



셀룰러 오토마타와 객체기반 시뮬레이션 환경에 의한 트러스 구조의 해석모델 개발

Development of a Truss Structure Analysis Model based on Cellular Automata and Object-oriented Simulation Environment

김태곤* · 이정재** · 서교***,†

Taegon Kim · Jeongjae Lee · Kyo Suh

ABSTRACT

The aim of this study is to develop a simulation model for analyzing 2D truss structure using Generic Agricultural System Simulator (GASS). Although the truss is simple structure, numerical methods based on matrix analysis are cumbersome and complicated. This study suggests simple and convenient methods to remove calculating steps for whole stiffness matrices. The simulation environment based on independency of object-oriented components on GASS consists of component development and component deploy stages. A component for a truss structure is implemented based on equilibrium equations at nodes. The simulator can analyze truss structures through deploying components with attributes and links. The examples using GASS show intuitive graphical results of the movements of truss nodes.

Keywords: Component based modelling; GASS; simulation; structure analysis; truss

1. 서 론

농업용 철근콘크리트 구조물의 설계에 있어서 적절한 위치에 철근을 배근하고, 콘크리트 단면의 형상을 최소화하기 위해서는 외부하중에 대한 내부 응력 흐름을 분석해야 한다 (Yoon and Lee, 2002). 일반적으로 스트럿-타이 모델이 가장 널리 이용되나, 해석과정에서 연구자의 주관적 견해에 따라 결과가 상이할 수 있다는 한계를 갖는다 (Kim et al., 2004). 이를 개선하기 위하여 연속체 형태의 구조물을 이산체인 트러스 형태로 치환하여 해석하는 연구가 이루어졌으며(Lee et al., 2009), 이러한 환경에서 구조문제는 비교적 단순하고, 비슷한 형태의 풀이과정을 반복적으로 거치게 된다. 하지만 기존의 구조 해석 방법은 문제를 해석할 때마다 강성행렬을 새롭게 생성하여 풀이함으로써 (Kim et al., 2011), 계산의 효율성이 떨어지고, 해석 과정이 복잡한 한계를 가진다.

Lee et al. (2012)는 이러한 원인에 대하여 절차지향적 프로그래밍 방식의 한계와 독립성이 보장되지 않는 객체지향적 프로그래밍 방식에 기인한다고 주장하였다. 독립적으로 계산 가능한 구조요소와 반복계산방법을 개발하여, 구조요소를 연결하는 방법으로 전체 강성행렬을 만들지 않고 보 구조물을 해석하였다. 특히 요소 독립성이 보장되도록 객체지향적으로 구현함으로써, 구조 해석 도중에 구조물의 요소가 결합되더라도 효과적으로 문제를 해결할 수 있음을 검증하였다. 이는 구조물에 대한 기존 해석 결과가 존재하는 경우, 유사한 구조물에 대하여 효과적인 해석이 가능함을 의미한다.

이러한 객체지향적 모델의 구현을 용이하게 할 목적으로 Yi (2003)는 컴포넌트 기반의 자기조직화가 가능한 시뮬레이터(Generic Agricultural Systems Simulator, GASS)를 개발하였고, 개발된 시뮬레이터를 통해 복잡해 보이는 현상을 여러 개의 단순한 컴포넌트의 조합으로 모의할 수 있음을 관계시스템의 모의와 배수 갑문의 응력해석을 통하여 검증하였다. 이후, 개발된 시뮬레이터는 유통 모의(Suh et al., 2005), 관개에 따른 논 담수심 모의(Kim and Lee, 2007), 2차원 열전도 모의(Kim, 2007)와 같은 문제들을 해결하는데 적용되면서 다양한 분야에 대한 활용가능성을 보여주었다.

본 연구에서는 트러스 구조 해석을 위한 트러스의 절점으로 연결된 객체기반 구조해석모델을 개발하였다. 셀룰러 오토마타와 객체기반 시뮬레이션 환경인 GASS를 이용하여 독립된 절점이 자기주도적으로 변위를 결정하여, 전체 구조물의 평형방정식을

* 서울대학교 생태조경·지역시스템공학부
 ** 서울대학교 조경·지역시스템공학부, 서울대학교 농업생명과학연구원
 *** 서울대학교 조경·지역시스템공학부, 서울대학교 농업생명과학연구원, 서울대학교 그린바이오과학기술연구원
 † Corresponding author Tel.: +82-2-880-4715
 Fax: +82-2-873-2087
 E-mail: kyosuh@snu.ac.kr

2013년 12월 2일 투고
 2014년 2월 25일 심사완료
 2014년 3월 24일 게재확정

만족시키는 방법으로 트러스 해석모델을 개발하였다. 기존 GASS (Yi, 2003; Kim, 2007)에서의 연결방법 및 컴포넌트 구현 방식을 개선하고, 새로운 컴포넌트 개발과정 및 컴포넌트를 이용한 시뮬레이션 설정과정을 제시하였다. 제시된 방법으로 모의된 결과는 기존 프로그램을 이용한 결과와 비교를 통하여 검증하였다.

