

3차원 일반 좌표계에서의 PISO, SIMPLE, SIMPLE-C 알고리즘의 비교

박준영*¹, 백제현*²

A Comparative Study of PISO, SIMPLE, SIMPLE-C Algorithms in 3-dimensional Generalized Coordinate Systems

J. Y. Park and J. H. Baek

The performance of the SIMPLE, SIMPLE-C and PISO algorithms for the treatment of the pressure-velocity coupling in fluid flow problems were examined by comparing the computational effort required to obtain the same level of the convergence. Example problems are circular duct and 90-degree bent square-duct. For circular duct case, laminar and turbulent flow were computed. For 90-degree bent square-duct case, laminar flow was simulated by the time-marching method as well as the iterative method. The convergence speed of the other two algorithms are not always superior to SIMPLE algorithm. SIMPLE algorithm is faster than SIMPLE-C algorithm in the simple laminar flow calculations. The application of the PISO algorithm in three dimensional general coordinates is not so effective as in two-dimensional ones. Since computational time of PISO algorithm is increased at each time step(or iterative step) in three dimension, the total convergence speed is not decreased. But PISO algorithm is stable for large time step by using time marching method.

1.1 서론

기호설명

A : link coefficient of finite difference equation
J : Jacobian of metric transformation

하첨자

E,W,N,S,T,B : position of cell boundary
i, j, k : Cartesian coordinate directions

비압축성 유동장의 수치 해석에 SIMPLE(semi-implicit method for the pressure linked equations) 알고리즘이 Pantanka와 Spalding(1972)에 의해 제안되었던 이래 많은 유동 문제의 해법에 폭넓게 사용되어져 왔다. 그후 SIMPLE 알고리즘의 수정된 형태로 SIMPLER(SIMPLE revised)[1]과 SIMPLE-C(SIMPLE consistent)[2] 알고리즘이 있다. 이러한 SIMPLE 계열 알고리즘들은 수렴 성이 항상 보장되는 것이 아니며, 특별한 under-relaxation 값을 요구하는 단점이 있다.

이후 이러한 SIMPLE 계열 알고리즘과는 다른

*¹ 학생 회원, 포항공과대학교 기계공학과

*² 정회원, 포항공과대학교 기계공학과

PISO(pressure-implicit with splitting equation) [3]알고리즘이 ISSA에 의하여 제안되었다. PISO 알고리즘은 각 시간 단계에서 구한 해가 차분화 운동량 방정식과 압력 보정 방정식의 정확한 해의 근사한 값이 되도록 차분화된 방정식의 해를 연속된 분할 연산에 의해 계산한다. 이러한 분할 (splitting)절차에 의해 전체적인 구조의 안정성에 손상을 주지 않고, 내재적(implicit)차분화의 이점을 유지하면서 반복 계산을 제거할 수 있다. 또한 비록 PISO알고리즘이 비정상(time-depend ent)형태의 식으로부터 만들어졌으나 상당히 큰 시간 간격에 대하여도 알고리즘의 안정성이 유지되므로 정상 상태(steady state)의 계산에도 유용하다. 그리하여 Jang[4]에 의해 제시된 것처럼 PISO알고리즘을 이용하여 정상 상태의 해를 구하는 방법으로는 SIMPLE계열의 알고리즘에 적용되는 방법들처럼 시간 전진법(time marchin g)과 반복 계산법(iteration method)이 있다. SI MPLE계열의 알고리즘과 PISO알고리즘의 공통 점은 각 단계에서의 압력장과 속도장에 대하여 연속 방정식과 운동량 방정식을 만족시키도록 가장 최근의 값으로 계속해서 바꾸어 주는 것이다. 이러한 특징들을 가지는 PISO알고리즘과 SI MPLE계열 알고리즘들의 비교는 주로 2차원 유 동과 축대칭 유동에 대하여 이루어 졌으며 3차 원 유동에 대하여는 비교된 바가 없다. 따라서 본 연구에서는 PISO알고리즘을 일반 좌표계에서 의 Navier-Stokes방정식에 적용하여 SIMPLE계 열 알고리즘의 수렴 속도와 비교하고자 한다.

