Computation speedup in the dynamic simulation of MEMS by macromodels

LIN Wuzhong[1], LIANG Yanchun[1,2*], LEE Kwok Hong[3], LIM Siak Piang[3] and LEE Heow Pueh[1]

(1. Institute of High Performance Computing, Singapore 117528; 2. College of Computer Science and Technology, Jilin University, Changchun 130012, China; 3. Department of Mechanical Engineering, National University of Singapore, Singapore 119260)

Received July 15, 2002; revised October 3, 2002

Abstract A doubly clamped microbeam actuated by electrostatic force with squeezed gas film damping is a well-known and standard micro-device in microelectromechanical system (MEMS) for many researchers to demonstrate how reduced-order dynamic macromodel is an effective way to faithfully capture the device behaviors. However it still takes time to directly recompute the time-dependant nonlinear terms in macromodels which are generated by a proper orthogonal decomposition (POD) method with Galerkin procedure at every time step during the macromodel simulation. This paper proposes two methods for speeding up the computation of macromodel simulations. In the first method, the computation speedup is achieved based on the concept of precomputation upon the basis functions are available. In the second method, cubic splines approximation is used to interpolate the basis functions and their first and second derivatives, and spatial integration is performed by application of the Gaussian quadrature. Numerical results show both methods could enhance the efficiency of the macromodel simulation compared with our previous computation results.

Keywords: MEMS, macromodels, model order reduction, proper orthogonal decomposition.

The modeling and simulation of the microelectromechanical system (MEMS) devices are usually presented by nonlinear partial differential equations (PDE) due to the multiple coupled energy domains involved in the MEMS device and the existence of inherent nonlinearity of electrostatic actuation forces and the geometric nonlinearities caused by a large deformation. In order to perform rapid design verification and optimization of MEMS device, it is essential to generate low-order dynamic models that permit fast simulation while capture most of the accuracy and flexibility of the fully meshed finite element methods (FEM) or finite difference methods (FDM) model simulations. These low-order models are called macromodels or reduced-order models.

In recent years, several approaches to generating the macromodels for dynamic simulation of the MEMS device have been presented, including lumped-parameter techniques[1], linear modal analysis techniques[2], Arnoldi-based model reduction approaches[3] and etc. More recently, the macromodels based on three proper orthogonal decomposition (POD) methods including singular value decomposition (SVD)[4], Karhunen-Loève decomposition (KLD)[5] and neural networks-based generalized Hebbian algorithm (GHA)[6] have been developed. It has been demonstrated in Refs. [4 ~ 6] how the macromodels were generated by extracting the global basis functions from a few fully meshed model runs in order to parameterize solutions with a far fewer degrees of freedom and how accurate and flexible the macromodel simulations are. It is however found in the real numerical experiments that much of computer time is spent on the recomputation of the time-dependant nonlinear terms at every time step during the numerical integration thus results in relatively low efficiency in these macromodels simulation. Two methods to speed up the computation of macromodel simulation are developed in this paper to further enhance the efficiency of macromodel simulations. In the first method, the computation speedup is achieved based on the precomputation concept, i. e. the nonlinear terms are firstly expressed explicitly in the time-dependant generalised modal coordinates and their coefficients are then precomputed prior to numerical time integration once the basis for macromodel is obtained. The second method is to use cubic splines approximation to interpolate the basis functions and their first and second derivatives and apply the Gaussian quadrature to scale down the spatial integration of the macromodel so as to improve the macromodel simulation efficiency. The numerical experiments have demonstrated that both methods could achieve successful computation efficiency enhancement of the

* To whom correspondence should be addressed. E-mail: linwz@ihpc.a-star.edu.sg

?1994-2018 China Academic Journal Electronic Publishing House. All rights reserved. http://www.cnki.net

macromodel dynamic simulation.

1 The system and governing equations

In order to demonstrate the model reduction techniques and the generation of the macromodel based on either the Karhunen-Loève decomposition (KLD) or the generalized Hebbian algorithm (GHA) neural network, a doubly clamped microbeam pulled in by the electrostatic actuation force with squeezed gas-film damping effect is examined. Fig. 1 shows a cross section of this device[7].

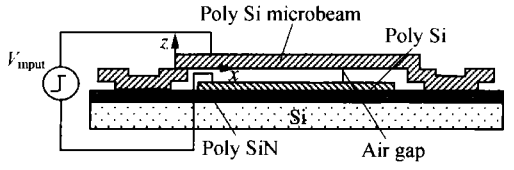

Fig. 1. A MEMS device.

When a voltage V is applied between the top and bottom electrodes, the top deformable microbeam is pulled downwards due to the electrostatic force. At the same time, the narrow air gap between the moving microbeam and the substrate will generate back pressure force on the microbeam due to the squeezed gas-film damping effect. The top microbeam will be pulled onto the bottom substrate when the applied voltage attains the pull-in voltage. The applied voltage is sensitive to the ambient pressure of the air, and thus this structure can be used as an accelerometer or a pressure sensor.

