내재적 방법을 이용한 비정렬 유동해석 기법의 병렬화

김 주 성[*1], 강 희 정[*1], 박 영 민[*1], 권 오 준[*2]

Parallelization of an Unstructured Implicit Euler Solver

J. S. Kim, H. J. Kang, Y. M. Park and O. J. Kwon

An unstructured implicit Euler solver is parallelized on a Cray T3E. Spatial discretization is accomplished by a cell-centered finite volume formulation using an unpwind flux differencing. Time is advanced by the Gauss-Seidel implicit scheme. Domain decomposition is accomplished by using the k-way N-partitioning method developed by Karypis. In order to analyze the parallel performance of the solver, flows over a 2-D NACA 0012 airfoil and a 3-D F-5 wing were investigated.

Key Words: 비정렬 격자(Unstructured Mesh), 내재적 방법(Implicit Method), 영역분할법(Domain Decomposition), 병렬화(Parallelization), 비정상 유동(Unsteady Flow), 로터 공기 역학(Rotor Aerodynamics)

1. 서론

단일 CPU의 성능이 시간에 대해 지수함수 적으로 증가함에도 불구하고, 이러한 성능향상이 물리적인 제약 등으로 인하여 계속 이어질 수는 없다. 현재 단일 CPU의 성능은 한계점에 빠르게 도달하고 있는 실정이며, 컴퓨터의 성능 향상을 위해서는 병렬 컴퓨터와 같은 개념의 하드웨어 방식이 필요하게 된다. 1980년대 들어 병렬 컴퓨터에 대한 기술이 비약적으로 발전하면서 우수한 성능의 병렬 컴퓨터가 많이 개발되었다. 현재 전산유체역학 분야에서 이러한 병렬 컴퓨터는 3차원에 대한 실제적인 유동을 해석할 수 있는 유용한 방법이 되어가고 있다.

전산유체역학 분야에서 비정렬 격자를 이용한 해석 기법은 복잡한 형상에 대한 격자형성의 용이성과, 적응격자를 통한 해의 정확성과 기법의 효용성을 증가시킬 수 있는 가능성에 힘입어 널리 사용되고 있다. 이러한 비정렬 격자에 대한 병렬화 기법은 Venkatakrishnan[1] 등에 의해 많이 연구되었으며, 주로 외재적 시간 적분법과 GMRES 방법을 이용한 내재적 시간 적분법을 사용한 경우에 대해 다루었다. 또한 Bruner[2]는 외재적 시간 전진 기법과 Gauss-Seidel 방법, SGS 방법, Block Jacobi 방법 등을 이용한 내재적 기법을 비교하여 성능을 평가하였다. 본 연구에서는 이 중에서 Gauss-Seidel 방식의 내재적 기법에 대한 병렬 처리를 수행하여, 2차원과 3차원에 대한 병렬화 특성과 성능에 대해 논의하고자 한다. 또한 비정상 유동과 헬리콥터 로터 주위의 유동 등에 적용하여 개발된 병렬 유동 해석

*1 한국과학기술원 항공우주공학과
*2 정회원, 한국과학기술원 항공우주공학과

기법이 실제의 복잡한 형상에 적용되었을 경우에 대해 논하고자 한다.

병렬 컴퓨터는 공유 메모리(shared memory) 형태와 분산 메모리(distributed memory) 형태로 구분된다. 공유 메모리(shared memory) 형태의 병렬 컴퓨터는 대체적으로 약 20개 이상의 CPU에서 포화(saturation)현상이 발생되어 현재 개발되는 병렬 컴퓨터는 대부분 분산 메모리(distributed memory) 형태를 갖는다. 그리고 병렬화 방법은 함수 분할(functional decomposition) 방법과 영역 분할(domain decomposition)방법으로 구분된다. 이러한 구분은 각각의 processor에 해당하는 일이 주어지는 방법에 따라 구분하는 방법이다. 본 연구에서는 분산 메모리 형태의 병렬 컴퓨터인 Cray T3E를 사용하여 계산하였으며, 영역 분할 방법을 이용하여 해석 프로그램을 병렬화 하였다. 또한 자료전달을 위해 Cray T3E의 MPI(Message Passing Interface)를 사용하였다.