2. 수치 해석

2.1 지배 방정식

비압축성, Newtonian 유동의 연속 방정식과 운동 방정식은 cartesian좌표계(x,y,z)에서 다음과 같다.

$$U_t + E_x + F_y + G_z = S \quad (1)$$

$$U = \begin{bmatrix} \rho \\ \rho u \\ \rho v \\ \rho w \end{bmatrix} \quad E = \begin{bmatrix} \rho u \\ \rho u u - \mu u_x \\ \rho u v - \mu v_x \\ \rho u w - \mu w_x \end{bmatrix}$$

$$F = \begin{bmatrix} \rho v \\ \rho v u - \mu u_y \\ \rho v v - \mu v_y \\ \rho v w - \mu w_y \end{bmatrix} \quad G = \begin{bmatrix} \rho w \\ \rho w u - \mu u_z \\ \rho w v - \mu v_z \\ \rho w w - \mu w_z \end{bmatrix}$$

$$S = \begin{bmatrix} (\mu u_x)_x + (\mu v_x)_y + (\mu w_x)_z - P_x \\ (\mu u_y)_x + (\mu v_y)_y + (\mu w_y)_z - P_y \\ (\mu u_z)_x + (\mu v_z)_y + (\mu w_z)_z - P_z \end{bmatrix}$$

위 지배 방정식을 일반 좌표계상으로 좌표변 환 시킨 후 hybrid scheme과 엇갈림 격자계를 사용하여 차분화하면 다음과 같은 차분 방정식을 얻을 수 있다.

$$A_P \phi_P = A_E \phi_E + A_W \phi_W + A_N \phi_N + A_S \phi_S + A_T \phi_T + A_B \phi_B + S \quad (2)$$

이때 각 격자점의 상대적 호칭과 엇갈림 격자계 는 Fig. 1과 Fig. 2에 나타나 있다.

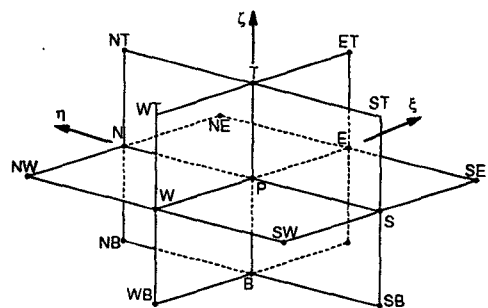


Fig. 1 Three-dimensional grid structure and labeling around a grid node P

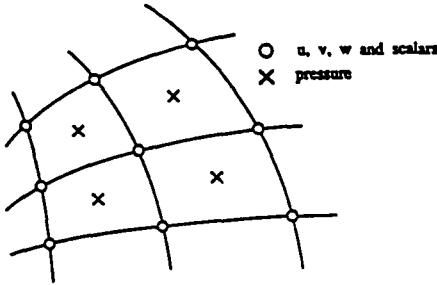


Fig. 2 Locations where the variables are stored(staggering grids are used)

2.2 수치 해석 알고리즘

2.2.1 SIMPLE 알고리즘

식(2)의 생성항에서 압력 변화율을 분리하여 운동방정식에 적용하면 다음과 같다.

$$\begin{aligned} A_P u_P &= \sum A_{nb} u_{nb} - P_x J + S_u \\ A_P v_P &= \sum A_{nb} v_{nb} - P_y J + S_v \\ A_P w_P &= \sum A_{nb} w_{nb} - P_z J + S_w \end{aligned} \quad (3)$$

또한 가정된 압력 P^* 와 이로부터 구해지는 부정확한 유속 u^*, v^*, w^* 와의 관계는 다음과 같다.

$$\begin{aligned} A_P u_P^* &= \sum A_{nb} u_{nb}^* - P_x^* J + S_u \\ A_P v_P^* &= \sum A_{nb} v_{nb}^* - P_y^* J + S_v \\ A_P w_P^* &= \sum A_{nb} w_{nb}^* - P_z^* J + S_w \end{aligned} \quad (4)$$

이때 유속과 압력의 정확한 값과 부정확한 값의 관계는 보정치(u', v', w', P')를 이용하여

$$\begin{aligned} u &= u^* + u' \\ v &= v^* + v' \\ w &= w^* + w' \\ P &= P^* + P' \end{aligned}$$

가되므로 식(3)에서 식(4)를 빼면,

$$\begin{aligned} u &= u^* + u' \\ v &= v^* + v' \\ w &= w^* + w' \\ P &= P^* + P' \end{aligned}$$

가되므로 식(3)에서 식(4)를 빼면,

$$\begin{aligned} A_P u'_P &= \sum A_{nb} u'_{nb} - P'_x J \\ A_P v'_P &= \sum A_{nb} v'_{nb} - P'_y J \\ A_P w'_P &= \sum A_{nb} w'_{nb} - P'_z J \end{aligned} \quad (5)$$

