Solving generalized eigenvalue problems for large scale fluid-structure computational models with mid-power computers

Quentin Akkaoui, Evangéline Capiez-Lernout, Christian Soize, Roger Ohayon

Solving generalized eigenvalue problems for large scale fluid-structure computational models with mid-power computers

Q. Akkaoui[a], E. Capiez-Lernout[a], C. Soize[a,*], R. Ohayon[b]

[a]*Laboratoire Modélisation et Simulation Multi Echelle (MSME) UMR 8208 CNRS, 5 Boulevard Descartes 77454 Marne-La-Vallée, France*

[b]*Structural Mechanics and Coupled System Laboratory, Conservatoire National des Arts et Métiers (CNAM), 2 rue Conté, 75003, Paris , France*

Abstract

This article proposes a method for solving generalized eigenvalue problems on medium-power computers with a moderate memory in the particular context of studying fluid-structure systems with sloshing and capillarity. This research was performed following many RAM problems encountered when computing the modal characterization of the system studied. The methodology proposed is one solution to reduce RAM and time required for the computation, by using methods such as double projection or subspace iterations.

Keywords: Generalized eigenvalue problem, large scale computational models, algorithms, fluid-structure, structure, acoustic, sloshing, capillarity, reduced-order model.

1. Introduction

The algorithms for solving eigenvalue problems (including generalized eigenvalue problems for which one matrix is positive definite) have received a very great attention this last 40 years from a mathematical point of view (see for instance, [1, 2, 3, 4, 5, 6, 7, 8, 9]), for algorithms adapted to parallel computation (see for instance, [10, 11, 12, 13, 14, 15, 16, 17, 18]), and also for massively parallel computers (see for instance, [19, 20, 21, 22, 23]). The majority of the efficient algorithms have been implemented in a mathematical library for computers, parallel computers, and massively parallel computers (see for instance, [24, 25, 26]).

This paper is devoted to the computation of very populated sparse matrices involved in generalized eigenvalue problems that have to be solved in the framework of fluid-structure problems. Concerning the algorithms for solving these generalized eigenvalue problems for which one of the two matrices is a positive-definite matrix, the mathematical libraries cited before could, *a*

*Corresponding author

Email addresses: quentin.akkaoui@univ-paris-est.fr (Q. Akkaoui), evangeline.capiez-lernout@univ-paris-est.fr (E. Capiez-Lernout), christian.soize@univ-paris-est.fr (C. Soize), roger.ohayon@lecnam.net (R. Ohayon)

priori, be used (these algorithms are really efficient and are adapted to large scale models using parallel and massively parallel computers). Although these algorithms are efficient on *mid-power computers* that we define as workstations with, for instance, 264 GB to 1 TB for the RAM and 12 to 72 cores for the processors, we have encountered huge difficulties due to the limitation of RAM and also to CPU-time consumption.

The framework of the developments proposed is the one relative to the computation of reduced-order bases (ROB) in order to construct a reduced-order model (ROM) of a fluid-structure computational model that corresponds to an elastic structure coupled with an internal acoustic liquid with a free surface for which there are sloshing phenomena and surface tension effects. This ROM is not constructed using a global ROB associated with the full coupled problem, but is constructed using the elastic modes of the structure with the added-mass effects, the acoustic modes of the liquid, and the sloshing/capillarity modes. The interest of such a formulation (see [27, 28, 29]) is to be able to select the modes that contribute to the responses in the frequency band of analysis and also to be able to implement the nonparametric probabilistic approach of model uncertainties in each part of the coupled system for which the level of uncertainties differs from a part to another one. It should be noted that this formulation differs from the vibroacoustics problems (without sloshing and surface tension effects) for which a ROM is constructed using a global ROB (see for instance [30]). The difficulties encountered in the computation depends on the type of modes that have to be computed. Concerning the computation of the elastic structural modes, the mass matrix of the generalized eigenvalue problem is made up of the sparse mass matrix of the structure in which is added the added-mass matrix of the internal liquid (the added-mass matrix is a full matrix with respect to the fluid-structure coupling dofs). Due to a RAM consumption problem, the computation of the added-mass matrix cannot be done as soon as the acoustic-stiffness matrix of the internal liquid is very populated. In addition, assuming that the added-mass matrix has been computed, if the stiffness matrix of the structure is also very populated, another difficulty arises for solving the generalized eigenvalue problem inducing the same type of RAM consumption. The difficulties are exactly of the same nature for the computation of the sloshing/capillarity modes. Concerning the computation of the acoustic modes of the internal liquid, the difficulties are due to the generalized eigenvalue problem that involves two very populated sparse matrices, the acoustic mass and the acoustic stiffness matrices. These difficulties are detailed in Section 5 for which the fluid-structure computational model has 2×10^6 dofs and requires, among others, to solve a linear equation for a positive-definite matrix that has 1.2×10^8 non-zeros entries requiring about 10^9 bytes.

Confronted with this situation, we have thus revisited the formulations in order to be able to solve the three generalized eigenvalue problems on a mid-power computer. The authors think that the substantial efforts, which have been performed, could be of interest for the community. It should be noted that the formulations/algorithms proposed allow for computing a large scale fluid-structure computational model on mid-power computers but certainly, would allow for computing very large scale fluid-structure computational models on high-power computers.

The computational model of the considered fluid-structure system is constructed using the

finite element method, assuming the structure is linear elastic and the internal acoustic liquid is dissipative. The free surface of the liquid is submitted to an acceleration field independent of time such as the gravitation field, inducing sloshing phenomena. The surface tension effects are taken into account.

In the particular context of this fluid-structure interaction problem for which sloshing and surface tension effects are taken into account, many research have been performed (see for instance, [27, 31, 32, 33]). In this paper, the formulation used is the one presented in [28, 29] for which the adapted reduced-order model (ROM) has been evoked and is more detailed hereinafter. The construction of the ROM requires a modal characterization of the different parts of the fluid-structure system. It consists in projecting the computational model using three ROB's that are computed by solving three generalized eigenvalue problems. The modal characterization of the structure is obtained by computing the elastic eigenmodes of the structure taking into account the influence of the internal acoustic liquid in order to assure a fast convergence with respect to the number of elastic modes retained in the ROM. The modal characterization of the internal acoustic liquid is obtained by computing the acoustic modes with a free surface on which the pressure is zero. Finally, the modal characterization of the free surface in presence of surface tensions is obtained by computing the sloshing modes that involve the internal acoustic liquid. The finite element meshes of the fluid-structure system that will be considered in Section 5 have a large number of dofs and a high connectivity, inducing very populated sparse matrices and consequently, leading us to an impossibility to construct the matrices and to compute the generalized eigenvalue problems on mid-power computers using the most adapted algorithms available in the mathematical libraries such as those proposed in Matlab.

