AIAA JOURNAL
Vol. 38, No. 8, August 2000

Unconditionally Stable Time-Step-Integration Algorithms Based on Hamilton's Principle

T. C. Fung*
Nanyang Technological University, Singapore 639798, Republic of Singapore

Unconditionally stable time-step-integration algorithms derived from Hamilton's principle are presented. The corresponding constrained variations are assumed to be in the form of $\Sigma \alpha_k \tau^k (1-\tau)$ to ensure vanishing variations at the ends of a time interval. The initial conditions are strongly enforced in the formulation. The order of accuracy can be $2n-1$ or $2n$, where n is the number of unknown variables. The ultimate spectral radii of the algorithms can be controlled directly. The approximate solutions are in fact equivalent to the generalized Padé approximations for linear second-order differential equations. Hence, Hamilton's principle can be used to construct unconditionally stable higher-order-accurate time-step-integration algorithms directly as well, provided the appropriate constrained variations are used.

Nomenclature

$[\boldsymbol{A}]$	= amplification matrix
a_k, b_k	= algorithmic parameters
$[\boldsymbol{C}]$	= damping matrix
c	= damping coefficient
f_i	= path-dependent nonconservative forces
$\{\boldsymbol{f}_k\}, \{\boldsymbol{F}_k\}$	= force vectors
g, h	= approximate homogeneous solutions of a mass-spring system
g_d, h_d	= approximate time derivatives of the homogeneous solutions
$\bar{g}(t), \bar{h}(t)$	= exact homogeneous solutions of a mass-spring system
$[\boldsymbol{H}_{n,n}]$	= nonsingular $n \times n$ square matrix
I	= action
J_{1i}, J_{2i}	= elements in the last two rows of $[\boldsymbol{H}_{n+2,n+2}]^{-1}$
$[\boldsymbol{K}]$	= stiffness matrix
k	= spring force coefficient
$[\boldsymbol{M}]$	= mass matrix
m	= mass
N_1, N_2, N_3, N_4	= cubic Hermitian polynomials
n	= number of unknown variables
$q_i, \dot{q}_i$	= generalized coordinates: displacement, velocity
T	= kinetic energy
T	= period, $2\pi/\omega$
t	= time
t_0	= initial time
t_1	= final time
U_i	= unknown variables
u_i	= unknown variables
$u(t)$	= displacement, function of time t
u_0, v_0	= initial displacement and initial velocity at $t=0$
u_1, v_1	= final displacement and final velocity at $t=\Delta t$
V	= potential energy
$\{\boldsymbol{V}\}$	= coefficient vector for the stabilized variation functions
W_{ij}	= algorithmic parameters
$\{\boldsymbol{Z}\}$	= coefficient vector for the stabilized variation functions
α_{ik}	= unknown coefficients for the variation functions
Δt	= time interval, $t_1 - t_0$
δ	= variation operator
$\lambda_{1,2}$	= eigenvalues
λ_∞	= ultimate spectral radius
μ	= algorithmic parameter controlling the ultimate spectral radius
ξ	= critical damping ratio
ρ	= $\omega \Delta t$
τ	= nondimensional time, $t/\Delta t$
$\phi_1(\tau), \phi_2(\tau)$	= stabilizing functions
$\psi_i(\tau)$	= variation functions
ω	= circular frequency, rad/s

Received 24 January 1999; revision received 20 November 1999; accepted for publication 12 January 2000.

*Associate Professor, School of Civil and Structural Engineering, Nanyang Avenue.

I. Introduction

FRAMEWORKS to construct time-step-integration algorithms can be based on finite difference methods,[1,2] weighted residual methods,[3–5] and variational principles/statements (for example, Hamilton's principle,[6–8] Hamilton's law of varying action,[9,10] and Gurtin's principle[11,12]). The finite difference methods and the weighted residual methods operate on the governing differential equations directly. The variational principles/statements make full use of the existence of kinetic and potential energies of the system directly without the need to formulate the governing differential equations. Many established time-step-integration algorithms derived based on these very different frameworks are in fact equivalent.

The time-step-integrational gorithms are of particular interest because in many cases the responses of a mechanical system to a particular forcing function are required. The classical methods of obtaining numerical results include the direct analytical solution of the equations of motion and the use of modal analysis with the evaluation of Duhamel's integral. These methods become very tedious as the number of degrees of freedom increases. It can also become very difficult to use with awkward forcing functions.

By using the time-step-integration algorithms, however, the responses of both damped and undamped systems having many degrees of freedom to any forcing function are readily obtainable. The general procedure is to compute the displacement and velocity at the end of a time interval from the known initial displacement and velocity at the beginning of the time interval. The results then become the initial values for the next time interval. As a result, by using a step-by-step method numerical values of displacement and velocity can be calculated to cover the whole time domain of interest.

In the following, the time-step-integration algorithms based on Hamilton's principle and Hamilton's law of varying action are reviewed.

A. Hamilton's Principle

Hamilton's principle has traditionally been used in analytical mechanics as a method of obtaining the equations of motion for dynamical systems. Hamilton's principle states that for conservative systems the true motion between time t_0 and time t_1 will be such

that the action I is stationary (but not necessarily minimum[13]), i.e., $\delta I = 0$, where

$$I = \int_{t_0}^{t_1} (T - V)\, \mathrm{d}t \tag{1}$$

$T(q_i, \dot{q}_i, t)$ is the kinetic energy of the system and $V(q_i, t)$ is the potential energy of the system. Nonconservative forces can be included in the extended Hamilton's principle as

$$\delta \int_{t_0}^{t_1} (T - V)\, \mathrm{d}t + \int_{t_0}^{t_1} \sum_i f_i \delta q_i \, \mathrm{d}t = 0 \tag{2}$$

where f_i is the ith path-dependent nonconservative force that cannot be included in the potential energy V.

In using Hamilton's principle in Eq. (2), it is important to ensure that the variations $\delta q_i(t_0)$ and $\delta q_i(t_1)$ must vanish. The principle had been successfully used to obtain numerical solutions for stationary dynamic problems. Argyris and Scharpf,[6] Fried,[7] and Oden[8] had also used Eq. (2) to construct time finite elements. To avoid solving too many equations simultaneously, the suggestion was made to solve the problem stepwise, resulting in time-step-integration algorithms. For example, the displacement of a generalized coordinate $u(t)$ can be interpolated by

$$u(t) = u_0 N_1(\tau) + v_0 \Delta t N_2(\tau) + u_1 N_3(\tau) + v_1 \Delta t N_4(\tau) \tag{3}$$

where $N_1(\tau)$ to $N_4(\tau)$ are the usual cubic Hermitian polynomials and are defined as

$$N_1(\tau) = 2\tau^3 - 3\tau^2 + 1, \qquad N_2(\tau) = \tau^3 - 2\tau^2 + \tau$$
$$N_3(\tau) = -2\tau^3 + 3\tau^2, \qquad N_4(\tau) = \tau^3 - \tau^2 \tag{4}$$

For the variation operation in Eq. (2), the variation of $u(t)$ is usually assumed to be made up of $N_2(\tau)$ and $N_4(\tau)$ to ensure vanishing initial and final variations, i.e., $\delta u(t = t_0) = \delta u(t = t_1) = 0$. In the sense of calculus of variations, this is usually interpreted as the initial and final coordinates being specified. However, for initial value problems the initial displacement and initial velocity are usually given instead. As a result, it is a common practice to assume the configurations of the dynamic system being known at $t = t_0$ and t_1 initially so that the variations over there are zero. After the formulation is established, the actual initial displacement and initial velocity are introduced to make the problem well defined. The displacement and velocity at the end of the time interval can then be solved.

Higher-order formulations have also been considered by Argyris and Scharpf[6] and Howard and Penny.[14] Peters and Izadpanah[15] concluded that for small error bounds higher-order formulations could be more efficient than the ordinary time-marching schemes. However, the preceding algorithms are only conditionally stable.[14] As a result, the allowed maximum time-step size would be limited by the maximum frequency ω in the system in order to maintain numerical stability. For example, the preceding algorithm employing the cubic Hermitian polynomials in Eq. (4) would have a stability limit $\omega \Delta t < \sqrt{\frac{28}{3}}$.

To bring the variations in line with those used in the calculus of variations, frameworks based on Hamilton's law of varying action and weak formulations have been proposed to construct time-step-integration algorithms.

B. Hamilton's Law of Varying Action

Bailey[9,10] advocated the use of Hamilton's law of varying action, which appeared as early as 1830s in Hamilton's papers[16,17] dealing with a general method in dynamics. The law can be written as

$$\delta \int_{t_0}^{t_1} (T - V)\, \mathrm{d}t + \int_{t_0}^{t_1} \sum_i f_i \delta q_i \, \mathrm{d}t - \sum_i \frac{\partial T}{\partial \dot{q}_i} \delta q_i \Bigg|_{t_0}^{t_1} = 0 \tag{5}$$

Equation (5) is similar to Eq. (2) but with an extra tailing term. Equation (5) reduces to Eq. (2) if $\delta q_i(t_0) = \delta q_i(t_1) = 0$.

Bailey combined the theory of Ritz and Hamilton's law of varying action to evaluate the numerical solutions for initial value problems directly. The solution is assumed to be in the form of truncated power series, which satisfy the requirement of time-space continuity and displacement and slope at the boundaries,[18] i.e.,

$$u(t) = u_0 + v_0 t + u_2 t^2 + \cdots + u_n t^n \tag{6}$$

and the variation of $u(t)$ is proposed to be

$$\delta u(t) = \delta u_2 t^2 + \cdots + \delta u_n t^n \tag{7}$$

where $u_2, \ldots, u_n$ are the unknown coefficients to be determined and $\delta u_2, \ldots, \delta u_n$ are the arbitrary coefficients for the variation $\delta u(t)$. In this case δu is the variation of u in the sense of calculus of variations.

Riff and Baruch[19,20] argued that there should be no direct relationship between the generalized coordinates and their variations. In other words, the approximate solutions and their variations could be mutually independent. As a result, u and δu could be built from different sets of functions. In fact, Peters and Izadpanah[15] showed that regarding δu as the variation of u could affect convergence. Baruch and Riff[21] showed that Hamilton's law of varying action can be used to generate many more alternative formulations when more flexible δu can be used. They also proposed to use the second derivative of the cubic Hermitian polynomials as the admissible variations.[19,20]

When Hamilton's law of varying action is used to construct time-step-integration algorithms, the algorithms are found to be conditionally stable only. Unconditionally stable algorithms can be obtained by weakly enforcing the given initial conditions and/or the velocity-displacement relation resulting in weak formulations.

C. Weak Formulations

Hamilton's principle can be regarded as a constrained variational principle. The constraints of satisfying the initial conditions can be relaxed by introducing the initial conditions into the variational statement as natural boundary conditions through Lagrange multipliers.[22] Because the constraints of the problem are introduced as natural boundary conditions, the constraints themselves would be subject to approximation through the variation process. As a result, discontinuities of velocity and/or displacement can be observed across two time intervals. Because unconstrained trial functions can be used, the variations are therefore having more flexibility.

