Int. J. Engng Ed. Vol. 8, No. 4, pp. 271–277, 1992
Printed in Great Britain.
0742–0269/91 $3.00+0.00

Three Methods for Performing Conditional Looping in a Spreadsheet*

C. R. THOMAS

Mechanical Engineering Technology, Purdue University, West Lafayette, Indiana 47907, *U.S.A.*

Three different methods of accomplishing conditional looping with an Excel spreadsheet are considered. The basic algorithms include a table lookup method, an iteration method, and a macro method; each method is developed in terms of the same simple conditional looping applied problem. Features and advantages of each of the methods are presented. The resulting richness of alternatives to performing conditional looping should allow spreadsheet users to easily select an appropriate and understandable method of conditional looping in the solution of new problems.

The paper should be of interest to engineering and engineering technology faculty who either teach spreadsheet programming to their students, or who require their students to program or develop problem-solving spreadsheets.

SUMMARY OF EDUCATIONAL ASPECTS OF THE PAPER

1. The paper describes software applications useful in the following engineering disciplines:

   All engineering and engineering technology disciplines.

2. The paper is suitable for teaching/classwork/self-study for engineering students at the following level:

   Both undergraduate and graduate levels.

3. What aspects of your contributions are new?

   All three methods have new or novel aspects in the presented algorithms which the reader or his students can incorporate in the solution of their own spreadsheet programming endeavours.

4. How is the material as presented to be incorporated in engineering teaching?

   The material may be incorporated directly into technical spreadsheet programming courses or alternately, may be made available to courses which require students to program or develop their own problem-solving spreadsheets.

5. Have the concepts presented been tested in the classroom or in project work? What conclusions have been drawn from the experience?

   Yes, to a limited extent in technology student computer programming projects in machine elements with select, spreadsheet capable students. Students need a formal introduction to the methods—a current goal is to expand the contact hours and content of our current introductory technology spreadsheet course to allow for the inclusion of the discussed three methods as well as other advanced level spreadsheet programming topics.

6. Other comments on the benefits of your approach for engineering education:

   A firm user-understanding of performing conditional looping with a spreadsheet, together with practical alternative choices of conditional looping algorithms allows spreadsheet programming to be used as a viable alternative to other high order languages such as FORTRAN, BASIC, and Pascal in engineering and engineering technology education.

INTRODUCTION

Three different methods of accomplishing conditional looping within an Excel spreadsheet are considered in terms of a typical problem, the solution of which requires conditional looping. These three methods or algorithm types are the table lookup method, the iteration method, and the macro method; each method is uniquely different and exists because of the wealth of commands, functions and structures available in modern spreadsheets. The same typical problem is used to illustrate an application of each method to a solution of a spreadsheet problem of conditional looping, thus avoiding ancillary considerations that might arise from a consideration of different problems for each method.

* Paper accepted 15 October 1992.

The conditional looping problem presented in all three spreadsheet approaches was selected because of its simplicity and illustrative value. The primary focus is on the three conditional looping algorithms, and consequently possible spreadsheet features, enhancements, and user instructions have not been included in the spreadsheets presented.

Basic and advanced references on the Excel spreadsheet such as Cobb [1], Townsend [2], Cobb and Mynhier [3], Matthews [4], Person and Campbell [5], Kyd [6], Hixson [7], Annaloro [8], and Eliason and Krumm [9] are useful for learning the basics of Excel and for simple examples of command and function syntax. However, the target audience of such references is fairly diverse and hence there are few, if any, developed problems and algorithms which address the more advanced needs of engineering and technology users of spreadsheets. Consequently, the conditional looping algorithms presented here should serve as a foundation for the solution of similar, and more complex problems encountered across the specialized disciplines in both engineering and technology.

A TYPICAL PROBLEM

A fairly simple problem from machine design or strength of materials will be used to illustrate the three alternatives to conditional looping in a spreadsheet. The problem is that of designing or selecting the diameter of a shaft subject to a torque and an axial load. The axial normal stress for the axial load acting alone is:

$$S = \frac{4P}{\pi d^2} \tag{1}$$

where S is in psi, P is the axial load in pounds, and d is the shaft diameter in inches. The torsional stress for the torque acting alone is:

$$S_s = \frac{16T}{\pi d^3} \tag{2}$$

where S_S is in psi and T is the torque in inch-pounds.

