PERIODICA POLYTECHNICA SER. TRANSP. ENG. VOL. 26, NO. 1-2, PP. 165-173 (1998)

NUMERICAL SIMULATION OF NONLINEAR VIBRATION SYSTEMS

György RICHLIK and Ferenc TAKÁCS

Research Group of the Dynamics of Machines and Vehicles
Hungarian Academy of Sciences
Department of Chassis and Lightweight Structures
Technical University Budapest
H-1521 Budapest, Hungary

Received: Oct. 30, 1998

Abstract

The object of the paper is to offer a brief survey of numerical simulations in nonlinear vibration systems. The development of computational methods provided an opportunity to create various kinds of software which are able to analyse the properties of the above mentioned systems. The possibilities of several kinds of software are introduced with the aid of a numerical example.

Keywords: nonlinear vibration systems, numerical simulation, finite element method.

1. Introduction

In research of vehicle system dynamics, investigation of parametral sensitivity of nonlinear systems is considered a very important question either in stability problems or in respect of chaotic behaviour. In studies of that purpose the computer aided numerical simulations are getting more and more important even if they consider relatively simple models. The development of computational methods and numerical mathematics makes easier to perform examinations that either could not be executed several years earlier or they would require a long and troublesome work.

Application of professional software, for example, Matlab, Maple, Phaser, Systus, etc. facilitates the work of researchers to a great extent and provides a chance to achieve good results for engineering practice within a short period, even if the problems cannot be treated analytically.

The subject of this paper is to survey the possibilities provided by the above mentioned program packages through analysis of a real vibrating system with particular respect to the less known Phaser and Systus software and to introduce numerical simulation with the aid of these softwares.

We premise that the first three types of software are mainly useful for investigation of minor, few degrees of freedom systems, but with the aid of Systus we can solve large systems of many degrees of freedom. For these large systems establishing of the differential equation system belonging to the physical model is difficult and the algorithms of the first three types of software need these equations. Unlike these purposeful algorithms, Systus is a finite element system where the starting point of the simulation is not a differential equation system to be solved but a geometrical – physical model which consists of elements and nodes, their geometrical and mechanical properties, initial and boundary conditions and loadings, according to the technique of the finite element method. The advantage of this technique appears in many degrees of freedom systems. Describing the complicated differential equation system is not required, it happens automatically during the process.

2. The Dynamical Model

Let us examine a damped, harmonically excited vibrating system of one degree of freedom in which the nonlinearity is derived from the approach of the spring force with a cubic parabola [1], [2]. Using the common literature terms, the equation of motion of the previously defined model can be written as

$$m\ddot{x} + k\dot{x} + \frac{1}{c}\left(x + \beta x^3\right) = F_0 \cos(\omega t), \tag{1}$$

which is a second order nonlinear ordinary differential equation. It is to be remarked that positive and negative values of β mean a hardening and softening spring characteristic, respectively. Divided by the mass the basic system can be written in the form

$$\ddot{x} + b\dot{x} + s\left(x + \beta x^3\right) = a\cos(\omega t). \tag{2}$$

We can observe that *Eq.* (2) contains the well known forced Duffing equation with a cubic stiffness [4], [5]. To perform numerical simulations, it will be convenient to rewrite *Eq.* (2) as an autonomous system [3],[5],[6],[10]

$$\begin{aligned} \dot{x}_1 &= x_2, \\ \dot{x}_2 &= -bx_2 - s\left(x_1 + \beta x_1^3\right) + a\cos(\omega x_3), \\ \dot{x}_3 &= 1. \end{aligned} \tag{3}$$

This first order differential equation system (3) is the only input form for Phaser. During the numerical experiments we do not deal with the physical meaning of the parameters in *Eq.* (1), neither with their values, nor with their dimensions.

3. Numerical Simulations with Matlab, Maple and Phaser

In the case of certain values of the parameters in *Eq.* (3) numerical simulations have been performed with the above mentioned types of software, and the results of examinations have been illustrated graphically in the usual way of the literature [4], [5], [6], [8]. In the auxiliary program made for using Matlab we considered the form of *Eq.* (2) and computed with the values of parameters according to *Fig. 1* and with zero initial conditions.

