



ELSEVIER

Comput. Methods Appl. Mech. Engrg. 113 (1994) 1-9

**Computer methods  
in applied  
mechanics and  
engineering**

# A unified set of single-step asymptotic annihilation algorithms for structural dynamics

Gregory M. Hulbert

*Department of Mechanical Engineering and Applied Mechanics, 321 W.E. Lay Automotive Laboratory,  
The University of Michigan, Ann Arbor, MI 48109-2121, USA*

Received 19 September 1991

Revised manuscript received 23 March 1993

---

## Abstract

When solving the equations of structural dynamics using direct time integration methods, algorithmic damping is useful in controlling spurious high-frequency oscillations. Ideally, an algorithm should possess asymptotic annihilation of the high-frequency response, i.e., spurious oscillations in the high frequencies are eliminated after one time step. In this paper, a new class of asymptotic annihilation algorithms for structural dynamics is presented that possesses little numerical dissipation in the low-frequency regime. The algorithms are based upon using finite elements in time. The displacement and velocity fields are interpolated independently using time-discontinuous functions. The equations of motion, displacement-velocity compatibility and the time continuity of displacement and velocity are satisfied weakly. Asymptotic annihilation is achieved by choosing the displacement and velocity fields to be of equal order. Algorithms of any desired order of temporal accuracy can be obtained by appropriate choice of the finite element interpolations in time. An analysis of the proposed class of algorithms is given proving the asymptotic annihilation property and the spectral equivalence of the algorithms to the upper diagonal of the Padé approximation table. Results from finite difference analyses are presented showing the spectral behavior of the algorithms as well as their dissipation and dispersion properties in the low frequency regime.

---

## 1. Introduction

When solving the equations of structural dynamics using direct time integration methods, algorithmic damping is useful to control spurious high-frequency oscillations. Algorithmic damping also has been found to be helpful when solving problems that include constraints, e.g., contact. Typically, the physical high-frequency characteristics of the structure are modeled inaccurately. These inaccuracies are manifested in the time response as spurious high-frequency oscillations. One approach towards minimizing spurious oscillations in the solution is to post-process the results by applying a low-pass filter (see [1, p. 513] for one suggested technique). However, this may not be a good strategy when solving problems exhibiting nonlinear material behavior, e.g., plasticity. For such problems, spurious oscillations enter into the constitutive routines and could adversely affect the computed response. A second problem with post-processing strategies may result if the unfiltered response is unstable. Such instabilities may occur, for example, when using the trapezoidal rule to integrate the response of a constrained dynamical system; see [2] for an example. While post-processing will remove the instability from the filtered response, the instability still remains in the unfiltered results. Because the filtered results are not fed back into the time integration algorithm, eventually, for long-time simulations, the instability of the unfiltered response will cause instabilities in the filtered response due to round-off and machine overflow.

Ideally, any nonzero response in the spurious, high-frequency modes should be eliminated (or nearly

eliminated) after one time step; we refer to this attribute as annihilation of the high-frequency response. The challenge is to design a time integration algorithm that possesses asymptotic annihilation (i.e., in the high-frequency limit, the response is annihilated) without introducing excessive numerical dissipation in the low-frequency response of interest in structural dynamics problems.

Perhaps the best-known asymptotic annihilation algorithm is the implicit Euler method. It is also well known that the implicit Euler method is inappropriate for structural dynamics as it is only first-order accurate and is overly dissipative in the low-frequency domain. Houbolt's method is the most commonly used algorithm that achieves asymptotic annihilation [3]. However, its asymptotic annihilation is achieved at the expense of introducing too much dissipation in the low-frequency regime; to obtain the same accuracy as other commonly used time integration algorithms, Houbolt's method requires using a smaller time step. It was shown in [4] that it is not possible to improve upon the dissipation characteristics of Houbolt's method if attention is restricted to asymptotic annihilation algorithms that may be cast in three-step linear multistep (LMS) form. Three-step LMS methods are of particular interest because nearly all of the time integration algorithms implemented in commercial finite element programs have this form. Thus, to obtain a practically useful asymptotic annihilation algorithm requires developing four-step (or higher) LMS methods or developing algorithms that do not fall within the class of LMS methods. For example, a successful asymptotic annihilation algorithm is the method of Park, which for structural dynamics problems, is a six-step LMS method [5].

