

# 유동계산을 위한 다단계 부분 구조법에 대한 연구

김진환<sup>1</sup>

## A STUDY ON A MULTI-LEVEL SUBSTRUCTURING METHOD FOR COMPUTATIONS OF FLUID FLOW

J.W. Kim

*Substructuring methods are often used in finite element structural analyses. In this study a multi-level substructuring(MLSS) algorithm is developed and proposed as a possible candidate for finite element fluid solvers. The present algorithm consists of four stages such as a gathering, a condensing, a solving and a scattering stage. At each level, a predetermined number of elements are gathered and condensed to form an element of higher level. At the highest level, each sub-domain consists of only one super-element. Thus, the inversion process of a stiffness matrix associated with internal degrees of freedom of each sub-domain has been replaced by a sequential static condensation of gathered element matrices. The global algebraic system arising from the assembly of each sub-domain matrices is solved using a well-known iterative solver such as the conjugate gradient(CG) or the conjugate gradient squared(CGS) method. A time comparison with CG has been performed on a 2-D Poisson problem. With one domain the computing time by MLSS is comparable with that by CG up to about 260,000 d.o.f. For 263,169 d.o.f. using 8 x 8 sub-domains, the time by MLSS is reduced to a value less than 30 % of that by CG. The lid-driven cavity problem has been solved for Re = 3200 using the element interpolation degree(Deg.) up to cubic. In this case, preconditioning techniques usually accompanied by iterative solvers are not needed. Finite element formulation for the incompressible flow has been stabilized by a modified residual procedure proposed by Ilinca et al.[9].*

**Key Words:** 부분구조(Substructure), 응축(Condensation), 부영역(Subdomain)

### 1. 서 론

영역 분할(domain decomposition) 기법은 최근 병렬 처리와 관련하여 많은 연구자들의 관심이다[1]. 이 영역 분할 방법은 대체로 부영역(subdomain)들의 겹침(overlapping)과 비겹침(non-overlapping) 방법의 두 종류로 나누어지며, 겹침 영역 분할법은 종종 Schwarz 방법이라 칭하고 비겹침 영역 분할법은 반복적 부구조화 방법(iterative substructuring method)이라고도 불린다[2]. 부영역이란 풀이 과정의 수월성을 도모한 기하학적 및 위상학적인 고려에 의한 영역의

일부라 정의한다. 본 연구에서는 두 가지 분할 기법 중 비겹침 영역 분할에 대하여 수행하고자 한다.

비겹침 영역 분할 혹은 부분 구조(substructuring)법은 부영역의 내부 자유도들을 소거한 Schure complement 행렬을 구성한 후, 이 행렬들을 조합(assembly)하여 보다 작은 수의 자유도에 대한 전체(global) 행렬을 구성하게 된다.

부분 구조법의 전개 과정을 살펴보면, 부영역(j)에 대한 강성 행렬, 즉  $A_{jj}^{(j)}$ 에 대한 Schure complement 행렬  $S^{(j)}$ 의 구성이 요구되며, 이를 위하여  $A_{jj}^{(j)}$ 의 역행렬이 구해져야 한다. 이는  $A_{jj}^{(j)}$ 에 대한 새로운 대수 방정식을 풀어야 함을 의미하며, 부영역의 수가 많고 또 각각의 부영역이 많은 수의 내부 자유도를 포함한다면, 이를 위한 효율적인 해결책이 먼저 선행되어야 한다. 이 연구는 영역

접수일: 2004년11월30일, 심사완료일: 2005년 5월 9일.

<sup>1</sup> 정회원, 동의대학교 기계공학부

E-mail: kimjw@deu.ac.kr

분할법(비결침)의 전개 과정이 유한 요소법에서 내부 자유도를 제거하기 위한 정적 응축(static condensation) 과정과 유사함에 주목하고, 이를 연속적으로 수행함으로써 동일한  $S^{(j)}$ 를 구성할 수 있음을 관찰하였다. 즉, 초기의 요소들은 응축과정을 통하여 더 큰 새로운 요소가 형성되고 이의 연속적 수행을 통하여 최종적으로 부 영역은 하나의 슈퍼(super) 요소로 구성된다. 이리하여 부 영역을 구성하는 최 상위 단계의 요소들에 대한 행렬 방정식을 조합하여 전체 행렬을 구성하면 계산상의 효율성을 도모할 수 있다.

이러한 개념은 유한 요소 구조 해석에서 이미 부분 구조법으로 통용되고 있었다. 일찍이 항공기나 선박 등의 대형 구조물 해석에서 각 부위별로 나누어 동시에 독립적으로 해석함이 더 효율적이라는 관점이 제기되었고, 이로부터 행렬을 나누어 계산하는 방법에 대한 연구가 시작되었다[3]. 또 부분 구조법의 효율성이 입증되면서 부분 구조 내부에 부분 구조를 가지는 다단계 부분 구조법이 Furuike[4]에 의하여 제안되었으며, 이 경우에는 실제 구조물의 계층적 특성 혹은 위상학적(topological) 관계를 이용하여 수행하게 된다. 최근 이에 대한 연구로는 코일 스프링의 해석을 위한 김 진영 외 1인[5,6]의 개발이 있었으며, 여기에서는 코일 스프링이 위상학적으로 한 방향으로 연결되어 있음을 이용하였다. 따라서 구조 해석을 위하여 제안된 다단계 부분 구조법은 다양한 기하학적 형상이 요구되는 유동장 해석에는 아직 적용되지 못하고 있다. 더욱이 일부 상용 프로그램(구조 해석용)에서는 다단계 부분 구조법을 적용하고 있다고 주장하고 있으나, 문헌상에서 이들의 개발에 대한 연구를 찾는 것은 쉽지 않다[7].

이러한 이유로 본 연구에서는 유동 해석에 적합한 다단계 부분 구조법을 제시하고자 한다.

