Having Fun with Concrete Structural Design Utilizing TK Solver Software

Edwin P. Foster[1]

Abstract – Designing a concrete beam is a juggle with many variables: width, depth, area of steel, strength of concrete, and strength of steel. None of these are specified in a "real design" problem. In class, some of these variables are specified and the student solves for only the one or two remaining variables. If the student solves for only one variable then it is an analysis problem and not a design problem. Assigning true design problems present a grading nightmare for faculty since no two designs in the class should be the same. My first concrete program written using TK Solver [Universal Technical Systems, Inc, 2] only took twenty minutes after the software was loaded onto my computer. I had never used the software before. TK lets you pick the input variables and it then solves for the output variables. An engineer who has never used TK Solver easily understands the printout of the variables and equations. Complex concrete column design problems are easily solved. Exact solutions for b, d, A_s, A'_s for a column with given f_c', f_y, load, moment and percent reinforcement can be obtained. This solution requires iterating on three variables until an exact solution is obtained. Spreadsheets cannot do that. TK can easily draw the column interaction diagram as well. With software like this, the grading of "real design" problems can be checked with just a few clicks on the mouse and a few keystrokes. TK can easily solve simultaneous nonlinear equations of any order. Input and output variables can be swapped without any change in programming. At first students objected to checking their hand solutions using the TK model. Then when the design problems became more complex and the TK model uncovered errors in their calculations, the students defined it as an extremely useful tool.

Keywords: TK Solver, concrete design, structural analysis, structural design

INTRODUCTION

In the thirty-six years in which I have been teaching how to design concrete structures the tools have changed drastically. The computer made design easier, however a good program took a great deal of time to develop and had rigid input and output. The design problems I assigned were not real design problems in that I assigned the value for a number of design variables so I could grade the homework in a decent amount of time. Things have changed since I wrote a few concrete programs using TK Solver. We now have "design contest" like: "Design the lightest weight concrete bridge beam with a length of fifty feet that will permit a moving two kip load to move from one end to the other".

I will describe the TK Solver software first and then demonstrate how the flexibility of this software can be utilized in the analysis and design of concrete structures.

TK SOLVER

TK-Solver is a general-purpose equation solver. You do not need to enter the equations in any particular order. You do not need to isolate one variable on the left of the equal sign. You can add more equations to the TK "model" at any time. Therefore you can have the model calculate the value of more variables by just adding more equations. There is no need to completely rework or check the old program to determine if the new equations will work by adding them to the end of the program.

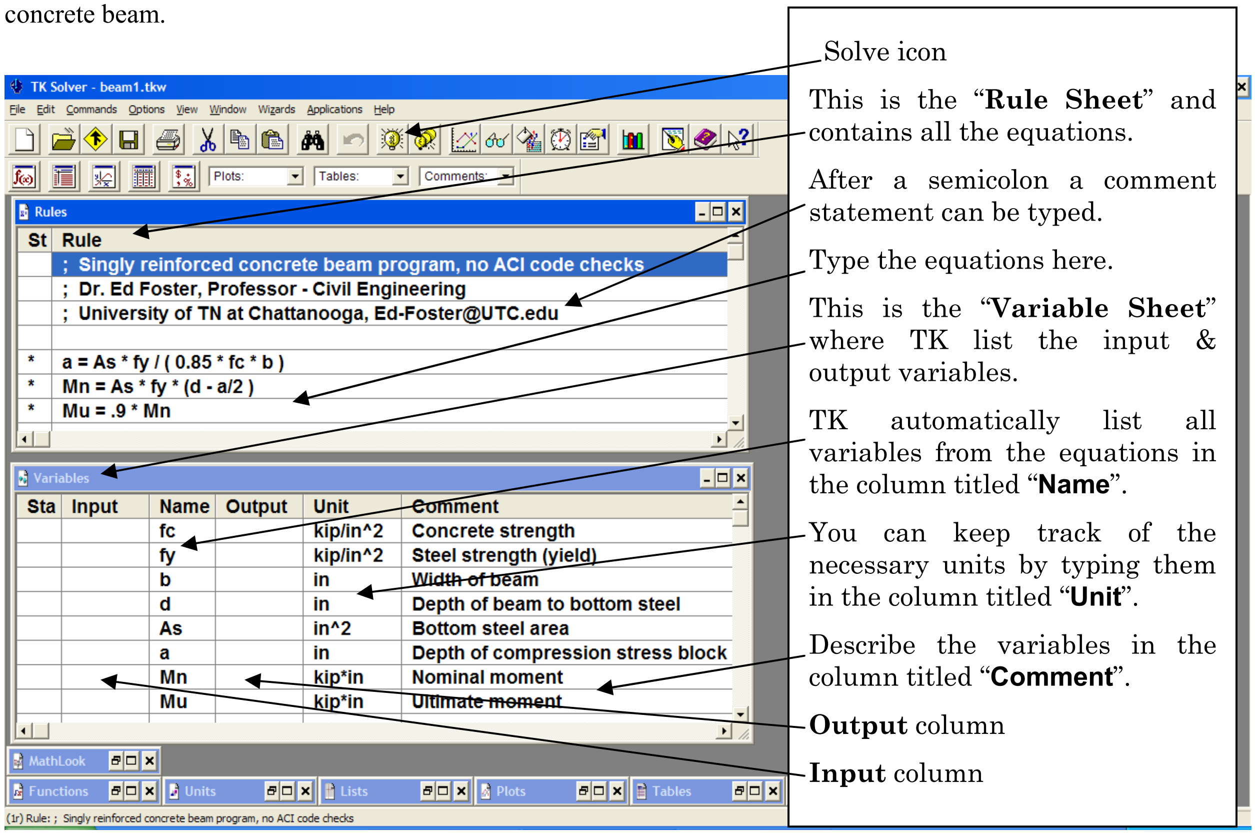
TK works by taking the input variables and substituting them into each of the equations in the program. When TK discovers an equation with only one unknown then the equation is solved for that unknown and from that point TK