Most time integration algorithms for structural dynamics have been developed by discretizing the semidiscrete equations of motion using finite difference methods for ordinary differential equations. An alternative approach is to discretize the equations of motion using finite elements in time. The basic idea is to permit the unknown fields to be discontinuous with respect to time; an appropriate variational formulation then involves integration over discrete time intervals. This approach, known as the time-discontinuous Galerkin (TDG) method, generates single-step time integration algorithms [6]. The resultant algorithms are not LMS methods (the one exception is the implicit Euler equivalent); thus the potential exists to develop useful asymptotic annihilation schemes.

In this paper, a time-discontinuous Galerkin method is presented that generates a class of unconditionally stable, asymptotic annihilation algorithms. The method is based upon employing displacement and velocity fields as the unknowns and using equal-order interpolation for both fields. Any desired order of temporal accuracy can be obtained by appropriate choice of the interpolation functions. For example, a third-order accurate asymptotic annihilation algorithm ensues by using linear interpolations for displacement and velocity. We show that the TDG method presented herein is equivalent to the upper diagonal of the Padé approximation table, in the sense of identical stability polynomials. Since Padé approximations represent the highest order accuracy possible for given rational numerators and denominators [7], the TDG method generates nearly optimal algorithms in terms of balancing computational cost and accuracy while achieving asymptotic annihilation.

The ordinary differential equations of linear structural dynamics may be written in the following first-order form:

$$M\dot{\mathbf{u}} + C\mathbf{v} + K\mathbf{u} = \mathbf{f} , \quad (1)$$

$$\dot{\mathbf{u}} = \mathbf{v} , \quad (2)$$

$$\mathbf{u}(0) = \mathbf{d}_0 , \quad (3)$$

$$\mathbf{v}(0) = \mathbf{v}_0 , \quad (4)$$

where  $M$ ,  $C$  and  $K$  are the respective mass, damping, and stiffness matrices;  $\mathbf{u} = \mathbf{u}(t)$  and  $\mathbf{v} = \mathbf{v}(t)$  are the vectors of unknown nodal displacements and velocities, respectively, both have length  $n_{eq}$ ;  $\mathbf{f}$  is the prescribed load vector of length  $n_{eq}$ ;  $\mathbf{d}_0$  and  $\mathbf{v}_0$  are the initial displacement and velocity vectors, respectively. A superposed dot denotes differentiation with respect to time.  $M$  is assumed to be symmetric positive-definite while  $C$  and  $K$  are assumed to be symmetric positive-semidefinite.

## 2. A time-discontinuous Galerkin formulation

Equations (1)–(4) may be written in the more compact form

$$A_0 \dot{U} + A_1 U = F, \quad (5)$$

$$U(0) = U_0, \quad (6)$$

where

$$A_0 = \begin{bmatrix} K & \mathbf{0} \\ \mathbf{0} & M \end{bmatrix}, \quad A_1 = \begin{bmatrix} \mathbf{0} & -K \\ K & C \end{bmatrix}, \quad (7)$$

$$U = \begin{Bmatrix} u \\ v \end{Bmatrix}, \quad U_0 = \begin{Bmatrix} d_0 \\ v_0 \end{Bmatrix}, \quad F = \begin{Bmatrix} \mathbf{0} \\ f \end{Bmatrix}. \quad (8)$$

Consider a partition of the time domain,  $I = ]0, T[$ , having the form:  $0 = t_0 < t_1 < \dots < t_N = T$ . Let  $I_n = ]t_{n-1}, t_n[$ . Let

$$U^h = \begin{Bmatrix} u^h \\ v^h \end{Bmatrix} \quad (9)$$

denote the finite element approximation to  $U$  and  $\mathcal{P}^k$  denote the space of  $k$ th order polynomials. Then the finite element interpolation functions are:

$$\mathcal{S}^h = \left\{ U^h \in \bigcup_{n=1}^N (\mathcal{P}^k(I_n))^{2n_{eq}} \right\} \quad (10)$$

This choice of interpolation permits the displacement and velocity fields to be discontinuous between time intervals. Assuming the function  $U^h(t)$  to be discontinuous at time  $t_n$ ,

$$U^h(t_n^\pm) = \lim_{\epsilon \rightarrow 0^\pm} U^h(t_n + \epsilon). \quad (11)$$

Particular examples of the time-discontinuous functions employed in the TDG method are shown in Fig. 1; note that the functions are continuous over each time interval but admit 'jumps' at discrete time values  $t_n$ .

