{D}}{h_u}\left\{\frac{\boldsymbol{\rho}_{u1}}{(\boldsymbol{\rho}_{u1}+1)\boldsymbol{\rho}_{u2}}+1\right\}, \tag{20}$$

$$\boldsymbol{\rho}_{u1}=\frac{\tanh(\frac{\sqrt{\boldsymbol{\Lambda}}h_u}{2})}{\frac{\sqrt{\boldsymbol{\Lambda}}h_u}{2}}\text{-}1, \tag{21}$$

$$\boldsymbol{\rho}_{u2}=\frac{\tanh(\frac{\sqrt{\boldsymbol{\Lambda}}h_u}{4})}{\frac{\sqrt{\boldsymbol{\Lambda}}h_u}{4}}\text{-}1, \tag{22}$$

$$\boldsymbol{\Lambda}_u=\boldsymbol{\Lambda}h_u^2, \tag{23}$$

이다.

아직 모든 연관 방정식을 다 얻은 것은 아니다. 성분별 방향별 중성자속과 중성자류를 정의할 때 성분별 평균 중성자속을 도입했음을 상기하면 성분별 평균 중성자속을 구할 수 있는 방정식이 필요하다는 것을 알 수 있다. 이 방정식은 식 (1)에 해당 성분의 가중 함수를 곱해서 노드 부피에 대하여 평균하여 얻는다. 예를 들면 성분 100의 평균 중성자속에 대한 방정식은

$$\frac{\mathbf{D}}{h_x h_y h_z}\int_{-\frac{h_x}{2}}^{\frac{h_x}{2}}\int_{-\frac{h_y}{2}}^{\frac{h_y}{2}}\int_{-\frac{h_z}{2}}^{\frac{h_z}{2}} w_1(x)w_0(y)w_0(z)\left\{-\nabla^2\hat{\boldsymbol{\phi}}(x,y,z)+\boldsymbol{\Lambda}\hat{\boldsymbol{\phi}}(x,y,z)\right\}dzdydx=\mathbf{0} \tag{24}$$

으로부터

$$\frac{\boldsymbol{\tau}_{x2}}{h_x}\hat{\boldsymbol{\omega}}^x_{100}+\frac{\boldsymbol{\tau}_{y0}}{h_y}\hat{\boldsymbol{\omega}}^y_{100}+\frac{\boldsymbol{\tau}_{z0}}{h_z}\hat{\boldsymbol{\omega}}^z_{100}-\boldsymbol{\Lambda}\bar{\boldsymbol{\phi}}_{100}=\mathbf{0} \tag{25}$$

으로 표시된다. 여기서

$$\boldsymbol{\tau}_{u2}=\frac{\mathbf{D}}{h_u}\frac{\boldsymbol{\Lambda}_u(\boldsymbol{\rho}_{u2}+1)}{2\boldsymbol{\rho}_{u2}}\ , \tag{26}$$

이다. 방향별 성분별 중성자류 방정식과 성분별 평균 중성자속 방정식의 계수들은 한 방향으로의 노드 크기에만 의존하고 전체 성분 모두에 대하여 필요한 계수의 수자도 방향별로 3개 밖에 안 된다. 성분 분해법을 사용하면 계수의 복잡성과 개수가 줄어든다.

8개의 성분별 평균 중성자속 방정식들은 노드를 x축, y축 및 z축으로 분할 할 때 생기는 8개의 8분 육면체에 각각에 대한 중성자 균형 방정식들과 수학적으로 동치라는 것을 알 수 있다. 결국 중성자속 분포를 제한한 제한 조건들은 8분육면체의 각 경계면에서의 중성자속 및 중성자류 연속 조건과 8분 육면체 각각에 대한 중성자 균형 조건들인 것이다.

NODOO는 부분 중성자류을 미지수로하는 반응행렬을 이용한다. 따라서 마지막으로 성분별 방향별 중성자류 방정식과 성분별 중성자 균형 방정식에 있는 성분별 방향별 중성자속과 중성자류는 부분 중성자류로 변환된다.