II. 이론적 배경

1. 셀룰러 오토마타를 이용한 트러스 구조해석

트러스는 힌지로 연결된 봉 부재로, 부재 방향으로 1차원 내력만 발생하며, 양쪽 끝단에서 모멘트에 저항하지 않고, 변위만 발생한다. 매트릭스 구조 해석 방법에서는 2차원 트러스를 해석하기 위하여 트러스를 하나의 요소로 설정하여 4×4 강성행렬을 만들어 평행방정식을 풀이한다. 본 연구에서는 자기주도적으로 변위를 찾아가는 구조해석 컴포넌트를 구성하기 위하여 구조물을 구성하는 요소들을 작은 요소들로 나누고, 이들 요소에 규칙을 정의하는 셀룰러 오토마타 개념을 차용하여, 절점을 중심으로 평행방정식을 정식화하였다. 단위 요소를 절점으로 선택한 이유는 절점 중심으로 규칙을 정의하면, 절점의 x, y축 변위를 속성으로 갖는 반면, 트러스 부재 혹은 더 많은 요소를 단위 부재로 정의하면 (Choi, 2012), 단위 부재가 갖는 속성정보가 많아지고, 이에 따라 셀룰러 오토마타가 더 복잡해지기 때문이다.

기본적인 구조물의 형태가 Fig. 1과 같이 절점들이 결합되어 구성된다고 가정(Gürdal and Brian, 2000)하면, 각 절점은 여러 개의 트러스와 연결된다. 따라서 각 절점에서 평행방정식은 식 (1), (2)와 같다.

$$\sum_{k=1}^n F_{xk} + f_x = 0$$

$$\sum_{k=1}^n F_{yk} + f_y = 0$$

여기서, n 은 절점에 연결된 트러스의 개수, F_{xk} , F_{yk} 는 k 번째

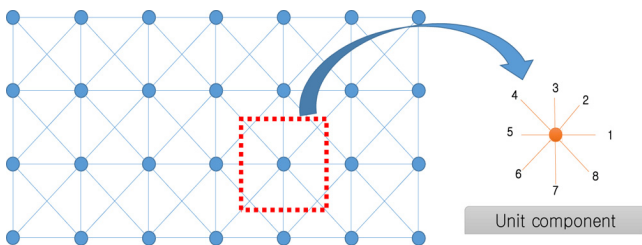


Fig. 1 A basic form of 2D truss structure and the modelling unit of truss node

트러스의 x, y 방향 내력, f_x , f_y 는 각 절점에 작용하는 외력을 의미한다. 절점에서 각 트러스 방향을 나타내는 방향 벡터($\cos \theta_k$, $\sin \theta_k$)를 고려하면, 각 트러스에 작용하는 x, y 방향 내력은 식 (3), (4)와 같이 트러스에 작용하는 내력인 인장력 혹은 압축력으로 정리할 수 있다.

$$F_{xk} = F_k \cos \theta_k$$

$$F_{yk} = F_k \sin \theta_k$$

각 트러스에 작용하는 내력은 탄성방정식에 따라 식 (5)와 같이 정리할 수 있다.

$$F_k = EA_k \epsilon_k$$

여기서, E 는 탄성계수이고, A_k 는 k 번째 트러스의 단면적을 의미한다. 또한 ϵ 은 변형률을 의미하는데, 트러스는 길이방향으로 변형되고, 미소 변형의 원리를 고려할 때, 식 (6)과 같이 정리할 수 있다.

$$\epsilon_k = \frac{\Delta l_k - l_k}{l_k}$$

여기서, ΔL_k 는 트러스의 변형량으로 절점의 변위로 구할 수 있으며, l_k 는 트러스의 길이로 미소변형의 원리에 따라 초기 절점 위치에 따라 결정된다. 이를 정리하면, 식 (7)과 같이 변형률을 절점의 변위로 기술할 수 있다.

$$\epsilon_k = \frac{(u_k - u_n) \cos \theta_k + (v_k - v_n) \sin \theta_k}{\sqrt{(x_k - x_n)^2 + (y_k - y_n)^2}}$$

여기서, u_n , v_n 은 기준 절점에서의 x, y방향의 변위를 의미하며, u_k , v_k 는 기준 절점에서 k 번째 트러스로 연결된 반대쪽 절점에서의 x, y방향 변위를 의미한다. x_n , y_n 은 기준 절점의 위치 좌표를 의미하며, x_k , y_k 는 각각 k 번째 트러스로 연결된 반대쪽 절점의 좌표를 의미한다.

2. 시스템 시뮬레이션 환경

GASS는 다양한 시뮬레이션 모델들의 모듈성 향상을 통하여 궁극적으로 복잡도를 해소하고, 자기 조직화를 모의함으로써, 복잡계인 농업 문제를 해석하도록 개발되었다 (Yi, 2003). 이를 위해 데이터와 메소드를 분리하고, 컴포넌트 간에 정보 전달을 일방향으로 제한하는 방식을 통하여, 모델 결합으로 발생하는 복잡도를 낮추었다. 또한 단위 시간을 중심으로 동시성(concurrent)

을 보장하여, 개별 컴포넌트는 비동기적으로 기능을 수행하되 전체를 통괄하는 시스템 시간과 동기화함으로써 전체 시스템의 모의가 가능하도록 설계하였다.

GASS를 이용한 분석문제에서의 가장 두드러지는 특징은 개별 컴포넌트가 전체 시스템을 알지 못한 상태에서 단위 요소로서 주어진 임무만을 정의한 후, 인접한 컴포넌트를 인지하여 전체 시스템을 모의한다는 점이다. 모델을 만들 때는 전체 시스템과 무관하게 단위 모델만을 개발하면 되고, 모델을 사용할 때는 개발된 모델을 연결하기만 하면 주어진 도메인에 대하여 문제를 모의할 수 있다. 이러한 특징은 관계 시스템 모의, 열전도 모의, 농산물 유통 모의와 같은 다양한 분야의 문제해결에 적용되었다 (Yi, 2003; Suh et al., 2005; Kim, 2007).

