

A Parallel and Broadband Helmholtz FMBEM Model for Large-Scale Target Strength Modeling

Daniel R. Wilkes^{*‡}, Alec J. Duncan^{*} and Steffen Marburg[†]

^{*}*Centre for Marine Science and Technology
Curtin University, GPO Box U1987, Perth WA 6845, Australia*

[†]*Chair of Vibroacoustics of Vehicles and Machines
Technische Universität München, Boltzmannstraße 15
D-85748 Garching bei München, Germany*

[‡]*D.Wilkes@curtin.edu.au*

Received 11 October 2019

Accepted 10 February 2020

Published 30 July 2020

The Fast Multipole Boundary Element Method (FMBEM) reduces the $O(N^2)$ computational and memory complexity of the conventional BEM discretized with N boundary unknowns, to $O(N \log N)$ and $O(N)$, respectively. A number of massively parallel FMBEM models have been developed in the last decade or so for CPU, GPU and heterogeneous architectures, which are capable of utilizing hundreds of thousands of CPU cores to treat problems with billions of degrees of freedom (dof). On the opposite end of this spectrum, small-scale parallelization of the FMBEM to run on the typical workstation computers available to many researchers allows for a number of simplifications in the parallelization strategy. In this paper, a novel parallel broadband Helmholtz FMBEM model is presented, which utilizes a simple columnwise distribution scheme, element reordering and rowwise compression of data, to parallelize all stages of the fast multipole method (FMM) algorithm with a minimal communication overhead. The sparse BEM near-field and sparse approximate inverse preconditioner are also constructed and executed in parallel, while the flexible generalized minimum residual (fGMRES) solver has been modified to apply the FMBEM matrix-vector products and corresponding minimum residual convergence within the parallel environment. The algorithmic and memory complexities of the resulting parallel FMBEM model are shown to reaffirm the above estimates for both the serial and parallel configurations. The parallel efficiency (PE) of the FMBEM matrix-vector products and fGMRES solution for the present model is shown to be satisfactory; achieving PEs up to 92.3% and 74.1% in the fGMRES solution using 3 and 6 CPU cores respectively, when applied to models having $> 10^4$ dof per CPU core. The PE of the precalculation stages of the FMBEM — in particular the FMM precomputation stage which is largely unparallelized — reduces the overall PE of the FMBEM model; resulting in average efficiencies of 68.3% and 47% for the 3-core and 6-core models when treating problems with $> 10^4$ dof per CPU core.

This is an Open Access article published by World Scientific Publishing Company. It is distributed under the terms of the Creative Commons Attribution 4.0 (CC BY) License which permits use, distribution and reproduction in any medium, provided the original work is properly cited.

The present model is able to treat large-scale acoustic scattering problems involving up to 10^7 dof on a workstation computer equipped with 128 GB of RAM, while acoustic target strength (TS) results calculated up to 3 kHz for the BeTSSi II submarine model demonstrate its capabilities for large-scale TS modeling.

Keywords: FMBEM; broadband; parallel; target strength.

1. Introduction

The Fast Multipole Boundary Element Method (FMBEM) is one of the several ‘fast’ methods which reduces the $O(N^2)$ computational and memory complexity of the conventional BEM when solving for N boundary unknowns. Other methods include, for example, the pre-corrected FFT,¹ the hierarchical adaptive cross approximation (ACA) method² and hybrid FMBEM-ACA methods.³ The Fast Multipole Method (FMM) was originally developed by Rokhlin and Greengard for solving 2D Laplace Boundary Integral Equations (BIEs)⁴ and particle interaction problems,⁵ respectively. The FMM was applied to the 3D Helmholtz BEM by Coifman *et al.*⁶ A number of 3D Helmholtz FMBEM models have since been proposed, which can be broadly classified as either ‘high-frequency’ formulations which are based on the far-field signature functions and diagonal forms of Rokhlin’s translation methods,⁷ and ‘low-frequency’ formulations, which use the spherical basis functions and Rotation, Coaxial translation, Rotation (RCR)⁸ or plane wave expansion⁹ translation methods. Implementations of both the high- and low-frequency FMBEMs for acoustic scattering/radiation problems can be found in, for example, Refs. 10–12, while articles reviewing the FMBEM literature were published in 2002 by Nishimura¹³ and 2011 by Liu *et al.*¹⁴

Both the high- and low-frequency FMBEMs cannot be readily applied across a broad frequency range. The low-frequency FMBEM provides poor algorithmic scaling at higher frequencies due to the cost of the translation operations, while the recursion relations for the RCR translation method become unstable at high frequencies (although different methods^{15,16} have now been proposed to overcome this issue). Conversely, the multipole-to-local translations in the high-frequency FMBEM break down at lower frequencies.^{6,13} These issues have spurred the development of ‘broadband’ (or ‘wideband’) FMBEM models which combine the high- and low-frequency formulations to provide a stable solution and optimal algorithmic scaling at all frequencies. Jiang and Chew developed the first broadband FMBEM (for electromagnetic problems) which utilized the respective far-field signature functions and spherical basis functions in the high- and low-frequency regimes, respectively.¹⁷ A similar model using the plane wave expansion translation method in the low-frequency regime was proposed by Cheng *et al.*¹⁸ Wallén *et al.* appear to have first applied the broadband FMBEM to acoustic scattering problems¹⁹ in 2006, and now a number of different broadband formulations for acoustic scattering can be found in the literature.^{20–22} Alternatively, broadband FMBEM models which modify the high- or low-frequency FMBEMs to make them suitable at all frequencies have also been developed. Darve and Havé stabilized the high-frequency diagonal translation method at lower frequencies,²³ while Chaillat and Collino reduced the computational cost of the low-frequency plane wave expansions.²⁴

Parallelization of the FMM algorithm has been pursued since the methods invention: Greengard and Groppe developed a parallel CPU implementation of the FMM²⁵ in 1990, while Gumerov and Duraiswami implemented the first FMM implementation for the GPU architecture²⁶ in 2008. A number of massively parallel FMM algorithms have since been developed for CPU,²⁷ GPU²⁸ and heterogeneous (mixed CPU–GPU) architectures,^{29,30} with the more recent ‘extreme-scale’ models treating acoustic scattering problems with more than 2 billion dof on a supercomputer with $\sim 2 \times 10^5$ CPU cores.³¹ The exa-scale FMBEM models developed by Yokota and Barba have also been made freely available under open-source licenses.³² Application of the parallel FMBEM for the Helmholtz equation appears to have been largely pursued for electromagnetic scattering problems, although several parallel FMBEM models for acoustic analysis can also be found in the literature.^{21,33,34}

This paper presents a broadband Helmholtz FMBEM which combines the high- and low-frequency FMM expansion/translation methods^{18,20} for ‘large-scale’ acoustic scattering and target strength (TS) modeling. The model executes all stages of the FMBEM algorithm and iterative solver in parallel, and is designed for the multicore processors typically found in standard workstation computers which are available to many researchers. This type of small-scale parallelization allows for a number of simplifications in the data distribution and communication between workers, which otherwise require more complex implementations to scale and optimize for larger parallel systems. Furthermore, the FMM algorithm has been shown to provide very good parallel efficiency³⁵ (PE) for smaller scale problems ($\sim 10^5$ – 10^7 unknowns) distributed across a limited number of CPU cores ($\lesssim 64$). A novel approach for parallelizing the FMM algorithm is presented here, based on a columnwise distribution of the equivalent BEM matrix, element reordering and rowwise compression of data, which minimizes the required data transfer between workers. Using this approach, a standard workstation computer with an i7 hexacore CPU and 128 GB of RAM was able to solve acoustic scattering problems involving up to 10^7 dof. Problems of this computational size and parallel scaling will be of interest to many researchers with access to similar computer hardware. The present algorithm can also be readily applied to extend the capabilities of other advanced BEM models, such as for acoustics involving viscous and thermal losses,³⁶ Non-Negative Intensity,³⁷ or to accelerate the IGABEM for acoustic shape optimization.³⁸

The paper is organized as follows. Sections 2 and 3 introduce the Helmholtz BIE and the present broadband FMBEM algorithm, respectively. The low- and high-frequency multipole expansion and translation methods, transformation algorithm, truncation of expansions in the multilevel FMBEM and the computational cost of the various components of the broadband algorithm are discussed. Section 4 then describes the parallelization algorithm for the FMBEM model. Numerical results are presented in Sec. 5 which exhibit the algorithmic complexity and parallel performance of the model, and include TS modeling results for the BeTSSi II generic submarine hull. The conclusions and recommendations for future work are finally presented in Sec. 6.

2. The Helmholtz BEM

The Helmholtz BIE for acoustic scattering exterior to a closed boundary surface S is

$$\frac{1}{2}p(\mathbf{y}) = \int_S \left[\frac{\partial G(\mathbf{x}, \mathbf{y})}{\partial \mathbf{n}(\mathbf{x})} p(\mathbf{x}) - G(\mathbf{x}, \mathbf{y}) q(\mathbf{x}) \right] dS(\mathbf{x}) + p^{\text{inc}}(\mathbf{y}), \quad (1)$$

where p is the total surface pressure, $q = \frac{\partial p}{\partial \mathbf{n}}$ is its normal derivative, \mathbf{n} is the surface normal vector (pointing outward from S), p^{inc} is the incident acoustic field, and \mathbf{x} & \mathbf{y} are points on S (which is assumed to be locally smooth at \mathbf{y}). The 3D Helmholtz Green's function is as follows:

$$G(\mathbf{x}, \mathbf{y}) = \frac{e^{ikr}}{4\pi r}, \quad r = |\mathbf{y} - \mathbf{x}|, \quad (2)$$

where $k = 2\pi f/c$ is the wavenumber, f is the frequency, c is the sound speed and $i = \sqrt{-1}$. Equation (1) will break down at frequencies corresponding to the eigenfrequencies of the adjoint interior problem.^{40,41} A stable solution is provided at all frequencies by combining Eq. (1) with the normal derivative of this equation (evaluated at point \mathbf{y} on S) to give⁴²

$$\begin{aligned} \frac{1}{2}[p(\mathbf{y}) + \alpha q(\mathbf{y})] &= [M(\mathbf{x}, \mathbf{y}) + \alpha M'(\mathbf{x}, \mathbf{y})]p(\mathbf{x}) \\ &\quad - [L(\mathbf{x}, \mathbf{y}) + \alpha L'(\mathbf{x}, \mathbf{y})]q(\mathbf{x}) + [p^{\text{inc}}(\mathbf{y}) + \alpha q^{\text{inc}}(\mathbf{y})], \end{aligned} \quad (3)$$

where q^{inc} is the normal derivative of the incident acoustic field, α is the Burton–Miller coupling coefficient and the L , M , L' and M' integral terms are defined as follows:

$$L(\mathbf{x}, \mathbf{y})q(\mathbf{x}) = \int_S G(\mathbf{x}, \mathbf{y})q(\mathbf{x})dS(\mathbf{x}), \quad (4)$$

$$M(\mathbf{x}, \mathbf{y})p(\mathbf{x}) = \int_S \frac{\partial G(\mathbf{x}, \mathbf{y})}{\partial \mathbf{n}(\mathbf{x})} p(\mathbf{x})dS(\mathbf{x}), \quad (5)$$

$$L'(\mathbf{x}, \mathbf{y})q(\mathbf{x}) = \int_S \frac{\partial G(\mathbf{x}, \mathbf{y})}{\partial \mathbf{n}(\mathbf{y})} q(\mathbf{x})dS(\mathbf{x}), \quad (6)$$

$$M'(\mathbf{x}, \mathbf{y})p(\mathbf{x}) = \frac{\partial}{\partial \mathbf{n}(\mathbf{y})} \int_S \frac{\partial G(\mathbf{x}, \mathbf{y})}{\partial \mathbf{n}(\mathbf{x})} p(\mathbf{x})dS(\mathbf{x}). \quad (7)$$

Equation (3) will provide a unique solution at all frequencies provided $\text{Im}(\alpha) \neq 0$, and is chosen here to be proportional to $+\frac{i}{k}$ as recommended by Marburg.⁴³ The conventional BEM discretizes Eq. (3) using (for example) the collocation or Galerkin methods,⁴⁴ to yield a set of N equations involving N boundary unknowns. Both the computational and memory requirements for the iterative solution of the resulting matrix equation are proportional to $O(N^2)$ as the BEM coefficient matrix is fully populated. This is a result of the BEM

explicitly calculating every pairwise element interaction of unknowns on the discretized boundary mesh. The FMBEM instead treats interactions between well-separated groups of elements and this alleviates the need to explicitly construct the full coefficient matrix: instead the matrix-vector product is directly calculated. As a result, the FMBEM has a computational complexity of between $O(N^{1.5})$ and $O(N \log N)$ (depending on the FMM algorithm and frequency range of interest) and a memory complexity of $O(N)$.³⁹

3. The Broadband FMBEM Algorithm

The present broadband FMBEM model employs a switch between the functional representations of the multipole expansions in the low- and high-frequency regimes to provide a stable solution at all frequencies. In the low-frequency regime, the spherical basis functions^{20,39} and RCR method⁸ are used for the multipole expansions and translations respectively, while in the high-frequency regime the far-field signature functions and diagonal translation methods⁷ are employed. The forward and inverse spherical transform is used to convert between the low- and high-frequency representations. Figure 1 shows a schematic of the present broadband FMBEM algorithm, compared to a reference implementation.²⁰

The following sections present an overview of the theory for the broadband FMBEM model in both the low- and high-frequency regimes, as well as the transformation between the functional representations of the multipole expansions via the spherical transform. The choice of the truncation number for the multilevel FMBEM and the corresponding algorithmic cost of the various components of the broadband model are also discussed.