유동 계산의 적합성을 보이기 위하여 비압축성 유동을 고려하였다. 특히 비압축성 유동 방정식은 지배 방정식이 비대칭 및 부정형(indefinite)인 관계로 이의 풀이가 쉽지 않다. 유한 요소법을 사용하는 수치 모사의 경우, Galerkin 기중 잔여치법을 그대로 적용하기에는 많은 무리가 따르며, 일찍이 Hughes와 Brooks[8]는 상류 방향의 유동 정보를 받아들일 수 있는 시험(test) 함수를 개발하여 Streamline Upwind Petrov Galerkin(SUPG)이라 하였다. 이리하여 대류항이 우세하여 야기되는 비대칭적 불안정성은 상당 부분 완화될 수 있으나, 비압축성 유동에서 속도 및 압력을 변수로 사용하는 지배 방정식의 경우에는 부정형의 문제가 남아있었다. 최근 Ilinca 등[9]은 SUPG 기법이 잔여치에 대한 풍상 차분(upwind differencing)과 유사하다는 관찰을 통

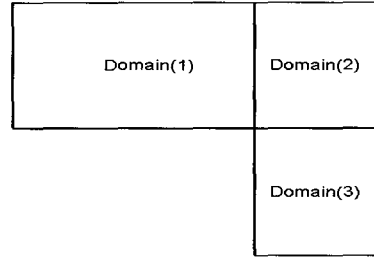


Fig. 1 An illustration of non-overlapping three sub-domains

하여 비압축성 유동 방정식의 잔여치에 대한 섭동(perturbation)을 도입한 수정 잔여치법을 제안하였다. 따라서 본 연구에서도 이를 적용하여 정식화하고 다단계 부분 구조법을 이용하여 풀이하였다. 현재의 연구는 유동 해석에 적용이 가능한 다단계 부분 구조법의 개발과 비압축성 유동 계산을 통한 이의 안정성 및 효율성 검증에 있다.

## 2. 다단계 부분 구조법

### 2.1. 부분 구조법

영역을 분할하지 않으면 다음과 같은 큰 행렬을 풀어야 한다.

$$A x = b \tag{1}$$

영역이 Fig. 1과 같이 분할된다면 비 결침 분할의 경우 알고리즘은 다음과 같다[2].

1) 각각의 부 영역(j)에 대한 강성(stiffness) 행렬  $A_j$ 를 내부 자유도와 경계(외부 및 내부) 자유도에 대하여 아래와 같이 구성한다.

$$A^{(j)} = \begin{Bmatrix} A_{bb}^{(j)} & A_{bi}^{(j)} \\ A_{ib}^{(j)} & A_{ii}^{(j)} \end{Bmatrix} \tag{2}$$

2) 각각의 부 영역(j)의 내부 자유도 행렬  $A_{ii}^{(j)}$ 에 대한 Schur complement  $S^{(j)}$ 를 아래와 같이 구성한다.

$$S^{(j)} = A_{bb}^{(j)} - A_{bi}^{(j)} A_{ii}^{(j)-1} A_{ib}^{(j)} \tag{3}$$

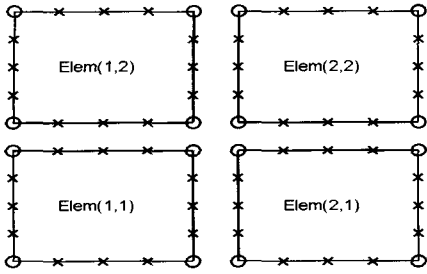


Fig. 2 Elements to be gathered

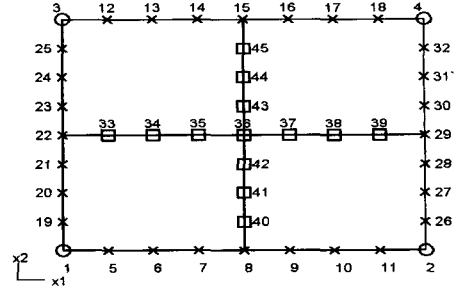


Fig. 3 A gathered element

3) 각각의 부 영역(j)에 대하여 우변 벡터  $\underline{b}^{(j)}$  를 구성한 후, 경계(외부 및 내부) 자유도에 대한 새로운 우변  $\underline{g}_b^{(j)}$  를 아래와 같이 구성한다.

$$\underline{g}_b^{(j)} = \underline{b}_b^{(j)} - A_{bi}^{(j)} A_{ii}^{(j)-1} \underline{b}_i^{(j)} \quad (4)$$

여기서  $\underline{b}^{(j)} = [\underline{b}_b^{(j)} \quad \underline{b}_i^{(j)}]^T$  이다.

4) 성긴(coarse) 강성 행렬  $S$ 를  $S^{(j)}$  의 조합(assembly)으로부터 구성하고, 또  $\underline{g}_b$  를  $\underline{g}_b^{(j)}$  의 조합으로 구성한다.

$$S = \sum_{(j)} S^{(j)} \quad (5)$$

$$\underline{g}_b = \sum_{(j)} \underline{g}_b^{(j)} \quad (6)$$

필요에 따라서는  $S$  및  $\underline{g}_b$  를 자유도 특성, 즉 면, 변 혹은 꼭지점에 따른 자유도 순으로 재구성할 수도 있다.

5) 적절한 풀이 방법을 도입하여 내부 및 외부 경계에 대한 해  $\underline{x}_b$  를 아래 식으로부터 구한다.

$$S \underline{x}_b = \underline{g}_b \quad (7)$$

6) 내부 자유도에 대한 우변 벡터  $\underline{g}_i^{(j)}$  를 아래와 같이 구성한다.

$$\underline{g}_i^{(j)} = \underline{b}_i^{(j)} - A_{ii}^{(j)-1} A_{ib}^{(j)} \underline{x}_b \quad (8)$$

7) 각각의 부 영역에 대한 내부 자유도 값들을 구한다.

$$\underline{x}_i^{(j)} = A_{ii}^{(j)-1} \underline{g}_i^{(j)} \quad (9)$$

이상과 같이 부분 구조법 혹은 영역 분할법의 성공 여부는 식 (7)의 적절한 풀이에 있다고 보여지며, 이는 또 문제의 특성에 많이 좌우된다. 만일 개별 부 영역의 내부 자유도 행렬  $A_{ii}^{(j)}$ 의 자유도 수가 많아서 역행렬 계산에 시간이 많이 걸리거나 혹은 행렬의 특성상 반올림 오차를 많이 유발한다면, 직접적인 역행렬 연산을 피해야 한다. 이 연구에서는 식 (2)에서 식 (9)까지의 과정을 요소 단계 별로 적용하여 부 영역내의 자유도 수가 많더라도  $A_{ii}^{(j)}$ 의 역행렬을 아래의 순차적 결합, 응축 및 분배를 통하여 효율적으로 구하고자 하였다.

