LINEAR ANALYSIS AND MODAL DECOMPOSITION MADE EASY BY A PC-BASED COMPUTER PROGRAM COVERING DISCRETE, UNDAMPED MECHANICAL SYSTEMS

Version 2

by **Mart Mägi**

Mechanical Engineering Department, Chalmers University of Technology
S-412 96 GÖTEBORG, Sweden

Introduction

Within the area of mechanical engineering in a broader sense, covering various machines, mechanisms and structures, there are numerous design situations, where the vibrational characteristics of the object are extremely important.

In many of such situations, the vibration could be treated as linear, allowing superposition of solutions, which is the pre-condition for modal analysis.

Nowadays, the FEM technique is considered to be an almost universal tool for solving various kinds of vibrational problems. This technique was originally developed for solving linear partial differential equations, which in context of vibration describe small amplitude vibration in continuous systems.

The FEM concept has nowadays been extended to the non-linear range, and the finite elements must not necessarily be approximations of continuously distributed mass and elasticity. They could as well describe real machine components, e.g., engine blocks, flywheels, helical springs and shafts, which all have well known mass and stiffness characteristics.

A broad variety of FEM programs have over the years been developed for automated computation, many of them also for the PC environment.

In typical machine design situations the machine components are specific elements, which with a reasonable degree of accuracy could be described as rigid bodies or massless elastic elements. Furthermore, machines normally carry some nominal load and its parts move in some prescribed pattern, either steadily, intermittently or transiently.

At operation, dynamic add-on loads are generated in machines in motion. Theoretically, by solving a set of non-linear ordinary differential equations, such dynamic add-on loads and unconstrained element motion could be predicted reasonably well. Such procedures for derivation and solution of equations of motion are nowadays well formalized and also implemented for automated computation. The corresponding scientific discipline is often referred to as MultiBody System (MBS) analysis. By its nature, MBS analysis is generally non-linear.

However, in some applications related to steadily operating machines (or general discrete or lumped parameter systems), vibration within the linear range is of primary interest. In such systems the vibrational characteristics depend not only on component mass and stiffness properties, but also on steady-state loads or pre-loads, as, e.g., is the case at pendulums.

How should such problems be analyzed? Should sophisticated non-linear FEM programs or complex MBS programs be preferred?

None of the mentioned two groups of analytical tools can really compete in efficiency and versatility with a recently developed alternative hybrid method, which basically implies a linearization of the complete equations of motion in the vicinity of the equilibrium or steady-state loaded system configuration.

The hybrid concept was briefly introduced at the 1st Modal Testing & FEM Seminar in Århus in 1988 [1] by the present author and the complete theory is currently in the process of international publication [2]. The main feature of the new hybrid theory is the systematic partition of the total restoring action in a system, represented by a symmetric matrix K, into an elastic matrix E and a pre-load related matrix P, whereafter the equations for undamped motion read with the displacement vector d as the independent variable:

$$M\ddot{d} + Kd = M\ddot{d} + (E+P)d = 0 \quad \Rightarrow \quad K = E + P \qquad \text{....Eq. 1}$$

The combined or hybrid restoring action requests on one hand input of both of the element stiffnesses and pre-loads, but allows on the other hand an automatic evaluation of both "elastic" and "pendulum type" vibration at the same time, no matter if they contribute with the same order of magnitude, no matter which of them that happens to be quantitatively dominating, and no matter if one of them vanishes. The hybrid theory a priori also assumes that the equilibrium configuration and load distribution are known for the system (which is far from obvious for hyperstatic systems).

Model Components and their Characteristics

The hybrid theory so far is developed for undamped systems only, which is a reasonable approximation for lightly damped systems as far as eigenfrequencies and eigenmodes are studied. (Inclusion of damping elements will not complicate the derivation of governing differential equations of motion but it will make the computation of eigenvalues much more complex). The undamped system contains the following types of components:

* rigid bodies
* massless springs/connectors

A rigid body is characterized by its **mass** and **principal moments of inertia** with reference to the **center of gravity** of the body.