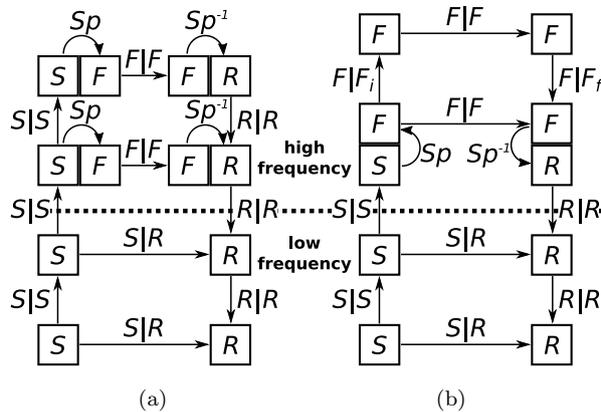


Fig. 1. The broadband FMBEM algorithm: Gumerov and Duraiswami's implementation²⁰ (a), and the present broadband algorithm (b). The boxes represent the expansion types: the S and R spherical basis functions and the far-field signature function F , while vectors indicate the translation operations: $S|S$, $R|R$ and $S|R$ translations of the spherical basis functions and $F|F$, $F|F_i$ and $F|F_f$ diagonal translations of the far-field signature functions. Interpolation and antepolation (adjoint-interpolation: filtering) operations are denoted by the i and f subscripts, respectively. The Sp and Sp^{-1} arrows indicate the forward and inverse spherical transforms for converting the spherical basis to far-field signature functions, and vice versa.

3.1. Broadband FMBEM: Low-frequency regime

3.1.1. Low-frequency FMBEM

The Helmholtz Green's function, Eq. (2), can be expressed as the sum of two independent series expansions constructed around the 'expansion center' \mathbf{c}

$$G(\mathbf{x}, \mathbf{y}) = ik \sum_{n=0}^{\infty} \sum_{m=-n}^n R_n^{-m}(\mathbf{y} - \mathbf{c}) S_n^m(\mathbf{x} - \mathbf{c}), \quad (8)$$

provided that the source and receiver points meet the 'well-separated' criterion $|\mathbf{y} - \mathbf{c}| < |\mathbf{x} - \mathbf{c}|$.³⁹ R and S denote the regular and singular spherical basis functions, which are defined in the spherical coordinate system $\mathbf{r} = r(\sin \theta \cos \phi, \sin \theta \sin \phi, \cos \theta)$ as

$$R_n^m(\mathbf{r}) = j_n(kr) Y_n^m(\theta, \phi), \quad S_n^m(\mathbf{r}) = h_n(kr) Y_n^m(\theta, \phi), \quad (9)$$

where $j_n(kr)$ and $h_n(kr)$ are, respectively, the spherical Bessel and Hankel functions of the first kind, $Y_n^m(\theta, \phi)$ is the spherical harmonic function

$$Y_n^m(\theta, \phi) = (-1)^m \sqrt{\frac{2n+1}{4\pi} \frac{(n-|m|)!}{(n+|m|)!}} P_n^{|m|}(\cos \theta) e^{im\phi}, \quad (10)$$

and $P_n^{|m|}$ is the associated Legendre function. The degree n and order m have integer ranges of $n \geq 0$ and $-n \leq m \leq n$, respectively.³⁹ Truncation of the infinite series expansions in Eq. (8) determines the accuracy of the FMM algorithm. Assuming a truncation number p , the L , M , L' and M' integral terms in Eq. (3) between two well-separated plane triangular elements containing the collocation points \mathbf{x} and \mathbf{y} at their centers can be approximated using piecewise constant unknowns as

$$\int_E L(\mathbf{x}, \mathbf{y}) q(\mathbf{x}) dE(\mathbf{x}) \approx ik A q(\mathbf{x}) \sum_{n=0}^p \sum_{m=-n}^n S_n^m(\mathbf{x} - \mathbf{c}) R_n^{-m}(\mathbf{y} - \mathbf{c}), \quad (11)$$

$$\int_E M(\mathbf{x}, \mathbf{y}) p(\mathbf{x}) dE(\mathbf{x}) \approx ik A p(\mathbf{x}) \sum_{n=0}^p \sum_{m=-n}^n [S_n^m(\mathbf{x} - \mathbf{c})]_{\mathbf{n}(\mathbf{x})} R_n^{-m}(\mathbf{y} - \mathbf{c}), \quad (12)$$

$$\int_E L'(\mathbf{x}, \mathbf{y}) q(\mathbf{x}) dE(\mathbf{x}) \approx ik A q(\mathbf{x}) \sum_{n=0}^p \sum_{m=-n}^n S_n^m(\mathbf{x} - \mathbf{c}) [R_n^{-m}(\mathbf{y} - \mathbf{c})]_{\mathbf{n}(\mathbf{y})}, \quad (13)$$

$$\int_E M'(\mathbf{x}, \mathbf{y}) p(\mathbf{x}) dE(\mathbf{x}) \approx ik A p(\mathbf{x}) \sum_{n=0}^p \sum_{m=-n}^n [S_n^m(\mathbf{x} - \mathbf{c})]_{\mathbf{n}(\mathbf{x})} [R_n^{-m}(\mathbf{y} - \mathbf{c})]_{\mathbf{n}(\mathbf{y})}, \quad (14)$$

where A is the area of the boundary element E to be integrated, and $[]_{\mathbf{n}(\mathbf{x})}$ and $[]_{\mathbf{n}(\mathbf{y})}$ denote the surface normal derivative of the (S, R) spherical basis functions³⁹ at \mathbf{x} and \mathbf{y} , respectively. The FMBEM treats element integrals between well-separated 'far-field' elements by constructing multipole expansions for local element groups about common expansion centers. Interactions between element groups via Eqs. (11)–(14) are applied by shifting or

‘translating’ the expansion centers to coincide. In the multilevel FMBEM a hierarchy of element groups with different sizes on each level is constructed using an octree structure. Elements which are spatially close to one another (i.e. not considered well-separated on the lowest level of the octree) cannot be treated with the FMM. These ‘near-field’ integrals are directly calculated and stored in a sparse BEM matrix using appropriate numerical quadrature rules for the regular,⁴⁵ near-singular⁴⁶ and singular⁴⁷ integrals.

3.1.2. Truncation of the multipole expansions

In the multilevel FMBEM, the truncation number p is selected on each octree level based on the size of the corresponding octree boxes. Several methods for determining p based on either empirical formulas^{6,48} or theoretical error bounds^{39,49,50} have been proposed. The present broadband model selects p using the formula proposed in Ref. 20

$$\begin{aligned}
p_l &= 1 - \frac{\log(\epsilon(1 - \sigma^{-1})^{3/2})}{\log(\sigma)}, \\
p_h &= ka_l + \frac{3 \log\left(\frac{1}{\epsilon}\right)^{2/3}}{2} (ka_l)^{1/3}, \\
p &= (p_l^4 + p_h^4)^{1/4},
\end{aligned} \tag{15}$$

where $\sigma = 2$, ϵ is the specified accuracy of the multipole expansion and a_l is half the diagonal box length on octree level l . Choosing $p = n - 1$ in Eq. (8) will yield p^2 (n, m) coefficients.

3.1.3. Translation of the spherical basis functions

In the low-frequency regime, the translation of the expansion center for an established set of S or R coefficients is conducted using the $R|R$, $S|S$ and $S|R$ translations

$$R_n^m(\mathbf{r} + \mathbf{t}) = \sum_{n'=0}^p \sum_{m'=-n'}^{n'} (R|R)(\mathbf{t})(R_{n'}^{m'}(\mathbf{r})), \tag{16}$$

$$S_n^m(\mathbf{r} + \mathbf{t}) = \sum_{n'=0}^p \sum_{m'=-n'}^{n'} (S|S)(\mathbf{t})(S_{n'}^{m'}(\mathbf{r})), \tag{17}$$

$$S_n^m(\mathbf{r} + \mathbf{t}) = \sum_{n'=0}^p \sum_{m'=-n'}^{n'} (S|R)(\mathbf{t})(R_{n'}^{m'}(\mathbf{r})), \tag{18}$$

where \mathbf{r} and \mathbf{t} are the respective expansion and translation vectors, and $|\mathbf{r}| > |\mathbf{t}|$ and $|\mathbf{r}| < |\mathbf{t}|$ for Eqs. (17) and (18), respectively.³⁹ The $S|S$ and $R|R$ translations are used to translate expansions between octree levels, while the $S|R$ translations are used to translate between well-separated expansions on each octree level. Direct application of Eqs. (16)–(18) will cost $O(p^4)$ operations to translate a truncated series with p^2 coefficients.⁸ The Rotation,

Co-axial translation, Rotation (RCR) method reduces this cost to $O(p^3)$ operations, while the required rotation and translation coefficients can be precalculated and stored in $O(p^3)$ operations.^{8,39} The low-frequency FMBEM thus has a $O(N^{1.5})$ algorithmic scaling and a $O(N)$ memory complexity.³⁹ Finally, it should be noted that the RCR translation method automatically incorporates the interpolation and antinterpolation operations required to vary the truncation number p between octree levels in the multilevel FMBEM.

3.2. Broadband FMBEM: High-frequency regime

3.2.1. High-frequency FMBEM

The Helmholtz Green's function, Eq. (2), can be equivalently expressed in the high-frequency regime using the far-field signature function defined on the unit sphere S^2 as

$$G(\mathbf{x}, \mathbf{y}) = \frac{ik}{(4\pi)^2} \int_{S^2} e^{iks \cdot (\mathbf{y} - \mathbf{c}_2)} (F | F)(\mathbf{c}_2 - \mathbf{c}_1) e^{iks \cdot (\mathbf{c}_1 - \mathbf{x})} dS(\mathbf{x}), \quad (19)$$

where the vector $\mathbf{y} - \mathbf{x}$ is separated as $\mathbf{c}_1 - \mathbf{x}$ and $\mathbf{y} - \mathbf{c}_2$ for the well-separated expansion centers \mathbf{c}_1 and \mathbf{c}_2 . $(F | F)(\mathbf{c}_2 - \mathbf{c}_1)$ is the multipole-to-local translation of the far-field signature function (equivalent to the $S | R$ translation in Sec. 3.1.3), and \mathbf{s} is an arbitrary location on the sphere. The far-field signature functions thus have the form

$$f(\theta, \phi) = e^{iks \cdot (\mathbf{c}_1 - \mathbf{x})}. \quad (20)$$

Analogous expressions to Eqs. (11)–(14) can thus be constructed with Eq. (20). The broadband FMBEM instead converts the spherical basis functions to the far-field signature functions by applying the forward spherical filter on the switching octree level between the low- and high-frequency regimes (see Fig. 1). Conversely, the inverse spherical filter transforms the signature functions to spherical basis functions in the downward pass of the FMM.

Numerical integration of the signature function in Eq. (19) is applied on S^2 using grid points defined in the angular spherical coordinates (θ_j, ϕ_k) for $j = 1, \dots, p$ and $k = 1, \dots, 2p$. Grid points have the following angular positions: $\theta_j = \cos^{-1} t_j$, $\phi_k = (k - 1)\frac{\pi}{p}$, and Gauss weights: $w_j^\theta = w_j$, $w_k^\phi = \frac{\pi}{p}$, where (t_n, w_n) are the points and weights for the n -point Gaussian quadrature rule. The quadrature weight at each spherical grid point is thus $w_j^\theta w_k^\phi$.

