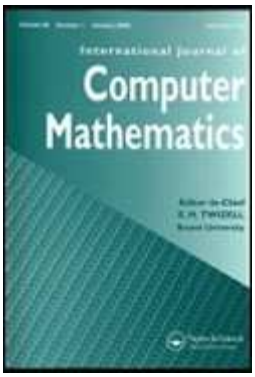
Solving 2D-wave problems by the Iterative Differential Quadrature method

Stefania Tomasiello



Solving 2D-wave problems by the Iterative Differential Quadrature method

Journal:	*International Journal of Computer Mathematics*
Manuscript ID:	GCOM-2010-0125-B.R3
Manuscript Type:	Original Article
Date Submitted by the Author:	10-Nov-2010
Complete List of Authors:	tomasiello, stefania; university of basilicata
Keywords:	discretized systems, quadrature rules, numerical stability, accuracy, wave

SOLVING 2D–WAVE PROBLEMS BY THE ITERATIVE DIFFERENTIAL QUADRATURE METHOD

(third revised version)

S. Tomasiello

DiSGG - Faculty of Engineering
University of Basilicata
C.da Macchia Romana, 85100 Potenza - Italy
Tel.: +39 (0) 971 205089; Fax: +39 (0) 971 205070
tomasiello@unibas.it

In this paper, the numerical stability of an iterative method based on differential quadrature (DQ) rules when applied to solve a two–dimensional (2D) wave problem is discussed. The physical model of a vibrating membrane, with different initial conditions, is considered. The stability analysis is performed by the matrix method generalized for a 2D space–time domain. This method was presented few years ago by the same author as an analytical support to check the stability of the iterative differential quadrature method in 1D space–time domains. The stability analysis confirms here the conditionally stable nature of the method. The accuracy of the solution is discussed too.

KEY WORDS: discretized systems, quadrature rules, numerical stability, accuracy

1. Introduction

Solving wave–type problems by numerical methods may be troublesome because of phenomena as dissipation and dispersion, affecting many numerical schemes. These phenomena impact on the temporal and the spatial discretization respectively. So, dispersion means that waves either travelling to different directions or having different frequencies propagates with different speed. These numerical drawbacks have encouraged research in the field: in particular, dispersion and dissipation have been largely investigated for finite difference schemes and Runge–Kutta schemes, especially in the case of the first–order hyperbolic equations [1–3]. The wave equation has been also pointed out, by means of numerical simulations, as an example of dynamic numerical instability due to the DQ discretization in the space (i.e. by using Runge–Kutta method to numerically integrate in time direction) [4] and in general to collocation methods [5]. In this paper, a typical second–order hyperbolic initial–boundary values problem, i.e. the vibrations of a rectangular membrane, is considered.

Membranes can be seen as plates with a negligible thickness, i.e. plates which can be devoid of flexural rigidity, carrying the lateral loads by axial tensile forces (and shear forces) acting in the plate middle surface.

Besides, well–known analogies exist between the separate problems of transverse free vibration and buckling of a polygonal plate simply supported all around and the problem of the transverse vibration of a prestretched membrane having no deflection at its edges [6]. This topic gained the attention of the researcher through the last three decades. Some recent papers has been devoted to the analysis of membranes [7-9].

In particular, Shu and co–workers developed a local radial basis function–based differential quadrature (LRBFDQ) method to handle arbitrarily shaped membranes [9]. The LRBFDQ method is applied to discretize the space domain, by obtaining accurate numerical results without any domain decomposition technique for highly concave–shaped membranes and multi–connected membranes with a hole.

The problem of the vibrations of a rectangular membrane is here solved by the Iterative Differential Quadrature (IDQ) method, a method based on the application of the DQ rules in space and time, proposed by the author and appeared for the first time in 2003 [10].

The method is inspired by the Differential Quadrature Method (DQM), introduced by Bellman and Casti [11], but, in order to ensure enough accurate results without diminishing the computational efficiency, it uses a general approach to generate the sampling points,

i.e. by means of the zeros of the first order derivative of Gegenbauer polynomials. In this way, not only the usual Gauss–Chebyshev–Lobatto (GCL) distribution can be used, but one can find the best sampling points distribution to satisfy stability and accuracy, which often requires a different distribution [12–13]. In the IDQ method this idea goes with a time marching schemes, so the solution is computed in a certain number of subdomains by retrieving each time the initial conditions by the immediately preceding calculus step. This time marching scheme has been adopted by other authors: independently in [14](i.e. the papers [10] and [14] were accepted approximately at the same time) and in 2007 in [15] to solve numerically two different problems defined in a two–dimensional space domain.

In this paper, the nature of the numerical solutions, obtained by the IDQ method, for a 2D–wave problem is formally assessed by means of a stability analysis based on the matrix method. To the best knowledge of the author, a similar application is not recoverable in the literature, *whereas discussion about the behaviour of the error function (and consistency) for the same problem can be retrieved in* [16].

In this paper, the wave equation is discretized in space and time and a recursive system is deduced, to support the discussion of the stability of the method.

The procedure is conditionally stable, i.e. a bounded response is ensured only below a certain limit on the length of the time interval with a careful distribution of the sampling points, as the stability analysis reveals. *Many numerical experiments have been carried out, but just some significant cases have been reported here for sake of brevity. The conclusions here outlined are in agreement with the ones referred to the 1D–wave problem* [17].

It should be pointed out that the recursive equation of the IDQ method here discussed has to be intended as a time advancing scheme and not as an iterative numerical scheme to extend the Newton's method in solving differential equations [18-19]. The goal of the IDQ method is to apply DQ rules in space and time, ensuring stable long term solutions, without combining the discretization scheme in the space domain and another numerical technique (e.g. Runge–Kutta) to compute the time dependent solution.

2. The IDQ method and the quadrature rules: an overview

The idea of IDQ method is to use the quadrature rules in space and time direction by using a suitable sampling points distribution, i.e. a distribution which allows stability and accuracy. In particular, for dimensionless problems, the time axis can be regarded as a unitary intervals series, after a time scaling operation by an interval $\Delta\tau$ of suitable length.

Let consider an initial boundary value problem with real variables $x, y, t \quad \epsilon\, [0,1] \times [0,1] \times [0, \infty)$ and its solution written as:

$$w(x,y,t) = f(x)g(y)h(t)$$

and by using the Lagrange interpolation polynomials as test functions one has:

$$f(x) = \sum_{k=0}^{N_x} L_k(x) f_k, \quad g(y) = \sum_{l=0}^{N_y} L_l(y) g_l, \quad h(t) = \sum_{p=0}^{N_t} L_p(t) h_p$$

where $L_k(x)$, $L_l(y)$, $L_p(t)$ are the Lagrange interpolation shape functions at points x_k, y_l and t_p respectively.

The rth–order x–partial derivative of the function $w(x,y,t)$ at a point $x = x_i$, for any $y = y_j$ and $t = t_m$, may be written as

$$\left[\frac{\partial^r w}{\partial x^r}\right]_{x=x_i} = \sum_{k=1}^{N_x} \frac{d^r L_k(x_i)}{dx^r} f_k g_i h_m = \sum_{k=1}^{N_x} A_{ik}^{(r)} w_{kjm} \qquad i = 1, 2, \ldots, N_x$$

the sth–order y–partial derivative of the function $w(x,y,t)$ at a point $y = y_j$, for any $x = x_i$ and $t = t_m$, may be written as

$$\left[\frac{\partial^s w}{\partial y^s}\right]_{y=y_j} = \sum_{l=1}^{N_y} \frac{d^s L_l(y_j)}{dy^s} f_i g_l h_m = \sum_{l=1}^{N_y} B_{jl}^{(s)} w_{ilm} \qquad j = 1, 2, \ldots, N_y$$

the qth–order t–partial derivative of the function $w(x,y,t)$ at a point $t = t_m$, for any $x = x_i$ and $y = y_j$, may be written as

$$\left[\frac{\partial^q w}{\partial t^q}\right]_{t=t_m} = \sum_{p=1}^{N_t} \frac{d^q L_p(t_m)}{dt^q} f_i g_j h_p = \sum_{p=1}^{N_t} C_{mp}^{(q)} w_{ijp} \qquad m = 1, 2, \ldots, N_t$$

where $A_{ik}^{(r)}$, $B_{jl}^{(s)}$ and $C_{mp}^{(q)}$ are the weighting coefficients in the x, y, t directions respectively, and N_x, N_y, N_t represent the number of sampling points in the cited directions.

The off–diagonal terms of the weighting coefficient matrix of the first–order derivative, in the x direction for example, turns out to be:

$$A_{ij}^{(1)} = \frac{\prod_{\substack{\nu=1 \\ \nu \neq i}}^{N_x} (x_i - x_\nu)}{(x_i - x_j) \prod_{\substack{\nu=1 \\ \nu \neq j}}^{N_x} (x_j - x_\nu)} \qquad i, j = 1, 2, \ldots, N_x \qquad j \neq i. \tag{1}$$

The off–diagonal terms of the weighting coefficient matrix of the higher–order derivative are obtained through the recurrence relationship:

$$A_{ij}^{(r)} = r\left[A_{ii}^{(r-1)}A_{ij}^{(1)} - \frac{A_{ij}^{(r-1)}}{(x_i - x_j)}\right] \qquad i,j = 1,2,\ldots,N_x \qquad j \neq i \tag{2}$$

where $2 \leq r \leq (N_x - 1)$.

The diagonal terms of the weighting coefficient matrix are given by:

$$A_{ii}^{(r)} = -\sum_{\substack{\nu=1\\ \nu\neq i}}^{N_x} A_{i\nu}^{(r)} \qquad i = 1,2,\ldots,N_x \tag{3}$$

where $1 \leq r \leq (N_x - 1)$.

Assuming the Lagrange interpolated polynomial as test functions, there is no restriction in the choice of the grid coordinates.

In the IDQ method, the quadrature grid can be regarded as a series of M sub–grids with $N_x \times N_y \times N_t$ points, where M is represantative of a time t. *The solution obtained by the IDQ method is computed for M subdomains one by one, by retriving each time the initial conditions by the immediately preceding calculus step. This procedure can be formalized by a recursive formula as explained in section 5. The subdomain is constructed by means of a suitable sampling points grid (for the problem here considered a 3D grid of dimension $N_x \times N_y \times N_t$). It should be pointed out that in general the distribution of the sampling points may be different for each subdomain and in this case the method iterates the computing of the the weighting coefficients too.*

The IDQ method computes the sampling points distribution as the zeros of the first order derivative of Gegenbauer polynomials.

Gegenbauer polynomials can be seen as a particular case of the Jacobi polynomials. An explicit representation of the ultraspherical polynomials of degree n and order λ, $G_n^{\lambda}(z)$, can be retrieved in [20].

For $\lambda = 1/2$, Gegenbauer polynomials reduce to Legendre polynomials; for $\lambda \to 0$, Gegenbauer polynomials multiplied by λ^{-1} differ in the limit by a constant respect to the Chebyshev polynomials and the shifted Gauss–Chebyshev–Lobatto (GCL) points can be retrieved.

Because of the separation of the variables, following previous results [12-14], the solution computed on the ith time interval by means of the IDQ method can be written as:

$$w(x,y,t) = \sum_{k=1}^{N_t}\sum_{j=1}^{N_y}\sum_{i=1}^{N_x} \bar{V}_i(x)\bar{V}_j(y)\bar{V}_k(t)w_{ijk} \tag{4}$$

where $\bar{V}(x)$, $\bar{V}(y)$, $\bar{V}(t)$ have the same expression, here reported for $\bar{V}(x)$

$$\bar{V}_i(x) = \delta_{1i} + \sum_{r=1}^{N_x-1} A_{1i}^{(r)}\frac{x^r}{r!}$$

It has to be underlined that if there is no time scaling operation from the interval $\Delta\tau$, the weighting coefficients in the time direction can be computed as $A_{ij}^{(r)}/\Delta\tau^r$, where the $A_{ij}^{(r)}$ terms are referred to the unitary interval.

