

## 論文

## 중첩 격자계에서의 새로운 탐색 알고리즘 개발

박종훈\*, 권장혁\*\*

## Development of A New Searching Algorithm for an Overlapped Grid System

Jong-Hoon Park\* and Jang-Hyuk Kwon\*\*

## ABSTRACT

A new algorithm for fast searching hole points, fringe points, interpolation elements and interpolation coefficients utilized in overlapping grid systems is presented. A fast searching algorithm is required in overlapped grid system since it can occur to pay much time to search hole points, fringe points and interpolation elements. Present algorithm searches fringe points prior to searching hole points in contrast with other algorithms, so fringe points have a sequence and it is not necessary to search hole points. The sequence of fringe points makes it possible to search donor elements fast since it can use the donor element of the prior fringe point for the initial guess of the donor element of the present fringe point. And present algorithm connects the fringe point of current step with the fringe point of previous step in case of relatively moving body problems so that present algorithm for moving body problems is much faster than that of steady problems since it can use the result of the previous step for the initial guess of the current step. Application problems of the present searching algorithm are presented for various 2-D and 3-D overlapped grid systems.

## 초 록

중첩 격자계에서 찾아야만 하는 홀점, 홀 경계점, 내삽 요소, 내삽 계수를 빠르게 찾는 새로운 알고리즘을 개발하였다. 중첩 격자계에서 이러한 부가적으로 찾아야만 하는 과정에 많은 시간을 할애해야만 하는 경우가 종종 발생하므로 빠른 탐색 알고리즘의 개발은 매우 중요하다. 여기에 소개된 알고리즘은 기존의 알고리즘들과는 달리 연속성을 갖는 홀 생성 경계점으로부터 홀 경계점을 먼저 찾기 때문에 홀 경계점들이 연속성을 갖게 되고 홀점을 찾을 필요가 없다. 이러한 연속성은 직전의 홀 경계점의 내삽 요소를 현재 찾고자 하는 내삽 요소의 초기 가정으로 사용할 수 있게 하여 빠르게 내삽 요소를 찾을 수 있게 한다. 또한 본 알고리즘은 상대 운동이 있는 경우에 이전 단계의 탐색 결과를 현 단계의 초기가정으로 사용할 수 있도록 이전 단계의 홀 경계점과 현 단계의 홀 경계점을 연결시켜 주어 상대 운동이 없는 문제보다 더욱 빠른 탐색을 수행한다. 2차원과 3차원 중첩 격자계에 대해 현재의 알고리즘을 적용시켜 보았다.

† 2000년 2월 2일 접수

\*\* 정희원, 한국과학기술원 항공우주공학과

\* 정희원, 서울대학교 항공우주공학과

## I. 서 론

전산 유체 역학의 커다란 두 가지 줄기 중의 하나는 '유동을 어떻게 해석할 것인가'이고 다른 하나는 '유동을 해석하기 위한 격자계를 어떻게 구성할 것인가'이다. 유동 해석의 측면은 컴퓨터의 계산 능력 향상으로 어느 정도 유용한 결과를 얻을 수 있게 되었다. 그러나 유동 해석의 근간이 되는 격자계의 생성의 경우에는 비교적 쉽게 격자계를 생성할 수 있는 2차원에서도 형상이 복잡하거나 상대 운동이 있는 물체의 경우에는 격자계의 생성이 용이하지 않다. 이렇게 형상이 복잡하거나 상대 운동이 있어 격자계의 생성이 용이하지 않은 문제를 해결하기 위해 Steger[1]에 의해 처음으로 중첩 격자계가 고안되었다.

중첩 격자계는 각각의 물체에 대해 각각의 격자를 형성하여 중첩시키므로 격자계의 형성을 용이하게 한다. 이러한 중첩 격자계는 다음과 같은 장점을 갖는다[2,3]. 첫째, 복잡한 형상을 갖는 물체의 정렬 격자를 쉽고 편리하게 생성시킬 수 있다. 둘째, 각각의 격자는 높은 격자 질을 갖는다. 셋째, 복잡하거나 상대 운동이 있는 물체의 유동장을 해석을 용이하게 한다. 넷째, 격자계를 쉽게 덧붙이고 떼어낼 수 있다.

이처럼 중첩 격자계는 많은 장점을 갖는 반면 격자계 상호간의 정보 전달을 위해서 내삽(interpolation)을 이용하므로 보전(conservation)에 문제가 생길 수 있고, 유동 해석에 불필요한 격자계가 물체(solid body)에 접치는 부분을 제외하기 위해 홀점(hole point)을 찾아야 한다. 또한 격자계 상호간의 정보 전달을 위해 홀 경계점(fringe point), 내삽 요소(interpolation element, donor element), 내삽 계수(interpolation coefficient) 등을 찾아야만 하는 부가적인 과정이 필요하다. 이러한 부가적인 과정에 종종 많은 시간을 할애해야만 하는 경우가 발생한다. 그러므로 중첩 격자계를 실제 문제에 효과적으로 이용하기 위해서는 부가적인 탐색 과정에 소요되는 시간이 해석자가 유동을 해석하는데 필요로 하는 시간 보다 빠르게 이루어져야 한다.

따라서 본 연구에서는 중첩 격자계가 부가적으로 필요로 하는 홀점, 홀 경계점, 내삽 요소, 내삽 계수 등의 탐색을 빠르고 정확하게 하는 새로운 방법을 개발하였다. 우선, 2차원 정상 상태에서 기존의 알고리즘들보다 빠르고 정확한 탐색 알고리즘(searching algorithm)을 개발한 후, 이 방법을 기초로 상대 운동이 있는 비정상 상태에서 더욱 빠른 알고리즘으로 발전시켰다. 또한 이

알고리즘을 3차원으로 확장하여 3차원 중첩 격자계에서도 빠른 탐색을 수행하도록 하였다. 특히, 상대 운동이 있는 비정상 유동에서의 빠른 탐색 알고리즘을 개발하는데 연구의 초점을 두었다.

