ARCHIVES OF ACOUSTICS
31, 2, 193–212 (2006)

FEM AND BEM COMPUTING COSTS FOR ACOUSTICAL PROBLEMS

R. BOLEJKO, A. DOBRUCKI

Wrocław University of Technology
Institute of Telecommunication Teleinformatics and Acoustics
Wybrzeże Wyspiańskiego 27, 50-370 Wrocław, Poland
e-mail: Romuald.Bolejko@pwr.wroc.pl

(*received September 30, 2005; accepted March 1, 2006*)

FEM and BEM computing costs are compared for acoustical problems. The cost analysis was carried out for bounded areas of simple shapes for objects with acoustical losses (e.g. with sound absorbing materials). BEM's variational-collocative scheme (DBEM) and its variational scheme (IBEM) were considered. Computing costs were calculated, taking into account main matrix composition costs and main system of equations solution costs. The costs were calculated for the type of adopted discrete elements and the order of quadrature used. Analytical relations for calculating main matrix composition costs for BEM have been derived.

The analysis shows that FEM computing costs can be lower than BEM computing costs. Moreover, BEM computing costs are strongly dependent on the order of the quadrature used.

The presented results provide a basis for the choice of the most cost-effective method depending on the size of an acoustical problem.

Key words: computing costs, FEM, BEM, efficiency of modelling, acoustic field.

1. Introduction

The finite element method (FEM) and the boundary element method (BEM) are used for the analysis of a wide range of acoustical problems. In acoustics, BEM's variational-collocative scheme (DBEM – Direct BEM) and its variational scheme (IBEM – Indirect BEM) are most commonly applied [1]. FEM and BEM enable the modelling of an acoustic field in areas with practically any geometry and any boundary conditions. Each of the methods has its own modelling effectiveness which may vary considerably, depending on the acoustical problem. One of the factors having a major bearing on the effectiveness is the cost of computing. Computing costs, besides the computing power, determine computing speed. Therefore in this paper the terms "computing costs" and "computing speed" are used interchangeably. In order to be optimised, acoustical systems usually must be repeatedly modelled. Hence it is essential to compare the different methods and formulate rules for selecting the method with the lowest computing costs.

A comparison of computing costs for FEM and BEM can be found in [2, 3] but the analysis is limited to lossless models, without taking into account the differences between DBEM and IBEM and the computing costs involved in composing a system of algebraic equations. It is shown here that the latter costs significantly contribute to BEM's total computing costs.

When an acoustic field is modelled by FEM and BEM, the considered continuous area is represented in a discrete form by finite or boundary elements. In FEM, the entire modelled area is discretized, whereas in BEM only the area's boundary is discretized (Fig. 1). Hence a discrete model in FEM consists of a larger number of elements than in BEM.

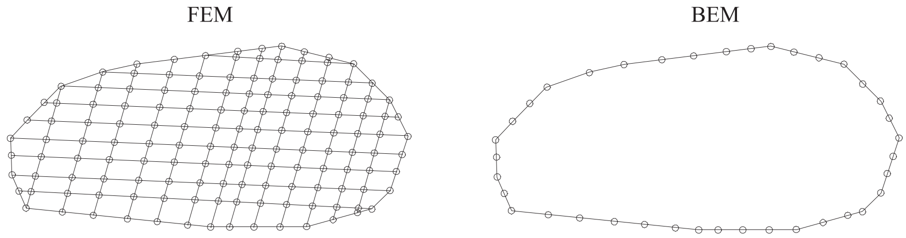


Fig. 1. Discretization of two-dimensional area modelled by respectively FEM and BEM.

For this reason it is generally believed that BEM computing costs are lower than those of FEM. But it is demonstrated here that for internal problems, the FEM computing costs may be much lower than those of BEM.

In FEM and BEM each vibroacoustic problem reduces itself to the following system of algebraic equations:

$$\mathbf{A}\mathbf{x} = \mathbf{b}. \tag{1}$$

Matrix **A** is the main matrix, **x** is a vector (a single-column matrix) of unknowns, vector **b** represents external excitations acting on the investigated system. Since systems with losses will be considered, matrix **A** will always be regarded as a matrix with complex terms.

The computing costs involved in the solution of system of Eq. (1) mainly depend on the properties of the main matrix. Thus the analysed numerical methods can be characterized through the properties of this matrix.

Computing costs are defined by the number of mathematical operations needed to solve a given problem and the number of additional operations involved in, for example, data transfer within the computer's memory. In FEM and BEM the number of additional operations is generally much lower than the number of mathematical operations and so it can be neglected in the total computing costs [3]. The number of mathematical operations can be measured in units called $flop$ (floating point operations). It is assumed here that a unit computing operation consists of one multiplication operation and one addition operation: $flop = z_1 + z_2 z_3$. Hence for complex number arithmetic, the number of mathematical operations performed in the course of solving system of Eq. (1) is four times larger than in the case of real number arithmetic.

System of Eq. (1) can be solved by direct methods, e.g. the Gaussian elimination method, or iterative methods. In practice, however, usually direct methods are used since they are easily implemented and numerically stable. Iterative methods are mainly used for the analysis of very large systems in the case of which their computing costs may be lower than those of direct methods. The computing costs of direct methods can be quite easily calculated. The computing costs of direct methods rise quickly with the modelled problem's number of degrees of freedom. The considerations in this paper are limited to direct methods.

System of Eq. (1) with non-singular matrix $\mathbf{A} \in C^{N,N}$ is solved in two steps by a direct method. First, the matrix **A** is factorised, i.e. represented in the form of a product of triangular matrices: lower triangular matrix **L** and upper triangular matrix **U**. In the second step, forward/backward elimination operations are performed [4]. The method's costs are determined taking into account the costs of its steps:

$$\begin{aligned} \mathbf{A} &= \mathbf{LU} && 4N^3/3\,flop, \\ \mathbf{Ly} &= \mathbf{b} && 2N^2\,flop, \\ \mathbf{Ux} &= \mathbf{y} && 2N^2\,flop. \end{aligned} \tag{2}$$

In order to factorise matrix **A** with complex terms, $4N^3/3$ mathematical operations, where N is the discrete model's number of degrees of freedom, must be performed. Forward/backward elimination requires a total of $4N^2$ mathematical operations.

Main matrix **A** has often certain properties, such as symmetry, banding and positive definiteness, which make it possible to employ faster factorisation and forward/backward elimination algorithms than the classic Gaussian algorithm.

