A NEW JACOBIAN MATRIX CALCULATION METHOD TO DECREASE COMPUTATIONAL TIME IN PERIODIC FORCED RESPONSE ANALYSIS OF NONLINEAR STRUCTURES

A THESIS SUBMITTED TO
THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES
OF
MIDDLE EAST TECHNICAL UNIVERSITY

BY

HAZIM SEFA KIZILAY

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR
THE DEGREE OF MASTER OF SCIENCE
IN
MECHANICAL ENGINEERING

AUGUST 2019

Approval of the thesis:

A NEW JACOBIAN MATRIX CALCULATION METHOD TO DECREASE COMPUTATIONAL TIME IN PERIODIC FORCED RESPONSE ANALYSIS OF NONLINEAR STRUCTURES

submitted by **HAZIM SEFA KIZILAY** in partial fulfillment of the requirements for the degree of **Master of Science in Mechanical Engineering Department, Middle East Technical University** by,

Prof. Dr. Halil Kalıpçılar
Dean, Graduate School of **Natural and Applied Sciences**

Prof. Dr. M. A. Sahir Arıkan
Head of Department, **Mechanical Engineering**

Prof. Dr. Ender Ciğeroğlu
Supervisor, **Mechanical Engineering, METU**

Examining Committee Members:

Assoc. Prof. Dr. Yiğit Yazıcıoğlu
Mechanical Engineering Dept., METU

Prof. Dr. Ender Ciğeroğlu
Mechanical Engineering, METU

Assoc. Prof. Dr. M. Bülent Özer
Mechanical Engineering Dept., METU

Assist. Prof. Dr. Gökhan Özgen
Mechanical Engineering Dept., METU

Assoc. Prof. Dr. Can U. Doğruer
Mechanical Engineering Dept., Hacettepe University

Date: 06.08.2019

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Name, Surname: Hazim Sefa Kızılay

Signature:

ABSTRACT

A NEW JACOBIAN MATRIX CALCULATION METHOD TO DECREASE COMPUTATIONAL TIME IN PERIODIC FORCED RESPONSE ANALYSIS OF NONLINEAR STRUCTURES

Kızılay, Hazim Sefa
Master of Science, Mechanical Engineering
Supervisor: Prof. Dr. Ender Ciğeroğlu

August 2019, 84 pages

The contact interfaces between the components used in high speed systems such as the turbo machinery cause nonlinear vibrations. In order to understand the dynamic characteristic of nonlinear systems, it is important to perform nonlinear vibration analysis. In nonlinear vibration analysis, due to the properties of nonlinear elements used, it is not possible to calculate the Jacobian matrix analytically or it becomes very complicated and difficult, therefore, Jacobian matrix is calculated as numerically. In each iteration, obtaining the Jacobian matrix by numeric methods greatly increases the overall calculation time. In this study, a new method is proposed for numerical Jacobian calculation of nonlinear vibration analysis of multidegree-of-freedom (MDOF) systems. The aim of the work is to obtain significant reduction of computational time for Jacobian calculation compared to classical Jacobian calculation. In order to reveal the effect of the suggested method, nonlinear equation set is derived by using receptance method. The nonlinear MDOF system is analyzed in frequency domain by using harmonic balance method (HBM) which makes nonlinear algebraic equations to be solved iteratively. The validation of the method is presented by comparing the computational times and computation reduction ratios obtained with classical Jacobian calculation and proposed Jacobian calculation method.

Keywords: Jacobian Calculation, Nonlinear Vibration, Harmonic Balance Method, Dry Friction

ÖZ

DOĞRUSAL OLMAYAN YAPILARIN PERIYODIK KUVVET CEVAP ANALIZI HESAPLAMA SURELERINI AZALTMAK ICIN YENI BIR JACOBIAN MATRIS HESAPLAMA YONTEMI

Kızılay, Hazim Sefa
Yüksek Lisans, Makina Mühendisliği
Tez Danışmanı: Prof. Dr. Ender Ciğeroğlu

Ağustos 2019, 84 sayfa

Turbo makineleri gibi yüksek hızlı sistemlerde kullanılan bileşenler arasındaki boşluk ve temaslı arayüzler, doğrusal olmayan titreşimlere neden olmaktadır. Doğrusal olmayan sistemlerin dinamik karakteristiğini anlamak için doğrusal olmayan titreşim analizlerini gerçekleştirmek önemlidir. Doğrusal olmayan titreşim analizinde, kullanılan doğrusal olmayan elemanların özelliklerinden dolayı, Jacobian matrisini analitik olarak hesaplamak mümkün olmadığı gibi bazı durumlarda ise çok karmaşık ve zor bir hale gelmektedir. Bu nedenle Jacobian matrisi sayısal olarak hesaplanmaktadır. Her yinelemede, Jacobian matrisini sayısal yöntemlerle elde etmek, toplam hesaplama süresini büyük ölçüde artırmaktadır. Bu tez çalışması kapsamında, doğrusal olmayan titreşim analizlerinde kullanılan sayısal Jacobian hesaplaması için yeni bir yöntem geliştirilmiştir. Çalışmanın amacı, klasik Jacobian hesaplamasına yöntemine kıyasla hesaplama süresinde önemli bir azalma elde etmektir. Önerilen yöntemin etkisini ortaya çıkarmak için, doğrusal olmayan denklem seti, receptance yöntemi kullanılarak türetilmiştir. Doğrusal olmayan çoklu serbestlik dereceli sistemi, frekans tabanında, doğrusal olmayan cebirsel denklemlerin yinelemeli olarak çözülmesini sağlayan harmonik denge yöntemi kullanılarak analiz edilir. Klasik Jacobian hesaplama yöntemi ve önerilen Jacobian hesaplama yöntemi ile elde edilen çözümlerin, hesaplama süreleri ve hesaplama azalım oranları karşılaştırılarak, yöntemin geçerliliği gösterilmiştir.

Anahtar Kelimeler: Jacobian Hesaplaması, Doğrusal Olmayan Titreşim, Harmonik Denge Metodu, Kuru Sürtünme

To My Life

ACKNOWLEDGEMENTS

First and foremost, I would like to express my sincere gratitude to my advisor Prof. Dr. Ender Ciğeroğlu for his invaluable guidance and continuous support to complete this study.

I would like to express my deepest gratitude to my parents Türkan and İsmail Kızılay for supporting me mentally throughout my life. I am also deeply grateful to my sisters Merve and Gamze Kızılay.

I had a very hard time between 2017 and 2018. I am grateful to my friends, Can Akın, Eren Öney, Gökhan Karapıstık who always support me in that period.

I would like to thank my dear friend, Gökhan Karapıstık for his technical suggestions and comments.

I would like to give special thanks to my mentor, Erkal Özbayramoğlu who helped me to find a new path in my life.

TABLE OF CONTENTS

ABSTRACT v

ÖZ vii

ACKNOWLEDGEMENTS x

TABLE OF CONTENTS xi

LIST OF TABLES xv

LIST OF FIGURES xviii

LIST OF ABBREVIATIONS xx

CHAPTERS

1. INTRODUCTION 1

1.1. Nonlinearity in Structures 1

1.2. Literature Survey 2

1.2.1. Set of Nonlinear Equation Solution Methods 2

1.2.1.1. Time Domain Methods 2

1.2.1.2. Frequency Domain Methods 4

1.3. Objective of the Thesis 10

1.4. Outline of the Thesis 10

2. PERIODIC RESPONSE ANALYSIS OF NONLINEAR STRUCTURES 13

2.1. Nonlinear Structures 13

2.2. Harmonic Balance Method 13

2.3. Nonlinearity Types 14

2.3.1. Cubic Stiffness 14

2.3.2. 1D-Dry Friction 16

2.4. Receptance Method 17
2.5. Modal Superposition Method 19
2.6. Solutions of Nonlinear Algebraic Equation 20
2.6.1. Newton's Solver on Receptance Method 20
2.6.2. Newton's Solver on Modal Super Position Method 21
2.6.3. Solver-1 22
2.6.4. Solver-2 22
2.6.5. Solver-3 23
2.6.6. Solver-4 23
2.6.7. Solver-5 23
2.6.8. Solver-6 24
2.6.9. Solver-7 24
2.7. Arc-length Continuation Method 25
2.8. Predictors for Path Following Methods 26
2.8.1. Tangent Predictor 26
2.9. Proposed Predictors 27
2.9.1. Linear Predictor 27
2.9.2. Second Order Polynomial Predictor 27
2.9.3. Third Order Polynomial Predictor 28
2.9.4. Logarithmic Predictor 29
2.9.5. Quadratic Spline Predictor 29
2.10. Classical Jacobian Matrix Calculation 31

3. A NEW JACOBIAN CALCULATION METHOD FOR PERIODIC FORCED RESPONSE ANALYSIS OF NONLINEAR STRUCTURES-JACOBIAN ELEMENT METHOD 33

3.1. Jacobian Element Method for Single Harmonic 33

3.2. Assembly Rule for Elemental Jacobian Matrices to Force Jacobian Matrix ..35

3.3. Application of Assembly Rule 36

3.4. Summary of the Proposed Method (JEM)...38

3.5. Theoretical Comparison of Computation Effort of Classical Jacobian and Jacobian Element Method 39

4. CASE STUDIES ON JACOBIAN ELEMENT METHOD 41

4.1. Three Different Lumped Models..41

4.2. Interpretation of the Worst Case Scenario for Lumped Parameter Model 45

4.3. Realistic Finite Element Model 46

4.4. Interpretation of the Worst Case Scenario for Realistic Finite Element Model with 1D-Dry Friction Nonlinearity..49

5. CASE STUDIES ON DIFFERENT NONLINEAR EQUATION SOLVERS AND PREDICTORS 51

5.1. Lumped Model for Different Nonlinear Solvers and Predictors 51

5.2. Realistic Finite Element Model for Different Nonlinear Solvers and Predictors 62

5.3. Detail Comparison of Different Solvers and Proposed Predictors for Both Cubic Stiffness and 1D-Dry Friction Nonlinearities 73

5.3.1. Effect of Error Criteria for Different Solvers 74

5.3.2. Effect of Adaptive Step Size for Different Solvers 75

5.3.3. Effect of Adaptive Step Size and Bypass for Different Solvers 76

5.3.4. Effect of Error Criteria for Different Predictors....................................... 77

5.3.5. Effect of Adaptive Step Size for Different Predictors.............................. 78

6. CONCLUSION 79

REFERENCES 81

LIST OF TABLES

TABLES

Table 2.1. *Specific calculation aspects of the solvers* 20

Table 4.1. *Computational time comparison of force Jacobian matrix with and without using Jacobian element method (JEM) for single harmonic solution of the lumped models* 44

Table 4.2. *Computational time comparison of force Jacobian matrix with and without using Jacobian element method (JEM) for two harmonics solution of the lumped models* 44

Table 4.3. *Computational time comparison of force Jacobian matrix with and without using Jacobian element method (JEM) for single harmonic solution of the bladed disk model* 49

Table 5.1. *Performance comparison of different algebraic nonlinear equation solvers with different constant arc-length parameters for cubic stiffness nonlinearity (Error Criterion=1e-6, 1e-8)* 53

Table 5.2. *Performance comparison of different algebraic nonlinear equation solvers with different constant arc-length parameters for cubic stiffness nonlinearity (Error Criterion=1e-10, 1e-12)* 54

Table 5.3. *Performance comparison of different algebraic nonlinear equation solvers with different constant arc-length parameters and passing unnecessary calculations for cubic stiffness nonlinearity (Error Criterion=1e-6, 1e-8)* 55

Table 5.4. *Performance comparison of different algebraic nonlinear equation solvers with different constant arc-length parameters and passing unnecessary calculations for cubic stiffness nonlinearity (Error Criterion=1e-10, 1e-12)* 56

Table 5.5. *Performance comparison of different algebraic nonlinear equation solvers with adaptive arc-length parameter for cubic stiffness nonlinearity (Error Criterion=1e-6, 1e-8)* 57

Table 5.6. *Performance comparison of different algebraic nonlinear equation solvers with adaptive arc-length parameter for cubic stiffness nonlinearity (Error Criterion=1e-10, 1e-12)* 57
Table 5.7. *Performance comparison of different algebraic nonlinear equation solvers with adaptive arc-length parameter and passing unnecessary calculations for cubic stiffness nonlinearity (Error Criterion=1e-6, 1e-8)* 58
Table 5.8. *Performance comparison of different algebraic nonlinear equation solvers with adaptive arc-length parameter and passing unnecessary calculations for cubic stiffness nonlinearity (Error Criterion=1e-10, 1e-12)* 58
Table 5.9. *Performance comparison of different predictors with different constant arc-length parameters for cubic stiffness nonlinearity (Error Criterion=1e-6, 1e-8)*..... 59
Table 5.10. *Performance comparison of different predictors with different constant arc-length parameters for cubic stiffness nonlinearity (Error Criterion=1e-10, 1e-12)* 60
Table 5.11. *Performance comparison of different predictors with adaptive arc-length parameter for cubic stiffness nonlinearity (Error Criterion=1e-6, 1e-8)* 61
Table 5.12. *Performance comparison of different predictors with adaptive arc-length parameter for cubic stiffness nonlinearity (Error Criterion=1e-10, 1e-12)* 61
Table 5.13. *Performance comparison of different algebraic nonlinear equation solvers with different constant arc-length parameters for 1D-Dry friction nonlinearity (Error Criterion=1e-6, 1e-8)* 64
Table 5.14. *Performance comparison of different algebraic nonlinear equation solvers with different constant arc-length parameters for 1D-Dry friction nonlinearity (Error Criterion=1e-10, 1e-12)* 65
Table 5.15. *Performance comparison of different algebraic nonlinear equation solvers with different constant arc-length parameters and passing unnecessary calculations for 1D-Dry friction nonlinearity (Error Criterion=1e-6, 1e-8)* 66
Table 5.16. *Performance comparison of different algebraic nonlinear equation solvers with different constant arc-length parameters and passing unnecessary calculations for 1D-Dry friction nonlinearity (Error Criterion=1e-10, 1e-12)* 67

Table 5.17. *Performance comparison of different algebraic nonlinear equation solvers with adaptive arc-length parameter for 1D-Dry friction nonlinearity (Error Criterion=1e-6, 1e-8)*..68
Table 5.18. *Performance comparison of different algebraic nonlinear equation solvers with adaptive arc-length parameter for 1D-Dry friction nonlinearity (Error Criterion=1e-10, 1e-12)*..68
Table 5.19. *Performance comparison of different algebraic nonlinear equation solvers with adaptive arc-length parameter and passing unnecessary calculations for 1D-Dry friction nonlinearity (Error Criterion=1e-6, 1e-8)* 69
Table 5.20. *Performance comparison of different algebraic nonlinear equation solvers with adaptive arc-length parameter and passing unnecessary calculations for 1D-Dry friction nonlinearity (Error Criterion=1e-10, 1e-12)* 69
Table 5.21. *Performance comparison of different predictors with different constant arc-length parameters for 1D-Dry friction nonlinearity (Error Criterion=1e-6, 1e-8*) 70
Table 5.22. *Performance comparison of different predictors with different constant arc-length parameters for 1D-Dry friction nonlinearity (Error Criterion=1e-10, 1e-12)* 71
Table 5.23. *Performance comparison of different predictors with adaptive arc-length parameter for 1D-Dry friction nonlinearity (Error Criterion=1e-6, 1e-8)* 72
Table 5.24. *Performance comparison of different predictors with adaptive arc-length parameter for 1D-Dry friction nonlinearity (Error Criterion=1e-10, 1e-12)* 72

LIST OF FIGURES

FIGURES

Figure 2.1. Force vs Displacement curve for cubic stiffness nonlinearity 15
Figure 2.2. Force vs Displacement curve for 1D-Dry friction nonlinearity 16
Figure 2.3. Quadratic spline explanation 30
Figure 3.1. 3-DOF lumped model 37
Figure 3.2. Summary of the proposed method 39
Figure 4.1. 4- DOF lumped model with ten cubic stiffness elements (Model-1) 42
Figure 4.2. 4- DOF lumped model with five cubic stiffness elements (Model-2) 42
Figure 4.3. 4- DOF lumped model with two cubic stiffness elements (Model-3) 42
Figure 4.4. Receptances of Model-1,2,3 for single harmonic solution 43
Figure 4.5. Receptances of Model-1,2,3 for two harmonics solution 43
Figure 4.6. Percentage reduction ratio in computational time of Jacobian calculation for the worst case scenario for lumped parameter model 45
Figure 4.7. Finite element model for the blade and Coulomb macroslip model between Shroud-1 and Shroud-2 47
Figure 4.8. Receptance of the system with four nonlinear friction elements for different sliding forces 48
Figure 4.9. Percentage reduction ratio in computational time of Jacobian calculation for contact nonlinearity case 50
Figure 5.1. 4- DOF lumped model with five cubic stiffness elements...................... 51
Figure 5.2. Receptance of the lumped model with five cubic stiffness elements 52
Figure 5.3. Realistic finite element model for the blade and Coulomb macroslip model between Shroud-1 and Shroud-2 62
Figure 5.4. Receptance of the system with four nonlinear friction elements 63
Figure 5.5. Performance comparison of different solvers for cubic stiffness nonlinearity according to different error criteria and constant step 74

Figure 5.6. Performance comparison of different solvers for contact nonlinearity according to different error criteria and constant step 74
Figure 5.7. Performance comparison of different solvers for cubic stiffness nonlinearity according to different adaptive step size parameters.............................75
Figure 5.8. Performance comparison of different solvers for contact nonlinearity according to different adaptive step size parameters 75
Figure 5.9. Performance comparison of different solvers for cubic stiffness nonlinearity according to different adaptive step size parameters and bypass 76
Figure 5.10. Performance comparison of different solvers for contact nonlinearity according to different adaptive step size parameters and bypass...............................76
Figure 5.11. Performance comparison of different predictors for cubic stiffness nonlinearity according to different error criteria and constant step 77
Figure 5.12. Performance comparison of different predictors for contact nonlinearity according to different error criteria and constant step 77
Figure 5.13. Performance comparison of different predictors for cubic stiffness nonlinearity according to different adaptive step size parameters.............................78
Figure 5.14. Performance comparison of different predictors for contact nonlinearity according to different adaptive step size parameters 78

LIST OF ABBREVIATIONS

CMS	Component Mode Synthesis
DFM	Describing Function Method
DOF	Degree of Freedom
FEM	Finite Element Model
FRF	Frequency Response Function
HBM	Harmonic Balance Method
IHBM	Incremental Harmonic Balance Method
IRM	Iterative Receptance Method
MDOF	Multi Degree of Freedom
MHBM	Multi-Harmonic Balance Method

Remove this page if it is not necessary

CHAPTER 1

INTRODUCTION

1.1. Nonlinearity in Structures

The aim of the modal analysis is to find the frequencies that the structure will increase the effect of a load and the shapes it will take at these frequencies. Here are some examples of why modal analysis is performed:

-to find dynamic behavior of structures with many DOFs

-to decide which rotation speeds are dangerous for the systems

-to understand which mode shapes and which operating frequency is important for structures.

In terms of modeling and solution simplicity, non-linear effects in systems are ignored most of the time, therefore modal analysis can be performed linearly. However, in systems where non-linear effects are dominant, it is not a correct approach to obtain modal information by using linear modal analysis.

In today's mechanical engineering, there are many systems where the effects of non-linear vibrations have to be considered. Some examples of these systems are listed as:

-friction contact nonlinearity between shrouds of gas turbine blades.

-friction contact nonlinearity at under platform dampers for turbine blades.

-gap contact nonlinearity between shrouds of gas turbine blades.

-friction contact nonlinearity at systems with bolted joints.

-velocity squared damping nonlinearity at tuned liquid column damper.

-material nonlinearity at viscoelastic vibration isolation systems.

-geometric nonlinearity at high level vibrating beams.

For those systems where nonlinearity changes the dynamic characteristics significantly, many problems are encountered in the process of obtaining reliable and accurate solutions. In the literature, many researchers study to develop new solution methods of the nonlinear vibratory systems in order to increase the accuracy and stability of the solution and reduce the computational effort. This fact shows how important and difficult it is to obtain solutions in the nonlinear systems. In the literature survey section, the solution methods of the nonlinear systems are discussed.

1.2. Literature Survey

1.2.1. Set of Nonlinear Equation Solution Methods

1.2.1.1. Time Domain Methods

Cameron and Griffin [1] proposed a new method to analyze dynamically nonlinear system in time domain. The main aim of their study is to obtain the steady-state response of the system under the harmonic excitation by using both frequency and time domain methods together. In the first step, the discrete Fourier transform of the system response is calculated iteratively. In the second step, returning to the time domain at each iteration ,nonlinear terms are calculated by using coefficient of the Fourier transform. After time domain iteration, to continue iteration on the steady state response in frequency domain, estimated nonlinear terms are converted to frequency domain. In order to validate their method, a nonlinear system with friction damper is studied. Results which are obtain by using proposed method are in good agreement with other method results. However, there are some main disadvantages of this method. Discrete Fourier transform can causes aliasing, leakage and roundoff errors therefore, it is possible to cause convergence problem in solution process.

Rezaiee et. al. [2] studied a new explicit higher order time integration algorithm to solve nonlinear dynamic system of equation of motion. In their study, state-space concept and the differential transform method theory is combined to solve the equation of motion. Linear and nonlinear system responses are studied to validate the new method. System responses obtained by new method are compared with analytical response and response calculated by Runke-Kutta method. However, as a result of their study, the differential transform method is a process to solve a differential equation in a Taylor series form iteratively which may causes to increase overall computation time.

T. Liu et. al. [3] proposed a new method derived from the Newmark method which uses the backward acceleration method and the trapezoidal rule. It is single step and second order accurate algorithm. The main advantage of this method is that in case of instability of solution process due to trapezoidal rule, this method provides to obtain stable solution in large deformation and long time analysis.

Balendra et al. [4] studied the effectiveness of passive TLCDs in reducing the wind-induced vibration of towers. Accordingly, authors used direct time integration method to solve the linearized nonlinear governing differential equations of motion. In their studies, optimum parameters were provided for a series of towers in relation to maximum reduction in acceleration and displacement under harmonic excitations. Those studies subsequently reveal that virtually the same level of reduction in acceleration is achievable for any tower of practical interest when a suitable opening ratio is used for the orifice in the TLCD connected rigidly to the structure.

Gao et al. [5] considered the same problem, where authors employed direct time integration, i.e. Newmark's constant average acceleration method, to solve the nonlinear differential equation of motion of the building coupled with a TLCD. Authors provided optimum TLCD parameters for a variety of flexible structures which reduced the peak structural response to harmonic excitation in a wide frequency range. Authors considered a variation of U- and V-shaped TLCD, which have different cross-sectional areas in vertical and horizontal sections.

Xie [6] studied on comparison of different time integration methods on nonlinear transient finite element analysis in terms of accuracy, stability and computational time. He showed whether Runge-Kutta, Newmark, Houbolt and Wilson-u method are suitable for transient nonlinear analysis. He found that the accuracy of the methods changes greatly depending on nonlinearity types in finite element model, method parameters and the time step sizes. In his study, the Newmark method, the Houbolt method and the Wilson-u method cause considerable numerical damping on solution for any application involving integration over a long time duration. He emphasized that Runge-Kutta method is more expensive than Newmark method for the same time step size in terms of computational time. Newmark method has less computational time than Houbolt method and Wilson-u method. He claimed that when the time step size is small than specific time step size which is given in the study, usage of Runge-Kutta method gives far more accurate results than other methods.

Literature survey about time domain methods for nonlinear dynamic systems is summarized. As a result, one can understand that time domain solution methods have some drawbacks in terms of stability, accuracy and computational time. Considering the optimization process of a large MDOF system, these drawbacks can cause waste of time.

1.2.1.2. Frequency Domain Methods

Menq and Griffin [7] proposed a new method for calculating the force response of structure with friction damper under by using simple finite element model. In their study, the steady state response is assumed as harmonic and nonlinear forcing is approximated by using the first term in a Fourier series expansion. In order to do this, receptance which is obtained by using results of finite element model without friction damper is utilized. Also, to verify the proposed method, they compare the results obtained by time integration method with the results obtained by new method. They show that there is good consistency between both results.

Menq et. al. [8] studied to calculate the peak vibratory response of a shrouded blade disk during increase or decrease of engine speed. Friction model that are used between shroud contact surfaces includes microslip and variable normal load variation. In order to decrease the total number of nonlinear equation, receptance method is utilized. Also, they express that the mode number that are used to obtain receptance effects the accuracy of the response.

Budak and Özgüven [9] suggested a new method for analyzing the harmonic response of multi degree of freedom system with nonlinearities such as spring and damper. Polynomial form of displacement and velocity responses are used to calculate the force of nonlinear spring and damper. In order to solve the response of the multi degree of freedom system efficiently, receptance matrix of the system is obtained by neglecting the nonlinearities. They states that the proposed method is an iterative method and effective for small nonlinearities. However, in case of high nonlinearities, their method can not be suitable to obtain the response of a system because of iterative procedure.

Budak and Özgüven [10] proposed a new solution methodology for the force response analysis of nonlinear multi degree of freedom systems which is named Iterative Receptance Method (IRM). Variety of nonlinearities are formed by using nonlinear spring and damping forces. In their study, displacement and velocities are expressed by polynomial form. Nonlinearity of the system is defined in a matrix form which is known as describing function method. In order to make analysis efficiently, IRM is modified for local nonlinearities. They state that convergence problem encountered in their previous studies are eliminated for highly nonlinear system by using IRM.

Tanrkulu et. al. [11] studied on the harmonic force response analysis of nonlinear multi degree of freedom system by utilizing the describing function method (DFM) for symmetric nonlinearities. It is a semi analytical quasi linearization method in frequency domain. They express that time domain solutions are very expensive for large structures, therefore, they obtain steady state response in frequency domain solution by using DFM. In order to reduce the total nonlinear equation number, receptance method is used and then it is combined with DFM. In their study, systems with coulomb friction and cubic stiffness are analyzed by using the proposed method.

Kuran and Özgüven [12] developed a new method for nonlinear vibratory system based on combining modal superposition method and multi-harmonic describing function method. In this method, equation of motion of the structure is transformed into a set of nonlinear algebraic equations in frequency domain, and then the number of nonlinear equation is reduced by using information of linear modes. The solutions obtained by using proposed method are compared with the time domain solutions. They show that the proposed method reduces computational time significantly and provides to obtain high accurate solutions by using a few number of modes. Also effects of higher mode number utilized in analysis procedure and higher harmonic components used for describing the response of the structure are investigated broadly.

E. Cigeroglu et. al. [13] proposed a modal superposition method for nonlinear harmonic forced response analysis of bladed disk structures with a two dimensional microslip friction nonlinearity. Also, in this nonlinearity, normal load variation is considered. The resulting nonlinear friction force and the stick-slip separation is determined by using distributed parameter model. In their study, harmonic balance method (HBM) is employed with an iterative multi-mode solution approach.

E. Cigeroglu et. al. [14] studied on contact interface between under platform damper and bladed disk. Modal superposition method is utilized to transform the equation of motion to a set of nonlinear algebraic equation. In their study, the effect of the number of harmonics to be used in analysis procedure is investigated by utilizing multi-harmonic balance method (MHBM).

Petrov [15] proposed a method based on frequency response function (FRF) matrix to reduce computational time and obtain high accurate results for reduced modelling of jointed structures in frequency domain nonlinear forced response analysis. Petrov states that nonlinear forces can be highly sensitive to relative displacement of contact surfaces, therefore, in order to analyze large scale nonlinear models accurately, the reduced model has to preserve main dynamic characteristics of the structure. Considering component mode synthesis (CMS) methods, the proposed method has major benefits and is easily applicable to finite element codes. In order to state main advantageous of the proposed method, bladed disks with contact interface model is investigated in detail by using harmonic balance method. Also, the effect of mode number on FRF matrix accuracy is studied for forced response analysis for wide range frequency. The force response results of bladed disk structure obtained by proposed method is compared with the results obtained by CMS method and both results are in very good agreement.

Zucca and Epureanu [16] derived a new method based on bilinear modes (BLMs) to reduce the size of the nonlinear structures for forced response analysis in frequency domain in order to reduce computational time. Dynamic behavior of the nonlinear structure is spatially associated in the interested frequency range. In this approach, the response of the nonlinear system is obtained by using harmonic balance method (HBM).

In their study, intermittent contact model is used for nonlinearity in two case studies which are force response analysis of a cracked plate and two coaxial cylinders. For both cases, different contact status is considered such as fully closed contact and partially closed contact. Results computed by the proposed method is compared with results of component mode synthesis (CMS) method. They express that the proposed method provides accurate results and significant reduction of computational time.

Jaumouillé et. al. [17] proposed an adaptive harmonic balance method to reduce computational time in case of nonlinear forced response analysis with higher harmonics. In order to verify the method, bolted joint structure with contact interface is used for case study. They state that when higher harmonics is employed in the frequency region where nonlinearity is dominant, the method gives more accurate results.

Von Groll and Ewins [18] applied arc-length continuation method to overcome the jump phenomenon of the response of nonlinear vibration of rotor with contact interface. Harmonic balance method (HBM) is used to calculate the force response of nonlinear structure. The proposed frequency domain algorithm calculates the response of the system, turning points, bifurcation points and stability of the response. They express that the algorithm can be used for various element types in nonlinear structural dynamics and it is more efficient way than time domain methods in terms of the computational speed.

Petrov and Ewins [19] proposed an analytical approach for multi-harmonic vibration analysis of nonlinear structures with contact interfaces. A friction model is developed for the friction forces under the variable normal load and unilateral interaction along the normal of a contact surface.In their study, the friction interface elements are derived analytically in order to obtain high accurate results and rapid convergence rate of Newton-Raphson method, which is an iterative solution process. They state that it helps to calculate the Jacobian matrix analytically. Utilizing analytic Jacobian matrix can overcome loss of the convergence of the numerical analysis in case of unexpected changes of contact conditions. In order to verify the approach, force response analysis of a sector of a high-pressure turbine-bladed disk is studied. The numerical efficiency and robustness of the analytic approach is proved.

Lewandowski [20] improved a method to analyze systems with geometric nonlinearities for the modal and force response analysis. In his study, multi-harmonic balance method is utilized to obtain higher order solutions.

Riberio and Petyt[20] [21] studied on harmonic forced response of geometric nonlinear thin plates by using harmonic balance method.

Xu et. al. [22] studied on structures with nonlinear piecewise linear viscous damper. Incremental harmonic balance method (IHBM) is employed to obtain response of the structure under harmonic force excitation. They state that there is very good agreement between the results obtained by using IHBM and time integration results.

Guskov et. al.[23] studied on nonlinear dynamic behavior of rotor system with multiple unbalances. Multi-harmonic balance method is employed with combining alternating frequency time domain method. In order to iterative solution procedure and overcome the jump phenomenon, arc-length continuation method is utilized.

Most of the nonlinear equation set obtained by the harmonic balance method (HBM) [24],[25] has to be solved iteratively and numerical solution techniques that are used in nonlinear vibration analysis has to calculate the Jacobian matrix with finite difference estimation since it is not always possible to calculate the Jacobian matrix analytically.

Borrajo et. al. [26] improved an analytic method to calculate Jacobian matrix of the dynamic system having wedge damper. It is exact and completely analytical. The method developed by Petrov and Ewins [19] for a particular friction damper model is extended to a wedge damper model. In their study, set of nonlinear equations is solved iteratively by employing Newton–Raphson method (NRM). The proposed analytic method has been compared to the classical numerical Jacobian calculated finite difference scheme. Comparison of both method shows that the proposed method reduces computational time significantly.

In recent years, the harmonic balance method has been frequently studied by many researchers, in addition, the describing function method has been used in many studies in order to transform equation of motion of the structure into a set of nonlinear algebraic equations in frequency domain for force response analysis.

In the literature, component mode synthesis, modal super position and receptance method have been employed by many researchers to reduce the number of nonlinear equation to be solved and decrease the computational time of the solution process for nonlinear vibration analysis in frequency domain.

In the literature, there are a few approaches to calculate nonlinear forcing terms analytically. Due to complexity of the nonlinearities such as friction element with normal load variation and gap element, most of the studies used numerical solution methods to obtain force response of the structure. In numerical solution methods, finite difference estimation is used to calculate the Jacobian matrix, which is the most important factor that increases the calculation time for iterative solution scheme.

1.3. Objective of the Thesis

In frequency domain nonlinear vibration analysis, iterative numerical methods are used to find harmonic response of the system. It is necessary to calculate Jacobian matrix numerically since it is not always possible to derive Jacobian matrix analytically. In the solution process of the nonlinear equation of motion, calculation of Jacobian matrix increases computational time significantly.

The main objective of this thesis is to develop a new method to reduce computational time of the Jacobian matrix for nonlinear vibratory systems. Validation of the proposed method is made by using lumped model and finite element model. For the models, cubic stiffness and macroslip friction nonlinearities are studied. Solutions obtained by using the proposed method are compared with classical Jacobian calculation method in terms of computational time and computational reduction ratio.

In addition, performance of different nonlinear equation solvers and predictors are investigated and they are compared in terms of computational time and total iteration number.

1.4. Outline of the Thesis

Chapter 2 covers equation of motion of the nonlinear vibratory systems and frequency domain solution method to obtain system response under harmonic excitation. Harmonic balance method is explained to transform the nonlinear equation of motion from time domain to frequency domain. Also, mathematical formulation of receptance method is given to understand how to reduce the nonlinear equation number of multi degree of freedom systems. Different algebraic nonlinear equation solvers are explained with classical numeric Jacobian calculation and different predictors of Arc-length continuation method are formulated. Cubic stiffness and 1-D dry-friction nonlinearities is expressed by using harmonic balance method.