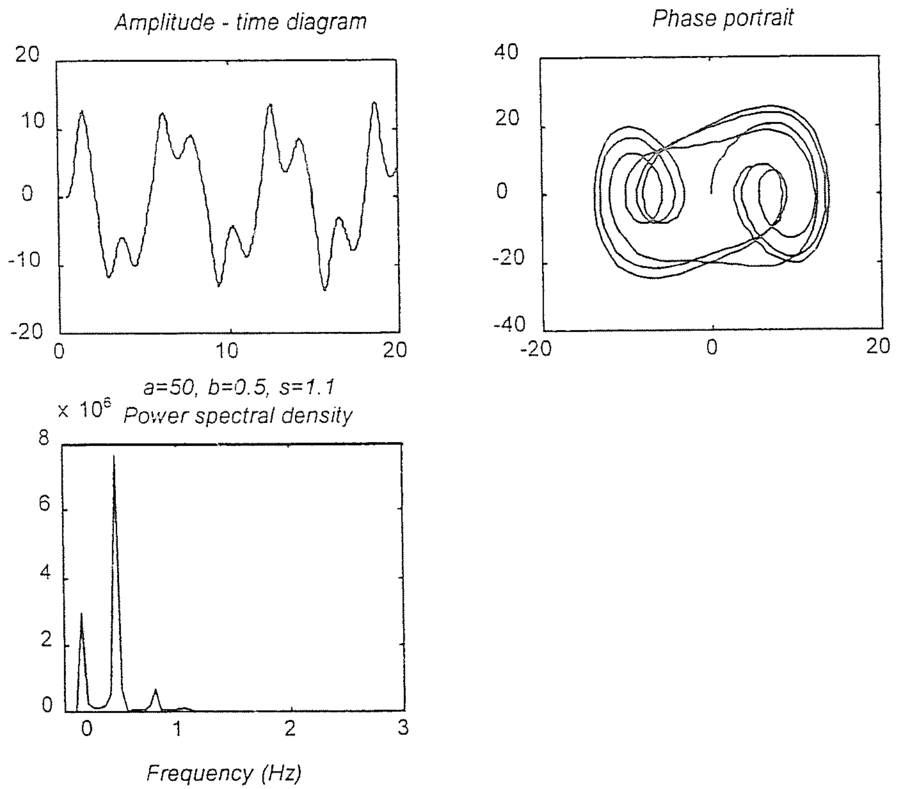

Fig. 1. Numerical simulation with Matlab ($\beta = 0.05$, $\omega = 1$)

Fig. 2 represents the results of the similar computation with Maple, where the phase-plane diagram and the curve of solution are shown.

The software Phaser, which can be obtained as an appendix of [10], is less spread in practice, but very useful to perform numerical simulations. This purposeful program system can be simply treated and need less memory. That software was used for the equation

$$m\ddot{x} + k\left(\dot{x} + a\dot{x}^3\right) + \frac{1}{c}\left(x + \beta x^3\right) = F_0 \cos(\omega t), \tag{4}$$

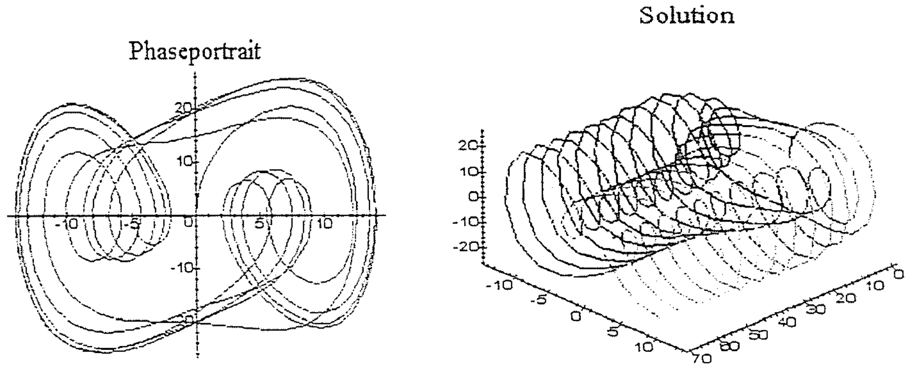

Fig. 2. Numerical experiments with Maple ($\beta = 0.05$, $\omega = 1$)

which is a little more complicated than *Eq.* (1). (4) is different from *Eq.* (1) in the form of damping force, which is also given by cubic parabola. We obtain from (4) the differential equation system

$$\begin{aligned} \dot{x}_1 &= x_2, \\ \dot{x}_2 &= -b\left(x_2 + \alpha x_2^3\right) - s\left(x_1 + \beta x_1^3\right) + a\cos(\omega x_3), \\ \dot{x}_3 &= 1 \end{aligned} \tag{5}$$

by appropriate transformation. *Fig. 3* represents the solution of this differential equation system (phase-plane diagram and amplitude–time function) obtained by numerical simulation Runge-Kutta method performed with Phaser.