3.2.2. The spherical transform and signature function filtering

The forward spherical transform^{18,51} is used to convert the spherical basis functions ($F = S, R$) to the far-field signature function as follows:

$$\beta_j^m = \sum_{n=|m|}^p F_n^m(\mathbf{r}) \sqrt{\frac{2n+1}{4\pi} \frac{(n-|m|)!}{(n+|m|)!}} P_n^{|m|}(\cos \theta_j), \quad j = 1, \dots, \tilde{p}, \quad m = -j, \dots, j, \quad (21)$$

$$f(\theta_j, \phi_k) = \sum_{m=-n}^n \beta_j^m e^{im\phi_k}, \quad j = 1, \dots, \tilde{p}, \quad k = 1, \dots, 2\tilde{p}, \quad (22)$$

where Eq. (21) has a computational cost of $O(\tilde{p}^2 p)$ and Eq. (22) is applied using the FFT in $O(\tilde{p}^2 \log \tilde{p})$ operations. The inverse spherical transform is applied as

$$\beta_j^m = \frac{2\pi}{\tilde{p}} \sum_{k=1}^{2\tilde{p}} f(\theta_j, \phi_k) e^{-im\phi_k}, \quad j = 1, \dots, \tilde{p}, \quad m = -p, \dots, p, \quad (23)$$

$$F_n^m(\mathbf{r}) = \sum_{j=1}^{\tilde{p}} w_j \beta_j^m \sqrt{\frac{2n+1}{4\pi} \frac{(n-|m|)!}{(n+|m|)!}} P_n^{|m|}(\cos \theta_j), \quad n = 0, \dots, p, \quad m = -p, \dots, p, \quad (24)$$

where Eq. (23) is calculated using the FFT in $O(\tilde{p}^2 \log \tilde{p})$ operations and Eq. (24) has a $O(p^2 \tilde{p})$ algorithmic complexity. The forward and inverse spherical transforms have an algorithmic complexity of $O(p^3)$ and must each be applied once per octree box on the switching octree level between the low- and high-frequency regimes. In the high-frequency regime re-sampling of the far-field signature function is achieved by applying Eqs. (23)–(24) with $p = \tilde{p}$ followed by Eqs. (21)–(22) with $p \neq \tilde{p}$ to resample the far-field signature function onto a finer or coarser grid of points. Combining Eqs. (21) and (24) using the Christoffel–Darboux formula^{18,51} allows the combined equations to be directly applied in $O(p^3)$ operations. This cost can be reduced by approximating the calculation using either a one-dimensional FMM,¹⁸ nonequispaced FFT⁵¹ or fast Legendre transform.³⁹ In the present model the nonequispaced FFT (NFFT) is employed,⁵¹ yielding a $O(p^2 \log p)$ algorithmic complexity. However, in the present model the NFFT filter only becomes faster than the direct $O(p^3)$ filtering operation for relatively large truncation numbers ($p \gtrsim 500$) due to the larger overhead in applying the NFFT.⁵² The NFFT filtering also reduces the memory storage requirements of the spherical filters from $O(p^3)$ to $O(p)$: a significant reduction for large- p filters. Furthermore, these filters must be replicated across workers in the parallel model. Thus, in the present formulation the NFFT filter is utilized for $p \geq 250$.

3.2.3. Translation of the far-field signature functions

Translations of the far-field signature functions between octree levels (analogous to the $S|S$ and $R|R$ translations) can be applied for an arbitrary translation vector $\mathbf{c}_2 - \mathbf{c}_1$ as

$$f_2(\theta, \phi) = f_1(\theta, \phi) e^{iks \cdot (\mathbf{c}_2 - \mathbf{c}_1)}, \quad (25)$$

which has an $O(p^2)$ algorithmic complexity. The $F|F_i$ and $F|F_f$ diagonal translations between octree levels are thus applied in the upward pass by first interpolating the signature functions on the current level l and then translating to the ‘parent’ level $l - 1$, and in the downward pass by translating to the ‘child’ level $l + 1$ and then antepolating. The $F|F$ multipole-to-local diagonal expansions, $(F|F)(\mathbf{c}_2 - \mathbf{c}_1)$ in Eq. (19), can be expressed as⁷

$$(F|F)(\mathbf{c}_2 - \mathbf{c}_1) = \sum_{n=0}^{p^*} i^n (2n+1) h_n(k|\mathbf{c}_2 - \mathbf{c}_1|) P_n \left(\mathbf{s} \cdot \frac{\mathbf{c}_2 - \mathbf{c}_1}{|\mathbf{c}_2 - \mathbf{c}_1|} \right), \quad (26)$$

where P_n is the Legendre polynomial of degree n and the truncation number should be selected as $p^* = 2p$ to translate a signature function with degree p .^{18,20} The $F|F$

translations thus require a $(2p) \times (4p)$ grid of sample points, which can again be constructed via spherical filtering. In the present model the truncation numbers above the switching octree level are set to $2p$, to minimize the number of $O(p^3)$ filtering operations in the high-frequency regime. The $F|F$ translations can then be applied without having to resample to/from the finer translation grid. The disadvantage of this approach is that the $F|F_i$ and $F|F_f$ translations between octree levels are also applied on an over-sampled grid i.e. using a truncation number of $2p$ when p would suffice. However, the cost of applying these translations is $O(p^2)$ compared to the $O(p^3)$ filtering operations, and so the reduction of the number of filtering operations is prioritized to minimize the overall cost of the broadband FMBEM.

The $F|F$ diagonal expansion is applied to the far-field signature functions as

$$f_2(\theta, \phi) = (F|F)(\mathbf{c}_2 - \mathbf{c}_1)f_1(\theta, \phi), \quad (27)$$

which converts the multipole expansion centered at \mathbf{c}_1 to a local expansion centered at \mathbf{c}_2 (analogous to the low-frequency $S|R$ translation), at a cost of $O(p^2)$ operations. The direct calculation of the $F|F$ translation coefficients in Eq. (26) has a $O(p^3)$ computational cost. Here, an approximate $O(p^2)$ calculation method is employed,¹⁸ where the $F|F$ coefficients are tabulated as a function of the argument of the Legendre polynomial, and a nearest-neighbor interpolation is applied at run time to calculate the translation coefficients.

The high-frequency FMBEM thus has an algorithmic complexity between $O(N^{1.5})$ and $O(N \log N)$ depending on the combination of $O(p^3)$ and $O(p^2 \log p)$ filtering operations, and a memory complexity of $O(N)$. The combination of the low- and high-frequency FMBEMs utilized in the broadband model will yield an algorithmic complexity between the $O(N^{1.5})$ and $O(N \log N)$ limits, depending on the frequency of the particular problem.

4. Parallelization Scheme for the FMBEM

The main computational work of the standard BEM is due to the repeated matrix-vector products applied during the iterative solution. As the BEM explicitly builds and stores the full coefficient matrix, an obvious distribution scheme to parallelize the matrix-vector product using a number of CPU cores, or ‘workers’, is to distribute the BEM coefficient matrix as contiguous blocks of rows across the workers, and replicate the vector across all workers, as shown in Fig. 2(a). The output vector from this parallel operation is distributed across the workers in blocks and does not require any further inter-worker communication (e.g. ‘gathering operations’ across workers) to recover the distributed solution vector.

The FMBEM only explicitly constructs and stores the vector of unknowns, not the full coefficient matrix, and so the matrix distribution scheme cannot be directly applied. An alternative scheme for the FMBEM is to instead distribute the vector of unknowns. Each worker may then apply the FMM algorithm in parallel to the section of the vector stored locally on that worker. This scheme necessitates a columnwise distribution of the coefficient matrix (see Fig. 2(b)) which results in a full vector being output on each worker. Components from each of these vectors must then be gathered onto the correct worker in

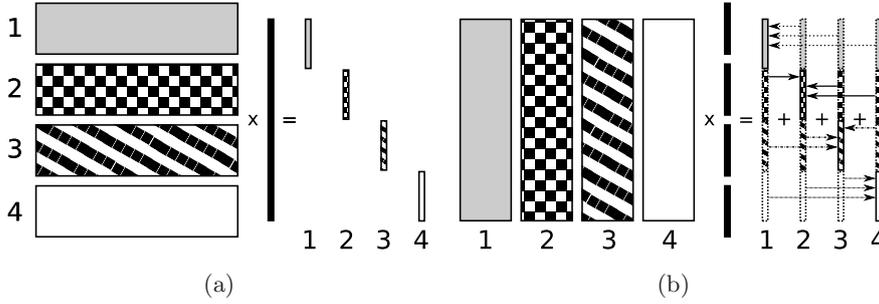


Fig. 2. General parallel distribution schemes for a matrix-vector product operation utilizing four workers: (a) rowwise distribution of the BEM coefficient matrix and (b) columnwise distribution of the equivalent full coefficient matrix for the FMBEM.

a subsequent inter-worker communication step to reconstruct the distributed matrix-vector product. The rowwise distribution requires the full vector of unknowns to be replicated across all n workers, while the columnwise distribution only requires a $1/n$ fraction of the vector to be transferred to each worker. However, in the subsequent inter-worker communication stage, each worker must both send and receive a $n-1/n$ fraction of the vector. Thus, the communication cost of the columnwise distribution is proportional to $2^{n-1}/n$, compared to the n -proportionality of the rowwise distribution scheme.

In this work, a novel parallelization strategy for the broadband FMBEM algorithm, based on the columnwise distribution of the equivalent FMBEM matrix is presented. This strategy utilizes element reordering and rowwise compression of sparse vectors to minimize the inter-worker communication. The FMBEM constructs local element groups to apply the multipole interactions, but the order of the elements within the full element list is arbitrary. Ordering the elements according to their octree box numbers (equivalent to the three-dimensional Morton order⁵³) yields a diagonally banded structure in the equivalent BEM matrix. An example of the element reordering for a spherical mesh, and the corresponding structure in the FMBEM sparse near-field matrix, is shown in Fig. 3.

The diagonal banding provided by the element reordering reduces the amount of inter-worker communication required in the columnwise distribution scheme. In this case the parallel matrix-vector product will yield sparsely populated vectors (Fig. 4(a)) which can be row-compressed to minimize the output vectors on each worker (Fig. 4(b)). As a result the main component of each vector is located on the correct worker and only a small component must be transferred between the workers.

The bandwidth of the equivalent FMBEM matrix varies with the octree level, where the lower octree levels involving small element groups have the narrowest bandwidth while larger element groups at higher octree levels are separated by larger distances and so will be located further from the main diagonal. The reduction in the inter-worker data transfer afforded by the element reordering thus becomes less effective as the octree level increases.

The following subsections describe the main stages of the parallel FMM algorithm (upward pass, downward pass and final summation), iterative solution and precalculation

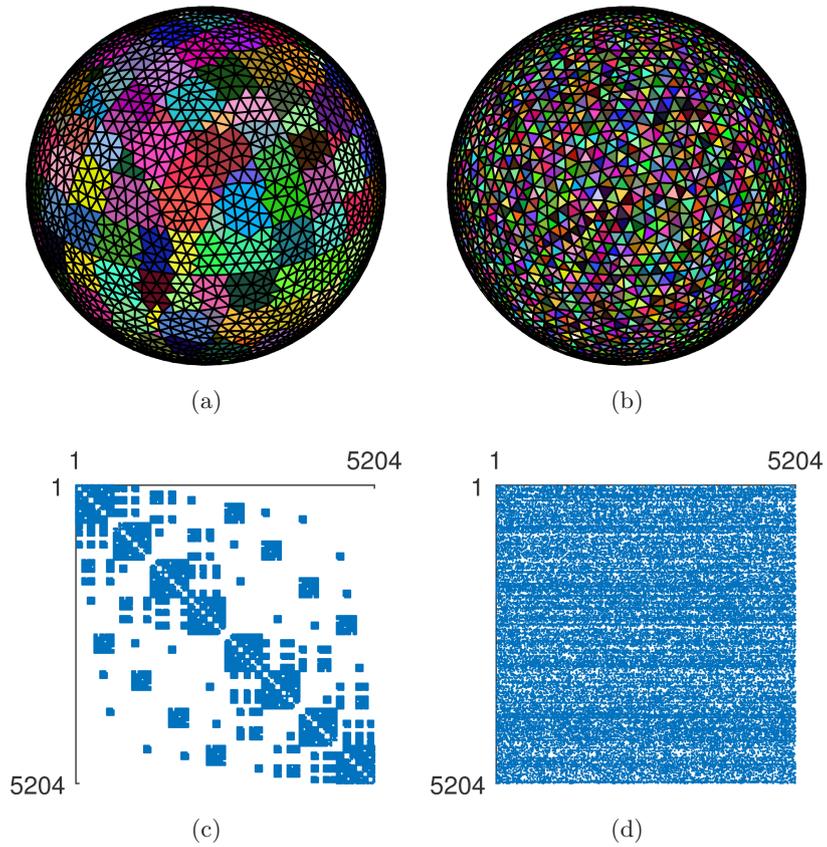


Fig. 3. Ordering the elements according to their octree box number (a) provides a diagonal banding in the equivalent BEM matrix (c). An arbitrary element order (b) (using the same element color ordering as in (a)) does not provide any structure in the equivalent matrix (d).

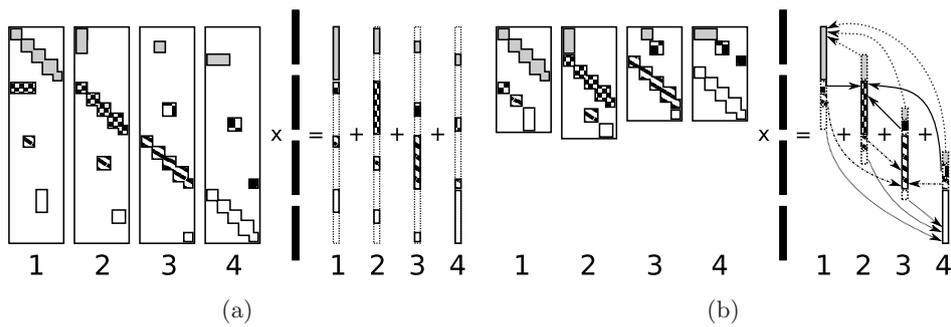


Fig. 4. The diagonal banding from the element reordering yields sparse matrix-vector products on each worker (a). Compression of the matrix blocks to remove the empty rows results in compressed output vectors (b), thus minimizing the subsequent data transfer required to reconstruct the final distributed vector.

