



상호 대화형 격자 생성 환경을 이용한 항공기 전기체 격자계 생성

Grid Generation about Full Aircraft Configuration Using Interactive Grid Generator

김윤식* 권장혁†
Y.S. Kim J.H. Kwon

Abstract

An Interactive grid generation program(KGRID) with graphical user interface(GUI) has been improved. KGRID works on the UNIX environment and GUI has been implemented with OSF/Motif and X Toolkit and the graphics language is OpenGL for visualization of the 3D objects. It supports more convenient user environment to generate 2D and 3D multi-block structured grid systems. It provides various useful field grid generation methods, which are the algebraic methods, the elliptic partial differential equations method and the predictor-corrector method. It also supports 3D surface grid generation with NURBS(Non-Uniform Rational B-Spline) and various stretching functions to control grid points distribution on curves and surfaces. And some menus are added to perform flexible management for the objects. We generated surface and field grid system about full aircraft configuration using KGRID. The performance and stability of the KGRID is verified through the generation of the grid system about a complex shape.

1 서론

1980년대 이후 컴퓨터 하드웨어는 비약적인 발전을 거듭하여 왔다. 또한, 컴퓨터를 이용하는 공학의 여러 분야 역시 비슷한 변화를 겪어왔다. 전산 유체 역학 분야에서도 실험이나 숙련자의 경험을 대체하기 위한 수단으로서 컴퓨터를 이용한 해석이나 설계는 하나의 중요한 도구로 정착되었다. 이와 같이 전산 유체 역학이 고가의 장비를 이용한 실험이나 풍부한 경험을 요구하는 설계과정을 대체할 수 있는 도구로서 자리잡을 수 있었던 과정에서 빼놓을 수 없는 부분으로서 유동장 해석을 위한 수치적 기법들의 발달을 들 수 있을 것이다. 이러한 수치해석 기법과 하드웨어적인 발달은 많은 연구자들로 하여금 3차원의 복잡한 형상에 대한 유동장 해석을 현실로 가능케 하였다.

그러나, 이러한 풍요를 누리는데 있어서 가장 큰 장애물은 복잡한 형상에 대한 격자계 생성 부분이라 할 수 있다. 실제로 3차원 항공기 등과 같이 복잡한 형상에 대한 유동장 해석과정에서 가장 많은 시간과 노력을 요하는 부분은 형상과 형상주위의 해석하고자 하는 유동장에 대한 격자계 생성 과정이다. 이와 같이 격자계 생성은 전산 유체 역학에서 가장 큰 병목점 중의 하나로 인식되고 있다. 이러한 병목 현상의 해소를 위하여 선진국에서는 GRIDGEN, GENIE++, EAGLEView 등과 같이 여러 격자 생성을 위한 소프트웨어에 대한 연구와 개발을 해오고 있으며, 이러한 소프트웨어들은 연구소, 산업체 등에서 실제적인 문제에 대한 유동장 해석을 위한 중요한 도구로서 활용되어지고 있다. 그러나 국내에서는 3차원 형상에 대한 격자계 생성을 위한 격자 생성 프로그램의 개발 및 활용은 극히 미미한 실정이다.

격자 생성 프로그램이 실제적인 문제에 대한 해석 과정에서 활용되기 위해서 갖추어야 할 중요한 요소들은 크게 다음과 같은 두가지를 들 수 있다. 먼저 격자 생성 프로그램은 실제 3차원 유동장 해석을 위하여 필요한 양질의 격자계를 제공할 수 있어야 한다. 그리고, 또 한 가지는 격자계 생성에 대한 경험이 많지 않은 연구자들도 사용하기에 편리하고 친숙한 환경을 제공해 줌으로써 사용상의 편

