AIAA Journal
Vol. 39, No. 12, December 2001

Precise Time-Integration Method with Dimensional Expanding for Structural Dynamic Equations

Yuanxian Gu,* Biaosong Chen,† Hongwu Zhang,‡ and Zhenqun Guan§
Dalian University of Technology, 116024 Dalian, People's Republic of China

The precise time-integration method proposed for a linear time-invariant dynamic system can give precise numerical results approaching the exact solution at the integration points. However, difficulties arise when the algorithm is used for nonhomogeneous dynamic systems due to the inverse matrix calculation required. A new algorithm is proposed to convert nonhomogeneous dynamic equations into homogeneous equations by means of the dimensional expanding method. With this conversion, the inverse matrix calculation is not required in the precise time-integration method. The new algorithm has enhanced the precise time-integration method by benefiting the programming implementation and the numerical stability; it has improved the computational efficiency as well. Numerical examples are given to demonstrate the validity and efficiency of the algorithm.

Nomenclature

$\boldsymbol{A}_i$	=	coefficient of the solution of Fourier series
$\boldsymbol{a}_i$	=	coefficient vector
$\boldsymbol{B}_i$	=	coefficient of the solution of Fourier series
$\boldsymbol{b}_i$	=	coefficient vector
$\boldsymbol{C}$	=	magnitude of the load on the nodes of the structure
$\boldsymbol{D}$	=	coefficient matrix of ordinary differential equations satisfied by the nonhomogeneous vector
d	=	number of terms of one series
$\boldsymbol{f}$	=	load vector
$\boldsymbol{G}$	=	damping matrix
$\boldsymbol{H}$	=	coefficient matrix of structural dynamic system
$\boldsymbol{H}^*$	=	coefficient matrix of expanding structural dynamic system
$\boldsymbol{I}$	=	identity matrix
K	=	number of time steps
$\boldsymbol{K}$	=	stiffness matrix
k	=	order of time step
l	=	number of dynamic modes
$\boldsymbol{M}$	=	mass matrix
m	=	number of 2^N
N	=	algorithm parameter of precise time integration
n	=	dimension of the structural dynamic system
$\boldsymbol{p}$	=	transformed system state variable
$\boldsymbol{q}$	=	displacement vector
$\boldsymbol{r}$	=	nonhomogeneous vector of structural dynamic system
$\boldsymbol{T}_a$	=	small part of the matrix exponential
$\boldsymbol{T}_c$	=	part of matrix exponential according to matrix $\boldsymbol{C}$
$\boldsymbol{T}_d$	=	part of matrix exponential according to matrix $\boldsymbol{D}$
t	=	time
$\boldsymbol{v}$	=	state variable of structure dynamic system
$\boldsymbol{v}^*$	=	state variable vector of expanding structural dynamic system
$\boldsymbol{x}$	=	displacement vector
Δt	=	time interval dependent on τ and N
τ	=	time step length
ω	=	angular rate of Fourier series

Received 18 August 2000; revision received 11 May 2001; accepted for publication 14 May 2001.

*Professor, State Key Laboratory of Structural Analysis for Industrial Equipment, Department of Engineering Mechanics; guyx96@dlut.edu.cn.

†Ph.D. Student, State Key Laboratory of Structural Analysis for Industrial Equipment, Department of Engineering Mechanics.

‡Professor, State Key Laboratory of Structural Analysis for Industrial Equipment, Department of Engineering Mechanics.

§Associate Professor, State Key Laboratory of Structural Analysis for Industrial Equipment, Department of Engineering Mechanics.

Introduction

STRUCTURAL dynamics is very important in practice and has attracted much attention during recent decades. One of the most popular methods of solving dynamic equations is direct time integration.[1,2] Many types of direct time-integration methods are based on finite difference in time, such as the Wilson-θ, the Newmark, etc. The numerical stability and precision of these methods are key problems to be concerned about.

Structural dynamic equations are second-order ordinary differential equations (ODEs). They can be transformed to first-order ODEs by mathematical manipulation so that some common algorithms used in control theory or systems engineering can be applied. In these algorithms, the computation of matrix exponential is important. Moler and Van Loan[3] listed 19 ways to evaluate a matrix exponential. Each of these ways has some advantages and shortcomings, and so it is difficult to tell which one is the best.

Zhong[4] and Zhong and Williams[5] proposed a precise time-integration (PTI) method for structural dynamics. The key component of the PTI method is computing the matrix exponential. It can give very precise results, and then the direct time-integration method is constructed based on this precise computation. This algorithm is unconditionally stable,[6,7] and its precision is commonly independent of time step length. Such high precision gives the PTI method many beneficial numerical properties such as zero rate of period elongation and zero rate of amplitude decay.

However, there are two difficulties for the implementation of the PTI method for structural dynamics. One is that the computational expense and storage requirement is high. The other is that, as for nonhomogeneous equations, it requires the computation of the inverse matrix. The first can be overcome with the method introduced by Fung,[7] which utilizes the symmetry and narrow bandwidth of the matrix exponential. It has the potential to improve the efficiency greatly. In this paper, the PTI method with dimensional expanding is presented. This scheme transforms the nonhomogeneous equations to homogeneous equations by means of the dimensional expanding method. Therefore, it does not need to compute the inverse matrix. The new algorithm not only benefits the programming implementation and the numerical stability but also improves the computational efficiency.

