Generalized Command Mode Finite Element Method Toolbox in CEMTool

Choon Ki Ahn* and Wook Hyun Kwon

*School of Electrical Engineering & Computer Science, Seoul National University, Seoul , 151-742, Korea
(Tel: +82-2-880-7314; http://cisl.snu.ac.kr; Email:hironaka@cisl.snu.ac.kr)

Abstract: CEMTool is a command style design and analyzing package for scientific and technological algorithm and a matrix based computation language. In this paper, we present a compiler based approach to the implementation of the command mode generalized PDE solver in CEMTool. In contrast to the existing MATLAB PDE Toolbox, our proposed FEM package can deal with the combination of the reserved words such as "laplace" and "convect". Also, we can assign the border lines and the boundary conditions in a very easy way. With the introduction of the lexical analyzer and the parser, our FEM toolbox can handle the general boundary condition and the various PDEs represented by the combination of equations. That is why we need not classify PDE as elliptic, hyperbolic, parabolic equations. Consequently, with our new FEM toolbox, we can overcome some disadvantages of the existing MATLAB PDE Toolbox.

Keywords: CEMTool, FEM package, Scientific Computing, Compiler based technique, General PDE solver

1. Introduction

Finite element method (FEM) provides a greater flexibility to model complex geometries than finite difference and finite volume methods do. It has been widely used in solving structural, mechanical, heat transfer, and fluid dynamics problems as well as problems of other disciplines. The advancement in computer technology enables us to solve even larger system of equations, to formulate and assemble the discrete approximation, and to display the results quickly and convienently. This has also helped the finite element method become a powerful tool. There are many kinds of the specialized FEM package for electromagnetics, structural mechanics, heat transfer, diffusion, and something like that. Among the numerical general-purpose packages for scientific computing, MATLAB is one of the well-known package for science and engineering that performs mathematical and engineering computation. As a tool to analyze the solution of PDE in MATLAB, the MATLAB PDE Toolbox [1] contains basic tools for the study and solution of PDE in two space dimensions (2-D) and time, using the finite element method. Its command line functions and graphical user interface can be used for mathematical modeling of PDE in some ranges of engineering and science applications. However, MATLAB PDE Toolbox in command-mode has some disadvantages as follows: (1) It can deal with only constant coefficients. (2) It is unclear and difficult to assign the border lines and the boundary conditions. (3) In order to use the FEM solver functions of PDE Toolbox, we have to classify PDE as elliptic, parabolic, and hyperbolic equations.

As another powerful numerical general-purpose package, CEMTool integrates mathematical computing, visualization, and a powerful high-level language to provide a flexible environment for technical computing [2], [3]. The powerful architecture makes it easy to use CEMTool and its companion toolboxes to explore data, create algorithms, and create custom tools that provide early insights and competitive advantages [4], [5], [6], [7], [8]. In this paper, we present a new FEM toolbox which overcomes the some existing dis-

advantages of MATLAB PDE Toolbox. Our FEM package in CEMTool contains the lexical analyzer and the parser to deal with the general combination of reserved words such as "sin", "cos", "convect", and the like. That is why we can solve the PDE represented by the combination of some reserved words and mathematical operators. In addition, with the introduction of the compiler based technique, we need not to classify the PDE we want to solve as elliptic or parabolic or hyperbolic equations. Compared to MATLAB PDE Toolbox, this is a very good feature because MATLAB PDE Toolox can deal with only constant coefficients. Since the border functions and the boundary condition functions of our FEM package are based on the geometric concept, it is easy to assign the boundary region and its conditions.

The overall procedure for the FEM analysis in CEMTool is described in Section 2. In Section 3, the detailed structure and the important features of various functions of CEMTool FEM Toolbox are discussed. The conclusion is given in Section 4.

