

## A Simple, Fast, and Accurate Time-Integrator for Strongly Nonlinear Dynamical Systems

T.A. Elgohary<sup>1,2</sup>, L. Dong<sup>3</sup>, J.L. Junkins<sup>2,4</sup> and S.N. Atluri<sup>1,4</sup>

**Abstract:** In this study, we consider Initial Value Problems (IVPs) for strongly nonlinear dynamical systems, and study numerical methods to analyze short as well as long-term responses. Dynamical systems characterized by a system of second-order nonlinear ordinary differential equations (ODEs) are recast into a system of nonlinear first order ODEs in mixed variables of positions as well as velocities. For each discrete-time interval Radial Basis Functions (RBFs) are assumed as trial functions for the mixed variables in the time domain. A simple collocation method is developed in the time-domain, with Legendre-Gauss-Lobatto nodes as RBF source points as well as collocation points. Three numerical examples are provided to compare the present algorithm with explicit as well implicit methods in terms of accuracy, required size of time-interval (or step) and computational cost. The present algorithm is compared against, the second order central difference method, the classical Runge-Kutta method, the adaptive Runge-Kutta-Fehlberg method, the Newmark- $\beta$  and the Hilber-Hughes-Taylor methods. First the highly nonlinear Duffing oscillator is analyzed and the solutions obtained from all algorithms are compared against the analytical solution for free oscillation at long times. A Duffing oscillator with impact forcing function is next solved. Solutions are compared against numerical solutions from state of the art *ODE45* numerical integrator for long times. Finally, a nonlinear 3-DOF system is presented and results from all algorithms are compared against *ODE45*. It is shown that the present *RBF-Coll* algorithm is very simple, efficient and very accurate in obtaining the solution for the nonlinear IVP. Since other presented methods require a much smaller step size and higher computational cost, the proposed algorithm is advantageous and has promising applications in solving nonlinear dynamical systems. The extension of the present algorithm to orbit propagation problems with perturbations, will be pursued in our future studies. Issues of numerical stability for various time-integrators will also be explored in future studies.

---

<sup>1</sup> Center for Aerospace Research & Education, University of California, Irvine, CA.

<sup>2</sup> Department of Aerospace Engineering, Texas A&M University, College Station, TX.

<sup>3</sup> Department of Engineering Mechanics, Hohai University, China. Email: dong.leiting@gmail.com

<sup>4</sup> Texas A&M University Institute for Advanced Study, College Station, TX.

**Keywords:** Radial Basis Function, Collocation, Implicit Methods, Explicit Methods, Numerical Integration.

## 1 Introduction

A second-order nonlinear dynamic system can be generally recast into a system of first-order ODEs as:

$$\begin{cases} \dot{\mathbf{x}}_1 = \mathbf{g}_1(\mathbf{x}_1, \mathbf{x}_2, \mathbf{f}, t) \equiv \mathbf{x}_2 \\ \dot{\mathbf{x}}_2 = \mathbf{g}_2(\mathbf{x}_1, \mathbf{x}_2, \mathbf{f}, t) \end{cases}, \quad t_0 \leq t \leq t_F \quad (1)$$

which can be simply rewritten as:

$$\dot{\mathbf{x}} = \mathbf{g}(\mathbf{x}, \mathbf{f}, t), \quad t_0 \leq t \leq t_F \quad (2)$$

where  $\mathbf{x}$  is the vector of mixed variables,  $\mathbf{x} \equiv [\mathbf{x}_1, \mathbf{x}_2]^T$ ,  $\dot{\mathbf{x}}_1 = \mathbf{x}_2$ ,  $\mathbf{f}$  is the force applied to the system. For a specified set of initial conditions  $\mathbf{x}_0$  at  $t = t_0$ , and being given the force function  $\mathbf{f}(t)$ , the initial value problem (IVP) of Eq. 2 can be numerically integrated and solved by various explicit and implicit methods of numerical integration.

In explicit methods, the future unknown state is directly expressed in terms of the currently-known system state with an explicit formula. The simplest explicit method is the forward Euler-method:

$$\mathbf{x}(t + \Delta t) = \mathbf{x}(t) + \Delta t \mathbf{g}(\mathbf{x}(t), \mathbf{f}(t), t) \quad (3)$$

which is a first-order Taylor series expansion in the time domain.

Another explicit method, the second order central difference method presented in [Belytschko (1976); Noor and Lambiotte Jr (1979); Belytschko, Lin, and Chen-Shyh (1984)], is widely used for transient finite element analyses of large scale nonlinear structures, such as crash simulation of automobiles. In this method, the velocity  $\mathbf{x}_2$  is firstly evaluated at  $t + \frac{\Delta t}{2}$ , and then  $\mathbf{x}_1$  is obtained at  $t + \Delta t$ :

$$\begin{aligned} \mathbf{x}_2(t + \frac{\Delta t}{2}) &= \mathbf{x}_2(t - \frac{\Delta t}{2}) + \Delta t \mathbf{g}_2(\mathbf{x}(t), \mathbf{f}(t), t) \\ \mathbf{x}_1(t + \Delta t) &= \mathbf{x}_1(t) + \Delta t \mathbf{x}_2(t + \frac{\Delta t}{2}) \end{aligned} \quad (4)$$

The Runge-Kutta, *RK*, family of methods can be considered as the most widely used explicit methods for numerical integration of general dynamical systems. The first order *RK* method is simply the forward Euler-method given in Eq. 3. The classical or the 4th-order *RK* method, which evaluates the solution in 4 steps, is the

most commonly used among various *RK* methods. In [Fehlberg (1969)], adaptive step-size 4th-order *RK* methods are developed and are now known as the Runge-Kutta-Fehlberg, *RKF*, methods. Several higher order adaptive *RKF* methods, [Filippi and Gräf (1986)], are widely used for very-high accuracy applications such as orbit propagation problems, see [Montenbruck (1992); Sharp (2006)].

Implicit methods put the currently-known state and the unknown future state in a set of linear or nonlinear algebraic equations, by solving which the future state can be obtained. Backward Euler-Method is an illustration of this concept:

$$\mathbf{x}(t + \Delta t) = \mathbf{x}(t) + \Delta t \mathbf{g}(\mathbf{x}(t + \Delta t), \mathbf{f}(t + \Delta t), t + \Delta t) \quad (5)$$

In [Newmark (1959)], Newmark introduced the Newmark- $\beta$  method based on the extended mean value theorem, which is among the most widely-used implicit methods for the numerically evaluating the dynamical response of engineering structures,

$$\begin{aligned} \mathbf{x}_1(t + \Delta t) &= \mathbf{x}(t) + \Delta t \mathbf{x}_2(t) + \frac{1}{2} \Delta t^2 [(1 - 2\beta) \mathbf{g}(\mathbf{x}(t), \mathbf{f}(t), t) \\ &\quad + 2\beta \mathbf{g}(\mathbf{x}(t + \Delta t), \mathbf{f}(t + \Delta t), t + \Delta t)] \\ \mathbf{x}_2(t + \Delta t) &= \mathbf{x}_2(t) + \Delta t [(1 - \gamma) \mathbf{g}(\mathbf{x}(t), \mathbf{f}(t), t) \\ &\quad + \gamma \mathbf{g}(\mathbf{x}(t + \Delta t), \mathbf{f}(t + \Delta t), t + \Delta t)] \end{aligned} \quad (6)$$

Typical values for  $\gamma$  and  $\beta$  are  $\gamma = 1/2$  and  $\beta = 1/4$ .

What is considered as a generalization of the Newmark- $\beta$  method is introduced in the Hilber-Hughes-Taylor or *HHT- $\alpha$*  method, [Hilber, Hughes, and Taylor (1977)]:

$$\begin{aligned} \mathbf{x}_1(t + \Delta t) &= \mathbf{x}(t) + \Delta t \mathbf{x}_2(t) + \frac{1}{2} \Delta t^2 [(1 - 2\beta) \mathbf{g}(\mathbf{x}(t), \mathbf{f}(t), t) + 2\beta \mathbf{a}(\alpha)] \\ \mathbf{x}_2(t + \Delta t) &= \mathbf{x}_2(t) + \Delta t [(1 - \gamma) \mathbf{g}(\mathbf{x}(t), \mathbf{f}(t), t) + \gamma \mathbf{a}(\alpha)] \\ \mathbf{a}(\alpha) &= (1 + \alpha) \mathbf{g}(\mathbf{x}(t + \Delta t), \mathbf{f}(t + \Delta t), t + \Delta t) - \alpha \mathbf{g}(\mathbf{x}(t), \mathbf{f}(t), t) \end{aligned} \quad (7)$$

where,  $\gamma = \frac{1-2\alpha}{2}$ ,  $\beta = (\frac{1-\alpha}{2})^2$  and  $\alpha \in [-\frac{1}{3}, 0]$ .

For all the above mentioned explicit and implicit methods, the size of time-steps plays an important role for the accuracy of computational results. Generally speaking, numerical stability is not guaranteed for explicit methods, thus a much smaller time step is necessary for explicit methods to obtain an accurate solution. On the other hand, because an implicit method requires the solution of set of linear/nonlinear algebraic equations in each time step, the computational burden/time of implicit methods in each time step is much higher than explicit methods. The reader is referred to [Subbaraj and Dokainish (1989); Dokainish and Subbaraj (1989)]

for a comprehensive review of such methods with applications in computational structural dynamics.

