韓國海洋工學會誌 第12卷 第3號, pp.86-95, 1998. 8

◉ 論 文

선체중앙부해석 지원시스템 개발에 관한 연구

신현경* · 손호철** · 권명준* · 송재영*** · 김종현*** · 이정렬*** · 강호승*** · 연규진***

(98년 3월 16일 접수)

A Study on the Support System for Midship Structural Analysis

Hyun-Kyoung Shin* · Ho-Cheol Son** · Myung-Joon Kwon* · Jae-Young Song*** · Jong-Hyun Kim*** · Jeong-Ryul Lee*** · Ho-Seung Kang*** · Kyu-Jin Yeon***

Key Words : Midship(선체중앙부), GUI(그래픽사용자 인터페이스), Top-down(하향식), Parametric design(파라메트릭설계), Object-oriented(객체지향), Data Structure(자료구조), Finite Element Creation(유한요소생성), MFC(마이크로소프트 파운데이션 클래스)

Abstract

CAD/CAM program developers have made substantial progress in enabling engineers to design, analyze, visualize, and simulate the performance of structures and systems. Many researchers, however, agree that there is still a way to go before the existing commercial computer software can be used in ship structural analysis to their full potential as solvers which can identify opportunities to reduce man-hours and cut costs. CAD systems used for these works in ship yards are general-purposed and not custom-tailored for ship structures, which force end-users to do tiresome and routine works. The new preprocessor "MeshGen" announced in this study can load several ship types and user-friendly generate their FE meshes employing the object-oriented approach based on the parametric and top-down design. Also "MeshGen" reduces dramatically the time required to prepare CAD models under window-based environments on desktop personal computers and makes it much easier and faster for end-users to change topology and material properties.

1. 서 언

산업기술은 Computer의 발전에 힘입어 눈부신 성장을 하였다. 특히 CAD/CAM분야는 기계, 건축, 토목, 화학공업 등 모든 공학분야의 설계, 구조해석, 생산자동화 등에 필수적이라고 할만큼 많은 진전이 있었다[1~8]. 이러한 CAD/CAM분야의 적극적인 활용은 조선현장 전반에 걸쳐 불필요한 공수의 절감, 생산성과 품질의 향상, 비용의 절감 등을 현실화시키고 있는 것이 사실이다. 그러나 위와

* 울산대학교 수송시스템공학부
** 현(사)한국선급 기술연구소, 연구 당시 울산대학교 수송시스템공학부
*** (사)한국선급 기술연구소

같은 사실에도 불구하고 선체구조해석의 경우 구조해석 모델링과정은 많은 공수의 낭비를 가져오고 있다. 일반적으로 초기 구조 설계 도면을 바탕으로 고가의 CAD시스템을 이용하여 모델링작업을 한다. 이때 모델링 및 해석에 소요되어지는 시간은 보통 1~3개월이라는 긴 시간이 필요하며 이 기간의 대부분이 모델링에 쓰여지고 있다. 이것은 작업시 사용되는 CAD시스템이 선체구조해석 모델링 전용이 아닌 범용의 CAD시스템이기 때문이다. 이러한 기존의 CAD시스템은 사용자들로 하여금 단순하고도 지루한 과정의 반복을 요구하고 있다. 반면 지금까지 개발된 선박의 중앙단면부의 구조해석요소 자동생성 프로그램들은 너무나 경직되어 있어 사용자의 경험과 공학적 감각 등이 거의 반영되지 못해 조선 현장에서 환영받지 못하고 있는 실정이다. 기존의 자동 유한요소 알고리즘은 요소생성이 너무 자유로우며, 선박의 특성 및 구조해석상 요소생성을 구속하는 조건이 많아 적용하기 어렵고, 상용소프트웨어의 유한요소생성 모듈을 사용하여 사용자가 쉽게 적용할 수 있는 범위는 한계가 있다[2,3]. 또한 선박의 중앙단면부 전체를 자동유한요소생성의 범위로 고려하므로, 사용자가 원하지 않는 요소생성의 원천을 가지고 있다[4~6]. 따라서 "조선전용 모델링프로그램"의 개발이 시급한 실정이다. 또한 이러한 CAD프로그램을 위해 개발되는 기반기술들은 앞으로 조선업계의 국가경쟁력 향상에 초석이 될 것이다. 이러한 선체구조해석전용 모델링 프로그램의 필요성에 따라 본 연구에서는 GUI(Graphical User Interface)를 통한 선체중앙단면부의 유한요소 해석모델링의 자동화 기법을 개발하였다.