One of the advantages of Phaser is the possibility of change of several parameters simultaneously. Therefore the initial examinations are to be performed practically with Phaser and later to continue the computation with the above introduced more advanced types of software.

4. Simulation with Systus Finite Element Program System

In the preceding chapters we presented several simulations of a single-degree-of-freedom nonlinear vibrating systems. These simulations were performed with the aid of different program systems, which based on similar numerical methods. The common property of these programs is that all of them solve differential equations or differential equation systems using somehow numerical methods. There are generally applied procedures for numerical methods not detailed in this paper.

Let us examine now the finite element simulation and its steps. Systus is a general purpose finite element method. The mechanical modules

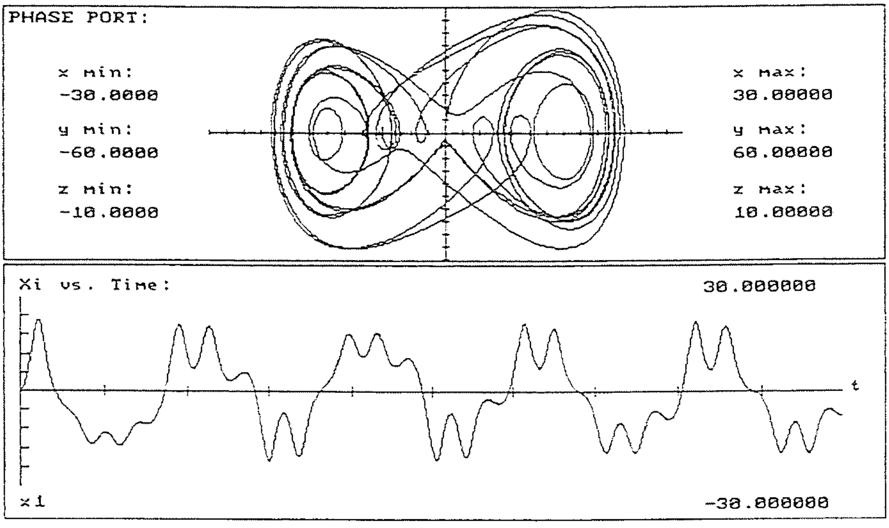

Fig. 3. Numerical simulation with Phaser $a = 200$, $b = 0.005$, $\alpha = 0.05$, $s = 1$, $\beta = 0.5$, $\omega = 1$

of these program systems were developed for static or dynamic, linear or nonlinear analysis of elastic or plastic structures [12]. Let us pay our attention to nonlinear transient calculations. In the standard case the nonlinear module of the system can be directly applied, no change in program is required, only the standard properties are to be given. Our task cannot be considered a standard one, therefore elements having the usual properties are not applicable.

The steps of creating a finite element model are: definition of geometry, material properties, boundary conditions and loadings. In the case of one-degree-of-freedom system geometry consists of two nodal points with their coordinates, a concentrated mass located in the second nodal point and a beam element between the two nodal points. The material properties are: value of the mass, elastic and damping data and cross-sectional area (regarding in Systus also as material property) of beam element. A material model found in Systus library is to be chosen, too, if the computation is not elastic. The boundary conditions are: all degrees of freedom of the first nodal point are fixed, only the displacement of the second nodal point remains in longitudinal direction. The loading is a longitudinal force acting in the second nodal point, which can be function of time in the case of transient calculation. That force has to be given in a table which can contain either connected values in pairs or parameters of the sinusoidal exciting function.

