COMPUTER PROGRAMME FOR THE DYNAMIC ANALYSIS OF TALL REGULAR FRAMES

ANYA, C. U
Department of Civil Engineering
Federal University of Technology Owerri, Nigeria

ABSTRACT

The traditional method of dynamic analysis of tall rigid frames assumes the shear frame model. Models that allow joint rotations with/without the inclusion of the column axial loads give improved results but pose much more computational difficulty. In this work a computer program Natfrequency that determines the dynamic stiffness matrix of tall frames and solves the eigenvalue problem when modeled as a shear frame and when joint rotations with/without the inclusion of the column axial loads was developed.

Results obtained using Natfrequency were in total agreement with those obtained manually and were obtained in seconds. It is recommended that with tall buildings becoming less stiff and more susceptible to dynamic disturbances, that models, which allow joint rotations with/without the inclusion of the columns' axial loads, be used for their analysis.

INTRODUCTION.

In spite of the 2001 bombing of the d Trade Centre the demand for tall buildings has not abated. Many high-rise buildings have since been completed or nearing completion. Notable examples [1] include The Tapei 101 storey building (the world's tallest building), was completed in 2004, the 88- story International finance centre, Hong (completed in 2003) and the 60 storey Wuham International Securities. Buildings (under construction). Even in conservative United Kingdom, tall buildings seem to be enjoying popularity unlike anything seen previously [2] Tall buildings are subjected to dynamic force which induce vibration in the buildings.

Prerequisite to the design of such a structure is a good insight into its vibration motions and, in particular, the natural frequency, w. The traditional method of dynamic analysis of tall frames assumes the shear frame model in which the horizontal members are assumed to be of infinite rigidity when compared to the columns. Its major advantage is its simplicity. With modern tall buildings becoming more flexible [3] the shear frame model may not always be justified. It has, however, been shown [4] that models which permit joint rotations with/without the inclusion of the columns axial loads give improved results. Such models pose great computational difficulty and can only be analyzed using computers. In this work a computer program that determines the natural frequencies of tall, regular, rigid frames when modelled as a shear frame and when joint rotations with/without the inclusion of the column axial loads will be developed using Matlab.

2.0 DESCRIPTION OF MODELS.

Tall frames are essentially systems with infinite degrees of freedom. However, some simplifications are made in their dynamic analyses by considering them as

two dimensional-multi-degrees of freedom systems with the masses lumped at the floor levels. The models are as described below:

2.1 Model 1: The Shear Frame

This model assumes that the floor slab acting integrally with the beams makes the beams infinitely rigid when compared to the columns; the deformation of the structure is independent of the axial forces in the columns. Thus a building frame can be modelled as a vertical pole with the masses concentrated at the floor levels and the rigidities of the vertical members of the original frame, say at the i^{th} floor level summed up to give the rigidity of the pole at the i^{th} floor [5].

2.2 Model 2: Model with no restriction on joint rotation

This, like the shear frame model, assumes that the. masses are lumped at the floor levels but the girders are not assumed to be of infinite rigidity when compared to the columns. The effect of vertical inertia is negligible and the axial deformation of the structure is independent of the axial forces in the columns. When the effects of column axial loads are included, the columns are considered as beam-column elements. The conventional beam element stiffness is then modified by the so-called stability functions

3. EQUATION OF FREE MOTION

The dynamic analysis of structure using the principles of mode superposition leads to the generalized eigenvalue problem:

$$K\lambda = \omega^2 M\lambda \qquad (1)$$

Where K is the lateral stiffness matrix and M is the mass matrix, both of order n. K and M are usually banded. The solutions to equation 1 can be written as

$$K\psi = M\psi\Omega \qquad (2)$$

Where, ψ is a matrix of eigenvectors, Ù is a diagonal matrix listing the eigenvalue (square of the free vibration frequencies) In a lumped mass analysis, which is usually assumed for tall frames, K and M are positive definite and M is also diagonal with M_{ii} positive.