The actual stresses in the shaft have values larger than these simple stresses, and these actual stresses can be computed with the aid of two principle or maximum stress equations. The maximum shear stress is found with the equation:

$$S_{s\max} = \surd(S^2 + 4S_s^2)/2 \tag{3}$$

where $S_{s\max}$ is the maximum shear stress in the shaft in psi. The maximum normal stress is found with the equation:

$$S_{\max} = (S/2) + S_{s\max} \tag{4}$$

where $S_{\max}$ is the maximum normal stress in psi. Shaft diameter must be large enough to assure that the maximum normal and maximum shear stresses are less than, or at most equal to, a set of design normal and shear stresses for the shaft material. These design stresses result in the constraints:

$$S_{s\max} < S_{s\text{design}} \tag{5}$$

$$S_{\max} < S_{\text{design}} \tag{6}$$

where $S_{s\text{design}}$ is the design shear stress for the shaft material in psi and S_{design} is the design normal stress for the shaft material in psi.

In FORTRAN, BASIC, or Pascal, this type of problem would be set up in a conditional loop which varies diameter and uses the two maximum stress conditions as loop exit criteria.

TABLE LOOKUP METHOD

The table lookup procedure is fairly straightforward and is accomplished entirely on a worksheet. This procedure is not actually a loop, but instead uses the satisfaction of the design stress constraints, to set a solution index or counter, which in turn is used to search for the solution in a table which contains all possible solution cases.

The table lookup procedure is discussed in terms of an Excel spreadsheet which assumes a shaft diameter solution in the range of 0.5–1.0 inch with standard shafts available in 0.0625 inch increments. The limited range of shaft diameters in the example spreadsheet was selected to keep the size of the spreadsheet presented here small; in practice a much larger selection of shaft sizes would simply result in a correspondingly larger spreadsheet. Data entry is in the spreadsheet range A5:B12 as shown in Fig. 1 with the actual stress, load, and torque data entered in the range B9:B12. Problem solutions for all cases of diameter, checks on the satisfaction of the stress constraints, and selection of a solution index are found in cells D1:N14 as shown in Fig. 2.

Shaft diameters are specified in cell E4:E12, stresses and maximum stresses are calculated in cells G4:J12, satisfaction of design stresses is checked in cells K4:M12, solution indexes used to select the actual solution are given in cells D4:D12, F4:F12, and N4:N12 and the index for the actual solution is selected in cell E14. Formulas for the cells in Fig. 2 are shown in Figs 3–6.

The basic logic for the table lookup algorithm is generated in row 4 and once it is working properly, it is copied to rows 5 to 12 using the Excel commands **edit copy** and **edit paste**. Index columns D, F, and N together with diameter column E are generated with the Excel command **data series** with step values of 1 and 0.0625 respectively. In all cases, formula references to the input data in cells B9:B12 are absolute references; all other cell references are relative references. Normal stresses, Eq. (1), and shear stresses, Eq. (2) are computed in columns G and H respectively. Maximum shear stresses, Eq. (3), and maximum normal stresses, Eq. (4), are computed in columns I and J respectively. Using YES for satisfied and NO for not

	A	B
5	GIVENS:	
6	**	***********
7	VARIABLE	VALUE
8	**	***********
9	ALLOWABLE (DESIGN) NORMAL STRESS (PSI)	15000
10	ALLOWABLE (DESIGN) SHEAR STRESS (PSI)	11000
11	TENSILE (COMPRESSIVE) LOAD (POUNDS)	2850
12	TORQUE (INCH-POUNDS)	1250

Fig. 1. Data entry.

	D	E	F	G	H	I	J	K	L	M	N
1						MAX	MAX				
2				NORMAL	SHEAR	SHEAR	NORMAL	SS	NS	SS & NS	
3	INDEX	DIAM	INDEX	STRESS	STRESS	STRESS	STRESS	COND	COND	COND	INDEX
4	1	0.5	1	14514.9	50929.6	51444.1	58701.5	NO	NO	NO	1
5	2	0.5625	2	11468.6	35769.5	36226.2	41960.5	NO	NO	NO	2
6	3	0.625	3	9289.6	26075.9	26486.4	31131.2	NO	NO	NO	3
7	4	0.6875	4	7677.3	19591.2	19963.8	23802.4	NO	NO	NO	4
8	5	0.75	5	6451.1	15090.2	15431.1	18656.7	NO	NO	NO	5
9	6	0.8125	6	5496.8	11868.9	12182.9	14931.3	NO	YES	NO	6
10	7	0.875	7	4739.6	9502.9	9793.9	12163.7	YES	YES	YES	7
11	8	0.9375	8	4128.7	7726.2	7997.2	10061.6	YES	YES	YES	8
12	9	1	9	3628.7	6366.2	6619.7	8434.1	YES	YES	YES	9
13											
14	INDEX=>	7									

Fig. 2. Solution table and solution index.

