Engineering Design Using Microcomputer-Based Spreadsheets

The first spreadsheet designed for use on microcomputers was introduced in 1978. Since that time, business managers have made extensive use of these new electronic "tools," resulting in documented increases in productivity. Practicing engineers have been relatively slow in adopting this technology. Specifically, little use has been made of the powerful graphics capabilities these programs possess.

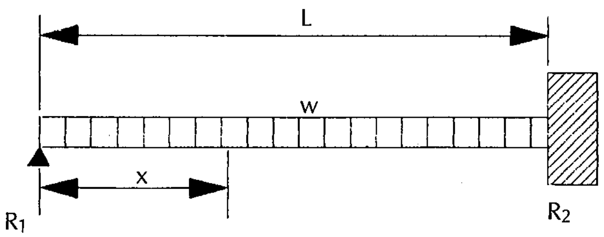
W. LEE SHOEMAKER & STEVE WILLIAMS

AS THE USE of microcomputers, or personal computers (PCs), has proliferated over the last ten years, civil engineers have put them to use in applications that require extensive and complicated calculations in fields such as structural engineering, hydrology, *etc.*

Engineers can use high-level programming languages (such as FORTRAN, Pascal, C, *etc.*) to write their own applications. The actual time spent writing, testing and debugging programs developed by engineers or software specialists, however, may not be cost-effective. Currently available commercial programming environments and program generators may reduce program development time, but it still may not be to the advantage of either the engineer or engineering firm to be in the "software development" business.

While off-the-shelf engineering software applications are available, they are in most cases too costly, or are either too broad or narrow in scope to meet the specific engineering requirements.

However, some engineers have begun to use commercial spreadsheet software that is readily available for use on personal computers for various engineering applications. Although the similarities between ledger calculations that this software was originally developed for and the solution of engineering equations might not be obvious, engineers are beginning to realize that spreadsheet concepts can be readily adapted to almost any manipulation of numeri-



$R_1 = (3wL)/8$

$R_2 = (5wL)/8$

$V_x = R_1 - wx$

$M_x = R_1x - (wx^2)/2$

$\Delta_x = ((wx)/48EI)(L^3 - 3Lx^2 + 2x^3)$

FIGURE 1. Beam formulas.

cal data.[1,2,3,4,5]

Spreadsheets are also being utilized as effective teaching tools, primarily because these programs offer the unique opportunity for changing parameters in order to see their effects on the solution.[6,7] Even though the quantity of spreadsheet applications in engineering design that are being reported has been increasing over the past few years, relatively little mention has been made of the type of design applications that can be readily handled by using the graphics capabilities built into most popular commercial PC spreadsheets.

Spreadsheet Fundamentals

An electronic spreadsheet is simply a matrix of

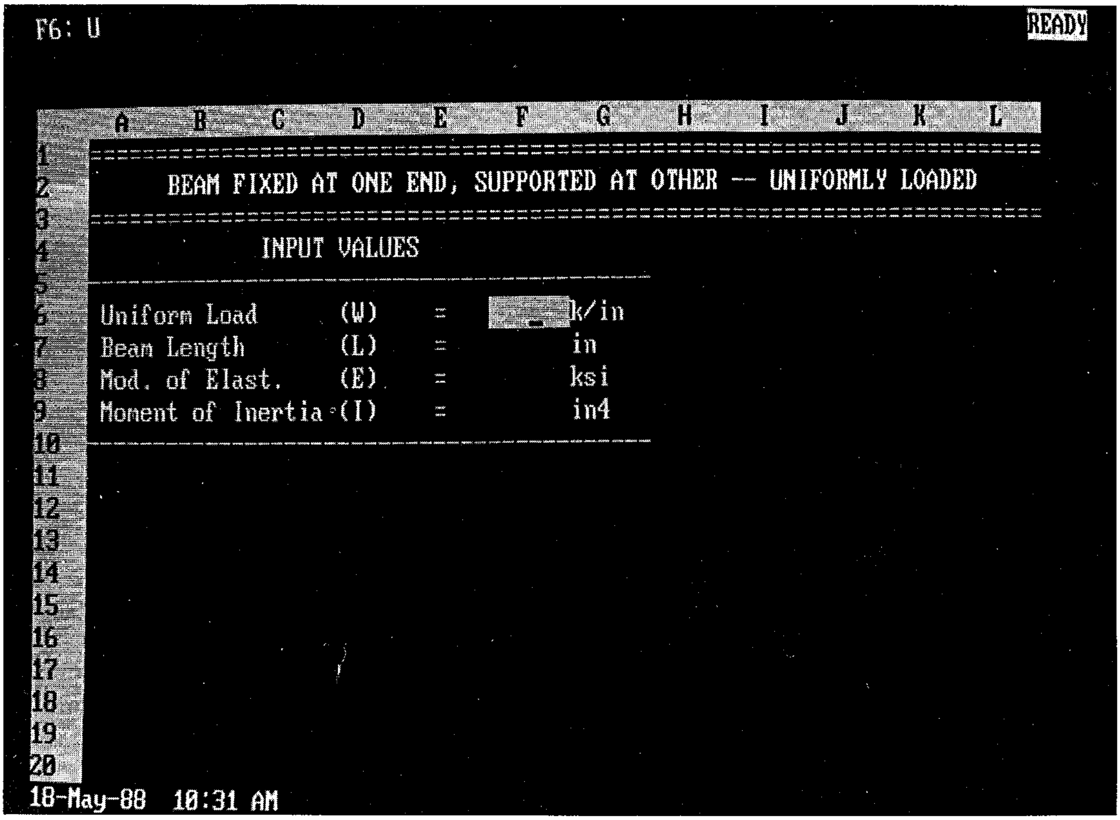


FIGURE 2. Input area of the spreadsheet.

