International Journal of Engineering and Mathematical Sciences
July – December 2016, Volume 10, Issue – 1, pp.36 – 44.
ISSN (Print) – 2319 - 4537, (Online) – 2319 - 4545.

IJEMS

Numerical Solution of Dynamic Vibration Equation, $\ddot{x} = \operatorname{sech} x$

H. A. Ochieng[1], T. J. O. Aminer[2] and N. Omolo Ongati[3]
[1,2,3]School of Mathematics and Actuarial Science,
Jaramogi Oginga Odinga University of Science and Technology,
P.O. Box 210-40601, Bondo, Kenya.
[1] helloe.ochieng@gmail.com.

Abstract

In this paper, we examine conservative autonomous dynamic vibration equation, f(x) = sech x which is time vibration of the displacement of a structure due to the internal forces, with no damping or external forcing. Numerical results using New mark method are tabulated and then represented graphically. Further the stability of the algorithms employed is also discussed.

Keywords: Numerical solution, dynamic vibration equation, stability, conservative, autonomous systems, restoring force.

Introduction

Studying the variation of some physical quantities on other physical quantities would lead to differential equations, [6]. Many engineering subjects, such as mechanical vibration or structural dynamics, heat transfer, or theory of electric circuits, are founded on the theory of differential equations. [4]

Systems described by differential equations are complex, or so large, that a purely analytical solution to the equations is not yielded, thus the use of computer simulations and numerical methods to produce results. The biggest numbers of structures are in a continuous state of dynamic motion because of random loading such as wind, vibration equipment, or human loads. Thus a lot of consideration has been given in the design of certain facilities or structures which need to resist sudden but strong vibrations. [2] and [5].

In this study, we are mainly interested in examining the time vibration of the displacement of a structure due to the internal forces, with no damping or external forcing.

Non-linear conservative autonomous second order system

We consider the second order systems of non-linear conservative autonomous equations generally given by

$$\kappa\ddot{x} = \xi\dot{x} - \rho f(x) \qquad (1)$$

with some initial conditions $x(0) = \omega_0$ and $\dot{x}(0) = \omega_1$, where κ, ξ and ρ are real positive numbers and $\xi\dot{x}$ is the damping force.

As an explanation to the new terms, we note that:

a) The system is conservative because dynamic systems obey the principal of conservation of energy which asserts that the sum of kinetic and potential energies is constant in a conservative force field.
b) The system is autonomous because we are concerned with a system of ordinary differential equations which does not explicitly but implicitly contain the independent variable t (time).
c) The restoring force, $f(x)$, defines the position of the moving object from its equilibrium point.
d) There is no damping force i.e. no resisting medium, $-\xi\dot{x} = 0$.

Thus substituting $\kappa = 1, -\xi\dot{x} = 0 \; and \; \rho = 1$, we have

$$\ddot{x} = -f(x) \tag{2}$$

and for our study let us consider

$$f(x) = \operatorname{sech} x \tag{3}$$

leading to the dynamic vibration equation

$$\ddot{x} = -f(x), x(0) = \alpha_0, \; and \; \dot{x}(0) = \alpha_1 \; at \; , t = 0. \tag{4}$$

From equation (2), we can derive an autonomous system, in the form of

$$\frac{dx}{dt} = y, \text{ and } \frac{dy}{dt} = -f(x) \tag{5}$$

Where the right hand side does not involve t explicitly but implicitly through the fact that x and y themselves depend on t and thus being self governing.

The above reduction of the second order non-linear to equivalent first order nonlinear is by introducing a new independent variable

$$y = \frac{dx}{dt}$$

and since

$$\frac{dy}{dt} = \frac{d^2x}{dt^2}$$

the function x, y satisfy the equivalent first-order system

$$\frac{dx}{dt} = y,$$

$$\frac{dy}{dt} = -f(x)$$

Where equivalent means that each solution to the first order system uniquely corresponds to a solution to the second order equation and vice versa, [2], Specifically, equation (2) is equivalent to the autonomous system, equation (6):

$$\frac{dx}{dt} = y, \frac{dy}{dt} = -f(x), x(0) = \alpha_0 \; at \; t = t_0 \tag{6}$$

