A Simple Stepped-Column Buckling Model and Computer Algorithm

JORGE VASQUEZ and RAFAEL RIDDELL

ABSTRACT

The paper presents a solution to the problem of obtaining the effective length ratios needed for the design of stepped columns. The formulation of a discrete model is presented, and a numerical algorithm is implemented to solve it as a matrix eigenvalue problem. The method is an alternative to using tabulated results, which have been obtained from closed-form solutions for a number of ideal boundary conditions, which in most practical cases is cumbersome and almost always unpredictably inaccurate. A MATLAB coding of the algorithm is provided, and examples illustrating its use are included.

Keywords: effective length, stepped columns.

The problem of obtaining the effective length ratios needed for the design of a stepped crane column (Figure 1a) can be stated in closed form using the beam-column theory (Timoshenko and Gere, 1961) to describe the flexibility of each of the column segments, together with equilibrium equations that include the *P*-delta effect. Thus the beam-column formulation takes care of what is often called the P-δ effect and global member equilibrium handles the P-Δ effect. The difficulty with this exact approach is that it leads to a transcendental eigenvalue problem, that is, one in which the eigenvalue is a multiplier of arguments of functions, as for instance, of the angle of a sine function. This is much more complicated than the usual discrete matrix eigenvalue problem appearing in many fields of practical engineering. For a general case, a rather elaborate solution algorithm (Watson and Howson, 2004) will be required. However, for some specific ideal boundary conditions (free, pinned, sliding, fixed), results have been made available under tabular form, as in AISC Steel Design Guide 7, *Industrial Buildings–Roofs and Anchor Rods* (Fisher, 2004) and in AISE Technical Report No. 13, *Guide for the Design and Construction of Mill Buildings* (2003). Common practice is to interpolate the values for more realistic elastic restraint boundary conditions, as well as, of course, for parameter combinations for which the table is lacking an entry. Charts have been also prepared for the pinned-base case (Fraser, 1990) and for the fixed-base case (Fraser and Bridge, 1990). But, as in many other instances, today's designers can easily replace tables and charts by short computer programs that, for the tabulated entries, readily yield the same or better results and, without interpolation, identical precision values for parameter combinations not tabulated. Furthermore, these programs

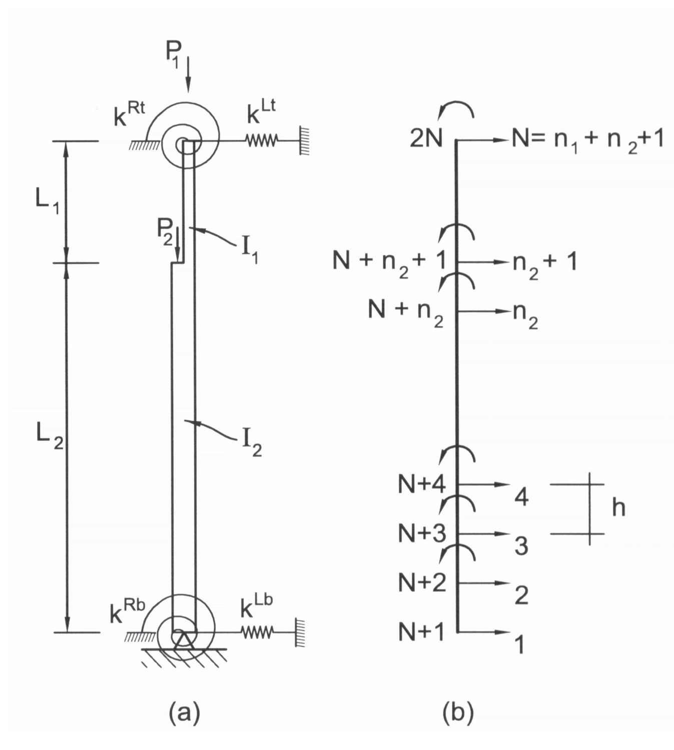

Fig. 1. The stepped-column: (a) idealization; (b) model.

Jorge Vasquez, Ph.D., Professor Emeritus, Universidad Catolica de Chile, Macul, Santiago, Chile. E-mail: mailto:jvasquez@ing.puc.cl

Rafael Riddell, Ph.D., Professor of Civil Engineering, Universidad Catolica de Chile, Macul, Santiago, Chile (corresponding author). E-mail: riddell@ing.puc.cl

often open the possibility of attaining greater generality in problem specification. This is the case of stepped-column design, in which a straightforward intuitive approach leads to a simple model that can include all of the features that need to be considered to evaluate the K factors to be used in analysis and design procedures that do require them. Some forms of second-order analysis, such as the direct analysis (Schmidt, 2001; AISC, 2005), which make no use of K factors, are unrelated to this paper. The purpose of the paper is to present the formulation of the stepped-column model and a computer algorithm that will provide the required results. A MATLAB (MATLAB, 2005) coding of this algorithm is presented. Application examples are included.

A DISCRETE MODEL FOR BEAM-COLUMN BEHAVIOR

By dividing a column into a number of short, elementary, bending-theory elements that are assembled taking into account the P-delta effect, beam-column behavior can be modeled with a precision that rapidly increases with the number of elements into which the column is divided. The division in a large number of elements numerically achieves the bending-axial interaction that in closed-form solutions would be established at a differential level.

