

대용량 해석을 지원하는 무기체계 연구 개발 전용 모델러 설계 기술 연구

송일환*, 김익준*, 리경호*, 유용균*, 한순홍**

Study on the Design of CAE Modeler for a Weapon System Supporting a Large Volume CAE Analysis

Ilhwan Song*, Ijkune Kim*, Jinggao Li*, Yonggyun Yu* and Soonhung Han**

ABSTRACT

Generation of over one hundred million mesh is essential for getting an exact analysis result of penetration, combustion, and explosion of missile. But because no domestic modeler to support this exists and a modeler only for missile also has not been developed yet, it is too difficult to get this goal. In this research we develop a modeler only for an engineering analysis of missile using 64bit computing system to solve current problems.

Key words : CAE, Mesh generation, modeler, Data management

1. 서 론

해석프로그램을 이용하여 문제를 해결할 때 가끔 예상치 못한 문제에 부딪히는 경우가 발생한다. 특히 mesh의 개수의 증가함에 따라 데이터 파일의 크기와 사용하는 메모리의 공간, 해석시간이 기하급수적으로 증가하는 문제는 해석을 어렵게 만드는 큰 요인이 되고 있다. Mesh 개수증가에 따른 이들 요소의 증가는 불가항력적인 사항이므로 이들을 좀 더 효과적으로 관리함으로써 이들을 최소화할 수 있는 것이 좋은 modeler의 역할이라 할 수 있다. 본 연구에서는 이를 위해 다음과 같은 네 가지 사항에 대한 연구를 수행하였다.

첫째는 대용량 데이터 관리에 대한 연구이다, 미국의 UIUC(University of Illinois at Urbana-Champaign)에서는 로켓관련 해석을 위해 대용량 데이터들을 다루는 방법에 대한 많은 연구를 하였고, 이에 따른 결과물로 Rocketeer란 프로그램을 개발하였다.

둘째는 Mesh 생성 기술이다. Mesh를 생성시키는 기술은 Solver에 입력하기 위한 입력파일의 크기와 해석시간뿐 아니라 해석결과와 질(Quality)과도 관계가 깊다. 미국에서 개발된 ANSA라는 pre & post-processor를 통해 mesh 생성기술과 보핑 기술에 대해 살펴보도록 한다.

셋째는 데이터와 해석시스템간의 인터페이스를 효율적으로 해주는 기능이다. 상용프로그램인 ABAQUS/CAE에서는 이를 위해 'include' 기능과 스크립트 기능을 지원한다. 'include' 기능은 데이터의 효과적인 입력을 위해 입력 파일을 나누는 기술이고, 스크립트 기능은 GUI를 통한 입력이 아닌 텍스트 입력을 통해 실행을 시키는 것이다.

본 연구의 목적은 mesh의 개수가 1억 개까지 생성이 가능한 modeler를 개발하는 것이다. 이를 위해 하드웨어의 지원도 중요하지만, 이를 뒷받침할 수 있는 소프트웨어도 상당히 중요하다. 현재 우리나라의 경우 천만 단위의 mesh의 개수로 연구가 이루어지고 있다. 외국의 경우, 미국, 유럽에서 개발되는 상용프로그램들은 억 단위로 개발이 이루어지고 있으며, 일본의 경우 2003년에 세계 5위 수준의 슈퍼컴퓨터를 이용하여 55억 개의 mesh를 생성한 전(주)지구모넬을 완성하기도 하였다(Fig. 1).

*학생회원, 한국과학기술원
**중신회원, 한국과학기술원
- 논문투고일: 2009. 03. 04
- 논문수정일: 2009. 11. 29
- 심사완료일: 2009. 12. 01

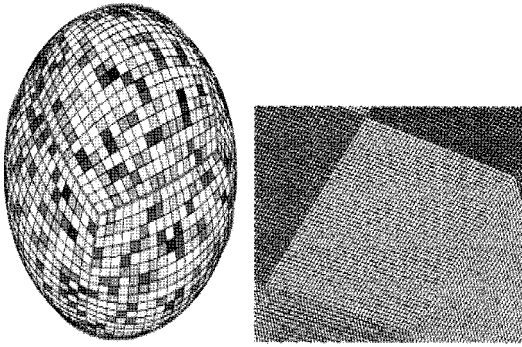


Fig. 1. 전 지구 모델.

