International Journal of Mathematics and Soft Computing
Vol.4, No.1 (2014), 7 - 15.

ISSN Print : 2249 - 3328
ISSN Online: 2319 - 5215

Wilson-Theta Algorithm Approach to solution of Dynamic Vibration Equations

Titus J. O. Aminer, N. B. Okelo
School of Mathematics and Actuarial Science,
Jaramogi Oginga Odinga University of Science and Technology,
P.O. Box 210-40601, Bondo, Kenya.
E-mail: titusaminer@yahoo.com, bnyaare@yahoo.com

Abstract

In this paper, we examine conservative autonomous dynamic vibration equation, $\ddot{x} = -\tanh^2 x$, which is time vibration of the displacement of a structure due to the internal forces, with no damping or external forcing. Numerical results using Wilson-theta method are tabulated and then represented graphically. Further the stability of the algorithms employed are also discussed.

Keywords: Numerical solution, dynamic vibration equation, stability.
AMS Subject Classification(2010): 35B35.

1 Introduction

Most structures are in a continuous state of dynamic motion because of random loading such as wind, vibration equipment, or human loads. Therefore a lot of consideration has been given in the design of certain facilities or structures which need to resist sudden but strong vibrations. Small surrounding vibrations are normally near the natural frequencies of the structure and are terminated by energy dissipation in the real structure.

In this study we examine the time vibration of the displacement of a structure due to the internal forces, with no damping or external forcing.

Practically, vibrations decay with time but in theory they are not so. For vibrations due to purely internal forces, the dynamic systems are referred to as conservative systems.

The methods of solution adopted for solving non-linear single-degree-of-freedom problems may be extended to multi-degree-of-freedom problems. There are a number of studies on the application of these methods of solution to linear problems but only a few have been applied to the non-linear problems.

2 Non-Linear Conservative Autonomous Second Order System

Let us consider the non-linear conservative autonomous second order system equation which is generally given by

$$\eta\ddot{x} = -\mu\dot{x} - \tau f(x) \tag{1}$$

with some initial conditions $x(0) = \alpha_0$ and $\dot{x}(0) = \alpha_1$, where η, μ and τ are real positive numbers and $-\mu\dot{x}$ is the damping force.
We note that:

(a) the system is conservative because dynamic systems obey the principle 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 that is, no resisting medium so that $-\mu\dot{x} = 0$.

Thus substituting $\eta = 1$, $-\mu\dot{x} = 0$ and $\tau = 1$ in (1) we have

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

and for this study let us consider

$$f(x) = \tanh^2 x \tag{3}$$

leading to the dynamic vibration equation

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

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

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

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 non-linear is by introducing a new independent variable $y = \frac{dx}{dt}$ and since $\frac{dy}{dt} = \frac{d^2x}{dt^2}$, the variables x and 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 vise versa.

Specifically, equation (2) is equivalent to the autonomous system,

$$\frac{dx}{dt} = y, \ \frac{dy}{dt} = -f(x), \ x(0) = \alpha_0 \text{ and } y(0) = \dot{x}(0) = \alpha_1 \text{ at } t = t_0 \tag{6}$$

From (6)

$$\frac{dy}{dx} = -\frac{f(x)}{y} \tag{7}$$

$$\Rightarrow \int_{\alpha_1}^{y} ydy = -\int_{\alpha_0}^{x} f(x)dx \text{ or } \frac{y^2}{2} - \frac{{\alpha_1}^2}{2} = -\{\int_{\alpha_0}^{0} f(x)dx + \int_{0}^{x} f(x)dx\}$$

$$\text{or } \frac{y^2}{2} + \int_0^x f(x)dx = \frac{\alpha_1^2}{2} + \int_0^{\alpha_0} f(x)dx \tag{8}$$

Thus, $KE + PE = c$ where $KE = \frac{y^2}{2}$ is the kinetic energy of the dynamic system (2), $PE = \int_0^x f(x)dx$ is the potential energy of the dynamic system (2) while $c = \frac{\alpha_1^2}{2} + \int_0^{\alpha_0} f(x)dx$ 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 behaviour.
equation (2) is said to represent the motion of:

(i) a 'hard' spring if the magnitude of the restoring force, $f(x)$ acting on the mass, does increase more rapidly than that of a linear spring,

(ii) a 'soft' spring if the magnitude of the restoring force, $f(x)$ acting on the mass, does increase less rapidly than that of a linear spring.