## II. 정상 유동에서의 탐색

### 2.1. 기본 개념

문헌 조사를 통해 살펴 본 결과 선행 연구자들은 모두 다음의 세 단계 탐색을 수행함을 알 수 있었다[4-9].

1. 홀점 탐색
2. 홀 경계점 탐색
3. 내삽 요소 탐색

빠른 탐색을 위해서는 불필요한 연산을 최소화해야 함에도 불구하고 위의 방법은 모든 격자점들에 대해 홀점 여부를 결정해야만 한다. 따라서 홀점 결정을 위해 많은 연산을 수행하게 되고, 홀 경계점이 홀점에 의해 수동적으로 결정됨에 따라 홀 경계점들이 연속적이지 않는다. 즉  $n$  번째 홀 경계점과  $n+1$  번째 홀 경계점은 이웃하지 않는다. 따라서 내삽 요소를 찾는 과정에 많은 시간을 할애해야만 한다. 만약 홀 경계점의 연속성을 얻기 위해서는 부가적인 sorting이 필요하게 된다. 중첩 격자계에서 요구되어지는 탐색 과정에서 가장 많은 시간을 할애해야만 하는 부분이 내삽 요소를 찾는 과정이다. 그러므로 내삽 요소 탐색을 빠르게 수행하도록 하는 것이 빠른 탐색 알고리즘 개발에 매우 중요한 점이 된다. 본 연구에서는 기존의 탐색 알고리즘의 단점을 없애기 위해 다음과 같은 탐색 수행 단계를 밟았다.

1. 홀 경계점 탐색
2. 홀점 결정
3. 내삽 요소 탐색

이와 같은 탐색을 수행하면 다음과 같은 장점이 있다. 첫째, 홀점을 탐색할 필요 없이 홀 경계점을 직접 결정할 수 있어 홀 경계점 내부를 홀점으로 결정하므로 홀점을 탐색하기 위해 필요했던 많은 과정을 없앨 수 있다. 둘째, 홀 경계점을 먼저 찾음으로 인해 홀 경계점들이 일정한 연속성을 갖게되어 내삽 요소를 빠르게 찾을 수 있다. 셋째, 홀 경계점들이 홀 생성 경계(hole creation boundary)로부터 직접 찾아지기 때문에 홀 생성 경계점을 매개체로 각 단계마다 이전 단계의 홀 경계점들과 현 단계의 홀 경계점들을 연결시킬 수 있어 내삽 요소를 빠르게 찾을 수 있

다. 따라서 상대 운동이 있는 격자계의 경우 빠른 탐색을 수행할 수 있다.

### 2.2. 홀 경계점 탐색

홀 생성 경계점은 홀점의 경계를 의미한다. 즉 홀 생성 경계점 내부는 홀점이 되어야 하고 바깥쪽은 정상점이 되어야 한다. 따라서 홀 생성 경계점은 물체를 둘러싸는 적절한 폐곡선이면 된다. 본 연구에서는 부격자계의 격자계 한 라인(O 혹은 C 유형의 격자계에서는 j 라인)이 되도록 하였다. 부격자계가 H-유형인 경우 격자계 한 라인이 물체를 둘러싸는 폐곡선을 이루지 못하므로 물체를 둘러싸는 세 개의 라인이 홀 생성 경계를 이룬다. 따라서 홀 생성 경계점들은 언제나 연속성을 갖고 있다.

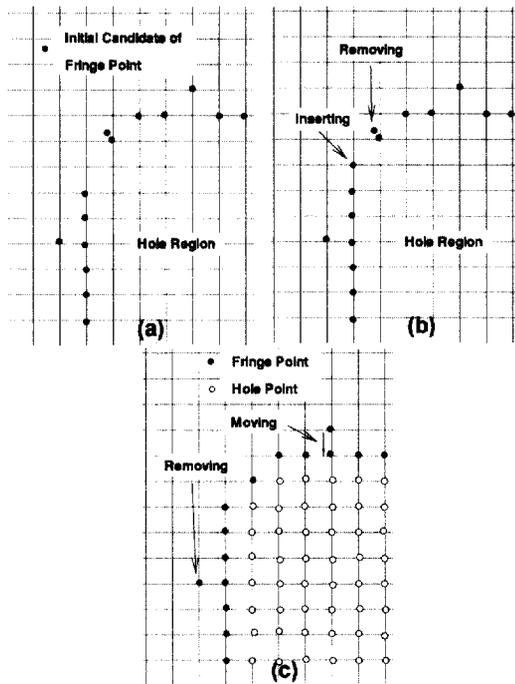


Fig. 1. 홀 경계점의 변화 과정

우선 홀 생성 경계점을 포함하는 주격자계(main grid)의 요소를 찾는다. 그 요소를 구성하는 각각의 격자점들과 홀 생성 경계점이 이루는 벡터와 두 개의 홀 생성 경계점이 이루는 벡터의 내적(3차원의 경우 외적)을 구한다. 이 내적(혹은 외적)의 부호를 이용하면 홀 생성 경계점을 포함하는 요소의 격자점들 중 홀 생성 경계점 바깥쪽에 존재하는 점을 찾을 수 있다. 이러한 점을 홀 경계점 후보로 결정한다. 찾아진 홀 경계점 후보

가 Fig 1의 (a)에 보여지고 있다. 이와 같이 하면 중복되게 찾아지는 홀 경계점도 있고 빠진 점도 생기게 되는데 중복되는 점은 제외하고, 빠진 점들이 있다면 채워주면 그림 1 (b)에서 보여지는 바와 같이 우리가 원하는 홀 경계점들이 연속성을 가지며 찾아진다. 여기서 연속성은 n-1번째 홀 경계점과 n+1번째 홀 경계점은 n번째 홀 경계점 바로 주위에 존재함을 의미한다. 여기서 구해진 홀 경계점들로부터 홀점을 결정하고 홀점을 결정한 후에 요철부와 불필요한 점을 제거하는 과정이 그림 1 (c)에 나타나 있다.