먼저 x축과 수직인 오른쪽 경계면에서 가중 부분 중성자류를 정의하자.

$$\tilde{\mathbf{j}}^x_{1jk+}=\frac{1}{h_y h_z}\int_{-\frac{h_y}{2}}^{\frac{h_y}{2}}\int_{-\frac{h_z}{2}}^{\frac{h_z}{2}} w_j(y)w_k(z)\left\{+\frac{\mathbf{D}}{2}\frac{\partial}{\partial x}\hat{\boldsymbol{\phi}}(x,y,z)+\frac{1}{4}\hat{\boldsymbol{\phi}}(x,y,z)\right\}dzdy\Bigg|_{x=\frac{h_x}{2}}=+\frac{\tilde{\mathbf{J}}^x_{1jk}}{2}+\frac{\tilde{\hat{\boldsymbol{\phi}}}^x_{1jk}}{4} \tag{27}$$

$$\tilde{\mathbf{j}}^x_{1jk-}=\frac{1}{h_y h_z}\int_{-\frac{h_y}{2}}^{\frac{h_y}{2}}\int_{-\frac{h_z}{2}}^{\frac{h_z}{2}} w_j(y)w_k(z)\left\{-\frac{\mathbf{D}}{2}\frac{\partial}{\partial x}\hat{\boldsymbol{\phi}}(x,y,z)+\frac{1}{4}\hat{\boldsymbol{\phi}}(x,y,z)\right\}dzdy\Bigg|_{x=\frac{h_x}{2}}=-\frac{\tilde{\mathbf{J}}^x_{1jk}}{2}+\frac{\tilde{\hat{\boldsymbol{\phi}}}^x_{1jk}}{4} \tag{28}$$

첨자 중 +와-는 각각 입사와 출사 가중 부분 중성자류을 가리킨다.

다시 성분별 방향별 부분 중성자류를 100성분의 경우에 대하여 정의한다.

$$\tilde{\xi}^x_{100\pm}=\frac{\tilde{\mathbf{j}}^x_{100\pm}-\tilde{\mathbf{j}}^x_{000\pm}}{2}-\left(\frac{1}{2}\mp 2\frac{\mathbf{D}}{h_x}\right)\bar{\boldsymbol{\phi}}_{100} \tag{29}$$

$$\widetilde{\boldsymbol{\xi}}^{y}_{100\pm} = \frac{\widetilde{\mathbf{j}}^{y}_{110\pm} + \widetilde{\mathbf{j}}^{y}_{100\pm}}{2} - \frac{\overline{\boldsymbol{\phi}}_{100}}{4} \tag{30}$$

$$\widetilde{\boldsymbol{\xi}}^{z}_{100\pm} = \frac{\widetilde{\mathbf{j}}^{z}_{101\pm} + \widetilde{\mathbf{j}}^{z}_{100\pm}}{2} - \frac{\overline{\boldsymbol{\phi}}_{100}}{4} \tag{31}$$

그러면 성분별 가중 중성자속과 가중 중성자류는 u=x, y, z에 대해

$$\hat{\boldsymbol{\omega}}^{u}_{100} = 2\left(\widetilde{\boldsymbol{\xi}}^{u}_{100+} + \widetilde{\boldsymbol{\xi}}^{u}_{100-}\right) \tag{32}$$

$$\boldsymbol{\eta}^{u}_{100} = \widetilde{\boldsymbol{\xi}}^{u}_{100+} - \widetilde{\boldsymbol{\xi}}^{u}_{100-} \tag{33}$$

이 된다.

이 두 식을 성분별 가중 중성자류와 가중 중성자속의 관계식 (16), (17)와 (18)에 대입하고 출사 가중 부분 중성자류에 대하여 풀면 결국 우리가 얻고자하는 반응 행렬식을 얻게 된다.

$$\widetilde{\boldsymbol{\xi}}^{x}_{100-} = -\widetilde{\boldsymbol{\xi}}^{x}_{100+} + \varsigma_{x1}\widetilde{\boldsymbol{\xi}}^{x}_{100+} \tag{34}$$

$$\widetilde{\boldsymbol{\xi}}^{y}_{100-} = -\widetilde{\boldsymbol{\xi}}^{y}_{100+} + \varsigma_{y0}\widetilde{\boldsymbol{\xi}}^{y}_{100+} \tag{35}$$

$$\widetilde{\boldsymbol{\xi}}^{z}_{100-} = -\widetilde{\boldsymbol{\xi}}^{z}_{100+} + \varsigma_{z0}\widetilde{\boldsymbol{\xi}}^{z}_{100+} \tag{36}$$

여기서

$$\varsigma_{ui} = 2\left(1 - 2\boldsymbol{\tau}_{ui}\right)^{-1} \quad \mathrm{u = x, y, z,\ i = 0, 1} \tag{37}$$

이다.