The stiffness equation of an element of length h resulting from such subdivision can be written as:

$$(\mathbf{K}^e + f\mathbf{K}^{ge})\mathbf{q}^e = \mathbf{Q}^e \tag{1}$$

where $\mathbf{K}^e$ is the ordinary elastic stiffness matrix, $\mathbf{K}^{ge}$ is the P-delta geometric stiffness matrix of the element considering the second-order equilibrium effect of its nominal axial force P, f is an amplification factor of the axial loads, $\mathbf{q}^e$ is the local degree-of-freedom displacement vector, and $\mathbf{Q}^e$ is the local degree-of-freedom force vector. For the degree-of-freedom numbering of Figure 2, the two matrices are given as follows (Przemieniecki, 1968):

$$\mathbf{K}^e = \frac{EI}{h^3}\begin{bmatrix} 12 & 6h & -12 & 6h \\ 6h & 4h^2 & -6h & 2h^2 \\ -12 & -6h & 12 & -6h \\ 6h & 2h^2 & -6h & 4h^2 \end{bmatrix}$$

$$\mathbf{K}^{ge} = \frac{P}{h}\begin{bmatrix} -1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \\ 1 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \tag{2}$$

where E is the modulus of elasticity, I is the moment of inertia of the cross section and P is the axial load of the element. The axial force, P, can change from element to element so that, if needed, a variable axial load beam-column can be modeled.

The assemblage of the total stiffness matrix of the structure $\mathbf{K}^{tot}$ can be performed using the standard direct stiffness procedure by adding the combined elastic-geometric stiffness:

$$\mathbf{K}^{e,tot} = \mathbf{K}^e + f\mathbf{K}^{ge} \tag{3}$$

as the contribution of each element. However, it is preferable to write:

$$\mathbf{K}^{tot} = \mathbf{K} + f\mathbf{K}^g \tag{4}$$

where the matrices $\mathbf{K}$ and $\mathbf{K}^g$ result from separately assembling, by direct stiffness, the $\mathbf{K}^e$ matrices of the elements into the structure's ordinary elastic stiffness matrix $\mathbf{K}$ and the $\mathbf{K}^{ge}$ matrices of the elements into the structure's geometric stiffness $\mathbf{K}^g$. With these definitions, the stiffness equation of the model will read:

$$(\mathbf{K} + f\mathbf{K}^g)\mathbf{q} = \mathbf{Q} \tag{5}$$

Analysis will show that the structural stiffness degrades as f increases. And it degrades until the stiffness matrix becomes singular for a certain critical amplification factor f^{cr}. The meaning of this singularity is that for that amplification of the load, instability (i.e., buckling) occurs.

As in the analytical counterpart of this numerical approach, f^{cr} is independent of the applied lateral load, as represented by the load vector $\mathbf{Q}$. It is also independent of the eccentricities that the axial load may eventually present throughout the length of the member. Such eccentricities will certainly induce deformation-independent additional contributions to the load vector $\mathbf{Q}$, which are relevant in

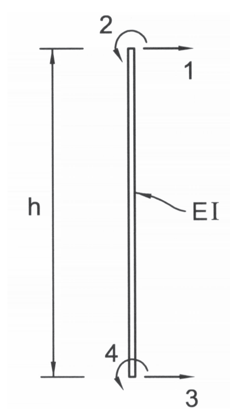

Fig. 2. Element local degrees of freedom.

beam-column stress analysis but not in buckling. Indeed, the "P-eccentricity" terms appearing in the equilibrium of the members will in no way alter the stiffness of the model, whereas instability requires a softening modification of the stiffness matrix.

The fact that buckling is eventually reached by successively increasing the f factor can be used to develop step-by-step schemes that will give f^{cr} within whatever precision is desired. Implementation of a scheme of this type is viable, but it will probably turn out to be a somewhat awkward procedure requiring an ad-hoc convergence strategy. Fortunately, there is a much more straightforward way to achieve the same goal. It arises from stating that there has to exist at least one displacement vector $\mathbf{q} = \mathbf{v}$ such that:

$$(\mathbf{K} + f\mathbf{K}^g)\,\mathbf{v} = \mathbf{0} \tag{6}$$

Actually there are as many such vectors as there are lateral displacement degrees of freedom in the model; they are the solution of the eigenvalue problem:

$$\mathbf{K}\,\mathbf{v} = -f\mathbf{K}^g\,\mathbf{v} \tag{7}$$

Consequently, for obvious physical reasons, f^{cr} is the lower eigenvalue (the one having the smallest magnitude), usually called the first eigenvalue. The corresponding $\mathbf{v}$ eigenvector is, of course, the associated buckling mode shape. The calculation of only the first eigenvalue is not at all costly; it can be easily obtained through inverse matrix iteration.

COMPUTER IMPLEMENTATION FOR CRANE COLUMNS

For the case of a typical crane column, there are only two distinct segments with different properties (Figure 1a): the upper shaft, with a moment of inertia I_1, length L_1 and a weight per unit length w_1, and the bottom shaft, with a moment of inertia I_7, length L_2 and a weight per unit length w_2. The column has a vertical applied load at the top of the column, P_1, and the crane load P_2 applied at the step. The maximum internal axial compression force in the upper shaft is then $P_U = P_1 + w_1L_1$, and in the lower shaft it is equal to the total load $P_T = P_1 + P_2 + w_1L_1 + w_2L_2$. In practical applications, the significance of w_1 and w_2 is minor.

