



농업시스템응용플랫폼을 이용한 2계 편미분 방정식의 해석 Numerical Solution of Second Order Linear Partial Differential Equations using Agricultural Systems Application Platform

이성용* · 김태곤**† · 서교*** · 한이철**** · 이제명***** · 이호재***** · 이정재*****
Lee, SungYong · Kim, Taegon · Suh, Kyo · Han, Yicheol · Lee, Jemyung · Yi, Hojae · Lee, JeongJae

Abstract

The Agricultural Systems Application Platform (ASAP) provides bottom-up modelling and simulation environment for agricultural engineer. The purpose of this study is to expand usability of the ASAP to the second order partial differential equations: elliptic equations, parabolic equations, and hyperbolic equations. The ASAP is a general-purpose simulation tool which express natural phenomenon with capsuled independent components to simplify implementation and maintenance. To use the ASAP in continuous problems, it is necessary to solve partial differential equations. This study shows usage of the ASAP in elliptic problem, parabolic problem, and hyperbolic problem, and solves of static heat problem, heat transfer problem, and wave problem as examples. The example problems are solved with the ASAP and Finite Difference method (FDM) for verification. The ASAP shows identical results to FDM. These applications are useful to simulate the engineering problem including equilibrium, diffusion and wave problem.

Keywords: ASAP; Object-oriented; Simulator; Second order partial differential equation

1. 서론

라플라스 방정식, 확산 방정식, 파동방정식으로 대표되는 2계 편미분 방정식은 자연계의 다양한 현상을 묘사하고 해석할 수 있다 (Kreyszig, 1988). 라플라스 방정식은 문제영역의 절점 혹은 요소에 유입되는 물질 혹은 에너지의 양이 유출되는 양과 동일한 정적상태를 설명하는 방정식으로, 평행상태

에 도달한 상태만을 설명할 뿐 시간에 따른 변화량을 고려하지 않는 특징을 갖는다. 이와 달리 확산 방정식은 시간에 따라 물질이나 에너지의 확산이 일어나는 경우에 이용할 수 있으며, 파동 방정식은 진동하는 물체의 운동과 같은 주기성을 갖는 문제를 해석할 수 있다. 편미분 방정식은 특수한 문제 형태에 대해서만 해석해가 존재하므로 현실 문제를 풀이하는데 한계가 있다. 따라서 공학분야에서는 다양한 형태의 문제를 풀기 위하여 유한요소해석, 유한차분해석, 경계요소해석과 같은 수치해석방법을 이용하여 문제를 해석한다. 공학에서 사용하는 구조해석 프로그램 및 유체해석 프로그램들은 일반적으로 수치해석방법에 기반하여 계산을 수행한다 (Choi and Kim, 2015; Jeon et al., 2012; Yeo et al., 2015; Yi, 2003).

공학에서 사용하는 수치해석 프로그램은 컴퓨터의 성능향상과 계산 알고리즘의 개선으로 단순화된 지배방정식을 푸는 형태에서 점차 다양한 요소들이 복합적으로 작용하는 문제의 해석으로 범위를 넓혀가고 있다. 최근에는 주어진 문제를 컴포넌트 단위로 나누고, 단위 컴포넌트를 연결하여 실제현상을 모의하는 방법으로 많은 연구가 진행되고 있다. Jun et al. (2014) 와 Kang et al. (2015)은 물순환 해석을 위해 농촌용수의 다양한 모의 요소들을 객체지향적 관점에서 독립적인 컴포넌트로 구성된 모의시스템을 개발하고자 하였으며, Park et al. (2015)는 웹기반 물순환 해석시스템을 개발하기 위해 데이터베이스와 개별 모델들이 유기적으로 연결된 시스템을 연구하였다. Lee et al. (2012)은 점진적으로 구조해석 수행이

- * Research Institute of Agriculture and Life Sciences, Seoul National University
- ** Institute on the Environment, University of Minnesota, USA
- *** Graduate School of International Agricultural Technology, Institute of Green Bio Science & Technology, Seoul National University
- **** Northeast Regional Center for Rural Development, The Pennsylvania State University, USA
- ***** Division of Environmental Science and Technology, Kyoto University, Japan
- ***** Department of Agricultural and Biological Engineering, The Pennsylvania State University, USA
- ***** Department of Landscape Architecture and Rural Systems Engineering, Institute of Green Bio Science & Technology, Seoul National University
- † Corresponding author
Tel.: +1-425-219-8377 Fax: +1-612-626-5555
E-mail: taegon.kim@daum.net

Received: December 10, 2015

Revised: January 19, 2016

Accepted: January 25, 2016

가능한 컴포넌트 기반의 유한요소해석 프로그램을 개발하였다. 그러나 이들 연구는 특정 문제를 풀이하기 위하여 개발한 시스템으로 각기 다른 자료구조와 프로그램 입출력 구조를 가지기 때문에, 연구자가 서로 다른 분야의 모델을 재이용하거나, 통합하여 새로운 모델로 구성하기 어렵다.

농업시스템응용플랫폼 (Agricultural Systems Application Platform, ASAP)은 다양한 분야의 모델들을 조합하여 원하는 목적의 모델을 쉽게 구성할 수 있도록 개발된 범용 시뮬레이션 환경이다 (Kim, 2014; Yi, 2003). 반복계산이 필요한 문제나 도메인 내외부의 조건이 변화하는 문제에 간편하게 적용할 수 있으며, 비교적 단순하고 직관적인 프로그래밍을 통하여 객체지향 모델링이 가능하다는 장점이 있다 (Kim and Lee, 2007, 2009; Kim et al., 2014a). 농업시스템응용플랫폼을 이용한 시뮬레이션을 위해서는 1) 컴포넌트 개발과 2) 컴포넌트의 조합을 통한 모델 구성 및 구동 단계가 요구된다 (Kim, 2014). 이전 연구는 특정 문제를 해결하기 위한 컴포넌트 개발 및 모델 구성을 제시하여 직관적인 피드백과 대화형 모델링과 같은 농업시스템응용플랫폼의 장점을 보여주었다. 그러나 제시한 연구결과와 유사한 대상이 아닌 경우에는 기존 컴포넌트를 재활용하기 어려워 활용분야가 제약되는 한계가 있었다. 이에 다양한 연구 분야에서 공통적으로 이용 가능한 기본 컴포넌트의 필요성이 대두되었다.