2. 수치적 방법

3차원 유동에 대해 적분 형식으로 표현된 Euler 방정식은 다음과 같다.

$$\frac{\partial}{\partial t}\int_{\Omega} Q dV + \int_{\partial\Omega} F(Q)\cdot\vec{n} dS = 0 \tag{1}$$

여기서 Q는 보존형 변수, F(Q)는 대류항이며 다음과 같다.

$$Q = [\rho,\ \rho u,\ \rho v,\ \rho w,\ e_0]^T \qquad F(Q)\cdot\vec{n} = (\vec{V}\cdot\vec{n})\begin{pmatrix}\rho\\ \rho u\\ \rho v\\ \rho w\\ e_0+p\end{pmatrix} + p\begin{pmatrix}0\\ n_x\\ n_y\\ n_z\\ 0\end{pmatrix}$$

위의 지배방정식은 자유류의 밀도 ρ_∞와 음속 a_∞로 무차원화 되었다. 또한 무차원화된 압력과 온도는 이상기체 가정을 통해서 다음과 같이 구해진다.

$$p = (\gamma-1)\left(e_0 - \frac{1}{2}\rho(u^2+v^2+w^2)\right)$$

$$T = \gamma p/\rho$$

2.1 공간차분법

Roe의 flux difference splitting을 사용하여 대류항을 계산하였으며, 각각의 셀의 경계면 k를 통한 flux는 다음과 같이 표현된다.

$$F_k = \frac{1}{2}[F(Q_L) + F(Q_R) - |\tilde{A}|(Q_R - Q_L)] \tag{2}$$

여기서 Q_L과 Q_R은 k면의 왼쪽과 오른쪽의 유동변수들을 나타낸다. 행렬 $|\tilde{A}|$는 Roe average된 유동량을 사용하여 계산하며, 자세한 내용은 참고문헌 [3]에 기술되어 있다.

셀 face에서 고차 정확도로 flux를 계산하기 위해 참고 문헌 [3]에 기술된 Reconstruction 방법을 사용하였다. 이 방법은 격자내의 격자점들로 구성된 폐다각형을 적분 경로로 하여 Green-Gauss 정리를 적용하고, 삼각형과 사면체의 기하학적 특성을 이용하는 방법으로 2차원 삼각형에 대해서는 다음과 같은 형태의 수식으로 표현된다.

$$q_{f_{1,2}} = q_c + \frac{1}{3}\left[\frac{1}{2}(q_{n1}+q_{n2}) - q_{n3}\right] \tag{3}$$

여기서 $f_{1,2}$는 격자점(node) n_1과 n_2로 구성된 face이며, n_3는 $f_{1,2}$와 마주보는 격자점이다. 또한 삼차원의 사면체에 대해서는 다음과 같다.

$$q_{f_{1,2,3}} = q_c + \frac{1}{4}\left[\frac{1}{3}(q_{n1}+q_{n2}+q_{n3}) - q_{n4}\right] \tag{4}$$

여기서 $f_{1,2,3}$는 격자점 n_1, n_2, n_3로 구성된 face이며, n_4는 $f_{1,2,3}$와 마주보는 격자점이다.

이러한 reconstruction 방법에서는 셀 중심의 값들을 이용하여 격자점에서의 값을 구하는 방법이 매우 중요하며, 본 연구에서는 전체적으로 2차 정확도를 얻을 수 있는 Holmes와 Connel[4]이 제안한 방법을 사용하였다.

2.2 시간적분법

지배방정식 (1)을 시간에 대해 Euler 후방차분법을 적용하고, flux에 대해 Newton 형식의 선형화를 하면 다음과 같은 형식으로 쓸 수 있다.

$$\left[D_i + \sum_j^{nface} S_{j,k}\right]\Delta Q = -R_i^n \tag{5}$$

여기서 $\Delta Q = Q^{n+1} - Q^n$을 나타낸다. Flux Jacobian의 대각항과 비대각항, D_i와 $S_{i,k}$는 다음과 같다.

$$D_i = \frac{V_i}{\Delta t} + \sum_j^{nface} S_{j,i} \quad S_{j,k} = \frac{\partial F_j}{\partial Q_k} \tag{6}$$

여기서 아래첨자 j는 i번째 셀의 주변 셀 들을 나타낸다. 식 (6)에서 flux Jacobian은 Roe의 average된 특성치를 이용하여 계산하였으며, 참고문헌 [5]에 자세히 기술되어 있다. 식 (5)의 선형 방정식은 반복계산 방법을 사용하여 계산되며, 본 연구에서는 Gauss-Seidel 방식을 사용하였다.