또 식 (32) 및 식 (34), (35)와 (36)를 식 (25)에 대입하면 단지 입사 가중 부분 중성자류만 포함하는 가중 평균 중성자 균형 방정식을 얻을 수 있다.

$$\frac{2\varsigma_{x2}}{h_x}\widetilde{\boldsymbol{\xi}}^{x}_{100+} + \frac{2\left(\varsigma_{y0} - 2\right)}{h_y}\widetilde{\boldsymbol{\xi}}^{x}_{100+} + \frac{2\left(\varsigma_{z0} - 2\right)}{h_z}\widetilde{\boldsymbol{\xi}}^{x}_{100+} - \boldsymbol{\Lambda}\overline{\boldsymbol{\phi}}_{100} = 0, \tag{38}$$

여기서

$$\varsigma_{x2} = \frac{\boldsymbol{\tau}_{x2}\varsigma_{x1}}{h_x} \tag{39}$$

이다.

이제 우리가 원하는 방정식들을 모두 얻었다. 참고 도서 [1]에 있는 원래의 방정식들 보다 이 방정식들이 훨씬 간단한 형태인 것을 상기하면 컴퓨터에서 풀 때 노력이 상당히 경감될 것으로 기대할 수 있다.

성분 분해법으로 연관 방정식을 유도하는 과정은 상당히 간단할 뿐만 아니라 성분에 대하여 완전히 체계적(Systematic)이다. 여기서 예를 든 100 성분이 아닌 110 성분에 대한 식을 유도한다고 가정하자. 이 성분의 중성자속 분포는 x 방향

뿐만 아니라 y방향으로도 기함수가 된다. 그러면 110 성분에 대한 식들은 100 성분의 식들로부터 각 변수들의 정의에서부터 최종 결과식까지 y방향 변수와 계수 모두를 x방향과 정확히 대칭이 되도록 바꾸기만 하면 얻을 수 있다. 따라서, 통상적인 미지수들을 성분별 미지수로 변환하는 과정이나, 성분별 연관 방정식들의 연산을 수행하는 과정, 그리고 다시 통상적인 변수로 재변환하는 과정 등을 8개의 성분을 모두 훑는 한 개의 순환 루프로 해결할 수 있는 것이다.

2.2 수치적 특이성 제거와 가속 기법

개량 AFEN도 단면적 행렬 Λ의 고유치가 영이 되면 수치적 특이성(Singularity)를 갖는다. NODOO에서는 이 문제를 해결하기 위하여 참고 도서 [1,4,5]에 있는 Continued Factoring 방법을 그대로 도입하였다.

가속 기법에서 성분 분해법의 이점이 다시 한 번 더 드러난다. 앞서 설명한 대로 성분 분해법의 000 성분의 중성자 균형 방정식은 그 노드 전체에 대한 균형 방정식이다. 이 성분에 대한 방향별 부분 중성자류 방정식과 균형 방정식을 푸는 것이 그대로 이 노드의 전체 32개 미지수의 소격격자 균형 방정식이 되는 것이다. 같은 이유로 100, 010, 001 성분 균형 방정식은 노드를 각각 x축, y축, z축으로 분할한 두 부분에 대한 중성자 균형의 의미가 있다. 따라서 각 성분 균형 방정식을 성분의 기함수 포함 정도에 따라 불균형적으로 풀면 자연스럽게 다중단계 (Multi-Level) 소격격자 가속이 되는 것이다. 성분 분해법 자체의 식을 다단계로 풀면 자연스럽게 가속이 되므로 따로 가속기법의 개발할 필요가 전혀 없다.

