

## Visual Basic을 이용한 구조해석 프로그램 개발에 관한 연구

### A study on the Development of Structural Analysis Program using Visual Basic

이 상 갑\*, 장 승 조\*\*1

Lee Sang-Gab and Chang Seung-Cho

#### ABSTRACT

The objective of this paper is to develop a finite element structural analysis program using VB(Visual Basic) which has been recently getting popular as development tools of application program for Windows. VB provides several functions to develop an application program easily by supporting event-driven programming method and graphic object as control data type. This system is an integrated processor including preprocessor, solver and postprocessor. Automatic mesh generation is available at preprocess stage, and graphic presentation of deformation and stress is also represented at postprocess one.

#### 1. 서 론

최근 PC(Personal Computer)의 성능향상 및 운영체제(Operating System)의 발전은 그동안 PC의 단점으로 여기던 처리속도와 GUI(Graphic User Interface), 다중처리(Multi-Tasking), 네트워크(Net-Working)등의 구현으로 응용범위가 크게 향상되었다. 특히 마이크로 소프트(Micro Soft)사에 의해 개발된 윈도우(Windows)[1]의 출현은 이러한 운영체제의 발전을 대표하는 것으로 볼 수 있다. 운영체제의 발전은 곧 그 운영체제에 맞는 응용 프로그램(Application Program)의 개발을 서두르게 하고 있으며, 프로그램 개발과 관련하여 이미 선진국에서는 윈도우 응용 프로그램 개발에만 집중하는 경향이 나타나고 있다. 운영체제의 발전은 사용자 입장에서는 편리성을 제공하지만 프로그램 개발자(Programmer)의 경우에는 프로그램 개발을 더욱 어렵게 하여 다양한 분야에서 요구되어지는 응용 프로그램 개발을 더디게 하는 결과를 낳고 있다.

그러나 윈도우 응용 프로그램을 개발하는데 사용되는 「Borland C++」[2]나 「Turbo Pascal for Windows」[3]와 같은 개발 도구(Development Tool)의 발전은 이러한 문제점들을 크게 개선하여 좀 더 쉽게 응용 프로그램을 개발할 수 있게 하였다. VB도 윈도우용 응용 프로그램 개발 도구로써 손쉽게 프로그램을 개발할 수 있는 기능을 제공하고 있다. 본질적인 의미의 객체지향(Object-Oriented)[4]을 지원하지는 않지만 컨트롤 형태(Control Type)의 그래픽 객체(Graphic Object)를 이용하여 개개의 요소로써 프로그램을 조립하듯 제작할 수 있다. VB에 있는 그래픽 객체들은 프로그램에서 필요로 하는 모든 객체와 그 객체가 반응하는 사건(Event)들을 이

\* 한국해양대학교 조선공학과 교수

\*\* 한국해양대학교 조선공학과 석사과정

미 갖추고 있으며 프로그램 인터페이스(Interface) 구성에 필요한 요소들도 거의 갖추고 있다[5]. 이러한 기능들은 윈도우즈 운영이 내·외부에서 발생하는 무수한 사건에 응답하는 방식으로 처리 되는 경우와 동일하여 프로그램의 신뢰성을 높일 수 있다. 또한 윈도우즈 API(Application Program Interface) 함수의 이용과 다중처리를 위하여 다른 언어로 제작된 동적 연결 라이브러리 (DLL : Dynamic Link Library)와의 접속은 VB의 기능을 크게 확장시킬 수 있고 실질적으로 매우 쉽게 이루어 진다. 그 밖에도 DDE(Dynamic Data Exchange)와 OLE(Object Linking and Embedding)를 이용하여 다른 프로그램과의 자료 교환 및 제어가 가능하다[6].