3. 병렬 처리

3.1 영역 분할법

영역 분할법을 통하여 계산 영역을 분할할 경우, 가장 중요한 문제는 나누어진 부 영역에서의 격자점이나 격자의 수를 같게 하여 병렬 계산 시 각각의 processor가 같은 시간 내에 작업을 끝낼 수 있게 하는 load balancing과 각 부영역 간의 경계면 수를 최소화하는 것이다. 나누어진 부영역 간의 경계는 병렬 계산 시 필요한 자료 전달량과 비례하며, 자료 전달에 소요되는 시간을 최소화하기 위해서는 영역간의 경계를 최소화하여야 한다. 이러한 영역 분할법으로 공간 좌표를 이용하는 좌표 분할법, Graph 분할법, Spectral 분할법 등이 있다. Venkatakrishnan[6]은 비정렬 격자계에서 좌표 분할법, RCM(Recursive Cuthill -McKee) 방법, Spectral 방법에 대한 고찰을 통해 Spectral 방법이 가장 적은 경계 영역을 가지면서 load balancing을 이룰 수 있음을 제시하였다. Simon 등은 이러한 Spectral 방법을 다단계 기법으로 발전시켜 영역분할 시간을 줄였다. 또한 다단계 Spectral 방법과 유사한 방법으로 미네소타 대학의 Karypis에 의해 개발된 k-way N-partitioning 방법이 있다[7]. 이 방법은 원래의 영역을 Graph 이론을 바탕으로 축소(coarsening)한 후 축소한 영역을 k-way 방법으로 분할하고, 다시 원래 영역으로 환원(uncoarsening)하는 방법이다. 또한 축소된 영역의 확장 시 영역의 경계를 줄일 수 있는 대응 기법을 활용하여 빠른 시간 내에 작은 경계 면을 가지는 영역으로 분할할 수 있는 특징이 있다. Karypis는 이러한 연구의 결과를 Metis라는 공개 프로그램으로 만들어 배포하고 있으며, 본 연구에서는 Metis 라이브러리를 사용하여 영역분할을 수행하였다.

3.2 해석 프로그램의 병렬 처리

병렬 처리 시 자료의 전달은 계산 시간 이외의 시간을 소요하는 overhead가 된다. 따라서 자료 전달 함수의 자료 전달 방식을 이해하고 이를 최적화 할 수 있어야 한다. 참고 문헌 [8]에서는 Cray T3E의 자료전달 특성에 대해 논하였으며, 많은 자료를 묶어서 전달하는 방식이 효율적임을 제시하고 있다.

본 연구의 해석 프로그램의 특성상 computa- tional boundary에서 3가지의 자료 전달이 필요하다. 첫째로 경계면에서 고차로 대류항을 계산하기 위한 경계면(face)에 대한 자료 전달이 필요하다. 식 (2)를 보면 computational boundary에 놓여 있는 격자면에서, 인접한 processor에서 계산된 격자면에 대

한 자료 전달을 통해서 고차로 flux가 계산될 수 있음을 알 수 있다. 두 번째로는 경계면에 접한 셀 내에서 reconstruction을 하기 위해 computational boundary에 놓여있는 격자점(node)에 대한 자료전달이 필요하다. 격자점에 대한 계산은 격자점을 둘러싸고 있는 셀 중심의 값들을 사용하여 구하므로, 격자점이 computational boundary에 있다면 이 격자점에 대한 외부 processor에서의 영향을 고려하여 구해야 한다. 세 번째로 경계면에 접한 셀에서 flux Jacobian을 계산하기 위해 셀 중심 값의 전달이 필요하다.