	D	E	F	G	H
1					
2				NORMAL	SHEAR
3	INDEX	DIAM	INDEX	STRESS	STRESS
4	1	0.5	1	=$B$11/(PI()*(E4^2)/4)	=16*$B$12/(PI()*(E4^3))
5	2	0.5625	2	=$B$11/(PI()*(E5^2)/4)	=16*$B$12/(PI()*(E5^3))
6	3	0.625	3	=$B$11/(PI()*(E6^2)/4)	=16*$B$12/(PI()*(E6^3))
7	4	0.6875	4	=$B$11/(PI()*(E7^2)/4)	=16*$B$12/(PI()*(E7^3))
8	5	0.75	5	=$B$11/(PI()*(E8^2)/4)	=16*$B$12/(PI()*(E8^3))
9	6	0.8125	6	=$B$11/(PI()*(E9^2)/4)	=16*$B$12/(PI()*(E9^3))
10	7	0.875	7	=$B$11/(PI()*(E10^2)/4)	=16*$B$12/(PI()*(E10^3))
11	8	0.9375	8	=$B$11/(PI()*(E11^2)/4)	=16*$B$12/(PI()*(E11^3))
12	9	1	9	=$B$11/(PI()*(E12^2)/4)	=16*$B$12/(PI()*(E12^3))

Fig. 3. Formulas in table cells D to H.

	I	J	K
1	MAX	MAX	
2	SHEAR	NORMAL	SS
3	STRESS	STRESS	COND
4	=SQRT((G4^2)+(4*(H4^2)))/2	=(G4/2)+I4	=IF(I4<=$B$10,"YES","NO")
5	=SQRT((G5^2)+(4*(H5^2)))/2	=(G5/2)+I5	=IF(I5<=$B$10,"YES","NO")
6	=SQRT((G6^2)+(4*(H6^2)))/2	=(G6/2)+I6	=IF(I6<=$B$10,"YES","NO")
7	=SQRT((G7^2)+(4*(H7^2)))/2	=(G7/2)+I7	=IF(I7<=$B$10,"YES","NO")
8	=SQRT((G8^2)+(4*(H8^2)))/2	=(G8/2)+I8	=IF(I8<=$B$10,"YES","NO")
9	=SQRT((G9^2)+(4*(H9^2)))/2	=(G9/2)+I9	=IF(I9<=$B$10,"YES","NO")
10	=SQRT((G10^2)+(4*(H10^2)))/2	=(G10/2)+I10	=IF(I10<=$B$10,"YES","NO")
11	=SQRT((G11^2)+(4*(H11^2)))/2	=(G11/2)+I11	=IF(I11<=$B$10,"YES","NO")
12	=SQRT((G12^2)+(4*(H12^2)))/2	=(G12/2)+I12	=IF(I12<=$B$10,"YES","NO")

Fig. 4. Formulas in table cells I to K.

The conditional looping problem presented in all three spreadsheet approaches was selected because of its simplicity and illustrative value. The primary focus is on the three conditional looping algorithms, and consequently possible spreadsheet features, enhancements, and user instructions have not been included in the spreadsheets presented.

Basic and advanced references on the Excel spreadsheet such as Cobb [1], Townsend [2], Cobb and Mynhier [3], Matthews [4], Person and Campbell [5], Kyd [6], Hixson [7], Annaloro [8], and Eliason and Krumm [9] are useful for learning the basics of Excel and for simple examples of command and function syntax. However, the target audience of such references is fairly diverse and hence there are few, if any, developed problems and algorithms which address the more advanced needs of engineering and technology users of spreadsheets. Consequently, the conditional looping algorithms presented here should serve as a foundation for the solution of similar, and more complex problems encountered across the specialized disciplines in both engineering and technology.

A TYPICAL PROBLEM

A fairly simple problem from machine design or strength of materials will be used to illustrate the three alternatives to conditional looping in a spreadsheet. The problem is that of designing or selecting the diameter of a shaft subject to a torque and an axial load. The axial normal stress for the axial load acting alone is:

$$S = \frac{4P}{\pi d^2} \tag{1}$$

where S is in psi, P is the axial load in pounds, and d is the shaft diameter in inches. The torsional stress for the torque acting alone is:

$$S_s = \frac{16T}{\pi d^3} \tag{2}$$

where S_S is in psi and T is the torque in inch-pounds.