본 연구에서는 VB를 이용하여 공학용 프로그램 설계를 쉽게 구현하는 방법을 제시하고 구조 해석용 유한요소 프로그램을 개발하고자 한다. 시스템 구성은 전처리, 해석, 후처리의 3단계로 이루어지는 통합운영 시스템(Integrated Process System)이다. 전·후 처리(Pre/Post Process)에 있어서는 자동요소분할생성(Automatic Mesh Generation)을 통한 그래픽 모델링(Graphic Modeling)과 응력분포(Stress Distribution), 변위(Displacement)등과 같은 결과를 그래픽으로 표현할 수 있는 기능들을 포함하고 있다. 프로그램은 전반적으로 컨트롤 배열(Control Array), 동적배열(Dynamic Array)과 같은 동적 요소들을 사용하여 메모리(Memory)의 사용 범위를 넓히고 동적 프로그램의 특성을 갖게끔 설계하였다. 본 논문에 기술되는 내용은 전·후처리기의 설계를 중심으로 표현하였다. 후처리의 내용은 전처리의 설계 기법과 거의 동일하기 때문에 전처리에서 제시되지 않은 사항들 위주로 기술하였다.

## 2. 시스템 설계 및 개요

본 유한요소 구조해석 시스템은 전처리, 해석, 후처리의 3단계로서 구성되어 있고 각 단계에서의 부메뉴를 포함하여 세부적인 처리과정을 갖는다. 시스템의 전체적인 구조는 Fig. 2.1과 같다.

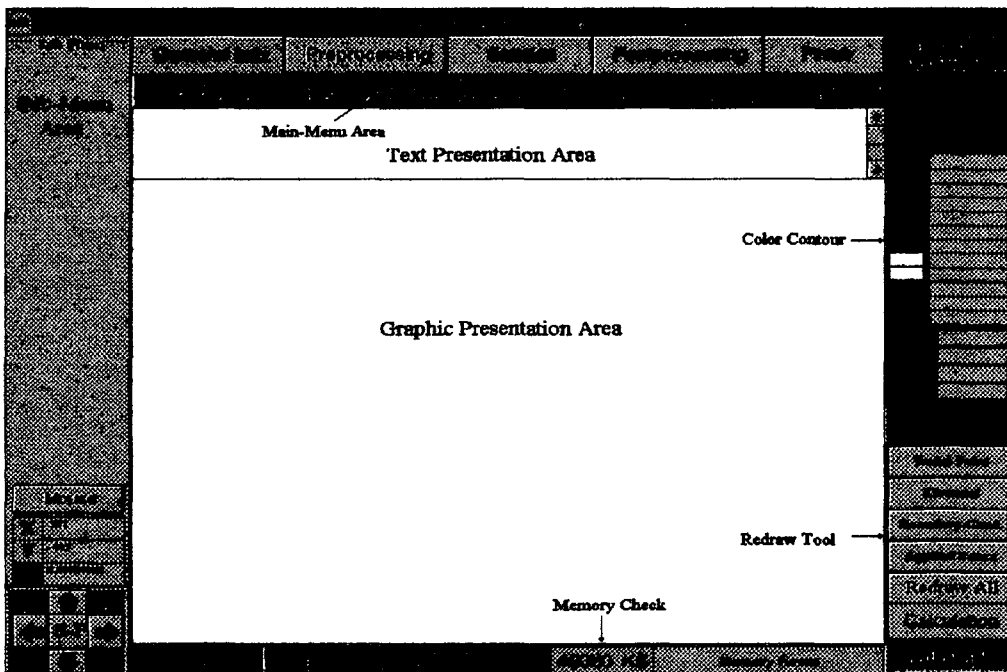


Fig. 2.1 Overall structure of integrated process system

시스템 구성요소는 주메뉴 영역(Main-Menu Area)과 부메뉴 영역(Sub-Menu Area), 그리고 그래픽 표현 영역(Graphic Presentation Area)으로 나뉘어진다. 주메뉴 영역은 VB에서 제공되는 버튼(Button) 컨트롤로써 구성되고, 부메뉴 영역에서의 그래픽 요소는 판넬(Panel) 및 버튼이 사용된다. 그래픽 표현 영역은 그림상자(Picture Box)의 요소로 구성된다. 각각의 메뉴를 구성하는 그래픽 객체의 특성이나 사건 프로시저(Procedure)의 작성, 자료의 범위(Scope)등은 이 후에 설명되는 각 단계의 세부 설계 방법에서 자세히 표현하고자 한다.

