'2001 추계학술발표회 논문집
한국원자력학회

최소자승해에 기초한 노내 출력 감시 프로그램에서 Pre-conditioned CGNR 반복계산법의 Preconditioner 탐색

Investigation of Preconditioners for the CGNR Iterative Method in Core Power Monitoring Program based on Least Square Solution

이기복, 주한규, 지성균, 장문희

한국원자력연구소
대전광역시 유성구 덕진동 150

김창효

서울대학교 원자핵공학과
서울특별시 관악구 신림동 산 56-1

요약

이 논문에서는 중성자 균형 방정식 체계와 계측기 신호 방정식 체계의 최소자승해를 구하는 최소자승 노심 출력 감시 프로그램의 반복계산법으로 사용된 Pre-conditioned CGNR 해법에서 정확성을 유지하면서 전산시간을 줄일 수 있는 유효 적절한 Preconditioner 를 다룬다. Pre-conditioned CGNR 반복계산법의 수렴성은 Preconditioner 에 따라 크게 달라지는데, 이 논문에서는 최소자승해를 구하는 정규식의 대각 요소만 취하는 DIAG preconditioner, IMGS QR factorization 에 기초한 IMGS preconditioner, 중성자 균형 방정식의 해법에서 사용된 BILU3D preconditioner 에 대해 전산 시간과 계산 정확성을 시험한다. 시험 결과는 DIAG preconditioner 가 가장 효율성이 뛰어나고, Preconditioner 가 없을 때보다 계산시간이 약 반으로 줄어듦을 보인다.

Abstract

Preconditioners are investigated to determine the best preconditioner that saves much computing time in the preconditioned CGNR calculation for the normal equation composed of the neutron balance equation and the detector signal equation. Since the convergence of preconditioned CGNR depends on the effectiveness of the preconditioner, several different preconditioners such as the DIAG preconditioner composed of diagonal element of normal equation, IMGS preconditioner based on IMGS QR factorization and BILU3D preconditioner used in neutron balance equation solver are examined. The computing time and solutions were compared. The DIAG preconditioner shows the best performance to reduce the computing time by about 50%.

1. 서론

현재 상용화된 노심 출력 감시 프로그램인 CECOR, INCORE, CARIN [1],[2],[3]등은 노심 거동에 대한 예측 능력이 없어, 최근 노심 출력 감시 프로그램은 운전중인 노심의 출력뿐만 아니라, 제어봉의 삽입 인출 등 운전행위와 지논 과도상태에 의해 유발되는 노심의 거동을 예측할 수 있는 기능을 갖출 수 있도록 개발되고 있다[4],[5]. 이러한 기능을 보유하기 위해서는 3 차원 중성자속에 대한 정보를 아는 것이 필수적이므로 노심 추적 계산에 의한 중성자 균형 방정식(NBE)과 계측기 신호 방정식 (DSE)를 결합하여 노심 출력 감시 프로그램을 구성하여야 한다. 최근 이러한 기능을 갖춘 노심 출력 감시 방법의 하나로 노심 중성자속의 비선형 거시격자 유한차분법에 기초한 중성자 균형 방정식(NBE) 체계와 계측기 신호(DSE) 방정식 체계의 최소자승해를 구하는 방법이 소개되었다[6].

최소자승법은 정규식(Normal Equation)을 형성하고, 이 정규식의 해를 구하는 방법으로 Preconditioned CGNR(Conjugate Gradient, N for Normal and R for Residual) 반복계산법이 효과적이라고 알려져 있다[7]. 이 방법은 Preconditioner 에 의해 CGNR 방법의 수렴성을 향상시켜 계산 시간을 줄일 수 있다. 그러나 Preconditioned CGNR 법의 수렴성은 Preconditioner 의 성능에 의해 크게 좌우되기 때문에 Preconditioner 는 원 정규식에 대한 근사이면서 해를 쉽게 구할 수 있도록 구성되어야 한다. 본 논문에서는 최소자승법에 의한 노심 감시 프로그램에 적용된 Preconditioned CGNR 방법에서 효율적인 Preconditioner 를 탐색한다.

2 장에서는 최소자승해를 구하는 정규식의 구성에 대해 알아보고 3 장에서는 Preconditioned CGNR 법을 소개하고 4 장에서는 적용된 Preconditioner 와 그 해법에 대해 서술하고 4 장에서 계산 결과와 결론을 제시한다.