The statement of the time-discontinuous Galerkin method is: Find  $U^h \in \mathcal{S}^h$  such that for all  $W^h \in \mathcal{S}^h$ ,

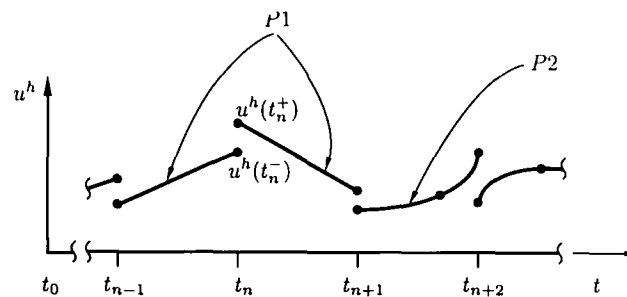


Fig. 1. Examples of time-discontinuous functions.

$$B(\mathbf{W}^h, \mathbf{U}^h)_n = L(\mathbf{W}^h)_n, \quad n = 1, 2, \dots, N, \quad (12)$$

where

$$B(\mathbf{W}^h, \mathbf{U}^h)_n = (\mathbf{W}^h, \mathbf{A}_0 \dot{\mathbf{U}}^h)_{I_n} + (\mathbf{W}^h, \mathbf{A}_1 \mathbf{U}^h)_{I_n} + \mathbf{W}^h(t_{n-1}^+) \cdot \mathbf{A}_0 \mathbf{U}^h(t_{n-1}^+), \quad (13)$$

$$L(\mathbf{W}^h)_n = (\mathbf{W}^h, \mathbf{F})_{I_n} + \mathbf{W}^h(t_{n-1}^+) \cdot \mathbf{A}_0 \mathbf{U}^h(t_{n-1}^-), \quad n = 2, 3, \dots, N, \quad (14)$$

$$L(\mathbf{W}^h)_1 = (\mathbf{W}^h, \mathbf{F})_{I_1} + \mathbf{W}^h(0^+) \cdot \mathbf{A}_0 \mathbf{U}_0, \quad (15)$$

in which

$$(\mathbf{W}, \mathbf{U})_{I_n} = \int_{I_n} \mathbf{W} \cdot \mathbf{U} \, dt. \quad (16)$$

Note that the initial conditions for each time interval are satisfied weakly, thus,  $\mathbf{W}^h \in \mathcal{S}^h$ . The given TDG method is identical to that presented in [6] with equal-order interpolation for displacement and velocity. With the appropriate definitions of  $\mathbf{A}_0$  and  $\mathbf{A}_1$  above, (12) is analogous to TDG methods of other investigators (see [6] for details).

As a specific example, let  $P_j$  denote the TDG algorithm obtained using  $j$ th-order interpolation functions. From (12)–(15), the  $P_1$  algorithm can be written explicitly as

$$\begin{bmatrix} \frac{1}{2}\mathbf{A}_0 + \frac{\Delta t}{3}\mathbf{A}_1 & -\frac{1}{2}\mathbf{A}_0 + \frac{\Delta t}{6}\mathbf{A}_1 \\ \frac{1}{2}\mathbf{A}_0 + \frac{\Delta t}{6}\mathbf{A}_1 & \frac{1}{2}\mathbf{A}_0 + \frac{\Delta t}{3}\mathbf{A}_1 \end{bmatrix} \begin{bmatrix} \mathbf{U}^h(t_n^-) \\ \mathbf{U}^h(t_{n-1}^+) \end{bmatrix} = \begin{bmatrix} \mathbf{0} \\ \mathbf{U}^h(t_{n-1}^-) \end{bmatrix} + \tilde{\mathbf{F}}_n, \quad (17)$$

where  $\mathbf{U}^h(t_{n-1}^+)$  and  $\mathbf{U}^h(t_n^-)$  are the solution vectors at the respective beginning and end of time interval  $I_n$  and

$$\tilde{\mathbf{F}}_n = \begin{bmatrix} \int_{t_{n-1}^+}^{t_n^-} (t - t_{n-1}) \mathbf{f} \, dt \\ \int_{t_{n-1}^+}^{t_n^-} (t_n - t) \mathbf{f} \, dt \end{bmatrix}. \quad (18)$$