Concerning the choice of the formulation, two possibilities can be envisaged. For computing the structural elastic modes with the added-mass effects or for computing the sloshing modes with capillarity effects, a first formulation could be based on the use of iterative algorithm for solving linear matrix equation (relative to all the physical dofs) for a very populated matrix and for a large number of right-hand side members. A second formulation would avoid to solve such a linear systems of equations in high dimension by using a double projection method, also known as the Rayleigh-Ritz method in the framework of eigenvalue problems. An analysis of the advantages/disadvantages of these two formulations has been performed in order to choose the most efficient one. This analysis is summarized in Section 3.3 and allows for concluding that the double projection method is more efficient and consequently, will be retained in this paper.

For solving the elastic and the sloshing/capillarity generalized eigenvalue problems, a double projection method is implemented. This approach allows for decreasing the CPU time and for reducing the RAM avoiding the out of memory and consequently, allowing the computation to be effectively performed. For solving the acoustic generalized eigenvalue problem, the main difficulty is induced by the RAM problem for which an out of memory is obtained. For circumventing this difficulty, we have implemented the subspace iteration method, first introduced in [4, 8], and which is particularly efficient for the problem that has to be solved. Such an approach increases the CPU time but it is the only solution that we have found for avoiding the out of memory. In this paper, we briefly summarize the subspace iteration algorithm because we need to adapt it to

the formulation used.

The paper is organized as follows. Section **??** introduces the fluid-structure computational model. In Section 2, we present a summary of the classical formulation of the elastic, the acoustic, and the sloshing/capillarity generalized eigenvalue problems. These eigenvalue problems allow for computing the projection bases required for constructing the ROM. Section 3 deals with the difficulties encountered with mid-power computers for a large scale computational model when classical algorithms are used. Section 4 is devoted to a new strategy for solving the three generalized eigenvalue problems without inducing a RAM overconsumption. Finally, Section 5 is devoted to an application that allows for quantifying the computer resources required for the computation of the projection bases. This analysis is carried out with respect to the number of dofs of finite element model. The results obtained validate the efficiency of the proposed algorithms in terms of RAM consumption and computational CPU-time cost.

2. Summary of the classical formulation of the generalized eigenvalue problems for the considered fluid-structure computational model

We consider the fluid-structure system in its reference configuration taken as the natural state without prestresses. The boundary conditions are such that there are no rigid body displacements for the structure. The structure is elastic and dissipative, and contains a dissipative acoustic fluid simply called "acoustic fluid". Furthermore, the gravitational and surface tension effects are taken into account, yielding a free surface vibrational motion induced by sloshing and capillarity effects. We are interested in analyzing the linear vibrations of the coupled system around its reference configuration. The unknowns of the problem are the vector **u** of the structural displacements, the vector **p** of the pressure in the acoustic fluid, and the vector **h** of the elevation of the free surface. In this framework, we present an adapted method for computing the projection bases allowing the ROM to be constructed for large scale computational models, which stay adapted to the mid-power computers.

As explained in Section 1, the formulation presented in [28, 29] is used. In order to assure the readability of this paper, this approach, that will be defined as the *classical formulation* is briefly summarized in this section.

In the following, index S, F, and H are used for referencing quantities related to the structural displacements, to the acoustic-fluid pressure, and to the free-surface elevation, which are respectively represented by vector **u**, **p**, and **h**.

Let n_S, n_F, n_H be the number of degrees of freedom of the structure, the fluid, and the free surface. Let M_S and K_S be the $(n_S \times n_S)$ mass and stiffness matrices related to the equations in **u** for the structure, M_F and K_F be the $(n_F \times n_F)$ "mass" and "stiffness" matrices related to the equations in **p** for the acoustic fluid, and K_{gc} be the $(n_H \times n_H)$ matrix related to the equations in **h** for the liquid free surface with capillarity. Let C_{pu} and $C_{p\eta}$ be the $(n_S \times n_F)$ and $(n_H \times n_F)$

rectangular matrices, representing the coupling between **p** and **u** for the acoustic fluid and the structure, and the coupling between **p** and **h** for the acoustic fluid and the free surface. Note that the null space of K_F is equal to 1 and consequently, matrix K_F is not positive definite but is only positive semidefinite. In the following, we then introduce the subspace $\mathcal{R}_F$ of $\mathbb{R}^{n_F}$ of the pressure vector **p** in $\mathbb{R}^{n_F}$ such that $\mathbf{p} = 0$ for the dofs related to the free surface.

2.1. Generalized eigenvalue problem for the structure (elastic eigenvalue problem)

In the framework of the considered fluid-structure computational model, the $N_S \ll n_S$ elastic modes of the structure with added-mass effect of the fluid, which have to be calculated, require to solve the following generalized eigenvalue problem,

$$K_S\,\Phi_S^{\text{ref}} = (M_S + M_A)\,\Phi_S^{\text{ref}}\,\Lambda_S^{\text{ref}}\,, \tag{1}$$

in which the fluid added-mass matrix M_A that characterizes the quasi-static effect of the acoustic fluid on the structure [27, 29], is an $(n_S \times n_S)$ positive-definite symmetric matrix that is formally written as,

$$M_A = C_{pu}\,(K_F)^{-1}\,(C_{pu})^T\,. \tag{2}$$

Matrix K_F is not invertible, $(K_F)^{-1}$ denotes its inverse in $\mathcal{R}_F$, and in addition, matrix K_F is not explicitly inverted in $\mathcal{R}_F$. Matrix M_A is computed by $M_A = C_{pu}\,D$ in which the $(n_F \times n_S)$ matrix D is computed by solving, in the subspace $\mathcal{R}_F$, the linear matrix equation,

$$K_F\,D = (C_{pu})^T\,. \tag{3}$$

Note that, Eq. (3) is solved using the minimum degree algorithm for minimizing the non-zeros elements in the sparse factor of the Cholesky factorization in $\mathcal{R}_F$ of sparse matrix K_F. In Eq. (1), the $(N_S \times N_S)$ diagonal matrix Λ_S^{ref} contains the first N_S smallest positive eigenvalues sorted by increasing order such that $\lambda_1^{S,\text{ref}} \leq \ldots \leq \lambda_{N_S}^{S,\text{ref}}$. The full $(n_S \times N_S)$ matrix Φ_S^{ref} is the matrix of the corresponding eigenvectors that satisfy the following orthogonality properties,

$$(\Phi_S^{\text{ref}})^T\,(M_S + M_A)\,\Phi_S^{\text{ref}} = \mathbb{I}_{N_S}\,, \tag{4}$$

$$(\Phi_S^{\text{ref}})^T\,K_S\,\Phi_S^{\text{ref}} = \Lambda_S^{\text{ref}}\,, \tag{5}$$

in which $\mathbb{I}_{N_S}$ is the $(N_S \times N_S)$ identity matrix.

