Computational Mechanics 23 (1999) 324–338 © Springer-Verlag 1999

A reproducing kernel particle method for meshless analysis of microelectromechanical systems

N. R. Aluru

Abstract Many existing computer-aided design systems for microelectromechanical systems require the generation of a three-dimensional mesh for computational analysis of the microdevice. Mesh generation requirements for microdevices are very complicated because of the presence of mixed-energy domains. Point methods or meshless methods do not require the generation of a mesh, and computational analysis can be performed by sprinkling points covering the domain of the microdevice. A corrected smooth particle hydrodynamics approach also referred to as the reproducing kernel particle method is developed here for microelectromechanical applications. A correction function that establishes the consistency and the stability of the meshless method is derived. A simple approach combining the constraint elimination and the Lagrange multiplier technique is developed for imposition of boundary conditions. Numerical results are shown for static and dynamic analysis of microswitches and electromechanical pressure sensors. The accuracy of the meshless method is established by comparing the numerical results obtained with meshless methods with previously reported experimental and numerical data.

1 Introduction

The present approach to modeling and design of microelectromechanical systems (MEMS) is to generate a geometric model for the complicated three-dimensional microdevice, to generate a mesh for the geometric model, to perform a self-consistent mesh-based numerical analysis of the mixed-energy domains governing the microdevice, and finally to perform post processing steps such as visualization. Computer-aided design (CAD) systems for MEMS based on such an approach were presented in ([29], [10], [8]). The time consuming steps in such an approach are to generate a three-dimensional mesh for the complicated microdevice and to perform self-consistent analysis of the mixed-energy domains governing the behavior of the device. Development of fast, efficient and reliable CAD systems for MEMS is more complicated when compared to the development of CAD systems for more traditional mechanical or electrical systems. This is because of the mixed-technology nature of microelectromechanical systems. To enable development of efficient CAD systems for MEMS, advances are needed to minimize the time spent on mesh generation and to perform rapid self-consistent analysis of mixed-energy domains.

Fast techniques for self-consistent analysis of mixed-energy domains is currently being pursued as a research area (see e.g. [28] for an overview). Mesh generation requirements for MEMS, are however, very demanding. For example, if we consider electromechanical systems involving coupled elastic and electrostatic energy domains, one needs to generate a volume mesh for the electromechanical microdevice to perform finite-element based elastic analysis and a surface mesh for the same microdevice to perform exterior electrostatic analysis based on accelerated boundary-element methods [1]. A requirement is that the surface mesh has to be compatible with the volume mesh so one does not have to worry about interpolating solutions from one mesh to another mesh. When a microfluidic energy domain is also involved such as in electrofluidicmechanical systems, then three different types of meshes are required. The complexity of mesh generation grows significantly when more than one energy-domain is involved and microelectromechanical system designs often involved at least two energy domains.

An efficient approach to MEMS modeling and design is to consider point-based methods or what are also referred to as meshless methods (see [4] for an overview and developments over the last five years). In these techniques one does not have to generate a mesh and a self-consistent numerical analysis of mixed or coupled energy domains can be carried out by just sprinkling points covering the domain of interest. Point methods have several advantages over traditional mesh based methods: the complicated mesh generation approach is replaced by a simpler point generation approach; for coupled domain analysis one does not have to worry about compatibility of meshes – in the electromechanical example discussed above the same points on the surface of the microdevice can be used for both elastic and electrostatic analysis; finally mesh adaptivity is an important step for generation of highly accurate solutions and with point based methods mesh adaptivity is replaced by point adaptivity, which is a lot simpler to implement.

N.R. Aluru
Beckman Institute and Department of General Engineering, University of Illinois at Urbana-Champaign, Urbana, IL 61801
e-mail: aluru@uiuc.edu, http://www.staff.uiuc.edu/~aluru/

This work is supported by a research initiation grant from the UIUC Research Board, and by a grant from the DARPA Composite CAD Program, Air Force Research Laboratory, Air Force Material Command under agreement number F30602-98-2-0178.

Several meshless techniques have been proposed recently and these are briefly discussed in Sect. 2. The meshless approach developed in this paper is based on the corrected smooth particle hydrodynamics approach. Smooth particle hydrodynamic approaches [22] are shown to be unstable for finite-domain boundaries and the stability of the approach is established by introducing a correction function. The approach involving a correction function is referred to as a reproducing kernel particle method [15]. The reproducing kernel particle method is extended in this paper for a fourth-order partial differential equation encountered in electromechanical switches and pressure sensors. The reproducing kernel particle method for static and dynamic analysis of electromechanical switches is discussed in detail in Sect. 3. Treatment of boundary conditions in meshless methods has been considered to be a major bottleneck and a simple approach which enforces the boundary conditions exactly is described in Sect. 3. Numerical results are presented in Sect. 4 and conclusions are presented in Sect. 5.

2
Meshless techniques

The origin of meshless methods can be traced to the development of smooth particle hydrodynamics (SPH) approach about two decades ago [19]. Further developments and overview articles on smooth particle hydrodynamics can be found in ([21], [22]). Liu and his co-workers have shown that the smooth particle hydrodynamics approach does not satisfy the consistency conditions ([13]–[17], [6]), and hence is not a stable method. The stability of SPH is investigated and discussed in detail in Sect. 2.3. The consistency of SPH can be established by introducing a correction function. This concept was pioneered by Liu in his work ([13]–[17], [6]) and the new approach is referred to as the reproducing kernel particle method. Other meshless methods developed over the last few years include the diffuse element method [25], element free Galerkin method ([5], [18], [30]), boundary node method [23], local boundary integral equations method ([32], [33]), a meshless method based on the local symmetric weak form and the moving least squares approximation [3], finite point method [26], cloud-based methods [9], and the partition of unity method [20].

The meshless method developed in this paper is based on the reproducing kernel particle method. Reproducing kernel particle methods have been applied for several problems in mechanics ([13]–[17], [6]), and in this paper we extend this approach to higher order partial differential equations encountered in MEMS. Specifically, a correction function that ensures consistency for higher order derivatives is investigated. Implementation of boundary conditions in meshless methods has been considered to be a major challenge. A discussion on the implementation of boundary conditions in meshless methods can be found in ([4], [6], [12], [15], [24], [31]). A popular approach has been to enforce boundary conditions through Lagrange multipliers ([5]). In this paper, boundary conditions are implemented through a combination of Lagrange multipliers and constraint elimination. Displacement, which is a primary variable in the applications considered in this paper is implemented through constraint elimination and the gradient of the displacement (i.e. the slope) which is not a primary variable is implemented through Lagrange multipliers.

2.1
Smooth particle hydrodynamics

Smooth particle hydrodynamics (SPH) approach was primarily developed to model problems encountered in astrophysics [19]. In the absence of boundaries, this technique has been shown to be very efficient to compute the motion of fluid masses or particles in three dimensions ([21], [22]). This technique, however, suffers from instabilities when applied to the solution of partial differential equations involving finite domains or boundaries. SPH can be simply understood as a kernel estimate technique. If we denote $u(x)$ to be any unknown function, then an approximation $u^a(x)$ to $u(x)$ in a domain Ω is generated by a kernel approximation i.e.

$$u^a(x) = \int_\Omega w_{\mathrm{d}}(x-s)u(s)\,\mathrm{d}s \tag{1}$$

where $w_{\mathrm{d}}(x-s)$ is the kernel or a weighting function, and d is the dilation parameter. The kernel function is nonzero $(w_{\mathrm{d}}(x-s) > 0)$ in a subdomain Ω_I of Ω and zero outside the subdomain Ω_I. The subdomain Ω_I is determined by the dilation parameter d. If the kernel function is the delta function, then $u^a(x) \to u(x)$ and the function is said to be reproduced exactly. Assuming that the domain Ω is finite and is represented by *NP* points or particles, a discrete approximation to Eq. (1) can be written as

$$u^a(x) = \sum_{I=1}^{NP} w(x-x_I)u(x_I)\Delta V_I \tag{2}$$

where $u(x_I)$ is defined as the nodal value at particle position x_I and ΔV_I is a measure of the domain surrounding node I. To introduce the concept of an SPH shape function, Eq. (2) can be rewritten as

$$u^a(x) = \sum_{I=1}^{NP} \Psi_I(x)u_I \tag{3}$$

$$\Psi_I(x) = w(x-x_I)\Delta V_I \tag{4}$$

where $\Psi_I(x)$ is referred to as the SPH shape function and $u_I = u(x_I)$. Note that in general $u^a(x_I) \neq u(x_I)$.

The discrete form of SPH given in (3) can be employed for the solution of partial differential equations describing various physical phenomena. For a method to be convergent, it has to be stable and consistent. The numerical solutions with a non-convergent method can exhibit amplitude and phase errors and a deterioration in the accuracy of the solution. It can be shown that the discrete form of SPH given in Eq. (2) is not consistent. An example illustrating the instabilities arising from the SPH discretization is provided in Sect. 2.3.

2.2
Reproducing kernel method

The consistency of SPH can be established by correcting the approximation introduced in Eq. (1). The concept of a

correction function was introduced by Liu in [16] and a corrected kernel approximation of Eq. (1) is written as

$$u^a(x) = \int_\Omega \bar{w}_\mathrm{d}(x-s)u(s)\,\mathrm{d}s \tag{5}$$

where $\bar{w}_\mathrm{d}(x-s)$ is the corrected kernel function which is given by

$$\bar{w}_\mathrm{d}(x-s) = C(x,s)w_\mathrm{d}(x-s) \tag{6}$$

where $C(x,s)$ is the correction function and $w_\mathrm{d}(x-s)$ is the kernel function introduced in Eq. (1).

The correction function is expressed as a linear combination of polynomial basis functions in the following form:

$$C(x,s) = c_0(x) + c_1(x)(x-s) + c_2(x)(x-s)^2 + \cdots + c_N(x)(x-s)^N \tag{7}$$

where $c_0,\ c_1,\ c_2,\ \ldots,\ c_N$ are functions of x which need to be determined. The number of c_i's involved in the definition of the correction function depends on the order of the highest derivative terms present in the governing partial differential equations. This issue is examined in more detail in Sect. 3, when the governing equations for microswitches are discussed. The unknown coefficients, $c_0,\ c_1,\ c_2,\ \ldots,\ c_N$, are determined by satisfying the reproducing conditions. Reproducing conditions are defined as those where the approximated function or the derivatives of the approximated function are reproduced exactly. A reproducing condition would, for example, enforce that the approximated function $u^a(x)$ in Eq. (5) be equal to the unknown function $u(x)$. Reproducing conditions can be derived by considering the Taylor series expansion of $u(s)$ i.e.

$$u(s) = u(x) + (s-x)u'(x) + \frac{(s-x)^2}{2!}u''(x) + \frac{(s-x)^3}{3!}u'''(x) + \cdots \tag{8}$$

where a prime denotes a differentiation. Substituting Eq. (8) in Eq. (5), we obtain

$$u^a(x) = u(x)\int_\Omega \bar{w}_\mathrm{d}(x-s)\,\mathrm{d}s + u'(x)\int_\Omega (s-x)\bar{w}_\mathrm{d}(x-s)\,\mathrm{d}s + u''(x)\int_\Omega \frac{(s-x)^2}{2!}\bar{w}_\mathrm{d}(x-s)\,\mathrm{d}s + \cdots \tag{9}$$

