Learning the Stiffness Method with FOTRAN, Excel and MathCAD

Peter W. Hoadley[1]

Abstract - The stiffness method for structural analysis has been taught at the undergraduate level for several years. Computer languages like BASIC, FORTRAN, Pascal, C++ and Visual Basic have all been used in the classroom to teach the fundamentals of the method. For the last several years in the civil engineering department at VMI, Excel supplemented by commercially available software has been used to present the method. Just recently a student wanted to use MathCAD rather than Excel. A comparison was made as to how well the stiffness method was *learned* using MathCAD as compared to using Excel in this given class. One could argue that the stiffness method can best be programmed using one software tool as compared to the other but that is not the question posed here. Each software tool has its strengths and weaknesses. It is the effectiveness of the tool for *learning* that is addressed by this paper.

Keywords: Stiffness method, student learning, software tools

Introduction

In 1930 Hardy Cross published a paper describing a technique that revolutionized structural analysis of indeterminate frames. [Cross, 3] The method is an iterative technique perfectly suited to hand calculations and is still taught today, even in this computer age, as a tool to explain how internal moments caused by external loads are distributed through a structure. Analysis techniques in the 1930's, including Cross's moment distribution method, relied on member-end *forces* as unknowns.

The roots of the stiffness method, which relies on member-end *deformations* as unknowns, date to the early 1900's. [Maney, 5] The method was not popular for complicated structures since it necessitated the solution of large sets of equations. Once the digital computer was developed the stiffness method became practical and it is the solution technique of choice today.

The stiffness method was taught to graduate students in the 1950's and as digital computers became available to undergraduates, the method was introduced to them. In the early days the method had to be programmed using machine code but higher level languages were soon developed. In the 1990's the development of spreadsheets and other software have made programming of the method far easier.

The civil engineering department at VMI offers an elective course presenting the stiffness method. Ten years ago FORTRAN was used by the students to program the method. In the last few years the Excel© spreadsheet has been used and most recently MathCAD© has been investigated to determine if it would enhance *learning*. The primary concern is that the chosen software promote and not hinder a student's comprehension of the stiffness method. Every software tool has its strengths and weaknesses but the purpose of this paper is to determine which software tool enhances *learning*. Often the programming exercise proves "too time consuming and students will spend most of their time debugging their computer programs rather than studying the subject. They may lose sight of the forest for the trees." [Chen, 2] Students may focus "on the notational details of a certain language" preventing them from "grasping the big picture." [ECOOP, 4]

Every software tool has its strengths and weaknesses and one may serve better than another in a given situation. It is not the purpose of this paper is to lift one software tool over another in a general sense. The purpose of this paper is to describe how *learning* was affected in a senior level undergraduate class when using FORTRAN, Excel and MathCAD to program the stiffness method.

[1] Department of Civil and Environmental Engineering, VMI, Lexington, VA 24450, email:hoadleypw@vmi.edu

The Stiffness Method

The stiffness method for structural analysis involves solving a set of equilibrium equations. Members of a structure are isolated and end forces are written in terms of loads and deformations. Material properties, geometry and member loads serve as input to the process. Support conditions and member connectivity are also needed as input. These member end force equations are written in matrix form and then modified for member orientation. Joint equilibrium equations are written in terms of these end forces resulting in a linear set of equations with deformations as the unknowns. This set of equations is modified for support conditions and then solved yielding values for the unknown deformations. These known deformations are substituted into the original equations for member end forces. Deformations and member end forces serve as output.

FOTRAN, MathCAD and Excel are excellent software tools that can be used to program the stiffness method. The question addressed here is, "Which software package is best for *teaching*, *learning* and *understanding* the stiffness method?"

