



# **An error indicator and automatic adaptive meshing for 3D electrostatic boundary element simulations**

M. Bächtold, J. G. Korvink and H. Baltes

*Physical Electronics Laboratory, Swiss Federal Institute of Technology Zurich, 8093 Zurich, Switzerland*

*Email: martinb@iqe.phys.ethz.ch*

## **1 Abstract**

Accurate electrostatic simulations are required in the areas of MEMS and VLSI interconnects. Typical simulations involve complex geometries and various boundary conditions. The boundary element method (BEM) is well suited to such computations. For highly accurate solutions the meshing of the geometry becomes increasingly important. A scheme is presented which allows generating a good mesh automatically. An error indicator based on boundary integral equations (BIE) monitors the simulation accuracy in each boundary element. Mesh refinement is applied to areas which contribute strongly to the overall error. The generated, refined meshes lead to significantly higher accuracy for a given computational effort.

## 2 The Boundary Element Method and Error Indicators

The BEM is widely applied to the solution of electrostatic simulations [1]. The method computes the electric potential and flux on the surface of the simulation domain, such that a weighted residual formulation of equation (1) is minimized.

$$u(\xi) = \frac{1}{c(\xi)} \int_{\Gamma} q(x) \cdot u^*(\xi, x) \cdot d\Gamma(x) - \frac{1}{c(\xi)} \int_{\Gamma} u(x) \cdot q^*(\xi, x) \cdot d\Gamma(x) \quad (1)$$

Here,  $\Gamma$  is the domain boundary,  $u(\xi)$  and  $q(\xi)$  are the boundary potential and flux, and  $u^*(\xi, x)$  and  $q^*(\xi, x)$  are the fundamental solution to Laplace's equation and its derivative in the direction of the outward normal on the boundary. The contribution of the singularity during the boundary integration is contained in  $c(\xi)$ . The domain surface is discretized into separate elements, each containing a number of collocation nodes. At these nodes the BIE is satisfied by the collocated BEM solution, at all other boundary positions the potential and flux is interpolated from nodal values using

$$u(\xi) = \sum_i N_i(\xi) \cdot u_i \quad \text{and} \quad q(\xi) = \sum_i N_i(\xi) \cdot q_i. \quad (2)$$

Here,  $N_i(\eta)$  are the element's shape functions, while  $u_i$  and  $q_i$  are nodal values for the boundary potential and flux. The BEM solution does not necessarily satisfy the BIE at positions other than the collocation nodes, and the mismatch between the interpolated value and the value computed from the BIE can be used to estimate the error in a particular boundary point  $\xi$  [2]

$$\hat{E}^2(\xi) = \left| \sum_i N_i(\xi) \cdot \hat{u}_i - \frac{1}{c(\xi)} \int_{\Gamma} (\hat{q}(x) \cdot u^*(\xi, x) - \hat{u}(x) \cdot q^*(\xi, x)) \cdot d\Gamma(x) \right| \quad (3)$$

Here,  $\hat{E}^2(\xi)$  is the error indicator for the boundary position  $\xi$ , while  $\hat{u}$  and  $\hat{q}$  are the approximated solutions delivered by an earlier BEM computation. An error indicator for an entire boundary element can be obtained by integrating the local error indicator over the element. This approach has been demonstrated to work well for 2D simulations [3], in 3D however, the integral over the local error indicator requires excessive computational effort. The element error indicator in 3D is obtained from a weighted sum of local error indicators

$$\hat{E}_{\Gamma_k}^2 = \sum_i w_i \cdot \hat{E}^2(\xi_i) \quad (4)$$

where  $\hat{E}_{\Gamma_k}$  is the error indicator for the element  $\Gamma_k$ , while  $\xi_i$  and  $w_i$  are set of suitably chosen evaluation positions and weights on the element [4]. The evaluation positions and weights for flat, discontinuous 2D boundary elements and triangular and quadrilateral 3D elements were determined experimentally using correlation plots, and are shown in Table 1.