*학생회원, 한국과학기술원 항공우주공학과

†정회원, 한국과학기술원 항공우주공학과

의성을 제공해 줄 수 있어야 한다. 이러한 두가지 기능은 격자 생성 프로그램의 활용성을 결정하는 가장 중요한 부분으로서 두가지 기능이 조화를 이룬 격자 생성 프로그램은 그 활용성을 극대화 할 수 있을것으로 생각된다. 일반적으로 유동장 해석을 위한 상용 프로그램들에서 제공하는 격자 생성 기능이 많이 활용되지 못하는 이유도 여기에서 찾을 수 있을 것이다. 상용 소프트웨어의 경우 다양한 사용자 편의성의 추구를 위하여 많은 부분에서 양질의 격자계 생성을 위한 기능보다는 편의성을 우선으로 하고 있다. 따라서 실제 유동장 해석을 위해 적합한 격자계를 생성하기 위해서는 매우 번거로운 절차를 거쳐야 하거나, 양질의 격자계를 생성하지 못하는 경우도 있다. 그리고, 일반인들이 쉽게 격자계를 생성할 수 있는 친숙한 환경을 제공하지 못하는 경우에는 양질의 격자계를 생성할 수 있는 기능은 보유하고 있더라도 일반 사용자들로 부터 외면 당하게 되기도 한다.

위에서 언급한 격자 생성을 위한 편의성면에서 필수적인 요소는 생성된 격자계에 대한 가시화 기능이다. 복잡한 형상에 대한 양질의 격자계 생성을 위해서는 격자 요소들의 생성, 확인 및 수정 등의 과정을 반복적으로 수행해야 하는 경우가 많 이 있다. 이러한 과정을 효율적으로 처리하기 위해서 생성 및 수정 결과를 매 번 확인할 수 있는 기능이 필요하다. 이에 본 연구자는 그래픽 환경을 이용한 3차원 격자 생성 프로그램을 개발, 소개한 바 있다.[1]

일반적인 3차원 형상에 대한 격자계 생성은 대체로 형상 및 해석하고자 하는 유동장에 대한 이해, 구획의 결정, 표면 격자의 생성 그리고 내부 격자의 생성의 과정을 거치게 된다. 이렇게 생성된 격자계는 유동장에 대한 수치적인 해가 만족할 만한 결과를 줄때까지 수정, 보완의 과정을 거쳐 양질의 격자계를 생성하고 이를 이용하여 여러가지 물리적 변수들을 수정하여 다양한 유동장에 대한 계산을 수행하게 된다. 격자 생성 과정에서의 난이도는 형상의 복잡성 정도, 해석하고자 하는 유동장의 물리적 성질이 복잡한 정도 그리고 사용하는 수치적 기법이 격자계에 대한 민감한 정도에 따라 결정되게 된다. 격자 생성 과정이 복잡한 경우 특히, 복잡한 3차원 형상의 경우 다양한 경험을 바탕으로 격자계를 생성함으로써 보다 효율적으로 격자계를 생성할 수 있게 된다.

본 연구에서는 앞서 소개한 격자 생성 프로그램을 기반으로하여 일반인들도 편리하게 격자계를 생성할 수 있도록 수정, 보완, 발전시킨 내용을 소개하고자 한다. 이 프로그램은 KGRID라 명명하였으며, KGRID를 이용하여 3차원 항공기 전기체에 대한 다중 격자계를 생성함으로써, 복잡한 3차원 형상에 대한 격자 생성을 효율적으로 처리하는 방법에 대한 연구를 수행하였다.

