

내재적 방법을 이용한 비정렬 유동해석 기법의 병렬화

김주성^{*1}, 강희정^{*1}, 박영민^{*1}, 권오준^{*2}

Parallelization of an Unstructured Implicit Euler Solver

J. S. Kim, H. J. Kang, Y. M. Park, 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 upwind 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 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)

1. 서 론

최근 단일 CPU의 성능이 급속도로 발전하고는 있지만, 이러한 성능 향상은 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 이러한 병렬 컴퓨터에 대한 기술이 비약적으로 발전하면서 우수한 성능의 병렬 컴퓨터가 많이 개발되었으며, 이로 인해 전산유체역학 분야에서도 이러한 병렬 컴퓨터는 3차원에 대한 실제적인 유동을 해석할 수 있는 유용한 방법이 되어가고 있다.

전산유체역학 분야에서 비정렬 격자를 이용한 해석기법은 복잡한 형상에 대한 격자 형성의

용이성과, 적응격자(adaptive grid)를 통한 해의 정확성과 기법의 효율성을 증가시킬 수 있는 가능성에 힘입어 널리 사용되고 있다. 이러한 비정렬 격자에 대한 병렬화 기법은 Venkatakrisnan[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 적분법과 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 논의 하고자 한다. 또한 병렬 환경에서 비정상 유동을 계산하기 위한 방법에 대해 논의하고자 한다.

병렬 컴퓨터는 메모리의 구조에 따라 공유 메모리(shared memory) 형태와 분산 메모리(distributed memory) 형태로 구분된다. 공유 메모리 형태의 병렬 컴퓨터는 대체적으로 약 20개

^{*1} 한국과학기술원 기계공학과 대학원 항공우주공학전공

^{*2} 한국과학기술원 기계공학과 항공우주공학전공

이상의 CPU에서 포화(saturation) 현상이 발생되어 현재 개발되는 병렬 컴퓨터는 대부분 분산 메모리 형태를 갖는다. 그리고 병렬화 방법은 함수 분할(functional decomposition) 방법과 영역 분할(domain decomposition)으로 구분된다. 이러한 구분은 각각의 프로세서에 해당되는 일이 주어지는 방법에 따라 구분하는 방법이다. 본 연구에서는 분산 메모리 형태의 병렬 컴퓨터인 Cray T3E를 사용하여 계산하였으며, 영역 분할 방법을 이용하여 해석 프로그램을 병렬화하였다. 또한 자료 전달을 위해 Cray T3E의 MPI(Message Passing Interface)를 사용하였다.

2. 수치적 방법

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

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

여기서 Q 는 보존형 변수, $F(Q)$ 는 대류항이며, 다음과 같다.

$$Q = [\rho, \rho u, \rho v, \rho w, e_0]^T \quad (2)$$

$$F(Q) \cdot \vec{n} = (\vec{V} \cdot \vec{n}) \begin{bmatrix} \rho \\ \rho u \\ \rho v \\ \rho w \\ e_0 \end{bmatrix} + \begin{bmatrix} 0 \\ p n_x \\ p n_y \\ p n_z \\ 0 \end{bmatrix} \quad (3)$$

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

$$p = (\gamma - 1) \left(e_0 - \frac{1}{2} \rho (u^2 + v^2 + w^2) \right) \quad (4)$$

$$T = \gamma p / \rho \quad (5)$$

2. 1 공간차분법

Roe의 flux difference splitting을 사용하여 대류항을 계산하였으며, 각각의 셀의 경계면 k 를 통한 플럭스는 다음과 같이 표현된다.

$$F_k = \frac{1}{2} [F(Q_L) + F(Q_R) - |\tilde{A}|(Q_R - Q_L)] \quad (6)$$

여기서 Q_L 과 Q_R 은 k 면의 왼쪽과 오른쪽의

유동변수들을 나타낸다. 행렬 $|\tilde{A}|$ 는 Roe의 평균화된 유동량을 사용하여 계산하였으며, 참고 문헌 [3]에 기술된 식을 사용하였다.

