Macro-modeling of Micro-electrical-mechanical System Devices

by

Qiutao Wang

Bachelor of Science, Peking University, 1994

Submitted to the Department of Mechanical Engineering in partial fulfillment of the requirements for the degree of Master of Science in Mechanical Engineering at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

June 1998

© Massachusetts Institute of Technology 1998. All rights reserved.

Signature of Author ________________________________

Department of Mechanical Engineering

May 8, 1998

Certified by ________________________________

Jacob White
Professor of Electrical Engineering and Computer Science
Thesis Supervisor

Accepted by ________________________________

Ain A. Sonin
Chairman, Departmental Committee on Graduate Students

04/1998
Macro-modeling of Micro-Electrical-Mechanical System Devices

by

Qiutao Wang

Submitted to the Department of Mechanical Engineering on May 8, 1998, in partial fulfillment of the requirements for the degrees of Master of Science in Mechanical Engineering and Master of Science in Electrical Engineering and Computer Science

Abstract

Efficient performance verification of systems that include micromachined components requires low-order macro-models that accurately reflect micro-machined device performance. This thesis extends the Arnoldi method for model order reduction, commonly used for circuits interconnect problems, to a coupled domain micromechanical problem. A linear model for a highly non-linear micro-electrical-mechanical system (MEMS) model is proposed. Then, the extraction of reduced order models is demonstrated. The result shows that the linear model proposed matches the non-linear solution, which was verified by experimental data at the linear region of the structure, and reduced-order models with very low orders capture the dynamics of the linear model accurately. The model order reduction algorithm proposed in this thesis is generally applicable to MEMS devices that operate in a linear region.

Thesis Supervisor: Jacob White
Title: Professor of Electrical Engineering and Computer Science
Acknowledgments

First and foremost, I would like to express my greatest gratitude to my research advisor, Professor Jacob White, for his support and guidance. His patience and acute insights to problems inspire me to explore the unknown world with persistency and confidence.

Next, I want to thank all the members of the VLSI-CAD group on the eighth floor who have provided me such a warm environment that I feel doing research is fun and the office is as comfortable as home. In particular, I would like to thank Mike Chou, my guide to research and my great pal, who has not only helped me with my research but also shown me what friendship means to a meaningful life. I would also like to thank Jurgen Singer for those stimulating discussions on both academic and non-academic topics. I must also thank Matt Kamon for being so patient and supportive who is always there to help.

I am grateful to my best friends, my all-time buddy Lei Wang and my first roommate in Ashdown Alen whose support gave me strength in those difficult times. My sincere gratitude also goes to Zhanfeng Wang, Susan Wang, George Chou, Yu Sui, Kathy Liu, Emerson Quan, Mary Chen, Min Wu and Bei Wang whom I have been lucky to know as great friends.

And my fondest gratitude goes to Rebecca Wen-Fei Xiong, without her unconditional support, I couldn’t have gotten this far.

In the end, I would like to thank Mom and Dad, who have turned so much older since a few years ago when I first left home. I dedicate this master thesis to them in the wish of soothing their worries over their spoiled young son.
To Mom and Dad, ...
# Contents

1 Introduction .......................... 9  
  1.1 Motivation ........................... 9  
  1.2 Problem Description ................. 10  
  1.3 Thesis Outline ...................... 11  

2 Formulation .......................... 13  
  2.1 Governing Equations .................. 13  
  2.2 Derivation of The Governing Equations 15  
  2.3 Linearization ....................... 16  

3 State Space Representation and Discretization ............. 18  
  3.1 State Space Representation .......... 18  
  3.2 Discretization ...................... 19  
    3.2.1 Discrete States Variables .......... 19  
    3.2.2 Boundary Conditions .............. 20  
    3.2.3 Discrete Linear Operator .......... 21  
    3.2.4 Discrete State Space Representation 25  

4 Model Order Reduction Using the Arnoldi Method ........... 28  
  4.1 Overview of Model Order Reduction .......... 28  
  4.2 Model Order Reduction using The Arnoldi Method ........ 30  
    4.2.1 Moment Matching and Ill-conditioning 30  
    4.2.2 The Arnoldi Method in Model Order Reduction 31  
    4.2.3 More on The Arnoldi Method .......... 35  
  4.3 Some Important Numerical Issues .......... 35  
    4.3.1 Computation Work .................. 35  

5
4.3.2 Scaling Problem 

5 Results
5.1 Time Response
5.2 Frequency Response

6 Conclusion
List of Figures

1-1 A schematic of a widely used MEMS structure. The upper plate is flexible and clamped on two edges. It vibrates when a voltage waveform is applied. The fluid film between the gap damps the motion of the plate. .................................................. 10

2-1 fixed-fixed beam pull-in experiment setup. The x and z axes are parallel to the length and width of the beam, respectively. And y axis is perpendicularly pointing into the page. ................................. 14

3-1 Finite difference discretization of the top plate. A \((N + 1) \times (M + 1)\) grid results in N inner grid points along \(i\) or \(x\) direction, and M inner grid points along \(j\) or \(y\) direction. .................................................. 20

3-2 Finite difference discretization of a beam with length \(l = 100\) and step \(\Delta x = 1\). .............................................................. 21

3-3 Five-point centroid finite difference scheme for Laplace operator. ... 23

3-4 Discrete Laplace Matrix operator \(L\). (Bold dash lines represent the same elements on the diagonal line within one sub-block, and bold dot lines stand for the many sub-blocks that are not shown.) ............... 24

3-5 The structure of matrix \(A\) from an example of mesh size of \(10 \times 5\). 26

4-1 \(A^{-k}b\) and \(A^{-k-1}b\) converging to the dominant eigenvector of \(A^{-1}\) makes the direct evaluation of the power series \(A^{-k}b\) ill-conditioned. 32

5-1 A comparison of the time responses of the non-linear model, linear model and 4th order reduced model at small voltage input \(V = 0.1\ volts\). The center point displacement is plotted against time. ................. 40
5-2 At $V = 2\text{volts}$, the linear model starts to deviate from the non-linear model. The reduced-order model still follows the linear model.  

5-3 A comparison of the time responses of the non-linear model, linear model and 4th order reduced model at small voltage input $V = 0.1\text{volts}$. The center point deflection is plotted against time.  

5-4 A comparison of the frequency responses of the full linear system, 2nd order reduced model and 10th order reduced model.  

5-5 A first 10 poles of the 20th order of reduced-order model captures those of the original 70th order linear system. The $q = 2$ macro-model captures the first two poles on the real axis.
Introduction

1.1 Motivation

The progress in integrated circuit technology has enabled the fabrication of micro-electrical-mechanical systems (MEMS), such as micro-accelerometers, pressure sensors, gyroscopes, micro-switches, micro-mirrors and resonators [1]. In the rapidly growing MEMS industry, the wide variety of micromachined devices has made it impossible to rely on the time-consuming and cost-ineffective "trial and error" design methodology. Instead, efficient computer aided design (CAD) technologies become increasingly important in the design of MEMS systems.

The design of large MEMS systems requires the ability to simulate the dynamics of many interacting devices involving multiple coupled energy domain and media. Finite element method (FEM) is a widely adopted approach in dynamical simulations. Many commercially available CAD softwares, which are based on FEM theory, can handle simulations of complex geometries and coupled domains very well. However, to achieve a required accuracy, FEM uses high resolution volume discretizations and usually generates large systems that require long time and large memory to solve. Therefore, the FEM approach is impractical in many cases.

Macro-Modeling techniques allow us to generate a much smaller representation to describe the characteristics of the dynamics of a certain MEMS device accurately. Using reduced-order models generated by macro-modeling algorithms, one can carry out any further simulation involving the same device. The macro-modeling strategy could also be used for system level design where it may be necessary to run numerous
Figure 1-1: A schematic of a widely used MEMS structure. The upper plate is flexible and clamped on two edges. It vibrates when a voltage waveform is applied. The fluid film between the gap damps the motion of the plate.

simulations of a system with many devices.

