Journal of Sound and Vibration (2001) **247**(2), 343–365
doi:10.1006/jsvi.2001.3750, available online at http://www.idealibrary.com on IDEAL®

UNCONDITIONALLY STABLE COLLOCATION ALGORITHMS FOR SECOND ORDER INITIAL VALUE PROBLEMS

T. C. FUNG

School of Civil and Structural Engineering, Nanyang Technological University, Nanyang Avenue, Singapore 639798, *Singapore. E-mail*: *ctcfung@ntu.edu.sg*

(*Received* 10 *October* 2000, *and in final form* 23 *April* 2001)

In this paper, unconditionally stable higher order accurate time step integration algorithms suitable for second order initial value problems in collocation form are presented. The second order equations are manipulated directly. If the approximate solution is expressed as a polynomial of degree $n+1$, there are n unknowns to be determined after taking into account the two given initial conditions. It is well known that by suppressing the residuals of the governing equations at n distinct collocation points only, the resultant algorithms are only conditionally stable. In this paper, linear combinations of the residuals at $n+1$ distinct collocation points are used to solve for the n unknowns. The collocation points and the relative weights between the residuals are derived from the weighted residual method. The weighting functions are arbitrary polynomials of degree not exceeding $n-1$. To control the accuracy and stability properties of the resultant algorithms, the reduced integration technique is used to evaluate the integrals in the formulation. Once the reduced integration rules are decided, the equivalent collocation form can be derived. It is found that the resultant algorithms cast in the collocation form are easy to implement and can be used to tackle non-linear problems directly. Numerical examples are given to illustrate the validity of the present formulation.

© 2001 Academic Press

1. INTRODUCTION

Many engineering problems are governed by the hyperbolic partial differential equations, for example, dynamic vibration of multi-rigid and flexible body systems and structures, wave transmission in fluid, etc. The finite element method is commonly used to discretize the spatial domain. The resultant ordinary differential equations are in general second order in time. For example, the governing equations for linear elastic structural dynamic problems can be written as

$$[\mathbf{M}]\{\mathbf{u}_{,tt}(t)\}+[\mathbf{C}]\{\mathbf{u}_{,t}(t)\}+[\mathbf{K}]\{\mathbf{u}(t)\}=\{\mathbf{F}(t)\} \tag{1}$$

with initial conditions $\{\mathbf{u}(0)\}=\{\mathbf{u}_0\}$ and $\{\mathbf{u}_{,t}(0)\}=\{\mathbf{v}_0\}$, where $[\mathbf{M}]$, $[\mathbf{C}]$, $[\mathbf{K}]$ are the mass, damping and stiffness matrices, respectively, $\{\mathbf{F}(t)\}$ is the applied load vector, $\{\mathbf{u}(t)\}$ is the unknown displacement vector, and the notation $(\cdot)_{,t}$ is used to denote differentiation with respect to time t.

It is usually suggested that the second order equations can be solved indirectly as a set of first order equations after introducing another set of variables $\{\mathbf{v}(t)\}=\{\mathbf{u}_{,t}(t)\}$. Many well-established unconditionally stable higher order accurate algorithms exist for the solution of first order equations (see, for example, books by Houwen [1], Hairer *et al.* [2], Hairer and Wanner [3], Hughes [4], Wood [5], Zienkiewicz and Taylor [6] and the two survey papers by Dokainish and Subbaraj [7, 8]). However, this procedure may not be desirable, as the number of variables would be twice of the original problems. It is sometimes more desirable to solve the second order equations directly.

The collocation method is commonly used in solving differential equations. It is easy to understand and also straightforward to implement. The approximate solutions are obtained by suppressing the residuals of the governing equations at selected collocation points in the domain. The selection of the collocation points is crucial in obtaining a well-conditioned system of equations and ultimately in obtaining an accurate solution.

Unconditionally stable time step integration algorithms for *first order equations* can be constructed systematically from the collocation method [1–6]. Kujawski [9, 10] has used the collocation time finite element method to tackle various hear transfer problems. Recently, the collocation points to generate unconditionally stable higher order accurate algorithms for first order equations with controllable numerical dissipation have also been derived [11–13].

It is also possible to construct lower order unconditionally stable time step integration algorithms for *second order equations* from the collocation method. The Wilson-θ method [14] is a classical example of a collocation method for structural dynamic problems. A linear variation of acceleration is assumed over a time interval Δt. The residual of the equation is suppressed at $t = \theta\Delta t$, rather than at the end of the time interval with $t = \Delta t$. Unconditionally stable algorithms are obtained when $\theta \geqslant 1{\cdot}37$. The method can be combined with the Newmark method to derive an optimal collocation method [15]. Other modifications, such as the HHT-α method [16] and the generalized-α method [17] are also possible. Similarly, the Houboldt method [18] is obtained if the collocation occurs at the end of a time step for a three-step method. Note that all these algorithms are second order accurate only. Higher order accurate algorithms can also be derived systematically from the truncated Taylor series collocation algorithms [6]. However, these algorithms are only conditionally stable as they are special cases of linear multi-step algorithms.

Argyris *et al.* [19] used the Hermitian shape functions and point collocation to derive algorithms corresponding to Padé approximations. Gellert [20] used the cubic Hermitian shape functions and applied the method of collocation on the original differential equation, its first derivative and its first and second integrals at the end of the time interval. Four equations were obtained. Two linear independent combinations were used to solve for the final displacement and velocity from the given initial conditions. The algorithm was shown to be unconditionally stable and fourth order accurate. Wood [5] showed that the method was equivalent to the (2,2) diagonal Padé approximations to the exponential function. However, it is not clear how to generalize the procedure to construct other unconditionally stable higher order algorithms.

It can be seen that a general framework based on the collocation method to construct unconditionally stable higher order accurate time step integration algorithms for second order equations is still missing. It is well known that higher order accurate algorithms can be made competitive with conventional time step integration algorithms, particularly if high accuracy is required (e.g. reference [21]). However, unconditional stability of the algorithms is also important so that the time step would not be restricted by the spurious high-frequency responses in the system.

Kramarz [22] had investigated the stability of direct collocation methods for the numerical solutions of second order initial value problems. Houwen *et al.* [23] carried out further investigation and commented that the direct collocation methods using the Gauss, Radau or Lobatto quadrature points as collocation points were of limited value due to the rather small stability or periodicity boundaries of the resultant algorithms. A-stable algorithms were constructed by preconditioning the stiff components or by combining several algorithms to form the composite methods. However, the order of accuracy of the proposed A-stable algorithms was not very high. In this paper, A-stable algorithms are constructed by linearly combining the residuals at $n+1$ collocation points to establish the required n equations.

Golley [24] also used a cubic function satisfying the initial displacement and initial velocity to interpolate the displacement over a time interval. It was found that if the second order Gauss quadrature points were used in the collocation method, the resultant approximate solutions would be fourth order accurate. However, the algorithm was only conditionally stable. This algorithm is in fact equivalent to the algorithm presented in reference [25] with $W_{13} = -\frac{1}{6}$.

Unconditionally stable higher order accurate time step integration algorithms for the second order equations can be constructed systematically from the differential quadrature method if the displacement–velocity relation is weakly enforced [26]. In this paper, the initial conditions and the displacement–velocity relation are strongly enforced. The approximate solutions are assumed to be polynomials of degree $n+1$. The weighting functions are simply polynomials of degree not exceeding $n-1$. The reduced integration technique is used to turn the resultant algorithms from conditionally stable to unconditionally stable.

The reduced integration technique has been used extensively to tackle locking problems encountered in solid mechanics [6]. For time step integration algorithms, Borri *et al.* [27] had used this technique to turn conditionally stable algorithms into unconditionally stable algorithms. In their formulation, the Hamilton's weak principle was used with the momentum at the beginning and end of the time interval treated as independent variables in the single field formulation. As a result, there may be jump discontinuity for the momentum across two time steps. In this paper, the weighted residual framework is used and the conditionally stable algorithms are turned into unconditionally stable algorithms by using the reduced integration technique. In the present formulation, both the displacement and velocity (and hence momentum) are continuous across two time steps.

The manuscript is organized as follows. In section 2, a linear and homogenous second order equation is considered. The approximate solutions are assumed to be polynomials of degree $n+1$ satisfying the two given initial conditions. There are n unknowns to be determined. The weighted residual method is used and the resultant algorithms are related to the algorithms established previously. It is found that the conventional collocation approach could not generate unconditionally stable algorithms. In section 3, the n linearly independent weighting functions are assumed to be polynomials of degree not exceeding $n-1$. The reduced integration technique is used to construct algorithms with desirable characteristics. In section 4, the proposed algorithms are recast into the collocation form once the reduced integration rules are decided. It is found that the residuals at $n+1$ collocation points are required to construct the n equations. In section 5, non-linear problems are considered. It is found that the collocation form makes the solution procedure very simple. In section 6, numerical examples are given to illustrate the validity of the present algorithms. It is found that the present collocation algorithms are more accurate and stable than the conventional collocation algorithms.

2. LINEAR HOMOGENOUS SECOND ORDER EQUATIONS

Consider a linear and homogenous second order equation in the form

$$u_{,tt}(t) + u_{,t}(t) + \omega^2 u(t) = 0 \tag{2}$$

with initial conditions $u(0) = u_0$ and $u_{,t}(0) = v_0$. Without loss of generality, the approximate solutions are sought over a time interval $0 \leqslant t \leqslant \Delta t$. Equation (2) can be rewritten as

$$\ddot{u}(\tau) + 2\xi\omega\Delta t\dot{u}(\tau) + \omega^2\Delta t^2 u(\tau) = 0 \tag{3}$$

with initial conditions $u(0) = u_0$ and $\dot{u}(0) = v_0\Delta t$, where $\tau = t/\Delta t$ is the non-dimensional time parameter ranging from 0 to 1, and an overdot is used to denote differentiation with respect to τ. For dynamic problems, ξ and ω are the damping ratio and undamped natural frequency of the single-degree-of-freedom system respectively. Furthermore, u and v ($= u_{,t}$) will be used to denote the displacement and velocity respectively.