GASS 시뮬레이션 환경에서 모델을 개발하는 데 있어서 가장 큰 특징 중 하나가 일방향 정보전달기법이다. 분석 대상 모델이 되는 단위 컴포넌트는 매 단위시간마다 인접한 컴포넌트의 값을 이용하여 자신의 컴포넌트 속성을 업데이트하며, 이 때 인접한 컴포넌트에 인위적으로 값을 주입할 수 없다. 이는 방위적 프로그래밍 개념 (Goodliffe, 2007)과 유사한데, 인접한 컴포넌트에서 원하는 값만 가져오고, 반대로 인접한 컴포넌트에 예상할 수 없는 값을 주지 않을 수 있게 된다.

III. 트러스 구조해석 컴포넌트의 개발

1. 컴포넌트 모델의 구성과 GASS 환경의 개선

기존의 GASS 환경은 인접한 컴포넌트를 정의하기 위하여 기하학적 거리를 이용하였다. 하지만 Kim et al. (2007)이 정보전달체계를 정보계와 물질계로 분리하면서, 기하학적 인접성의 한계를 지적한 바 있다. 본 연구에서는 인접성을 기하학적 관계에서 위상학적으로 관계로 변경하고, 위상학적 관계를 정의한 인접리스트에 따라 인접성을 파악하도록 GASS 환경을 개선하였다. 또한 인접행렬로 구성된 네트워크에서 각 컴포넌트별로 인접 컴포넌트를 찾을 수 있도록 너비우선탐색 (Breadth First Search, BFS) 알고리즘을 이용하여 네트워크를 신장 트리로 변환하였다.

시뮬레이션을 관장하는 GASSEnvironment 클래스에서는 Fig. 2에서 도시한 바와 같이, IGASSComponent 인터페이스를 상속한 인스턴스를 소유하고, simulation() 메소드가 호출되면, 단위시간 (tick)을 증가시키면서 각 컴포넌트가 preProcess(), process(), postProcess() 메소드를 수행되도록 관장한다. 특히 process() 메소드는 앞서 각 컴포넌트 별로 산정한 신장트리를 이용하여 인접한 컴포넌트를 전달함으로써, 컴포넌트 간에 정보전달이 가능토록 설계하였다.

컴포넌트의 프로토콜 역할을 수행하는 IGASSComponent 인터

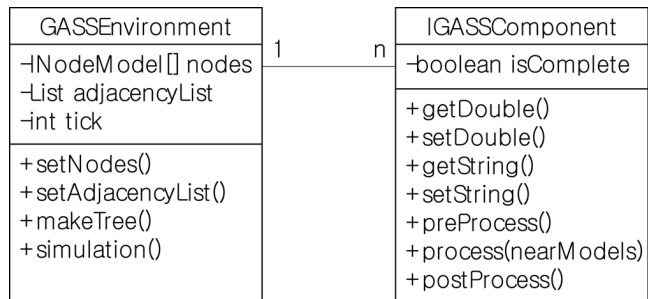


Fig. 2 Class diagram of main class and component interface

페이스는 각 컴포넌트의 종료여부를 판단할 수 있는 isComplete 라는 필드를 가지고 있으며, 실수형과 문자형 정보를 저장, 열람할 수 있도록 접근자를 가지고 있다. 연구자는 IGASSComponent를 구현한 DefaultGASSComponent를 상속함으로써, 모델 구현 과정을 간소화할 수 있다. 주로 process() 메소드만 구현하면 원하는 모델을 구현할 수 있도록 설계하였다.

2. 트러스절점 컴포넌트의 구현

앞서 트러스 구조해석을 위하여 절점을 분석 단위로 설정하고, 절점이 8개의 트러스를 갖는 구조로 지배방정식을 정식화하였다. 이에 따라 트러스절점 컴포넌트는 Fig. 3과 같이 구현하였다. 만약 트러스절점이 경계조건으로 고정단인 경우에는 변위가 발생할 수 없으므로, 계산과정을 생략하고, 처음 조건을 그대로 유지하게 된다. 따라서 3번째 라인처럼 절점컴포넌트의 속성 중 “fixed”라고 고정단 표시가 된 경우에는 process() 메소드에서 아무런 처리를 하지 않는다.

트러스절점 컴포넌트가 일반 절점인 경우에는 주변 트러스 8개의 합력과 해당 절점의 변위를 산정해야 한다. 앞서 식 (1), (2)와 같이 2개의 방정식을 만족하는 변위 값 u , v 를 계산해야 하므로 미지수가 2개인 1차 연립방정식이 된다. 이는 2×2 행렬과 2×1 행렬의 곱으로 계산되므로, 2×2 행렬을 저장하기 위한 배열 uv 변수와 2×1 행렬을 저장하기 위한 $sumB$ 변수를 코드 4째 줄부터 10째 줄에서 정의하고, 초기화하였다.

코드에서 11번째 줄부터는 인접 컴포넌트의 값을 이용하여 행렬 값을 채우는 과정을 서술하였다. 이렇게 채워진 값은 postProcess()에서 역행렬을 구함으로써, 해당 절점에서의 변위 u , v 를 산정할 수 있다.