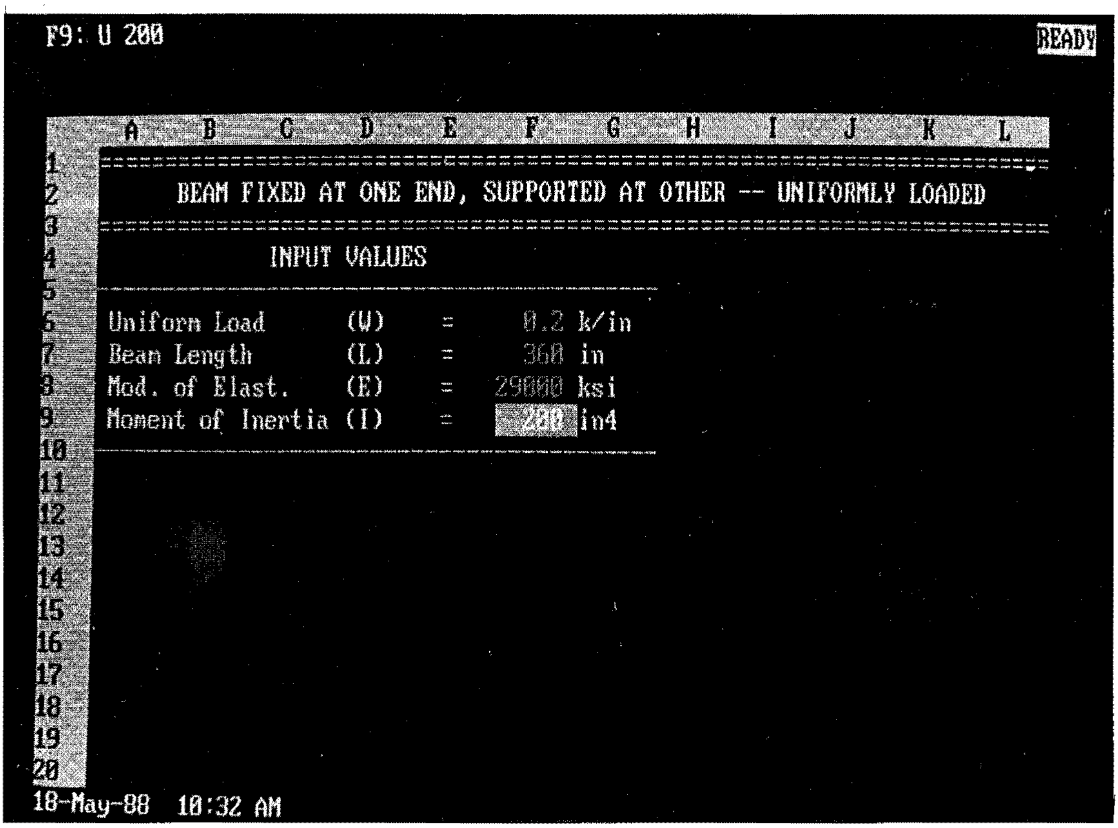


FIGURE 3. Input area with data entered.

cells. Each cell may contain text, a number or a formula, and is most commonly identified by a column letter and a row numeral (*e.g.*, A1 represents the uppermost left cell of the spreadsheet, C4 represents the cell at the intersection of the third column and fourth row). Usually, only a portion of the available spreadsheet is visible on the microcomputer's screen at one time — the size of the viewing area is regulated by the type of monitor and video display adapter that the personal computer is equipped with. However, the user can view other parts of the spreadsheet by using cursor control keys or a mouse. The maximum spreadsheet size varies for different software vendors, but it is typically of the order of 8,000 rows by 250 columns (2,000,000 cells).

To solve an engineering problem, the spreadsheet must be designed to accommodate input data, governing equations and the displayed results. One of the most useful features of a spreadsheet is that after setting up the governing equations, any or all of the input data may be changed and the new results will automatically be recalculated. This feature lets the user quickly iterate on design alternatives. Recalculation is also performed extremely fast; it is virtually instantaneous for normal-size spreadsheets that have less than a few hundred equations.

Example of Spreadsheet Setup. In order to illustrate the setup and use of a spreadsheet, values for the shear force, bending moment and deflection will be calculated for a common loading condition for an indeterminate beam that is fixed at one end and pinned at the other. The problem is illustrated in Figure 1 along with the governing equations.[8] The required input data are:

- uniform load
- beam length
- modulus of elasticity
- moment of inertia

The desired result is to produce a graphical rep-

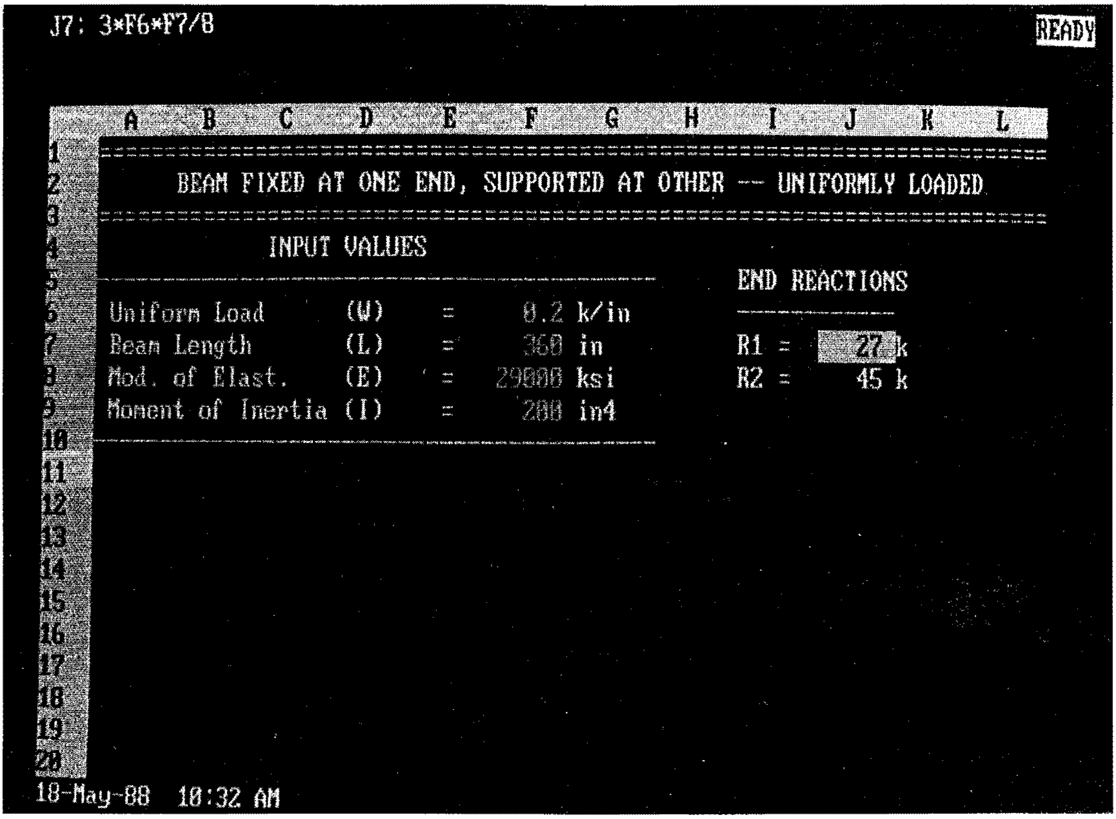


FIGURE 4. Formula entry in terms of cell address.