본 연구는 자연계를 기술하는 기초 지배방정식인 2계 편미분 방정식을 농업시스템응용플랫폼으로 풀이함으로써, 다양한 분야의 연구자가 효과적으로 컴포넌트를 개발하고, 시뮬레이션 모델을 구축할 수 있는 기본 모형을 제시하는데 목적이 있다. 2계 편미분 방정식은 시간이 흐름에 따라 필드가 변화하는데, 이를 컴포넌트로 구현하고, 모델을 구성하여 시뮬레이션을 수행하였다. 모델의 검증을 위하여 2계 선형 편미분 방정식의 3가지 형태인 평형 방정식, 확산 방정식, 파동 방정식에 대한 예제를 구성하여 해석하였다.

II. 2계 편미분방정식 컴포넌트의 정식화

편미분 방정식은 여러 개의 독립변수로 구성된 함수와 그 함수의 편미분으로 구성된 방정식이다. 각각의 변수들의 상관관계를 고려하지 않고 독립적인 변화량을 이용하여 간결하게 문제를 기술할 수 있기 때문에 열의 확산이나 소리의 전파, 유체역학, 전자기학과 같이 다양한 분야에서 이용되고 있다 (Kreyszig, 1988). 2계 편미분 방정식의 일반 형태 (General form)는 식 (1)과 같다.

$$a \frac{\partial^2 T}{\partial x^2} + b \frac{\partial^2 T}{\partial x \partial y} + c \frac{\partial^2 T}{\partial y^2} + d \frac{\partial T}{\partial x} + e \frac{\partial T}{\partial y} + fu + g = 0 \quad (1)$$

편미분 방정식의 풀이과정은 판별식 $b^2 - 4ac$ 의 값에 따라 달라지게 되는데, 이에 따라 수치해석기법이 달라지게 된다. 판별식의 값이 0보다 작은 경우는 경계조건으로 내부의 문제영역을 해석하는 타원형 문제에 해당하며, 판별식 값이 0인 경우는 포물선형 문제이며, 0보다 큰 경우에는 쌍곡선형 문제가 된다.

1. 평형 방정식

타원형 문제로도 알려져 있는 평형 방정식은 단위 시간 동안 유입되는 에너지 혹은 물질의 양이 같은 정상상태에서의 문제이며, 이를 방정식으로 표현한 것이 라플라스 방정식이다. 라플라스 방정식은 충분한 시간이 지난 후 대상체의 변화과정이 안정된 상태를 기술한 방정식으로 시간의 영향이 없는 상태를 의미하기 때문에, 포물선형과 쌍곡선형 문제와 달리 시간 항을 무시한다. 고체의 열 분포나 지하수의 흐름과 같이 일정 상태가 지속되고 있는 문제가 라플라스 방정식의 대표적인 형태이다.

라플라스 방정식을 텐서형으로 표현하면 아래 식 (2)와 같으며,

$$\Delta T = f \quad (2)$$

이때의 편미분 기호로서 아래 식 (3)과 같이 각각의 축방향으로 편미분한 항의 총합을 의미한다.

$$\Delta T = \frac{\partial^2 T}{\partial x_1^2} + \dots + \frac{\partial^2 T}{\partial x_n^2} \quad (3)$$

위 식 (3)을 2차원 평면에서 $\Delta x = \Delta y$ 인 경우에, 유한차분법 (Finite Difference Method, FDM)으로 정리하면 아래 식과 같다.

$$T_{m+1,n} + T_{m-1,n} + T_{m,n+1} + T_{m,n-1} - 4T_{m,n} = 0 \quad (4)$$

여기에서 m 과 n 은 수치모형에서 임의 절점의 위치를 나타낸다.

2. 확산 방정식

확산 방정식은 포물선형 방정식으로도 알려져 있으며 2계

편미분 방정식의 계수가 $b^2 - 4ac = 0$ 인 경우에 해당한다. 평형 방정식과 달리 정상상태에 도달하지 않은 상황에서 시간에 따라 변화하는 현상을 표현할 때 적합하다. 열의 확산이나 유체 내에서의 물질 확산과 같이 질량이 존재하지 않거나 무시할 수 있을 정도로 작은 경우에 주로 사용하기 때문에 확산 방정식으로 알려져 있으며, 식 (5)와 같은 형태로 기술된다 (Kreyszig, 1988).

$$\left(\frac{\partial}{\partial t} - \Delta\right)T = f \quad (5)$$

일반적으로 확산 방정식은 유한차분법을 이용하여 해석이 가능하다. 확산방정식의 대표적인 예인 열 확산 문제를 1차원 평면 벽으로 기술하면 아래 식 (6)으로 나타낼 수 있다.

$$\frac{\partial^2 T}{\partial x^2} = \frac{1}{\alpha} \frac{\partial T}{\partial t} \quad (6)$$

이 식은 경계조건과 초기조건이 주어졌을 때 위치에서 시점에서의 온도 함수 $T(x, t)$ 에 관한 식이다. 위 식을 통해 시간 대별로 변화하는 온도를 계산할 수 있다. 위의 식 (6)을 2차원 양함수 유한차분법으로 전개하면 아래 식 (7)과 같이 나타낼 수 있다. 여기서 m 과 n 은 절점의 위치를 의미한다.

$$T_{m,n}^{t+\Delta t} = F_0(T_{m+1,n}^t + T_{m-1,n}^t + T_{m,n+1}^t + T_{m,n-1}^t) + (1 - 4F_0)T_{m,n}^t \quad (7)$$

F_0 는 아래 식 (8)으로 나타낼 수 있으며, 그 값이 1/4보다 작은 경우에만 그 해를 신뢰할 수 있다 (Courant et al., 1967).

$$F_0 = \frac{1}{\alpha} \frac{\Delta t}{(\Delta x)^2} \quad (8)$$

3. 파동 방정식

파동 방정식은 2계 편미분 방정식의 계수가 $b^2 - 4ac > 0$ 인 경우로, 진동하는 현상을 설명하는 방정식으로 쌍곡선형 방정식으로도 지칭된다. 일반적으로 식 (9)로 나타낼 수 있으며, 이를 2차원 문제로 전개하면, 식 (10)으로 표현된다 (Kreyszig, 1988).

$$\left(\frac{\partial^2}{\partial t^2} - \Delta\right)\mathbf{U} = f \quad (9)$$

$$\frac{\partial^2}{\partial t^2} u = c^2 \left(\frac{\partial^2}{\partial x^2} u + \frac{\partial^2}{\partial y^2} u \right) \quad (10)$$

이 식에서 c 는 파동의 전파계수를 의미한다. 이 식을 2차원 편미분 방정식으로 전개하면, 식 (11)이 된다.