### 2.3. 홀점 결정

기존의 방법들은 홀점을 홀 경계점보다 먼저 결정하므로 전체 격자계에 대해 탐색을 수행해야만 하였다. 그러나 현재의 알고리즘은 홀 경계점이 이미 결정되어 있으므로 단순하고 빠르게 홀점을 결정한다. 기본적으로 홀 경계점 사이의 점들이 홀점이 된다.

가장 아래쪽 라인(line)에 대해 홀 경계점 사이의 점들에 대해 정확하게 홀점을 결정한 후, 위쪽 방향으로 단 한번만 스위프(sweep)하면 전체 면에 대해 정확한 홀점 결정을 수행할 수 있다. 홀점 결정의 정확성은 다음과 같이 수학적 귀납법에 의해 증명된다. 첫째, 가장 아래쪽 라인(n=1)은 홀점이 있을 수 없으므로 홀 경계점 결정만으로 언제나 정확히 결정된다. 둘째, 만약 n번째 라인까지 정확한 홀점 결정을 수행하였다 가정하면 n+1번째 라인에서의 홀점 결정은 홀점과 정상점은 이웃할 수 없다는 사실을 이용하여 정확하게 홀점을 결정할 수 있다. 따라서 모든 영역에서 정확한 홀점 결정을 수행할 수 있다. 좀더 자세한 알고리즘에 대한 설명은 참고문헌 [10]에 기술되어 있다.

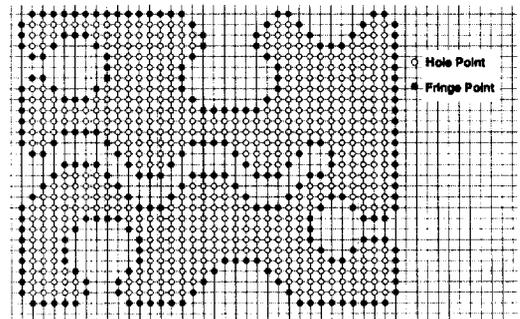


Fig. 2. 복잡한 형상에 대한 홀 결정 결과

Fig 2는 홀 경계점과 홀 경계점 사이의 모든

점이 홀점이 아닌 복잡한 형상의 물체에 대해서 홀점 결정 알고리즘을 적용한 결과이다. 현재의 알고리즘이 정확한 홀점 결정을 수행하고 있음을 알 수 있다.

3차원 문제의 경우, 3차원 형상도 결국 2차원 단면들의 결합으로 생각할 수 있으므로 각각의 단면을 정확하게 결정할 수 있다면 임의의 3차원 형상에 대해서도 정확한 홀점 결정을 수행할 수 있다. 단지 2차원 문제에서는 발생할 수 없는 중앙이 비어 있는 3차원 단면에 대해서도 정확한 결정을 수행할 수 있으면 된다. 그림 3은 이와 같은 3차원적 단면에 대해서도 현재의 알고리즘이 정확하게 홀점을 결정하고 있음을 보여 준다.

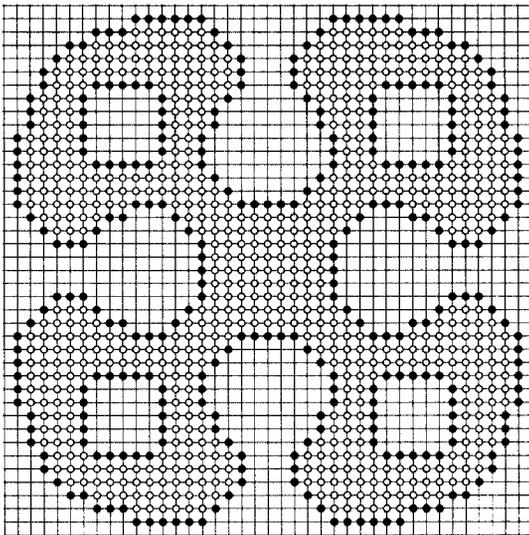


Fig. 3. 3차원 단면에 대한 홀점 결정 결과

홀점 결정이 끝나면 홀점과 정상점에 동시에 인접하고 있지 않은 불필요한 홀 경계점을 제거하고 불필요한 요철이 있는 홀 경계점은 유동 해석시 문제를 일으킬 수 있으므로 이동시켜준다. 그림 1은 홀 경계점의 변화 과정을 보여주고 있다. (c) 과정이 홀점을 결정한 후의 과정을 보여주고 있다. 만약 n번째 홀 경계점이 제거되었다면 n+1번째 홀 경계점이 n번째 홀 경계점이 되고 이 과정은 단순히 필요 없는 홀 경계점을 제거하고 유동 해석시 보전에 문제를 야기 시킬 수 있는 요철부만 제거(홀 경계점 이동)시키므로 '홀 경계점을 먼저 결정하고 홀점을 결정한다'는 현 탐색 알고리즘의 근간을 전혀 해치지 않는다.

### 2.4. 내삽 요소 탐색

현재까지 내삽 요소를 탐색하는 방법에는 안쪽

/바깥쪽 검사 방법, 내삽 계수를 구하여 한 요소씩 이동하면서 테스트하는 구배 탐색 방법 (gradient search)[4], 초기 가정 요소를 홀 경계점에 가장 가까운 점으로 하여 내삽 요소를 탐색하는 Stencil Walk[5,9], 수여 격자계가 속한 공간을 몇 개의 공간으로 나누어 그 공간 내에서 안쪽/바깥쪽 검사를 수행하는 공간 분할 방법(spatial partitioning), 구배 탐색 방법과 공간 분할 방법을 동시에 이용하는 절삭 탐색 방법(search by truncation)[7], 경계점 한정 방법(boundary point localization)[8] 등이 개발되어졌다. 기존의 방법들에 대한 설명은 참고문헌 [10]에 자세히 기술되어져 있다.