3. The model

Let consider a rectangular membrane with dimensions $a \times b$; if one assumes that the horizontal tensile (prestressing) force per unit lenght H is uniform in all directions and without other loads, the deflection of the membrane satisfies the differential equation

$$\mu\frac{\partial^2 w}{\partial T^2} - H\left(\frac{\partial^2 w}{\partial X^2} + \frac{\partial^2 w}{\partial Y^2}\right) = 0 \tag{5}$$

where μ is the mass per unit area.

By using the following dimensionless variables

$$u = \frac{W}{a}, \quad x = \frac{X}{a}, \quad y = \frac{Y}{b}, \quad t = \sqrt{\frac{H}{\mu a^2}}T$$

one has

$$\frac{\partial^2 u}{\partial t^2} - \left(\frac{\partial^2 u}{\partial x^2} + \beta^2\frac{\partial^2 u}{\partial y^2}\right) = 0 \tag{6}$$

where $\beta = \frac{a}{b}$ is the aspect ratio.

In what follows, $a = b$ (i.e. $\beta = 1$) and null displacements on the boundary will be considered.

Two cases will be examined for as many initial conditions.

First case

By assuming the following initial conditions

$$u(x,y,0) = x(x-1)(y-1)y, \quad \frac{\partial u}{\partial t} = 0 \tag{7}$$

the exact solution for the problem is [21]

$$u(x,y,t) = \frac{16}{\pi^6} \sum_{l,j=1}^{\infty} \frac{\left[(-1)^l - 1\right]\left[(-1)^j - 1\right] \sin l\pi x \sin j\pi y \cos \omega_{lj} t}{l^3 j^3}$$

being $\omega_{lj} = \pi\sqrt{l^2 + j^2}$ the natural frequencies of the system.

Second case

If the initial conditions are (as proposed in [4])

$$u(x,y,0) = \exp -200[(x-0.5)^2 + (y-0.5)^2], \quad \frac{\partial u}{\partial t} = 0 \tag{8}$$

the exact solution is

$$u(x,y,t) = \frac{\pi}{50} \sum_{l,j=1}^{\infty} \sin\frac{l\pi}{2} \sin\frac{j\pi}{2} \exp\frac{-(i^2+j^2)\pi^2}{800} \sin l\pi x \sin j\pi y \cos \omega_{lj} t$$

being $\omega_{lj} = \pi\sqrt{l^2 + j^2}$ the natural frequencies of the system.

4. The discretized equations

The solution obtained by the IDQ method is computed for M subdomains. The solution over the ith interval can be denoted as $u^{[i]}(x,y,t)$, even if in the following equations the index $[i]$ has been omitted for simplicity.

In order to overcome the possible drawback of having a small time interval length $\Delta\tau_i$ as denominator of the terms of the matrix of the weighting coefficients in the time direction (see section 2), i.e. by using just the weighting coefficients computed on the unitary time interval for computational advantage, the following variable change has to be considered:

$$\tau^{[i]} = \alpha_i (t - \sum_{k=1}^{i-1} \Delta\tau_k), \quad \alpha_i = \frac{1}{\Delta\tau_i}$$

where the new variable τ belongs to the interval $[0,1]$; if all the time intervals have the same length, one has $t = \tau^{[i]}\Delta\tau_i + i - 1$. So the 2D–wave eq. (6) becomes (with $\beta = 1$) the discretized system over the ith sub–domain:

$$\alpha_i^2 \sum_{s=1}^{N_t} C_{ks}^{(2)} u_{ljs} - \left(\sum_{p=1}^{N_x} A_{lp}^{(2)} u_{pjk} + \sum_{q=1}^{N_y} B_{jq}^{(2)} u_{lqk} \right) = 0$$

$$l = 1, \ldots, N_x, \quad j = 1, \ldots, N_y, \quad k = 1, \ldots, N_t \tag{9}$$

with $i = 1, \dots, M$ and where $A^{(2)}_{lp}$, $B^{(2)}_{jq}$ and $C^{(2)}_{ks}$ are the weighting coefficients in the x, y and τ directions respectively.

By introducing the boundary conditions in eq. (9), then $l = 2, \dots, N_x - 1$ and $j = 2, \dots, N_y - 1$; after some algebra, one has the system

$$\mathbf{C}\bar{\mathbf{u}} + \mathbf{D}\bar{\mathbf{u}} = \mathbf{0} \tag{10}$$

where the vector $\bar{\mathbf{u}}$ have as components the N_r subvectors $\bar{\mathbf{u}}_l$ of dimension N_t, obtained by writing in orderly way the components of the order 3 tensor of the displacements u_{ljk}, the matrices $\mathbf{C}$ and $\mathbf{D}$ have dimensions $N_r \times N_t$, being $N_r = (N_x - 2)(N_y - 2)$; $\mathbf{C}$ can be intended as a matrix whose diagonal elements are given by the matrix of the weighting coefficients $C^{(2)}_{ks}$ repeated N_r times, whereas $\mathbf{D}$ can be seen as the sum of the two matrices $\mathbf{D}^1$ and $\mathbf{D}^2$

$$\mathbf{D}^1 = \begin{bmatrix} d^1_{11}\mathbf{I}_{N_t} & 0 & d^1_{12}\mathbf{I}_{N_t} & \dots & d^1_{1(N_x-2)}\mathbf{I}_{N_t} & 0 \\ 0 & d^1_{11}\mathbf{I}_{N_t} & 0 & \dots & 0 & d^1_{1(N_x-2)}\mathbf{I}_{N_t} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & d^1_{(N_x-2)1}\mathbf{I}_{N_t} & 0 & \dots & 0 & d^1_{(N_x-2)(N_x-2)}\mathbf{I}_{N_t} \end{bmatrix}$$

$$\mathbf{D}^2 = \begin{bmatrix} d^2_{11}\mathbf{I}_{N_t} & \dots & d^1_{1(N_y-2)}\mathbf{I}_{N_t} & 0 & \dots & 0 \\ \vdots & \dots & \vdots & \vdots & \dots & \vdots \\ d^1_{(N_y-2)1}\mathbf{I}_{N_t} & \dots & d^1_{(N_y-2)(N_y-2)}\mathbf{I}_{N_t} & 0 & \dots & 0 \\ 0 & \dots & 0 & d^2_{11}\mathbf{I}_{N_t} & \dots & d^1_{1(N_y-2)}\mathbf{I}_{N_t} \\ \vdots & \dots & \vdots & \vdots & \dots & \vdots \\ 0 & \dots & 0 & d^1_{(N_y-2)1}\mathbf{I}_{N_t} & \dots & d^1_{(N_y-2)(N_y-2)}\mathbf{I}_{N_t} \end{bmatrix}$$

where $\mathbf{I}_{N_t}$ is the identity matrix of dimension N_t, $d^1_{pj} = -A^{(2)}_{(p+1)(j+1)}$, $d^2_{lk} = -B^{(2)}_{(l+1)(k+1)}$, with $p, j = 1, \dots, N_x - 2$ and $l, k = 1, \dots, N_y - 2$.

The eigenvalues of the matrix $\mathbf{D}$ are the square of the natural frequencies of the system.

Where accuracy requires more sampling points, as for the second case (see next section), the symmetry properties can be invoked by considering a quarter of the membrane, by having so mixed boundary conditions, i.e.

$$u(0, y, t) = u(x, 0, t) = 0$$

and

$$\left[\frac{\partial u}{\partial x}\right]_{x=\frac{1}{2}} = 0, \quad \left[\frac{\partial u}{\partial y}\right]_{y=\frac{1}{2}} = 0$$

So, by applying the quadrature rules:

$$u_{1jk} = u_{i1k} = 0, \quad u_{\overline{N}_x jk} = -\sum_{p=1}^{N_x-1} \frac{A^{(1)}_{\overline{N}_x p}}{A^{(1)}_{\overline{N}_x \overline{N}_x}} u_{pjk}, \quad u_{i\overline{N}_y k} = -\sum_{p=1}^{N_y-1} \frac{B^{(1)}_{\overline{N}_y p}}{B^{(1)}_{\overline{N}_y \overline{N}_y}} u_{ipk}$$

with $i = 2, \ldots, \overline{N}_x - 1$, $j = 2, \ldots, \overline{N}_y - 1$, $k = 1, \ldots, N_t$ and where $\overline{N}_x$ and $\overline{N}_y$ are the sampling points, referred to a quarter of membrane, in the x and y direction respectively. *In this case, the non–null elements of the matrices* $\mathbf{D}^1$ *and* $\mathbf{D}^2$ *become:*

$$d^1_{pj} = -A^{(2)}_{(p+1)(j+1)} + \frac{A^{(1)}_{\overline{N}_x (j+1)}}{A^{(1)}_{\overline{N}_x \overline{N}_x}} A^{(2)}_{(p+1)\overline{N}_x}, \quad p, j = 1, \ldots, \overline{N}_x - 2 \tag{11}$$

$$d^2_{lk} = -B^{(2)}_{(l+1)(k+1)} + \frac{B^{(1)}_{\overline{N}_y (k+1)}}{B^{(1)}_{\overline{N}_y \overline{N}_y}} B^{(2)}_{(l+1)\overline{N}_y}, \quad l, k = 1, \ldots, \overline{N}_y - 2 \tag{12}$$

5. Stability analysis and discussion

The stability analysis is here performed by the matrix method. *It should be pointed out that formulas and results obtained in this section do not depend on the separation of variables: the recursive formula of the IDQ method, which will be discussed in this section, allows to compute the unknown function values* u_{ijk} *at the grid points. The separation of variables instead allows to easily draw the approximate solution in each subdomain as explained in section 2 and it is useful to deduce the behaviour of the error function as discussed in* [16]. In order to construct a recursive formula and perform the stability analysis, one has to determine displacements and velocities at τ_i, i.e. $u_{lj(i+1)}$ and $\dot{u}_{lj(i+1)}$ for $l = 2, \ldots, N_x - 1$ and $j = 2, \ldots, N_y - 1$, with sufficient accuracy.

Obviously, on the local ith sub–domain, one can read:

$$u_{lj1} = u_{lji}, \qquad u_{ljN} = u_{lj(i+1)}$$

$$\dot{u}_{lj1} = \dot{u}_{lji}, \qquad \dot{u}_{ljN} = \dot{u}_{lj(i+1)}$$

Thanks to quadrature rules and after some algebra, one can write the following equations, in addition to equations (10):

$$\begin{bmatrix} u_{22(i+1)} \\ \vdots \\ u_{(N_x-1)(N_y-1)(i+1)} \end{bmatrix} = -\frac{C^{(1)}_{11}}{C^{(1)}_{1N_t}} \begin{bmatrix} u_{22i} \\ \vdots \\ u_{(N_x-1)(N_y-1)i} \end{bmatrix} + \frac{1}{C^{(1)}_{1N_t}} \begin{bmatrix} \dot{u}_{22i} \\ \vdots \\ \dot{u}_{(N_x-1)(N_y-1)i} \end{bmatrix} - \frac{1}{C^{(1)}_{1N_t}} \mathbf{V} \begin{bmatrix} \bar{\mathbf{u}}_1 \\ \vdots \\ \bar{\mathbf{u}}_{N_r} \end{bmatrix} \tag{13}$$

$$\begin{bmatrix} \dot{u}_{22(i+1)} \\ \vdots \\ \dot{u}_{(N_x-1)(N_y-1)(i+1)} \end{bmatrix} = q_1 \begin{bmatrix} u_{22i} \\ \vdots \\ u_{(N_x-1)(N_y-1)i} \end{bmatrix} + \frac{C^{(1)}_{N_tN_t}}{C^{(1)}_{1N_t}} \begin{bmatrix} \dot{u}_{22i} \\ \vdots \\ \dot{u}_{(N_x-1)(N_y-1)i} \end{bmatrix} + \mathbf{W} \begin{bmatrix} \bar{\mathbf{u}}_1 \\ \vdots \\ \bar{\mathbf{u}}_{N_r} \end{bmatrix} \tag{14}$$

where $\mathbf{V}$ can be intended as a diagonal matrix whose diagonal terms are given by the row vector $\bar{\mathbf{C}}$, of dimension $N_t - 2$, repeated N_r times

$$\bar{\mathbf{C}} = \left[C^{(1)}_{12}, \dots C^{(1)}_{1(N_t-1)} \right]$$

Analougsly $\mathbf{W}$ can be intended as a diagonal matrix whose diagonal terms are given by the row vector $\bar{\mathbf{Q}}$, of dimension $N_t - 2$, repeated N_r times

$$\bar{\mathbf{Q}} = [q_2, \dots q_{N_t-1}]$$

and

$$q_j = C^{(1)}_{N_tj} - \frac{C^{(1)}_{N_tN_t}}{C^{(1)}_{1N_t}} C^{(1)}_{1j}, \qquad j = 1, \dots, N_t - 1$$

Vectors $\bar{\mathbf{u}}_l$ can be obtained from the equations (10) and substituted into equations (13–14) to give the required displacements and velocities.