Learning the Stiffness Method with FORTRAN

FORTRAN is an acronym that stands for "FORmula TRANslation" and is a programming language developed in the late 1950's. [Backus, 1] It was, in some cases perhaps still is, a very popular language in engineering circles. Other software claim superiority but FORTRAN seems to endure. At VMI during the 1980's and early 1990's, FORTRAN was taught in a dedicated course during the sophomore year and used as the primary software tool for teaching the stiffness method. The stiffness method includes the use of matrices, loops and conditional branching all of which are easy to program in FORTRAN. It is the author's experience that the language is not well suited for *learning* the stiffness method.

There are three basic problems with FORTRAN when learning the stiffness method – input/output, syntax and conceptual understanding. Regarding input, a data file has to be created including information about geometry, materials, loads and supports for the structure. The student then has to write the FORTRAN code to properly read the data. This leads to the second problem regarding syntax. Too often the student does not write the proper code to correctly read the data file. It is surprising how much time a student will spend trying to debug their program only to discover that "FORMAT" was spelled incorrectly! The format of output is relatively tedious and is merely a list of numbers that can be difficult to interpret.

Syntax in general is a problem students find difficult when programming FORTRAN. Experience shows that what at first glance appeared to be a major problem in a student's program is merely a missing "comma" or a command incorrectly spelled. It is remarkable how many creative ways a student can spell "CONTINUE!"

Matrix creation and manipulation is central to the stiffness method. It is the author's experience that college students do not find matrix manipulation difficult but do find it difficult to program. For example, the set of equilibrium equations necessary to solve for unknown displacements can be created by hand and visually offers a powerful representation of how the stiffness method works. Students must take this process and define it in FORTRAN code which requires nested loops and conditional branching. This is a great programming exercise but the energy required to program the process interferes with *learning*.

One advantage of FORTRAN is that the same program that is written to solve for deformations and member-end forces in a simple frame can be used to solve the same in a complicated frame. A generalized program is useful but does not help in *learning* the method.

Learning the Stiffness Method with MathCAD

MathCAD© is a powerful equation solving software tool ideally suited for engineering problems. The software can be used to solve implicit equations, sets of equations and symbolic equations. MathCAD includes programming tools which can be used to solve structural analysis problems using the stiffness method.

One advantage MathCAD has over FORTRAN is that it is far more visual. Matrices are displayed in a way that students can easily understand what each term represents as shown in Figure 1. This figure illustrates a MathCAD function called "LMSM" with length, "Len", area, "Ar", moment of inertia "I" and modulus, "E" as arguments. The function shows clearly how all variables influence each term in the matrix. The visual nature of the matrix makes it easy to demonstrate to students how it can be partitioned and what each partition represents. Matrix multiplication need not be programmed as in FORTRAN, therefore, the student does not lose sight of the purpose of the matrix multiplication that tedious programming may hide.

Assembling the set of joint equilibrium equations using MathCAD is awkward. Each term in the matrix could be created individually which would be tedious. The matrix is symmetric which may simplify the process but this is not a practical option. The process can be programmed in MathCAD much as it is in FORTRAN.

