

3차원 비정렬격자계를 이용한 Navier-Stokes해의 Preconditioning에 관한 연구

남영석* · 최형권** · 유정열*

(2001년 3월 7일 접수, 2001년 8월 14일 심사완료)

Study on Preconditioning of the Navier-Stokes Equations Using 3-Dimensional Unstructured Meshes

Young Sok Nam, Hyoung Gwon Choi and Jung Yul Yoo

Key Words : Variable-Reordering(변수재배열), Navier-Stokes Equations(Navier-Stokes 방정식) ILU Preconditioning(ILU 예조건화)

Abstract

An efficient variable-reordering method for finite element meshes is used and the effect of variable-reordering is investigated. For the element renumbering of unstructured meshes, Cuthill-McKee ordering is adopted. The newly reordered global matrix has a much narrower bandwidth than the original one, making the ILU preconditioner perform better. The effect of variable reordering on the convergence behaviour of saddle point type matrix is studied, which results from P2/P1 element discretization of the Navier-Stokes equations. We also propose and test 'level(0) ILU preconditioner' and 'level(2) ILU preconditioner', which are another versions of the existing 'level(1) ILU preconditioner', for the global matrix generated by P2/P1 finite element method of incompressible Navier-Stokes equations. We show that 'level(2) ILU preconditioner' performs much better than the others only with a little extra computations.

1. 서론

예조건화(preconditioning)는 풀고자 하는 행렬이 강성(stiff)일 경우 반복해법(iterative solvers)의 효율적인 사용을 위한 필수적인 도구이다. 다양한 종류의 예조건인자(preconditioner)가 알려져 있는데 그 중 불완전 LU분해(incomplete LU

decomposition)은 대각항 예조건인자(diagonal preconditioner)와 함께 가장 많이 쓰인다. 이들은 특히 타원형 편미분 방정식의 효율적인 해법을 위하여 흔히 사용된다.⁽¹⁻⁴⁾ 불완전 LU분해는 전산 유체역학에서 타원형 형태의 비압축성유체의 압력방정식을 효과적으로 풀기 위해 Krylov 형태의 반복해법들(Conjugate Gradient Method, Generalized Minimum Residual Method, etc.)과 함께 사용된다.^(3,4) 타원형 압력방정식은 Patankar에 의해 제시된 SIMPLE 알고리즘⁽⁵⁾이나 Chorin에 의해 제시된 투영해법(projection method)⁽⁶⁾에서 연속방정식으로부터 유도된다. SIMPLE이나 투영해법과 같은 분리해법(segreated formulation)에서는 속도장과 압력장을 따로 풀게되므로 각 단계에서 얻어지는 행렬은

* 서울대학교 대학원 기계항공공학부
** 회원, 서울대학교 BK21 기계분야 연구인력양성사업단
E-mail : choi@vortex.snu.ac.kr
† 책임저자, 회원, 서울대학교 기계항공공학부
E-mail : jyyoo@plaza.snu.ac.kr
TEL : (02)880-7112 FAX : (02)883-0179

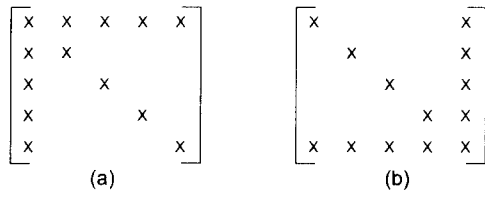


Fig. 1 Comparison of two matrices : (a) upper arrow matrix, (b) lower arrow matrix

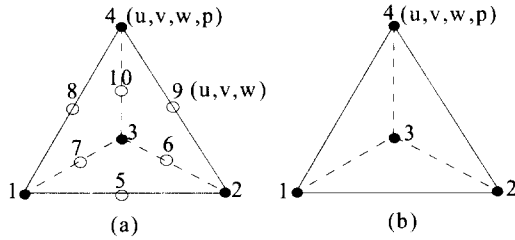


Fig. 2 Example of element type : (a) P2/P1 tetra element, (b) P1/P1 tetra element

속도장과 압력장을 동시에 풀게되는 통합공식(integrated formulation)의 그것보다 더 작고 예조건을 하기가 쉬우며 불완전 LU(0)분해를 포함한 다양한 형태의 예조건인자를 사용할 수 있다. 불완전 LU(0)분해시 예조건인자의 nonzero map은 전체조립행렬의 그것과 같다. 반면에 비압축성 Navier-Stokes 방정식을 P2P1 유한요소를 이용하여 통합공식화할 경우 생성된 전체행렬의 예조건화는 분리해법의 그것보다 상당히 복잡하며 많이 알려져 있지 않다. Navier-Stokes 방정식의 통합공식화는 분리해법에 비해 압력의 경계조건을 부과할 필요가 없고 모든 변수들이 내재적으로 결합되어 있으므로 강한 선회유동장과 같이 유동장이 복잡한 경우 좀더 견고한 수렴이력(7) 보인다는 장점이 있다. 하지만, P2P1요소를 이용한 통합공식을 Navier-Stokes 방정식의 해법을 위하여 사용할 경우 불완전 LU(0)분해는 분리해법의 그것보다 더 복잡한데, 이는 P2P1요소를 이용한 통합공식의 경우 연속방정식으로부터 영 구역(zero block)이 전체조립행렬(global assembled matrix)에 나타나 불완전 LU(0)분해시 영 피벗(zero pivot)이 발생하기 때문이다. 이 때, 전체조립행렬은 안장

점형태(saddle point type matrix)의 행렬이 된다. Hu(8)등은 고체입자-유체 상호작용을 위한 직접수치해석코드를 P2P1 유한요소 통합공식을 이용하여 개발했는데, 그들은 전체조립행렬의 예조건화를 위하여 수정된 불완전 LU(0)분해를 이용하였다. 수정된 불완전 LU(0)분해시 영 피벗을 피하기위해서 변수재배열(variable reordering) 기법이 사용되었다. 똑같은 목적을 위해서 최형권(9)은 P2P1 유한요소를 이용한 통합공식의 사용시 수정된 불완전 LU(0)분해를 예조건인자로 사용하기위해서 절점재배열(node reordering) 기법을 이용하였으며, 본 연구에서 P2P1요소의 통합공식화의 수정된 불완전 LU(0)분해를 위해 사용하였다. 본 연구에서는 이를 'level(1) ILU preconditioner'라 명명하며 2.4.2절에서 설명한다.

