Chimera 기법의 병렬처리에 관한 연구
A Study of Parallel Implementations of the Chimera Method

조 금 원* 권 장 혁† 이 승 수‡
K.W. Cho J.H. Kwon S.Lee

The development of a parallelized aerodynamic simulation process involving moving bodies is presented. The implementation of this process is demonstrated using a fully systemized Chimera methodology for steady and unsteady problems. This methodology consist of a Chimera hole-cutting, a new cut-paste algorithm for optimal mesh interface generation and a two-step search method for donor cell identification. It is fully automated and requires minimal user input. All procedures of the Chimera technique are parallelized on the Cray T3E using the MPI library. Two and three-dimensional examples are chosen to demonstate the effectiveness and parallel performance of this procedure.

1 서 론

로터 블레이드와 헬리콥터 동체의 간섭현상 해석, 비행중인 항공기로부터 연료탱크의 분리, 다단계 로켓의 분리 등 물체간의 상대운동이 있는 문제는 전산유체역학(Computational Fluid Dynamics : CFD)이 극복해야할 중요한 문제 중의 하나이며 이에 대한 많은 연구가 진행되고 있다[1][2]. 정적인 문제가 움직이는 동적인 문제로 해석대상이 바뀔 경우, 격자계는 해석대상의 움직임에 따라 적응되어져야 한다. 이는 주어진 격자계의 초기 위상적 특징을 깨뜨리는 일로써 단순히 격자계의 회전과 병진운동만으로 만들어지지 않는다. 따라서 격자계는 해석대상과 독립적으로 취급될 수 있는 방법이 필요하게 된다. 이를 만족하는 대표적인 방법이 비정렬 격자기법과 Chimera 격자기법[3]이다. 비정렬 격자기법은 자료구조 및 점성유동의 제한성과 움직이는 격자계에 대해 전체 격자를 새롭게 생성해야 하는 제약점 때문에 그 응용범위가 제한되고 있다. 반면 정렬격자계를 사용한 Chimera 기법은 정적 격자계 구성시 비정렬 격자계에 비해 유연성이 부족하나 정렬격자계의 특징을 유지하고 움직이는 매 시간 단계에서 격자를 새롭게 구성하지 않으므로 적은 삽간량을 갖게 되는 장점을 지니고 있다.

Steger에 의해 처음 제안된 Chimera 기법[3]은 현재 다양한 알고리즘들을 이용하여 일반 사용자들이 쉽게 접근 할 수 있는 코드들로 발전하였다. 대표적인 코드들로 PEGASUS[4], BEGGAR Code[5], DCF3D[6] 등이 있다. 일반적인 Chimera 격자구성 방법은 Chimera 홀 절단(Hole-Cutting), 삽간점 찾기 (Donor Cell Identification)와 영역 연결(Domain Connectivity) 과정으로 구성된다. 고체 경계면 내에 포함된 사용되지 않은 점을 제거하는 과정인 Chimera 홀 절단은 벡터 내적을 이용한 방법[3], 교차횟수를 이용하는 방법[7] 그리고 Hole-Map[8]방법 등이 사용되고 있다. 이중에서 Hole-Map방법은 벡터 내적을 사용하지 않고 구성된 Cartesian 격자계의 값의 비교만으로 홀점을 찾아내는 방법으로 벡터화가 가능하다. 벡터 교차 횟수를 이용한 PEGASUS 코드는 홀 경계 및 홀 격자들을 정해 주어야하는 어려움이 있으며 해석적인 함

*학생회원, 한국과학기술원 항공우주공학과
†정회원, 한국과학기술원 항공우주공학과
‡정회원, 국방과학 연구소 3-1-2

수를 이용하는 DCF3D 코드는 계산시간이 빠른 반면 일반성이 부족한 단점을 가지고 있다. 본 논문에서는 해석자(solver)에서 사용되는 NPBC(Non-Penentrable Boundary Condition)들을 이용하여 홀 경계를 자동적으로 설정한 후 이를 Hole-Map방법에 적용하여 일반성이 있고 효율적인 홀 절단방법으로 사용하였다[8].

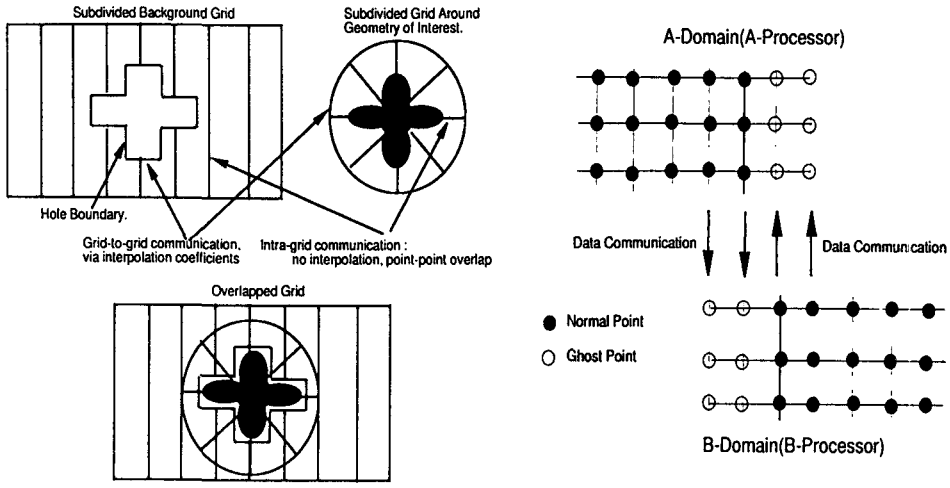
삽간점 찾기 방법으로는 tree구조 방법[5], inverse map 방법[6]과 stencil-walk[9] 방법 등이 대표적으로 사용되고 있다. Stencil-walk방법은 삽간점과 후보 donor점 사이에 단순히 거리 비교를 통해 가장 가까운 점을 찾는 방법으로 $O(n)$의 계산 시간이 필요하다. 이를 정렬격자계에 적용시킬 경우 고체 경계면, 특이선 및 branch-cut 등의 경계면을 자동적으로 뛰어넘을 수 있는 추가적인 조건이 요구되나 tree 구조($O(nlogn)$)보다 빠르며 추가적인 자료구조를 요구하지 않는 장점을 가지고 있다. 이들 방법은 삽간점에 대해 가장 가까운 격자점의 정보만을 제공할 뿐 실제적으로 그 점이 donor요소로 사용될 수 있는지에 대한 판별 기준을 제공하지 못한다. 따라서 추가적인 과정이 필요하게 되며 이를 in/out 시험이라 한다. 본 논문에서는 stencil-walk방법을 사용하여 donor 후보점들을 구하고 in/out 시험으로 iso-parametric 좌표에서 Newton-Raphson방법을 사용하는 구배 찾기(gradient search)방법을 사용하였다. 이때 구배 찾기방법은 대개 2-3회 반복계산이 소요된다. 본 논문에서 stencil-walk과 구배 찾기방법을 결합하여 2단계 찾기(two-step search)방법이라 명하였다.