After some algebra, from the discretized equations (10) one can deduce:

$$\begin{bmatrix} \bar{\mathbf{u}}_1 \\ \vdots \\ \bar{\mathbf{u}}_{N_r} \end{bmatrix} = -\mathbf{P}^{-1} \begin{bmatrix} \mathbf{V}_A u_{22i} \\ \vdots \\ \mathbf{V}_A u_{(N_x-1)(N_y-1)i} \end{bmatrix} - \mathbf{P}^{-1} \begin{bmatrix} \dot{u}_{22i} \\ \vdots \\ \dot{u}_{(N_x-1)(N_y-1)i} \end{bmatrix} \tag{15}$$

where the matrix $\mathbf{P}$ is the sum of the matrices $\mathbf{D}$ and $\mathbf{T}$, whose elements are

$$t_{kj} = \alpha^2 \left(C^{(2)}_{kj} - C^{(2)}_{kN_t} \frac{C^{(1)}_{1j}}{C^{(1)}_{iN_t}} \right), \qquad k, j = 2, \dots, N_t - 1$$

and

$$\mathbf{V}_A^T = \begin{bmatrix} t_{21} & \dots & t_{(N_t-1)1} \end{bmatrix}$$

$$\mathbf{V}_B^T = \frac{\alpha^2}{C_{1N_t}^{(1)}} \begin{bmatrix} C_{1N_t}^{(2)} & \dots & C_{N_tN_t}^{(2)} \end{bmatrix}$$

By substituting equation (15) into equations (13–14), one has:

$$\begin{bmatrix} \mathbf{u}_{i+1} \\ \dot{\mathbf{u}}_{i+1} \end{bmatrix} = \mathbf{U} \begin{bmatrix} \mathbf{u}_i \\ \dot{\mathbf{u}}_i \end{bmatrix} \tag{16}$$

where the vectors $\mathbf{u}$ and $\dot{\mathbf{u}}$ at positions i and $i+1$ have, at the same positions, N_r components, and

$$\mathbf{U} = \begin{bmatrix} \mathbf{U}_{11} & \mathbf{U}_{12} \\ \mathbf{U}_{21} & \mathbf{U}_{22} \end{bmatrix}$$

with

$$\mathbf{U}_{11} = \frac{1}{C_{1N_t}^{(1)}}\mathbf{M} - \frac{C_{11}^{(1)}}{C_{1N_t}^{(1)}}\mathbf{I}_{N_r}$$

$$\mathbf{U}_{12} = \frac{1}{C_{1N_t}^{(1)}}\mathbf{M}' + \frac{1}{C_{1N_t}^{(1)}}\mathbf{I}_{N_r}$$

$$\mathbf{U}_{21} = \mathbf{G} + q_1\mathbf{I}_{N_r}$$

$$\mathbf{U}_{22} = \mathbf{G}' + \frac{C_{N_tN_t}^{(1)}}{C_{1N_t}^{(1)}}\mathbf{I}_{N_r}$$

Matrices $\mathbf{M}$, $\mathbf{M}'$, $\mathbf{G}$ and $\mathbf{G}'$ have dimension N_r; $\mathbf{I}_{N_r}$ is the identity matrix of dimension N_r. Their elements are obtained by means of the products:

$$M_{kj} = \bar{\mathbf{C}} \cdot \bar{\mathbf{P}}_{kj} \cdot \mathbf{V}_A, \qquad M'_{kj} = \bar{\mathbf{C}} \cdot \bar{\mathbf{P}}_{kj} \cdot \mathbf{V}_B,$$

$$G_{kj} = -\bar{\mathbf{Q}} \cdot \bar{\mathbf{P}}_{kj} \cdot \mathbf{V}_A, \qquad G'_{kj} = -\bar{\mathbf{Q}} \cdot \bar{\mathbf{P}}_{kj} \cdot \mathbf{V}_B,$$

with $k, j = 1, \dots, N_r$, being $\bar{\mathbf{P}}_{kj}$ submatrices of the matrix $\mathbf{P}^{-1}$.

A bounded response depends upon the matrix $\mathbf{U}$. As the mathematical theory states, stability is ensured if the spectral radius ρ_U of the matrix $\mathbf{U}$ does not exceed the unitary value.

To achieve a satisfactory algorithmic performance, i.e. to avoid the so–called algorithmic damping, the presence of complex eigenvalues with a modulus of 1 is needed.

Because of the complexity of a symbolic proof, discussion about ρ_U (i.e. stability) is conducted by significant cases.

In what follows, λ_x, λ_y and λ_t indicate the order of the Gegenbauer polynomials to be used to construct the distributions of sampling points in x, y and τ directions respectively.

In order to limit the effect of round–off errors on the numerical results, all the computations have been done by using high working precision (i.e. 80 digits).

In all the computations and for all the sub–domains, $N_y = N_x$ and $\lambda_y = \lambda_x$ was assumed. No damping appeared.

In order to ensure stability, for a certain N_t and by increasing N_x, a smaller $\Delta\tau$ is needed; $\lambda_x = 0$ or $\lambda_x = 0.5$ give the same solution with regard to stability and not so different results with regard to accuracy: one can see as an example Table 1, where the natural frequencies for $N_x = 8$, $\lambda_x = 0.5$ and $\lambda_x = 0$ are tabled (the bold values in these tables are referred to not repeated frequency values). In all the figures where the time axis appears, the units on this axis are obtained by $\Delta\tau^{-1}$ sub-intervals. Figure 1 is referred to the first case and shows the maximum $\Delta\tau$ required by stability (i.e. the $\Delta\tau$ giving $\rho_U = 1$) for different values of N_x, with $N_t = 6$ and $\lambda_x = \lambda_t = 0.5$. These values of $\Delta\tau$ are obtained by a recursive procedure *(see also* [17]*)*. Obviously, $\Delta\tau$ (i.e. the stability) is influenced by the number of sampling points in the τ direction: as an example, one can consider that $N_t = 6$ and $N_x = 10$ require $\Delta\tau = 0.1$ to ensure stability, whereas the same result can be obtained with $N_t = N_x = 10$ and $\Delta\tau = 0.25$, but for computational advantage the smallest N_t is preferred; in fact, the computing time involved by the solution obtained with $N_x = 10$, $N_t = 6$ and $\Delta\tau = 0.1$ is 10.2 % less than the solution obtained with $N_x = N_t = 10$ and $\Delta\tau = 0.25$. In Figure 2, where the behaviour of the error $e(324)$ (i.e. the difference between exact and approximated solution for the displacement u_{324} with $N_x = N_y = 6$, $\lambda_x = \lambda_t = 0.5$) is showed for different $\Delta\tau$, one can see that by decreasing the value of the maximum $\Delta\tau$ which ensures stability, the accuracy is not improved (the behaviour of the other errors is similar and is not reported for brevity), even if the error exhibits a more regular behaviour. Accuracy can be improved by increasing the number of sampling points in the space directions as one can see in Figures 3, 4 and 5, where, at different time, the exact solution is compared with the numerical solution computed for $N_x = 6$, $N_x = 10$ and $N_x = 16$ respectively, by means of 2D–plots captured at $y = 0.5$ in order to visualize

better the discrepancies between the two solutions. It is worth noting that for $N_x = 6$ (Figure 3) the error has high oscillations, i.e. the numerical solution is accurate enough only at some times.

Figure 6 is referred to the second case and shows the exact and numerical solution for $N_x = 19$, $N_t = 6$ and $\Delta\tau = 0.01$ ($\lambda_x = \lambda_t = 0.5$): the accuracy is poor, so a higher number of sampling points in the space directions is required, but to achieve both accuracy and stability one has to reduce $\Delta\tau$ drastically (i.e. values less than 10^{-4}). Because of the symmetry of the problem, a quarter of the membrane can be considered and Figure 7 shows the maximum $\Delta\tau$ required by stability for this system, with $N_t = 10$ and $\lambda_x = \lambda_t = -1.4$ (for $\lambda_x = \lambda_t = 0.5$ or $\lambda_x = \lambda_t = 0$ stability is not ensured); $N_t = 10$ is the smallest value to be considered with the maximum $\Delta\tau$ to achieve stability and with a suitable value of N_x to achieve accuracy. A numerical solution accurate enough can be achieved by using $\overline{N}_x = 15$, $N_t = 10$ and $\Delta\tau = 0.01$, as one can see in Figure 8, depicting solutions at different time, apart from those referred to time less than 1 (i.e. the ones considered in [4]).

6. Conclusions

In this paper, the conditions to obtain stable and enough accurate solution for a 2D–wave problem by the iterative differential quadrature method are investigated.

In order to have an analytical support, to the best knowledge of the author not recoverable in literature, a stability analysis, by the matrix method, is performed with regard to two cases, i.e. Dirichlet and mixed boundary conditions. These two cases requires different time grid distributions and time interval lenght, in order to achieve stability, with a certain number of sampling points in space directions in order to achieve accuracy. The method is conditionally stable and the solutions are enough accurate, confirming the method as a powerful tool to compute numerical solutions in the space–time domain.

Acknowledgements

This work was partially funded by INPES S.p.A. The author is in debt with the team at INPES for executing numerical simulations.

References

[1] F. Q. HU, M. Y. HUSSAINI, J. MANTHEY 1996, *Journal of Computational Physics* **124**, 177-191. Low–dissipation and dispersion Runge–Kutta schemes for computational acoustics.

[2] M. BERNARDINI, S. PIROZZOLI 2009, *Journal of Computational Physics* **228**, 4182-4199. A general strategy for the optimization of Runge–Kutta schemes for wave propagation phenomena.

[3] A.SESCU, A. AFJEH and R.HIXON 2009, *Electronic Journal of Differential Equations* **17**, 213-225. Optimized difference schemes for multidimensional hyperbolic partial differential equations.

[4] Z. ZONG and K. Y. LAM 2002, *Computational Mechanics* **29**, 382-391. A localized differential quadrature (LDQ) method and its application to the 2D wave equation.

[5] E. HAIRER, C. LUBICH and D. COHEN 2007, *Isaac Newton Institute Technical Report Spectral semi–discretizations of weakly nonlinear wave equations over long times.*

[6] H.D. CONWAY 1960, *Can. Aeron. Journal* **6(7)**, 263–273 Analogies Between the Buckling and Vibration of Polygonal Plates and Membranes.