3. 트러스 구조 컴포넌트의 검증

구현한 트러스절점 컴포넌트를 이용하여 트러스를 해석할 수 있는지 검증하기 위하여, Gürdal and Brian(2000)이 제시한 문제를 적용하였다. 컴포넌트의 속성으로 절점의 좌표, 트러스 단

```

01: @Override
02: public void process(List<TreeNode> neighborhoods) {
03:     if (getString("fixed") != null) { return; }
04:     sumB[0] = getDouble("fx");
05:     sumB[1] = getDouble("fy");
06:     for (int i = 0; i < 2; i++) {
07:         for (int j = 0; j < 2; j++) {
08:             uv[i][j] = 0;
09:         }
10:     }
11:     for (TreeNode n : neighborhoods) {
12:         IGASSComponent nm = n.getModel();
13:         double x2 = nm.getDouble("x");
14:         double y2 = nm.getDouble("y");
15:         double x1 = nm.getDouble("x");
16:         double y1 = nm.getDouble("y");
17:         double A = getDouble("A_" + nm.getId());
18:         double E = 10000.;
19:         double u = nm.getDouble("u");
20:         double v = nm.getDouble("v");
21:         double L = Math.sqrt(Math.pow(x2 - x1, 2) + Math.pow(y2 - y1, 2));
22:         double cos = (x2 - x1) / L;
23:         double sin = (y2 - y1) / L;
24:         sumB[0] += (E*A*cos*u/L + E*A*sin*v/L) * cos;
25:         sumB[1] += (E*A*cos*u/L + E*A*sin*v/L) * sin;
26:         uv[0][0] += E*A*cos/L*cos;
27:         uv[0][1] += E*A*sin/L*cos;
28:         uv[1][0] += E*A*cos/L*sin;
29:         uv[1][1] += E*A*sin/L*sin;
30:     }
31: }
    
```

Fig. 3 A main program code for node component

면적 하중, 경계조건 탄성계수를 설정하였다. 탄성계수는 $10,000 \text{ kip/in}^2$ 으로 가정하여 계산한 결과, 2번 절점에서 $u = -7.366855$, $v = -18.021140$ 와 3번 절점에서 변위가 $u = -9.522367$, $v = -39.395724$ 로서, Gürdal and Brian(2000)이 제시한 결과인 $\{u_2, v_2\} = \{-7.36686, -18.0212\}$ 와 $\{u_3, v_3\} = \{-9.52237, -39.3957\}$ 과 비교할 때 유효숫자 내에서 정확히 일치한 결과를 보여주었다. 또한 반복횟수는 Mathematica®를 이용하여 703번 반복한 반면, 본 연구에서는 357번 반복계산을 통하여 결과를 산출하였다.

IV. 결과 및 고찰

1. 5-트러스 구조의 해석

개발한 트러스 구조 컴포넌트의 적용 예를 살펴보기 위하여, 모델개발 과정에 가정한 8방향 형태의 트러스와는 다른 형태인 삼각형 트러스 구조를 풀이하였다. 결과의 검증을 위하여 프로그램으로 해석한 결과가 제공되는 문제를 선정하였으며, 문제의 형상은 Fig. 5와 같이 도시할 수 있다. 트러스와 절점의 속성은 Table 1과 같이 정리할 수 있으며, 해석 결과는 웹을 통해 Matlab®

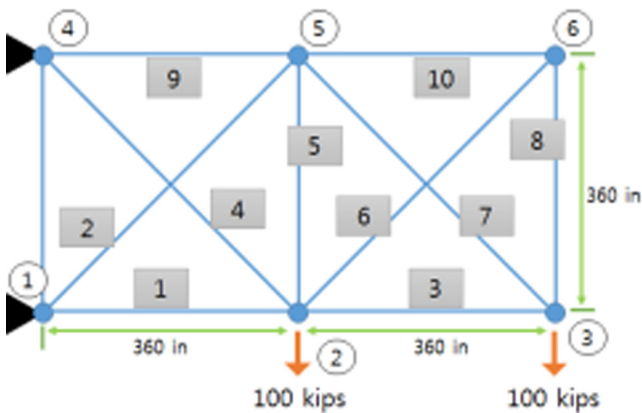
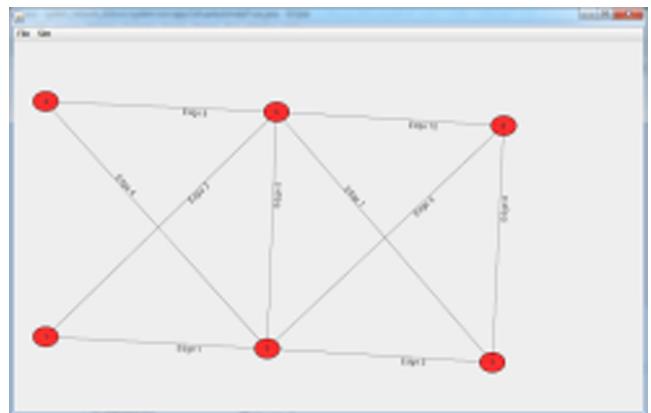


Fig. 4 Ten bar structure and structural analysis (This study)



