

Research Journal of Applied Sciences, Engineering and Technology 4(10): 1277-1286 2012

ISSN: 2040-7467

© Maxwell Scientific Organization, 2012

Submitted: November 28, 2011

Accepted: January 04, 2012

Published: May 15, 2012

Teaching Finite Element Method of Structural Line Elements Assisted by Open Source FreeMat

¹Waluyo Adi Siswanto and ²Agung Setyo Darmawan

¹Department of Engineering Mechanics, Universiti Tun Hussein Onn Malaysia (UTHM),
86400 Parit Raja, Batu pahat, Johor, Malaysia

²Jurusan Teknik Mesin, Universitas Muhammadiyah Surakarta, Pabelan,
Kartasura, Surakarta, Indonesia

Abstract: One of the important objectives in teaching finite element method at introductory level is to bring students into the comprehension of finite element procedures. This study presents a strategy of teaching structural line elements involving an open source computer-aided learning tool FreeMat integrated with another open source CALFEM finite element toolbox. FreeMat, which is a programming based learning tool, is used together with other higher level learning tools; Open/Libre Office Spreadsheet and LISA finite element analysis application package. The spreadsheet is the main learning tool for students to implement finite element procedures whereas FreeMat is used for verification purpose in programming approach and LISA provides a practical skill in using finite element package program. Involving FreeMat in the learning process provides a quick verification check for the finite element solution. This verification tool helps students when they implement finite element procedures to solve structural problems.

Key words: Calfem, computer aided learning, finite element method, FreeMat, open source software

INTRODUCTION

Numerical analysis of structural problems based on Finite Element Methods (FEM) requires some basic knowledge of matrix operations. Since the matrices involved in the analysis are usually complicated, writing programming codes or routines sometimes are necessary to help the calculation procedures.

In the beginner level of finite element methods, however, introduction to programming approach will distract students' focus on the understanding of the theory of finite element methods. Therefore, using higher level of learning tools could be more appropriate and useful. Students will not be busy with writing codes, but employ a mathematical Computer-Aided Learning (CAL) tool to analyze the finite element problems. Students can concentrate in learning finite element theory and the numerical procedures before implementing them to solve example problems with the help of learning tools. The mathematical learning tools could be spreadsheets MS Excel (Microsoft, 2011) or Open/Libre Office Calc (LibreOffice, 2011), computer algebra systems and FEM application software.

There are many computer algebra systems available to help students in doing all matrix calculations required in FEM. Some computer algebra systems are commercial products such as Mathematica (Wolfram Research, 2011),

Maple (Maplesoft, 2011), MATLAB (Mathworks, 2011), and Mathcad (Parametric Technology Corp, 2011). For that reason, students are required to have proper licenses to use them.

Alternatively, free/Open Source Software (OSS) can also be used as FEM learning tool. The OSS includes Scilab (Scilab Consortium, 2011), Maxima (Maxima, 2011), SMath (Ivashov, 2011) and FreeMat (Basu, 2011). For FEM application software, among all the commercial software, LISA (Sonnenhof, 2011) is an alternative solution offering a limited number of nodes for free usage. It also provides a free learning tool for students either in or off campus.

FreeMat including the script m-file is compatible with the commercial mathematical tool MATLAB. Most of MATLAB commands are the same with FreeMat commands. If students know how to use FreeMat, they should be able to use the commercial MATLAB as well. The use of MATLAB of Mathwoks as a calculation tool has been extensively implemented for finite element analysis. Users have their own freedom in preparing high level language routine codes in its m-file editor and can interactively view the calculation process. There are many researchers who have documented the success of this tool for finite element analysis of structural problems. Amongst them are well documented. (Mueller, 2005; Kattan, 2007; Ferreira, 2009; Oluwole, 2011).

Corresponding Author: Waluyo Adi Siswanto, Department of Engineering Mechanics, Universiti Tun Hussein Onn Malaysia (UTHM), 86400 Parit Raja, Batu pahat, Johor, Malaysia, Tel.: +60-7-4537745; Fax: +60-7-4536080

