

마이크로 컴퓨터를 이용한 유한요소해석의 효율적 Postprocessing

Efficient Postprocessing for Finite Element Analysis on Microcomputers

이 재 영*
Jae Young Lee

ABSTRACT

This study was intended to provide efficient algorithms for high quality postprocessing on the basis of microcomputers with limited capacity and functionality. Improved methods of postprocessing including stress contouring, internal force diagraming, and displacement animation, were proposed and implemented into a new finite element system. Visualization of three-dimensional structural behaviors was treated with special emphasis.

1. 서론

최근에 마이크로컴퓨터는 성능이 급속도로 향상되고 있는 반면에 가격이 점차 저렴해져서 그 보급이 가속화되고 있으며, 이로 인해 컴퓨터의 대중화가 실현되고 있다. 광범위한 여러 분야에서 마이크로컴퓨터를 위한 소프트웨어가 쏟아져 나오고 있으며, 대형 컴퓨터용의 기존 프로그램들이 마이크로 컴퓨터용으로 전환되어 보급되는 추세에 있다. 유한요소해석도 그 예외는 아니다. 최근에 마이크로컴퓨터를 겨냥한 유한요소해석 프로그램들이 상품화되어 보급되었다^(2,3,4,12). 마이크로 컴퓨터를 이용한 유한요소해석이 일층더 원활하고 효율적으로 이루어지기 위해서는 마이크로컴퓨터의 특성에 적합한 algorithm과 운용방법이 마련되어야 하며, 이를 바탕으로 프로그램이 개발되어야 할 것이다⁽⁷⁾.

이 연구는 마이크로 컴퓨터용의 새로운 유한요소해석 시스템 개발과 관련하여 수행되었으며, 대형 컴퓨터용의 기존 프로그램^(5,8)을 능가하는 postprocessor를 개발하는데 목표를 두고, 이를 위한 효율적인 postprocessing algorithm을 마련하려는 의도에서 이루어졌다. 이 연구에서는 應力圖(또는 變形率圖), 주응력도 및 주응력의 방향 표시, 변위형태의 도화 및 animation, 골조구조물의 굽힘모멘트도 및 전단력도의 도화를 중심으로 postprocessing의 문제를 다루었다. 특히 입체적인 구조물의 해석결과를 마이크로 컴퓨터를 이용하여 쉽게 이해할 수 있도록 可視化하는 새로운 방법을 제시하였으며 이를 프로그램으로 이행하였다.

2. 마이크로 컴퓨터의 제한성

마이크로 컴퓨터를 이용한 유한요소해석도 대형 시스템에 의한 해석과 그 과정에 있어서는 별로 다를 바가 없다. 그러나 실제적인 실행에 있어서는 마이크로 컴퓨터가 지니고 있는 성능과 용량의 한계로 인하여 발생하는 여러가지 문제점들을 고려하지 않으면 안된다⁽⁹⁾. 그중에서도 postprocessing과정에서 극복해야 할 마이크로 컴퓨터의 중요한 제한성은 다음과 같다.

가. 기억용량

응용프로그램에서 직접적으로 할당하여 사용할 수 있는 기억용량은 내장된 RAM의 용량뿐만 아니라 operating system과 프로그램의 compiler에 따라서 달라진다. 지금 현재 가장 널리 쓰이고 있는 마이크로 컴퓨터의 operating system은 DOS(M.S. 또는 P.C.)이며, 이 경우에 프로그램에서 사용할 수 있는 기억용량은 640Kbyte 이하로 제한되어 있다. 이 연구에서는 640Kbyte 이하의 기억용량을 전제로 하였다. 유한요소해석의 postprocessing에서 많은 기억공간을 차지하는 데이터의 종류는 다음과 같이 세가지로 분류할 수 있다.

- 1) 요소, 절점 등 구조물의 형상 데이터
- 2) 변위, 응력, 변형률, 에너지 밀도 등 해석결과로 얻어지는 데이터

3) Postprocessing image를 만들기 위해 필요한 데이터

Graphic display의 bit plane수가 많거나 depth buffer가 있다면 세번째 부류의 데이터가 차지하는 기억공간은 크게 문제되지 않는다. 그러나 마이크로 컴퓨터용의 graphic display (display processor와 monitor)는 대부분 그렇지 못하므로 기존의 방법에 의한 hidden surface처리, shading 및 응력도의 도화 등은 비교적 큰 기억용량을 필요로 한다.

나. 실행 속도

최근에 microprocessor의 처리속도가 급격히 향상되고 있다. 그러나 아직은 대체로 2 MIPS 이하이며, 마이크로 컴퓨터의 큰 약점으로 남아있다. Postprocessing과정에서는 graphic display의 도화속도도 문제가 된다. Interactiveness를 유지하기 위해서는 빠른 속도로 계산과 display를 실행하는 algorithm을 적용하지 않으면 안된다.