본 논문의 첫부분에선 절점재배열이 level(1) ILU preconditioner의 성능에 미치는 영향에 대해서 알아보고자 한다. 임의의 형상을 가지는 3차원 물체 주위의 유동장을 해석하기 위해서 생성되는 비정렬격자계는 국소적으로 격자를 집중(mesh adaption)시킬 수 있으며, 물체의 형태에 상관없이 좋은 성질의 격자를 쉽게 생성할 수 있는 장점이 있다. 그러나 정렬격자계와 비교할 때, 변수 배열(variable ordering)을 적절히 하지 못할 경우에는 생성된 전체조립행렬 G의 밴드폭(bandwidth)이 대단히 커지게 되어 LU분해(LU factorization)가 사용될 경우 L과 U행렬은 원래 행렬 G와 비교할 때 상당히 많은 추가적인 영이 아닌 항들(additional fill-ins)을 가지게 된다. 이런 결과는 LU 분해후 후진(진진)대입법을 실행할 때 계산시간의 증가를 가져오며, 그 양은 추가적인 fill-ins의 수에 비례한다. 변수배열(variable ordering)이 LU분해의 영이 아닌 항들에 미치는 영향을 잘 나타내 주는 예로서 Fig. 1에 화살표형 행렬(arrow matrix)들을 제시하였다. Fig. 1(a)에서처럼 변수배열이 되어있는 행렬을 Fig. 1(b)로 변수배열을 변환시키면 LU분해를 실행할 경우 영이 아닌 항들의 수가 매우 감소하게 되어 계산시간을 절약할 수 있다. 더욱 중요한 점은 불완전 LU분해를 두 가지 경우의 행렬에 적용하면 Fig. 1(b)로부터 생성된 불완전 LU 예조건 인자의 성능이 Fig. 1(a)에서 생성된 것보다 더 뛰어나다는 것이다. 이와 같은 결과는 불완전 LU분해로부터 나온

LU 행렬이 가진 LU에 좀더 가깝기 때문이라고 할 수 있다. 따라서 비정렬격자계를 사용하여 계산할 경우 전체 행렬의 밴드폭을 최적화시키는 것은 효율적인 예조건화를 위해 매우 중요하다.⁽¹⁰⁾ 본 연구에서는 Navier-Stokes 방정식의 이산화를 위하여 LBB조건⁽¹¹⁾을 만족시키는 P2/P1요소를 이용한 통합공식을 사용하였다. 참고로 Fig. 2에 P2/P1요소와 P1/P1요소를 도시하였다. Fig. 2(a)와 같이 속도변수는 사면체의 꼭지점과 각 변의 중점에서 정의되는 반면 압력변수는 4개의 꼭지점에서만 정의되므로, 속도장의 격자분해도가 같을 경우 (Navier-Stokes 방정식의 해의 정확도는 속도장의 분해능과만 관련이 있고, 압력변수는 단지 연속방정식을 만족시키기 위한 추가 변수로 생각할 수 있다. 압력변수의 수학적 의미는 연속방정식이란 구속조건에 대한 Lagrangian 승수가 된다.) 3차원 계산시 압력변수의 개수가 동차 선형요소(P1/P1)에 비해 약 1/8 정도로 줄어들어 기억용량을 절약할 수 있다. 반면에 P1/P1요소를 사용한 경우 LBB조건을 피하기 위해 압력안정화기법(pressure stabilizing method)⁽¹²⁾을 사용하며 이 방법은 P2P1과 비교할 때 영 구역(zero block)이라 불리는 작용자로 대치되므로 사용하기 편한 대각항예조건인자를 사용할 수 있으며, 기타 다른 예조건인자들을 좀 더 용이하게 사용할 수 있는 이점이 있다.

본 논문의 두 번째 부분에서는 기존의 'level(1) ILU preconditioner'⁽¹³⁾의 변형된 형태인 'level(0) ILU preconditioner'와 'level(2) ILU preconditioner'를 새롭게 제안한다. 변수 재배열 기법 및 제시된 예조건인자들이 수렴속도에 미치는 영향을 검토하기 위하여 3차원 구 주위의 유동과 MIRA 자동차모델 주위의 유동에 적용한다. 수치실험을 통해 변수재배열 및 개선된 ILU 알고리즘이 예조건화된 행렬의 조건수(condition number)를 줄여 수렴을 가속화시킴을 확인하고자 한다.

2. 변수재배열이 ILU preconditioner의 성능에 미치는 영향

2.1 지배방정식과 유한요소 공식화

3차원 비정상 비압축성유동장의 지배방정식은

다음과 같이 표시된다.

$$\begin{aligned} \frac{D\vec{u}}{Dt} &= \nabla \cdot \vec{\sigma} & (\vec{\sigma} &= -pI + \vec{\tau}) \\ \nabla \cdot \vec{u} &= 0 \end{aligned} \quad (1)$$

위의 지배방정식을 P2/P1 통합공식을 이용하여 유한요소 공식화하면 다음의 식이 얻어진다:

$$\begin{aligned} \text{Find } \vec{u} \in H^1, p \in L^2 \text{ such that} \\ \int_{\Omega} [\vec{u}(\frac{D\vec{u}}{Dt}) + \vec{\sigma} \cdot \nabla \vec{w}] d\Omega - \int_{\Gamma} \vec{w} \cdot \vec{\sigma} \cdot \vec{n} d\Gamma \\ + \int_{\Omega} q \nabla \cdot \vec{u} d\Omega = 0 \\ \text{for all } \vec{w} \in H^1 \text{ and } q \in L^2 \end{aligned} \quad (2)$$

공간 이산화를 위하여 P2/P1요소를 채택한 Galerkin 유한요소법을 이용하고 시간 차분에 대하여는 내재적인 방법을 이용하여 반이산화(semi-discretization)를 하면 결과적으로 아래와 같은 형태의 전체행렬을 얻을 수 있다:

$$\begin{pmatrix} A & B \\ B^T & 0 \end{pmatrix} \begin{pmatrix} \vec{u} \\ p \end{pmatrix} = \begin{pmatrix} g \\ 0 \end{pmatrix} \quad (3)$$

식 (3)의 행렬은 BICGSTAB⁽¹⁴⁾(Bi-Conjugate Gradient Stabilized)을 이용하여 풀며, 예조건화로는 수정된 불완전 LU분해를 이용한다.

2.2 ILU형 예조건화

예조건화란 어떤 선형 시스템을 주어진 반복해법(iterative solver)으로 풀기 쉽도록 강성을 완화하는 작업이다. 즉 아래 식과 같이 주어진 행렬 A가 있을 때 예조건인자로서 행렬 P를 곱해주면 행렬 A'을 얻게 되고 예조건화의 결과로 얻은 A'

은 A보다 작은 조건수 ($\kappa = \frac{|\lambda_{\max}|}{|\lambda_{\min}|}$)를 가지게

된다. 즉,

$$\begin{aligned} A\vec{x} &= \vec{b} \\ PA\vec{x} &= P\vec{b} \\ A'\vec{x} &= \vec{b}' \end{aligned} \quad (4)$$

where $\kappa(A') < \kappa(A)$

대부분의 CG(Conjugate Gradient)계열의 반복해법들의 경우 반복계산 수는 \sqrt{x} 에 비례하므로, 예조건화이란 주어진 행렬의 x 를 작게 해 주는 작업이라 말할 수 있다.

