Available online at www.prace-ri.eu

Partnership for Advanced Computing in Europe

PARTS — Electromagnetic Simulations with Fast Boundary Element Methods using HPC

Jürgen Zechner[a,*], Lars Kielhorn[a], Thomas Rüberg[a], Soner Steiner[b]

[a]TAILSIT GmbH, Graz, Austria
[b]VSC Research Center, TU Wien, Vienna, Austria

Abstract

TAILSIT is a company based in Graz, Austria, that produces custom-fit simulation software tools for computational electromagnetics and structural analysis. Its unique selling proposition is a numerical software library based on a coupled Finite/Boundary Element Method (FEM/BEM) for the efficient analysis of electromagnetic problems. A major building block of the software is represented by the BEM which, in its naive implementation, has a quadratic complexity: runtime and memory requirements scale with the second power of the system size. In order to overcome this limitation, TAILSIT uses the Fast Multipole Method (FMM) where the scaling of the computational costs reduce to almost linear complexity. But even with this acceleration technique, the use of the BEM is rather limited on an average desktop workstation. During this SHAPE project TAILSIT's electromagnetic simulation software has been ported to HPC machines with the support of the Vienna Scientific Cluster (VSC, https://vsc.ac.at) at TU Wien and using their computing resources. Thorough performance analyses have led to significant runtime improvements and a good overall scaling for up to a few thousand CPUs. Whereas TAILSIT's previous implementation of the FMM could handle about 10^6 degrees of freedom, within this project we achieved problem sizes of up to $50 \cdot 10^9$.

1. Motivation

In many engineering disciplines, the use of Boundary Element Methods (BEM, often referred to as Method of Moments) poses an advantage because a volume mesh is avoided for at least some parts of the domain. For example, in problems of acoustics, electromagnetics or scattering, it can often be very cumbersome to discretise the air region. Moreover, solid parts might move with respect to each other such that tedious remeshing is required. This has led TAILSIT in the past to combine the Finite Element Method (FEM) with BEM in order to exploit the respective advantages of each method [1]. The FEM-BEM coupling approach has been successfully applied to many problems from electromagnetics. The analysis of actuators, such as magnetic valves, or the snapping of permanent magnets are examples where avoiding a volume mesh for the air region is particularly advantageous [2]. Fig. 1 shows the simulation results of two benchmark examples for that approach.

Unfortunately, the standard BEM has an inherent quadratic complexity similar to N-body problems: each degree of freedom directly interacts with every other, as illustrated by the evaluation of the Coulomb potential

$$\varphi(x_i) = \frac{1}{4\pi\varepsilon_0} \sum_{j=1, j\neq i}^{N} \frac{q_j}{|x_i - x_j|}, \qquad 1 \leq i \leq N \tag{1}$$

where N bodies are located at the positions x_i, and each carries a charge q_i. This equation is prototypical for the matrix-vector product within the iterative solution of a system of equations resulting from a BEM discretisation: the vector of charges is multiplied by a fully-populated matrix in order to obtain the vector of potential values. Therefore, the applicability of the BEM for nowadays industrial-sized engineering problems is only feasible with acceleration methods which reduce the scaling of the costs to lower than N^2.

*Corresponding author, e-mail: juergen.zechner@tailsit.com

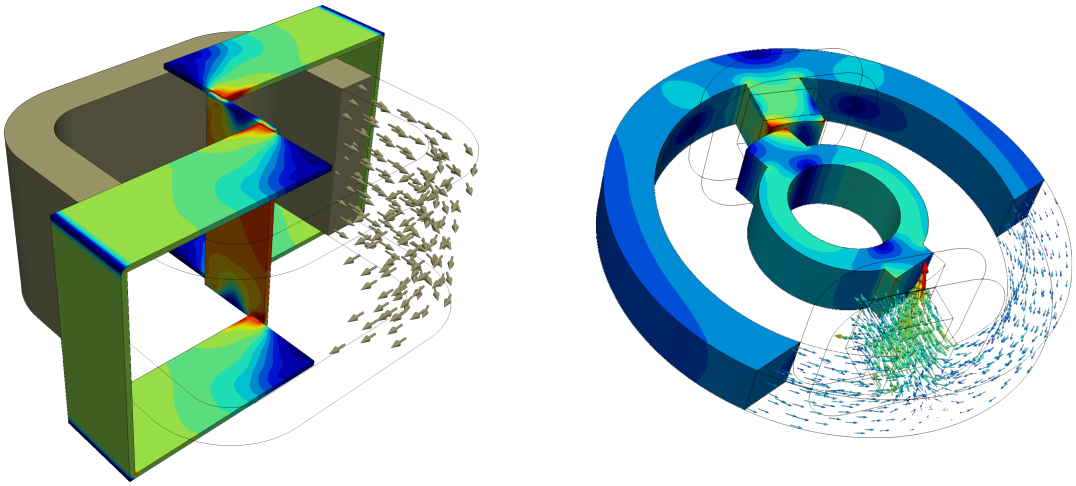


Figure 1. The International Compumag Society's TEAM 13 (left) and TEAM 24 (right) benchmark examples were simulated with a coupled FEM-BEM approach using the Fast Multipole Method

To this end, the Fast Multipole Method (FMM) [3] has been chosen. The FMM is based on a hierarchical decomposition of the computational domain and an approximation scheme for distant interactions between clusters of particles or degrees of freedom. A careful implementation of this method enables to handle BEM-based systems with linear computational complexity.