다. Display의 解像度

마이크로 컴퓨터용 graphic display의 解像度는 400×200, 640×480, 1024×768, 1280×1024 등 여러가지가 있으며, 지금 현재는 640×480 (VGA, PGC 등)이 가장 많이 사용되고 있으나, 곧 1024×768 (IBM 8514/A 등)급이 보편화될 전망이다. Postprocessing에 있어서는 복잡한 해석결과를 해상도가 낮은 display의 화면에 표현하는데 어려움이 있다.

라. Display의 bit plane數

Bit plane 수는 동시 표현 가능한 색상의 수와 직접적인 관련이 있으며, bit plane masking에 의해서 독립적으로 저장할 수 있는 單色 畫像의 수라고 볼 수도 있다. 마이크로 컴퓨터용 graphic display의 bit plane 수는 보통 8 이하이며, 1, 2 또는 4인 경우가 흔하다. 최근 들어서 8 bit plane의 display (PGC, VGA, IBM 8514/A 등)가 많이 보급되고 있다. Postprocessing 과정에서 stress contouring과 shading을 원활하게 실행하기 위해서는 많은 bit plane이 필요하다.

마. Firmware의 기능

마이크로 컴퓨터용 graphic display의 firmware는 line drawing, polygon fill, character drawing 등 극히 제한된 기능만을 갖고 있다고 가정하여, 그 외의 모든 기능이 소프트웨어적으로 처리되도록 하여야 한다.

이 연구에서는 위에 열거한 여러가지의 제약조건을 갖고 있는 마이크로 컴퓨터의 환경하에서 고성능의 graphic 기능을 갖춘 대형 시스템에서와 동등하거나 또는 그 이상의 결과를 달성하기 위한 postprocessing algorithm을 개발하고, 이를 실용적인 프로그램으로 이행하여 그 유효성과 효율성을 검토하였다.

*정회원, 전북대학교 농공학과 조교수

3. 응력도의 도화 (stress contouring)

Vector refresh type이나 storage tube의 graphic display, 또는 plotter를 이용한 응력도의 도화는 line drawing으로 이루어질 수 밖에 없으나 raster graphic display를 이용할 경우에는 이 보다는 point drawing 또는 area fill에 의하여 응력의 크기를 여러가지 색으로 표시하는 것이 더 효과적이다⁽¹³⁾. 마이크로 컴퓨터의 환경하에서는 raster graphics가 보편화 되고 있는 점을 감안하여, 이 연구에서는 area fill에 의한 다각형법과 point drawing에 의한 주사선법을 고안하고, 기존의 등변수보간법을 보완하였으며, 이들 세 방법을 비교하였다.

가. 다각형법

다각형법은 구조물 전체의 최대 및 최소응력을 구하고, 그 사이를 몇개의 응력구간으로 나누어, 각 구간을 지정된 색으로 polygon fill에 의해 채색하는 방법이다. 이 방법에 의하면 먼저 각 요소를 몇개의 小三角형으로 분할한다. 이는 polygon fill을 효율적으로 실행하기 위한 것이다. 이들 소삼각형들을 독립적으로 응력의 크기에 따라서 몇개의 다각형으로 분할하여 해당 되는 색으로 채워나가면 응력도가 완성된다. 이 연구에서는 소삼각형을 채우는 방법을 두가지로 나누어서 검토하였다. 그 첫번째는 그림 1(a)와 같이 소삼각형 내에서 두 등응력선 사이를 다각형으로 채우는 방법이다. 먼저 좌표를 정해진 응력 값에 따라 분할하고, 이 점들을 연결하는 다각형을 polygon fill에 의해 해당되는 색으로 채색한다. 이 연구에서는 편의상 이 방법을 다각형법 A라고 표시하였다.

두번째는 그림 1(b)와 같이 소삼각형 안에서 등응력선의 한 쪽 부분, 즉 응력이 큰 부분을 모두 내포하는 다각형으로 채워나가는 방법이다. 이 방법에 의하면 작은 응력의 색으로 채색되었던 다각형 안에 큰 응력의 다각형이 겹치게 된다. 이 연구에서는 편의상 이 방법을 다각형법 B라고 표시하였으며, 개략적인 실행 순서는 다음과 같다.

- 1) 응력의 크기가 작은 순서부터 절점의 번호를 부여한다.
- 2) 절점1과 절점3의 응력을 비교하여 소삼각형을 통과하는 등응력선數를 계산한다.
- 3) 절점1의 응력에 해당되는 색으로 소삼각형 전체를 polygon fill에 의해 채색한다.
- 4) 소삼각형을 통과하는 등응력선이 내포하는 면적을 해당되는 색으로 polygon fill에 의해 채색한다. 이때 응력이 작은 값으로부터 큰값으로 진행한다.
- 5) Rasterization error로 인하여 결손된 pixel들을 보완한다.
- 6) 1)~5)의 과정을 모든 요소에 대해서 반복하면 구조물 전체에 대한 응력도가 완성된다.

나. 주사선법

주사선법은 그림 1(c)와 같이 display 화면의 주사선(scan line)을 따라서 한 pixel씩 그 위치의 응력을 계산하여 해당되는 색으로 채색하는 point drawing의 방법이다. 개략적인 실행 순서는 다음과 같다.