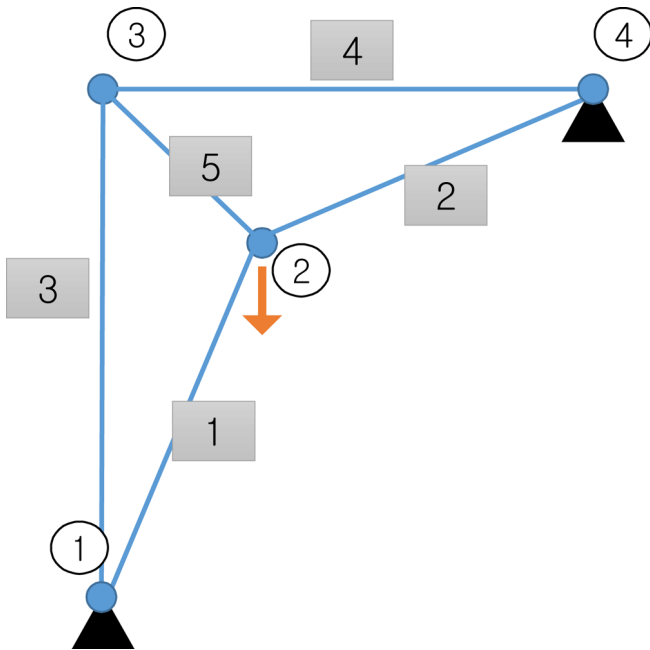


Fig. 5 Problem set for 5-truss structure

Table 1 Properties of nodes and edges for 5-truss structure

Node ID	x (cm)	y (cm)	Element ID	E (GPa)	A (cm ²)
1	0	0	1	200	40
2	150	350	2	200	40
3	0	500	3	200	30
4	500	500	4	200	30
			5	70	20

코드를 제공하고 있다 (Bhatti, 2005).

절점 2에 작용하는 하중은 150kN으로 문제를 풀이하였을 때, 고정단을 제외한 절점 2와 절점 3에서의 변위는 각각 (0.5390, -0.9531), (0.2647, -0.2647)로 계산되었다.

본 연구에서 개선한 GASS 환경에서 문제를 해결하기 위하여, Fig. 6 과 같이 문제를 코딩하였다. 시뮬레이션 환경인 GASSEnvironment 인스턴스에 구동하기 위한 컴포넌트 객체를 생성하여 알려주면, GASSEnvironment가 각 컴포넌트 객체를 주어진 정보에 따라 연결하여 시뮬레이션을 수행한다. 코드에서 살펴보면, 2-6라인에서 앞서 설계한 트러스 구조 컴포넌트를 생성하고, 7-21라인에서 트러스 절점의 속서인 좌표, 고정단 여부, 그래프 표시 여

```

01: public void problem5Truss(GASSEnvironment app) {
02:     int nodeCount = 4;
03:     IGASSComponent[] nodes = new IGASSComponent[nodeCount];
04:     for (int i = 0; i < nodeCount; i++) {
05:         nodes[i] = new GASSTrussComponent(i+1);
06:     }
07:     nodes[0].setDouble("x", 0);
08:     nodes[0].setDouble("y", 0);
09:     nodes[0].setString("fixed", "fixed");
10:     nodes[1].setDouble("x", 150);
11:     nodes[1].setDouble("y", 350);
12:     nodes[1].setDouble("fy", -150);
13:     nodes[1].setString("plot_item", "v");
14:     (...)
22:     app.setNode(nodes);
23:
24:     int[][] adjacencyMatrix = new int[][]{{
25:         {-1, 0, 2, -1}, {-1, -1, 4, 1}, {-1, -1, -1, 3}, {-1, -1, -1, -1}
26:     }};
27:     for (int i = 0; i < adjacencyMatrix.length; i++) {
28:         for (int j = 0; j < adjacencyMatrix[i].length; j++) {
29:             if (adjacencyMatrix[i][j] >= 0 &&adjacencyMatrix[i][j] < 2) {
30:                 nodes[i].setDouble("A_"+nodes[j].getId(), 40);
31:                 nodes[j].setDouble("A_"+nodes[i].getId(), 40);
32:                 nodes[i].setDouble("E_"+nodes[j].getId(), 200);
33:                 nodes[j].setDouble("E_"+nodes[i].getId(), 200);
34:             } else if (adjacencyMatrix[i][j] >= 2 &&adjacencyMatrix[i][j] < 4) {
35:                 (...)
39:             }
40:         }
41:     }
42:     (...)
44: }
45: }
46: app.setAdjacencyMatrix(adjacencyMatrix);
47: }
    
```

Fig. 6 A main program to set up attributes of truss

부 등을 정의한다. 22라인에서 GASSEnvironment 인스턴스에 컴포넌트 정보를 넘겨주면, 시스템을 구성하는 요소에 대한 정의는 끝난다. 이후 코드는 요소들 간의 관계를 정의하기 위한 코드로서, 24-26라인은 인접행렬을 배열로 정의하였다. 27-45 라인까지는 연결 속성, 즉 트러스 속성을 의미하는 단면적, 탄성계수를 정의하고 있다. 이렇게 연결속성을 모두 정의하고 나면, 컴포넌트 정보와 마찬가지로 46라인에서 GASSEnvironment 인스턴스에 연결정보를 알려준다.

제시한 코드를 이용하여 시뮬레이션을 수행하면, Fig. 7과 같은 결과가 화면에 표시된다. 코드에서 입력한 컴포넌트의 위치 및 인접행렬을 이용하여 GASS는 Fig. 7(a)와 같이 네트워크 형태로 문제를 도식해 준다. 그리고 주어진 조건에 따라 시뮬레이션을 수행하고, 각 절점의 변위를 Fig. 7(b)와 같이 그래프로 나타낸다. 그래프는 앞서 Fig. 6에서 작성한 코드에서 컴포넌트 속성으로 "plot_item"이라는 변수에 원하는 변수 이름 (여기서는

y축 변위량인 "v"를 입력함)을 매개변수로 넘겨주면, GASS에서 자동으로 값의 변화를 도식한다. 앞서 Matlab® 코드의 해석 결과 (-0.9531 mm, -0.2647 mm)와 같은 결과로 접근하고 있음을 Fig. 7(b)에서 확인할 수 있다. 모의된 모든 값은 sqlite라는 파일 데이터베이스로 저장되는데, 계산결과 기존 연구결과와 유효숫자 내에서 동일한 결과를 확인하였다.