If $u(\tau)$ over a time interval $0 \leqslant \tau \leqslant 1$ (or $0 \leqslant t \leqslant \Delta t$) is assumed to be in the form of a polynomial of degree $n + 1$, it can be expresses as

$$u(\tau) = u_0 + v_0\Delta t\,\tau + U_1\,\tau^2 + \cdots + U_n\,\tau^{n+1}, \tag{4}$$

where $U_1, \ldots, U_n$ are the n unknown coefficients to be determined. The first and second time derivatives of u can be obtained by differentiating equation (4) with respect to t and can be written as

$$u_{,t}(\tau) = \dot{u}(\tau)/\Delta t = v_0 + (2U_1\,\tau + \cdots + (n+1)\,U_n\,\tau^n)/\Delta t, \tag{5a}$$

$$u_{,tt}(t) = \ddot{u}(\tau)/\Delta t^2 = (2\times 1\,U_1 + \cdots + (n+1)\times n\,U_n\tau^{n-1})/\Delta t^2. \tag{5b}$$

Substituting equations (4) and (5) into equation (3), the residual of the differential equation $R(\tau)$ is given by

$$R(\tau) = \frac{1}{\Delta t^2}\,[\mathbf{T}]\,([\boldsymbol{\Lambda}_2] + 2\xi\omega\Delta t\,[\boldsymbol{\Lambda}_1] + \omega^2\Delta t^2[\boldsymbol{\Lambda}_0])\,\{\mathbf{U}\} + (2\xi\omega + \omega^2\,\tau\Delta t)\,v_0 + \omega^2\,u_0, \tag{6}$$

where $[\mathbf{T}] = [1, \tau, \ldots, \tau^{n+1}]$, $\{\mathbf{U}\} = [U_1, \ldots, U_n]^{\mathrm{T}}$ and $[\boldsymbol{\Lambda}_2]$, $[\boldsymbol{\Lambda}_1]$, $[\boldsymbol{\Lambda}_0]$ are $n+2$ by n matrices defined as

$$[\boldsymbol{\Lambda}_2] = \left[\begin{array}{ccc} 1\times 2 & & \\ & \ddots & \\ & & n\times(n+1) \\ \hdashline 0 & \cdots & 0 \\ 0 & \cdots & 0 \end{array}\right], \quad [\boldsymbol{\Lambda}_1] = \left[\begin{array}{ccc} 0 & \cdots & 0 \\ \hdashline 2 & & \\ & \ddots & \\ & & n+1 \\ \hdashline 0 & \cdots & 0 \end{array}\right], \quad [\boldsymbol{\Lambda}_0] = \left[\begin{array}{ccc} 0 & \cdots & 0 \\ 0 & \cdots & 0 \\ \hdashline 1 & & \\ & \ddots & \\ & & 1 \end{array}\right]. \tag{7}$$

If the weighted residual method is used to solve for the n unknowns $U_1, \ldots, U_n$, n linearly independent weighting functions $\psi_1(\tau), \ldots, \psi_n(\tau)$ can be specified such that

$$\int_0^1 \psi_i(\tau)R(\tau)\,\mathrm{d}\tau = 0 \quad \text{for } 1 \leqslant i \leqslant n. \tag{8}$$

Let W_{ij} be the weighting parameter defined as

$$W_{ij} = \int_0^1 \psi_i(\tau)\tau^j\,\mathrm{d}\tau \quad \text{for } 1 \leqslant i \leqslant n \quad \text{and} \quad 0 \leqslant j \leqslant n+1 \tag{9}$$

and [**W**] be the weighting parameter matrix defined as

$$[\mathbf{W}] = \begin{bmatrix} W_{10} & W_{11} & \cdots & W_{1,n+1} \\ W_{20} & W_{21} & \cdots & W_{2,n+1} \\ \vdots & \vdots & \ddots & \vdots \\ W_{n0} & W_{n1} & \cdots & W_{n,n+1} \end{bmatrix}. \tag{10}$$

Equation (8) can be collectively written in a matrix form as

$$[\mathbf{W}]\,([\boldsymbol{\Lambda}_2] + 2\xi\omega\Delta t\,[\boldsymbol{\Lambda}_1] + \omega^2\Delta t^2\,[\boldsymbol{\Lambda}_0])\{\mathbf{U}\} + [\mathbf{W}\}\,\{\mathbf{U}_0\} = 0, \tag{11}$$

where $[\mathbf{U}_0\} = [2\xi\omega\Delta t^2 v_0 + \omega^2\Delta t^2 u_0, \omega^2\Delta t^3 v_0, 0, \dots, 0]^{\mathrm{T}}$.

Since the problem should still be solvable even when $\omega = 0$, the matrix

$$[\mathbf{W}]\,[\boldsymbol{\Lambda}_2] = \begin{bmatrix} W_{10} & W_{11} & \cdots & W_{1,n-1} \\ W_{20} & W_{21} & \cdots & W_{2,n-1} \\ \vdots & \vdots & \ddots & \vdots \\ W_{n0} & W_{n1} & \cdots & W_{n,n-1} \end{bmatrix} \begin{bmatrix} 1\times 2 & & & \\ & 2\times 3 & & \\ & & \ddots & \\ & & & n\times(n+1) \end{bmatrix} \tag{12}$$

must be non-singular. Hence, the n weighting functions $\psi_1(\tau), \dots, \psi_n(t)$ can be recombined to form another set of n linearly independent weighting functions $\bar{\psi}_1(\tau), \dots, \bar{\psi}_n(\tau)$ such that [**W**] can be normalized to $[\bar{\mathbf{W}}]$ as

$$[\bar{\mathbf{W}}] = \begin{bmatrix} 1 & & & a_1 & b_1 \\ & \ddots & & \vdots & \vdots \\ & & 1 & a_n & b_n \end{bmatrix}, \tag{13}$$

where

$$\begin{Bmatrix} a_1 \\ \vdots \\ a_n \end{Bmatrix} = \begin{bmatrix} W_{10} & \cdots & W_{1,n-1} \\ \vdots & & \vdots \\ W_{n0} & \cdots & W_{n,n-1} \end{bmatrix}^{-1} \begin{Bmatrix} W_{1,n} \\ \vdots \\ W_{n,n} \end{Bmatrix}, \begin{Bmatrix} b_1 \\ \vdots \\ b_n \end{Bmatrix} = \begin{bmatrix} W_{10} & \cdots & W_{1,n-1} \\ \vdots & & \vdots \\ W_{n0} & \cdots & W_{n,n-1} \end{bmatrix}^{-1} \begin{Bmatrix} W_{1,n+1} \\ \vdots \\ W_{n,n+1} \end{Bmatrix}$$

(14a, b)

and

$$\begin{Bmatrix} \bar{\psi}_1 \\ \vdots \\ \bar{\psi}_n \end{Bmatrix} = \begin{bmatrix} W_{10} & \cdots & W_{1,n-1} \\ \vdots & & \vdots \\ W_{n0} & \cdots & W_{n,n-1} \end{bmatrix}^{-1} \begin{Bmatrix} \psi_1 \\ \vdots \\ \psi_n \end{Bmatrix}. \tag{14c}$$

Obviously, the accuracy of the approximate solutions and the algorithmic characteristics of the resultant algorithms depend on the algorithmic parameters $a_1, \dots, a_n, b_1, \dots, b_n$. It has been shown in detail in reference [28] that the resultant algorithms would be unconditionally stable if

$$a_k = \frac{(-1)^{n-k} n!n!(n+k-2)!}{(k-1)!(k-1)!(n+1-k)!2n!} \frac{2(n+(k-1)\mu)}{(1+\mu)} \quad \text{for } 1 \leqslant k \leqslant n, \tag{15a}$$

$$b_1 = \frac{n+1}{n} W_1 W_n \quad \text{and} \quad b_k = \frac{n+1}{k-1} W_{k-1} + \frac{n+1}{n} W_k W_n \quad \text{for } 2 \leqslant k \leqslant n \tag{15b}$$

and $-1 < \mu \leqslant 1$. In fact, μ corresponds to the ultimate spectral radius of the resultant algorithm as $\omega \Delta t$ approaches infinity. Besides, the order of accuracy of the solutions at the end of the time interval (i.e., $\tau = 1$ or $t = \Delta t$) is $2n - 1$ if $-1 < \mu < 1$ and $2n$ if $\mu = 1$.

If the weighting functions are Dirac delta functions at n distinct collocation points $\tau_1, \ldots, \tau_n$ (i.e., $\psi_i(\tau) = \delta(\tau - \tau_i)$ for $1 \leqslant i \leqslant n$), the formulation is equivalent to the collocation method. Furthermore, $W_{ij} = \tau_i^j$. It is therefore required that

$$\begin{Bmatrix} a_1 \\ a_2 \\ \vdots \\ a_n \end{Bmatrix} = \begin{bmatrix} 1 & \tau_1 & \cdots & \tau_1^{n-1} \\ 1 & \tau_2 & \cdots & \tau_1^{n-1} \\ \vdots & \vdots & & \vdots \\ 1 & \tau_n & \cdots & \tau_n^{n-1} \end{bmatrix}^{-1} \begin{Bmatrix} \tau_1^n \\ \tau_2^n \\ \vdots \\ \tau_n^n \end{Bmatrix}, \quad \begin{Bmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{Bmatrix} = \begin{bmatrix} 1 & \tau_1 & \cdots & \tau_1^{n-1} \\ 1 & \tau_2 & \cdots & \tau_1^{n-1} \\ \vdots & \vdots & & \vdots \\ 1 & \tau_n & \cdots & \tau_n^{n-1} \end{bmatrix}^{-1} \begin{Bmatrix} \tau_1^{n+1} \\ \tau_2^{n+1} \\ \vdots \\ \tau_n^{n+1} \end{Bmatrix}. \tag{16a, b}$$

From equation (16a), it can be seen that $\tau_1, \ldots, \tau_n$ are completely determined by the zeros of the polynomial

$$\tau^n = a_1 + a_2 \tau + \cdots + a_n \tau^{n-1}. \tag{17}$$

However, it can be verified that in general, these zeros will not satisfy equation (16b) or the polynomial

$$\tau^{n+1} = b_1 + b_2 \tau + \cdots + b_n \tau^{n-1}. \tag{18}$$

Hence, it is impossible to construct unconditionally stable higher order accurate time step integration algorithms with the parameters shown in equation (15) by using n collocation points only.

It is interesting to note that for the first order equations, only equation (16a) is required and it is possible to construct unconditionally stable higher order accurate time step integration algorithms by using just n collocation points [12, 13]. In this paper, it is shown that in general, $n + 1$ collocation points are required to solve second order equations.

Various types of weighting functions to generate the unconditionally stable higher order accurate time step integration algorithms with the parameters shown in equation (15) have been reported in reference [28]. It should be noted that if polynomials are used as the weighting functions, the weighting functions should include the stabilizing weighting functions. Hence, the weighting functions are polynomials of degree $n + 1$ in general. It can be verified that restricting the weighting functions to polynomials of degree not exceeding $n - 1$ only would not be able to generate unconditionally stable algorithms. In the next section, the weighting functions are simply polynomials of degree not exceeding $n - 1$. The resultant algorithms are made unconditionally stable by using reduced integration.

3. REDUCED INTEGRATION

If the weighting functions are polynomials of degree not exceeding $n - 1$, without loss of generality, $\psi_1(\tau), \ldots, \psi_n(\tau)$ can be assumed to be

$$\psi_i(\tau) = \tau^{i-1} \quad \text{for } 1 \leqslant i \leqslant n. \tag{19}$$

If the integrals in equation (9) are evaluated exactly, W_{ij} should be given by

$$W_{ij} = \int_0^1 \tau^{i-1} \tau^j \, d\tau = \frac{1}{i+j} \quad \text{for } 1 \leqslant i \leqslant n \quad \text{and} \quad 0 \leqslant j \leqslant n. \tag{20}$$

It can be verified that the resultant algorithms so constructed are only conditionally stable.