- 1) 주사선과 요소 경계선의 교차점을 구한다.
- 2) 주사선과 경계선의 교차점에서 응력의 값을 계산한다.
- 3) 주사선을 따라가며 개개의 pixel에 대한 응력을 선형 보간에 의해 구하여, 해당되는 색으로 채색한다.
- 4) 위의 과정을 화면내의 모든 주사선에 대해 실행하면 응력도가 완성된다.

다. 등변수 보간법

이 연구의 등변수보간법은 기존의 방법^(1, 6)을 raster graphics에 맞도록 변형한 방법이다. 그림 1(d)와 같이 절점의 좌표와 절점의 응력을 동일한 보간함수로 보간하여 각각 pixel의 위치와 pixel의 색을 결정하여 point drawing에 의해 채색 하므로서 응력도가 완성된다.

라. 응력도 도화 방법의 비교

이 연구에서는 위에 기술한 응력도를 도화하는 세 방법들을 프로그램으로 이행하였으며, 이들을 비교 검토하여 다음과 같은 결과를 얻었다.

- 1) 표 1과 같이 다각형법과 주사선법 및 등변수보간법은 실행 시간에 있어서 상당한 차이를 보인다. 다각형법 B가 실행 속도에서 가장 유리한 것으로 나타났다.
- 2) 등응력선의 수가 증가함에 따라서 다각형법의 실행 시간은 늘어나는데 반해서 주사선법과 등변수보간법은 거의 영향을 받지 않는다.
- 3) 응력도의 형태는 다각형법과 주사선법은 동일하다. 高次 요소의 경우에는 등변수보간법에 의하여 비교적 완전한 등응력선을 얻을 수 있다. (그림 3)
- 4) 프로그램 이행상의 복잡성이나 난이도의 정도는 등변수 보간법이 가장 간단하고 쉬우며, 나머지 방법들은 모두 비슷하다.
- 5) 이상의 결과를 검토해 보면 마이크로 컴퓨터의 환경하에서 응력도의 도화방법으로는 다각형법이 적합하며, 그 중에서도 다각형법 B가 가장 유리한 것으로 판단된다.

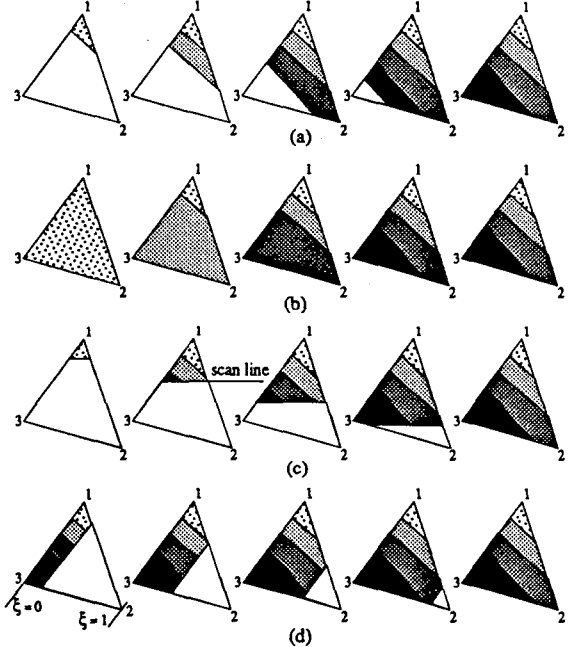


그림 1. 응력도의 도화 방법

- (a) 다각형법 A
(b) 다각형법 B
(c) 주사선법
(d) 등변수보간법

표 1. 응력도 도화 방법의 실행속도 비교 (단위: 초)

등고선수	구 분		다각형법 A	다각형법 B	주사선법	등변수보간법
	무	유				
15	무		26	25	543	1378
	유		64	62	556	1390
30	무		42	40	546	1387
	유		81	78	561	1401

(20 MHz 마이크로 컴퓨터에 의한 194개의 8절점 요소의 응력도 도화시간)

4. 입체구조물의 응력도

셸(shell)이나 3차원고체등의 입체적인 구조물의 응력도는 도화방법은 앞에서 기술한 평면적인 구조물의 경우와 동일한 방법으로 도화한다. 그러나 입체적인 경우에는 3차원적인 좌표변환과 hidden surface의 제거가 필요하며, 입체감을 부여하기 위한 shading의 문제가 수반된다.

가. 3차원적인 좌표변환

마이크로 컴퓨터의 환경하에서는 3차원적인 좌표변환을 위한 firmware의 지원이 없는 것이 일반적이므로 프로그램에 의한 처리가 요구된다. 3차원적인 좌표변환은 회전, 축척 및 이동을 합성한 (4x4)의 변환행렬을 모든 질점의 좌표에 곱해 주므로서 간단하게 실행된다.

나. Hidden surface의 제거

입체구조물의 응력도는 hidden surface가 제거된 형태로 도화되어야 한다. Hidden surface의 제거 방법은 응력도의 도화방법에 따라서 다소 차이가 있다. 이 연구에서는 마이크로 컴퓨터에 hidden surface의 firmware와 depth buffer가 없는 것으로 가정하여 이러한 기능을 이용하지 않는 algorithm을 개발하였다.