Denoting the moment of the corrected kernel function as

$$\bar{m}_k(x) = \int_\Omega (x-s)^k \bar{w}_\mathrm{d}(x-s)\,\mathrm{d}s \quad k = 0,1,2,\ldots \tag{10}$$

the approximated solution through the corrected kernel function given in Eq. (9) can be rewritten as

$$u^a(x) = u(x)\bar{m}_0(x) - u'(x)\bar{m}_1(x) + \frac{u''(x)}{2!}\bar{m}_2(x) - \frac{u'''(x)}{3!}\bar{m}_3(x) + \cdots \tag{11}$$

For $u^a(x) = u(x)$, the reproducing conditions can be written as

$$\begin{aligned} &\bar{m}_0(x) = 1 \\ &\bar{m}_k(x) = 0 \quad k = 1,2,\ldots \end{aligned} \tag{12}$$

Substituting the definitions for the corrected kernel function and the correction function given in Eqs. (6) and (7), respectively, the moment of the corrected kernel function can be written as

$$\bar{m}_k(x) = c_0(x)m_k(x) + c_1(x)m_{k+1}(x) + c_2(x)m_{k+2}(x) + \cdots + c_N(x)m_{k+N}(x) \tag{13}$$

where

$$m_k(x) = \int_\Omega (x-s)^k w_\mathrm{d}(x-s)\,\mathrm{d}s \tag{14}$$

Note that the definition of moment given in Eq. (14) involves only the kernel function and not the corrected kernel function. Assuming that all the N-terms are required in the correction function definition, the unknowns $c_0, c_1, c_2, \ldots, c_N$ can be determined as

$$\begin{bmatrix} m_0(x) & m_1(x) & \cdots & m_N(x) \\ m_1(x) & m_2(x) & \cdots & m_{N+1}(x) \\ \vdots & \vdots & & \vdots \\ m_N(x) & m_{N+1}(x) & \cdots & m_{2N}(x) \end{bmatrix} \begin{Bmatrix} c_0(x) \\ c_1(x) \\ \vdots \\ c_N(x) \end{Bmatrix} = \begin{Bmatrix} \bar{m}_0(x) \\ \bar{m}_1(x) \\ \vdots \\ \bar{m}_N(x) \end{Bmatrix} \tag{15}$$

or

$$\mathbf{M}\bar{\mathbf{C}} = \bar{\mathbf{M}} \tag{16}$$

where $\bar{\mathbf{M}} = [\bar{m}_0(x), \bar{m}_1(x), \ldots, \bar{m}_N(x)]^\mathrm{T} = [1,0,\ldots,0]^\mathrm{T}$ is the $(N+1)\times 1$ known right-hand side vector, $\mathbf{M}$ is the $(N+1)\times(N+1)$ known coefficient matrix, and $\bar{\mathbf{C}} = [c_0(x),\ldots,c_N(x)]^\mathrm{T}$ is the $(N+1)\times 1$ vector of unknown correction coefficients.

2.2.1 Reproducing conditions for the first derivative

Reproducing conditions for the first derivative of an approximated function can be derived in a manner similar to the procedure outlined above. From Eq. (5), the derivative of an approximate function can be written as

$$\frac{\mathrm{d}}{\mathrm{d}x}u^a(x) = \int_\Omega \frac{\mathrm{d}}{\mathrm{d}x}[\bar{w}_\mathrm{d}(x-s)]u(s)\,\mathrm{d}s = \int_\Omega \bar{w}'_\mathrm{d}(x-s)u(s)\,\mathrm{d}s \tag{17}$$

Substituting the Taylor series expansion for $u(s)$, the derivative of an approximate function can be written as

$$[u^a(x)]' = u(x)\bar{m}_0'(x) - u'(x)\bar{m}_1'(x) + \frac{u''(x)}{2!}\bar{m}_2'(x) - \frac{u'''(x)}{3!}\bar{m}_3'(x) + \cdots \quad (18)$$

where

$$\bar{m}_k'(x) = \int_\Omega (x-s)^k \bar{w}_\mathrm{d}'(x-s)\,\mathrm{d}s \quad k = 0, 1, 2, \ldots \quad (19)$$

The reproducing conditions to satisfy $[u^a(x)]' = u'(x)$ are summarized as

$$\begin{aligned} &\bar{m}_0'(x) = 0 \\ &\bar{m}_1'(x) = -1 \\ &\bar{m}_k'(x) = 0 \qquad k = 2, 3, \ldots \end{aligned} \quad (20)$$

The gradients of the correction function coefficients, denoted by $c_0'(x), \ldots, c_N'(x)$ are related to the moments by the expression

$$[\mathbf{M}'\mathbf{M}]\begin{bmatrix} \bar{\mathbf{C}} \\ \bar{\mathbf{C}}' \end{bmatrix} = [0] \quad (21)$$

where $\mathbf{M}$ is as defined in Eq. (15), and $\mathbf{M}'$ is defined as

$$\mathbf{M}' = \begin{bmatrix} m_0'(x) & m_1'(x) & \ldots & m_N'(x) \\ m_1'(x) & m_2'(x) & \ldots & m_{N+1}'(x) \\ \vdots & \vdots & & \vdots \\ m_N'(x) & m_{N+1}'(x) & \ldots & m_{2N}'(x) \end{bmatrix} \quad (22)$$

and $\bar{\mathbf{C}}' = [c_0'(x), \ldots, c_N'(x)]^\mathrm{T}$ is the vector comprising the gradients of the correction function coefficients. From Eq. (21) the gradients of the correction function coefficients are computed as

$$\bar{\mathbf{C}}' = -\mathbf{M}^{-1}\mathbf{M}'\bar{\mathbf{C}} \quad (23)$$

2.2.2
Reproducing conditions for the second derivative

Reproducing conditions for the second derivative of an approximated function can also be derived in a manner similar. From Eq. (5), the second derivative of an approximate function can be written as

$$\frac{\mathrm{d}^2}{\mathrm{d}x^2}(u^a(x)) = \int_\Omega \frac{\mathrm{d}^2}{\mathrm{d}x^2}[\bar{w}_\mathrm{d}(x-s)]u(s)\,\mathrm{d}s = \int_\Omega \bar{w}_\mathrm{d}''(x-s)u(s)\,\mathrm{d}s \quad (24)$$

Substituting the Taylor series expansion for $u(s)$, the second derivative of an approximate function can be written as

$$[u^a(x)]'' = u(x)\bar{m}_0''(x) - u'(x)\bar{m}_1''(x) + \frac{u''(x)}{2!}\bar{m}_2''(x) - \frac{u'''(x)}{3!}\bar{m}_3''(x) + \cdots \quad (25)$$

where

$$\bar{m}_k''(x) = \int_\Omega (x-s)^k \bar{w}_\mathrm{d}''(x-s)\,\mathrm{d}s \quad k = 0, 1, 2, \ldots \quad (26)$$

The reproducing conditions to satisfy $[u^a(x)]'' = u''(x)$ are summarized as

$$\begin{aligned} &\bar{m}_0''(x) = 0 \\ &\bar{m}_1''(x) = 0 \\ &\bar{m}_2''(x) = 2 \\ &\bar{m}_k''(x) = 0 \quad k = 3, 4, \ldots \end{aligned} \quad (27)$$

The gradients of the correction function coefficients, denoted by $c_0''(x), \ldots, c_N''(x)$ are related to the moments by the expression

$$[\mathbf{M}''\ 2\mathbf{M}'\ \mathbf{M}]\begin{bmatrix} \bar{\mathbf{C}} \\ \bar{\mathbf{C}}' \\ \bar{\mathbf{C}}'' \end{bmatrix} = [0] \quad (28)$$

where $\mathbf{M}''$ is defined as

$$\mathbf{M}'' = \begin{bmatrix} m_0''(x) & m_1''(x) & \ldots & m_N''(x) \\ m_1''(x) & m_2''(x) & \ldots & m_{N+1}''(x) \\ \vdots & \vdots & & \vdots \\ m_N''(x) & m_{N+1}''(x) & \ldots & m_{2N}''(x) \end{bmatrix} \quad (29)$$

and $\bar{\mathbf{C}}'' = [c_0''(x), \ldots, c_N''(x)]^\mathrm{T}$ is the vector comprising the gradients of the correction function coefficients. From Eq. (28), the second derivatives of the correction function coefficients are computed as

$$\bar{\mathbf{C}}'' = -\mathbf{M}^{-1}(\mathbf{M}''\bar{\mathbf{C}} + 2\mathbf{M}'\bar{\mathbf{C}}') \quad (30)$$

2.2.3
Reproducing kernel particle method

The discretized form of the reproducing kernel method, which is referred to as the reproducing kernel particle method (RKPM), can be written from Eq. (5) as

$$u^a(x) = \sum_{I=1}^{NP} \bar{w}_\mathrm{d}(x - x_1)u(x_1)\Delta V_I \quad (31)$$

Even though the summation in Eq. (31) is written over all the points in the domain, it is effectively over the points x_I where $\bar{w}_\mathrm{d}(x - x_I) > 0$. Equation (31) can be rewritten as

$$u^a(x) = \sum_{I=1}^{NP} N_I(x)u_I \quad (32)$$

$$N_I(x) = C(x - x_1)w_\mathrm{d}(x - x_1)\Delta V_I \quad (33)$$

where $u_I = u(x_I)$ is the value of the function u at node x_I, and N_I is defined as the RKPM shape function for node I.

2.3
Failure of SPH for finite domains

As mentioned earlier, SPH in its original form is unstable for partial differential equations posed over finite domains. The stability of SPH is established by satisfying the consistency conditions or the reproducing conditions. A comparison between the SPH and the RKPM shape

functions and the solutions obtained with both methods are presented for a simpler one-dimensional Laplacian equation. The problem considered here is the meshless solution of

$$\frac{\partial^2 u}{\partial x^2} = 0 \tag{34}$$

posed over the domain $x \in (0,1)$ with boundary conditions $u(0) = 0$ and $u(1) = 1$. The exact solution is the linear function $u(x) = x$. A cubic spline kernel function is employed for both the SPH and RKPM analysis. The kernel function $w_d(x - x_I)$ is defined as

$$w_d(x - x_I) = \frac{1}{d} w\left(\frac{x - x_I}{d}\right) \tag{35}$$

Denoting $y = (x - x_I)/d$, the cubic spline kernel function $w(y)$ is defined as

$$w(y) = \begin{cases} 0 & y < -2 \\ \frac{1}{6}(y+2)^3 & -2 \le y \le -1 \\ \frac{2}{3} - y^2\left(1 + \frac{y}{2}\right) & -1 \le y \le 0 \\ \frac{2}{3} - y^2\left(1 - \frac{y}{2}\right) & 0 \le y \le 1 \\ -\frac{1}{6}(y-2)^3 & 1 \le y \le 2 \\ 0 & y > 2 \end{cases} \tag{36}$$

The correction function for RKPM analysis is taken to be

$$C(x,s) = c_0(x) + c_1(x)(x - s) \tag{37}$$

and the coefficients c_0 and c_1 are determined by satisfying the reproducing conditions. A total of 101 particles are employed for numerical analysis. The SPH and the RKPM shape functions for the particles at the two boundaries and a particle in the interior are shown in Figs. 1 and 2, respectively. The correction function coefficients and their derivatives employed in the RKPM analysis are shown in Figs. 3 and 4, respectively. The solution obtained with SPH and RKPM techniques is compared in Fig. 5. The SPH solution is unstable and exhibits spurious oscillations. The

