NiHu: A multi-purpose open source fast multipole solver

Péter Fiala, Péter Rucz
Laboratory of Acoustics and Studio Technologies,
Budapest University of Technology and Economics, 1117 Budapest, Hungary, Email: fiala@hit.bme.hu

Introduction

NiHu [1] is an open source C++ template library for the efficient discretisation of integral equations, with the main application area in the field of boundary element methods. The library, published in 2014, is capable of discretising integral equations related to a large variety of kernel functions, handles 2D and 3D problems, scalar and tensor valued kernels, and enables using various formalisms, such as the collocation and the Galerkin methods.

The present paper introduces the fast multipole extension of the open source library. The upcoming section briefly describes the main design considerations and architectural setup of the framework. This is followed by two numerical examples that demonstrate versatile application possibilities in the field of computational acoustics.

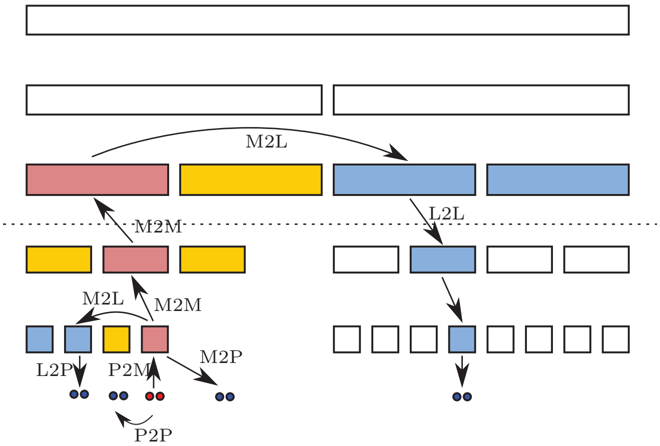


Figure 1: Scheamatic plot of the hierarchical clustering of the computational domain and the recursive application of the FMM operators.

Software architecture

Conventional BEM

The conventional part of the NiHu library is used to discretise integrals of the form

$$K_{ij} = \int_A a_i(x) \int_B K(x,y) b_j(y) \mathrm{d}y \mathrm{d}x, \qquad (1)$$

where K is a kernel function, a_i denotes a test fuction and b_j is a trial function. The software package consists of a core module and a library. As NiHu is a template library, the core describes the interfaces of the classes involved, and the library defines specific component implementations, such as element types, weighting functions, kernels, quadratures, and specialised integration methods for singular kernels. C++ template metaprogramming is extensively exploited in the core, resulting in optimized algorithm and data structure selection during compilation, and yielding good runtime performance, whilst maintaining a high level of polymorphism in the programming environment.

Fast multipole BEM

The fast multipole extension is designed for computing matrix–vector products

$$f_i = \sum_j K_{ij} s_j \qquad (2)$$

based on the Fast Multipole Method (FMM). The FMM relies on the expansion of the kernel function $K(x, y)$

$$K(x,y) \approx L2P(x,X) \cdot M2L(X,Y) \cdot P2M(Y,y), \qquad (3)$$

where X and Y denote cluster centres such that $|Y-X| > |X-x|$ and $|Y-X| > |Y-y|$, and the operators are the particle-to-multipole (P2M), local-to-particle (L2P) operators, and the translation operator (M2L). The expansions can be defined in a kernel-specific manner, like the fast multipole method for the Helmholtz equation in 3D [2], or a kernel-independent manner, like the black-box FMM for asymptotically smooth kernels [3, 4]. Making use of a hierarchical clustering of the computational domain, and recursively summing up all the source contributions when traversing the cluster tree, results in a fast, $O(N)$ algorithm of the matrix vector product. This traversal is schematically displayed in Figure 1.

When designing the FMM extension of the NiHu library, the primary goal was to establish a programming environment, into which different versions of the general FMM method are easily incorporated, and where a transparent and straightforward transition between the FMM and a fast multipole boundary element method (FMBEM) can be defined.