PTI Method

To give a complete statement of our method, the original PTI method proposed by Zhong[4] and Zhong and Williams[5] is introduced briefly. The equations of motion for structural dynamics in discrete form with n dimension can be written as

$$\boldsymbol{M}\ddot{\boldsymbol{x}} + \boldsymbol{G}\dot{\boldsymbol{x}} + \boldsymbol{K}\boldsymbol{x} = \boldsymbol{f}(t) \tag{1}$$

where the initial conditions $\boldsymbol{x}(0)$ and $\dot{\boldsymbol{x}}(0)$ are given. $\boldsymbol{M}$, $\boldsymbol{G}$, and $\boldsymbol{K}$ are time-invariant matrices with $n \times n$ dimensions, respectively. If $\boldsymbol{f}(t) = \{0\}$, then Eqs. (1) are homogeneous equations, otherwise they are nonhomogeneous. Let $\boldsymbol{p} = \boldsymbol{M}\dot{\boldsymbol{x}} + \boldsymbol{G}\boldsymbol{x}/2$ and $\boldsymbol{q} = \boldsymbol{x}$; Eq. (1) becomes

$$\dot{\boldsymbol{v}} = \boldsymbol{H}\boldsymbol{v} + \boldsymbol{r} \tag{2a}$$

$$\boldsymbol{v} = \begin{Bmatrix} \boldsymbol{q} \\ \boldsymbol{p} \end{Bmatrix}, \qquad \boldsymbol{H} = \begin{bmatrix} -\boldsymbol{M}^{-1}\boldsymbol{G}/2, & \boldsymbol{M}^{-1} \\ \boldsymbol{G}\boldsymbol{M}^{-1}\boldsymbol{G}/4 - \boldsymbol{K}, & -\boldsymbol{G}\boldsymbol{M}^{-1}/2 \end{bmatrix}$$

$$\boldsymbol{r} = \begin{Bmatrix} \boldsymbol{0} \\ \boldsymbol{f} \end{Bmatrix} \tag{2b}$$

where $\boldsymbol{H}$ is a matrix with $2n \times 2n$ dimensions. This transformation is not the only one that can be adopted.[7] With other transformation methods, the results are almost identical and do not affect the algorithm. The general solution $\boldsymbol{v}$ of Eq. (2) consists of two parts, that is, $\boldsymbol{v}_1$ and $\boldsymbol{v}_2$, due to the nonhomogeneous term $\boldsymbol{r}(t)$:

$$\boldsymbol{v}(t) = \boldsymbol{v}_1(t) + \boldsymbol{v}_2(t) \tag{3a}$$

$$\boldsymbol{v}_1(t) = \exp(\boldsymbol{H}t) \cdot \boldsymbol{v}_0 \tag{3b}$$

$$\boldsymbol{v}_2 = \int_0^t \exp[\boldsymbol{H}(t - s)] \cdot \boldsymbol{r}(s)\,\mathrm{d}s \tag{3c}$$

where $\boldsymbol{v}_0$ are the initial conditions. Then the solution of the system at time t_k is

$$\boldsymbol{v}_k = \exp(\boldsymbol{H}t_k)\boldsymbol{v}_0 + \int_0^{t_k} \exp[\boldsymbol{H}(t_k - s)] \cdot \boldsymbol{r}(s)\,\mathrm{d}s \tag{4}$$

According to Eq. (4), the solution at time t_{k+1} $(\tau = t_{k+1} - t_k)$ is

$$\boldsymbol{v}_{k+1} = \exp(\boldsymbol{H}t_{k+1})\boldsymbol{v}_0 + \int_0^{t_{k+1}} \exp[\boldsymbol{H}(t_{k+1} - s)] \cdot \boldsymbol{r}(s)\,\mathrm{d}s \tag{5}$$

When the relationship between $\boldsymbol{v}_k$ and $\boldsymbol{v}_{k+1}$ is deduced, the integration scheme can be obtained:

$$\begin{aligned}
\boldsymbol{v}_{k+1} &= \exp(\boldsymbol{H}t_{k+1})\boldsymbol{v}_0 + \int_0^{t_{k+1}} \exp[\boldsymbol{H}(t_{k+1} - s)] \cdot \boldsymbol{r}(s)\,\mathrm{d}s \\
&= \exp(\boldsymbol{H}t_{k+1})\boldsymbol{v}_0 + \int_0^{t_k} \exp[\boldsymbol{H}(t_{k+1} - s)] \cdot \boldsymbol{r}(s)\,\mathrm{d}s \\
&\quad + \int_{t_k}^{t_{k+1}} \exp[\boldsymbol{H}(t_{k+1} - s)] \cdot \boldsymbol{r}(s)\,\mathrm{d}s \\
&= \exp(\boldsymbol{H}\tau)\left\{ \exp(\boldsymbol{H}t_k)\boldsymbol{v}_0 + \int_0^{t_k} \exp[\boldsymbol{H}(t_k - s)] \cdot \boldsymbol{r}(s)\,\mathrm{d}s \right\} \\
&\quad + \int_0^{\tau} \exp[\boldsymbol{H}(\tau - s)] \cdot \boldsymbol{r}(t_k + s)\,\mathrm{d}s \\
&= \exp(\boldsymbol{H}\tau)\boldsymbol{v}_k + \int_0^{\tau} \exp[\boldsymbol{H}(\tau - s)] \cdot \boldsymbol{r}(t_k + s)\,\mathrm{d}s
\end{aligned} \tag{6}$$

The preceding formula is the integration scheme of the PTI method. If the initial conditions are known, the unknown vector $\boldsymbol{v}$ at the next time can be evaluated, and step by step the vectors $\boldsymbol{v}$ at all times can be obtained. The important characteristic of the PTI method is evaluating the matrix exponential. Let

$$\boldsymbol{T} = \exp(\boldsymbol{H} \cdot \tau) \tag{7}$$

To compute the matrix exponential $\boldsymbol{T}$ accurately, the 2^N algorithm can be used. Let $N = 20$, $m = 2^N = 1{,}048{,}576$, then $\Delta t = \tau/m$ is an extremely small time interval. By the use of the superposition of the exponential function,

$$\begin{aligned}
\boldsymbol{T} &= \exp(\boldsymbol{H}\tau) = [\exp(\boldsymbol{H} \cdot \tau/m)]^m = [\exp(\boldsymbol{H} \cdot \tau/2^N)]^{2^N} \\
&= [\exp(\boldsymbol{H} \cdot \Delta t)]^{2^N}
\end{aligned} \tag{8a}$$