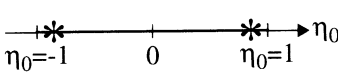
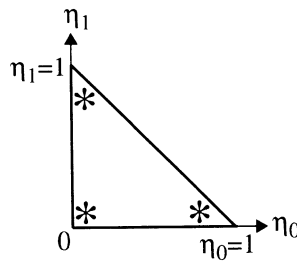
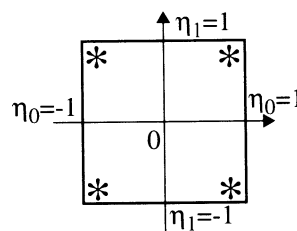
Reference Element Shape	Error indicator evaluation positions	Error indicator weights
<p>2D Line Element</p> 	<p>2 evaluation positions (indicated by * )</p> <p><math>\zeta_0 = -0.875</math>  <math>\zeta_1 = 0.875</math></p>	<p>Element length: <math>L</math></p> <p><math>\omega_{0,1} = \frac{L}{2}</math></p> <p>unit: [m]</p>
<p>3D Triangle Element</p> 	<p>3 evaluation positions (indicated by * )</p> <p><math>\zeta_0 = (0.062, 0.062)</math>  <math>\zeta_1 = (0.035, 0.929)</math>  <math>\zeta_2 = (0.929, 0.035)</math></p>	<p>Element area: <math>A</math></p> <p><math>\omega_{0,1,2} = \frac{A}{3}</math></p> <p>unit: [m<sup>2</sup>]</p>
<p>3D Quadrilateral Element</p> 	<p>4 evaluation positions (indicated by * )</p> <p><math>\zeta_0 = (-0.875, -0.875)</math>  <math>\zeta_1 = (-0.875, 0.875)</math>  <math>\zeta_2 = (0.875, -0.875)</math>  <math>\zeta_3 = (0.875, 0.875)</math></p>	<p>Element area: <math>A</math></p> <p><math>\omega_{0,1,2,3} = \frac{A}{4}</math></p> <p>unit: [m<sup>2</sup>]</p>

Table 1: Evaluation positions and the corresponding weights for computing the boundary element error indicator on element reference geometries.

### 3 Validation of the Error Indicator

The effectiveness of the 3D boundary element error indicator was verified on a typical MEMS geometry, see Figure 1. A reference solution was computed using cubic order elements and an extremely dense, uniform mesh. The correlation between the element error measure

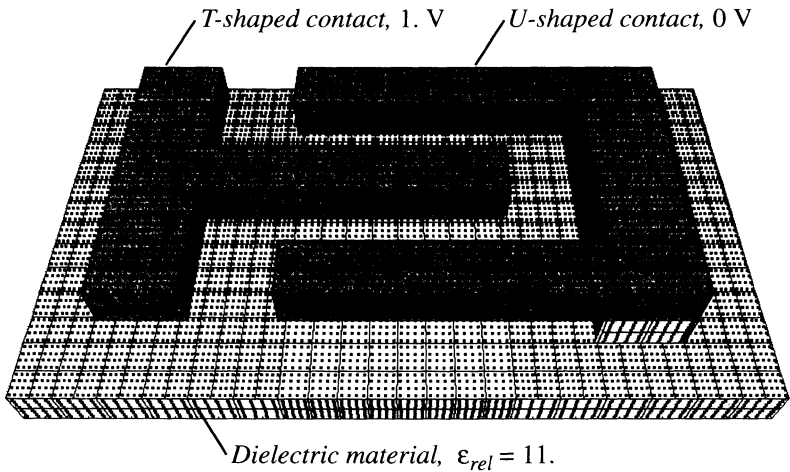
$$E_{\Gamma_k} = \int_{\Gamma_k} |q_{ref}(\mathbf{x}) - q(\mathbf{x})| d\Gamma(\mathbf{x}) \quad (5)$$

and the error indicator was evaluated for different discretizations. Here,  $q_{ref}(\mathbf{x})$  is the trusted reference solution. The error indicator quality increases with the solution accuracy, but even for a coarse mesh elements that contribute strongly