[1] Professor, Civil Engineering Program, University of Tennessee at Chattanooga, www.utc.edu/Academic/Engineering/, E-mail: ed-foster@utc.edu

has another input variable for the following equations. When TK gets to the last equation it has accumulated the value of more variables in addition to the initial input variables. TK then starts at the first equation again and goes down the list of equations searching for "one equation" with "one unknown". TK keeps looping through the list of equations until it can no longer solve for anything else or until it finds two equations that are inconsistent.

CONCRETE BEAM PROGRAM (MODEL)

Now we will look at a very simple "TK model" that solves for the moment carrying capacity of a reinforced concrete beam.



Enter the value of the variables that are known in the "Input" column to the left of the variable name. Then click on the solve icon and the value of the "output" variables will appear in the "Output" column to the right of the variable. Since we have three equations we can have three unknown variables and therefore must have five known variables. The following table shows typical input and output scenarios.

Following that we will expand the Rule Sheet to include the ACI code calculations and checks. The Variable Sheet shows a typical set of input variables, which completely define the concrete beam shape. The output variables are code values and the moment carrying capacity of the concrete beam.

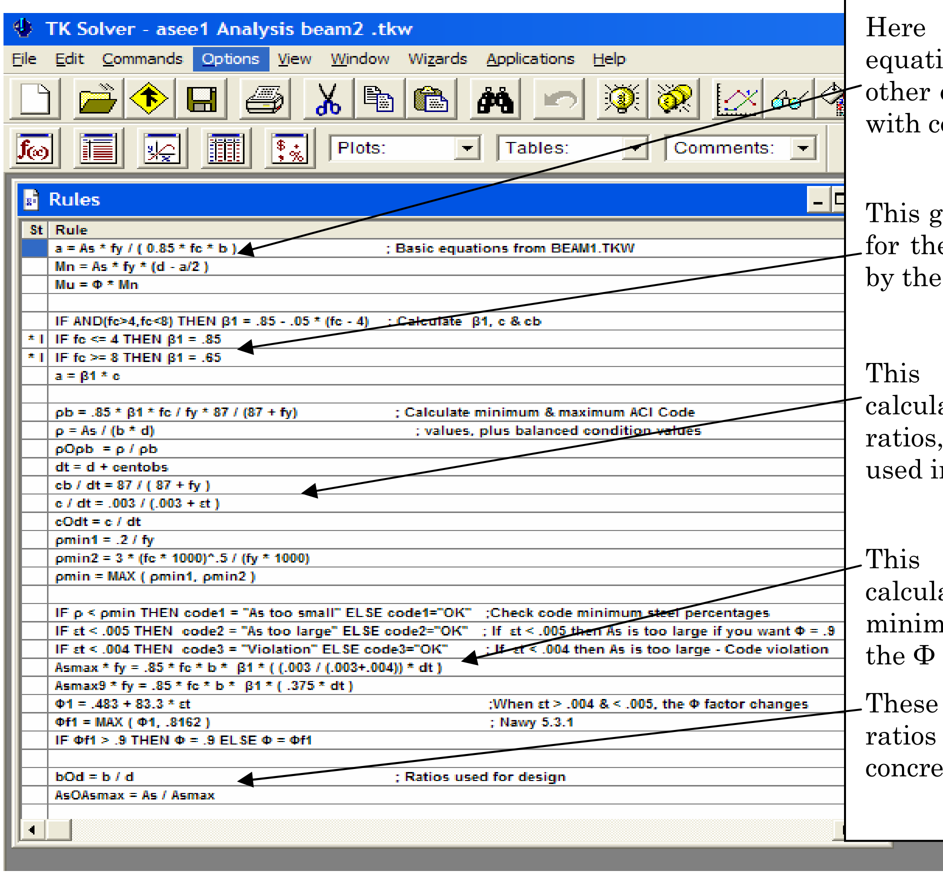
Table-1

Input Variables

1	b	d	As	fc	fy	
2	b	d		fc	fy	Mu
3		d	As	fc	fy	Mu
4	b		As	fc	fy	Mu

Output Variables

1	Mu	Mn	a
2	As	Mn	a
3	b	Mn	a
4	d	Mn	a



Rule Sheet

Here are the same three equations as before. All the other equations are concerned with code checks.

This group of equations solves for the variable β1 as defined by the ACI code.

This group of equations calculates balanced variables, ratios, and steel percentages used in the ACI code.

This group of equations calculates maximum and minimum area of steel and the Φ factor of safety.

These two equations define ratios used for design of concrete beams.

A typical concrete beam "analysis" is shown in the Variable Sheet. This is case-1 in Table-2. If we are concerned with designing the concrete beam then one or more of the typical design variables would need to be an output variable. Case 2 in Table-2 has Mu as an input variable and eliminates the input value of As, therefore As becomes an output variable. In case-1 and case-2 we can solve for only one design variable since we have only one equation that relates those variables. If we want to solve for three design variables then we would need to add two more equations that relates those design variables. Two equations were therefore added to the Rule Sheet above. They relate the b/d ratio and the As/Asmax ratio. Those two ratios are set equal to the variable names bOd and

AsOAsmax respectively. Case 3 in Table-2 now shows that if we input the value of the bOd and AsOAsmax ratios we were able to solve for all three beam design variables.