$$\begin{aligned} \frac{\partial^2}{\partial t^2} u_{i,j}^k &\approx \frac{1}{dt^2} (u_{i,j}^{k+1} - 2u_{i,j}^k + u_{i,j}^{k-1}) \\ \frac{\partial^2}{\partial x^2} u_{i,j}^k &\approx \frac{1}{dx^2} (u_{i+1,j}^k - 2u_{i,j}^k + u_{i-1,j}^k) \\ \frac{\partial^2}{\partial y^2} u_{i,j}^k &\approx \frac{1}{dy^2} (u_{i,j+1}^k - 2u_{i,j}^k + u_{i,j-1}^k) \end{aligned} \quad (11)$$

식 (10)에 식 (11)을 대입하여 전개하면, 식 (12)와 같다.

$$u_{i,j}^{k+1} = 2u_{i,j}^k - u_{i,j}^{k-1} + (dt \cdot c) \left[\frac{1}{dx^2} (u_{i+1,j}^k - 2u_{i,j}^k + u_{i-1,j}^k) + \frac{1}{dy^2} (u_{i,j+1}^k - 2u_{i,j}^k + u_{i,j-1}^k) \right] \quad (12)$$

만약 $dx = dy$ 라면 식 (13)이 된다.

$$u_{i,j}^{k+1} = 2u_{i,j}^k - u_{i,j}^{k-1} + (dt \cdot c) \left[\frac{1}{dx^2} (u_{i+1,j}^k + u_{i,j+1}^k + u_{i-1,j}^k + u_{i,j-1}^k - 4u_{i,j}^k) \right] \quad (13)$$

III. 농업시스템응용플랫폼 컴포넌트의 구현

1. 농업시스템응용플랫폼 컴포넌트

농업시스템응용플랫폼은 Java 언어로 개발된 시뮬레이션 플랫폼으로 시스템을 구성하는 기본 구성요소를 독립적으로 구동 가능한 컴포넌트로 정의하고, 이들 컴포넌트를 단순히 연결시키는 작업만으로 구성요소간의 유기적 상호작용을 통하여 전체 시스템의 창발적 거동을 모의하도록 개발되었다 (Yi, 2003; Kim, 2014). 시뮬레이션을 위해서는 독립적으로 구동가능한 컴포넌트를 작성하는 단계와 개발된 컴포넌트의 기본 속성값을 지정하고, 물리계와 정보계의 연결망을 설정하여 시뮬레이션을 수행하는 단계로 구성된다.

컴포넌트는 농업시스템응용플랫폼에서 제공하는 컴포넌트 인터페이스를 상속하여 주요 거동규칙을 기술해야 한다 (Kim et al., 2014b). 주요한 코드는 시뮬레이션 시간 단위인 틱 (tick)마다 컴포넌트의 값을 조정하는 `preProcess()`, `postProcess()` 메소드와 주변 컴포넌트의 정보를 가져와서 정보처리를 수행하는 필수적인 `process()` 메소드가 있다. 본 연구는 다양한 분야에서 활용 가능한 기초 컴포넌트를 개발하

는데 목적이 있으므로, 편미분 컴포넌트에 대하여 구동에 필요한 코드 전체를 제시하였다.

농업시스템응용플랫폼에서 시뮬레이션 모델을 구성하는 과정은 개발된 컴포넌트를 화면 상에 적절히 배치하고, 초기 값을 지정하는 과정이 전부이다. 인접한 컴포넌트는 인접성을 기반으로 자동으로 연결되며 (Yi, 2003), 정보계의 연결망은 공간적으로 표현하기 어려워 연결망을 사용자가 지정해주어야 한다 (Kim and Lee, 2007). 본 연구에서 제시한 2계 편미분 방정식은 주로 연속된 장을 설명하기 위한 식이므로 컴포넌트를 적절히 위치시키는 것만으로 모델 구성을 완료할 수 있다. 구성된 모델은 시작, 정지, 재시작 버튼을 통해서 시뮬레이션을 구동할 수 있으며, 시뮬레이션 결과는 데이터베이스에 저장되고, 그 결과는 그래프와 컴포넌트의 색상 및 위치 변화로 직관적으로 확인 가능하다.

2. 평형 방정식

일반적으로 평형 방정식의 수치해석은 식 (4)와 같이 유한차분법으로 전개한 후 이에 경계조건을 대입한 후 해석하는 방법을 택한다. ASAP은 객체지향기법을 이용하여 컴포넌트 단위로 모델링을 수행하며, 각 컴포넌트는 자신의 속성값과 이웃한 컴포넌트들의 속성값을 비교하여 지배방정식을 평가하고, 이때 생기는 오차를 분배하는 방식의 반복법으로 문제를 해결한다. 상대적 위치는 컴포넌트를 연결시키는 컴포넌트 조합과정에서 반영되므로, 컴포넌트의 고유 행태에 집중하여 모델을 구현할 수 있다. 식 (4)는 임의의 유한요소의 값은 이웃한 4개의 유한요소의 평균값이라는 의미로 치환할 수 있다. 이러한 개념을 바탕으로 정상상태에 도달하는 열전달 메커니즘을 다음과 같이 정의하였다. 임의의 열전달 컴포넌트는 매 시간대마다 이웃한 컴포넌트들의 값을 확인하고 자신

의 값이 이웃 컴포넌트의 온도보다 낮을 경우, 일정량의 온도를 취하며, 이때 온도 전달량을 식 (14)의 열전도 공식에 의해 결정하도록 구현하였다.

$$H = -\frac{k}{l}(T_{neighbor} - T_{me}) \quad (14)$$

여기서, H 는 열류량을 의미하며, k 는 열전도율, l 는 거리를 의미한다. 구현한 코드는 Fig. 1에서 process() 메소드에 해당하며, 매 틱마다 전달량을 이용하여 현재 온도를 갱신하도록 preProcess() 메소드를 구현하였다.