[7] C.P. FILIPICH and M.B. ROSALES 2010, *Journal of the Acoustical Society of America* **127(44)**, 2295–2300 A generalization of the membrane plate analogy to non–homogeneous polygonal domains consisting of homogeneous subdomains.

[8] C.P. FILIPICH and E.A. BAMBILL 2008, *Journal of the Sound and Vibration* **310 (1-2)**, 21–37 Natural frequencies of rectangular membranes with partial intermediate supports.

[9] W.X. Wu, C. Shu, C.M. Wang 2007, *Journal of the Sound and Vibration* **306**, 252–270 Vibration analysis of arbitrarily shaped membranes using local radial basis function-based differential quadrature method. subdomains.

[10] S. TOMASIELLO 2003, *Journal of Sound and Vibration* **265(3)**, 507–525 Simulating nonlinear coupled oscillators by an iterative differential quadrature method.

[11] R. BELLMAN and J. CASTI 1971, *Journal of Mathematical Analysis and Applications* **34**, 235–238. Differential quadrature and long–term integration.

[12] S. TOMASIELLO 2007, *Journal of Sound and Vibration* **301**, 374–390 A generalization of the IDQ method and a DQ based method to approximate non–smooth solutions.

[13] S. TOMASIELLO 2007, *Nonlinear Dynamics* **50**, 37–48. A DQ based approach to simulate the vibrations of buckled beams.

[14] C. SHU, Q. YAO, K. S. YEO 2002, *Computer Methods in Applied Mechanics and Engineering* **191(41–42)**, 4587–4597. Block–marching in time with DQ discretization: an efficient method for time–dependent problems.

[15] P. MALEKZADEH and H. RAIDEH 2007, *Energy Conversion and Management* **48**, 269–276. IDQ two–dimensional nonlinear transient heat transfer analysis of variable section annular fins.

Typeset by $\mathcal{A}\mathcal{M}\mathcal{S}$-TEX

[16] S. TOMASIELLO 2009, *AIP Conference Proceedings - International Conference of Computational Methods in Sciences and Engineering 2008 (ICCMSE 2008)* **1148**, 103–107. DQ solutions of a vibrating membrane model: an error analysis.

[17] S. TOMASIELLO 2010, *Numerical Algorithms Numerical stability of DQ solutions of wave problems (online), DOI 10.1007/s11075-010-9429-2.*

[18] Z. GIRGIN 2010, *International Journal for Numerical Methods in Biomedical Engineering Solution of the Blasius and Sakiadis equation by generalized iterative differential quadrature method (online).*

[19] S.K. KHATTRI and R.P. AGARWAL 2000, *ArXiv e-prints (arxiv.org) Quadrature based optimal iterative methods (http://arxiv.org/abs/1004.2930v1).*

[20] G. SZEGO 1939, *AMS Colloquium Publications* **32 Orthogonal Polynomials**.

[21] I.N. SNEDDON 1962, 293–296 Elements of Partial Differential Equations. Warsaw: PWN.

Table Captions

Table 1 - Natural frequencies for $N_x = N_y = 8$

$\lambda_x = 0$	error (%)	$\lambda_x = 0.5$	error (%)
4.44289	-0.0002	**4.44288**	-7E-05
7.02477	0.00055	7.0247	0.00153
8.88569	0.00085	**8.88559**	0.00205
9.97211	-0.3776	9.95205	-0.1757
11.3601	-0.2901	11.3424	-0.1344
12.3775	4.44396	12.5526	3.09168
13.3846	-0.4199	**13.3547**	-0.1956
13.5207	3.76437	13.6812	2.622
15.2612	2.84416	15.3907	2.02021
16.9312	-5.6942	17.1871	-7.2922

Table 1

Figure Captions

Figure 1 – First case: a stability graph N_x vs $\Delta\tau$ ($\lambda_x = \lambda_t = 0.5$, $N_t = 6$)

Figure 2 – First case: the error behaviour in time for u_{324} with $N_x = N_t = 6$ and (a) $\Delta\tau = 0.35$ (b) $\Delta\tau = 0.25$ (c) $\Delta\tau = 0.1$

Figure 3 – First case: *sectional plot* with $N_x = N_t = 6$, $\lambda_x = \lambda_t = 0.5$, $\Delta\tau = 0.25$ at y=0.5 and (a) t = 1 (b)
t = 10 (c) t = 19 (d) t = 39.9 (dashed line - approximated solution, thick line - exact solution)

Figure 4 – First case: *sectional plot* with $N_x = 10$, $N_t = 6$, $\lambda_x = \lambda_t = 0.5$, $\Delta\tau = 0.1$ at y=0.5 and (a) t = 1 (b) t = 10 (c) t = 19 (d) t = 39.9 (dashed line - approximated solution, thick line - exact solution)

Figure 5 – First case: *sectional plot* with $N_x = 16$, $N_t = 6$, $\lambda_x = \lambda_t = 0.5$, $\Delta\tau = 0.02$ at y=0.5 and (a) t = 1 (b) t = 10 (c) t = 19 (d) t = 39.9 (dashed line - approximated solution, thick line - exact solution)

Figure 6 – Second case: *sectional plot* with $N_x = 19$, $N_t = 6$, $\lambda_x = \lambda_t = 0.5$, $\Delta\tau = 0.01$ at y=0.5 and (a) t = 0 (b) t = 0.5 (c) t = 0.9 (d) t = 1 (dashed line – approximated solution, thick line – exact solution)

Figure 7 - Second case: a stability graph $\overline{N}_x$ vs $\Delta\tau$ ($\lambda_x = \lambda_t = -1.4$, $N_t = 10$)

Figure 8 – Second case: *sectional plot* with $\overline{N}_x = 15$, $N_t = 10$, $\lambda_x = \lambda_t = -1.4$, $\Delta\tau = 0.01$ at y=0.5 and (a) t = 0 (b) t = 0.2 (c) t = 0.9 (d) t = 1 (e) t = 2 (f) t = 3 (g) t = 4 (h) t = 5 (dashed line – approximated solution, thick line – exact solution)

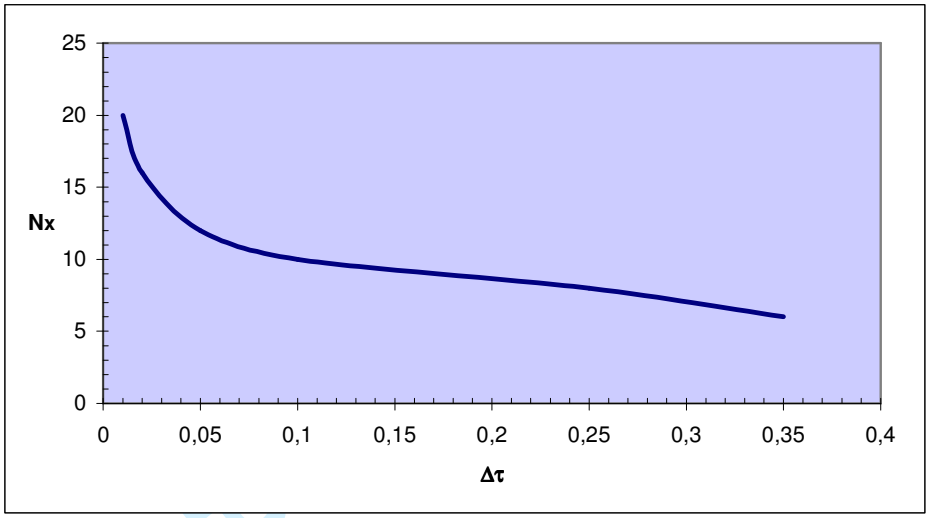
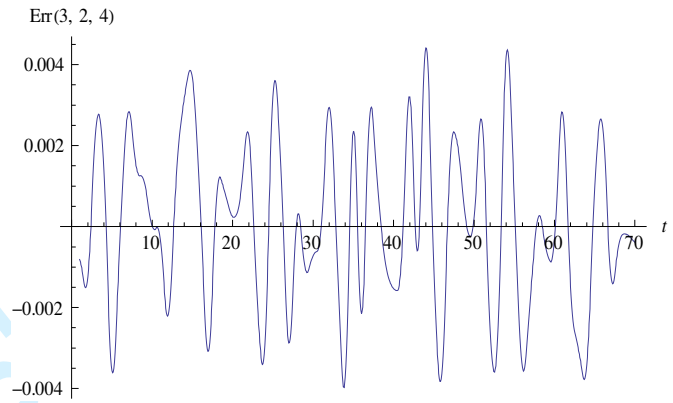
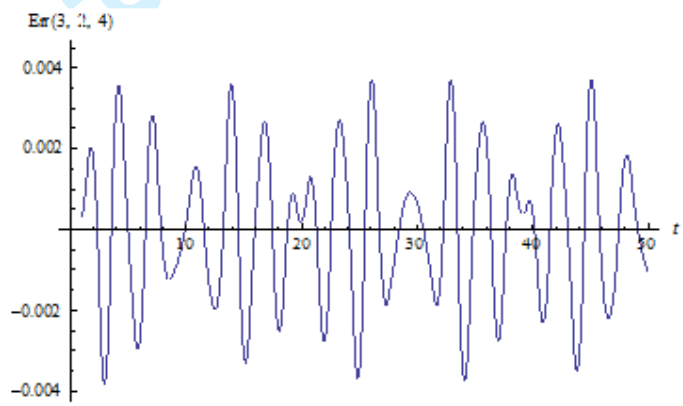


Figure 1

(a)


Err(3, 2, 4)
0.004
0.002
−0.002
−0.004
10
20
30
40
50
60
70
t

(b)


Err(3, 2, 4)
0.004
0.002
−0.002
−0.004
10
20
30
40
50
t

(c)

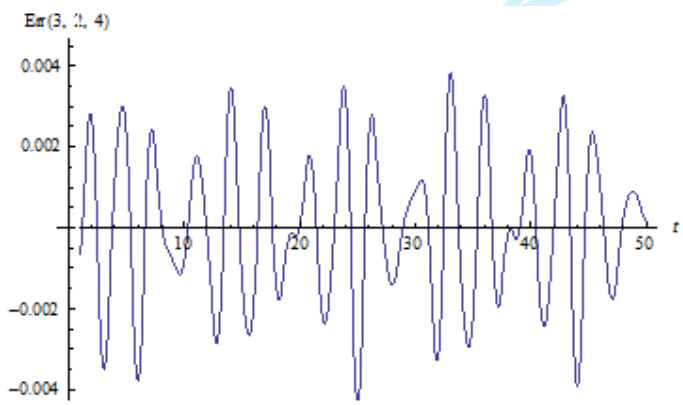
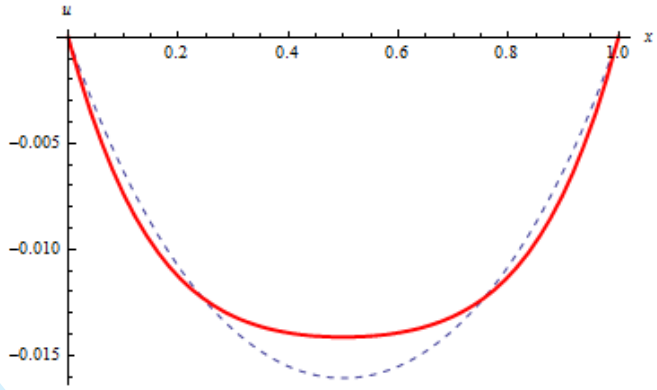
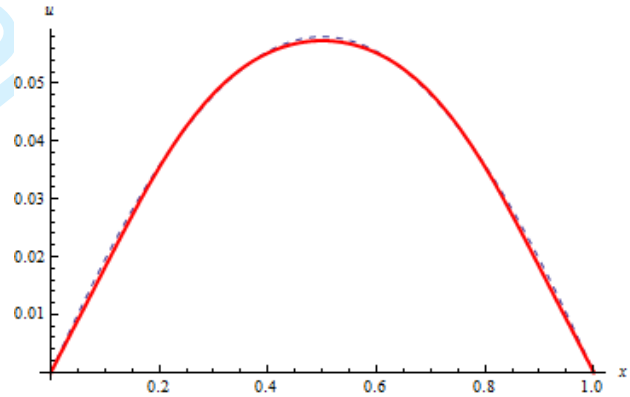

