

**비압축성 Navier-Stokes 방정식의 수렴 가속을 위한 예조건화 Krylov
부공간법과 다중 격자법의 결합**
**(Combination of Preconditioned Krylov Subspace Methods and
Multi-grid Method for Convergence Acceleration of the
incompressible Navier-Stokes Equations)**

맹주성* (한양대학교), 최일곤 (한양 대학원), 임연우 (한양 대학원)

Joo Sung Maeng (HanYang Univ.), IL Kon Choi (Graduate School, HanYang Univ.)

Youn Woo Lim (Graduate School, HanYang Univ.).

Key Words : Multi-grid method(다중격자법), Krylov subspace method(Krylov부공간법), Cavity flow(공동 유동), Incompressible flow(비압축성 유동), Segregated algorithm(분리형 알고리즘), FAS(Full approximation storage), FMG(Full multi-grid method)

ABSTRACT : In this article, combination of the FAS-FMG multi-grid method and the Krylov subspace method was presented in solving two dimensional driven-cavity flows. Three algorithms of the Krylov subspace method, CG, CGSTAB(Bi-CG Stabilized) and GMRES method were tested with MILU preconditioner. As a smoother of the pressure correction equation, the MILU-CG is recommended rather than MILU-GMRES(k) or MILU-CGSTAB, since the MILU-GMRES(k) preconditioner has too much computation on the coarse grid compared to the MILU-CG one. As for the momentum equation, relatively cheap smoother like SIP solver may be sufficient.

1. 서론

현재 비압축성 Navier-Stokes 방정식의 해를 얻는 가장 대중적 방법은 압력기반 해법에 기초를 두고 있는 분리형 알고리즘(segreated algorithm)이다. 이 방법에서는 계수가 고도로 변하는 타원형 압력 또는 압력수정 방정식을 비선형 반복법에 의하여 풀게 된다. 그러나 실제로 압력 솔버는 강한 타원성 때문에 해를 얻는 과정에서 가장 많은 시간을 소비하는 부분이며 따라서 CPU 시간이라는 면에서 강력하고도 경쟁력 있는 솔버는 필수적이라고 할 수 있다. 이를 위한 연구 방법론에는 주로 두 가지를 생각할 수 있는데 즉, 다중격자법(multi-grid method)과 Krylov 부공간법(Krylov subspace method)이다. 일반적으로 선형 솔버는 격자가 커질 수록 반복수 증가로 인해 수렴이 급격하게 늦어진다. 일련의 계층격자를 사용하는 다중격자법은 성긴 격자와 밀한 격자를 이용함으로써 각각 다른 영역의 주파수를 갖는 오차를 없애으로써 단일 격자 경우보다 효과적 수렴율을 갖게된다. 따라서 점차 대규모 격자계에 대한 계산이 요구되는 지금 다중 격자법은 많은 연구자들의 관심을 집중시키게 되었다. 그러나 다중격자에 사용되는 Smoother가 열동한 경우 전체적인 다중격자의 성능을 악화시키기 때문에 smoother의 선택이 중요시된다. 따

라서 본 논문에서는 smoother로서 효과적인 Krylov 부공간법을 사용하여 다중격자의 전체적 성능에 미치는 영향을 알아보려는 것이다. 이것은 기존 다중격자법과 선형 솔버의 장점을 상호 보완하여 더욱 효과적인 수렴성을 얻으려는 시도이다. 본 연구자는 최근에 다중격자법 및 Krylov 부공간법에 대한 다양한 수치 실험을 행하여 성공적 결과를 얻은 바 있으며[1], 본 연구의 목적은 이에 대한 결과를 바탕으로 개발된 솔버를 직교 및 비직교 격자로 구성되는 몇 가지 예제 문제에 적용하여 다중격자법과 Krylov 부공간법을 조합한 해법을 개발하려는 것이다. 그러나 지금까지 두 방법의 결합에 관한 연구는 많지 않다. S. Zeng et al.[2]은 다중격자를 내부 루프에 사용한 GCR법과 종래의 SIP[참고]를 smoother로 한 다중 격자 등과 성능을 비교하였다. 또한 C.W. Oosterlee[3]는 3차원의 비등방성의 유체유동 문제에 대해 GMRES[8]를 smoother로 사용한 다중 격자를 사용하여 향상된 결과를 얻었다. 본 연구자는 [1]에서 제시한 효율적 Krylov 부공간법을 다중격자법의 smoother 후보로 선정하였다. 또한 이외에 가장 대중적으로 사용되는 방법중의 하나인 예조건화된 BICGTAB[13], SIP[14]등의 smoother등과도 그 성능을 비교함으로써 비압축성 N-S방정식에 가장 적합한 다중 격자를 얻고자 하였으며 정사각 공동(Square Cavity Flow)문제와 L-형 공동 문제