## 712 Boundary Elements

*Infinite domain,  $\epsilon_{rel} = 1$ .*

*Finger width:  $3\ \mu\text{m}$*



*Reference geometry for  
3D error indicator validation*

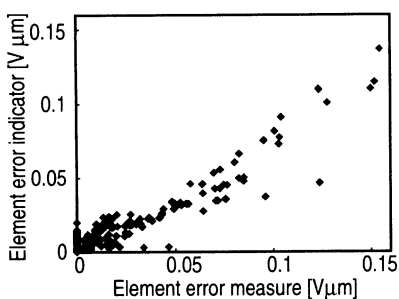
<i>Bicubic elements:</i>	<i>1560</i>	<i>Charge (T-contact): <math>1.36 \cdot 10^{-15}\ \text{C}</math></i>
<i>Collocation nodes:</i>	<i>24,960</i>	<i>Charge (U-contact): <math>-1.02 \cdot 10^{-15}\ \text{C}</math></i>
<i>Degrees of freedom:</i>	<i>32,256</i>	

Figure 1: Single finger pair of a MEMS comb drive used for validation of the element error indicator.

to the overall error are correctly identified, see Figure 2. Multipole (MP) techniques [4], [5], [6] accelerate the computation of the boundary integrals in equation (3), thereby allowing to compute the element error indicator in constant time, independent of the overall simulation size, see Figure 3.

## 4 Adaptive Refinement

Once the elements responsible for most of the error in the current solution have been identified using the error indicator, different refinement strategies are possible [3], [4], [7], [8]: the elements are split anisotropically into several, smaller elements (h-type refinement), or the number of nodes on the element is increased, together with the order of the element's interpolation functions (p-type refinement). H-type refinement not only increases the density of collocation nodes on a specific boundary segment, it also allows improving the element



*Coarse mesh:*

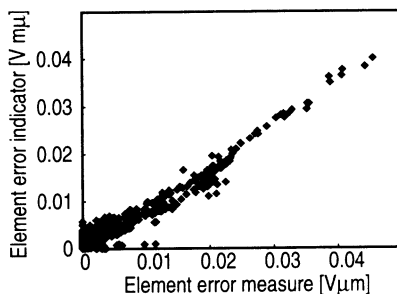
390 constant elements

390 collocation nodes

Area per node: 12.9  $\mu\text{m}^2$

Overall error measure: 8.73  $\text{V}\mu\text{m}$

Overall error indicator: 6.93  $\text{V}\mu\text{m}$



*Fine mesh:*

1560 constant elements

1560 collocation nodes

Area per node: 2.97  $\mu\text{m}^2$

Overall error measure: 6.96  $\text{V}\mu\text{m}$

Overall error indicator: 7.07  $\text{V}\mu\text{m}$

Figure 2: Correlation between the element error indicator and the element error measure for a coarse mesh using constant elements, and a fine mesh using discontinuous bilinear order elements.

### 3D Element error indicator computation

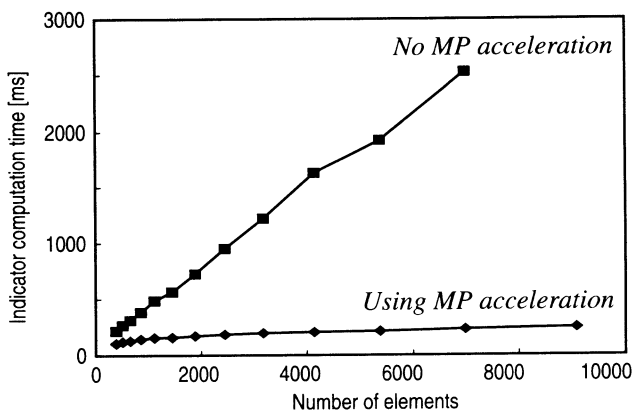


Figure 3: Average computational cost for evaluating the error indicator in a 3D boundary element on a SPARC workstation, with and without using multiple acceleration of the boundary integrals.