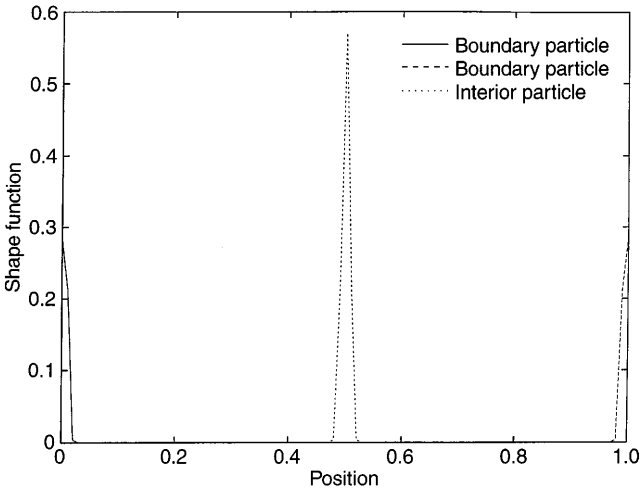


Fig. 1. SPH shape functions for particles at the two boundaries and for a particle in the interior of the domain

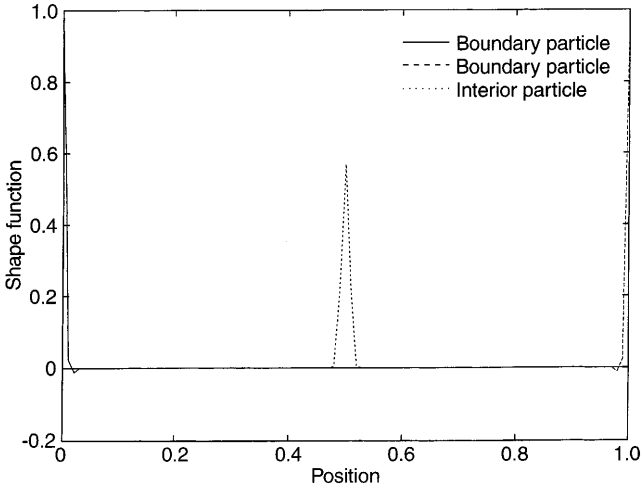


Fig. 2. RKPM shape functions for particles at the two boundaries and for a particle in the interior of the domain

instability of the SPH solution can be attributed to the lack of consistency of the SPH shape functions, especially near the boundaries. From the plots of the correction function coefficients shown in Fig. 3, it can be observed that the coefficients are constant in the interior of the domain and

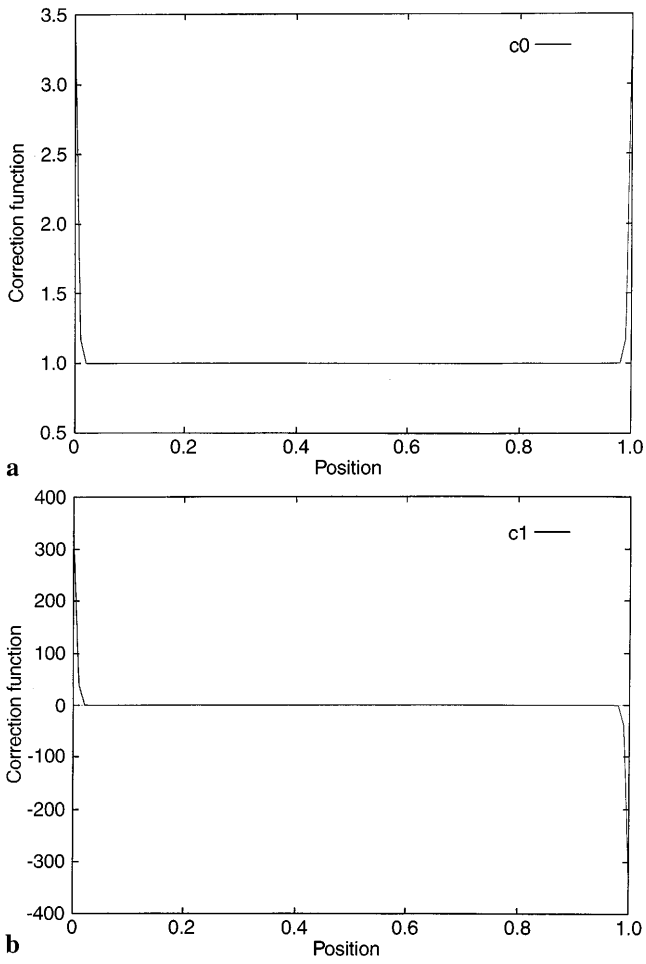
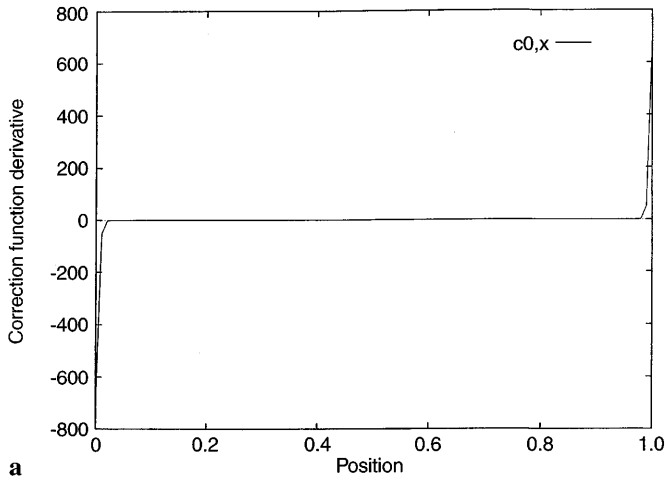


Fig. 3a, b. Correction function coefficients for RKPM analysis **a** coefficient c_0, **b** coefficient c_1



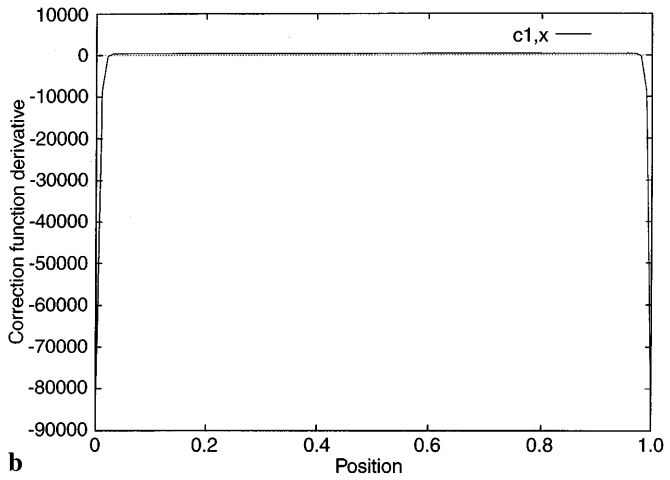
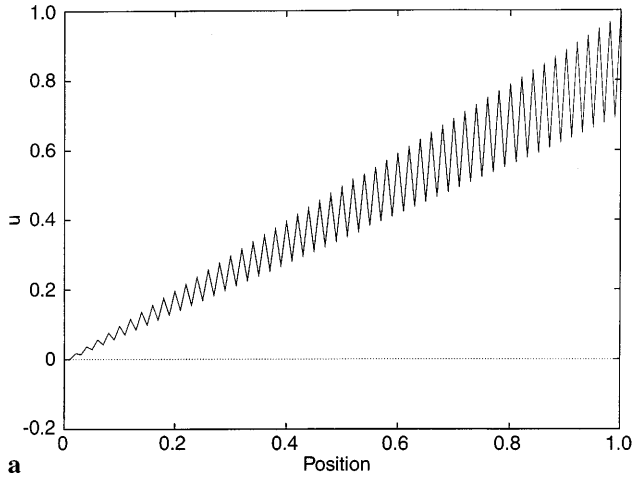


Fig. 4a, b. Derivatives of correction function coefficients for RKPM analysis **a** $c_{0,x}$, **b** $c_{1,x}$



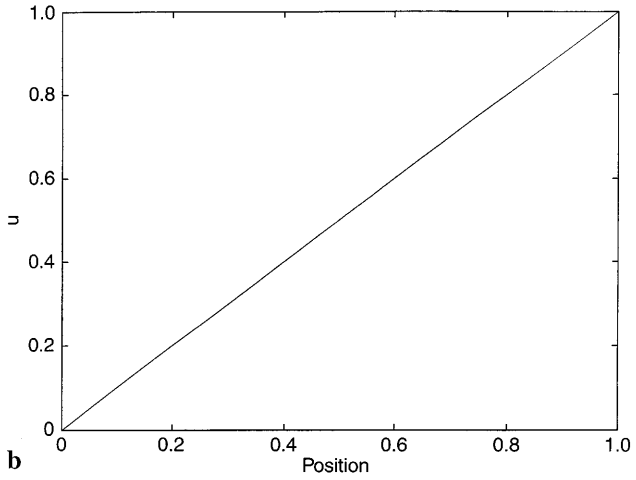


Fig. 5a, b. Solution obtained with **a** SPH, **b** RKPM

vary rapidly only near the boundaries. It can be concluded that the variation of the correction function coefficients near the boundaries provides the stability of the RKPM technique.

2.4
Reproducing kernel particle method in two-dimensions

The reproducing kernel particle method extends in a straight-forward manner for multi-dimensional applications. In two-dimensions, a corrected kernel approximation can be written as

$$u^a(x,y) = \int_\Omega \bar{w}_\mathrm{d}(x-s, y-s)u(s)\,\mathrm{d}s \tag{38}$$

where $\bar{w}_\mathrm{d}(x-s, y-s)$ is the two-dimensional corrected kernel function which is given by

$$\bar{w}_\mathrm{d}(x-s, y-s) = C(x,y,s)w_\mathrm{d}(x-s, y-s) \tag{39}$$

where $C(x,y,s)$ is called the two-dimensional correction function and $w_\mathrm{d}(x-s, y-s)$ is the two-dimensional kernel function.

The correction function in two-dimensions to exactly reproduce the second derivatives is given as:

$$\begin{aligned} C(x,y,s) = {} & c_0(x,y) + c_1(x,y)(x-s) + c_2(x,y)(y-s) \\ & + c_3(x,y)(x-s)^2 + c_4(x,y)(y-s)^2 \\ & + c_5(x,y)(x-s)(y-s) \end{aligned} \tag{40}$$

Note that if only the first derivative calculation is required, then only the first three coefficients need to be considered in the correction function definition. The correction function coefficients $c_0(x,y), c_1(x,y), \ldots, c_5(x,y)$ can be determined by establishing the reproducing conditions. Consider a Taylor series expansion for $u(s)$ in two-dimensions

$$\begin{aligned} u(s) = {} & u(x,y) + (x-s)\frac{\partial u}{\partial x} + (y-s)\frac{\partial u}{\partial y} \\ & + \frac{(x-s)^2}{2!}\frac{\partial^2 u}{\partial x^2} + \frac{(y-s)^2}{2!}\frac{\partial^2 u}{\partial y^2} \\ & + (x-s)(y-s)\frac{\partial^2 u}{\partial x \partial y} + \cdots \end{aligned} \tag{41}$$

Substituting the definition for $u(s)$ in Eq. (38), and by writing the reproducing conditions for $u^a(x,y)$, the correction function coefficients $c_0(x,y), c_1(x,y), \ldots, c_5(x,y)$ can be determined by solving the matrix problem

$$\begin{bmatrix} m_{00} & m_{10} & m_{01} & m_{20} & m_{02} & m_{11} \\ m_{10} & m_{20} & m_{11} & m_{30} & m_{12} & m_{21} \\ m_{01} & m_{11} & m_{02} & m_{21} & m_{03} & m_{12} \\ m_{20} & m_{30} & m_{21} & m_{40} & m_{22} & m_{31} \\ m_{02} & m_{12} & m_{03} & m_{22} & m_{04} & m_{13} \\ m_{11} & m_{21} & m_{12} & m_{31} & m_{13} & m_{22} \end{bmatrix} \begin{Bmatrix} c_0 \\ c_1 \\ c_2 \\ c_3 \\ c_4 \\ c_5 \end{Bmatrix} = \begin{Bmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{Bmatrix} \tag{42}$$

where

$$m_{ij}(x,y) = \int_\Omega (x-s)^i (y-s)^j w_{\mathrm{d}}(x-s, y-s)\,\mathrm{d}s \tag{43}$$

The first and second derivatives of the correction function coefficients can be determined in a similar manner. The RKPM shape function in two-dimensions for a particle I can be written as

$$N_I(x,y) = C(x-x_I, y-y_I) w_{\mathrm{d}}(x-x_I, y-y_I)\Delta V_I \tag{44}$$

where

$$\begin{aligned} C(x-x_I, y-y_I) = {} & c_0 + c_1(x-x_I) + c_2(y-y_I) \\ & + c_3(x-x_I)^2 + c_4(y-y_I)^2 \\ & + c_5(x-x_I)(y-y_I) \end{aligned} \tag{45}$$

$$w_{\mathrm{d}}(x-x_I, y-y_I) = \frac{1}{d_x} w\left(\frac{x-x_I}{d_x}\right) \frac{1}{d_y} w\left(\frac{y-y_I}{d_y}\right) \tag{46}$$

$d_x,\ d_y$ are the dilation parameters along the x and y directions, respectively, and kernel function w is the cubic spline defined in Eq. (36).