Device shown in Fig. 1 is a mechanical-electrostatic-fluidic coupled domain system. In general, the microbeam can be modeled as Bernoulli-Euler beam with electrostatic actuation force, and the back pressure force can be modeled by nonlinear Reynold's squeezed gas-film damping equation to yield the following nonlinear PDE[4]:

$$EI\frac{\partial^4 v}{\partial x^4} - T\frac{\partial^2 v}{\partial x^2} = -\frac{\varepsilon_0 b V^2}{2v^2} + \int_0^b (p - p_a)\mathrm{d}y - \rho\frac{\partial^2 v}{\partial t^2}, \quad (1)$$

$$\nabla\cdot(v^3 p\nabla p) = \frac{12\mu}{1+6K}\frac{\partial(pv)}{\partial t}, \quad (2)$$

where E is the Young's modulus; $I = bh^3/12$ is the moment of inertia in which b is the width and h the thickness of the microbeam; T is the residual stress; ρ is the density; μ is the air viscosity and equals 1.82×10^{-5} kg/(m·s); $v(x,t)$ is the height of the microbeam above the substrate; $K(x,t) = \lambda/v$ is the Knudson number in which λ is the mean-free path of the air and equals 0.064 μm; $-\varepsilon_0 bV^2/(2v^2)$ is the electrostatic actuation force where V is the applied voltage; ε_0 is the permittivity of free space and equals 8.854×10^{-12} Farad/m; $p(x,y,t)$ is the back pressure force caused by the squeezed gas-film in which an isothermal process is assumed; p_a is the ambient pressure and equals 1.013×10^5 Pa.

2 Model reduction techniques

There are techniques such as finite difference methods (FDM) or finite element methods (FEM) to convert continuous dynamic nonlinear systems with an infinite number of degrees of freedom to discrete finite dimensional models. But the resulting number of degrees of freedom is usually so large that it is extremely computationally intensive and time-consuming for practical use. Reduced order dynamic macromodel is a way to convert the dynamic nonlinear system to a model with a small number of degrees of freedom while captures all the essential behaviors of the original system efficiently and accurately. Two methods based on Karhunen-Loève decomposition and generalized Hebbian algorithm to obtain the lower order dynamic macromodels for rapid and accurate MEMS device simulation have been developed[4,5] recently. We briefly present the basic idea of the KLD and GHA neural network and then demonstrate their applications to the lower order macromodel generation for dynamic nonlinear systems simulation.

2.1 Model reduction based on Karhunen-Loève decomposition

The Karhunen-Loève decomposition (KLD) is a procedure for extracting an empirical basis for a modal decomposition from an ensemble of signals. Assume the signals are an ensemble of the functions $u_n(x)$ with $n = 1, 2, \cdots, N$. The objective to find a single deterministic function $\phi(x)$ that is the most similar to the members of $u_n(x)$ on average is to solve the following integral eigenvalue problem[8]

$$\int_\Omega K(x,x')\phi(x')\mathrm{d}x' = \lambda\phi(x). \quad (3)$$

We define two-point spatial correlation $K(x, x')$ as follows:

$$K(x,x') = E_n[u_n(x)u_n(x')], \quad (4)$$

where E_n is the expected value over n. Following the method of snapshots[9], we assume the eigenfunction

?1994-2018 China Academic Journal Electronic Publishing House. All rights reserved. http://www.cnki.net

is a linear combination of the snapshots $u_n(x)$,

$$\phi(x) = \sum_{n=1}^{N} a_n u_n(x). \tag{5}$$

Substituting (5) into (3) yields a matrix eigenvalue problem that determines a set of eigenvectors $\boldsymbol{\alpha}$ with their associated eigenvalues λ,

$$C\boldsymbol{\alpha} = \lambda\boldsymbol{\alpha},$$
$$c_{nk} = \frac{1}{N}(u_n, u_k) = \frac{1}{N}\int_{\Omega} u_n(x') u_k^T(x')\mathrm{d}x'. \tag{6}$$

Solving for Eq. (6) and substituting the eigenvector $\boldsymbol{\alpha}$ into Eq. (5), we can obtain a set of eigenfunctions $\phi_i(x)$, the order of these eigenfunctions $\phi_1(x), \phi_2(x), \cdots, \phi_N(x)$ corresponds to the order of the magnitude of the corresponding eigenvalues $\lambda_1 > \lambda_2 > \cdots > \lambda_N$. The symmetric and non-negative definiteness of $K(x, x')$ assures that $\lambda_i \geqslant 0$.

Finally, every member of the ensemble could be reproduced by a modal decomposition with respect to the eigenfunctions as basis

$$u_n(x) = \sum_{k=1}^{m} a_k \phi_k(x), \quad m \leqslant N, \tag{7}$$

where a_k are the generalised modal coordinates. Eq. (7) is called Karhunen-Loève decomposition. The efficiency of the KLD arises from the fact that it is optimal on average in the sense that the first few m eigenfunctions capture the most informative contents, or more energy, of the signals than the first m functions of any other basis. This energy captured by the first m eigenfunctions is

$$E_n[u_n(x)u_n(x)] = \sum_{k=1}^{m} E_n(a_k^2) = \sum_{k=1}^{m} \lambda_k. \tag{8}$$

We make use of this optimality feature of the KLD and the Galerkin procedure employing these eigenvectors as basis functions to generate the macromodel so as to represent the original continuous system with minimum number of degrees of freedom.