When choosing how to divide the total length $L = L_1 + L_2$ of the column into short elements, it is convenient to make sure one gets elements of approximately equal length in both segments. With such a criterion, the total number of elements n will be found to be distributed into n_1 elements of the upper segment of the column and n_2 of the lower segment (Figure 1b).

To keep degree-of-freedom numbering as orderly as possible, it is preferable to assemble the structural system's $\mathbf{K}$ and $\mathbf{K}^g$ matrices first and only then suppress the fixed degrees of freedom the structure may have. Figure 1b shows the structural model, where, for the sake of keeping notation simple, N stands for the total number of divisions incremented by 1 ($N = n_1 + n_2 + 1$). From the model, it is easy to read the incidence index vectors of the elements needed for the direct stiffness assemblage into the $\mathbf{K}$ and $\mathbf{K}^g$ matrices. They are the vectors containing the numbers of the global degrees of freedom that coincide with the four local degrees of freedom of the element. For the first—the bottom—element its incidence vector is:

$$\mathbf{a} = \begin{bmatrix} 2 & N+2 & 1 & N+1 \end{bmatrix} \tag{8}$$

as can be read from comparing the local degrees of freedom of Figure 2 with the global degrees of freedom of Figure 1b. For the second element, the same vector $\mathbf{a}$ is used by adding 1 to all of its four components; then for the third element, the vector $\mathbf{a}$ is once again updated by adding 1 to all its components; and so on.

The model allows a quite general set of different boundary conditions. The lateral and rotational displacements at the bottom end can be specified as either fixed or as having elastic restraints, k^{Lb} and k^{Rb}, and in the same way the lateral and rotational displacements at the top end can be specified as either fixed or as having elastic restraints, k^{Lt} and k^{Rt}. The elastic restraint boundary conditions must be imposed by adding to the diagonal of $\mathbf{K}$ the stiffnesses k^{Lb} in the first row, k^{Rb} in the Nth row, k^{Lt} in the $(N + 1)th$ row and k^{Rt} in the $2Nth$ row. Finally, the rows and columns of $\mathbf{K}$ and $\mathbf{K}^g$ corresponding to whatever degrees of freedom are prescribed as fixed must be suppressed. These are the degrees of freedom assigned with the numbers 1, N, $N + 1$, or $2N$, according to whether the corresponding restraint k^{Lb}, k^{Rb}, k^{Lt} or k^{Rt} is specified as infinite, or not. A vector $\mathbf{r}$, containing the indices of the fixed degrees of freedom, determines the rows and columns of matrices $\mathbf{K}$ and $\mathbf{K}^g$ that, after all the elastic restraints have been added, have to be suppressed.

There is also allowance for the specification of lateral and rotational displacement boundary conditions at an intermediate point. This feature is intended to account for the "lateral support usually provided at the level of the crane runway girder seat" (Galambos, 1989) when analyzing weak axis buckling. The point is specified through its distance L_s to the bottom end, which might be different from L_2, and it will be readily associated to the corresponding lateral degree of freedom n_s. The lateral and rotational displacements can be either fixed or have elastic restraints, k^{Ls} and k^{Rs}. The stiffness k^{Ls} has to be added to the n_sth row of the diagonal of $\mathbf{K}$, or, if specified as infinite, the degree of freedom n_s will have to be incorporated into the index vector $\mathbf{r}$. The same holds for the rotational stiffness k^{Rs}, but with the degree of freedom $N + n_s$ taking the place of n_s.

Table 1. Computed Equivalent Length Factors for I_1/I_2 = 0.3 and L_1/L = 0.5														
	Pin-Pin		Fixed-Free		Fixed-Pin		Fixed-Slider		Fixed-Fixed		Pin-Fixed		Pin-Slider	
P_2/P_T	K_1	K_2	K_1	K_2	K_1	K_2	K_1	K_2	K_1	K_2	K_1	K_2	K_1	K_2
0.0	0.833	1.520	1.332	2.432	0.578	1.055	0.718	1.310	0.407	0.743	0.533	0.972	1.876	3.435
0.2	0.872	1.424	1.344	2.196	0.598	0.977	0.746	1.219	0.427	0.697	0.563	0.919	2.047	3.343
0.4	0.934	1.321	1.367	1.933	0.632	0.894	0.797	1.127	0.458	0.647	0.610	0.863	2.305	3.260
0.6	1.049	1.212	1.415	1.634	0.697	0.805	0.901	1.040	0.516	0.595	0.696	0.804	2.751	3.176
0.8	1.341	1.095	1.586	1.295	0.870	0.711	1.181	0.963	0.661	0.540	0.910	0.743	3.788	3.093
1.0		0.969		1.000		0.612		0.902		0.481		0.681		3.010

Once the rows and columns of **K** and $\mathbf{K}^g$ specified by the index vector **r** are suppressed, the first eigenvalue problem is solved through inverse matrix iteration. With the eigenvalue f^{cr}, the K_1 and K_2 equivalent-length factors of AISC Steel Design Guide 7, *Industrial Buildings–Roofs and Anchor Rods* (Fisher, 2004), are calculated as:

$$K_1 = \sqrt{\frac{\pi^2 EI_1}{f^{cr} P_U L^2}} \tag{9a}$$

$$K_2 = \sqrt{\frac{\pi^2 EI_2}{f^{cr} P_T L^2}} \tag{9b}$$

The Appendix contains the MATLAB coding of this implementation. In addition to the main function's output, the code will plot the buckling mode shape. Translation of the MATLAB coding to some other numerical computation software package should prove to be quite direct.