Expressing Hamilton's law of varying action in the weak form provides a powerful alternative to numerical solution of ordinary differential equations in the time domain. Borri et al.[23] showed that in using Hamilton's law of varying action the interpolation functions had to ensure continuity of displacement but not the velocity. The generalized momenta at the ends of the time interval could be treated as independent variables. Unconditionally stable time-step-integration algorithms were obtained by using reduced/selective element quadrature.

Peters and Izadpanah[15] analyzed the convergence of an integration procedure derived from a bilinear weak formulation, which included Hamilton's law of varying action as a special case. They concluded that the initial conditions should be enforced weakly to ensure convergence if the variations were conformed to the calculus of variations. Cannarozzi and Mancuso[24] studied various variational formulations for solid mechanics (including the principle of minimum potential energy, the Hellinger–Reissner principle and the Hu–Washizu principle) with the weak enforcement of initial displacement and velocity. They observed that the algorithms were usually conditionally stable when the initial conditions were enforced a priori. Besides, if the corresponding algorithm were unconditionally stable, it would not be as accurate as those algorithms with initial conditions weakly enforced.

In this paper we show that this is not necessarily true. Unconditionally stable algorithms with high-order accuracy can be constructed by enforcing the initial conditions in the approximate solutions as well, provided the appropriate variations are chosen. To the best knowledge of the author, this is the first time that

unconditionally stable time-step-integration algorithms are constructed directly from Hamilton's principle with the initial conditions strongly enforced.

Apart from the initial conditions, the velocity-displacement relation can be weakly enforced as well to give a mixed formulation. In this case the generalized coordinates and the generalized momenta (or generalized velocities) appear as independent variables. Unconditionally stable time-step-integration algorithms have been successfully generated.[25–28] Some of these algorithms are equivalent to the first subdiagonal and diagonal Padé approximations. However, the total number of unknowns would be double. In this paper only the displacement-based formulations are considered.

D. Outline of the Paper

In this paper time-step-integration algorithms are derived based on the primal form (i.e., displacement based) of Hamilton's principle. Single-degree-of-freedom systems are used to study the algorithm characteristics. Because the numerical stability is of interest, only unforced vibration is considered. The corresponding constrained variations are assumed to be in the form of $\Sigma\, \alpha_k \tau^k(1-\tau)$ to ensure vanishing variations at the ends of a time interval. The initial conditions are strongly enforced in the formulation. Unconditionally stable time-step-integration algorithms with order of accuracy $2n-1$ or $2n$ are constructed systematically, where n is the number of unknown variables. The ultimate spectral radius of the algorithm can be controlled directly. In fact, the approximate solutions are equivalent to the generalized Padé approximations for second-order differential equations. The required variation functions are then evaluated and expressed in terms of two stabilizing functions. Hence, Hamilton's principle can be used to construct unconditionally stable higher-order-accurate time-step-integration algorithms systemically as well, provided appropriate constrained variations are used.

II. Single-Degree-of-Freedom Systems

Because any linear N-degree-of-freedom system in general can be decomposed into N uncoupled scalar equations, it can be established that the entire coupled system reduces to the consideration of the modal equations. Thus it is a common practice to analyze a proposed algorithm using a simple scalar mass-spring system. The resultant algorithm can be extended to multi-degree-of-freedom systems systematically.[6]

The kinetic energy T and potential energy V of a mass-spring system are given by

$$T = \tfrac{1}{2} m \dot{u}^2, \qquad V = \tfrac{1}{2} k u^2 \tag{8}$$

Assume that there is also a nonconservative dissipative force proportional to the velocity in the form $-c\dot{u}$. The extended Hamilton's principle in Eq. (2) can be written as

$$\delta \int_0^{\Delta t} \left(\frac{1}{2} m \dot{u}^2 - \frac{1}{2} k u^2 \right) \mathrm{d}t - \int_0^{\Delta t} c \dot{u}\, \delta u \, \mathrm{d}t = 0 \tag{9}$$

or

$$\delta \int_0^{\Delta t} \left(\frac{1}{2} \dot{u}^2 - \frac{1}{2} \omega^2 u^2 \right) \mathrm{d}t - \int_0^{\Delta t} 2\xi\omega \dot{u}\, \delta u \, \mathrm{d}t = 0 \tag{10}$$

where $\omega = \sqrt{(k/m)}$ is the undamped natural frequency of the system and $\xi = c/[2\sqrt{(km)}]$ is the critical damping ratio. In Eqs. (9) and (10), without loss of generality, t_0 and t_1 have been assumed to be zero and Δt, respectively. If there is no other external excitation, the exact solutions for displacement $\bar{u}_1$ and velocity $\bar{v}_1$ at $t = \Delta t$ with initial displacement u_0 and initial velocity v_0 at $t = 0$ are given by[29]

$$\begin{Bmatrix} \bar{u}_1 \\ \bar{v}_1 \end{Bmatrix} = \begin{pmatrix} \bar{g}(\Delta t) & \bar{h}(\Delta t) \\ \dot{\bar{g}}(\Delta t) & \dot{\bar{h}}(\Delta t) \end{pmatrix} \begin{Bmatrix} u_0 \\ v_0 \end{Bmatrix} = [\bar{\boldsymbol{A}}] \begin{Bmatrix} u_0 \\ v_0 \end{Bmatrix} \tag{11}$$

where

$$\bar{g}(t) = \exp(-\xi\omega t)[\cos(\omega_d t) + (\xi\omega/\omega_d)\sin(\omega_d t)] \tag{12a}$$

$$\bar{h}(t) = \exp(-\xi\omega t)[(1/\omega_d)\sin(\omega_d t)], \qquad \omega_d = \sqrt{1-\xi^2}\,\omega \tag{12b}$$

All of the numerical methods are trying to approximate this amplification matrix. In these methods the approximate displacement u_1 and velocity v_1 at $t = \Delta t$ are related to u_0 and v_0 by

$$\begin{Bmatrix} u_1 \\ v_1 \end{Bmatrix} = [\boldsymbol{A}] \begin{Bmatrix} u_0 \\ v_0 \end{Bmatrix} \tag{13}$$

Different algorithms would have different numerical amplification matrices. The stability property depends on the magnitude of the eigenvalues of $[\boldsymbol{A}]$. The order of accuracy can be obtained by finding the truncation errors between $[\boldsymbol{A}]$ and $[\bar{\boldsymbol{A}}]$.

III. Approximate Solutions and Admissible Variations

To use Hamilton's principle correctly, the variations at the beginning and at the end of a time interval must vanish. Assume $u(t)$ is approximated by

$$u(t) = u(\tau\Delta t) = u_0(1-\tau) + u_1\tau + u_2\tau(1-\tau) + u_3\tau^2(1-\tau) + \cdots + u_{n+1}\tau^n(1-\tau) \tag{14}$$

Alternatively, $u(t)$ can be expressed in terms of some orthogonal functions, for example Legendre polynomials[30] or Jacobi polynomials.[26] Making use of the orthogonal properties of these functions, the resultant coefficient matrix can be very sparse and not ill-conditioned even when n becomes large. In the present formulation the variations are chosen so that the resultant coefficient matrix can be sparse.

If u_0 and u_1 in Eq. (14) are assumed to be known and if the calculus of variations is followed strictly, then usually δu is assumed to be made up of the n linearly independent functions $\tau(1-\tau), \tau^2(1-\tau), \ldots, \tau^n(1-\tau)$ only. Because there are $n+2$ coefficients in Eq. (14), they can be determined from the two initial conditions and the n equations obtained from the n linearly independent variation functions of δu. However, the resultant algorithms are found to be conditionally stable only.

Of course, other forms of variations are possible as long as the variations vanish at the two endpoints. In the following, the variations are assumed to be in the form of $\Sigma\, \alpha_k \tau^k(1-\tau)$ so that the variations would vanish at the two endpoints. However, the range of the index k may exceed n. Let such variations of δu be made up of n linearly independent functions $\psi_1(\tau), \ldots, \psi_n(\tau)$. The explicit form of $\psi_i(\tau)$ will be determined later. The equation corresponding to the variation related to $\psi_i(\tau)$ is given by

$$\int_0^1 \left(\frac{1}{\Delta t^2} \frac{\mathrm{d}\psi_i(\tau)}{\mathrm{d}\tau} \frac{\mathrm{d}u(\tau)}{\mathrm{d}\tau} - \omega^2 \psi_i(\tau) u(\tau) - 2\xi\omega\psi_i(\tau) \frac{1}{\Delta t} \frac{\mathrm{d}u(\tau)}{\mathrm{d}\tau} \right) \mathrm{d}\tau = 0 \tag{15}$$

or using integrating by parts

$$\int_0^1 \psi_i(\tau) \left(\frac{1}{\Delta t^2} \frac{\mathrm{d}^2 u(\tau)}{\mathrm{d}\tau^2} + 2\xi\omega \frac{1}{\Delta t} \frac{\mathrm{d}u(\tau)}{\mathrm{d}\tau} + \omega^2 u(\tau) \right) \mathrm{d}\tau = 0 \tag{16}$$

because $\psi_i(0) = \psi_i(1) = 0$. Equation (16) can also be obtained from the weighted residual method by using $\psi_i(\tau)$ as the corresponding weighting function.

In Eq. (14), $u_1, u_2, \ldots, u_{n+1}$ are not mutually independent as $\dot{u}(0) = v_0$ is the given initial velocity:

$$u_1 = u_0 + v_0\Delta t - u_2 \tag{17}$$

Hence, Eq. (14) can be written as

$$u(\tau) = u_0(1-\tau) + (u_0 + v_0\Delta t - u_2)\tau + u_2\tau(1-\tau) + u_3\tau^2(1-\tau) + \cdots + u_{n+1}\tau^n(1-\tau) \tag{18}$$

or in a more concise form

$$u(\tau) = u_0 + v_0\Delta t\tau + U_2\tau^2 + \cdots + U_{n+1}\tau^{n+1} \tag{19}$$

where $U_k = u_{k+1} - u_k$ for $k = 2, \ldots, n$ and $U_{n+1} = -u_{n+1}$.