4. 결과 및 고찰

4.1 2차원 유동에 대한 해석

사용된 격자는 advancing front 방법을 사용하여 생성하였다. 셀 수는 8,915개이며, 격자점의 개수는 4,560개이고, 이 중에 131개가 익형 표면에 위치한다. Fig. 1은 k-way N-partitioning 방법을 이용하여 NACA 0012 익형 주위의 격자를 8개의 영역으로 분할한 결과이다. Table 1은 분할된 격자에 대한 각각의 격자, 격자점, 격자면에 대한 개수와 computational boundary에서의 격자와 격자점의 개수를 나타내고 있다. 일반적으로 어떠한 영역 분할 기법을 사용하더라도 셀 수와 격자면 수, 그리고 격자점의 수를 모두 같게 할 수는 없다. 이러한 이유로 본 연구에서 사용되는 셀 중심 기법에서는 셀 수의 load balancing에 초점을 맞추었다. Table 1에서 볼 수 있듯이 셀 수에 대한 load balancing이 잘 되어 있음을 확인할 수 있다. Fig. 2는 8개의 영역으로 나뉘어진 경우에 대해 유동 조건이 마하수가 0.8, 받음각이 1.25°인 비점성 유동에 대한 등 마하선도이다. 각각의 영역의 경계면에서 부드러운 연결을 보이고 있으며, 이를 통해 각각의 processor에서의 자료전달이 적절히 이루어지고 있음을 알 수 있다. Fig. 3은 하나의 processor를 사용한 결과와 2개, 8개의 processor를 사용한 결과를 비교한 그림이며, 계산 결과가 거의 일치함을 볼 수 있다. 영역의 수를 달리하면서 계산을 반복하여 일반적인 전산학적 관점에서의 병렬 처리의 효율을 계산하면 Fig. 4와 같다. 적은 격자수로 인하여 8개 이상의 processor를 사용할 경우 병렬 처리에 따른 overhead가 큰 것으로 나타났다. 그러나, 전체적인 성능에 있어서는 Cray C90의 한 CPU 성능을 상회하는 것으로 나타났다.

4.2 3차원 유동에 대한 해석

3차원의 비점성 유동에 대한 병렬화 기법의 검증을 위하여 F-5 날개를 선택하였다. 계산은 마하수는 0.9이며, 받음각이 0°인 비점성 유동에 대해서 수행하였다. Fig. 5는 8개의 영역으로 나뉘어진 F-5 날개주위의 표면 격자이며, Fig. 6은 이 격자계에 대해 계산된 등마하선도 이다. 그림에서 볼 수 있듯이 2차원의 경우와 마찬가지로 영역의 경계면에서 부드러운 연결을 보이고 있다. Fig. 7은 스팬의 위치에 따라 실험치와 계산된 결과를 나타낸 그림이며, 대체적으로 실험결과와 잘 일치하고 있다. Fig. 8은 영역의 수에 따른 반복계산에 대한 수렴도를 나타낸 그림이다. 영역의 수가 증가할수록 수렴성의 저하현상을 볼 수 있다. 이러한 현상은 Gauss-Seidel 내부 반복계산(sub-iteration)시에는 연결된 영역에 대한 자료 전달이 이루어지지 않기 때문에 발생되는 현상이다. Fig. 9는 자료전달 시간과 계산 시간 그리고 전체 걸린 시간을 영역의 수에 따라 나타낸 그림이다. 자료전달 시간 측면에서는, 영역의 수가 10개 이상이 되면 자료 전달의 양이 증가함에도 불구하고 자료 전달 시간이 거의 비슷한 것을 볼 수 있다. 이러한 원인은 대부분의 자료전달 시간이 각각의 processor가 계산을 끝낸 후 동기화 되기 위해 대기하는 시간이기 때문에 발생되는 현상이다. 계산 시간 측면에서는 영역의 분할수에 관계없이 계산 양이 같음에도 불구하고 영역의 수가 증가함에 따라 계산 시간이 감소함을 볼 수 있다. 이러한 현상은 Cray T3E의 cache 구조의 특성과 virtual memory의 사용 등의 원인으로 예상하고 있다[9]. 이러한 영역의 증가와 함께 계산 시간이 감소하는 현상으로 인하여, 전체적으로 걸린 시간은 processor수가 증가할수록 감소하게 된다. Fig. 10은 이러한 결과를 이용하여 이상적인 병렬화 효율과 본 연구에서 개발된 병렬 프로그램의 병렬 효율을 비교한 그림이다. 영역의 증가와 함께 계산시간이 감소하게 되어 이상적인 병렬화 효율보다 높은 것을 볼 수 있다.