가된다. 정리하면 아래와 같다.

$$\begin{aligned} u'_P &= -DP'_x \\ v'_P &= -DP'_y \\ w'_P &= -DP'_z \\ D &= \frac{J}{A_P} \end{aligned} \quad (6)$$

따라서 아래의 식이 도출된다.

$$\begin{aligned} \rho u &= \rho u^* + \rho u' = \rho u^* - DP'_x \\ \rho v &= \rho v^* + \rho v' = \rho v^* - DP'_y \\ \rho w &= \rho w^* + \rho w' = \rho w^* - DP'_z \end{aligned}$$

윗식을 연속 방정식에 대입하여 정리하면 아래와 같은 최종 압력 보정 방정식을 얻을 수 있다.

$$A_P P'_P = A_E P'_E + A_W P'_W + A_N P'_N + A_S P'_S + A_{\text{GB}}$$

앞서 설명한 바에서 알 수 있듯이, SIMPLE 알고리즘의 수치해석 순서는 다음과 같다.

1. 속도장과 압력장 등의 초기값을 가정한다.
2. 운동방정식에서 속도장을 계산한다.
3. 기타 필요한 여러 방정식(에너지, 난류)을 계산한다.
4. 압력 보정방정식을 계산한다.
5. 압력장과 속도장을 보정한다.
6. 수렴된 해를 얻을 때까지 (2)-(5)의 과정을 반복한다.

2.2.2 SIMPLE-C 알고리즘

SIMPLE-C 알고리즘은 SIMPLE 알고리즘과 동일한 순서를 따른다. 두 알고리즘에 있어서 주요한 차이는 속도 보정 방정식의 형태에 있다. 즉

식(5)의 양변에 $\Sigma A_{nb}u'_{nb}$ 를 빼주면 다음과 같다.

$$\begin{aligned} (A_P - \Sigma A_{nb})u'_P &= \Sigma A_{nb}(u'_{nb} - u'_P) - P'_x J \\ (A_P - \Sigma A_{nb})v'_P &= \Sigma A_{nb}(v'_{nb} - v'_P) - P'_y J \\ (A_P - \Sigma A_{nb})w'_P &= \Sigma A_{nb}(w'_{nb} - w'_P) - P'_z J \end{aligned}$$

오른쪽 첫째항을 무시하고 정리하면 다음과 같다.

$$D = \frac{I}{A_P - \Sigma A_{nb}}$$

2.2.3 PISO알고리즘

Issa에 의하여 제안되어진 PISO알고리즘은 각각의 시간 간격 동안에 하나의 예측 단계(predictor step)와 하나 또는 그 이상의 수정 단계(corrector step)가 있는 시간 전진법(time-marching)이다. 부호의 편의를 위하여 이전 단계의 수렴에는 첨자 n으로 나타내며, 현 단계에서 예측 단계에서의 해는 첨자 *로, 첫번째 수정단계에서의 해는 첨자 **로 나타낸다. 또, 두 번째 수정 단계에서의 수렴해는 ***로 나타낸다.

1) Predictor Step

n번째 계산 단계에서의 압력 P^n 이 u^*, v^*, w^* 를 구하는 내재적(implicit)운동량 방정식을 계산하는데 사용되어진다.

$$\begin{aligned} A_P u^*_P &= \Sigma A_{nb} u^*_{nb} - P^n_x J + S_u \\ A_P v^*_P &= \Sigma A_{nb} v^*_{nb} - P^n_y J + S_v \\ A_P w^*_P &= \Sigma A_{nb} w^*_{nb} - P^n_z J + S_w \end{aligned} \quad (10)$$

여기에서 구해진 속도장은 운동량 방정식은 만족하나 연속방정식은 만족하지 않는다.

2) First Corrector Step

새로운 압력장 P^* 와 연속 방정식을 만족시키는 속도장(u^{**}, v^{**}, w^{**})으로 운동량 방정식을 나타내면 아래와 같다.

$$\begin{aligned} A_P u^{**}_P &= \Sigma A_{nb} u^{**}_{nb} - P^{*}_x J + S_u \\ A_P v^{**}_P &= \Sigma A_{nb} v^{**}_{nb} - P^{*}_y J + S_v \\ A_P w^{**}_P &= \Sigma A_{nb} w^{**}_{nb} - P^{*}_z J + S_w \end{aligned} \quad (11)$$

위 식을 이용하여 압력 보정 방정식을 구하면, SIMPLE알고리즘에서와 같은 압력 보정 방정식을 얻을 수 있다.

