Tạp chí Cơ Học Journal of Mechanics, NCSR of Vietnam T. XV, 1993, No 2 (37 — 41)

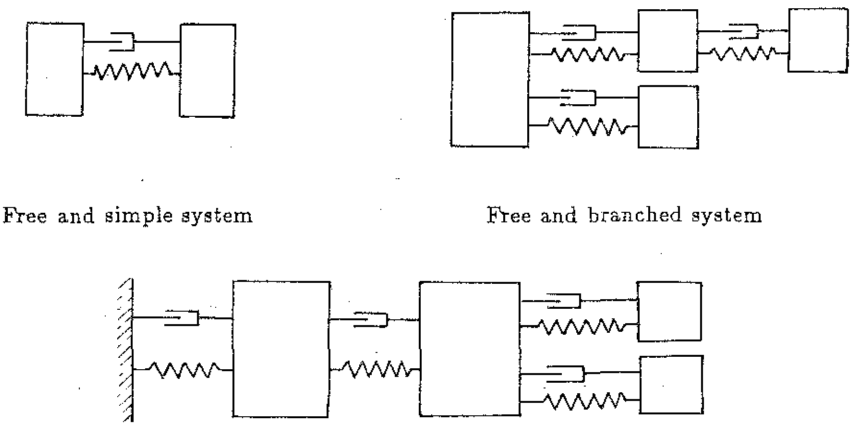
SOLUTION OF DISCRETE OSCILLATING SYSTEM ON PERSONAL COMPUTER

DINH VAN PHONG

Faculty of Applied Mechanics
Technical University of Hanoi

SUMMARY. The article is devoted a algorithm for deriving mass matrix, stiffness matrix and damping matrix for oscillating discrete system. The algorithm is common setting equation of motion. This technique enables solving different problems of oscillating system, especialy a problem of parameters optimization, by numerical methods. Comparison of different methods realized on personal computers was done.

Oscillating systems find many applications in a technical life. In this article some results of researching on computers are discussed. Most of techniques mentioned below were realized on 286X and 386X based personal computer and FORTRAN-77 is choosen as a programming language. Because of simplicity we will consider only discrete systems. It means that system could consist of elements of 3 types: mass, spring and damper and these features of system are concentrated. But the scheme is arbitrary: free or with a frame, simple or branched. Some of them are shown on the Fig. 1



Free and simple system

Free and branched system

Framed and branched system

Fig. 1. Examples of discrete oscillating system

Just now we will work in a different way than normal methods when solving "by hand". We will give one common algorithm of setting equations of motion for all types of oscillating system. Let's assume that forces will affect masses which are concentrated, rigid and unsprung. Springs and dampers are massies and dampers are viscous e. g. derived force is proportional to a speed. For such a system with n degrees of freedom we have a well-known system of equations of motion:

$$\underline{M} + \ddot{\underline{X}} + \underline{B}\,\dot{\underline{X}} + \underline{K}\,\underline{X} = f(t) \tag{1}$$

where

$\underline{M}$ is a mass matrix of the system

$\underline{K}$ is a stifness matrix

$\underline{B}$ is a damping matrix

$\underline{f}$ is a vector of acting force

$\underline{X}$ is a vector of displacements.

For deriving matrices $\underline{M}$, $\underline{K}$, $\underline{B}$ we use the absolute system of coordinates. In our case of discrete system with properties mentioned above the mass matrix is diagonal and on the diagonal there are masses of bodies

$$\underline{M} = \begin{bmatrix} m_1 & & & & O \\ & m_2 & & & \\ & & \ddots & & \\ & & & \ddots & \\ O & & & & m_n \end{bmatrix} \qquad \underline{K} = \begin{bmatrix} \sum C_i & \dots & -C_m & & O \\ \vdots & \ddots & \vdots & \ddots & \\ -C_m & \dots & \sum C_i & & \ddots \\ & \ddots & & \ddots & \\ & O & \ddots & & \sum C_i \end{bmatrix}$$

Deriving the stifness matrix is more complicated. Here we show directly its properties:

- A stiffness matrix is symmetric

- Elements of the stifness would be

$k_{ij} = 0 \quad i \neq j$ if i-th and j-th bodies are not connected

$k_{ij} = -C_m \quad i \neq j$ if i-th and j-th bodies are connected by m-th spring which has the stiffness C_m

On the diagonal:

$K_{ii} = \sum C_m$ where $\sum C_m$ is a sum of all stiffness at springs that are connected directly with i-th body.

This formule is general and it pays as well as for systems with a frame. It is useful when we want to have automatic setting equation of motion.

A similar formule is derived for the damping matrix. Instead of stiffness of a spring we should have damping. A form of a vector of forces is evident.