The actual version of the applied FMM algorithm is defined by the decomposition of the kernel. The decomposed operators are template parameters of the general FMM algorithm, and are used in the library as follows in order to implement a specific FMBEM algorithm.

Operator assembly

Figure 2 shows the assembly of the FMM operators in the framework before they are applied in the parallel matrix–vector product routines. In the first stage, the FMM operators acting on particles (P2P, P2M, and L2P) are subjected to numerical integration, defined by the ap-

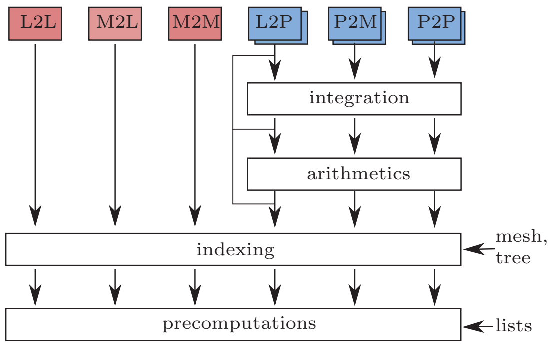


Figure 2: Steps of operator assembly in the NiHu FMM framework.

plied element type, weighting functions, and quadratures. This is the stage where the FMM is transformed into a FMBEM.

The subsequent arithemetics layer is used for defining arithemetic operations on kernels. These operations are useful for combined integral equations, such as the Burton and Miller method, where the L2P operator of the original kernel and that of its gradient need to be combined by a coupling coefficient. An other application example of the arithmetics layer is the assembly of concatenated operators consisiting of the P2M and its gradient that are used for the computation of field point pressures in the acoustical BEM.

After the arithmetic layer, the mesh and the cluster tree are attached to the system, enabling indexing the operators using element and cluster indices. From this point on, the operators can be interpreted as sparse matrices.

Finally, the interaction lists are added to the operators, making accelerations based on translation and rotation invariance or symmetry of the operators possible. The resulting operators are used as inputs of the parallel matrix–vector product routines.

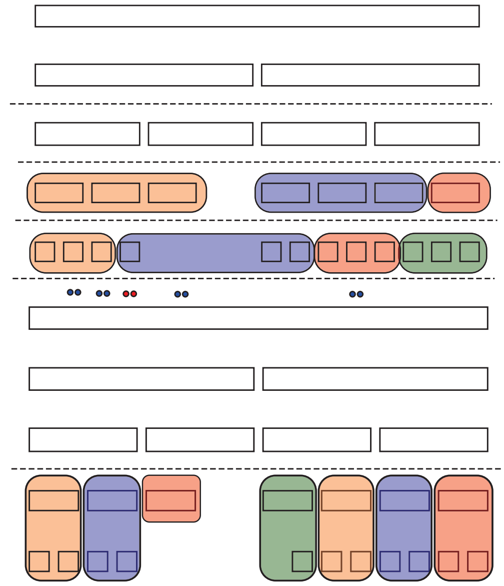
Parallel matrix–vector products

Parallel evaluation of the matrix–vector products is a key feature of the fast multipole method. In the present version of the toolbox, shared memory systems are supported, and parallel algorithms are implemented using the OpenMP standard.

Two main approaches of parallel matrix–vector products are implemented. The so-called grouped fork–join architecture [5] divides the tree traversal algorithm into sequential traversing of levels, and groups of clusters within each level are assigned to individual processor cores, as shown in Figure 3(a). The aim of grouping is to achieve better locality of threads working in parallel. The drawback of the grouped fork–join is the execution barrier between subsequent levels, where threads need to wait for concurrent threads before entering the next level.

The other implemented approach is displayed in Figure 3(b). In this architecture, the tree is cut at a specific level defined by the number of working threads, and separate subtrees below the cut level are assigned to parallel

(a)



(b)

Figure 3: Schematic illustrations of (a) the grouped fork–join and (b) the depth-first-search traversal arhitecture.

threads. This approach has the advantage of less execution barriers, but can be disadvantageous from the perspective of memory management.