All the above optimality carries directly over to the discrete form of KLD that we will apply. This is because the application of KLD to structure analysis typically requires an ensemble of snapshots of structural dynamical characteristics at different locations. It is naturally a set of discrete data. For example, the deflections of the microbeam in Fig. 1 at M location sampled N times in a temporal space can be formed as an ensemble of snapshots,

$$\boldsymbol{v}(x, t) = [v(x_1, t), v(x_2, t), \cdots, v(x_M, t)]^T, \tag{9}$$

where each row of $\boldsymbol{v}$ represents the history array of the deflection of the microbeam in the co-ordinate space $v(x_i, t_s) = [v(x_i, t_1), v(x_i, t_2), \cdots, v(x_i, t_N)]$, $i = 1, 2, \cdots, M$. The $M \times M$ correlation matrix $\boldsymbol{K} = E[\boldsymbol{v}\boldsymbol{v}^T]$ can be then generated. With this discrete procedure for the continuous system in the coordinate space, the set of the eigenfunctions of Eq. (3) is obtained as a set of vector functions or eigenvectors of the following correlation matrix

$$(\boldsymbol{K} - \lambda\boldsymbol{I})\phi(x) = 0, \tag{10}$$

where $\boldsymbol{I}$ is the identity matrix. From now on, we will call the discrete form of eigenfunctions as eigenvectors $\phi_k(x)$.

2.2 Model reduction based on GHA neural network

The principal components are the most important linear features of the random observation vectors, and the purpose of principal component analysis (PCA) is to identify the dependent structure behind a multivariate stochastic observation in order to obtain a compact description of it. Through the PCA many variables can be represented by a few principal components, so the PCA can be considered as a feature extraction technique. Performing the PCA on a set of multivariate random data means computing the eigenvectors of its correlation matrix corresponding to the largest eigenvalues, and the projection of the data over the eigenvectors to obtain a number of principal components. Several neural network architectures and learning rules for performing the PCA have been proposed in the scientific literature. The well-known GHA by Sanger[10] to extract the principal eigenvectors of the correlation matrix from an ensemble of signals is adopted in this paper.

Let the inputs to a single-layer neural network be an n-dimensional column vector $\boldsymbol{x}$, the weights a $m \times n$ matrix $\boldsymbol{W}$, and the outputs an m-dimensional column vector $\boldsymbol{y} = \boldsymbol{W}\boldsymbol{x}$ with $m < n$. And assume that the values of x are generated by a stationary white random vector stochastic process with a correlation matrix $\boldsymbol{R} = E[\boldsymbol{x}\boldsymbol{x}^T]$. Therefore, $\boldsymbol{x}$ and $\boldsymbol{y}$ are both time-varying, and $\boldsymbol{W}$ will be time-varying as a result of adaptation through the training algorithm.

The GHA is given by

$$w_{ij}(t+1) = w_{ij}(t) + \eta(t)$$

?1994-2018 China Academic Journal Electronic Publishing House. All rights reserved. http://www.cnki.net

$$\cdot \left(y_i(t)x_j(t) - y_i(t)\sum_{k=1}^{i} w_{kj}(t)y_k(t) \right),$$
$$i = 1,2,\cdots,m; \quad j = 1,2,\cdots,n, \tag{11}$$

where w_{ij} is the element of the weight matrix $\boldsymbol{W}$, which is the connection strength between the jth input neuron and the ith output neuron, x_j is the jth component of the input vector $\boldsymbol{x}$, y_i the ith component of the output vector $\boldsymbol{y}$, and $\eta(t)$ a time-varying learning rate parameter which satisfies

$$\lim_{t\to\infty}\eta(t) = 0 \quad \text{and} \quad \sum_{t=0}^{\infty}\eta(t) = \infty. \tag{12}$$

The second term of GHA in Eq. (11) is the Hebbian term, and the third term ensures that the algorithm learns successive eigenvectors $\boldsymbol{q}_1(x)$, $\boldsymbol{q}_2(x)$, $\cdots$, $\boldsymbol{q}_N(x)$ of the correlation matrix $\boldsymbol{R}$ ordered by descending eigenvalues. Under condition (11), Sanger[10] proved a theorem stating that if the weight matrix $\boldsymbol{W}$ is assigned random values at time zero, then with probability 1, Eq. (11) will converge to a fixed point with $\boldsymbol{W}$ approaching a matrix whose rows are the first m eigenvectors of the $n \times n$ input correlation matrix $\boldsymbol{R}$ of the $n \times 1$ input vector, ordered by decreasing eigenvalue.

The significance of the theorem is that it guarantees the GHA to find the eigenvectors of the correlation matrix $\boldsymbol{R}$, equally important is the fact that there is no need to pre-compute the correlation matrix $\boldsymbol{R}$. Rather, the eigenvectors of $\boldsymbol{R}$ are computed by the algorithm directly from the input vector. This is an important feature that results in enormous computation saving, particularly when the number of inputs n is so large that computation and manipulation of $\boldsymbol{R}$ are not feasible and the required number of eigenvectors associated with the m largest eigenvalues of the correlation matrix $\boldsymbol{R}$ is a small fraction of n. The GHA takes advantage of this network structure. In the MEMS model reduction in this paper, for example the device shown in Fig. 1, the input vector is one snapshot of the deflection or pressure data at one temporal sampling. In general, the number of inputs is large but the number of required outputs is small, therefore, the GHA provides a practical and useful procedure for finding required few eigenvectors $\boldsymbol{q}(x)$.

3 Macromodel generation

In order to generate the macromodel for MEMS dynamic simulation using the above-mentioned KLD or GHA techniques, we need to obtain the ensemble of signals or snapshots from the numerical solution of the original nonlinear dynamic system first. The snapshots must be representative of the dynamic characteristics of the system or device under consideration. For the system shown in Fig. 1, the pull-in dynamics of the microbeam at a series of different times is simulated using FDM for an ensemble of applied step voltage to obtain the time-dependant microbeam deflection $v(x_i, t_s)$ and the back air pressure $p(x_i, y_j, t_s)$ ensembles as the ensemble of snapshots.