2. 관련 연구

2.1 Rocketeer

Rocketeer는 미국 UIUC에 위치한 CSAR(Center for Simulation of Advanced Rockets)을 통해서 만들어진 로켓해석용 Post-processor이다. 이 프로그램은 미국 정부(NSF)에서의 자금지원과 슈퍼컴퓨터센터인 NCSA의 도움을 받아 만들어졌다. 또한 로켓해석용 프로그램이기 때문에 NASA와 Boeing도 참여하게 되었다.

로켓해석 특성상 대용량의 자료들을 다루어야 하기 때문에 데이터관리가 이 프로그램에 있어서 매우 큰 문제였다. 그리하여 NASA는 Boeing과의 연구를 통해 CGNS라는 데이터 포맷을 개발하였으며, 또 미 에너지 국(Department of Energy)과 함께 HDF라는 데이터 포맷도 개발하였다. 이들 데이터 포맷의 특징은 대용량을 다루기 쉽다는 것과 다양한 종류의 데이터들을 포함할 수 있다는 것이다. 특히 해석용 데이터를 저장하기에도 좋을 뿐 아니라, 사진이나 그림 등의 다양한 데이터들을 저장하기 좋다. Fig. 2는 UIUC와

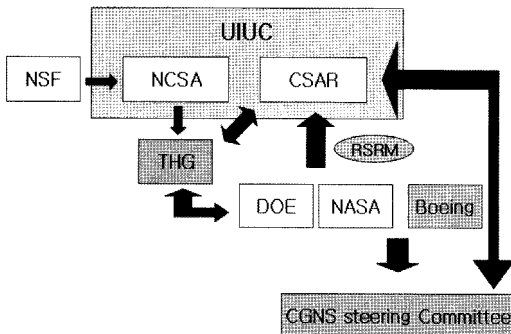


Fig. 2. Rocketeer 관련 조직도.

NASA 등의 Rocketeer와 관련된 조직들의 조직도를 보여준다.

Rocketeer는 현재 세가지 버전이 개발되었다. 첫 번째는 소프트웨어인 Rocketeer이다. Visualization toolkit과 C++를 이용하여 만들어 졌으며, Linux와 윈도우에서 모두 사용이 가능하다. 그 다음은 batch 버전인 Rocketeer/voyager로써 병렬처리 기법으로 해석 및 후처리를 지원한다. 마지막 버전은 server/client 개념의 Rocketeer Apollo/Houston이다. Fig. 3은 Rocketeer의 개발 단계 및 개발 환경, OS 등을 도식화 한 것이다.

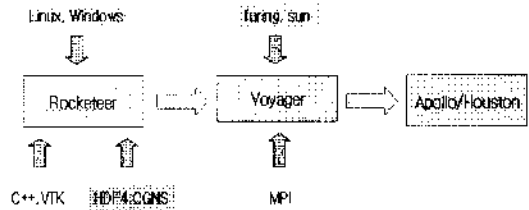


Fig. 3. Rocketeer 발전도.

2.2 Abaqus/CAE

ABAQUS는 ABAQUS.Inc에서 개발한 유한 요소 해석 프로그램으로 선형, 비선형, 충돌 및 다물체 동역학 문제들에 대해 솔루션을 제공하고 있다.

ABAQUS/CAE는 ABAQUS의 Pre와 Post-processor를 지원하는 모듈이다. 사용자가 빠른 시간 내에 2차원 및 3차원 유한 요소 모델링이 가능하도록 다양한 기능을 제공하고 있다. 특히 ABAQUS에서는 대용량의 데이터를 관리하기 위해 INCLUDE 기능을 제공하고 있다. 또한 Python을 이용한 스크립트 기능을 제공하여 사용자의 편의에 맞게 GUI 및 일부 기능 등을 수정할 수도 있다.