Chapter 3 reviews theory of a new Jacobian calculation method for nonlinear vibratory system response. Mathematical formulation of the proposed method is given in detail for single and multi-harmonic balance method. Total number of necessary calculation to obtain global Jacobian matrix by using classical Jacobian method and the proposed method are compared theoretically.

Chapter 4 copes with case studies to validate the proposed method. Three different lumped parameter models with cubic stiffness nonlinearity and finite element model of bladed disk with dry friction nonlinearity are studied under the harmonic excitation. In the case studies, advantage of the proposed method is shown for different nonlinear equation number and harmonic number. Actual and expected computational reduction ratios are tabulated.

Chapter 5 deals with performance of different nonlinear algebraic equation solvers and predictors of Arc-length continuation method. Total computational time and total iteration number of the solution obtained are compared.

Chapter 6 concludes benefits of the proposed method and the outcomes in this thesis.

CHAPTER 2

PERIODIC RESPONSE ANALYSIS OF NONLINEAR STRUCTURES

2.1. Nonlinear Structures

The equation of motion (EOM) of a vibratory system with nonlinear elements can be given as follows

$$\mathbf{M}\ddot{\mathbf{x}}(t)+\mathbf{C}\dot{\mathbf{x}}(t)+\mathbf{K}\mathbf{x}(t)+\mathbf{f}_N(t)=\mathbf{f}(t), \quad (2.1)$$

where $\mathbf{x}$ is the displacement vector and dot means differentiation with respect to time. Here, $\mathbf{M},\mathbf{C}$ and $\mathbf{K}$ denote the mass, viscous damping, structural damping and stiffness matrices, respectively. $\mathbf{f}_N$ is the vector of internal nonlinear forcing and $\mathbf{f}$ is the harmonic excitation force. For system given in Eq. (2.1), the nonlinear force vector $\mathbf{f}_N$ has elements other than zero for only the DOFs where the nonlinear elements are attached.

2.2. Harmonic Balance Method

In order to calculate the steady- state response of nonlinear differential equations, time and frequency domain methods can be used. However, computational time of the solutions obtained using frequency domain methods is less than the time integration methods. Therefore, it is more advantageous to use frequency domain methods.

Harmonic balance is a method which is mostly used in nonlinear vibration analysis in frequency domain [16], [19], [23], [25]. In this study, the response is assumed to be harmonic so that harmonic balance methods (HBM) is used.

Considering the nonlinear equation of motion defined by Eq. (2.1), displacement response, nonlinear internal force vector and external forcing are stated by using

Fourier series and neglecting bias term in Eq. (2.2), Eq. (2.3) and Eq. (2.4), respectively where θ is ωt .

$$\mathbf{x}_n(t) = \sum_{r=1}^{N_h} \left[\mathbf{u}^{sr} \sin(r\theta) + \mathbf{u}^{cr} \cos(r\theta) \right] \tag{2.2}$$

$$\mathbf{f}_N(t) = \sum_{r=1}^{N_h} \left[\mathbf{f}_N^{sr} \sin(r\theta) + \mathbf{f}_N^{cr} \cos(r\theta) \right] \tag{2.3}$$

$$\mathbf{f}(t) = \sum_{r=1}^{N_h} \left[\mathbf{f}^{sr} \sin(r\theta) + \mathbf{f}^{cr} \cos(r\theta) \right] \tag{2.4}$$

Fourier coefficients of nonlinear internal force vector, $\mathbf{f}_N^{sr}$ and $\mathbf{f}_N^{cr}$, can be calculated as follows,

$$\mathbf{f}_N(t) = \sum_{r=1}^{N_h} \left[\mathbf{f}_N^{sr} \sin(r\theta) + \mathbf{f}_N^{cr} \cos(r\theta) \right], \tag{2.5}$$

$$\mathbf{f}_N^{sr} = \frac{1}{\pi} \int_0^{2\pi} \mathbf{f}_N \sin(r\theta) d\theta, \tag{2.6}$$

$$\mathbf{f}_N^{cr} = \frac{1}{\pi} \int_0^{2\pi} \mathbf{f}_N \cos(r\theta) d\theta. \tag{2.7}$$

2.3. Nonlinearity Types

In this thesis, two different nonlinearities which are cubic stiffness and 1D-Dry friction are investigated to validate the computational performance of the proposed method. In this part, harmonic coefficients of cubic stiffness and 1D-Dry friction nonlinearity are derived by using single and two harmonics.

2.3.1. Cubic Stiffness

Harmonic balance is a method which is mostly used in nonlinear vibration analysis in frequency domain. In this study, the response is assumed to be harmonic so that harmonic balance methods (HBM) is used.

If harmonic balance method is explained by an example in which cubic stiffness is considered as nonlinear element. For cubic stiffness, the nonlinear force is written as

$$F_{kc} = k_c x^3 \tag{2.8}$$

where, k_c is the coefficient of cubic stiffness nonlinearity. For single harmonic balance method, assuming the response as

$$u = U_s \sin(\theta) + U_c \cos(\theta) \tag{2.9}$$

where, U_s and U_c are relative displacement coefficients of harmonics and $\theta = \omega t$. Using single HBM, nonlinear forcing terms are calculated as follows

$$F_N = F_{Ns} \sin(\theta) + F_{Nc} \cos(\theta) \tag{2.10}$$

$$F_{Ns} = \frac{1}{\pi} \int_0^{2\pi} k_c (U_s \sin(\theta) + U_c \cos(\theta))^3 \sin(\theta) d\theta = \frac{3}{4} k_c (U_s^3 + U_c^2 U_s) \tag{2.11}$$

$$F_{Nc} = \frac{1}{\pi} \int_0^{2\pi} k_c (U_s \sin(\theta) + U_c \cos(\theta))^3 \cos(\theta) d\theta = \frac{3}{4} k_c (U_c^3 + U_s^2 U_c) \tag{2.12}$$

Thus, the nonlinear internal forcing becomes

$$F_{Ns} = \frac{3}{4} k_c (U_s^3 + U_c^2 U_s) \sin(\theta) + \frac{3}{4} k_c (U_c^3 + U_s^2 U_c) \cos(\theta) \tag{2.13}$$

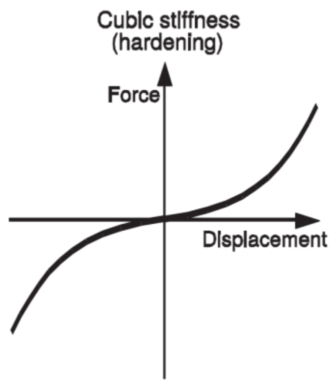

Figure 2.1. Force vs Displacement curve for cubic stiffness nonlinearity

2.3.2. 1D-Dry Friction

Hysteresis loop of 1D friction model with constant normal load is given in Figure 2.2. The given figure is obtained by assuming single harmonic response.

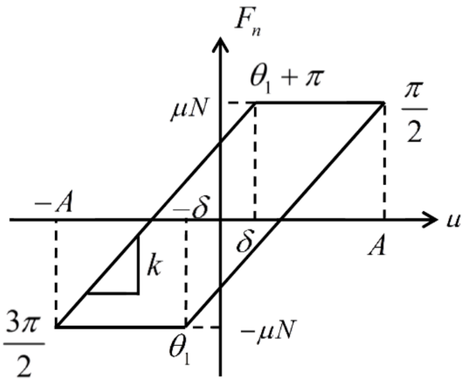

Figure 2.2. Force vs Displacement curve for 1D-Dry friction nonlinearity

For single harmonic balance method, assuming the displacement as

$$u = A\sin(\theta) \tag{2.14}$$

where, A is the displacement amplitude of vibration .

For 1D friction element with constant normal load, the nonlinear force is written as

$$F_N = \begin{cases} -\mu N + k(u+\delta) & \text{for} \quad \frac{\pi}{2} \le \theta \le \theta_1 \\ -\mu N & \text{for} \quad \theta_1 \le \theta \le \frac{3\pi}{2} \end{cases}, \tag{2.15}$$

where, $\delta = \frac{2\mu N - kA}{k}$, $\theta_1 = a\sin(\frac{\delta}{A})$. F_N, μ and N are nonlinear internal forcing, friction coefficient, constant normal load, respectively. Tangential contact stiffness and transition angle are represented by k , θ_1, respectively.

Using single HBM, nonlinear forcing terms are calculated as follows

$$F_N = F_{Ns}\sin(\theta) + F_{Nc}\cos(\theta) \tag{2.16}$$

$$F_{Ns} = \frac{2}{\pi}\left(\int_{\pi/2}^{\theta_1} \left(-\mu N + k\left(A\sin(\theta)+\delta\right)\right)\sin(\theta)d\theta + \int_{\theta_1}^{3/2} \left(-\mu N\right)\sin(\theta)d\theta \right) \quad (2.17)$$

$$F_{Ns} = \frac{kA}{2\pi}(2\theta_1 - \sin(2\theta_1) - \pi) + \frac{2}{\pi}(kA - 2\mu N) \quad (2.18)$$

$$F_{Nc} = \frac{2}{\pi}\left(\int_{\pi/2}^{\theta_1} \left(-\mu N + k\left(A\sin(\theta)+\delta\right)\right)\cos(\theta)d\theta + \int_{\theta_1}^{3/2} \left(-\mu N\right)\cos(\theta)d\theta \right) \quad (2.19)$$

$$F_{Nc} = -\frac{kA\cos^2(\theta_1)}{\pi} + \frac{4\mu \mathrm{N}}{\pi} + \frac{2kA - 4\mu N}{\pi k}(1+\sin(\theta_1)) \quad (2.20)$$

2.4. Receptance Method

For systems with many degrees of freedom, the number of nonlinear equations obtained can be very large. Therefore, several numerical difficulties may merge in the solution process of large systems. In addition to that fact, computational time can increase enormously. In order to decrease the nonlinear equations to be solved, receptance method is used. [7], [13], [14]

Receptance method is employed if the number of DOFs where nonlinearities connected is less than the total DOFs of system. If these DOFs are grouped and the EOM in Eq. (2.1) is combined with harmonic balance method, Eq. (2.1) can be rewritten

$$\left[\mathbf{K} - (r\omega)^2\mathbf{M} + \mathrm{i}r\omega\mathbf{C}\right]\begin{Bmatrix} \mathbf{x}_l^r \\ \mathbf{x}_n^r \end{Bmatrix} + \begin{Bmatrix} \mathbf{0} \\ \mathbf{f}_N^r(\mathbf{x}_n) \end{Bmatrix} = \begin{Bmatrix} \mathbf{f}_l^r \\ \mathbf{f}_n^r \end{Bmatrix}, \quad (2.21)$$

where, sub-indices l and n denote linear and nonlinear DOFs, respectively. Here, $r = q_1, q_2, q_3, ..., N_h$ is the harmonic index, N_h is the number of harmonics used and i represents the unit imaginary number. $\boldsymbol{\alpha}(r\omega)$ is the receptance of the linear part of the structure is expressed in Eq. (2.22).

$$\boldsymbol{\alpha}(r\omega) = \left[\mathbf{K} - (r\omega)^2\mathbf{M} + \mathrm{i}r\omega\mathbf{C}\right]^{-1} \quad (2.22)$$

Multiplying both sides of Eq. (2.21) by $\boldsymbol{\alpha}(r\omega)$ and partitioning the equations to sub-matrices and sub-vectors, following Eq. (2.23) is obtained as follows,

$$\begin{Bmatrix} \mathbf{x}_l^r \\ \mathbf{x}_n^r \end{Bmatrix} + \begin{bmatrix} \boldsymbol{\alpha}_{ll}(r\omega) & \boldsymbol{\alpha}_{nl}(r\omega) \\ \boldsymbol{\alpha}_{nl}(r\omega) & \boldsymbol{\alpha}_{nn}(r\omega) \end{bmatrix} \begin{Bmatrix} \mathbf{0} \\ \mathbf{f}_N^r(\mathbf{x}_n) \end{Bmatrix} - \begin{bmatrix} \boldsymbol{\alpha}_{ll}(r\omega) & \boldsymbol{\alpha}_{nl}(r\omega) \\ \boldsymbol{\alpha}_{nl}(r\omega) & \boldsymbol{\alpha}_{nn}(r\omega) \end{bmatrix} \begin{Bmatrix} \mathbf{f}_l^r \\ \mathbf{f}_n^r \end{Bmatrix} = 0, \qquad (2.23)$$

where, $r = 1,2,3,...,N_h$. By using Eq. (2.23), nonlinear equation set can be clearly re-written in terms of a residual vector as

$$\mathbf{x}_n^r + \boldsymbol{\alpha}_{nn}(r\omega)\mathbf{f}_N^r(\mathbf{x}_n) - \boldsymbol{\alpha}_{nl}(r\omega)\mathbf{f}_l^r(\mathbf{x}_n) - \boldsymbol{\alpha}_{nn}(r\omega)\mathbf{f}_n^r(\mathbf{x}_n) = \mathbf{0}. \qquad (2.24)$$

Solving only Eq. (2.24), results in a considerable decrease in the number of nonlinear equations. In order to obtain total solution, firstly nonlinear algebraic equation set Eq. (2.25) has to be solved simultaneously for all harmonics ($r = 1,2,3,...,N_h$) to obtain response of nonlinear dofs which is $\mathbf{x}_n$.

$$\mathbf{R}(\mathbf{x}_n,\omega) = \begin{Bmatrix} \mathbf{x}_n^1 \\ \mathbf{x}_n^2 \\ \mathbf{x}_n^3 \\ \vdots \\ \mathbf{x}_n^{N_h} \end{Bmatrix} + \begin{bmatrix} \boldsymbol{\alpha}_{nn}(\omega)\mathbf{f}_N^1(\mathbf{x}_n) \\ \boldsymbol{\alpha}_{nn}(2\cdot\omega)\mathbf{f}_N^2(\mathbf{x}_n) \\ \boldsymbol{\alpha}_{nn}(3\cdot\omega)\mathbf{f}_N^3(\mathbf{x}_n) \\ \vdots \\ \boldsymbol{\alpha}_{nn}(N_h\cdot\omega)\mathbf{f}_N^{N_h}(\mathbf{x}_n) \end{bmatrix} - \begin{bmatrix} \boldsymbol{\alpha}_{nl}(\omega)\mathbf{f}_l^1(\mathbf{x}_n) + \boldsymbol{\alpha}_{nn}(\omega)\mathbf{f}_n^1(\mathbf{x}_n) \\ \boldsymbol{\alpha}_{nl}(2\cdot\omega)\mathbf{f}_l^2(\mathbf{x}_n) + \boldsymbol{\alpha}_{nn}(2\cdot\omega)\mathbf{f}_n^2(\mathbf{x}_n) \\ \boldsymbol{\alpha}_{nl}(3\cdot\omega)\mathbf{f}_l^3(\mathbf{x}_n) + \boldsymbol{\alpha}_{nn}(3\cdot\omega)\mathbf{f}_n^3(\mathbf{x}_n) \\ \vdots \\ \boldsymbol{\alpha}_{nl}(N_h\cdot\omega)\mathbf{f}_l^{N_h}(\mathbf{x}_n) + \boldsymbol{\alpha}_{nn}(N_h\cdot\omega)\mathbf{f}_n^{N_h}(\mathbf{x}_n) \end{bmatrix} = \mathbf{0} \qquad (2.25)$$

$$\text{where } r = 1,2,3,...,N_h$$

Then nonlinear internal force vector, $\mathbf{f}_N^r(\mathbf{x}_n)$, can be calculated by the known value of response of nonlinear dofs, $\mathbf{x}_n$.

In order to solve response of the linear DOFs, $\mathbf{x}_l$, can be found by using Eq. (2.26) and the known value of $\mathbf{f}_N^r(\mathbf{x}_n)$.

$$\mathbf{x}_l^r = -\boldsymbol{\alpha}_{nl}(r\omega)\mathbf{f}_N^r(\mathbf{x}_n) + \boldsymbol{\alpha}_{ll}(r\omega)\mathbf{f}_l^r(\mathbf{x}_n) + \boldsymbol{\alpha}_{nl}(r\omega)\mathbf{f}_l^r(\mathbf{x}_n) \quad (r = 1,2,3,...,N_h) \qquad (2.26)$$

2.5. Modal Superposition Method

Another method for the solution of systems with many DOFs is the utilization of modal superposition method. This method is especially important if the number of nonlinear elements is large, where receptance method will result also a large number of nonlinear equations even though this number may be less than the total number of DOFs of the system.

In modal superposition method, the response is written in terms of the linear system modes in matrix form, which can be expressed as

$$\mathbf{x} = \mathbf{\Phi a} e^{i\omega\tau}, \tag{2.27}$$

where, $\mathbf{\Phi}$ and $\mathbf{a}$ are mode shape matrix and unknown modal coefficient vector, respectively. Substituting this equation into the equation of motion in Eq. (2.1) and multiplying both sides by $\mathbf{\Phi}^T$, following equation is obtained.

$$-\omega^2 \mathbf{\Phi}^T \mathbf{M \Phi a} + \mathrm{i}\omega \mathbf{\Phi}^T \mathbf{C \Phi a} + \mathbf{\Phi}^T \mathbf{K \Phi a} + \mathbf{\Phi}^T \mathbf{f}_n(\mathbf{a}) = \mathbf{\Phi}^T \mathbf{f} \tag{2.28}$$

If $\mathbf{\Phi}$ is mass normalized mode shapes of the linear system, this equation simplified as,

$$\left[-\omega^2 \mathbf{I} + \mathrm{i}\omega \mathbf{C}_r + \mathbf{\Omega}\right]\mathbf{a} + \mathbf{\Phi}^T \mathbf{f}_n(\mathbf{a}) = \mathbf{\Phi}^T \mathbf{f}, \tag{2.29}$$

where, $\mathbf{I}$ is the identity matrix, $\mathbf{C}_r = \begin{bmatrix} 2\xi_1\omega_1 & 0 & 0 \\ 0 & \ddots & 0 \\ 0 & 0 & 2\xi_n\omega_n \end{bmatrix}$ and $\mathbf{\Omega} = \begin{bmatrix} \omega_1^{\ 2} & 0 & 0 \\ 0 & \ddots & 0 \\ 0 & 0 & \omega_n^{\ 2} \end{bmatrix}$.

In this nonlinear equation, the unknowns are the modal coefficients given vector $\mathbf{a}$, which are complex. Nonlinear equation set can be clearly re-written in terms of a residual vector as

$$\mathbf{R}(\mathbf{a},\omega) = \left[-\omega^2 \mathbf{I} + \mathrm{i}\omega \mathbf{C}_r + \mathbf{\Omega}\right]\mathbf{a} + \mathbf{\Phi}^T \mathbf{f}_n(\mathbf{a}) - \mathbf{\Phi}^T \mathbf{f}. \tag{2.30}$$

2.6. Solutions of Nonlinear Algebraic Equation

In this section, seven different Newton-like solvers in the literature are given. Each solver has different calculation scheme. Different characteristics of the solvers such as order of convergence, number of Jacobian calculation are specified in Table 2.1.

Table 2.1. *Specific calculation aspects of the solvers*

	Number of Jacobian Calculation	Number of Vector Function Calculation	Number of Matrix Inversion Calculation	Order of Convergence
Newton	1	1	1	2
Solver-1	1	2	1	3
Solver-2	2	3	2	6
Solver-3	1	3	1	5
Solver-4	2	2	2	5
Solver-5	3	1	2	3
Solver-6	3	1	2	3
Solver-7	2	1	2	3

2.6.1. Newton's Solver on Receptance Method

The residual vector for receptance method given by Eq. (2.24) is displacement dependent; therefore, a numerical method is necessary to obtain the solution. Solution of the nonlinear algebraic given by Eq. (2.24) is obtained by using Newton's method. A single iteration for Newton's method can be expressed as

$$(\mathbf{x}_n)_{k+1} = (\mathbf{x}_n)_k - \mathbf{J}\left((\mathbf{x}_n)_k\right)^{-1} \cdot \mathbf{R}\left((\mathbf{x}_n)_k, \omega\right), \tag{2.31}$$

where, k and $\mathbf{J}\left((\mathbf{x}_n)_k\right) = \partial\mathbf{R}\left((\mathbf{x}_n)_k, \omega\right) / \partial(\mathbf{x}_n)_k$ are the iteration number and the Jacobian matrix, respectively. Iterations are terminated when the residual norm falls below a predefined error tolerance.

Global Jacobian matrix, $\mathbf{J}\left(\left(\mathbf{x}_n\right)_k\right)$, calculation is given as follows

$$\mathbf{J}\left(\left(\mathbf{x}_n\right)_k\right)=\mathbf{I}+\begin{bmatrix}\boldsymbol{\alpha}_{nn}(\omega) & \mathbf{0} & \mathbf{0} & \mathbf{0}\\ \mathbf{0} & \boldsymbol{\alpha}_{nn}(2\cdot\omega) & \mathbf{0} & \mathbf{0}\\ \mathbf{0} & \mathbf{0} & \ddots & \mathbf{0}\\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \boldsymbol{\alpha}_{nn}(N_h\cdot\omega)\end{bmatrix}\cdot\mathbf{J}^{fn}\left(\left(\mathbf{x}_n\right)_k\right) \tag{2.32}$$

$$\mathbf{J}^{fn}\left(\left(\mathbf{x}_n\right)_k\right)=\frac{\partial\mathbf{f}_N(\mathbf{x}_n)_k}{\partial\left(\mathbf{x}_n\right)_k} \tag{2.33}$$

where $\mathbf{J}^{fn}\left(\left(\mathbf{x}_n\right)_k\right)$ and $\mathbf{I}$ are Jacobian of nonlinear internal force vector and identity matrix, respectively. Here, calculation of Jacobian of nonlinear internal force vector causes to increase computational time significantly, therefore, the main aim of the proposed method is to decrease computational time.

2.6.2. Newton's Solver on Modal Super Position Method

The residual vector for modal super position method given by Eq. (2.29) is modal coefficient dependent; therefore, a numerical method is applied to obtain the solution. A single iteration for Newton's method can be expressed as

$$\left(\mathbf{a}\right)_{k+1}=\left(\mathbf{a}\right)_k-\mathbf{J}\left(\mathbf{a}_k\right)^{-1}\cdot\mathbf{R}\left(\mathbf{a}_k,\omega\right), \tag{2.34}$$

where, k and $\mathbf{J}\left(\mathbf{a}_k\right)=\partial\mathbf{R}\left(\mathbf{a}_k,\omega\right)/\partial\mathbf{a}_k$ are the iteration number and the Jacobian matrix, respectively. Iterations are terminated when the residual norm falls below a predefined error tolerance.

Global Jacobian matrix, $\mathbf{J}\left(\mathbf{a}_k\right)$, calculation is given as follows

$$\mathbf{J}\left(\mathbf{a}_k\right)=\frac{\partial\mathbf{R}\left(\mathbf{a}_k,\omega\right)}{\partial\mathbf{a}_k}=\left[-\omega^2\mathbf{I}+\mathrm{i}\omega\mathbf{C}_r+\mathbf{\Omega}\right]\mathbf{I}+\mathbf{\Phi}^T\frac{\partial\mathbf{f}_n(\mathbf{x}_k)}{\partial\mathbf{x}_k}\frac{\partial\mathbf{x}_k}{\partial\mathbf{a}_k}\cdot \tag{2.35}$$

where, $\mathbf{J}^{fn}\left(\mathbf{x}_k\right)=\dfrac{\partial\mathbf{f}_N(\mathbf{x}_k)}{\partial\mathbf{x}_k}$ and $\dfrac{\partial\mathbf{x}_k}{\partial\mathbf{a}_k}=\mathbf{\Phi}$, substituting the into Eq. (2.35), following equation is obtained.

$$\mathbf{J}\left(\mathbf{a}_k\right)=\frac{\partial\mathbf{R}\left(\mathbf{a}_k,\omega\right)}{\partial\mathbf{a}_k}=\left[-\omega^2\mathbf{I}+\mathrm{i}\omega\mathbf{C}_r+\mathbf{\Omega}\right]\mathbf{I}+\mathbf{\Phi}^T\mathbf{J}^{fn}\left(\mathbf{x}_k\right)\mathbf{\Phi} \tag{2.36}$$

where $\mathbf{J}^{fn}(\mathbf{x}_k)$ and $\mathbf{I}$ are Jacobian of nonlinear internal force vector and identity matrix, respectively. The purpose of the proposed method is to decrease computational cost of specified Jacobian. In the following sections, the proposed method is explained in detail.

2.6.3. Solver-1

Darvish and Barati [27] proposed a third order newton type method to solve systems of nonlinear equations. Comparing the Newton's method, additional 1 Jacobian and 2 vector function evaluations are necessary, however it has 3rd order convergence rate. The algorithm is given as follows,

$$\mathbf{q}_{k+1} = \mathbf{q}_k - \mathbf{J}(\mathbf{q}_k)^{-1}\left(\mathbf{R}(\mathbf{q}_k) + \mathbf{R}(\mathbf{q}^*_{k+1})\right), \tag{2.37}$$

where, $\mathbf{q}^*_{k+1} = \mathbf{q}_k - \mathbf{J}(\mathbf{q}_k)^{-1}\mathbf{R}(\mathbf{q}_k)$. $\mathbf{J}(\mathbf{q}_k)$ and $\mathbf{R}(\mathbf{q}_k)$ are Jacobian matrix and residual vector of k^{th} iteration, respectively.

2.6.4. Solver-2

Cordero et. al. [28] derived a new technic to design predictor–corrector methods for systems of nonlinear equations. Additional 1 Jacobian matrices and 2 vector function evaluations are done for each step when comparing the Newton's method. The main advantage is to obtain high accurate solutions since it is 6th order convergence algorithm. Algorithm-2 is expressed as follows,

$$\mathbf{y}_k = \mathbf{q}_k - \frac{1}{2}\mathbf{J}(\mathbf{q}_k)^{-1}\mathbf{R}(\mathbf{q}_k) \tag{2.38}$$

$$\mathbf{z}_k = \mathbf{q}_k + \left[\mathbf{J}(\mathbf{q}_k) - 2\mathbf{J}(\mathbf{y}_k)\right]^{-1}\left(3\mathbf{R}(\mathbf{q}_k) - 4\mathbf{R}(\mathbf{y}_k)\right) \tag{2.39}$$

$$\mathbf{q}_{k+1} = \mathbf{z}_k + \left[\mathbf{J}(\mathbf{q}_k) - 2\mathbf{J}(\mathbf{y}_k)\right]^{-1}\mathbf{R}(\mathbf{z}_k) \tag{2.40}$$

2.6.5. Solver-3

Sharma et. al. [29] developed Newton-like method for systems of nonlinear equations. Additional 2 vector function evaluations are done for each step when comparing the Newton's method. The main advantage is to obtain high accurate solutions since it is 5th order convergence algorithm. Algorithm-3 is given as follows,

$$\mathbf{y}_k = \mathbf{q}_k - \mathbf{J}(\mathbf{q}_k)^{-1}\mathbf{R}(\mathbf{q}_k) \tag{2.41}$$

$$\mathbf{z}_k = \mathbf{y}_k - 5\mathbf{J}(\mathbf{q}_k)^{-1}\mathbf{R}(\mathbf{y}_k) \tag{2.42}$$

$$\mathbf{q}_{k+1} = \mathbf{y}_k - \frac{9}{5}\mathbf{J}(\mathbf{q}_k)^{-1}\mathbf{R}(\mathbf{y}_k) - \frac{1}{5}\mathbf{R}(\mathbf{z}_k)\mathbf{J}(\mathbf{q}_k)^{-1} \tag{2.43}$$

2.6.6. Solver-4

Sharma et. al. [30] suggested an efficient fifth order method for solving systems of nonlinear equations. 2 Jacobian matrices and 3 vector function evaluations are necessary for each step to obtain solution. The main advantage of this algorithm over the Newton's method is to obtain high accurate solutions since it is 5th order convergence algorithm. The proposed method is expressed as follows,

$$\mathbf{y}_k = \mathbf{q}_k - \frac{1}{2}\mathbf{J}(\mathbf{q}_k)^{-1}\mathbf{R}(\mathbf{q}_k) \tag{2.44}$$

$$\mathbf{z}_k = \mathbf{q}_k - \mathbf{J}(\mathbf{y}_k)^{-1}\mathbf{R}(\mathbf{q}_k) \tag{2.45}$$

$$\mathbf{q}_{k+1} = \mathbf{z}_k - \left[2\mathbf{J}(\mathbf{y}_k)^{-1} - \mathbf{J}(\mathbf{q}_k)^{-1}\right]\mathbf{R}(\mathbf{z}_k) \tag{2.46}$$

2.6.7. Solver-5

Liu et. al. [31] suggested a a new Newton-type method with third-order for solving systems of nonlinear equations. 2 Jacobian matrices, 1 vector function evaluations and 2 inverse of Jacobian matrices are necessary for each step to obtain solution. It is 3rd order convergence algorithm. The proposed method is formulated as follows,

$$\mathbf{y}_k = \mathbf{q}_k - \mathbf{J}(\mathbf{q}_k)^{-1}\mathbf{R}(\mathbf{q}_k) \tag{2.47}$$

$$\mathbf{q}_{k+1} = \mathbf{q}_k - \left[\frac{1}{3}\mathbf{J}(\mathbf{y}_k) + \frac{2}{3}\mathbf{J}\left(\frac{\mathbf{q}_k + 3\mathbf{y}_k}{4}\right)\right]^{-1}\mathbf{R}(\mathbf{q}_k) \tag{2.48}$$

2.6.8. Solver-6

Noor et. al. [32] suggested a method for solving systems of nonlinear equations. 2 Jacobian matrices and 1 vector function evaluations are necessary for each step to obtain solution. It is 3rd order convergence algorithm. The proposed method is formulated as follows,

$$\mathbf{y}_k = \mathbf{q}_k - \mathbf{J}(\mathbf{q}_k)^{-1}\mathbf{R}(\mathbf{q}_k) \tag{2.49}$$

$$\mathbf{q}_{k+1} = \mathbf{q}_k - 4\left[\mathbf{J}(\mathbf{y}_k) + 3\mathbf{J}\left(\frac{\mathbf{q}_k - 2\mathbf{y}_k}{3}\right)\right]^{-1}\mathbf{R}(\mathbf{q}_k) \tag{2.50}$$

2.6.9. Solver-7

Cordero et. al. [33] suggested a variant of Newton’s method to solve functions of several variables. 2 Jacobian matrices and 1 vector function evaluations are necessary for each step to obtain solution. It is 3rd order convergence algorithm. The proposed method is formulated as follows,

$$\mathbf{y}_k = \mathbf{q}_k - \mathbf{J}(\mathbf{q}_k)^{-1}\mathbf{R}(\mathbf{q}_k) \tag{2.51}$$

$$\mathbf{q}_{k+1} = \mathbf{q}_k - \mathbf{J}\left(\frac{\mathbf{q}_k + \mathbf{y}_k}{2}\right)\mathbf{R}(\mathbf{q}_k) \tag{2.52}$$

This solution scheme is named as *Midpoint Newton’s method.*

2.7. Arc-length Continuation Method

The response curve of the nonlinear systems may have turn back behavior, which means there is more than a solution for a specific frequency value. This fact may result in convergence problem or occurrence of the jump phenomenon during the solution process. In order to avoid that kind of problems, arc-length continuation method which is a path following solution method is used.

In this path following method, a new parameter which is frequency ω for the nonlinear response analysis is added to nonlinear equations set. This provides to obtain stable Jacobian matrix for turning back points. The radius of a hypothetical sphere r_s is selected to find the next solution point on the desired path.

There is a new unknown parameter ω , therefore the unknown vector is updated as follows,

$$\mathbf{q} = \begin{Bmatrix} \mathbf{x} \\ \omega \end{Bmatrix} \tag{2.53}$$

Equation of the sphere centered at the previous solution is expressed as follows,

$$\left(\Delta\mathbf{x}_k\right)^2 + \left(\Delta\omega_k\right)^2 = r_s^2, \tag{2.54}$$

where, $\Delta\mathbf{x}_k = \left(\mathbf{x}_k - \mathbf{x}_{k-1}\right)$, $\Delta\omega_k = \left(\omega_k - \omega_{k-1}\right)$. Substituting $\Delta\mathbf{q}_k = \begin{Bmatrix} \Delta\mathbf{x}_k \\ \Delta\omega_k \end{Bmatrix}$ into the equation of sphere, the extra equation needed can be obtained as

$$h(\mathbf{x}_k, \omega_k) = \Delta\mathbf{q}_k^T \cdot \Delta\mathbf{q}_k - r_s^2 = 0. \tag{2.55}$$

Considering the Eq. (2.55) together with nonlinear equation of motion, new equation to be solved is obtained as,

$$\Delta\mathbf{q}_{k+1} = \mathbf{q}_k - \begin{bmatrix} \dfrac{\partial\mathbf{R}(\mathbf{x}_k, \omega_k)}{\partial\mathbf{x}} & \dfrac{\partial\mathbf{R}(\mathbf{x}_k, \omega_k)}{\partial\omega} \\ \dfrac{\partial h(\mathbf{x}_k, \omega_k)}{\partial\mathbf{x}} & \dfrac{\partial h(\mathbf{x}_k, \omega_k)}{\partial\omega} \end{bmatrix}^{-1} \begin{Bmatrix} \mathbf{R}(\mathbf{x}_k, \omega_k) \\ h(\mathbf{x}_k, \omega_k) \end{Bmatrix}, \tag{2.56}$$

where, $\begin{bmatrix} \frac{\partial h(\mathbf{x}_k, \omega_k)}{\partial \mathbf{x}} & \frac{\partial h(\mathbf{x}_k, \omega_k)}{\partial \omega} \end{bmatrix} = \begin{bmatrix} 2\Delta \mathbf{q}_k \end{bmatrix}^T$. It is Newton's iterative solution process.