이때 P는 최대한 A^{-1} 에 가깝고 경제적으로 얻어질 수 있어야 한다. 본 연구에서는 예조건인자로서 수정된 불완전 LU분해를 사용하였다. 이 때 주의해야 할 점은 불완전 LU분해형 예조건인자의 성능을 최대한 향상시키기 위해서는 행렬 A의 nonzero 패턴을 최적화해야 한다는 사실이다. 특히 비정렬격자계를 사용할 경우 Fig. 1에서 보듯이 효과적인 nonzero map을 만들 경우 큰 성능향상을 가져올 수 있다.

본 연구에서 풀어야 할 행렬은 식 (3)과 같이 안장점형의 행렬이다. 이 때 전체행렬의 영 구역(zero block)은 연속방정식에 압력항이 포함되어 있지 않기 때문에 생기게 된다. 이 영 구역은 불완전 LU분해를 사용하는 예조건인자의 성능을 떨어뜨리기 때문에 연속방정식에 압력변수가 포함되어 있다고 가정하여 불완전 LU분해 예조건인자의 영 구역의 nonzero map 형태에 임의의 압력변수가 정의된 꼭지점에 대해 이웃하고 있는 요소 내의 꼭지점들에서 정의된 압력변수들을 내재적으로 연결된 변수들로 고려하여 추가한다. 결과적으로 원래 행렬의 제로 블록에 해당하는 부분은 불완전 LU분해된 예조건화 행렬에서는 nonzero항들을 포함하는 sparse한 행렬이 된다. 예조건화 행렬은 Fig. 3과 같이 영 구역 내에 nonzero항이 존재하는 sparse한 행렬이 되며 그 부분의 nonzero 형태는 주어진 비정렬격자계의 connectivity와 관련이 있다. 본 연구에서는 이와 같은 수정된 ILU(0) preconditioner를 사용하였으며, 이를 제로 블록의 nonzero map을 어떻게 구성하느냐에 따라 'level(#) ILU preconditioner'라 명칭하였다.

2.3 밴드폭 최적화의 기본 개념

무작위로 배열된 2차원 또는 3차원 비정렬격자계가 주어졌을 때 전체행렬의 밴드폭을 최적화하기 위해서는 먼저 어떤 절점의 번호와 그 절점이 속해 있거나 이웃하고 있는 요소 내의 절점 번호

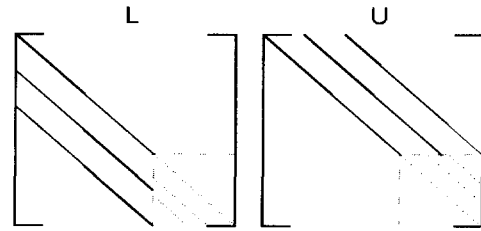


Fig. 3 Nonzero pattern of level(#) ILU preconditioner. (Dotted lines denotes additional nonzero entries in a preconditioning matrix)

들이 최대한 가까워야 하고, 각 절점들에서 정의된 변수들도 최대한 가깝게 모여있어야 한다. Fig. 4의 간단한 2-D 경우를 예로 들어보자. Fig. 4(a)의 경우처럼 절점 1에서 정의된 변수에 대해 이산화된 대수식을 유도할 때 이웃 절점(100, 150, 300)의 변수들과 내재적으로 결합이 되고, 이때 절점 1의 이웃 절점들을 Fig. 4(b)와 같이 재배열하면, 절점 2, 3, 4번의 변수값들이 배열 상에서 절점 1의 변수값들과 대단히 가까운 곳에 위치하게 되므로 전체행렬의 밴드폭을 현저히 줄일 수 있다.

2.4 변수 재배열

2.4.1 요소 재배열

절점, 변수 재배열을 위해 가장 먼저 할 일은 요소 재배열이다. 이웃하고 있는 요소 순으로 요소를 재배열한 후에 그 요소의 순서를 따라가면서 절점, 변수를 재배열해 주면 밴드폭이 작은 전체행렬 G를 얻을 수 있다.

요소 재배열에 필요한 입력자료는 각 요소에 이웃하고 있는 요소 번호이다. 본 연구에서는 사면체 격자를 사용하였는데 이 경우에는 어떤 요소가 주어졌을 때 그 요소와 면을 공유하는 요소들이 이웃하는 요소가 된다. 이 자료를 입력으로 하여 Cuthill-McKee 기법을 이용하여 요소를 재배열한다. 다음에 이를 바탕으로 하여 변수를 재배열하여야 한다. 이 연구에서 사용한 수치해석 방법은 P2/P1 사면체요소를 사용한 유한요소법이다. P2/P1 사면체 요소에서 속도는 사면체의 4개의

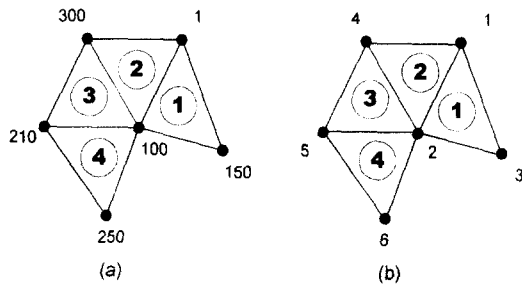


Fig. 4 Simple two-dimensional unstructured grid system : (a) before renumbering, (b) after renumbering

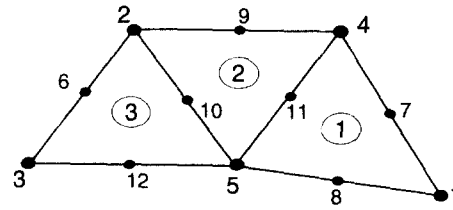


Fig. 5 Node and element numbering of a simple two-dimensional quadratic element mesh (Note that the number of a mid-node is larger than that of a vertex-node)

꼭지점 및 6개의 중점(각 꼭지점 사이의 중점)에서 정의되며 압력은 4개의 꼭지점에서만 정의된다.

2.4.2 절점 및 변수 재배열 과정의 개요

(1) 재배열된 요소 순서를 따라 각 요소 안의 6개의 중점을 순차적으로 배열하고 4개의 꼭지점을 중점 다음에 배열한다. (두 개 이상의 요소에 중복되어 있는 절점의 경우에는 그 절점이 속해 있는 요소 중 가장 큰 요소 번호에서 치환이 된다.) 중점 절점을 꼭지점보다 먼저 배열하는 이유는 불완전 LU분해 연산시 영 피벗이 나타날 경우를 피하기 위해서이다.(불완전 LU분해 연산시 영 피벗은 영 구역에서 생길 수 있으며 영 구역은 연속방정식에서 나오고 연속방정식의 가중합수는 압력변수와 같은 공간에 속함을 주목해야 한다.)

