17
Reduced-Order Modeling

In this chapter, we introduce several reduced-order modeling techniques for analyzing microsystems following the discussion of Section 1.7. Specifically, techniques such as generalized Kirchhoffian networks, black box models, and Galerkin methods are described in detail. In generalized Kirchhoffian networks, a complex microsystem is decomposed into lumped elements that interact with each other as constituent parts of a Kirchhoffian network. Techniques such as equivalent circuit representations and description-language-based approaches are described under generalized Kirchhoffian networks. In black box models, detailed results from simulations are used to construct simplified and more abstract models. Methods such as nonlinear static models and linear and nonlinear dynamic models are described under the framework of black box models. Finally, Galerkin methods, where the basic idea is to create a set of coupled ordinary differential equations, are described. Both linear and nonlinear Galerkin methods are discussed. The advantages and limitations of the various techniques are highlighted.

17.1 Classification

Several techniques have been developed for reduced-order modeling or macromodeling of microsystems. Each technique has its own advantages and disadvantages, and the selection of a technique for a particular problem depends on a number of parameters such as the desired accuracy and nonlinearity. Many of the macromodels are created directly from physical-level simulations and often require human input at some stage of the process; i.e.,

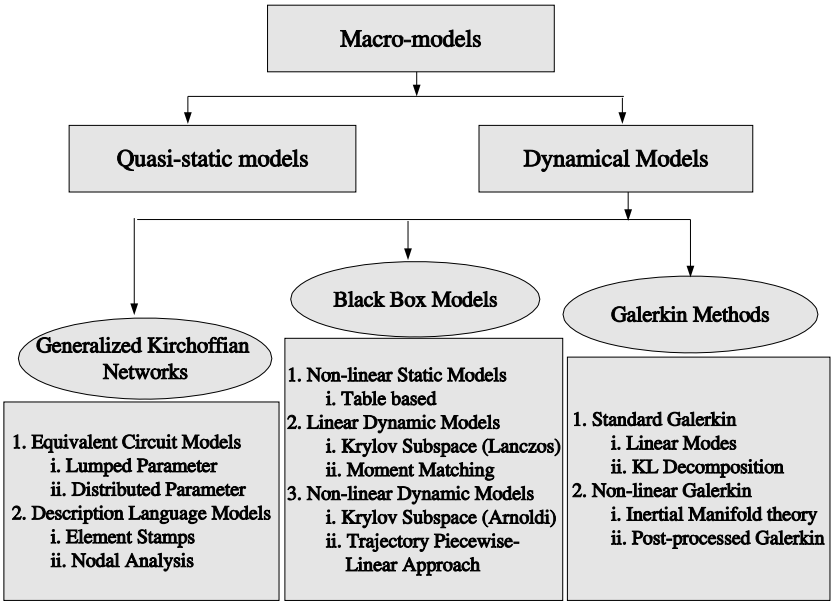


FIGURE 17.1. Classification of macro-models used in microsystem design.

there exists no systematic procedure to extract them automatically from the physical simulations. To identify macromodel extraction steps that can be automated in these cases is an important research topic in the field of microsystem simulation. In this section, we introduce the different macromodels and classify them into several broad categories. Figure 17.1 shows the classification of the various types of reduced-order models.

17.1.1 Quasi-Static Reduced-Order Modeling

Quasi-static macromodels are particularly useful for conservative systems with no dissipative terms. The distinction between energy domains in which the energy is strictly conserved (such as ideal elasticity, electromagnetic fields in linear lossless media, and inviscid flows) and those that have intrinsic dissipation mechanisms (fluidic viscosity, friction, heat flow, viscoelasticity, and internal loss mechanisms such as domain–wall motion that can lead to hysteresis (Senturia, 1998a)) is important, since dynamical behavior in a conservative domain can be derived from quasi-static behavior. All forces can be expressed as gradients of suitable potential-energy functions. If only conservative mechanisms are involved, one can use quasi-static sim-

ulations together with the mass distribution to fully characterize the dynamical behavior. Quasi-static macromodels are appropriate for cases in which a steady-state behavior is a reasonable assumption. In many cases, such as the squeeze film damping in a moving MEMS structure (see Chapter 18), such an assumption may be incorrect, in which case a dynamical macromodel is needed.

This procedure is fairly accurate only for conservative energy domains. For a more accurate analysis, the inertia and damping terms must be considered. Forces are expressed as appropriate gradients of suitably constructed potential energy or coenergy functions, and these functions are calculated quasi-statically. If one has knowledge of mass distribution, one can assess accelerations and kinetic energy in response to these forces and hence can construct complete dynamic models of the device using only quasi-static simulations in the potential energy domain.

The steps followed in the quasi-static reduced-order modeling are as follows (Senturia et al., 1997):

1. Select an idealized structure that is close to the desired model.
2. Model the idealized problem analytically, either by solving the governing differential equation, or by approximating the solution with Rayleigh–Ritz energy minimization methods.
3. Identify a set of nondimensionalized numerical constants that can be varied within the analytical form of the solution.
4. Perform meshed numerical simulations of the desired structure over the design space of interest, and adjust the nondimensionalized numerical quantities in the macromodel for agreement with the numerical simulations.

The method has some advantages: (i) simple to use and easy to implement; (ii) reasonably accurate for conservative energy systems when mass distribution is known; (iii) can be used to determine material constants; (iv) even if analytical functions exist for nonlinear behavior, in most cases nonlinearities can be taken care of by a simple fit function. However, the major disadvantage of this method is that it cannot be used in a nonconservative energy system, i.e., when dissipation is involved.

17.1.2 Dynamical Reduced-Order Modeling

Explicit dynamical formulation of microsystems can be very time-consuming, and computationally expensive to insert in a system-level simulator. As a result, it is difficult for the designer to use it in an iterative design cycle or to probe sensitivities to variations in the geometry and material constants by repeated simulations. This demands the development of dynamical reduced-order models for projecting the results of the fully meshed

analysis onto physically meaningful reduced variables, containing algebraic dependencies on structural dimensions and material constants. Dynamical reduced-order modeling is much more challenging than the quasi-static reduced-order modeling, since the design space involves large motions and nonlinear forces. The various reduced-order modeling methods that fall under dynamical methods are shown in Figure 17.1. These methods can be broadly classified into three categories: (1) generalized Kirchhoffian networks, (2) black box models, and (3) Galerkin methods. In the following sections, we will look into the different methods that fall under these three categories in detail.

17.2 Generalized Kirchhoffian Networks

In this method, a complex microsystem is decomposed into components (or lumped elements) that interact with each other as constituent parts of a Kirchhoffian network (Voigt and Wachutka, 1997). Compact models with very few degrees of freedom are formulated for each of the components. All the system components are given a mathematical description in terms of conjugate thermodynamic state variables and the pertinent currents (fluxes or through quantities) and the driving forces (affinities or across quantities) such as mass flow and pressure gradient, electrical current, and voltage drop. A system component is called a "block" and is characterized by the number and nature of its terminals, which allow for the exchange of flux quantities across subsystem boundaries. The components can be decomposed further by either (a) interconnection of basic components (structural modeling) or (b) description by a set of algebraic equations (behavioral modeling). The models of reusable components are stored in a block library.

Behavioral description of devices and subsystems when supported by hardware description languages (HDL) leads to the approach of macromodeling based on HDLs. Some examples of HDLs are HDL-A[1] (Mentor Graphics, 1995), MAST[2] (Mantooth and Vlach, 1992), and SpectreHDL[3] (Cadence, Ltd., 1997). The features of these HDLs are (i) Multidomain description, (ii) clear distinction between interface and algorithmic kernel of the model, (iii) interface to embedded C programs, and (iv) mechanisms for handling nonlinear ODEs to be solved by the internal algorithm.

[1]Hardware description language for analog and mixed signal applications: Mentor Graphics.

[2]Analog hardware description language used in the SABER simulator from Synopsys Inc.

[3]Analog Hardware Description Language: Cadence Ltd.

17.2.1 Equivalent Circuit Representation

In the equivalent circuits approach, the microsystem comprising the mechanical, electrical, and fluidic components is represented by their electrical equivalents (Tilmans, 1996). The approach is based on the mathematical analogy between electrical and mechanical systems; specifically, the formal similarities between the integrodifferential equations governing the behavior of electrical and mechanical systems is the basis of the analogy. Newton's second law of motion relates the force F acting on a body of mass m by the relation

$$F = m\frac{\partial u}{\partial t} = m\frac{\partial^2 x}{\partial t^2},$$

where u and x are the velocity and displacement of the mass. This is analogous to the constitutive equation of an electrical inductor

$$V = L\frac{\partial i}{\partial t} = L\frac{\partial^2 q}{\partial t^2},$$

where V is voltage, i is current, q is charge, and L is the inductance in the electrical circuit. Here F plays the same role as the voltage V, the velocity u as the current i, and the displacement u as the charge q. The mass m in mechanical systems represents the inductance L in an electrical circuit. However, while force is a "through variable" in mechanical systems, voltage (its representative in the electrical system) is an "across variable" in electrical systems. In general, it can be seen that a through variable in the mechanical system is represented as an across variable in an electrical system and vice versa. Hence, in the equivalent circuit representation, series connections in mechanical systems are represented by parallel connections in electrical systems and vice versa.

Once all the mechanical systems are converted to electrical counterparts, a single representation of a system operating in more than one energy domain is finally obtained. Kirchhoffian conservation laws are then applied to solve the system. Commercial packages like SPICE[4] (Quarles et al., 1987) can be used for such purposes. This method gives a better understanding and visualization of the system and facilitates further analysis of the system in order to investigate the effects of the connecting subsystems or modifications to the system.

Lumped Parameter System

The lumped parameter approach is a reduced-order modeling method based on the equivalent circuit representation of microsystems. The basic idea in this method is to concentrate or lump the physical properties of the system, such as mass, stiffness, capacitance, and inductance, into single physical elements (Tilmans, 1996). The elements representing the mass are assumed

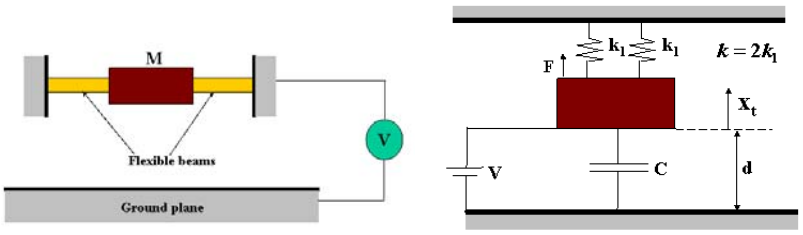
[4]Integrated Circuit Simulator: UC Berkeley

TABLE 17.1. Direct electromechanical analogies for lumped translational systems.

Mechanical Quantity	Electrical Quantity
Force: F	Voltage: V
Velocity: u	Current: i
Displacement: x	Charge: q
Mass: m	Inductance: L
Compliance: $1/k$	Capacitance: C
Viscous resistance: c	Resistance: R

to be perfectly rigid, and conversely, elastic elements have no mass. Lumped parameter modeling is typically valid as long as the wavelength of the signal is greater than all the dimensions of the device or the system under consideration. The device is finally represented as a set of lumped electrical network elements using the electrical-mechanical analogies. Once the circuit representation is constructed, commercially available circuit simulators such as SPICE can be used. Alternatively, dynamical state equations can be obtained from the network and simulated using standard math packages like MATLAB. There are two major issues in creating lumped-element macromodels: the first issue is to partition the continuum device into a network of lumped elements, and the second issue is to determine the parameter values for each element.

The partitioning problem can be particularly troublesome, because unlike purely electric circuits, general mechanical structures do not offer a clean mapping between geometry and the corresponding network topology. For example, when considering the electrostatic pull-in of the beam, the moving beam simultaneously serves as the moving boundary of a capacitor used for actuation and sets a moving boundary condition at the bounding surface of a gas film. The lumped parameter values for the physical elements are typically determined from a combination of analysis, numerical simulation, and constitutive properties obtained from test structures. An alternative approach to determine the lumped parameters is to use energy methods. In energy methods, a reasonable shape function with one or more undetermined parameters is assumed, the total stored energy is calculated with that shape function, and the stored energy is then minimized with respect to the parameters. The accuracy of energy methods depends on the quality of the shape function employed and on how well the stored energy



(a) (b)

FIGURE 17.2. (a) A parallel plate electrostatic actuator. (b) Schematic representation of the parallel plate electrostatic actuator.

calculation is implemented. For a first-order analysis of device behavior, accounting for the dependence of device behavior on the geometry and material properties, energy methods are unmatched in terms of simplicity and speed. A generalized procedure for lumped parameter modeling is as follows:

1. The device is decomposed into a combination of rigid bodies, ideal springs, and ideal variable capacitors. The damping is considered external to the transducer.

2. The governing equations are linearized about an equilibrium signal.

3. The characteristic equations, which relate the effort variables as a function of state variables, and the transfer matrix, which relates the effort-flow variables at the electrical port directly to those at the mechanical port, are derived.

4. The transfer matrix is used to obtain the equivalent circuit representation. Typically, the equivalent circuit representation may not be unique. So a practical situation is chosen.