고정단을 제외한 두 절점인 2번 절점과 3번 절점의 y방향 변위를 도식한 Fig. 7(b)는 반복계산횟수가 증가함에 따라 특정 값에 수렴하는 형태를 보여주었다. 이는 두 절점이 주변의 응력에 따라 자기의 위치를 주도적으로 변화시키면서 적응하는 행태로 이해할 수 있다. 그래프를 좀 더 자세히 살펴보면, 2번 절점에 작용하는 힘에 의해서 2번 절점이 먼저 움직이게 되고, 다음 반복계산 시 2번 절점의 변위에 따라 3번 절점도 평행을 이루기 위해서 움직인다. 반복계산이 진행됨에 따라, 3번 절점의 변위로 인해 2번 절점이 다시 평행을 이루기 위하여 위치를 이동하며, 이와 같은 반복 수행을 통하여 모든 절점이 평행상태에 도달하게 된다.

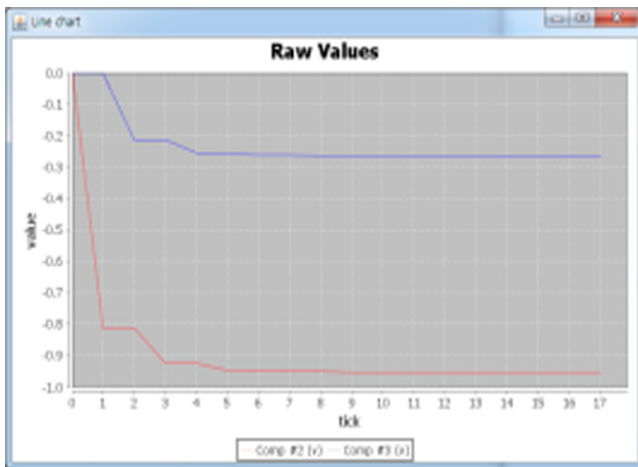
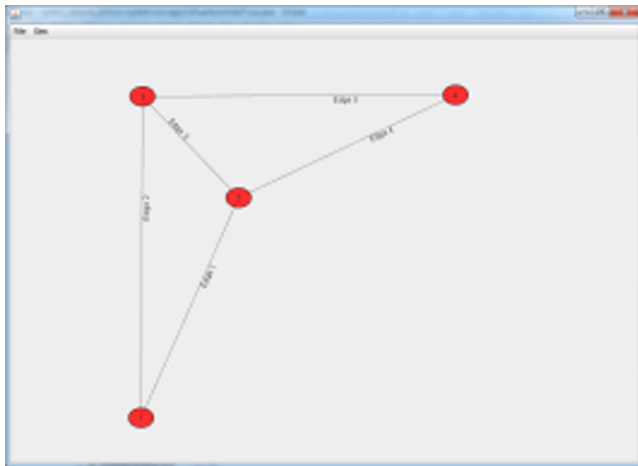


Fig. 7 Results of 5-truss analysis using GASS

2. 11-트러스 구조의 해석

대표적인 구조해석 프로그램 중 하나인 ANSYS®를 이용한 해석 결과 (University of Alberta, 2001)와 비교하기 위하여 Fig. 8과 같이 11-트러스 구조(Chandrupatla and Belegunda, 2002)를 해석하였다. 트러스는 모두 동일한 물성치를 가지며, 탄성계수는 200 GPa, 단면적은 32.5 cm²으로 설정하였다. 특히 7번 절점이 롤러 고정단으로 y축으로는 변위가 없되, x축으로는 변위를 가질 수 있다는 점에서 이전 문제와는 다른 특징을 가지고 있다.

상용 구조해석 프로그램인 ANSYS® 7.0를 이용하여 해석한 결과와 본 연구에서 개발한 트러스 구조 컴포넌트를 이용한 해석 결과를 Table 2에 정리하였다. 두 해석 결과 모두 모든 절점에서 동일한 변위 값을 확인할 수 있었다.

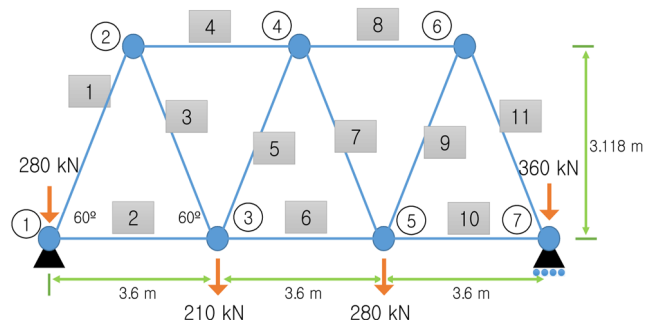


Fig. 8 Example for 11-truss structure (Modified from Chandrupatla and Belegunda, 2002)

Table 2. Displacements comparison of ANSYS and this study

Node	ANSYS®		GASSTrussComponent	
	Ux	Uy	u	v
1	0.0000	0.0000	0.00000	0.00000
2	3.0836	-3.5033	3.08364	-3.50328
3	0.74604	-6.5759	0.74604	-6.57587
4	1.5916	-7.2363	1.59155	-7.23630
5	2.3127	-6.9923	2.31273	-6.99228
6	-0.49736E-01	-3.7330	-0.04973	-3.73301
7	3.1334	0.0000	3.13337	0.00000

3. GASS 트러스 모델과 기존 해석 프로그램 비교

이상에서 살펴본 바와 같이 독립성이 유지되는 시뮬레이션 환경인 GASS를 이용하여 트러스 구조를 정확하게 해석할 수 있음을 확인하였다. 기존 구조해석과 GASS를 이용한 구조해석의 차이를 비교하기 위하여, 기존 구조해석 프로그램의 대표적인 방법인 유한요소법 (Lee, 1991)과 GASS 트러스 모델의 계산 순서도를 Fig. 9와 같이 도시하였다.

