

a 529594 172
3x
D/Blue

Improved Precise Time-Step Integration Algorithms for Dynamic Problems

Chen ZhenLin



School of Civil and Environmental Engineering

A thesis submitted to the Nanyang Technological University
in fulfilment of the requirement for the degree of
Doctor of Philosophy

2007

CONTENTS

Acknowledgement	i
Summary	ii
List of Tables	iv
List of Figures	vi
Nomenclature	vii
Chapter 1 Introduction	1
1.1 Background	1
1.2 Objective	2
1.3 Major Contributions	3
1.4 Organization	4
Chapter 2 Literature Review	7
2.1 Overview	7
2.2 Dynamic Equations' Solution Methods	7
2.2.1 Analytical Methods	8
2.2.2 Numerical Methods	10
2.3 Time Integration Algorithms	12
2.3.1 Explicit Time Integration Methods	12
2.3.2 Implicit Time Integration Methods	17
2.4 Methods for Solving Exponential Matrix	22
2.4.1 Series Methods	22
2.4.2 Scaling and Squaring Techniques	25
2.4.3 Efficiency of Various Methods	25
2.5 Precise Time-Step Integration methods	26
2.6 Krylov Subspace Method	27
2.6.1 Background of Krylov Subspace Method	28
2.6.2 Application of Krylov Subspace Methods	28
2.7 Concluding Remarks	30
Chapter 3 Precise Time-Step Integration Method	33
3.1 Basic Theory of Precise Time-Step Integration Method	34

3.2 Extension of Precise Time-Step Integration Method	36
3.2.1 Dimensional Expanding Methods	36
3.3 Operation Counts	41
3.4 Derived Algorithms Analysis	44
3.5 Numerical Example	44
3.6 Summary	46
Chapter 4 Precise Time-Step Integration Algorithms Using Response Matrices with Dimensional Expansion	48
4.1 PTI Algorithms by Step-Response and Impulsive-Response Matrices	49
4.1.1 Step-Response and Impulsive-Response Matrices	51
4.1.2 Steady-State Response	52
4.1.3 Symmetric Matrices	53
4.2 Application of Dimensional Expanding Method	54
4.3 Excitation Described by Second-Order Differential Equation	55
4.4 Excitation Described by First-Order Differential Equation	57
4.4.1 Computation of $[\mathbf{E}(t)]$, $[\mathbf{D}(t)]$ and $[\dot{\mathbf{D}}(t)]$	58
4.4.2 Relation of $[\mathbf{D}(t)]$ and $[\dot{\mathbf{D}}(t)]$	60
4.5 Computational Effort	63
4.6 Numerical Examples	67
4.7 Summary	74
Chapter 5 Unconditionally Stable PTI Algorithms by Padé Approximations	80
5.1 Introduction	81
5.2 Padé Approximations	81
5.3 Scaling and Squaring Algorithms for Evaluation of Exponential Matrix	83
5.4 Padé Approximations for $[\mathbf{H}]$ and $[\dot{\mathbf{H}}]$	85
5.4.1 Ordinary Padé Series Approximation	86
5.4.2 New Padé Approximation	88
5.4.2.1 Implementation of Rayleigh Damping	91
5.4.2.2 Symmetric Matrices	93

5.5 Computational Effort	95
5.6 Algorithms Stability	96
5.7 Numerical Examples	98
5.8 Summary	105
Chapter 6 Krylov Precise Time-Step Integration Algorithms	123
6.1 Introduction	124
6.2 General Theory of Krylov Subspace Method	124
6.2.1 Implementation of Krylov Subspace Method	126
6.3 Algorithms Analysis	128
6.3.1 Error Estimation	128
6.3.2 Algorithms Stability	129
6.3.3 Non-Linear Systems	130
6.4 Computational Effort	130
6.5 Numerical Examples	133
6.6 Summary	138
Chapter 7 Application of the Improved Precise Time-Step Integration Algorithms	148
7.1 Introduction	149
7.2 Improvement for Solving ODEs	150
7.3 Dynamic Analysis of the Structures under Moving Loads	154
7.4 Analysis of Critical Wind Velocity of Galloping Oscillation of High-Rise Structures	157
7.5 Summary	160
Chapter 8 Conclusions and Future Work	166
8.1 Conclusions	166
8.2 Recommendations	170
References	172
Publications	184

Acknowledgements

After going through almost three years of hard work it is time to thank all those who have pulled me through this period and made my stay at NTU a pleasant one.

I would like to express my sincere gratitude and thanks to Prof. Fung Tat Ching for his invaluable guidance and moral support.

My special gratitude is due to all my friends for making my time spent at NTU an unforgettable memory.

I would like to thank the School of Civil and Environmental Engineering for the full financial support and the research facilities they provided during my study.

Summary

A study on the properties of the precise time-step integration methods for the simulation of dynamic responses of engineering systems is presented in this thesis. The improved precise time-step integration algorithms can reduce the computational cost significantly and maintain the accuracy as well.

The PTI method has been improved incorporating (1) the extended dimensional expanding method, (2) Duhamel-response matrix, (3) Padé approximation, and (4) Krylov subspace method.

The dimensional expanding method is imported to improve the efficiency of the original precise time-step integration method and simplify the computing process. The particular solution is avoided to be computed by transforming the non-homogeneous equations into homogeneous equations. The dimensional expanding method is extended to solve the equations directly when excitation is described by first-order and second-order differential equation. The efficiency of the original PTI method can be improved significantly.

The precise time-step integration method by step-response and impulsive-response matrices is further developed by incorporating the Duhamel Integral using the Duhamel-response matrix. The efficiency has been improved and the implementation process has been simplified. The computation of the Duhamel-response matrix and its time derivatives are studied in this research. The symmetric property can be preserved to reduce much computational cost.

The Padé series approximation is used to improve the stability and accuracy of the precise time-step integration algorithm. The precise time-step integration method by step-response and impulsive-response matrices will be unconditionally stable by using a new Padé approximation. With an accurate computation of the exponential matrix, the low order Padé approximation can generate highly accurate results. The first four-order Padé approximation is given explicitly in this research.

A great deal of attention has been devoted to the Krylov subspace techniques for reduced-order modeling of large-scale dynamical systems. The Krylov precise time-step integration algorithm incorporating with Padé series approximation and dimensional expanding method can reduce the computational effort significantly and improve the stability, especially for solving large-scale systems. The criteria to choose N (number of recursive evaluations), p (order of the Padé approximation) and the efficiency range of m (order of the Krylov subspace) are studied. The present algorithms can also be extended to tackle non-linear problems without difficulty.

The proposed precise time step integration algorithms in this research are excellent numerical schemes for ODE (Ordinary Differential Equations), and can be extended to other field such as non-linear transient heat conduction and moving load problems, etc. A summary of the application of the proposed precise time-step integration algorithms is given in this research.

List of Tables

Table 1-1	The brief summary of techniques of the methods in this thesis	6
Table 2-1	Comparison of explicit and implicit time integration method	31
Table 3-1	Computational costs of the methods (A1 , A2 and A3)	47
Table 4-1	The operation count of the method (A3)	76
Table 4-2	The operation count of the method (A4)	76
Table 4-3	The operation count of the method (A5)	77
Table 4-4	The operation count of the method (A6)	77
Table 4-5	Computational effort (linear excitation) to evaluate the same accurate result u_{N_s} at $t=10s$	78
Table 4-6	Computational effort (Polynomial excitation) to evaluate the same accurate result u_{N_s} at $t=10s$	78
Table 4-7	Computational effort (Complex excitation) to evaluate the same accurate result u_{N_s} at $t=10s$	79
Table 4-8	Computational effort to evaluate the results at the end of two periods ($t=12.0s$) in Example 2	79
Table 5-1	$[\mathbf{T}_a^*]$ for various values of p	113
Table 5-2	Formulae of q_i and d_i	113
Table 5-3 (a)	Formulae of first four-order new Padé approximation to $g(t)$	114
Table 5-3 (b)	Formulae of first four-order new Padé approximation to $h(t)$	115
Table 5-3 (c)	Formulae of first four-order new Padé approximation to $\dot{g}(t)$	116
Table 5-3 (d)	Formulae of first four-order new Padé approximation to $\dot{h}(t)$	117
Table 5-4 (a)	Comparison of the stability of the PTI methods (A1 and A7) at $t=100s$, $N=1, 2$	118
Table 5-4 (b)	Comparison of the stability of the PTI methods (A1 and A7) at $t=100s$, $N=5, 6$	119
Table 5-5	Comparison of the efficiency and accuracy of the PTI methods (A1 and A7) ($N_s=100$)	120
Table 5-6	Comparison of the efficiency and accuracy of the PTI	121

	method (A7) with different order (p) ($N_s=100$)	
Table 5-7	Comparison of the efficiency of the PTI methods (A4 and A8)	122
Table 6-1 (a)	Operation counts of the method A3 (Banded matrices)	139
Table 6-1 (b)	Operation counts of the method A10 (Banded matrices)	139
Table 6-2 (a)	Operation counts of the method A3 (Full matrices)	140
Table 6-2 (b)	Operation counts of the method A10 (Full matrices)	140
Table 6-3	Computational effort to evaluate the result u_{N_s} at $t=100s$ for various N_s	141
Table 6-4	Computational effort to evaluate the result u_{N_s} at $t=100s$ with $\Delta t = 1.0s$ for various N_s	142
Table 6-5	Computational effort for the presented method A10 to evaluate the result u_{N_s} at $t=100s$ ($N_s=100$)	143
Table 6-6	Computational effort for the presented method A10 to evaluate the result u_{N_s} at $t=100s$ ($N_s=200$).	144
Table 6-7	Computational effort for the presented method A10 to the evaluate the result u_{N_s} at $t=100s$ ($N_s=400$)	145
Table 6-8	The influence of ill-conditioned stiffness matrices to proposed Krylov PTI method (A10)	146
Table 6-9	Computational effort to evaluate result u_{N_s} at $t=1.0s$ in Example 2	147
Table 7-1	Comparison of Newmark method, method A6 and method A10	161

List of Figures

Figure 2-1	Methods for computing the exponential matrix function	32
Figure 4-1	The truss with six storey and three bars and the excitation	75
Figure 4-2	The discontinue periodic external loading form	75
Figure 5-1 (a)	$h(t)$ and its Padé approximations ($\xi = 0, \omega = 1$)	106
Figure 5-1 (b)	$h(t)$ and its Taylor series approximations ($\xi = 0, \omega = 1$)	106
Figure 5-1 (c)	$g(t)$ and its Padé approximations ($\xi = 0, \omega = 1$)	107
Figure 5-1 (d)	$g(t)$ and its Taylor series approximations ($\xi = 0, \omega = 1$)	107
Figure 5-2 (a)	$h(t)$ and its Padé approximations ($\xi = 1, \omega = 1$)	108
Figure 5-2 (b)	$h(t)$ and its Taylor series approximations ($\xi = 1, \omega = 1$)	108
Figure 5-2 (c)	$g(t)$ and its Padé approximations ($\xi = 1, \omega = 1$)	109
Figure 5-2 (d)	$g(t)$ and its Taylor series approximations ($\xi = 1, \omega = 1$)	109
Figure 5-3 (a)	$h(t)$ and its Padé approximations ($\xi = 2, \omega = 1$)	110
Figure 5-3 (b)	$h(t)$ and its Taylor series approximations ($\xi = 2, \omega = 1$)	110
Figure 5-3 (c)	$g(t)$ and its Padé approximations ($\xi = 2, \omega = 1$)	111
Figure 5-3 (d)	$g(t)$ and its Taylor series approximations ($\xi = 2, \omega = 1$)	111
Figure 5-4	A series of eight mass-springs with a damper at the right-hand connection	112
Figure 7-1	A simple supported beam under different moving loads	162
Figure 7-2	Schematic vehicle-bridge model	163
Figure 7-3	A simple supported beam under moving mass	163
Figure 7-4	Schematic of wind loads on structure cross sections where $F_D(\alpha_x)$ is downwind stress and $F_L(\alpha_x)$ is crosswind stress	164
Figure 7-5	Deflection of beam at mid-span under the moving load, $\Delta t = 2.0 \times 10^{-2} s$	165
Figure 7-6	Deflection of beam at mid-span under the moving load, $\Delta t = 5.0 \times 10^{-3} s$	165

Nomenclature

A1	=	The original PTI method
A2	=	The PTI method combined with the dimensional expanding method (Gu et al., 2001)
A3	=	The PTI method combined with the dimensional expanding method (Wang et al., 2002)
A4	=	The PTI algorithm by step-response and impulsive-response matrices (Fung, 1997)
A5	=	The PTI algorithm by step-response and impulsive-response matrices with dimensional expansion
A6	=	The PTI algorithm by step-response, impulsive-response matrices and Duhamel-response matrix with dimensional expansion
A7	=	The original PTI method with ordinary Padé series approximation
A8	=	The PTI method by step-response and impulsive-response matrices with the new Padé approximation
A9	=	The Krylov precise time-step integration algorithm
A10	=	The Krylov precise time-step integration algorithm with dimensional expansion
$[\mathbf{c}_v]$	=	Damping of the vehicle
$\{\mathbf{R}_I\}$	=	Independent of temperature of heat load vector
$[\mathbf{k}_v]$	=	Stiffness of the vehicle
α_x	=	The angle of wind flow and horizontal axis
μ	=	The constant mass per unit length of the beam
$F_y(z)$	=	The galloping oscillation stress at structure height z
x_{ci}	=	The position of contract point in vehicle-bridge system
$\{\mathbf{R}_r\}$	=	The radioactive part of heat load vector
$\{\mathbf{N}_c\}$	=	Vector containing cubic Hermitian interpolation functions

ω_1	=	y-axis natural frequency of the first order model
$\{\mathbf{f}_c\}$	=	The interaction force between the suspension unit and the
$[\mathbf{J}_a(t)]$	=	$[\mathbf{J}(t)] - [\mathbf{I}]$
τ	=	$\Delta t / 2^N$
$\{\mathbf{R}\}$	=	$[\mathbf{M}]^{-1} \{\mathbf{r}(t)\}$
$[\mathbf{A}]$	=	$-[\mathbf{M}]^{-1}[\mathbf{K}]$
$[\mathbf{B}]$	=	$-[\mathbf{M}]^{-1}[\mathbf{C}]$
m_1	=	2^N
$[\mathbf{H}_m]$	=	$m \times m$ upper Hessenberg matrix
$[\overline{\overline{\mathbf{C}}}]$	=	$[\mathbf{L}]^{-1}[\mathbf{C}][\mathbf{L}]^{-T}$
$[\overline{\overline{\mathbf{K}}}]$	=	$[\mathbf{L}]^{-1}[\mathbf{K}][\mathbf{L}]^T$
$[\mathbf{E}(t)]$	=	Additional response matrix
$\ \bullet\ $	=	Arbitrary order norm
$[\mathbf{A}_r]$	=	Arbitrary square matrix
$[\mathbf{f}]$	=	Coefficient matrix corresponding to external excitation
$[Inte]$	=	Computational cost for solving particular solution
$[\mathbf{T}_a]$	=	Computing matrix of exponential matrix function
$\cos()$	=	Cosine function
$[\mathbf{c}_b]$	=	Damping matrices of the beam element
c_s	=	Damping of Single-Degree-of-Freedom system
$\mathbf{Q}_{pq}(t)$	=	Denominator of ordinary (p, q) Padé series approximation
m	=	Dimension of Hessenberg matrix
N_s	=	Dimension of system
$[\mathbf{T}^*]$	=	Dimensional expanding exponential matrix function
$[\mathbf{S}_1]$	=	Dimensional expanding matrix for first-order equation
$[\mathbf{S}_2]$	=	Dimensional expanding matrix for second-order equation
$\{\mathbf{u}^*\}$	=	Dimensional expanding time-dependent displacement vector

$[\mathbf{W}]$	=	Dimensional matrix
$\{\mathbf{y}\}$	=	Dimensional vector
$\{\mathbf{d}_b\}$	=	Displacement vector of beam element
$\{\mathbf{d}_c\}$	=	Displacement vector of the contact points on the bridge
$\{\mathbf{d}_v\}$	=	Displacement vector of vehicle
$\{\mathbf{d}_w\}$	=	Displacement vector of wheel part
$[\mathbf{D}(t)]$	=	Duhamel-response matrix
$\exp(\bullet)$	=	Exponential matrix function
$[\mathbf{T}]$	=	Exponential matrix function
$\{\mathbf{r}(t)\}$	=	External excitation vector
$\{\bar{\mathbf{r}}(t)\}$	=	External excitation with time shift
$\{r(t)\}$	=	External force
$\{\mathbf{R}_q\}$	=	Heat load vector
$[\mathbf{I}]$	=	Identity matrix
$h(t)$	=	Impulsive-response function
$[\mathbf{H}(t)]$	=	Impulsive-response matrix
$\ \bullet\ _\infty$	=	Infinite order norm
$\{\mathbf{u}_0\}$	=	Initial condition of displacement vector
$\{\mathbf{v}_0\}$	=	Initial condition of velocity vector
ε	=	Inverse error
$[\mathbf{L}]$	=	Lower triangular matrix
$[\mathbf{m}_b]$	=	Mass matrices of the beam element
$[\mathbf{m}_v]$	=	Mass matrices of the vehicle
m_s	=	Mass of Single-Degree-of-Freedom system
\mathbf{K}_m	=	m -th Krylov subspace
$\mathbf{N}_{pq}(t)$	=	Numerator of ordinary (p, q) Padé series approximation
$\mathbf{P}_{pq}(t)$	=	Ordinary (p, q) Padé series approximation
$\mathbf{P}_p(t)$	=	Ordinary diagonal (p, p) Padé series approximation

$[\mathbf{V}_m]$	=	Orthogonal basis for the Krylov subspace
$\ \bullet\ _2$	=	Second order norm
$\sin()$	=	Sinusoidal function
$\{\mathbf{u}_s\}$	=	Steady-state response
$g(t)$	=	Step-response function
$[\mathbf{G}(t)]$	=	Step-response matrix
$[\mathbf{k}_b]$	=	Stiffness matrices of the beam element
k_s	=	Stiffness of Single-Degree-of-Freedom system
$[\overline{\mathbf{H}}(t)]$	=	Symmetric matrices of $[\dot{\mathbf{H}}(t)]$ when $[\mathbf{M}]$ is diagonal
$[\overline{\overline{\mathbf{H}}}(t)]$	=	Symmetric matrices of $[\dot{\mathbf{H}}(t)]$ when $[\mathbf{M}]$ is not diagonal
$[\overline{\mathbf{J}}(t)]$	=	Symmetric matrices of $[\mathbf{J}(t)]$ when $[\mathbf{M}]$ is diagonal
$[\overline{\overline{\mathbf{J}}}(t)]$	=	Symmetric matrices of $[\mathbf{J}(t)]$ when $[\mathbf{M}]$ is not diagonal
$\{\mathbf{q}\}$	=	temperature vector
g	=	Terms of external polynomial excitation
n	=	Terms of Taylor series approximation
ρ	=	The air density
L	=	The beam length
ω_b	=	The circular frequency of damping of the beam
γ	=	The coefficient of Krylov subspace
h_{ij}	=	The coefficients of Hessenberg matrix
J	=	The constant moment of inertia of the beam cross section
ξ	=	The damping ratio
$\{\mathbf{p}_v\}$	=	The external force components in vehicle-bridge system
$\{\mathbf{p}_b\}$	=	The external nodal forces
$\{\mathbf{e}_i\}$	=	The last column of the identity matrix $[\mathbf{I}]$
N	=	The number of recursive evaluations
f_s	=	The number of terms of Fourier series
p, q	=	The order of the Padé series approximation

		matrix function
$\mathbf{Q}_p(t)$	=	Denominator of ordinary diagonal (p, p) Padé series approximation
$[\mathbf{H}^*(t)]$	=	Dimensional expanding impulsive-response matrix
$[\mathbf{G}^*(t)]$	=	Dimensional expanding step-response matrix
$\{\mathbf{d}_u\}$	=	Displacement vector of the non-contact part consists of the car body
$\mathbf{N}_p(t)$	=	Numerator of ordinary diagonal (p, p) Padé series approximation
$[\dot{\mathbf{H}}^*(t)]$	=	Time differential of dimensional expanding impulsive response matrix
$[\mathbf{J}^*(t)]$	=	Time differential of dimensional expanding impulsive response matrix
$[\dot{\mathbf{G}}^*(t)]$	=	Time differential of dimensional expanding step-response matrix
η, c_i	=	Coefficients of general external excitation
α, β	=	Coefficients of Rayleigh damping
β_1, β_2	=	Newmark parameters
$\{\mathbf{a}_i\}, \{\mathbf{b}_i\}$	=	Coefficients of general external excitation
$\alpha'_{ij}, \beta'_{ij}$	=	Coefficients of ordinary (p, q) Padé series approximation to $\dot{h}(t)$
α_{ij}, β_{ij}	=	Coefficients of ordinary (p, q) Padé series approximation to $h(t)$
$\{\mathbf{r}_0\}, \{\mathbf{r}_1\}$	=	Time-invariant vectors of external excitation
PTI	=	The Precise Time-step Integration method

CHAPTER ONE

Introduction

1.1 Background

In the last decade, significant advancements have been achieved in the development and application of numerical methods to solve dynamic problems. Recently, numerous research work has been directed towards improving the computational efficiency in solving dynamic problems such as temporal integration of discrete equations of motion and solution of the resulting nonlinear algebraic matrix equation systems. This thesis focuses on the numerical solution of the governing equations for dynamic vibration of structures.

Time integration technique was first applied to solve the partial differential equations of mixed type (elliptic and hyperbolic) in the mid-1960s. The steady state solution was obtained by starting with the unsteady equation, and marching the solution along the time coordinate until a steady state response was achieved. Nowadays, time integration algorithms are widely adopted in computational dynamics. Dokainish and Subbaraj (1989) reviewed the direct time-integration methods in computational structural

dynamic including the explicit and implicit time integration algorithms. In the family of explicit algorithms, central difference method (Krieg, 1976; Warburton, 1985), Runge-Kutta method, stiffly stable method (Jensen, 1974, 1976; Park, 1975), and Predictor-Corrector method were widely used. For the implicit methods, Newmark family of methods (Newmark, 1959), Wilson- θ method (Wilson et al. 1973), and Houbolt method were developed extensively. It should be noted that most of these methods make use of very small time step and therefore the computational cost is high, although some implicit algorithms can overcome this disadvantage. Thus, many modified algorithms have been developed recently to solve dynamic problems.

The precise time-step integration (PTI) method was first proposed and developed by Zhong and Williams (1994). This method can compute the exponential matrix accurately. Fung (1997) presented a precise time-step integration method by step-response and impulsive-response for dynamic problems as well. The second order differential equations for dynamic problems were manipulated directly. With the implementation of the dimensional expanding method, non-homogeneous equations can be transformed into homogeneous equations. But the computational cost of this expanding method is high. Gu et al. (2001) and Wang et al. (2002) developed the precise time-step integration (PTI) method by using two kinds of dimensional expanding method respectively.

1.2 Objective

Currently, time-step integration methods are widely used in the simulation of dynamic responses of engineering systems. In practice, the major part is to solve the governing dynamic equations efficiently and accurately. The current precise time-step integration algorithms for dynamic response of structures are widely used in engineering. This

research focuses on improvement of the accuracy, efficiency and stability of the precise time-step integration algorithms.

1.3 Major Contributions

In this research, the precise time-step integration algorithms are investigated and developed comprehensively. The major contributions of the present work are:

- (1) The original precise time-step integration method is further studied and developed for solving dynamic problems. The application of the critical principle for choosing the optimum value of N (scale and squaring factor) is further studied.
- (2) The computational cost can be significantly reduced, when the dimensional expanding method is employed to improve the derived PTI algorithms. The dimensional expanding method is extended to tackle first-order and second-order equations directly and the arbitrary excitation is considered. The computational cost could be reduced about half.
- (3) The precise time-step integration algorithm by step-response and impulsive-response matrices is further developed. The Duhamel matrix is imported to solve the particular solution instead of the Duhamel integral. The symmetric property is used to further reduce the computational cost. Two new algorithms with the excitation described as First-order and Second-order differential equation are proposed based on the extension of the dimensional expanding method.
- (4) The Padé approximation instead of the Taylor series approximation is employed to improve the efficiency and stability of the precise time-step integration algorithms. It renders the precise time-step integration

algorithms more stable. An unconditionally stable precise time-step integration algorithm is obtained using the new proposed Padé approximation.

- (5) For large-scale systems, the Krylov precise time-step integration method is proposed to improve the efficiency. The permissible efficient range of order of the Krylov subspace is obtained from the analysis of computational efforts. The Krylov precise time-step integration method can also be extended to solve the non-linear problems. The characteristics of the efficiency and stability of the Krylov precise time-step integration method are investigated.

Examples are given to illustrate the high accuracy and efficiency of the proposed algorithms. The proposed precise time-step integration algorithms can be widely employed to solve the dynamic problems, such as, ODEs, moving load problems and non-linear transient heat conduct problems. Table 1-1 summarizes the characteristics of different methods (A1 to A10) mentioned in this thesis.

1.4 Organization

The outline of the thesis is as follows:

Chapter 2 presents a literature review of the current direct time step integration algorithms.

In Chapter 3, a general framework of the precise time integration algorithm is outlined. The dimensional expanding method and its extension are discussed.

Chapter 4 proposes a new precise time-step integration method by step-response, impulse-response and Duhamel-response matrices for solving second-order dynamic problems directly.

Chapter 5 develops the unconditionally stable precise time-step integration algorithm by Padé approximation.

Chapter 6 proposes the Krylov precise time-step integration method by using the Krylov subspace method and Padé approximation. Large-scale problems can be tackled and the computational efficiency is improved significantly.

Chapter 7 applies to present proposed methods to solve some practical engineering problems.

Lastly, general conclusions of this research are present in Chapter 8. Recommendations are also given for future research.

Table 1-1 The brief summary of techniques of the methods in this thesis

Method	A1	A2	A3	A4	A5	A6	A7	A8	A9	A10
Old or New Algorithm	O	O	O	O	N	N	N	N	N	N
2^N technique	*	*	*	*	*	*	*	*	*	*
Taylor series approximation	*	*	*	*	*	*				
Padé series approximation							*	*	*	*
Dimensional expanding method		*	*		*	*				*
Order of computing equation	1	1	1	2	1	1	2	2	1	1
Krylov subspace method									*	*

Note: the symbol “*” denotes the method include the technique; “O” denotes the old method; “N” denotes the new method proposed in this thesis.

CHAPTER TWO

Literature Review

2.1 Overview

The focus of this research is on the time-step integration algorithm for solving dynamic problems. A large number of researches have been carried out in this field in the past few decades. Recently, numerous analytical and numerical methods were proposed. Efficient higher order accurate time integration algorithms, including both explicit and implicit time integration algorithms, have been widely developed for dynamic problems.

2.2 Dynamic Equations' Solution Methods

The dynamic motion equation of single-degree-of-freedom system can be written as

$$m_s \cdot \ddot{u}(t) + c_s \cdot \dot{u}(t) + k_s \cdot u(t) = r(t) \quad (2-1)$$

where m_s , c_s , k_s are the mass, damping, and stiffness, respectively. $r(t)$ is the external force.

The equations of a multi-degree freedom system after spatial discretization using the finite element method can be written as:

$$[\mathbf{M}]\{\ddot{\mathbf{u}}(t)\} + [\mathbf{C}]\{\dot{\mathbf{u}}(t)\} + [\mathbf{K}]\{\mathbf{u}(t)\} = \{\mathbf{r}(t)\} \quad (2-2)$$

where $[\mathbf{M}]$, $[\mathbf{C}]$ and $[\mathbf{K}]$ are the time-invariant mass, damping and stiffness matrices, respectively, and $\{\mathbf{r}(t)\}$ is the known external force vector. Many engineers prefer to solve the second order equation by direct methods. But there are some advantages of reducing the second order equation to an equivalent first order system. A very simple formula is given here, i.e. Equation (2-2) can be rewritten as:

$$\begin{Bmatrix} \dot{\mathbf{u}}(t) \\ \ddot{\mathbf{u}}(t) \end{Bmatrix} = \begin{bmatrix} 0 & \mathbf{I} \\ -\mathbf{M}^{-1}\mathbf{K} & -\mathbf{M}^{-1}\mathbf{C} \end{bmatrix} \cdot \begin{Bmatrix} \mathbf{u}(t) \\ \dot{\mathbf{u}}(t) \end{Bmatrix} + \begin{bmatrix} 0 \\ \mathbf{M}^{-1}\mathbf{r}(t) \end{bmatrix} \quad (2-3)$$

Numerical methods for first order equations can be applied to the system (2-3).

In order to obtain the dynamic response of the structures, the dynamic equations established by discrete or continuum method should be solved accurately. There are two types of methods that can be used to solve the dynamic response, namely, analytical methods and numerical methods.

2.2.1 Analytical Methods

Solution of differential equations can be obtained analytically. For first order differential equations, $\dot{u} = f(t, u)$, there are two classes of equations possessing solutions in elementary forms:

$$(1) \quad \text{Separable Equations: } \frac{du}{dt} = r(t)s(u)$$

The method of separation of variables applies to the case where $\dot{u} = f(t, u)$ can be written as

$$\frac{du}{dt} = r(t)s(u) \quad (2-4)$$

If $s(u_0) = 0$, then $u(t) = u_0$.

If $s(u) \neq 0$, then the variables can be separated by rewriting the equation as

$$\frac{du}{s(u)} = r(t)dt, \text{ and } \int \frac{du}{s(u)} = \int r(t)dt \quad (2-5)$$

If the integrals can be computed,

$$\int \frac{du}{s(u)} = S(u) + C_s; \int r(t)dt = R(t) + C_r \quad (2-6)$$

then the solution has the form

$$S(u) = R(t) + C \quad (2-7)$$

In dynamic problems studied in this research, $\dot{u} = f(t, u)$ does not depend explicitly on u , then $\dot{u} = r(t)$, and to find the solutions is simply equivalent to find the antiderivative of $r(t)$.

$$(2) \quad \text{Linear Equations:} \quad \dot{u} = p(t)u + q(t)$$

The solution to $\dot{u} = p(t)u + q(t)$ with initial condition $u(t_0) = u_0$ is

$$u(t) = u_h(t) + u_p(t) \quad (2-8)$$

where u_h and u_p are the solutions of the homogeneous equation and the particular solution of the non-homogeneous equation, respectively. According to the initial condition, $u_c = u_0 e^{\int p(t)dt}$.

Green function formula, Galerkin space-time method and series expansion methods were developed extensively for solving dynamic response of second-order differential equations. Yang (1996a) obtained the closed-form transient response of

distributed damped systems by modal analysis and Green's function formula based on a state-space formulation. The convergence of the modal superposition is considered without the completeness assumption for state-space eigenfunctions. Besides, Yang (1996b) studied the Green's function formulation for transient response prediction.

Ekevid et al. (2001) developed an efficient numerical procedure for solving problem associated with wave propagation in the track-ground system. The approach is based on the discontinuous Galerkin space-time method, where the finite element discretization is employed both spatially and temporally.

The solution of the moving oscillator problem can be obtained as the series expansion in terms of the eigenfunctions of the distributed system (Pesterev and Bergman, 1997a, b and 1998a, b). Pesterev et al. (2000) derived a new series representation. Later, Pesterev et al. (2001a, b) presented a new series expansion for calculating the bending moment and the shear force in a proportionally damped, one-dimensional distributed parameter system due to moving loads.

2.2.2 Numerical Methods

For analytical methods, there are many limitations, such as the complexity of algorithm, difficulties in implementation, etc. Numerical methods become increasingly popular with rapid advances of computer techniques. Many new numerical methods appeared, such as FEM, BEM, time step integration algorithms etc. The stability and accuracy characteristics can be considered at the same time. There are many numerical schemes available in the literature, such as Taylor series method, Galerkin method, Euler and Modified Euler method, Runge-Kutta methods, Milne method and Adams-Moulton method. Recently, Zhong and Williams (1994)

presented a precise time-step integration algorithm (PTI), which will be discussed later.

Pan and Atluri (1995) presented a coupled boundary element method/finite element method (BEM/FEM) to solve the problem of a moving load on a finite elastic plate, resting on an elastic half space. Yang et al. (2000) studied a versatile element that is capable of solving dynamic response of various vehicle-bridge interaction effects using Newmark's finite difference scheme to discretize the vehicle equations of motion. Rasmussen et al. (2001) investigated a new BEM formulation in a moving coordinate system. The formulation makes use of Green's functions for a moving force and shows how the normal BEM formulation for a fixed coordinate system can be modified to a formulation in the moving coordinate system by this treatment. They compared the BEM results with the ones obtained by a FEM formulation. In comparison, the results obtained using these two kinds of methods are in good agreement.

When the dynamic equations were established using FEM or other methods, the main task is to solve Ordinary Differential Equations (ODEs). Plenty of time integration methods were developed recently. For second order ODE problems, there are two kinds of solution methods:

- One is to solve the second-order equation directly. For example, the central difference methods and Newmark methods can be used to solve the equations directly.
- The other one is to transform the second order equations into the first order equations. There are many techniques to solve the first-order equations, such as the Finite Difference Methods, Runge-Kutta methods, etc, which techniques will be introduced in next section.

2.3 Time Integration Algorithms

It is a common practice in analyzing an engineering structure under shock, gust or similar loading to perform a transient analysis to check its time history behavior. Finite element discretization is usually applied for the space coordinates and yields a multi-degree of freedom vibration system. The eigenvector expansion method usually selects only a small set of the lowest frequency eigensolutions, which is inadequate for checking the time history behavior.

There are several well-established time step integration methods, for example, central difference method, Newmark method and *Wilson - θ* method. Basically they can be classified into two categories as implicit or explicit integration schemes. Simple comparison between explicit and implicit time-integration was shown in Table 2-1. In implicit algorithms, a matrix system is solved one or more times per time step to obtain the solution, while in explicit algorithms the solution may be obtained without storing a matrix or solving a system of equations. Implicit algorithms tend to be numerically stable, permitting large time steps. But the cost per time step is high and the storage requirement increases dramatically with the size of the finite element mesh, particularly in large three-dimensional problems. It becomes more and more important to improve the computational efficiency and convergence of the numerical algorithms.

2.3.1 Explicit Time Integration Methods

The first approach, which is based on the explicit time integration techniques, employs finite difference methods. Finite difference method is particularly well suited for short duration dynamical problems or wave propagation problems such as structures subjected to blast or high velocity impact. Computational cost per time-step is generally much less for explicit methods. However, explicit time integration schemes are only conditionally stable and generally required small time steps to be

employed to ensure numerical stability. Here, the step size restriction is often more critical than accuracy considerations requirement (Subbaraj and Dokainish, 1989). This restriction limits the effectiveness of the approach when used to study dynamic problems of moderate duration, e.g. earthquake response problems. The conditionally stable algorithms require the time step size employed to be inversely proportional to the highest frequency of the discrete system. Some simple explicit time integration procedures are introduced below.

(1) *Second Order Central Difference Method*

The second order central difference explicit method is one of the most widely used algorithms among explicit (numerical integration) techniques in large-scale structural dynamics programs. The central difference method is proved to have the highest accuracy and maximum stability limit among any explicit method of order two. However, it has the disadvantage of requiring small time steps. The central difference method is based on the following formulas:

$$\{\dot{\mathbf{u}}_{t+\Delta t/2}\} = (\{\mathbf{u}_{t+\Delta t}\} - \{\mathbf{u}_t\}) / \Delta t \quad (2-9)$$

$$\{\dot{\mathbf{u}}_{t-\Delta t/2}\} = (\{\mathbf{u}_t\} - \{\mathbf{u}_{t-\Delta t}\}) / \Delta t \quad (2-10)$$

$$\{\dot{\mathbf{u}}_t\} = (\{\mathbf{u}_{t+\Delta t/2}\} - \{\mathbf{u}_{t-\Delta t/2}\}) / \Delta t \quad (2-11)$$

$$\{\ddot{\mathbf{u}}_t\} = (\{\dot{\mathbf{u}}_{t+\Delta t/2}\} - \{\dot{\mathbf{u}}_{t-\Delta t/2}\}) / \Delta t \quad (2-12)$$

$$\{\mathbf{u}_{t+\Delta t/2}\} = \frac{1}{2}(\{\dot{\mathbf{u}}_{t+\Delta t}\} + \{\mathbf{u}_t\}) \quad (2-13)$$

$$\{\mathbf{u}_{t-\Delta t/2}\} = \frac{1}{2}(\{\mathbf{u}_t\} + \{\dot{\mathbf{u}}_{t-\Delta t}\}) \quad (2-14)$$

where $t - \Delta t$, t and $t + \Delta t$ are three successive time levels.

The effectiveness of the procedure depends on the use of a diagonal mass matrix obtained by mass lumping, and the omission of the velocity-dependent damping forces. If only a diagonal damping matrix is included, the advantages of performing the solution on the element level can be preserved. On the other hand, if the damping matrix was non-diagonal, the solution required factorization of the

effective mass matrix. There is a modified explicit form suggested by Warburton (1985) (for un-damped structural system with diagonal masses), which requires no matrix factorization. Similar explicit integration procedures using a central difference scheme have been proposed by Belytschko and Mullen (1976) and Noor and Lambiotte (1979) for un-damped structural systems.

(2) Runge-Kutta Method

The classical fourth order Runge-Kutta method has long been popular and is often recommended by mathematicians for accurate numerical computations of solutions of ordinary differential equations. Hull et al. (1972) conducted a comprehensive survey of these methods and others, such as Euler's method. The solution of a discrete system of dynamic equations of motion can be readily accomplished using the Runge-Kutta formulas as

$$\{\mathbf{u}_{t+\Delta t}\} = \{\mathbf{u}_t\} + \Delta t\{\dot{\mathbf{u}}_t\} + \frac{\Delta t^2}{6}\{\mathbf{A}_0 + \mathbf{A}_1 + \mathbf{A}_2\} + O(\Delta t^5) \quad (2-15)$$

$$\{\dot{\mathbf{u}}_{t+\Delta t}\} = \{\dot{\mathbf{u}}_t\} + \frac{\Delta t}{6}\{\mathbf{A}_0 + 2\mathbf{A}_1 + 2\mathbf{A}_2 + \mathbf{A}_3\} + O(\Delta t^5), \quad (2-16)$$

where

$$\mathbf{A}_0 = \{\ddot{\mathbf{u}}(t, \mathbf{u}_t)\} \quad (2-17a)$$

$$\mathbf{A}_1 = \{\ddot{\mathbf{u}}(t + \frac{1}{2}\Delta t, \mathbf{u} + \frac{1}{2}\Delta t\dot{\mathbf{u}}_t)\} \quad (2-17b)$$

$$\mathbf{A}_2 = \{\ddot{\mathbf{u}}(t + \frac{1}{2}\Delta t, \mathbf{u}_t + \frac{1}{2}\Delta t\dot{\mathbf{u}}_t + \frac{1}{4}\Delta t^2\mathbf{A}_0)\} \quad (2-17c)$$

$$\mathbf{A}_3 = \{\ddot{\mathbf{u}}(t + \Delta t, \mathbf{u}_t + \Delta t\dot{\mathbf{u}}_t + \frac{1}{2}\Delta t^2\mathbf{A}_1)\}. \quad (2-17d)$$

The accelerations in Equations (2-17a-d) are calculated by solving dynamic equations using the time and displacement values indicated therein.

The one-step algorithm possesses several desirable features: (1) the scheme is self-starting, (2) the time step may be easily changed, (3) the explicit nature of formulation negates the need for iteration in nonlinear problems, and (4) the method is of a relatively high order and possesses at worst a weak instability.

These desirable features are somewhat offset by the fact that the acceleration vector must be computed four times per time step. The time step must be small in order to get accurate results. Also the computational time required for the solution of a problem by this method may be rather large compared to other numerical integration methods. This method does not have a simple error estimate, but modified methods such as the Runge-Kutta-Fehlberg methods of orders 1 to 3 (Fehlberg, 1970) and an adaptive Runge-Kutta method (Dahlquist and Björck, 1974) with optimized stability properties have been developed and equipped with automatic step control based on local error estimates (Shampine, 1977). Later, Braeklus and Aasen (1981) have compared the performance of these methods with the explicit central difference and implicit average acceleration methods by means of numerical experiments on structural dynamics problems. It should be noted that this method is not to be recommended for solving large-scale problems. Current use of the method was therefore almost completely limited to obtain initial values required for other procedures and in rare instances as an aid in changing the time step in codes using multi-step methods of solution.

(3) *Other Explicit Methods*

A number of other explicit techniques can be used in linear and nonlinear structural problems. These include stiffly stable methods, Predictor-Corrector iterative methods, (predict-evaluate-correct) PEC algorithms, high order Taylor series schemes and others. These methods are briefly described as follows.

- *Stiffly Stable Methods*

The stiffly stable solution procedures had been developed to extend the range of stability of explicit calculation for solving the equations produced by finite element discretization process, in which some frequency components vary rapidly in time while others do not. Jensen (1974) carried out an excellent review on stiffly stable methods. More extension methods were presented by Jensen (1976) and Park (1975).

- *Predictor-Corrector Methods*

The Predictor-Corrector (PC) methods are iterative methods that use the information at one or more previous time points to assist in evaluating the dependent variable at each successive time point. There had been an evolution in PC methods (Henrici, 1968). Humar and Wright (1974) conducted an excellent review of the PC methods used in structural dynamics problems.

- *Explicit-Explicit Subcycling Time Integration Method*

Explicit-explicit subcycling is frequently adopted in dynamic finite element problems in order to improve computational efficiency when the element size varies over the mesh. This is accomplished by separating the elements or nodes into groups and assigning a different time step to each group. The time step of each group depends only on the frequencies of the elements in that group. Belytschko and Mullen (1976) first introduced a mixed time integration method. This was a nodal partition where both implicit and explicit methods were used in the time integration process. Hughes and Liu (1978a, b) later introduced an implicit-explicit method that divided the mesh into groups of elements rather than nodes, known as an element partition. Both methods were able to improve computational efficiency substantially.

- *Taylor Series Schemes*

The methods based on Taylor series, which also include the well-known Runge-Kutta methods, have received extensive treatment in the literature (Park et al., 1977). In fact, for Taylor series methods, the maximum time step is often dictated by the stability limit and not by the accuracy requirements. Both for linear and nonlinear cases the ratio of computation of the Taylor series to that of the central difference scheme increases as the number of terms increases. The only advantage of the Taylor series over the central difference scheme is the improved accuracy at the expense of an increment of computational effort. However, this advantage is not

guaranteed for nonlinear problems (Park, 1975). A Taylor series scheme proposed by Melosh (1975) was

$$\{\mathbf{u}_{t+\Delta t}\} = \sum_{i=0}^n \frac{(\Delta t)^i}{(i)!} \frac{d^i \{\mathbf{u}_t\}}{dt^i}. \quad (2-18)$$

Belytschko and Hesieh (1973) had employed a convected coordinate formulation with an explicit solution operator that relied upon a Taylor series expansion (at time t) to determine the displacements (at time $t + \Delta t$) and an average acceleration during the time step to compute the velocity components.

Belytschko and Lu (1993) studied the explicit multi-time-step (sub-cycling) integration algorithms based on nodal partitions for both first and second order systems. Consistency, convergence and stability analyses of this algorithm for first order systems were conducted. Smolinski and Wu (1998) presented an explicit sub-cycling time integration algorithm based on an element partition for first-order finite element systems. This method uses linear interpolation to compute intermediate values at nodes in the sub-domains integrated with larger time steps.

2.3.2 Implicit Time Integration Methods

Generally, implicit algorithms are more effective for structural dynamics problems. The conventional implicit time integration procedures, Newmark, Wilson- θ , and Houbolt methods are described in this section. Subbaraj and Dokainish (1989) summarized the implicit methods for linear and non-linear problems. Many implicit methods are unconditionally stable for linear analysis and the maximum time step that can be employed is governed by the accuracy of solution and not by the stability of the integration process.

(1) *Newmark Family Methods*

The most widely used family of implicit methods of direct time integration for solving semi-discrete equations of motion is Newmark's family of methods (Newmark, 1959). The Newmark method is based on the following assumptions:

$$\{\dot{\mathbf{u}}_{t+\Delta t}\} = \{\dot{\mathbf{u}}_t\} + \Delta t \cdot \left((1 - \beta_2) \{\ddot{\mathbf{u}}_t\} + \beta_2 \cdot \{\ddot{\mathbf{u}}_{t+\Delta t}\} \right) \quad (2-19a)$$

$$\{\mathbf{u}_{t+\Delta t}\} = \{\mathbf{u}_t\} + \Delta t \{\dot{\mathbf{u}}_t\} + (\Delta t)^2 \left(\left(\frac{1}{2} - \beta_1 \right) \{\ddot{\mathbf{u}}_t\} + \beta_1 \{\ddot{\mathbf{u}}_{t+\Delta t}\} \right) \quad (2-19b)$$

where the parameters β_1 and β_2 determine the stability and accuracy of the algorithm. For $\beta_2 = 1/2$ and $\beta_1 = 1/6$, the relations (2-19a) and (2-19b) correspond to the linear acceleration method. In addition to Equations (2-19a) and (2-19b), for solution of displacements, velocities and accelerations at time $t + \Delta t$:

$$[\mathbf{M}] \{\ddot{\mathbf{u}}_{t+\Delta t}\} + [\mathbf{C}] \{\dot{\mathbf{u}}_{t+\Delta t}\} + [\mathbf{K}] \{\mathbf{u}_{t+\Delta t}\} = \{\mathbf{r}_{t+\Delta t}\} \quad (2-20)$$

The recurrence relation for solving $\{\mathbf{u}_{t+\Delta t}\}$ is:

$$\begin{aligned} & \left([\mathbf{K}] + \frac{\beta_2}{\beta_1 \Delta t} [\mathbf{C}] + \frac{1}{\beta_1 (\Delta t)^2} [\mathbf{M}] \right) \{\mathbf{u}_{t+\Delta t}\} \\ & = \{\mathbf{r}_{t+\Delta t}\} + [\mathbf{C}] \left(\frac{\beta_2}{\beta_1 \Delta t} \{\mathbf{u}_t\} + \left(\frac{\beta_2}{\beta_1} - 1 \right) \{\dot{\mathbf{u}}_t\} + \Delta t \left(\frac{\beta_2}{2\beta_1} - 1 \right) \{\ddot{\mathbf{u}}_t\} \right) \\ & \quad - [\mathbf{M}] \left(\frac{1}{\beta_1 (\Delta t)^2} \{\mathbf{u}_t\} + \frac{1}{\beta_1 \Delta t} \{\dot{\mathbf{u}}_t\} + \left(\frac{1}{2\beta_1} - 1 \right) \{\ddot{\mathbf{u}}_t\} \right) \end{aligned} \quad (2-21)$$

In 1985, Warburton had suggested a direct approach in which the right-hand-side of equation (2-21), so-called the effective load vector, depends only on the displacements at t and $t - \Delta t$. The recurrence relation can be obtained as follows: use the equilibrium equations at times $t - \Delta t$, t , $t + \Delta t$ and Newmark's basic assumptions, equation (2-19a) and (2-19b), at times t , $t - \Delta t$. There are seven equations with six unknowns, $\{\ddot{\mathbf{u}}_{t+\Delta t}\}$, $\{\ddot{\mathbf{u}}_t\}$, $\{\ddot{\mathbf{u}}_{t-\Delta t}\}$, $\{\dot{\mathbf{u}}_{t+\Delta t}\}$, $\{\dot{\mathbf{u}}_t\}$ and $\{\dot{\mathbf{u}}_{t-\Delta t}\}$, which can be eliminated. For $\beta_2 = 1/2$, it is given by the following set of simultaneous equations:

$$\begin{aligned}
& \left([\mathbf{M}] + \frac{\Delta t}{2} [\mathbf{C}] + \beta_1 (\Delta t)^2 [\mathbf{K}] \right) \cdot \{\mathbf{u}_{t+\Delta t}\} \\
& = (\Delta t)^2 (\beta_1 \{\mathbf{r}_{t+\Delta t}\} + (1 - 2\beta_1) \{\mathbf{r}_t\} + \beta_1 \{\mathbf{r}_{t-\Delta t}\}) \\
& + \left(2[\mathbf{M}] - (\Delta t)^2 (1 - 2\beta_1) [\mathbf{K}] \right) \cdot \{\mathbf{u}_t\} - \left([\mathbf{M}] - \frac{\Delta t}{2} [\mathbf{C}] + \beta_1 (\Delta t)^2 [\mathbf{K}] \right) \cdot \{\mathbf{u}_{t-\Delta t}\}
\end{aligned} \tag{2-22}$$

Here, the solution for $\{\mathbf{u}_{t+\Delta t}\}$ requires knowledge of $\{\mathbf{u}_t\}$ and $\{\mathbf{u}_{t-\Delta t}\}$, hence a special starting procedure is necessary.