Now we can see that the stiffness matrix (as well as the damping matrix) is symetric and thin. So by using convenient numbering of bodies this matrix will have non-zero elements concentrated in a band along a diagonal, and the width of the band is:

$$w = \max|m - k| + 1 \tag{2}$$

where m and k are the numbers of arbitrary bodies which are directly connected.

This diagonal form of stiffness matrix is more convenient both for storage in a memory and time consuming on computers. We recommend using a algorithm of Cuthill-McKee [2], (both backwards and forewards) for renumbering masses.

The algorithm for setting equation of motion is a base point for a solution of considered system. We will show some results in following domains: eigenvalues and eigenvectors of the system, time-history of vibration of the system and a optimization of parameters

1) In order to find eigenvectors and eigenvalues of the system we consider a matrix equation:

$$\underline{M}\,\ddot{\underline{X}} + \underline{K}\,\underline{X} = \underline{O} \tag{3}$$

This equation leads to a equation:

$$\lambda \underline{M}\ \underline{C} = \underline{K}\ \underline{C} \tag{4}$$

where λ is a eigenvalue, $\underline{C}$ is eigenvector.

Because of a symetricity of matrix $\underline{K}$ a QL method for a tridiagonal matrix is suitable [3]. So in advance we must transform equation (4) into standard form by using Choleski's separation:

$$\underline{M} = \underline{L}\ \underline{L}^T \tag{5}$$

As the mass matrix $\underline{M}$ is diagonal, $\underline{L}$ is diagonal again and $\underline{L} = \underline{L}^T$, so we obtain a standard equation:

$$\left(\underline{L}^{-1}\ \underline{K}\ \underline{L}^{-1}\right)\ \left(\underline{L}\ \underline{C}\right) = \lambda(\underline{L}\ \underline{C}) \tag{6}$$

A riding matrix $\left(\underline{L}^{-1}\ \underline{K}\ \underline{L}^{-1}\right)$ is symetric again and we can use the mentioned QL method.

2) Time-History of vibration

When damping of the system is general we should transform equation (1) into the form:

$$-\begin{bmatrix} -\underline{K} & \underline{O} \\ \underline{O} & \underline{M} \end{bmatrix} \cdot \begin{bmatrix} \underline{X} \\ \dot{\underline{X}} \end{bmatrix} + \begin{bmatrix} \underline{B} & \underline{M} \\ \underline{M} & \underline{O} \end{bmatrix} \cdot \begin{bmatrix} \dot{\underline{X}} \\ \ddot{\underline{X}} \end{bmatrix} = \begin{bmatrix} \underline{f} \\ \underline{O} \end{bmatrix} \tag{7}$$

or:

$$-\underline{P}\ \underline{u} + \underline{N}\ \dot{\underline{u}} = \underline{g}$$

where

$$\underline{P} = \begin{bmatrix} -\underline{K} & \underline{O} \\ \underline{O} & \underline{M} \end{bmatrix}; \quad \underline{u} = \begin{bmatrix} \underline{X} \\ \dot{\underline{X}} \end{bmatrix}; \quad \underline{N} = \begin{bmatrix} \underline{B} & \underline{M} \\ \underline{M} & \underline{O} \end{bmatrix}; \quad \underline{g} = \begin{bmatrix} \underline{f} \\ \underline{O} \end{bmatrix}$$

The dimension of the problem is two-times greater but $\underline{F}$ and $\underline{N}$ are both symetric. A well-known method of Runge-Kutta could be used to solve equations [7].

In the cases where we could suppose the condition of proportional damping some methods of numerical intergration are better. For example when we have Rayleigh's damping:

$$\underline{B} = a_0\ \underline{M} + a_1\ \underline{K} \tag{8}$$

two methods of Newmark and Wilson are very elegant and convennient as a proportionality of $\underline{B}$ enables one useful transform of matrix $\underline{K}$.

Another way is using main coordinates. This technique leads to a system of n single differential equations which could be solved separately by Duhamell's integral.

Two points of this domain should be treated carefully on PC: a problem of a memory and time-step for numerical integration. A size of required memory depends on used language and compiler but perhaps the size of a data segment exceeds 64 Kbyte that will complicate a storage and handling data and influence time consuming. As a time-step we recommend to choose a value:

$$\Delta t < 1/10 f_{\max} \tag{9}$$

where $f_{\max}$ is the greatest frequency of a system

$$f = \sqrt{\lambda}/2\pi.$$

A greater step leads to accumulation of numerical errors and they can affect like a undesirable damping.

3) Optimization

Higher stage of solution is optimization of parameters (mass, stiffness and damping). Author of this article have tested a lot of methods of optimization on computers and the comparison of these methods was done in the frequency domain. We can set a following task:

Riding function

$$F = \sum_i w_i(\lambda_i - \lambda_{ib})^2 \longrightarrow \min$$

where λ_{ib} $i = 1, n$ is desirable set of eigenvalues,

λ_i are eigenvalues of the system.