APPLICATION EXAMPLE 1

As a first test of the numerical procedure, and for comparison purposes, a block of the table in AISC Steel Design Guide 7, *Industrial Buildings–Roofs and Anchor Rods* (Fisher, 2004), was recalculated. The block chosen is the one for ratio I_1/I_2 equal to 0.3 and ratio L_1/L equal to 0.5. As in the reference, it is tabulated in terms of the ratio between the crane load P_2 and the total load P_T (which, because the reference ignores self-weight, is just equal to $P_1 + P_2$). The results K_1 and K_2, all computed with n equal to 100, are the entries of Table 1. Within four-figure precision, the values obtained undergo absolutely no change for n larger than 100 so that the numerical processes can be regarded as having converged to the exact value. Comparison shows almost total agreement, with negligible differences no larger than 0.2%, except for the pin-slider end condition, where there seems to exist a small 1% error in some of the entries of the reference's table. The following MATLAB calling sequence was used to have the function output the entries in the first row and first column of Table 1 ($P_1 = 1$, $P_2 = 0$):

```
E = 1;
I1 = 0.3; I2 = 1;
L1 = 0.5; L2 = 0.5;
P1 = 1.0; P2 = 0.0;
kLt = Inf; kRt = 0; kLb = Inf; kRb = 0;
Ls = 0; kLs = 0; kRs = 0;
w1 = 0; w2 = 0;
n = 100;
[fcr,K1,K2,Dinfo,Vinfo] = SteppedColumn(E,I1,I2,L1,L2,P1,P2,...
kLt,kRt,kLb,kRb,Ls,kLs,kRs,w1,w2,n);
```

The relevant results are $K_1 = 0.83265$ and $K_2 = 1.52020$. The rest of the pin–pin column of the table will be obtained by modifying the definition of P_1 and P_2 as appropriate. Other columns of the table are obtained by modifying the end restrictions.

APPLICATION EXAMPLE 2

As a second test, the classical problem of finding the critical load factor of a fixed-free column under its own weight was solved. The analytical solution of the simple case of a prismatic bar is given by Timoshenko and Gere (1961). Naturally, in this case, the ratio L_1/L_2 into which the column is divided turns out to be immaterial. With the data of the following MATLAB calling sequence:

```
E = 1;
I1 = 1; I2 = 1;
L1 = 0.5; L2 = 0.5;
P1 = 0.0; P2 = 0.0;
kLt = 0; kRt = 0; kLb = Inf; kRb = Inf;
Ls = 0; kLs = 0; kRs = 0;
w1 = 1; w2 = 1;
n = 100;
[fcr,K1,K2,Dinfo,Vinfo] = SteppedColumn(E,I1,I2,L1,L2,P1,P2,...
kLt,kRt,kLb,kRb,Ls,kLs,kRs,w1,w2,n); fcr
```

The result for the critical load factor given by the function SteppedColumn is indeed the same $f^{cr} = 7.837$ value derived by Timoshenko and Gere.

APPLICATION EXAMPLE 3

The SteppedColumn function is now used to obtain the effective length factors for a crane stepped-column part of the mill building frame shown in Figure 3, which corresponds to an actual design, considering the computed elastic support provided by the rest of the frame.

Due to wind load effects, exterior columns control. One crane operates in each of the bays of the building. The most critical condition is when the two cranes align with one and the same frame, having their maximum load next to the exterior column of the corresponding bay. Both exterior columns would simultaneously be subjected to their maximum load condition. The lighter-loaded central columns will provide in-plane sway constraint to the heavier-loaded columns. Symmetry leads to each interior column providing lateral restraint to its neighboring exterior column. The restraint is exerted through the roof beam, which will also provide rotational restraint to the column. The translational and rotational restraints offered to the exterior columns at their top can be regarded as being provided by the substructure shown in Figure 4. The designer can estimate the values of the corresponding elastic constants through different schemes. The one chosen here was static condensation of the substructure's stiffness matrix in terms of the two corresponding degrees of freedom (lateral and rotational displacement at the connection point). With that approach, the result $k^{Lt} = 4.868$ kip/in. and $k^{Rt} = 1.293\text{e}6$ kip-in. was reached. A coupling stiffness coefficient was also found, −604.4 kip-in./in., that, through minor modifications of the computer code, can perfectly well be handled but whose effect was found to be not sufficiently significant to worry about it. The axial loads applied are, at the top, $P_1 = 75$ kips, and, at the step, $P_2 = 180$ kips. The column bases are to be considered fully fixed. Using this data, the following MATLAB calling sequence can be written:

```
E = 30000;  %ksi
%Upper shaft data; W27x178
P1 = 75;    %kips
L1 = 10;    %ft
I1 = 6990;  %in^4
w1 = 178;   %lb/ft
%Lower shaft data; W40x298
P2 = 180;   %kips
L2 = 59;    %ftt
I2 = 24200; %in^4
w2 = 298;   %lb/ft
%Unit conversions
w1 = w1/12/1000; w2 = w2/12/1000;   %kip/in
L1 = L1*12; L2 = L2*12;             %in
```

```
%Boundary conditions data
kLt = 4.868;         %kip/in
kRt = 1.292e+006;    %kip-in
kLb = Inf; kRb = Inf;Ls = 0; kLs = 0; kRs = 0;
n = 100;
[fcr,K1,K2,Dinfo,Vinfo] = SteppedColumn(E,I1,I2,L1,L2,P1,P2,...
kLt,kRt,kLb,kRb,Ls,kLs,kRs,w1,w2,n);
```