셀의 경계면(face)에서 고차 정확도로 플럭스를 계산하기 위해 참고문헌 [3]에 기술된 재구성(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.

$$q_{f_{1,2}} = q_c + \frac{1}{3} \left[\frac{1}{2} (q_{n_1} + q_{n_2}) - q_{n_3} \right] \quad (7)$$

여기서 q_c 는 셀 중심에서의 값이며, $f_{1,2}$ 는 격자점(node) n_1 과 n_2 로 구성된 면이고, n_3 는 $f_{1,2}$ 와 마주보는 격자점이다. 또한 삼차원의 사면체에 대해서는 다음과 같다.

$$q_{f_{1,2,3}} = q_c + \frac{1}{4} \left[\frac{1}{3} (q_{n_1} + q_{n_2} + q_{n_3}) - q_{n_4} \right] \quad (8)$$

여기서 $f_{1,2,3}$ 는 격자점 n_1, n_2, n_3 로 구성된 면이며, n_4 는 $f_{1,2,3}$ 와 마주 보는 격자점이다.

이러한 재구성 방법에서는 셀 중심의 값들을 이용하여 격자점에서의 값을 내삽(interpolation)하는 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, 플럭스에 대해 Newton 형식의 선형화를 하면 다음과 같은 형식으로 쓸 수 있다.

$$\left[D_i + \sum_j^{n_{face}} S_{j,k} \right] \Delta Q = -R^n \quad (9)$$

여기서 $\Delta Q = Q^{n+1} - Q^n$ 을 나타낸다. 플럭스 자코비안의 대각항과 비대각항, D_i 와 $S_{j,k}$ 는 다음과 같다.

$$D_i = \frac{V_i}{\Delta t} + \sum_j^{n_{face}} S_{j,i} \quad (10)$$

$$S_{j,k} = \frac{\partial F_j}{\partial Q_k} \quad (11)$$

여기서 아래첨자 j 는 i 번째 셀의 주변 셀들을 나타낸다. 식 (11)에서 플럭스 자코비안은 Roe의 평균화된 특성치를 이용하여 계산하였으며, 참고문헌 [5]에 자세히 기술되어 있다. 식 (9)의 선형방정식은 왼쪽의 행렬에 대한 역행렬을 직접 계산하여 구할 수 있지만 이러한 방법은 많은 계산 시간과 기억용량을 필요로 한다. 본 연구에서는 이러한 문제를 해결하기 위해 Gauss-Seidel 방법을 사용하였으며, 유연화(relaxation) 과정에서의 오차를 줄이기 위해 내부 반복계산(sub-iteration)을 사용하였다[5].

3. 병렬처리

3. 1 영역 분할법

영역 분할법을 통하여 계산 영역을 분할할 경우, 가장 중요한 문제는 나누어진 부영역에서의 격자점이나 셀의 수를 같게 하여 병렬 계산 시 각각의 프로세서가 같은 시간 내에 작업을 끝낼 수 있게 하는 부하 분배(load balancing)와 각 부영역간의 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. 현재 많이 쓰이고 있는 영역 분할법으로는 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 적절한 부하 분배를 이룰 수 있음을 제시하였다. Simon[7] 등은 이러한 Spectral 방법을 다단계 기법으로 발전시켜 영역 분할 시간을 줄였다. 또한 다단계 Spectral 방법과 유사한 방법으로 미네소타 대학의 Karypis에 의해 개발된 k-way n-partitioning 방법이 있다[8]. 이 방법은 원래의 영역을 그래프 이론을 바탕으로 축소(coarsening)한 후 축소된 영역을 k-way 방법으로 분할하고, 다시 원래 영역으로 환원(un-coarsening)하는 방법이다. 또한 축소된 영

역의 확장 시 영역의 경계를 줄일 수 있는 대응 기법을 활용하여 빠른 시간 내에 작은 경계면을 가지는 영역으로 분할할 수 있는 특징이 있다. Karypis는 이러한 연구의 결과를 Metis라는 공개 프로그램으로 만들어 배포하고 있으며, 본 연구에서는 Metis 라이브러리(library)를 사용하여 영역 분할을 수행하였다.