3 Meshless analysis of electromechanical systems

Over the last decade several micro sensors and micro actuators have been developed based on the electromechanical principle. The key problem in electromechanical systems is to accurately predict the deflection or deformation of a micromechanical structure when subjected to electrostatic forces. While accurate modeling and design of electromechanical systems requires three-dimensional analysis [1], critical insights into the behavior of electromechanical systems can be obtained by performing one and two dimensional analysis of microstructures subjected to electrostatic potentials. In this paper we will focus on just two two applications of MEMS – microwitches and pressure sensors. The governing equations, RKPM formulations and numerical results for the two applications are presented in the following sections.

3.1 Governing equations

The static and dynamic behavior of many microswitches can be characterized by studying the deformation of fixed-fixed and cantilever beams subjected to electrostatic voltages. The governing equation of a beam subjected to electrostatic forces is given by [27]

$$\frac{\rho}{EI}\frac{\partial^2 u}{\partial t^2} + \frac{\partial^4 u}{\partial x^4} = -\frac{\tilde{w}\varepsilon_0 V^2}{2EIg^2}\left(1 + 0.65\frac{g}{\tilde{w}}\right) \tag{47}$$

where ρ is the mass density per unit length of the beam, u is the displacement of the beam, E is the Young's modulus of the material, I is the moment of inertia, $\tilde{w}$ is the width of the beam, ε_0 is the permittivity of free space, V is the applied voltage and g is the gap between the beam and the ground electrode. Equation (47) is nonlinear because the gap, g, depends on the displacement of the beam i.e. $g = g(u)$.

For a fixed-end, boundary conditions are imposed on the displacement and its slope i.e.

$$\begin{aligned} u &= 0 \\ \frac{\mathrm{d}u}{\mathrm{d}x} &= \bar{u}_{,x} = 0 \end{aligned} \tag{48}$$

An RKPM formulation for Eq. (47) with fixed-boundary conditions at both ends is developed in the next section. A free-end boundary condition can be treated in a straightforward manner.

3.2 RKPM Formulation

Denoting v to be an arbitrary function, a weak-form to the strong form given in Eq. (47) can be developed from the following equation (see. e.g. [11] for details on developing weak formulations)

$$\begin{aligned} & \int_\Omega v\left(\frac{\rho}{EI}\right) u_{,tt}\ \mathrm{d}\Omega + \int_\Omega v(u_{,xxxx} - P(u))\,\mathrm{d}\Omega \\ & \quad + \int_\Gamma \delta\lambda(u_{,x} - \bar{u}_{,x}) n\,\mathrm{d}\Gamma = 0 \end{aligned} \tag{49}$$

where $P(u)$ is the nonlinear right-hand side term from Eq. (47), Ω is the domain, Γ is the boundary of the domain, the boundary-integral in Eq. (49) is required for the imposition of the gradient boundary condition through a Lagrange multiplier technique, λ is the Lagrange multiplier, $\delta\lambda$ is the variation of the Lagrange multiplier and n is the unit outward normal. The imposition of boundary conditions on u is discussed in Sect. 3.2.4. Integrating Eq. (49) by parts and noting that

$$\begin{aligned} \lambda &= -u_{,xx} \\ \delta\lambda &= -v_{,xx} \end{aligned} \tag{50}$$

the weak formulation is summarized as

$$\begin{aligned} & \int_\Omega v\left(\frac{\rho}{EI}\right) u_{,tt}\ \mathrm{d}\Omega + \int_\Omega v_{,xx}\, u_{,xx}\ \mathrm{d}\Omega - \int_\Gamma v_{,x}\, u_{,xx}\, n\,\mathrm{d}\Gamma \\ & \quad - \int_\Gamma v_{,xx}\, u_{,x}\, n\,\mathrm{d}\Gamma = \int_\Omega vP(u)\ \mathrm{d}\Omega - \int_\Gamma v_{,xx}\, \bar{u}_{,x}\, n\,\mathrm{d}\Gamma \end{aligned} \tag{51}$$

To obtain a matrix form from Eq. (51), the displacement field u and the function v are approximated by using the RKPM shape functions i.e.

$$u = \sum_{A=1}^{NP} N_A u_A \tag{52}$$

$$v = \sum_{A=1}^{NP} N_A v_A \tag{53}$$

where N_A is the RKPM shape function developed in Eq. (33) and u_A, v_A are the unknowns associated with particle A. Substituting the RKPM approximations for u and v into the weak formulation, a nonlinear residual equation for a particle A can be written as

$$\mathbf{R}_A(u) = \mathbf{R}_A^{\mathrm{Dyn}}(u) + \mathbf{R}_A^{\mathrm{Stat}}(u) = 0 \tag{54}$$

where $\mathbf{R}_A^{\mathrm{Dyn}}(u)$ is the dynamic residual, $\mathbf{R}_A^{\mathrm{Stat}}(u)$ is the static residual and

$$\mathbf{R}_A^{\mathrm{Dyn}}(u) = \int_\Omega \left(N_A \sum_{B=1}^{NP} N_B u_{B,tt} \right) \mathrm{d}\Omega \tag{55}$$

$$\begin{aligned} \mathbf{R}_A^{\mathrm{Stat}}(u) = & \int_\Omega \left(N_{A,xx} \sum_{B=1}^{NP} N_{B,xx} u_B \right) \mathrm{d}\Omega \\ & - \int_\Gamma \left(N_{A,x} \sum_{B=1}^{NP} N_{B,xx} u_B n \right) \mathrm{d}\Gamma \\ & - \int_\Gamma \left(N_{A,xx} \sum_{B=1}^{NP} N_{B,x} u_B n \right) \mathrm{d}\Gamma \\ & - \int_\Omega N_A P(u)\, \mathrm{d}\Omega + \int_\Gamma N_{A,xx} \bar{u}_{,x}\, n\, \mathrm{d}\Gamma \end{aligned} \tag{56}$$

The integrals in Eq. (56) are evaluated through a simple quadrature rule. For example, in a one-dimensional case a trapezoidal rule can be employed i.e.

$$\int_\Omega f(x)\, \mathrm{d}\Omega = \sum_{I=1}^{NP} f(x_I) \Delta V_I \tag{57}$$

Since the construction of the shape functions or the evaluation of the integrals does not require a mesh or connectivity between various nodes covering the domain, this approach is also referred to as a meshless approach.

3.2.1 Static analysis

For static analysis, the dynamic residual term in Eq. (54) is not considered and the residual $\mathbf{R}_A(u)$ is simply the static residual. Equation (56) (without the dynamic residual term) can then be solved by employing a Newton's method. The displacement increment within each Newton iteration can be computed by solving the following equation

$$\frac{\partial \mathbf{R}_A^{\mathrm{Stat}}}{\partial \mathbf{u}_B} \Delta \mathbf{u}_B = -\mathbf{R}_A^{\mathrm{Stat}}(u) \tag{58}$$

In matrix form, Eq. (58) can be stated as

$$\mathbf{J}(u) \Delta \mathbf{u} = -\mathbf{R}^{\mathrm{Stat}}(u) \tag{59}$$

where $\mathbf{J}(u) \in \mathscr{R}^{NP \times NP}$ is the Jacobian matrix, $\Delta \mathbf{u} \in \mathscr{R}^{NP \times 1}$ is the displacement increment vector, and $\mathbf{R}^{\mathrm{Stat}}(u) \in \mathscr{R}^{NP \times 1}$ is the static residual vector. The entries of the Jacobian matrix are given by

$$\begin{aligned} \mathbf{J}_{AB}(u) = & \int_\Omega N_{A,xx} N_{B,xx}\, \mathrm{d}\Omega - \int_\Gamma N_{A,x} N_{B,xx} n\, \mathrm{d}\Gamma \\ & - \int_\Gamma N_{A,xx} N_{B,x} n\, \mathrm{d}\Gamma - \int_\Omega N_A \frac{\partial P}{\partial u} N_B\, \mathrm{d}\Omega \end{aligned} \tag{60}$$

where $\mathbf{J}_{AB}$ is the Ath row and Bth column element of $\mathbf{J}$.

3.2.2 Dyanamic analysis

For dynamic analysis, we need to solve a second order system of nonlinear ordinary differential equations. Using a central difference rule to compute the time-derivatives of u, the displacement at time $(n+1)$ can be computed by solving the following system of equations

$$\mathbf{M}\left(\frac{u_{n+1} - 2u_n + u_{n-1}}{\Delta t^2} \right) + \mathbf{R}^{\mathrm{Stat}}(u_{n+1}) = 0 \tag{61}$$

where u_{n+1}, u_n and u_{n-1} are the displacements at times $(n+1)$, n and $(n-1)$, respectively, Δt is the time step, and $\mathbf{M}$ is the mass matrix. The entries of the mass matrix are computed as

$$\mathbf{M}_{AB} = \int_\Omega N_A \left(\frac{\rho}{EI} \right) N_B\, \mathrm{d}\Omega \tag{62}$$