which leads to the buckling mode shape of Figure 5. The output arguments K_1 and K_2 are posted in the figure together with f^{cr}, as well as information that may be useful in the design procedure and a complete listing of the input data.

Reading the K factors from the table in AISC Steel Design Guide 7 (Fisher, 2004) would be a quite cumbersome procedure. Indeed, it would first require performing for three cases, the fixed-free, the fixed-pin and the fixed-slider—a numerical three-way interpolation for the ratios $I_1/I_2 = 0.284$, $L_1/L_2 = 0.855$ and $P_2/(P_1{+}P_2) = 0.706$—using four entry readings for each interpolated result. This is a tedious task but straightforward. What is not at all straightforward is the next step, which calls for interpolation between the values obtained for the three ideal cases, assigning them weights supposed to reflect the effect of the lateral and rotational elastic restraint constants involved. Clearly, such is a guesswork process that can hardly be considered reliable.

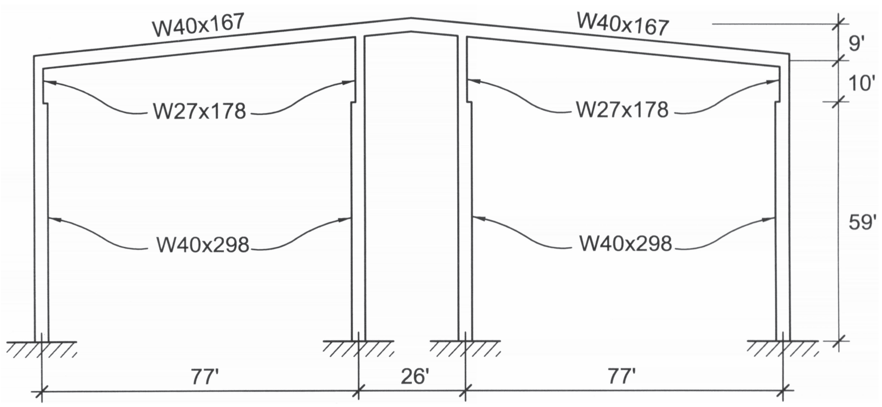

Fig. 3. Mill building frame of Example 3.

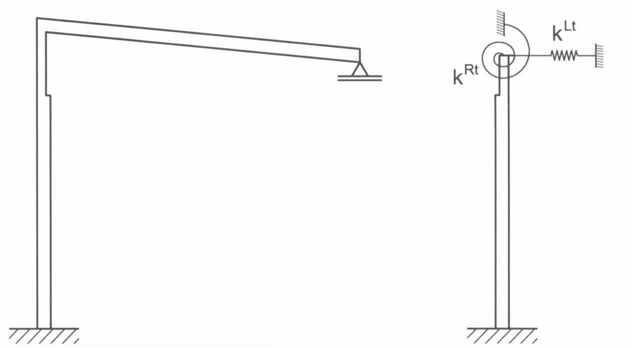

Fig. 4. Substructure providing restraint to exterior columns.

CONCLUSIONS

A simple model for a stepped column was formulated, in which the column is divided into a number of segments that are considered linear in their deformation, linear-elastic in their flexibility, but subjected to second-order effects in their equilibrium. Thus, the so-called P-Δ aspect of the P-delta effect imposed at each segment level achieves the inclusion of the P-δ aspect, as the number of segments increases, that would otherwise require the use of beam-column analytical formulas. The problem of finding the critical load, and hence the effective lengths, is then rendered a discrete matrix eigenvalue problem instead of a continuous transcendental one. A computer code, a MATLAB function, implementing an algorithm based on the model was developed, and applied to three examples. A division into 100 segments was found to be suitable in all three cases, while keeping processing requirements well within very reasonable time limits. The first example shows the complete coincidence of the results obtained from the function with entries from the well-known table in AISC Steel Design Guide 7 (Fisher, 2004). The second example is a successful comparison with a theoretical problem discussed by Timoshenko and Gere (1961), using the available, but not all that important, self-weight capability. The third example computes the "exact" K factors required for the design for individual member stability of a crane column of an actual mill building; reading the K factors from the table in AISC Steel Design Guide 7 would be a very cumbersome and unreliable procedure.

REFERENCES

AISC (2005), *Steel Construction Manual*, 13th ed., American Institute of Steel Construction, Chicago, IL.