2. 개발된 유한요소생성 프로그램의 특징

유한요소생성프로그램은 Windows95에서 개발되었다. 유한요소프로그램에서 생성된 데이터는 FTP를 이용해 PATRAN의 입력데이터로 들어가며 NASTRAN으로 해석한다(Fig. 1-1). 이는 주로 Workstation에서 작업이 가능하던 유한요소생성을 일반 PC에서 가능하게 하였으며, PATRAN용 입력데이터를 자동으로 생성하게 함으로 범용 CAD/CAM시스템과의 데이터호환이 가능하게 하였다. 또한 Class 즉 객체를 중심으로 구성함으로 객체중심적인 프로그램이 가능하게 하였다(3장 참조).

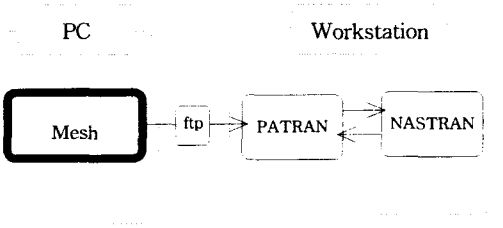

Fig. 1-1 Data flow diagram

선체중앙단면부 메쉬생성 Tool로서 개발된 프로그램의 주요특징은 다음과 같다[7~12].

◉ Top-down설계

선박을 대구조물, 중구조물, 소구조물로 나누어 모델링한다. 즉, 구조물의 문제분할을 Top-down방식으로 처리한다. 먼저 대구조물을 먼저 생성하고 중구조물을 계획하고, 또한 이들의 상관관계를 고려하여 하위레벨을 배치한다(Fig. 5-4~Fig. 5-6 참조).

◉ 사용자 편의 향상

현재 각광받고 있는 GUI를 이용하고, Parametric design기능을 접목하였다. 예를 들면 선체의 Bottom girder의 높이가 바뀔 경우 Hopper plate나 Hopper web plate의 형상도 바뀌어야 한다. 또한 완성된 모델링데이터가 있다면, 그 데이터를 이용하여 유사선박의 모델링을 쉽게 할 수 있는 재설계(Redesign)기능도 제공하였다(Fig. 5-4, Fig. 5-8 참조).

◉ 늑골, 종보강재 정보를 이용한 부재의 자동배치

선박의 Frame과 Longi. stiffener정보는 선박의 기본적인 형상정보이다. 모델링 작업시에 대부분의 선박에 사용되는 부재는 이들의 정보를 이용하여 배치된다. 이러한 형상정보는 사용자가 원하는 위치에 필요한 부재를 삽입하는 기준이 된다(Fig.

5-3 참조).

◉ 객체지향적 프로그래밍(3장 참조)

3. 객체 지향 모델링

개발된 프로그램에 적용된 객체 지향의 개념은 다음과 같은 특징과 장점을 가지고 있다[10].

◉ 캡슐화(Encapsulation)
◉ 정보은닉(Information Hiding)
◉ 상속(Inheritance)
◉ 다형성(Polymorphism)
◉ 생산성 향상
◉ 유지보수용이

유한요소프로그램에서 적용된 class는 다음과 같다. class CPoint_3D는 아래와 같이 x,y,z 값을 데이터로 가지며, 데이터를 처리할 함수들로 구성되어 있다.

```
class CPoint_3D{
protected:
        double x;
        double y;
        double z;
public:
    CPoint_3D(void);
    CPoint_3D(double _x, double _y, double _z);
    ~~~~~ 중 략 ~~~~~
    CLineP     operator|=(CPoint_3D _point);
    ~~~~~ 중 략 ~~~~~
    double Z(void) {return z;};
};
```

class CPoint_3D는 위상정보는 갖지 못하고 있다. 즉 기하학적인 정보만을 가지고 있으며 이 절점이 어떤 부재인지는 알 수 없다. 이의 해결을 위해 새로운 위상정보를 나타내는 class CGeoObject를 만들어서 이에 파생된 새로운 class를 만들고 CPoint_3D를 데이터로 가진다면 기존의 코드를 재사용할 수 있게 되는 것이다. CGeoObject는 name과 id라는 두 개의 데이터를 가지고 있다. name에는 부재의 이름이 포함되며, id에는 부재의 나타내기 위상정보가 들어간다.

```
class CGeoObject {
#define CGeoObject_SIZE
protected:
    char    name[20];
    int     id;
public:
    ~~~~~ 중 략 ~~~~~
};
```