3. 확산 방정식

본 연구에서 확산 방정식의 적용으로 동적 열확산 문제를 선정하였다. 동적 열확산방정식은 시간항을 포함한 유한차분식의 형태로, 식 (7)과 (8)을 이용하여 계산할 수 있다. 각 컴포넌트는 가로, 세로가 동일한 정방 유한체로 가정하여, 컴포넌트의 길이를 모두 동일하게 dx 로 설정하였으며, 각 컴포넌트는 속성값으로 공간차분항 dx 와 시간차분항 dt 변수를 갖도록 정의하였다. 확산방정식 풀이를 위한 컴포넌트 코드를 Fig. 2와 같이 제시하였다. 매 틱마다 각 컴포넌트는 준비 단계로 preProcess()을 실행하는데, 확산 문제에서는 이전 스텝의 온도를 과거의 온도로 저장하고, 온도 전달량을 0으로 초기화하도록 코드를 작성하였다. 주변 컴포넌트와 정보를 주고받으며, 정보를 처리하는 주요 코드인 process()에서 이웃한 컴포넌트들의 값을 이용하여 자신의 값을 갱신할 수 있도록 구현함으로써 시간에 따른 온도의 변화를 계산하고 저장하도록 구현하였다.

```

1. @Override
2. public void preProcess() {
3.     set("preTemp", get("temp"));
4.     set("temp", get("temp") + get("trans"));
5.     set("trans", 0);
6. }
7.
8. @Override
9. public void process(List<TreeNode> neighborhoods) {
10.    for (TreeNode neighborhood : neighborhoods) {
11.        double neighborhoodTemp = neighborhood.getModel().get("preTemp");
12.        double temp = get("preTemp");
13.
14.        if (temp < neighborhoodTemp) {
15.            double trans = k/l*(neighborhoodTemp - temp);
16.
17.            if (trans > (neighborhoodTemp - temp) / 2) {
18.                trans = (neighborhoodTemp - temp) / 2;
19.            }
20.
21.            set("trans", trans + get("trans"));
22.
23.            if (neighborhood.getModel().get("isTrans") > 0) {
24.                neighborhood.getModel().set("trans",
25.                    neighborhood.getModel().get("trans") - trans);
26.            }
27.        }
28.    }
29. }
    
```

Fig. 1 Component source code for static heat transfer problem

```

1. @Override
2. public void preProcess() {
3.     set("preTemp", get("temp"));
4.     set("trans", 0);
5. }
6.
7. @Override
8. public void process(List<TreeNode> neighborhoods) {
9.
10.    double dt = this.get("dt");
11.    double dx = this.get("dx");
12.    double alpha = this.get("thermalDiffusivity");
13.
14.    double Fo = alpha * dt / (dx * dx);
15.
16.    double sumNeighborhoodsTemp = 0.0;
17.    double preTemp = get("preTemp");
18.
19.    for (TreeNode neighbor : neighborhoods) {
20.        if (!neighbor.getModel().hasValueKey("preTemp")) continue;
21.        double neighborhoodTemp = neighbor.getModel().get("preTemp");
22.
23.        sumNeighborhoodsTemp += neighborhoodTemp;
24.    }
25.
26.    double temp = Fo * sumNeighborhoodsTemp + (1 - 4 * Fo) * preTemp;
27.
28.    this.set("temp", temp);
29. }
    
```

Fig. 2 Component source code for dynamic heat transfer problem

```

1. @Override
2. public void preProcess(){
3.     set("prePreU", get("preU"));
4.     set("preU", get("U"));
5. }
6.
7. @Override
8. public void process(List<TreeNode> neighborhoods){
9.     double preU = this.get("preU");
10.    double prePreU = this.get("prePreU");
11.
12.    double sumNeighborPreU = 0.0;
13.
14.    for(TreeNode neighbor : neighborhoods){
15.        sumNeighborPreU += neighbor.getModel().get("preU");
16.    }
17.
18.    double dt = this.get("dt");
19.    double dx = this.get("dx");
20.    double c = this.get("c");
21.
22.    double newU = 2.0 * preU - prePreU + ( dt * c ) * ( 1.0 / dx / dx )
23.        *(sumNeighborPreU - 4 * preU);
24.
25.    this.set("U", newU);
26. }
    
```

Fig. 3 Component source code for wave problem

4. 파동 방정식

파동방정식의 가장 대표적인 예로 식 (12)를 이용하여 Fig. 3과 같이 컴포넌트를 구현하였다. 기본적인 코드 구성은 앞서 확산 컴포넌트와 유사하나, 식 (12)에서 시간에 따른 변화량과 그 변화량의 변화율이 필요하므로, 한 스텝 이전의 값과 두 스텝 이전의 값을 필요하다. 두 개의 변수값을 컴포넌트에서 다루기 위하여 preProcess() 메소드에서 두 변수를 추가하였다. 주요 처리공정인 process() 메소드에서 이웃한 컴포넌트들의 값을 이용하여 현 시점의 값을 계산할 수 있도록 구현하였다. 확산 컴포넌트의 주요 거동은 Fig. 2에서 26번째 라인에 기술된 식이며, 파동 컴포넌트의 주요 거동은 Fig. 3에서 22번째 라인에 기술한 식으로, 두 코드를 비교하면 컴포넌트 개발 과정에서 연구자가 기술해야 하는 주요코드의 형태를 이해할 수 있다.

IV. 결과 및 고찰

1. 평형 방정식

가. 문제 정의

정상상태에 도달했을 때의 열분포를 모의하기 위해서 Fig. 4와 같이 가로와 세로 모두 1.1×10^{-2} 가 되도록 문제영역을 설정하고, 경계요소로 초기값을 입력하였다. 경계조건은 좌측과 상단의 온도를 100°C , 우측을 50°C , 하단을 0°C 로 설정하였다. 각각의 컴포넌트의 변위는 균등하다고 가정하여 $l = 1.0 \times 10^{-2}$ 로 설정하고, 열전도율 $k = 7.0 \times 10^{-4}$ 로 설정하였다. 각 컴포넌트는 상하좌우의 네 개의 컴포넌트와 연결이 되도록 컴포넌트 관계망을 구성하여 이웃 컴포넌트들의 값을 바탕으로 자신의 값을 계산하도록 환경을 설정하였다.