A massless spring or connector is related to two rigid bodies in a virtually unsymmetrical way: one of the two bodies is arbitrarily defined to be a **base body**, the other becomes then automatically a **connected body**. The spring is considered to be a part of the base body, and just one of the spring ends has significance in the analysis: the "upper" end, where the connected body is attatched to the spring. That point is by definition the **connecting point**. The **spring stiffness** is defined in relative terms as the linear relationships between the vectorial displacement of the connecting point relative the base body, and the vectorial change of load, or elastic load, at the connecting point, cf. Fig. 1 and Eqs 2 and 3. It should be observed that the initial load or **pre-load** is also a characteristic of the spring. It maintains its magnitude and orientation relative to the base body when the spring is arbitrarily deformed.

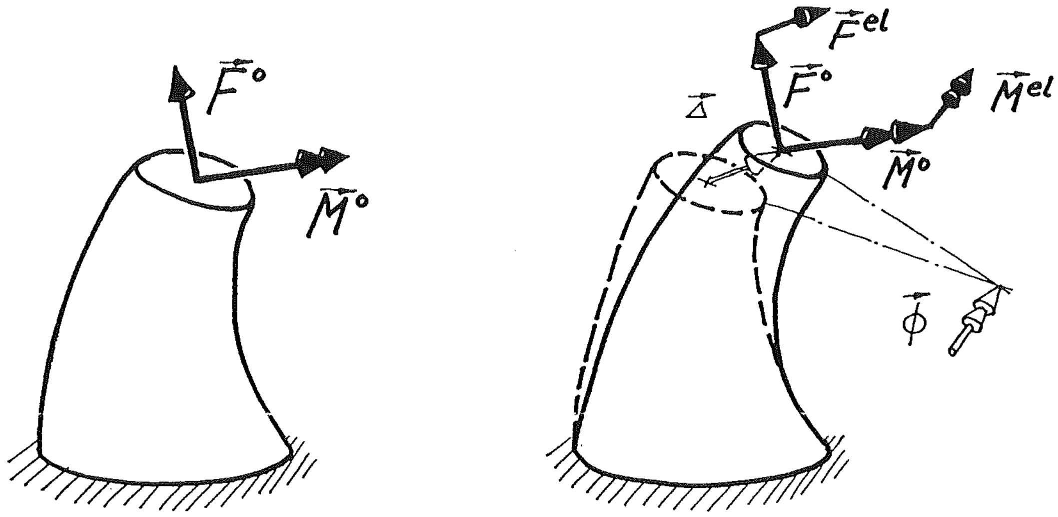

Fig. 1: Spring in initial and displaced/deformed configuration

$$\vec{F} = \vec{F}^o + \vec{F}^{el}; \qquad \vec{M} = \vec{M}^o + \vec{M}^{el}$$ ....Eq. 2

$$\begin{pmatrix} \vec{F}^{el} \\ \vec{M}^{el} \end{pmatrix} = \mathcal{S} \begin{pmatrix} \vec{\Delta} \\ \vec{\phi} \end{pmatrix}$$ ....Eq. 3

It is understood that for a sufficiently small spring displacement from its original equilibrium state (the initial deformation may, however, be large!), a linear relationship will be found between the incremental displacement and load change vectors, expressed as a six by six element symmetric **stiffness matrix** $\mathcal{S}$, if the vectors consist of three components each, as is generally the case. The stiffness matrix has the characteristics of a tensor.

Simple and reliable theoretical procedures for the evaluation of the complete stiffness matrix $\mathcal{S}$ are available only for a limited number of types of pre-loaded springs, e.g., **axially** pre-loaded strings, beams and helical springs. For other types and/or differnetly pre-loaded springs the stiffness matrix may be obtained either experimentally or analytically by using advanced finite element techniques.

Topology and Geometrical Description of the System

Systems to be considered, may contain an unlimited number of rigid bodies and massless springs. However, for programming reasons in the PC environment, the number of bodies is limited to 15 (due to maximum variable size in Turbo Pascal), and the number of springs is limited to 100 (to give a realistic size to the problem definition file, as that file in Turbo Pascal cannot be dimensioned dynamically).

Components are identified by ordinal numbers as subscripts. Typical body ordinals are i and j and a typical spring ordinal is k. Lowest body ordinal, zero, is reserved for the inertial frame or "earth". Highest body ordinal is denoted q (max. 15) and highest spring ordinal is denoted p (max. 100). A typical system may look like Fig. 2.