Fig. 4. 내삽 요소 탐색 과정

현재의 방법 역시 기존의 방법들과 같이 쌍방 선형 내삽 계수(bilinear interpolation coefficient)를 구하여 찾는다. 그러나 기존의 방법들과는 달리 홀 경계점들이 연속성을 갖고 있으므로 n번째 홀 경계점의 내삽 요소는 n-1번째 홀 경계점의 내삽 요소 주위에 존재한다. 따라서 n-1번째 홀 경계점의 내삽 요소를 초기 가정으로 이용할 수 있어 초기 가정 요소를 다시 탐색할 필요가 없어졌고 가정 요소와 실제 내삽 요소에 가까워 빠른 탐색을 수행할 수 있다. 또한, 정확한 초기 가정으로 선형 내삽 계수를 구하는 Newton-Rapson 방법의 불안정성을 제거하였다. 따라서 기존의 방법들이 한 요소씩을 이동하며 접근했던 것을 그림 4에서와 같이  $\xi, \eta, \zeta$  값을 절삭한 정수값을 이용하여 곧장 다음 요소로 이동할 수 있다. 그러므로 기존의 방법들보다 실제 내삽 요소를 찾기 위해 이동하는 회수를 현저하게 줄일 수 있어 내삽 요소를 빠르게 찾을 수 있다.

기존의 방법들 중 가장 빠르게 내삽 요소를 찾았다고 알려진 방법은 절삭 탐색 방법[7]이다. 이 방법은 가상의 직교 격자계를 전 영역에 대해 구성하여 각각의 직교 격자계를 이용하여 홀 경계점이 될 수 있는 격자점과 내삽 요소가 될 수

있는 요소들을 역사상(inverse map)을 찾아 연결시켜 줌으로써 초기 가정이 필요 없는 방법이다. 그러나 이러한 역사상을 찾는 과정에 많은 시간이 요구되어지고 직교 격자계를 구성하여 연결하므로 많은 양의 부가적인 메모리를 요구한다.

### III. 비정상 유동에서의 탐색

#### 3.1. 기본 개념

기존의 탐색 알고리즘들은 비정상 상태에 대해 다르게 처리하는 알고리즘은 없다. 여기서 비정상 상태라 함은 물체 상호간의 상대 운동이 있는 경우를 나타낸다. 단지 절삭 탐색 방법의 경우 역사상을 한번 찾으면 다시 찾을 필요가 없으므로 이 방법을 이용한 중첩 격자계 상용 코드인 DCF3D의 경우 비정상 상태에서 정상 상태보다 월등히 빠른 탐색 결과를 보여준다.

현재의 알고리즘은 비정상 상태에 대해 정상 상태의 알고리즘을 약간 수정하여 비정상 상태에서 빠른 탐색을 수행하도록 하였다. 기본적인 탐색 방법은 정상 유동에서와 같으나 비정상 유동에서의 탐색은 좀더 빠르고 정확한 탐색을 위해 이전 단계에서 찾은 결과를 얼마나 효율적으로 이용하느냐에 초점이 맞춰진다. 홀점 결정의 경우 이전 단계의 정보를 알고 있다 하더라도 현 단계의 홀점 결정에 아무런 영향을 주지 않는다. 그러나 홀 경계점, 내삽 요소의 경우 이전 단계의 결과를 현 단계의 초기 가정으로 이용할 수 있다면 빠르게 탐색을 수행할 수 있다. 현재의 알고리즘은 내삽 요소가 홀 경계점에 귀속되어 있으므로 이전 단계의 정보를 이용하기 위해서는 현 단계의 홀 경계점이 이전 단계의 어떤 홀 경계점으로부터 이동하였는가를 알아야만 한다. 그러나 불행하게도 이전 단계의 홀 경계점과 현 단계의 홀 경계점을 직접 연결할 수는 없다. 이는 현재와 이전 단계의 홀 경계점 수가 다를 뿐만 아니라 그 순서도 같지 않기 때문이다. 따라서 전체 숫자와 순서가 언제나 같은 매개체를 이용하여 둘 사이를 연결시켜 주어야만 한다. 이런 매개체 역할을 수행하는 것이 홀 생성 경계점들이다.

#### 3.2. 홀 생성 경계점

2차원의 경우 초기의 홀 생성 경계점은 앞에서 언급한 바와 같이 부격자계의 한 라인이 되었다. 정상 상태에서의 홀 생성 경계를 비정상 상태에서 그대로 이용하는 것은 홀 생성 경계점 수

와 홀 경계점 수의 차이가 클 수 있어 약간 비효율적이다. 따라서 비정상 상태에서는 정상 상태에서의 홀 경계점이 속하는 내삽 요소의 중간 점을 홀 생성 경계점으로 이용한다. 이 점들은 실제 홀 경계점 수와 비슷하여 효과적이며, 정상 유동에서 수행하였던 적절한 점을 찾는 수고를 덜 수 있고 그 전체 숫자가 변하지 않으며 순서도 언제나 일정하므로 이전 단계와 현 단계를 연결하는 매개체 역할을 효과적으로 수행하게 된다. 이 점들은 부격자계를 따라서 이동하는 실제 점들이 아니므로 부격자계가 이동한 만큼 이동하도록 하였으며 가상의 점(fictitious point)이라 하였다.

3차원의 경우, 홀 경계점으로는 전체 홀 생성 경계면(hole creation boundary surface)을 구성할 수 없으므로 초기에 정상 상태 탐색에서 주어진 홀 생성 경계면을 그대로 이용한다. 또한 홀 경계점을 연결하는 매개체 역할도 수행한다. 따라서 2차원보다는 약간 비효율적일 수밖에 없다.

#### 3.3. 홀 경계점 연결

홀점 결정 방법이나 홀 경계점 결정 방법은 정상 상태와 같다. 단지 홀 생성 경계점들이 자신을 포함하는 요소를 갖고 있으므로 홀 경계점 결정 과정에서 홀 생성 경계점에 속하는 요소를 찾을 때 이전 단계의 요소를 초기 가정으로 이용하여 빠르게 속한 요소를 찾을 수 있다는 점만 다르다.