다각형법에 있어서 hidden surface는 다음과 같이 처리한다. 먼저 전체 영역을 소삼각형으로 분할하고, 좌표변환된 z방향의 좌표에 따라서 소삼각형을 sorting 한다. 즉 z 방향에서 뒷쪽에 있는 소삼각형에서 부터 순서를 부여한다. 이 순서에 따라서 소삼각형들을 앞쪽에 설명한 방법으로 채색 하면 자동으로 hidden surface가 처리된 응력도가 완성된다.

주사선법의 경우에는 주사선을 따라가며 pixel 단위로 응력도를 채색함과 동시에 hidden surface를 처리한다. 즉 지정된 pixel을 내포하는 요소가 두개 이상일 경우에는 pixel 위치에서 각 요소의 z 좌표를 비교하여 가장 앞쪽에 있는 요소의 응력을 채색하므로써 hidden surface가 처리된다.

등변수보간법에서는 다각형법과 같은 방법으로 hidden surface가 처리된다. 그러나 이 경우에는 소삼각형 대신에 요소들 기준으로 실행된다.

다. Shading

입체적인 구조물에 응력도를 도화하면 입체적인 형태를 식별하기가 어려워진다. 기존의 프로그램^(2,3,4,5,9)에서는 이를 해결하기 위해 그림 4(c)와 같이 응력도 위에 유한요소망(mesh)을 겹쳐 보이는 방법이 많이 쓰이고 있다. 이 방법의 단점은 사실적인 입체감을 얻기가 힘들고, 복잡한 유한요소망으로 인하여 응력도의 일부가 가리어지며, 이로 인해 제한된 해상력을 갖는 마이크로컴퓨터의 display에서는 많은 정보를 상실한다는 점이다. 이를 개선하기 위해 이 연구에서는 그림 4(d)와 같이 陰影(shading)이 부여된 응력도의 도화 방법을 개발하고, 그 효율성을 검토하였다.

유한요소해석의 postprocessing에 있어서는 單一光源을 가정한 simple shading model이 적합하다. 이 model은

$$E = C \{ r \cos i + w(i) \cos^n s \} \quad (1)$$

에 의해서 물체 표면의 밝기를 계산한다. 여기서 i는 입사광선이 표면의 수직선과 이루는 각도이며, s는 반사광선과 관측자의 시선이 이루는 각도이다. w(i)는 반사계수로서 입사각의 함수이다. 지수 n은 표면의 光澤 정도에 따라서 결정되는 값이다. r은 반사비율로서 0과 1사이의 값을 가지며, C는 표면에 도달하는 광원의 조도를 의미한다. 지면의 제한으로 인하여 이들의 산출 방법에 관한 구체적인 설명⁽¹¹⁾은 생략한다.

Shading방법은 응력도의 도화 방법과 근본적으로는 동일하다. 주사선법에 의하면 개개의 pixel마다 대응되는 요소내의 위치를 구하여 표면의 밝기를 계산하고, 해당되는 색으로 채색하므로써 shading이 실행된다. 다각형법과 등변수

보간법에 의한 shading은 질점의 표면 밝기를 구한 다음에 응력도의 도화와 유사한 과정을 거쳐서 실행된다.

라. Shading된 응력도의 도화

주사선법에 의하면 contouring과 shading을 동시에 실행하여 shading된 응력도를 도화할 수 있다. 즉, 주어진 pixel 위치에서 응력과 shading을 위한 음영의 밝기를 계산하여 색을 결정하게 된다. 그러나 다각형법에 의하면 shading과 contouring의 동시 실행이 불가능하다. 이 연구에서는 이를 해결하기 위해 shading과 contouring을 단계적으로 분리시켜서 실행하고, 그림 4(a) 및 (b)와 같이 각각의 과정에서 얻어진 독립적인 畫像을 합성하여 그림 4(d)와 같은 화상을 만드는 방법을 고안하였다. 그 실행과정의 개략적인 흐름도는 그림 2와 같다.