Besides all of the above-mentioned widely-used numerical integrators, Eq. 2 as a set of first order ODEs can be numerically solved by a wide spectrum of computational methods, see [Atluri (2005)]. These methods in the time domain, such as collocation, finite volume, Galerkin, MLPG, are all essentially branches from the same tree, using the concept of weighted-residual weak-forms, and with different trial and test functions, see [Dong, Alotaibi, Mohiuddine, and Atluri (2014)]. Among these methods, collocation is one of the simplest and the most efficient ones. In [Dai, Schnoor, and Atluri (2012)], a collocation method with harmonic trial functions was developed for studying the periodic responses of nonlinear Duffing oscillators and aeroelastic systems. In [Elgohary, Dong, Junkins, and Atluri (2014b)], a time-domain collocation method with Radial Basis Functions as trial function was also developed for direct solution of various time-domain inverse problems such as optimal control and orbital transfer.

In this study, the previous work of [Elgohary, Dong, Junkins, and Atluri (2014b)] is further recast as a general time-domain step-wise numerical integrator, to numerically integrate the IVP in Eq. 2 for arbitrary nonlinear systems. The algorithm is compared against the central difference *Explicit* method, the classical Runge-Kutta, *RK4*, method, the *Newmark- $\beta$*  method, and the *HHT- $\alpha$*  method in terms of accuracy, step size and computational time, using various free-vibration, forced vibration, impact load problems of single-DOF as well as coupled multi-DOF Duffing oscillators. These numerical examples clearly show the advantages of the present *RBF-Coll* algorithm which enables a much larger time step, and produces high solution accuracy while maintaining a relatively low computational cost.

## 2 Radial Basis Function & Collocation Time Integrator

Radial Basis Functions (RBFs) are real-valued functions with values depending on the distance from a source point,  $\phi(\mathbf{x}, \mathbf{x}_c) = \phi(\|\mathbf{x} - \mathbf{x}_c\|) = \phi(r)$ . Some of the commonly used types for RBFs are as follows, [Buhmann (2003)]:

$$\begin{aligned} \phi(r) &= e^{-(cr)^2} && \text{Gaussian} \\ \phi(r) &= \frac{1}{1 + (cr)^2} && \text{Inverse quadratic} \\ \phi(r) &= \sqrt{1 + (cr)^2} && \text{Multiquadric} \\ \phi(r) &= \frac{1}{\sqrt{1 + (cr)^2}} && \text{Inverse multiquadric} \end{aligned} \tag{8}$$

where  $c > 0$  is a shaping parameter.

In the currently-developed *RBF-Coll* numerical integrator, the time domain of interest, for long-time responses, is divided into a set of time steps [or time intervals] with  $t_0, t_1, t_2, \dots, t_F$ , with  $t_i = t_{i-1} + \Delta t$ . For each time step or interval  $t_{i-1} \leq t \leq t_i$ , the trial functions in the time-domain are expressed as a linear combination of Radial Basis Functions. In this study, Gaussian functions are used because of its simplicity. Following the previous work of [Elgohary, Dong, Junkins, and Atluri (2014b)], the source points of RBFs are located at  $N$  Legendre-Gauss-Lobatto (LGL) nodes within each time step, i.e.  $t_i^1 = t_{i-1}, t_i^2, t_i^3, \dots, t_i^N = t_i$ , where the subscript denotes the time step, and the superscript denotes the number of LGL node.

The state of the dynamical system at each LGL node can therefore be expressed in terms of the undetermined coefficients of RBF basis functions :

$$\mathbf{x}(t_i^j) = \sum_{k=1}^N \phi(t_i^j, t_i^k) \mathbf{a}_k, \quad i = 1, \dots, N \quad (9)$$

In matrix-vector form Eq. 9 can be rewritten as,

$$\mathbf{X} = \Phi \mathbf{A} \quad (10)$$

where  $\Phi$  represents the matrix of basis functions,  $\mathbf{A}$  is the vector of undetermined coefficients, and  $\mathbf{X}$  is the vector of unknown states at each LGL node. The time-differentiation of Eq. 10 can be then expressed as,

$$\dot{\mathbf{X}} = \dot{\Phi} \mathbf{A} \quad (11)$$

Hence, combining Eq. 10 and Eq. 11,  $\dot{\mathbf{X}}$  is related to  $\mathbf{X}$  by,

$$\dot{\mathbf{X}} = D \mathbf{X}, \quad D \equiv \dot{\Phi} \Phi^{-1} \quad (12)$$

where,  $D$  is called the derivative matrix, which is numerically evaluated from the RBF basis functions and their time-derivatives evaluated at the LGL nodes.

With the derivative matrix being defined, and with  $\mathbf{x}_{i-1} = \hat{\mathbf{x}}_{i-1}$  being known from the previous time step, we collocate Eq. 2 at  $t_i^2, t_i^3, \dots, t_i^N$ , and collocate the initial condition at  $t_i^1$ , leading to the following set of discretized equations:

$$\begin{aligned} \mathbf{x}_i^1 - \hat{\mathbf{x}}_{i-1} &= 0 \\ D \mathbf{x}_j - \mathbf{g}(\mathbf{x}_j, \mathbf{f}, t_j) &= 0, \quad j = 2, 3, \dots, N \end{aligned} \quad (13)$$

The algorithm as presented above is very simple and easy to implement. The set of nonlinear algebraic equations is solved with the classical Newton's method in this study, whereas other Jacobian-inverse-free methods can also be applied as in [Liu,

Yeih, Kuo, and Atluri (2009); Liu and Atluri (2012); Elgohary, Dong, Junkins, and Atluri (2014a)]. With Eq. 13 being solved, the unknown states at each LGL node, as well as the unknown state at the end of this time interval or step ( $x_i$ ) are obtained. In this way, the states at each time step within the entire time history, i.e.  $t_0, t_1, t_2, \dots, t_F$  can be numerically evaluated by applying the above numerical algorithm in a sequential procedure.

### 3 Numerical Results

In this section numerical experiments are presented for three cases of nonlinear dynamical systems, namely: the highly nonlinear unforced single Duffing oscillator, a single Duffing oscillator subjected to an impact load, and a 3-Degree-of-Freedom (3-DOF) coupled nonlinear Duffing oscillator system subjected to a harmonic load. For each case, the present RBF-Collocation, *RBF-Coll*, algorithm is compared against the 4th order Runge-Kutta (*RK4*), the *Explicit* method, [Belytschko (1976); Belytschko, Lin, and Chen-Shyh (1984)], the *Newmark- $\beta$*  method, [Newmark (1959)] and the *HHT- $\alpha$*  method, [Hilber, Hughes, and Taylor (1977)]. A table is presented to show the time step  $\Delta t$ , the Root-Mean-Square (RMS) error of the position state and the computation time for each case. Plots of the error time history normalized by the response amplitude, Eq. 14, are also presented for each case to demonstrate the accuracy of each algorithm. Furthermore, the analytical solution and the solutions obtained by various numerical methods are plotted to highlight the magnitudes of the errors for each algorithm.

$$\Delta x(t) \equiv \left\| \frac{x(t)_{\text{Ana}} - x(t)_{\text{Num}}}{\text{Amp}[x(t)_{\text{Ana}}]} \right\| \quad (14)$$

For each method, two numerical cases are presented. The first, denoted by the superscript 1, uses the same  $\Delta t$  as the *RBF-Coll* algorithm, for all the other algorithms. This establishes a baseline for comparing solution accuracy and computational time of all algorithms. The second, denoted by the superscript 2, explores the ability of each algorithm to achieve/maintain higher accuracy by decreasing  $\Delta t$ . As expected, and also shown, all methods will require reducing the step size to achieve a higher solution accuracy except for the *RBF-Coll* method, which can maintain the solution accuracy by increasing the number of collocation points without decreasing the step size.

#### 3.1 Free Vibrations of A Highly Nonlinear Duffing Oscillator

The unforced Duffing oscillator is described by,

$$\begin{aligned} \ddot{x} + \omega_n^2 x + \eta x^3 &= 0 & 0 \leq t \leq t_F \\ x(0) = x_0 \quad \dot{x}(0) &= \dot{x}_0 \end{aligned} \quad (15)$$

For a high hardening nonlinearity  $\eta \geq 1$ , Jacobi-elliptic functions provide an analytical solution for Eq. 15, [Cvetićanin (2013)], as,

$$x(t) = X \operatorname{cn}(\omega t + \theta, k^2) \quad (16)$$

where,  $\operatorname{cn}$  is the the Jacobi-elliptic function,  $\omega$  the frequency,  $k$  the modulus,  $X$  the amplitude, and  $\theta$  the phase angle of the function. The reader is referred to [Cvetićanin (2013)] for the detailed expression of each parameter in Eq. 16.