Err(3, 2, 4)
0.004
0.002
−0.002
−0.004
10
20
30
40
50
t

Figure 2

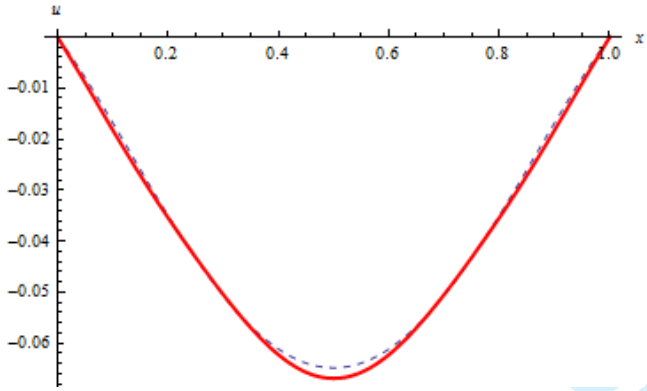
(a)



(b)



(c)



(d)

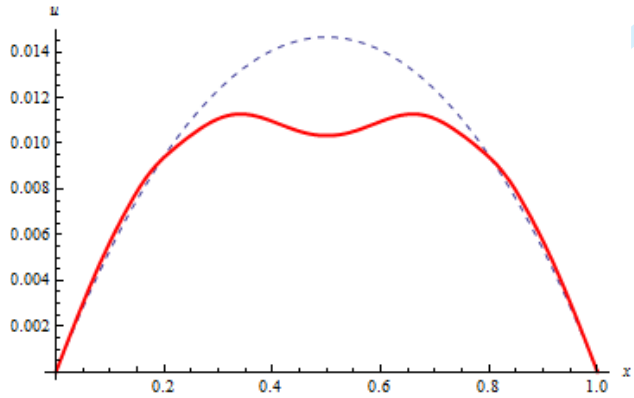
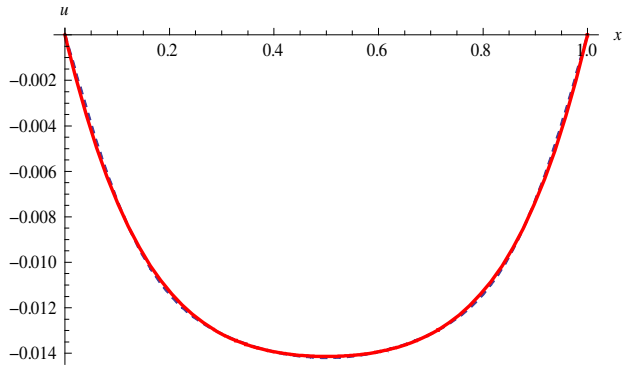
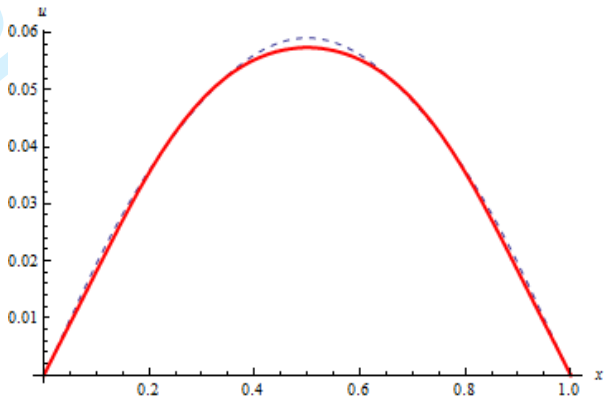


Figure 3

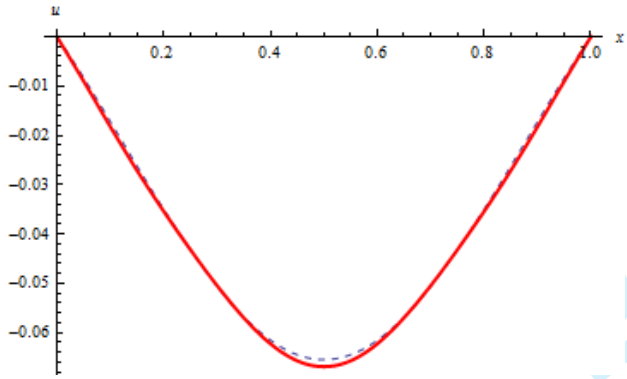
(a)



(b)



(c)



(d)

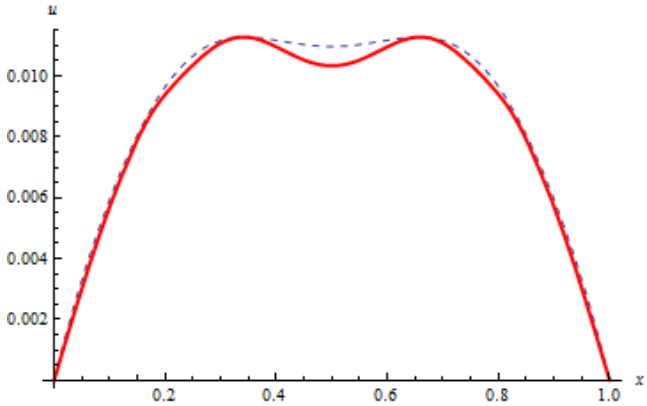
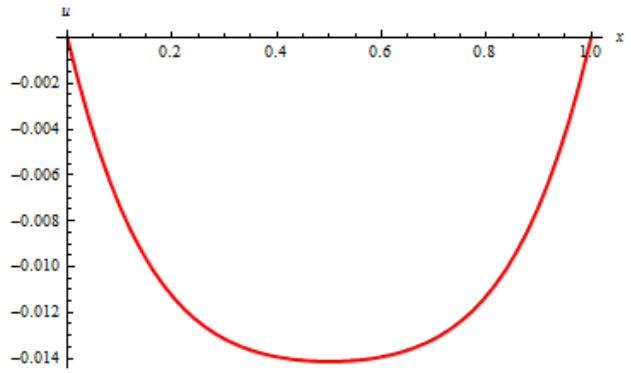
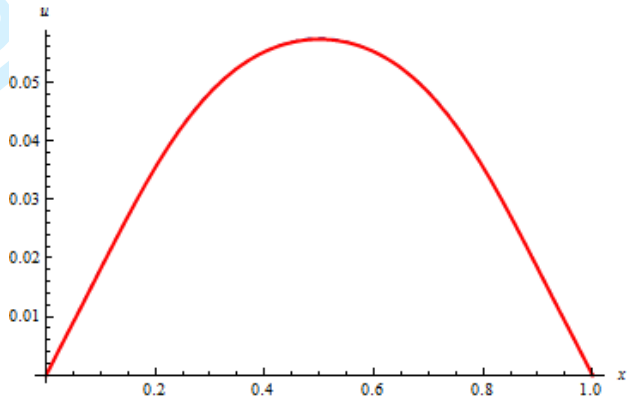


Figure 4

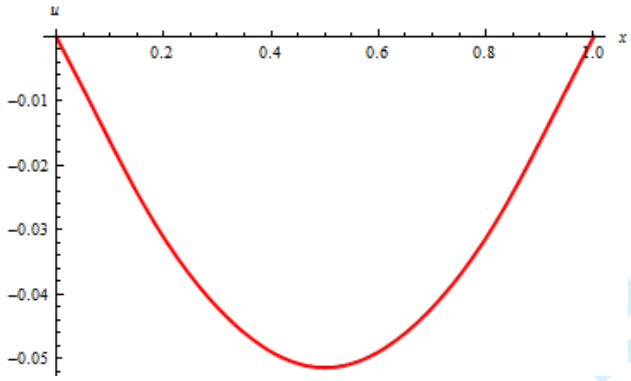
(a)



(b)



(c)



(d)

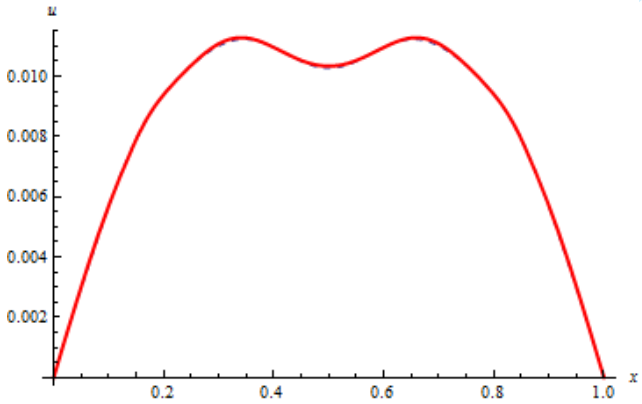
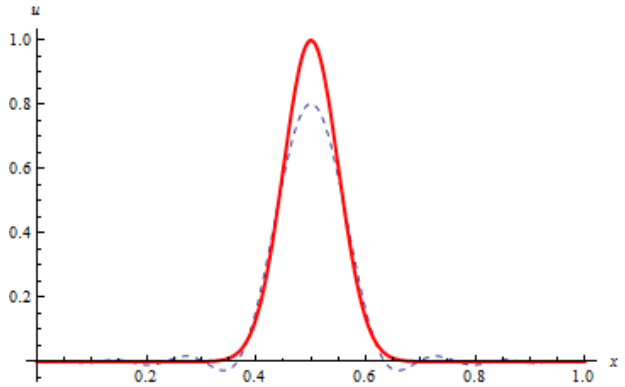
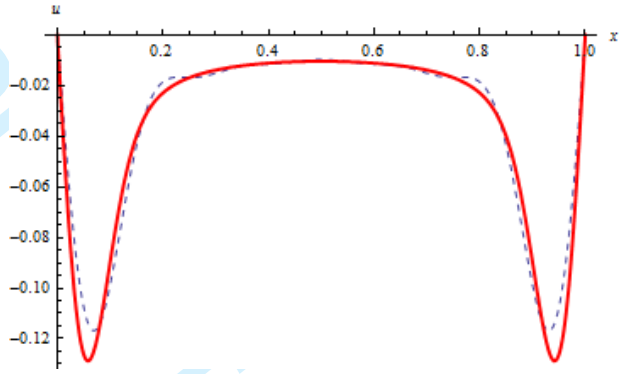


Figure 5

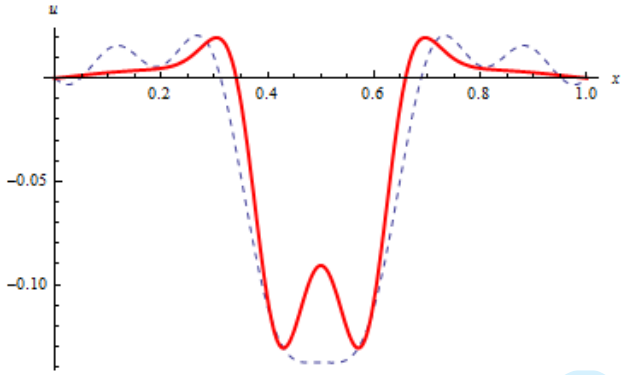
(a)



(b)



(c)



(d)