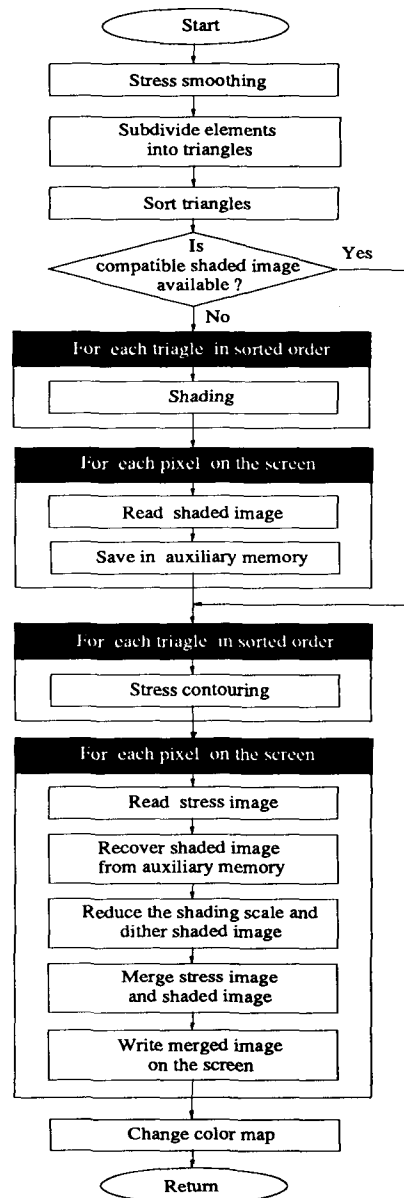
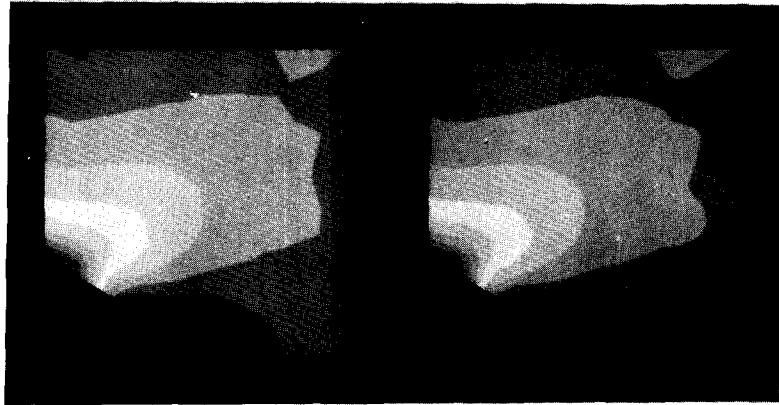


그림 2. Shading된 응력도의 도화 과정

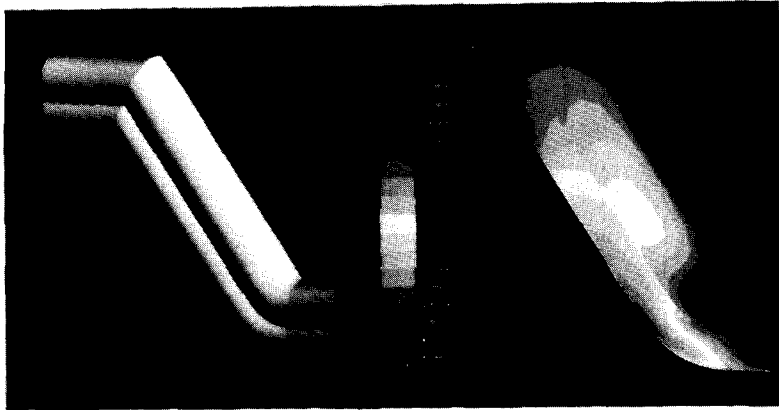


(a)

(b)

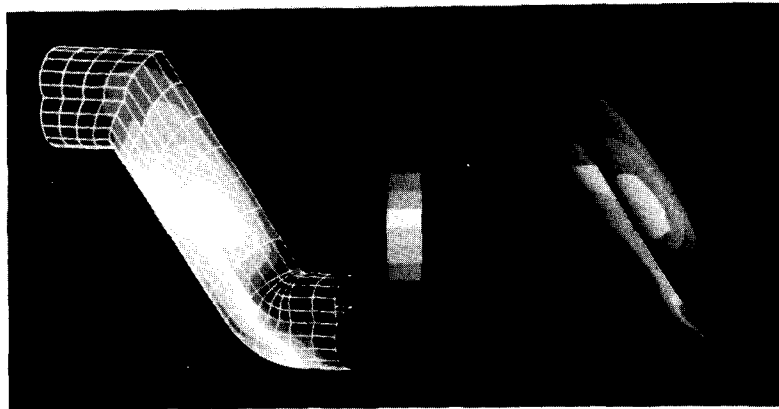
그림 3. 응력도 도화방법의 비교 (6절점 삼각형요소)

(a) 다각형법 또는 주사선법 (b) 등변수 보간법



(a)

(b)



(c)

(d)

그림 4. 입체적 구조물의 응력도

(a) Shading image

(b) 응력도

(c) 응력도에 mesh를 겹쳐서 구조물의 형태를 표시하는 예

(d) Shading image (a)와 응력도 (b)를 합성한 image (본연구)

Shading에 의하여 명암을 자연스럽게 나타내기 위해서는 최소한 60개 이상의 gray scale이 필요하다. 입체구조물의 응력도에 입체감을 부여하기 위해서는 shading과 stress contouring이 동시에 이루어져야 하므로 응력도를 이루고 있는 띠 하나하나의 색이 모두 60개 이상의 단계로 구분되어야 한다. 따라서 띠의 갯수가 15개일 경우에는 최소한 900개의 색을 동시에 나타낼 수 있어야 하며, 띠의 갯수가 30개일 경우에는 1800개의 색이 필요하다. 반면에 마이크로컴퓨터용 graphic display의 bit plane수는 제한되어 있다. 예를 들어서 8 bit plane의 graphic display가 동시에 나타낼 수 있는 색의 수는 256개이다. 이 연구에서는 dithering 기법을 이용하여 제한된 색(bit plane 수)의 문제를 해결하였다. 그림 5는 dithering에 의해 shading이 된 응력도를 확대한 사진이다.