The Newmark method is unconditionally stable if $\beta_2 \geq 0.5$ and $\beta_1 \geq (2\beta_2 + 1)^2 / 16$. A summary of stability conditions for the Newmark method (Goudreau and Taylor, 1972 and Hughes et al. 1977) is given as following:

Unconditional

$$2\beta_1 \geq \beta_2 \geq 1/2 \tag{2-23}$$

Conditional

$$\beta_2 \geq 1/2, \beta_1 < \beta_2/2, \omega\Delta t \geq \Omega_{crit}$$

$$\text{and } \Omega_{crit} = \frac{\xi(\beta_2 - 1/2) + (\beta_2/2 - \beta_1 + \xi^2(\beta_2 - 1/2)^2)^{1/2}}{(\beta_2/2 - \beta_1)} \tag{2-24}$$

where ω is the undamped frequency of vibration and $\xi = (\alpha/\omega + \beta\omega)/2$ is the damping ratio. α and β are Rayleigh constants of equation, $[\mathbf{C}] = \alpha[\mathbf{M}] + \beta[\mathbf{K}]$. Note that if $\beta_2 = 1/2$, viscous damping has no effect on stability. In Newmark method, $\beta_2 > 1/2$ is necessary to introduce high frequency dissipation.

(2) Wilson- θ Method

The Wilson- θ method (Wilson et al. 1973) is essentially an extension of the linear acceleration method in which a linear variation of acceleration from time t to time $t + \Delta t$ was assumed. In the Wilson- θ method the acceleration is assumed to be linear from time t to time $t + \theta\Delta t$, where $\theta \geq 1.0$. For linear problems, the method is unconditionally stable if $\theta \geq 1.37$ (Wood, 1977), so $\theta = 1.4$ is usually employed.

(3) *Houbolt Method*

In this approach, standard finite difference expressions are used to approximate the acceleration and velocity components in terms of the displacement components. The following backward difference formulae with errors of order $(\Delta t)^2$ are used in the Houbolt integration method:

$$\{\ddot{\mathbf{u}}_{t+\Delta t}\} = \frac{1}{(\Delta t)^2} (2\{\mathbf{u}_{t+\Delta t}\} - 5\{\mathbf{u}_t\} + 4\{\mathbf{u}_{t-\Delta t}\} - \{\mathbf{u}_{t-2\Delta t}\}) \quad (2-25)$$

$$\{\dot{\mathbf{u}}_{t+\Delta t}\} = \frac{1}{6\Delta t} (11\{\mathbf{u}_{t+\Delta t}\} - 18\{\mathbf{u}_t\} + 9\{\mathbf{u}_{t-\Delta t}\} - 2\{\mathbf{u}_{t-2\Delta t}\}) \quad (2-26)$$

In order to obtain the solution at $t + \Delta t$, using this implicit method, equilibrium conditions are considered at time $t + \Delta t$ (not at time t as for the central difference method):

$$[\mathbf{M}]\{\ddot{\mathbf{u}}_{t+\Delta t}\} + [\mathbf{C}]\{\dot{\mathbf{u}}_{t+\Delta t}\} + [\mathbf{K}]\{\mathbf{u}_{t+\Delta t}\} = \{\mathbf{r}_{t+\Delta t}\} \quad (2-27)$$

Substituting Equations (2-25) and (2-26) into (2-27) and rearranging all known vectors to the right hand side, $\{\mathbf{u}_{t+\Delta t}\}$ can be obtained:

$$\begin{aligned} \left(\frac{2}{(\Delta t)^2}[\mathbf{M}] + \frac{1}{6\Delta t}[\mathbf{C}] + [\mathbf{K}] \right) \{\mathbf{u}_{t+\Delta t}\} = \{\mathbf{r}_{t+\Delta t}\} + \left(\frac{5}{(\Delta t)^2}[\mathbf{M}] + \frac{3}{\Delta t}[\mathbf{C}] \right) \{\mathbf{u}_t\} - \\ \left(\frac{4}{(\Delta t)^2}[\mathbf{M}] + \frac{3}{2\Delta t}[\mathbf{C}] \right) \cdot \{\mathbf{u}_{t-\Delta t}\} + \left(\frac{1}{(\Delta t)^2}[\mathbf{M}] + \frac{1}{3\Delta t}[\mathbf{C}] \right) \cdot \{\mathbf{u}_{t-2\Delta t}\} \end{aligned} \quad (2-28)$$

Thus, the solution for $\{\mathbf{u}_{t+\Delta t}\}$ requires knowledge of $\{\mathbf{u}_t\}$, $\{\mathbf{u}_{t-\Delta t}\}$ and $\{\mathbf{u}_{t-2\Delta t}\}$, and so a special starting procedure is necessary. Although the knowledge of $\{\mathbf{u}_0\}$, $\{\dot{\mathbf{u}}_0\}$ and $\{\ddot{\mathbf{u}}_0\}$ is useful to start the Houbolt integration scheme, it is more common and accurate to calculate $\{\mathbf{u}_{\Delta t}\}$ and $\{\mathbf{u}_{2\Delta t}\}$ by some other algorithms such as the explicit central difference scheme with a fraction of Δt as the time step. The most significant drawback of the Houbolt integrator was the algorithm damping which was inherent in the numerical procedure and was introduced into the response when large time steps were used.

(4) *Implicit Algorithms for Non-Linear Problems*

Subbaraj and Dokainish (1989) extended the solution algorithms for linear problems described in the previous section to account for nonlinear behavior. The equivalent internal (nodal) elastic resisting forces (of the continuum or structure), for small displacement and linearly elastic problems, must be replaced by their nonlinear counterparts (involving large deformations and/or the physical behavior of nonlinear materials).

In order to make the displacements and the stresses fully satisfy the nonlinear condition of the problems, it is in general necessary to perform an equilibrium iteration sequence at each time step or pre-selected time steps. Nelson and Mak (1982) studied nonlinear problems by using the Newmark implicit time integration method for elastic-plastic materials with material unloading.

Generally, the explicit integration method is very efficient for one time step computation. However, a very small time step size must be selected to guarantee the stability of integration. Mizukami (1986) had relaxed the time step size restriction for first order ordinary differential equations with semi-discretization of the diffusion equation. In 1989, Neal and Belytschko proposed an explicit-explicit sub-cycling procedure for structural dynamics that does not require integer time step ratios for adjacent groups.

In fact, using different time step for each group is not very convenient. A major disadvantage to all these previous time integration methods is that computational costs of these methods are expensive. The development of high order accurate and efficient algorithms therefore becomes necessary and important.

The problem of computing exponential matrix can be solved efficiently using the precise time-step integration algorithm. The major feature of the precise time integration algorithm is the exact computation of the exponential matrix. A

comprehensive survey about the exponential matrix computation will be given in next section.

2.4 Methods for Solving Exponential Matrix

When the second-order dynamic equations are transformed into the first-order equations, the solution of the dynamic system, $\{\dot{\mathbf{U}}\} = [\mathbf{W}]\{\mathbf{U}\}$, includes an exponential matrix $\exp(\Delta t \cdot [\mathbf{W}])$. This formula can be formally defined by the convergent power series

$$e^{\Delta t[\mathbf{W}]} = [\mathbf{I}] + \Delta t \cdot [\mathbf{W}] + \frac{\Delta t^2 [\mathbf{W}]^2}{2!} + \frac{\Delta t^3 [\mathbf{W}]^3}{3!} + \dots \quad (2-29)$$

In principle, the exponential of a matrix could be computed in many ways. Methods involving approximation theory, differential equations, the matrix eigenvalues, and the matrix characteristic polynomial have been proposed. In practice, consideration of computational stability and efficiency proves that some of the methods are preferable to others, but that none is completely satisfactory. Moler and Loan (1978) gave a very comprehensive review about computing the exponential of a matrix. They analyzed the advantages and disadvantages of many methods to compute the exponential matrix. The effective computation of this matrix function is one of the main topics of this study. Figure 2-1 summarized the most frequently adopted methods for computing exponential matrix, including the algorithms proposed in this research. In the following sections, only the series methods and the 2^N algorithm methods are discussed, which will be employed in this research.

2.4.1 Series Methods

Series methods are standard direct approximation techniques for the exponential function.

(1) *Taylor Series*

The definition is

$$e^{[\mathbf{W}]} = [\mathbf{I}] + [\mathbf{W}] + [\mathbf{W}]^2 / 2! + \dots, \text{ and } \mathbf{T}_n([\mathbf{W}]) = \sum_{j=0}^n [\mathbf{W}]^j / j! \quad (2-30)$$

It should be emphasized that the difficulty is not the truncation at the series, but the truncation of the arithmetic (i.e. round off error). Concern over where to truncate the series is important if efficiency is considered. Liou (1966) considered the truncation error of Taylor series. He mentioned that if δ was some prescribed error tolerance, n should be large enough so that

$$\|\mathbf{T}_n([\mathbf{W}]) - e^{[\mathbf{W}]}\| \leq \left(\frac{\|[\mathbf{W}]\|_2^{n+1}}{(n+1)!} \right) \left(\frac{1}{1 - \|[\mathbf{W}]\|_2 / (n+2)} \right) \leq \delta \quad (2-31)$$

Later, Bickhart (1968) considered the relative error instead of absolute error. Unfortunately, all these approaches ignore the effects of round off error and so must fail in actual computation with certain matrices.

(2) *Padé Series Approximation*

Baker (1975) and Baker & Graves-Morris (1996) gave a basic theory of the Padé approximant. An (p, q) approximant of a function $f(t)$ is written as

$$f(t) \approx \frac{N_p(t)}{Q_q(t)} \quad (2-32)$$

where

$$\begin{aligned} N_p(t) &= a_0 + a_1 t + a_2 t^2 + \dots + a_p t^p, \\ Q_q(t) &= b_0 + b_1 t + b_2 t^2 + \dots + b_q t^q. \end{aligned} \quad (2-33)$$

The coefficients a_p and b_q are determined from the Taylor-series expansion of $f(t)$ at any regular point (with $b_0 = 1$). Without loss of generality, the expansion point is taken as $t = 0$. The Maclaurin series is used for $f(t)$. Let $f(t)$ have the series expression as

$$f(t) = \sum_{i=0}^{\infty} c_i t^i \quad (2-34)$$

where c_i is known. In the development that follows the computations are simplified by taking the expansion about 0, instead of around another point. The restriction to the Maclaurin series does not represent any significant loss of generality. In equation (2-33), let $b_0 = 1$ and consider the difference

$$\frac{N_p(t)}{Q_q(t)} + O(\alpha^{p+q+1}) = \sum_{i=0}^{\infty} c_i t^i. \quad (2-35)$$

The coefficients of $t^{p+1}, t^{p+2}, \dots, t^{p+q}$ directly yield the coefficients

$$\begin{pmatrix} b_q \\ b_{q-1} \\ \vdots \\ b_1 \end{pmatrix} = - \begin{pmatrix} c_{p-q+1} & c_{p-q+2} & \cdots & c_p \\ c_{p-q+2} & c_{p-q+3} & \cdots & c_{p+1} \\ \vdots & \vdots & \ddots & \vdots \\ c_p & c_{p+1} & \cdots & c_{p+q+1} \end{pmatrix}^{-1} \begin{pmatrix} c_{p+1} \\ c_{p+2} \\ \vdots \\ c_{p+q} \end{pmatrix} \quad (2-36)$$

where $c_i \equiv 0$ for $i < 0$. If the inverse matrix of c_i elements does not exist then the particular (p, q) approximant is not defined. Once b_i are found, then the remaining equations (t^0, \dots, t^p) yield the a_i coefficients.

Druskin and Moskov (1991) considered the following one-dimensional problem

$$\frac{d^2 u}{dx^2} - \lambda u = 0 \quad (2-37)$$

On the interval $[0, 1]$, with boundary conditions

$$-\left. \frac{du}{dx} \right|_{x=0} = 1 \text{ and } u(1) = 0 \quad (2-38)$$

They discussed two types of the Padé approximants.

(1) Simple Padé Approximant:

$$\frac{d^i}{d\lambda^i} [f_k(\lambda) - f(\lambda)]_{\lambda=b} = 0, \quad i = 0, \dots, 2k-1 \quad (2-39)$$

for some b outside the spectrum of (2-37).

(2) Multipoint Padé Approximant.

$$[f_k(\lambda) - f(\lambda)]_{\lambda=b_i} = 0, \quad i = 1, \dots, 2k \quad (2-40)$$

where $b_1 < b_2 < \dots < b_{2k}$ are outside of the spectrum of (2-37).

Wragg and Davies (1975) considered the advantages of various representations of these rational approximations as well as the choice of p and q to obtain the prescribed accuracy. In Chapters 5 and 6, the application of the Padé approximation will be discussed in details.

2.4.2 Scaling and Squaring Techniques

The round off error difficulty and the computing cost of the Taylor and Padé approximants increase as the matrix norm, $\|\Delta t \cdot [\mathbf{W}]\|_2$, increase. Exploiting a fundamental property of the exponential function can control these difficulties:

$$e^{\Delta t[\mathbf{W}]} = (e^{\Delta t[\mathbf{W}]/m_1})^{m_1} \quad (2-41)$$

Here $m_1 = 2^N$ is the scaling factor. The idea is to choose m_1 for which $e^{\Delta t[\mathbf{W}]/m_1}$ can be reliably and efficiently computed, and then to compute the matrix $(e^{\Delta t[\mathbf{W}]/m_1})^{m_1}$ by repeated squaring. One commonly used criterion for choosing N is to make it the smallest power of two with which $\|[\mathbf{W}]\|_2 / m_1 \leq 1$. Under this restriction, the exponential matrix, $e^{\Delta t[\mathbf{W}]/m_1}$, can be satisfactorily computed by using either the Taylor or Padé series approximation.

2.4.3 Efficiency of Various Methods

According to the study of Moler and Loan (1978), the polynomial methods are not

satisfactory. Some of them require the characteristic polynomial. The splitting methods are largely speculative and probably only of interest in special settings. Specializations of ODE methods for the $e^{[W]}$ problem have not yet been implemented. The best method would appear to involve a variable order, variable step difference scheme. This method would be stable and reliable but expensive.

The only generally competitive method is the scaling and squaring method of series algorithm. This method required much work. Therefore, more efficient and precise algorithms for computing exponential matrix become very important.

2.5 Precise Time-Step Integration Methods

Zhong and Williams (1994) presented a precise time step integration method for a linear-invariant structural dynamic system. The main feature of precise time step integration (PTI) method is the precise computation of the exponential matrix.

Later, Lin et al. (1995) extended this method to tackle sinusoidal loading and more general loading forms by Fourier series. Shen et al. (1995) improved the efficiency of the method by parallel computing method. A mixed fine grain and coarse grain strategy for parallel computing had been developed.

In 1996, Zhong et al. discussed a class of sub-domain precise tie integrations. This algorithm is explicit in nature and unconditionally stable for the single-point sub-domain integration. But most the discussions are limited to one-dimensional cases.

Fung (1997) proposed a precise time-step integration method by step-response and impulsive-response matrices for dynamic problems. The second-order differential equations were manipulated directly without transforming to first order equations. By considering the symmetry of the matrices, the computational cost could be reduced by half.

Chen et al. (2001) proposed the precise time-step integration for the numerical analysis of nonlinear heat conduction. The symmetry of matrix exponential of precise time integration in heat conduction problem is proved and used in sub-domain integration to reduce computational expense in great deal. Zhu and Law (2001) studied the dynamic response of a continuous beam under moving loads by precise time-step integration method. Zhong (2001) extended the precise time-step integration method to solve the asymmetric Riccati differential equations.

To avoid computing the particular solutions separately, the dimensional expanding method is developed. Gu et al. (2001) presented a new algorithm to convert non-homogeneous 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. This method expended the applicability of the PTI method, but the computational cost will be increased. Wang et al. (2002) also investigated the dimensional expanding method by using another expression. More details will be given in Chapter 3.

2.6 Krylov Subspace Method

The Krylov subspace method have become a very useful and popular tool for solving large sets of linear and nonlinear equations, and large eigenvalue problems. One of the reasons for their popularity is their simplicity and generality. These methods have been increasingly accepted as efficient and reliable alternative to the more expensive methods that are usually employed for solving dense problems. This trend is likely to accelerate as models are becoming more complex and give rise to large matrix problems.

For the generality of the Krylov subspace method, this method is studied for solving the dynamic problems by the time integration method. The good property of Krylov subspace method for solving the large sparse linear systems can be used to

get the solutions. Chapter 6 will give more details about the Krylov subspace method.

2.6.1 Background of Krylov Subspace Method

Methods based on orthogonalization were developed by a number of authors in the early '50s. Lanczos' method (Lanczos, 1950) was based on two mutually orthogonal vector sequences, and his motivation came from eigenvalue problems. Lanczos later applied his method to solve linear systems, in particular, the symmetric ones. An important property for proving convergence of the method when solving linear systems is that the iterations are related to the initial residual by multiplication with a polynomial in the coefficient matrix.

The research by Arnoldi (1951) further discussed the Lanczos biorthogonalization method, but it also presented a new method, combining features of the Lanczos method and the method developed by Hestenes and Stiefel (1952).

Several methods have been developed in later years that employ, most often implicitly, the upper Hessenberg matrix of the Arnoldi method. The overview and characterization of these orthogonal projection methods for non-symmetric systems is in the research of Ashby et al., (1990) and Jea and Young (1980).

2.6.2 Application of Krylov Subspace Method

The Krylov subspace method can be used to solve various problems, such as linear system, eigenvalue problems, parabolic equations, Markov processes, and exponential matrix. Saad (1982) studied the Krylov subspace method for solving linear systems and eigenvalue problems. Function approximation and solving large

sparse linear equations are the most important applications of the Krylov subspace method.

Firstly, the desirable property of approximation to the exponential function multiplied a vector is investigated in this research. For parabolic equations, an ideal one-step method would consist of a scheme as:

$$\{\mathbf{U}(t)\} = \mathbf{T}(\Delta t)(\{\mathbf{U}(t_0)\} - \{\mathbf{U}_p(t_0)\}) + \{\mathbf{U}_p(t)\} \quad (2-42)$$

where $\mathbf{T}(\Delta t) = \exp(\Delta t \cdot [\mathbf{W}])$, $\{\mathbf{U}_p(t)\}$ is the particular solution of non-homogeneous equation. Hence the basic operation in the above formula is the computation of the exponential of a given matrix times a vector. It should be noted that there is no need to evaluate the matrix exponential $\exp(\Delta t \cdot [\mathbf{W}])$ but its product with a given vector.

Hochbruck and Lubich (1997) studied the Krylov subspace methods for the approximation of $\exp(\Delta t \cdot [\mathbf{W}])\{\mathbf{U}\}$ when $[\mathbf{W}]$ was a matrix of large dimension; $\{\mathbf{U}\}$ was a given vector. They obtained error bounds for Galerkin-type Krylov methods for linear equations, namely, the bi-conjugate gradient method and the full orthogonalization method. The Krylov subspace method can work well for computing $\exp(\Delta t \cdot [\mathbf{W}])$. Chapters 6 and 7 present more details about the applications of Krylov subspace method.

The Krylov subspace method has been very successful in solving large eigenvalue problems (Saad, 1981), parabolic equations (Gallopoulos and Saad, 1992), matrix exponential operator problems (Hochbruck and Lubich, 1997), reduced-order modelling in circuit simulation problems (Freund, 2000), and engineering analysis (Roland, 2000; Bai, 2002). Later, the Krylov subspace method was developed to solve large linear systems (Bergamaschi and Vianello, 2000). One of the reasons for their popularity is their simplicity and their generality. This method has been increasingly adopted as efficient and reliable alternative to the more complicated and expensive methods.

Secondly, many efficient iterative methods are presented based on the Krylov subspace method to solve the large linear systems. The CG (conjugate gradients) method was introduced in 1954 by Hestenes and Stiefel. But it did not get much attention, until Reid (1971) showed how this method can be used to our advantage for certain systems. Since then there has been a growing interest in the CG method mainly for its use as an acceleration technique. GMRES (Generalized Minimal Residual) and variants use orthogonal basis. Algorithms of this type were proposed by Young and Jea (1980), Saad (1981), Elman (1982), Axelsson (1977) and others. The GMRES algorithm of Saad and Schultz (1986) is the most popular one among this type. In 1950, Lanczos used bi-orthogonality relations to reduce iteratively a matrix to tri-diagonal form and suggested how to solve non-symmetric linear systems from the reduced tri-diagonal form. Later, his ideas were adopted for a variety of methods, such as Bi-CG (Fletcher, 1975), QMR (Quasi Minimal Residual) (Freund and Nachtigal, 1991) and others. In 1989, Sonneveld introduced a “squared Bi-CG algorithm”. The steps in the GMRES algorithms are increasingly expensive while the algorithms of Bi-CG type often converge slowly and are sensitive to evaluation errors. This field remains open and further investigations are imperative.

2.7 Concluding Remarks

A comprehensive review of time integration algorithms for solving dynamic equations is presented in this chapter. Most popular methods of explicit and implicit time integration methods are briefly introduced. The focus of this research is on the direct explicit precise time step integration algorithms. In later chapters, dimensional expanding method, Padé series approximation and Krylov subspace method are studied in-depth to improve the precise time-step integration methods.

Table 2-1 Comparison of explicit and implicit time integration method

Method	Explicit	Implicit
Equilibrium step	t	$t + \Delta t$
Tangent stiffness	Not required	Required
Stability	Conditional	Unconditional
Accuracy	Depends on Δt	Depends on Δt
Efficiency	For regular mesh with lumped mass and damping	For irregular mesh with short period (stiff) structures
Example	Central difference	Newmark
Typical applications	Soil or fluid media in interaction problems	Most structural and interaction problems

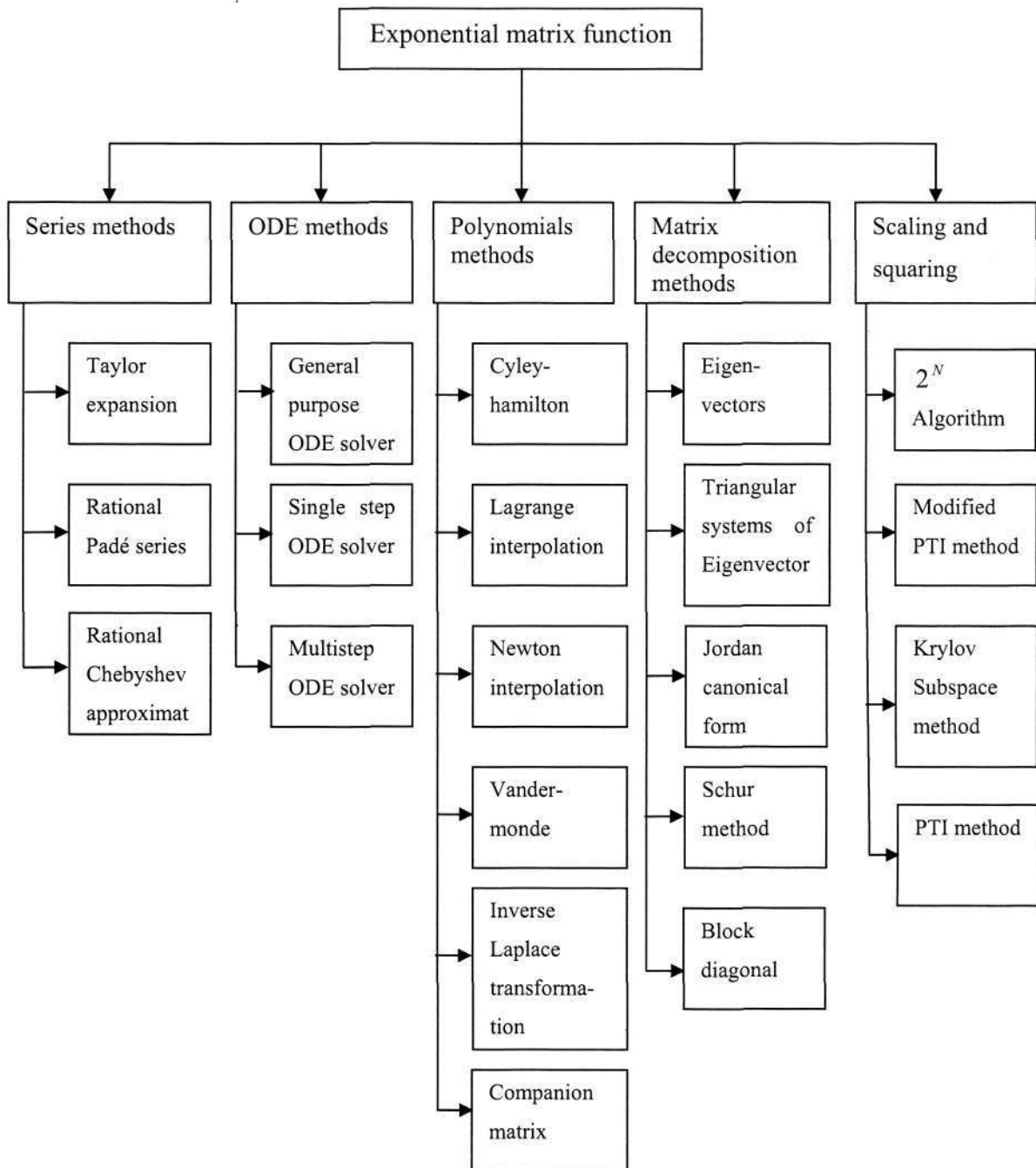


Figure 2-1 Methods for computing the exponential matrix function

CHAPTER THREE

Precise Time-Step Integration Method

Overview

In this chapter, the original precise time-step integration algorithm is further studied. The dimensional expanding method is employed to improve the efficiency of the precise time-step integration algorithms. Later, the dimensional expanding method is extended so that it can be used in the algorithms for solving both the first-order and the second-order equations directly. The extended dimensional expanding method can be used in the algorithms for solving the dynamic equation directly when the excitation is described by first-order and second-order differential equation. A numerical example is used to investigate the accuracy and efficiency.

3.1 Basic Theory of Precise Time-Step Integration Method

The precise numerical time-step integration method was presented for a linear-invariant structural dynamic system. The main feature of the precise time-step integration method is the precise computation of the exponential matrix. The precise time-step integration method is briefly introduced below. For simple, the original precise time-step integration method (PTI) is denoted as Algorithm 1 (A1).

The motion equations of a discrete structural model can be written as

$$[\mathbf{M}]\{\ddot{\mathbf{u}}(t)\} + [\mathbf{C}]\{\dot{\mathbf{u}}(t)\} + [\mathbf{K}]\{\mathbf{u}(t)\} = \{\mathbf{r}(t)\} \quad (3-1)$$

where $[\mathbf{M}]$, $[\mathbf{C}]$ and $[\mathbf{K}]$ are time-invariant mass, damping and stiffness matrices, respectively, $\{\mathbf{r}(t)\}$ is the known external force vector. $\{\mathbf{u}(t)\}$ is the unknown displacement vector. The initial conditions at $t=0$ are given as $\{\mathbf{u}(0)\}=\{\mathbf{u}_0\}$ and $\{\dot{\mathbf{u}}(0)\}=\{\dot{\mathbf{u}}_0\}=\{\mathbf{v}_0\}$. Normally, $[\mathbf{M}]$ is symmetric and positive definite while $[\mathbf{C}]$ and $[\mathbf{K}]$ are symmetric and semi-positive definite. Let

$$\{\mathbf{x}_1\} = [\mathbf{M}]\{\dot{\mathbf{u}}\} + [\mathbf{C}]\{\mathbf{u}\}/2 \text{ or } \{\dot{\mathbf{u}}\} = [\mathbf{M}]^{-1}\{\mathbf{x}_1\} - [\mathbf{M}]^{-1}[\mathbf{C}]\{\mathbf{u}\}/2 \quad (3-2)$$

So that Equation (3-1) becomes

$$\{\dot{\mathbf{x}}_1\} = -([\mathbf{K}] - [\mathbf{C}][\mathbf{M}]^{-1}[\mathbf{C}]/4)\{\mathbf{u}\} - [\mathbf{C}][\mathbf{M}]^{-1}\{\mathbf{x}_1\}/2 + \{\mathbf{r}\} \quad (3-3)$$

Equations (3-2) and (3-3) can be combined to give

$$\begin{Bmatrix} \dot{\mathbf{u}} \\ \dot{\mathbf{x}}_1 \end{Bmatrix} = \begin{bmatrix} -\mathbf{M}^{-1}\mathbf{C}/2 & \mathbf{M}^{-1} \\ \mathbf{C}\mathbf{M}^{-1}\mathbf{C}/4 - \mathbf{K} & -\mathbf{C}\mathbf{M}^{-1}/2 \end{bmatrix} \begin{Bmatrix} \mathbf{u} \\ \mathbf{x}_1 \end{Bmatrix} + \begin{Bmatrix} \mathbf{0} \\ \mathbf{r} \end{Bmatrix} \quad (3-4)$$

And this formula can be rewritten by matrix form:

$$\{\dot{\mathbf{y}}\} = [\mathbf{W}_1]\{\mathbf{y}\} + \{\tilde{\mathbf{r}}\} \quad (3-5)$$

The general solution $\{\mathbf{y}\}$ to Equation (3-5) can be written as

$$\{\mathbf{y}(t)\} = \exp([\mathbf{W}_1] \cdot t) \cdot (\{\mathbf{y}_0\} - \{\mathbf{y}_p(0)\}) + \{\mathbf{y}_p(t)\} \quad (3-6)$$

where $\{\mathbf{y}_0\}$ is the given initial condition and $\{\mathbf{y}_p\}$ is the particular solution depending on the excitation $\{\tilde{\mathbf{r}}\}$. The exponential matrix $\exp([\mathbf{W}_1] \cdot t)$ can be expressed in Taylor series as

$$\exp([\mathbf{W}_1] \cdot t) = \sum_{i=0}^{\infty} \frac{t^i}{i!} [\mathbf{W}_1]^i = [\mathbf{I}] + [\mathbf{W}_1] \cdot t + \frac{1}{2} [\mathbf{W}_1]^2 \cdot t^2 + \cdots + \frac{1}{k!} [\mathbf{W}_1]^k \cdot t^k + \cdots \quad (3-7)$$

It is required to compute the exponential matrix $\exp([\mathbf{W}_1] \cdot t)$ at $t = \Delta t$ accurately. In the precise time-step integration algorithm, the following property of exponential function is used in the computation:

$$[\mathbf{T}(\Delta t)] = \exp([\mathbf{W}_1] \cdot \Delta t) = \{\exp([\mathbf{W}_1] \cdot \tau)\}^{2^N} \quad (3-8)$$

where $\tau = \Delta t / 2^N$, and N is an integer. In the original precise time-step integration method (**A1**), $N = N_{PTI} = 20$ is recommended (Zhong and Williams, 1994). Since the time interval Δt is not large, τ will be an extremely small. The following truncated Taylor series expansion can be used to evaluate $\exp([\mathbf{W}_1] \cdot \tau)$

$$\exp([\mathbf{W}_1] \times \tau) \cong [\mathbf{I}] + [\mathbf{T}_a] \quad (3-9)$$

where

$$[\mathbf{T}_a] = ([\mathbf{W}_1] \times \tau) + ([\mathbf{W}_1] \times \tau)^2 \times \frac{[\mathbf{I}] + ([\mathbf{W}_1] \times \tau) / 3 + ([\mathbf{W}_1] \times \tau)^2 / 12}{2} \quad (3-10)$$

In the actual computation, $[\mathbf{T}_a]$ is computed and stored to avoid unnecessary truncation errors.

Using Equations (3-9) and (3-10), $[\mathbf{T}]$ can be factorized as

$$[\mathbf{T}] \cong ([\mathbf{I}] + [\mathbf{T}_a])^{2^{N_{PTI}}} = ([\mathbf{I}] + [\mathbf{T}_a])^{2^{(N_{PTI}-1)}} \times ([\mathbf{I}] + [\mathbf{T}_a])^{2^{(N_{PTI}-1)}} \quad (3-11)$$

This is also known as the scaling and squaring technique. The matrix $[\mathbf{T}]$ can be computed recursively by starting from Equation (3-10) and then executing the following instruction:

$$\text{for } (i=0; i < N_{PTI}; i++) [\mathbf{T}_a] = 2 \times [\mathbf{T}_a] + [\mathbf{T}_a] \times [\mathbf{T}_a] \quad (3-12)$$

$$\text{Then, } [\mathbf{T}] = ([\mathbf{I}] + [\mathbf{T}_a]) \quad (3-13)$$

Equations (3-10), (3-12) and (3-13) can be used to compute the exponential matrix accurately.

Obviously, no matter how large the step size Δt may be, exact solutions will be produced provided that the matrix $[\mathbf{T}(\Delta t)]$ has been accurately generated. The time step of the special explicit direct integration scheme was not constrained by any of the natural periods of the discrete structure.

3.2 Extension of Precise Time-Step Integration Method

As mentioned in Section 3.1, the main feature of the precise time-step integration method (PTI) is the precise computation of the exponential matrix. The PTI method proposed for a linear time invariant dynamic system can give the accurate numerical results at the time integration points. However, difficulties arise when the algorithm is used for non-homogeneous dynamic systems due to the inverse matrix calculation required in Equation (3-5). The dimensional expanding methods are introduced to overcome the disadvantage.

3.2.1 Dimensional Expanding Methods

From Equation (3-8), the form of particular solution $\{y_p\}$ is very complex and is not easy to get. In the original precise time-step integration method, the following linearly varying loading form is considered,

$$\{r(t)\} = \{r_0\} + \{r_1\} \times (t - t_k) \text{ for } t \text{ between } (t_k, t_{k+1}) \quad (3-14)$$

where $\{r_0\}$ and $\{r_1\}$ are time-invariant vectors. The particular solution $\{y_p\}$ in Equation (3-8) can be expressed as

$$\{y_p(t)\} = ([W_1]^{-1} + [I] \times t) \cdot (-[W_1]^{-1} \cdot \{r_1\}) - [W_1]^{-1} \cdot (\{r_0\} - \{r_1\} \times t_k) \quad (3-15)$$

It can be seen that the particular solution involves the computation of matrix inversion. Normally, the particular solutions arising from the excitation have to be computed separately.

To avoid computing the particular solutions directly, Gu et al. (2001) transformed the governing equations into an equivalent homogenous form by expanding the dimensions of the problems. However, the computational efficiency is found to be low because the size of the computing matrices will increase by about 50%. Hence, the computational expense cost will increase tremendously if no special technique is employed. The PTI method combined with this dimensional expanding method was denoted as Algorithm 2 (A2).

Later, Wang et al.(2002) developed another kind of dimensional expanding method where the excitation is separated into two parts, normally, a constant part and a time-varying part. The dimension of the expanded matrices is controlled by the time-varying part of the excitation. The increase of dimension is usually small and the computational effort is lower than the computational effort required in computing the particular solution. This dimensional expanding method combined with PTI algorithm is denoted as algorithm 3 (A3).

In fact, the dimensional expanding method can be extended to tackle more general and complex excitation in first-order, second-order or even higher-order equation. Without reducing the order of the equation, the second-order even high-order equation all can be solved directly in theoretically. The first and second-order formulae are discussed in this research. The excitations could be described by first-order or second-order differential equations, i.e. $\{\mathbf{r}(t)\} = [\mathbf{f}]\{\mathbf{Z}(t)\}$ where $\{\mathbf{Z}(t)\}$ is the solution of a system of first-order equation $\{\dot{\mathbf{Z}}(t)\} = [\mathbf{S}_1]\{\mathbf{Z}(t)\}$ or second-order equation $\{\ddot{\mathbf{Z}}(t)\} = [\mathbf{S}_2]\{\mathbf{Z}(t)\}$.

In the following, consider the general excitation in the form

$$\{\mathbf{r}(t)\} = e^{\eta t} \cdot \sum_{i=1}^g [\{\mathbf{a}_i\}t^{i-1} \sin(c_i t) + \{\mathbf{b}_i\}t^{i-1} \cos(c_i t)] \quad (3-16)$$

Equation (3-16) treats many loading forms, for example, linear, polynomial, sinusoidal, Fourier as special cases, These forms can be obtained from the general loading form by adjusting the coefficients $(\eta, c_i, \{\mathbf{a}_i\}, \{\mathbf{b}_i\})$.

The external loading form Equation (3-16) can be written as:

$$\begin{aligned} \{\mathbf{r}(t)\} = & \{\mathbf{a}_1\} \cdot e^{\eta t} \cdot \sin(c_1 t) + \{\mathbf{b}_1\} \cdot e^{\eta t} \cdot \cos(c_1 t) + \{\mathbf{a}_2\} \cdot e^{\eta t} \cdot t \cdot \sin(c_2 t) + \\ & \{\mathbf{b}_2\} \cdot e^{\eta t} \cdot t \cdot \cos(c_2 t) + \dots + \{\mathbf{a}_g\} \cdot e^{\eta t} \cdot t^{g-1} \sin(c_g t) + \{\mathbf{b}_g\} \cdot e^{\eta t} \cdot t^{g-1} \cos(c_g t) \end{aligned} \quad (3-17)$$

Let

$$z_{i1} = e^{\eta t} \cdot t^{i-1} \cdot \sin(c_i t) \text{ and } z_{i2} = e^{\eta t} \cdot t^{i-1} \cdot \cos(c_i t), \quad i=1, 2, \dots, g \quad (3-18)$$

Then,

$$[\mathbf{f}] = [\{\mathbf{r}_0\} \ \{\mathbf{r}_1\}], \ \{\mathbf{Z}(t)\} = \{1 \ t\}^T \quad (3-24)$$

$$[\mathbf{S}_1] = \begin{bmatrix} 0 & 0 \\ 1 & 0 \end{bmatrix}, \ [\mathbf{S}_2] = [\mathbf{0}] \quad (3-25)$$

(2) Polynomial Excitation

$$\{\mathbf{r}(t)\} = \{\mathbf{r}_0\} + \{\mathbf{r}_1\}t + \dots + \{\mathbf{r}_{g-2}\}t^{g-2} + \{\mathbf{r}_{g-1}\}t^{g-1} \quad (3-26)$$

where

$$[\mathbf{f}] = [\{\mathbf{r}_0\} \ \{\mathbf{r}_1\} \ \dots \ \{\mathbf{r}_{g-1}\}], \ \{\mathbf{Z}(t)\} = \{1 \ t \ t^2 \ \dots \ t^{g-1}\}^T \quad (3-27)$$

$$[\mathbf{S}_1] = \begin{bmatrix} 0 & 0 & \dots & 0 & 0 \\ 1 & 0 & 0 & \dots & 0 \\ 0 & 2 & 0 & \ddots & \vdots \\ \vdots & 0 & \ddots & \ddots & 0 \\ 0 & 0 & 0 & g-1 & 0 \end{bmatrix}, \quad (3-28)$$

$$[\mathbf{S}_2] = \begin{bmatrix} 0 & 0 & \dots & & 0 & 0 \\ 0 & 0 & \dots & & 0 & 0 \\ 2 \cdot 1 & 0 & 0 & & \vdots & \vdots \\ 0 & 3 \cdot 2 & 0 & \ddots & & \\ \vdots & \ddots & \ddots & \ddots & & 0 & 0 \\ 0 & \dots & 0 & (g-1) \cdot (g-2) & 0 & 0 \end{bmatrix} \quad (3-29)$$

(3) Fourier Representation

$$\{\mathbf{r}(t)\} = \{\mathbf{r}_0\} + \sum_{i=1}^g (\{\mathbf{r}_i^c\} \cos(c_i t) + \{\mathbf{r}_i^s\} \sin(c_i t)) \quad (3-30)$$

where

$$[\mathbf{f}] = [\{\mathbf{r}_0\} \ \{\mathbf{r}_1^c\} \ \{\mathbf{r}_1^s\} \ \{\mathbf{r}_2^c\} \ \{\mathbf{r}_2^s\} \ \dots \ \{\mathbf{r}_g^c\} \ \{\mathbf{r}_g^s\}] \quad (3-31)$$

$$\{\mathbf{Z}(t)\} = \{1 \ \cos(c_1 t) \ \sin(c_1 t) \ \cos(c_2 t) \ \sin(c_2 t) \ \dots \ \cos(c_g t) \ \sin(c_g t)\}^T \quad (3-32)$$

In other way, the second-order equation can be solved directly without reducing the order of the equation. Firstly, the excitation described by first-order differential equation is considered.

$$\begin{bmatrix} \mathbf{M} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} \begin{Bmatrix} \ddot{\mathbf{u}}(t) \\ \dot{\mathbf{Z}}(t) \end{Bmatrix} + \begin{bmatrix} \mathbf{C} & \mathbf{0} \\ \mathbf{0} & \mathbf{I} \end{bmatrix} \begin{Bmatrix} \dot{\mathbf{u}}(t) \\ \mathbf{Z}(t) \end{Bmatrix} + \begin{bmatrix} \mathbf{K} & -\mathbf{f} \\ \mathbf{0} & \mathbf{S}_1 \end{bmatrix} \begin{Bmatrix} \mathbf{u}(t) \\ \mathbf{Z}(t) \end{Bmatrix} = \mathbf{0} \quad (3-37)$$

$$\{\dot{\mathbf{Z}}(t)\} = [\mathbf{S}_1] \{\mathbf{Z}(t)\} \quad (3-38)$$

Secondly, the excitation described by second-order differential equation as

$$\begin{bmatrix} \mathbf{M} & \mathbf{0} \\ \mathbf{0} & \mathbf{I} \end{bmatrix} \begin{Bmatrix} \ddot{\mathbf{u}}(t) \\ \ddot{\mathbf{Z}}(t) \end{Bmatrix} + \begin{bmatrix} \mathbf{C} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} \begin{Bmatrix} \dot{\mathbf{u}}(t) \\ \dot{\mathbf{Z}}(t) \end{Bmatrix} + \begin{bmatrix} \mathbf{K} & -\mathbf{f} \\ \mathbf{0} & \mathbf{S}_2 \end{bmatrix} \begin{Bmatrix} \mathbf{u}(t) \\ \mathbf{Z}(t) \end{Bmatrix} = \mathbf{0} \quad (3-39)$$

$$\{\ddot{\mathbf{Z}}(t)\} = [\mathbf{S}_2] \{\mathbf{Z}(t)\} \quad (3-40)$$

The details of the computing matrices $[\mathbf{S}_1]$ and $[\mathbf{S}_2]$ can be found in Equations (3-22a-b). In this chapter and Chapter 6, Equation (3-36) is employed. Equations (3-37) and (3-39) will be discussed in Chapter 4 and Chapter 5.

3.3 Operation Counts

The three algorithms described in this chapter have the similar computing process structures. The solutions for the methods **(A1, A2, and A3)** respectively can be written as

$$\{\mathbf{U}_{k+1}\} = \exp([\mathbf{W}_1] \cdot \Delta t)_{2N_s \times 2N_s} \cdot \{\mathbf{U}_k\} + \int \exp([\mathbf{W}_1] \cdot (t-s)) \cdot \{\tilde{\mathbf{r}}(t_k + s)\} ds \quad (3-41)$$

$$\{\tilde{\mathbf{U}}_{k+1}\} = \exp([\mathbf{W}_2] \cdot \Delta t)_{3N_s \times 3N_s} \cdot \{\tilde{\mathbf{U}}_k\} \quad (3-42)$$

and

$$\{\mathbf{U}_{k+1}^*\} = \exp([\mathbf{W}_3] \cdot \Delta t)_{(2N_s+g) \times (2N_s+g)} \cdot \{\mathbf{U}_k^*\} \quad (3-43)$$

where

$$[\mathbf{W}_2] = \begin{bmatrix} \mathbf{0} & \mathbf{I} & \mathbf{0} \\ \mathbf{A} & \mathbf{B} & \mathbf{M}^{-1} \\ \mathbf{0} & \mathbf{0} & \mathbf{S}_0 \end{bmatrix}, [\mathbf{W}_3] = \begin{bmatrix} \mathbf{0} & \mathbf{I} & \mathbf{0} \\ \mathbf{A} & \mathbf{B} & \mathbf{M}^{-1}\mathbf{f} \\ \mathbf{0} & \mathbf{0} & \mathbf{S}_1 \end{bmatrix}, \quad (3-44)$$

$$\{\dot{\mathbf{r}}(t)\} = [\mathbf{S}_0]\{\mathbf{r}(t)\}, \{\tilde{\mathbf{U}}\} = \begin{Bmatrix} \mathbf{u}(t) \\ \mathbf{v}(t) \\ \mathbf{r}(t) \end{Bmatrix}, \text{ and } \{\mathbf{U}^*\} = \begin{Bmatrix} \mathbf{u}(t) \\ \mathbf{v}(t) \\ \mathbf{Z}(t) \end{Bmatrix} \quad (3-45)$$

N_s means the dimension of the system. According to the analysis in Section 3.2, the method (A2) will be more expensive than the method (A1) normally. Hence, we just consider the comparison of the computational expense of methods (A1 and A3). For one step, the computational costs in Equations (3-41) and (3-43) are given by

(1) Method A1

Category	Operation counts
Form $[\mathbf{T}_a]$	$2(2N_s)^3$
Form $[\mathbf{T}]$	$N \cdot (2N_s)^3$
$[\mathbf{T}]\{\mathbf{U}\}$	$(2N_s)^2$
Integration part	$[Inte]$

(2) Method A3

Category	Operation counts
Form $[\mathbf{T}_a^*]$	$2(2N_s + g)^3$
Form $[\mathbf{T}^*]$	$N \cdot (2N_s + g)^3$
$[\mathbf{T}^*]\{\mathbf{U}^*\}$	$(2N_s + g)^2$

The dimensional expanding precise time-step integration algorithm (A3) will be more efficient than the original precise time-step integration algorithm (A1) if

$$\begin{aligned} & 2(2N_s)^3 + N \cdot (2N_s)^3 + (2N_s)^2 + [Inte] \\ & > 2(2N_s + g)^3 + N \cdot (2N_s + g)^3 + (2N_s + g)^2 \end{aligned} \quad (3-46)$$

Equation (3-46) can be simplified as

$$[Inte] > (N + 2)(12N_s^2g + 6N_s g^2 + g^3) + (4N_s g + g^2) \quad (3-47)$$

Since the value of g is defined by the external loading form, g in general is not very large and Equation (3-47) can be satisfied easily. In other words, methods with dimensional expanding are in general more efficient.