## 2.1 전처리기의 설계

전처리 단계에서는 절점좌표(Nodal Coordinates), 요소연결(Element Connectivity), 경계조건(Boundary Condition), 작용하중(Applied Loading), 재료 특성치(Material Properties)등과 같은 새로운 데이터를 생성하는 과정과 기존의 데이터를 읽는 과정으로 구성되어 있고, 확인 및 수정할 수 있다. 전처리를 구동시킬 때는 주메뉴 영역에 있는 "Preprocess" 버튼의 사건 프로시저를 작성하여 부메뉴 영역의 세부적인 입력 메뉴의 속성을 변경 또는 정의함으로써 이루어진다. 적용되는 사건 프로시저는 마우스 클릭(Mouse\_Click) 사건에 응답하도록 되어있다. 전처리기의 부메뉴는 동일한 그래픽 객체의 이름(Name)을 갖는 동적 객체 배열로 구성된다. Fig. 2.2은 전처리기의 데이터 처리 과정 및 객체 구성을 나타낸다.

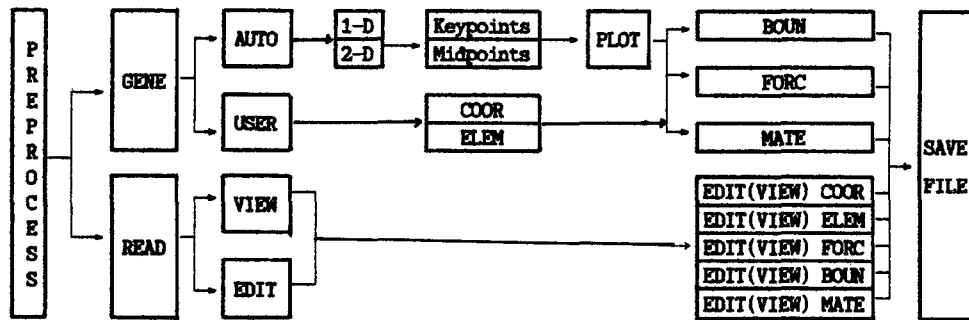


Fig. 2.2 Data input process at Preprocessor

주메뉴 「Preprocess」 버튼은 Table 2.1과 같은 사건 프로시저를 가지고 있고, 버튼 객체가 가지는 사건 프로시저와 응답도 표현되어 있다. 사건 프로시저는 모든 것을 표현하지는 않고, 개발 시스템에서 주로 사용되는 사건에 대한 것만 기술하고 있다.

Table 2.1 Event procedure and response of "Preprocess" button

Graphic Object	Name	Event Procedure	Response
Button	PREBUT	Mouse_Click()	Sub-Menu.Visible=TRUE
		Mouse_DblClick()	
		Get_Focus()	prebut.forecolor=&HFF0000
		Lost_Focus()	prebut.forecolor=&H80&
		Key_Press()	

객체에 대한 응답 프로시저는 개발자가 필요로 하는 것에만 코딩을 한다. 즉, 마우스 클릭 사건에 응답하는 동작을 필요로 할 경우 「Mouse\_Click()」 프로시저에 동작되는 코드를 작성한다. 마우스의 촛점(Focus)과 관련한 프로시저들은 현재의 단계를 표현하는 방법으로 주로 사용되고

동작하지 않는 메뉴들과 비교될 수 있게 배경에 대한 글자색을 변화시킨 것이다. 이것은 버튼 객체에 대한 객체속성(Object Attribute)을 변화시킨 것으로, VB에서 지원되는 그래픽 객체는 객체의 동작을 결정하는 사건 프로시저와 크기, 위치, 이름등의 객체의 상태를 나타내는 속성을 가지고 있다. 부메뉴의 경우 같은 종류의 객체들을 중복해서 사용해야 할 경우 객체 이름 및 속성을 정의하고, 「Index」를 첨자로 하여 배열화하는 동적 객체배열을 적용한다. 동적 객체배열을 사용하면 같은 사건 프로시저를 공유하게되며 코드를 매우 간략하게 할 수 있다. Table 2.2는 자동 요소분할과정[7]의 「Index」를 이용한 부메뉴의 일부 구조를 나타낸다.