2. 최소자승 해를 구하는 정규식

노심 내 3차원 출력 분포를 구하는 전체 체계는 중성자 균형 방정식 체계와 계측기 신호 방정식 체계를 합하여 이루어진다. 중성자 균형 방정식 체계는 격자계산 코드나 정밀3차원계산 코드로부터 생산된 거시 단면적자료를 사용하여 노심내의 중성자속 분포를 계산하는 체계이다. 이 체계로부터 계산에 의해 예측된 중성자속을 알 수 있다.

$$\mathbf{M}f = \frac{1}{k_{eff}}\mathbf{F}f \tag{1}$$

계측기 신호 방정식 체계는 계측기로부터 측정신호를 받아 적절한 처리를 하여 계측기가 위치한 지점의 중성자속을 알아내는 체계로 측정신호와 측정중성자속과의 관계를 형성하는 체계이다.

$$\mathbf{D}f = \mathbf{s} \tag{2}$$

두 체계를 합한 전체 체계는 수식이 결정해야 할 미지수보다 더 많게 되어 소위 over-determined system 이 된다.

$$\mathbf{A}f = b \tag{3}$$

여기서,

$$\mathbf{A} = \begin{bmatrix} \mathbf{M} - \dfrac{1}{keff}\mathbf{F} \\ \mathbf{D} \end{bmatrix}, \quad b = \begin{bmatrix} q \\ s \end{bmatrix} \tag{4}$$

이 두 체계의 오차가 최소가 되게 하는 해는 유일하게 되며 그 방정식은 다음과 같은 선형 체계로 전환되어 잔여치를 최소가 되게 하는 식으로 전환되는데, 이것을 정규식(Normal Equations)이라 한다.

$$\text{minimize} \quad \|b\text{-}\mathbf{A}f\|_2 \quad \Leftrightarrow \quad \text{minimize residual}(=b\text{-}\mathbf{A}f\,) \tag{5}$$

$$\mathbf{A}^{\mathrm{T}}\mathbf{A}f = \mathbf{A}^{\mathrm{T}}b \quad \Leftrightarrow \quad \mathbf{R}f = c \tag{6}$$

여기서 $\mathbf{A}^{\mathrm{T}}\mathbf{A}$ 행렬은 대칭 정치정부호행렬(Symmetric Positive Definite Matrix)가 된다. 전체 계산 노드 수를 N 이라 하고 중성자군 수를 G 라 하고, 전체 계측기 수를 L 이라 하면 $\mathbf{A}$ 행렬의 차원은 [(NG+L) x NG] 가 되고 $\mathbf{A}^{\mathrm{T}}\mathbf{A}$ (=$\mathbf{R}$)행렬은 [NG x NG]가 된다. 행렬의 크기가 클 때 $\mathbf{R}$ 행렬을 구한 다음 $\mathbf{R}f = c$ 선형 체계의 해를 직접 구하는 방법은 연산의 수가 방대하여 기억용량이나 전산 시간의 소모가 크다. 따라서 $\mathbf{A}f = b$ 선형체계에서 반복 계산법을 통하여 식(6)을 만족하는 해를 구하는 것이 바람직하다. 본 논문에서는 Preconditioned

CGNR 반복계산법을 적용하고 효율적인 Preconditioner 를 탐색하였다.

3. Preconditioned CGNR

정규식의 계산 시간을 줄이기 위해 CGNR 방법의 수렴성을 향상 시키는 반복계산법으로 preconditioned CGNR 방법이 소개되어 있다. 이 방법에서 preconditioned residual z_i 는 $z_i = \mathbf{M}^{-1} y_i$ 로 계산된다. 스칼라 값 a_i 는 다음과 같이 주어진다.

$$a_i = (z_i, y_i) / (\mathbf{A}^{\mathbf{T}} \mathbf{A} p_i, p_i) = (z_i, y_i) / (\mathbf{A} p_i, \mathbf{A} p_i) \quad (7)$$

이때 벡터 $w_i = \mathbf{A} p_i$ 를 도입하면 다음과 같이 Algorithm 을 전개할 수 있다.