또 성분별 방향별 부분 중성자류 방정식과 균형 방정식을 다음과 같이 에너지군 축약만 하면 에너지군 축약 소격 격자 가속을 추가할 수 있다.

$$\bar{\tilde{\xi}}^{x}_{100-} = -\bar{\tilde{\xi}}^{x}_{100+} + \varsigma_{x100}\bar{\tilde{\xi}}^{x}_{100+} \tag{40}$$

여기서

$$\varsigma_{x100} = \frac{\sum_{g\in G} \varsigma_{x1g}\tilde{\xi}^{x}_{100\ g+}}{\bar{\tilde{\xi}}^{x}_{100+}} \tag{41}$$

$$\bar{\tilde{\xi}}^{x}_{100+} = \sum_{g\in G} \tilde{\xi}^{x}_{100\ g+} \tag{42}$$

이다. 물론 축약된 식의 형태는 원래 식의 형태와 똑 같다.

2.3 객체 지향 노달 코드의 개발

노달 코드 NODOO는 순차적 프로그램(Sequential Program)이 아닌 객체 지

향 프로그램이다. 이 프로그램 개발의 중요 목표는 1) 훌륭한 정확도, 2) 유지 보수의 용이성, 3) 사용의 편의성, 4) 용이한 확장성, 5) 효율적 기억 장소 관리 등이다. 이런 목표를 달성하기 위하여 반드시 지켜야할 요건으로는 1) 기능별, 구조별 클래스 정의에 의한 철저한 모듈화, 2) 동적으로만 기억 장소 할당, 3) 물리적 개념적으로 하나만 있는 현상의 중복 정의 금지, 4) 자유로운 구조의 확장, 축소, 저차원 투영 가능, 5) 실제 원자로심의 구성 방식의 구현, 6) 형식에 얽매이지 않는 입력 양식, 6) 자료의 은닉, 상속과 다중성, 연산자 중복 정의 등의 객체 지향 프로그램 고유 개념의 충분한 활용 등이다.

NODOO는 크게 독립된 입출력, 에너지군 구조 처리, 단면적 처리, 노심 구조 처리 모듈로 나누어져 있다. 향후 연소, 제어봉, 재장전, 운전 지원 모듈 등을 추가할 계획이다. 각 모듈간의 연계는 최소화하고 그 마저도 자료 자체가 아니라 연계 함수(Interface Function)로만 이루어진다.

2.3.1 입출력 모듈

입출력 모듈은 C++의 ifstream와 ofstream 클래스들을 공적 상속 (Public Inheritance) 하는 mifstream과 mofstream 클래스로 구성된다. 따라서 기존의 ifstream과 ofstream이 할 수 있는 일은 그대로 할 수 있다. mifstream 클래스는 사용자가 지정하는 입력 파일을 열어서 한 줄을 string으로 읽거나 실수형 및 정수형 변수를 읽는다. 이 때 각 입력 변수 사이는 변수 분리자인 blank, comma, slash, tab으로 구분되어 있기만 하면 된다. 사이의 설명문은 그냥 건너뛰기 때문에 사용자가 입력과 입력 사이에 설명문을 삽입할 수 있다. 이 때 발생하는 rdstate상의 error bit들은 다시 good 상태로 치환되므로 건너뛰는 도중에는 stream 오류가 발생하지 않는다. 따라서 신문과 같이 장문의 text 파일에 수치만 순서대로 나열해도 정상적으로 읽어 낼 수 있다. 물론 정수형이 요구 되는 곳에 실수형을 만나면 원래 ifstream 클래스가 하는 대로 fail bit를 1로 만든다. 16 * 3.14는 3.14란 입력의 16번 반복으로 처리한다. 두 수자 사이에는 tab과 blank가 존재해도 되고 단지 한 개의 asterisk만 있으면 된다. 또 갖은 종류의 일련의 입력을 card라는 입력 단위로 인식한다. card와 card 사이는 ‘card’라는 글자와 따라오는 수자로 구분한다. 예를 들면 ‘card1’이란 string을 보면 이전 카드 입력이 끝나고 card1 입력이 시작되는 것으로 인식한다. 이 때 원래 ifstream 클래스가 eof bit을 1로 만드는 것과 똑 같은 방법으로 eoc bit을 1로 만든다. 따라서 eof를