class CGeoObject를 상속받은 class CPoint_3D를 CGeoPoint_3D라 하자. 아래에서와 같이 두 개의 class를 적절히 조합하여 사용함으로 새로운 class를 생성한다. 그리고 객체지향프로그램의 특징을 사용함으로 보다 효율적인 작업이 가능하게 된다.

```
class CGeoPoint_3D : public CGeoObject {
public:
CPoint_3D    Point;
~~~~~ 중 략 ~~~~~
  };
```

4. 유한요소생성 프로그램의 데이터구조

여러 가지 정보를 컴퓨터 내부에 구축하기 위해서는 데이터를 구조화하여야 한다[13,14].

4.1 다각형(Polygon)의 자료구조(Data structure)

연결리스트의 이론을 바탕으로 하여 다각형을 구현하였다. class Node는 순환 이중 연결리스트로서 데이터 멤버로 자신의 앞을 가리키는 *_prev와 자신의 다음을 가리키는 *_next로 된 두 개의 address를 가진다. 그리고 class Point는 x, y, z의 좌표값 만을 가진다.

```
class Node{
protected:
  Node *_next;
  Node *_prev;

public:
~~~~  중 략 ~~~~
  Node    *remove(void);
};

class Point{
public:
        double x,y,z;
public:
 ~~~~중 략 ~~~~
        double length(void);

};
```

위의 두 개의 class를 상속받아 class Vertex를 만들어 class Node 및 class Point를 조작할 수 있게 한다.

```
class Vertex:public Node, public Point{
public:
        Vertex():
        Point(ident,x,y,z){}
        ~~~~~~ 중략 ~~~~~~
        friend class Polygon;
};
```

class Vertex는 class Point에게 x, y, z 값을 할당하며, class Node에는 address를 할당한다. 또한 friend class로 class Polygon을 정의함으로 class Vertex의 모든 기능들을 사용할 수 있게 하였다.

```
class Polygon{
private:
   Vertex *_OutVertex;
   ~~~~~~ 중 략 ~~~~~~
   int get_angle_num();
};
```

class Polygon은 class Vertex들을 이중 연결리스트(Doubly linked list)가 되도록 class Vertex의 class Node를 조작하며, x, y, z의 값은 class Vertex에게 x, y, z의 값을 넘겨준 다음, class Vertex가 class Node에게 정보를 넘겨주게 하였다.

4.2 Template class ListNode

Template는 여러가지의 데이터형을 허락함으로써 프로그램의 가독성을 높이는 도구이다. List는 ListNode를 조작한다. 그리면 다음과 같다.

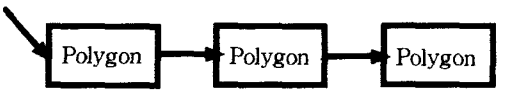

Fig. 4-1 Template class ListNode

5. 유한요소생성프로그램

5.1 좌표계

좌표계는 아래와 같이 설정하였다.

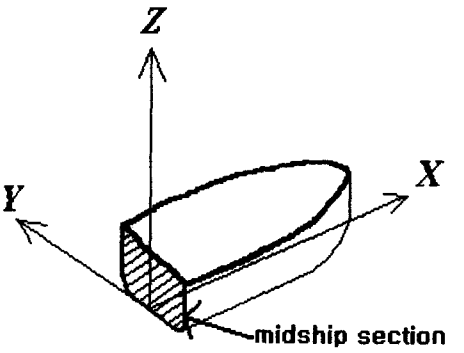

Fig. 5.1 Coordinate system

5.2 선종의 선택

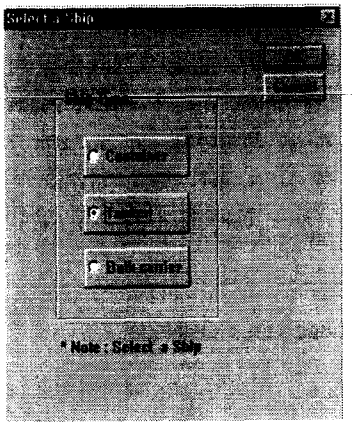

Fig. 5-2 Selection of a ship type

유한요소생성을 하기 위해 선종을 선택하는 과정이다.

5.3 늑골, 종보강재, 측판보강재 정보

늑골(Frame)정보와 종보강재(Longi. stiffener), 측판보강재(Sideshell stiffener)들의 정보를 입력한다. 이는 유한요소화 된 부재를 늑골이나 각종 보강재의 적정위치에 배치하기 위한 것이다.