There are essentially two sets of displacement and velocity vectors for each discrete time  $t_n$ . The set that corresponds to solutions obtained from standard time integration methods is that at the end of a time interval, i.e.,  $\mathbf{U}^h(t_n^-)$ . For the  $P_1$  formulation,  $\mathbf{U}^h(t_n^-)$  can be expressed directly in terms of  $\mathbf{U}^h(t_{n-1}^-)$  and  $\tilde{\mathbf{F}}_n$  by eliminating  $\mathbf{U}^h(t_{n-1}^+)$  from (17); the resultant equations have the form

$$\begin{bmatrix} \mathbf{M} + \frac{5}{18}\Delta t^2 \mathbf{K} & -\frac{1}{9}\Delta t^2 \mathbf{K} \\ -(\mathbf{M} - \frac{7}{18}\Delta t^2 \mathbf{K}) & \mathbf{M} - \frac{1}{18}\Delta t^2 \mathbf{K} \end{bmatrix} \begin{bmatrix} \mathbf{u}^h(t_n^-) \\ \Delta t \mathbf{v}^h(t_n^-) \end{bmatrix} = \Delta t \tilde{\mathbf{F}}_n + \begin{bmatrix} \mathbf{M} - \frac{2}{9}\Delta t^2 \mathbf{K} & \mathbf{M} \\ -(\mathbf{M} + \frac{1}{9}\Delta t^2 \mathbf{K}) & \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathbf{u}^h(t_{n-1}^-) \\ \Delta t \mathbf{v}^h(t_{n-1}^-) \end{bmatrix}, \quad (19)$$

where  $\Delta t = t_n - t_{n-1}$ . From (19), it is clear that the  $P_1$  formulation results in a system of coupled equations having twice the bandwidth and twice the number of unknowns as commonly used time integration algorithms.

### 3. Stability analysis

In this section, we derive the stability polynomial expression of (12) for arbitrary-order interpolation functions. This stability polynomial is in a form identical to the upper diagonal of the Padé table (Padé methods of type  $(k-1, k)$ ) for approximating  $\exp(i\omega^h \Delta t)$ ; thus (12) engenders a class of asymptotic annihilation methods for structural dynamics.

For purposes of analysis, it is convenient to work with the undamped, single degree-of-freedom model problem,

$$\ddot{u} + \omega^2 u = 0, \quad (20)$$

where  $u(0) = u_0$  and  $\dot{u}(0) = v_0$ . For the system of first-order equations (1)–(4), the corresponding scalar equation is

$$\dot{y} + \lambda y = 0, \quad (21)$$

$$y(0) = y_0, \quad (22)$$

where  $\lambda = \pm i\omega$ . See, e.g., [1] for the steps required to reduce (1)–(4) to (21)–(22). The TDG method for (21) is given by: Find  $y^h \in \mathcal{S}^h$  such that for all  $w^h \in \mathcal{S}^h$ ,

$$b(w^h, y^h)_n = l(w^h)_n, \quad n = 1, 2, \dots, N, \quad (23)$$

where

$$b(w^h, y^h)_n = (w^h, \dot{y}^h)_{I_n} + (w^h, \lambda y^h)_{I_n} + w^h(t_{n-1}^+) y^h(t_{n-1}^+), \quad (24)$$

$$l(w^h)_n = w^h(t_{n-1}^+) y^h(t_{n-1}^-), \quad n = 2, 3, \dots, N, \quad (25)$$

$$l(w^h)_1 = w^h(0^+) y_0, \quad (26)$$

and the interpolation function space,  $\mathcal{S}^h$  is given by

$$\mathcal{S}^h = \left\{ y^h \in \bigcup_{n=1}^N (\mathcal{P}^k(I_n)) \right\}. \quad (27)$$