AISE (2003), Technical Report No. 13, *Guide for the Design and Construction of Mill Buildings*, Association of Iron and Steel Engineers, Pittsburg, PA.

Fisher, J.M. (2004), *Steel Design Guide 7, Industrial Buildings—Roofs and Anchor Rods*, American Institute of Steel Construction, Chicago, IL.

Fraser, D.J. (1990), "The In-Plane Stability of a Frame Containing Pin-Based Stepped Columns," *Engineering Journal*, AISC, 2nd Quarter, pp. 49–53.

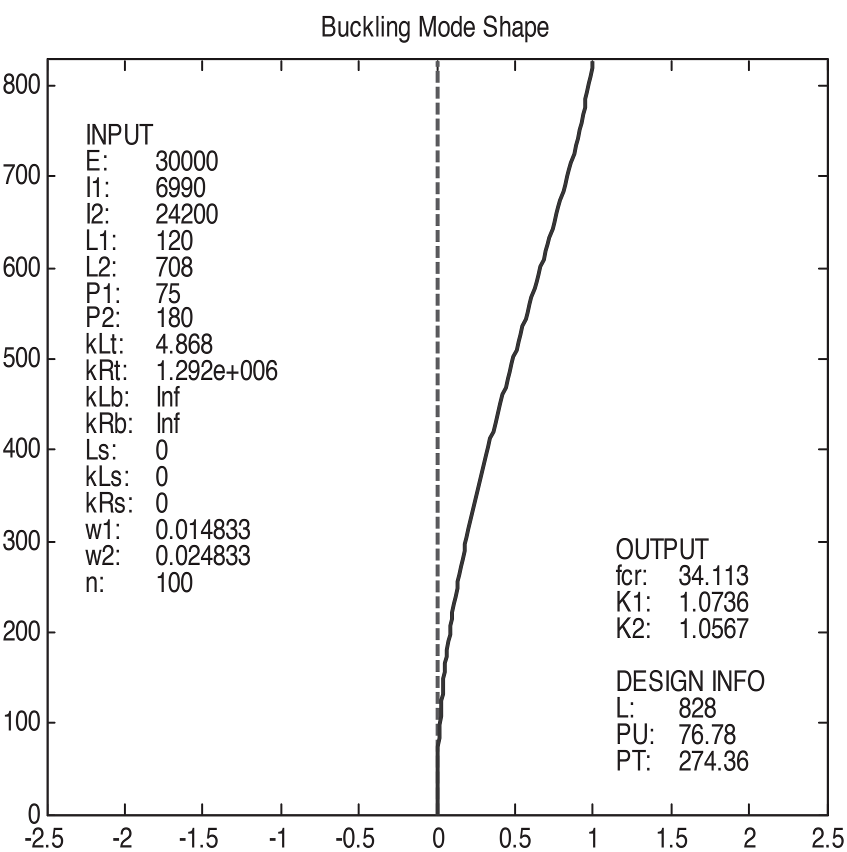

Fig. 5. Graphical MATLAB output for Example 3.

Fraser, D.J. and Bridge, R.Q. (1990), “Buckling of Stepped Crane Columns,” *Journal of Constructional Steel Research*, Vol. 16, pp. 23–28.

Galambos, T.V. (1998), *Guide to Stability Design Criteria for Metal Structures*, 5th ed., John Wiley & Sons, Inc., New York, NY.

MATLAB (2005), *The Language of Technical Computing*, Version 7.0.4, The MathWorks, Inc., Natick, MA.

Przemienecki, J.S. (1968), *Theory of Matrix Structural Analysis*, McGraw-Hill, Inc., New York, NY.

Schmidt, J. (2001), “Design of Mill Building Columns Using Notional Loads,” *Engineering Journal*, AISC, 2nd Quarter, pp. 90–99.

Timoshenko, S.P. and Gere, J.M. (1961), *Theory of Elastic Stability*, McGraw-Hill, Inc., New York, NY.

Watson, A. and Howson, N.P. (2004), “Buckling of Spring Supported Tapered Columns Allowing for Shear Deformation,” *Proceedings of the International Conference on Computational & Experimental Engineering,* Madeira, Portugal, pp. 2105–2110.

