Published in final edited form as:

IEEE Trans Ultrason Ferroelectr Freq Control. 2016 November ; 63(11): 1967–1979. doi:10.1109/

TUFFC.2016.2591920

Efficient broadband simulation of fluid-structure coupling for membrane-type acoustic transducer arrays using the multi-level fast multipole algorithm

Bernard Shieh, Karim Sabra, and F. Levent Degertekin

Abstract

A boundary element model provides great flexibility for the simulation of membrane-type micromachined ultrasonic transducers (MUTs) in terms of membrane shape, actuating mechanism, and array layout. Acoustic crosstalk is accounted for through a mutual impedance matrix which captures the primary crosstalk mechanism of dispersive-guided modes generated at the fluid-solid interface. However, finding the solution to the fully-populated boundary element matrix equation using standard techniques requires computation time and memory usage which scales by the cube and by the square of the number of nodes, respectively, limiting simulation to a small number of membranes. We implement a solver with improved speed and efficiency through the application of a multi-level fast multipole algorithm (FMA). By approximating the fields of collections of nodes using multipole expansions of the free-space Green's function, an FMA solver can enable the simulation of hundreds of thousands of nodes while incurring an approximation error that is controllable. Convergence is drastically improved using a problem-specific block-diagonal preconditioner. We demonstrate the solver's capabilities by simulating a 32-element 7 MHz 1-D CMUT phased array with 2880 membranes. The array is simulated using 233,280 nodes for a very wide frequency band up to 50 MHz. For a simulation with 15,210 nodes, the FMA solver performed 10-times faster and used 32-times less memory than a standard solver based on LU decomposition. We investigate the effects of mesh density and phasing on the predicted array response and find that it is necessary to use about 7 nodes over the width of the membrane to observe convergence of the solution—even below the first membrane resonance frequency—due to the influence of higher-order membrane modes.

I. Introduction

By coupling the extensional motion of an electromechanical driver to the flexural motion of a thin membrane shell, flexensional acoustic transducers are characterized by their power, efficiency, and desirable broadband response in immersion. Modern incarnations of the flexensional design (see, for example, the cymbal transducer [1]) have been demonstrated for a variety of applications, including underwater object identification [2], transdermal delivery of drugs and insulin [3], [4], and treatment of ulcerations by ultrasonic therapy [5]. With the advent of precision machining and, more recently, micromachining, large transducer arrays composed of hundreds of flexural membranes can now be manufactured. These membrane-type transducer arrays typically use piezoelectric or capacitive layers coupled directly to thin boundary-clamped membrane structures which are excited into flexural motion by an alternating current. Fabrication of arrays for low-frequency operation,

with membrane lateral dimensions and thicknesses on the order of millimeters, has been demonstrated with adequate precision using standard, widely-available, low-cost technologies [6], [7].

In recent years, intense research effort has been focused on the realization of microscale transducer arrays using microfabrication techniques adopted from the semi-conductor industry. Micromachined Ultrasonic Transducers (MUTs), such as those actuated by capacitive layers (CMUTs) or by piezoelectric layers (PMUTs), are an emerging technology with relevance to a diversity of applications. For example, MUT arrays have been designed for the purposes of rangefinding [8], sound projection [9], and fingerprint identification [10]. In the field of medical sonography, MUT arrays are particularly attractive due to their greater layout flexibility and their low mechanical impedance, the latter of which obviates the need for complicated impedance matching layers. In addition, successful integration with CMOS electronics has been demonstrated using methods such as flip chip bonding [11] and monolithic integration [12]–[15]. Prototype devices for intracardiac echocardiography [16], intravascular ultrasound [17], [18], and photoacoustic imaging [19] provide a glimpse into future imaging platforms based on MUT technology. MUTs are also a unique candidate for super-resolution imaging of biological samples using acoustic time-reversal [20].

It is well known that membrane-type arrays are susceptible to acoustic cross-talk which can negatively affect their performance. In [21], through finite element analysis (FEA) of a 1-D CMUT array, it was determined that the primary cross-talk mechanism was dispersive guided modes traveling along the fluid-solid interface, with smaller contributions from substrate-born waves. A similar conclusion was reached with 3-D modeling of CMUTs based on a periodic finite element analysis/boundary element method (FEA/BEM) [22]. The dispersive guided modes of CMUT arrays have been studied in detail with 1-D and 2-D modal analysis of a CMUT array [23].

To date, the simulation of large membrane-type arrays with several thousand membranes has not been achieved without significant assumptions. Modeling full arrays using FEA requires a considerable computational effort, and generally rely on one or more symmetry conditions in order to reduce the problem size (see for example [21]). FEA with periodic boundary conditions can be used to simulate membrane-type arrays [22], [24], [25] with the assumptions of an unbounded array with periodic layout and constrained phasing. More efficient simulations have been proposed, including analysis using a semi-analytical model based on the first vibration mode of circular membranes [26]–[29] and a model based on the eigenmodes of circular membranes in free-vibration [30]. It is important to note that although these studies include membrane-to-membrane impedances, for the sake of efficiency, the individual membrane modeshapes are determined considering only the radially-symmetric modes and for a single membrane in isolation (without neighbors).

A BEM model based on finite difference approximations of thin clamped plates [31], [32] provides both the efficiency and flexibility desired for the simulation of realistic membrane-type arrays. Recently, this BEM approach has been verified with FEA software (COMSOL) and experimentally for a dual-ring CMUT array with good agreement [33]. Unfortunately, the BEM model suffers from unsatisfactory memory and runtime scaling due to its

dependence on a fully-populated mutual impedance matrix, limiting its utility for the simulation of large arrays. In this paper, we address the scaling problem of the BEM model directly through the application of a fast multipole algorithm (FMA). The FMA was first introduced in 1987 by Rokhlin and Greengard [34] for the rapid evaluation of electric and gravitational fields involving a large number of particles. Since then, it has been applied in various forms to problems in electromagnetics [35], molecular dynamics [36], and acoustics [37]–[39], amongst others.

An FMA-accelerated BEM model for membrane-type arrays can handle several thousand membranes with reasonable computational resources while retaining the flexibility of the BEM. Its distinguishing features include: simulation of arrays of finite size; simulation of arbitrary array layouts, i.e. no symmetry or periodicity required; support for arbitrary membrane shapes, e.g. circular and rectangular; simultaneous simulation of different membrane types, e.g. different gap sizes, membrane thicknesses, isolation thicknesses; and the ability to phase each membrane individually.

The structure of the paper is as follows. We review the BEM model for membrane-type arrays and the underlying BEM equation which must be solved. A brief overview of the fast multipole algorithm is provided and the operations of a multi-level FMA scheme are explained in detail. We address the practical matters of optimization, preconditioning, and controlling error. We implemented an FMA solver for the BEM model and simulated two cases of interest. A single element of a 1-D CMUT array with 90 membranes and 15,210 nodes was used to compare the performance of the FMA solver with the standard direct BEM method. A large 32-element 1-D CMUT array with 2880 membranes was simulated to demonstrate our FMA solver's ability to handle large arrays. We investigate the effects of mesh density and excitation conditions on the predicted cross talk and array modes. Finally, computation and memory usage are compared for all the simulated cases.