2 연구 내용 및 KGRID의 주요 기능

2.1 개발 환경

일반적으로 복잡한 3차원 형상에 대한 격자 생성 및 수치적 해석 과정에는 매우 많은 계산을 필요로 한다. 따라서, 고성능 컴퓨터를 사용해야만하는 경우가 많다. 이러한 컴퓨터 환경에서 사용되고 있는 운영체제(operating system)로써는 주로 UNIX가 사용되고 있으며, 이에 맞추어 UNIX환경에서 사용할 수 있는 격자 생성 프로그램을 개발하고자 하였다. 그리고, 생성된 격자계나 계산 결과를 가시화 하기 위해서는 3차원 그래픽을 처리할 수 있는 라이브러리(library)가 필수적이다. 이러한 기능을 가진 라이브러리들 중 뛰어난 기능을 가진것으로써 Silicon Graphics사에서 개발된 OpenGL[2][3][4] 라이브러리가 상용으로 제공되고 있으며 다양한 UNIX환경과 PC의 Windows환경에서도 사용되어 지고 있다. 그러나, OpenGL 라이브러리는 상용으로 제공되고있기 때문에 프로그램의 개발 및 사용에 불편을 겪게 된다. 이를 피하기위해 KGRID에서는 Linux와 같이 공개되어 있고 OpenGL과 모든 함수가 같으며 다양한 종류의 UNIX환경에서 사용할 수 있는 Mesa 라이브러리를 사용하여 개발하였다. 그리고, OpenGL 라이브러리는 다양한 시스템에서 사용될 수 있도록 하기위해서 창을 관리할 수 있는 기능을 없애고, 각 시스템에서 사용하고 있는 창관리 소프트웨어와 함께 결합하여 사용하도록 설계되었다. 따라서, OpenGL 라이브러리를 이용하여 프로그램을 개발할 경우 창을 관리할 수 있는 별도의 라이브러리를 필요로 한다. 그래서, KGRID에서는 친숙하고 편리한 사용자 환경을 제공하고 다양한 종류의 UNIX환경에서 사용되고 있는 X-Windows[5][6] 라이브러리와 Motif[7][8] 라이브러리를 사용함으로써 프로그램 사용자의 편의성을 극대화 하도록 하였다. 사용된 컴퓨터 언어는 메모리 관리 및 그래픽 사용자 환경의 구현을 위하여 C 언어를 사용하여 ANSI C 표준[9][10]을 따랐으며, 격자계 생성을 위한 계산 부분은 FORTRAN77을 사용하여 개발하였다. 이렇게 개발된 KGRID는 현재 Silicon Graphics의 Irix와 PC용 UNIX라 할 수있는 Linux에 이식(porting), 검증하였다.



2.2 자료 구조

격자계 생성 프로그램과 같은 전 처리기(pre-processor)의 경우 일반적인 해석자(solver)나 후 처리기(post-processor)에 비해서 자료 구조(data structure)의 결정이 전체 프로그램의 성능에 미치는 영향이 매우 크다 할 수 있다. 그 이유는 전 처리기의 경우 다양한 개체의 생성 및 삭제, 작업 과정의 취소(undo) 등의 과정을 지원할 수 있어야 하며, 이러한 기능들의 수행을 통해서 실제 필요로 하는 데이터의 크기가 항상 변할 수 있다. 이런 경우 프로그램은 전체 데이터의 크기와 각 요소들의 위치를 항상 일관적이고 안정적으로 처리할 수 있어야 하기 때문이다. 부적절한 자료구조의 선택은 프로그램의 실행 과정에서 불필요한 작업을 수행해야 하거나, 데이터에 접근(access)하는데 걸리는 시간을 늘어나게 할 수도 있으며, 이로 인해 처리시간을 필요 이상으로 사용하는 경우들도 발생할 수 있다. 그리고, 개발된 프로그램의 확장성과 안정성을 결정하는데도 매우 중요한 역할을 하게 된다.

KGRID에서 사용한 자료 구조는 기본적으로 double linked-list 자료 구조를 사용하였으며, 메모리의 관리 및 이중 언어와의 연계성에 편리한 1차원 배열을 이용하여 격자점에 대한 데이터를 저장 관리 하도록 하였다. 그림 1은 KGRID에서 사용한 자료 구조를 도식으로 나타낸 것이다. 최상위 부분은 사용자의 작업과정에 대한 정보를 기록하고 있으며, 두번째 계층에 있는 데이터는 생성된 개체의 형태와 가시화를 위한 정보 등을 저장하고 있다. 첫번째와 두번째 계층에 있는 linked-list는 좌우와 상하의 데이터에 대한 포인터를 가지고 있음으로써 새로운 개체의 추가, 삭제 등을 지원할 수 있도록 하였다. 그리고, 세번째 계층에 있는 데이터는 생성된 격자 요소의 차원과 포인터를 가지고 있으며 네번째 계층에는 실제 생성된 격자 요소의 격자점에 대한 정보를 기록하고 있다. 그리고 최하단에는 삭제되거나 수정된 데이터를 임시파일로 저장하여 수정 작업 등에 대한 취소 기능의 수행시 다시 복구될 데이터를 저장하게 된다. 최상위에서 네번째 계층까지는 메모리상에서 관리된다. 새로운 개체의 생성 과정에서는 필요한 메모리가 할당되며 불필요한 개체에 대해서는 파일에 저장후 삭제되게 된다.

2.3 주요 기능

현재 KGRID(Ver.1.0)에서 구현된 주요 기능들은 아래와 같다. KGRID는 3차원 공간상의 격자 요소들의 생성 및 수정 기능을 지원하며, 표면 격자 및 내부 격자 생성을 위해서 지원하는 주요 기능은 다음과 같다.

