



## 병렬화된 Chimera 격자 기법을 이용한 다단 로켓의 단분리 운동 해석

Numerical Analysis on Separation Dynamics of Multi-stage Rocket System Using  
Parallelized Chimera Grid Scheme

\*고 순 흠<sup>1)</sup>, 최 성 진<sup>2)</sup>, 김 중 암<sup>3)</sup>, 노 오 현<sup>3)</sup>, 박 정 주<sup>4)</sup>

Soon-Heum Ko, Seongjin Choi, Chongam Kim, Oh-Hyun Rho, Jeong-joo Park

### Abstract

The supersonic flow around multi-stage rocket system is analyzed using 3-D compressible unsteady flow solver. A Chimera overset grid technique is used for the calculation of present configuration and grid around the core rocket is composed of 3 zones to represent fins in the core rocket. Flow solver is parallelized to reduce the computation time, and an efficient parallelization algorithm for Chimera grid technique is proposed. AUSMPW+ scheme is used for the spatial discretization and LU-SGS for the time integration. The flow field around multi-stage rocket was analyzed using this developed solver, and the results were compared with that of a sequential solver. The speed-up ratio and the efficiency were measured in several processors. As a result, the computing speed with 12 processors was about 10 times faster than that of a sequential solver. Developed flow solver is used to predict the trajectory of booster in separation stage. From the analyses, booster collides against core rocket in free separation case. So, additional jettisoning forces and moments needed for a safe separation are examined.

**Key words** : Chimera Overset Grid, Parallel Computation, Multi-stage Rocket System, Separation Dynamics

### 1. 서론

본 연구에서는 초음속 영역을 비행하는 편이 달린 다단 로켓의 단 분리시 부스터의 운동을 해석하고자 한다. Navier-Stokes 방정식을 해석하여 유동을 계산한 후 6자유도 운동 방정식을 이용하여 부스터의 움직임을 해석하는 방식은 Choi *et al.*[1]의 논문에 자세히 언급되어 있다. 본 논문에서는 비정상 유동 해석에 소요되는 과도한 계산 시간을 감소시키기 위하여 해석자를 효율적으로 병렬화하는 것에 그 초점을 맞추었다. Chimera 격자 기법을 사용하는 해석자의 병렬화에 대한 기존의 많은 연구들이 공통적으로 중시하는 문제는 flow solution part의 병렬화 효율에 비하여 domain connectivity

solution part의 병렬화 효율이 상대적으로 떨어진다는 것이며, 이에 대한 개선을 그 목표로 삼아왔다. 본 연구에서는 기존의 많은 연구들로부터 다단 로켓 시스템의 해석에 적합하다고 사려되는 M\_PPCG 알고리즘을 개발하여 비정상 유동장 해석에 적용하였다. 그리고 더 나아가 점성력의 영향을 확인하기 위해 비점성 유동 해석에 의한 결과와 점성 유동 해석에 의한 결과를 비교하여 점성 유동장에서 부스터의 운동 특성 변화에 대하여 고찰하였다.

### 2. 수치 해법

#### 2. 1 지배 방정식

압축성 Navier-Stokes 방정식을 3차원  $x, y, z$  직교 좌표계에서 무차원화된 보존형(conservative

1) Ph. D Candidate, 서울대학교 기계항공공학부

(151-742 서울특별시 관악구 신림동 산 56-1, E-mail : floydfan@hanmail.net, Tel : 02-880-1903)

2) Post Doctor, 서울대학교 기계항공공학부 (151-742 서울특별시 관악구 신림동 산 56-1)

3) Professor, 서울대학교 기계항공공학부 (151-742 서울특별시 관악구 신림동 산 56-1)

4) Head of Rocket System Development Team, 한국항공우주연구원 (305-606 대전시 유성구 어은동)

$$\frac{\partial Q}{\partial t} + \frac{\partial E}{\partial x} + \frac{\partial F}{\partial y} + \frac{\partial G}{\partial z} = \frac{1}{Ra} \left[ \frac{\partial E_v}{\partial x} + \frac{\partial F_v}{\partial y} + \frac{\partial G_v}{\partial z} \right]$$