The solution of the generalized eigenvalue problem has been discussed extensively [6-8]. In many methods the problem is first reduced to the standard form which, when M is diagonal, is given as:

$$K\lambda = \omega^2 M\lambda \qquad (3)$$

Where; $K = M^{-1}K \qquad (4)$

It has been shown that the most time consuming phase in most dynamic analysis of structures [8] is the solution of the resulting eigenvalue problem. The solution is time consuming especially for tall frame structures. In this work, great advantage will be taken by the use of Matlab programming [9] through which the solution of such problems can be achieved in seconds.

4.0 EVALUATION OF THE LATERAL STIFFNESS MATRIX ELEMENTS.

The major difference in the models lies in lateral stiffness matrices, *K* of equation 3 The element K_{jj} of the lateral stiffness matrix is the restoring force at floor level i when floor level *i* is given a unit sway, all other floor displacement being assumed to be zero.

Stiffness matrix elements for Model l

if floor i (i=1 to n, the degree of freedom) a unit sway then the relevant stiffness matrix elements are given as:

$$K_{i,i} = 12\left(\frac{EI_{i-1}}{l_{i-1}^3} + \frac{EI_i}{l_i^3}\right), K_{i+1,i} = \frac{-12EI}{l_i^3}, K_{i-1,i} = \frac{-12EI_i}{l_{i-1}^3} \qquad (5)$$

$$K_{j,i} = 0, \; for \neq i-1, i, i+1$$

Stiffness matrix elements for Model 2

stiffness matrix elements can only be

obtained after a complete static analysis of the structure, using, say, the classical displacement method.

To evaluate the elements $K_{i,j}$ *(i=1* to *v,j=1 to v; v* being the dynamic degree of freedom) the structure is assumed is to be given a unit lateral displacement at floor level *j,* other lateral displacements r floors being assumed to be zero. The compatibility equations are then set up and are given as:

$$r_{1,1}Y_1 + r_{1,2}Y_2 + ... + r_{1,v}Y_v = -R_1X_k$$

$$r_{2,1}Y_1 + r_{2,2}Y_2 + ... + r_{2,v}Y_v = -R_2X_k$$

.. ..

$$r_{v,1}Y_1 + r_{v,2}Y_2 + ... + r_{v,v}Y_v = -R_vX_k \quad \ldots. \quad (5)$$

Where r_{ij} (*i* = 1 to *v,* j = 1 to *v)* is the moment 'generated at joint *i* due to a unit rotation at joint j. R_iX_K is the moment generated at joint *i* due to a unit sway at floor level *k,* Y_i, (i = 1 to *v)* is the unknown rotation at joint *i.*

Equation 5 is solved and the final member-end moments obtained. The shear forces and hence the stiffness matrix elements, $K_{i,k}$ *(i* = 1 to *n,)* at each floor level can now be calculated using the local equilibrium principle.

The process is repeated for each degree of freedom. It should be noted that only the right hand side of equation 5 changes each time the process is repeated.

5.0 COMPUTER PROGRAMMING

5.1 Description of the structure and arrays to the computer.

Tall frames usually have a regular geometry and distribution of joints that make the assignment of structural information to the computer and its subsequent handling and transfer relatively easy. Each member, joint or force action, and any quantity which can be used to describe the structure, or its behaviour, can be represented by the content of an element in a rectangular array. The array element and the quantity that it represents occupy the same relative positions in the array and structure' respectively. This one to one relationship greatly simplifies the identification of numerical information with its origin in the computer. Neither joints nor members need be formally numbered; their locations are known when the integer variables *y* and z are given values which specify the floor level and the bay or vertical column line the arithmetic operations are to be carried out. The method outlined in [10] will be adopted for this work and it is summarized below. Figure 5.1 shows the arrays associated with the definition of multi-storey, multi- bay frame. Consider the frame shown in figure 5.la. If c is the number of storeys and *b* the number of bays the structure will have c x *d* beams. Some information pertaining to these beams may be conveniently held in arrays of c rows and *d* columns (figure 5.lb). Thus BS ( ) could be an array containing the beam spans and BS (1,2) will be the value of the span at the roof and the second bay. The frame also has c x *(d+I)* columns and *(c=L)* x *(d=l)* joints. Associated joint arrays will have *(c+ 1)* rows and *(d+* 1) columns whilst the column member arrays will have c rows and *(d+ 1)* columns as shown in fig 5.lc and 5.ld respectively. Using the convention that at a four-member joint the end of the members are labelled North, South, East and West (figure 5.le) the bending moments at member ends are identified as *NM, SM, EM,* and WM. The member shear forces are similarly identified. Each family of force action is stored in its own array that has *(c+L)* rows and *(d+l.)* columns.