- 대수적(algebraic)[11] 방법을 이용한 곡면상의 표면 격자 및 3차원 내부 격자계 생성
- 타원형(elliptic) 편미분 방정식을 이용한 방법으로서 Sorenson[12]이 제안한 기법을 이용한 평면상의 내부 격자계 생성, Thomas-Middlecoff[13]가 제안한 기법을 이용한 평면 및 3차원 내부 격자계 생성
- 쌍곡선형(hyperbolic) 편미분 방정식을 이용한 방법을 수정한 예측-수정 기법[14]을 이용한 평면 및 3차원 내부 격자계 생성
- NURBS[15][16]를 이용한 곡선 및 곡면상의 표면 격자계 생성
- Vinokur[17]에 의해서 제안된 격자점 분포 함수들(stretching functions) 및 기존의 격자 요소의 분포 함수의 복사를 통한 격자점 분포 함수들

그리고, 격자점 저장 방향의 가시화 및 수정, 격자점 위치의 가시화 및 확인, 격자 요소의 ID 출력 및 수정, natural cubic spline[18]을 이용한 격자점 재배치, 3차원 곡면 가시화를 위한 은선삭제(hidden line removal), 생성 및 수정과정의 취소 및 불필요한 요소들의 삭제 등의 기능을 지원하고, 문자 입출력 방식의 격자 생성 프로그램[19]의 장점인 파일을 통한 수정 및 재생성 기능을 지원하기 위한 script file을 통한 격자 생성 기능 등을 구현하였다.

3 항공기 주위의 다중 격자계 생성

격자계 생성을 위한 항공기의 형상은 그림 2와 같다. 항공기의 형상은 상반각이 있는 주익과 수평 꼬리날개, 수직 꼬리날개와 같은 일반적인 항공기 형상을 갖추고 있다. 그림에서 나타난 Missile에 대한 격자계는 중첩격자계 구성을 위하여 항공기 형상의 격자계와는 별도로 생성되었다.

3.1 구획의 결정

주어진 형상에 대한 다구획(multi-block) 격자계 생성시 구획의 형상 및 구획 경계의 결정은 전체 격자점의 수와 생성된 격자계의 질에 매우 큰 영향을 미친다. 그리고, 구획 형상에 따라 표면 격자계 생

성부터 내부 격자계 생성과정에 이르기 까지 모든 과정에 큰 영향을 미치게 되므로 매우 신중하게 선택하여야 한다. 일반적으로 H-H형의 격자계는 복잡한 형상에 대해 쉽게 격자계를 생성할 수 있는 반면 비교적 많은 격자점을 생성하게 된다. O-O형의 격자계가 가장 적은 격자점을 생성할 수 있는 장점이 있다. 그러나 본 연구에서 사용된 항공기와 같은 형상에서는 사용하기 어려운 형태의 구획이다. 본 연구에서는 복잡한 형상에 대해서 다구획 격자계 생성이 용이한 H-H형의 격자계를 이용하였으며 구획 경계의 형태는 그림 3과 같으며, 전체 구획수는 8개의 구획으로 이루어져 있다. H-H형의 구획 경계면상의 격자점의 생성과정에서 격자점들의 분포는 대체로 마주보는 경계에서의 격자점 분포 합수를 비슷하게 사용하는 경우가 많으며 이러한 경우에 KGRID의 격자점 분포 함수 복사 기능을 이용하였다.