From equation (6), $\frac{dy}{dx} = -\frac{f(x)}{y}$ (7)

$$\Rightarrow \int_{\alpha_0}^{y} y dy = -\int_{\alpha_0}^{x} f(x)dx$$

or

$$\frac{y^2}{2} - \frac{\alpha_0^2}{2} = -\int_{\alpha_0}^{0} f(x)dx + \int_{0}^{x} f(x)dx$$

or

$$\frac{y^2}{2} + \int_0^x f(x)dx = \frac{\alpha_0^2}{2} + \int_0^{\alpha_0} f(x)dx \quad (8)$$

i.e. $KE + PE = C$ where $KE = \frac{y^2}{2}$ is the kinetic energy of the dynamic system,

$PE = \int_0^x f(x)dx$ is the potential energy of the dynamic system, equation (2) while

$C = \frac{\alpha_0^2}{2} + \int_0^{\alpha_0} f(x)dx$ is the constant (energy level). So equation (8) expresses the law of conservation of energy. For the physical interpretation of the study, the non-linear restoring force, $f(x)$ above, gives rise to special cases of non-linear spring motion according to its behavior. Equation (2) is said to represent motion of:

a) a hardening spring if free vibration frequencies increase along with the amplitude i.e. the magnitude of the restoring force, $f(x)$ acting on the mass, does increase more rapidly than that of a linear spring,
b) a softening spring if free vibration frequencies decrease along with the amplitude i.e. the magnitude of the restoring force, $f(x)$ acting on the mass, does increase less rapidly than that of a linear spring, [4].

The above mentioned two special cases of equation (2) form the central subject of discussion. Considering function (3) and another situation where the restoring force is

$$f(x) = x^2 + x - 2 \quad (9)$$

we have two cases

a) $\frac{d^2x}{dt^2} + x^2 + x - 2 = 0$

b) $\frac{d^2x}{dt^2} + \operatorname{sech} x = 0$

The behavior of the graphs shows clearly the idea of the hardening spring and softening spring for the two non-linear restoring forces given. Considering the magnitude of the non-linear restoring force, $f(x) = x^2 + x - 2$, in case (a), since it does increase more rapidly than that of a linear spring i.e. $f(x) = x$, it represents a hardening spring. On the other hand, considering the magnitude of the non-linear restoring force $f(x) = sech\, x$ in case (b), since it does increase less rapidly than that of a linear spring, i.e. $f(x) = x$ it represents a softening spring.

Numerical Solution of equation (2)

Numerical methods available for application in determining solution of Nonlinear systems for the implicit non-linear dynamic system (4) are numerous and include Newmark, Wilson-theta, Hilbert-

Hunges-Taylor and Houbolt. In determining the solution for the implicit non-linear dynamic system (4), the implicit dynamic method we will apply is Newmark algorithm method since it is unconditionally stable and able to converge rapidly to a meaningful solution.

Newmark's Algorithm

Considering the equation (2) given by $\ddot{x} + f(x) = 0$, Newmark's originally proposed method applied to it is of the form

$$x_{n+1} = x_n + h\dot{x} + \left(\frac{1}{2} - \beta\right) h^2 \ddot{x}_n + \beta h^2 \ddot{x}_{n+1} \tag{10}$$

$$\dot{x}_{n+1} = \dot{x}_n + (1-\gamma) h \ddot{x}_n + \gamma h \ddot{x}_{n+1} \tag{11}$$

$$\ddot{x}_n + f(x_n) = 0 \quad or \quad \ddot{x}_n + f_n = 0 \tag{12}$$

Where h is the time step and β and γ are the two Newmark parameters.[6] and [8]

write $n + 1$ for n into (12) to get

$$\ddot{}x_{n+1} + f_{n+1} = 0 \tag{13}$$

Eliminating $\ddot{x}_{n+1}$ from (12) and (10) we have

$$x_{n+1} + h^2 \beta f_{n+1} = x_n + h\dot{x}_n + h^2 \left(\frac{1}{2} - \beta\right) \ddot{x}_n \tag{14}$$

Similarly, when we eliminate $\ddot{x}_{n+1}$ from (12) and (11) we have

$$\dot{x}_{n+1} + \gamma h f_{n+1} = \dot{x}_n + (1-\gamma) h \ddot{x}_n \tag{15}$$