II. Theory

A. Boundary element model

In contrast with 3-D FEA, the 2-D boundary element model reduces the complexity of the problem by modeling the membranes using stiffness and mass matrices. This avoids the need to mesh the interior structures and the fluid, but limits the cross-talk mechanism to fluid-born waves, i.e. the dispersive guided modes, since the substrate is not simulated. The membranes can be modeled as thin plates, where finite difference approximations (fourth-order in this study) are used to discretize the partial differential equations with the assumption that the membranes are rigidly clamped at their boundaries. After Fourier decomposition with radial frequency ω , the full mechanical system is described by the following frequency-dependent matrix equation

$$(\mathbf{G}_{mech} + \mathbf{G}_{rad})\mathbf{u} = \mathbf{G}\mathbf{u} = \mathbf{f} \quad (1)$$

$$\mathbf{G}_{mech} = -\omega^2 \mathbf{M} + i\omega \mathbf{C} + \mathbf{K} \quad (2)$$

$$\mathbf{G}_{rad} = i\omega \mathbf{Z}_{rad}(\omega) \quad (3)$$

where $\mathbf{M}$ is the diagonal nodal mass matrix, $\mathbf{C}$ is the diagonal nodal damping matrix, $\mathbf{K}$ is a block-diagonal stiffness matrix relating the nodes within each membrane, $\mathbf{Z}_{rad}$ is the mutual acoustic impedance matrix, $\mathbf{f}$ is the external forcing on each node, and $\mathbf{u}$ is the displacement of the nodes. Detailed derivations for each matrix can be found in [31], [32], [40]. As an alternative to thin plates, the equivalent stiffness matrix for more complicated structures can be determined using FEA of a single membrane [40] [41].

Radiation from the membranes are handled by node-to-node interactions via the Green's function for a baffled source. The effect of radiation on the node itself is approximated by the radiation impedance of a circular piston with an equivalent area [42]

$$\mathbf{Z}_{rad}^{mn} = \begin{cases} i\omega\rho a_n \frac{e^{-ik|\mathbf{r}_m - \mathbf{r}_n|}}{2\pi|\mathbf{r}_m - \mathbf{r}_n|} & m \neq n \\ \rho c \left[\frac{1}{2\pi} k^2 a_n + i \frac{8}{3\pi} k \left(\frac{a_n}{\pi} \right)^{1/2} \right] & m = n \end{cases}$$

where $\mathbf{r}_m$ is the receiver node, $\mathbf{r}_n$ is the source node with surface area a_n , ρ is the fluid density, and c is the fluid sound speed.

The actuating mechanism of the membrane is included generally as an external forcing on each node by the vector $\mathbf{f}$. The source of the forcing will depend on the particular device design, e.g. stress gradients in thin piezoelectric films in the case of PMUTs, and electrostatic force between capacitive layers for CMUT devices. For CMUTs specifically, a useful approximation is to consider the small harmonic actuation of the membrane about a static deflection. The forcing in this case is related to the applied AC voltages $\mathbf{v}$ by the transformer ratios $\boldsymbol{\eta}$ and to the displacement $\mathbf{u}$ by a spring-softening matrix $\mathbf{K}_{ss}$ [31]

$$\mathbf{f} = \boldsymbol{\eta} \mathbf{v}^T + \mathbf{K}_{ss} \mathbf{u} \quad (4)$$

The numerical solution to (1) can be found simply by calculating $\mathbf{G}^{-1}$, or more efficiently, using direct solvers based on the QR or LU decomposition of $\mathbf{G}$. However, as the scale of the problem extends beyond 30,000 nodes, the global nature of boundary element models ($\mathbf{Z}_{rad}$ is a symmetric, fully-populated matrix) makes the direct approach unviable. For instance, at this scale, the storage of $\mathbf{G}_{rad}$ would require 4.5 million entries, around 7 GiB (as 128-bit complex float including symmetry) and more than 5 hours to solve (at 1 GFLOPS). The fast multipole algorithm is a solution to the poor scaling of BEM by providing a matrix-free method for calculating the matrix-vector product $\mathbf{G}\mathbf{u}$, which can be performed with

$O(N \log N)$ complexity in both storage and runtime [43] compared with the $O(\frac{2}{3}N^3)$ complexity of a standard LU solver [44]. Algebraically, the FMA approach can be considered as a type of hierarchical low-rank matrix approximation, although these approximations are not explicitly stored due to its matrix-free property [45]. When paired with fast iterative solvers, e.g. General Minimum Residual (GMRES) [46] or Biconjugate Gradient (BiCG) [44], it becomes possible to simulate hundreds of thousands of nodes with reasonable computational resources.

B. Fundamental pressure evaluation in the fast multipole algorithm

Consider the evaluation of pressure at the m -th node located at $\mathbf{r}_m$ due to the aggregate effect of all the nodes, described mathematically by the action of the matrix-vector product $\mathbf{G}_{rad}\mathbf{u}$

$$p_m = i\omega\rho \sum_{n \neq m} q_n \frac{e^{-ik|\mathbf{r}_m - \mathbf{r}_n|}}{2\pi|\mathbf{r}_m - \mathbf{r}_n|} + \rho c q_m \left[\frac{1}{2\pi}(ka_m)^2 + i\frac{8}{3\pi}ka_m \left(\frac{a_m}{\pi}\right)^{1/2} \right] \quad (5)$$

where $q_n = i\omega a_n u_n$ is the complex source strength of the n -th node with location $\mathbf{r}_n$ under a baffled condition. The principle idea behind the fast multipole algorithm is to approximate the fields from distant nodes with truncated multipole expansions. Nodes are collected into clusters based on proximity, and cluster-to-cluster interactions replace node-to-node interactions whenever the clusters are sufficiently separated. The Green's function in (5) is replaced with an expansion derived from a combination of the Gegenbauer addition theorem (10.1.45/46 in [47]) with a plane wave expansion (refer to [48] for a detailed derivation)

$$\frac{e^{-ik|\mathbf{x}+\mathbf{y}|}}{|\mathbf{x}+\mathbf{y}|} = \frac{ik}{4\pi} \sum_{l=0}^{\infty} (2l+1) i^l h_l^{(2)}(kx) \int_{S_1} e^{iky \cdot \hat{\mathbf{s}}} P_l(\hat{\mathbf{x}} \cdot \hat{\mathbf{s}}) dS \quad (6)$$

where $\mathbf{x}$ and $\mathbf{y}$ are arbitrary vectors with $x > y$, $h_l^{(2)}(z)$ is the spherical Hankel function of the second kind, and $P_l(z)$ are the Legendre polynomials. The unit vectors $\hat{\mathbf{s}}$ are angles of the unit sphere S_1 , defined in spherical coordinates by the azimuth angle θ and polar angle ϕ , or in Cartesian coordinates by $\langle \cos\theta \sin\phi, \sin\theta \sin\phi, \cos\phi \rangle$. The integration is defined over the surface of S_1 , where $dS = \sin\theta d\theta d\phi$.

To understand how this expansion can be used for the evaluation of pressure, let $\mathbf{x} = \mathbf{b} - \mathbf{a}$ and $\mathbf{y} = \mathbf{r}_m - \mathbf{b} + \mathbf{a} - \mathbf{r}_n$ for locations $\mathbf{a}$ and $\mathbf{b}$, maintaining the stipulation that $|\mathbf{b} - \mathbf{a}| > |\mathbf{r}_m - \mathbf{b}| + |\mathbf{a} - \mathbf{r}_n|$. Substitution into (6) yields

$$\frac{e^{-ik|\mathbf{r}_m - \mathbf{r}_n|}}{|\mathbf{r}_m - \mathbf{r}_n|} = \lim_{L \rightarrow \infty} \frac{ik}{4\pi} \int_{S_1} e^{ik(\mathbf{r}_m - \mathbf{b}) \cdot \hat{\mathbf{s}}} T_L(\hat{\mathbf{s}}, \mathbf{b} - \mathbf{a}) e^{ik(\mathbf{a} - \mathbf{r}_n) \cdot \hat{\mathbf{s}}} dS \quad (7)$$

where we have interchanged the summation and integration, and defined a truncated *translation operator* T_L

$$T_L(\hat{\mathbf{s}}, \mathbf{b}-\mathbf{a}) = \sum_{l=0}^L (2l+1) i^l h_l^{(2)}(k|\mathbf{b}-\mathbf{a}|) P_l\left(\hat{\mathbf{s}} \cdot \frac{\mathbf{b}-\mathbf{a}}{|\mathbf{b}-\mathbf{a}|}\right) \quad (8)$$

We refer to $\mathbf{a}$ and $\mathbf{b}$ as the source and evaluation cluster centers, respectively, for clusters of nodes located within some expansion radius about each location (see Fig. 1a). Note that T_L depends only on the vector separating the cluster centers and not on the spatial distribution or the monopole strengths of the nodes within the clusters. This mathematical separation plays a critical role in the resulting computational speed-up of the algorithm.

