

< 논문 >

영역분할 기법에 기초한 축소시스템 구축에 관한 연구

김현기[†] · 조맹효* · 김 혁** · 최형길** · 최재락**
(2006년 1월 16일 접수, 2006년 5월 31일 심사완료)

Study on the Reduced System Based on the Sub-Domain Method

Hyungi Kim, Maenghyo Cho, Hyuk Kim, Hyoung-Gil Choi and Jae-Rack Choi

Key Words : Two-Level Condensation Scheme(2 단계 축소기법), Sub-Domain Method(영역분할 기법), Reduced System(축소시스템), Sequential Elimination Scheme(순차적 소거법)

Abstract

Reduction schemes approximate the lower eigenvalues that represent the global behavior of the structures. But, they are not efficient to be applied to large-scaled problems because these schemes require considerable amount of computing time in constructing reduced one from the original large-scaled systems. In addition, the selection of the primary degrees of freedom might be localized to cause the excessive emphasis of the lower mode or lost of the important modes. In the present study, a new reduction method combined with the sub-domain method is proposed. For the construction of the final reduced system, the system of each domain subdivided into primary, slave and interface degrees of freedom. It is remarkably efficient and accurate comparable to full-scale system. Numerical examples demonstrate that the proposed method saves computational cost effectively and provides a reduced system which predicts accurate eigen-pairs of global system.

1. 서 론

최근의 컴퓨터 시스템의 비약적인 발전으로 구조물의 동적 해석은 과거에 비해 보다 빠르게 수행될 수 있다. 최근, 병렬 컴퓨터나 슈퍼 컴퓨터를 이용하여 대형 구조물 해석이 수행되고 있으며, 효율적인 알고리즘의 개발로 그 성능은 날로 향상되고 있다. 그러나, 일반인들이 이러한 시스템에 접근 또는 사용하는 것은 용이하지 않다. 또한, 현재 구조물의 규모가 대형화되고 형상이 복잡해지며, 다양한 기능을 충족시켜야 하는 등 해석에 있어서 고려해야 할 조건과 설계를 위한 변수 등이 증가되고 있는 추세다. 이런 요구들은 전산 시스템의 발전에도 불구하고, 최적화 문제 또는 동적 해석 등에서 충분한 전산자원과 많은 연산시간을 요구하기 때문에

많은 제약의 원인이 되고 있다. 이런 문제 때문에 구조물의 신뢰성을 유지하면서, 시간과 전산자원 등의 효율성 문제를 개선하고자 하는 연구들이 과거부터 많이 진행되어 왔다. 이런 기법 중에서 가장 널리 제안 되어온 것이 축소모델을 구성하는 방법이다.⁽¹⁻³⁾

본 연구에서는 대형구조물의 효율적인 해석을 수행하기 위하여 기존의 기법들의 문제점을 개선하고, 신뢰성을 확보할 수 있는 축소 기법을 제안한다. 축소기법은 주자유도를 기초로 시스템을 구축하기 때문에, 구조물에서 중요 자유도를 적절히 선택하는 것이 매우 중요하다. 본 연구에서는 효율적인 주자유도 선정 기법을 제안한다. 그러나, 축소 기법은 최종 시스템을 구축하기 위해서 많은 전산 자원과 계산 시간을 필요로 한다. 따라서, 전체 시스템에 축소 기법이 바로 적용된다면, 엄청난 전산 자원이 요구되는 것은 자명하다. 본 연구에서는 축소기법과 영역분할 기법과의 연동을 통해 효율성을 확보하면서 신뢰성도 유지할 수 있는 방법을 제안한다.

[†] 회원, 서울대학교 대학원 기계항공공학부

* 회원, 서울대학교 기계항공공학부

E-mail : mhcho@snu.ac.kr

TEL : (02)880-1693

** 삼성전자 메카트로닉스 연구소

2. 축소시스템 구축 기법

본 연구에서는 기존의 축소시스템 기법들이 갖고 있는 문제점을 개선하여 보다 효율적이고 신뢰도가 높은 축소시스템 구성 방법인 2 단계 축소기법을 사용한다. 이 방법은 두 가지 단계로 구성된다. 첫 번째 단계에서는 요소단위의 에너지 평가에 의해 주자유도 선정을 위한 후보 영역을 결정하고, 두 번째 단계에서는 순차적 소거법을 적용하여 최종적인 축소시스템을 구성하게 된다.

일반적으로 순차적 소거법은 축소시스템을 구성하기에는 신뢰성이 높지만, 매우 비효율적인 방법이다.⁽⁶⁾ 그러나, 2 단계 축소기법에서는 순차적 소거법이 제 1 단계 축소과정을 거친 축소시스템에 적용되므로, 많은 시간을 요구하지 않으면서, 순차적 소거법이 갖고 있는 신뢰성을 확보할 수 있다. Figure 1은 2단계 축소 기법의 Schematic 이다. Figure 1에서 (a)는 해석 구조물이고, (b)는 요소별 에너지 평가에 의해 선정된 주자유도 후보 영역이다. (c)는 순차적 소거법에 의해 최종적으로 선정된 주자유도를 나타낸다. 그림에서 표시된 위치는 주자유도가 포함된 절점을 의미한다.

2.1 제 1 단계 축소 과정 : 후보 영역 선정

2.1.1 리츠(Ritz) 벡터 구성

후보영역 선정은 리츠벡터를 이용한 요소단위의 레일리(Rayleigh) 에너지 지수 평가를 통해 이루어진다. 리츠 벡터는 정적 해석을 통해 얻어지는 근사화된 모드이므로, 많은 계산시간을 필요로 하지 않는다. 리츠 벡터 계산을 위한 초기 가상 하중은 질량 행렬의 대각항 또는 각 행의 합으로 근사화되고, 식 (1)과 같은 정적 문제를 구성하게 된다. 구해진 해는 정규화 과정을 통해 첫 번째 리츠벡터를 얻게 된다.

$$[K]\{x^{(0)}\} = \{M_{ii}\}, \{x^{(0)}\} = \{x^{(0)}\} / \sqrt{\{x^{(0)}\}^T [M] \{x^{(0)}\}} \quad (1)$$

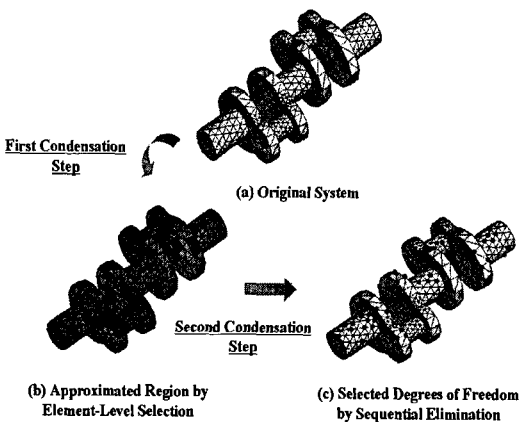


Fig. 1 Schematic of two-level condensation scheme

식 (1)에서 구해진 리츠벡터는 질량 행렬과의 연산을 통해 두 번째 리츠벡터를 구하기 위한 가상하중 벡터로 사용된다. 이전 단계에서 구해진 벡터들은 Gram-Schmit 과정을 통해 현재 단계의 벡터로부터 그 성분들을 제거함으로써 상호 직교성을 보장하게 된다. 식 (2)는 Gram-Schmit 과정이며, 원하는 개수의 리츠벡터를 얻을 때까지 반복된다.

$$\{x^{(i)}\} = \{x^{(0)}\} - \sum_{k=1}^{i-1} [\{x^{(k)}\}^T [M] \{x^{(i)}\}] \{z^{(k)}\} \quad (2)$$

2.1.2 요소단위의 에너지 평가

구해진 리츠벡터에 의해 요소단위의 레일리 에너지 지수가 계산된다.

$$\lambda_e^i = \sum_{k=1}^n \frac{\{z^{(k)}\}^T [K_e^i] \{z^{(k)}\}}{\{z^{(k)}\}^T [M_e^i] \{z^{(k)}\}} \quad (3)$$

여기서, i 는 요소번호, $[K_e^i], [M_e^i]$ 는 요소단위의 강성 및 질량 행렬을 나타낸다. $\{z^{(k)}\}$ 는 k 번째 리츠 벡터를 의미한다. 식 (3)에 의해 각 요소의 에너지가 계산되면, 작은 값을 가지는 요소부터 후보영역으로 선정된다. 경험적으로, 선정되는 요소 개수는 전체 시스템의 약 10% 정도로 결정하면 신뢰할 수 있는 결과를 얻을 수 있다.