Eliminate $\dot{x}_n$ from (14) and (15) i.e. (14) $-$ h(15), we have

$$x_{n+1} + (\beta h^2 - \gamma h^2) f_{n+1} = h\dot{x}_{n+1} + h^2 \left(\gamma - \beta - \frac{1}{2}\right) \ddot{x}_n + x_n \tag{16}$$

Write $n + 1$ for n in (14)

$$x_{n+2} + c f_{n+2} = x_{n+1} + h\dot{x}_{n+1} + b\ddot{x}_{n+1} \tag{17}$$

Where $a = (1-\gamma)h, b = h^2 \left(\frac{1}{2} - \beta\right)$ $and\ c = h^2 \beta$

Eliminate $\dot{x}_{n+1}$ from (17) and (15) to get

$$x_{n+2} + c f_{n+2} = x_{n+1} - \gamma h^2 f_{n+1} + h\dot{x}_n + ah\ddot{x}_n + b\ddot{x}_{n+1} \tag{18}$$

using equation (14) to eliminate $\dot{x}_n$ in 18)

$$x_{n+2} + c f_{n+2} = 2x_{n+1} + c f_{n+1} - x_n - b\ddot{x}_n + ah\ddot{x}_n - \gamma h^2 f_{n+1} + b\ddot{x}_{n+1} \tag{19}$$

and (13) to substitute $\ddot{x}_n$ with $-f(x_n)$ in (19) we get

$$x_{n+2} + c f_{n+2} = 2x_{n+1} - (b + \gamma h^2) f_{n+1} + c f_{n+1} + (b - ah) f_n - x_n,$$

or

$$x_{n+2} + h^2 \beta f_{n+2} = 2x_{n+1} + \left(2\beta - \gamma - \frac{1}{2}\right) h^2 f_{n+1} - x_n + \left(\gamma - \beta - \frac{1}{2}\right) h^2 f_n \tag{20}$$

$x_{n+2} + h^2 \beta f_{n+2} = Q$ where Q is known value.

The scheme (20) in the displacement only is a two-step (three-time-level) scheme.

For $\beta = 0$ the scheme (20) become explicit, i.e.

$$x_{n+2} = 2x_{n+1} - \left(\gamma + \tfrac{1}{2}\right) h^2 f_{n+1} - x_n + \left(\gamma - \tfrac{1}{2}\right) h^2 f_n \qquad (21)$$

The maximum accuracy for equation (2) is achieved when $\beta_0 = \frac{1}{6}$ i.e. $\beta = \frac{1}{12}$ and $\beta_1 = \frac{1}{2}$ i.e. $\gamma = \frac{1}{2}$. This is the trapezium rule for the linear case.

substitute x_{n+1}, $\dot{x}_{n+1}$ in our test equation (2) and we have

$$\ddot{x}_{n+1} + f\left(x_n + h\dot{x}_n + h^2\left(\tfrac{1}{2} - \beta\right)\ddot{x}_n + h^2\beta\ddot{x}_{n+1}\right) = 0 \text{ or}$$

$$\ddot{x}_{n+1} + f(a_0 + h^2\beta\ddot{x}_{n+1}) = 0 \qquad (22)$$

Where $a_0 = x_n + h\dot{x}_n + h^2\left(\frac{1}{2} - \beta\right)\ddot{x}_n$ is known value. (22) is non-linear in $\ddot{x}_{n+1}$ (or implicit) provided $\beta \neq 0$, and requires a non-linear iterative method such as Newton-Raphson for solution.

Numerical Results of the Equation

Using the Newmark Algorithm developed above, (0.0.20), we obtained the required numerical results as follows:

Considering the scheme (20) i.e.

$$x_{n+2} + h^2\beta \operatorname{sech} x_{n+2} = Q$$

Where $Q = 2x_{n+1} + \left(2\beta - \gamma - \frac{1}{2}\right) h^2 \operatorname{sech} x_{n+1} - x_n + \left(\gamma - \beta - \frac{1}{2}\right) h^2 \operatorname{sech} x_n$

Which is the displacement only and two-step (three-time-level) scheme.