2. The Overall Procedure for the FEM Analysis in CEMTool FEM Toolbox

Basically, the border function of CEMTool FEM Toolbox requires some information on the boundary which is represented by the combination of the reserved words in x and y coordinates. After defining the boundary region, in order to generate the triangular mesh based on a Delaunay-Voronoi algorithm, we can employ the makemesh function of CEMTool FEM Toolbox by using the information on the maximum number of vertices as the only parameter of the makemesh function. For the assignment of the boundary conditions, the boundary condition function can be used with the information of the identification numbers and the general boundary conditions represented by some reserved words of CEMTool FEM Toolbox. In order to construct the solution of the PDE we want to solve, we need the FEM solver funtion of CEMTool FEM Toolbox with the information of the PDE which is represented by the combination of the basic mathematical operator and the reserved words such

as "laplace" and "sin". Finally, the post-processing functions such as "meshplot", "equi-potential plot", and "3D plot" can be used to visualize the solution of the PDE generated by CEMTool FEM Toolbox. The basic procedure for the FEM analysis in CEMTool FEM Toolbox can be described in Fig 1.
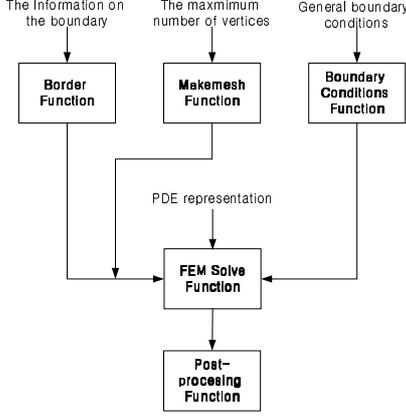
Fig. 1. The basic procedure for the FEM analysis in CEM-Tool FEM Toolbox

3. The Detailed Structure and Functions of CEMTool FEM Toolbox

3.1. Border Function

In this subsection, we will discuss the structure of the border function and its features. The first feature of our border function is that it is based on an equation based approach. Since the border function has the lexical analyzer and the parser, it is possible to use the reserved words such as "sin" and "tan" to describe the boundary region. The prototype of the border function can be described as follows:

$$bound(in, t_m, t_M, nb, x_{eq}, y_{eq}) \tag{1}$$

where the first parameter in is the identification number of the border, t_m and t_M stand for the minimum time and the maximum time, respectively. nb is the number of division, x_{eq} and y_{eq} describe the boundary in x and y coordinates, respectively. First of all, as seen in Fig 2, the border function has the lexical analyzer and the parser in order to deal with various reserved words of CEMTool FEM Toolbox. With the information on equations in x and y coordinates, the lexical analyzer generates the parsing tree. The parser can evaluate the commands or the reserved words at each node of the parsing tree. With the time information represented as t_m, t_M, and nb, the border function generates a sequence of points representing the boundary in x and y coordinates. If we use some equations representing the first part of the boundary region with the time parameter "t" in x and y coordinates as

$$bound(1, 0, 1, 20, "t", "0") \tag{2}$$

then we can assign the line from $(0,0)$ to $(1,0)$ with the division number 20 as the identification number 1. The
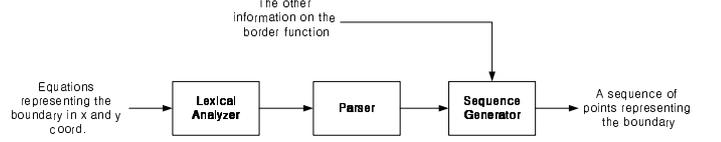
Fig. 2. The structure of the border function
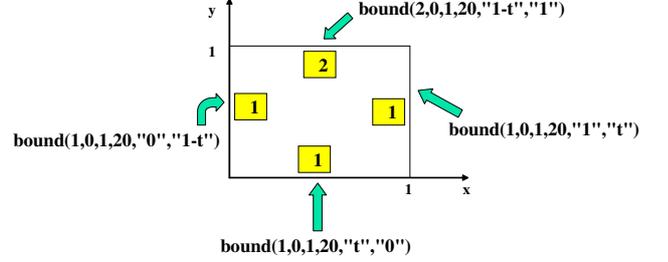
Fig. 3. The description of rectangular region with the identification numbers 1 and 2

rectangular region represented as

$$bound(1, 0, 1, 20, "t", "0");$$
$$bound(1, 0, 1, 20, "1", "t");$$
$$bound(2, 0, 1, 20, "1 - t", "1"); \tag{3}$$
$$bound(1, 0, 1, 20, "0", "1 - t");$$