resentation of the shear, bending moment and deflection for a beam of this type.

Spreadsheet Input Area. The area on the PC's screen for data input is created in a portion of the spreadsheet as shown in Figure 2. Text is entered by locating the cursor at a cell and typing in the desired characters. At times, some of the text will extend into the next column. If one of these overlapped cells was subsequently filled by an entry at that location, it would simply hide from view the extended portion of the text.

Setting up an input section in a spreadsheet is much like using any common PC word-processing program. Unlike setting up an input area using a conventional programming language, a spreadsheet lets the user see the formatting of the text, data input areas, equations and results display areas on the screen as it will appear when the spreadsheet is completed. Note that the cursor is positioned at the first data entry at cell F6. Input values are entered by locating the cursor at the appropriate cell (cells F6 to F9 in this case) and typing in the numerical value. Figure 3 shows the spreadsheet with a set of input data entered. The format of the data in a cell may be set to any desired number of decimal places. Symbols such as hyphens and equal signs can be entered across a row of cells to isolate and highlight portions of the spreadsheet as was done in this example.

Equation Entry. The next step is to enter the appropriate equations that serve to manipulate the input data in order to produce the desired results. The equations most often utilize standard arithmetic operations, but can take advantage of any functions from the spreadsheet's library of mathematical functions such as sine, square root, *etc.* These functions are similar to those provided by any common programming language. Descriptive text can also be entered in nearby cells to clearly identify the results and/or the procedure used. In the example here, the equations for the vertical reactions have been entered into cells J7 and J8

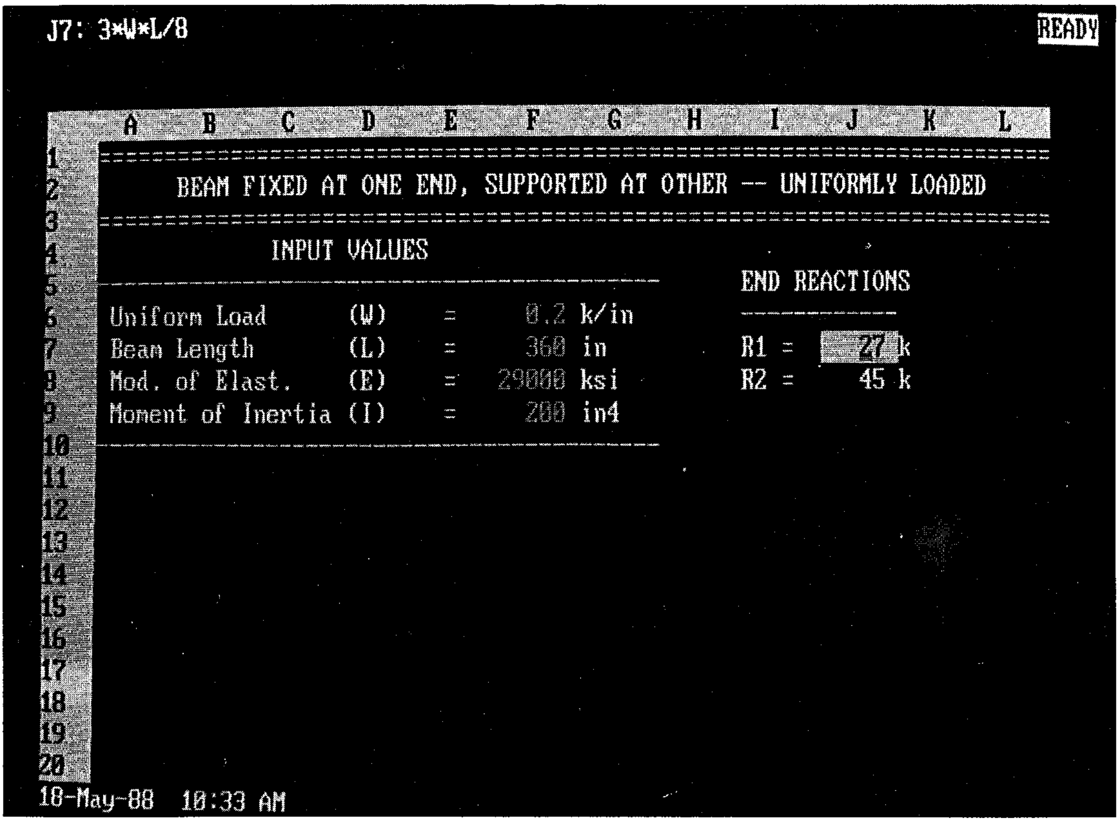


FIGURE 5. Formula entry using range naming.

as well as identifying text and unit labels in adjoining cells as shown in Figure 4.

The equations entered can refer to the cell addresses in manipulating the data they contain as shown at the top of Figure 4. Alternately, a technique of interest to engineers accustomed to dealing with variable names may be used. This technique, called range naming, involves assigning a name to a cell or group of cells similar to a variable name in other programming languages. This name may then be used in place of the cell address in the equation. For example, by naming cells F6 and F7, w and L, respectively, the equation in cell J7 for the left reaction takes on the more meaningful form as shown at the top of Figure 5. This technique is highly recommended since it eliminates the need to keep track of cell addresses and it is much easier to debug equations in this form.