To simulate the pull-in dynamics of the microbeam using FDM, we discretize the Bernoulli-Euler beam equation (1) and Reynold equation (2) in space to generate a $(M+1)\times(N+1)$ mesh with $M \times N$ inner grids and $2M+2N+4$ boundary grids as shown in Fig. 2. The central difference is used to discretize the spatial partial derivation operators in Eqs. (1) and (2), and the trapezoidal rule is adopted to discretize the integral operator. The states of three unknowns $v(x,t)$, $\partial v(x,t)/\partial t$ and $p(x,y,t)$ are then projected onto each grid point. This discretization will transform Eqs. (1) and (2) into a set of $M \times N + 2M$ nonlinear ODE. We can use the following state space to represent the unknowns on the grids,

$$\dot{x} = \left(\frac{\partial v_1}{\partial t} \cdots \frac{\partial v_M}{\partial t} \frac{\partial^2 v_1}{\partial t^2} \cdots \frac{\partial^2 v_M}{\partial t^2} \frac{\partial p_{11}}{\partial t} \cdots \frac{\partial p_{MN}}{\partial t} \right)^T. \tag{13}$$

They are integrated numerically by using Runge-Kutta method with the following boundary conditions

$$v = v_0, \quad \frac{\partial v}{\partial x} = 0, \quad \frac{\partial p}{\partial n} = 0 \quad (\text{at } x = 0, l),$$
$$p = p_a, \quad (\text{at } y = 0, b), \tag{14}$$

and initial conditions

$$v = v_0, \quad \frac{\partial v}{\partial t} = 0, \quad p = p_a \quad (\text{at } t = 0). \tag{15}$$

After deflection and back pressure ensembles are

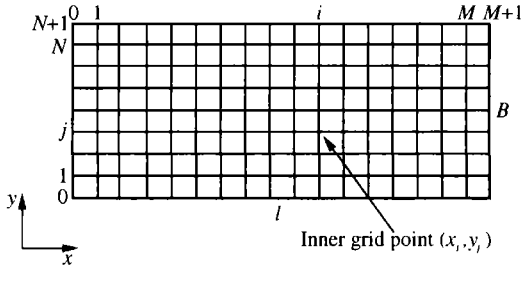

Fig. 2. Finite difference mesh of the microbeam.

?1994-2018 China Academic Journal Electronic Publishing House. All rights reserved. http://www.cnki.net

obtained, they are then used as the snapshots, i. e. the ensemble of signals for the Karhunen-Loève decomposition or as the inputs to the GHA neural network to generate the eigenvectors. The Galerkin procedure employing these eigenvectors as basis functions is then applied to the original nonlinear governing PDE (1) and (2) to convert them to the macromodel with a small number of ordinary differential equations (ODE). Because independent deflection and pressure basis functions make the Galerkin derivation simpler and also physically make sense of the problem, we perform the eigenvectors extraction by either KLD or GHA corresponding to deflection and back pressure, respectively.

We have proved that the eigenvectors extracted by either KLD or GHA for the same system and the same ensemble of signals are the same because both techniques are used to handle the same random vector and the same correlation matrix that is associated with the random vector, and both techniques have the same objective function for finding the optimal basis vector. The detailed derivation can be found in Ref. [11]. As such we denote both the eigenvectors extracted by KLD and GHA with respect to deflection as $\phi_i^v(x)$, and those with respect to the back pressure as $\phi_j^p(x, y)$, respectively.

The deflection $v(x, t)$ and pressure $p(x, y, t)$ can then be represented as a linear combination of the eigenvector as follows

$$v(x,t) - v_0 = \sum_{i=1}^{I} a_i^v(t)\phi_i^v(x) = \tilde{v}, \quad (16)$$

$$p(x,y,t) - p_a = \sum_{j=1}^{J} a_j^p(t)\phi_j^p(x,y) = \tilde{p}, \quad (17)$$

where v_0 is the initial gap between the deformable microbeam and the substrate, p_a the gap air ambient pressure, $a_i^v(t)$ and $a_j^p(t)$ are time-dependant generalised modal coordinates, I and J are the number of basis for deflection and back pressure respectively.

Substituting Eqs. (16) and (17) into the governing Eqs. (1) and (2), and applying the Galerkin procedure, we have

$$M_j \frac{d^2 a_j^v}{dt^2} + \sum_{i=1}^{I} K_{ji} a_i^v + F_j = 0, \quad (j = 1, 2, \cdots, I), \quad (18)$$

$$\sum_{i=1}^{J} H_{ji} \frac{da_i^p}{dt} + \sum_{i=1}^{J} S_{ji} a_i^p + C_j = 0, \quad (j = 1, 2, \cdots, J), \quad (19)$$

where

$$M_j = \int_L \rho(\phi_j^v)^2 dx, \quad (20)$$

$$K_{ji} = K_{ij} = \int_L \left(EI \frac{\partial^2 \phi_j^v}{\partial x^2} \frac{\partial^2 \phi_i^v}{\partial x^2} + T \frac{\partial \phi_j^v}{\partial x} \frac{\partial \phi_i^v}{\partial x} \right) dx, \quad (21)$$

$$F_j = \int_L \left(\frac{\varepsilon_0 b V^2}{2v^2} - \int_0^b (p - p_a) dy \right) \phi_j^v dx, \quad (22)$$

$$H_{ji} = H_{ij} = \int_A \frac{12\mu}{1 + 6K} v \phi_j^p \phi_i^p dx dy, \quad (23)$$

$$S_{ji} = S_{ij} = \int_A \left\{ v^3 p \left(\frac{\partial \phi_j^p}{\partial x} \frac{\partial \phi_i^p}{\partial x} + \frac{\partial \phi_j^p}{\partial y} \frac{\partial \phi_i^p}{\partial y} \right) + \frac{12\mu}{1 + 6K} \phi_j^p \phi_i^p \frac{\partial v}{\partial t} \right\} dx dy, \quad (24)$$

$$C_j = \int_A \frac{12\mu}{1 + 6K} p_a \phi_j^p \frac{\partial v}{\partial t} dx dy, \quad (25)$$

in which $\int_L$ indicates the integration along the length of the microbeam and $\int_A$ indicates the integration over the microbeam area.