Using Eq. (16) with i ranging from 1 to n, the n unknowns $U_1, \ldots, U_n$ can be obtained by solving

$$\left(\begin{bmatrix} 1\times 2\times W_{10} & 2\times 3\times W_{11} & \cdots & n(n+1)W_{1,n-1} \\ 1\times 2\times W_{20} & 2\times 3\times W_{21} & \cdots & n(n+1)W_{2,n-1} \\ \vdots & \vdots & \ddots & \vdots \\ 1\times 2\times W_{n0} & 2\times 3\times W_{n1} & \cdots & n(n+1)W_{n,n-1} \end{bmatrix} + 2\xi\omega\Delta t \begin{bmatrix} 2W_{11} & 3W_{12} & \cdots & (n+1)W_{1,n} \\ 2W_{21} & 3W_{22} & \cdots & (n+1)W_{2,n} \\ \vdots & \vdots & \ddots & \vdots \\ 2W_{n1} & 3W_{n2} & \cdots & (n+1)W_{n,n} \end{bmatrix} + \omega^2\Delta t^2 \begin{bmatrix} W_{12} & W_{13} & \cdots & W_{1,n+1} \\ W_{22} & W_{23} & \cdots & W_{2,n+1} \\ \vdots & \vdots & \ddots & \vdots \\ W_{n2} & W_{n3} & \cdots & W_{n,n+1} \end{bmatrix} \right) \begin{Bmatrix} U_1 \\ U_2 \\ \vdots \\ U_n \end{Bmatrix} = -\begin{pmatrix} \omega^2\Delta t^2 u_0 + \\ 2\xi\omega\Delta t^2 v_0 \end{pmatrix} \begin{Bmatrix} W_{10} \\ W_{20} \\ \vdots \\ W_{n0} \end{Bmatrix} - \omega^2\Delta t^3 v_0 \begin{Bmatrix} W_{11} \\ W_{21} \\ \vdots \\ W_{n1} \end{Bmatrix} \tag{20}$$

where W_{ij} is defined as

$$W_{ij} = \int_0^1 \psi_i(\tau)\tau^j \, d\tau \tag{21}$$

The velocity, which is the time derivative of displacement, can be obtained by differentiating Eq. (19) with respect to t and can be written as

$$v(t) = v(\tau\Delta t) = v_0 + \frac{\left[2U_1\tau + \cdots + (n+1)U_n\tau^n\right]}{\Delta t} \tag{22}$$

The approximate displacement and velocity at the end of the time interval ($t = \Delta t$ or $\tau = 1$) are therefore given by

$$u_1 = u(\Delta t) = u_0 + v_0\Delta t + U_1 + \cdots + U_n \tag{23a}$$

$$v_1 = v(\Delta t) = v_0 + \frac{[2U_1 + \cdots + (n+1)U_n]}{\Delta t} \tag{23b}$$

Because $\psi_1(\tau), \ldots, \tau_n(\tau)$ are linearly independent, they can be recombined to form another set of n linearly independent functions. The resultant Eq. (20) may be different, but the numerical results obtained would remain the same. Hence, without loss of generality the algorithmic parameters W_{ij} can be normalized so that

$$W_{i,i-1} = 1 \qquad \text{for} \qquad 1 \le i \le n \tag{24a}$$

$$W_{i,n} = a_i, \qquad W_{i,n+1} = b_i, \qquad \text{for} \qquad 1 \le i \le n \tag{24b}$$

$$W_{ij} = 0 \qquad \text{otherwise} \tag{24c}$$

Equation (20) is then simplified to

$$\left(\begin{bmatrix} 1\times 2 & & & & \\ & 2\times 3 & & & \\ & & 3\times 4 & & \\ & & & \ddots & \\ & & & & n(n+1) \end{bmatrix} + 2\xi\omega\Delta t \begin{bmatrix} 0 & & & & (n+1)a_1 \\ 2 & 0 & & & (n+1)a_2 \\ & 3 & \ddots & & (n+1)a_3 \\ & & \ddots & 0 & \vdots \\ & & & n & (n+1)a_n \end{bmatrix} + \omega^2\Delta t^2 \begin{bmatrix} 0 & & & a_1 & b_1 \\ 0 & \ddots & & a_2 & b_2 \\ 1 & \ddots & 0 & a_3 & b_3 \\ & \ddots & 0 & \vdots & \vdots \\ & & 1 & a_n & b_n \end{bmatrix} \right) \begin{Bmatrix} U_1 \\ U_2 \\ U_3 \\ \vdots \\ U_n \end{Bmatrix} = -\begin{pmatrix} \omega^2\Delta t^2 u_0 + \\ 2\xi\omega\Delta t^2 v_0 \end{pmatrix} \begin{Bmatrix} 1 \\ 0 \\ 0 \\ \vdots \\ 0 \end{Bmatrix} - \omega^2\Delta t^3 v_0 \begin{Bmatrix} 0 \\ 1 \\ 0 \\ \vdots \\ 0 \end{Bmatrix} \tag{25}$$

The n unknowns $U_1, U_2, \ldots, U_n$ can be evaluated by using Cramer's rule. Reference 31 shows that for arbitrary values of a_k and b_k the orders of accuracy for u_1 and v_1 in Eq. (23) are at least $n+1$ and n, respectively. The formulation is therefore consistent. Besides, u_1 and v_1 in Eq. (23) can be expressed as

$$\begin{Bmatrix} u_1 \\ v_1 \end{Bmatrix} = \begin{pmatrix} g(\Delta t) & h(\Delta t) \\ g_d(\Delta t) & h_d(\Delta t) \end{pmatrix} \begin{Bmatrix} u_0 \\ v_0 \end{Bmatrix} = [\boldsymbol{A}] \begin{Bmatrix} u_0 \\ v_0 \end{Bmatrix} \tag{26}$$

where $g(\Delta t)$, $h(\Delta t)$, $g_d(\Delta t)$, $h_d(\Delta t)$ are approximations to $\bar{g}(\Delta t)$, $\bar{h}(\Delta t)$, $\dot{\bar{g}}(\Delta t)$, and $\dot{\bar{h}}(\Delta t)$ in Eq. (11), respectively. Obviously, the algorithm characteristics (for example, stability and accuracy, etc.) depend on the algorithmic parameters a_k and b_k.

In Ref. 31, by considering the equivalent first-order formulation, if a_k and b_k are chosen to be

$$a_k = \frac{(-1)^{n-k} n!n!(n+k-2)!}{(k-1)!(k-1)!(n+1-k)!2n!} \frac{2[n+(k-1)\mu]}{(1+\mu)} \tag{27}$$

$$b_k = \frac{(-1)^{n-k}(n+1)!(n-1)!(n+k-3)!}{(k-1)!(k-1)!(n-k)!(2n-1)!} \times \frac{\left\{ \begin{matrix} (k-1)[(n-1)(k+n-1)+1]\mu^2 + \\ [(n+k-2)\mu + n](kn + n^2 - n - 1) \end{matrix} \right\}}{(2n-1)(n-k+2)(1+\mu)^2} \tag{28}$$

then the resultant algorithms will be unconditionally C-stable and $(2n-1)$th-order accurate if $-1 < \mu < 1$ and $2n$th-order accurate if $\mu = 1$. The detailed derivation can be found in Ref. 31. In fact, the resultant algorithms are equivalent to the generalized Padé approximations for second-order equations.

The generalized Padé approximations to exponential function $\exp(z)$ is defined as

$$A_{2n-1}(z) = \frac{(1-\mu)P_{n-1,n}(z) + 2\mu P_{n,n}(z)}{(1-\mu)Q_{n-1,n}(z) + 2\mu Q_{n,n}(z)} \tag{29}$$

where

$$P_{k,j}(z) = \sum_{h=0}^{k} \frac{k!(j+k-h)!}{(k-h)!(j+k)!}\frac{z^h}{h!}$$

$$Q_{k,j}(z) = \sum_{h=0}^{j} (-1)^h \frac{j!(j+k-h)!}{(j-h)!(j+k)!}\frac{z^h}{h!} \tag{30}$$

The magnitude of $A_{2n-1}(z)$ as $|z| \to \infty$ is given by

$$\lim_{|z|\to\infty} |A_{2n-1}(z)| = \left| \frac{2\mu(n!/2n!)}{(1-\mu)[(-1)^n(n-1)!/(2n-1)!] + 2\mu[(-1)^n n!/2n!]} \right| = |\mu| \tag{31}$$

Hence, the ultimate amplification factor is a directly controllable algorithmic parameter. Furthermore, $A_{2n-1}(z)$ is $(2n-1)$th-order accurate if $\mu \neq 1$ and $2n$th-order accurate if $\mu = 1$. Besides, $A_{2n-1}(z)$ is A-stable if $-1 \leq \mu \leq 1$ and L-stable if $\mu = 0$. Furthermore, when $\mu = 0$, $A_{2n-1}(z)$ is equivalent to the first subdiagonal $(n-1, n)$ Padé approximation, and when $\mu = 1$, $A_{2n-1}(z)$ is equivalent to the diagonal (n, n) Padé approximation. A more detailed discussion on the generalized Padé approximations can be found in Ref. 32. Algorithms giving results equivalent to the generalized Padé approximations can be constructed by using the weighted residual method,[31,32] the complex time-step method,[33,34] and the least-squares method.[35]

As a result, the algorithmic parameter μ in Eqs. (27) and (28) can be used to control the ultimate spectral radius of the resultant algorithm. When $\mu = 0$, the algorithms are asymptotically annihilating. When $\mu = 1$, the algorithms are nondissipative.

Once the algorithmic parameters a_k and b_k are known, $\psi_1(\tau), \ldots, \psi_n(\tau)$ (and hence the corresponding variations) can be worked out systematically. Because the n variations $\psi_1(\tau), \ldots, \psi_n(\tau)$ are to be constructed from $\bar{\psi}_k(\tau) = \tau^k(1-\tau)$, $\psi_i(\tau)$ can be written as

$$\psi_i(\tau) = \Sigma\, \alpha_{ik}\bar{\psi}_k(\tau), \qquad i = 1, \ldots, n \tag{32}$$

where the number of $\bar{\psi}_k(\tau)$ can be more than n. As a result,

$$W_{ij} = \int_0^1 \psi_i(\tau)\tau^j\, d\tau = \Sigma \int_0^1 \alpha_{ik}\bar{\psi}_k(\tau)\tau^j\, d\tau = \Sigma\, \alpha_{ik}\bar{W}_{kj} \tag{33}$$

In matrix notation

$$[\boldsymbol{W}] = [\alpha][\bar{\boldsymbol{W}}] \tag{34}$$

where

$$[\boldsymbol{W}] = \begin{bmatrix} 1 & & & a_1 & b_1 \\ & 1 & & a_2 & b_2 \\ & & \ddots & \vdots & \vdots \\ & & & 1 & a_n & b_n \end{bmatrix} \tag{35a}$$

is an $n \times n+2$ matrix, and

$$[\bar{\boldsymbol{W}}] = \begin{bmatrix} \bar{W}_{10} & \bar{W}_{11} & \cdots & \bar{W}_{1,n+1} \\ \bar{W}_{20} & \bar{W}_{21} & \cdots & \bar{W}_{2,n+1} \\ \vdots & \vdots & \cdots & \vdots \end{bmatrix} \tag{35b}$$

is a matrix with $n+2$ columns. $[\alpha]$ is the unknown coefficient matrix to be determined. $[\alpha]$ can be determined if $[\bar{\boldsymbol{W}}]$ is a nonsingular $n+2 \times n+2$ square matrix. In other words, there should be $n+2$ linearly independent functions $\bar{\psi}_k(\tau)$. $[\alpha]$ is then given by

$$[\alpha] = [\boldsymbol{W}][\bar{\boldsymbol{W}}]^{-1} \tag{36}$$

If $\{\psi\} = [\psi_1, \psi_2, \ldots, \psi_n]^T$ and $\{\bar{\psi}\} = [\bar{\psi}_1, \bar{\psi}_2, \ldots, \bar{\psi}_{n+2}]^T$, then

$$\{\psi\} = [\alpha]\{\bar{\psi}\} = [\boldsymbol{W}][\bar{\boldsymbol{W}}]^{-1}\{\bar{\psi}\} \tag{37}$$

and $\psi_i(\tau)$ can be worked out systemically.