The above mentioned two special cases of equation (2) form the central subject of discussion in this paper.
Considering the function (2) with the restoring force

$$f(x) = x^5 + x \tag{9}$$

we have two cases: (see the figures below)

(a) $\frac{d^2x}{dt^2} + x^5 + x = 0,$

(b) $\frac{d^2x}{dt^2} + \tanh^2 x = 0.$

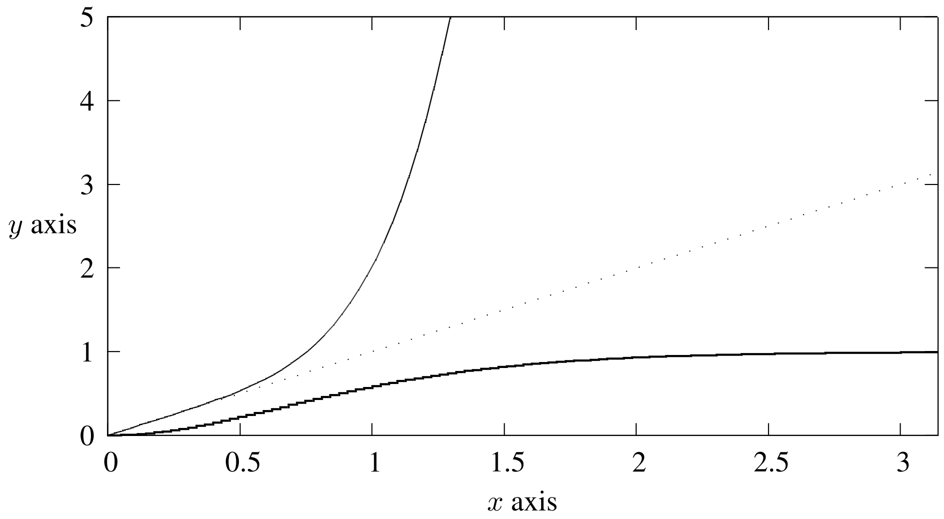


Figure 1:

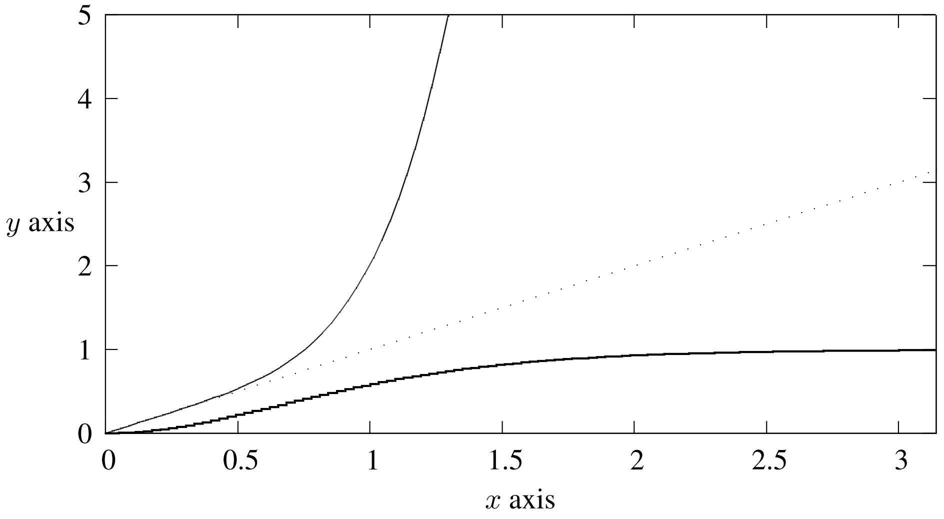


Figure 2:

The behaviour of the graphs depict clearly the idea of the 'hard' spring and 'soft' spring for the two non-linear restoring forces given.

Considering the magnitude of the non-linear restoring force, $f(x) = x^5 + x$ in case (a), since it does increase more rapidly than that of a linear spring i.e. $f(x) = x$, it represents a 'hard' spring.

On the other hand, considering the magnitude of the non-linear restoring force $f(x) = \tanh^2 x$ in case (b), since it increases less rapidly than that of a linear spring, that is, $f(x) = x$, it represents a 'soft' spring.