3) Second Corrector Step

1차 수정된 **속도장과 압력장은 운동량 방정식과 연속 방정식을 동시에 만족시키지 못한다. 따라서 두 번째 수정 단계가 계산되어진다. 적절한 압력장 P^{**} 와 속도장으로 운동량 방정식을 나타내면 다음과 같다.

$$\begin{aligned} A_P u^{***}_P &= \Sigma A_{nb} u^{***}_{nb} - P^{**}_x J + S_u \\ A_P v^{***}_P &= \Sigma A_{nb} v^{***}_{nb} - P^{**}_y J + S_v \\ A_P w^{***}_P &= \Sigma A_{nb} w^{***}_{nb} - P^{**}_z J + S_w \end{aligned} \quad (12)$$

위의 운동량 방정식을 증분 형태로 바꾸어 차분화된 연속 방정식에 대입하면 첫 번째 예측 단계에서와 같은 형태의 압력 보정 방정식을 얻을 수 있다. 두 연속 방정식에 있어서의 차이점은 생성항만 다를 뿐 나머지 계수들은 동일하다.

난류 유동 해석의 경우 이와 같은 절차는 동일하게 $k-\epsilon$ 방정식에 적용되어 진다.

전체 PISO알고리즘의 해석 순서는 다음과 같다.

1. 이전 계산 단계의 해(u^n, v^n, w^n, P^n)를 사용하여, 운동량 방정식의 계수를 계산한다. 그리고 내재적 방법으로 방정식을 푼다. 이것이 속도에 대한 예측 단계이다.
2. 예측된 속도 성분들을 이용하여 압력 방정식의 계수를 계산한 다음 압력장을 구한다. 이것이 압력에 대한 예측 단계이다.
3. 외재적(explicit)형태의 방정식을 이용하여 속도장을 수정한다.
4. 수정된 속도 성분을 이용 압력 방정식의 계수를 다시 계산하여 압력장을 수정한다.
5. 수정된 압력장을 이용하여 속도장을 다시 계산한다.
6. 수렴된 값을 가질 때 까지 위의 과정을 반복한다.

2.3 수렴 판정 조건

수렴하는 아래와 같은 조건을 만족시킬 때 얻어진다.

ERROR

$$= \frac{(|\Delta u|_{\max} + |\Delta v|_{\max} + |\Delta w|_{\max})}{U_{ref}} + \frac{|P|_{\max}}{\rho U_{ref}^2}$$

$$\leq 3 \times 10^{-4}$$

여기서 $\Delta u, \Delta v, \Delta w$ 는 운동량 방정식의 해를 구하기 위한 각 iteration(또는 time step)에서의 속도 변화를 나타낸다.

3. 결 과

이미 기존의 많은 연구[2][4][7][9][10]에서 2차원 유동과 축대칭 유동에 대하여 많은 비교가 있었으나 3차원 유동에 대하여는 비교된 적이 없으므로, 각 알고리즘 비교 대상을 3차원 유동으로 제한한다. 본 연구에서는 3차원 일반 좌표계에 있어서 SIMPLE-C로 짜여진 code를 SIMPLE과 PISO로 각각 수정하여 사용하였다. 적용 대상은 3차원 관 유동과 3차원 곡관 유동에서의 정상 상태(steady state)유동장이다. 일반적으로 정상 상태를 얻는 방법으로는 시간 전진법과 반복 계산법이 있으며, 본 연구에서는 두가지 방법 각각에 대한 수렴 속도를 비교하고자 한다.

3.1 3차원 관유동

Fig. 3은 수치 해석에 사용된 관의 형상과 격자계를 보여준다. 관의 지름은 층류 유동과 난류 유동에 관계없이 일정하고, 관의 길이는 완전 발달 영역을 보여 주기 위하여 충분히 길게 설정하여 층류 유동시 관의 길이는 5m이며, 난류 유동의 경우 13m이다. 관을 통해 흘러가는 유체는 공기로서 Reynolds number는 층류 유동시 100이며, 난류 유동일 때 10000이다. Fig. 4는 반복 계산법에 의한 층류 유동일 경우 각 알고리즘에서 수렴에 소요된 시간을 보여 준다. 기존의 2차