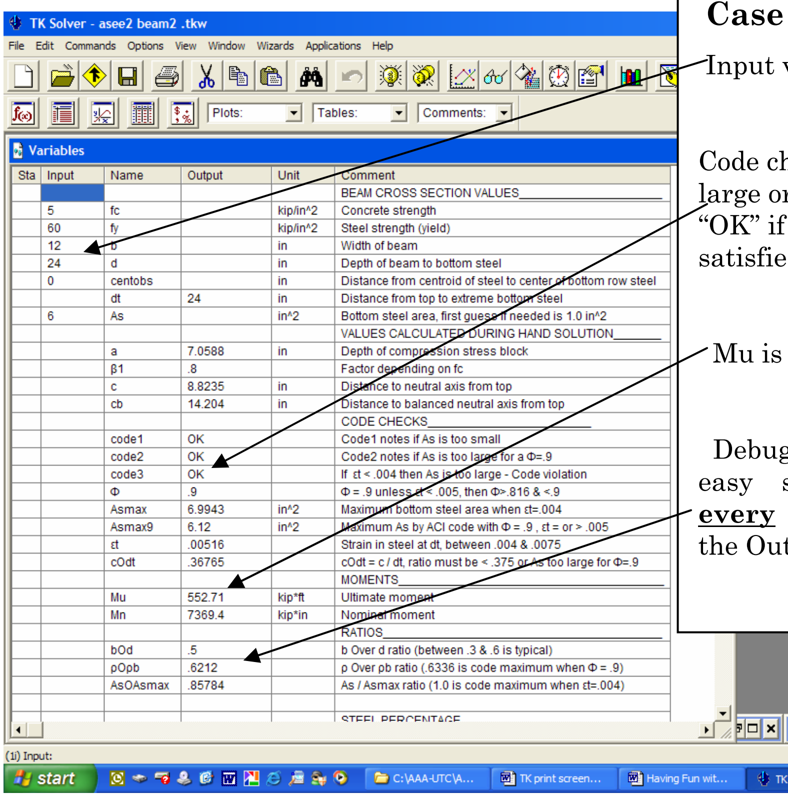
Table-2

Input Variables

1	b	d	As	fc	fy			
2	b	d		fc	fy	Mu		
3				fc	fy	Mu	bOd	AsOAsmax

Output Design Variables

1	Mu		
2	As		
3	b	d	As



Variable Sheet

Case 1 from Table 2

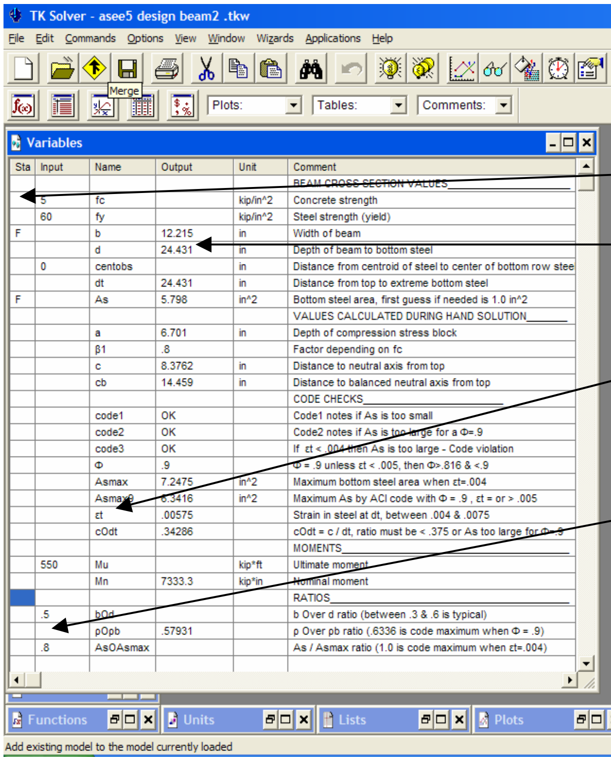
Input variable values

Code checks see if As is too large or too small. Outputs "OK" if code checks are satisfied.

Mu is an output variable

Debugging a TK program is easy since the value of **every** variable is listed in the Output column.

The following Variable Sheet shows the input and output for Case 3 from Table-2. Also note that there is an "F" in the far left "Status" column next to the two variables b and As. The "F" designates that a first guess value was assigned to get the Newton-Raphson nonlinear solution procedure started.



Sta	Input	Name	Output	Unit	Comment
					BEAM CROSS SECTION VALUES__________
	5	fc		kip/in^2	Concrete strength
	60	fy		kip/in^2	Steel strength (yield)
F		b	12.215	in	Width of beam
		d	24.431	in	Depth of beam to bottom steel
	0	centobs		in	Distance from centroid of steel to center of bottom row stee
		dt	24.431	in	Distance from top to extreme bottom steel