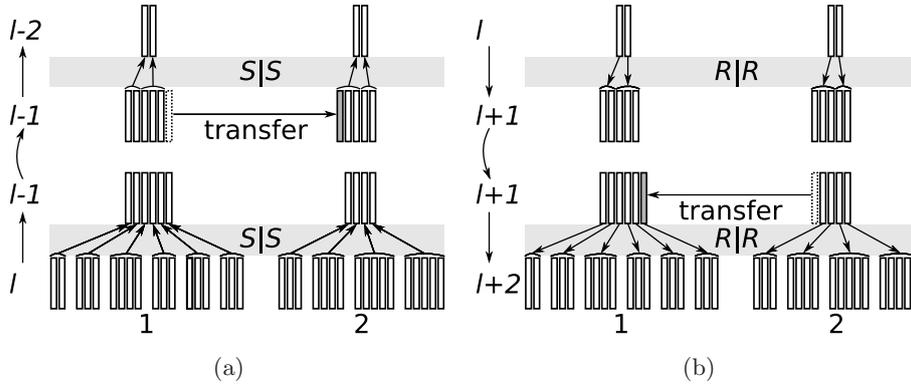


Fig. 5. The parallel $S|S$ and $R|R$ translations demonstrated for two workers. In the upward pass (a) the S expansion sets for each level l octree box (rectangles, grouped according to their parent octree box) are $S|S$ translated to the parent level ($l-1$) and summed into a single expansion set per box. Data transfer between workers is then applied to rebalance the distribution of expansion sets before applying the $S|S$ translations to level $l-2$. Conversely in the downward pass (b) the $R|R$ translations from level l to level $l+1$ is followed by an inter-worker communication step to reverse the data transfer applied in the upward pass.

procedures for the parallel implementation. Finally, an overview of the parallelization strategy which considers the load balancing of the workers for the different components of the broadband FMM algorithm is discussed. Full details on the implementation of the FMM for the Helmholtz equation can be found in, for example, the textbooks on the method.^{39,55}

4.1. Parallel FMBEM: Upward pass

The S -type spherical basis functions are calculated for every element in parallel using an approximately equal partition. The exact distribution is selected to ensure that (a) all elements contained within the same box on the lowest octree level l reside on the same worker, and (b), that each set of boxes with same parent box reside on the same worker (Fig. 5(a)). The expansions are first multiplied by the vector of unknown coefficients and combined into a single expansion set per octree box. Each set of expansions is then $S|S$ translated to the center of the larger parent boxes on level $l-1$ and again combined into a single set per parent octree box. At this stage the distribution of expansions sets across the workers may be somewhat nonuniform. An inter-worker data transfer is thus applied to rebalance the number of S expansions sets across the workers. Again the repartition of the data is chosen to ensure that the sets of $l-1$ box expansions with a common parent box on level $l-2$ are retained on the same worker. This process is repeated up to octree level 2, yielding an approximately equal distribution of S expansion sets on every octree level.

The parallel implementation of the upward pass follows the same procedure in both the low- and high-frequency regimes using the spherical basis functions and far-field signature functions, respectively. At the switching octree level the spherical transform is applied before the $S|S$ translations to the parent octree level, as indicated in Fig. 1(b).

4.2. Parallel FMBEM: Downward pass

The downward pass of the FMM algorithm applies the interactions between all well-separated boxes (element groups) on every octree level, beginning at $l = 2$. The S expansion sets for all well-separated ‘source’ boxes are $S|R$ translated to each ‘receiver’ box; the translations also transform the S -type expansions to R expansions. The R expansions from all well-separated boxes are each summed into a single set per receiver box and then $R|R$ translated from the parent box centers on level l to each of the children boxes on level $l + 1$. This process is then repeated on subsequent octree levels but with the additional step that the sets of R expansions from the parent octree level l are first combined with the R expansions on the current level, $l + 1$, before applying the $R|R$ translations to level $l + 2$.

In the parallel algorithm each worker only stores part of the full set of box S expansions on each octree level, whereas the well-separated interactions may involve source boxes which are stored across several workers. However, the element reordering procedure ensures that the boxes which are spatially close to one another will generally reside on the same worker, as shown in Fig. 6(a). Therefore, in the parallel model each worker applies all $S|R$ translations and summations for the set of source boxes residing on that worker to all receiver boxes on the octree level, including those residing on other workers. An inter-worker communication step is then applied to transfer the partial sets of summed R expansions calculated on each worker to the receiver boxes residing on other workers (see Fig. 6(b)). After the communication step each worker contains the complete sets of R expansions for all of the boxes residing on that worker. The $R|R$ translations are then applied on each worker and another data transfer is applied to reverse that conducted in the upward pass (see Fig. 5(b)). This reversal ensures that the data partition of the level l R expansions will match that on the below ($l + 1$) octree level (determined by the partition of the S expansions) and so the expansions can be combined in the subsequent stage of the downward pass.

The parallel implementation of the downward pass in the broadband model follows the same procedure for both the low- and high-frequency regimes. At the switching octree level the far-field signature functions are converted to the spherical basis functions before applying the downward $R|R$ translations to the child octree level (see Fig. 1).

The present model employs translation stencils²⁶ in both the low- and high-frequency regimes for sets of eight, four and two receiver boxes, to reduce the number of applied translations from 189 $S|R$ to 71 $S|R$ and 4 $R|R$ translations per receiver box.⁵⁴ The required $R|R$ translations for the stencils are applied on the same octree level (from the center point of the local receiver group to each of the individual boxes), and so do not require any resampling operations. To maximize the computational savings from the eight-box stencils the $S|R$ translations are applied on the parent octree level and then combined with the parent-level R expansions before applying the $R|R$ translations to the child octree level as part of the standard procedure for the downward pass. In the low-frequency regime the spherical basis functions with different truncation numbers (i.e. on the parent/child octree levels) can be directly combined without requiring any additional operations. In the high-frequency regime the far-field signature functions must be filtered to combine the

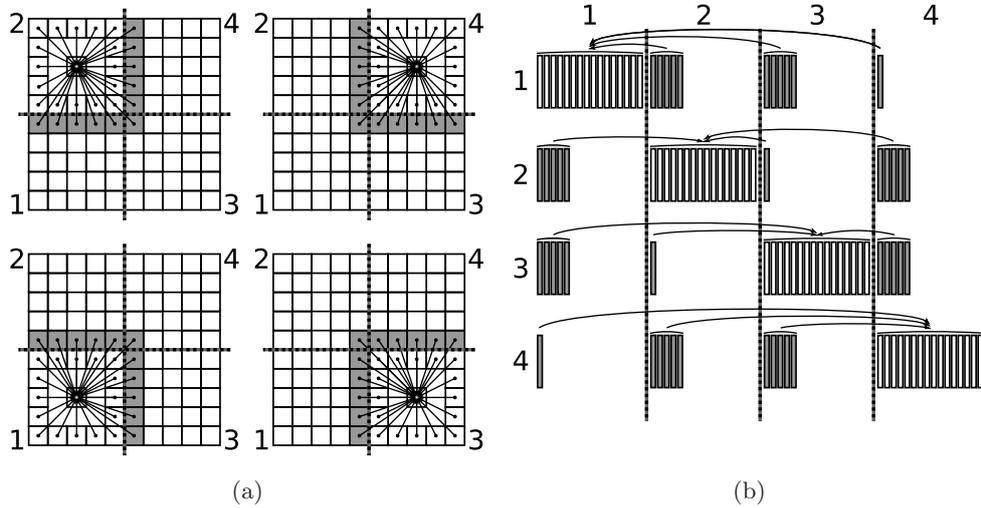


Fig. 6. The $S|R$ translations in the parallel scheme between sets of well-separated source boxes and four arbitrary receiver boxes in a two-dimensional domain (a) utilizing four workers. Octree boxes are represented by squares, $S|R$ translations by lines between box centers and the partition of the boxes amongst the four workers is indicated by the dashed lines. The partitioning scheme ensures that the majority of the set of well-separated boxes for a particular receiver box will be stored on the same worker (white translation boxes). The $S|R$ translations for the well-separated boxes stored on other workers (gray boxes) are calculated on those workers. An inter-worker communication step is then applied to transfer the sets of R expansions calculated on the other workers back to the worker which contains the receiver box (b).

signature functions with differing truncation numbers. Therefore, in the present model the eight-box stencils in the high-frequency regime are applied on the current octree level and then $R|R$ translated from the center of the eight-box receiver group to each box. This increases the cost of the translation stencils in the high-frequency regime to 71 $S|R$ and 5 $R|R$ translations per receiver box but avoids additional $O(p^3)$ filtering operations.

4.3. Parallel FMBEM: Final summation

At the conclusion of the downward pass the R expansions for each box on the lowest octree level represent the integrals for all boundary elements except for those residing in the nearest neighbor boxes. Elements contained within these boxes are directly applied via the conventional BEM (i.e. as explicit pairwise element interactions). In the present model, these element integrals are precalculated and stored in a distributed sparse matrix using the columnwise distribution and row compression scheme indicated in Fig. 4(b). Thus, the final summation proceeds by applying Eqs. (11)–(14) using the final sets of R expansions for each receiver box and the corresponding S expansions for each element contained within the box, to recover the far-field part of the discretized boundary integrals. The near-field element integrals are calculated on each worker via a parallel sparse-matrix vector product and added to the far-field result. The final result of the FMBEM procedure is the equivalent matrix-vector product between the BEM coefficient matrix and vector of unknowns.

4.4. *Parallel FMBEM: Iterative solution*

The FMBEM can be combined with an iterative solution method to solve large-scale acoustic problems. The GMRES method has been shown to be an effective iterative solver for acoustic problems modeled with both the BEM⁵⁶ and FMBEM.⁵⁷ The present model uses a flexible GMRES⁵⁸ solver, where a variable preconditioner is applied in each iteration of the solution. The fGMRES is implemented in the present model as follows: for each iteration of the main ‘outer’ GMRES, an inner GMRES first solves the problem using the current iteration of the solution vector from the outer GMRES. The resulting ‘preconditioned’ solution vector is then used in the next outer GMRES iteration. The inner GMRES uses a fast low accuracy FMBEM²⁰ which is solved to a coarse convergence tolerance, and so the inner GMRES acts as a fast approximate preconditioner. Conversely the outer GMRES is solved to a fine convergence tolerance using the full accuracy fast multipole expansions.

Furthermore, the inner GMRES loop is also preconditioned using a sparse approximate inverse (SAI) preconditioner⁵⁹ constructed from the sparse matrix of near-field integrals. The SAI preconditioner approximates the inverse of the BEM sparse near-field matrix by calculating a local inverse for each boundary element using the BEM coefficients for that element and its geometrical neighboring elements. This results in a small system of coefficients for each boundary element which can be directly inverted using standard inverse techniques for a small computational cost. The vectors of matrix inverse coefficients for each element are then constructed into a global sparse preconditioner (SAI) matrix.

The present parallel fGMRES solver replicates the main GMRES operations across the workers to locally update the distributed solution vector on each worker with minimal inter-worker communication. The only part of the GMRES algorithm which requires inter-worker communication is the calculation of the Euclidean norm of the solution vector in each iteration. This is calculated in parts on each worker by taking the inner product of the complex solution (stored in parts across the workers), gathering the results — a single complex number per worker — into a vector replicated on every worker and then calculating the magnitude of this vector. The other stages of the GMRES solution (modified Gram–Schmidt orthonormalization, QR decomposition, etc.) are then calculated in parallel across the workers for each section of the distributed solution vector. Similarly the SAI preconditioner matrix is precalculated in parallel and stored as a distributed sparse matrix (using the columnwise distribution and row compression scheme indicated in Fig. 4(b)) to apply the SAI matrix-vector product in parallel. This approach minimizes the inter-worker communication required during the fGMRES solution, as the distributed components of the solution vector do not need to be copied to and from the host worker in each outer/inner GMRES iteration. Furthermore, the initialization of the parallel calculation environment must only be done once for the entire fGMRES solution. Once the outer GMRES has converged to the specified convergence tolerance, the final distributed solution vector can be gathered on the host worker for post-processing. It has been shown⁶⁰ that redundantly replicating the serial operations of the fGMRES reduces the PE. This is in contrast to the present model, where replicating these operations yielded an improved PE compared

to the additional communication overhead (and corresponding reduction in PE) required to parallelize all stages of the fGMRES. These differences are likely due to the relative communication overheads associated with the different programming languages.

Finally, it should be noted that the standard GMRES solver algorithm (as implemented here) will break down for certain problem configurations which yield successive Krylov vectors with rapidly increasing magnitudes⁶¹ and thus a poorly conditioned Krylov subspace. Improved GMRES methods have been developed to alleviate this issue,⁶¹ and this is an obvious avenue for future development of the present model.