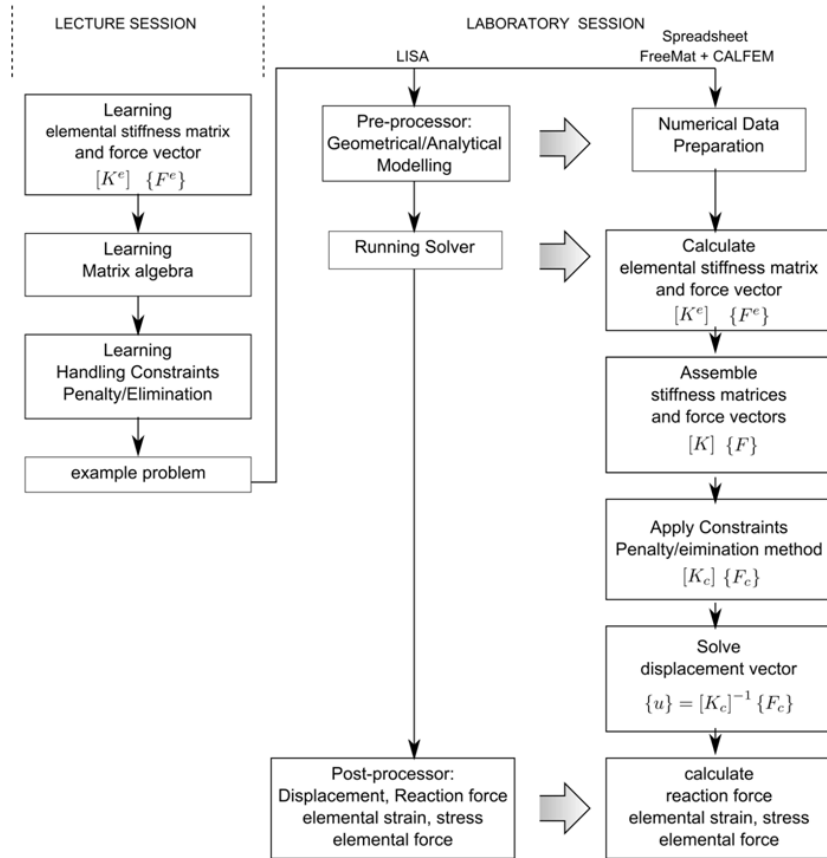


Fig. 1: Teaching approach diagram

Finite element functions have been defined in computer program CALFEM as a MATLAB toolbox for finite element applications (Calfem, 2011). FreeMat and CALFEM are licensed under GNU General Public License Version 2 which means users can use them for free, distribute and also modify them from the source. Although CALFEM has been widely used in MATLAB (Dahlblom *et al.*, 1986; Wideberg, 2004; Svedberg and Riunesson, 1998; Ottoson, 2010) there are no reports found when it is integrated in FreeMat.

Considering that students should have the accessibility to use the learning tools during the finite element method course, the calculation tools implemented for the course are Open/Libre Office Calc, computer algebra system FreeMat and LISA software. Libre/Open Office Calc, which is fully compatible with the famous commercial Microsoft Excel, is the main learning platform to implement finite element procedures based on matrix operations.

This study presents the integration of FreeMat and its usage in tertiary education to be implemented in the teaching and learning of finite element course. All functions defined in CALFEM are not modified since these are the representation of the original MATLAB

codes that should be called from within FreeMat. However, during the course, students are introduced to calling CALFEM build-in functions in different styles to encourage them to be more creative in writing codes.

The implementation and usage of open sources in formal education has been reported by several researchers (Lin and Zini, 2008; Van Rooij, 2007a; Van Rooij, 2007b). The adoption of Open Source Software (OSS) for teaching and learning is gaining traction, with dramatic increase in awareness and in campus-wide deployment of OSS over the past 3 years (Van Rooij, 2011).

Teaching approach: Structural line elements are taught in the first half of the finite element method course for Bachelor Degree Program at the Department of Engineering Mechanics, Universiti Tun Hussein Onn Malaysia (UTHM). The topics of line elements are taught before introducing students to heat transfer elements.

While learning structural line elements, students are not only learning the concept of finite elements theoretically but also trained to be familiar with the concept of finite element procedures. In the second half of the course, students are no longer guided to do the finite element calculations since they should have been gaining

sufficient knowledge to apply the procedures in higher application levels.

In static structural analysis, the simultaneous finite element equation is always in a form of:

$$[K_C]\{u\} = \{F_C\} \quad (1)$$

where $[K_C]$ is the element dependent assembled global stiffness matrix after considering all boundary condition, $\{u\}$ is the displacement vector and $\{F_C\}$ is the adjusted assembled force vector due to the present of boundary conditions. The displacement vector $\{u\}$ should be then solved to obtain the response of a structure under loading and constraint conditions.

To be able to find the displacement vector and other output results such as elemental forces, strains, stresses and reaction forces, students must learn numerical aspects from the theoretical side of elemental stiffness matrix, matrix algebra, handling boundary conditions or constraints to the solution of simultaneous equations. The line elements covered are one dimensional axial element, two-dimensional plane truss element and three-dimensional space truss element. The comprehension of this theoretical knowledge gained from lecture sessions must be implemented in the solution of real structural problems. Students are then guided in laboratory sessions to implement what they have learned.

In general, the teaching approach is as shown in Fig. 1. Students first study the elemental stiffness and basic matrix algebra from lecture sessions. At this stage students are expected to acquire the finite element theory and finite element calculation procedures. In the laboratory sessions, they have to do finite element analysis of given example problems employing FEM application software LISA and do their own calculations in Spreadsheet and FreeMat. While using LISA, students have to define the geometrical model and assign material properties in LISA pre-processor, run the solver, and see the results from the post-processor. Students obtain the skill on how to use a FEM application package but not directly implementing what they have learned from lecture sessions.

In the next stage, students should implement the theories they have learned in their spreadsheet. A basic template is provided to give them an idea on how to use the spreadsheet in a structured manner following the finite element calculation steps. For a quick calculation and verifications, some build-in FEM functions in CALFEM (Calfem, 2011) that is integrated to FreeMat can be used. This stage will show the comprehension of students in regard to the theory and the finite element procedures they have learned. Once students have completed their calculations and get the outputs, they can verify them with LISA results. Students have been observed building their own solver and able to assess their own comprehension.

