

CEMTool에서 유한요소법을 이용한 PDE Solver구현에 관하여

안춘기, 이태일, 한수희, 이영삼, 권옥현
서울대학교 전기컴퓨터공학부

On the Implementation of PDE Solver using Finite Element Method in CEMTool

Choon Ki Ahn, Tae Il Lee, Soo Hee Han, Young Sam Lee, and Wook Hyun Kwon
School of Electrical Engineering & Computer Science, Seoul National University.

Abstract - 유한요소법(FEM)은 많은 공학문제를 해결하는 가장 중요한 방법 중 하나로 인식되고 있다. 본 논문에서는 자동제어 및 신호처리 문제해결에 효율적이며 강력한 수치해석 패키지인 CEMTool환경에서 유한요소법을 이용하여 일반적인 편미분방정식 Solver 구현에 관한사항을 논의하고자 한다. 기본적으로 영역정보 및 노드수 등의 정보를 입력받아 각 노드의 정보를 출력하는 Mesh 함수를 구현하며, 생성된 요소방정식들을 조립하는 Assemble 함수를 작성한 뒤, Boundary 함수를 통해 경계조건을 적용시킨 후 선형행렬 방정식을 풀어 전체노드의 값을 찾아내는 Solve 함수를 구현하는 과정을 알아본다. 구현된 FEM Solver의 전체적인 구조를 통해 구현시 고려해야 할 사항을 논의하며 기본적인 편미분방정식의 예제를 통해 FEM PDE Solver의 동작과정을 검증할 것이다.

1. 서 론

1960년대부터 본격적으로 연구가 된 유한요소법(FEM: Finite Element Method)[1]을 기반으로 하는 해석과 설계는 여러 가지 세계적인 패키지들이 활용되어 매우 급속히 발전하였다. 특히 산업체에서 생산에 관련된 자동차, 가전, 항공, 조선, 전자분야 등에서 사용되는 부품이나 소재를 제작하는데 있어서 발생하는 여러 현상에 대한 해석의 핵심 알고리즘인 FEM은 현재 대부분의 공학문제 해결에 적용되는 매우 중요한 방법론이 되었다. 국외의 연구사례를 살펴보면, 미항공우주국에서 개발을 시작하여 30년간 개발된 세계적인 구조/진동/부품소재 해석 패키지인 NASTRAN을 들 수 있다. 이 제품은 기능별로 모듈화가 되어있고 최상의 신뢰성을 보장하며 선형/비선형 해석이 특징이다. 또한 과학기술에 전반적으로 이용되는 패키지인 MATLAB은 세계적으로 가장 유명한 패키지로 행렬 구조 및 미적분 수치해석코어로 구성되어 FEM 해석에 적합하며 FEM 라이브러리와 PDE 라이브러리가 구성되어있다. 국내의 연구사례를 살펴보면, 전자장 해석 등의 특수해석을 위한 전자장 해석나라와 소프트웨어등이 개발되었지만 일반적인 해석을 위한 국산 FEM 패키지는 전무한 것이 사실이다.

FEM의 특정문제 풀이를 위한 전용 프로그램의 경우는, 사실상 범용 프로그램 개발과는 달리 그 개발 규모나 위험부담이 상당히 낮기 때문에 국내에서 일부 개발이 가능하지만, FEM의 일반적인 문제를 해결하면서 다른 공학 문제들 (행렬, 미분방정식, 라플라스 변환, 주파수영역 해석 등)까지 포함하여 함께 처리할 수 있는 범용 프로그램은 긴 개발 기간과 많은 개발 비용이 필요하여 국내에서 뿐만 아니라, 세계적으로도 몇 개의 국가를 제외하면 개발된 제품이 없으며 개발 능력 또한 갖추지 못하고 있다. 하지만 과학기술의 기반기술 확충과 기술력의 한 단계 도약을 위해서는 반드시 선행되어야 하는 중요한 과제로 많은 과학 기술자들이 인정하고 있는 실정이다.