기존 구조해석 프로그램은 전체 강성행렬을 만든 후, 행렬을 계산하고 그 결과를 후처리 과정에서 절점과 트러스에 배분하는 구조를 갖는다. 반면, GASS를 이용한 트러스 구조해석 모형은 각 절점에서 지배방정식을 만든 후, 주변 절점에서 정보를

취득하여 대상 절점의 지배방정식을 계산하고, 그 결과를 화면에 반영한 후, 다시 주변에서 변경된 정보를 갱신하여 지배방정식을 계산하는 과정을 반복한다. 이는 기존 프로그램에서 강성행렬을 반복법으로 풀다면 동일한 해석과정으로 볼 수도 있고, 행렬 계산을 위한 다양한 알고리즘을 적용할 수 있는 기존 방식이 계산 속도 및 효율은 더 좋다고 평가할 수 있다.

그러나 전체 강성행렬을 만들어서 풀게 되면, 전체 강성행렬을 만드는 조합과정이 매우 번거롭고, 전체 강성행렬을 해석할 때까지 결과를 알 수 없다는 단점이 있다. 반면 GASS 환경에서는 독립적으로 컴포넌트가 해석될 수 있으며, 연구자는 단순히 절점의 속성과 트러스의 속성을 입력함으로써 손쉽게 시뮬레이션 수행이 가능하다. 시뮬레이션이 수행되는 동안 컴포넌트 내에서 속성 값이 변화하는 과정을 동적으로 시각화함으로써 현상을 이해하는데, 도움을 준다.

또한 트러스가 정적 평행상태를 이루지 못하고 rigid motion이 되는 경우, 기존 프로그램은 전체 강성행렬을 생성하여 해석하므로 행렬을 계산할 수 없다는 정보만을 사용자에게 제공한다. 반면 본 연구의 결과물은 반복해법을 적용하여 실시간으로 계산결과를 시각화하기 때문에 구조물이 형상을 유지하지 못하고, 값이 발산하는 그래프를 통하여 형상의 구조적 문제가 있음을 연구자에게 쉽게 전달한다.

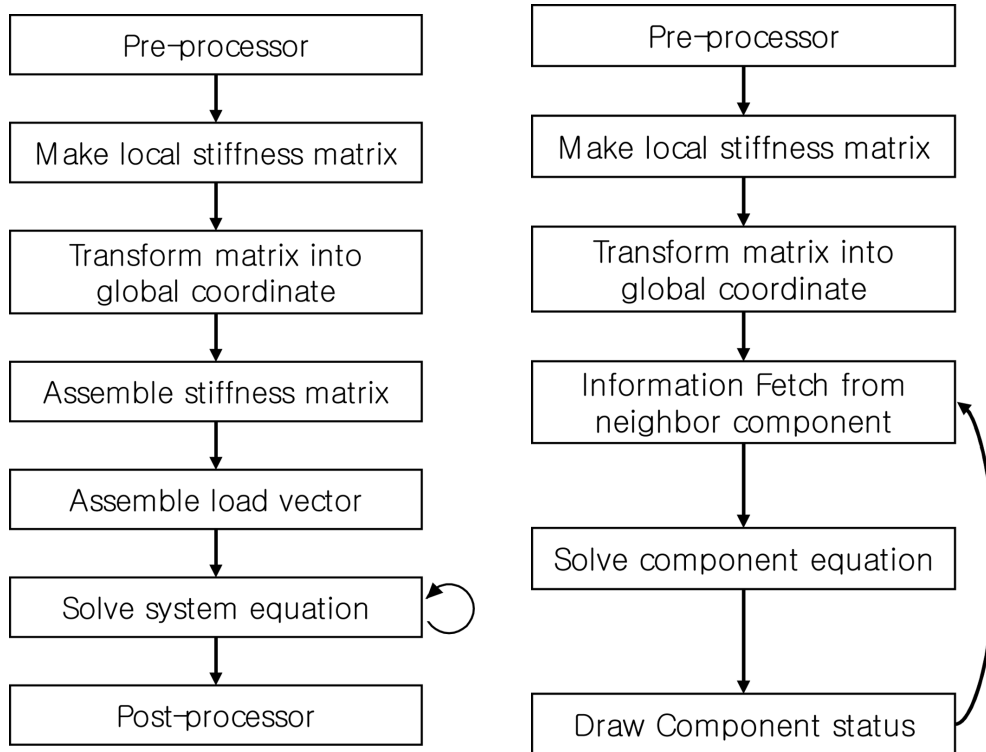


Fig. 9 Flow chart of program FEM and truss component using GASS

V. 결 론

구조물을 설계하기 위해서는 구조물의 내부 응력 흐름을 파악할 필요가 있고, 이를 위하여 트러스를 이용하여 구조물을 해석하려는 많은 연구가 이루어지고 있다. 트러스는 비교적 간단한 구조물인데 반하여, 일반적으로 행렬을 이용하여 계산하므로, 전체 강성행렬을 생성하는 과정이 복잡하고, 계산 과정을 이해하기 어려운 단점이 있다. 본 연구에서는 이와 같은 문제를 해결하기 위하여 독립적으로 실행이 가능한 트러스 구조 컴포넌트를 개발하고, 개발한 컴포넌트를 연결하여 GASS 환경에서 시뮬레이션하였다.