처리하는 것과 같은 방법으로 eoc를 처리할 수 있다. 현재 NODOO에는 크게 title, 단면적, 노심 입력의 3가지 card가 정의 되어 있다. 어떤 card에서 충분히 입력을 읽지 못하면 그 때까지 읽었던 그 card의 입력을 취소하고 나머지 입력에서 다시 그 card 입력을 찾으려 한다. 반대로 충분한 입력을 읽었는데도 card가 아직 끝나지 않았다면 나머지 입력은 eoc를 만날 때까지 건너뛴다. 이 원칙은 n * m형 반복 입력을 읽을 때도 적용이 되어서 eoc를 만나는 순간 남은 반복 회수는 소진된다. 이 기능을 이용하면 정상적인 입력 자료 아래에다 시험용 입력 등을 붙여 놓고 쓰는 것이 가능하다. mifstream 클래스는은 echo라는 ofstream 클래스를 사적(Private) 멤버로 가지고 있어서 유효한 입력을 읽는 순간 사용자가 제공한 파일 이름의 확장자만 .echo로 바꾼 메아리 파일에 읽은 입력을 자동으로 출력한다. 따라서 이 파일을 이용하면 신문 같이 잡다한 입력을 콤팩트한 입력으로 바꿀 수도 있고 어떤 입력을 제대로 인식했는지 여부도 확인할 수도 있다. 출력 양식은 사용자의 편의를 생각해서 다른 노달 코드들이 출력하는 양식과 유사하게 하였다. 또 다른 모듈에 정의된 대부분의 클래스는 저 자신만의 입출력 연산자 <<와 >>를 중복 정의하고 있어서 이를 사용하여 간단하게 그 클래스의 중요 정보를 입출력할 수 있다. 예를 들면 집합체 클래스 객체인 a에 대해 cout << a를 사용하면 그 집합체의 중요 정보 정보를 출력 할 수 있다.

2.3.2 에너지군 처리 모듈

이 모듈에는 에너지군 차원의 벡터와 대각 행렬를 포함한 행렬을 처리한다. 에너지 군에 대한 반복 루프는 단지 이 모듈에만 나타나고 다른 모듈에는 전혀 나타나지 않는다. 다른 모듈에서 보이지는 않지만 이 모듈에 있는 클래스들은 모두 동적 메모리 할당을 하는 일차원 배열로만 되어 있다. 하지만 행렬의 경우 a[i][j]와 같이 2차원 배열 Access 방법을 그대로 사용할 수 있다. 실제 1차원 배열을 최대 4차원 즉 a[i][j][k][l] 형태로까지 접근할 수 있다. 이 모듈에는 모든 벡터와 행렬의 사칙 연산자가 정의 되어 있어서 a, b, c가 실수 형이든 벡터 형이든 행렬 형이든 상관 않고 a += 1.0 + b * c와 같은 연산을 수행할 수 있다. 또 역행렬이나 고유치, 고유 벡터를 구할 수도 있다. 복소수 행렬은 한 열에 그 복소수의 실수부를 다른 한 열에는 허수부를 저장하고 그에 맞는 사칙 연산을 정상적으로 수행할 수 있다. 따라서 다군 AFEN에서와 같이 고유치가 복소수가 되는 경우의 행렬 연산도 아무런 추가적인 계산 시간 없이 수행 할 수 있다.

이 모듈의 다른 역할은 다른 모듈을 포함한 전체 코드의 동적 메모리 할당을 수행하는 것이다. 예를 들면 다른 모듈에 있는 집합체 클래스는 연료봉 클래스들에 대한 이 모듈의 벡터 클래스를 공적 상속하여 정의된다. 이는 이 모듈의 행렬과 벡터 클래스가 어떤 형이든 상관 않는 Template 클래스로 정의되어 있기 때문이다. 따라서 어떤 형태의 다차원 배열도 원하는 크기대로 만들 수 있는 것이다. 원래 행렬, 벡터에 대하여 정의되어 있는 사칙 연산자를 그대로 이용하여 집합체 객체의 평균을 봉 객체 합을 봉수로 단순히 나누어서 계산하는 것 등이 가능하다. 심지어 노심 객체의 벡터를 정의하면 노심끼리 더하고 빼는 일도 가능할 것이다. 노심의 저차원 투영은 주로 이 기능을 가지고 수행된다.