For example, if the external load is assumed to be linear as in the original PTI method (Zhong and Williams, 1994), $g=2$, $N = N_{PTI} = 20$, and $[Inte] = (2N_s)^3 + 2(2N_s)^2$. The dimensional expanding methods are more efficient if

$$(2N_s)^3 + 2(2N_s)^2 > (N + 2)(12N_s^2g + 6N_s g^2 + g^3) + (4N_s g + g^2) \quad (3-48)$$

Then,

$$\begin{aligned} N_s + 1 &> \left(33g + \frac{33}{2}g \cdot \frac{g}{N_s} + \frac{11}{4}g \cdot \frac{g^2}{N_s^2} \right) + \left(\frac{1}{2} \frac{g}{N_s} + \frac{1}{8} \frac{g^2}{N_s^2} \right) \\ &> \left(33 \frac{g}{N_s} + \frac{33}{2} \left(\frac{g}{N_s} \right)^2 + \frac{11}{4} \left(\frac{g}{N_s} \right)^3 \right) \cdot N_s + \left(\frac{1}{2} \frac{g}{N_s} + \frac{1}{8} \left(\frac{g}{N_s} \right)^2 \right) \end{aligned} \quad (3-49)$$

If $1 > 33 \frac{g}{N_s} + \frac{33}{2} \left(\frac{g}{N_s} \right)^2 + \frac{11}{4} \left(\frac{g}{N_s} \right)^3$, Equation (3-49) will be satisfied after some simplification.

$$\frac{g}{N_s} < \frac{1}{11} \sqrt[3]{11132} - 2 \approx 0.0299 \quad (3-50)$$

or $N_s > g/0.0299$. Hence, if $N_s > 66.9$, the precise time-step integration with dimensional expanding (**A3**) is more efficient than the original precise time-step integration method (**A1**) when the external loading form is linear.

Normally, the method (**A3**) is more effective than the method (**A1**) if Equation (3-47) is satisfied. The value of g is often much smaller than that of N_s according to the analysis in Subsection 3.2.1. Since the value of g is defined by the external loading form and g is not very high in general, Equation (3-47) is satisfied for most cases. The computational cost of the method (**A3**) will be much less than that of methods (**A1** and **A2**).

3.4 Derived Algorithms Analysis

The precise time-step integration method is more general and useful than the other established direct time integration method. Zhu and Law (2001) have investigated that the precise method is better than Newmark method to solve moving load problems. Accurate solutions can be derived in terms of these classic integration algorithms, but the computational expense cost is considerable. The value of Δt can be selected as 1 second, 10 seconds, even 100 seconds as time step size while the accuracy is still maintained (the time period is 100s).

The main feature of the PTI method is very simple because the precision of integration depends on the computation accuracy of the exponential matrix function. The further study on the stability of the PTI method using Padé series approximation and Taylor series approximation will be given in Chapter 5.

The algorithmic properties of the precise time integration method combined with the dimensional expanding method are the same as that of the original algorithms. The method (A2) introduced in Subsection 3.2.1 is not good because the computational cost of this algorithm is still very large. With the dimensional expanding conversion, the method (A3) presented in Subsection 3.2.1 can avoid this disadvantage.

3.5 Numerical Example

One numerical example is used to investigate the efficiency of the presented algorithms. A comparison of methods (A1, A2 and A3) is given. In order to investigate the validity of the PTI methods (A1, A2 and A3) the illustrative numerical example in this section is demonstrated below. All computing programs in this thesis are run on a PC Pentium III 1GHz CPU.

$$\{\mathbf{Z}_0\} = \{1 \ 0\}^T \quad (3-58)$$

The comparison of the computational cost of the methods (A1, A2 and A3) is shown in Table 3-1. The N is 8 and p is 2. The results show that the efficiency of the results cannot be improved by using the method (A2). From the Table 3-1, the computational cost by using the method (A3) is less than that of the method (A2) when the same accurate results are derived. The computing efficiency can be improved by using the method (A3) and the computational expense cost is reduced significantly. With the increasing of the scale, the method (A3) can save more computational cost.

3.6 Summary

This Chapter discussed methods A2 and A3. The property of accuracy and efficiency of the original PTI method (A1) and two other methods (A2 and A3) are investigated. The dimensional expanding method is imported to improve the efficiency of the original PTI method and simplify the computing process. The extension of the dimensional expanding method is studied. More application of the extension of the dimensional expanding method will be discussed in later chapters. The particular solution is avoided to be computed by transforming the non-homogeneous equation into homogeneous equation. From the comparison of the algorithms, it can be seen that method (A3) is much better than the two other methods (A1 and A2).

Table 3-1. Computational costs of the methods (A1, A2 and A3)

Dimension of system (N_s)	Error	Result u_{N_s}	Computational costs (s)		
			Method A1	Method A2	Method A3
40	10^{-7}	565.459646	0.520	1.031	0.400
100	10^{-7}	825.991465	16.634	28.474	13.559
200	10^{-7}	852.437005	149.435	234.192	111.520

Notes: $t=100s$, $\Delta t = 100s$, $N=20$.

CHAPTER FOUR

Precise Time-Step Integration Algorithms Using Response Matrices with Dimensional Expansion

Overview

In this chapter, the precise time-step integration method by step-response and impulsive-response matrices is further developed by incorporating the dimensional expanding method. Two new precise time-step integration algorithms with excitations described by first-order and second-order differential equations are proposed. The first method is a direct extension of the existing algorithm. However, the extended system matrices are not symmetrical. In the second method, the Duhamel integrals are used as the particular solutions. As a result, the responses can be expressed in terms of the given initial conditions and the step-response matrix, impulsive-response matrix and a newly derived Duhamel-response matrix. The symmetry property of the system matrices can be utilized in the computation. To further reduce the computational effort, the relation between the Duhamel-response matrix and its derivative is established. A special computational procedure for periodic excitation is also discussed.

4.1 PTI Algorithm by Step-Response and Impulsive-Response Matrices

The PTI algorithm by step-response and impulsive-response matrices (Fung, 1997) is defined as Algorithm 4 (A4). In practice, the second-order differential equations can be solved individually. This algorithm is introduced briefly below.

Consider the governing equation of a discrete structural model,

$$[\mathbf{M}]\{\ddot{\mathbf{u}}(t)\} + [\mathbf{C}]\{\dot{\mathbf{u}}(t)\} + [\mathbf{K}]\{\mathbf{u}(t)\} = \{\mathbf{r}(t)\} \quad (4-1)$$

with initial conditions

$$\{\mathbf{u}(0)\} = \{\mathbf{u}_0\} \text{ and } \{\dot{\mathbf{u}}(0)\} = \{\mathbf{v}_0\} \quad (4-2)$$

Equation (4-1) can be solved by the modal decomposition method (Clough and O'Kelly, 1993). The second-order differential equations can be solved individually. The displacement response $\{\mathbf{u}(t)\}$ and the velocity response $\{\mathbf{v}(t)\}$ can be written as

$$\{\mathbf{u}(t)\} = [\mathbf{G}(t)]\{\mathbf{u}_0 - \mathbf{u}_s(0)\} + [\mathbf{H}(t)]\{\mathbf{v}_0 - \mathbf{v}_s(0)\} + \{\mathbf{u}_s(t)\} \quad (4-3)$$

$$\{\mathbf{v}(t)\} = [\dot{\mathbf{G}}(t)]\{\mathbf{u}_0 - \mathbf{u}_s(0)\} + [\dot{\mathbf{H}}(t)]\{\mathbf{v}_0 - \mathbf{v}_s(0)\} + \{\mathbf{v}_s(t)\} \quad (4-4)$$

where $\{\mathbf{u}_s(t)\}$ is the steady-state response corresponding to $\{\mathbf{r}(t)\}$, $\{\mathbf{v}_s(t)\} = \{\dot{\mathbf{u}}_s(t)\}$, $\{\mathbf{u}_0\}$ and $\{\mathbf{v}_0\}$ are the given initial conditions, $[\mathbf{G}(t)]$ and $[\mathbf{H}(t)]$ are the step-response and impulsive-response matrices respectively. These response matrices are evaluated without solving the eigenvalue problem and a general damping matrix is considered in the paper (Clough and O'Kelly, 1993). Hughes (1987) and Leung (1986 and 1993) have derived the steady state solutions using the dynamic stiffness and dynamic substructure methods. The formula of excitation can be harmonic, polynomial-like or exponentially varying.

In order to obtain the recurrence formulae of this algorithm, combining equations (4-3) and (4-4), the displacement response $\{\mathbf{u}(t)\}$ and the velocity response $\{\mathbf{v}(t)\}$ with given initial conditions $\{\mathbf{u}_0\}$ and $\{\mathbf{v}_0\}$ can be written as:

$$\begin{Bmatrix} \mathbf{u}(t) \\ \mathbf{v}(t) \end{Bmatrix} = \begin{bmatrix} \mathbf{G}(t) & \mathbf{H}(t) \\ \dot{\mathbf{G}}(t) & \dot{\mathbf{H}}(t) \end{bmatrix} \begin{Bmatrix} \mathbf{u}_0 \\ \mathbf{v}_0 \end{Bmatrix} \quad (4-5)$$

Alternatively, the same solution can be obtained by considering the displacement and velocity at $t/2$, in other words,

$$\begin{aligned} \begin{Bmatrix} \mathbf{u}(t) \\ \mathbf{v}(t) \end{Bmatrix} &= \begin{bmatrix} \mathbf{G}(t/2) & \mathbf{H}(t/2) \\ \dot{\mathbf{G}}(t/2) & \dot{\mathbf{H}}(t/2) \end{bmatrix} \begin{Bmatrix} \mathbf{u}(t/2) \\ \mathbf{v}(t/2) \end{Bmatrix} \\ &= \begin{bmatrix} \mathbf{G}(t/2) & \mathbf{H}(t/2) \\ \dot{\mathbf{G}}(t/2) & \dot{\mathbf{H}}(t/2) \end{bmatrix} \begin{bmatrix} \mathbf{G}(t/2) & \mathbf{H}(t/2) \\ \dot{\mathbf{G}}(t/2) & \dot{\mathbf{H}}(t/2) \end{bmatrix} \begin{Bmatrix} \mathbf{u}_0 \\ \mathbf{v}_0 \end{Bmatrix} \end{aligned} \quad (4-6)$$

Comparing Equation (4-5) and Equation (4-6), one has

$$[\mathbf{G}(t)] = [\mathbf{G}(t/2)][\mathbf{G}(t/2)] + [\mathbf{H}(t/2)][\mathbf{G}(t/2)] \quad (4-7a)$$

$$[\mathbf{H}(t)] = [\mathbf{G}(t/2)][\mathbf{H}(t/2)] + [\mathbf{H}(t/2)][\dot{\mathbf{H}}(t/2)] \quad (4-7b)$$

$$[\dot{\mathbf{G}}(t)] = [\dot{\mathbf{G}}(t/2)][\mathbf{G}(t/2)] + [\dot{\mathbf{H}}(t/2)][\dot{\mathbf{G}}(t/2)] \quad (4-7c)$$

$$[\dot{\mathbf{H}}(t)] = [\dot{\mathbf{G}}(t/2)][\mathbf{H}(t/2)] + [\dot{\mathbf{H}}(t/2)][\dot{\mathbf{H}}(t/2)] \quad (4-7d)$$

$[\dot{\mathbf{G}}(t)]$ and $[\dot{\mathbf{H}}(t)]$ can be obtained from $[\mathbf{G}(t)]$ and $[\mathbf{H}(t)]$. In practice computation, $[\mathbf{H}(t)]$ and $[\mathbf{J}(t)] = [\dot{\mathbf{H}}(t)]$ are evaluated recursively. $[\mathbf{G}(t)]$ and $[\dot{\mathbf{G}}(t)]$ are evaluated from $[\mathbf{H}(t)]$ and $[\mathbf{J}(t)]$. The recurrence formula for $[\mathbf{J}(t)]$ can be similarly found to be

$$[\mathbf{J}(t)] = [\mathbf{J}(t/2)][\mathbf{J}(t/2)] + [\mathbf{H}(t/2)][\mathbf{A}][\mathbf{H}(t/2)] \quad (4-8)$$

$[\mathbf{G}(t)]$ and $[\dot{\mathbf{G}}(t)]$ can be expressed in terms of $[\mathbf{H}(t)]$ and $[\mathbf{J}(t)]$ using equations (4-13a) and (4-13b) to reduce the computational effort. $[\mathbf{G}(t)]$ and $[\dot{\mathbf{G}}(t)]$ do not be evaluated recursively. Only $[\mathbf{H}(t)]$ and $[\mathbf{J}(t)]$ are evaluated recursively by equations (4-7b) and (4-8). With the step-response and impulsive-response

matrices and their time derivatives, the time-step integration can be carried out repeatedly at a time-step size Δt .

4.1.1 Step-Response and Impulsive-Response Matrices

For a given $[\mathbf{M}]$, $[\mathbf{C}]$, $[\mathbf{K}]$, Δt and N , the step-response matrix $[\mathbf{G}(t)]$ and the impulsive-response matrix $[\mathbf{H}(t)]$ and their derivatives at $t = \Delta t$ can be computed as follows:

(1) Compute the initial matrix $[\mathbf{H}(\Delta t/2^N)]$ and $[\mathbf{J}(\Delta t/2^N)] = [\dot{\mathbf{H}}(\Delta t/2^N)]$:

$$\begin{aligned} & [\mathbf{H}(\Delta t/2^N)] \\ &= [\dot{\mathbf{H}}(0)] \cdot \Delta t/2^N + [\ddot{\mathbf{H}}(0)] \cdot \frac{(\Delta t/2^N)^2}{2!} + \dots + [\mathbf{H}^{(n)}(0)] \cdot \frac{(\Delta t/2^N)^n}{n!} + \dots \end{aligned} \quad (4-9a)$$

$$\begin{aligned} & [\mathbf{J}(\Delta t/2^N)] = [\dot{\mathbf{H}}(\Delta t/2^N)] \\ &= [\dot{\mathbf{H}}(0)] + [\ddot{\mathbf{H}}(0)] \cdot (\Delta t/2^N) + \dots + [\mathbf{H}^{(n)}(0)] \cdot \frac{(\Delta t/2^N)^{(n-1)}}{(n-1)!} + \dots \end{aligned} \quad (4-9b)$$

where

$$\begin{aligned} & [\dot{\mathbf{H}}(0)] = [\mathbf{I}], \quad [\ddot{\mathbf{H}}(0)] = [\mathbf{B}] \\ & [\mathbf{H}^{(i+2)}(0)] = [\mathbf{H}^{(i)}(0)][\mathbf{A}] + [\mathbf{H}^{(i+1)}(0)][\mathbf{B}], \quad i = 1, 2, 3, \dots \end{aligned} \quad (4-10)$$

To reduce the truncation error, an auxiliary matrix $[\mathbf{J}_a(\Delta t/2^N)] = [\mathbf{J}(\Delta t/2^N)] - [\mathbf{I}]$ is computed instead of $[\mathbf{J}(\Delta t/2^N)]$.

$$\begin{aligned} & [\mathbf{J}_a(\Delta t/2^N)] = [\mathbf{J}(\Delta t/2^N)] - [\mathbf{I}] \\ &= [\ddot{\mathbf{H}}(0)] \cdot (\Delta t/2^N) + [\mathbf{H}^{(3)}(0)] \cdot \frac{(\Delta t/2^N)^2}{2!} + \dots + [\mathbf{H}^{(n)}(0)] \cdot \frac{(\Delta t/2^N)^{(n-1)}}{(n-1)!} + \dots \end{aligned} \quad (4-11)$$

(2) Compute $[\mathbf{H}(\Delta t)]$ and $[\mathbf{J}_a(\Delta t)] = [\mathbf{J}(\Delta t)] - [\mathbf{I}]$ recursively

$$\begin{aligned} [\mathbf{H}(\Delta t/2^{k-1})] &= [\mathbf{G}(\Delta t/2^k)][\mathbf{H}(\Delta t/2^k)] + [\mathbf{H}(\Delta t/2^k)][\mathbf{J}(\Delta t/2^k)] \\ &= ([\mathbf{J}_a(\Delta t/2^k)] - [\mathbf{H}(\Delta t/2^k)][\mathbf{B}])[\mathbf{H}(\Delta t/2^k)] \\ &\quad + [\mathbf{H}(\Delta t/2^k)][\mathbf{J}_a(\Delta t/2^k)] + 2[\mathbf{H}(\Delta t/2^k)] \end{aligned} \quad (4-12a)$$

$$\begin{aligned} [\mathbf{J}_a(\Delta t/2^{k-1})] \\ = [\mathbf{J}_a(\Delta t/2^k)]^2 + 2[\mathbf{J}_a(\Delta t/2^k)] + [\mathbf{H}(\Delta t/2^k)][\mathbf{A}][\mathbf{H}(\Delta t/2^k)] \end{aligned} \quad (4-12b)$$

where $k = N, \dots, 1$

(3) Compute $[\dot{\mathbf{H}}(\Delta t)]$, $[\mathbf{G}(\Delta t)]$ and $[\dot{\mathbf{G}}(\Delta t)]$

$$[\dot{\mathbf{H}}(\Delta t)] = [\mathbf{J}(\Delta t)] = [\mathbf{J}_a(\Delta t)] + [\mathbf{I}] \quad (4-13a)$$

$$[\mathbf{G}(\Delta t)] = [\mathbf{J}(\Delta t)] - [\mathbf{H}(\Delta t)][\mathbf{B}] \quad (4-13b)$$

$$[\dot{\mathbf{G}}(\Delta t)] = [\mathbf{H}(\Delta t)][\mathbf{A}] \quad (4-13c)$$

4.1.2 Steady-State Response

The computation of the steady-state response $\{\mathbf{u}_s(t)\}$ depends on the form of the excitation (Leung, 1985 and 1986). For example, if the excitation is given by

$$\{\mathbf{r}(t)\} = e^{nt} \cdot \sum_{j=0}^{q_s} \{\mathbf{f}_j\} t^j, \quad (4-14)$$

then, from function approximation (Leung, 1986),

$$\{\mathbf{u}_s(t)\} = e^{\eta t} \cdot \sum_{j=0}^{q_s} \{\mathbf{u}_j\} t^j \quad (4-15)$$

where

$$\begin{aligned} \{\mathbf{u}_j\} &= (\eta^2 [\mathbf{M}] + \eta [\mathbf{C}] + [\mathbf{K}])^{-1} \cdot \left\{ \{\mathbf{f}_j\} - j([\mathbf{C}] + 2\eta [\mathbf{M}]) \{\mathbf{u}_{j+1}\} - j(j+1) \{\mathbf{u}_{j+2}\} \right\} \\ j &= 0, 1, \dots, q_s \text{ with } \{\mathbf{u}_{q_s+1}\} = \{\mathbf{u}_{q_s+2}\} = \{\mathbf{0}\} \end{aligned} \quad (4-16)$$

Also

$$\{\mathbf{v}_s(t)\} = \frac{d}{dt} \{\mathbf{u}_s(t)\} = \eta \cdot \{\mathbf{u}_s(t)\} + e^{\eta t} \cdot \sum_{j=1}^{q_s} j \{\mathbf{u}_j\} t^{j-1} \quad (4-17)$$

4.1.3 Symmetric Matrices

The symmetric property (Fung, 1997) is mentioned because this character can be used to improve the computational efficient. It can be shown that while $[\mathbf{H}(t)]$ may not be symmetric, $[\mathbf{M}][\mathbf{H}(t)]$ is symmetric. As a result, $[\mathbf{M}][\mathbf{J}(t)]$ is symmetric as well. Let

$$[\bar{\mathbf{H}}(t)] = [\mathbf{M}][\mathbf{H}(t)] \text{ and } [\bar{\mathbf{J}}(t)] = [\mathbf{M}][\mathbf{J}(t)] \quad (4-18)$$

Denote the two symmetric matrices. The recurrence formulae for $[\bar{\mathbf{H}}(t)]$ and $[\bar{\mathbf{J}}(t)]$ are

$$\begin{aligned} [\bar{\mathbf{H}}(t)] &= [\bar{\mathbf{J}}(t/2)][\mathbf{M}]^{-1}[\bar{\mathbf{H}}(t/2)] + [\bar{\mathbf{H}}(t/2)][\mathbf{M}]^{-1}[\bar{\mathbf{J}}(t/2)] \\ &\quad - [\bar{\mathbf{H}}(t/2)][\mathbf{B}][\mathbf{M}]^{-1}[\bar{\mathbf{H}}(t/2)] \end{aligned} \quad (4-19a)$$

$$[\bar{\mathbf{J}}(t)] = [\bar{\mathbf{J}}(t/2)][\mathbf{M}]^{-1}[\bar{\mathbf{J}}(t/2)] + [\bar{\mathbf{H}}(t/2)][\mathbf{A}][\mathbf{M}]^{-1}[\bar{\mathbf{H}}(t/2)] \quad (4-19b)$$

It can be seen that if $[\bar{\mathbf{H}}(t/2)]$ and $[\bar{\mathbf{J}}(t/2)]$ are symmetric, $[\bar{\mathbf{H}}(t)]$ and $[\bar{\mathbf{J}}(t)]$ so computed would be symmetric as well.

If $[\mathbf{M}]$ is diagonal, $[\mathbf{M}]^{-1}$ is obtained readily. The computation of $[\bar{\mathbf{H}}(t)]$ and $[\bar{\mathbf{J}}(t)]$ is very efficient since $[\mathbf{C}]$ and $[\mathbf{K}]$ are usually banded matrices. Since $[\bar{\mathbf{H}}(t)]$ and

$[\bar{\mathbf{J}}(t)]$ are symmetric, only the diagonal and half of the remaining entries need to be computed.

If $[\mathbf{M}]$ is not diagonal, it can be decomposed by Cholesky factorization (Bathe, 1982) as

$$[\mathbf{M}] = [\mathbf{L}][\mathbf{L}]^T \text{ and } [\mathbf{M}]^{-1} = [\mathbf{L}]^{-T}[\mathbf{L}]^{-1} \quad (4-20)$$

where $[\mathbf{L}]$ is a lower triangular matrix. Let

$$[\bar{\bar{\mathbf{H}}}(t)] = [\mathbf{L}]^{-1}[\bar{\mathbf{H}}(t)][\mathbf{L}]^{-T} = [\mathbf{L}]^T[\mathbf{H}(t)][\mathbf{L}]^{-T} \quad (4-21a)$$

$$[\bar{\bar{\mathbf{J}}}(t)] = [\mathbf{L}]^{-1}[\bar{\mathbf{J}}(t)][\mathbf{L}]^{-T} = [\mathbf{L}]^T[\mathbf{J}(t)][\mathbf{L}]^{-T} \quad (4-21b)$$

denote two symmetric matrices. The recurrence formulae for these two matrices are

$$[\bar{\bar{\mathbf{H}}}(t)] = [\bar{\bar{\mathbf{J}}}(t/2)][\bar{\bar{\mathbf{H}}}(t/2)] + [\bar{\bar{\mathbf{H}}}(t/2)][\bar{\bar{\mathbf{J}}}(t/2)] + [\bar{\bar{\mathbf{H}}}(t/2)][\bar{\bar{\mathbf{C}}}] [\bar{\bar{\mathbf{H}}}(t/2)] \quad (4-22a)$$

$$[\bar{\bar{\mathbf{J}}}(t)] = [\bar{\bar{\mathbf{J}}}(t/2)][\bar{\bar{\mathbf{J}}}(t/2)] - [\bar{\bar{\mathbf{H}}}(t/2)][\bar{\bar{\mathbf{K}}}] [\bar{\bar{\mathbf{H}}}(t/2)] \quad (4-22b)$$

Where $[\bar{\bar{\mathbf{C}}}] = [\mathbf{L}]^{-1}[\mathbf{C}][\mathbf{L}]^{-T}$ and $[\bar{\bar{\mathbf{K}}}] = [\mathbf{L}]^{-1}[\mathbf{K}][\mathbf{L}]^{-T}$ are symmetric and can be pre-computed to increase the computational efficiency.

4.2 Application of Dimensional Expanding Method

To avoid computing the particular solutions, the dimensional expanding method is used to transform the non-homogeneous equations into homogeneous equations. Eventually, only a system first-order homogeneous equations needs to be studied.

In this chapter, the second-order equations are manipulated directly. The excitation could be described by first-order or second-order equations. As described in Chapter 3, the dimensional expanding method can be extended to be used in second-order equations directly. In the following, the corresponding differential equations for some typical excitations forms are given explicitly.

When the excitation is described by second-order differential equation $\{\mathbf{r}(t)\} = [\mathbf{f}]\{\mathbf{Z}\}$, where $\{\ddot{\mathbf{Z}}\} = [\mathbf{S}_2]\{\mathbf{Z}\}$, Equation (4-1) can be rewritten as

$$\begin{bmatrix} \mathbf{M} & \mathbf{0} \\ \mathbf{0} & \mathbf{I} \end{bmatrix} \begin{Bmatrix} \ddot{\mathbf{u}}(t) \\ \ddot{\mathbf{Z}}(t) \end{Bmatrix} + \begin{bmatrix} \mathbf{C} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} \begin{Bmatrix} \dot{\mathbf{u}}(t) \\ \dot{\mathbf{Z}}(t) \end{Bmatrix} + \begin{bmatrix} \mathbf{K} & -\mathbf{f} \\ \mathbf{0} & -\mathbf{S}_2 \end{bmatrix} \begin{Bmatrix} \mathbf{u}(t) \\ \mathbf{Z}(t) \end{Bmatrix} = \{\mathbf{0}\} \quad (4-23)$$

Attentively, if the excitation is described as first-order differential equation $\{\mathbf{r}(t)\} = [\mathbf{f}]\{\mathbf{Z}\}$ with $\{\dot{\mathbf{Z}}(t)\} = [\mathbf{S}_1]\{\mathbf{Z}(t)\}$, then Equation (4-1) becomes

$$\begin{bmatrix} \mathbf{M} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} \begin{Bmatrix} \ddot{\mathbf{u}}(t) \\ \ddot{\mathbf{Z}}(t) \end{Bmatrix} + \begin{bmatrix} \mathbf{C} & \mathbf{0} \\ \mathbf{0} & \mathbf{I} \end{bmatrix} \begin{Bmatrix} \dot{\mathbf{u}}(t) \\ \dot{\mathbf{Z}}(t) \end{Bmatrix} + \begin{bmatrix} \mathbf{K} & -\mathbf{f} \\ \mathbf{0} & -\mathbf{S}_1 \end{bmatrix} \begin{Bmatrix} \mathbf{u}(t) \\ \mathbf{Z}(t) \end{Bmatrix} = \{\mathbf{0}\} \quad (4-24)$$

Note that if a time shift is applied to the excitation, i.e.

$$\{\bar{\mathbf{r}}(t)\} = \{\mathbf{r}(t-t^*)\} \quad (4-25)$$

then, $\{\bar{\mathbf{r}}(t)\} = [\mathbf{f}]\{\bar{\mathbf{Z}}\}$, where $\{\bar{\mathbf{Z}}(t)\} = \{\mathbf{Z}(t-t^*)\}$ with $\{\bar{\mathbf{Z}}\}$ is governed by the same $[\mathbf{S}_1]$ and $[\mathbf{S}_2]$.

It should be noted that while Equation (4-23) can be tackled by many established algorithms, Equation (4-24) may impose some difficulties as the first matrix is singular. A more detailed discussion on how to solve the two equations is given in the following two sections.

4.3 Excitation Described by Second-Order Differential Equation

The algorithm presented in this subsection is denoted simply as Algorithm 5 (**A5**). This method (**A5**) does not compute the particular solution comparing with the method (**A4**). But the method (**A5**) enlarged the dimension of the computing matrices $[\mathbf{H}^*(t)]$ and $[\mathbf{J}^*(t)]$. The method (**A5**) is a simplified algorithm from the method (**A4**) in another word.

The PTI method by step-response and impulsive-response matrices (A4) can be used to solve the homogeneous second-order differential equation in Equation (4-23) directly without computing the steady-state response. Equation (4-23) can be expressed as:

$$[\mathbf{M}^*]\{\ddot{\mathbf{u}}^*(t)\} + [\mathbf{C}^*]\{\dot{\mathbf{u}}^*(t)\} + [\mathbf{K}^*]\{\mathbf{u}^*(t)\} = 0 \quad (4-26)$$

Where

$$\{\mathbf{u}^*(t)\} = \begin{Bmatrix} \mathbf{u}(t) \\ \mathbf{Z}(t) \end{Bmatrix}, [\mathbf{M}^*] = \begin{bmatrix} \mathbf{M} & \\ & \mathbf{I} \end{bmatrix}, [\mathbf{C}^*] = \begin{bmatrix} \mathbf{C} & \\ & \mathbf{0} \end{bmatrix} \text{ and } [\mathbf{K}^*] = \begin{bmatrix} \mathbf{K} & -\mathbf{f} \\ \mathbf{0} & -\mathbf{S}_2 \end{bmatrix}$$

As described in Section 4.2, the solution at time $t = \Delta t$ of Equation (4-26) can be obtained as

$$\begin{Bmatrix} \mathbf{u}^*(\Delta t) \\ \mathbf{v}^*(\Delta t) \end{Bmatrix} = \begin{bmatrix} \mathbf{G}^*(\Delta t) & \mathbf{H}^*(\Delta t) \\ \dot{\mathbf{G}}^*(\Delta t) & \dot{\mathbf{H}}^*(\Delta t) \end{bmatrix} \cdot \begin{Bmatrix} \mathbf{u}_0^* \\ \mathbf{v}_0^* \end{Bmatrix} \quad (4-27)$$

$$\text{with } \{\mathbf{u}_0^*\} = \begin{Bmatrix} \mathbf{u}(0) \\ \mathbf{Z}(0) \end{Bmatrix} \text{ and } \{\mathbf{v}_0^*\} = \begin{Bmatrix} \dot{\mathbf{u}}(0) \\ \dot{\mathbf{Z}}(0) \end{Bmatrix}$$

For one time step Δt , the initial recurrence formulae of $[\mathbf{H}^*(\Delta t/2^N)]$ and $[\mathbf{J}_a^*(\Delta t/2^N)] = [\mathbf{J}^*(\Delta t/2^N)] - [\mathbf{I}]$ can be computed from Taylor series solutions. The recurrence formulae of the matrices $[\mathbf{G}^*(\Delta t)]$, $[\mathbf{H}^*(\Delta t)]$ and their derivatives can be evaluated by replacing the $[\mathbf{M}]$, $[\mathbf{C}]$ and $[\mathbf{K}]$ by $[\mathbf{M}^*]$, $[\mathbf{C}^*]$ and $[\mathbf{K}^*]$ in Equations (4-9a-b), (4-11), (4-12a-b) and (4-13a-b). The 4-term truncated Taylor series approximation is recommended in (Fung, 1997). The recursive computational procedure is then carried out N times to get $[\mathbf{H}^*(\Delta t)]$ and $[\mathbf{J}^*(\Delta t)]$. A disadvantage of this algorithm is that symmetric property can not be used as $[\mathbf{K}^*]$ is not a symmetric matrix anymore. Hence, the computational effort will be higher.

4.4 Excitation Described by First-Order Differential Equation

As mentioned in Section 4.2, Equation (4-24) can not be tackled directly. However, the solution of Equation (4-24) could be expressed in terms of the given initial conditions $\{\mathbf{u}_0\}$, $\{\mathbf{v}_0\}$ and $\{\mathbf{Z}_0\}$. In other words, the displacement response $\{\mathbf{u}(t)\}$ and the velocity response $\{\mathbf{v}(t)\}$ are given by

$$\{\mathbf{u}(t)\} = [\mathbf{G}(t)]\{\mathbf{u}_0\} + [\mathbf{H}(t)]\{\mathbf{v}_0\} + [\mathbf{D}(t)]\{\mathbf{Z}_0\} \quad (4-28a)$$

$$\{\mathbf{v}(t)\} = [\dot{\mathbf{G}}(t)]\{\mathbf{u}_0\} + [\dot{\mathbf{H}}(t)]\{\mathbf{v}_0\} + [\dot{\mathbf{D}}(t)]\{\mathbf{Z}_0\} \quad (4-28b)$$

It should be noted that the solution in Equation (4-24) can also be written as

$$\{\mathbf{u}(t)\} = [\mathbf{G}(t)]\{\mathbf{u}_0\} + [\mathbf{H}(t)]\{\mathbf{v}_0\} + \{\mathbf{u}_d(t)\} \quad (4-29)$$

where $\{\mathbf{u}_d(t)\}$ is a particular solution given by the Duhamel integral with zero initial conditions.

Comparing Equation (4-28a) and Equation (4-29), it reveals that

$$\{\mathbf{u}_d(t)\} = [\mathbf{D}(t)]\{\mathbf{Z}_0\} \quad (4-30)$$

As a result, the particular solution corresponding to the Duhamel integral can be expressed in terms of $\{\mathbf{Z}_0\}$ and a Duhamel-response matrix $[\mathbf{D}(t)]$.

The computation of $[\mathbf{H}(t)]$ and $[\mathbf{G}(t)]$ (or $[\mathbf{J}(t)]$) has been discussed in Section 4.1. In the following, the computation of $[\mathbf{D}(t)]$ is considered.

Note that $[\mathbf{D}(t)]$ satisfy the following equation.

$$[\mathbf{M}][\ddot{\mathbf{D}}(t)]\{\mathbf{Z}_0\} + [\mathbf{C}][\dot{\mathbf{D}}(t)]\{\mathbf{Z}_0\} + [\mathbf{K}][\mathbf{D}(t)]\{\mathbf{Z}_0\} = [\mathbf{f}]\{\mathbf{Z}(t)\} \quad (4-31)$$

with $[\mathbf{D}(0)] = [\dot{\mathbf{D}}(0)] = [\mathbf{0}]$, where $[\mathbf{0}]$ is the zero matrix, and $\{\dot{\mathbf{Z}}(t)\} = [\mathbf{S}_1]\{\mathbf{Z}(t)\}$.

The Taylor series of $[\mathbf{D}(t)]$ and $[\dot{\mathbf{D}}(t)]$ at $t=0$ can be written as

$$[\mathbf{D}(t)] = [\ddot{\mathbf{D}}(0)] \cdot \frac{t^2}{2!} + [\mathbf{D}^{(3)}(0)] \cdot \frac{t^3}{3!} + [\mathbf{D}^{(4)}(0)] \cdot \frac{t^4}{4!} + \cdots + [\mathbf{D}^{(n+1)}(0)] \cdot \frac{t^{n+1}}{(n+1)!} + \cdots \quad (4-32a)$$

$$[\dot{\mathbf{D}}(t)] = [\ddot{\mathbf{D}}(0)] \cdot t + [\mathbf{D}^{(3)}(0)] \cdot \frac{t^2}{2!} + [\mathbf{D}^{(4)}(0)] \cdot \frac{t^3}{3!} + \cdots + [\mathbf{D}^{(n+1)}(0)] \cdot \frac{t^n}{n!} + \cdots \quad (4-32b)$$

Differentiating Equation (4-31) with respect to t and set $t=0$ repeatedly, one has

$$[\mathbf{M}][\ddot{\mathbf{D}}(0)]\{\mathbf{Z}_0\} + [\mathbf{C}][\dot{\mathbf{D}}(0)]\{\mathbf{Z}_0\} + [\mathbf{K}][\mathbf{D}(0)]\{\mathbf{Z}_0\} = [\mathbf{f}]\{\mathbf{Z}_0\} \quad (4-33)$$

$$[\mathbf{M}][\mathbf{D}^{(3)}(0)]\{\mathbf{Z}_0\} + [\mathbf{C}][\ddot{\mathbf{D}}(0)]\{\mathbf{Z}_0\} + [\mathbf{K}][\dot{\mathbf{D}}(0)]\{\mathbf{Z}_0\} = [\mathbf{f}]\{\dot{\mathbf{Z}}(0)\} = [\mathbf{f}][\mathbf{S}_1]\{\mathbf{Z}_0\} \quad (4-34)$$

⋮

$$\begin{aligned} [\mathbf{M}][\mathbf{D}^{(i+1)}(0)]\{\mathbf{Z}_0\} + [\mathbf{C}][\mathbf{D}^{(i)}(0)]\{\mathbf{Z}_0\} + [\mathbf{K}][\mathbf{D}^{(i-1)}(0)]\{\mathbf{Z}_0\} &= [\mathbf{f}]\{\mathbf{Z}^{(i-1)}(0)\} \\ &= [\mathbf{f}][\mathbf{S}_1]^{i-1}\{\mathbf{Z}_0\} \end{aligned} \quad (4-35)$$

Making use of Equation (4-10) and (4-31), $[\mathbf{D}^{(i)}(0)]$ can be written as

$$[\ddot{\mathbf{D}}(0)] = [\mathbf{M}]^{-1}[\mathbf{f}], \quad [\mathbf{D}^{(3)}(0)] = [\ddot{\mathbf{H}}(0)][\mathbf{M}]^{-1}[\mathbf{f}] + [\ddot{\mathbf{D}}(0)][\mathbf{S}_1], \quad (4-36)$$

$$[\mathbf{D}^{(i+1)}(0)] = [\mathbf{H}^{(i)}(0)][\mathbf{M}]^{-1}[\mathbf{f}] + [\mathbf{D}^{(i)}(0)][\mathbf{S}_1], \quad i = 2, 3, 4, \dots$$

Hence the matrix $[\mathbf{D}(t)]$ can be evaluated efficiently.

Similar to $[\mathbf{H}(t)]$ and $[\mathbf{G}(t)]$, using the Taylor series solution to compute the additional matrix $[\mathbf{E}(t)]$ and matrix $[\mathbf{D}(t)]$ at $t = \Delta t$ would require Δt to be very small. In the following, the squaring and scaling technique is used to compute $[\mathbf{E}(t)]$ and $[\mathbf{D}(t)]$ recursively.

4.4.1 Computation of $[\mathbf{E}(t)]$, $[\mathbf{D}(t)]$ and $[\dot{\mathbf{D}}(t)]$

From Equations (4-28a) and (4-28b), the solution of Equation (4-1) at $t = \Delta t$ can be expressed as:

$$\begin{Bmatrix} \mathbf{u}(\Delta t) \\ \mathbf{v}(\Delta t) \\ \mathbf{Z}(\Delta t) \end{Bmatrix} = \begin{bmatrix} \mathbf{G}(\Delta t) & \mathbf{H}(\Delta t) & \mathbf{D}(\Delta t) \\ \dot{\mathbf{G}}(\Delta t) & \dot{\mathbf{H}}(\Delta t) & \dot{\mathbf{D}}(\Delta t) \\ \mathbf{0} & \mathbf{0} & \mathbf{E}(\Delta t) \end{bmatrix} \begin{Bmatrix} \mathbf{u}_0 \\ \mathbf{v}_0 \\ \mathbf{Z}_0 \end{Bmatrix} \quad (4-37)$$

where $[\mathbf{E}(\Delta t)] = \exp([\mathbf{S}_1]\Delta t)$ so that $\{\mathbf{Z}(\Delta t)\} = [\mathbf{E}(\Delta t)] \cdot \{\mathbf{Z}_0\}$ is the solution of $\{\dot{\mathbf{Z}}(t)\} = [\mathbf{S}_1] \cdot \{\mathbf{Z}(t)\}$. The matrix $[\mathbf{E}(\Delta t)]$ also can be obtained by using Taylor series.

$$[\mathbf{E}(\Delta t)] = \sum_{i=0}^{\infty} [\mathbf{S}_1]^i \cdot \frac{t^i}{i!} = [\mathbf{I}] + [\mathbf{S}_1] \cdot \Delta t + [\mathbf{S}_1]^2 \cdot \frac{\Delta t^2}{2!} + [\mathbf{S}_1]^3 \cdot \frac{\Delta t^3}{3!} + \dots \quad (4-38)$$

or using the squaring and scaling technology as

$$[\mathbf{E}(\Delta t)] = [\mathbf{E}(\frac{\Delta t}{2^N})]^{2^N} = ([\mathbf{I}] + [\mathbf{E}_a])^{2^N} \quad (4-39)$$

$$\text{where } [\mathbf{E}_a] = \sum_{i=1}^4 \frac{[\mathbf{S}_1]}{i!} \left(\frac{\Delta t}{2^N} \right)^i.$$

Obviously, the same solution can be obtained by applying a time step $\Delta t/2$ first and then followed by another $\Delta t/2$. In other words,

$$\begin{Bmatrix} \mathbf{u}(\Delta t) \\ \mathbf{v}(\Delta t) \\ \mathbf{Z}(\Delta t) \end{Bmatrix} = \begin{bmatrix} \mathbf{G}(\Delta t/2) & \mathbf{H}(\Delta t/2) & \mathbf{D}(\Delta t/2) \\ \dot{\mathbf{G}}(\Delta t/2) & \dot{\mathbf{H}}(\Delta t/2) & \dot{\mathbf{D}}(\Delta t/2) \\ \mathbf{0} & \mathbf{0} & \mathbf{E}(\Delta t/2) \end{bmatrix} \begin{Bmatrix} \mathbf{u}(\Delta t/2) \\ \mathbf{v}(\Delta t/2) \\ \mathbf{Z}(\Delta t/2) \end{Bmatrix} \quad (4-40a)$$

$$= \begin{bmatrix} \mathbf{G}(\Delta t/2) & \mathbf{H}(\Delta t/2) & \mathbf{D}(\Delta t/2) \\ \dot{\mathbf{G}}(\Delta t/2) & \dot{\mathbf{H}}(\Delta t/2) & \dot{\mathbf{D}}(\Delta t/2) \\ \mathbf{0} & \mathbf{0} & \mathbf{E}(\Delta t/2) \end{bmatrix} \begin{bmatrix} \mathbf{G}(\Delta t/2) & \mathbf{H}(\Delta t/2) & \mathbf{D}(\Delta t/2) \\ \dot{\mathbf{G}}(\Delta t/2) & \dot{\mathbf{H}}(\Delta t/2) & \dot{\mathbf{D}}(\Delta t/2) \\ \mathbf{0} & \mathbf{0} & \mathbf{E}(\Delta t/2) \end{bmatrix} \begin{Bmatrix} \mathbf{u}_0 \\ \mathbf{v}_0 \\ \mathbf{Z}_0 \end{Bmatrix} \quad (4-40b)$$

Comparing Equation (4-40b) and Equation (4-37), it can be seen that the recurrence equations for $[\mathbf{E}(\Delta t)]$, $[\mathbf{D}(\Delta t)]$ and $[\dot{\mathbf{D}}(\Delta t)]$ are

$$[\mathbf{E}(\Delta t)] = [\mathbf{E}(\Delta t/2)][\mathbf{E}(\Delta t/2)] \quad (4-41a)$$

$$[\mathbf{D}(\Delta t)] = [\mathbf{G}(\Delta t/2)][\mathbf{D}(\Delta t/2)] + [\mathbf{H}(\Delta t/2)][\dot{\mathbf{D}}(\Delta t/2)] + [\mathbf{D}(\Delta t/2)][\mathbf{E}(\Delta t/2)] \quad (4-41b)$$

$$[\dot{\mathbf{D}}(\Delta t)] = [\dot{\mathbf{G}}(\Delta t/2)][\mathbf{D}(\Delta t/2)] + [\dot{\mathbf{H}}(\Delta t/2)][\dot{\mathbf{D}}(\Delta t/2)] + [\dot{\mathbf{D}}(\Delta t/2)][\mathbf{E}(\Delta t/2)] \quad (4-41c)$$

4.4.2 Relation of $[\mathbf{D}(t)]$ and $[\dot{\mathbf{D}}(t)]$

The present precise time-step integration method by step, impulsive and Duhamel-response matrices with dimensional expansion requires the computation of matrices $[\mathbf{D}(t)]$ and $[\dot{\mathbf{D}}(t)]$ at $t = \Delta t$. It is not efficient to compute the Duhamel-response matrix $[\mathbf{D}(t)]$ and its derivative $[\dot{\mathbf{D}}(t)]$ separately. To reduce the computational cost, the relation between Duhamel-response matrix $[\mathbf{D}(t)]$ and its derivative $[\dot{\mathbf{D}}(t)]$ is established below:

$$[\dot{\mathbf{D}}(t)] = [\mathbf{D}(t)] \cdot [\mathbf{S}_1] + [\mathbf{H}(t)] \cdot [\mathbf{M}]^{-1} [\mathbf{f}]. \quad (4-42)$$

Equation (4-42) is investigated as following.