Remark. The matrix $M_S + M_A$ can be viewed as the Schur complement [6] in $\mathcal{R}_F$ of the matrix block K_F, which is defined as

$$\begin{bmatrix} M_S & C_{pu} \\ -(C_{pu})^T & K_F \end{bmatrix}$$

The method proposed in Section 4.1 will allow for avoiding the computational difficulties related to the RAM consumption induced by the use of the classical algorithms for computing a Schur complement. It should also be noted that the method proposed in Section 4.1 would be very efficient for a hydroelastic problem related to an incompressible liquid in presence of a free surface on which there is a zero pressure condition.

2.2. Generalized eigenvalue problem for the acoustic fluid (acoustic eigenvalue problem)

The $N_F \ll n_F$ acoustic modes of the acoustic fluid, which have to be computed, are obtained by solving the following generalized eigenvalue problem on $\mathcal{R}_F$ (that is to say with $\mathbf{p} = 0$ for the degrees of freedom related to the free surface),

$$K_F \, \Phi_F^{\text{ref}} = M_F \, \Phi_F^{\text{ref}} \, \Lambda_F^{\text{ref}} \, , \tag{6}$$

in which the $(N_F \times N_F)$ diagonal matrix Λ_F^{ref} contains the first N_F smallest positive eigenvalues sorted by increasing order such that $\lambda_1^{F,\text{ref}} \leq \ldots \leq \lambda_{N_F}^{F,\text{ref}}$. The full $(n_F \times N_F)$ matrix Φ_F^{ref} is the matrix of the acoustic modes, for which the columns are the corresponding eigenvectors that satisfy the following orthogonality properties,

$$(\Phi_F^{\text{ref}})^T \, M_F \, \Phi_F^{\text{ref}} = \mathbb{I}_{N_F} \, , \tag{7}$$

$$(\Phi_F^{\text{ref}})^T \, K_F \, \Phi_F^{\text{ref}} = \Lambda_F^{\text{ref}} \, . \tag{8}$$

2.3. Generalized eigenvalue problem for the sloshing with capillarity (sloshing/capillarity eigenvalue problem)

The computation of the $N_H \ll n_H$ sloshing/capillarity modes consists in finding the eigenvalues represented by the $(N_H \times N_H)$ diagonal matrix Λ_H^{ref} and the associated eigenvectors represented by the $((n_H + n_F) \times N_H)$ matrix Ψ_H^{ref} that is written by blocks as

$$\Psi_H^{\text{ref}} = \begin{bmatrix} \Phi_H^{\text{ref}} \\ \Phi_{FH}^{\text{ref}} \end{bmatrix} \, , \tag{9}$$

in which Φ_H^{ref} is a $(n_H \times N_H)$ matrix and Φ_{FH}^{ref} is a $(n_F \times N_H)$ matrix, such that

$$K_F \, \Phi_{FH}^{\text{ref}} + (C_{p\eta})^T \, \Phi_H^{\text{ref}} \, \Lambda_H^{\text{ref}} = 0 \, , \tag{10}$$

$$C_{p\eta} \, \Phi_{FH}^{\text{ref}} + K_{gc} \, \Phi_H^{\text{ref}} = 0 \, , \tag{11}$$

that has to be solved with a constant pressure condition on the free surface of the acoustic fluid. Matrix Λ_H^{ref} contains the first N_H smallest positive eigenvalues sorted by increasing order such that $\lambda_1^{H,\text{ref}} \leq \ldots \leq \lambda_{N_H}^{H,\text{ref}}$. Eliminating Φ_{FH}^{ref} in Eqs. (10) and (11) by ensuring the constant pressure condition on the free surface, is equivalent to solve the following generalized eigenvalue problem,

$$K_{gc} \, \Phi_H^{\text{ref}} = M_{gc} \, \Phi_H^{\text{ref}} \, \Lambda_H^{\text{ref}} \, , \tag{12}$$

in which M_{gc} is a positive-definite $(n_H \times n_H)$ matrix that is formally written as $M_{gc} = C_{p\eta} \, (K_F)^{-1} \, (C_{p\eta})^T$ because K_F is not invertible, and which is rewritten as $M_{gc} = C_{p\eta} \, S$ where the $(n_F \times n_H)$ matrix S is computed by solving the linear matrix equation

$$\begin{bmatrix} K_F & (\zeta)^T \\ \zeta & 0 \end{bmatrix} \begin{bmatrix} S \\ L \end{bmatrix} = \begin{bmatrix} (C_{p\eta})^T \\ 0 \end{bmatrix} \, , \tag{13}$$

in which L is the $(1 \times n_H)$ matrix of the Lagrange multipliers. The $(1 \times n_F)$ matrix ζ is written as $\zeta = [1 \dots 1]\, C_{p\eta}$ where $[1 \dots 1]$ is a $(1 \times n_H)$ matrix with all the entries are 1. The orthogonality properties related to the generalized eigenvalue problem defined by Eq. (12) are written as

$$(\Phi_H^{\text{ref}})^T\, M_{gc}\, \Phi_H^{\text{ref}} = \mathbb{I}_{N_H}\,, \tag{14}$$

$$(\Phi_H^{\text{ref}})^T\, K_{gc}\, \Phi_H^{\text{ref}} = \Lambda_H^{\text{ref}}\,. \tag{15}$$

Once Eq. (12) is solved and therefore, matrix Φ_H^{ref} is known, matrix Φ_{FH}^{ref} is computed by

$$\Phi_{FH}^{\text{ref}} = -S\, \Phi_H^{\text{ref}}\, \Lambda_H^{\text{ref}} \quad . \tag{16}$$

3. Computational limitations induced by the classical formulation for a large scale computational model if classical algorithms are used

This section is related to the classical formulation of the generalized eigenvalue problem, which has been defined in Section 2.

3.1. Brief description of the algorithms required for solving the generalized eigenvalue problems of the introduced formulation

The formulation that has been presented in Section 2 requires to use an algorithm for solving a given linear matrix equation for a positive-definite matrix and an algorithm for solving a generalized eigenvalue problem for two real symmetric matrices for which one is positive definite.

- The first classical algorithm (ALG1) is used for solving a linear matrix equation of the type $A\,X = B$ in which A is a symmetric positive-definite matrix (see, for instance, Eq. (2)). Such a classical algorithm consists in computing the Cholesky factorization $C\,(C)^T$ of A using a minimum degree algorithm for optimizing the sparsity of C. Then the solution of the linear matrix equation is obtained using the standard successive back-substitutions that is formally written as $X = (C^{-1})^T\,(C^{-1}\,B)$.

- The second classical algorithm (ALG2) is used for computing the N first smallest eigenvalues and their associated eigenvectors of the generalized eigenvalue problem of the type $A\,X = \lambda\,B\,X$ in which A and B are two symmetric positive-definite matrices (see, for instance, Eq. (6)). The classical algorithm for solving such a generalized eigenvalue problem consists in transforming it into a classical eigenvalue problem $P\,Y = \lambda\,Y$. For that, algorithm ALG1 is used for computing P. This type of algorithm is used by Matlab that calls the standard library LAPACK [25, 6].

