

중첩 격자계의 효율적 Hole Finding 기법

김 병 수^{*1}, 고 성 호²

EFFICIENT HOLE FINDING METHOD FOR OVERSET GRID SYSTEM

B.S. Kim^{*1} and S.H. Ko²

In this paper an automated hole-finding method for overset grids is introduced which uses recursive octree-cell division. A graphic program which enables the user to do the hole-cutting with ease is also introduced. Using this program it is found that there is an optimum combination of the level of octree division and vector calculation for the efficient and fast hole finding.

Key Words: 격자 생성(Grid generation), 중첩격자(Overset grid), 홀점찾기(Hole finding), 옥트리 셀(Octree cell)

1. 서 론

전산유체역학 기법에 의한 유동장 계산을 위해서는 격자 생성이 선행되어야 한다. 전산유체역학 분야에서 사용되는 격자의 종류는 크게 정렬 격자와 비정렬 격자로 나눌 수 있고, 사용하는 유동장 해석 프로그램에 따라 해당 프로그램이 지원하는 격자 타입을 선택하게 되겠으나, 만일 추가적인 선택이 가능하다면 사용할 격자의 타입은 격자 생성의 용이성, 요구되는 유동해의 정확도, 그리고 대상 유동장의 기하학적 특성에 의해서 결정될 것이다.

중첩 격자계는 Steger 등이 Chimera scheme이라는 이름으로 처음 그 기법을 발표한 후로, 정렬 격자계의 생성이 까다로운 복잡한 형상의 물체에 대한 유동 계산에 매우 긴요하게 사용되어 왔고[1], 특히 항공기에서의 무장 분리 문제와 같이 상대적 운동이 있는 다물체 문제나 복잡한 실제 형상에 대한 유동 계산에 현재까지도 많이 이용되고 있다[2]. 중첩 격자계는 전체적으로 단일 블록 격자계나 패치 타입의 다중-블록 격자계의 생성이 용이치 않은 유동장 경계 형상에 대하여 격자 생성이 용이한 부분별로 단순한 격자계를 생성하고 이들을 중첩하여 전체 유동장을 채우는 방식을 사용하게 되는데, 이와 같은 접근 방식으로 인하여 개개의 단순 블록 격자계의

생성은 매우 쉬워지는 반면, 중첩된 부분에 대한 각 격자 블록에 속하는 물체 표면 내부 공간에 위치한 격자점, 즉 홀점(Hole points)의 선별 및 제거 작업 등 유동장 해석 코드에 사용되기 전에 전처리 작업이 필요하고, 바로 이 부분이 중첩 격자계의 원활한 활용에 있어서 가장 커다란 장애 요소에 해당한다. 이와 같은 중첩격자계의 생성 및 전처리 작업에 활용되는 대표적인 코드로는 PEGASUS, DCF3D 등을 들 수 있다[3,4].

매우 실제적 형상의 전기체 주변 유동장 계산 등을 포함한 여러 가지 문제에 많이 사용되고는 있으나 여전히 사용자의 많은 수작업과 경험을 요하는 PEGASUS 코드를 개선하기 위하여 미국의 NASA 연구진들은 최근에 많은 연구 끝에 PEGASUS5라는 새 버전을 발표하기도 하였다.[5] 한편, DCF3D 코드와 같은 경우 물체 표면을 둘러싸는 hole cutter라고 부르는 해석적 표면식을 사용하여 hole finding을 가속시키는 기법을 사용하기도 하나, 이 hole cutter의 생성을 위해서는 여전히 많은 시행착오와 수작업이 요구된다[4].

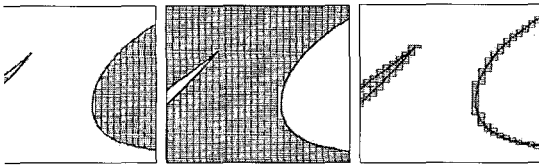
중첩 격자계를 유동장 코드에서 사용하기 위하여 각 격자 블록간의 상호 관계와 중첩 격자점간 내삽을 위한 정보, 그리고 물체 표면 내부 공간 등에 의하여 계산에서 제외되어야 하는 hole 등을 찾아야 하는데, 본 연구에서는 주어진 중첩 격자계의 물체 표면 격자를 이용하여 표면 내부에 위치한 격자점을 제거하기 위한 hole 찾기 작업을 Octree hole-map 기법을 활용하여 수행하는 기법을 소개하고 이를 그래픽 프로그램으로 구현하였다.