The small set of coupled ODE Eqs. (18) and (19) constitutes the macromodel with global basis functions, which is the low-order dynamic simulation of the original nonlinear PDE system, Eqs. (1) and (2). Since this dynamic macromodel of ODE is generated by Galerkin procedure employing the basis functions extracted from the Karhunen-Loève decomposition or GHA network, the resulting degree of the freedom is usually very small, just a few basis functions compared to full model FEM/FDM simulation which contains a large number of degrees of freedom.

4 Macromodel simulation

The macromodel of Eqs. (18) and (19) is a set of ODE which could be integrated numerically in time by Runge-Kutta method to simulate the dynamics of the system with the applied voltage as an input. Examining the coefficients expressed in Eqs. (20) ~ (25) for the unknown modal coordinates $a_i^v(t)$ and $a_j^p(t)$ in Eqs. (18) and (19), we can see that some terms can be precalculated without difficulties once the basis functions are known, for example, M_j in (20) and K_{ji} in (21). However, many of the terms that associated directly to the time-dependant terms $v(x, t)$, $\partial v(x, t)/\partial t$ and $p(x, y, t)$ could not be precalculated so easily. Since $v(x, t)$, $\partial v(x, t)/\partial t$ and $p(x, y, t)$ are time dependant, the terms F_j,

?1994-2018 China Academic Journal Electronic Publishing House. All rights reserved. http://www.cnki.net

H_{ji}, S_{ji} and C_j in Eqs. (22)~(25) must be recalculated at every time step during the numerical time integration of macromodel. Also note that the computation of these terms is performed as spatial integration in the original Cartesian coordinate system.

The most direct method to perform this spatial integration numerically is to use the classical formulas for equally spaced abscissas, for example, the trapezoidal rule. Since we have obtained the basis functions known on every discrete grid point as shown in Fig. 2 and we have demonstrated in Refs. [5, 6] that by doing so the macromodel simulation can achieve around 11 times faster than FDM simulation when the device shown in Fig. 1 is applied with an input step voltage of 10.25 V. It is however expected that this direct numerical integration could not achieve the best time efficiency because the time-dependant terms in the macromodel must be recomputed at every time step and in the situation that the number of grid points for integration is very large. In order to speed up the macromodel simulation while keeping the accuracy, we have developed two methods to achieve the better time efficiency of the macromodel simulation. We will describe these two methods in detail and demonstrate their applications in the efficient and accurate simulation of the system.

4.1 Precomputation method

It is impractical to fully precompute Eqs. (22)~(25) since they are time-dependant. However, we could express these equations explicitly in the time-dependant generalised modal coordinates $a_i^v(t)$ and $a_j^p(t)$, and precompute their coefficients which are the known spatial integration after the basis functions are obtained to avoid repeatedly computing them at every time step during numerical time integration of macromodel so as to improve the macromodel simulation efficiency. We call this procedure precomputation method and the details are described below.

Once the number of the deflection and back pressure basis functions is chosen, H_{ji}, S_{ji} and C_j and in macromodel of Eqs. (23~25) can be explicitly expressed in time-dependant modal coordinates $a_i^v(t)$ and $a_j^p(t)$ by making use of multinomial theorem,

$$H_{ji} = h_{ji,0}^{(0)} + \sum_{k=1}^{I} h_{ji,k}^{(1)} a_k^v(t), \tag{26}$$

$$\begin{aligned} S_{ji} = {} & s_{ji,0}^{(0)} + \sum_{k=1}^{I} s_{ji,k}^{(1)} a_k^v(t) + \sum_{k=1}^{J} s_{ji,k}^{(2)} a_k^p(t) \\ & + \sum_{k=1}^{I}\sum_{l=1}^{J} s_{ji,kl}^{(3)}\, a_k^v(t) a_l^p(t) \\ & + \sum_{k_1+\cdots+k_I=2} s_{ji,k_1+\cdots+k_I}^{(4)} (a_1^v(t))^{k_1}\cdots(a_I^v(t))^{k_I} \\ & + \sum_{k_1+\cdots+k_I=2}\sum_{l}^{J} s_{ji,k_1+\cdots+k_I}^{(5)} (a_1^v(t))^{k_1}\cdots(a_I^v(t))^{k_I} a_l^p(t) \\ & + \sum_{k_1+\cdots+k_I=3} s_{ji,k_1+\cdots+k_I}^{(6)} (a_1^v(t))^{k_1}\cdots(a_I^v(t))^{k_I} \\ & + \sum_{k_1+\cdots+k_I=3}\sum_{l}^{J} s_{ji,k_1+\cdots+k_I}^{(7)} (a_1^v(t))^{k_1}\cdots(a_I^v(t))^{k_I} a_l^p(t) \\ & + \sum_{k=1}^{I} s_{ji,k}^{(8)} \frac{\mathrm{d}a_k^v(t)}{\mathrm{d}t}, \end{aligned} \tag{27}$$