여기서, 보존변수  $Q$  및 플럭스 벡터들은 다음과 같이 정의된다.

$$Q = \begin{pmatrix} \rho \\ \rho u \\ \rho v \\ \rho w \\ \rho e \end{pmatrix}, E = \begin{pmatrix} \rho u \\ \rho u^2 + p \\ \rho uw \\ \rho uw \\ (\rho e + p)u \end{pmatrix},$$

$$F = \begin{pmatrix} \rho v \\ \rho v^2 + p \\ \rho vw \\ \rho vw \\ (\rho e + p)v \end{pmatrix}, G = \begin{pmatrix} \rho w \\ \rho w \\ \rho w^2 + p \\ \rho w \\ (\rho e + p)w \end{pmatrix}$$

$$E_v = \begin{pmatrix} 0 \\ \tau_{xx} \\ \tau_{xy} \\ \tau_{xz} \\ (e_v)_5 \end{pmatrix}, F_v = \begin{pmatrix} 0 \\ \tau_{yx} \\ \tau_{yy} \\ \tau_{yz} \\ (f_v)_5 \end{pmatrix}, G_v = \begin{pmatrix} 0 \\ \tau_{zx} \\ \tau_{zy} \\ \tau_{zz} \\ (g_v)_5 \end{pmatrix}$$

$$(e_v)_5 = u\tau_{xx} + v\tau_{xy} + w\tau_{xz} + \frac{\mu}{(\gamma-1)Pr} \frac{\partial T}{\partial x}$$

$$(f_v)_5 = u\tau_{yx} + v\tau_{yy} + w\tau_{yz} + \frac{\mu}{(\gamma-1)Pr} \frac{\partial T}{\partial y}$$

$$(g_v)_5 = u\tau_{zx} + v\tau_{zy} + w\tau_{zz} + \frac{\mu}{(\gamma-1)Pr} \frac{\partial T}{\partial z}$$

$$\tau_{ij} = \mu \left[ \frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} \right] + \lambda \left[ \frac{\partial u_k}{\partial x_k} \right] \delta_{ij}$$

$$Ra = \frac{\rho_\infty c_\infty D}{\mu_\infty} = \frac{Re_\infty}{M_\infty}$$

비정상 유동장의 해석을 위하여는 dual time stepping 기법이 사용되었다. 2차의 시간 정확도를 가지도록 하기 위하여 다음과 같은 식을 사용한다.

$$\frac{\partial \hat{Q}}{\partial \tau} = - \hat{R}^{n+1,s+1} - \frac{3 \hat{Q}^{n+1,s+1} - 4 \hat{Q}^n + \hat{Q}^{n-1}}{2\Delta t}$$

여기서  $\hat{R} = \partial \hat{E} / \partial \xi + \partial \hat{E} / \partial \eta$ 이고,  $\tau$ 는 pseudo time항,  $n$ 은 physical time level이고,  $s$ 는 pseudo

time level을 나타낸다. 또,  $t$ 는 물리적 시간을 의미한다. 본 문제에서는  $\Delta t = 0.005$  sec로 고정한다.

## 2. 2 공간 이산화

2.1절의 지배방정식을 유한체적법을 이용하여 공간 이산화하고 각 비점성 플럭스들은 풍상차분법 중 AUSMPW+ 기법을 이용하여 구하였다.

AUSMPW+ 기법은 기존의 AUSM 계열 기법들의 장점을 살리고 AUSMD/V의 과도한 수치점성이나 AUSM+의 강한 충격파에서의 진동 현상, 벽면에서의 압력진동을 개선하였다[2]. 또한, 고차의 정확도를 갖는 플럭스를 계산하기 위해 MUSCL (Monotone Upstream Centered Schemes for Conservation Laws)형 방법을 사용하였고, 불연속 부근에서의 수치 진동 억제에 위한 제한자로는 Van Albada Limiter를 도입하였다.