**2.2. 순차적 결합, 응축 그리고 분배 (Sequential Gathering, Condensation and Scattering)**

부 영역을 구성하는 요소들은 부 영역을 연속적으로 세분화하는 과정에서 생성시킬 수 있으며, 이 개념은 프랙탈(fractal) 알고리즘과 비슷하다. 즉, 요소의 구성을 단계(level)별로 하여, 하위 단계의 요소(세분화된 요소)들은 상위 단계의 요소(큰 요소)의 분할로 이루어진다면 양 단계 간에 어떤 규칙성을 찾을 수 있다. 따라서 식 (2)에서 식 (6)까지의 과정을 각 단계(level)별 요소들에 대하여 수행하면 최종 단계에서는 부 영역들에 대한 요소 행렬식을 구성하게 된다.

Fig. 2에는 임의 단계(l-1)에 대한 2 x 2 요소를, Fig. 3에는 2 x 2 요소들의 결합을, 그리고 Fig. 4에는 상위 단계(l)의 요소를 각각 보여준다. 우리는 Fig. 2에서 Fig. 3로 진행되는 과정을 결합(gathering) 과정이라 부르기로 하며, 이 과정은 아래와 같이 경계 및 내부 절점들에 대한 식들이

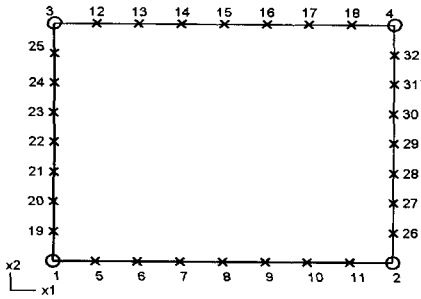


Fig. 4 An element of higher level

분리될 수 있는 방정식을 구성함으로써 정의된다.

$$\begin{pmatrix} A_{bb}(l) & A_{bi}(l) \\ A_{ib}(l) & A_{ii}(l) \end{pmatrix} \begin{pmatrix} \underline{u}_b(l) \\ \underline{u}_i(l) \end{pmatrix} = \begin{pmatrix} f_b(l) \\ f_i(l) \end{pmatrix} \quad (10)$$

여기서, 아래 첨자  $b$ 는 요소 경계상의 절점을 의미하고, 아래 첨자  $i$ 는 요소 내부 절점들을 의미한다. 식 (10)의 좌변 행렬을 단계  $l$ 의 결합 행렬  $A(l)$ 이라 하고, 해를  $\underline{u}(l)$  그리고 우변 벡터를 결합 벡터  $f(l)$ 이라 하면 식 (10)은 다음과 같이 쓸 수 있다.

$$A(l)\underline{u}(l) = f(l) \quad (11)$$

$$\begin{aligned} \text{여기서 } \underline{u}(l) &= [\underline{u}_b(l), \underline{u}_i(l)]^T, \\ f(l) &= [f_b(l), f_i(l)]^T \end{aligned}$$

이다. Fig. 3의 결합 요소에서 Fig. 4의 요소를 구성하기 위하여, 식 (10)으로부터 내부 자유도를 소거한 새로운 강성 행렬  $S(l)$ 과 우변 벡터  $g(l)$ 을 다음과 같은 응축(condensation) 과정을 이용하여 구한다.

$$S(l) = A_{bb} - A_{bi}A_{ii}^{-1}A_{ib} \quad (12-1)$$

$$g(l) = f_b - A_{bi}A_{ii}^{-1}f_i \quad (12-2)$$

여기서  $L$ 을 부영역의 단계(최대 단계)라 하면  $l=2, \dots, L$ 이 된다. 따라서 행렬  $S(l)$ 과 벡터  $g(l)$ 은 단계- $l$ 의 새로운 강성 행렬과 우변 벡터로서 아래 식을 만족시킨다.

$$S(l)\underline{u}_b(l) = g(l) \quad (13)$$

식 (13)은 단계- $l$ 에 대한 새로운 요소 방정식이다. 다시 이들을 결합하여 더 상위 단계의 요소를 구성하기 위해서는 식 (10) 혹은 식 (11)의 과정을 반복하면 된다. 따라서 최 상위 단계, 즉 부영역이 하나의 요소가 될 때까지 식 (10)에서 식 (12)의 과정이 반복된다.

Fig. 3에 보인 요소들이 선형 요소들의 제 3 단계의 요소라 정의하고 임의의  $(i, j)$  요소의 결합을 변환 행렬  $T$ 를 도입하여 수식으로 표현하여 보면 다음과 같다.

$$T\underline{u}(3) = \underline{u}(4) \quad (14)$$

여기서  $T$ 는  $45 \times 16$  크기의 행렬로서 단위 열(row) 벡터(예:  $[1, 0, 0, \dots, 0]$ )들로 구성된다.  $T$ 의 열 번호는 Fig. 3에 나타난 절점 번호이고 그 열 벡터에서  $(i, j)$  요소의 절점 순서 자리에 "1"이 저장된다. 따라서 제 3 단계의  $(i, j)$  요소 방정식의 제 4 단계 요소 방정식으로서의 결합은 다음과 같이 표현된다.

$$T_{i,j}A_{i,j}(3)T_{i,j}^T\underline{u}(4) = T_{i,j}f_{i,j}(3) \quad (15)$$

제 4 단계 요소 방정식의 결합 행렬과 우변 벡터는 아래와 같이 표현된다.

$$A(4) = \sum_{i,j=(1,1)}^{(M1, M2)} T_{i,j}A_{i,j}(3)T_{i,j}^T \quad (16-1)$$

$$f(4) = \sum_{i,j=(1,1)}^{(M1, M2)} T_{i,j}f_{i,j}(3) \quad (16-2)$$

여기서  $(M1, M2)$ 는 결합할 요소의 수를 나타내며 통상  $(2, 2)$ 이다. 식 (15)에서 보듯이 변환 행렬  $T$ 는 단계별과 결합 요소 수에 따라 달라진다.