3. 개발 환경

3.1 64비트 컴퓨팅

본 연구의 목표인 1억 개 이상의 요소를 생성하고 관리하기 하려 할 때, 발생하는 가장 큰 문제점은 메모리 문제이다. 현재 32비트의 CPU와 OS환경에서는 메모리의 주소를 표현하는 방식에 한계가 있어, 물리적으로 메모리를 추가한다고 하더라도 약 2G이상의 메모리를 관리할 수가 없다. 본 과제의 목표인 1억 개의 Mesh를 생성한다면 순수하게 노드와 요소만의 데이터를 고려하여도 그 한계를 넘어버리기 때문에 아래와 같은 해결점을 제시하였다.

- 메모리 대신 데이터베이스 파일을 통한 데이터 관리
- 소프트웨어 자체적으로 하드디스크 스왑의 구현
- 64비트 환경에서의 소프트웨어 개발

앞의 두 방법은 하드디스크를 이용하기 때문에 속도적인 측면에서 많은 문제가 발생할 수 있을 것이라 예상되어, 많은 주목을 받는 64비트 환경에서 소프트웨어를 개발하도록 설정하였다. Fig. 4에서 볼 수 있듯이 램의 Transfer Rate가 월등히 좋은 것을 알 수 있다. 평균적으로 쓰기 모드에서 하드디스크의 경우 약 33Mb/s를 보인 반면 램의 경우 약 400Mb/s로 13배 정도의 성능 향상이 기대된다. 32비트 환경에서 Mesh를 생성했을 때 이의 용량은 4.6GB로 최소 1.6GB는 하드디스크에서 작업을 하여야 한다. 현재 시스템의 램 용량은 7GB로(8GB인데 시스템 영역이 1GB차지) 1억 1천만 개의 Mesh생성에 적합하다.

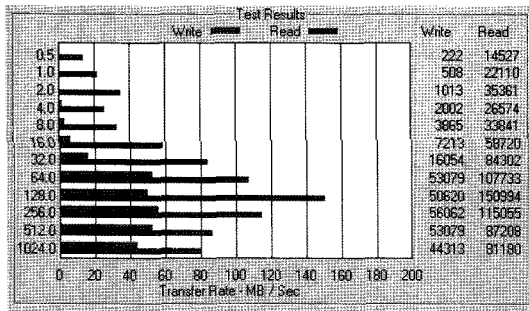


Fig. 4. 64비트 컴퓨팅 성능 테스트.

3.2 모델 가시화 환경

3.2.1 모델링 커널 (ACIS)

ACIS는 객체 지향 3-D 형상 모델링 엔진으로써 Spatial 사에서 제작되었다. 3D 모델링의 가상의 엔드 유저와 함께 형상 기반을 사용하도록 디자인되었다. c++ 기반으로 하여, 일반적인 자료 구조로부터 wireframe, surface, solid 모델링 등의 아키텍처를 제공한다.

3.2.2 HOOPS

HOOPS 3D 그래픽 시스템(HOOPS/3dGS)은 Windows나 UNIX 또는 인터넷 기반의 application을 제작하는 제작자들을 위한 고성능의 3D 그래픽 Toolkit이다. HOOPS/3dGS의 최적화된 자료 구조와 알고리즘들은 2D나 3D, CAD/CAM/CAE, GIS (Geographical Information System) 등의 개발 등을

단순화 시켜준다.

4. 메쉬 생성 기술

4.1 삼각형/사면체 자동요소 생성 프로그램의 알고리즘

삼각형과 사각형의 요소를 자동으로 생성하는데 사용하는 가장 유명한 이론은 Delaunay criterion이다. Delaunay criterion은 때때로 “empty sphere”라고도 불린다. 간단하게 설명하면 하나의 삼각형의 외접원 안에 그 삼각형의 절점 이외에 다른 삼각형의 절점이 들어오면 Delaunay criterion을 만족하지 못하는 것이다.

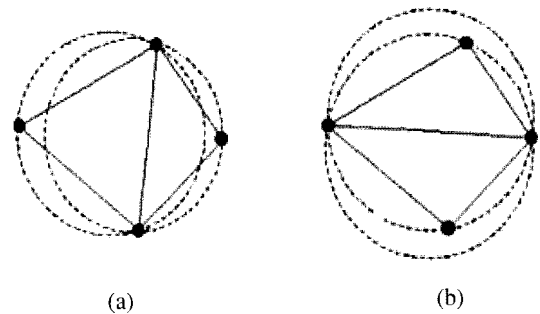


Fig. 5. 2차원의 삼각형.