shapes, i.e. to break up very large elements, or to split elements with high aspect ratios. Large element aspect ratio values are undesirable from the numerical point of view, causing strong distortion of the element's shape functions and reducing the accuracy of the numerical quadrature formula used for boundary

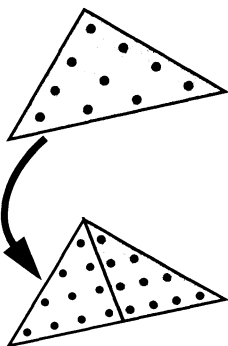
## 714 Boundary Elements

integration. The refinement process can be repeated iteratively and indefinitely to further improve mesh quality and simulation accuracy.

The current implementation supports arbitrary h-type refinement and p-type refinement up to cubic order for 2D elements. For 3D elements, the maximum shape function order supported for triangular elements is cubic, each element containing 10 internal collocation nodes. Quadrilateral elements are implemented up to bicubic order, containing 16 nodes. H-type refinement splits the elements, such that optimal element shape is attained for the child elements, in the sense that element aspect ratios are improved and the internal collocation nodes are more evenly distributed, as shown in Figure 4.

*Cubic triangular element*

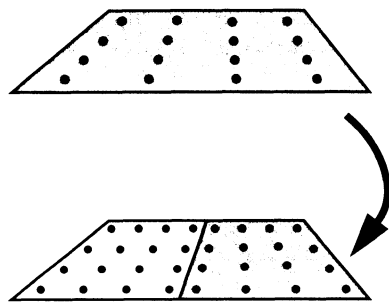
*10 internal collocation nodes*



*H-type refinement splits along longest side of a triangular element.*

*Bicubic quadrilateral element*

*16 internal collocation nodes*



*H-type refinement splits element such that the child elements have improved aspect ratio and more evenly distributed collocation nodes.*

Figure 4: 3D boundary elements are split anisotropically during h-type refinement such that child elements with good shape are created.

The accuracy of a solution obtained for a particular discretization is given by the overall error measure  $E_{\Gamma}$  which is computed by adding up the individual element error measures given by equation (5):

$$E_{\Gamma} = \int_{\Gamma} |q_{ref}(\mathbf{x}) - q(\mathbf{x})| d\Gamma(\mathbf{x}) = \sum_k E_{\Gamma_k} \quad (6)$$

Repeated steps of adaptive refinement lead to a heterogeneous mesh containing elements of various order and size. The 3D geometry used for indicator validation (Figure 1) was meshed using 7 automatic adaptive refinement iterations, resulting in an anisotropic mesh containing elements of various order based on a target overall error measure of 35 V $\mu$ m (Figure 5). Adaptive meshing using the

boundary element error indicator strongly improves solution convergence in respect to uniform mesh refinement, as shown in Figure 6. As a consequence, the adaptively refined mesh that satisfies a given accuracy goal is much smaller than an equivalent uniformly refined mesh, resulting in faster computations.

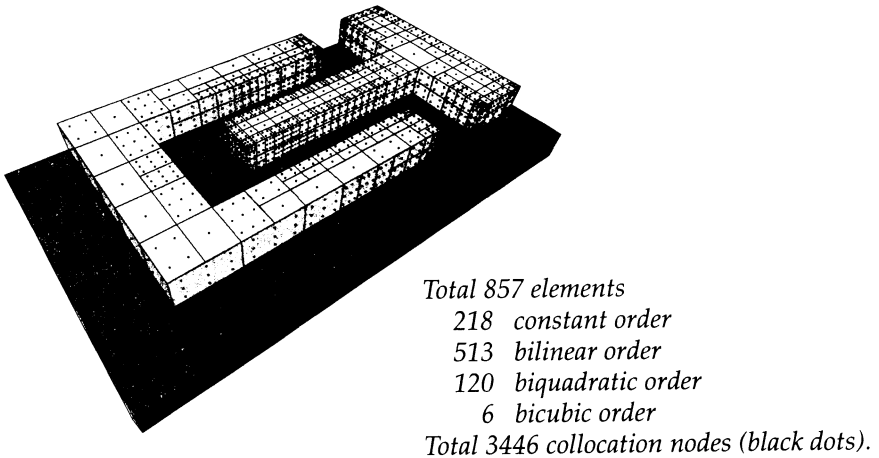


Figure 5: Mesh generated for the 3D error indicator validation geometry using 7 steps of automatic adaptive refinement.