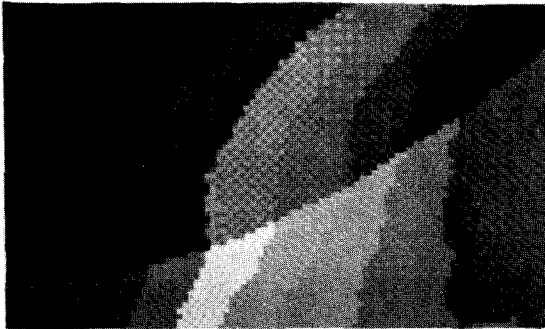


그림 5. Dithering을 이용하여 shading된 응력도 (부분적인 확대)

5. 주응력의 크기와 방향의 도화

주응력의 크기는 앞에서 기술한 응력도에 의해서 나타낸다. 주응력의 방향은 그림 6과 같이 짧은 선분에 의해서 표시하는 것이 일반적이다. 입체적인 구조물의 주응력 방향은 도형의 좌표변환에 따라서, 이를 화면좌표로 전환하여 표시한다. 예를 들어서 최대주응력 σ_1 이 x, y 및 z 축과 이루는 각도를 각각 α_1, β_1 및 γ_1 라고하고, 실제좌표를 화면좌표로 변환하는 (4×4) 의 변환행렬을 T 라고 하면

$$\cos \phi_1 = T_{11} \cos \alpha_1 + T_{21} \cos \beta_1 + T_{31} \cos \gamma_1 \quad (2)$$

에 의해서 σ_1 이 화면의 수평좌표축과 이루는 각도 ϕ_1 을 구하여 표시한다. 같은 방법으로 σ_2 와 σ_3 의 방향을 화면좌표로 전환하여 표시한다.

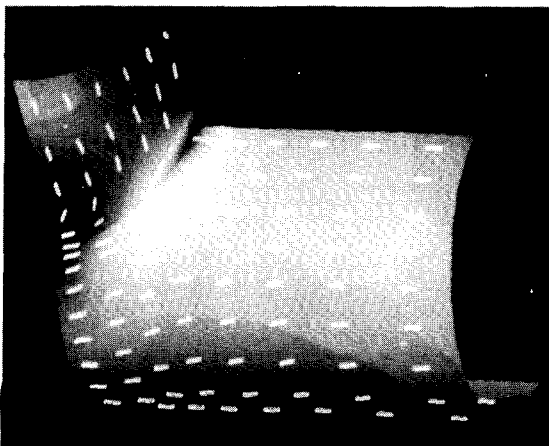


그림 6. 주응력 σ_1 의 크기 및 방향 표시

6. 골조구조물 해석의 Postprocessing

Truss와 frame의 골조구조물 해석에서는 일반적으로 응력이 아니라 內力의 분포를 필요로 한다. 그러므로 전체 평형 방정식을 풀어서 절점의 변위, 반력 및 부재의 내력을 구한 다음에는 이를 바탕으로 전단력선도나 굽힘모멘트선도와 같은 도형으로 나타낸다. 따라서 앞절에서 설명한 연속체와는 다른 내용과 방법의 postprocessing이 필요하다.

가. 내력도의 도화

이 연구에서는 부재위에 등분포하중, 직선분포하중 및 일련의 집중하중과 모멘트가 작용하는 것으로 가정하여 내력분포를 이들 하중과 절점력의 함수로 만든 일반형 공식을 유도하였다. Preprocessing과정에서 하중조건은 임의의 순서로 부여된다. Postprocessing과정에서는 요소 위에 작용하는 하중을 그 위치에 따라서 좌측에서부터 우측으로 정렬한 다음에 이를 일반형 공식에 대입하여, 한쪽 방향으로 진행하면서 순차적으로 내력을 계산한다. 내력은 분포하중이 작용하는 구간에서는 일정하게 짧은 간격으로 나누어서 계산하고, 그렇지 않은 구간에서는 집중하중이 작용하는 위치에서만 계산한다. 계산된 내력은 즉시 disk file에 기록하거나 diagram의 형태로 전환하여 그림7과 같이 graphic display와 plotter에 도화한다. 이는 여러 위치에서 계산된 내력을 일일이 RAM에 기억시키려면 컴퓨터의 막대한 기억 용량이 소요되기 때문이다. 또한 graphic display에는 서로 대응되는 내력(예를 들어서 전단력 S_x 와 굽힘모멘트 M_x)의 도형을 짝지어서 동시에 도화하므로서 계산량을 줄일 수 있다. 이때 이들 도형은 각기 다른 bit plane에 그려서, 독립된 도형이나 또는 중복된 도형을 선택적으로 도화할 수 있도록 한다. 수치는 제3의 bit plane에 표시하는 것이 바람직하다. Graphic display에서와 동일한 algorithm에 의해서 plotter에 도화한다. 그러나 이 경우에는 bit plane 대신에 여러개의 pen을 이용하여 도화한다.