The actual stresses in the shaft have values larger than these simple stresses, and these actual stresses can be computed with the aid of two principle or maximum stress equations. The maximum shear stress is found with the equation:

$$S_{s\max} = \sqrt{(S^2 + 4S_s^2)}/2 \tag{3}$$

where $S_{s\max}$ is the maximum shear stress in the shaft in psi. The maximum normal stress is found with the equation:

$$S_{\max} = (S/2) + S_{s\max} \tag{4}$$

where $S_{\max}$ is the maximum normal stress in psi. Shaft diameter must be large enough to assure that the maximum normal and maximum shear stresses are less than, or at most equal to, a set of design normal and shear stresses for the shaft material. These design stresses result in the constraints:

$$S_{s\max} < S_{s\text{design}} \tag{5}$$

$$S_{\max} < S_{\text{design}} \tag{6}$$

where $S_{s\text{design}}$ is the design shear stress for the shaft material in psi and S_{design} is the design normal stress for the shaft material in psi.

In FORTRAN, BASIC, or Pascal, this type of problem would be set up in a conditional loop which varies diameter and uses the two maximum stress conditions as loop exit criteria.

TABLE LOOKUP METHOD

The table lookup procedure is fairly straightforward and is accomplished entirely on a worksheet. This procedure is not actually a loop, but instead uses the satisfaction of the design stress constraints, to set a solution index or counter, which in turn is used to search for the solution in a table which contains all possible solution cases.

The table lookup procedure is discussed in terms of an Excel spreadsheet which assumes a shaft diameter solution in the range of 0.5–1.0 inch with standard shafts available in 0.0625 inch increments. The limited range of shaft diameters in the example spreadsheet was selected to keep the size of the spreadsheet presented here small; in practice a much larger selection of shaft sizes would simply result in a correspondingly larger spreadsheet. Data entry is in the spreadsheet range A5:B12 as shown in Fig. 1 with the actual stress, load, and torque data entered in the range B9:B12. Problem solutions for all cases of diameter, checks on the satisfaction of the stress constraints, and selection of a solution index are found in cells D1:N14 as shown in Fig. 2.

Shaft diameters are specified in cell E4:E12, stresses and maximum stresses are calculated in cells G4:J12, satisfaction of design stresses is checked in cells K4:M12, solution indexes used to select the actual solution are given in cells D4:D12, F4:F12, and N4:N12 and the index for the actual solution is selected in cell E14. Formulas for the cells in Fig. 2 are shown in Figs 3–6.

The basic logic for the table lookup algorithm is generated in row 4 and once it is working properly, it is copied to rows 5 to 12 using the Excel commands **edit copy** and **edit paste**. Index columns D, F, and N together with diameter column E are generated with the Excel command **data series** with step values of 1 and 0.0625 respectively. In all cases, formula references to the input data in cells B9:B12 are absolute references; all other cell references are relative references. Normal stresses, Eq. (1), and shear stresses, Eq. (2) are computed in columns G and H respectively. Maximum shear stresses, Eq. (3), and maximum normal stresses, Eq. (4), are computed in columns I and J respectively. Using YES for satisfied and NO for not

	A	B
5	GIVENS:	
6	**	************
7	VARIABLE	VALUE
8	**	************
9	ALLOWABLE (DESIGN) NORMAL STRESS (PSI)	15000
10	ALLOWABLE (DESIGN) SHEAR STRESS (PSI)	11000
11	TENSILE (COMPRESSIVE) LOAD (POUNDS)	2850
12	TORQUE (INCH-POUNDS)	1250

Fig. 1. Data entry.

	D	E	F	G	H	I	J	K	L	M	N
1						MAX	MAX				
2				NORMAL	SHEAR	SHEAR	NORMAL	SS	NS	SS & NS	
3	INDEX	DIAM	INDEX	STRESS	STRESS	STRESS	STRESS	COND	COND	COND	INDEX
4	1	0.5	1	14514.9	50929.6	51444.1	58701.5	NO	NO	NO	1
5	2	0.5625	2	11468.6	35769.5	36226.2	41960.5	NO	NO	NO	2
6	3	0.625	3	9289.6	26075.9	26486.4	31131.2	NO	NO	NO	3
7	4	0.6875	4	7677.3	19591.2	19963.8	23802.4	NO	NO	NO	4
8	5	0.75	5	6451.1	15090.2	15431.1	18656.7	NO	NO	NO	5
9	6	0.8125	6	5496.8	11868.9	12182.9	14931.3	NO	YES	NO	6
10	7	0.875	7	4739.6	9502.9	9793.9	12163.7	YES	YES	YES	7
11	8	0.9375	8	4128.7	7726.2	7997.2	10061.6	YES	YES	YES	8
12	9	1	9	3628.7	6366.2	6619.7	8434.1	YES	YES	YES	9
13											
14	INDEX=>	7									

Fig. 2. Solution table and solution index.