(L-shaped Cavity Flow)에 적용하여 그 성능을 비교 하였다.

2. 수치해석 과정

2.1. 비압축성 유동 해석 과정

방정식들을 구조화, 비직교 격자 상에서 원시 변수의 지배 방정식을 이산화하였는데 모든 셀 중심에 카테션 속도와 압력을 저장하였다. 유한 체 적법을(FVM) 사용하여 이산화하며 대류항은 Van Leer[4]의 κ 스킴으로 1차에 대한 지연수정(deferred correction)을 이용하여 구현하였다. 본 연구에서는 $\kappa = 1/3$ 인 CUI(cubic upwind interpolation)를 사용 하였으며 확산항은 2차 중앙차분법으로 이산화하였다. 압력장과 속도장은 SIMPLE 알고리즘[5]을 사용하여 연립시켰으며 Rhie와 Chow[6]의 내삽법을 사용하여 압력과 속도를 제어 체적의 중앙에 동치시킨 비엇갈림 격자제를 사용하였다.

2.2. 다중격자법(Multi-grid Method)

위에서 언급한 분리형 SIMPLE 알고리즘에서는 내부 변수간 연립과 속도-압력 같은 비연립 과정 내의 비선형성이 전체적인 수렴에 결정적이다. 대류가 지배적인 유동에 대해서는 압력 수정 방정식은 수렴이 느리다. 압력 수정법은 전형적인 평활 양상을 보여주는데 즉, 처음의 초기 몇 번의 수렴은 어느 정도 빠르지만 장파장 성분이 오차를 지배하게 되면 느려진다. 이러한 현상은 격자가 밀해지면 더욱 열악해진다. 비선형(외부) 반복의 수렴을 가속시키기 위해 비선형 연립방정식 시스템에 대하여 전근사기법(FAS)[1]을 사용한다. 다중 격자과정은 모든 격자수준에서 일정한 반복수를 가진 V-cycle에 기반을 둔 전다중격자법(FMG)[7]으로 구현한다. 먼저 가장 성긴 격자에서 해를 얻는다. 그리고 나서 이 해를 다음의 밀한 격자로 외삽하고 거기서 V-사이클을 다중 격자과정에 대한 초기 추정치(initial guess)로 사용한다. 단일 격자에 대하여 위에서 묘사한 해법을 모든 격자 수준에서 적용한다. 유일한 차이는 성긴 격자 방정식이 추가적인 원천항을 가지고 있다는 것이다. 이 과정을 가장 밀한 격자에서 수렴된 해에 이를 때까지 반복한다. 성긴 격자해를 얻기 위한 비용은 성긴 격자상의 모든 변수에 대한 양질의 초기 근사를 가지게 됨으로써 보상된다. 다중 격자 알고리즘은 격자점수에 무관한 수렴율을 보여준다. 이것은 보완적 관계를 가지는 방법을 구성함으로써 이루어지게 된다. 즉, 고주파 오차는 평활자(smoother)에 의해서 효과적으로 감소되는 반면 성긴 격자로 이루어진 계층을 적용한 성긴 격자 수정은 저주파 오차 성분을 제거한다. 그 과정은 아래와 같다. 다음과 같은 비선형 방정식을 고려해보자.

$$L(\phi) = S \tag{1}$$

격자 수준 $k(k=1, 2, \dots)$ 에서 완화스킴 (8)을 적

용하면 ϕ_k^* 가 얻어진다. 그러면 잔차는 다음과 같이 계산된다.