4.5. *Parallel FMBEM: Precalculation stage*

The precalculation stage of the parallel FMBEM model constructs the octree structure, multipole expansions and translation coefficients for each level of the octree. The sparse distributed near-field BEM and SAI matrices are also precalculated and stored.

In the present model the S , R , and normal derivative of the S -type expansions are precalculated and stored in parallel across the workers. The octree structure is then constructed on the host worker (i.e. as a serial process), and both the level-dependent truncation numbers and switching octree level between the low- and high-frequency regimes are determined by the problem frequency and the specified tolerances of the full and preconditioner multipole expansions.²⁰ The sets of unique $S|S$ and $R|R$ multipole translation coefficients — totalling 8 unique sets per octree level (from the children to the parent box center and vice versa) — are calculated on the host and transferred to each worker. Similarly, the full sets of unique $S|R$ translations are calculated for each octree level on the host and replicated to each worker. There are 316 unique $S|R$ translation vectors per octree level in the complete set of well-separated boxes. The translation stencils increase the total number of uniquely defined translation vectors to several hundred, as the centers of the eight, four and two box groups can be located at the corners, edges or side faces of the different group of receiver boxes. In the low-frequency regime the sets of RCR translation coefficients are only calculated and stored for the full accuracy FMBEM, while a subset of these coefficients can be directly utilized for the smaller truncation numbers in the preconditioner FMBEM. In the high-frequency regime separate sets of translation coefficients must be calculated and stored for the full accuracy and preconditioner FMBEM. Finally, both the sparse matrix of near-field integrals and the SAI preconditioner matrix are calculated in parallel across the workers and stored in the column distributed and row-compressed format.

4.6. *Parallel FMBEM: Choice of parallelization strategy*

The present parallelization strategy prioritizes the equal distribution of the octree boxes across the workers, in lieu of balancing the numbers of $S|R$ translations applied by each worker. This design choice balances the more costly filtering operations for the far-field signature functions, which have an $O(p^3)$ or $O(p^2 \log p)$ computational cost, compared to the $O(p^2)$ translation operations in the high-frequency regime. In the low-frequency regime both the filtering and translation operations have the same $O(p^3)$ computational cost. In

practice the number of $S|R$ translations per worker varies by $\sim 10\%$ or less for boundary element meshes with relatively uniform element distributions whose discretization is dictated by the acoustic wavelength.⁶² This distribution scheme may become suboptimal for highly nonuniform meshes treated via adaptive octree structures,³⁹ or for highly parallel (many core) implementations, where the smaller numbers of $S|R$ translation per worker may become significantly unbalanced. A more efficient strategy could be to prioritize the distribution of the $S|R$ translations in the low-frequency regime, and the number of octree boxes (filtering operations) in the high-frequency regime. However, this approach would necessitate increased worker communication and so the overall PE of the algorithm would depend on the relative cost of the increased data transfer compared to the optimized load balancing between workers. It will be shown in the next section that the data communication for the present model is a relatively small component of the total calculation time for meshes which have $> 10^4$ dof per CPU core.

5. Numerical Results

The parallel broadband FMBEM model has been implemented in MATLAB. An analysis of the algorithmic/memory complexities and PE of the model is presented in Sec. 5.1. Target-strength modeling of the BeTTSi II generic submarine hull model is presented in Sec. 5.2. The outer and inner loops of the fGMRES were set to convergence tolerances of 10^{-4} and 0.15, respectively, and the truncation numbers were selected via Eq. (15) to provide multipole expansion errors of 10^{-4} in the ‘full accuracy’ (outer) and 0.2 in the preconditioner (inner) FMBEM calculations. All calculations were performed on a desktop workstation equipped with an Intel i7 hexacore CPU running at 3.30 GHz, and 128 GB of RAM. It should be noted that Intel’s automatic ‘turbo boost’ feature for this particular processor allows a single CPU core to operate up to 3.60 GHz; allowing relatively faster serial calculations and thus lower efficiencies for the parallel models (a theoretical maximum of 91.67% PE). However, it is shown in Sec. 5.1.2 that some parts of the model achieve close to linear scaling, and so the effect of the turbo boost is assumed to be relatively small.

5.1. Algorithmic performance and efficiency of parallel implementation

5.1.1. Algorithmic and memory complexity of FMBEM model

This section presents the algorithmic complexity and PE of the present parallel broadband FMBEM model. The problem configuration for a plane-wave scattering from spherical surface under a rigid (Neumann) boundary condition, which has an analytic solution, is used to characterize the model’s performance. Solutions are calculated over a range of wavenumbers using mesh discretizations that provide a minimum of 10 constant elements per acoustic wavelength (that is, the solution frequency increases with the number of dof to provide a constant element per wavelength discretization). The boundary element meshes range from 10^3 to 10^7 elements/dofs, a maximum kD value of 1299 (where k is the wavenumber and D is the size of the computational domain), and are solved using 1, 3 and 6 CPU cores. Model

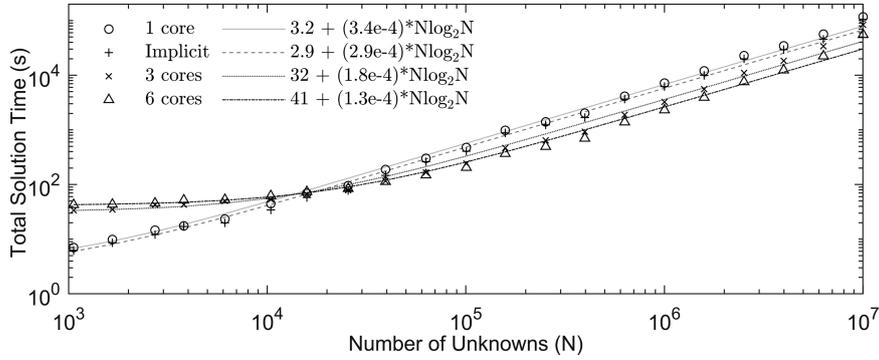


Fig. 7. The total calculation time of the parallel broadband FMBEM as a function of N , for a constant number of elements per wavelength. Lines of best fit are shown for each set of model results.

results from Matlab's implicit parallelization of its own inbuilt functions are also presented for reference (denoted as 'Implicit'). Relative error norms for the model solutions varied between 1.18% and 6.75%, with a mean value of 2.16%. The largest solution error occurred for the 10^7 element model, where a maximum of only 2 elements per lowest octree box was used in the FMBEM calculation due to the memory limitations for running this model.

Figure 7 presents the total calculation time of the parallel broadband FMBEM as a function of N , for a constant number of elements per wavelength. The total calculation time includes all stages of the FMBEM model, including the set-up of the octree structure, near-field and SAI precalculations, and fGMRES solution time. Lines of best fit are also presented in the figure for each serial/parallel model configuration. The broadband model exhibits the expected $N \log_2 N$ algorithmic complexity.²⁰ Increasing the number of CPU cores in the parallel solution reduces the scaling factor of the $N \log_2 N$ slope (i.e. a linear reduction in the calculation time) while the constant y -intercept increases due to the increased overhead in setting up the parallel algorithm across multiple cores. For the present problem configuration the parallel model becomes faster than the single-threaded model for $N \sim 10^4$.

Figure 8 shows the calculation time required for a single matrix-vector product (FMBEM far-field and sparse BEM near-field) using the full accuracy multipole expansions (expansion error = 10^{-4}). The lines of best fit again show an algorithmic complexity proportional to $N \log_2 N$. Furthermore, the parallel matrix-vector calculations are faster than the serial model results for all problem sizes (and similarly for the 'preconditioner accuracy' matrix-vector calculation times; not shown), indicating that the slower total calculation time of the parallel model for $N < 10^4$ is due to the set-up of the model.

The memory storage requirements of the model are presented in Fig. 9 as a function of N . In this case the results for the MATLAB implicit parallelization are identical to the 1-core results and so are not shown. The fitted lines indicate a memory scaling proportional to N , which agrees with the theoretical prediction for the broadband FMBEM model.³⁹ The parallel model requires more memory as the number of CPU cores is increased, but this inversely proportional to N (as indicated by the different slopes of the fitted lines). This is

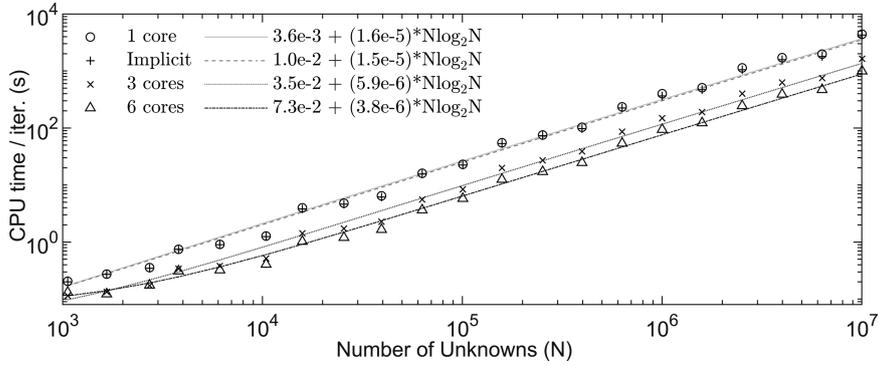


Fig. 8. Averaged calculation time for a single FMBEM matrix-vector product as a function of N , for a constant number of elements per wavelength. Lines of best fit are shown for each set of model results.

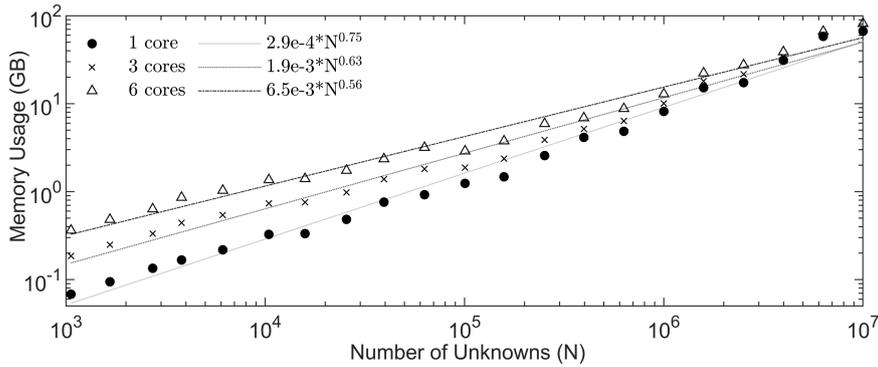


Fig. 9. The memory storage requirements of the parallel broadband FMBEM in (GB) as a function of N .

due to the memory footprint being dominated by the translation and filtering matrices at lower frequencies, which are replicated across the workers, and by the multipole expansions and sparse near-field matrix at higher frequencies, which are distributed across the workers. Finally, it should be noted that the instantaneous peak memory usage of the FMBEM is somewhat larger than that indicated in Fig. 9 (typically by 10–20%) as various temporary arrays are built during execution of the algorithm.

The results presented in this section thus reaffirm the expected algorithmic and memory complexities of the broadband Helmholtz FMBEM. Comparative results for the parallel FMBEM model indicate that the present parallelization strategy does not affect the overall $O(N \log_2 N)$ complexity, only the scaling and offset parameters of the line-fit (corresponding to the total calculation and initialization times for the parallel models).

5.1.2. Parallel efficiency of the FMBEM model

This section presents results for the PE of the FMBEM, for both the total solution time and the time required to perform the far-field part of a single matrix-vector product using

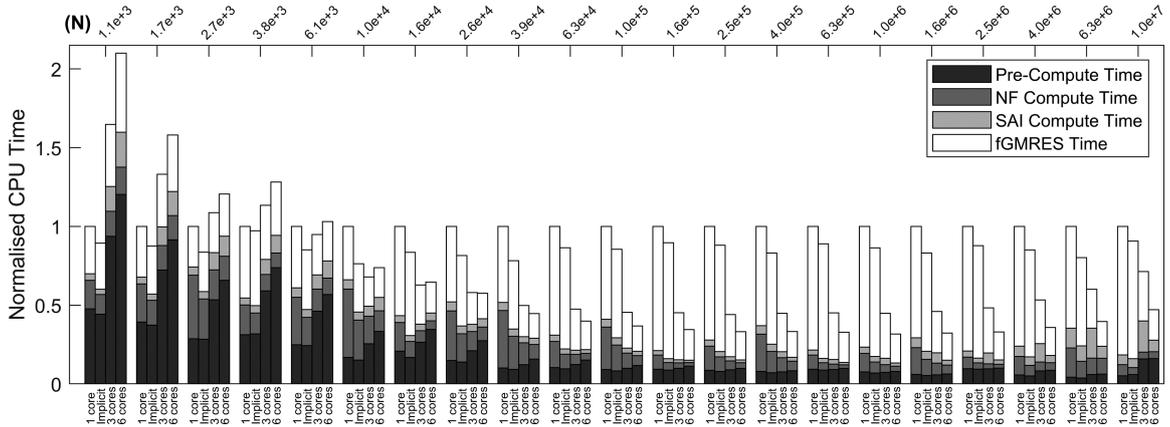


Fig. 10. Components of the total calculation time of the parallel broadband FMBEM as a function of N , for a constant number of elements per wavelength. The bottom and top x -axes indicate the number of CPU cores and the approximate number of dof for each group of bars, respectively.

the FMM algorithm. The FMM matrix-vector results were averaged over a number of calculations (100 iterations for problems with $N \leq 10^6$ and 25 iterations for problems with $N > 10^6$) inside a single call to Matlab’s ‘spmd’ parallel execution environment to minimize the time delay for initialising MATLAB’s parallel environment. The calculation time for the sparse near-field matrix-vector product was negligibly small; ranging from 10^{-2} – 10^{-4} % of the total matrix-vector calculation time, and so was not included in the timing results. This contrasts with some other Helmholtz FMBEM models in the literature,²⁶ where the near-field calculation contributed a significant cost to the total calculation time.