Fig. 5는 2차원의 삼각형일 경우를 나타낸다. (a)는 Delaunay criterion에 만족하는데 하나의 삼각형의 외접원 안에 다른 삼각형의 절점이 위치하지 않는다. 그러나 (b)의 경우는 하나의 삼각형의 외접원 내부에 다른 삼각형의 절점이 위치한다. 그러므로 (b)는 Delaunay criterion을 만족하지 못한다.

Delaunay criterion은 그 자체로 요소들 생성하는 알고리즘으로 사용할 수 없다. 이 이론은 이미 존재하는 점들을 이용하여 점들의 위치를 바꾸면서 요소의 질을 높이는 것이기 때문이다. 그래서 요소를 생성하기 위해서는 처음에 절점들을 생성하는 것이 필요하다. 가장 일반적인 초기의 절점들을 생성하는 방법은 물체의 표면의 절점들을 그대로 사용하는 것이다. 초기의 절점들을 이용하여 삼각형 요소들을 생성하고, 그 후 Delaunay criterion 이론을 사용하여 중간에 절점들을 추가 하면서 요소들을 생성하는 것이다.

4.2 사면체 요소 자동생성 프로그램

현재 공개된 프로그램들 중에서 사면체 요소를 자동 생성하는 프로그램들을 표로 정리하였다.

Table 1. 사면체 요소 프로그램

Program	Platform	Element	Algorithm
Tctgen	Windows	Tetrahedra	Delaunay
Grummp	Unix	Triangle, Tetrahedra	Delaunay
Geopack90	Windows	Triangle, Tetrahedra, Quadrilateral, Hexahedra, Surface	Delaunay
Qhull	Windows, Unix	Triangle, Tetrahedra	Delaunay
Mesh-maker	Unix	Tetrahedra	Delaunay

Tetgen의 경우는 C언어로 작성된 프로그램으로서 Windows기반으로 개발된 프로그램이다. Grummp의 경우는 Unix기반으로 개발한 프로그램으로서 Windows기반으로 소스코드를 수정하여 테스트를 해 보았으나, 계산 시간이 오래 걸리고 프로그램 자체도 불안정 했다. 본 연구의 목적은 64 bit Windows 기반으로 개인용 컴퓨터에서 1억 개 이상의 요소를 생성하는 것이기 때문에, 요소를 생성하는 효율도 좋고, 안정적으로 실행되는 Tetgen을 본 연구에서 사용할 사면체 요소를 생성하는 프로그램으로 선정하였다.

4.3 Tetgen

Tetgen은 앞에서 설명한 바와 같은 Dealunay triangulation 알고리즘을 이용한 사면체 요소 생성 프로그램이다. 이 프로그램은 Weierstrass Institute for Applied Analysis and Stochastics의 Research Group of Numerical Mathematics and Scientific Computing에 소속된 Hang-si에 의해 제작되었다. 프로그램 언어는 C 언어를 사용하였다. Unix와 Windows기반으로 모두 사용할 수 있다.

4.3.1 프로그램 사용방법과 기능

Tetgen은 기본적으로 PLC(piecewise linear complex)라는 .poly파일을 입력으로 하여 사용한다. 예를 들면 example.poly이라는 이름을 가진 파일을 입력으로 하면, 파일 내부에 들어있는 요소를 나누고자 하는 모델의 정보로 받아 들여 요소를 나누게 된다. Fig. 6은 Tetgen의 실행 예제를 보여준다.

4.3.2 Tetgen의 64비트 포팅 및 1억 개 단위의 요소생성 앞서 비교하였던 여러 가지 공개된 요소 생성 코드들 중 가장 우수한 성능을 보이는 Tetgen을 앞의 장

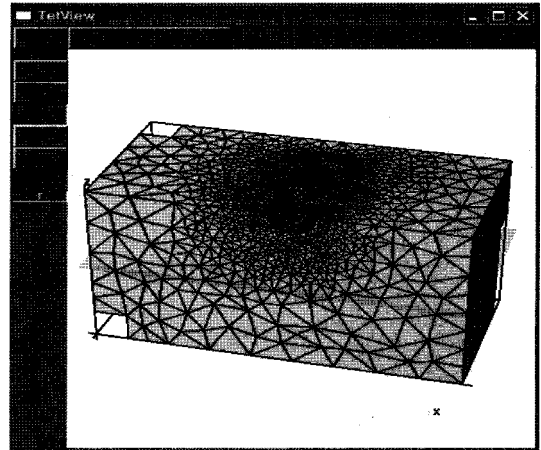


Fig. 6. Tetgen 실행 화면.