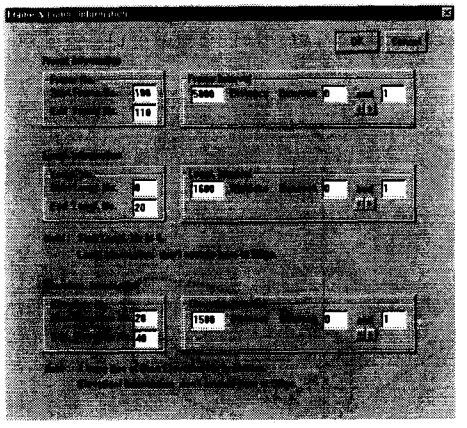

Fig. 5-3 Input of information of bottom longi. stiffener, sideshell longi. stiffener and frame

5.4 대부재 생성

대부재를 생성하기 위한 과정이다. 선체를 구성하는 부재 중에서 선체외곽을 형성하는 부분으로 초기 설계단계에서 이루어진다. 다음은 종방향의 종통부재들의 단면이다.

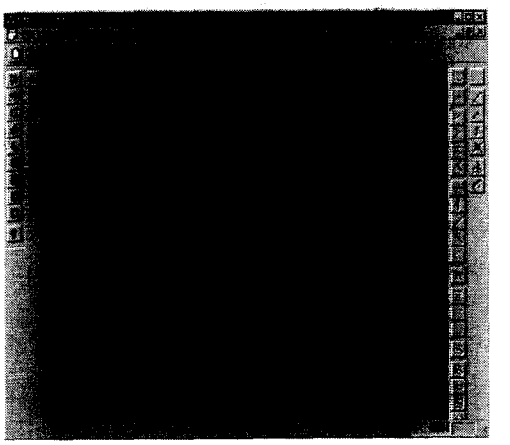

Fig. 5-4 View of information of bottom longi. stiffener, sideshell longi. stiffener and frame

5.5 Longitudinal bar 생성

위에서 생성된 대부재 가운데 종방향으로 배치되는 Bar를 생성한다.

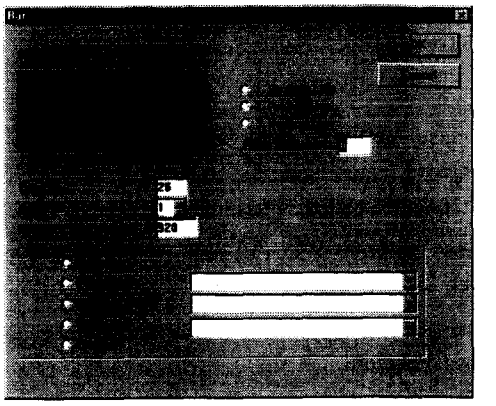

Fig. 5-5 Input of bar properties

종방향으로 배치되는 모든 Bar를 입력하였을 때의 단면형상이다

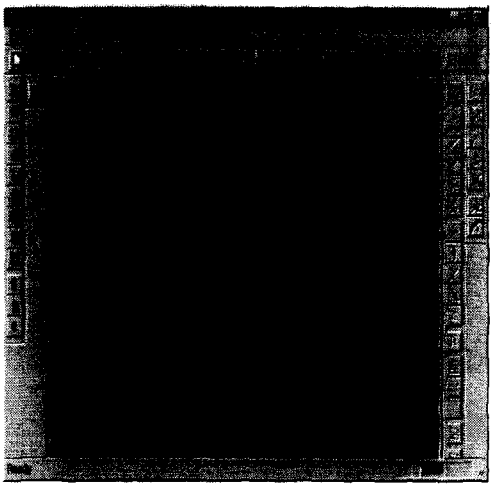

Fig. 5-6 View of a cross section of a tanker

이제 생성된 Line과 Point들을 Sweep을 통해 Plate와 Bar를 만든다[11]. 종통부재들의 분할은 늑골(Frame)정보를 이용하여 분할된다.