2.1.3 가중치를 고려한 에너지 평가

구조물에 따라서 저차모드와 고차모드의 레일리-에너지 지수값의 편차가 큰 경우, 식 (3)에 의한 후보 영역의 선정은 저차모드를 강조하는 경향으로 나타날 수 있다. 이것은 구조물에 높은 주파수 하중이 가해지거나, 일정한 스펙트럼에서 구조물의 주파수를 살펴보고자 하는 경우, 해석 결과의 신뢰성을 떨어뜨리게 된다. 이런 문제점은 가중치를 고려함으로써 해결할 수 있다. 요소별 가중치 값을 강성과 질량행렬의 대각항으로부터 얻어진다. 강성행렬과 질량행렬이 $(N \times N)$ 행렬일 경우, 강성 및 질량 행렬의 대각항으로 구성된 벡터의 크기는 $(N \times 1)$ 이 된다. 이 벡터에서 각 요소의 자유도에 해당되는 성분을 추출하고, 내적을 수행하여 요소별 가중치를 계산하게 된다. 식 (4)는 요소별 가중치 계산식이다. 가중치에 대해서는 참고문헌 [5]를 참조하면 된다.

$$w_e^i = (\psi_i / \psi_{max})^2, \psi_i = [K_e^i / M_e^i]^T [K_{ii}^e / M_{ii}^e] \quad (4)$$

가중치를 고려하여 요소별 에너지 평가값을 다시 계산하면 식 (5)로 주어지게 된다.

$$\Pi_e^i = \sum_{j=1}^k \lambda_e^j w_e^j \quad (5)$$

식 (5)의 가중치 값이 작을수록 저차 모드에 큰 영향을 주는 요소로 판정하며, 고주파 영역으로 갈수록 가중

치의 값이 증가하게 됨을 알 수 있다. 만약, 주파수의 분포가 선형적으로 증가한다면, 가중치의 영향력에 의한 요소 선정 결과는 큰 변화가 없다. 그러나, 저주파와 고주파 사이의 간격이 큰 경우, 가중치가 고려되지 않는다면 요소 선정은 저차 모드에만 치우치는 결과를 얻게 되어 몇 개의 저차 모드에서만 신뢰성을 갖게 된다. 따라서, 이 경우에는 가중치를 포함함으로써 고차 모드 영역에 영향력을 갖는 요소를 선정하게 된다.

2.1.4 제 1 단계 축소시스템 구성

후보영역 선정에 의해 주절점을 선정 한 후, 첫 번째 축소시스템이 구성된다. 본 연구에서는 Guyan 방법의 오차를 상당히 감소할 수 있고, 여타 개선 방법과 비교하여 오차가 크지 않은 IRS 방법을 사용하였다. 일반적으로 고유치 문제는 식 (6)으로 주어진다.

$$[K]\{\phi\} = \lambda[M]\{\phi\} \quad (6)$$

λ 와 $\{\phi\}$ 는 각각 고유값과 고유모드를 나타낸다. 식 (6)의 강성행렬과 질량행렬을 주자유도와 부자유도로 분할하면 식 (7)과 같다.

$$\begin{Bmatrix} K_{pp} & K_{ps} \\ K_{sp} & K_{ss} \end{Bmatrix} \begin{Bmatrix} \phi_p \\ \phi_s \end{Bmatrix} = \lambda \begin{Bmatrix} M_{pp} & M_{ps} \\ M_{sp} & M_{ss} \end{Bmatrix} \begin{Bmatrix} \phi_p \\ \phi_s \end{Bmatrix} \quad (7)$$

여기서, $\{\phi_p\}$ 은 주자유도, $\{\phi_s\}$ 부자유도를 나타낸다. 식 (7)에서 얻어지는 두 개의 식에서 주자유도와 부자유도의 관계를 결정하는 변환행렬 $[T]$ 를 얻을 수 있고, 이와 관련된 행렬 연산은 식 (8)과 식 (9)에 주어진다.

$$\{\phi\} = \begin{Bmatrix} \phi_p \\ \phi_s \end{Bmatrix} = \begin{Bmatrix} I \\ T_{IRS} \end{Bmatrix} \{\phi_p\} = [T]\{\phi_p\} \quad (8)$$

$$\begin{aligned} [T_{IRS}] &= [T_s] + [K_{ss}]^{-1} ([M_{sp}] + [M_{ss}][T_s])[M_G]^{-1} [K_G] \\ [T_s] &= -[K_{ss}]^{-1} [K_{sp}] \\ [K_G] &= [K_{pp}] - [K_{ps}][K_{ss}]^{-1}[K_{sp}] \end{aligned} \quad (9)$$

$$[M_G] = [M_{pp}] + [M_{ps}][T_s] + [T_s]^T [M_{sp}] + [T_s]^T [M_{ss}][T_s]$$

식 (8)의 변환 행렬에 의해 축소시스템의 고유치 문제는 식(10)으로 재구성되고, 강성 및 질량 행렬은 식 (11)과 같이 주어지게 된다.

$$[K_R]\{\phi_p\} = \lambda_R [M_R]\{\phi_p\} \quad (10)$$

$$\begin{aligned} [K_R] &= [K_{pp}] + [T_{IRS}]^T [K_{sp}] + [K_{ps}][T_{IRS}] + [T_{IRS}]^T [K_{ss}][T_{IRS}] \\ [M_R] &= [M_{pp}] + [T_{IRS}]^T [M_{sp}] + [M_{ps}][T_{IRS}] + [T_{IRS}]^T [M_{ss}][T_{IRS}] \end{aligned} \quad (11)$$

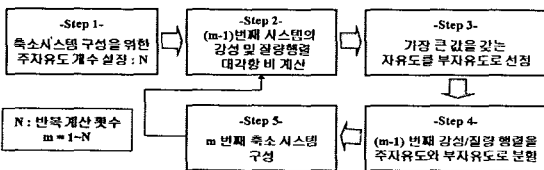


Fig. 2 Overall algorithm of sequential elimination scheme

2.2 제 2 단계 축소 과정 : 최종 주자유도 선정

제 1 단계에서 레일리 에너지 지수 평가에 의해 선정된 요소는 주자유도 선정을 위한 후보 영역이다.

제 2 단계에서는 순차적 소거법이 적용되어 최종 주자유도를 선택하게 된다. 일반적으로, 순차적 소거법은 많은 시간 비용을 요구하지만, 여기에서는 1 단계 축소과정을 거친 시스템에 적용되므로 많은 전산비용과 시간을 요구하지 않는다. Figure 2 는 2 단계에서 적용되는 순차적 소거법의 전체적인 알고리즘을 보여주고 있다.