의 내용을 바탕으로 64비트 환경으로 수정하였다. 64비트 환경으로 새롭게 컴파일 하고자 할 때 발생하는 가장 큰 문제는 메모리의 주소를 표현하는 포인터의 자료 형이 변화함에 따라 코드 내에 메모리 주소에 관련된 자료 형은 64형으로 수정해 주어야 한다.

기존의 코드는 32비트 환경에서 약 2000만개까지의 Mesh를 생성하는 성공하였다. 64비트 환경에서는 약 4000만개의 요소 생성하는 것에 성공하였지만, 메모리의 한계로 더 이상의 Mesh가 생성되지 않았다. 코드를 64비트에 맞게 수정하면서 점차적으로 요소 숫자를 늘리는 과정을 반복하여 과제의 목표인 1억 개의 Mesh를 생성하는 것에 성공하였다. 총 수행시간은 약 48시간(173064초)이고 생성된 요소 수는 104,871,844개이다.

5. 모델 가시화

5.1 TransCAD

모델러에 있어 가장 기본적이고도 핵심이 되는 부분이 바로 모델을 생성하는 부분이다. 이 부분은 모델 생성하여 보여주는 기능 뿐 아니라, Mesh 생성 등의 작업을 위해 생성된 모델의 폴리곤 값 등의 각종 정보들을 포함하고 있어야 하며, 잘게 잘라진 Mesh의 가시화도 가능해야 한다. 또한 Solver에서의 해석 이후 생성되는 output 파일을 가시화 할 수 있어야 한다. TransCAD는 이러한 기능들을 수행하기 위해 개발된 것으로써 이번 과제의 핵심 프로그램 역할을 수행하게 된다.

TransCAD는 ACIS나 Hoops 등의 외부와의 interface 뿐 아니라, 내부에서도 많은 함수들로 이루어져 있

며, Fig. 7과 같은 UML 다이어그램으로 나타낼 수 있다.

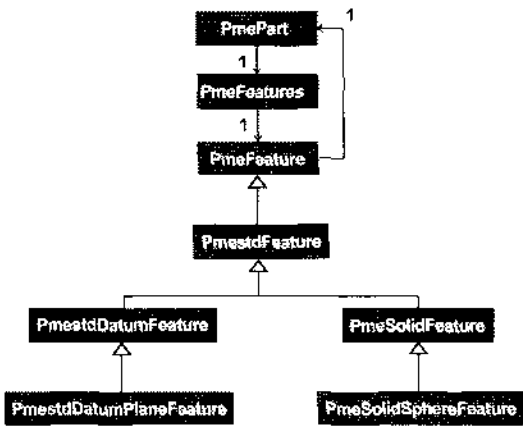


Fig. 7. TransCAD의 UML 다이어그램.

GUI는 사용자가 직접적으로 모델 생성을 하기 위한 기능을 구현해 놓은 부분으로써 프로그램에 있어 사용자 편의 측면에서 상당히 중요한 부분을 차지하고 있으며, 많은 개발자들이 많은 시간과 노력을 투자하는 부분이기도 하다.

본 연구에서는 기본적으로 CAE modeler의 MenuTree를 모델로 구현을 하고 있으며 GUI 작동 방식이나 호출 방식은 최적의 CAD modeling을 가능하도록 직접 설계 하였다. Fig. 8은 이러한 분석을 통해 구현된 최종 버전의 TransCAD의 실행장면이다.