Test cases

The European Acoustics Association has launched a benchmark project, aiming to provide useful benchmark cases for acoustic radiation and scattering problems [6]. Two exterior problems are investigated using the NiHu framework: a 2D scattering and a 3D radiation problem.

The Pac-Man

The Pac-Man problem is a 2D exterior acoustic problem defined and analytically solved in [7]. The goal is to compute the pressure field radiated by a radially pulsating Pac-Man, as shown in Figure 4(a), or to compute the pressure field scattered by the Pac-Man when illuminated by a line source as shown in Figure 4(b).

The applied formalism is an exterior direct collocational BEM with the Burton and Miller formalism, and the Pac-Man is meshed using constant line elements. Nearly singular and singular integrals of the weakly, strongly and hypersingular kernels are handled by a static part extraction technique, where the singular static parts are integrated analytically.

For the far field computations, the wideband FMM of the 2D Helmholtz equation is applied [8, 9], where the translation operators are written as convolutions by cylindrical harmonics in the low wavenumber range $kD < 3.0$ (D denoting the cluster size), and the translation is accelerated by means of FFT in the large cluster domain. The applied cluster tree is imbalanced, and the kernel expansion error, i.e. the error introduced by (3), is set to 10^{-13}. Due to the imbalancedness of the tree, P2L and M2P operations are possible when traversing the tree. In order

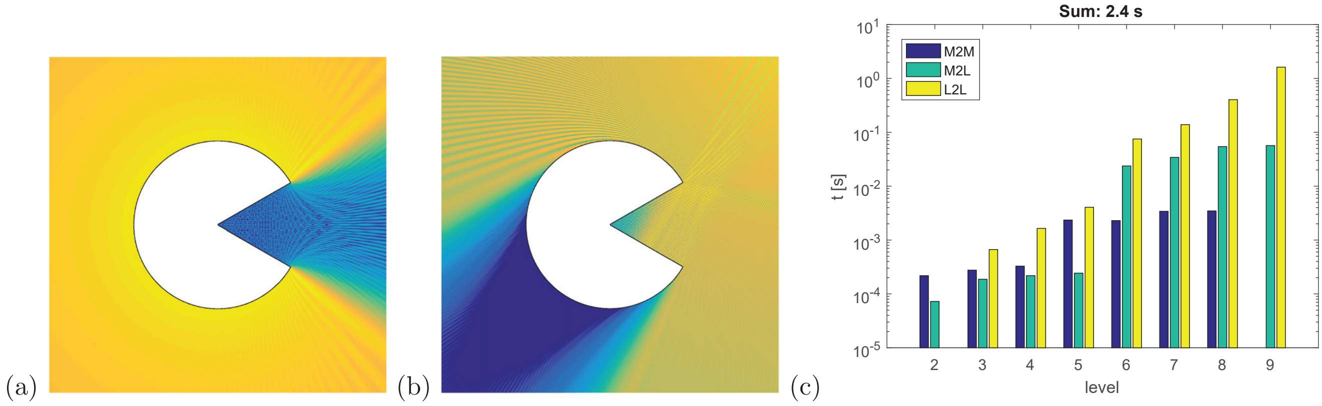


Figure 4: Solution of the Pac-Man problem. (a) pressure field radiated by a radially pulsating Pac-Man. (b) pressure field scattered by the Pac-Man, when illuminated by a line source. (c) computation times of the single matrix–vector product evaluating the field point pressures.

to limit the expansion lengths near the leaf level, where the number of clusters is very large, these interactions were replaced by near field P2P interactions.

Figure 4(c) shows the execution time of the field-point matrix–vector products (radiation from 10^4 elements to 13×10^6 field points). Apparently, the small-cluster range (high levels) take up significant part of the execution time, meaning that both the grouped fork–join, both the depth-first-search traversal parallel algorithms can successfully speed up the matrix–vector product.