For most cases, Arc-length parameter r_s^2 is kept constant however it may increase the solution time when it is selected too small. Also, it may cause the convergence problem at sharp turning points. Therefore, the best way to avoid that kind of problems is to use adaptive arc-length parameter algorithm [34]. This algorithm provides to update r_s parameter for each iteration by comparing the total iteration number and optimum iteration number.

The adaptive arc-length is formulated as

$$\left(r_s\right)_k = \left(r_s\right)_{k-1} \left(\frac{n_{opt}^{iter}}{n_{k-1}^{iter}} \right)^{1/3}, \tag{2.57}$$

where, n_{k-1}^{iter} is the total number of iteration at solution of $k-1$ and n_{opt}^{iter} is the optimum iteration number.

2.8. Predictors for Path Following Methods

2.8.1. Tangent Predictor

A good initial guess is necessary to converge to the solution of nonlinear system by using Newton's method. Better initial guess helps to reduce iteration number therefore it is significantly important considering computational time.

Tangent predictor is a first order predictor to find initial guess. In order to use tangent predictor, expanding the solution at the next frequency step by using first order Tailor series is necessary. The initial guess can be calculated as

$$\mathbf{q}_{IG}(r_s + \delta r_s) = \mathbf{q}_k(\omega) + \left(-\mathbf{J}(\mathbf{q}_k, r_s)^{-1} \cdot \frac{\partial \mathbf{R}(\mathbf{q}_k, r_S)}{\partial s} \right) \delta r_s, \tag{2.58}$$

where $\mathbf{q}_{IG}$ is the initial guess for the $k+1$ solution.

2.9. Proposed Predictors

2.9.1. Linear Predictor

In this part, linear curve fit predictor for arc-length continuation method is formulated. All the unknown values are function of r_s . In order to find the coefficients of liner equation, Eq. (2.59) and (2.60) are solved together by knowing the solutions $\mathbf{q}_{k-1}$ and $\mathbf{q}_k$ since there are two unknown coefficient vectors, $\mathbf{a}$ and $\mathbf{b}$.

$$\mathbf{q}_{k-1} = \mathbf{a}\sum_{l=1}^{k-1}(r_s)_l + \mathbf{b} \tag{2.59}$$

$$\mathbf{q}_k = \mathbf{a}\sum_{l=1}^{k}(r_s)_l + \mathbf{b} \tag{2.60}$$

After finding coefficient vectors which are $\mathbf{a}$ and $\mathbf{b}$, initial guess for $k+1$ iteration is calculated as

$$(\mathbf{q}_{IG})_{k+1} = \mathbf{a}\sum_{l=1}^{k+1}(r_s)_l + \mathbf{b}. \tag{2.61}$$

2.9.2. Second Order Polynomial Predictor

Second order polynomial curve fit for arc-length continuation method is formulated. All the unknown values are function of r_s . In order to find the coefficients of second order polynomial equation, Eq. (2.62),(2.63) and (2.64) are solved together by knowing the solutions $\mathbf{q}_{k-2}$, $\mathbf{q}_{k-1}$ and $\mathbf{q}_k$ since there are three unknown coefficient vectors, $\mathbf{a}$, $\mathbf{b}$ and $\mathbf{c}$.

$$\mathbf{q}_{k-2} = \mathbf{a}\left(\sum_{l=1}^{k-2}(r_s)_l\right)^2 + \mathbf{b}\sum_{l=1}^{k-2}(r_s)_l + \mathbf{c} \tag{2.62}$$

$$\mathbf{q}_{k-1} = \mathbf{a}\left(\sum_{l=1}^{k-1}(r_s)_l\right)^2 + \mathbf{b}\sum_{l=1}^{k-1}(r_s)_l + \mathbf{c} \tag{2.63}$$

$$\mathbf{q}_k = \mathbf{a}\left(\sum_{l=1}^{k}(r_s)_l\right)^2 + \mathbf{b}\sum_{l=1}^{k}(r_s)_l + \mathbf{c} \tag{2.64}$$

After finding coefficient vectors which are $\mathbf{a}$, $\mathbf{b}$ and $\mathbf{c}$, initial guess for $k+1$ iteration is calculated as

$$(\mathbf{q}_{IG})_{k+1} = \mathbf{a}\left(\sum_{l=1}^{k+1}(r_s)_l\right)^2 + \mathbf{b}\sum_{l=1}^{k+1}(r_s)_l + \mathbf{c}. \tag{2.65}$$

2.9.3. Third Order Polynomial Predictor

Third order polynomial curve fit for arc-length continuation method is formulated. All the unknown values are function of r_s . In order to find the coefficients of second order polynomial equation, Eq. (2.66), (2.67), (2.68) and (2.69) are solved together by knowing the solutions $\mathbf{q}_{k-3}$, $\mathbf{q}_{k-2}$, $\mathbf{q}_{k-1}$ and $\mathbf{q}_k$ since there are three unknown coefficient vectors, $\mathbf{a}$, $\mathbf{b}$, $\mathbf{c}$ and $\mathbf{d}$.

$$\mathbf{q}_{k-3} = \mathbf{a}\left(\sum_{l=1}^{k-3}(r_s)_l\right)^3 + \mathbf{b}\left(\sum_{l=1}^{k-3}(r_s)_l\right)^2 + \mathbf{c}\sum_{l=1}^{k-3}(r_s)_l + \mathbf{d} \tag{2.66}$$

$$\mathbf{q}_{k-2} = \mathbf{a}\left(\sum_{l=1}^{k-2}(r_s)_l\right)^3 + \mathbf{b}\left(\sum_{l=1}^{k-2}(r_s)_l\right)^2 + \mathbf{c}\sum_{l=1}^{k-2}(r_s)_l + \mathbf{d} \tag{2.67}$$

$$\mathbf{q}_{k-1} = \mathbf{a}\left(\sum_{l=1}^{k-1}(r_s)_l\right)^3 + \mathbf{b}\left(\sum_{l=1}^{k-1}(r_s)_l\right)^2 + \mathbf{c}\sum_{l=1}^{k-1}(r_s)_l + \mathbf{d} \tag{2.68}$$

$$\mathbf{q}_{k} = \mathbf{a}\left(\sum_{l=1}^{k}(r_s)_l\right)^3 + \mathbf{b}\left(\sum_{l=1}^{k}(r_s)_l\right)^2 + \mathbf{c}\sum_{l=1}^{k}(r_s)_l + \mathbf{d} \tag{2.69}$$

After finding coefficient vectors which are $\mathbf{a}$, $\mathbf{b}$, $\mathbf{c}$ and $\mathbf{d}$ initial guess for $k+1$ iteration is calculated as

$$(\mathbf{q}_{IG})_{k+1} = \mathbf{a}\left(\sum_{l=1}^{k+1}(r_s)_l\right)^3 + \mathbf{b}\left(\sum_{l=1}^{k+1}(r_s)_l\right)^2 + \mathbf{c}\sum_{l=1}^{k+1}(r_s)_l + \mathbf{d} \tag{2.70}$$

2.9.4. Logarithmic Predictor

Logarithmic curve fit predictor for arc-length continuation method is formulated. All the unknown values are function of r_s . In order to find the coefficients of liner equation, Eq. (2.71)and (2.72) are solved together by knowing the solutions $\mathbf{q}_{k-1}$ and $\mathbf{q}_k$ since there are two unknown coefficient vectors, $\mathbf{a}$ and $\mathbf{b}$.

$$\mathbf{q}_{k-1} = \mathbf{a}\ln\left(\sum_{l=1}^{k-1}(r_s)_l\right) + \mathbf{b} \tag{2.71}$$

$$\mathbf{q}_{k} = \mathbf{a}\ln\left(\sum_{l=1}^{k}(r_s)_l\right) + \mathbf{b} \tag{2.72}$$

After finding coefficient vectors which are $\mathbf{a}$ and $\mathbf{b}$, initial guess for $k+1$ iteration is calculated as

$$(\mathbf{q}_{IG})_{k+1} = \mathbf{a}\ln\left(\sum_{l=1}^{k+1}(r_s)_l\right) + \mathbf{b}. \tag{2.73}$$

2.9.5. Quadratic Spline Predictor

The most common interpolants are polynomials since it is easy to evaluate, differentiate and integrate. At least 3 data points have to be necessary to use quadratic interpolation method. Two second order polynomial functions are placed between 3 data points. First polynomial function is fitted between point 1 which is $\mathbf{q}_{k-2}$ and point 2 which is $\mathbf{q}_{k-1}$ and second polynomial function is fitted between point 2 which is $\mathbf{q}_{k-1}$ and point 3 which is $\mathbf{q}_k$. First polynomial and second polynomial values at $\mathbf{q}_{k-1}$ are equal to each other and also first derivative of them at point $\mathbf{q}_{k-1}$ is equal to each other and given in (2.78). Six unknown coeffieints are $\mathbf{a}_{k-1}, \mathbf{b}_{k-1}, \mathbf{c}_{k-1}, \mathbf{a}_k, \mathbf{b}_k$ and $\mathbf{c}_k$ however total number of equalities are five. Therefore, for the first interval which is between point 1 and point 2, first polynomial is assumed linear, which means that $\mathbf{a}_{k-1} = 0$. This assumption provides to reduce the number of unknowns. The other unknowns can be found by solving Eq. (2.74)-(2.78).

$$\mathbf{q}_{k-2}=\mathbf{a}_{k-1}\left(\sum_{l=1}^{k-2}(r_s)_l\right)^2+\mathbf{b}_{k-1}\sum_{l=1}^{k-2}(r_s)_l+\mathbf{c}_{k-1} \tag{2.74}$$

$$\mathbf{q}_{k-1}=\mathbf{a}_{k-1}\left(\sum_{l=1}^{k-1}(r_s)_l\right)^2+\mathbf{b}_{k-1}\sum_{l=1}^{k-1}(r_s)_l+\mathbf{c}_{k-1} \tag{2.75}$$

$$\mathbf{q}_{k-1}=\mathbf{a}_{k}\left(\sum_{l=1}^{k-1}(r_s)_l\right)^2+\mathbf{b}_{k}\sum_{l=1}^{k-1}(r_s)_l+\mathbf{c}_{k} \tag{2.76}$$

$$\mathbf{q}_{k}=\mathbf{a}_{k}\left(\sum_{l=1}^{k}(r_s)_l\right)^2+\mathbf{b}_{k}\sum_{l=1}^{k}(r_s)_l+\mathbf{c}_{k} \tag{2.77}$$

$$2\mathbf{a}_{k-1}\left(\sum_{l=1}^{k-1}(r_s)_l\right)+\mathbf{b}_{k-1}=2\mathbf{a}_{k}\left(\sum_{l=1}^{k-1}(r_s)_l\right)+\mathbf{b}_{k} \tag{2.78}$$

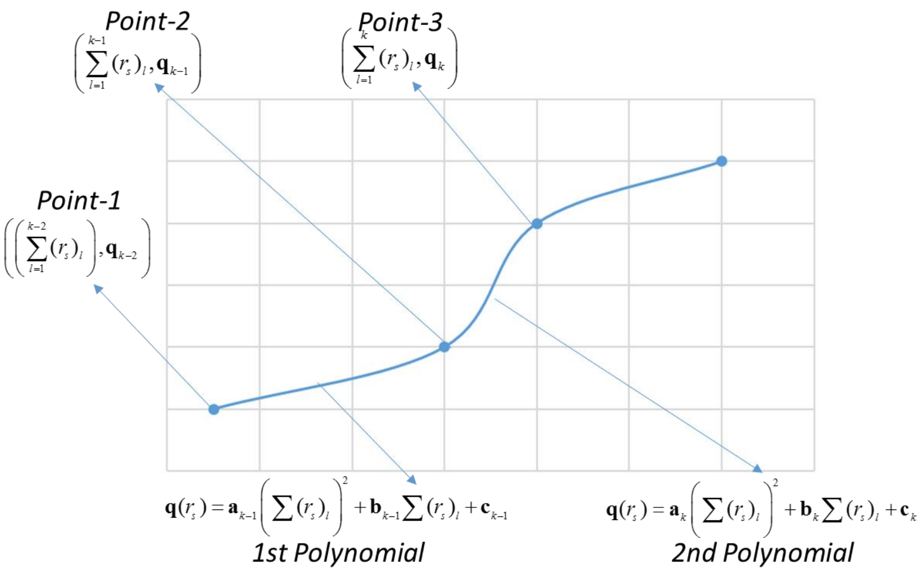

Figure 2.3. Quadratic spline explanation

2.10. Classical Jacobian Matrix Calculation

It is not always available to calculate the analytical Jacobian and also, it is complicated to calculate for highly nonlinear equations analytically, therefore, numerical approximations to the Jacobian can be used instead. Forward difference method is the one of the methods to calculate Jacobian numerically [35] . It has first order error O(h) , but it requires p^2 calculation for an $p \times p$ system. Forward difference formula is given as follows

$$\mathbf{J}_{ij}\left(\left(\mathbf{x}_n\right)_k, \omega\right) = \frac{\mathbf{R}_i\left(\left(\mathbf{x}_n\right)_k + h_i \mathbf{e}_j, r, \omega\right) - \mathbf{R}_i\left(\left(\mathbf{x}_n\right)_k, r, \omega\right)}{h_j}, \qquad (2.79)$$

where, $\left\{e_j\right\}$ is the unit vector in the j^{th} direction and h_j is a scaled step size.

CHAPTER 3

A NEW JACOBIAN CALCULATION METHOD FOR PERIODIC FORCED RESPONSE ANALYSIS OF NONLINEAR STRUCTURES-JACOBIAN ELEMENT METHOD

3.1. Jacobian Element Method for Single Harmonic

Nonlinear internal forcing vector between two DOFs (n = 2) for single harmonic balance method can be expressed as

$$f_n(u_i^{s1} - u_j^{s1}, u_i^{c1} - u_j^{c1}) = f_n(y_s, y_c) = f_{ns}(y_s, y_c)\sin(\theta) + f_{nc}(y_s, y_c)\cos(\theta) \quad (3.1)$$

$$f_N = \begin{pmatrix} f_{ns}(y_s, y_c) \\ -f_{ns}(y_s, y_c) \\ f_{nc}(y_s, y_c) \\ -f_{nc}(y_s, y_c) \end{pmatrix} = \begin{pmatrix} f_1 \\ f_2 \\ f_3 \\ f_4 \end{pmatrix} = \begin{pmatrix} f_1 \\ -f_1 \\ f_3 \\ -f_3 \end{pmatrix} \quad (3.2)$$

where, $y_s = u_i^{s1} - u_j^{s1}$ and $y_c = u_i^{c1} - u_j^{c1}$ are harmonic coefficient differences of i^{th} and j^{th} dofs. By using Eq. (3.2), 4x4 elemental Jacobian matrix for a single nonlinear element can be constructed for single harmonic balance method where unknown harmonic coefficient vector $\mathbf{x}_n^r$ is arranged $\left\{u_i^{s1}, u_j^{s1}, u_i^{c1}, u_j^{c1}\right\}^{\mathbf{T}}$. All first order partial derivatives of the nonlinear forcing vector in terms of harmonic coefficient differences are given in Eq. (3.3)-(3.18)

$$\frac{\partial f_1}{\partial u_i^{s1}} = \frac{\partial f_{ns}(y_s, y_c)}{\partial y_s} \cdot \frac{\partial y_s}{\partial u_i^{s1}} = \frac{\partial f_{ns}(y_s, y_c)}{\partial y_s} \cdot (1) = a \quad (3.3)$$

$$\frac{\partial f_1}{\partial u_j^{s1}} = \frac{\partial f_{ns}(y_s, y_c)}{\partial y_s} \cdot \frac{\partial y_s}{\partial u_j^{s1}} = \frac{\partial f_{ns}(y_s, y_c)}{\partial y_s} \cdot (-1) = -a \quad (3.4)$$

$$\frac{\partial f_1}{\partial u_i^{c1}} = \frac{\partial f_{ns}(y_s, y_c)}{\partial y_c} \cdot \frac{\partial y_c}{\partial u_i^{c1}} = \frac{\partial f_{ns}(y_s, y_c)}{\partial y_c} \cdot (1) = b \quad (3.5)$$

$$\frac{\partial f_1}{\partial u_j^{c1}} = \frac{\partial f_{ns}(y_s, y_c)}{\partial y_c} \cdot \frac{\partial y_c}{\partial u_j^{c1}} = \frac{\partial f_{ns}(y_s, y_c)}{\partial y_c} \cdot (-1) = -b \tag{3.6}$$

$$\frac{\partial f_2}{\partial u_i^{s1}} = -\frac{\partial f_{ns}(y_s, y_c)}{\partial y_s} \cdot \frac{\partial y_s}{\partial u_i^{s1}} = -\frac{\partial f_{ns}(y_s, y_c)}{\partial y_s} \cdot (1) = -a, \text{ since } f_2 = f_1 \tag{3.7}$$

$$\frac{\partial f_2}{\partial u_j^{s1}} = -\frac{\partial f_{ns}(y_s, y_c)}{\partial y_s} \cdot \frac{\partial y_s}{\partial u_j^{s1}} = -\frac{\partial f_{ns}(y_s, y_c)}{\partial y_s} \cdot (-1) = a, \text{ since } f_2 = f_1 \tag{3.8}$$

$$\frac{\partial f_2}{\partial u_i^{c1}} = -\frac{\partial f_{ns}(y_s, y_c)}{\partial y_c} \cdot \frac{\partial y_c}{\partial u_i^{c1}} = -\frac{\partial f_{ns}(y_s, y_c)}{\partial y_c} \cdot (1) = -b, \text{ since } f_2 = f_1 \tag{3.9}$$

$$\frac{\partial f_2}{\partial u_j^{c1}} = -\frac{\partial f_{ns}(y_s, y_c)}{\partial y_c} \cdot \frac{\partial y_c}{\partial u_j^{c1}} = -\frac{\partial f_{ns}(y_s, y_c)}{\partial y_c} \cdot (-1) = b, \text{ since } f_2 = f_1 \tag{3.10}$$

$$\frac{\partial f_3}{\partial u_i^{s1}} = \frac{\partial f_{nc}(y_s, y_c)}{\partial y_s} \cdot \frac{\partial y_s}{\partial u_i^{s1}} = \frac{\partial f_{nc}(y_s, y_c)}{\partial y_s} \cdot (1) = c \tag{3.11}$$

$$\frac{\partial f_3}{\partial u_j^{s1}} = \frac{\partial f_{nc}(y_s, y_c)}{\partial y_s} \cdot \frac{\partial y_s}{\partial u_j^{s1}} = \frac{\partial f_{nc}(y_s, y_c)}{\partial y_s} \cdot (-1) = -c \tag{3.12}$$

$$\frac{\partial f_3}{\partial u_i^{c1}} = \frac{\partial f_{nc}(y_s, y_c)}{\partial y_c} \cdot \frac{\partial y_c}{\partial u_i^{c1}} = \frac{\partial f_{nc}(y_s, y_c)}{\partial y_c} \cdot (1) = d \tag{3.13}$$

$$\frac{\partial f_3}{\partial u_j^{c1}} = \frac{\partial f_{nc}(y_s, y_c)}{\partial y_c} \cdot \frac{\partial y_c}{\partial u_j^{c1}} = \frac{\partial f_{nc}(y_s, y_c)}{\partial y_c} \cdot (-1) = -d \tag{3.14}$$

$$\frac{\partial f_4}{\partial u_i^{s1}} = -\frac{\partial f_{nc}(y_s, y_c)}{\partial y_s} \cdot \frac{\partial y_s}{\partial u_i^{s1}} = -\frac{\partial f_{nc}(y_s, y_c)}{\partial y_s} \cdot (1) = -c, \text{ since } f_4 = f_3 \tag{3.15}$$

$$\frac{\partial f_4}{\partial u_j^{s1}} = -\frac{\partial f_{nc}(y_s, y_c)}{\partial y_s} \cdot \frac{\partial y_s}{\partial u_j^{s1}} = -\frac{\partial f_{nc}(y_s, y_c)}{\partial y_s} \cdot (-1) = c, \text{ since } f_4 = f_3 \tag{3.16}$$

$$\frac{\partial f_4}{\partial u_i^{c1}} = -\frac{\partial f_{nc}(y_s, y_c)}{\partial y_c} \cdot \frac{\partial y_c}{\partial u_i^{c1}} = -\frac{\partial f_{nc}(y_s, y_c)}{\partial y_c} \cdot (1) = -d, \text{ since } f_4 = f_3 \tag{3.17}$$

$$\frac{\partial f_4}{\partial u_j^{c1}} = -\frac{\partial f_{nc}(y_s, y_c)}{\partial y_c} \cdot \frac{\partial y_c}{\partial u_j^{c1}} = -\frac{\partial f_{nc}(y_s, y_c)}{\partial y_c} \cdot (-1) = d, \text{ since } f_4 = f_3 \tag{3.18}$$

For single harmonic balance method, calculated 4×4 elemental Jacobian matrix, $\mathbf{J}_e^{ij}$, for a nonlinear element between 2 dofs is given in Eq. (3.19).

$$\mathbf{J}_e^{ij} = \begin{bmatrix} \frac{\partial f_1}{\partial u_i^{s1}} & \frac{\partial f_1}{\partial u_j^{s1}} & \frac{\partial f_1}{\partial u_i^{c1}} & \frac{\partial f_1}{\partial u_j^{c1}} \\ \frac{\partial f_2}{\partial u_i^{s1}} & \frac{\partial f_2}{\partial u_j^{s1}} & \frac{\partial f_2}{\partial u_i^{c1}} & \frac{\partial f_2}{\partial u_j^{c1}} \\ \frac{\partial f_3}{\partial u_i^{s1}} & \frac{\partial f_3}{\partial u_j^{s1}} & \frac{\partial f_3}{\partial u_i^{c1}} & \frac{\partial f_3}{\partial u_j^{c1}} \\ \frac{\partial f_4}{\partial u_i^{s1}} & \frac{\partial f_4}{\partial u_j^{s1}} & \frac{\partial f_4}{\partial u_i^{c1}} & \frac{\partial f_4}{\partial u_j^{c1}} \end{bmatrix} = \begin{bmatrix} a^{ij} & -a^{ij} & b^{ij} & -b^{ij} \\ -a^{ij} & a^{ij} & -b^{ij} & b^{ij} \\ c^{ij} & -c^{ij} & d^{ij} & -d^{ij} \\ -c^{ij} & c^{ij} & -d^{ij} & d^{ij} \end{bmatrix} \tag{3.19}$$

For multi-harmonic balance method, elemental Jacobian matrix size is equal to $4mx4m$ where m is the number of harmonics used in the harmonic motion response. One can realize that it is not efficient way to calculate all partial derivatives of the nonlinear forcing vector because the calculation of the required 4 partial derivatives is sufficient to obtain elemental Jacobian matrix for single harmonic balance method. Therefore, to reduce the size of matrix $\mathbf{J}_e^{ij}$, $\mathbf{T}^{ij}$ matrix is defined in Eq.(3.20), which is named as reduced elemental Jacobian matrix.

$$\mathbf{T}^{ij} = \begin{bmatrix} a & b \\ c & d \end{bmatrix} \tag{3.20}$$

So far, Jacobian element method is explained for single harmonic balance method and 1D nonlinear element model.

3.2. Assembly Rule for Elemental Jacobian Matrices to Force Jacobian Matrix

For some cases where a nonlinear forcing term can be coupled other dofs, for example friction contact model with normal load variation. In order to apply Jacobian element method for general cases where MDOF system and mutli harmonic balance method is used, firstly, unknown harmonic coefficients vector $\mathbf{x}_n^r$ is arranged according to harmonic numbers and dofs with following form in Eq. (3.21).

$$\mathbf{u} = \left\{u_{1,}^{s1}...u_{nn}^{s1}, u_{1,}^{c1}...u_{nn}^{c1},....,u_{1,}^{sr}...u_{nn}^{sr}, u_{1,}^{cr}...u_{nn}^{cr}\right\}^{\mathbf{T}}$$
$$\mathbf{v} = \left\{v_{1,}^{s1}...v_{nn}^{s1}, v_{1,}^{c1}...v_{nn}^{c1},....,v_{1,}^{sr}...v_{nn}^{sr}, v_{1,}^{cr}...v_{nn}^{cr}\right\}^{\mathbf{T}}$$
$$\mathbf{w} = \left\{w_{1,}^{s1}...w_{nn}^{s1}, w_{1,}^{c1}...w_{nn}^{c1},.....,w_{1,}^{sr}...w_{nn}^{sr}, w_{1,}^{cr}...w_{nn}^{cr}\right\}^{\mathbf{T}} \tag{3.21}$$
$$\mathbf{x}_n^r = \begin{Bmatrix} \mathbf{u} \\ \mathbf{v} \\ \mathbf{w} \end{Bmatrix}$$

Secondly, size of the reduced elemental Jacobian matrix $(s \times s)$ should be calculated as follows,

$$s = 2N_h n_d, \tag{3.22}$$

where, n_d is the number of degree of freedom which effects the nonlinear element. For example, n_d is equal to 3, if a nonlinear element is coupled with three different directions such as u, v, w.

Elemental Jacobian matrices should be assembled into the appropriate locations of the force Jacobian matrix $\mathbf{J}^{fn}$. Assembly rule for elemental Jacobian matrices into force Jacobian matrix is given as follows

$$\mathbf{J}^{fn}_{i+s_{row}\cdot n, i+s_{col}\cdot n} = \sum_{j=1}^{n} \mathbf{T}^{i,j}_{s_{row}+1, s_{col}+1}, \tag{3.23}$$

where, $i = 1, 2, ..., n, \quad s_{row} = 0, 1, ..., s-1, \quad s_{col} = 0, 1, ..., s-1.$

$$\mathbf{J}^{fn}_{i+s_{row}\cdot n, j+s_{col}\cdot n} = -\mathbf{T}^{i,j}_{s_{row}+1, s_{col}+1}, \tag{3.24}$$

where, $i \neq j, \quad s_{row} = 0, 1, ..., s-1, \quad s_{col} = 0, 1, ..., s-1.$ n is the total number of node with nonlinear element connection.

3.3. Application of Assembly Rule

In order to clarify the assembly rule for elemental Jacobian matrices to force Jacobian matrix, 3 DOFs lumped model with four nonlinear elements in Figure 3.1 is investigated by using single harmonic balance method. For this model, $N_h = 1$, $n_d = 1$, $s = 2$ and $nn = 3$. To apply assembly rule correctly, also unknown harmonic coefficient vector $\mathbf{x}_n^r$ has to be arranged as $\left\{u_1^{s1}, u_2^{s1}, u_3^{s1}, u_1^{c1}, u_2^{c1}, u_3^{c1}\right\}^{\mathbf{T}}$.

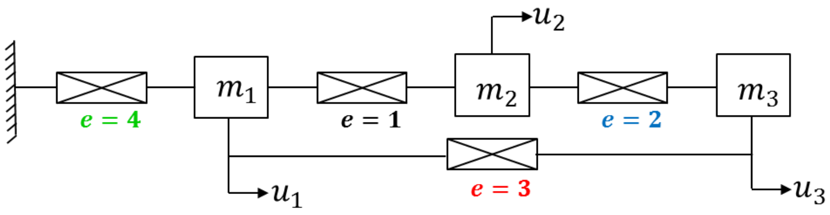

Figure 3.1. 3-DOF lumped model

In the first step, all reduced Jacobian element matrices have to be calculated and they are expressed in Eq. (3.25) and (3.26). 2nd and 3rd DOF is not connected to ground with a nonlinear element, therefore, $\mathbf{T}^{2,2} = \mathbf{T}^{3,3} = 0$. In order to obtain the force Jacobian matrix for 3-DOF lumped model, only four reduced elemental Jacobian matrix calculation is enough and this fact can clearly be seen from Eq. (3.25) and (3.26),

$$\mathbf{T}^{1,2} = \mathbf{T}^{2,1} = \begin{bmatrix} a^{1,2} & b^{1,2} \\ c^{1,2} & d^{1,2} \end{bmatrix}, \quad \mathbf{T}^{1,3} = \mathbf{T}^{3,1} = \begin{bmatrix} a^{1,3} & b^{1,3} \\ c^{1,3} & d^{1,3} \end{bmatrix}, \quad \mathbf{T}^{2,3} = \mathbf{T}^{3,2} = \begin{bmatrix} a^{2,3} & b^{2,3} \\ c^{2,3} & d^{2,3} \end{bmatrix} \quad (3.25)$$

$$\mathbf{T}^{1,1} = \begin{bmatrix} a^{1,1} & b^{1,1} \\ c^{1,1} & d^{1,1} \end{bmatrix}, \qquad \mathbf{T}^{2,2} = \mathbf{0}, \qquad \mathbf{T}^{3,3} = \mathbf{0} \quad (3.26)$$

By using Eq. (3.23), some elements of the force Jacobian matrix are calculated as follows,

$$\begin{aligned} &\mathbf{J}^{fn}_{1,1} = a^{1,1} + a^{1,2} + a^{1,3}, && \mathbf{J}^{fn}_{2,2} = a^{2,1} + a^{2,2} + a^{2,3}, && \mathbf{J}^{fn}_{3,3} = a^{3,1} + a^{3,2} + a^{3,3}, \\ &\mathbf{J}^{fn}_{1,4} = b^{1,1} + b^{1,2} + b^{1,3}, && \mathbf{J}^{fn}_{2,5} = b^{2,1} + b^{2,2} + b^{2,3}, && \mathbf{J}^{fn}_{3,6} = b^{3,1} + b^{3,2} + b^{3,3}, \\ &\mathbf{J}^{fn}_{4,1} = c^{1,1} + c^{1,2} + c^{1,3}, && \mathbf{J}^{fn}_{5,2} = c^{2,1} + c^{2,2} + c^{2,3}, && \mathbf{J}^{fn}_{6,3} = c^{3,1} + c^{3,2} + c^{3,3}, \\ &\mathbf{J}^{fn}_{4,4} = d^{1,1} + d^{1,2} + d^{1,3}, && \mathbf{J}^{fn}_{5,5} = d^{2,1} + d^{2,2} + d^{2,3}, && \mathbf{J}^{fn}_{6,6} = d^{3,1} + d^{3,2} + d^{3,3}. \end{aligned} \quad (3.27)$$

By using Eq. (3.24), remaining elements of the force Jacobian matrix are calculated as follows,

$$
\begin{array}{lll}
\mathbf{J}^{fn}_{1,2} = -a^{1,2}, & \mathbf{J}^{fn}_{1,3} = -a^{1,3}, & \mathbf{J}^{fn}_{2,1} = -a^{2,1}, \\
\mathbf{J}^{fn}_{1,5} = -b^{1,2}, & \mathbf{J}^{fn}_{1,6} = -b^{1,3}, & \mathbf{J}^{fn}_{2,4} = -b^{2,1}, \\
\mathbf{J}^{fn}_{4,2} = -c^{1,2}, & \mathbf{J}^{fn}_{4,3} = -c^{1,3}, & \mathbf{J}^{fn}_{5,1} = -c^{2,1}, \\
\mathbf{J}^{fn}_{4,5} = -d^{1,2}, & \mathbf{J}^{fn}_{4,6} = -d^{1,3}, & \mathbf{J}^{fn}_{5,4} = -d^{2,1}, \\
\mathbf{J}^{fn}_{2,3} = -a^{2,3}, & \mathbf{J}^{fn}_{3,1} = -a^{3,1}, & \mathbf{J}^{fn}_{3,2} = -a^{3,2}, \\
\mathbf{J}^{fn}_{2,6} = -b^{2,3}, & \mathbf{J}^{fn}_{3,4} = -b^{3,1}, & \mathbf{J}^{fn}_{3,6} = -b^{3,2}, \\
\mathbf{J}^{fn}_{5,3} = -c^{2,3}, & \mathbf{J}^{fn}_{6,1} = -c^{3,1}, & \mathbf{J}^{fn}_{6,2} = -c^{3,2}, \\
\mathbf{J}^{fn}_{5,6} = -d^{2,3}, & \mathbf{J}^{fn}_{6,4} = -d^{3,1}, & \mathbf{J}^{fn}_{6,5} = -d^{3,2}.
\end{array}
\tag{3.28}
$$

The element in $\mathbf{T}^{ji}$ matrices are replaced with the elements in $\mathbf{T}^{ij}$ since they are equal and $\mathbf{T}^{2,2} = \mathbf{T}^{3,3} = 0$. After all the necessary replacement is made in Eq. (3.27) and Eq. (3.28), the force Jacobian matrix for 3-DOF lumped model with four nonlinear elements is obtained by performing the assembly rule and given in Eq. (3.29).

$$
\mathbf{J}^{fn} = \left[\begin{array}{ccc|ccc}
a^{1,1} + a^{1,2} + a^{1,3} & -a^{1,2} & -a^{1,3} & b^{1,1} + b^{1,2} + b^{1,3} & -b^{1,2} & -b^{1,3} \\
-a^{1,2} & a^{1,2} + a^{2,3} & -a^{2,3} & -b^{1,2} & b^{1,2} + b^{2,3} & -b^{2,3} \\
-a^{1,3} & -a^{2,3} & a^{1,3} + a^{2,3} & -b^{1,3} & -b^{2,3} & b^{1,3} + b^{2,3} \\
\hline
c^{1,1} + c^{1,2} + c^{1,3} & -c^{1,2} & -c^{1,3} & d^{1,1} + d^{1,2} + d^{1,3} & -d^{1,2} & -d^{1,3} \\
-c^{1,2} & c^{1,2} + c^{2,3} & -c^{2,3} & -d^{1,2} & d^{1,2} + d^{2,3} & -d^{2,3} \\
-c^{1,3} & -c^{2,3} & c^{1,3} + c^{2,3} & -d^{1,3} & -d^{2,3} & d^{1,3} + d^{2,3}
\end{array}\right]
\tag{3.29}
$$

After finding $\mathbf{J}^{fn}$, global Jacobian matrix $\mathbf{J}$ is calculated by using Eq. (2.32).