3. 영역분할 기법과 축소시스템의 연동

축소기법은 중요하지 않은 자유도를 주자유도와 인터페이스 자유도로 표현하여 시스템의 크기를 줄이는 방법이다. 그러나, 간단한 시스템에서는 축소시스템을 구축하는데 시간소요가 부담이 되지 않지만, 대형구조물에서는 축소시스템을 구성하기 위하여 상당한 계산 시간을 필요로 한다. 그리고, 대형구조물에서는 중요 자유도의 추출 결과가 지엽적이거나, 복잡한 형상에서는 저차 모드만을 강조하는 경향으로 뽑힐 가능성이 많다. 그러므로, 축소시스템을 구축할 때, 축소시스템의 효율성을 보장하기 위한 방법이 필요하며 이것은 영역분할 기법과의 연동을 통해 확보될 수 있다. 또한, 해석 영역을 적절하게 몇 개의 부영역으로 분할한 후, 각 부영역에 추출할 주자유도의 개수를 할당함으로써, 전 영역에서 고르게 주자유도를 결정할 수가 있다. Figure 3 은 전체시스템을 2 개의 부영역으로 나눈 형상이다. 각 부영역은 식 (12)와 식 (13) 처럼, 주자유도 u_p , 부자유도 u_s , 인터페이스 자유도 u_b 로 분리된다. 각 부영역의 결합에 의해 전체시스템은 식 (14)로 주어진다.

$$\begin{Bmatrix} K_{pp} & K_{ps} & K_{pb} \\ K_{sp} & K_{ss} & K_{sb} \\ K_{bp} & K_{bs} & K_{bb} \end{Bmatrix}_{(1)} \begin{Bmatrix} u_p \\ u_s \\ u_b \end{Bmatrix}_{(1)} = \lambda \begin{Bmatrix} M_{pp} & M_{ps} & M_{pb} \\ M_{sp} & M_{ss} & M_{sb} \\ M_{bp} & M_{bs} & M_{bb} \end{Bmatrix}_{(1)} \begin{Bmatrix} u_p \\ u_s \\ u_b \end{Bmatrix}_{(1)} \quad (12)$$

$$\begin{Bmatrix} K_{pp} & K_{ps} & K_{pb} \\ K_{sp} & K_{ss} & K_{sb} \\ K_{bp} & K_{bs} & K_{bb} \end{Bmatrix}_{(2)} \begin{Bmatrix} u_p \\ u_s \\ u_b \end{Bmatrix}_{(2)} = \lambda \begin{Bmatrix} M_{pp} & M_{ps} & M_{pb} \\ M_{sp} & M_{ss} & M_{sb} \\ M_{bp} & M_{bs} & M_{bb} \end{Bmatrix}_{(2)} \begin{Bmatrix} u_p \\ u_s \\ u_b \end{Bmatrix}_{(2)} \quad (13)$$

$$\begin{Bmatrix} K_{pp}^{(1)} & K_{ps}^{(1)} & K_{pb}^{(1)} \\ K_{sp}^{(1)} & K_{ss}^{(1)} & K_{sb}^{(1)} \\ K_{bp}^{(1)} & K_{bs}^{(1)} & K_{bb}^{(1)} \end{Bmatrix} \begin{Bmatrix} u_p^{(1)} \\ u_s^{(1)} \\ u_b^{(1)} \end{Bmatrix} + \begin{Bmatrix} K_{pp}^{(2)} & K_{ps}^{(2)} & K_{pb}^{(2)} \\ K_{sp}^{(2)} & K_{ss}^{(2)} & K_{sb}^{(2)} \\ K_{bp}^{(2)} & K_{bs}^{(2)} & K_{bb}^{(2)} \end{Bmatrix} \begin{Bmatrix} u_p^{(2)} \\ u_s^{(2)} \\ u_b^{(2)} \end{Bmatrix} = \lambda \begin{Bmatrix} M_{pp}^{(1)} & M_{ps}^{(1)} & M_{pb}^{(1)} \\ M_{sp}^{(1)} & M_{ss}^{(1)} & M_{sb}^{(1)} \\ M_{bp}^{(1)} & M_{bs}^{(1)} & M_{bb}^{(1)} \end{Bmatrix} \begin{Bmatrix} u_p^{(1)} \\ u_s^{(1)} \\ u_b^{(1)} \end{Bmatrix} + \begin{Bmatrix} M_{pp}^{(2)} & M_{ps}^{(2)} & M_{pb}^{(2)} \\ M_{sp}^{(2)} & M_{ss}^{(2)} & M_{sb}^{(2)} \\ M_{bp}^{(2)} & M_{bs}^{(2)} & M_{bb}^{(2)} \end{Bmatrix} \begin{Bmatrix} u_p^{(2)} \\ u_s^{(2)} \\ u_b^{(2)} \end{Bmatrix} \quad (14)$$

여기서, $K_{bb} = K_{bb}^{(1)} + K_{bb}^{(2)}, M_{bb} = M_{bb}^{(1)} + M_{bb}^{(2)}$

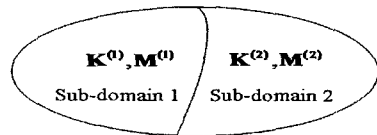


Fig. 3 Assembly of two sub-domains

각 부영역의 부자유도를 제거하기 위해서, 식 (14)의 2 번째와 4 번째 행으로부터 다음과 같은 2 개의 방정식을 얻을 수 있다.

$$\begin{aligned} \mathbf{K}_{sp}^{(1)} \mathbf{u}_p^{(1)} + \mathbf{K}_{ss}^{(1)} \mathbf{u}_s^{(1)} + \mathbf{K}_{sb}^{(1)} \mathbf{u}_b &= \lambda (\mathbf{M}_{sp}^{(1)} \mathbf{u}_p^{(1)} + \mathbf{M}_{ss}^{(1)} \mathbf{u}_s^{(1)} + \mathbf{M}_{sb}^{(1)} \mathbf{u}_b) \\ \mathbf{K}_{sb}^{(2)} \mathbf{u}_b + \mathbf{K}_{ss}^{(2)} \mathbf{u}_s^{(2)} + \mathbf{K}_{sp}^{(2)} \mathbf{u}_p^{(2)} &= \lambda (\mathbf{K}_{sb}^{(2)} \mathbf{u}_b + \mathbf{K}_{ss}^{(2)} \mathbf{u}_s^{(2)} + \mathbf{K}_{sp}^{(2)} \mathbf{u}_p^{(2)}) \end{aligned} \quad (15)$$

여기서, 식 (15)의 첫 번째 식은 부영역 1 과 관련된 부자유도 방정식이다. 변환식을 얻는 과정을 간략히 보이기 위해 부영역 1 에 대해서만 식 전개 과정을 보이도록 한다. 축소시스템을 구성하기 위해서는 중요하지 않은 부자유도를 주자유도와 인터페이스 자유도로 표현해야 하며, 식 (15)의 첫 번째 식을 정리하여, 식 (16)과 같은 형태로 나타낼 수 있다.

$$(\mathbf{K}_{ss}^{(1)} - \lambda \mathbf{M}_{ss}^{(1)}) \mathbf{u}_s^{(1)} = -(\mathbf{K}_{sp}^{(1)} - \lambda \mathbf{M}_{sp}^{(1)}) \mathbf{u}_p^{(1)} - (\mathbf{K}_{sb}^{(1)} - \lambda \mathbf{M}_{sb}^{(1)}) \mathbf{u}_b \quad (16)$$