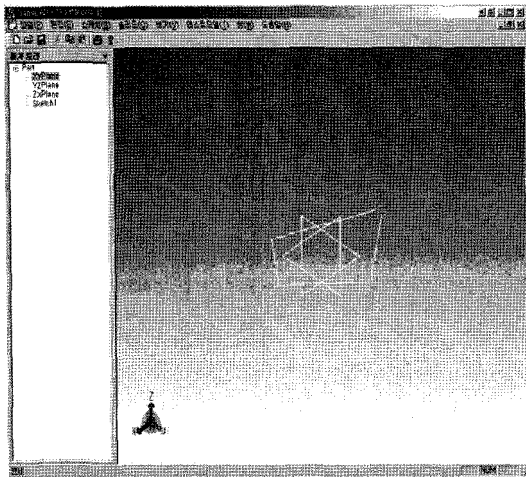


Fig. 8. TransCAD의 실행도.

5.2 시험 모델링

본 연구를 통해 완성된 모델러의 완성도 테스트를 위해서는 테스트 모델의 선정이 필수적이다. 과제의 목적과 난이도, 완성도등을 고려한 여러 선별 과정을 거쳐, 본 연구에서는 로켓의 발사체 모델을 최종 테스트 모델로 결정하였다. 우선적으로 ADD측으로부터 로켓의 발사체 2D 도면을 입수하였다.

본 연구에서 수행되는 작업은 3D이고, 위의 입수한 데이터들은 2D 자료들이기 때문에 3D로 다시 구현해야 하는 작업을 시켜야 했다. 단순히 도면의 스캔 파일만을 입수하였기 때문에 3D CAD 프로그램을 이용해 완성된 모델을 얻어내는 작업이 필요했다. 이는 모델링 전문가의 도움을 받아 CATIA 프로그램을 이용해 3D 모델을 구현하였다. 또한 이러한 작업을 통해서 모델 구현에 필요한 원수들과 각 부분의 사이즈들을 알아내었다. Fig. 9는 3D 데이터를 얻은 작업을 캡처한 것이다.

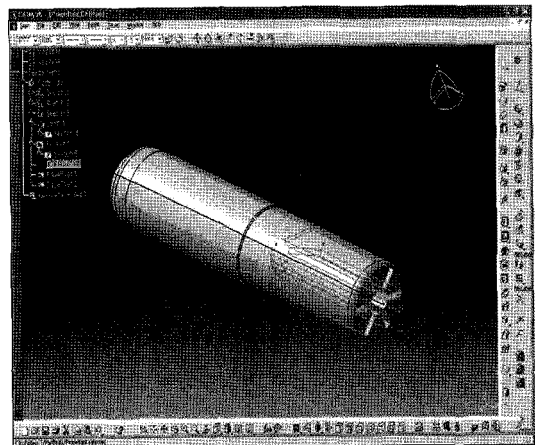


Fig. 9. 미사일 발사체 모델(CATIA).

6. 구 현

6.1 발사체 모델의 생성

앞서 다루었던 일련의 과정들을 통해 얻은 미사일 발사체 모델의 propeller 부분을, 본 연구를 위해 개발한 TransCAD상에서 구현해 보았다(Fig. 10).

보는 바와 같이 원 모양 자체는 CATIA 모델과 크게 다르지 않음을 확인할 수 있었다. 향후에는 propeller 외에 insulation 등의 남은 발사체 부분도 모델링하여 assembly 과정을 거쳐 최종 완성 모델을 구현해 볼 계획을 가지고 있다.

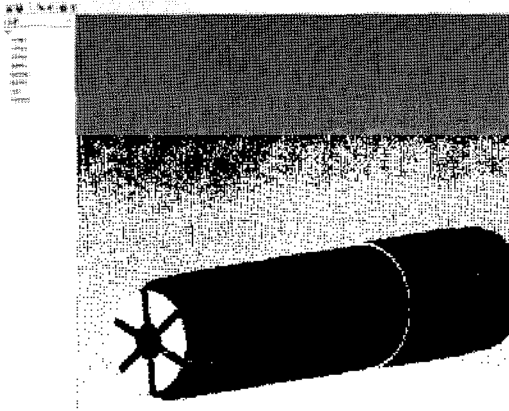


Fig. 10. 미사일 발사체 모델(TransCAD).