From fig 5.le, it can be seen that the bending moment at the left hand end of the beam identified by *SB(y,* z) is *EM(y,* z) and at the right end is *WM(y,* z+ 1).

The stiffness matrix of equation 5 is symmetric and banded with a half

bandwidth of *b+l.* so that only the upper triangular elements need be stored. Since it is only the right hand side of equation 5 that changes each time the process is repeated, it is advantageous to determine the values of the right hand side for each degree of freedom and store them in a single array so that the resulting sets of simultaneous equations could all be solved at the same time using the Gauss elimination method.

5.2 Computer program *Natfrequency.*

A computer program *Natfrequency* was developed for the generation of the stiffness matrix of tall regular frames. The necessary input data are the number of storeys, *nstoreys*, and the number of bays, *nbays*, beam and column lengths and EI values (kNm^2), Others are the *roof load* (kN), the floor load, *floor-load* (kN), the model type *mod;* mod is 1 for a shear frame and 2 for frames that allow joint rotations. If mod is 2, then one has to input the *type; type* is 1 when only rotations are allowed and 2 when rotations and axial load effects are considered.

The column and beam properties and the lumped masses are entered from the topmost roof from left to the right. The lumped masses are calculated using only the un-factored dead mass of the slabs and neglecting the claddings. The column axial loads are calculated on the simple assumption that a column carries a half of the loads of the beams framing into it. To run the program, type *Natfrequency* after the Matlab command window prompt.

Natfrequency first formulates the left hand side coefficients of the compatibility equation and stores them in a banded form the half bandwidth being *nbays +1*. The right hand side of the equation is then formulated for all the degrees of freedom. It then solves the resulting simultaneous equations and calculates the member end shear forces from which the stiffness elements are determined. It then transforms the generalised eigenvalue problem to the standard form and determines the eigenvalues using the Matlab command *eig.* The result is the display of the stiffness matrix, and the natural frequencies. When the effect of joint loads is included, a column is assumed to carry a half of the loads on the beams framing into it. The associated modes of vibration can also be displayed, if required. The listing of *Natfrequency* is given in Appendix 1 while the definition of the terms and variables is given in Appendix 2.

6.0 TEST PROBLEMS AND RESULTS

Natfrequency will be used to determine the natural frequencies of two regular frames. The first, the simple three storey-two-bay building frame of figure 6.1, has been chosen to illustrate the use of *Natfrequency* and to also serve as a basis for comparing the computer and the manual solutions. The second is the fifteen storey-three bay building frame of figure 6.2

The input data for the three storey frame of figure 6.1 is as below:

nbays = 2, *nstoreys* = 3, *roof_load* = 9.12, *floor _load* = 13.62 *bl* = [6,4] *cl* = [3.5,3.5,5.5]

be = [13021, 13021, 13021, 13021, 13021, 13021]

ce = [13021, 15625, 13021; 13021, 15625, 13021; 13021, 15625, 13021];

1m = [14400, 18900, 18900];

The results obtained for the various models are given below; input mod> 1

The lateral stiffness matrix for shear frame model (kN/m)

1.0e+004 *

1.1662 -1.1662 0

-1.1662 2.3324 -1.1662

0 -1.1662 1.4667

The natural frequencies (rads/secs) are:
7.1266 28.1213 44.4788
input mod> 2
input type> 1
Lateral stiffness matrix (kN /m) when joint rotations are considered.
1.0e+004 *