The geometrical quantification of the system needs to locate points (centers of gravity, connecting points and additional points, needed in the analysis of forced harmonic vibration, to define locuses of applied loads and evaluated responses). It also needs to specify the orientations of principal axes of inertia of rigid bodies and characteristic axes of elasticity of springs.

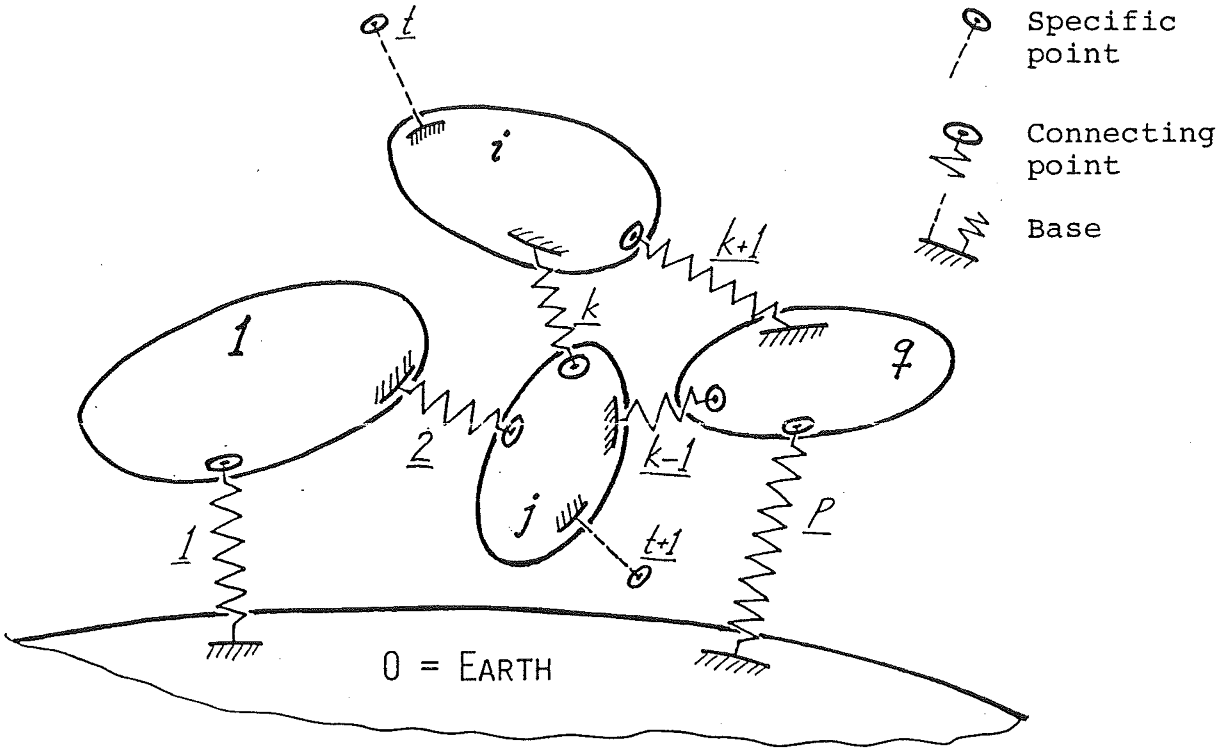

Fig. 2: Topology of a typical system

The quantitative description of the geometry of the system is facilitated by three different categories of Cartesian frames, all of them immobile in the inertial space:

* One arbitrarily located global frame of reference, (l,m,n).
* A number of body-specific local frames, (x,y,z), with their origins at the center of gravity and axes along principal axes of inertia of **each body in its equilibrium position.** Each of these frames has a physically well defined location and orientation.
* A number of spring-specific local frames, (a,b,c), with their origins at the connecting points of each spring and with the the orientation in a convenient direction for specification of the spring stiffness. These frames may be chosen arbitrarily within limits of practicality.

Each local frame needs the following information for its definition:

* A position vector defining its origin, $\vec{u}$, as an absolute or relative quantity, e.g., $\vec{u}_{o,k} = \vec{u}_{o,i} + \vec{u}_{i,k}$.
* The Euler angles - or an equivalent set of data - expressing a transformation *ROT* that describes rotation of frames, specifying the orientation of a coordinate system absolutely or relatively, e.g., $ROT_{o,k} = ROT_{i,k} \cdot ROT_{o,i}$.