3 Numerical Solution of Equation (2)

Several numerical methods are available for the applications in determining solutions of non-linear systems. For the implicit non-linear dynamic system (4), the implicit dynamic methods applicable are Wilson-theta, Newmark, Hilber-Hughes-Taylor and Houbolt. Out of these, Wilson-theta method is highly stable numerically and able to converge rapidly to a meaningful solution:

3.1 Wilson-theta's Algorithm

We consider the equation

$$K\frac{d^2x}{dt^2} + f(x) = 0 \tag{10}$$

where x = displacement of a mass K at the end of a spring whose response is non-linear and $\frac{d^2x}{dt^2}$ is the acceleration.

Let $\frac{dx}{dt} = V$, $\frac{d^2x}{dt^2} = \dot{V}$; V-the velocity.

Without loss of generality the mass K is taken as unity, that is, $K = 1$.

Reduced to its equivalent first order system of ordinary differential equations, equation (10) becomes

$$\frac{d}{dt}\begin{pmatrix} x \\ V \end{pmatrix} = \begin{pmatrix} V \\ -f(x) \end{pmatrix} = \underline{f}(\underline{Z}) \text{ where } \underline{Z} = \begin{pmatrix} x \\ V \end{pmatrix} \tag{11}$$

$\Leftrightarrow$

$$\dot{\underline{Z}} = \underline{f}(\underline{Z}) \tag{12}$$

In order to make the Wilson-theta method unconditionally stable, Wilson [11] proposed the idea of applying the equation of motion not at time $t + \Delta t$ but at $t + \theta\Delta t$ and thus Wilson-Theta Method is given by

$$\underline{Z}_{n+1} - \underline{Z}_n = \theta\Delta t \underline{f}(\underline{Z}_{n+1}) + (1-\theta)\Delta t.\underline{f}(\underline{Z}_n) \tag{13}$$

$\Leftrightarrow$

$$aV_{n+1} + bV_n = x_{n+1} - x_n \tag{14}$$

$$V_{n+1} - V_n = -af_{n+1} - bf_n \tag{15}$$

with $a = h\theta$, $b = h(1-\theta)$, $h = \Delta t$ substitute $n-1$ for n in (14) and (15):

$$aV_n + bV_{n-1} = x_n - x_{n-1} \tag{16}$$

$$V_n - V_{n-1} = -af_n - bf_{n-1} \tag{17}$$

Eliminating V_{n+1} from (14) and (15) and eliminating V_{n-1} from (16) and (17) we have

$$x_{n+1} + a^2 f_{n+1} = (a+b)V_n + x_n - abf_n \tag{18}$$

$$(a+b)V_n = x_n - x_{n-1} - abf_n - b^2 f_{n-1} \tag{19}$$

Substitute $(a+b)V_n$ from (19) in (18), we get

$$x_{n+1} + a^2 f_{n+1} = 2x_n - 2abf_n - (x_{n-1} + b^2 f_{n-1}) \tag{20}$$

or

$$x_{n+1} + h^2\theta^2 f_{n+1} = 2x_n - 2h^2\theta(1-\theta)f_n - (x_{n-1} + h^2(1-\theta)^2 f_{n-1}) \tag{21}$$

or

$$x_{n+1} + h^2\theta^2 f_{n+1} = A \tag{22}$$

where A is a known value.

The left-hand sides of equations (20-22) are non-linear in x_{n+1} and so require an iterative method of solution such as Newton-Raphson.

When $\theta = 0$ then we obtain the explicit scheme

$$x_{n+1} = 2x_n - (x_{n-1} + h^2 f_{n-1}) \tag{23}$$

4 Numerical Results of The Equation

Using the numerical algorithms developed above(22), we obtain the required numerical results as follows:

4.1 Wilson-theta Algorithm

Considering the scheme (22)

$$x_{n+1} + h^2\theta^2 \tanh^2 x_{n+1} = A$$

where

$$A = 2x_n - 2h^2\theta(1-\theta)\tanh^2 x_n - (x_{n-1} + h^2(1-\theta)^2 \tanh^2 x_{n-1})$$