	D	E	F	G	H
1					
2				NORMAL	SHEAR
3	INDEX	DIAM	INDEX	STRESS	STRESS
4	1	0.5	1	=$B$11/(PI()*(E4^2)/4)	=16*$B$12/(PI()*(E4^3))
5	2	0.5625	2	=$B$11/(PI()*(E5^2)/4)	=16*$B$12/(PI()*(E5^3))
6	3	0.625	3	=$B$11/(PI()*(E6^2)/4)	=16*$B$12/(PI()*(E6^3))
7	4	0.6875	4	=$B$11/(PI()*(E7^2)/4)	=16*$B$12/(PI()*(E7^3))
8	5	0.75	5	=$B$11/(PI()*(E8^2)/4)	=16*$B$12/(PI()*(E8^3))
9	6	0.8125	6	=$B$11/(PI()*(E9^2)/4)	=16*$B$12/(PI()*(E9^3))
10	7	0.875	7	=$B$11/(PI()*(E10^2)/4)	=16*$B$12/(PI()*(E10^3))
11	8	0.9375	8	=$B$11/(PI()*(E11^2)/4)	=16*$B$12/(PI()*(E11^3))
12	9	1	9	=$B$11/(PI()*(E12^2)/4)	=16*$B$12/(PI()*(E12^3))

Fig. 3. Formulas in table cells D to H.

	I	J	K
1	MAX	MAX	
2	SHEAR	NORMAL	SS
3	STRESS	STRESS	COND
4	=SQRT((G4^2)+(4*(H4^2)))/2	=(G4/2)+I4	=IF(I4<=$B$10,"YES","NO")
5	=SQRT((G5^2)+(4*(H5^2)))/2	=(G5/2)+I5	=IF(I5<=$B$10,"YES","NO")
6	=SQRT((G6^2)+(4*(H6^2)))/2	=(G6/2)+I6	=IF(I6<=$B$10,"YES","NO")
7	=SQRT((G7^2)+(4*(H7^2)))/2	=(G7/2)+I7	=IF(I7<=$B$10,"YES","NO")
8	=SQRT((G8^2)+(4*(H8^2)))/2	=(G8/2)+I8	=IF(I8<=$B$10,"YES","NO")
9	=SQRT((G9^2)+(4*(H9^2)))/2	=(G9/2)+I9	=IF(I9<=$B$10,"YES","NO")
10	=SQRT((G10^2)+(4*(H10^2)))/2	=(G10/2)+I10	=IF(I10<=$B$10,"YES","NO")
11	=SQRT((G11^2)+(4*(H11^2)))/2	=(G11/2)+I11	=IF(I11<=$B$10,"YES","NO")
12	=SQRT((G12^2)+(4*(H12^2)))/2	=(G12/2)+I12	=IF(I12<=$B$10,"YES","NO")

Fig. 4. Formulas in table cells I to K.

	L	M	N
1			
2	NS	SS & NS	
3	COND	COND	INDEX
4	=IF(J4<=$B$9,"YES","NO")	=IF(AND(K4="YES",L4="YES"),"YES","NO")	1
5	=IF(J5<=$B$9,"YES","NO")	=IF(AND(K5="YES",L5="YES"),"YES","NO")	2
6	=IF(J6<=$B$9,"YES","NO")	=IF(AND(K6="YES",L6="YES"),"YES","NO")	3
7	=IF(J7<=$B$9,"YES","NO")	=IF(AND(K7="YES",L7="YES"),"YES","NO")	4
8	=IF(J8<=$B$9,"YES","NO")	=IF(AND(K8="YES",L8="YES"),"YES","NO")	5
9	=IF(J9<=$B$9,"YES","NO")	=IF(AND(K9="YES",L9="YES"),"YES","NO")	6
10	=IF(J10<=$B$9,"YES","NO")	=IF(AND(K10="YES",L10="YES"),"YES","NO")	7
11	=IF(J11<=$B$9,"YES","NO")	=IF(AND(K11="YES",L11="YES"),"YES","NO")	8
12	=IF(J12<=$B$9,"YES","NO")	=IF(AND(K12="YES",L12="YES"),"YES","NO")	9

Fig. 5. Formulas in table cells L to N.

	D	E
14	INDEX=>	=VLOOKUP("NO",M4:N12,2)+1

Fig. 6. Table solution index.

satisfied, satisfaction of the design stress constraints, Eqs (5) and (6), is determined with the Excel **if** function and is reported in columns K and L and their simultaneous satisfaction is determined with an Excel logical **and** function inside of an **if** function and is reported in column M.