독립적으로 구성된 Chimera 격자계간에 독립변수들의 전달은 영역연결 과정을 통해 이루어지며 일반적인 방법으로 비보존적인 bi(2차원) 또는 tri(3차원)-linear 삽간 방법이 사용되고 있다. 이는 구배가 약한 부분에서 삽간이 이루어질 경우 보존적인 삽간만큼의 결과를 얻을 수 있다고 알려져 있다[10]. 그러나 일반적인 형상에 대해 유동 구배가 상대적으로 약한 부분에서 삽간이 이루어지도록 Chimera 격자계를 구성하는 것은 쉬운 일이 아니다. 본 논문에서는 이를 가능하도록 하는 cut-paste 알고리즘을 고안하여 적용했다[8]. 이는 유동 구배가 큰 고체 경계면에서 멀리 떨어져 삽간 영역이 구성되도록 하며 다중 물체가 중첩된 경우 각 물체의 중앙 부분에 삽간점이 위치하도록 해준다. 이 알고리즘은 2차원 및 3차원 Eglin, TER의 예제에 적용되었으며[2] 그 타당성이 검증되었다. 본 논문에서는 2층의 삽간 영역을 구성하여 삽간영역에서 2차의 정확도를 유지하도록 하였다.

본 논문은 위에서 언급된 Chimera 기법의 병렬처리에 대한 연구를 수행하였다. 다중블럭(multi-block)을 포함하는 해석자 및 Chimera 격자기법의 모든 단계 - Chimera 홀 절단, 삽간점 찾기 및 영역연결 - 가 병렬화 되었다. 기존의 Chimera 격자계 코드에 영역 분할된 격자에 대응하는 프로세서와 찾기 과정에 사용되는 수정된 후보 격자계 입력 목록이 입력조건으로 요구된다. 2차원 상대운동과 3차원 정상상태 유동을 Cray T3E을 이용하여 해석하였으며 Cray C90에서의 결과와 비교하였다.

2 Chimera 기법의 병렬처리

Chimera기법을 이용하는 코드들의 병렬화는 매우 최근에 이루어지고 있다. BREAKUP 코드[11]는 PEGASUS 코드의 출력 결과를 해석자 병렬효율을 높이기 위한 전처리 단계로 개발되었으며 병렬화된 BEGGAR[12] 코드는 병렬 프로세서를 FE(Front-End)와 BE(Back-End)로 구분한 후 Chimera 격자계 구성은 FE의 단일 프로세서를 이용하고 해석자는 BE의 다중 프로세서를 이용하는 해석 방법을 취하고 있다. 이들 두 방법은 Chimera 기법을 직접적으로 병렬화한 방법이 아니다. OVERFLOW와 결합된 DCF3D 코드[13]는 해석자인 OVERFLOW 뿐 아니라 Chimera 격자계 구성 코드인 DCF3D 코드를 직접 병렬화한 방법이다. DCF3D의 경우 해석 함수를 사용하여 홀 절단을 행하며 일반적인 영역연결 및 교차 영역구성 방법을 사용하고 있다. 본 논문의 코드는 DCF3D와 같이 해석자와 Chimera 격자계의 병렬처리를 동시에 수행하였으며, Chimera 격자계 구성시 일반성을 높이기 위하여 해석자의 NPBC를 이용, 홀 경계



(a) Grid-to-grid 및 intra-grid communication (b) Data Communication Technique

그림[1] Parallel Data Communication

를 구성하고 Hole-Map방법을 적용하여 홀 절단 과정을 수행한다. 또한 주어진 격자계에서 해의 질을 높일 수 있는 cut-paste 알고리즘과 2단계 찾기 과정을 적용하고 있다.

Chimera 기법의 병렬화 과정은 영역분할과 각 영역에서 계산된 값들을 영역경계를 통해 전달해 주는 과정으로 분리될 수 있다. 전 계산과정 동안 각 프로세서에 걸리는 부하를 동일하게 유지시켜 주는 과정을 dynamic load balancing 이라 하며 단순히 영역을 균등히 분할시켜 프로세서에 할당시켜 주는 과정을 static load balancing이라 한다. 본 논문에서는 계산 초기에 영역을 적절히 분할하여 각 프로세서가 비슷한 양의 일을 할 수 있도록 하였다. Chimera 기법의 병렬화는 해석자와 격자계 구성의 두 분야에 대한 load balancing이 필요하다. 그러나 격자계 구성 시간은 실제 해석자의 계산 시간에 비해 대략 10% 정도 차지하게 되므로 load balancing은 해석자를 기준으로 정해지는 것이 타당하다. 따라서 삽간갯수의 균등한 분포에 따라 load balancing을 하지 않고 전체 격자수의 균등한 분포를 이루도록 하였다.