The Radiatterer

The Radiatterer [10] is a 3D brick-like rectangular geometry with lots of rectangular exclusions and cavities, containing a Helmholtz resonator. The Radiatterer radiates with constant normal surface velocity, and the goal is to compute the radiated field in a number of field points. Although the mesh is simple in the sense that it can easily be meshed using square surface elements, the iterative solution is computationally demanding due to the lots of resonances in the radiated pressure field (see Figure 5).

In this case, the applied formalism is a collocational direct BEM with the Burton – Miller formalism, and the model is discretised using discontinuous linear square boundary elements. For the near field, weakly singular integrals are computed by a polar coordinate transform, while strongly and hypersingular integrals are evaluated using Guiggiani's method [11]. For the accurate handling of nearly hypersingular integrals, the adaptive quadrature proposed by Telles [12] is applied.

The far field is integrated using the high-frequency diagonal FMM for the 3D Helmholtz kernel [2]. The cluster tree is balanced, and the leaf level is defined where $kD < \pi/4$. The solver is GMRES without restarts.

As there is no analytical solution for this complex radiation problem, the radiated pressure field was compared to results of finite element–infinite element computations. The results of this comparison are shown in Figure 6. Apparently, apart from the numerical dispersion of the FE model, the two approaches yield the same results. The

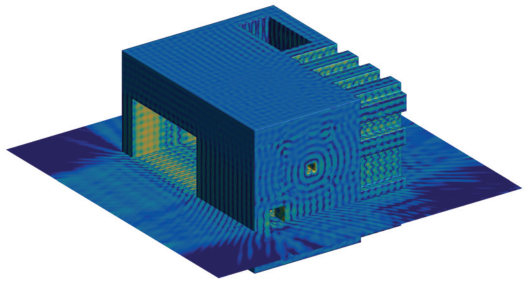


Figure 5: The Radiatterer radiating at $f = 5000$ Hz, corresponding to a Helmholtz number ≈ 600. The surface is meshed using 600 000 elements, resulting in 2 400 000 DOF.

BE computations were performed using the conventional BEM in the low frequency range, and the FMBEM in the higher frequency range. The two frequency ranges were defined with an overlapping segment in order to check the validity of both approaches.

Figure 6(b) shows the number of iterations needed to reach a backward relative solution error 10^{-8}. It is worth noting that due to the complex radiation pattern, the BiCGStab solver failed to converge within an acceptable number of iterations, and even GMRES with restart after a few hundreds of inner iterations did not converge. The number of required iterations exceeds 1 000 in the higher frequency range, meaning that an efficient preconditioner would be very important in this case.

Conclusions

The NiHu library was extended by a Fast Multipole module, capable of evaluating matrix–vector products after the discretisation of integral equations, using the Fast Multipole Method. It was demonstrated that the library can be used to implement different versions of the FMM in a generic environment, and a transparent transition between the FMM and the FMBEM can be defined.

The generic implementation of the library makes it possible to investigate FMM-specific algorithmic developments, such as parallel matrix–vector product evalua-