Typical coordinate systems and position vectors are shown in Fig. 3.

In addition to the minimum required geometrical information as stated above, by means of position vectors only, a number of additional points of interest could be defined, where harmonic excitation could be applied or where the response could be studied. Each of such points, identified by an ordinal, typically t, must belong to a specific base body.

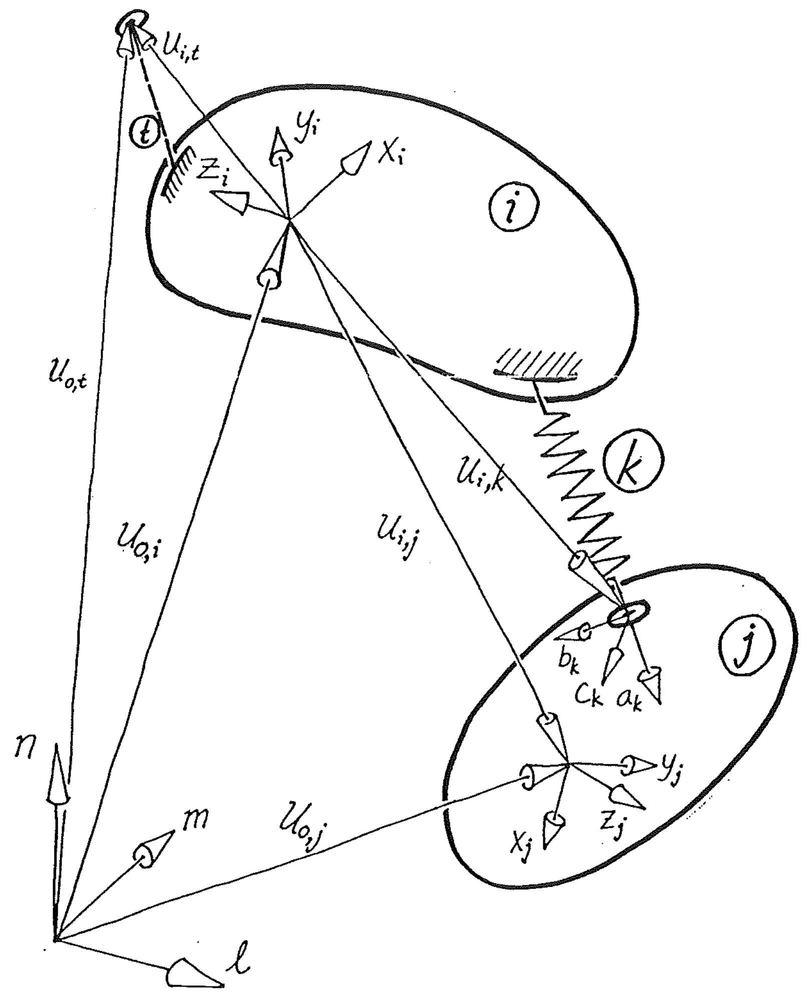

Fig. 3: Typical frames of reference and position vectors

Types of Springs or Connectors

In order to facilitate the creation of the element stiffness matrix S, a number of frequently occurring types of springs have been pre-defined in the program, for which the stiffness matrix could be generated automatically.

From the theoretical point of view, S so far is consistently defined for only such springs, which display symmetry about an "axial" direction. This means that stiffness matrices could be evaluated theoretically for **axially pre-loaded** springs, only. The axial orientation must then coinside with the "a" axis of the spring-specific local frame, the pre-load must be colinear with that axis, and it is defined as positive, when acting upon the base side of the connecting point.

For springs without axial symmetry the stiffness matrix must be given as an input quantity. Then the pre-load could have an arbitrary orientation (but must be consistent with the given stiffness data, which is not checked by the program).

At the generation of stiffness data, the program assumes as a basic case clamped conditions at both of the spring ends. However, if the spring is symmetrical about its axial midlength, the program is also able to generate automatically hinged end conditions at either one or both of the spring ends.