It should be noted that the computational difficulties related to the RAM consumption are generally not due to the eigenvalue problem $P\,Y = \lambda Y$ but are due to the use of ALG1 by ALG2.

3.2. Limitations related to RAM consumption for large scale computational models on mid-power computers

As explained in Section 1, the difficulties occur with ALG1 when sparse matrices M_F and K_F are very populated. This is the case, for instance, for a medium-scaled fluid-structure computational model for which 20-node 3D finite elements are used for the structure in order to improve the accuracy of the finite element approximation. These difficulties are induced by the use of ALG1 in the following numerical steps.

- The elastic eigenvalue problem defined in Section 2.1 requires to compute matrix M_A (see Eq. (2)) solving the matrix equation defined by Eq. (3) using ALG1 involving matrix K_F. The memory overconsumption is mainly due to the number of columns in matrix $(C_{pu})^T$.

- The acoustic eigenvalue problem defined in Section 2.2 requires to solve the generalized eigenvalue problem defined by Eq. (6) using ALG2 that uses ALG1 involving matrix M_F. The memory overconsumption is mainly due to the computation of the Cholesky factorization of matrix M_F.

- Finally, the sloshing/capillarity eigenvalue problem is obtained by solving the linear matrix equation defined by Eq. (13) using ALG1 and then solving the generalized eigenvalue problem defined by Eq. (12) with ALG2 involving matrix K_F. The memory overconsumption is mainly due to the number of columns in matrix $(C_{p\eta})^T$.

3.3. Remark concerning the choice of a formulation

In this section, we analyze the CPU time induced by an iterative algorithm with respect to ALG1. For solving Eq. (3) in which K_F is a very populated sparse matrix and where the number of active columns in the right-hand side member is large, an iterative solver could be used (the problem is similar for Eq. (13)). As K_F is positive, the preconditioned conjugate gradient iterative algorithm, denoted as "PCG", is used for computing matrix D. This iterative algorithm is known for being very efficient when solving $A\,\mathbf{x} = \mathbf{b}$ in which A is a symmetric positive-definite matrix. Such an algorithm requires that the right-hand side member $\mathbf{b}$ be a vector (and not a matrix), which is not the case for the problem that we have to solve. This is why, the use of such an iterative algorithm is not, *a priori*, the best choice for the computation of matrix D, because the PCG algorithm should be used for each column of matrix $(C_{pu})^T$. The PCG algorithm requires the use of a preconditioner to speed up the convergence, which is chosen as the incomplete Cholesky factorization of matrix K_F. This incomplete Cholesky factorization requires a filling parameter named as "drop tolerance" that has to be optimized for using it in the PCG algorithm. This drop tolerance optimization must take into account the time required to compute the incomplete Cholesky factorization of K_F and also the time required to solve the linear system with this preconditioner. The analysis of the CPU-time consumption for such iterative solver has been performed on the smallest mesh of the application presented in Section 5. The obtained results show that the optimum value of the drop tolerance is 1.8×10^{-4} for the incomplete Cholesky preconditioner. The comparison of the CPU time required to solve the linear matrix equation Eq. (3) is defined as follows. Let t_{ALG1} be the CPU time required to solve Eq. (3) with ALG1 defined in Section 3.1 and let t_{PCG}

be the CPU time required with the PCG algorithm. The CPU times are $t_{\mathrm{ALG1}} = 8.92$ hours and $t_{\mathrm{PCG}} = 57.16$ hours, and the corresponding elapsed times are $t^e_{\mathrm{ALG1}} = 0.96$ hours and $t^e_{\mathrm{PCG}} = 30.1$ hours. This difference between ALG1 and PCG can be explained by the loop required to solve the linear system for each column of matrix C^T_{pu}, which is heavily time consuming. For solving the linear matrix equation, Eq. (3), with the PCG algorithm, the loop could be parallelized to speed up the computation but would highly increase the RAM consumption. Finally, it is concluded that PCG will not be retained.

4. Adapted numerical strategy for solving the generalized eigenvalue problems related to the considered fluid-structure computational model

Taking into account the limitations highlighted in Section 3.2, an alternative numerical strategy is proposed, allowing the elastic, acoustic, and sloshing/capillarity eigenvalue problems, to be solved for large scale fluid-structure computational models on mid-power computers with a moderate RAM.

4.1. Double projection algorithm for solving the elastic eigenvalue problem

The proposed double projection method allows for circumventing the difficulties induced by the computation of matrix M_A. It consists in introducing a second projection for solving the generalized eigenvalue problem defined by Eq. (1). The modal matrix Φ_S is then rewritten as,

$$\Phi_S = \Phi^i_S \, \widetilde{\Phi}_S \, , \tag{17}$$

in which Φ^i_S is the full $(n_S \times N_q)$ matrix that corresponds to the first projection basis on a subspace of dimension $N_q > N_S$, and where $\widetilde{\Phi}_S$ is the full $(N_q \times N_S)$ matrix that corresponds to the second projection basis on a subspace of dimension N_S. In the present case, the first projection basis is constructed by solving the following eigenvalue problem for the structure in vacuo (without the acoustic fluid),

$$K_S \, \Phi^i_S = M_S \, \Phi^i_S \, \Lambda^i_S \, . \tag{18}$$

If N_q is sufficiently large, the N_S eigenvectors computed using Eq. (1) belong to the subspace spanned by Φ^i_S. Note that for $N_q = n_S$, Φ^i_S is a vector basis of the admissible set and consequently, N_q can always be found for obtaining the convergence. Therefore the double projection method requires a convergence analysis with respect to N_q. Matrix Φ^i_S satisfies the following orthogonality properties,

$$(\Phi^i_S)^T \, M_S \, \Phi^i_S \;=\; \mathbb{I}_{N_q} \, , \tag{19}$$

$$(\Phi^i_S)^T \, K_S \, \Phi^i_S \;=\; \Lambda^i_S \, . \tag{20}$$

Left multiplying Eq. (1) by $(\Phi^i_S)^T$ and using Eq. (17) yield the following generalized eigenvalue problem with a very small dimension N_q (computationally solved without any problem),

$$\Lambda^i_S \, \widetilde{\Phi}_S = \left(\mathbb{I}_{N_q} + \mathcal{M}_A \right) \widetilde{\Phi}_S \, \Lambda_S \, , \tag{21}$$

for which the following orthogonality conditions are satisfied,

$$(\widetilde{\Phi}_S)^T(\mathbb{I}_{N_q} + \mathcal{M}_A)\,\widetilde{\Phi}_S = \mathbb{I}_{N_S}\,, \tag{22}$$

$$(\widetilde{\Phi}_S)^T\,\Lambda_S^i\,\widetilde{\Phi}_S = \Lambda_S\,. \tag{23}$$