3. 2 해석 프로그램의 병렬 처리

각각의 프로세서간의 자료 전달을 위해서 MPI 라이브러리가 사용되었다. 병렬 처리 시 자료의 전달은 계산 시간 이외의 overhead가 된다. 따라서 자료 전달 함수의 자료 전달 방식을 이해하고 이를 최적화 할 수 있어야 자료전달 시간을 최소로 줄일 수 있다. 참고문헌 [9]에서는 Cray T3E의 자료전달 특성에 대해 연구하여, 많은 자료를 묶어서 전달하는 방식이 효율적임을 제시하였으며, 본 연구에서도 이러한 방식을 사용하였다.

본 연구의 해석 프로그램의 특성상 계산 영역 경계면에서 3가지의 자료 전달이 필요하다. 첫째로 경계면에서 고차로 대류항을 계산하기 위해 경계면 값에 대한 자료 전달이 필요하다. 식 (6)을 보면 계산 영역 경계면에 놓여 있는 격자점에서, 인접한 프로세서에서 계산된 격자점에 대한 자료 전달을 통해서 고차로 플럭스가 계산될 수 있음을 알 수 있다. 두 번째로는 경계면에 접한 셀 내에서 재구성을 하기 위해 계산 영역 경계면에 놓여있는 격자점에 대한 자료전달이 필요하다. 격자점에 대한 계산은 격자점을 둘러싸고 있는 셀 중심의 값들을 사용하여 구하므로, 격자점이 계산 영역 경계면에 있다면 이 격자점에 대한 외부 프로세서에서의 영향을 고려하여 구해야 한다. 세 번째로 경계면에 접한 셀에서 플럭스 자코비안을 계산하기 위해 셀 중심 값에 대한 전달이 필요하다.

이에 더하여 비정상 유동의 계산 시, 계산 경계면에서 정확한 시간 전진을 위하여, 내부 반복계산을 통하여 매트릭스를 계산하는 동안에 새로 계산된 값(updated value)의 전달이 필요하다. 자코비 반복계산을 수행한다면 한 번의

스윕(sweep)이 끝난 후 새로 계산된 값의 전달로 충분하지만, 본 연구에서와 같이 Gauss-Seidel 방식에서는 새로 계산되어짐과 동시에 자료의 전달을 필요로 한다. 하지만 이러한 새로 계산되어짐과 동시에 자료를 전달하는 방법의 구현은 기술적으로 난해하고, 자료가 하나씩 전달되는 특성을 보이므로 자료전달 효율이 매우 낮아지게 된다[9]. 이러한 이유로 본 연구에서는 자코비 반복계산의 경우와 같이 한 번의 스윕이 끝난 후 자료 전달이 이루어지도록 하였다. 이러한 적용은 계산 영역 경계면에서 자코비 반복계산의 성격을 지니게 되며, 수렴성의 저하 현상을 일으킬 수 있다. 하지만 본 연구에서는 이러한 수렴성의 저하 현상은 경험하지 못하였으며, 이는 주로 계산 영역 경계면의 양이 내부 영역에 비해 상대적으로 매우 작기 때문인 것으로 판단된다. 만약에 수렴성의 저하 현상이 발생하게 된다면, 내부 반복계산의 수를 증가시켜 해결할 수 있다.

4. 결과 및 고찰

4. 1. 2차원 유동에 대한 해석

계산에 사용된 격자는 advancing front 방법을 사용하여 생성하였다. 셀 수는 8,915개, 격자점의 개수는 4,560개이고, 이 중에서 131개가 익형 표면에 위치한다. Fig. 1은 k-way n-partitioning 방법을 이용하여 NACA 0012 익형 주위의 격자를 8개의 영역으로 분할한 결과이다. Tabel 1은 분할된 격자에 대한 각각의 셀(cell), 격자점(node), 격자면(face)에 대한 개수와 계산 영역 경계면(computational boundary)에서의 셀과 격자점의 개수를 나타내고 있다. 일반적으로 어떠한 영역 분할 기법을 사용하더라도 셀 수와 격자점 수, 그리고 격자면의 수를 모두 같게 할 수는 없다. 이러한 이유로 본 연구에서 사용되는 셀 중심 기법에서는 셀 수의 부하 분배에 초점을 맞추었다. Table 1을 보면 셀 수에 대한 부하 분배는 매우 잘 이루어졌음을 확인할 수 있다. 또한 계산 영역 경계면에서