1.2 Problem Description

In order to illustrate the macro-modeling techniques, we examine the dynamics of a typical MEMS structure, see Figure 1-1. The top plate is flexible and clamped on both sides. The bottom plate is fixed. The voltage actuates the flexible top plate electrostatically. As the top plate vibrates, it will squeeze out or suck in the air between the gap. The back pressure force of the air dampens the plate motion. This dynamic model is familiar to us as the squeeze-film damping structure in macro scale [2].

This model represents a coupled-domain problem and is highly non-linear when a large deformation solution is of interest. There is no analytical solution known to this problem. Therefore numerical solutions are necessary. In order to avoid costly repetitive full scale numerical solving, one can devise macro-models based on a set of valid assumptions for a specific application. In most of the applications, the ratio of length to thickness of the device is very large: \(\text{length} \gg \text{thickness of the gap}\), and the working fluid between the gap is a thin film of ideal gas under isothermal conditions. For a small amplitude oscillation, Langlois [2] derived a linear model of the air film. Based on a lumped spring-damper model [3] with flexible moving boundary, Yang, Gretillat and Senturia [4] showed that a macro-model consists of the first dominating modal shape of the beam produces accurate results of quality factors and resonant frequencies of a micro relay. Then, Hung, Yang and Senturia [5]
demonstrated a numerical macro-model extraction algorithm for the time-dependent non-linear dynamics of the structure shown in Figure 1-1. This algorithm, which is mathematically equivalent to the Karhunen-Loeve approach [6, 7], uses Galerkin methods to decompose the non-linear equations into a set of given basis functions. The pull-in dynamics of macro-models of very low orders generated by this algorithm matches the full finite difference model and experimental results, which implies great speed up.

However, these approaches require some knowledge about the characteristics of the device as a priori. When changes of a MEMS device design are frequent, the initial computing cost may become a constraint. This thesis presents a new approach that automatically extracts macro-models for MEMS devices that operate in linear region. There are many MEMS devices, such as accelerometers, resonator, gyroscopes, pressure sensor and etc., that have linear working regions in which they operate with a small input signal and therefore a small amplitude output. For the clamped-clamped plate example in this thesis, we linearize to the non-linear coupled governing equations used in [5]. We then applied spatial discretization schemes to generate a system of linear dynamic ordinary differential equations in the state space representation. For a high resolution discretization, the number of state variables is greater than a few thousand making it too expensive to use in higher level simulation. Motivated by the success of model order reduction techniques used in electrical engineering applications [8, 9], we use the Arnoldi method to compute a reduced-order model, or macro-model. The Arnoldi method computes a specified order of a macro-model by matching the moments of truncated Taylor series of the transfer function. The Arnoldi method also guarantees the stability of the reduced-order model.

The results show that for a fine discretization, a macro-model of very low order generated using the Arnoldi method accurately describes both the characteristics of the frequency response and the time domain behavior of the original large linear system.

1.3 Thesis Outline

In Chapter 2, the first section presents the coupled governing equations. Then, the second section describes the derivation of these equations. The last section shows the linearization of the original model. In Chapter 3, the first section describes the
transformation of the linear model into state space representation. The second section describes the discretization using finite difference schemes. Chapter 4 covers model order reduction and the Arnoldi method. The first section overviews model order reduction techniques. The second section explains the procedure of generating reduced-order model using the Arnoldi method. The last section of Chapter 4 deals with a few important issues that emerge from numerical analysis of this problem. The results are in Chapter 5. Chapter 6 presents the conclusion.
Figure 2-1: fixed-fixed beam pull-in experiment setup. The $x$ and $z$ axes are parallel to the length and width of the beam, respectively. And $y$ axis is perpendicularly pointing into the page.

We use the same formulations as Hung et al [5] to model the squeeze-damping structure shown in Figure 2-1. The model has of two equations. The first one is an 1D Euler beam equation (2.1) that comprises the inertia effect, the elastic bending effect, the residual stress effect, squeeze-film damping force and electrostatic force. The second equation is the 2D Reynold’s equation that models the compressible isothermal air film with slip flow effect included (2.2). They are shown as follows,

\[
EI \frac{\partial^4 u}{\partial x^4} - S \frac{\partial^2 u}{\partial x^2} = F_{elec} + \int_0^w (p - p_a) dz - \rho \frac{\partial^2 u}{\partial t^2} \tag{2.1}
\]

\[
\nabla \cdot (u^3 p \nabla p) = \frac{12\mu}{1 + 6K} \frac{\partial (pu)}{\partial t}, \tag{2.2}
\]

where $\nabla = \frac{\partial}{\partial x} i + \frac{\partial}{\partial y} j$ is the gradient operator. In the equations shown above, $F_{elec} \approx -\frac{eqV^2}{2a^2}$ is the electrostatic actuation force produced by the applied voltage input $V = V(t)$. $u = u(x,t)$ is the height of the beam above the substrate (or the difference of the unperturbed gap distance and the displacement of the 1D beam), and $p(x, y, t)$ is the pressure distribution in the fluid. Since the air film between the gap is very thin, we can assume pressure is uniform across the film in the direction of $z$ [2].

For reference purpose, we list the same set of parameters used in [5] as follows.

The dimensions of the beam are: length $l = 610 \mu m$, width $w = 40 \mu m$ and height $h = 2.2 \mu m$, the distance of the unperturbed gap is $d = 2.3 \mu m$; The material properties of the beam are: Young's module of the beam $E = 149 GPa$, moment of inertia $I = wh^3/12$, residual stress coefficient $[S/(hw)] = -3.7 MPa$, density $[\rho/(hw)] = 2330 kg/m^3$ (where $\rho$ is the line density); The properties of the air are: viscosity $\mu = 1.82 \times 10^{-5} kg/(m \cdot s)$, Knudsen's number $K(x, t) = \lambda/u$ where $\lambda$ is the mean-free
path of air and equal to 0.064 \mu m. (Knudsen’s number determines the proper regime the flow should be modeled as. Here, Knudsen’s number is \( K \cong 0.028 \). According to Yang [10] the air flow here is in a slip flow regime. From the Appendix of [11], we found the effective viscosity \( \mu^* = \frac{\mu}{1+6K} \). Outside the gap, the ambient air pressure is \( p_a = 1.013 \times 10^5 Pa \). The permittivity of air is \( \varepsilon_0 = 8.854 \times 10^{-12} F/m \). The Euler beam equation (2.1) is an integro-differential equation. It is linear in the dependent variables \( u(x, t) \) and \( p(x, y, t) \). The Reynolds equation (2.2) is a parabolic partial differential equation, and non-linear in the dependent variables. The system has three independent variables, namely, \( x, y, t \). The two equations are coupled as that the beam deforms downwards, squeezes out the air between the gap; meanwhile, the air acts on the beam through the back pressure force, in which there are both damping and spring components [4]. Besides the non-linearity of the dependent variables in the Reynolds equation (2.2), the electrostatic force and the Knudsen’s number are also non-linear in \( u \).

2.2 Derivation of The Governing Equations

The Euler beam equation (2.1) is essentially an expressive form of Newton second law with four force terms, namely, the elasticity bending force of the beam, the residual stress, the electrostatics actuation force, and the damping force from the air pressure.