Proof: Further studying the Duhamel integral part (Leung, 1988), there has

$$\begin{aligned} [\mathbf{D}(t)] \cdot \{\mathbf{Z}_0\} &= \int_0^t [\mathbf{H}(t-s)] \cdot \{\mathbf{R}(t)\} ds \\ &= \int_0^t [\mathbf{H}(s)] \cdot \{\mathbf{R}(t-s)\} d(t-s) \\ &= \int_0^t [\mathbf{H}(s)] \cdot \{\mathbf{R}(t-s)\} ds \end{aligned} \quad (4-43)$$

$$\text{With } \{\mathbf{R}(t)\} = [\mathbf{M}]^{-1} [\mathbf{f}] \{\mathbf{Z}(t)\} \text{ and } \{\mathbf{Z}(t)\} = [\mathbf{E}(t)] \{\mathbf{Z}_0\} = \exp([\mathbf{S}_1] \cdot t) \cdot \{\mathbf{Z}_0\} \quad (4-44)$$

$$[\mathbf{D}(t)] \cdot \{\mathbf{Z}_0\} = \int_0^t [\mathbf{H}(s)] \cdot [\mathbf{M}]^{-1} [\mathbf{f}] \cdot \exp([\mathbf{S}_1](t-s)) \cdot \{\mathbf{Z}_0\} ds \quad (4-45)$$

Or

$$[\mathbf{D}(t)] \cdot \{\mathbf{Z}_0\} = \int_0^t [\mathbf{H}(s)] \cdot [\mathbf{M}]^{-1} [\mathbf{f}] \cdot \exp([\mathbf{S}_1](t-s)) \cdot \{\mathbf{Z}_0\} ds \quad (4-46)$$

Equation (4-46) was differentiated over t .

$$\begin{aligned} [\dot{\mathbf{D}}(t)] \cdot \{\mathbf{Z}_0\} &= ([\mathbf{H}(s)] \cdot [\mathbf{M}]^{-1} [\mathbf{f}] \cdot \exp([\mathbf{S}_1](t-s)) \{\mathbf{Z}_0\}) \Big|_{s=t} \\ &+ \int_0^t [\mathbf{H}(s)] \cdot [\mathbf{M}]^{-1} [\mathbf{f}] \cdot \exp([\mathbf{S}_1](t-s)) \cdot [\mathbf{S}_1] \{\mathbf{Z}_0\} ds \end{aligned} \quad (4-47)$$

$$= [\mathbf{H}(t)] \cdot [\mathbf{M}]^{-1} [\mathbf{f}] \cdot \{\mathbf{Z}_0\} + \int_0^t [\mathbf{H}(s)] \cdot [\mathbf{M}]^{-1} [\mathbf{f}] \cdot \exp([\mathbf{S}_1](t-s)) \cdot ds \cdot [\mathbf{S}_1] \{\mathbf{Z}_0\}. \quad (4-48)$$

Make use of Equation (4-43),

$$[\dot{\mathbf{D}}(t)] \cdot \{\mathbf{Z}_0\} = [\mathbf{H}(t)] \cdot [\mathbf{M}]^{-1} \cdot \{\mathbf{f}\} + [\mathbf{D}(t)] \cdot [\mathbf{S}_1] \cdot \{\mathbf{Z}_0\} \quad (4-49)$$

Since $\{\mathbf{Z}_0\}$ can be arbitrary, $[\dot{\mathbf{D}}(t)]$ can be obtained through Duhamel-response matrix $[\mathbf{D}(t)]$ and impulsive-response matrix $[\mathbf{H}(t)]$ as following:

$$[\dot{\mathbf{D}}(t)] = [\mathbf{H}(t)] \cdot [\mathbf{M}]^{-1} \cdot \{\mathbf{f}\} + [\mathbf{D}(t)] \cdot [\mathbf{S}_1] \quad (4-50)$$

In the present algorithm, given $[\mathbf{H}(\Delta t/2)]$, $[\mathbf{J}(\Delta t/2)]$ and $[\mathbf{D}(\Delta t/2)]$, the matrices $[\mathbf{E}(\Delta t)]$ and $[\mathbf{D}(\Delta t)]$ can be computed from Equation (4-41a) and (4-41b), respectively. The matrix $[\dot{\mathbf{D}}(\Delta t)]$ in Equation (4-41c) can be obtained from Equation (4-50). The equal order truncated Taylor series approximations can be used for $[\mathbf{H}(\Delta t/2^N)]$, $[\mathbf{J}(\Delta t/2^N)]$, $[\mathbf{D}(\Delta t/2^N)]$ and $[\mathbf{E}(\Delta t/2^N)]$ initially. To reduce the truncation error, the auxiliary matrices $[\mathbf{J}_a(\Delta t)] = [\mathbf{J}(\Delta t/2)] - [\mathbf{I}]$ and $[\mathbf{E}_a(\Delta t)] = [\mathbf{E}(\Delta t/2)] - [\mathbf{I}]$ are computed. The whole computing flow diagram of this new algorithm is given as following:

- (1) Compute the initial condition $[\mathbf{H}(\Delta t/2^N)]$, $[\mathbf{D}(\Delta t/2^N)]$, $[\mathbf{J}_a(\Delta t/2^N)] = [\mathbf{J}(\Delta t/2^N)] - [\mathbf{I}]$ and $[\mathbf{E}_a(\Delta t/2^N)] = [\mathbf{E}(\Delta t/2^N)] - [\mathbf{I}]$ from:

$$\begin{aligned} & [\mathbf{H}(\Delta t/2^N)] \\ &= [\mathbf{I}] \cdot (\Delta t/2^N) + [\mathbf{B}] \cdot \frac{(\Delta t/2^N)^2}{2!} + ([\mathbf{B}]^2 + [\mathbf{A}]) \cdot \frac{(\Delta t/2^N)^3}{3!} \\ & \quad + ([\mathbf{B}]^3 + [\mathbf{B}][\mathbf{A}] + [\mathbf{A}][\mathbf{B}]) \cdot \frac{(\Delta t/2^N)^4}{4!} \end{aligned} \quad (4-51a)$$

$$\begin{aligned} & [\mathbf{J}_a(\Delta t/2^N)] = [\mathbf{J}(\Delta t/2^N)] - [\mathbf{I}] \\ &= [\mathbf{B}] \cdot \Delta t/2^N + ([\mathbf{B}]^2 + [\mathbf{A}]) \cdot \frac{(\Delta t/2^N)^2}{2!} \\ & \quad + ([\mathbf{B}]^3 + [\mathbf{B}][\mathbf{A}] + [\mathbf{A}][\mathbf{B}]) \cdot \frac{(\Delta t/2^N)^3}{3!} \end{aligned} \quad (4-51b)$$

$$\begin{aligned}
 & [\mathbf{D}(\Delta t/2^N)] \\
 &= [\mathbf{M}]^{-1}[\mathbf{f}] \cdot \frac{(\Delta t/2^N)^2}{2!} + ([\mathbf{B}][\mathbf{M}]^{-1}[\mathbf{f}] + [\mathbf{M}]^{-1}[\mathbf{f}][\mathbf{S}_1]) \cdot \frac{(\Delta t/2^N)^3}{3!} \\
 & \quad + \left\{ \begin{aligned} & ([\mathbf{B}]^2 + [\mathbf{A}]) \cdot [\mathbf{M}]^{-1}[\mathbf{f}] + \\ & ([\mathbf{M}]^{-1}[\mathbf{f}][\mathbf{S}_1] + [\mathbf{B}][\mathbf{M}]^{-1}[\mathbf{f}]) \cdot [\mathbf{S}_1] \end{aligned} \right\} \cdot \frac{(\Delta t/2^N)^4}{4!} \quad (4-51c)
 \end{aligned}$$

$$\begin{aligned}
 & [\mathbf{E}_a(\Delta t/2^N)] = [\mathbf{E}(\Delta t/2^N)] - [\mathbf{I}] \\
 &= [\mathbf{S}_1] \cdot (\Delta t/2^N) + [\mathbf{S}_1]^2 \cdot \frac{(\Delta t/2^N)^2}{2!} + [\mathbf{S}_1]^3 \cdot \frac{(\Delta t/2^N)^3}{3!} + [\mathbf{S}_1]^4 \cdot \frac{(\Delta t/2^N)^4}{4!} \quad (4-51d)
 \end{aligned}$$

(2) Compute $[\mathbf{H}(\Delta t)]$, $[\mathbf{J}_a(\Delta t)]$, $[\mathbf{D}(\Delta t)]$ and $[\mathbf{E}_a(\Delta t)]$ recursively

$$[\mathbf{H}(\Delta t/2^{k-1})] = [\mathbf{G}(\Delta t/2^k)][\mathbf{H}(\Delta t/2^k)] + [\mathbf{H}(\Delta t/2^k)][\mathbf{J}_a(\Delta t/2^k)] + [\mathbf{I}] \quad (4-52a)$$

$$[\mathbf{J}_a(\Delta t/2^{k-1})] = [\mathbf{J}_a(\Delta t/2^k)]^2 + 2[\mathbf{J}(\Delta t/2^k)] + [\mathbf{H}(\Delta t/2^k)][\mathbf{A}][\mathbf{H}(\Delta t/2^k)] \quad (4-52b)$$

$$\begin{aligned}
 [\mathbf{D}(\Delta t/2^{k-1})] &= [\mathbf{G}(\Delta t/2^k)][\mathbf{D}(\Delta t/2^k)] + [\mathbf{D}(\Delta t/2^k)] \cdot ([\mathbf{E}_a(\Delta t/2^k)] + [\mathbf{I}]) \\
 & \quad + [\mathbf{H}(\Delta t/2^k)]([\mathbf{H}(\Delta t/2^k)] \cdot [\mathbf{M}]^{-1}[\mathbf{f}] + [\mathbf{D}(\Delta t/2^k)] \cdot [\mathbf{S}_1]) \quad (4-52c)
 \end{aligned}$$

$$[\mathbf{E}_a(\Delta t/2^{k-1})] = [\mathbf{E}_a(\Delta t/2^k)]^2 + 2[\mathbf{E}_a(\Delta t/2^k)], \quad k = N, \dots, 1 \quad (4-52d)$$

(3) Compute $[\mathbf{G}(\Delta t)]$, $[\dot{\mathbf{G}}(\Delta t)]$, $[\dot{\mathbf{D}}(\Delta t)]$ and $[\mathbf{E}(\Delta t)]$ from

$$[\mathbf{G}(\Delta t)] = [\mathbf{J}_a(\Delta t)] + [\mathbf{I}] - [\mathbf{H}(\Delta t)][\mathbf{B}] \quad (4-53a)$$

$$[\dot{\mathbf{G}}(\Delta t)] = [\mathbf{H}(\Delta t)][\mathbf{A}] \quad (4-53b)$$

$$[\dot{\mathbf{D}}(\Delta t)] = [\mathbf{H}(\Delta t)] \cdot [\mathbf{M}]^{-1}[\mathbf{f}] + [\mathbf{D}(\Delta t)] \cdot [\mathbf{S}_1] \quad (4-53c)$$

$$[\mathbf{E}(\Delta t)] = [\mathbf{E}_a(\Delta t)] + [\mathbf{I}] \quad (4-53d)$$

Note that the Duhamel-response matrix $[\mathbf{D}(\Delta t)]$ in general is not a square matrix (and hence is not symmetrical normally). However, the computation of $[\mathbf{H}(\Delta t)]$ and $[\mathbf{J}(\Delta t)]$ can still make use of the symmetric property as described in the research (Fung, 1997) to reduce the computational cost. Furthermore, the

computation of the additional matrix $[\mathbf{E}(\Delta t)]$ only requires one matrix multiplication in each recursively evaluation. This new PTI method is denoted as algorithm 6. It is denoted as method **A6**.

In the present two new methods, the desirable time-step size Δt can be chosen independent of the highest frequency in the model. After the response matrices are evaluated, the time-step integration can be carried out with the time step size chosen. The proposed methods are therefore very flexible. The time-step size need not be small and can be several times of the longest period in the system!

4.5 Computational Effort

In this section, the computational effort of the dimensional expanding PTI method (**A3**), the method (**A4**) and the two new proposed methods (**A5** and **A6**) are studied.

Let N_s be the dimension of the system and g be the dimension of the vector $\{\mathbf{Z}(t)\}$. The solutions by different time-step algorithms are given by:

- (1) The dimensional expanding precise time-step integration method (**A3**);

$$\{\mathbf{U}^*(\Delta t)\} = \exp([\mathbf{W}_3] \cdot \Delta t)_{(2N_s+g) \times (2N_s+g)} \cdot \{\mathbf{U}_0^*\} \quad (4-54)$$

- (2) The precise time-step integration method by step-response and impulsive-response matrices (**A4**);

$$\begin{Bmatrix} \mathbf{u}(\Delta t) \\ \mathbf{v}(\Delta t) \end{Bmatrix} = \begin{bmatrix} \mathbf{G}(\Delta t) & \mathbf{H}(\Delta t) \\ \dot{\mathbf{G}}(\Delta t) & \dot{\mathbf{H}}(\Delta t) \end{bmatrix}_{2N_s \times 2N_s} \cdot \begin{Bmatrix} \mathbf{u}_0 - \mathbf{u}_s(0) \\ \mathbf{v}_0 - \mathbf{v}_s(0) \end{Bmatrix} + \begin{Bmatrix} \mathbf{u}_s(\Delta t) \\ \mathbf{v}_s(\Delta t) \end{Bmatrix} \quad (4-55)$$

- (3) The new precise time-step integration method by step-response and impulsive-response matrices with dimensional expansion (A5);

$$\begin{Bmatrix} \mathbf{u}^*(\Delta t) \\ \mathbf{v}^*(\Delta t) \end{Bmatrix} = \begin{bmatrix} \mathbf{G}^*(\Delta t) & \mathbf{H}^*(\Delta t) \\ \dot{\mathbf{G}}^*(\Delta t) & \dot{\mathbf{H}}^*(\Delta t) \end{bmatrix}_{2(N_s+g) \times 2(N_s+g)} \cdot \begin{Bmatrix} \mathbf{u}_0^* \\ \mathbf{v}_0^* \end{Bmatrix} \quad (4-56)$$

- (4) The new precise time-step integration method by step-response, impulsive-response and Duhamel-response matrices with dimensional expansion (A6) are given as following, respectively.

$$\begin{Bmatrix} \mathbf{u}(\Delta t) \\ \mathbf{v}(\Delta t) \\ \mathbf{Z}(\Delta t) \end{Bmatrix} = \begin{bmatrix} \mathbf{G}(\Delta t) & \mathbf{H}(\Delta t) & \mathbf{D}(\Delta t) \\ \dot{\mathbf{G}}(\Delta t) & \dot{\mathbf{H}}(\Delta t) & \dot{\mathbf{D}}(\Delta t) \\ 0 & 0 & \mathbf{E}(\Delta t) \end{bmatrix}_{(2N_s+g) \times (2N_s+g)} \begin{Bmatrix} \mathbf{u}_0 \\ \mathbf{v}_0 \\ \mathbf{Z}_0 \end{Bmatrix} \quad (4-57)$$

Assuming the all matrices are full, the computational efforts to get the results of Equations (4-54), (4-55), (4-56) and (4-57) are studied. For one step, the operation counts of these methods are given in the Table 4-1, Table 4-2, Table 4-3 and Table 4-4.

The computing process of the precise time-step integration algorithm by step-response and impulsive-response matrices with dimensional expansion (A5) is similar to the method (A4) according to the analysis in Section 4.3. Without the symmetric property, the method (A5) lost the efficiency comparable with the method (A4). Hence, the method (A4) will be more efficient than the method (A5).

The method (A5) proposed in this research will be better than the method (A3) if the following condition is satisfied,

$$\begin{aligned} & 2(2N_s + g)^3 + N(2N_s + g)^3 + (2N_s + g)^2 \frac{T}{\Delta t} \\ & > 2n \cdot (N_s + g)^3 + 4N \cdot (N_s + g)^3 + 4(N_s + g)^2 \frac{T}{\Delta t} \end{aligned} \quad (4-58)$$

or

$$(N_s + g)^3(16 + 4N - 2n) + 6(N_s + g)g^2(2 + N) > g^3(2 + N) + 12(N_s + g)^2g(2 + N) + (4N_s g + 3g^3) \frac{T}{\Delta t} \quad (4-59)$$

or

$$(N_s + g)(16 + 4N - 2n) \frac{1}{(2 + N)} + 6 \frac{g^2}{(N_s + g)} > \frac{g^3}{(N_s + g)^2} + 12g + (4N_s g + 3g^3) \frac{T}{\Delta t} \frac{1}{(2 + N)(N_s + g)^2} \quad (4-60)$$

If $N=20$ and $n=4$,

$$4N_s + 6 \frac{g^2}{(N_s + g)} > \frac{g^3}{(N_s + g)^2} + 8g + \frac{(2N_s g + 1.5g^3)T}{11\Delta t(N_s + g)^2} \quad (4-61)$$

Equation (4-61) can be simplified as

$$4N_s^3 - N_s(6g^2 + \frac{2T}{11\Delta t}g) - (3 + \frac{3T}{22\Delta t})g^3 > 0 \quad (4-62)$$

or

$$4N_s^2 > (6g^2 + \frac{2T}{11\Delta t}g) + (3 + \frac{3T}{22\Delta t}) \frac{g^3}{N_s} \quad (4-63)$$

Hence the precise time-step integration algorithm by step-response and impulsive-response matrices with dimensional expansion (A5) will be more efficient than the dimensional expanding PTI method (A3) if Equation (4-63) is satisfied. The value of Δt can be large using the precise time-step integration method while the accuracy is still maintained. So the value of $\frac{T}{\Delta t}$ is normally not very large.

Equation (4-63) can be satisfied easily when the dimension of the system is large.

For example, if $g=6$ and $\frac{T}{\Delta t} = 1.0$, Equation (4-63) gives $N_s > 9$.

Furthermore, the precise time-step integration algorithm by step-response, impulsive-response and Duhamel-response matrices (A6) with dimensional expansion will be more efficient than the precise time-step integration algorithm by step-response and impulsive-response matrices method (A4) if

$$2nN_s^3 + nN_s^2 \cdot g + n \cdot g^3 + 4N \cdot N_s^3 + 4N \cdot N_s^2 \cdot g + Ng^3 + (4N_s^2 + 2N_s \cdot g + 2g^2) \cdot \frac{T}{\Delta t} < 2n \cdot N_s^3 + 4N \cdot N_s^3 + 4N_s^2 \frac{T}{\Delta t} + [Inte] \quad (4-64)$$

or

$$(4N + n)N_s^2 g + (N + n)g^3 + (2gN_s + g^2) \frac{T}{\Delta t} < [Inte] \quad (4-65)$$

Since the value of g is defined by the external loading form, g in general is not very large. The part of $[Inte]$ is determined by the particular solution. If the excitation is complex, this part will become large. Hence, the methods with dimensional expanding method are more efficient than the original methods. The precise time-step integration method with step-response, impulsive-response and Duhamel-response matrices with dimensional expansion (A6) will be more efficient than the precise time-step integration algorithm by step-response and impulsive-response matrices method (A4) when Equation (4-64) is satisfied.

For example, if the external load is assumed to be linear, $g=2$, $N=20$, $n=4$, then

$[Inte] = N_s^3 + N_s^2 \cdot \frac{T}{\Delta t}$. Equation (4-64) is rewritten as:

$$(4N + n)N_s^2 g + (N + n)g^3 + (2gN_s + g^2) \frac{T}{\Delta t} < N_s^3 + N_s^2 \cdot \frac{T}{\Delta t} \quad (4-66)$$

or

$$N_s^3 - (178 - \frac{T}{\Delta t})N_s^2 - 4 \frac{T}{\Delta t} N_s - (192 + 4 \frac{T}{\Delta t}) > 0 \quad (4-67)$$

If $\frac{T}{\Delta t} = 1.0$, then there is $N_s > 177$ from Equation (4-67).

From the Table 4-3 and Table 4-4, the presented PTI method by step-response, impulsive-response and Duhamel-response matrices with dimensional expansion (A6) will be more efficient than PTI method by step-response and impulsive-response matrices with dimensional expansion (A5) if the following equation is satisfied

$$2nN_s^3 + nN_s^2 \cdot g + n \cdot g^3 + 4N \cdot N_s^3 + 4N \cdot N_s^2 \cdot g + N \cdot g^3 + \quad (4-68)$$

$$(4N_s^2 + 2N_s \cdot g + 2g^2) \frac{T}{\Delta t} < 2n(N_s + g)^3 + 4N(N_s + g)^3 + 4(N_s + g)^2 \frac{T}{\Delta t}$$

Or

$$0 < (5nN_s^2g + 6nN_s g^2 + ng^3) + (8NN_s^2g + 12NN_s g^2 + 3Ng^3) + (6N_s g + 2g^2) \frac{T}{\Delta t} \quad (4-69)$$

Note that Equation (4-69) is always satisfied. Hence, the computational speed of the PTI method by step-response, impulsive-response and Duhamel-response matrices with dimensional expansion (A6) is faster than that of the PTI method by step and impulsive-response matrices with dimensional expansion (A5) normally.

In conclusion, the PTI method by step-response and impulsive-response matrices with dimensional expansion (A5) and the PTI method by step-response, impulsive-response and Duhamel-response matrices with dimensional expansion (A6) can reduce much more computational cost than the dimensional expanding PTI algorithm (A3). The new PTI method by step-response, impulsive-response and Duhamel-response matrices with dimensional expansion (A6) proposed in this chapter will be more efficient than the PTI method by step-response and impulsive-response matrices (A4) if Equation (4-65) is held. The PTI method by step-response, impulsive-response and Duhamel-response matrices with dimensional expansion (A6) is more efficient than the other PTI methods discussed in this research.

4.6 Numerical Examples

Two numerical examples are used to illustrate the efficiency of the presented algorithms. The computational effort comparison of the original PTI method (A1), the dimensional expanding PTI method (A3), the PTI method by step-response and impulsive-response matrices (A4) and the two new PTI methods by response matrices with dimensional expansion (A5 and A6) is given in the examples. The validity and efficiency of the new methods proposed in this chapter are demonstrated.

results show that the computing efficiency can be improved significantly using the new PTI method proposed in this research. With the symmetric property, the PTI method by step-response, impulsive-response and Duhamel-response matrices with dimensional expansion (A6) needs much less computational effort than the other methods (A1, A3 and A5).

(2) Polynomial Excitation

Assume that the external loading is in the form.

$$\{\mathbf{r}(t)\} = e^{\eta t} \left(\{\mathbf{r}_0\} + \{\mathbf{r}_1\} \cdot t + \{\mathbf{r}_2\} \cdot t^2 + \{\mathbf{r}_3\} \cdot t^3 + \{\mathbf{r}_4\} \cdot t^4 + \{\mathbf{r}_5\} \cdot t^5 + \{\mathbf{r}_6\} \cdot t^6 \right) \quad (4-77)$$

where $\eta = 0.01$.

$$\{\mathbf{r}_0\} = \{1, 2, 0, 1, 2, 0, \dots\} \quad (4-78a)$$

$$\{\mathbf{r}_1\} = 10^{-1} \times \{1, 2, 0, 1, 2, 0, \dots\} \quad (4-78b)$$

$$\{\mathbf{r}_2\} = 10^{-2} \times \{1, 2, 0, 1, 2, 0, \dots\} \quad (4-78c)$$

$$\{\mathbf{r}_3\} = 10^{-3} \times \{2, 4, 0, 2, 4, 0, \dots\} \quad (4-78d)$$

$$\{\mathbf{r}_4\} = 10^{-4} \times \{2, 4, 0, 2, 4, 0, \dots\} \quad (4-78e)$$

$$\{\mathbf{r}_5\} = 10^{-5} \times \{2, 4, 0, 2, 4, 0, \dots\} \quad (4-78f)$$

$$\{\mathbf{r}_6\} = 10^{-6} \times \{2, 4, 0, 2, 4, 0, \dots\} \quad (4-78g)$$

As shown in Section 4.2, the related matrices for these two methods are

$$[\mathbf{S}_1] = \begin{bmatrix} \eta & & & & & & \\ 1 & \eta & & & & & \\ & 2 & \eta & & & & \\ & & 3 & \eta & & & \\ & & & 4 & \eta & & \\ & & & & 5 & \eta & \\ & & & & & 6 & \eta \end{bmatrix}, [\mathbf{S}_2] = \begin{bmatrix} \eta^2 & & & & & & \\ 2\eta & \eta^2 & & & & & \\ 2 & 4\eta & \eta^2 & & & & \\ & 6 & 6\eta & \eta^2 & & & \\ & & 12 & 8\eta & \eta^2 & & \\ & & & 20 & 10\eta & \eta^2 & \\ & & & & 30 & 12\eta & \eta^2 \end{bmatrix} \quad (4-79)$$

From the Table 4-6, the PTI method (A4) with the symmetric property is faster than the PTI method by step-response and impulsive-response matrices with dimensional

expansion (A5). However, the PTI method by step-response, impulsive-response and Duhamel-response matrices with dimensional expansion (A6) will cost less than the PTI method by step-response and impulsive-response matrices (A4) because method proposed in this chapter can also make use of the symmetric property of the matrix. Furthermore, the computational cost for computing the particular response in the PTI method by step-response and impulsive-response matrices (A4) will increase with the increasing of the excitation terms. Furthermore, the PTI method by step-response, impulsive-response and Duhamel-response matrices with dimensional expansion (A6) will be more efficient than the PTI method by step-response and impulsive-response matrices (A4) with the increase of the excitation terms. For example, Table 4-5 notes that the PTI method by step-response, impulsive-response and Duhamel-response matrices with dimensional expansion only saves 0.246s computational cost compared with the PTI method by step-response and impulsive-response matrices (A4) when the number of excitation terms is 2. From Table 4-6, it notes that the saved computational cost increases to 0.918s when the number of excitation terms is 7.

(3) Complex Excitation

Consider the following loading form

$$\begin{aligned} \{\mathbf{r}(t)\} &= \sum_{i=1}^3 \{\mathbf{r}_i\} t^{i-1} (\sin(t) + \cos(t)) \\ &= \{\mathbf{r}_1\} \sin(t) + \{\mathbf{r}_1\} \cos(t) + \{\mathbf{r}_2\} t \sin(t) + \{\mathbf{r}_2\} t \cos(t) + \{\mathbf{r}_3\} t^2 \sin(t) + \{\mathbf{r}_3\} t^2 \cos(t) \end{aligned} \quad (4-80)$$

where

$$\{\mathbf{r}_1\} = \{1, 2, 0, 1, 2, 0, \dots\} \quad (4-81a)$$

$$\{\mathbf{r}_2\} = \{0.1, 0.2, 0, 0.1, 0.2, 0, \dots\} \quad (4-81b)$$

$$\{\mathbf{r}_3\} = \{0.01, 0.02, 0, 0.01, 0.02, 0, \dots\} \quad (4-81c)$$

As shown in Section 4.2, the related matrices for these two methods are

$$[\mathbf{S}_1] = \begin{bmatrix} 0 & 1 & & & & 0 \\ -1 & 0 & 0 & & & \\ 1 & 0 & 0 & 1 & & \\ & 1 & -1 & 0 & 0 & \\ & & 2 & 0 & 0 & 1 \\ 0 & & & 2 & -1 & 0 \end{bmatrix}, [\mathbf{S}_2] = \begin{bmatrix} -1 & 0 & & & & 0 \\ 0 & -1 & 0 & & & \\ 0 & 2 & -1 & 0 & & \\ -2 & 0 & 0 & -1 & 0 & \\ 2 & 0 & 0 & 4 & -1 & 0 \\ 0 & 2 & -4 & 0 & 0 & -1 \end{bmatrix} \quad (4-82)$$

For such a complex external loading form, the particular solution will become complicated. This example also compares the efficiency of the current two new methods (A5 and A6). The results are listed in the Table 4-7. The symmetric property still can be used in the PTI method by step-response, impulsive-response and Duhamel-response matrices with dimensional expansion (A6). The PTI method by step-response, impulsive-response and Duhamel-response matrices with dimensional expansion (A6) is more efficient than the PTI method by step-response and impulsive-response matrices with dimensional expansion (A5).

Example 4.2: A Truss Structure with Six Storey and Three Bars

Figure 4-1 shows a truss with six storey and three bars. It has 28 nodal points, 60 truss elements and 48 degrees of freedom. The Young's modulus of each element, whose cross-surface is $5 \times 10^{-3} \text{ m}^2$, is 200 GPa. In the truss structure, the height of each storey and the width of each bar are 3m and 4m , respectively. The lumped mass at each node of the truss is $M_i = 500\text{kg}$, ($i = 5, 6, \dots, 28$). The loading, $\{\mathbf{r}(t)\}$, is a trapezoidal function as shown in the Figure 4-1.

$$\{\mathbf{r}(t)\} = \begin{cases} \frac{6h}{T}(t-iT) & iT - \frac{T}{6} < t < iT + \frac{T}{6} \\ h & iT + \frac{T}{6} \leq t \leq iT + \frac{T}{3} \\ -\frac{6h}{T}(t - \frac{2i+1}{2}T) & \frac{2i+1}{2}T - \frac{T}{6} < t < \frac{2i+1}{2}T + \frac{T}{6} \\ -h & \frac{2i+1}{2}T + \frac{T}{6} \leq t \leq \frac{2i+1}{2}T + \frac{T}{3} \end{cases}, i = 0 \quad 1 \quad 2 \quad \dots \quad (4-83)$$

The excitation can be approximated by Fourier series (Wang et. al., 2002) as

$$\{\mathbf{r}(t)\} \approx \sum_{i=1}^{f_n} a_i (\sin(\omega_i t)) \quad (4-84)$$

$$\text{Where, } a_i = \frac{12h}{(2i-1)^2} \sin\left(\frac{2i-1}{3}\pi\right), \text{ and } \omega_i = (2i-1)\frac{2\pi}{T}$$

In this example, $h = 10^{-4}$, $T = 6.0s$.

(1) Analysis of Periodic Excitation

Normally, if the periodic excitation is described by piecewise functions, the current PTI methods would approximate the external load by Fourier series. The accuracy and efficiency therefore related to the number of the Fourier terms used.

The dynamic problems under periodic excitation can be solved efficiently if the computing matrices need not be computed at every period. With the dimensional expanding method, the excitation can be separated into two parts $[\mathbf{f}]$ and $\{\mathbf{Z}(t)\}$ in every segment of the piecewise excitation $\{\mathbf{r}(t)\}$. The matrix $[\mathbf{S}_1]$ (or $[\mathbf{S}_2]$) is determined by the time-varying function $\{\mathbf{Z}(t)\}$. The formula of function $\{\mathbf{Z}(t)\}$ can be modified so that the matrices $[\mathbf{f}]$ and $[\mathbf{S}_1]$ (or $[\mathbf{S}_2]$) will be remained unchanged in different period. In fact, if the matrices $[\mathbf{f}]$ and $[\mathbf{S}_1]$ (or $[\mathbf{S}_2]$) do not change, the solutions of these response matrices need not be reevaluated every period. Hence the response matrices evaluated in the first period can be stored for other periods. Only initial conditions of $\{\mathbf{Z}(t)\}$ may need to be modified.

Only two kinds of periodic external loading form (continuous and discontinuous) are discussed in this chapter. For this example, in Figure 4-1, the universal formula of the first and fifth part of the piecewise function in every period is

$$r_1 = \frac{6h}{T}(t - iT), \quad iT - \frac{T}{6} < t < iT + \frac{T}{6} \quad (4-85)$$

where T is the period and $i = 0 \quad 1 \quad 2 \quad \dots$

Let $t^* = t - iT$, then the r_1 can be rewritten as: $r_1 = \frac{6h}{T} \cdot t^*$. The matrix $[\mathbf{f}]$ and $[\mathbf{S}_1]$ (or $[\mathbf{S}_2]$) do not change in different period relating to the variable t^* instead of

variable t . Hence the computing matrices only need be recomputed four times at most.

For discontinuous periodic external load, this conversion is more useful because it is more difficult to get a better approximation function with low number of terms. Figure 4-2 shows a simple kind of discontinuous excitation. The general formula can be written as

$$r_1 = h_1 + \frac{h_2 - h_1}{T} t^*, \text{ where } t^* = t - iT \text{ and } i = 0 \ 1 \ 2 \ \dots \quad (4-86)$$

The computing matrices calculated in the first period can be used in the remaining computational process. The only disadvantage is that the time step size must be restricted according to the period. This method can work well when the number of the sub-functions in the piecewise function in one period is not too many. In fact, this method can be used to reduce the computational cost in all dimensional expanding PTI methods.

The time-step integration procedures are the same as the Example 4.1. Two PTI methods (**A3** and **A6**) are investigated. For periodic excitation, more computational cost can be saved using the periodic property instead of Fourier series approximation to the excitation. Only four sets of response matrices need be evaluated in the whole computing process. The computational costs are listed in the Table 4-8. Though, the two PTI methods (**A3** and **A6**) only do one step iterative computation with Fourier series approximation. The computational effort is still very high. The computational costs of the PTI method **A3** using Fourier series approximation and the periodic property described in this section are 17.815s and 3.595s, respectively. The method **A3** using Fourier series approximation is about five times more than using periodic property. As a result, the PTI methods (**A3** and **A6**) using the periodic property described in this section can reduce much more computational costs. This example also investigated that the PTI method by step-response, impulsive-response and Duhamel-response matrices with dimensional expansion (**A6**) are more efficient than dimensional expanding PTI method (**A3**).

4.7 Summary

The precise time-step integration method (A4) is further developed in this chapter. The second-order equation is tackled directly. The dimensional expanding method can also be used to simplify the computational algorithm by transforming the non-homogeneous second-order equation into homogeneous second-order equation. The precise time-step integration method by step-response and impulsive-response matrices (A5) and the precise time-step integration method by step-response, impulsive-response and Duhamel-response matrices (A6) based on the dimensional expanding method for solving the second-order equation directly are proposed in this chapter. The method (A5) simplifies the method (A4). But the computational cost of method (A5) is increased obviously. The new method (A6) is more efficient than the dimensional expanding PTI method (A3) and the PTI method (A5). The new method (A6) is slightly more efficient than the PTI method (A4) when the symmetric property of matrix is used in the algorithm. In the method (A6), the symmetric property can still be used to reduce much computational cost. If the excitation is periodic, more computational effort can be reduced using the conversion described in Example 4.2. Accuracy and efficiency of the new precise time-step integration methods are investigated.

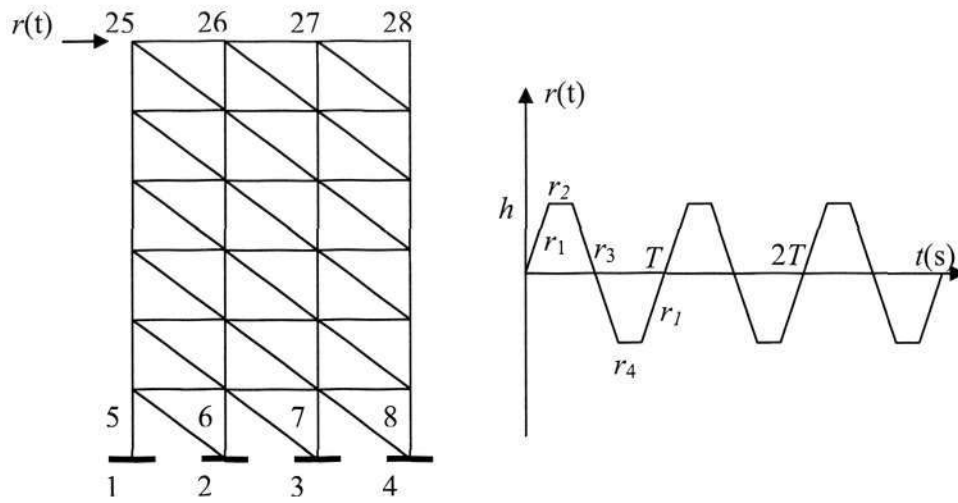


Figure 4-1 The truss with six storey and three bars and the excitation

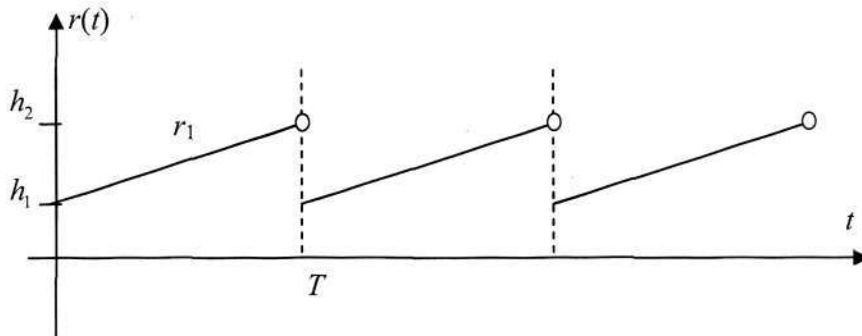


Figure 4-2 The discontinuous periodic external loading form

Table 4-1 The operation count of the method (A3)

Category	Operation count
Form $[\mathbf{T}_a^*]$	$2(2N_s + g)^3$
Form $[\mathbf{T}^*]$	$N \cdot (2N_s + g)^3$
$[\mathbf{T}^*]\{\mathbf{u}^*\}$	$(2N_s + g)^2$

Table 4-2 The operation count of the method (A4)

Category	Operation count
Initial Taylor series approximation	$2nN_s^3$
$[\mathbf{H}]$ and $[\mathbf{J}]$ matrices formulation	$4N \cdot N_s^3$
Particular solution	$[Inte]$
One time-step advancement	$4N_s^2$

Table 4-3 The operation count of the method (A5)

Category	Operation count
Initial Taylor series approximation	$2n \cdot (N_s + g)^3$
[H [*]] and [J [*]] matrices formulation	$4N \cdot (N_s + g)^3$
One time-step advancement	$4(N_s + g)^2 \frac{T}{\Delta t}$

Table 4-4 The operation count of the method (A6)

Category	Operation count
Initial Taylor series approximation	$2nN_s^3 + nN_s^2g + ng^3$
[H], [J], [D] and [E] matrices formulation	$4N \cdot N_s^3 + 4N \cdot N_s^2 \cdot g + N \cdot g^3$
One time-step advancement	$(4N_s^2 + 2N_s g + g^2) \frac{T}{\Delta t}$

Table 4-5 Computational effort (linear excitation) to evaluate the same accurate result u_{N_s} at $t=10s$

Methods	Order of Computing Equation	Result u_{N_s}	Computational cost (s)
Method A1	First-order	9.09462099	149.475
Method A3	First-order	9.09462099	111.520
Method A4	Second-order	9.09462099	53.492
Method A5	Second-order	9.09462099	90.319
Method A6	Second-order	9.09462099	53.246

Notes: $n=4$, $N=20$, $\Delta t = 10.0s$, $N_s=200$.

Table 4-6 Computational effort (Polynomial excitation) to evaluate the same accurate result u_{N_s} at $t=10s$

Methods	Order of Computing Equation	Result u_{N_s}	Computational cost (s)
Method A3	First-order	18.18160768	114.665
Method A4	Second-order	18.18160768	54.304
Method A5	Second-order	18.18160768	97.730
Method A6	Second-order	18.18160768	53.386

Notes: $n=4$, $N=20$, $\Delta t = 10.0s$, $N_s=200$.

Table 4-7 Computational effort (Complex excitation) to evaluate the same accurate result u_{N_s} at $t=10s$

	Computational cost (s)		
	Method A3	Method A5	Method A6
Order of Computing Equation	First-order	Second-order	Second-order
$N_s=100$	15.164	7.663s	5.190
$N_s=200$	112.060	95.826s	53.306

Notes: $n=4, N=20, \Delta t = 10.0s, \text{Error}=10^{-7}$

Table 4-8 Computational effort to evaluate the results at the end of two periods ($t=12.0s$) in Example 4.2

Excitation form	Time step size	Computational cost (s)	
		Method A3	Method A6
Fourier series approximation with $f_s=60$	$\Delta t = 12.0s$	17.815	3.995
Periodic property Described in Section 4.6	$\Delta t = 1.0s$	3.595	0.671

Notes: $n=4, N=20, \text{Error}=10^{-7}, f_s$ is the number of terms of Fourier series.

CHAPTER FIVE

Unconditionally Stable PTI Algorithms by Padé Approximations

Overview

This chapter investigates the computational advantages of using the Padé approximations to replace the Taylor approximations in the precise time-step integration method. Both the PTI methods for the first and second order equations are considered. The new algorithms become unconditionally stable using the Padé approximations. For first-order equations, the existing ordinary Padé approximations can be used. For second-order equations, new Padé approximations to generate unconditionally stable algorithms are derived for MDOF systems with Rayleigh damping. It is also shown in this chapter that by making use of the symmetric properties and the special forms of the new Padé approximations, the computational efforts of the improved unconditionally stable algorithms are comparable to the original algorithms which are only conditionally stable with comparable accuracy.

5.1 Introduction

The original PTI method (**A1**) and the PTI method by step-response and impulsive-response matrices (**A4**) are further developed in this chapter. More details about these two methods are discussed in Chapter 3 and Chapter 4 respectively. They are all conditionally stable, though the computing time step can be chosen large. The Padé approximations are used to improve the stability properties of the PTI methods in this chapter. Furthermore, the computational efforts are also comparable to the existing algorithms.

Normally, Taylor series approximation is employed to evaluate the initial matrices for recursive evaluation in these two PTI methods (**A1** and **A4**). Theoretically, the algorithm using Taylor series approximation is simple and efficient. The use of Taylor series generates only conditionally stable algorithms and the stable ranges are very limited. Moler and Loan (1978) gave a comprehensive review on the computation of the exponential matrix function. Padé approximations can be employed to make current PTI methods more efficient and stable.

5.2 Padé Approximations

Padé approximations are natural generalization of Taylor polynomials. Bendazzoli et al. (1980) presented and discussed a Padé-type algorithm suitable for large scale systems and having better convergence properties than the classical Jacobi procedure. Carter et al. (1984) proposed the higher order implicit integration techniques for dynamic response equations utilizing the conjugate gradient method and Padé approximations. Later, Cochelin et al. (1994) applied asymptotic numerical methods for computing non-linear equilibrium paths of elastic beam, plate and shell structures. They made use of Padé approximations to control the size of the domain of convergence. Fung (1999) presented an unconditionally stable higher-order accurate time step integration algorithms suitable for linear first-

order differential equations based on the weighted residual method. The generalized Padé approximations were employed to determine the weighting parameters. The various details and theorems on the issue of convergence of Padé approximations are given in the research (Baker and Graves-Morris, 1996; Lubinsky, 1997). For completeness the basic theories of Padé approximations are given in this chapter.

The ordinary Padé series approximation can derive the more accurate solution of $[\mathbf{T}(\Delta t)] = \exp([\mathbf{W}] \cdot \Delta t)$ than that using Taylor series approximation. The ordinary (p, q) Padé series approximations to $\exp([\mathbf{W}] \cdot \Delta t)$ is defined by

$$\mathbf{P}_{pq}([\mathbf{W}] \cdot \Delta t) = [\mathbf{Q}_{pq}([\mathbf{W}] \cdot \Delta t)]^{-1} \mathbf{N}_{pq}([\mathbf{W}] \cdot \Delta t), \quad (5-1)$$

where

$$\mathbf{N}_{pq}([\mathbf{W}] \cdot \Delta t) = \sum_{j=0}^p \frac{(p+q-j)! p!}{(p+q)! j! (p-j)!} ([\mathbf{W}] \cdot \Delta t)^j \quad (5-2)$$

And

$$\mathbf{Q}_{pq}([\mathbf{W}] \cdot \Delta t) = \sum_{j=0}^q \frac{(p+q-j)! q!}{(p+q)! j! (q-j)!} (-[\mathbf{W}] \cdot \Delta t)^j \quad (5-3)$$

The diagonal Padé approximation with $p = q$ are better than the off-diagonal Padé approximation with $p \neq q$. The argument is as follows. Suppose $p < q$, then, qN_s^3 flops are required to evaluate $\mathbf{P}_{pq}([\mathbf{W}] \cdot \Delta t)$, and the order of accuracy is $(p+q)$. The same amount of work is needed to compute $\mathbf{P}_q([\mathbf{W}] \cdot \Delta t)$ and the order of accuracy $2q > p+q$. A similar argument can be applied to the other off-diagonal Padé approximations with $p > q$. In view of the above, the diagonal Padé series approximation with $(p = q)$ are used.

The ordinary diagonal (p, p) Padé series approximation can improve the accuracy and stability in stead of Taylor series approximations in the PTI algorithm for computing exponential matrix. Hence the exponential matrix can be written as

$$\exp([\mathbf{W}] \cdot \Delta t) \cong \mathbf{P}_p([\mathbf{W}] \cdot \Delta t) = [\mathbf{Q}_p([\mathbf{W}] \cdot \Delta t)]^{-1} [\mathbf{N}_p([\mathbf{W}] \cdot \Delta t)]. \quad (5-4)$$

5.3 Scaling and Squaring Algorithms for Evaluation of Exponential Matrix

As mentioned in Chapter 3, the main feature of the original PTI method is the precise computation of the exponential matrix. The PTI methods proposed for a linear time invariant dynamic system can give the accurate numerical results at the time integration points. Besides, the algorithms using Taylor series approximation will not be very efficient and not be unconditionally stable. Padé approximations are introduced to overcome the disadvantage. The precise time-step integration method incorporating with ordinary Padé series approximation is denoted as Algorithm 7 (A7).

With the scaling and squaring techniques, an acceptable accuracy can be obtained even for a small degree p if we make use of the exponential property

$$[\mathbf{T}(\Delta t)] = \exp([\mathbf{W}] \cdot \Delta t) = \{\exp([\mathbf{W}] \cdot \tau)\}^{2^N} \quad (5-5)$$

where $\tau = \Delta t / 2^N$, and N is an integer. If the time step Δt is not very large, τ will be extremely small. To initiate the iterations, the following truncated Taylor series can be used to evaluate $\exp([\mathbf{W}] \cdot \tau)$:

$$\exp([\mathbf{W}] \times \tau) \cong [\mathbf{I}] + [\mathbf{T}_a] \quad (5-6)$$

where

$$[\mathbf{T}_a] = ([\mathbf{W}] \times \tau) + \frac{([\mathbf{W}] \times \tau)^2}{2} + \frac{([\mathbf{W}] \times \tau)^3}{3!} + \frac{([\mathbf{W}] \times \tau)^4}{4!} \quad (5-7)$$

With the scaling and squaring techniques, the high precise solution of the exponential matrix can be obtained by the truncated Taylor series. But this method is conditionally stable. The convergence of this method will be influenced by the scaling factor N . For large Δt , the results may diverge if the value of N is not enough large.

The diagonal (p, p) Padé series approximation to the exponential function in PTI method is considered. The exponential matrix $\exp([\mathbf{W}] \cdot \Delta t)$ is to be approximated by $\mathbf{P}_p(\mathbf{W} \cdot \Delta t)$ as

$$\begin{aligned} \exp([\mathbf{W}] \cdot \Delta t) &\approx (\mathbf{P}_p([\mathbf{W}] \cdot \tau))^{2^N} = ([\mathbf{I}] + [\mathbf{Q}_p([\mathbf{W}] \cdot \tau)]^{-1} \mathbf{N}_p([\mathbf{W}] \cdot \tau) - [\mathbf{I}])^{2^N} \\ &= ([\mathbf{I}] + [\mathbf{T}_a^*])^{2^N} \end{aligned} \quad (5-8)$$

The formula of $[\mathbf{T}_a^*]$ with $p=1, 2, 3, 4$ and 5 are listed in Table 5-1. The matrix $[\mathbf{T}_a^*]$ should be computed and stored to avoid rounding errors. Eventually, the scaling and squaring procedure can be used to compute $\exp([\mathbf{W}] \cdot \Delta t)$ by executing the following instruction:

$$\text{For } (i=0; i<N; i++) [\mathbf{T}_a^*] = 2 \times [\mathbf{T}_a^*] + [\mathbf{T}_a^*] \times [\mathbf{T}_a^*] \quad (5-9)$$

$$\text{Then, } [\mathbf{T}] = ([\mathbf{I}] + [\mathbf{T}_a^*]) \quad (5-10)$$

An inverse error analysis of the ordinary Padé series approximation given by Moler and Loan (1978) can be used. If

$$\frac{\|\Delta t \cdot \mathbf{W}\|_2}{2^N} \leq \frac{1}{2}, \quad (5-11)$$

then the inverse error (ε) of the Padé series approximation can be written as:

$$\begin{aligned} \varepsilon &\leq 8 \left[\frac{\|\Delta t \cdot \mathbf{W}\|_2}{2^N} \right]^{2p} \left(\frac{(p!)^2}{(2p)!(2p+1)!} \right) \\ &\leq \frac{(p!)^2}{(2p)!(2p+1)!} \left(\frac{1}{2} \right)^{2p-3} \approx \begin{cases} 0.694 \times 10^{-3} & (p=2) \\ 0.124 \times 10^{-6} & (p=3) \\ 0.123 \times 10^{-8} & (p=4) \\ 0.777 \times 10^{-12} & (p=5) \end{cases} \end{aligned} \quad (5-12)$$

This analysis provided a criterion for selecting the diagonal Padé series approximation for a particular computer. $p=2$ or 4 is recommended to be used in this research.