$$R_k = S_k^* - L(\phi_k^*) \tag{2}$$

여기에서 S_k^* 는 ϕ_k^* 를 사용하여 값을 정한다. 만약 $|R_k|$ 가 사전 수렴값 보다 적으면, '최종해'로 생각할 수 있으며 더 이상의 반복이 필요치 않다. 그렇지 않다면 수정량 ϕ_k^* 는 다음과 같은 방법으로 구한다.

$$L(\phi_k) = S_k + L(\phi_k^*) - S_k^* + R_k \tag{3}$$

방정식(2)와 (3)을 합하면

$$L(\phi_k) = S_k + L(\phi_k^*) - S_k^* + R_k^* \tag{4}$$

여기에서 전근사 (the full approximation) ϕ_k 는 다음으로 정의된다.

$$\phi_k = \phi_k^* + \phi_k^* \tag{5}$$

방정식(4)를 다음 성긴 격자 수준 ($k-1$)에서 푼다. 그리고 성긴 격자 방정식은 다음과 같이 구성된다.

$$L(\phi_{k-1}) = S_{k-1} + L(\hat{\phi}_{k-1} - \hat{S}_{k-1} + \hat{R}_{k-1}) \tag{6}$$

여기에서 $\hat{\phi}_{k-1}$ 와 \hat{R}_{k-1} 는 제한 보간된 변수이며 \hat{S}_{k-1} 는 $\hat{\phi}_{k-1}$ 로부터 구하는 $(k-1)$ 수준 원천항이다.

$$\hat{\phi}_{k-1} = I_k^{k-1} \phi_k^*, \quad \hat{R}_{k-1} = \hat{I}_k^{k-1} R_k \tag{7}$$

I_k^{k-1} 와 \hat{I}_k^{k-1} 증속변수와 잔차에 대한 제한 연산자를 나타낸다. (k 에 $k-1$ 로의) 방정식(6)은 이제 풀 수 있게 되며 그 해 ϕ_{k-1}^* 는 k 수준 격자로 연장보간된다.

$$\phi_k = \phi_k^* + I_{k-1}^k (\phi_{k-1}^* - \hat{\phi}_{k-1}) \tag{8}$$

여기에서 I_{k-1}^k 는 연장보간 연산자를 나타낸다. 방정식(6)에서 밀줄친 항은 완화층에 일정한 값을 유지한다.

2.3. 예조건화 Krylov 부공간법(Preconditioned Krylov Subspace Method)

지배 방정식을 이산화하면 모두 $Ax = b$ 와 선형 대수 방정식이 된다. Krylov 부공간이란 r_0 이 임의의 초기벡터 x_0 에 대한 초기 잔차일 때, $K_m(A, r_0) = \text{span}\{r_0, Ar_0, A^2r_0, \dots, A^{m-1}r_0\}$ 으로 주어지는 공간을 말한다. 근사해 x_m 은 변위된 공간(shifted space)인 $x_0 + K_m$ 위에 있으며, 이 때 잔차는 $b - Ax_m \perp l_m$ 이라는 Petrov-Galerkin 조건을 만족한다. 이러한 일련의 과정을 통해 해를 구하는 것을 Krylov 부공간법이라 한다. 이 방정식에 적용할 3 가지의 Krylov subspace법은 CG, CGSTAB(Bi-CG Stabilized) 그리고 GMRES(Generalized Minimal Residual)이다.[8] 그리고 CG법에 적용한 preconditioner는 다음의 5가지이다. Jacobi, SSOR (Symmetric Successive Over-Relaxation), IC(Incomplete Choleski Factorization), ILU (Incomplete Lower Upper Decomposition), MILU (Modified ILU)이다.