여기서, 역행렬 $(\mathbf{K}_{ss}^{(1)} - \lambda \mathbf{M}_{ss}^{(1)})^{-1}$ 의 전개 방식은 중요한 의미를 갖게 된다. 이 식에서 고유치 항을 무시하고 전개하면, Guyan 방법에 의해 축소시스템을 구축하게 된다. IRS 기법은 고유치의 첫 번째 항까지 고려한 것이며, 식 (17)에 표현된다. 이 식을 통해 부영역 1 의 부자유도는 주자유도와 인터페이스 자유도로 표현될 수 있으며, 부영역 2 의 변환관계도 유사하게 구해진다. 이 관계로부터, 식 (18)의 전체변환 행렬에서 보여지는 것처럼, 주자유도, 부자유도, 인터페이스 자유도로 구성된 5 개의 자유도영역이 주자유도와 인터페이스 자유도로 구성된 3 개의 영역으로 축소된다.

$$\begin{aligned} \mathbf{u}_s^{(1)} &= -[\mathbf{K}_{ss}^{(1)}]^{-1} \mathbf{K}_{sp}^{(1)} \mathbf{u}_p^{(1)} - [\mathbf{K}_{ss}^{(1)}]^{-1} \mathbf{K}_{sb}^{(1)} \mathbf{u}_b \\ &+ [\mathbf{K}_{ss}^{(1)}]^{-1} (\mathbf{M}_{sp}^{(1)} - \mathbf{M}_{ss}^{(1)} [\mathbf{K}_{ss}^{(1)}]^{-1} \mathbf{K}_{sp}^{(1)}) \lambda \mathbf{u}_p^{(1)} \\ &+ [\mathbf{K}_{ss}^{(1)}]^{-1} (\mathbf{M}_{sb}^{(1)} - \mathbf{M}_{ss}^{(1)} [\mathbf{K}_{ss}^{(1)}]^{-1} \mathbf{K}_{sb}^{(1)}) \lambda \mathbf{u}_b \end{aligned} \quad (17)$$

각 성분은 식 (19)에 주어져 있다. $\mathbf{u}_p^{(1)}$, \mathbf{u}_b , $\mathbf{u}_p^{(2)}$ 은 각각 부영역 1 의 주자유도 변위, 인터페이스 변위, 부영역 2 의 주자유도 변위를 의미한다.

$$\begin{bmatrix} \mathbf{u}_p^{(1)} \\ \mathbf{u}_s^{(1)} \\ \mathbf{u}_b \\ \mathbf{u}_s^{(2)} \\ \mathbf{u}_p^{(2)} \end{bmatrix} = \begin{bmatrix} \mathbf{I} & 0 & 0 \\ \mathbf{T}_{IRS}^{sp(1)} & \mathbf{T}_{IRS}^{sb(1)} & 0 \\ 0 & \mathbf{I} & 0 \\ 0 & \mathbf{T}_{IRS}^{sb(2)} & \mathbf{T}_{IRS}^{sp(2)} \\ 0 & 0 & \mathbf{I} \end{bmatrix} \begin{bmatrix} \mathbf{u}_p^{(1)} \\ \mathbf{u}_b \\ \mathbf{u}_p^{(2)} \end{bmatrix} = \mathbf{T} \begin{bmatrix} \mathbf{u}_p^{(1)} \\ \mathbf{u}_b \\ \mathbf{u}_p^{(2)} \end{bmatrix} \quad (18)$$

$$\begin{aligned} \mathbf{T}_{IRS}^{sp(1)} &= \mathbf{T}_{sp}^{(1)} + [\mathbf{K}_{ss}^{(1)}]^{-1} (\mathbf{M}_{sp}^{(1)} + \mathbf{M}_{ss}^{(1)} \mathbf{T}_{sp}^{(1)}) \lambda \\ \mathbf{T}_{IRS}^{sb(1)} &= \mathbf{T}_{sb}^{(1)} + [\mathbf{K}_{ss}^{(1)}]^{-1} (\mathbf{M}_{sb}^{(1)} + \mathbf{M}_{ss}^{(1)} \mathbf{T}_{sb}^{(1)}) \lambda \\ \mathbf{T}_{IRS}^{sb(2)} &= \mathbf{T}_{sb}^{(2)} + [\mathbf{K}_{ss}^{(2)}]^{-1} (\mathbf{M}_{sb}^{(2)} + \mathbf{M}_{ss}^{(2)} \mathbf{T}_{sb}^{(2)}) \lambda \\ \mathbf{T}_{IRS}^{sp(2)} &= \mathbf{T}_{sp}^{(2)} + [\mathbf{K}_{ss}^{(2)}]^{-1} (\mathbf{M}_{sp}^{(2)} + \mathbf{M}_{ss}^{(2)} \mathbf{T}_{sp}^{(2)}) \lambda \\ \mathbf{T}^{(1)} &= -[\mathbf{K}_{ss}^{(1)}]^{-1} \mathbf{K}_{sp}^{(1)}, \mathbf{T}^{(2)} = -[\mathbf{K}_{ss}^{(2)}]^{-1} \mathbf{K}_{sp}^{(2)} \end{aligned} \quad (19)$$

Guyan 방법은 식 전개가 쉽고, 계산시간도 많이 걸리지 않으며, 프로그램화 하는 것도 상대적으로 간편하다. 그러나, 이 방법은 상당한 오차를 포함하기

때문에 공학적인 의미를 갖는 해를 얻기는 어렵다. 따라서, Guyan 시스템에서 신뢰성 있는 해를 얻기 위해서는 sub-space iteration 과 같은 반복 계산과정이나, 추가적인 계산이 필요하게 된다. 그러나, 대형 구조물에서는 시스템이 축소되어 구성된다 하더라도, 수십 회의 반복계산은 상당한 전산자원과 계산시간을 요구하게 되고, 동적 해석이나 최적화 문제에서는 그 부담은 더욱 커지게 된다. 그러므로, 본 연구에서는 고유치 1 차 항을 포함함으로써, 반복계산을 하지 않더라도 초기 계산에서 정확한 해를 제공하는 IRS 시스템을 채용한다.

식 (19)에서, λ 은 $[\mathbf{M}_G]^{-1} [\mathbf{K}_G]$ 과 같이 표현될 수 있다. 여기서, $[\mathbf{K}_G], [\mathbf{M}_G]$ 는 식 (18)에서 고유치 항을 무시하고 구축된 변환행렬로부터 구성된 Guyan 시스템을 의미한다. 식 (18)의 변환관계에 의해 구성되는 최종 축소시스템은 식 (20)과 같으며, 축소된 강성행렬의 성분은 식 (21)과 같다(1st version system). 질량 행렬의 성분도 동일하게 구해진다. 실제 내부 프로그램에서는 각 부영역마다 독립적으로 변환과정이 진행되기 때문에 전체시스템을 구성할 필요가 없으며, 전체 시스템에 해당하는 메모리가 요구되지 않는다. 즉, 각 부영역에 해당하는 메모리만을 설정해서 축소시스템을 구성하면 된다.