(2) 이렇게 절점들을 배열한 후에 각 절점에서 정의된 변수들을 모아준다.

- a. 중점 : 3개의 속도 성분(u, v, w)이 정의되어 있으며 이 3개의 변수들의 순서는 임의로 선택할 수 있다.
- b. 꼭지점 : 3개의 속도 성분 외에 압력이 추가로 정의되어 있고 u, v, w, p중에서 압력은 3개의 속도 변수들 다음에 배열해 준다.

(3) iperm, jperm 두 1차원 배열을 이용하여 아래와 같이 바뀐 절점 번호와 변수 배열 정보를 저장한다.

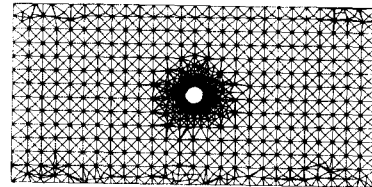


Fig. 6 Three-dimensional unstructured mesh around a sphere on x-z plane

$$jperm \text{ (재배열 후 절점번호)} = \text{원래 절점 번호}$$

$$iperm \text{ (원래 변수 번호)} = \text{재배열 후 변수번호}$$

2.4.3 변수 재배열 과정의 예

Fig. 5의 간단한 2차원 사면체요소중 ①번 요소의 변수가 재배열되는 과정을 기술하면 다음과 같다. 먼저 i 번째 요소의 유한요소 connectivity를 배열 icon(i,1~6)에 저장한다. 그러면, icon(1,1~6)는 다음과 같이 표시된다.

$$icon(1,1\sim 6) = (1, 4, 5, 7, 11, 8)$$

jperm 과 iperm 배열을 생성하기 위한 알고리즘은 다음과 같이 표시된다. 변수 nnode는 전체 절점의 수, Ne는 전체 요소의 수를 나타낸다.

Algorithm 1 : iwk 배열 생성

- 1 iwk = 0 ;초기화
- 2 For each element e_i, i=1,2,...,Ne Do:
- 3 For each node of e_i, j=1~6 Do:

```

4   iwkw(icon(i,j))=i
5   Enddo
6   Enddo

```

Algorithm II : iperm, jperm 배열 생성

```

1   jperm = 0, iperm = 0 ; 초기화
2   j0 = 0 ; jperm 배열을 위한 정수형 counter
3   i0 = 0 ; iperm 배열을 위한 정수형 counter
4   For each element ei, i=1,2,...,Ne Do:
5     For each mid-node of ei, icon(i,j), j=4~6 Do:
6       if iwkw(icon(i,j)) = i then;
7         j0 = j0+1
8         jperm(j0) = icon(i,j)
9         i0 = i0+1
10        ipermw(icon(i,j)) = i0 ; x 방향속도 성분
11        i0 = i0+1
12        ipermw(icon(i,j)+nnode) = i0 ; y 방향속도
13        i0 = i0+1
14      else; go to 5
15    Enddo
16    For each vertex-node of ei, icon(i,j), j=1~3
17    Do:
18      if iwkw(icon(i,j)) = i then;
19        j0 = j0+1
20        jperm(j0) = icon(i,j)
21        i0 = i0+1
22        ipermw(icon(i,j)) = i0 ; x 방향속도 성분
23        i0 = i0+1
24        ipermw(icon(i,j)+nnode) = i0 ; y 방향속도
25        ipermw(icon(i,j)+2*nnode) = i0 ; 압력변수
26      else; go to 16
27    Enddo
28  Enddo

```

위의 과정을 재배열된 요소 번호를 따라가면서 반복하면 아래와 같이 변수 배열순서가 바뀌게 된다.

a. 재배열 전

$$(u_1 \sim u_{12} : v_1 \sim v_{12} : p_1 \sim p_5)$$

b. 재배열 후

$$(u_7, v_7, u_8, v_8, u_1, v_1, p_1 \dots)$$

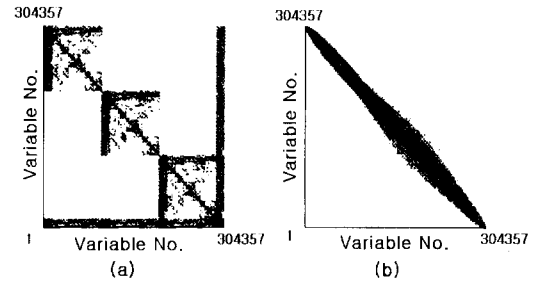


Fig. 7 Nonzero pattern of a globally assembled matrix for the flow around a sphere: (a) before variable reordering, (b) after variable reordering

2.5 변수재배열의 응용에

위에서 기술한 변수 재배열이 계산에 미치는 영향을 알아보기 위해서 3차원 구 주위의 유동과 MIRA 자동차 모델 주위 유동 계산에 적용해 보았다. 전체조립행렬의 해법을 위해 BiCGSTAB을 사용하였다.

2.5.1 구 주위의 유동계산

이 계산에 사용했던 격자계는 P2/P1 사면체요소로서 절점수, 요소수 그리고 총변수 수는 각각 97193, 69451, 304357이며 Fig. 6은 한 단면에서 격자분포를 보여준다. 앞서 기술한 변수 재배열이 전체행렬의 nonzero map에 미치는 영향을 확인하기 위해서 ICEM-CFD로부터 얻어진 초기 격자계와 재배열 후의 격자계의 전체행렬의 nonzero map을 Fig. 7에 제시하였다. Fig. 7을 통해서 변수 재배열 후의 전체행렬의 밴드폭이 재배열 전에 비해 대단히 많이 줄어들었음을 확인할 수 있다. 특히 Fig. 7(a)의 재배열 전의 전체행렬 G 를 보면 Fig. 1(a)와 같은 위화살표 형태의 행렬이 나타나 있음을 볼 수 있다.

구 주위 유동 계산은 $Re = 100$ 에서 $dt = 0.01$ 로 실행되었다. 이 계산에서 얻은 압력분포와 유선을 Fig. 8에 제시하였다. 그리고 구의 후류 중심선을 따른 압력의 변화를 Fig. 9에 제시하고 기존의 계산결과⁽¹⁵⁾와 비교하여 잘 일치함을 확인하

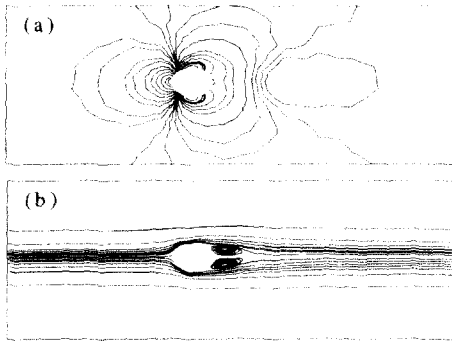


Fig. 8 (a) Pressure contour, (b) streamline around a sphere at $Re = 100$ (x-z plane)

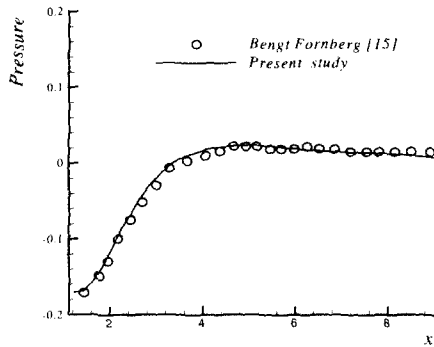


Fig. 9 Pressure distribution along the center line of the wake for the flow around a sphere at $Re = 100$

었다.