접수일: 2006년 6월 5일, 심사완료일: 2006년 6월 13일.

1 정희원, 충남대학교 항공우주공학과

2 정희원, 충남대학교 기계설계공학과

* Corresponding author E-mail: kbskbs@cnu.ac.kr



(a) Inside (b) Outside (c) Fringe

Fig. 1 Cartesian hole-map의 셀의 종류

2. 본 론

Octree hole-map 기법

중첩 격자계 상의 한 격자점이 타 블록의 물체 표면 경계부에 있는지를 확인하는 것이 hole 찾기의 궁극적인 내용이다. 즉 3차원 격자계의 경우 3차원 공간상에 있는 한 점이 3원 공간상의 폐곡면 내부에 위치하는지 또는 외부에 위치하는지를 찾아내는 문제인 것이다. PEGASUS5의 경우 이와 같은 hole cutting 작업의 자동화를 위하여 Cartesian hole-map 방법을 이용한다. 이 기법에 따르면 물체표면을 포함하는 대 공간을 적절한 해상도의 직육면체들, 즉 Cartesian 셀들로 나누고 각 셀들이 물체 표면보다 바깥쪽(즉, 유동장 부분)에 있는지, 또는 물체 표면 내부(유동 계산에서 제외되어야 고체부)에 있는지, 그리고 물체 표면 경계에 걸쳐 있는지를 찾아서 각각을 내부셀(inside cell), 외부셀(outside cell), 그리고 경계셀(fringe cell)로 구분을 짓는다.

일단 Cartesian cell들의 종류가 확인이 되고 나면, hole 여부를 확인코자하는 공간상의 임의점에 대하여 그 점을 포함하는 Cartesian cell은 Cartesian cell의 특성상 쉽게 확인이 될 것이고, 따라서 속한 셀이 내부셀이면 고체 내부의 점으로, 그리고 외부셀이면 유동장에 속하는 점으로 처리가 된다. 단, 강점이 경계셀에 속할 경우에는 경계셀에 저장된 물체 표정보를 이용하거나, 또는 PEGASUS5에서 사용하는 방법처럼 line-of-sight 기법을 이용하여 물체 표면의 내부점인가 또 외부점인가를 확인하기 위한 추가적인 작업이 이루어지게 된다. Fig.1은 PEGASUS5에 의한 셀 종류를 2차원에 대하여 시하고 있다[5].

Line-of-sight 방법은 Fig. 2에서 예시하고 있는 것과 같이 세셀에 속한 점(A와 B)들에 대하여 해당 경계셀 주위의 가까운 비경계셀(non-fringe cell)들 중 물체 표면 건너에 있는, 즉 시야에 들어오는 셀의 종류에 따라서 해당점이 내부에 있는지 또는 외부에 있는지를 판단하는 방법이다.

한편, Cartesian hole-map 기법의 경우 물체 표면의 표현 해상도를 높이기 위해서는 Cartesian cell의 숫자를 전체적으로 가시시켜야 하고, 이는 메모리 용량에 의해서 제한을 받게 되

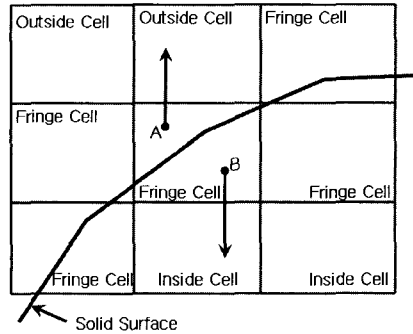
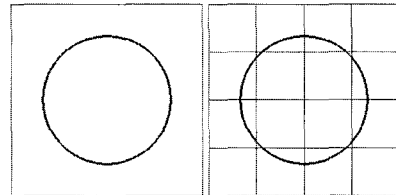
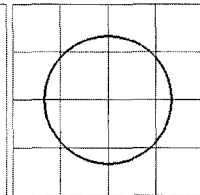


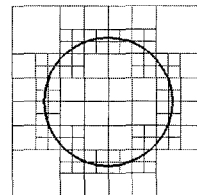
Fig. 2 Line-of-sight 알고리즘



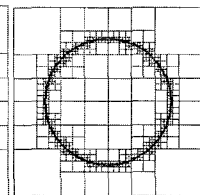
(a) Level=0



(b) Level=2



(c) Level=4



(d) Level=6

Fig. 3 Octree 기법에 의한 셀 분할

고 또 계산량 증가에 직접적인 영향을 미치게 된다. 본 논문에서는 비정렬 격자계의 일종인 Octree 셀 생성 기법을 이용하여 물체 경계를 포함하는 셀에 대해서만 회귀적으로 재분할 하는 기법을 적용하였다.