모든 Preconditioner는 계수 행렬 A 로부터 유도된다. Jacobi Preconditioner는 원래 A 행렬의 가중된 주대각 성분을 이용하여 만들어지며 식 (9)와 같다.

$$M = \frac{1}{\omega} D \tag{9}$$

단, 여기서 D 는 행렬 A 의 주대각 성분을 가리키며 ω 는 이완 계수를 나타낸다. 본 연구에서는 $\omega = 1.0$ 의 값을 사용하였는데 최적화된 값은 아니다. SSOR Preconditioner에서는 행렬 A 가 식 (10)과 같이 대각 성분인 D 와 하삼각행렬 L , 상삼각행렬 U 로 분해된다.

$$A = D + L + U \tag{10}$$

SSOR Preconditioner는 다음과 같다.

$$M = \left(\frac{1}{2-\omega} \right) \left(\frac{D}{\omega} + L \right) \left(\frac{D}{\omega} \right)^{-1} \left(\frac{D}{\omega} + L \right)^T \tag{11}$$

여기에서 사용한 상향 이완 인자는 ω 는 1.4이다. ILU Preconditioner는 행렬 A 의 불완전 분해를 한다고 가정한다.

$$A = M + R = (D + L)D^{-1}(D + U) + R \tag{12}$$

여기서 D, L, U 는 일종의 대각, 하삼각, 상삼각 행렬을 나타낸다. 그리고, R 은 fill-in 성분을 갖는 행렬인데, 불완전 분해를 할 때 무시된다. Preconditioner M 은 다음과 같다.

$$M = LDU \tag{13}$$

IC Preconditioner는 ILU 중에서 두 삼각행렬의 주대각 성분이 같다는 조건을 갖고 분해를 한 경우이다. 따라서 대각 행렬인 D 가 ILU Preconditioner와 다르다. 그리고 행렬 D 과 U 의 Non-zero 성분은 같은 위치의 행렬 A 의 성분과 같다. IC 분해법은 행렬 A 의 $a_{ij} = 0$ 인 성분과 대응하는 행렬 L 의 성분 $l_{ij} = 0$ 으로 한 뒤 분해된 행렬 L 의 성분을 계산한다. 여기에서 $a_{ij} \neq 0, l_{ij}$ 는 다음과 같이 구할 수 있다.

$$l_{ij} = \frac{\left(a_{ij} - \sum_{k=1}^{j-1} l_{ik}l_{jk} \right)}{l_{jj}} \quad i \neq j \tag{14}$$

$$l_{ii} = \left(a_{ii} - \sum_{k=1}^{i-1} l_{ik}^2 \right)^{1/2} \quad i = j \tag{15}$$

MILU Preconditioner의 경우는 ILU Preconditioner와 달리 행렬 L, U 의 Non-zero 성분이 같은 위치의 행렬 A 의 성분과 같지 않을 수도 있다. 대신 행렬 A 의 행의 합과 행렬 L, U 의 행의 합이 일치하도록 성분을 정한다.

$$\text{rowsum}(A) = \text{rowsum}(LDU) \tag{16}$$

또한 이는 대각행렬 D 의 성분을 달리 표현함으로써 MILU Preconditioner를 손쉽게 구현할 수 있다. 위에서 설명한 ILU와 IC, MILU의 대각행렬의 성분을 행렬 A 가 식(17)과 같을 때 예로 들어 표현해 보면 다음과 같다.

$$\text{entry}(A) = (A^S, A^W, A^P, A^E, A^M) \tag{17}$$

$$\text{entry}(L) = \left(A^S, A^W, \frac{1}{D^P}, 0, 0 \right) \tag{18}$$

$$\text{entry}(D) = (0, 0, D^P, 0, 0) \tag{19}$$

$$\text{entry}(U) = \left(0, 0, \frac{1}{D^P}, A^E, A^M \right) \tag{20}$$

$$D_{i,j}^P = 1 / \left(A_{i,j}^P - D_{i-1,j}^P A_{i,j}^W (A_{i-1,j}^E + \alpha A_{i-1,j}^N) - D_{i,j-1}^P A_{i,j}^S (A_{i,j-1}^N + \alpha A_{i,j-1}^E) \right) \tag{21}$$

이때 α 가 0이면 ILU Preconditioner가 되고 α 가 1이면 MILU Preconditioner가 된다. 본 연구에서는 MILU의 경우에 $\alpha = 0.99$ 를 사용하였다. IC Preconditioner의 경우 다음과 같다.

$$D_{i,j}^P = \frac{1}{A_{i,j}^P - D_{i-1,j}^P [A_{i,j}^W]^2 - D_{i,j-1}^P [A_{i,j}^S]^2} \tag{22}$$

이상의 모든 Preconditioner는 대칭과 비대칭

행렬 모두에게 사용할 수 있다. 특히 대칭행렬인 경우 대칭 조건인 $L = U^T$ 로 간주할 수 있다.