$$C_j = \sum_{k}^{I} c_{j,k} \frac{\mathrm{d}a_k^v(t)}{\mathrm{d}t}, \tag{28}$$

where $h_{ji,k}^{(l)}$, $s_{ji,k}^{(l)}$ and $c_{j,k}$ are constants that are the known spatial integration once the basis functions are obtained. Special attention needs to be paid to the term related to electrostatic force $\varepsilon_0 bV^2/(2v^2)$ in F_j of (22) since the time-dependant term $v(x,t)$ is the denominator and it will also approach zero and hence cause the singularity at the pull-in. In order to express F_j explicitly in time-dependant modal coordinates $a_i^v(t)$ and $a_j^p(t)$, we propose a fourth order polynomial function to approximate $1/v^2$ with the microbeam deflection range of $\tilde{v}/v_0 \in (-0.7, 0.0)$,

$$\begin{aligned} \frac{1}{v^2} = \frac{1}{v_0^2(1+\tilde{v}/v_0)^2} \approx \frac{1}{v_0^2}[& p_4(\tilde{v}/v_0)^4 + p_3(\tilde{v}/v_0)^3 \\ & + p_2(\tilde{v}/v_0)^2 + p_1(\tilde{v}/v_0) + p_0], \end{aligned} \tag{29}$$

where $p_0 = 1.1095$, $p_1 = 2.6136$, $p_2 = 43.4482$, $p_3 = 117.8236$, $p_4 = 127.9959$.

Since the maximum stable deflection of the microbeam is about 67% of the original gap v_0 in static case[1], we assume the above lower order polynomial function for approximation of nonlinear term $1/v^2$ is sufficient for the deflection range of 0.0 to 0.7 (70% of the deflection). The trade off is that it could not be used to simulate the system near pull-in area where $\tilde{v}/v_0$ approaches -1. Using the multinomial theorem, we can then rewrite Eq. (22) as

$$\begin{aligned} F_j = {} & f_{j,0}^{(0)} + \sum_{k=1}^{4}\sum_{k_1+\cdots+k_I=k} f_{j,k_1+\cdots+k_I=k}^{(1)} (a_1^v(t))^{k_1}\cdots \\ & (a_I^v(t))^{k_I} + \sum_{l=1}^{J} f_{j,l}^{(2)} a_l^p(t). \end{aligned} \tag{30}$$

Similarly, the value of $f_j^{(k)}$ is also constant in the form of spatial integration that can be calculated once the basis functions are known.

?1994-2018 China Academic Journal Electronic Publishing House. All rights reserved. http://www.cnki.net

4.2 Cubic splines approximation and Gaussian quadrature

As one may expect, in general, Gaussian quadrature is simpler and more effective compared with the traditional quadrature formulas for equally spaced abscissas, e.g. trapezoidal rule. As such, one can improve the macromodel performance and efficiency by computing Eqs. (20)~(25) using following Gaussian-Legendre quadrature during the numerical time integration of the macromodel simulation,

$$M_j = \sum_{k=1}^{m} w_k \rho (\phi_j^v(x_k))^2, \tag{31}$$

$$K_{ji} = \sum_{k=1}^{m} w_k \left[EI \frac{\partial^2 \phi_j^v(x_k)}{\partial x^2} \frac{\partial^2 \phi_i^v(x_k)}{\partial x^2} + T \frac{\partial \phi_j^v(x_k)}{\partial x} \frac{\partial \phi_i^v(x_k)}{\partial x} \right], \tag{32}$$

$$F_j = \sum_{k=1}^{m} w_k \frac{\varepsilon_0 b V^2}{2v^2(x_k, t)} \phi_j^v(x_k) - \sum_{k=1}^{m} \sum_{l=1}^{n} w_k w_l [p(x_k, y_l, t) - p_a] \phi_j^v(x_k), \tag{33}$$

$$H_{ji} = \frac{12\mu}{1 + 6K} \sum_{k=1}^{m} \sum_{l=1}^{n} w_k w_l \cdot v(x_k, t) \phi_j^p(x_k, y_l) \phi_i^p(x_k, y_l), \tag{34}$$

$$S_{ji} = \sum_{k=1}^{m} \sum_{l=1}^{n} w_k w_l v^3(x_k, t) p_{kl}(x_k, y_l, t) \cdot \left[\frac{\partial \phi_j^p(x_k, y_l)}{\partial x} \frac{\partial \phi_i^p(x_k, y_l)}{\partial x} + \frac{\partial \phi_j^p(x_k, y_l)}{\partial y} \frac{\partial \phi_i^p(x_k, y_l)}{\partial y} \right] + \frac{12\mu}{1 + 6K} \sum_{k=1}^{m} \sum_{l=1}^{n} w_k w_l \cdot \phi_j^p(x_k, y_l) \phi_i^p(x_k, y_l) \frac{\partial v(x_k, t)}{\partial t}, \tag{35}$$

$$C_j = \frac{12\mu}{1 + 6K} p_a \sum_{k=1}^{m} \sum_{l=1}^{n} w_k w_l \phi_j^p(x_k, y_l) \frac{\partial v(x_k, y_l)}{\partial t}, \tag{36}$$

where m and n are the numbers of integration points, w_k and w_l are the associated weighting factors respectively.