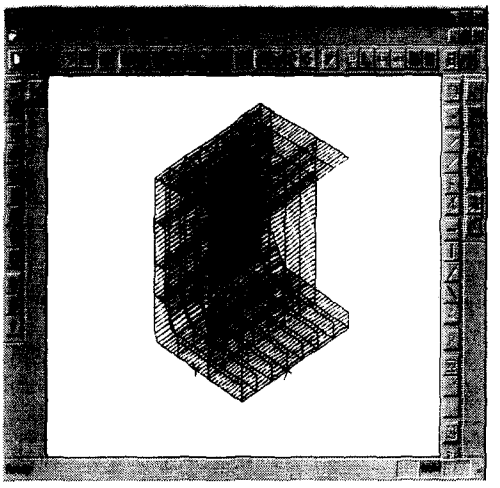

Fig. 5-7 Sweep of a cross section of a tanker

5.6 횡부재 생성

대구조물을 지탱하는 부재로 선체구조를 보강하는 평판들로서 Bulkhead, Web cross section, Stringer plate 등을 생성한다. 본 논문에서는 Web cross section이나 Bulkhead, Stringer plate 등과 같이 임의의 늑골 위에 위치하는 단면들을 요소생성 및 수정하기 위해 2차원 View를 생성하고 가상의 교차선(Intersection line)을 생성하여 사용자가 원하는 모양의 유한요소를 생성할 수 있도록 하였다.

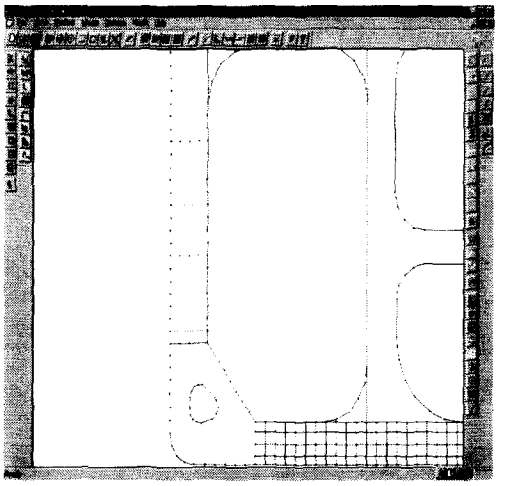

Fig. 5-8 Transverse web section of a tanker

다음은 Web cross section의 분할 예이다. 이 과정은 선박의 형상을 고려하여 요구조건, 규정조건, 재료의 유용성 등의 조건 및 사용자의 경험, 지식을 이용하여 유한요소를 생성할 수 있게 하였다. 분할된 Plate사이에 위치하는 Bar요소들의 특성(Property)을 주기 위하여 마우스로 원하는 부분의 Bar들을 펜스(Fence)기능을 이용하여 선택한다.

다음은 2차원 단면의 구멍이 있는 평판을 분할한 모습이다. Hopper plate의 경우 먼저 구멍주위의 Bar들을 기준으로 가상의 교차선(Intersection line)을 생성하여 유한요소를 생성한다. 그리고 구멍의 주위의 Plate는 사용자가 적당한 크기로 유한요소를 생성하게 한다.

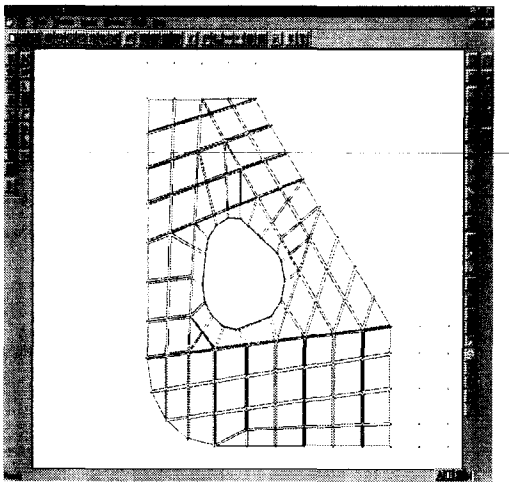

Fig. 5-9 Hopper plate after division

격벽 앞, 뒤에서 Web section의 형상을 고려할 필요가 있다. 그러므로 분할된 Plate를 제거하거나 생성할 수 있는 기능이 필요하다. 다음은 Web cross section가운데 Plate를 제거한 모양이다.

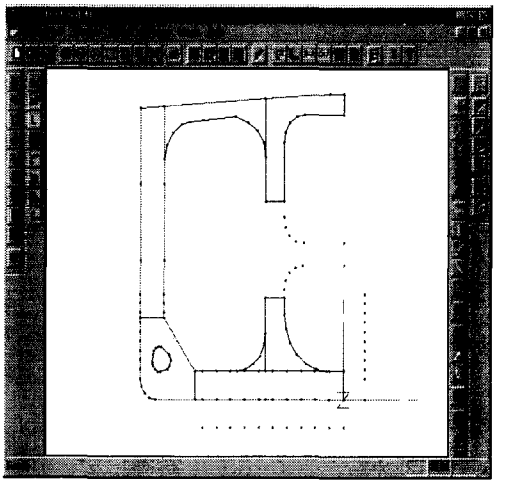

Fig. 5-10 Mesh deletion

그리고 새로운 Plate를 생성하였을 때의 모습이다.