$$\mathrm{LMSM}(\mathrm{Len},\mathrm{Ar},\mathrm{I},\mathrm{E}) := \begin{pmatrix} \frac{\mathrm{Ar}\cdot\mathrm{E}}{\mathrm{Len}} & 0 & 0 & -\frac{\mathrm{Ar}\cdot\mathrm{E}}{\mathrm{Len}} & 0 & 0 \\ 0 & \frac{12\cdot\mathrm{E}\cdot\mathrm{I}}{\mathrm{Len}^3} & \frac{6\cdot\mathrm{E}\cdot\mathrm{I}}{\mathrm{Len}^2} & 0 & -\frac{12\cdot\mathrm{E}\cdot\mathrm{I}}{\mathrm{Len}^3} & \frac{6\cdot\mathrm{E}\cdot\mathrm{I}}{\mathrm{Len}^2} \\ 0 & \frac{6\cdot\mathrm{E}\cdot\mathrm{I}}{\mathrm{Len}^2} & \frac{4\cdot\mathrm{E}\cdot\mathrm{I}}{\mathrm{Len}} & 0 & -\frac{6\cdot\mathrm{E}\cdot\mathrm{I}}{\mathrm{Len}^2} & \frac{2\cdot\mathrm{E}\cdot\mathrm{I}}{\mathrm{Len}} \\ -\frac{\mathrm{Ar}\cdot\mathrm{E}}{\mathrm{Len}} & 0 & 0 & \frac{\mathrm{Ar}\cdot\mathrm{E}}{\mathrm{Len}} & 0 & 0 \\ 0 & -\frac{12\cdot\mathrm{E}\cdot\mathrm{I}}{\mathrm{Len}^3} & -\frac{6\cdot\mathrm{E}\cdot\mathrm{I}}{\mathrm{Len}^2} & 0 & \frac{12\cdot\mathrm{E}\cdot\mathrm{I}}{\mathrm{Len}^3} & -\frac{6\cdot\mathrm{E}\cdot\mathrm{I}}{\mathrm{Len}^2} \\ 0 & \frac{6\cdot\mathrm{E}\cdot\mathrm{I}}{\mathrm{Len}^2} & \frac{2\cdot\mathrm{E}\cdot\mathrm{I}}{\mathrm{Len}} & 0 & -\frac{6\cdot\mathrm{E}\cdot\mathrm{I}}{\mathrm{Len}^2} & \frac{4\cdot\mathrm{E}\cdot\mathrm{I}}{\mathrm{Len}} \end{pmatrix}$$

Figure 1 A MathCAD function that creates a member stiffness matrix.

The author and his students find programming in MathCAD rather awkward. The author was told by colleagues who use MathCAD extensively that they will use another language for programming before they use MathCAD. The programming process in MathCAD requires a function and several nested loops. For example, what a student would type to create the first few lines of a routine for assembling the GSSM might look something like the following (what is actually typed or selected from a menu is in bold):

Build(A,B,RB,RE,CB,CE):]k{0
Select the next empty box
Cntrl=", Select the first box, **i** Select the second box, **RB;RE**
Select the next empty box
]m{ 0
Etc.

The resulting MathCAD code is shown in Figure 2.

```
Build(A,B,RB,RE,CB,CE) := | k ← 0
                          | for i ∈ RB..RE
                          |     m ← 0
```

Figure 2. Example of MathCAD code

Some of the elements of the MathCAD code can be selected from menus making the process easier.

In the statement shown in Figure 2, "Build" creates a function. The arguments of the function include matrix "A" representing the member end forces for a given member, matrix "B" representing the set of joint equilibrium equations for the whole structure and the other arguments represent joint numbers used as indices in programming loops. The vertical line or "Add line" in MathCAD is used to create a section that is not unlike a subroutine in FORTRAN. The symbol "←" or "local definition arrow" in MathCAD is the same as an equal sign in a FORTRAN subroutine. The "for" statement sets up a loop. The "∈" symbol means "in the range of" and is used

for defining the beginning and end of the loop. In terms of syntax the student must learn about functions, "add line", local definitions, how to use "for-next" loops, and more. Learning how to program in MathCAD becomes an impediment to *learning* the stiffness method.

Other syntax hurdles include the "equal" sign as there are four different kinds (or six depending on who's counting)! Note that in Figure 2 three "equal" signs are used: ":=", "←" and the "∈." The first is an assignment and is inserted by typing a "colon." The second is a local assignment statement and is inserted by typing a "left bracket." The third is automatically created by the "if" statement. Other equal signs include one which yields a value of an expression, one used in symbolic equations (this is the algebraic equal sign) and a one used in global definitions.

There are two types of subscripts – text and matrix index. They look alike but the text subscript is created by using a "period" and the matrix index subscript is created by using a "[". To enter $r_A=k_0*C_A$ where r_A is a subscripted variable while k_0 and C_A are elements of an array one must type "**r.A:k[0*C[A**". The resulting MathCAD code is shown in Figure 3. This is a feature that is ripe for errors difficult to debug.

$$r_A := k_0 \cdot C_A$$

Figure 3 Example of MathCAD code

Correcting subscript errors can be difficult. For example, to delete a matrix index the " [" must be deleted but it is not displayed on the screen. One has to know that it is hidden.