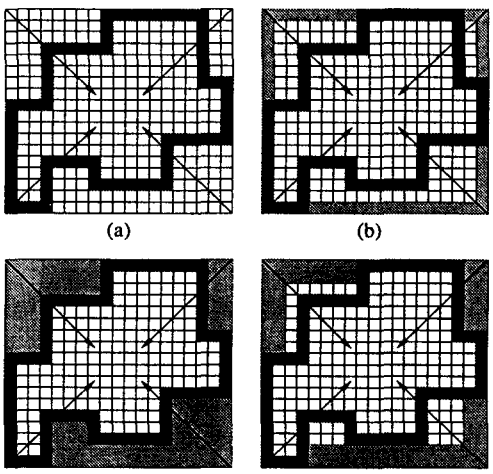
그림[1](a)와 같이 Chimera 기법의 병렬처리는 두가지 자료 전달과정이 필요하게 된다. 즉, 내부격자 간의 자료전달 (intra-grid communication)과 격자계 간 자료전달 (grid-to-grid communication)로 분류될 수 있다. 내부격자 간 자료전달은 단일프로세서의 다중블럭 경계에 해당되는 것으로 격자점 단위로 경계가 일치해 있다. 이에 대한 프로세서 간의 정보전달은 삽간계수에 대한 정보 없이 경계점 위치만을 이용하여 변수들을 교환해 주면 된다. 전체 정보전달 시간에 대해 내부격자 간 자료 전달시간은 적으며 해석자만을 병렬처리할 경우의 필요한 정보전달 과정이다. 본 논문에서는 그림[1](b)와 같이 2층의 추가 요소를 만들어 이들 사이에서 자료교환이 이루어 지도록 하였으며 이는 단일프로세서의 다중블럭 경계처리와 동일하게 사용된다. 반면, 격자계 간 자료전달은 그림[1](a)에서와 같이 중첩된 격자계 간 정보전달을 행하는 과정이다. 이는 Chimera 격자 기법에 필요한 정보전달 과정으로 삽간계수 및 삽간위치에 대한 정보의 전달을 필요로 한다. 격자계 간 자료전달은 내부격자간의 자료전달에 비해 계산시간이 많이 요구되며, 따라서 계산시간을 줄이기 위해 한번에 각 영역에 대한 유동변수의 전달이 이루어지도록 하는 과정이 필요하다.

2.1 해석자의 병렬처리

정상 및 비정상 유동장 해석이 가능한 2차원 및 3차원 Euler 해석자[14]가 병렬화되었다. 본 논문에서 사용된 해석자는 Chimera 격자계 및 다중블럭 격자계를 동시에 사용할 수 있어, 해석자의 수정없이 복잡한 유동장을 해석할 수 있다. 사용된 수치기법은 공간 이산화 방법으로 유한 체적법을 사용한 Roe의 FDS(Flux Difference Scheme)와 2차 이상의 공간 이산화 오차를 갖도록 MUSCL(Monotone Upwind Scheme for Conservation Law) 방법을 사용하였다. Minmod 또는 Van Alabada 제한자(Limiter)를 사용하여 TVD(Total Variation Diminishing) 성질을 갖도록 하였으며 비정상 유동을 해석하기 위하여 정상류 해석 프로그램에 이중시간 적분법(Dual Time Stepping)을 적용하였다. 이중시간 적분법은 내재적 기법이 갖는 선형화 오차와 ADI(Alternate Direction Implicit)방법의 근사 인수분해에 의한 오차를 제거하기 위해 개발된 방법이다. 이는 임의의 시간항을 더하여 이 시간항으로 ADI방법의 내재적 적분방법을 적용시킨 것으로 시간축에 대해 2차의 정확도를 가지고 있다. MIMD(Multiple Instruction Multiple Data) 형태의 분산 메모리를 갖는 컴퓨터인 Cray T3E와 MPI 라이브러리를 사용하여 병렬화하였으며 Blocking SEND/RECV와 Non-Blocking SEND/RECV를 조합하여 사용하였고 격자형태에 무관하게 자료전송이 이루어지도록 하였다.

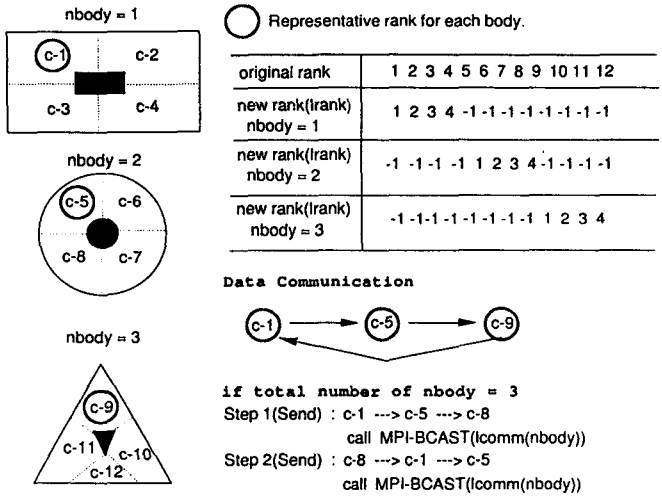
2.2 병렬 Chimera 홀 절단

단일 프로세서에 Hole-Map을 적용한 Chimera 홀 절단 과정은 그림[2]에서와 같이 홀 경계를 사이에 두고 내부와 외부를 구분할 수 있는 Cartesian 격자를 구성해 주는 과정으로 자세한 알고리즘은 참고문헌[8]에 잘 나타나 있다. 이의 과정을 병렬화 시키기 위해 본 논문에서는 다음과 같은 과정을 사용하였다.



그림[2] Hole-Map Algorithm

Chimera 격자계의 각 격자 그룹은 하나의 몸체(body)를 형성하고 있게 된다. 예로 에어포일에 플랩이 달려 있을 경우 에어포일을 형성하는 격자계와 플랩을 형성하는 격자계가 각각 몸체로 구분되는 것으로 이해될 수 있다. 따라서 Chimera 홀 절단을 행할 경우 각 몸체에 해당하는 격자계는 자기자신에 의해 잘리지 않게 된다. 즉, 이들 몸체에 의해 잘리는 격자계는 다른 몸체에 속한 격자계가 된다. 따라서 정보의 효율적인 전달을 위해 각 몸체에 해당하는 격자사이에는 구별된 RANK를 가지는 것이 유리하다. 그림[3]은 임의의 몸체에 대해, 원래 그룹과 새롭게 생성된 그룹의 RANK를 나타내며 각 몸체에 해당하는 RANK는 양의 값을 가짐을 볼 수 있다. 새로운 그룹과 해당하는 RANK를 생성하기 위하여 MPI-COMM-GROUP,