The basic algorithm for finding a design diameter solution is based on the index reported in cell E14. A search for the bottommost NO in the table M4:N12 is made with the table **vlookup** function and 1 plus the index found is recorded as the solution index in cell E14. This solution index is then in turn used to look up the resulting solution as reported in Figs 7 and 8. Normal stress, shear stress, maximum normal stress, maximum shear stress, and shaft diameter are respectively reported in cell B18:B22. The Figs 7 and 8 cells show the **vlookup** function using the solution index to look up this solution.

If data is directly entered into cells B9:B12, Excel will recalculate all spreadsheet values after each cell data entry, this recalculation can be quite lengthy for large tables. To avoid immediate cell recalculation of each data entry, calculation may be suspended until all data entry is completed. This is accomplished by using **options calculation** and selecting **manual**, typing all input data into appropriate cells, then using **options calculate now**.

	B
15	************************************
16	VALUE
17	************************************
18	=VLOOKUP($E$14,F4:G12,2)
19	=VLOOKUP($E$14,F4:H12,3)
20	=VLOOKUP($E$14,F4:J12,5)
21	=VLOOKUP($E$14,F4:I12,4)
22	=VLOOKUP($E$14,D4:E12,2)

Fig. 8. Formulas in table solution cells,

There are several advantages to using table lookup to perform what normally would be a conditional looping procedure. First, algorithm development is entirely on a worksheet, familiarity with Macros is not required and since all cases are included in the table developed, both algorithm development and debugging are usually quite straightforward. Secondly, assuming a problem of known solution is used in spreadsheet development and debugging, cells with formula and/or logic errors are immediately obvious. The major disadvantage to this method is for problems with very large numbers of cases, or repetitions, with a correspondingly large table, which might exceed the limits of available computer memory.

ITERATION METHOD

The iteration procedure is not always straightforward, but it is accomplished entirely on a worksheet. The procedure takes advantage of the Ecel

	A	B
14	RESULTS:	
15	**	****************
16	VARIABLE	VALUE
17	**	****************
18	TENSION(COMPRESSION) NORMAL STRESS (PSI)	4739.569
19	TORSION SHEAR STRESS(PSI)	9502.896
20	MAXIMUM NORMAL STRESS (PSI)	12163.707
21	MAXIMUM SHEAR STRESS (PSI)	9793.922
22	SHAFT DIAMETER (INCHES)	0.875

Fig. 7. Table solution results.

iteration option available through **options calculation**.

The iteration procedure is discussed in terms of the same available shaft diameter range used in the table procedure and data entry is exactly the same as shown in Fig. 1. The solution is reported, as shown in Fig. 9, in a format almost identical to that in Fig. 7 for table lookup. Fig. 10 shows the primary difference from the table solution, that is the contents of cells B19:B24, which now contain all of the formulas and logic involved in arriving at a solution. Normal stress, Eq. 1, and shear stress, Eq. 2, are computed in cells B19 and B20 respectively. Maximum shear stress, Eq. 3, and maximum normal stress, Eq. 4, are computed in cells B21 and B22 respectively. A diameter initialization, incrementation, and new problem reset strategy is set in cell B23. This strategy is vital to a successful problem solution. On loading the iteration spreadsheet into Excel, a dialog box may appear on the screen with the message 'Can't Resolve Circular Reference'. This means that iteration has not been selected and can be remedied by selecting from the menu bar **options calculation** and selecting **iteration** in the resulting dialog box.

At this point, it is worthwhile discussing possible contents of the diameter cell B23 and consequent spreadsheet outcomes. While it might seem sufficient, an entry in cell B23 of the form:

=IF(OR(B22>B10,B21>B9),B23+0.0625,B23) (7)

will result in the message '$DIV/0!' in cells B19:B23. This results because Excel initially sets the cell or cells responsible for the circular reference equal, in this case cell B23, to zero; in the current problem this results in division by a zero diameter in cells B19:B20, the effects of which also carry into cells B21:B23. Furthermore, even if cell B23 was not involved in division, its first value under iteration would be 0.0625 and not 0.5 as desired. Both of these problems can be remedied by using the **iserr** error function inside of a logical **or** function to modify the formula for cell B23 as:

=IF(OR(ISERR(B23),ISERR(B19),ISERR(B20),ISERR(B21),ISERR(B22)),0.5,(IF(OR(B22>B10,B21>B9),B23+0.0625,B23))) (8)

which at first glance resolves the problem. Everything certainly works quite well the first time through the problem. However the last solution of 0.875 inches in cell B23, becomes the starting value for the next round of iteration for a new problem which unfortunately only works properly if the solution is greater than, or equal, to 0.875 inches.