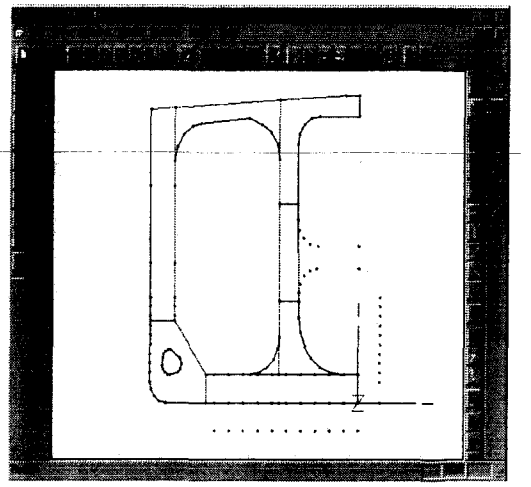

Fig. 5-11 Mesh Creation

유한요소를 생성 한 후 절점이 공유되었는지 검사할 필요가 있다. 이를 위하여 공유되지 않은 절점이나 5각형 이상의 요소는 2차원 View에 보여지게 하였다.

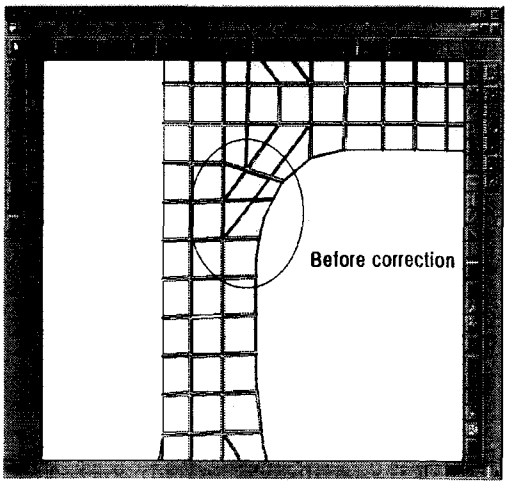

Fig. 5-12 Meshes before correction

공유가 되지 않은 절점은 절점을 이동함으로써 절점의 공유가 가능하도록 하였다.

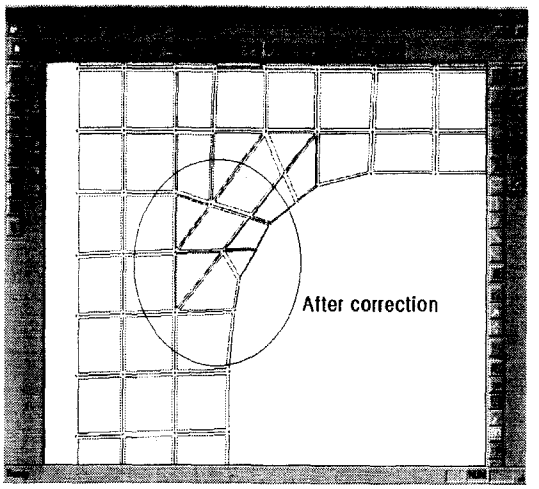

Fig. 5-13 Meshes after correction

다음은 Web cross section이 완전하게 분할된 모습이다.

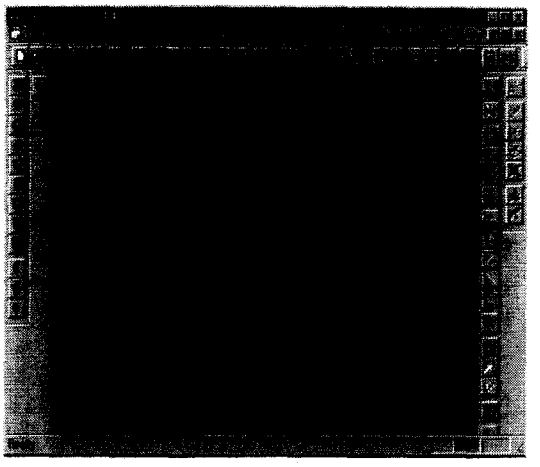

Fig. 5-14 Transverse web section after mesh generation

이렇게 완성된 Web cross section이 위치하는 곳을 늑골의 위치정보를 이용하여 원하는 곳에 배치한다.