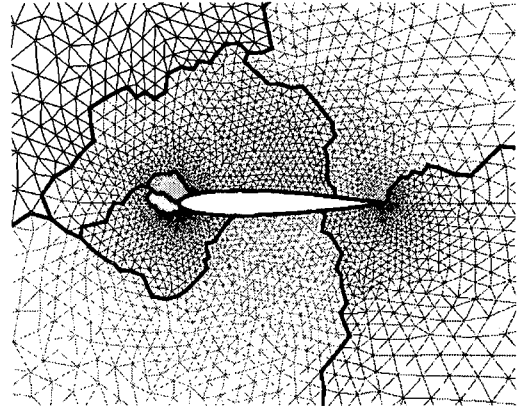


Fig. 1 8-way decomposition around a NACA 0012 airfoil

Table 1 8-way domain decomposition data (C-: Computational boundary)

Domain number	# of Cell	# of C-Cell	# 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

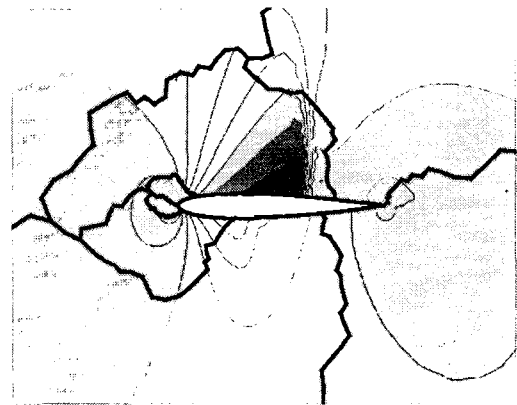


Fig. 2 Mach number contour for 8-way decomposition of domain

셀 수 등은 자료 전달량의 정도를 나타내 주며, 8개의 영역으로 분할된 경우 내부 영역에 비해 상대적으로 매우 적음을 볼 수 있다. 내부 영역에 비해 계산 영역 경계면이 커질 수록 단일 영역에 대해 계산하는 경우보다 병렬 계산 시 수렴성이 저하된다. 이러한 현상은 계산 영역 경계면에서의 값들이 이전 시간 단계의 값에 대한 자료 전달에 의해 계산되어지기 때문에 발생한다. Fig. 2는 8개의 영역으로 나뉘어진 경우에 대해 마하수가 0.8, 받음각이 1.25° 인 비점성 유동에 대해 계산된 등마하선도이다. 영역의 경계면에서 부드러운 연결을 보이고 있으며, 이를 통해 각각의 프로세서에서의 자료 전달이 적절히 이루어지고 있음을 알 수 있다. Fig. 3은 하나의 프로세서를 사용한 결과와 8개의 프로세서를 사용한 결과를 비교한 그림이며, 계산 결과가 거의 일치함을 볼 수 있다. Fig. 4는 계산 영역을 2, 4, 8, 및 16 개로 각각 분할했을 시의 speed up 곡선이다. Speed up은 1개의 프로세서를 사용했을 경우에 걸린 시간을 병렬 계산 시 걸린 시간으로 나누어 계산하였다. 적은 셀 수로 인하여 8개 이상의 프로세서를 사용할 경우 프로세서간의 자료 교환에 드는 시간 (communication time)이 전체 시간에서 차지하는 비율이 높아져서, 그 효율이 이상적인 경우보다 크게 밀릴게 된다. 16개의 프로세서를 사용했을 때의 효율은 76% 정도이다.