본 논문에서는, 국내에 이미 많이 보급되어있는 CEMTool 패키지[2,3]에 편미분 방정식에 관련된 공학문제를 해결할 수 있도록 FEM기법을 이용하여 편미분 방정식을 풀 수 있는 여러 가지 함수구현 과정을 논의해볼 것이다. 본 논문의 2장에서는 과거에 구현되었던 유한요소법 라이브러리의 단점을 언급하며, 3장에서는 구현된 FEM PDE Solver의 구조 및 구현과정을 살펴보고, 간단한 예제를 들어 구현된 PDE Solver의 기능을 알아본다. 마지막으로 4장에서는 결론을 내린다.

2. CEMTool FEM 라이브러리의 단점

과거에 CEMTool에 구현되었던 유한요소법 라이브러리는 서로 여러개의 파일을 주고받는 구조로 되어있다 [4]. 편리한 그래픽 인터페이스를 제공하며 강력한 후처리 성능을 자랑하는 유한요소 라이브러리였지만 가장 큰 단점은 CEMTool에서 유한요소 데이터를 쉽게 접근하지 못한다는 단점이 있었다. 유한요소 데이터를 접근하기 위해 할 수 있는 방법은 2가지로 나뉘는데, 첫째로 유한요소 전처리기를 다시 실행하여 전처리 작업을 반복하는 것이다. 둘째로, FEM Property Setup이라는 편리한 프로그램을 이용하여 영역별 물질정보의 값을 변경하는 방법이다. 하지만 FEM Property Setup을 이용하면 영역단위로만 접근가능하며 노드 및 요소별로 접근은 불가능하기 때문에 상당히 많은 수의 데이터를 다루는 유한요소해석에서 데이터의 대부분정정은 문제가 아닐 수 없다. 또한 CEMTool에는 여러 종류의 강력한 내부 함수가 존재하지만 과거의 유한요소 라이브러리는 파일로 데이터를 주고받는 형태이므로 이를 전혀 사용할 수 있는 구조가 아니었다. 추가로, 과거에 구현되었던 FEM 라이브러리는 전자기장장용의 해석틀로 요소행렬을 계산할 때 수치적분계산이 이미 코드화되어 일반적인 편미분방정식을 푸는데 한계가 있었다. 따라서 본 논문에서는 과거에 구현되었던 유한요소법 라이브러리의 문제점을 극복하여 CEMTool 내부 기능을 모두 사용할 수 있는 새로운 형태의 FEM 라이브러리의 구현과정을 자세히 알아 볼 것이다.

3. 유한요소법을 이용한 PDE Solver 구현

3.1 Mesh 함수

과거에 구현되었던 유한요소법 라이브러리의 전처리기능은 해석모델의 형상, 매질특성, 경계조건 및 요소분할 등의 정보를 입력받아 상당히 복잡한 6개의 파일을 생성해낸다. 하지만 본 논문에서 구현될 Mesh함수는 편미분방정식의 해를 구할 수 있도록 기본적인 정보만을 입력받아 메모리의 낭비를 방지할 것이다. 자동 mesh를 생성하기 위해 필요한 정보는 PDE를 해결할 영역의 영역정보와 그 영역을 요소로 나누기 위한 요소수이다. 이러한 정보를 입력 받아 Mesh함수는 노드의 정보를 출력하고 이에 따른 각 노드의 좌표정보를 출력한다. 또한 입력된 노드수의 정보로 초기화된 강성행렬의 정보도