The fundamental cluster-to-cluster pressure evaluation occurs in three steps. First, the nodes in the source cluster are *aggregated* about $\mathbf{a}$ by calculation of their *far-field signature*

$$F_{\mathbf{a}}(\hat{\mathbf{s}}) = \sum_n q_n e^{ik(\mathbf{a}-\mathbf{r}_n) \cdot \hat{\mathbf{s}}} \quad (9)$$

where q_n and $\mathbf{r}_n$ are the source strength and position of the n -th node, respectively, and $F_{\mathbf{a}}(\hat{\mathbf{s}})$ is a function on the unit sphere. Second, the far-field signature is multiplied by the translation operator, converting it into a *near-field signature* and shifting the cluster center from $\mathbf{a}$ to $\mathbf{b}$

$$N_{\mathbf{b},L}(\hat{\mathbf{s}}) = T_L(\hat{\mathbf{s}}, \mathbf{b}-\mathbf{a}) F_{\mathbf{a}}(\hat{\mathbf{s}}) \quad (10)$$

$N_{\mathbf{b},L}(\hat{\mathbf{s}})$ is likewise a function on the unit sphere. Finally, the near-field signature is *disaggregated* to determine the pressure at the m -th node in the evaluation cluster by carrying out the integration

$$p_m = \lim_{L \rightarrow \infty} -\rho c \frac{k^2}{8\pi^2} \int_{S_1} e^{ik(\mathbf{r}_m-\mathbf{b}) \cdot \hat{\mathbf{s}}} N_{\mathbf{b},L}(\hat{\mathbf{s}}) dS \quad (11)$$

The numerical implementation of this procedure requires two approximations: truncation of the translation operator at the L -th term and evaluation of the integral using numerical quadrature over the unit sphere. A trapezoidal quadrature rule with a uniform sampling of the sphere is a straight-forward way to handle the numerical integration. In this case, the weights w_s are constant and depend on the total number of sampling points N_θ and N_ϕ in each direction.

$$p_m \approx -\rho c \frac{k^2}{8\pi^2} \sum_{\hat{s}} w_s e^{ik(\mathbf{r}_m - \mathbf{b}) \cdot \hat{s}} N_{\mathbf{b},L}(\hat{s}) \quad (12)$$

$$w_s = \frac{2\pi}{N_\theta} \frac{\pi}{N_\phi} \quad (13)$$

C. Multi-level adaptive scheme

To achieve an optimal algorithm, the cluster-to-cluster pressure evaluation described above is paired with a scheme to adaptively adjust the precision of the expansion. Larger clusters (in terms of expansion radius) reduce the total number of interactions necessary but require more terms of T_L and a finer quadrature sampling in order to adequately sample the field. By scaling the cluster size with the distance of the interaction, a balance is struck between the number of interactions and the computational cost per interaction.

A multi-level algorithm introduces a hierarchical tree structure to manage such a scheme in the form of a quadtree (specific to problems in two dimensions). A quadtree is composed of multiple levels (refer to Fig. 1b). The top level $\mathcal{L}_0$ (the trunk) contains a single bounding box which encloses the entire problem domain. Subsequent levels are formed by repeated bisection of the bounding box in both dimensions, up to a desired maximum level $\mathcal{L}_{max}$ (the leaves). We refer to the bisected box as the *parent* and the four resulting boxes as the *children*. It follows that the trunk will have no parent and the leaves will not have any children.

At the maximum level, multipole expansions are calculated for the node clusters within each box in the form of far-field signatures. Rather than calculate the signatures for the boxes in the remaining levels, the signatures of parent boxes are acquired from their children through an efficient process. The child signatures, which are not valid expansions for the parent box, must first be manipulated through the application of several operations. These are illustrated in Fig. 1c.

A *shift* operation is defined which relocates a signature's center from one location to another. Shifting the center of a far-field (or near-field) signature from $\mathbf{c}$ to $\mathbf{d}$ is a simple multiplication with a complex exponential

$$F_{\mathbf{d}}(\hat{s}) = e^{ik(\mathbf{d} - \mathbf{c}) \cdot \hat{s}} F_{\mathbf{c}}(\hat{s}) \quad (14)$$

For a given parent box, the signature of each child box is shifted from its geometric center to the center of the parent box and then summed together.

Next, the signature's expansion radius must be enlarged to account for the increased detail required of the parent box. Each signature, as functions on the unit sphere, is *interpolated*

onto the sampling points of a finer quadrature rule with order selected appropriately for the size of the parent box. Much has been written on the optimal method for performing this interpolation process (and the reciprocal *filtering* process), the choice of which will be tied intimately with the selected quadrature scheme. We opt for a Fourier-based method [49] [50] for its ease of implementation and dependence on widely-available Fast Fourier Transform (FFT) routines. The reader should be aware of the alternatives, which include Lagrange polynomials [51], spherical filtering [52], and many others. After interpolation, the resulting signature is a valid expansion of the field from the cluster containing all the nodes within the parent box. In a similar fashion, child boxes can inherit signatures from their parent boxes, a process which is used to efficiently transfer near-field signatures down the quadtree. When moving from a parent to a child, the signature is shifted and then *filtered* onto the sampling points of a coarser quadrature rule.

The adaptive nature of the calculation is realized discretely by classifying all box-to-box interactions on a given level into three categories. The closest interactions are those from within the box and from neighboring boxes, i.e. those sharing a border or a vertex. A given box may therefore have a maximum of eight neighbors. These interactions are always computed directly using (5). The intermediate category includes interactions from non-touching neighbors (*ntn*), defined as the children of the neighbors of the parent box, excluding the children which are also neighbors. A given box may have up to 27 non-touching neighbors. Interactions in this category are computed with the FMA using a cluster size equal to the box size for the level. Finally, the interactions with the remaining boxes in the level are categorized as far-away and handled by levels above using larger clusters. Defined in this way, the three categories—neighbors, non-touching neighbors, and faraway—are mutually exclusive. As a matter of implementation, each box of the quadtree should maintain lists identifying its neighboring boxes and non-touching neighbors on the same level.

D. Pressure evaluation using a tree traversal

Recall the ultimate purpose of the algorithm: to replace the costly matrix-vector product $\mathbf{G}_{rad}\mathbf{u}$ with a more efficient calculation. For a given set of node displacements $\mathbf{u}$ (or monopole strengths $\mathbf{q}$), the evaluation of acoustic pressure at every node is carried out by a single traversal of the quadtree. The traversal can be divided into three parts: an upward pass, a downward pass, and an evaluation; we describe these parts explicitly here.

Upward pass:

- 1) At the bottommost level $\mathcal{L}_{max}$, the far-field signatures are calculated using (9) for every box

$$F_{\mathbf{a}}(\hat{\mathbf{s}}) = \sum_{\text{node } n} q_n e^{ik(\mathbf{a}-\mathbf{r}_n) \cdot \hat{\mathbf{s}}} \text{ for } box \in \mathcal{L}_{max}$$

where $\mathbf{a}$ is the box center, and q_n and $\mathbf{r}_n$ are the strength and position, respectively, of the nodes in the box.