Given initial conditions for the displacement at times n and $n-1$, the displacement at time $(n+1)$ is computed by a Newton method. The i-th displacement increment in a Newton approach to compute the displacement u_{n+1} is given as

$$\begin{aligned} \left[\frac{\mathbf{M}}{\Delta t^2} + \mathbf{J}\left(u_{n+1}^{(i)} \right) \right] \delta u_{n+1}^{(i)} = - \Bigg[& \mathbf{M}\left(\frac{u_{n+1}^{(i)} - 2u_n + u_{n-1}}{\Delta t^2} \right) \\ & + \mathbf{R}^{\mathrm{Stat}}\left(u_{n+1}^{(i)} \right) \Bigg] \end{aligned} \tag{63}$$

where $u_{n+1}^{(i)}$ denotes the displacement guess to the i-th Newton iteration at time level $(n+1)$, and $\delta u_{n+1}^{(i)}$ is the increment computed from the i-th Newton iteration. A displacement guess to the $(i+1)$th Newton iteration is computed as

$$u_{n+1}^{(i+1)} = u_{n+1}^{(i)} + \delta u_{n+1}^{(i)} \tag{64}$$

3.2.3 Shape function and its derivatives

The definition of the RKPM shape function given in Eq. (33) involves the correction function and this needs to be determined before the shape function and its derivatives can be computed. For the problem considered in this paper, the highest order derivative contained in the weak formulation (see Eq. (51)) is two. For the shape functions to satisfy consistency conditions, the correction function is chosen to be

$$C = c_0(x) + c_1(x)(x-s) + c_2(x)(x-s)^2 \tag{65}$$

It can easily be verified that a correction function involving just two coefficients (c_0 and c_1) does not satisfy the consistency conditions when the highest derivative con-

tained in the weak form is two. From the general case presented in Eq. (15), the matrix form for determining the correction coefficients can be written as

$$\begin{bmatrix} m_0 & m_1 & m_2 \\ m_1 & m_2 & m_3 \\ m_2 & m_3 & m_4 \end{bmatrix} \begin{Bmatrix} c_0 \\ c_1 \\ c_2 \end{Bmatrix} = \begin{Bmatrix} 1 \\ 0 \\ 0 \end{Bmatrix} \tag{66}$$

where the moments are as defined in Eq. (14). Expressing Eq. (66) as $\mathbf{M}\bar{\mathbf{C}} = \bar{\mathbf{M}}$, the first and second gradients of the correction function coefficients can be computed as

$$\bar{\mathbf{C}}_{,x} = -\mathbf{M}^{-1}\mathbf{M}_{,x}\,\bar{\mathbf{C}} \tag{67}$$

$$\bar{\mathbf{C}}_{,xx} = -\mathbf{M}^{-1}(\mathbf{M}_{,xx}\,\bar{\mathbf{C}} + 2\mathbf{M}_{,x}\,\bar{\mathbf{C}}_{,x}) \tag{68}$$

where

$$\bar{\mathbf{C}}_{,x} = \begin{bmatrix} c_{0,x} \\ c_{1,x} \\ c_{2,x} \end{bmatrix} \quad \bar{\mathbf{C}}_{,xx} = \begin{bmatrix} c_{0,xx} \\ c_{1,xx} \\ c_{2,xx} \end{bmatrix} \tag{69}$$

$$\mathbf{M}_{,x} = \begin{bmatrix} m_{0,x} & m_{1,x} & m_{2,x} \\ m_{1,x} & m_{2,x} & m_{3,x} \\ m_{2,x} & m_{3,x} & m_{4,x} \end{bmatrix}$$

$$\mathbf{M}_{,xx} = \begin{bmatrix} m_{0,xx} & m_{1,xx} & m_{2,xx} \\ m_{1,xx} & m_{2,xx} & m_{3,xx} \\ m_{2,xx} & m_{3,xx} & m_{4,xx} \end{bmatrix} \tag{70}$$

The shape function for particle I can then be written as

$$\begin{aligned} N_I(x) &= (c_0 + c_1(x - x_I) + c_2(x - x_I)^2) w_{\mathrm{d}}(x - x_I)\Delta V_I \\ &= C w_{\mathrm{d}}(x - x_I)\Delta V_I \end{aligned} \tag{71}$$

The first and second derivatives of the shape function can be computed by simply differentiating the shape function with respect to x i.e.

$$N_{I,x}(x) = [C_{,x}\, w_{\mathrm{d}}(x - x_I) + C w_{\mathrm{d},x}(x - x_I)]\Delta V_I \tag{72}$$

$$\begin{aligned} N_{I,xx}(x) = [&C_{,xx}\, w_{\mathrm{d}}(x - x_I) + 2C_{,x}\, w_{\mathrm{d},x}(x - x_I) \\ &+ C w_{\mathrm{d},xx}(x - x_I)]\Delta V_I \end{aligned} \tag{73}$$

where

$$\begin{aligned} C_{,x} = {} & c_{0,x} + c_1 + c_{1,x}(x - x_I) \\ & + 2c_2(x - x_I) + c_{2,x}(x - x_I)^2 \end{aligned} \tag{74}$$

$$\begin{aligned} C_{,xx} = {} & c_{0,xx} + 2c_{1,x} + c_{1,xx}(x - x_I) + 2c_2 \\ & + 4c_{2,x}(x - x_I) + c_{2,xx}(x - x_I)^2 \end{aligned} \tag{75}$$

3.2.4 Boundary conditions

The boundary conditions on the gradient of the displacement (i.e. the slope) are treated through a Lagrange multiplier technique and the application of Lagrange multipliers to the weak formulation is presented in Sect. 3.2. The Dirichlet boundary conditions on the displacemnt, such as the condition $u = 0$ in Eq. (48), are treated by eliminating the displacement unknowns associated with the constrained particles from RKPM analysis. Assuming that there are m constrained particles, the displacements at the particle positions can be written as

$$\begin{bmatrix} u(x_1) \\ \cdot \\ \cdot \\ u(x_m) \\ u(x_{m+1}) \\ \cdot \\ \cdot \\ u(x_{NP}) \end{bmatrix} = \left[\begin{array}{ccc:ccc} N_1(x_1) & \cdots & N_m(x_1) & N_{m+1}(x_1) & \cdots & N_{NP}(x_1) \\ \cdot & \cdots & \cdot & \cdot & \cdots & \cdot \\ \cdot & \cdots & \cdot & \cdot & \cdots & \cdot \\ N_1(x_m) & \cdots & N_m(x_m) & N_{m+1}(x_m) & \cdots & N_{NP}(x_m) \\ \hdashline N_1(x_{m+1}) & \cdots & N_m(x_{m+1}) & N_{m+1}(x_{m+1}) & \cdots & N_{NP}(x_{m+1}) \\ \cdot & \cdots & \cdot & \cdot & \cdots & \cdot \\ \cdot & \cdots & \cdot & \cdot & \cdots & \cdot \\ N_1(x_{NP}) & \cdots & N_m(x_{NP}) & N_{m+1}(x_{NP}) & \cdots & N_{NP}(x_{NP}) \end{array}\right] \begin{bmatrix} u_1 \\ \cdot \\ \cdot \\ u_m \\ u_{m+1} \\ \cdot \\ \cdot \\ u_{NP} \end{bmatrix} \tag{76}$$

where $u(x_i)$ is the displacement of the i-th particle, x_i is the position of the i-th particle and u_i is an unknown associated with the i-th particle. Note that u_i is not the displacement of the i-th particle. The unknowns $u_1, \ldots, u_{NP}$ are computed through RKPM analysis, and the displacements of the particles are computed through the shape function interpolation as described in Eq. (76). Since there are m constrained particles, the displacements for the m particles are known i.e.

$$\begin{aligned} u(x_1) &= g_1 \\ &\cdot \\ &\cdot \\ u(x_m) &= g_m \end{aligned} \tag{77}$$

where g_i is the prescribed displacement of the i-th particle. From Eq. (76), the constrained displacements can be expressed as

$$\mathbf{g} = \mathbf{A}\tilde{u}_1 + \mathbf{B}\tilde{u}_2 \tag{78}$$

where $\mathbf{g} = [g_1, \ldots, g_m]^{\mathrm{T}} \in \mathscr{R}^{m\times 1}$ is the vector of constrained displacements, $\mathbf{A} \in \mathscr{R}^{m\times m}$ is the matrix containing the first m rows and m columns of the shape function coefficient matrix given in Eq.(76), $\mathbf{B} \in \mathscr{R}^{m\times(NP-m)}$ is the matrix containing the first m rows and the last $(NP - m)$ columns of the shape function coefficient matrix, $\tilde{\mathbf{u}}_1 = [u_1, \ldots, u_m]^{\mathrm{T}} \in \mathscr{R}^{m\times 1}$ are the unknowns associated with the m constrained particles, $\tilde{\mathbf{u}}_2 = [u_{m+1}, \ldots, u_{NP}]^{\mathrm{T}} \in \mathscr{R}^{(NP-m)\times 1}$ are the unknowns associated with the $(NP - m)$ unconstrained particles. The

unknowns associated with the m constrained particles can be expressed as

$$\tilde{\mathbf{u}}_1 = \mathbf{A}^{-1}(\mathbf{g} - \mathbf{B}\tilde{u}_2) \tag{79}$$

The unknowns involved in the RKPM analysis can then be written as

$$\begin{Bmatrix} \tilde{\mathbf{u}}_1 \\ \tilde{\mathbf{u}}_2 \end{Bmatrix} = \begin{bmatrix} \mathbf{A}^{-1} & -\mathbf{A}^{-1}\mathbf{B} \\ 0 & \mathbf{I} \end{bmatrix} \begin{Bmatrix} \mathbf{g} \\ \tilde{\mathbf{u}}_2 \end{Bmatrix} \tag{80}$$

where $\mathbf{I}$ is the identity matrix. Substituting (80) into (76), the unknown displacements for the particles are given by

$$\begin{Bmatrix} u(x_1) \\ . \\ . \\ u(x_{NP}) \end{Bmatrix} = \begin{bmatrix} \mathbf{A} & \mathbf{B} \\ \mathbf{C} & \mathbf{D} \end{bmatrix} \begin{bmatrix} \mathbf{A}^{-1} & -\mathbf{A}^{-1}\mathbf{B} \\ 0 & \mathbf{I} \end{bmatrix} \begin{Bmatrix} \mathbf{g} \\ \tilde{\mathbf{u}}_2 \end{Bmatrix} \tag{81}$$

where $\mathbf{C} \in \mathscr{R}^{(NP-m)\times m}$ is the matrix containing the last $(NP-m)$ rows and the first m columns of the shape function coefficient matrix given in Eq. (76) and $\mathbf{D} \in \mathscr{R}^{(NP-m)\times(NP-m)}$ is the matrix containing the last $(NP-m)$ rows and columns of the shape function coefficient matrix. The Dirichlet boundary conditions for the displacement field can now be applied using one of the two approaches suggested below:

The first approach is to compute a modified shape function coefficient matrix that accounts for the boundary conditions. From Eq. (81), the modified shape function coefficient matrix can be computed as

$$\tilde{\mathbf{N}} = \begin{bmatrix} \mathbf{I} & 0 \\ \mathbf{C}\mathbf{A}^{-1} & \mathbf{D} - \mathbf{C}\mathbf{A}^{-1}\mathbf{B} \end{bmatrix} \tag{82}$$

where $\tilde{\mathbf{N}}$ is the modified coefficient matrix. The elements of the matrix $\tilde{\mathbf{N}}$ are the RKPM shape functions evaluated at all the particle positions. Likewise the first and second gradients of the shape functions modified for the displacement conditions can also be calculated.

The second approach to prescribing displacement boundary conditions is to compute the Jacobian matrices for the static and dynamic analysis using the RKPM shape functions with no modifications for the boundary conditions i.e. the displacement vector is approximated as given in Eq. (76). Denoting

$$\mathscr{L} = \begin{bmatrix} \mathbf{A}^{-1} & -\mathbf{A}^{-1}\mathbf{B} \\ 0 & \mathbf{I} \end{bmatrix} \tag{83}$$

the boundary conditions on the displacement can be prescribed by modifying the Jacobian matrix as

$$\tilde{\mathbf{J}} = \mathscr{L}^{\mathrm{T}}\mathbf{J}\mathscr{L} \tag{84}$$

Both approaches give identical results and it can easily be verified that the Jacobian obtained with the first approach, where the shape functions and its derivatives are modified as discussed in Eq. (82), is identical to the Jacobian, $\tilde{\mathbf{J}}$, computed using the second approach.