The remainder of the equations are set up in a table as shown in Figure 6, with the shear, moment and deflection calculated at 1/10 span intervals. At first glance, it may seem like a great deal of work to enter the 33 equations to produce the results shown in the table, but most spreadsheet software provide sophisticated means of copying cells, either exactly or by incrementing certain cell addresses as needed. For example, in this case, all of the equations at the different intervals are identical except for the variation in the distance x. The following steps represent the method used to copy the deflection equations:

1. Enter the "basic" equation as shown at the top of Figure 6 for the deflection in cell B19.

2. Specify that the cell address of variable x is the only one in the equation that is to be incremented as the equation is copied to the next cell.

3. Copy the basic equation to cells C19 to L19.

This copying sequence is carried out using only a few keystrokes. Note that range names have

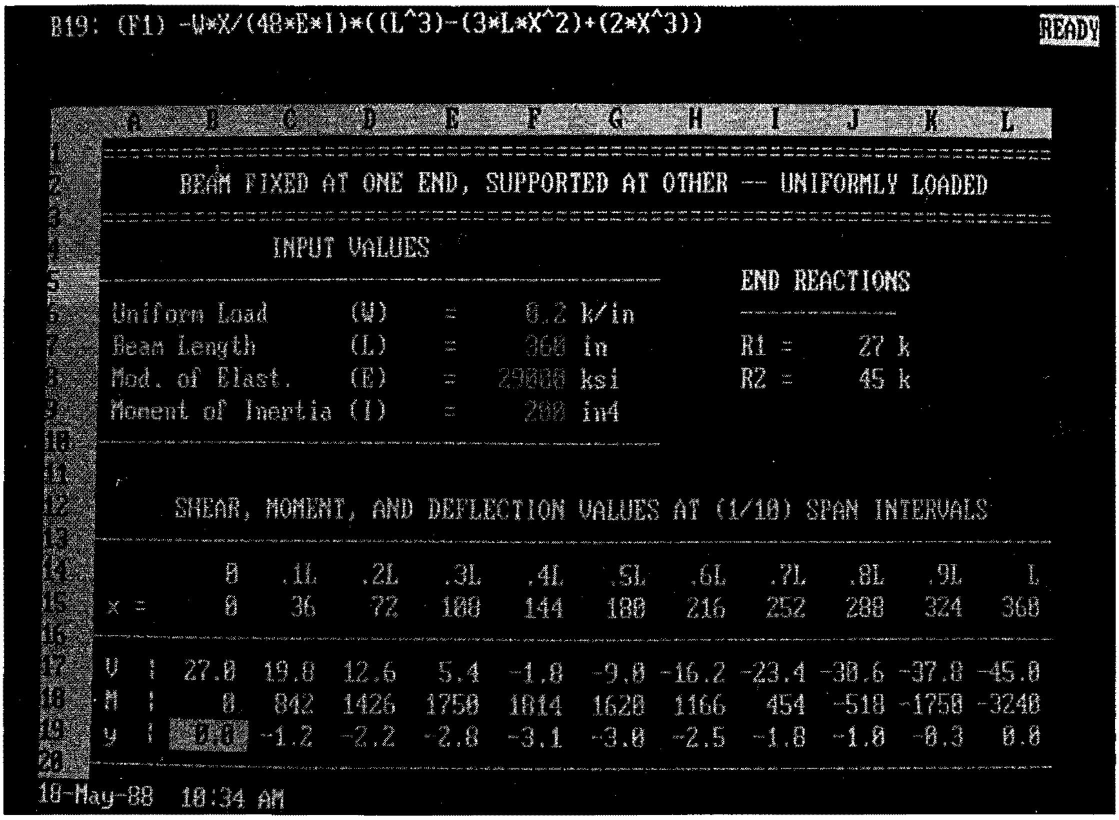


FIGURE 6. Finished spreadsheet.

been given to all of the input variables as specified in Figure 7.

Spreadsheet Graphics. A plot of the deflection values (see Figure 8) is easily obtained using the integrated graphics found in most spreadsheets. Depending on the spreadsheet used, the software's command menu will essentially prompt the user for the information required after selecting the graph option (see Table 1). The plot is then displayed as shown in Figure 8. Embellishments such as axis names can be easily added by going through more graph command sequences in a similar procedure. Once the plotting parameters have been defined, it is not necessary to go through the graph creation steps again. In most cases, the user just pushes a key to view the deflected shape after making any changes to the input data. In a similar manner, the bending moment and shear diagrams can be "programmed" into the spreadsheet.

Once the spreadsheet is set up as in this example, it may be saved and reused on any similar problem. It is then an easy task to adjust

Table 1

Typical Command Menu Options for Spreadsheet Graphics

Screen Prompt/Action	Response	Comments
Select Graph Type	Line	Choice of line, bar, X-Y or pie graph
Set X-Axis Range	X_Values	Select previously defined range (Fig. 7)
Set First Data Range	Y_Values	Select previously defined range (Fig. 7)
View Graph	View	Deflection diagram appears

```
B11:                                                                        READY

     A     B     C     D     E     F     G     H     I     J     K     L
1   ========================================================================
2         BEAM FIXED AT ONE END, SUPPORTED AT OTHER -- UNIFORMLY LOADED
3   ========================================================================
4                INPUT VALUES
5   ------------------------------------------      END REACTIONS
6    Uniform Load        (W)    =     0.2 k/in      ------------
7    Beam Length         (L)    =     360 in        R1 =        27 k
8    Mod. of Elast.      (E)    =   29000 ksi       R2 =        45 k
9    Moment of Inertia   (I)    =     200 in4
10  ------------------------------------------
11
12        SHEAR, MOMENT, AND DEFLECTION VALUES AT (1/10) SPAN INTERVALS
13  ------------------------------------------------------------------------
14                0    .1L    .2L    .3L    .4L    .5L    .6L    .7L    .8L    .9L      L
15   x =          0     36     72    108    144    180    216    252    288    324    360
16  ------------------------------------------------------------------------
17   V   |     27.0   19.8   12.6    5.4   -1.8   -9.0  -16.2  -23.4  -30.6  -37.8  -45.0
18   M   |        0    842   1426   1750   1814   1620   1166    454   -518  -1750  -3240
19   y   |      0.0   -1.2   -2.2   -2.8   -3.1   -3.0   -2.5   -1.8   -1.0   -0.3    0.0
20
18-May-88   04:08 PM
```

W
L
E
I

Y_Values
M_Values
V_Values
X_Values

FIGURE 7. The range names used in the spreadsheet example (equivalent to variable names).