- 2) Moving up one level, the boxes in this level acquire the far-field signatures from their children by a shift-sum-interpolate operation

$$F_{\mathbf{a}}(\hat{\mathbf{s}}) = \text{Interp} \left[\sum_{child\ j} e^{ik(\mathbf{a}-\mathbf{a}_j)\cdot\hat{\mathbf{s}}} F_{\mathbf{a}_j}(\hat{\mathbf{s}}) \right] \text{ for } box \in \mathcal{L}$$

where $\mathbf{a}$ is the box center and $F_{\mathbf{a}_j}(\hat{\mathbf{s}})$ is the far-field signature of the j -th child about its center $\mathbf{a}_j$.

- 3) Step 2 is repeated for the remaining levels up to and including $\mathcal{L}_2$. At the conclusion of the upward pass, every box in the levels $\mathcal{L}_{max}, \dots, \mathcal{L}_2$ will have a far-field signature.

Downward pass:

- 4) Beginning with $\mathcal{L}_2$, each box in the level acquires the far-field signatures from its non-touching neighbors by translating them one-by-one (converting them to near-field signatures in the process) and summing them together

$$N_{\mathbf{b},L}(\hat{\mathbf{s}}) = \sum_{n \neq j} T_L(\hat{\mathbf{s}}, \mathbf{b} - \mathbf{a}_j) F_{\mathbf{a}_j}(\hat{\mathbf{s}}) \text{ for } box \in \mathcal{L}$$

where $\mathbf{b}$ is the box center and $F_{\mathbf{a}_j}(\hat{\mathbf{s}})$ is the far-field signature of the j -th non-touching neighbor about its center $\mathbf{a}_j$.

- 5) Moving down one level, each box inherits the near-field signatures from their parent by a filter-shift operation

$$N_{\mathbf{b}_j,L}(\hat{\mathbf{s}}) = \text{Filter} \left[e^{ik(\mathbf{b}_j - \mathbf{b})\cdot\hat{\mathbf{s}}} N_{\mathbf{b}}(\hat{\mathbf{s}}) \right] \text{ for } box \in \mathcal{L}$$

where $\mathbf{b}_j$ is the center of the j -th child box and $N_{\mathbf{b}}(\hat{\mathbf{s}})$ is the near-field signature of the parent about its center $\mathbf{b}$.

- 6) Steps 4 and 5 are repeated for the remaining levels up to and including $\mathcal{L}_{max}$. For every box, the near-field signatures acquired from non-touching neighbors are always aggregated with the signature inherited from its parent. At the conclusion of the downward pass, every box in $\mathcal{L}_{max}$ will have a near-field signature which represents the fields from all non-touching neighbors and far-away boxes.

Evaluation:

- 7) For node m , the pressures due to a node n within the same box is evaluated directly and added to the node's self pressure

$$p_{m,box} = i\omega\rho \sum_{n \neq m} q_n \frac{e^{-ik|\mathbf{r}_m - \mathbf{r}_n|}}{2\pi|\mathbf{r}_m - \mathbf{r}_n|} + \rho c q_m \left[\frac{1}{2\pi} (ka_m)^2 + i \frac{8}{3\pi} ka_m \left(\frac{a_m}{\pi} \right)^{1/2} \right] \quad (15)$$

- 8) The pressures due to a node n in a neighboring box is also evaluated directly

$$p_{m,neighbors} = i\omega\rho \sum_n q_n \frac{e^{-ik|\mathbf{r}_m - \mathbf{r}_n|}}{2\pi|\mathbf{r}_m - \mathbf{r}_n|} \quad (16)$$

- 9) The pressures due to the remaining nodes (those in non-touching neighbor and far-away boxes) are included in the near-field signature of the box which is evaluated using (12)

$$p_{m,other} \approx -\rho c \frac{k^2}{8\pi^2} \sum_{\hat{s}} w_s e^{ik(\mathbf{r}_m - \mathbf{b}) \cdot \hat{s}} N_{b,L}(\hat{s}) \quad (17)$$

where $\mathbf{b}$ is the box center, and w_s is the quadrature weight for angle $\hat{s}$.

The total pressure at the node is calculated by summing the contributions from each part.

$$p_m = p_{m,box} + p_{m,neighbors} + p_{m,other}$$

E. Optimization and preconditioning

Each traversal of the quadtree—consisting of an upward pass, downward pass, and a pressure evaluation—takes as an input the node strengths and returns an approximation of the pressure on each node. Because an iterative solver will perform a traversal in every iteration, redundant operations should be moved outside the loop to reduce the total computation time (at the expense of a small memory cost). For example, the Green's functions used in the direct pressure evaluation can be computed in advance since they depend only on node-to-node distances which remain static during the iterations.

More importantly, the shift and translation operators used in the upward and downward passes can be computed prior to the iterations as they depend only on the box-to-box geometry. The precomputation of the translation operators is of particular importance because doing so will isolate the most expensive calculation in the FMA. Because the translations depend only on the relative position between box pairs, the same translation may be encountered multiple times in the algorithm—a redundancy that can be easily avoided. In a multi-level 2-D algorithm with a quadtree structure, there are a total of 40 unique translation operators which may be used (consider the four possible child box positions and the potential non-touching neighbors in each case—there are 40 total translations covering all these cases). If the total number of quadrature angles in azimuth is divisible by 4, symmetry can be exploited to reduce the number of translation operators that need to be computed

from 40 to 7 per level (per frequency) with the remaining 33 translation operators constructed from simple rotations or reflections.

The total computation time can also be reduced by decreasing the number of iterations needed. Proper preconditioning of the linear system can drastically improve the rate of convergence of iterative methods such as GMRES and BiCG. We use a block-diagonal preconditioner P where the blocks are the BEM matrices (refer to (1)) for the single membrane problem (wherein no membrane acoustic cross-coupling is assumed). P and P^{-1} are sparse and very cheap to construct, with a block size depending on the number of nodes per membrane and a number of matrix inversions equal to the number of unique membrane specifications (typically not more than 2).

$$\mathbf{P}^{mn} = \begin{cases} \mathbf{G}_{mech}^{mn} + i\omega \mathbf{Z}_{rad}^{mn} & \text{if } m, n \text{ are in same membrane} \\ 0 & \text{otherwise} \end{cases}$$

P^{-1} is also block-diagonal with blocks that are the inverse of the corresponding blocks in P .

F. Controlling error

A powerful feature of the FMA is the ability to optimize the trade-off between runtime and accuracy for a particular application. The numerical error incurred by the algorithm come from the truncation of the infinite series and evaluation of the integral by numerical quadrature in (6). The error is controlled primarily by adjusting the order of the truncated operator T_L ; enough terms of the series should be kept so that T_L converges within the desired tolerance, but not so many as to incur an unnecessary computational penalty.

A well-known deficiency of the FMA is the susceptibility of the operator T_L to numerical instabilities due to the asymptotic behavior of the spherical Hankel functions [43]. For $l \gg kx$, $h_l(kx)$ diverges, and its representation in floating point introduces round-off error that will compromise the overall accuracy of the algorithm. This effectively imposes an upper bound on the truncation order L , and therefore a limit on the achievable accuracy. The upper bound depends on the product kx_{min} where x_{min} is the minimum distance between box centers which the translation operator will be used. The breakdown therefore becomes a problem at low frequencies—where the wavelength is large compared to the size of the boxes—and will also depend on the maximum quadtree level used.

To determine the optimum truncation order at each frequency, we consider the worst-case scenario of a translation from one box to another box separated by a one-box buffer of length d (see Fig. 2). This empirical case also serves to determine the onset of the low-frequency breakdown for a typical problem size and frequency range of interest. Nine sources and nine evaluation locations are simulated (at the center and on the periphery of each box), for a total of 81 combinations. Starting with $L = 3$, the truncation order is increased until either breakdown occurs or the maximum relative error reaches 1% or less. The box sizes and separation distances were selected for a 4×4 mm design space.