The Reynolds equation (2.2) is derived from the Navior-Stokes equation of fluid. The derivation of it is rather involved. However, since the derivation of non-linear Reynold’s equation (2.2) reveals the underlying physical assumptions and understanding it better prepares us for further derivations of a linear model. Therefore, we briefly describe the major steps of Langlois’s derivation [2]. Although as we mentioned above the air flow between the very narrow gap is in slip flow region, the Navier-Stokes equations still apply but with a modified viscosity \( \frac{\mu}{1+6K} \), usually referred to as the effective viscosity [10]. In a non-dimensional analysis of the Navier-Stokes equations, we pick the characteristic constants of length, width and frequency to scale independent variables, and the unperturbed gap distance, the ambient pressure to scale dependent variables. We can see that the ratio of the unperturbed gap distance to the characteristic width of the beam is a very small number. By neglecting terms with the second power in this small quantity, the variation of pressure along z direction van-
ishes from the Navier-Stoke equations. It confirms the assumption of the pressure being constant across the thin film of the air. The assumption of the air being an ideal gas under isothermal conditions allows us to replace the density variable with the pressure. Then, we combine the boundary equations, (which says the velocity of the air on the contact surface with the top beam is the same as the velocity of the moving beam and zero for the bottom beam,) and the continuity equation, then plug them into the Navier-Stokes equations. By assuming a low Reynolds number which implies the fluid inertia and dilational stresses are negligible, we arrived at the final form of the non-linear Reynolds equation (2.2). To repeat, the fluid we consider here is a very thin film of isothermal gas with inertia and the dilation stresses neglected under the small Reynolds number assumption.

2.3 Linearization

The system of the coupled non-linear governing equations (2.1-2.2) has no analytical solution available. In order to solve for the dynamics, a full numerical treatment to these equations is necessary. A finite difference solution of the non-linear coupled problem is available from Hung et al [5]. However, many MEMS devices with a similar structure as the one in Figure 2-1, operate in a linear region. In other words, the actuation signal or the disturbance force is so small that the response of the system in the form of beam displacement is very small comparing to the gap distance. Thus, we can linearize the dependent variables in equations (2.1) and (2.2) around their initial equilibrium point. We assume

\[ u = u_0 + \delta u \]
\[ p = p_0 + \delta p \]  

(2.3)

with \( u_0 = g \) where \( g \) is the unperturbed the gap distance and \( p_0 = p_a \) where \( p_a \) is the ambient pressure, \( \delta u \) and \( \delta p \) are perturbations. If we substitute equation (2.3) into the Reynold's equation (2.2) and drop terms of second and higher order in \( \delta u \) and \( \delta p \), we get

\[ u_0^3 p_0 \nabla^2 p = \frac{12\mu}{1 + 6K}(\frac{u_0 \partial \delta p}{\partial t} + \frac{p_0 \partial \delta u}{\partial t}). \]  

(2.4)

Here we assume \( \nabla u_0 = 0 \) and \( \nabla p_0 = 0 \).

A complication here is worthwhile for some attention. If \( u_0 \) and \( p_0 \) are not the initial states of zero voltage input but some other steady state equilibrium point
that \( u_0 \) and \( p_0 \) are some functions of spatial independent variables, then during the linearization, many terms with the gradients, \( \nabla u_0 \) and \( \nabla p_0 \) won’t vanish. As a result, the linear equation (2.4) will look much more complicated but still linear in \( \delta u \) and \( \delta p \). However, we believe that the analysis we will illustrate next should still apply. Exploiting this point further would allow us to extend this work to solve for linear perturbations around non-zero steady states of the system. For simplicity, we restrict ourselves with the initial equilibrium state, or equivalently, \( \nabla u_0 = 0 \) and \( \nabla p_0 = 0 \).

A perturbation analysis for the Euler beam equation (2.1) is rather straightforward, since it is already linear. The linearization of the parameter \( K \) and the electrostatic force term \( F_{elec} \) follows naturally that, \( F_{elec} = -\frac{\varepsilon_0 w y^2}{2(u_0 + \delta u)^2} \approx -\frac{\varepsilon_0 w y^2}{2g^2} \). In the same manner, \( K = \frac{\lambda}{u} \approx \frac{\lambda}{g} \).

We can normalize the linearized Reynold’s equation (2.4) using \( g \) and \( p_a \) as

\[
\begin{align*}
\tilde{u} &= \frac{\delta u}{g} \\
\tilde{p} &= \frac{\delta p}{p_a}.
\end{align*}
\]  

(2.5)

If we replace \( \tilde{u}, \tilde{p} \) with \( u, p \) for the simplicity of notation, we have the normalized linear Reynold’s equation as

\[
g^2 p_a \nabla^2 p = \frac{12\mu}{1 + 6K} \left( \frac{\partial p}{\partial t} + \frac{\partial u}{\partial t} \right). \tag{2.6}
\]

This equation resembles the equation of small periodic variation of the gap between infinitely long parallel plates that Langois has derived in [2].

We perform the same operation to the Euler beam equation (2.1), obtain the normalized Euler beam equation,

\[
EI \frac{\partial^4 u}{\partial x^4} - S \frac{\partial^2 u}{\partial x^2} = F_{elec}^* + \frac{p_a}{g} \int_0^w p dy - \rho \frac{\partial^2 u}{\partial t^2}, \tag{2.7}
\]

where \( F_{elec}^* = -\frac{\varepsilon_0 w y^2}{2g^2} \) is the linear actuation force.
State Space Representation and Discretization

In the previous chapter we derived a linear model. In this chapter we will change the linear model into matrix form, so that we can apply model order reduction in the next chapter. In the following sections, we first transform the linear model into standard state space representation. Then, we carry out the discretization using finite difference schemes.

3.1 State Space Representation

For a dynamical problem, if we define the state variables properly, a system with $n$ states can be represented by $n$ first-order differential equations, which may be combined into a first-order vector-matrix differential equation. The use of vector-matrix notation greatly simplifies the mathematical representation of the system of equations. The increase in the number of state variables, which means increasing the resolution of the discretization mesh in our case, does not increase the complexity of the equations. The state space representation enables the computation of the evolution of the states over time.

First, we rearrange the normalized linear the Euler beam equation (2.7) and the Reynold’s equation (2.6) into

$$\frac{\partial^2 u}{\partial t^2} = -\left(\frac{EI}{\rho}\right) \frac{\partial^4 u}{\partial x^4} + \left(\frac{S}{\rho}\right) \frac{\partial^3 u}{\partial x^3} + \frac{F_{elec}}{\rho} + \frac{p_a}{\rho g} \int_0^w pdy \quad (3.1)$$
\[
\frac{\partial p}{\partial t} = \frac{(1+6K)g^2p_a}{12\mu} \nabla^2 p - \frac{\partial u}{\partial t}. \tag{3.2}
\]

In the equations above it is natural to choose position, velocity and pressure as three different states. Thus, the state variables are defined as follows,

\[
x_1 = u, \quad x_2 = \frac{\partial u}{\partial t}, \quad x_3 = p. \tag{3.3}
\]

After change of variables, we get the state space representation of the linear dynamical system,

\[
\dot{x}_1 = x_2, \tag{3.4}
\]

\[
\dot{x}_2 = \left[-\left(\frac{EI}{\rho}\right) \frac{\partial}{\partial x^4} + \left(\frac{S}{\rho}\right) \frac{\partial^2}{\partial x^2}\right]x_1 + \left[\frac{p_a}{\rho g}\right] \int_0^w dy]x_3 + \frac{c_0 w V^2}{2g^3 \rho}, \tag{3.5}
\]

\[
\dot{x}_3 = -x_2 + \left[\frac{(1+6K)g^2p_a}{12\mu} \nabla^2\right]x_3, \tag{3.6}
\]

where \(\dot{x}_i = \frac{\partial x_i}{\partial t}, \ i = 1, 2, 3.\)

However, we need to find the discrete forms for the linear operator as well in order to take advantage of the vector-matrix representation.

### 3.2 Discretization

In this thesis we use finite difference for the discretization. Since we are dealing with rather simple geometry, finite difference is sufficiently accurate for the illustration purpose of this thesis. For structures with more complicate geometries, we need more robust discretization techniques, such as the finite element method.