최종적으로 각각의 부 영역에 대한 식 (12)와 같은 강성 행렬과 우변 벡터를 구성하게 되면, 이들을 조합(assembly)하여 전체 영역에 대한 선형 대수 방정식을 풀게 되며, 이 과정은 식 (5)~(7)과 동일하다. 식 (7)에 의하여 구해진 자유도 값( $\underline{u}_b$ )은 다시 하위 단계(부 영역 단계)로 전달된다. 부 영역에 경계 자유도 값  $\underline{u}_b(L)$ , ( $L = \text{Max. Level}$ )가 각각의 부영역에 전달되면 임의 단계- $l$ 에 대한 내부 자유도에 대한 해  $\underline{u}_i(l)$ 은 다음과 같이 구

한다

$$\underline{u}_i(l) = A_{ii}^{-1}(l) (f_i(l) - A_{ib}(l) \underline{u}_b(l)) \tag{17}$$

여기서  $l=L, \dots, 2$ 가 된다. 참고로 식 (17)의  $A_{ii}^{-1}$ ,  $f_i$  및  $A_{ib}$ 는 각 단계별 결합 과정에서 Schur complement 행렬(식 (12-1))을 구성할 때 이미 만들어져 있음을 주목한다.

다시 정리하면, 식 (10)을 결합(gathering) 과정, 식 (12)를 응축(condensing) 과정이라 구별하기로 한다. 식 (14)의 과정이 수행되면 식 (11)의  $\underline{u}(l)$ 이 구해지고, 또 이  $\underline{u}(l)$ 은 하위 단계의 해  $\underline{u}_b(l-1)$ 을 포함하므로 이 값을 하위 단계로 넘길 수 있다. 우리는 이 과정을 분배

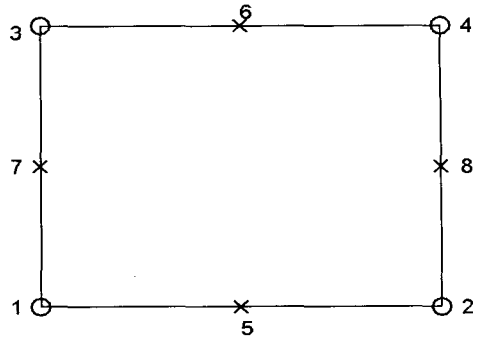


Fig. 5 A node numbering for a quadrilateral element

(scattering) 과정이라 부르기로 한다.

결합, 응축 및 분배 과정의 단계별 수행을 보기 위하여 Table. 1을 작성하여 현재의 다단계 방법의 흐름을 살펴보

Table. 1 A summary of Multi-Level procedure

Level	Restriction	Prolongation
1	Set: $S(1) \leftarrow A_{elem}$ , $g(1) \leftarrow f_{elem}$ ↓	Scatter: $\underline{u}_{elem} \leftarrow \underline{u}_b(1)$
2	1) Gather: $A(2) \leftarrow \sum S(1)$ $f(2) \leftarrow \sum g(1)$ 2) Compute and Store: $A_{ii}^{-1}(2)$ , $A_{ib}(2)$ , $f_i(2)$ 3) Condense: $S(2)$ , $g(2) \leftarrow A(2), f(2)$ ↓	↑ 3) Scatter: $\underline{u}_b(1) \leftarrow \underline{u}(2)$ 2) Make: $\underline{u}(2) \leftarrow [\underline{u}_b(2), \underline{u}_i(2)]^T$ 1) Compute: $\underline{u}_i(2) = A_{ii}^{-1}(2) [f_i(2) - A_{ib}(2) \underline{u}_b(2)]$
...	...	...
l	1) Gather: $A(l) \leftarrow \sum S(l-1)$ $f(l) \leftarrow \sum g(l-1)$ 2) Compute and Store: $A_{ii}^{-1}(l)$ , $A_{ib}(l)$ , $f_i(l)$ 3) Condense: $S(l)$ , $g(l) \leftarrow A(l), f(l)$ ↓	↑ 3) Scatter: $\underline{u}_b(l-1) \leftarrow \underline{u}(l)$ 2) Make: $\underline{u}(l) \leftarrow [\underline{u}_b(l), \underline{u}_i(l)]^T$ 1) Compute: $\underline{u}_i(l) = A_{ii}^{-1}(l) [f_i(l) - A_{ib}(l) \underline{u}_b(l)]$
...	...	...
L	...	...
	Gather: $S \leftarrow \sum S(L)$ , $g \leftarrow \sum g(L)$ Solve: $S \underline{u} = g$ ⇒⇒⇒	↑ Scatter: $\underline{u}_b(L) \leftarrow \underline{u}$ ↑

았다. Table. 1의 좌측을 **restriction**이라 하고 우측을 **prolongation**이라 하였다. 이는 잘 알려진 다중 격자법과의 유사성에서 독자들의 이해를 돕기 위함이다.

**2.3 요소의 형상함수**

이 연구에서 사용할 요소는 serendipity 형의 요소로 내부 절점이 없으며, 꼭지 점(vertex node), 변 절점(edge node) 그리고 면 절점(face node)의 절점들과 그에 상응하는 형상 함수로 정의된다. Fig. 5에는 2차원 사변형 (quadrilateral) 이차(quadratic) 요소의 절점 번호를 보여주며, 제 1 변은 절점 번호 5, 제 2 변은 절점 번호 6, 제 3 변은 절점 번호 7, 그리고 제 4 변은 절점 번호 8이 위치한 변들이다. 3차 요소는 제 1 변에 5, 6 번 절점을, 제 2 변에는 7, 8 번, 제 3 변에는 9, 10 번 그리고 제 4 변에는 11, 12 번의 절점들을 가진다. 절점 번호 순에 따른 이차 원 형상 함수( $\phi$ )는 다음과 같이 정의된다[10].