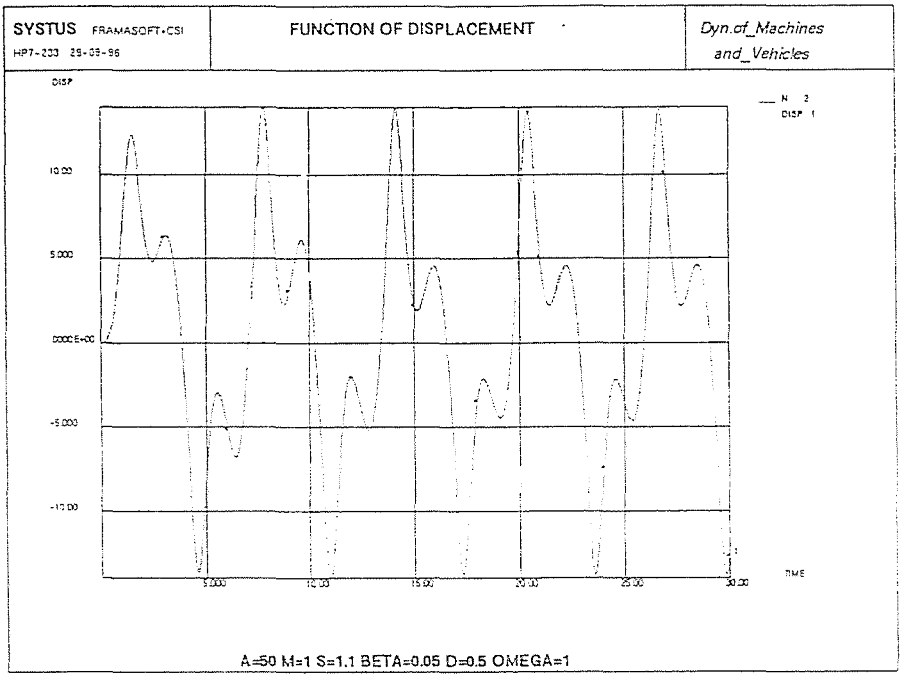

Fig. 4.

In the case of an elastic or other standard model of material the finite element model has been completed. Nevertheless in our task the spring characteristic is nonlinear and contains not only linear but cubic components, too. To consider this fact in the Systus system is possible in the following ways:

a) The elementary stiffness, mass and damping matrices can be defined by the user [13], [14]. To compute the values of each matrix, we can use not only the constants of material properties but the co-ordinates, displacements and velocities of the nodal points belonging to the element. In our case the values of the stiffness and the damping matrices belonging to the longitudinal direction contain terms proportional to third power of the displacement and velocity, respectively. To solve this problem we created a user-defined element formulated by a FORTRAN subroutine. This FORTRAN subroutine was linked with Systus shared element library and we have got a new element type 1802. Because we carry out a transient calculation, the above mentioned matrices will be updated in every time step from the formula of new element.

b) In the case of one-degree-of-freedom system (in our example) the following way is simpler and more efficient. The forces acting to the nodal

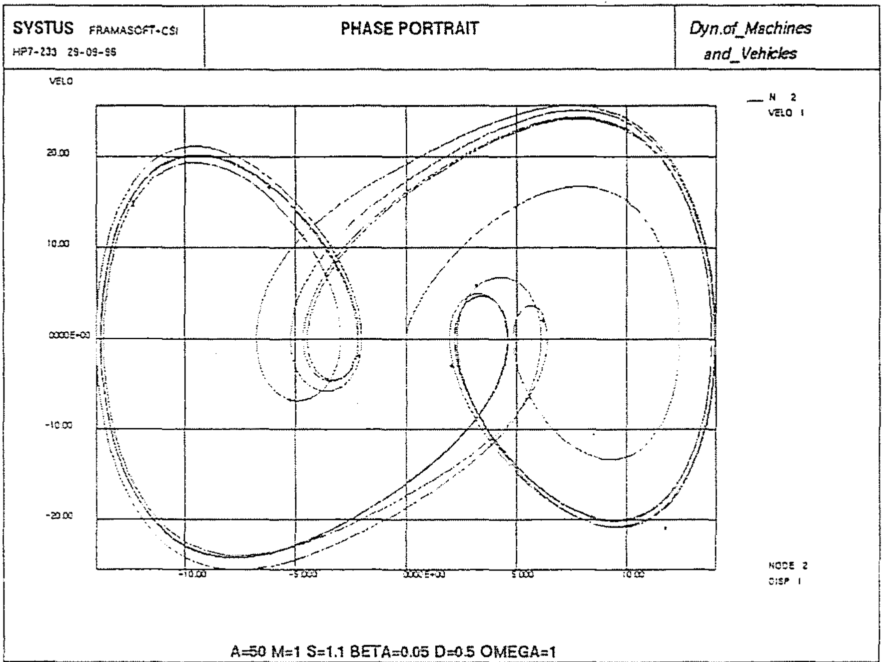

Fig. 5.

points also can be given by a FORTRAN subroutine, which provides access to the coordinates, displacements and velocities of the actual nodal point. As the displacement and velocity of the second nodal point is equal to the deformation of the spring and the relative velocity of the damper, respectively, two forces depending on displacement and velocity can be derived from the cubic terms.