#### 3.2.1 Discrete States Variables

For the rectangular top plate in Figure 1-1, we apply a \((N+1) \times (M+1)\) meshing, see Figure 3-1. Note that the grid-spacings along \(x\) and \(y\) axes are \(\Delta x = \frac{\text{length}}{N+1}\) and \(\Delta y = \frac{\text{width}}{M+1}\), respectively. For each of the three states, we specify one discrete variable
Figure 3-1: Finite difference discretization of the top plate. A \((N+1) \times (M+1)\) grid results in \(N\) inner grid points along \(i\) or \(x\) direction, and \(M\) inner grid points along \(j\) or \(y\) direction.

per inner point grid. The pressure state \(\mathbf{x}_3\) in (3.4-3.6) becomes,

\[
\mathbf{x}_3 = p_{M \times N} = \begin{bmatrix}
    p_{11} & p_{12} & \cdots & p_{1N} \\
    p_{21} & p_{22} & \cdots & p_{2N} \\
    \vdots & \vdots & \ddots & \vdots \\
    p_{M1} & p_{M2} & \cdots & p_{MN}
\end{bmatrix}
\]  

(3.7)

Since the beam equation (2.7) is treated as 1D, the position and velocity states extend uniformly along \(j\) direction from \(y = 0\) to \(y = \omega\). Thus, both states discretize into vectors of \(\mathbb{R}^N\),

\[
\mathbf{x}_1 = \mathbf{u} = \begin{bmatrix}
    u_1 \\
    u_2 \\
    \vdots \\
    u_N
\end{bmatrix}_{N \times 1}, \quad \mathbf{x}_2 = \dot{\mathbf{u}} = \begin{bmatrix}
    \dot{u}_1 \\
    \dot{u}_2 \\
    \vdots \\
    \dot{u}_N
\end{bmatrix}_{N \times 1}
\]  

(3.8)

3.2.2 Boundary Conditions

At the boundary grid points in Figure 3-1, variables are known from the boundary conditions specified. In Figure 2-1, the setup of the fixed-fixed beam determines the boundary conditions for the beam equation that the displacements at both fixed ends
Figure 3-2: Finite difference discretization of a beam with length \( l = 100 \) and step \( \Delta x = 1 \).

are zero, and so the deviation angles,

\[
\begin{align*}
    u(0, t) = u(l, t) &= 0, \\
    \frac{\partial u(x, t)}{\partial x} \big|_{x=0} = \frac{\partial u(x, t)}{\partial x} \big|_{x=l} &= 0. \\
\end{align*}
\]  

(3.9)

The pressure is equal to the ambient pressure along the two edges that are parallel to \( x \) axis. And a zero-flux condition is applied to the two clamped edges that are parallel to \( y \) axis, which essentially says, there is no flow in the direction normal to the boundary. They take the form as,

\[
\begin{align*}
    p(x, 0, t) = p(x, l, t) &= p_a, \\
    \frac{\partial p(x, y, t)}{\partial x} \big|_{x=0} = \frac{\partial p(x, y, t)}{\partial x} \big|_{x=l} &= 0. \\
\end{align*}
\]  

(3.10)

3.2.3 Discrete Linear Operator

Now, we need to find discrete forms of the linear operators appearing in equation (3.1) and (3.2). We choose to use central finite difference schemes for spatial differential operators, and the Trapezoidal rules for the integral operator. At this intermediate step, different schemes can be chosen according to different applications.

Figure 3-2 shows an example of the discretization of the 1D beam with length \( l = 100 \) and step \( \Delta x = 1 \). The dynamic equation of the 1D beam (3.1) contains a fourth order differentiation operator with respect to \( x \), a second order one and a integrator as well. The fourth order central difference scheme for \( \frac{\partial^4 u}{\partial x^4} \) is,

\[
u_i^{(4)} = \frac{1}{(\Delta x)^4}(u_{i-2} - 4u_{i-1} + 6u_i - 4u_{i+1} + u_{i+2}), \quad i = 3, 4, \cdots, 97.
\]  

(3.11)

As always, we need to pay special attention to equations involving the boundary points. Since the Neumann condition takes zero at both ends, it implies a symmetry
in the solution on both sides of \( x = 0 \), therefore \( u_{-1} = u_1 \) and \( u_{-2} = u_2 \). It is the same for \( x = 100 \). Thus, for \( i = 1, 2 \), equation (3.11) becomes,

\[
\begin{align*}
  u_1^{(4)} &= \frac{1}{(\Delta x)^4}(7u_1 - 4u_2 + u_3) \\
  u_2^{(4)} &= \frac{1}{(\Delta x)^4}(-4u_1 + 6u_2 - 4u_3 + u_4)
\end{align*}
\]  

(3.12) (3.13)

respectively. The same treatment applies for \( i = 98, 99 \) as well. If we define vector

\[
u = \begin{bmatrix} u_1 \\ u_2 \\ \vdots \\ u_{99} \end{bmatrix},
\]

(3.14)

then the term \( \frac{\partial^4 u}{\partial x^4} \) can be represented in a matrix operator form, \( Mu \) where

\[
M = \frac{1}{(\Delta x)^4} \begin{bmatrix}
7 & -4 & 1 & \cdots & 0 \\
-4 & 6 & -4 & \ddots & \vdots \\
1 & -4 & 6 & \ddots & \vdots \\
\vdots & \ddots & \ddots & \ddots & -4 \\
0 & \cdots & \cdots & -4 & 7
\end{bmatrix}.
\]

(3.15)

\( M \) is a five-banded \( N \times N \) matrix. This scheme is accurate to \( O(\Delta x^2) \), see [13]. Similarly, the term of \( \frac{\partial^2 u}{\partial x^2} \) in equation (3.1) becomes \( Su \), where \( S \) is a three-banded matrix.

For the integral operator, we apply the Trapezoidal rule [14], which states that

\[
\int_0^w p dy \approx \Delta y \left( \frac{1}{2}p_0 + p_1 + p_2 + \ldots + p_M \right) + \frac{1}{2} p_{M+1})
\]

(3.16)

Because the boundary condition of pressure distribution (see equation (3.10)) is equal to ambient pressure, the normalized pressure perturbation is zero at both edges at \( y = 0 \) and \( y = w \). Therefore, for the \( i \)th column of grid points (see Figure 3-1), the matrix operator of the integral (which is the integration of perturbation pressure along the width at a fixed \( x \) or \( i \)) takes the form of \( T^T p_i \), where

\[
T = \Delta y \begin{bmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{bmatrix}_{M \times 1}, \quad p_i = \begin{bmatrix} p_{1i} \\ p_{2i} \\ \vdots \\ p_{Mi} \end{bmatrix}_{M \times 1}
\]

(3.17)
Figure 3-3: Five-point centroid finite difference scheme for Laplace operator.