Employing a second-order Taylor expansion,

$$\exp(\boldsymbol{H} \cdot \Delta t) \approx \boldsymbol{I} + \boldsymbol{H} \cdot \Delta t + (\boldsymbol{H} \cdot \Delta t)^2/2 = \boldsymbol{I} + \boldsymbol{T}_a \tag{8b}$$

then

$$\boldsymbol{T} \approx [\boldsymbol{I} + \boldsymbol{T}_a]^{2^N} \tag{8c}$$

Thus, $\boldsymbol{T}$ can be evaluated after N times the matrix multiplication of Eq. (8c) by using the product notation

$$\boldsymbol{T} \approx \prod_{i=1}^{N} \left([\boldsymbol{I} + \boldsymbol{T}_a]^2\right) \tag{9}$$

In the preceding formula, higher-order Taylor expansion is recommended. Note that Δt is very small, and so is $\boldsymbol{T}_a$ compared to the identity matrix $\boldsymbol{I}$. The direct addition of $\boldsymbol{T}_a$ and $\boldsymbol{I}$ will cause a great loss of significant digits. To avoid this, the algorithm executes the addition of small matrix $\boldsymbol{T}_a$ first and then adds it to identity matrix $\boldsymbol{I}$. When the relation

$$(\boldsymbol{I} + \boldsymbol{T}_a)^2 = \boldsymbol{I} + 2\boldsymbol{T}_a + \boldsymbol{T}_a \times \boldsymbol{T}_a \tag{10}$$

is taken into account, the algorithm is

$$\text{for (iter} = 0;\ \text{iter} < N;\ \text{iter}{+}{+})\boldsymbol{T}_a = 2\boldsymbol{T}_a + \boldsymbol{T}_a \times \boldsymbol{T}_a \tag{11}$$

When the loop is over, the addition can be done:

$$\boldsymbol{T} = \boldsymbol{I} + \boldsymbol{T}_a \tag{12}$$

Equations (11) and (12) are the key steps to compute the matrix exponential in the PTI method. If the nonhomogeneous vector is linear in the time interval (t_k, t_{k+1}), $\boldsymbol{v}_{k+1}$ can be expressed as

$$\boldsymbol{r}(t) = \boldsymbol{r}_0 + \boldsymbol{r}_1(t - t_k)$$

$$\boldsymbol{v}_{k+1} = \boldsymbol{T}\left[\boldsymbol{v}_k + \boldsymbol{H}^{-1}\left(\boldsymbol{r}_0 + \boldsymbol{H}^{-1}\boldsymbol{r}_1\right)\right] - \boldsymbol{H}^{-1}\left[\boldsymbol{r}_0 + \boldsymbol{H}^{-1}\boldsymbol{r}_1 + \boldsymbol{r}_1\tau\right] \tag{13}$$

Lin et al.[6] proposed a computing method for nonhomogeneous vectors in sinusoidal or Fourier series forms. If $\boldsymbol{r}(t)$ can be expressed as

$$\boldsymbol{r}(t) = \boldsymbol{b}_0 + \sum_{i=1}^{d} [\boldsymbol{a}_i \sin(i\omega t) + \boldsymbol{b}_i \cos(i\omega t)] \tag{14}$$

then, by the PTI method,

$$\begin{aligned}
\boldsymbol{v}_{k+1} &= \boldsymbol{T}\left\{ \boldsymbol{v}_k - \boldsymbol{B}_0 - \sum_{i=1}^{d} [\boldsymbol{A}_i \sin(i\omega t_k) + \boldsymbol{B}_i \cos(i\omega t_k)] \right\} \\
&\quad + \boldsymbol{B}_0 + \sum_{i=1}^{d} [\boldsymbol{A}_i \sin(i\omega t_{k+1}) + \boldsymbol{B}_i \cos(i\omega t_{k+1})]
\end{aligned} \tag{15a}$$

where

$$\boldsymbol{B}_0 = -\boldsymbol{H}^{-1}\boldsymbol{b}_0 \tag{15b}$$

$$\boldsymbol{A}_i = (i^2\omega^2\boldsymbol{I} + \boldsymbol{H}^2)^{-1}(-\boldsymbol{H}\boldsymbol{a}_i + i\omega\boldsymbol{b}_i)$$

$$\boldsymbol{B}_i = (i^2\omega^2\boldsymbol{I} + \boldsymbol{H}^2)^{-1}(-i\omega\boldsymbol{a}_i - \boldsymbol{H}\boldsymbol{b}_i) \qquad (i = 1, 2, \ldots, d) \tag{15c}$$

Calculations of the inverse matrix are required in the original PTI method just presented. This makes the applications of the PTI method difficult.