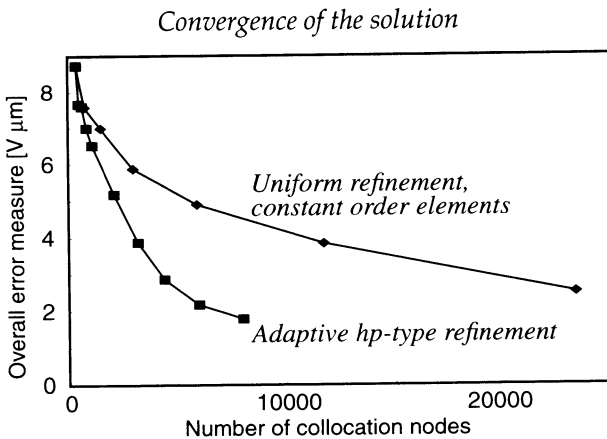


Figure 6: Solution convergence using adaptive or uniform meshing, measured using the overall error measure defined in equation

## 5 Results

Error estimation and automatic adaptivity is easily applied to large, complex simulations. An adaptive, anisotropic mesh was constructed for a 3D MEMS accelerometer using the following procedure: The original geometry was designed in an interactive solid modeling program. Note that the central electrode, which serves as a seismic mass, is tilted by an angle of 4.6 degrees, thereby preventing the exploitation of symmetry planes in the simulation.