It is convenient to express the value at the end of a time interval,  $y^h(t_n^-)$ , in terms of the value at the end of the previous time interval. This can be written in the form

$$y^h(t_n^-) = A y^h(t_{n-1}^-), \quad (28)$$

where  $A$  is the amplification factor. An algorithm achieves asymptotic annihilation provided that

$$\lim_{|\bar{\lambda}| \rightarrow \infty} |A| = 0, \quad (29)$$

where  $\bar{\lambda} = \lambda \Delta t$ . In the numerical analysis literature, this property is also referred to as  $L$ -stability [8].

To derive a general expression for  $A$ ,  $y^h$ , restricted to  $I_n$ , is expressed as the Taylor series

$$y^h(t) = \sum_{i=0}^k \frac{(t - t_{n-1})^i}{i!} y_{n-1}^{(i)+}, \quad t_{n-1} \leq t \leq t_n, \quad (30)$$

where  $y_{n-1}^{(i)+}$  denotes the approximation to the  $i$ th derivative of  $y$  at  $t_{n-1}$  and  $k$  is the order of the interpolation polynomial (for notational simplicity, the  $h$ -superscript has been dropped from the approximations of the derivatives). For an arbitrary time step  $n$ , the matrix form of (23) is obtained using (30) and a similar expression for  $w^h$ . Noting that

$$y^h(t_n^-) = \sum_{i=0}^k \frac{\Delta t^i}{i!} y_{n-1}^{(i)+}, \quad (31)$$

the matrix form can be written as

$$\bar{A}Y_n = R_n, \quad (32)$$

where  $\bar{A}$  is a  $k \times k$  matrix,

$$Y_n^t = \{y_{n-1}^+, \Delta t^{(1)+} y_{n-1}^+, \dots, \Delta t^{(k)+} y_{n-1}^+\} \quad (33)$$

and

$$R_n^t = \{y^h(t_{n-1}^-), \mathbf{0}^t\}. \quad (34)$$

The elements of  $\bar{A}$  are given by

$$\bar{A}_{ij} = \begin{cases} 1 + \bar{\lambda}, & \text{if } i = j = 1, \\ -\frac{\bar{\lambda}}{j(j-2)!}, & \text{if } i = 1, j > 1, \\ \frac{\bar{\lambda}}{i!}, & \text{if } i > 1, j = 1, \\ \frac{1}{(i-1)!(j-2)!} \left( \frac{1}{(i-1) + (j-1)} - \frac{\bar{\lambda}}{i(i+(j-1))} \right), & \text{otherwise.} \end{cases} \quad (35)$$

For the  $P1$  algorithm,

$$\bar{A} = \begin{bmatrix} 1 + \bar{\lambda} & -\frac{1}{2}\bar{\lambda} \\ \frac{1}{2}\bar{\lambda} & \frac{1}{2} - \frac{1}{6}\bar{\lambda} \end{bmatrix}. \quad (36)$$

The amplification factor is then determined by employing Cramer's rule. The denominator of  $A$  is given by  $\det \bar{A}$  while the numerator is computed from  $\det \tilde{A}$  where  $\tilde{A}$  is obtained by replacing the first column of  $\bar{A}$  by  $R_n$ . The resultant stability polynomial is identical to the upper diagonal of the Padé approximation table in which the numerator,  $P$ , and the denominator,  $Q$ , may be expressed as [8]

$$P = 1 + \sum_{i=1}^k \left\{ \prod_{j=0}^{i-1} \left( \frac{k-j}{2k+1-j} \right) \right\} \frac{(-\bar{\lambda})^i}{i!}, \quad (37)$$

$$Q = 1 + \sum_{i=1}^{k+1} \left\{ \prod_{j=0}^{i-1} \left( \frac{k+1-j}{2k+1-j} \right) \right\} \frac{\bar{\lambda}^i}{i!}. \quad (38)$$

Since the power on  $\bar{\lambda}$  is one order higher for  $Q$  than for  $P$ ,

$$\lim_{|\bar{\lambda}| \rightarrow \infty} A = \lim_{|\bar{\lambda}| \rightarrow \infty} \frac{P}{Q} = 0. \quad (39)$$