In Eq. (37) $\psi_1(\tau), \ldots, \psi_n(\tau)$ would give the normalized algorithmic parameters W_{ij} in Eq. (24). These n linearly independent functions can be recombined to form another set of n linearly independent functions. Equations (24) and (20) can be different, but the numerical results would be the same. As a result, the same time-step-integration algorithm is reproduced. The new set of functions can be obtained by premultiplying $[\boldsymbol{W}][\bar{\boldsymbol{W}}]^{-1}\{\bar{\psi}\}$ by any nonsingular square matrix $[\boldsymbol{H}]$. As a result, it is more general to consider

$$\{\psi\} = [\boldsymbol{H}][\boldsymbol{W}][\bar{\boldsymbol{W}}]^{-1}\{\bar{\psi}\} \tag{38}$$

instead of Eq. (37).

IV. Stabilizing Functions

Let $[\boldsymbol{H}_{j,k}]$ be a matrix with j rows and k columns in the form

$$[\boldsymbol{H}_{j,k}] = \begin{bmatrix} 1/(2\times 3) & \cdots & 1/[(k+1)(k+2)] \\ \vdots & & \vdots \\ 1/[(j+1)(j+2)] & \cdots & 1/[(j+k)(j+k+1)] \end{bmatrix} \tag{39}$$

Premultiplying $[\boldsymbol{H}_{n,n}]$ to $[\boldsymbol{W}]$ could simplify $[\boldsymbol{W}]$ to

$$[\boldsymbol{H}_{n,n}][\boldsymbol{W}] = [\boldsymbol{H}_{n,n+2}] + [0\,\boldsymbol{Z}\boldsymbol{V}] \tag{40}$$

where $\{\boldsymbol{Z}\}$ and $\{\boldsymbol{V}\}$ are column vectors with n elements. $Z_1 = \cdots = Z_{n-3} = V_1 = \cdots = V_{n-4} = 0$. The values of Z_k and V_k for various n are shown in Tables 1 and 2, respectively.

If there are $n+2$ rows in Eq. (35b), $[\bar{\boldsymbol{W}}] = [\boldsymbol{H}_{n+2,n+2}]$. As a result, $\{\psi\}$ in Eq. (38) can be written as

$$\{\psi\} = [\boldsymbol{H}_{n,n}][\boldsymbol{W}][\boldsymbol{H}_{n+2,n+2}]^{-1}\{\bar{\psi}\} = \left([\boldsymbol{I}_{n,n+2}] + [0\,\boldsymbol{Z}\boldsymbol{V}][\boldsymbol{H}_{n+2,n+2}]^{-1}\right)\{\bar{\psi}\} \tag{41}$$

where

$$[\boldsymbol{I}_{n,n+2}] = \begin{bmatrix} 1 & & & 0 & 0 \\ & \ddots & & \vdots & \vdots \\ & & 1 & 0 & 0 \end{bmatrix} \tag{42}$$

is an $n \times n+2$ matrix. Equation (41) shows that only the last two rows of $[\boldsymbol{H}_{n+2,n+2}]^{-1}$ are required in the calculation. The last two rows of $[\boldsymbol{H}_{n+2,n+2}]^{-1}$ can be found explicitly as

$$\begin{pmatrix} J_{11} & J_{12} & \cdots & J_{1,n+1} & J_{1,n+2} \\ J_{21} & J_{22} & \cdots & J_{2,n+1} & J_{2,n+2} \end{pmatrix} \tag{43}$$

where

$$J_{1i} = (-1)^{n+i}\frac{(n+1+i)!(2n+4)![(n+2+i)(2n+5) - (n+3+i)(n+3)(n+2)]}{(n+3)!(n+2)!(n+2-i)!i!(i-1)!} \tag{44a}$$

$$J_{2i} = (-1)^{n+i}\frac{(n+2+i)!(2n+5)!}{(n+3)!(n+2)!(n+2-i)!i!(i-1)!} \tag{44b}$$

The last two rows of $[\boldsymbol{H}_{n+2,n+2}]^{-1}$ for various n are shown explicitly in Table 3.

Table 1 Z_j values for various n

n	Z_j values
1	$-\frac{1}{12}\frac{-1+\mu}{1+\mu}$
2	$\frac{1}{180}\quad\frac{1}{180}\frac{1}{1+\mu}$
3	$\frac{1}{600}\frac{-1+\mu}{1+\mu}\quad\frac{1}{4{,}200}\frac{-2+5\mu}{1+\mu}\quad\frac{1}{1{,}400}\frac{\mu}{1+\mu}$
4	$0\quad\frac{1}{9{,}800}\frac{-1+\mu}{1+\mu}\quad\frac{1}{88{,}200}\frac{-7+11\mu}{1+\mu}\quad\frac{1}{44{,}100}\frac{-2+5\mu}{1+\mu}$
5	$0\quad 0\quad\frac{1}{158{,}760}\frac{-1+\mu}{1+\mu}\quad\frac{1}{1{,}746{,}360}\frac{-14+19\mu}{1+\mu}\quad\frac{1}{698{,}544}\frac{-5+9\mu}{1+\mu}$
6	$0\quad 0\quad 0\quad\frac{1}{2{,}561{,}328}\frac{-1+\mu}{1+\mu}\quad\frac{1}{33{,}297{,}264}\frac{-23+29\mu}{1+\mu}\quad\frac{1}{11{,}099{,}088}\frac{-9+14\mu}{1+\mu}$
7	$0\quad 0\quad 0\quad 0\quad\frac{1}{41{,}225{,}184}\frac{-1+\mu}{1+\mu}\quad\frac{1}{618{,}377{,}760}\frac{-34+41\mu}{1+\mu}\quad\frac{1}{88{,}339{,}680}\frac{-7+10\mu}{1+\mu}$
8	$0\quad 0\quad 0\quad 0\quad 0\quad\frac{1}{662{,}547{,}600}\frac{-1+\mu}{1+\mu}\quad\frac{1}{11{,}263{,}309{,}200}\frac{-47+55\mu}{1+\mu}\quad\frac{1}{2{,}815{,}827{,}300}\frac{-20+27\mu}{1+\mu}$
9	$0\quad 0\quad 0\quad 0\quad 0\quad 0\quad\frac{1}{10{,}637{,}569{,}800}\frac{-1+\mu}{1+\mu}\quad\frac{1}{202{,}113{,}826{,}200}\frac{-62+71\mu}{1+\mu}\quad\frac{1}{44{,}914{,}183{,}600}\frac{-27+35\mu}{1+\mu}$
10	$0\quad 0\quad 0\quad 0\quad 0\quad 0\quad 0\quad\frac{1}{170{,}673{,}897{,}680}\frac{-1+\mu}{1+\mu}\quad\frac{1}{3{,}584{,}151{,}851{,}580}\frac{-79+89\mu}{1+\mu}\quad\frac{1}{716{,}830{,}370{,}256}\frac{-35+44\mu}{1+\mu}$

Table 2 V_j values for various n

n	V_j values
1	$-\frac{1}{60}\frac{-17+6\mu+3\mu^2}{(1+\mu)^2}$
2	$\frac{1}{180}\frac{4\mu-1}{1+\mu}\quad\frac{1}{1260}\frac{5+10\mu+12\mu^2}{(1+\mu)^2}$
3	$\frac{1}{10{,}500}\frac{-47+4\mu+23\mu^2}{(1+\mu)^2}\quad\frac{1}{21{,}000}\frac{-39+34\mu+45\mu^2}{(1+\mu)^2}\quad\frac{1}{63{,}000}\frac{-35+98\mu+97\mu^2}{(1+\mu)^2}$
4	$-\frac{1}{5{,}880}\frac{\mu-1}{1+\mu}\quad\frac{1}{123{,}480}\frac{-29+2\mu+13\mu^2}{(1+\mu)^2}\quad\frac{1}{61{,}740}\frac{-14+8\mu+13\mu^2}{(1+\mu)^2}\quad\frac{1}{679{,}140}\frac{-102+126\mu+15\mu^2}{(1+\mu)^2}$
5	$0\quad-\frac{1}{105{,}840}\frac{\mu-1}{1+\mu}\quad\frac{1}{10{,}478{,}160}\frac{-139+8\mu+59\mu^2}{(1+\mu)^2}\quad\frac{1}{10{,}478{,}160}\frac{-235+102\mu+205\mu^2}{(1+\mu)^2}\quad\frac{1}{27{,}243{,}216}\frac{-605+506\mu+747\mu^2}{(1+\mu)^2}$
6	$0\quad 0\quad-\frac{1}{1{,}829{,}520}\frac{\mu-1}{1+\mu}\quad\frac{1}{261{,}621{,}360}\frac{-203+10\mu+83\mu^2}{(1+\mu)^2}\quad\frac{1}{52{,}324{,}272}\frac{-105+37\mu+90\mu^2}{(1+\mu)^2}\quad\frac{1}{261{,}621{,}360}\frac{-689+442\mu+786\mu^2}{(1+\mu)^2}$
7	$0\quad 0\quad 0\quad-\frac{1}{30{,}918{,}888}\frac{\mu-1}{1+\mu}\quad\frac{1}{2{,}009{,}727{,}720}\frac{-93+4\mu+37\mu^2}{(1+\mu)^2}\quad\frac{1}{4{,}019{,}455{,}440}\frac{-679+202\mu+581\mu^2}{(1+\mu)^2}$ $\frac{1}{9{,}761{,}534{,}640}\frac{-2{,}695+1{,}410\mu+2{,}949\mu^2}{(1+\mu)^2}$
8	$0\quad 0\quad 0\quad 0\quad-\frac{1}{515{,}314{,}800}\frac{\mu-1}{1+\mu}\quad\frac{1}{131{,}405{,}274{,}000}\frac{143\mu^2+14\mu-367}{(1+\mu)^2}\quad\frac{1}{9{,}386{,}091{,}000}\frac{-128+33\mu+110\mu^2}{(1+\mu)^2}$ $\frac{1}{89{,}167{,}864{,}500}\frac{-2{,}380+1{,}054\mu+2{,}541\mu^2}{(1+\mu)^2}$
9	$0\quad 0\quad 0\quad 0\quad 0\quad-\frac{1}{8{,}510{,}055{,}840}\frac{\mu-1}{1+\mu}\quad\frac{1}{2{,}748{,}748{,}036{,}320}\frac{179\mu^2+16\mu-467}{(1+\mu)^2}\quad\frac{1}{1{,}374{,}374{,}018{,}160}\frac{1{,}269\mu^2+334\mu-1{,}467}{(1+\mu)^2}$ $\frac{1}{3{,}206{,}872{,}709{,}040}\frac{8{,}207\mu^2+3{,}002\mu-7{,}809}{(1+\mu)^2}$
10	$0\quad 0\quad 0\quad 0\quad 0\quad 0\quad-\frac{1}{139{,}642{,}279{,}920}\frac{\mu-1}{1+\mu}\quad\frac{1}{18{,}572{,}423{,}229{,}360}\frac{73\mu^2+6\mu-193}{(1+\mu)^2}\quad\frac{1}{6{,}190{,}807{,}743{,}120}\frac{440\mu^2+103\mu-505}{(1+\mu)^2}$ $\frac{1}{28{,}477{,}715{,}618{,}352}\frac{6{,}296\mu^2+2{,}058\mu-6{,}055}{(1+\mu)^2}$