APPENDIX

MATLAB Code

```
function [fcr,K1,K2,Dinfo,Vinfo] = SteppedColumn(E,I1,I2,L1,L2,P1,P2,...
        kLt,kRt,kLb,kRb,Ls,kLs,kRs,w1,w2,n)
% DESCRIPTION:
%       computes critical load factors for a stepped column formed by two
%       segments of different cross-sectional shapes with loads both at the
%       top end and at the step, with choice of different elastic or fixed
%       end conditions; the own weight of shafts can de handled separately
% USE:
%       [fcr,K1,K2,Dinfo,Vinfo] = SteppedColumn(E,I1,I2,L1,L2,P1,P2,...
%        kLt,kRt,kLb,kRb,Ls,kLs,kRs,w1,w2,n)
% INPUT:
%       E       Modulus of elasticity
%       I1      Moment of inertia of the upper shaft cross-section
%       I2      Moment of inertia of the lower shaft cross-section
%       L1      Length of the upper shaft
%       L2      Length of the lower shaft
%       P1      Vertical load applied at the top of the column
%       P2      Vertical load applied at the step
%       kLt     Lateral spring constant at the top (Inf for fixed)
%       kRt     Rotational stiffness at the top (Inf for fixed)
%       kLb     Lateral spring constant at the base (Inf for fixed)
%       kRb     Rotational stiffness at the base (Inf for fixed)
%       Ls      Distance from base to intermediate step support point
%       kLs     Lateral spring constant at the step (Inf for fixed)
%       kRs     Rotational stiffness at the step (Inf for fixed)
%       w1      Weight per unit length of the upper shaft
%       w2      Weight per unit length of the lower shaft
%       n       Specified number of divisions of the length of the column
% OUTPUT:
%       fcr     Critical factor for the specified loads
%       K1      Upper shaft equivalent length factor =pi*sqrt(E*I1/(fcr*PU))/L
%       K2      Lower shaft equivalent length factor =pi*sqrt(E*I2/(fcr*PT))/L
```

```
%       Dinfo   Vector containing information needed for design
%               L  = L1 + L2
%               PU = P1 + w1*L1
%               PT = P1 + P2 + w1*L1 + w2*L2
%       Vinfo   Buckling mode shape information given through a three column
%               matrix: first column contains the lateral displacements;
%               second column, the rotational displacements; third column,
%               the corresponding coordinates (from the bottom)
% ALGORITHM:
%       The column is divided into n bending elements with inclusion of the
%       P-delta effect; the elastic stiffness matrix K and the geometric
%       stiffness matrix Kg are assembled, and the eigenvalue problem
%       Kc*v = f*Kg*v is solved, with a result in which fcr equal to lowest
%       eigenvalue is the critical factor of the loads
%       The first eigenvalue and eigenvector are obtained by inverse matrix
%       iteration using the adjoining function inv_mat_iter
% NOTES:
%       ALL INPUT ARGUMENTS MUST BE SPECIFIED; ENTER 0 WHERE NOT RELEVANT
%       ALL OUTPUT ARGUMENTS MUST BE SPECIFIED

L = L2 + L1;                      %Total column length
PU = P1 + w1*L1;                  %Axial force at bottom end of upper shaft
PT = PU + P2 + w2*L2;             %Axial force at bottom end of lower shaft
dL = L/n;                         %Divisions to have about the same length
n2 = round(L2/dL);
n1 = round(L1/dL);
N = n2 + n1 + 1;
dL2 = L2/n2;
dL1 = L1/n1;
kg  = [-1  0  1  0                %Base matrix for geometric stiffness matrices
        0  0  0  0
        1  0 -1  0
        0  0  0  0];
h = dL2;                          %Bottom segment elements stiffness matrix
Ke2 = E*I2*[12     6*h   -12    6*h
            6*h  4*h^2  -6*h  2*h^2
            -12   -6*h    12   -6*h
            6*h  2*h^2  -6*h  4*h^2]/h^3;
h = dL1;                          %Top segment elements stiffness matrix
Ke1 = E*I1*[12     6*h   -12    6*h
            6*h  4*h^2  -6*h  2*h^2
            -12   -6*h    12   -6*h
            6*h  2*h^2  -6*h  4*h^2]/h^3;
%ASSEMBLE STRUCTURAL SYSTEM'S ELASTIC AND GEOMETRIC STIFFNESS MATRICES
K  = zeros(2*N);
Kg = zeros(2*N);
a = [2 N+2 1 N+1];
z(1) = 0; iz = 1;