can be described in Fig 3 with the identification numbers 1 and 2. With the border function of CEMTool FEM Toolbox, we can describe the circle region by using the command as

$$bound(1, 0, "2 * pi", 20, "cos(t)", "sin(t)") \tag{4}$$

In this case, we can represent the boundary region of the circle by using only one command as seen in Fig 4.
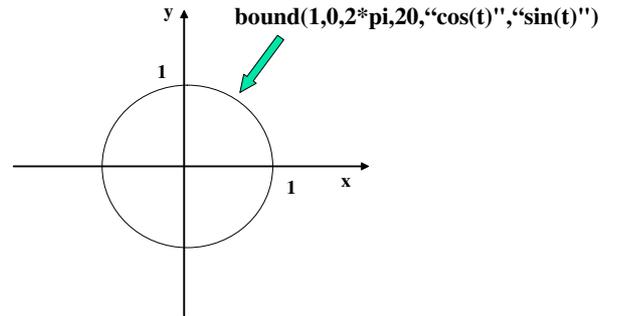
Fig. 4. The description of the circle region with the identification number 1

3.2. Automatic Mesh Generating Function

The automatic mesh generating function of CEMTool FEM Toolbox is based on a Delaunay-Voronoi algorithm in order to generate the triangular mesh. Since this function needs only one parameter to generate the mesh, it is very easy to use. The prototype of the mesh generation function can be described as follows:

$$makemesh(nb_{max}) \tag{5}$$
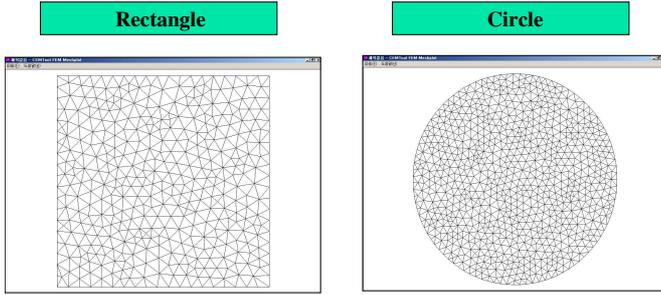
Fig. 5. The mesh generation result in the rectangular region and the circle region

Fig. 8. The boundary conditions of the rectangular region

where nb_{max} is the maximum number of elements to generate. The makemesh function requires the maximum number of elements as the parameter and generates a sequence of information representing the points of triangles and the rules of connection. If we want to generate the triangular mesh with the maximum number of elements 100 in the rectangular region and the circle region, respectively, we can use the following command

$$makemesh(100); \qquad (6)$$

and obtain the mesh generation result in Fig 5. In addition, for a specific region seen in Fig 6, we can generate the triangular mesh by using the appropriate combination of the border functions of CEMTool FEM Toolbox.
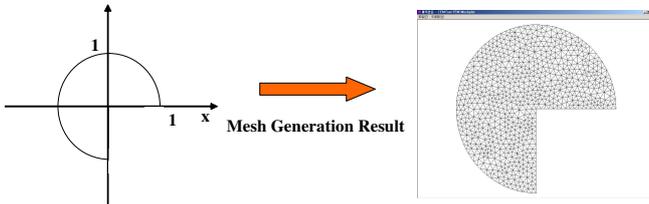
Fig. 6. The mesh generation result in a specific region
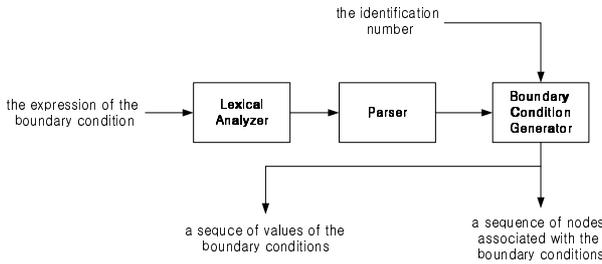
Fig. 7. The structure of the boundary function