In this method, once the discrete basis functions are obtained, the natural cubic splines approximation in which the second derivative on both boundaries of basis function is set to be zero, is used to interpolate the values of basis functions as well as their first and second-order derivatives at the kth Gaussian integration point. The Gaussian quadrature will then be used to perform the spatial integration of (31)~(36) during the macrodome simulation.

5 Numerical results

In order to validate the improved macromodel simulation efficiency enhanced by the above two speedup methods, we examine a simulated experience on a MEMS device as shown in Fig. 1 with physical and geometric parameters of $E = 149$ GPa, $T/(hb) = -3.7$ GPa, $\rho = 2330$ kg/m^3, $l = 610$ μm, $b = 40$ μm, $h = 2.2$ μm and $v_0 = 2.3$ μm. The snapshots are obtained from the solution of Eqs. (1) and (2) by using FDM for an ensemble of step voltages of $V_1 = 10$ V and $V_2 = 16$ V which are assumed to be the device operating voltage under consideration. Two sets of eigenvectors could then be obtained by application of the KLD or GHA neural network to these snapshots, respectively, and are employed as basis functions in the Galerkin procedure to the original nonlinear Eqs. (1) and (2) to generate the macromodel. The two sets of the eigenvector obtained independently by KLD and GHA are the same. As the example of the real numerical practice, Fig. 3 shows the same first two order eigenvectors for deflection obtained by KLD and GHA neural network, respectively while Figs. 4 and 5 plot the same first and second order eigenvectors for back pressure obtained from these two different methods. The number of the bases for deflection and back pressure chosen in the macromodel for system simulation can be determined by the system energy level captured by these eigenvectors (Eq. (8)). It has been shown in Ref. [5] that, for microbeam deflection simulation, the first eigenvector $\phi_1^v(x)$ can capture 99.99% of the system energy while it takes at least four first eigenvectors for the back pressure $\phi_i^p(x, y)$ to capture the same level in the back pressure simulation. As such, we choose on-

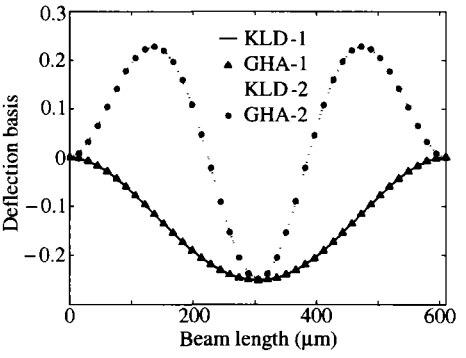

Fig. 3. Basis functions for deflection.

?1994-2018 China Academic Journal Electronic Publishing House. All rights reserved. http://www.cnki.net

ly one deflection bases and four back pressure basis in the macromodel simulation to ensure better system representation.

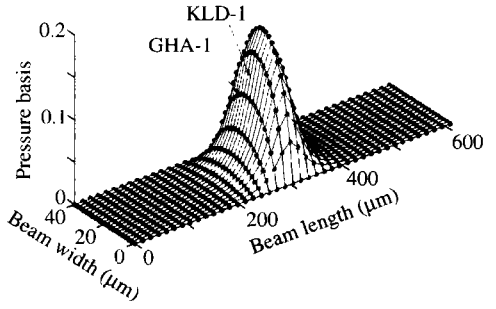

Fig. 4. The first basis functions for back pressure.

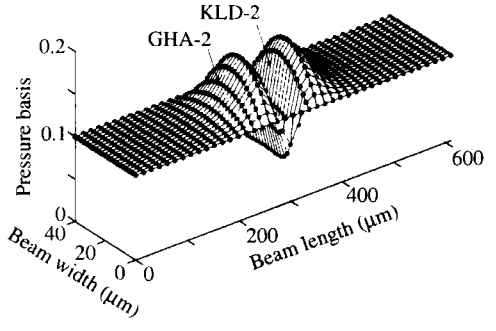

Fig. 5. The second basis functions for back pressure.

We define the mean square error between the numerical results of the deflection of center point of microbeam obtained by macromodel (MM) and by FDM as follows:

$$MSE = \frac{1}{N}\sum_{i=1}^{N}[v_{MM}(x_c, t_i) - v_{FDM}(x_c, t_i)]^2, \tag{37}$$

where x_c denotes the center point of the microbeam, t_i the sampled time instant, N the number of sampled time series, v_{MM} the simulation result by macromodel and v_{FDM} the FDM solution of the original PDE (1) and (2). Based on the numerical experiment, FDM solution with mesh size of 40×20 is able to generate sufficient accuracy.

Fig. 6 shows the comparison of the deflection of the center point of microbeam between the FDM solution of the original nonlinear PDE (1) and (2) and the macromodel simulations when the system is applied with a step input voltage of 8 V. Macromodel simulations are carried out by numerical integration of ODE Eqs. (18) and (19) with (i) the direct classical formulas for equally spaced abscissas, e.g. trapezoidal rule (TR), (ii) precomputation method (speedup 1) and (iii) cubic splines approximation and simplified Gaussian quadrature with 7×7 Gaussian integration grid size (speedup 2). It is noted that the minimum step pull-in voltage for this device is calculated at 8.87 V by FDM code so the pull-in does not happen when the input step voltage is less than its pull-in voltage. The performance of the computation comparison shown in Table 1 indicates that the speedup

Table 1. Performance comparison between macromodel and FDM simulations for an input step voltage of 8 V, period of time from 0 to 200 μs

Method for simulation	Number of ODE	Mean square error (10^{-4})	Computer system time (s)	Speedup factor
FDM	819	0	1048	1
Macromodel (TR)	6	2.15	145	7 (1)
Macromodel (Speedup 1)	6	2.85	1.7	616 (85)
Macromodel (Speedup 2)	6	3.95	27.3	38 (5)

factor of the macromodels simulation with speedup 1 and speedup 2 can achieve up to 85 and 5 times, respectively, compared with the macromodel simulation with TR method which has already achieved 7 times faster than FDM simulation, and all the macromodel simulation errors are small compared with FDM simulation. It is clear that when the applied input voltage is less than the pull-in voltage, macromodel simulation with speedup 1 or speedup 2 are very attractive as both methods are accurate and the computation effort required is much less than that by TR and FDM.