However, if the reduced integration technique is used to evaluate the integrals, let

$$W_{ij} = \sum_{k=0}^{m} \sigma_k \tau_k^{i+j-1} = J_{i+j} \quad \text{for } 1 \leqslant i \leqslant n \quad \text{and} \quad 0 \leqslant j \leqslant n, \tag{21}$$

where $m + 1$ is the number of integration points, τ_k and σ_k are abscissa and weight of the kth integration point respectively. It can be seen that the weighting parameter matrix $[\mathbf{W}]$ can be written as

$$[\mathbf{W}] = \begin{bmatrix} 1 & 1 & \cdots & 1 \\ \tau_0 & \tau_1 & \cdots & \tau_m \\ \vdots & \vdots & & \vdots \\ \tau_0^{n-1} & \tau_1^{n-1} & \cdots & \tau_m^{n-1} \end{bmatrix} \begin{bmatrix} \sigma_0 & & & \\ & \sigma_1 & & \\ & & \ddots & \\ & & & \sigma_m \end{bmatrix} \begin{bmatrix} 1 & \tau_0 & \cdots & \tau_0^{n+1} \\ 1 & \tau_1 & \cdots & \tau_1^{n+1} \\ \vdots & \vdots & & \vdots \\ 1 & \tau_m & \cdots & \tau_m^{n+1} \end{bmatrix}$$

$$= \begin{bmatrix} J_1 & J_2 & \cdots & J_{n+2} \\ J_2 & J_3 & \cdots & J_{n+3} \\ \vdots & \vdots & & \vdots \\ J_n & J_{n+1} & \cdots & J_{2n+1} \end{bmatrix}. \tag{22}$$

To reproduce the unconditionally stable higher order accurate algorithms with the parameters given by equation (15), it is therefore required that

$$\begin{Bmatrix} a_1 \\ a_2 \\ \vdots \\ a_n \end{Bmatrix} = \begin{bmatrix} J_1 & J_2 & \cdots & J_n \\ J_2 & J_3 & \cdots & J_{n+1} \\ \vdots & \vdots & & \vdots \\ J_n & J_{n+1} & \cdots & J_{2n-1} \end{bmatrix}^{-1} \begin{Bmatrix} J_{n+1} \\ J_{n+2} \\ \vdots \\ J_{2n} \end{Bmatrix}, \quad \begin{Bmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{Bmatrix}$$

$$= \begin{bmatrix} J_1 & J_2 & \cdots & J_n \\ J_2 & J_3 & \cdots & J_{n+1} \\ \vdots & \vdots & & \vdots \\ J_n & J_{n+1} & \cdots & J_{2n-1} \end{bmatrix}^{-1} \begin{Bmatrix} J_{n+2} \\ J_{n+3} \\ \vdots \\ J_{2n+1} \end{Bmatrix}. \tag{23a, b}$$

Since the values of $J_1, J_2, \ldots, J_{2n+1}$ can be scaled up proportionally by an arbitrary constant, without loss of generality, J_1 can be assumed to be unity (i.e., $J_1 = 1$). The remaining values of $J_2, \ldots, J_{2n+1}$ can be obtained from the following $2n$ equations in terms of a_k and b_k and hence n and μ in general.

$$\left[\begin{array}{cccc|cccc} a_2 & \cdots & a_n & -1 & 0 & & & \\ a_1 & a_2 & \cdots & a_n & -1 & 0 & & \\ & \ddots & \ddots & & \ddots & \ddots & \ddots & \\ & & a_1 & a_2 & \cdots & a_n & -1 & 0 \\ \hline b_2 & \cdots & b_n & 0 & -1 & & & \\ b_1 & b_2 & \cdots & b_n & 0 & -1 & & \\ & \ddots & \ddots & & \ddots & \ddots & \ddots & \\ & & b_1 & b_2 & \cdots & b_n & 0 & -1 \end{array}\right] \left\{\begin{array}{c} J_2 \\ J_3 \\ \vdots \\ J_{n+1} \\ \hline J_{n+2} \\ J_{n+3} \\ \vdots \\ J_{2n+1} \end{array}\right\} = \left\{\begin{array}{c} -a_1 \\ 0 \\ \vdots \\ 0 \\ \hline -b_1 \\ 0 \\ \vdots \\ 0 \end{array}\right\}. \tag{24}$$

Some of the J_k values are shown in Table 1. Note that in general, $J_k \neq 1/k$, except when $\mu = 1$, $J_k = 1/k$ for $2 \leqslant k \leqslant 2n$. Yet, J_{2n+1} is still different from $1/(2n+1)$.

Once the values of $J_1, \ldots, J_{2n+1}$ are determined, τ_k and σ_k can be determined as follows. Since from equation (21)

$$\left\{\begin{array}{c} J_1 \\ J_2 \\ \vdots \\ J_{2n} \\ J_{2n+1} \end{array}\right\} = \begin{bmatrix} 1 & 1 & \cdots & 1 \\ \tau_0 & \tau_1 & \cdots & \tau_m \\ \vdots & \vdots & & \vdots \\ \tau_0^{2n-1} & \tau_1^{2n-1} & \cdots & \tau_m^{2n-1} \\ \tau_0^{2n} & \tau_1^{2n} & \cdots & \tau_m^{2n} \end{bmatrix} \left\{\begin{array}{c} \sigma_0 \\ \sigma_1 \\ \vdots \\ \sigma_m \end{array}\right\} \tag{25}$$

there are $2m+2$ unknowns ($\sigma_0, \sigma_1, \ldots, \sigma_m$, $\tau_0, \tau_1, \ldots, \tau_m$) to be solved from the $2n+1$ equations. For a solution to exist in general, m should be at least equal to n. In the following, m is chosen as n. As a result, the number of unknowns ($2n+2$) is more than the number of equations ($2n+1$) by one. An additional equation can be specified. In the following, a simple condition $\tau_0 = 0$ is used although other conditions, such as $\tau_m = 1$, are also possible.

If $\tau_0 = 0$ and $m = n$, then from equation (25), $\sigma_0 = 1 - \sigma_1 - \cdots - \sigma_n$, and $\tau_1, \ldots, \tau_n$, and $\sigma_1, \ldots, \sigma_n$ can be determined from

$$\left\{\begin{array}{c} J_2 \\ J_3 \\ \vdots \\ J_{n+1} \end{array}\right\} = \begin{bmatrix} 1 & 1 & \cdots & 1 \\ \tau_1 & \tau_2 & \cdots & \tau_n \\ \vdots & \vdots & & \vdots \\ \tau_1^{n-1} & \tau_2^{n-1} & \cdots & \tau_n^{n-1} \end{bmatrix} \left\{\begin{array}{c} \hat{\sigma}_1 \\ \hat{\sigma}_2 \\ \vdots \\ \hat{\sigma}_n \end{array}\right\} \quad \text{and} \quad \left\{\begin{array}{c} J_{n+2} \\ J_{n+3} \\ \vdots \\ J_{2n+1} \end{array}\right\}$$

$$= \begin{bmatrix} \tau_1^n & \tau_2^n & \cdots & \tau_n^n \\ \tau_1^{n+1} & \tau_2^{n+1} & \cdots & \tau_n^{n+1} \\ \vdots & \vdots & & \vdots \\ \tau_1^{2n-1} & \tau_2^{2n-1} & \cdots & \tau_n^{2n-1} \end{bmatrix} \left\{\begin{array}{c} \hat{\sigma}_1 \\ \hat{\sigma}_2 \\ \vdots \\ \hat{\sigma}_n \end{array}\right\}, \tag{26}$$

TABLE 1

Values of $J_1, J_2, \dots, J_{2n+1}$ for various n

n	$J_1, J_2, \dots, J_{2n+1}$
2	$1, \frac{1}{2}+\frac{1}{2}\frac{1-\mu^2}{\mu^2+\mu+1}, \frac{1}{3}+\frac{1}{3}\frac{(2+\mu)(1-\mu)}{\mu^2+\mu+1}, \frac{1}{4}+\frac{1}{4}\frac{(1-\mu)(\mu^2+3\mu+3)}{(1+\mu)(\mu^2+\mu+1)},$ $\frac{1}{2}\frac{\mu^2+2\mu+2}{(\mu^2+\mu+1)(1+\mu)^2}$
3	$1, \frac{1}{2}+\frac{1}{2}\frac{(1-\mu^2)(3\mu^2+4\mu+3)}{3\mu^4+6\mu^3+22\mu^2+6\mu+3}, \frac{1}{3}+\frac{(1-\mu^2)(\mu^2+2\mu+2)}{3\mu^4+6\mu^3+22\mu^2+6\mu+3},$ $\frac{1}{4}+\frac{3}{4}\frac{(1-\mu^2)(\mu^2+2\mu+3)}{3\mu^4+6\mu^3+22\mu^2+6\mu+3}, \frac{1}{5}+\frac{1}{5}\frac{(1-\mu)(3\mu^3+9\mu^2+16\mu+12)}{3\mu^4+6\mu^3+22\mu^2+6\mu+3},$ $\frac{1}{6}+\frac{1}{30}\frac{(1-\mu)(15\mu^4+60\mu^3+128\mu^2+150\mu+75)}{(1+\mu)\ (3\mu^4+6\mu^3+22\mu^2+6\mu+3)},$ $\frac{1}{75}\frac{153\mu^4+394\mu^3+498\mu^2+450\mu+225}{(3\mu^4+6\mu^3+22\mu^2+6\mu+3)\ (1+\mu)^2}$
4	$1, \frac{1}{2}+\frac{(1-\mu^2)(8\mu^4+20\mu^3+49\mu^2+20\mu+8)}{16\mu^6+48\mu^5+314\mu^4+189\mu^3+314\mu^2+48\mu+16},$ $\frac{1}{3}+\frac{1}{3}\frac{(1-\mu^2)(16\mu^4+48\mu^3+147\mu^2+72^3\mu+32)}{16\mu^6+48\mu^5+314\mu^4+189\mu^3+314\mu^2+48\mu+16},$ $\frac{1}{4}+\frac{1}{2}\frac{(1-\mu^2)(8\mu^4+24\mu^3+85\mu^2+48\mu+24)}{16\mu^6+48\mu^5+314\mu^4+189\mu^3+314\mu^2+48\mu+16},$ $\frac{1}{5}+\frac{4}{5}\frac{(1-\mu^2)(4\mu^4+12\mu^3+45\mu^2+28\mu+16)}{16\mu^6+48\mu^5+314\mu^4+189\mu^3+314\mu^2+48\mu+16},$ $\frac{1}{6}+\frac{1}{3}\frac{(1-\mu^2)(8\mu^4+24\mu^3+93\mu^2+60\mu+40)}{16\mu^6+48\mu^5+314\mu^4+189\mu^3+314\mu^2+48\mu+16},$ $\frac{1}{7}+\frac{1}{7}\frac{(1-\mu)(16\mu^5+64\mu^4+238\mu^3+315\mu^2+216\mu+96)}{16\mu^6+48\mu^5+314\mu^4+189\mu^3+314\mu^2+48\mu+16},$ $\frac{1}{8}+\frac{1}{56}\frac{(1-\mu)(112\mu^6+560\mu^5+2134\mu^4+3935\mu^3+3842\mu^2+2352\mu+784)}{(1+\mu)\ (16\mu^6+48\mu^5+314\mu^4+189\mu^3+314\mu^2+48\mu+16)},$ $\frac{1}{196}\frac{2944\mu^6+8192\mu^5+16\,361\mu^4+23\,508\mu^3+18\,792\mu^2+9408\mu+3136}{(16\mu^6+48\mu^5+314\mu^4+189\mu^3+314\mu^2+48\mu+16)\ (1+\mu)^2}$

where $\hat{\sigma}_k = \sigma_k/\tau_k$. As a result, $\tau_1, \dots, \tau_n$ satisfy the following equation:

$$\begin{Bmatrix} J_{n+2} \\ J_{n+3} \\ \vdots \\ J_{2n+1} \end{Bmatrix} = \begin{bmatrix} \tau_1^n & \tau_2^n & \cdots & \tau_n^n \\ \tau_1^{n+1} & \tau_2^{n+1} & \cdots & \tau_n^{n+1} \\ \vdots & \vdots & & \vdots \\ \tau_1^{2n-1} & \tau_2^{2n-1} & \cdots & \tau_n^{2n-1} \end{bmatrix} \begin{bmatrix} 1 & 1 & \cdots & 1 \\ \tau_1 & \tau_2 & \cdots & \tau_n \\ \vdots & \vdots & & \vdots \\ \tau_1^{n-1} & \tau_2^{n-1} & \cdots & \tau_n^{n-1} \end{bmatrix}^{-1} \begin{Bmatrix} J_2 \\ J_3 \\ \vdots \\ J_{n+1} \end{Bmatrix}. \quad (27)$$