3.3. Boundary Condition Function

In this subsection, we will discuss the boundary condition function of CEMTool FEM Toolbox. The boundary condition function of CEMTool FEM Toolbox has the lexical analyzer and the parser to handle the general boundary conditions represented by the combination of the reserved words of CEMTool FEM Toolbox. In this function, we can deal with the boundary conditions such as Neumann condition, Dirichlet condition, Mixed condition, and something
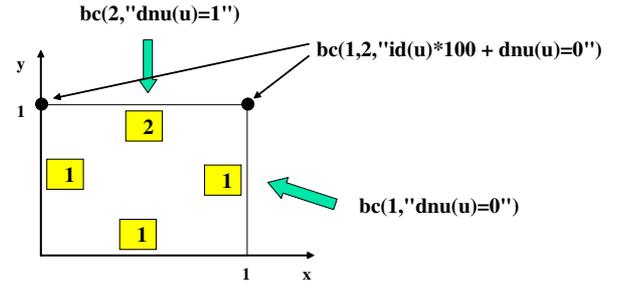
like that. In contrast to MATLAB PDE Toolbox, we can assign the boundary conditions for the parts of the boundary region specified by each identification number. As seen in Fig 7, in the lexical analyzer, the expression of the boundary condition is broken into tokens and recognized as symbols. The parser evaluates all the symbols of the parsing tree. With the identification number, the boundary condition generator makes an array which contains the values of the boundary conditions and another array which includes the information on nodes associated with the boundary conditions. The prototype of the boundary condition function can be described as follows:

$$bc(in, "id(u) * f + dnu(u) * g = h") \qquad (7)$$

where in stands for the identification number and "id(u)*f + dnu(u)*g=h" describes the expression of the boundary condition for each identification number. f, g, and h are the generic expressions. In order to assign the boundary conditions of the rectangular region defined by (3), we can use the following commands

$$bc(1, "dnu(u) = 0")$$
$$bc(2, "dnu(u) = 1");$$
$$bc(1, 2, "id(u) * 100 + dnu(u) = 0");$$

where the first command and the second command represent Neumann conditions for the regions specified by the identification numbers 1 and 2, respectively, and the final command describes the mixed condition at the corner of the regions specified by identification numbers 1 and 2. The diagram for this assignment is given in Fig 8.
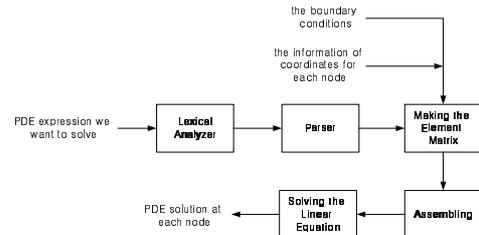
Fig. 9. The structure of the FEM solve function

3.4. FEM Solve Function

The FEM solve function of CEMTool PDE Toolbox can solve various kinds of PDE such as elliptic equations, hyperbolic
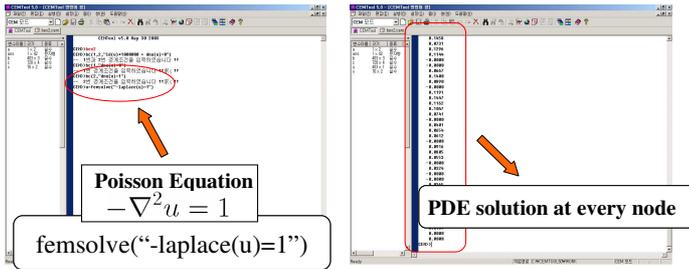
Fig. 10.   The solutions of Poisson equation at every node

Fig. 12.   The CEMTool FEM 3D plot

equations, and parabolic equations. The only one parameter of this function is the the PDE expression we want to solve. As seen in Fig 9, after making the parsing tree in the lexical analyzer, the parser evaluates the reserved words or the mathematical operators from the bottom of the parsing tree. With the information of coordinates for each node and the boundary conditions for the region specified by the border functions, the FEM solve function makes the element matrix and the force vector at every element. For the total system matrix and the total system vector, this function assembles the element matrices and the force vectors. Since the obtained system is a linear equation, we can calculate the solutions of PDE at every node by using the linear solver in Fig 10. The prototype of the FEM solver function can be described as follows:

$$femsolve("P.D.E.") \qquad (8)$$

where $P.D.E.$ represents the expression of the partial differential equation we want to solve. Consider the example on solving Poisson equation as

$$-\nabla^2 u = 1 \qquad (9)$$

If we assume that we use the rectangular boundary region defined by (3), we can employ the following command to solve the above Poisson equation

$$femsolve(" - laplace(u) = 1") \qquad (10)$$

where $laplace$ is the reserved word that stands for Laplacian.