Cholesky's algorithm, which reduces by half, in comparison with factorisation **LU** [4], the computing costs, is used for the factorisation of a positive defined symmetric matrix. Unfortunately, since matrix **A** is usually a complex matrix, it is not a positive definite matrix. For this reason instead of the classic Cholesky algorithm, its modified version [5] or factorisation $\mathbf{U}^{\mathrm{T}}\mathbf{DU}$ [4] is used:

$$\mathbf{A} = \mathbf{U}^{\mathrm{T}}\mathbf{DU} \qquad 2N^3/3\,flop, \tag*{(3)$_1$}$$

where matrix **D** is a diagonal matrix with elements different from zero,

$$\begin{aligned} \mathbf{U}^{\mathrm{T}}\mathbf{z} &= \mathbf{b} && 2N^2 flop, \\ \mathbf{Dy} &= \mathbf{z} && 4N\,flop, \\ \mathbf{Ux} &= \mathbf{y} && 2N^2 flop. \end{aligned} \tag*{(3)$_{2-4}$}$$

The number of mathematical operations performed during factorisation is directly proportional to the third power of the number of equations (nodes in the discrete mesh), whereas the number of operations performed in the course of forward/backward elimination is proportional to the second power of the number of equations. In the case of models with a number of nodes larger than a few dozen, the cost of factorisation of matrix **A** determines the total computing costs.

The number of mathematical operations performed while solving the system of Eq. (1) may be 1.5–2 times larger than the one specified, since there are additional operations involved in the manipulation of the main matrix's elements during the selection of elements along the main diagonal aimed at obtaining the required accuracy of the mathematical operations [4].

2. FEM and BEM computing costs

Computing time in FEM and BEM is made up of the times needed for: system of equations composition, main matrix factorisation and forward/backward elimination. Since the stiffness, mass and damping matrices in FEM do not depend on frequency, they are composed only once at the beginning of the computations. In BEM, the main matrix depends on frequency and so the main matrix is composed separately for each frequency. An analysis of the actual system is generally made for a relatively wide frequency range and so the time allocated for the composition of a system of equations in FEM can be neglected when comparing the computing costs of FEM and BEM.

The main matrix in FEM is symmetric and banded. Factorisation $\mathbf{U}^{\mathrm{T}}\mathbf{DU}$ or a modified Cholesky algorithm is usually applied to the matrix. The number of mathematical operations performed in the course of factorisation and forward/backward elimination for matrix $\mathbf{A}$ with complex terms is [4]:

$$\begin{aligned} \mathbf{A} = \mathbf{U}^{\mathrm{T}}\mathbf{DU} \qquad & 2b_w^2 N\mathit{flop}, \\ \mathbf{U}^{\mathrm{T}}\mathbf{z} = \mathbf{b} \qquad & 4b_w N\mathit{flop}, \\ \mathbf{Dy} = \mathbf{z} \qquad & 4N\mathit{flop}, \\ \mathbf{Ux} = \mathbf{y} \qquad & 4b_w N\mathit{flop}, \end{aligned} \tag{4}$$

where b_w is the mean value of the half-band width of the main matrix.

The above formulas (4) are true for $N \gg 1$ and $N \gg b_w$, which usually is the case when real systems are modelled by FEM.

According to formulas (4), the banding of matrix $\mathbf{A}$ greatly affects the computing speed. The matrix band width depends on the method of numbering the finite elements and it is determined by the maximum difference between the numbers of nodes connected with the same element. Main matrix half-band width b_w for an optimally defined mesh of finite elements is by one order smaller than the number of nodes N_w [3]. Using the above relation, Eq. (4) can be simplified by eliminating parameter b_w. In the case of systems with a very complicated geometry, the relation between b_w and N_w does not always hold; nevertheless, it holds for many real models.

In BEM, computing speed depends on the costs of solving system of Eq. (1), the main matrix composition costs and the costs of computing the acoustic parameters in observation points. The acoustic parameters in observations points are determined on the basis of the calculated pressure and the acoustic velocity distributions on the boundary element mesh. The associated costs depend on the number of observation points and they are generally much lower than the costs of solving the main system of equations and the costs of composing the main matrix. In this paper, the costs of computing acoustic parameters in observation points are neglected.

In BEM, it is necessary to perform integration on all the elements of the discrete mesh. Hence one must calculate integrals in the form [1]:

$$\begin{aligned} I_{i,j}^{(DBEM)}(r_i) &= \int\limits_{S_j} \left(\aleph\left(r_j\right) J\left(r_i, r_j\right)\right) \mathrm{d}S(r_j) \\ I_{i,j}^{(IBEM)}(r_i, r_j) &= \int\limits_{S_i} \int\limits_{S_j} \left(\aleph\left(r_i\right) \aleph\left(r_j\right) J\left(r_i, r_j\right)\right) \mathrm{d}S(r_i) \mathrm{d}S(r_j), \end{aligned} \tag{5}$$

where r_i is a point belonging to element e_i, r_j is a point belonging to element e_j, $\aleph(r)$ is a shape function.

$J(r_i, r_j)$ is a Green function or its first derivative in DBEM or a Green function or its first or second derivative in IBEM:

$$J\left(r_i, r_j\right) = G\left(r_i, r_j\right) \text{ or } J\left(r_i, r_j\right) = \frac{\partial G\left(r_i, r_j\right)}{\partial n_j} \text{ or } J\left(r_i, r_j\right) = \frac{\partial^2 G\left(r_i, r_j\right)}{\partial n_i \partial n_j}. \tag{6}$$

The method of calculating numerical integrals, the so-called quadrature, greatly affects the speed of main matrix composition. In BEM, Gaussian quadrature is usually used.

Once the integrals (5) in nodes j belonging to the element e_j on which integration is performed are determined, the integrated function is singular because of the properties of Green function. The singularity can be eliminated by replacing the coordinate system in which integration takes place with polar coordinates or by other techniques [6]. Each of the above mathematical operations increases the method's computing costs. The number of points at which singularity occurs mainly depends on the number of nodes belonging to the element.

The total cost may differ significantly between DBEM and IBEM. In IBEM, integration is performed on the area of a pair of elements twice in the course of determining the terms of the main matrix, whereas in DBEM it is performed only once on an area of a single element. The composition of the main matrix in DBEM entails the determination of N_w^2 of individual integrals $(5)_1$, whereas in IBEM about $N_e^2/2$ double integrals $(5)_2$ must be determined since a pair of elements occurs under the shape function, where N_w is the number of nodes and N_e – the number of the discrete elements. Because of the algorithms used, composition costs in DBEM and IBEM should be estimated depending on, respectively, the number of nodes and the number of discrete elements. The composition of the main matrix in IBEM is on the whole more time-consuming than in DBEM.