In order to find the discrete form for the Laplace operators in equation (3.2), we need to decide on an ordering rule to change the discrete pressure states in (3.7) from 2D or \( \mathbb{R}^{M \times N} \) into 1D or \( \mathbb{R}^{MN \times 1} \). We choose to wrap \( p_{ij} \) along \( y \) direction, in other words, pile \( p_{j1}, p_{j2}, \ldots, \) and \( p_{jN} \) up sequentially,

\[
\mathbf{p} = \begin{bmatrix}
    p_{11} \\
    \vdots \\
    p_{M1} \\
    p_{12} \\
    \vdots \\
    p_{M2} \\
    \vdots \\
    \vdots \\
    p_{1N} \\
    \vdots \\
    p_{MN}
\end{bmatrix}_{MN \times 1}
\]  

(3.18)

Since the efficiency of numerical algorithms is always preferred whenever possible, we choose such an ordering that it renders a matrix that has a narrower band structure. Therefore, sparse Gaussian elimination algorithm is more efficient on such a matrix than the one generated by wrapping \( p_{ij} \) along \( x \) direction, because there will be less fill-in’s generated [15]. Now, we apply five-point centroid scheme [13] to the single vector of pressure states,

\[
\nabla^2 p_{ij} \approx \frac{1}{\Delta x^2}(p_{(i-1)j} - 2p_{ij} + p_{(i+1)j}) + \frac{1}{\Delta y^2}(p_{i(j-1)} - 2p_{ij} + p_{i(j+1)}),
\]  

(3.19)
Figure 3-4: Discrete Laplace Matrix operator L. (Bold dash lines represent the same elements on the diagonal line within one sub-block, and bold dot lines stand for the many sub-blocks that are not shown.)

see Figure 3-3 for an illustration of this scheme. Together with the boundary conditions, we obtain the discrete form of the Laplace operator, $\nabla^2 p \sim Lp$. Figure 3-4 shows that matrix L also has a banded structure, where $e = \frac{1}{\Delta x^2}$, $f^* = -\left(\frac{1}{\Delta x^2} + \frac{2}{\Delta y^2}\right)$, $f = -\left(\frac{2}{\Delta x^2} + \frac{2}{\Delta y^2}\right)$ and $g = \frac{1}{\Delta y^2}$. 
3.2.4 Discrete State Space Representation

In order to assemble all the pieces into state space representation, we define

\[
x = \begin{bmatrix} u \\ \dot{u} \\ p \end{bmatrix} = \begin{bmatrix} u_1 \\ \vdots \\ u_N \\ \dot{u}_1 \\ \vdots \\ \dot{u}_N \\ p_{11} \\ \vdots \\ p_{MN} \end{bmatrix}_{(2N+MN)\times 1}
\]

(3.20)

as the global state variables. Since voltage is the only input to the system (see Figure 2-1), we can define the input function as, \( v(t) = V(t)^2 \). Assuming that for an electrostatic voltage supply, the charges are uniformly distributed over the top plate, along with the linearization of the actuation force, we can further assume that the load is uniformly distributed over the 1D beam, see Figure 3-2. The output can be any linear combination of the state variables, e.g. the displacement of the center point of the beam.

With the discrete linear operators we derived previously, we are able to transform the continuous state space equations (3.4-3.6) into its discrete form,

\[
\dot{x}(t) = Ax(t) + Bv(t) \quad \text{(3.21)}
\]

\[
r(t) = CTx(t) \quad \text{(3.22)}
\]
Figure 3-5: The structure of matrix $A$ from an example of mesh size of $10 \times 5$.

where

$$
A = \begin{bmatrix}
\theta_{N \times N} & I_{N \times N} & \theta_{N \times NM} \\
[aM + bS]_{N \times N} & \theta_{N \times N} & [c\Delta y \ \cdots \ c\Delta y]_{1 \times M} \\
\theta_{NM \times N} & -1 & -1 \\
& \vdots & \vdots \\
& -1 & -1 \\
& & -I_{M \times 1} \\
& & & dL_{NM \times NM} \\
& & & cT^T
\end{bmatrix}
$$

(3.23)

Note that in the matrix $A$, the matrices $M$, $S$, $T^T$ and $L$ are the linear operators defined in the previous section and $a$, $b$, $c$ and $d$ are the coefficients in front of each of the operators in equation (3.4-3.6). The $cT^T$'s and the $-I_{M \times 1}$'s carry the coupling effect between the beam and the air film. Figure 3-5 shows the matrix $A$ of a $10 \times 5$ mesh size example. Apparently, the matrix $A$ has an important structure i.e. sparsity.
If we supply a step function of $V(t)$ as voltage input, then

$$
B = F[0 \ldots 0 \underbrace{1 \ldots 1}_{{(N+1)\sim 2N}} 0 \ldots 0]_{1 \times (2N+NM)}^T
$$

(3.24)

where $F = -\frac{\epsilon_0 \omega V^2}{2g^2 \rho}$. For the output, it is convenient to choose the center point displacement of the beam as the output function, which means $C = e_{\lfloor \frac{N}{2} \rfloor}$ where $e_k$ is a unit vector and $e_k \in \mathbb{R}^{(2N+NM)}$. 
Model Order Reduction Using the Arnoldi Method

In the previous chapter, we derived a linear model and its discrete state space representation. In this chapter, we discuss the numerical analysis that is involved in the model order reduction of a large linear system. The first section introduces some previous approaches. Then, the second section illustrates the mechanics of the Arnoldi method. In the last section of this chapter, we deal with some numerical issues which arise when solving the linear system.

4.1 Overview of Model Order Reduction

In the last decade, the topic of macro-modeling, or model order reduction, became very popular in many disciplines of electrical engineering [8], such as control engineering, circuit simulation [9]. However, it is rarely applied to mechanical engineering areas, particularly to structure dynamics or fluid dynamics. Some recent work of MIT’s MEMCAD was ground breaking in this respect. Yang et al [4] presented a methodology for extracting macro-model parameters of the aforementioned structure (see Figure 2-1), assuming a small amplitude oscillation. The proposed method uses a lumped-element macro-modeling treatment of the flexible beam. In other words, the flexible beam is approximated by using its first modal shape. The fluid is modeled by a linearized Reynold’s equation with a moving boundary. The extracted macro-model produces good results for the quality factor and the resonant frequencies. The modal shape of interest is obtained from commercial finite-element codes.
Another approach proposed by Hung and Senturia [5] generates macro-models directly from the non-linear model (see equations 2.1-2.2). In this approach, an arbitrary number of spatial basis functions are extracted orthogonally from time simulation data. The unknowns can then be projected onto the spatial basis functions to form a separable form of approximating series expansion, in which the functions of time and the functions of spatial coordinates are separated. Then, a Galerkin method is used to generate the optimal coefficients which are functions of time only. As a result, the system of the non-linear partial differential equations transforms into a system of ordinary differential equations whose order is on par with the number of basis functions used. This method is mathematically equivalent to the Karhunen-Loeve approach [6, 7], only more robust. This algorithm has excellent performance for the full range of non-linear behavior in the time domain. These two approaches are quasi-automatic in the sense that they only require solving the full system of equations just once.

In this thesis, we propose an automatic macro-model extraction approach. The motivation comes from the success of model order reduction techniques in electrical engineering applications. In control engineering, some well known eigenvalue methods of model order reduction approaches, such as Truncated Balanced Realizations (TBR) [16], Hankel Singular Values (HSV) method [17], Selective Modal Analysis (SMA) [18], etc. are very popular in solving problems that have a small number of state variables. Unfortunately, these direct eigenvalue methods require $O(n^3)$ computation, because they require explicit knowledge of the entire eigenspectrum of the system. For a large discretization, the number of discrete state variables $n$ can easily exceed $10^4$. These methods would require days to compute and gig-bytes of memory.

In the area of circuit simulation, Asymptotic Waveform Evaluation (AWE) [19] and Padé-via-Lanczos (PVL) [20] has popularized the use of the Padé approximation [21] for model order reduction. However, if we compute Padé approximates by direct evaluation of the moments, the computation becomes very ill-conditioned when the order of approximation is high. Arnoldi based methods solve this ill-conditioning problem by constructing orthogonal bases of Krylov subspaces. In addition, the Arnoldi method guarantees the stability of the reduced-order model. We explain the Arnoldi method in detail in the next section.
4.2 Model Order Reduction using The Arnoldi Method

4.2.1 Moment Matching and Ill-conditioning

From the system dynamics point of view, the state space representation in the system of equations of (3.21-3.22) is a Single-Input-Single-Output (SISO) linear, time-invariant (LTI) system. This LTI system is of order \( n = 2N + N \times M \), where \( N \) and \( M \) are the discretization size of the plate in \( x \) and \( y \) direction, respectively. Transforming the state space representation into frequency domain greatly simplifies the solving of LTI systems. The Laplace transform of (3.21-3.22) is,

\[
\begin{align*}
    sX(s) & = AX(s) + BV(s) \\
    R(s) & = C^T X(s)
\end{align*}
\]

where \( X(s) \), \( V(s) \) and \( R(s) \) denote the Laplace transforms of \( x(t) \), \( v(t) \) and \( r(t) \), respectively. The transfer function of a LTI system is defined as \( F(s) = \frac{R(s)}{V(s)} \). From the algebraic equations (4.1-4.2), it is trivial to get

\[
F(s) = C^T (I - A)^{-1} B.
\]

(4.3)

A straightforward manipulation of equation (4.3) shows that

\[
F(s) = -C^T (I - sA^{-1})^{-1} A^{-1} B = C^T (I - sA^{-1}) b
\]

(4.4)

where \( b = -A^{-1} B \), given that \( A \) is non-singular.