Using a technique presented in reference [29], it can be shown that $\tau_1, \dots, \tau_n$ are in fact given by the zeros of the following polynomial:

$$\tau^n = s_1 + s_2\tau + \cdots + s_n\tau^{n-1}, \tag{28}$$

where the coefficients $s_1, \dots, s_n$ are given by

$$\begin{Bmatrix} s_1 \\ s_2 \\ \vdots \\ s_n \end{Bmatrix} = \begin{bmatrix} J_2 & J_3 & \cdots & J_{n+1} \\ J_3 & J_4 & \cdots & J_{n+2} \\ \vdots & \vdots & & \vdots \\ J_{n+1} & J_{n+2} & \cdots & J_{2n} \end{bmatrix}^{-1} \begin{Bmatrix} J_{n+2} \\ J_{n+3} \\ \vdots \\ J_{2n+1} \end{Bmatrix}. \tag{29}$$

To include $\tau_0 = 0$, the polynomial to generate the required $\tau_0, \dots, \tau_n$ are

$$P(\tau) = \tau^{n+1} - (s_1\tau + s_2\tau^2 + \cdots + s_n\tau^n). \tag{30}$$

It can be verified that $P(\tau)$ is also equivalent to

$$P(\tau) = \frac{(-1)^n(n+1)!}{(1+\mu)(2n)!}\left(2n(1-\mu)\frac{\mathrm{d}^{n-2}}{\mathrm{d}\tau^{n-2}}(\tau^{n-1}(1-\tau)^n) + 2\mu\frac{\mathrm{d}^{n-1}}{\mathrm{d}\tau^{n-1}}(\tau^n(1-\tau)^n)\right). \tag{31}$$

Hence, if $\mu = 1$, $\tau_0, \dots, \tau_n$ are the zeros of

$$\frac{\mathrm{d}^{n-1}}{\mathrm{d}\tau^{n-1}}(\tau^n(1-\tau)^n) \tag{32}$$

and are therefore the Lobatto quadrature points. It is well known that by using the $n+1$ Lobatto quadrature points, polynomials of degree not exceeding $2n-1$ can be integrated exactly. In this case, $J_k = 1/k$ for $1 \leqslant k \leqslant 2n$ and in general, J_{2n+1} is different from $1/(2n+1)$. It should be noted that the present reduced integration rules are different from those presented in reference [27] where the Gauss quadrature points were proposed.

Once $\tau_0, \dots, \tau_n$ are decided, $\sigma_0, \dots, \sigma_n$ can be determined from

$$\begin{Bmatrix} \sigma_0 \\ \sigma_1 \\ \vdots \\ \sigma_n \end{Bmatrix} = \begin{bmatrix} 1 & 1 & \cdots & 1 \\ \tau_0 & \tau_1 & \cdots & \tau_n \\ \vdots & \vdots & & \vdots \\ \tau_0^n & \tau_1^n & \cdots & \tau_n^n \end{bmatrix}^{-1} \begin{Bmatrix} J_1 \\ J_2 \\ \vdots \\ J_{n+1} \end{Bmatrix}. \tag{33}$$

For example, when $n = 2$, it can be verified that

$$\tau_0 = 0, \quad \tau_1 = \frac{1}{1+\mu}, \quad \tau_2 = 1, \quad \sigma_0 = \frac{\mu^2}{2(1+\mu+\mu^2)}, \quad \sigma_1 = \frac{1+2\mu+\mu^2}{2(1+\mu+\mu^2)}, \quad \sigma_2 = \frac{1}{2(1+\mu+\mu^2)}. \tag{34}$$

Similarly, when $n = 3$, it can be verified that

$$\tau_0 = 0, \quad \tau_1 = \frac{7+3\mu-\sqrt{9\mu^2+2\mu+9}}{10+10\mu}, \quad \tau_2 = \frac{7+3\mu+\sqrt{9\mu^2+2\mu+9}}{10+10\mu}, \quad \tau_3 = 1, \tag{35}$$

$$\sigma_0 = \frac{\mu^2(9\mu^2+2\mu+9)}{18\mu^4+36\mu^3+132\mu^2+36\mu+18}, \quad \sigma_3 = \frac{9\mu^2+2\mu+9}{18\mu^4+36\mu^3+132\mu^2+36\mu+18},$$

$$\sigma_1 = \frac{9\mu^4+34\mu^3+114\mu^2+34\mu+9+(3\mu^3+11\mu^2-11\mu-3)\sqrt{9\mu^2+2\mu+9}}{36\mu^4+72\mu^3+264\mu^2+72\mu+36},$$

$$\sigma_2 = \frac{9\mu^4+34\mu^3+114\mu^2+34\mu+9-(3\mu^3+11\mu^2-11\mu-3)\sqrt{9\mu^2+2\mu+9}}{36\mu^4+72\mu^3+264\mu^2+72\mu+36}.$$

For a large value of n, it is difficult to express $\tau_0, \ldots, \tau_n$ and $\sigma_0, \ldots, \sigma_n$ explicitly in terms of n and μ. Table 2 shows some of the values of $\tau_0, \ldots, \tau_n$ and $\sigma_0, \ldots, \sigma_n$ for various n when μ varies from 0·1 to 1·0. Note that $\tau = 1$ is always one of the roots for equation (31). Hence, in general, τ_n is assigned as unity (i.e., $\tau_n = 1$). Besides, it can be verified that μ cannot be zero as in this case, all the J_k values in equation (24) are unity (i.e., $J_k = 1$ for $1 \leqslant k \leqslant 2n+1$). As a result, the matrix in equation (29) cannot be inverted and there is no solution for σ_k and τ_k.

4. EQUIVALENT COLLOCATION FORM

Once the values for σ_k and τ_k are determined, the reduced integration rules are decided. Hence, the formulation is basically established and the approximate solutions can be evaluated. However, to make the formulation more versatile, it is of advantage to cast the algorithms in an equivalent collocation form. It can be seen that the collocation form is most obvious for non-linear problems, as the non-linear algebraic equations are simply the residuals at the collocation points [6].

Since the weighting functions $\psi_1(\tau), \ldots, \psi_n(\tau)$ are polynomials of degree not exceeding $n-1$, without loss of generality, the weighting functions can be recombined to form the $(n-1)$th order Lagrange functions, i.e.,

$$\psi_i(\tau) = L_i^{n-1}(\tau) = \prod_{\substack{k=1 \\ k \neq i}}^{n} \frac{\tau - \tau_k}{\tau_i - \tau_k} \quad \text{for } 1 \leqslant i \leqslant n \tag{36}$$

such that $L_i^{n-1}(\tau_j) = 1$ if $i = j$ and $= 0$ otherwise for $1 \leqslant i \leqslant n$. Note that $L_i^{n-1}(\tau_0)$ is in general not zero or unity but can be evaluated as

$$L_i^{n-1}(\tau_0) = -\frac{s_1}{\tau_1 Q(\tau_i)} = \frac{(-1)^{n+1}(n-1)!(n+1)!}{(1+\mu)(2n-1)!\,\tau_i Q(\tau_i)} \quad \text{for } 1 \leqslant i \leqslant n \tag{37}$$

since $\tau_0 = 0$ and

$$\prod_{\substack{k=1 \\ k \neq i}}^{n} \tau_0 - \tau_k = \frac{(-1)^{n-1}}{\tau_i} \prod_{k=1}^{n} \tau_k = \frac{-s_1}{\tau_i} = \frac{(-1)^{n+1}(n-1)!(n+1)!}{(1+\mu)(2n-1)!\,\tau_i} \quad \text{for } 1 \leqslant i \leqslant n \tag{38}$$

and

$$\prod_{\substack{k=1 \\ k \neq i}}^{n} \tau_i - \tau_k = \frac{\mathrm{d}}{\mathrm{d}\tau}\left(\frac{P(\tau)}{\tau}\right)\bigg|_{\tau=\tau_i} = n\tau_i^{n-1} - (s_2 + \cdots + (n-1)s_n\tau_i^{n-2})$$

$$= Q(\tau_i) \quad \text{for } 1 \leqslant i \leqslant n. \tag{39}$$

Equation (8) then becomes

$$\int_0^1 \psi_i(\tau)\,R(\tau)\,\mathrm{d}\tau = \int_0^1 L_i^{n-1}(\tau) R(\tau)\,\mathrm{d}\tau \quad \text{for} = \sum_{k=0}^{n} \sigma_k L_i^{n-1}(\tau_k) R(\tau_k)$$

$$= \sigma_0 L_i^{n-1}(\tau_0)\,R(\tau_0) + \sigma_i R(\tau_i) \quad \text{for } 1 \leqslant i \leqslant n. \tag{40}$$

TABLE 2

Collocation points and weights

μ	τ_1	τ_2	τ_3	σ_0	σ_1	σ_2	σ_3	σ_4
(a) *For* $n = 4$ $(\tau_0 = 0, \tau_4 = 1)$								
0·1	0·220862	0·614595	0·956751	0·003575	0·024254	0·059213	0·555467	0·357491
0·2	0·215176	0·598474	0·924445	0·009728	0·065412	0·152945	0·528705	0·243210
0·3	0·209515	0·582742	0·900050	0·015532	0·102947	0·226194	0·482747	0·172580
0·4	0·203909	0·567772	0·881380	0·020775	0·135096	0·275852	0·438431	0·129847
0·5	0·198385	0·553793	0·866869	0·025707	0·163339	0·308429	0·399696	0·102829
0·6	0·192967	0·540911	0·855408	0·030518	0·188826	0·329478	0·366404	0·084773
0·7	0·187673	0·529143	0·846209	0·035313	0·212185	0·342703	0·337730	0·072068
0·8	0·182519	0·518451	0·838712	0.040145	0.233749	0·350494	0·312887	0·062726
0·9	0·177517	0·508765	0·832516	0·045037	0·253714	0·354424	0·291225	0·055601
1.0	0·172673	0·500000	0·827327	0·050000	0·272222	0·355556	0·272222	0·050000

μ	τ_1	τ_2	τ_3	τ_4	σ_0	σ_1	σ_2	σ_3	σ_4	σ_5
(b) *For* $n = 5$ $(\tau_0 = 0, \tau_5 = 1)$										
0·1	0·142675	0·425129	0·738515	0·966409	0·003114	0·020292	0·042556	0·086406	0·536272	0·311361
0·2	0·139760	0·416410	0·723157	0·942896	0·007453	0·048326	0·099563	0·189373	0·468949	0·186335
0·3	0·136855	0·407818	0·708591	0·926224	0·011096	0·071345	0·143002	0·249291	0·401984	0·123284
0·4	0·133969	0·399461	0·695276	0·914152	0·014338	0·091165	0·176414	0·279793	0·348675	0·089614
0·5	0·131110	0·391423	0·683387	0·905191	0·017442	0·109366	0·203169	0·293538	0·306718	0·069766
0·6	0·128287	0·383765	0·672914	0·898368	0·020524	0·126619	0·224941	0·297703	0·273201	0·057012
0·7	0·125506	0·376521	0·663748	0·893049	0·023637	0·143173	0·242663	0·296285	0·246001	0·048240
0·8	0·122773	0·369710	0·655743	0·888811	0·026804	0·159115	0·256985	0·291597	0·223618	0·041881
0·9	0·120094	0·363334	0·648747	0·885369	0·030035	0·174468	0·268427	0·285027	0·204964	0·037080
1·0	0·117472	0·357384	0·642616	0·882528	0·033333	0·189238	0·277429	0·277429	0·189237	0·033333