The optimal orders determined by this method and the errors incurred are plotted in Fig. 9a and Fig. 9b, respectively. As expected, the error remains within 1% for most frequencies in the range of interest. Breakdown occurs for frequencies below 780 kHz for $\mathcal{L}_7$, 390 kHz for $\mathcal{L}_6$, and 190 kHz for $\mathcal{L}_5$ and the error in the breakdown region does not exceed 3.2%. These small error penalties occur at frequencies that are not relevant for most medical imaging applications.

III. Simulation examples

We developed an implementation of the fast multipole algorithm for membrane-type arrays in Python [53] with the use of additional open-source packages: SciPy (scientific computing) [54], NumPy (N-dimensional arrays and linear algebra) [55], and Cython (optimising compiler) [56]. The Loose General Minimum Residual (LGMRES) algorithm [57] implemented in NumPy is used for the iterative solver in our FMA code. Direct solutions are obtained using NumPy's linear algebra solver which is a wrapper for the LAPACK gesv routine [58].

Our implementation of the FMA uses Fourier-based interpolation and filtering procedures which require special modifications to the calculation of T_L to handle discontinuities in the integration weight $/\sin(\theta)/$ (refer to [50] for details). The Fourier-based scheme is used in conjunction with a trapezoidal quadrature rule with a uniform sampling in θ and ϕ

$$\theta_m = \frac{\pi m}{M} \text{ for } -M+1 \leq m \leq M-1 \quad (18)$$

$$\phi_n = \frac{\pi n}{N} \text{ for } -N \leq n \leq N-1 \quad (19)$$

where the number of quadrature points M and N are related to the order of the translation operator L by $M = 2L + 1$ and $N = 2(L + 1)$. Storage of the sampled functions can be reduced in half by following the guidelines provided in [50], [49].

Pre-caching of the translation operators was performed for a 6-level FMA scheme (levels $\mathcal{L}_2$ to $\mathcal{L}_7$) and a 4×4 mm design area. This step took approximately one day of computation on a 12-core computer for a very fine frequency sweep from 0 – 50 MHz in 0.05 MHz steps. Once computed and stored, the translation operators can be reused for any arbitrary configuration of nodes which fit into the design area. Although the computation of the translation operators takes a significant amount of time, its computational cost is amortized over all subsequent simulations.

Simulations were carried out on a GNU/Linux computer (4x AMD Opteron 6376 CPUs) with 64 effective threads running at 1.4 GHz each and 256 GB of total memory. Frequencies were simulated from 0 – 50 MHz in 0.25 MHz steps.

A. Simulation of a 1-D CMUT array element

A single element of a larger 32-element 1-D CMUT array was simulated to compare our FMA solver with the direct solution. The element consists of a 45×2 grid of CMUT membranes with a pitch of $55 \mu\text{m}$ in both directions. The CMUT membranes are $45 \times 45 \mu\text{m}$ squares with a total membrane thickness of $2.2 \mu\text{m}$, a silicon nitride isolation layer of $0.2 \mu\text{m}$, gap of 47 nm , and a damping coefficient of $10,000 \text{ Pa} \cdot \text{s/m}$. Each membrane was meshed with a 13×13 grid of nodes (excluding clamped nodes), chosen such that the total node count of 15,210 is just within the limit of feasibility for the direct solver. At each frequency, the membranes were given a 9 V DC bias and were excited uniformly with a 1 V AC signal (a plane wave excitation). The simulations were performed under fluid loading conditions with a fluid density of 1000 kg/m^3 and a sound speed of 1540 m/s . These membrane and fluid properties were used in all subsequent simulations unless otherwise noted.

The nodal displacements were calculated using our FMA solver and compared with the direct solution. Two measures were used to determine the displacement error: normalized root mean squared error (NRMSE) which normalizes to the range of observed values and the relative error (RE) which normalizes to the L_2 norm.

$$\begin{aligned} \text{NRMSE} &= \frac{\sqrt{\frac{1}{N} \sum (|\hat{x}_{FMA}| - |\hat{x}_{direct}|)^2}}{\max(|\hat{x}_{direct}|) - \min(|\hat{x}_{direct}|)} \\ \text{RE} &= \frac{\|\hat{x}_{FMA} - \hat{x}_{direct}\|}{\|\hat{x}_{direct}\|} \end{aligned}$$

The RE is bounded above by the residuals in proportion to the condition numbers κ of the preconditioner P and the right preconditioned system GP^{-1} .

$$\frac{1}{\kappa(P)} \frac{\|\hat{x} - x\|}{\|x\|} \leq \frac{\|P\hat{x} - Px\|}{\|Px\|} \leq \kappa(GP^{-1}) \frac{\|\hat{b} - b\|}{\|b\|}$$

This expression is useful as a guess for the necessary tolerance needed in LGMRES because the condition numbers can be estimated ($\kappa(P)$ can be found exactly and cheaply from its block-diagonal property, while a guess for $\kappa(GP^{-1})$ must be assumed).

The NRMSE and RE at each frequency is plotted in Fig. 5. We can see that the FMA solution has converged sufficiently to the direct solution with errors much less than 1%. With the block-diagonal preconditioner, the FMA solver converges rapidly with an average of 2.5 iterations of LGMRES and a maximum of 4 iterations. Several peaks in the error are observed at frequencies which likely correspond to particular resonances (and anti-resonances) of the array or the membrane where the conditioning of the system is highest.

The pressure response at 3 cm from the center of the array is shown in Fig. 6. The FMA solver recovers both pressure magnitude and phase with close agreement.

B. Simulation of a large 1-D CMUT array

We simulated several excitation cases for a large 32-element 1-D CMUT array to demonstrate the ability of our FMA solver to simulate realistic imaging arrays in their entirety. The array is composed of a 45×64 grid of CMUT membranes (2880 total membranes) with a pitch of $55 \mu\text{m}$ in both directions (see Fig. 7). The membranes are organized into 32 elements, yielding an effective element pitch of $110 \mu\text{m}$ which satisfies the $\lambda/2$ criterion for a 7 MHz imaging array.

To study the effect of the higher-order membrane modes, simulations were performed for membrane mesh grids of 3×3 , 5×5 , 7×7 , and 9×9 moving nodes. The total node count for each simulation was 25,920, 72,000, 141,120, and 233,280 nodes, respectively. It is expected that a finer node density will improve the sampling of higher-order membrane mode shapes and sampling of the array edges which may impact the predicted array modes.

First, the array was simulated at each node density with a uniform excitation of the first element only (the leftmost element). The single element excitation case is useful for understanding the crosstalk tendencies of the array and also for practical scenarios such as synthetic beamforming where a small number of elements are excited sequentially. The excited element was given a 9 V DC bias, while the remaining elements were unbiased.

For this case, the pressure response was calculated at 3 cm from the center of the array and is shown in Fig. 9, where w is the membrane width. Significant cross-talk is observed in the 3 – 7 MHz range (the single membrane fundamental mode is around 6.1 MHz), related to modes of the array similar to those predicted by eigenanalysis of small CMUT arrays [23]. Interestingly, while the crosstalk is predicted in all the simulations, convergence is not observed until a sampling length of $w/8$ or smaller is used. 2-D images of the mean membrane displacement magnitude at 5.5 MHz (see Fig. 10) reveal critical differences in the structure of the predicted array mode between the $w/4$ and $w/10$ meshes, likely due in part to insufficient sampling of the array edges. Many membranes were also found to be moving in higher-order modes, which would not be properly sampled by the coarse mesh. The displacement profiles, constructed from the displacement of the center node of each membrane, along the leftmost column and bottommost row is shown in Fig. 11. The results indicate that a fine mesh density, about 7 nodes over the width of the membrane, is necessary to accurately predict crosstalk effects, even in the lower operating band of the transducer.