The program will automatically compose the stiffness matrix for the following types of axially symmetrical springs:

* helical springs (incl. lateral stiffness)
* beams (2nd order theory)
* fictitious springs:
  * groove ball bearings
  * cylindrical roller bearings
  * spherical ball and roller bearings
  * Hookean joints
  * guided plates
  * completely rigid connections

Some of the pre-programmed fictitious springs are depicted in Fig. 4.

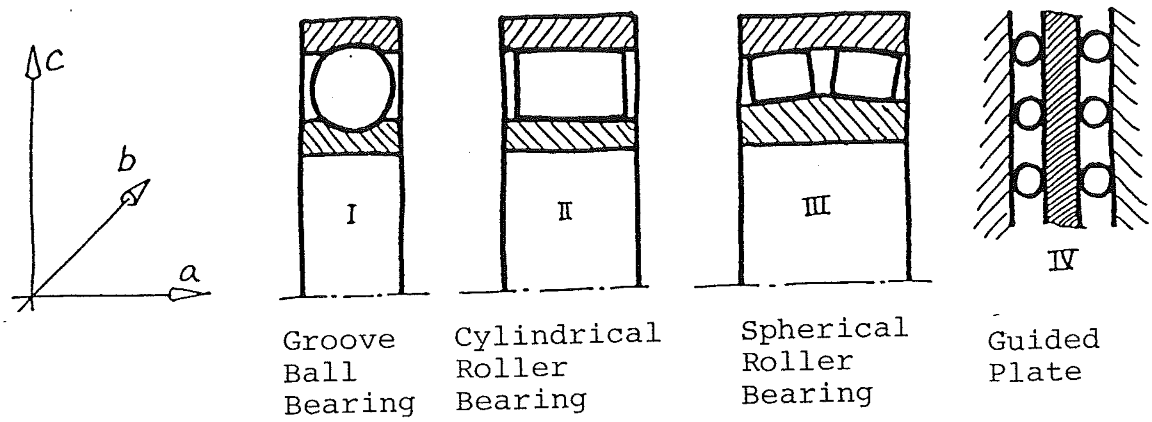

Fig. 4: Some pre-programmed types of fictitious springs

General Structure of the Program

The program is written in Turbo Pascal 4.0 by Jan Larsson and Lars Lindkvist as a thesis for their M.Sc. degree in Mechanical Engineering at Chalmers University of Technology, Göteborg, Sweden [3]. The work was supervised by the present author. The first version, incl. manual, was in Swedish. Later on the program was translated to English.

The program is menu driven and intended to be almost self-instructive if some main definitions as described above are familiar to the user. The main menu contains the following alternatives:

* **Case Description**, with input data editing and filing facilities
* **Computation**, with options free and harmonically forced vibration
* **Output** of input and computed data to Screen, Printer or ASCII file
* **Miscellaneous**, listing defined cases, directories, free memory, etc.
* **Quit**

Case editing facilities include the commands **Add**, **Modify**, **Copy** and **Delete** for all full or partial body or spring input data. Case filing facilities include the commands **Retrieve**, **Create**, **Modify** and **Delete**. Cases are identified by a max. 8 character name in accordance with the MS-DOS filename rules. Some **Current Case** is always active in the computer memory. Data is normally saved continuously upon acceptance to the **Current Case** files in binary file format, unless ASCII file output as a complement is ordered from the Output menu option.

It should be observed that input data for forced vibration (additional points, exciting harmonic loads) are requested from the **Computation** menu option, and **Add**, **Modify** and **Delete** commands apply.

The Computation option includes as a first step the creation of the system inertia and restoring matrices, M and K, respectively. The **Free Vibration** option comprises the calculation of all eigenfrequencies and eigenmodes, which - at the present stage - are evaluated by using the **MATHPAK 87** machine code procedure package [4], which is based upon the QR Triangularization method. Eigenvectors are scaled by division with the magnitude of the largest component, making that component to unity. The method evaluates also repeated eigenvalues (e.g., at symmetry), zero eigenvalues (at rigid body motion) and imaginary eigenvalues (at system instability).

The **Forced Vibration** option accepts excitation with one frequency at a time. However, the excitation load may have many simultaneous components, all in phase or antiphase. The steady-state response at all centers of gravity is evaluated by using MATHPAK 87 procedures, based upon Gaussian elimination. The response of any center of gravity is later easily transformed to any other point of the same rigid body.