A \( q \)th reduced-order model for the SISO LTI system (3.21-3.22) would be,

\[
\begin{align*}
    \dot{x}_q & = A x_q + B_q w \\
    \tilde{r} & = C^T_q x_q
\end{align*}
\]

(4.5)

(4.6)

where \( x_q, B_q \) and \( C_q \in \mathbb{R}^q \), \( A_q \in \mathbb{R}^{q \times q} \) and \( q \) is presumably much smaller than \( n \). In term of the transfer functions, the task is to find \( A_q, B_q, C_q \) such that

\[
F_q(s) = \frac{\tilde{R}(s)}{V(s)} = C^T_q (I_q s - A_q)^{-1} B_q
\]

(4.7)

approximates the original transfer function \( F(s) \).
It was shown that the squeeze film damping structure (see Figure 2-1) mostly operates in low frequency range [4]. Thus, it is natural to expand the transfer function into Taylor series at $s = 0$, which gives

$$F(s) = C^T(I - sA^{-1})^{-1}b = C^T(I + sA^{-1} + s^2A^{-2} + \ldots)b = \sum_{k=0}^{\infty} m_k s^k,$$  \hspace{1cm} (4.8)

where

$$m_k = C^T(A^{-k})b$$  \hspace{1cm} (4.9)

is called moments. Intuitively, we can use the truncated Taylor expansion to approximate the transfer function $F(s)$, which means we compute the $F_q(s)$ by matching the first $q$ moments of the expansion of $F(s)$ in (4.8). This is usually called the moment matching procedure.

In order to get a $q$th-order Padé approximation, one needs to compute the first $2q - 1$ moments by evaluating the power terms $A^{-k}b$ directly. Although the Padé method are straightforward in the computation of low order of approximations, for a higher order of approximation it won’t give accurate answers due to ill-conditioning. The reason is that, as $k$ increases, $A^{-k}b$ and $A^{-k-1}b$ quickly line up against each other and both converge to the dominant eigenvector of $A^{-1}$ and thus become linear dependent to numerical precision. This phenomena is illustrated in Figure 4-1 [9]. To avoid this problem, we shall use the Arnoldi method instead [22].

### 4.2.2 The Arnoldi Method in Model Order Reduction

In the moment matching procedure, we compute a vectors sequence, $A^{-k}b$, $k = 0, \ldots, q - 1$. These vectors span a Krylov subspace defined as

$$\mathcal{K}_q(A^{-1}, b) = \text{span}\{b, A^{-1}b, A^{-2}b, \ldots, A^{-(q-1)}b\}.$$  \hspace{1cm} (4.10)

The Arnoldi method overcomes the ill-conditioning problem by construction of a set of orthonormal bases for the Krylov subspace. The Arnoldi algorithm usually uses normalized $b$ as the starting Arnoldi vector $v_1 = \frac{b}{\|b\|_2}$. Then, it takes the next vector $A^{-1}v_1$ and projects it orthogonally onto $v_1$. The second Arnoldi vector is generated as $\bar{v}_2 = A^{-1}b - h_{11}v_1$ and $v_2 = \frac{\bar{v}_2}{\|\bar{v}_2\|_2}$, which is orthogonal to the first vector $v_1$. Let’s denote the projection as $h_{11} = \langle v_1, A^{-1}v_1 \rangle$, $h_{21} = \frac{v_2}{\|v_2\|_2}$. For the third Arnoldi
Figure 4-1: $A^{-k}b$ and $A^{-k-1}b$ converging to the dominant eigenvector of $A^{-1}$ makes the direct evaluation of the power series $A^{-k}b$ ill-conditioned.

vector, we take $A^{-2}v_2$, which is essentially a linear combination of the first three Krylov subspace basis vectors $b$, $A^{-1}b$ and $A^{-2}b$, and project it onto the previous two Arnoldi vectors $(v_1, v_2)$ and take the difference between itself and the sum of the two projected vectors. By normalizing the resulting vector, we obtain the third Arnoldi vector. If we continue this iterative procedure for $q$ steps, we will obtain a set of orthonormal vectors $v_i, i = 1, \ldots, q$, which is also called an Arnoldi basis. It is easy to show that the Arnoldi bases span the same Krylov subspace in (4.10). The algorithm below shows the outline of a modified Arnoldi algorithm 4.2.1.
Algorithm 4.2.1 (Arnoldi process).

\[ \text{arnoldi}(\text{in: } A, b, q; \text{ out: } V_q, v_{q+1}, H_q, h_{j+1,j}) \]
\[
\begin{align*}
\{ \\
\quad v_1 &= b/\|b\|_2 \\
\quad &\text{compute sparse LU factorization of } A \\
\quad &\text{for } (j = 1; j <= q; j++) \{ \\
\quad &\quad \text{/* Get next vector in space */} \\
\quad &\quad \text{solve } Ly = v_j \\
\quad &\quad \text{solve } Uw = y \\
\quad &\quad \text{/* Orthogonalize new vector to previous vectors */} \\
\quad &\quad \text{for } (i = 1; i <= j - 1; i++) \\
\quad &\quad \quad h_{i,j} = w^Tv_i \\
\quad &\quad \quad w = w - h_{i,j}v_i \\
\quad &\quad \} \\
\quad h_{j+1,j} &= \|w\|_2 \\
\} \\
\quad V_q &= [v_1 \cdots v_q] \\
\quad H_q &= (h_{i,j}), \ i, j = 1, \cdots, q
\}
\]

If we collect the projection coefficients \( h_{ij} \) into a matrix, we get an upper Hessenberg matrix with one extra row at the bottom.

\[
\tilde{H}_q = 
\begin{bmatrix}
    h_{11} & h_{12} & \cdots & h_{1n} \\
    h_{21} & h_{22} & \cdots & h_{2n} \\
    h_{32} & \ddots & \ddots & \ddots \\
    \cdots & \ddots & h_{qq} \\
    h_{q+1,1} & \cdots & h_{q,1} & h_{q+1,q}
\end{bmatrix}_{(q+1) \times q}
\]  

(4.11)
From Algorithm 4.2.1, we see that $A$, $\tilde{H}_q$ and Arnoldi bases $v_i$'s are related as,

$$
A^{-1} \begin{bmatrix}
v_1 \\
\vdots \\
v_g
\end{bmatrix} = \begin{bmatrix}
v_1 \\
\vdots \\
v_{q+1}
\end{bmatrix} \begin{bmatrix}
h_{11} & h_{12} & \cdots & h_{1n} \\
h_{21} & h_{22} & \cdots & h_{2n} \\
h_{31} & \ddots & \ddots & \ddots \\
\vdots & & \ddots & \ddots \\
h_{q1} & \cdots & \cdots & h_{qq} \\
h_{q+1,1} & \cdots & \cdots & h_{q+1,q}
\end{bmatrix}
$$

(4.12)

Let $V_q$ be the $n \times q$ matrix whose columns are the orthonormal Arnoldi bases,

$$
V_q = \begin{bmatrix}
v_1 & v_2 & \cdots & v_q
\end{bmatrix}_{n \times q}
$$

(4.13)

If we split the Matrix (4.11) into a $q \times q$ upper Hessenberg matrix $H_q$ and a rank one matrix $v_{q+1} e_q^T$, then (4.12) becomes

$$
AV_q = V_q H_q + h_{q+1,q} v_{q+1} e_q^T,
$$

(4.14)

where $e_q$ is the $q$th column of the identity matrix in $\mathbb{R}^{q \times q}$ and matrix $H_q$ is the upper $q \times q$ square block of $\tilde{H}_q$. Since $v_1 = \frac{b}{\|b\|_2}$ and with repeated use of (4.14), it can be shown that after $q$ steps of the Arnoldi process, for $k < q$,

$$
A^{-k}b = \|b\|_2 A^{-k} V_q e_1 = \|b\|_2 V_q H_q^k e_1.
$$

(4.15)

With this relation, the moments (4.9) can be related to $H_q$ by

$$
m_k = C^T A^{-k} b = \underbrace{\|b\|_2 C^T V_q H_q^k}_{C_q^T A_q^{-k}} e_1.
$$

(4.16)

This derivation is the same as in [23] except that $A$ in [23] is replaced by $A^{-1}$ here.