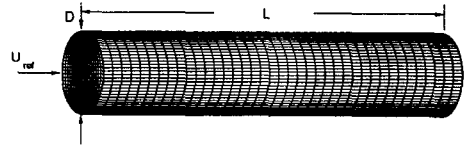


Fig. 3 Geometry and grid of duct

원 연구들에서는 일반적으로 PISO, SIMPLE-C, SIMPLE의 순으로 수렴 속도가 빠른 것으로 알려져 있으나, 3차원 비직교 일반 좌표계에서의 최종 수렴 속도는 SIMPLE, PISO, SIMPLE-C의 순으로 나타나고 있다. Fig. 4에서 보는바와 같이 PISO알고리즘이 SIMPLE알고리즘보다 느린 이유는 PISO알고리즘의 식 전개 과정에서 살펴 볼 수 있다. 즉 PISO알고리즘은 첫 번째 corrector단계까지는 SIMPLE알고리즘과 동일하나, 추가되어지는 두 번째 수정 단계에 소요되는 시간이 증가함에 따라 전체적으로 수렴 속도가 늦어짐을 알 수 있다. 이러한 현상은 2차원 계산에서는 잘 나타나지 않으나 3차원 일반 좌표계로 바뀔에 따라 계산되어질 변수의 수가 증가하여 나타나는 결과로 판단된다. Fig. 5는 시간 전진법에 의한 층류 유동시의 각 알고리즘에서의 수렴 속도를 보여준다. Issa와 그 밖의 많은 참고 문헌에서 나타난 것처럼 SIMPLE 과 SIMPLE-C알고리즘에 비하여 좋은 PISO알고리즘의 수렴에 대한 안정성을 보여 준다. PISO알고리즘은 수렴성이 보장되는 시간 간격의 범위가 SIMPLE과 SIMPLE-C알고리즘에 비하여 매우 넓음을 알 수 있다. 이에 비해 SIMPLE, SIMPLE-C알고리즘은 적절한 시간 간격을 정하지 않으면 수렴에 상당한 시간이 소요됨을 알 수 있다.

Fig. 6은 반복 계산법에 의한 난류 유동시의 각 알고리즘에서의 수렴 속도를 보여준다. 최종적인 수렴 속도는 SIMPLE-C, PISO, SIMPLE의 순이며, ERROR의 값이 10^{-2} 까지는 PSIO, SIMPLE-C, SIMPLE순이다. 이와 같이 PISO알고리즘의 수렴성이 좋은 이유는 추가된 수정 단

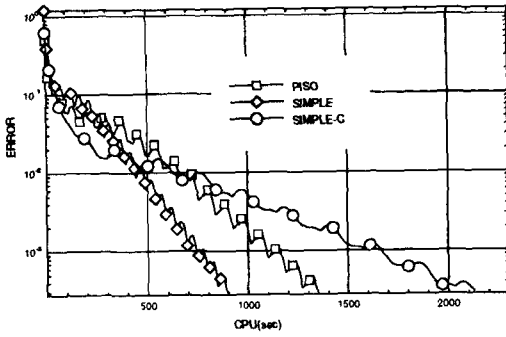


Fig. 4 Convergence time by iterative method in laminar duct flow

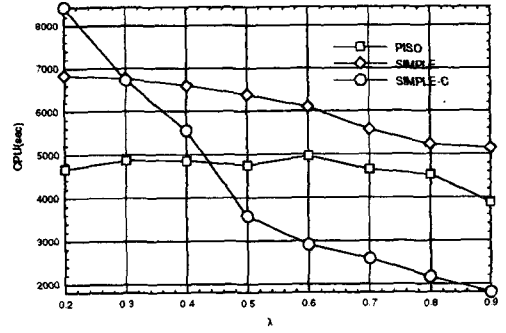


Fig. 7 Convergence time due to variation of relaxation factor(laminar flow)