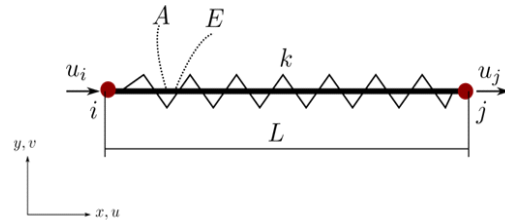


Fig. 2: One-dimensional Axial/Spring element

As a result of this approach, students will know how the finite element application package works and they can develop their calculation routines in spreadsheet, and also in FreeMat. Once students have done the simple axial element, they can do the same routines for 2D and 3D line elements using different stiffness matrices with higher level of complexity.

Stiffness matrix definition and FreeMat FEM functions:

To use FreeMat computer algebra system for FEM learning tool, integration with the CALFEM tool box for MATLAB is required. Although the open source CALFEM is designed to suit MATLAB environment, the FEM functions defined in CALFEM can be integrated in FreeMat. All predefined functions located in a local folder must be recognized by FreeMat and should be able to be called in the main program when it is necessary.

A simple path tool in FreeMat defines the location of the CALFEM functions. If the extraction folder is ~/calfem-3.4/, the path tool definition should refer to this directory; otherwise FreeMat cannot use all predefined CALFEM functions.

Stiffness matrix in use: The structural line elements defined in this course are those line elements with transitional degrees of freedoms. These types of elements are one-dimensional axial or spring element, two-dimensional plane truss element and three-dimensional space truss element.

The stiffness matrix of one dimensional axial element (Hutton, 2004) can be written in an equation with the indicated translational degrees of freedoms u_i and u_j as illustrated in Fig. 2:

$$[K^e] = k \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix} \begin{matrix} u_i \\ u_j \end{matrix} \quad (2)$$

In this Eq. (2), k indicates the spring constant. In the case that the element is not a spring but a rod (length L) with cross section A and the modulus elasticity E , the spring constant k can be replaced by the stiffness of the rod calculated by AE/L .

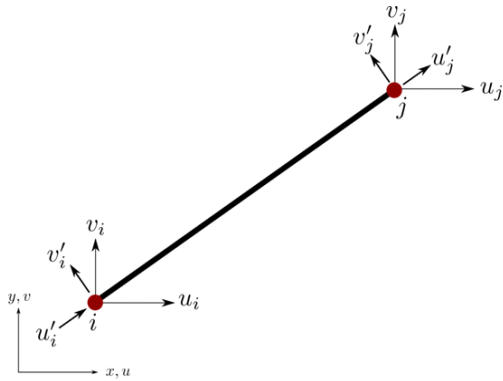


Fig. 3: Two-dimensional truss element

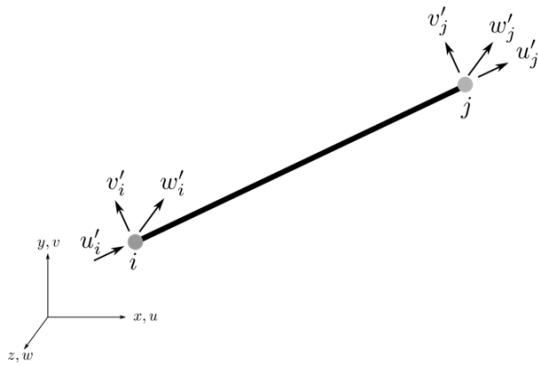


Fig. 4: Three-dimensional truss element

In two dimensional plane truss element (Fig. 3), the stiffness matrix is a 4x4 order matrix accommodating two degrees of global translational freedoms (u, v) in each node, as written in Eq. (3), whereas three-dimensional truss element (Fig. 4) is in 6x6 order matrix as shown in Eq. (4).

$$[K^e] = \frac{AE}{L} \begin{bmatrix} c^2 & cs & -c^2 & -cs \\ cs & s^2 & -cs & -s^2 \\ -c^2 & -cs & c^2 & cs \\ -cs & s^2 & cs & s^2 \end{bmatrix} \begin{matrix} u_i \\ v_i \\ u_j \\ v_j \end{matrix} \quad (3)$$

$$[K^e] = \begin{bmatrix} c^2 & cs & cz & -c^2 & -cs & -cz \\ cs & s^2 & sz & -cs & -s^2 & -sz \\ cz & sz & z^2 & -cz & -sz & -z^2 \\ -c^2 & -cs & -cz & c^2 & cs & cz \\ -cs & -s^2 & -sz & cs & s^2 & sz \\ -cz & -sz & -z^2 & cz & sz & z^2 \end{bmatrix} \begin{matrix} u_i \\ v_i \\ w_i \\ u_j \\ v_j \\ w_j \end{matrix} \quad (4)$$

The cosine direction constants of c, s and z are expressed as:

$$c = \frac{x_j - x_i}{L}, \quad s = \frac{y_j - y_i}{L}, \quad z = \frac{z_j - z_i}{L} \quad (5)$$

and the length of the element can be calculated from the coordinates of the node:

$$L = \sqrt{(x_j - x_i)^2 + (y_j - y_i)^2 + (z_j - z_i)^2} \quad (6)$$

Elemental stiffness functions in FreeMat: The original FEM functions defined in CALFEM are in simple forms. The functions are written as bar1e, bar2e and bar3e.