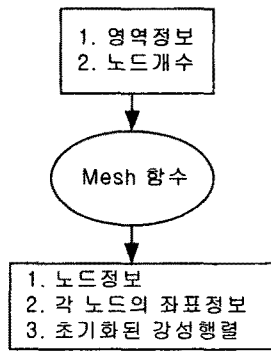


그림 1 Mesh함수의 입출력

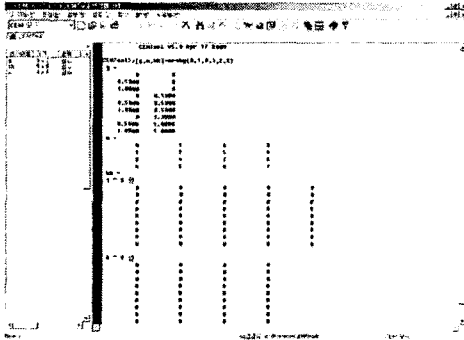


그림 2 CEMTool에 구현된 Mesh 함수

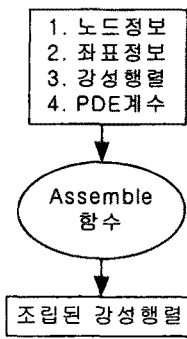


그림 3 Assemble함수의 입출력

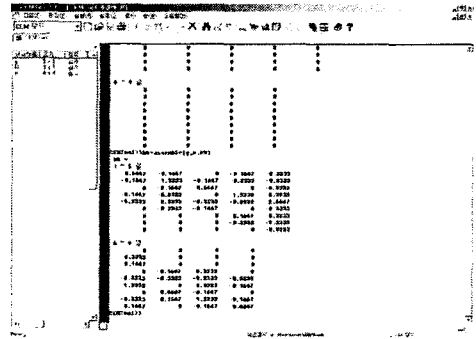


그림 4 CEMTool에 구현된 Assemble함수

출력한다. 과거에 구현된 유한요소 전처기리는 노드수 및 요소수를 정확히 제어하기 불가능했지만 구현된 Mesh함수는 사용자가 나누어질 요소수를 직접 입력하여 그에 따른 정보를 얻어내는 방식이므로 정확한 요소수의 제어가 가능하다고 볼 수 있다. <그림1>은 Mesh함수의 입출력구조를 나타내며, <그림2>는 CEMTool에 구현된 Mesh 함수의 동작을 보여준다. <그림2>에서 보듯이 Mesh함수는 메모리 절약을 위해 PDE 해결을 위한 최소한의 정보만을 출력한다.

3.2 Assemble 함수

과거에 구현되었던 CEMTool 유한요소법 라이브러리는 FEM Solver 내부에서 6개의 유한요소 정보파일의 내용을 읽어 이를 조립한 후 경계조건을 적용한 뒤 전체적인 강성행렬에 NDM(Nested Dissection Method)를 적용하여 고속의 Solver를 실현하였다. 고속의 NDM Solver가 내장되어 있었지만 강성행렬의 구조 및 내용을 파악하기 힘들었을 뿐만 아니라 조립과정에서 얻어지는 여러 정보에 접근하기가 불가능했다. 이러한 단점을 극복하기 위해 명령어 방식의 FEM Solving기법을 도입하여 조립된 강성행렬의 구조 및 내용에 대한 직접적인 정보를 얻을 수 있을 뿐만 아니라 필요하다면 이를 수정할 수도 있다. CEMTool PDE Solver에 내장되어있는 Assemble함수는 Mesh함수에 의해 생성된 노드정보, 좌표정보, 강성행렬 뿐만 아니라 편미분방정식의 계수 정보를 추가로 입력받아 요소별로 강성행렬을 생성한 후 이를 조립하는 방식을 사용한다. Mesh함수에 의해 생성된 강성행렬은 모두 0으로 초기화 되어있으나 Assemble함수에 의해 요소별로 강성행렬이 생성되어 for루프를 통해 조립하는 과정이 이루어진다. <그림3>은 Assemble함수의 입출력 구조를 나타내며 <그림4>는 Mesh함수 적용후 얻은 정보를 이용하여 Assemble함수를 실제 적용시키는 모습을 보여준다.

<그림4>에서 보듯이 Assemble함수 적용 후 얻은 조립된 강성행렬은 Sparse Matrix 형태로 Zero terms가 많이 존재하므로 이러한 특성을 이용해 FEM Solver를 구성하면 고속의 Solver를 구현할 수 있다. 이에 관련된 내용은 Solver를 다루는 부분에서 다시 언급하도록 하겠다.