먼저 선형 요소(Deg. 1 element)에 대한 형상함수들은

$$\begin{aligned} \phi(1) &= 0.25 (1 - \eta_1) (1 - \eta_2) \\ \phi(2) &= 0.25 (1 + \eta_1) (1 - \eta_2) \\ \phi(3) &= 0.25 (1 - \eta_1) (1 + \eta_2) \\ \phi(4) &= 0.25 (1 + \eta_1) (1 + \eta_2) \end{aligned} \tag{18}$$

이다. 2차 요소에 대한 형상 함수들은 다음과 같다.

$$\begin{aligned} \phi(1) &= 0.25 (1 - \eta_1) (1 - \eta_2) (-\eta_1 - \eta_2 - 1) \\ \phi(2) &= 0.25 (1 + \eta_1) (1 - \eta_2) (\eta_1 - \eta_2 - 1) \\ \phi(3) &= 0.25 (1 - \eta_1) (1 + \eta_2) (-\eta_1 + \eta_2 - 1) \\ \phi(4) &= 0.25 (1 - \eta_1) (1 - \eta_2) (\eta_1 + \eta_2 - 1) \\ \phi(5) &= 0.5 (1 - \eta_1 \eta_2) (1 - \eta_2) \\ \phi(6) &= 0.5 (1 - \eta_1 \eta_2) (1 + \eta_2) \\ \phi(7) &= 0.5 (1 - \eta_2 \eta_2) (1 - \eta_1) \\ \phi(8) &= 0.5 (1 - \eta_2 \eta_2) (1 + \eta_1) \end{aligned} \tag{19}$$

3차 요소 그리고 3차원 요소에 대한 serendipity 형상함수는 Lapidus, et. al.[10]을 참조하면 된다.

**3. 수렴성과 효율성**

**3.1 수렴성(Convergence)**

이 절에서 우리는 해의 정밀성(accuracy)이 보간 차수와 요소 수의 증가에 따라 어떻게 변화하는지를 보기 위하여, 이차원 영역에서의 아래의 Poisson 방정식을 이용하기로

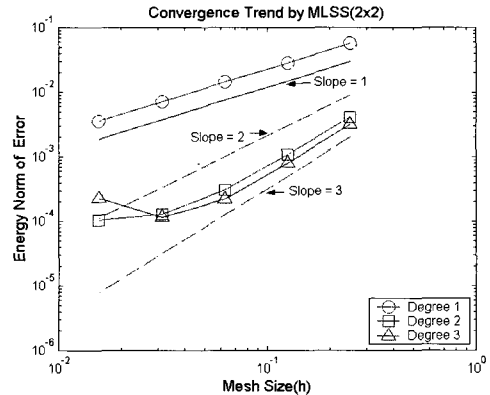


Fig. 6 Convergence rate by MLSS

한다.

$$-\nabla^2 u = 1, \quad 0 \leq x \leq 1, \quad 0 \leq y \leq 1 \tag{20}$$

경계 조건은

$$u = 0 \text{ at } x = 1 \text{ or } y = 1, \text{ 그리고 } \partial u / \partial n = 0$$

여기서  $n$ 은 외향 법선 벡터이다.

식 (20)에 대한 오차의 에너지 norm은 적절한 요소 길이,  $h$  와 형상 함수의 차수,  $k$  에 대하여 다음과 같이 정의된다.

$$\|u - u^h\|_E \leq Ch^k \tag{21}$$

오차에 대한 에너지 norm은 다음과 같이 정의된다.

$$\|u - u^h\|_E = \left[ \sum_{elem} \int_{\Omega} \left\{ \left( \frac{\partial u}{\partial x} - \frac{\partial u^h}{\partial x} \right)^2 + \left( \frac{\partial u}{\partial y} - \frac{\partial u^h}{\partial y} \right)^2 \right\} d\Omega \right]^{1/2} \tag{22}$$

식 (21)을 확인하기 위하여 우리는 요소 수를 4 x 4, ..., 64 x 64로 변화시켜 에너지 norm에 대한 log-log plot을 Fig. 6과 7에 나타내었다. Fig. 6은 현재 제시된 다단계 방법을 2 x 2 부영역으로 나누어 수행한 결과이며, Fig. 7에 나타낸 값들은 CG 방법에 의한 결과이다.

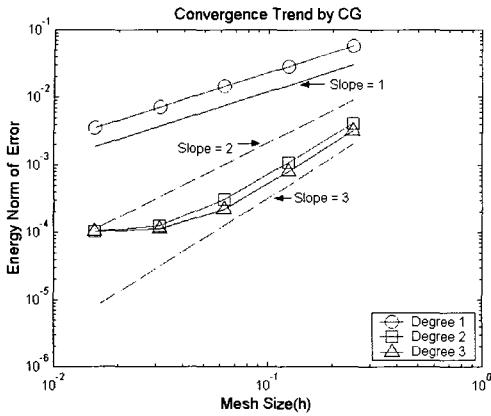


Fig. 7 Convergence rate by CG

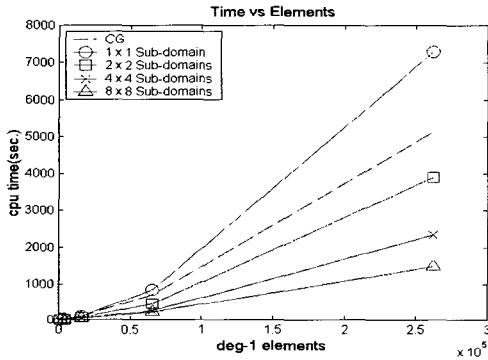


Fig. 8 Cpu time comparison with the number of elements

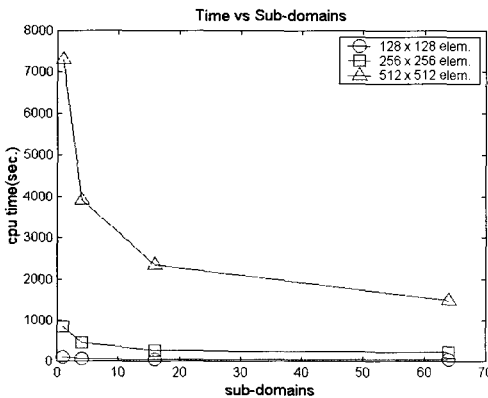


Fig. 9 CPU time v.s. number of sub-domains

Fig. 6에 의하면, 선형(Deg-1) 요소의 경우에는 이론을 잘 따르고 있음을 확인할 수 있으며, 2차(Deg-2) 요소의 경우, 요소 수가 16 x 16 까지는 이론적 수렴보다 더 좋은

수렴성을 보이다가 수렴성이 저하됨을 볼 수 있고, 3차 (Deg-3) 요소의 경우, 요소 수가 16 x 16까지는 이론적 수렴성을 따르다가 수렴성이 상당히 저하됨을 볼 수 있다.