For an axial element:

$$k = \text{bar1e}(k) \text{ or } k = \text{bar1e}(AE/L)$$

For two and three dimensional truss elements:

$$k = \text{bar2e}(\text{ex}(I, :), \text{ey}(I, :), \text{ep}(I, :))$$

$$k = \text{bar3e}(\text{ex}(I, :), \text{ey}(I, :), \text{ez}(I, :), \text{ep}(I, :))$$

where ex, ey, and ez (only applicable for bar3e) define the coordinates of the nodes:

$$\text{ex} = [x1, x2]$$

$$\text{ey} = [y1, y2]$$

$$\text{ez} = [z1, z2]$$

and ep defines the property of the element represented by the cross section A and the modulus of elasticity E:

$$\text{ep} = [E, A]$$

The original CALFEM commands are modified in FreeMat to provide a convenient way for students to define the number of element. The commands are then written as:

$$k = \text{bar2e}(\text{ex}(I, 2:\text{end}), \text{ey}(I, 2:\text{end}), \text{ep}(I, 2:\text{end}))$$

$$k = \text{bar3e}(\text{ex}(I, 2:\text{end}), \text{ey}(I, 2:\text{end}), \text{ez}(I, 2:\text{end}), \text{ep}(I, 2:\text{end}))$$

where ex, ey, and ez are written with an additional information of element number I:

$$\text{ex} = [I, x1, x2]$$

$$\text{ey} = [I, y1, y2]$$

$$\text{ep} = [I, E, A]$$

The topology of bar2e element i is expressed by four degree of freedoms of the two nodes. The first node consists of two topology freedoms $u1$ and $u2$. The second node is represented by $u3$ and $u4$.

$$\text{edof} = [I, u1, u2, u3, u4]$$

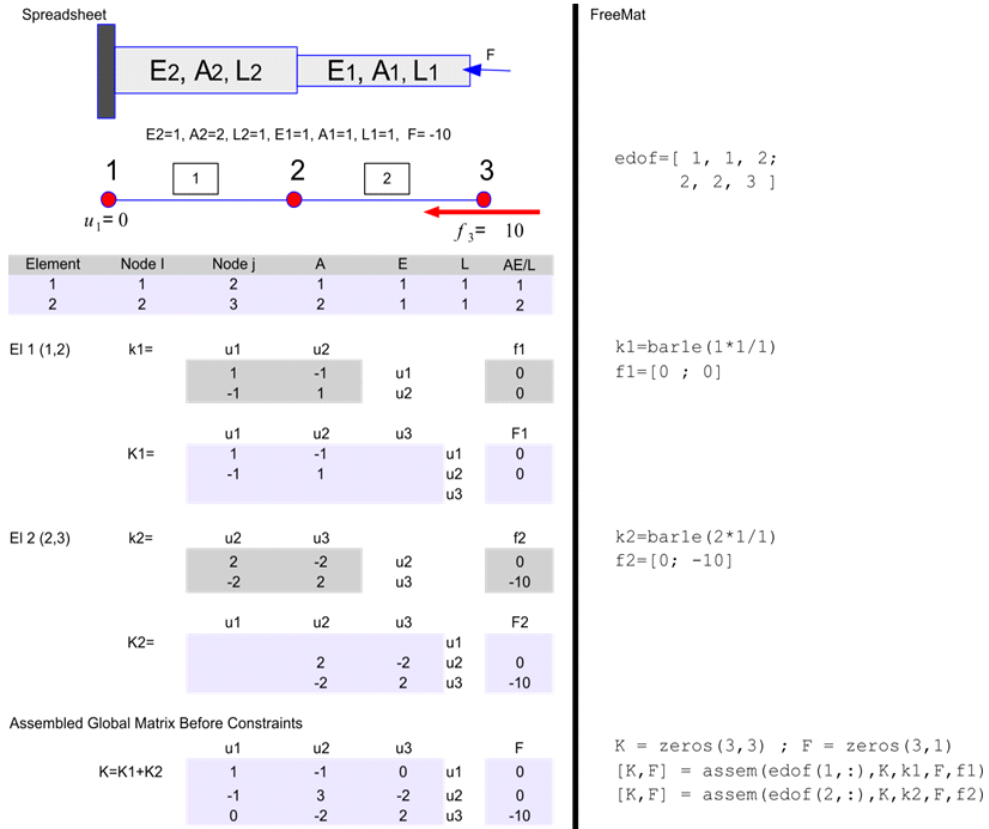


Fig. 5: Assembling process in Spreadsheet and FreeMat

The topology of bar3e element i is expressed by six degrees of freedoms of the two nodes.

$$edof = [I, u_1, u_2, u_3, u_4, u_5, u_6]$$

Elemental force vector: The elemental force vector is simply written in a vector form following the degrees of freedoms of the element. In three-dimensional truss, the elemental force vector should be written as:

$$f = [f_{ix}; f_{iy}; f_{iz}; f_{jx}; f_{jy}; f_{jz}]$$

For two-dimensional truss, there are only x and y components while one-dimensional only use x component.