Using R i386 3.3.1 computer programming with Newton-Raphson's iteration,

$$x_{n+2} = x_{n+1} - \frac{F(x_{n+1})}{F'x_{n+1}}$$

from scheme (20)

$$Fx_{n+1} = x_{n+1} + h^2\beta \operatorname{sech} x_{n+1} - Q$$

Thus $F'x_{n+1} = 1 + h^2\beta(-\operatorname{sech} x_{n+1}\, tanhx_{n+1})$

$$F'^{x_{n+1}} = 1 - h^2\beta \operatorname{sech} x_{n+1}\, tanhx_{n+1} \qquad (23)$$

therefore

$$x_{n+2} = x_{n+1} - \frac{x_{n+1} + h^2\beta \operatorname{sech} x_{n+1} - Q}{1 - h^2\beta \operatorname{sech} x_{n+1} tanhx_{n+1}} \qquad (24)$$

where $\operatorname{sech} x_{n+1}$ is expressed as $\frac{1}{coshx_{n+1}}$ to be used in the computer programmer as follows:

Variables $x0 = x_n$, $x01 = \dot{x}_0$,$x1 = x_{(n+1)}$,

$x2 = x_{(n+2)}, B = Beta(\beta), r = alpha(\alpha)$, $a_0 = tanhx_n$,$b_0 = coshx_n$, $b_1 = coshx_{n+1}$, $w_0 = sechx_n$, $w_1 = sechx_{n+1}$, R Script: Newmark Algorithm

$i < -list$ (0, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1.0, 1.1, 1.2, 1.3, 1.4, 1.5, 1.6, 1.7, 1.8, 1.9, 2.0, 2.1, 2.2, 2.3, 2.4, 2.5, 2.6, 2.7, 2.8, 2.9, 3.0)

j < - *list* (0.2, 0.0833, 0.05) #*List of Beta*

s < -*c( )*

set < -*c( )*

z < -*c( )*

#Given that; $x0 \ < -0.2$; $x01 \ < -01{:}5$

for (B in j)

#print(B)

#B refers to Beta (β) and r refers to Alpha (α)

If (B = = 0.2)[r < -0.1; k < -"$Beta_{0.2}$ $alpha_{0.1}$"]

If (B = = 0.05)[r < -0.4; k < -"$Beta_{0.05}$ $alpha_{0.4}$"]

if(B = = 0.0833)[r <- 0.1; k < -"$Beta_{0.0833}$ $alpha_{0.5}$"]

for (*h in i*)

#print(*h*)

$a0 = -\tanh(x0)\,; b0 < -\cosh\,(x0)$

$x1 < -x0 + (h * x01)$

$a1 < -\tanh(x1)\,; b1 < -\cosh(x1)$

$w0 < -\left({}^{1}/_{b0}\right)$#sech $(x0)$

$w1 < -\left({}^{1}/_{b1}\right)$#sech $(x1)$

$$Q = -(2 * x1) + \Big(\big((2 * B) - r - 0.5\big) * h * h * w1\Big) - (x0 - \big((r - B - 0.5) * h * h * w0\big)$$

$$x2 = -x1 - \frac{x1 + (h * h * B * w1) - Q}{(1 - (2 * h * h * B * w1 * a1))}$$

#print $(x2)$

$s \ < -c(s, h)$

$set \ < -c(set, k)$

$z \ < -c(z, x2)$

$t \ < -$ as. data.frame(as. numeric(s))

$X \ < -$ as.data.frame (as. numeric (z))

$set < -$ as.data.frame(set)

library (reshape)

data $< -$cbind(t,set,X)

data$< -$ rename(data, c("as.numeric(s)" ="t","set"= "set","as.numeric(z)"="X"))

library(ggplot2) ggplot(data = data; aes(x = t, y = X, group = set, shape =
set, colour = set))+geomline(size = 1)+labs(shape = "Key")+geompoint(size =

2) + labs(list(title = "", x = "time", y = "x"))

The following conditions were taken into account when compiling the results:

i. It is clearly stated just before equation (22) that the maximum accuracy is achieved when $\beta = \frac{1}{12}$ and $\gamma = \frac{1}{2}$ and so our choice of the parameters was influenced by the given parameters, according to Zienkiewicz [9] and Wood [8];
ii. The stability of the numerical schemes is governed by small step size, h, according to Hughes, Caughey and Liu [3]. Given that $x_1 = x_0 + h\dot{x}_0$ let $x_0 = 0.2$ and $\dot{x}_0 = 1.5$ thus $x_1 = 0.2 + 0.1(1.5) = 0.35$.