$$\begin{bmatrix} \mathbf{K}_{IRS}^{(11)} & \mathbf{K}_{IRS}^{(12)} & 0 \\ \mathbf{K}_{IRS}^{(21)} & \mathbf{K}_{IRS}^{(22)} & \mathbf{K}_{IRS}^{(23)} \\ 0 & \mathbf{K}_{IRS}^{(32)} & \mathbf{K}_{IRS}^{(33)} \end{bmatrix} \begin{bmatrix} \mathbf{u}_p^{(1)} \\ \mathbf{u}_b \\ \mathbf{u}_p^{(2)} \end{bmatrix} = \lambda \begin{bmatrix} \mathbf{M}_{IRS}^{(11)} & \mathbf{M}_{IRS}^{(12)} & 0 \\ \mathbf{M}_{IRS}^{(21)} & \mathbf{M}_{IRS}^{(22)} & \mathbf{M}_{IRS}^{(23)} \\ 0 & \mathbf{M}_{IRS}^{(32)} & \mathbf{M}_{IRS}^{(33)} \end{bmatrix} \begin{bmatrix} \mathbf{u}_p^{(1)} \\ \mathbf{u}_b \\ \mathbf{u}_p^{(2)} \end{bmatrix} \quad (20)$$

$$\begin{aligned} \mathbf{K}_{IRS}^{(11)} &= \mathbf{K}_{pp}^{(1)} + \mathbf{K}_{ps}^{(1)} \mathbf{T}_{IRS}^{sp(1)} + \mathbf{T}_{IRS}^{sp(1)\top} \mathbf{K}_{sp}^{(1)} + \mathbf{T}_{IRS}^{sp(1)\top} \mathbf{K}_{ss}^{(1)} \mathbf{T}_{IRS}^{sp(1)} \\ \mathbf{K}_{IRS}^{(12)} &= \mathbf{K}_{pb}^{(1)} + \mathbf{K}_{ps}^{(1)} \mathbf{T}_{IRS}^{sb(1)} + \mathbf{T}_{IRS}^{sp(1)\top} \mathbf{K}_{sb}^{(1)} + \mathbf{T}_{IRS}^{sp(1)\top} \mathbf{K}_{ss}^{(1)} \mathbf{T}_{IRS}^{sb(1)} \\ \mathbf{K}_{IRS}^{(22)} &= \mathbf{K}_{bb}^{(2)} + \mathbf{K}_{bs}^{(2)} \mathbf{T}_{IRS}^{sb(2)} + \mathbf{T}_{IRS}^{sb(2)\top} \mathbf{K}_{sb}^{(2)} + \mathbf{T}_{IRS}^{sb(2)\top} \mathbf{K}_{ss}^{(2)} \mathbf{T}_{IRS}^{sb(2)} \\ &+ \mathbf{K}_{bs}^{(2)} + \mathbf{K}_{bs}^{(2)} \mathbf{T}_{IRS}^{sp(2)} + \mathbf{T}_{IRS}^{sb(2)\top} \mathbf{K}_{sp}^{(2)} + \mathbf{T}_{IRS}^{sb(2)\top} \mathbf{K}_{ss}^{(2)} \mathbf{T}_{IRS}^{sp(2)} \\ \mathbf{K}_{IRS}^{(23)} &= \mathbf{K}_{bp}^{(2)} + \mathbf{K}_{bs}^{(2)} \mathbf{T}_{IRS}^{sp(2)} + \mathbf{T}_{IRS}^{sb(2)\top} \mathbf{K}_{sp}^{(2)} + \mathbf{T}_{IRS}^{sb(2)\top} \mathbf{K}_{ss}^{(2)} \mathbf{T}_{IRS}^{sp(2)} \\ \mathbf{K}_{IRS}^{(33)} &= \mathbf{K}_{pp}^{(2)} + \mathbf{K}_{ps}^{(2)} \mathbf{T}_{IRS}^{sp(2)} + \mathbf{T}_{IRS}^{sp(2)\top} \mathbf{K}_{sp}^{(2)} + \mathbf{T}_{IRS}^{sp(2)\top} \mathbf{K}_{ss}^{(2)} \mathbf{T}_{IRS}^{sp(2)} \end{aligned} \quad (21)$$

4. 솔리드 요소에서 영역분할기법 개선

4.1 솔리드 요소에서의 시스템 구축 문제점

앞에서 제안한 영역분할 기법은 인터페이스 자유도가 많지 않은 2 차원 구조물에서 적절한 기법이다. 인터페이스 자유도는 시스템 거동에 미치는 중요성과는 무관하게 각 부영역의 결합을 위해서는 반드시 포함되어야 하며, 이는 최종적인 축소시스템의 크기를 증가시킬 뿐만 아니라, 구성 과정에서도 연산시간과 전산자원을 증가시키는 원인이 된다. 그러나, 쉘 구조물은 주자유도 또는 부자유도와 비교해 볼 때, 많은 인터페이스 자유도를 포함하지 않는 것이 일반적이므로 큰 문제가 되지 않는다. 그러나, 솔리드 요소에서는

인터페이스 자유도를 무조건 포함시키면 많은 문제를 일으키게 된다.

Figure 4 는 flip chip 에서 하단의 PCB 기판과 solder 부분을 Ansys 에서 모델링 한 것이며, 9 개의 부영역에 의해 결합되었다. Figure 5 는 솔리드 요소에서 축소 시스템을 구성할 때, 인터페이스 자유도에 의해 발생하는 문제점을 보여주고 있다. 나머지 부영역과의 결합을 위해서는 인터페이스 자유도 부분이 포함되어야 하며, 그림에서 과도한 인터페이스 자유도가 포함되고 있음을 알 수 있다. 또한, Figure 5 는 7 개의 부영역이 이미 결합된 상태로서 그림상에는 표시되지 않았으나, 상당수의 인터페이스 자유도가 각 부영역간에 포함되어 있다. 이 문제에서는 선정되는 주자유도는 860 개로 전체시스템의 약 1.3% 정도 밖에 되지 않지만, 많은 수의 인터페이스 자유도(7752 개)가 추가되므로 전체시스템 크기의 약 12%로 축소시스템의 크기가 증가되며, 이는 계산 시간의 증가뿐만 아니라, 전산자원의 부족으로 인해 최종 시스템 구축을 불가능하게 만드는 원인이 된다. 따라서, 솔리드 요소가 사용되거나, 상당수의 인터페이스 자유도를 포함하는 경우, 이 문제의 해결이 필요하며, 시스템의 새로운 정식화가 요구된다.

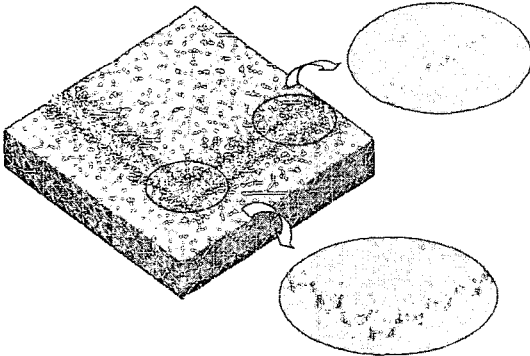


Fig. 4 Element configuration of the flip chip

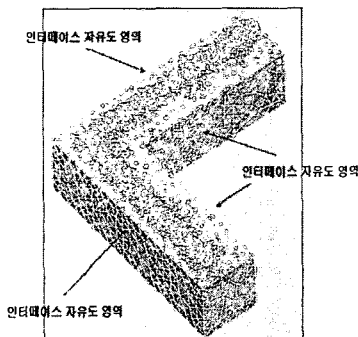


Fig. 5 Example of the interface area in the flip chip

4.2 영역 분할 기법의 개선

개선된 정식화는 아래 식에 주어져 있다. Figure 3 에 주어 진 2 개의 부영역으로 구성된 구조물을 통해 그 과정을 간략히 설명하면, 처음 단계에서 각 부영역을 3 개의 변위장(주자유도, 부자유도, 인터페이스 영역)으로 분리 한 후, 부자유도 부분을 제거한다. 이 과정을 통해 주자유도와 인터페이스 자유도 만으로 구성된 각 부영역의 축소시스템이 구축된 후, 인터페이스 영역을 주자유도와 부자유도 영역으로 분리하여 시스템을 다시 재분할한다. 즉, 각 부영역의 축소시스템이 구축된 후, 각 부영역의 축소시스템을 구성하는 주자유도와 인터페이스 자유도를 다시 주자유도, 인터페이스 주자유도 그리고 인터페이스 부자유도의 3 개 영역으로 분리한다. 여기서, 주자유도와 인터페이스 주자유도를 재구성되는 시스템의 주자유도로 할당하고, 인터페이스 부자유도를 부자유도로 할당한다. 식 (22)는 인터페이스 영역을 주자유도와 부자유도로 분리한 후, 시스템을 재구성한 것이다. 이 관계식에서 인터페이스 영역의 부자유도를 제거하기 위해 두 번째 행으로부터 식 (23)의 관계식을 얻을 수 있다.