The PE for the total calculation time of the FMBEM model as a function of N , is presented in Fig. 10. Each stack of bars indicates the constituent calculation times for the main components of the FMBEM algorithm — the FMM precomputation, BEM sparse near-field, SAI preconditioner and fGMRES calculation time — versus N , using different numbers of CPU cores for the parallel algorithm (indicated by the groups of bars). The computational time for each set of solutions has been normalized relative to the corresponding 1-core solution for each problem size. Table 1 presents the corresponding parallel efficiencies for the components of the FMBEM algorithm versus N . The PE ratio is defined as $\frac{\text{Time}_{1\text{core}}}{(\text{Time}_{C\text{core}} \times C)}$, where C is the number of CPU cores used in the parallel calculation and ‘Time_{1core}’ and ‘Time_{Ccore}’ are the respective serial and parallel calculation times being compared.

Figure 10 shows that the total calculation time of the parallel models is dominated by the FMM precomputation time at smaller values of N (lower frequencies). This is due to the fact that the majority of operations in this stage are performed in serial and then transferred to the workers (thus making this stage slower than the single-threaded model). In the case of the smaller models the data distribution cost is dominated by the constant overhead in initiating the parallel operations. At higher frequencies, the parallel initialization cost is relatively small compared to the larger data transfer, and so the precomputation scaling improves as N increases. The parallel scaling from the near-field and SAI calculations

Table 1. Parallel efficiencies of the components of the FMBEM algorithm versus N for the 3-core and 6-core solutions. All PE values are presented as percentages.

N	Total Time		FMM Pre.		NF Compute		SAI Compute		fGMRES	
	3-Core	6-Core	3-Core	6-Core	3-Core	6-Core	3-Core	6-Core	3-Core	6-Core
1.1×10^3	20.2	7.9	17.0	6.6	37.8	17.4	8.8	3.1	25.4	10.0
1.7×10^3	25.0	10.5	18.1	7.2	52.0	26.3	12.2	4.8	32.1	14.9
2.7×10^3	30.7	13.8	17.9	7.3	70.1	44.2	15.6	6.8	33.8	16.0
3.8×10^3	29.4	13.0	17.6	7.0	60.3	33.9	15.5	6.6	44.0	22.3
6.1×10^3	35.1	16.2	17.9	7.3	71.7	48.1	22.3	9.2	50.7	26.0
1.0×10^4	49.0	22.6	22.0	8.4	82.5	55.1	31.2	11.7	60.9	29.9
1.6×10^4	53.1	25.8	26.2	10.0	82.9	55.8	35.0	14.7	76.0	48.2
2.6×10^4	57.5	28.9	23.5	9.0	86.0	60.5	41.9	18.3	79.2	49.2
3.9×10^4	67.0	37.3	27.6	10.7	87.8	65.5	44.7	21.7	81.2	51.3
6.3×10^4	70.2	41.8	28.1	11.5	85.9	64.0	49.4	28.2	88.4	63.6
1.0×10^5	73.3	45.3	30.6	13.0	93.3	71.9	51.5	29.4	86.5	61.1
1.6×10^5	73.7	48.3	31.2	13.8	86.6	64.7	48.8	31.3	88.0	67.5
2.5×10^5	75.7	50.2	31.8	14.5	91.7	71.1	49.2	36.5	89.8	67.0
4.0×10^5	74.3	50.0	34.4	15.7	87.2	65.1	47.0	35.7	86.3	64.3
6.3×10^5	73.8	50.7	34.0	15.8	85.9	64.4	36.8	35.1	88.4	68.2
1.0×10^6	74.4	52.6	34.6	16.1	80.9	57.7	32.4	34.1	89.6	69.3
1.6×10^6	72.4	51.5	34.6	16.0	76.7	50.0	31.7	33.4	89.8	68.3
2.5×10^6	69.0	50.5	33.6	16.1	77.5	50.1	18.5	23.7	92.3	74.1
4.0×10^6	62.5	46.7	22.9	11.0	69.2	40.1	18.9	24.3	91.3	71.0
6.3×10^6	55.5	42.0	24.0	11.5	59.2	30.7	21.9	27.4	87.3	68.2
1.0×10^7	46.7	35.4	10.8	5.3	54.6	27.3	10.5	14.2	86.5	70.2

improve as N increases as they are performed in parallel and require minimal data-transfer between workers: becoming faster than the serial model for relatively small problem sizes. However, these gains appear to scale poorly with the number of cores: exhibiting average parallel efficiencies of 50.7% and 21.4% for the 6-core near-field and SAI calculations, respectively (see Table 1). The numerical results for the present model configuration indicate that the fGMRES solution becomes the dominant computational cost for problems involving $\gtrsim 3.9 \times 10^4$ dof. The fGMRES solution also exhibits good PE — up to 92.3% in the best case — due to the large computational cost and minimized data-transfer for this part of the model. The average PE of the parallel FMBEM model is 68.3% and 47% for the 3-core and 6-core models respectively, for problems with $> 10^4$ dof per core. The poorer PE of the set-up stages of the model clearly limit the overall PE achieved. Future work will focus on improving the PE of these parts of the model.

Finally, it should be noted that the largest few models exhausted the available RAM during the precomputation, as evidenced by the relatively slower set-up calculation times (due to file paging) exhibited by those models. The other stages of the analysis were successfully performed within memory but also suffered some latency as data was transferred back into RAM. Thus the parallel efficiencies exhibited by the largest models is suboptimal.

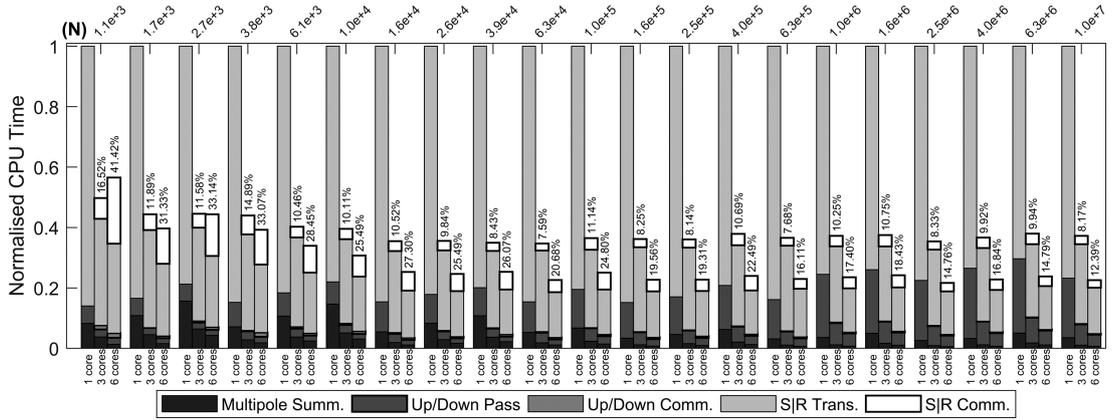


Fig. 11. Components of the FMM matrix-vector product calculation time versus N for the serial/parallel models. The bottom and top x -axes indicate the number of CPU cores and the approximate number of dof for each group of bars, respectively.

The calculation time for the FMM matrix-vector product versus N using the full accuracy multipole expansions is shown in Fig. 11. The main components of the FMM algorithm are (1) the operations for the initial multipole summation (element grouping) and final far-field evaluation at each element (denoted as ‘Multipole Summ.’), (2) the upward and downward pass operations (‘Up/Down Pass’) for translations/filtering between octree levels, and (3) the $S|R$ translations across octree levels (‘ $S|R$ Trans.’) for interacting well-separated element groups. The parallel models also show the worker communication times in the upward/downward pass (‘Up/Down Comm.’) and $S|R$ translation (‘ $S|R$ Comm.’) stages of the algorithm. The computational times for the parallel solutions has been normalized relative to the 1-core solution for each problem size. The percentage of the total calculation time spent on inter-worker communication is indicated in Fig. 10 above the 3- and 6-core results. The corresponding PEs for the components of the FMM matrix-vector product versus N (including worker communication times, where appropriate) are presented in Table 2.

Figure 11 shows that the $S|R$ translations are the dominant computational cost of the FMM matrix-vector product for all problem sizes considered, as expected. The combined cost of the initial and final multipole summations constitutes only a relatively small component of the total computational cost of the matrix-vector product, and this trend is inversely proportional to N . Conversely, the calculation time required to apply the $S|S$ and $R|R$ translations in the upward and downward passes increases with N . The number of octree levels and associated translations between octree levels increases with N , and these have a higher computational than the corresponding $S|R$ translations. Thus, for the large models with $k \propto N$ the $S|S$ and $R|R$ translations constitute a more significant component of the total FMM matrix-vector calculation time. Figure 11 also shows that the inter-worker communication is dominated by the $S|R$ translations, while that for the $S|S$ and $R|R$ translations is negligible for $N > 10^4$ or so. The average total communication time of the

Table 2. Parallel efficiencies for the components of the FMM matrix-vector product versus N for the 3-core and 6-core solutions. All PE values are presented as percentages.

N	Multipole Summ.		Up/Down Pass		$S R$ Trans.	
	3-Core	6-Core	3-Core	6-Core	3-Core	6-Core
1.1×10^3	73.2	101.7	50.5	26.3	67.9	27.8
1.7×10^3	80.0	115.5	82.3	37.8	74.1	39.0
2.7×10^3	81.1	61.3	71.0	34.0	74.0	35.1
3.8×10^3	84.2	64.9	87.0	39.6	74.3	41.5
6.1×10^3	94.7	73.7	75.8	49.4	82.1	47.0
1.0×10^4	95.8	78.6	76.2	48.1	83.2	51.9
1.6×10^4	95.2	80.4	99.6	70.0	93.2	64.7
2.6×10^4	96.2	81.4	104.2	73.5	92.4	65.9
3.9×10^4	97.0	81.2	100.4	64.4	94.7	64.3
6.3×10^4	96.4	79.4	90.9	67.8	96.8	74.2
1.0×10^5	96.3	78.5	94.5	67.1	90.5	65.6
1.6×10^5	96.5	79.3	89.9	67.2	92.3	74.2
2.5×10^5	97.5	80.3	91.5	66.8	92.4	74.0
4.0×10^5	99.6	84.9	90.9	69.4	86.9	68.6
6.3×10^5	96.2	78.3	92.1	69.8	90.8	72.8
1.0×10^6	101.4	83.0	92.8	74.9	88.1	69.4
1.6×10^6	98.8	81.0	95.1	75.6	86.7	66.6
2.5×10^6	95.8	81.4	99.4	81.1	92.9	75.9
4.0×10^6	94.2	77.6	98.8	82.2	88.6	70.5
6.3×10^6	94.9	76.4	95.3	79.7	84.9	66.9
1.0×10^7	112.9	85.7	91.7	78.5	88.1	72.3

FMM matrix-vector product is 8.4% and 18.1% for the 3-core and 6-core solutions respectively, for models with $> 10^4$ dof per CPU core. Analogous results for the preconditioner accuracy FMM matrix-vector products are 13.9% and 24.2%, respectively.

The parallel efficiencies for the main components of the FMM matrix-vector product are shown in Table 2. The initial and final multipole summations do not require any worker communication and so should present close to ideal (100%) scaling. This is indeed the case for all but the smallest 3-core solutions, whereas the 6-core results exhibit a PE of around 80% for models with $N \geq 10^4$ dof. The initialization cost for the parallel execution environment has been minimized in these tests, while any effect from Intel CPU's turbo boost feature would equally affect the PE of the 3-core results. Thus, it is not obvious what additional effect causes the poorer PE in the 6-core results. One possible explanation is that the operating system must share some utilization of the CPU cores when all cores are used in the parallel FMBEM, and this causes additional latency in the parallel calculations. Some multipole summation results in Table 2 also display apparent super-scaling ($> 100\%$ efficiency). These are likely due to small errors in the average calculation times for the smaller models, while the 112.9% PE for the 10^7 dof 3-core model is due to the fact that this model exceeded the available RAM in the pre-calculation stage. The multipole expansions are automatically transferred to hard disk storage when the memory is exceeded, and so

transferring this data back into RAM during the first iteration of the multipole summation timing calculations impeded the serial calculation speed. Conversely, the parallel models transfer the multipole expansion data back into RAM when distributing the data to the workers during the FMM precomputation stage, and so no similar delay is present in the parallel model results. A similar effect is also apparent in the 6-core 10^7 dof model.