## 2. 3 시간 적분

### 2.3.1. LU-SGS 기법

계산시간과 계산용량, 계산오차등을 고려하여, 시간적분법으로는 LU-SGS 기법 (Lower-Upper Symmetric- Gauss-Seidel Scheme)을 사용하였다 [3]. 이 기법은 근사 선형화 행렬을 이용하여 별도의 역행렬 계산 없이 대수적 연산만으로 행렬식을 계산하기 때문에 ADI 기법에 비하여 계산 시간이 현저히 줄어드는 장점을 가지고 있다. 반면 엄밀한 선형화 행렬 대신에 근사 선형화 행렬을 사용함으로써 수렴성과 안정성이 떨어지는 단점도 가진다.

### 2.3.2. 시간 간격의 결정

시간 간격은 각 cell에서 선형화 행렬의 고유값이 CFL 안정조건을 만족시키는 국소 시간 간격 (local time stepping)으로 그 크기를 결정하였다.

## 3. 병렬화 알고리즘

병렬 효율을 높이기 위한 다양한 방법이 제안되어 왔다. Rantakokko[4]는 multiblock 격자 기법을 사용하여 제작된 격자계에서 사용하는 processor의 수에 관계없이 적절한 병렬 효율을 유지할 수 있도록 하는 격자 분할 기법인 graph-partitioning

방법을 제안하였고, Wissink *et al.*[5]은 각 processor에서 해석을 수행하는 계산량을 서로 유사하도록 하는 dynamic load balance scheme을 제안하였다. 국내에서는 김창성 등[6]이 PPCG 알고리즘을 제안하여 Chimera 격자 기법을 이용하는 문제에서의 효율적인 병렬화 방안을 제안하였고, 조금원 등[7]은 병렬 영역연결 기법을 이용하여 스토어 분리 문제에 대한 해석을 수행하였다.

본 연구에서는 위에 언급한 논문들에서 제시한 방법들의 장단점을 검토하고, 이로부터 현재의 해석 문제에 가장 적합하다고 판단되는 병렬화 기법을 제시하고자 한다.

### 3. 1 Partitioning 기법

Multiblock 격자를 사용하는 해석자의 병렬화와 관련하여 우선적으로 결정할 사항은 어떤 분할 기법을 채용하는가 하는 것이다. Multiblock 격자를 사용하는 문제에서의 격자 분할 방법은 대략 다음의 네 가지 방법들이 있다[4].

- Block Partitioning Strategy
- Cluster Partitioning Strategy
- Single Grid Partitioning Strategy
- Graph Partitioning Strategy

이 중 본 연구에서는 data 교환에 따른 효율 감소를 최소화하고, 프로그래밍이 용이한 Cluster 분할 기법을 이용하고자 한다. Cluster된 격자 분포와 사용하는 CPU의 개수는 3.3절에서 자세히 기술하도록 한다.

### 3. 2 Chimera 기법을 사용하는 해석자의 병렬화

Chimera 격자 기법을 사용하는 해석자의 병렬화에서는 flow solution part의 load balance 뿐 아니라 domain connectivity solution part의 load balance 역시 고려해 주어야 하며, 이를 위하여는 dynamic load balance algorithm[5]을 이용하여야 한다. 그러나, 본 문제에서와 같이 특정 block에 IGBP들이 집중되는 경우에는 특정 block에 부여하는 CPU가 급증하여 오히려 flow solution 부분에서의 load balance가 나빠지게 된다. 그리고, domain connectivity solution에 소요되는 시간은 flow solution에 필요한 시간에 비하여 10% 안팎

이므로[7] flow solution 부분에 대한 load balance가 만족되도록 전체 격자수의 균등한 분포를 이루도록 하는 것이 중요하다 할 수 있다.

따라서 본 연구에서는 Cluster Partitioning시에 각 CPU들이 동일한 수의 격자점을 가지도록 만들고, PPCG(Parallel Processing for Chimera Grid) 알고리즘[6]을 개선한 M\_PPCG 알고리즘을 이용하여 병렬 처리를 수행하였다. M\_PPCG 알고리즘은 다음과 같다.