which is also the displacement only and two-step (three-time-level) scheme.
Using C++ computer programming with Newton-Raphson's iteration,
$x_{n+1} = x_n - \frac{F(x_n)}{F'(x_n)}$,
from the scheme (22)
$F(x_n) = x_n + h^2\theta^2 \tanh^2(x_n) - A$
thus
$F'(x_n) = 1 + 2h^2\theta^2 \tanh(x_n)\text{sech}^2(x_n)$
leading to
$x_{n+1} = x_n - \frac{x_n + h^2\theta^2 \tanh^2(x_n) - A}{1 + 2h^2\theta^2 \tanh(x_n)\text{sech}^2(x_n)}$ where $\text{sech}^2(x)$ is expressed as $\frac{1}{\cosh^2(x)}$ to be used in the computer programme as follows:

```
♯include ⟨cmath⟩
♯include ⟨cstdlib⟩
♯include ⟨iostream⟩
♯include ⟨cstdlib⟩
inline double A( double theta, double h, double x0, double x1 )
    {double u = tanh( x0) * tanh( x0);
    double v = tanh( x1) * tanh( x1);
     return 2 * x1 − 2 * h * h * theta * (1 − theta) * v − (x0 + h * h * (1 − theta) * (1 − theta * u);}
inline double ThetaNewton( double result, double h, double theta, doublex1)
        {doublev = tanh( x1) * tanh( x1);
        double z = cosh( x1) * cosh( x1);
        returnx1−(x1+h*h*theta*theta*v−result)/(1+2*h*h*theta*theta*(tanh( x1))/z);}
main( int argc, char** argu, char** argv, char** argz )
    {double theta, h, result, x0, x1, peg0, peg1;
    std::cout << "Enter the values :" << std::end1;
    std::cout << "parameter theta :"; std::cin>> theta;
    std::cout << "step size, h :"; std::cin>> h;
    std::cout << "Enter x0 :"; std::cin>> x0;
    std::cout << "Enter x1 :"; std::cin>> x1;
    for (short n = 0; n < 30; n++)
            {peg0 = x1;
            result = A(theta, h, x0, x1);
            std::cout<< "With A" << n << " = " << result <<std::end1;
```

```
        std::cout << "x" << n << " = " << x0 << std::end1;
        std::cout << "x" << n+1 << " = " << x1 <<        std::end1;
        for (short m = 0; std::abs (x0-x1) > 1e − 50; m++)
          {x0 = x1;
          x1 = ThetaNewton(result, h, theta, x1);}
        x0 = peg0;
        std::cout << std::end1;}
    system("pause"); }
```

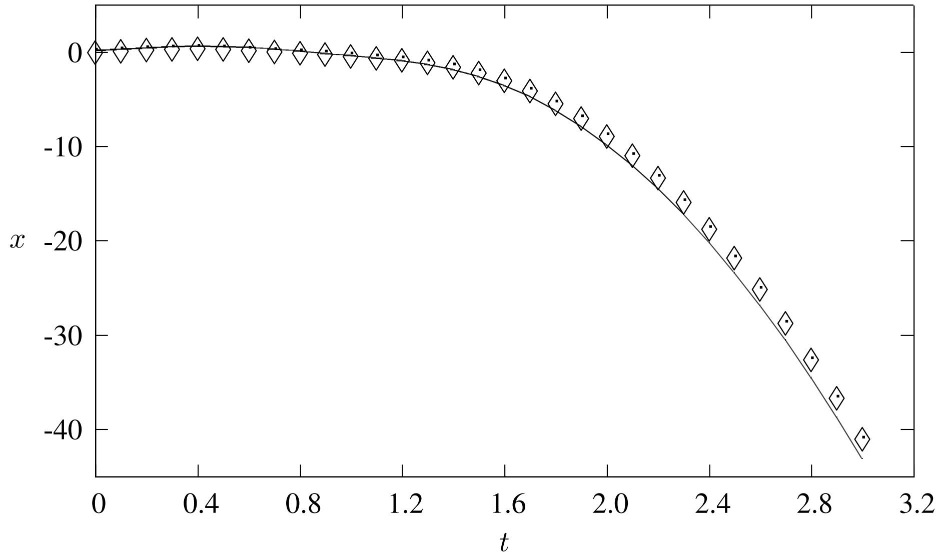
The following conditions are taken into account when compiling the results:

1. In equation (13), $0 \leq \theta \leq 1$ with maximum accuracy achieved when $\theta = \frac{1}{2}$ according to Zienkiewicz [12] and Wood [11];
2. 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 $\dot{x}_0 = 0.15$.
Thus $x_1 = 0.2 + 0.1(1.5) = 0.35$ and we have the following results:

t	x when $\theta = 0.475$	x when $\theta = \frac{1}{2}$	x when $\theta = 0.12$
0	0.2	0.2	0.2
0.1	0.35	0.35	0.35
0.2	0.472285	0.47136	0.485752
0.3	0.548559	0.546053	0.587862
0.4	0.565595	0.561539	0.634747
0.5	0.519875	0.515115	0.609886
0.6	0.4188	0.414786	0.50761
0.7	0.27831	0.27661	0.336085
0.8	0.117071	0.118834	0.116551
0.9	-0.0510805	-0.0453672	-0.124068
1.0	-0.223213	-0.213464	-0.368547
1.1	-0.41005	-0.39608	-0.623647
1.2	-0.636896	-0.618078	-0.920918
1.3	-0.944288	-0.919641	-1.30808
1.4	-1.38515	-1.35378	-1.84002
1.5	-2.01301	-1.97438	-2.56766
1.6	-2.86625	-2.82017	-3.52548
1.7	-3.96306	-3.90946	-4.72861
1.8	-5.3088	-5.24766	-6.18106
1.9	-6.90443	-6.83574	-7.88344
2.0	-8.75005	-8.67381	-9.83582

t	x when $\theta = 0.475$	x when $\theta = \frac{1}{2}$	x when $\theta = 0.12$
2.1	-10.8457	-10.7619	-12.0382
2.2	-13.1913	-13.1	-14.4906
2.3	-15.7869	-15.688	-17.193
2.4	-18.6325	-18.5261	-20.1453
2.5	-21.7281	-21.6142	-23.3477
2.6	-25.0738	-24.9523	-26.8001
2.7	-28.6694	-28.5403	-30.5025
2.8	-32.515	-32.3784	-34.4549
2.9	-36.6106	-36.4665	-38.6572
3.0	-40.9562	-40.8046	-43.1096



Key

◇ results when $\theta = \frac{1}{2}$.

...... results when $\theta = 0.475$.

$-$ results when $\theta = 0.12$.

5 Stability of The Numerical Algorithm Employed

From the results of the two-step (three-time level) scheme tabulated above and their corresponding graphical representation, it is clearly evident that for zero damping or no damping dynamic equation, the Theta method is asymptotically stable when the parameters chosen for instance $\theta = 0.475$ are within the neighbourhood of maximum accuracy parameter i.e. $\theta = \frac{1}{2}$. Otherwise, moving away from the maximum stability parameter, the method is no longer asymptotically stable.

6 Conclusion

In the Wilson-theta method, we use displacement only together with a two-step (three-time-level) scheme and a C++ computer programming which is fast and accurate in producing results. The equation has been solved using Newton-Raphson iteration method. The results are tabulated and also presented graphically through a graphical package known as GNUplot.

For the stability of the numerical schemes, a small step size is needed, with maximum accuracy achieved when the theta parameter, θ is $\frac{1}{2}$. The results of this study indicate that Wilson-theta algorithm exhibit stable case 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 parameter.

References

[1] P. Blanchard, R.L. Devancy and G.R. Hall, *Differential Equations*, Brooks/Cole, Pacific Grove, 1996.

[2] A. Chopra, *Dynamics of Structures*, Englewood Cliffs, New Jersey 07632, ISBN 0-13-855214-2, 1995.

[3] T. J. R. Hughes, Caughey and W. K. Liu, *Finite-Element Methods for Non-Linear Elastodynamics which conserve energy*, 1978.

[4] R. Kühn, *Introduction To Dynamic Systems*, King's College London, 2005.

[5] J. D. Lambert, *Computation Methods in Ordinary Differential Equations*, Wiley, New York, 1973.

[6] P.J. Olver, *Non-linear Ordinary Differential Equations*, Prentice-Hall, Inc., Upper Saddle River, N.J., 2006.

[7] R.H. Rand, *Lecture Notes on Nonlinear Vibration*, Ithaca NY 14853, 2005.

[8] H. Ricardo, *A Modern Introduction To Differential Equations*, University of New York, 2006.

[9] Simons, *Differential Equations With Applications and Historical Notes*, McGraw-Hill, 1972.

[10] G. Teschl, *Ordinary Differential Equations and Dynamical Systems*, University of Vienna, Vienna Austria, 2005.

[11] E.L. Wilson, *A Computer Program for the Dynamic Stress Analysis of Underground Structures*, SESM Report No.68-1, Division of Structural Engineering Structural Mechanics, University of California, Berkeley, 1968.

[12] C.O. Zienkiewics, *The Finite Element Method*, McGraw-Hill, 1977.