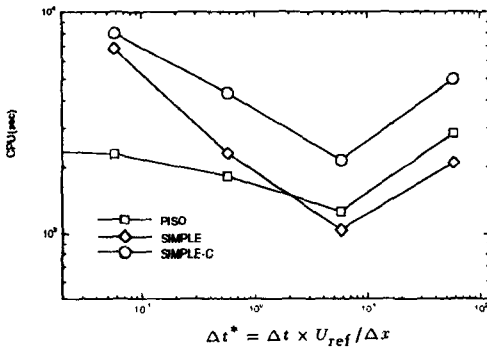


Fig. 5 Convergence time by time-marching method in laminar duct flow

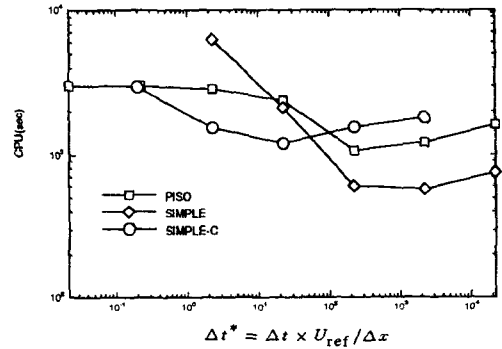


Fig. 8 Convergence time due to variation of time step(turbulence flow)

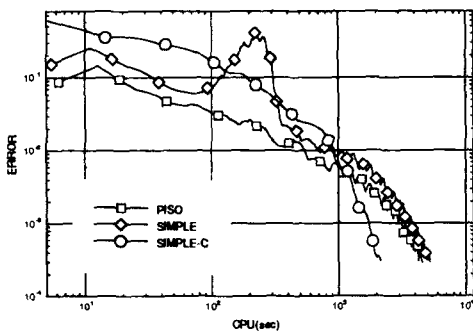


Fig. 6 Convergence time by iterative method in turbulence duct flow

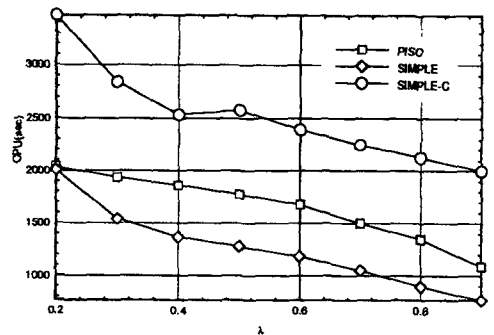


Fig. 9 Convergence time due to variation of relaxation factor(laminar flow)

제가 초기값들로부터 수렴값들을 찾아가는데 있어 보다 효율적임을 보여준다. 하지만 어느 정도 수렴이 이루어진 후에는 추가된 단계에 소요되는 시간이 수렴 속도를 보완하지 못함을 알 수

있다. 또, 층류 유동에서와는 달리 PISO알고리즘이 SIMPLE알고리즘보다 빠른 이유는 k, ϵ 에 적용된 수정 단계가 수렴에 상당한 도움을 준다는 것을 알 수 있다. Fig. 7은 반복계산법에서 λ 의

값의 변화에 따른 난류 유동에서의 각 알고리즘의 수렴 시간의 변화를 보여 준다. 그림에서 나타나듯이 SIMPLE-C 알고리즘은 λ 값의 변화에 아주 민감하며, PISO 알고리즘은 λ 의 변화에 안정적이다. 이는 PISO 알고리즘이 시간 전진법으로 계산 할 경우 넓은 범위의 시간 간격에서 안정성, 수렴성이 보장되는 특징이 λ 의 변화에도 같이 적용됨을 알 수 있다. Fig. 8에서 보여주는 것은 시간 전진법에 의한 난류 유동시의 각 시간 간격에 대한 알고리즘들의 수렴 속도이다. Fig. 5에서의 수렴순서와는 차이가 있으나 각 알고리즘의 시간 간격의 변화에 대한 의존성은 유사하다. 즉 PISO 알고리즘이 시간 간격의 변화에 대하여 가장 안정적이며, 넓은 범위의 시간 간격에 대하여 수렴성이 보장된다. 그리고 SIMPLE 알고리즘이 시간 간격의 변화에 가장 민감하다.

Fig. 9는 층류 유동에서 λ 를 변화했을 때 각 알고리즘에 있어서의 수렴 속도의 변화를 보여 준다.

3.2 3차원 곡관 유동

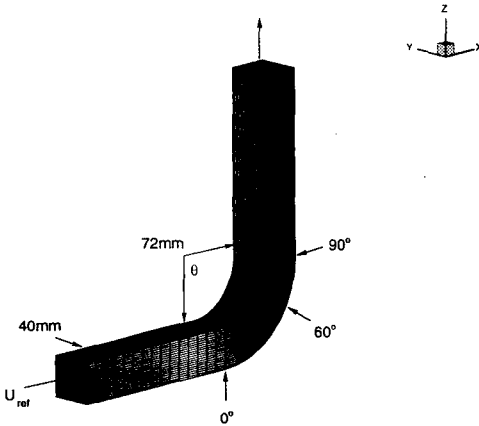


Fig. 10 Geometry and grid of 90-degree bent square duct

실제 알고리즘을 통해 계산되어진 결과들의 정확성을 확인하기 위하여 3차원 곡관 유동을 해석하였다. 계산 결과들의 비교 자료로는 Humphrey[5]에 의한 실험 결과와 Rhie[6]에 의