F		As	5.798	in^2	Bottom steel area, first guess if needed is 1.0 in^2
					VALUES CALCULATED DURING HAND SOLUTION_______
		a	6.701	in	Depth of compression stress block
		β1	.8		Factor depending on fc
		c	8.3762	in	Distance to neutral axis from top
		cb	14.459	in	Distance to balanced neutral axis from top
					CODE CHECKS__________
		code1	OK		Code1 notes if As is too small
		code2	OK		Code2 notes if As is too large for a Φ=.9
		code3	OK		If εt < .004 then As is too large - Code violation
		Φ	.9		Φ = .9 unless εt < .005, then Φ>.816 & <.9
		Asmax	7.2475	in^2	Maximum bottom steel area when εt=.004
		Asmax9	6.3416	in^2	Maximum As by ACI code with Φ = .9 , εt = or > .005
		εt	.00575		Strain in steel at dt, between .004 & .0075
		cOdt	.34286		cOdt = c / dt, ratio must be < .375 or As too large for Φ=.9
					MOMENTS__________
	550	Mu		kip*ft	Ultimate moment
		Mn	7333.3	kip*in	Nominal moment
					RATIOS__________
	.5	bOd			b Over d ratio (between .3 & .6 is typical)
		pOpb	.57931		p Over pb ratio (.6336 is code maximum when Φ = .9)
	.8	AsOAsmax			As / Asmax ratio (1.0 is code maximum when εt=.004)

Variable Sheet

Case 3 from Table 2

Status column

Now there are three output design variables; b, d, & As.