1. 주 격자계와 부 격자계를 분할하고 각 CPU에 격자를 할당한다.
2. 동일한 격자계에 속하는 프로세서들을 동일 그룹으로 지정한다.
3. 각 CPU는 자신에게 할당된 격자들 중 hole과 fringe cell을 찾고, 다른 그룹의 CPU들로 자신의 fringe cell의 위치 정보를 전달한다.
5. 각 CPU는 자신이 받아온 상대 격자계의 fringe cell들에 대한 donor cell을 찾는다.
6. Fringe cell이 있는 CPU로 donor cell에 관련된 정보를 보내어 준다.
7. 각 CPU는 다른 그룹의 CPU들로부터 전달받은 donor cell의 정보에 대하여 분석하고, 물리적으로 타당한 데이터를 donor cell의 정보로 채택한다.

### 3. 3 격자계 형성

본 수치 해석에 사용된 격자계는 전체적으로는 Chimera Overset Grid를 사용하며, 주 격자계는 3개의 block, 부 격자계는 단일 block으로 구성되어 있다. 비점성 해석에서 사용된 격자계는 그림 1과 같다. 주 격자계는  $81 \times 15 \times 54$  (1st Block),  $81 \times 29 \times 54$  (2nd Block) 및  $81 \times 15 \times 54$  (3rd Block)의 격자점을 가지며, 부 격자계는  $50 \times 53 \times 48$ 의 격자점을 가진다. 점성 계산에서는 위와 같은 구성의 격자계를 사용하되 사용하는 격자점의 수를 바꾸고 벽면으로 격자를 밀집하여 주었다. 또한, 병렬 처리를 위하여 전체 격자계는 총 12개의 영역으로 나누어주었다. 수월한 경계 조건의 적용과 각 CPU당 할당되는 계산량의 load balance를 고려하여 주 격자계는 각각  $41 \times 15 \times 54$ 개의 격자점을 가지는 영역들로 분할되었으며, 부 격자계는  $50 \times 14 \times 48$ 개의 격자점을 가지도록 하여 각 CPU는 대략 33000개의 격자점들에 대한 계산을 수행하게 된다.

## 4. 계산 결과

### 4. 1 정상 상태의 유동 해석

본 유동 해석에 사용된 해석자의 정확성은 기존의 연구들을 통해 입증되었으므로[8], 해석자의 병렬화 과정에서 생길 수 있는 오차 여부의 판별을 위하여 병렬화한 해석자를 이용하여 얻은 계산 결과를 기존의 단일 프로세서를 사용하는 해석자를 통한 결과와 비교하여 병렬화에 따른 정확성을 검증하였다.

병렬화된 해석자를 이용하여 KSR-III에 대한 정상 상태의 유동을 해석하였다. 유동 조건은  $Ma=1.7$ ,  $Re=1.431 \times 10^7$ 이며, 받음각은  $0^\circ$ 이다. 수렴 판정은 비점성 해석시 5 order 이상, 점성 해석시 4 order 이상 오차가 줄어드는 것을 기준으로 하였다. 주 로켓과 부스터의 중심선을 따른 단면에서의 압력 분포는 그림 2와 같다. 이를 통해 부스터의 Nose 부분에서 생성된 Shock에 의한 주 로켓과 부스터간의 상호작용을 관찰할 수 있다. 그리고, 표면 압력 분포를 통하여 주 로켓의 핀에서 생성되는 Shock이 부스터에 미치는 영향을 확인할 수 있다.

주 로켓과 부스터 각각에 대하여 길이 방향으로의 표면 압력 계수를 나타낸 그래프는 그림 3과 같다. 이 결과를 통해 주 로켓과 부스터간 간격이 가장 좁은 원주각(주 로켓= $90^\circ$ , 부스터= $270^\circ$ )에서 주 로켓과 부스터간의 상호작용에 의해 상대적인 압력의 증가가 큼을 알 수 있다. 원주각이  $270^\circ$  일 때의 부스터의 표면 압력 계수를 살펴보면 nose부의 끝에서 순간적으로 큰 압력 증가가 나타나는데, 이는 부스터의 위에서 생성된 궁형 충격파가 주 로켓에 부딪친 후 반사하여 다시 부스터에 부딪치기 때문에 나타나는 현상이다.

해석자의 병렬화에 대한 검증을 위해 핀에 의한 유동의 변화가 두드러지는 원주각  $45^\circ$ 에서의 표면 압력 계수 분포에 대한 단일 프로세서를 이용한 해석 결과를 병렬화된 해석 결과와 비교하여 해석자의 정확성을 입증할 수 있었다(그림 4).