3.4. Summary of the Proposed Method (JEM)

In order to clarify Jacobian Element Method (JEM), calculation methodology is given in Figure 3.2. In the first step, all reduced elemental Jacobian matrices have to be calculated for each nonlinear element. In the second step, assembly rule to obtain force Jacobian matrix has to be used. In the final step, global Jacobian matrix is calculated by using equation given in Figure 3.2.

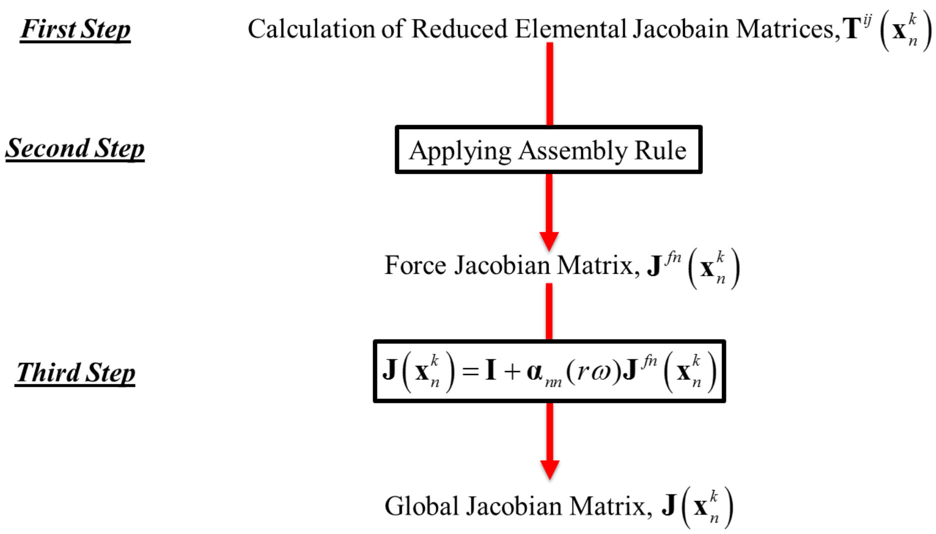

Figure 3.2. Summary of the proposed method

3.5. Theoretical Comparison of Computation Effort of Classical Jacobian and Jacobian Element Method

If computational effort is examined for 3-DOF lumped model, the main advantage of Jacobian element method according to classical Jacobian method is that while 36 computations are made by classical Jacobian method, 16 calculations are made by Jacobian element method.

For multi-harmonic balance method, in order to obtain Global Jacobian matrix by using the classical Jacobian method, $(2n_d N_h n)^2$ calculation has to be done however, $(2N_h n_d)^2 n_e$ calculation is enough when using the Jacobian element method. n_e is the total number of nonlinear element connected to different nodes. Percentage of the computational reduction ratio for multi-harmonic balance method between two methods can be determined by using Eq. (3.30).

$$RR = 100\cdot\left(1-\frac{(2N_h n_d)^2 n_e}{(2n_d N_h n)^2}\right) = 100\cdot\left(1-\frac{n_e}{n^2}\right) \tag{3.30}$$

One can understand that there is no effect of harmonic number used in analysis over the computational reduction ratio. It can clearly be seen from Eq. (3.30). The increase in the number of harmonic used in analysis increases the total number of nonlinear differential equations. Although, the computational reduction rate does not change with the harmonic number, Jacobian element method provides a significant reduction in computational time.

CHAPTER 4

CASE STUDIES ON JACOBIAN ELEMENT METHOD

4.1. Three Different Lumped Models

Three different lumped models are created to examine the effect of the number of harmonics and the number of nonlinear elements used in the models on the computational time. Using the proposed method and classical Jacobian method, the obtained solutions are compared in terms of the computational time and the computational reduction ratio while the number of iterations made in solution process for each model are kept close to each other. The calculations are done on a computer having a processor Intel Core i7-3630QM CPU @ 2.40 GHz with 16,00 GB of RAM.

In order to have equally weighted nonlinearity between DOFs, cubic stiffness is used for nonlinear element in the system. The response of the 4-DOF systems are obtained by using single and multi-harmonic balance method. The response of the 4-DOF systems are studied by keeping the excitation force amplitude constant at 40N. It is assumed that the system has proportional damping with a damping ratio 0.1%.

Three lumped model systems with the following properties are given as

$$m_{1,2,3,4} = 1kg, \quad k_{1,2,3,4,5} = 2500\,N/m, \quad \eta = 0.01, \quad k_c = 1000\,N/m, \quad F = 40\,N.$$

Investigated Model-1, Model-2 and Model-3 are given in Figure 4.1, Figure 4.2 and Figure 4.3, respectively. In order to simulate worst case scenario in terms of computational time where each dof is connected to other dofs and ground with a nonlinear element, 4-DOF lumped model (Model-1) with ten cubic stiffness elements is used, which is given in Figure 4.1. In Model-2 and Model-3, five and two cubic stiffness elements are used respectively.

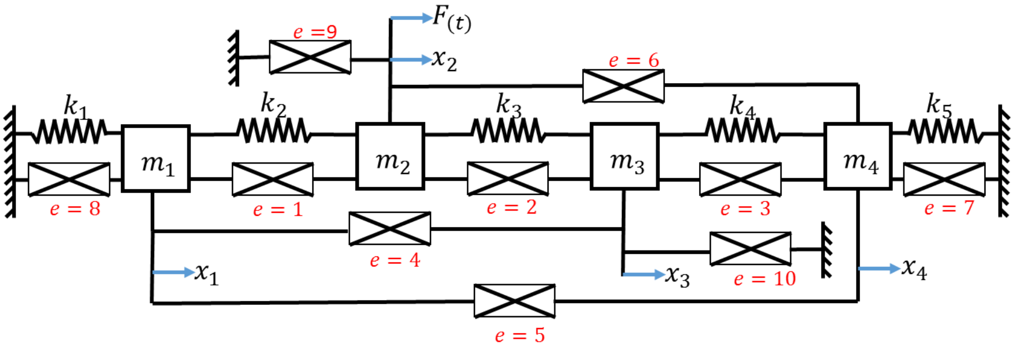

Figure 4.1. 4- DOF lumped model with ten cubic stiffness elements (Model-1)

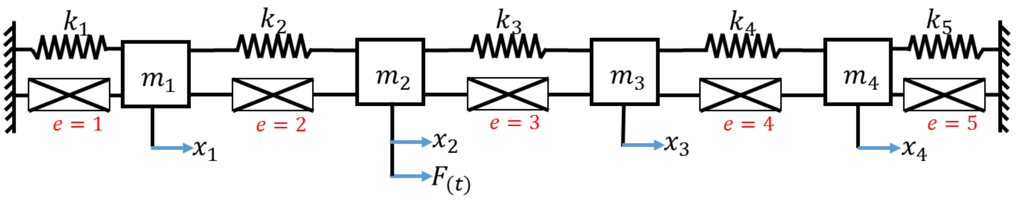

Figure 4.2. 4- DOF lumped model with five cubic stiffness elements (Model-2)

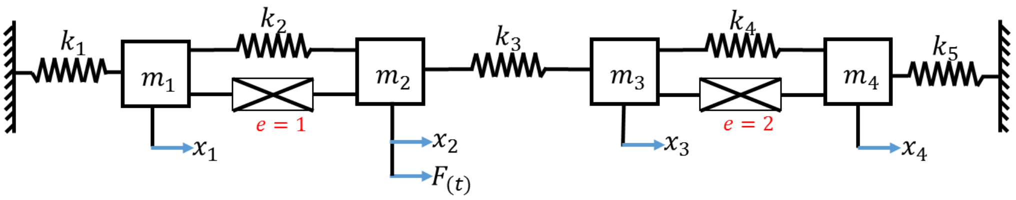

Figure 4.3. 4- DOF lumped model with two cubic stiffness elements (Model-3)

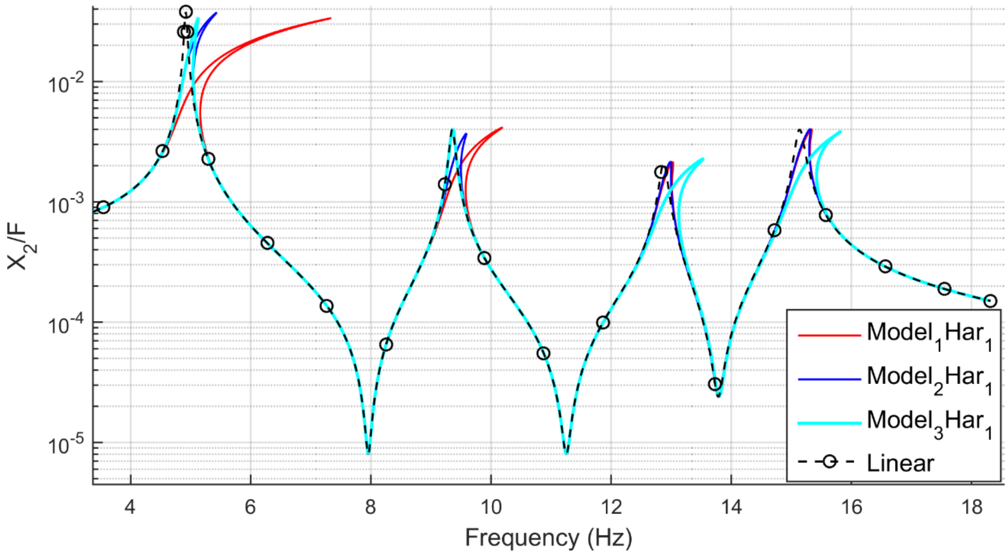

Figure 4.4. Receptances of Model-1,2,3 for single harmonic solution

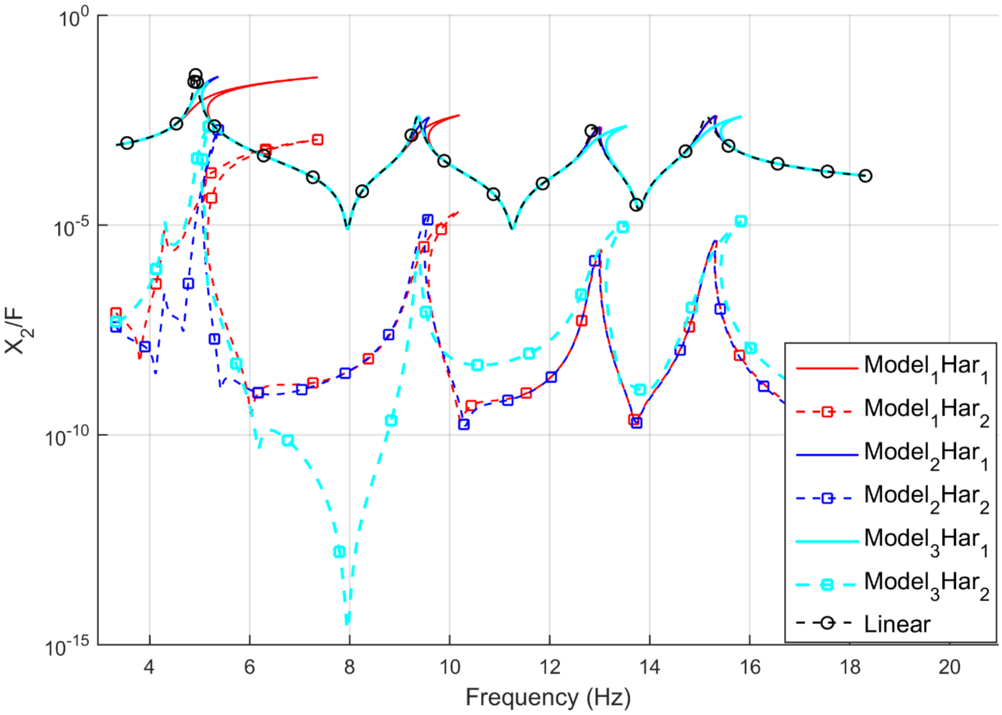

Figure 4.5. Receptances of Model-1,2,3 for two harmonics solution

Receptances obtained by using single and two harmonics are shown in Figure 4.4 and Figure 4.5 respectively. The effect of cubic stiffness nonlinearity can be seen clearly from Figure 4.4 and Figure 4.5 by observing the resonance regions where the stiffening effect occurs for all three models.

Table 4.1. *Computational time comparison of force Jacobian matrix with and without using Jacobian element method (JEM) for single harmonic solution of the lumped models*

	Nonlin. element #	**Nonlin. equation #**	**J^{fn} comp. time (s) with/ without JEM**	**J^{fn} actual / expected reduction ratio (%)**	**Total solution reduction ratio (%)**
Model-1	10		7.92 / 12.36	35.92 / 37.50	23.94
Model-2	5	8	3.38 / 10.35	67.34 / 68.75	42.84
Model-3	2		1.45 / 8.16	82.23 / 84.38	56.58

Table 4.2. *Computational time comparison of force Jacobian matrix with and without using Jacobian element method (JEM) for two harmonics solution of the lumped models*

	Nonlin. element #	**Nonlin. equation #**	**J^{fn} comp. time (s) with/ without JEM**	**J^{fn} actual / expected reduction ratio (%)**	**Total solution reduction ratio (%)**
Model-1	10		26.92 / 42.35	36.43 / 37.50	27.56
Model-2	5	16	10.75 / 32.76	67.19 / 68.75	49.76
Model-3	2		3.96 / 23.43	83.10 / 84.38	61.46

Table 4.1 and Table 4.2 show the reduction ratio in computational time required to solve the lumped models. It can be concluded that the computational time of force Jacobian matrix decreases by using Jacobian Element Method compared to classical Jacobian Method. Expected reduction ratio is calculated from Eq. (3.30) and actual

reduction ratio is taken from the solution time of MATLAB process. It is verified that there is a good agreement between actual and expected reduction ratios. Moreover, it can be concluded that calculation time of $\mathbf{J}^{fn}$ is independent of the number of harmonics by comparing Table 4.1 and Table 4.2. Total solution reduction ratio is the ratio of total time required to solve the nonlinear system equations by using JEM to time by using classical Jacobian. The other fact is that although the number of nonlinear equations are same, decreasing the number of nonlinear elements in a higher total solution reduction ratio.

4.2. Interpretation of the Worst Case Scenario for Lumped Parameter Model

For the worst case scenario, maximum possible number of nonlinear elements and relation between number of nonlinear DOFs are given as follows,

$$n_e = C(n,2) + n \rightarrow \frac{n^2 + n}{2} \tag{4.1}$$

There is a nonlinear element between each dof and each dof is attached to the ground with a nonlinear element.

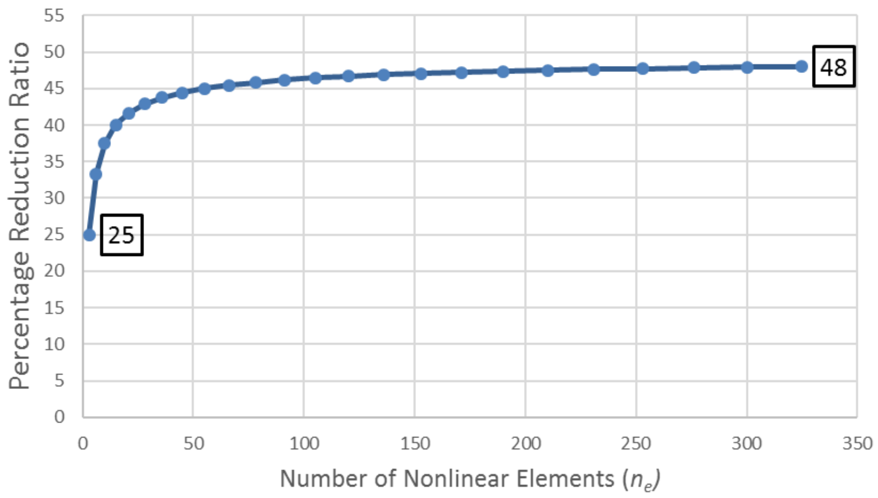

Figure 4.6. Percentage reduction ratio in computational time of Jacobian calculation for the worst case scenario for lumped parameter model

As you can see in Figure 4.6, If Jacobian element method is used for worst case scenario, minimum twenty five percent reduction is obtained and max reduction ratio converges to fifty percent.

4.3. Realistic Finite Element Model

Aerodynamic forces are major excitation sources in turbomachinery and bladed disk systems The airflow passing through these systems results in flow induced vibrations and due to its nature this type of excitation causes blades to vibrate at their cantilever beam modes. Therefore, the finite element model with two shrouded blade sectors analyzed is given at Figure 4.7, in which *A* and *A'* are the points, where harmonic excitation forces applied in $+y$ and $-y$ direction, respectively. *B* and *B'* are the contact surfaces, where nonlinear elements are placed. *C* and *C'* are the surfaces, where fixed boundary condition applied. Natural frequencies and mode shapes of the system are obtained by finite element software ABAQUS. By using modal information, α the receptance of the linear part of the system is calculated by using first six natural frequencies and mode shapes. It is assumed that the system has proportional damping with a damping ratio 0.1%.

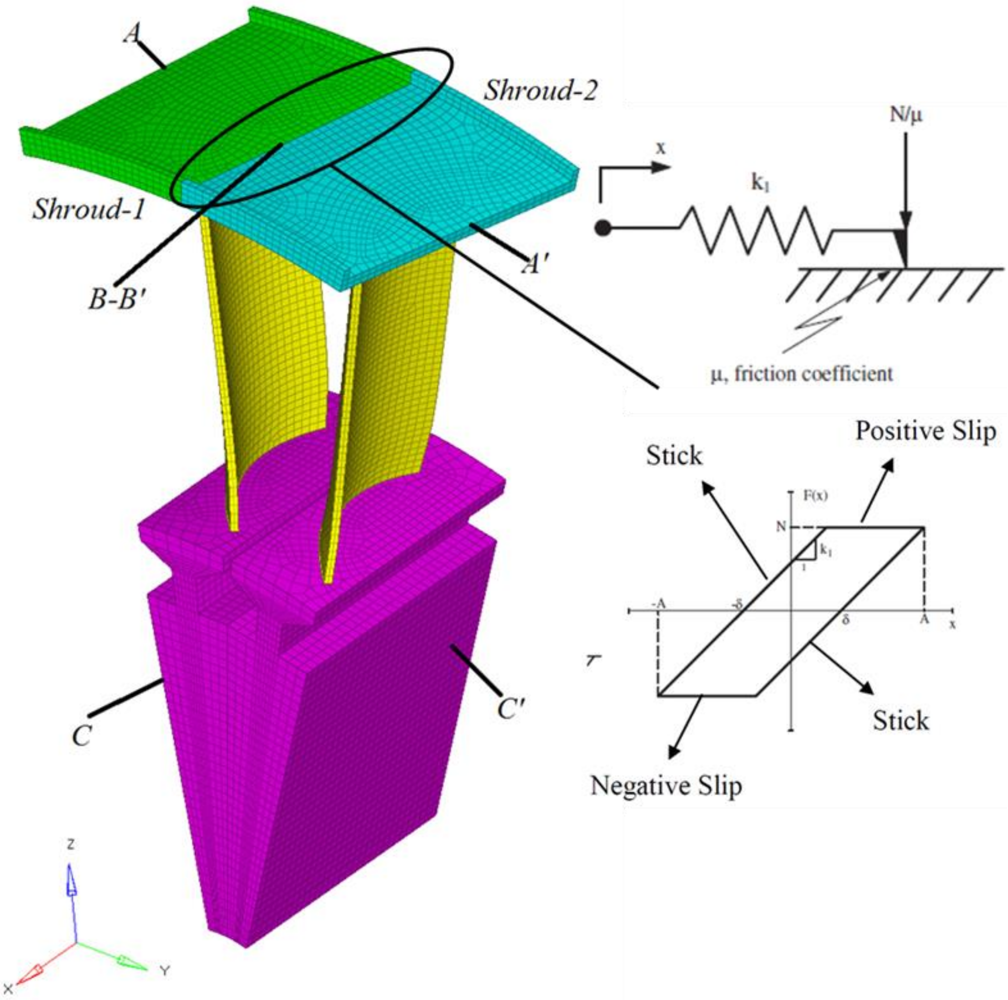

Figure 4.7. Finite element model for the blade and Coulomb macroslip model between Shroud-1 and Shroud-2

Between the nodes located on the contact surface of *Shroud 1* and *Shroud 2*, different number of nonlinear friction elements are connected to each other by node to node in order to investigate the effect of it on computational time and corresponding results are tabulated in Table 4.3.

The force response of the system where four nonlinear elements are placed between shrouds are analyzed and receptance of point *A* to *B* is given in Figure 4.8. For the specified system, the single harmonic response of finite element model is studied by changing the sliding friction force, μN, between 40 and 600 N and applying the constant amplitude excitation force at 400N.

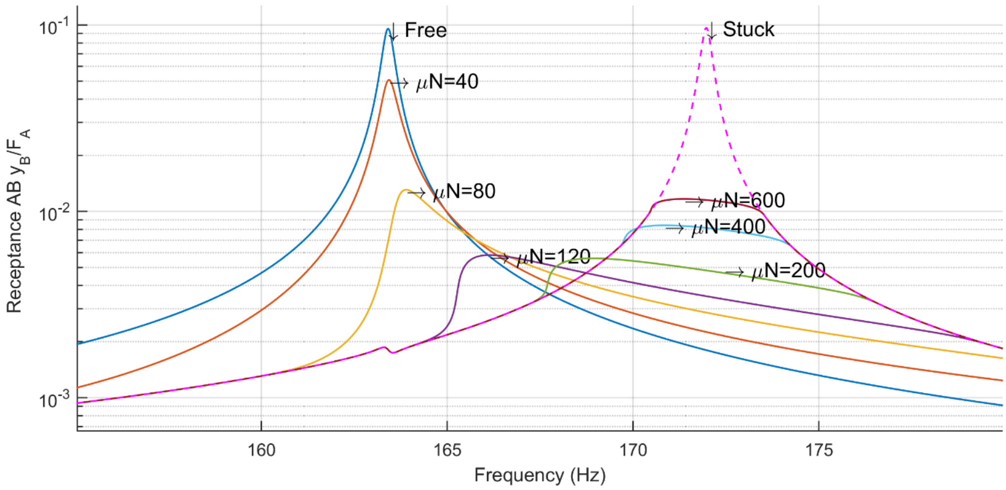

Figure 4.8. Receptance of the system with four nonlinear friction elements for different sliding forces

When the results are inspected from Figure 4.8, the resonance frequency of the system increases as the sliding friction force increases. Moreover, resonance amplitude first decreases down to a certain amplitude then increases with the increasing sliding friction force. In other words, there exists an optimum value which minimizes the resonance amplitude which is 200 N sliding friction force for this specific case. As a result, the resonance amplitude of the bladed disks can be minimized by adjusting the normal load acting on the friction damper to the optimum value. Furthermore, the system becomes completely stuck at higher sliding forces. In completely stuck case, there is no relative displacement between the shrouds, hence the effect of frictional damping is lost which results in high resonance amplitudes.

Table 4.3. *Computational time comparison of force Jacobian matrix with and without using Jacobian element method (JEM) for single harmonic solution of the bladed disk model*

Nonlin. equation #	Nonlin. element #	Nonlin. DOF #	J^{fn} comp. time (s) with/ without JEM	J^{fn} actual / expected reduction ratio (%)	Total solution reduction ratio (%)
8	2	4	4.96 / 31.68	84.34 / 87.5	56.59
16	4	8	9.14 / 126.72	92.78 / 93.75	70.58
32	8	16	16.61 / 506.88	96.72 / 96.88	87.04
64	16	32	34.63/2027.52	98.29 / 98.44	93.22

The developed method is applied on large FEM of a bladed disk. In the FEM, friction nonlinearity is used between the shroud contact surfaces. The effect of different numbers of nonlinear elements and nonlinear equation numbers on computational time are observed. Theoretical and actual computational reduction ratios are also presented and the results are in good agreement. It is concluded that there is a significant reduction in the computational time with increasing number of nonlinear elements more than 90%.

4.4. Interpretation of the Worst Case Scenario for Realistic Finite Element Model with 1D-Dry Friction Nonlinearity

Maximum possible number of nonlinear elements is equal to number of nonlinear DOFs on one contact surface of shroud. This relation is given as follows,

$$\text{For node to node connection, } n_e = n/2. \tag{4.2}$$

As you can see in Figure 4.9, if Jacobian element method is used for contact nonlinearity, minimum seventy five percent reduction in Jacobian calculation is obtained and max reduction ratio converges to one hundred percent.

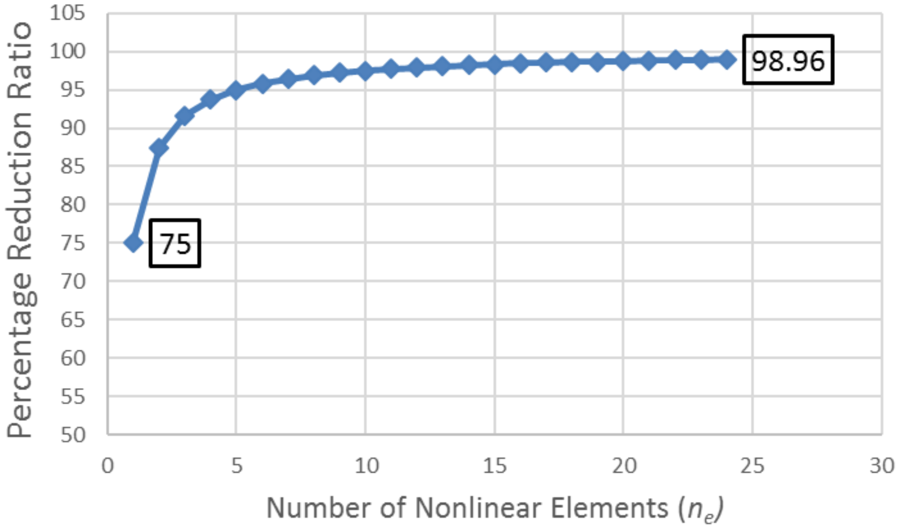

Figure 4.9. Percentage reduction ratio in computational time of Jacobian calculation for contact nonlinearity case

CHAPTER 5

CASE STUDIES ON DIFFERENT NONLINEAR EQUATION SOLVERS AND PREDICTORS

5.1. Lumped Model for Different Nonlinear Solvers and Predictors

Lumped model with cubic stiffness nonlinearity is examined in order to compare the performance of the different nonlinear equation solvers and predictors.

The effect of arc-length parameter and error criterion on the computational time and iteration numbers are investigated for different parameters.

The response of the 4-DOF systems are obtained by using single harmonic balance method. The response of the 4-DOF systems are studied by keeping the excitation force amplitude constant at 1500N. It is assumed that the system has proportional damping with a damping ratio 0.1%.

Three lumped model systems (4-DOF) with the following properties are given as

$$m_{1,2,3,4} = 1kg, \quad k_{1,2,3,4,5} = 25000\,N/m, \quad \eta = 0.01, \quad k_c = 1500\,N/m, \quad F = 1500\,N.$$

Investigated lumped model with five cubic stiffness element is given in Figure 5.1 and the solution obtained is given in .

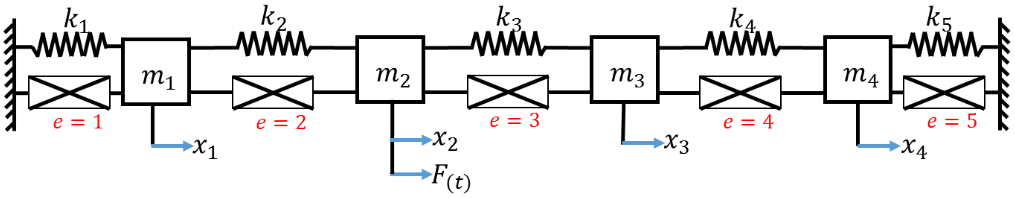

Figure 5.1. 4- DOF lumped model with five cubic stiffness elements

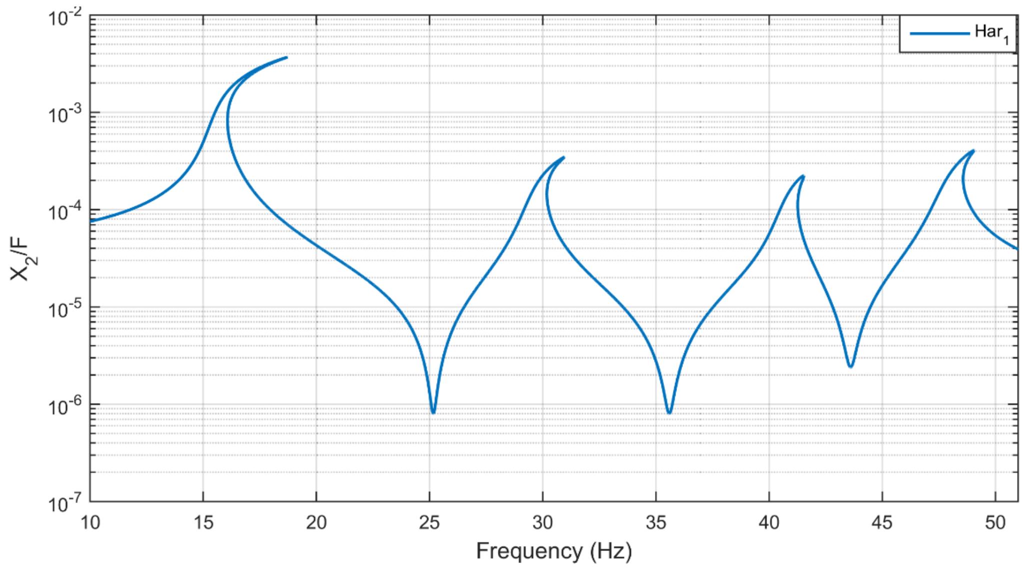

Figure 5.2. Receptance of the lumped model with five cubic stiffness elements

Firstly, performance comparison of different algebraic nonlinear equation solvers with different constant arc-length parameters are studied in Table 5.1 and Table 5.2.

Secondly, solver performance are compared by using bypass method. It provides to avoid unnecessary calculations for solvers other than Newton's method. Bypass method is applicable to solvers which use Newton's method at the first step. If the solution of the first step calculation satisfies the error criterion, additional vector and Jacobian calculations for other steps are not made. Performance of the solvers with bypass method is given in Table 5.3 and Table 5.4.Thirdly, effect of adaptive arc-length algorithm, which is given in Eq. (2.47), is studied for different optimum iteration numbers, $n_{opt}^{iter} = 1, 1.5, 2.$ and $s_{\max} = 0.32$, $s_{\min} = 0.04$. The corresponding results are given in Table 5.5 and Table 5.6. Finally, effect of adaptive arc-length algorithm and bypass are investigated together. Comparison results are given in Table 5.7 and Table 5.8.

Cubic stiffness is a continuous nonlinearity which means it is effective all frequency range of analysis. In addition, it leads to sharp turning points in the solutions. When

Figure 5.2 is examined, it is understood that the obtained solution has sharp turning points. This is specifically chosen to further challenge the solvers.

The cells marked in red in the flag column of the tables indicate that the solution could not be obtained and there is a convergence problem. The cells marked in red in the time column of the tables indicate the longest computational time. If it is marked in green, it is the fastest computational time.