Instead of inputting AsOAsmax we could have input the steel strain, εt.

Since the input variables b & d were erased we need to specify two other input variables. In this case bOd and AsOAsmax were used.

It is impressive to note that this unbelievably flexible analysis/design program only took twenty-seven statements. Let us now assume that we want to design a concrete beam for a simply supported and uniformly loaded span. Let us also consider a concentrated load at the center of the beam span and include the beam weight in the calculations. The Rule Sheet would contain just two more equations at the bottom of the sheet. Those equations are shown below. The maximum moment is at the center of the beam. If we give that moment the variable name Mu, which is the same name as the moment carrying capacity of the cross section, then the two must be equal to each other. We now have a classic iterative solution – the beam size is unknown and cannot be calculated without knowing the weight of the beam – but the weight of the beam depends on the beam size. TK uses a Newton-Raphson iteration for its Iteration Solver. All we need to do is give TK a "first guess" value to start the iteration. The variables b & As require a first guess. This can easily be done by typing a "G" in the far left column of the Variable Sheet marked "Status" and placing the first guess in the "input" column next to the variable or going to the variable subsheet and inputting a permanent first guess.

Rule
WTperL = b * (dt + 3) * WTperVOL
Mu = ((1.2 * (WTperL + DL) + 1.6 * LL) * L^2 / 8) + 1.6 * P * L/4

The Variable Sheet would now have these added variables included at the end of the sheet.

Status	Input	Name	Output	Unit	Comment
					LOADS ON BEAM ______________________
		WTperVOL		lb/ft^3	Weight of concrete per unit volume (normal = 150 lb/ft^3)
		WTperL		lb/ft	Weight of the concrete beam per length
		DL		lb/ft	Dead load other than beam weight
		LL		lb/ft	Live load on the beam
		P		kip	Concentrated live load at center of beam
		L		ft	Length of the beam

Table 3 shows the input and output variables for the design of a beam size when the beam weight is considered as a part of the load on the beam. This beam solution requires three simultaneous equations, two of which are nonlinear.

Table-3

Input Variables

fc	fy	bOd	εt	WTperVOL	DL	LL	P	L

Output Variables

Mu	WTperL	As	b	d

CONCRETE COLUMN PROGRAM (MODEL)

The next TK model presented is for a non-slender rectangular concrete column. Since this column can have three rows of steel it took seventy-seven rules (statements) to completely define the column equilibrium and code equations. In order to analyze a specific column (Example 9.11 [Nawy, 1]) the input variables needed are:

Status	Input	Name	Output	Unit	Comment
		__________		_________	BEAM_DESIGN_VARIABLES_________________
	1	ReInf			Reinforcement Type 1 =Tied, 2 = Spiral
	0	DC			Displaced concrete? Yes: DC=1, No: DC=0
	29000	Es		Ksi	Modulus of elasticity
	6	fc		Ksi	Strength of concrete
	60	fy		Ksi	Strength of steel
	12	b		in	Width of Column
	14	h		in	Depth of Column
	11	d		in	Dist. to bottom steel, d = h - dp1
	3.12	As		in.^2	Area of bottom steel
	3	dp1		in.	Dist. to top steel
	3.12	Asp1		in.^2	Area of top steel
	3	dp2		in.	Dist. to middle steel, use dp1 if Asp2=0
	0	Asp2		in.^2	Area of middle steel

The main output variables are then:

