

자동화된 CFD 적용을 위한 통합형 S/W의 개발에 관한 연구

Development of an Integrated S/W for Automated CFD Application

김 기연¹⁾, 김 병수²⁾

Kiyou Kim, Byoungsoo Kim

In this paper, the on-going effort and progress for developing an integrated software for automated CFD application is described. As an outcome of the effort devoted so far, a new system, ICFDIT, is developed and introduced in this paper. The new system can be used to solve fluid dynamics problems in a convenient graphical environment, and it includes a pre-processor, a main-processor, and a post-processor. Usage of the system and examples are demonstrated, and some issues for improvement of the system are discussed.

1. 서론

컴퓨터 기술의 발전, 유동해 알고리즘의 개선, 그리고 빠른 데이터 가시화 기법의 발전 등의 복합적인 배경으로, 컴퓨터를 이용하여 유동현상을 수치 모사/해석하는 응용역학인 전산유체역학(Computational Fluid Dynamics, CFD)은 그 활용도가 점점 커지고 있는 실정이다. 전산유체역학에 의한 유동 해석을 위해서는, 연속체로 가정된 유동영역을 유한한 개수의 계산점, 즉 격자 격자로 대체하고 각 격자에서의 유동 지배 방정식의 해를 수치적 기법으로 계산하게 된다. 이러한 일련의 유동 해석 과정은 크게 전처리/유동장계산/후처리의 세 개의 과정으로 나누어 볼 수 있다.

최근의 많은 연구로 격자 생성 기법이나 유동해 기법 등에 많은 발전이 있어온 것은 주지의 사실이고, 컴퓨터 성능의 획기적인 발전으로 엔지니어링을 위한 실제적 도구로서 전산유체역학이 활용되고 있다. 물론 난류 문제나 복잡한 3차원 유동에 대한 문제 등 현재로서도 그 해결의 기미가 쉽게 보이지 않는 문제들이 있긴 하지만, CFD 관련 상업용 소프트웨어들이 여러 분야에서 광범위하게 사용되고 있다. 그러나, 이러한 상업용 소프트웨어의 대부분은 격자 생성 소프트웨어와 유동장 해석 코드가 별개로 되어있고, 외국에서 개발되어 국내에 수입 활용되고 있는 대부분의 상업용 코드가 소스-코드 없이 실행 파일만 제공됨으로써 해결 대상 문제에 따른 프로그램의 적절한 변경이나 효율 극대화가 불가능하고, 이는 곧 기술 의존의 심화 가능성마저 내포하고 있다. 이와 같은 문제점에 대응하기 위해서는, 기존의 CFD 관련 외국 S/W와 비슷한 기능의 제품을 국내 기술로 개발하는 노력이 시급하기도 하거니와, 후발주자로서의 핸디캡을 극복하기 위하여 다른 접근 방식의 S/W 개발에 대한 시도 또한 의미있는 작업이라 하겠다.

본 연구는 그러한 작업을 위한 초기 단계라고 할 수 있고, 본 논문에서는 전산유체역학에 의한 유동 해석의 전과정을 일괄 처리할 수 있는 자동화된 통합형 S/W의 개발을 지향하면서 이루어진 그간의 연구 성과와 현재까지 구현된 시스템을 소개하고, 본 연구과정에서 파악된 문제점등을 살펴봄으로써 향후 연구 개발의 방향 등에 대해서 살펴보고자 한다.

2. 본론

본 연구가 지향하는 자동화된 CFD 적용을 위한 통합형 S/W는, 유동 해석 작업의 전처리 단계인 격자 생성과 주 단계인 유동장 계산, 그리고 유동장 반복 계산중이나 작업후의 데이터 분석을 위한 후처리 단계인 데이터 가시화용 프로그램을 모두 포함하는 형태로서, 이러한 S/W의 개발을

1) 충남대학교, 항공우주공학과 (305-764, 대전시 유성구 궁동 220, 전화: 042-821-7775)

2) 충남대학교, 항공우주공학과 (305-764, 대전시 유성구 궁동 220, 전화: 042-821-6686, bskim@aero.chungnam.ac.kr)

위해서는 각 단계별로 필요로 하는 여러 가지 기능을 골고루 갖추어야한다. 각 단계별로 기능 구현에 사용된 기법과 전체적인 시스템에 대해서 살펴보면 다음과 같다.