Strong crosstalk is also predicted in the 14 – 17 MHz range (see Fig. 9), related to cross-coupling of higher-order membrane modes. The difference in the predicted frequency of the crosstalk differs by nearly 2 MHz between the $w/4$ and $w/10$ meshes, indicating the significance of the mesh density on the accuracy of the result. A detailed plot of the membrane motion is shown in Fig. 12 for a frequency of 16.5 MHz, confirming the presence of a variety of higher-order membrane modes.

Next, the array was simulated with a phased excitation of all the elements at a focus 3 cm from the center of the array. The pressure response magnitude for each node density is plotted in Fig. 9. The cross-talk effects observed in the previous case are smoothed out due

to forcing of all the membranes. Above 18 MHz (about 3 times the single membrane fundamental), we see that the prediction for the first anti-resonance, the second resonance peak, and the overall shape of the response vary significantly depending on the node density. We begin to see convergence of the solutions with a sampling length of $w/8$ and below. A 2-D image of the mean membrane displacement magnitude at 5.5 MHz for the phased case is shown in Fig. 10. Strong periodicity is observed in the array modes in both dimensions, indicating that the finiteness of the array is involved in determining the overall array behavior due to the standing evanescent waves.

C. Computation time and memory usage

Resource usage was measured at each frequency for both our FMA solver and the direct solver. To ensure fairness and accuracy, each frequency was simulated in its own process which was limited to running in a single thread. Solution time for the FMA solver includes the necessary FMA-related overhead, e.g. the setup of the quadtree, loading of the translation operator pre-cache, and the time spent in iterations of LGMRES. The time spent constructing the translation operator pre-cache is not included. Since the design space is generally known in advance and because the pre-cache is reusable for all simulations fitting into the same space, this step is considered as a one-time cost. Solution time for the direct solver includes the generation of the mutual impedance matrix and the time spent performing LU factorization. In all cases, memory usage is reported as the peak usage by the process during its lifetime.

The solution time for all the simulations are shown in Fig. 13a. For the simulation of a single element with 15k nodes, the solution time of the direct solver was constant at around 21 min per frequency. In comparison, the FMA solver averaged about 2 min per frequency with a maximum of 13 min, a 10-fold reduction in the average computation time. For the full 32-element 1-D CMUT array with 230k nodes, the FMA solver spent on average 33 min per frequency and a maximum of 125 min.

Similarly, the FMA solver shows a significant improvement over the direct solver in memory usage (see Fig. 13b). For 15k nodes, the direct solver uses an average of about 16.4 GB per frequency whereas the FMA solver uses an average of only 500 MB, a 32-fold improvement. The memory usage increases for the larger cases but never exceeds 5.0 GB per frequency. The memory usage is observed to increase as a function of frequency as a result of the finer quadrature sampling needed.

IV. Conclusion

We applied the fast multipole algorithm to improve the computational efficiency of a BEM model for membrane-type ultrasonic transducers. With an FMA-accelerated BEM model, simulations of large arrays with thousands of membranes is realized without constraints on the finiteness, periodicity, membrane-type, or phasing of the array. Crucially, by including a mutual radiation impedance term, the model captures the acoustic cross-talk of arrays which produce the dispersive-guided modes that may degrade imaging performance. For a single array element with 90 membranes and 15,210 nodes, our FMA solver was found to be 10 times faster and 32 times more memory efficient than a standard solver using LU

decomposition. This improvement was demonstrated over a wide frequency range up to 50 MHz. We demonstrated the ability of our FMA solver to handle large arrays by simulating a full 32-element CMUT array with 2880 membranes and 233,280 nodes. It was found that the higher-order membrane modes play a significant role in the frequency response and directivity of the array.

A number of improvements can be made to further reduce computational costs. First, while we have demonstrated simulations over a wide 50 MHz frequency band, the total computation time of the FMA solver will be reduced greatly by considering only the bandwidth of interest for a particular application. In addition, an adaptive frequency mesh can be used to avoid unnecessary calculation at frequencies that do not contribute significantly. For example, a coarse frequency sweep could be performed first to determine the general shape and locations of the resonances followed by more refined sweeps adjusted according to the strength of the coarse response.

So far, we have demonstrated the ability to simulate a large array excited by a single phased excitation condition (in the form of a pressure exerted on each node). These types of simulations can be a valuable tool for understanding the underlying cross-talk phenomena of imaging arrays, and for predicting the general characteristics of the array in terms of its frequency response and directivity. For many applications, it is useful to fully characterize the behavior of the array by calculation of its element-to-element lumped mutual impedances. Doing so will yield a set of transfer functions which can be used in electrical models (e.g. large signal analysis for CMUTs [40]) and to simulate arbitrary phasing conditions (e.g., for reconstruction of simulated images). The issue of combining the FMA solver with electrical models for array optimization and electrical analysis will be addressed in future work.

Acknowledgments

This work was partly supported by the National Science Foundation under Grant No. ECCS-1202118 and the National Institutes of Health under Grant No. U01-HL121838. The authors would like to thank Sarp Satir for his useful consultation towards a revised manuscript.

References

1. Newnham, R.; Zhang, J.; Meyer, R, Jr. Cymbal transducers: a review. 2000 12th IEEE Int Symp Applications Ferroelectrics (ISAF 2000); 2000. p. 29-32.
2. Houston, B.; Bucaro, J.; Yoder, T.; Kraus, L.; Tressler, J.; Fernandez, J.; Montgomery, T.; Howarth, T. OCEANS'02 MTS/IEEE. Vol. 1. IEEE; 2002. Broadband low frequency sonar for non-imaging based identification; p. 383-387.
3. Maione E, Shung KK, Meyer RJ Jr, Hughes JW, Newnham RE, Smith NB. Transducer design for a portable ultrasound enhanced transdermal drug-delivery system. IEEE Trans Ultrason, Ferroelect, Freq Contr. 2002; 49(10):1430–1436.
4. Luis J, Park E-J, Meyer RJ, Smith NB. 9F-4 rectangular cymbal arrays for ultrasonic transdermal insulin delivery,” in. 2007 IEEE Int Ultrason Symp. 2007:840–843.
5. Kavros SJ, Schenck EC. Use of noncontact low-frequency ultrasound in the treatment of chronic foot and leg ulcerations: a 51-patient analysis. J Am Podiatr Med Assoc. 2007; 97(2):95–101. [PubMed: 17369314]
6. Savoia A, Mauti B, Caliano G. A low frequency broadband flextensional ultrasonic transducer array. IEEE Trans Ultrason, Ferroelect, Freq Contr. 2015