점성에 의한 유동 분포의 변화를 고찰하기 위해 점성 및 비점성 계산에 따른 부스터의 표면 압력 분포의 차이를 나타내면 그 결과는 그림 5와 같이 나타난다. 전체적인 경향을 분석하면 일단 주 로켓의 fin에서 생성된 충격파가 미치는 영향이 다르게 나타남을 확인할 수 있다. 비점성 해석 결과에서는 상대적으로 leading edge에서 생성된

oblique shock의 전파 영역이 넓게 잡히며, 대신에 압력의 증가는 상대적으로 낮음을 알 수 있다. 또한, 부스터의 base부를 비교하면 flare skirt가 시작하는 부분에서 압력의 증가가 점성 해석시 훨씬 적다는 것을 확인할 수 있다. 이는 fin의 leading edge를 지난 이후 계속적으로 생성되는 팽창파에 의한 압력 감소의 영향을 점성 계산에서 더 잘 모사할 수 있다는 것으로 해석 가능하며, 또 이러한 base부에서의 압력 차이는 단 분리시 부스터의 하강 속도에 영향을 미칠 것임을 쉽게 유추할 수 있다.

### 4. 2 병렬화 효율

해석자를 병렬화하는 주된 이유는 단일 프로세서를 사용할 때 요구되는 과도한 해석 시간을 단축하고자 하는 것이 가장 주된 이유이다. 따라서 해석자의 병렬화에서 주안점을 두는 것은 병렬화에 따른 효율이 어느 정도이냐 하는 것이다. 이상적으로는 병렬화 효율이 1이 되도록 하는 것이 좋으나, 프로세서간의 data 교환 및 serial 계산으로 인해 실제 효율은 1에는 미칠 수가 없다.

본 해석자의 효율을 알아보기 위하여 병렬화된 해석자와 단일 프로세서를 사용하는 해석자를 사용할 때 유동 해석에 소요되는 계산 시간을 비교하였다. 효과적인 비교를 위하여 해석자를 크게 네 부분(해석 전 단계, domain connectivity 계산, metric 계산, 유동 계산)으로 구분하고 각 부분의 계산 수행에 소요되는 시간들을 체크하였다. 그 결과는 표 1과 같다.

초기 계산에서는 12개의 CPU를 사용할 때 10.9 정도의 속도 향상을 보였다. 이 부분에는 변수의 정의 및 표면 격자의 인식과 같은 serial 계산이 포함되므로 병렬화 효율이 1이 되지 않는다는 점이다.

Hole과 fringe cell 및 donor cell을 찾고 그 결과를 서로 교환하는 Chimera Calculation 부분에서는 효율이 매우 떨어지는 것으로 나타났다. 이는 각 프로세서가 donor cell을 찾을 때 자신과 다른 격자계에 속하는 모든 fringe cell에 대하여 donor cell 찾기를 수행하고 그 데이터를 전송하므로 과도한 데이터 통신량에 의해 효율 감소가 일어난다.

Metric을 계산하는 부분은 프로세서간의 data 통신도 없고 하나의 CPU가 계산해야 하는 계산량도 사용하는 CPU의 수만큼 감소하므로 병렬화 효율은 이상적인 병렬화의 결과를 나타내게 된다. 측정된 병렬화 효율은 1보다 약간 큰 값이 되는

데, 이는 각 프로세서가 계산하는 양이 줄어들면서 cache를 사용할 확률이 늘어나게 되므로 효율이 1보다도 커지는 결과를 보여준다.

전체 계산 시간의 90% 가까운 시간을 소요하는 유동 해석 부분에서의 속도 향상은 12개의 CPU를 사용할 때 10.8 정도이며, 효율 감소의 원인은 경계에서의 유동 변수에 대한 데이터 통신이 필요하기 때문이다.

전체적으로는 12개의 CPU를 사용할 때 9.9 정도의 속도 향상을 보이게 되며, 효율 향상을 위해 domain connectivity 계산에서의 병렬화 효율을 향상시키는 것이 전체 해석자의 효율 향상에 매우 중요함을 알 수 있다.