Assembling stiffness matrix, force vector and handling constraints: All elemental stiffness matrices should be assembled into one global stiffness matrix. The assembling process should be done in the same way for the elemental force vectors. The constraints are then considered to be implemented to the assembled global stiffness matrix and the assembled global force vector.

The assembling process starts with the preparation of the global stiffness matrix and the global force vector. If the number of nodes in the structure is n with the number

of degrees of freedoms per node is dof , the order of matrix will be $n \times dof$.

Initially, the global stiffness matrix and the global force vector must be zero. To occupy the global stiffness matrix, the calculated elemental stiffness matrix must be relocated into the global matrix according to the degrees of freedoms in global system.

The implementation of assembling process for an axial structure in FreeMat and Spreadsheet as a comparison is illustrated in Fig. 5. Other structures involving two and three-dimensional elements follow the same procedures. In FreeMat window environment, the assembling process can be seen element by element. Students will really see the matrix assembling process and have a better comprehension from that.

An automatic calculation in the background instead of interactive elemental calculation can be introduced to students. Material properties can be defined first before looping the assembling process:

```

%--properties [element number, A, E, L ]
datake =[1, 1, 1, 1; 2, 2, 1, 1];
    
```

```

%--force data [element number, force node i, force node j]
datafe = [ 1, 0, 0; 2, 0, -10];
    
```

	A	B	C	D	E	F	G	H	I	J	K
34	Assembled Global Matrix Before Constraints										
35			u1	u2	u3	F					
36		K=	1	-1	0	u1	0				
37			-1	3	-2	u2	0				
38			0	-2	2	u3	-10				
39											
40											
41	Elimination										
42			(a)	(b)							
43		Kc=	1	0	0				Fc	constraint	
44			0	=D37	=E37				=K43	0	
45			0	=D38	=E38				=G37-C37*K43		
46									=G38-C38*K43		

Fig. 6: Elimination method in Spreadsheet

```
%--Assemble ke and fe into global K and F
K = zeros(3, 3);
F = zeros(3, 1);
```

```
for i = 1:3
    printf('Element %d\n', i);
    ke_data = datake(i, 2)*datake(i, 3)/datake(i, 4);
    ke = bar1e(ke_data);
    fe_data = datafe(i, 2:end);
    fe = fe_data';
    [K, F] = assem(edof(i, :), K, ke, F, fe)
end
```

The boundary conditions or constraints of the structural nodes are considered after the assembling process has been completed. In structural problems, the constraints are defined by the displacement conditions. If a node is fixed, then it is simply defined as zero displacement at the respective degree of freedom. For a predefined displacement β the solution should satisfy the defined value β .

There are two approaches in handling constraints: penalty method and elimination method. In elimination method, all predefined displacements must be eliminated since the solutions are already known. This will reduce the order number of matrix. Other remaining equations are then adjusted due to the elimination.

The elimination process can be explained in spreadsheet as illustrated in Fig. 6. This illustrates the elimination process of the problem depicted in Fig. 5. The constraint occurs at node 1 with a constraint value $\beta = 0$. The elimination process:

- Replaces the stiffness component where a constraint is applied by 1
- Draws cross lines through the constraint location
- Adjusts the force vector

If there is a constraint in any row, the value of force vector at that particular row must be directly replaced by the constraint value. Other rows should be adjusted to balance the altered values in the stiffness matrix. As a

result, the first row is solved and can be eliminated. The unsolved equations are then reduced. In this example (Fig. 6), only two equations remain unsolved.

In penalty method, there will be no elimination. The procedures are as follows:

- A constraint penalty constant C must be introduced and need to be higher than any other constants in the stiffness matrix, and
- Any cells with constraints must be penalized by adding the penalty constant.

In this example (Fig. 7) the constraint is $u_1 = 0$, and at location (u_1, u_2) the penalty applies. As for the last procedure (c), the force vector at the constrained row must be penalized by adding βC .

It can be seen that when the penalty constant is very high, the constraint can be satisfied. The first row where constraint occurs, the equation is written as:

$$(k_{11} + C)u_1 + k_{12}u_2 + k_{13}u_3 = f_1 + C\beta \quad (7)$$

when the penalty constant C is dominant because of its high value, other terms will not be significant, and become:

$$Cu_1 = C\beta \quad (8)$$

which satisfy the constraint $u_1 = 0$ for $\beta = 0$.

The boundary condition function has been defined in CALFEM and can be called in FreeMat. The functions must be defined in each affected node. As an example used in Fig. 5, at node 1 with constraint $\beta = 0$, the boundary condition variable bc can be coded as:

$$bc = [1, 0]$$

The general code of bc should be in a form of:

$$bc = [1, \beta_1; 2, \beta_2; n, \beta_n]$$

34	A	B	C	D	E	F	G	H	I	J
35	Assembled Global Matrix Before Constraints									
36			u1	u2	u3		F			
37		K=	1	-1	0	u1	0			
38			-1	3	-2	u2	0			
39			0	-2	2	u3	-10			
40										
41	Penalty	1.0E+30	(a)	(b)		(c)	Fc	constraint		
42										
43		Kc=	=C36+B41	=D36	=E36	=G36+I43*B41				0
44			=C37	=D37	=E37	=G37				
45			=C38	=D38	=E38	=G38				
46										