PTI with Dimensional Expanding

Basic Formulations

Computing the inverse matrix is required due to the existence of the nonhomogeneous terms $\boldsymbol{f}(t)$. To eliminate the inverse matrix calculation, the nonhomogeneous equations need to be transformed into homogeneous equations. If the nonhomogeneous vector is a solution to the ODEs of constant coefficients with n dimensions, then $\boldsymbol{r}$ satisfies

$$\dot{\boldsymbol{r}} = \boldsymbol{D}\cdot\boldsymbol{r} \tag{16a}$$

where

$$\dot{\boldsymbol{r}} = \dot{\boldsymbol{f}}, \qquad \boldsymbol{r} = \boldsymbol{f} \tag{16b}$$

The equations associated with Eq. (2) form homogeneous ODEs whose dimension is larger than the original nonhomogeneous ODEs. The homogeneous ODEs are mathematically equivalent to Eq. (2). Let

$$\dot{\boldsymbol{v}}^* = \begin{Bmatrix}\dot{\boldsymbol{q}}\\ \dot{\boldsymbol{p}}\\ \dot{\boldsymbol{r}}\end{Bmatrix} = \boldsymbol{H}^*\boldsymbol{v}^* = \boldsymbol{H}^*\cdot\begin{Bmatrix}\boldsymbol{v}\\ \boldsymbol{r}\end{Bmatrix} = \boldsymbol{H}^*\begin{Bmatrix}\boldsymbol{q}\\ \boldsymbol{p}\\ \boldsymbol{r}\end{Bmatrix} \tag{17a}$$

$$\boldsymbol{v} = \begin{Bmatrix}\boldsymbol{q}\\ \boldsymbol{p}\end{Bmatrix}, \qquad \boldsymbol{H}^* = \begin{bmatrix}\boldsymbol{H} & \boldsymbol{C}\\ \boldsymbol{0} & \boldsymbol{D}\end{bmatrix} \tag{17b}$$

Then the system equation has the form

$$\dot{\boldsymbol{v}}^* = \begin{Bmatrix}\dot{\boldsymbol{v}}\\ \dot{\boldsymbol{r}}\end{Bmatrix} = \begin{bmatrix}\boldsymbol{H} & \boldsymbol{C}\\ \boldsymbol{0} & \boldsymbol{D}\end{bmatrix}\begin{Bmatrix}\boldsymbol{v}\\ \boldsymbol{r}\end{Bmatrix} \tag{17c}$$

where $\boldsymbol{H}^*$ is a $3n\times 3n$ matrix and $\boldsymbol{H}$, $\boldsymbol{C}$, and $\boldsymbol{D}$ are matrices with $2n\times 2n$, $2n\times n$, and $n\times n$ dimensions, respectively. $\boldsymbol{H}$ is the same as before, $\boldsymbol{C}$ represents the relationship between the loads and nodes, and $\boldsymbol{D}$ is dependent on the nonhomogeneous vector. When this method is used, the nonhomogeneous vector is dealt with as a state variable of the system, and the nonhomogeneous equations are transformed into homogeneous equations with dimensional expanding. This new method is called the PTI method with dimensional expanding. The next step is computing the matrix exponential

$$\boldsymbol{T}^* = \exp(\boldsymbol{H}^*\cdot\tau) \tag{18}$$

The solution of time integration with the PTI method is

$$\boldsymbol{v}^*_{k+1} = \boldsymbol{T}^*\boldsymbol{v}^*_k \tag{19}$$

Compared to Eqs. (13) and (15), the new method is easier than the original PTI method and does not require the inverse matrix calculations. This advantage is particularly beneficial to programming implementation and to the system whose $\boldsymbol{H}$ is singular or almost singular. However, the computation expense of the matrix $\boldsymbol{T}^*$ with dimensional expanding is noticeable. Therefore, some approaches to be introduced can be used to improve the computational efficiency.

Constant Approximation of the Nonhomogeneous Vector

If the nonhomogeneous term $\boldsymbol{r}(t)$ varies smoothly with time, it can be assumed that $\boldsymbol{r}(t)$ is constant within the time interval (t_k, t_{k+1}). With this assumption,

$$\boldsymbol{r}_k = \boldsymbol{f}_k, \qquad \boldsymbol{f}_k = [\boldsymbol{f}(t_k) + \boldsymbol{f}(t_{k+1})]/2 \tag{20}$$

The constant coefficient matrix of the ODEs that are satisfied by $\boldsymbol{r}(t)$ is $\boldsymbol{D} = [0]$. Following the implementation of Taylor expansion in Eq. (8),

$$\boldsymbol{T}^*_a = \begin{bmatrix}\boldsymbol{T}_a & \boldsymbol{T}_c\\ 0 & \boldsymbol{T}_d\end{bmatrix} = \begin{bmatrix}\boldsymbol{H}\cdot\Delta t + \boldsymbol{H}\cdot\boldsymbol{H}\cdot(\Delta t)^2/2 & \boldsymbol{C}\cdot\Delta t + \boldsymbol{H}\cdot\boldsymbol{C}\cdot(\Delta t)^2/2\\ 0 & 0\end{bmatrix} \tag{21}$$

in which $\boldsymbol{T}_a$ is the same as before and $\boldsymbol{T}_c$ is a $2n\times n$ matrix:

$$\boldsymbol{T}_c = \boldsymbol{C}\cdot\Delta t + \boldsymbol{H}\cdot\boldsymbol{C}\cdot(\Delta t)^2/2, \qquad \boldsymbol{T}_d = [0] \tag{22}$$

When the loop from Eq. (11) is followed and Eq. (12) is executed,

$$\boldsymbol{T}^*_a = 2\boldsymbol{T}^*_a + \boldsymbol{T}^*_a\times\boldsymbol{T}^*_a = 2\begin{bmatrix}\boldsymbol{T}_a & \boldsymbol{T}_c\\ 0 & 0\end{bmatrix} + \begin{bmatrix}\boldsymbol{T}_a & \boldsymbol{T}_c\\ 0 & 0\end{bmatrix}\times\begin{bmatrix}\boldsymbol{T}_a & \boldsymbol{T}_c\\ 0 & 0\end{bmatrix} = \begin{bmatrix}2\boldsymbol{T}_a + \boldsymbol{T}_a\times\boldsymbol{T}_a & 2\boldsymbol{T}_c + \boldsymbol{T}_a\times\boldsymbol{T}_c\\ 0 & 0\end{bmatrix} \tag{23a}$$

$$\boldsymbol{T}^* = \begin{bmatrix}\boldsymbol{T} & \boldsymbol{T}_c\\ \boldsymbol{0} & \boldsymbol{I}\end{bmatrix} = \begin{bmatrix}\boldsymbol{I} + \boldsymbol{T}_a & \boldsymbol{T}_c\\ \boldsymbol{0} & \boldsymbol{I}\end{bmatrix} \tag{23b}$$