### 4.3 단 분리시 유동 해석 결과

병렬화된 해석자를 이용하여 다단 로켓의 단 분리시 유동을 해석하였다. 분리 운동을 모사하기 위하여 6자유도 운동 방정식을 유동 해석자에 접목하여 시간 전진에 따른 부스터의 위치 변화와 유동장의 유동 변화를 나타내었다. 단 분리력이 주어지지 않은 경우 분리된 부스터는 0.22초가 지나면 주 로켓에 충돌하게 된다. 이는 주 로켓의 base부에 있는 skirt에서 생겨난 경사 충격파가 부스터의 하단부에 영향을 주어 음의 방향으로 모멘트가 생기도록 영향을 주기 때문이다.

이와 같은 충돌 방지를 위하여 단 분리력 및 모멘트를 주어 단 분리를 수행할 수 있으며, 단 분리력 40kN 및 단 분리 모멘트 43.6kN·m를 주어 단 분리를 수행하는 경우의 해석 결과는 그림 6과 같다. 이와 같은 단 분리력 및 모멘트를 주면 안전 한 단 분리가 일어나며, 부스터의 nose 부분에서 생성되는 충격파가 주 로켓에 미치는 영향도 크게 작용하지 않음을 확인할 수 있다.

### 5. 결론 및 향후과제

본 연구에서는 Chimera 격자 기법을 사용하는 해석자의 효율적인 병렬 처리 방안에 대하여 살펴보고 발사체의 단 분리시의 운동에 대하여 예측해 보는 것을 그 목표로 삼았다. 현재 개발된 해석자는 12개의 CPU를 사용하였을 때 대략 0.825의 병렬화 효율을 나타내었으며, 현재의 해석자의 병렬화 효율을 높이기 위하여는 domain connectivity 계산 부분을 개선하는 것이 필요함을 알 수 있었다. 그리고, 병렬화된 해석자를 이용하여 다단 로

켓의 단 분리 운동에 대한 해석을 수행한 결과 주 로켓의 skirt에서 생성되는 경사 충격파로 인해 부스터가 주 로켓에 충돌하게 됨을 알 수 있었다. 따라서 강제적인 단 분리력을 작용하여 해석을 수행하였으며, 안정적인 단 분리를 확인할 수 있었다. 그러나, 좀 더 정확한 해석을 수행하기 위하여는 base 영역을 포함하도록 해석 범위를 넓혀줄 필요가 있다.

### 후기

본 연구는 한국항공우주연구원 위탁연구과제 결과의 일부임을 밝히며, 지원에 감사드립니다.

### 참고 문헌

- [1] Choi, S., Ko, S. H., Kim, C., Rho, O. H. and Park, J., "Numerical Analysis of Separation Dynamics of Strap-On Boosters in the Dense Atmosphere," *KSAS International Journal*, vol. 2, No. 2, pp. 1-18, 2001
- [2] Kim, K. H., Kim, C. and Rho, O. H., "Accurate Computations of Hypersonic Flows Using AUSMPW+ Scheme and Shock-Aligned Grid Techniques," *AIAA Paper 98-2442*, 1998
- [3] Jameson, A. and Yoon, S., "Lower-Upper Implicit Schemes with Multiple Grids for the Euler Equations," *AIAA J.* vol. 25, No. 7, pp. 929-935, 1987
- [4] Rantakokko, J., "Partitioning Strategies for Structured Multiblock Grids", *Parallel Computing*, vol. 26, pp. 1661-1680, 2000
- [5] Wissink, A. M., and Meakin, R. L., "On Parallel Implementations of Dynamic Overset Grid Methods", *Supercomputing 97: High Performance Networking and Computing*, San Jose, CA, Nov. 15-21 1997
- [6] 김창성, 김종암, 노오현, "Chimera 격자기법의 병렬처리를 이용한 고양력 익형 주위 난류 유동장의 효율적 해석," *한국항공우주학회지*, vol. 27, No. 4, pp.1-12, 1999
- [7] 조금원, 권장혁, 이승수, "Chimera 기법의 병렬처리에 관한 연구," *한국전산유체공학회 춘계학술대회*, pp. 35-47, 1999
- [8] 고순흠, 김종암, 노오현, "Chimera 격자 기법을 이용한 Fin달린 다단로켓의 유동해석," *제2회 우주발사체기술 심포지움*, 2001