Once the modified Jacobian $\tilde{\mathbf{J}}$ is computed, the unknowns $\tilde{u}_2$ can be computed by considering a partitioning of $\tilde{\mathbf{J}}$ as

$$\begin{bmatrix} \tilde{\mathbf{J}}_1 & \tilde{\mathbf{J}}_2 \\ \tilde{\mathbf{J}}_3 & \tilde{\mathbf{J}}_4 \end{bmatrix} \begin{Bmatrix} \mathbf{g} \\ \tilde{\mathbf{u}}_2 \end{Bmatrix} = -\begin{Bmatrix} \tilde{\mathbf{R}}_1 \\ \tilde{\mathbf{R}}_2 \end{Bmatrix} \tag{85}$$

where $\tilde{\mathbf{J}}_1 \in \mathscr{R}^{m\times m}$ is the matrix containing the first m rows and m columns of $\tilde{\mathbf{J}}$, $\tilde{\mathbf{J}}_2 \in \mathscr{R}^{m\times(NP-m)}$ is the matrix containing the first m rows and the last $(NP-m)$ columns of $\tilde{\mathbf{J}}$, $\tilde{\mathbf{J}}_3 \in \mathscr{R}^{(NP-m)\times m}$ is the matrix containing the last $(NP-m)$ rows and the first m columns of $\tilde{\mathbf{J}}$, and $\tilde{\mathbf{J}}_4 \in \mathscr{R}^{(NP-m)\times(NP-m)}$ is the matrix containing the last $(NP-m)$ rows and columns of $\tilde{\mathbf{J}}$. $\tilde{\mathbf{R}}_1 \in \mathscr{R}^{m\times 1}$ is the residual vector associated with the m constrained nodes and $\tilde{\mathbf{R}}_2 \in \mathscr{R}^{(NP-m)\times 1}$ is the residual vector associated with the $(NP-m)$ unconstrained nodes. The definitions for these residuals are given in Sect. 3.2.1 and 3.2.2, respectively, for the static and dynamic cases. From Eq. (85), $\tilde{u}_2$ can be computed as

$$\tilde{\mathbf{J}}_4\tilde{\mathbf{u}}_2 = -\tilde{\mathbf{R}}_2 - \tilde{\mathbf{J}}_3\mathbf{g} \tag{86}$$

Once $\tilde{u}_2$ is computed, the unknowns associated with the m constrained particles, $\tilde{u}_1$, can be computed through Eq. (79). Once the $\tilde{u}_1$ and $\tilde{u}_2$ vectors are known, the displacement at each particle position can be computed by using Eq. (76).

4 Results

Numerical results are shown for three devices: a fixed-fixed beam over a ground plane is analyzed in Sect. 4.1, a cantilever beam over a ground plane is analyzed in Sect. 4.2, and a pressure sensor is analyzed in Sect. 4.3. All results are obtained with the RKPM meshless formulation. The kernel function employed for all the calculations is the cubic spline function given in Eq. (36).

4.1 Fixed-fixed beam

The first example considered here is a fixed-fixed beam over a ground plane as shown in Fig. 6. The beam is 80 μm long, 10 μm wide, and 0.5 μm thick. The initial gap between the beam and the bottom electrode is 0.7 μm. A Young's modulus of 169 GPa and a mass density of 2231 kg/m^3 are employed in the simulations shown here. The same device with the material properties has been simulated in the work presented in [2] and the static and dynamic pull-in values obtained through experiments and simulations were reported there. This device is simulated by employing 101 particles. For boundary conditions, the displacement and the slope are assumed constrained at both ends of the beam. Each end of the beam is represented by a particle, and m (see Sect. 3.2.4) for this case is 2.

Figure 7 shows the deflection of the beam as a function of a series of applied voltages. As the potential difference

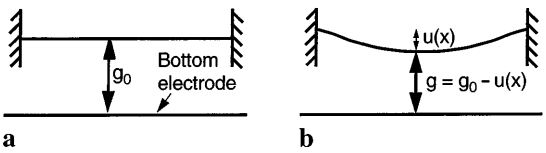


Fig. 6. **a** A fixed-fixed beam and a bottom electrode separated by a gap g_0, **b** deflection of the beam when a potential difference is created between the beam and the bottom electrode

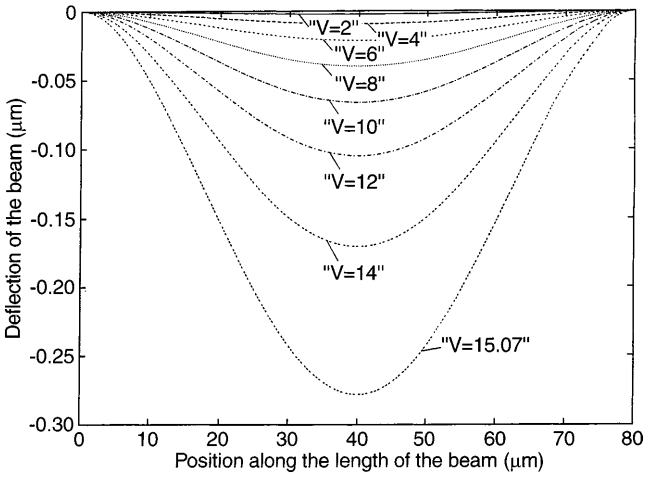


Fig. 7. Deflection of a fixed-fixed beam for a series of applied voltages. The pull-in voltage is 15.08 V

between the beam and the ground electrode increases, the beam deflects closer to the ground electrode and the gap between and the bottom electrode decreases. As the beam deflects closer to the ground electrode, the electrostatic force on the beam increases highly nonlinearity and the deflection of the beam is also highly nonlinear. At a certain voltage, defined as the pull-in voltage, the beam becomes unstable and collapses onto the ground electrode. For the beam under consideration, the pull-in voltage is computed to be 15.08 volts. The pull-in value reported in [2] is 15.17 volts and the error is less than 1%.

The dynamic calculations on the beam using the meshless approach are summarized in Figs. 8–10. A time step of 5×10^{-4} μs is employed for the fixed-fixed beam case. The peak deflection of the beam, which is at the center of the beam, for an applied bias of 4 and 8 volts is shown in Fig. 8 and the dynamic response for an applied bias of 12 and 13.6 volts is shown in Fig. 9. Note that the results make sense as the problem under consideration is a second-order non-linear dynamical system with no damping. Also note that as the applied bias on the beam is increased, the time period, which is the time required for the beam to make one complete cycle, increases. Such results can be used to calculate the fundamental or resonant frequency of the device. For an applied bias of 13.7 volts, the beam deflects and make a contact with the ground electrode. This voltage is defined as the dynamic pull-in voltage. The dynamic pull-in voltage reported here matches with the dynamic pull-in value reported in [2] for the same fixed-fixed beam example. The peak deflection of the beam when a dynamic pull-in voltage is applied is shown in Fig. 10. It is observed that the dynamic pull-in

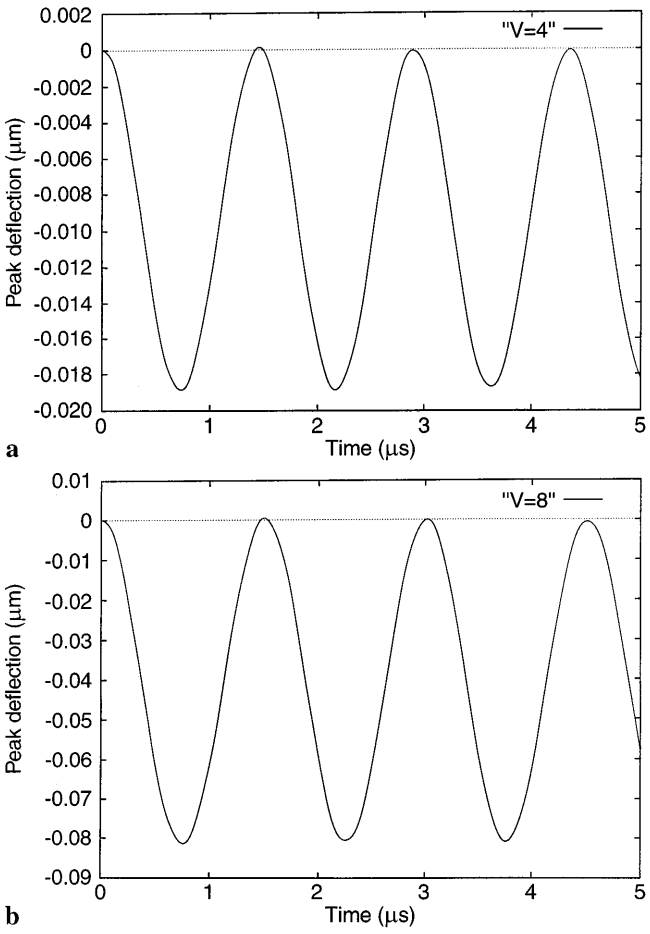


Fig. 8a, b. Dynamic analysis of a fixed-fixed beam. **a** Applied bias is 4 volts, **b** applied bias is 8 volts

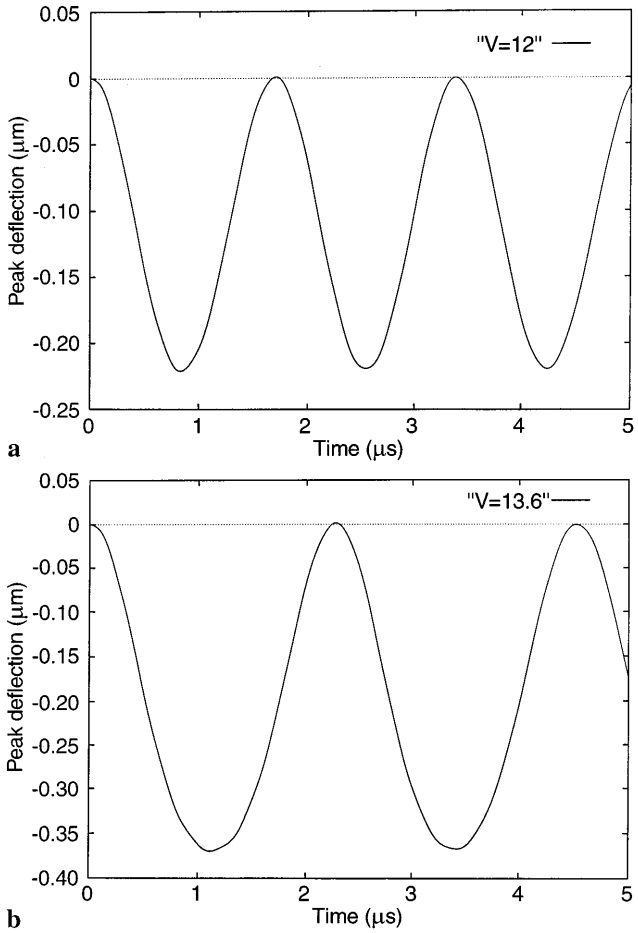


Fig. 9a, b. Dynamic analysis of a fixed-fixed beam. **a** Applied bias is 12 volts, **b** applied bias is 13.6 volts – this calculation is just before pull-in

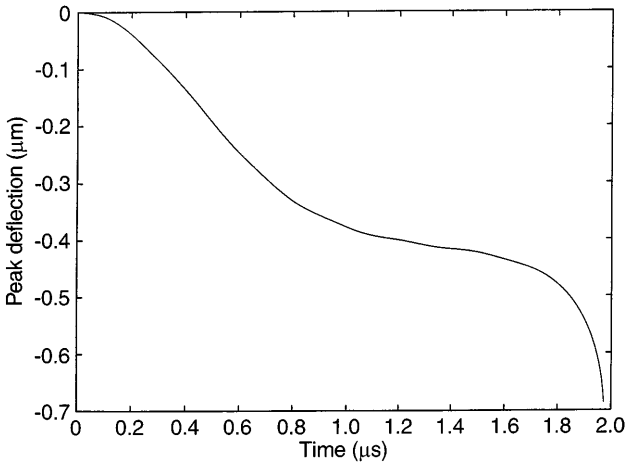


Fig. 10. Pull-in behavior of the fixed-fixed beam for a bias of 13.7 volts

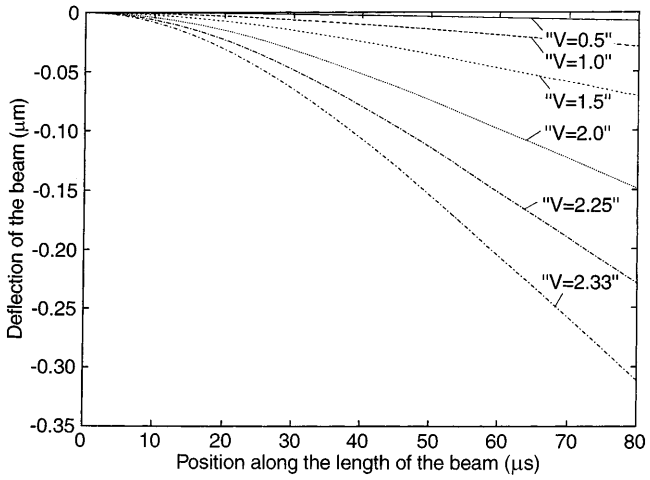


Fig. 12. Deflection of a cantilever beam for a series of applied voltages. The pull-in voltage is 2.34 volts

voltage is smaller than the quasi-static pull-in voltage by about 9% .