Fig. 11.   The meshplot and the equiplot in a rectangular region

Since we employ the compiler technique in the FEM solve function, compared to MATLAB PDE Toolbox, we need not classify the PDE we want to solve as elliptic equations, parabolic equations, and hyperbolic equatioins.

3.5. Post-processing Functions

In order to visualize the solutions of PDE in a user-defined region, CEMTool FEM Toolbox has the powerful plot functions. The meshplot function shows the triangular mesh generated by the makemesh function without the information on height. Also, the equiplot function of CEMTool FEM Toolbox can visualize the equi-potential lines with the information on height in 2-dimensional coordinates. The mesh generation result and the equi-potential lines in the rectangular region defined by (3) are given in Fig 11. Another feature of the post-processing functions of CEMTool FEM Toolbox is the FEM 3D plot. The FEM 3D plot uses OpenGL library for the enhanced 3D graphic. This function shows the triangular meshes and the equi-potential lines in 3-dimensional coordinates with the information on height. In Fig 12, the 3D plot for Poisson equation (9) in the rectangular region defined by (3) is given. In addition, The FEM 3D plot has the various functions such as rotation, transition, zoom-in , zoom-out, and more.

4. Conclusion

In this paper, we present an equation based approach to the implementation of generalized PDE solver in CEMTool. Our FEM package can overcome the existing disadvantages of MATLAB PDE Toolbox with the introduction of the lexical analyzer and the parser. Due to the these techniques, in contrast to MATLAB PDE solver, we can deal with the combination of the reserved words. In order to perform the FEM analysis, we need not classify PDE as elliptic equations, hyperbolic equations, and parabolic equations. Also, it is easy to assign the border lines and the boundary conditions with our proposed FEM Toolbox.

References

[1]   Partial Differential Equation Toolbox : User's Guide," *The Mathworks, Inc.*, 2002.

[2]   S. H. Han, S. K. Choi, K. H. Lee, J. S. Lee, and W. H. Kwon, "OThe implementation and the application of CEMTool," *in proceedings of the 14th KACC*, 1999.

[3]   W. H. Kwon, K. B. Kim, S. K. Choi, H. J. Kim, and S. H. Han, "The technical trend and the practical application of the real-time control system design in CEMTool," *ICASE Magazine*, vol. 5, pp. 18-26, 1999.

[4]   W. H. Kwon, K. B. Kim, and S. G. Choi, "Control System Design and Analysis with CEMTool for Control

Education," *Transaction on Control, Automation, and Systems Engineering*, March, 1999.

[5] W. H. Kwon, K. B. Kim, and S. G. Choi, "Control System Design and Analysis with SIMTool and CEM-Tool for Control Education," *Transaction on Control, Automation, and Systems Engineering*, March, 2000.

[6] C. K. Ahn, S. H. Han, Y. S. Lee, and W. H. Kwon, "On the Implementation of GUI Based Finite Element Method Library in CEMTool," *In the proceedings of KIME Annual Conference*, 2002.

[7] C. K. Ahn, T. I. Lee, S. H. Han, Y. S. Lee, and W. H. Kwon, "On the Implementation of PDE Solver using Finite Element Method in CEMTool," *In the proceedings of KIEE Annual Conference*, 2003.

[8] C. K. Ahn, S. H. Han, and W. H. Kwon, "An Equation Based Approach to the Implementation of Generalized PDE Solver in CEMTool," *Submitted to 16th International Conference on Computer Applications in Industry and Engineering* , Las Vegas, Nevada USA, 2003.