Fig. 10은 변수 재배열이 해의 수렴속도에 미치는 영향을 보여 준다. Fig. 10(a)에서 볼 수 있듯이 초기 시간 단계에서부터 두 경우의 수렴속도는 상당한 차이를 나타내며, 100번째 시간 단계 정도가 되면 Fig. 10(b)에서 확인할 수 있듯이 변수 재배열을 해 준 경우의 수렴속도가 그렇지 않은 경우의 속도에 비해 약 두 배 이상 향상됨을 확인할 수 있었다. 두 경우 모두 preconditioner로는 level(1) ILU preconditioner⁽¹³⁾를 사용하였다.

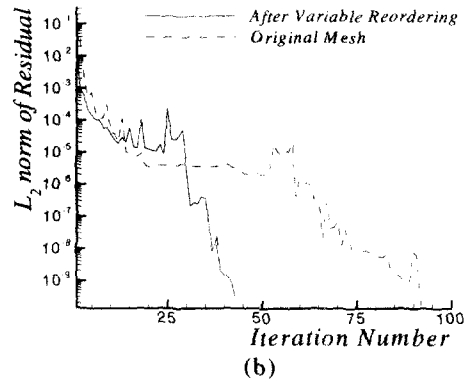
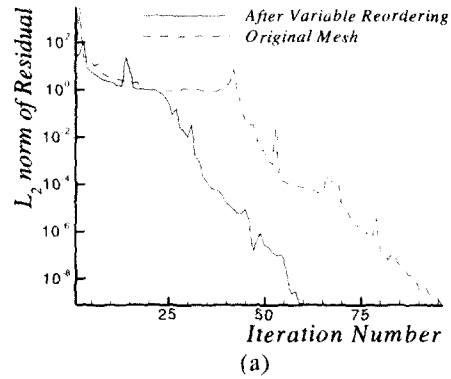


Fig. 10 Convergence history (sphere case at $Re = 100$) : (a) at 1st time step, (b) at 100th time step

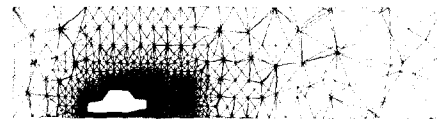


Fig. 11 Three-dimensional unstructured mesh around a MIRA model (x-z plane)

2.5.2 MIRA 자동차 모델 주위의 유동계산
다른 격자계에 대해서도 수렴속도 향상이 효과적으로 일어나는지 확인하기 위하여 Fig. 11에서 제

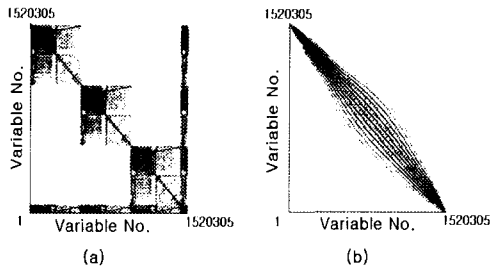


Fig. 12 Nonzero pattern of a globally assembled matrix for the flow around a MIRA: (a) before variable reordering, (b) after variable reordering



Fig. 13(a) Pressure contour around a MIRA model at Re = 260

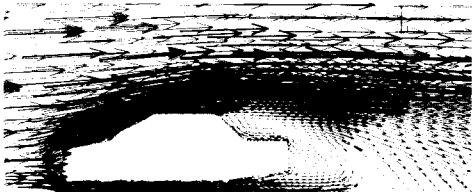


Fig. 13(b) Velocity field around a MIRA model at Re = 260

시된 MIRA 모델⁽¹⁶⁾ 주위의 격자계에 적용하였다. P2/P1 격자계의 요소수, 절점수, 변수수는 각각 347958, 485581, 1520305이다.

그리고 앞서 구의 경우와 마찬가지로 변수 재배열 전후의 전체행렬의 nonzero map을 Fig. 12에 제시하였다. 이 경우에도 구 주위의 격자계에서와 같이 변수 재배열 전의 전체행렬의 각 블록내에서 위화살표행렬과 유사한 형태가 나타나고 변수 재배열을 통하여 전체행렬의 밴드폭이 대단히 많이 줄어들었음을 확인할 수 있다. MIRA

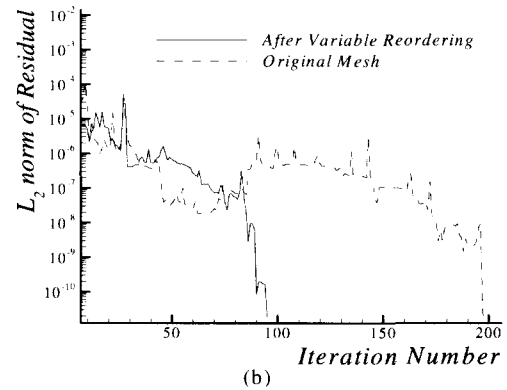
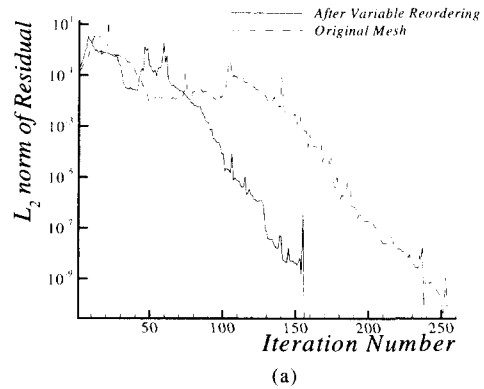
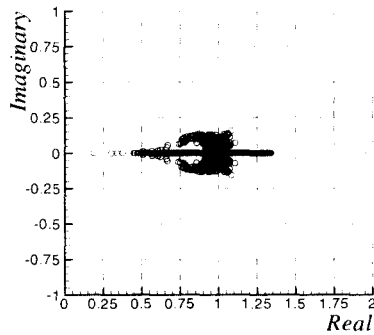


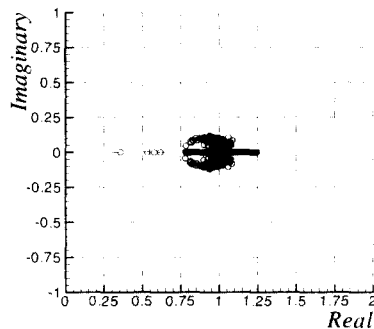
Fig. 14 Convergence history (MIRA case at Re = 260) : (a) 1st time step, (b) 100th time step

모델의 계산은 Re = 260에서 dt = 0.01로 실행되었으며, 비록 실제 유동장과는 많은 차이가 있으나 수치해석코드의 검증, 수렴성 확인 등에 유용하며, 또한 향후 실제적인 레이놀즈수에서의 연구 수행에 기초가 된다. 이 계산에서 얻은 압력과 속도분포를 Fig. 13에 제시하였다.