앞의 Fig. 3에 2차원에 대한 예가 나와 있듯이, 물체 표면을 포함하는 충분히 큰 직육면체 공간을 초기 셀로 하여 물체 표면을 포함하는 셀에 한하여 회귀적으로 8개의 셀로 분할하게 된다. 여기서 물체 표면 데이터라 함은 중첩 격자계의 홀점 찾기 대상이 되는 블록의 표면 격자계를 의미하고, 이 표면 격자가 일부라도 포함되어 있는 셀은 경계셀로서 회귀적 재분할의 대상이 된다. 경계셀 여부는 물체 표면의 포함 여부에 의해서 결정되는데, 그 판단은 물체 표면 격자계에 의해서 내부가 잘리는지(즉, 표면 격자계가 통과하는지)를 기하학적 계산을 통하여 이루어지게 된다.

이 기법에 의하면 일단 물체 표면을 내부에 포함하지 않는 셀은 추가적인 분할이 수행되지 않으므로 물체 표면에 대한

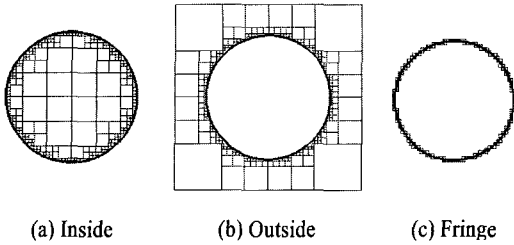


Fig. 4 Octree hole-map의 예 (Level=6)

표현 해상도를 앞의 Cartesian hole-map과 동일한 정도로 유지할 경우 cell의 숫자는 훨씬 작아지게 된다는 장점이 있다. 여기서 level은 회귀적 분할의 정도를 의미하며, level=0은 초기셀을 의미함을 할 수 있다. 물체 표면을 내부에 포함하는 셀에 대해서는 추가적으로 8개의 서브-셀로 분할되고, 이 때 8개의 서브-셀에 대하여 물체 표면 포함 여부가 테스트되고 물체 표면을 포함할 경우, 포함된 부분 물체 표면의 해당 정보도 할당된다. 원하는 해상도까지의 회귀적 분할이 끝나면, 모든 셀은 fringe 셀과 non-fringe 셀의 두 종류로 구분되어 있다. Non-fringe 셀의 경우 inside 셀과 outside 셀로 구별되어야 한다. Octree 셀들로 채워져 있는 공간은 폐쇄적으로 연결된 Fringe 셀들을 경계로 2개의 영역, 즉 물체 표면 외부와 물체 표면 내부로 나누어져 있다. 맨 바깥셀은 외부셀이라는 가정하에 바깥셀부터 차례로 서로 이웃한 non-fringe 셀들을 찾아 나가면 이들은 모두 외부셀에 해당한다. 그리고, 더 이상 서로 이웃한 non-fringe 셀을 못 찾게 되면 나머지 셀들은 모두 내부셀이 된다. 이로써 해당 공간의 물체 표면에 대한 위치정보, 즉 octree hole-map이 얻어지게 된다. Fig. 4에 level=6로 분할된 경우의 내부셀, 외부셀, 그리고 경계셀 등 octree hole-map을 2차원 예로 보여주고 있다.

일단 octree hole-map 작업이 끝나면, hole point 찾기 작업이 수행되는데, non-fringe 셀들, 즉 외부셀과 내부셀에 포함된 격자점은 쉽게 hole point 여부가 정해진다. 한편, fringe 셀에 포함된 점들에 대해서는 각 셀이 갖고 있는 부분 물체 표면 정보와 이로부터 알 수 있는 법선벡터를 이용하여 표면 내/외부 검사를 수행하게 된다. 즉, Fig. 5에 예시되어 있는 것처럼, fringe 셀에 포함된 A점과 B점에 대하여 해당 fringe 셀이 갖고 있는 부분 물체 표면 정보로부터 계산되는 물체 표면 법선 벡터(n)와 물체 표면에서 대상점까지의 방향 벡터를 이용하여 물체 내외부를 계산하게 된다. 이러한 벡터 계산을 이용하여 Fig. 5의 경우 점A는 물체 외부점이고, 점B는 내부점임을 확인할 수 있다.