We now illustrate the equivalent circuit representation of a parallel plate electrostatic actuator (in the absence of air-damping) using the above procedure. Figure 17.2(a) shows the parallel plate electrostatic actuator consisting of a rigid mass suspended by two flexible beams. A potential difference is applied between the ground plane and the mass, giving rise to attractive electrostatic forces. The mass moves down, and the beams bend due to these forces. The schematic representation of the device is given in Figure 17.2(b). Since the mass is rigid, the suspension structure can be modeled as a mass–spring system. The two identical beams represent the two springs. The spring constant ($k = 2k_1$) can be derived from the beam flexure formula. We use the energy method as described earlier to generate

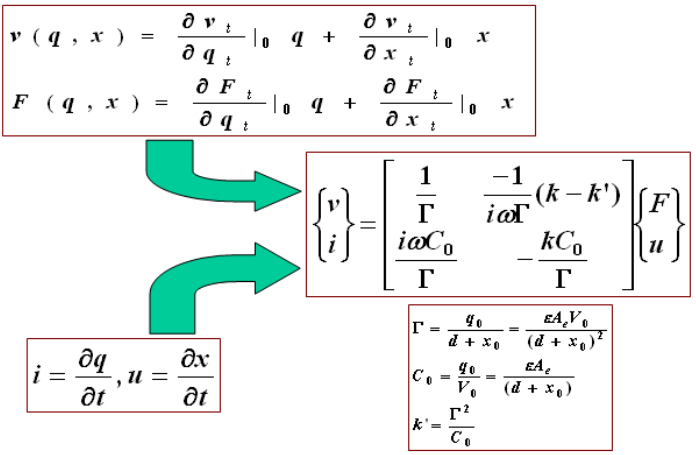


FIGURE 17.3. Transfer matrix computation for the parallel plate actuator using lumped parameters.

the lumped parameters. The total energy W of the system, consisting of electrical (W_e) and mechanical (W_m) energies, is given by

$$W = W_e(q_t, x_t) + W_m(q_t, x_t) = \frac{q_t^2}{2C(x_t)} + \frac{1}{2}k(x_t - x_r)^2, \tag{17.1}$$

where q_t and x_t are the charge and displacement at the time instant t, and x_r is the equilibrium position of the mass; C is the capacitance of the system at the time instan, which is a function of x_t and is given as $C(x_t) = \epsilon_0 A/(d+x_t)$; ϵ_0 is the permittivity of vacuum and A denotes area. Taking the total differential of the energy represented by equation (17.1), we obtain

$$\delta W = \frac{\partial W}{\partial q_t}\delta q_t + \frac{\partial W}{\partial x_t}\delta x_t = v_t dq_t + F_t dx_t, \tag{17.2}$$

where v_t is the voltage between the plates and F_t is the mechanical force acting on the movable plate. Using equations (17.1), (17.2) and the expression for capacitance, we obtain

$$v_t(q_t, x_t) = \frac{\partial W}{\partial q_t} = \frac{q_t(d + x_t)}{\epsilon_0 A} \quad \text{and} \quad F_t(q_t, x_t) = \frac{\partial W}{\partial x_t} = \frac{q_t^2}{2\epsilon_0 A} + k(x_t - x_r).$$

Since the equations are nonlinear, they are linearized using the Taylor's series expansion around some bias point (x_0, q_0). The constitutive equation, describing the linear relations between the incremental or small signal effort variables and the state variables, for voltage at the bias point (x_0, q_0) is

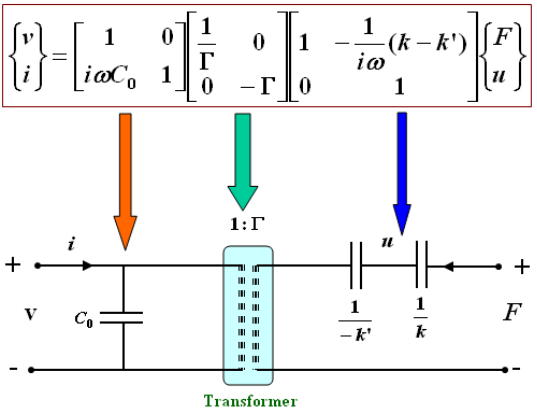


FIGURE 17.4. Decomposition of the transfer matrix into elemental matrices for circuit representation.

given by

$$v(q,x) = \left.\frac{\partial v_t}{\partial q_t}\right|_0 q + \left.\frac{\partial v_t}{\partial x_t}\right|_0 x = \frac{(d+x_0)}{\epsilon_0 A} q + \frac{q_0}{\epsilon_0 A} x = \frac{q}{C_0} + \frac{v_0}{(d+x_0)} x. \quad (17.3)$$

Using the constitutive equation for force and the expression for F_t, we obtain the final expression for the force, i.e.,

$$F(q,x) = \left.\frac{\partial F_t}{\partial q_t}\right|_0 q + \left.\frac{\partial F_t}{\partial x_t}\right|_0 x = \frac{q_0}{\epsilon_0 A} q + kx = \frac{v_0}{(d+x_0)} q + kx. \quad (17.4)$$

The constitutive equations and the final expressions for v and F as given by equations (17.3) and (17.4), respectively, can be used to construct the transfer matrix as shown in Figure 17.3. The transfer matrix can be decomposed into elemental matrices in several ways, giving rise to many feasible circuit representations of the device. Figure 17.4 shows one decomposition of the transfer matrix and the corresponding circuit representation. There are several other circuit representations possible for the same device (see (Tilmans, 1996), for details) including some with pure capacitive circuits. Typically, the designer chooses the most appropriate circuit representation based on the application.

The advantages of the lumped parameter method are as follows: (i) It is easy to use and can be easily incorporated into system simulators. (ii) Equivalent lumped resistors treated external to the system can be used to model dissipation. (iii) The equivalent circuit representation can be used to analyze complex structures and coupled subsystems with several electrical and mechanical ports. The disadvantages of the method are these: (i) Unlike pure electric circuits, mechanical structures do not offer a clean

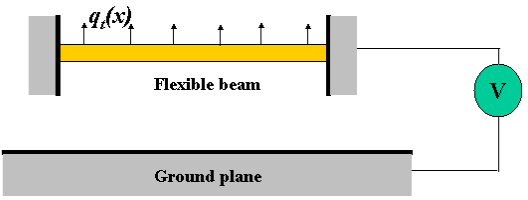


FIGURE 17.5. Continuously distributed system with infinite degrees of freedom.

mapping between the geometry and the corresponding network analogy. (ii) No CAD tools are currently available that can automatically construct an energetically correct lumped-element topology directly for an arbitrary device geometry. (iii) Large-signal and nonlinear analysis is cumbersome, difficult, and error-prone. (iv) In most cases the conservative and dissipative energy domains are to be modeled separately.

Distributed Parameter System

In a distributed parameter system (see (Tilmans, 1997)), the mass, compliance, capacitance, etc., are not easily identifiable as lumped elements at individual points. These elements are, instead, continuously distributed throughout the system. Figure 17.5 shows another parallel plate actuator, but in this case, in contrast to the lumped parameter case, the mass is a continuously deformable beam with a uniformly distributed load on it. The electrical and mechanical domains are coupled either through the boundary of the flexible beam or throughout the entire system as in the case of transducers employing piezoelectric materials. In such cases, it is difficult to distinguish between the mechanical and electrical forces, and the lumped parameter system cannot be easily used to extract the circuit parameters. Instead, the distributed parameter approach needs to be employed. The fundamental difference between lumped parameter models and distributed parameter systems is that while the former method has a finite number of degrees of freedom, the latter has an infinite number of degrees of freedom. The lumped parameter models and the distributed parameter models are just two distinct mathematical models of the same physical system and the distributed parameter approach can be considered as a more general approach compared to the lumped parameter approach (Tilmans, 1997).

The procedure for distributed parameter modeling is given by:

1. A quasi-one-dimensional modeling of the system is first performed, where only the displacements associated with the dominant modes are considered, while the ones in the other directions are neglected. This reduces the dependence of the system behavior to a single coordinate.

2. The characteristic equations are derived using modal analysis, which is described in detail later in this section.

3. Modal analysis techniques are used to find the solution to the governing equations in terms of normalized mode shapes and generalized coordinates by the use of the mode-superposition principle.

4. A Galerkin-like approach is then used to generate an infinite set of uncoupled ODEs that represent the system by means of an infinite number of single-degree-of-freedom lumped-parameter systems.

5. The characteristic equations of the system that describe the linear relations between incremental variations of the port variables around a stable bias point are derived. The equations are then linearized around the bias point.

6. Using the possible characteristic equations and the numerous equivalent circuit representations, a circuit representation that represents the practical situation in the most appropriate way is selected.

The steps in the distributed parameter approach are similar to those in the lumped parameter approach, except that the continuous system is modeled using modal analysis in the distributed parameter approach to reduce the degrees of freedom. We consider the actuator shown in Figure 17.5 to illustrate the distributed parameter approach. The electrical and mechanical domains are coupled through the surface of the flexible beam, which can be thought of as an infinite number of localized individual electrostatic transducers. The electrical energy stored in each such element of infinitesimal area δA is given by

$$U\delta A = \frac{1}{2}\frac{(\sigma_t(x,y)\delta A)^2}{\epsilon_0 \delta A/(d+w_t(x))} = \frac{1}{2}\frac{\sigma_t(x,y)^2}{\epsilon_0/(d+w_t(x))}\delta A, \tag{17.5}$$

where $\sigma_t(x,y)$ denotes the surface electric charge density as a function of position. The total differential of equation (17.5) is given by

$$\delta U = \frac{\partial U}{\partial \sigma_t}\delta\sigma_t + \frac{\partial U}{\partial w_t}\delta w_t.$$

The voltage at time instant t, v_t (see the discussion leading to equation (17.2)), can be expressed as

$$v_t = \frac{\partial U}{\partial \sigma_t} = \frac{\sigma_t(x,y)}{\epsilon_0/(d+w_t(x))}.$$

Similarly, the mechanical pressure acting along the surface p_t can be expressed as

$$p_t = \frac{\partial U}{\partial w_t} = -\frac{\sigma_t(x,y)^2}{2\epsilon_0}.$$

Similar to the derivation of equations (17.3), (17.4), the expressions for $v \equiv \Delta v_t$ and $p \equiv \Delta p_t$ in terms of $w \equiv \Delta w_t$ and $\sigma \equiv \Delta \sigma_t$ are given by

$$v = \frac{d + w_0}{\epsilon_0}\sigma + \frac{v_0}{d + w_0}w \quad \text{and} \quad p = -\frac{v_0}{d + w_0}\sigma + 0 \cdot w.$$

The operating point is indicated by the subscript 0, and the coefficient "0" arises because the stiffness properties of the beam are considered to be external to the transducer (Woodson and Melcher, 1968). Rewriting the above equations, the expressions for σ and p are given by

$$\sigma = \frac{\epsilon_0}{d + w_0}v - \frac{\epsilon_0 v_0}{(d + w_0)^2}w \quad \text{and} \quad p = -\frac{\epsilon_0 v_0}{(d + w_0)^2}v + \frac{\epsilon_0 v_0^2}{(d + w_0)^3}w.$$

In the absence of elastic stiffness, the exterior mechanical pressure p is completely counterbalanced by the electrostatic pressure p_e, and hence

$$p_e = -p = \frac{\epsilon_0 v_0}{(d + w_0)^2}v - \frac{\epsilon_0 v_0^2}{(d + w_0)^3}w.$$

Next, the elastic properties of the beam are taken into account. Employing energy methods (Shames and Dym, 1985) and excluding dynamic terms, the differential equation of motion is

$$L\left[w(x)\right] = -q_e(x) + q(x), \tag{17.6}$$

where $q_e(x)$ and $q(x)$ are transverse forces per unit length of electrical and mechanical origin, and L is the differential operator given by

$$L = EI\frac{\partial^4}{\partial x^4} - N\frac{\partial^2}{\partial x^2}, \tag{17.7}$$

where E and I are the Young's modulus and the second moment of inertia of the beam, respectively, and N is the applied axial force. Rewriting equation (17.6), we have

$$q(x) = \frac{b\epsilon_0 v_0}{(d + w_0(x))^2}v + L_e\left[w(x)\right], \tag{17.8}$$

where

$$L_e \equiv L - \frac{b\epsilon_0 v_0^2}{(d + w_0(x))^3} = L - k_e(x).$$

The eigenvalues and the mode shapes (eigenvectors) of the governing equation (17.8) are now computed by solving a standard eigenvalue problem. Using the mode superposition principle, an infinite number of ordinary differential equations are formed (this topic is discussed in more detail in Section 17.4). Typically, a few mode shapes contain most of the mechanical energy, and these few modes can satisfactorily capture the mechanical deformation, thereby reducing the order of the problem.

In summary, we have a few ODEs to describe the parallel plate actuator, which can now be used to construct the equivalent circuit of the system in the same way as described earlier for the lumped parameter modeling. The advantages of the distributed parameter approach are the following: (i) It can be used to model continuous systems where most other methods fail. (ii) It can be incorporated into system simulators. (iii) Distributed parameter electrical devices can be coupled to the mechanical and the electrical terminal pairs as done in the lumped networks case. In a general case, the system has one electrical port characterized by the voltage v and the current i, and an infinite number of mechanical ports characterized by a generalized load and a generalized velocity. The disadvantages of the method are these: (i) It needs designer input, and test structures are required to verify whether the modeling results are correct. (ii) In most cases the conservative and dissipative energy domains have to be modeled separately. (iii) Since there is no unique representation possible, macromodel generation cannot be automated easily.

17.2.2 Description Languages

Even though the equivalent circuit approach is popular, there are several drawbacks to using an equivalent circuit representation, the most important one being that not all microdevices can be represented by equivalent circuits, and even if an equivalent circuit representation exists, its construction may not be trivial. Besides, the physical meaning of the problem gets complicated due to representation of nonelectrical quantities such as force and velocity in terms of electrical quantities such as current and voltage. As a result, it may not be easy to understand how well the model captures the physics of the device.