A similar task could be set in the amplitude domain but the form of riding function should be choosen carefully. Note that we are speaking about numerical optimization, it means that at each stage of a optimization process the system is better if it gives a smaller value of the riding function. Also, we consider only parameter optimization.

The tests on computers showed that following methods are convennient for our task:

- Conjugate gradient method (Fletcher-Reevse's method)
- Fletcher-Powell's method
- Rosenbrock's method.

The best of them is conjugate gradient method for its precision and time-consuming.

Other methods: Box's algorithm or Glass-Copper's method were tested but the results were not so good.

One difficult can appear in a optimization process: Values of stiffness of springs are greater a lot than values of masses. This leads to a "zigzag" aproaching and doesn't allow to reach precise result. Some specials technique of optimization should be applied in order to gain better results [2].

A simple example:

We consider an example to illustrate a problem. A torsional system in Fig. 2 has 5 inertia bodies and 6 springs. The values of parameters are following:

$m_1 = 5kgm^2$	$m_2 = 3kgm^2$	$m_3 = 5kgm^2$
$m_4 = 6kgm^2$	$m_5 = 2kgm^2$	
$c_1 = 10^5\ Nm/rad$	$C_2 = 2 \cdot 10^5\ Nm/rad$	$c_3 = 4 \cdot 10^5\ Nm/rad$
$C_4 = 7 \cdot 10^5\ Nm/rad$	$C_5 = 2 \cdot 10^5\ Nm/rad$	$C_6 = 10^5\ Nm/rad$

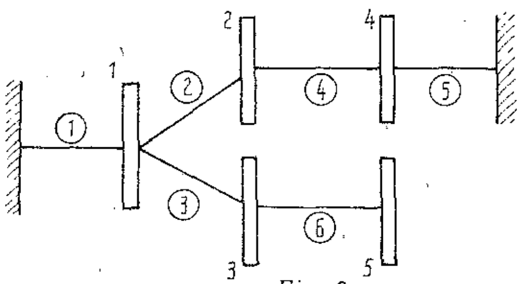


Fig. 2

The problem of eigenvalues is solved easily. The system has a spectrum:

104.2 201.1 269.3 449.9 643.2 rad/s

The most difficult task is a problem of optimization. Let's suppose that retained spectrum is not convennient and a constructor want to achieve better spectrum, say for example

90 250 300 450 650 rad/s.

Not all of parameters could be changed but only 1-st, 3-rd and 4-th bodies and 1-st, 5-th and 6-th springs.

In most cases the task of optimization require a lot of CPU time. For illustration we give in the Tab. 1 some value of solution on different 286X and 386X based PC. Rosenbrock's method is used and gives the result:

$$m_1 = 4.056 \qquad m_3 = 6.584 \qquad m_4 = 6.329\ kgm^2$$
$$C_1 = 1.063 \cdot 10^4 \qquad C_5 = 4.832 \cdot 10^5 \qquad C_6 = 1.34 \cdot 10^5\ Nm/rad$$

Tab. 1

PC		Math. Coprocessor	CPU time
Fujitkma 286,	8Mhz	none	11 min 32 sec
Universal Computer 286,	8Mhz	none	22 min 52 sec
Winn 286,	16Mhz	none	15 min 35 sec
Winn 286,	8Mhz	none	23 min 47 sec
Olivetti 286,	16Mhz	present	2 min 32 sec
Unitron 386,	32Mhz	none	5 min 52 sec

CONCLUSION

The algorithm for deriving mass, stiffness and damping matrix enables automatic solving the problem on personal computers. But experience showed that some difficulties could appear and we should be careful when using different methods because only some of them are convennient for our purpose.

This Publication is completed with financial support from the National Basic Research Program in Natural Sciences.

REFERENCES

1. Wright H., Gill D., Murray W. Practical Optimization, Academic Press, London, 1981.
2. Jennings A. Matrix computation for engineers and scientists. John Willey, 1977.
3. Garbow G., Boyle J., Dongara J., Moler C. Matrix eigensystem Routines-Eispack Guide Extension, Springer, New York, 1977.

Received June 29, 1992

GIẢI HỆ DAO ĐỘNG RỜI RẠC TRÊN MÁY VI TÍNH

Bài báo đưa ra một thuật toán để xây dựng các ma trận độ cứng, ma trận hệ số cản và ma trận khối lượng đối với hệ dao động rời rạc. Thuật toán này là tổng quát cho tất cả các dạng khác nhau của hệ dao động và tạo cơ sở cho việc lập phương trình dao động trực tiếp trên máy tính. Phương pháp này cho phép giải quyết nhiều bài toán khác nhau bằng các phương pháp số, đặc biệt là bài toán tối ưu tham số. Các phương pháp khác nhau cũng được so sánh khi được sử dụng trên máy tính.