변수 재배열이 수렴속도에 미치는 영향을 확인하기 위해서 Fig. 14에 변수 재배열 전 후의 수렴이력곡선을 제시하였는데, 앞의 구의 경우와 비교적 흡사한 경향을 가지고 있음을 알 수 있다. 즉, Fig. 14(a)에서는 초기 시간단계에서부터 이미 재배열 한 경우의 수렴속도가 뚜렷이 증가함을 확인할 수 있고, Fig. 14(b)에서 보듯이 100



(a)



(b)

Fig. 15 Eigenvalue distribution of a globally assembled matrix for the flow around sphere at $Re = 100$: (a) before variable reordering, (b) after variable reordering (Number of variables = 6377)

Table 1 Effect of node reordering on condition number for the sphere case at $Re = 100$. (level(1) ILU preconditioner is used.)

	without reordering	with reordering
Dirichlet B.C.	149.31	87.67
Neumann B.C.	7.09	3.47

번째 시간단계에서는 수렴속도가 2배 이상 증가하였다. 예조건인자로는 구와 마찬가지로 level(1)

$$G = \begin{bmatrix} D & B \\ B^T & 0 \end{bmatrix} \quad P = \begin{bmatrix} D & B \\ B^T & B' \end{bmatrix}$$

(a) (b)

Fig. 16 Change of nonzero pattern of a globally assembled matrix and the corresponding preconditioning matrix: (a) Global matrix (b) Preconditioner

ILU preconditioner를 사용하였다.

2.6 변수 재배열 전 후의 고유치분포와 조건수의 변화

이 절에서는 변수 재배열 후 행렬의 고유치 분포와 조건수의 변화를 통하여 변수 재배열 과정이 전체행렬의 강성을 얼마나 감소시키는지 살펴보았다. 가장 이상적인 경우는 행렬의 고유치가 모두 (1,0)에 모여있는 경우다. 단위행렬이 이 예에 속한다. 즉 조건수가 (1,0)에 가까울수록 이상적인 행렬이다. 앞서 계산한 구와 MIRA 자동차 모델의 경우는 각각 변수가 304357개, 1520305개여서 전체행렬의 고유치를 구하려면 대단히 많은 메모리와 시간이 필요하므로 작은 절점을 가지는 3차원 구 주위의 격자계를 이용하였다. 이 경우 변수 수는 6377이다. 면 지역의 경계조건으로 Neumann 경계조건을 적용하고 예조건인자로는 level(1) ILU를 적용하였을 때, 변수 재배열 전 후의 고유치분포를 Fig. 15에 제시하였다. Fig. 15는 변수 재배열 전에 비해 변수 재배열 후의 고유치들이 복소평면 위의 (1, 0)인 점으로 모여 있음을 보여준다. 면 지역에 Dirichlet 경계조건도 적용하였는데 이 경우의 고유치분포 역시 Neumann 경계조건을 적용한 경우와 흡사하였다. 각 경우의 변수 재배열 전 후의 조건수를 Table 1에 제시하였다. Table 1에서 경계조건에 관계없이 변수 재배열이 전체행렬의 조건수를 상당히 감소시킴을 확인하였다. 즉 변수 재배열이 행렬의 강성을 줄여준다는 사실을 확인할 수 있었다.

3. 다양한 ILU type preconditioner의 제시 및 성능 검토

본 연구에서는 level(1) ILU preconditioner의

Table 2 Matrix size for various preconditioners of sphere case (Number of variable = 304357)

	영 구역	전체행렬
level(0)	12778 (1)	12236823 (1)
level(1)	181608 (14.21)	12236970 (1)
level(2)	765352 (59.50)	12820567 (1.04)

변형된 형태인 level(0) ILU preconditioner와 level(2) ILU preconditioner를 제안하였으며 각각의 성능을 비교해 보았다.

3.1 level(#) ILU preconditioner

2.2의 Fig. 3은 level(#) ILU preconditioner에서 기존의 ILU(0) 예조건인자와는 다르게 전체 행렬의 영 구역에 예조건인자로서 ILU 행렬 생성시 추가적인 nonzero 항들이 계산됨을 보여준다. 즉, 전체 조립된 행렬은 Fig. 16의 (a)에서처럼 영 구역이 존재하지만 level(#) ILU preconditioner의 nonzero 형태는 Fig. 16의 (b)에서처럼 B' 을 영이 아닌 구역으로 가지게 된다. 만약에 기존의 ILU(0) 예조건인자를 적용하면 ILU(0) 예조건인자의 nonzero pattern은 정의에 의해 전체조립행렬의 그것과 같게되어 B' 블록이 영 구역으로 된다. 따라서 ILU 연산을 영 피벗을 만나게 될 경우 더 이상 수행할 수 없게 된다. 따라서, 예조건인자의 B' 블록에 nonzero 항을 추가하는 것이 안정적 형태 행렬의 예조건인자를 생성하는 데 있어 필수적인 요소가 된다. B' 블록의 nonzero map을 NZ(B')로 정의하면 각각의 preconditioner는 NZ(B')의 형태에 따라 다음과 같이 정의된다.

(1) level(1) ILU preconditioner

- NZ(B') : B' 블록의 임의의 행에 대응하는 압력변수가 정의된 각 꼭지점에서, 그 절점을 공유하는 변들에 의하여 연결되어 있는 꼭지점들에서 정의된 압력변수들이 내재적으로 결합되어 있다고 가정하여 NZ(B')를 구성한다. 즉, 이 경우 NZ(B')는 SIMPLE 알고리즘과 같은 분리해법에서 생성되는 압력방정식과 같은 nonzero map을

가지게 된다.

(2) level(0) ILU preconditioner

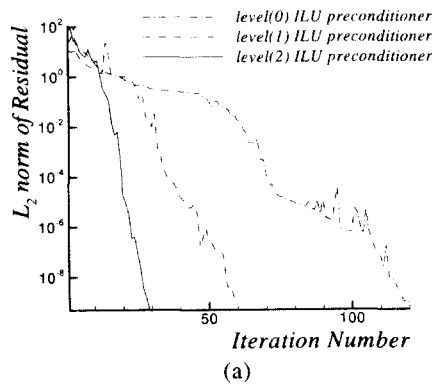
- NZ(B') : B' 블록의 대각선 항만 nonzero 항으로 계산된다. 이는 단순히 예조건인자의 ILU 연산시 영 피벗을 피하기 위한 수단으로 B' 블록만을 고려했을 경우 대각항 예조건인자와 유사하다.