For the parameters values shown in Tab. 1, the analytical solution is compared against MATLAB *ODE45*, *Explicit* method, *RK4*, *Newmark- $\beta$* , *HHT- $\alpha$*  and the present *RBF-Coll* algorithm. For *ODE45*<sup>1</sup> and *ODE45*<sup>2</sup> the tolerances are set to  $10^{-6}$  and  $10^{-9}$ , respectively. For *RBF-Coll*<sup>1</sup>, the number of collocation points  $N = 7$  and shaping parameter  $c = 0.77$ . For *RBF-Coll*<sup>2</sup>,  $N = 35$  and  $c = (N - 1)/4\Delta t$ . Tab. 2 shows the step size, the positions state,  $x_1(t)$ , RMS error and the computation time for all the tested methods. Fig. 1 through Fig. 5 show the normalized state error time history for each method, Fig. 6 shows the normalized state error time history for the present *RBF-Coll* algorithm and finally Fig. 7 show the comparison of computations results by various methods in the last period of the freely vibrating duffing oscillator.

Table 1: Parameters for the Freely-Vibrating Duffing System

Parameter	Value
$t_0$	0
$t_F$	500
$x_1(0)$	1
$x_2(0)$	0
$\omega_n$	1
$\eta$	1
Period, $P$	4.77

Given the nonlinear oscillator period shown in Tab. 2, the numerical integration is performed for slightly above a total of 100 cycles. Clearly, all the other presented numerical methods require a much smaller step size to achieve an acceptable accuracy, whereas the *RBF-Coll* algorithm can maintain high accuracy with a very large step size as demonstrated in Tab. 2 and Fig. 6. It should also be noted that *Explicit*, *Newmark- $\beta$* , and *HHT- $\alpha$*  methods perform reasonably well for a short term, but accuracy is gradually lost in the long term even if a very tiny time step is used.

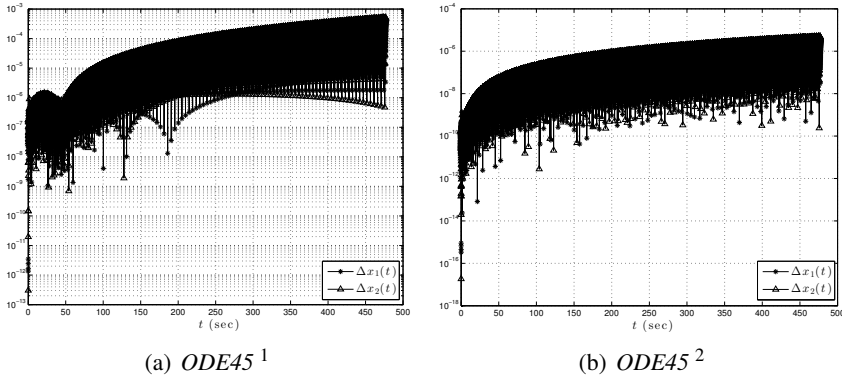


Figure 1: Normalized State Error Time History,  $ODE45^{1,2}$

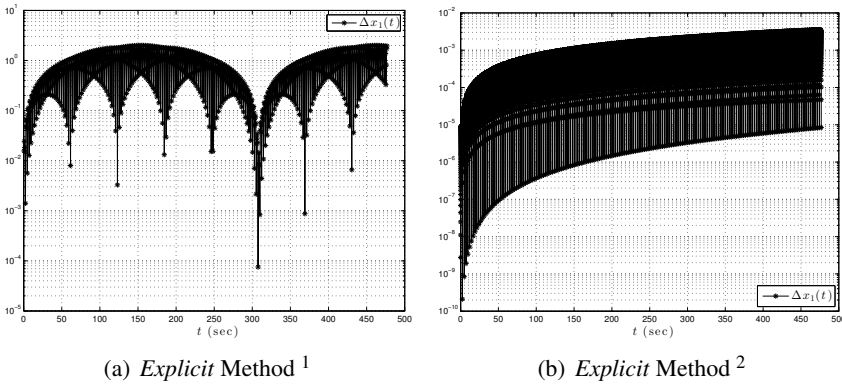


Figure 2: Normalized Position Error Time History,  $Explicit\ Method^{1,2}$

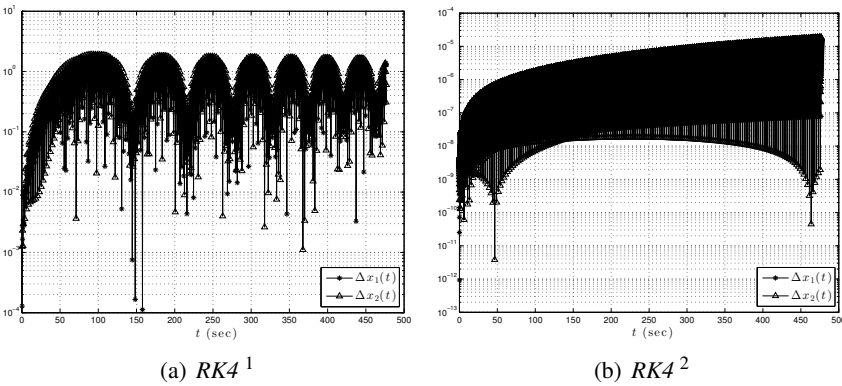
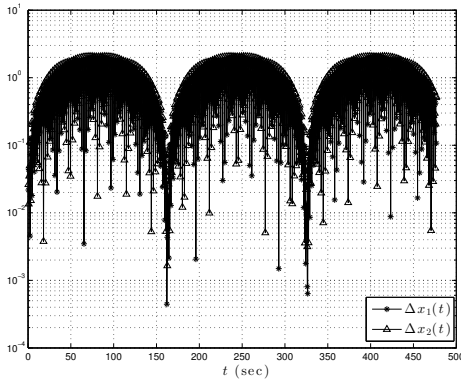
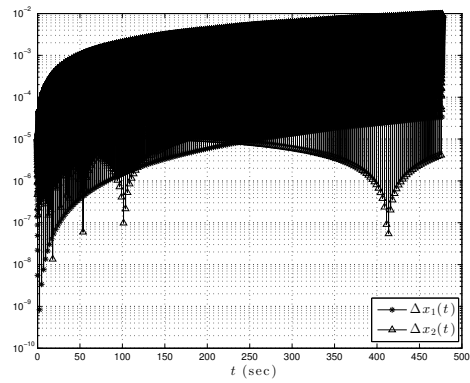


Figure 3: Normalized State Error Time History,  $RK4^{1,2}$



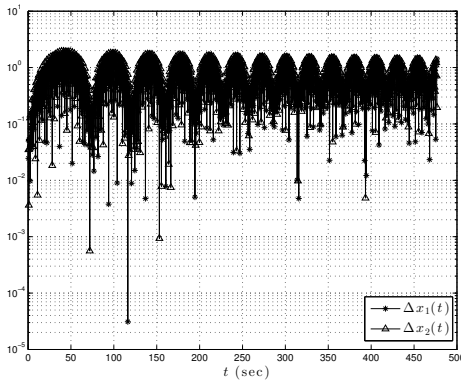


(a)  $Newmark-\beta^1$

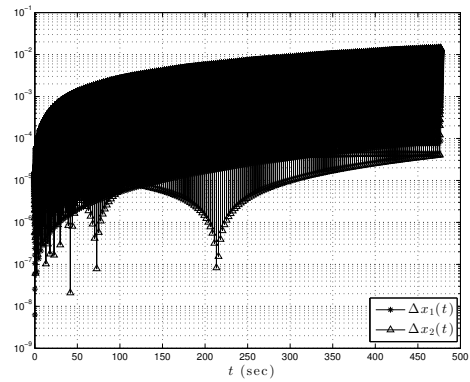


(b)  $Newmark-\beta^2$

Figure 4: Normalized State Error Time History,  $Newmark-\beta^{1,2}$



(a)  $HHT-\alpha^1$



(b)  $HHT-\alpha^2$

Figure 5: Normalized State Error Time History,  $HHT-\alpha^{1,2}$

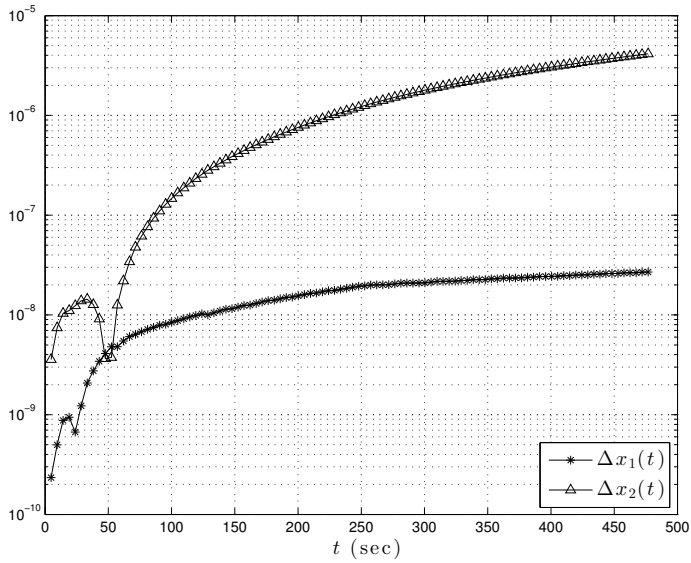
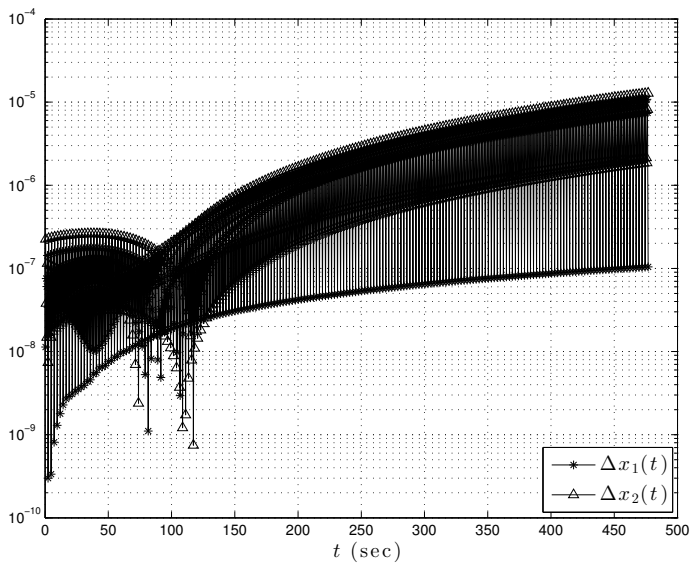
(a) *RBF-Coll*<sup>1</sup>(b) *RBF-Coll*<sup>2</sup>Figure 6: Normalized State Error Time History, *RBF-Coll*<sup>1,2</sup>