Thus, the TDG method is asymptotically annihilating. For example, for the  $P1$  algorithm,

$$A = \frac{1 - \frac{1}{3}\bar{\lambda}}{1 + \frac{2}{3}\bar{\lambda} + \frac{1}{6}(\bar{\lambda})^2}. \quad (40)$$

**REMARK.** The TDG method generates optimal asymptotic annihilation schemes in terms of balancing computational expense and accuracy. Padé approximations are rational functions (of  $\bar{\lambda}$ ) such that for a given degree of numerator and denominator the resultant approximation has the highest order of accuracy [7] (the order of accuracy is  $j+k$ , where  $j$  and  $k$  are the degrees of the numerator and denominator, respectively). All Padé methods of type  $(j, k)$ , with  $j < k$ , correspond to asymptotic annihilation methods. Because computational cost increases as the degree increases, the upper diagonal of the Padé approximation table represents an optimal balance between computational cost, accuracy and attaining asymptotic annihilation.

#### 4. Comparison of asymptotic annihilation algorithms

In this section, we compare the spectral radii and numerical dissipation and dispersion characteristics of previously developed asymptotic annihilation algorithms and several of the TDG-based algorithms. These analytical measures for the TDG algorithm are obtained by working directly with (20) and the single degree-of-freedom implementation of the undamped, unforced form of (12) rather than (21). Following the approach used to derive (19), the values at the end of a time interval, e.g.,  $u^h(t_n^-)$  and  $v^h(t_n^-)$ , may be expressed in terms of the values at the end of the previous time interval by

$$\begin{bmatrix} u^h(t_n^-) \\ \Delta t v^h(t_n^-) \end{bmatrix} = \mathbf{A} \begin{bmatrix} u^h(t_{n-1}^-) \\ \Delta t v^h(t_{n-1}^-) \end{bmatrix}, \quad (41)$$

where  $\mathbf{A}$  is the amplification matrix of dimension  $2 \times 2$ . For example, for the  $P1$  method,

$$\mathbf{A} = D^{-1} \begin{bmatrix} 1 - \frac{7}{18} \Omega^2 & 1 - \frac{1}{18} \Omega^2 \\ -\Omega^2(1 - \frac{1}{18} \Omega^2) & 1 - \frac{7}{18} \Omega^2 \end{bmatrix} \quad (42)$$

in which

$$D = 1 + \frac{1}{9} \Omega^2 + \frac{1}{36} \Omega^4 \quad (43)$$

and where  $\Omega = \Delta t \omega$ .

The spectral radius,  $\rho(\mathbf{A})$ , is defined by

$$\rho(\mathbf{A}) = \max |\lambda_1(\mathbf{A}), \lambda_2(\mathbf{A})|, \quad (44)$$

where  $\lambda_i(\mathbf{A})$  denotes the  $i$ th eigenvalue of  $\mathbf{A}$ .

Provided that the eigenvalues of  $\mathbf{A}$  remain complex ( $\lambda_{1,2}(\mathbf{A}) = A \pm iB$ ,  $B \neq 0$ ),

$$u^h(t_n^-) = e^{-\bar{\xi} \bar{\omega} t_n} (c_1 \cos(\bar{\omega} t_n) + c_2 \sin(\bar{\omega} t_n)), \quad (45)$$

in which

$$\bar{\omega} = \Delta t^{-1} \tan^{-1} \left( \frac{B}{A} \right), \quad (46)$$

$$\bar{\xi} = -\frac{\ln \rho}{\Delta t \bar{\omega}}, \quad (47)$$

where  $\bar{\xi}$  is the algorithmic damping ratio,  $\bar{\omega}$  is the approximate frequency and the coefficients  $c_1$  and  $c_2$  are determined from the initial conditions. The algorithmic damping ratio provides a measure of the numerical dissipation while the relative frequency error, calculated using

$$\frac{\omega}{\bar{\omega}} - 1, \quad (48)$$

is a measure of numerical dispersion. The relative frequency error may also be expressed as

$$\frac{\Omega}{\bar{\Omega}} - 1, \quad (49)$$

where  $\Omega = \Delta t \omega$  and  $\bar{\Omega} = \Delta t \bar{\omega}$ . For details on the derivation of the above analytical measures, see [1, 9].