그림 5는 현 단계의 홀 경계점과 이전 단계의 홀 경계점이 가상의 점(홀 생성 경계점)을 매개체로 하여 어떻게 연결되는가를 보여주고 있다. 그림 5에서 알 수 있듯이 각각의 홀 생성 경계점으로부터 찾아지는 홀 경계점을 저장하여 이를  $C_1$  연결이라 한다. 또한 반대로 각각의 홀 경계점에 대응하는 홀 생성 경계점을 저장하여 이를  $C_2$  연결이라 한다. 예를 들면  $i$ 번째 홀 생성 경계점으로부터  $n$ 번째 홀 경계점이 찾아졌다면  $C_1(i)$ 는  $n$ 이고,  $C_2(n)$ 은  $i$ 이다. 만약 한 개의 홀 생성 경계점이 여러 개의 홀 경계점이 찾아졌다면, 예를 들면  $i$ 번째 홀 생성 경계점이  $n-1, n, n+1$ 번째 홀 경계점을 찾았다면,  $C_1(i)$ 는  $n-1, n, n+1$  중 어느 한 값을 갖으면 되며,  $C_2(n-1), C_2(n), C_2(n+1)$ 은 모두  $i$ 이다. 만약,  $i-1, i, i+1$ 번째 홀 생성 경계점이 모두  $n$ 번째 홀 경계점을 찾았다면  $C_1(i-1), C_1(i), C_1(i+1)$ 은 모두  $n$ 이고,  $C_2(n)$ 은  $i-1, i, i+1$  중 어느 값을 갖어도 무방하다. 이러한 두 연결을 이용하면 현재의 홀 경계점이 이전 단계의 어떤 홀 경계점으로부터 이동하였는가를 쉽게 알 수

있다. 예를 들면, \*가 이전 단계의 연결을 의미한다고 할 때, 현재 n번째 홀 경계점은 이전 단계의  $C_1^*(C_2(n))$ 번째 홀 경계점으로부터 이동한 점이다.

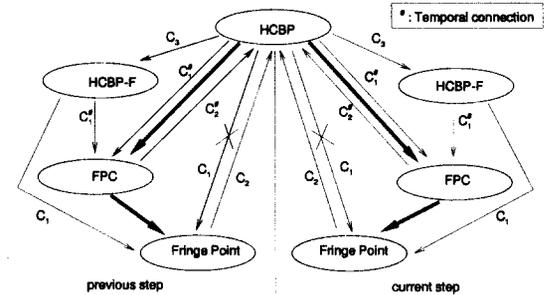
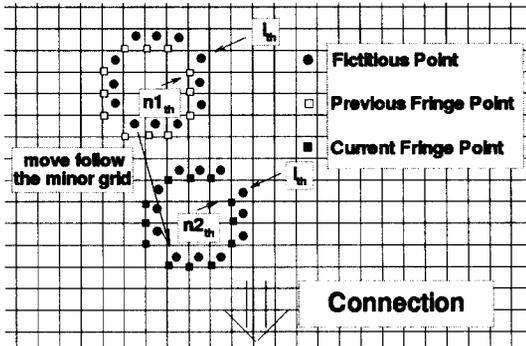


Fig. 6. 3차원 홀 경계점 연결

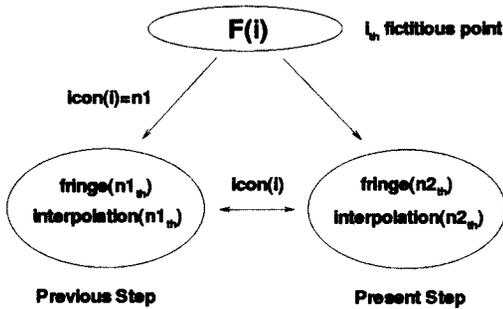


Fig. 5. 2차원 홀 경계점 연결

3차원의 경우 홀 생성 경계점으로 홀 경계점을 이용할 수 없어서 하나의 연결이 더 필요하게 된다. 그림 6은 3차원에서의 홀 경계점 연결을 보여 주고 있다.  $C_3$  연결은 홀 생성 경계점과 홀 경계점 후보를 찾은 홀 생성 경계점을 연결해 주게 되고,  $C_2$  연결은 홀 경계점과 홀 생성 경계점을,  $C_1$  연결은 홀 경계점 후보를 찾은 홀 생성 경계점과 홀 경계점을 연결시킨다. 따라서 현재 n번째 홀 경계점은 이전 단계의  $C_1^*(C_3^*(C_2(n)))$  번째 홀 경계점으로부터 이동한 점이다.

3.4. 내삽 요소 탐색

비정상 상태에서의 내삽 요소 탐색은 두 가지 점에서 정상 상태에서의 탐색과 다르다. 첫째로, 모든 홀 경계점에 대해 이전 단계와 연결을 찾았으므로 내삽 요소의 초기 가정으로 이전 단계의 홀 경계점의 내삽 요소를 이용할 수 있다. 둘째, 모든 홀 경계점에 대해 이전 단계의 내삽 요소 초기 가정으로 이용할 수 있으므로 정상 상태에서 필요하였던 첫 번째 홀 경계점에 대해서도 초

기 가정 요소를 찾을 필요가 없다. 이와 같은 두 가지 다른 점으로 인하여 정상 상태에서의 내삽 요소 탐색보다 빠른 탐색을 수행할 수 있다.

IV. 결 과

현재의 알고리즘을 테스트하기 위해 2차원 증첩 격자계와 3차원 증첩 격자계에 대해 현재의 알고리즘을 적용시켜 보았다. 그림 7은 주격자계가 H-유형이고 부격자계가 NACA0012 O-유형인 경우에 대해 현재의 알고리즘을 적용시킨 결과이다. 그림 7의 (a)는 정상 상태의 탐색 결과이고 (b)는 비정상 상태의 탐색 결과이다. 비정상 상태의 결과는 부격자계가 에어 포일 코드 길이의 절반 이동한 후의 탐색 결과를 보여준다. 부격자계가 이동하여도 주격자계의 형상이 변하지 않는 H-유형의 격자계이므로 정상 상태와 비정상 상태의 결과가 완전히 일치한다.