Leading to the following results:

t	X when $\beta = 0.2$ and $\gamma = 0.1$	X when $\beta = \frac{1}{12}$ and $\gamma = \frac{1}{2}$	X when $\beta = 0.05$ and $\gamma = 0.4$
0.0	0.20000	0.20000	0.20000
0.1	0.49044	0.49043	0.49055
0.2	0.76316	0.76308	0.76418
0.3	1.02065	1.02051	1.0249
0.4	1.26543	1.26546	1.27668
0.5	1.49934	1.50034	1.52383
0.6	1.72348	1.72697	1.76945
0.7	1.93829	1.94654	2.01582
0.8	2.14372	2.15970	2.26436
0.9	2.33942	2.36668	2.51574
1.0	2.52494	2.56742	2.77011
1.1	2.69982	2.76168	3.02724
1.2	2.86363	2.94914	3.28665
1.3	3.01605	3.12942	3.54773
1.4	3.15685	3.30215	3.80984
1.5	3.28586	3.46697	4.07231
1.6	3.40301	3.62357	4.33453
1.7	3.50827	3.77166	4.59592
1.8	3.60163	3.91099	4.85596
1.9	3.68315	4.04138	5.11422
2.0	3.75287	4.16263	5.37029
2.1	3.81084	4.27463	5.62386
2.2	3.85713	4.37725	5.87466
2.3	3.89179	4.47040	6.12246
2.4	3.91487	4.55402	6.36708
2.5	3.92640	4.62806	6.60839
2.6	3.92642	4.69247	6.84629
2.7	3.91495	4.74722	7.08069
2.8	3.89200	4.79230	7.31154
2.9	3.85757	4.82769	7.53881
3.0	3.81167	4.85337	7.76247

Stability of the Numerical Algorithms Employed

From the results of the two-step (three-time level) scheme tabulated above and its corresponding graphical representation, it is clearly evident that for zero damping or no damping dynamic equation, the Newmark method is conditionally stable when the parameters chosen, for instance $\beta = 0.05$ and $\gamma = 0.4$ are within the neighbourhood of the parameters associated with maximum accuracy i.e. $\beta = \frac{1}{12}$ and $\gamma = \frac{1}{2}$. As we move away from maximum accuracy parameters, for instance taking $\beta = 0.2$ and $\gamma = 0.1$ the method is no longer conditionally stable.

Conclusion

In this study, we have looked at Newmark numerical scheme that can be used to solve the implicit nonlinear dynamic vibration equations. We used a displacement only, two-step (three-time-level) schemes with a R studio computer programmer which is very effective and fast enough in producing results. The equation has been solved using Newton-Raphson iteration method that converges fast to a meaningful solution. The results are tabulated and given graphically through an efficient and accurate graphical package known as the Rplot.

From the stability of the numerical scheme, a small step size is needed, with maximum accuracy achieved when the Newmark parameters, β and γ and are $\frac{1}{12}$ *and* $\frac{1}{2}$ respectively. The results of our study indicate that Newmark algorithm exhibit stable cases for the solution of the softening spring, equation (4) when parameters chosen are very close to the maximum accuracy parameters, otherwise unstable when parameters chosen are not close to the maximum accuracy parameters.