The integration formula is

$$\boldsymbol{v}_{k+1} = \boldsymbol{T}\boldsymbol{v}_k + \boldsymbol{T}_c\boldsymbol{r}_k \tag{24}$$

Linear Approximation of the Nonhomogeneous Vector

Assume that the nonhomogeneous vector varies linearly within the time interval (t_k, t_{k+1}); then

$$\boldsymbol{r}_k = \begin{Bmatrix}\boldsymbol{f}_k\\ \dot{\boldsymbol{f}}_k\end{Bmatrix}, \qquad \boldsymbol{f}_k = \boldsymbol{f}(t_k), \qquad \dot{\boldsymbol{f}}_k = \frac{\boldsymbol{f}(t_{k+1}) - \boldsymbol{f}(t_k)}{\tau} \tag{25}$$

This way, $\boldsymbol{r}$ with $2n\times 1$ satisfies the following ODEs with $2n\times 2n$ dimensions:

$$\dot{\boldsymbol{r}} = \boldsymbol{D}\cdot\boldsymbol{r} = \begin{Bmatrix}\dot{\boldsymbol{f}}\\ \ddot{\boldsymbol{f}}\end{Bmatrix} = \begin{bmatrix}0 & \boldsymbol{I}\\ 0 & 0\end{bmatrix}\begin{Bmatrix}\boldsymbol{f}\\ \dot{\boldsymbol{f}}\end{Bmatrix}, \qquad \boldsymbol{D} = \begin{bmatrix}0 & \boldsymbol{I}\\ 0 & 0\end{bmatrix} \tag{26}$$

where matrix $\boldsymbol{D}$ has $2n\times 2n$ dimensions and matrix $\boldsymbol{C}$ has $2n\times 2n$ dimensions. Because $\boldsymbol{D}\cdot\boldsymbol{D} = [0]$, following the implementation of Taylor expansion as in Eq. (8) yields the result

$$\boldsymbol{T}_c = \boldsymbol{C}\cdot\Delta t + \boldsymbol{H}\cdot\boldsymbol{C}(\Delta t)^2/2 + \boldsymbol{C}\cdot\boldsymbol{D}(\Delta t)^2/2, \qquad \boldsymbol{T}_d = \boldsymbol{D}\cdot\Delta t \tag{27}$$

Here, the matrix $\boldsymbol{T}_a$ is the same as before. $\boldsymbol{T}_c$ is a $2n\times 2n$ matrix that is larger than that in the constant approximation method because of the existence of $\dot{\boldsymbol{f}}$. The calculation of $\boldsymbol{T}_d$ is very easy to implement because $\boldsymbol{D}$ is a simple matrix. When the loop from Eq. (11) is followed, $\boldsymbol{T}_a$ and $\boldsymbol{T}_c$ are the same as in Eq. (23). $\boldsymbol{T}_d$ is computed as

$$\boldsymbol{T}_d = 2\boldsymbol{T}_d \tag{28}$$

Compared to the constant approximation method, additional calculation required in the linear approximation method is computing matrix $\boldsymbol{T}_c$ with larger dimensions. When the loop in Eq. (11) is followed, the integration solution of the PTI method is

$$\boldsymbol{v}_{k+1} = \boldsymbol{T}\boldsymbol{v}_k + \boldsymbol{T}_c\boldsymbol{r}_k \tag{29}$$

Sinusoidal Approximation of the Nonhomogeneous Vector

If $r = a\sin(\omega t) + b\cos(\omega t)$, then r satisfies the equation

$$\begin{Bmatrix}\dot{r}\\ \ddot{r}\end{Bmatrix} = \begin{bmatrix}0 & 1\\ -\omega^2 & 0\end{bmatrix}\begin{Bmatrix}r\\ \dot{r}\end{Bmatrix} \tag{30}$$

with the initial conditions $r(0) = b$ and $\dot{r}(0) = a\omega$. According to this, if the nonhomogeneous vector can be expressed as a linear combination of several sinusoidal functions, such as

$$\boldsymbol{r}(t) = \sum_{i=1}^{d}(\boldsymbol{a}_i\sin\omega_i t + \boldsymbol{b}_i\cos\omega_i t) \tag{31}$$

then the coefficient matrix can be defined as

$$\boldsymbol{D} = \begin{bmatrix}0 & \boldsymbol{I}\\ \boldsymbol{D}_{21} & 0\end{bmatrix} \tag{32}$$

where $\boldsymbol{D}_{21}$ is diagonal matrix and its diagonal entries are the negative square of the frequencies of these sinusoidal functions. When the loop is followed from Eq. (11), $\boldsymbol{T}_a$ and $\boldsymbol{T}_c$ are the same as in Eq. (23). $\boldsymbol{T}_d$ can be computed as

$$\boldsymbol{T}_d = \boldsymbol{D} \cdot \Delta t + \boldsymbol{D} \cdot \boldsymbol{D} \cdot (\Delta t)^2/2 \tag{33}$$

$$\text{for (iter} = 0\text{; iter} < N\text{; iter++)}, \qquad \boldsymbol{T}_d = 2\boldsymbol{T}_d + \boldsymbol{T}_d \times \boldsymbol{T}_d \tag{34}$$

In which $\boldsymbol{T}_d$ is decomposed as

$$\boldsymbol{T}_d = \begin{bmatrix} \boldsymbol{T}_d^{11} & \boldsymbol{T}_d^{12} \\ \boldsymbol{T}_d^{21} & \boldsymbol{T}_d^{22} \end{bmatrix} \tag{35}$$

In the preceding formula, the submatrices are diagonal. This helps reduce the computational expense. Compared to the earlier two cases, the computation in this case is the most expensive. However, in practical engineering, loads usually vary in sinusoidal form, and this method can obtain more accurate numerical results.