In Eq. (21), the $(N_q \times N_q)$ positive-definite matrix $\mathcal{M}_A$ is written as $\mathcal{M}_A = (\Phi_S^i)^T\,M_A\Phi_S^i$. Using Eq. (2), this matrix can be rewritten as

$$\mathcal{M}_A = \mathcal{C}_{pu}\,(K_F)^{-1}\,(\mathcal{C}_{pu})^T\,, \tag{24}$$

in which $\mathcal{C}_{pu} = (\Phi_S^i)^T\,C_{pu}$ is a $(N_q \times n_F)$ sparse rectangular matrix and where $(K_F)^{-1}$ is a formal writing that is specified hereinafter. In practice, matrix $\mathcal{M}_A$ is computed by $\mathcal{M}_A = \mathcal{C}_{pu}\,X$ in which the $(n_F \times N_q)$ matrix X is the solution of the linear matrix equation $K_F\,X = (\mathcal{C}_{pu})^T$ that is solved with ALG1 in subspace $\mathcal{R}_F$. Such calculations are done with a reasonable computational time and RAM consumption.

4.2. Subspace iterations for the acoustic eigenvalue problem

For solving the acoustic eigenvalue problem, a method for circumventing the difficulties appearing in ALG2 when using ALG1 that involves matrix M_F consists in using the subspace iteration method [4, 8], which is briefly summarized and adapted to our context.

First, an initial projection basis, represented by the $(n_F \times N_p)$ matrix X_0 with $N_F < N_p \ll n_F$, is computed using the initialization procedure described in [8]. Then the projection basis is updated using the following iterative algorithm in which the subscript k, which belongs to $\{0, 1, 2, \ldots\}$, denotes the current iteration,

$$K_F\,\widetilde{X}_{k+1} = M_F\,X_k\,, \tag{25}$$

$$K_F^{k+1} = (\widetilde{X}_{k+1})^T\,K_F\,\widetilde{X}_{k+1}\,, \tag{26}$$

$$M_F^{k+1} = (\widetilde{X}_{k+1})^T\,M_F\,\widetilde{X}_{k+1}\,, \tag{27}$$

$$K_F^{k+1}\,Q_{k+1} = M_F^{k+1}\,Q_{k+1}\,\Lambda_F^{k+1}\,, \tag{28}$$

$$X_{k+1} = \widetilde{X}_{k+1}\,Q_{k+1}\,. \tag{29}$$

This iterative procedure is stopped when the following convergence criterion is reached,

$$\left\{1 - \frac{(\lambda_i^{(k+1)})^2}{(\mathbf{q}_i^{(k+1)})^T\,\mathbf{q}_i^{(k+1)}}\right\}^{\frac{1}{2}} \leq \mathrm{tol} \quad , \quad \mathrm{i} = 1, \ldots, \mathrm{N_p}\,,$$

where $\mathbf{q}_i^{(k+1)}$ is the i-th vector in matrix Q_{k+1} corresponding to $\lambda_i^{(k+1)}$, and $\mathrm{tol} = 10^{-2\mathrm{s}}$ with accuracy of $2s$ digits in the required eigenvalues. Finally, a Sturm sequence checking is performed to ensure that the correct eigenvalues and their associated eigenvectors have been calculated. This classical iterative procedure could certainly be speed up using the method recently proposed in [34]. It would be an additional improvement of the method proposed for limiting the RAM consumption.

4.3. Double projection algorithm for the sloshing/capillarity eigenvalue problem

Again, for avoiding the difficulties induced solving Eq. (13), a double projection method similar to the one described in Section 4.1 is used for computing the sloshing/capillarity eigenvalue problem defined in Section 2.3.

4.3.1. Comments about the construction of an approximation of matrix K_F

In order to circumvent the difficulties due to the RAM overconsumption, several possibilities have been explored for constructing a very sparsely populated matrix K_F^a that approximates the very populated matrix K_F.

(i) - A first approximation would consist in using for K_F^a the restriction of K_F to the pressure dofs related to the free surface of the acoustic liquid (this means that the corresponding rows and columns of K_F and the corresponding columns of $C_{p\eta}$ are removed). This approximation leads a very slow convergence of the double projection method.

(ii) - A second one would consist in taking for K_F^a a few diagonals of K_F in order to take into account that K_F corresponds to the finite element approximation of the Laplace operator. This approach is difficult because K_F^a must stay nonnegative and in practice, such a property cannot easily be assured except if an incomplete factorization of K_F is performed (see below).

(iii) - A more natural construction of K_F^a would consist in computing an incomplete factorization of K_F. Since this matrix is positive semi-definite (and not positive definite), the incomplete Cholesky factorization of K_F cannot be used with the standard libraries such as LAPACK. Therefore, an incomplete LU factorization must be used that avoids the RAM overconsumption but which induces a very CPU-time consuming.

(iv) - The Airy infinitesimal wave theory shows that the pressure field exponentially decreases as a function of the distance (depth) to the free surface. This means that an approximation K_F^a can be constructed by keeping the pressure dofs related to a small layer (with height h_K) of the acoustic liquid under the free surface. Such an approximation is general, is very efficient, and is the one that we propose to use.

4.3.2. Double projection algorithm

Let $n_{h_K} \ll n_F$ be the number of dofs related to the small layer of the acoustic fluid of height h_K under the free surface. Note that we have $n_H < n_{h_K}$. We then have K_F^a and $C_{p\eta}^a$ as the $(n_{h_K} \times n_{h_K})$ and the $(n_H \times n_{h_K})$ matrices corresponding to the restriction of matrices K_F and $C_{p\eta}$ to the pressure dofs related to this small layer of the acoustic fluid. Consequently, matrix K_F^a is positive definite. Let Φ_H^i be the $(n_H \times N_d)$ matrix in which $N_H < N_d \ll n_H$ and let Λ_H^i be the $(N_d \times N_d)$ diagonal positive-definite matrix satisfying the following equation that corresponds to the restriction of Eqs. (10) and (11) to the n_{h_K} pressure dofs,

$$K_F^a \, \Phi_{FH}^a + (C_{p\eta}^a)^T \, \Phi_H^i \, \Lambda_H^i = 0 \,, \tag{30}$$

$$C_{p\eta}^a \, \Phi_{FH}^a + K_{gc} \, \Phi_H^i = 0 \,, \tag{31}$$

in which Φ^a_{FH} is a $(n_{h_K} \times N_d)$ matrix that is the restriction of matrix Φ_{FH} to the n_{h_K} pressure dofs related to the small layer of acoustic fluid. The elimination of Φ^a_{FH} between Eqs. (30) and (31) yields the generalized eigenvalue problem,