Table 5.1. Performance comparison of different algebraic nonlinear equation solvers with different constant arc-length parameters for cubic stiffness nonlinearity (Error Criterion=1e-6, 1e-8)

	Iter. #	Time(s)	s	Error Crit.	Flag	Bypass		Iter. #	Time(s)	s	Error Crit.	Flag	Bypass
Newton	950	1.53	0.32	1.00E-06	1	0	Newton	1192	1.81	0.32	1.00E-08	1	0
Solver-1	124	0.28	0.32	1.00E-06	1	0	Solver-1	96	0.18	0.32	1.00E-08	1	0
Solver-2	124	0.47	0.32	1.00E-06	1	0	Solver-2	96	0.29	0.32	1.00E-08	1	0
Solver-3	124	0.31	0.32	1.00E-06	1	0	Solver-3	96	0.18	0.32	1.00E-08	1	0
Solver-4	124	0.46	0.32	1.00E-06	1	0	Solver-4	96	0.29	0.32	1.00E-08	1	0
Solver-5	1085	3.38	0.32	1.00E-06	1	0	Solver-5	1259	3.79	0.32	1.00E-08	1	0
Solver-6	1083	3.32	0.32	1.00E-06	1	0	Solver-6	1252	3.72	0.32	1.00E-08	1	0
Solver-7	811	2.5	0.32	1.00E-06	1	0	Solver-7	1253	3.67	0.32	1.00E-08	1	0
	Iter. #	Time(s)	s	Error Crit.	Flag	Bypass		Iter. #	Time(s)	s	Error Crit.	Flag	Bypass
Newton	2285	3.64	0.16	1.00E-06	1	0	Newton	2666	4.11	0.16	1.00E-08	1	0
Solver-1	250	0.47	0.16	1.00E-06	1	0	Solver-1	209	0.39	0.16	1.00E-08	1	0
Solver-2	250	0.78	0.16	1.00E-06	1	0	Solver-2	209	0.65	0.16	1.00E-08	1	0
Solver-3	250	0.49	0.16	1.00E-06	1	0	Solver-3	209	0.41	0.16	1.00E-08	1	0
Solver-4	250	0.79	0.16	1.00E-06	1	0	Solver-4	209	0.65	0.16	1.00E-08	1	0
Solver-5	1325	4.05	0.16	1.00E-06	1	0	Solver-5	1688	5.01	0.16	1.00E-08	1	0
Solver-6	1327	4.08	0.16	1.00E-06	1	0	Solver-6	1686	5.01	0.16	1.00E-08	1	0
Solver-7	1839	5.64	0.16	1.00E-06	1	0	Solver-7	1690	5.02	0.16	1.00E-08	1	0
	Iter. #	Time(s)	s	Error Crit.	Flag	Bypass		Iter. #	Time(s)	s	Error Crit.	Flag	Bypass
Newton	5747	9.27	0.08	1.00E-06	0	0	Newton	6506	10.27	0.08	1.00E-08	0	0
Solver-1	771	1.43	0.08	1.00E-06	1	0	Solver-1	448	0.83	0.08	1.00E-08	1	0
Solver-2	771	2.42	0.08	1.00E-06	1	0	Solver-2	448	1.4	0.08	1.00E-08	1	0
Solver-3	771	1.53	0.08	1.00E-06	1	0	Solver-3	448	0.88	0.08	1.00E-08	1	0
Solver-4	771	2.43	0.08	1.00E-06	1	0	Solver-4	448	1.39	0.08	1.00E-08	1	0
Solver-5	4025	12.52	0.08	1.00E-06	1	0	Solver-5	4837	14.87	0.08	1.00E-08	1	0
Solver-6	4026	12.5	0.08	1.00E-06	1	0	Solver-6	1686	5.01	0.08	1.00E-08	1	0
Solver-7	4026	12.52	0.08	1.00E-06	1	0	Solver-7	1690	5.02	0.08	1.00E-08	1	0
	Iter. #	Time(s)	s	Error Crit.	Flag	Bypass		Iter. #	Time(s)	s	Error Crit.	Flag	Bypass
Newton	9990	16.75	0.04	1.00E-06	0	0	Newton	11884	19.23	0.04	1.00E-08	0	0
Solver-1	2047	3.78	0.04	1.00E-06	1	0	Solver-1	1471	2.71	0.04	1.00E-08	1	0
Solver-2	2047	6.7	0.04	1.00E-06	1	0	Solver-2	1471	4.61	0.04	1.00E-08	1	0
Solver-3	2047	4.32	0.04	1.00E-06	1	0	Solver-3	1471	2.92	0.04	1.00E-08	1	0
Solver-4	2047	6.52	0.04	1.00E-06	1	0	Solver-4	1470	4.63	0.04	1.00E-08	1	0
Solver-5	8861	27.72	0.04	1.00E-06	0	0	Solver-5	9952	30.54	0.04	1.00E-08	0	0
Solver-6	8860	27.6	0.04	1.00E-06	0	0	Solver-6	10019	30.92	0.04	1.00E-08	0	0
Solver-7	8861	27.48	0.04	1.00E-06	0	0	Solver-7	10086	31.07	0.04	1.00E-08	0	0

Table 5.2. *Performance comparison of different algebraic nonlinear equation solvers with different constant arc-length parameters for cubic stiffness nonlinearity (Error Criterion=1e-10, 1e-12)*

	Iter. #	Time(s)	s	Error Crit.	Flag	Bypass		Iter. #	Time(s)	s	Error Crit.	Flag
Newton	1402	2.06	0.32	1.00E-10	1	0	Newton	164	0.26	0.32	1.00E-12	1
Solver-1	36	0.07	0.32	1.00E-10	1	0	Solver-1	22	0.04	0.32	1.00E-12	1
Solver-2	36	0.11	0.32	1.00E-10	1	0	Solver-2	22	0.06	0.32	1.00E-12	1
Solver-3	36	0.07	0.32	1.00E-10	1	0	Solver-3	22	0.04	0.32	1.00E-12	1
Solver-4	36	0.11	0.32	1.00E-10	1	0	Solver-4	22	0.06	0.32	1.00E-12	1
Solver-5	1690	4.88	0.32	1.00E-10	1	0	Solver-5	390	1.08	0.32	1.00E-12	1
Solver-6	1685	4.9	0.32	1.00E-10	1	0	Solver-6	159	0.45	0.32	1.00E-12	1
Solver-7	1693	4.89	0.32	1.00E-10	1	0	Solver-7	330	0.93	0.32	1.00E-12	1
	Iter. #	Time(s)	s	Error Crit.	Flag	Bypass		Iter. #	Time(s)	s	Error Crit.	Flag
Newton	3559	5.37	0.16	1.00E-10	1	0	Newton	519	0.75	0.16	1.00E-12	1
Solver-1	133	0.25	0.16	1.00E-10	1	0	Solver-1	22	0.04	0.16	1.00E-12	1
Solver-2	133	0.42	0.16	1.00E-10	1	0	Solver-2	22	0.06	0.16	1.00E-12	1
Solver-3	133	0.28	0.16	1.00E-10	1	0	Solver-3	22	0.04	0.16	1.00E-12	1
Solver-4	133	0.42	0.16	1.00E-10	1	0	Solver-4	22	0.06	0.16	1.00E-12	1
Solver-5	2210	6.47	0.16	1.00E-10	1	0	Solver-5	493	1.43	0.16	1.00E-12	1
Solver-6	2225	6.44	0.16	1.00E-10	1	0	Solver-6	238	0.68	0.16	1.00E-12	1
Solver-7	2263	6.56	0.16	1.00E-10	1	0	Solver-7	468	1.29	0.16	1.00E-12	1
	Iter. #	Time(s)	s	Error Crit.	Flag	Bypass		Iter. #	Time(s)	s	Error Crit.	Flag
Newton	7902	11.99	0.08	1.00E-10	0	0	Newton	664	0.99	0.08	1.00E-12	1
Solver-1	348	0.64	0.08	1.00E-10	1	0	Solver-1	161	0.29	0.08	1.00E-12	1
Solver-2	348	1.08	0.08	1.00E-10	1	0	Solver-2	151	0.46	0.08	1.00E-12	1
Solver-3	348	0.68	0.08	1.00E-10	1	0	Solver-3	110	0.21	0.08	1.00E-12	1
Solver-4	348	1.09	0.08	1.00E-10	1	0	Solver-4	156	0.48	0.08	1.00E-12	1
Solver-5	5960	17.66	0.08	1.00E-10	1	0	Solver-5	727	2.08	0.08	1.00E-12	1
Solver-6	6025	17.83	0.08	1.00E-10	1	0	Solver-6	1075	3.06	0.08	1.00E-12	1
Solver-7	6057	17.85	0.08	1.00E-10	1	0	Solver-7	1112	3.13	0.08	1.00E-12	1
	Iter. #	Time(s)	s	Error Crit.	Flag	Bypass		Iter. #	Time(s)	s	Error Crit.	Flag
Newton	13883	21.49	0.04	1.00E-10	0	0	Newton	1439	2.12	0.04	1.00E-12	1
Solver-1	808	1.49	0.04	1.00E-10	1	0	Solver-1	602	1.06	0.04	1.00E-12	1
Solver-2	808	2.53	0.04	1.00E-10	1	0	Solver-2	596	1.82	0.04	1.00E-12	1
Solver-3	808	1.6	0.04	1.00E-10	1	0	Solver-3	322	0.62	0.04	1.00E-12	1
Solver-4	808	2.52	0.04	1.00E-10	1	0	Solver-4	599	1.83	0.04	1.00E-12	1
Solver-5	12069	36.29	0.04	1.00E-10	0	0	Solver-5	1576	4.52	0.04	1.00E-12	1
Solver-6	12085	36.31	0.04	1.00E-10	0	0	Solver-6	507	1.49	0.04	1.00E-12	1
Solver-7	12237	36.64	0.04	1.00E-10	0	0	Solver-7	1838	5.21	0.04	1.00E-12	1

Table 5.1 and Table 5.2 shows that convergence problems are experienced with increasing arc-length parameters. In addition, all solvers is not able to obtain solutions at low absolute error criteria. In some cases where other solvers had convergence problems, the solution was obtained by Newton method. It shows that convergence domain of the solvers other than Newton's is narrow.

Table 5.3. *Performance comparison of different algebraic nonlinear equation solvers with different constant arc-length parameters and passing unnecessary calculations for cubic stiffness nonlinearity (Error Criterion=1e-6, 1e-8)*

	Iter. #	Time(s)	s	Error Crit.	Flag	Bypass		Iter. #	Time(s)	s	Error Crit.	Flag	Bypass
Newton	950	1.51	0.32	1.00E-06	1	1	Newton	1192	1.79	0.32	1.00E-08	1	1
Solver-1	286	0.54	0.32	1.00E-06	1	1	Solver-1	176	0.31	0.32	1.00E-08	1	1
Solver-2	124	0.51	0.32	1.00E-06	1	1	Solver-2	96	0.3	0.32	1.00E-08	1	1
Solver-3	284	0.59	0.32	1.00E-06	1	1	Solver-3	176	0.32	0.32	1.00E-08	1	1
Solver-4	449	1.39	0.32	1.00E-06	1	1	Solver-4	323	0.96	0.32	1.00E-08	1	1
Solver-5	1086	2.54	0.32	1.00E-06	1	1	Solver-5	1257	3.11	0.32	1.00E-08	1	1
Solver-6	1087	2.53	0.32	1.00E-06	1	1	Solver-6	1255	3.12	0.32	1.00E-08	1	1
Solver-7	811	1.91	0.32	1.00E-06	1	1	Solver-7	1257	3.1	0.32	1.00E-08	1	1
	Iter. #	Time(s)	s	Error Crit.	Flag	Bypass		Iter. #	Time(s)	s	Error Crit.	Flag	Bypass
Newton	2285	3.63	0.16	1.00E-06	1	1	Newton	2666	4.19	0.16	1.00E-08	1	1
Solver-1	448	0.81	0.16	1.00E-06	1	1	Solver-1	566	1.03	0.16	1.00E-08	1	1
Solver-2	250	0.78	0.16	1.00E-06	1	1	Solver-2	209	0.66	0.16	1.00E-08	1	1
Solver-3	443	0.83	0.16	1.00E-06	1	1	Solver-3	561	1.03	0.16	1.00E-08	1	1
Solver-4	1251	3.87	0.16	1.00E-06	1	1	Solver-4	601	1.82	0.16	1.00E-08	1	1
Solver-5	1327	3.06	0.16	1.00E-06	1	1	Solver-5	1686	3.91	0.16	1.00E-08	1	1
Solver-6	1327	3.07	0.16	1.00E-06	1	1	Solver-6	1687	3.89	0.16	1.00E-08	1	1
Solver-7	2492	5.63	0.16	1.00E-06	0	1	Solver-7	1690	3.89	0.16	1.00E-08	1	1
	Iter. #	Time(s)	s	Error Crit.	Flag	Bypass		Iter. #	Time(s)	s	Error Crit.	Flag	Bypass
Newton	5747	9.34	0.08	1.00E-06	0	1	Newton	6506	10.34	0.08	1.00E-08	0	1
Solver-1	1087	2.04	0.08	1.00E-06	1	1	Solver-1	1148	1.98	0.08	1.00E-08	1	1
Solver-2	771	2.42	0.08	1.00E-06	1	1	Solver-2	448	1.47	0.08	1.00E-08	1	1
Solver-3	1087	2.09	0.08	1.00E-06	1	1	Solver-3	1102	1.98	0.08	1.00E-08	1	1
Solver-4	2286	7.15	0.08	1.00E-06	1	1	Solver-4	1150	3.5	0.08	1.00E-08	1	1
Solver-5	4025	9.11	0.08	1.00E-06	1	1	Solver-5	4835	11.05	0.08	1.00E-08	1	1
Solver-6	4026	9.22	0.08	1.00E-06	1	1	Solver-6	1687	3.89	0.08	1.00E-08	1	1
Solver-7	4026	9.07	0.08	1.00E-06	1	1	Solver-7	1690	3.89	0.08	1.00E-08	1	1
	Iter. #	Time(s)	s	Error Crit.	Flag	Bypass		Iter. #	Time(s)	s	Error Crit.	Flag	Bypass
Newton	9990	16.81	0.04	1.00E-06	0	1	Newton	11884	19.35	0.04	1.00E-08	0	1
Solver-1	4411	8.22	0.04	1.00E-06	1	1	Solver-1	2291	4.16	0.04	1.00E-08	1	1
Solver-2	2047	6.42	0.04	1.00E-06	1	1	Solver-2	1471	4.66	0.04	1.00E-08	1	1
Solver-3	4403	8.34	0.04	1.00E-06	1	1	Solver-3	2291	4.35	0.04	1.00E-08	1	1
Solver-4	4540	14.31	0.04	1.00E-06	1	1	Solver-4	4752	14.85	0.04	1.00E-08	1	1
Solver-5	8860	17.99	0.04	1.00E-06	0	1	Solver-5	9977	22.07	0.04	1.00E-08	0	1
Solver-6	8860	18.14	0.04	1.00E-06	0	1	Solver-6	10000	22.38	0.04	1.00E-08	0	1
Solver-7	8861	18.17	0.04	1.00E-06	0	1	Solver-7	10092	22.39	0.04	1.00E-08	0	1

Troubles encountered in Table 5.1 and Table 5.2 can not be resolved despite use of bypass. It can be seen from Table 5.3 and Table 5.4. However, the computational times of solver-5,6,7 are fallen by almost half since additional calculations are avoided.

Table 5.4. *Performance comparison of different algebraic nonlinear equation solvers with different constant arc-length parameters and passing unnecessary calculations for cubic stiffness nonlinearity (Error Criterion=1e-10, 1e-12)*

	Iter. #	Time(s)	s	Error Crit.	Flag	Bypass		Iter. #	Time(s)	s	Error Crit.	Flag	Bypass
Newton	1402	2.05	0.32	1.00E-10	1	1	Newton	164	0.24	0.32	1.00E-12	1	1
Solver-1	222	0.37	0.32	1.00E-10	1	1	Solver-1	223	0.36	0.32	1.00E-12	1	1
Solver-2	36	0.11	0.32	1.00E-10	1	1	Solver-2	22	0.06	0.32	1.00E-12	1	1
Solver-3	218	0.38	0.32	1.00E-10	1	1	Solver-3	242	0.41	0.32	1.00E-12	1	1
Solver-4	231	0.68	0.32	1.00E-10	1	1	Solver-4	107	0.31	0.32	1.00E-12	1	1
Solver-5	1693	4.13	0.32	1.00E-10	1	1	Solver-5	363	0.89	0.32	1.00E-12	1	1
Solver-6	1688	4.12	0.32	1.00E-10	1	1	Solver-6	352	0.86	0.32	1.00E-12	1	1
Solver-7	1695	4.1	0.32	1.00E-10	1	1	Solver-7	322	0.78	0.32	1.00E-12	1	1
	Iter. #	Time(s)	s	Error Crit.	Flag	Bypass		Iter. #	Time(s)	s	Error Crit.	Flag	Bypass
Newton	3559	5.29	0.16	1.00E-10	1	1	Newton	519	0.75	0.16	1.00E-12	1	1
Solver-1	371	0.63	0.16	1.00E-10	1	1	Solver-1	384	0.62	0.16	1.00E-12	1	1
Solver-2	133	0.41	0.16	1.00E-10	1	1	Solver-2	22	0.06	0.16	1.00E-12	1	1
Solver-3	363	0.65	0.16	1.00E-10	1	1	Solver-3	453	0.75	0.16	1.00E-12	1	1
Solver-4	430	1.27	0.16	1.00E-10	1	1	Solver-4	517	1.48	0.16	1.00E-12	1	1
Solver-5	2205	5.42	0.16	1.00E-10	1	1	Solver-5	415	0.98	0.16	1.00E-12	1	1
Solver-6	2219	5.43	0.16	1.00E-10	1	1	Solver-6	238	0.56	0.16	1.00E-12	1	1
Solver-7	2235	5.5	0.16	1.00E-10	1	1	Solver-7	470	1.11	0.16	1.00E-12	1	1
	Iter. #	Time(s)	s	Error Crit.	Flag	Bypass		Iter. #	Time(s)	s	Error Crit.	Flag	Bypass
Newton	7902	12.15	0.08	1.00E-10	0	1	Newton	664	0.98	0.08	1.00E-12	1	1
Solver-1	1163	1.97	0.08	1.00E-10	1	1	Solver-1	510	0.85	0.08	1.00E-12	1	1
Solver-2	348	1.1	0.08	1.00E-10	1	1	Solver-2	151	0.46	0.08	1.00E-12	1	1
Solver-3	1073	1.9	0.08	1.00E-10	1	1	Solver-3	676	1.16	0.08	1.00E-12	1	1
Solver-4	1233	3.67	0.08	1.00E-10	1	1	Solver-4	156	0.48	0.08	1.00E-12	1	1
Solver-5	5950	14.21	0.08	1.00E-10	1	1	Solver-5	727	1.69	0.08	1.00E-12	1	1
Solver-6	6041	14.53	0.08	1.00E-10	1	1	Solver-6	1034	2.46	0.08	1.00E-12	1	1
Solver-7	6047	14.45	0.08	1.00E-10	1	1	Solver-7	973	2.3	0.08	1.00E-12	1	1
	Iter. #	Time(s)	s	Error Crit.	Flag	Bypass		Iter. #	Time(s)	s	Error Crit.	Flag	Bypass
Newton	13883	21.84	0.04	1.00E-10	0	1	Newton	1439	2.11	0.04	1.00E-12	1	1
Solver-1	2233	3.84	0.04	1.00E-10	1	1	Solver-1	892	1.54	0.04	1.00E-12	1	1
Solver-2	808	2.55	0.04	1.00E-10	1	1	Solver-2	596	1.83	0.04	1.00E-12	1	1
Solver-3	2157	3.85	0.04	1.00E-10	1	1	Solver-3	1167	2.14	0.04	1.00E-12	1	1
Solver-4	2233	6.74	0.04	1.00E-10	1	1	Solver-4	662	2.04	0.04	1.00E-12	1	1
Solver-5	12046	27.58	0.04	1.00E-10	0	1	Solver-5	1576	3.79	0.04	1.00E-12	1	1
Solver-6	12087	27.71	0.04	1.00E-10	0	1	Solver-6	507	1.21	0.04	1.00E-12	1	1
Solver-7	12210	27.84	0.04	1.00E-10	0	1	Solver-7	1838	4.3	0.04	1.00E-12	1	1

When adaptive arc-length method and bypass are used together, it is easier to obtain solution in error criteria which can not be solved and one can understand from Table 5.7 and Table 5.8. Thus, the use of adaptive arc length is shown to be advantageous. This fact can be seen from Table 5.5 and Table 5.6 .

Table 5.5. *Performance comparison of different algebraic nonlinear equation solvers with adaptive arc-length parameter for cubic stiffness nonlinearity (Error Criterion=1e-6, 1e-8)*

	Iter. #	Time(s)	N_opt	s_max	s_min	Error Crit.	Flag	Bypass		Iter. #	Time(s)	N_opt	s_max	s_min	Error Crit.	Flag	Bypass
Newton	9268	15.73	1	0.32	0.04	1.00E-06	0	0	Newton	11323	18.1	1	0.32	0.04	1.00E-08	0	0
Solver-1	124	0.32	1	0.32	0.04	1.00E-06	1	0	Solver-1	96	0.17	1	0.32	0.04	1.00E-08	1	0
Solver-2	124	0.48	1	0.32	0.04	1.00E-06	1	0	Solver-2	96	0.29	1	0.32	0.04	1.00E-08	1	0
Solver-3	124	0.33	1	0.32	0.04	1.00E-06	1	0	Solver-3	96	0.19	1	0.32	0.04	1.00E-08	1	0
Solver-4	124	0.52	1	0.32	0.04	1.00E-06	1	0	Solver-4	96	0.29	1	0.32	0.04	1.00E-08	1	0
Solver-5	7570	23.76	1	0.32	0.04	1.00E-06	0	0	Solver-5	9268	28.21	1	0.32	0.04	1.00E-08	0	0
Solver-6	7567	23.69	1	0.32	0.04	1.00E-06	0	0	Solver-6	9312	28.54	1	0.32	0.04	1.00E-08	0	0
Solver-7	7575	23.67	1	0.32	0.04	1.00E-06	0	0	Solver-7	9422	28.35	1	0.32	0.04	1.00E-08	0	0
	Iter. #	Time(s)	N_opt	s_max	s_min	Error Crit.	Flag	Bypass		Iter. #	Time(s)	N_opt	s_max	s_min	Error Crit.	Flag	Bypass
Newton	9247	15.72	1.5	0.32	0.04	1.00E-06	0	0	Newton	11323	17.93	1.5	0.32	0.04	1.00E-08	0	0
Solver-1	124	0.23	1.5	0.32	0.04	1.00E-06	1	0	Solver-1	96	0.17	1.5	0.32	0.04	1.00E-08	1	0
Solver-2	124	0.39	1.5	0.32	0.04	1.00E-06	1	0	Solver-2	96	0.29	1.5	0.32	0.04	1.00E-08	1	0
Solver-3	124	0.27	1.5	0.32	0.04	1.00E-06	1	0	Solver-3	96	0.18	1.5	0.32	0.04	1.00E-08	1	0
Solver-4	124	0.42	1.5	0.32	0.04	1.00E-06	1	0	Solver-4	96	0.29	1.5	0.32	0.04	1.00E-08	1	0
Solver-5	7543	23.34	1.5	0.32	0.04	1.00E-06	0	0	Solver-5	9280	28.11	1.5	0.32	0.04	1.00E-08	0	0
Solver-6	7544	23.29	1.5	0.32	0.04	1.00E-06	0	0	Solver-6	9274	28.64	1.5	0.32	0.04	1.00E-08	0	0
Solver-7	7546	23.2	1.5	0.32	0.04	1.00E-06	0	0	Solver-7	9427	28.39	1.5	0.32	0.04	1.00E-08	0	0
	Iter. #	Time(s)	N_opt	s_max	s_min	Error Crit.	Flag	Bypass		Iter. #	Time(s)	N_opt	s_max	s_min	Error Crit.	Flag	Bypass
Newton	2919	4.47	2	0.32	0.04	1.00E-06	0	0	Newton	5210	7.77	2	0.32	0.04	1.00E-08	0	0
Solver-1	124	0.23	2	0.32	0.04	1.00E-06	1	0	Solver-1	96	0.17	2	0.32	0.04	1.00E-08	1	0
Solver-2	124	0.39	2	0.32	0.04	1.00E-06	1	0	Solver-2	96	0.29	2	0.32	0.04	1.00E-08	1	0
Solver-3	124	0.24	2	0.32	0.04	1.00E-06	1	0	Solver-3	96	0.19	2	0.32	0.04	1.00E-08	1	0
Solver-4	124	0.38	2	0.32	0.04	1.00E-06	1	0	Solver-4	96	0.29	2	0.32	0.04	1.00E-08	1	0
Solver-5	1511	4.68	2	0.32	0.04	1.00E-06	0	0	Solver-5	2747	8.01	2	0.32	0.04	1.00E-08	0	0
Solver-6	879	2.67	2	0.32	0.04	1.00E-06	1	0	Solver-6	2193	6.37	2	0.32	0.04	1.00E-08	1	0
Solver-7	886	2.65	2	0.32	0.04	1.00E-06	1	0	Solver-7	2211	6.42	2	0.32	0.04	1.00E-08	1	0

Table 5.6. *Performance comparison of different algebraic nonlinear equation solvers with adaptive arc-length parameter for cubic stiffness nonlinearity (Error Criterion=1e-10, 1e-12)*

	Iter. #	Time(s)	N_opt	s_max	s_min	Error Crit.	Flag	Bypass		Iter. #	Time(s)	N_opt	s_max	s_min	Error Crit.	Flag	Bypass
Newton	13670	21.4	1	0.32	0.04	1.00E-10	0	0	Newton	1020	1.49	1	0.32	0.04	1.00E-12	1	0
Solver-1	36	0.06	1	0.32	0.04	1.00E-10	1	0	Solver-1	22	0.04	1	0.32	0.04	1.00E-12	1	0
Solver-2	36	0.1	1	0.32	0.04	1.00E-10	1	0	Solver-2	22	0.06	1	0.32	0.04	1.00E-12	1	0
Solver-3	36	0.07	1	0.32	0.04	1.00E-10	1	0	Solver-3	22	0.04	1	0.32	0.04	1.00E-12	1	0
Solver-4	36	0.11	1	0.32	0.04	1.00E-10	1	0	Solver-4	22	0.06	1	0.32	0.04	1.00E-12	1	0
Solver-5	11801	35.05	1	0.32	0.04	1.00E-10	0	0	Solver-5	1444	4.06	1	0.32	0.04	1.00E-12	1	0
Solver-6	11815	35	1	0.32	0.04	1.00E-10	0	0	Solver-6	940	2.66	1	0.32	0.04	1.00E-12	1	0
Solver-7	12021	35.57	1	0.32	0.04	1.00E-10	0	0	Solver-7	776	2.2	1	0.32	0.04	1.00E-12	1	0
	Iter. #	Time(s)	N_opt	s_max	s_min	Error Crit.	Flag	Bypass		Iter. #	Time(s)	N_opt	s_max	s_min	Error Crit.	Flag	Bypass
Newton	13651	21.08	1.5	0.32	0.04	1.00E-10	0	0	Newton	1547	2.23	1.5	0.32	0.04	1.00E-12	1	0
Solver-1	36	0.06	1.5	0.32	0.04	1.00E-10	1	0	Solver-1	22	0.04	1.5	0.32	0.04	1.00E-12	1	0
Solver-2	36	0.1	1.5	0.32	0.04	1.00E-10	1	0	Solver-2	22	0.06	1.5	0.32	0.04	1.00E-12	1	0
Solver-3	36	0.07	1.5	0.32	0.04	1.00E-10	1	0	Solver-3	22	0.04	1.5	0.32	0.04	1.00E-12	1	0
Solver-4	36	0.1	1.5	0.32	0.04	1.00E-10	1	0	Solver-4	22	0.06	1.5	0.32	0.04	1.00E-12	1	0
Solver-5	11755	34.8	1.5	0.32	0.04	1.00E-10	0	0	Solver-5	1095	3.1	1.5	0.32	0.04	1.00E-12	1	0
Solver-6	11808	35.17	1.5	0.32	0.04	1.00E-10	0	0	Solver-6	1724	4.84	1.5	0.32	0.04	1.00E-12	1	0
Solver-7	11947	35.37	1.5	0.32	0.04	1.00E-10	0	0	Solver-7	1515	4.28	1.5	0.32	0.04	1.00E-12	1	0
	Iter. #	Time(s)	N_opt	s_max	s_min	Error Crit.	Flag	Bypass		Iter. #	Time(s)	N_opt	s_max	s_min	Error Crit.	Flag	Bypass
Newton	9659	14.59	2	0.32	0.04	1.00E-10	0	0	Newton	758	1.1	2	0.32	0.04	1.00E-12	1	0
Solver-1	36	0.07	2	0.32	0.04	1.00E-10	1	0	Solver-1	22	0.04	2	0.32	0.04	1.00E-12	1	0
Solver-2	36	0.11	2	0.32	0.04	1.00E-10	1	0	Solver-2	22	0.06	2	0.32	0.04	1.00E-12	1	0
Solver-3	36	0.07	2	0.32	0.04	1.00E-10	1	0	Solver-3	22	0.04	2	0.32	0.04	1.00E-12	1	0
Solver-4	36	0.14	2	0.32	0.04	1.00E-10	1	0	Solver-4	22	0.06	2	0.32	0.04	1.00E-12	1	0
Solver-5	5556	16.14	2	0.32	0.04	1.00E-10	0	0	Solver-5	690	1.94	2	0.32	0.04	1.00E-12	1	0
Solver-6	5542	15.9	2	0.32	0.04	1.00E-10	0	0	Solver-6	1597	4.44	2	0.32	0.04	1.00E-12	1	0
Solver-7	5674	16.15	2	0.32	0.04	1.00E-10	0	0	Solver-7	1037	2.92	2	0.32	0.04	1.00E-12	1	0

Table 5.7. *Performance comparison of different algebraic nonlinear equation solvers with adaptive arc-length parameter and passing unnecessary calculations for cubic stiffness nonlinearity (Error Criterion=1e-6, 1e-8)*

	Iter. #	Time(s)	N_opt	s_max	s_min	Error Crit.	Flag	Bypass		Iter. #	Time(s)	N_opt	s_max	s_min	Error Crit.	Flag	Bypass
Newton	9268	15.65	1	0.32	0.04	1.00E-06	0	1	Newton	11323	17.93	1	0.32	0.04	1.00E-08	0	1
Solver-1	3123	5.89	1	0.32	0.04	1.00E-06	1	1	Solver-1	1578	2.94	1	0.32	0.04	1.00E-08	1	1
Solver-2	124	0.48	1	0.32	0.04	1.00E-06	1	1	Solver-2	96	0.29	1	0.32	0.04	1.00E-08	1	1
Solver-3	3114	5.99	1	0.32	0.04	1.00E-06	1	1	Solver-3	1574	3.03	1	0.32	0.04	1.00E-08	1	1
Solver-4	3249	10.38	1	0.32	0.04	1.00E-06	1	1	Solver-4	4016	12.63	1	0.32	0.04	1.00E-08	1	1
Solver-5	7572	16.21	1	0.32	0.04	1.00E-06	0	1	Solver-5	9274	20.84	1	0.32	0.04	1.00E-08	0	1
Solver-6	7568	15.61	1	0.32	0.04	1.00E-06	0	1	Solver-6	9294	20.92	1	0.32	0.04	1.00E-08	0	1
Solver-7	7568	15.94	1	0.32	0.04	1.00E-06	0	1	Solver-7	9416	21.1	1	0.32	0.04	1.00E-08	0	1
	Iter. #	Time(s)	N_opt	s_max	s_min	Error Crit.	Flag	Bypass		Iter. #	Time(s)	N_opt	s_max	s_min	Error Crit.	Flag	Bypass
Newton	9247	15.37	1.5	0.32	0.04	1.00E-06	0	1	Newton	11323	17.96	1.5	0.32	0.04	1.00E-08	0	1
Solver-1	3097	5.7	1.5	0.32	0.04	1.00E-06	1	1	Solver-1	1505	2.64	1.5	0.32	0.04	1.00E-08	1	1
Solver-2	124	0.39	1.5	0.32	0.04	1.00E-06	1	1	Solver-2	96	0.29	1.5	0.32	0.04	1.00E-08	1	1
Solver-3	3088	5.8	1.5	0.32	0.04	1.00E-06	1	1	Solver-3	1505	2.8	1.5	0.32	0.04	1.00E-08	1	1
Solver-4	3223	10.05	1.5	0.32	0.04	1.00E-06	1	1	Solver-4	3962	12.17	1.5	0.32	0.04	1.00E-08	1	1
Solver-5	7546	15.58	1.5	0.32	0.04	1.00E-06	0	1	Solver-5	9269	20.48	1.5	0.32	0.04	1.00E-08	0	1
Solver-6	7544	15.59	1.5	0.32	0.04	1.00E-06	0	1	Solver-6	9285	20.49	1.5	0.32	0.04	1.00E-08	0	1
Solver-7	7545	15.57	1.5	0.32	0.04	1.00E-06	0	1	Solver-7	9385	20.62	1.5	0.32	0.04	1.00E-08	0	1
	Iter. #	Time(s)	N_opt	s_max	s_min	Error Crit.	Flag	Bypass		Iter. #	Time(s)	N_opt	s_max	s_min	Error Crit.	Flag	Bypass
Newton	2919	4.47	2	0.32	0.04	1.00E-06	0	1	Newton	5210	7.84	2	0.32	0.04	1.00E-08	0	1
Solver-1	286	0.49	2	0.32	0.04	1.00E-06	1	1	Solver-1	176	0.31	2	0.32	0.04	1.00E-08	1	1
Solver-2	124	0.39	2	0.32	0.04	1.00E-06	1	1	Solver-2	96	0.29	2	0.32	0.04	1.00E-08	1	1
Solver-3	284	0.51	2	0.32	0.04	1.00E-06	1	1	Solver-3	176	0.31	2	0.32	0.04	1.00E-08	1	1
Solver-4	844	2.52	2	0.32	0.04	1.00E-06	1	1	Solver-4	1638	4.84	2	0.32	0.04	1.00E-08	1	1
Solver-5	880	2	2	0.32	0.04	1.00E-06	1	1	Solver-5	2593	6.27	2	0.32	0.04	1.00E-08	1	1
Solver-6	1557	3.58	2	0.32	0.04	1.00E-06	0	1	Solver-6	2129	5.12	2	0.32	0.04	1.00E-08	1	1
Solver-7	886	2.01	2	0.32	0.04	1.00E-06	1	1	Solver-7	2481	5.96	2	0.32	0.04	1.00E-08	1	1