3.3 Boundary 함수

CEMTool PDE Solver에 내장되어 있는 Boundary 함수는 경계조건을 적용하여 강성행렬을 비정칙으로 바꾸어주는 역할을 한다. 일반적으로 요소행렬을 조립하여 얻은 Sparse Matrix형태의 강성행렬은 정칙행렬로 역행렬이 존재하지 않으므로 유일한 선형방정식의 해를 보장할 수 없게 된다. CEMTool PDE Solver에 내장되어 있는 Boundary함수는 요소행렬을 모두 조립하여 얻은 강성행렬에서 각 노드에 해당되는 경계조건을 1로 놓고 하중벡터에 경계치를 넣는 방법을 사용하고 있다. 이를 구현하기 위해서 경계조건에 해당되는 노드정보를 포함하는 bc라는 배열이 필요하며, 각 배열에 따른 경계조건의 값을 갖고 있는 bcval이라는 배열이 추가적으로 필요하다. 과거에 구현되었던 유한요소법 라이브러리는 경계조건을 적용하는 부분이 FEM Solver에 모두 포함되어있었기 때문에 그 구조 및 기능을 알 수 없었지만 개발된 CEMTool PDE Solver는 Boundary함수의 적용 전과 후의 강성행렬 및 하중 벡터의 정보를 바로 알 수 있을 뿐만 아니라 필요하다면 경계조건을 하중벡터를 통해 바로 수정가능 하기 때문에 사용자가 유한요소법을 다룰 때 더욱 세밀한 자유도를 부여하고 있다. 하중벡터는 초기에 모두 0으로 초기화 되어있었지만 경계조건을 적용시킨 후에는 0벡터가 아니므로 선형방정식의 유일한 해를 보장받을 수 있다. Boundary함수는 조립된 강성행렬의 정보와 경계조건에 정보를 입력받아 최종적인 강성행렬과 하중벡터에 관한 정보를 출력하여 최종적인 선

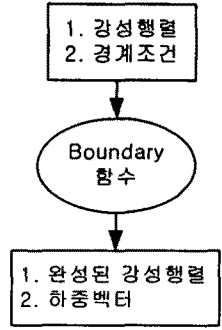


그림 5 Boundary 함수의 입출력

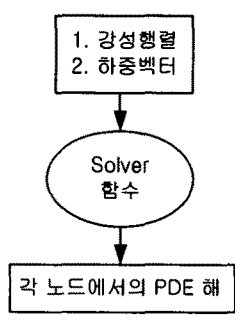


그림 7 Solver 함수의 입출력

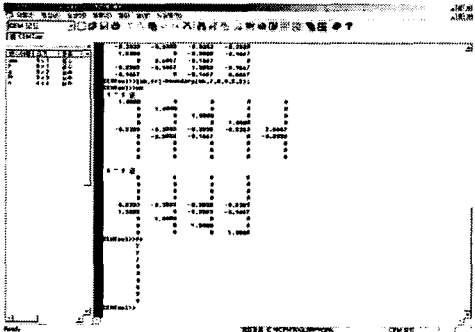


그림 6 CEMTool에 구현된 Boundary 함수

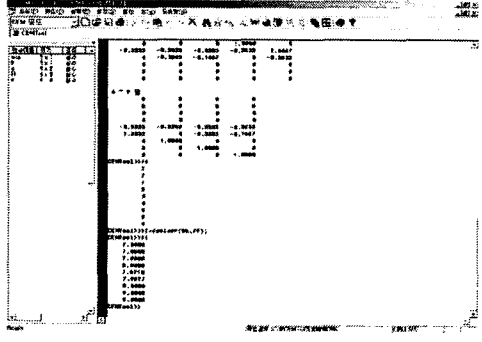


그림 8 CEMTool에 구현된 Solver 함수

형방정식의 정보를 생성해 낸다. <그림5>는 Boundary 함수의 입출력 구조를 나타내며 <그림6>은 CEMTool에 구현된 Boundary 함수의 동작을 보여주고 있다. <그림6>에서 보듯이 구현된 Boundary 함수를 통해 정적이었던 강성행렬이 비정칙으로 변경되었으며 필요하다면 CEMTool의 변수보기창을 통해 하중벡터의 값을 부분적으로 수정할 수도 있다.