Table 2: Comparison of Numerical Methods, Freely Vibrating Highly Nonlinear Duffing Oscillator

Method	$\Delta t$ (sec)	Position RMS Error	Simulation Time (sec)
<i>ODE45</i> <sup>1</sup>	Variable	$1.18 \times 10^{-4}$	0.48
<i>ODE45</i> <sup>2</sup>	Variable	$1.14 \times 10^{-6}$	1.37
<i>Explicit</i> <sup>1</sup>	<i>P</i> /10	0.1	0.006
<i>Explicit</i> <sup>2</sup>	<i>P</i> /500	$1.5 \times 10^{-3}$	0.2
<i>RK4</i> <sup>1</sup>	<i>P</i> /10	0.86	0.03
<i>RK4</i> <sup>2</sup>	<i>P</i> /250	$5 \times 10^{-6}$	0.56
<i>Newmark-β</i> <sup>1</sup>	<i>P</i> /10	0.998	3.62
<i>Newmark-β</i> <sup>2</sup>	<i>P</i> /500	$3.1 \times 10^{-3}$	147
<i>HHT-α</i> <sup>1</sup>	<i>P</i> /10	0.841	3.53
<i>HHT-α</i> <sup>2</sup>	<i>P</i> /500	$4.4 \times 10^{-3}$	149.1
<i>RBF-Coll</i> <sup>1</sup>	<i>P</i> /10	$3.5 \times 10^{-6}$	0.32
<i>RBF-Coll</i> <sup>2</sup>	<i>P</i>	$1.8 \times 10^{-6}$	0.19

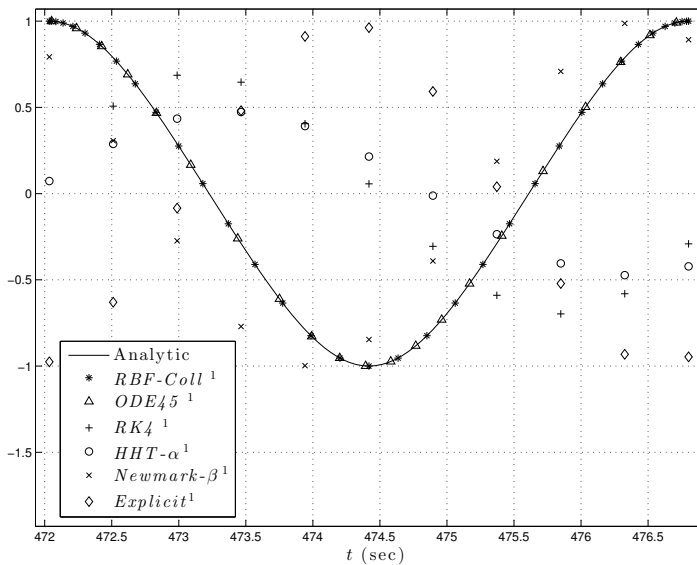


Figure 7: Solution Comparison Last Period of Integration

### 3.2 Forced Duffing Oscillator with Impact Loading

An impact triangular forcing function is applied to the Duffing oscillator in Eq. 15 as,

$$\begin{aligned} \ddot{x} + \omega_n^2 x + \eta x^3 &= f(t) \quad 0 \leq t \leq t_F \\ x(0) = x_0 \quad \dot{x}(0) &= \dot{x}_0 \\ f(t) &= \begin{cases} at & 0 \leq t \leq t_1/2 \\ a(1-t) & t_1/2 < t \leq t_1 \end{cases} \end{aligned} \tag{17}$$

where  $t_1$  defines the time interval of the applied impact force.

For the parameters shown in Tab. 3, MATLAB *ODE45* is used to obtain the numerical solution which, in the absence of an analytical solution, is treated as the reference solution for Eq. 17.

Table 3: Forced Duffing System Paramaters Values

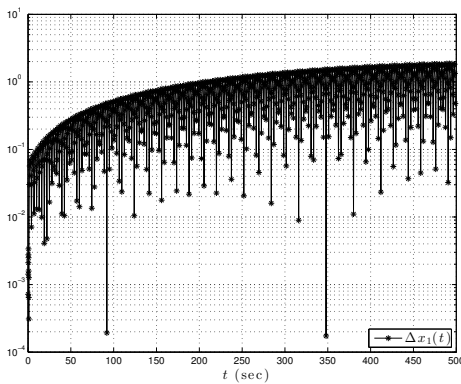
Parameter	Value
$t_0$	0
$t_1$	1
$t_F$	500
$x_1(0)$	0
$x_2(0)$	0
$\omega_n$	1
$\eta$	1
$a$	2

In order to capture the effect of the impact forcing function, the dynamical system in Eq. 17 is solved in two intervals defined by  $t_0 \leq t \leq t_1$  and  $t_1 \leq t \leq t_F$ . The *ODE45* (reference) solution is obtained by setting the tolerances in the numerical solver to the lowest possible values,  $10^{-13}$ . For *RBF-Coll*<sup>1</sup> the number of collocation points  $N = 7$  and the shaping parameter  $c = 2.5$ . And for *RBF-Coll*<sup>2</sup>,  $N = 35$  and  $c = (N - 1)/4\Delta t$ . Tab. 4 shows the comparison of step size, accuracy and computation time for various numerical methods. Fig. 8 shows the normalized state error time history for *Explicit*<sup>1</sup> and *Explicit*<sup>2</sup>. Errors associated with *RK4*<sup>1,2</sup>, *Newmark-β*<sup>1,2</sup> and *HHT-α*<sup>1,2</sup> are presented in Fig. 9, Fig. 10 and Fig. 11, respectively. Fig. 12 shows the normalized state error time history for the present *RBF-Coll* algorithm and finally, Fig. 13 shows the reference solution and solutions from *Method*<sup>1</sup> for the last 5 sec. of integration.

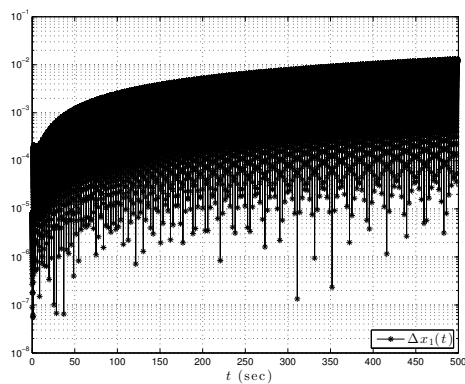
Similar to the results obtained in the previous section, the present *RBF-Coll* algorithm is shown to maintain the high solution accuracy and the low computational

Table 4: Comparison of Numerical Methods, Forced Duffing Oscillator

Method	$\Delta t_1$ (sec)	$\Delta t_2$ (sec)	Position RMS Error	Simulation Time (sec)
<i>ODE45</i>	Variable	Variable	N/A	4.85
<i>Explicit</i> <sup>1</sup>	0.1	0.5	0.41	0.01
<i>Explicit</i> <sup>2</sup>	0.005	0.006	0.0025	0.18
<i>RK4</i> <sup>1</sup>	0.1	0.5	0.399	0.03
<i>RK4</i> <sup>2</sup>	0.05	0.03	$2.88 \times 10^{-6}$	0.32
<i>Newmark-<math>\beta</math></i> <sup>1</sup>	0.1	0.5	0.457	3.6
<i>Newmark-<math>\beta</math></i> <sup>2</sup>	0.05	0.03	$1.46 \times 10^{-2}$	49.25
<i>HHT-<math>\alpha</math></i> <sup>1</sup>	0.1	0.5	0.405	3.51
<i>HHT-<math>\alpha</math></i> <sup>2</sup>	0.05	0.03	$2.41 \times 10^{-2}$	49.2
<i>RBF-Coll</i> <sup>1</sup>	0.1	0.5	$1.19 \times 10^{-6}$	0.31
<i>RBF-Coll</i> <sup>2</sup>	0.5	6	$7.98 \times 10^{-7}$	0.14