μ	τ_1	τ_2	τ_3	τ_4	τ_5	σ_0	σ_1	σ_2	σ_3	σ_4	σ_5	σ_6
(c) *For* $n = 6$ ($\tau_0 = 0$, $\tau_6 = 1$)												
0·1	0·099662	0·308023	0·568484	0·811317	0·972846	0·002687	0·017186	0·034035	0·058934	0·110231	0·508191	0·268736
0·2	0·097972	0·302798	0·558827	0·797462	0·955062	0·005779	0·036834	0·072205	0·121773	0·208546	0·410390	0·144475
0·3	0·096287	0·297627	0·549451	0·784768	0·943195	0·008221	0·052110	0·100512	0·162991	0·250376	0·334448	0·091343
0·4	0·094610	0·292551	0·540539	0·773651	0·935012	0·010424	0·065579	0·123820	0·191414	0·263661	0·279954	0·065149
0·5	0·092944	0·287604	0·532207	0·764145	0·929160	0·012570	0·078344	0·144173	0·211447	0·263470	0·239716	0·050280
0·6	0·091293	0·282816	0·524514	0·756099	0·924824	0·014726	0·090771	0·162250	0·225318	0·256903	0·209126	0·040906
0·7	0·089659	0·278207	0·517474	0·749299	0·921510	0·016919	0·102977	0·178313	0·234497	0·247459	0·185305	0·034529
0·8	0·088046	0·273792	0·511067	0·743539	0·918910	0·019161	0·114991	0·192519	0·240093	0·236930	0·166367	0·029939
0·9	0·086455	0·269580	0·505258	0·738631	0·916822	0·021457	0·126809	0·204996	0·242973	0·226242	0·151033	0·026490
1·0	0·084888	0·265576	0·500000	0·734424	0·915112	0·023809	0·138413	0·215873	0·243810	0·215873	0·138413	0·023810

In matrix form, the n equations can be expressed as

$$\left[\begin{array}{c|cccc} \sigma_0 L_1^{n-1}(\tau_0) & \sigma_1 & & & \\ \sigma_0 L_2^{n-1}(\tau_0) & & \sigma_2 & & \\ \vdots & & & \ddots & \\ \sigma_0 L_n^{n-1}(\tau_0) & & & & \sigma_n \end{array}\right] \left\{\begin{array}{c} R(\tau_0) \\ \hline R(\tau_1) \\ \vdots \\ R(\tau_{n-1}) \\ R(\tau_n) \end{array}\right\} = \left\{\begin{array}{c} 0 \\ 0 \\ \vdots \\ 0 \end{array}\right\}. \tag{41}$$

Equation (41) establishes the relations to the residuals at the $n+1$ collocation points $\tau_0, \ldots, \tau_n$ in order to generate algorithms that are unconditionally A-stable and $(2n-1)$th order accurate. Note that the residuals $R(\tau_0), \ldots, R(\tau_n)$ do not vanish individually.

If $\sigma_0 = 0$, the formulation in equation (41) is equivalent to the conventional collocation method with n collocation points at $\tau_1, \ldots, \tau_n$. In this case, the weights $\sigma_1, \ldots, \sigma_n$ are not required in the formulation as usual.

5. NON-LINEAR PROBLEMS

When the algorithms are cast in the collocation form, the algorithms can be extended to solve the non-linear problem directly. Suppose the governing equation is given by

$$F(\ddot{u}, \dot{u}, u, \tau) = 0 \tag{42}$$

with initial conditions $u(0) = u_0$ and $\dot{u}(0) = v_0 \Delta t$. The residual at $\tau = \tau_k$ is $R(\tau_k) = F(\ddot{u}_k, \dot{u}_k, u_k, \tau_k)$ where $\ddot{u}_k, \dot{u}_k, u_k$ are the approximate solutions for $\ddot{u}, \dot{u}, u$ at $\tau = \tau_k$. $R(\tau_k)$ can be evaluated if $\ddot{u}_k$ and $\dot{u}_k$ can be expressed in terms of u_k directly. The relations can be obtained as follows. Let

$$u(\tau) = u_0 J_u^n(\tau) + v_0 \Delta t J_v^n(\tau) + u_1 J_1^n(\tau) + \cdots + u_n J_n^n(\tau), \tag{43}$$

where the functions $J_u^n(\tau)$, $J_v^n(\tau)$, $J_1^n(\tau), \ldots, J_n^n(\tau)$ are chosen such that

$$J_u^n(\tau_0) = 1, \quad \dot{J}_u^n(\tau_0) = 0, \quad J_v^n(\tau_0) = 0, \quad \dot{J}_v^n(\tau_0) = 1, \tag{44a}$$

$$J_u^n(\tau_i) = J_v^n(\tau_i) = 0, \quad \text{for } i = 1, \ldots, n, \tag{44b}$$

$$J_k^n(\tau_i) = \delta_{ik} \quad \text{for } i = 0, 1, \ldots, n \text{ and } k = 1, \ldots, n, \text{ and } \dot{J}_k^n(\tau_0) = 0, \quad k = 1, \ldots, n, \tag{44c}$$

where $\delta_{ik} = 1$ if $i = k$ and 0 otherwise.

If the functions $J_u^n(\tau)$, $J_v^n(\tau)$ and $J_k^n(\tau)$ are constructed from polynomials of degree not exceeding $n+1$, they can be expressed as

$$J_u^n(\tau) = (1 - \dot{L}_0^n(\tau_0)(\tau - \tau_0))\, L_0^n(\tau), \qquad J_v^n(\tau) = (\tau - \tau_0) L_0^n(\tau), \tag{45a, b}$$

$$J_k^n(\tau) = \frac{\tau - \tau_0}{\tau_k - \tau_0}\, L_k^n(\tau) \quad \text{for } k = 1, \ldots, n, \tag{45c}$$

where $L_k^n(\tau)$ is the nth order Lagrange polynomial and is given by

$$L_k^n(\tau) = \prod_{\substack{j=0 \\ j \neq k}}^{n} \frac{\tau - \tau_j}{\tau_k - \tau_j}. \tag{46}$$

In this case, $\ddot{u}_k$ and $\dot{u}_k$ can be expressed in terms of u_k as

$$\begin{Bmatrix} \dot{u}_0 \\ \vdots \\ \dot{u}_n \end{Bmatrix} = \begin{Bmatrix} B^{(1)}_{0u} \\ \vdots \\ B^{(1)}_{nu} \end{Bmatrix} u_0 + \begin{Bmatrix} B^{(1)}_{0v} \\ \vdots \\ B^{(1)}_{nv} \end{Bmatrix} v_0 \Delta t + \begin{bmatrix} B^{(1)}_{01} & \cdots & B^{(1)}_{0n} \\ \vdots & & \vdots \\ B^{(1)}_{n1} & \cdots & B^{(1)}_{nn} \end{bmatrix} \begin{Bmatrix} u_1 \\ \vdots \\ u_n \end{Bmatrix} \tag{47a}$$

and

$$\begin{Bmatrix} \ddot{u}_0 \\ \vdots \\ \ddot{u}_n \end{Bmatrix} = \begin{Bmatrix} B^{(2)}_{0u} \\ \vdots \\ B^{(2)}_{nu} \end{Bmatrix} u_0 + \begin{Bmatrix} B^{(2)}_{0v} \\ \vdots \\ B^{(2)}_{nv} \end{Bmatrix} v_0 \Delta t + \begin{bmatrix} B^{(2)}_{01} & \cdots & B^{(2)}_{0n} \\ \vdots & & \vdots \\ B^{(2)}_{n1} & \cdots & B^{(2)}_{nn} \end{bmatrix} \begin{Bmatrix} u_1 \\ \vdots \\ u_n \end{Bmatrix}, \tag{47b}$$

where

$$B^{(r)}_{iu} = \frac{\mathrm{d}^r}{\mathrm{d}\tau^r} J^n_u(\tau)\bigg|_{\tau=\tau_i}, \quad B^{(r)}_{iv} = \frac{\mathrm{d}^r}{\mathrm{d}\tau^r} J^n_v(\tau)\bigg|_{\tau=\tau_i} \quad \text{and} \quad B^{(r)}_{ij} = \frac{\mathrm{d}^r}{\mathrm{d}\tau^r} J^n_j(\tau)\bigg|_{\tau=\tau_i} \tag{48}$$

for $r = 1$ and 2 and can be expressed as

$$B^{(r)}_{iu} = (1 - (\tau_i - \tau_0) A^{(1)}_{00}) A^{(r)}_{i0} - r A^{(1)}_{00} A^{(r-1)}_{i0}, \quad i = 0, \dots, n,$$

$$B^{(r)}_{iv} = (\tau_i - \tau_0) A^{(r)}_{i0} + r A^{(r-1)}_{i0}, \quad i = 0, \dots, n,$$

$$B^{(r)}_{ij} = \frac{\tau_i - \tau_0}{\tau_j - \tau_0} A^{(r)}_{ij} + \frac{r}{\tau_j - \tau_0} A^{(r-1)}_{ij}, \quad i = 0, \dots, n \quad \text{and} \quad j = 1, \dots, n \tag{49}$$

and $A^{(1)}_{ik}$ and $A^{(2)}_{ik}$ can be conveniently obtained from

$$A^{(2)}_{ik} = \frac{\mathrm{d}^2}{\mathrm{d}\tau^2} L^n_j(\tau)\bigg|_{\tau=\tau_k} = 2\left[A^{(1)}_{ii} A^{(1)}_{ik} - \frac{A^{(1)}_{ik}}{x_i - x_k} \right] \quad \text{for } i \neq k, \quad 0 \leqslant i, k \leqslant n \quad \text{and}$$

$$A^{(1)}_{ik} = \frac{P'(\tau_i)}{(\tau_i - \tau_k) P'(\tau_k)} \quad \text{for } i \neq k,\ 0 \leqslant i, k \leqslant n \quad \text{and}$$

$$A^{(r)}_{ii} = -\sum_{\substack{j=0 \\ j \neq i}}^{n} A^{(r)}_{ij} \quad \text{for } 1 \leqslant r \leqslant 2, \quad 0 \leqslant i \leqslant n, \tag{50}$$

where $P'(\tau)$ is the first derivative of $P(\tau)$ in equation (30) or

$$P'(\tau) = (n+1)\tau^n - s_1 - 2s_2\tau - 3s_3\tau^2 - \cdots - n s_n \tau^{n-1}. \tag{51}$$

When $\mu = 1$, the collocation points are equivalent to the Lobatto quadrature points. However, the present algorithms are different from the direct algorithms in reference [23] since the present algorithms are A-stable while the direct algorithms in reference [23] are only conditionally stable.