Fig. 7에 의하면, 선형 요소와 2차 요소의 경우, 수렴성은 Fig. 6과 대동소이하며 3차 요소의 경우에는 32 x 32 및 64 x 64 요소의 수렴성은 저하되나 어떤 값으로 수렴하고 있음을 관찰할 수 있다. 참고로 이 값은 대략 10<sup>-4</sup>이며 현재 사용 중인 컴퓨터로서는 어떠한 방법으로도 그 이하로 내릴 수 없는 한계인 것으로 추정된다. 다단계법이 CG 방법에 비하여 수렴 불안정성을 보이는 이유는 식 (12-1)의 A<sub>ii</sub>의 직접적인 역행렬 계산에 의한 반올림 오차 때문이다.

해의 정확성에 영향을 주는 인자는 대체로 두 가지로 볼 수 있다. 첫째는 풀이 방법의 부적절함에 의하여 야기되는 오차이고 둘째는 계산기의 유한성 때문에 야기되는 오차이다. Fig. 7의 Deg-3 해는 첫 번째 유형의 오차를 보여주고 Fig. 8의 Deg-2와 Deg-3의 해는 두 번째 유형의 오차를 보여준다. 첫 번째 오차는 식 (12)에서 A<sub>ii</sub>의 직접적인 역행렬 계산으로 야기되며, 이의 해결로는 행렬과 벡터의 곱을 이용한 새로운 시도가 요구된다. 이는 현재의 다단계 풀이법을 예조건화 기법으로 사용하는 것과 관련이 있기에 추후의 연구로 남겨두고자 한다.

최근 64-bit 컴퓨터의 출현은 매우 고무적이며 상기 한 두 가지 오차를 상당 부분 완화시키리라 기대한다.

3.2 효율성(Efficiency)

이 절에서 우리는 MLSS와 CG와의 cpu 시간 비교를 하여 Table. 2에 나타내었다. 사용된 pc는 pentium 486급으로 2.GHz의 속도와 256MB의 RAM을 가졌다. 다단계 풀이법에서 괄호내의 숫자는 식 (7)의 풀이(최종 행렬에 대한 풀이)에 대한 시간이다. 단순 영역 분할법 혹은 부분 구조법에 대한 시간은 요소 수가 16 x 16일 경우 4초 그리고 32

Table. 2 CPU time comparison for Deg-1 elements(time in sec.)

# of elem.	CG	MLSS			
		1 x 1	2 x 2	4 x 4	8 x 8
32 x32	1	1(0.2)	1(0.3)	-	-
64 x 64	11	12(2)	8(2)	6(3)	6(4)
128x128	90	102(11)	54(13)	38(16)	34(21)
256x256	695	833(58)	456(81)	250(94)	205(118)
512x512	5122	7295 (361)	3910 (453)	2333 (594)	1484 (724)

x 32의 경우에는 256초로 64배가 소요되므로( $n^3$ 에 비례한다는 이론적 예측을 충실히 따르는 것으로 보임.) 비교 대상에서 제외하였다. Fig. 8에는 요소 수의 변화에 따른 cpu 시간의 변화를 CG와 MLSS에 대하여 비교하였다. 먼저 영역을 분할하지 않을 경우(# of sub-domain = 1), 256 x 256 요소(# of dof = 66,049)까지는 큰 차이(20% 이내)가 없으나 512 x 512 요소(# of d.o.f. = 263,169)의 경우 MLSS는 CG에 비하여 약 1.4배 정도 시간이 소요되었다. 이는 예측보다 상당히 양호한 결과이며, 영역을 분할할수록 효율성이 증가됨을 볼 수 있었다. 영역을 8 x 8로 분할하면 CG에 비하여 0.29 정도의 시간이 소요됨을 알 수 있다. Fig. 9에는 부영역의 증가에 따른 계산 시간을 나타내었다. 여기에서 시간 감소의 추세를 살펴보면 부영역의 증가에도 한계가 있음을 알 수 있으며, 이는 작은 수의 자유도 경우에는 더욱 그러하다.

현재의 MLSS 방법은 결합과정에서 직접 풀이법으로  $A_{ii}$ 의 역행렬을 계산하므로 부영역에서 자유도 수가 많으면 효율은 떨어진다. 따라서 영역을 분할할수록 좋으나 이는 또 최종 행렬의 자유도 수를 증가시켜 풀이 시간을 증가시킨다. 현재 Table. 2에 제시된 값들을 참조하면 결합 시간과 최종 행렬 방정식의 풀이 시간이 같거나 비슷하면 최적인 것으로 보이나, 이 부분은 별도의 연구가 필요하리라 본다. 현재의 단일 프로세스에서의 비교로 미루어 보건데, 다단계 부분구조법의 병렬 연산은 매우 효율적이라 판단된다.

#### 4. 이차원 비압축성 유동

##### 4.1 수정 잔여치법에 의한 정식화

이차원 유동 계산을 위한 지배 방정식은 다음과 같다.

$$\begin{aligned} -\nu \nabla^2 u + u \frac{\partial u}{\partial x} + v \frac{\partial u}{\partial y} + \frac{1}{\rho} \frac{\partial p}{\partial x} &= f_u \\ -\nu \nabla^2 v + u \frac{\partial v}{\partial x} + v \frac{\partial v}{\partial y} + \frac{1}{\rho} \frac{\partial p}{\partial y} &= f_v \\ \frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} &= 0 \end{aligned} \quad (23)$$

여기서  $u, v, p$  그리고  $\nu$ 는 각각 x-, y-방향의 속도, 압력 그리고 동점성계수이다. 위의 방정식에 적절한 가중 함수(weighting function)와 보간 함수를 취하여 내적하면 Galerkin 잔여치법에 의하여 아래의 비선형 연립 대수 방정식을 얻게 된다.

$$\begin{aligned} A_{11} \cdot U + A_{12} \cdot V + A_{13} \cdot P &= R_1 \\ A_{21} \cdot U + A_{22} \cdot V + A_{23} \cdot P &= R_2 \\ A_{31} \cdot U + A_{32} \cdot V &= R_3 \end{aligned} \quad (24)$$

여기서  $A_{11}, \dots, A_{32}$ 들은 미지수  $U, V$  및  $P$ 들에 대한 좌변 행렬들이며,  $R_1, R_2$  그리고  $R_3$ 들은 중력항 등과 경계조건 처리에 의하여 형성된 우변 벡터들이다. 식 (24)에서 행렬  $A_{33}$ 가 영 행렬인 관계로 피보팅(pivoting)을 이용한 직접 풀이법 외에는 대안이 없으며, 레이놀즈(Re) 수가 커지게 되면 직접 풀이법조차 적절한 해를 주지 못한다. 따라서 우리는 먼저 수정 잔여치법[9]을 도입하여 식 (23)을 안정화(stabilization)시키고자 한다.