Further Extensions

The preceding method can also be adapted to other forms of functions that satisfy constant coefficient ODEs, such as the polynomial functions

$$\boldsymbol{r}(t) = \sum_{i=0}^{d} \boldsymbol{a}_i t^i \tag{36}$$

$$\boldsymbol{r}(t) = \left(\sum_{i=0}^{d} \boldsymbol{a}_i t^i \right) e^{\alpha t} \tag{37}$$

$$\boldsymbol{r}(t) = \left(\sum_{i=0}^{d} \boldsymbol{a}_i t^i \right) [\alpha \sin(\omega t) + \beta \cos(\omega t)] \tag{38}$$

$$\boldsymbol{r}(t) = e^{\lambda t} \left[\left(\sum_{i=1}^{d} \boldsymbol{a}_i t^i \right) \sin(\omega t) + \left(\sum_{i=1}^{d} \boldsymbol{b}_i t^i \right) \cos(\omega t) \right] \tag{39}$$

If the preceding complex functions are taken into account, the only work required is to modify the matrix $\boldsymbol{D}$ as in Eq. (32). The others are the same as stated earlier. Therefore the PTI method presented in this paper has broad applications.

Computing Expense Analysis

When the condition that loads acting on a structure can be a linear combination of some basic functions satisfying homogeneous ODEs is utilized, the computational expense can be reduced even further. For example, if function f_i satisfies one ODE, or $\boldsymbol{f}_i$ can be transformed to satisfy other ODEs with l_i dimensions,

$$\dot{\boldsymbol{f}}_i = \boldsymbol{D}_i \boldsymbol{f}_i \tag{40}$$

The function $\boldsymbol{f}_i$ is called the ith dynamic mode with l_i dimensions. For example, a nodal load is expressed as

$$\boldsymbol{f}(t) = \sum_{i=1}^{d} a_i \boldsymbol{f}_i \tag{41}$$

where a_i is a constant coefficient. It represents the magnitude of each of the linear combinations of the dynamic modes. There are d dynamic modes, and these linear relationships are represented in matrix $\boldsymbol{C}$. When it is supposed that the sum of the modes dimensions

$$l = \sum_{i=1}^{d} l_i$$

then the dimension of matrix $\boldsymbol{C}$ is $2n \times l$ and the dimension of matrix $\boldsymbol{D}$ is $l \times l$. If $l < n$, the computational expense can be reduced. The computational expense of Eq. (29) should be compared with that of Eq. (13). In Eq. (13), the computing complexity of $\boldsymbol{H}^{-1}$ is $\mathcal{O}(8n^3)$, and that of $\boldsymbol{T}_c$ in Eq. (29) is $\mathcal{O}(2N \times n \times l^2)$, where N is the iteration number in Eq. (11). When it is supposed that the number of time steps is K and the computing complexity of $\boldsymbol{T}$ is taken into account, the total computing complexity of Eq. (13) is $\mathcal{O}[8(N+2) \times n^3 + 16n^2 \times K]$ and that of Eq. (29) is $\mathcal{O}[8(N+1) \times n^3 + 2N \times n^2 \times l + 2n \times K(l+2n)]$. If l is small and $l \times N < n$, then the Eq. (29) is more efficient than Eq. (13). This requirement is usually satisfied because in general the structural degrees of freedom n are larger than the number of dynamic modes l. For the same reason, the new method with dimensional expanding is also more efficient than the original method in Eq. (15). Because in Eq. (15) an inverse matrix calculation is needed for each frequency, however, the new method only needs to add a dynamic mode. In particular, when the structural degrees of freedom and the number of items of the Fourier series are very large, the advantage of the new method is more remarkable.

To summarize, the general algorithm of the new PTI method is outlined as follows:

$$\boldsymbol{T}_a = \boldsymbol{H} \cdot (\boldsymbol{I} \cdot \Delta t + \boldsymbol{H} \cdot \Delta t^2/2);$$

$$\boldsymbol{T}_c = \boldsymbol{C} \cdot \Delta t + \boldsymbol{H} \cdot \boldsymbol{C}(\Delta t)^2/2 + \boldsymbol{C} \cdot \boldsymbol{D} \cdot (\Delta t)^2/2;$$

$$\boldsymbol{T}_d = \boldsymbol{D} \cdot \Delta t + \boldsymbol{D} \cdot \boldsymbol{D} \cdot \Delta t^2/2;$$

for (iter = 0; iter $< N$; iter++) {

$$\boldsymbol{T}_c = 2\boldsymbol{T}_c + \boldsymbol{T}_a \cdot \boldsymbol{T}_c + \boldsymbol{T}_c \cdot \boldsymbol{T}_d;$$

$$\boldsymbol{T}_a = 2\boldsymbol{T}_a + \boldsymbol{T}_a \cdot \boldsymbol{T}_a;$$

$$\boldsymbol{T}_d = 2\boldsymbol{T}_d + \boldsymbol{T}_d \cdot \boldsymbol{T}_d;$$

}

$$\boldsymbol{T} = \boldsymbol{I} + \boldsymbol{T}_a;$$

Numerical Examples

Example 1

This example is used to show the applicability of the new PTI algorithm by solving a one-dimensional ODE. The equation of the problem is

$$y'(x) = -y + f(x)$$

with initial conditions $y(0) = 1.0$; $f(x)$ is the nonhomogeneous function, which can take the following forms.