In fact, the improvement of accuracy will not be significant when the value of N is increased because the accuracy does not only depend on the computational procedure but also the accuracy of the selecting approach series. Furthermore, the

computational cost will be increased when the value of N is increased so that these two kinds of situations should be considered together. Equation (5-11) can be used to determine the value of N . More details of the implementation of this property are given in Chapter 6.

In Equation (5-8), since the time interval Δt is not large, $\tau = \Delta t / 2^N$ will be an extremely small time interval. For a given Δt , the value of N should be chosen carefully so that the algorithm will be efficient. To give better results, a smaller $\Delta t / 2^N$ may be required. In fact, a more accurate series approximation can derive more accurate results without increasing the value of N . Moler and Loan (1978) gave a comparison for optimum scaling and squaring parameters with diagonal Padé and Taylor series approximation. It is noted that the Padé series approximation are generally more efficient than the Taylor series approximation when $\|[\mathbf{W}] \cdot \Delta t\|_2$ is not too large. In this chapter, the Padé approximations are considered to be used in the PTI method by step-response and impulsive-response matrices instead of Taylor series approximations to improve the stability.

5.4 Padé Approximations for $[\mathbf{H}]$ and $[\dot{\mathbf{H}}]$

The PTI method by step-response and impulsive-response matrices (A4) extended the stable range. To further improve the stability of this PTI method, the Padé approximations can be used to evaluate the initial matrices $[\mathbf{H}(\tau)]$ and $[\mathbf{J}(\tau)] = [\dot{\mathbf{H}}(\tau)]$. Two kinds of Padé approximations are considered.

5.4.1 Ordinary Padé Series Approximation

First, the ordinary Padé series approximation is considered. The single-degree-of-freedom system can be written as

$$m_s \cdot \ddot{u}(t) + c_s \cdot \dot{u}(t) + k_s \cdot u(t) = 0 \quad (5-13)$$

From the research (Fung, 1997), the displacement response $u(t)$ and the velocity response $v(t)$ with given initial conditions u_0 and v_0

$$\begin{Bmatrix} u(t) \\ v(t) \end{Bmatrix} = \begin{bmatrix} g(t) & h(t) \\ \dot{g}(t) & \dot{h}(t) \end{bmatrix} \begin{Bmatrix} u_0 \\ v_0 \end{Bmatrix}. \quad (5-14)$$

Normally, the response function $h(t)$ and derivative $\dot{h}(t)$ at $t = \tau$ are computed by n -term Taylor series approximation which is only conditionally stable.

$$h(\tau) \cong \dot{h}(0) \cdot \tau + \ddot{h}(0) \cdot \tau^2/2! + h^{(3)}(0) \cdot \tau^3/3! + h^{(4)}(0) \cdot \tau^4/4! + \dots + h^{(n)}(0) \cdot \tau^n/n! \quad (5-15a)$$

$$\dot{h}(\tau) \cong \dot{h}(0) + \ddot{h}(0) \cdot \tau + h^{(3)}(0) \cdot \tau^2/2! + h^{(4)}(0) \cdot \tau^3/3! + \dots + h^{(n)}(0) \cdot \tau^{n-1}/(n-1)! \quad (5-15b)$$

Where

$$\dot{h}(0) = 1, \quad \ddot{h}(0) = b, \quad h^{(i+2)}(0) = h^{(i)}(0) \cdot a + h^{(i+1)}(0) \cdot b, \quad i = 1, 2, 3, \dots \quad (5-15c)$$

$$a = -m_s^{-1}k_s \quad \text{and} \quad b = -m_s^{-1}c_s$$

With the ordinary (p, p) diagonal Padé series approximation, these functions $h(\tau)$ and $\dot{h}(\tau)$ can be written directly as

$$\begin{aligned} h(\tau) &\cong \left(\sum_{i=0}^p q_i \cdot \tau^i \right)^{-1} \cdot \left(\sum_{i=0}^p d_i \cdot \tau^i \right) \cdot \tau \\ &= \left(1 + q_1 \cdot \tau + q_2 \cdot \tau^2 + \dots + q_p \cdot \tau^p \right)^{-1} \cdot \left(\tau + d_1 \cdot \tau^2 + d_2 \cdot \tau^3 + \dots + d_p \cdot \tau^{p+1} \right) \end{aligned} \quad (5-16a)$$

$$\begin{aligned} \dot{h}(\tau) &\cong \left(\sum_{i=0}^p q'_i \cdot \tau^i \right)^{-1} \cdot \left(\sum_{i=0}^p d'_i \cdot \tau^i \right) \\ &= \left(1 + q'_1 \cdot \tau + q'_2 \cdot \tau^2 + \dots + q'_p \cdot \tau^p \right)^{-1} \cdot \left(1 + d'_1 \cdot \tau + d'_2 \cdot \tau^2 + \dots + d'_p \cdot \tau^p \right) \end{aligned} \quad (5-16b)$$

The formulae of q'_i and d'_i are almost the same as q_i and d_i . The only difference is that the parameters, α_{ij} and β_{ij} are denoted as α'_{ij} and β'_{ij} in q'_i and d'_i . The first four-order q_i and d_i are given in Table 5-2. The higher order q_i and d_i (q'_i and d'_i) can be derived from the Table 5-2 with deductive method.

The parameter equations for solving α_{ij} , β_{ij} , α'_{ij} and β'_{ij} could be evaluated by comparing the right part in Equation (5-15) with the right part in Equation (5-16) under the condition $2p \leq n$ as following:

$$\sum_{j=0}^{i-1} q_j \cdot h^{(i-j)}(0) \cdot \frac{1}{(i-j)!} = d_{(i-1)}, \quad i = 2, 3, \dots, p+1 \quad (5-17a)$$

$$\sum_{j=0}^p q_j \cdot h^{(i-j)}(0) \cdot \frac{1}{(i-j)!} = 0, \quad i = p+2, p+3, \dots, 2p+1 \quad (5-17b)$$

and

$$\sum_{j=0}^{i-1} q'_j \cdot h^{(i-j)}(0) \cdot \frac{1}{(i-j-1)!} = d'_{(i-1)}, \quad i = 2, 3, \dots, p+1 \quad (5-18a)$$

$$\sum_{j=0}^p q'_j \cdot h^{(i-j)}(0) \cdot \frac{1}{(i-j-1)!} = 0, \quad i = p+2, p+3, \dots, 2p+1 \quad (5-18b)$$

From Equations (5-17a-b) and (5-18a-b), the parameters (α_{ij} , β_{ij} , α'_{ij} and β'_{ij}) can be determined. For example, the coefficients (α_{ij} , β_{ij}) of ordinary (2, 2) diagonal Padé series approximation can be obtained form

$$q_0 \cdot \ddot{h}(0) \frac{1}{2!} + q_1 \cdot \dot{h}(0) = d_1 \quad (5-19a)$$

$$q_0 \cdot h^{(3)}(0) \frac{1}{3!} + q_1 \cdot \ddot{h}(0) \frac{1}{2!} + q_2 \cdot \dot{h}(0) = d_2 \quad (5-19b)$$

$$q_0 \cdot h^{(4)}(0) \frac{1}{4!} + q_1 \cdot h^{(3)}(0) \frac{1}{3!} + q_2 \cdot \ddot{h}(0) \frac{1}{2!} = 0 \quad (5-19c)$$

$$q_0 \cdot h^{(5)}(0) \frac{1}{5!} + q_1 \cdot h^{(4)}(0) \frac{1}{4!} + q_2 \cdot h^{(3)}(0) \frac{1}{3!} = 0 \quad (5-19d)$$

With Table 5-2, Equations (5-15c) and (5-19a-d), the coefficients α_{11} , α_{21} , α_{22} , β_{11} , β_{21} , and β_{22} can be evaluated by comparing the coefficients of a , b , ab , a^2 , b^2 and b^3 , respectively as following

$$b \cdot \frac{1}{2} + \alpha_{11} \cdot b = \beta_{11} \cdot b \quad (5-20a)$$

$$a \cdot \frac{1}{3!} + \alpha_{21} \cdot a = \beta_{21} \cdot a \quad (5-20b)$$

$$2ab \cdot \frac{1}{4!} + \alpha_{11}ab \cdot \frac{1}{3!} + \alpha_{21}ab \cdot \frac{1}{2} = 0 \quad (5-20c)$$

$$a^2 \cdot \frac{1}{5!} + \alpha_{21} \cdot a^2 \cdot \frac{1}{3!} = 0 \quad (5-20d)$$

$$b^2 \cdot \frac{1}{3!} + \alpha_{11}b^2 \cdot \frac{1}{2} + \alpha_{22} \cdot b^2 = \beta_{22} \cdot b^2 \quad (5-20e)$$

$$b^3 \cdot \frac{1}{4!} + \alpha_{11}b^3 \cdot \frac{1}{3!} + \alpha_{22}b^3 \cdot \frac{1}{2} = 0 \quad (5-20f)$$

Further study the ordinary Padé series approximation to $h(\tau)$ and $\dot{h}(\tau)$, it should note that the algorithm by the ordinary Padé series approximation is still conditionally stable, though the stable range is extended. More details are given in Section 5.6. The computational cost will be very high in evaluating $[\mathbf{H}(\tau)]$ and $[\mathbf{J}(\tau)] = [\dot{\mathbf{H}}(\tau)]$ in multi-degree-of-freedom system by ordinary Padé series approximation directly because there are two inverse matrices need be evaluated. Hence the main work is to find the unconditionally stable and efficient Padé approximation.

5.4.2 New Padé Approximation

A new efficient Padé approximation is developed to approximate $[\mathbf{H}(\tau)]$ and $[\mathbf{J}(\tau)]$ in this section. It is found that the new Padé approximation to these two matrices can be obtained directly from the solutions of first-order equations. This Padé

approximation can be employed to compute the matrices $[\mathbf{H}(\tau)]$ and $[\mathbf{J}(\tau)]$ expecting to obtain the unconditionally stable property. With the scaling and squaring method, low order Padé approximations can be employed without loss of high accuracy. Hence, only the first four-order explicit formula of the current new Padé approximation is considered. The PTI method by step-response and impulsive-response matrices with the new Padé approximation is defined as Algorithm 8, which is denoted as **A8**. This method will require less computational cost than the original precise time-step integration methods because that the matrices are expressed in terms of two symmetric matrices that can also be evaluated recursively. Both single-degree-of-freedom system and multi-degree-of-freedom system are considered.

(1) Single-Degree-of-Freedom System

In the following, the new Padé approximation for single-degree-of-freedom system (5-13) is considered firstly. To obtain an unconditionally stable Padé approximation, rewrite the solution of Equation (5-13) as

$$\begin{Bmatrix} u(t) \\ v(t) \end{Bmatrix} = \exp\left(\begin{bmatrix} 0 & 1 \\ a & b \end{bmatrix} \cdot t\right) \cdot \begin{Bmatrix} u_0 \\ v_0 \end{Bmatrix} \quad (5-21)$$

The exponential function expanded by diagonal (p, p) Padé series approximation can be written as

$$\begin{aligned} \exp\left(\begin{bmatrix} 0 & 1 \\ a & b \end{bmatrix} \cdot t\right) &\cong P_p\left(\begin{bmatrix} 0 & 1 \\ a & b \end{bmatrix} \cdot t\right) = [\mathbf{Q}_p(t)]^{-1} \cdot [\mathbf{N}_p(t)] \\ &= \left\{ \sum_{j=0}^p \frac{(2p-j)!p!}{(2p)!j!(p-j)!} \left(\begin{bmatrix} 0 & 1 \\ a & b \end{bmatrix} \cdot t\right)^j \right\}^{-1} \cdot \left\{ \sum_{j=0}^p \frac{(2p-j)!p!}{(2p)!j!(p-j)!} \left(-\begin{bmatrix} 0 & 1 \\ a & b \end{bmatrix} \cdot t\right)^j \right\} \end{aligned} \quad (5-22)$$

The final result of Equation (5-22) can be written as

$$\exp\left(\begin{bmatrix} 0 & 1 \\ a & b \end{bmatrix} \cdot t\right) \cong \begin{bmatrix} Q_{11} & Q_{12} \\ Q_{21} & Q_{22} \end{bmatrix}^{-1} \cdot \begin{bmatrix} N_{11} & N_{12} \\ N_{21} & N_{22} \end{bmatrix} \quad (5-23)$$

$$= (Q_{11}Q_{22} - Q_{12}Q_{21})^{-1} \begin{bmatrix} Q_{22}N_{11} - Q_{12}N_{21} & Q_{22}N_{12} - Q_{12}N_{22} \\ Q_{11}N_{21} - Q_{21}N_{11} & Q_{11}N_{22} - Q_{21}N_{12} \end{bmatrix}$$

where Q_{ij} and N_{ij} are factors of matrices $[Q_p(t)]$ and $[N_p(t)]$.

Hence the new Padé approximation to the response function $g(t)$, $h(t)$, $\dot{g}(t)$ and $\dot{h}(t)$ can be generated from Equation (5-23) directly as

$$g(t) = (Q_{11}Q_{22} - Q_{12}Q_{21})^{-1} \cdot (Q_{22}N_{11} - Q_{12}N_{21}) \quad (5-24a)$$

$$h(t) = (Q_{11}Q_{22} - Q_{12}Q_{21})^{-1} \cdot (Q_{22}N_{12} - Q_{12}N_{22}) \quad (5-24b)$$

$$\dot{g}(t) = (Q_{11}Q_{22} - Q_{12}Q_{21})^{-1} \cdot (Q_{11}N_{21} - Q_{21}N_{11}) \quad (5-24c)$$

$$\dot{h}(t) = (Q_{11}Q_{22} - Q_{12}Q_{21})^{-1} \cdot (Q_{11}N_{22} - Q_{21}N_{12}) \quad (5-24d)$$

The first four-order formulae of these response functions are given in Table 5-3. From Table 5-3, it also investigates that the solutions of $g(t)$ and $\dot{g}(t)$ can be obtained finally by

$$g(t) = \dot{h}(t) - h(t) \cdot b \quad (5-25a)$$

$$\dot{g}(t) = h(t) \cdot a \quad (5-25b)$$

(2) Multi-Degree-of-Freedom System

Consider the multi-degree-of-freedom system given by

$$[M]\{\ddot{\mathbf{u}}(t)\} + [C]\{\dot{\mathbf{u}}(t)\} + [K]\{\mathbf{u}(t)\} = \{\mathbf{r}(t)\}. \quad (5-26)$$

Similar to the analysis of single-degree-of-freedom, the response matrices can be obtained by following

$$\begin{bmatrix} \mathbf{G}(t) & \mathbf{H}(t) \\ \dot{\mathbf{G}}(t) & \dot{\mathbf{H}}(t) \end{bmatrix} \cong \exp\left(\begin{bmatrix} \mathbf{0} & \mathbf{I} \\ \mathbf{A} & \mathbf{B} \end{bmatrix} \cdot t\right) \cong \mathbf{P}_p\left(\begin{bmatrix} \mathbf{0} & \mathbf{I} \\ \mathbf{A} & \mathbf{B} \end{bmatrix} \cdot t\right) \quad (5-27)$$

$$= [\mathbf{Q}_p(t)]^{-1} [\mathbf{N}_p(t)] = \begin{bmatrix} \mathbf{Q}_{11} & \mathbf{Q}_{12} \\ \mathbf{Q}_{21} & \mathbf{Q}_{22} \end{bmatrix}^{-1} \begin{bmatrix} \mathbf{N}_{11} & \mathbf{N}_{12} \\ \mathbf{N}_{21} & \mathbf{N}_{22} \end{bmatrix}$$

Where $[Q_{ij}]$ and $[N_{ij}]$ are the sub-matrix of matrices $[Q_p(t)]$ and $[N_p(t)]$. They are the functions including the matrix $[A]$ and $[B]$.

Normally, the rational approximation formulae of response matrices $[\mathbf{G}(t)]$ and $[\mathbf{H}(t)]$ ($[\dot{\mathbf{G}}(t)]$ and $[\dot{\mathbf{H}}(t)]=[\mathbf{J}(t)]$) can be computed directly from Equation (5-27) as following

$$\begin{aligned} \begin{bmatrix} \mathbf{G}(t) & \mathbf{H}(t) \\ \dot{\mathbf{G}}(t) & \dot{\mathbf{H}}(t) \end{bmatrix} &\cong \begin{bmatrix} \tilde{\mathbf{Q}}_{11} & \tilde{\mathbf{Q}}_{12} \\ \tilde{\mathbf{Q}}_{21} & \tilde{\mathbf{Q}}_{22} \end{bmatrix} \begin{bmatrix} \mathbf{N}_{11} & \mathbf{N}_{12} \\ \mathbf{N}_{21} & \mathbf{N}_{22} \end{bmatrix} \\ &= \begin{bmatrix} \tilde{\mathbf{Q}}_{11}\mathbf{N}_{11} + \tilde{\mathbf{Q}}_{12}\mathbf{N}_{21} & \tilde{\mathbf{Q}}_{11}\mathbf{N}_{12} + \tilde{\mathbf{Q}}_{12}\mathbf{N}_{22} \\ \tilde{\mathbf{Q}}_{21}\mathbf{N}_{11} + \tilde{\mathbf{Q}}_{22}\mathbf{N}_{21} & \tilde{\mathbf{Q}}_{21}\mathbf{N}_{12} + \tilde{\mathbf{Q}}_{22}\mathbf{N}_{22} \end{bmatrix} \end{aligned} \quad (5-28)$$

where

$$[\tilde{\mathbf{Q}}_{11}] = [\mathbf{Q}_{11}]^{-1} + ([\mathbf{Q}_{11}]^{-1}[\mathbf{Q}_{12}]) \cdot [\tilde{\mathbf{Q}}_{22}]([\mathbf{Q}_{21}][\mathbf{Q}_{11}]^{-1}) \quad (5-29a)$$

$$[\tilde{\mathbf{Q}}_{21}] = -[\tilde{\mathbf{Q}}_{22}] \cdot ([\mathbf{Q}_{21}][\mathbf{Q}_{11}]^{-1}) \quad (5-29b)$$

$$[\tilde{\mathbf{Q}}_{12}] = -([\mathbf{Q}_{11}]^{-1}[\mathbf{Q}_{12}]) \cdot [\tilde{\mathbf{Q}}_{22}] \quad (5-29c)$$

$$[\tilde{\mathbf{Q}}_{22}] = ([\mathbf{Q}_{22}] - [\mathbf{Q}_{21}][\mathbf{Q}_{11}]^{-1}[\mathbf{Q}_{12}])^{-1} \quad (5-29d)$$

From Equations (5-29a-d), it should note that the computational cost will become higher because two matrices need be inverted when the damping matrices are general. The computing process for getting the unconditionally stable Padé approximation will be very complex. In order to simply the computing process, the Rayleigh damping approximation is imported to overcome this disadvantage.

5.4.2.1 Implementation of Rayleigh Damping

In dynamic analysis of structures, damping plays an important role. It is important to analysis the dynamic response including the damping influence. In general, Rayleigh damping can be used. With this damping, the formulae of new Padé approximation would be greatly simplified. The present formulation in single-degree-of-freedom system can also be applied to the multi-degree-of-freedom system directly.

In Rayleigh damping, the damping matrix is defined as a linear combination of the mass matrix and stiffness matrix as follows.

$$[\mathbf{C}] = \alpha[\mathbf{M}] + \beta[\mathbf{K}] \quad (5-30)$$

where α and β are pre-defined constants. The coefficients α and β can be determined when any two modal damping ratios and the corresponding modal frequencies are specified (Bathe, 1982).

Considering Rayleigh damping in Equation (5-30), the matrix $[\mathbf{B}]$ will be written as

$$[\mathbf{B}] = -[\mathbf{M}]^{-1}[\mathbf{C}] = -\alpha[\mathbf{I}] + \beta[\mathbf{A}] \quad (5-31)$$

Then

$$[\mathbf{A}][\mathbf{B}] = [\mathbf{A}] \cdot (-\alpha[\mathbf{I}] + \beta[\mathbf{A}]) = [\mathbf{B}][\mathbf{A}] \quad (5-32)$$

As a result, the functions $[\mathbf{Q}_{ij}]$ and $[\mathbf{N}_{ij}]$ only include matrix $[\mathbf{A}]$. Then,

$$[\mathbf{Q}_{ij}][\mathbf{Q}_{ji}] = [\mathbf{Q}_{ji}][\mathbf{Q}_{ij}] \quad (5-33a)$$

$$[\mathbf{N}_{ij}][\mathbf{N}_{ji}] = [\mathbf{N}_{ji}][\mathbf{N}_{ij}] \quad (5-33b)$$

$$[\mathbf{Q}_{ij}][\mathbf{N}_{ij}] = [\mathbf{N}_{ij}][\mathbf{Q}_{ij}] \quad (5-33c)$$

Hence the inverse matrix in Equation (5-27) can be written as

$$\begin{aligned} & \begin{bmatrix} \mathbf{Q}_{11} & \mathbf{Q}_{12} \\ \mathbf{Q}_{21} & \mathbf{Q}_{22} \end{bmatrix}^{-1} \\ &= \begin{bmatrix} (\mathbf{Q}_{11}\mathbf{Q}_{22} - \mathbf{Q}_{12}\mathbf{Q}_{21})^{-1} & \mathbf{0} \\ \mathbf{0} & (\mathbf{Q}_{11}\mathbf{Q}_{22} - \mathbf{Q}_{12}\mathbf{Q}_{21})^{-1} \end{bmatrix} \begin{bmatrix} \mathbf{Q}_{22} & -\mathbf{Q}_{12} \\ -\mathbf{Q}_{21} & \mathbf{Q}_{11} \end{bmatrix} \end{aligned} \quad (5-34)$$

Substituting Equation (5-34) into Equation (5-27),

$$\begin{bmatrix} \mathbf{G}(t) & \mathbf{H}(t) \\ \dot{\mathbf{G}}(t) & \dot{\mathbf{H}}(t) \end{bmatrix} \cong \begin{bmatrix} \mathbf{Q}_{11} & \mathbf{Q}_{12} \\ \mathbf{Q}_{21} & \mathbf{Q}_{22} \end{bmatrix}^{-1} \begin{bmatrix} \mathbf{N}_{11} & \mathbf{N}_{12} \\ \mathbf{N}_{21} & \mathbf{N}_{22} \end{bmatrix} \quad (5-35)$$

$$\begin{aligned} &= \begin{bmatrix} (\mathbf{Q}_{11}\mathbf{Q}_{22} - \mathbf{Q}_{12}\mathbf{Q}_{21})^{-1} & \mathbf{0} \\ \mathbf{0} & (\mathbf{Q}_{11}\mathbf{Q}_{22} - \mathbf{Q}_{12}\mathbf{Q}_{21})^{-1} \end{bmatrix} \\ &\quad \cdot \begin{bmatrix} \mathbf{Q}_{22}\mathbf{N}_{11} - \mathbf{Q}_{12}\mathbf{N}_{21} & \mathbf{Q}_{22}\mathbf{N}_{12} - \mathbf{Q}_{12}\mathbf{N}_{22} \\ \mathbf{Q}_{11}\mathbf{N}_{21} - \mathbf{Q}_{21}\mathbf{N}_{11} & \mathbf{Q}_{11}\mathbf{N}_{22} - \mathbf{Q}_{21}\mathbf{N}_{12} \end{bmatrix} \end{aligned} \quad (5-36)$$

Then, the formulae of response matrices $[\mathbf{G}(t)]$, $[\mathbf{H}(t)]$, $[\dot{\mathbf{G}}(t)]$ and $[\dot{\mathbf{H}}(t)] = [\mathbf{J}(t)]$ can be obtained directly as following

$$[\mathbf{G}(t)] = ([\mathbf{Q}_{11}][\mathbf{Q}_{22}] - [\mathbf{Q}_{12}][\mathbf{Q}_{21}])^{-1} \cdot ([\mathbf{Q}_{22}][\mathbf{N}_{11}] - [\mathbf{Q}_{12}][\mathbf{N}_{21}]) \quad (5-37a)$$

$$[\mathbf{H}(t)] = ([\mathbf{Q}_{11}][\mathbf{Q}_{22}] - [\mathbf{Q}_{12}][\mathbf{Q}_{21}])^{-1} \cdot ([\mathbf{Q}_{22}][\mathbf{N}_{12}] - [\mathbf{Q}_{12}][\mathbf{N}_{22}]) \quad (5-37b)$$

$$[\dot{\mathbf{G}}(t)] = ([\mathbf{Q}_{11}][\mathbf{Q}_{22}] - [\mathbf{Q}_{12}][\mathbf{Q}_{21}])^{-1} \cdot ([\mathbf{Q}_{11}][\mathbf{N}_{21}] - [\mathbf{Q}_{21}][\mathbf{N}_{11}]) \quad (5-37c)$$

$$[\mathbf{J}(t)] = ([\mathbf{Q}_{11}][\mathbf{Q}_{22}] - [\mathbf{Q}_{12}][\mathbf{Q}_{21}])^{-1} \cdot ([\mathbf{Q}_{11}][\mathbf{N}_{22}] - [\mathbf{Q}_{21}][\mathbf{N}_{12}]) \quad (5-37d)$$

Equations (5-37a-d) have the similar structure as Equations (5-24a-d). In fact, it shown that the formulae of response matrices $[\mathbf{H}(t)]$ and $[\mathbf{J}(t)] = [\dot{\mathbf{H}}(t)]$ can use the formulae in Table 5-3 by replacing a and b with $[\mathbf{A}]$ and $[\mathbf{B}]$. In fact, the matrices $[\mathbf{G}(t)]$ and $[\dot{\mathbf{G}}(t)]$ can also be determined form $[\mathbf{H}(t)]$ and $[\mathbf{J}(t)]$ directly as

$$[\mathbf{G}(t)] = [\dot{\mathbf{H}}(t)] - [\mathbf{H}(t)] \cdot [\mathbf{B}] \quad (5-38a)$$

$$[\dot{\mathbf{G}}(t)] = [\mathbf{H}(t)] \cdot [\mathbf{A}] \quad (5-38b)$$

There only one matrix needs to be inverted in all computing formulae of this Padé approximation because the denominators are the same in Equations (5-37a-d). This property will save much computational effort and make the computing process algorithm be simplified. Furthermore, the symmetric property described in the research (Fung, 1997) can also be used in this chapter to reduce the computational cost.

5.4.2.2 Symmetric Matrices

The implementation of symmetric property has been introduced in Chapter 4. This method can also be used to reduce the computational cost of the PTI method by step-response and impulsive-response matrices using the new Padé approximation.

Equations (5-37b) and (5-37d) can be written as:

$$\begin{aligned} [\bar{\mathbf{H}}(t)] &= [\mathbf{M}][\mathbf{H}(t)] \\ &= [\mathbf{M}] \cdot ([\mathbf{Q}_{11}][\mathbf{Q}_{22}] - [\mathbf{Q}_{12}][\mathbf{Q}_{21}])^{-1} \cdot ([\mathbf{Q}_{22}][\mathbf{N}_{12}] - [\mathbf{Q}_{12}][\mathbf{N}_{22}]) \end{aligned} \quad (5-39a)$$

$$\begin{aligned} [\bar{\mathbf{J}}(t)] &= [\mathbf{M}][\mathbf{J}(t)] \\ &= [\mathbf{M}] \cdot ([\mathbf{Q}_{11}][\mathbf{Q}_{22}] - [\mathbf{Q}_{12}][\mathbf{Q}_{21}])^{-1} \cdot ([\mathbf{Q}_{11}][\mathbf{N}_{22}] - [\mathbf{Q}_{21}][\mathbf{N}_{12}]) \end{aligned} \quad (5-39b)$$

With Rayleigh damping, the structure of the denominator and numerator of response matrix $[\mathbf{H}(t)]$ by Padé series approximation is similar to the structure of $[\mathbf{H}(t)]$ by Taylor series approximation. From Table 5-3, it can be shown that while the denominator and numerator of $[\mathbf{H}(t)]$ and $[\mathbf{J}(t)]$ may not be symmetric, $[\mathbf{M}][\mathbf{H}(t)]$ and $[\mathbf{M}][\mathbf{J}(t)]$ are symmetric. Since $[\bar{\mathbf{H}}(t)]$ and $[\bar{\mathbf{J}}(t)]$ are symmetric, only the diagonal and half of the remaining entries (say the upper triangular part) need to be computed (Fung, 1997). Hence the symmetric property can also be used to reduce the computational cost in this chapter. From Table 5-3, it notes that the numerator of $[\mathbf{H}(t)]$ in Equations (5-39) only includes odd items. More computational cost can be reduced.

If $[\mathbf{M}]$ is not diagonal, it can be decomposed by Cholesky factorization (1982) as

$$[\mathbf{M}] = [\mathbf{L}][\mathbf{L}]^T \quad \text{and} \quad [\mathbf{M}]^{-1} = [\mathbf{L}]^{-T}[\mathbf{L}]^{-1} \quad (5-40)$$

where $[\mathbf{L}]$ is a lower triangular matrix. Let

$$\begin{aligned} [\bar{\bar{\mathbf{H}}}(t)] &= [\mathbf{L}]^{-1}[\bar{\mathbf{H}}(t)][\mathbf{L}]^{-T} = [\mathbf{L}]^T[\mathbf{H}(t)][\mathbf{L}]^{-T} \\ &= [\mathbf{L}]^T ([\mathbf{Q}_{11}][\mathbf{Q}_{22}] - [\mathbf{Q}_{12}][\mathbf{Q}_{21}])^{-1} \cdot ([\mathbf{Q}_{22}][\mathbf{N}_{12}] - [\mathbf{Q}_{12}][\mathbf{N}_{22}])[\mathbf{L}]^{-T} \end{aligned} \quad (5-41a)$$

$$\begin{aligned} [\bar{\bar{\mathbf{J}}}(t)] &= [\mathbf{L}]^{-1}[\bar{\mathbf{J}}(t)][\mathbf{L}]^{-T} = [\mathbf{L}]^T[\mathbf{J}(t)][\mathbf{L}]^{-T} \\ &= [\mathbf{L}]^T ([\mathbf{Q}_{11}][\mathbf{Q}_{22}] - [\mathbf{Q}_{12}][\mathbf{Q}_{21}])^{-1} \cdot ([\mathbf{Q}_{11}][\mathbf{N}_{22}] - [\mathbf{Q}_{21}][\mathbf{N}_{12}])[\mathbf{L}]^{-T} \end{aligned} \quad (5-41b)$$

denote two symmetric matrices.

5.5 Computational Effort

In this chapter, the original PTI method (**A1**) and the PTI method by step-response and impulsive-response matrices (**A4**) are considered. The Padé approximations are employed to improve the efficiency and the stability of the PTI methods.

In general, the Taylor series approximation is more popular than Padé series approximation for exponential matrix approximation in the PTI methods. The computational cost by high order Padé series approximation will become high for evaluating the inverse matrix though the accuracy can be improved. With scaling and squaring techniques, the lower order series approximations can be used to obtain the accurate results while the efficiency is high. For example, the computational costs will be almost the same at same error level using 4-term Taylor series approximation and 2-term diagonal Padé series approximation. This property will be investigated in the following numerical example. The results derived in the example in this paper are coincident with the analysis by Moler and Loan (Moler and Loan, 1978).

Later, the more efficient PTI method by step-response and impulsive-response matrices (**A4**) is considered. The PTI method by step-response and impulsive-response matrices can reduce about half efforts than original PTI method because the symmetric property can be used to compute the response matrices by Taylor series approximation. If the symmetric property can not be used in the PTI method with step-response and impulsive-response matrices, the method (**A4**) will lose the efficiency compared with the method (**A1**). The PTI method using new Padé approximation (**A8**) could keep the symmetric properties comprehensively.

With Rayleigh damping, the problems in the multi-degree-of-freedom system can be tackled using the same method derived in the single-degree-of-freedom system. In the current new Padé approximation, the denominators of the matrices [**H**] and [**J**] are the same. Only one inverse matrix need be evaluated. Furthermore, only odd

items in the numerator of $[\mathbf{H}]$ need be computed according to Table 5-3. Hence, the computational effort is comparable with the Taylor series approximation.

In general, the efficiency of the PTI method using the proposed new Padé approximation (A8) is come form following three aspects:

- (1) The symmetric property can be used.
- (2) The nominator of the current Padé approximation for impulsive-response matrix $[\mathbf{H}]$ just keeps the odd items.
- (3) Only one inverse matrix need be done for both $[\mathbf{H}]$ and $[\mathbf{J}]$.

5.6 Algorithm Stability

The paper (Fung, 1997) gave a comprehensive analysis of the PTI method by step-response and impulse-response matrices using Taylor series approximation. The stability characteristic of the PTI algorithms with Padé series approximation can be investigated by considering the following equivalent transformation.

$$\begin{Bmatrix} \mathbf{u}(\Delta t) \\ \mathbf{v}(\Delta t) \end{Bmatrix} = \begin{bmatrix} \mathbf{G}(\Delta t) & \mathbf{H}(\Delta t) \\ \dot{\mathbf{G}}(\Delta t) & \dot{\mathbf{H}}(\Delta t) \end{bmatrix} \cdot \begin{Bmatrix} \mathbf{u}_0 \\ \mathbf{v}_0 \end{Bmatrix} = \begin{bmatrix} \mathbf{G}(\Delta t/2^N) & \mathbf{H}(\Delta t/2^N) \\ \dot{\mathbf{G}}(\Delta t/2^N) & \dot{\mathbf{H}}(\Delta t/2^N) \end{bmatrix}^{2^N} \cdot \begin{Bmatrix} \mathbf{u}_0 \\ \mathbf{v}_0 \end{Bmatrix} \quad (5-42)$$

The stability of the final matrix of time-step size Δt depends on the stability of the initial matrix of time-step size $\Delta t/2^N$ obtained from the Taylor series solutions or new Padé series solutions. Later, this section will discuss the difference of approximation to $[\mathbf{G}]$ and $[\mathbf{H}]$ functions using the Taylor series approximation, the ordinary Padé series approximation and the new Padé approximation respectively.

To compare the stability characteristic of the Taylor series solutions, the ordinary Padé series solutions and current new Padé solutions, the un-damped free vibration with $[\mathbf{C}]=[\mathbf{0}]$ and $\{\mathbf{r}(t)\}=\{\mathbf{0}\}$ is considered. To further simplify the stability analysis, the modal decomposition method is used to uncouple the system of differential

equations. As a result, consider the stability of a single-degree-of-freedom system governed by

$$\ddot{u} + \omega^2 \cdot u = 0 \tag{5-43}$$

The displacement and velocity at $t = \Delta t$ can be written as

$$\begin{Bmatrix} u(\Delta\tau) \\ v(\Delta\tau) \end{Bmatrix} = \begin{bmatrix} g(\Delta\tau) & h(\Delta\tau) \\ -\omega^2 h(\Delta\tau) & g(\Delta\tau) \end{bmatrix} \begin{Bmatrix} u_0 \\ v_0 \end{Bmatrix} \tag{5-44}$$

Where $\tau = \omega \cdot t, \Delta\tau = \omega \cdot \Delta t$

The squares of the moduli of the eigenvalues $|\lambda|^2$ are given by

$$|\lambda|^2 = g^2(\Delta\tau) + \omega^2 h^2(\Delta\tau) \tag{5-45}$$

The algorithm is unstable if $|\lambda|^2 > 1$.

The response functions $g(\Delta\tau)$ and $h(\Delta\tau)$ by ordinary Padé series approximation can be obtained from Table 5-2. The coefficients ($\alpha_{ij}, \beta_{ij}, \alpha'_{ij}$ and β'_{ij}) can be evaluated from Equations (5-17a-b) and (5-18a-b). The stability properties of first four-order ordinary Padé series approximation are given below.

Using the formulae of the current new Padé approximation described in Table 5-3, there has

$$|\lambda|^2 = g^2(\Delta\tau) + \omega^2 h^2(\Delta\tau) \equiv 1 \tag{5-46}$$

For example, consider the first-order Padé series approximation,

$$\begin{aligned} |\lambda|^2 &= g^2(\Delta\tau) + \omega^2 h^2(\Delta\tau) \\ &= \frac{(1 - \frac{1}{4}\tau^2)^2}{(1 + \frac{1}{4}\tau^2)^2} + \frac{\tau^2}{(1 + \frac{1}{4}\tau^2)^2} = \frac{(1 - \frac{1}{2}\tau^2 + \frac{1}{16}\tau^4) + \tau^2}{(1 + \frac{1}{2}\tau^2 + \frac{1}{16}\tau^4)} = 1. \end{aligned} \tag{5-47}$$

More details can be found in Example 5.1 in this chapter.

The stability of the PTI algorithm using Taylor series approximation, ordinary Padé series approximation and the proposed new Padé approximation are given below.

n	Taylor approximation	p	Ordinary Padé Series Approximation	p	New Padé Approximation
2	Unconditionally unstable	1	Stable if $\tau < 1.752$	1	Unconditionally Stable
4	Stable if $\tau < 2.828$	2	Stable if $\tau < 0.385$ or $1.680 < \tau < 4.821$	2	Unconditionally Stable
6	Unconditionally unstable	3	Stable if $\tau < 1.647$ or $4.991 < \tau < 7.448$	3	Unconditionally Stable
8	Stable if $\tau < 3.395$	4	Stable if $\tau < 4.900$ or $8.505 < \tau < 9.467$	4	Unconditionally Stable

It notes that the proposed first four-order Padé approximation is unconditionally stable.

5.7 Numerical Examples

Example 5.1: Single-Degree-of-Freedom System

To compare the convergence and stability of Taylor series approximation and the proposed new Padé series approximation, a single-degree-of-freedom system is considered.

$$\ddot{u}(t) + 2\xi\omega \cdot \dot{u}(t) + \omega^2 \cdot u(t) = 0 \tag{5-48}$$

where ξ and ω are the damping ratio and undamped natural frequency of the system respectively. The analytical solution is given by

$$u(t) = u_0 \cdot g(t) + v_0 \cdot h(t) \tag{5-49}$$

where the step-response and impulsive-response functions $g(t)$ and $h(t)$ have different forms depending on whether the system is under-critically damped ($\xi < 1$),

critically damped ($\xi = 1$) and over-critically damped ($\xi > 1$). For under-critically damped system, $g(t)$ and $h(t)$ are given by

$$g(t) = \exp(-\xi\omega t) \left(\cos(\omega_d t) + \frac{\xi\omega}{\omega_d} \sin(\omega_d t) \right) \quad (5-50a)$$

$$h(t) = \exp(-\xi\omega t) \left(\frac{1}{\omega_d} \sin(\omega_d t) \right) \text{ and } \omega_d = \sqrt{1 - \xi^2} \cdot \omega \quad (5-50b)$$

For critically damped system, $g(t)$ and $h(t)$ are given by

$$g(t) = (1 + \omega t) \exp(-\omega t) \text{ and } h(t) = t \cdot \exp(-\omega t). \quad (5-51)$$

For over-critically damped system, $g(t)$ and $h(t)$ are given by

$$g(t) = (\omega_{d1} \exp(\omega_{d2} t) - \omega_{d2} \exp(\omega_{d1} t)) / (\omega_{d1} - \omega_{d2}) \quad (5-52a)$$

$$h(t) = (\exp(\omega_{d2} t) - \exp(\omega_{d1} t)) / (\omega_{d2} - \omega_{d1}) \quad (5-52b)$$

$$\text{where } \omega_{d1}, \omega_{d2} = (-\xi \pm \sqrt{\xi^2 - 1})\omega$$

More details of the Taylor series expansions for $g(t)$ and $h(t)$ in different cases can be found in the paper (Fung, 1997). In the present formulation, there is also no need to differentiate three cases. The new Padé series expansions for $g(t)$ and $h(t)$ are all given by

$$g(t) = (Q_{11}Q_{22} - Q_{12}Q_{21})^{-1} \cdot (Q_{22}N_{11} - Q_{12}N_{21}) \quad (5-53a)$$

$$h(t) = (Q_{11}Q_{22} - Q_{12}Q_{21})^{-1} \cdot (Q_{22}N_{12} - Q_{12}N_{22}) \quad (5-53b)$$

where the special formula can be found in Table 5-3 (a-d).

Let $Tg_n^0(t)$ and $Th_n^0(t)$ denote the n th order Taylor series approximations of $g(t)$ and $h(t)$. $Pg_n^0(t)$ and $Ph_n^0(t)$ denote the n th order new Padé approximations of $g(t)$ and $h(t)$, respectively. The N th level recursive Padé approximations of $g(t)$ and $h(t)$ are denoted by $Pg_n^N(t)$ and $Ph_n^N(t)$ ($Tg_n^N(t)$ and $Th_n^N(t)$ in Taylor series approximation), respectively. The corresponding time derivatives are evaluated from

$$P\dot{g}_n^N(t) = -\omega^2 Ph_n^N(t) \text{ and } Ph_n^N(t) = Pg_n^N(t) - 2\xi\omega Ph_n^N(t) \quad (5-54)$$

The recurrence formulae between $Pg_n^N(t)$, $Ph_n^N(t)$, $Pg_n^{N-1}(t/2)$ and $Ph_n^{N-1}(t/2)$ are

$$\begin{bmatrix} Pg_n^N(t) & Ph_n^N(t) \\ P\dot{g}_n^N(t) & Ph_n^N(t) \end{bmatrix} = \begin{bmatrix} Pg_n^{N-1}(t/2) & Ph_n^{N-1}(t/2) \\ P\dot{g}_n^{N-1}(t/2) & Ph_n^{N-1}(t/2) \end{bmatrix}^2 \quad (5-55)$$

As a result, $Pg_n^N(t)$ and $Ph_n^N(t)$ are just different approximations for $g(t)$ and $h(t)$ similar to Taylor series approximation, respectively.

If ξ is zero, the step-response and impulsive-response functions $g(t)$ and $h(t)$ are given by

$$g(t) = \cos(\omega t) \quad (5-56a)$$

$$h(t) = \frac{1}{\omega} \sin(\omega t) \quad (5-56b)$$

For example, the $Pg_2^0(t)$ and $Ph_2^0(t)$ with $\xi = 0$ can be obtained from Table 5-3 as

$$Pg_2^0(t) = \left(1 - \frac{1}{12}at^2 + \frac{1}{144}a^2t^4\right)^{-1} \cdot \left(1 + \frac{5}{12}at^2 + \frac{1}{144}a^2t^4\right) \quad (5-57a)$$

$$Ph_2^0(t) = \left(1 - \frac{1}{12}at^2 + \frac{1}{144}a^2t^4\right)^{-1} \cdot \left(t + \frac{1}{12}at^3\right) \quad (5-57b)$$

Then, the explicit forms of $Pg_2^1(t)$, $Ph_2^1(t)$, $Pg_2^2(t)$ and $Ph_2^2(t)$ with $\xi = 0$ are

$$Pg_2^1(t) = [Pg_2^0(t/2)]^2 - [Ph_2^0(t/2)]^2 \quad (5-58a)$$

$$= \left(1 - \frac{1}{12}a\left(\frac{t}{2}\right)^2 + \frac{1}{144}a^2\left(\frac{t}{2}\right)^4\right)^{-2}$$

$$\left\{ \left(1 + \frac{5}{12}a\left(\frac{t}{2}\right)^2 + \frac{1}{144}a^2\left(\frac{t}{2}\right)^4\right)^2 - \left(\left(\frac{t}{2}\right) + \frac{1}{12}a\left(\frac{t}{2}\right)^3\right)^2 \right\}$$

$$Ph_2^1(t) = 2 \cdot Pg_2^0(t/2) \cdot Ph_2^0(t/2) \quad (5-58b)$$

$$= \left(1 - \frac{1}{12} a \left(\frac{t}{2} \right)^2 + \frac{1}{144} a^2 \left(\frac{t}{2} \right)^4 \right)^{-2} \cdot \left\{ 2 \left(1 + \frac{5}{12} a \left(\frac{t}{2} \right)^2 + \frac{1}{144} a^2 \left(\frac{t}{2} \right)^4 \right) \cdot \left(\left(\frac{t}{2} \right) + \frac{1}{12} a \left(\frac{t}{2} \right)^3 \right) \right\}$$

$$Pg_2^2(t) = \left((Pg_2^0(t/4))^2 - (Ph_2^0(t/4))^2 \right)^2 - (2 \cdot Pg_2^0(t/4) \cdot Ph_2^0(t/4))^2 \quad (5-59a)$$

$$Ph_2^2(t) = 4 \cdot \left((Pg_2^0(t/4))^2 - (Ph_2^0(t/4))^2 \right) \cdot Pg_2^0(t/4) \cdot Ph_2^0(t/4) \quad (5-59b)$$

Various numerical approximations by Taylor approximation (4-term) and the new Padé approximation (2-term) and the exact analytical solution for the under-critically damped critically damped and over-critically damped systems are shown in Figure 5-1, Figure 5-2 and Figure 5-3. The numerical solutions obtained by new Padé approximation do not diverge even the number of recursive evaluation is very low. The results obtained by Taylor series approximation will go to infinite with low N . It can be seen that the new Padé approximation converges faster than Taylor series approximation. It will be unconditionally stable using the new Padé approximation.

Example 5.2: Padé Series Approximation in PTI Methods

(1) *Mass-Spring System*

Consider an 8-DOF system as shown in Figure 5-4. The governing equation is

$$[\mathbf{M}] \cdot \{\ddot{\mathbf{u}}\} + [\mathbf{C}] \cdot \{\dot{\mathbf{u}}\} + [\mathbf{K}] \cdot \{\mathbf{u}\} = \{\mathbf{r}(t)\} \quad (5-60)$$

The matrices are

(10^{-5}) will be higher than that of Taylor series approximation (10^{-4}). The improvement of the efficiency of the PTI method using the Padé series approximation (A7) will be slight at the same error level. Furthermore, the PTI method (A7) is unconditionally stable.

Table 5-6 gives the comparisons of different p in Padé series approximation at different time step size. According to the analysis in Equation (5-12) and Table 5-6, the odd p (2 or 4) will be much better than even p (1 or 3) in ordinary Padé series approximation. In general, p can be chosen as 2 or 4 while the scaling and squaring techniques is used. The bold data in Table 5-6 are the least computational cost at same error 10^{-7} and 10^{-9} for different time step size. $p=2$ is recommended to be used in this example.