수정 잔여치(modified residual)법은 Ilinca 등[9]이 제안한 방법으로 잔여치에 대한 섭동을 원래의 Galerkin 식에 추가한다. 따라서 이를 만족시키는 해는 원래의 Galerkin 식도 만족시키게 된다. 이러한 관점에서 SUPG[8]도 일종의 잔여치 풍상(residual upwinding) 차분법으로 설명하고 있다. 이를 통하여 Ilinca 등은 대류 우세에 의한 불안정뿐만 아니라 식 (24)의 부정행적 불안정성(약조건 행렬)도 안정화시키고자 하였다.

2차원 정상상태 비압축성 Navier-Stokes 방정식은 다음과 같이 정의한다.

$$M_i = \tilde{u}_j \frac{\partial u_i}{\partial x_j} + \frac{1}{\rho} \frac{\partial p}{\partial x_i} - \frac{\mu}{\rho} \frac{\partial^2 u_i}{\partial x_j^2} - \frac{f_i}{\rho} \quad (25)$$

$$C = \frac{\partial u_i}{\partial x_j} \quad (i, j = 1, 2) \quad (26)$$

위의 식에서 tilde(~) 기호를 붙인 속도 성분 ( $\tilde{u}_i$ )은 요소 중심점에서 계산한 값을 의미하며, 편도함수 행렬(Jacobian)을 계산할 때 상수로 취급한다. Navier-Stokes 방정식에 대한 수정 잔여치 방법을 적용하면 다음과 같다.

$$\begin{aligned} R_i &= \int_{\Omega} \psi_i M_i d\Omega \\ &+ \sum_{elem} \int_{\Omega_k} \tau \left( \tilde{u}_j \frac{\partial \psi_i}{\partial x_j} \right) M_i d\Omega_k \\ &+ \sum_{elem} \int_{\Omega_k} \delta \frac{\partial \psi_i}{\partial x_j} C d\Omega_k = 0 \end{aligned} \quad (27)$$



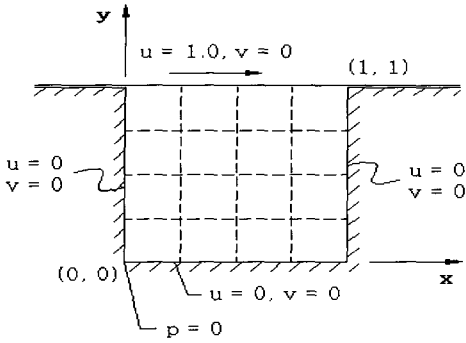


Fig. 10 Domain and boundary conditions for lid-driven cavity problem

$$R_p = \int_{\Omega} \psi_p C d\Omega + \sum_{elem} \int_{\Omega_k} \tau \frac{\partial \psi_p}{\partial x_i} M_i d\Omega_e = 0 \quad (28)$$

위의 식에서  $\Omega$ 는 경계를 포함하는 전체 해석 영역을 의미하며,  $\Omega_k$ 는 각 요소의 내부 영역만을 의미한다. 따라서 식 (27)과 (28)의 첫 번째 항들에는 부분 적분을 적용하여, 미분 차수를 한 단계 내리고 또 미분 경계 조건을 적용하게 할 수 있다.

섭동항들의 크기를 결정하는  $\tau$ 와  $\delta$ 는 레이놀즈 수 ( $Re$ )와 요소의 대표 길이  $h$ 의 함수로 Ilinca 등[9]을 참조하면 다음과 같다.

$$\tau = \frac{h_k}{2|\vec{v}|} \xi(Re_k); \quad \delta = |\vec{v}| h_k \xi(Re_k)$$

$$\xi(Re_k) = \begin{cases} Re_k & , 0 \leq Re_k < 1 \\ 1 & , Re_k \geq 1 \end{cases}$$

$$Re_k = \frac{m_k |\vec{v}| h_k}{2\nu} \quad \text{or} \quad \frac{m_k |\vec{v}| h_k}{2k/\rho c_p}$$

(  $m_k$  is 1/3 for linear element ) (29)

위의 식 (27)과 (28)을 전개하면, 종속 변수  $u_i$ 의 2차 미분항,  $\partial^2 u_i / \partial x_1^2$ 와  $\partial^2 u_i / \partial x_2^2$ 들이 포함되어 있음을 볼 수 있다. 그러나 C0-연속성을 갖는 형상함수로 이들을 취급하기가 쉽지 않으므로 현재의 정식화 과정에서는 무시하기로 한다.

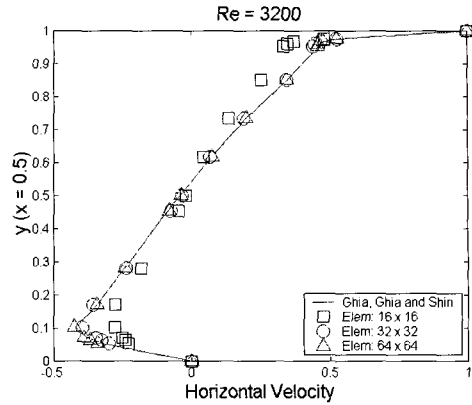


Fig. 11 Comparison of linear element solutions for  $Re = 3200$

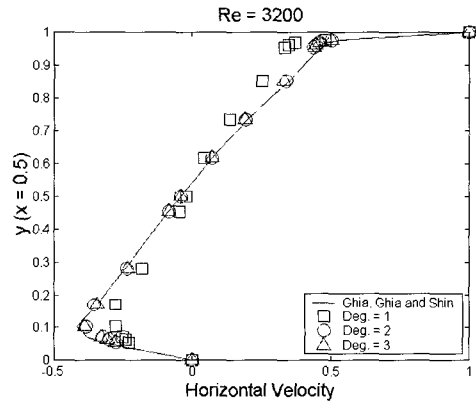


Fig. 12 Comparison of solutions from degree 1, 2 and 3 using  $16 \times 16$  elements