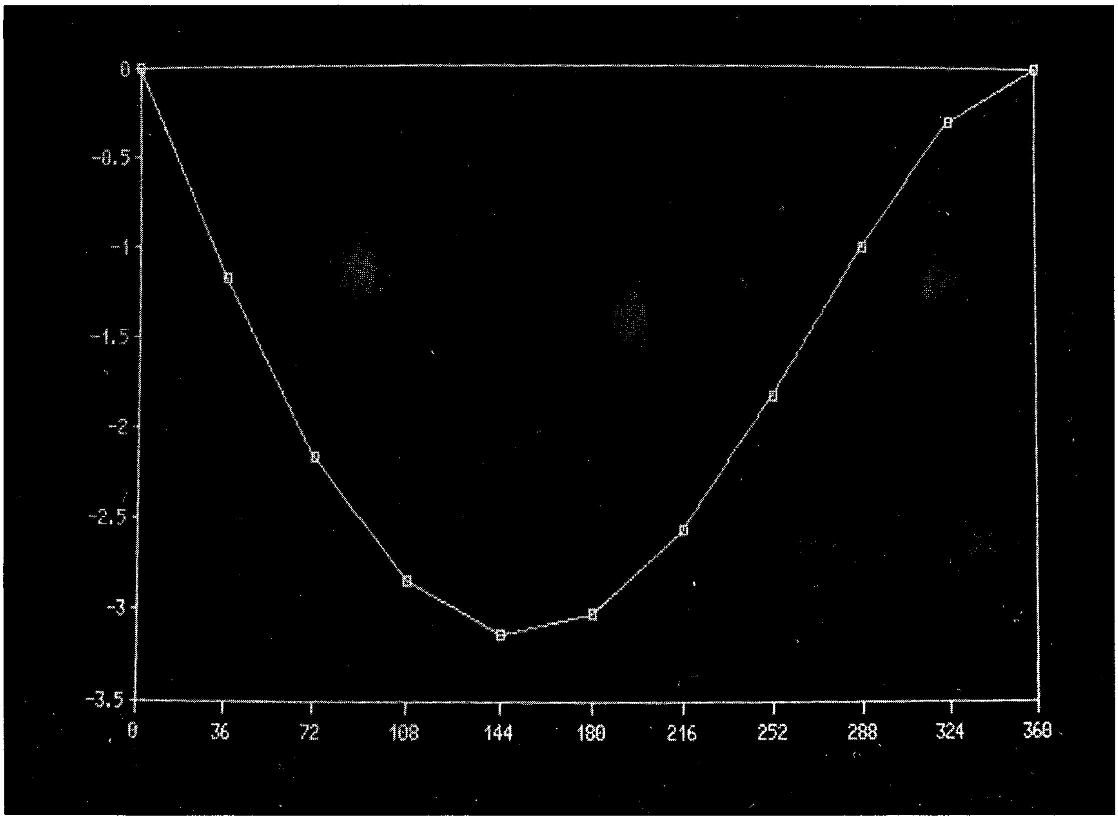


FIGURE 8. Deflected plot as displayed using spreadsheet graphics capabilities.

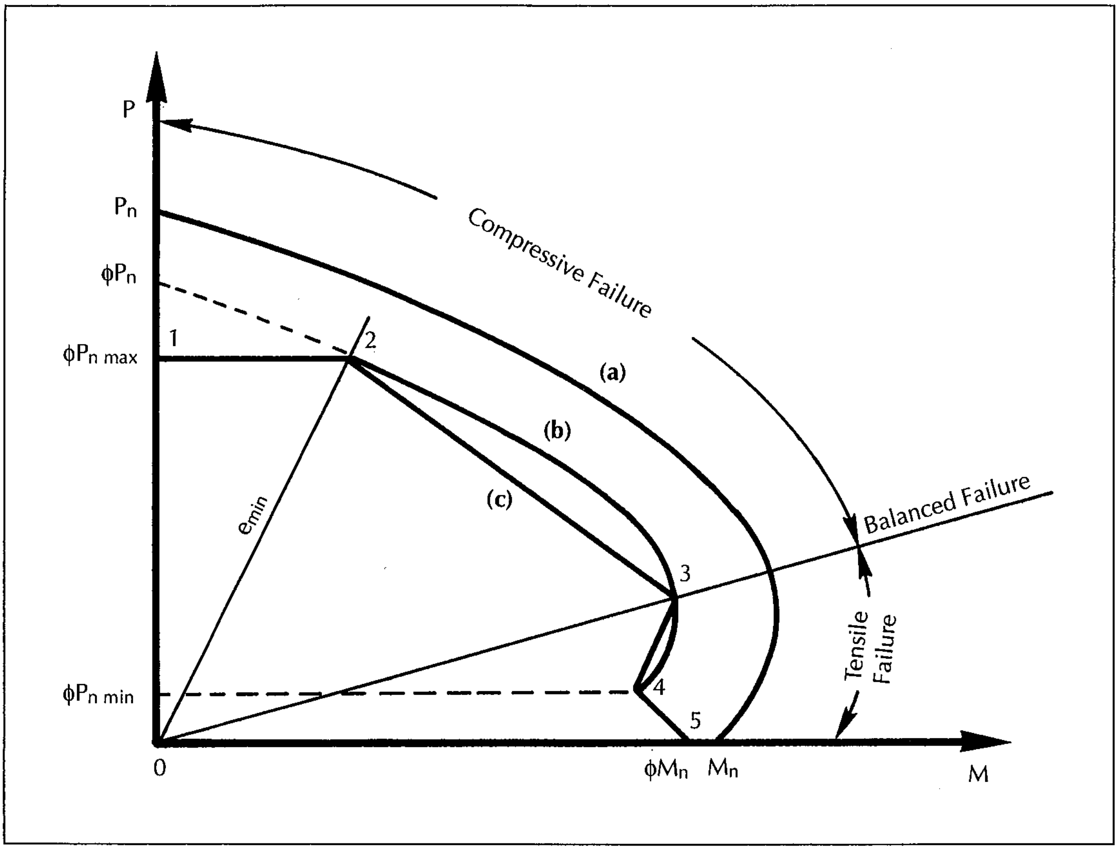


FIGURE 9. Axial/bending interaction diagram.

any input quantity and see the effect on the results, either numerically or graphically, almost instantaneously. This simple example illustrates some of the advantages that spreadsheets have over conventional programming languages. One of the most time-consuming programming tasks is dealing with input and output. Writing additional program source code to provide graphical display of results greatly increases the time needed to develop the software.