(3) level(2) ILU preconditioner

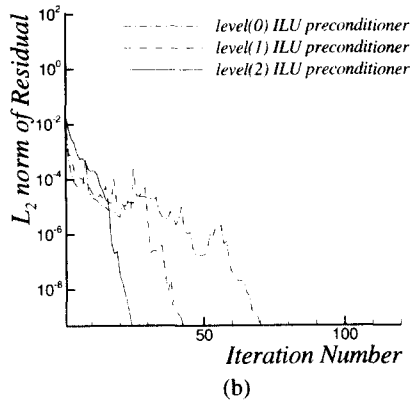
- NZ(B') : 이는 level(1)의 B' 블록을 확장한 것으로서, level(1)의 B' 블록의 주어진 행의 각 nonzero 항들에 대해 level(1)의 NZ(B')를 구성할 때와 같은 방식의 의해 내재적으로 결합되어 있는 압력변수를 추가하여 nonzero map을 구성한다. 이 경우 B' 블록의 크기는 level(1)의 그것보다 훨씬 커지게 되지만 P2P1 통합공식화의 특성에 의해서 B' 블록의 nonzero 항들의 수는 전체 예조건인자 행렬의 nonzero 항들의 수와 비교할 때 여전히 작은 부분을 차지한다.

한편, 예조건화기법에 따라 ILU 행렬의 B' block의 크기가 달라지기 때문에 예조건화행렬의 크기가 달라지게 된다. 예조건화를 위한 행렬의 크기가 크게 늘어나게 되면 예조건화때 연산횟수가 증가하게 된다. Table 2는 2절에서 구 주위 유동계산에 사용했던 격자계의 경우의 예조건화기법에 따른 전체행렬과 B' 블록의 크기를 보여준다.

Table 2에서 볼 수 있듯이 level(1) ILU preconditioner를 사용하게 되면 level(0) ILU preconditioner를 사용할 경우에 비해 제로 블록의 크기는 14.21배, 전체행렬의 크기는 1.000012배가 된다. 또한 level(2) ILU preconditioner를 사용한 경우를 level(0) ILU preconditioner를 사용한 경우와 비교해 보면 제로 블록의 크기는 59.89배, 전체행렬의 크기는 1.04배가 된다. 즉 Table 2를 통해 각 예조건화기법에 따라 제로 블록의 크기는 크게 증가하지만 전체 행렬의 크기는 거의 변화가 없음을 확인할 수 있다. 이는 전체행렬에서 제로 블록이 차지하는 비율이 대단히 낮기 때문이다. 따라서, level(2) preconditioner의 성능이 level(0)나 level(1)에 비해 매우 우수한데 비해 예조건화작업시 추가되는 연산의 비율은 대단히 미미



(a)



(b)

Fig. 17 Convergence history at $Re = 100$ (sphere case) : (a) 1st time step, (b) 100th time step

하므로 level(2) 예조건화는 매우 효율적인 방법이라 할 수 있다.

3.2 각 예조건화기법의 성능비교

실제 유동계산에 각 예조건화기법을 적용하여 그 성능을 비교해 보았다. 2장에서와 같이 구 주위의 유동과 MIRA 자동차 모델 주위의 유동계산에 적용해 보았으며 2장에서 사용했던 격자계를 사용하였다. 이 때 격자계는 모두 변수 재배열된 격자계를 사용하였다.

3.2.1 구 주위의 유동계산

구 주위의 유동계산시의 수렴이력곡선을 Fig.

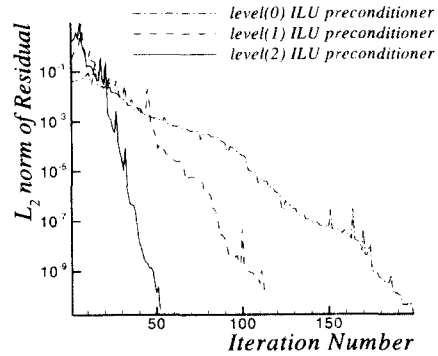


Fig. 18 Convergence history at $Re = 260$ (MIRA case) : 1st time step

17에 제시하였다. Fig. 17은 초기 시간 단계에서부터 각 예조건화기법에 따른 수렴속도의 차이가 뚜렷이 나타남을 보여준다. level(0) ILU preconditioner를 사용한 경우에 비해 level(1) ILU preconditioner를 사용하게 되면 수렴속도가 2배 가량 증가했으며 level(2) ILU preconditioner를 사용하게 되면 다시 수렴속도가 2배 가량 증가함을 확인할 수 있다. Fig. 17(b)의 100번째 시간 단계에서도 이런 경향은 거의 유사하게 나타남을 확인하였다.

3.2.2 MIRA 자동차 모델 주위의 유동계산

MIRA 자동차 모델 주위의 유동계산시의 수렴이력곡선을 Fig. 18에 제시하였다. Fig. 18은 MIRA 계산의 경우에도 앞서의 구 주위의 유동계산과 비슷한 수렴이력곡선을 보여준다. 이 경우에도 level(0) ILU preconditioner를 사용한 경우에 비해 level(1) ILU preconditioner를 사용한 경우 수렴속도가 2배 가량 증가하였으며, level(2) ILU preconditioner를 적용한 경우 다시 수렴속도가 2배 가량 증가하였다.

3.3 예조건화기법에 따른 고유치분포와 조건수의 변화

이 절에서는 각 예조건화기법이 행렬의 고유치분포와 조건수를 어떻게 변화시키는지 살펴보았다. 2장에서 사용했던 변수 개수 6377개인 격자계로 구 주위 유동을 계산하였으며 이 경우 역시 면 지역에 Neumann 경계조건과 Dirichlet 경계

Table 3 Effect of various preconditioners on condition number of preconditioned matrix for the sphere case at $Re = 100$

	Neumann B.C.	Dirichlet B.C.
without preconditioning	2135709	39250028
level(0)	15.04	149.31
level(1)	3.47	87.67
level(2)	1.59	17.23

조건을 적용하였을 때 조건수의 변화를 보았다.

Fig. 19는 Neumann 경계조건을 사용했을 때 고유치들의 분포를 보여준다. 여기서 Fig. 19(a), Fig. 19(b), Fig. 19(c), Fig. 19(d) 순으로 고유치들이 복소평면 상의 (1, 0)지점으로 모이고 있음을 확인할 수 있다. Table 3은 예조건화기법에 따른 조건수의 변화를 보여준다. Table 3에서 예조건화기법을 개선시킬수록 조건수가 (1,0)에 가까워짐을 확인할 수 있다. 특히 변수 재배열된 격자계에 level(2) ILU preconditioner 를 적용한 경우의 조건수는 1.59로 행렬의 성질이 대단히 좋아졌음을 확인할 수 있었다.