$$\begin{bmatrix} \mathbf{K}_{pp}^{(1)} & \mathbf{K}_{ps}^{(1)} & \mathbf{K}_{pp}^{(1)} & 0 \\ \mathbf{K}_{sp}^{(1)} & \mathbf{K}_{ss}^{(1)} & \mathbf{K}_{sp}^{(1)} & \mathbf{K}_{ss}^{(1,2)} \\ \mathbf{K}_{pp}^{(1)} & \mathbf{K}_{ps}^{(1)} & \mathbf{K}_{pp}^{(1)} & \mathbf{K}_{ps}^{(1,2)} \\ 0 & \mathbf{K}_{ps}^{(2)} & \mathbf{K}_{pp}^{(2)} & \mathbf{K}_{ps}^{(2)} \end{bmatrix} \begin{bmatrix} u_p^{(1)} \\ u_s^{(1)} \\ u_p^{(1)} \\ u_p^{(2)} \end{bmatrix} = \lambda \begin{bmatrix} \mathbf{M}_{pp}^{(1)} & \mathbf{M}_{ps}^{(1)} & \mathbf{M}_{pp}^{(1)} & 0 \\ \mathbf{M}_{sp}^{(1)} & \mathbf{M}_{ss}^{(1)} & \mathbf{M}_{sp}^{(1)} & \mathbf{M}_{ss}^{(1,2)} \\ \mathbf{M}_{pp}^{(1)} & \mathbf{M}_{ps}^{(1)} & \mathbf{M}_{pp}^{(1)} & \mathbf{M}_{ps}^{(1,2)} \\ 0 & \mathbf{M}_{ps}^{(2)} & \mathbf{M}_{pp}^{(2)} & \mathbf{M}_{ps}^{(2)} \end{bmatrix} \begin{bmatrix} u_p^{(1)} \\ u_s^{(1)} \\ u_p^{(1)} \\ u_p^{(2)} \end{bmatrix} \quad (22)$$

$$\begin{aligned} 2^{nd} \text{ row} : & \mathbf{K}_{sp}^{(1)} u_p^{(1)} + \mathbf{K}_{ss}^{(1)} u_s^{(1)} + \mathbf{K}_{sp}^{(1)} u_p^{(1)} + \mathbf{K}_{ss}^{(1,2)} u_p^{(2)} \\ & = \lambda \{ \mathbf{M}_{sp}^{(1)} u_p^{(1)} + \mathbf{M}_{ss}^{(1)} u_s^{(1)} + \mathbf{M}_{sp}^{(1)} u_p^{(1)} + \mathbf{M}_{ss}^{(1,2)} u_p^{(2)} \} \end{aligned} \quad (23)$$

식 (23)로부터, 식 (24)의 주자유도와 부자유도 사이의 관계를 얻을 수 있다.

$$(\mathbf{K}_{ss}^{(1)} - \lambda \mathbf{M}_{ss}^{(1)}) u_s^{(1)} = - \left(\left[\mathbf{K}_{sp}^{(1)} \quad \mathbf{K}_{sp}^{(1)} \quad \mathbf{K}_{sp}^{(1,2)} \right] - \lambda \left[\mathbf{M}_{sp}^{(1)} \quad \mathbf{M}_{sp}^{(1)} \quad \mathbf{M}_{sp}^{(1,2)} \right] \right) \begin{bmatrix} u_p^{(1)} \\ u_p^{(1)} \\ u_p^{(2)} \end{bmatrix} \quad (24)$$

식 (24)의 행렬은 식 (25)와 같이 정의된다.

$$\begin{aligned} \mathbf{A}^{\nu 3} &= \mathbf{K}_{ss}^{(1)}, \mathbf{B}^{\nu 3} = \mathbf{M}_{ss}^{(1)} \\ \mathbf{C}^{\nu 3} &= \left[\mathbf{K}_{sp}^{(1)} \quad \mathbf{K}_{sp}^{(1)} \quad \mathbf{K}_{sp}^{(1,2)} \right], \mathbf{D}^{\nu 3} = \left[\mathbf{M}_{sp}^{(1)} \quad \mathbf{M}_{sp}^{(1)} \quad \mathbf{M}_{sp}^{(1,2)} \right] \end{aligned} \quad (25)$$

여기서, 고유치 항 λ 을 고려하여, IRS 시스템을 구축하면 식 (26)과 같다.

$$u_s^{(1)} = \left\{ -(\mathbf{A}^{\nu 3})^{-1} \mathbf{C}^{\nu 3} + (\mathbf{A}^{\nu 3})^{-1} (\mathbf{D}^{\nu 3} - \mathbf{B}^{\nu 3} (\mathbf{A}^{\nu 3})^{-1} \mathbf{C}^{\nu 3}) \lambda \right\} \begin{bmatrix} u_p^{(1)} \\ u_p^{(1)} \\ u_p^{(2)} \end{bmatrix} \quad (26)$$

주자유도와 인터페이스 주자유도 그리고 인터페이스 영역의 부자유도 사이의 관계를 고려하여, 각 부영역의 주자유도와 인터페이스 주자유도로 구성된 3 개의 변위장으로 축소된다. 식 (26)에서 IRS 시스템 구성을 위해서 먼저 질량 효과를 제거한 Guyan 시스템이 구성 되어야 한다. 관련된 식은 (27)에 주어져 있다. $\mathbf{T}_e^{\nu 3}$ 은

Guyan 시스템과 관련된 변환행렬, $T_R^{\nu 3}$ 은 IRS 시스템 구축을 위한 변환행렬이다.

$$T_G^{\nu 3} = -(A^{\nu 3})^{-1} C^{\nu 3} \quad (a)$$

$$T_R^{\nu 3} = T_G^{\nu 3} + (A^{\nu 3})^{-1} (D^{\nu 3} - B^{\nu 3} (A^{\nu 3})^{-1} C^{\nu 3}) \lambda \quad (b)$$

식 (27)의 변환관계를 행렬형태로 정리하면 식 (28)과 같다.

$$\begin{bmatrix} u_p^{(1)} \\ u_p^{(i)} \\ u_p^{(j)} \\ u_p^{(2)} \end{bmatrix} = \begin{bmatrix} I & 0 & 0 \\ G T_{sp}^{(1)} & G T_{sp}^{(i)} & G T_{sp}^{(2)} \\ 0 & I & 0 \\ 0 & 0 & I \end{bmatrix} \begin{bmatrix} u_p^{(1)} \\ u_p^{(i)} \\ u_p^{(j)} \\ u_p^{(2)} \end{bmatrix} = T_G^{\nu 3} \begin{bmatrix} u_p^{(1)} \\ u_p^{(i)} \\ u_p^{(j)} \\ u_p^{(2)} \end{bmatrix} \quad (28)$$