Several other methods have also been developed which are based directly on the algebraic-differential equations that describe the device behavior. Suitable languages are used to describe the equations, with hardware description language being one of them. Simulations using description languages are, however, slower. The speed can be increased by using built-in libraries (stamps) for some standard structures or devices. Nodal analysis is another method, where the differential equations are solved directly. These two techniques are summarized below.

Element Stamps

One way of building coupled systems of equations in the electrical and mechanical domain is through the use of element stamps. Element stamps are the building blocks of conventional circuit simulators. They are derived from lumped-constant models of individual microdevices, and built into the circuit simulator (Casinovi, 2002). The use of element stamps for the constitutive elements allows one to simulate a system in a faster and more ef-

ficient way compared to models written in hardware description languages. Lumped constant models are sufficiently accurate for many applications. Ordinary circuit simulators use lumped constant models for electronic devices. This approach enables the simulation of complex mixed technology systems starting from their constitutive elements. Like Kirchhoff's laws for electrical components, the equations governing the dynamics of constrained rigid bodies are additive with respect to the number of elements in the system. As a consequence, models of microdevices can be represented by stamps, which contain all the terms that contribute to the global system of equations.

The Modified Nodal Analysis (MNA) technique for circuit simulation is based on Kirchhoff's current law, which states that the sum of all outgoing currents at each node is equal to zero, i.e.,

$$\sum_k i_k = 0.$$

Kirchhoff's current law, when considered at each node, generates a set of algebraic differential equations, which can be transformed into a set of algebraic equations by applying a suitable numerical integration scheme. The nonlinear algebraic equations can be solved by numerical methods (e.g., a Newton's method), which require repeated solution of a set of linear equations of the form

$$\mathbf{A}\mathbf{v} = \mathbf{b}.$$

The observation that each element in the circuit contributes to the above equation leads to the element stamp concept. Figure 17.6 shows a simple resistor (of conductance G) connected between nodes t and o. The branch current, i, given by $i = G(v_t - v_0)$, appears with a positive sign in the current equation at node t and with a negative sign at node o. Hence, the resistor contributes the term $+G(v_t - v_0)$ to the tth equation and the term $-G(v_t - v_0)$ to the oth equation. An element stamp (which is the coefficient matrix $\mathbf{A}$ having rows corresponding to each node, and columns corresponding to each variable) for the resistor is shown in Figure 17.6. If v_t and v_o are the tth and the oth elements of the voltage vector, $\mathbf{v}$, respectively, the resistor's contribution to the coefficient matrix adds a quantity $+G$ to positions (t, t) and (o, o) and a $-G$ to the positions (t, o) and (o, t). The lumped-constant models for MEMS is based on the fact that all the MEMS structures are built from a common set of basic (or atomic) elements, such as beams, anchors, and plates. Though these are less accurate than distributed constant models, they reduce the computational effort by a great deal. For many applications the accuracy obtained from a lumped-constant model could be sufficient. Just as the conventional circuit simulation relies on the lumped-constant models of the electronic devices, microsystem simulation can be achieved by extending the concept of element stamps to microdevices and their constitutive elements. In this

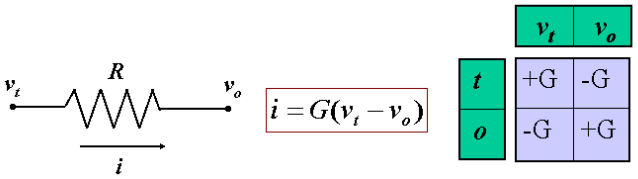


FIGURE 17.6. Element stamp for a resistor.

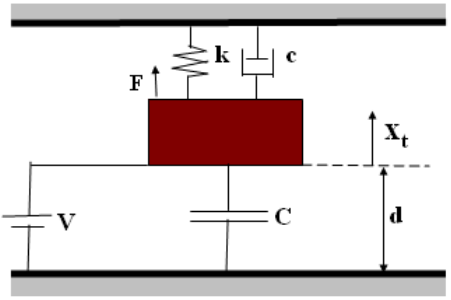


FIGURE 17.7. Schematic diagram of the parallel plate actuator with damping.

case, the Kirchhoff's circuit laws are replaced by the Newtonian equations of motion. The equations governing the dynamics of rigid bodies are given by

$$\sum_k F_k = 0 \quad \text{and} \quad \sum_k T_k = 0,$$

where F and T stand for forces and torques, respectively. The similarity between Kirchhoffian current laws and these equations is apparent. However, one difference is that since microsystems involve both mechanical and electrical domains, there are both mechanical and electrical variables and equations contained in element stamps for microsystems. We illustrate the development of an element stamp for the parallel plate actuator shown in Figure 17.2. External (air) damping is also considered and is represented by the external damper (c) as shown in the schematic diagram in Figure 17.7. The governing equations for the system are given by

$$m\frac{\partial^2 x}{\partial t^2} + c\frac{\partial x}{\partial t} + kx = F_e = -\frac{\epsilon_0 A}{2}\left[\frac{V^2}{(d-x)^2}\right] \quad \text{and} \quad i = \frac{\partial q}{\partial t} = \frac{\partial}{\partial t}\left[\frac{\epsilon_0 A V}{d+x}\right].$$

Trapezoidal integration and linearization of the governing equations gives

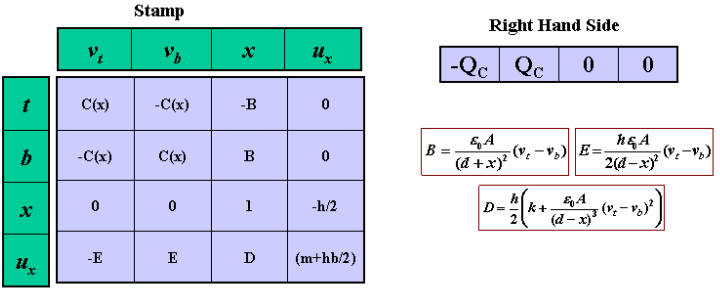


FIGURE 17.8. Element stamp for the parallel plate actuator.

rise to the following simple algebraic equation:

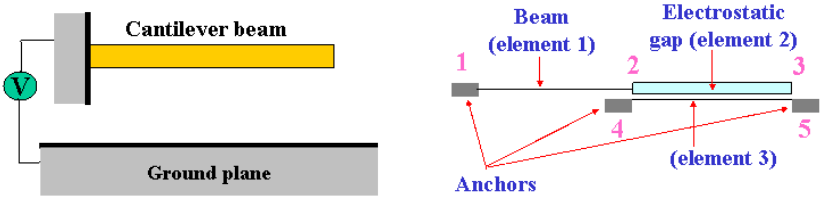
$$mu_x + \frac{h}{2}cu_x + k\frac{h}{2}x + \frac{h}{2}\left[\frac{\partial F_e}{\partial v_t}v_t + \frac{\partial F_e}{\partial v_b}v_b\right] = 0,$$

where $u_x = \partial x/\partial t$, h is the time step of integration, and v_t and v_b are the voltages on the beam and the ground element, respectively. Using the nodewise analysis as described above for the resistor, the element stamp for this MEM actuator is shown in Figure 17.8. The stamp has four rows, two for the electrical equations and two for the mechanical-dynamical equations. The four columns correspond to the four variables, namely, v_t, v_b, x, and u_x. This method takes much less time to simulate in a circuit simulator. It provides a compact and efficient way of adding the contribution of a particular element to the overall system. Since built-in models are used, this procedure can handle only those devices that can be described by the built-in models. This might not be a disadvantag, since all MEMS devices can be described in terms of a set of basic elements. The computational effort required is much less compared to general purpose simulators like SABER[5] (Mantooth and Vlach, 1992) or MATLAB that rely on user-provided HDL models.

Nodal Analysis

Nodal analysis has been widely used for formulating system equations in circuit analysis tools such as SPICE. The circuit is decomposed into N-terminal devices, and each device is modeled by ordinary differential equations (ODEs) with coefficients parameterized by device geometry and material properties (Zhou et al., 1998). The devices are linked together at their terminals or nodes, and the resulting coupled differential equations can be solved as a system of nonlinear ODEs. This approach is fast, reasonably accurate, flexible, and can be used for a higher-level simulation of

[5]Mixed mode circuit simulators from Synopsys Inc.



(a) (b)

FIGURE 17.9. (a) A cantilever-beam-based microdevice. (b) Nodal representation of the microdevice.

microsystems.

In nodal analysis, the microdevice is represented using atomic elements like anchors, gaps, and beams (Zhou et al., 1998). Figure 17.9 shows a microdevice and its nodal representation. The nodal representation contains three anchor elements, one beam element, and an electrostatic gap element. Each atomic element has a lumped behavioral model with geometric parameters that can be specified individually. This simplifies the evaluation of changes in size on the device performance in each design iteration. The system matrices formed are much smaller than those in finite element analysis, and the models are implemented in analog HDLs supporting mixed physical domain simulations. The total system is formulated by formulating each individual element first. For the beam element defined between nodes 1 and 2, we have

$$f_n^1 = f_n^1(q_1, q_2), \qquad n = 1, 2,$$

and for the gap element (nodes 2, 3, 4, 5) we have

$$f_n^2 = f_n^2(q_2, q_3, q_4, q_5), \qquad n = 2, 3, 4, 5,$$

where f_n, represents the internal forces (the forces in the x-direction, y-direction, and the moment) acting at node n, and q_n represents the node displacements (the displacements in the x- and y-directions and the angle of rotation). The superscript and the subscript denote the element number and the node number, respectively. Each node has three degrees of freedom in 2D: the displacements in the x- and y-directions and the rotation. The sum of all the internal forces (f_n) acting at a given node is equal to the external load P acting at the node, which in this case is the electrostatic

force. The assembled equations for each node are given by

$$\begin{aligned}
P_1 &= f_1^1(q_1, q_2),\\
P_2 &= f_2^1(q_1, q_2) + f_2^2(q_2, q_3, q_4, q_5),\\
P_3 &= f_3^2(q_2, q_3, q_4, q_5),\\
P_4 &= f_4^2(q_2, q_3, q_4, q_5),\\
P_5 &= f_5^2(q_2, q_3, q_4, q_5).
\end{aligned}$$

The displacements and the rotations associated with nodes 1, 4, and 5 are zero, so they are removed. The final system of equations is given by

$$\begin{aligned}
P_2 &= f_2^1(q_2) + f_2^2(q_2, q_3),\\
P_3 &= f_3^2(q_2, q_3).
\end{aligned}$$

This system of equations can be solved by standard numerical methods.

NODAS (Fedder and Jing, 1998) is a circuit-level behavioral simulation tool that uses the concept of nodal analysis. Design with NODAS starts from schematic entry, where microsystem elements (such as beams and fluidic channels) and circuit elements (such as transistors) can be wired together. A composite net list for the entire system is generated and sent to the circuit simulator. In the schematic generation phase, terminals of element instances are represented by groups of pins. Each pin has an associated discipline determining its physical nature. Since schematic assembly consumes a lot of effort and is prone to error, if pins were used for each degree of freedom, buses are used in digital circuit schematics for compactness of schematic representation. Similarly, for the same reason analog buses are used in NODAS; however, since existing limitations in analog HDLs (hardware description languages) allow only pins of the same discipline to be grouped as one bus, they result in three buses per terminal: translational, rotational, and electrical. This compact terminal representation reduces wiring effort and wiring errors. Splitters are the behavioral blocks used to convert scalar wires to bus wires. They also apply stimuli and monitor simulation results at the individual degree of freedom. Global acceleration and rotational rate pins are used and shared by all elements in combination with the hierarchical schematic for each model to reduce clutter in the schematic. These pins take care of the external dynamics influence. The "through" and "across" variables are chosen in accordance with Kirchhoff's laws. The across variables are chosen depending on the output required. Since Kirchhoff's network laws are applied in the chip's reference frame, coordinate transformation matrices are used to transform from one coordinate system to another. Some of the basic lumped models used are the linear beam model, nonlinear beam model, and nonlinear gap model. SUGAR (Zhou et al., 1998) uses a similar approach by modeling the MEMS structures in terms of three basic elements (i.e., beams, gaps, and anchors) and builds the ODE models for each kind. The system equations

are then formulated according to node connectivity information given as an input file and solved using nodal analysis.

The advantages of the nodal analysis method for microsystem design are as follows: (i) It can solve coupled nonlinear differential equations. (ii) It is fast, reasonably accurate, flexible, and can be used for higher-level simulation of microsystems. (iii) It can perform DC, steady-state, and transient analysis. The disadvantages of the nodal analysis technique are these: (i) The approach can still be expensive for complex systems. (ii) It cannot account for all the nonlinear behavior encountered in microsystems. More details on nodal analysis as well as more examples can be found in (Fedder and Jing, 1998; Vandemeer et al., 1998; Mukherjee and Fedder, 1998; Baidya and Mukherjee, 2002).

17.3 Black Box Models

Black box models stem from basic ideas in system and control theory. Black box models are based on measured input-output behavior, hence the name "black box models." Detailed results from simulations are used to construct simplified and more abstract models. The various models that fall under this category can be broadly classified into:

1. Nonlinear static models: These models use mathematical optimization, approximation, and interpolation methods for curve fitting and parameter adaptation. Table-based numerical reduced-order modeling falls in this category.

2. Linear dynamic models: These are usually formulated in the Laplace domain. The system is simulated in the time or frequency domain. Algorithms from control and system theory are used to calculate the transfer function. The response function is calculated using the convolution integral principles on the impulse function and the actual input function. If the system is complicated, random input functions may be needed to simulate the system instead of step or impulse functions. Krylov subspace techniques and moment matching methods fall under this category.