4.2 Cantilever beam

The second example considered here is a cantilever beam over a ground plane as shown in Fig. 11. The beam dimensions are identical to the fixed-fixed beam i.e. the beam is 80 μm long, 10 μm wide, and 0.5 μm thick. The initial gap between the beam and the bottom electrode is 0.7 μm. A total of 101 particles are employed to simulate this problem. Since the right-end of the beam is left free, no boundary conditions are imposed on this end. For the left-end of the beam, the displacement and the slope are assumed constrained. For this example, m is just 1.

Figure 12 shows the deflection of the cantilever beam as a function of a series of applied voltages. When a potential difference is created between the cantilever beam and the ground electrode, the tip of the cantilever beam deflects towards the ground electrode. Similar to the fixed-fixed beam example, the cantilever beam also becomes unstable for a certain bias, defined as the pull-in voltage, and when this voltage is applied, the beam collapses and makes a contact with the ground electrode. The quasi-static pull-in voltage for the cantilever beam is computed as 2.34 volts.

The dynamic calculations for the cantilever beam are summarized in Figs. 13–15. A time step of 5×10^{-3} μs is employed and the simulation is carried out for a total of 25 μs. The peak deflection of the beam, which is at the tip of the beam, for an applied bias of 1 and 1.5 volts is shown in Fig. 13 and the dynamic response for an applied bias of 2 and 2.11 volts is shown in Fig. 14. As the applied bias increases, the time period to complete one full cycle increases. For an applied bias of 2.12 volts, the tip of the beam deflects and makes a contact with the ground electrode. The peak deflection of the beam when a dynamic pull-in voltage (2.12 volts) is applied is shown in Fig. 15.

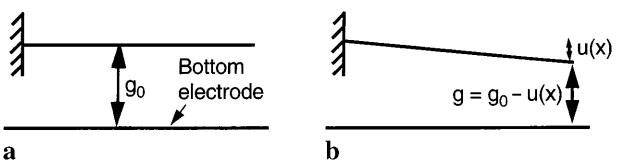


Fig. 11. a A cantilever beam and a bottom electrode separated by a gap g_0, **b** deflection of the beam when a potential difference is created between the beam and the bottom electrode

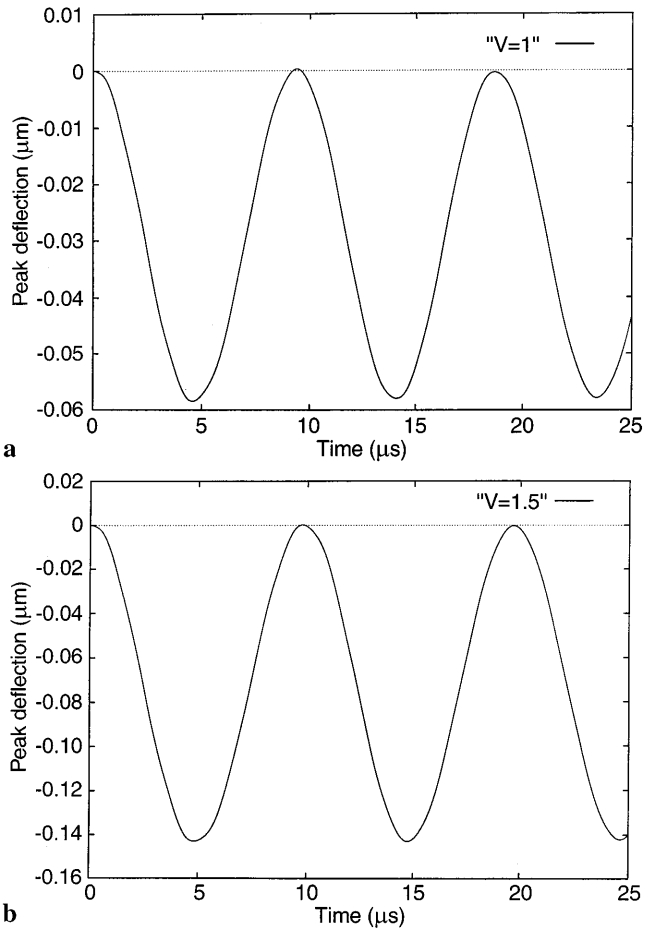


Fig. 13a, b. Dynamic analysis of a cantilever beam. **a** Applied bias is 1 volt, **b** applied bias is 1.5 volts

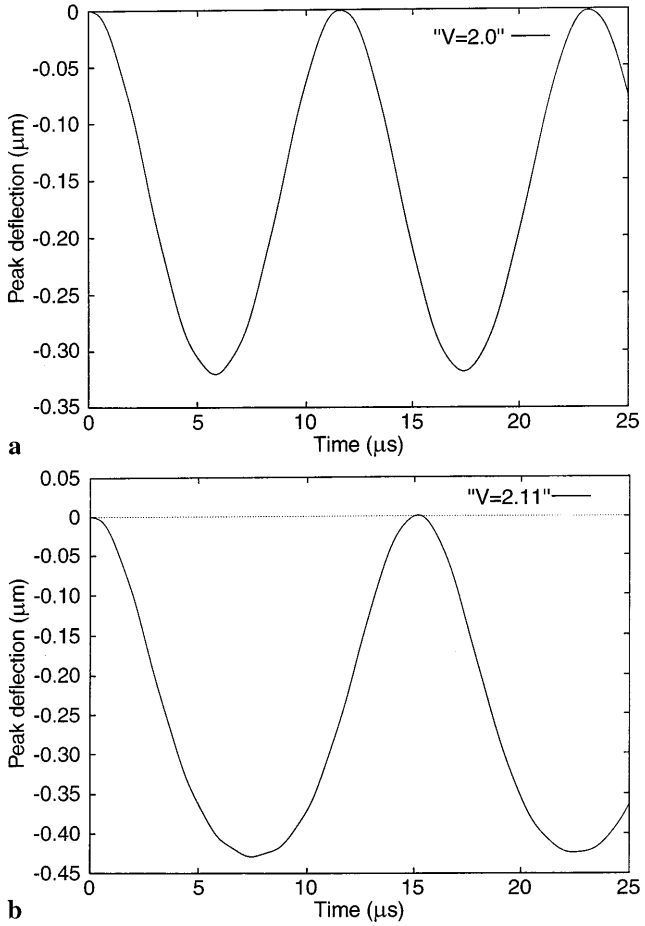


Fig. 14a, b. Dynamic analysis of a cantilever beam. **a** Applied bias is 2 volts, **b** applied bias is 2.11 volts – this calculation is just before pull-in

It is observed that the dynamic pull-in voltage is smaller than the quasi-static pull-in voltage by about 9%.

4.3 Two-dimensional example: Electromechanical pressure sensor

The governing equation for the electromechanical pressure sensor is given by the thin plate theory [7] i.e.

$$\frac{\partial^4 u}{\partial x^4} + \frac{\partial^4 u}{\partial y^4} + 2G\frac{\partial^4 u}{\partial x^2 \partial y^2} = \frac{p}{D} \tag{87}$$

where u is the deflection of the thin plate, p is the applied pressure, D is the bending rigidity given by

$$D = \frac{\tilde{E}h^3}{12(1-\upsilon^2)} \tag{88}$$

$$G = 2\frac{\tilde{G}}{\tilde{E}}(1-\upsilon^2) + \upsilon \tag{89}$$

$\tilde{E} = E_x = E_y$ is the Young's modulus, $\tilde{G} = G_{xy}$ is the shear modulus, h is the thickness of the plate, and υ is the Poisson's ratio.

We will consider the solution of a thin square plate ($0 \le x \le 2a,\ 0 \le y \le 2a$) with fixed boundary conditions along the four edges. Because of symmetry, only a quarter

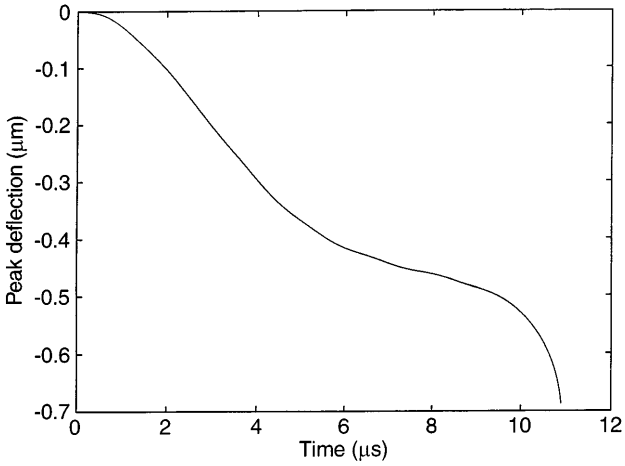


Fig. 15. Pull-in behavior of the cantilever beam for a bias of 2.12 volts

of a plate is simulated i.e. $0 \le x \le a$ and $0 \le y \le a$. Assuming v is a weighting function, a finite-element weak-formulation accounting for the imposition of the slope boundary conditions $\left(\frac{\partial u}{\partial x} = 0 \text{ or } \frac{\partial u}{\partial y} = 0\right)$ through Lagrange multipliers is given as

$$\begin{aligned}
&\int_\Omega \left(\frac{\partial^2 v \partial^2 u}{\partial x^2 \partial x^2} + \frac{\partial^2 \partial^2 u}{\partial y^2 \partial y^2} + 2G\frac{\partial^2 v}{\partial x \partial y}\frac{\partial^2 u}{\partial x \partial y} \right) d\Omega \\
&\quad + \int_{\Gamma_2} \frac{\partial v \partial^2 u}{\partial x \partial x^2} dy + \int_{\Gamma_2} \frac{\partial^2 v \partial u}{\partial x^2 \partial x} dy \\
&\quad + \int_{\Gamma_1} \frac{\partial v \partial^2 u}{\partial y \partial y^2} dx + \int_{\Gamma_1} \frac{\partial^2 v \partial u}{\partial y^2 \partial y} dx \\
&\quad - \int_{\Gamma_3} \frac{\partial v \partial^2 u}{\partial x \partial x^2} dy - \int_{\Gamma_3} \frac{\partial^2 v \partial u}{\partial x^2 \partial x} dy \\
&\quad - \int_{\Gamma_4} \frac{\partial v \partial^2 u}{\partial y \partial y^2} dx - \int_{\Gamma_4} \frac{\partial^2 v \partial u}{\partial y^2 \partial y} dx = \int_\Omega v \frac{p}{D}
\end{aligned} \tag{90}$$

where Γ_1 is the fixed boundary parallel to x-axis (the boundary defined by $y = 0$), Γ_2 is the fixed boundary that is parallel to y-axis (the boundary defined by $x = 0$), Γ_3 is the symmetric boundary parallel to y-axis (the boundary defined by $x = a$) and Γ_4 is the symmetric boundary that is parallel to x-axis (the boundary defined by $y = a$). Equation (90) is solved by employing the two-dimensional RKPM presented in Sect. 2.4. The Dirichlet boundary conditions are imposed by employing the technique described in Sect. 3.2.4.