2.2 OGAP 프로그램

이상과 같은 octree hole-map 기법을 이용한 중첩 격자계의

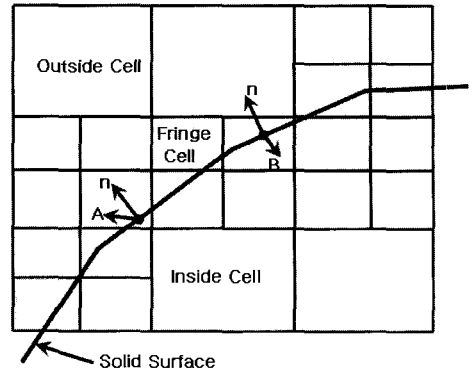


Fig. 5 경계셀 내부점에 대한 벡터 계산의 예

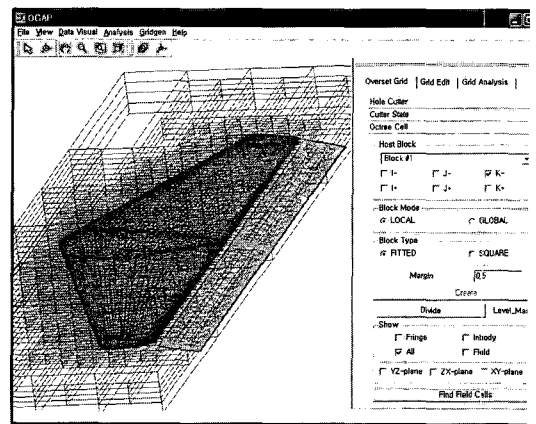


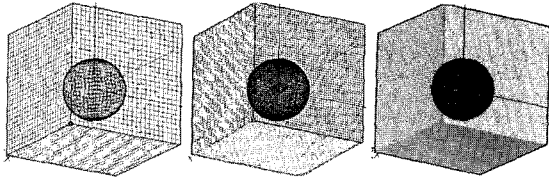
Fig. 6 OGAP의 실행 예

hole finding 작업을 편리한 그래픽 환경에서 수행할 수 있는 프로그램(OGAP : Overset Grid Analysis Program)을 개발하였다.

OGAP 프로그램은 중첩 격자계와 관련하여 필요한 전처 단계의 여러 작업을 하기 위하여 제작된 프로그램으로서, 의 격자 Jacobian의 원인이 되는 격자선 꼬임을 푼다든지 하는 격자의 질 개선 작업, 그리고 격자의 질을 점검하는 등과 DCF3D의 이용에 필요한 hole-cutter의 생성을 그래픽 환경에서 대화식으로 할 수 있는 기능 등을 제공한다. Fig. OGAP 프로그램의 실행 예를 보여준다. 이 그림에서는 2개와 플랩으로 구성된 고양력 장치에서의 hole cutting 작업을 위하여 주날개 표면에 대한 octree hole-map을 생성하고 그 예를 볼 수 있다.

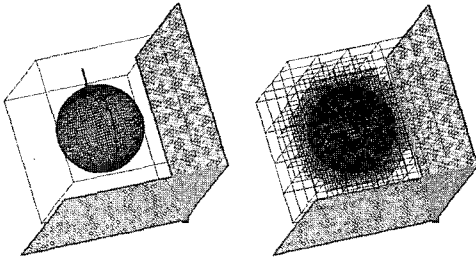
3. 계산 예 및 토론

본 octree hole-map 기법에 의하면 hole finding 작업은 3



(a) Case 1 (b) Case 2 (c) Case 3

Fig. 7 계산예의 격자계



(a) Level=0 (b) Level=4

Fig. 8 구형 물체에 대한 octree hole-map

hole-map 생성 작업과 물체 표면 법선벡터를 이용한 hole point 찾기 작업으로 구성된다. 회귀적 분할을 조금만 할 경우는 hole-map 생성 작업의 양은 줄어들지만, 각 octree 셀에 포함된 부분물체 표면 정보가 많기 때문에 법선 벡터를 이용한 hole point 찾기 계산이 많아지고, 반대로 회귀적 분할을 많이 할 경우 hole-map 생성 작업은 많아지는 반면 각 셀에 포함된 물체 표면 정보가 적어지므로 각 셀당 hole point 찾기 계산이 줄어들게 될 것을 예상할 수 있다. 그러나, 어느 수준의 분할이 이루어지고 나면 그 이상 분할을 하더라도 각 셀에는 최소한의 부분 표면 정보가 유지될 것이므로 과도한 수준의 분할은 오히려 계산 시간의 증가를 가져오게 될 것을 예상할 수 있다. 이를 확인하기 위하여 구형물체를 내부에 갖고 있는 육면체 공간에 격자를 구성하고 물체 내부에 포함된 hole finding을 수행하였다. 구형 물체의 표면 격자 크기와 육면체 격자 크기의 크기를 다음의 Table. 1과 같이 세 가지의 서로 다른 사이즈의 격자계 조합에 대하여 수행하였고, 격자계는 Fig. 7에 예시되어 있다.