Spectral radii are shown in Fig. 2 for the  $P0$ ,  $P1$ ,  $P2$  and  $P3$  TDG methods as well as the Houbolt

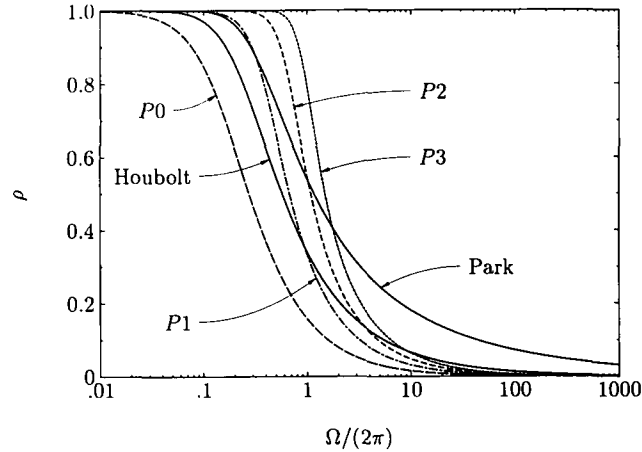


Fig. 2. Comparison of spectral radii for asymptotic annihilation algorithms.

and Park methods. All algorithms achieve asymptotic annihilation; the smaller values of spectral radius for the Houbolt and  $P0$  methods in the low frequency region reflect the overly dissipative properties of these methods. This may be seen more clearly in Fig. 3 in which the numerical damping ratio is shown. In the region of low-frequency accuracy ( $\Omega < \pi/5$ ), the  $P1$  and Park algorithms have nearly identical numerical dissipation. The  $P3$  algorithm has virtually no numerical dissipation in the low-frequency regime. Figure 4 depicts the frequency errors of the various algorithms. The third-order accuracy of the

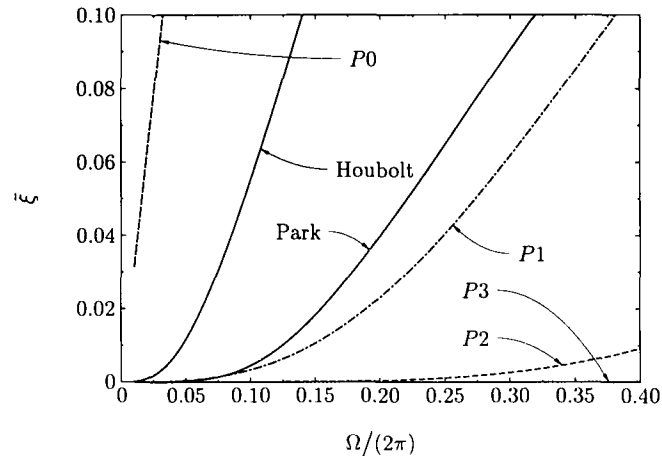


Fig. 3. Comparison of algorithmic damping for asymptotic annihilation algorithms.

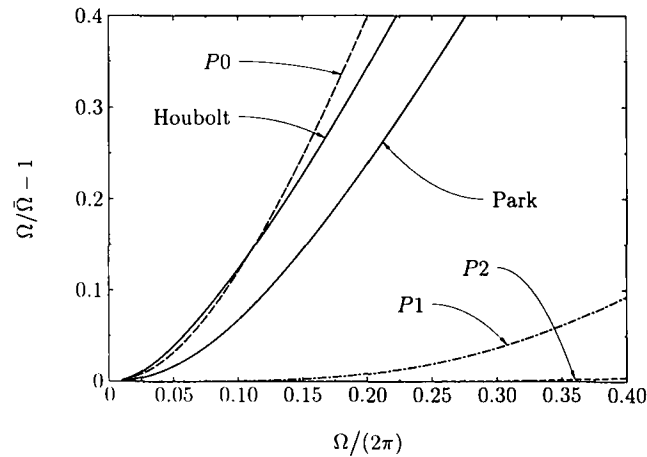


Fig. 4. Comparison of frequency error for asymptotic annihilation algorithms.