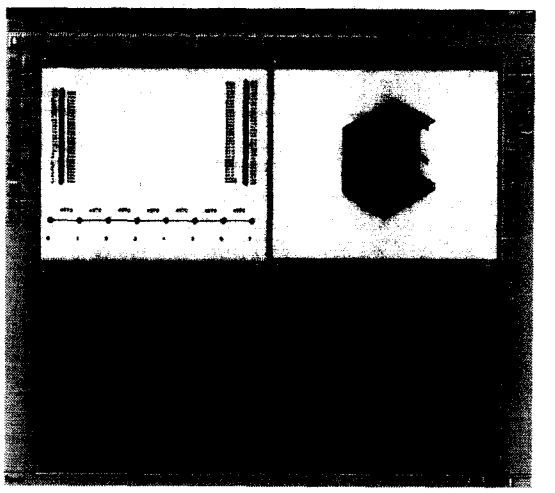

Fig. 5-15 Transverse web section mounted on the selected frame position

격벽의 분할과정도 Web cross section의 유한요소생성과 동일하다. 다음은 격벽이 완전히 분할된 모습이다.

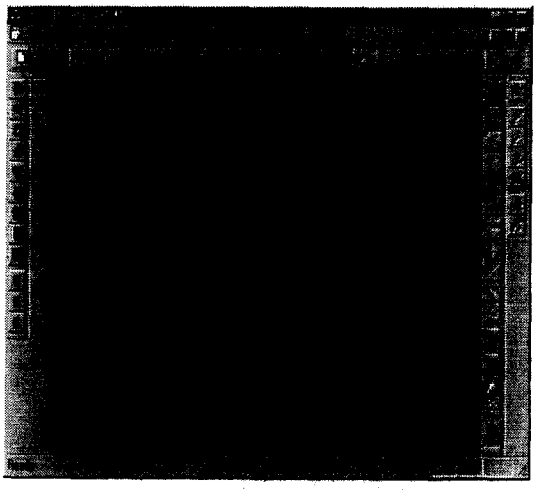

Fig. 5-16 Division of a bulkhead

다음은 Stringer plate의 분할된 모습이다.

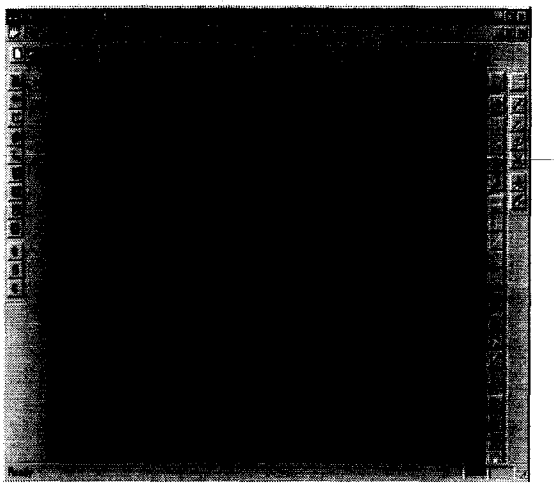

Fig. 5-17 Division of a stringer

유조선의 전체 유한요소생성후의 모습은 다음과 같다.

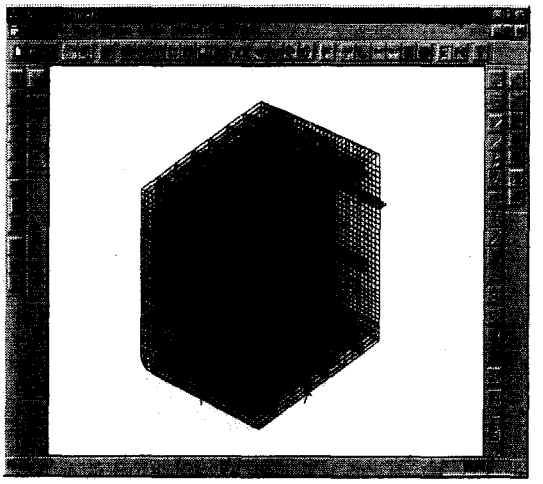

Fig. 5-18 Finite element model of a middle body of a tanker

마찬가지의 과정을 거쳐서 컨테이너(Container)의 전체유한요소를 생성했을 때의 모습은 다음과 같다.