From an analysis of the structure of integrals (5) and system of Eq. (1) (in the discrete form for BEM [6]), one can derive some simplified relations for the number of mathematical operations performed during the composition of the main matrix:

$$\begin{aligned} K_K^{(DBEM)} &= C_0^{(DBEM)} C_1 N_w^2, \\ K_K^{(IBEM)} &= C_0^{(IBEM)} C_1^2 C_2^2 N_e^2, \end{aligned} \tag{7}$$

where C_0 – a coefficient dependent on the actual computer implementation, C_1 – the number of nodes in the quadrature applied, C_2 – a coefficient dependent on the boundary conditions.

Coefficient C_0 depends on the computer implementation of the method, among others on the algorithm for eliminating the Green function's indefiniteness. For a given implementation, a value of the coefficient can be assigned to particular types of boundary elements: one-dimensional linear, one-dimensional parabolic, two-dimensional tetragonal linear, etc.

In the case of IBEM, the cost of main matrix composition also depends on the kind of boundary conditions. Computations are the most time-consuming for Neumann boundary conditions and the least time-consuming for Dirichlet boundary conditions [7]. The equation for the terms of the main matrix in a model with Neumann boundary conditions includes four times more terms than in a model with Dirichlet boundary conditions [6]. Thus the cost of determining the particular terms for Neumann boundary conditions is about four times higher than that for Dirichlet boundary conditions. Coefficient C_2, whose value ranges from 0.5 to 1, takes the above into account. Coefficient C_2 for a model with exclusively Neumann boundary conditions or exclusively with Dirichlet boundary conditions is equal respectively to unity ($C_2 = 1$) and one half ($C_2 = 0.5$). Neuman boundary conditions occur in most practical vibroacoustic problems. Therefore the most time-consuming case – a model with Neumann boundary conditions ($C_2 = 1$) – is considered in the analysis of IBEM computing costs.

Prior to a comparison of computing costs for FEM and BEM, formulas (7) were verified for a two-dimensional model of a rectangular chamber. The results of the numerical experiment are shown in Figs. 2 and 3.

The results of modelling confirm that the derived relations (7) are correct.

In DBEM, matrix **A** is an asymmetric, fully populated matrix with complex terms. Factorisation **LU** is applied to the matrix. The number of mathematical operations performed on complex numbers is $4N_w^3/3$ and $4N_w^2$ for factorisation and elimination respectively.

In IBEM, matrix **A** is a fully populated and symmetric matrix with complex terms. Factorisation $\mathbf{U}^{\mathrm{T}}\mathbf{DU}$ is applied to the matrix. The number of mathematical operations performed on complex numbers is $2N_w^3/3$ and $4N_w^2$ for factorisation and elimination respectively.

Factorisation is twice faster in IBEM than in DBEM. The costs of forward/backward elimination are the same for DBEM and IBEM.

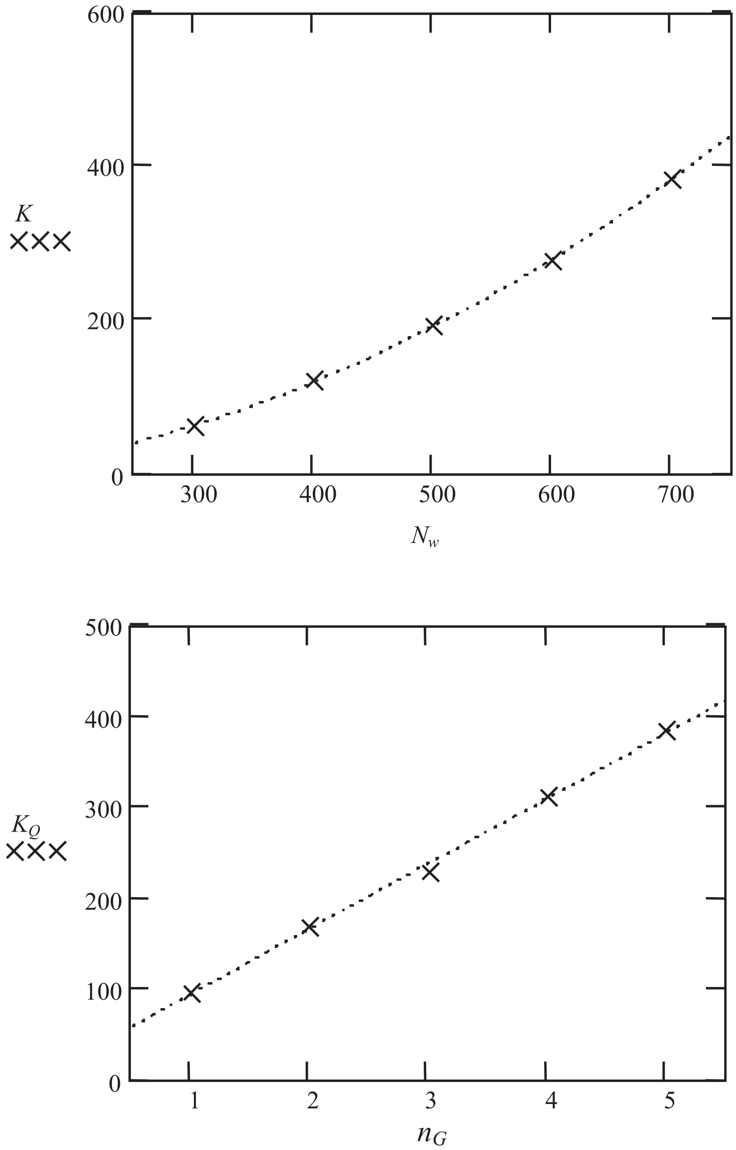
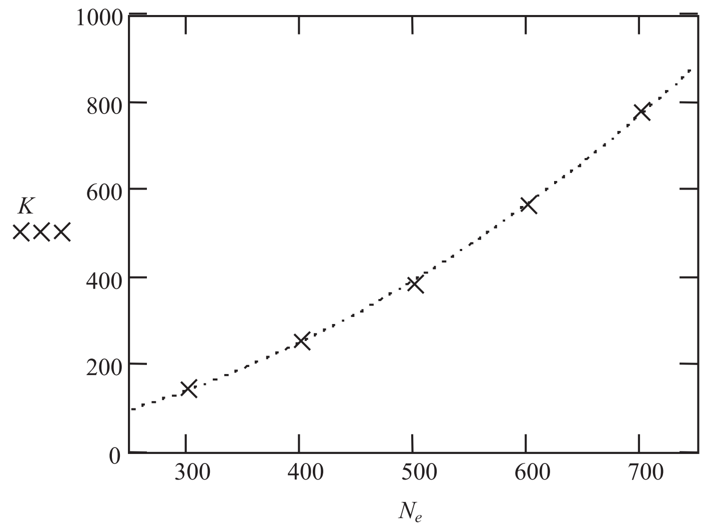


Fig. 2. Main matrix composition costs for DBEM depending on the number of nodes N_w and number of Gaussian quadrature nodes n_G in comparison with theoretically determined curve.