0.6122 -0.7923 0.1954
-0.7923 1.6480 -0.9752
0.1954 -0.9752 1.1342

The natural frequencies (rads/secs) are:
5.4529 19.8987 38.3599
input mod> 2
input type> 1
Lateral stiffness matrix (kN /m) when joint rotations are considered.
1.0e+004 *
0.6122 '-0.7923 0.1954
-0.7923 1.6480 -0.9752
0.1954 -0.9752 1.1342
The natural frequencies (rads/secs) are:
5.4529 19.8987 38.3599
For the fifteen storey-three bay frame of fig 6.2, the input data are as follows: *nbays* = 3; *nstoreys* = 15; *roof-load* = 10; *floor-load* = 13; *wf* = 3.6;
cl = [3,3,3,3,3,3,3,3,3,3,3,3,3,3,5];
be=[151200, 151200, 151200; 151200, 151200, 151200; 151200, 151200, 151200 151200, 151200, 151200; 151200, 151200, 151200; 151200,151200,151200 151200, 151200, 151200; 151200, 151200, 151200; 151200,151200,151200 151200,151200, 151200; 151200, 151200, 151200; 151200, 151200,151200 151200, 151200, 151200; 151200, 151200, 151200; 151200, 151200, 151200];
ce = [192238, 145833, 145833, 192238; 192238, 145833, 145833, 192238 192238, 145833, 145833, 192238; 192238, 145833, 145833, 192238 192238, 145833, 145833, 192238; 192238, 416515, 416515,192238 192238,416515,416515, 192238; 192238, 416515, 416515, 192238, 192238, 416515, 416515, 192238; 192238, 416515, 416515, 192238 256317
bl=[7.2,3.6,7.2];
Im=[23328, 23328, 23328, 23328, 23328, 23328, 23328, 23328, 23328, 23328, 23328, 23328, 23328, 23328, 23328, 23328, 23328];
The natural frequencies (rads/ sec) obtained using *Natjrequency* for the various models are:
For shear frame (model 1)
Columns 1 through 8
0.4339 1.1569 1.9925 2.8888 3.6791 4.5081 5.1553 5.9406
Columns 9 through 15
6.4729 6.9088 7.4618 8.3537 9.1787

When only joint rotations are considered (model 2 type 1)
Columns 1 through 8
0.1989 0.5820 0.9999 1.4710 1.9691 2.5566 3.1852 3.8824
Columns 9 through 15
4.6558 5.3836 6.2505 6.8164 7.8668 8.9347 10.2055
When joint rotations and column axial loads are considered (model 2 type 2)
Columns 1 through 8
0.1988 0.5817 0.9994 1.4703 1.9680 2.5552 3.1833 3.8799
Columns 9 through 15
4.6529 5.3802 6.2471 6.8125 7.8614 8.9298 10.1996.

7.0 DISCUSSION OF RESULTS

A comparison of the natural frequencies obtained for the three storey building frame of Figure 6.1 to those obtained using manual computations [3] shows no differences; thus confirming the accuracy of *Natfrequency*. The computer results for each of the models were however obtained within seconds.

The beauty of the *Natfrequency* is further appreciated when it is realized that to

determine the stiffness elements of the fifteen storey frame of figure 6.2, for model 2, one has to solve a set simultaneous equation with 60 unknown joint rotations and for 15 times. This will be impossible to achieve manually. Yet the total solutions, including the solution of the resulting eigenvalue problem were accomplished within seconds using the computer program. Improved results were obtained when joint rotations were included. The results were further improved when the column axial loads and joint rotations were considered.

8.0 CONCLUSION

A computer program for determining the natural frequencies of tall, regular, rigid frames using various models was developed. Results obtained using the computer program were in total agreement with those obtained manually. Tall rigid regular frames can be modeled using any of the models discussed. The choice of which model to use is determined by the degree of accuracy required.