7. Eriksson T, Ramadas S, Unger A, Hoffman M, Kupnik M, Dixon S. Flexural transducer arrays for industrial non-contact applications. 2015 IEEE Int Ultrason Symp. 2015:1–4.
8. Przybyla RJ, Shelton SE, Guedes A, Izyumin II, Kline MH, Horsley DA, Boser BE. In-air rangefinding with an aln piezoelectric micromachined ultrasound transducer. IEEE Sensors J. 2011; 11(11):2690–2697.
9. Wygant IO, Kupnik M, Windsor JC, Wright WM, Wochner MS, Yaralioglu GG, Hamilton MF, Khuri-Yakub BT. 50 kHz capacitive micromachined ultrasonic transducers for generation of highly directional sound with parametric arrays. IEEE Trans Ultrason, Ferroelect, Freq Contr. 2009; 56(1): 193–203.
10. Lu Y, Tang H, Fung S, Wang Q, Tsai J, Daneman M, Boser B, Horsley D. Ultrasonic fingerprint sensor using a piezoelectric microma-chined ultrasonic transducer array integrated with complementary metal oxide semiconductor electronics. Appl Phys Lett. 2015; 106(26):263503.
11. Wygant IO, Zhuang X, Yeh DT, Oralkan Ö, Ergun AS, Karaman M, Khuri-Yakub BT. Integration of 2D cmut arrays with front-end electronics for volumetric ultrasound imaging. IEEE Trans Ultrason, Ferroelect, Freq Contr. 2008; 55(2):327–342.
12. Cheng X, Lemmerhirt D, Kripfgans O, Zhang M, Yang C, Rich C, Fowlkes J. Cmut-in-CMOS ultrasonic transducer arrays with on-chip electronics. Transducers 2009 – 2009 Int Solid-State Sensors, Actuators and Microsystems Conf. 2009:1222–1225.
13. Eccardt PC, Niederer K. Micromachined ultrasound transducers with improved coupling factors from a CMOS compatible process. Ultrasonics. 2000; 38(1):774–780. [PubMed: 10829770]
14. Noble R, Davies R, Day M, Koker L, King D, Brunson K, Jones A, McIntosh J, Hutchins D, Robertson T, et al. Cost-effective and manufacturable route to the fabrication of high-density 2d micromachined ultrasonic transducer arrays and (cmos) signal conditioning electronics on the same silicon substrate. 2001 IEEE Ultrasonics Symposium. 2001; 2:941–944.
15. Zahorian J, Hochman M, Xu T, Satir S, Gurun G, Karaman M, Degertekin FL. Monolithic CMUT-on-CMOS integration for intravascular ultrasound applications. IEEE Trans Ultrason, Ferroelect, Freq Contr. 2011; 58(12):2659–67.
16. Yeh DT, Member S, Wygant IO, Donnell MO, Khuri-yakub BT, Oralkan O, Wygant IO, O'Donnell M, Khuri-yakub BT. 3-D ultrasound imaging using a forward-looking CMUT ring array for intravascular / intracardiac applications. IEEE Trans Ultrason, Ferroelect, Freq Contr. 2006; 53(6): 1202–1211.
17. Tekes, C.; Zahorian, J.; Xu, T.; Rashid, MW.; Satir, S.; Gurun, G.; Karaman, M.; Hasler, J.; Degertekin, FL. 2013 SPIE Medical Imaging. International Society for Optics and Photonics; 2013. Cmut-based volumetric ultrasonic imaging array design for forward looking Ice and ivus applications,” in; p. 86750B-86750B.
18. Dausch DE, Castellucci JB, Chou DR, Von Ramm OT. Theory and operation of 2-D array piezoelectric micromachined ultrasound transducers. IEEE Trans Ultrason, Ferroelect, Freq Contr. 2008; 55(11):2484–2492.
19. Liao, W.; Liu, W.; Rogers, J.; Usmani, F.; Tang, Y.; Wang, B.; Jiang, H.; Xie, H. Piezoelectric micromachined ultrasound transducer array for photoacoustic imaging. 2013 Transducers & Eurosensors XXVII: 17th Int. Conf. Solid-State Sensors, Actuators, Microsystems; 2013. p. 1831-1834.
20. Lani S, Sabra KG, Degertekin FL. Super-resolution ultrasonic imaging of stiffness variations on a microscale active metasurface. Appl Phys Lett. 2016; 108(8):084104.
21. Bayram B, Kupnik M, Yaralioglu GG, Oralkan O, Ergun AS, Lin DS, Wong SH, Khuri-Yakub BT. Finite element modeling and experimental characterization of crosstalk in 1-D CMUT arrays. IEEE Trans Ultrason, Ferroelect, Freq Contr. 2007; 54(2):418–429.
22. Wilm M, Reinhardt A, Laude V, Armati R, Daniau W, Ballandras S. Three-dimensional modelling of micromachined-ultrasonic-transducer arrays operating in water. Ultrasonics. 2005; 43(6):457–465. [PubMed: 15823320]
23. Lani S, Sabra KG, Degertekin FL. Modal and transient analysis of membrane acoustic metasurfaces. J Appl Phys. 2015; 117(4):045308.

24. Ballandras S, Wilm M, Daniau W, Reinhardt a, Laude V, Armati R. Periodic finite element/ boundary element modeling of capacitive micromachined ultrasonic transducers. *J Appl Phys.* 2005; 97(3)
25. Caronti A, Savoia A, Caliano G, Pappalardo M. Acoustic coupling in capacitive microfabricated ultrasonic transducers: modeling and experiments. *IEEE Trans Ultrason, Ferroelect, Freq Contr.* 2005; 52(12):2220–2234.
26. Senlik MN, Olcum S, Köymen H, Atalar A. Radiation impedance of an array of circular capacitive micromachined ultrasonic transducers. *IEEE Trans Ultrason, Ferroelect, Freq Contr.* 2010; 57(4): 969–976.
27. Oguz H, Atalar A, Koymen H. Equivalent circuit-based analysis of cmut cell dynamics in arrays. *IEEE Trans Ultrason, Ferroelect, Freq Contr.* 2013; 60(5):1016–1024.
28. Atalar A, Koymen H, Oguz HK. Rayleigh-Bloch waves in CMUT arrays. *IEEE Trans Ultrason, Ferroelect, Freq Contr.* 2014; 61(12):2139–2148.
29. Akhbari S, Sammoura F, Lin L. Equivalent circuit models for large arrays of curved and flat piezoelectric micromachined ultrasonic transducers. *IEEE Trans Ultrason, Ferroelect, Freq Contr.* 2016; 63(3):432–447.
30. Rønnekleiv A. CMUT array modeling through free acoustic CMUT modes and analysis of the fluid CMUT interface through Fourier transform methods. *IEEE Trans Ultrason, Ferroelect, Freq Contr.* 2005; 52(12):2173–84.
31. Certon D, Teston F, Patat F. A finite difference model for cMUT devices. *IEEE Trans Ultrason, Ferroelect, Freq Contr.* 2005; 52(12):2199–2210.
32. Meynier C, Teston F, Certon D. A multiscale model for array of capacitive micromachined ultrasonic transducers. *J Acoust Soc Am.* 2010; 128(5):2549–2561. [PubMed: 21110553]
33. Satir S, Degertekin FL. A nonlinear lumped model for ultrasound systems using cmut arrays. *IEEE Trans Ultrason, Ferroelect, Freq Contr.* Oct.2015 62:1865–1879.
34. Greengard L, Rokhlin V. A fast algorithm for particle simulations. *J Comput Phys.* 1987; 73(2): 325–348.
35. Song J, Lu Cai-Cheng, Chew Weng Cho. Multilevel fast multipole algorithm for electromagnetic scattering by large complex objects. *IEEE Trans Antennas Propagat.* 1997; 45(10):1488–1493.
36. Ohno Y, Yokota R, Koyama H, Morimoto G, Hasegawa a, Masumoto G, Okimoto N, Hirano Y, Ibeid H, Narumi T, Taiji M. Petascale molecular dynamics simulation using the fast multipole method on K computer. *Comput Phys Commun.* 2014; 185(10):2575–2585.
37. Tong MS, Chew WC, White MJ. Multilevel fast multipole algorithm for acoustic wave scattering by truncated ground with trenches. *J Acoust Soc Am.* 2008; 123(5):2513–2521. [PubMed: 18529170]
38. Wu H, Liu Y, Jiang W. A fast multipole boundary element method for 3D multi-domain acoustic scattering problems based on the BurtonMiller formulation. *Eng Anal Bound Elem.* 2012; 36(5): 779–788.
39. Wilkes DR, Duncan AJ. Acoustic coupled fluid-structure interactions using a unified fast multipole boundary element method. *J Acoust Soc Am.* 2015; 137(4):2158–2167. [PubMed: 25920865]
40. Satir S, Zahorian J, Degertekin FL. A large-signal model for CMUT arrays with arbitrary membrane geometry operating in non-collapsed mode. *IEEE Trans Ultrason, Ferroelect, Freq Contr.* Nov.2013 60:2426–2439.
41. Zahorian J, Satir S, Degertekin FL. Analytical-Finite Element hybrid model for CMUT arrays with arbitrary membrane geometry. 2012 IEEE Int Ultrason Symp. 2012; (3):584–587.
42. Kinsler, LE.; Frey, AR.; Coppers, AB.; Sanders, JV. *Fundamentals of Acoustics.* 4. John Wiley & Sons, Inc; 2000.
43. Coifman R, Rokhlin V, Wandzura S. Fast multiple method for the wave equation: a pedestrian prescription. *IEEE Antennas Propagat Mag.* 1993; 35(3):7–12.
44. Gerald, CF.; Wheatley, PO. *Applied Numerical Analysis.* 7. Pearson Education; 2004.
45. Yokota R, Ibeid H, Keyes D. Fast multipole method as a matrix-free hierarchical low-rank approximation. 2016 arXiv preprint arXiv:1602.02244.