### 4.2 Lid Driven Cavity 유동

Cavity 유동에 대한 영역 및 경계조건의 정의는 Fig. 10에 하였다. 여기서 압력에 대한 기준 값으로 좌측 하단 모서리 점에 "0"의 값을 취하였다. 현재 제시되는 결과들은 부 영역의 수를 4개( $2 \times 2$ )로 하고 또 각각의 모서리에 어느 정도 세분화된 격자체에 대한 것들이다. 그리고 유한 요소 해의 검증은 위해 수평 속도 성분에 대하여만 보기로 한다.

Fig. 11에는 레이놀즈 수 3200에 대하여 선형 요소를  $16 \times 16$ ,  $32 \times 32$  그리고  $64 \times 64$  요소들로 세분화함에 따른 수렴성을 보여주고 있다. 이 경우,  $32 \times 32$ 와  $64 \times 64$  요소의 해는 Ghia 등[12]의 해의 추세를 따르고는 있다. Fig. 12에는  $16 \times 16$  요소 분할에 의한 1차, 2차 그리고 3차 요소에 대한 해를 보여준다. 이 경우에 2차와 3차 요소의 해는 Ghia 등의 해와 대체로 잘 일치함을 알 수 있으며, 두

해간의 우열을 가리기 어렵다. 이는 Fig. 6에서 관찰했던 바와 같이 이차 요소 해의 수렴성이 탁월했던 것과 무관하지는 않으리라 본다.

## 5. 결 론

본 연구에서는 영역 분할 방법에서 나타나는 부영역에 대한 풀이 과정의 효율성을 도모하기 위하여 순차적 결합, 응축 및 분배의 방법을 이용한 다단계 방법을 제시하였다. 이러한 방법은 유한 요소법의 사용과 관련하여 매우 자연스러운 접근이며, 이차원 Poisson 방정식을 이용하여 수렴성과 효율성을 살펴보았다.

부영역의 내부 자유도 행렬에 대한 직접적 역행렬 계산으로 인한 수렴 불안정성이 관찰되었으며, 이는 최근 출시되는 64-bit 컴퓨터를 사용하거나 혹은 부영역의 수를 증가 시킴으로 해결할 수 있으리라 본다. 효율성에 대하여도, 현재의 MLSS는 CG 방법과도 단일 영역에 있어서도 수행 시간을 비교할만하며 영역을 분할하면 CG 방법보다 더 효율적이었다. 부분 구조법은 병렬 연산에 매우 적합한 풀이 방법임으로, 병렬 처리 방법을 도입한다면 그 효율성을 극대화할 수 있으리라 본다.

현재 제시된 방법은 특정 구조물에 한정되어 사용하던 범위를 벗어나서 일반 유동장 해석에도 안정적으로 적용될 수 있음을 살펴보았다. 비압축성 유동 계산에 연계하여 수행한 결과, 일반적으로 요구되는 예 조건화가 필요하지 않았으며, 이 점은 현재의 정적 응축을 이용한 결합 과정이 일종의 예조건화임을 의미한다고 하겠다.

## 참고문헌

- [1] Hodgeson, D.C. and Jimack, P.K., 1997, "Ain Decomposition preconditioner for a Parallel Finite Element Solver on Distributed Unstructured Grids," *Parallel Computing*, Vol.24, pp.1157-1181.
- [2] Smith, B.F., 1990, "Domain Decomposition Algorithms for the Partial Differential Equations of Linear Elasticity," *Technical Report 517*, Courant Institute of Mathematical Sciences.
- [3] Logan, D.L., 1992, "A First Course in the Finite Element Method," 2nd Ed., PWS-Kent Publishing Co., Boston.
- [4] Furuike, T., 1972, "Computerized Multiple Level Substructuring Analysis," *Comput. & Struct.*, Vol.2, pp.1063-1073.
- [5] 김진영, 허훈, 2000, "다단계 부분구조법을 이용한 코일스프링의 유한요소 응력해석 I : 스프링 슈퍼요소," *한국 자동차 공학회 논문집*, 제8권, 제2호, pp.130-150.
- [6] 김진영, 허훈, 2000, "다단계 부분구조법을 이용한 코일스프링의 유한요소 응력해석 II : 검증 및 해석," *한국 자동차 공학회 논문집*, 제8권, 제3호, pp.151-162.
- [7] Elwi, A.E. and Murray, D.W., 1985, "Skyline Algorithm for Multilevel Substructure Analysis," *Int. J. for Numer. Methods in Eng.*, Vol.21, pp.465-479.
- [8] Hughes, T.J.R. and Brooks, A.N., 1979, "A Multi-dimensional upwind scheme with no crosswind diffusion," *T.J.R. Hughes(Ed.), Finite Element Methods for Convection Dominated Flows*, Vol.34, ASME, New York, pp.19-35.
- [9] Ilinca, F., Hetu, J.F. and Pelletier, D., 2000, "On stabilized finite element formulations for incompressible advective-diffusive transport and fluid flow problems," *Comput. Methods Appl. Mech. Engrg.*, Vol.188, pp.235-255.
- [10] Lapidus, L and G.F. Pinder, 1982, "Numerical Solution of Partial Differential Equations in Science and Engineering," John Wiley & Sons, Inc., New York.
- [11] Barrett, R., Berry, M., Chan, T., Demmel, J., Donato, J., Dongarra, J., Eijkhout, V., Pozo, R., Romine, C. and van der Vorst, H., 1994, "Templates for the Solution of Linear Systems; Building Blocks for Iterative Methods," *SIAM*, Philadelphia.
- [12] Ghia, U., Ghia, K.N., and Shin, C.T., 1982, "High-Re Solutions for Incompressible Flow Using the Navier-Stokes Equations and a Multigrid Method," *J. of Comp. Physics*, Vol.48, pp.387-411.