For form a, the $f(x) = e^{-x}$, the analytical solution is $y(x) = e^{-x}(1+x)$, where $f(x)$ satisfies the differential equation $f'(x) = -f$, with initial condition $f(0) = 1.0$. To use the new PTI method, the equation can be transformed as

$$\begin{Bmatrix} y' \\ f' \end{Bmatrix} = \begin{bmatrix} -1 & 1 \\ 0 & -1 \end{bmatrix} \begin{Bmatrix} y \\ f \end{Bmatrix}$$

For form b, $f(x) = (1+x)e^{-x}$, the analytical solution is $y = \frac{1}{2} e^{-x}(2 + 2x + x^2)$, where $f(x)$ satisfies the differential equation with initial conditions $f(0) = 1.0$ and $f'(0) = 0.0$:

$$\begin{Bmatrix} f' \\ f'' \end{Bmatrix} = \begin{bmatrix} 0 & 1 \\ -1 & -2 \end{bmatrix} \begin{Bmatrix} f \\ f' \end{Bmatrix}$$

To use the new PTI method, the equation can be transformed as

$$\begin{Bmatrix} y' \\ f' \\ f'' \end{Bmatrix} = \begin{bmatrix} -1 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & -1 & -2 \end{bmatrix} \begin{Bmatrix} y \\ f \\ f' \end{Bmatrix}$$

For form c, $f(x) = (1+x) \cdot \sin(x) \cdot e^{-x}$, the analytical solution is

$$y(x) = -e^{-x}[-2 + \cos(x) + x\cos(x) - \sin(x)]$$

Table 1 Numerical results and analytical solutions for function forms a–c

Result	$x = 0.5$	$x = 1.0$	$x = 2.0$	$x = 4.0$
		Form a		
Analytical	0.90979598956895	0.73575888234288	0.40600584970984	0.09157819444367
$\tau = 0.5$	0.90979598956893	0.73575888234287	0.40600584970984	0.09157819444368
$\tau = 1.0$	——	0.73575888234288	0.40600584970984	0.09157819444367
$\tau = 2.0$	——	——	0.40600584970984	0.09157819444367
		Form b		
Analytical	0.98561232203303	0.91969860292861	0.67667641618306	0.23810330555354
$\tau = 0.5$	0.98561232203303	0.91969860292860	0.67667641618305	0.23810330555355
$\tau = 1.0$	——	0.91969860292858	0.67667641618302	0.23810330555356
$\tau = 2.0$	——	——	0.67667641618290	0.23810330555359
		Form c		
Analytical	0.70542651231445	0.64778653730317	0.56268864125539	0.08262945917163
$\tau = 0.5$	0.70542651231449	0.64778653730321	0.56268864125537	0.08262945917163
$\tau = 1.0$	——	0.64778653730334	0.56268864125532	0.08262945917164
$\tau = 2.0$	——	——	0.56268864125514	0.08262945917169

Table 2 Numerical results and analytical solutions

Solution	$t = 1$, s	$t = 3$, s	$t = 5$, s	$t = 9$, s	$t = 11$, s	$t = 13$, s
Analytic, x_1	2.28168	−0.67259	−3.16659	1.90916	−0.35879	−1.95860
Constant approximation, x_1, $\tau = 0.2$	2.28007	−0.66495	−3.16452	1.90692	−0.34399	−1.95868
	(0.07%)[a]	(1.14%)	(0.07%)	(0.12%)	(4.12%)	(0.004%)
Linear approximation, x_1, $\tau = 0.2$	2.28218	−0.66790	−3.16583	1.90820	−0.34990	−1.95851
	(0.02%)	(0.70%)	(0.02%)	(0.05%)	(2.5%)	(0.005%)
Sinusoid approximation, x_1, $\tau = 1.0$	2.28168	−0.67259	−3.16659	1.90916	−0.35879	0.95860
	(0.0%)	(0.0%)	(0.0%)	(0.0%)	(0.0%)	(0.0%)
Analytic, x_2	1.76227	−0.84713	−1.16062	2.40173	−1.90372	0.31212
Constant approximation, x_2, $\tau = 0.2$	1.76246	−0.84718	−1.15374	2.39969	−1.89223	0.30876
	(0.01%)	(0.01%)	(0.59%)	(0.08%)	(0.60%)	(1.08%)
Linear approximation, x_2, $\tau = 0.2$	1.76206	−0.84587	−1.15798	2.40067	−1.89742	0.31098
	(0.01%)	(0.15%)	(0.20%)	(0.04%)	(0.33%)	(0.37%)
Sinusoid approximation, x_2, $\tau = 1.0$	1.76227	−0.84713	−1.16062	2.40173	−1.90372	0.31212
	(0.0%)	(0.0%)	(0.0%)	(0.0%)	(0.0%)	(0.0%)

[a]Parentheses give relative errors.

where $f(x)$ satisfies the differential equation with initial conditions:

$$f^{(4)} + 4f^{(3)} + 8f'' + 8f' + 4f = 0$$

$$f(0) = 0.0, \qquad f'(0) = 1.0, \qquad f''(0) = 0.0$$

$$f^{(3)}(0) = -4.0$$

To use the new PTI method, the equation can be transformed as

$$\begin{Bmatrix} y' \\ f' \\ f'' \\ f^{(3)} \\ f^{(4)} \end{Bmatrix} = \begin{bmatrix} -1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & -4 & -8 & -8 & -4 \end{bmatrix} \begin{Bmatrix} y \\ f \\ f' \\ f'' \\ f^{(3)} \end{Bmatrix}$$

The numerical results and analytical solutions are listed in the Table 1 for each form. The 10th digits of the numerical results behind the decimal point are the same as the analytical solutions, and the precision does not depend on the time step length. This special advantage is due to the accurate computation of the matrix exponential by using the PTI method.