그림[3] Parallel Chimera Hole-Cutting

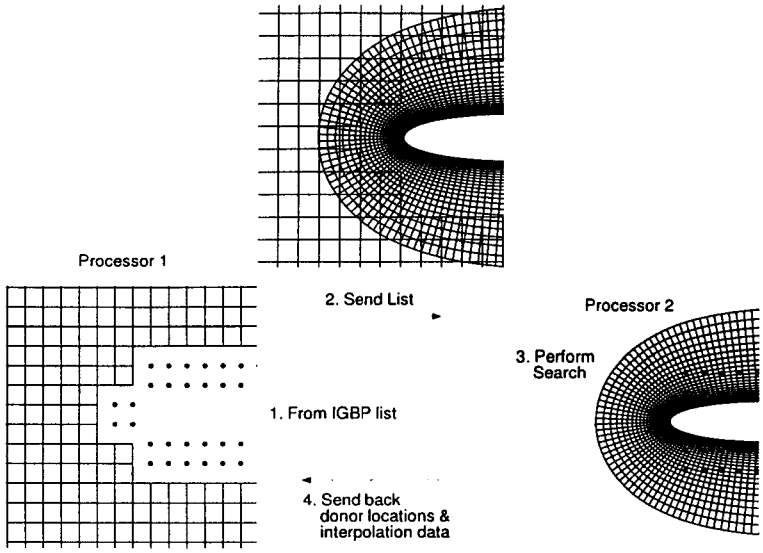
MPI-GROUP-INCL, MPI-COMM-CREATE와 MPI-GROUP-RANK를 사용하였다. 이렇게 정의된 새로운 그룹에 대해 해당하는 홀 경계점들을 모아 단일 프로세서의 그림[1](a)와 같은 초기 Hole-Map을 만들게 되며 최종적으로 그림[2](d)의 Hole-Map이 각 프로세서에서 만들어지게 된다.

두번째의 과정은 Hole-Map을 이용하여 불필요한 격자점을 제거하는 것이다. 이때 상대편의 격자점을 제거하기 위해서는 자신이 만든 Hole-Map을 상대편 그룹의 프로세서들에게 전달해 주어야 한다. 만약 한 그룹(또는 몸체)의 모든 프로세서가 상대편 프로세서의 모두에게로 자료를 전달한다면 몸체별로 서로 다른 프로세서 갯수를 갖고 있는 상황에서는 불필요한 작업이 요구되며 계산시간이 증가하게 된다. 따라서 본 논문에서는 각 그룹에 대해 대표(representitive) RANK를 정의해 주고 이들 프로세서 사이에서 Hole-Map 자료의 전달이 이루어지게 하였다. 즉, 그림[3]에서 몸체는 3개이고 각 몸체가 4개의 영역으로 나뉘어 있을 경우 C-1, C-5 그리고 C-9를 대표 RANK로 정의하게 된다. 이들 사이에 정보 전달은 그림에서와 같이 만약 몸체가 3개이면 2번의 자료전달을 하게 된다. 각 대표 프로세서가 받은 Hole-Map은 앞에서 새롭게 정의된 그룹별로 MPI-BCAST 라이브러리를 사용하여 전달하게 된다. 여기서 각 단계에서 프로세서들이 받은 Hole-Map은 자신이 속한 몸체의 Hole-Map이 아니라 상대편 몸체가 가지고 있던 Hole-Map이 된다. 따라서 각 프로세서는 자신이 가지고 있는 격자점들을 대입하여 홀점과 정상점임을 판별하게 된다. 각각의 자료전송은 계산속도를 증가시키고 추가적인 배열이 필요치 않도록 Non-Blocking SEND/RECV를 사용하였으며 Chimera 홀 절단 과정은 전체 Chimera 격자계 구성시간의 10-15% 정도 소요된다.

2.3 병렬 영역연결 기법

격자계간 종속 변수의 정보전달을 행하는 영역연결 기법은 Chimera 기법의 병렬과정에서 가장 시간이 많이 소요되는 부분이다. 본 논문에서는 stencil-walk과 구배 찾기를 결합한 2단계 찾기 과정을 이용하여 주어진 삽간점(InterGrid Boundary Point : IGBP)들에 대해 donor점들을 구하였다. 영역연결 과정을 효율적으로 수행하기 위해 본 논문에서는 후보격자계를 나타내는 사용자 목록을 정의하였다. 영역연결 과정은 다음과 같다(그림[4]).

- 각 프로세서는 자신에 속한 격자계에서 삽간점들을 구한다(1 단계).



그림[4] Parallel Search Procedure

- 각 프로세서는 구해진 삽간점들을 사용자 목록에 따라 상대편 프로세서로 보낸다(2 단계).
- 각 프로세서는 삽간점에 대해 2단계 찾기를 수행하며 찾아진 donor 요소의 상태를 확인한다(3 단계).
- 각 프로세서는 donor 요소의 인덱스(index)들과 삽간 계수를 원 프로세서로 돌려보낸다(4 단계).
- Donor 요소가 모두 정상점들로 구성되어 있지 않은 삽간점을 포함한 프로세서들은 사용자 목록의 다음 프로세서로 남아있는 삽간점들을 보낸다(2단계).

위의 과정은 상대운동이 있는 비정상 문제에 대해 사용자 간섭없이 해석을 수행할 수 있으며 또한 삽간 영역을 최적화하기 위한 cut-paste 알고리즘의 병렬화가 포함되어 있다. Cut-paste 알고리즘에 대한 자세한 내용은 참고문헌[8]에 나타나 있다.

3 결과 및 고찰

병렬 효율 및 알고리즘의 정확성을 검증하기 위해 3자유도를 갖고 날개로 부터 분리되는 2차원 Eglin 날개/스토어 예제를 해석하였다. Cray T3E를 사용하였으며 Cray C90에서의 결과와 비교하였다. 3차원 예제로 ONERA M6 날개에 대해 해석자를 검증하고 ONERA M6 날개와 타원형 스토어로 구성된 Chimera 격자계에 대해 해석하였다.

3.1 2차원 날개로부터 분리되는 스토어

사용된 격자계는 C-유형 에어포일 격자계(189 × 31), O-유형 스토어 격자계(141 × 25)와 H-유형의 배경 격자(148 × 145)이다. 자유흐름 마하수는 0.6이며 받음각은 0도이다. 무차원 시간 변수는 $t^* = ta/c$이며 a는 음속, c는 에어포일 코드길이를 나타낸다. 스토어의 질량과 관성 모멘트는 공력 및 모멘트보다 크게 선택되었다($m/\rho_\infty L^3 = 10, I_y/\rho_\infty L^5 = 10$). 여기서, m은 스토어의 질량, ρ_∞는 자유흐름 밀도, I_y는 관성모멘트 그리고 L은 스토어의 길이를 나타낸다.