6. NUMERICAL EXAMPLES

6.1. EXAMPLE 1: LINEAR SECOND ORDER EQUATIONS

Consider the free vibration of a single-degree-of-freedom system governed by

$$\ddot{u}(\tau) + 2\xi\omega\Delta t\dot{u}(\tau) + \omega^2 \Delta t^2 u(\tau) = 0 \tag{52}$$

with initial condition $u(0) = u_0$ and $\dot{u}(0) = v_0 \Delta t$. Consider the present third order accurate algorithm with $n = 2$. It can be verified that from equation (15)

$$a_1 = -\frac{1}{3}\frac{1}{1+\mu}, \quad a_2 = \frac{2}{3}\frac{2+\mu}{1+\mu}, \quad b_1 = -\frac{1}{3}\frac{2+\mu}{(1+\mu)^2} \quad \text{and} \quad b_2 = \frac{1}{3}\frac{5+5\mu+2\mu^2}{(1+\mu)^2}. \tag{53}$$

Hence, the required J_1, J_2, J_3, J_4 and J_5 obtained from equation (24) for the reduced integration are

$$J_1 = 1, \quad J_2 = \frac{1}{2} + \frac{1-\mu^2}{2(1+\mu+\mu^2)}, \quad J_3 = \frac{1}{3} + \frac{(1-\mu)(2+\mu)}{3(1+\mu+\mu^2)},$$
$$J_4 = \frac{1}{4} + \frac{(1-\mu)(3+3\mu+\mu^2)}{4(1+\mu)(1+\mu+\mu^2)}, \quad J_5 = \frac{(1+\mu)^2+1}{2(1+\mu)^2(1+\mu+\mu^2)}. \tag{54}$$

It can be seen that if $\mu = 1$, $J_k = 1/k$ for $1 \leqslant k \leqslant 4$. In other words, the present reduced integration rule can be used to integrate polynomials of degree not exceeding 3 correctly. If $\mu \neq 1$, only J_1 is correctly integrated. For example, if $\mu = 1/2$, $J_1, \ldots, J_5$ are given by

$$J_1 = 1, \quad J_2 = \tfrac{5}{7}, \quad J_3 = \tfrac{4}{7}, \quad J_4 = \tfrac{10}{21}, \quad J_5 = \tfrac{26}{63}. \tag{55}$$

However, this does not affect the validity of the present formulation. The approximate solutions still have very good accuracy.

The collocation points and the corresponding weights are given in equation (34) as

$$\tau_0 = 0, \quad \tau_1 = \frac{1}{1+\mu}, \quad \tau_2 = 1, \quad \sigma_0 = \frac{\mu^2}{2(1+\mu+\mu^2)}, \quad \sigma_1 = \frac{1+2\mu+\mu^2}{2(1+\mu+\mu^2)}, \quad \sigma_2 = \frac{1}{2(1+\mu+\mu^2)}. \tag{56}$$

From equation (37), $L_1^1(\tau_0) = (1+\mu)/\mu$ and $L_2^1(\tau_0) = -1/\mu$. From equation (43), the approximate solution is given by

$$u(\tau) = u_0\left(\frac{(\tau-\tau_1)(\tau-\tau_2)}{(\tau_0-\tau_1)(\tau_0-\tau_2)} - \frac{(\tau-\tau_0)(\tau-\tau_1)(\tau-\tau_2)}{(\tau_0-\tau_1)^2(\tau_0-\tau_2)^2}(2\tau_0-\tau_1-\tau_2)\right) \tag{57}$$
$$+ v_0\Delta t\,\frac{(\tau-\tau_0)(\tau-\tau_1)(\tau-\tau_2)}{(\tau_0-\tau_1)(\tau_0-\tau_2)} + u_1\frac{(\tau-\tau_0)^2(\tau-\tau_2)}{(\tau_1-\tau_0)(\tau_1-\tau_2)} + u_2\frac{(\tau-\tau_0)^2(\tau-\tau_1)}{(\tau_2-\tau_0)(\tau_2-\tau_1)}.$$

The residuals $R(\tau_0)$, $R(\tau_1)$ and $R(\tau_2)$ at τ_0, τ_1 and τ_2 are given by

$$\begin{Bmatrix} R(\tau_0) \\ R(\tau_1) \\ R(\tau_2) \end{Bmatrix} = \begin{Bmatrix} \ddot{u}_0 \\ \ddot{u}_1 \\ \ddot{u}_2 \end{Bmatrix} + 2\xi\omega\Delta t \begin{Bmatrix} \dot{u}_0 \\ \dot{u}_1 \\ \dot{u}_2 \end{Bmatrix} + \omega^2\Delta t^2 \begin{Bmatrix} u_0 \\ u_1 \\ u_2 \end{Bmatrix}, \tag{58}$$

where from equation (47)

$$\begin{Bmatrix} \dot{u}_0 \\ \dot{u}_1 \\ \dot{u}_2 \end{Bmatrix} = \begin{Bmatrix} 0 \\ -\dfrac{(3+2\mu)\mu}{(1+\mu)} \\ (3+\mu)\mu \end{Bmatrix} u_0 + \begin{Bmatrix} 1 \\ -\dfrac{\mu}{(1+\mu)} \\ \mu \end{Bmatrix} v_0\Delta t + \frac{1}{\mu}\begin{bmatrix} 0 & 0 \\ (2\mu-1)(1+\mu) & \dfrac{1}{(1+\mu)} \\ -(1+\mu)^3 & 3\mu+1 \end{bmatrix} \begin{Bmatrix} u_1 \\ u_2 \end{Bmatrix}, \tag{59a}$$

$$\begin{Bmatrix}\ddot{u}_0\\\ddot{u}_1\\\ddot{u}_2\end{Bmatrix}=\begin{Bmatrix}-2\mu^2-6\mu-6\\-2\mu^2+6\\4\mu^2+12\mu+6\end{Bmatrix}u_0+\begin{Bmatrix}-2\mu-4\\-2\mu+2\\4\mu+2\end{Bmatrix}v_0\Delta t+\frac{1}{\mu}\begin{bmatrix}2(1+\mu)^3 & -2\\2(\mu-2)(1+\mu)^2 & 4\\-4(1+\mu)^3 & 6\mu+4\end{bmatrix}\begin{Bmatrix}u_1\\u_2\end{Bmatrix}. \tag{59b}$$

The two equations to solve for u_1 and u_2 are hence given by equation (41) as

$$\left[\begin{array}{c|cc}\sigma_0 L_1^1(\tau_0) & \sigma_1 & \\ \sigma_0 L_2^1(\tau_0) & & \sigma_2\end{array}\right]\left\{\begin{array}{c}R(\tau_0)\\\hdashline R(\tau_1)\\R(\tau_2)\end{array}\right\}=\begin{Bmatrix}0\\0\end{Bmatrix}. \tag{60}$$

It can be verified that $u(\tau)$ and $v(\tau)$ ($= u_{,t}(\tau) = \dot{u}(\tau)/\Delta t$) at the end of the time step are related to u_0 and v_0 by

$$u(\tau=1)=A_{11}u_0+A_{12}v_0, \qquad v(\tau=1)=A_{21}u_0+A_{22}v_0, \tag{61a, b}$$

where (with $\rho=\omega\Delta t$)

$$A_{11}=$$
$$\frac{\mu\rho^4-4(2\mu+1)\xi\rho^3-2(7\mu^2+16\mu+7-12(\mu+1)\xi^2)\rho^2+24(\mu+1)(\mu+2)\xi\rho+36(1+\mu)^2}{\rho^4+4(\mu+2)\xi\rho^3+4(\mu^2+\mu+1+6(\mu+1)\xi^2)\rho^2+24(\mu+1)(\mu+2)\xi\rho+36(1+\mu)^2}, \tag{62a}$$

$$A_{12}=$$
$$\frac{-(2(\mu^2+4\mu+1)\rho^2+12(\mu-1)(\mu+1)\xi\rho-36(1+\mu)^2)\rho/\omega}{\rho^4+4(\mu+2)\xi\rho^3+4(\mu^2+\mu+1+6(\mu+1)\xi^2)\rho^2+24(\mu+1)(\mu+2)\xi\rho+36(1+\mu)^2}, \tag{62b}$$

$$A_{21}=$$
$$\frac{(2(\mu^2+4\mu+1)\rho^2+12(\mu-1)(\mu+1)\xi\rho-36(1+\mu)^2)\rho\omega}{\rho^4+4(\mu+2)\xi\rho^3+4(\mu^2+\mu+1+6(\mu+1)\xi^2)\rho^2+24(\mu+1)(\mu+2)\xi\rho+36(1+\mu)^2}, \tag{62c}$$

$$A_{22}=$$
$$\frac{\mu\rho^4+4\mu(\mu+2)\xi\rho^3-2(7\mu^2+16\mu+7-12\mu(\mu+1)\xi^2)\rho^2-24(\mu+1)(2\mu+1)\xi\rho+36(1+\mu)^2}{\rho^4+4(\mu+2)\xi\rho^3+4(\mu^2+\mu+1+6(\mu+1)\xi^2)\rho^2+24(\mu+1)(\mu+2)\xi\rho+36(1+\mu)^2}. \tag{62d}$$

It can be verified that the results are indeed third order accurate as

$$u(\tau=1)-\bar{u}(\tau=1)=\frac{1-\mu}{72(1+\mu)}((4\xi^2-1)\omega u_0+4\xi(2\xi^2-1)v_0)\omega^3\Delta t^4+\cdots, \tag{63}$$

$$v(\tau=1)-\bar{v}(\tau=1)=\frac{\mu-1}{72(1+\mu)}(4\xi(2\xi^2-1)\omega u_0+(16\xi^4-12\xi^2+1)v_0)\omega^4\Delta t^4+\cdots, \tag{64}$$

where $\bar{u}(\tau)$ and $\bar{v}(\tau)$ are the exact solutions for equation (52). If $\mu = 1$, the algorithm is fourth order accurate.

6.1.1. *Conventional algorithms*

Consider the conventional algorithm using two collocation points $\tau_1 = \beta$ and $\tau_2 = 1$. It can be verified that the truncation errors for $u(\tau)$ and $v(\tau)$ at the end of the time interval ($\tau = 1$) are

$$u(\tau = 1) - \bar{u}(\tau = 1) = \tfrac{1}{24}(4\beta - 1)((4\xi^2 - 1)\omega u_0 + 4\xi(2\xi^2 - 1)v_0)\,\omega^3 \Delta t^4 + O(\Delta t^5), \tag{65a}$$

$$v(\tau = 1) - \bar{v}(\tau = 1) = \tfrac{1}{12}(3\beta - 1)((4\xi^2 - 1)\omega u_0 + 4\xi(2\xi^2 - 1)v_0)\,\omega^3 \Delta t^3 + O(\Delta t^4), \tag{65b}$$

As a result, the algorithm is in general second order accurate only due to the poor accuracy of $v(\tau = 1)$ in equation (65b). If $\beta = \frac{1}{3}$, the truncation error for $v(\tau = 1)$ can be improved to

$$v(\tau = 1) - \bar{v}(\tau = 1) = (\tfrac{1}{54}\,\xi(1 - 6\xi^2)\omega u_0 + \tfrac{1}{216}(1 + 20\xi^2 - 48\xi^4)v_0)\,\omega^4 \Delta t^4 + O(\Delta t^5). \tag{66}$$

Hence, the algorithm is third order accurate if $\beta = \frac{1}{3}$. However, the conventional algorithm is only conditionally stable. In fact, it can be shown that the algorithms are unconditionally stable if $1 > \beta \geqslant 0{\cdot}5948$. Otherwise, it is only conditionally stable and it is required that

$$\omega \Delta t \leqslant \frac{\sqrt{2\beta(1 + 4\beta - 2\beta^2 - \sqrt{1 - 4\beta + 12\beta^2 - 16\beta^3 + 4\beta^4})}}{\beta} \quad \text{for } 0 < \beta < 0.5948 \tag{67}$$

to maintain numerical stability. For example, if $\beta = 1/2$, $\omega \Delta t \leqslant 2\sqrt{2}$ and if $\beta = 1/3$, $\omega \Delta t \leqslant (\sqrt{37} - 1)/\sqrt{3}$. In other words, in using $\tau_1 = \frac{1}{2}$ and $\tau_2 = 1$, the present algorithm would be fourth order accurate (since $\mu = 1$) and unconditionally A-stable while the conventional algorithm is only second order accurate and conditionally stable. It can be seen clearly that the present algorithms are better than the conventional algorithms in terms of stability and accuracy.