Table 5.8. *Performance comparison of different algebraic nonlinear equation solvers with adaptive arc-length parameter and passing unnecessary calculations for cubic stiffness nonlinearity (Error Criterion=1e-10, 1e-12)*

	Iter. #	Time(s)	N_opt	s_max	s_min	Error Crit.	Flag	Bypass		Iter. #	Time(s)	N_opt	s_max	s_min	Error Crit.	Flag	Bypass
Newton	13670	21.09	1	0.32	0.04	1.00E-10	0	1	Newton	1020	1.49	1	0.32	0.04	1.00E-12	1	1
Solver-1	1755	2.89	1	0.32	0.04	1.00E-10	1	1	Solver-1	651	1.12	1	0.32	0.04	1.00E-12	1	1
Solver-2	36	0.11	1	0.32	0.04	1.00E-10	1	1	Solver-2	22	0.06	1	0.32	0.04	1.00E-12	1	1
Solver-3	1625	2.86	1	0.32	0.04	1.00E-10	1	1	Solver-3	1022	1.79	1	0.32	0.04	1.00E-12	1	1
Solver-4	1828	5.45	1	0.32	0.04	1.00E-10	1	1	Solver-4	1242	3.6	1	0.32	0.04	1.00E-12	1	1
Solver-5	11771	26.8	1	0.32	0.04	1.00E-10	0	1	Solver-5	1926	4.51	1	0.32	0.04	1.00E-12	1	1
Solver-6	11806	26.87	1	0.32	0.04	1.00E-10	0	1	Solver-6	1766	4.11	1	0.32	0.04	1.00E-12	1	1
Solver-7	11941	26.86	1	0.32	0.04	1.00E-10	0	1	Solver-7	1536	3.56	1	0.32	0.04	1.00E-12	1	1
	Iter. #	Time(s)	N_opt	s_max	s_min	Error Crit.	Flag	Bypass		Iter. #	Time(s)	N_opt	s_max	s_min	Error Crit.	Flag	Bypass
Newton	13651	21.13	1.5	0.32	0.04	1.00E-10	0	1	Newton	1547	2.23	1.5	0.32	0.04	1.00E-12	1	1
Solver-1	1700	2.8	1.5	0.32	0.04	1.00E-10	1	1	Solver-1	738	1.24	1.5	0.32	0.04	1.00E-12	1	1
Solver-2	36	0.11	1.5	0.32	0.04	1.00E-10	1	1	Solver-2	22	0.06	1.5	0.32	0.04	1.00E-12	1	1
Solver-3	1639	2.83	1.5	0.32	0.04	1.00E-10	1	1	Solver-3	1254	2.15	1.5	0.32	0.04	1.00E-12	1	1
Solver-4	1722	5.01	1.5	0.32	0.04	1.00E-10	1	1	Solver-4	546	1.65	1.5	0.32	0.04	1.00E-12	1	1
Solver-5	11749	26.85	1.5	0.32	0.04	1.00E-10	0	1	Solver-5	1095	2.51	1.5	0.32	0.04	1.00E-12	1	1
Solver-6	11792	26.91	1.5	0.32	0.04	1.00E-10	0	1	Solver-6	1025	2.36	1.5	0.32	0.04	1.00E-12	1	1
Solver-7	11918	27.01	1.5	0.32	0.04	1.00E-10	0	1	Solver-7	1534	3.56	1.5	0.32	0.04	1.00E-12	1	1
	Iter. #	Time(s)	N_opt	s_max	s_min	Error Crit.	Flag	Bypass		Iter. #	Time(s)	N_opt	s_max	s_min	Error Crit.	Flag	Bypass
Newton	9659	14.25	2	0.32	0.04	1.00E-10	0	1	Newton	758	1.1	2	0.32	0.04	1.00E-12	1	1
Solver-1	238	0.39	2	0.32	0.04	1.00E-10	1	1	Solver-1	580	0.93	2	0.32	0.04	1.00E-12	1	1
Solver-2	36	0.11	2	0.32	0.04	1.00E-10	1	1	Solver-2	22	0.06	2	0.32	0.04	1.00E-12	1	1
Solver-3	218	0.37	2	0.32	0.04	1.00E-10	1	1	Solver-3	742	1.22	2	0.32	0.04	1.00E-12	1	1
Solver-4	955	2.77	2	0.32	0.04	1.00E-10	1	1	Solver-4	516	1.47	2	0.32	0.04	1.00E-12	1	1
Solver-5	5501	12.9	2	0.32	0.04	1.00E-10	0	1	Solver-5	690	1.61	2	0.32	0.04	1.00E-12	1	1
Solver-6	5507	12.92	2	0.32	0.04	1.00E-10	0	1	Solver-6	1104	2.59	2	0.32	0.04	1.00E-12	1	1
Solver-7	5566	13.04	2	0.32	0.04	1.00E-10	0	1	Solver-7	1171	2.8	2	0.32	0.04	1.00E-12	1	1

Table 5.9. *Performance comparison of different predictors with different constant arc-length parameters for cubic stiffness nonlinearity (Error Criterion=1e-6, 1e-8)*

	Iter. #	Time(s)	s	Error Crit.	Flag		Iter. #	Time(s)	s	Error Crit.	Flag
Tangent	999	1.54	0.32	1.00E-06	1	Tangent	1242	1.78	0.32	1.00E-08	1
Linear	1076	1.51	0.32	1.00E-06	1	Linear	1405	1.85	0.32	1.00E-08	1
Second	838	1.22	0.32	1.00E-06	1	Second	943	1.29	0.32	1.00E-08	1
Third	716	1.33	0.32	1.00E-06	1	Third	856	1.43	0.32	1.00E-08	1
Log	1318	1.8	0.32	1.00E-06	1	Log	1497	1.97	0.32	1.00E-08	1
Spline	838	1.25	0.32	1.00E-06	1	Spline	943	1.32	0.32	1.00E-08	1
	Iter. #	Time(s)	s	Error Crit.	Flag		Iter. #	Time(s)	s	Error Crit.	Flag
Tangent	1766	2.68	0.16	1.00E-06	1	Tangent	2034	2.99	0.16	1.00E-08	1
Linear	3433	4.73	0.16	1.00E-06	0	Linear	4155	5.58	0.16	1.00E-08	0
Second	1452	2.05	0.16	1.00E-06	1	Second	1726	2.38	0.16	1.00E-08	1
Third	1313	2.52	0.16	1.00E-06	1	Third	1452	2.67	0.16	1.00E-08	1
Log	3644	4.98	0.16	1.00E-06	0	Log	3553	4.7	0.16	1.00E-08	1
Spline	1452	2.11	0.16	1.00E-06	1	Spline	1730	2.44	0.16	1.00E-08	1
	Iter. #	Time(s)	s	Error Crit.	Flag		Iter. #	Time(s)	s	Error Crit.	Flag
Tangent	5979	9.25	0.08	1.00E-06	0	Tangent	6764	10.16	0.08	1.00E-08	0
Linear	4590	6.36	0.08	1.00E-06	1	Linear	5199	7.08	0.08	1.00E-08	1
Second	4846	6.98	0.08	1.00E-06	0	Second	5557	7.83	0.08	1.00E-08	0
Third	4134	9.17	0.08	1.00E-06	1	Third	4320	9.38	0.08	1.00E-08	1
Log	5922	8.24	0.08	1.00E-06	1	Log	6163	8.25	0.08	1.00E-08	1
Spline	4845	7.21	0.08	1.00E-06	0	Spline	5559	8.03	0.08	1.00E-08	0
	Iter. #	Time(s)	s	Error Crit.	Flag		Iter. #	Time(s)	s	Error Crit.	Flag
Tangent	10237	16.4	0.04	1.00E-06	0	Tangent	12361	18.91	0.04	1.00E-08	0
Linear	11385	15.97	0.04	1.00E-06	0	Linear	12894	17.79	0.04	1.00E-08	0
Second	9219	13.39	0.04	1.00E-06	0	Second	9711	13.96	0.04	1.00E-08	0
Third	9111	21.3	0.04	1.00E-06	0	Third	9281	21.39	0.04	1.00E-08	0
Log	11417	16.03	0.04	1.00E-06	0	Log	13645	18.66	0.04	1.00E-08	0
Spline	9219	13.87	0.04	1.00E-06	0	Spline	9709	14.51	0.04	1.00E-08	0

Performance comparison of different predictors with different constant arc-length parameters for cubic stiffness nonlinearity is given in Table 5.9 and Table 5.10. It can be seen that third order polynomial and logarithmic predictors are the worst predictors and second order polynomial and quadratic spline are the promising predictors comparing the other.

Table 5.10. *Performance comparison of different predictors with different constant arc-length parameters for cubic stiffness nonlinearity (Error Criterion=1e-10, 1e-12)*

	Iter. #	Time(s)	s	Error Crit.	Flag		Iter. #	Time(s)	s	Error Crit.	Flag
Tangent	1461	2.08	0.32	1.00E-10	1	Tangent	265	0.38	0.32	1.00E-12	1
Linear	1512	2.18	0.32	1.00E-10	1	Linear	336	0.46	0.32	1.00E-12	1
Second	1105	1.67	0.32	1.00E-10	1	Second	261	0.36	0.32	1.00E-12	1
Third	949	1.77	0.32	1.00E-10	1	Third	342	0.52	0.32	1.00E-12	1
Log	1615	3	0.32	1.00E-10	1	Log	367	0.49	0.32	1.00E-12	1
Spline	1109	1.66	0.32	1.00E-10	1	Spline	314	0.43	0.32	1.00E-12	1
	Iter. #	Time(s)	s	Error Crit.	Flag		Iter. #	Time(s)	s	Error Crit.	Flag
Tangent	2729	4.02	0.16	1.00E-10	1	Tangent	518	0.75	0.16	1.00E-12	1
Linear	5198	7.15	0.16	1.00E-10	0	Linear	553	0.76	0.16	1.00E-12	1
Second	1948	2.75	0.16	1.00E-10	1	Second	463	0.64	0.16	1.00E-12	1
Third	1774	3.21	0.16	1.00E-10	1	Third	328	0.54	0.16	1.00E-12	1
Log	3958	5.41	0.16	1.00E-10	1	Log	877	1.17	0.16	1.00E-12	1
Spline	1953	2.91	0.16	1.00E-10	1	Spline	486	0.68	0.16	1.00E-12	1
	Iter. #	Time(s)	s	Error Crit.	Flag		Iter. #	Time(s)	s	Error Crit.	Flag
Tangent	8186	12.62	0.08	1.00E-10	0	Tangent	396	0.6	0.08	1.00E-12	1
Linear	6666	9.26	0.08	1.00E-10	1	Linear	1006	1.36	0.08	1.00E-12	1
Second	6511	9.54	0.08	1.00E-10	0	Second	881	1.22	0.08	1.00E-12	1
Third	4726	10.74	0.08	1.00E-10	1	Third	305	0.55	0.08	1.00E-12	1
Log	7239	10.07	0.08	1.00E-10	1	Log	1256	1.68	0.08	1.00E-12	1
Spline	6495	9.57	0.08	1.00E-10	0	Spline	356	0.55	0.08	1.00E-12	1
	Iter. #	Time(s)	s	Error Crit.	Flag		Iter. #	Time(s)	s	Error Crit.	Flag
Tangent	14433	22.41	0.04	1.00E-10	0	Tangent	1635	2.44	0.04	1.00E-12	1
Linear	15423	22.42	0.04	1.00E-10	0	Linear	1560	2.23	0.04	1.00E-12	1
Second	11540	17.18	0.04	1.00E-10	0	Second	635	0.97	0.04	1.00E-12	1
Third	9678	23.81	0.04	1.00E-10	0	Third	1198	2.07	0.04	1.00E-12	1
Log	18434	25.58	0.04	1.00E-10	0	Log	2134	2.89	0.04	1.00E-12	1
Spline	11369	17.09	0.04	1.00E-10	0	Spline	1069	1.57	0.04	1.00E-12	1

The main advantage of adaptive arc-length method is to reduce the computational time significantly. It can be seen from Table 5.11 and Table 5.12. If adaptive arc-length method is combined with second order and quadratic spline predictors, it is possible to obtain fastest computational time.

Table 5.11. *Performance comparison of different predictors with adaptive arc-length parameter for cubic stiffness nonlinearity (Error Criterion=1e-6, 1e-8)*

	Iter. #	Time(s)	N_opt	s_max	s_min	Error Crit.	Flag		Iter. #	Time(s)	N_opt	s_max	s_min	Error Crit.	Flag
Tangent	9523	16.36	1	0.32	0.04	1.00E-06	0	Tangent	11810	18.77	1	0.32	0.04	1.00E-08	0
Linear	10728	16.04	1	0.32	0.04	1.00E-06	0	Linear	12421	18.25	1	0.32	0.04	1.00E-08	0
Second	7890	12.26	1	0.32	0.04	1.00E-06	0	Second	8900	13.76	1	0.32	0.04	1.00E-08	0
Third	7061	18.74	1	0.32	0.04	1.00E-06	0	Third	8081	20.75	1	0.32	0.04	1.00E-08	0
Log	11355	16.91	1	0.32	0.04	1.00E-06	0	Log	13575	20.19	1	0.32	0.04	1.00E-08	0
Spline	7827	12.43	1	0.32	0.04	1.00E-06	0	Spline	8842	14.27	1	0.32	0.04	1.00E-08	0
	Iter. #	Time(s)	N_opt	s_max	s_min	Error Crit.	Flag		Iter. #	Time(s)	N_opt	s_max	s_min	Error Crit.	Flag
Tangent	9498	15.95	1.5	0.32	0.04	1.00E-06	0	Tangent	11805	18.92	1.5	0.32	0.04	1.00E-08	0
Linear	10732	16	1.5	0.32	0.04	1.00E-06	0	Linear	12421	18.3	1.5	0.32	0.04	1.00E-08	0
Second	7870	12.54	1.5	0.32	0.04	1.00E-06	0	Second	8874	13.65	1.5	0.32	0.04	1.00E-08	0
Third	6964	17.83	1.5	0.32	0.04	1.00E-06	0	Third	8012	20.43	1.5	0.32	0.04	1.00E-08	0
Log	11184	16.79	1.5	0.32	0.04	1.00E-06	0	Log	13568	19.97	1.5	0.32	0.04	1.00E-08	0
Spline	7817	12.37	1.5	0.32	0.04	1.00E-06	0	Spline	8834	14.06	1.5	0.32	0.04	1.00E-08	0
	Iter. #	Time(s)	N_opt	s_max	s_min	Error Crit.	Flag		Iter. #	Time(s)	N_opt	s_max	s_min	Error Crit.	Flag
Tangent	3115	4.81	2	0.32	0.04	1.00E-06	0	Tangent	5502	8.36	2	0.32	0.04	1.00E-08	0
Linear	3792	5.34	2	0.32	0.04	1.00E-06	0	Linear	6777	9.45	2	0.32	0.04	1.00E-08	0
Second	1768	2.55	2	0.32	0.04	1.00E-06	1	Second	2975	4.2	2	0.32	0.04	1.00E-08	0
Third	1426	2.85	2	0.32	0.04	1.00E-06	1	Third	2227	4.4	2	0.32	0.04	1.00E-08	0
Log	4703	6.62	2	0.32	0.04	1.00E-06	0	Log	10047	14.25	2	0.32	0.04	1.00E-08	0
Spline	1000	1.48	2	0.32	0.04	1.00E-06	1	Spline	2957	4.32	2	0.32	0.04	1.00E-08	0

Table 5.12. *Performance comparison of different predictors with adaptive arc-length parameter for cubic stiffness nonlinearity (Error Criterion=1e-10, 1e-12)*

	Iter. #	Time(s)	N_opt	s_max	s_min	Error Crit.	Flag		Iter. #	Time(s)	N_opt	s_max	s_min	Error Crit.	Flag
Tangent	14227	22.13	1	0.32	0.04	1.00E-10	0	Tangent	1012	1.55	1	0.32	0.04	1.00E-12	1
Linear	15307	22	1	0.32	0.04	1.00E-10	0	Linear	1699	2.33	1	0.32	0.04	1.00E-12	1
Second	11004	16.38	1	0.32	0.04	1.00E-10	0	Second	990	1.43	1	0.32	0.04	1.00E-12	1
Third	8915	22.39	1	0.32	0.04	1.00E-10	0	Third	267	0.48	1	0.32	0.04	1.00E-12	1
Log	18342	26.19	1	0.32	0.04	1.00E-10	0	Log	1964	2.68	1	0.32	0.04	1.00E-12	1
Spline	10799	16.53	1	0.32	0.04	1.00E-10	0	Spline	251	0.39	1	0.32	0.04	1.00E-12	1
	Iter. #	Time(s)	N_opt	s_max	s_min	Error Crit.	Flag		Iter. #	Time(s)	N_opt	s_max	s_min	Error Crit.	Flag
Tangent	14214	22.03	1.5	0.32	0.04	1.00E-10	0	Tangent	518	0.77	1.5	0.32	0.04	1.00E-12	1
Linear	15281	21.96	1.5	0.32	0.04	1.00E-10	0	Linear	818	1.14	1.5	0.32	0.04	1.00E-12	1
Second	10976	16.34	1.5	0.32	0.04	1.00E-10	0	Second	373	0.56	1.5	0.32	0.04	1.00E-12	1
Third	8908	22.24	1.5	0.32	0.04	1.00E-10	0	Third	846	1.44	1.5	0.32	0.04	1.00E-12	1
Log	18342	26.08	1.5	0.32	0.04	1.00E-10	0	Log	1964	2.68	1.5	0.32	0.04	1.00E-12	1
Spline	10792	16.78	1.5	0.32	0.04	1.00E-10	0	Spline	525	0.8	1.5	0.32	0.04	1.00E-12	1
	Iter. #	Time(s)	N_opt	s_max	s_min	Error Crit.	Flag		Iter. #	Time(s)	N_opt	s_max	s_min	Error Crit.	Flag
Tangent	10103	15.34	2	0.32	0.04	1.00E-10	0	Tangent	565	0.83	2	0.32	0.04	1.00E-12	1
Linear	12185	17.16	2	0.32	0.04	1.00E-10	0	Linear	123	0.18	2	0.32	0.04	1.00E-12	1
Second	4586	6.51	2	0.32	0.04	1.00E-10	0	Second	931	1.31	2	0.32	0.04	1.00E-12	1
Third	3012	5.79	2	0.32	0.04	1.00E-10	0	Third	511	0.79	2	0.32	0.04	1.00E-12	1
Log	17968	25.49	2	0.32	0.04	1.00E-10	0	Log	1743	2.39	2	0.32	0.04	1.00E-12	1
Spline	4614	6.65	2	0.32	0.04	1.00E-10	0	Spline	882	1.24	2	0.32	0.04	1.00E-12	1

5.2. Realistic Finite Element Model for Different Nonlinear Solvers and Predictors

The finite element model with two shrouded blade sectors analyzed is given at Figure 5.3, in which *A* and *A'* are the points, where harmonic excitation forces applied in *+y* and *–y* direction, respectively. *B* and *B'* are the contact surfaces, where nonlinear elements are placed. *C* and *C'* are the surfaces, where fixed boundary condition applied. Natural frequencies and mode shapes of the system are obtained by finite element software ABAQUS. By using modal information, α the receptance of the linear part of the system is calculated by using first six natural frequencies and mode shapes. It is assumed that the system has proportional damping with a damping ratio 0.1%.

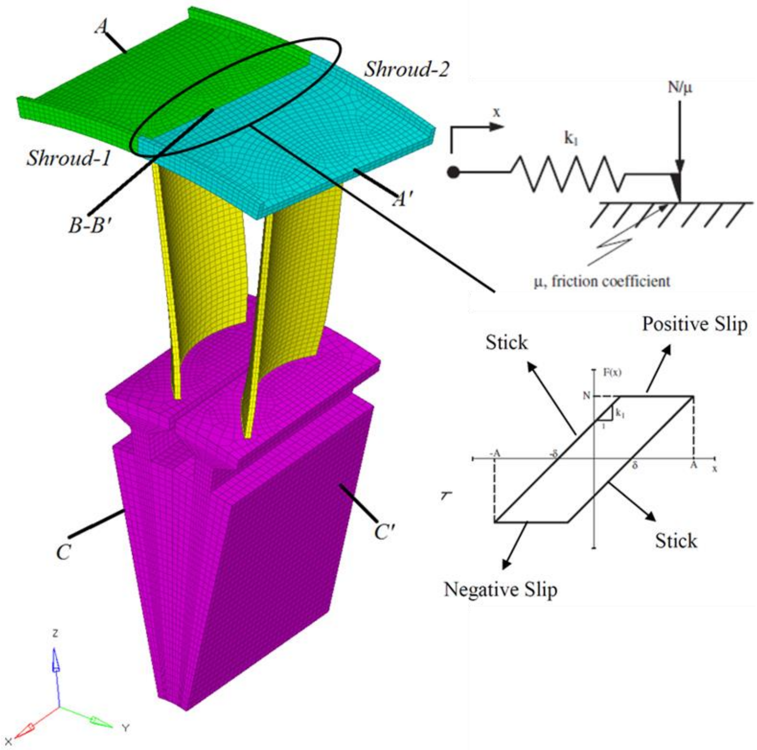

Figure 5.3. Realistic finite element model for the blade and Coulomb macroslip model between Shroud-1 and Shroud-2

Between the nodes located on the contact surface of *Shroud 1* and *Shroud 2*, four nonlinear friction elements are connected to each other by node to node.

Due to the nature of 1D-dry friction model without normal load variation, there is no turning back points and jump phenomenon in receptance. Also, 1D-dry friction is a piecewise continuous nonlinearity which means it is effective at specific frequency range of analysis. In order to investigate performance of the different predictors and solvers, the case where nonlinear effect is highly dominant is selected.

The force response of the system where four nonlinear elements are placed between shrouds are analyzed and receptance of point *A* to *B* is given in Figure 5.4. For the specified system, the single harmonic response of finite element model is studied for friction force, μN, 120 N and applying the constant amplitude excitation force at 400N.

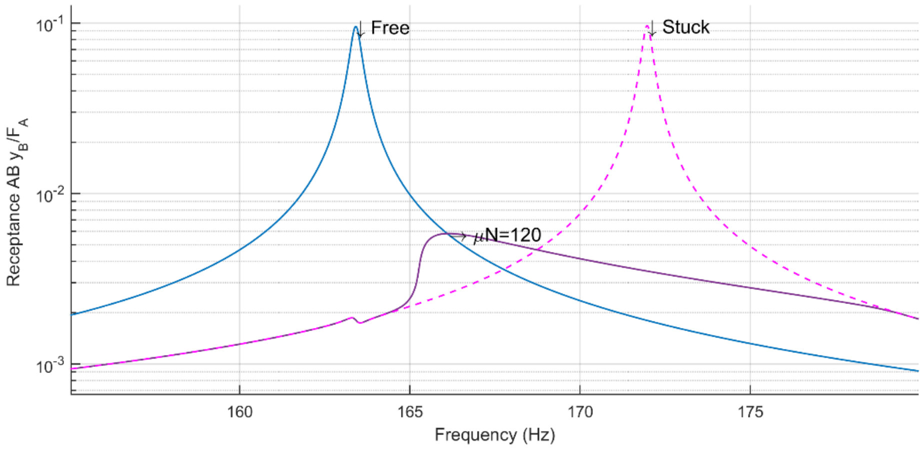

Figure 5.4. Receptance of the system with four nonlinear friction elements

When the results are inspected from Figure 5.4, the resonance frequency of the system increases as the sliding friction force increases. The system becomes completely stuck at higher sliding forces. In completely stuck case, there is no relative displacement between the shrouds, hence the effect of frictional damping is lost which results in high resonance amplitudes.

The cells marked in red in the flag column of the tables indicate that the solution could not be obtained and there is a convergence problem. The cells marked in red in the time column of the tables indicate the longest computational time. If it is marked in green, it is the fastest computational time.

Table 5.13. *Performance comparison of different algebraic nonlinear equation solvers with different constant arc-length parameters for 1D-Dry friction nonlinearity (Error Criterion=1e-6, 1e-8)*

	Iter. #	Time(s)	s	Error Crit.	Flag	Bypass		Iter. #	Time(s)	s	Error Crit.	Flag	Bypass
Newton	596	4.63	0.32	1.00E-06	0	0	Newton	617	4.89	0.32	1.00E-08	0	0
Solver-1	593	4.84	0.32	1.00E-06	0	0	Solver-1	320	2.78	0.32	1.00E-08	1	0
Solver-2	591	8.71	0.32	1.00E-06	0	0	Solver-2	317	4.97	0.32	1.00E-08	1	0
Solver-3	592	5.07	0.32	1.00E-06	0	0	Solver-3	317	2.87	0.32	1.00E-08	1	0
Solver-4	591	9.3	0.32	1.00E-06	0	0	Solver-4	317	4.97	0.32	1.00E-08	1	0
Solver-5	592	9.14	0.32	1.00E-06	0	0	Solver-5	599	8.68	0.32	1.00E-08	0	0
Solver-6	592	8.96	0.32	1.00E-06	0	0	Solver-6	599	8.88	0.32	1.00E-08	0	0
Solver-7	592	9.01	0.32	1.00E-06	0	0	Solver-7	599	8.98	0.32	1.00E-08	0	0
	Iter. #	Time(s)	s	Error Crit.	Flag	Bypass		Iter. #	Time(s)	s	Error Crit.	Flag	Bypass
Newton	1181	9.49	0.16	1.00E-06	0	0	Newton	1198	9.52	0.16	1.00E-08	0	0
Solver-1	1180	9.93	0.16	1.00E-06	0	0	Solver-1	619	5.4	0.16	1.00E-08	1	0
Solver-2	1180	17.96	0.16	1.00E-06	0	0	Solver-2	618	9.75	0.16	1.00E-08	1	0
Solver-3	1180	10.35	0.16	1.00E-06	0	0	Solver-3	618	5.63	0.16	1.00E-08	1	0
Solver-4	1180	18.85	0.16	1.00E-06	0	0	Solver-4	618	9.77	0.16	1.00E-08	1	0
Solver-5	1180	18.31	0.16	1.00E-06	0	0	Solver-5	1183	17.53	0.16	1.00E-08	0	0
Solver-6	1180	17.92	0.16	1.00E-06	0	0	Solver-6	1183	17.92	0.16	1.00E-08	0	0
Solver-7	1180	17.88	0.16	1.00E-06	0	0	Solver-7	1183	17.96	0.16	1.00E-08	0	0
	Iter. #	Time(s)	s	Error Crit.	Flag	Bypass		Iter. #	Time(s)	s	Error Crit.	Flag	Bypass
Newton	2358	18.83	0.08	1.00E-06	0	0	Newton	2371	18.92	0.08	1.00E-08	0	0
Solver-1	2358	19.71	0.08	1.00E-06	0	0	Solver-1	2358	19.94	0.08	1.00E-08	0	0
Solver-2	2358	35.6	0.08	1.00E-06	0	0	Solver-2	2358	35.84	0.08	1.00E-08	0	0
Solver-3	2358	20.56	0.08	1.00E-06	0	0	Solver-3	2358	20.58	0.08	1.00E-08	0	0
Solver-4	2358	35.73	0.08	1.00E-06	0	0	Solver-4	2358	35.64	0.08	1.00E-08	0	0
Solver-5	2358	35.7	0.08	1.00E-06	0	0	Solver-5	2358	35.71	0.08	1.00E-08	0	0
Solver-6	2358	35.55	0.08	1.00E-06	0	0	Solver-6	2358	35.63	0.08	1.00E-08	0	0
Solver-7	2358	37.55	0.08	1.00E-06	0	0	Solver-7	2358	35.66	0.08	1.00E-08	0	0
	Iter. #	Time(s)	s	Error Crit.	Flag	Bypass		Iter. #	Time(s)	s	Error Crit.	Flag	Bypass
Newton	4714	37.98	0.04	1.00E-06	0	0	Newton	4714	37.72	0.04	1.00E-08	0	0
Solver-1	4714	39.72	0.04	1.00E-06	0	0	Solver-1	4714	39.56	0.04	1.00E-08	0	0
Solver-2	4714	71.09	0.04	1.00E-06	0	0	Solver-2	4714	71.82	0.04	1.00E-08	0	0
Solver-3	4714	41.35	0.04	1.00E-06	0	0	Solver-3	4714	40.94	0.04	1.00E-08	0	0
Solver-4	4714	71.27	0.04	1.00E-06	0	0	Solver-4	4714	71.06	0.04	1.00E-08	0	0
Solver-5	4714	71.06	0.04	1.00E-06	0	0	Solver-5	4714	71.04	0.04	1.00E-08	0	0
Solver-6	4714	71.3	0.04	1.00E-06	0	0	Solver-6	4714	73.16	0.04	1.00E-08	0	0
Solver-7	4714	71.28	0.04	1.00E-06	0	0	Solver-7	4714	71.66	0.04	1.00E-08	0	0

Table 5.14. *Performance comparison of different algebraic nonlinear equation solvers with different constant arc-length parameters for 1D-Dry friction nonlinearity (Error Criterion=1e-10, 1e-12)*

	Iter. #	Time(s)	s	Error Crit.	Flag	Bypass		Iter. #	Time(s)	s	Error Crit.	Flag	Bypass
Newton	667	5.26	0.32	1.00E-10	0	0	Newton	799	6.09	0.32	1.00E-12	0	0
Solver-1	280	2.48	0.32	1.00E-10	1	0	Solver-1	273	2.41	0.32	1.00E-12	1	0
Solver-2	280	4.5	0.32	1.00E-10	1	0	Solver-2	273	4.35	0.32	1.00E-12	1	0
Solver-3	280	2.59	0.32	1.00E-10	1	0	Solver-3	273	2.51	0.32	1.00E-12	1	0
Solver-4	280	4.48	0.32	1.00E-10	1	0	Solver-4	273	4.37	0.32	1.00E-12	1	0
Solver-5	633	9.57	0.32	1.00E-10	0	0	Solver-5	706	10.28	0.32	1.00E-12	0	0
Solver-6	633	9.6	0.32	1.00E-10	0	0	Solver-6	706	10.33	0.32	1.00E-12	0	0
Solver-7	633	9.64	0.32	1.00E-10	0	0	Solver-7	706	10.57	0.32	1.00E-12	0	0
	Iter. #	Time(s)	s	Error Crit.	Flag	Bypass		Iter. #	Time(s)	s	Error Crit.	Flag	Bypass
Newton	1255	9.98	0.16	1.00E-10	0	0	Newton	1391	10.83	0.16	1.00E-12	0	0
Solver-1	546	4.86	0.16	1.00E-10	1	0	Solver-1	535	4.74	0.16	1.00E-12	1	0
Solver-2	546	8.79	0.16	1.00E-10	1	0	Solver-2	535	8.57	0.16	1.00E-12	1	0
Solver-3	546	5.08	0.16	1.00E-10	1	0	Solver-3	535	4.94	0.16	1.00E-12	1	0
Solver-4	546	8.79	0.16	1.00E-10	1	0	Solver-4	535	8.58	0.16	1.00E-12	1	0
Solver-5	1211	18.37	0.16	1.00E-10	0	0	Solver-5	1307	19.57	0.16	1.00E-12	0	0
Solver-6	1211	18.32	0.16	1.00E-10	0	0	Solver-6	1307	19.57	0.16	1.00E-12	0	0
Solver-7	1211	18.44	0.16	1.00E-10	0	0	Solver-7	1307	19.6	0.16	1.00E-12	0	0
	Iter. #	Time(s)	s	Error Crit.	Flag	Bypass		Iter. #	Time(s)	s	Error Crit.	Flag	Bypass
Newton	2429	19.42	0.08	1.00E-10	0	0	Newton	2591	20.45	0.08	1.00E-12	0	0
Solver-1	1210	10.69	0.08	1.00E-10	1	0	Solver-1	1061	9.4	0.08	1.00E-12	1	0
Solver-2	1210	19.33	0.08	1.00E-10	1	0	Solver-2	1061	17.01	0.08	1.00E-12	1	0
Solver-3	1210	11.18	0.08	1.00E-10	1	0	Solver-3	1061	9.83	0.08	1.00E-12	1	0
Solver-4	1210	19.26	0.08	1.00E-10	1	0	Solver-4	1061	17.02	0.08	1.00E-12	1	0
Solver-5	2380	35.76	0.08	1.00E-10	0	0	Solver-5	2482	37.36	0.08	1.00E-12	0	0
Solver-6	2380	35.84	0.08	1.00E-10	0	0	Solver-6	2482	37.64	0.08	1.00E-12	0	0
Solver-7	2380	35.89	0.08	1.00E-10	0	0	Solver-7	2482	37.92	0.08	1.00E-12	0	0
	Iter. #	Time(s)	s	Error Crit.	Flag	Bypass		Iter. #	Time(s)	s	Error Crit.	Flag	Bypass
Newton	4767	37.78	0.04	1.00E-10	0	0	Newton	4922	39.03	0.04	1.00E-12	0	0
Solver-1	4714	39.29	0.04	1.00E-10	0	0	Solver-1	2229	19.73	0.04	1.00E-12	1	0
Solver-2	4714	71.29	0.04	1.00E-10	0	0	Solver-2	2229	35.82	0.04	1.00E-12	1	0
Solver-3	4714	41.18	0.04	1.00E-10	0	0	Solver-3	2229	20.71	0.04	1.00E-12	1	0
Solver-4	4714	71.07	0.04	1.00E-10	0	0	Solver-4	2229	35.31	0.04	1.00E-12	1	0
Solver-5	4714	70.99	0.04	1.00E-10	0	0	Solver-5	4798	70.78	0.04	1.00E-12	0	0
Solver-6	4714	71.02	0.04	1.00E-10	0	0	Solver-6	4798	70.32	0.04	1.00E-12	0	0
Solver-7	4714	71.44	0.04	1.00E-10	0	0	Solver-7	4798	70.89	0.04	1.00E-12	0	0

Table 5.13 and Table 5.14 shows that convergence problems are experienced with increasing arc-length parameters and decreasing error criteria. In addition, all solvers is not able to obtain solutions at low absolute error criteria. In some cases where other solvers had convergence problems, the solution was obtained by Newton method. It shows that convergence domain of the solvers other than Newton's is narrow.