The speed of matrix composition is significantly affected by the number of nodes in the quadrature used (parameter C_1). The costs of main matrix composition in DBEM and IBEM increase, respectively linearly and with the square, with the number of quadrature nodes.



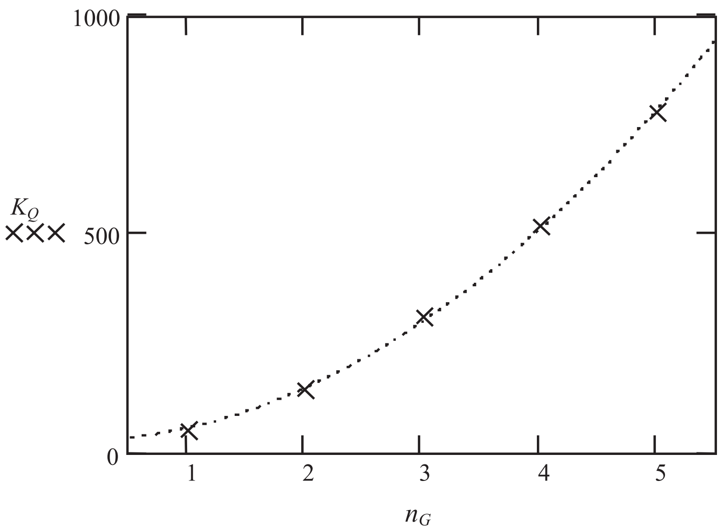


Fig. 3. Main matrix composition costs for IBEM method depending on the number of elements N_e and number of Gaussian quadrature nodes n_G in comparison with theoretically determined curve (broken line).

3. Coupled BEM/FEM and FEM/FEM computing costs

Coupled BEM/FEM and FEM/FEM are used for the analysis of vibroacoustic problems which takes interaction between the structure and fluid into account. One of the methods in a couple is used for modelling the acoustic field while the other one is used for modelling the vibrations of the elastic structure. In BEM/FEM method, the FEM generally is used for modelling the elastic structure while the BEM – for modelling the fluid.

In DBEM/FEM, system of Eqs. (1) assumes the following form [6]:

$$\begin{bmatrix} \mathbf{K}_s + j\omega\mathbf{C}_s - \omega^2\mathbf{M}_s & \mathbf{S}_{\text{DBEM}}^{\text{T}} \\ \rho_0\omega^2\mathbf{E} & \mathbf{B} \end{bmatrix} \begin{Bmatrix} \mathbf{u} \\ \mathbf{p} \end{Bmatrix} = \begin{Bmatrix} \mathbf{F}_s \\ \mathbf{\Lambda}_a \end{Bmatrix}, \tag{8}$$

where $\mathbf{K}_s$, $\mathbf{C}_s$, $\mathbf{M}_s$ – stiffness, damping and mass matrices in an FEM model of the elastic structure, $\mathbf{B}, \mathbf{E}$ – coefficient matrices of DBEM, $\mathbf{S}_{\text{DBEM}}$ – a matrix defining the coupling between the acoustic field and the elastic structure, $\mathbf{u}$, $\mathbf{p}$ – vectors of displacement and acoustic pressure, $\mathbf{F}_s$ – vector of external mechanical force, $\mathbf{\Lambda}_a$ – vector of external acoustic load, ρ_0, ω – the density of the fluid and circular frequency.

A coupled model's number of degrees of freedom is a sum of the degrees of freedom of the acoustic model and the degrees of freedom of the elastic structure. Since different matrices $\mathbf{S}_{\text{DBEM}}$ and $\mathbf{E}$ are situated symmetrically relative to the main diagonal and matrices $\mathbf{B}$, $\mathbf{E}$, $\mathbf{S}_{\text{DBEM}}$ are fully populated and asymmetric matrices, the main matrix of system of Eqs. (8) is also fully populated and asymmetric. Thus the main matrix in DBEM/FEM has the same properties as in DBEM, whereas the model's number of degrees of freedom increases by the number of degrees of freedom associated with the elastic structure.

In IBEM/FEM, system of Eqs. (1) assumes the following form [6]:

$$\begin{bmatrix} \mathbf{K}_s + j\omega\mathbf{C}_s - \omega^2\mathbf{M}_s & \mathbf{G}^{\text{T}} \\ \mathbf{G} & \mathbf{H}/\rho\omega^2 \end{bmatrix} \begin{Bmatrix} \mathbf{u} \\ \boldsymbol{\mu} \end{Bmatrix} = \begin{Bmatrix} \mathbf{F}_s \\ \mathbf{\Lambda}_a \end{Bmatrix}, \tag{9}$$

where $\mathbf{G}, \mathbf{H}$ – coefficient matrices of IBEM, $\boldsymbol{\mu}$ – the vector of double-layer acoustic potential.

Matrices $\mathbf{K}_s$, $\mathbf{C}_s$, $\mathbf{M}_s$ are symmetric banded matrices, $\mathbf{H}$ is a symmetric fully populated matrix, $\mathbf{G}$ is a fully populated matrix. Thus the main matrix in IBEM/FEM is fully populated and symmetric. The number of degrees of freedom of the coupled IBEM/FEM model is a sum of the degrees of freedom of the acoustic model and those of the elastic structure model, whereas the main matrix has the same properties as the matrix in IBEM.

Since the main matrices of DBEM/FEM and DBEM have identical properties as well as the main matrices of IBEM/FEM and IBEM have identical properties, the comparison of the coupled methods will yield the same results as the comparison of the uncoupled DBEM and IBEM.

In the case of classic FEM/FEM, system of Eqs. (1) is defined as [8]:

$$\begin{bmatrix} \mathbf{K}_s + j\omega\mathbf{C}_s - \omega^2\mathbf{M}_s & -\rho^{-1}\mathbf{S}_{\text{FEM}}^{\text{T}} \\ -\omega^2\mathbf{S}_{\text{FEM}} & \mathbf{K}_a + j\omega\mathbf{C}_a - \omega^2\mathbf{M}_a \end{bmatrix} \begin{Bmatrix} \mathbf{u} \\ \mathbf{p} \end{Bmatrix} = \begin{Bmatrix} \mathbf{F}_s \\ \mathbf{\Lambda}_a \end{Bmatrix}, \tag{10}$$

$\mathbf{K}_a$, $\mathbf{C}_a$, $\mathbf{M}_a$ – stiffness, damping and mass matrices in the acoustic model, $\mathbf{S}_{\text{FEM}}$ – a matrix defining the coupling between the acoustic field and the elastic structure.