$$K_{gc}\,\Phi^i_H = M^a_{gc}\,\Phi^i_H\,\Lambda^i_H\,, \tag{32}$$

in which the positive-definite $(n_H \times n_H)$ matrix M^a_{gc} is written as,

$$M^a_{gc} = C^a_{p\eta}\,(K^a_F)^{-1}\,(C^a_{p\eta})^T\,. \tag{33}$$

In practice, matrix M^a_{gc} is computed by $M^a_{gc} = C^a_{p\eta}\,X$ in which the $(n_{h_K} \times n_H)$ matrix X is solution of the linear matrix equation $K^a_F\,X = (C^a_{p\eta})^T$ that is solved with ALG1. If N_d is sufficiently large, the N_H sloshing/capillarity modes defined in Section 2.3 belong to the subspace spanned by Φ^i_H. Note that, for all n_{h_K} such that $n_H \leq n_{h_K} \leq n_F$, if $N_d = n_H$ then Φ^i_H is a vector basis of the admissible set of **h** and consequently, N_d can always be found for obtaining the convergence. Therefore the double projection method requires a convergence analysis with respect to N_d. Matrix Φ^i_H verifies the following orthogonality properties,

$$(\Phi^i_H)^T\,M^a_{gc}\,\Phi^i_H = \mathbb{I}_{N_d}\,, \tag{34}$$

$$(\Phi^i_H)^T\,K_{gc}\,\Phi^i_H = \Lambda^i_H\,. \tag{35}$$

The solution of Eq. (32) provides a reasonable approximation of the sloshing/capillarity modes compared to those computed in Section 2.3. The double projection method then consists in writing the block matrix Φ_H appearing in Eq. (9) as,

$$\Phi_H = \Phi^i_H\,\widetilde{\Phi}_H\,, \tag{36}$$

in which the full $(N_d \times N_H)$ matrix $\widetilde{\Phi}_H$ corresponds to the second projection basis on a subspace of dimension $N_H < N_d$. Substituting Eq. (36) in Eq. (12) and left multiplying by $(\Phi^i_H)^T$ yield the following eigenvalue problem in the subspace with a very small dimension N_d (computationally solved without any problem),

$$\Lambda^i_H\,\widetilde{\Phi}_H = \mathcal{M}_{gc}\,\widetilde{\Phi}_H\,\Lambda_H\,, \tag{37}$$

in which $\mathcal{M}_{gc}$ is a positive-definite $(N_d \times N_d)$ matrix that is written as $\mathcal{M}_{gc} = \mathcal{C}_{p\eta}\,\mathcal{S}$, where the $(N_d \times n_F)$ matrix $\mathcal{C}_{p\eta}$ is defined by $\mathcal{C}_{p\eta} = (\Phi^i_H)^T\,C_{p\eta}$, and where matrix $\mathcal{S} = S\,\Phi^i_H$ is computed as in Section 2.3 by solving the linear matrix equation,

$$\begin{bmatrix} K_F & (\zeta)^T \\ \zeta & 0 \end{bmatrix} \begin{bmatrix} \mathcal{S} \\ \mathcal{L} \end{bmatrix} = \begin{bmatrix} (\mathcal{C}_{p\eta})^T \\ 0 \end{bmatrix}\,, \tag{38}$$

in which $\mathcal{L}$ is the $(1 \times N_d)$ matrix of Lagrange multipliers and where ζ is the matrix introduced in Section 2.3. Matrix $\widetilde{\Phi}_H$ satisfies the following orthogonal properties,

$$(\widetilde{\Phi}_H)^T\,\mathcal{M}_{gc}\,\widetilde{\Phi}_H = \mathbb{I}_{N_H}\,, \tag{39}$$

$$(\widetilde{\Phi}_H)^T\,\Lambda^i_H\,\widetilde{\Phi}_H = \Lambda_H\,. \tag{40}$$

In practice, matrix $\mathcal{S}$ is computed by solving the linear matrix equation Eq. (38) using ALG1; then the solutions of the sloshing/capillarity eigenvalue problem defined in Eq. (37) are computed using ALG2 involving ALG1. Such computation is done with a reasonable time and RAM consumption. We then obtain the block matrix Φ_{FH} of the sloshing/capillarity modes by,

$$\Phi_{FH} = -\mathcal{S}\,\widetilde{\Phi}_H\,\Lambda_H\,. \tag{41}$$

5. Quantification and limitations of the computer resources used for the computation of the projection bases

In the present numerical study, all the computations are made on a workstation with 264 GB RAM and 12 Intel(R) Xeon(R) CPU E5-2620 0 with a frequency of $2\,GHz$.

5.1. Finite element model of the fluid-structure system

The fluid-structure system is the one described in [35, 36] for which the retained dimensions are those of [35] (note that these two references are only used for defining the fluid-structure system for which capillarity effects are not analyzed). The structure is a steel tank constituted of a thin circular cylinder closed at both ends by circular plates (see Figure 1). The external radius is $R_e = 3.78 \times 10^{-2}\,m$, the thickness is $e = 2 \times 10^{-4}\,m$, and the height is $h = 0.23\,m$. This tank is partially filled with an acoustic liquid with height $h_f = 0.12\,m$. The origin O of the Cartesian coordinates system $Oxyz$ is located at the center of the bottom of the cylindrical tank. Axis Oz coincides with the axis of revolution of the system. The boundary conditions are those defined in [35, 36]. The finite element model of the fluid-structure system is constructed using 20-node 3D finite elements for the structure and for the acoustic fluid. The free surface of the liquid is meshed using 8-node 2D finite elements and the triple line is meshed using 3-node 1D finite elements. All the meshes of the computational model are compatible. On this basis, several finite element meshes with different sizes have been constructed in order to quantify and to identify the limitations of the computer resources used for the computation. Table 1 summarizes the finite element models built in order to study the efficiency of the algorithms. Figure 1 shows an example of the finite element mesh of the considered fluid-structure system for which the number of dofs is 578,000. Let n_{dof}^{system}, n_{dof}^{F}, and n_{nz}^{K} be the total number of dofs of the computational model, the number of dofs of the acoustic fluid, and the number of non-zero entries in matrix K_F. For the three generalized eigenvalue problems, we are interested in computing the first $N_S = 100$ eigenvectors of the elastic eigenvalue problem, the first $N_F = 100$ eigenvectors of the acoustic eigenvalue problem, and the first $N_H = 70$ eigenvectors of the sloshing/capillarity eigenvalue problem.

Figure 1: Example of mesh of the fluid-structure system with 578,000 dofs (the dimensions of the domains have voluntarily been modified for suiting them to the page size)

$n_{dof}^{system}\ /10^3$	$n_{dof}^{F}\ /10^3$	$n_{nz}^{K}\ /10^5$
74	44	24
127	83	46
196	134	75
362	262	147
578	441	249
850	679	387
1,142	945	541
1,664	1,411	811
2,298	1,990	1,147

Table 1: Numerical data of the mesh and sparsity of matrix K_F.