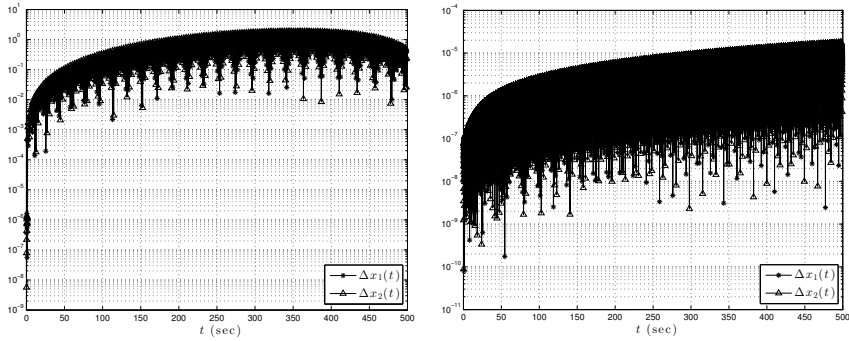


(a) *Explicit Method*<sup>1</sup>



(b) *Explicit Method*<sup>2</sup>

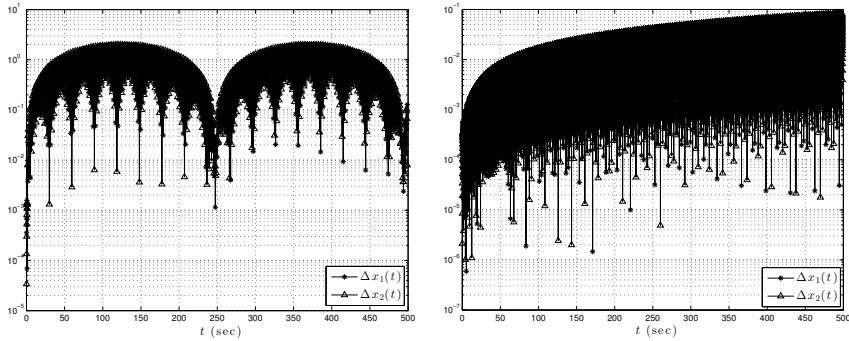
Figure 8: Normalized Position Error Time History, *Explicit Method*<sup>1,2</sup>



(a)  $RK4^1$

(b)  $RK4^2$

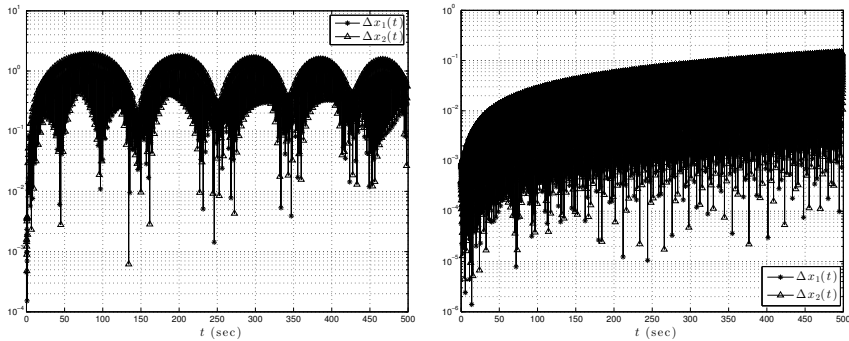
Figure 9: Normalized State Error Time History,  $RK4^{1,2}$



(a)  $Newmark-\beta^1$

(b)  $Newmark-\beta^2$

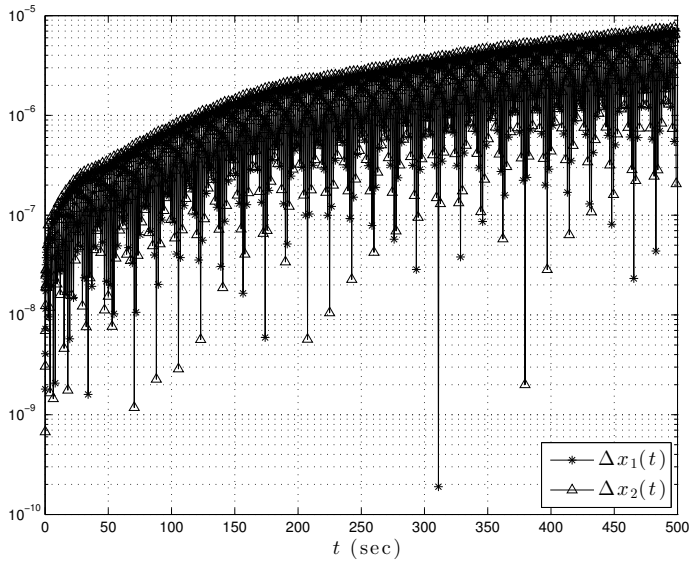
Figure 10: Normalized State Error Time History,  $Newmark-\beta^{1,2}$



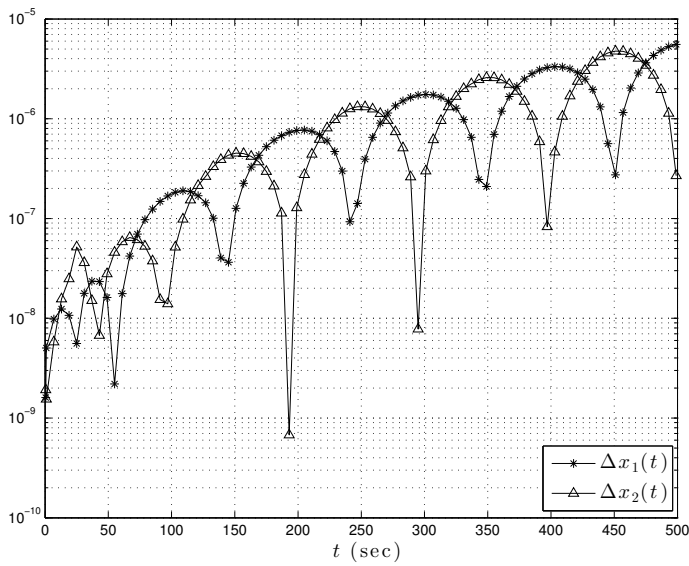
(a)  $HHT-\alpha^1$

(b)  $HHT-\alpha^2$

Figure 11: Normalized State Error Time History,  $HHT-\alpha^{1,2}$



(a) *RBF-Coll*<sup>1</sup>



(b) *RBF-Coll*<sup>2</sup>

Figure 12: Normalized State Error Time History, *RBF-Coll*<sup>1,2</sup>

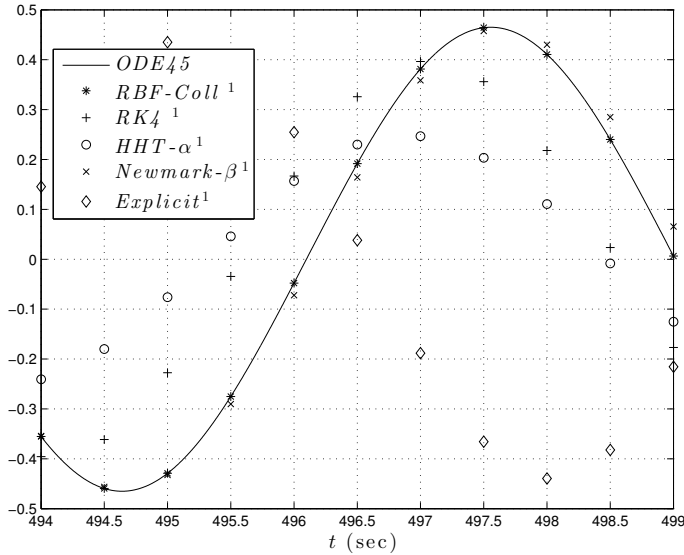


Figure 13: Computational results at the Last Period of the Free-Vibrating Duffing Oscillator

cost when compared to the other existing numerical methods by varying the step size and the number of collocation points. By comparing the results from *Method*<sup>1</sup> and *Method*<sup>2</sup>, it can be seen that the step size is the major contributor to the computational cost associated with each method. In that sense *RBF-Coll* is superior to all other methods as there is no need to take smaller steps in order to achieve higher solution accuracy. Thus, *RBF-Coll* may be a useful method to study a periodic and chaotic responses in nonlinear dynamical systems.