(3) *New Padé Approximation to Response Matrices with Rayleigh Damping*

The governing equation of this example is the same as Equation (5-63). The matrices $[\mathbf{M}]$, $[\mathbf{K}]$ and the initial displacement and velocity $N_s \times 1$ vectors $\{\mathbf{u}_0\}$ and $\{\dot{\mathbf{u}}_0\}$ are the same as Equation (5-66). The damping matrix $[\mathbf{C}]$ is described by Rayleigh damping as

$$[\mathbf{C}] = \alpha[\mathbf{M}] + \beta[\mathbf{K}], \text{ where } \alpha = 0.01 \text{ and } \beta = 0.0555 \quad (5-67)$$

$\{\mathbf{r}_0\}$ and $\{\mathbf{r}_1\}$ are $N_s \times 1$ vector composed of the repeated identical triplets of elements shown

$$\{\mathbf{r}_0\} = \{0.01, 0.02, 0, 0.01, 0.02, 0, \dots\} \quad (5-68a)$$

$$\{\mathbf{r}_1\} = \{0.001, 0.002, 0, 0.001, 0.002, 0, \dots\} \quad (5-68b)$$

The computing formulae of the current new (2, 2) Padé approximation can be obtained from Table 5-3. The computational cost of the PTI method using Taylor series approximation (A4) and new Padé approximation are given (A8) in Table 5-

7. The symmetric properties are used in Taylor series approximation and new Padé approximation. As discussed in front numerical example, the computational cost of the 4th-order Taylor series approximation and new (2, 2) Padé approximation are almost the same if there are no special algorithms for Padé approximation. But with Rayleigh damping, the PTI method using current new Padé approximation (**A8**) will be more efficient than the PTI method (**A4**). The symmetric property can be used to reduce the computational cost of the PTI algorithm using the new Padé approximation. More computational cost can be saved because there only one inverse matrix is evaluated for both **[H]** and **[J]**.

5.4 Summary

This chapter discussed the implementation of Padé approximations in the original PTI method and the PTI method by step-response and impulsive-response matrices. The stability of the Taylor and Padé series approximation are also discussed. The PTI method (**A7**) is more efficient and stable than the original PTI method (**A1**). Later, the new Padé approximation is generated and employed in PTI method by step-response and impulsive-response matrices to improve the efficiency and stability. The PTI method using the new Padé approximation (**A8**) is unconditionally stable for both single and multi-degree-of-freedom system with Rayleigh damping. Furthermore, the step-response and impulsive-response matrices in the PTI method (**A8**) can also be expressed in terms of two symmetric matrices. Hence more computational cost can be saved. Two numerical examples are used to illustrate the efficient and stable algorithms in this chapter.

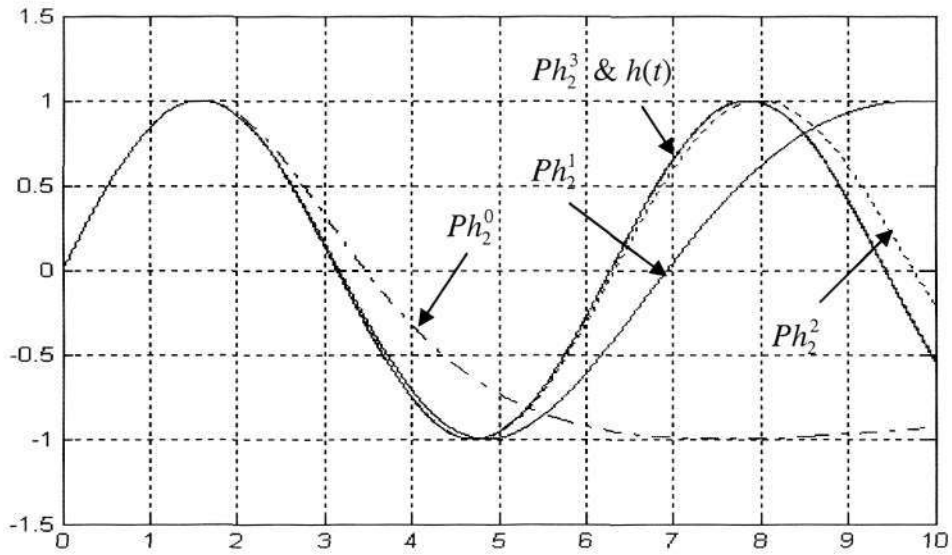


Figure 5-1 (a) $h(t)$ and its Padé approximations ($\xi = 0, \omega = 1$)

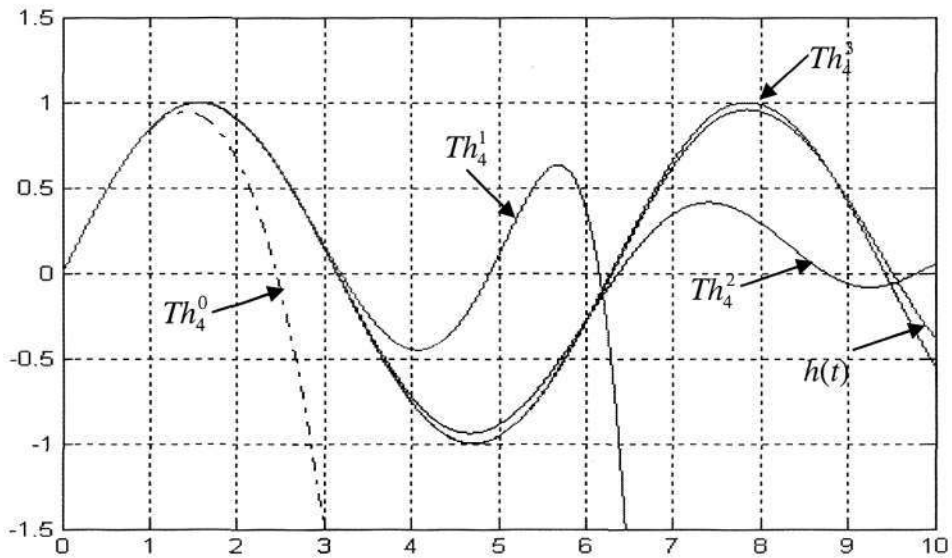


Figure 5-1 (b) $h(t)$ and its Taylor series approximations ($\xi = 0, \omega = 1$)

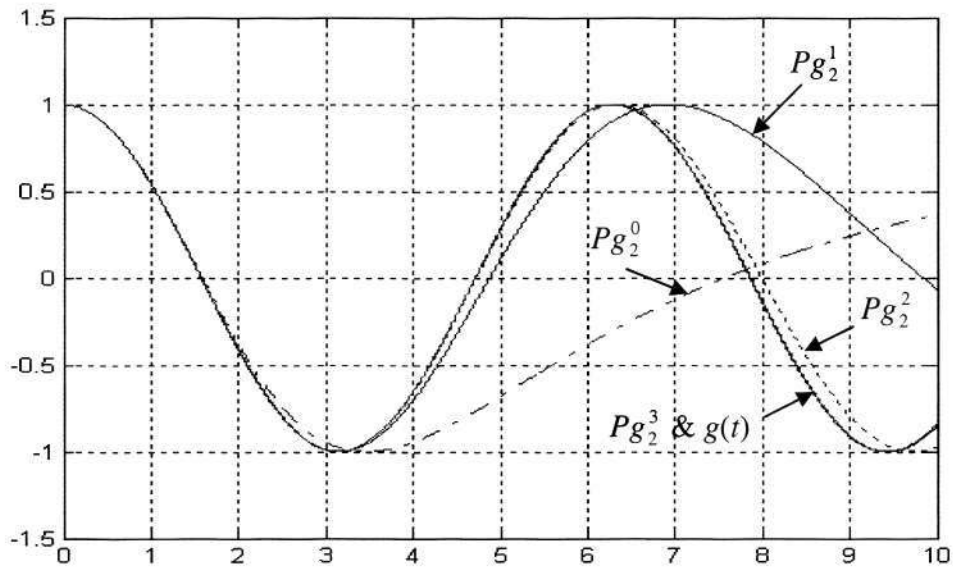


Figure 5-1 (c) $g(t)$ and its Padé approximations ($\xi = 0, \omega = 1$)

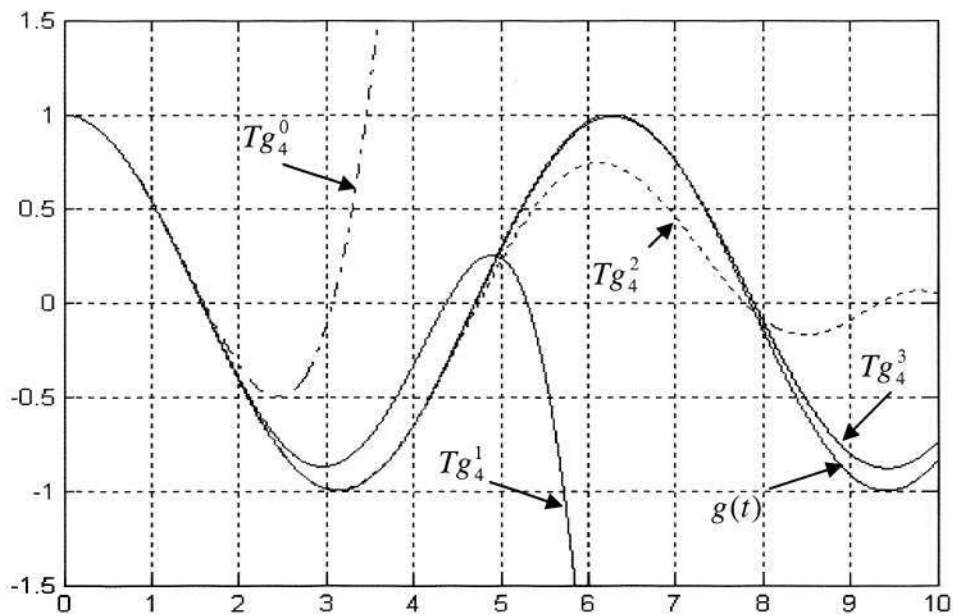


Figure 5-1 (d) $g(t)$ and its Taylor series approximations ($\xi = 0, \omega = 1$)

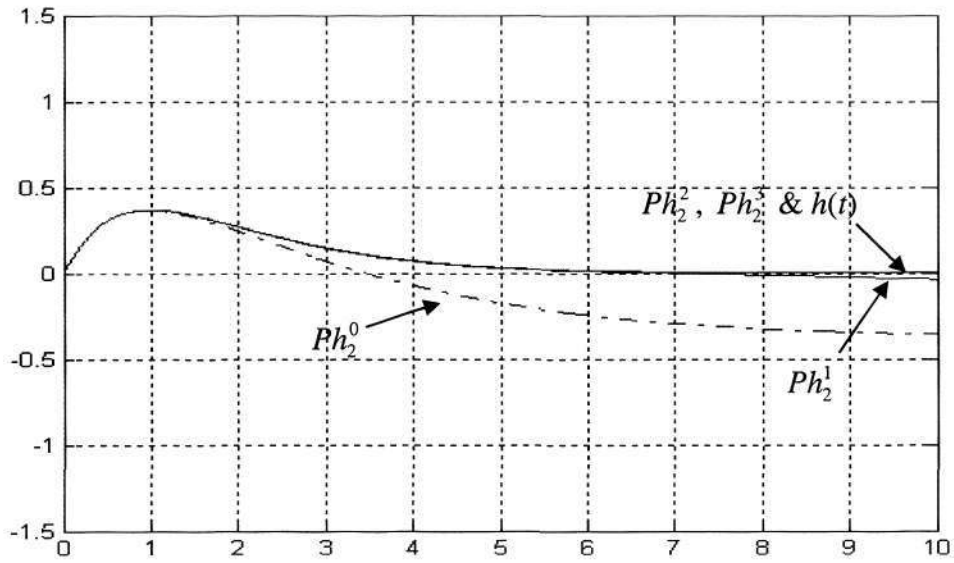


Figure 5-2 (a) $h(t)$ and its Padé approximations ($\xi = 1, \omega = 1$)

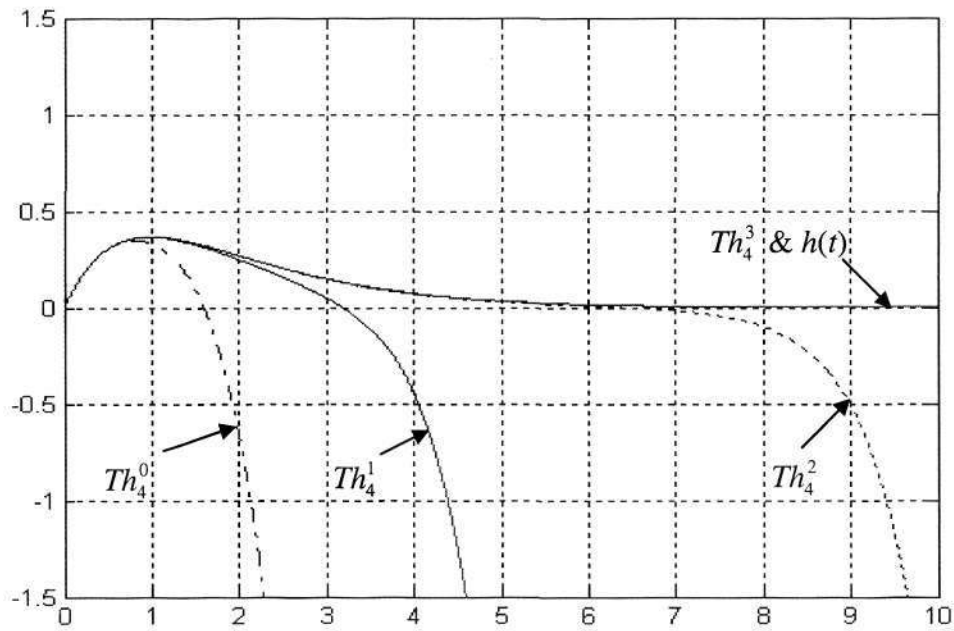


Figure 5-2 (b) $h(t)$ and its Taylor series approximations ($\xi = 1, \omega = 1$)

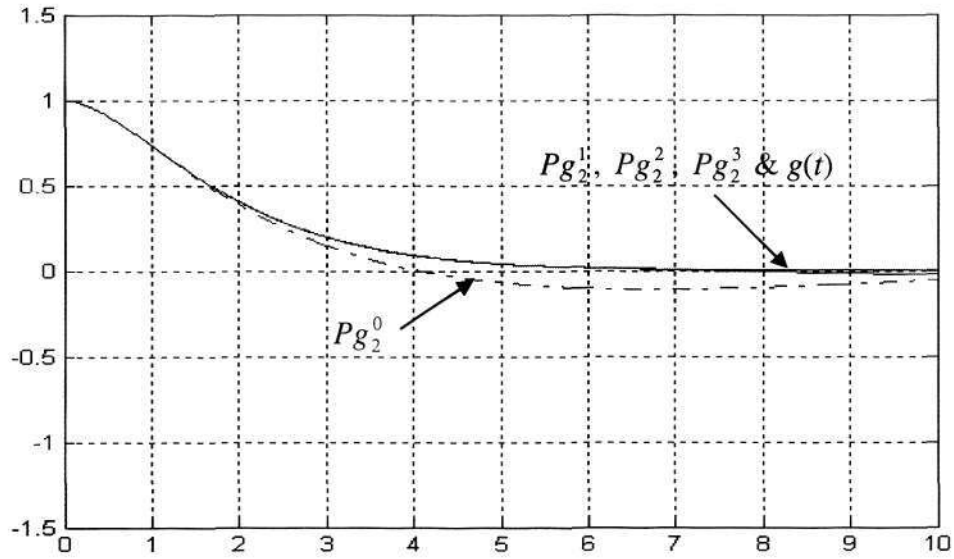


Figure 5-2 (c) $g(t)$ and its Padé approximations ($\xi=1, \omega=1$)

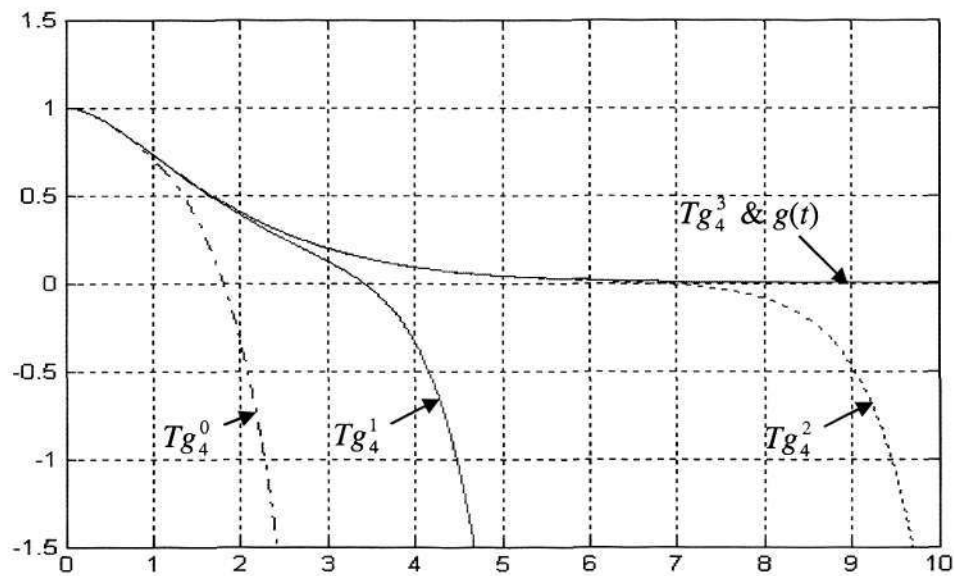


Figure 5-2 (d) $g(t)$ and its Taylor series approximations ($\xi=1, \omega=1$)

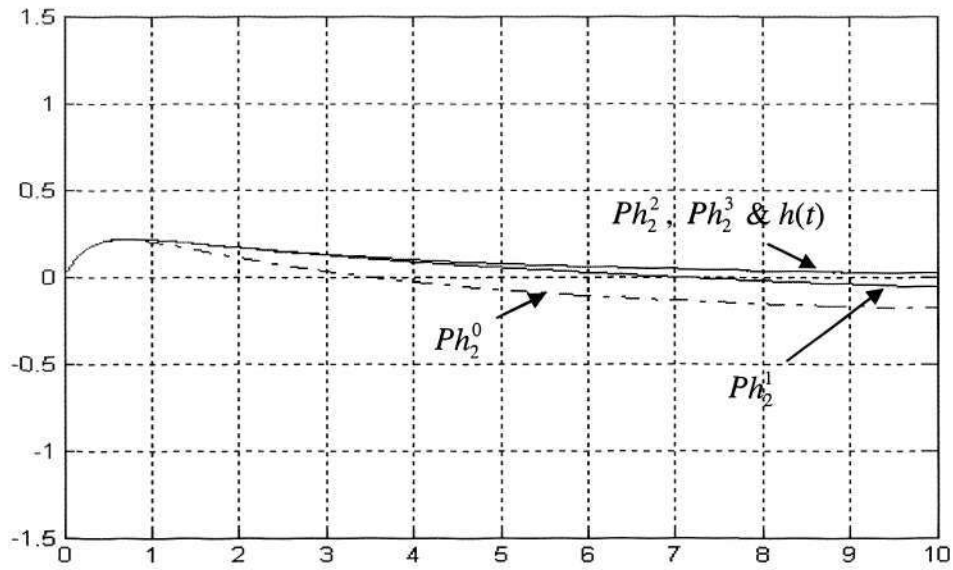


Figure 5-3 (a) $h(t)$ and its Padé approximations ($\xi = 2, \omega = 1$)

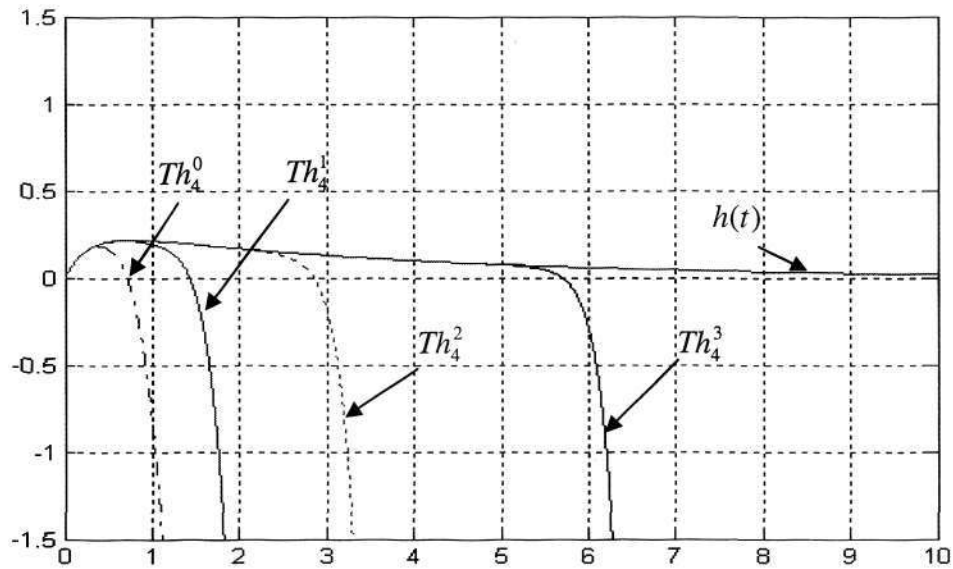


Figure 5-3 (b) $h(t)$ and its Taylor series approximations ($\xi = 2, \omega = 1$)

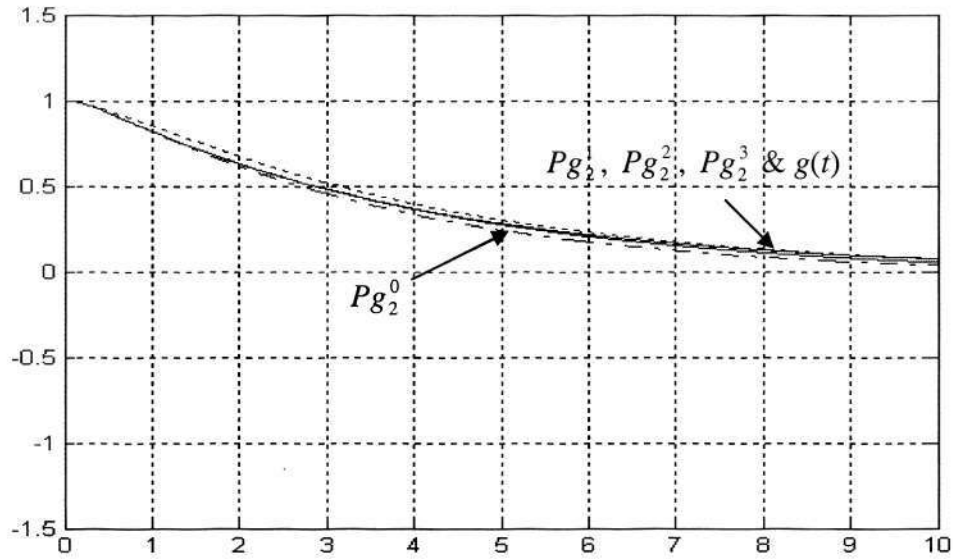


Figure 5-3 (c) $g(t)$ and its Padé approximations ($\xi = 2, \omega = 1$)

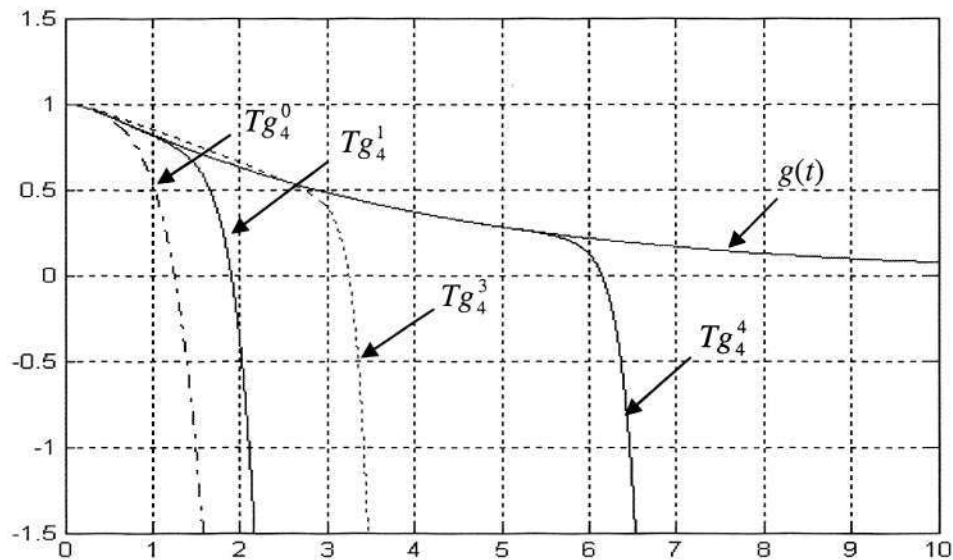


Figure 5-3 (d) $g(t)$ and its Taylor series approximations ($\xi = 2, \omega = 1$)

Improved Precise Time-Step Integration Algorithms for Dynamic Problems

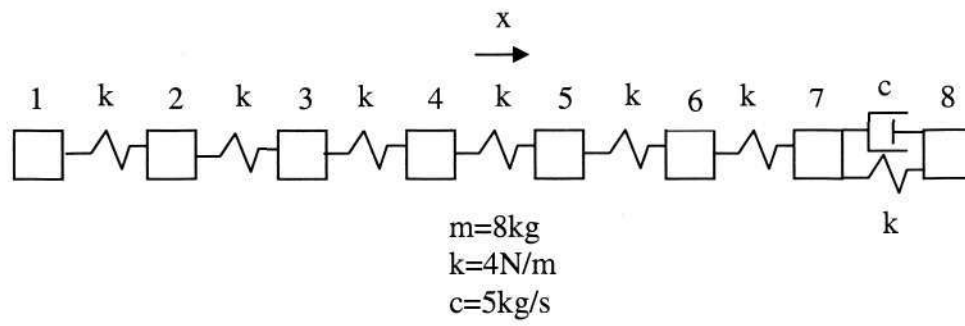


Figure 5-4 A series of eight mass-springs with a damper at the right-hand connection

Table 5-1 $[\mathbf{T}_a^*]$ for various values of p

order (p)	$[\mathbf{T}_a^*] = \mathbf{P}_p([\mathbf{W}]) - [\mathbf{I}]$
1	$([\mathbf{I}] - \frac{1}{2}[\mathbf{W}])^{-1} \cdot [\mathbf{W}]$
2	$([\mathbf{I}] - \frac{1}{2}[\mathbf{W}] + \frac{1}{12}[\mathbf{W}]^2)^{-1} \cdot [\mathbf{W}]$
3	$([\mathbf{I}] - \frac{1}{2}[\mathbf{W}] + \frac{1}{12}[\mathbf{W}]^2 - \frac{1}{120}[\mathbf{W}]^3)^{-1} \cdot ([\mathbf{H}] + \frac{1}{60}[\mathbf{H}]^3)$
4	$([\mathbf{I}] - \frac{1}{2}[\mathbf{W}] + \frac{3}{28}[\mathbf{W}]^2 - \frac{1}{84}[\mathbf{W}]^3 + \frac{1}{1680}[\mathbf{W}]^4)^{-1} \cdot ([\mathbf{W}] + \frac{1}{42}[\mathbf{W}]^3)$
5	$([\mathbf{I}] - \frac{1}{2}[\mathbf{W}] + \frac{1}{9}[\mathbf{W}]^2 - \frac{1}{72}[\mathbf{W}]^3 + \frac{1}{1008}[\mathbf{W}]^4 - \frac{1}{30240}[\mathbf{W}]^5)^{-1} \cdot ([\mathbf{W}] + \frac{1}{36}[\mathbf{W}]^3 + \frac{1}{15120}[\mathbf{W}]^5)$

Table 5-2 Formulae of q_i and d_i

i	q_i	d_i
0	1	1
1	$\alpha_{11} \cdot b$	$\beta_{11} \cdot b$
2	$\alpha_{21} \cdot a + \alpha_{22} \cdot b^2$	$\beta_{21} \cdot a + \beta_{22} \cdot b^2$
3	$\alpha_{31} \cdot ab + \alpha_{32} \cdot b^3$	$\beta_{31} \cdot ab + \beta_{32} \cdot b^3$
4	$\alpha_{41} \cdot a^2 + \alpha_{42} \cdot ab^2 + \alpha_{43} \cdot b^4$	$\beta_{41} \cdot a^2 + \beta_{42} \cdot ab^2 + \beta_{43} \cdot b^4$

Table 5-3 (a) Formulae of first four-order new Padé approximation to $g(t)$

order (p)	Formula of $g(t)$
1	$(1 - \frac{1}{2}bt - \frac{1}{4}at^2)^{-1}(1 - \frac{1}{2}bt + \frac{1}{4}at^2)$
2	$\left(1 - \frac{1}{2}bt + \frac{1}{12}(b^2 - a)t^2 + \frac{1}{24}abt^3 + \frac{1}{144}a^2t^4\right)^{-1} \cdot \left(1 - \frac{1}{2}bt + \frac{1}{12}(b^2 + 5a)t^2 - \frac{1}{24}abt^3 + \frac{1}{144}a^2t^4\right)$
3	$\left(1 - \frac{1}{2}bt + \frac{1}{12}(b^2 - a)t^2 + \frac{1}{120}(2ab - b^3)t^3 - \frac{1}{720}(a^2 + 3ab^2)t^4 - \frac{1}{1440}a^2bt^5 - \frac{1}{14400}a^3t^6\right)^{-1} \cdot \left(1 - \frac{1}{2}bt + \frac{1}{12}(b^2 + 5a)t^2 - \frac{1}{120}(6ab + b^3)t^3 + \frac{1}{720}(11a^2 + 3b^2)t^4 - \frac{1}{1440}a^2bt^5 + \frac{1}{14400}a^3t^6\right)$
4	$\left(1 - \frac{1}{2}bt + \frac{1}{28}(3b^2 - a)t^2 + \frac{1}{168}(3ab - 2b^3)t^3 + \frac{1}{1680}\left(\frac{9}{7}a^2 - 3ab^2 + b^4\right)t^4 + \frac{1}{23520}(111a^2b + 67ab^3)t^5 + \frac{1}{141120}(-2a^3 + 9a^2b^2)t^6 + \frac{1}{141120}a^3bt^7 + \frac{1}{1680} \frac{1}{1680}a^4t^8\right)^{-1} \cdot \left(1 - \frac{1}{2}bt + \frac{1}{28}(13a + 3b^2)t^2 - \frac{1}{168}(9ab + 2b^3)t^3 + \frac{1}{11760}(289a^2 + 98ab^2 + 7b^4)t^4 - \frac{1}{23520}(37a^2b + 7ab^3)t^5 + \frac{1}{141120}(29a^3 + 9a^2b^2)t^6 - \frac{1}{141120}a^3bt^7 + \frac{1}{1680} \frac{1}{1680}a^4t^8\right)$

Table 5-3 (b) Formulae of first four-order new Padé approximation to $h(t)$

order (p)	Formula of $h(t)$
1	$(1 - \frac{1}{2}bt - \frac{1}{4}at^2)^{-1} \cdot t$
2	$\left(1 - \frac{1}{2}bt + \frac{1}{12}(b^2 - a)t^2 + \frac{1}{24}abt^3 + \frac{1}{144}a^2t^4\right)^{-1} \cdot \left(t + \frac{1}{12}at^3\right)$
3	$\left(1 - \frac{1}{2}bt + \frac{1}{12}(b^2 - a)t^2 + \frac{1}{120}(2ab - b^3)t^3 - \frac{1}{720}(a^2 + 3ab^2)t^4 - \frac{1}{1440}a^2bt^5 - \frac{1}{14400}a^3t^6\right)^{-1} \cdot \left(t + \frac{1}{60}(b^2 + 6a)t^3 + \frac{1}{720}a^2t^5\right)$
4	$\left(1 - \frac{1}{2}bt + \frac{1}{28}(3b^2 - a)t^2 + \frac{1}{168}(3ab - 2b^3)t^3 + \frac{1}{1680}\left(\frac{9}{7}a^2 - 3ab^2 + b^4\right)t^4 + \frac{1}{23520}(111a^2b + 67ab^3)t^5 + \frac{1}{141120}(-2a^3 + 9a^2b^2)t^6 + \frac{1}{141120}a^3bt^7 + \frac{1}{1680} \frac{1}{1680}a^4t^8\right)^{-1} \cdot \left(t + \frac{1}{84}(11a + 2b^3)t^3 + \frac{1}{11760}(37a^2 + 7ab^3)t^5 + \frac{1}{70560}a^3t^7\right)$

Table 5-3 (c) Formulae of first four-order new Padé approximation to $\dot{g}(t)$

order (p)	Formula of $\dot{g}(t)$
1	$(1 - \frac{1}{2}bt - \frac{1}{4}at^2)^{-1} \cdot at$
2	$\left(1 - \frac{1}{2}bt + \frac{1}{12}(b^2 - a)t^2 + \frac{1}{24}abt^3 + \frac{1}{144}a^2t^4\right)^{-1} \cdot \left(at + \frac{1}{12}a^2t^3\right)$
3	$\left(1 - \frac{1}{2}bt + \frac{1}{12}(b^2 - a)t^2 + \frac{1}{120}(2ab - b^3)t^3 - \frac{1}{720}(a^2 + 3ab^2)t^4 - \frac{1}{1440}a^2bt^5 - \frac{1}{14400}a^3t^6\right)^{-1} \cdot \left(at + \frac{1}{60}(ab^2 + 6a^2)t^3 + \frac{1}{720}a^3t^5\right)$
4	$\left(1 - \frac{1}{2}bt + \frac{1}{28}(3b^2 - a)t^2 + \frac{1}{168}(3ab - 2b^3)t^3 + \frac{1}{1680}\left(\frac{9}{7}a^2 - 3ab^2 + b^4\right)t^4 + \frac{1}{23520}(111a^2b + 67ab^3)t^5 + \frac{1}{141120}(-2a^3 + 9a^2b^2)t^6 + \frac{1}{141120}a^3bt^7 + \frac{1}{1680} \frac{1}{1680}a^4t^8\right)^{-1} \cdot \left(at + \frac{1}{84}(11a^2 + 2ab^3)t^3 + \frac{1}{11760}(37a^3 + 7a^2b^3)t^5 + \frac{1}{70560}a^4t^7\right)$

Table 5-3 (d) Formulae of first four-order new Padé approximation to $\dot{h}(t)$

order (p)	Formula of $\dot{h}(t)$
1	$(1 - \frac{1}{2}bt - \frac{1}{4}at^2)^{-1}(1 + \frac{1}{2}bt + \frac{1}{4}at^2)$
2	$\left(1 - \frac{1}{2}bt + \frac{1}{12}(b^2 - a)t^2 + \frac{1}{24}abt^3 + \frac{1}{144}a^2t^4\right)^{-1} \cdot$ $\left(1 + \frac{1}{2}bt + \frac{1}{12}(b^2 + 5a)t^2 + \frac{1}{24}abt^3 + \frac{1}{144}a^2t^4\right)$
3	$\left(1 - \frac{1}{2}bt + \frac{1}{12}(b^2 - a)t^2 + \frac{1}{120}(2ab - b^3)t^3 - \frac{1}{720}(a^2 + 3ab^2)t^4 - \frac{1}{1440}a^2bt^5 - \frac{1}{14400}a^3t^6\right)^{-1} \cdot$ $\left(1 + \frac{1}{2}bt + \frac{1}{12}(b^2 + 5a)t^2 + \frac{1}{120}(6ab + b^3)t^3 + \frac{1}{720}(11a^2 + 3ab^2)t^4 + \frac{1}{1440}a^2bt^5 + \frac{1}{14400}a^3t^6\right)$
4	$\left(1 - \frac{1}{2}bt + \frac{1}{28}(3b^2 - a)t^2 + \frac{1}{168}(3ab - 2b^3)t^3 + \frac{1}{1680}\left(\frac{9}{7}a^2 - 3ab^2 + b^4\right)t^4 + \frac{1}{23520}(111a^2b + 67ab^3)t^5 + \frac{1}{141120}(-2a^3 + 9a^2b^2)t^6 + \frac{1}{141120}a^3bt^7 + \frac{1}{1680} \frac{1}{1680}a^4t^8\right)^{-1} \cdot$ $\left(1 + \frac{1}{2}bt + \frac{1}{28}(3b^2 + 13a)t^2 + \frac{1}{168}(11ab + 2b^3)t^3 + \frac{1}{11760}(289a^2 + 98ab^2 + 7b^4)t^4 + \frac{1}{23520}(37a^2b + 7ab^3)t^5 + \frac{1}{141120}(38a^3 + 9a^2b^2)t^6 + \frac{1}{141120}a^3bt^7 + \frac{1}{1680} \frac{1}{1680}a^4t^8\right)$

Table 5-4 (a) Comparison of the stability of the PTI methods (A1 and A7) at $t=100s$, $N=1, 2$

	Exact Result	Results (Method A1) ($N=1$)		Results (Method A7) ($N=1$)	
		$\Delta t = 100s$	$\Delta t = 10s$	$\Delta t = 100s$	$\Delta t = 10s$
u_1	2.070749238	0	7.30482E+36	2.338508748	2.065395499
u_2	2.141458217	0	-2.04967E+37	2.191775874	1.902319571
u_3	1.876903554	0	2.97057E+37	1.889946823	1.93284633
u_4	1.803990314	42385525174	-3.31275E+37	1.416153404	1.672514708
u_5	1.092064632	-1.79842E+11	3.00647E+37	0.744738687	0.847753897
u_6	0.577849786	8481799655	-2.10666E+37	-0.159170468	0.27614994
u_7	0.359324835	5.71299E+11	7.83694E+36	-1.340236528	0.588232003
u_8	0.077951697	-4.42325E+11	-2.21411E+35	2.91828346	0.714788052
	Exact Result	Results (Method A1) ($N=2$)		Results (Method A7) ($N=2$)	
		$\Delta t = 100s$	$\Delta t = 10s$	$\Delta t = 100s$	$\Delta t = 10s$
u_1	2.070749238	-3.29344E+16	-1.048E+20	-0.193282693	2.096347417
u_2	2.141458217	1.98788E+17	2.92109E+20	-0.090354506	1.936485676
u_3	1.876903554	6.04937E+17	4.1726E+20	0.261447918	1.915551986
u_4	1.803990314	1.07755E+18	4.53542E+20	1.055086708	1.883186185
u_5	1.092064632	1.2142E+18	3.93078E+20	2.303005718	1.157105425
u_6	0.577849786	1.06842E+18	2.48518E+20	3.418279025	0.806179176
u_7	0.359324835	2.31226E+16	5.04154E+19	2.501784419	0.220726811
u_8	0.077951697	-4.69556E+17	-2.86167E+19	0.74403341	-0.015582675

Table 5-4 (b) Comparison of the stability of the PTI methods (**A1** and **A7**)
at $t=100s$, $N=5, 6$

	Exact Result	Results (Method A1) ($N=5$)		Results (Method A7) ($N=5$)	
		$\Delta t = 100s$	$\Delta t = 10s$	$\Delta t = 100s$	$\Delta t = 10s$
u_1	2.070749238	2.25437E+30	2.071181056	1.935377938	2.070810114
u_2	2.141458217	-6.75737E+30	2.139924106	2.035133967	2.141386381
u_3	1.876903554	1.11414E+31	1.878255595	1.921088681	1.876881793
u_4	1.803990314	-1.50318E+31	1.805100678	1.616911235	1.804403381
u_5	1.092064632	1.77508E+31	1.088191303	1.570710649	1.09121457
u_6	0.577849786	-1.84277E+31	0.58208929	1.043930769	0.57848632
u_7	0.359324835	1.62472E+31	0.358189124	0.023119858	0.359114154
u_8	0.077951697	-7.1768E+30	0.077068848	-0.146273096	0.077703287
	Exact Result	Results (Method A1) ($N=6$)		Results (Method A7) ($N=6$)	
		$\Delta t = 100s$	$\Delta t = 10s$	$\Delta t = 100s$	$\Delta t = 10s$
u_1	2.070749238	2.076363297	2.070816697	2.111210199	2.070804105
u_2	2.141458217	2.052099692	2.141423958	1.933411929	2.141485891
u_3	1.876903554	1.939019139	1.87693196	2.186436594	1.876893118
u_4	1.803990314	1.670418013	1.804071628	1.60518649	1.803972489
u_5	1.092064632	1.24243086	1.091741355	1.083761734	1.091969405
u_6	0.577849786	0.730261232	0.578026162	0.739382884	0.577791519
u_7	0.359324835	0.25956904	0.359181141	0.296312842	0.359230497
u_8	0.077951697	0.029838727	0.077807099	0.044297327	0.077852977

Table 5-5 Comparison of the efficiency and accuracy of the PTI methods

(A1 and A7) ($N_s=100$)

Time step size Δt (s)	$\ \Delta t \mathbf{H}\ _2$	Error	Method A1		Method A7	
			Optimum selection (p, N)	Computational cost (s)	Optimum selection (p, N)	Computational cost (s)
10.0	10^2	10^{-4}	(3,5)	5.097	(2,5)	5.768
		10^{-5}	(3,6)	6.890	(3,4)	6.629
		10^{-6}	(3,7)	7.521	(2,6)	7.360
		10^{-7}	(3,8)	7.790	(2,7)	7.390
1.0	10^1	10^{-4}	(4,2)	4.156	(2,1)	4.116
		10^{-5}	(4,3)	4.997	(2,2)	4.737
		10^{-6}	(3,5)	5.628	(2,3)	5.287
		10^{-7}	(3,6)	6.219	(2,4)	6.179
		10^{-8}	(4,6)	6.850	(4,1)	6.349
		10^{-9}	(4,7)	7.631	(4,2)	7.010
		10^{-10}	(4,10)	10.014	(4,5)	9.373
0.1	10^0	10^{-7}	(3,1)	8.432		
		10^{-8}	(3,3)	9.504	(2,1)	9.363
		10^{-9}	(4,4)	10.585	(2,2)	10.235
		10^{-10}	(4,5)	11.326	(2,3)	11.126

Improved Precise Time-Step Integration Algorithms for Dynamic Problems

Table 5-6 Comparison of the efficiency and accuracy of the PTI method (A7) with different order (p) ($N_s=100$)

Time step Δt (s)	Error	$p=1$		$p=2$		$p=3$		$p=4$	
		Scale factor N	Time cost (s)	Scale factor N	Time cost (s)	Scale factor N	Time cost (s)	Scale factor N	Time cost (s)
10.0	10^{-3}	3	4.116	4	5.317	3	6.149	1	5.878
	10^{-5}	10	8.983	5	5.768	4	6.629	2	6.264
	10^{-6}	12	10.946	6	6.519	10	11.546	3	7.010
	10^{-7}	15	13.189	7	7.360	12	13.629	4	7.490
1.0	10^{-4}	6	7.551	1	4.116	4	7.200		
	10^{-5}	7	8.202	2	4.737	6	8.182		
	10^{-6}	8	9.003	3	5.287	7	9.894		
	10^{-7}	10	10.565	4	6.179	8	11.096	1	6.349
0.1	10^{-4}	2	9.754			1	10.545		
	10^{-5}	4	11.386			2	11.476		
	10^{-6}	5	12.107			4	13.159		
	10^{-7}	6	12.788			6	14.521		
	10^{-8}			1	9.363	7	15.162		
	10^{-9}			2	10.235	8	16.333	1	11.917
0.05	10^{-4}	2	15.362						
	10^{-5}	3	16.383			1	16.724		
	10^{-6}	4	17.154			3	18.166		
	10^{-7}	5	17.906			5	19.658		
	10^{-9}			1	15.283	6	20.269		
	10^{-10}			2	16.063	7	20.88	1	17.645

Table 5-7 Comparison of the efficiency of the PTI methods (**A4** and **A8**)

Dimension of system (N_s)	Result of u_{N_s}	Computational cost (s)	
		Method A4 ($n=4$)	Method A8 ($p=2$)
50	1.60962609	0.610	0.514
100	2.03285323	5.588	4.416
200	3.89832698	61.819	49.421
300	6.87310525	242.598	178.095

Notes: $N=20$, $t=10.0s$, $\Delta t=10.0s$, Error= 10^{-7} .

CHAPTER SIX

Krylov Precise Time-Step Integration Algorithms

Overview

This chapter presents a further development of the precise time-step integration algorithms discussed in previous chapters. The dimensional expanding method has been investigated and employed to solve the dynamic equations in the precise time-step integration algorithms. These algorithms have shown the characters of efficiency and accuracy. In fact, these algorithms can be improved further through choosing the efficient algorithm for tackling the computing matrix generated by the dimensional expanding method. The Krylov subspace method and Padé series approximation are applied to modify the precise time-step integration algorithm in order to improve the computational efficiency. Both the stability and accuracy characteristics of the resultant algorithms are investigated. The efficiency can be further improved by expanding the dimension to avoid the computation of the particular solution. The present algorithms can also be extended to tackle nonlinear problems without difficulty. The Krylov precise time-step integration Algorithms are proposed in this chapter.

6.1 Introduction

In Chapters 3, 4 and 5, the modified precise time-step integration algorithms for normal dynamic problems are presented. The derived algorithms are very efficient for solving dynamic equations when the dimension of the system is not very large. The computational cost will increase tremendously as the dimension of the system becomes large.

In all the previous studies, multiplications of the system matrices are required. As the dimension of the matrices increases, the computational cost increases tremendously. As a result, it is necessary to find an efficient method to make the current algorithms applicable for solving large-scale dynamic problems. The Krylov subspace method can be used to reduce the computational cost.

In this chapter, the Krylov subspace method is combined with the precise time-step integration method to tackle large-scale transient problems. The computational efficiency is improved by reducing the dimension of the exponential matrix required in the evaluation. Furthermore, the present algorithms can still be used for non-linear systems. Also, the computational efficiency can be further improved by

1. employing the Padé approximations in computing the initial matrices required for the recurrence evaluations of the exponential matrices of reduced size, and
2. transforming the governing equations to an equivalent homogenous form by expanding the dimensions of the problems.

6.2 General Theory of Krylov Subspace Method

For a given N_s -square matrix $[A_r]$ and a nonzero vector $\{v_r\}$, the Krylov subspace defined by

$$\mathbf{K}_m \equiv \text{span}\{\{\mathbf{v}_r\}, [\mathbf{A}_r]\{\mathbf{v}_r\}, [\mathbf{A}_r]^2\{\mathbf{v}_r\}, \dots, [\mathbf{A}_r]^{m-1}\{\mathbf{v}_r\}\} \quad (6-1)$$

is referred as the m -th Krylov subspace associated with the pair $([\mathbf{A}_r], \{\mathbf{v}_r\})$ and is denoted by $\mathbf{K}_m([\mathbf{A}_r], \{\mathbf{v}_r\})$ or simply \mathbf{K}_m if there is no ambiguity.

The main feature of the Krylov subspace method is to seek an approximate solution to the original problem from this subspace. Thus, the original matrix problem of size N_s is approximated by another problem of size m , typically much smaller than N_s . Hence the computational cost will be reduced significantly.

Most of the Krylov subspace methods (e.g. Roland, 2000; Zhang et al., 2003) utilize either orthogonal or bi-orthogonal bases of \mathbf{K}_m . For non-symmetric matrices, the Arnoldi algorithm (Saad, 1981) can be used to build an orthogonal basis $[\mathbf{V}_m] = [\{\mathbf{v}_1\}, \{\mathbf{v}_2\}, \dots, \{\mathbf{v}_m\}]$ for the Krylov subspace \mathbf{K}_m such that $\{\mathbf{v}_i\}^T \{\mathbf{v}_j\} = 1$ if $i = j$ and $\{\mathbf{v}_i\}^T \{\mathbf{v}_j\} = 0$ otherwise for $1 \leq i \leq m$ and $1 \leq j \leq m$.