다음으로 피칭(pitching) 운동을 하는 F-5 날개 주위의 비정상 유동에 대해 해석하였다. 마하수는

0.9이며, 날개는 진폭이 0.1°, reduced frequency가 0.137인 사인파 형상의 주기적인 운동을 한다. Fig. 11은 받음각의 변화에 따른 양력 계수에 대한 곡선이며, 계산은 4개의 processor를 사용하여 수행하였다.

4.3 로터 블레이드 유동 해석

헬리콥터 로터는 Caradonna와 Tung이 제자리 비행시에 대해 실험한 형상을 사용하였다. 로터는 2개의 블레이드로 구성되며, twist와 taper가 없는 직사각형 형상이다. 종횡비는 6이며, 날개단면은 NACA 0012로 끝단은 square 형상이다. Fig. 12는 계산에 사용된 격자이며, 셀 수는 201,799개, 격자점은 37,770개, 경계면에 7,006개의 격자점으로 구성되어 있다. 4개의 영역을 사용하여 병렬 계산을 하였으며, 유동조건은 collective 각이 8°, 끝단 마하수 M_T가 0.877인 천음속 유동이다. Fig. 13은 압력계수 분포를 각각의 영역에 대해 보이고 있으며, 영역의 경계면에서 자료전달이 잘 이루어지고 있음을 확인할 수 있다. Fig. 14는 스팬의 위치에 따른 압력계수 분포를 실험치와 비교한 그림이며, 대체적으로 잘 일치하고 있다.

5. 결론

본 논문에서는 비정렬 격자계에서의 내재적 방법을 이용한 유동해석 기법의 병렬화에 대한 연구를 수행하였다. 2차원 NACA 0012 익형과 3차원 F-5 날개에 대한 해석을 통해 본 연구에서 개발된 병렬화 기법의 정확성과 효율성에 대해 검증하였다. 정확성의 비교를 위하여 단일 processor에서의 결과 및 실험치와 비교를 하였으며, 대체적으로 정확한 계산결과를 얻을 수 있었다. 또한 3차원 유동에 대해서는 높은 병렬화 효율을 얻을 수 있었다.

후기

본 연구는 과학기술정책연구소 미래원천기술개발사업의 초고속 컴퓨터 기반 소프트웨어 및 응용 기술 분야중 병렬 3차원 CFD 소프트웨어 개발 과제지원에 의해 수행된 결과의 일부이며, 이에 감사드립니다.

참고문헌

[1] Venkatakrishnan, V., "Implicit Schemes and Parallel Computing in Unstructured Grid CFD", ICASE Report No. 95-28, 1995.

[2] Bruner, C. W., "Parallelization of the Euler Equations on Unstructured Grids", Virginia Polytechnic Institute and State University, Ph. D. Thesis, 1996.

[3] Frink, N. T., "Three Dimensional Upwind Scheme for Solving the Euler Equations on Unstructured Tetrahedral Grids", Virginia Polytechnic Institute and State University, Ph. D. Thesis, 1991.

[4] Holmes, D. G. and Connel S. D., "Solution of the 2-d Navier Stokes Equations on Unstructured Adaptive Grids", AIAA Paper 89-1932, 1989.

[5] 강희정, 권오준, "2차원 비정렬 격자에서 의 내재적 기법을 이용한 난류 유동 계산", 한국전산유체공학회 추계학술대회, 1997.

[6] Venkatakrishnan, V. Simon, H. D. and Barth, T. J., "A MIMD Implementation of a Parallel Euler Solver for Unstructured Grids", The Journal of Supercomputing, Vol. 6, pp. 117-137, 1992.

[7] Karypis, G., http://www/cs/num.edu/~karypis/metis.

[8] 김성호, "비정렬 격자를 사용한 병렬 전산 유체 해석 기법에 관한 연구", 한국과학기술원 항공우주공학과, 박사학위 논문, 1999.
[9] Xian-He Sun, "A Fast Tridiagonal Algorithm for a Class of CFD Applications", NASA Technical Paper 3585, 1996.
[10] 강희정, 권오준, "비정렬 격자를 이용한 제자리 비행시 헬리콥터 로터 공력 성능 계산", 한국항공우주학회 춘계학술발표회, 1999.