Table 2.2 Sub-menu structure of Preprocessor using object array

Object	Name	Procedure	Index	Caption	Code
Panel	prepro()	Mouse_Click()	1	1-D	Select Case Index+1
			2	2-D	Case 1
			1	KeyPoints	:
			2	MidPoints	End Select
			3	PLOT	
			1	BOUND	For i=1 to 2 (or 3)
			2	FORC	prepro(i).visible=False
			3	MATE	Next i

부메뉴를 구성하는 그래픽 객체는 판넬로써 대부분 3차원 효과를 가지는 배경으로 많이 사용되나 프로그램의 흐름을 표현하는 속성이 다양하여 메뉴로 사용된다. 전처리기의 부메뉴에서 입력되는 내용은 그래픽으로 동시에 표현된다. Fig. 2.3는 중앙에 구멍이 있는 평판에 인장력이 작용하고 있는 경우를 평면응력(Plane Stress) 문제로 해석하기 위하여, 대칭조건을 고려한 1/4 자동요소분할을 전처리기에서 실행을 마친 화면이다. 부메뉴에 있어서 마우스 클릭 사건에 대한 객체의 응답관계 및 조건식, 호출함수는 Table 2.3과 같다.

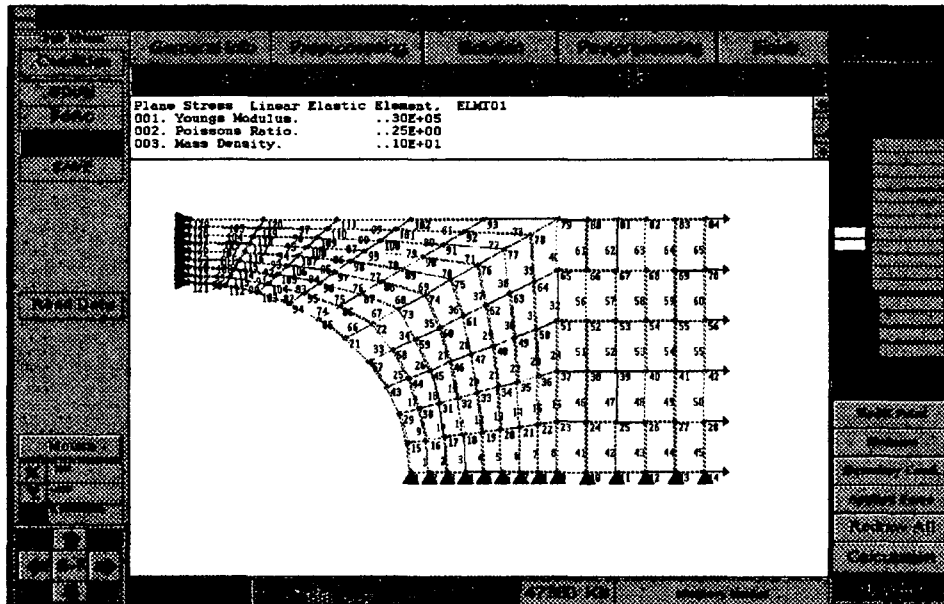


Fig. 2.3 Automatic mesh generation of input data at Preprocessor

Table 2.3 Action by sub-menu attribute transform

Name	Index	Condition	Caption Attribute	Call Procedure	Module File(.BAS)	Note
Prepro()	1	.and.	1-D		Form Level	TRUSS, BEAM
	2	.and.	2-D		..	TRIANGLE, QUADRILATERAL
	1	.and.	KeyPoint	Keypoint()	Mesh	Extraction of Keypoint
	2	.and.	MidPoint	Midpoint()	..	Extraction of Midpoint
	3	.and.	PLOT	Plot()	..	Data Creation & Graphic Modelling
	1	.and.	BOUN	Boun_Click()	Prepro	B.C. & Graphic Presentation
	2	.and.	FORC	Forc_Click()	Prepro	Loading & Graphic Presentation
	3	.and.	MATE	Mate_Click()	Prepro	Material Properties