3.4 Solver 함수

일반적으로 유한요소해석 과정에서 얻어지는 강성행렬은 Sparse Matrix 형태로 행렬의 특수한 구조를 이용하면 고속의 Solver를 구현할 수 있으며 과거에 구현되었던 FEM Solver는 직접법의 일종인 NDM을 채용하여 MATLAB의 속도를 증가하는 성능을 보장하였다. 과거에 구현되었던 FEM 라이브러리는 Sparse Matrix로 구성된 선형연립방정식에 대한 효율적인 메모리저장을 위해 non-zero storage 방식을 채택하고 있으며 선형방정식의 신속한 처리를 위하여 NDM기법을 채용하였다. 하지만 이전에 구현되었던 FEM Solver는 행렬의 크기 및 성과는 관계없이 하나의 Solving기법에 의존하였으나 FEM PDE Solver는 현존하는 Sparse Matrix Solving 기법뿐만 아니라 가우스 소거법까지 여러 가지 Solving 기법들을 채용할 수 있어 행렬의 크기나 형태에 맞게 원하는 성능을 보장하는 Solver를 선택할 수 있어 사용자의 자유도를 극대화하였다. CEMTool PDE Solver에 내장되어 있는 Solver함수는 Boundary 함수의 출력인 강성행렬 및 하중벡터의 정보를 이용하여 각 노드에서의 편미분방정식의 해를 출력한다. <그림7>은 Solver함수의 입출력 구조를 나타내며 <그림8>은 CEMTool에서 Solver함수의 동작을 보여주고 있다. <그림8>에서 보듯이 출력인 각 노드에서의 해는 벡터형태로 보기 어려운 구조로 되어있지만 다른 추가적인 함수를 이용하면 기하학적 구조를 고려하여 해

를 재배치 한 결과를 얻을 수 있어 편미분방정식의 해를 해석하는데 무척 편리하지만 지면상 추가함수를 언급하는 것은 생략하겠다.

4. 결 론

본 논문에서는 국산 수치해석 패키지인 CEMTool에 유한요소법을 이용하여 PDE Solver를 구현하는 과정과 특징을 자세히 살펴보았다. 과거에 구현되었던 FEM 라이브러리의 서로 파일을 주고받는 구조에서 벗어나 유한요소법의 데이터를 CEMTool의 명령어 모드 상에서 처리하여 일반화된 편미분방정식을 다룰 수 있게 해주기 때문에 CEMTool의 강력한 기능을 한 단계 높여주는 결과를 가져왔다. 더욱 일반화된 PDE 포함, 다양한 경계조건 포함, 후처리기에서의 CEMTool 그래프 연동, 속도 개선 및 안정도와 신뢰성의 철저한 검증등의 과제가 남아있지만 과학기술 전반의 범용성이 높아진 CEMTool의 개발 및 보급은 국내 과학기술 발전에 크게 이바지할 것이라고 본다.

[참 고 문 헌]

[1] Y.W. Kwon and H.C. Bang, "The Finite Element Method using MATLAB". CRC Press, 2000.
 [2] 한수희, 최성규, 이관호, 이준석, 권옥현, "CEMTool의 구현 및 응용", in Proceedings of the 14th KACC, 1999.
 [3] 권옥현, 김기백, 최성규, 김형진, 한수희, "실시간 제어 시스템 설계 지원 기술동향과 실제 : CEMTool을 중심으로", ICASE Magazine, vol. 5, pp. 18-26, 1999.
 [4] 안춘기, 한수희, 이영삼, 권옥현, "CEMTool에서 유한요소법 라이브러리 구현에 관하여", 대한기계학회 추계학술대회 논문집, 2002.