먼저 절점을 중심으로 트러스 구조를 정식화하여 트러스 구조 컴포넌트를 개발하고, GASS 환경에서 구현하여 시뮬레이션한 결과, 트러스의 해석 결과가 유요소자 내에서 일치하고, 계산과정동안 실시간으로 절점의 변위를 확인할 수 있었다.

절점을 중심으로 8방향 트러스를 갖지 않는 삼각형 트러스 구조에 대하여 Matlab® 코드와 결과를 비교하였다. 비교 결과 유요소자 내에서 동일한 결과를 보였으며, 강성 행렬을 만들지 않고, 단지 47라인만으로 문제를 정의할 수 있음을 확인하였다.

상용 프로그램인 ANSYS® 7.0와 비교하기 위하여 11-트러스 문제를 해석해 보았으며, 시뮬레이션 결과 동일한 처짐량을 구할 수 있었다. 또한, 본 연구 결과를 이용하여 롤러 힌지에 대해서도 올바른 해석결과를 얻을 수 있었다. 문제가 rigid motion을 보일 경우, 상용프로그램에서는 계산 오류만을 표시해주지만, 본 시뮬레이션 환경에서는 절점의 변위가 발산하는 모습을 확인함으로써 현상을 이해하는데 도움을 준다고 판단된다.

향후 GASS를 이용하기 위한 전처리 프로그램이 개발되면, 일반 사용자가 다양한 문제에 대하여 코딩을 하지 않고도 손쉽게 시뮬레이션을 수행할 수 있을 것으로 기대된다. 또한 트러스 구조 컴포넌트의 확장을 통하여 판(plate) 문제 해석을 위한 컴포넌트로 확장이 가능할 것으로 사료되며, GASS는 반복 계산을 통하여 구조 최적화를 손쉽게 구현할 수 있는 환경으로 발전이 가능할 것으로 판단된다.

본 논문은 2013년도 정부(교육과학기술부)의 재원으로 한국연구재단의 기초연구사업 지원(과제번호: NRF-2012R1A1A1015755)에 의해 수행되었습니다.

REFERENCES

1. Bhatti, M. A., 2005. Fundamental finite element analysis and applications: with Mathematica and Matlab computations,

- pp. 25-28. Wiley.
2. Chandrupatla, T. R. and A. D. Belegunda, 2002. Introduction to finite elements in engineering, p.123, Prentice Hall.
3. Choi, W., 2012. Regular and irregular truss structure models equivalent to continuum structure. Ph.D. diss., Seoul National University.
4. Goodliffe, P., 2007. Code Craft the practice of writing excellent code. ch. 1. (translator: Kim, Y. M., 2006). pp. 3-22. Hanbit Media Inc.
5. Gürdal, Z. and T. Brian, 2000. Cellular automata for design of truss structures with linear and nonlinear response, AIAA Structures, Structural Dynamics and Materials Conference. AIAA Paper 1580.
6. Kim, S. H., J. B. Jeon, J. S. Park, and S. S. Yoon, 2011. The Research of stress route for concrete structure using advanced progressive optimization. Journal of the Korean Society of Agricultural Engineers 53(6): 153-163 (in Korean).
7. Kim, T. G., 2007. System simulator improvement using centralized component attribute control and diverse communication layers of material and information. MS. thesis, Seoul National University.
8. Kim, T. G. and J. J. Lee, 2007. GASS Improvement using diverse communication layers of material and information. Journal of the Korean Society of Agricultural Engineers 49(1): 101-109 (in Korean).
9. Kim, Y. S., N. S. Jung, J. J. Lee, and S. S. Yoon, 2004. Development of a stress Path search model of evolutionary structural optimization using TIN. Journal of the Korean Society of Agricultural Engineers 46(4): 65-71 (in Korean).
10. Lee, H. K., T. G. Kim and J. J. Lee, 2012. Development of an object-oriented finite element model through iterative method ensuring independency of elements. Journal of the Korean Society of Agricultural Engineers 54(2): 115-125 (in Korean).
11. Lee, J. J., 1991. Reliability analysis modeling of frame structures based on discretized ideal plastic method. Ph.D. diss., Seoul National University.
12. Lee S. Y., T. G. Kim, and J. J. Lee, 2009. An equivalent truss model by discretizing continuum structure. Journal of the Korean Society of Agricultural Engineers 51(3): 45-52 (in Korean).
13. Suh, K., T. G. Kim, H. J. Lee, and J. J. Lee, 2005. Simulation

- of agricultural marking based on GASS. Journal of the Korean Society of Agricultural Engineers 47(4): 3-12 (in Korean).
14. University of Alberta, 2001. University of Alberta - ANSYS Tutorials: Two Dimensional Truss, <http://www.mece.ualberta.ca/tutorials/ansys/BT/Truss/Truss.html> Accessed 10 Nov. 2013.
15. Yi, H. J., 2003. Simplifying complexity and emerging self-organization of integrated agricultural system models. Ph.D. diss., Seoul National University.
16. Yoon, S. S and J. J. Lee, 2002. A Study on the reinforcement of reinforced concrete using evolutionary structural optimization. Journal of the Korean Society of Agricultural Engineers 44(2): 127-135 (in Korean).