2.1. 격자 생성

본 연구에서 전처리 단계라 함은 유동해 계산의 대상인 유한한 크기의 유동장에 적절한 형태의 격자를 분포시키는 작업을 의미한다. 전산유체역학에서 사용되는 격자의 종류에는 여러 가지가 있으나, 크게 비정렬 격자와 다중-블록 정렬 격자의 두 가지로 구분할 수 있다. 전산유체역학 분야에서 상대적으로 뒤늦게 활용되기 시작한 비정렬 격자는 나름의 장점이 있긴 하지만, 현재까지도 다중-블록 정렬 격자가 보편적으로 더 많이 쓰이는 격자 형태이다. EAGLE, GRIDGEN 등 여러 가지 종류의 다중-블록 정렬 격자 생성 코드들이 상용화되거나 사용되고 있다. 한편, 이와 같은 다중-블록 정렬 격자 생성 코드들을 사용함에 있어서, 사용자들은 먼저 대상 유동장을 여러 개의 블록으로 나누고, 각 블록의 경계들을 정의하여야 하는데, 이 과정은 많은 경험과 수작업을 요하는 작업으로서, 격자 생성 작업 또는 나아가 전산유체역학의 원활한 적용을 어렵게 하는 주된 요인이 된다. 본 연구의 지향 목표가 가능한 한 모든 과정을 자동화하는 것으로서, 이와 같은 블록 생성 작업을 자동화 하기 위하여 Kim이 제시한 자동 격자 생성 기법을 적용하였다.[1-2] 그 기법은, 주어진 물리 영역의 유동장 경계는 그 형상에 관계없이 여러 개의 꼭지점을 가진 위상학적 다각형으로 생각할 수 있다는 점에 근거하고 있다. 즉, 기하학적 다각형은 각 꼭지점간의 변이 항상 직선이어야 하나, 위상학적 다각형은 각 꼭지점사이의 변이 곡선이나 호 등 일반적인 형태의 선분일 수 있다는 것이다. 그리고, 유동장의 블록화는 결국 해당 유동장을 상대적으로 단순한 형태의 여러 개의 위상학적 사각형으로 채워넣는 것이므로, 유동장을 몇 가지의 기본 원칙에 따라 반복적으로 분할하여 더 적은 수의 꼭지점을 가진 여러 개의 위상학적 다각형으로 나누어 가는 기법을 활용하여, 궁극적으로 위상학적 사각형들로만 유동장이 채워질 때까지 자동적으로 유동장 분할을 수행하는 기법이다. 이 기법에 대한 구체적 기술은 Kim의 참고문헌에 나와있다.[1-2]

2.2. 유동장 계산

유동장 계산 결과가, 사용된 격자의 질이나 종류에 따라서 많이 좌우되고 격자 생성 작업이 많은 man-hour를 필요로 하는 것은 사실이나, 유동장 계산과정이 곧 CPU시간의 대부분을 차지하기 때문에 효율적이고 빠른 알고리즘의 적용이 필요하다. 본 연구에서는 기존에 잘 알려진 기법들을 이용하여 solver를 구현하였다. 공간 이산화에는 Roe의 Flux-Difference-Splitting에 근거한 풍상차분법, 시간 적분에는 LU-SGS 알고리즘을 사용하였고, 유한체적법으로 구현함으로써 블록들간의 유량 보존이 이루어지도록 하였다.