The combined upward and downward pass of the FMM matrix-vector product exhibits good PE for models with $N > 10^4$ dof (see Table 2), which can be seen in Fig. 11 to correspond to those models which have small ratios of worker communication to calculation time in the upward/downward passes. The 6-core results again exhibit poorer PE than the corresponding 3-core models. Finally, the larger component of inter-worker communication for the $S|R$ translations yields a reduced PE for this component: averaging 90.6% and 69.7% for the 3-core and 6-core models with $N > 10^4$ dof. Excluding the $S|R$ communication time yields corresponding PEs of 101.4% and 89.1% respectively, which again suggests that the parallel calculations are applied with suboptimal scaling for the 6-core model. This may again be due to latency related to utilizing all available CPU core, or from a more significant nonuniformity in the number of $S|R$ translations applied by each worker (or both).

The PE of the present model is comparatively poor compared to others in the published literature, mainly due to the inefficient precalculation steps, while that for the fGMRES solution (the most computationally expensive component for large-scale models) is relatively good. This is due to the minimized inter-worker communication resulting from the columnwise distribution, element reordering and row compression implemented in the present algorithm. As a result, all stages of the FMBEM matrix-vector products exhibit good PE for the model configurations considered. Future work will focus on improving the PE of the precalculation steps of the model, while testing with larger multicore systems will resolve any systematic latencies that are suspected in the present 6-core results.

5.2. TS modeling of the BeTSSi II submarine hull model

In this section, the parallel broadband Helmholtz FMBEM is applied to TS modeling of the BeTSSi II generic submarine hull model, which was developed in 2014⁶³ to provide a generic but realistic submarine hull shape for benchmarking various TS modeling approaches. A summary of the results from the workshop can be found in Ref. 64. TS predictions for the model have been presented from a number of modeling methods, including BEM, FMBEM, H-matrix, SCSD and ray-tracing methods.^{65–70} The FMBEM is suitable for modeling the submarine hull at low-to-mid frequencies (up to a few kHz), while at higher frequencies the fast approximate methods (such as ray-tracing) are better suited, as they do not require an explicit frequency-dependent discretization of the scattering surface. Monostatic TS modeling results are presented here for the outer submarine hull at frequencies of 1 kHz and 3 kHz. It should be noted that the performance of the FMBEM can be improved for predominantly long and thin ‘1D-shaped’ objects by optimally placing the object within the octree structure⁷¹: however these optimizations have not been pursued in the present analysis. The overall dimensions of BeTSSi II are 62 m length by 7 m diameter (extending to 7.5 m for the flat deck) for the main hull (see Ref. 63 for full details).

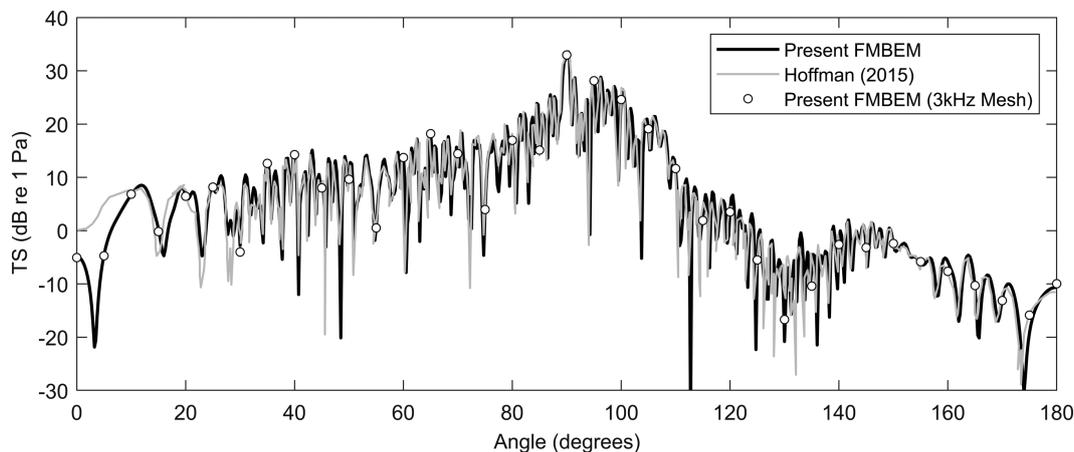


Fig. 12. Target strength results in dB re 1 Pa for the BeTSSi II submarine hull model at 1 kHz. Results from the present model are compared to digitized results from Ref. 66, as well as similar FMBEM results calculated on a higher resolution mesh.

5.2.1. Monostatic TS modeling at 1 kHz

The monostatic TS of the outer BeTSSi II submarine hull as a function of incident angle in the horizontal plane was calculated at 1 kHz using the FMBEM model, assuming a rigid (Neumann) BC. The mesh consisted of 159 780 plane triangular boundary elements with constant unknowns (corresponding to a mesh resolution of 10 elements per wavelength at 1 kHz) and the FMBEM model was solved using 6 CPU cores. The TS of the model was solved over the 0° to 180° azimuth angular range (from bow to stern) with a 0.25° resolution. The average calculation time (including initialization of the MATLAB parallel environment) was approximately 725 s per model. Figure 12 presents the FMBEM TS results versus backscatter angle, in comparison to analogous digitized numerical results published by Hoffmann.⁶⁶ Additionally, TS results in 5° increments over the angular range have been calculated on a higher resolution mesh consisting of 1 538 580 plane triangular boundary elements (10 elements per wavelength at 3 kHz).

It can be seen from Fig. 12 that the present model results and those from Hoffmann⁶⁶ are generally in good agreement, except near the bow (0°) where the present model predicts lower TS levels and a null at approximately 3° . This may be due to variations in the discretized curvature of the outer pressure hull (nose and top casing section which melds into the flat deck). The 1 kHz TS results calculated on the higher resolution mesh also show good agreement with those calculated on the 159 780 element mesh; indicating that the TS results for the BeTSSi II model are sufficiently converged when using 10 elements per wavelength.

5.2.2. Monostatic TS modeling at 3 kHz

Analogous monostatic TS results were calculated for the outer BeTSSi II submarine hull at a frequency of 3 kHz using the higher resolution mesh (1 538 580 dof) introduced in the

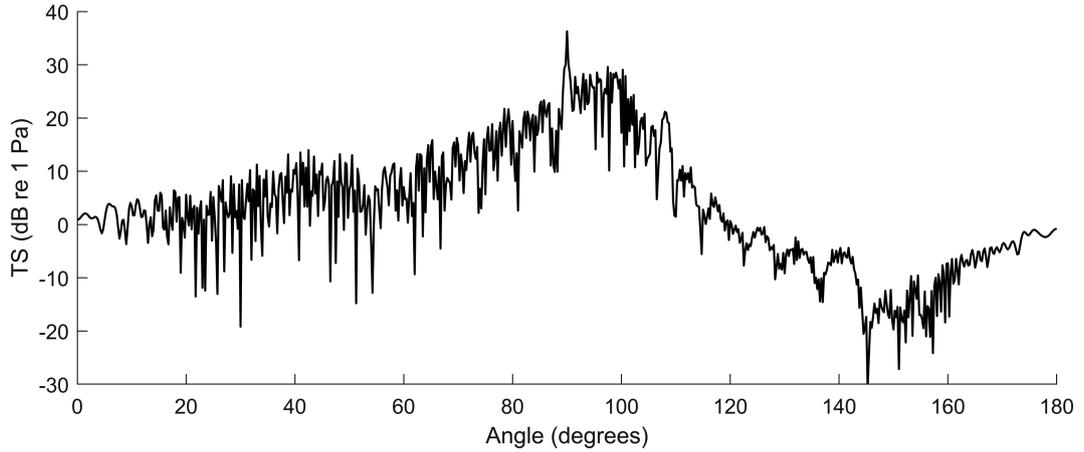


Fig. 13. Target strength results in dB re 1 Pa for the BeTSSi II submarine hull model at 3 kHz.

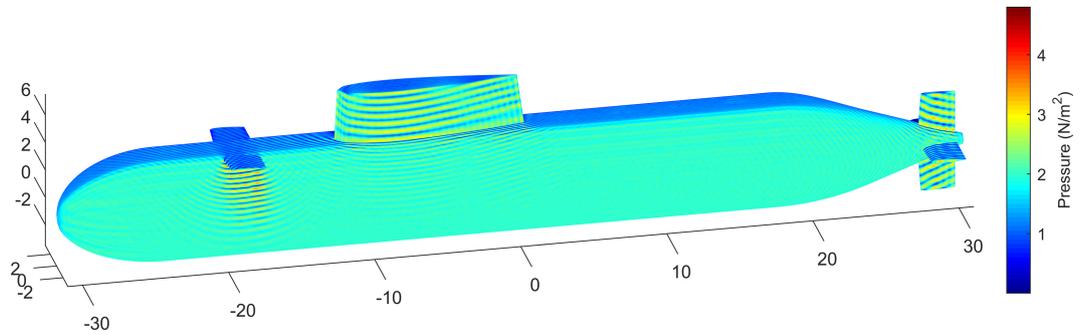


Fig. 14. The absolute value of the total pressure field for the BeTSSi II submarine hull model for a 3 kHz plane-wave impinging at broadside incidence.

previous section. The TS of the model was solved over the 0° to 180° azimuth angular range with a 0.25° resolution. The average calculation time (including initialization of the MATLAB parallel environment) was approximately 90 min per model run using 6 CPU cores. Comparatively, the calculation time for the single-threaded model was approximately 300 min per model run, corresponding to a PE of approximately 56%. Figure 13 presents the FMBEM TS results versus backscatter angle at 3 kHz, while Fig. 14 presents the absolute value of the surface pressure field at 3 kHz and broadside incidence.

6. Conclusions

This work presents a parallel broadband Helmholtz FMBEM for large-scale acoustic scattering problems, which employs a simple novel parallelization strategy based on the column-wise distribution of the equivalent BEM matrix, as well as element reordering and rowwise compression of data. This approach allows for the parallel calculation of the

FMBEM matrix-vector products using a small number of CPU cores (≤ 64) with a minimized communication overhead. The sparse BEM near-field and sparse approximate inverse preconditioner are also constructed and executed in parallel, while the fGMRES solver has been modified to execute within the parallel environment; thus minimizing the initialization cost and data transfer for the parallel calculations. Numerical tests were conducted on a workstation computer (Intel i7 hexacore, 128GB RAM) which demonstrated the broadband FMBEM models expected $O(N \log N)$ computational cost and $O(N)$ memory complexity. Acoustic scattering problems with up to 10^7 dof and $kD = 1299$ were considered in the analysis. The parallel fGMRES iterative solver exhibited good parallel efficiencies for large-scale models with more than 10 000 dof per CPU core; up to 92.9% and 75.9% in the best case for the 3-core and 6-core models, respectively. Furthermore, some test results indicated that the models utilizing all available CPU cores may suffer from an additional latency and a reduced PE due to the processors sharing resources with the background operations (operating system, etc.). The precalculation stages of the FMBEM model (in particular, the FMM precomputation stage) exhibited poor PE, and so the complete FMBEM model solution only achieved average parallel efficiencies of 68.3% and 47% for the 3-core and 6-core models when treating problems with $> 10^4$ dof per CPU core. Future work will focus on improving the parallel performance of the precalculation stages and to extend this analysis to multi-core Xeon processors found in typical high-end workstations.

References

1. Z. Y. Yan and X. W. Gao, The development of the pFFT accelerated BEM for 3-D acoustic scattering problems based on the Burton and Miller's integral formulation, *Eng. Anal. Bound. Elem.* **37** (2013) 409–418.
2. A. Brancati, M. H. Aliabadi and I. Benedetti, Hierarchical adaptive cross approximation GMRES technique for solution of acoustic problems using the boundary element method, *CMES: Computer Modeling and Engineering Sciences* **43** (2009) 149–172.
3. X. Liu, H. Wu and W. Jiang, Hybrid approximation hierarchical boundary element methods for acoustic problems, *J. Comput. Acoust.* **25** (2017) 1750013.
4. V. Rokhlin, Rapid solution of integral equations of classical potential theory, *J. Comput. Phys.* **60**(1) (1985) 187–207.
5. L. Greengard and V. Rokhlin, A fast algorithm for particle simulations, *J. Comput. Phys.* **73** (1987) 325–348.
6. R. Coifman, V. Rokhlin and S. Wandzura, The fast multipole method for the wave equation: A pedestrian prescription, *IEEE Antennas Propag. Mag.* **35** (1993) 7–12.
7. V. Rokhlin, Diagonal forms of translation operators for the Helmholtz equation in three dimensions, *Appl. Comput. Harmonic. Anal.* **1** (1993) 82–93.
8. N. A. Gumerov and R. Duraiswami, Recursions for the computation of multipole translation and rotation coefficients for the 3-D Helmholtz equation, *SIAM J. Sci. Comput.* **25** (2003) 1344–1381.
9. L. Greengard, J. Huang, V. Rokhlin and S. Wandzura, Accelerating fast multipole methods for the Helmholtz equation at low frequencies, *IEEE Comput. Sci. Eng.* **25** (1998) 32–38.
10. Y. Yasuda, T. Oshima, T. Sakuma, A. Gunawan and T. Mauumoto, Fast multipole boundary element method for low-frequency acoustic problems based on a variety of formulations, *J. Comput. Acoust.* **18** (2010) 363–395.