Fig. 7: Penalty method in Spreadsheet

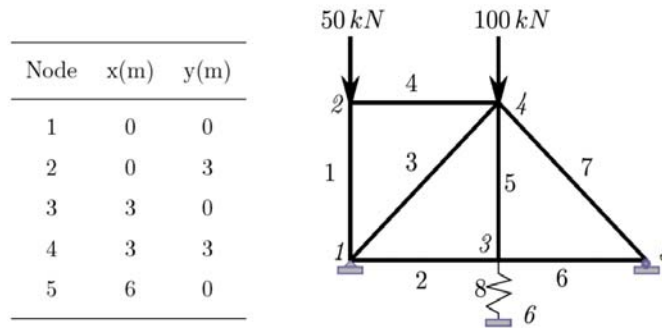


Fig. 8: Two-dimensional problem

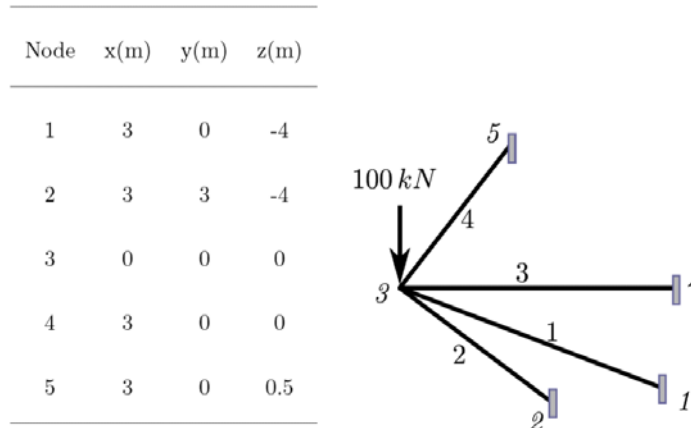


Fig. 9: Three-dimensional model

After the boundary conditions have been incorporated, the solution can be initiated. In spreadsheet, the displacements and reaction forces are obtained by running the following procedures involving build-in functions MINVERSE and MMULT:

$$\text{Inverse}[Kc] = \text{MINVERSE}(kc_{11}:kc_m)$$

$$\text{displacement} = \text{MMULT}(\text{Inverse}[Kc]; \text{Vector}[Fc])$$

$$\text{reactionforce} = \text{MMULT}([K]; [u]) - [F]$$

Free Mat calculates displacement and reaction force at once using a single command:

$$[u, rf] = \text{solveq}[K, F, bc]$$

which return in parallel the displacement and reaction force vectors that help students to verify their calculations in Spreadsheet.

Implementation in two and three-dimensional: In two and three dimensional problems, students follow the same procedures in Fig. 7 by doing the exercises as illustrated in Fig. 8 and 9. For the two-dimensional problem, all elements are made of the same material with $E = 70 \text{ GPa}$ and cross section $A = 3/10^4 \text{ m}^2$. The spring stiffness is 10 kN/m .

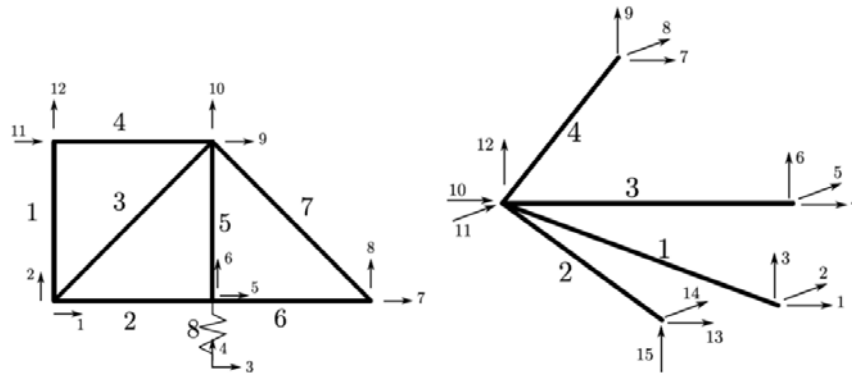


Fig. 10: Topology displacement numbering in FreeMat

The two-dimensional CALFEM function to use is `bar2e`. Typically the FreeMat codes to calculate the elemental stiffness matrix and elemental force vector and to assemble them into global stiffness matrix and global force vector are the following:

```
% ---- 2D assemble stiffness matrix and force vector
for i = 1:8
    ke = bar2e(ex(i, 2:end),ey(i, 2:end),ep(i, 2:end));
    fe_data = datafe(i, 2:end);
    fe = fe_data';
    [K, F] = assem(edof(i, :), K, ke, F, fe);
end;
```

end;

For three-dimensional problem, all elements have $E = 100 \text{ GPa}$ and $A = 5/10^4 \text{ m}^2$. The applied concentrated loading is -100 kN (z direction).