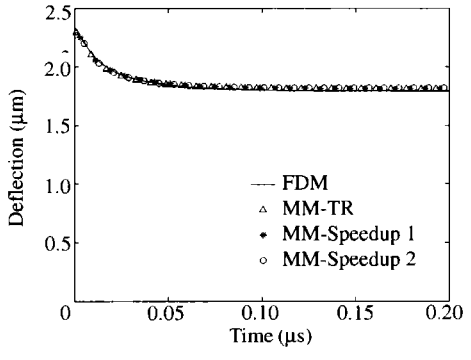

Fig. 6. Comparison of the microbeam dynamics for an input step voltage of $V = 8$ V.

Fig. 7 shows the comparison of the deflection of the center point of beam simulations when the system is applied with a step input voltage of 10.25 V which is larger than pull-in voltage. As expected, macromodel simulation with speedup 1 would not deliver

?1994-2018 China Academic Journal Electronic Publishing House. All rights reserved. http://www.cnki.net

accurate simulation results near pull-in area since the $1/v^2$ approximation does not count in the pull-in where v approaches zero. Table 2 shows that the

Table 2. Performance comparison between macromodel and FDM simulations for an input step voltage of 10.25 V, period of time from 0 till pull-in

Method for simulation	Number of ODE	Mean square error (10^{-4})	Computer system time (s)	Speedup factor
FDM	819	0	1952	1
Macromodel (TR)	6	2.27	182	11 (1)
Macromodel (Speedup 1)	6	27.06	1.4	1394 (130)
Macromodel (Speedup 2)	6	6.87	20.5	95 (9)

mean square error of macromodel simulation with speedup 2 is small compared with FDM but that of speedup 1 is large, and the computation efficiency is much better than FDM (95 times faster) and macromodel simulation with TR method (9 times faster). Thus when the input voltage is beyond pull-in voltage, macromodel with speedup 2 simulation is a desirable tool for system designer.

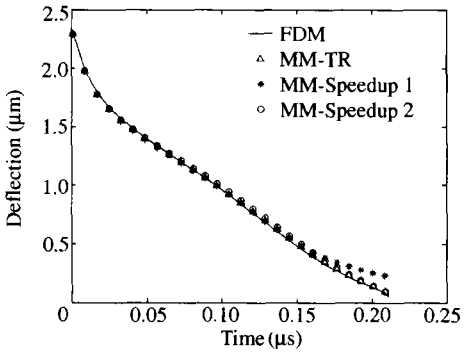

Fig. 7. Comparison of the microbeam dynamics for an input step voltage of $V = 10.25$ V.

6 Conclusions

Two methods to speed up the computation of macromodels which are generated by either the KLD or the GHA neural network-based together with Galerkin procedure are presented in this paper. It has demonstrated in numerical experiments that both methods can enhance the computation efficiency of macromodel simulation for MEMS device. Although the precomputation method does not deliver favorite results near pull-in, however, it can still give satisfying results for non pull-in dynamics analysis.

References

1 Tilmans, H. A. C. Equivalent circuit representation of electromechanical transducers I. Lumped-parameter systems (micromechanical systems). J. Micromech. Microeng., 1996, 6: 157.
2 Anathasuresh, G. K. et al. An approach to macromodelling of MEMS for nonlinear dynamic simulation, microelectromechanical systems (MEMS). ASME, DSC, 1996, 59: 401.
3 Wang, F. et al. Automatic model order reduction of a microdevice using the arnoldi approach, Microelectromechanical Systems (MEMS). ASME, DSC, 1998, 66: 527.
4 Hung, E.S. et al. Generating efficient dynamical models for microelectromechanical systems from a few finite-element simulation runs. J. of Microelectromechanical Systems, 1999, 8: 280.
5 Lin, W.Z. et al. A model reduction method for the dynamic analysis of microelectromechanical systems. Int. J. Nonlinear Sci., 2001, 2: 89.
6 Liang, Y.C. et al. A neural-network-based method of model reduction for the dynamic simulation of MEMS. J. Micromech. Microeng., 2001, 11: 226.
7 Gupta, R.K. et al. Pull-in time dynamics as a measure of absolute pressure. Proc. MEMS, 1997, 290~294.
8 Berkooz, G. et al. The proper orthogonal decomposition in the analysis of turbulent flows. Annu. Rev. Fluid Mech., 1993, 25: 539.
9 Sirovich, L. et al. Turbulent thermal convection in a finite domain: part I, II. Phys. Fluids A, 1990, 2: 1649.
10 Sanger, T.D. Optimal unsupervised learning in a single-layer linear feedforward neural network. Neural Netw, 1989, 2: 459.
11 Liang, Y.C. et al. Proper orthogonal decomposition and its applications, part I: theory. J. Sound Vibration, 2002, 252: 527.

?1994-2018 China Academic Journal Electronic Publishing House. All rights reserved. http://www.cnki.net