Table 3 Last two rows of $[\boldsymbol{H}_{n+2,n+2}]^{-1}$

n	Last two rows of $[\boldsymbol{H}_{n+2,n+2}]^{-1}$
1	$60\begin{bmatrix} -8 & 37 & -35 \\ 7 & -35 & 35 \end{bmatrix}$
2	$840\begin{bmatrix} 5 & -43 & 97 & -63 \\ 3 & 27 & -63 & 42 \end{bmatrix}$
3	$1{,}260\begin{bmatrix} -24 & 326 & -1{,}274 & 1{,}876 & -924 \\ 11 & -154 & 616 & -924 & 462 \end{bmatrix}$
4	$5{,}544\begin{bmatrix} 35 & -685 & 4{,}040 & -9{,}960 & 10{,}830 & -4{,}290 \\ -13 & 260 & -1{,}560 & 3{,}900 & -4{,}290 & 1{,}716 \end{bmatrix}$
5	$72{,}072\begin{bmatrix} -16 & 425 & -3{,}495 & 12{,}675 & -22{,}605 & 19{,}437 & -6{,}435 \\ 5 & -135 & 1{,}125 & -4{,}125 & 7{,}425 & -6{,}435 & 2{,}145 \end{bmatrix}$
6	$102{,}960\begin{bmatrix} 63 & -2{,}177 & 23{,}695 & -117{,}425 & 302{,}995 & -421{,}421 & 299{,}299 & -85{,}085 \\ -17 & 595 & -6{,}545 & 32{,}725 & -85{,}085 & 119{,}119 & -85{,}085 & 24{,}310 \end{bmatrix}$
7	437,580 [−80, 3,484, −48,356, 312,004, −1,085,084, 2,158,156, −2,454,452, 1,483,768, −369,512] [19, −836, 11,704, −76,076, 266,266, −532,532, 608,608, −369,512, 92,378]
8	5,542,680 [33, −1,767, 30,408, −246,792, 1,104,558, −2,931,474, 4,726,176, −4,540,224, 2,388,126, −529,074] [−7, 378, −6,552, 53,508, −240,786, 642,096, −1,039,584, 1,002,456, −529,074, 117,572]
9	7,759,752 [−120, 7,745, −161,655, 1,607,970, −8,962,590, 30,346,680, −64,788,360, 87,635,340, −72,810,660, 33,885,930, −6,760,390] [23, −1,495, 31,395, −313,950, 1,758,120, −5,977,608, 12,809,160, −17,383,860, 14,486,550, −6,760,390, 1,352,078]
10	32,449,872 [143, −10,945, 272,195, −3,251,325, 22,018,920, −92,143,128, 249,285,960, −443,840,760, 516,428,550, −377,790,490, 157,632,398 −28,601,650] [−25, 1,925, −48,125, 577,500, −3,927,000, 16,493,400, −44,767,800, 79,942,500, −93,266,250, 68,395,250, −2,860,150, 5,200,300]

As a result, $[0\,\boldsymbol{ZV}][\boldsymbol{H}_{n+2,n+2}]^{-1}\{\bar{\psi}\}$ can be evaluated explicitly without matrix inversion. Let $\phi_1(\tau)$ and $\phi_2(\tau)$ be the two functions obtained by multiplying the second last row and the last row of $[\boldsymbol{H}_{n+2,n+2}]^{-1}$ with $\{\bar{\psi}\}$, i.e.,

$$\phi_1(\tau) = \sum_{k=1}^{n+2} J_{1k}\tau^k(1-\tau) \tag{45a}$$

$$\phi_2(\tau) = \sum_{k=1}^{n+2} J_{2k}\tau^k(1-\tau) \tag{45b}$$

Hence, the linearly independent functions $\psi_1(\tau), \ldots, \psi_n(\tau)$ in Eq. (41) are given explicitly as

$$\psi_j(\tau) = \tau^j(1-\tau) + Z_j\phi_1(\tau) + V_j\phi_2(\tau), \qquad 1 \le j \le n \tag{46}$$

Because $Z_1 = \cdots = Z_{n-3} = V_1 = \cdots = V_{n-4} = 0$,

$$\psi_j(\tau) = \tau^j(1-\tau), \qquad 1 \le j \le n-4 \tag{47}$$

Note that $\psi_{n-3}(\tau)$, $\psi_{n-2}(\tau)$, $\psi_{n-1}(\tau)$, and $\psi_n(\tau)$ are not simply $\tau^{n-3}(1-\tau)$, $\tau^{n-2}(1-\tau)$, $\tau^{n-1}(1-\tau)$, and $\tau^n(1-\tau)$ but are modified by $\phi_1(\tau)$ and $\phi_2(\tau)$ with appropriate coefficients Z_j and V_j. The resultant algorithms are unconditionally stable and at least $(2n-1)$th-order accurate. Hence $\phi_1(\tau)$ and $\phi_2(\tau)$ can be regarded as stabilizing functions.

If the stabilizing functions $\phi_1(\tau)$ and $\phi_2(\tau)$ are ignored, i.e., $\psi_k(\tau) = \tau^k(1-\tau)$ for k from 1 to n, the resultant algorithms are only conditionally stable and nth-order accurate. The inclusion of stabilizing functions $\phi_1(\tau)$ and $\phi_2(\tau)$ therefore improve both the stability and accuracy simultaneously. The variations to generate the normalized algorithmic parameters W_{ij} in Eq. (24) can be obtained by premultiplying $[\boldsymbol{H}_{n,n}]^{-1}$ and $\{\psi\}$ in Eq. (41) if required.

V. Algorithm Characteristics

A. Spectral Radius

The spectral radius is defined as the largest magnitude of the eigenvalues of the numerical amplification matrix $[\boldsymbol{A}]$ in Eq. (26). If $\xi = 0$, the eigenvalues of the numerical amplification matrix are complex conjugates, and the spectral radius (SR) for various n is given by

$$\mathrm{SR}^2 = 1 - \left[(1-\mu^2)\rho^{2n} \Big/ \rho^{2n} + \sum_{k=1}^{n} \frac{1}{2}\frac{(2k)!(n+k-1)!}{k!k!(n-k)!} \times (n\mu^2 + 2k\mu + n)\rho^{2(n-k)}\right] \tag{48}$$

where $\rho = \omega\Delta t$. As ρ approaches infinity, the magnitude of the spectral radius approaches $|\mu|$, as expected. Hence, for a desirable ultimate spectral radius two values of μ are possible. However, the magnitude of the leading truncation error terms in Eqs. (50a) and (50b) increase as μ decreases from 1. As a result, for a desirable ultimate spectral radius $|\lambda_\infty|$, $\mu = |\lambda_\infty|$ would give better results than $\mu = -|\lambda_\infty|$. In conclusion, only algorithms with $0 \le \mu \le 1$ would be of interest.

The denominator in Eq. (48) is greater than zero for all ρ, n, and μ. As a result, the spectral radius is less than unity for $-1 < \mu \le 1$. In other words, the algorithms are unconditionally stable if $-1 < \mu \le 1$ for all n, as expected. Furthermore, as all of the coefficients in the denominator polynomial are greater than zero, the spectral radius decreases monotonically from unity toward the ultimate spectral radius as ρ increases. This shows a good numerical dissipation property as the high-frequency responses are damped progressively.

The present algorithms are unconditionally stable and higher-order accurate. By using n unknown variables, the order of accuracy achievable is $2n-1$ or $2n$. Table 4 shows the orders of accuracy and stability conditions for various algorithms based on displacement-based formulations of Hamilton's principle and Hamilton's law of varying action. Many algorithms are conditionally stable only. Unconditionally stable algorithms are either asymptotically annihilating (L_1, L_2 in Ref. 24) or nondissipative at the high-frequency regime (displacement-based discontinuous Galerkin–Hamilton's law: P1-P0, P2-P1, and P3-P2 in Ref. 28).

Figure 1 shows the spectral radii of various unconditionally stable algorithms. The following notations are used (the present algorithms are more favorable as the numerical dissipation increases progressively with $\Delta t/T$): Houbolt—Houbolt method; Park—Park method; HHT—Hilber–Hughes–Taylor method ($\alpha = -0.3$); NM—trapezoidal rule (Newmark method with $\beta = \frac{1}{4}$, $\gamma = \frac{1}{2}$); P12—(1, 2) first subdiagonal Padé approximation; P22—(2, 2) diagonal

Table 4 Accuracy and stability conditions for various algorithms

Formulation	Number of unknowns	Order of accuracy	Stability conditions
Variations[19]: 1, τ	2	4	$\omega\Delta t < \sqrt{10}$
Variations: τ, τ^2	2	2	$\omega\Delta t < \sqrt{60}$
Variations: τ^2, τ^3, Bailey[9,10], $\bar{L}_3$ (Ref. 24)	2	2	$\omega\Delta t < \sqrt{84}$
Variations: N_3, N_4 (F1 formulation[21])	2	2	$\omega\Delta t < \sqrt{84}$
Variations: N_2, N_4 (F2 formulation[21]) Hamilton's Principle[6,7,8,14]	2	2	$\omega\Delta t < \sqrt{28/3}$
Variations: N_1, N_3 (F3 formulation[21])	2	4	$\omega\Delta t < \sqrt{168/17}$
Variations: N_1, N_2 (F4 formulation[21])	2	2	Unstable
Variations: N_2, N_3 (F5 formulation[21])	2	2	$\omega\Delta t < \sqrt{(498 - 6\sqrt{29})/7}$
Variations: N_1, N_4 (F6 formulation[21])	2	2	Unstable
Displacement-based bidiscontinuous Galerkin–Hamilton's law[28]			
P1-P1	2	2	$\omega\Delta t < \sqrt{12}$
P2-P2	3	4	$\omega\Delta t < \sqrt{10}$
P3-P3	4	6	$\omega\Delta t < \sqrt{42}$
Displacement-based discontinuous Galerkin–Hamilton's law[28]			
P1-P0	1	1	Unconditionally stable
P2-P1	2	3	Unconditionally stable
P3-P2	3	5	Unconditionally stable
Hamilton's law with initial conditions strongly enforced[9,10,24]			
$\bar{L}_4$	3	3	Unstable
$\bar{L}_5$	4	4	Unstable
$\bar{L}_6$	5	5	$\omega\Delta t < \sqrt{(7{,}722 - 78\sqrt{6{,}083})/13}$
$\bar{L}_7$	6	6	$\omega\Delta t < \sqrt{(406{,}824 - 402\sqrt{343{,}893})/67}$
Hamilton's law with truncated power series[9,10]	n	n	Conditionally stable
Hamilton's law with initial conditions weakly enforced[24]			
L_1	2	2	Unconditionally stable asymptotically annihilating
L_2	3	4	Unconditionally stable asymptotically annihilating
Present formulation	n	$2n - 1$	Unconditionally stable

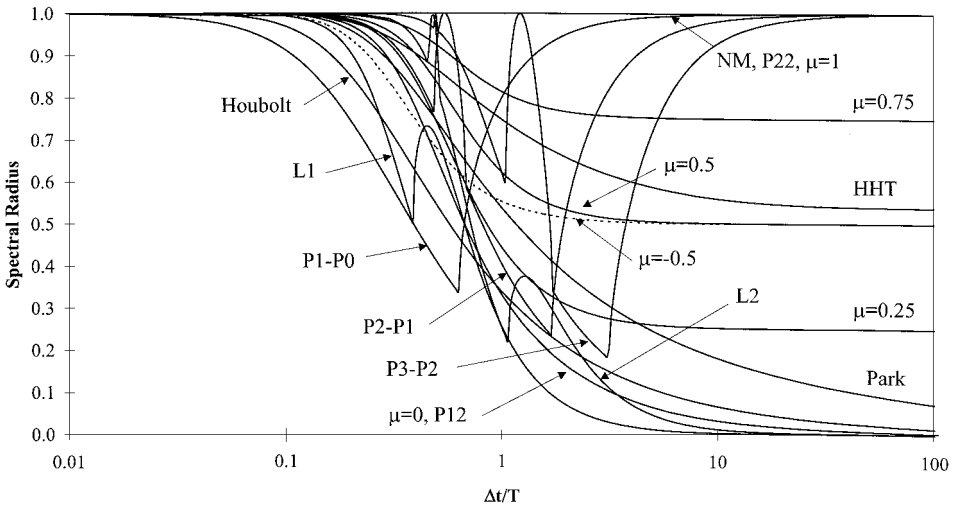

Fig. 1 Comparison of spectral radii for the present algorithms ($n = 2$, $\xi = 0$).