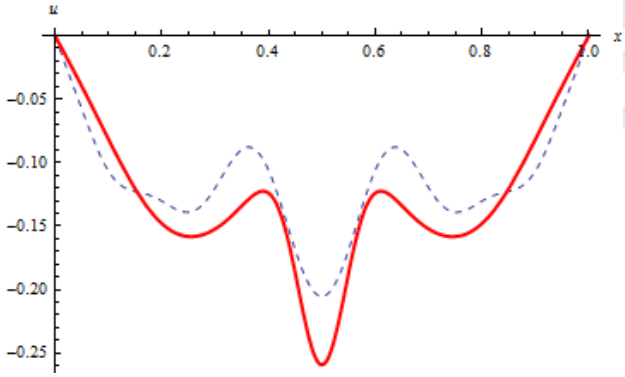


Figure 6

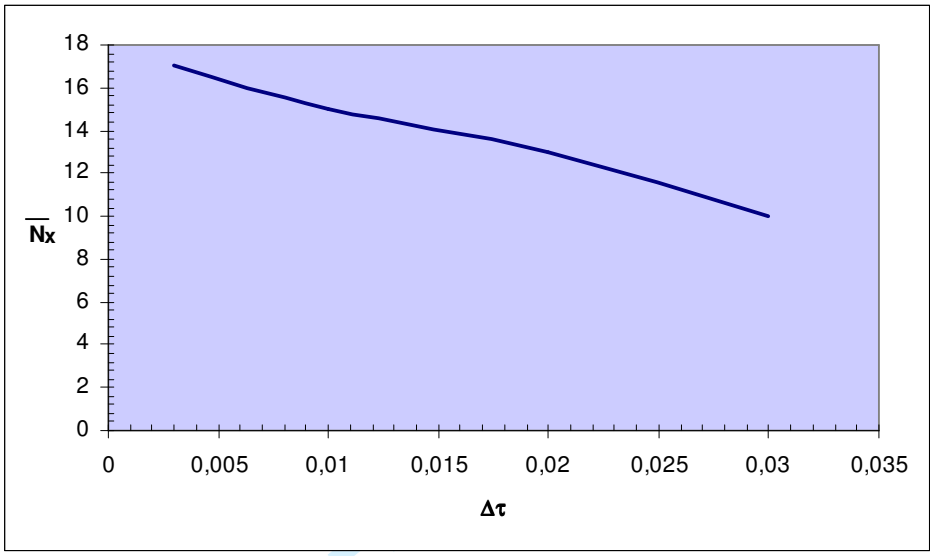


Figure 7

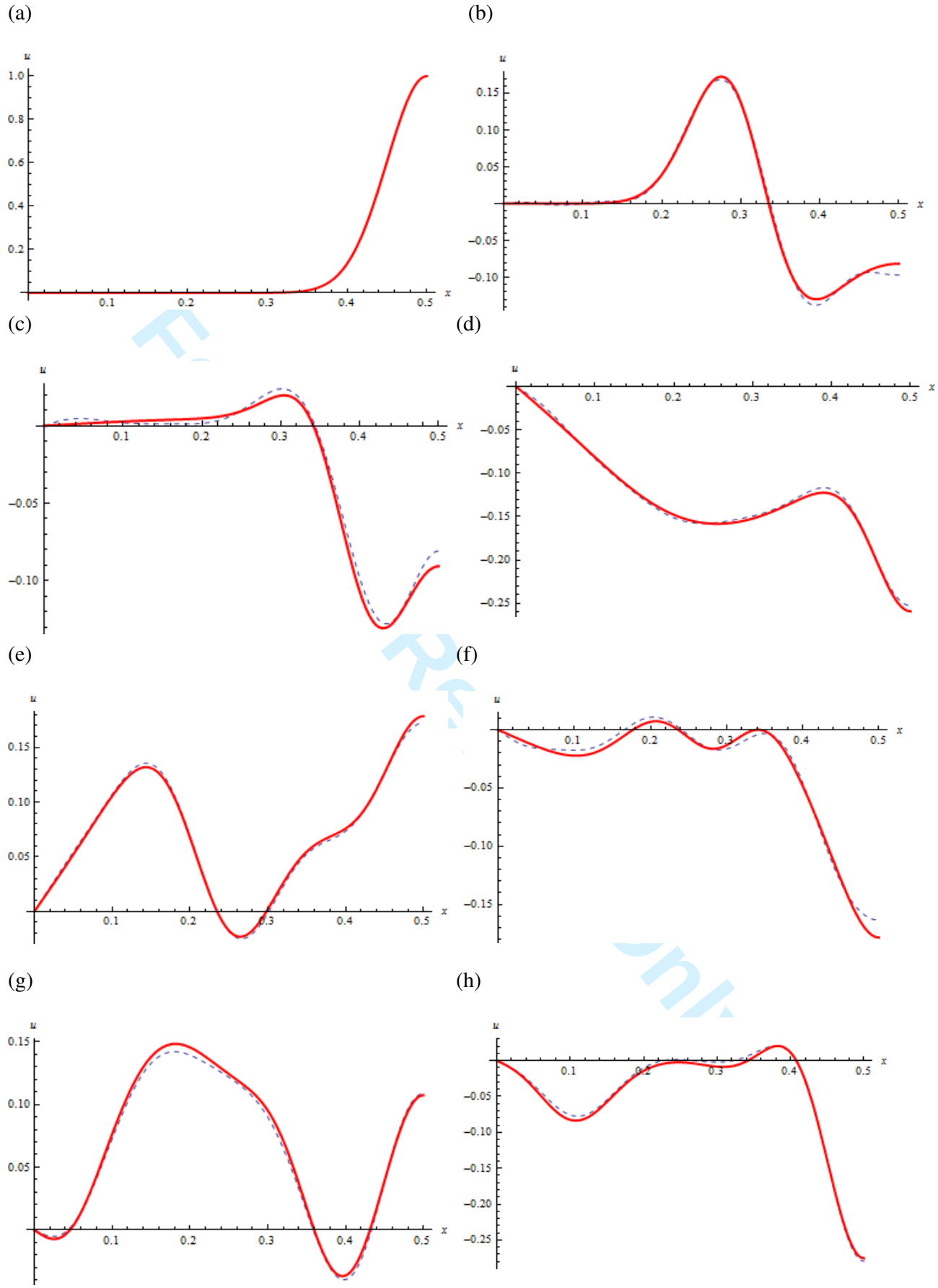


Figure 8

Table Captions

Table 1 - Natural frequencies for $N_x = N_y = 8$

$\lambda_x = 0$	error (%)	$\lambda_x = 0.5$	error (%)
4.44289	-0.0002	**4.44288**	-7E-05
7.02477	0.00055	7.0247	0.00153
8.88569	0.00085	**8.88559**	0.00205
9.97211	-0.3776	9.95205	-0.1757
11.3601	-0.2901	11.3424	-0.1344
12.3775	4.44396	12.5526	3.09168
13.3846	-0.4199	**13.3547**	-0.1956
13.5207	3.76437	13.6812	2.622
15.2612	2.84416	15.3907	2.02021
16.9312	-5.6942	17.1871	-7.2922

Table 1

Figure Captions

Figure 1 – First case: a stability graph N_x vs $\Delta\tau$ ($\lambda_x = \lambda_t = 0.5$, $N_t = 6$)

Figure 2 – First case: the error behaviour in time for u_{324} with $N_x = N_t = 6$ and (a) $\Delta\tau = 0.35$ (b) $\Delta\tau = 0.25$ (c) $\Delta\tau = 0.1$

Figure 3 – First case: *sectional plot* with $N_x = N_t = 6$, $\lambda_x = \lambda_t = 0.5$, $\Delta\tau = 0.25$ at y=0.5 and (a) t = 1 (b)
t = 10 (c) t = 19 (d) t = 39.9 (dashed line - approximated solution, thick line - exact solution)

Figure 4 – First case: *sectional plot* with $N_x = 10$, $N_t = 6$, $\lambda_x = \lambda_t = 0.5$, $\Delta\tau = 0.1$ at y=0.5 and (a) t = 1 (b) t = 10 (c) t = 19 (d) t = 39.9 (dashed line - approximated solution, thick line - exact solution)

Figure 5 – First case: *sectional plot* with $N_x = 16$, $N_t = 6$, $\lambda_x = \lambda_t = 0.5$, $\Delta\tau = 0.02$ at y=0.5 and (a) t = 1 (b) t = 10 (c) t = 19 (d) t = 39.9 (dashed line - approximated solution, thick line - exact solution)

Figure 6 – Second case: *sectional plot* with $N_x = 19$, $N_t = 6$, $\lambda_x = \lambda_t = 0.5$, $\Delta\tau = 0.01$ at y=0.5 and (a) t = 0 (b) t = 0.5 (c) t = 0.9 (d) t = 1 (dashed line – approximated solution, thick line – exact solution)

Figure 7 - Second case: a stability graph $\overline{N}_x$ vs $\Delta\tau$ ($\lambda_x = \lambda_t = -1.4$, $N_t = 10$)

Figure 8 – Second case: *sectional plot* with $\overline{N}_x = 15$, $N_t = 10$, $\lambda_x = \lambda_t = -1.4$, $\Delta\tau = 0.01$ at y=0.5 and (a) t = 0 (b) t = 0.2 (c) t = 0.9 (d) t = 1 (e) t = 2 (f) t = 3 (g) t = 4 (h) t = 5 (dashed line – approximated solution, thick line – exact solution)

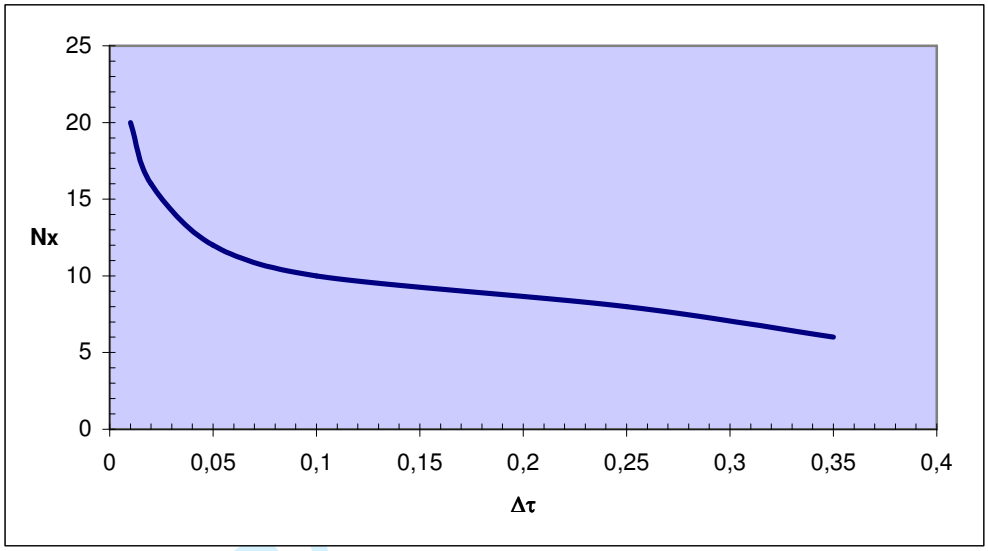
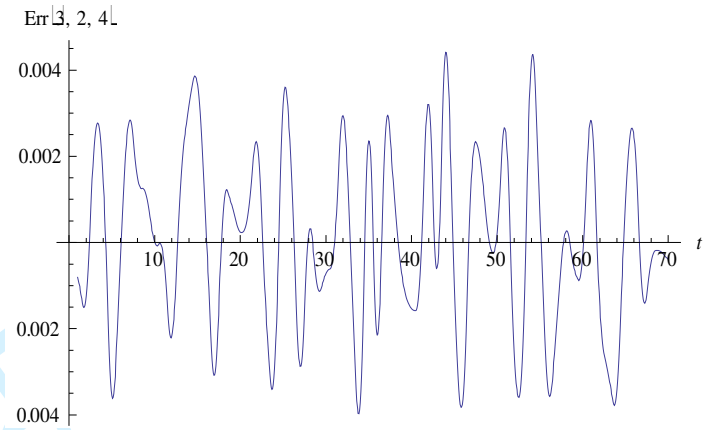
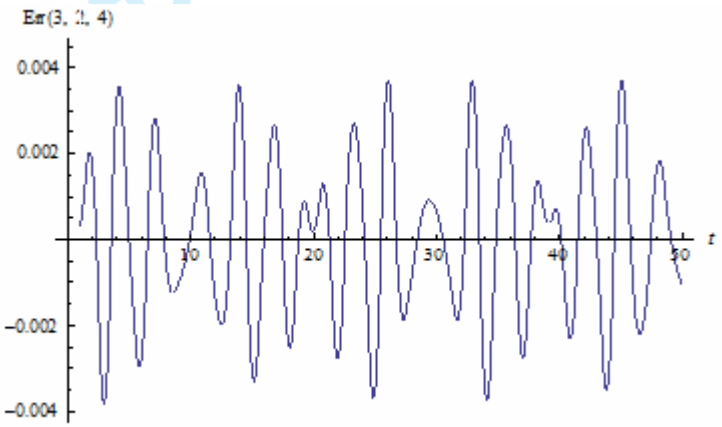