표 1. 2차원 문제의 탐색 시간 비교

주 격자계	부 격자계	홀 경계점/s	
		정상	비정상
H-유형	NACA O	26626	167308
H-유형	NACA C	29484	180617
NACA O	NACA O	29807	148675
NACA O	NACA C	33523	141997
Sine H	NACA O	30385	114417
Sine H	NACA C	35605	118372
Fine H	NACA O	29252	134375
Fine H	NACA C	36390	145245

그림 8은 부격자계가 NACA0012 C-유형인 경우의 탐색 결과이다. 위의 두 결과에서 보듯이 현재의 알고리즘이 주격자계가 H-유형인 간단한 문제에 대해 정확한 탐색을 수행하고 있음을 알

수 있다. 그림 9는 주격자계가 NACA0012 O-유형이고 부격자계가 NACA0012 C-유형인 경우의 탐색 결과이다. (a)는 정상 상태, (b)는 비정상 상태의 탐색 결과를 보여준다. 주격자계가 H-유형이 아닌 격자계에 대해서도 정확한 탐색을 수행하고 있음을 알 수 있다.

보여 주고 있다. 복잡한 형상을 갖는 3-요소 익형에 대해서도 현재의 알고리즘이 정확한 탐색을 수행하고 있음을 알 수 있다.

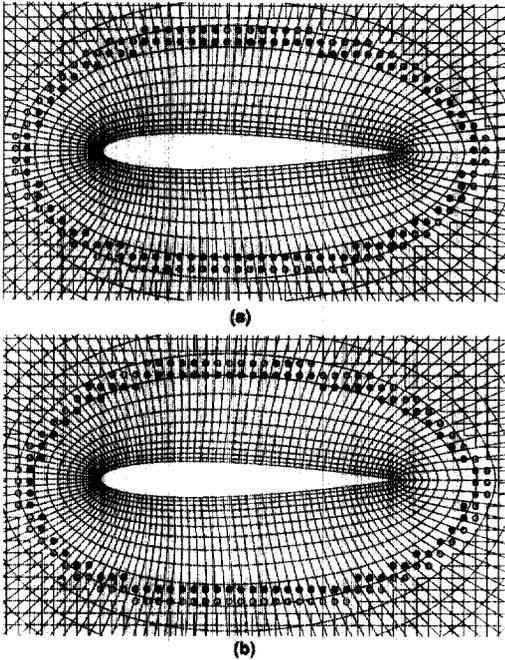


Fig. 7. 주: H-유형, 부: NACA0012 O-유형

이 이외에도 많은 경우에 대해 현재의 알고리즘을 적용시켜 보았다. 표 1은 탐색을 수행해 본 격자계들에 대한 탐색 시간을 보여 주고 있다. 표 1에서 'Fine'은 조밀한 격자계를 의미하며 'Sine'은 사인(sine) 곡선처럼 굴곡이 있는 격자계를 의미한다. 이 결과들은 work station에서 실행한 결과들이다. 테스트한 모든 격자계에 대해 정상 상태나 비정상 상태나 모두 정확한 탐색 결과를 내고 있으며 비정상 상태의 경우가 정상 상태보다 약 5-7배 빠른 탐색을 수행하고 있음을 알 수 있다. 격자계의 유형이나 형상에 대해서는 크게 영향을 받지 않음을 알 수 있다.

좀 더 실용적인 문제에 대해 현재의 알고리즘을 테스트하기 위해 3-요소 익형(3-element airfoil)에 대해 중첩 격자계에 현재 알고리즘을 적용시켜 보았다. 그림 10은 3-요소 익형 주위의 중첩 격자계를 보여 주고 있다. 그림 11은 3-요소 익형의 안쪽 흡 경계점 탐색 결과를 보여 주고 있으며, 그림 12는 바깥쪽 경계점 탐색 결과를