그림[5]에서 3개의 프로세서와 12개의 프로세서를 사용하여 구성된 Chimera 격자계를 나타내며 삽간점을 제외하고 정상점들만을 나타냈다. 구성된 격자계가 최적의 삽간영역을 가짐을 보이기 위해 경계선들을 나타냈으며 병렬화에 사용된 영역분할 경계도 그림[5](b)에 나타내었다. 3개의 프로세서를 사용할 경우 각 격자계는 각각 한개의 프로세서에서 실행되게 된다. 이때 각 프로세서들에 대한 격자갯수 비(최소/최대)는 0.164이다. 그림[6]에 등압력 선도를 도시하였으며 최소값이 0.25, 최대값이 0.85 그리고 증분치가 0.04이다. 그림[7], [8]은 무차원 시간이 5.45와 12.45일때의 등압력 선도를 나타내며 Cray C90을 사용하여 해석한 결과와 잘 일치함을 보이고 있다. 그림[9]는 스토어의 궤적과 모멘트 계수들을 나타내고 있으며 Cray C90의 결과와 거의 일치하는 것을 볼 수 있다. 초기에 스토어는 자중과 스틱힘에 의해 머리를 숙이는 운동을 진행하다 공력효과가 커짐에따라 머리를 드는 운동을 하고 있음을 알 수 있다. 이는 2차원 물체에 대한 타당한 결과이다. 그림[10](a)는 자동 모드(cut-paste를 적용한 방법), 수동모드(일반적인 Chimera방법) 그리고 해석자만의 병렬효율을 나타낸다. Chimera 격자 구성의 병렬 효율은 삽간점 갯수가 균등히 분포되었을 때 효율적이나, 영역 분할이 해석자 중심으로 구성된 경우에 프로세서들이 포함하고 있는 삽간점 분포는 매우 차이가 나게 되며, 이것은 해석자만을 병렬처리하는 경우에 비해 자료교환을 증가시키고 이의 영향으로 병렬효율을 저하된다. 자동모드를 사용한 경우 수동모드에 비해 효율이 저하되는 이유는 자동모드가 반복과정을 포함하고 있으며 그 과정동안 자료교환을 하기 때문이다. 그림[10](b)에 각 모드에 대한 전체 계산시간 및 Cray C90에 대한 비교 속도를 나타내었다. 12개의 프로세서를 사용할 경우 전체 계산시간은 3개의 프로세서를 사용한 경우보다 약 2.6배(수동모드)와 2.2배(자동모드) 그리고 Cray C90 보다 4.4배(수동모드)와 3배(자동모드) 정도의 계산이득을 얻을 수 있었다. 6개의 프로세서와 12개의 프로세서에 대한 격자갯수 비(최소/최대)는 각각 0.6과 0.7이다.

3.2 ONERA M6 날개 주위의 유동장 해석

3차원 병렬해석자를 검증하기 위하여 ONERA M6 날개 주위의 천음속 유동장을 해석하였다. 사용된 격자계는 O-H 유형으로 129×33×33의 격자 수를 갖는다. 2차원의 경우와 동일하게 해석자의 병렬 효율을 높일 수 있도록 영역 분할을 하였으며 프로세서를 4, 8 그리고 16개로 나누어 계산하였다. 계산 격자의 형태에 무관하게 자료전송을 할 수 있도록 하였으며 각 프로세서가 다중블록 격자계를 포함할 수도 있도록 하였다. 사용된 유동 조건은 자유흐름 마하수가 0.839이고 받음각이 0도 이다. 그림[11](a)에 16개로 분할된 격자계를 나타내었으며 그림[11](b)에 이에 해당하는 등압력 선도를 나타내었다. 그림[13](a)-(d)에 스팬(span) 위치에 따른 표면 압력 분포를 나타내었으며 Cray C90에서의 결과 및 실험치와 비교하였다. 그림[14](a)에 해석자의 병렬효율을 나타내었으며 프로세서 수에 따라 거의 이상적인 속도증진이 이루어짐을 볼 수 있다.

3.3 ONERA M6 날개와 타원형 스토어 주위의 유동장 해석

앞절에서 병렬화된 해석자와 Chimera 격자 기법의 효율성을 검증하기 위하여 ONERA M6날개에 타원체의 스토어가 결합된 예제를 해석하였다. 날개의 격자계는 앞절과 동일하게 사용하였으며 스토어 격자계는 O-O 유형으로 65×25×33의 격자수를 가진다. 스토어의 위치는 날개 스팬 방향으로 55% 정도에 위치한다. 그림[12](a)는 자동모드를 이용하여 생성된 Chimera 격자계를 나타내고 있으며 격자 겹침영역이 효율적으로 구성된 것을 볼 수 있다. 영역분할은 앞절에서 분할된 날개 격자계를 동일하게 이용하였으며 스토어 격자계는 x축방향에 대해 2개로 나누었다. 사용된 유동조건은 날개만 있을 경우와 동일하며 그림[12](b)에 등압력 선도를 나타내었다. 압력선도가 경계에 걸쳐 잘 연결되고 있으며 다소간의 오차는 날개와 스토어 위치가 정확히 일치하지 않았기 때문에 발생하는 것이다. 날개에 대한 스토어의 영향을 살펴보기 위하

여 그림[13](a)-(d)에 스팬위치에 따른 압력계수 분포를 나타내었으며 Cray C90의 결과와 비교해 보았다. 계산된 결과는 잘 일치하고 있으며 현재의 병렬접근 과정이 타당함을 확인 할 수 있다. 그림에서 보듯이 스토어 근처 날개 스팬위치, 44%와 65%에서 스토어의 영향이 크게 나타나고 있음을 볼 수 있다. 그림[14](a)는 자동모드와 수동모드를 사용했을 경우 Chimera 격자계 구성시간을 나타낸 것이다. 그림에서 18개의 프로세서를 사용했을 경우 6개의 프로세서를 사용했을 경우 보다 약 25%의 계산시간의 감소를 얻을 수 있었다. 또한 그림[14](b)는 전체 계산시간을 나타내며 18개의 프로세서를 사용한 경우 20-25%의 계산 이득을 얻을 수 있었다. 그림[14](c)에 자동모드와 수동모드를 사용한 전체 계산시간을 Cray C90과 비교하였으며 18개의 프로세서를 사용할 경우, 각각에 대해 약 60% 및 80 %의 계산시간 감소를 이룰 수 있었다. 6, 10 그리고 18개의 프로세서에 대한 격자갯수 비(최소/최대)는 각각 0.76, 0.68 그리고 0.35이다.