$P1$  method is evident in this plot (the Houbolt and Park algorithms are second-order accurate); the frequency error of the  $P2$  algorithm is virtually negligible in the low-frequency regime which reflects its fifth-order accuracy. The  $P3$  method has been omitted from the plot as its numerical dispersion is nearly zero for the given range of  $\Omega$ .

The spectral properties of an algorithm depict long-term stability behavior. However, large amplitude oscillations in the first time step of a computation have been observed in algorithms that have good spectral properties. This anomalous behaviour is referred to as overshoot [11]. None of the TDG algorithms presented exhibits overshoot since the TDG algorithms do not require calculating an initial acceleration vector.

## 5. Conclusions

A class of time integration algorithms for structural dynamics has been presented that asymptotically annihilates spurious high-frequency behavior without introducing excessive dissipation in the low-frequency regime. These algorithms are based upon the time-discontinuous Galerkin method in which equal-order interpolation is employed for displacement and velocity. It was shown that the algorithm stability polynomial is identical to the upper diagonal of the Padé approximation table; thus, the TDG method achieves the asymptotic annihilation property with a nearly optimal balance of computational expense and accuracy. Results from analyses of the algorithms obtained by several interpolation choices were compared with commonly used asymptotic annihilation algorithms. These results demonstrate the improved accuracy characteristics of the TDG method.

Because the higher-order accurate TDG methods belong to the class of high-order implicit methods, they have the disadvantage of increased computational expense due to the larger system of coupled equations to be solved. One approach towards developing a useful family of algorithms is to recast the algorithms as a hierarchy of methods in which the matrices and solution variables from the lower-order schemes may be employed without modification in the higher-order schemes. This strategy can be coupled with reducing algorithmic expense by casting the methods in predictor-multicorrector form. Predictor-multicorrector algorithms based upon time finite element methods were presented in [10]; however, the emphasis was not on retaining asymptotic annihilation characteristics. Further study is needed to extend the predictor-multicorrector approach to the algorithms presented in this paper to develop a general class of computationally efficient, high-order accurate asymptotic annihilation methods for structural dynamics.

## References

- [1] T.J.R. Hughes, *The Finite Element Method: Linear Static and Dynamic Finite Element Analysis* (Prentice-Hall, Englewood Cliffs, NJ, 1987).
- [2] A. Cardona and M. Geradin, Time integration of the equations of motion in mechanism analysis, *Comput. & Structures* 33 (1989) 801–820.
- [3] J.C. Houbolt, A recurrence matrix solution for the dynamic response of elastic aircraft, *J. Aeronautical Sci.* 17 (1950) 540–550.
- [4] G.M. Hulbert, Limitations on linear multistep methods for structural dynamics, *Earthquake Engrg. Structural Dynam.* 20 (1991) 191–196.
- [5] K.C. Park, Evaluating time integration methods for nonlinear dynamic analysis, in T. Belytschko et al., *Finite Element Analysis of Transient Nonlinear Behavior*, AMD-14 (ASME, New York, 1975) 35–58.
- [6] G.M. Hulbert, Time finite element methods for structural dynamics, *Internat. J. Numer. Methods Engrg.* 33 (1992) 307–331.
- [7] W.B. Gragg, The Padé table and its relation to certain algorithms of numerical analysis, *SIAM Rev.* 14 (1972) 1–62.
- [8] E. Hairer and G. Wanner, *Solving Ordinary Differential Equations II: Stiff and Differential-Algebraic Problems* (Springer-Verlag, New York, 1991).
- [9] H.M. Hilber, Analysis and design of numerical integration methods in structural dynamics, Ph.D. Thesis, Earthquake Engineering Research Center, University of California, Berkeley, 1976.
- [10] G.M. Hulbert, Space-time finite element methods for second-order hyperbolic equations, Ph.D. Thesis, Department of Mechanical Engineering, Stanford University, Stanford, 1989.
- [11] H.M. Hilber and T.J.R. Hughes, Collocation, dissipation and ‘overshoot’ for time integration schemes in structural dynamics, *Earthquake Engrg. Structural Dynam.* 6 (1978) 99–118.