2.3. 격자 적용

격자 적용에 대한 필요성의 이유에는 여러 가지가 있겠으나, 그 중 하나는 유동장의 해를 알기 이전에 유한한 수의 격자점을 대상 유동장에 분포시켜야 하고, 그렇게 분포시킨 격자점들이 특정 유동장의 해를 구하는데에 있어서 “최선”이 아닐 수도 있다는 점에 있다. 특히, 본 연구가 지향하는 전처리 단계에서의 격자 생성 기법은 사용자의 입력을 최소화하는 것이므로, 특정 문제를 대상으로 격자 수나 조밀도 등을 수작업을 통하여 조절하는 기존의 격자 생성 작업 방식과는 다르게 프로그램에 제공된 결정 법칙에 근거하여 격자를 생성하므로, 그러한 격자계는 그 자체로는 격자점의 효율적 분포나 적절한 조밀도 등에서 만족할 만한 질을 기대하기는 어렵다. 그러나, 일단 적절한 정확도의 유동해가 있다면 그 해의 분석을 통하여 대상 유동장에 “최선” 내지는 “보다 나은” 격자계의 구성이 가능하게 될 것이다. 또 다른 이유로는 격자간격과 해의 length scale과의 관계에 있다. 구하고자 하는 어떤 해의 length scale이 격자간격보다 작은 경우 그러한 length scale을 가지는 특성은 계산되어질 수 없기 때문에 이를 해결하기 위해서는

격자구성에 있어 격자간격을 length scale보다 작게 구성해야 한다.[3] 이렇게 할 경우 컴퓨터 용량을 많이 차지하며, 또한 계산 시간도 늘어나야 한다. 예를 들어 어떤 한 물체의 표면을 지나는 점성유동이 이에 해당한다. 경계층 유동의 해석에서 격자간격이 충분히 작지 않으면 Heat transfer, skin friction값들은 의미를 가지지 못하게 된다.[4]

본 연구에서는 가용한 유동해를 이용하여 유동변수의 분포 양상에 따라 격자를 적절히 재분포하는 유동해 적응(solution adaptation) 기법을 구현함으로써, 전처리 단계에서 생성된 격자의 질을 개선하도록 하였다. 격자점의 재분포는 유동변수의 격자선을 따른 변화율의 균일분포 조건을 적용함으로써, 유동해의 변화가 급격한 부분에서는 격자 간격이 조밀하게 되고, 변화가 완만한 부분에서는 격자 간격이 커지게 되도록 함으로써 이루어진다. 이 기법에 대한 구체적 기술은 참고문헌에 나와있다.[5] 나아가, 격자점의 전체 숫자를 늘리거나 줄이는 기능을 구현함으로써 사용자가 이 기능을 통하여 원하는 정확도의 유동해를 얻을 수 있도록 하였다.

2.4 시스템 소개

아직도 완벽하지 않아서 개선의 여지가 많이 있긴 하지만, 현재까지의 연구 결과로 만들어진 S/W의 이름을 ICFDIT (Interactive CFD with Integrated Tool)이라고 부르기로 한다. ICFDIT은 크게 launch module과 work module들로 이루어져 있고, ICFDIT은 전체 시스템의 명칭이자 하부 launch module들을 총괄하는 launch module이기도하다. 하부 launch module은 단계별로 IPREIT(pre-process), IMAINIT(main-process), 그리고 IPOSTIT(post-process)로 이루어져 있고, 이러한 launch module들은 단위 work module들을 발진시키는 역할을 하고, 작업 흐름상 필요한 후속 work module을 각 work module들이 발진시키기도 한다. Work module들에는 전처리 단계의 ISTARTIT, IBOUNDIT, IBLOCKIT, IGRIDIT, IOPTIMIZEIT 등이 있고, 유동 해석 단계의 ISETIT, ISOLVEIT, IADAPTIT등과 후처리 단계의 IPLOTIT이 있다. 이와 같은 ICFDIT의 module 구성과 작업 흐름도가 Fig. 1에 나와있다.