The Arnoldi algorithm can be summarized as follows

(1) Initialization:

$$\text{Compute } \{\mathbf{v}_1\} = \{\mathbf{v}_r\} / \|\mathbf{v}_r\|_2, \quad \gamma = \|\mathbf{v}_r\|_2, \quad (6-2)$$

(2) Iteration: Do $j = 1, 2, \dots$

$$(i) \quad \text{Compute } \{\tilde{\mathbf{h}}\} = [\mathbf{A}_r] \{\mathbf{v}_j\} \quad (6-3)$$

(ii) Do $i = 1, 2, \dots, j$

$$\text{Compute } h_{i,j} = (\tilde{\mathbf{h}}, \mathbf{v}_i) \quad (6-4)$$

$$\text{Compute } \{\tilde{\mathbf{h}}\} = \{\tilde{\mathbf{h}}\} - h_{i,j} \{\mathbf{v}_i\} \quad (6-5)$$

$$(iii) \quad \text{Compute } h_{j+1,j} = \|\tilde{\mathbf{h}}\|_2 \quad \text{and} \quad \{\mathbf{v}_{j+1}\} = \{\tilde{\mathbf{h}}\} / h_{j+1,j} \quad (6-6)$$

If the $[\mathbf{H}_m]$ is the $m \times m$ upper Hessenberg matrix containing the coefficients h_{ij} computed from the above algorithm, then it can be verified that

$$[\mathbf{H}_m] = [\mathbf{V}_m]^T [\mathbf{A}_r] [\mathbf{V}_m] \quad (6-7)$$

represents the projection of $[\mathbf{A}_r]$ into the subspace \mathbf{K}_m with respect to the basis $[\mathbf{V}_m]$. As mentioned in the research (Gallopoulos and Saad, 1992; Hochbruck and Lubich, 1997), the main idea of the Krylov subspace method is to approximate the matrix functions $f([\mathbf{A}_r])\{\mathbf{v}_r\}$ by

$$f([\mathbf{A}_r] \cdot \Delta t)\{\mathbf{v}_r\} \approx \gamma[\mathbf{V}_m] \cdot f([\mathbf{H}_m] \cdot \Delta t)\{\mathbf{e}_1\} \quad (6-8)$$

where $\{\mathbf{e}_1\} = \{1, 0, \dots, 0\}^T$. In other words, the computation of $f([\mathbf{A}_r] \cdot \Delta t)$ is substituted by the computation of $f([\mathbf{H}_m] \cdot \Delta t)$ with smaller dimensions.

Hence, the computation of exponential matrix $\exp([\mathbf{A}_r] \cdot \Delta t)$ can be replaced by the computation of $\exp([\mathbf{H}_m] \cdot \Delta t)$. If the initial vector is given as $\{\mathbf{v}_r\} = \{\mathbf{U}\}$ and $\gamma = \|\mathbf{U}\|_2$, then Equation (6-8) becomes

$$\exp([\mathbf{A}_r] \cdot \Delta t)\{\mathbf{U}\} \approx \gamma[\mathbf{V}_m] \cdot \exp([\mathbf{H}_m] \cdot \Delta t)\{\mathbf{e}_1\} \quad (6-9)$$

6.2.1 Implementation of Krylov Subspace Method

For hyperbolic problems (such as dynamic vibration of structures, wave transmission in fluids etc.), the resultant semi-discretized equations are second-order differential equations in time and can be written as

$$[\mathbf{M}]\{\ddot{\mathbf{u}}(t)\} + [\mathbf{C}]\{\dot{\mathbf{u}}(t)\} + [\mathbf{K}]\{\mathbf{u}(t)\} = \{\mathbf{r}(t)\}, \text{ where } \{\mathbf{r}(t)\} = [\mathbf{f}]\{\mathbf{Z}(t)\}. \quad (6-10)$$

With some conversion, Equation (6-10) can be rewritten as:

$$\{\dot{\mathbf{U}}(t)\} = [\mathbf{W}_4]\{\mathbf{U}(t)\} + \{\tilde{\mathbf{R}}(t)\} \quad (6-11)$$

where

$$\{\mathbf{U}(t)\} = \begin{Bmatrix} \mathbf{u}(t) \\ \mathbf{v}(t) \end{Bmatrix}, \{\tilde{\mathbf{R}}(t)\} = \begin{Bmatrix} 0 \\ \mathbf{M}^{-1}\mathbf{r}(t) \end{Bmatrix}, [\mathbf{W}_4] = \begin{bmatrix} \mathbf{0} & \mathbf{I} \\ \mathbf{A} & \mathbf{B} \end{bmatrix}. \quad (6-12)$$

According to the analysis in Chapter 3, $\{\tilde{\mathbf{R}}(t)\}$ can be removed by using of the dimensional expanding method. Then,

$$\{\dot{\mathbf{U}}^*(t)\} = [\mathbf{W}_3]\{\mathbf{U}^*(t)\} \quad (6-13)$$

The formula of vector $\{\mathbf{U}^*(t)\}$ and matrix $[\mathbf{W}_3]$ are given in Chapter 3. Equation (6-13) has the analytic solution:

$$\begin{cases} \{\mathbf{U}_0^*\} = \{\mathbf{U}^*(0)\} \\ \{\mathbf{U}_{k+1}^*(t)\} = \exp([\mathbf{W}_3] \cdot \Delta t) \cdot \{\mathbf{U}_k^*\}, k = 0, 1, \dots \end{cases} \quad (6-14)$$

In this chapter, the dimensional expanding method can also be used to eliminate the computation of the particular solutions. Since the Krylov subspace method can be used to reduce the dimension of the computing matrices, a slightly increase in the dimension of the original problem is not a concern even when dimension of matrix $[\mathbf{S}_1]$ gets larger. By combining the dimensional expanding method and the Krylov subspace method with the precise time-step integration method, the resultant algorithm is found to be very efficient.

By incorporating the Krylov subspace method, Equation (6-14) can be rewritten as:

$$\{\mathbf{U}_{k+1}^*\} = \exp([\mathbf{W}_3] \cdot \Delta t) \cdot \{\mathbf{U}_k^*\} = \gamma[\mathbf{V}_m] \cdot \exp([\mathbf{H}_m] \cdot \Delta t) \cdot \{\mathbf{e}_1\} \quad (6-15)$$

The stability of the Krylov subspace method process is based on the accuracy of computing $\exp([\mathbf{H}_m] \cdot \Delta t)$. Hence the exponential matrix $\exp([\mathbf{H}_m] \cdot \Delta t)$ should be tackled carefully. Since the dimension of the Hessenberg matrix $[\mathbf{H}_m]$ is small even though the dimension of original system may be large, the diagonal Padé series approximation is proposed to construct an approximation to $\exp([\mathbf{H}_m] \cdot \Delta t)$ accurately and efficiently.

$$\exp([\mathbf{H}_m] \cdot \Delta t) = \mathbf{P}_p([\mathbf{H}_m] \cdot \Delta t) = (\mathbf{P}_p([\mathbf{H}_m] \cdot \tau))^{2^N}, \tau = \Delta t/2^N \quad (6-16)$$

It should be noted that $[\mathbf{V}_m]$, $[\mathbf{H}_m]$ and $\mathbf{P}_p([\mathbf{H}_m] \cdot \tau)$ would have to be updated at every time step since the vector $\{\mathbf{U}\}$ in Equation (6-9) changes at every time step.

The reduced size matrix $[\mathbf{H}_m]$ is used to replace the computing matrix $[\mathbf{W}_3]$ in the Krylov subspace method. The scaling and squaring procedure in the precise time-step integration method described in Chapter 3 still can be used to compute $\exp([\mathbf{H}_m] \cdot \Delta t)$ accurately in the presented Krylov subspace process. As the

dimension of $[\mathbf{H}_m]$ is not large, it is possible to improve the computational efficiency by constructing $\exp([\mathbf{H}_m] \cdot \Delta t)$ in Equation (6-8) by using Padé series approximation instead of Taylor series approximation. The Krylov subspace method can make the expanded matrix dimension approximated by m , typically much smaller than the dimension of the expanding matrix.

The method (A1) incorporating with the current Krylov subspace method and Padé series approximation is denoted as the Krylov precise time-step integration algorithm (A9). The method (A3) incorporating with the Krylov subspace method and Padé series approximation is denoted as the Krylov precise time-step integration algorithm with dimensional expansion (A10).

6.3 Algorithm Analysis

An important issue of the present Krylov precise time-step integration method is on how to choose p (the order of the Padé approximation), N (the number of recursive evaluations) and m (the order of the Krylov subspace) so that the resultant algorithms are accurate and efficient. The criterion of selection the parameters p has been discussed in Example 5.2 in Chapter 5. The value of p is recommended as 2 or 4 in this chapter. The selection of the scale factor N and the criterion of selection m in the Krylov precise time-step integration methods can be obtained from the analysis of computational effort.

6.3.1 Error Estimation

For the present algorithm, the error estimation for the Padé series approximation and the Krylov subspace approximation should be considered together since the error for the Krylov subspace method depends on the computational accuracy of the

Padé series approximation. An inverse error analysis of the Padé series approximation in Chapter 5 can be used. If

$$\frac{\|[\mathbf{H}_m] \cdot \Delta t\|_\infty}{2^N} \leq \frac{1}{2} \quad (6-17)$$

then,

$$N \geq 1 + \frac{\ln(\|[\mathbf{H}_m] \cdot \Delta t\|_\infty)}{\ln 2} \quad (6-18)$$

Also, Equation (6-18) gives a criterion on the selection of N . In the original precise time-step integration method, $N = 20$ is used in all cases (Zhong and Williams, 1994). In the present algorithm, the value of N can be chosen according to Equation (6-18) easily since the dimension of matrix $[\mathbf{H}_m]$ is normally small. For example, $\|[\mathbf{H}_m] \cdot \Delta t\|_\infty$ in example 6.1 (2) ranges from 13.6 to 52.5, N ranges from 5 to 7 could be used in the computation.

The relative error level can be controlled by increasing the value of m . Obviously, the accuracy will increase with the dimension of the Krylov subspace m being enlarged. However, the computational effort will increase with m also. The algorithm may not be efficient if a large value of m is used. In fact, it has been observed that very accurate solutions can be obtained even when a relative small value of m is used. In Section 6.4, the range of m giving efficient algorithms can be obtained from the computational effort analysis.

6.3.2 Algorithm Stability

This section studies the stability of the present Krylov precise time-step integration methods (A9 and A10). These methods include two different parts, one is the Krylov subspace method process and the other is the Padé series approximation process. The stability of these two processes should all be considered.

Gallopoulos and Saad (1992) gave a detailed discussion on the stability behavior of Krylov subspace method. Their conclusion was based on the assumption that the exponential matrix $\exp([\mathbf{H}_m] \cdot \Delta t)$ could be computed accurately. It can be inferred that the present Krylov precise time-step integration methods will be stable also if the exponential matrix approximation is accurate enough. As the Padé series approximation and the scaling and squaring procedure in the precise time-step integration method are used in the computation, the matrix exponent indeed is very accurately computed. Hence the stability of the present algorithms (A9 and A10) can be guaranteed.

6.3.3 Non-Linear Systems

So far, the present methods are used to solve linear problems with time independent matrices. The present methods can also be extended to solve non-linear problems. For non-linear problems, the system matrices need to be reevaluated at every time step and the computational cost would be higher. However, the present Krylov precise time-step integration algorithm is still efficient because the Krylov subspace and the exponential matrix of reduced size would have to be recomputed at every time step whether the system is linear or non-linear. Hence the computational cost would not increase significantly when the method is applied to solve non-linear problems. In other words, the present method should still be efficient.

6.4 Computational Effort

In the following, the computational efforts for the proposed Krylov precise time-step integration methods (A9 and A10) and the traditional precise time-step integration method are considered.

The main equations of the precise time-step integration method with and without the dimensional expanding method are considered respectively as following:

$$\{\mathbf{U}_{k+1}^*\} = \exp([\mathbf{W}_3] \cdot \Delta t)_{(2N_s+g) \times (2N_s+g)} \cdot \{\mathbf{U}_k^*\} \quad (6-19)$$

and

$$\{\mathbf{U}_{k+1}\} = \exp([\mathbf{W}_4] \cdot \Delta t)_{2N_s \times 2N_s} \cdot \{\mathbf{U}_k\} + \int_0^{\Delta t} \exp([\mathbf{W}_4] \cdot (t-s)) \cdot \{\tilde{\mathbf{R}}(s)\} ds. \quad (6-20)$$

The Hessenberg matrix $[\mathbf{H}_m]$ and exponential function $\exp([\mathbf{H}_m] \cdot \Delta t)$ need be evaluated every time step in the present Krylov precise time-step integration methods. Fortunately, the dimension of the Hessenberg matrix $[\mathbf{H}_m]$ is very small. According to the analysis in Chapter 3, the dimensional expanding method can reduce the computational cost significantly when the dimension of the system is large. Hence the method (A10) will be better than the method (A9) in practice computing process.

Next, the computational effort for the present Krylov precise time-step integration method (A10) is investigated. It is compared with the precise time-step integration method with dimensional expansion (A3). Both banded matrices and full matrices are considered.

Firstly, if the half bandwidth of the problem $b_h \gg 1$ and the total number of degree-of-freedom $N_s \gg b_h$, the number of operations for the dimensional expanding precise time-step integration method (A3) and the Krylov precise time-step integration method with dimensional expansion (A10) are given in the Table 6-1(a, b). N_{PTI} is the scaling factor used in the original PTI method. Normally, the value of N_{PTI} is chosen as 20.

It can be verified that the present method (A10) is more efficient if

$$\begin{aligned} & \frac{T}{\Delta t} [2mb_h(2N_s + g) + 2m^2(2N_s + g) + (N + 1.5)m^3] \\ & < \frac{T}{\Delta t} (2^{N_{PTI}+1} \cdot 4b_h)(2N_s + g) + 20b_h \cdot (2N_s + g) + \frac{4^3(4^{N_{PTI}} - 1)}{3} b_h^2(2N_s + g) \end{aligned} \quad (6-21)$$

or

$$\frac{T}{\Delta t} \left[(2mb_h + 2m^2 - 2^{N_{PTI}+1} \cdot 4b_h) + \frac{(N+1.5)m^3}{(2N_s + g)} \right] < 20b_h + \frac{4^3(4^{N_{PTI}} - 1)}{3} b_h^2 \quad (6-22)$$

Secondly, if the matrices are full, the number of operations for the precise time-step integration method with dimensional expanding and the Krylov precise time-step integration method with dimensional expanding (A10) are also given in Table 6-2 (a, b). Hence, for the dimensional expanding precise time-step integration method via Equation (6-13), the total computational counts should be $[(2N_s + g)^2 + (N_{PTI} + 2)(2N_s + g)^3]$. For the Krylov precise time-step integration method with dimensional expansion (A10), the total computational counts should be $[m(2N_s + g)^2 + 2m^2(2N_s + g) + (N + 1.5)m^3]$.

As a result, the Krylov subspace method with dimensional expanding (A10) will be more efficient than the dimensional expanding precise time-step integration method (A3) if

$$\begin{aligned} & \frac{T}{\Delta t} [m(2N_s + g)^2 + 2m^2(2N_s + g) + (N + 1.5)m^3] \\ & < \frac{T}{\Delta t} (2N_s + g)^2 + (N_{PTI} + 2)(2N_s + g)^3 \end{aligned} \quad (6-23)$$

or

$$\frac{T}{\Delta t} \cdot \frac{1}{(N_{PTI} + 2)} \cdot \left[\frac{(m-1)}{(2N_s + g)} + \frac{2m^2}{(2N_s + g)^2} + \frac{(N+1.5)m^3}{(2N_s + g)^3} \right] < 1 \quad (6-24)$$

It can be seen from Equation (6-24) that the range of m increases with Δt . If $\Delta t = T$, then m can be a very large value (even near $2N_s$). On the other hand, the Krylov precise time-step integration method will not be efficient when the time step size is too small. Fortunately, the restriction of the time step size is relaxed by using the precise time-step integration method. In conclusion, the Krylov precise time-step integration method with dimensional expansion (A10) can further improve the efficiency of the original dimensional expanding precise time-step integration method (A3) when m is in the range given by Equation (6-24).

For non-linear problems, the matrix $[\mathbf{W}_3]$ changes in each time step. For the precise time-step integration method with dimensional expanding, this will increase the computational effort enormously. On the other hand, the impact on the present Krylov precise time-step integration method is minimal as $[\mathbf{H}_m]$ has to be reformulated at every time-step anyway. Hence, for non-linear problems, Equation (6-23) becomes

$$\begin{aligned} & \frac{T}{\Delta t} [m(2N_s + g)^2 + 2m^2(2N_s + g) + (N + 1.5)m^3] \\ & < \frac{T}{\Delta t} [(2N_s + g)^2 + (N_{PTI} + 2)(2N_s + g)^3] \end{aligned} \quad (6-25)$$

Then, it can be simplified as

$$\frac{1}{(N_{PTI} + 2)} \cdot \left[\frac{(m-1)}{(2N_s + g)} + \frac{2m^2}{(2N_s + g)^2} + \frac{(N + 1.5)m^3}{(2N_s + g)^3} \right] < 1 \quad (6-26)$$

In fact, if m satisfies Equation (6-24), Equation (6-26) will be satisfied automatically. The current Krylov precise time-step integration method will be efficient if the value of m used in the computation is not beyond the permitting efficient range determined by Equation (6-24) (or Equation (6-26) for non-linear problems). Hence Equation (6-24) (or Equation (6-26) for non-linear problems) can be used check the value of m .

6.5 Numerical Examples

In the following, two numerical examples are given to demonstrate the numerical accuracy, flexibility and efficiency of the present Krylov precise time-step integration method (with or without dimensional expanding) in solving large-scale dynamic problems.

Example 6.1: General Dynamic Systems

The Example is used whose details were introduced in Chapter 3. All results are compared at time $t=100.0s$.

(1) *The PTI and the Present Krylov PTI Methods*

The results given by the original precise time-step integration method with and without dimensional expansion and the present Krylov precise time-step integration method (A9) are given in Table 6-3.

For the present Krylov precise time-step integration method (A9), the order of Padé approximations p is chosen as 4. The values of N are determined from Equation (6-18) at every time step and the values of m used in the computing process are in the efficient range determined from the Equation (6-24).

For original precise time-step integration method, the larger time step size, the more computational cost can be saved. The computational cost of the original precise time-step integration method will be the least using the largest time step size (one step iteration). The Krylov precise time-step integration method need consider Δt and m together so that more computational cost can be saved. The influence of Δt and m will be discussed later. In fact, the largest time step size is not the most optimal choice in Krylov precise time-step integration method in this example. The computational cost of Krylov precise time-step integration method will increase significantly with m increasing because the big m must be chosen to guarantee the accuracy the for large time step size. Table 6-3 notes that the Krylov precise time-step integration method (A9) has the high efficiency by using $\Delta t = 10.0s$ instead of $\Delta t = 100.0s$.

Table 6-3 shows that the Krylov precise time-step integration method (A9) is more efficient than the original PTI method with and without dimensional expansion when the dimension of the system is lager than 100 at $\Delta t = 10.0s$. As the

dimension of the system increases, the advantage of the present algorithm becomes more obvious.

(2) *Krylov PTI Method With and Without Dimensional Expansion*

The dimensional expanding method can be used to improve the computing efficiency. In the example, the linear loading form can be expressed as:

$$\{\mathbf{r}(t)\} = \{\mathbf{r}_0\} + \{\mathbf{r}_1\} \times z(t) \text{ where } z(t) = t, \text{ and } z_0(t) = 1, z_1(t) = t \quad (6-27)$$

The governing equations after the application dimensional expanding method are

$$\begin{Bmatrix} \dot{\mathbf{u}} \\ \dot{\mathbf{v}} \\ \dot{\mathbf{Z}} \end{Bmatrix}_{(2N_s+2) \times 1} = \begin{bmatrix} \mathbf{0} & \mathbf{I} & \mathbf{0} \\ \mathbf{A} & \mathbf{B} & \mathbf{M}^{-1}\mathbf{f} \\ \mathbf{0} & \mathbf{0} & \mathbf{S}_1 \end{bmatrix}_{(2N_s+2) \times (2N_s+2)} \begin{Bmatrix} \mathbf{u} \\ \mathbf{v} \\ \mathbf{Z} \end{Bmatrix}_{(2N_s+2) \times 1} \quad (6-28)$$

where

$$[\mathbf{f}] = [\mathbf{r}_0 \quad \mathbf{r}_1], [\mathbf{S}_1] = \begin{bmatrix} 0 & 0 \\ 1 & 0 \end{bmatrix}. \quad (6-29)$$

The initial displacement and velocity vectors $\{\mathbf{u}_0\}$, $\{\dot{\mathbf{u}}_0\}$ and $\{\mathbf{Z}_0\}$ are given by

$$\{\mathbf{u}_0\} = \{0 \quad 0 \quad \dots \quad 0 \quad 1\}^T, \{\dot{\mathbf{u}}_0\} = \{0 \quad 0 \quad \dots \quad 0 \quad 1\}^T \text{ and } \{\mathbf{Z}_0\} = \{1 \quad 0\}^T \quad (6-30)$$

The numerical results given by the present Krylov precise time-step integration with and without dimensional expanding (A10 and A9) are shown in Table 6-4. It can be seen that the efficiency can be improved by incorporating the dimensional expanding method into the Krylov precise time-step integration method when the system dimension is large than 300. Hence, the Krylov precise time-step integration method with dimensional expansion (A10) can be used to solve large-scale transient problems more efficient than the original Krylov precise time-step integration method (A9).

(3) *Influence of m and Δt*

Table 6-5, 6-6 and 6-7 show the results given by the present Krylov precise time-step integration method with dimensional expansion (A10) for $N_s=100, 200$ and 400 ,

respectively, with different values of m and Δt . From Table 6-5 and 6-7, it can be seen that in order to reach a certain error level, m decreases with Δt . For example, if $N_s=100$, $g=2$, $\Delta t=1.0s$, $N_{PTI} = N = 20$, and the order of the Padé approximation $p=4$, $m=10$ is sufficient to bring the relative error for the result x_{N_i} to around 10^{-7} . This value of m satisfies the range ($m < 28$) determined by the Inequality (6-24). Hence, the Krylov precise time-step integration method is more efficient than the original precise time-step integration method. Furthermore, it can be seen that the present Krylov precise time-step integration methods (A9 and A10) can be used to obtain accurate results even when the time step size is large.

(4) Influence of Ill-Conditioned Stiffness Matrix

The ill-conditioned stiffness matrix would yield inaccurate solutions for static problems only. For dynamic problems, the singularity in the stiffness matrices is not a major concern as the mass matrices are in general non-singular. The original precise time-step integration need compute the inverse matrix of $[\mathbf{W}_3]$. The ill-conditioned stiffness matrix $[\mathbf{K}]$ will influence the accuracy. But in the presented Krylov precise time-step integration with dimensional expanding method, the inverse matrix of $[\mathbf{W}_3]$ need not be computed. The ill-conditioned stiffness matrix would not influence the computing accuracy and stability. Furthermore, the stiffness matrix $[\mathbf{K}]$ even can be a zero matrix. This example consider with zero-stiffness.

Let $[\mathbf{K}]$ be a zero matrix, the singular computing matrix $[\mathbf{W}_3]$ will be written as:

$$[\mathbf{W}_3] = \begin{bmatrix} 0 & 0 & \mathbf{I} \\ 0 & \mathbf{B} & \mathbf{M}^{-1}\mathbf{f} \\ 0 & 0 & \mathbf{S}_1 \end{bmatrix} \quad (6-31)$$

The results computed by Krylov precise time-step integration with dimensional expansion method and the precise time-step integration with dimensional expansion method are given in Table 6-8. It notes that the presented Krylov precise time-step

integration method with dimensional expansion (A10) still can get the accurate results same as the original precise time-step integration method with dimensional expansion (A3).

Example 6.2: Non-Linear Problem

Consider the following N_s -degree-of-freedom nonlinear system as:

$$[\mathbf{M}]\{\ddot{\mathbf{u}}(t)\} + [\mathbf{C}]\{\dot{\mathbf{u}}(t)\} + [\mathbf{K}(t)]\{\mathbf{u}(t)\} = \{\mathbf{r}_0\} + t\{\mathbf{r}_1\}, \quad t \in [0,1] \quad (6-32)$$

Where $[\mathbf{M}]$, $[\mathbf{C}]$, $\{\mathbf{r}_0\}$, $\{\mathbf{r}_1\}$ and the initial conditions $\{\mathbf{u}_0\}$ and $\{\dot{\mathbf{u}}_0\}$ are refer to the Equations (3-52) and (3-53) in Chapter 3. The matrix $[\mathbf{K}(t)]$ is assumed to be

$$[\mathbf{K}(t)] = (1 + \cos(t)) \cdot [\mathbf{K}_0] \quad (6-33)$$

Where

$$[\mathbf{K}_0] = \begin{bmatrix} 8 & -4 & & & \\ -4 & 8 & -4 & 0 & \\ & -4 & \ddots & \ddots & \\ & & 0 & \ddots & 8 & -4 \\ & & & & -4 & 4 \end{bmatrix} \quad (6-34)$$

The computational costs with error level 10^{-3} and 10^{-4} are shown in Table 6-9. it can be seen that the original precise time-step integration method is not suitable for non-linear problems as the computational cost would increase tremendously with the dimension of the matrices. On the other hand, the increase of the computational effort for the present method (A10) is moderate. The advantages of the present Krylov precise time-step integration method with dimensional expansion (A10) have been demonstrated clearly.

6.6 Summary

In this chapter, the Krylov precise time-step integration methods (**A9** and **A10**) are presented. The Krylov subspace method is used to reduce the dimensions of the computing matrices. To improve the computational efficiency, the equal order Padé approximation is used to evaluate the initial matrix to be used in the scaling and squaring procedure. To improve the computational efficiency further, the dimensional expanding method can be incorporated to transform the non-homogeneous differential equations into an equivalent homogeneous form. The criteria to choose N (number of recursive evaluations), p (order of the Padé approximations) and the efficient range of m (order of the Krylov subspace) have been studied. The Krylov precise time-step integration methods (**A9** and **A10**) are shown to be more efficient than the original precise time-step integration methods. The Krylov precise time-step integration method with dimensional expansion (**A10**) has also been extended to solve non-linear problems. The additional computational costs are not very high and certainly will be much less than the original precise time-step integration method. Two numerical examples are included to illustrate the highly accurate and efficient algorithms.

Table 6-1 (a) Operation counts of the method **A3** (Banded matrices)

Category	Banded Matrices
Form $[\mathbf{T}_a^*]$	$2b \cdot 2b \cdot (2N_s + g) + 4b \cdot 4b \cdot (2N_s + g)$ $= 20b(2N_s + g)$
Form $[\mathbf{T}^*]$	$(8b)^2 + (2 \cdot 8b)^2(2N_s + g) + \dots + (2^{N_{PTI}-1} \cdot 8b)^2(2N_s + g)$ $= \frac{4^3(4^{N_{PTI}} - 1)}{3} b^2(2N_s + g)$
$[\mathbf{T}^*]\{\mathbf{u}^*\}$	$(2^{N_{PTI}+1} \cdot 4b)(2N_s + g)$

Table 6-1 (b) Operation counts of the method **A10** (Banded matrices)

Category	Banded Matrices
Form $[\mathbf{H}_m]$	$m(2b)(2N_s + g) + \frac{m[(2N_s + g) + m(2N_s + g)]}{2}$ $= 2mb(2N_s + g) + \frac{1}{2}m^2(2N_s + g) + \frac{1}{2}m(2N_s + g)$ $\approx 2mb(2N_s + g) + m^2(2N_s + g)$
Form $[\mathbf{T}_{ma}]$	$\frac{1}{2}m^3 + m^3$
Form $[\mathbf{T}_m]$	Nm^3
$[\mathbf{T}^*]\{\mathbf{u}^*\}$	$m^2(2N_s + g)$

Table 6-2 (a) Operation counts of the method **A3** (Full matrices)

Category	Full Matrices
Form $[\mathbf{T}_a^*]$	$2(2N_s + g)^3$
Form $[\mathbf{T}^*]$	$N_{PTI}(2N_s + g)^3$
$[\mathbf{T}^*]\{\mathbf{u}^*\}$	$(2N_s + g)^2$

Table 6-2 (b) Operation counts of the method **A10** (Full matrices)

Category	Full Matrices
Form $[\mathbf{H}_m]$	$m(2N_s + g)^2 + \frac{m[(2N_s + g) + m(2N_s + g)]}{2}$ $= m(2N_s + g)^2 + \frac{1}{2}m^2(2N_s + g) + \frac{1}{2}m(2N_s + g)$ $= m(2N_s + g)^2 + m^2(2N_s + g)$
Form $[\mathbf{T}_{ma}]$	$\frac{1}{2}m^3 + m^3$
Form $[\mathbf{T}_m]$	Nm^3
$[\mathbf{T}^*]\{\mathbf{u}^*\}$	$m^2(2N_s + g)$

Table 6-3 Computational effort to evaluate the result u_{N_s} at $t=100s$ for various N_s

Dimension of system (N_s)		100	150	200	300	400
Result u_{N_s}		825.991465	825.991465	852.437005	825.813988	825.991465
Error		10^{-7}	10^{-7}	10^{-7}	10^{-7}	10^{-7}
Method A1 ($N=20$)	Time step	$\Delta t = 10s$	$\Delta t = 10s$	$\Delta t = 10s$	$\Delta t = 10s$	$\Delta t = 10s$
	Time cost (s)	17.589	61.999	152.950	524.867	1307.785
	Time step	$\Delta t = 100s$	$\Delta t = 100s$	$\Delta t = 100s$	$\Delta t = 100s$	$\Delta t = 100s$
	Time cost (s)	16.634	58.484	149.435	516.512	1243.054
Method A3 ($N=20$)	Time step	$\Delta t = 10s$	$\Delta t = 10s$	$\Delta t = 10s$	$\Delta t = 10s$	$\Delta t = 10s$
	Time cost (s)	15.021	49.681	124.338	411.732	956.065
	Time step	$\Delta t = 100s$	$\Delta t = 100s$	$\Delta t = 100s$	$\Delta t = 100s$	$\Delta t = 100s$
	Time cost (s)	13.559	46.097	111.520	381.398	885.152
Method A9 ($p=4$)	Time step	$\Delta t = 10s$	$\Delta t = 10s$	$\Delta t = 10s$	$\Delta t = 10s$	$\Delta t = 10s$
	m	28	28	29	28	29
	Time cost (s)	9.564	26.187	47.909	106.473	256.829
	Time step	$\Delta t = 100s$	$\Delta t = 100s$	$\Delta t = 100s$	$\Delta t = 100s$	$\Delta t = 100s$
	m	165	165	165	165	165
	Time cost (s)	57.813	173.640	295.655	705.813	1186.106

Notes: For the Krylov precise time-step integration method (**A9**), N in each time step depends on $\|\mathbf{H}_m \Delta t\|_\infty$. It is ranging from 11 to 13 at $\Delta t = 10s$ and from 19 to 20 at $\Delta t = 100s$ in the present calculations.

Table 6-4 Computational effort to evaluate the result u_{N_s} at $t=100s$
with $\Delta t = 1.0s$ for various N_s

Dimension of system (N_s)	Method A9			Method A10		
	Time cost (s)	m	Error	Time cost (s)	m	Error
40	1.862	10	10^{-7}	2.393	11	10^{-7}
100	14.190	10	10^{-7}	15.843	11	10^{-7}
150	39.076	10	10^{-7}	41.209	11	10^{-7}
200	63.392	10	10^{-7}	65.765	10	10^{-7}
300	153.410	10	10^{-7}	150.045	11	10^{-7}
400	287.654	10	10^{-7}	269.076	10	10^{-7}
500	477.777	9	10^{-7}	415.177	10	10^{-7}
600	750.328	9	10^{-7}	597.429	10	10^{-7}
700	1039.725	9	10^{-7}	830.504	10	10^{-7}
800	1726.673	9	10^{-7}	1072.032	10	10^{-7}

Notes: The order of the Padé series approximation p is 4. N in each time step depends on $\|\mathbf{H}_m \Delta t\|_\infty$. The range of N for method A9 is from 5 to 7. The range of N for method A10 is from 5 to 8.

Table 6-5 Computational effort for the presented method **A10** to evaluate the result u_{N_s} at $t=100s$ ($N_s=100$)

Time step Δt (s)	m	Time cost (s)	Error
0.01	3	199.356	10^{-3}
	5	390.151	10^{-7}
	10	1259.881	10^{-8}
0.1	5	40.919	10^{-3}
	6	55.279	10^{-7}
	7	70.381	10^{-8}
1	6	6.679	10^{-5}
	8	11.006	10^{-6}
	10	15.843	10^{-7}
	12	23.774	10^{-8}
10	24	6.870	10^{-3}
	26	7.891	10^{-5}
	29	10.215	10^{-7}
	30	10.575	10^{-8}

Table 6-6 Computational effort for the presented method **A10** to evaluate the result u_{N_s} at $t=100s$ ($N_s=200$)

Time step Δt (s)	m	Time cost (s)	Error
0.01	4	1377.440	10^{-6}
	5	1992.405	10^{-7}
	8	4519.5090	10^{-8}
0.1	5	198.0450	10^{-6}
	6	267.334	10^{-8}
1	8	45.105	10^{-3}
	9	55.149	10^{-5}
	10	65.765	10^{-7}
	12	92.062	10^{-8}
	13	105.582	10^{-8}
10	22	28.120	10^{-3}
	23	30.454	10^{-4}
	29	48.960	10^{-7}
	30	52.005	10^{-8}

Table 6-7 Computational effort for the presented method **A10** to evaluate the result u_{N_s} at $t=100s$ ($N_s=400$)

Time step Δt (s)	m	Time cost (s)	Error
0.01	4	5667.169	10^{-7}
	5	8087.079	10^{-8}
0.1	5	811.267	10^{-6}
	6	1094.053	10^{-7}
	7	1422.986	10^{-8}
1	8	180.730	10^{-5}
	9	222.870	10^{-6}
	10	269.076	10^{-7}
	12	373.507	10^{-8}
10	23	128.084	10^{-3}
	25	138.709	10^{-4}
	29	226.936	10^{-7}
	31	270.969	10^{-8}

Table 6-8 The influence of ill-conditioned stiffness matrices to
the proposed Krylov PTI method (A10)

Dimension of system (N_s)	Exact Results of u_{N_s}	Results of u_{N_s}	Method A3 ($N=20$)	Method A10	
			Time cost (s)	Time cost (s)	m
50	1.246671	1.246671	0.697	1.181	20
100	1.309723	1.309723	12.362	4.476	20
150	1.340487	1.340487	41.430	9.764	20
200	1.246671	1.246671	103.588	17.034	20
300	1.340487	1.340487	359.153	40.328	20

Notes: $T=100s$, $\Delta t = 10.0s$. For the Krylov PTI method, N in each time step depends on $\|\mathbf{H}_m \cdot \Delta t\|_\infty$. It is ranging from 10 to 13.

Improved Precise Time-Step Integration Algorithms for Dynamic Problems

Table 6-9 Computational effort to evaluate result u_{N_s} , at $t=1.0s$ in Example 2

Dimension of system (N_s)			40	100	200	300
Method A1	Error= 10^{-3}	Δt	0.01	0.01	0.01	0.01
		Time cost (s)	51.634	1373.385	11763.185	41378.319
	Error= 10^{-4}	Δt	0.001	0.001	0.001	0.001
		Time cost (s)	545.103	13632.392	117586.256	413650.523
Method A10	Error= 10^{-3}	Δt	0.01	0.01	0.01	0.01
		Time cost (s)	0.65	6.279	25.467	59.706
		m	5	6	6	6
	Error= 10^{-4}	Δt	0.001	0.001	0.001	0.001
		Time cost (s)	6.899	46.046	198.295	454.023
		m	6	5	5	5

CHAPTER SEVEN

Application of the Improved Precise Time-Step Integration Algorithms

Overview

The previous chapters have investigated the efficiency of the proposed precise time-step integration algorithms and the implementation of the algorithms. The proposed precise time-step integration algorithms in this research are efficient numerical schemes for ODEs (Ordinary Differential Equations), which can be extended to other field such as transient heat conduction, physical problems, and multi-body dynamic, etc. To form a systematic introduction of the practice application of the proposed precise time-step integration algorithms, a short summary of the application of the proposed precise time-step integration algorithms is presented in this chapter.

7.1 Introduction

In order to obtain an effective solution of a dynamic response, it is important to choose an appropriate time integration scheme. This choice depends on the finite element idealization, which in turn depends on the actual physical problem to be analyzed. The selection of an appropriate finite element idealization of a problem and the choice of an effective integration scheme for the response solution are closely related and must be considered together. With the precise time-step integration algorithms proposed in this thesis, different kinds of dynamic problems after the finite element model idealization can be solved efficiently. Further dynamic analyses of different problems are all based on solving the dynamic equations accurately. The computational cost of the whole analysis process can be reduced when the dynamic equations are solved efficiently.

Recently, the precise time-step integration (PTI) method has been employed to find the structural responses under the evolutionary random excitations (Leung, 1993; Lin et al., 1995), to solve time dependent partial differential equations (Zhong et al., 1996) and nonlinear transient heat conduction problems (Chen et al., 2001), to determine the material derivative of the convective function in the operator splitting procedure (Li et al., 2000), to evaluate the Floquet transition matrix (Cai et al., 2001), to solve the asymmetric Riccati differential equations (Zhong, 2001; Zhong 2004) and to study the dynamic responses of a continuous beam under moving loads (Zhu and Law, 2001).

Wang et al. (2002) studied the high precise integration scheme for analysis of dynamical response of the structures with elastic or rigid modes. The scheme can be used in the excitations with linear as well as polynomial forms. Moreover, if the non-linear excitation can be presented in the polynomial series, the dynamic response of structure is also obtained by the proposed method, and the computed precision can be adjusted by changing the order of the polynomial.

Wu and Jin (2004) developed the precise time-step integration method to analyze arch dam-foundation system that combines pseudo excitation with the finite element, boundary element and infinite boundary element coupled model. The precise time integration method also was used to analyze the critical wind velocity of galloping oscillation of high-rise structures (Zhang et al., 2004). Later, the precise time-step integration method was also applied to solve wave equation (Zhang and Zhang, 2004), viscoelastic solid constitutive equation (Liu et al., 2004) and large-scale structure by using the sparse transform of dynamic equations and series solution of integral equations (Li et al., 2005), etc. Most of those methods are efficient and accurate.

The application of the proposed precise time-step integration algorithms can be widely used, such as general ODE problems, moving load problems and analysis of critical wind velocity of galloping oscillation of high-rise structures. These problems are considered briefly in this chapter.

7.2 Improvement for Solving ODEs

Computation of the ODEs is the most important part in the analysis process for solving dynamic response. Most dynamic analysis requires evaluating the dynamic response by solving the ODEs. Hence, the development of the algorithms for solving the ODEs efficient and accurate is very important.

The extended dimensional expanding method, Padé series approximation and Krylov subspace method are employed to improve the precise time-step integration (PTI) methods in this research. When the extended dimensional expanding method is used, the first-order and second-order non-homogeneous equations can be tackled directly without conversion. Thus, the method not only avoids the computation of the inverse matrix but also improves the computational efficiency. In order to further introduce the

Where $\{\mathbf{u}_i\}$, $\{\dot{\mathbf{u}}_i\}$ and $\{\ddot{\mathbf{u}}_i\}$ are the approximations to the nodal displacement, velocity, and acceleration vectors respectively at time $t = i \cdot \Delta t$, Δt is the time step and β_1 and β_2 are the Newmark parameters.

Wilson (1962) formulated Newmark's method in matrix notation, adding stiffness and mass proportional damping, and eliminating the need for iteration by introducing the direct solution of equations at each time step. This requires that Equations (7-5a) and (7-5b) to be rewritten in the following forms:

$$\{\ddot{\mathbf{u}}_{i+1}\} = b_1 \cdot (\{\mathbf{u}_{i+1}\} - \{\mathbf{u}_i\}) + b_2 \{\dot{\mathbf{u}}_i\} + b_3 \{\ddot{\mathbf{u}}_i\} \quad (7-6a)$$

$$\{\dot{\mathbf{u}}_{i+1}\} = b_4 \cdot (\{\mathbf{u}_{i+1}\} - \{\mathbf{u}_i\}) + b_5 \{\dot{\mathbf{u}}_i\} + b_6 \{\ddot{\mathbf{u}}_i\} \quad (7-6b)$$

where,

$$b_1 = \frac{1}{\beta_1 \Delta t^2}, b_2 = -\frac{1}{\beta_1 \Delta t}, b_3 = 1 - \frac{1}{2\beta_1}, \quad (7-6c)$$

$$b_4 = \beta_2 \Delta t b_1, b_5 = 1 + \beta_2 \Delta t b_2, b_6 = \Delta t (1 - \beta_2 + \beta_2 b_3).$$

Substituting Equations (7-6a) and (7-6b) into Equation (7-1), we have

$$\begin{aligned} (b_1[\mathbf{M}] + b_4[\mathbf{C}] + [\mathbf{K}])\{\mathbf{u}_{i+1}\} &= \{\mathbf{r}_{i+1}(t)\} + [\mathbf{M}](b_1\{\mathbf{u}_i\} - b_2\{\dot{\mathbf{u}}_i\} - b_3\{\ddot{\mathbf{u}}_i\}) \\ &+ [\mathbf{C}](b_4\{\mathbf{u}_i\} - b_5\{\dot{\mathbf{u}}_i\} - b_6\{\ddot{\mathbf{u}}_i\}) \end{aligned} \quad (7-7)$$

The Newmark method is unconditionally stable if the parameters β_1 and β_2 are selected as

$$2\beta_1 \geq \beta_2 \geq \frac{1}{2} \quad (7-8)$$

Note in this example, the condition, $2\beta_1 = \beta_2 = \frac{1}{2}$, is used.

(2) Implementation of Method A6

Equation (7-1) can be transformed into the following formula by using the method A6.

$$\{\dot{\mathbf{U}}^*\} = [\mathbf{W}_3] \cdot \{\mathbf{U}^*\} \quad (7-9)$$

Where

$$\{\mathbf{U}^*\} = \begin{Bmatrix} \mathbf{u} \\ \mathbf{v} \\ \mathbf{Z} \end{Bmatrix}, [\mathbf{W}_3] = \begin{bmatrix} \mathbf{0} & \mathbf{I} & \mathbf{0} \\ \mathbf{A} & \mathbf{B} & \mathbf{M}^{-1}\mathbf{f} \\ \mathbf{0} & \mathbf{0} & \mathbf{S}_1 \end{bmatrix}, [\mathbf{S}_1] = \begin{bmatrix} 0 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 2 & 0 \end{bmatrix} \quad (7-10)$$

$$\{\mathbf{r}(t)\} = [\mathbf{f}]\{\mathbf{Z}(t)\}, [\mathbf{f}] = [\{\mathbf{f}_0\} \quad \{\mathbf{f}_1\} \quad \{\mathbf{f}_2\}], \{\mathbf{Z}(t)\} = \{1 \quad t \quad t^2\}^T. \quad (7-11)$$

Thus, the solution for Equation (7-9) can be written as:

$$\begin{Bmatrix} \mathbf{u}(t) \\ \mathbf{v}(t) \\ \mathbf{Z}(t) \end{Bmatrix} = \begin{bmatrix} \mathbf{G}(t) & \mathbf{H}(t) & \mathbf{D}(t) \\ \dot{\mathbf{G}}(t) & \dot{\mathbf{H}}(t) & \dot{\mathbf{D}}(t) \\ \mathbf{0} & \mathbf{0} & \mathbf{E}(t) \end{bmatrix} \cdot \begin{Bmatrix} \mathbf{u}_0 \\ \mathbf{v}_0 \\ \mathbf{Z}_0 \end{Bmatrix} \quad (7-12)$$

The response matrices can be obtained by the algorithm from Chapter 4. After the response matrices are evaluated, the time-step integration can be carried out with the selected time-step size. Hence, $\{\mathbf{u}_{i+1}\}$ and $\{\mathbf{v}_{i+1}\}$ at $t = t_{i+1}$ can be obtained from $\{\mathbf{u}_i\}$ and $\{\mathbf{v}_i\}$ at $t = t_i$ by matrix multiplication. In present PTI method, the desirable time-step size Δt can be chosen independent of the highest frequency in the model. The number of recursions required is determined from the stability requirement.

(3) Implementation of Method A10

The solution of Equation (7-9) can be obtained directly as

$$\begin{cases} \{\mathbf{U}_0^*\} = \{\mathbf{U}^*(0)\} \\ \{\mathbf{U}_{k+1}(t)\} = \exp(\Delta t \cdot [\mathbf{W}_3])\{\mathbf{U}_k\}, k = 0, 1, 2, \dots \end{cases} \quad (7-13)$$

With the method (A10),

$$\{\mathbf{U}_{k+1}(t)\} = \exp(\Delta t \cdot [\mathbf{W}_3])\{\mathbf{U}_k\} = \gamma[\mathbf{V}_m] \exp([\mathbf{H}_m] \cdot \Delta t) \cdot \{\mathbf{e}_1\} \quad (7-14)$$

$$\text{Then, } \exp([\mathbf{H}_m] \cdot \Delta t) = (\mathbf{P}_p([\mathbf{H}_m] \cdot \tau))^{2^N}, \tau = \frac{\Delta t}{2^N} \quad (7-15)$$

Where $[\mathbf{V}_m]$ and $\mathbf{P}_p([\mathbf{H}_m] \cdot \tau)$ should be updated at every time step regardless that the system is linear or non-linear. The Krylov subspace method can make the expanded matrix dimension approximated by m , which is typically much smaller than the

dimension of the expanding matrix. The selection of m follows the formula proposed in Chapter 6.

The comparison of the Newmark method, the PTI algorithm by step-response, impulsive-response matrices and Duhamel-response matrix with dimensional expansion (A6), and the Krylov precise time-step integration algorithm with dimensional expansion (A10) for computing Example 7.1 is given in Table 7-1. It shows that the method (A6) is more efficient than the Newmark method when the scale of system is not very large. The high precise solutions can be obtained by the proposed PTI methods (A6 and A10) using large time step, even one time step iterative. The time step size in the Newmark method should be selected carefully if high accuracy is required. With system dimension increasing, the method (A10) can reduce more computational cost.