REFERENCES

1. Council on Tall buildings and Urban Habitat (2005) www.ct.org.

2. Wood A New Paradigms in high-rise design *CTBUH Journal Issue No 3, 2004.*

3. Rahgozar, R, Safari, H. and Kaviani, P. *Free vibration of tall buildings using Timoshenko beam with variable cross-sections.* In Jones, N. and Brebbia, C.A (Editors) Structures under shock and impact viii, WIT Press, 20043.

4. Anya, C. U. *Dynamic Analysis of Tall Buildings* -Unpublished Msc project report, Department of Civil Engineering, University of Nigeria, Nsukka.1995.

5. Ymanskago, A A, *Handbook for designers* ii (in Russian language) Stroiizdat, Moscow, 1973.

6. Bathe, K. J, *Finite Element Procedures* in *Engineering Analysis,* Prentice-Hall, N.}, 1982

7. Bathe, K.J and Wilson, E. L *Large Eigenvalue Problems In Dynamic Analysis,* ASCE, Mechanics Division. Pp1471-1485, 1972.

8. Bathe, K.J and Wilson, E. L *Solution methods for Eigenvalue problems in structural mechanics,* International journal of Numerical Methods in Engineering, Vol 6 pp 213-226, 1973

9. Mathworks Inc. *Using Matlab Version* 6, Natick, Massashusetts, 2002.

10. Clark D. *Computer Aided Structural Design* John Wiley and sons, 1980.

APPENDIX 1

LIST OF SOME ARRAYS AND VARIABLES USED IN NATFREQUENCY

Arrays

bl, be – contain beam lengths and EI values respectively
ca, ce, cl – contain column axial loads, lengths and EI values respectively
f, h, g – contain beam EI/l. column EI/l and column EI/l^2 values respectively.
nm, sm, em, wm – contain North, South, East and West member-end moments respectively
ns, ss, es, ws - contain North, South, East and West member-end shears forces respectively.
lm – contains the lumped masses at the floor levels.
k - Initially contains the stiffness matrix elements and finally the dynamic stiffness matrix.
p – Initially contains constants of the compatibility equation but finally contains the joint rotations.
r – contains the co-efficient matrix in banded form
nf – contains the natural frequencies.
nn – contains the eigenvectors
sd – contains the eigenvalues.

Variables

nstoreys, nstoreys – The number of storeys and bays respectively.
mod – The type of model (mod =1 for a shear frame and 2 for joint rotations with/without axial loads).
type – for joint rotations only (type = 1 when only axial loads are considered and 2 when the column axial loads are included).
v = total number of joints excluding the supports.
e, q, y, z – operation counters;
t – half bandwidth
roof_load – design roof load in kN/m^2.
floor_load – design floor load in kN/m^2.