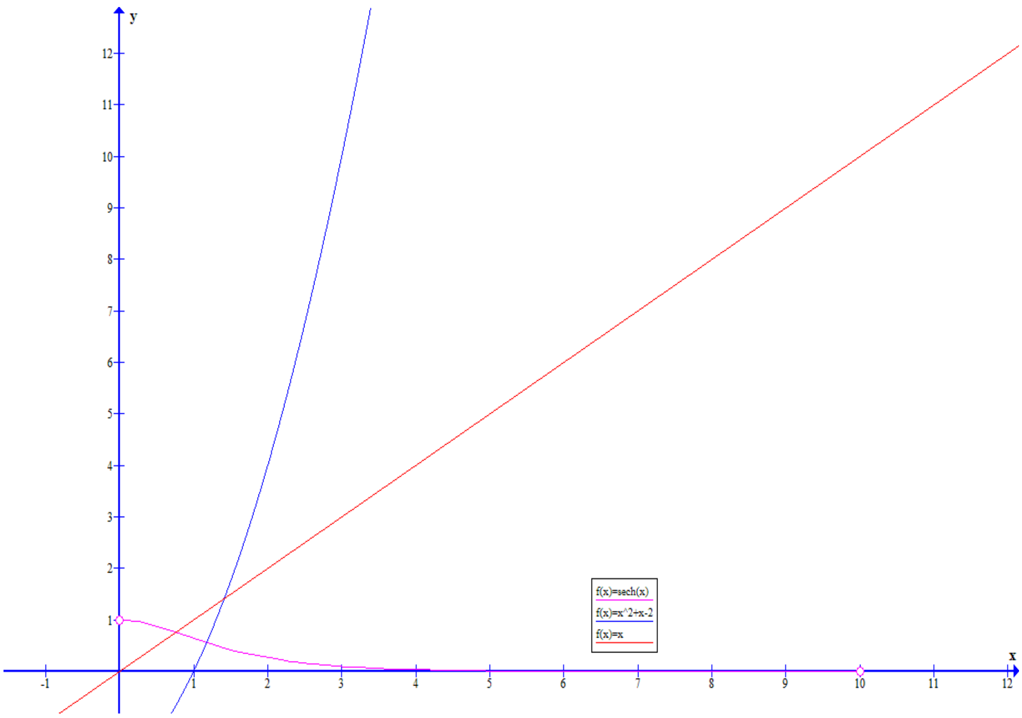

Figure 1: The Plot of Restoring Forces

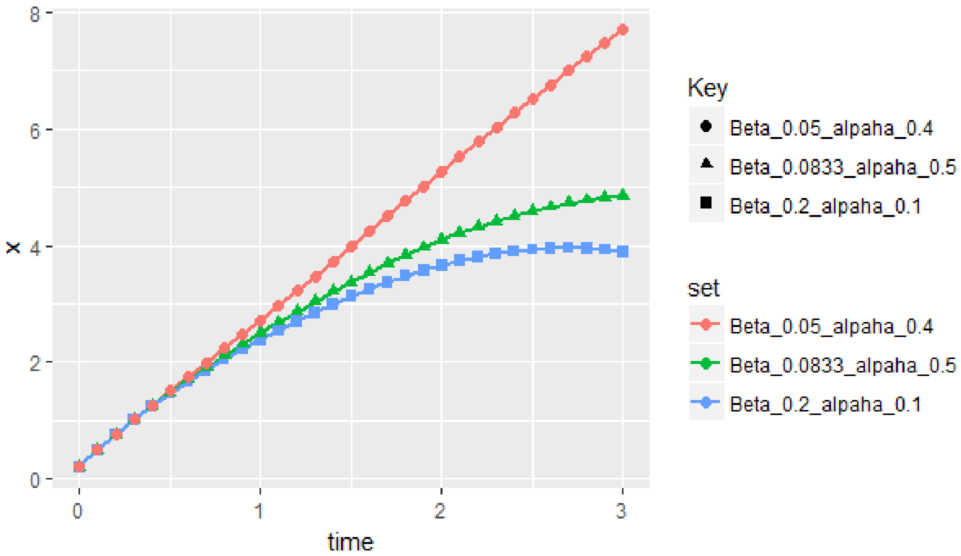

Figure 2: The graph of the Newton-Raphson results

References

1. Aminer Titus J.O. and Okelo N.B., Wilson-Theta Algorithm Approach to solution of Dynamic Vibration Equations, Journal of Mathematics and Soft Computing, Vol 4, No.1, (2014), 7-15.
2. Blanchard P., Devancy R.L. and Hall G.R., Differential Equations, Brooks/Cole, Pacific Grove, 1996.
3. Hughes,T. J. R., Caughey and Liu, W. K., Finite-Element Methods for Non-Linear Elastodynamics which conserve energy, 1978.
4. Ricardo H.A., Modern Introduction to Differential Equations, University of NewYork, 2006.
5. Steven C. and Raymond P., Numerical Methods For Engineers, 5th ED., McGraw-Hili, New York, 2006.
6. Teschl G., Ordinary Differential Equations and Dynamical Systems, University of Vienna, Vienna Austria, 2005.
7. Wei-Chau X., Differential Equations for Engineers, Cambridge University Press, New York, 2010.
8. Wood W.L., Numerical Integration of Structural Dynamic Equations including Natural Damping and Periodic Forcing terms, Journal of Numerical Methods in Engineering, Vol.17, No.2, (1981), 281-289.
9. Zienkiewics O.C., The Finite Element Method, McGraw-Hill, 1977.