위의 변환식에 의해 Guyan 시스템, $K_G^{\nu 3}, M_G^{\nu 3}$ 이 구축된다. Guyan 시스템은 식 (27-b)의 변환 행렬 구축시, λ 를 $(M_G^{\nu 3})^{-1} K_G^{\nu 3}$ 으로 근사화하기 위해 사용된다. 최종 IRS 시스템은 식 (29)와 같이 구축된다.

$$K_R^{\nu 3} = T_R^{\nu 3 T} \begin{bmatrix} K_{pp}^{(11)} & K_{ps}^{(1i)} & K_{pp}^{(1j)} & 0 \\ K_{sp}^{(i1)} & K_{ss}^{(ii)} & K_{sp}^{(ij)} & K_{ss}^{(22)} \\ K_{ps}^{(j1)} & K_{ps}^{(ji)} & K_{pp}^{(jj)} & K_{ps}^{(22)} \\ 0 & K_{ps}^{(2i)} & K_{ps}^{(2j)} & K_{pp}^{(22)} \end{bmatrix} T_R^{\nu 3} = \begin{bmatrix} K_{pp}^{(11)} & K_{pp}^{(1i)} & 0 \\ K_{pp}^{(i1)} & K_{pp}^{(ii)} & K_{ps}^{(i2)} \\ 0 & K_{ps}^{(2i)} & K_{ps}^{(22)} \end{bmatrix} \quad (29)$$

$$M_R^{\nu 3} = T_R^{\nu 3 T} \begin{bmatrix} M_{pp}^{(11)} & M_{ps}^{(1i)} & M_{pp}^{(1j)} & 0 \\ M_{sp}^{(i1)} & M_{ss}^{(ii)} & M_{sp}^{(ij)} & M_{ss}^{(22)} \\ M_{ps}^{(j1)} & M_{ps}^{(ji)} & M_{pp}^{(jj)} & M_{ps}^{(22)} \\ 0 & M_{ps}^{(2i)} & M_{ps}^{(2j)} & M_{pp}^{(22)} \end{bmatrix} T_R^{\nu 3} = \begin{bmatrix} M_{pp}^{(11)} & M_{pp}^{(1i)} & 0 \\ M_{pp}^{(i1)} & M_{pp}^{(ii)} & M_{ps}^{(i2)} \\ 0 & M_{ps}^{(2i)} & M_{ps}^{(22)} \end{bmatrix}$$

5. 수치예제

5.1 Flip chip

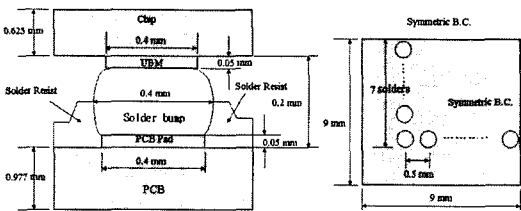
Figure 6 과 7 은 flip chip 형상이다. 축소시스템을 구성하기 위해 Fig. 8(a)와 같이 9 개의 부영역으로 분리하였다. 분리된 각 부영역을 구성하는 총 자유도 수, 주자유도 수, 인터페이스 자유도 수는 Table 1 에 주어져 있다. 이 표에서 인터페이스 자유도는 해의 영역에 중요한 영향을 미치는 성분을 거의 포함하지 못하고 있음을 알 수 있다. 예를 들어, 5,6,7 번째 부영역에서는 인터페이스에

서 주자유도를 하나도 포함하지 못하고 있으며, 대부분의 자유도가 부자유도로 할당됨을 알 수 있다. 따라서, 모든 인터페이스를 포함하여 축소시스템을 구성하는 것은 매우 비효율적이라는 것을 예측할 수 있다.

각 부영역으로부터 주자유도 선정을 위해 약 10% 의 요소를 후보영역으로 선정하였다. 후보영역 선정 결과는 Fig. 8(b)와 같다. 각 부영역의 후보 영역으로부터 선정된 주자유도들에 의해 약 1.5% 크기의 최종 시스템이 구축되었으며, 고유치 해석 결과는 Table 2 에 주어져 있다. 만약, 각 부영역 사이에 존재하는 인터페이스 자유도를 모두 포함하게 된다면, 약 8%의 자유도를 포함하는 시스템으로 축소시스템의 크기가 증가하게 된다. 이 자유도 중 대부분은 해의 정확성에 거의 영향력이 없기 때문에, 이런 시스템을 구축하는 것은 상당히 비효율적이라고 할 수 있으며, 반복계산이 요구될 경우에는 지나친 전산자원과 계산시간 증가의 원인이 된다. Table 2 는 1 차부터 15 차 모드까지 축소시스템과 ANSYS 에 의한 고유치 해석 결과를 비교한 것이다. 최대 오차는 약 0.7% 정도로 나타나고 있으며, 제안된 기법을 통해 대형시스템에서도 매우 신뢰할 수 있는 결과를 제공할 수 있다.

Table 1 Total number of PDOF and IDOFs in the sub-domains of the flip chip system

Sub-domain	Number of Node	Number of DOF	PDOF	IDOFs		
				IDOF	IPDOF	ISDOF
1	2,242	6,726	98	816	2	814
2	1,529	4,587	93	1200	7	1193
3	3,204	9,612	88	816	12	804
4	2,233	6,699	82	768	18	750
5	2,778	8,334	100	768	0	768
6	2,093	6,279	100	768	0	768
7	3,041	9,123	100	816	0	816
8	2,159	6,477	77	900	3	897
9	2,729	8,187	76	900	4	896
Sum	20,676	62,028	814	7752	46	7706



(a) Structure I of Flip chip (b) Structure II of Flip chip

Fig. 6 Configuration of the flip chip

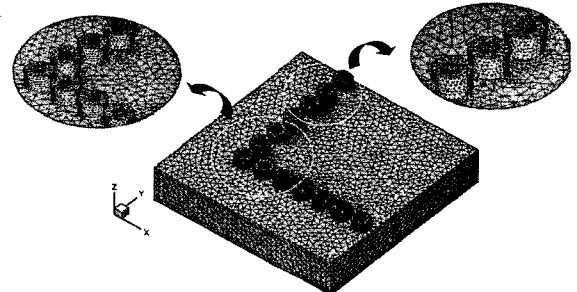


Fig. 7 Mesh configuration of the flip chip

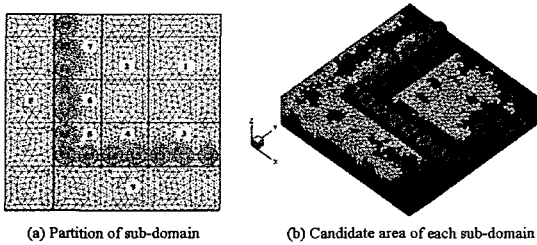


Fig. 8 Configuration of each sub-domain and candidate area in the flip chip

Table 2 Eigenvalue analysis of global system and reduced system in the flip chip

Mode No.	Full system (Ansys) Freq.(Hz)	Reduced system Freq.(Hz)	
		3 rd version (1.5%)	Error (%)
1	255.99	255.13	0.33
2	266.36	265.43	0.34
3	267.75	267.59	0.05
4	283.64	282.89	0.26
5	309.64	310.23	0.19
6	336.12	336.11	0.00
7	346.39	346.16	0.06
8	378.16	378.65	0.13
9	379.41	380.21	0.21
10	383.12	381.32	0.46
11	393.38	391.43	0.49
12	409.72	408.71	0.24
13	409.90	410.26	0.08
14	432.18	431.07	0.25
15	453.93	455.10	0.25