4. 2 3차원 유동에 대한 해석

3차원의 비점성 유동에 대한 병렬화 기법의 검증을 위하여 F-5 날개를 선택하였다. 유동 조건은 마하수 0.9, 받음각이 0° 이다. Fig. 5는 8개의 영역으로 나뉘어진 F-5 날개주위의 표면 격자이며, Fig. 6은 8개로 나뉘어진 격자계에 대해 계산된 등마하선도 이다. 그림에서 볼 수 있듯이 2차원의 경우와 마찬가지로 영역의 경계면에서 부드러운 연결을 보이고 있다. Fig. 7과 Fig. 8은 스펠의 위치에 따라 실험치와 계산된 결과를 나타낸 그림이며, 대체적으로 실험결과와 잘 일치하고 있다. Fig. 9는 영역의 수에 따라 반복계산에 대한 수렴도를 나타낸 그림이다. 영역의 수가 증가할수록 수렴성의 저하현상을 볼 수 있다. 본 연구에서 정상 상태의 해를

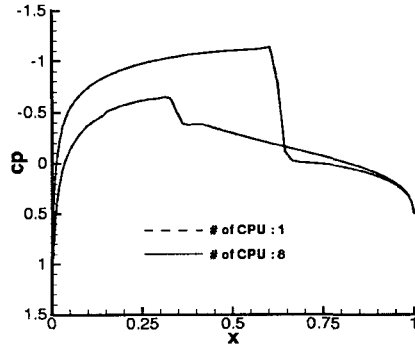


Fig. 3 Comparison of surface pressure distribution with single and 8 CPU

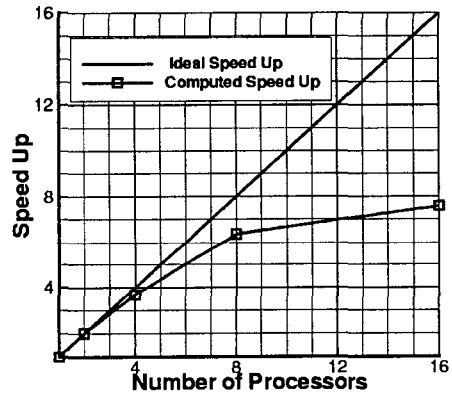


Fig. 4 Parallel efficiency for 2-D case

계산하기 위해서는 자료 전달에 드는 시간을 줄이기 위해 내부 반복계산 동안에 자료 전달을 수행하지 않는다. 이러한 방법으로 인하여 영역의 수가 증가하여 계산 경계면이 상대적으로 많아지면 수렴성이 저하된다. Fig. 10은 전체 프로세서에 대해 합쳐진 자료 전달 시간 (communication time)과 계산 시간(computation time) 그리고 전체 걸린 시간(total elapsed time)을 영역의 수에 따라 나타낸 그림이다. 계산은 2, 4, 8, 16, 32, 64개의 프로세서에 대해 수행하였다. 자료전달 시간 측면에서, 영역의 수가 16개까지는 증가하다가 그 이후로는 거의 비슷해진다. 영역의 수가 증가하면 계산 경계면은 계속 증가하며, 이에 따라 자료전달량도 증가하게 된다. 하지만 Cray T3E에서 실제적인 자료

전달(data transport)은 다른 시간에 비해 상대적으로 매우 빠르게 일어나며, 대부분의 자료 전달 시간은 각각의 프로세서가 계산을 끝낸 후 동기화(synchronization) 되기 위해 대기하는 시간이 차지한다. 이러한 원인에 의해 영역의 수의 증가에 의한 자료 전달량의 증가는 자료 전달 시간에 많은 부분을 차지하지 못한다. 그러므로 영역 수가 적정수 이상이 되면 자료 전달 시간은 증가하지 않고 비슷해진다. 또한 영역의 수가 많더라도 부하 분배가 잘 되어 동기화 되기 위한 시간이 작다면 자료 전달 시간은 오히려 감소하게 된다. 실제로 32개 영역에서는 16개 영역에서 보다 자료전달 시간이 약간 감소하였다. 계산 시간 측면에서는 영역의 수가 증가

함에 따라 계산 시간이 감소함을 볼 수 있다. 나뉘어진 영역의 수가 증가하여도 전체적인 계산량은 항상 같다. 하지만 영역의 수가 증가함에 따라 각각의 프로세서의 캐쉬 메모리의 효율이 상대적으로 높아지게 되어, 계산 시간(computation time)이 감소하게 된다[10], [11]. 영역의 수가 증가함에 따라 계산 시간은 감소하고 자료 전달 시간은 거의 일정하므로 전체적으로 걸린 시간은 감소하게 된다. Fig. 11은 이러한 결과를 이용하여 이상적인 병렬화 speed up 과 본 연구에서 개발된 병렬 프로그램의 speed up을 비교한 그림이다. 영역의 수가 증가함에 따라 전체적인 계산시간이 감소하게 되어 이상적인 병렬화 효율보다 높은 것을 볼 수 있다 [10].