Algorithm : Preconditioned CGNR

1. Compute $r_0 = b - \mathbf{A} x_0$, $y_0 = \mathbf{A}^{\mathbf{T}} r_0$, $z_0 = \mathbf{M}^{-1} y_0$, $p_0 = z_0$
2. For i =0,…, until convergence Do:
3. $w_i = \mathbf{A} p_i$
4. $a_i = (z_i, y_i) / \|w_i\|_2^2$
5. $x_{i+1} := x_i + a_i p_i$
6. $r_{i+1} = r_i - a_i w_i$
7. $y_{i+1} = \mathbf{A}^{\mathbf{T}} r_{i+1}$
8. $z_{i+1} = \mathbf{M}^{-1} y_{i+1}$
9. $b_i = (z_{i+1}, y_{i+1}) / (z_i, y_i)$
10. $p_{i+1} := z_{i+1} + b_i p_i$
11. EndDo

여기서 preconditioner 행렬 **M** 은 $\mathbf{A}^{\mathbf{T}}\mathbf{A}$ 행렬의 근사로, group 수를 G 노드 수를 N 계측기 신호 수를 D 라 할 때 차원이 [(GN+D)x(GN+D)]인 Symmetry Positive Definite 행렬이 된다. Preconditioned CGNR 방법의 수렴성은 preconditioner 행렬 **M** 을 어떻게 정하느냐에 따라 결정된다. 여기서는 $\mathbf{A}^{\mathbf{T}}\mathbf{A}$ 행렬의 대각요소만 취하는 방식(DIAG preconditioner), Incomplete Modified Gram Shmidt factorization 에 의한 방식(IMGS preconditioner)과 NBE 체계에서 사용한 Blockwise Incomplete LU factorization 에 의한 방식(BILU preconditioner)을 시험하여 그 중 가장 좋은 preconditioner 를 선택한다.

4. Preconditioner

DIAG preconditioner

$\mathbf{A}^{\mathrm{T}}\mathbf{A}$ 행렬의 대각요소만 취하는 방식의 DIAG preconditioner 는 그 대각요소가 다음과 같이 간단하게 전개된다. 즉 NBE system 과 DSE system 으로 이루어진 **A** 행렬의 column vector 의 제곱이 preconditioner 행렬 **M** 의 대각요소 d_i 를 구성하게 되어, preconditioned residual z_i 는 $z_i = y_i / d_i$ 로 직접 계산할 수 있다.

IMGS preconditioner

IMGS preconditioner 는 $\mathbf{A}^{\mathrm{T}}\mathbf{A}$ 행렬을 직접 계산하지 않고, **A** 행렬을 drop set **P** 를 이용하여 upper triangular matrix **R** 과 unitary matrix **Q** 로 분해하여 $\mathbf{M}(\approx \mathbf{A}^{\mathrm{T}}\mathbf{A})$ 행렬의 요소를 구하는 방법이다[7]. 이때 **Q** 행렬은 다음과 같은 성질이 있다.

$$\mathbf{Q}^{\mathrm{T}}\mathbf{Q} = \mathbf{I} \tag{8}$$

따라서,

$$\mathbf{A}^{\mathrm{T}}\mathbf{A} = \mathbf{R}^{\mathrm{T}}\mathbf{Q}^{\mathrm{T}} \cdot \mathbf{Q}\mathbf{R} = \mathbf{R}^{\mathrm{T}}\mathbf{R} \tag{9}$$

여기서 행렬의 차원을 살펴보면 **A** 의 차원은[**m**x**n**], **m**>**n** 일 때 **Q** 는 [**m**x**n**]이고 **R** 은 [**n**x**n**]이 된다. Drop set **P** 는 구하고자 하는 incomplete **R** 행렬의 요소에서 제외하게 되는 drop position 의 집합으로 $P_n = \{(i, j) | 1 \le i < j \le n\}$ 이다.

Drop set **P** 로 **A** 행렬을 **QR** factorization 하는 Algorithm 은 다음과 같다.

Algorithm [Q, R] = IMGS [A, P]

For k=1, 2, 3, …, n, (column vector index)

(1) $r_{kk} = \|a_k\|_2$

(2) $q_k = a_k / r_{kk}$

for j=k+1, k+2, …,n

(3) $r_{kj} = \begin{cases} 0 & (k, j) \in \mathbf{P} \\ q_k^T a_j & (k, j) \notin \mathbf{P} \end{cases}$

(4) $a_j = a_j - q_k r_{kj}$

endfor

endfor

Drop set **P** 를 정하는 방법에는 factorization 을 수행하면서 요소값이 정해진 drop tolerance e (대개 최고값의 2%)보다 작으면 배제하는 dynamic method 와 미리 계산 수행전에 기하학적으로 drop set 을 정하는 static method 가 있다. 여기서는 모든 요소를 계산하지 않고 drop set 을 기하학적 구성에 따라 미리 정하는 방식을 선택하였다. 즉 가장 coupling 이 큰 인접한 노드에 대해서만 요소를 계산하여 **R** 행렬의 요소를 구성한다. 이러한 방법에 따라 그림 1 의 **A** 행렬에서 east, south, top nodes 의 column vector 만 drop set 에서 제외되어, **QR** factorization 의 결과로 upper diagonal 행렬 **R** 은 4 개의 diagonal elements(own, east, south, top)로 구성된다.