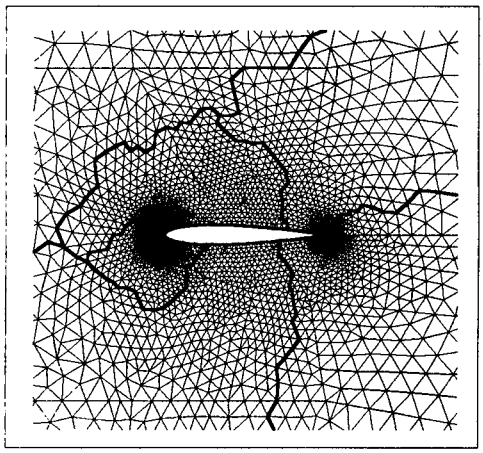

Fig. 1 8개로 분할된 NACA 0012 익형 주위의 격자

Domain number	# of Cells	# of C-Cells	# of Nodes	# of C-Nodes	# of Faces
0	1,115	86	614	91	1,728
1	1,114	59	600	62	1,713
2	1,115	77	607	81	1,721
3	1,114	71	601	73	1,714
4	1,115	94	627	99	1,741
5	1,114	92	625	96	1,738
6	1,114	70	600	73	1,713
7	1,114	81	605	85	1,718
Total	8,915	630	4,879	660	13,786