46. Saad Y, Schultz MH. GMres: A generalized minimal residual algorithm for solving nonsymmetric linear systems. *SIAM J Sci Stat Comput.* 1986; 7(3):856–869.
47. Abramowitz, M.; Stegun, IA. *Handbook of mathematical functions: with formulas, graphs, and mathematical tables.* Vol. 55. Courier Corporation; 1964.
48. Rahola J. Diagonal forms of the translation operators in the fast multipole algorithm for scattering problems. *BIT.* 1996; 36(2):333–358.
49. Sarvas J. Performing interpolation and antepolation entirely by fast Fourier transform in the 3-D multilevel fast multipole algorithm. *SIAM J Numer Anal.* 2003; 41(6):2180–2196.
50. Cecka C, Darve E. Fourier-based fast multipole method for the helmholtz equation. *SIAM J Sci Comput.* 2013; 35(1):A79–A103.
51. Ohnuki S, Chew WC. Numerical analysis of local interpolation error for 2D-MLFMA. *Microw Opt Techn Let.* 2003; 36(1):8–12.
52. Jakob-Chien R, Alpert BK. A Fast Spherical Filter with Uniform Resolution. *J Comput Phys.* 1997; 136(2):580–584.
53. Python Software Foundation. <http://www.python.org>
54. Jones, E.; Oliphant, T.; Peterson, P., et al. SciPy: Open source scientific tools for Python. 2001. <http://www.scipy.org/>
55. Van Der Walt S, Colbert SC, Varoquaux G. The numpy array: a structure for efficient numerical computation. *Comput Sci Eng.* 2011; 13(2):22–30.
56. Behnel S, Bradshaw R, Citro C, Dalcin L, Seljebotn D, Smith K. Cython: The best of both worlds. *Comput Sci Eng.* 2011; 13(2):31–39.
57. Baker AH, Jessup ER, Manteuffel T. A technique for accelerating the convergence of restarted GMRES. *SIAM J Matrix Anal A.* 2005; 26(4):962–984.
58. Anderson, E.; Bai, Z.; Bischof, C.; Blackford, S.; Demmel, J.; Dongarra, J.; Du Croz, J.; Greenbaum, A.; Hammarling, S.; McKenney, A.; Sorensen, D. *LAPACK users' guide.* 3. Philadelphia, PA: Society for Industrial and Applied Mathematics; 1999.

Fig. 1.

(a) Geometry of the fundamental pressure evaluation problem in the fast multipole algorithm. (b) Multi-level quadtree structure which organizes nodes into boxes of decreasing size. (c) Diagram illustrating the shift, interpolate, and filter operations of the multi-level algorithm. In the upward pass, parent boxes acquire far-field signatures from their children using a shift-sum-interpolate operation. In the downward pass, child boxes inherit near-field signatures from their parents using a filter-shift operation.

Fig. 2.

The worst-case translation in the multi-level FMA with a one-box buffer scheme is used to determine the optimal translation order L for each level at each frequency. The worst-case considers the translation from each of the nine locations in the source box to the nine locations in the target box.

Fig. 3.

(a) Optimum translation order L needed to achieve 1% error tolerance as a function of frequency, determined using the worst-case translation. (b) The maximum error in the worst-case translation if the optimum translation order L is used. Breakdown of the translation operator is a problem at frequencies below 780 kHz, limiting the achievable tolerance to about 3%.

Fig. 4.

A single element of a 1-D CMUT array with 90 membranes (arranged in two columns) and 15,210 nodes. The membranes are $45 \times 45 \mu\text{m}$ squares with a pitch of $55 \mu\text{m}$. Because the BEM equations for a simulation of this size can be solved directly, this array is used to validate our FMA solver.

Fig. 5.

Normalized root mean squared error (NRMSE) and relative error (RE) for the node displacements of the simulated element. The element was simulated with FMA and compared with the direct solution. The agreement is within 0.5% at all frequencies.

Fig. 6.

Pressure response magnitude (top) and phase (bottom) for the simulated element at 3 cm from the center of the array. The membranes of the element were given a 9 V DC bias and excited uniformly.

Fig. 7.

A large 32-element 1-D CMUT array with 2880 membranes and up to 233,280 nodes (for a 9×9 node mesh per membrane). The membranes are $45 \times 45 \mu\text{m}$ in size with a membrane pitch of $55 \mu\text{m}$ and an element pitch of $110 \mu\text{m}$. Each element is formed from a 45×2 grid of CMUT membranes.

Fig. 8.

An example of the box-to-box interactions of the multi-level FMA used in the simulation of a 1-D CMUT array. The nodes of the array are assigned to boxes (shown in a black outline) based on the subdivision of a 4×4 mm space using a quadtree. The source boxes get larger as they get farther away from the target box (shown in red) in order to reduce the total number of interactions that need to be calculated.

Fig. 9.

Pressure response magnitude for the 1-D CMUT array at 3 cm from the center of the array for two excitation cases. The simulation was performed for decreasing node sampling lengths where w is the membrane width. Top: Only the first element of the array is excited. Strong cross-talk is observed in the 3 – 7 MHz and 14 – 17 MHz bands (close-up is shown). Center: Full response for the same case. Bottom: All 32 elements are phased. Significant differences are observed above 18 MHz due to higher-order membrane modes that are not sufficiently sampled with a coarse node mesh.

Fig. 10.

2-D images of the mean membrane displacement of the 1-D CMUT array at 5.5 MHz. The images are plotted on a log scale with 25 dB dynamic range. Note that the spaces between the membranes have been removed for these plots. Top: Array mode when only the first element is excited, simulated using the coarsest mesh ($w/4$). Center: Array mode when only the first element is excited, simulated using the finest mesh ($w/10$). Bottom: Array mode when all elements are phased, simulated using the finest mesh ($w/10$).

Fig. 11.

Displacement profiles of the array at 5.5 MHz, constructed from the displacement of the center node of each membrane, for the case of excitation of the first element only. Top: The profile of the leftmost column of membranes which are all excited. Bottom: The profile of the center row of membranes where the first two membranes are excited.

Fig. 12.

The membranes of the 1-D CMUT array at a frequency of 16.5 MHz are excited into a variety of higher-order modes due to cross-talk over the entire array.

Fig. 13.

(a) Comparison of simulation times for the FMA solver and direct solver. The FMA solver is about 10 times faster for the simulation of a single element and remains within reasonable speeds for simulations of the full array. (b) Comparison of peak memory usage for the FMA solver and direct solver. The FMA solver uses around 32 times less memory for the simulation of a single element. The memory usage for the full array remains below 5.0 GB and within the range of feasibility.