Tables for input and output can be created and this is an improvement over FORTRAN. Only one format may be used for variables and constants which limits how input/output is displayed. Order of computations is purely linear so output cannot be displayed next to input.

Though the stiffness method can be programmed in MathCAD it is not well suited for *learning* the method by undergraduate students.

Learning the Stiffness Method with Excel

Using Excel to learn the stiffness method has definite advantages over MathCAD and FORTRAN. Consider the three-member frame in Figure 4. The frame consists of three members supported by two pins subjected to a horizontal load, vertical load and a moment. All three members are rotated from the horizontal. A table well labeled and formatted is used for the input. All input is placed in the yellow-shaded cells and is easily modified for modifications in material, stiffness, geometry, support conditions or loads. Output including joint deformations and member end forces can be displayed on the same page in well labeled and formatted tables. All output is displayed in the blue-shaded cells. Having the output displayed on the same page as input makes it easy to perform "what-if" studies of the frame. All computations are executed in other locations of the spreadsheet.

A template for member matrices can be created, as shown in Figure 5, and "cut and pasted" for multiple members with different inputs. In MathCAD the terms of the matrix equation are in symbolic form making it clear what variables affect each value. Excel merely displays the numerical result for each term and does not show what values are included in the computations. As in MathCAD matrices are displayed visually making it easy for the student to understand what each term represents. Excel has advanced formatting options allowing one to display the matrices into logical partitions. This is very useful in understanding the nature of these member matrices. It also aids in understanding how the member matrices are used to create the primary matrix of joint equilibrium equations.

The joint equilibrium equations (JEE) are difficult to assemble in a general way in Excel. Technically each term in the matrix is unique; however, since member matrices may be partitioned, multiple terms in the JEE can be created with one simple equation. Symmetry further decreases the number of unique equations necessary to form the JEE matrix.

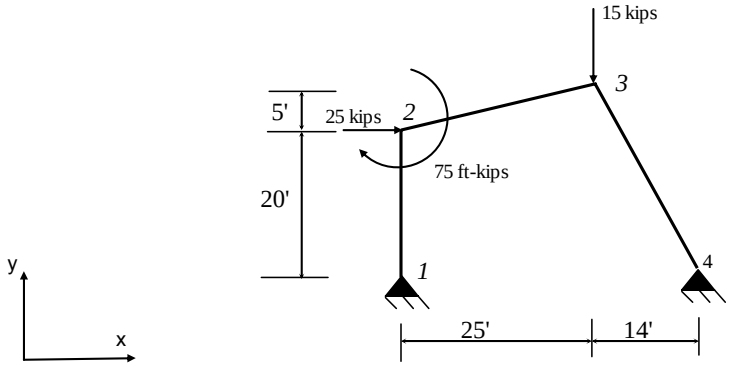
The JEE must be modified for support conditions which necessitates conditional branching. This is relatively easy to implement in Excel.

The most tedious step is the use of the nodal deformations in the computation of member end loads. All necessary matrices exist but some accurate accounting is necessary to match the right matrix with the right deformations.

Since the mechanics of the stiffness method are relatively easy to program in Excel more time can be spent focusing on the derivation of the stiffness coefficients in the member matrices. The transformation matrices and subsequent matrix multiplications are easily executed in Excel so the meaning of the process is not clouded by the complication of the execution. The partitioning of the stiffness matrices can be easily illustrated in Excel, which

Three-bar Frame with Nodal Loads

Nodal deformations and member end forces are computed for the 3-bar frame shown below.
Yellow regions are for input and units of inches and kips were used throughout.
A "U" was used for unknown nodal deformations. Output is shown in the blue regions.