This dilemma is resolved by adding a reinitializing feature consisting of the new cells A24:A25 and B24 along with the inclusion of the additional term B24<>"BEGIN" inside the first logical **or** in the equation for cell B23 as shown in Fig. 10. Before starting a new problem solution, the user now must first type the word 'RESET' into cell B24 and wait until the resulting iteration ceases. Typing 'BEGIN' in cell B24 then begins a new iteration with a starting diameter of 0.5 inches.

To again avoid immediate cell recalculation upon data entry, calculation may be suspended until all data entry is completed by choosing **options calculation** and selecting **manual**. This is also a good time to select **iteration** in the dialog box and to look at the other dialog box options. The **maximum iterations** may be set at 9, since only 9 different shaft diameters are considered, and **maximum change** has no real effect on the current

	A	B
15	RESULTS:	
16	**	************
17	VARIABLE	VALUE
18	**	************
19	TENSION(COMPRESSION) NORMAL STRESS (PSI)	4739.569
20	TORSION SHEAR STRESS(PSI)	9502.896
21	MAXIMUM NORMAL STRESS (PSI)	12163.71
22	MAXIMUM SHEAR STRESS (PSI)	9793.922
23	SHAFT DIAMETER (INCHES)	0.875
24	TYPE BEGIN AND HIT THE ENTER KEY TO START>>>>>>	BEGIN
25	RESET AND HIT THE ENTER KEY TO RESET	

Fig. 9. Iteration solution.

	B
19	=$B$11/(PI()*((B23)^2)/4)
20	=16*$B$12/(PI()*(B23^3))
21	=(B19/2)+B22
22	=SQRT((B19^2)+(4*(B20^2)))/2
23	=IF(OR(B24<>"BEGIN",ISERR(B23),ISERR(B19),ISERR(B20),ISERR(B21),ISERR(B22)),0.5,(IF(OR(B22>B10,B21>B9),B23+0.0625,B23)))
24	BEGIN

Fig. 10. Formulas in iteration solution cells.

problem. Once data entry is complete, **options calculation** with **automatic** selected will restore immediate calculation. Typing 'RESET' in cell B24 followed by 'BEGIN' results in a solution for the newly entered problem.

MACRO METHOD

The macro procedure is not always straightforward and it requires the development of a macro sheet in addition to the main worksheet. The procedure takes advantage of the inherent power of Excel's macro commands.

Data entry is exactly the same as for the table method and the iteration method as shown in Fig. 1. The solution is reported in exactly the same form as for the table method as shown in Fig. 7, except now cells B18:B22 do not contain any formulas. The macro actually returns solution values directly to cells B18:B22. The macro sheet is shown in Fig. 11.

While there are many possible styles and preferences for macro format, the advantages behind the current macro format will be detailed. The macro itself is developed in the A column and values computed by the macro are recorded in the B column by using the Excel **set.value** function. This strategy allows the formulas in the macro column to be viewed by selecting **options display formulas** while the computed numerical values are simultaneously available, this situation is extremely helpful in macro debugging.

Cells A2:A3 set initially high values for the calculated maximum stresses to assure a proper start of the macro iteration process. Cells A4:A5 and A8:A9 import allowable stress, load, and torque values from the data input worksheet named EXMACRO.XLS while cells A6:A7 and A10:A11 place these values in macro cells B4:B5 and B6:B7 respectively. Cells A12:A13 set the pre-loop diameter at 0.5 minus the 0.0625 increment. Iteration towards a solution is accomplished with the **while** loop that spans the cells A14:A20; a logical **or** function inside the **while** function in cell A14 checks for satisfaction of the Eqs 5 and 6 design stress constraints. Cells A15:A19 iterate the diameter and place calculated values of stresses and maximum stresses in cells B10:B11 and B2:B3 respectively. The **formula** functions in cells B21:B25 return solution stresses and diameter to the worksheet EXMACRO.XLS.

The macro is declared as a command macro and can be invoked by simultaneously depressing the CTRL key and the 'a' key. Specification of the destination worksheet in the **formula** functions allows the macro to be run and tested when the macro is the active sheet, and still return the argument values to the main worksheet, rather than to the macro sheet itself. Since the macro does not become active until called, a switch to manual calculation is not required prior to data entry.