4 결 론

상대운동 및 정상상태 유동장을 해석 할 수 있는 병렬프로그램이 개발되었다. 자유도 운동을 포함하는 해석자 및 Chimera 격자기법의 모든 단계인 Chimera 홀절단, 영역연결 기법들이 병렬화 되었다. Chimera 홀 절단과정의 Hole-Map알고리즘을 병렬화하기 위한 효율적인 과정이 본 연구에서 제안되었으며 최적의 삽간영역을 구성할 수 있는 cut-paste 알고리즘이 본 연구에서 병렬화 되었다. 내부격자간 정보전달과 격자계간 정보전달을 포함하는 Chimera 격자계에서도 병렬코드가 효율적으로 작동함을 알 수 있었으며 3개이상의 프로세서에서 Cray C90보다 좋은 성능을 보이고 있음을 알 수 있었다.

5 후 기

본 연구는 한국과학재단 특정기초연구(과제 번호 : 96-0200-03-01-3)과제의 일부로 수행되었으며 지원에 감사드립니다.

References

[1] L.E. Lijewski and N.E. Suhs, *"Time-Accurate Computational Fluid Dynamics Approach to Transonic Store Seperation Trajectory Prediction,"* Journal of Aircraft, Vol.31, No.4, July-Aug, 1994

[2] S. Lee, M. Park, K.W. Cho and J.H. Kwon, *"A New Automated Chimera Method for the Prediction of Store Trajectory,"* AIAA paper 99-3131, 1999

[3] J.L. Steger and F.C. Dougherty and J.A. Benek, *"A Chimera Grid Scheme,"* In Advances in Grid Generation, pp.59-69, ASME FED-Vol.5, New York, NY, June 1985

[4] N.E. Suhs, and R.W. Tramel, *"PEGASUS 4.0 User's Manual,"* AEDC-TR-91-8, June 1991.

[5] D.M. Belk *"A New Approach to Domain Decomposition : The Beggar Code,"* 2nd Overset Composite Grid and Solution Technology Symposium, Fort Waltson Beach, Florida, October 25-28, 1994

[6] R.L. Meakin, *"A New Method for Establishing Intergrid Communication Among Systems of Overset Grids,"* AIAA paper 91-1586, 1991.

[7] M.S. Milgram, *"Does a Point Lie Inside a Polygon ?,"* Journal of Computational Physics Vol.84, 1989, pp.134-144.

[8] K.W. Cho, J.H. Kwon, and S. Lee, *"Development of an Efficient Overlapped Grid System for the Steady and Unsteady Problems,"* Numerical Grid Generation in Computational Field Simulations, edited by M. Cross, B. K. Soni, J.F. Thompson, and J. Hauser, Proceedings of the 6th International Conference, 1998.

[9] F.C. Dougherty, *"Development of a Chimera Grid Scheme with Applications to Unsteady Problems,"* Ph.D Thesis, Stanford Univ., June, 1985.

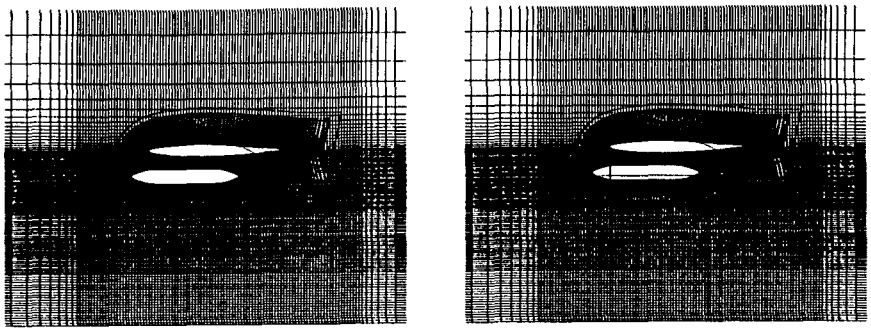
[10] G. Chesshire, W.D. Henshaw, *"Composite Overlapping Meshes for the Solution of Partial Differential Equations,"* Journal of Computational Physics 90, pp.1-64, 1990

[11] D.W. Barnette, *"BREAKUP : A Computer Code for Parallelizing the Overset Grid Approach,"* AIAA paper 98-2732, 1998.

[12] N.C. Prewitt, D.M. Belk and W. Shyy, *"Parallel Grid Assembly Within the Beggar Code,"*, The 4th Symposium on Overset Composite Grid and Solution Technology, 1998.

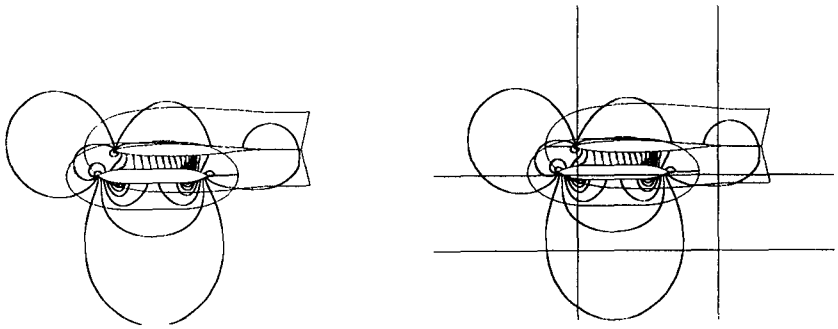
[13] A.W. Winssink and R.L. Meakin, *"On Parallel Implementations of Dynamic Overset Grid Methods,"* SC97 : High Performance Networking and Computing, 1997

[14] S. Lee, *"Numerical Computation of Flows about an Aircraft with Flow-through Inlet,"* Journal of The Korean Society for Aeronautical and Space Sciences, Vol.25, No.5, 1997, pp.16-23.