3. 결과 및 고찰

위에서 상술한 다중격자법의 Smoother로서 적합한 Krylov 부공간법을 알아보기 위하여 먼저 정사각 상판 끌림 공동(Square lid-driven cavity)에 적용하였으며 레이놀즈 수는 5,000이다. 사용한 제어체의 수는 128×128 이다. 수렴 판정은 총 잔차를 $R = (b - A\phi)$ 로 표시했을 때 모든 격자 점에서의 절대 잔차의 합을 그 초기값으로 정규화(normalization)한 것이 10^{-5} 보다 적으면 수렴한 것으로 정의하였다. 본 연구에서의 모든 결과는 Intel사의 Pentium 166 MHz Processor를 갖는 개인용 컴퓨터로 수행되었다. 먼저 압력수정 방정식을 푸는 데 적합한 솔버를 구하기 위해 각 경우에 대하여 공통적으로 운동량 방정식에는 SIP를 이용하고 압력 수정 방정식에는 각각 SIP, MILU-CG, MILU-CGSTAB, MILU-GMRES(k)를 이용하여 계산하였다. 그 이유는 본 연구자의 기존 연구[1]에서 밝혀진 바와 같이 MILU-CG와 MILU-GMRES가 좋은 성능을 보였기 때문이다. Table 1에서 보듯이 압력 수정 방정식에 SIP솔버를 사용한 결과와 비교하였을 때, MILU-CG 결과는 최대 1.4배정도 연산시간을 단축하였다. 그러나 MILU-CGSTAB과 MILU-GMRES 결과는 전체적으로 많은 연산 시간을 소비하였다. 다중 격자는 그 알고리즘의 특성상 성긴 격자에서의 연산이 필수적이다. 그러나 성긴 격자에서는 그 계산량에 비하여 상대적으로 자체 연산량이 많은 솔버를 사용할 경우에 SIP보다 전체적으로 연산시간의 증가를 가져왔다. 특히 이 점은 GMRES(k)는 메모리 사용량 및 연산 시간이 차원수(dimension)에 비례하기 때문에 k가 2에서 5로 증가한 경우 그로 인한 수렴율의 가속효과에 비하여 연산량의 증가가 더욱 커서 전체적인 연산 시간이 커짐을 알 수 있다. 따라서 압력수정 방정식을 계산할 때 일반적인 단일 격자계에서 좋은 수렴 율을 보인 MILU-CG법이 다중 격자에서도 smoother로서 좋은 수렴 양상을 보임을 알 수 있다. 두 번째로 운동량 방정식을 푸는 데 적합한 솔버를 선정하기 위해 각 경우에 대하여 압력수정 방정식에는 MILU-CG를 공통적으로 이용하고, 운동량 방정식은 SIP, MILU-CGSTAB, MILU-GMRES(k)를 이용하여 계산하였다. Table 2에서 나타난 바와 같이 기존의 SIP가 다른 MILU-CGSTAB, MILU-GMRES(1)보다 더 빠른 수렴 속도를 보인다. 이는 압력 수정 방정식과 달리 운동량방정식은 상대적으로 적은 내부 반복으로 쉽게 정해진 수준의 잔차감소를 이루기 때문에 자체 연산량이 적은 SIP가 다른 MILU-CGSTAB, MILU-GMRES(1)보다 운동량 방정식을 계산하기에 적합하다는 것을 알 수 있다. 따라서 Table 1과 2를 보면 기존의 운동량방정식/압력수정 방정식을

Table 1. Comparison of Performance of solvers for Pressure Correction Eq.