The **Output** menu option allows viewing on screen, documentation on printer and saving as a file in the neutral ASCII format all input and computed data. Data could be selected in full, or by body or spring ordinals, and by eigenvalue ordinals, arranged in the order of ascending eigenfrequencies.

Hard- and Software Requirements

The program runs on all IBM compatibles under MS-DOS 2.0 or higher. It requires, however, a 80x87 math coprocessor to speed up heavy numerical tasks. 384 KB RAM is recommended. A hard disk is not an imperative, but it will facilitate storage of various cases. All Screen and Printer output is alphanumeric, why graphic adapters are not required.

Computation Time

Time required for solving a problem depends on two major processes:

* data input
* computation of solution

The time needed for data preparation and data input is reduced to a minimum by a high degree of interactivity and pre-programmed features (cf. automatic generation of stiffness matrices for the most common types of spring elements), as well as a reduction of geometric input to a minimum (just one node per spring). Still, this process seems to be the most time-consuming part of the total problem solving process, because most components tend to have fully unique data.

The computation time needed depends partly of the problem to be solved, partly on the computer that is used. Obviously, the number of rigid bodies in the system is a very significant factor (it determines the size of the matrices M and K), but the number of springs is almost not significant at all. Also, computation of forced vibration is considerably less time-consuming than computation of free vibration.

The computation time depends, of course, very much of the processor used, but less of the speed of the mass storage device (because disk access is minimal during computation, however higher at data input).

Some representative computation times are given in Tab. 1, where Case 1 is an 8088 processor running at 4.77 MHz and a floppy disk, whilst Case 2 is the same processor running at 8 MHz and a 68 ms hard disk.

Tab. 1: Some representative computation times (in seconds)

No. of	Forced vibration		Free vibration	
bodies	Case 1	Case 2	Case 1	Case 2
1	10	1	10	1
2	13	2	20	3
3	15	3	35	10
4	20	4	60	19
5	27	5	90	33
6	30	6	120	50
7	40	7	195	90
8	45	8	255	120
9	50	9	325	155
10	55	11	450	230
11	57	12	610	320
12	60	13	720	335
13	65	14	840	440
14	75	16	1050	555
15	80	20	1515	615

Numerical Accuracy

The program is written in standard Turbo Pascal double precision floating point format. This accuracy is definitely maintained through the composition of the matrices M and K. The computation of the free and forced vibration is formally also carried out in double precision format, but as the eigenvalue and Gaussian elimination algorithms are to some extent iterative and the iteration limits are not controllable by the program user, some loss of accuracy is expected in the last phase of the computation.

The accuracy has been checked by numerical experiments, which at the same time also demonstrate the error introduced by discretization or lumping. Vibration of a slender, circular beam of steel has been studied at three different support arrangements. The length of the beam is 1.10 m or 1.20 m and the diameter is 50 mm. The various support arrangements are shown in Fig. 5.