4. 결론

(1) 본 연구에서는 임의로 절점번호가 매겨진 3차원 비정렬격자계가 주어졌을 때 변수 재배열을 통해 이 격자로 인해 생기는 전체행렬의 밴드폭을 최적화시키고 변수 재배열이 수렴속도에 미치는 영향을 고찰하였다. 또한 level(#) ILU preconditioner들을 제안하고 각 예조건화기법의 성능을 비교해 보았다.

(2) 변수 재배열과 각 예조건화기법을 구 주위의 유동계산과 MIRA 자동차 모델 주위의 유동계산에 적용하였으며, 각 경우의 수렴이력곡선을 통해 수렴속도에 미치는 영향을 살펴보았다. 또한 작은 개수의 격자계로 구성된 표준예제에 변수 재배열기법과 각 예조건화기법을 적용하여

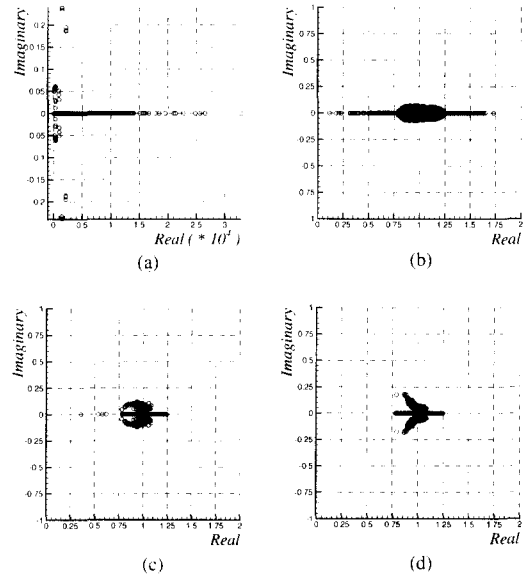


Fig. 19 Eigenvalue distribution by each preconditioner for the sphere case at $Re=100$: (a) without ILU preconditioning, (b) level(0), (c) level(1), (d) level(2) (Note that reordering is applied for all cases.)

고유치분포, 조건수를 살펴보았으며, 이를 통해 행렬의 강성이 어떻게 변하는지 확인하였다.

(3) 변수 재배열 후 수렴속도는 2배 가량 증가하였고 전체행렬의 고유치들도 (1,0)에 좀 더 가깝게 모이는 것을 확인하였다. Level(#) ILU preconditioner의 경우 level(0), level(1), level(2)의 순으로 수렴속도가 2배 정도씩 증가하였으며, 고유치들과 조건수도 (1,0)근처로 모임을 확인하였다.

(4) 본 연구에서는 임의의 비정렬격자계가 주어졌을 때 변수를 재배열 해주고 level(2) ILU preconditioner를 적용하게 되면 수렴속도가 대단히 빨라진다는 사실을 확인하였으며, 이는 대용량 계산등에서 많은 시간과 비용을 절약해 줄 것이다.

후 기

이 연구는 차세대자동차기술개발사업과 BK21

사업의 지원으로 수행되었으며, 논문교정을 도와 준 김범준에게 감사드립니다.

참고문헌

- (1) Meijerink, J. A. and Van der Vorst H. A., 1977, "An Iterative Solution Method for Linear Systems of Which the Coefficient Matrix is a Symmetric M -Matrix," *Mathematics of Computations*, Vol. 31, pp. 148~162.
- (2) Kershaw, D. S., 1978, "The Incomplete Cholesky-Conjugate Gradient Method for the Iterative Solution of Systems of Linear Equations," *Journal of Computational Physics*, Vol. 26, pp. 43-65.
- (3) Ferziger, J. H. and Peric, M., 1996, *Computational Methods for Fluid Dynamics*, Springer.
- (4) Chung, S. T., Choi, H. G. and Yoo, J. Y., 1998, "An Analysis of Turbulent Flow Around a NACA4421 Airfoil by Using a Segregated Finite Element Method," *KSME International Journal*, Vol. 12, pp. 1194~1199.
- (5) Patankar, S. V., 1980, *Numerical Heat Transfer and Fluid Flow*, McGraw-Hill, New York.
- (6) Chorin, A. J., 1968, "Numerical Solution of the Navier-Stokes Equations," *Mathematics of Computations*, Vol. 22, pp. 745~762.
- (7) Hanby, R. F., Silvester, D. J. and Chew, J. W., 1996, "A Comparison of Coupled and Segregated Iterative Solution Techniques for Incompressible Swirling Flow," *International Journal for Numerical Methods in Fluids*, Vol. 22, pp. 353~373.
- (8) Hu, H. H., Joseph, D. D. and Crochet, M. J., 1992, "Direct Simulation of Flows of Fluid-Particle Motions," *Theoretical Computational Fluid Dynamics*, Vol. 3, pp. 285~306.
- (9) Choi, H. G., 2000, "Splitting Method for the Combined Formulation of Fluid-Particle Problem," *Computer Methods in Applied Mechanics and Engineering*, Vol. 190, pp. 1367~1378.
- (10) Saad, Y., 1996, *Iterative Methods for Sparse Linear Systems*, PWS Publishing Company, pp. 72~76.
- (11) Carey, G. F. and Oden, J. T., 1986, *Finite Elements: Fluid Mechanics Vol. VI*, PRENTICE-HALL, Englewood Cliffs, New Jersey, Section 3.3.3.
- (12) Hughes, T. J. R., Franca, L. P. and Balestre, M., 1986, "A New Finite Element Formulation for Computational Fluid Dynamics: V. Circumventing the Babuska-Brezzi Condition: A Stable Petrov-Galerkin Formulation of Stokes Problems Accommodating Equal Order Interpolations," *Computer Methods in Applied Mechanics and Engineering*, Vol. 59, pp. 85~99.
- (13) Joseph, D. D. et al., 1998~2001, *Direct Simulation of the Motion of Particles in Flowing Liquids*, NSF KDI/New Computational Challenge.
- (14) van der Vorst, H. A., 1992, "Bi-CGSTAB: A Fast and Smoothly Converging Variant of Bi-CG for the Solution of Non-Symmetric Linear Systems," *SIAM Journal on Scientific and Statistical Computing*, Vol. 12, pp. 631~634.
- (15) Fornberg, B., 1988, "Steady Viscous Flow Past a Sphere at High Reynolds Numbers," *J. Fluid Mech.*, Vol. 190, pp. 471~489.
- (16) Anonymous, 1997, "Validation of CFD Codes for Predicting Aerodynamic Performance," *Automotive Engineer*, Vol. 17, pp. 46~49.