자료의 입력은 키보드에 의해 정확한 좌표점을 입력하는 방법과 마우스를 직접 이용하여 좌표값을 입력하는 방법이 있다. 키보드에 의한 경우 VB에서 지원되는 대화 상자(Dialog Box)를 이용한다. 그림상자도 컨트롤 개념의 그래픽 객체로 모듈 수준에서의 접근은 모듈에 포함되어 있는 프로시저가 그래픽 객체에 출력을 필요로 할 경우 프로시저의 매개변수로 그래픽 객체를 넘기면 된다. 마우스를 이용하여 Keypoint 와 Midpoint 기능을 수행하는 과정은 Fig. 2.4과 같이 나타낼 수 있다. 마우스를 이용하는 경우 정확한 좌표값을 추출하기 위해 현재의 마우스 위치를 값으로 표현하여 사용자에게 알려준다. 위치에 대한 표현은 Fig. 2.4에서의 「Mouse\_Move()」 프로시저를 작성하면 된다. 마우스의 제어(Control) 기능을 가지는 버튼에서 반환되는 값은 ON으로 가정하여 나타낸다. 즉, 그래픽 표현 영역에서의 마우스 기능은 가능하다.

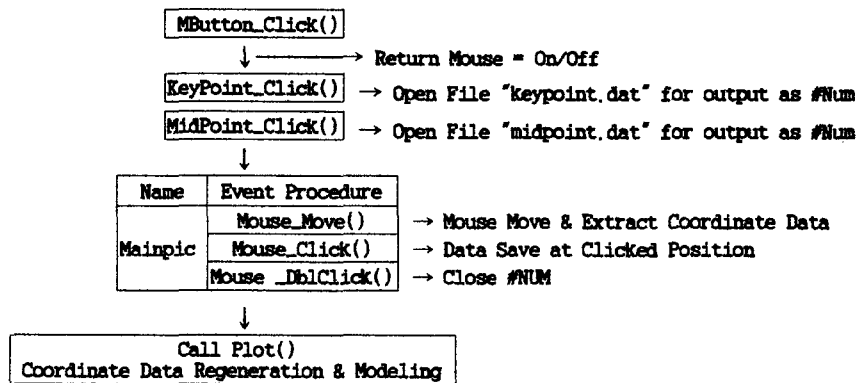


Fig. 2.4 Graphic modelling process using Mouse

## 2-2. 후처리 단계

후처리 단계는 해석 단계에서 얻어진 결과를 그래픽 및 텍스트로 표현하고, 프린터(Printer) 출력의 처리 과정을 갖는다. 따라서 외부 장치(External Device)와 연결되는 프로그램 코드와 API 함수 호출, DLL 연결등 복잡한 요소로써 설계된다. 부메뉴의 그래픽 객체의 구성은 패널과 버튼을 동시에 사용하며 그룹상자(Group Box)의 형태를 취한다. 후처리 단계의 전체적인 부메뉴 구성은 Fig. 2.5와 같다.