Example 2

To solve the differential equations with initial conditions $x_1(0) = 2.5, x_2(0) = 0.0, \dot{x}_1(0) = 1.0$, and $\dot{x}_2(0) = 1.0$, the equations and their analytical solutions are as follows:

$$\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{Bmatrix} \ddot{x}_1 \\ \ddot{x}_2 \end{Bmatrix} + \begin{bmatrix} 1 & -1 \\ -1 & 2.5 \end{bmatrix} \begin{Bmatrix} x_1 \\ x_2 \end{Bmatrix} = \begin{Bmatrix} -\sin t \\ 0.5\sin t \end{Bmatrix}$$

$$x_1 = 2\cos\left(\sqrt{2}/2t\right) + 0.5\cos\left(\sqrt{3}t\right) + \sin t$$

$$x_2 = \cos\left(\sqrt{2}/2t\right) - \cos\left(\sqrt{3}t\right) + \sin t$$

Table 3 Time expense (seconds) with linear approximation

	100 Elements			200 Elements		
Results	$10N$	$15N$	$20N$	$10N$	$15N$	$20N$
Original method in Ref. 5	11.24	16.70	21.96	133.40	189.37	243.99
Linear approximation	9.93	15.59	20.66	111.70	165.90	217.74
Time cost reduction rate, %	11.7	6.6	5.9	16.3	12.4	10.8

This problem has one dynamic mode, $\sin(t)$. The numerical results and analytical solutions are shown in Table 2. The new PTI method with sinusoid approximation for the nonhomogeneous vector can obtain the exact solution even though its time step length τ is the largest one. The results obtained by linear and constant approximations for the nonhomogeneous vectors also have good accuracy.

Example 3

The torsion vibration of a tube with length $L = 1$, torsional stiffness per unit length $k_t = 1$, polar inertial moment $J_R = 1$, and density $\rho = 1$ is studied. One end of the tube is fixed, and the other end is subjected to a time-dependent torsion $T_n = \sin(\omega t)$. The example is used to test the computational efficiency of the new PTI methods. Here 100 elements and 200 elements are employed, respectively, and the algorithm parameter N of the PTI method is selected as 10, 15, and 20. The problem is run on a CPU PentiumIII500. The time cost of the original PTI method and new algorithm are given in Tables 3 and 4. It is shown that the new PTI method proposed in this paper is more efficient than the original one in time consumption. The greater the number of elements we use, the more time cost reduction we can get by the new algorithm. The reason is that for a larger problem the inverse computation of matrix $\boldsymbol{H}$ increases greatly. It is also shown that, as the algorithm parameter N gets larger, the time cost reduction rate become smaller because

Table 4 Time expense (seconds) with sinusoid approximation

Results	100 Elements			200 Elements		
	$10N$	$15N$	$20N$	$10N$	$15N$	$20N$
Original method in Ref. 6	11.91	17.17	22.09	141.05	198.05	252.16
Sinusoid approximation	9.96	15.53	20.65	111.45	166.26	218.08
Time cost reduction rate, %	16.4	9.6	6.5	21.0	16.1	13.5

as N gets larger, the computational amount of matrix $\boldsymbol{T}$ becomes the dominant factor in both the original method and the new method. Because N is the important parameter that controls the stability and precision of the PTI method, it cannot be too small. How to choose the parameter can be a research problem.

Conclusions

PTI with dimensional expanding is proposed. When dimensional expanding is used, the nonhomogeneous vector is viewed as the state variable of the system, and the original equations are converted into homogeneous equations. Thus, the method not only avoids the computation of the inverse matrix, but also improves the computational efficiency. In particular, the method is independent of the quality of the matrix $\boldsymbol{H}$. If the matrix $\boldsymbol{H}$ is singular or approximately singular, the advantages of the method are remarkable. If the nonhomogeneous vector is the solution of ODEs, the method can give exact results. Otherwise, linear or sinusoid treatment methods can also give good results. The PTI method is an excellent numerical scheme for ODEs, and it can be extended to other fields such as transient heat conduction, physical problems, and multibody dynamics, etc.

Acknowledgments

The research project presented is supported by the Scientific Fund for National Outstanding Youth of China (Grant 19525206), the National Natural Science Foundation of China (Grant 19872017), the Special Funds for National Key Basic Research of China (G1999032805), and the China–Russia collaborative research project. The authors would like to acknowledge the sponsors.

References

[1] Bathe, K. J., and Wilson, E. L., *Numerical Methods in Finite Element Analysis*, Prentice–Hall, New York, 1976, pp. 345–362.

[2] Subbaraj, K., and Dokainish, M. A., "A Survey of Direct Time Integration Methods in Computational Structural Dynamics: I Explicit Methods, II Implicit Methods," *Computers and Structures*, Vol. 32, No. 6, 1989, pp. 1371–1401.

[3] Moler, C. B., and Van Loan, C. V., "Nineteen Dubious Ways to Compute the Exponential of a Matrix," *SIAM Review*, Vol. 20, No. 4, 1978, pp. 801–836.

[4] Zhong, W. X., "On Precise Time-Integration Method for Structural Dynamics," *Journal of Dalian University of Technology*, Vol. 34, No. 2, 1994, pp. 131–136 (in Chinese).

[5] Zhong, W. X., and Williams, F. W., "A Precise Time Step Integration Method," *Journal of Mechanical Engineering Science*, Vol. 208, No. 6, 1994, pp. 427–430.

[6] Lin, J. H., Shen, W. P., and Williams, F. W., "A High Precision Direct Integration Scheme for Structures Subjected to Transient Dynamic Loading," *Computers and Structures*, Vol. 56, No. 1, 1995, pp. 113–120.

[7] Fung, T. C., "A Precise Time-Step Integration Method by Step-Response and Impulsive Response Matrices for Dynamic Problems," *International Journal for Numerical Methods in Engineering*, Vol. 40, No. 24, 1997, pp. 4501–4527.

A. Berman
Associate Editor