This example shows that spreadsheet "programming" is similar to working out a problem with a handheld calculator and a piece of paper. There is no need to learn a specific programming language — spreadsheet "programming" is menu- or command-driven. This makes spreadsheet solutions amenable to some unique problems that may not warrant the time to program conventionally. The more complex design example in the next section uses these same techniques to make the interactive graphics a key design aid as opposed to simply a "nice effect."

Design Example

Problem Statement. The design of reinforced concrete beams or columns for combined axial loads and bending moments can present a formidable task with regard to the number of calculations and iterations required to converge on an economical solution. The assumptions and recommendations presented by the American Concrete Institute (ACI) reduce the analysis of a trial section to a pure statics problem.[9] However, depending on the relative magnitude of the axial load and bending moment, the column may have a primary failure mode caused by compression in the concrete or it may fail due to a yielding of the reinforcing steel. Solutions to this type of problem are presented in virtually all reinforced concrete design textbooks (see, for example, Wang and Salmon[10]).

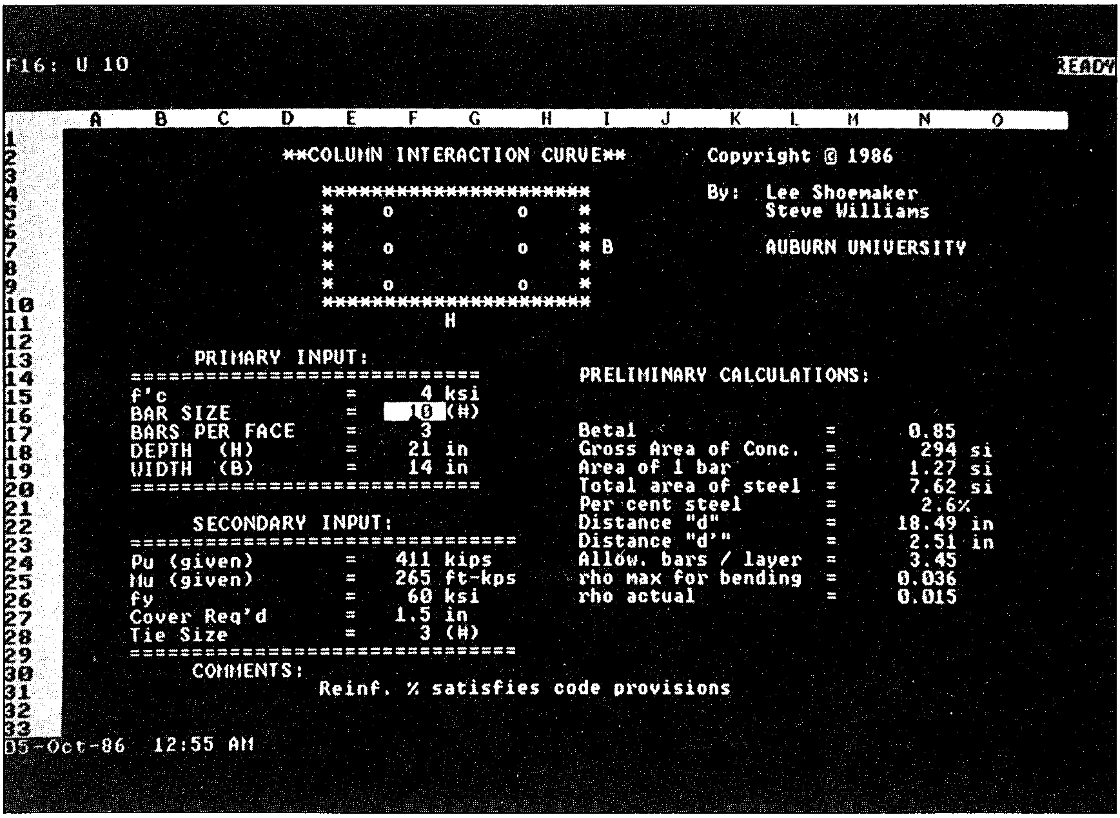


FIGURE 10. Spreadsheet "home" screen.

The manual computations required to produce an interaction diagram for a given cross-section are very tedious. Non-dimensionalized diagrams are available as design aids,[11] as well as tabular design aids.[12] While these design aids do serve to reduce the work (as compared with a totally manual analysis), they still have some major limitations. Most require some preliminary calculations prior to using a non-dimensionalized chart, and few give the user an opportunity to view an interaction diagram for a specific section. Computer programs are available for the design of reinforced concrete columns,[13] and for determining key points required to plot an interaction diagram. However, the use of a spreadsheet with built-in graphics capabilities can allow a designer to easily vary any parameter and instantly see the interaction diagram.

The interaction diagram is shown in Figure 9. Curve *(a)* represents the combinations of axial load, *P*, and bending moment, *M*, that would cause ultimate failure in the column. Curve *(b)* represents the design interaction curve that incorporates a strength reduction factor φ, a specified minimum eccentricity that flattens out the curve from points 1 to 2, and a linear transition from a specified minimum axial load to pure bending that takes into account a transition in strength reduction factors from points 4 to 5. Curve *(c)* is a modified design curve taking into account the five points shown. This modified curve would be conservative, but yet not overly so. The modified design curve is the one produced by the spreadsheet application described.