Numerical results are obtained for a thin square plate whose side is 1000 μm long and 10 μm thick. The Young's modulus is taken as $\tilde{E} = 1.698 \times 10^{11}$ N/m^2, the shear modulus is taken as $\tilde{G} = 0.622 \times 10^{11}$ N/m^2, and the Poisson's ratio is 0.066. For an applied pressure of 100 mmHg, the deflection of the plate is shown in Fig. 16. The peak deflection obtained with the RKPM meshless approach is 1.24 μm, and this matches closely with the peak deflection of 1.27 μm reported in [7]. The peak deflection, which is at the center of the plate, as a function of applied pressure is shown in Fig. 17. Also shown in Fig. 17 is the variation of the peak deflection as a function of the

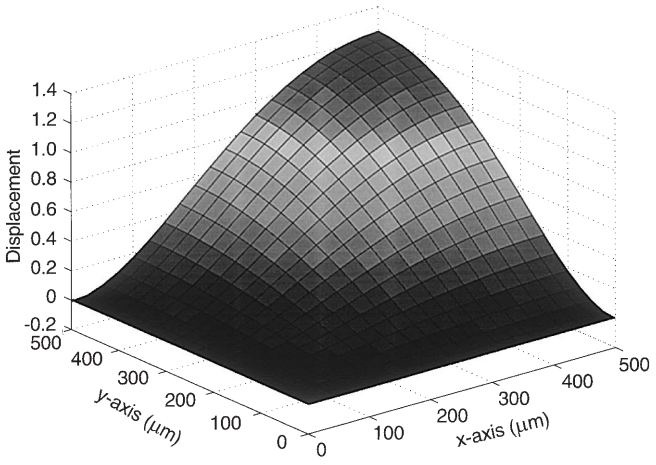


Fig. 16. Deflection of the plate for an applied pressure of 100 mmHg

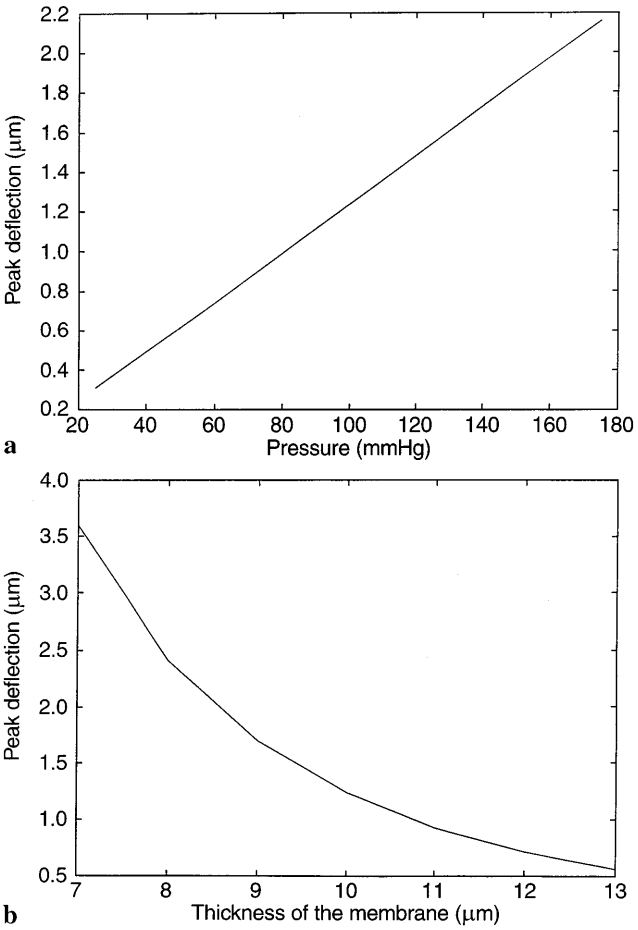


Fig. 17. a Peak deflection in micro meters as a function of applied pressure, **b** peak deflection in micro meters as a function of the thickness of the plate

thickness of the plate. These results are obtained for an applied pressure of 100 mmHg. As the thickness decreases, the plate becomes more flexible and the peak deflection increases. This behavior is captured correctly by the RKPM meshless approach.

5 Conclusions

A meshless technique for analysis of microelectromechanical devices is presented here for the first time. The meshless approach is based on consistent reproducing kernel approximations. A correction function that satisfies consistency conditions is derived when second derivatives of the function are involved in the weak form of the governing equations. A simple approach combining constraint elimination and Lagrange multipliers is presented to enforce boundary conditions exactly.

Numerical results are presented for nonlinear static and dynamic analysis of microswitches and for two-dimensional analysis of electromechanical pressure sensors. The meshless approach is shown to be accurate by comparing the computed pull-in voltages and peak deflections with the previously reported data. Finally, it is anticipated that with meshless techniques in place, radically simpler and easy to use computer-aided design tools will be available for MEMS designers.

References

[1] **Aluru NR, White J** (1997) An efficient numerical technique for electromechanical simulation of complicated microelectromechanical structures. Sensors and Actuators A , 58: 1–11

[2] **Ananthasuresh GK, Gupta RK, Senturia SD** (1996). An approach to macromodeling of MEMS for nonlinear dynamic simulation. Microelectromechanical Systems (MEMS), ASME Dynamic Systems and Control (DSC) series, New York 59: 401–407

[3] **Atluri SN, Zhu T** (1998) A new meshless Local Petrov-Galerkin (MLPG) approach in computational mechanics. Comput. Meth. 22: 117–127

[4] **Belytschko T, Krongauz Y, Organ D, Fleming M, Krysl P** (1996) Meshless methods: An over-view and recent developments. Comput. Meth. Appl. Mech. Eng. 139: 3–47

[5] **Belytschko T, Lu YY, Gu L** (1994) Element free Galerkin methods. Int. J. Num. Meth. Eng. 37: 229–256

[6] **Chen J, Pan C, Wu C, Liu WK** (1996) Reproducing kernel particle methods for large deformation analysis of non-linear structures. Comput. Meth. Appl. Mech. Eng. 139: 195–227

[7] **Clark SK, Wise KD** (1979) Pressure sensitivity in anisotropically etched thin-diaphragm pressure sensors. IEEE Trans. on Electron Devices ED-26(12): 1887–1896

[8] **Crary SB, Zhang Y** CAEMEMS: An integrated computer-aided engineering work-bench for microelectromechanical systems. Proc. MEMS'90, pp. 113–114 Napa Valley, CA, Feb. 1990

[9] **Duarte CA, Oden JT** (1996) An h-p adaptive method using clouds. Comput. Meth. Appl. Mech. Eng. 139: 237–262

[10] **Funk J, Korvink JG, Buhler J, Bachtold M, Baltes H** (1997) SOLIDIS: A tool for microactuator simulation in 3D. IEEE J. Microelectromechanical Systems 6(1): 70–82

[11] **Hughes TJR** (1987) The finite element method. Prentice-Hall

[12] **Krongauz Y, Belytschko T** (1996) Enforcement of essential boundary conditions in meshless approximations using finite elements. Comput. Meth. Appl. Mech. Eng. 131: 133–145

[13] **Liu WK, Chen Y** (1995) Wavelet and multiple scale reproducing kernel methods. International Journal for Numerical Methods in Fluids 21: 901–931

[14] **Liu WK, Chen Y, Chang CT, Belytschko T** (1996) Advances in multiple scale kernel particle methods. Comput. Mech. 18: 73–111

[15] **Liu WK, Chen Y, Jun S, Chen JS, Belytschko, Pan C, Uras RA Chang CT** (1996) "Overview and applications of the reproducing kernel particle methods," Archives of computational Methods in Engineering: State of the art reviews 3: 3–80
[16] **Liu WK, Jun S, Li S, Adee J, Belytschko T** (1995) Reproducing kernel particle methods for structural dynamics. Int. J. Num. Meth. Eng. 38: 1655–1679
[17] **Liu WK, Jun S, Zhang YF** (1995) Reproducing kernel particle methods. Int. J. Num. Meth. Fluids 20: 1081–1106
[18] **Lu YY, Belytschko T, Gu L** (1994) A new implementation of the element free Galerkin method. Comput. Meth. Appl. Eng. 113: 397–414
[19] **Lucy LB** (1977) A numerical approach to the testing of the fission hypothesis. The Astron J. 8(12): 1013–1024
[20] **Melenk JM, Babuska I** (1996) The partition of unity finite element method: Basic theory and applications. Comput. Meth. Appl. Mech. Eng. 139: 289–314
[21] **Monaghan JJ** (1982) Why particle methods work. SIAM J. Sci. Stat. Comput. 3(4): 422-433
[22] **Monaghan JJ** (1988) An introduction to SPH. Comput. Phys. Comm. 48: 89–96
[23] **Mukherjee YX, Mukherjee S** (1997) The boundary node method for potential problems. Int. J. Numer. Meth. Eng. 40: 797–815
[24] **Mukherjee YX, Mukherjee S** (1997) On boundary conditions in the element-free Galerkin method. Comput. Mech. 19: 264–270
[25] **Nayroles B, Touzot G, Villon P** (1992) Generalizing the finite element method: diffuse approximation and diffuse elements. Comput. Mech. 10: 307–318
[26] **Onate E, Idelsohn S, Zienkiewicz OC, Taylor RL** (1996) A finite point method in computational mechanics. Applications to convective transport and fluid flow. Int. J. Num. Meth. Eng. 39: 3839–3866
[27] **Osterberg PM, Senturia SD** (1997) M-TEST: a test chip for MEMS material property measurement using electrostatically actuated test structures. J. Microelectromechanical Systems 6(2): 107–118
[28] **Senturia SD, Aluru NR, White JK** (1997) Simulating the behavior of MEMS devices: Computational methods and needs. IEEE Computational Science and Engineering 4(1): 30–43
[29] **Senturia SD, Harris RM, Johnson BP, Kim S, Nabors K, Shulman MA, White JK** (1992) A computer-aided design system for Microelectromechanical Systems. J. Microelectromechanical Systems 1: 3–13
[30] **Yu X, Saigal S** (1998) An element free Galerkin formulation for stable crack growth in an elastic solid. Comp. Meth. Appl. Mech. Eng. 154: 331–343
[31] **Zhu T, Atluri SN** (1998) A modified collocation method and a penalty formulation for enforcing the essential boundary conditions in the element free Galerkin method. Comput. Mech. 21: 211–222
[32] **Zhu T, Zhang J-D, Atluri SN** (1998) A local boundary integral equation (LBIE) method in computational mechanics, and a meshless discretization approach. Comput. Mech. 21: 223–235
[33] **Zhu T, Zhang J, Atluri SN** (1998) A meshless local boundary integral equation (LBIE) method for solving nonlinear problems. Comput. Mech. 22: 174–186