APPENDIX II
Listing of *Natfrequency.*

```
% This program Natfrequency solves the natural frequencies...
% of tall regular frames when modelled as a shear frame, when...
% joint rotations are permitted with/without the inclusion of...
% the column axial loads
% Inputs are the number of storeys nstoreys, the number of bays nbays,
% the beam and column EI values, the roof and floor loads, the lumped
% masses at the floor levels.
nstoreys = input (' input number of storeys, nstoreys, >');
nbays = input (' input no of bays, nbays >');
roof_load = input (' input roofload (kN/m^2), rl, >');
floor_load = input (' input floor load (kN/m^2) fl >');
lm= input ('input lumped mass(kg) at floor levels >');
be = input (' input beam EI values (kNm^2), be, > ' );
bl = input (' input beam spans (m), bl > ' );
cl = input (' input column lengths values (m), cl > ' );
ce = input (' input column EI values (kNm^2), ce > ' );
wf = input (' input floor panel width (m), wf > ' );
b = nstoreys + 1; e = nbays +1; v = nstoreys * e; t = nbays + 2;
mod = input ('input mod > '); % mod = 1 or 2. It is 1 for a shear frame...
        % and 2 when joint rotations are allowed.
mod = input ('input mod > '); % mod = 1 or 2. It is 1 for a shear frame...
        % and 2 when joint rotations are allowed.
 if mod < 1 | mod > 2; error ('wrong model! rerun'); end
switch mod
case 1; {mod==1}; % shear frame
  for q = 1:nstoreys;
   ns = zeros(b,e);  ss = zeros(b,e);   ws = zeros(b,e); es = zeros(b,e);
   for y = 1:e; ss(q,y)=12*ce(q,y)/cl(q)^3; ns(q+1,y)=ss(q,y);
      if q==1; continue; else
         ss(q-1,y) = -12*ce(q-1,y)/cl(q-1)^3; ns(q,y) = ss(q-1,y);
   end; end
      for z=1: nstoreys; k(z,q)=0;
       for y = 1:e; k(z,q) = k(z,q) - (ns(z,y) - ss(z,y)); end; end; end
fprintf('\n The lateral stiffness matrix for shear frame model (kN/m)\n' ); disp(k)
case 2 % models that allow joint rotations
   {mod==2};
   repmat(bl,nstoreys,1);   f=be./ans;   repmat(cl',1,e);
 type = input ('input type > '); %type = 1 when only joint rations...
 % are allowed and 2 when axial loads are included.
    if type > 2; error ('wrong type! rerun');  end
   if type==1; h = ce./ans; g = h./ans; % vectorized calculation
   else
ca = zeros(nstoreys,e); % columnm axial loads for type 2
x=wf*roof_load; xy=wf*floor_load;
ca(1,1) = x * bl(1)/2; ca(1,e) = x * bl(nbays)/2; zs(1) = xy * bl(1)/2; zs(e) = xy * bl(nbays)/2;
for y=2:nbays
   ca(1,y)= x*(bl(y-1)+bl(y))/2; zs(y) = xy*(bl(y-1) + bl(y))/2;
end
for y=2:nstoreys; for z=1:e; ca(y,z) = ca(y-1,z)+ zs(z); end; end
% stability functions and element stiffness arrays
A = sqrt(ca./ce).*ans; A1= 2 - 2 * cos (A) - A .* sin (A); h =(A.*(sin (A) - A .* cos (A)) .*ce)./(4.* A1 .*ans);
g = (A.^2 .*(1-cos (A)) .*ce)./(6.* A1 .*ans.^2); u = (A.*(A - sin (A)) .* ce)./(2.* A1 .*ans); end
r = zeros(v,t); % elements of co-efficient matrix in banded form
for y=1:nstoreys;   for z = 1:e; pp=(y-1) * e + z;
```

```
      if y==1; r(pp,1) = r(pp,1) + 4*h(y,z);   else; r(pp,1) = 4 * h(y-1,z) + 4 * h(y,z); end
      if y < nstoreys; if type==1; r(pp,t) = 2 * h(y,z);