The main matrix in FEM/FEM is an asymmetric matrix and it needs large computational memory. For this reason FEM/FEM in the classic form is seldom used in practical computer applications for the analysis of coupling vibration and acoustic field. There are several ways in which the main matrix in FEM/FEM can be made symmetric again [8–10], but all of them are applicable to only a small class of finite elements or require the definition of an additional variable in the form of an acoustic displacement potential or an acoustic velocity potential. As a result of using additional variable, the number of degrees of freedom in a discrete model usually doubles.

Since there are many techniques of restoring the main matrix symmetry in coupled FEM/FEM, BEM/FEM and FEM/FEM should be compared separately for each of the techniques, which goes beyond the paper's scope. In the case of techniques which do not use an additional variable in the form of an acoustic displacement or velocity potential but restore the main matrix symmetry [9], the results of a comparison of BEM/FEM and FEM/FEM will be identical as the ones obtained for uncoupled BEM and FEM.

4. Comparison of FEM and BEM computing costs

A comparison of computing costs and computational memory between the methods for two-dimensional rectangular and three-dimensional parallelepiped models (Fig. 4) was made.

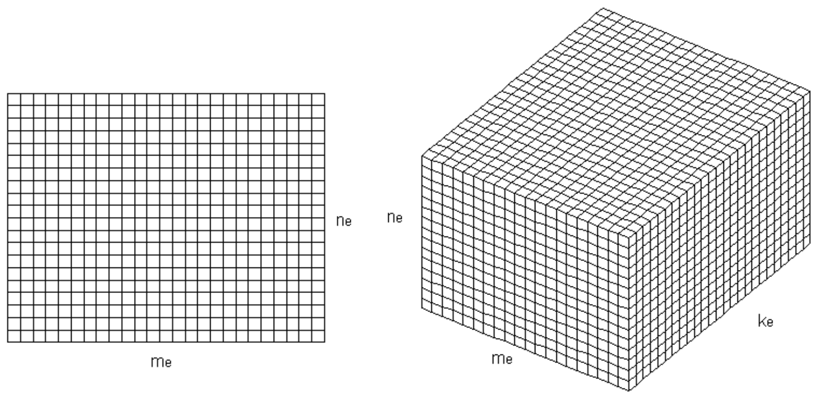


Fig. 4. Shape of analysed two- and three-dimensional areas.

Parameters m_e, n_e and k_e specify the number of discrete elements per one dimension of the modelled area. DBEM, IBEM and FEM were compared. The analysis was carried out for models made up of linear and parabolic discrete elements. The methods were compared according to the total number of mathematical operations needed to solve the system of Eqs. (1) and the size of memory needed to store the main matrix. In the case of BEMs, the analysis was made for quadrature with a different number of nodes.

The basic parameters of the discrete meshes for a two-dimensional rectangular area and a three-dimensional parallelepiped area for the most common types of elements are shown in Tables 1 and 2. The formulas (in the tables) for the number of nodes are exact for meshes made up of at least a dozen or so discrete elements. The formulas can also be used for rough estimates in the analysis of computing costs for other than cuboidal chambers.

Table 1. Number of discrete mesh nodes and elements in BEM for two-dimensional rectangular and three-dimensional parallelepiped areas. Calculated values of coefficient C_0 for BEMs.

Area	Type of elements	Number of elements N_e	Number of nodes/ equations N_w	Coefficient C_0	
				DBEM	IBEM
$m_e \times n_e$		$2(n_e + m_e)$	$2(n_e + m_e)$	255	109
		$2(n_e + m_e)$	$4(n_e + m_e)$	158	126
$m_e \times n_e \times k_e$		$2(n_e m_e + n_e k_e + m_e k_e)$	$2(n_e m_e + n_e k_e + m_e k_e + 1)$ $\approx 2(n_e m_e + n_e k_e + m_e k_e)$	238	82
		$4(n_e m_e + n_e k_e + m_e k_e)$	$2(n_e m_e + n_e k_e + m_e k_e + 1)$ $\approx 2(n_e m_e + n_e k_e + m_e k_e)$	396	65
		$2(n_e m_e + n_e k_e + m_e k_e)$	$2(3n_e m_e + 3n_e k_e + 3m_e k_e + 1)$ $\approx 6(n_e m_e + n_e k_e + m_e k_e)$	138	216
		$4(n_e m_e + n_e k_e + m_e k_e)$	$2(4n_e m_e + 4n_e k_e + 4m_e k_e + 1)$ $\approx 8(n_e m_e + n_e k_e + m_e k_e)$	192	146

In the analysis of BEM computing costs, the number of mathematical operations performed during the composition of the main matrix is calculated from Eq. (7). For this purpose, the computer program Sysnoise v.5.2® for the FEM and BEM analysis of vibroacoustic problems [6] was employed to calculate coefficient C_0 from the following relations:

$$\begin{aligned} C_0^{(DBEM)} &= \frac{T^{(DBEM)} \times cpm}{C_1 N_w^2}, \\ C_0^{(IBEM)} &= \frac{T^{(IBEM)} \times cpm}{C_1^2 N_e^2}, \end{aligned} \tag{11}$$

where T – matrix composition time [s], cpm – computing power [flop].

The calculated values of coefficient C_0 are shown in Table 1.

The value of coefficient C_0 may depend on the actual implementation of the method and the algorithm employed for the elimination of singularities associated with the Green function. But the difference in the results of the comparison of DBEM and IBEM as well as FEM and DBEM/IBEM by other computing programs than Sysnoise should not be large. In the case of DBEM and IBEM, the obtained results can be easily rescaled to other values of coefficient C_0 [7].

Table 2. Number of discrete mesh nodes and elements in FEM for two-dimensional rectangular and three-dimensional parallelepiped areas.