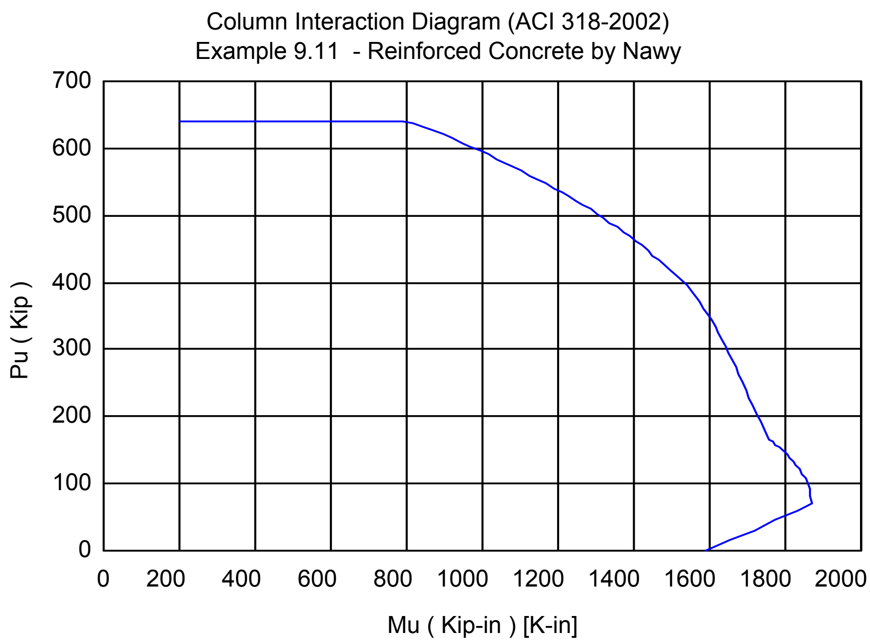
Status	Input	Name	Output	Unit	Comment
					P VALUES BASED ON MINIMUM OR NO e
		Po	1231.2	Kip	Nominal load with Mn = 0
		Puo	800.28	Kip	Ultimate load with Mu = 0
		Pnmax	984.96	Kip	Maximum nominal load with minimum e
		Pumax	640.22	Kip	Maximum ultimate load with minimum e
					BALANCED VARIABLES
		Cb	6.5102	in.	Depth of bal. neutral axis
		ab	4.88	in.	Depth of bal. stress block
		εb	.00206		Balanced strain in bottom steel
		Pnb	257.97	Kip	Nominal balanced strength

		Pub	167.68	Kip	Ultimate balanced strength
		Mnb	2696.4	K-in	Nominal balanced Moment
		Mub	1752.6	K-in	Ultimate balanced Moment
		eb	10.452	in.	Load eccentricity

If a number of C values are automatically generated and the corresponding values of PuPLOT and MuPLOT are used to generate a table then the column interaction diagram can be plotted from that table.

Status	Input	Name	Output	Unit	Comment
L	5	C		in.	Var. Depth of comp. neutral axis
L		PuPLOT	117.77	Kip	Ultimate column load - must be < Pumax
L		MuPLOT	1841.6	K-in	Ultimate column moment

The column interaction diagram for this column is:



The Variable Sheet listed next is the TK solution for a column design. The problem specifies that the column load = 358 kip, moment = 5728 kip-in, square shape, 3% symmetrical steel, fc = 4 ksi, fy = 60 ksi, d' = 3 in, and it is a tied column. This solution consists of solving for four unknowns; b, h, As, and As1. This solution also requires a starting value for three variables; b, As & C. Spreadsheets can only iterate on one variable without a great deal of extra programming effort.

Sta tus	Input	Name	Output	Unit	Comment
				____	BEAM_DESIGN_VARIABLES________________
	1	ReInf			Reinforcement Type 1 =Tied, 2 = Spiral
	0	DC			Displaced concrete? Yes: DC=1, No: DC=0
	29000	Es		Ksi	Modulus of elasticity
	4	fc		Ksi	Strength of concrete
	60	fy		Ksi	Strength of steel
F		b	20.013	in	Width of Column
		h	20.013	in	Depth of Column