Pressure Correction Equation	Computation Time(sec)
SIP	682.23
MILU-CG	486.97
MILU-CGSTAB	506.79
MILU-GMRES(2)	670.00
MILU-GMRES(5)	803.89

Table 2. Comparison of Performance of solvers for Momentum Equation

Momentum Equation	Computation Time(sec)
SIP	486.97
MILU-CGSTAB	543.10
MILU-GMRES(1)	563.00

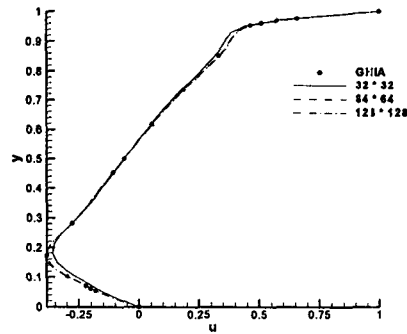


Fig. 1 Comparison of u-velocity along vertical lines in Square cavity (Re =1000)

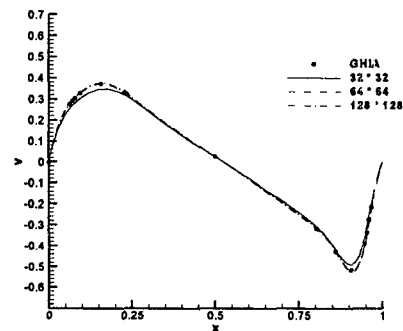


Fig. 2 Comparison of v-velocity along horizontal lines in Square cavity (Re =1000)

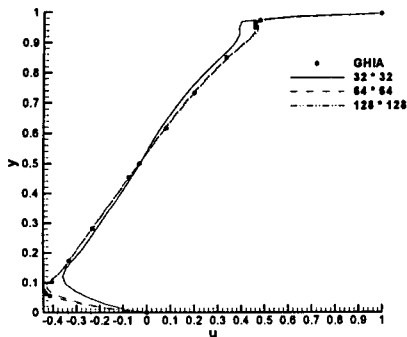


Fig. 3 Comparison of u-velocity along vertical lines in Square cavity (Re =5000)

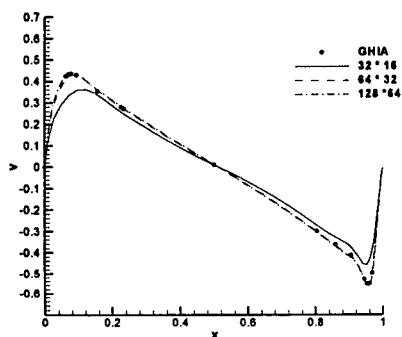


Fig. 4 Comparison of v-velocity along horizontal lines in Square cavity (Re=5000)

SIP/SIP의 조합보다는 SIP/MILU-CG의 조합을 이용하여 계산하는 것이 더 좋을 것으로 예상된다. Fig.1 ~ Fig.4는 레이놀즈 수 1000과 5000일 때의 정사각 공동 내에서 중앙 수직선과 수평선에서의 속도 u , v 의 단면을 나타낸 그림이다. 32×32 , 64×64 , 128×128 의 격자계에 대한 연산을 수행하였다. Fig. 1 ~ Fig. 4에서 보는 바와 같이 32×32 인 격자계를 제외하고는 다른 경우에는 각각 [9]의 결과와 매우 정확히 일치한다.

Table 3. Performance in Square Cavity (Re=1,000)

Grid Size	Computation Time(sec)	
	SIP/SIP	SIP/MILU-CG
32×32	23.23	22.47
64×64	77.17	73.32
128×128	211.79	212.91

Table 4. Performance in Square Cavity (Re=5,000)

Grid Size	Computation Time(sec)	
	SIP/SIP	SIP/MILU-CG
32×32	70.75	64.65
64×64	318.52	265.62
128×128	682.23	486.97

Table 3과 4는 레이놀즈 수에 따른 각 솔버 조합의 성능을 나타낸 것이다. Table 3에서 두 솔버에 의한 계산시간은 거의 비슷하지만 Table 4에서 레이놀즈 수 5000인 때는 SIP/MILU-CG에 의해 시간이 최고 1.4배정도 단축된다. 이를 보면 레이놀즈 수와 격자 수가 커질수록 SIP/MILU-CG의 성능이 우수함을 알 수 있다. Fig.5의 L-형 공동 유동 문제는 좌표계 곡률이 심한 경우에 대한 결과를 보기 위한 벤치 마크 문제이다.[10] 격자의 비직교성은 필연적으로 압력 수정 방식의 수렴을 악화시키는 요인이 되므로 수렴율의 저하를 가져온다. Fig.6은 [12]의 방법을 사용한 타원형 격자 생성법에 의하여 생성된 격자를 나타낸 것이다. Γ_2 의 상단면에는 수평 방향속도(u)를 크기 1로 하고 다른 부분은 모두 0이다.