Figure 1

(a)


Err[3, 2, 4]
0.004
0.002
0.002
0.004
10
20
30
40
50
60
70
t

(b)


Err(3, 2, 4)
0.004
0.002
−0.002
−0.004
10
20
30
40
50
t

(c)

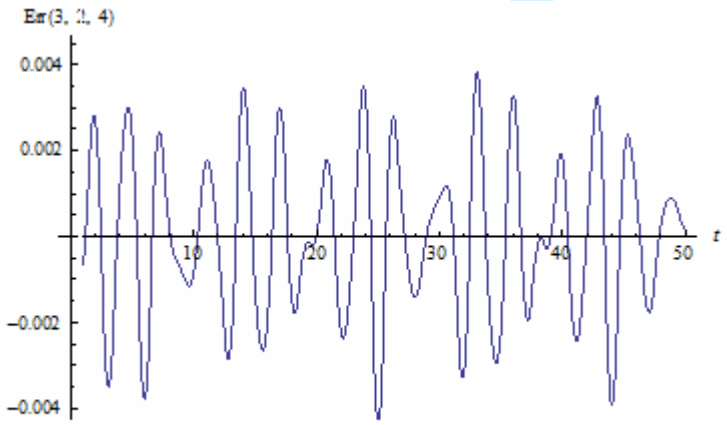
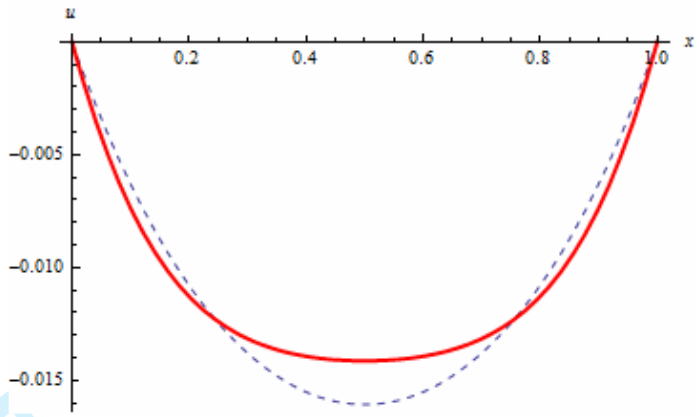
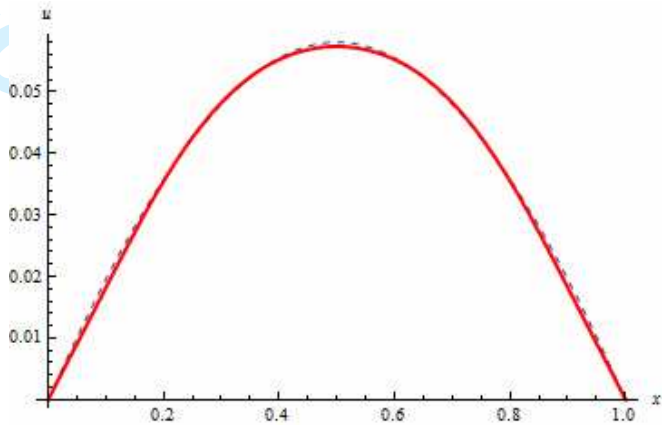

Err(3, 2, 4)
0.004
0.002
−0.002
−0.004
10
20
30
40
50
t

Figure 2

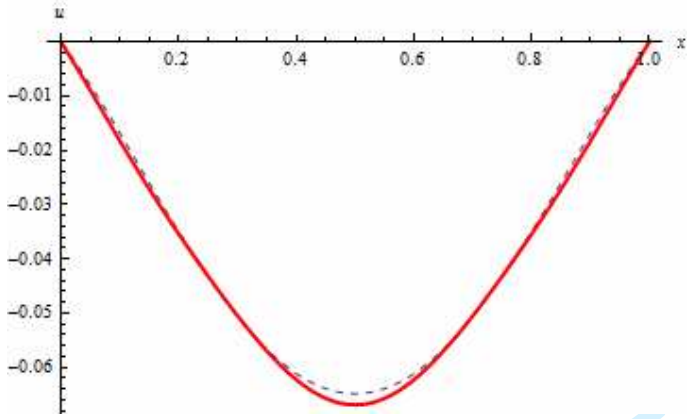
(a)



(b)



(c)



(d)

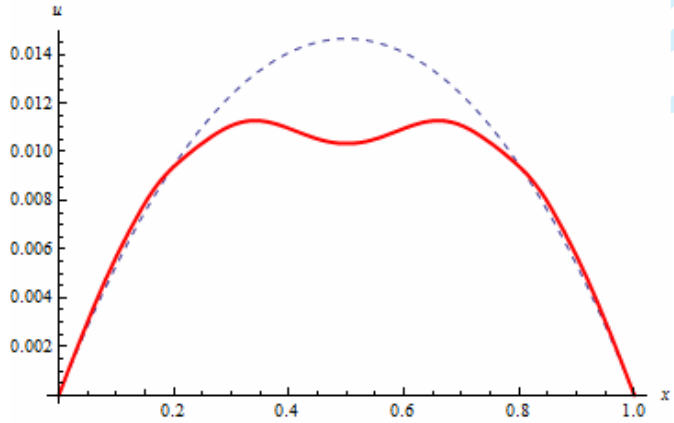
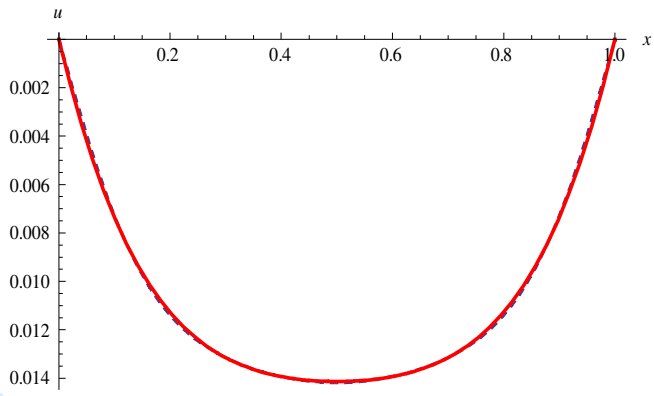
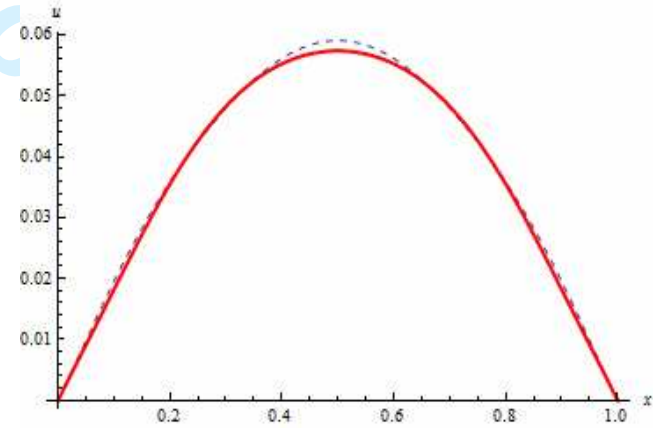


Figure 3

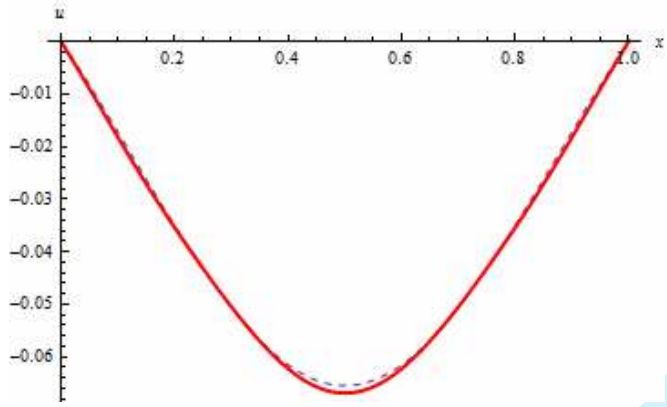
(a)



(b)



(c)



(d)

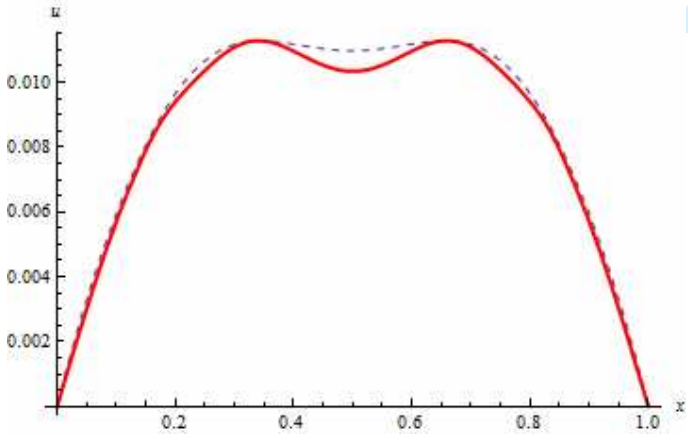
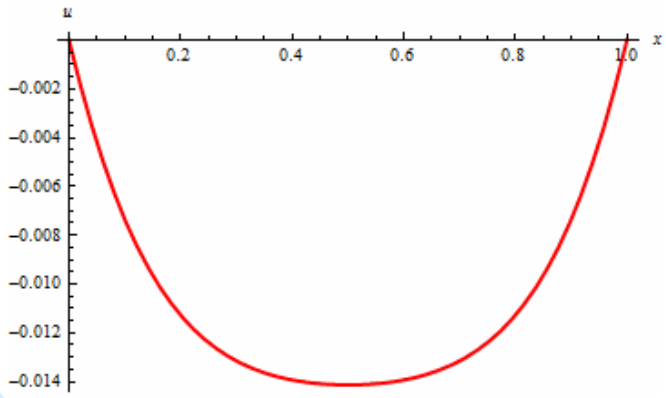
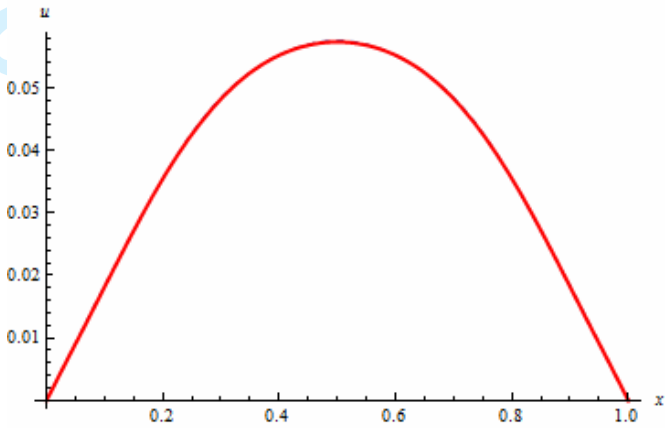