3. Nonlinear dynamic models: In these models, the modeling is done based on assumptions about the internal structure composed of basic functional blocks. A few control theory approaches are also available. Krylov subspace methods fall under this category.

17.3.1 Nonlinear Static Models

The method that falls in this category is table-based reduced-order modeling.

Table-Based Reduced-Order Modeling

Table-based reduced-order models use tables of numerical data to describe the relationship between variables (Wu and Carley, 2001). The table-based models are built directly from data obtained from measurements or device simulations without detailed knowledge of underlying physics. Hence, the difficulties and errors associated with extracting analytical models are eliminated and the process can be automated. For behavioral-level simulations, cubic spline interpolation is used to evaluate the functions. In addition to the nonlinear behavioral numerical model, a set of linear numerical models is constructed to assist in solving for operating points, to perform AC analysis, and to design closed-loop feedback systems. The procedure for table-based macromodeling can be described as:

1. The device is described by a set of ODEs, which are solved using standard numerical methods.

2. The functions that describe the relationships between the variables are represented in numerical tables, obtained from device simulations.

3. During simulations, the models are evaluated by interpolating the data in the tables.

The microdevice shown in Figure 17.5 is modeled using the steps outlined above. The governing equation for the dynamics of the device is given by

$$m\frac{\partial^2 x}{\partial t^2} + c\frac{\partial x}{\partial t} + f_s = f_e,$$

where m is the effective mass, c is the damping factor, f_s is the spring force, and f_e is the electrostatic actuation force. First, a full simulation of the device is performed to generate the table of data. The mechanical part (or the spring force) is computed using any standard finite element solver, and the electrostatic force is computed using the capacitance solver based on the boundary element method. The functions that represent the relationship between the variables, in this case f_s and f_e, are generated from tables of numerical data obtained from device simulations. During the simulations, the tabl-based models are evaluated by interpolating the data; typically, a cubic spline interpolation is used. As an example, suppose y depends on x, and for some value of x between x_i and x_{i+1}, y can be computed by

$$y = a_1 y_i + a_2 y_{i+1} + a_3 \frac{\partial^2 y_i}{\partial x^2} + a_4 \frac{\partial^2 y_{i+1}}{\partial x^2},$$

$$a_1 = \frac{x_{i+1} - x}{x_{i+1} - x_i}, \qquad a_2 = \frac{x - x_i}{x_{i+1} - x_i},$$

$$a_3 = \frac{1}{6}(a_1^3 - a_1)(x_{i+1} - x_i)^2, \qquad a_4 = \frac{1}{6}(a_2^3 - a_2)(x_{i+1} - x_i)^2.$$

Since second derivatives are required, typically, a second table comprising second derivatives is also constructed.

The merits of table-based reduced-order modeling are: (i) The procedure eliminates the difficulties and errors associated with analytical model extraction. (ii) The method can be easily automated once the order and state variables of the ODE are decided. The problems with table-based macromodeling are: (i) The method does not preserve the physical meaning of the system. (ii) It is not easy for the designer to propose modifications and expect the system to behave as desired. (iii) At least some expensive full-scale physical simulations are required to generate the table(s) of data.

17.3.2 Linear Dynamic Models

The reduced-order modeling methods that fall in this category are the Krylov subspace technique and the moment matching technique, which are discussed in this section.

Krylov Subspace Technique Based on the Lanczos Method

The governing equation for a continuous time-invariant multi-input multi-output (MIMO) system (e.g., a comb-drive microresonator) is of the form (Srinivasan et al., 2001; Bai, 2002)

$$\mathbf{C}\dot{\mathbf{x}}(t) + \mathbf{G}\mathbf{x}(t) = \mathbf{B}\mathbf{u}(t), \qquad \mathbf{y}(t) = \mathbf{L}^T\mathbf{x}(t), \tag{17.9}$$

where t is the time variable, $\mathbf{x}(t) \in \Re^N$ is a state vector, $\mathbf{u}(t) \in \Re^m$ is the input excitation vector, and $\mathbf{y}(t) \in \Re^p$ is the output vector. Here $\mathbf{C}, \mathbf{G} \in \Re^{N\times N}$ are system matrices, $\mathbf{B} \in \Re^{N\times m}$ and $\mathbf{L} \in \Re^{N\times p}$ are the input and output distribution arrays, respectively, N is the state space dimension, m and p are much smaller than N, and $m \geq p$.

A variety of analyses can be performed for the linear dynamical system given in equation (17.9). For example:

1. A static analysis to compute the equilibrium condition.

2. A steady-state analysis, also called the frequency response analysis, to determine the frequency responses of the system to external steady-state oscillatory (e.g., sinusoidal) excitation.

3. A transient analysis to compute the output behavior $\mathbf{y}(t)$ subject to time varying excitation $\mathbf{u}(\mathbf{t})$.

4. A sensitivity analysis to determine the proportional changes in the time response $\mathbf{y}(t)$ and/or steady-state response to a proportional change in system parameters.

Some of these analyses can be very expensive, especially if performed using equation (17.9). If a reduced-order system to equation (17.9) can be developed, the analysis can be performed quickly. The reduced-order system should have the following desirable attributes:

1. The reduced system should have a much smaller state-space dimension compared to the state-space dimension of the full-order system.

2. The error between the full-order and the reduced-order models should be as small as possible.

3. The reduced-order model should preserve the essential properties of the full-order system.

Therefore, the reduced-order linear system should be of the form

$$\mathbf{C}_n\dot{\mathbf{z}}(t) + \mathbf{G}_n\mathbf{z}(t) = \mathbf{B}_n\mathbf{u}(t), \qquad \mathbf{y}(t) = \mathbf{L}_n^T\mathbf{z}(t), \tag{17.10}$$

where $\mathbf{z}(t) \in \Re^n$ is a state vector, $\mathbf{u}(t) \in \Re^m$ is the input excitation vector, and $\mathbf{y}(t) \in \Re^p$ is the output vector. Here $\mathbf{C}_n, \mathbf{G}_n \in \Re^{n\times n}$ are system matrices, $\mathbf{B}_n \in \Re^{n\times m}$ and $\mathbf{L}_n \in \Re^{n\times p}$ are the input and output distribution arrays, respectively; n is the state space dimension, which should be much smaller than N. Assuming a single-input single-output (SISO) system for simplicity, $p = m = 1$. In this case, we use $\mathbf{b}$ and $\mathbf{l}$ for input and output vectors, respectively. The MIMO system can be dealt with in a similar manner.

The Krylov subspace technique (Srinivasan et al., 2001) reduces the original system (equation (17.9)) to the reduced system (equation (17.10)). Before we discuss the reduction method, it is important to understand the concept of Krylov subspaces. A Krylov subspace is a subspace spanned by a sequence of vectors generated by a given matrix and a vector as follows. Given a matrix $\mathbf{A}$ and a starting vector $\mathbf{r}$, the nth Krylov subspace $K_n(\mathbf{A}, \mathbf{r})$ is spanned by a sequence of n column vectors:

$$K_n(\mathbf{A}, \mathbf{r}) = \text{span}[\mathbf{r}, \mathbf{A}\mathbf{r}, \mathbf{A}^2\mathbf{r}, \mathbf{A}^3\mathbf{r}, \ldots, \mathbf{A}^{n-1}\mathbf{r}].$$

This is called the right Krylov subspace. When $\mathbf{A}$ is asymmetric, there exists a left Krylov subspace generated by $\mathbf{A}^T$ and a starting vector $\mathbf{l}$ defined by

$$K_n(\mathbf{A}^T, \mathbf{l}) = \text{span}[\mathbf{l}, \mathbf{A}^T\mathbf{l}, (\mathbf{A}^T)^2\mathbf{l}, \ldots, (\mathbf{A}^T)^{n-1}\mathbf{l}].$$

Next, we need to define a set of basis functions such that they span the desired Krylov subspaces. Let $\mathbf{V} = [v_1, v_2, v_3, \ldots, v_n]$ and $\mathbf{W} = [w_1, w_2, w_3, \ldots, w_n]$ be basis vectors such that

$$K_n(\mathbf{A}, \mathbf{r}) = \text{span}[v_1, v_2, v_3, \ldots, v_n], \qquad K_n(\mathbf{A}^T, \mathbf{l}) = \text{span}[w_1, w_2, w_3, \ldots, w_n].$$

The Lanczos process is an elegant way of generating such basis vectors. The Lanczos vectors $\mathbf{V}$ and $\mathbf{W}$ are constructed to be biorthogonal, i.e.,

$$w_j^T v_k = 0 \qquad \forall \quad j \neq k.$$

The Lanczos algorithm to generate $\mathbf{V}$ and $\mathbf{W}$ from $\mathbf{A}$, $\mathbf{r}$, and $\mathbf{l}$ can be found in (Freund, 1999).

The reduced system, using Krylov subspaces, can be generated using the following steps (see also Figure 17.10):

1. Taking s_0 as the expansion point of equation (17.9), $\mathbf{A}$ and $\mathbf{r}$ are defined as

$$\mathbf{A} = -(\mathbf{G} + s_0\mathbf{C})^{-1}\mathbf{C} \quad \text{and} \quad \mathbf{r} = (\mathbf{G} + s_0\mathbf{C})^{-1}\mathbf{b}.$$

2. Using the Lanczos method, $\mathbf{V}$ and $\mathbf{W}$ are computed as

$$\text{span}[\mathbf{V}] = K_n(\mathbf{A}, \mathbf{r}), \qquad \text{span}[\mathbf{W}] = K_n(\mathbf{A}^T, \mathbf{l}).$$

3. The reduced matrices $\mathbf{C_n}, \mathbf{G_n}, \mathbf{b_n}, \mathbf{l_n}$ are computed as follows: For double-sided projection

$$\mathbf{C}_n = \mathbf{V}^T\mathbf{C}\mathbf{W}, \quad \mathbf{G}_n = \mathbf{V}^T\mathbf{G}\mathbf{W}, \quad \mathbf{b}_n = \mathbf{W}^T\mathbf{b}, \quad \mathbf{l}_n = \mathbf{V}^T\mathbf{l},$$

and for single-sided projection,

$$\mathbf{C}_n = \mathbf{V}^T\mathbf{C}\mathbf{V}, \quad \mathbf{G}_n = \mathbf{V}^T\mathbf{G}\mathbf{V}, \quad \mathbf{b}_n = \mathbf{V}^T\mathbf{b}, \quad \mathbf{l}_n = \mathbf{V}^T\mathbf{l}.$$

The double-sided projection formula does not always guarantee a stable reduced-order model except for certain trivial cases (e.g., RC networks), whereas single-sided projection onto $\mathbf{V}$ guarantees an unconditionally stable reduced-order system. However, double-sided projection generally gives more accurate results than single-sided projection. The order of the subspace is chosen according to the frequency range where matching is required. For matching of q resonant peaks, n has to be at least $2q$.

Often, second-order systems are encountered in microsystems (Bai, 2002; Ramaswamy and White, 2001), of the form

$$\mathbf{M}\ddot{\mathbf{q}}(t) + \mathbf{D}\dot{\mathbf{q}}(t) + \mathbf{K}\mathbf{q}(t) = \mathbf{P}\mathbf{u}(t), \qquad \mathbf{y}(t) = \mathbf{Q}^T\mathbf{q} + \mathbf{R}^T\dot{\mathbf{q}}(t),$$

where $\mathbf{Q}$ and $\mathbf{R}$ are chosen depending on the output variable of interest. The second-order system can be formulated into an equivalent linear system of the form given in equation (17.10) such that the symmetry of the original

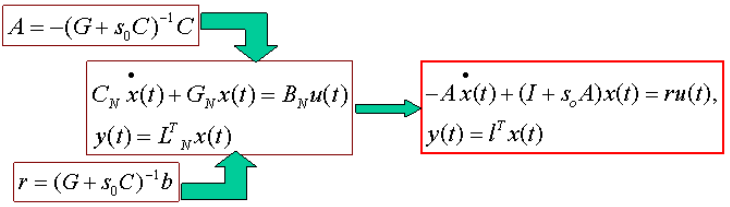


FIGURE 17.10. Basic transformations in the Krylov subspace method for macromodeling.

system is preserved, i.e.,

$$\mathbf{x}(t) = \begin{bmatrix} \mathbf{q}(t) \\ \dot{\mathbf{q}}(t) \end{bmatrix}, \qquad \mathbf{C} = \begin{bmatrix} \mathbf{D} & \mathbf{M} \\ \mathbf{F} & 0 \end{bmatrix},$$

$$\mathbf{G} = \begin{bmatrix} \mathbf{K} & 0 \\ 0 & -\mathbf{F} \end{bmatrix}, \qquad \mathbf{B} = \begin{bmatrix} \mathbf{p} \\ 0 \end{bmatrix}, \qquad \mathbf{l} = \begin{bmatrix} \mathbf{Q} \\ \mathbf{R} \end{bmatrix},$$

where $\mathbf{F}$ can be any $N \times N$ nonsingular matrix. Generally, $\mathbf{F}$ is chosen to be the identity matrix, $\mathbf{I}$, while $\mathbf{F} = \mathbf{M}$ is also a reasonable choice if $\mathbf{M}$, $\mathbf{D}$, and $\mathbf{K}$ are symmetric. The advantages of the Krylov subspace method are: (i) It is fairly accurate for linear systems and can be automated. (ii) It is computationally very effective. The disadvantages are: (i) It is not very accurate for highly nonlinear systems. (ii) It does not preserve the physical meaning of the original system.