The FreeMat codes to construct global stiffness matrix and force vector of three-dimensional problem are written as:

```
% ---- 3D assemble stiffness matrix and force vector
for i = 1:4
    ke = bar3e (ex(i, 2:end), ey(i, 2:end),ez(i, 2:end),
    ep(i, 2:end));
    fe_data = datafe(i, 2:end);
    fe = fe_data';
    [K, F] = assem(edof(i, :),K, ke, F, fe);
end;
```

The degrees of freedoms are defined as numbers starting from 1. The numbering displacements used for the two models are illustrated in Fig. 10.

Students are trained to develop their own solver implemented in spreadsheet. They can always do crosschecking with FreeMat that can provide fast calculation results. They can also see whether their calculations are correct or not by comparing them with LISA analysis results.

Table 1: Two-dimensional displacement comparison results

Node	Displ np (Fig. 10)	LISAS	Spreadsheet (Penalty method)	Free Mat
1	u_1	0	$-1.231e-41 (\neq 0)$	0
1	u_2	0	$-9.986e-26 (\neq 0)$	0
6	u_3	0	0	0
6	u_4	0	$-2.273e-28 (\neq 0)$	0
3	u_5	$7.12341e-03$	$7.12341e-03$	$7.12341e-03$
3	u_6	$-2.72325e-02$	$-2.72325e-02$	$-2.72325e-02$
5	u_7	$1.42468e-02$	$1.42468e-02$	$1.42468e-02$
5	u_8	0	0	0
4	u_9	$7.12341e-03$	$7.12341e-03$	$7.12341e-03$
4	u_{10}	$-2.72714e-02$	$-2.72714e-02$	$-2.72714e-02$
2	u_{11}	$7.12341e-03$	$7.12341e-03$	$7.12341e-03$
2	u_{12}	$-7.14286e-03$	$-7.14286e-03$	$-7.14286e-03$

Table 2: Three-dimensional displacement comparison results

Node	Displ no (Fig. 10)	LISA	Spreadsheet (penalty method)	Free Mat
3	u_{10}	$-2.65257e-03$	$-2.65257e-03$	$-2.65257e-03$
3	u_{11}	$-2.22616e-04$	$-2.22617e-04$	$-2.22616e-04$
3	u_{12}	$-1.11998e-02$	$-1.11998e-02$	$-1.11998e-02$
1	u_1, u_2, u_3	0	$2.7e-36 (\approx 0)$	0
4	u_4, u_5, u_6	0	$4.3e-42 (\approx 0)$	0
5	u_7, u_8, u_9	0	$6.2e-39 (\approx 0)$	0
2	u_{13}, u_{14}, u_{15}	0	$5.7e-46 (\approx 0)$	0

In completing their exercises, students should have comparable results as listed in Table 1 and 2. Students are also suggested to do numerical experiments by using different penalty constants. They will learn the effect of penalty constant to the accuracy that the higher penalty constant the more accurate results they can get.

Students opinion on open source Free Mat: In order to assess the prospect of FreeMat to be further involved in other topics of FEM, students are asked their opinion regarding FreeMat as a new learning tool for FEM course after they have completed the course. The number of students participated is 87. These students have never used FreeMat before taking the FEM course.

The number of students saying that FreeMat does not help in understanding FEM concept is 7% (Fig. 11) while other tools; Spreadsheet (Open Office) and LISA, none of the students consider them as not useful.

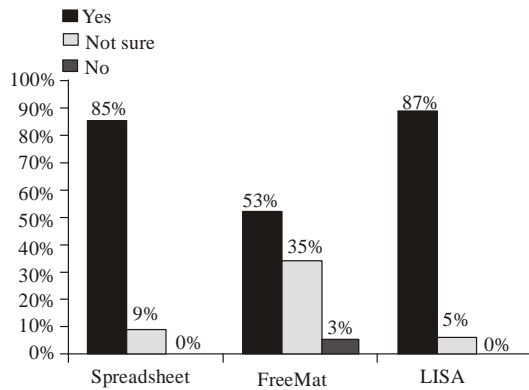


Fig. 11: Usefulness of Spreadsheet, FreeMat and LISA

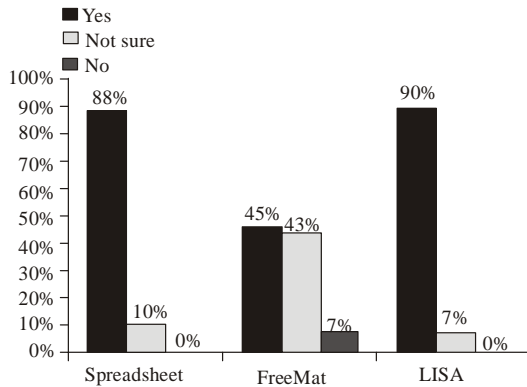


Fig. 12: Students confidence in using learning tools

Students rely on their spreadsheet in implementing finite element procedures. The Fig 11 shows that 88% of the students agree that the spreadsheet is really helpful, almost similar percentage in LISA, which is 90%. The percentage of students who consider FreeMat which is used as an optional tool to verify their calculation in spreadsheet as useful learning tool is as high as 45%. On the other hand another 43% of the students think that it is slightly useful (Fig. 11).