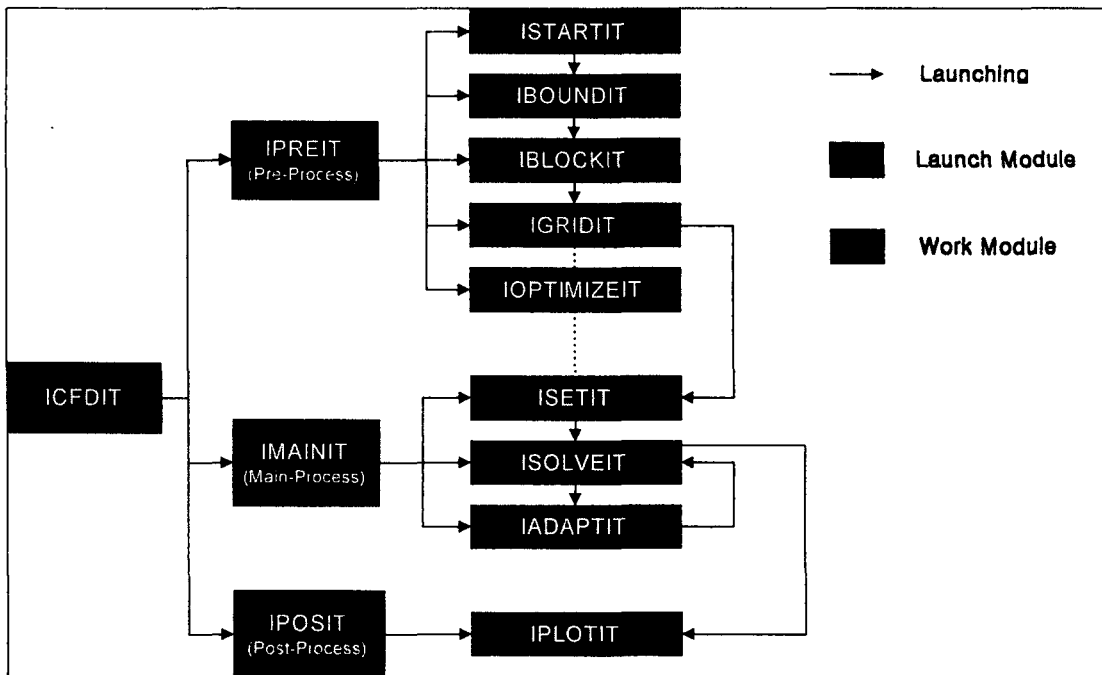


Fig. 1 ICFDIT의 구성모듈과 작업흐름도

본 시스템의 GUI는 대부분의 workstation들이 기본적으로 지원하는 X-window system에 근거하고 있고, X11/Motif/Xt-Toolkit 그래픽 라이브러리들을 사용하여 구현하였다. ICFDIT 모듈과

단계별 launch module들의 형태가 Fig. 2 에 나와있다. 상업용 시스템들의 화려한 모습에 비하면 현재로서는 무미건조해 보이지만, 필요한 작업의 수행에는 별 지장이 없다.