향후 이 모듈은 연소 모듈의 기본 클래스로도 사용될 것이다. 즉 연소를 핵종 차원의 벡터, 행렬간의 연산으로 정의하면 이 모듈의 기능을 그대로 활용할 수 있기 때문이다.

이 모듈에 정의된 대부분의 벡터와 행렬 클래스는 입출력 연산자 <<와 >>를 정의하고 있어서 그 자체를 한꺼번에 쉽게 입출력 할 수 있다.

2.3.3 단면적 처리 모듈

단면적 처리 모듈에는 단면적과 관련된 클래스들이 정의되어 있다. 단일 단면적 클래스는 에너지군 벡터와 행렬 클래스의 객체들로 정의된 확산 계수 등 균질화 상수들을 사적 멤버로 하여 구성된다. 단면적 표(Table) 클래스는 이 단면적 클래스의 벡터를 공적 상속하여 정의된다. 따라서 단면적 간의 사칙 연산이 가능하다. 또 단면적을 이용하여 개량 AFEN의 반복 계산 계수를 구하는 일도 이 모듈에서 한다. 각 클래스 모두 입출력 연산자가 정의되어 있다. 즉 단면적 표 클래스 객체인 t를 cin >> t로 간단히 읽을 수 있다. 이 명령문이 수행되는 경로를 보자. 먼저 단면적 표 클래스의 >> 연산자 함수를 수행하면 각 표내의 벡터의 요소 즉 단면적 객체에 대해 단면적 클래스의 >> 연산자가 수행되며 최종적으로 이 단면적 클래스의 멤버인 확산 계수 등과 같은 에너지군 벡터 혹은 행렬 클래스의 >> 가 수행되는 것이다.

2.3.4 노심 구조 모듈

노심 구조 모듈의 기본은 핵종 벡터 클래스와 중성자속 벡터 클래스 등을 사적 멤버로 하는 연소점 클래스이다. 이 연소점 클래스들이 모아서 노심 클래스까

지를 구성하는 원리는 실제 노심 구성 원리와 같다. 먼저 연료봉 클래스는 이 연소 점들의 축상 벡터를 공적 상속하여 정의되고 다시 집합체 클래스는 연료봉 클래스들의 벡터를 공적 상속하여 정의된다. 개량형 AFEN에서 집합체는 각 사분면에 있는 4개의 연료봉만으로 집합체가 구성된다. 노심 클래스에는 이 집합체 객체를 지시할 수 있는 포인터 변수들이 채워져 있다가 사용자가 지시하는 구조대로 집합체 클래스의 객체들을 동적 할당하여 노심을 구성한다. 축상 벡터 배열 크기를 1로, 혹은 연료봉 개수를 1로 고정함으로써 따로 코딩하는 수고 없이 같은 소스 리스트로 축방향 1차원 혹은 반경 방향 2차원 계산을 할 수 있다. 취급할 수 있는 구조는 8분 노심, 사분 노심, 이분 노심 등 다양하고 전반사 대칭 조건, 회전 대칭 조건 등도 취급할 수 있다. 노심을 45도와 135도 각도로 잘라서 대칭 조건을 주는 것도 가능하다. 각각 다른 대칭 구조로의 변경도 단 몇 줄의 코딩으로 구현 하였다. 구조 변경 도중 메모리 폭주가 일어나지 않게 불필요한 중간 메모리를 잡지 않는다. 각 노심 객체는 사적 멤버로 연료봉 객체 하나를, 각 연료봉 객체는 바닥 연소점 객체 하나를 추가로 가지고 있어서 이들과 벡터 행렬 클래스에서 정의된 사칙 연산자를 이용하여 노심의 축방향 일차원 투영이나 반경 방향 2차원 투영을 수행한다. 투영된 노심을 마치 독자적인 노심처럼 1차원 계산이나 2차원 노달 계산이 가능하다.