These forces are added to the exciting force at the right side of the equation.

The code belonging to the dynamic model defined in Chapter 2 can be done by Appendix. The results of numerical simulation with Systus are shown in *Fig. 4* and *Fig. 5.*

5. Conclusion

Summarizing the above results, we can state that software, for example Phaser, Matlab and Maple are very useful to perform numerical simulations. Advantages of Phaser are the simplicity, the less memory and the possibility of change of several parameters simultaneously. Therefore the initial examinations are to be performed practically with Phaser and later to continue the computation with some more advanced software such as Matlab and Maple.

In the end we underline that the advantage of the finite element procedure (Systus) appears for simulation of many-degrees-of-freedom systems. In this case the establishment of the coupled differential equation system is troublesome and time consuming, to avoid difficulties we think fit to use finite element techniques.

6. Appendix

Definition of the finite element model and the nonlinear spring characteristic

```
definition
 EXCITATION: A*SIN(OMEGA*T)
option beam
geometry
nodes
 1 / 0. 0. 0.
 2 / 1. 0. 0.
elements
 1 / 1 2
material properties
 e 1 / E 1. NU 0.3 AX 1. IX IY IZ 1.
constraints
 n 1 / ux uy uz rx ry rz
 n 2 / uy uz rx ry rz
loadings
1 A=50 m=1 s=1.1 beta=0.05 d=0.5
 n 2 / fx 1. vari -1
masses
 n 2 / ax 1.
damping
 n 2 / ax 0.5
table
 1 / PROG
return

       SUBROUTINE USFONC (NAME,X,M,F,FP,FS)
       IMPLICIT INTEGER*4 (I-N)
       DIMENSION X(*),F(*),FP(*),FS(*)
       NAME=4HPROG
C COORDINATES X(1), X(2), X(3)
C TIME X(4)
```

```
C DISPLACEMENTS X(5), X(6), X(7)
      TIME=X(4)
      OMEGA=1.
      UX=X(5)
      A0=50.
      EXC1=A0*SIN(OMEGA*TIME)
      S=0.55
      EXC2=-OMEGA*OMEGA*S*UX*UX*UX
      F(1)=EXC1+EXC2
      RETURN
      END
```

References

[1] GEDEON, J. (1981): Mechanika IV/1. Lengéstan, Tankönyvkiadó, Budapest.

[2] MICHELBERGER, P. – HORVÁTH, S. (1981): Mechanika V. Válogatott fejezetek, Tankönyvkiadó, Budapest.

[3] CORDUNEANU, C. (1977): Principles of Differential and Integral Equations, Chelsea Publishing Company, The Bronx, New York.

[4] THOMSON, J. M. T. – STEWART, H. B. (1986): Nonlinear Dynamics and Chaos, John Wiley & Sons Ltd. Chichester.

[5] GUCKENHEIMER, J. – HOLMES, P. (1997): Nonlinear Oscillations, Dynamical Systems, and Bifurcations of Vector Fields, Springer-Verlag, New York.

[6] STROGATZ, S. H. (1994): Nonlinear Dynamics and Chaos, Addison Wesley Publishing Company, Reading.

[7] BRIAN, A. – BREINER, M. (1995): MATLAB for Engineers, Addison Wesley Publishing Company, Harlow.

[8] RANDALL, R. B. (1987): Frequency Analysis, Brüel & Kjaer, Naerum.

[9] MOLNÁRKA, GY. – GERGÓ, L. – WETTL, F. – HORVÁTH, A. – KALLÓS, G. (1996): A Maple V és alkalmazásai, Springer Hungarica Kiadó Kft., Budapest.

[10] KOCAK, H. (1986): Differential and Difference Equations through Computer Experiments, with Diskettes Containing PHASER: An Animator/Simulator for Dynamical Systems for IBM Personal Computers, Springer-Verlag, New York.

[11] GOVEKAR, E. – GRABEC, I. (1988): Chaotic Oscillations of a Constrained Resonant System, *Strojniski Vestnik – Mechanical Engineering Journal,* Vol. 34, No. 7–9, pp. E7–E12.

[12] BATHE, K. J. (1982): Finite Element Procedures in Engineering Analysis, Princeton Hall.

[13] Systus Theoretical Manual: Chapter 2. Element Library. Framatome, Paris, 1989.

[14] SYSTUS User's Manual: Chapter VIII. 8. Special Elements, Framatome Information Systems, 1996. pp. VIII. 190–VIII. 200.