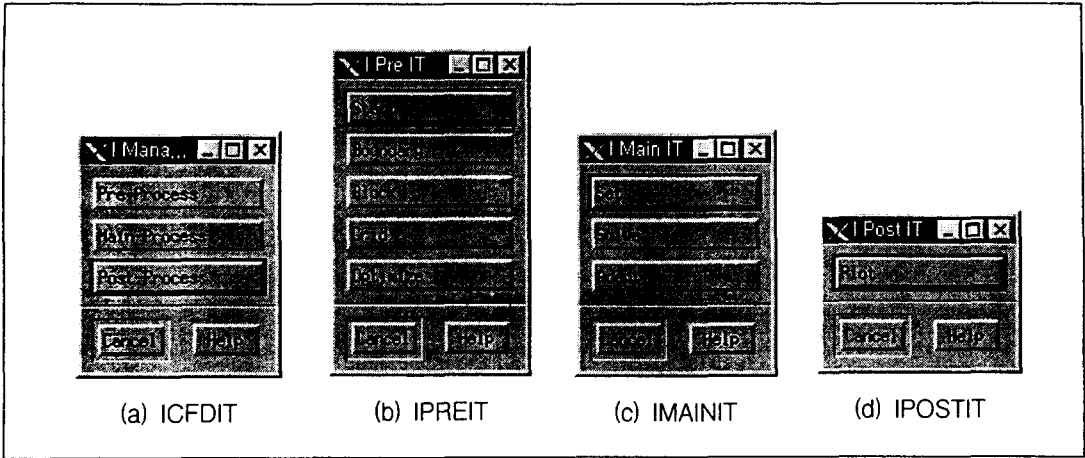


Fig. 2 Launch module들의 작업 환경

여러 가지의 work module에서 이루어지는 작업을 정리하면 다음과 같다.

- **ISTARTIT**: 작업의 고유 명칭을 부여함으로써 후속 작업들의 일관성 유지를 가능케한다.
- **IBOUNDIT**: 유동장 경계 형상을 설정하는 작업을 하는 module로서, 여러 가지 기본적인 선 요소와 에어포일 등의 물체 형상 입력을 2차원 CAD와 같은 환경에서 할 수 있다.
- **IBLOCKIT**: 입력된 경계 형상을 이용하여 유동장을 다중 블록으로 구성하는 작업을 한다.
- **IGRIDIT**: 블록 경계들로부터 블록간 격자선의 C^0 -연속성을 만족시키는 다중-블록 격자를 생성하는 작업을 하며, 대수형 격자 생성 기법을 기본으로 하고, 타원형 격자 생성 기능을 가지고 있다.
- **IOPTIMIZEIT**: 경우에 따라서는 블록의 수나 블록의 위상학적 형상이 유동해 수렴 속도등에 영향을 미치게 되는데, 이러한 경우를 대비하여 블록의 최적화 작업을 할 수 있는 module인데, 현재로서는 구현되지 않는다.
- **ISETIT**: 유동장 solver에 필요한 여러 가지 유동 조건이나 solver option들을 선택 및 입력하고, 경계조건등을 그래픽 환경으로 입력하는 작업을 한다.
- **ISOLVEIT**: 유동장 계산을 실행하고, 계산 수렴 상황등을 모니터링하는 모듈로서, 계산중 필요하면 작업을 일시 중단하거나 중단되었던 계산을 재개, 또는 처음부터 다시 시작시킬 수 있는 기능이 있다.
- **IADAPTIT**: 유동장 계산 결과를 이용하여 격자 적응을 그래픽 환경에서 실행한다. 격자가 적응된 후에는 ISOLVEIT을 발진시켜 높은 정확도의 유동장 계산을 실행하게 된다.
- **IPLOTIT**: 유동장 계산 중이나 계산 후 유동장의 분석에 사용되는 module로서 기존의 상업용 가시화 프로그램들에 비하여 그 기능은 많지 않지만, 필요한 정보 입수나 유동장 계산 상황에 대한 모니터링 작업을 하는데 유용하다.

단위 작업을 수행하는 이러한 work module들은 해당 작업의 종류에 따라서 필요한 기능들이 풀-다운 메뉴방식으로 구현되도록 하였는데, 그 외형적인 작업 환경은 주메뉴의 약간의 차이들만

있을 뿐 대략 비슷하다. 그 중 IBOUNDIT의 작업 환경이 선요소 입력이 정확하게끔 해주는 snapping 기능을 위한 snapping point들을 ON된 상태로 Fig. 3(a)에 나와있고, IGRIDIT의 작업 환경이 Fig. 3(b)에 격자 변경 기능을 위한 풀-다운 메뉴를 보여주는 상태로 나와있다.