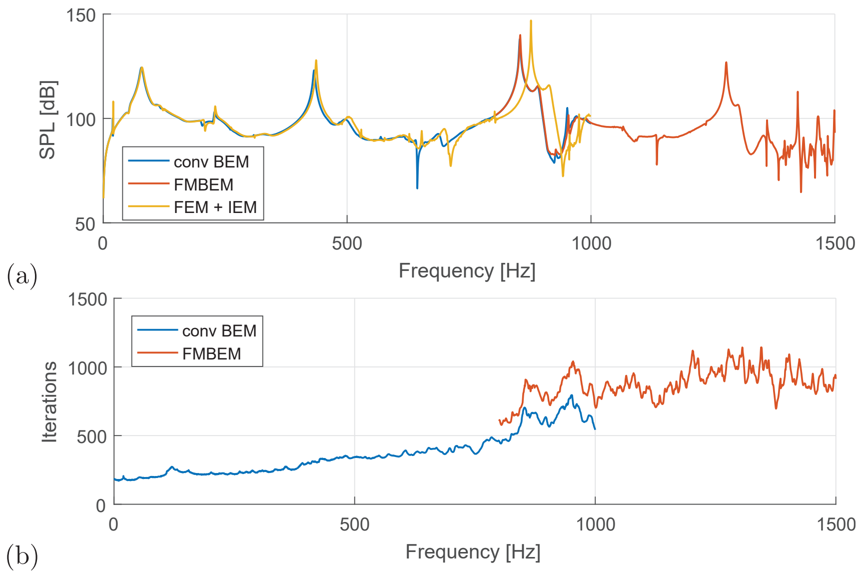


Figure 6: (a) The pressure field radiated by the radiatterer (field point #4 at $[0.5, 1.6, 0.8]$ m), and (b) the required number of iterations to reach a relative backward solution error 10^{-8}.

tions or nested FMM preconditioners on a large variety of applications. Although not demonstrated in this paper, it is worth mentioning that the generalized FMM matrices of NiHu can also be used together with the built-in eigensolvers of Matlab enabling the eigenvalue–eigenvector analysis of large-scale problems using fast multipole methods.

Acknowledgements

P. Rucz acknowledges the support of the Bolyai János research grant provided by the Hungarian National Academy of Sciences.

Emberi Erőforrások Minisztériuma

Supported by the ÚNKP-18-4 New National Excellence Program of the Ministry of Human Capacities.

References

[1] P. Fiala and P. Rucz. NiHu: an open source C++ BEM library. *Advances in Engineering Software*, 75:101–112, 9 2014.

[2] N.A. Gumerov and R. Duraiswami. A broadband fast multipole accelerated boundary element method for the three dimensional Helmholtz equation. *Journal of the Acoustical Society of America*, 125(1):191–205, 2009.

[3] W. Fong and E. Darve. The black-box fast multipole method. *Journal of Computational Physics*, 228:8712–8725, 2009.

[4] C. Schwab and R.A. Todor. Karhunen-Loéve approximation of random fields by generalized fast multipole methods. *Journal of Computational Physics*, 32(217):100–122, 2006.

[5] E. Agullo, B. Bramas, O. Coulaud, M. Khannouz, and L. Stanisic. Task-based fast multipole method for clusters of multicore processors. Technical Report RR-8970, Inria Bordeaux Sud-Ouest, 2017.

[6] M. Hornikx, M. Kaltenbacher, and S. Marburg. A platform for benchmark cases in computational acoustics. *Acustica-Acta Acustica*, 101(4):811–820, 2015.

[7] H. Ziegelwanger and P. Reiter. The PAC-MAN model: Benchmark case for linear acoustics in computational physics. *Journal of Computational Physics*, 346:152–171, 2017.

[8] W Crutchfield, Z Gimbutas, L Greengard, J Huang, V Rokhlin, N Yarvin, and J Zhao. Remarks on the implementation of the wideband FMM for the Helmholtz equation in two dimensions. *Contemporary Mathematics*, 408, 01 2006.

[9] 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. *Journal of Computational Physics*, 216(300–325), 2006.

[10] S. Marburg. The Burton and Miller method: Unlocking another mystery of its coupling parameter. *Journal of Computational Acoustics*, 24(1):1550016:1–20, 2016.

[11] M. Guiggiani, G. Krishnasamy, T.J. Rudolph, and F.J. Rizzo. A general algorithm for the numerical solution of hypersingular boundary integral equations. *Journal of Applied Mechanics, Transactions of the ASME*, 59:604–614, 1992.

[12] J.C.F. Telles. A self-adaptive co-ordinate transformation for efficient numerical evaluation of general boundary element integrals. *International Journal for Numerical Methods in Engineering*, 24:959–973, 1987.