W = 0.5*w2*dL2;
```

```
for i=1:n2
    K(a,a)  = K(a,a)  + Ke2;
    Kg(a,a) = Kg(a,a) + (PT - W)*kg/dL2;
    a = a + 1;
    W = W + w2*dL2;
    iz = iz + 1;
    z(iz) = z(iz-1) + dL2;
end
W = 0.5*w1*dL1;
for i=1:n1
    K(a,a)  = K(a,a)  + Ke1;
    Kg(a,a) = Kg(a,a) + (PU - W)*kg/dL1;
    a = a + 1;
    W = W + w1*dL1;
    iz = iz + 1;
    z(iz) = z(iz-1) + dL1;
end
%ADD SUPPORT STIFFNESSES AND/OR PREPARE FOR FIXED DOF SUPPRESSION
r = [];
if isinf(kLt)
    r = [r N];
else
    K(N,N) = K(N,N) + kLt;
end
if isinf(kRt)
    r = [r 2*N];
else
    K(2*N,2*N) = K(2*N,2*N) + kRt;
end
if isinf(kLb)
    r = [r 1];
else
    K(1,1) = K(1,1) + kLb;
end
if isinf(kRb)
    r = [r N+1];
else
    K(N+1,N+1) = K(N+1,N+1) + kRb;
end
ns = round(n2*Ls/L2) + 1;       %Find ns: "step" lateral DOF
if isinf(kLs)
    r = [r ns];
else
    K(ns,ns) = K(ns,ns) + kLs;
end
if isinf(kRs)
    r = [r N+ns];
else
    K(N+ns,N+ns) = K(N+ns,N+ns) + kRs;
end
rc = 1:2*N; rc(r) = [];         %Indices of non-fixed DOFs
K  = K(rc,rc);                  %Suppress row and columns of fixed DOFs
Kg = Kg(rc,rc);
```

```
%SOLVE FOR LOWER OR FIRST EIGENVALUE PROBLEM
[V,fcr] = inv_mat_iter(K,-Kg);
%PROCESS INFORMATION FOR OUTPUT
K1 = pi*sqrt(E*I1/(fcr*PU))/L;
K2 = pi*sqrt(E*I2/(fcr*PT))/L;
Dinfo = [L; PU; PT];            %Information that may be useful for design
q = zeros(2*N,1);               %Full 2*N DOF buckling mode shape vector
q(rc) = V;
Vinfo = [q(1:N) q(N+1:2*N) z']; %Buckling mode shape information matrix
draw_mode_shape(q,N,fcr,K1,K2,L,PU,PT,z,E,I1,I2,L1,L2,...
        P1,P2,kLt,kRt,kLb,kRb,Ls,kLs,kRs,w1,w2,n)

%%%%%%%%BEGIN ADJOINING INVERSE MATRIX ITERATION FUNCTION%%%%%%%%%%%%
function [V,f] = inv_mat_iter(K,Kg);
R = chol(K);
V = R\(R'\diag(Kg)); V = V/max(abs(V));
f0 = 0;
for j=1:100
    V = R\(R'\(Kg*V));
    f = V'*K*V/(V'*Kg*V);
    if abs(f - f0)<f*1e-6;
        break
    end
    f0 = f;
    V = V/max(abs(V));
end
%%%%%%%%%%END ADJOINING INVERSE MATRIX ITERATION FUNCTION%%%%%%%%%%%%%

%%%%%%BEGIN ADJOINING DRAWING BUCKLING MODE SHAPE FUNCTION%%%%%%%%%%%
function draw_mode_shape(q,N,fcr,K1,K2,L,PU,PT,z,E,I1,I2,L1,L2,...
        P1,P2,kLt,kRt,kLb,kRb,Ls,kLs,kRs,w1,w2,n)
q = q(1:N);
MM = max(q); mm = min(q);
if abs(MM)>abs(mm)
    q = q/MM;
else
    q = q/mm;
end
clf;
plot([0 0],[0 z(N)],'r--',q,z,'b','LineWidth',2)
title('Buckling Mode Shape')
t = 0.035*L;
d = -2.25; e = -1.80; s = 0.90*L;
text(d,s,'INPUT'); s = s - t;
text(d,s,'E:')  ; text(e,s,sprintf('%0.5g',E))  ; s = s - t;
text(d,s,'I1:') ; text(e,s,sprintf('%0.5g',I1)) ; s = s - t;
text(d,s,'I2:') ; text(e,s,sprintf('%0.5g',I2)) ; s = s - t;
text(d,s,'L1:') ; text(e,s,sprintf('%0.5g',L1)) ; s = s - t;
text(d,s,'L2:') ; text(e,s,sprintf('%0.5g',L2)) ; s = s - t;
text(d,s,'P1:') ; text(e,s,sprintf('%0.5g',P1)) ; s = s - t;
text(d,s,'P2:') ; text(e,s,sprintf('%0.5g',P2)) ; s = s - t;
text(d,s,'kLt:'); text(e,s,sprintf('%0.5g',kLt)); s = s - t;
text(d,s,'kRt:'); text(e,s,sprintf('%0.5g',kRt)); s = s - t;
```

```
text(d,s,'kLb:'); text(e,s,sprintf('%0.5g',kLb)); s = s - t;
text(d,s,'kRb:'); text(e,s,sprintf('%0.5g',kRb)); s = s - t;
text(d,s,'Ls:') ; text(e,s,sprintf('%0.5g',Ls)) ; s = s - t;
text(d,s,'kLs:'); text(e,s,sprintf('%0.5g',kLs)); s = s - t;
text(d,s,'kRs:'); text(e,s,sprintf('%0.5g',kRs)); s = s - t;
text(d,s,'w1:') ; text(e,s,sprintf('%0.5g',w1)) ; s = s - t;
text(d,s,'w2:') ; text(e,s,sprintf('%0.5g',w2)) ; s = s - t;
text(d,s,'n:')  ; text(e,s,sprintf('%0.5g',n))  ; s = s - t;
d = 1.15; e = 1.55; s = 0.35*L;
text(d,s,'OUTPUT'); s = s - t;
text(d,s,'fcr:'); text(e,s,sprintf('%0.5g',fcr)); s = s - t;
text(d,s,'K1:') ; text(e,s,sprintf('%0.5g',K1)) ; s = s - t;
text(d,s,'K2:') ; text(e,s,sprintf('%0.5g',K2)) ; s = s - 2*t;
text(d,s,'DESIGN INFO'); s = s - t;
text(d,s,'L:')  ; text(e,s,sprintf('%0.5g',L)) ;  s = s - t;
text(d,s,'PU:') ; text(e,s,sprintf('%0.5g',PU));  s = s - t;
text(d,s,'PT:') ; text(e,s,sprintf('%0.5g',PT));
axis ([-2.5 2.5 0 z(N)]);
%%%%%%%%%END ADJOINING DRAWING BUCKLING MODE SHAPE FUNCTION%%%%%%%%%%
```