Preconditioner $\mathbf{M}(\approx \mathbf{A}^T\mathbf{A})$ 행렬 계산을 위해 실제 필요한 행렬은 **R** 행렬이므로, **R** 행렬의 요소가 모두 결정되면 Preconditioned CGNR algorithm 에서 preconditioned residual vector z 를 다음과 같이 차례로 forward substitution 과 backward substitution 을 수행하여 구할 수 있다.

<< Solving Strategy for z : $z = \mathbf{M}^{-1} y$ **>>**

1. y = **M***z =(**R'R**)*z = **R'***(**R***z) , let v=**R***z
2. y= **R'**v --> v = inv(**R'**)*y

   solve v by forward substitution
3. v = **R***z --> z = inv**R** * v

   solve z by backward substitution

<u>BILU preconditioner</u>

BILU preconditioner 는 3 차원 CMFD equation system 을 축방향 plane 별로 순서를 정한 다음, radial coupling 으로 이루어진 submatrix **A** 와 axial coupling 으로 이루어진 lower submatrix **L**, upper submatrix **U** 에 대해 incomplete **LU** factorization 을 통해 차원에 따른 3 개의 단계별로 해를 구해나간다[9]. 이 것은 NBE system 자체에 대한 preconditioner 로서, **A** 행렬이나 $\mathbf{A}^T\mathbf{A}$ 행렬의 근사는 아니다. 따라서 DSE system 의 요소가 개입되어 있지 않으므로 좋은 preconditioner 는 아니나 prediction mode 에서 이미 구해놓았으므로 따로 행렬 요소를 결정하지 않아도 된다는 장점이 있다.

5. 결과 및 결론

표 1은 영광 3호기 1주기를 대상으로 연소도별로 36 n/FA 계산 결과를 참값으로 간주하고, 이로부터 가상 출력신호를 생산하여 1 n/FA과 4 n/FA 계산을 통해 최소자승법을 적용하였을 때, 각각의 preconditioner에 따른 오차와 계산 시간을 비교한 것이다.

표 1에서 보는 바와 같이 IMGS preconditioner는 preconditioner를 형성하는 데 소요되는 시간과 변수가 많아 수렴성을 향상시키지 못하고 있으며, BILU preconditioner는 $\mathbf{A}^T\mathbf{A}$ 행렬의 근사가 아니기 때문에 수렴되지 않은 결과를 보이는 반면 DIAG preconditioner는 간단한 근사로 결과는 동일하면서 수렴성을 크게 향상시켜 계산속도를 약 50%로 줄이는 결과를 보이고 있다. 따라서 최소자승해에 기반을 둔 노심 출력 감시 프로그램의 preconditioned CGNR 반복계산법에 DIAG preconditioner를 사용한다.

감사의 글

본 연구는 과학기술부의 원자력연구개발사업의 일환으로 수행되었다.

References

[1] Final Safety Analysis Report for YGN Unit 3 & 4, Korea Electric Power Company.

[2] Final Safety Analysis Report for YGN Unit 1 & 2, Korea Electric Power Company.

[3] A. J. Harris and K. A. Jones, “INCORE3 Program,” WCAP-8492, March 1975.

[4] B. O. Cho et al., “Core On-line Monitoring and Protection Aid Surveillance System COMPASS for SMART,” KAERI/TR-1149/98, 1998.

[5] L. R. Pogosbekyan and D. A. Lysov, "Quasi Solution of Radiation Transport Equation," *Nucl. Sci. Eng*., **121**, 345-351 (1995).

[6] 이기복외, “계측기 신호를 이용하여 노내 감시 출력분포의 최소자승 해를 구하기 위한 효율적인 CGNR 해법의 적용,” 한국원자력학회 2000년 춘계학술발표회 논문집.