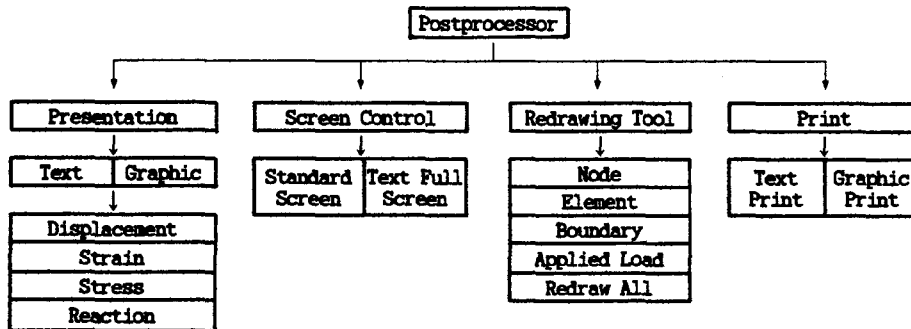


Fig. 2.5 Sub-menu structure at Postprocessor

후처리 단계는 위와 같이 크게 4 부분으로 나뉘어지고 각 부분들은 그림상자와 글상자를 중심으로 상호 연결성을 갖고 결과를 표현한다. 응력 분포등과 같이 색표현(Color Contour) 방식에 의한 결과의 표현은 각 요소의 가우스 점(Gauss Point)에서의 결과치 또는 요소 전체에 걸친 평균치를 이용하여 표현할 수 있다. 그러나 실질적으로 결과의 색표현에 있어서 직사각형의 형태를 갖는 요소는 VB의 함수로써 가능하지만 그외의 다각형 요소들은 VB 함수로써 표현하는 것이 불가능하다. 따라서 본 연구에서는 임의 형상의 요소에 색표현을 위해서 API 함수를 이용한다. API 함수의 호출은 아래의 코드와 같은 방식으로 표현할 수 있다[8]. 호출되는 함수는 다각형(Polygon)의 형상을 구성하는 함수와 내부의 색을 결정하는 함수로 구성된다. API 함수선언은 후처리 단계를 구성하는 모듈의 일반 선언부(General Declaration)에서 선언된다[9].

```

Declare Function Polygon Lib "GDI" (ByVal hdc As Integer, lpPoints As POINTAPI, ByVal nCount As Integer) As Integer
Declare Function SetPolyFillMode Lib "GDI" (ByVal hdc As Integer, ByVal nPolyFillMode As Integer) As Integer
  
```

Parameter	Type/Description
lpPoints	POINTAPI - The first POINTAPI structure in an array of nCount POINTAPI structures.
nCount	Integer - Number of points in the polygon.
hDC	Integer - Handle to a device context.
nPolyFillMode	Integer - Alternates filling or Fill based on drawing direction

변형된 요소의 표현방법은 그림상자의 그림 그리기(DrawStyle)속성을 점선으로 전환하여 표현하였다. Fig. 2.3에서 자동요소 분할된 평판에 인장력이 작용하고 있는 평면응력문제를 해석한 결과를 그래픽과 텍스트로 나타내면 각각 Fig. 2.6 ~ Fig. 2.8과 같다.

### 3. 결 론

본 연구에서는 VB를 이용하여 효율적으로 유한요소 구조해석 프로그램을 개발하였다. 윈도우즈 응용 프로그램이 필수적으로 갖게되는 GUI 구현은 VB가 지원하는 그래픽 객체로써 쉽게 이룰 수 있었고, 미리 정의된 그래픽 객체의 프로시저를 이용하여 다른 도구보다 개발 시간을 단축할 수 있었다. 윈도우즈 API 함수와 DLL을 사용함으로써 임의 요소형상의 그래픽 표현과정에서 발생하는 문제점을 해결하였다. API와 DLL의 적용은 VB 기능을 확대하여 구조해석 프로그램 설계 시에 발생하는 VB의 제한적 기능들을 충분히 확장시킬 수 있음을 알 수 있었다.

추후 지속적인 연구과제로서 요소 라이브러리(Element Library)의 확장과 세부적인 기능의 추

가가 필요하다고 사료된다.