		d	17.013	in	Dist. to bottom steel, d = h - dp1
F		As	6.0079	in.^2	Area of bottom steel
	3	dp1		in.	Dist. to top steel
		Asp1	6.0079	in.^2	Area of top steel
	3	dp2		in.	Dist. to middle steel, use dp1 if Asp2=0
	0	Asp2		in.^2	Area of middle steel
				____	RATIOS_FOR_DESIGN_____________________
	.03	ρ			Steel percentage, ρ=Ast / (b*h)
	1	bOh			Ratio of: (b / h)
		Cod	.53196		C/d, Between .375 & .6 for transition zone between T & C failure
	.5	RatioA1			Ratio of: (Asp1 / Ast)
		RatioA2	0		Ratio of: (Asp2 / Ast)
	.5	RatioAs			Ratio of: (As / Ast)
				____	P VALUES BASED ON MINIMUM OR NO e
		Po	2082.7	Kip	Nominal load with Mn = 0
		Puo	1353.8	Kip	Ultimate load with Mu = 0
		Pnmax	1666.2	Kip	Maximum nominal load with minimum e
		Pumax	1083	Kip	Maximum ultimate load with minimum e
				____	BALANCED_VARIABLES____________________
		Cb	10.069	in.	Depth of bal. neutral axis
		ab	8.5586	in.	Depth of bal. stress block
		εb	.002069		Balanced strain in bottom steel
		Pnb	582.37	Kip	Nominal balanced strength
		Pub	378.54	Kip	Ultimate balanced strength
		Mnb	8386.7	K-in	Nominal balanced Moment
		Mub	5451.4	K-in	Ultimate balanced Moment
		eb	14.401	in.	Load eccentricity
				____	ONE/MULTIPLE POINTS FOR Pu-Mu GRAPH
LF		C	9.0504	in.	Var. Depth of comp. neutral axis
		εt	.0026395		Var. strain in bottom steel: C = + T = -
		av	7.6928	in.	Var. Depth of stress block
		Pnv	512.41	Kip	Var. Nominal strength
		Mnv	8198.5	K-in	Var. Nominal Moment
		Puv	333.06	Kip	Var. Ultimate strength only when **εt** < yield
		e	16	in.	Var. Load eccentricity
		Muv	5329	K-in	Var. Ultimate Moment only when **εt** < yield
					Pu & Mu FOR COLUMN INTERACTION DIAGRAM
L	358	PuPLOT		Kip	Ultimate column load - must be < Pumax
L	5728	MuPLOT		K-in	Ultimate column moment
				____	WHEN **εt>** yield THE Phi FACTOR OF SAFETY CHANGES.
		Φf	.69866		Revised "final" Φ (**εt>** yield then „ > .65 or .7 but<= .9)

CONCLUSIONS

At first, students were resistant to checking their homework using the TK models. However, after the problems became more complex and the TK solution uncovered numerical errors in their work; they decided the little extra time was well worth the effort. In essence the students were checking and correcting their own work before submitting it to me. When a student's solution and TK model matched, the work lost points only for lack of neatness. Real design problems can now be assigned where all students arrive at widely varying solutions. I just describe the problem and the students are required to come up with unique solutions. The grading of these sets was completed quickly using the concrete design programs presented. I ran a TK solution when a student's TK model was in error, and if that happened, their TK probably did not match their hand calculations. It took me less than a minute to input and run a correct TK solution. I then could compare my TK to their hand solution and discover any errors in their solution quickly. I stapled my TK printout to their work. The students gain a "feel" for designing concrete structures since they can quickly obtain an alternate solution with the click of a few buttons. Students can see quickly what happens if a different concrete or steel strength is used, or a little deeper or wider cross-section is used. The ultimate test of TK's usefulness is verified when students come to me and describe how they used it in another course to reduce the amount of time spent on calculations or graphing.

REFERENCES

[1] Nawy, Edward G., Reinforced Concrete: A Fundamental Approach, Pearson Prentice Hall, Upper Saddle River, NJ 07458, 2005

[2] Universal Technical Systems, Inc., 1220 Rock Street, Rockford, IL 61101, www.uts.com

Edwin P. Foster, PhD, P.E.

Professor of Civil Engineering and Director of the Graduate Engineering Program at the University of Tennessee at Chattanooga. He received his B.E., M.S. and Ph.D. from Vanderbilt University with a year of postgraduate work at the University of Illinois at Urbana. He has over fifty publications in five countries and has been with the University of Tennessee for thirty-six years. He held engineering positions with American Bridge Division of U. S. Steel in Cocoa Beach, Florida; and Brown Engineering in Huntsville, Alabama and has consulted with many civil engineering firms. He was a NASA-ASEE Summer Faculty Fellow at NASA/Langley and has served as an elected officer in ASEE, TSPE, and has been the president of the Tennessee Section of ASCE.