Moment Matching Techniques

The main idea behind the moment matching technique (Ismail, 2002) is to construct the transfer function directly from the system equations using Laplace transformation, and then to approximate the transfer function by some rational function. Consider again equation (17.9). Taking the Laplace transform of this equation, the frequency domain formulation is given by

$$s\mathbf{C}\dot{\mathbf{X}}(s) + \mathbf{G}\mathbf{X}(s) = \mathbf{B}\mathbf{U}(s), \qquad \mathbf{Y}(s) = \mathbf{L}^T\mathbf{X}(s). \tag{17.11}$$

Eliminating the variable $\mathbf{X}(s)$ from equation (17.11), the input and output are related by a $p \times m$ matrix-valued rational function

$$\mathbf{H}(s) = \mathbf{L}^T(\mathbf{G} + s\mathbf{C})^{-1}\mathbf{B},$$

where $\mathbf{H}(s)$ is known as the transfer function of the linear system, and the state-space dimension of the system is N. The Taylor series expansion of

the scalar transfer function $H(s)$ about s_0 is given by

$$H(s) = \mathbf{l}^T(\mathbf{I}-(s-s_0)\mathbf{A})^{-1}\mathbf{r} = \mathbf{l}^T\mathbf{r}+(\mathbf{l}^T\mathbf{A}\mathbf{r})(s-s_0)+(\mathbf{l}^T\mathbf{A}^2\mathbf{r})(s-s_0)^2+\cdots$$
$$= m_0 + m_1(s-s_0) + m_2(s-s_0)^2 + \cdots,$$

where m_j are the moments about s_0. The objective is to approximate $H(s)$ by a rational function $H_q(s) \in \Re^{q-1,q}$ over the range of frequencies of interest, where $q \leq N$. One choice is the Padé approximation (Bultheel and Barvel, 1986). A function $H_q(s) \in \Re^{q-1,q}$ is said to be a qth Padé approximant of $H(s)$ about an expansion point s_0 if it matches with moments of $\mathbf{H}(s)$ as far as possible. It is required that (Bai, 2002)

$$H(s) = H_q(s) + O((s-s_0)^{2q}). \tag{17.12}$$

Note that we have $2q$ conditions on the $2q$ degrees of freedom that describe the approximation function. Specifically, let

$$H_q(s) = \frac{P_{q-1}(s)}{Q_q(s)} = \frac{a_0 + a_1 s + a_2 s^2 + \cdots + a_{q-1}s^{q-1}}{1 + b_1 s + b_2 s^2 \cdots + b_q s^q}.$$

The coefficients $[a_i]$ and $[b_i]$ of the polynomials $P_{q-1}(s)$ and $Q_q(s)$, and also the moments can be computed by multiplying both sides of equation (17.12) by $Q_q(s)$ and comparing the first $q(s-s_0)^k$ terms for $k = 0, 1, 2, \ldots, n-1$. A system of $2q$ nonlinear equations is solved to find the $2q$ unknowns. This takes into account the dominant q poles, while the poles larger than this value are neglected.

The advantages of this method are: (i) Reduction in computational effort. (ii) A wide variety of physical phenomena encountered in microsystems, including dissipation, can be modeled. (iii) The accuracy can be improved by taking more moments at each node. (iv) Static, steady-state, and transient analysis can be performed. The disadvantages of this approach are: (i) It is applicable for linear dynamical systems only. (ii) It is inefficient if the number of inputs is large. (iii) It is not stable for higher-order approximations. (iv) It is computationally expensive for each expansion point.

The multinode moment matching (MMM) method (Ismail, 2002) is an extension of the single point moment matching (SMM) method and has much better efficiency than to the SMM technique. The MMM technique simultaneously matches the moments at several nodes of a circuit using explicit moment matching around $s = 0$. MMM requires a smaller computational effort, since only $(q+1)$ moments are required (see (Ismail, 2002) for details). MMM is numerically stable, as the higher powers are not used in the expansion, avoiding truncation errors.

17.3.3 Nonlinear Dynamic Models

Nonlinear dynamic models are frequently encountered in microsystems. Linearizing the nonlinear equations and using reduced-order methods like

the linear Krylov subspace method based on a Lanczos process or other linear basis function techniques may not be sufficient to capture the nonlinear behavior of the system. Arnoldi-based Krylov subspace methods (Chen and White, 2000) and the trajectory piecewise linear approach (Rewienski and White, 2001) and its modifications are found to work better for nonlinear systems. These techniques are summarized in this section.

Krylov Subspace Technique Based on the Arnoldi Method

Consider a nonlinear system of the form

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}) + \mathbf{b}\mathbf{u}(t), \qquad \mathbf{y} = \mathbf{l}^T\mathbf{x}, \tag{17.13}$$

where $\mathbf{x}$ is a vector of length n, $\mathbf{f}$ is a nonlinear vector function, $\mathbf{u}(t)$ is the input of the system, and $\mathbf{y}(t)$ is the output. Taylor series expansion of the function $\mathbf{f}$ about the origin (the equilibrium point) to second order yields a quadratic approximation of the form

$$\dot{\mathbf{x}} = \mathbf{J_f}\mathbf{x} + \mathbf{x}^T\mathbf{W}\mathbf{x} + \mathbf{b}\mathbf{u}(t), \qquad \mathbf{y} = \mathbf{l}^T\mathbf{x}, \tag{17.14}$$

where $\mathbf{J_f}$ is the Jacobian of $\mathbf{f}$ evaluated at the origin and $\mathbf{W}$ is an $N \times N \times N$ Hessian tensor. The matrices $\mathbf{J_f}$ and $\mathbf{W}$ are given by

$$\mathbf{J_f}_{i,j} = \frac{\partial f_i}{\partial x_j} \quad \text{and} \quad \mathbf{W}_{i,j,k} = \frac{\partial^2 f_i}{\partial x_j \partial x_k}.$$

We assume that $\mathbf{J_f}$ is nonsingular. Let $\mathbf{A} = \mathbf{J_f}^{-1}$ be the inverse of the Jacobian. Multiplying equation (17.14) by $\mathbf{A}$ yields

$$\mathbf{A}\dot{\mathbf{x}} = \mathbf{x} + \mathbf{A}\mathbf{x}^T\mathbf{W}\mathbf{x} + \mathbf{A}\mathbf{b}\mathbf{u}(t), \qquad \mathbf{y} = \mathbf{l}^T\mathbf{x}. \tag{17.15}$$

The orthogonal basis for the Krylov subspace span$[\mathbf{Ab}, \mathbf{A}^2\mathbf{b}, \ldots, \mathbf{A}^q b]$, where $q \ll N$, is the size of the reduced system that will be generated by using the Arnoldi method (Chen and White, 2000) for numerical stability. The Arnoldi process generates $\mathbf{V}$, an $n \times q$ orthonormal matrix whose columns span the Krylov subspace. Using the change of variables $\mathbf{x} = \mathbf{V}\mathbf{z}$ in equation (17.15), we have

$$\mathbf{A}\mathbf{V}\dot{\mathbf{z}} = \mathbf{V}\mathbf{z} + \mathbf{A}\mathbf{z}^T\mathbf{V}^T\mathbf{W}\mathbf{V}\mathbf{z} + \mathbf{A}\mathbf{b}\mathbf{u}(t), \qquad \mathbf{y} = \mathbf{l}^T\mathbf{V}\mathbf{z}.$$

Left-multiplying by $\mathbf{V}^T$, and defining $\mathbf{H} = \mathbf{V}^T\mathbf{A}\mathbf{V}$, we have

$$\mathbf{H}\dot{\mathbf{z}} = \mathbf{z} + \mathbf{V}^T\mathbf{A}\mathbf{z}^T\mathbf{V}^T\mathbf{W}\mathbf{V}\mathbf{z} + \mathbf{V}^T\mathbf{A}\mathbf{b}\mathbf{u}(t), \qquad \mathbf{y} = \mathbf{l}^T\mathbf{V}\mathbf{z}.$$

The system can be expressed in the original form (equation (17.13)) by left-multiplying by $\mathbf{H}^{-1}$ to obtain

$$\dot{\mathbf{z}} = \mathbf{H}^{-1}\mathbf{z} + \mathbf{H}^{-1}\mathbf{V}^T\mathbf{A}\mathbf{z}^T\mathbf{V}^T\mathbf{W}\mathbf{V}\mathbf{z} + \mathbf{H}^{-1}\mathbf{V}^T\mathbf{A}\mathbf{b}\mathbf{u}(t), \qquad \mathbf{y} = \mathbf{l}^T\mathbf{V}\mathbf{z}. \tag{17.16}$$

Setting

$$\mathbf{J}' = \mathbf{H}^{-1}, \qquad \mathbf{b}' = \mathbf{H}^{-1}\mathbf{V}^T\mathbf{b}, \qquad \mathbf{l}' = \mathbf{V}^T\mathbf{l},$$

where $\mathbf{H}^{-1}\mathbf{V}^T\mathbf{A}\mathbf{z}^T\mathbf{V}^T\mathbf{W}\mathbf{V}\mathbf{z}$ is quadratic in $\mathbf{z}$ and can be written in the form $\mathbf{z}^T\mathbf{W}'\mathbf{z}$ for some $\mathbf{W}'$. Then, equation (17.16) can be reduced to a quadratic system of the form

$$\dot{\mathbf{z}} = \mathbf{J}'\mathbf{z} + \mathbf{z}^T\mathbf{W}'\mathbf{z} + \mathbf{b}'\mathbf{u}(t), \qquad \mathbf{y}' = \mathbf{l}'^T\mathbf{z},$$

where $\mathbf{y}'$ is an approximation to $\mathbf{y}$. The key step in this approach is the use of Arnoldi projection to reduce the large quadratic tensor to a small quadratic tensor.

The merits of this approach are: (i) This method is much more accurate than the linearized models and can be automated. (ii) It is computationally very effective. The problem with this approach is that it is not very accurate for highly nonlinear systems even though it has a quadratic nonlinear term. If higher-order terms are included, the cost of the reduced order model increases as $O(n^4)$, and the number of coefficients to be evaluated is very large.

Trajectory Piecewise-Linear Approach

The key idea in the trajectory piecewise-linear approach is based on representing the nonlinear system with a piecewise-linear system and then reducing each of these pieces with Krylov subspace projection methods (Rewienski and White, 2001). Instead of approximating the individual components as piecewise-linear and then composing hundreds of components to make a system with exponentially many different linear regions, a small set of linearizations is generated about the state trajectory, which is the response to a "training input." Introducing the change of variables $\mathbf{x} = \mathbf{V}\mathbf{z}$ in equation (17.13) and multiplying the resulting equation by $\mathbf{V}^T$ yields

$$\dot{\mathbf{z}} = \mathbf{V}^T\mathbf{f}(\mathbf{V}\mathbf{z}) + \mathbf{V}^T\mathbf{b}\mathbf{u}(t) \quad \text{and} \quad \mathbf{y} = \mathbf{l}^T\mathbf{V}\mathbf{z}.$$

The two key issues here are first, selecting a reduced basis $\mathbf{V}$ such that the system provides a good approximation of the original system. This has already been addressed in the previous sections. The second issue, which makes the Taylor-series-expansion-based reduced-models inefficient, is the computation of the term $\mathbf{V}^T\mathbf{f}(\mathbf{V}\mathbf{z})$. For linear and quadratic reduced-order models, the linear and the quadratic terms from the Taylor expansion about an equilibrium point $\mathbf{x}_0$ are considered, and all higher-order terms are neglected, i.e.,

$$\mathbf{f}(\mathbf{x}) \approx \mathbf{f}(\mathbf{x}_0) + \mathbf{A}_0(\mathbf{x} - \mathbf{x}_0) + \frac{1}{2}\mathbf{W}_0(\mathbf{x} - \mathbf{x}_0) \otimes (\mathbf{x} - \mathbf{x}_0),$$

where $\otimes$ is the Kronecker product and $\mathbf{A}_0$ and $\mathbf{W}_0$ are the Jacobian and the Hessian of $\mathbf{f}(\cdot)$. For the linear case, the reduced-order model becomes

$$\dot{\mathbf{z}} = \mathbf{V}^T\mathbf{f}(\mathbf{x}_0) + \mathbf{A}_{0r}\mathbf{z} + \mathbf{V}^T\mathbf{b}\mathbf{u}(t) \quad \text{and} \quad \mathbf{y} = \mathbf{l}^T\mathbf{V}\mathbf{z},$$

while for the quadratic case, the reduced order model becomes

$$\dot{\mathbf{z}} = \mathbf{V}^T\mathbf{f}(\mathbf{x}_0) + \mathbf{A}_{0r}\mathbf{z} + \frac{1}{2}\mathbf{W}_{0r}(\mathbf{z}\otimes\mathbf{z}) + \mathbf{V}^T\mathbf{b}\mathbf{u}(t) \quad \text{and} \quad \mathbf{y} = \mathbf{l}^T\mathbf{V}\mathbf{z},$$

where $\mathbf{A}_{0r} = \mathbf{V}^T\mathbf{A}_0\mathbf{V}$ and $\mathbf{W}_{0r} = \mathbf{V}^T\mathbf{W}_0(\mathbf{V}\otimes\mathbf{V})$ are $q\times q$ and $q\times q^2$ matrices, respectively, which are typically dense and must be represented explicitly. As a result, the cost of computing $\mathbf{V}^T\mathbf{f}(\mathbf{V}\mathbf{z})$ and the cost of storing the matrices $\mathbf{A}_{0r}$ ($\mathbf{A}_{0r}$ and $\mathbf{W}_{0r}$ for the quadratic case) are $O(q^2)$ in the linear case and $O(q^3)$ in the quadratic case. Hence, although the method based on Taylor expansion may be extended to higher orders of nonlinearities, this approach is limited in practice to cubic expansions due to exponentially growing memory and computational costs.