11. S. Schneider, Application of fast methods for acoustic scattering and radiation problems, *J. Comput. Acoust.* **11** (2003) 387–401.
12. Z. Chen, H. Waubke and W. Kreuzer, A formulation of the fast multipole boundary element method (FMBEM) for acoustic radiation and scattering from three-dimensional structures, *J. Comput. Acoust.* **16** (2008) 303–320.
13. N. Nishimura, Fast multipole accelerated boundary integral equation methods, *Appl. Mech. Rev.* **63** (2002) 299–324.
14. Y. J. Liu, S. Mukherjee, N. Nishimura, M. Schanz, W. Ye, A. Sustradhar, E. Pan, N. A. Dumont, A. Frangi and A. Saez, Recent advances and emerging applications of the boundary element method, *Appl. Mech. Rev.* **64** (2011) 030802.
15. Z. Gimbutas and L. Greengard, A fast and stable method for rotating spherical harmonic expansions, *J. Comput. Phys.* **228** (2009) 5621–5627.
16. N. A. Gumerov and R. Duraiswami, Recursive computation of spherical harmonic rotation coefficients of large degree, in *Excursions in Harmonic Analysis*, R. Belu, M. Begú, J. J. Benedetto, W. Czaja and M. Okoudjou (eds.) (Birkhäuser, Switzerland, 2015), pp. 105–141.
17. L. J. Jiang and W. C. Chew, A mixed-form fast multipole algorithm, *Microw. Opt. Technol. Lett.* **53** (2005) 4145–4156.
18. H. Cheng, W. Y. Crutchfield, Z. Gimbutas, L. F. Greengard, J. F. Ethridge, J. Huang, V. Rokhlin, N. Yarvin and J. Zhao, A wideband fast multipole method for the Helmholtz equation in three dimensions, *J. Comput. Phys.* **216** (2006) 300–325.
19. H. Wallén, S. Järvenpää and P. Ylä-Oijala, Broadband multilevel fast multipole algorithm for acoustic scattering problems, *J. Comput. Acoust.* **14** (2006) 507–526.
20. N. A. Gumerov and R. Duraiswami, A broadband fast multipole accelerated boundary element method for the 3D Helmholtz equation, *J. Acoust. Soc. Am.* **125** (2009) 191–205.
21. W. R. Wolfe and S. K. Lele, Wideband fast multipole boundary element method: Application to acoustic scattering from aerodynamic bodies, *Int. J. Numer. Meth. Fluids* **67** (2011) 2108–2129.
22. C. Zheng, T. Matsumoto, T. Takahashi and H. Chen, A wideband fast multipole boundary element method for three dimensional acoustic shape sensitivity analysis based on direct differentiation method, *Eng. Anal. Bound. Elem.* **36** (2012) 361–371.
23. E. Darve and P. Havé, A fast multipole method for Maxwell equations stable at all frequencies, *Phil. Trans. Roy. Soc. Lond. A* **362** (2004) 603–628.
24. S. Chaillat and F. Collino, A wideband fast multipole method for the Helmholtz kernel: Theoretical developments, *Math. Model. Numer. Anal.* **70** (2015) 660–678.
25. L. Greengard and W. D. Gropp, A parallel version of the fast multipole method, *Comput. Math. Appl.* **20** (1990) 63–71.
26. N. A. Gumerov and R. Duraiswami, Fast multipole methods on graphics processors, *J. Comp. Phys.* **227** (2008) 8290–8313.
27. Ö. Ergül and L. Gürel, A hierarchical partitioning strategy for an efficient parallelisation of the multilevel fast multipole algorithm, *IEEE Trans. Antennas Propag.* **57** (2009) 1740–1750.
28. R. Yokota and L. A. Barba, A tuned scalable fast multipole method as a preeminent algorithm for exascale systems, *Int. J. High Perform. Comput. Appl.* **26** (2012) 337–346.
29. Q. Hu, N. A. Gumerov and R. Duraiswami, Scalable fast multipole methods on distributed heterogeneous architectures, *Proc. 2011 Int. Conf. High Performance Computing, Networking, Storage and Analysis*, Seattle, WA, USA, 2011, pp. 1–12.
30. I. Lashuk, A. Chandramowlishwaran, H. Langston, T. Nguyen, R. Sampath, A. Shringarpure, R. Vuduc, L. Ying, D. Zorin and G. Biros, A massively parallel adaptive fast-multipole method on heterogeneous architectures, *Commun. ACM* **55** (2012) 101–109.

31. M. Abduljabbar, M. A. Farhan, N. Al-Harthi, R. Chen, R. Yokota, H. Bagci and D. Keyes, Extreme scale FMM-accelerated boundary integral equation solver for wave scattering, *SIAM J. Sci. Comput.* **41** (2019) C245–C268.
32. R. Yokota and L. A. Barba, ExaFMM, <http://www.bu.edu/exafmm/> (last viewed January 21, 2020).
33. L. Giraud, J. Langou and G. Sylvand, On the parallel solution of large industrial wave propagation problems, *J. Comput. Acoust.* **14** (2006) 1–29.
34. J. A. López-Fernández, M. L. Portugués, J. M. Taboada, H. J. Rice and F. Obelleiro, HP-FASS: A hybrid parallel fast acoustic scattering solver, *Int. J. Comput. Math.* **88** (2011) 1960–1968.
35. A. Chandramowlishwaran, K. Madduri and R. Vuduc, Diagnosis, tuning and redesign for multi-core performance: A case study of the fast multipole method, *Proc. 2010 ACM/IEEE Int. Conf. High Performance Computing, Networking, Storage and Analysis*, New Orleans, LA, USA, 2010, pp. 1–12.
36. V. Cutanda Henríquez and P. Risby Andersen, A three-dimensional acoustic boundary element method formulation with viscous and thermal losses based on shape function derivatives, *J. Theor. Comput. Acoust.* **26** (2018) 1850039.
37. D. Liu, S. Marburg, C. Geweth and N. Kessissoglou, Non-negative intensity for structures with inhomogeneous damping, *J. Theor. Comput. Acoust.* **27** (2019) 1850050.
38. L. L. Chen, H. Lian, Z. Liu, H. B. Chen, E. Atroshchenko and S. P. A. Bordas, Structural shape optimization of three dimensional acoustic problems with isogeometric boundary element methods, *Comput. Methods Appl. Mech. Eng.* **355** (2019) 926–951.
39. N. A. Gumerov and R. Duraiswami, *Fast Multipole Methods for the Helmholtz Equation in Three Dimensions* (Elsevier, Oxford, 2004).
40. S. Marburg and S. Amini, Cat’s eye radiation with boundary elements: Comparative study on treatment of irregular frequencies, *J. Comput. Acoust.* **13** (2005) 21–45.
41. S. Kirkup, The boundary element method in acoustics, <http://www.boundary-element-method.com/acoustics/index.htm> (last viewed February 14, 2017).
42. A. J. Burton and G. F. Miller, The application of integral equation methods to the numerical solution of some exterior boundary-value problems, *Proc. Roy. Soc. Lond. A.* **323** (1971) 201–210.
43. S. Marburg, The Burton and Miller method: Unlocking another mystery of its coupling parameter, *J. Comput. Acoust.* **24** (2016) 1550016.
44. S. Marburg and B. Nolte, A unified approach to finite and boundary element discretization in linear time-harmonic acoustics, in *Computational Acoustics of Noise Propagation in Fluids*, S. Marburg and B. Nolte (eds.) (Springer, Berlin, 2008), pp. 1–34.
45. H. T. Rathod, K. V. Nagaraja, B. Venkatesuda and N. L. Ramesh, Gauss Legendre quadrature over a triangle, *J. Indian Inst. Sci.* **84** (2004) 183–188.
46. L. Scuderi, On the computation of nearly singular integrals in 3D BEM collocation, *Int. J. Numer. Meth. Eng.* **74** (2008) 1733–1770.
47. T. Matsumoto, C. Zheng, S. Harada and T. Takahashi, Explicit evaluation of hypersingular boundary integral equation for 3-D Helmholtz equation discretized with constant triangular element, *J. Comput. Sci. Technol.* **4** (2010) 194–206.
48. J. Song, C. Lu and W. C. Chew, Multilevel fast multipole algorithm for electromagnetic scattering by large complex objects, *IEEE Trans. Antennas Propag.* **45** (1997) 1488–1493.
49. J. Rahola, Diagonal forms of the translation operators in the fast multipole algorithm for scattering problems, *BIT* **36** (1996) 333–358.
50. E. Darve, The fast multipole method I: Error analysis and asymptotic complexity, *SIAM J. Numer. Anal.* **38** (2000) 98–128.

51. M. Böhme and D. Potts, A fast algorithm for filtering and wavelet decomposition on the sphere, *Electron. Trans. Numer. Anal.* **16** (2003) 70–93.
52. D. R. Wilkes and A. J. Duncan, Fast spherical filtering in the broadband FMBEM using a non-equally spaced FFT, *Proc. Acoustics 2017*, Perth, WA, Australia, 2017, pp. 1–10.
53. G. M. Morton, A computer oriented geodetic data base; and a new technique in file sequencing, Technical Report, IBM Ltd., Ottawa (1966).
54. D. R. Wilkes and A. J. Duncan, A low frequency elastodynamic fast multipole boundary element method in three dimensions, *Comput. Mech.* **56** (2015) 829–848.
55. Y. Liu, *Fast Multipole Boundary Element Method: Theory and Applications in Engineering* (Cambridge University Press, New York, 2009).
56. S. Marburg and S. Schneider, Performance of iterative solvers for acoustic problems. Part I. Solvers and effect of diagonal preconditioning, *Eng. Anal. Bound. Elem.* **27** (2003) 727–750.
57. Y. Yasuda, S. Sakamoto, Y. Kosaka, T. Sakuma, N. Okamoto and T. Oshima, Numerical analysis of large-scale sound fields using iterative methods part I: Application of Krylov subspace methods to boundary element analysis, *J. Comput. Acoust.* **15** (2007) 449–471.
58. Y. Saad, A flexible inner-outer preconditioned GMRES algorithm, *SIAM J. Sci. Comput.* **14** (1993) 461–469.
59. K. Chen and P. J. Harris, Efficient preconditioners for iterative solution of the boundary element equations for the three-dimensional Helmholtz equation, *Appl. Numer. Math.* **36** (2001) 475–489.
60. B. DeVries, J. Iannelli, C. Trefftz, K. A. O’Hearn and G. Wolffe, Parallel implementations of FGMRES for solving large, sparse non-symmetric linear systems, *Procedia Comput. Sci.* **18** (2013) 491–500.
61. M. Ochmann, A. Homm, S. Makarov and S. Semenov, An iterative GMRES-based boundary element solver for acoustic scattering, *Eng. Anal. Bound. Elem.* **27** (2003) 717–725.
62. S. Marburg, A pollution effect in the boundary element method for acoustic problems, *J. Theor. Comput. Acoust.* **26** (2018) 1850018.
63. L. Gilroy, C. DeJong, B. Nolte and I. Schäfer, BeTSSi II Benchmark Target Strength Simulation, Benchmark Target Strength Simulation II Workshop/Conference, <https://www.dwt-sgw.de/wt-aks/arbeitskreis-akustik/veranstaltungen-2014/> (last viewed February 27, 2017).
64. M. Östberg, BeTSSi II — A benchmarking workshop on target echo strength, Technical Report, FOI, Stockholm (2015).
65. B. Nolte, J. Ehrlich, H.-G. Hofmann, I. Schäfer, A. Schäfke, A. Stoltenberg and R. Burgschweiger, Numerical methods in underwater acoustics — Sound propagation and backscattering, *Hydroacoustics* **18** (2015) 127–140.
66. C. Hoffmann, Simulation based design approach in sonar sensor-platform interaction, *Proc. 22nd Int. Congress on Sound and Vibration*, Florence, Italy, 2015, pp. 1–8.
67. T. Avsic, Taking the next step in stealth submarine development, *Proc. 22nd Int. Congress on Sound and Vibration*, Florence, Italy, 2015, pp. 1–7.
68. L. Gilroy, D. Pearson and D. Brennan, Contributing geometric features of submarine target echo strength, *Proc. 22nd Int. Congress on Sound and Vibration*, Florence, Italy, 2015, pp. 1–8.
69. M. Östberg, Target echo strength modelling at FOI, including results from the BeTSSi II workshop, *Proc. 39th Scandinavian Symp. Physical Acoustics*, Geilo, Norway, 2016, pp. 1–9.
70. F. Alouges, M. Aussal and E. Parolin, Fast boundary element method for acoustics with the sparse cardinal sine decomposition, *Revue Européenne de Mécanique Numérique* **26** (2017) 1–17.
71. Y. Yasuda and T. Sakuma, An effective setting of hierarchical cell structure for the fast multipole boundary element method, *J. Comput. Acoust.* **13** (2005) 47–70.