Table 5.15. *Performance comparison of different algebraic nonlinear equation solvers with different constant arc-length parameters and passing unnecessary calculations for 1D-Dry friction nonlinearity (Error Criterion=1e-6, 1e-8)*

	Iter. #	Time(s)	s	Error Crit.	Flag	Bypass		Iter. #	Time(s)	s	Error Crit.	Flag	Bypass
Newton	596	4.65	0.32	1.00E-06	0	1	Newton	617	5.04	0.32	1.00E-08	0	1
Solver-1	593	4.84	0.32	1.00E-06	0	1	Solver-1	600	5.08	0.32	1.00E-08	0	1
Solver-2	591	8.71	0.32	1.00E-06	0	1	Solver-2	317	5.04	0.32	1.00E-08	1	1
Solver-3	592	4.84	0.32	1.00E-06	0	1	Solver-3	599	5.07	0.32	1.00E-08	0	1
Solver-4	591	8.69	0.32	1.00E-06	0	1	Solver-4	599	9.12	0.32	1.00E-08	0	1
Solver-5	592	4.88	0.32	1.00E-06	0	1	Solver-5	599	5.17	0.32	1.00E-08	0	1
Solver-6	592	4.84	0.32	1.00E-06	0	1	Solver-6	599	5.22	0.32	1.00E-08	0	1
Solver-7	592	4.87	0.32	1.00E-06	0	1	Solver-7	599	5.21	0.32	1.00E-08	0	1
	Iter. #	Time(s)	s	Error Crit.	Flag	Bypass		Iter. #	Time(s)	s	Error Crit.	Flag	Bypass
Newton	1181	9.1	0.16	1.00E-06	0	1	Newton	1198	9.63	0.16	1.00E-08	0	1
Solver-1	1180	9.53	0.16	1.00E-06	0	1	Solver-1	1184	10.03	0.16	1.00E-08	0	1
Solver-2	1180	17.2	0.16	1.00E-06	0	1	Solver-2	618	9.89	0.16	1.00E-08	1	1
Solver-3	1180	9.51	0.16	1.00E-06	0	1	Solver-3	1182	10.03	0.16	1.00E-08	0	1
Solver-4	1180	17.22	0.16	1.00E-06	0	1	Solver-4	1182	18.16	0.16	1.00E-08	0	1
Solver-5	1180	9.51	0.16	1.00E-06	0	1	Solver-5	1183	10.14	0.16	1.00E-08	0	1
Solver-6	1180	9.51	0.16	1.00E-06	0	1	Solver-6	1183	10.11	0.16	1.00E-08	0	1
Solver-7	1180	9.52	0.16	1.00E-06	0	1	Solver-7	1183	10.1	0.16	1.00E-08	0	1
	Iter. #	Time(s)	s	Error Crit.	Flag	Bypass		Iter. #	Time(s)	s	Error Crit.	Flag	Bypass
Newton	2358	18.23	0.08	1.00E-06	0	1	Newton	2371	18.97	0.08	1.00E-08	0	1
Solver-1	2358	19.07	0.08	1.00E-06	0	1	Solver-1	2358	19.74	0.08	1.00E-08	0	1
Solver-2	2358	34.59	0.08	1.00E-06	0	1	Solver-2	2358	35.78	0.08	1.00E-08	0	1
Solver-3	2358	19.1	0.08	1.00E-06	0	1	Solver-3	2358	19.8	0.08	1.00E-08	0	1
Solver-4	2358	34.55	0.08	1.00E-06	0	1	Solver-4	2358	35.76	0.08	1.00E-08	0	1
Solver-5	2358	19.08	0.08	1.00E-06	0	1	Solver-5	2358	19.79	0.08	1.00E-08	0	1
Solver-6	2358	19.05	0.08	1.00E-06	0	1	Solver-6	2358	19.78	0.08	1.00E-08	0	1
Solver-7	2358	19	0.08	1.00E-06	0	1	Solver-7	2358	19.77	0.08	1.00E-08	0	1
	Iter. #	Time(s)	s	Error Crit.	Flag	Bypass		Iter. #	Time(s)	s	Error Crit.	Flag	Bypass
Newton	4714	36.49	0.04	1.00E-06	0	1	Newton	4714	38.21	0.04	1.00E-08	0	1
Solver-1	4714	38.29	0.04	1.00E-06	0	1	Solver-1	4714	39.87	0.04	1.00E-08	0	1
Solver-2	4714	69.19	0.04	1.00E-06	0	1	Solver-2	4714	71.99	0.04	1.00E-08	0	1
Solver-3	4714	38.32	0.04	1.00E-06	0	1	Solver-3	4714	39.81	0.04	1.00E-08	0	1
Solver-4	4714	69.11	0.04	1.00E-06	0	1	Solver-4	4714	72.02	0.04	1.00E-08	0	1
Solver-5	4714	39.54	0.04	1.00E-06	0	1	Solver-5	4714	39.87	0.04	1.00E-08	0	1
Solver-6	4714	40.26	0.04	1.00E-06	0	1	Solver-6	4714	40.01	0.04	1.00E-08	0	1
Solver-7	4714	39.77	0.04	1.00E-06	0	1	Solver-7	4714	39.92	0.04	1.00E-08	0	1

Troubles encountered in Table 5.13 and Table 5.14 are resolved by using bypass. It can be seen from Table 5.15 and Table 5.16. However, the computational times of solver-5,6,7 are fallen by almost half since additional calculations are avoided.

Table 5.16. *Performance comparison of different algebraic nonlinear equation solvers with different constant arc-length parameters and passing unnecessary calculations for 1D-Dry friction nonlinearity (Error Criterion=1e-10, 1e-12)*

	Iter. #	Time(s)	s	Error Crit.	Flag	Bypass		Iter. #	Time(s)	s	Error Crit.	Flag	Bypass
Newton	667	5.28	0.32	1.00E-10	0	1	Newton	799	6.18	0.32	1.00E-12	0	1
Solver-1	636	5.32	0.32	1.00E-10	0	1	Solver-1	706	5.84	0.32	1.00E-12	0	1
Solver-2	280	4.53	0.32	1.00E-10	1	1	Solver-2	273	4.41	0.32	1.00E-12	1	1
Solver-3	632	5.33	0.32	1.00E-10	0	1	Solver-3	702	5.88	0.32	1.00E-12	0	1
Solver-4	632	9.59	0.32	1.00E-10	0	1	Solver-4	701	10.56	0.32	1.00E-12	0	1
Solver-5	633	5.69	0.32	1.00E-10	0	1	Solver-5	706	6.98	0.32	1.00E-12	0	1
Solver-6	633	5.7	0.32	1.00E-10	0	1	Solver-6	706	7.04	0.32	1.00E-12	0	1
Solver-7	633	5.69	0.32	1.00E-10	0	1	Solver-7	706	7.02	0.32	1.00E-12	0	1
	Iter. #	Time(s)	s	Error Crit.	Flag	Bypass		Iter. #	Time(s)	s	Error Crit.	Flag	Bypass
Newton	1255	9.98	0.16	1.00E-10	0	1	Newton	1391	11.21	0.16	1.00E-12	0	1
Solver-1	1211	10.21	0.16	1.00E-10	0	1	Solver-1	1301	11	0.16	1.00E-12	0	1
Solver-2	546	8.97	0.16	1.00E-10	1	1	Solver-2	535	8.74	0.16	1.00E-12	1	1
Solver-3	1211	9.95	0.16	1.00E-10	0	1	Solver-3	1300	10.87	0.16	1.00E-12	0	1
Solver-4	1211	17.9	0.16	1.00E-10	0	1	Solver-4	1300	19.71	0.16	1.00E-12	0	1
Solver-5	1211	10.26	0.16	1.00E-10	0	1	Solver-5	1307	12.03	0.16	1.00E-12	0	1
Solver-6	1211	10.26	0.16	1.00E-10	0	1	Solver-6	1307	12.04	0.16	1.00E-12	0	1
Solver-7	1211	10.24	0.16	1.00E-10	0	1	Solver-7	1307	12.03	0.16	1.00E-12	0	1
	Iter. #	Time(s)	s	Error Crit.	Flag	Bypass		Iter. #	Time(s)	s	Error Crit.	Flag	Bypass
Newton	2429	19.22	0.08	1.00E-10	0	1	Newton	2591	20.78	0.08	1.00E-12	0	1
Solver-1	2380	20.24	0.08	1.00E-10	0	1	Solver-1	2481	21.05	0.08	1.00E-12	0	1
Solver-2	1210	19.21	0.08	1.00E-10	1	1	Solver-2	1061	17.39	0.08	1.00E-12	1	1
Solver-3	2380	19.89	0.08	1.00E-10	0	1	Solver-3	2481	21.34	0.08	1.00E-12	0	1
Solver-4	2380	36.28	0.08	1.00E-10	0	1	Solver-4	2481	38.67	0.08	1.00E-12	0	1
Solver-5	2380	20.6	0.08	1.00E-10	0	1	Solver-5	2482	22.17	0.08	1.00E-12	0	1
Solver-6	2380	20.56	0.08	1.00E-10	0	1	Solver-6	2482	22.26	0.08	1.00E-12	0	1
Solver-7	2380	20.56	0.08	1.00E-10	0	1	Solver-7	2482	22.29	0.08	1.00E-12	0	1
	Iter. #	Time(s)	s	Error Crit.	Flag	Bypass		Iter. #	Time(s)	s	Error Crit.	Flag	Bypass
Newton	4767	38.6	0.04	1.00E-10	0	1	Newton	4922	39.4	0.04	1.00E-12	0	1
Solver-1	4714	40.02	0.04	1.00E-10	0	1	Solver-1	4798	40.47	0.04	1.00E-12	0	1
Solver-2	4714	71.92	0.04	1.00E-10	0	1	Solver-2	2229	36.11	0.04	1.00E-12	1	1
Solver-3	4714	39.65	0.04	1.00E-10	0	1	Solver-3	4797	40.79	0.04	1.00E-12	0	1
Solver-4	4714	73.21	0.04	1.00E-10	0	1	Solver-4	4798	73.52	0.04	1.00E-12	0	1
Solver-5	4714	40.35	0.04	1.00E-10	0	1	Solver-5	4798	42.74	0.04	1.00E-12	0	1
Solver-6	4714	40.7	0.04	1.00E-10	0	1	Solver-6	4798	40.76	0.04	1.00E-12	0	1
Solver-7	4714	40.65	0.04	1.00E-10	0	1	Solver-7	4798	40.36	0.04	1.00E-12	0	1

Thus, the use of adaptive arc length is shown to be advantageous. This fact can be seen from Table 5.17 and Table 5.18. When adaptive arc-length method and bypass are used together, it is easier to obtain solution in error criteria which can not be solved and one can understand from Table 5.19 and Table 5.20. Also, Table 5.19 and Table 5.20 show that some solvers show better performance than Newton's method.

Table 5.17. *Performance comparison of different algebraic nonlinear equation solvers with adaptive arc-length parameter for 1D-Dry friction nonlinearity (Error Criterion=1e-6, 1e-8)*

	Iter. #	Time(s)	N_opt	s_max	s_min	Error Crit.	Flag	Bypass		Iter. #	Time(s)	N_opt	s_max	s_min	Error Crit.	Flag	Bypass
Newton	884	7.02	1	0.32	0.04	1.00E-06	0	0	Newton	2654	20.86	1	0.32	0.04	1.00E-08	0	0
Solver-1	667	5.53	1	0.32	0.04	1.00E-06	0	0	Solver-1	1228	10.19	1	0.32	0.04	1.00E-08	0	0
Solver-2	591	8.87	1	0.32	0.04	1.00E-06	0	0	Solver-2	317	4.99	1	0.32	0.04	1.00E-08	1	0
Solver-3	666	5.79	1	0.32	0.04	1.00E-06	0	0	Solver-3	317	2.88	1	0.32	0.04	1.00E-08	1	0
Solver-4	591	8.94	1	0.32	0.04	1.00E-06	0	0	Solver-4	317	5.02	1	0.32	0.04	1.00E-08	1	0
Solver-5	667	9.92	1	0.32	0.04	1.00E-06	0	0	Solver-5	1227	18.06	1	0.32	0.04	1.00E-08	0	0
Solver-6	667	9.9	1	0.32	0.04	1.00E-06	0	0	Solver-6	1227	18.12	1	0.32	0.04	1.00E-08	0	0
Solver-7	667	9.86	1	0.32	0.04	1.00E-06	0	0	Solver-7	1227	18.12	1	0.32	0.04	1.00E-08	0	0
	Iter. #	Time(s)	N_opt	s_max	s_min	Error Crit.	Flag	Bypass		Iter. #	Time(s)	N_opt	s_max	s_min	Error Crit.	Flag	Bypass
Newton	730	5.6	1.5	0.32	0.04	1.00E-06	0	0	Newton	2311	18.35	1.5	0.32	0.04	1.00E-08	0	0
Solver-1	654	5.37	1.5	0.32	0.04	1.00E-06	0	0	Solver-1	320	2.81	1.5	0.32	0.04	1.00E-08	1	0
Solver-2	591	8.7	1.5	0.32	0.04	1.00E-06	0	0	Solver-2	317	5.05	1.5	0.32	0.04	1.00E-08	1	0
Solver-3	621	5.26	1.5	0.32	0.04	1.00E-06	0	0	Solver-3	317	2.91	1.5	0.32	0.04	1.00E-08	1	0
Solver-4	591	8.68	1.5	0.32	0.04	1.00E-06	0	0	Solver-4	317	5.04	1.5	0.32	0.04	1.00E-08	1	0
Solver-5	621	9.07	1.5	0.32	0.04	1.00E-06	0	0	Solver-5	874	13.11	1.5	0.32	0.04	1.00E-08	0	0
Solver-6	621	9.1	1.5	0.32	0.04	1.00E-06	0	0	Solver-6	874	13.1	1.5	0.32	0.04	1.00E-08	0	0
Solver-7	621	9.19	1.5	0.32	0.04	1.00E-06	0	0	Solver-7	874	13.13	1.5	0.32	0.04	1.00E-08	0	0
	Iter. #	Time(s)	N_opt	s_max	s_min	Error Crit.	Flag	Bypass		Iter. #	Time(s)	N_opt	s_max	s_min	Error Crit.	Flag	Bypass
Newton	596	4.67	2	0.32	0.04	1.00E-06	0	0	Newton	620	4.95	2	0.32	0.04	1.00E-08	0	0
Solver-1	593	4.88	2	0.32	0.04	1.00E-06	0	0	Solver-1	320	2.8	2	0.32	0.04	1.00E-08	1	0
Solver-2	591	8.83	2	0.32	0.04	1.00E-06	0	0	Solver-2	317	5.05	2	0.32	0.04	1.00E-08	1	0
Solver-3	592	5.1	2	0.32	0.04	1.00E-06	0	0	Solver-3	317	2.91	2	0.32	0.04	1.00E-08	1	0
Solver-4	591	8.8	2	0.32	0.04	1.00E-06	0	0	Solver-4	317	5.05	2	0.32	0.04	1.00E-08	1	0
Solver-5	592	8.74	2	0.32	0.04	1.00E-06	0	0	Solver-5	599	9.12	2	0.32	0.04	1.00E-08	0	0
Solver-6	592	8.76	2	0.32	0.04	1.00E-06	0	0	Solver-6	599	9.11	2	0.32	0.04	1.00E-08	0	0
Solver-7	592	8.73	2	0.32	0.04	1.00E-06	0	0	Solver-7	599	9.17	2	0.32	0.04	1.00E-08	0	0

Table 5.18. *Performance comparison of different algebraic nonlinear equation solvers with adaptive arc-length parameter for 1D-Dry friction nonlinearity (Error Criterion=1e-10, 1e-12)*

	Iter. #	Time(s)	N_opt	s_max	s_min	Error Crit.	Flag	Bypass		Iter. #	Time(s)	N_opt	s_max	s_min	Error Crit.	Flag	Bypass
Newton	2798	21.4	1	0.32	0.04	1.00E-10	0	0	Newton	3113	23.73	1	0.32	0.04	1.00E-12	0	0
Solver-1	280	2.46	1	0.32	0.04	1.00E-10	1	0	Solver-1	273	2.41	1	0.32	0.04	1.00E-12	1	0
Solver-2	280	4.48	1	0.32	0.04	1.00E-10	1	0	Solver-2	273	4.36	1	0.32	0.04	1.00E-12	1	0
Solver-3	280	2.6	1	0.32	0.04	1.00E-10	1	0	Solver-3	273	2.51	1	0.32	0.04	1.00E-12	1	0
Solver-4	280	4.47	1	0.32	0.04	1.00E-10	1	0	Solver-4	273	4.4	1	0.32	0.04	1.00E-12	1	0
Solver-5	2693	39.35	1	0.32	0.04	1.00E-10	0	0	Solver-5	2988	43.86	1	0.32	0.04	1.00E-12	0	0
Solver-6	2693	39.44	1	0.32	0.04	1.00E-10	0	0	Solver-6	2988	43.62	1	0.32	0.04	1.00E-12	0	0
Solver-7	2693	39.36	1	0.32	0.04	1.00E-10	0	0	Solver-7	2988	44.19	1	0.32	0.04	1.00E-12	0	0
	Iter. #	Time(s)	N_opt	s_max	s_min	Error Crit.	Flag	Bypass		Iter. #	Time(s)	N_opt	s_max	s_min	Error Crit.	Flag	Bypass
Newton	2783	21.99	1.5	0.32	0.04	1.00E-10	0	0	Newton	3115	24.04	1.5	0.32	0.04	1.00E-12	0	0
Solver-1	280	2.54	1.5	0.32	0.04	1.00E-10	1	0	Solver-1	273	2.43	1.5	0.32	0.04	1.00E-12	1	0
Solver-2	280	4.62	1.5	0.32	0.04	1.00E-10	1	0	Solver-2	273	4.43	1.5	0.32	0.04	1.00E-12	1	0
Solver-3	280	2.79	1.5	0.32	0.04	1.00E-10	1	0	Solver-3	273	2.54	1.5	0.32	0.04	1.00E-12	1	0
Solver-4	280	4.59	1.5	0.32	0.04	1.00E-10	1	0	Solver-4	273	4.39	1.5	0.32	0.04	1.00E-12	1	0
Solver-5	2543	37.42	1.5	0.32	0.04	1.00E-10	0	0	Solver-5	2988	44.66	1.5	0.32	0.04	1.00E-12	0	0
Solver-6	2543	37.49	1.5	0.32	0.04	1.00E-10	0	0	Solver-6	2988	44.26	1.5	0.32	0.04	1.00E-12	0	0
Solver-7	2543	37.59	1.5	0.32	0.04	1.00E-10	0	0	Solver-7	2988	44.67	1.5	0.32	0.04	1.00E-12	0	0
	Iter. #	Time(s)	N_opt	s_max	s_min	Error Crit.	Flag	Bypass		Iter. #	Time(s)	N_opt	s_max	s_min	Error Crit.	Flag	Bypass
Newton	718	5.69	2	0.32	0.04	1.00E-10	0	0	Newton	1164	8.72	2	0.32	0.04	1.00E-12	0	0
Solver-1	280	2.61	2	0.32	0.04	1.00E-10	1	0	Solver-1	273	2.43	2	0.32	0.04	1.00E-12	1	0
Solver-2	280	4.55	2	0.32	0.04	1.00E-10	1	0	Solver-2	273	4.4	2	0.32	0.04	1.00E-12	1	0
Solver-3	280	2.59	2	0.32	0.04	1.00E-10	1	0	Solver-3	273	2.54	2	0.32	0.04	1.00E-12	1	0
Solver-4	280	4.48	2	0.32	0.04	1.00E-10	1	0	Solver-4	273	4.39	2	0.32	0.04	1.00E-12	1	0
Solver-5	633	9.61	2	0.32	0.04	1.00E-10	0	0	Solver-5	736	11.01	2	0.32	0.04	1.00E-12	0	0
Solver-6	633	9.95	2	0.32	0.04	1.00E-10	0	0	Solver-6	736	10.89	2	0.32	0.04	1.00E-12	0	0
Solver-7	633	10.24	2	0.32	0.04	1.00E-10	0	0	Solver-7	720	11.32	2	0.32	0.04	1.00E-12	0	0

Table 5.19. *Performance comparison of different algebraic nonlinear equation solvers with adaptive arc-length parameter and passing unnecessary calculations for 1D-Dry friction nonlinearity (Error Criterion=1e-6, 1e-8)*

	Iter. #	Time(s)	N_opt	s_max	s_min	Error Crit.	Flag	Bypass		Iter. #	Time(s)	N_opt	s_max	s_min	Error Crit.	Flag	Bypass
Newton	884	7.16	1	0.32	0.04	1.00E-06	0	1	Newton	2654	20.45	1	0.32	0.04	1.00E-08	0	1
Solver-1	667	5.78	1	0.32	0.04	1.00E-06	0	1	Solver-1	1228	10.54	1	0.32	0.04	1.00E-08	0	1
Solver-2	591	9.48	1	0.32	0.04	1.00E-06	0	1	Solver-2	317	5.37	1	0.32	0.04	1.00E-08	1	1
Solver-3	666	5.77	1	0.32	0.04	1.00E-06	0	1	Solver-3	1036	8.54	1	0.32	0.04	1.00E-08	0	1
Solver-4	591	9.31	1	0.32	0.04	1.00E-06	0	1	Solver-4	1036	15.36	1	0.32	0.04	1.00E-08	0	1
Solver-5	667	5.74	1	0.32	0.04	1.00E-06	0	1	Solver-5	1227	10.09	1	0.32	0.04	1.00E-08	0	1
Solver-6	667	5.74	1	0.32	0.04	1.00E-06	0	1	Solver-6	1227	10.12	1	0.32	0.04	1.00E-08	0	1
Solver-7	667	5.7	1	0.32	0.04	1.00E-06	0	1	Solver-7	1227	10.06	1	0.32	0.04	1.00E-08	0	1
	Iter. #	Time(s)	N_opt	s_max	s_min	Error Crit.	Flag	Bypass		Iter. #	Time(s)	N_opt	s_max	s_min	Error Crit.	Flag	Bypass
Newton	730	5.83	1.5	0.32	0.04	1.00E-06	0	1	Newton	2311	17.84	1.5	0.32	0.04	1.00E-08	0	1
Solver-1	654	5.48	1.5	0.32	0.04	1.00E-06	0	1	Solver-1	875	7.24	1.5	0.32	0.04	1.00E-08	0	1
Solver-2	591	9.02	1.5	0.32	0.04	1.00E-06	0	1	Solver-2	317	5.03	1.5	0.32	0.04	1.00E-08	1	1
Solver-3	621	5.21	1.5	0.32	0.04	1.00E-06	0	1	Solver-3	874	7.25	1.5	0.32	0.04	1.00E-08	0	1
Solver-4	591	8.97	1.5	0.32	0.04	1.00E-06	0	1	Solver-4	874	13.12	1.5	0.32	0.04	1.00E-08	0	1
Solver-5	621	5.18	1.5	0.32	0.04	1.00E-06	0	1	Solver-5	874	7.37	1.5	0.32	0.04	1.00E-08	0	1
Solver-6	621	5.22	1.5	0.32	0.04	1.00E-06	0	1	Solver-6	874	7.35	1.5	0.32	0.04	1.00E-08	0	1
Solver-7	621	5.24	1.5	0.32	0.04	1.00E-06	0	1	Solver-7	874	7.77	1.5	0.32	0.04	1.00E-08	0	1
	Iter. #	Time(s)	N_opt	s_max	s_min	Error Crit.	Flag	Bypass		Iter. #	Time(s)	N_opt	s_max	s_min	Error Crit.	Flag	Bypass
Newton	596	5.23	2	0.32	0.04	1.00E-06	0	1	Newton	620	4.95	2	0.32	0.04	1.00E-08	0	1
Solver-1	593	5.09	2	0.32	0.04	1.00E-06	0	1	Solver-1	600	5.05	2	0.32	0.04	1.00E-08	0	1
Solver-2	591	9.1	2	0.32	0.04	1.00E-06	0	1	Solver-2	317	5.02	2	0.32	0.04	1.00E-08	1	1
Solver-3	592	5.02	2	0.32	0.04	1.00E-06	0	1	Solver-3	599	5.05	2	0.32	0.04	1.00E-08	0	1
Solver-4	591	9.13	2	0.32	0.04	1.00E-06	0	1	Solver-4	599	9.12	2	0.32	0.04	1.00E-08	0	1
Solver-5	592	5.07	2	0.32	0.04	1.00E-06	0	1	Solver-5	599	5.17	2	0.32	0.04	1.00E-08	0	1
Solver-6	592	5.01	2	0.32	0.04	1.00E-06	0	1	Solver-6	599	5.18	2	0.32	0.04	1.00E-08	0	1
Solver-7	592	4.98	2	0.32	0.04	1.00E-06	0	1	Solver-7	599	5.19	2	0.32	0.04	1.00E-08	0	1

Table 5.20. *Performance comparison of different algebraic nonlinear equation solvers with adaptive arc-length parameter and passing unnecessary calculations for 1D-Dry friction nonlinearity (Error Criterion=1e-10, 1e-12)*

	Iter. #	Time(s)	N_opt	s_max	s_min	Error Crit.	Flag	Bypass		Iter. #	Time(s)	N_opt	s_max	s_min	Error Crit.	Flag	Bypass
Newton	2798	21.63	1	0.32	0.04	1.00E-10	0	1	Newton	3113	24.57	1	0.32	0.04	1.00E-12	0	1
Solver-1	2693	21.92	1	0.32	0.04	1.00E-10	0	1	Solver-1	2988	24.43	1	0.32	0.04	1.00E-12	0	1
Solver-2	280	4.5	1	0.32	0.04	1.00E-10	1	1	Solver-2	273	4.39	1	0.32	0.04	1.00E-12	1	1
Solver-3	2693	21.88	1	0.32	0.04	1.00E-10	0	1	Solver-3	2988	24.36	1	0.32	0.04	1.00E-12	0	1
Solver-4	2693	39.6	1	0.32	0.04	1.00E-10	0	1	Solver-4	2988	44.3	1	0.32	0.04	1.00E-12	0	1
Solver-5	2693	22.35	1	0.32	0.04	1.00E-10	0	1	Solver-5	2988	26.29	1	0.32	0.04	1.00E-12	0	1
Solver-6	2693	22.32	1	0.32	0.04	1.00E-10	0	1	Solver-6	2988	25.74	1	0.32	0.04	1.00E-12	0	1
Solver-7	2693	21.96	1	0.32	0.04	1.00E-10	0	1	Solver-7	2988	26.14	1	0.32	0.04	1.00E-12	0	1
	Iter. #	Time(s)	N_opt	s_max	s_min	Error Crit.	Flag	Bypass		Iter. #	Time(s)	N_opt	s_max	s_min	Error Crit.	Flag	Bypass
Newton	2783	20.85	1.5	0.32	0.04	1.00E-10	0	1	Newton	3115	24.17	1.5	0.32	0.04	1.00E-12	0	1
Solver-1	2543	21.39	1.5	0.32	0.04	1.00E-10	0	1	Solver-1	2985	24.35	1.5	0.32	0.04	1.00E-12	0	1
Solver-2	280	4.54	1.5	0.32	0.04	1.00E-10	1	1	Solver-2	273	4.43	1.5	0.32	0.04	1.00E-12	1	1
Solver-3	2543	20.84	1.5	0.32	0.04	1.00E-10	0	1	Solver-3	2985	25.34	1.5	0.32	0.04	1.00E-12	0	1
Solver-4	2543	37.73	1.5	0.32	0.04	1.00E-10	0	1	Solver-4	2985	43.86	1.5	0.32	0.04	1.00E-12	0	1
Solver-5	2543	21.16	1.5	0.32	0.04	1.00E-10	0	1	Solver-5	2988	25.29	1.5	0.32	0.04	1.00E-12	0	1
Solver-6	2543	21.18	1.5	0.32	0.04	1.00E-10	0	1	Solver-6	2988	25.33	1.5	0.32	0.04	1.00E-12	0	1
Solver-7	2543	21.19	1.5	0.32	0.04	1.00E-10	0	1	Solver-7	2988	25.29	1.5	0.32	0.04	1.00E-12	0	1
	Iter. #	Time(s)	N_opt	s_max	s_min	Error Crit.	Flag	Bypass		Iter. #	Time(s)	N_opt	s_max	s_min	Error Crit.	Flag	Bypass
Newton	718	5.67	2	0.32	0.04	1.00E-10	0	1	Newton	1164	8.57	2	0.32	0.04	1.00E-12	0	1
Solver-1	642	5.39	2	0.32	0.04	1.00E-10	0	1	Solver-1	764	6.17	2	0.32	0.04	1.00E-12	0	1
Solver-2	280	4.53	2	0.32	0.04	1.00E-10	1	1	Solver-2	273	4.34	2	0.32	0.04	1.00E-12	1	1
Solver-3	632	5.35	2	0.32	0.04	1.00E-10	0	1	Solver-3	732	6	2	0.32	0.04	1.00E-12	0	1
Solver-4	632	9.67	2	0.32	0.04	1.00E-10	0	1	Solver-4	701	10.43	2	0.32	0.04	1.00E-12	0	1
Solver-5	633	5.76	2	0.32	0.04	1.00E-10	0	1	Solver-5	736	8.05	2	0.32	0.04	1.00E-12	0	1
Solver-6	633	5.73	2	0.32	0.04	1.00E-10	0	1	Solver-6	736	7.58	2	0.32	0.04	1.00E-12	0	1
Solver-7	633	5.74	2	0.32	0.04	1.00E-10	0	1	Solver-7	720	7.17	2	0.32	0.04	1.00E-12	0	1

Table 5.21. *Performance comparison of different predictors with different constant arc-length parameters for 1D-Dry friction nonlinearity (Error Criterion=1e-6, 1e-8)*

	Iter. #	Time(s)	s	Error Crit.	Flag		Iter. #	Time(s)	s	Error Crit.	Flag
Tangent	596	4.84	0.32	1.00E-06	0	Tangent	617	4.92	0.32	1.00E-08	0
Linear	598	4.42	0.32	1.00E-06	0	Linear	625	4.54	0.32	1.00E-08	0
Second	597	4.43	0.32	1.00E-06	0	Second	616	4.51	0.32	1.00E-08	0
Third	592	4.95	0.32	1.00E-06	0	Third	614	5.05	0.32	1.00E-08	0
Log	922	6.76	0.32	1.00E-06	0	Log	1226	8.72	0.32	1.00E-08	0
Spline	594	4.48	0.32	1.00E-06	0	Spline	614	4.62	0.32	1.00E-08	0
	Iter. #	Time(s)	s	Error Crit.	Flag		Iter. #	Time(s)	s	Error Crit.	Flag
Tangent	1181	9.52	0.16	1.00E-06	0	Tangent	1198	9.66	0.16	1.00E-08	0
Linear	1183	8.67	0.16	1.00E-06	0	Linear	1209	8.89	0.16	1.00E-08	0
Second	1180	8.65	0.16	1.00E-06	0	Second	1196	8.82	0.16	1.00E-08	0
Third	1175	9.92	0.16	1.00E-06	0	Third	1190	10.03	0.16	1.00E-08	0
Log	1350	9.85	0.16	1.00E-06	0	Log	2388	17.14	0.16	1.00E-08	0
Spline	1177	8.76	0.16	1.00E-06	0	Spline	1193	9.64	0.16	1.00E-08	0
	Iter. #	Time(s)	s	Error Crit.	Flag		Iter. #	Time(s)	s	Error Crit.	Flag
Tangent	2358	18.88	0.08	1.00E-06	0	Tangent	2371	18.92	0.08	1.00E-08	0
Linear	2357	17.18	0.08	1.00E-06	0	Linear	2378	17.31	0.08	1.00E-08	0
Second	2358	17.26	0.08	1.00E-06	0	Second	2361	17.21	0.08	1.00E-08	0
Third	2353	20.6	0.08	1.00E-06	0	Third	2355	20.5	0.08	1.00E-08	0
Log	2441	17.78	0.08	1.00E-06	0	Log	3197	23.09	0.08	1.00E-08	0
Spline	2355	17.53	0.08	1.00E-06	0	Spline	2358	17.49	0.08	1.00E-08	0
	Iter. #	Time(s)	s	Error Crit.	Flag		Iter. #	Time(s)	s	Error Crit.	Flag
Tangent	4714	37.51	0.04	1.00E-06	0	Tangent	4714	37.47	0.04	1.00E-08	0
Linear	4713	34.25	0.04	1.00E-06	0	Linear	4721	35.29	0.04	1.00E-08	0
Second	4714	34.01	0.04	1.00E-06	0	Second	4714	35.24	0.04	1.00E-08	0
Third	4709	41.55	0.04	1.00E-06	0	Third	4709	43.14	0.04	1.00E-08	0
Log	4755	34.6	0.04	1.00E-06	0	Log	5134	38.29	0.04	1.00E-08	0
Spline	4711	35.19	0.04	1.00E-06	0	Spline	4711	35.83	0.04	1.00E-08	0

Performance comparison of different predictors with different constant arc-length parameters for 1D-dry friction nonlinearity is given in Table 5.21 and Table 5.22. It can be seen that third order polynomial and logarithmic predictors are the worst predictors and second order polynomial and quadratic spline are the promising predictors comparing the other.