In the piecewise-linear approach, s linearized models of the nonlinear system with expansion points $\mathbf{x}_0,\ldots,\mathbf{x}_{s-1}$ are considered, i.e.,

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}_i) + \mathbf{A}_i(\mathbf{x}-\mathbf{x}_i) + \mathbf{b}\mathbf{u}(t),$$

where $\mathbf{x}_0$ is the initial state of the system and $\mathbf{A}_i$ are the Jacobians of $\mathbf{f}(\cdot)$ evaluated at the states $\mathbf{x}_i$. Considering a weighted combination form,

$$\dot{\mathbf{x}} = \sum_{i=0}^{s-1}[w_i(\mathbf{x})\mathbf{f}(\mathbf{x}_i) + w_i(\mathbf{x})\mathbf{A}_i(\mathbf{x}-\mathbf{x}_i)] + \mathbf{b}\mathbf{u}(t),$$

where $w_i(\mathbf{x})$ are weights that sum to 1. Assuming that a qth-order basis $\mathbf{V}$ has already been generated, the following reduced-order model is obtained:

$$\dot{\mathbf{z}} = (\mathbf{A}_r\cdot\mathbf{w}'(\mathbf{z})^T)\mathbf{z} + \gamma\cdot\mathbf{w}'(\mathbf{z})^T + \mathbf{b}_r\mathbf{u}(t), \qquad \mathbf{y} = \mathbf{l}_r\mathbf{z},$$

where $\gamma = \left[\mathbf{V}^T(\mathbf{f}(\mathbf{x}_0)-\mathbf{A}_0\mathbf{x}_0),\ldots,\mathbf{V}^T(\mathbf{f}(\mathbf{x}_{s-1})-\mathbf{A}_{s-1}\mathbf{x}_{s-1})\right]$, $\mathbf{b}_r = \mathbf{V}^T\mathbf{b}$, $\mathbf{l}_r = \mathbf{l}^T\mathbf{V}$, $\mathbf{A}_r = \left[\mathbf{A}_{0r}\mathbf{A}_{1r}\ldots\mathbf{A}_{(s-1)r}\right]$, $\mathbf{w}' = [w_0, w_1, \ldots, w_{s-1}]$ is a vector of weights, and $\mathbf{A}_{ir} = \mathbf{V}^T\mathbf{A}_i\mathbf{V}$. The weights are computed in the following manner.

1. For $i = 0,\ldots,(s-1)$ compute $d_i = \|\mathbf{z}-\mathbf{z}_i\|_2$.
2. Compute $m = \min[d_i : i = 1,\ldots,(s-1)]$.
3. For $i = 0,\ldots,(s-1)$ compute $w_i = (\exp(d_i)/m))^{-25}$.
4. Normalize w_i.

This implies that the linearized point that is closest to the current position gets the maximum weight. Instead of finding linearized models covering the entire N-dimensional state space, a collection of models is generated along a single trajectory of the system. This is done by simulating the system at the initial point and moving ahead by a very small interval from that point to get a new point, and the process is repeated for each point. However, this method requires performing simulation of the initial nonlinear system, which may be very costly due to the initial size of the problem. This problem is avoided by using the Arnoldi-based Krylov subspace method, instead of the full simulation, to simulate the nonlinear system and obtain approximate trajectory and linearization points, making the process faster.

Further improvements to the method have been reported in (Rewienski and White, 2002), where a richer aggregated reduced basis is obtained by applying the Arnoldi method at each linearization point instead of only once. This results in improved accuracy and consequently reduces the order of the reduced model further. In the original implementation, the projection matrix $\mathbf{V}$ was constructed using a Krylov subspace based on a linearization about the initial state x_0. In the new implementation, the above approach has been replaced by a three-step procedure. First, at each of the linearization points $\mathbf{x}_i$, a reduced-order basis is generated in a suitable Krylov space corresponding to a linearized model generated at $\mathbf{x}_i$. Second, the union of all the bases is formed, and third, the set is reduced using singular value decomposition. The method for basis generation was replaced from the Arnoldi-based Krylov subspace method to the TBR (Truncated Balanced Realization) algorithm in (Vasilyev et al., 2003), and a hybrid method using both TBR and Krylov subspace has been implemented. It was found that the TBR-based methods gave better accuracy than the method in which the Krylov subspace was used solely.

17.4 Galerkin Methods

Galerkin methods are popular techniques for reduced-order modeling. In this section, we summarize both linear and nonlinear Galerkin methods for reduced-order modeling.

17.4.1 Linear Galerkin Methods

The objective is to create a set of coupled ordinary differential equations that give an accurate representation of the dynamical behavior of the device. The approach is to formulate the dynamical behavior in terms of a finite set of orthonormal spatial basis functions, each with a time-dependent coefficient. Though this method is not typically analytical, it still forms a very important tool in the reduced-order modeling of microsystems that

cannot be represented as lumped elements. Because of the relatively small number of state variables, the models can be quickly evaluated, and integrated over time. Such models can be readily inserted into circuit simulators for behavioral representation at the system level, including feedback effects around nonlinear devices. For completely numerical sets of ODEs, automatic model-order reduction can be implemented, at least for linear problems, and for nonlinear problems by using a combination of methods like the Krylov subspace techniques. The two most popular methods that fall under this category are the linear modes of vibration and Karhunen–Loève decomposition.

Linear Modes of Vibration

The basic idea in this method is to represent the physical variable, e.g., the deformed shape of the microdevice, as a summation of the linear normal mode shapes. This results in the transformation from the nodal coordinates to the time-dependent coefficients of the mode shapes, called modal coordinates (Ananthasuresh et al., 1996). This approach also eliminates the coupling between the inertia and stiffness matrices of the governing equations. Assuming that higher modes of vibration have negligible effect on the system's response, a reduced-order model is obtained by using only the first few modes. Instead of the original system of N coupled equations, N being the total number of degrees of freedom, only n equations need to be solved in the reduced-order model, where n is the number of modes considered. The number of modes considered determines both the accuracy and the computation time of the system. A general procedure for this method is given as (Ananthasuresh et al., 1996):

1. Derive basis functions from an initially meshed structure by solving for the small-amplitude (linear) modes of a structure.
2. Form a basis set that is orthonormal in the state space.
3. Consider the first few modes to represent the physical variable(s) of interest (e.g., structural deformation).
4. Represent the solution to the system as a linear combination of the modes with time-dependent coefficients.

The undamped dynamical behavior of a fully meshed structure is given by

$$\mathbf{M}\ddot{\mathbf{x}} + \mathbf{K}\mathbf{x} = \mathbf{F}(\mathbf{x}, t), \tag{17.17}$$

where $\mathbf{M}$ is the global inertial matrix, $\mathbf{K}$ is the global stiffness matrix, and $\mathbf{F}(\mathbf{x}, t)$ is the nonlinear external force. Let $\mathbf{S}$ be the modal matrix, i.e., an $N \times N$ matrix whose columns are the mode shape vectors. The generalized inertia matrix $\mathbf{M_G}$ and the generalized stiffness matrix $\mathbf{K_G}$ are defined as

$$\mathbf{M}_G = \mathbf{S}^T\mathbf{M}\mathbf{S} \quad \text{and} \quad \mathbf{K}_G = \mathbf{S}^T\mathbf{K}\mathbf{S},$$

where both $\mathbf{M}_G$ and $\mathbf{K}_G$ are diagonal. Substituting $\mathbf{x} = \mathbf{Sq}$ in equation (17.17) and premultiplying by $\mathbf{S^T}$, we get

$$\mathbf{M}_G\ddot{\mathbf{q}} + \mathbf{K}_G\mathbf{q} = \mathbf{S}^T\mathbf{F}(\mathbf{Sq}, t).$$

Since $\mathbf{M}_G$ and $\mathbf{K}_G$ are diagonal matrices, the coupling of the $\mathbf{q}$'s is through the nonlinear force term. In practice, only a few modes are sufficient to describe the deformation. So the N equations are reduced to a smaller number of m equations, where m is the number of mode shapes considered.

If damping properties are to be included, the damping force is added as an additional term on the right-hand side of the equation, or a new set of geometric basis functions is generated by including the space external to the structure where the damping is present. If an additional force term is added, it contains a dependence on velocity. Another approach to account for damping is to assume Rayleigh damping, in which case the linear modes obtained from $\mathbf{M}$ and $\mathbf{K}$ would be sufficient to capture the full behavior. However, in general, to include the effect of damping, one may have to solve the quadratic eigenvalue problem $(\lambda^2\mathbf{M} + \lambda\mathbf{D} + \mathbf{K})\mathbf{s} = 0$ for the desired modal matrix $\mathbf{S}$.

If the structure undergoes large-amplitude deformation, then in an ideal case, the stiffness matrix needs to be recomputed as the amplitude changes. An alternative approach is to retain the original stiffness matrix and add an extra force term to account for the large-amplitude effects, such as stress stiffening of the structure. It is convenient to express the right-hand side in terms of modal coordinates instead of the meshed coordinates. The energy method (Senturia, 1998b) can be used for this purpose, and this procedure is summarized below:

1. Find the linear modes for the elastic problem assuming the no-load condition.

2. Perform quasi-static simulation over a design space that includes the deformations described by a superposition of p modes and develop a suitable potential energy function for other conservative forces (e.g., electrostatic) and large-amplitude elastic effects (e.g., for stress stiffening). Create analytical expressions for the variation of potential energy as a function of the selected mode set (this function is nonlinear and must include products of modal amplitudes, etc.).

3. Replace the right-hand side of the modal dynamic formulation with suitable derivatives of the potential energy functions with respect to modal amplitudes. The net result is a small coupled set of $2p$ (2 state variables per mode) ODEs that can be easily integrated forward in time, without requiring any conversions to and from the original meshed space.

The advantage with modal methods is that they break open the coupled-domain problem. The original modal basis functions are obtained from a

single energy domain, e.g., elasticity, together with the associated mass distribution, and the nonlinear potential energy functions can be computed one energy domain at a time. Therefore, it is not necessary to perform complex self-consistent coupled-domain simulations. This approach requires many (single-energy domain) simulations combined with fitting parameters to obtain analytical functions. So it is difficult to do it manually. It is possible to automate the procedure for nonlinear conservative problems. There are limits to the basis-function approach. Thus far, it has been difficult to calculate accurately the stress stiffening of an elastic body undergoing large-amplitude deformation using superposition of modal coordinates. Additionally, when the device undergoes nonlinear motion, such as contact, modal approaches fail. However, the class of microsystems that can now be handled with the automated basis-function approach is large enough to be interesting. More examples using modal basis functions for MEMS simulations are given in (Gabbay and Senturia, 1998; Varghese et al., 1999).

Karhunen–Loève Decomposition Method

The basic idea in the Karhunen-Loève (KL) decomposition method is similar to the basic idea in the linear modes method, i.e., to develop a few global basis functions to represent the entire system by a reduced-order model. In the case of the linear modes method, the linear modes of the system obtained through the solution of the generalized eigenvalue problem form the set of basis functions. Karhunen–Loève decomposition is another method to generate global basis functions, and the advantage with the KL decomposition is that it works better than the linear modes technique for nonlinear cases. The procedure for the extraction of the basis functions in the KL decomposition method is summarized below:

1. Simulate the entire system dynamics first by using a time-stepping scheme that is stable and known to give accurate results.

2. The spatial distributions of each state variable $\mathbf{u}(x,t)$ are sampled at a series of t_n different time instants during the simulations. These sampled distributions are stored as a series of vectors $\mathbf{u}_i$, and each vector represents a "snapshot" in time.

3. The basis functions are determined using either the singular value decomposition (SVD) or the KL approach.

In the SVD approach, which is mathematically equivalent to the KL decomposition technique, n orthogonal basis functions $[\mathbf{a}_1, \ldots, \mathbf{a}_n]$ are determined by minimizing the following expression:

$$\sum_{i=1}^{t_n} \mid \mathbf{u}_i - \text{proj}(\mathbf{u}_i, \text{span}[\mathbf{a_1}, \ldots, \mathbf{a_n}]) \mid^2 . \tag{17.18}$$

This is accomplished by taking the singular value decomposition (SVD) of the matrix $\mathbf{U}$, whose columns are $\mathbf{u}_i$. The SVD of $\mathbf{U}$ is given by

$$\mathbf{U} = \mathbf{V}\boldsymbol{\Sigma}\mathbf{W}^T,$$

where $\mathbf{V}$ contains the eigenvector as columns and $\boldsymbol{\Sigma}$ contains the eigenvalues in the diagonal. By taking $\mathbf{a}_i = \mathbf{v}_i$ for $i = 1, 2, \ldots, n$, the minimization of the expression in equation (17.18) is accomplished.