When using FreeMat, students are expected to learn the CALFEM available functions to do the analysis. Even if they do not know the procedure they can do the analysis by using the functions. Therefore FreeMat is considered as slightly useful since it provides shortcut calculations using its build-in functions. For verification purpose, FreeMat is a good tool to help students to check their calculation results in spreadsheet.

Although FreeMat is a new learning tool, 53% of the students are confident in using FreeMat after the course is completed (Fig. 12). This is a good indication that students know how to use it after being guided throughout the course. Therefore, it can be inferred that more than half of the respondents are confident in using FreeMat to solve structural line element problems.

CONCLUSION

The involvement of an integrated open source FreeMat and CALFEM helps students in understanding the concept of finite element method of structural line problems. FreeMat provides a fast verification results for students working with spreadsheet calculation tool.

Throughout the course involving LISA, Spreadsheet and FreeMat, students are not only able to use finite element application software LISA, but they also get know how to develop the finite element solver for structural line problems. Students are trained to implement the finite procedures in spreadsheet and at the same time they need to get familiar with programming approach in FreeMat. The comprehension of the students in this line element topic is very high indicated by their ability to implement the knowledge in solving problems.

There is a small number of students (7%) with negative opinion that FreeMat is not a useful FEM learning tool. This Fig. 11 shows that FreeMat is prospective to help students to study FEM better. There are some possibilities to involve FreeMat in higher level of FEM topics.

REFERENCES

- Basu, S., 2011. Freemat 4.0 Online Documentation. Retrieved from: <http://freemat.sourceforge.net/help/index.html>, (Accessed on: November 27, 2011).
- CalfeM, 2011. CALFEM: A Finite Element Toolbox: Version 3.4, Lund University, Sweden, Retrieved from: <http://sourceforge.net/projects/calfeM>, (Accessed on: July 8, 2011).
- Dahlblom, O., A. Peterson and H. Petersson, 1986. CalfeM a program for computer-aided learning of the finite element method. Eng. Comp., 3(2): 155-160.
- Ferreira, A.J.M., 2009. MATLAB Codes for Finite Element Analysis. Springer.
- Hutton, D.V., 2004. Fundamentals of Finite Element Analysis. McGraw Hill, New York, pp: 21-25.
- Ivashov, A., 2011. Smith Version 0.89. Retrieved from: <http://en.smath.info>, (Accessed on: November 27, 2011).
- Kattan, I., 2007. MATLAB Guide to Finite Elements. Springer.
- Libre Office, 2011. Libre Office Spreadsheet Calc. Retrieved from: <http://www.libreoffice.org/features/calc/>, (Accessed on: November 27, 2011).
- Lin, Y.W. and E. Zini, 2008. Free/libre open source software implementation in schools: Evidence from the field and implications for the future. Comp. Educ., 50(3): 1092-1102.
- Maplesoft, 2011. Maple. Retrieved from: <http://www.maplesoft.com/products/Maple>, (Accessed on: November 27, 2011).
- Mathworks, 2011. MATLAB. Retrieved from: <http://www.mathworks.com/products/matlab>, (Accessed on: November 27, 2011).

- Maxima, 2011. Maxima. Retrieved from: <http://maxima.sourceforge.net>, (Accessed on: November 27, 2011).
- Microsoft, 2011. Microsoft Excel 2010. Retrieved from: <http://office.microsoft.com/en-us/excel>, (Accessed on: November 27, 2011).
- Mueller Jr., D.W., 2005. An introduction to the finite element method using matlab. *Int. J. Mech. Eng. Educ.*, 33(3): 260-277.
- Oluwole, O., 2011. *Finite Element Modeling for Material Engineers Using MATLAB*, Springer.
- Ottoson, A., 2010. Implementation of CALFEM in python, Master's thesis, Lund University, Department of Construction Sciences.
- Parametric Technology Corp, 2011. Mathcad. Retrieved from: <http://www.ptc.com/products/mathcad>, (Accessed on: November 27, 2011).
- Scilab Consortium, 2011. Scilab. Retrieved from: <http://www.scilab.org/products/scilab>, (Accessed on: November 27, 2011).
- Sonnenhof, 2011. LISA Version 7.6.0 Sonnenhof Holdings. Retrieved from: <http://www.lisa-fet.com> (Accessed on: November 27, 2011).
- Svedberg, T. and K. Riunesson, 1998. An algorithm for gradient-regularized plasticity coupled to damage based on a dual mixed fe-formulation. *Comp. Meth. Appl. Mech. Eng.*, 161(1-2): 49-65.
- Van Rooij, S.W., 2007a. Open source software in higher education: Reality or illusion? *Educ. Inf. Tech.*, 12(4): 191-209.
- Van Rooij, S.W., 2007b. Perceptions of open source versus commercial software: Is higher education still on the fence? *J. Res. Tech. Educ.*, 39(4): 433-453.
- Van Rooij, S.W., 2011. Higher education sub-cultures and open source adoption. *Comp. Educ.*, 57(1): 1171-1183.
- Wideberg, J.P., 2004. Simplified method for evaluation of the lateral dynamic behaviour of a heavy vehicle. *Int. J. Heavy Veh. Sys.*, 11: 195-207.
- Wolfram Research, 2011. Mathematica. Retrieved from: <http://www.wolfram.com/mathematica>, (Accessed on: November 27, 2011).