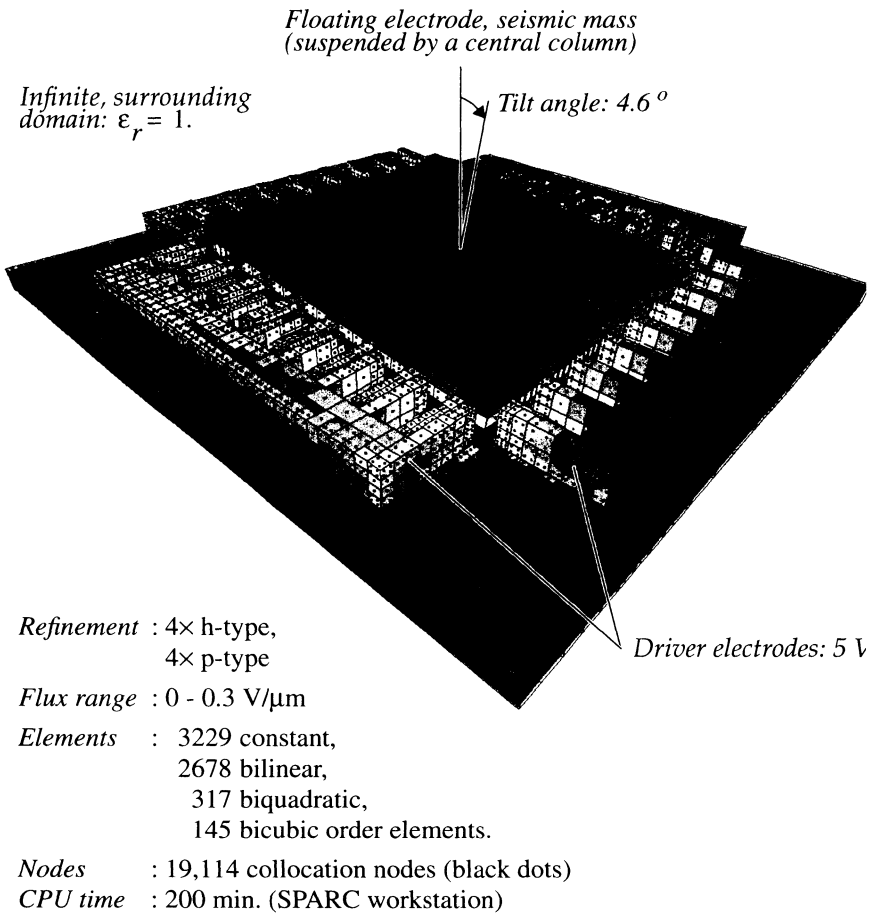


Figure 7: Electric flux on an electrostatic 3D MEMS accelerometer. Adaptively refined mesh and distribution of the BEM elements and collocation nodes.



The panels delivered by the solid modeler were subjected to a series of h-type refinements based on panel size and aspect ratio to produce the initial mesh for automatic adaptive refinement. The maximum element size for the initial mesh is  $500 \mu\text{m}^2$ , the maximum aspect ratio 5. The adaptive refinement process does not require any user intervention during the mesh construction, apart from an acceptable target overall error indicator. The discretization and the distribution of the collocation nodes after four steps of automatic adaptive h-type refinement and another four steps of p-type refinement is shown together with the electric flux in Figure 7. All areas where careful meshing is important, such as on the electrode's comb tips where charges tend to concentrate, are automatically identified and refined. The specified target value for the overall error indicator allows controlling the accuracy of the computed solution to an arbitrary degree.

## 6 Conclusion

The presented method efficiently computes accurate capacitance and flux values in large, geometrically complex structures such as electrostatically driven microactuators and VLSI interconnects. The proposed element error indicator shows good correlation to the actual element error and allows controlling the accuracy of a computed solution. Multipole acceleration allows computing element error indicators in  $O(\log N)$  time. H-type and p-type refinement steps are combined to produce a heterogeneous mesh containing elements of various shape size and order. The generated adaptive mesh allows calculating solutions with much higher accuracy than would be possible on a uniform mesh with the same computational effort. The presented approach was demonstrated to work for engineering applications from the area of electrostatic microactuator simulation.

## References

- [1] Brebbia, C.A., Telles, J.C.F., Wrobel, L.C., *Boundary Element Techniques*, Springer Verlag, Berlin and New York, 1984
- [2] Kamita, N., Koide, M., Aikawa, Y., Kawaguchi, K., Adaptive Boundary Element Scheme by Sample Point Error Analysis and its Extension, *BEM XVI*, Southampton, 1994, pp. 239-247
- [3] Bächtold, M., Korvink, J.G., Baltes, H., Automatic Adaptive Meshing for Efficient Boundary Element Simulations, in *SISPAD/96*, pp. 127 to 128, *Proceedings of the 1996 Int. Conf. on Simulation of Semiconductor Processes and Devices*, Tokyo, Japan, 1996



## 718 Boundary Elements

- [4] Bächtold, M., *Efficient 3D Computation of Electrostatic Fields and Forces in Microsystems*, Ph.D. Thesis, ETH Zurich, Physical Electronics Lab., Zurich 1997
- [5] Nabors, K., White, J., FastCap: A multipole accelerated 3D capacitance extraction program, *IEEE Trans. on CAD of Integrated Circuits and Systems*, 1991, **10**:10, 1447-1459
- [6] Bächtold, M., Korvink, J.G., Baltes, H., Enhanced Multipole Acceleration Technique for the Solution of Large Poisson Computations, *IEEE Trans. on CAD of Integrated Circuits and Systems*, 1996, **15**:12, 1541-1546
- [7] Guiggian, M., Error Indicators for Adaptive Mesh Refinement in the Boundary Element Methods - a new Approach, *Int. J. Numer. Meth. Engng.*, 1990, **29**, 1247-1269
- [8] Rank, E., Adaptive h-, p- and hp-versions for boundary integral methods, *Int. J. Numer. Meth. Engng.*, 1989, **28**, 1335-1349