3.2 표면 격자계의 생성

항공기나 자동차의 표면과 같은 자유 곡면에 대한 형상 설계 및 표면 격자의 생성 과정에는 NURBS를 이용한 기법이 많이 이용되어지고 있다. NURBS를 이용한 곡선이나 곡면에 대한 표현은 CAD/CAM/CAE 시스템의 데이터 교환을 위한 표준으로 많이 사용되어지고있는 IGES(Initial Graphics Exchange Specification) 이나 STEP(Standard for the Exchange of Product Model Data) 등에서도 정의되어 있다. 아직 KGRID에서는 IGES나 STEP 데이터의 번역기는 지원하지 않고 있지 않다. 그러나 이러한 부분이 보완되면 IGES 데이터로부터 바로 표면 격자계를 생성할 수 있을 것이다. 본 연구에서는 형상을 나타내기 위한 기본적인 데이터로부터 control net과 knot vector를 결정하고 이를 이용하여 곡면상의 표면 격자계를 생성하였다. 다구획 격자계 생성에서 표면 격자계를 생성할 때 부딪치는 또 하나의 문제점은 표면상의 구획 경계에서 격자점이 서로 일치해야 하는 것이다. 그러나, NURBS를 이용하여 곡면에 격자점들을 재구성하는 경우 원하는 격자점 조절 함수가 정확히 반영되지 못하는 문제점이 있으며, 본 연구에서는 참고 문헌[16]에서 사용한 기법을 도입하여 NURBS곡면들간의 경계점 일치를 구현하였다. 그리고, 주날개와 수평 꼬리날개, 수직 꼬리날개의 표면 격자계는 CATIA 데이터로부터 추출된 이형 데이터를 cubic spline을 이용하여 격자점을 재 분포 시키고, 대수적 기법을 이용하여 생성하였다. 그림 4은 생성된 항공기의 표면 격자계를 보여주고 있다.

3.3 내부 격자계의 생성

곡률이 큰 곡면을 포함하고 있는 구획에서는 대수적 방법을 이용한 내부 격자계를 초기치로 하여 타원형 편미분 방정식을 이용하여 생성하였으며, 비교적 단순한 형상에 대해서는 대수적 방법을 이용하여 생성한 내부 격자계를 그대로 사용하였다. 전체 격자계는 약 70만개의 격자점으로 구성되어 있다. 그림 5은 항공기 주위에 생성된 내부 격자계를 나타낸다. 그림 우측 하단의 대화상자는 3차원 내부 격자계의 가시화기능을 조절하기 위한 조절판을 나타내고 있다.

4 결론 및 추후 연구 방향

4.1 결론

그래픽 환경을 이용한 상호 대화 방식의 3차원 격자 생성 프로그램인 KGRID를 이용하여 항공기 전 기체 주위의 다중 격자계를 생성하였으며, 이러한 과정에서 다음과 같은 결론을 얻었다.

- 가시화 기법과 격자 생성 기능의 통합을 통한 보다 효율적인 격자 생성 환경을 구축하고, 일반 인들도 복잡한 3차원 형상에 대한 격자계를 구성할 수 있는 소프트웨어를 개발하였다.
- KGRID는 복잡한 3차원 형상에 대한 표면 격자 및 내부 격자의 생성을 위해서 활용될 수 있는 다양하고 편리한 기능들을 안정적으로 제공한다.

4.2 추후 연구 과제

KGRID를 복잡한 형상에 대한 격자계 생성과정에 활용과정에서 파악된 추후 보완되어야 할 주요 부분들은 다음과 같다.

- IGES, STEP 등과 표준 데이터 형식의 형상 데이터로부터 표면 격자계 생성을 위한 변환기의 개발
- 타원형 편미분 방정식을 이용한 다구획 격자계 생성시 다양한 구획 경계면 처리 기능