Spreadsheet Operations. As noted in Figure 10, the "home" screen of the spreadsheet includes areas for primary and secondary input. Primary input includes five design parameters that vary frequently. The secondary input includes five other design parameters that may be changed, but usually remain constant and are thus set to default values. Also shown in this figure is an area for preliminary calculations. These calculations are performed automatical-

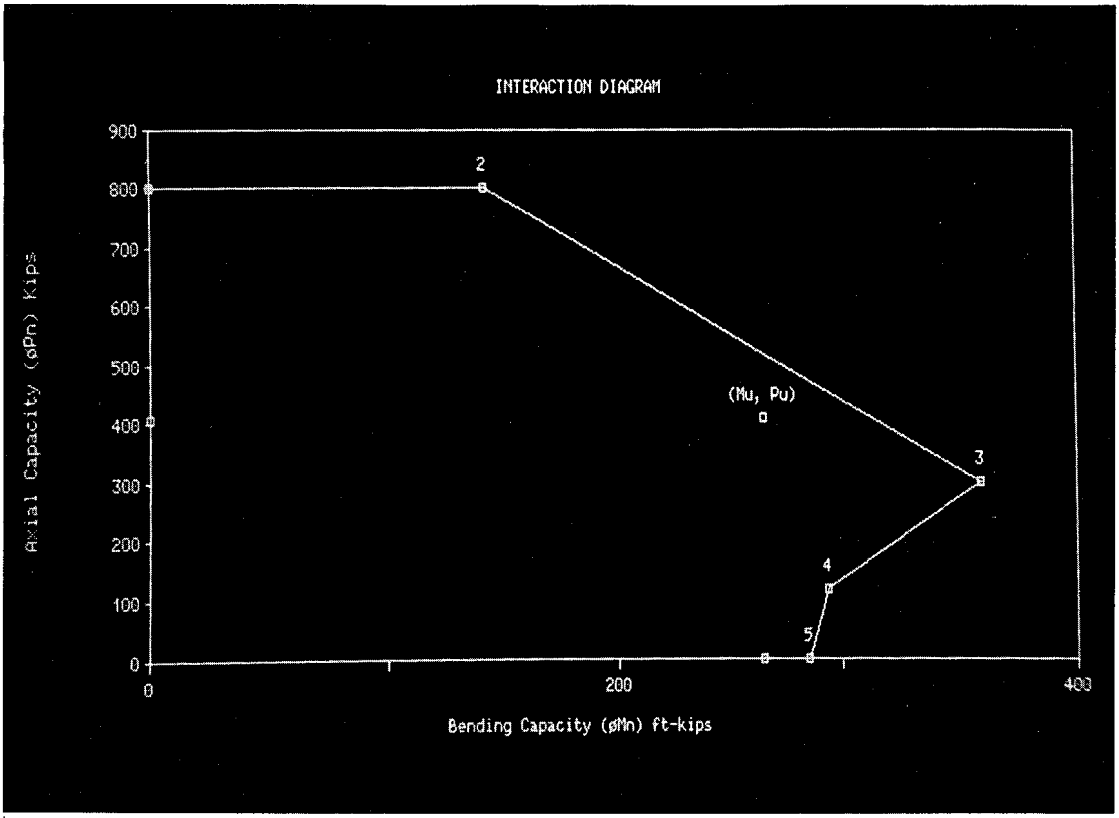


FIGURE 11. Interaction diagram for the data shown in the "home" screen in Figure 10.

ly by the spreadsheet and are based on the current input values. All input data are automatically checked to insure that the appropriate ACI Code provisions are satisfied. Messages regarding conformance or non-conformance to these code requirements will cause a "flag" to appear in the comments section of the screen as shown in Figure 10.

After the designer has entered the desired input, an interaction diagram for this particular column can be displayed. For example, Figure 11 shows the screen display of the interaction diagram produced from the input data of Figure 10.

In addition to the column material and cross-sectional data, the designer may wish to enter the factored axial load, P_u, and bending moment, M_u, that are acting on the column as a secondary input item. This applied loading combination is plotted on the interaction diagram as shown in Figure 11. If this point is within the boundary of the interaction diagram, the chosen design parameters represent an acceptable solution. Generally, the closer this point is to the boundary, the more efficient is the solution.

After viewing an interaction diagram, the designer returns to the original input screen. If any input changes are necessary, the cursor is moved to the appropriate cell and new data is entered. The spreadsheet instantaneously recalculates all values anytime a change is made in either the primary or secondary input. The subsequent interaction diagram may again be viewed. This process may be repeated many times within a short period of time in order to determine the most desirable solution.

A situation commonly encountered in reinforced concrete design is the adjustment of the area of steel within a given column size to provide a required capacity. A change in column size could require a reanalysis of the building frame since reinforced concrete structures are generally monolithic and statically indeterminate. It is also economically unsound to vary column size to suit the load on each floor