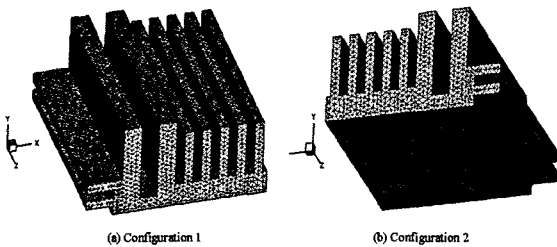


Fig. 9 Mesh configuration of the CPU fin system (E = 22000 Mpa, poisson ratio = 0.28, density = 7850kg/m³)

5.2 CPU fin

Figure 9는 CPU fin 의 3 차원 모델이다. 이 시스템은 10 개의 부영역으로 분리되었고, 각 부영역 형상과 부영역에서 선정된 후보영역이 Fig. 10 에 주어져 있다. 최종적으로 선정된 주자유도 개수는 780 개이며, 이것은 전체시스템의 1.2%에 해당한다. 앞의 예제와

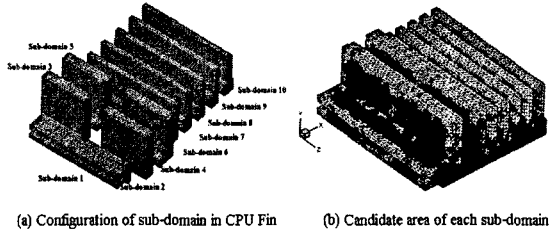


Fig. 10 Configuration of each sub-domain and candidate area in the CPU fin

Table 3 Comparison of the eigenvalue analysis in the CPU fin

Mode No.	Full system (Ansys) Freq.(Hz)	Reduced system Freq.(Hz)	
		3 rd version (1.2%)	Error (%)
1	265.65	265.75	0.04
2	280.10	279.82	0.10
3	281.81	282.19	0.13
4	282.60	282.11	0.17
5	287.35	287.45	0.03
6	289.14	289.23	0.03
7	309.05	309.17	0.04
8	340.27	340.45	0.05
9	341.91	342.01	0.03
10	34207	34215	0.02
11	346.28	346.78	0.14
12	347.94	347.73	0.06
13	35259	35228	0.09
14	391.61	391.75	0.04
15	499.15	500.04	0.18

유사하게 인터페이스 자유도를 모두 포함하게 되면 약 10% 이상의 자유도를 포함하는 시스템이 구축되며, 이것은 해의 정확성에는 거의 영향이 없으면서, 시스템의 자원만을 증가 시키게 된다. Table 3 은 1 차부터 15 차 모드까지 Ansys 에서 전체시스템의 고유치를 해석한 결과와 축소시스템 해석 결과를 비교한 것이다. 최대 오차는 0.2% 이하로 나타나고 있으며, 대형 시스템에서도 상당히 정확한 고유치 결과를 보여주고 있음을 알 수 있다.

5.3 축소시스템의 효율성

Figure 11 은 전체시스템과 축소시스템간의 최종 시스템 구축과 고유치 계산까지의 시간 비교를 보여주고 있다. Figure 11 의 결과값은 Figure 9 의 CPU fin 예제의 절점 수를 4800, 6500, 12000 으로 바꿔가면서 계산 시간을 구한 것이다. Figure 11(a)의 전체시스템에서는 모드 수와 절점 수가 증가하면서 계산시간이 exponential 형태로 증가하고 있음을 알 수 있으며, 12,000 절점에서 20 개의 모드

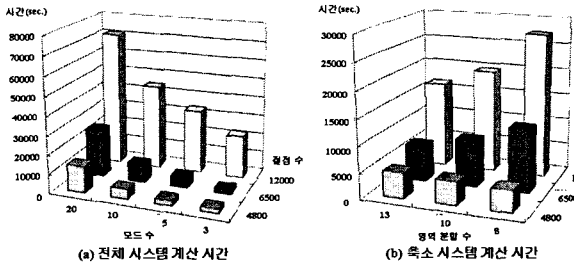


Fig. 11 Time comparison between global system and reduced system

선정하는 방법으로 2 단계 축소기법을 채용하였다. 그리고, 축소시스템의 효율성을 확보하기 위해 영역분할 기법의 연동을 통한 시스템 구축기법을 제안하였다. 수치예제를 통해 제안된 기법의 신뢰성을 제공하였으며, 본 연구에서 제안한 기법은 대형구조물의 최적화 및 동적 해석 등 과도한 시스템 자원이 요구되는 문제에서 매우 유용할 것으로 사료된다. 추후 연구에서는 비선형 영역에서 축소시스템을 구축하는 문제로 확장될 것이며, 구조물의 손상탐지, 자기진단, 실시간 시뮬레이션 분야로 응용분야를 넓혀 나갈 수 있다.

를 구할 때, 약 80,000 초 정도의 계산 시간이 요구됨을 알 수 있다. Figure 11(b)는 축소시스템의 계산시간을 보여주고 있다. 축소 시스템에서는 이미 구축된 시스템을 이용한 계산은 시간소요가 거의 없으므로, 최종 시스템이 구축될 때까지의 소요시간이 중요하다. 결과에서는 구축될 때까지의 시간과 고유치 계산시간을 모두 포함한 것이지만, 대부분의 시간이 시스템 구축에 소요된다고 할 수 있다.

후 기

본 연구는 2005 년 삼성전자 메카트로닉스 연구소의 지원으로 연구가 수행되었으며, 관계자 여러분께 감사드립니다.

결과에서 분할되는 영역의 수가 증가할수록 시스템 구축을 위해 소요되는 시간은 감소하고 있으며, 그 경향은 점점 수가 증가할수록 더욱 확연하게 나타나고 있다. 즉, 시스템의 크기가 증가하게 되면, 전체시스템의 소요 시간은 증가하게 되고, 축소시스템에서는 영역분할 수가 증가할수록 시간 감소에 따른 효율성은 증가하게 된다. 이런 경향은 고유치 계산뿐만 아니라, 최적화나 비선형 해석과 같이 반복적인 계산이 요구되는 문제에서 매우 확연하게 보여진다. 그러나, 최적의 축소시스템을 구축하기 위해 전체시스템을 몇 개의 영역으로 분할해야 하고, 각 영역마다 어느 정도의 자유도를 부여해야 하는 지에 대한 연구는 계속 되어야 할 것으로 사료된다.

참고문헌

- (1) Gyan, R.J., 1965, Reduction of Stiffness and Mass Matrices, *AIAA Journal*, Vol. 3, No. 2, p. 380.
- (2) O'Callahan, J., 1989, A Procedure for an Improved Reduced System(IRS) Model. Proceedings of the 7th International Modal Analysis Conference, Union college, Schenectady, NY, pp. 17~21.
- (3) Zhang, D.W., Li, S., 1995, Succession-Level Approximate Reduction(SAR) Technique for Structural Dynamic Model. Analysis Conference (Nashville,TN), Union college press, Schenectady, NY, pp. 435~441.
- (4) Shah, V.N., Raymund, M., 1982, Analytical Selection of Masters for the Reduced Eigenvalue Problem, *Int.J.Numer. Mech. Engng.*, Vol. 18, No. 1, pp. 89~98.
- (5) Kim, K.O., Choi, Y.J., 2000, Energy Method for Selection of Degrees of Freedom in Condensation. *AIAA Journal*, Vol. 38, No. 7, pp. 1253~1259

6. 결 론

본 연구에서는 축소시스템 구축을 위해 중요 자유도를