        else; r(pp,t) = 2 * u(y,z);   end; end
       if z==e; r(pp,1)=r(pp,1) + 4*f(y,z-1);
       elseif z==1; r(pp,1) = r(pp,1) + 4 * f(y,z); r(pp,2) = 2 * f(y,z);
      else; r(pp,1) = r(pp,1) + 4 * f(y,z-1)+ 4 * f(y,z); r(pp,2) = 2*f(y,z); end; end; end
p=zeros(v,nstoreys); % array of constants
for q= 1:nstoreys; nm=zeros(nstoreys,e); sm = zeros(nstoreys,e);
  for z=1:e
    sm(q,z) = -6*g(q,z); nm(q+1,z)= sm(q,z);
    if q > 1; sm(q-1,z) = 6*g(q-1,z);  nm(q,z) = sm(q-1,z); end; end
   for y=1:nstoreys; for z=1:e; pp=e*(y-1)+z; p(pp,q) = -(sm(y,z)+nm(y,z)); end; end; end
for k = 1:v-1 % solve simultaneous equation using Gauss elimination
  k0 = r(k,1);   k1=k+1; k2 = k1+ nbays;  kk=min(v,k2);
   z = 2: nbays + 2; dd(z) = r(k,z); z = k1:kk; k3 = z-k+1; r(k,k3)= r(k,k3)/k0;
   z = 1:nstoreys;  p(k,z) = p(k,z)/k0;
  for y = k1:kk;
    k3 = y-k1+2; k0=dd(k3); z = y:kk;
      k3 = z-y+1; k4 = z-k+1; r(y,k3) = r(y,k3)-k0*r(k,k4);
     z=1:nstoreys; p(y,z) = p(y,z)-k0*p(k,z);  end; end
z = 1:nstoreys;  p(v,z) = p(v,z)/r(v,1);
for y = 1:v-1; k = v-y; k1 = k+1; k2 = k1 + nbays;  kk = min(v,k2);
    for z = k1:kk; k3 = z-k+1; n= 1:nstoreys; p(k,n)= p(k,n)-r(k,k3)*p(z,n); end; end
   for q=1:nstoreys; % member-end moment and shear forces
  ns = zeros(b,e); ss = zeros(b,e); ws = zeros(b,e); es =zeros(b,e);
  nm = zeros(b,e); sm = zeros(b,e); wm = zeros(b,e); em = zeros(b,e);
 for z=1:e; sm(q,z) = -6*g(q,z); nm(q+1,z) = sm(q,z);
   if q > 1; sm(q-1,z) = 6*g(q-1,z); nm(q,z) = sm(q-1,z); end; end
 for y=1:nstoreys; for z=1:e % final column-end moments
pp=(y-1)*e+z; sf=pp+e; if y == nstoreys; if type == 2
   sm(y,z) = sm(y,z)+4*h(y,z)*p(pp,q); nm(y+1,z) = nm(y+1,z)+2*u(y,z)*p(pp,q);
else
   sm(y,z) = sm(y,z)+4*h(y,z)*p(pp,q); nm(y+1,z) = nm(y+1,z)+2*h(y,z)*p(pp,q); end; end
if y ~=nstoreys;    if type == 2
   sm(y,z) = sm(y,z)+4*h(y,z)*p(pp,q)+2*u(y,z)* p(sf,q);
   nm(y+1,z) = nm(y+1,z)+4*h(y,z)*p(sf,q)+2*u(y,z)*p(pp,q);
else; sm(y,z) = sm(y,z)+4*h(y,z)*p(pp,q)+2*h(y,z)* p(sf,q);
   nm(y+1,z) = nm(y+1,z)+4*h(y,z)*p(sf,q)+2*h(y,z)*p(pp,q); end; end; end; end
for y=1:nstoreys % for beams
  for z = 1:nbays; pp = (y-1)*e+z; s = pp+1; em(y,z)=4*f(y,z)*p(pp,q)+2*f(y,z)*p(s,q);
    wm(y,z+1)=4*f(y,z)*p(s,q)+2*f(y,z)*p(pp,q); end; end
for y=1:nstoreys % column end shear forces
  for z =1:e; ns(y+1,z)= -(sm(y,z)+nm(y+1,z))/cl(y); ss(y,z) = ns(y+1,z); end; end
for y = 1:nstoreys % beam end shear forces
  for z = 1:nbays; es(y,z) = -(wm(y,z+1)+em(y,z))/bl(z); ws(y,z+1) = es(y,z); end; end
for y=1:nstoreys % stiffness elements
  k(y,q)=0; for z=1:e; k(y,q)=k(y,q)-(ns(y,z) - ss(y,z)); end; end; end
  if type == 1
 fprintf('Lateral stiffness matrix (kN/m) when joint rotations are considered.\n')
else
 fprintf('Lateral stiffness matrix (kN/m) when joint rotations \n')
 fprintf('are considered and column axial loads included.\n')
end; disp(k); end; lm = lm/1000; repmat(lm',1,nstoreys); % convert to standard form
k=k./ans; [nn,sd]=eig(k); % solve for eigenvalues and eigenvectors
sdd = diag(sd); ds=sort(sdd); nf=sqrt(ds);
fprintf('The natural frequencies (rads/secs) are :'); disp (nf');
```