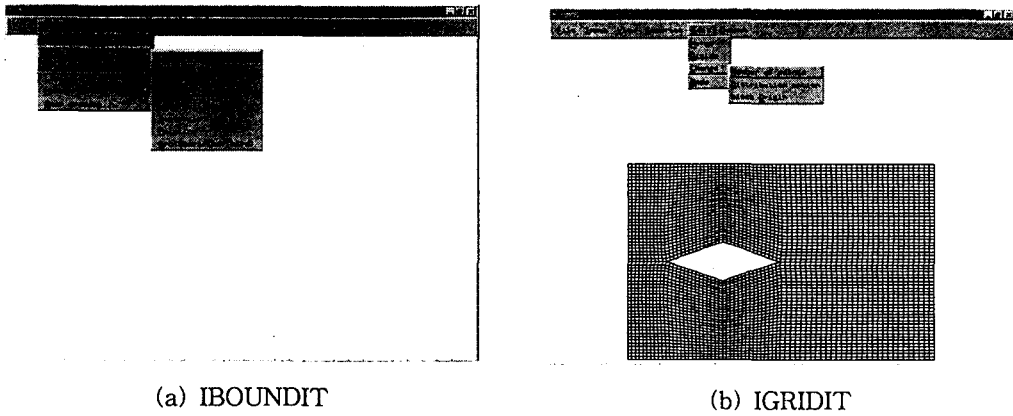


Fig. 3 Work module의 작업 환경 예

3. 활용에 및 토론

이와같은 ICFDIT 시스템을 이용한 계산예로서, 폭이 5인 channel 내부 중심에 가로가 5, 세로가 1인 다이아몬드 형상의 물체가 마하수 2.5인 유동장 속에 놓였을 때의 비점성 유동 계산 과정을 작업의 흐름에 따른 work module별로 Fig. 4에 보여주고 있다. 우선 Fig. 4(a)에 IBOUNDIT에 의해 snapping 기능을 이용하여 손쉽게 생성된 유동장 경계를 보이고 있고, (b)에는 IBLOCKIT에 의해 자동적으로 생성된 블록 경계가 나와있다. (c)에는 IGRIDIT에 의해 생성된 대수형 격자를 보여주고 있는데, IGRIDIT이 자동적으로 블록 사이즈에 따른 적절한 격자수 결정 능력이 없기 때문에 default로 설정된 블록별 격자점 수를 사용자가 상황에 맞게 변경시킨 결과이다. 이와같이 (a),(b), 그리고 (c)에 해당하는 작업이 전처리 단계에서 이루어지는 작업에 해당한다.

Fig. 4(d)에는 ISETIT에서의 유동 경계조건 입력예를 보여주고 있는데, 블록간 경계는 자동적으로 처리가 되므로, 실제 유동장 경계에 대한 경계 조건 입력 작업을 마우스 클릭만으로도 손쉽게 실행할 수 있음을 예시하고 있다. (e)에는 초기 격자계에 대한 유동 계산의 수렴상황을 log-scale로 보여주는 ISOLVEIT의 작업을 보여주고 있다. (f)에는 유동 계산의 결과를 IPLOTIT을 이용하여 등마하수 선도로 보여주고 있는데, 웨지 형상에 대한 경사충격파와 벽면에서의 반사충격파, 팽창파, 뒷진충격파등을 비롯하여, 충격파끼리의 교차에 의한 굴절, 팽창파의 충격파에 의한 굴절등을 잘 보여주고 있다. 한편, Fig. 4(c)에서 볼 수 있듯이 초기 격자는 유동장의 실제 상황과는 무관하게 등간격의 격자로 생성되어 있음을 알 수 있다. 이와 같은 등간격 격자로는 충격파등의 높은 유동 구배를 정확히 계산하는데에 적절치않다. 따라서, Fig. 4(g)에 유동장 계산 결과를 이용하여 IADAPTIT에 의해 격자 적응을 실행한 후의 격자계를 보여주고 있다. 이처럼 적응된 격자계를 이용하여 다시 유동장 계산을 했을 때의 반복 계산의 수렴 상황과 최종 결과의 등마하수 선도를 각각 Fig. 4(h)와 (i)에서 볼 수 있다.