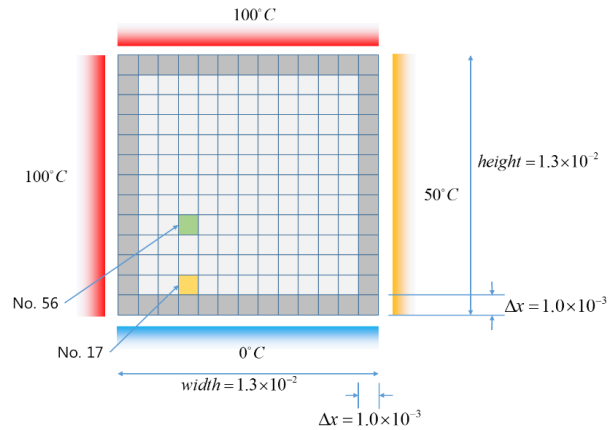


Fig. 4 Problem configuration for heat transfer phenomenon

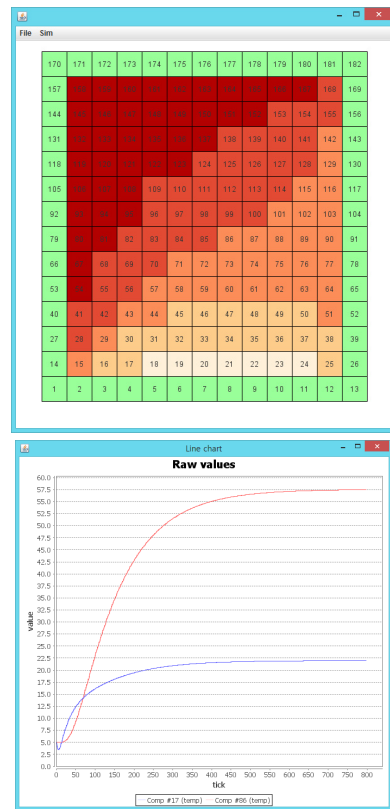


Fig. 5 Screenshot of ASAP for static temperature example

나. 적용 결과

타원형 방정식을 해석하기 위하여 구현한 열확산 컴포넌트를 Fig. 4의 조건과 같이 설정하고 시뮬레이션을 수행하였다. 해석 결과는 Fig. 5와 같이 실시간으로 변화하는 그림과 특정 컴포넌트의 속성값을 관찰할 수 있는 그래프로 도시된다. 화면 왼쪽에는 온도에 따라 컴포넌트의 색상을 나타내고 있는데, 온도가 높은 경계조건을 갖고 있는 좌측과 상단 쪽으

로 갈수록 높은 온도를 보이고 있으며, 우측은 덜 높은 온도, 차가운 경계조건을 갖고 있는 아래쪽은 더 흐린 색을 보여준다. 오른쪽 창에서는 계산과정을 반복해가며 해로 수렴해가는 과정을 보여주고 있다. 컴포넌트의 수가 많기 때문에, Fig. 4에서 표면과 인접한 17번 컴포넌트와 중심부에 위치한 86번 컴포넌트의 값 변화를 도시하도록 설정하였다.

농업시스템응용플랫폼은 문제영역 전체를 한번에 해석하는 것이 아니라 개별 컴포넌트 단위로 이웃한 컴포넌트들과의 계산을 통해 반복적으로 해에 수렴해가는 특징이 있다. 따라서 타원형 방정식과 같이 정상상태에 관한 문제를 해석할 때에도 수렴해가는 과정의 중간값이 존재하나, 수렴하기 이전의 값은 물리적으로는 큰 의미가 없는 값이다. 개별 컴포넌트의 초기온도는 5 °C로 설정하였는데, 수렴과정에서 86번 컴포넌트는 지속적으로 온도가 상승하여 57.5 °C에 수렴하고, 17번 컴포넌트는 하단에 접한 0 °C의 경계조건에 의해 다소 온도가 하락했다가 다시 주변의 다른 컴포넌트들로 인해 온도가 상승하여 22.4 °C로 수렴하였다. 이 두 값을 포함한 모든 컴포넌트들의 수렴값은 유한차분법을 이용하여 계산한 값과 일치하였다.

2. 확산 방정식

가. 문제 정의

확산 방정식은 경계조건과 초기조건이 모두 필요한 문제이다. 앞서 평형 방정식을 이용한 정상상태에 도달한 열확산 문제와의 비교를 위해 Fig. 4의 조건을 동일하게 적용하였다. 추가적으로 설정해야 하는 속성값은 시간간격 dt 와 열확산 계수 α 가 있다. 컴포넌트의 변위는 균등하다고 가정하여 $dx = dy = 0.001$ 로 설정하였고, 시간 간격은 $dt = 1.71$ 로 설정하고, 열확산계수는 $\alpha = 1.43 \times 10^{-1}$ 으로 설정하여 시간에 따른 온도의 변화를 파악하고자 하였다.

나. 적용 결과

열확산문제를 위해 구현한 ASAP 컴포넌트를 연결하고, 위의 문제 조건을 적용하여 모의하였다. 앞서 타원형 방정식을 해석할 때와 달리, 포물선형 방정식을 해석하는 과정에서는 시간 간격 dt 에 따라 변화하는 온도 변화를 모의하는 것이기 때문에 매시간마다 온도 값이 의미를 갖는다. 시간이 흐름에 따라 Fig. 6에 도시한 바와 같이 점차적으로 열이 확산하는 과정을 확인할 수 있다. 사용자가 시각적으로 그 과정을 관찰할 수 있도록 높은 온도일수록 진한 색으로 표시하였다. 시물