이 문제는 구의 크기를 알고 있는 상태에서 육면체 격자점

Table. 1 Grid sizes for the test calculation

	육면체 체적 격자계	구형 표면 격자계
Case 1	26×26×26	51×26
Case 2	51×51×51	101×51
Case 3	101×101×101	201×101

들 중에서 hole cutting이 되어야 하는 대상 격자점을 이미 알고 있는 문제에 해당한다. 본 기법에 의한 계산에서는 이러한 정보를 이용하지 않고 프로그램이 자동적으로 앞에서 설명한 알고리즘에 의해서 hole finding을 수행하도록 하였고, 계산예 중 Case 2에 대한 octree 분할의 예를 Fig. 8에서 보여주었고 있다.

Table. 1과 같이 세 가지 서로 다른 사이즈의 격자계 조합에 대하여 octree 분할 level을 달리 하면서 hole finding에 소요된 계산 시간을 측정하여 이를 Table. 2, 3, 4에 각각 정리하였다.

Table. 2 Case 1: Level별 hole finding 계산시간 (단위:초)

Level	소요시간	Octree 셀 생성	Hole point 찾기	합 계
0		0.0	8.4	8.4
1		0.1	1.5	1.6
2		0.3	0.4	0.6
3		0.5	0.2	0.7
4		0.7	0.4	1.1
5		1.5	1.7	3.2
6		4.9	8.3	13.2
7		20.3	34.5	54.7
8		95.9	135.0	230.9

Table. 3 Case 2: Level별 hole finding 계산시간 (단위:초)

Level	소요시간	Octree 셀 생성	Hole point 찾기	합 계
0		0.0	289.5	289.5
1		0.4	40.0	40.4
2		0.8	5.9	6.7
3		1.6	1.6	3.2
4		2.3	2.6	4.8
5		3.6	13.4	17.0
6		7.5	67.8	75.2
7		21.7	279.4	301.1
8		87.7	1,121.9	1,209.6

Table. 4 Case 3: Level별 hole finding 계산시간 (단위:초)

Level	소요시간	Octree 셀 생성	Hole point 찾기	합 계
0		NA	NA	NA
1		3.5	1,199.6	1,203.1
2		10.8	151.8	162.6
3		20.3	27.7	48.0
4		37.6	22.7	60.3
5		49.9	114.6	164.5
6		63.3	564.1	627.4
7		82.7	2,317.8	2,400.5
8		NA	NA	NA

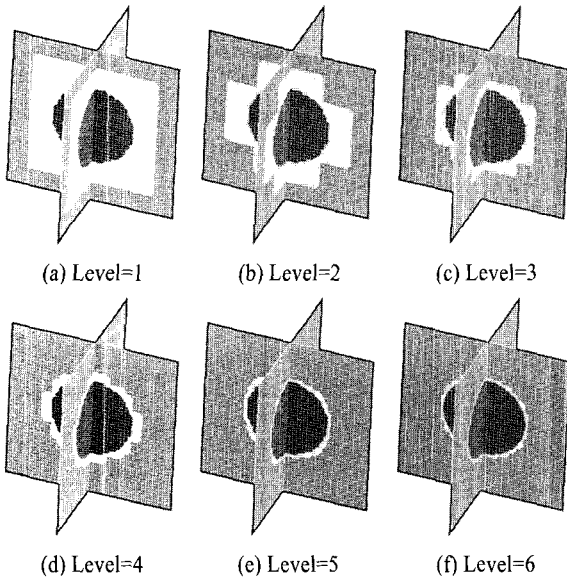


Fig. 9 Octree 분할 level별 hole finding 결과 (Case 2)