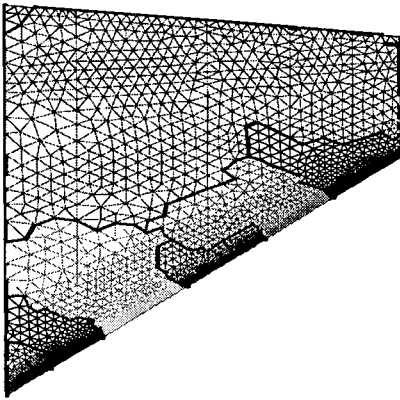


Fig. 5 8-way decomposition around a F-5 wing

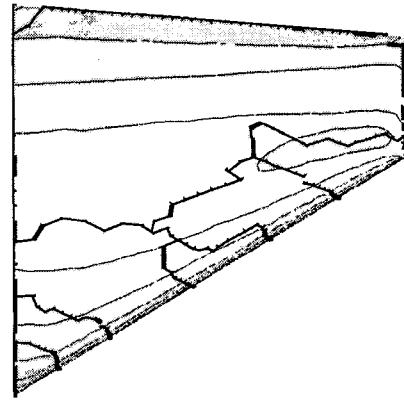


Fig. 6 Mach number contour for 8-way decomposition of domain

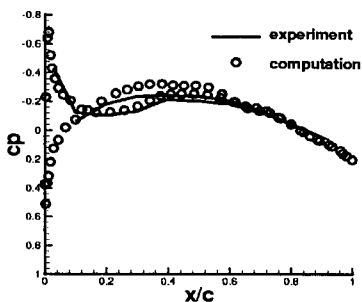


Fig. 7 Surface pressure distribution for a F-5 wing (51% span)

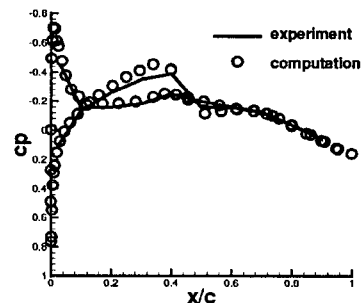


Fig. 8 Surface pressure distribution for a F-5 wing (88% span)