5.2. Elastic eigenvalue problem

The double projection method requires to find the optimum size of the initial projection basis Φ_S^i in order to accurately calculate the eigenvalues and to ensure a low computation-time and memory usage. Let $0 < \lambda_1^S(N_q) \leq \ldots \leq \lambda_{N_S}^S(N_q)$ be the eigenvalues computed as explained in Section 4.1. A convergence analysis of the largest eigenvalue, $\lambda_{N_S}^S$, with respect to N_q is performed for mesh with $n_{dof}^{system} = 196{,}000$. Figure 2 displays the graph of the largest eigenfrequency

$$\text{Conv}_{Ns}(N_q) = \sqrt{\frac{\lambda_{N_S}^S(N_q)}{\lambda_{N_S}^{S,\text{ref}}}}\,,$$

as a function of N_q, in which $\lambda_{N_S}^{S,\text{ref}}$ is the eigenvalue of rank N_S corresponding to the reference solution (see Section 2.1). It can be seen that a good approximation is obtained with $N_q = 500$. From now on, we use $N_q = 500$. An analysis of the computer resources for solving the elastic

Figure 2: Graph of function $N_q \mapsto \text{Conv}_{N_S}(N_q)$ showing the convergence of the highest normalized eigenfrequency that is considered with respect to N_q using the double projection algorithm.

Figure 3: CPU time required for solving the elastic eigenvalue problem using standard method and double projection method as a function of the dofs number defined in Table 1

Figure 4: RAM use for solving the elastic eigenvalue problem using standard method and double projection method as a function of the dofs number defined in Table 1.

eigenvalue problem has been performed with respect to the size of the finite element mesh, for both standard and double projection methods. Figures 3 and 4 display the graph of the CPU time and the graph of the maximum memory required for the computation for the values of n_{dof}^{system} given in Table 1. It can be seen that the RAM required for the computation with $n_{dof}^{system} = 362{,}000$

exceeds 264 GB when the standard method is used, implying an out of memory and a stop of computation. Consequently, the computation cannot be carried out for $n_{dof}^{system} > 196{,}000$. The double projection method proposed allows for solving the elastic eigenvalue problem for all the values of n_{dof}^{system} considered in Table 1. In addition, Figure 3 shows that the double projection yields a considerable gain for the CPU time with respect to the standard method.

5.3. Acoustic eigenvalue problem

Figure 5: CPU time required for solving the acoustic eigenvalue problem using standard method and subspace iteration method as a function of the dofs number defined in Table 1.

Figure 6: RAM use for solving the acoustic eigenvalue problem using standard method and subspace iteration method as a function of the dofs number defined in Table 1.

Figures 5 and 6 display the graph of the computational CPU time and the graph of the maximum memory used for solving the acoustic eigenvalue problem for all the values of n_{dof}^{system} considered in Table 1 using both standard and subspace iteration methods. It can be seen that, for the standard method, the RAM required for $n_{dof}^{system} = 850{,}000$ exceeds 264 GB, and consequently, could not be carried out. In comparison, the subspace iteration method allows for solving the acoustic eigenvalue problem for larger values of n_{dof}^{system} until $n_{dof}^{system} = 1{,}142{,}000$ but in counterpart generates a higher CPU-time consumption, which is compensated by the fact that the computation can effectively be done. Note that the CPU time with subspace iteration is correlated to the choice of the tolerance for the convergence and can be decreased if a less demanding tolerance is used.

5.4. Sloshing/capillarity eigenvalue problem

The double projection method requires to find the optimum size of the initial projection basis Φ_H^i in order to accurately solve the sloshing/capillarity eigenvalue problem and to ensure a low-computation time and RAM usage. Let $0 < \lambda_1^H(N_d, h_K) \leq \ldots \leq \lambda_{N_H}^H(N_d, h_K)$ be the sloshing/capillarity eigenvalues computed as explained in Section 4.3. A convergence analysis of the largest eigenvalue, $\lambda_{N_H}^H(N_d, h_K)$, is performed with respect to the size N_d of the initial subspace and to the height h_K of the small acoustic-fluid layer used for constructing the approximation K_F^a of K_F. Figure 7 displays the graph of

$$\mathrm{Conv}_{N_H}(N_d, h_K) = \sqrt{\frac{\lambda_{N_H}^H(N_d, h_K)}{\lambda_{N_H}^{H,\mathrm{ref}}}},$$

in which $\lambda_{N_H}^{H,\mathrm{ref}}$ is the eigenvalue of rank N_H corresponding to the reference solution (see Section 2.3). It can be seen that a good approximation is obtained for $N_d = 500$ and h_K corresponding to 7% of acoustic-fluid depth that is retained in the following.
The sloshing/capillarity eigenvalue problem is analyzed with respect to the number of dofs n_{dof}^{system}

Figure 7: For given h_K corresponding to 4%, 6%, and 7% of acoustic-fluid depth, graph of function $N_d \mapsto \text{Conv}_{N_H}(N_d, h_K)$ showing the convergence of the highest normalized eigenfrequency that is considered with respect to N_d using the double projection algorithm.

Figure 8: CPU time required for solving the sloshing/capillarity eigenvalue problem using standard method and double projection method as a function of the dofs number defined in Table 1.

Figure 9: RAM use for solving the sloshing/capillarity eigenvalue problem using standard method and double projection method as a function of the dofs number defined in Table 1 .

of the finite element mesh described in Table 1 for both standard and double projection methods. Figures 8 and 9 display the graph of the CPU time and the graph of the maximum RAM required for the computation. It can be seen that the RAM required for the computation with $n_{dof}^{system} = 850{,}000$ exceeds 264 GB when the standard method is used, implying an out of memory and a stop of computation. Consequently, the computation cannot be carried out for $n_{dof}^{system} > 578{,}000$. In comparison, the double projection method allows for solving the sloshing/capillarity eigenvalue problem for all the values of n_{dof}^{system} considered in Table 1. In addition Figure 8 shows that the double projection yields a considerable gain for the CPU time with respect to the standard method.

6. Conclusion

Non-standard algorithms have been proposed for solving generalized eigenvalue problems related to large-scale fluid-structure computational models that are simulated with mid-power computers. In this framework, the limitations of the computer resources are principally due to the RAM limitations. The double projection method and the subspace iteration method that are proposed allow for solving problems that cannot be treated with standard algorithms. In addition, the use of the double projection method not only allows for circumventing the RAM limitation but also allows for considerably decreasing the CPU time with respect to the standard algorithms. The algorithms proposed allow for computing a large scale fluid-structure computational model on mid-power computers but certainly, would allow for computing very large scale fluid-structure computational models on high-power computers. The substantial efforts performed in this work should be of interest for the fluid-structure community, which is confronted to computational resources limitations for analyzing large-scale fluid-structure computational models.