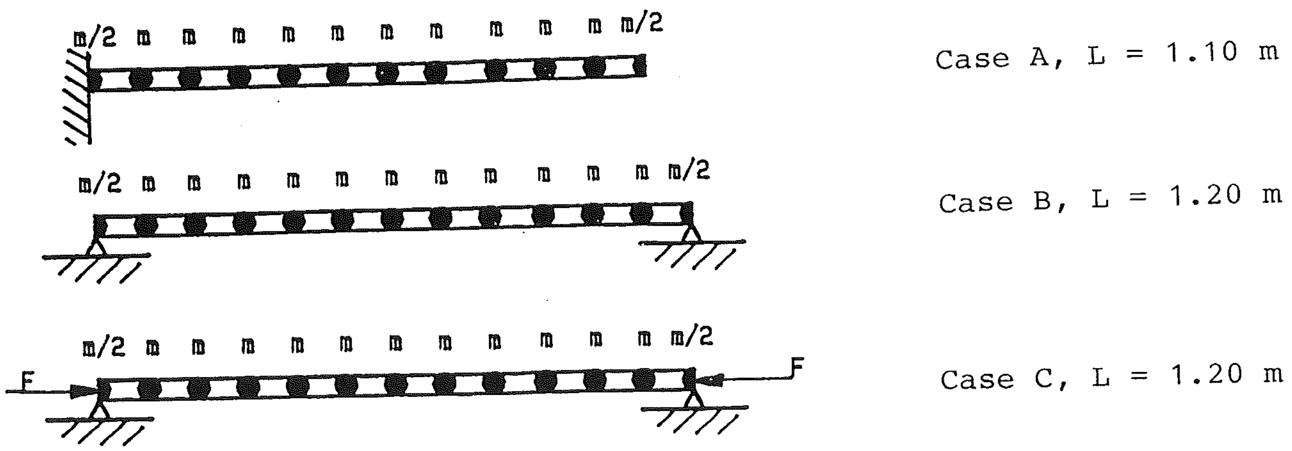

Fig. 5: End conditions of slender beam

The computed data display complete agreement between exact discertized and computed results for Case A in longitudinal and torsional vibration. The difference betweeen continuous and lumped systems is negligible at the lowest modes but of some significance at higher modes, cf. Tab. 2.

The computed data for Case B in bending vibration display reasonable to excellent agreement between continuous vs. exact discretized vs. computed results for all modes, cf. Tab 3.

The influence of axial load at the lowest mode is identical for the continuous and discretized model for the lowest modes within the prebuckling range, cf. Tab. 4 (buckling load = 441 kN).

Tab. 2: Case A, longitudinal and torsional vibration angular frequencies (1/s)

Mode	Longitudinal		Torsional	
No.	continuous	computed = lumped	continuous	computed = lumped
1	7409.51	7403.22	4595.18	4591.28
2	22228.53	22058.94	13785.55	13680.38
3	37047.55	36265.61	22975.91	22490.98
4	51866.58	49734.01	32166.28	30843.73
5	66686.60	62189.98	41356.65	38568.59
6	81504.62	73379.94	50547.02	45508.31
7	96323.64	83076.09	59737.39	51521.60
8	111142.66	91081.05	68927.75	56486.07
9	156236.24	97231.86	78118.12	60300.64
10	174616.98	101403.32	87308.49	62887.67
11	192999.71	103510.49	96498.85	64194.49

Tab. 3: Case B, bending vibration angular frequencies (1/s)

Mode	Bending		
No.	continuous	lumped	computed
1	444.54	444.54	444.54
2	1778.15	1778.06	1778.06
3	4000.85	3999.63	3999.63
4	7112.62	7104.98	7104.98
5	11113.47	11080.17	11080.17
6	16003.39	15887.22	15887.21
7	21782.40	21434.05	21434.04
8	28450.47	27517.48	27517.44
9	36007.64	33732.02	33731.96
10	44453.87	39371.22	39371.14
11	53789.19	43436.44	43436.34

Tab. 4: Case C, bending vibration angular frequencies (1/s)

F	Bending	
kN	continuous	computed
100	390.08	390.08
200	328.80	328.80
300	251.71	251.71

The numerical values for exact discretized systems in Tabs 2 and 3 are computed according to [5].

Versatility

The versatility of the program is demonstrated best by some hands on examples. Its strength is the combined handling of elastic and pendulum types of vibration within the same algorithm.

References

1. **Mägi, M.:** The Significance of System Pre-load at Modal Analysis of Low-resonant Mechanical Systems; 1st Modal Testing & FEM Seminar, Århus, 1988

2. **Mägi, M.:** On the Significance of Pre-load State in Linearized Analysis of Vibration in Lumped Parameter Mechanical Systems; Mechanical Engineering Department, Chalmers University of Technology, Göteborg, Sweden, 1988 (to be published).

3. **Larsson, J. & Lindkvist, L.:** Konstruktion av användarvänligt persondatorprogram för beräkning av svängningar i diskreta flermassesystem; M.Sc. Thesis, Mechanical Engineering Department, Chalmers University of Technology, Göteborg, Sweden, 1988

4. **MATHPAK 87:** User's Guide and Reference Manual (Version 3.0); Precision Plus Software, Scarborough, Ontario, Canada, 1988

5. **McCallion, H.:** Vibration of Linear Mechanical Systems, Chap. 6; Longman Group Ltd, 1973