FINITE ELEMENTS EF3D PROGRAM

USER'S CONDENSED MANUAL

By Jean-Pierre Dumont

English Version By Jean-Pierre Moreau

1. Introduction

Why EF3D ? This small program does not aim to compete with sophisticated finite elements softwares used in the aerospace and automobile fields. It is just a small interactive tool allowing an engineer with a PC or any small computer to rapidly calculate the eigenfrequencies and eigenmodes of mechanical systems with some dozens of degrees of freedom (dof). It can be easily used for parametric studies in pre-design phase of any project.

The mode frequencies and shapes can then be reused for response or transfer functions calculations.

This software was implemented with (C) Borland International Turbo-Pascal for Windows 1.5 from a former pascal version for Dos. A 2D graph capability is provided by unit graph_2d.pas. The data text file can be written with any word processing unit.

2. Structural elements available

With present release, you can evaluate the eigenfrequencies and eigenvectors of some simple linear, plane or three-dimensional pieces of structure.

The available structural elements are the following:

1. **BARRE**: Bar in tension/compression

This element has stiffness only in tension or compression towards the main longitudinal axis; it has no stiffness at all towards a transverse axis. The corresponding mass matrix is coherent.

2. **TORS**: Bar in torsion

This element only has stiffness in differential rotation of both ends around the main longitudinal axis. The corresponding mass matrix is coherent.

3. **BEAM**: General beam element in bending

This element has also the behavior of a bar in tension/compression and in torsion. The mass matrix is coherent and not diagonal. However, the rotatory energy of beam sections are not taken into account.

4. **BEAMS**: General beam element in bending

Same as BEAM but with shearing also taken into account. Same mass matrix as previous element.

3. Input data text file

Before using EF3D you have to prepare a data text file. This section explains how to make a data text file. The data file name will be first asked for by the program. You can use any name, for instance Pb007.dat, the « dat » suffix reminds you it is a data file. This file is supposed to be located in the same disk sub-directory (or else specify complete path). You can use any word processing software to prepare this data file, notepad included.

3.1 file composition

The data file imperatively consists of four parts:

a. meshing description: nodal coordinates and description of dof (free or fixed),

b. description of elements: numbering of ending nodes, material and section properties,

c. description of optional elastic springs between two dof in structure,

d. description of local masses and inertias added in some nodal points and towards some degrees of freedom.

3.2 detailed data description

Data concerning the structure are given line by line. The prescribed order in each line must be respected with at least one space between two consecutive data. Real numbers can use a decimal point or not. However, the program does not accept a real number when an integer number is required (example: number of nodes).

3.2.1 **Part I**: data concerning nodal points.

These data are obligatory.

- Line #1: problem caption (max. 80 characters)
- Line #2: total number of nodes (integer) from 1 to 89
- Lines #3 to #n: Num X Y Z idl(1,Num) idl(2,Num) ... idl(6,Num)

Num =	node number (integer) from 1 to 32767.
X,Y,Z =	coordinates in chosen cartesian system of axes.
Idl(1,Num)	Statute of first dof of node #Num
	=0 free
	=1 fixed
	The code for 6 possible dof is: 1: displacement towards X 2: displacement towards Y 3: displacement towards Z 4: rotation around X 5: rotation around Y 6: rotation around Z

Note 1: fixed dof are along axes of general axes system.

Note 2: there are as many such lines as the number of nodal points.

· Last line: 999 (integer): end of part I.

3.2.2 **Part II**: data concerning structural elements.

These data are optional since you have the possibility to define the model elasticity by using elastic springs (see Part III). Elements data consist of as many distinct blocks as you have different kinds of elements: for example, if you have BARRE and BEAM elements in your model, there will be two blocks. You can have more than four blocks but in each block it must be the same kind of element.

· Line #1: Nel, Type_element

Nel: numner of elements of kind Type_element

Type-element: one of (BARRE, TORS, BEAM, BEAMS)

Other lines: depend on kind of element:

Case of BARRE:

N1 N2 E Dext Dint Rho

N1 number of first ending node (integer)

N2: number of second ending node (integer)

E: Young's modulus of material (real)

Dext: external diameter of section (real)

Dint: internal diameter of section (real)

Rho: volumic mass (real)

Case of TORS:

N1 N2 G Dext Dint Rho

N1:Case of TORS:

N1: number of first ending node (integer)

N2: number of second ending node (integer)

G: shearing modulus of material (real)

Dext: external diameter of section (real)

Dint: internal diameter of section (real)

Rho: volumic mass (real)

Case of BEAM:

N1 N2 E G Dext Dint Rho

N1:Case of TORS:

N1: number of first ending node (integer)

N2: number of second ending node (integer)

E: Young's modulus of material (real)

G: shearing modulus of material (real)

Dext: external diameter of section (real)

Dint: internal diameter of section (real)

Rho: volumic mass (real)

Case of BEAMS:

N1 N2 E G Dext Dint Rho Ky

N1:Case of TORS:

N1: number of first ending node (integer)

N2: number of second ending node (integer)

E: Young's modulus of material (real)

G: shearing modulus of material (real)

Dext: external diameter of section (real)

Dint: internal diameter of section (real)

Rho: volumic mass (real)

Ky: (real) = 0.5 thin circular section = 0.9 full circular section.

- Last Line: 999 (integer) end of Part II.

3.2.3 **Part III**: data concerning added springs (optional).

- Line #1: Number_of_Springs (integer)

There can be as many elastic springs as couples of free dof. If you define a spring between two fixed dof, there will be a warning but the program will continue all the same (no action).