The KL approach (Sirovich, 1987) is a procedure for extracting an empirical basis for a modal decomposition from an ensemble of signals. Assuming that the signals are an ensemble of the functions $\mathbf{u}_i$, the objective is to find a single deterministic function that is most similar to members of $\mathbf{u}_i$ on average. In other words, one needs to find a function that maximizes the inner product with the field $\mathbf{u}_i$. That is, one needs to maximize

$$\lambda = \frac{\langle (\phi, \mathbf{u}_i)^2 \rangle}{(\phi, \phi)},$$

where $\langle\rangle$ is the averaging operator, which can be a time, space, or ensemble average, and $(\phi, \mathbf{u}_i) = \int_\Omega \phi(x)\mathbf{u}_i(x)d\Omega$ is the inner product defined in the function space Ω. It turns out that this condition is met when $\phi(x)$ is an eigenfunction of the two-point tensor given by

$$K\phi = \int_\Omega K(x, x')\phi(x')dx' = \lambda\phi(x),$$

where $K(x, x')$ is a nonnegative Hermitian operator given by

$$K(x, x') = \langle \mathbf{u}_i(x)\mathbf{u}_i(x') \rangle = \frac{1}{t_n} \sum_{i=1}^{t_n} \mathbf{u}_i(x)\mathbf{u}_i^T(x').$$

The above equation can be solved by the direct method or by the method of "snapshots" (Sirovich, 1987). In the method of "snapshots" the eigenfunction can be represented as the summation of snapshots $\mathbf{u}_i$, i.e.,

$$\phi(x) = \sum_k \alpha_k \mathbf{u}_k(x). \tag{17.19}$$

Substituting this into the two-point tensor equation yields a matrix eigenvalue problem that determines the eigenvalues and eigenvectors α. Substituting this set of eigenvectors in equation (17.19) yields a set of eigenfunctions $\phi(x)$, which is the set of global basis functions. It is important to note that the eigenfunction corresponding to the largest eigenvalue corresponds to the most energetic of the snapshots ensemble followed by the eigenfunction corresponding to the next-largest eigenvalue, and so on. After the basis functions are generated by using either the SVD or the KL method, the Galerkin procedure is employed with the basis functions to

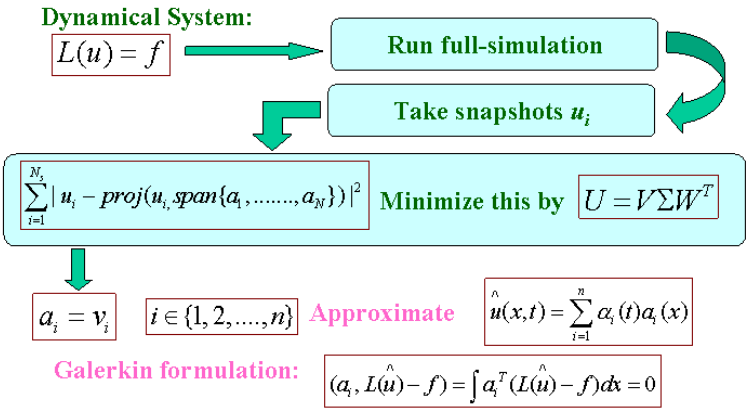


FIGURE 17.11. Basic steps in the Karhunen–Loève decomposition method.

convert the original nonlinear governing PDE to a set of coupled ODEs much smaller in size. Figure 17.11 explains the various steps for implementing the Karhunen–Loève decomposition method.

The advantages of this method are: (i) In general, very few basis functions are required. (ii) Nonlinear behavior is efficiently captured. (iii) The same basis functions can be employed to simulate different input parameters, even though regeneration of the basis functions for the new input parameters can provide more accurate results. The disadvantages of the method are: (i) Using a combined state vector instead of independent basis functions can result in a reduced number of basis functions, but this can distort the physical meaning of the problem. (ii) Problems can still occur, since the basis functions chosen may not capture the entire state space. This is usually the case when there are multiple attractors, rare intermittent fluctuations, or bifurcations in the parameter space. (iii) Complications can result in systems exhibiting intermittent chaotic behavior. (iv) The overhead with the initial full-scale simulation is quite high. Low-dimensional flow dynamical systems may converge to erroneous states after long-time integration and can be observed in reduced-order models constructed from the Karhunen–Loève decomposition method. A dissipative model based on a spectral viscosity diffusion convolution operator has been used in (Sirisup and Karniadakis, 2004), for improving the asymptotic behavior of KL predictions.

In spite of the existing voids, the KL decomposition technique is promising for nonlinear dynamic simulation, especially as the complexity increases. As coupled domain simulation tools improve, this approach can be a powerful tool for microsystem simulation. The standard KL decomposition has been modified under several circumstances to generate more

efficient macromodels. Nonlinear responses have been well captured by an arc-length-based KL decomposition presented in (Chen and Kang, 2001b). This method is motivated by the fact that while rapidly varying events can occur in a very short time period, they typically traverse a relatively large interval in the phase space. Hence, considering an ensemble average based not only on time but also on arc length in phase space can lead to better macromodels. This has been shown in (Chen and Kang, 2001b), for a capacitive pressure sensor, where the arc-length-based approach is found to capture the rapidly changing dynamics of the device better than the standard KL approach. Further analysis of this approach shows that the arc-length-based ensemble average is a weighted time average, with the weighting function equal to the magnitude of the vector field, thereby stressing the event of rapid change.

Weighted Karhunen–Loève Decomposition Method

Another modification to the standard KL decomposition is the use of a weighting function (Qiao and Aluru, 2003c; Graham and Kevrekidis, 1996; Zhang et al., 2003). The basic idea is that instead of trying to minimize equation (17.18), we assign different weights to different snapshots and try to minimize the weighted residual, i.e.,

$$\sum_{i=1}^{N_s} \mid w_i \mathbf{u}_i - \text{proj}\,(w_i \mathbf{u}_i, \text{span}\{\boldsymbol{a}_1, \ldots, \boldsymbol{a}_N\}) \mid^2. \tag{17.20}$$

Observe the difference between equations (17.18) and (17.20): w_i is the weighting assigned to snapshot $\mathbf{u}_i$. In the weighted KL approach, instead of minimizing a least-squares measure of "error" between the linear subspace spanned by the basis functions and the observation space, we minimize the weighted "error" between these two spaces.

By using the fact that the SVD of a snapshot ensemble gives the basis that minimizes equation (17.18), it is easy to show that the basis that minimizes equation (17.20) is the column vector of matrix $\boldsymbol{L_2}$:

$$\tilde{\boldsymbol{U}}\boldsymbol{W} = \boldsymbol{L_2}\boldsymbol{\Sigma_2}\boldsymbol{R_2}^T, \tag{17.21}$$

where $\boldsymbol{W}$ is an $N_s \times N_s$ diagonal matrix whose diagonal elements are the weighting coefficients for each snapshot, i.e., $[\boldsymbol{W}]_{i,i} = w_i$.

Remarks:

1. If the weighting function matrix is the identity, i.e., $w_i = 1$ $(i = 1, 2, \ldots, N_s)$, then the weighted KL technique and the standard KL technique produce identical bases.

2. In the implementation of the weighted KL technique, once the snapshots are obtained, a weight is assigned to each snapshot.

3. Since the snapshot ensemble matrix is multiplied by a diagonal matrix, the computational cost of a weighted KL decomposition based on equation (17.21) is almost the same as the computational cost of the classical KL decomposition.

Significance of Weighting: The concept of assigning different weights to different snapshots is useful when the transient behavior of certain variables (for example, velocity or pressure) changes significantly with time. For example, in the case of electroosmotic transport, the flow gets to steady state at different times for different locations in the channel (see the discussion in (Qiao and Aluru, 2003c), for more details). If a higher weighting is assigned to those snapshots taken during the fast-changing transient, then the basis obtained with SVD will, according to equation (17.20), be able to produce more accurate results. In other words, the new basis obtained with weighted snapshots will be able to represent the system behavior much better than the basis functions obtained with the classical KL decomposition technique. If the transient behavior of the system is gradual, then the use of the weighting function is limited, and both weighted and classical KL decomposition techniques can be expected to produce comparable accuracy results.

A feasible approach for rapidly varying transient solutions is to obtain more snapshots during the time when the solution is changing rapidly and to compute the basis using the classical KL decomposition technique. However, there are several situations in which obtaining snapshots is not straight-forward. For example, when snapshots are obtained from experiments, repeating the experiment to obtain more snapshots can be very expensive. Similarly, if the snapshots are obtained from numerical simulations and if a rapidly varying transient is represented by a few snapshots, repeating the simulation to get more snapshots with a smaller time step can be very expensive. A good compromise in such cases is to use weighted snapshots to get better basis functions, instead of repeating the experiments or the numerical simulations. Many times it is difficult to foresee the various time scales encountered in the system. The concept of weighting in a KL decomposition technique introduces more flexibility and accuracy to represent multiple time scales in a dynamical system.

Discussion on Weighted Basis versus Error in the Solution: It is important to note that weighting is a concept introduced to improve the accuracy over certain time scales or periods rather than a technique that can be used to improve the accuracy over the *entire* time period. In fact, reduced-order modeling using a weighted basis usually exhibits slightly higher error in the time period that is less significantly weighted. Typically, reduced-order modeling exhibits a very nonuniform error in the whole time domain, i.e., it might behave very well in certain time periods but not be able to capture the basic characteristics in certain other time periods. By using a weighted KL basis, it is possible to achieve a more uniform reduction in error in the

solution.

A second question of significant interest is, how does a weighted basis compare with other bases in capturing the system transient? There is no easy answer without a detailed mathematical analysis. However, we do know that (1) increasing the number of basis functions used in approximating, for example, the velocities and pressure generally improves the accuracy of the simulation; (2) different methods generate different bases, and the number of significant basis functions that need to be included in each method is different. The accuracy of the solution is influenced by both the number of basis functions and the quality of the bases. Typically, the number of basis functions that need to be included in a weighted approach is less than the number of basis functions that need to be included in the classical KL approach for comparable accuracy.

Example: Transient Analysis of Electroosmotic Flow

Electroosmotic flow (see Chapter 7 for governing equations and other details) in a cross channel, shown in Figure 17.12, is used as an example to demonstrate the KL and the weighted KL techniques. The cross channel is an interesting problem, since the flow in the intersection exhibits many interesting characteristics. In the case of balanced applied potentials, the net flow into the side channel is negligible but the fluid velocity does not vanish in the side channel. A good reduced-order model should capture both the flow in the main channel and flow within the intersection of the cross channel. The flow in the cross channel exhibits multiple time scales, i.e., the flow within the intersection reaches steady state much more quickly than the flow in the main channel. In addition, the velocity profile within the intersection is more complex than the velocity profile in the main channel. To capture the multiple time scales encountered in the cross channel example, a weighted KL decomposition is used to generate the basis functions for the reduced-order model.

Sixty snapshots are used to generate a reduced-order model. The snapshots are equispaced in time with a time period of 8.85 μs between snapshots. Figure 17.13(a) shows the weighting function employed to generate the weighted KL basis. The weighting coefficient, $w(i)$, for the ith snapshot is calculated by

$$w(i) = r + (1 - r)\frac{e^{-(i/c)^2} - e^{-(N_s/c)^2}}{1 - e^{-(N_s/c)^2}}, \tag{17.22}$$

where N_s is the total number of snapshots, r is the minimal weighting for all snapshots, and c is a parameter controlling the steepness of the weighting function. In this calculation, since the first few snapshots contain the information of how V-velocity near the intersection reaches steady state, they are weighted more heavily than the other snapshots. The snapshots closer to the steady-state value are not critical, so they are assigned a lower

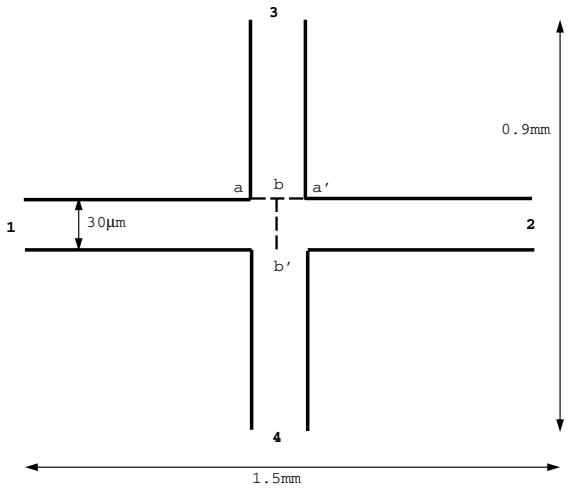
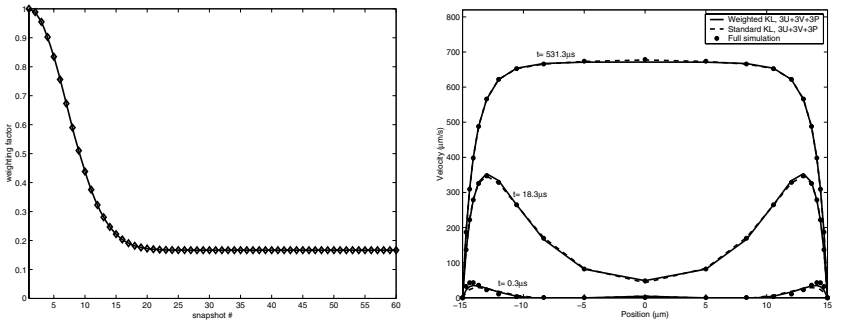


FIGURE 17.12. Geometry of a cross channel used in transient analysis.

weighting. The minimal weighting is set to be 1/6. The steepness parameter is chosen to be 8.5 in this case.