7.3 Dynamic Analysis of the Structures under Moving Loads

The moving loads problem is always an important dynamic response problem in practical engineering field. The current precise time-step integration algorithms can be extended to this field. Zhu and Law (2001) compared the precise time-step integration algorithm with Newmark method for solving a problem of a uniform continuous beam under a system of moving loads. They observed that the precise time-step integration method is more efficient than the Newmark method. The modified precise time-step integration methods in this research have been investigated and it is shown that they are much better than the original precise time-step integration method in previous chapters. These methods can be used to solve the dynamic response of a uniform or non-uniform continuous beam under the action of moving loads. The improvement of current precise time-step integration methods can be demonstrated again in solving the moving load system. The modified precise time integration method can be applied to obtain the response time histories in which more economical computational cost could be achieved

with the same time step of integration, compared with the computational cost of original precise time integration method.

In fact, the moving load problems can be divided into four kinds of problems, namely, moving force, moving mass, moving sprung mass and moving vehicles problems, as shown in Figure 7-1. Without loss of generality, a simply supported beam under the action of a moving vehicle in Figure 7-2 is considered. The vehicle is assumed to be composed of two parts. The upper or non-contact part consists of the car body, suspension systems and bogies, which has in total l DOFs, as indicated by the vector $\{\mathbf{d}_u\} = \{d_{u1} \ d_{u2} \ \dots \ d_{ul}\}$. The wheel or contact part consists of n wheelsets. Assuming that each wheelset is represented by one vertical DOF, the wheel part can be denoted as $\{\mathbf{d}_w\} = \{v_{w1} \ v_{w2} \ \dots \ v_{wn}\}^T$, where v_{wi} denotes the displacement of the i -th wheel. Correspondingly, there are n contact points on the bridge, which may be denoted as $\{\mathbf{d}_c\} = \{v_{c1} \ v_{c2} \ \dots \ v_{cn}\}^T$, where the v_{ci} denotes the displacement of the bridge at the i -th contact point.

Let $[\mathbf{m}_v]$, $[\mathbf{c}_v]$ and $[\mathbf{k}_v]$ denote the mass, damping and stiffness matrices of the vehicle, respectively. $\{\mathbf{d}_v\}$ is the displacement vector of the vehicle, i.e., $\{\mathbf{d}_v\} = \{\{\mathbf{d}_u\}^T \ \{\mathbf{d}_w\}^T\}^T$. The equation of motion for the vehicle is given by

$$[\mathbf{m}_v]\{\ddot{\mathbf{d}}_v\} + [\mathbf{c}_v]\{\dot{\mathbf{d}}_v\} + [\mathbf{k}_v]\{\mathbf{d}_v\} = \{\mathbf{p}_v\} + \{\mathbf{f}_b\} \quad (7-16)$$

Where,

$$[\mathbf{m}_v] = \begin{bmatrix} \mathbf{m}_{uu} & \mathbf{m}_{uw} \\ \mathbf{m}_{wu} & \mathbf{m}_{ww} \end{bmatrix}, [\mathbf{c}_v] = \begin{bmatrix} \mathbf{c}_{uu} & \mathbf{c}_{uw} \\ \mathbf{c}_{wu} & \mathbf{c}_{ww} \end{bmatrix}, [\mathbf{k}_v] = \begin{bmatrix} \mathbf{k}_{uu} & \mathbf{k}_{uw} \\ \mathbf{k}_{wu} & \mathbf{k}_{ww} \end{bmatrix}, \{\mathbf{f}_b\} = \begin{Bmatrix} \mathbf{f}_c \\ \mathbf{0} \end{Bmatrix} \quad (7-17)$$

The vector $\{\mathbf{p}_v\} = \{\{\mathbf{p}_u\}^T \ \{\mathbf{p}_w\}^T\}^T$ is the external force components, $\{\mathbf{f}_c\}$ is the interaction force between the suspension unit and the bridge element in contact. Let x_{ci} denote the position of contract point, and $\{\mathbf{N}_c\}$ a vector containing cubic Hermitian interpolation functions for the vertical displacement of the beam evaluated at the contact

point x_{ci} , that is, $\{\mathbf{N}_c\} = \{\mathbf{N}(x_{ci})\}$. The equation of motion for the beam element can be written as

$$[\mathbf{m}_b]\{\mathbf{d}_b\} + [\mathbf{c}_b]\{\mathbf{d}_b\} + [\mathbf{k}_b]\{\mathbf{d}_b\} = \{\mathbf{p}_b\} - \{\mathbf{N}_c\}\{\mathbf{f}_c\} \quad (7-18)$$

Where $[\mathbf{m}_b]$, $[\mathbf{c}_b]$, and $[\mathbf{k}_b]$ denote the mass, damping, and stiffness matrices of the beam element, respectively. $\{\mathbf{p}_b\}$ is the external nodal forces. More details can be found in the literature (Yang and Wu, 2001).

In fact, Equations (7-16) and (7-18) can be incorporated into a normative computing formula of the precise time integration algorithm as

$$[\mathbf{M}]\{\ddot{\mathbf{u}}(t)\} + [\mathbf{C}]\{\dot{\mathbf{u}}(t)\} + [\mathbf{K}]\{\mathbf{u}(t)\} = \{\mathbf{r}(t)\} \quad (7-19)$$

Hence, the precise time-step integration methods proposed in this research can be employed to evaluate the dynamic response. A general moving force example is discussed in the following example.

Example 7.2: Moving Force Problem

The dynamic problem of a simply supported beam under a moving load is considered. The model is shown in Figure 7-3. The cross-section area and the material density of the beam are, respectively, $1.146 \times 10^{-3} \text{ m}^2$ and 7700 kg/m^3 . The moving force is $P=1000 \text{ Newton}$. Young's modulus is $2.07 \times 10^5 \text{ MPa}$. The speed of the moving load is 1.0 m/s . The beam length L is 1.0 m . The computation of the responses was done using the first 12 vibration modes. Frýba (1972) gave the exact solution as

$$V(x,t) = V_0 \sum_{j=1}^{\infty} \frac{1}{j^2 [j^2 (j^2 - x_1^2)^2 + 4x_1^2 x_2^2]} [j^2 (j^2 - x_1^2) \sin j\omega t - \frac{j\alpha [j^2 - x_1^2] - 2x_2^2}{(j^4 - x_2^2)^{1/2}} e^{-\omega_b t \sin \omega'_{(j)} t} - 2jx_1 x_2 (\cos j\omega t - e^{-\omega_b t} \cos \omega'_{(j)} t)] \sin \frac{j\pi x}{L} \quad (7-20)$$

where $\omega_{(j)}^2 = \frac{j^4 \pi^4 EJ}{L^4 \mu}$, and J is the constant moment of inertia of the beam cross section. Here, $J = 1.09443 \times 10^{-7}$. μ is the constant mass per unit length of the beam.

$$x_1 = \frac{cL}{\pi} \left(\frac{\mu}{EJ}\right)^{1/2}, x_2 = \frac{\omega_b L^2}{\pi^2} \left(\frac{\mu}{EJ}\right)^{1/2}, V_0 = \frac{PL^3}{48EJ} \quad (7-21)$$

where ω_b is the circular frequency of damping of the beam. For this example, from Equation (7-20) with $x_2 = 0$, and then:

$$V(x, t) = V_0 \sum_{j=1}^{\infty} \sin \frac{j\pi x}{L} \frac{1}{j^2(j^2 - x_1^2)} (\sin j\omega t - \frac{x_1}{j} \sin \omega_{(j)} t). \quad (7-22)$$

The example shows the accuracy of the precise time-step integration algorithm for solving the moving loads problems. The precise time-step integration methods can be used to solve the engineering dynamic problems easily because the computation process is simple and the accuracy is very high. This example only compared the method (A1) with Newmark method because all other precise time-step integration methods proposed in this research are more efficient and accurate than method (A1). Hence, the comparison of the accuracy of the method (A1) and the Newmark method are given here. More efficient results can be obtained by using the precise time-step integration algorithms instead of the method (A1) that has been investigated in precious chapters. Figures 7-5 and 7-6 show that the results of the precise time-step integration method (A1) closely match with the exact solution. The method (A1) is much better than the Newmark method when time step size is large. The time step size restriction of the Newmark method is obvious.

7.4 Analysis of Critical Wind Velocity of Galloping Oscillation of High-Rise Structures

High-rise buildings may produce across-instability by flow. The danger of galloping oscillation of a high-rise building is not negligible. The current proposed precise time-

step integration methods in this thesis can be used to analysis critical wind velocity of galloping oscillation of thigh-rise buildings. If the cross section of a flexural high-rise structures is shown in Figure 7-4. The angle of wind flow and horizontal axis is α_x . The crosswind vibration equation of the system can be written as

$$m(z)\frac{\partial^2 y}{\partial t^2} + c(z)\frac{\partial y}{\partial t} + k(z)y = F_y(z, \alpha_x) \quad (7-23)$$

Where $m(z)$, $c(z)$ and $k(z)$ are unit mass, damping and stiffness at structure height z , respectively. $F_y(z)$ is the galloping oscillation stress at structure height z .

$$F_y(z, \alpha_x) = \frac{1}{2} \rho \cdot v_x^2(z) B(z) C_y(\alpha_x) \quad (7-24)$$

where ρ is the air density; $v_x(z)$ is the wind velocity along x axis at height z ; $B(z)$ is the transverse character size of the structure at height z ; C_y is the y -axis wind stress coefficient (Emil and Robert, 1981)

$$C_y(\alpha_x) = \sum_{i=1}^n a_i \alpha_x^i = \sum_{i=1}^n a_i (v_x)^i \quad (7-25)$$

where a_i is the coefficient of the polynomial.

Using the model analysis method, $y(z, t)$ in Equation (7-23) can be written as

$$y(z, t) = \sum_{j=1}^{\infty} \phi_j(z) q_j(t) \quad (7-26)$$

Substituting Equation (7-26) into Equation (7-23), the first order model oscillation equation comes

$$\ddot{q}_1(t) + 2\zeta_1 \omega_1 \dot{q}_1(t) + \omega_1^2 q_1(t) = f_1 \quad (7-27)$$

where ζ_1 and ω_1 are y -axis damping ratio and natural frequency of the first order model, respectively;

$$f_1 = \frac{1}{M_1^*} \sum_{i=1}^n R_i \dot{q}_i'(t) \quad (7-28)$$

$$M_1^* = \int_0^h m(z)\phi_1^2(z)dz, R_i = \frac{1}{2} \rho \alpha_i \int_0^h B(z)\phi_1^{i+1} v_x^{2-i} dz, h \text{ is the height of structure.}$$

The excitation f_1 is a time-dependent variable. At every time step, the excitation should be reevaluated from the response of the last time step. According to the methods proposed in this research, Equation (7-27) can be solved accurately and efficiently.

Equation (7-27) can be rewritten as

$$\{\dot{\mathbf{U}}^*(t)\} = [\mathbf{W}^*]\{\mathbf{U}^*(t)\} \tag{7-29}$$

where

$$\{\mathbf{U}^*(t)\} = \begin{Bmatrix} q_1(t) \\ \dot{q}_1(t) \\ \mathbf{Z}(t) \end{Bmatrix}, [\mathbf{W}^*] = \begin{bmatrix} 0 & 1 & 0 \\ -\omega_1^2 & -2\xi_1\omega_1 & \mathbf{F} \\ 0 & 0 & \mathbf{S}_1 \end{bmatrix},$$

$$[\mathbf{S}_1] = \begin{bmatrix} 0 & 0 \\ 1 & 0 \end{bmatrix}, \{\mathbf{F}\} = \{f_1 \ 0\} \text{ and } \{\mathbf{Z}(t)\} = \{1 \ t\}^T$$

Hence, by using the precise time-step integration methods, the solutions will be

$$\{\mathbf{U}^*(t)\} = \exp([\mathbf{W}^*]\Delta t)\{\mathbf{U}^*(t - \Delta t)\} \tag{7-30}$$

Thus, with a given initial condition, $\{q_1(0) \ \dot{q}_1(0)\}^T = \{1.0 \ 0\}^T$, the computing procedure for analyzing of critical wind velocity of galloping oscillation of high-rise structures is listed as follows:

- (1) compute the dynamic characters of structure;
- (2) choose a critical wind velocity;
- (3) modify the excitation term $[\mathbf{F}]$ and compute Equation (7-30) by proposed precise time-step integration methods;
- (4) compute the attenuation ratio of swing $\delta q = \ln(a_n / a_{n+k})$, where a_n and a_{n+k} are swing and there are k periodic between them. If $\delta q > 0$, the structure oscillation is stable, enlarge the wind velocity, return to step (3); if $\delta q < 0$, the structure

oscillation is emanative, reduce the wind velocity, return to step (3); if $\delta q = 0$, the structure oscillation is galloping oscillation, stop computation.

Using the precise time-step integration methods proposed in this research, the critical wind velocity of galloping oscillation of high-rise structures and the system response of whole process can be obtained.

7.5 Summary

The application of the proposed precise time-step integration algorithms in this research are briefly summarized here. This chapter presents some simple application of these precise time-step integration algorithms. It demonstrates the implementation of the proposed algorithms for solving the practical engineering problems. Dynamic analysis of the structures under moving load is very sensitive. The precise time-step integration algorithms proposed in this research can be extended widely to other relevant fields.

Table 7-1 Comparison of Newmark method, method A6 and method A10

Dimension of System (N_s)	Time step Δt (s)	Newmark method			Method A6			Method A10			
		Result u_{N_s}	Time cost (s)	Error	Result u_{N_s}	Time cost (s)	Error	Result u_{N_s}	Time cost (s)	Error	m
50	10.000	98.16964123	0.010	10^1	72.65435742	0.691	10^{-7}	72.65435734	0.270	10^{-7}	27
	1.000	72.84725567	0.020	10^{-1}							
	0.100	72.65584875	0.170	10^{-3}							
	0.010	72.65437229	1.101	10^{-5}							
	0.001	72.65435769	6.689	10^{-7}							
100	10.000	97.72552284	0.100	10^1	72.20140767	5.648	10^{-7}	72.2014076	1.121	10^{-7}	27
	1.000	72.43310780	0.130	10^{-1}							
	0.100	72.20413547	0.300	10^{-3}							
	0.010	72.20143501	2.343	10^{-5}							
	0.001	72.20140806	20.910	10^{-7}							
200	10.000	98.16977791	0.951	10^1	72.65435742	81.897	10^{-7}	72.65435737	4.517	10^{-7}	28
	1.000	72.84725567	1.051	10^{-1}							
	0.100	72.65584875	1.853	10^{-3}							
	0.010	72.65437229	9.073	10^{-5}							
	0.001	72.65435769	80.646	10^{-7}							
300	10.000	110.7877554	3.445	10^1	84.53798407	261.200	10^{-7}	84.53798410	10.786	10^{-7}	28
	1.000	84.88054262	3.585	10^{-1}							
	0.100	84.54121179	5.247	10^{-3}							
	0.010	84.53801629	21.080	10^{-4}							
	0.001	84.53798453	177.815	10^{-7}							

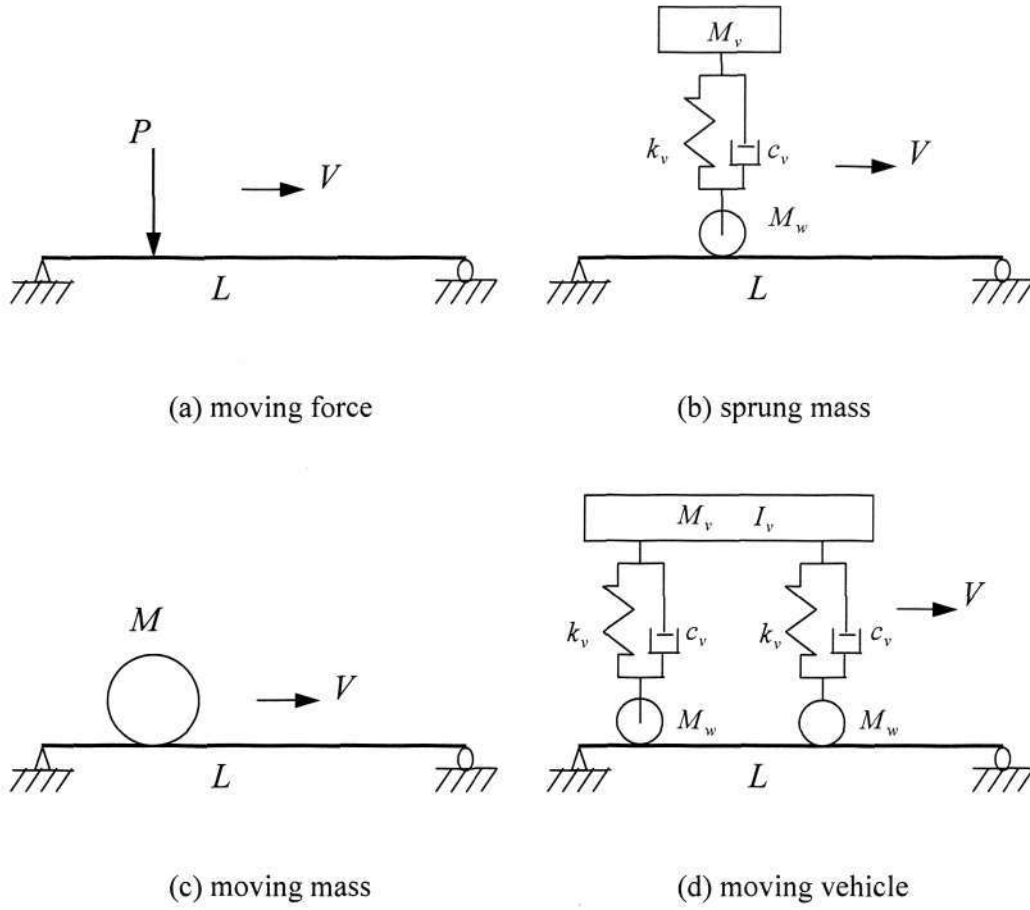


Figure 7-1 A simple supported beam under different moving loads

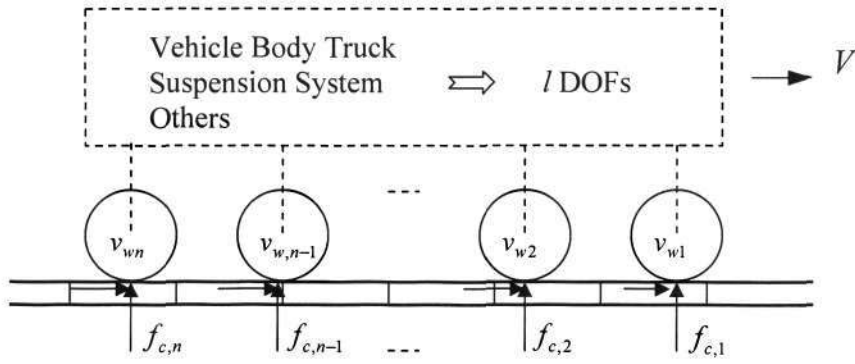


Figure 7-2 Schematic vehicle-bridge model

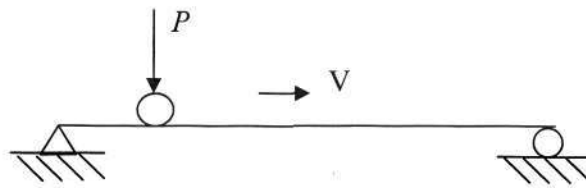


Figure 7-3 A simple supported beam under moving mass

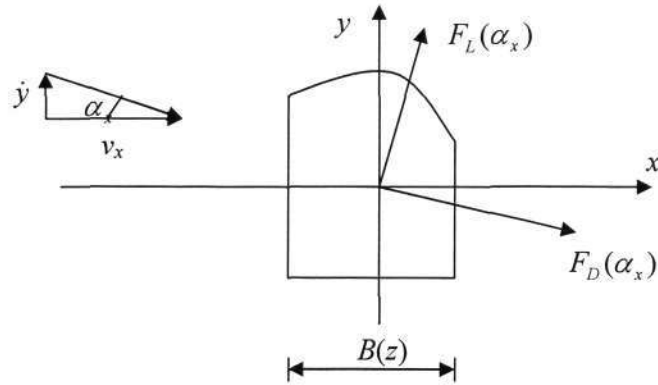


Figure 7-4 Schematic of wind loads on structure cross sections where $F_D(\alpha_x)$ is downwind stress and $F_L(\alpha_x)$ is crosswind stress

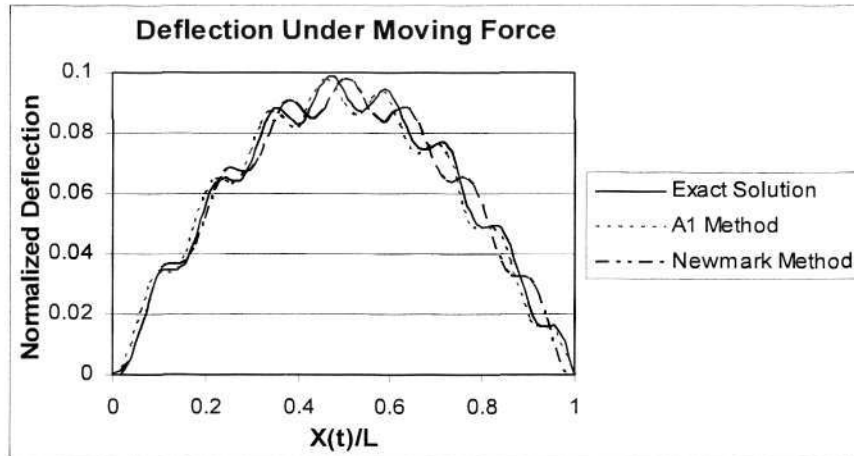


Figure 7-5 Deflection of beam at mid-span under the moving load, $\Delta t = 2.0 \times 10^{-2} s$

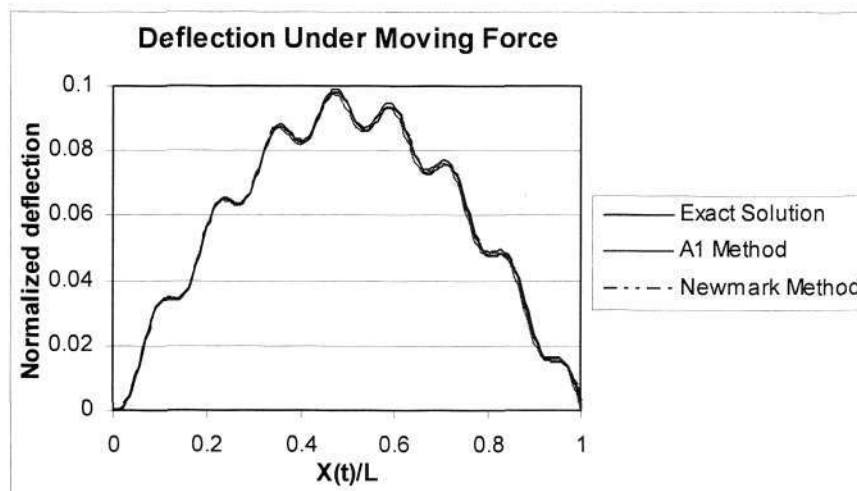


Figure 7-6 Deflection of beam at mid-span under the moving load $\Delta t = 5.0 \times 10^{-3} s$

CHAPTER EIGHT

Conclusions and Recommendations

8.1 Conclusions

In this research, the precise time-step integration algorithm is further developed to solve dynamic problems efficiently. The accuracy and efficiency of the current precise time-step integration algorithms have been improved.

Three kinds of methods, i.e. Padé series approximation, dimensional expanding method and Krylov subspace method, are applied to improve the stability, accuracy and efficiency of the precise time-step integration methods. Unconditionally stable time-step integration algorithms are obtained by using newly derived Padé approximations instead of the ordinary Taylor approximations. The extended dimensional expanding method is further employed to improve the current precise time-step integration algorithms. The Krylov precise time-step integration method is proposed for solving the large-scale dynamic systems and non-linear problems.

The derived time integration algorithms in this research are different from the conventional direct explicit time integration methods, such as second order central difference method, Newmark method and so on. The second-order governing differential equations can be transformed into the first-order equations or solved directly using the proposed time-step integration algorithms. The methods discussed in the thesis are denoted as symbols **A1** to **A10**. The details of the ten symbols are given as:

A1 method	The original PTI method (Zhong and Williams, 1994)
A2 method	The PTI method combined with the dimensional expanding method (Gu et al., 2001)
A3 method	The PTI method combined with the dimensional expanding method (Wang et al., 2002)
A4 method	The PTI algorithm by step-response and impulsive-response matrices (Fung, 1997)
A5 method	The PTI algorithm by step-response and impulsive-response matrices with dimensional expansion
A6 method	The PTI algorithm by step-response, impulsive-response matrices and Duhamel-response matrix with dimensional expansion
A7 method	The PTI method with Padé series approximation
A8 method	The PTI method by step-response and impulsive-response matrices with the Padé approximation
A9 method	The Krylov precise time-step integration algorithm with Padé approximation
A10 method	The Krylov precise time-step integration algorithm with dimensional expansion and Padé approximation

The major work of this thesis is listed below:

- (1) The time-step integration methods are established as general numerical methods for solving dynamic response. The efficiency and accuracy of the precise time-step integration algorithms are further investigated.
- (2) The precise time-step integration methods are improved by using the dimensional expanding method. This method can transform the non-homogenous equations into homogeneous equations. Some improvements have been made in this research to improve the computing efficiency and accuracy. With the improved dimensional expanding method, the precise time-step methods for solving the first-order and second-order equations directly have been proposed. The new precise time-step integration algorithms with the extended dimensional expanding method for first-order and second-order equation are presented in this thesis. The improved algorithms exhibit desirable characteristics.
- (3) A new precise time-step integration method by step-response, impulsive-response and Duhamel-response matrices has been proposed for solving the second-order dynamic problems directly. The steady state responses for various types of excitations can be readily obtained. The improvement of the precise time-step integration method by step-response and impulsive-response matrices has been achieved to enhance the efficiency. To further improve efficiency of the precise time-step integration methods, the particular solutions derived by Duhamel integral are evaluated by the Duhamel-response matrix. The relation of the Duhamel-response matrix and its time derivatives has been studied. In addition, the step-response and impulse-response matrices can be conveniently expressed in terms of two symmetric matrices to improve the efficiency.

- (4) Considering the approximation to the exponential matrix in the precise time-step integration process, it is proven more accurate with Taylor series approximation being replaced by Padé series approximation. From the approximation to the exponential function, a better approximation is likely to yield more accurate solutions. The ordinary Padé series approximations are used to improve the accuracy of the precise time-step integration algorithms.
- (5) The Padé approximation is also employed to improve the stability of the precise time-step integration algorithm by step-response and impulsive-response matrices. An unconditionally stable precise time-step integration algorithm by a new Padé approximation is proposed in this research. The first four-order parameters of the new Padé approximation in the unconditionally stable algorithm are given explicitly.
- (6) The computational cost can be reduced significantly by employing the Krylov subspace method. It is found that the optimum value of N (scale and squaring factor) can easily be chosen and used in the Krylov precise time-step integration method. Furthermore, the formulae for choosing the efficient range of m (the dimension of upper Hessenberg Matrix) are given. Hence, the Krylov subspace method is easily applied to improve the efficiency of the precise time integration method for solving large-scale systems. By using the Krylov subspace method, a more effective and simpler algorithm can be derived. The performance of the derived algorithms is found to be excellent. Tremendous computational cost can be saved by using this method. This method is also extended to solve the nonlinear problems.

8.2 Recommendations

Based on the present research, further research work can be carried out and more fruitful results could be expected. To gain a complete understanding and application of time-step integration algorithm to moving loads and other dynamic problems, further research should be carried out in the following subjects.

(1) Implementation in Transient Heat Conduction Analysis

In general, the heat conduction is non-linear problem and its nonlinearities usually include the temperature dependent material properties and the radiation boundary condition. The transient non-linear equations are

$$[\mathbf{C}(\mathbf{q})]\{\dot{\mathbf{q}}\} + [\mathbf{K}(\mathbf{q})]\{\mathbf{q}\} = \{\mathbf{R}_q\} \quad (8-1)$$

where the matrices symbolic meanings are the same as before, except they are not constants but depended on temperature. As for finite element methods, the relative matrices $[\mathbf{C}(\mathbf{q})]$, $[\mathbf{K}(\mathbf{q})]$ and $\{\mathbf{R}_q\}$ are functions of temperature.

The original precise time-step integration algorithm needs some transformations and assumptions so that the dynamic response can be obtained (Chen et. al., 2001). The transformations and assumptions will increase the additional truncation error and the computational efforts. But the method (A10) can solve Equation (8-1) accurately and efficiently. The additional computational cost is to form the computing matrix at each time step when the method (A10) is employed. More further research work should be studied to obtain the accurate and reasonable assumption of the matrices $[\mathbf{C}(\mathbf{q})]$ and $[\mathbf{K}(\mathbf{q})]$.

(2) Further study of the proposed precise time-step integration methods

The improved precise time-step integration algorithms in this research are explicit direct time-step integration method for first order and second order

equations. The current improved precise time-step integration algorithms can be extended to solve higher order differential equations directly.

(3) Unconditionally stable algorithms

In this research, an unconditionally stable precise time-step integration algorithm is proposed. The other conventional precise time-step integration algorithms are conditionally stable. More efficient unconditionally stable algorithms could be generated based on the current precise time-step integration algorithms.

(4) Error estimation

Error estimation can help to further control the accuracy of the current precise time-step integration methods, which can make the algorithms more efficient. More great achievements can be expected in this field.

References

Argyris J. H, Vaz L. E and Willam K. J. (1997), "Higher order methods for transient diffusion analysis", Computational Methods in Applied Mechanics and Engineering, Vol. 12: 243-278

Arnoldi W. E. (1951), "The principle of minimized iteration in the solution of the matrix eigenvalue problem", Quarterly of Applied Mathematics, Vol. 9: 17~29

Ashby S. F., Manteuffel T.A. and Saylor P. E. (1990), "A taxonomy for conjugate gradient methods", SIAM Journal of Numerical Analysis, Vol. 27: 1542~1568

Axelsson O. (1977), "Solution of linear systems of equations: iterative methods", In V.A. Barker, editor, Sparse Matrix Techniques

Bai Z. J. (2002), "Krylov subspace techniques for reduced-order modeling of large-scale dynamical systems". Applied Numerical Mathematics, Vol. 43: 9~44

Baker G. A. (1975), Essentials of Padé approximants, New York: Academic press

Baker G. A. and Graves-Morris P. (1996), Padé approximants, 2nd edition, Cambridge: University Press

Bathe K. J. (1982), Finite Element Procedures in Engineering Analysis, Prentice-Hall, Englewood Cliffs, N.J.

Belytschko T. and Hesieh B. J. (1973), "Nonlinear transient finite element analysis with convected coordinates", International Journal of numerical mechanic engineering, Vol. 7: 255~271

Belytschko T. and Lu Y.Y. (1993), "Explicit multi-time step integration for first and second order finite element semidiscretizations", Computer methods in applied mechanics and engineering, Vol. 108: 335~383

Belytschko T. and Mullen R. (1976), "Mesh partitions of explicit-implicit time integrators", In formulations and Computational algorithms in finite element analysis(Edited by K.J.Bathe, J.T.Oden and W.Wunderlich), 673~690, MIT Press, Cambridge

Bergamaschi L, Vianello M. (2000), "Efficient computation of the exponential operator for large, sparse, symmetric matrices". Numerical Linear Algebra with Applications, Vol. 7: 27~45

Bickart T.A. (1968), "Matrix exponential: Approximation by truncated power series", Proc. IEEE, Vol. 56: 372~373

Braekhus J. and Aasen J.O. (1981), "Experiments with direct integration algorithms for ordinary differential equations in structural dynamics", Computers and Structures, Vol. 13: 95~96

Cai ZQ, Gu YX, Zhong WX., (2001), "A new approach of computing Floquet transition matrix". Computers and Structures, Vol. 79: 631~635

Chen Biaosong, Gu Yuanxian, Zhang Hongwu, Guan Zhenqun, (2001), "Non-linear transient heat conduction analysis with precise time integration method", www.paper.edu.cn, (中国论文网), In: Numerical Heat Transfer, part B, Vol. 40(4): 325~341

Clough R. W. and O'Kellyk M. E. J. (1993), Dynamics of Structures, 2nd, edn., McGraw-Hill, New York,

- Dahlquist G. and Björck Å. (1974), Numerical method, Prentice-Hall, Englewood
- Dokainish M.A. and Subbaraj K. (1989), “A survey of direct time-integration methods in computational structural dynamics-I. Explicit methods”, Computers and Structures Vol. 32(6): 1371~1386
- Druskin V. and Moskow S. (1991), “three-point finite difference schemes, Padé and the spectral Galerkin method 1. one-sided impedance approximation”. Mathematics of computation
- Ekevid T, Martin X.D.L and Wiberg N.E., (2001), “Adaptive FEA of wave propagation induced by high-speed trains”, Computers and Structures, Vol. 79: 2693~2704
- Elman H. C., (1982), “Iterative methods for large, sparse, nonsymmetric systems of linear equations”, PhD thesis and Res. Rep., Dept. of Comp. Sci., Yale U
- Emil Simiu and Robert H.S. (1981), Wind effects on structures [M], A Wiley-Interscience Publication
- Fehlberg E., (1970), “klassische Runge-Kutta formula vierter und niedriger ordnung”, Computing, Vol. 6: 61~67
- Fletcher R. (1975), “Conjugate gradient methods for indefinite systems”, In G. Watson, ed., Proc. Of the Dundee Biennial Conference on Numerical Analysis, (Springer-Verlag, New york)
- Freund R. W. and Nachtigal N. M. (1991), “QMR: aquasi-minimal residual method for non-Hermitian linear system”, Numerical Mathematic, Vol. 60: 315~339
- Fryba L. (1972), Vibration of solids and structures under moving loads, 3rd edition

Fung T.C. (1997), "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: 4501~4527

Gallopoulos E. and Saad Y., (1992), "Efficient solution of Parabolic equations by Krylov approximation methods", SIAM Journal of Scientific and Statistic. Computing., Vol. 13: 1236~1264

Goudreau G.L. and Taylor R.L. (1972), "Evaluation of numerical integration methods in elastodynamics", Computer Methods in Applied Mechanics Engineering , 69~97

Gu Y.X., Chen B.S., Zhang H.W., and Guan Z.Q. (2001), "Precise Time-integration method with dimensional expanding for structural dynamic equations", AIAA Journal, Vol. 39(12): 2394~2399

Henrici P. (1968), Discrete Variable Methods in Ordinary Differential equations, John Wiley, NY.

Hestenes M. R and Stiefel E., (1954), Methods of conjugate gradients for solving linear systems, J. Res. Nat. Bur. Stand., Vol. 49: 409~436

Hochbruck M. and Lubich C, (1997), "On Krylov subspace approximations to the matrix exponential operator", SIAM Journal on Numerical Analysis. Vol. 34(5): 1911~1925

Hughes T. J. R., (1977), "A note on the stability of Newmark's method for numerical integration in structural dynamics", International Journal of Numerical Mathematical in Engineering, Vol. 11: 383~386

Hughes T. J. R., (1987), The Finite element Method: Linear static and Dynamic Finite elements Analysis, Prentice-Hall, Englewood Cliffs, N.J.

Hughes T.J.R. and Liu W.K. (1978a), "Implicit-explicit finite elements in transient analysis: Implementation and numerical examples", Journal of Application in Mechanic, Vol. 45: 375~378

Hughes T.J.R. and Liu W.K. (1978b), "Implicit-explicit finite elements intransient analysis: Stability theory", Journal of Application in Mechanic, Vol. 45:371~374

Hull T.E., Enright W.H., Fellen B.M. and Sedwich A.E. (1972), "Comparing numerical methods for ordinary differential equations", SIAM Journal Numerical. Analysis, Vol. 9: 603~637

Humar J.L. and Wright E.W. (1974), "Numerical methods in structural dynamics", Can. J. civ. Engineering Vol. 1: 179~193

Jea K. and Young D., (1980), "Generalized conjugate-gradient acceleration of nonsymmetrizable iterative methods", Linear Algebra and it's Applications, Vol. 34: 159~194

Jensen P.S. (1974), "Transient analysis of structures by stiffly stable methods", Computers and Structures, Vol. 4: 615~626

Jensen P.S. (1976), "Stiffly stable methods for undamped second order equations of motion", Journal of Numerical Analysis, Vol. 13: 5549~563

Krieg R.D., (1976), "Unconditional stability in numerical time integration methods", Journal of applied mechanics, Vol. 40: 417~420

Lanczos C., (1950), "An iteration method for the solution of the eigenvalue problem of linear differential and integral operators", J. Res. Nat. Bur. Stand., Vol. 45: 255~282

Leung A. Y. T. (1993), Dynamic stiffness and Substructures, Springer, London

Leung, A. Y. T. (1985), "Structural response to exponentially varying harmonic excitations", Earthquake Engineering and structure dynamic, Vol. 13: 667~681

Leung, A. Y. T. (1986), "Steady state response of undamped systems to excitations expressed as polynomials in time", Journal of Sound and Vibration, Vol. 106: 145~151

Leung A. Y. T. (1988), "Direct Method for the Steady State Response of Structures", Journal of Sound and Vibration, Vol. 124(1): 135~139

Li XK, Wu WH, Zienkiewicz OC., (2000), "Implicit characteristic Galerkin method for convection-diffusion equations". International Journal for Numerical Methods in Engineering; Vol. 47: 1689~1708

Li yuanyin, Jin xianlong, Li lijun and Guo yizhi. (2005), "Application of Precise time integration method for dynamic response analysis of large-scale structures", Nongye Jixie Xuebao Vol. 36(8): 98~102

Lin J.H., Shen W.P. and Williamms F.W., (1995), "A high precision direct integration scheme for structures subjected to transient dynamic loading", Computers and structures, Vol. 56(1): 113~120

Liu jinhua, Fan yiqun and Zhong wanxie, (2004). "Precise integration finite element algorithm of viscoelastic solid", Chinese Journal of Computational Mechanics, Vol. 21(1): 109~114

Liou M.L., (1967), "A novel method of evaluating transient response", Proc. IEEE, 54

Lubinsky H.D.S (1997), "The Convergence of Padé Approximants to Functions with Branch Points", Journal of Approximation Theory, Volume 91 (2): 139~204

Melosh R.J. (1975), "integration of linear equations of motion", Journal of structure Division, ASCE Vol. 101: 1551~1558

Mizukami A. (1986), "Variable explicit finite element methods for unsteady heat conduction equations". Computer Methmatic Applied Mechanic Engineering 59, pp.101~109

Moler C. and Loan C.V., (1978), "Nineteen dubious ways to compute the exponential of a matrix", Society for industrial and applied mathematics (SIAM), Vol. 20(4)

Neal M.O. and Belytschko T. (1989), "Explicit-explicit subcycling with non-integer time step ratios for structural dynamic systems", Computers & structures, Vol. 31(6): 871~880

Nelson R.B. and Mak R., (1982), "A method for improving numerical stability of implicit time integration for nonlinear dynamic structural response". Nucl. Engng Design Vol. 70: 37~43

Newmark N.M. (1959), "A method of computation for structural dynamics", Journal of Engineering Mechanic division, ASCE Vol. 85: 67~94

Noor A.K. and Lambiotte J.J. (1979), Finite element dynamics analysis on CDC star-100 computer, Computers and Structures, Vol. 10: 7~9

Pan G., Atluri S.N. (1995), "Dynamic response of finite sized elastic runways subjected to moving loads: a coupled BEM/FEM approach", International Journal Numerical Methods Engineering, Vol. 38: 3143~3166

Park K.C., (1975), "An improved stiffly stable method for direct integration of nonlinear structural dynamic equations". Journal of Applied Mechanics. Vol. 42: 464~470

Park K.C., Felippa C.A. and Deruntz J.A. (1977), "stabilization of staggered solution procedures for fluid structure integration analysis". Computational Methods for fluid-structure interaction problems, 95~124, ASME, Applied mechanics symposia series, AMD-26

Pesterev A.V. and Bergman L.A. (1997a), "Vibration of elastic continuum carrying moving linear oscillator", ASCE Journal of Engineering mechanics, Vol. 123: 878~884

Pesterev A.V. and Bergman L.A. (1997b), "Vibration of elastic continuum carrying acceleration oscillator," ASCE, Journal of engineering mechanics, Vol. 123, : 886~889

Pesterev A.V. and Bergman L.A. (1998a), "response of a nonconservative continuous system to a moving concentrated load," ASME, Journal of applied mechanics, Vol. 65: 436~444

Pesterev A.V. and Bergman L.A. (1998b), "A contributed to the moving mass problem," ASME, Journal of vibration and acoustics, Vol. 120: 824~826

Pesterev A.V. and Bergman L.A. (2000), "An improved series expansion of the solution to the moving oscillator problem", Journal of Vibration and acoustics, January, Vol. 122: 54~61

Pesterev A.V., Tan C.A. and Bergman L.A. (2001a), "A new method for calculating bending moment and shear force in moving load problems", Journal of applied mechanics, March, Vol. 68: 252~259

Pesterev A.V., Yang B. and Bergman L.A. (2001b), "Response of elastic continuum carrying multiple moving oscillators", Journal of Engineering Mechanics, March, 260~265

Rasmussen K.M., Nielsen S.R.K., and Kirkegaard P.H. (2001), "Boundary element method solution in the time domain for a moving time-dependent force", Computers and structures, Vol. 79: 691~701

Reid J. K., (1971), "On the method of conjugategradients for the solution of large sparse systems of linear equations", Proc. Conf. On large sparse sets of linear equations, Academic Press, New York

Roland W. Freund, (2000), "Krylov-subspace methods for reduced-order modeling in circuit simulation", Journal of Computational and Applied Mathematics, Vol. 123: 395~421

Saad Y. (1981), "Krylov subspace methods for solving large unsymmetrical linear systems", Mathematic and Computers, Vol. 37: 105~126

Saad Y. (1982), "Projection methods for solving large sparse eigenvalue problems", Matrix pencils, proceedings, InB. Kagstrom and A.Ruhe, editors, Pitea Havsbad, 121~144

Saad Y. and M. H. Schultz, (1986), "GMRES: A generalized minimal residual algorithm for solving nonsymmetric linear systems", SIAM Journal of Scientific and Statistical Computing, Vol. 7(3): 856~869

Shampine L.F. (1977), "Local error controls in codes for ordinary differential equations", Application of Mathematic and Computation, Vol. 3: 189~210

Shen W.P., Lin J.H., and Williams F.W. (1995), "Parallel computing for the high precision direct integration method", Computer Methods Application in Mechanic Engineering, Vol. 126: 315~331

Smolinski P., Wu Y.S. (1998), "Stability of explicit subcycling time integration with linear interpolation for first-order finite element semidiscretizations", Computer Methods in Applied Mechanics and Engineering, Vol. 151: 311~324

Sonneveld P., (1989), "CGS, a fast Lanczos-type solver for nonsymmetric linear systems", SIAM Journal Science Statistic Computation, Vol. 10: 36~52

Subbaraj K. and Dokainish M.A. (1989), "A survey of direct time-integration methods in computational structural dynamics- II. Implicit methods", Computers and Structures, Vol. 32(6): 1378~1401

Wang Y.X., Tian X.D., and Zhou G. (2002), "Homogenized high precise direct integration scheme and its applications in engineering", Communications in numerical methods in engineering, Vol. 18: 429~439

Warburton G. B., (1985), "Some recent advances in structural vibrations of engineering structures", (Edited by L.A.Brebbia and S.A.Orszag), Lecture notes in engineering, No.10, Dpringer, Berlin

Wilson E.L, Farhoomand I. and Bathe K.J. (1973), "Nonlinear dynamic analysis of complex structures", Earthquake Engineering and Structural dynamics, Vol. 1: 241~252

Wilson. E.L., (1962), "Dynamic response by step-by-step matrix analysis", proceedings, symposium on the use of Computers in Civil Engineering, Labortorio Nacional de Engenharia Civil. Lisbon, Portugal, October 1-5

Wood W.L., (1977), "On the Zienkiewicz four-time level scheme for the numerical integration of vibration problems", International Journal of Numerical Mechanic engineering, Vol. 11: 1519~1528

Wragg A. and Davies C., (1975), Computation of the exponential of a matrix II: Practical considerations, Ibid, Vol. 15: 273~278

Wu J.J., Whitaker A.R., and Cartnell M.P. (2001), “Dynamic responses of structures to moving bodies using combined finite element and analytical methods”, International journal of Mechanical Sciences, Vol. 43: 2555~2579

Wu jian and Jin feng, (2004). “Precise direct integration method for random vibration analysis of arch dams”, Journal of Tsinghua University (Sci & Tech), Vol. 44(9): 1227~1230

Yang B. (1996, a), “Closed-form transient response of distributed damped systems, part1: Modal analysis and Green’s function formula”, Journal of Applied Mechanics, Vol. 63: 997~1003

Yang B. (1996, b), “Integral formulas for Non-Self-Adjoint distributed dynamic systems”, AIAA, Journal, Vol. 34(10): 2132~2139

Yang B., Tan C.A. and Bergman L.A. (2000), “Direct numerical problem for solution of moving oscillator problems”, Journal of engineering mechanics, 462~469

Yang Y.B. and Wu Y.S., (2001), “A versatile element for analyzing vehicle-bridge interaction response”, Engineering Structures, Vol. 23: 452~469

Young D. M and Jea K. C., (1980), “Generalized conjugate-gradient acceleration of nonsymmetrizable iterative methods”, Linear Algebra Applied, Vol. 34: 159~194

Zhang aishe, Zhang lin, Zhou jinxiong. (2004), “Analysis of critical wind velocity of galloping oscillation of high-rise structures by precise time integration method”, Engineering Mechanics, Vol. 21(1): 98~101

Zhang Hongwu, Zhang Xinwei and Chen Jiun-Shyan (2003), "A new algorithm for numerical solution of dynamic elastic-plastic hardening and softening problems", Computers and Structures Vol. 81: 1739~1749

Zhang min and Zhang xiaodan, (2004), "Precise integration method for solving wave equation and its numerical stability", Journal of University of Petroleum, Vol. 28(6): 129~132

Zhong W.X., and Williams F.W. (1994), "A precise time step integration method", Journal of Mechanical engineering Science, Vol. 208(6): 427~430

Zhong Wanxie, Zhu Jianing and Zhong Xiangxang. (1996), "On a new time integration method for solving time dependent partial differential equations", Computer Methods in Applied Mechanics and Engineering, Vol. 130: 163~178

Zhong WX. (2001), "Combined method for the solution of asymmetric Riccati differential equations". Computer Methods in Applied Mechanics and Engineering, Vol. 191: 93~102

Zhong WX. (2004), "On precise integration method", Journal of Computational and Applied Mathematics; Vol. 163: 59~78

Zhu X.Q. and Law S.S. (2001), "Precise time-step integration for the dynamic response of a continuous beam under moving loads", Journal of sound and vibration, Vol. 240(5): 962~970

Publications

1. T. C. Fung and Z. L. Chen, 2006, "Krylov Precise Time-step Integration Method", *International Journal for Numerical Methods in Engineering*, Volume 68, Issue 11, Date: 10 December 2006, Pages: 1115-1136
2. T. C. Fung and Z. L. Chen, 2006, "Precise Time-Step Integration Algorithms Using Response Matrices with Dimensional Expansion for Dynamic Problems", *AIAA Journal* (In press).