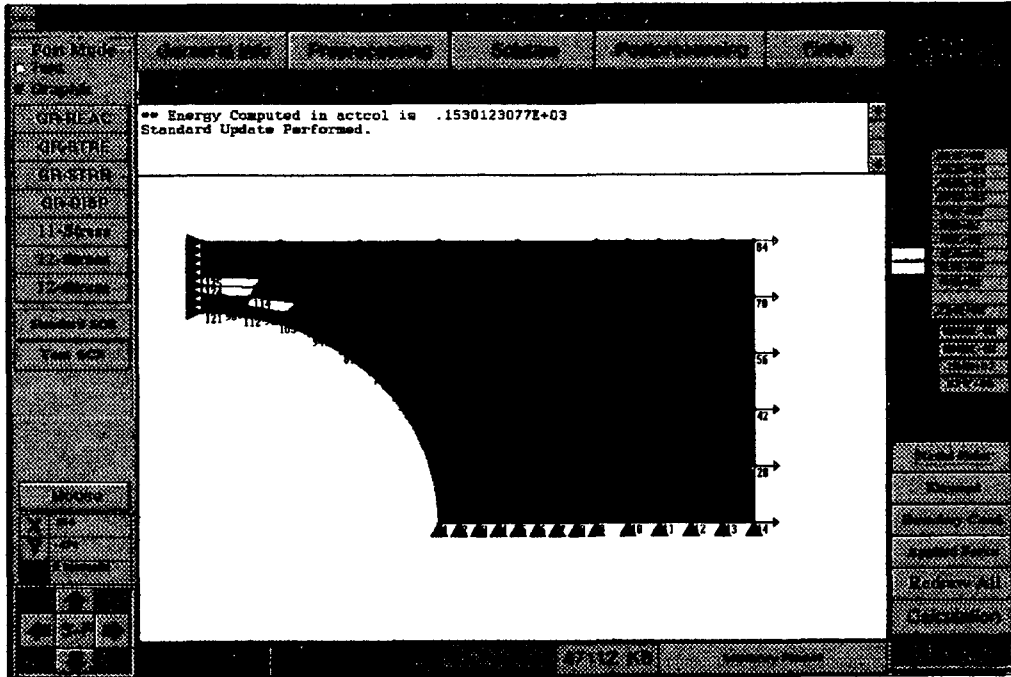


Fig. 2.6 Graphic presentation of stress distribution at Postprocessor

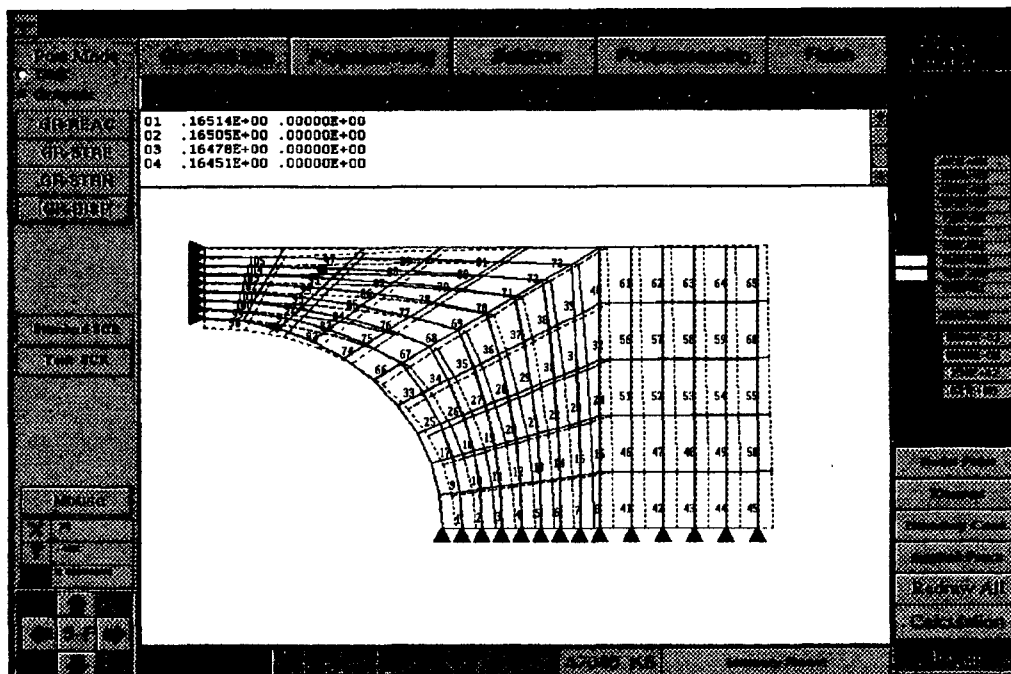


Fig. 2.7 Graphic presentation of deformed configuration at Postprocessor

Element	Node	DOF	U1	U2	U3	U4	U5	U6
7.044	3.522	1 1	001	5.954E+01	1.679E-02	1.772E+01		
26.289	3.522	1 2	001	5.954E+01	5.884E-02	-3.418E+00		
26.289	13.145	1 3	001	3.135E+01	5.818E-02	-3.410E+00		
7.044	13.145	1 4	001	3.135E+01	1.613E-02	1.773E+01		
40.377	3.522	2 1	001	3.530E+01	2.602E-01	1.219E+01		
59.623	3.522	2 2	001	3.530E+01	7.697E-01	4.064E-01		
59.623	13.145	2 3	001	1.959E+01	7.693E-01	5.019E-01		
40.377	13.145	2 4	001	1.959E+01	2.598E-01	1.229E+01		
73.711	3.522	3 1	001	1.184E+01	-1.885E-01	6.858E+00		
92.956	3.522	3 2	001	1.184E+01	-3.314E+00	4.775E+00		
92.956	13.145	3 3	001	6.396E+00	-3.314E+00	4.189E+00		
73.711	13.145	3 4	001	6.396E+00	-1.886E-01	8.272E+00		
7.044	20.189	4 1	001	1.215E+01	2.847E-04	1.880E+01		
26.289	20.189	4 2	001	1.215E+01	2.847E-04	5.802E-01		
26.289	29.811	4 3	001	-1.215E+01	-2.847E-04	5.802E-01		
7.044	29.811	4 4	001	-1.215E+01	-2.847E-04	1.880E+01		
40.377	20.189	5 1	001	7.991E+00	1.873E-04	1.730E+01		
59.623	20.189	5 2	001	7.991E+00	1.873E-04	5.315E+00		
59.623	29.811	5 3	001	-7.991E+00	-1.873E-04	5.315E+00		
40.377	29.811	5 4	001	-7.991E+00	-1.873E-04	1.730E+01		
73.711	20.189	6 1	001	2.542E+00	5.959E-05	1.286E+01		
92.956	20.189	6 2	001	2.542E+00	5.959E-05	9.046E+00		
92.956	29.811	6 3	001	-2.542E+00	-5.959E-05	9.046E+00		
73.711	29.811	6 4	001	-2.542E+00	-5.959E-05	1.286E+01		