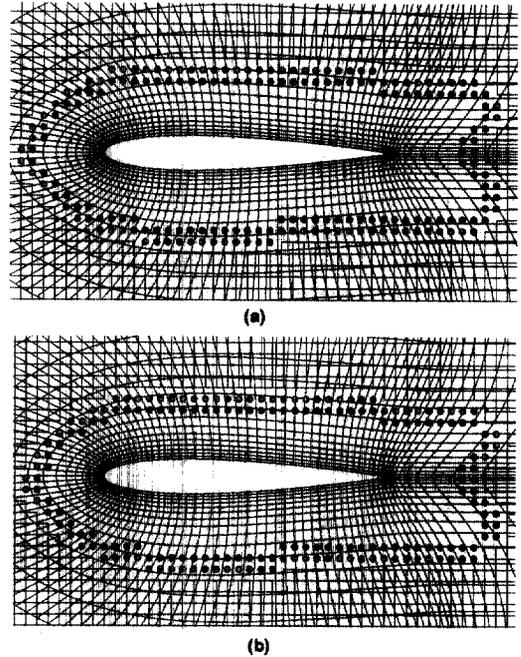


Fig. 8. 주: H-유형, 부: NACA0012 C-유형

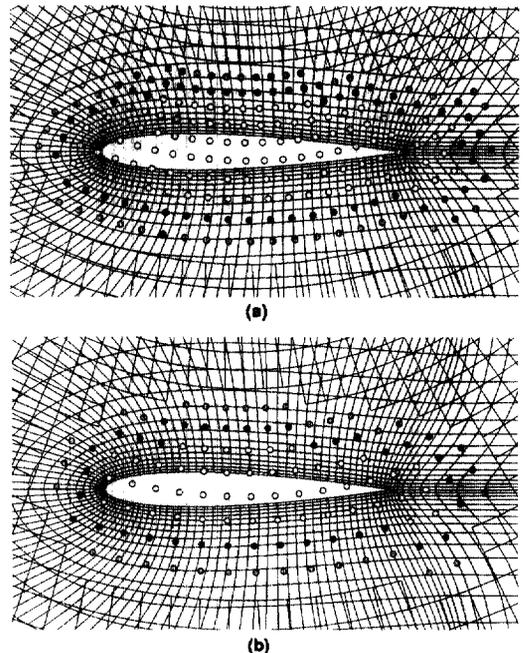


Fig. 9. 주: NACA0012, 부: NACA0012

현재의 알고리즘이 3차원에 대해서도 적용 가능한 지를 알아보기 위해 그림 13과 같은 3차원 타원체에 대한 3차원 중첩 격자계를 구성하여 탐색 알고리즘을 적용시켜 보았다. 그림 14는 정상 상태에 탐색 알고리즘을 적용시킨 결과이고 그림 15는 비정상 상태에 대해 탐색 알고리즘을 테스트한 결과이다. 좌측 상단 그림의 Z축 값을 보면 알 수 있듯이 비정상 상태는 음의 Z축 방향으로 타원체 길이의 절반을 이동한 후의 결과이다. 두 결과로부터 현재의 알고리즘이 3차원에 대해서도 정확한 탐색을 수행하고 있음을 알 수 있다.

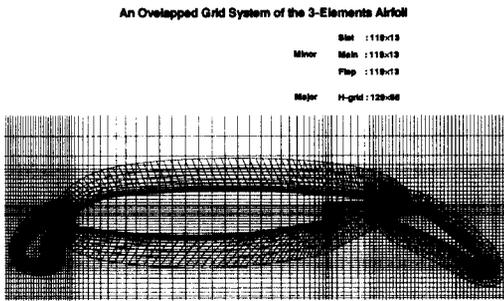
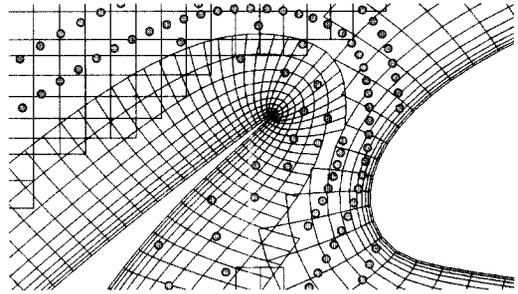
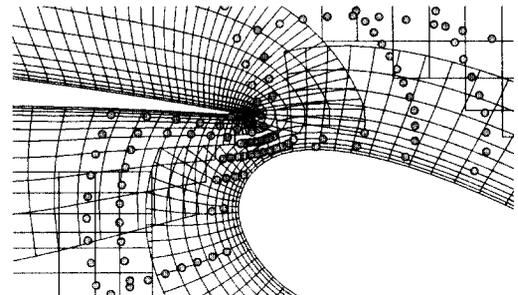


Fig. 10. 3-요소 익형의 중첩 격자계

차원의 경우 홀 생성 경계로 홀 경계점을 이용할 수 없어서 현 단계와 이전 단계의 연결이 2차원의 경우보다 약해져 2차원에 비해 많이 빠르지는 않다.



( a ) : overlapped region, slat with main airfoil

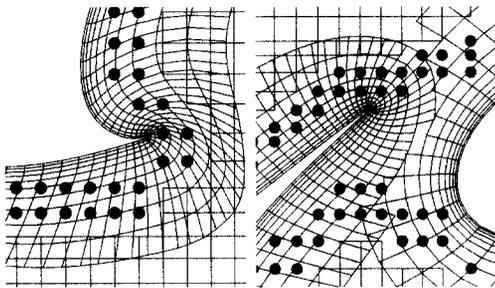


( b ) : overlapped region, main airfoil with flap

Fig. 12. 3-요소 익형의 바깥쪽 경계점

표 2. 3차원 타원체에 대한 탐색 시간

상태	경계점 수	시간(초)	경계점/s
정상	5504	0.29	18979
비정상	5517	0.15	36423

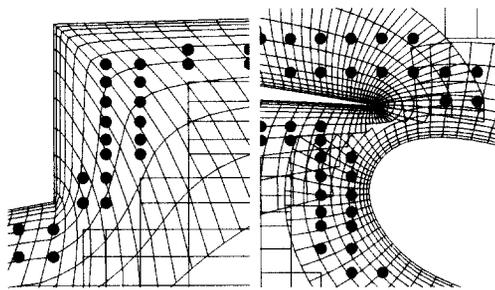


( a )

( b )

표 3. 현재 알고리즘과 다른 알고리즘들과의 탐색 시간 비교

알고리즘	경계점/s	
	정상	비정상
Present	18979	36423
DCF3D	472	3500
PEGASUS	45.3	123



( c )

( d )

Fig. 11. 3-요소 익형의 안쪽 경계점

표 2는 타원체에 대한 탐색 시간을 보여 주고 있다. 비정상 상태가 정상 상태보다 약 2배정도 더 빠른 탐색을 수행하고 있음을 알 수 있다. 3

표 3은 현재의 알고리즘과 다른 알고리즘들의 결과를 비교하고 있다. 여기서 DCF3D 코드는 구배 탐색 방법과 공간 분할 방법을 동시에 이용하는 절삭 탐색 방법을 이용하고 있다[7]. PEGASUS 코드의 결과는 참고 문헌 [7]을 참고 하였다. DCF3D 코드와 PEGASUS 코드는 CRAY-2에서 실행한 결과이고, 현 알고리즘의 결