Nodal Properties **Nodal Forces** **Known Deformations**

Node #	x	y	F_x	F_2	M	δ_x	δ_y	θ
1	0	0	0	0	0	0	0	U
2	0	240	25	0	-900	U	U	U
3	300	300	0	-15	0	U	U	U
4	468	0	0	0	0	0	0	U

Member Properties and Connectivity

Member #	Area	Moment of Inertia	Modulus	Length	Angle	i-node	j-node
1	110	5500	4500	240	90	1	2
2	240	12000	4500	306	11	2	3
3	144	9000	4500	344	-60.75	3	4

Deformations

Node #	δ_x	δ_2	θ
1	0	0	-0.007444
2	1.248560	0.004538	-0.000718
3	1.122555	0.609994	0.001021
4	0	0	-0.006083

Member Forces *(according to the global axis system)*

	i-node			j-node		
Member #	Fx	Fy	M	Fx	Fy	M
1	-5.78	-9.36	0	5.78	9.36	1387
2	19.22	-9.36	-2287	-19.22	9.36	-1674
3	19.22	-24.36	1674	-19.22	24.36	0

Figure 4. Excel Spreadsheet - Input and Output

helps students understand the proper position of member matrices in the JEE. Excel's syntax is easy to learn and does not confuse the computational process.

When learning the stiffness method, Excel is a relatively easy tool to use that visually demonstrates the process. Of course for the general case some programming language is best. Excel does include programming tools through Visual Basic but has the same disadvantages as in FORTRAN and MathCAD.

Local Member Stiffness Matrix for Member 1

Member	i-node	j-node	Area	MOI	Modulus	Length	Angle
1	1	2	25	1200	29000	216	90

Local Member Stiffness Matrix (LMSM)

		δ_{ix}	δ_{iy}	θ_i	δ_{jx}	δ_{jy}	θ_j	
P_i		3356	0	0	-3356	0	0	δ_{ix}
V_i		0	41	4475	0	-41	4475	δ_{iy}
M_i	=	0	4475	644444	0	-4475	322222	θ_i
P_j		-3356	0	0	3356	0	0	δ_{jx}
V_j		0	-41	-4475	0	41	-4475	δ_{jy}
M_j		0	4475	322222	0	-4475	644444	θ_j

Figure 5. Excel member stiffness matrix

Conclusion

Learning the stiffness method is the primary goal in a senior level civil engineering elective at VMI. FORTRAN, Excel and MathCAD have all been used by students to program the method. Both FORTRAN and MathCAD present obstacles to learning since writing and debugging can be difficult. Excel is easy to program and debug and has proven to be the best software tool for students learning the method. Since using Excel the author has found that more material is covered in the course with better understanding than when using MathCAD or FORTRAN.

REFERENCES

[1] Backus, J., "The History of FORTRAN I, II and III," *History of Programming Languages*, Edited by Richard L. Wexelblat, Academic Press, Blue Bell, Pennsylvania, 1981, pg. 25-74.

[2] Chen, HH, "Pedagogically Effective Programming Environment for Teaching Mechanism Design," Computer Applications in Engineering Education, Vo. 2, No. 1, 1994, pp. 23-39.

[3] Cross, Hardy, "Analysis of Continuous Frames by Distributing Fixed-End Moments" Proceedings of the American Society of Civil Engineers, ASCE, May 1930.

[4] European Conference on Object-Oriented Programming, as quoted from the "Call for Participation" for the Tenth Workshop on Pedagogies and Tools for the Teaching and Learning of Object Oriented Concepts, ECOOP, July 3, 2006, Nantes, France.

[5] Maney, G.B., "Studies in Engineering", No. 1, University of Minnesota, Minneapolis, 1915.

Peter W. Hoadley

Hoadley received his undergraduate degree from Vanderbilt University and his graduate degrees from The University of Texas at Austin. He is a professor in the Department of Civil and Environmental Engineering at The Virginia Military Institute. He is a licensed engineering in the state of Virginia.