Thus, we finally get the corresponding reduced-order model of the original state space representation (see 3.21-3.22) which is,

$$
\dot{x}_q(t) = A_q x_q(t) + B_q u(t)
$$

(4.17)

$$
\ddot{\tau}(t) = C_q^T x_q(t)
$$

(4.18)
where \( A_q = H_q^{-1} \), \( B_q = e_1 \) and \( C_q^T = \| b \|_2 V_q^T C \). Also, by analogy with (4.9), the \( q \)th order Arnoldi-based approximation to the original transfer function \( F(s) \) in (4.2.1) can be written down as,

\[
F_q(s) = \| b \|_2 C_q^T V_q (I - sH_q^t)^{-1} e_1. \tag{4.19}
\]

### 4.2.3 More on The Arnoldi Method

The Arnoldi method is an iterative eigenvalue method. In eigenvalue problems, noniterative or "direct" algorithms require \( O(n^3) \) work. This makes the simulation of large systems prohibitively expensive. In comparison, iterative methods exploit the special structure of a matrix, i.e. the sparsity, and reduce the work to \( O(n) \). Thus, the Arnoldi iterative algorithm is a good choice for model order reduction problems with matrices that are large but sparse.

Another good property of the Arnoldi method is that, it guarantees the stability of the reduced-order model [24]. A system is asymptotically stable if it has all its poles on the left half of the complex plane. Or in terms of the state space matrices \( A \) and its reduced model \( H_q^{-1} \), it means that given \( A \) is negative definite, the Arnoldi algorithm guarantees its approximate \( H_q^{-1} \) is also negative definite [24].

In this thesis, we expand the transfer function around zero frequency. As a result, the reduced-order model generated by the Arnoldi algorithm (Algorithm 4.2.1) captures poles that locate close to (0,0) on the left half of the complex plane. However, some applications require the expansion be at high frequencies. To these applications, we can apply the shifted Arnoldi method [25]. The Arnoldi method also can handle Multi-Input-Multi-Output (MIMO) problems [26].

### 4.3 Some Important Numerical Issues

#### 4.3.1 Computation Work

As we have shown in last chapter, the matrix \( A \) we are dealing with is sparse (see Figure 3-5). In the Arnoldi algorithm (see Algorithm 4.2.1), \( A^{-1} \) appears in the model order reduction process. Fully factoring this inverse of matrix \( A \) is computationally demanding. Fortunately, what we actually need is the matrix-vector product, e.g.
Let $y = A^{-1}b$, the problem transforms into the one of solving linear system $Ay = b$.

The order of the system that we are dealing with is usually around $10^3$. Thus, we can use a sparse LU algorithm to factor the matrix $A = LU$ once and store the lower and upper triangular matrices $L$ and $U$. Inside the Arnoldi loop (see Algorithm 4.2.1), we solve the two triangular systems, namely $Ly = v_i$ for $y$ and then $Uw = y$ for $w$ both require $O(n^2)$ work. Coincidentally, the matrices $L$ and $U$ are sparse also. Therefore, the total work for Algorithm 4.2.1 is $O(n)$. However, in most cases LU algorithm creates fill-ins and destroys the sparsity of sparse matrices. Thus, it usually requires $O(n^3)$ work which should be avoided since $n$ can be very large. In case of high resolution discretizations which generate large linear systems, we can consider iterative linear system solving algorithms, such as GMRES whose work is $O(n)$ [27].

### 4.3.2 Scaling Problem

The matrix $A$ generated by the discretization procedure described in Chapter 3, is actually very ill-conditioned. Let’s define the condition number of matrix $A$ as

$$
\kappa(A) = \|A^{-1}\|_2 \|A\|_2
$$

For a very coarse $4 \times 3$ discretization, the system we defined in (3.21-3.22) generates a relatively small state space matrix $A$ of a size of $20 \times 20$. However, it is highly ill-conditioned, $\kappa(A) \sim 10^{13}$. It imposes a serious problem for floating point computation, since computers have only finite precision.

The ill-conditioning comes from a combination of two sources, i.e. the discretization, and that the problem that we formulated has different scales for different variables. The first one is inevitable, because as the discretization resolution increases the ill-conditioning grows with it. However, we can scale the matrix to improve its conditioning.

In order to improve the conditioning of $A$, we use a diagonal scaling matrix $P$. If we left-multiply equations (3.21) and (3.22) by $P$ and change variable as $x = P^{-1}\dot{x}$, we get

$$
\dot{x} = PAP^{-1}\dot{x} + PBv
$$

$$
r = P^{-1}C.
$$
Thus, the scaling of $A$ is described as

$$\tilde{A} = PAP^{-1}. \quad (4.23)$$

It is easy to show that the scaling in (4.23) does not affect the eigenvalues of $A$, but it changes the singular values. Note that $\|A\|_2$ is equal to the largest singular value of $A$. Therefore, scaling can improve the conditioning of the matrix by decreasing $\kappa(A)$ in (4.20).

By studying the elements in the matrix structure of $A$ whose structure is shown in Figure 3-5 and (3.23), we discover that those elements in the rows and columns that correspond to the second state variable $x_2$, i.e. the velocity variable $\dot{z}$, are many orders of magnitude larger than those of the position state variables $x_1$ and the pressure state variable $x_3$. This suggests that we scale the velocity components in the state space vector (see 3.8) and in correspondence also those ODE’s describing the velocities in the state space equation (3.21) by a scaling constant. Expressing this scaling process in terms of the scaling matrix $P$, we get

$$P = \begin{bmatrix}
I_{N \times N} & \alpha & \cdots & \alpha \\
\vdots & & & \\
\alpha & I_{N N \times N M}
\end{bmatrix} \quad (4.24)$$

where $\alpha$ is the scaling factor.

By numerical experiments, we found the optimal scaling factor is about $1.82 \times 10^5$. Then, we discovered that this number is actually the bandwidth frequency of the frequency response of the linear system, which we will show as the result in the next chapter.

This is not surprising at all, once we understand that the scales of the non-dimensional dependent variables are much different. In order to see this, we need to perform a dimensional analysis of the three dependent variables. We normalize them in the same way as in (2.5). Thus, the non-dimensional variables are compared as

$$x_1 = u \sim 1 \quad x_2 = \frac{\delta u}{\delta t} \sim \frac{1}{\omega} \quad x_3 = p \sim 1 \quad (4.25)$$

where $\omega$ is a characteristic frequency or the inverse of a time constant. If the system has a resonant frequency of a few mega-hertz, then the velocity term will be smaller.
than the other two by an order of $10^6$, in turn the coefficients in front of it will be much larger than the rest.