Area	Type of elements	Number of elements N_e	Number of nodes/ equations N_w
$m_e \times n_e$		$n_e m_e$	$(n_e+1)(m_e+1)$ $\approx n_e m_e + n_e + m_e$
		$2n_e m_e$	$(n_e+1)(m_e+1)$ $\approx n_e m_e + n_e + m_e$
		$n_e m_e$	$(n_e+1)(2m_e+1)+n_e(m_e+1)$ $\approx 3n_e m_e+2(n_e + m_e)$
		$2n_e m_e$	$(n_e+1)(2m_e+1)+n_e(m_e+1)+n_e m_e$ $\approx 4n_e m_e+2(n_e + m_e)$
$m_e \times n_e \times k_e$		$n_e m_e k_e$	$(n_e+1)(m_e+1))(k_e+1)$ $\approx n_e m_e k_e + n_e m_e + n_e k_e + m_e k_e + n_e + m_e + k_e$
		$2n_e m_e k_e$	$(n_e+1)(m_e+1))(k_e+1)$ $\approx n_e m_e k_e + n_e m_e + n_e k_e + m_e k_e + n_e + m_e + k_e$
		$n_e m_e k_e$	$4n_e m_e k_e+3(n_e m_e + n_e k_e + m_e k_e)$ $+2(n_e + m_e + k_e)+1$ $\approx 4n_e m_e k_e+3(n_e m_e + n_e k_e + m_e k_e)+2(n_e + m_e + k_e)$
		$2n_e m_e k_e$	$5n_e m_e k_e+4n_e m_e+3(n_e k_e + m_e k_e)$ $+2(n_e + m_e + k_e)+1$ $\approx 5n_e m_e k_e+4n_e m_e+3(n_e k_e + m_e k_e)+2(n_e + m_e + k_e)$

When comparing the methods, a two-dimensional rectangular area and a three-dimensional area, having the shape of a cube divided into respectively $n_e \times n_e$ and $n_e \times n_e \times n_e$ elements, were analysed.

DBEM, IBEM and FEM were compared by determining the number of discrete mesh elements for which their computing costs are the same. Since the modelled area was represented by $n_e \times n_e$ (2D) or $n_e \times n_e \times n_e$ (3D) elements, one parameter n_e, which specifies the number of finite/boundary elements per one dimension of the modelled area, was used. In the case of triangular elements, parameter n_e stands for half the number of triangular finite/boundary elements per one dimension. The methods were compared by solving the following equations:

methods: IBEM - DBEM

$$(K_K + K_F + K_S)^{(IBEM)} = (K_K + K_F + K_S)^{(DBEM)},$$

methods: BEM - FEM

$$(K_K + K_F + K_S)^{(BEM)} = (K_F + K_S)^{(FEM)}, \tag{12}$$

where K_K, K_F, K_S – costs of respectively main matrix composition, factorization and forward/backward elimination for comparable methods.

Computing costs for meshes made up of the same elements in the case of BEM and for meshes consisting of elements having a similar shape in the case of BEM/FEM: one-dimensional linear element for BEM, two-dimensional linear quadrangular/triangular element for FEM, one-dimensional parabolic element for BEM, two-dimensional parabolic quadrangular/triangular element for FEM, two-dimensional linear quadrangular element for BEM, three-dimensional linear hexahedral element for FEM and so on, were analysed. Tables 3–5 show the results of the comparison of DBEM/IBEM and BEM/FEM.

In the case of BEM/FEM, besides the parameter n_e and total number of discrete elements N_e, also the computer memory $M^{(\mathrm{Method})}$ needed to store main matrix $\mathbf{A}$ is given. In the "Notes" column, a computing costs relationship between the considered methods for models made up of a larger number of elements than the one shown in column "N_e" is given. If instead of a numerical value, a cost relationship for comparable methods is shown in column "N_e", this means that for the considered quadrature the cost relationship does not depend on model size.

For the comparison of DBEM and IBEM the results are shown for quadrature with 1–5 nodes for 2D area and with 1–25 nodes for 3D area. For a comparison of BEM and FEM the results are shown for quadrature with 2–4 nodes for 2D area and with 4–16 nodes for 3D area. In the case of linear and parabolic elements, quadrature with a larger number of nodes is very seldom used in modelling and a cost analysis is then of no practical importance. Because of the different computing algorithms used in the Gaussian quadrature for triangular and quadrangular elements, the number of nodes for a quadrature with the same order is different for the elements.

Relationships between the computing costs and the number of linear elements per one model dimension n_e for 2D and 3D areas and quadrature with respectively two and four nodes are shown in Figs. 5 and 6.

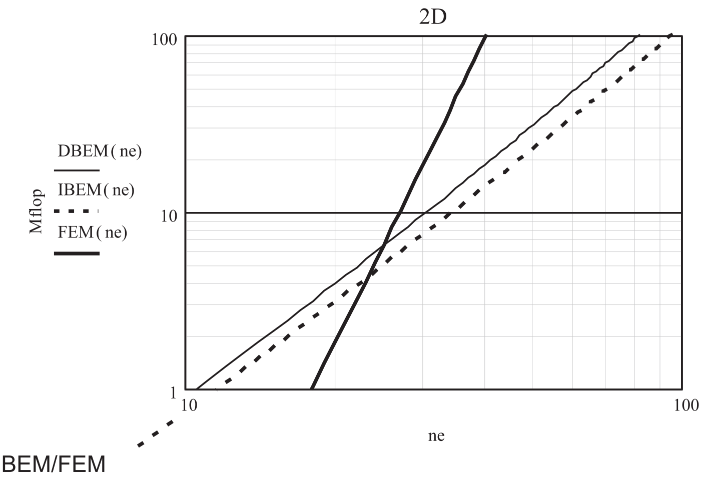


Fig. 5. Number of mathematical operations versus number of linear elements for two-dimensional area and two-node quadrature.

Table 3. Comparison of DBEM and IBEM computing costs.

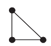
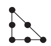
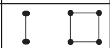
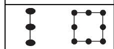
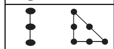
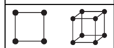
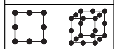
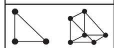
Area	Type of elements	n_e	N_e	n_e	N_e	n_e	N_e	n_e	N_e	n_e	N_e	Notes
2D		Number of quadrature nodes *nG*										
		1		2		3		4		5		
		$K^{(IBEM)} < K^{(DBEM)}$		$K^{(IBEM)} < K^{(DBEM)}$		82	328	276	1104	552	2208	$K^{(IBEM)} < K^{(DBEM)}$
		$K^{(IBEM)} < K^{(DBEM)}$		$K^{(IBEM)} < K^{(DBEM)}$		$K^{(IBEM)} < K^{(DBEM)}$		$K^{(IBEM)} < K^{(DBEM)}$		$K^{(IBEM)} < K^{(DBEM)}$		
3D		Number of quadrature nodes *nG*										
		1		4		9		16		25		
		$K^{(IBEM)} < K^{(DBEM)}$		10	600	34	6936	66	26136	106	67416	$K^{(IBEM)} < K^{(DBEM)}$
		$K^{(IBEM)} < K^{(DBEM)}$		$K^{(IBEM)} < K^{(DBEM)}$		8	384	18	1944	31	5766	
		Number of quadrature nodes *nG*										
		1		4		7		13		16		
		$K^{(IBEM)} < K^{(DBEM)}$		25	7500	50	30000	99	117612	123	181548	$K^{(IBEM)} < K^{(DBEM)}$
		$K^{(IBEM)} < K^{(DBEM)}$		$K^{(IBEM)} < K^{(DBEM)}$		5	300	15	2700	20	4800	

Table 4. Comparison of DBEM and IBEM computing costs.