Table 5.22. *Performance comparison of different predictors with different constant arc-length parameters for 1D-Dry friction nonlinearity (Error Criterion=1e-10, 1e-12)*

	Iter. #	Time(s)	s	Error Crit.	Flag		Iter. #	Time(s)	s	Error Crit.	Flag
Tangent	667	5.38	0.32	1.00E-10	0	Tangent	799	6.22	0.32	1.00E-12	0
Linear	693	5.09	0.32	1.00E-10	0	Linear	886	6.44	0.32	1.00E-12	0
Second	647	4.83	0.32	1.00E-10	0	Second	697	5.2	0.32	1.00E-12	0
Third	647	5.36	0.32	1.00E-10	0	Third	683	5.55	0.32	1.00E-12	0
Log	1323	9.52	0.32	1.00E-10	0	Log	1595	11.39	0.32	1.00E-12	0
Spline	647	4.62	0.32	1.00E-10	0	Spline	697	5.2	0.32	1.00E-12	0
	Iter. #	Time(s)	s	Error Crit.	Flag		Iter. #	Time(s)	s	Error Crit.	Flag
Tangent	1255	10.11	0.16	1.00E-10	0	Tangent	1391	11.02	0.16	1.00E-12	0
Linear	1290	9.51	0.16	1.00E-10	0	Linear	1477	10.69	0.16	1.00E-12	0
Second	1224	9.38	0.16	1.00E-10	0	Second	1284	9.61	0.16	1.00E-12	0
Third	1218	10.46	0.16	1.00E-10	0	Third	1270	10.79	0.16	1.00E-12	0
Log	2460	18.26	0.16	1.00E-10	0	Log	2666	19.11	0.16	1.00E-12	0
Spline	1224	9.08	0.16	1.00E-10	0	Spline	1284	9.47	0.16	1.00E-12	0
	Iter. #	Time(s)	s	Error Crit.	Flag		Iter. #	Time(s)	s	Error Crit.	Flag
Tangent	2429	19.83	0.08	1.00E-10	0	Tangent	2591	20.85	0.08	1.00E-12	0
Linear	2454	18.37	0.08	1.00E-10	0	Linear	2675	19.48	0.08	1.00E-12	0
Second	2395	18.34	0.08	1.00E-10	0	Second	2636	18.66	0.08	1.00E-12	0
Third	2385	21.33	0.08	1.00E-10	0	Third	2423	20.39	0.08	1.00E-12	0
Log	4783	34.48	0.08	1.00E-10	0	Log	4939	33.54	0.08	1.00E-12	0
Spline	2395	17.91	0.08	1.00E-10	0	Spline	2457	17.59	0.08	1.00E-12	0
	Iter. #	Time(s)	s	Error Crit.	Flag		Iter. #	Time(s)	s	Error Crit.	Flag
Tangent	4767	39.22	0.04	1.00E-10	0	Tangent	4922	37.71	0.04	1.00E-12	0
Linear	4800	36.2	0.04	1.00E-10	0	Linear	5035	35.23	0.04	1.00E-12	0
Second	4735	36.42	0.04	1.00E-10	0	Second	6524	44.82	0.04	1.00E-12	0
Third	4727	43.09	0.04	1.00E-10	0	Third	4759	40.77	0.04	1.00E-12	0
Log	8758	63.19	0.04	1.00E-10	0	Log	9574	64.9	0.04	1.00E-12	0
Spline	4729	35.81	0.04	1.00E-10	0	Spline	4795	34.31	0.04	1.00E-12	0

The main advantage of adaptive arc-length method is to reduce the computational time significantly. It can be seen from Table 5.11 and Table 5.12. If adaptive arc-length method is combined with second order and quadratic spline predictors, it is possible to obtain fastest computational time.

Table 5.23. *Performance comparison of different predictors with adaptive arc-length parameter for 1D-Dry friction nonlinearity (Error Criterion=1e-6, 1e-8)*

	Iter. #	Time(s)	N_opt	s_max	s_min	Error Crit.	Flag		Iter. #	Time(s)	N_opt	s_max	s_min	Error Crit.	Flag
Tangent	884	7.07	1	0.32	0.04	1.00E-06	0	Tangent	2654	19.78	1	0.32	0.04	1.00E-08	0
Linear	1306	9.35	1	0.32	0.04	1.00E-06	0	Linear	2745	18.91	1	0.32	0.04	1.00E-08	0
Second	1060	8.3	1	0.32	0.04	1.00E-06	0	Second	1959	13.66	1	0.32	0.04	1.00E-08	0
Third	1176	10.04	1	0.32	0.04	1.00E-06	0	Third	1786	15.42	1	0.32	0.04	1.00E-08	0
Log	4695	33.39	1	0.32	0.04	1.00E-06	0	Log	5050	35.98	1	0.32	0.04	1.00E-08	0
Spline	881	6.29	1	0.32	0.04	1.00E-06	0	Spline	1791	12.64	1	0.32	0.04	1.00E-08	0
	Iter. #	Time(s)	N_opt	s_max	s_min	Error Crit.	Flag		Iter. #	Time(s)	N_opt	s_max	s_min	Error Crit.	Flag
Tangent	730	5.58	1.5	0.32	0.04	1.00E-06	0	Tangent	2311	17.21	1.5	0.32	0.04	1.00E-08	0
Linear	975	6.87	1.5	0.32	0.04	1.00E-06	0	Linear	2751	19.15	1.5	0.32	0.04	1.00E-08	0
Second	988	7.14	1.5	0.32	0.04	1.00E-06	0	Second	1821	12.71	1.5	0.32	0.04	1.00E-08	0
Third	1080	8.8	1.5	0.32	0.04	1.00E-06	0	Third	1647	14.12	1.5	0.32	0.04	1.00E-08	0
Log	3715	26.34	1.5	0.32	0.04	1.00E-06	0	Log	5046	36.18	1.5	0.32	0.04	1.00E-08	0
Spline	770	5.49	1.5	0.32	0.04	1.00E-06	0	Spline	1676	11.84	1.5	0.32	0.04	1.00E-08	0
	Iter. #	Time(s)	N_opt	s_max	s_min	Error Crit.	Flag		Iter. #	Time(s)	N_opt	s_max	s_min	Error Crit.	Flag
Tangent	596	4.63	2	0.32	0.04	1.00E-06	0	Tangent	620	4.81	2	0.32	0.04	1.00E-08	0
Linear	600	4.44	2	0.32	0.04	1.00E-06	0	Linear	642	4.57	2	0.32	0.04	1.00E-08	0
Second	602	4.52	2	0.32	0.04	1.00E-06	0	Second	628	4.46	2	0.32	0.04	1.00E-08	0
Third	601	4.78	2	0.32	0.04	1.00E-06	0	Third	623	4.94	2	0.32	0.04	1.00E-08	0
Log	1147	8.06	2	0.32	0.04	1.00E-06	0	Log	3318	22.79	2	0.32	0.04	1.00E-08	0
Spline	594	4.28	2	0.32	0.04	1.00E-06	0	Spline	615	4.42	2	0.32	0.04	1.00E-08	0

Table 5.24. *Performance comparison of different predictors with adaptive arc-length parameter for 1D-Dry friction nonlinearity (Error Criterion=1e-10, 1e-12)*

	Iter. #	Time(s)	N_opt	s_max	s_min	Error Crit.	Flag		Iter. #	Time(s)	N_opt	s_max	s_min	Error Crit.	Flag
Tangent	2798	20.77	1	0.32	0.04	1.00E-10	0	Tangent	3113	23.15	1	0.32	0.04	1.00E-12	0
Linear	2953	20.07	1	0.32	0.04	1.00E-10	0	Linear	3304	22.6	1	0.32	0.04	1.00E-12	0
Second	2971	20.3	1	0.32	0.04	1.00E-10	0	Second	4755	31.49	1	0.32	0.04	1.00E-12	0
Third	3038	26.35	1	0.32	0.04	1.00E-10	0	Third	3137	27.18	1	0.32	0.04	1.00E-12	0
Log	8647	59.33	1	0.32	0.04	1.00E-10	0	Log	9461	66.14	1	0.32	0.04	1.00E-12	0
Spline	2802	19.48	1	0.32	0.04	1.00E-10	0	Spline	2932	20.29	1	0.32	0.04	1.00E-12	0
	Iter. #	Time(s)	N_opt	s_max	s_min	Error Crit.	Flag		Iter. #	Time(s)	N_opt	s_max	s_min	Error Crit.	Flag
Tangent	2783	20.66	1.5	0.32	0.04	1.00E-10	0	Tangent	3115	22.94	1.5	0.32	0.04	1.00E-12	0
Linear	2931	20.17	1.5	0.32	0.04	1.00E-10	0	Linear	3316	22.47	1.5	0.32	0.04	1.00E-12	0
Second	2923	19.96	1.5	0.32	0.04	1.00E-10	0	Second	4720	31.33	1.5	0.32	0.04	1.00E-12	0
Third	3022	26.2	1.5	0.32	0.04	1.00E-10	0	Third	3109	26.57	1.5	0.32	0.04	1.00E-12	0
Log	8647	59.57	1.5	0.32	0.04	1.00E-10	0	Log	9461	64.2	1.5	0.32	0.04	1.00E-12	0
Spline	2732	18.95	1.5	0.32	0.04	1.00E-10	0	Spline	2920	20.14	1.5	0.32	0.04	1.00E-12	0
	Iter. #	Time(s)	N_opt	s_max	s_min	Error Crit.	Flag		Iter. #	Time(s)	N_opt	s_max	s_min	Error Crit.	Flag
Tangent	718	5.52	2	0.32	0.04	1.00E-10	0	Tangent	1164	8.31	2	0.32	0.04	1.00E-12	0
Linear	808	5.61	2	0.32	0.04	1.00E-10	0	Linear	1515	10.06	2	0.32	0.04	1.00E-12	0
Second	666	4.82	2	0.32	0.04	1.00E-10	0	Second	762	5.44	2	0.32	0.04	1.00E-12	0
Third	667	5.25	2	0.32	0.04	1.00E-10	0	Third	726	5.59	2	0.32	0.04	1.00E-12	0
Log	7913	54.24	2	0.32	0.04	1.00E-10	0	Log	9452	64.05	2	0.32	0.04	1.00E-12	0
Spline	660	4.74	2	0.32	0.04	1.00E-10	0	Spline	754	5.31	2	0.32	0.04	1.00E-12	0

Computational performance of the different solvers and different predictors are evaluated on models containing cubic stiffness and dry friction nonlinearities. When the results are evaluated, it is found to be advantageous to use the adaptive arc-length and bypass methods together.

When computational performance of Solver-1 and Solver-3 on 1D dry friction nonlinearity are examined, they were observed to stand out among different solution methods. The use of these two methods together with bypass and adaptive arc-length reduces the solution time and the number of iterations. Comparing Solver-1 and Solver-3 with Newton's method, they reduce the calculation time by at least 10 percent and at most 50 percent. Thus, if a model includes only 1D dry friction nonlinearity without normal load variation, it is beneficial to use Solver-1 and Solver-3.

Different predictor algorithms are investigated for 1D dry friction nonlinearity. Second order polynomial and quadratic spline are promising methods to reduce computational time and iteration numbers for both constant and adaptive arc-length parameter. However, logarithmic and third order polynomial fit are not beneficial to use since these two methods increase computational time and iteration numbers.

Particularly in cases where adaptive arc-length method is used, quadratic spline predictor yields promising results. The reason is that quadratic spline has a continuous derivative on the intervals.

5.3. Detail Comparison of Different Solvers and Proposed Predictors for Both Cubic Stiffness and 1D-Dry Friction Nonlinearities

In this section, obtained results of case studies on different solvers and predictors are explained according to minimum calculation time.

5.3.1. Effect of Error Criteria for Different Solvers

Comparison results of different solvers according to different error criteria are given in Figure 5.5 and Figure 5.6 for both cubic stiffness and 1D dry friction nonlinearities. For constant step size (*s=0.04*), Solver-1-2-3-4 fail, while error criterion decreases, since their convergence field is narrower than the other solvers. Newton has better performance than other solvers for both cases when "*s*" is constant.

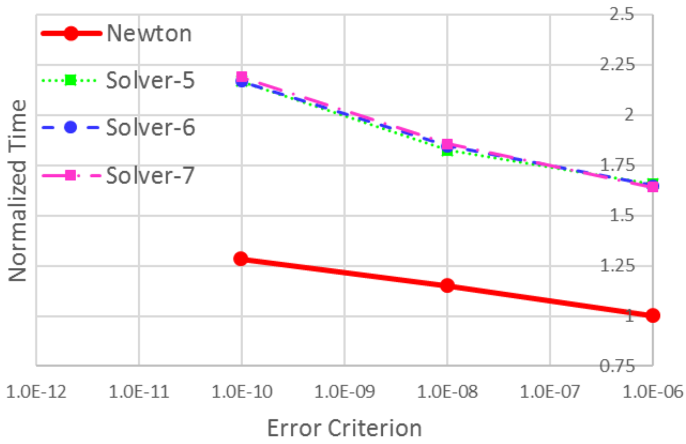

Figure 5.5. Performance comparison of different solvers for cubic stiffness nonlinearity according to different error criteria and constant step

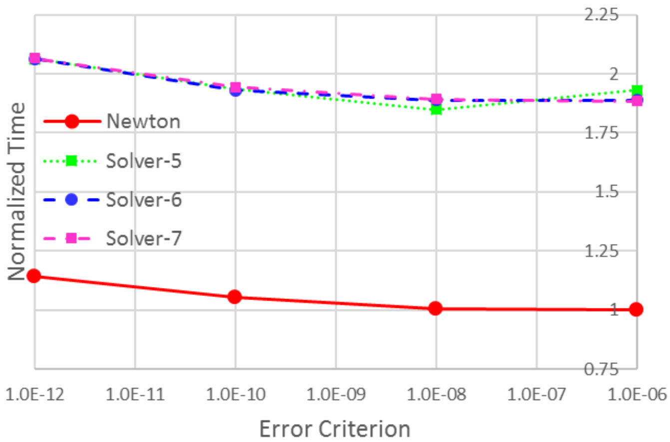

Figure 5.6. Performance comparison of different solvers for contact nonlinearity according to different error criteria and constant step

5.3.2. Effect of Adaptive Step Size for Different Solvers

Comparison results of different solvers according to different adaptive step size parameters are given in Figure 5.7 and Figure 5.8 for both cubic stiffness and 1D dry friction nonlinearities. For error criterion (*1e-10*) and specific adaptive step size parameters, Solver-5-6-7 have better performance than Newton.

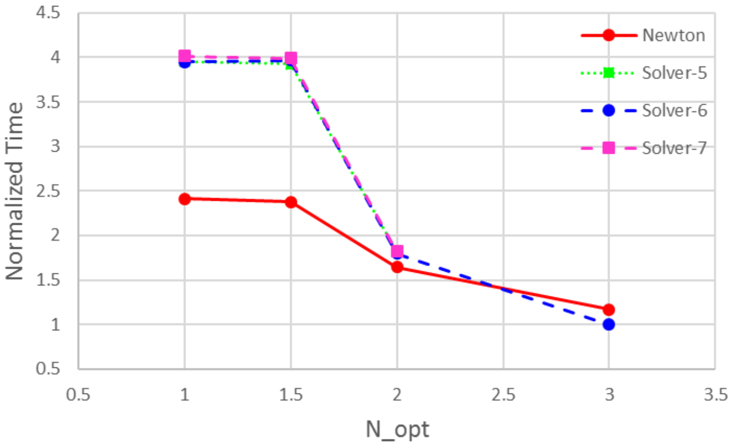

Figure 5.7. Performance comparison of different solvers for cubic stiffness nonlinearity according to different adaptive step size parameters

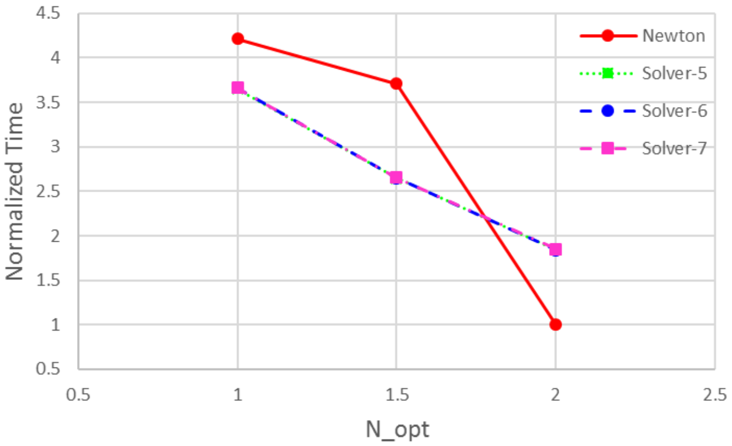

Figure 5.8. Performance comparison of different solvers for contact nonlinearity according to different adaptive step size parameters

5.3.3. Effect of Adaptive Step Size and Bypass for Different Solvers

Comparison results of different solvers according to different adaptive step size parameters are given in Figure 5.9 and Figure 5.10 for both cubic stiffness and 1D dry friction nonlinearities. For error criterion (*1e-10*) and specific adaptive step size parameter (*N_opt=2*), Solver-5-6-7 have better performance than Newton.

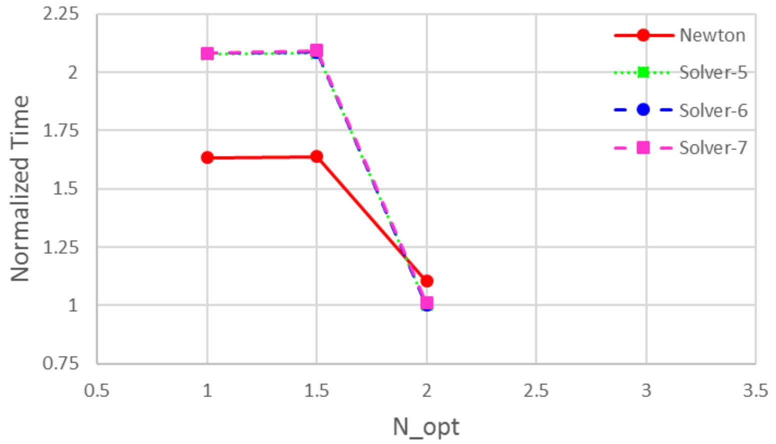

Figure 5.9. Performance comparison of different solvers for cubic stiffness nonlinearity according to different adaptive step size parameters and bypass

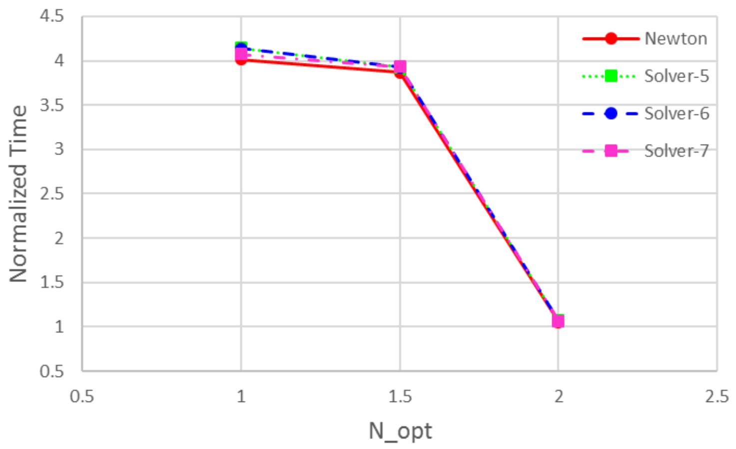

Figure 5.10. Performance comparison of different solvers for contact nonlinearity according to different adaptive step size parameters and bypass

5.3.4. Effect of Error Criteria for Different Predictors

Comparison results of different predictors according to different error criteria are given in Figure 5.11 and Figure 5.12 for both cubic stiffness and 1D dry friction nonlinearities. For constant step size (*s=0.04*), second order and quadratic spline have better performance than tangent predictor.

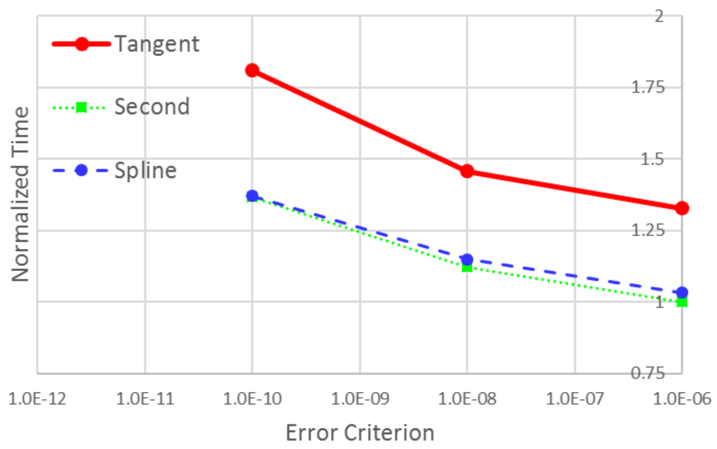

Figure 5.11. Performance comparison of different predictors for cubic stiffness nonlinearity according to different error criteria and constant step

Figure 5.12. Performance comparison of different predictors for contact nonlinearity according to different error criteria and constant step

5.3.5. Effect of Adaptive Step Size for Different Predictors

Comparison results of different predictors according to different adaptive step size parameters are given in Figure 5.13 and Figure 5.14 for both cubic stiffness and 1D dry friction nonlinearities. For constant step size (*s=0.04*), quadratic spline have better performance than tangent predictor.

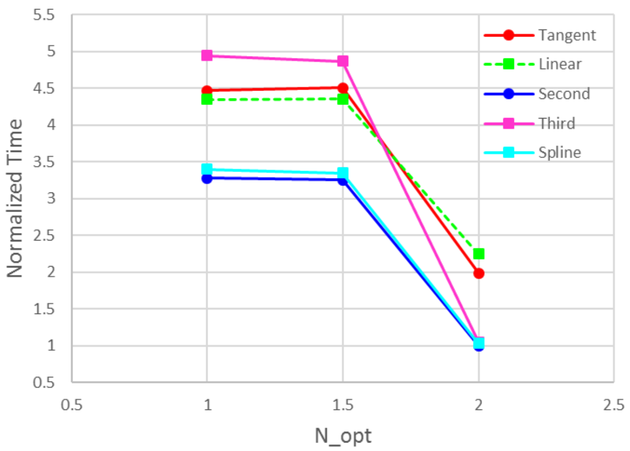

Figure 5.13. Performance comparison of different predictors for cubic stiffness nonlinearity according to different adaptive step size parameters

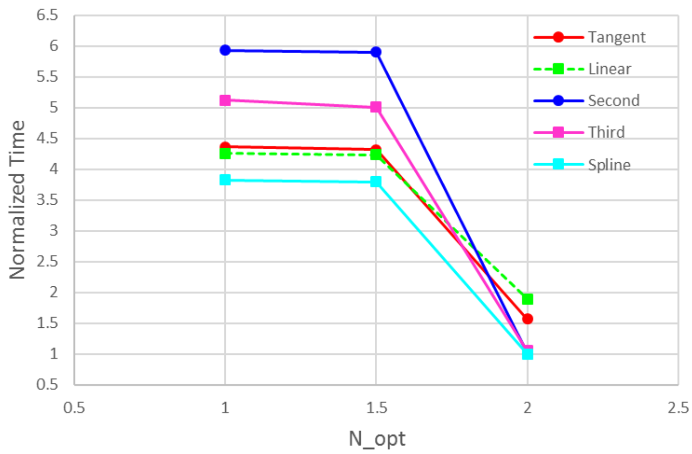

Figure 5.14. Performance comparison of different predictors for contact nonlinearity according to different adaptive step size parameters

CHAPTER 6

CONCLUSION

In this study, a new method, Jacobian Element Method (JEM), is proposed for numerical Jacobian calculation of nonlinear vibration analysis of multidegree-of-freedom (MDOF) systems to obtain steady state response. Also, computational performance of different solution methods and predictor algorithms are investigated. The aim of the work is to obtain significant reduction of computational time for Jacobian calculation compared to classical Jacobian calculation.

In order to reduce number of nonlinear equations, receptance method is applied. Also, Harmonic Balance Method (HBM) which transform the nonlinear differential equations into a set of nonlinear algebraic equations, are used. HBM provides to solve nonlinear algebraic equations iteratively.

In order to validate the proposed method, different case studies are investigated. Force response analysis of a 4-DOF lumped parameter model and a realistic finite element model of bladed disk sector are performed.

Three different 4-DOF lumped models are constructed by changing number of nonlinear elements. Cubic stiffness nonlinearity is considered in lumped parameter model. It is observed the effect of the number of harmonics and the number of nonlinear elements used in the models on the computational time. It is proven that computational reduction ratio is independent of number of harmonics. Also, it is shown that Jacobian element method provides to reductions in total computational times up to sixty percent. The comparison study on the lumped models is made by using the developed JEM and classical Jacobian method. The effect of different nonlinear element numbers on the computational time is studied while keeping the nonlinear equation numbers constant. Also, the effect of number of harmonics used in the response analysis on computational time is analyzed. Theoretical and actual computational reduction ratios are also presented and the results are in good agreement.

Force response analysis of a bladed disk sector is analyzed by using finite element model. 1D-dry friction nonlinearity is placed node to node connection between shroud contact surfaces. The effect of number of nonlinear elements and nonlinear equations are investigated. It is shown that Jacobian element method provides to reductions in total computational times up to ninety percent.

Comparison of computational performance of different nonlinear algebraic equation solvers and different predictors are studied where, solvers with different convergence order are selected based on the number of Jacobian matrix and vector function evaluations. In order to compare the performance of these selected nonlinear solvers, a lumped parameter model with cubic stiffness nonlinearity and bladed disk sector with 1D dry friction nonlinearity are considered. Several case studies are performed and nonlinear solvers are compared to each other in terms of solution time based on error tolerance used. When the results are evaluated, it is found to be advantageous to use the adaptive arc-length and bypass methods together in solution process for different solvers. Comparing different solvers with Newton's method, it is shown that they can reduce the calculation time by at least 10 percent and at most 50 percent for specific adaptive step size parameter. Also, it is concluded that second order polynomial and quadratic spline are promising methods to reduce computational time and iteration numbers for both constant and adaptive arc-length parameter.

REFERENCES

[1] T.M. Cameron, J.H. Griffin, An alternating frequency/time domain method for calculating the steady-state response of nonlinear dynamic systems, J. Appl. Mech. 56(1) (1989) 149-154.

[2] M. Rezaiee-Pajand, M. Hashemian, A. Bohluly, A novel time integration formulation for nonlinear dynamic analysis, Aerospace Science and Tech. 69 (2017) 625-635.

[3] T. Liu, Q. Li, C. Zhao, An efficient time-integration method for nonlinear dynamic analysis of solids and structures, Sci. China Phys. Mech. Astron. 56(4), (2013), 798–804.

[4] Balendra, T., Wang, C. M., & Cheong, H. F. (1995). Effectiveness of tuned liquid column dampers for vibration control of towers. Engineering Structures, 17(9), 668–675. https://doi.org/10.1016/0141-0296(95)00036-7

[5] Gao, H., Kwok, K. C. S., & Samali, B. (1997). Optimization of tuned liquid column dampers. Engineering Structures, 19(6), 476–486. https://doi.org/10.1016/S0141-0296(96)00099-5

[6] Xie, Y. (1996). An Assessment Of Time Integration Schemes For Non-Linear Dynamic Equations. Journal of Sound and Vibration, 192(1), 321-331. doi:10.1006/jsvi.1996.0190

[7] C-H. Menq, J.H. Griffin, A comparison of transient and steady state finite element analyses of the forced response of a frictionally damped beam, J. Vib., Acoust., Stress, and Reliab. 107(1) (1985) 19-25.

[8] C-H. Menq, J.H. Griffin, J. Bielak, The forced response of shrouded fan stages, J. Vib., Acoust., Stress, and Reliab. 108(1) (1986) 50-55.

[9] E. Budak, H.N. Özgüven, A dynamic analysis method for harmonically excited nonlinear structures, Struct. Vib. and Acoust. ASME 18(3) (1989) 23-29. 74

[10] E. Budak, H.N. Özgüven, Iterative receptance method for determining harmonic response of structures with symmetrical non-linearities, Mech. Syst. Signal Process. 7(1) (1993) 75-87

[11] O. Tanrikulu, B. Kuran, H.N. Özgüven, M. Imregun, Forced harmonic response analysis of nonlinear structures using describing functions, AIAA J. 31(7) (1993) 1313-1320.

[12] B.Kuran, H.N. Özgüven, A modal superposition method for non-linear structures, J Sound Vib. 189(3) (1996) 315-339.

[13] E. Cigeroglu, N. An, C.H. Menq, A microslip friction model with normal load variation induced by normal motion, Nonlinear Dyn. 50(3) (2007) 609.

[14] E. Cigeroglu, N. An, C.H. Menq, Forced response prediction of constrained and unconstrained structures coupled through frictional contacts, J. Eng. Gas Turbines Power 131(2) (2009) 022505.

[15] E.P. Petrov, A high-accuracy model reduction for analysis of nonlinear vibrations in structures with contact interfaces, J. Eng. Gas Turbines Power 133(10) (2011) 102503

[16] S. Zucca, B.I. Epureanu, Reduced order models for nonlinear dynamic analysis of structures with intermittent contacts, J. Vib. Control (2017) 1077546316689214

[17] V. Jaumouillé, J.J. Sinou, B. Petitjean, An adaptive harmonic balance method for predicting the nonlinear dynamic responses of mechanical systems—application to bolted structures, 329(19) (2010) 4048-4067.

[18] G. Von Groll, D.J. Ewins, The harmonic balance method with arc-length continuation in rotor/stator contact problems, J Sound Vib. 241(2) (2001) 223-233.

[19] E.P. Petrov, D.J. Ewins, Analytical formulation of friction interface elements for analysis of nonlinear multi-harmonic vibrations of bladed discs, ASME Turbo Expo 2002: Power for Land, Sea, and Air 4 (2002) 899-908.

[20] R. Lewandowski, Computational formulation for periodic vibration of geometrically nonlinear structures—part 2: numerical strategy and examples, Int. J. Solids Struct. 34(15) (1997) 1949-1964.

[21] P. Riberio, M. Petyt, Geometrical non-linear, steady state, forced, periodic vibration of plates, part I: model and convergence studies, J Sound Vib. 226(5) (1999) 955-983.

[22] L. Xu, M.W. Lu, Q. Cao, Nonlinear vibrations of dynamical systems with a general form of piecewise-linear viscous damping by incremental harmonic balance method, Phys. Letters A, 301(1) (2002) 65-73.

[23] M. Guskov, J.J. Sinou, F. Thouverez, Multi-dimensional harmonic balance applied to rotor dynamics, Mech. Research Communications 35(8) (2008) 537-545.

[24] R. Gasch, K. Knothe, Kontinua und ihre Diskretisierung, Strukturdynamik, Band 2, Springer, Berlin (1989)

[25] G. Genta, Vibration of Structures and Machines: Practical Aspects, Springer, New York (1999)

[26] Borrajo, J., Zucca, S., & Gola, M. (2006). Analytical formulation of the Jacobian matrix for non-linear calculation of the forced response of turbine blade assemblies with wedge friction dampers. International Journal of Non-Linear Mechanics, 41(10), 1118-1127. doi:10.1016/j.ijnonlinmec.2006.11.003

[27] Darvish, M.T. and Barati, A. (2007) A Third-Order Newton-Type Method to Solve Systems of Nonlinear Equations. *Applied Mathematics and Computation*, **187**, 630-635. http://dx.doi.org/10.1016/j.amc.2006.08.080

[28] Cordero, Alicia, Juan R. Torregrosa, and María P. Vassileva. "Pseudocomposition: a technique to design predictor–corrector methods for systems of nonlinear equations." *Applied Mathematics and Computation* 218.23 (2012): 11496-11504. http://dx.doi.org/10.1016/j.amc.2012.04.081

[29] Sharma, Janak Raj, and Rangan K. Guha. "Simple yet efficient Newton-like method for systems of nonlinear equations." Calcolo (2015): 1-23.

[30] Sharma, Janak Raj, and Puneet Gupta. "An efficient fifth order method for solving systems of nonlinear equations." Computers & Mathematics with Applications 67.3 (2014): 591-601. http://dx.doi.org/10.1016/j.camwa.2013.12.004

[31] Liu, Zhongli, and Quanyou Fang. "A new Newton-type method with third-order for solving systems of nonlinear equations." Journal of Applied Mathematics and Physics 3.10 (2015): 1256.

[32] Noor, Muhammad Aslam, and Muhammad Waseem. "Some iterative methods for solving a system of nonlinear equations." Computers & Mathematics with Applications 57.1 (2009): 101-106.

[33] Cordero, Alicia, and Juan R. Torregrosa. "Variants of Newton's method for functions of several variables." Applied Mathematics and Computation 183.1 (2006): 199-208.

[34] Crisfield M. A., A Fast Incremental/Iterative Solution Procedure that Handles ″Snap-through″, Computers and Structures, vol. 13, pp. 55-62, 1981.

[35] P. E. Gill, W. Murray, and M. H. Wright, Practical Optimization, Academic Press, London, 1981.