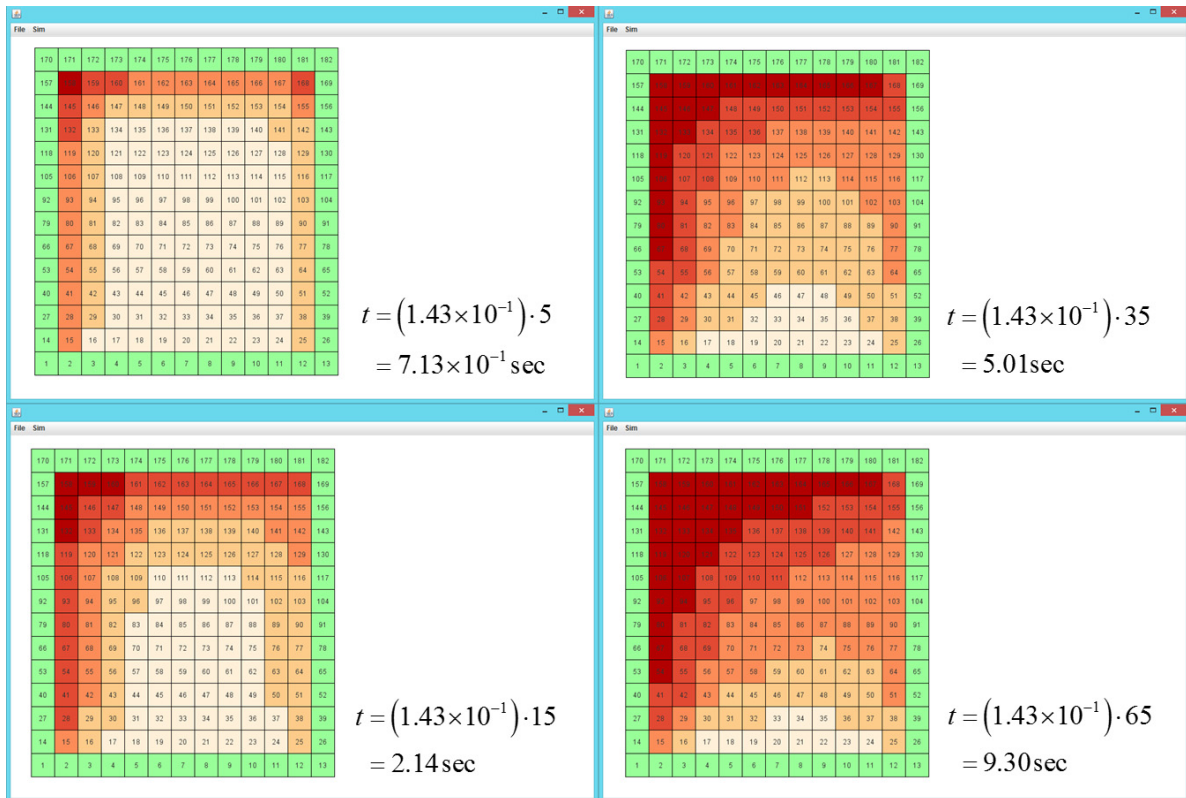


Fig. 6 Dynamic heat transfer simulation by ASAP

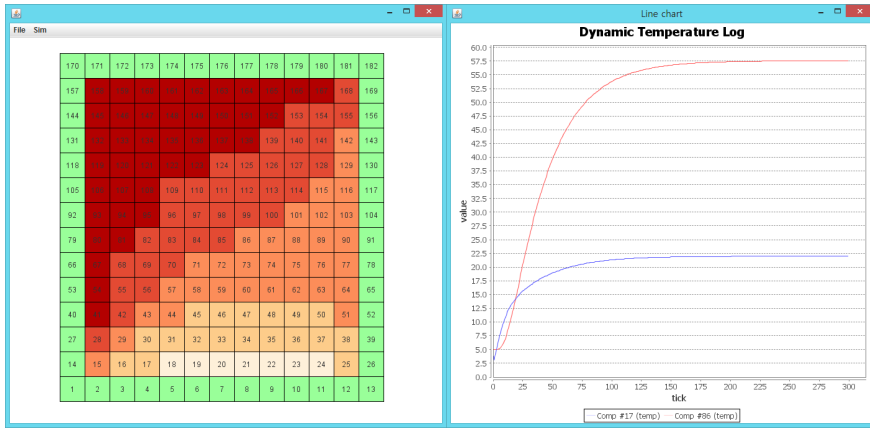


Fig. 7 The screenshot of ASAP for dynamic temperature example

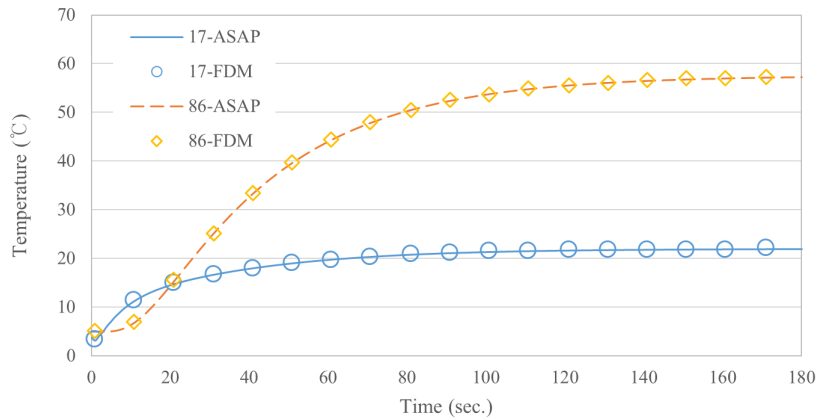


Fig. 8 Comparison of ASAP (this study) and FDM in dynamic temperature simulation

레이션이 종료된 이후 화면 구성은 Fig. 7과 같다. 좌측에는 각 컴포넌트별로 온도 상태를 표현하고 우측에는 시간에 따른 온도그래프를 도시하였다. 모든 컴포넌트의 온도는 데이터베이스에 저장되어 추적 가능하지만, 앞서 평형 방정식 문제의 결과와 비교하기 위하여 동일하게 0 °C로 설정한 경계 영역과 맞닿은 17번 컴포넌트와 중앙의 86번 컴포넌트의 온도를 도시하였다. 시간이 지남에 따라 컴포넌트의 온도값은 각각 22.4 °C와 57.5 °C로 수렴하였다.

본 연구에서 제안하는 ASAP와 유한차분법을 이용하여 해석한 결과를 비교하여 Fig. 8과 같이 시뮬레이션 결과를 비교하여 도시하였다. 그래프에서 원형 표식은 17번 컴포넌트 위치에서 시간별 온도 변화를 유한차분법으로 계산한 값이고, 실선으로 표시된 값은 17번 컴포넌트의 온도변화를 ASAP를 이용하여 모의한 결과이다. 86번 컴포넌트는 유한차분법으로 계산한 값을 마름모 표식으로 도시하였으며, ASAP로 계산한 결과는 점선으로 도시하였다. 이처럼 ASAP는 유한차분법으로 해석할 때와 동일한 모의 결과를 확인할 수 있었다.

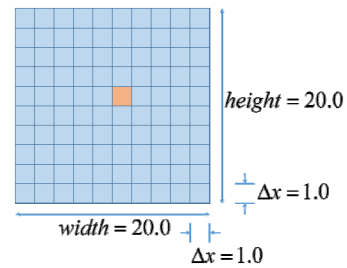


Fig. 9 Problem configuration for fluctuation phenomenon

3. 파동 방정식

가. 문제 정의

파동 방정식의 풀이 예로 Fig. 9와 같이 문제를 정의하였다. 파동전파계수는 0.5로 설정하였으며, 가로, 세로 길이 20인 정사각형으로 문제영역을 정의하였다. 격자크기는 $\Delta x = \Delta y = 1.0$ 이 되도록 하였으며, 단위시간 $\Delta t = 1.0$ 으로 설정하였다. 파동문제는 동적인 문제이기 때문에 한 시점의 u 값을 계산하기

위해서 그 이전의 값이 필요하다. 따라서 $t=0, t=-1$ 시점에서의 초기값을 설정해야 하는데, 이번 예제에서는 전체 영역의 값을 모두 0.0으로 설정하고, Fig. 9에서 주황색으로 표시한 부분의 값만 $t=0$ 일 때는 0.0, $t=-1$ 일 때는 10.0으로 설정하였다.