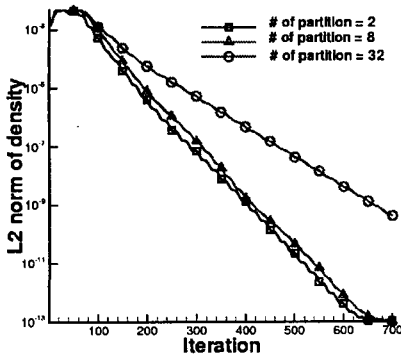


Fig. 9 Convergence history

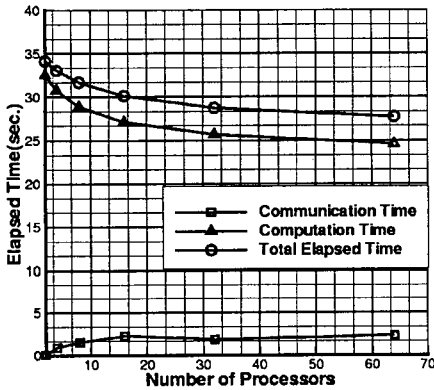


Fig. 10 Data communication time, computing time and total elapsed time

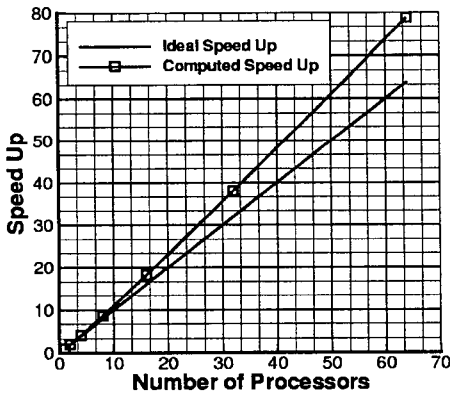


Fig. 11 Parallel efficiency for 3-D steady flow case

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. 12는 받음각의 변화에 따른 양력 계수에 대한 곡선이며, 4개의 프로세서를 사용하여 계산한 결과이다. Fig. 13은 비정상 유동에 대한 speed up 곡선이다. 앞에서 설명한 바와 같이 비정상 유동을 정확히 계산하기 위해서는 내부 반복 계산 동안 자료 전달이 이루어져야 한다. 이러한 자료 전달에 의해 자료 전달 시간이 영역이 증가함에 따라 급속히 증가하여, 전체적인 병렬화 효율이 정상 상태의 경우보다 감소하였다.

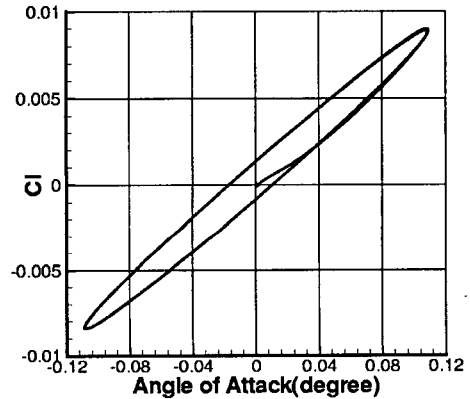


Fig. 12 Lift history for a F-5 wing

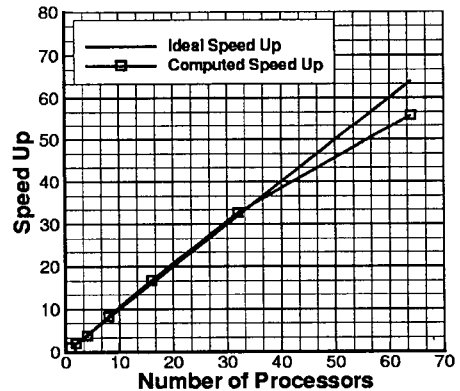


Fig. 13 Parallel efficiency for 3-D unsteady flow case

5. 결 론

본 연구에서는 비정렬 격자계에서의 내재적 방법을 이용한 2차원과 3차원 유동해석 기법의 병렬화에 대한 연구를 수행하였다. 영역 분할 기법으로 Karypis가 제안한 k-way n-partitioning 방법을 사용하였으며, 이 방법을 사용하여 본 연구에서 사용된 셀 중심 기법에 대해 적절한 부하 분배를 얻을 수 있었다. 개발된 병렬 유동 해석 기법을 검증하기 위해 2차원과 3차원 비점성 유동에 대해 계산하였으며, 병렬 효율 및 계산 정확도에 대해서 좋은 결과를 확인할 수 있었다. 또한 비정상 유동에 대해서도 계산을 수행하여 개발된 기법이 시간 정확도를 유지할 수 있음을 확인하였다.

후 기

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

참고문헌

- [1] Venkatakrisnan, 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 격자에서의 내재적 기법을 이용한 난류 유동 계산," 한국전산유체공학회 추계학술대회, 1997.
- [6] Venkatakrisnan, 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] Banard, S. and H. Simon., "A Fast Multilevel Implementation of Recursive Spectral Bisection for Partitioning Unstructured Problems," RNR Report 94-004, NASA, 1994.
- [8] Karypis, G., <http://www.cs.umn.edu/~karypis/metis>
- [9] 김성호, "비정렬 격자를 사용한 병렬 전산 유체 해석 기법에 관한 연구," 한국과학기술원 항공우주공학과, 박사학위 논문, 1999.
- [10] Strang, W. Z., Tomaro, R. F., and Grismer, M. J., "The Defining Methods of Cobalt₆₀ : A Parallel, Implicit, Unstructured Euler/Navier-Stokes Flow Solver," AIAA Paper 99-0786, 1999.
- [11] 옥호남, 박승오, "혼합격자를 이용한 2차원 난류 유동장 해석 프로그램의 병렬화," 한국전산유체공학회 추계학술대회 논문집, pp. 115-126, 1999.