7.044	36.855	7 1	001	-3.135E+01	-1.613E-02	1.773E+01		
26.289	36.855	7 2	001	-3.135E+01	-5.818E-02	-3.410E+00		
26.289	46.478	7 3	001	-5.954E+01	-5.884E-02	-3.418E+00		
7.044	46.478	7 4	001	-5.954E+01	-1.679E-02	1.772E+01		
40.377	36.855	8 1	001	-1.959E+01	-2.598E-01	1.229E+01		
59.623	36.855	8 2	001	-1.959E+01	-7.693E-01	5.019E-01		
59.623	46.478	8 3	001	-3.530E+01	-7.697E-01	4.064E-01		
40.377	46.478	8 4	001	-3.530E+01	-2.602E-01	1.219E+01		
73.711	36.855	9 1	001	-6.396E+00	1.886E-01	8.272E+00		
92.956	36.855	9 2	001	-6.396E+00	3.314E+00	4.189E+00		
92.956	46.478	9 3	001	-1.184E+01	3.314E+00	4.775E+00		
73.711	46.478	9 4	001	-1.184E+01	1.885E-01	6.858E+00		

Fig. 2.8 Text presentation of stress results at Postprocessor

### 참고 문헌

- [1] Microsoft, Window 3.1 User Guide, 1993.
- [2] Clayton, W., Borland C++ for Windows Programming, 2Ed., Prentice-Hall, 1994.
- [3] Leestma, M., Object-Oriented Programming TurboPascal, Merrill-Macmillan, 1993.
- [4] James Martin, Principles of Object-Oriented Analysis and Design, Prentice-Hall, 1993.
- [5] Robert, W. Stewart, Graphic Programming with Visual Basic, SAMS, 1995.
- [6] Ori Gurewich & Nathan Gurewich, How to Create Real-World Application with Visual Basic, SAMS, 1994.
- [7] Zienkiewicz, O. C. and Philips, D. V., "An automatic mesh generation scheme for plane and curved surfaces by isoparametric coordinates", International Journal for Numerical Engineering, Vol. 3, pp. 519-528, 1971.
- [8] Daniel Appleman, Visual Basic Programmer Guide to the Windows API, Ziff-Davis, 1993.
- [9] Zane Thomas & Robert Arnson, Visual Basic How-To, 2Ed., Waite-Group, 1993.