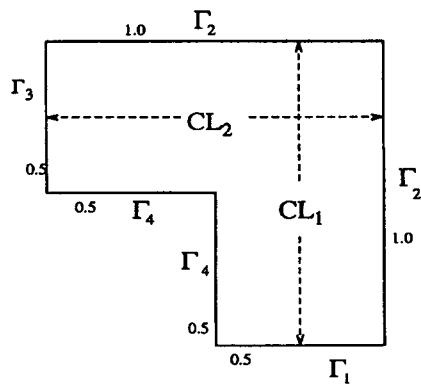


Fig.5 The Domain for a L-shaped cavity

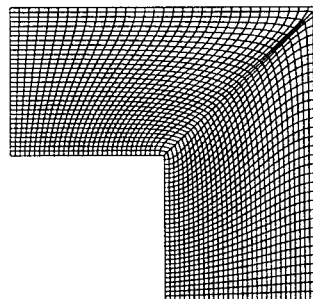


Fig. 6 A smooth grid in L-shaped channel

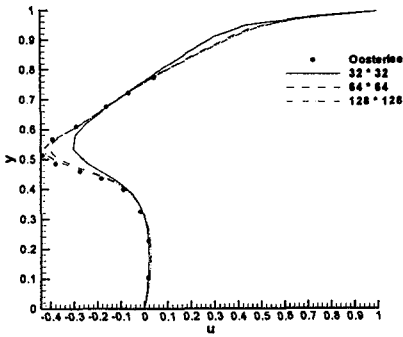


Fig. 7 Comparison of u-velocity along CL_1 in L- shaped cavity

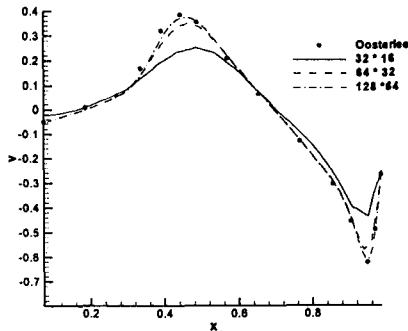


Fig. 8 Comparison of v-velocity along CL_2 in L- shaped cavity

Fig. 7과 Fig. 8은 각각 CL_1 상의 속도 u 의 단면과 CL_2 상의 v 의 단면을 나타낸 그림이다. 정사각 공동 유동 문제와 마찬가지로 32×32 인 격자계를 제외하고는 다른 경우에서는 각각 [10]의 결과와 매우 정확히 일치한다. Table 5, 6은 레이놀즈수가 각각 1000, 5000일 때 각 솔버 조합의 성능을 나타낸 것이다.

Table 5. Performance in L-shaped Cavity (Re=1,000)

Grid Size	Computation Time(sec)	
	SIP/SIP	SIP/MILU-CG
32×16	9.22	9.94
64×32	44.38	44.66
128×64	106.02	106.43

Table 6. Performance in L-shaped Cavity (Re=5,000)

Grid Size	Computation Time(sec)	
	SIP/SIP	SIP/MILU-CG
32×16	14.28	14.38
64×32	144.01	139.73
128×64	490.82	481.73

Table 5에서 레이놀즈수가 1,000일 때는 두 솔버의 경우 계산시간이 거의 비슷하였으나, 레이놀즈수가 5,000인 경우 Table 6에서 보면 SIP/MILU-CG를 이용한 경우 계산시간이 단축된다. 계산 영역의 곡률이 심한 경우에도 SIP/MILU-CG는 SIP/SIP에 비해 같거나 더 좋은 성능을 보인다는 것을 알 수 있었으며 레이놀즈수가 갈수록 두 솔버에 의한 계산시간의 차이가 커짐을 알 수 있었다.