- Lines #2 ... n: Node1 dof1 Node2 dof2 Stiffness

Node1: (integer) number of first connected node

dof1: (integer) from 1 to 6, code of first dof connected

Node2: (integer) number of second connected node

dof2: (integer) from 1 to 6, code of second dof connected

stiffness: (real) spring stiffness = Force applied to spring to get a unitary elongation.

There are as many such lines as the number of added springs.

- Last line: 999 (integer) end of Part III.

3.2.4 **Part IV**: data concerning added masses (optional).

Local masses are masses when connected to translation dof or inertias when connected to rotation dof. Inertias are defined in an axes system parallel to the general axes system passing by the nodal point.

- Line #1: Number_of_Masses (integer)

There can be as many as six local masses for each nodal point (one per free dof). If you connect a mass or an inertia to a fixed dof, there will be a warning but the program will continue all the same (no action).

- Lines #2 ... n: Node dof mass (or Inertia)

Node: (integer) number of node

dof: (integer) from 1 to 6, code of connected dof

Node2: (integer) number of second connected node

dof2: (integer) from 1 to 6, code of second dof connected

mass: (real) local mass (if dof<4) or local inertia (if dof>3)

There are as many such lines as the number of added local masses.

· Last line: 999 (integer) end of Part IV.

Note: this last line is obligatory even if there are no added local masses.

4. Data Set Example

The model hereafter described consists of one beam with six BEAM elements, fixed in x=0, of length 6 meters, diameter 10 mm made of steel. This beam rests on an elastic foundation made of six springs, connecting the Y translation dof of nodes 2 to 7 for the beam to fixed nodes 8 to 13 of a rigid basis. These springs have a stiffness of 1000 N/m.

Here is the corresponding data text file:

```
Beam on elastic foundation                   (problem caption)
13                                           (number of nodal points)
1 0 0 0 1 1 1 1 1 1                          (node #1 X=Y=Z=0 all six dof fixed)
2 1 0 0 1 0 1 1 1 0                          (node #2 X=1, Y=Z=0, dof 2 and 6 free)
3 2 0 0 1 0 1 1 1 0                          (etc.)
4 3 0 0 1 0 1 1 1 0
5 4 0 0 1 0 1 1 1 0
6 5 0 0 1 0 1 1 1 0
7 6 0 0 1 0 1 1 1 0                          (end beam)
8 1 0 0 1 1 1 1 1 1                          (begin foundation)
9 2 0 0 1 1 1 1 1 1
10 3 0 0 1 1 1 1 1 1
11 4 0 0 1 1 1 1 1 1
12 5 0 0 1 1 1 1 1 1
13 6 0 0 1 1 1 1 1 1
999                                          (end section nodal points)
6 Beam                                       6 elements of kind BEAM)
1 2 200E9 76.92E9 0.010 0. 7800              (begin/end nodes, Young's modulus,
2 3 200E9 76.92E9 0.010 0. 7800               shearing modulus, ext./int. diameter,
3 4 200E9 76.92E9 0.010 0. 7800               volumic mass )
4 5 200E9 76.92E9 0.010 0. 7800
5 6 200E9 76.92E9 0.010 0. 7800
6 7 200E9 76.92E9 0.010 0. 7800
999                                          (end section elements)
```

6	(number of added springs)
2 2 8 2 1000	(node1 dof1 node2 dof2 stiffness)
3 2 9 2 1000	
4 2 10 2 1000	
5 2 11 2 1000	
6 2 12 2 1000	
7 2 13 2 1000	
999	(end of springs section)
999	(end of masses section)
<end of data file>	

5. Phisical units

Here we use the european MKS system of units (meter, kilogram, second). Any coherent system of unit can be used with EF3D.

6. Presentation of results

The program calculates all the eigenfrequencies and eigenmodes of the structure and displays them in ascending frequencies. The used method (Householder tridiagonal reduction and QR algorithm) gives a good accuracy, even when some eigenvalues are multiple.

Shapes of modes are normalized to greatest translation (=unity). In the case where all translation are fixed (shaft in torsion), the shapes are normalized to greatest rotation (=unity).

The results menu is the following:

General Parameters:	1
Table of modes:	2
Shape of modes:	3
Draw a mode:	4
Orthogonality Test:	5
Exit:	6
Other Problem:	7

7. Program limits

7.1 To avoid exceeding the 64 ko limit for tables in Turbo-pascal, the program limits are:

Number of nodal points:	89
Number of free dof:	89
Number of elements:	89

In double precision, 8 x 89 x 89 = 63 368.

(In simple precision, 6 x 100 x 100 = 60 000, so the limit could be 100 instead of 89).

7.2 The mass matrix must not be singular. So if you neglect the mass of some elements (rho=0), be sure that every free dof for each node is connected to a local mass or inertia.

The stiffness matrix may be singular, so you can obtain rigid modes (rather linear combinations of classical translation and rotation modes) of free-free structures. In that case, the numerical method may lack accuracy with following cosequences:

a. either a warning that an eigenvalue is negative and set to zero,

b. or a positive frequency small with respect to other frequencies of soft modes.

8. Save and Restart

You can save to disk the mass and stiffness matrices, the eigenfrequencies and eigenvectors, the generalized masses and the shapes of modes.

Let us suppose that your data input file is called Pb007.dat.

During execution, if you ask to save the results, EF3D will create 4 binary files:

Pb007.gen	General Data
Pb007.stf	Siffness matrix
Pb007.mas	Mass matrix
Pb007.mod	Modal data

Note: on request, we can send you a small program in Pascal to read the saved data.

9. Copyright

Paris, May 9th, 2006.