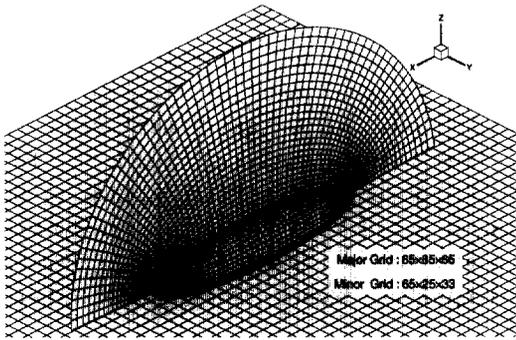


Fig. 13. 3차원 타원체의 중첩 격자계

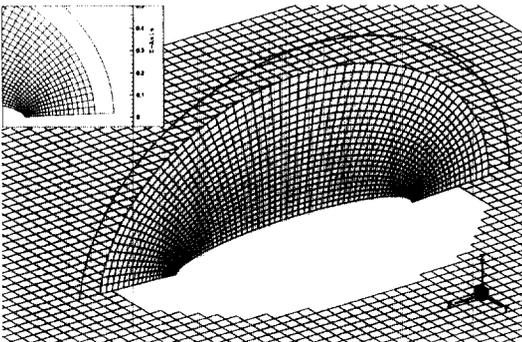


Fig. 14. 3차원 타원체에 대한 정상 상태 탐색 결과

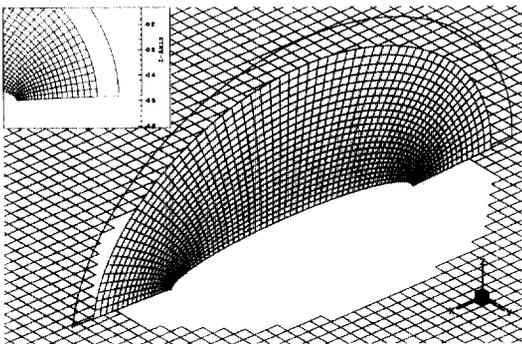


Fig. 15. 3차원 타원체에 대한 비정상 상태 탐색 결과

과는 HP-workstation의 결과이다.

중첩 격자계는 격자계에 따라 탐색 시간에 차이를 보일 수 있어 단순 비교는 어렵다. 그러나 기존의 알고리즘 중 가장 빠르다고 알려진 절삭 탐색 방법을 적용한 상용 코드인 DCF3D의 탐색 결과보다 현재의 알고리즘을 이용한 탐색 결과가

비정상 상태에서 약 10배가 정상 상태의 경우 약 40배 빠름을 알 수 있다. 이는 현재의 알고리즘이 기존의 방법들이 택하지 않았던 홀 경계점을 홀점보다 먼저 찾음으로 인해 얻어지는 결과이다. 단순한 비교는 힘이 들겠지만 위의 결과들로부터 현재의 알고리즘이 빠른 탐색을 수행하고 있다고 할 수 있다. 참고로 DCF3D의 경우 비정상 상태가 정상 상태에서의 탐색에 비해 약 8배 빠른 탐색을 수행하는 것은 앞에서 설명했듯이 역사상을 찾을 필요가 없어지기 때문이다.

### V. 결 론

본 연구에서 중첩 격자계에서 꼭 필요한 홀점 결정, 홀 경계점 결정, 내삽 요소 결정, 내삽 계수 결정 등의 탐색 과정을 빠르게 수행할 수 있도록 하는 새로운 알고리즘을 개발하였다.

새로 개발된 탐색 알고리즘의 특징은 홀 생성 경계점으로부터 홀 경계점을 홀점보다 먼저 결정하여 홀점 결정이 빠르고 정확하며, 홀 경계점들이 연속성을 갖는다. 이러한 연속성은 내삽 요소 및 내삽 계수를 빠르게 찾아준다. 또한 정상 상태와 비정상 상태를 홀 생성 경계점을 매개로 하여 연실 시킴으로 비정상 상태에서 매우 빠른 탐색을 수행한다.

2차원에서 개발된 알고리즘을 3차원으로 확장하여 3차원에서도 빠른 탐색 결과를 보이고 있다. 또한 본 탐색 알고리즘은 2차원 비정상 탐색 알고리즘을 3차원에 적용하여 3차원에서도 비정상 상태에서 정상 상태보다 빠른 탐색을 수행한다.

### 참고문헌

1. Steger, J., L., Dougherty, F., C., and Benek, J., A., "A Chimera Grid Scheme," in Ghia, K., N., and Ghia, U.(eds), *Advancing in Grid Generation*, FED Vol. 5, ASME, New York, pp. 59-69, 1983.
2. Cho, K., W., and Kwon, J., H., "A study of Structured Overlapped Grid Generation Method using Interactive Shooting Method", *Korea-France Joint Conference on Military Aircraft Technology*, Taejoen, Korea, pp. 44-51, 1997.
3. 조금원, 권장혁, "A Study of the Generation Mesh Interface Regions for Structured

- Overlapped Grids", 전산유체공학회 춘계학술대회, pp. 222-228, 1997.
4. Benek, J., Steger, J., Dougherty, F., and Buning, P., "Chimera: Agrid-Embedding Technique," AEDC-TR-85-64, 1986.
  5. Dougherty, F. C., "Development of a Chimera Grid Scheme with Application to Unsteady Problems", Ph.D. Dissertation, Stanford Univ. April, 1984.
  6. Milgram, M. S., "Does a Point Lie Inside a Polygon?", Journal of Computational Physics 84, pp. 134-144, 1989.
  7. Meakin, R., N., "A New Method for Establishing Intergrid communication among Systems of Overset Grids", AIAA Paper 91-1586-CP, 1991.
  8. Ismail, H., T, "A 2-D Unsteady Navier-Stocks Solution Method with Overlapping/Overset Moving Grids", AIAA Paper 96-0822, 1996.
  9. Dietz, W., Jacocks, J., and J. Fox, "Application of Domain Decomposition to the Analysis of Complex Aerodynamic Configurations", SIAM Conf. Domain Decomposition Meths., Houston, TX, March, 1989.
  10. 박종훈, "중첩 격자계에서의 새로운 탐색 알고리즘 개발", 한국과학기술원 석사 학위 논문, 1998.