Table 1. 8개의 영역으로 분할된 격자에 대한 격자, 격자점, 격자면의 개수

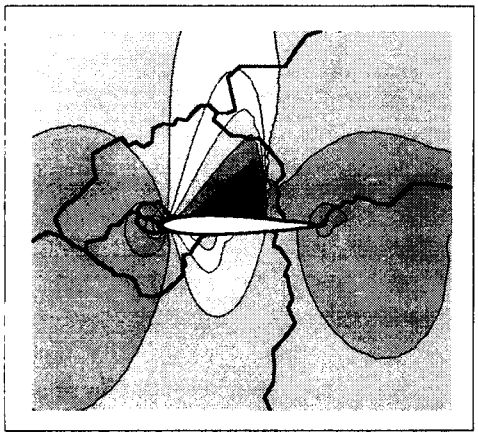

Fig. 2 8개 영역을 사용한 경우의 등 마하선도

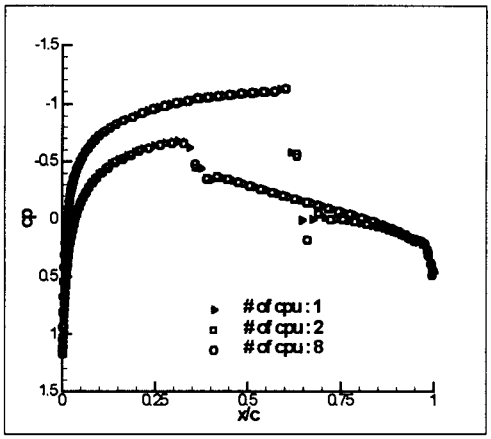

Fig. 3 여러 개의 processor를 사용한 경우의 압력 분포

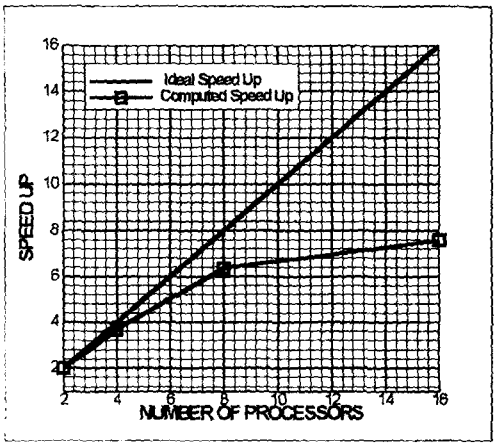

Fig. 4 2차원 유동에 대한 병렬화 효율

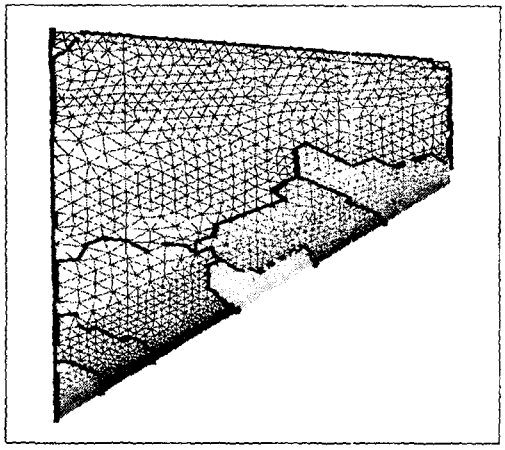

Fig. 5 8개의 영역으로 분할된 F-5 날개 주위의 표면 격자

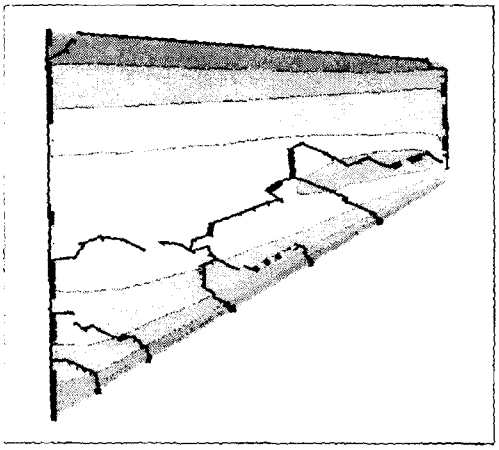

Fig. 6 8개의 영역을 사용한 경우의 F-5 날개 표면의 등마하선도

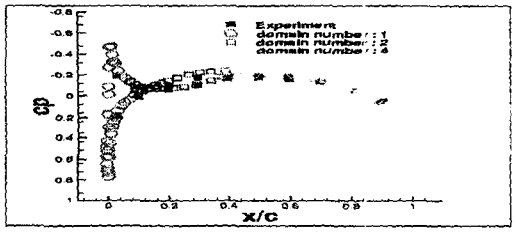

Fig. 7 F-5 날개주위의 압력계수 분포 : 18% span

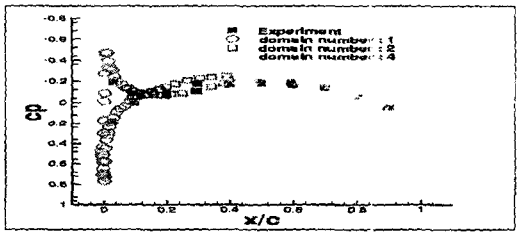

Fig. 7(계속) 51% span