Handpicking the optimal frequency scaling factor is tedious. In order to come up with the right frequency scaling factor automatically, a dimensional analysis on the system of dynamic equations (3.4-3.6) is necessary. For the non-dimensionalization, we need to find characteristic constants for all variables that appear in the equations. It is reasonable to use the width of the beam $w$ as the length scale, and the same scaling factors for the dependent variables as in (2.5). A characteristic time constant is not apparent to us unless we solve the coupled system somehow. However, we proceed with $t$ being the only dimensional parameter who has unit $[s]$. In equations (3.4-3.6), if we let

$$X = \frac{x}{w}, \quad Y = \frac{y}{w}$$  \hspace{1cm} (4.26)

then we get a quasi-non-dimensional expression of equations (3.4-3.6),

$$\dot{x}_1 = x_2$$  \hspace{1cm} (4.27)

$$\dot{x}_2 = -\left(\frac{EI}{\rho w^4}\right) \frac{\partial^4 x_1}{\partial X^4} + \left(\frac{S}{\rho w^2}\right) \frac{\partial^2 x_1}{\partial X^2} + \frac{F_{elec}}{\rho} + \frac{p_0 w}{\rho g} \int_0^1 x_3 dY$$  \hspace{1cm} (4.28)

$$\dot{x}_3 = -x_2 + \tilde{\sigma}^{-1}\nabla^2 x_3.$$  \hspace{1cm} (4.29)

where $\tilde{\sigma} = \frac{12\mu w^2}{(1+6K_0)G^2 p_0}$. Note that the left hand side of these two equations have units $[1/s^2]$ and $[1/s]$, respectively. This means that the coefficients in front of those non-dimensional terms on the right hand side contain information of characteristic frequencies for different interactions that are involved. For example, the first term in the Euler beam equation (2.1) models the pure elastic bending effect, and its characteristic frequency is $\omega_{bending} = \sqrt{\frac{EI}{\rho w^4}} \approx 3.2 \times 10^6$ which corresponds to the natural frequency of an undamped fix-fixed beam structure. Although these characteristic frequencies are not the optimal in the sense of improving the conditioning, they are readily computed. Thus, we can use one of them as the frequency scaling factor to scale the matrix $A$.

Another coefficient $\tilde{\sigma}$ appearing in (4.29) is worth of more attention. Multiply it by a characteristic frequency $\omega$ we get the squeeze number we mentioned earlier.
during the derivation of the governing equation in Chapter 2,

$$\sigma = \bar{\sigma} \omega = \frac{12\mu w^2 \omega}{(1 + 6K) g^2 p_a} \sim 0.1.$$  \hspace{1cm} (4.30)

The squeeze number for this squeeze damping system is fairly small which implies that low frequency motions dominants. Therefore, it verifies the assumption of low frequency we have made earlier in this chapter.
Results

In this chapter, we present the results of comparing the reduced-order models generated using the Arnoldi method with the non-linear solution provided by Hung [5]. Hung verified this non-linear solution with experimental data. For comparison purpose, we used the same discretization resolution as Hung [5], which is a $40 \times 20$ mesh of the top plate. As we will show in the following sections, a reduced-order model of very low order is able to describe the original large linear system accurately, which implies a great speed-up in computation.

5.1 Time Response

Results in this section demonstrate that the linear model is valid for small input voltage. The results also show that the macromodel generated automatically by the Arnoldi method is efficient and accurate in solving the large linear model.

In the non-linear simulation results presented by Hung et al, they showed that at a critical voltage $V = 9\text{volts}$ which is called the pull-in voltage, the top plate is entirely pulled down. In other words, at the pull-in voltage, the top plate collapses onto the substrate at the bottom (see Figure 2-1). This is of course a very non-linear behavior. We use this voltage as a reference to locate the linear operating region of the parallel plates structure.

With the mesh size of $40 \times 20$, the order of the linear system is 880. For the time response comparison, we compare the time history of the displacement of the center point of the beam (which is $r(t)$ in Figure 2-1 and the output function in state space representation (3.22)) in three models, namely, the non-linear model, the large linear
Figure 5-1: A comparison of the time responses of the non-linear model, linear model and 4th order reduced model at small voltage input $V = 0.1\,\text{volts}$. The center point displacement is plotted against time.
Figure 5-2: At $V = 2\text{volts}$, the linear model starts to deviate from the non-linear model. The reduced-order model still follows the linear model.

system and a 4th order macromodel, at different input voltages. The result in Figure 5-1 shows that at a small input voltage $V = 0.1\text{volts}$ the three curves representing solutions of each of the three models overlap with one another. It means that with such a small input voltage that the original system in (2.1)-(2.2) behaves completely linearly. Therefore, the linear model proposed in (2.6) and (2.7) is accurate. It also shows that the 4th reduced-order model, which is very small comparing to the original 880th order linear model, captures the linear model accurately. The reasons are that the Arnoldi method approximates those slow poles or low frequencies near $s = 0$ (see Figure 5-5), and the mechanical system mainly operates at low frequencies. Therefore, the Arnoldi based macromodel shows great accuracy in the linear behavior analysis of the coupled MEMS system of interest.

In Figure 5-2, we see that the linear model starts to deviate from the non-linear solution. And the macromodel of order 4th still follows the linear model exactly. This shows that when the input voltage exceeds $2\text{volts}$, the structure migrates into non-linear region and the linear model proposed starts to fail. Figure 5-3 demonstrates
Figure 5-3: A comparison of the time responses of the non-linear model, linear model and 4th order reduced model at small voltage input $V = 0.1\,\text{volts}$. The center point deflection is plotted against time.
that at the pull-in voltage, the time response of the structure is extremely non-linear. The linear model and the 4th order macromodel is accurate to about 15% of the total gap distance, or 15% the maximum deformation of the beam at the pull-in voltage.

## 5.2 Frequency Response

For the linear model and different order of macro-models generated using the Arnoldi method, we compare their response in frequency domain for a mesh size of 10 × 5, or a 70th order linear system. So that important characteristics of the linear system and its approximates can be revealed.

Figure 5-4 shows the comparison of the frequency responses of the large linear system and its two macro-model approximations. The two macro-models are of the order of 2 and 10, respectively. We discover from Figure 5-4 that the original linear system is a well damped system. The original linear system has a bandwidth frequency of 1.8 × 10^8 which was used as the optimal scaling factor in Chapter 4.

This figure also shows that the 2nd order macro-model matches the linear model in a low frequency range up to 10^6 Hz including the bandwidth frequency. And the 10th order model is able to follow all the oscillatory behavior both in the gain plot and the phase plot. This result verifies that the lumped-element macro-models proposed by Yang et al [4] are accurate enough for most purposes.

Figure 5-5 demonstrates that indeed the Arnoldi algorithm captures slow poles accurately. Other poles of the original model extend far beyond the leftmost pole shown in Figure 5-5.
Figure 5-4: A comparison of the frequency responses of the full linear system, 2nd order reduced model and 10th order reduced model.
Figure 5-5: A first 10 poles of the 20th order of reduced-order model captures those of the original 70th order linear system. The $q = 2$ macro-model captures the first two poles on the real axis.
Conclusion

In this thesis, we have proposed a new macromodeling approach which can be used for automatic macromodel extractions for MEMS devices that operates with small input signals. This approach utilizes the better conditioned Arnoldi method as the model order reduction algorithm.

The approach is demonstrated on a case study of a typical MEMS structure, a set of parallel plates with squeeze-film damping. In order to perform the macromodeling, we first derived a linear model from the nonlinear governing equations. Then, we applied finite difference schemes to discretize the linear model into the vector-matrix representation. From this discrete representation of the system, we transformed the dynamics equations into state space representation. This representation allowed us to apply the Arnoldi method to get the reduced order model.

The results showed that very low order models can accurately describe both the time and frequency responses of the original linear system, which is usually of order of thousands or more. Thus, these macromodels can be used as substitutes for the original large system for fast computation, especially where repetitive simulations are necessary.
Bibliography