Obviously, some familiarity with macros is required to develop a conditional iteration algorithm with macros. The macro method algorithm is structurally similar to the type of algorithm one

	A	B	C
1	EXMAC1		
2	=SET.VALUE(B2,1000000)	12163.70	NS MAX CALC
3	=SET.VALUE(B3,1000000)	9793.921	SS MAX CALC
4	=EXMACRO.XLS!B9	15000	NS MAX CRIT
5	=EXMACRO.XLS!B10	11000	SS MAX CRIT
6	=SET.VALUE(B4,A4)	2850	LOAD
7	=SET.VALUE(B5,A5)	1250	TORQUE
8	=EXMACRO.XLS!B11	0.4375	START DIAM
9	=EXMACRO.XLS!B12	0.875	DIAM
10	=SET.VALUE(B6,A8)	4739.569	NS
11	=SET.VALUE(B7,A9)	9502.895	SS
12	=SET.VALUE(B8,0.4375)		
13	=SET.VALUE(B9,B8)		
14	=WHILE((OR(B2>B4,B3>B5))=TRUE)		
15	=SET.VALUE(B9,B9+0.0625)		
16	=SET.VALUE(B10,B6/(PI()*(B9^2)/4))		
17	=SET.VALUE(B11,16*B7/(PI()*(B9^3)))		
18	=SET.VALUE(B3,SQRT((B10^2)+(4*(B11^2)))/2)		
19	=SET.VALUE(B2,(B10/2)+B3)		
20	=NEXT()		
21	=FORMULA(B10,"EXMACRO.XLS!R18C2")		
22	=FORMULA(B11,"EXMACRO.XLS!R19C2")		
23	=FORMULA(B2,"EXMACRO.XLS!R20C2")		
24	=FORMULA(B3,"EXMACRO.XLS!R21C2")		
25	=FORMULA(B9,"EXMACRO.XLS!R22C2")		
26	=RETURN()		

Fig. 11. The macro sheet.

familiar with languages such as BASIC, FORTRAN, and Pascal would develop for a conditional looping problem. This fact might well make the macro method the most likely method of spreadsheet development for individuals with a higher order programming language background.

CONCLUSIONS

Spreadsheet algorithms for conditional looping based on table lookup, the iteration option of calculate and macros demonstrate the diversity of programming choices available within the Excel spreadsheet. This richness of alternatives should allow beginning, intermediate, and advanced spreadsheet users to select a method of conditional looping which they can utilize in the solution of their own problems.

Spreadsheet developers should keep an open mind regarding the table lookup, iteration, and macro algorithms for conditional looping as any one of these methods may turn out to be the most suitable method for matching the unique characteristics of a new problem. Some complex problems may be best solved using a combination of two or more of the basic algorithms discussed. If spreadsheets are developed for use by others, end-user familiarity and level of usage ability regarding spreadsheets, may be an important consideration in the selection of a conditional looping algorithm; particularly in terms of user manipulation and operations required to successfully run the resulting spreadsheet.

REFERENCES

1. S. Cobb, *The Stephen Cobb User's Handbook to Excel for the IBM*, Windcrest Books, Pennsylvania (1989).
2. C. Townsend, *Mastering Excel 3 for Windows*, Sybex Inc., California (1991).
3. D. Cobb and J. Mynhier, *Running Microsoft Excel*, Microsoft Press, Washington (1988).
4. M. S. Matthews, *Excel 2.1 Made Easy: IBM PC Version*, Osborne McGraw-Hill, California (1990).
5. R. Person and M. Campbell, *Using Excel: IBM Version*, Que Corporation, Indiana (1988).
6. C. W. Kyd, *The Microsoft Excel Business Sourcebook*, Microsoft Press, Washington (1988).
7. A. C. Hixson, *Advanced Excel for the PC*, Osborn McGraw-Hill, California (1988).
8. J. Annaloro, *Excel Macros for the IBM*, Scott, Foresman and Company, Illinois (1989).
9. A. L. Eliason and R. Krumm, *Microsoft Excel Student Edition*, Prentice Hall, Inc., New Jersey (1989).

Charles R. Thomas is a Professor of Mechanical Engineering Technology at Purdue University, where he has taught since 1976. He received the BS Degree in Mechanical Engineering from Clarkson College in 1966, and the MS Degree in Mechanics from Rensselaer Polytechnic Institute in 1969. From 1969–76 he worked as a mechanical engineer at the Benet Weapons Laboratory at Watervliet Arsenal. Charles is a member of ASEE, ASME, and AAM. He has been the recipient of ten awards for excellence in teaching and oustanding undergraduate teaching, and is the author of over 90 papers and publications in the areas of solid mechanics, educational measurement, technical education, computer progamming in technology, technology student characteristics, field-independence, and the Myers–Briggs type indicator.