Area	Type of elements	n_e	BEM N_e	$M^{(BEM)}$ [kB]	FEM N_e	$M^{(FEM)}$ [kB]	n_e	BEM N_e	$M^{(BEM)}$ [kB]	FEM N_e	$M^{(FEM)}$ [kB]	n_e	BEM N_e	$M^{(BEM)}$ [kB]	FEM N_e	$M^{(FEM)}$ [kB]	Notes
		Number of quadrature nodes *nG*															
		2					3					4					
2D		22	88	61	484	218	24	96	72	576	304	26	104	85	676	414	$K^{(DBEM)} < K^{(FEM)}$
		14	56	98	196	325	15	60	113	225	423	16	64	128	256	542	
		11	44	61	242	219	12	48	72	288	305	13	52	85	338	415	
3D		Number of quadrature nodes *nG*															
		4					9					16					
				[MB]		[MB]			[MB]		[MB]			[MB]		[MB]	
		23	3174	77	12167	146	24	3456	91	13824	186	26	4056	126	17576	296	$K^{(DBEM)} < K^{(FEM)}$
		17	1734	207	4913	381	18	1944	260	5832	530	19	2166	322	6859	724	
		Number of quadrature nodes *nG*															
		4					7					13					
		24	6912	91	27648	186	25	7500	107	31250	236	27	8748	146	39366	368	$K^{(DBEM)} < K^{(FEM)}$
		18	3888	462	11664	806	18	3888	462	11664	806	19	4332	573	13718	1103	

Table 5. Comparison of DBEM and IBEM computing costs.

Area	Type of elements	n_e	BEM N_e	$M^{(BEM)}$ [kB]	FEM N_e	$M^{(FEM)}$ [kB]	n_e	BEM N_e	$M^{(BEM)}$ [kB]	FEM N_e	$M^{(FEM)}$ [kB]	n_e	BEM N_e	$M^{(BEM)}$ [kB]	FEM N_e	$M^{(FEM)}$ [kB]	Notes
		Number of quadrature nodes *nG*															
		2					3					4					
2D		23	92	33	529	258	29	116	53	841	631	33	132	68	1089	1042	$K^{(IBEM)} < K^{(FEM)}$
		11	44	30	121	130	13	52	42	169	245	15	60	56	225	423	
		7	28	12	98	40	11	44	30	242	219	12	48	36	288	305	
3D		Number of quadrature nodes *nG*															
		4					9					16					
				[MB]		[MB]			[MB]		[MB]			[MB]		[MB]	
		20	2400	22	8000	65	26	4056	63	17576	296	32	6144	144	32768	985	$K^{(IBEM)} < K^{(FEM)}$
		13	1014	35	2197	82	15	1350	63	3375	186	18	1944	130	5832	530	
		Number of quadrature nodes *nG*															
		4					7					13					
		24	6912	46	27648	186	29	10092	97	48778	556	36	15552	231	93312	1957	$K^{(IBEM)} < K^{(FEM)}$
		14	2352	84	5488	190	16	3072	144	8192	408	18	3888	231	11664	806	

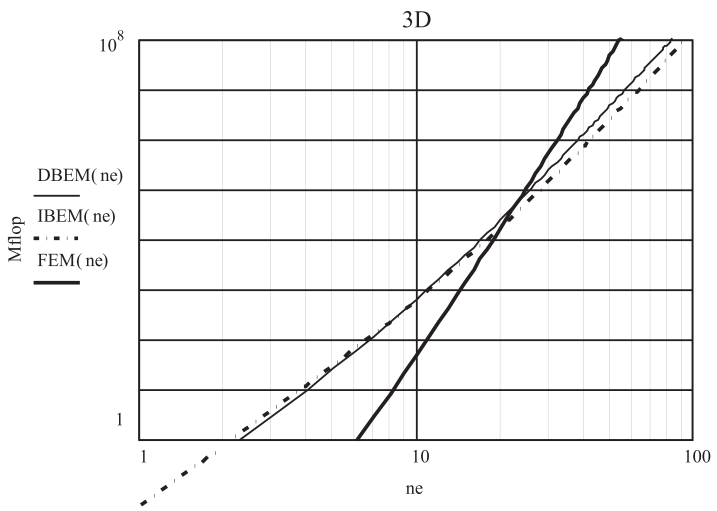


Fig. 6. Number of mathematical operations versus number of linear quadrangular/hexahedral elements for three-dimensional area and four-node quadrature.

According to Figs. 5 and 6, the IBEM computing costs for 2D area and a two-node quadrature are always lower than the computing costs of DBEM, while the FEM computing costs are lower than the costs of BEM only for small models (less than 484 linear finite elements). In the case of 3D area, the IBEM computing costs are lower than those of DBEM only for large models (more than 600 linear quadrangular boundary elements) while the FEM computing costs are lower than those of BEMs for small, medium-sized and large models (less than 8000 linear hexahedral finite elements). The computing costs of FEM increase with the number of discrete elements much faster than those of BEMs.

5. Discussion of results

5.1. Comparison of IBEM and DBEM methods

It is commonly thought that since integration in IBEM is performed twice over all the elements during the composition of the main matrix, the method is faster than DBEM only for very large models. But the presented results show that in some cases IBEM may be faster than DBEM for both large and small models, e.g. for quadrature with five nodes (Table 3).

In the case of two-dimensional models built from linear elements, IBEM is faster than DBEM, regardless of the model size, for quadrature with one or two nodes. For quadrature with three or more nodes, IBEM is faster than DBEM only for medium-sized and large (more than 328 elements) models. The point of intersection of computing costs for the methods depends largely on the number of nodes of the quadrature used, i.e. on the latter's order.

For 2D models made up of parabolic elements, IBEM is faster than DBEM, regardless of model size, for quadrature with $1 \div 5$ nodes. IBEM is faster than DBEM for the above models for quadrature with as many as 5 nodes because the costs of composing the main matrix in the IBEM method are directly proportional to the second power of the number of elements, while in the DBEM – to the second power of the number of nodes. 2D models built from parabolic elements have twice as many nodes as elements (Table 1), whereas the costs of factorising matrix **A** in IBEM are by half lower than in DBEM.

In the case of 3D models made up of linear elements, IBEM is faster than DBEM, regardless of model size, only for quadrature with one node. If quadrature with a larger number of nodes is used, IBEM is faster than DBEM only for large models (more than 600 elements).

For 3D models made up of linear triangular boundary elements, IBEM is much slower than DBEM because of the much larger number of elements at the same number of nodes.

In the case of 3D models made up of parabolic parameters, IBEM is faster than DBEM for small models only for quadratures with 1 and 4 nodes. For quadratures with a larger number of nodes, IBEM is faster than DBEM for medium-sized and large (more than 300 elements) models.