한편, IADAPTIT의 기능중에는 격자점 재분포에 의한 격자 적응만이 아니라 격자 밀도를 증감시킬 수도 있는데, Fig. 4(j)에는 다시 한번 격자 적응을 통한 재분포 후 격자 밀도를 각방향으로 2배, 즉 4배한 격자계를 보여주고 있다. 그리고, 계산 과정의 수렴 상황과 최종 계산 결과를 Fig. 4(k)와 (l)에서 보여주고 있다.

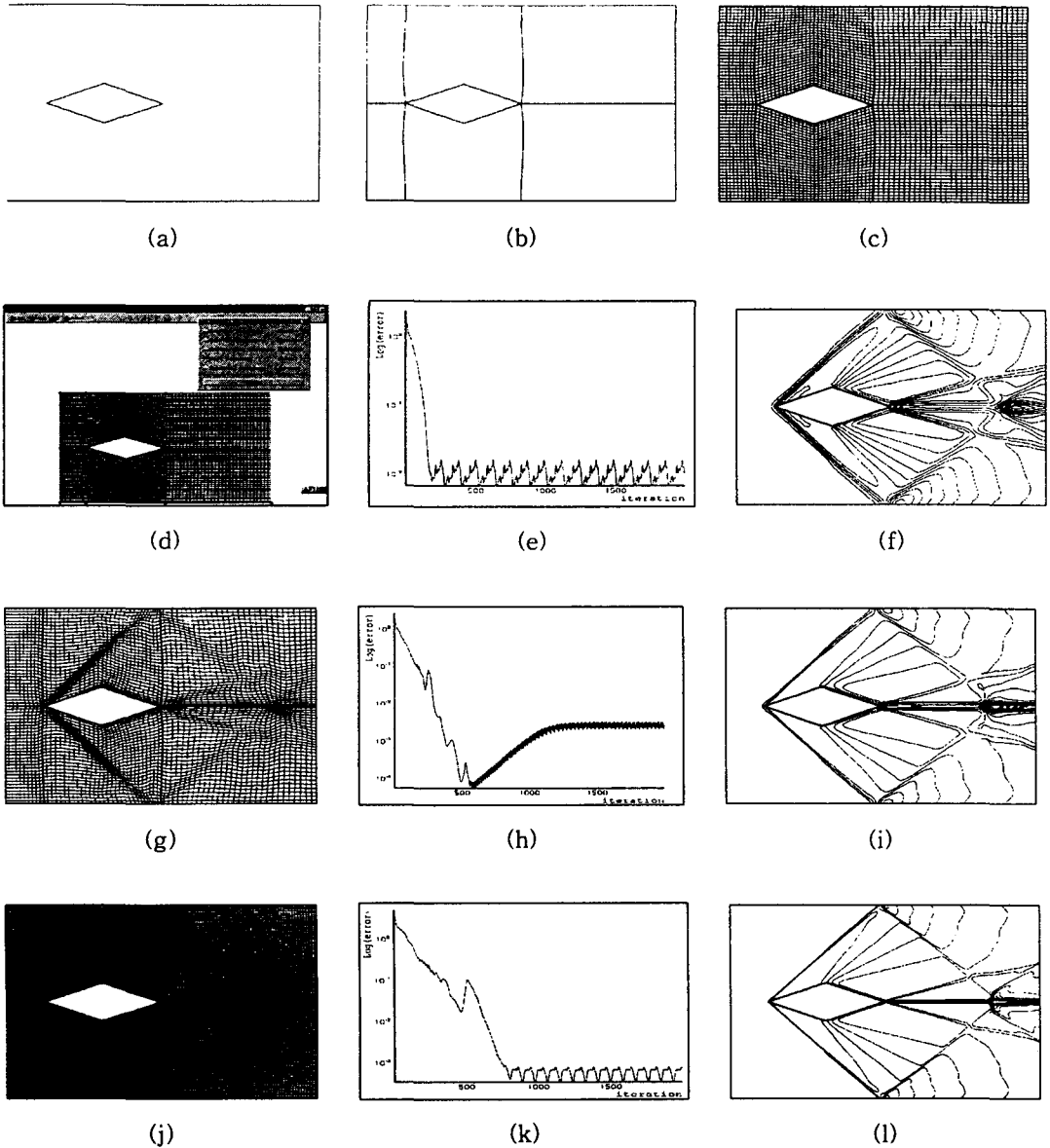


Fig. 4 ICFDIT에 의한 작업 예

Fig. 4(f)와 (i), 그리고 (l)로부터 확인할 수 있듯이 격자 적용을 통하여 충격파가 점점 정확하고 선명하게 잡히고 있음을 확인할 수 있고, 이러한 일련의 유동해 개선 과정이 ICFDIT이 제공하는 그래픽 환경에서 손쉬운 마우스 클릭으로 구현될 수 있음은 본 시스템의 커다란 장점이라고 할 수 있겠다. 물론, 여러 가지 면에서 개선이나 연구의 여지가 없는 것은 아니다. 그 중 몇가지를 살펴보면, 우선 유동해 계산 과정에서 반복계산 오차의 수렴여부를 문제의 종류에 상관없이 타당하게 결정할 수 있는 방법이 있으면하는 점이 그 한가지이다. 이는 사용자가 초기에 입력하는 최대 반복 계산 회수까지 반복계산이 이루어져야 하는 것을 방지하면서 과도하게 일찍 계산을 중단함으로써 덜 수렴된 계산 결과를 이용하게되는 문제도 해결할 수 있을 것이다. 그리고, 다중-블록 격자계에 대한 격자 적용은 각 블록 내부의 격자수는 유지하면서 위치만 재분포하는 셈인데, 좀더 적극적 의미에서는 전체 격자점 수는 유지하면서 블록간의 격자점 상호 교환이 가능한 격자 적용 기법이 찾아진다면 효율적인 격자 사용이 가능하게 될 것이

다. 이와 같은 문제들에 대한 적절한 해결책들이 찾아진다면, 좀더 충실한 의미의 자동화된 통합형 CFD 해석 S/W가 가능할 것으로 보여진다.

4. 결론