[7] Yousef Saad, *Iterative Methods for Sparse Linear Systems*, PWS Publishing Company ,p.318,(1996).

[8] X. Wang, K. A. Gallivan, and R. Bramley, "CIMGS: AN INCOMPLETE ORTHOGONAL FACTORIZATION PRECONDITIONER," SIAM J. SCI. COMPUT. VOL. 18, NO. 2, PP. 516-536, MARCH 1997.

[9] H. G. Joo, G. Jiang, and T. J. Downar, “Stabilization Techniques for the Nonlinear Analytic Nodal

Method," *Nucl. Sci. Eng.*, **130**, 47 (1998).

표 1. Preconditioner에 따른 Preconditioned CGNR 반복계산법의 오차 및 계산 시간 비교

* **LS Method(1) : 1 node/FA 계산, PC-733MHz WIN2000-OS Machine**

BU (MWD/KgU)	Power (%)	Predicted	CGNR	Preconditioned CGNR		
				BILU	IMGS	DIAG
Avg. Error(%)						
0.000	20.0	1.05	0.19	0.68	0.19	**0.19**
1.423	80.0	1.28	0.15	0.81	0.15	**0.15**
2.357	100.0	1.41	0.15	0.90	0.15	**0.15**
2.880	100.0	1.52	0.16	0.96	0.16	**0.16**
3.703	100.0	1.52	0.18	0.98	0.18	**0.18**
5.030	100.0	1.35	0.18	0.88	0.18	**0.18**
5.804	100.0	1.30	0.17	0.84	0.17	**0.17**
6.601	100.0	1.23	0.16	0.79	0.16	**0.16**
8.082	100.0	1.16	0.15	0.75	0.15	**0.15**
8.844	100.0	1.14	0.15	0.74	0.15	**0.15**
13.418	100.0	1.11	0.14	0.72	0.14	**0.14**
CPU Time(sec.)						
0.000	20.0		3.93	Not converged	4.20	**1.75**
1.423	80.0		4.07		4.27	**1.55**
2.357	100.0		4.22		4.62	**2.01**
2.880	100.0		4.86		4.36	**1.95**
3.703	100.0		4.67		4.46	**1.79**
5.030	100.0		4.59		4.40	**1.70**
5.804	100.0		4.56		4.39	**1.70**
6.601	100.0		4.21		4.24	**1.63**
8.082	100.0		4.51		4.04	**1.44**
8.844	100.0		3.67		4.40	**1.57**
13.418	100.0		3.95		4.07	**1.45**

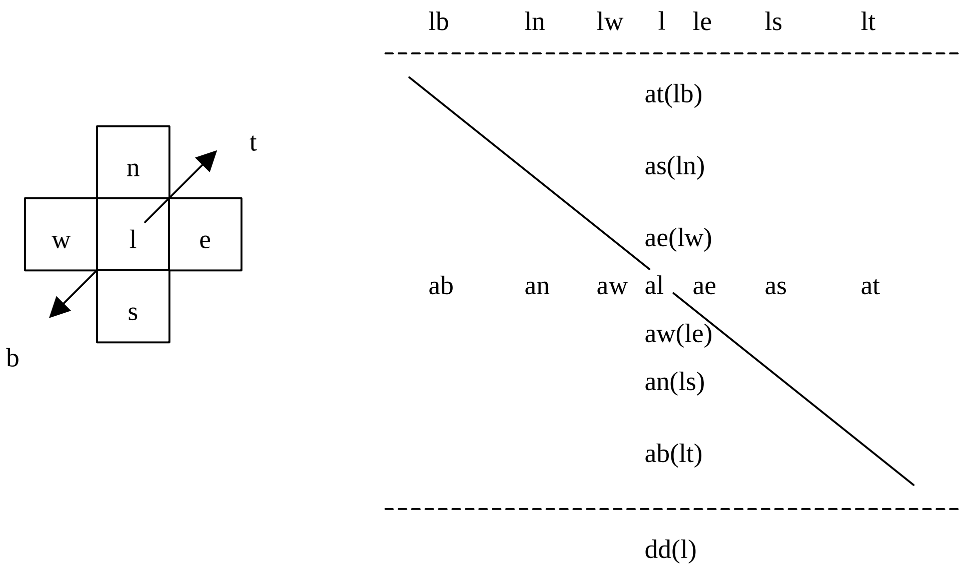

그림 1. Node Index and Elements of Matrix **A**