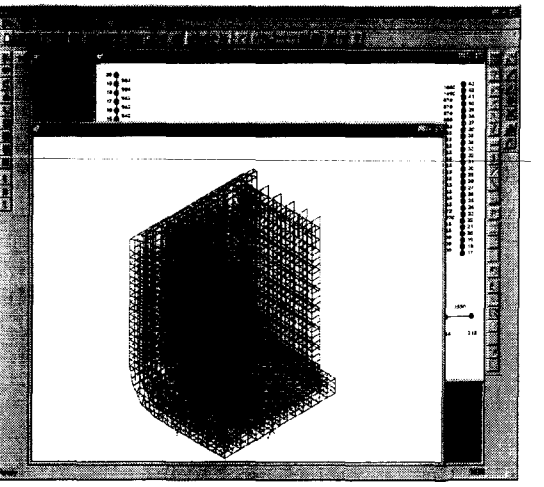

Fig. 5-19 Finite element model of a middle body of a container

6. 결 론

본 논문에서 전용 유한요소생성프로그램을 개발함으로써 기존의 범용CAD/CAM에서의 유한요소생성의 공수낭비를 절감하고자 하였다. 이는 선체구조가 일정한 형상을 유지하기 때문에 선체구조의 특성을 데이터베이스화하여 필요한 부재형상을 불러와 쉽게 선체모델의 생성이 가능하게 하였다. 또한 구조해석모델링의 경험이 없는 초보자라도 손쉽게 사용할 수 있도록 GUI를 구현하였으며, Workstation이 아닌 일반 PC에서 사용이 가능하도록 Windows95 환경하에서 MFC(Microsoft Foundation Class)를 기반으로 한 Visual C++로 개발하였다. 또한 객체지향프로그램 기법과 CAD/CAM기술의 도입으로 생산성 향상 및 프로그램의 유지 및 보수가 용이하도록 하였다. 선종당 보통 1~3개월 소요되던 선체중앙부 유한요소생성이 2일 이내에 가능하여 보다 빠른 시간 내에 유한요소해석이 가능하도록 하였다. 따라서 선박구조해석시 구조해석의 효율성을 증대시켰으며, 선박구조해석의 기술을 한단계 높이는 계기가 되었다. 계속되는 연구에서는 자동하중생성을 위한 알고리듬의 개발이 필요하다.

8. 후 기

본 연구는 사단법인 한국선급 지원에 의한 "선체유한요소해석 모델링의 자동화 기법에 관한 연구"의 연구결과 중 일부이다.

참고문헌

[1] 신현경, 박규원, "선체해석을 위한 3차원 모델링 전처리 프로그램 개발", 대한조선학회 추계연구발표회, 1993

[2] 신보성, 채수원, 박삼진, "사각형 유한요소의 자동생성 기법에 관하여", 대한기계학회 추계학술대회 논문집, 1992

[3] 정현석, 김용환, "Delaunary 삼각형기법을 이용한 유한요소망의 자동생성과 격자 재구성에의 응용", 대한기계학회 논문집, 1996

[4] 박성환, 신종계, "유조선 구조해석을 위한 유한요소입력자동화 프로그램의 개발", 대한조선학회 논문집, 1992

[5] 곽원섭, 홍윤환, 이건종, 권진철, "PRODUCT CARRIER선의 구조 해석 전용모델링 프로그램 개발", 대한조선학회 추계연구발표, 1995

[6] 염재선, "객체지향개념을 이용한 이중선각 유조선의 모델링과 구조해석모델의 자동생성에 관한 연구", 서울대학교 박사학위논문, 1995

[7] 신현경, 박규원, 손호철, 권명준, 송재영, 김종현, 강호승, 이정렬, "파형격벽을 갖는 산적화물선 중앙 단면부의 요소 생성에 관한 연구", 대한조선학회 춘계연구발표, 1997

[8] 김이두, "Top-down개념에 의한 건축구조물 설계", 한국과학기술원 박사학위논문, 1993

[9] 荷井康治, 伊澤田野己등, "NK Advanced Ship Structural Analysis Support System", 일본해사협회회지, No.232, 1995

[10] Bjarne Stroustrup, "The C++ Programming Language", Addison Wesley, 1991

[11] Michael E. Mortenson, "Geometric Modeling", John Wiley & Sons, 1985

[12] David F. Rogers, J Alan Adams, "Mathematical Elements for Computer Graphics", McGraw-Hill, 1985

[13] Ellis Horowitz, Sartaj Sahni, Dinesh Mehta, "Fundamentals of Data Structures in C++", Computer Science Press, 1995

[14] Adam Drozdek, "Data Structure and Algorithms in C++", PWS Publishing Company, 1995