(a) Grid System : 3-CPU (b) Grid System : 12-CPU

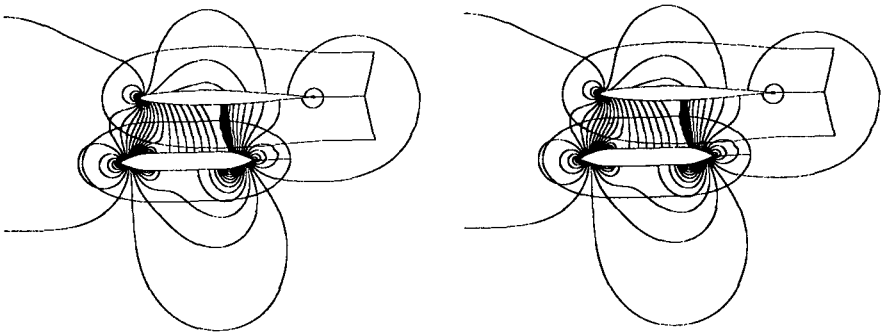
그림[5] Airfoil + Store drop test case : Grid System



(a) Pressure Contours(Min. Value=0.25, Max. Value=0.85, Δ=0.04) : 3-CPU

(b) Pressure Contours(Min. Value=0.25, Max. Value=0.85, Δ=0.04) : 12-CPU

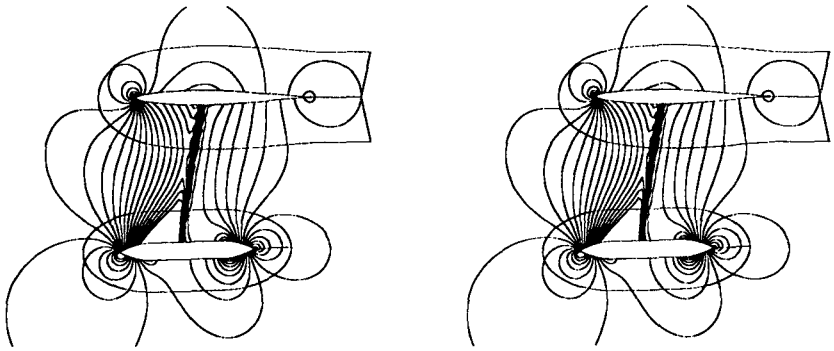
그림[6] Airfoil + Store drop test case $M_\infty = 0.6$



(a) Pressure Contours(Min. Value=0.4, Max. Value=0.84, Δ=0.02) : C-90

(b) Pressure Contours(Min. Value=0.4, Max. Value=0.84, Δ=0.02) : T3E, 3-CPU

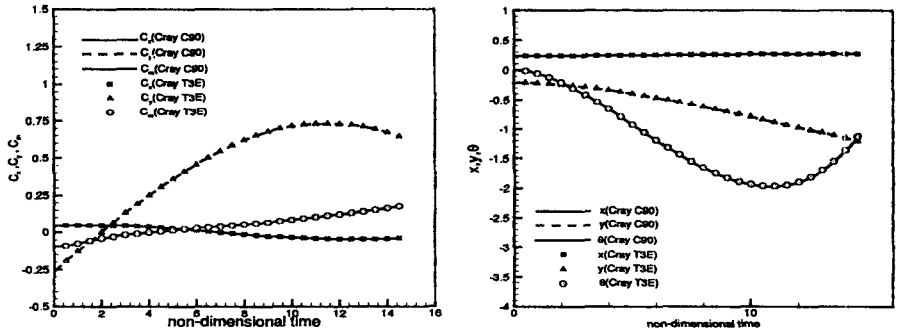
그림[7] Airfoil + Store drop test case $M_\infty = 0.6$, $t^* = 5.45$



(a) Pressure Contours(Min. Value=0.4, Max. Value=0.84, Δ=0.02) : C-90

(b) Pressure Contours(Min. Value=0.4, Max. Value=0.84, Δ=0.02) : T3E, 3-CPU

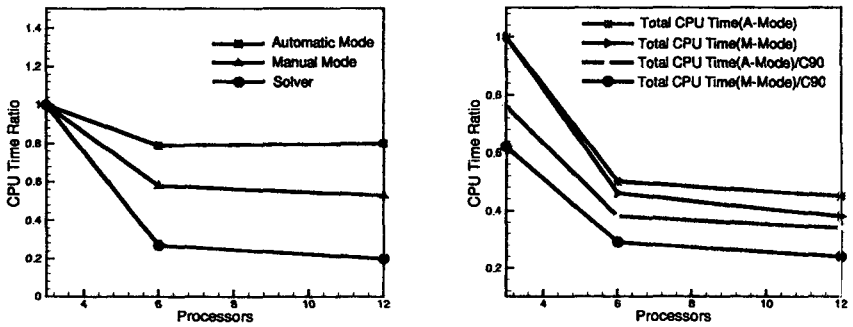
그림[8] Airfoil + Store drop test case $M_\infty = 0.6$, $t^* = 12.45$



(a) Force and Moment Coefficients

(b) Store Trajectory

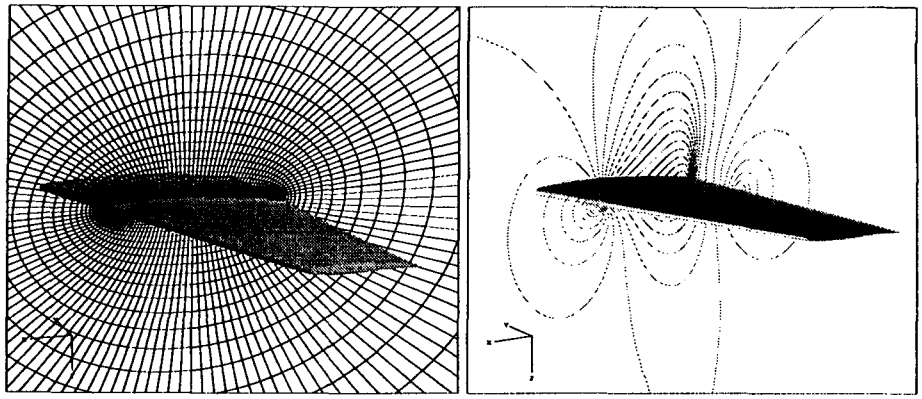
그림[9] Airfoil + Store drop test case $M_\infty = 0.6$