6.2 Mesh 가시화(Mesh Visualization)

Mesh를 가시화하는 작업은 Mesh를 잘 자르는 것만큼 중요한 작업이고, 그 만큼 시간이 오래 걸리는 작업이기도 하다. 앞의 Mesh 알고리즘들을 통해 얻어낸 Mesh 데이터 들을 효과적으로 보여주기 위한 작업이 이번 연구에서 수행되었다.

앞에서 언급되었듯이, Mesh 데이터는 크게 node, element, face 정보로 구성되어 있다. node 정보는 mesh를 자르기 위해 쓰일 점들의 좌표 값들의 정보이고, 이 점들의 연결 관계를 정의하여 Mesh를 생성할 수 있게 만들어 주는 정보가 element 정보이다. 실제 solver에서의 해석에서는 이 두 가지 정보가 사용된다. 가시화에는 모델 내부의 정보가 꼭 필요하지 않기 때문에, 본 가시화 작업에서는 node 정보와 모델의 결면, 즉 face 정보들 이용해 가시화를 하도록 하였다. 이를 통해, 가시화에 필요한 시간을 줄이면서 동시에 가시화에 이용되는 메모리의 크기를 크게 줄일 수 있었다.

가시화에서의 중요한 요소가 Mesh 데이터를 읽어들이고, 화면상에 가시화하는 데까지 걸리는 시간이다. 물론 CPU나 메모리 등의 하드웨어 적인 요소도 중요하지만, 어떤 알고리즘을 이용하여 가시화하였는지도 중요하다. 본 테스트에서 이용할 모듈에서는 여러 번의 실험의 거쳐 가장 빠른 속도를 보인 알고리즘을 채택하였다.

각 경우에 따라서 정확한 속도를 측정하기 위해 시간을 재는 프로그램을 따로 작성하여 측정하였으며, 소수점 3자리까지 측정하였다. 아래의 표는 각 경우에 따른 시간을 측정한 결과로써, 가시화에 걸리는 시간이 거의 Mesh 개수에 비례하는 것을 볼 수

가 있었다. 또 하나의 중요한 요소인 최대 사용 메모리 또한 그 크기를 측정하려 했지만, 정확한 측정치를 얻는 데에 어려움이 있어서 측정을 하지 못하였지만, 대략적인 크기역시도 Mesh 개수에 비례함을 볼 수 있었다.

Table 2. Mesh 개수에 따른 생성 시간

Mesh 개수 (개)	시간 (sec)
10만	0.453
100만	4.015
1000만	36.437
5000만	173
1억	335

위의 기록을 보면, 1억 개의 가시화에만 걸리는 시간이 335초 약 5분 30초가 걸림을 알 수 있다. 현재 32비트 상에서의 상용프로그램에서도 수백만 개의 가시화를 위해 몇 분의 시간이 걸리지만, 사용자 측면에서 생각해 보았을 때, 이는 매우 긴 시간이며, 또한 가시화를 위해 컴퓨터 자체의 속도가 현저히 떨어지는 것을 알 수 있었다.

Fig. 11은 미사일 발사체 배쉬 데이터를 TransCAD에 연동시켜 가시화한 그림이다.

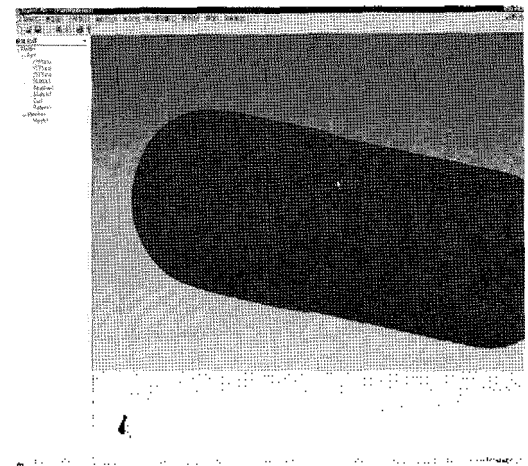


Fig. 11. 발사체 모델의 메쉬 생성 (1억 개).

7. 결론 및 향후 과제

애초의 연구 목표였던 64비트 환경을 이용한 1억개의 메쉬 생성과 가시화는 성공적으로 수행되었다. 하지만, 여기에서 멈추지 않고, 본 연구의 성과가 더 완

벽한 전처리기로 거듭나기 위해서는 몇가지 수정보완하여야 할 점이 있다.