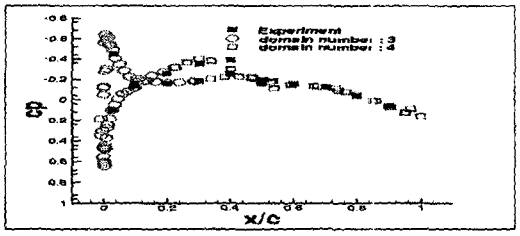

Fig. 7(계속) 88% span

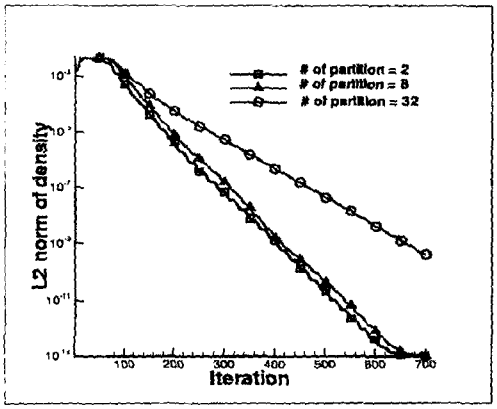

Fig. 8 CPU 개수에 따른 잔류치 수렴 곡선

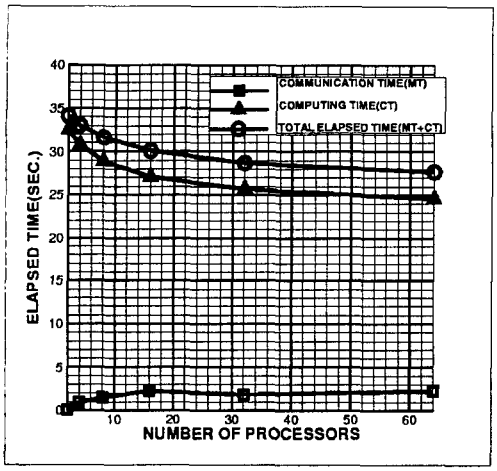

Fig. 9 자료 전달 시간, 계산 시간 및 전체 걸린 시간

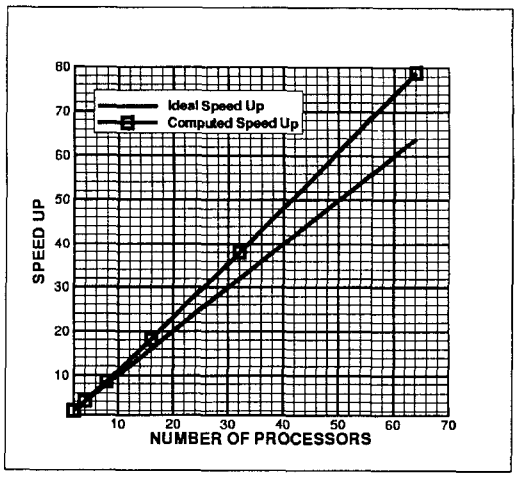

Fig. 10 3차원 유동에 대한 병렬화 효율

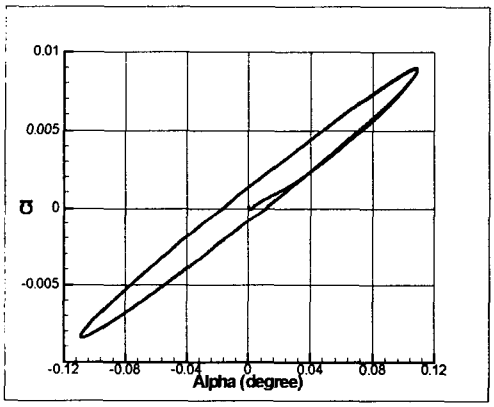

Fig. 11 F-5 날개에 대한 받음각에 따른 양력 계수 변화

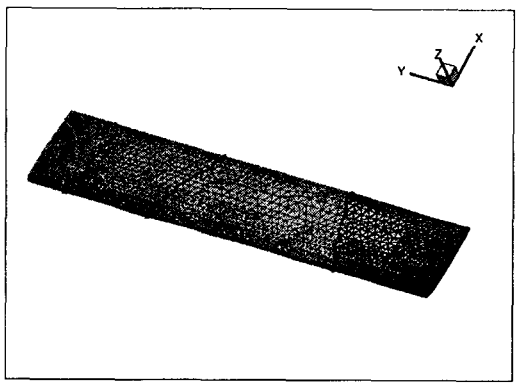

Fig. 12 4개의 영역으로 나누어진 로터 블레이드 표면 격자

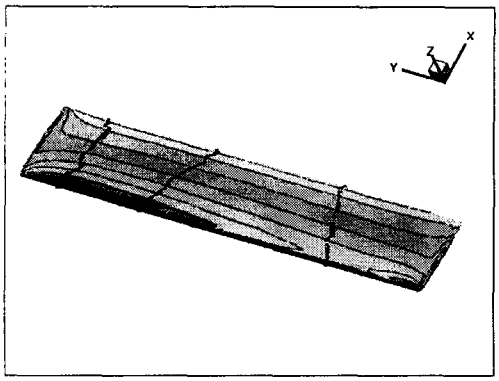

Fig. 13 로터 블레이드 주위의 등압력선도

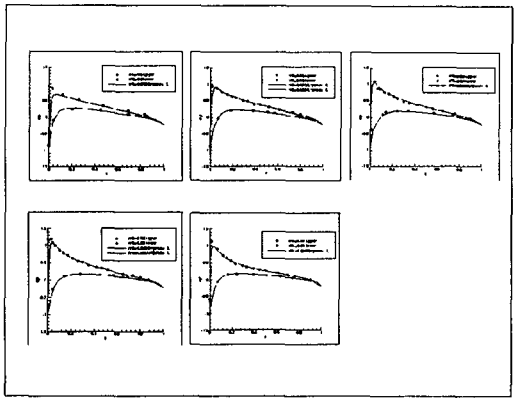

Fig. 14 로터 블레이드 주위의 압력 계수 분포