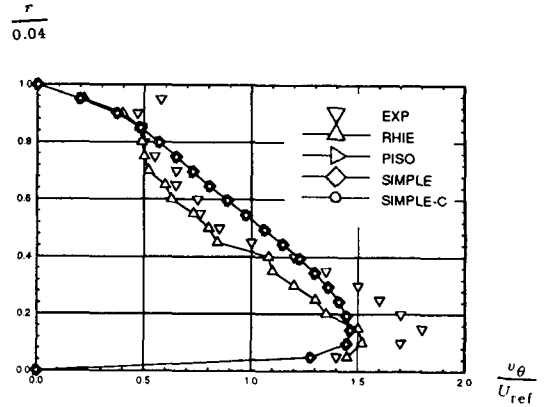


Fig. 11 Velocity profile normal to 60° section (y=0.02m)

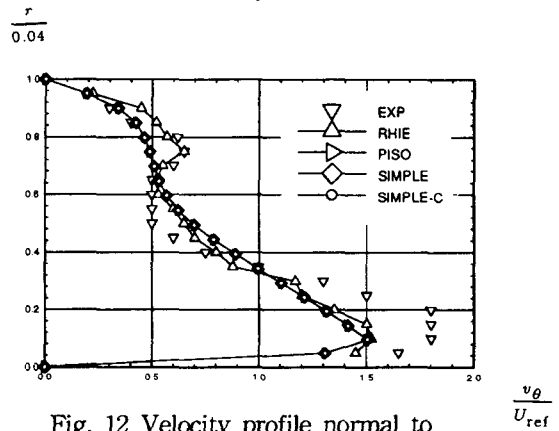


Fig. 12 Velocity profile normal to 90° section (y=0.02m)

한 계산 결과들이다. Fig. 10에서 보여지는 것은 90°로 꺾여지는 사각 곡관의 형상과 격자계의 모습이다. 입구 조건으로 주어지는 유동 조건은 층류 완전 발달류이다. 이때 흘러가는 유동의 Reynolds number는 790이다. Fig. 11은 $y=0.02$ 이며, $\theta=60^\circ$ 인 단면에서 수직인 속도 성분의 분포를 보여준다. 각 알고리즘의 계산 결과들은 서로 일치하나 실험 결과와는 차이가 있다. 특히 양쪽 벽부근에서 많은 차이가 발생한다. 이는 계산 격자의 균일한 분포와 격자수의 영향으로 생각된다. Fig. 12는 $\theta=90^\circ$ 인 단면에서의 속도 분포를 보여준 것으로 여기서는 바깥쪽 벽부근에서 실험 결과와 수치 해석 결과들이 차이가 남을 알 수 있다. 그러나 여기서도 각 알고리즘

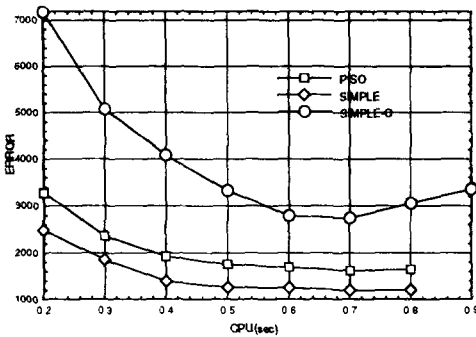


Fig. 13 Convergence time due to variation of relaxation factor in 90°bent square duct

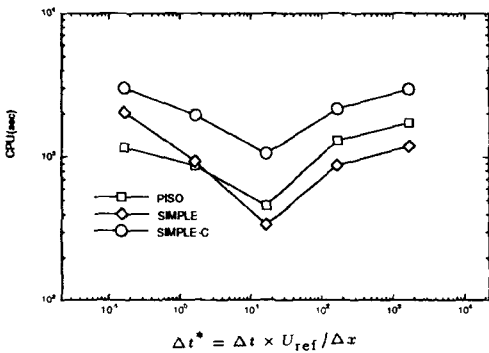


Fig. 14 Convergence time due to variation of time step in 90°bent square duct

들의 계산 결과들은 일치한다. 이상의 결과들로부터 한 프로그램 내에서의 알고리즘의 변화는 각 알고리즘에 동일한 수렴 조건을 적용하면 수렴된 계산 결과는 서로 일치함을 보여 준다. 즉 알고리즘의 변화는 수렴 속도와 수렴 여부에는 영향을 미치나 수렴된 결과에는 영향을 미치지 않음을 알 수 있다.