Padé approximation; P1-P0—displacement-based discontinuous Galerkin–Hamilton's law,[28] linear trial interpolation and constant test interpolation; P2-P1—displacement-based discontinuous Galerkin–Hamilton's law,[28] parabolic trial interpolation and linear test interpolation; P3-P2—displacement-based discontinuous Galerkin–Hamilton's law,[28] cubic trial interpolation and parabolic test interpolation; L1—Hamilton's law with initial conditions weakly enforced,[24] linear interpolation; L2—Hamilton's law with initial conditions weakly enforced,[24] parabolic interpolation.

B. Algorithmic Damping Ratios and Relative Period Errors

The errors of an algorithm can be measured by the algorithmic damping ratio and the relative period error. They are related to the numerical dissipation and dispersion (or the amplitude and phase errors). If the complex eigenvalues λ_1 and λ_2 of the undamped numerical amplification matrix (i.e., $\xi = 0$) are expressed in the following form:

$$\lambda_{1,2} = \exp[\bar{\omega}\Delta t(-\bar{\xi} \pm i)] \tag{49}$$

where $i = \sqrt{-1}$, then $\bar{\omega}$, $\bar{\xi}$ are defined as the algorithmic frequency and algorithmic damping ratio, respectively. The relative period error is defined as $(\bar{T} - T)/T$, where $T = 2\pi/\omega$ and $\bar{T} = 2\pi/\bar{\omega}$.

The relative period error and algorithmic damping ratio for the present algorithms in general can be written as

relative period error

$$= \frac{(n - \mu + n\mu^2)\omega^{2n}\Delta t^{2n}}{(2n+1)(2n-1)[(2n-1)!!]^2 2^{2n-2}(1+\mu)^2} + \mathcal{O}(\Delta t^{2n+2}) \tag{50a}$$

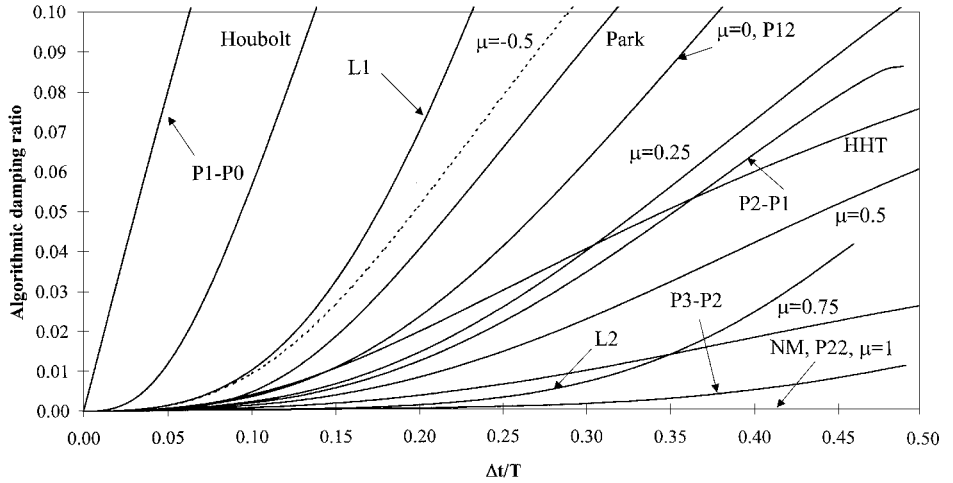

Fig. 2 Comparison of algorithmic damping ratios for the present algorithms ($n = 2, \xi = 0$).

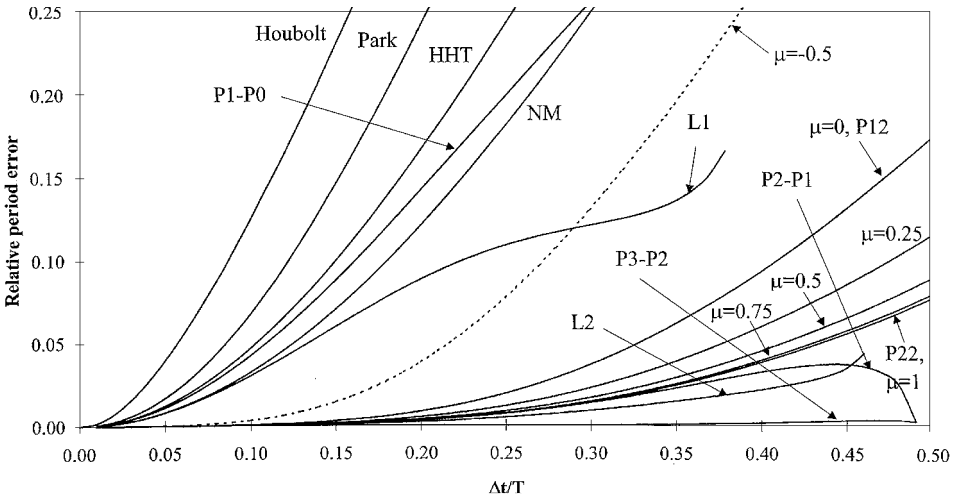

Fig. 3 Comparison of relative period errors for the present algorithms ($n = 2, \xi = 0$).

algorithmic damping ratio

$$= \frac{(1-\mu)\omega^{2n-1}\Delta t^{2n-1}}{[(2n-1)!!]^2 2^{2n-1}(1+\mu)} + \mathcal{O}(\Delta t^{2n+1}) \tag{50b}$$

where $(2n-1)!! = (2n-1)(2n-3)\cdots 5\times 3\times 1$. From Eq. (50a) one can see that no real value of μ could eliminate the leading-error term further. However, the leading-error term is minimized if $\mu = 1$. The algorithms are nondissipative as well when $\mu = 1$.

Figure 2 shows the algorithmic damping ratios for various algorithms. The present third-order algorithms ($n = 2$) are comparable to the second-order algorithms. Figure 3 shows the relative period errors for various algorithms. The relative period errors for the present third-order algorithms are smaller than those for the second-order algorithms.

In Figs. 1–3 the spectral radius, the algorithmic damping ratio, and the relative period error for the present formulation with $n = 2$ and $\mu = -0.5$ are shown. The algorithm has the same ultimate spectral radius as $\mu = 0.5$. However, the algorithm is not as accurate as the algorithm with $\mu = 0.5$. This is also true for other values of $|\mu|$ for other higher-order algorithms. As a result, only when $0 \le \mu \le 1$ would it be of interest in general.

VI. Multi-Degree-of-Freedom Systems

The present formulations can be extended to multi-degree-of-freedom systems directly. The corresponding Hamilton's principle can be written as

$$\delta \int_0^{\Delta t} (T - V)\,\mathrm{d}t + \int_0^{\Delta t} \{\delta \boldsymbol{u}\}^T \{\boldsymbol{F}_c\}\,\mathrm{d}t = 0 \tag{51}$$

with initial conditions $\{\boldsymbol{u}(0)\} = \{\boldsymbol{u}_0\}$ and $\{\dot{\boldsymbol{u}}(0)\} = \{\boldsymbol{v}_0\}$ where the kinetic energy T, the potential energy V, and the nonconservative viscous damping force vector $\{\boldsymbol{F}_c\}$ are given by

$$T = \frac{1}{2}\{\dot{\boldsymbol{u}}(t)\}^T [\boldsymbol{M}]\{\dot{\boldsymbol{u}}(t)\}$$

$$V = \frac{1}{2}\{\boldsymbol{u}(t)\}^T [\boldsymbol{K}]\{\boldsymbol{u}(t)\} - \{\boldsymbol{F}\}^T \{\boldsymbol{u}(t)\}$$

$$\{\boldsymbol{F}_c\} = -[\boldsymbol{C}]\{\dot{\boldsymbol{u}}(t)\}, \qquad \{\boldsymbol{F}\} = \sum_p \{\boldsymbol{f}_p\} t^p \tag{52}$$

Similar to the single-degree-of-freedom systems, $\{\boldsymbol{u}(t)\}$ is approximated by

$$\{\boldsymbol{u}(t)\} = \{\boldsymbol{u}_0\} + \{\boldsymbol{v}_0\}t + \{\boldsymbol{u}_2\}t^2 + \cdots + \{\boldsymbol{u}_{n+1}\}t^{n+1}$$

$$\text{for} \quad 0 \le t \le \Delta t \tag{53a}$$

or

$$\{\boldsymbol{u}(\tau)\} = \{\boldsymbol{u}_0\} + \{\boldsymbol{v}_0\}\Delta t\,\tau + \{\boldsymbol{U}_1\}\tau^2 + \cdots + \{\boldsymbol{U}_n\}\tau^{n+1}$$

$$\text{for} \quad 0 \le \tau \le 1 \tag{53b}$$

Hence

$$\{\boldsymbol{v}(\tau)\} = \{\boldsymbol{v}_0\} + \frac{\left(2\{\boldsymbol{U}_1\}\tau + \cdots + (n+1)\{\boldsymbol{U}_n\}\tau^n\right)}{\Delta t}$$

$$\text{for} \quad 0 \le \tau \le 1 \tag{54}$$

Similar to Eq. (25), the unknowns $\{\boldsymbol{U}_1\}, \{\boldsymbol{U}_2\}, \ldots, \{\boldsymbol{U}_n\}$ are obtained by solving