7. Acknowledgment

This work has been supported by DGA (Direction Générale de l'Armement) in France.

References

[1] D. C. Sorensen, Implicit application of polynomial filters in ak-step Arnoldi method, SIAM Journal on Matrix Analysis and Applications 13 (1) (1992) 357–385.

[2] R. B. Lehoucq, D. C. Sorensen, Deflation techniques for an implicitly restarted Arnoldi iteration, SIAM Journal on Matrix Analysis and Applications 17 (4) (1996) 789–821.
[3] B. N. Parlett, The Symmetric Eigenvalue Problem, SIAM, 1998.
[4] K.-J. Bathe, Finite Element Procedures, Klaus-Jurgen Bathe, 2006.
[5] Y. Saad, Numerical Methods for Large Eigenvalue Problems: Revised Edition, SIAM, 2011.
[6] G. H. Golub, C. F. Van Loan, Matrix Computations, 4th Edition, SIAM, 2013.
[7] F. Chatelin, Eigenvalues of Matrices: Revised Edition, SIAM, 2012.
[8] K.-J. Bathe, The subspace iteration method–Revisited, Computers & Structures 126 (2013) 177–183.
[9] V. Kalantzis, J. Kestyn, E. Polizzi, Y. Saad, Domain decomposition approaches for accelerating contour integration eigenvalue solvers for symmetric eigenvalue problems, Preprint, Dept. Computer Science and Engineering, University of Minnesota.
[10] R. P. Brent, F. T. Luk, The solution of singular-value and symmetric eigenvalue problems on multiprocessor arrays, SIAM Journal on Scientific and Statistical Computing 6 (1) (1985) 69–84.
[11] I. C. Ipsen, Y. Saad, The Impact of Parallel Architectures on the Solution of Eigenvalue Problems, Yale University, Department of Computer Science, 1985.
[12] J. J. Dongarra, D. C. Sorensen, A fully parallel algorithm for the symmetric eigenvalue problem, SIAM Journal on Scientific and Statistical Computing 8 (2) (1987) s139–s154.
[13] J. Gotze, S. Paul, M. Sauer, An efficient Jacobi-like algorithm for parallel eigenvalue computation, IEEE transactions on computers 42 (9) (1993) 1058–1065.
[14] J. J. Dongarra, D. W. Walker, Software libraries for linear algebra computations on high performance computers, SIAM review 37 (2) (1995) 151–180.
[15] F. Tisseur, J. Dongarra, A parallel divide and conquer algorithm for the symmetric eigenvalue problem on distributed memory architectures, SIAM Journal on Scientific Computing 20 (6) (1999) 2223–2236.
[16] K. Wu, H. Simon, A parallel Lanczos method for symmetric generalized eigenvalue problems, Computing and Visualization in Science 2 (1) (1999) 37–46.
[17] H. M. Aktulga, L. Lin, C. Haine, E. G. Ng, C. Yang, Parallel eigenvalue calculation based on multiple shift–invert Lanczos and contour integral based spectral projection method, Parallel Computing 40 (7) (2014) 195–212.
[18] V. Dolean, P. Jolivet, F. Nataf, An Introduction to Domain Decomposition Methods: Algorithms, Theory, and Parallel Implementation, SIAM, 2015.
[19] D. S. Watkins, QR-like algorithms for eigenvalue problems, Journal of Computational and Applied Mathematics 123 (1) (2000) 67–83.
[20] T. Katagiri, Y. Kanada, An efficient implementation of parallel eigenvalue computation for massively parallel processing, Parallel Computing 27 (14) (2001) 1831–1845.
[21] V. Hernández, J. E. Román, A. Tomás, Parallel Arnoldi eigensolvers with enhanced scalability via global communications rearrangement, Parallel Computing 33 (7) (2007) 521–540.
[22] D. Rodríguez, V. Theofilis, Massively parallel solution of the biglobal eigenvalue problem using dense linear algebra, AIAA Journal 47 (10) (2009) 2449.
[23] H. Imachi, T. Hoshi, Hybrid numerical solvers for massively parallel eigenvalue computations and their benchmark with electronic structure calculations, Journal of Information Processing 24 (1) (2016) 164–172.
[24] J. Choi, J. Demmel, I. Dhillon, J. Dongarra, S. Ostrouchov, A. Petitet, K. Stanley, D. Walker, R. C. Whaley, ScaLAPACK: A portable linear algebra library for distributed memory computers. Design issues and performance, in: International Workshop on Applied Parallel Computing, Springer, 1995, pp. 95–106.
[25] E. Anderson, Z. Bai, C. Bischof, S. Blackford, J. Demmel, J. Dongarra, J. Du Croz, A. Greenbaum, S. Hammarling, A. McKenney, D. Sorensen, LAPACK Users' Guide, 3rd Edition, Society for Industrial and Applied Mathematics, Philadelphia, PA, 1999.
[26] Z. Bai, J. Demmel, J. Dongarra, A. Ruhe, H. van der Vorst, Templates for the solution of algebraic eigenvalue problems: a practical guide, SIAM, 2000.
[27] H. J. Morand, R. Ohayon, Fluid Structure Interaction, John Wiley & Sons, New York, 1995.
[28] R. Ohayon, C. Soize, Advanced Computational Vibroacoustics: Reduced-Order Models and Uncertainty Quantification, Cambridge University Press, New York, 2014.
[29] R. Ohayon, C. Soize, Vibration of structures containing compressible liquids with surface tension and sloshing

effects. Reduced-order model, Computational Mechanics 55 (6) (2015) 1071–1078.
[30] L. G. Olson, K. J. Bathe, Analysis of fluid-structure interactions - a direct symmetric coupled formulation based on a fluid velocity potential, Computers & Structures 21 (1985) 21–32.
[31] P. Concus, R. Finn, On the behavior of a capillary surface in a wedge, Proceedings of the National Academy of Sciences of the United States of America 63 (2) (1969) 292.
[32] R. Finn, On the equations of capillarity, Journal of Mathematical Fluid Mechanics 3 (2) (2001) 139–151.
[33] R. Finn, The contact angle in capillarity, Physics of Fluids 18 (4) (2006) 047102.
[34] K. T. Kim, K. J. Bathe, The bathe subspace iteration method enriched by turning vectors, Computers & Structures 186 (2017) 11–21.
[35] D. D. Kana, U. S. Lindholm, H. N. Abramson, An Experimental Study of Liquid Instability in a Vibrating Elastic Tank, Southwest Research Institute, 1963.
[36] U. S. Lindholm, D. D. Kana, H. N. Abramson, Breathing vibrations of a circular cylindrical shell with an internal liquid, J. Aerospace Sci 29 (9) (1962) 1052–1059.