4. 결론

본 연구에서는 비압축성 유동 모사에 있어 다중격자법과 Krylov subspace법을 효과적으로 결합하여 기존의 다중격자법의 성능을 향상시키고자 하였다. 운동량 방정식에는 MILU-CGSTAB, MILU-GMRES를, 압력 수정 방정식에는 MILU-CG, MILU-CGSTAB, MILU-GMRES를 적용하여 두 방정식 모두 SIP로 계산한 솔버와 비교하였다. 다중격자법에 Krylov 부공간법을 적용할 때는 솔버 자체의 수렴율뿐만 아니라 연산량이 전체의 성능에 더욱 중요하다는 것을 알 수 있었으며 다음과 같은 결론을 도출할 수 있었다.

- (1) 압력 수정 방정식의 smoother로서 MILU-CG에 비하여 MILU-GMRES(k)는 그 자체의 큰 연산량으로 인해 적합하지 않음을 알았다.
- (2) 압력 수정 방정식에 MILU-CG를 사용한 경우가 본 연구에서 고려한 것 중 가장 탁월한 것으로 판명되었다.
- (3) 압력 수정방정식을 MILU-CG로 풀 경우 격자에 대한 의존도의 감소로 인해 대규모의 문제에 적합함을 알 수 있다.
- (4) 운동량 방정식은 상대적으로 적은 내부 반복으로 정해진 수준의 잔차감소를 이루기 때문에 자체 연산량이 적은 SIP가 다른 MILU-CGSTAB, MILU-GMRES보다 운동량 방정식을 계산하기에 적합함이 판명되었다.

참고문헌

1. 맹주성, 임연우, 최일곤, “비압축성 Navier-Stokes 방정식에 대한 Krylov Subspace법의 적용”, 대한 기계학회 춘계학술대회 논문집 B,



pp.309-314, 1999

2. S. Zeng, C. VuikB., and P. Wesseling, "Further Investigation on the Solution of the Incompressible Navier-Stokes Equations by Krylov Subspace and Multigrid Methods", Report 93-93, Faculty of Technical Mathematics and Informatics, 1993, TU Delft, The Netherlands
3. C. W. Oosterlee, "A GMRES-based plane smoother in multigrid to solve 3D anisotropic fluid flow problems", J. Comp. Phys., Vol. 130, 1997, pp.41-53
4. B. Van-Leer, "Towards the Ultimate Conservative Difference Scheme; A Second-order Sequel to Gudnov's Method", J. Comp. Phys., Vol. 32, 1979, pp.101-136
5. S.V. Patankar, Numerical Heat and Fluid Flow, 1980, Hemisphere, New York
6. C.M. Rhie and W.L. Chow, "A Numerical Study of the Turbulent Flow Past an Isolated Airfoil with Trailing Edge Separation", AIAA J., Vol. 21, 1983, pp. 1525-1532
7. Z. Lilek, S. Muzaferiza, and M., Peric, "Efficiency and Accuracy Aspects of a Full-Multigrid SIMPLE Algorithm for Three-Dimensional Flows", Numerical Heat Transfer, Part B, Vol.31, 1997, pp.23-42
8. Y. Saad and M.H. Schultz, "A Generalized Minimal Residual Algorithm for Solving Nonsymmetric Linear Systems" SIAM J. Sci. Stat. Comput., Vol.7, 1986, pp.856-869
9. U. GHIA K. N. GHIA and C. T.Shin,"High-Resolutions for incompressible flow using the Navier-Stokes equations and a multigrid" J. Comp. Phys., Vol. 48, 1982, pp.387-411
10. C.W. Oosterlee and P. Wesseling, "Benchmark solution for the incompressible Navier-Stokes equations in general coordinates on staggered grids" Report 92-67, Faculty of Technical Mathematics and Informatics, 1992, TU Delft, The Netherlands
11. Shyy, W. "Computational modelling for fluid flow and interfacial transport" Elsevier, pp.184-196
12. 최일곤, 맹주성, "자동화된 경계 고정 좌표 생성법과 자연 대류 문제에 대한 적용", 1999, 6 대한 기계학회(계제예정)
13. Barret, R., Berry, M., Chan, T., Demmel, J., et al., 1993, Templates for the Solution of Linear Systems: Building Blocks for the Iterative Methods, SIAM Publications.
14. Stone, H.L., "Iterative solution of implicit approximations of multi-dimensional equations", SIAM J. Numerical Analysis, 5, pp530-558