노심의 일정 부분을 Access하는 방법으로 노드라는 클래스가 구현된다. 이 클래스에서는 앞서 설명한 성분 분해법으로 유도한 개량 AFEN 방정식들을 푼다. 노심 외곽에서 부분 입력 중성자류를 가져 올 때는 출력 중성자류에 알베도 값이 곱해져서 가져온다. 이 같은 작업이 불과 몇 줄의 단 한 번의 코딩으로 이루어져 있어서 유지 보수 노력이 거의 들지 않는다. 향후 이 클래스에서 연소도 할 작정이다. 연소를 8분 노드 각각에 대하여 수행하고 노드내의 단면적 분포를 고려할 수 있게 함으로서 축방향 노드 크기를 100cm 이상 키워도 정확도를 유지하게 할 것이다.

3. 계산 결과 및 결론

본 연구에서 개발한 NODOO 코드로 EPRI-9[6]의 집합체를 축방향으로 360cm로 확장하고 양쪽에 20cm 크기의 반사체를 붙인 3차원 노심의 노달 계산을 하였다. 노드 크기를 반경 방향으로는 집합체 전체 크기로, 축방향으로는 30cm로 하고 에너지군을 축약하는 소격격자 가속을 적용하지 않았을 때 전체 노심의 노달

계산에 대략 1분 정도가 소요된다. 이는 객체 지향 NODOO가 다른 노달 코드에 비해 매우 느린 것을 의미하지만 이는 당연하고 예상한 일이다. 앞으로 에너지군 축약 가속을 추가하면 한 주기 연소 계산을 다해도 대략 10분 정도 걸린다고 보고 C++가 갖는 다양한 그래픽 기능을 추가하여 사용자의 편의를 도모하면 충분히 실제 노심 설계에 사용할 수 있을 것으로 판단한다. 앞으로 NODOO에는 연소, 제어봉, 재장전, 운전 지원 모듈 등을 추가할 계획이다.

본 연구에서 개발한 성분 분해법을 NEM과 ANM과 같은 다른 노달 방법에도 적용하면 그 노달 계산의 효율을 증대할 수 있다. 특히 유도된 연관 방정식 자체가 다중 소격격자 가속 방정식 역할도 함으로서 별도의 가속 기법의 개발이 필요 없게 된다.

감사의 글

본 연구는 과학기술부가 지원하는 원자력연구개발 중장기과제 중 “미래형핵연료 노심분석기술개발” 과제의 일환으로 수행하였다.

참고 문헌

[1] S. W. Woo, N. Z. Cho, and J. M. Noh, "The Analytic Function Expansion Nodal Method Refined with Transverse Gradient Basis Functions and Interface Flux Moments," Nucl. Sci. Eng., 139, 156 (2001)

[2] K. S. SMITH, "An Analytic Nodal Method for Solving the Two-Group Multidimensional, Static and Transient Neutron Diffusion Equation," Nuclear Engineering Thesis, Massachusetts Institute of Technology (1979).

[3] J. M. NOH, et al., "A General Approach to Multigroup Extension of the Analytic Function Expansion Nodal Method Based on Matrix Function Theory," Proc. 1996 Joint Intl. Conf. Mathematical Methods and Super Computing for Nuclear Applications, Saratoga Springs, New York, Octorber 6-10, 1997, Vol. 1, p. 144, American Nuclear Society (1997).

[4] N. Z. Cho, C. J. Park, and J. M. Noh, "Removal of Numerical Singularity in the Transverse-Integrated Analytic Nodal Method via Continued Factoring," Trans. Am. Nucl. Soc., 86, 361 (2002). .

[5] S. W. Woo, N. Z. Cho, and J. M. Noh, "Removal of Numerical Singularity in Analytical Nodal Methods via Continued Factoring," Trans. Am. Nucl. Soc., 84, 205 (2001).

[6] H. S. Khalil, “The Applications of Nodal Methods to PWR Analysis,” PhD Thesis, Massachusetts Institute of Technology (1983).