$$\left(\begin{bmatrix} 2\boldsymbol{M} & & & & & a_1\boldsymbol{K}\Delta t^2 & (n+1)a_1\boldsymbol{C}\Delta t + b_1\boldsymbol{K}\Delta t^2 \\ 2\boldsymbol{C}\Delta t & 6\boldsymbol{M} & & & & a_2\boldsymbol{K}\Delta t^2 & (n+1)a_2\boldsymbol{C}\Delta t + b_2\boldsymbol{K}\Delta t^2 \\ \boldsymbol{K}\Delta t^2 & 3\boldsymbol{C}\Delta t & 12\boldsymbol{M} & & & a_3\boldsymbol{K}\Delta t^2 & (n+1)a_3\boldsymbol{C}\Delta t + b_3\boldsymbol{K}\Delta t^2 \\ & \ddots & \ddots & \ddots & & \vdots & \vdots \\ & & \ddots & \ddots & (n-2)(n-1)\boldsymbol{M} & a_{n-2}\boldsymbol{K}\Delta t^2 & (n+1)a_{n-2}\boldsymbol{C}\Delta t + b_{n-2}\boldsymbol{K}\Delta t^2 \\ & & & \ddots & (n-1)\boldsymbol{C}\Delta t & (n-1)n\boldsymbol{M} + a_{n-1}\boldsymbol{K}\Delta t^2 & (n+1)a_{n-1}\boldsymbol{C}\Delta t + b_{n-1}\boldsymbol{K}\Delta t^2 \\ & & & & \boldsymbol{K}\Delta t^2 & n\boldsymbol{C}\Delta t + a_n\boldsymbol{K}\Delta t^2 & n(n+1)\boldsymbol{M} + (n+1)a_n\boldsymbol{C}\Delta t + b_n\boldsymbol{K}\Delta t^2 \end{bmatrix}\right)\begin{Bmatrix} \boldsymbol{U}_1 \\ \boldsymbol{U}_2 \\ \boldsymbol{U}_3 \\ \vdots \\ \boldsymbol{U}_{n-2} \\ \boldsymbol{U}_{n-1} \\ \boldsymbol{U}_n \end{Bmatrix}$$

$$= -\begin{Bmatrix} \boldsymbol{K}\boldsymbol{u}_0\Delta t^2 + \boldsymbol{C}\boldsymbol{v}_0\Delta t^2 \\ \boldsymbol{K}\boldsymbol{v}_0\Delta t \\ 0 \\ \vdots \\ 0 \\ 0 \\ 0 \end{Bmatrix} + \sum_p \Delta t^{p+2}\begin{Bmatrix} \boldsymbol{f}_p W_{1p} \\ \boldsymbol{f}_p W_{2p} \\ \boldsymbol{f}_p W_{3p} \\ \vdots \\ \boldsymbol{f}_p W_{n-2,p} \\ \boldsymbol{f}_p W_{n-1,p} \\ \boldsymbol{f}_p W_{np} \end{Bmatrix} = \begin{Bmatrix} \boldsymbol{F}_1 \\ \boldsymbol{F}_2 \\ \boldsymbol{F}_3 \\ \vdots \\ \boldsymbol{F}_{n-2} \\ \boldsymbol{F}_{n-1} \\ \boldsymbol{F}_n \end{Bmatrix} \tag{55}$$

In the present formulation there are only nN unknowns if $[\boldsymbol{M}]$, $[\boldsymbol{C}]$, and $[\boldsymbol{K}]$ are $N \times N$ matrices. If the problem is cast into the first-order form and then solved by the mixed formulation, there would be $2nN$ unknowns. Therefore, it is computationally more efficient to have the second-order equations solved directly, instead of transforming to the equivalent first-order forms.

The coefficient matrix in Eq. (55) is very sparse and almost in an upper triangular form. The orders of the equations and the unknown variables could be rearranged to give a half-bandwidth thrice of the original problem for most of the columns in the lower half triangle. For other conventional higher-order methods the coefficient matrix would have a half-bandwidth n times of the original problem even after variables and equations rearrangement. The computational efficiency of the present formulation over the conventional formulations is obvious for very high-order algorithms. Furthermore, the present algorithms are unconditionally stable.

In addition, because of the special structure of the coefficient matrix in Eq. (55), Gauss elimination can be carried out by keeping track of the last two columns of submatrices only. The overall $nN \times nN$ coefficient matrix need not be assembled at all.

VII. Numerical Examples

A. Example 1: Equivalence to the Generalized Padé Approximations

Consider the free vibration of a single-degree-of-freedom mass-spring system given by Eq. (10) with initial condition $u(0) = u_0$ and $\dot{u}(0) = v_0$. The present third-order-accurate algorithm with $n = 2$ is employed. The variations are considered to be made up of

$$\psi_1(\tau) = \tau(1-\tau) + \frac{1}{180}\phi_1(\tau) + \frac{1}{180}\frac{4\mu - 1}{1+\mu}\phi_2(\tau)$$

$$\psi_2(\tau) = \tau^2(1-\tau) + \frac{1}{180}\frac{1}{1+\mu}\phi_1(\tau) + \frac{1}{1260}\frac{5 + 10\mu + 12\mu^2}{(1+\mu)^2}\phi_2(\tau) \tag{56}$$

where

$$\phi_1(\tau) = 840(5 - 43\tau + 97\tau^2 - 63\tau^3)\tau(1-\tau)$$

$$\phi_2(\tau) = 840(-3 + 27\tau - 63\tau^2 + 42\tau^3)\tau(1-\tau)$$

$u(t)$ is assumed to be in the form

$$u(t) = u_0 + v_0\Delta t\,\tau + U_1\tau^2 + U_2\tau^3 \tag{57}$$

The approximate solutions of u_1 and v_1 at $t = \Delta t$ are given by

$$u_1 = u_0 + v_0\Delta t + U_1 + U_2 \tag{58a}$$

$$v_1 = v_0 + \frac{(2U_1 + 3U_2)}{\Delta t} \tag{58b}$$

From Eq. (25) U_1 and U_2 are obtained by solving

$$\left(\begin{pmatrix} 2 & 0 \\ 0 & 6 \end{pmatrix} + 2\xi\omega\Delta t\begin{pmatrix} 0 & 3a_1 \\ 2 & 3a_2 \end{pmatrix} + \omega^2\Delta t^2\begin{pmatrix} a_1 & b_1 \\ a_2 & b_2 \end{pmatrix}\right)\begin{Bmatrix} U_1 \\ U_2 \end{Bmatrix} = -\begin{Bmatrix} \omega^2\Delta t^2 u_0 + 2\xi\omega\Delta t^2 v_0 \\ \omega^2\Delta t^3 v_0 \end{Bmatrix} \tag{59}$$

here

$$a_1 = -\frac{1}{3}\frac{1}{1+\mu}, \qquad a_2 = \frac{2}{3}\frac{2+\mu}{1+\mu}$$

$$b_1 = -\frac{1}{3}\frac{2+\mu}{(1+\mu)^2}, \qquad b_2 = \frac{1}{3}\frac{5 + 5\mu + 2\mu^2}{(1+\mu)^2} \tag{60}$$

After solving for U_1 and U_2 and some algebraic manipulations, u_1 and v_1 in Eq. (58) are related to u_0 and v_0 by

$$u_1 = A_{11}u_0 + A_{12}v_0 \tag{61a}$$

$$v_1 = A_{21}u_0 + A_{22}v_0 \tag{61b}$$

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

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

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

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

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

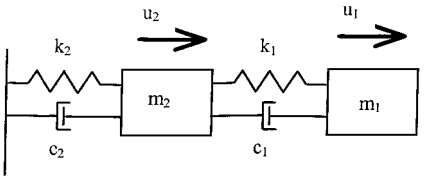

Fig. 4 Two-degree-of-freedom system ($m_1 = m_2 = m$, $c_1 = c_2 = c$, $k_1 = k_2 = k$).

which are the generalized Padé approximations for second-order equations. When $\mu = 0$, Eq. (62) is equivalent to the Padé (1, 2) approximation. When $\mu = 1$, Eq. (62) is equivalent to the Padé (2, 2) approximation.

B. Example 2: Two-Degree-of-Freedom System

To illustrate the truncation errors of the present higher-order algorithms on multi-degree-of-freedom systems, consider a two-degree-of-freedom system as shown in Fig. 4. The kinetic energy T, the potential energy V, and the nonconservative viscous damping forces f_1 and f_2 are given by

$$T = \tfrac{1}{2}m_1\dot{u}_1^2 + \tfrac{1}{2}m_2\dot{u}_2^2, \qquad V = \tfrac{1}{2}k_1(u_1 - u_2)^2 + \tfrac{1}{2}k_2u_2^2$$
$$f_1 = -c_1\dot{u}_1 + c_1\dot{u}_2, \qquad f_2 = c_1\dot{u}_1 - (c_1 + c_2)\dot{u}_2 \tag{63}$$

The initial conditions are

$$\begin{Bmatrix} u_1(0) \\ u_2(0) \end{Bmatrix} = \begin{Bmatrix} u_{10} \\ u_{20} \end{Bmatrix}, \qquad \begin{Bmatrix} \dot{u}_1(0) \\ \dot{u}_2(0) \end{Bmatrix} = \begin{Bmatrix} v_{10} \\ v_{20} \end{Bmatrix} \tag{64}$$

The present fifth-order-accurate algorithm with $n = 3$ is used. The variations are considered to be made up of

$$\psi_1(\tau) = \tau(1-\tau) + \frac{1}{600}\frac{\mu - 1}{1+\mu}\phi_1(\tau) + \frac{1}{10{,}500}\frac{-47 + 4\mu + 23\mu^2}{(1+\mu)^2}\phi_2(\tau)$$

$$\psi_2(\tau) = \tau^2(1-\tau) + \frac{1}{4200}\frac{5\mu - 2}{1+\mu}\phi_1(\tau) + \frac{1}{21{,}000}\frac{-39 + 34\mu + 45\mu^2}{(1+\mu)^2}\phi_2(\tau)$$

$$\psi_3(\tau) = \tau^3(1-\tau) + \frac{1}{1400}\frac{\mu}{1+\mu}\phi_1(\tau) + \frac{1}{63{,}000}\frac{-35 + 98\mu + 97\mu^2}{(1+\mu)^2}\phi_2(\tau)$$

where

$$\phi_1(\tau) = 1260(-24 + 326\tau - 1274\tau^2 + 1876\tau^3 - 924\tau^4)\tau(1-\tau)$$

and

$$\phi_2(\tau) = 1260(11 - 154\tau + 616\tau^2 - 924\tau^3 + 462\tau^4)\tau(1-\tau) \tag{65}$$

$u_1(t)$ and $u_2(t)$ are assumed to be in the form

$$u_1(t) = u_1(\tau\Delta t) = u_{10} + v_{10}\Delta t\tau + U_{11}\tau^2 + U_{12}\tau^3 + U_{13}\tau^4$$
$$u_2(t) = u_2(\tau\Delta t) = u_{20} + v_{20}\Delta t\tau + U_{21}\tau^2 + U_{22}\tau^3 + U_{23}\tau^4 \tag{66}$$