6.1.2. *Two-point collocation*

If the two collocation points are chosen arbitrarily as τ_1 and τ_2, the stability and accuracy properties of the resulting algorithm can be studied by evaluating the parameters given in reference [25]. The two weighting functions are $\psi_1(\tau) = \delta(\tau - \tau_1)$ and $\psi_2(\tau) = \delta(\tau - \tau_2)$. Hence, the normalized weighting parameters are given by

$$\begin{bmatrix} \hat{W}_{12} & \hat{W}_{13} \\ \hat{W}_{22} & \hat{W}_{23} \end{bmatrix} = \begin{bmatrix} 1 & \tau_1 \\ 1 & \tau_2 \end{bmatrix}^{-1} \begin{bmatrix} \tau_1^2 & \tau_1^3 \\ \tau_2^2 & \tau_2^3 \end{bmatrix}. \tag{68}$$

It has been shown in reference [25] that for the algorithm to be fourth order accurate, it is required that

$$\hat{W}_{12} = -\tfrac{1}{6}, \quad \hat{W}_{22} = 1 \quad \text{and} \quad \hat{W}_{23} = \tfrac{1}{2} - 2\hat{W}_{13}. \tag{69a–c}$$

From equations (69a) and (69b), τ_1 and τ_2 should be the zeros of the equation

$$6\tau^2 - 6\tau + 1 = 0. \tag{70}$$

Hence, τ_1 and τ_2 are the Gauss quadrature points given as

$$\tau_1 = \frac{1}{2} - \frac{\sqrt{3}}{6} \quad \text{and} \quad \tau_2 = \frac{1}{2} + \frac{\sqrt{3}}{6}. \tag{71}$$

It can then be verified that $\hat{W}_{13} = -\frac{1}{6}$ and $\hat{W}_{23} = \frac{5}{6}$. Since $\hat{W}_{13}$ and $\hat{W}_{23}$ satisfy equations (69c), the algorithm is fourth order accurate. However, the algorithm is only conditionally stable, as shown in reference [25]. Hence, it can be seen that it is not possible to construct fourth order accurate algorithms from the direct collocation method.

6.2. EXAMPLE 2 : NON-LINEAR PROBLEMS

Consider a simple pendulum consisting of a mass attached to a hinged weightless rod of length L. The equation of motion can be written as

$$\frac{d^2}{dt^2}\Theta + \omega^2 \sin(\Theta) = 0, \tag{72}$$

where $\Theta(t)$ is the angle between the rod and the vertical at time t,

$$\omega = \sqrt{\frac{g}{L}} \text{ and } g \text{ is the gravity acceleration.} \tag{73}$$

The non-linearity of the problem is due to large motion and is represented by the $\sin(\Theta)$ term in equation (72). If the initial angular velocity and the initial angular displacement are denoted as Ω_0 and Θ_0, respectively, the analytical relation between the angular displacement and time is given by

$$t = \pm \int_{\Theta_0}^{\Theta} \frac{1}{\sqrt{\Omega_0^2 + 2\omega^2(\cos(\theta) - \cos(\Theta_0))}} d\theta. \tag{74}$$

Without loss of generality, assumes $\Theta_0 = 0$ and equation (74) then reduces to

$$t = \pm \frac{1}{|\Omega_0|} \int_0^{\Theta} \frac{d\theta}{\sqrt{1 - \kappa^2 \sin^2(\theta/2)}} = \frac{2}{|\Omega_0|} Ei\left(\sin\left(\frac{\Theta}{2}\right), \kappa\right), \tag{75}$$

where $\kappa = 2\omega/\Omega_0$ and $Ei(z, \kappa)$ is the elliptical integral of the first kind. If $\kappa < 1$, the pendulum would rotate around the pivot point and Θ would increase indefinitely with time. If $\kappa \geqslant 1$, the pendulum would oscillate about the vertical equilibrium position with $|\Theta| \leqslant \Theta_{max}$ where Θ_{max} is given by

$$\Theta_{max} = 2\sin^{-1}\left(\frac{\Omega_0}{2\omega}\right) = 2\sin^{-1}\left(\frac{1}{\kappa}\right). \tag{76}$$

Equation (72) can be re-written in a non-dimensional form as

$$\frac{d^2}{d\tau^2}\Theta + \omega^2 \Delta t^2 \sin(\Theta) = 0. \tag{77}$$

The residuals at $\tau_0, \tau_1, \ldots, \tau_n$ over a time interval Δt are given by

$$\begin{Bmatrix} R(\tau_0) \\ R(\tau_1) \\ \vdots \\ R(\tau_n) \end{Bmatrix} = \begin{Bmatrix} \ddot{\Theta}_0 \\ \ddot{\Theta}_1 \\ \vdots \\ \ddot{\Theta}_n \end{Bmatrix} + \omega^2 \Delta t^2 \begin{Bmatrix} \sin(\Theta_0) \\ \sin(\Theta_1) \\ \vdots \\ \sin(\Theta_n) \end{Bmatrix} = \begin{Bmatrix} 0 \\ 0 \\ \vdots \\ 0 \end{Bmatrix}, \tag{78}$$

where

$$\Theta(\tau) = \Theta_0\, J_u^n(\tau) + \Omega_0 \Delta t J_v^n(\tau) + \Theta_1 J_1^n(\tau) + \cdots + \Theta_n J_n^n(\tau), \tag{79a}$$

$$\dot{\Theta}(\tau) = \Theta_0\, \dot{J}_u^n(\tau) + \Omega_0 \Delta t \dot{J}_v^n(\tau) + \Theta_1 \dot{J}_1^n(\tau) + \cdots + \Theta_n \dot{J}_n^n(\tau), \tag{79b}$$

$$\ddot{\Theta}(\tau) = \Theta_0\, \ddot{J}_u^n(\tau) + \Omega_0 \Delta t \ddot{J}_v^n(\tau) + \Theta_1 \ddot{J}_1^n(\tau) + \cdots + \Theta_n \ddot{J}_n^n(\tau). \tag{79c}$$

and

$$\begin{Bmatrix} \dot{\Theta}_0 \\ \dot{\Theta}_1 \\ \vdots \\ \dot{\Theta}_n \end{Bmatrix} = \begin{Bmatrix} B_{0u}^{(1)} \\ B_{1u}^{(1)} \\ \vdots \\ B_{nu}^{(1)} \end{Bmatrix} \Theta_0 + \begin{Bmatrix} B_{0v}^{(1)} \\ B_{1v}^{(1)} \\ \vdots \\ B_{nv}^{(1)} \end{Bmatrix} \Omega_0 \Delta t + \begin{bmatrix} B_{01}^{(1)} & \cdots & B_{0n}^{(1)} \\ B_{11}^{(1)} & \cdots & B_{1n}^{(1)} \\ \vdots & & \vdots \\ B_{n1}^{(1)} & \cdots & B_{nn}^{(1)} \end{bmatrix} \begin{Bmatrix} \Theta_1 \\ \vdots \\ \Theta_n \end{Bmatrix} \tag{80a}$$

and

$$\begin{Bmatrix} \ddot{\Theta}_0 \\ \ddot{\Theta}_1 \\ \vdots \\ \ddot{\Theta}_n \end{Bmatrix} = \begin{Bmatrix} B_{0u}^{(2)} \\ B_{1u}^{(2)} \\ \vdots \\ B_{nu}^{(2)} \end{Bmatrix} \Theta_0 + \begin{Bmatrix} B_{0v}^{(2)} \\ B_{1v}^{(2)} \\ \vdots \\ B_{nv}^{(2)} \end{Bmatrix} \Omega_0 \Delta t + \begin{bmatrix} B_{01}^{(2)} & \cdots & B_{0n}^{(2)} \\ B_{11}^{(2)} & \cdots & B_{1n}^{(2)} \\ \vdots & & \vdots \\ B_{n1}^{(2)} & \cdots & B_{nn}^{(2)} \end{bmatrix} \begin{Bmatrix} \Theta_1 \\ \vdots \\ \Theta_n \end{Bmatrix}. \tag{80b}$$

Hence,

$$\begin{Bmatrix} R(\tau_0) \\ R(\tau_1) \\ \vdots \\ R(\tau_n) \end{Bmatrix} = \begin{Bmatrix} B_{0u}^{(2)} \\ B_{1u}^{(2)} \\ \vdots \\ B_{nu}^{(2)} \end{Bmatrix} \Theta_0 + \begin{Bmatrix} B_{0v}^{(2)} \\ B_{1v}^{(2)} \\ \vdots \\ B_{nv}^{(2)} \end{Bmatrix} \Omega_0 \Delta t + \begin{bmatrix} B_{01}^{(2)} & \cdots & B_{0n}^{(2)} \\ B_{11}^{(2)} & \cdots & B_{1n}^{(2)} \\ \vdots & & \vdots \\ B_{n1}^{(2)} & \cdots & B_{nn}^{(2)} \end{bmatrix} \begin{Bmatrix} \Theta_1 \\ \vdots \\ \Theta_n \end{Bmatrix} \tag{81}$$

$$+ \omega^2 \Delta t^2 \begin{Bmatrix} \sin(\Theta_0) \\ \sin(\Theta_1) \\ \vdots \\ \sin \Theta_n \end{Bmatrix}.$$

The n equations to solve for $\Theta_1, \ldots, \Theta_n$ are given by

$$\{\mathbf{r}\} = [\mathbf{\Lambda}]\{\mathbf{B}_u^{(2)}\}\Theta_0 + [\mathbf{\Lambda}]\{\mathbf{B}_v^{(2)}\}\Omega_0 \Delta t + [\mathbf{\Lambda}][\mathbf{B}^{(2)}]\{\mathbf{\Theta}\} + [\mathbf{\Lambda}]\{\mathbf{S}_0\}\omega^2 \Delta t^2 = \{\mathbf{0}\}, \tag{82}$$

where

$$[\mathbf{\Lambda}] = \left[\begin{array}{c|cccc} \sigma_0 L_1^{n-1}(\tau_0) & \sigma_1 & & & \\ \sigma_0 L_2^{n-1}(\tau_0) & & \sigma_2 & & \\ \vdots & & & \ddots & \\ \sigma_0 L_n^{n-1}(\tau_0) & & & & \sigma_n \end{array}\right], \quad \{\mathbf{B}_u^{(2)}\} = \begin{Bmatrix} B_{0u}^{(2)} \\ B_{1u}^{(2)} \\ \vdots \\ B_{nu}^{(2)} \end{Bmatrix}, \quad \{\mathbf{B}_v^{(2)}\} = \begin{Bmatrix} B_{0v}^{(2)} \\ B_{1v}^{(2)} \\ \vdots \\ B_{nv}^{(2)} \end{Bmatrix},$$

$$[\mathbf{B}^{(2)}] = \begin{bmatrix} B_{01}^{(2)} & \cdots & B_{0n}^{(2)} \\ B_{11}^{(2)} & \cdots & B_{1n}^{(2)} \\ \vdots & & \vdots \\ B_{n1}^{(2)} & \cdots & B_{nn}^{(2)} \end{bmatrix}, \quad \{\mathbf{\Theta}\} = \begin{Bmatrix} \Theta_1 \\ \vdots \\ \Theta_n \end{Bmatrix}, \quad \{\mathbf{S}_0\} = \begin{Bmatrix} \sin(\Theta_0) \\ \sin(\Theta_1) \\ \vdots \\ \sin(\ddot{\Theta}_n) \end{Bmatrix}. \tag{83}$$