Figure 17.13(b) shows the U-velocity prediction in the main channel, and both methods (standard and weighted KL techniques) produce almost identical results at steady state, though the reduced-order model using weighted KL basis gives slightly better accuracy at the beginning of the simulation. In Figure 17.14(a), we compare the performance of weighted and standard KL techniques by fixing the number of snapshots and investigating the number of basis functions required with each technique to get comparable accuracy. We use 60 snapshots in each approach. For the weighted KL technique we use 4 basis functions for the x-component of the velocity, 4 basis functions for the y-component of the velocity, and 3 basis functions for the pressure (referred to as (4U+4V+3P) in Figure 17.14(a)). To reproduce the results obtained by the weighted KL technique, 6 basis functions had to be used for the x-velocity, 6 basis functions for the y-velocity, and 6 basis functions for the pressure (denoted by (6U+6V+6P)). For comparable accuracy, we need almost *twice* the number of basis functions in a standard KL approach than in the weighted KL technique. This result indicates that when the number of snapshots is fixed, a weighted KL technique needs fewer basis functions than a standard KL technique to reproduce full simulation results. The use of fewer basis functions leads to a computationally more efficient approach. We also observed that a technique with (4U+4V+3P) basis functions is almost twice as fast as the technique with (6U+6V+6P) basis functions; i.e., a reduced-order model based on the weighted KL technique can be twice as fast as the reduced-order model based on the standard KL technique while achieving essentially the same

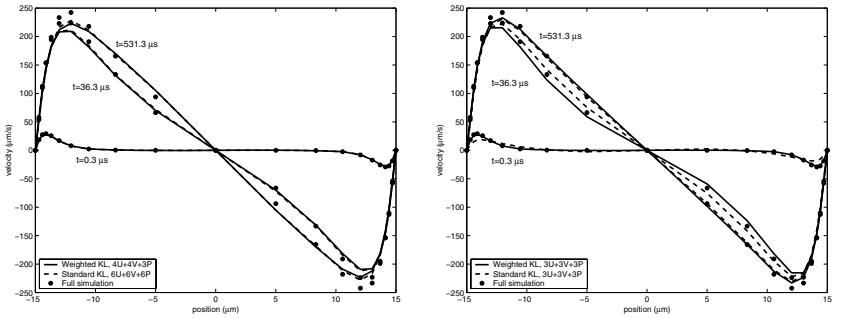


(a) (b)

FIGURE 17.13. (a) A weighting function that assigns different weighting coefficients for different snapshots. (b) Comparison of U-velocity in the main channel far upstream of the intersection. 60 snapshots are used and 3U+3V+3P basis functions are used in both methods.

accuracy.

In Figure 17.14(b), we compare the performance of weighted and standard KL techniques by fixing the number of snapshots and basis functions. The number of snapshots is fixed to 20, and the number of basis functions for U, V, and P is fixed to 3, i.e., (3U+3V+3P). The snapshots are equispaced in time with a time period of 26.6 μs between snapshots. The weighting coefficients for the weighted KL technique are computed using equation (17.22). The minimal weighting r is 1/6, and the steepness parameter is set to be 9. Figure 17.14(b) compares the weighted and standard KL techniques with the full transient simulation. The results indicate that the weighted KL basis is able to capture the velocity profile during the initial transient much more effectively than the standard KL approach. The steady-state solution predicted by both techniques is almost the same and compares well with the full transient simulation. From this we can conclude that with the same number of snapshots and basis functions, the weighted KL approach can offer better accuracy in resolving multiple time scales than the standard KL approach. In Figure 17.15, we compare the performance of weighted and standard KL techniques by fixing the number of basis functions (3U+3V+3P) and using different snapshots with each approach. The weighted KL method uses 22 snapshots to generate the basis functions (the weighting coefficients are again selected by the approach described in the previous paragraph), and the standard KL method uses 66 snapshots to generate basis functions. The result in Figure 17.15 indicates that the weighted KL technique offers better accuracy during the initial transient than the standard KL method, while both methods produce comparable accuracy at steady state. From this result, we can conclude that for a fixed number of basis functions, a weighted KL technique using fewer snapshots



(a) (b)

FIGURE 17.14. (a) Comparison of V-velocity at position $a - a'$ of Figure 17.12. 60 snapshots are used in both methods. (b) Comparison of V-velocity at position $a - a'$ of Figure 17.12. 20 snapshots and 3U+3V+3P basis functions are used in both reduced-order modeling techniques.

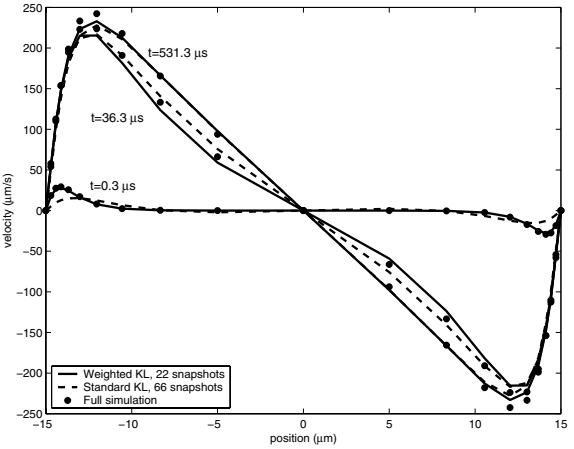


FIGURE 17.15. Comparison of V-velocity at position $a - a'$ of Figure 17.12. 3U+3V+3P basis functions are used in both reduced-order modeling techniques. 22 snapshots are used in the weighted KL method and 66 snapshots are used in the standard KL method to generate basis functions.

can produce better accuracy than a standard KL technique.

17.4.2 Nonlinear Galerkin Methods

A dynamical system can be represented by a differential equation of the form

$$\dot{\mathbf{v}} = \mathbf{G}(\mathbf{v}, \lambda), \tag{17.23}$$

where in the general case $\mathbf{v}$ is an element of a Hilbert space $\mathbf{E}$ and $\mathbf{G}(\mathbf{v}, \lambda)$ is a smooth nonlinear operator. We investigate the loss of stability of an equilibrium $\mathbf{v}_e$ of equation (17.23) under quasi-static variation of a distinguished system parameter λ. Equation (17.23) can be rewritten in the form

$$\dot{\mathbf{u}} = \mathbf{L}(\lambda)\mathbf{u} + \mathbf{g}(\mathbf{u}, \lambda), \tag{17.24}$$

where $\mathbf{L} = \mathbf{G}_v(\mathbf{v}_e)$ is the linearization of the operator $\mathbf{G}$ at $\mathbf{v}_e$, the equilibrium position; $\mathbf{g}$ is a smooth nonlinear operator, and $\mathbf{u} = \mathbf{v} - \mathbf{v}_e$ is the deviation from $\mathbf{v}_e$. From the point of stability we assume that equation (17.24) has an asymptotically stable equilibrium position $\mathbf{u}_e = 0$ for a range of parameter values λ. Now λ is varied quasi-statically, and it is assumed that for $\lambda = \lambda_c$ a loss of stability occurs at $\mathbf{u}_e = 0$. Then, two cases exist for which a proper dimension reduction can be performed ($\epsilon \ll 1$):

1. For $\mid \lambda - \lambda_c \mid = O(\epsilon)$, the center manifold theory applies.
2. For $\mid \lambda - \lambda_c \mid = O(1)$, the Galerkin methods apply.

According to the center manifold theory (Troger and Steindl, 1991), the field variable can be decomposed into a form:

$$\mathbf{u}(\mathbf{x}, t) = \mathbf{u}_c(\mathbf{x}, t) + \mathbf{u}_s(\mathbf{x}, t) = \sum_{i=1}^{n_c} q_i(t)\chi_i(\mathbf{x}) + \mathbf{U}(q_i(t), \mathbf{x}), \tag{17.25}$$

where $\chi_i(\mathbf{x})$ are the active spatial modes, obtained from the solution of the eigenvalue problem related to the linear system

$$\dot{\mathbf{u}} = \mathbf{L}(\lambda_c)\mathbf{u}.$$

Also, $q_i(t)$ are the time-dependent amplitudes, and $\mathbf{u}_s(\mathbf{x}, t)$ can be represented by an infinite sum. The key point is that the influence sum of the higher modes contained in $\mathbf{u}_s(\mathbf{x}, t)$ can be expressed in terms of the lower-order modes by the function $\mathbf{U}(q_i(t), \mathbf{x})$.

In applying the Galerkin methods, the field variable $\mathbf{u}(\mathbf{x}, t)$ is expressed in the form

$$\mathbf{u}(\mathbf{x}, t) = \sum_{j=1}^{m} q_j(t)\phi_j(\mathbf{x})$$

by a set of m comparison vectors $\phi_j(\mathbf{x})$ called the Galerkin basis, which satisfies the geometric and natural boundary conditions. Two subdivisions in the Galerkin methods exist, namely, the standard (linear) Galerkin method

and the nonlinear Galerkin method. In the linear Galerkin method one neglects $\mathbf{u}_s(\mathbf{x}, t)$ in equation (17.25). Therefore, the fast dynamics taken into account by the center manifold theory are completely ignored in the reduction process. Nonlinear Galerkin methods take into consideration the influence of higher modes and are also known by the name inertial manifolds in the mathematical literature. Two nonlinear Galerkin methods that have gained importance are (Steindl and Troger, 2001):

1. Approximate inertial manifold theory.
2. Postprocessed Galerkin method.

Application of Karhunen–Loève decomposition in nonlinear Galerkin methods has been presented in (Bangia et al., 1997). The dynamics of incompressible Navier–Stokes flow in a spatially periodic array of cylinders in a channel (for a mixing application) have been investigated using this method.

Approximate Inertial Manifold Theory

Equation (17.24) can be rewritten in the form (Steindl and Troger, 2001)

$$\dot{\mathbf{u}}_c = P\mathbf{L}\mathbf{u}_c + P\mathbf{g}(\mathbf{u}_c + \mathbf{u}_s), \tag{17.26}$$

$$\dot{\mathbf{u}}_s = Q\mathbf{L}\mathbf{u}_s + Q\mathbf{g}(\mathbf{u}_c + \mathbf{u}_s), \tag{17.27}$$

by decomposing $\mathbf{E} = \mathbf{E}_c \oplus \mathbf{E}_s$, where $\mathbf{E}_c$ is finite-dimensional and $\mathbf{E}_s$ is closed. This decomposition is achieved by defining the projection P onto $\mathbf{E}_c$ along $\mathbf{E}_s$, giving $\mathbf{u}_c = P\mathbf{u} \in \mathbf{E}_c$ and $\mathbf{u}_s = Q\mathbf{u} \in \mathbf{E}_s$, where $Q = I - P$ (see (Troger and Steindl, 1991), for details). In the standard Galerkin approximation of equation (17.24), from the eigenfunctions of $\mathbf{L}$, m modes are selected, equation (17.27) is completely ignored, and $\mathbf{u}_s$ is set to be zero in equation (17.26) to obtain

$$\dot{\mathbf{u}}_{ml} = P\mathbf{L}\mathbf{u}_{ml} + P\mathbf{g}(\mathbf{u}_{ml}),$$

where the index l denotes linear approximation. The influence of fast dynamics on the slow (essential) dynamics is completely ignored. Sometimes, a much better approximation is obtained if it is assumed that equation (17.24) has an inertial manifold of dimension m that can be realized as the graph of a function $\mathbf{h} : P\mathbf{E} \to Q\mathbf{E}$, or in other words, $\mathbf{u}_s = \mathbf{h}(\mathbf{u}_{mn})$. The projection of the inertial form onto $P\mathbf{E}$ is then given by

$$\dot{\mathbf{u}}_{mn} = P\mathbf{L}\mathbf{u}_{mn} + P\mathbf{g}(\mathbf{u}_{mn} + \mathbf{h}(\mathbf{u}_{mn})). \tag{17.28}$$

Now the approximation of $\mathbf{u}$ is given by $\mathbf{u}_{\text{approx}} = \mathbf{u}_{mn} + \mathbf{h}(\mathbf{u}_{mn})$, which is analogous to equation (17.25). In the actual process, first, one makes a standard Galerkin approximation using n nodes. Then the m-dimensional

approximation of the inertial manifold, $\mathbf{h}$, is calculated (see (Brown et al., 1990), for details) and used in equation (17.28) and in the expression for $\mathbf{u}_{\text{approx}}$. This method can capture nonlinear behavior better than standard Galerkin methods, but it involves extra cost, since the inertial manifold needs to be computed at every integration step.

Postprocessed Galerkin Method

This method is computationally more efficient than the approximate inertial manifold theory (Garcia-Archilla et al., 1998). In this method, first the standard Galerkin method is used, and at time (t) only when some output is required, the variables are approximated by the inertial manifold. That is, the solution $\mathbf{q}_m = \mathbf{q}_{ml}$ is calculated from

$$\dot{\mathbf{q}}_{ml} = P\mathbf{L}\mathbf{q}_{ml} + P\mathbf{g}(\mathbf{q}_{ml}),$$

which requires less effort than computing $\mathbf{q}_m = \mathbf{q}_{mn}$ from

$$\dot{\mathbf{q}}_{mn} = P\mathbf{L}\mathbf{q}_{mn} + P\mathbf{g}(\mathbf{q}_{mn} + \mathbf{h}_{\text{approx}}(\mathbf{q}_{mn})).$$

The final approximate solution for $\mathbf{u}$, $\mathbf{u}_{\text{approx}}$, is computed by $\mathbf{u}_{\text{approx}} = \mathbf{u}_{ml} + \mathbf{h}_{\text{approx}}(\mathbf{u}_{ml})$, where $\mathbf{h}_{\text{approx}}$ is the approximate inertial manifold. The computational cost is reduced greatly as a result of this simplification.

The concept of dynamic postprocessing has been introduced in (Margolin et al., 2003), for highly oscillatory systems. For a variety of systems, the normal postprocessed Galerkin method has been found to be a very efficient technique for improving the accuracy of ordinary Galerkin/nonlinear Galerkin methods with very little extra computational cost. The normal postprocessed Galerkin methods are based on truncation analysis using asymptotic (in time) estimates for the low and high mode components, which hold only when the solutions are on or near an attractor. As a result, these estimates may not hold for nonautonomous systems with highly oscillatory (in time) forcing, long transients, etc. Dynamic postprocessing can handle such situations by integration along transients (see (Margolin et al., 2003), for more details).