The point of intersection of computing costs for the methods for 3D area is also strongly dependent on the number of nodes of the quadrature used.

In IBEM, the computing memory needed to store the main matrix is twice smaller than in DBEM, which is a further argument for using IBEM for modelling large acoustical systems.

5.2. Comparison of FEM and BEM methods

From the comparative analysis of FEM and BEM one can draw the following conclusions.

In the case of a 2D area and quadrature with $2 \div 4$ nodes, FEM is faster than DBEM for models made up of maximum $22 \div 26$ linear elements or maximum $11 \div 16$ parabolic elements per one dimension (Table 4), which corresponds to a mesh with $88 \div 104$ linear boundary elements or $484 \div 676$ linear finite elements and to a mesh with $44 \div 64$ parabolic boundary elements or $220 \div 340$ parabolic finite elements. The point of intersection of computing costs for DBEM and FEM is weakly dependent on the order of the quadrature used.

In the case of 2D area and quadrature with $2 \div 4$ nodes, FEM is faster than IBEM for models consisting of maximum $23 \div 33$ linear elements or maximum $7 \div 15$ parabolic elements per one dimension (Table 5), which corresponds to a mesh with $92 \div 132$ linear boundary elements or $529 \div 1089$ linear finite elements and to a mesh with $28 \div 60$ parabolic boundary elements or about $100 \div 290$ parabolic finite elements. Thus FEM is faster than DBEM and IBEM for small and medium-sized models. The point of intersection of computing costs depends largely on the order of the quadrature used.

In the case of 3D models and quadrature with $4 \div 16$ nodes, FEM is faster than BEM for small, medium-sized and large models. In practice, a quadrature with the number of nodes below 4 is not used for 3D areas. In the case of a quadrature with 4 nodes, BEM is faster than FEM only for models made up of 2400 linear boundary elements or 8000 linear finite elements and at least 1014 parabolic boundary elements or 2197 parabolic finite elements (Table 5). For models with the same computing costs, the size of memory needed to store the main matrix in FEM is about twice as large as in IBEM and about 2–8 time larger than in IBEM. For example, IBEM's and FEM's computing costs for a three-dimensional model, using linear quadrangular elements and a quadrature with four nodes, are the same for meshes consisting of 2400 boundary elements (22 MB) or 8000 finite elements (65 MB). It may be difficult to store a 65 MB main matrix in the operating memory while the use of algorithms dividing the matrix into smaller matrices stored on the hard disk dramatically increases the computing time. Therefore when selecting a method for modelling a 3D area, one should consider both the computing cost and the size of computer memory needed for the computations.

Computing costs for 3D models are strongly dependent on the order of the quadrature used.

6. Conclusions

The computing costs of FEM and BEM and coupled FEM/FEM and FEM/BEM for acoustical problems have been compared. The analysis was made for FEM, the collocative-variational boundary element method (DBEM) and the variational boundary element method (IBEM). The costs of the above methods were compared for bounded areas and criteria for selecting the method with the lowest costs for a given discrete model size have been determined.

The analysis has shown some interesting facts.

In the case of acoustic field modelling by DBEM and IBEM, the computing costs are strongly dependent on the order of the quadrature used. The costs of computing quadrature in IBEM increase much faster with quadrature's number of nodes than in DBEM.

DBEM computing costs are generally considerably lower than those of IBEM only for small models. In some cases, IBEM is faster than DBEM regardless of the size of the discrete model. For example, in the case of 2D models made up of parabolic elements for quadrature with as many as 5 nodes, the computing costs of IBEM are much lower than those of DBEM regardless of the size of the discrete model. This is so because of the dominance of factorisation costs in the total computing costs for two-dimensional discrete models made up of parabolic elements.

In the case of three-dimensional models made up of triangular boundary elements, IBEM computing costs are generally much higher than those of DBEM. Thus for the analysis of such models the latter method is more advantageous.

FEM computing costs may be lower than those of BEM. For two-dimensional areas FEM is faster than BEM only for small models. In the case of three-dimensional areas, FEM is faster than BEM for small, medium-sized and large models made up of as many as tens of thousands of linear finite elements. When selecting a method for modelling three-dimensional area by FEM and BEM, one should consider both the computing costs and the size of computer memory needed for the computations.

The analysis of computing costs was carried out for areas with a simple geometry, represented by homogenous meshes. The analysis was limited to such models since no analytical equation for the number of degrees of freedom for a discrete model with any geometry or with an inhomogeneous mesh could be formulated. But practical experience indicates that the actual relationship for FEM and BEM computing costs for areas with an uncomplicated geometry, but different from that of the parallelepiped (rectangular), and with inhomogeneous discrete meshes is very similar to the relationship determined in this paper.

References

[1] ESTORFF O VON, *Boundary elements in acoustics*, WIT Press, Southampton 2000.

[2] BETTESS P., *Operation counts for boundary integral and finite element methods*, International Journal of Numerical Methods in Engineering, **17,** 306–308 (1981).

[3] HARARI I., GROSH K., HUGHES T.J.R., MALHOTRA M., PINSKY P.M., STEWARD J.R., THOMPSON L.L., *Recent developments in finite element methods for structural acoustics*, Archives of Computational Methods in Engineering, **3,** 204–226 (1996).

[4] HORN R.A., JOHNSON C.R., *Matrix analysis*, Cambridge University Press, Cambridge 1989.

[5] DOBRUCKI A.B., *Numerical modelling of the moving system of electrodynamic loudspeaker* [in Polish], Scientific Papers of the Institute of Telecommunications and Acoustics of the Wrocław University of Technology, Wrocław 1992.

[6] Sysnoise rev. 5.2, Theoretical manual, Rev.5.2, Numerical Integration Technologies, Belgium, 1995.

[7] Bolejko R., *Numerical modelling of the acoustic field in interior and exterior domain, considering impedance boundary condition* [in Polish], PhD Thesis, Wrocław University of Technology, Wrocław 2004.

[8] DING W.P., CHEN H.L., *A symmetrical finite element model for structure-acoustic coupling analysis of an elastic, thin-walled cavity*, Journal of Sound and Vibration, **243,** 547–559 (2001).

[9] KAGAWA Y., YAMABUSHI T., SUGIHARA K., *A finite element approach to a coupled structural-acoustic radiation system with application to loudspeaker characteristic calculation*, Journal of Sound and Vibration, **69,** 229–243 (1980).

[10] SANDBERG G., GÖRANSSON P., *A symmetric finite element formulation for acoustic fluid-structure interaction analysis*, Journal of Sound and Vibration, **123,** 507–515 (1988).