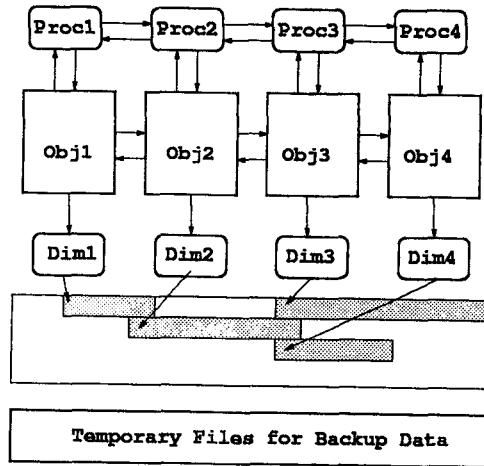


Figure 1: 자료 구조

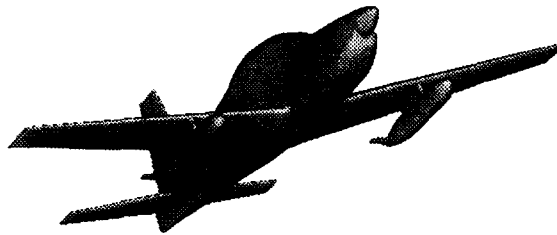


Figure 2: 항공기의 형상

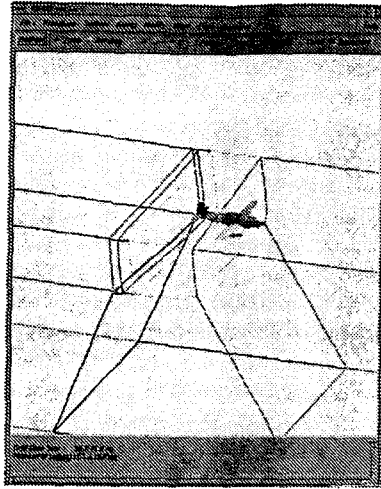


Figure 3: 항공기 주위의 다구획 설정



Figure 4: NURBS를 이용한 표면 격자계 생성

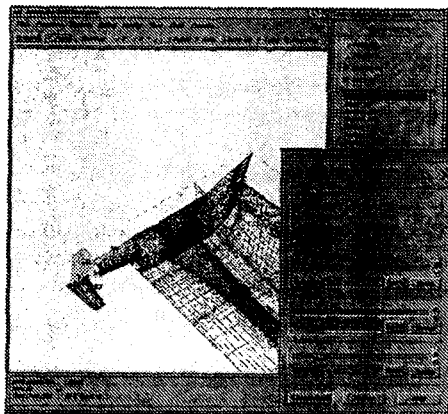


Figure 5: 내부 격자계의 생성 및 가시화



References

- [1] 김 윤식, 권 장혁, "그래픽 환경을 이용한 상호 대화 방식의 3차원 격자 생성 프로그램 개발", 한국항공우주공학회 춘계 학술발표회 논문집, 1998, pp67-70
- [2] Jackie Neider, Tom Davis and Mason Woo, 1993, *OpenGL Programming Guide: the official guide to learning OpenGL, Release 1*, Addison-Wesley.
- [3] Mark J. Kilgard, 1996, *OpenGL programming for the X Window System*, Addison-Wesley Developers Press.
- [4] Document File, *OpenGL Reference Manual*, Silicon Graphics
- [5] Adrian Nye, 1992, *Xlib Reference Manual*, 2nd ed., O'Reilly & Associates, Inc.
- [6] David Flanagan, 1990, *X Toolkit Intrinsics Reference Manual*, 3rd ed., O'Reilly & Associates, Inc.
- [7] Douglas A. Young, 1994, *The X Window System Programming and Applications with Xt*, 2nd ed., Prentice Hall
- [8] *OSF/Motif Programmer's Reference*, Open Software Foundation
- [9] Mark Williams Company, 1988, *ANSI C : A Lexical Guide*, Prentice Hall
- [10] W. Richard Stevens, 1992, *Advanced Programming in the UNIX Environment*, Addison Wesley
- [11] Joe F. Thompson, Z.U.A. Warsi and C. Wayne Mastin, 1985, *Numerical Grid Generation : Foundations and Applications*, North-Holland.
- [12] Steger, J.L. and Sorenson, R.L., "Automatic Mesh-Point Clustering Near a Boundary in Grid Generation with Elliptic Partial Differential Equations", *J. Comp. Phys.*, vol.33, no.3, Dec. 1979, pp405-410.
- [13] P.D. Thomas, J.F. Middlecoff, "Direct Control of the Grid Point Distribution in Meshes Generated by Elliptic Equations", *J. AIAA*, Vol.18, No.6, 1979, pp652-656
- [14] 김 병수, "Predictor-Corrector를 활용한 외부 유동장 격자 생성 기법", 한국전산유체공학회 제2권 제1호, 1997, 4월, pp84-92
- [15] Les Piegl and Wayne Tiller, 1995, *The NURBS Book*, Springer-Verlag.
- [16] TzuYi Yu and Bharat K Soni, "Application of NURBS in numerical grid generation", *CAD*, Vol.27, No.2, 1995, pp147-157
- [17] Vinokur, Marcel, "On One-Dimensional Stretching Functions for Finite-Difference Calculations", *J. Comp. Phys.*, Vol.50, 1983, pp215-234
- [18] David F. Rogers, 1990, *Mathematical Elements for Computer Graphics*, 2nd ed., McGraw-Hill
- [19] Yong-Hyun Yoon, "Enhancements and Extensions of EAGLE Grid Generation System", 1991, Ph.D Thesis, Mississippi State University