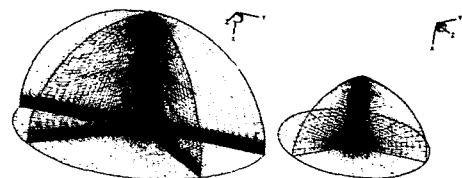


Fig. 1 Core rocket grid & booster grid

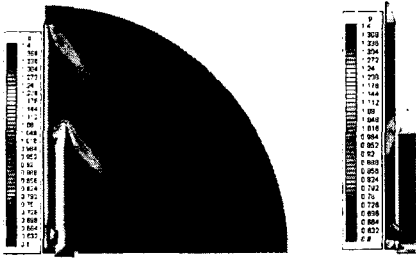


Fig. 2 Pressure contour along the XZ-plane & surface pressure contour

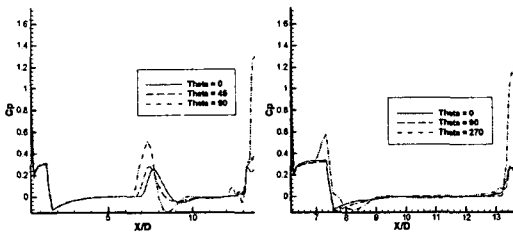


Fig. 3 Cp Distribution of core rocket & booster

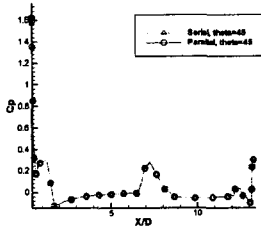


Fig. 4 Comparison of Cp distribution between serial solution and parallel one

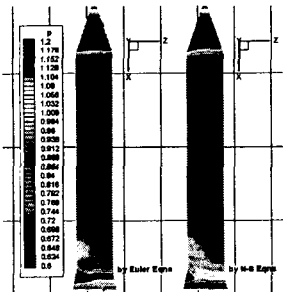
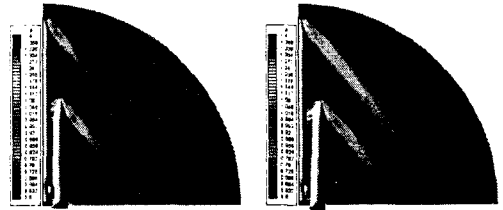
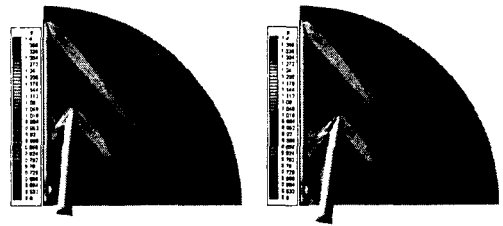


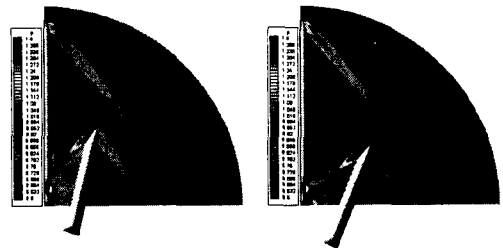
Fig. 5 Comparison of Euler and Navier-Stokes solution in the surface of booster



a) Time = 0.00 sec    b) Time = 0.06 sec



c) Time = 0.12 sec    d) Time = 0.18 sec



e) Time = 0.24 sec    f) Time = 0.30 sec

Fig. 6 Pressure Contours in Separation Stage

	1 CPU	6 CPUs	12 CPUs
Pre-calculation	16.441 (1)	3.834 (4.288)	1.508 (10.903)
Chimera Calculation	3181.651 (1)	1154.977 (2.755)	643.278 (4.946)
Metric Calculation	1.886 (1)	0.305 (6.184)	0.157 (12.013)
Flow Calculation	36724.768 (1)	6526.105 (5.627)	3384.893 (10.850)
Total	39924.746 (1)	7685.221 (5.195)	4029.836 (9.907)

Table 1. Calculation Time of Parallel Code and Serial One