(a) Parallel Performance

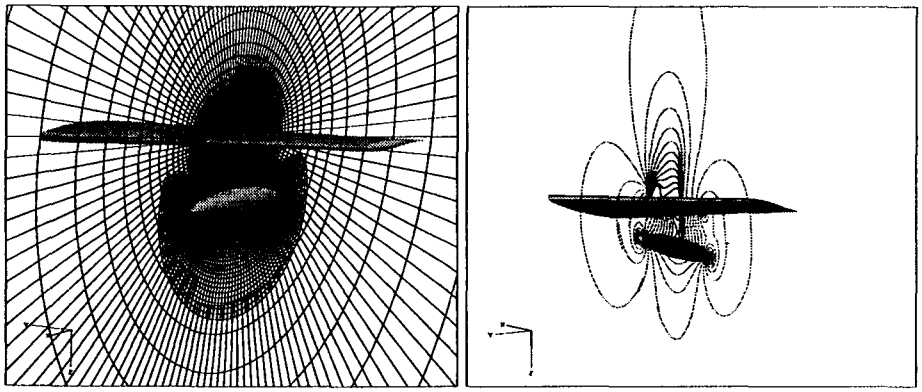
(b) Parallel Performance

그림[10] Airfoil + Store drop test case $M_\infty = 0.6$



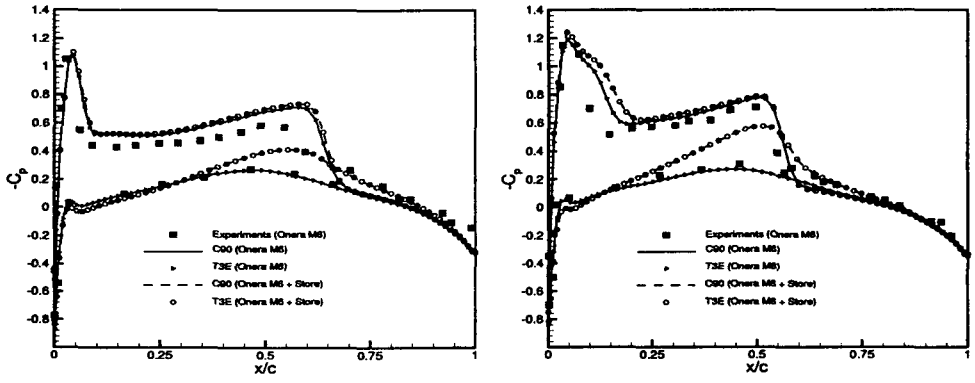
(a) Grid System (b) Pressure Contours

그림[11] ONERA M6 Wing $M_\infty = 0.839, \alpha = 3.06^o$: 16-CPU

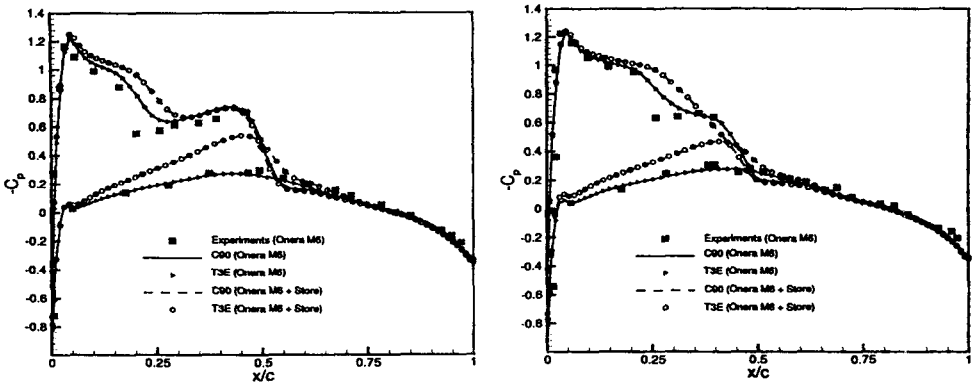


(a) Grid System (b) Pressure Contours

그림[12] ONERA M6 Wing + Ellipsoidal Store $M_\infty = 0.839, \alpha = 3.06^o$: 18-CPU

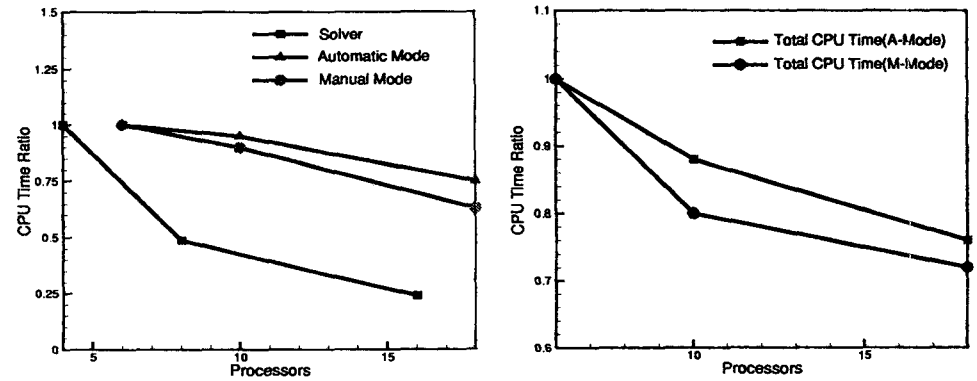


(a) Span Station : 20% (b) Span Station : 44%

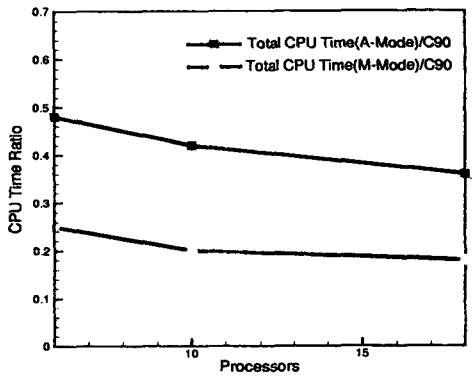


(c) Span Station : 65% (d) Span Station : 80%

그림[13] Pressure Coefficients Distribution : $M_\infty = 0.839, \alpha = 3.06^o$



(a) Parallel Performance (b) Parallel Performance



(c) Parallel Performance

그림[14] ONERA M6 Wing + Ellipsoidal Store