An iterative procedure can be used to determine $\Theta_1, \ldots, \Theta_n$ for given Θ_0 and Ω_0. If the Newton–Raphson method is used, the incremental form of equation (82) is

$$([\mathbf{\Lambda}][\mathbf{B}^{(2)}] + \omega^2 \Delta t^2 \operatorname{diag}(\sigma_k \cos(\Theta_k)))\{\Delta\mathbf{\Theta}\} + \{\mathbf{r}\} = \{\mathbf{0}\}, \tag{84}$$

TABLE 3

Numerical results for Example 2 by various methods

Method/grid points	$T_f/\Delta t$	$\Theta(T_f)$	% Error in $\Theta(T_f)$
(a) $\Omega_0 = 1$, $T_f = 1{\cdot}6858$			
Exact	—	1·04720	—
$n = 2$, $\mu = 1$	1	1·05028	0·2940
$n = 2$, $\mu = 1$	2	1·04730	0·0097
2 Gauss points	1	1·05769	1·0020
2 Gauss points	2	1·04787	0·0645
[3/4, 1] (reference [23])	5	1·03737	– 0·9384
[3/4, 1] (reference [23])	10	1.04490	– 0.2195
[3/4, 1] (reference [23])	20	1·04664	– 0·0529
$n = 3$, $\mu = 1$	1	1·04719	– 0.0007
$n = 3$, $\mu = 1$	2	1·04721	0·0009
3 Gauss points	1	1·04727	0·0072
3 Gauss points	2	1·04719	– 0.0009
[– 1/5, 9/10, 1] (reference [23])	2	1·03933	– 0·7512
[– 1/5, 9/10, 1] (reference [23])	5	1·04664	– 0·0531
(b) $\Omega_0 = 1{\cdot}84776$, $T_f = 2{\cdot}4001$			
Exact	—	2·35619	—
$n = 2$, $\mu = 1$	1	2·57230	9·1717
$n = 2$, $\mu = 1$	3	2·35489	– 0·0553
$n = 2$, $\mu = 1$	5	2·35614	– 0·0022
2 Gauss points	1	2·22588	– 5·5307
2 Gauss points	3	2·35998	0·1608
2 Gauss points	5	2·35654	0·0149
$n = 3$, $\mu = 1$	1	2·32635	– 1·2667
$n = 3$, $\mu = 1$	2	2·35684	0·0273
$n = 3$, $\mu = 1$	3	2·35631	0·0050
3 Gauss points	1	2·40153	1·9240
3 Gauss points	2	2·35579	– 0·0170
3 Gauss points	3	2·35609	– 0·0046

where $\{\Delta\boldsymbol{\Theta}\} = [\Delta\Theta_1, \dots, \Delta\Theta_n]^{\mathrm{T}}$ contains the increments. A better approximation can be obtained by replacing Θ_k by $\Theta_k + \Delta\Theta_k$. The iteration is repeated until the residuals are acceptably small. The displacement and velocity at the end of the time step can be used as the initial conditions for the next time step. The procedure is repeated until the time range of interest, say from 0 to T_f is covered. Note that if $\sigma_0 = 0$, the present formulation is equivalent to the conventional method with collocation points at $\tau_1, \dots, \tau_n$.

Consider $\omega = 1$, the initial angular displacement $\Theta_0 = 0$ and the initial angular velocity $\Omega_0 = 1$. The maximum angular displacement is $\pi/3$ ($= 1{\cdot}0472$) with a period of the oscillation $4 \times 1{\cdot}6858$. The present method is used to find the response time history over a quarter of the period (i.e., $T_f = 1{\cdot}6858$). The results are shown in Table 3(a). It can be seen that with $n = 3$ and $\mu = 1$, the response can be evaluated very accurately by using just one time step (i.e., $\Delta t = T_f$). The conventional approaches using the Gauss points or other points suggested in reference [23] give results not as accurate. Using two or more time steps could improve the solution accuracy given by the conventional methods. However, the solutions given by using five time steps (i.e., $\Delta t = T_f/5$) are still not as accurate as the solutions given by the present method by using just one time step. The present algorithms are therefore very efficient in terms of computational effort.

The non-linearity of the problem increases with the amplitude Θ_{max}. In Table 3(b), $\Theta_0 = 0$ and the initial angular velocity Ω_0 is chosen to be $\sqrt{2+\sqrt{2}}$ ($= 1{\cdot}84776$) so that Θ_{max} is $3\pi/4$ ($= 2{\cdot}35619$). The period of the oscillation is now $4 \times 2{\cdot}4001$. The present method again can be used to solve the problem without difficulty. From the numerical results shown in Table 3(b), it can be seen that the present method again gives more accurate numerical results than the conventional methods. The present algorithms are still very efficient in terms of computational effort.

7. CONCLUSIONS

In this paper, unconditionally stable time step integration algorithms in the collocation form are constructed for the second order equations. The algorithms are derived from the weighted residual method with reduced integration. The approximate solutions are assumed to be polynomials of degree $n + 1$ satisfying the two given initial conditions. There are n unknowns to be solved from n equations. The n linearly independent weighting functions are assumed to be polynomials of degree not exceeding $n - 1$. The reduced integration rules are used to evaluate the integrals so that the algorithms with desirable characteristics can be generated. Once the reduced integration rules are decided, the formulations are recast into the collocation form. It is found that residuals at $n + 1$ collocation points are required to construct the n equations. Numerical examples are used to illustrate the validity, stability and accuracy properties of the present algorithms. It is found that the present formulation is better than the conventional formulations in terms of stability and accuracy.

REFERENCES

1. P. J. VAN DER HOUWEN 1977 *Construction of Integration Formulas for Initial Value Problems.* Amsterdam: North-Holland.
2. E. HAIRER, S. P. NORSETT and G. WANNER 1987 *Solving Ordinary Differential Equations* I. Berlin: Springer.
3. E. HAIRER and G. WANNER 1991 *Solving Ordinary Differential Equations* II. Berlin: Springer.
4. T. J. R. HUGHES 1987 *The Finite Element Method: Linear Static and Dynamic Finite Element Analysis.* Englewood Cliffs, NJ: Prentice-Hall.
5. W. L. WOOD 1990 *Practical Time-Stepping Schemes.* Oxford: Clarendon Press.
6. O. C. ZIENKIEWICZ and R. L. TAYLOR 1991 *The Finite Element Method.* New York: McGraw-Hill; fourth edition.
7. M. A. DOKAINISH and K. SUBBARAJ 1989 *Computers and Structures* **32**, 1371–1386. A survey of direct time-integration methods in computational structural dynamics—I. Explicit methods.
8. M. A. DOKAINISH and K. SUBBARAJ 1989 *Computers and Structures* **32**, 1387–1401. A survey of direct time-integration methods in computational structural dynamics—II. Implicit methods.
9. J. KUJAWSKI 1986 *International Journal for Numerical Methods in Engineering* **23**, 579–589. Collocation time finite element procedures for integration of strongly nonlinear initial value problems.
10. J. KUJAWSKI 1987 *Communications in Applied Numerical Methods* **3**, 103–107. Analysis of the collocation time finite element method for the nonlinear hear transfer equation.

11. T. C. FUNG 1999 *International Journal for Numerical Methods in Engineering* **45**, 941–970. Weighting parameters for unconditionally stable higher-order accurate time step integration algorithms—Part 1. First order equations.
12. T. C. FUNG 2000 *Computer Methods in Applied Mechanics and Engineering* **190**, 1651–1662. Unconditionally stable higher-order accurate collocation time step integration algorithms for first order equations.
13. T. C. FUNG 2001 *International Journal for Numerical Methods in Engineering* **50**, 1411–1427. Solving initial value problems by differential quadrature method—Part 1: First order equations.
14. E. L. WILSON 1968 *SESM Report No.* 68-1, *Division of Structural Engineering and Structural Mechanics, University of California, Berkeley.* A computer program for the dynamic stress analysis of underground structures.
15. H. M. HILBER and T. J. R. HUGHES 1978 *Earthquake Engineering and Structural Dynamics* **6**, 99–118. Collocation, dissipation and overshoot for time integration schemes in structural dynamics.
16. H. M. HILBER, T. J. R. HUGHES and R. L. TAYLOR 1977 *Earthquake Engineering and Structural Dynamics* **5**, 283–292. Improved numerical dissipation for time integration schemes in structural dynamics.
17. J. CHUNG and G. M. HULBERT 1993 *Journal of Applied Mechanics* **60**, 371–375. A time integration algorithm for structural dynamics with improved numerical dissipation: the generalized-α method.
18. J. C. HOUBOLT 1950 *Journal of the Aeronautical Sciences* **17**, 540–550. A recurrence matrix solution for the dynamic response of elastic aircraft.
19. J. H. ARGYRIS, L. E. VAZ and K. J. WILLAM 1977 *Computer Methods in Applied Mechanics and Engineering* **12**, 243–278. Higher order methods for transient diffusion analysis.
20. M. GELLERT 1978 *Computers and Structures* **9**, 401–408. A new algorithm for integration of dynamic systems.
21. D. A. PETERS and A. P. IZADPANAH 1988 *Computational Mechanics* **3**, 73–88. hp-version finite elements for the space-time domain.
22. L. KRAMARZ 1980 *BIT* **20**, 215–222. Stability of collocation methods for the numerical solution of $y'' = f(x, y)$.
23. P. J. VAN DER HOUWEN, B. P. SOMMEIJER and N. H. CONG 1991 *BIT* **31**, 469–481. Stability of collocation based Runge–Kutta–Nyström methods.
24. B. W. GOLLEY 1996 *International Journal for Numerical Methods in Engineering* **39**, 3985–3998. A time-stepping procedure for structural dynamics using Gauss point collocation.
25. T. C. FUNG 1996 *International Journal for Numerical Methods in Engineering* **39**, 3475–3495. Unconditionally stable higher-order accurate Hermitian time finite elements.
26. T. C. FUNG 2001 *International Journal for Numerical Methods in Engineering* **50**, 1429–1454. Solving initial value problems by differential quadrature method—Part 2. Second and higher order equations.
27. M. BORRI, G. L. GHIRINGHELLI, M. LANZ, P. MANTEGAZZA and T. MERLINI 1985 *Computers and Structures* **20**, 495–508. Dynamic response of mechanical systems by a weak Hamilton formulation.
28. T. C. FUNG 1999 *International Journal for Numerical Methods in Engineering* **45**, 971–1006. Weighting parameters for unconditionally stable higher-order accurate time step integration algorithms—Part 2. Second order equations.
29. T. C. FUNG 1998 *International Journal for Numerical Methods in Engineering* **41**, 65–93. Complex-time-step Newmark methods with controllable numerical dissipation.