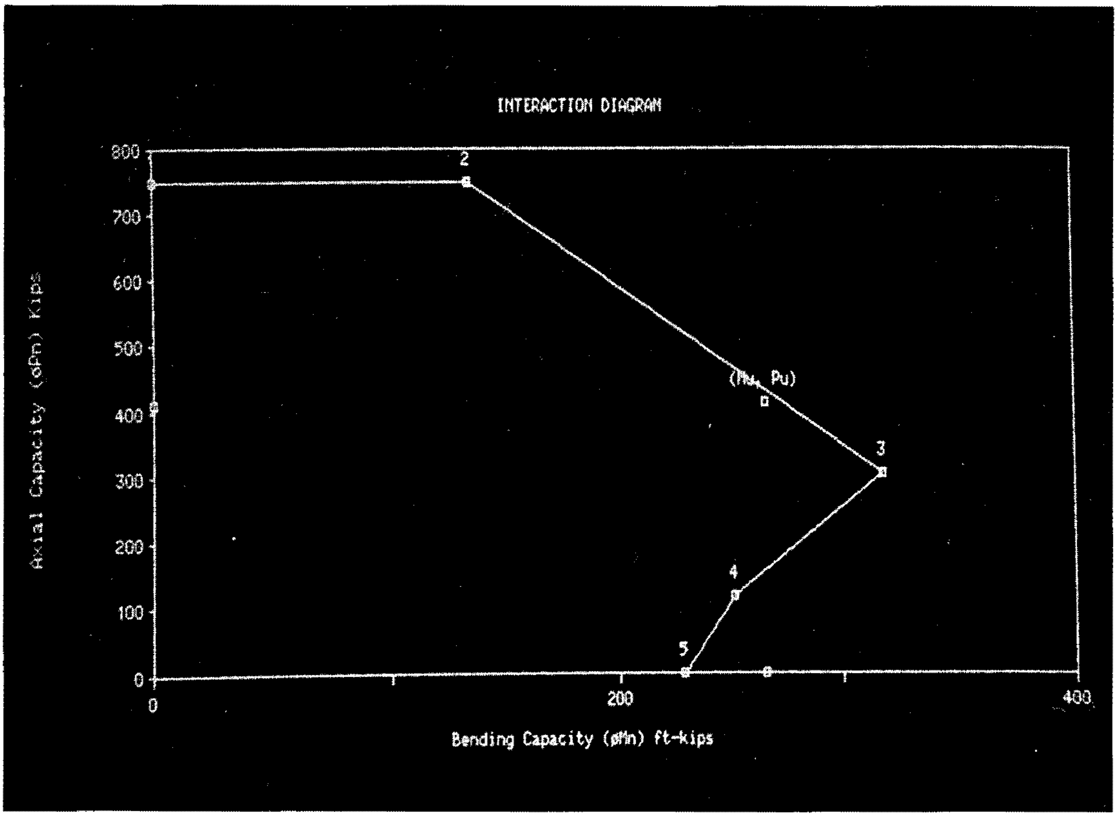


FIGURE 12. Interaction diagram after changing the reinforcing bar size from No. 10 to No. 9.

level. Greater economies are achieved by maintaining the same column size for the entire building height and varying the reinforcement to correspond to lighter loads at higher levels.[14] As an example of varying this design parameter, Figure 12 was generated by changing the size of the reinforcing bars from No. 10 (diameter equal to 1.27 in.) to No. 9 (diameter equal to 1.128 in.) on the original input screen shown in Figure 10.

The software used for this example generated the new interaction diagram in under 10 seconds. A designer using hand computations, even with the aid of design table or diagrams, will usually be satisfied when a trial section is within allowable limits and will be less likely to check for more economical alternatives. This example gives an indication of how this spreadsheet approach would enable a designer to refine a design very quickly and produce a more economical solution. The designer would also know exactly where the applied loading combination fell on the interaction diagram, providing more confidence in the design.

Conclusions

The spreadsheet application example discussed herein was not developed with the intention of competing with existing computer programs or design aids, but rather to illustrate the tremendous potential these programs have with regard to engineering applications.

Spreadsheets offer four distinct advantages over conventional programming languages, programming environments and specific off-the-shelf engineering applications software.

- They can be developed without requiring the knowledge of higher level programming languages and programming skills.
- They offer superior input/output capabilities.
- They can provide "instant" graphical solutions.

- They are specifically designed for the "What if?" effects required for sensitivity analysis.

In addition to these primary benefits of using commercially available spreadsheet software for PCs, there are other adjunct advantages such as:

- the ability to import and export data to and from other database or word-processing applications software;
- built-in database functions that are available with some commercial spreadsheet software;
- options to protect data, input areas and formulas from accidental change or intentional tampering; and,
- the ability to create a "template" spreadsheet that can be used for different applications and by many users.

Engineers need to give serious consideration to the many features that electronic spreadsheets have to offer. Because of the potentially large increase in productivity, engineers who do make use of this new technology will have time to investigate more variations, thus creating better, more cost effective designs.

NOTE — *The spreadsheet software used for this article was Lotus 1-2-3 Version 2.0 run on an IBM PC. Menu and command sequences, as well as the specific options and functions, provided by specific commercial spreadsheet software for microcomputers varies widely and cannot be generalized. Most commercial PC software do fulfill the functions/requirements as set forth in this article.*

W. LEE SHOEMAKER *is an Assistant Professor in the Civil Engineering Department at Auburn University. He is a registered PE and a member of ASCE and the American Concrete Institute's Committee 224 on the Use of Computers.*

STEVE WILLIAMS *is an Associate Professor in the Building Science Department at Auburn University. He is a member of ASCE and a registered PE, and he has conducted seminars and workshops on microcomputer usage and spreadsheet applications.*

REFERENCES

1. Trost, S., and Pomernacki, C., *VisiCalc for Science and Engineering*, Sybex, Inc., Berkeley, CA, 1983.

2. Kleiner, D., "Engineering With Spreadsheets," *Civil Engineering*, October 1985, p. 55.

3. Casas, A., "Spreadsheets: A New Design Tool," *Civil Engineering*, December 1986, p. 66.

4. Casas, A., and Oppenheim, I., "Spreadsheet Programming for Structural Design," *Computer Applications in Concrete Technology, ACI SP-98*, American Concrete Institute, Detroit, MI, 1987.

5. Shoemaker, W.L., and Williams, S., "Prestressed Concrete Design Using Spreadsheets," *Prestressed Concrete Institute Journal*, Vol. 33, March/April 1988, p. 110.

6. Stiemer, S., "Microcomputers in Teaching: Steel Design With Spreadsheets," *Microcomputers in Civil Engineering*, Vol. 1, October 1986, p. 165.

7. Wenzel, T., "Use of Spreadsheet Programs in Teaching Reinforced Concrete Design," *Computer Applications in Concrete Technology, ACI SP-98*, American Concrete Institute, Detroit, MI, 1987.

8. American Institute of Steel Construction, *Manual of Steel Construction*, Eighth Edition, Chicago, IL, 1980.

9. American Concrete Institute, *Building Code Requirements for Reinforced Concrete (ACI 318-83)*, ACI, Detroit, MI, 1983.

10. Wang, C., and Salmon, C., *Reinforced Concrete Design*, Fourth Edition, Harper and Row, New York, 1985.

11. American Concrete Institute, *Design Handbook, Vol. 2, Columns (ACI SP-17a)*, ACI, Detroit, MI, 1984.

12. Concrete Reinforcing Steel Institute, *CRSI Handbook*, CRSI, Schaumburg, IL, 1984.

13. Portland Cement Association, *Strength Design of Reinforced Concrete Column Sections*, PCA, Skokie, IL, 1974.

14. Cole, L., *Simplified Design of Columns: Simplified Design of Reinforced Concrete Buildings of Moderate Size and Height*, Portland Cement Association, Skokie, IL, 1984.