### 3.3 Multi Degrees of Freedom Coupled Nonlinear Dynamical System

The 3-DOF coupled nonlinear system shown in Fig. 14 is analyzed in this section. With  $F(t) = F \cos \Omega t$ , the set of 3 coupled ODEs for this nonlinear system are,

$$\begin{aligned}
 m_1 \ddot{x}_1 + c_1 \dot{x}_1 + k_1 x_1 + l_1 x_1^3 + k_2(x_1 - x_2) + c_2(\dot{x}_1 - \dot{x}_2) + l_2(x_1 - x_2)^3 &= 0 \\
 m_2 \ddot{x}_2 - c_2(\dot{x}_1 - \dot{x}_2) - k_2(x_1 - x_2) - l_2(x_1 - x_2)^3 + c_3(\dot{x}_2 - \dot{x}_3) & \\
 + k_3(x_2 - x_3) + l_3(x_2 - x_3)^3 &= F \cos \Omega t \\
 m_3 \ddot{x}_3 - c_3(\dot{x}_2 - \dot{x}_3) - k_3(x_2 - x_3) - l_3(x_2 - x_3)^3 + c_4 \dot{x}_3 + k_4 x_3 + l_4 x_3^3 &= 0
 \end{aligned}
 \tag{18}$$

The parameters selected for the 3-DOF system are given in Tab. 5. The reference



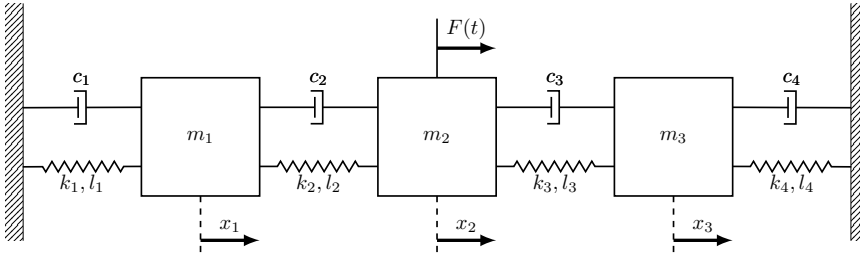


Figure 14: 3-DOF Nonlinear System

solution for this coupled nonlinear system, as in the previous section, is obtained with MATLAB *ODE45* and shown in Fig. 15. The comparison between various

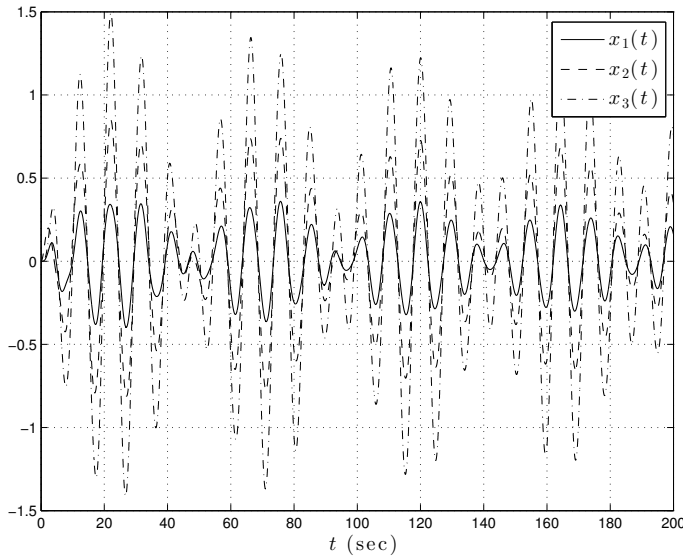


Figure 15: Dynamical response of the 3-DOF System

numerical integrators is given in Tab. 6. For *RBF-Coll*<sup>1</sup>, the number of collocation points  $N = 7$  and the shaping parameter  $c = 0.35$  and for *RBF-Coll*<sup>2</sup>,  $N = 35$  and  $c = 0.39$ . Fig. 16 through Fig. 22 show the normalized state error time history for *Explicit*<sup>1,2</sup>, *RK4*<sup>1,2</sup>, *Newmark- $\beta$* <sup>1,2</sup> and *HHT- $\alpha$* <sup>1,2</sup>. The *RBF-Coll* normalized state error time history is shown in Fig. 24. Finally, Fig. 25 shows the reference solution and *Method*<sup>1</sup> solutions for the last 10 sec. of integration.

Table 5: Parameters for the 3-DOF Coupled System

Parameter	Value
$t_0$	0
$t_F$	200
$\mathbf{x}_0$	$\mathbf{0}$
$m_1$	2
$m_2$	1
$m_3$	0.5
$c_1$	0
$c_2$	0.05
$c_3$	0
$c_4$	0
$k_1$	2
$k_2$	1
$k_3$	0.5
$k_4$	0
$l_1$	0.2
$l_2$	0
$l_3$	0
$l_4$	0
$F$	0.2
$\Omega$	$\sqrt{0.5}$

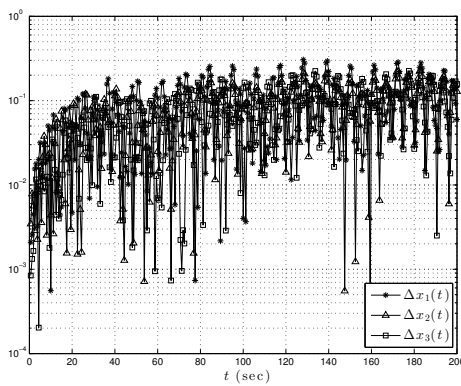
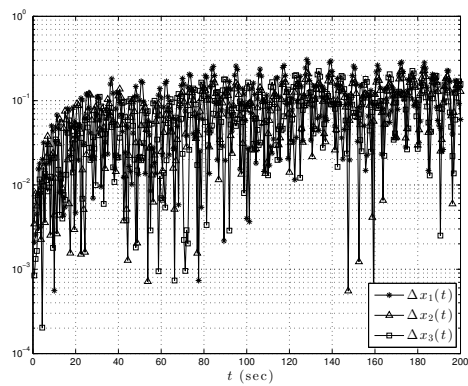
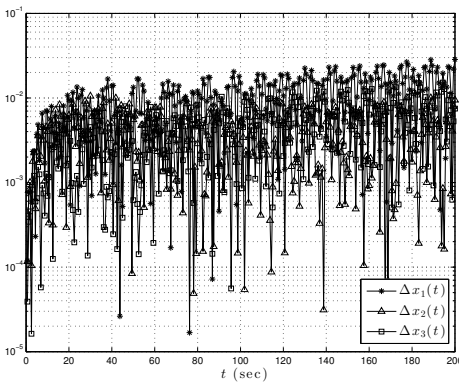
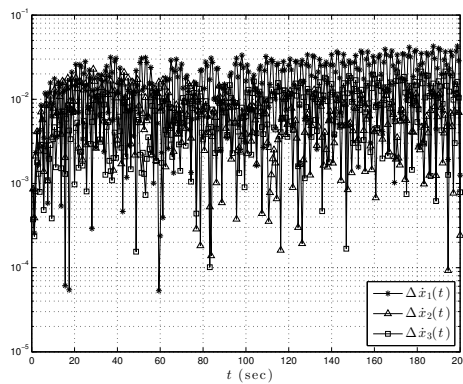
(a) *Explicit Method*<sup>1</sup>(b) *Explicit Method*<sup>2</sup>Figure 16: Normalized Position Error Time History, *Explicit Method*<sup>1,2</sup>

Table 6: Comparison of Numerical Methods, 3-DOF System

Method	$\Delta t$ (sec)	$\mathbf{x}_1(t)$ RMS Error	Simulation Time (sec)
<i>ODE45</i>	Variable	N/A	1.64
<i>Explicit</i> <sup>1</sup>	0.625	0.131	0.034
<i>Explicit</i> <sup>2</sup>	0.00625	$1.44 \times 10^{-5}$	2.49
<i>RK4</i> <sup>1</sup>	0.625	$4.6 \times 10^{-3}$	0.017
<i>RK4</i> <sup>2</sup>	0.0625	$5.1 \times 10^{-7}$	0.1
<i>Newmark-<math>\beta</math></i> <sup>1</sup>	0.625	$6.64 \times 10^{-2}$	1.42
<i>Newmark-<math>\beta</math></i> <sup>2</sup>	0.0625	$1.2 \times 10^{-3}$	12.76
<i>HHT-<math>\alpha</math></i> <sup>1</sup>	0.625	$7.6 \times 10^{-2}$	1.42
<i>HHT-<math>\alpha</math></i> <sup>2</sup>	0.0625	$1.5 \times 10^{-3}$	13.15
<i>RBF-Coll</i> <sup>1</sup>	0.625	$2.01 \times 10^{-8}$	0.282
<i>RBF-Coll</i> <sup>2</sup>	20	$4.5 \times 10^{-9}$	0.095