첫 째는 GUI의 완성이다. 현재는 시험모델이었던 발사체 모델만을 생성하는데 필요한 함수들만이 구현되어 있어, 완벽한 모습을 갖추지 못한 상태이다. 좀 더 다양한 함수들을 추가함으로써 완벽한 GUI를 구축할 예정이다.

둘 째는 해석을 위한 각종 정보 입력을 위한 모듈을 추가하는 것이다. 현재의 구현된 기능들은 모델 생성과 Mesh 정보를 얻기 위한 기능 두 가지 뿐이다. 하지만 완벽한 형태의 input file에는 이들 정보뿐 아니라, 각종 material 정보와 Boundary condition, Load 정보 등이 필요하다. 또한 이를 위한 GUI도 따로 구축할 예정이다.

감사의 글

이 논문은 국방과학연구소의 연구비 지원에 의하여 연구되었음.

참고문헌

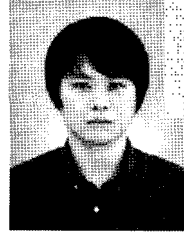
1. CSAR homesite, <http://www.csar.uiuc.edu>
2. Rocketeer, http://www.csar.uiuc.edu/F_software/rocketeer/
3. Said, R. and Weatherill, N. P., "Distributed Parallel Delaunay Mesh Generation", *Comput. Method Appl. Engrg.*, Vol. 177, pp. 109-125, 1999.
4. Qiang Du and Desheng Wang, "Recent Progress in Robust and Quality Delaunay Mesh Generation". *Journal of Computational and Applied Mathematics*, Vol. 195, pp. 8-23, 2006.
5. Robert A. Fiedler and John C. Norris, "Massively Parallel Visualization on Linux Clusters with Rocketeer Voyager", NCSA conference, 2001.
6. Robert H. Bush, "CFD General Notation System: Status and Future Direction", AIAA, Jan. 2001.
7. NCSA HDF version 5, <http://hdf.ncsa.uiuc.edu/HDF5>
8. ANSA Homesite, <http://www.beta-cae.gr/ansa.htm>
9. Dimitri Komatitsch *et al.*, "A 14.6 Billion Degrees of Freedom, 5 Teraflops, 2.5 Terabyte Earthquake Simulation on the Earth Simulator", Proc. of the ACM/IEEE, 2003.
10. P. L. George, "Automatic Mesh Generation", John Wiley & Sons, 1991.



송 일 환

2004년 한국과학기술원 기계공학과 학사
2007년 한국과학기술원 기계공학과 석사
2007년 3월~현재 한국과학기술원 기계공학과 박사과정

관심분야: CAE, Exchange of CAD data, Product Data Management (PDM), ISO 10303 STEP



김 익 준

2007년 한양대학교 기계공학부 학사
2009년 한국과학기술원 기계공학과 석사
2009년~현재 한국과학기술원 기계공학과 박사과정

관심분야: STEP, Ship-CAD, Exchange of CAD data



리 경 호

1999년 9월~2003년 7월 CUST. 학사
2003년 9월~2006년 7월 Jilin Univ., 석사
2007년 3월~현재, KAIST, 박사과정
관심분야: CAD, CAGD, STEP, CAD Model Data Exchange



유 용 균

2001년 한국과학기술원 기계공학과 학사
2003년 한국과학기술원 기계공학과 석사
2003년 3월~현재 한국과학기술원 기계공학과 박사과정

관심분야: CAE, Data format, Optimal Design, Post Process



한 순 홍

한국과학기술원 기계공학과 교수이며, 웹서널인 International Journal of CAD/CAM(www.ijcc.org)의 편집장으로 활동하고 있다. 2003년까지 STEP센터(www.kstep.or.kr)의 회장과 전자기래학회(www.calsec.or.kr)의 회장을 맡았으며, 관심분야는 STEP, 가상현실 응용, 지능형 CAD이다. 연락처는 shhan@kaist.ac.kr, 홈페이지 <http://icad.kaist.ac.kr>, 미국 미시간 대학에서 1990년 박사학위