문제영역의 가장자리를 이루는 컴포넌트들은 경계영역으로 설정하고 (15)와 같이 시간 t 와 관계없이 항상 0.0으로 설정하였다.

$$\begin{aligned} u(0, y, t) &= u(20.0, y, t) = 0 \\ u(x, 0, t) &= u(x, 20.0, t) = 0 \end{aligned} \tag{15}$$

나. 적용 결과

구현한 컴포넌트와 초기조건 및 경계조건을 이용하여 시뮬레이션이 수행하고, 그 과정을 Fig. 10과 같이 도시하였다. ASAP를 이용하여 파동을 모의하여 시간대별로 캡처한 모습으로 문제 영역의 중심부에 위치한 251번 컴포넌트에서 값이 진동하는 결과를 확인할 수 있다. 전체 도메인에 대해서 주어진 파동방정식과 초기조건에 따라 각 컴포넌트의 값이 진동하며 전파되어 나가는 현상을 시각적으로 관찰할 수 있다.

본 연구에서 제한한 ASAP와 유한차분법 (FDM)으로 파동문제를 모의한 결과를 비교하여 Fig. 11에 도시하였다. 실선으로 표시한 값은 하단부 중심에 위치한 51번 컴포넌트를 ASAP로 해석한 것이며, 원으로 도시한 값은 유한차분법으

로 해석한 결과이다. 마찬가지로 문제의 중심부에 위치한 251번 컴포넌트의 ASAP 해석결과는 점선으로, 유한차분법 해석 결과는 마름모로 표시하였다. 두 방법을 비교한 결과는 유효숫자 6자리 내에서 정확히 일치하는 것으로 모의되었다.

4. 고찰

본 연구에서 2계 편미분 방정식을 풀이할 수 있는 컴포넌트를 개발하고, 개발한 컴포넌트를 이용하여 평형, 확산, 파동 방정식을 풀이하였다. 사용자가 연구목적에 따라 작성해야 하는 코드는 제시한 예제코드가 대부분이며, 많은 부분 공통적인 코드로 이루어져 있어 본 연구 결과를 이용하여 다른 연구에 활용하는 데 큰 어려움이 없을 것으로 판단된다. 특히 2계 편미분 방정식은 많은 연구의 기본 지배방정식이므로, 본 연구에서 개발한 컴포넌트를 활용하여 변수를 추가하거나 계수를 변경하는 정도의 작업으로 컴포넌트 개발이 가능하다. 개발된 컴포넌트는 간단한 스크립트를 통하여 문제를 구성할 수 있으며, 구성된 시뮬레이션 환경은 Fig. 6과 같이 도형, 색상, 그래프 등을 통하여 확인할 수 있다. 이러한 일련의 과정을 통하여 프로그래밍에 익숙하지 않은 연구자도 시뮬레이션 모델을 개발할 수 있을 것으로 기대한다.

또한 Lee et al. (2012)가 주장한 바와 같이 객체지향 패러다임을 지향하는 많은 연구들이 기술적으로는 컴포넌트를 이용하나, 실질적으로는 여전히 절차적 모델링 기법에 의존하고 있다. 농업시스템응용플랫폼은 컴포넌트를 개발한 이후

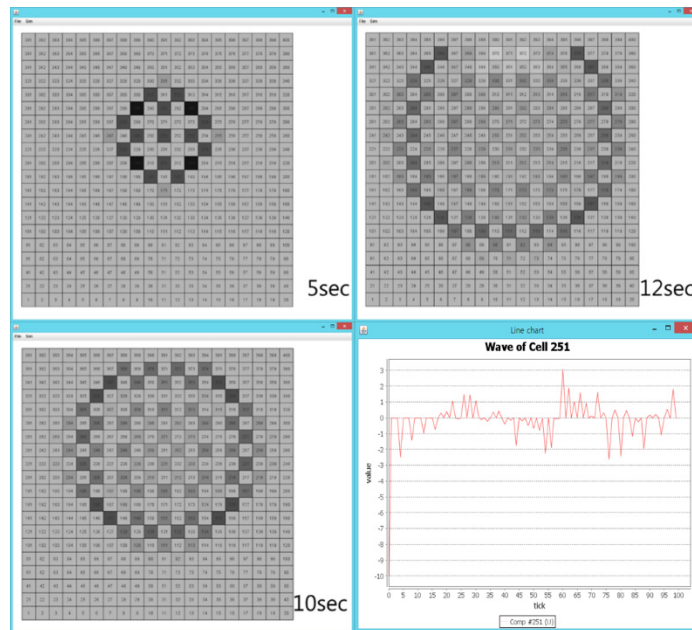


Fig. 10 Screenshots of wave problem simulation by ASAP

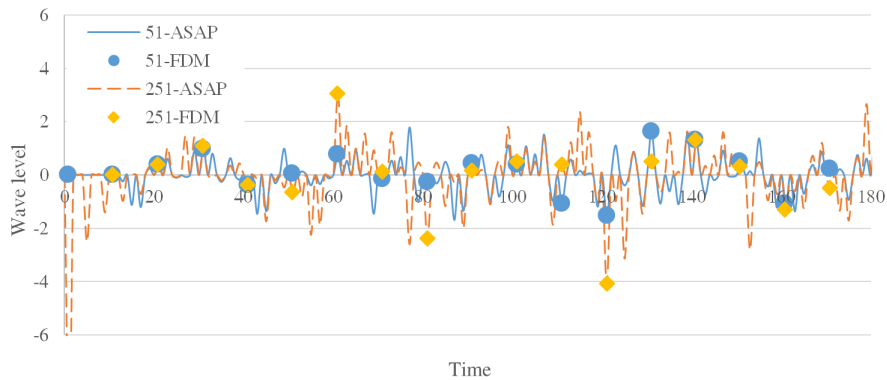


Fig. 11 Comparison of ASAP (this study) and FDM in wave simulation

에는 연결된 컴포넌트끼리 정해진 규칙에 따라 시뮬레이션이 구동되므로, 사용자가 시스템 구성에 관여할 필요 없이 개별 구성 요소에 역량을 집중할 수 있도록 도와준다. 본 연구에서 평행방정식과 확산방정식은 문제 구성이 동일하고, 개발한 컴포넌트만 대체하여 시뮬레이션을 수행하였다. 이러한 사고의 변환이 시뮬레이션 모델을 쉽게 개발하고, 기존 모델을 결합하여 더 큰 시스템을 만드는 데 일조하기를 기대한다.