내력도의 축척은 프로그램이 자동으로 정해 주고, 필요에 따라서 인위적으로 변경시킬 수 있어야 한다. 적당한 축척을 결정하기 위해서는 내력의 최대 및 최소값이 필요하다. 위에서 언급했듯이 내력은 계산과 동시에 도화되며 그 값들이 기억되지 않는다. 그러므로 먼저 구조물 전체의 내력을 계산하여 그 최대, 최소값을 구하고, 이에 의거하여 축척을 결정한 다음에 재차 내력을 구하여 도화한다.

축척을 결정하기 위해서 뿐만 아니라 내력의 전체적인 경향을 나타내는데 있어서 중요한 것은 내력의 최대 및 최소값과 그 발생 위치이다. 이는 하중과 내력, 또는 내력과 내력 사이의 관계를 이용하여 산출한다. 예를 들어서 굽힘모멘트의 최대 또는 최소값의 위치는 대응되는 전단력이 0인 위치를 수치적인 방법에 의하여 계산하여 구한다.

나. 구조물 변형의 도화

Truss부재의 변위는 절점의 변위만으로 완전히 정의되지만 frame의 경우에는 요소내의 변위형태는 곡선적인 탄성선을 이루게 된다. 요소의 탄성선은 절점의 변위와 변위각으로 인한 것과 양단고정의 요소에 작용하는 일련의 하중, 즉 등분포하중, 직선분포하중 및 집중하중으로 인한 탄성선을 중첩하여 구한다. 즉 총변위는

$$\Delta = \Delta_0 + \mathbf{R}^T \left\{ \Delta_R + \Delta_U + \Delta_D + \sum_{i=1}^N \Delta_i \right\} \quad (3)$$

에 의해서 구한다. 여기서 Δ_0 는 절점의 변위를 직선적으로 연결하여 얻은 탄성선을 표시하며, Δ_R, Δ_U 및 Δ_D 는 각각 절점의 변위각만으로 인한 변위, 등분포하중으로 인한 변위 및 직선분포하중으로 인한 변위이며, Δ_i 는 i 번째 집중하중으로 인한 변위이고, n 은 요소위에 작용하는 집중하중의 수이다. \mathbf{R} 은 요소의 국부좌표계를 전체좌표계로 변환하는 회전행렬이다. 절점의 변위는 전체좌표계를 기준으로 한 값인 반면에 탄성선은 요소의 국부좌표계를 기준으로 산출된다. 그러므로

요소의 탄성곡선을 구한 다음에 이를 전체좌표계로 좌표변환을 하여 절점의 변위와 중첩시키므로서 전체좌표계를 기준으로 한 요소내의 임의점의 변위가 산출된다. 그림 8과 같은 구조물의 변형된 형태는 구조물의 원형에 변위를 가하므로서 얻을 수 있다. Truss의 경우에는 구조물의 절점 좌표에 변위값을 가하여 변형 후의 절점 좌표를 구한 다음에 절점들을 서로 연결해 주면 변형된 형태가 도화된다. Frame의 경우에는 요소를 미소한 간격으로 분할하여, 이들 분할점에서 구조물 원형의 좌표와 과장된 축척의 변위를 합산하여 변형 후의 좌표를 계산한 다음에 이들 분할점을 연결해주면 변형후의 형태가 완성된다.

다. 구조물 변형의 animation

마이크로 컴퓨터의 환경하에서는 비록 변형에 관한 모든 데이터가 미리 계산되어 컴퓨터에 기억되어 있는 상태라고 하더라도 이를 display하는 속도가 느리므로 사실적인 animation을 달성하기가 어렵다. 그러나 graphic display의 각 bit plane에 독립된 화상을 저장한 후에 순차적으로 masking 하면 animation의 효과를 얻을 수 있다.