(a) Normalized Position Error



(b) Normalized Velocity Error

Figure 17: Normalized State Error Time History, *RK4*<sup>1</sup>

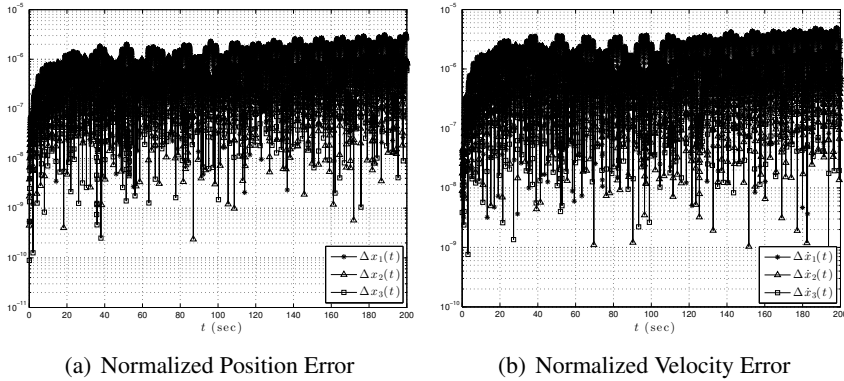


Figure 18: Normalized State Error Time History,  $RK4^2$

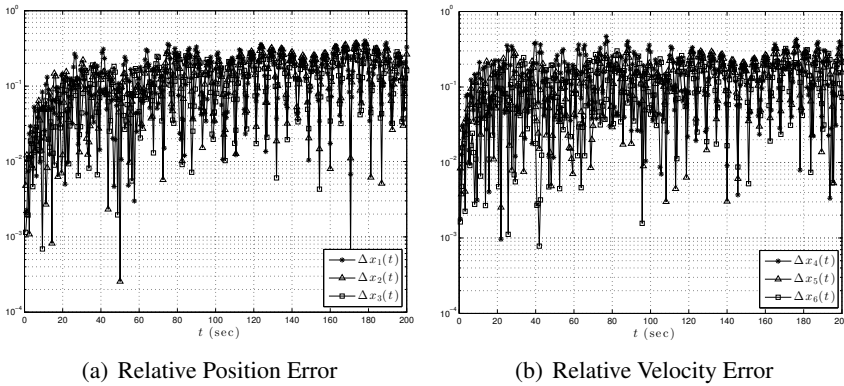


Figure 19: Normalized State Error Time History,  $Newmark-\beta^1$

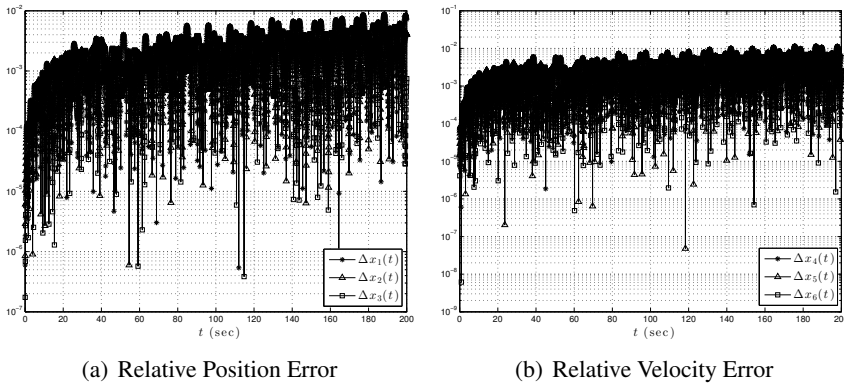
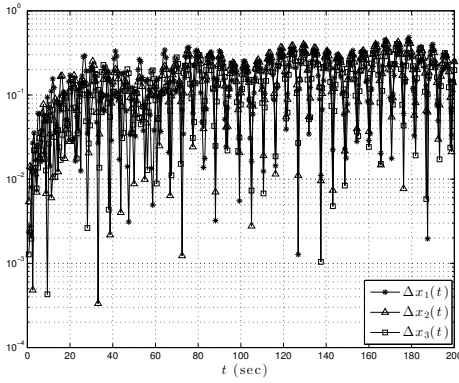
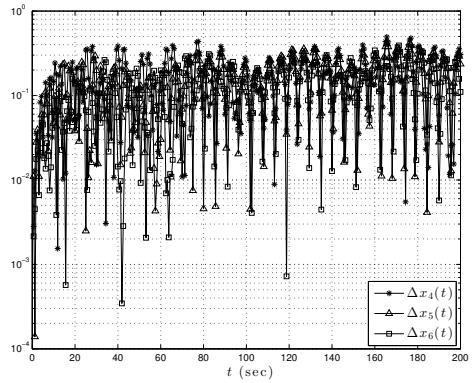