V. 결 론

공학문제를 구성하는 지배방정식의 대표적인 형태인 2계 선형편미방정식을 단계적 학습을 통하여 모델링할 수 있도록 농업시스템응용플랫폼을 활용하였다. 대표적 형태의 편미방정식에 대하여 기본요소인 컴포넌트를 개발하고, 이를 결합하여 시뮬레이션을 수행하는 과정을 제시하였다. 일반적인 수치해석 방법론은 절차지향적이거나, 기술적 의미에서 객체지향적이어서 연구자가 모델을 만들어가는 과정에서 점진적으로 결과를 확인하기 어렵다는 한계를 갖고 있었다. 본 연구에서는 농업시스템응용플랫폼을 이용하여 2계 편미방정식의 평행방정식, 확산방정식, 파동방정식 풀이를 위한 컴포넌트를 구현하고, 개발한 컴포넌트를 결합하여 문제를 해석함으로써, 점진적으로 모델링을 수행하고 실시간으로 변화하는 해석결과를 시각적으로 확인하였다.

평행방정식은 열확산이 진행되어 정상상태에 도달했을 때의 열분포를 해석하는 문제에 대해 적용하였으며, 확산방정식은 열확산이 진행되는 과정을 모델링하였다. 전자는 동적으로 변화하는 결과가 시간항이 무시된 결과로써, 단순히 속성값의 수렴속도를 의미할 뿐이며, 후자는 시간항이 고려된 결과로 속성값의 동적해석 결과를 의미한다는 차이가 있다. 두 해석 결과 모두 기존의 유한차분해석의 결과와 동일하

였다. 파동방정식은 임의의 파동문제에 적용하여 해석하고, 유한차분법과 비교하여 같은 결과를 보임을 확인하였다.

각각의 문제에 대한 컴포넌트는 농업시스템응용플랫폼에서 제공하는 `preProcess()`, `process()` 메소드를 코딩하여 구현하였으며, 재현성을 위하여 주요 코드를 기술하였다. 본 연구를 통해 개발한 2계 편미방정식 컴포넌트는 다양한 자연현상을 모의하는 데 기초자료로 활용할 수 있으며, 그 계산과정과 결과를 그래프와 그림, 데이터 파일로 제공하므로, 다양한 형태로 변환하여 이용할 수 있다.

향후 사용자 인터페이스를 개선하여 프로그래밍에 익숙하지 않은 연구자가 드래그 앤 드롭 방식으로 수치해석 시뮬레이션을 수행할 수 있을 것으로 예상된다. 이를 위해 다양한 실용적 문제를 해석할 필요가 있고, 사용자 편의성 개선을 위한 농업시스템응용플랫폼의 전처리 및 후처리 기능의 개선이 요구된다. 시뮬레이션을 통하여 현상을 해석하는 과정이 원활해지면, 최적화 기법을 접목하여 설계 및 계획에 이용 가능한 의사지원도구로 발전할 수 있을 것으로 기대된다.

사 사

본 논문은 2015년 국민안전처 연구개발사업의 연구비지원(MPSS-소방안전_2015-76)에 의해 수행되었습니다. 이에 감사드립니다.

REFERENCES

1. Choi, W., and H. J. Kim. 2015. Development of Three Dimensional Chloride Ion Penetration Model Based on Finite Element Method. *Journal of The Korean Society of Agricultural Engineers* 57(5):43-49. (in Korean)

2. Courant, R., K. Friedrichs, and H. Lewy. 1967. On the partial difference equations of mathematical physics. *IBM journal of Research and Development* 11(2):215-234.
3. Jeon, J. B., S. S. Yoon, J. S. Park, and H. R. Lee. 2012. Probability Analysis of Plane Strain Element using Boundary Element Method. *Journal of The Korean Society of Agricultural Engineers* 54(4):39-46. (in Korean)
4. Jun, S. M., M. S. Kang, I. Song, J.-H. Song, J. Park, and W. Kee. 2014. Development of agricultural reservoir water supply simulation system. *Journal of Korean Society of Rural Planning* 20(2):103-114. (in Korean)
5. Kang, M. S., J. H. Song, J. H. Park, I. H. Song, W. S. Ki, and J. T. Kim. 2015. Introduction of COMFARM: Component Based Modeling Framework for Rural Water Cycle Analysis. *Rural resource* 57(1):55-62. (in Korean)
6. Kim, T. G. 2014. Development of Agricultural Systems Application Platform enhancing Connection, Communication, and Cloning. Ph.D. Diss., Seoul, South Korea: Seoul National University. (in Korean)
7. 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)
8. Kim, T. G., and J. J. Lee. 2009. Development of a Graphical Modeler for Manipulating Series Data Based on Object-Oriented Technique. *Journal of The Korean Society of Agricultural Engineers* 51(2):43-49. (in Korean)
9. Kim, T. G., J. J. Lee, and K. Suh. 2014a. Development of a Truss Structure Analysis Model based on Cellular Automata and Object-oriented Simulation Environment. *Journal of The Korean Society of Agricultural Engineers* 56(3):1-9. (in Korean)
10. Kim, T. G., S. Y. Lee, H. J. Yi, J. J. Lee, and K. Suh. 2014b. Development of GASS2 through Improving Inter-component Connection and Communication Modules. *Journal of The Korean Society of Agricultural Engineers* 56(5):29-36. (in Korean)
11. Kreyszig, E. 1988. *Advanced engineering mathematics*. John Wiley & Sons.
12. Park, J. H., M. S. Kang, J. H. Song, S. M. Jun, K. U. Kim, and J. H. Ryu. 2015. Web-Based Data Processing and Model Linkage Techniques for Agricultural Water-Resource Analysis. *Journal of The Korean Society of Agricultural Engineers* 57(5):101-111. (in Korean)
13. Yeo, U. H., Y. S. Jo, K. S. Kwon, T. H. Ha, S. J. Park, R. W. Kim, S. Y. Lee, S. N. Lee, I. B. Lee, and I. H. Seo. 2015. Analysis on Ventilation Efficiency of Standard Duck House using Computational Fluid Dynamics. *Journal of The Korean Society of Agricultural Engineers* 57(5):51-66. (in Korean)
14. Yi, H. J. 2003. Simplifying complexity and emerging self-organization of integrated agricultural system models. Ph.D. Diss., Seoul, South Korea: Seoul National University.