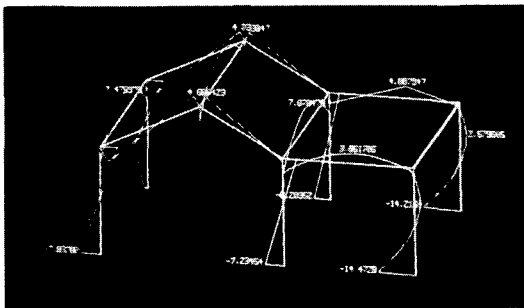


그림 7. 3차원 frame의 굽힘모멘트선도

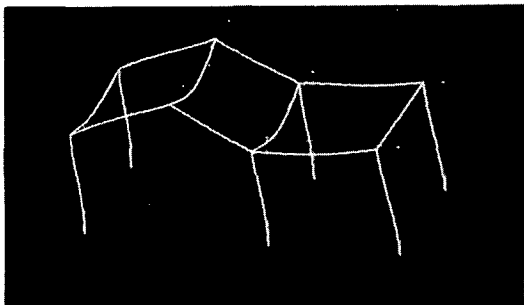


그림 8. 3차원 frame의 탄성선

7. 결론

이 연구에서는 마이크로 컴퓨터의 제한성으로 인하여 유한 요소해석 postprocessing에 있어서 발생하는 문제점을 검토하고 그 해결방안을 모색하였다. 구체적으로는 응력도의 도화 방법을 개발하고, 이들을 프로그램으로 이행하여, 마이크로 컴퓨터의 환경에 적합한 방법을 도출하였다. 특히 입체적인 구조물의 shading된 응력도를 도화하는 새로운 방법을 개발하였다. 골조구조해석의 효율적인 postprocessing 방법을 제시하였다.

마이크로 컴퓨터를 이용한 postprocessing 방법의 개발과 이의 프로그래밍을 통해서 얻은 결론은 다음과 같다.

- 1) 마이크로 컴퓨터를 이용한 유한요소해석 postprocessing은 마이크로 컴퓨터의 특성과 제한성을 고려한 algorithm에 바탕을 둔 프로그램에 의해서 효율적으로 실행될 수 있다.
- 2) 마이크로 컴퓨터의 다양한 graphics 성능을 최대한 활용하면 기존의 프로그램들 보다는 고급의 postprocessing

을 달성할 수 있다.

3) 마이크로 컴퓨터를 이용한 postprocessing에 있어서 응력도의 도화 방법으로는 주사선법이나 등변수보간법과 같은 point drawing의 방법보다는 다각형법과 같은 area fill의 방법이 더 적합하며, 이 연구에서 비교한 방법중에서는 다각형법 B가 가장 유리한 것으로 판단된다.

4) 입체구조물의 postprocessing을 위하여 이 연구에서 고안한 응력도와 shading image의 합성 방법을 이용하므로써 기존의 방법보다 발전된 형태의 응력도를 얻을 수 있다.

5) 이 연구에서 제안하는 전단력선도, 굽힘모멘트선도, 탄성선도의 도화 방법에 의하여 골조구조해석의 postprocessing을 효과적으로 달성할 수 있다.

유한요소법이 널리 활용되고, 이를 위한 상업적 소프트웨어들이 경쟁적으로 개발, 보급됨에 따라서 preprocessing과 postprocessing이 차지하는 비중이 커지고 있다. 다른 한편으로는 소형시스템에 의한 유한요소해석이 점차 보편화되는 추세이다. 이러한 관점에서 마이크로 컴퓨터를 이용한 유한요소해석의 preprocessing과 postprocessing의 중요성은 증대될 것으로 전망된다. 앞으로 이와 관련된 연구가 깊이 있고 활발하게 진행되어야 할 것이다.

참고 문헌

1. Akin, J. E., and W. H. Gray, Contouring on isoparametric surfaces, *International Journal for Numerical Methods in Engineering*, Vol. 11, pp1893-1897, 1977.
2. Algor Interactive System, Inc., *Algor Supersap, A Comprehensive Finite Element Analysis System*, Algor Interactive Systems, Inc., Pennsylvania, 1987.
3. Baran, N. M., ANSYS-PC/Linear and MSC/Pal2, *Byte*, pp205-212, Nov., 1987.
4. Baran, N. M., *Finite Element Analysis on Microcomputers*, McGraw-Hill, New York, 1987.
5. Fredricksson, B., and J. Mackerles, Partial list of major finite element programs and description of some of their capabilities, *State-of-the-art Survey on Finite Element Technology*, A.S.M.E., pp363-402, 1983.
6. Gray, W. H., and J. E. Akins, An improved method for contouring on isoparametric surfaces, *International Journal for Numerical Methods in Engineering*, Vol. 14, pp451-472, 1979.
7. Griffen, O. H., and C. R. Wilson, Finite element analysis on a microprocessor-based personal workstation, Technical note, *Computers and Structures*, Vol. 17, pp617-619, 1983.
8. Hamilton, C. H., and R. S. Gallagher, PATRAN : The computational laboratory, pp577-594, *Finite Element Systems*, 3rd rev., Springer-Verlag, New York, 1985.
9. Kamel, H. A., and M. W. Macabe, GIFTS: Graphic Oriented Interactive Finite Element Time Sharing System, *Structural Mechanics Software Series*, University Press of Virginia, Charlottesville, pp 145-226, 1977.
10. Lee, J. Y., and J. R. Cooke, *A Pre- and Postprocessing for Finite Element Analysis on IBM PC and Professional Graphic Display*, presented at IBM PC Software Conference, Alexandria, Virginia, 1985.
11. Newman, M. W., and R. F. Sproull, *Principle of Interactive Computer Graphics*, 2nd ed., McGraw-Hill, New York, 1979.
12. Rouse, N. E., FEA for the masses, *Machine Design*, pp62-68, July, 1985.
13. Schulman, M. A., *The interactive display of parameters on two- and three-dimensional surfaces*, M.S. Theses, Cornell University, Ithaca, New York, 1981.