전산유체역학에 의한 유동 해석의 전과정을 일괄 처리할 수 있으면서도 사용이 쉽고 효율적인 S/W의 개발을 지향하면서 이루어진 본 연구의 현재까지의 성과와 구현된 시스템에 대하여 소개하였다. 본 시스템을 이용함으로써, 사용자들은 최소한의 사용자 입력만으로도 유동장 계산을 할 수 있는 예를 보였다.

그러나, 본 시스템이 아직도 많은 개선의 여지가 있는 것은 사실이고, 좀더 빠르고 정확한 유동해 프로그램, 효율적이고도 강건한 다중-블록 격자 적용 기법, 반복 계산 수렴 조건의 체계화 등과 결합되면, 좀더 실용성이 큰 시스템이 될 수 있을 것이다.

참고문헌

- [1] B.S. Kim, "Development of automatic multi-block grid generation technique adaptive to the boundary geometry of flow fields," Computational Fluid Dynamics JOURNAL, Vol. 6, No. 4, pp.527-536, Japan Society of Computational Fluid Dynamics, 1998
- [2] 김병수, "경계 형상에 적응하는 다중 블록 격자의 자동 생성 기법," 한국항공우주학회, 춘계 학술발표회, 1996
- [3] D. Anderson, "Adaptive Grid Methods for Partial Differential Equations," Advances in Grid Generation Journal, June, pp.1-15, 1983
- [4] H.A. Dwyer, R.J. Kee, and B.R. Sander, "Adaptive Grid Method for Problems in Mechanics and Heat Transfer," AIAA Journal, Vol. 18, 1980
- [5] 허지행, 김병수, "다중 유동 구배에 대한 격자 적용 기법," 제6차 유도무기(기체/발사대) 학술대회, 1996