Fig. 13는 반복 계산법에 의한 층류유동시의 곡관에서의 각 알고리즘의 λ 값의 변화에 따른 수렴 시간을 보여 준다. Fig. 9와 그림은 유사하나 곡관 유동에서는 관 유동에서와는 달리 PISO, SIMPLE알고리즘은 λ 값이 0.9일 때 수렴하지 못 한다. 이는 유동이 복잡해 질 때 PISO, SIMPLE알고리즘의 불안정성을 드러낸다. Fig. 14의 그림은 Fig. 5와 같은 형태의 그림을 보여 준다. 따라서 시간 전진법에 의한 층류 유동시의 수렴 시간은 관유동과 곡관 유동에 있어서 유사

함을 보여 준다.

4. 결론

본 연구에서는 비압축성 유동 해석을 위한 3차원 수치 해석 방법에 있어서 SIMPLE, SIMPLE-C 그리고 PISO의 수렴 시간을 비교, 검토하여 다음과 같은 결론을 얻었다.

1. 3차원 유동에서의 PISO알고리즘의 적용은 2차원에서 3차원으로의 확장에 따른 계산 소요 시간의 증가로 2차원에서와 같이 다른 알고리즘에 비하여 큰 수렴 시간의 절약을 기대 할 수 없다.

2. 3차원 일반 좌표계에서 SIMPLE과 SIMPLE-C알고리즘은 기존의 많은 2차원 연구에서처럼 SIMPLE-C의 일방적인 우수성이 확인 되지 않으며, 오히려 관, 곡관의 층류 유동에서는 SIMPLE알고리즘의 수렴 속도가 훨씬 빠르다. SIMPLE-C알고리즘은 유동의 종류에 관계 없이 SIMPLE알고리즘보다 안정된 수렴성을 보장한다.

3. 3차원 일반 좌표계에서의 PISO알고리즘의 적용 효과는 관 유동과 곡관 유동의 계산 결과에서 나타난 듯이 반복 계산법에서보다 시간 전진법에 더욱 효과적이며 넓은 범위에서 안정성있게 사용가능함을 알 수 있다.

참 고 문 헌

- (1)Pantanka, S. V., 1980 "Numerical Heat transfer and Fluid Flow", McGraw Hill
- (2)Van Doormall, J. P. and Rathby, G. D., 1984, "Enhancements of the SIMPLE Method for Predicting Incompressible Fluid Flows", J. Numerical Heat Transfer, Vol 7, pp. 147-163
- (3)Issa, R. I. " Solution of the Implicit Discretized fluid flow equations By operator

Splitting", J. Comp. Phys, Vol 62, pp. 40-65

(4)Jang, D. S., Jelti, R. and Acharay, S., 1986, "Comparison of the PISO, SIMPLER, and SIMPLE-C algorithms for the treatment of the Pressure-Velocity coupling in steady flow problems", J. Numerical Heat Transfer, Vol. 10, pp. 209-228

(5)Humphrey, J. A. C., Taylor, A. M. K. and Whitelaw, J. H. "Laminar Flow in a Square Duct of Strong Curvature", J. Fluid Mech, Vol. 83, pp. 509

(6)Rhie, C. M., 1985, "Block-Implicit Calculations of three-Dimensional Laminar Flow in Strongly Curved Ducts", AIAA J, Vol. 23

(7)Wanik, A. and Schnell, U., 1989, "Some Remarks on the PISO and SIMPLE Algorithms for Steady Turbulence Flow Problems", Computers & Fluids, Vol. 17, pp. 555-570

(8)Launder, B. E. and Spalding, D. B., "Mathematical Models of Turbulence", Academic Press, London

(9)Issa, R. I, Gosman. and Wankins, A. P. "The Computation of Compressible and Incompressible recirculating flows by a non-iterate implicit scheme", J. Comp. Phys, Vol. 62, pp. 66-82

(10)권용훈, 1993, "Benchmark tests on PISOC for Navier-Stokes Equations", AFR-92-A, pp. 123-169