Figure 4

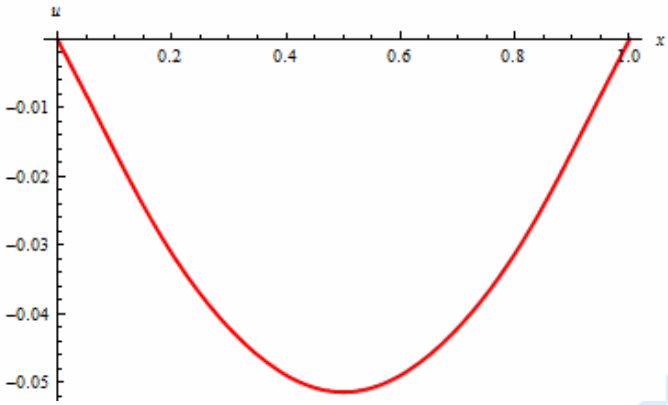
(a)



(b)



(c)



(d)

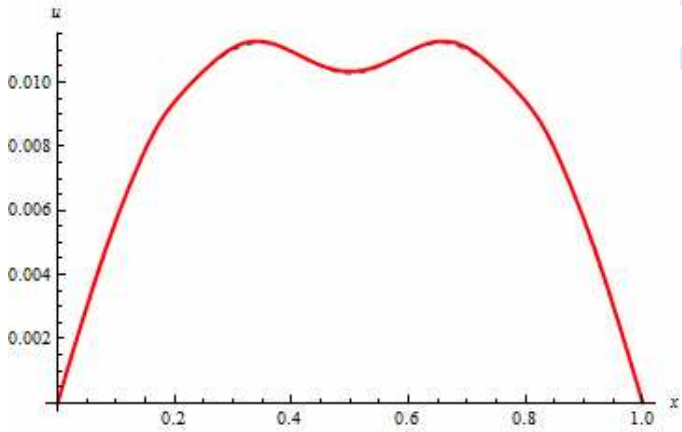
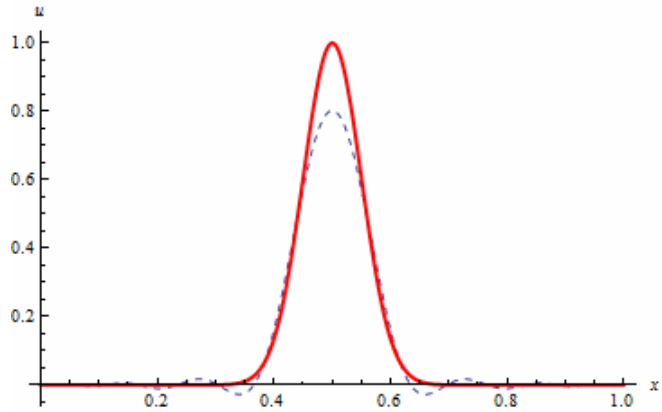
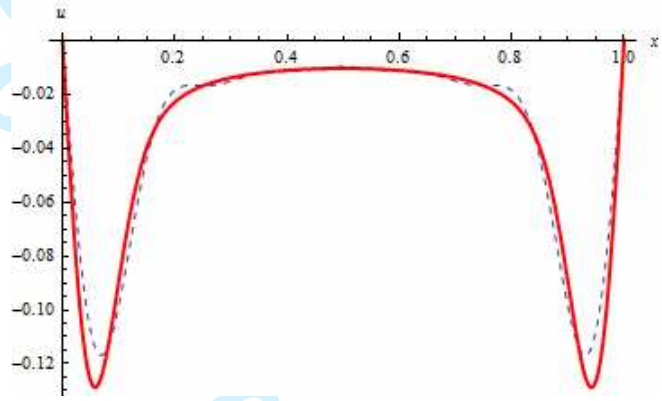


Figure 5

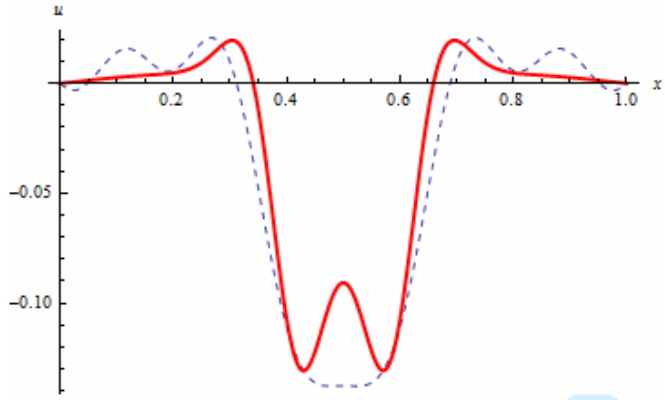
(a)



(b)



(c)



(d)

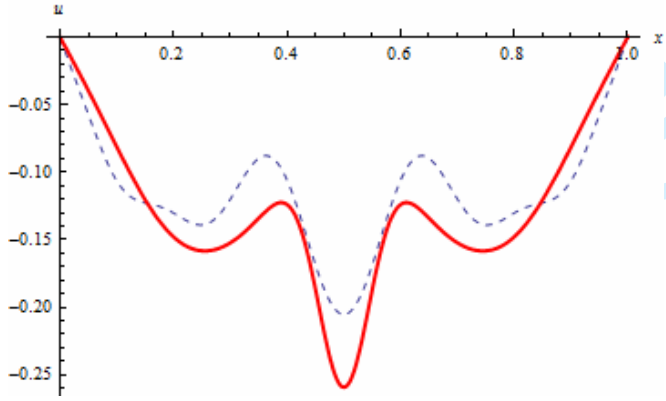


Figure 6

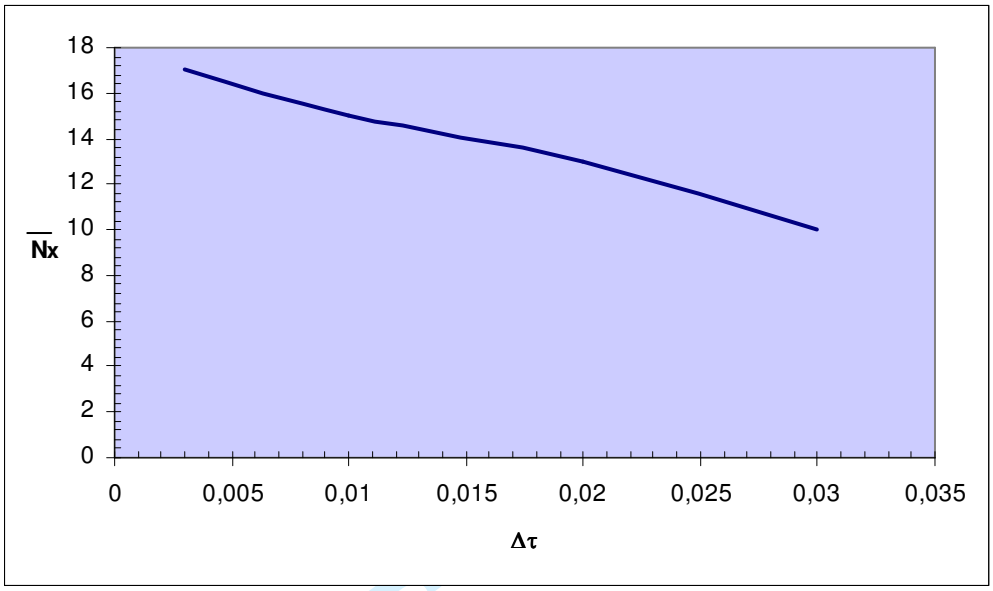


Figure 7

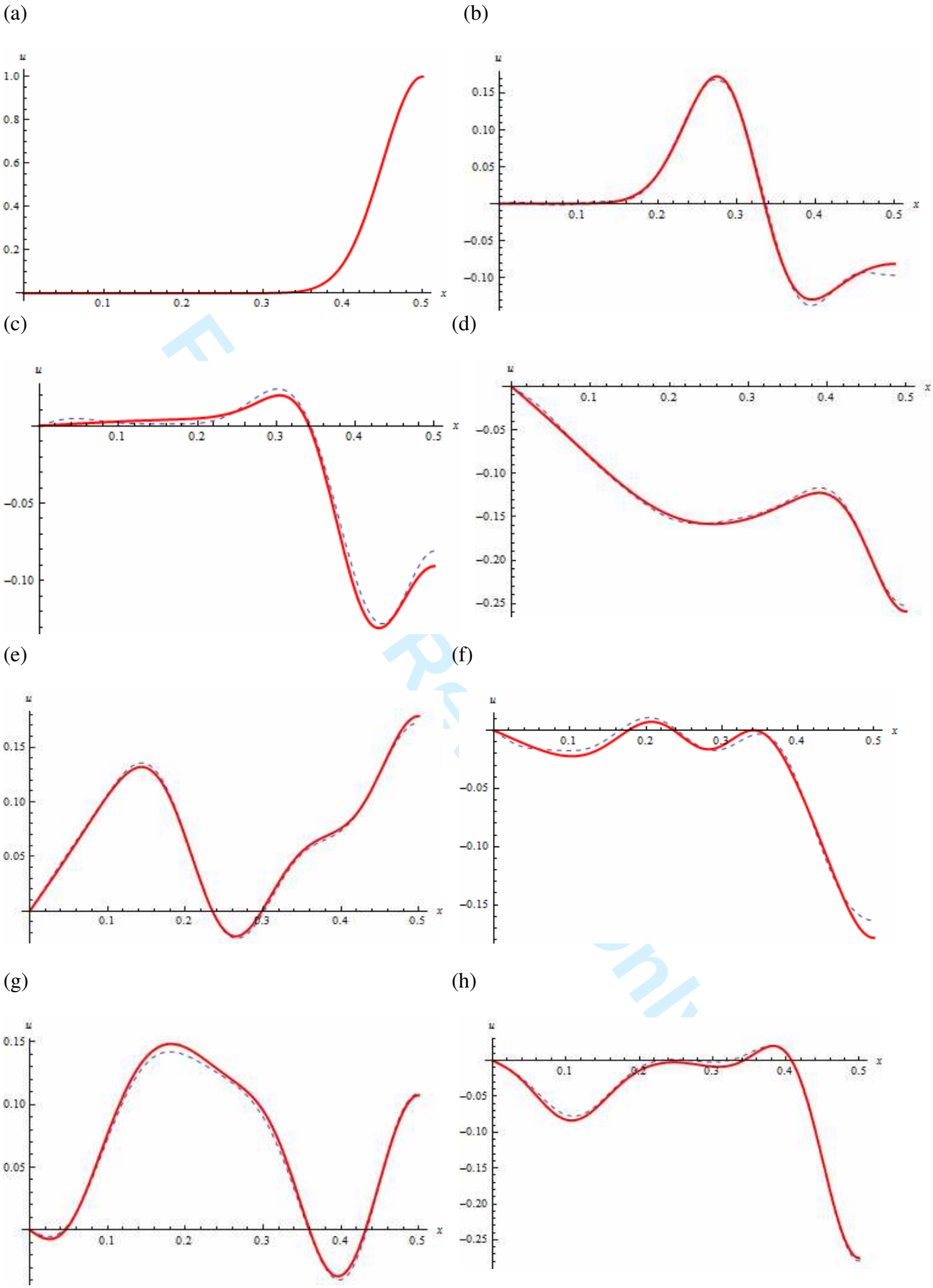


Figure 8