Figure 20: Normalized State Error Time History,  $Newmark-\beta^2$

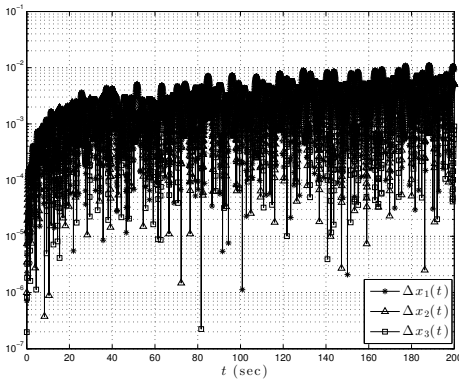


(a) Normalized Position Error

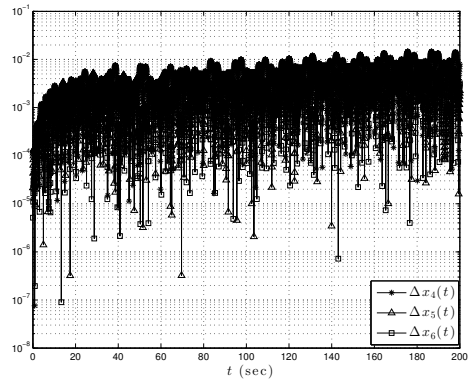


(b) Normalized Velocity Error

Figure 21: Normalized State Error Time History,  $HHT-\alpha^1$

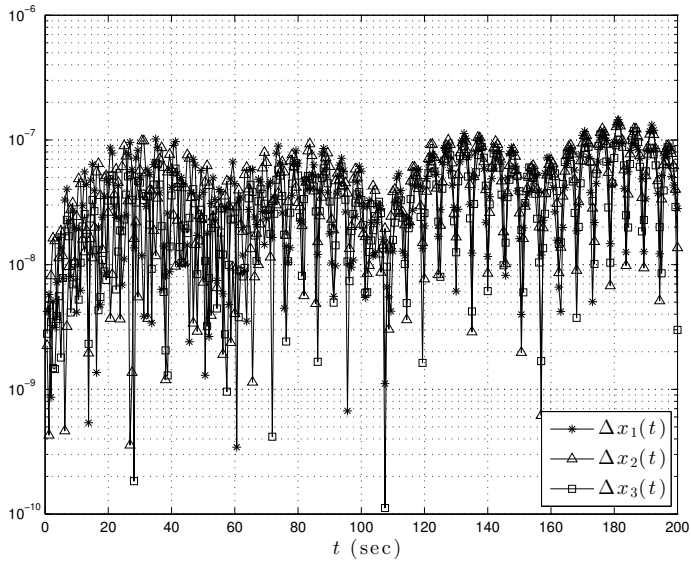


(a) Normalized Position Error

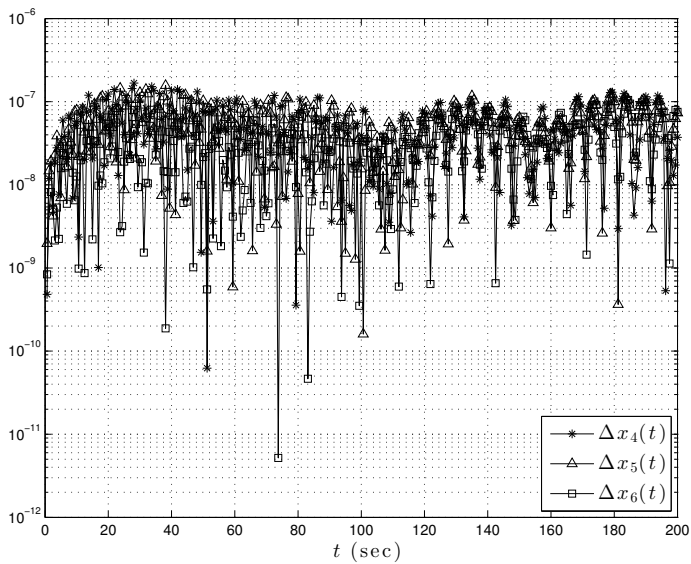


(b) Normalized Velocity Error

Figure 22: Normalized State Error Time History,  $HHT-\alpha^2$

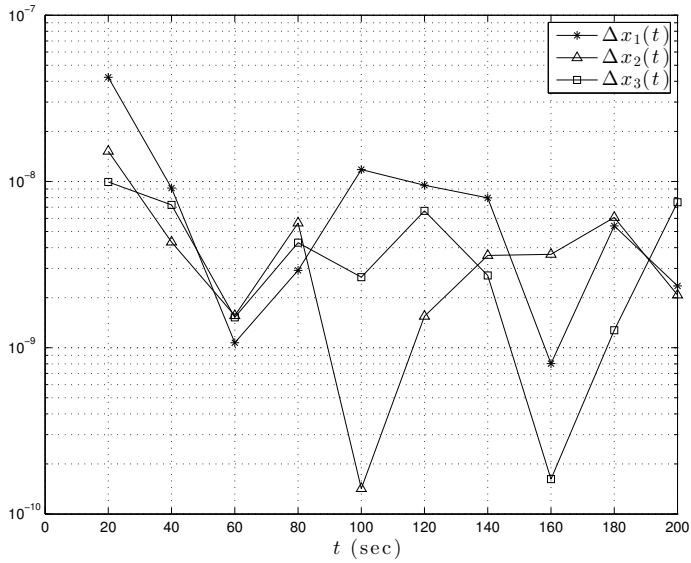


(a) Normalized Position Error

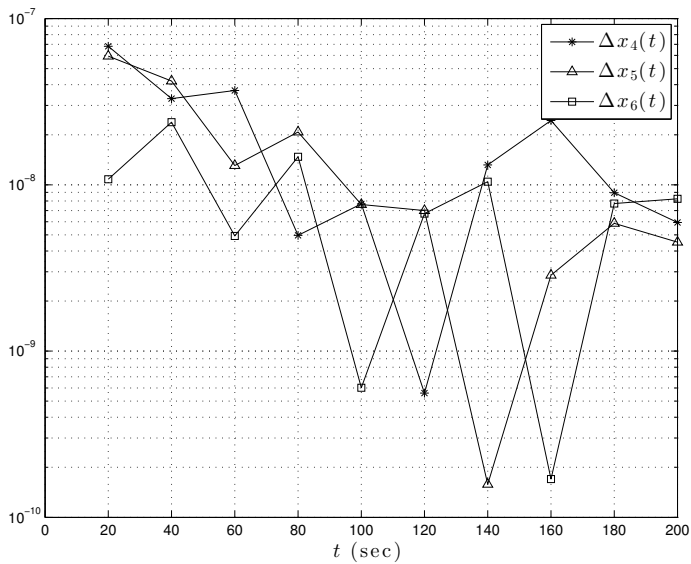


(b) Normalized Velocity Error

Figure 23: Normalized State Error Time History, *RBF-Coll*<sup>1</sup>

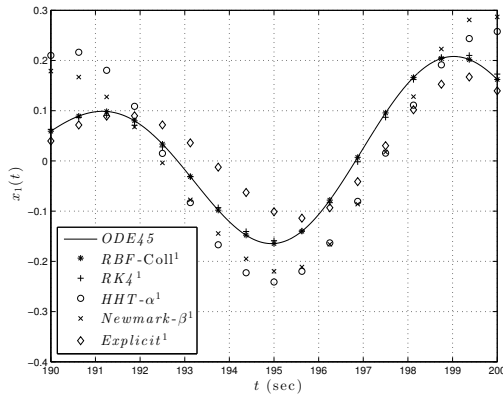


(a) Normalized Position Error

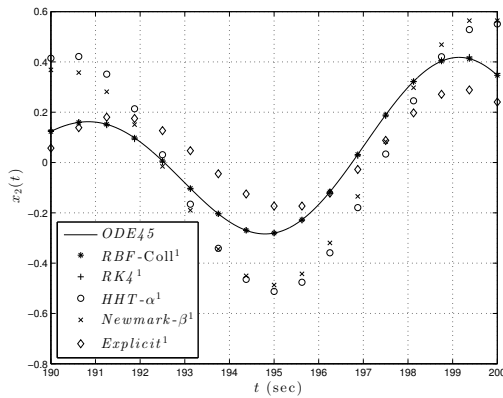


(b) Normalized Velocity Error

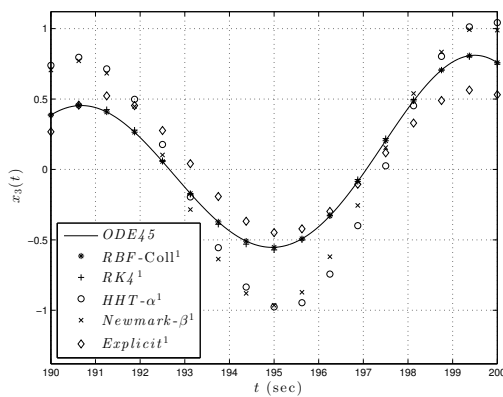
Figure 24: Normalized State Error Time History, *RBF-Coll*<sup>2</sup>



(a) Solution Comparison  $x_1(t)$



(b) Solution Comparison  $x_2(t)$



(c) Solution Comparison  $x_3(t)$

Figure 25: Solution Comparison,  $190 \leq t \leq 200$



As in the previous two sections the present *RBF-Coll* algorithm has the best combination of accuracy and computational cost which allows the user to take a larger time step and maintain a very high solution accuracy. In general, the computational cost of *RBF-Coll* is very comparable to the fastest, and as a consequence least accurate methods, which gives the algorithm a significant competitive advantage among various numerical integrators of general dynamical systems.

#### 4 Conclusion

The present *RBF-Coll* algorithm is shown to be highly accurate, fast and very simple to implement for various types of dynamical systems. Comparing the algorithm versus several implicit and explicit numerical integration methods clearly shows the advantages of such an algorithm that enables larger time step, high solution accuracy while maintaining a relatively low computational cost. For all the numerical examples shown in this study, *RBF-Coll*'s combination of accuracy and computational cost is superior to all other existing methods. The algorithm is shown to accurately and simply handle short and long periods of time integration, single and multi degrees of freedom system and finally transient and periodic solutions. Thus the *RBF-Coll* algorithm has a significant potential in handling various types of dynamical systems governed by second or higher order differential equations. Applications of the algorithm are, but not limited to, orbit propagation in celestial mechanics, dynamic buckling problems, optimal control and two-point boundary value problems. Areas of these studies will be explored in future works.

**Acknowledgement:** This work is supported by the US Army Research Labs VT Division, under an AFRL / UCI collaborative research agreement. The encouragement of Messrs. Dy Le and Jarret Riddick is thankfully acknowledged. This work is also supported by the Texas A&M Institute for Advanced Study (TIAS). It was initiated while S.N. Atluri visited TIAS briefly in January, 2014.

#### References

- Atluri, S. N.** (2005): *Methods of computer modeling in engineering & the sciences*, volume 1. Tech Science Press Palmdale.
- Belytschko, T.** (1976): A survey of numerical methods and computer programs for dynamic structural analysis. *Nuclear Engineering and Design*, vol. 37, no. 1, pp. 23–34.
- Belytschko, T.; Lin, J. I.; Chen-Shyh, T.** (1984): Explicit algorithms for the nonlinear dynamics of shells. *Computer methods in applied mechanics and engineering*, vol. 42, no. 2, pp. 225–251.

**Buhmann, M. D.** (2003): *Radial basis functions: theory and implementations*, volume 5. Cambridge university press Cambridge.

**Cvetićanin, L.** (2013): Ninety years of duffing's equation. *Theoretical and Applied Mechanics*, vol. 40, no. 1, pp. 49–63.

**Dai, H.-H.; Schnoor, M.; Atluri, S. N.** (2012): A simple collocation scheme for obtaining the periodic solutions of the duffing equation, and its equivalence to the high dimensional harmonic balance method: subharmonic oscillations. *Computer Modeling in Engineering and Sciences*, vol. 84, no. 5, pp. 459–497.

**Dokainish, M.; Subbaraj, K.** (1989): A survey of direct time-integration methods in computational structural dynamics-i. explicit methods. *Computers & Structures*, vol. 32, no. 6, pp. 1371–1386.

**Dong, L.; Alotaibi, A.; Mohiuddine, S.; Atluri, S.** (2014): Computational methods in engineering: a variety of primal & mixed methods, with global & local interpolations, for well-posed or ill-posed bcs. *CMES: Computer Modeling in Engineering & Sciences*, vol. 99, no. 1, pp. 1–85.

**Elgohary, T. A.; Dong, L.; Junkins, J. L.; Atluri, S. N.** (2014): Solution of post-buckling & limit load problems, without inverting the tangent stiffness matrix & without using arc-length methods. *CMES: Computer Modeling in Engineering & Sciences*, vol. 98, no. 6, pp. 543–563.

**Elgohary, T. A.; Dong, L.; Junkins, J. L.; Atluri, S. N.** (2014): Time domain inverse problems in nonlinear systems using collocation & radial basis functions. *CMES: Computer Modeling in Engineering & Sciences*, vol. 100, no. 1, pp. 59–84.

**Fehlberg, E.** (1969): Low-order classical runge-kutta formulas with stepsize control and their application to some heat transfer problems. Technical report, NASA, 1969.

**Filippi, S.; Gräf, J.** (1986): New runge–kutta–nyström formula-pairs of order 8 (7), 9 (8), 10 (9) and 11 (10) for differential equations of the form  $y'' = f(x, y)$ . *Journal of computational and applied mathematics*, vol. 14, no. 3, pp. 361–370.

**Hilber, H. M.; Hughes, T. J.; Taylor, R. L.** (1977): Improved numerical dissipation for time integration algorithms in structural dynamics. *Earthquake Engineering & Structural Dynamics*, vol. 5, no. 3, pp. 283–292.

**Liu, C.-S.; Atluri, S. N.** (2012): A globally optimal iterative algorithm using the best descent vector  $\dot{\mathbf{x}} = \lambda [\alpha_c \mathbf{F} + \mathbf{B}^T \mathbf{F}]$ , with the critical value  $\alpha_c$ , for solving a system of nonlinear algebraic equations  $\mathbf{F}(\mathbf{x}) = \mathbf{0}$ . *Computer Modeling in Engineering and Sciences*, vol. 84, no. 6, pp. 575.

- Liu, C. S.; Yeih, W.; Kuo, C. L.; Atluri, S. N.** (2009): A scalar homotopy method for solving an over/under determined system of non-linear algebraic equations. *Computer Modeling in Engineering and Sciences*, vol. 53, no. 1, pp. 47–71.
- Montenbruck, O.** (1992): Numerical integration methods for orbital motion. *Celestial Mechanics and Dynamical Astronomy*, vol. 53, no. 1, pp. 59–69.
- Newmark, N. M.** (1959): A method of computation for structural dynamics. *Journal of the Engineering Mechanics Division*, vol. 85, no. 3, pp. 67–94.
- Noor, A. K.; Lambiotte Jr, J. J.** (1979): Finite element dynamic analysis on cdc star-100 computer. *Computers & Structures*, vol. 10, no. 1, pp. 7–19.
- Sharp, P. W.** (2006): N-body simulations: The performance of some integrators. *ACM Transactions on Mathematical Software (TOMS)*, vol. 32, no. 3, pp. 375–395.
- Subbaraj, K.; Dokainish, M.** (1989): A survey of direct time-integration methods in computational structural dynamics-ii. implicit methods. *Computers & Structures*, vol. 32, no. 6, pp. 1387–1401.