The approximate solutions of u_1 and u_2 at $t = \Delta t$ are given by

$$u_{11} = u_{10} + v_{10}\Delta t + U_{11} + U_{12} + U_{13} \tag{67a}$$

$$u_{21} = u_{20} + v_{20}\Delta t + U_{21} + U_{22} + U_{23} \tag{67b}$$

From Eq. (55), U_{11}, U_{12}, U_{13}, U_{21}, U_{22}, and U_{23} are obtained by solving

$$\begin{bmatrix}
2m & 0 & a_1k\Delta t^2 & -a_1k\Delta t^2 & 4ca_1\Delta t + b_1k\Delta t^2 & -4ca_1\Delta t - b_1k\Delta t^2 \\
0 & 2m & -a_1k\Delta t^2 & 2a_1k\Delta t^2 & -4ca_1\Delta t - b_1k\Delta t^2 & 8ca_1\Delta t + 2b_1k\Delta t^2 \\
2c\Delta t & -2c\Delta t & 6m + a_2k\Delta t^2 & -a_2k\Delta t^2 & 4ca_2\Delta t + b_2k\Delta t^2 & -4ca_2\Delta t - b_2k\Delta t^2 \\
-2c\Delta t & 4c\Delta t & -a_2k\Delta t^2 & 6m + 2a_2k\Delta t^2 & -4ca_2\Delta t - b_2k\Delta t^2 & 8ca_2\Delta t + 2b_2k\Delta t^2 \\
k\Delta t^2 & -k\Delta t^2 & 3c\Delta t + a_3k\Delta t^2 & -3c\Delta t - a_3k\Delta t^2 & 12m + 4ca_3\Delta t + b_3k\Delta t^2 & -4ca_3\Delta t - b_3k\Delta t^2 \\
-k\Delta t^2 & 2k\Delta t^2 & -3c\Delta t - a_3k\Delta t^2 & 6c\Delta t + 2a_3k\Delta t^2 & -4ca_3\Delta t - b_3k\Delta t^2 & 12m + 8ca_3\Delta t + 2b_3k\Delta t^2
\end{bmatrix}$$

$$\times \begin{Bmatrix} U_{11} \\ U_{21} \\ U_{12} \\ U_{22} \\ U_{13} \\ U_{23} \end{Bmatrix} = \begin{Bmatrix} -\Delta t^2(ku_{10} - ku_{20} + cv_{10} - cv_{20}) \\ -\Delta t^2(-ku_{10} + 2ku_{20} - cv_{10} + 2cv_{20}) \\ -k\Delta t^3(v_{10} - v_{20}) \\ -k\Delta t^3(-v_{10} + 2v_{20}) \\ 0 \\ 0 \end{Bmatrix} \tag{68}$$

where

$$a_1 = \frac{1}{10}\frac{1}{1+\mu}, \qquad a_2 = -\frac{3}{10}\frac{3+\mu}{1+\mu}, \qquad a_3 = \frac{3}{5}\frac{3+2\mu}{1+\mu} \tag{69a}$$

$$b_1 = \frac{2}{25}\frac{3+2\mu}{(1+\mu)^2}, \qquad b_2 = -\frac{4}{25}\frac{11 + 11\mu + 3\mu^2}{(1+\mu)^2}$$
$$b_3 = \frac{3}{25}\frac{21 + 28\mu + 11\mu^2}{(1+\mu)^2} \tag{69b}$$

The leading truncation error terms, when comparing u_{11} and u_{21} in Eq. (67) with the exact solutions, are

$$\frac{1}{7200m^5}\frac{\mu - 1}{1+\mu}\begin{pmatrix} k(c^2 - km)(34c^2 - 5km)u_{10} - k(c^2 - km)(55c^2 - 8km)u_{20} + \\ c(34c^4 - 52mkc^2 + 15m^2k^2)v_{10} - c(55c^4 - 84mkc^2 + 24m^2k^2)v_{20} \end{pmatrix}\Delta t^6 \tag{70a}$$

$$\frac{1}{7200m^5}\frac{1 - \mu}{1+\mu}\begin{pmatrix} k(c^2 - km)(55c^2 - 8km)u_{10} - k(c^2 - km)(89c^2 - 13km)u_{20} + \\ c(55c^4 - 84mkc^2 + 24m^2k^2)v_{10} - c(89c^4 - 136mkc^2 + 39m^2k^2)v_{20} \end{pmatrix}\Delta t^6 \tag{70b}$$

respectively. Obviously, the present algorithm is fifth-order accurate if $\mu \neq 1$ and sixth-order accurate if $\mu = 1$.

VIII. Conclusions

In this paper unconditionally stable higher-order-accurate time-step-integration algorithms based on Hamilton's principle are constructed systematically. The numerical dissipation is directly controllable by the algorithmic parameter μ. The initial conditions are strongly enforced in the formulation. The variations are assumed to be in the form of $\Sigma\, \alpha_k \tau^k (1 - \tau)$. Instead of specifying the variation functions explicitly, the algorithmic parameters a_k and b_k are used to investigate the algorithm characteristics. By choosing the algorithmic parameters a_k and b_k carefully, the algorithms can be unconditionally stable and at least $(2n - 1)$th-order accurate. In fact, the resultant algorithms are equivalent to the generalized Padé approximations for second-order equations. The required variation functions can be evaluated from the algorithmic parameters a_k and b_k.

Hamilton's principle can be used to construct unconditionally stable higher-order-accurate time-step-integration algorithms directly and systemically as well, provided appropriate constrained variations are used. There is no need to weakly enforce the initial conditions (as in the discontinuous formulations) and/or the velocity-displacement relation (as in the mixed formulations).

References

[1]Hairer, E., Nørsett, S. P., and Wanner, G., *Solving Ordinary Differential Equations I*, Springer-Verlag, Berlin, 1987.

[2]Hairer, E., and Wanner, G., *Solving Ordinary Differential Equations II*, Springer-Verlag, Berlin, 1991.

[3]Hughes, T. J. R., *The Finite Element Method: Linear Static and Dynamic Finite Element Analysis*, Prentice–Hall, Upper Saddle River, NJ, 1987.

[4]Wood, W. L., *Practical Time-Stepping Schemes*, Clarendon, Oxford, 1990.

[5]Zienkiewicz, O. C., and Taylor, R. L., *The Finite Element Method*, 4th ed., McGraw–Hill, New York, 1991.

[6]Argyris, J. H., and Scharpf, D. W., "Finite Elements in Time and Space," *Journal of the Royal Aeronautical Society*, Vol. 73, Dec. 1969, pp. 1041–1044.

[7]Fried, I., "Finite-Element Analysis of Time-Dependent Phenomena," *AIAA Journal*, Vol. 7, No. 6, 1969, pp. 1170–1173.

[8]Oden, J. T., "A General Theory of Finite Element.2. Application," *International Journal for Numerical Methods in Engineering*, Vol. 1, 1969, pp. 247–259.

[9]Bailey, C. D., "Application of Hamilton's Law of Varying Action," *AIAA Journal*, Vol. 13, No. 9, 1975, pp. 1154–1157.

[10]Bailey, C. D., "The Method of Ritz Applied to the Equation of Hamilton," *Computer Methods in Applied Mechanics and Engineering*, Vol. 7, 1976, pp. 235–247.

[11]Gurtin, M. E., "Variational Principles for Linear Elastodynamics," *Archive for Rational Mechanics and Analysis*, Vol. 16, 1963, pp. 34–50.

[12]Gurtin, M. E., "Variational Principles for Linear Initial Value Problems," *Quarterly Journal of Applied Mathematics*, Vol. 22, No. 3, 1964, pp. 262–256.

[13]Smith, D. R., and Smith, C. V., Jr., "When is Hamilton's Principle an Extremum Principle," *AIAA Journal*, Vol. 12, No. 11, 1974, pp. 1573–1576.

[14]Howard, G. F., and Penny, J. E. T., "The Accuracy and Stability of Time Domain Time Finite Element Solutions," *Journal of Sound and Vibration*, Vol. 61, 1978, pp. 585–595.

[15]Peters, D. A., and Izadpanah, A. P., "hp-Version Finite Elements for the Space-Time Domain," *Computational Mechanics*, Vol. 3, 1988, pp. 73–88.

[16]Hamilton, W. R., "On a General Method in Dynamics," *Philosophical Transactions of the Royal Society of London*, 1834, pp. 247–308.

[17]Hamilton, W. R., "Second Essay on a General Method in Dynamics," *Philosophical Transactions of the Royal Society of London*, 1835, pp. 95–144.

[18]Bailey, C. D., "Further Remarks on the Law of Varying Action and the Symbol δ," *Journal of Sound and Vibration*, Vol. 131, 1989, pp. 331–344.

[19]Riff, R., and Baruch, M., "Stability of Time Finite Element," *AIAA Journal*, Vol. 22, No. 8, 1984, pp. 1171–1173.

[20]Riff, R., and Baruch, M., "Time Finite Element Discretization of Hamilton's Law of Varying Action," *AIAA Journal*, Vol. 22, No. 9, 1984, pp. 1310–1318.

[21]Baruch, M., and Riff, R., "Hamilton's Principle, Hamilton's Law-6^n Correct Formulations," *AIAA Journal*, Vol. 20, No. 5, 1982, pp. 687–692.

[22]Simkins, T. E., "Unconstrained Variational Statements for Initial and Boundary Value Problems," *AIAA Journal*, Vol. 16, No. 6, 1978, pp. 559–563.

[23]Borri, M., Ghiringhelli, G. L., Lanz, M., Mantegazza, P., and Merlini, T., "Dynamic Response of Mechanical Systems by a Weak Hamilton Formulation," *Computers and Structures*, Vol. 20, 1985, pp. 495–508.

[24]Cannarozzi, M., and Mancuso, M., "Formulation and Analysis of Variational Methods for Time Integration of Linear Elastodynamics," *Computer Methods in Applied Mechanics and Engineering*, Vol. 127, 1995, pp. 241–257.

[25]Hodges, D. H., and Bless, R. R., "Weak Hamiltonian Finite Element Method for Optimal Control Problems," *Journal of Guidance, Control, and Dynamics*, Vol. 14, No. 1, 1991, pp. 148–155.

[26]Hodges, D. H., and Hou, L. J., "Shape Functions for Mixed p-Version Finite Elements in the Time Domain," *Journal of Sound and Vibration*, Vol. 145, 1991, pp. 169–178.

[27]Aharoni, D., and Bar-Yoseph, P., "Mixed Finite Element Formulations in the Time Domain for Solution of Dynamics," *Computational Mechanics*, Vol. 9, 1992, pp. 359–374.

[28]Borri, M., and Bottasso, C., "A General Framework for Interpreting Time Finite Element Formulations," *Computational Mechanics*, Vol. 13, 1993, pp. 133–142.

[29]Clough, R. W., and Penzien, J., *Dynamics of Structures*, McGraw–Hill, New York, 1993.

[30]Agrawal, O. P., and Saigal, S., "A Novel, Computationally Efficient, Approach for Hamilton's Law of Varying Action," *International Journal of Mechanical Sciences*, Vol. 29, 1987, pp. 285–292.

[31]Fung, T. C., "Weighting Parameters for Unconditionally Stable Higher-Order Accurate Time Step Integration Algorithms Part 2. Second Order Equations," *International Journal for Numerical Methods in Engineering*, Vol. 45, 1999, pp. 971–1006.

[32]Fung, T. C., "Weighting Parameters for Unconditionally Stable Higher-Order Accurate Time Step Integration Algorithms Part 1. First Order Equations," *International Journal for Numerical Methods in Engineering*, Vol. 45, 1999, pp. 941–970.

[33]Fung, T. C., "Complex-Time-Step Newmark Methods with Controllable Numerical Dissipation," *International Journal for Numerical Methods in Engineering*, Vol. 41, 1998, pp. 65–93.

[34]Fung, T. C., "Complex-Time-Step Methods for Transient Analysis," *International Journal for Numerical Methods in Engineering*, Vol. 46, 1999, pp. 1253–1271.

[35]Fung, T. C., "Higher-Order Accurate Least-Squares Methods for First-Order Initial Value Problems," *International Journal for Numerical Methods in Engineering*, Vol. 45, 1999, pp. 77–99.

S. Saigal
Associate Editor