본 계산은 Pentium 4(2.66GHz, 1GB)에서 수행되었고, 계산 결과에 따르면 세 가지 계산에 모두 octree 셀 생성은 분할 level이 커질수록 시간이 늘어나는 반면, hole point 찾기는 줄어들다가 다시 커짐을 알 수 있다. 그리고 이 두 가지 작업 소요시간의 합계는 감소하다가 증가함을 알 수 있다. 이 계산에 만을 가지고 최적의 분할 level을 결론짓기에는 아직 이르지만, 적어도 적절한 수준의 octree 셀 분할과 법선 벡터 계산을 조합함으로써 효율적인 hole finding 작업이 가능할 것임을 확인할 수 있다. 본 예제의 경우 물체 표면이 구 형상인 경우로서, level 3이나 4정도의 octree 분할을 하면 구형 내부셀이 생기기 시작하고 물체 표면에 걸친 경계셀들이 어느 정도 물체 표면 형상을 단순하게나마 표현하는 단계가 된다. 즉, 각 경계셀에 관련되는 물체 표면 정보량이 적정량 이하가 되면 서 그러한 경계셀의 숫자가 과도하게 많지도 않은 상황이 된다. 이는 곧 경계셀당 벡터 계산량이 줄어드는 효과와 함께 그러한 벡터 계산이 필요한 경계셀의 숫자를 적정 수준으로 유지함으로써 전체적인 계산 시간을 최소화 할 수 있게 된다.

이처럼 분할의 정도에 따라 전체 계산 소요시간은 달라지는 반면, 본 방법에 의하여 얻어지는 계산 결과 Fig. 9에서 보이는 것처럼 구형 물체 내부점들(그림 내부 진한 부분)은 모두 동일하게 얻어짐을 알 수 있으며, 선택한 분할 level에 따라 표면 법선 벡터 계산을 하기 전에는 경계셀에 속했던 점들 중 벡터 계산 결과 물체 외부점임이 확인된 점들(가운데의 원형 부분을 둘러싼 연한 부분)과 처음부터 외부셀에 속해 있었던 점들(바깥쪽의 진한 부분)을 Fig. 9에서 확인할 수 있다.

4. 결 론

본 연구에서는 octree hole-map의 생성을 통한 중첩 격자에서의 hole finding 작업을 효율적으로 수행할 수 있는 방법에 대하여 살펴보았다. 구체적 수치나 최적 분할 level은 추가적인 연구를 통하여 확인되어야 하겠지만, 다음과 같은 결론을 얻을 수 있다.

- (1) 셀 분할이 거의 없는 경우 hole finding을 위한 벡터 계산량이 증가하므로 전체 계산 시간은 증가한다.
- (2) 셀 분할을 증가시키면 셀당 물체 표면의 정보량이 줄어들므로써 셀당 벡터 계산이 줄어들고, 따라서 어느 정도까지는 전체 hole-finding 계산시간의 감소를 가져온다.
- (3) 그러나, 분할의 정도가 어느 수준 이상이 되면 셀당 벡터 계산 시간은 거의 변화가 없는 반면, 셀 분할에 의한 계산 대상 셀의 증가로 인하여 총 계산 시간의 증가가 야된다.
- (4) 따라서, 본 기법에 따르면 적절한 수준의 octree 셀 분할을 통하여 효율적인 hole-finding 작업이 가능함을 확인할 수 있다.

후 기

본 연구는 스마트 무인기 기술개발사업의 위탁과제 "B2-1 신개념 비행체 공력해석 기술개발" 과제에 의해 수행된 연구 결과의 일부이며, 과제를 지원해 주신 관계자 여러분께 감사드립니다.

참고문헌

- [1] J.L. Steger, F.C. Dougherty, and J.A. Benek, 1983, "Chimera Grid Scheme," *Advances in Grid Generation*, American Society of Mechanical Engineers, Vol.5, p.59-69.
- [2] 이승수, 2004, "MSAP을 이용한 F/A-18C로부터 JDAM 리 해석," *한국전산유체공학회*, 제9권 제1호, p.41-47.
- [3] J.A. Benek, F.C. Dougherty, and P.G. Buning, 1985, "Chimera: A Grid-Embedding Technique," *Armstrong Engineering Development Center*, AEDC-TR-85-64.
- [4] R.L. Meakin, 1991, "DCF3D User's Manual: Version 1.0," *NASA Ames Research Center*.
- [5] S.E. Rogers, N.E. Suhs, W.E. Dietz, 2003, "PEGASUS: An Automated Preprocessor for Overset-Grid Computational Fluid Dynamics," *AIAA Journal*, Vol.41, No.6, p.1037-1044.