The FMM is inherently based on octrees, and therefore its parallelisation is non-trivial. In order to address most modern computer architectures, a hybrid strategy is aimed at in which shared-memory (multi-threading) and distributed-memory paradigms are combined. Within this project, the implementation's performance for both strategies are analysed and improved significantly with the goal of achieving a high scalability of the algorithm.

The objectives of the project can be summarized as follows:

- Implementation, analysis and improvement of the FMM algorithm for the solution of N-body problems on distributed memory systems using the Message Passing Interface (MPI).
- Adaptation of the algorithms for the solution of electrostatic problems with the BEM.
- Software profiling and elimination of bottlenecks with the help of HPC tools and experts.
- Performance analysis and benchmarks in order to achieve scaling with respect to the number of processes.

2. Results

The test examples that are used here are of academic nature and provide a basis to demonstrate the scalability of the implementation. The scaling or parallel speedup of an application is defined as

$$S(P,N) = \frac{P_i T(P_i,N)}{T(P,N)}, \tag{2}$$

where P denotes the number of processes, N the problem size and $T(P,N)$ is the computation time required for the potential evaluation given P and N. P_i refers to an initial number of processes and we typically have $P_i = 1$. In cases where $T(1,N)$ is not feasible because of limited computing resources, P_i is chosen as the smallest number of processes for which the computation can be carried out. For the analysis of *strong scaling*, the size N is held constant and P is varied.

Depending on the problem size N and the geometry, two parameters must be specified by the user for FMM: The minimal leaf size ℓ_{min}, which indicates the number of particles or elements to be collected in a leaf box of the octree, and the expansion order p, which determines the quality of the multipole approximation for far-field interactions.

All the calculations presented here were done on the Vienna Scientific Cluster (VSC) [4]. In 2021, the VSC-4 was the the newest cluster of the VSC family and was the most powerful supercomputer in Austria, reaching a performance (Rmax) of 2.7 PFlop/s. The VSC-4 consists of 790 water cooled nodes (Lenovo SD650), each with two Intel Skylake Platinum 8174 processors with 24 cores, interconnected with 100 Gbit/s OmniPath.

2.1. Particle Simulation

As mentioned previously, the scalability of the implemented parallel FMM is first examined by means of the N-body problem (1). Therefore, random points are located in a three-dimensional cube and uniformly distributed across the processes. Each point is assigned a random point charge q_i. Figure 2 shows an example of such a point cloud.

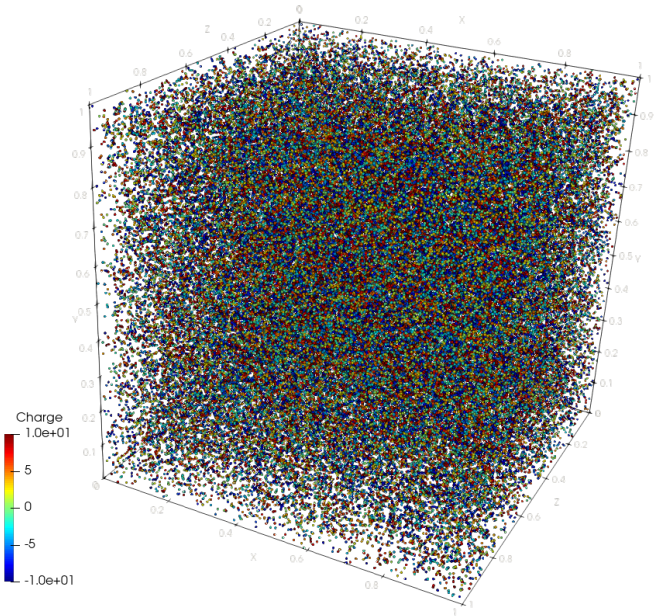


Figure 2. Randomly charged point cloud in a unit cube

In the study, the potential (1) at each particle is evaluated on up to $P = 1008$ processes for $N = 10^7$ particles and on up to $P = 3072$ processes for $N = 10^8$ particles. The minimal leaf size $\ell_{\min}$ is carefully adjusted with respect to the total number of particles or elements. In order to study the influence of the approximation order, it is varied between $p = 8$ and $p = 20$.

Ideally, one has $P_i = 1$ in (2), but that was not feasible for the considered problem sizes. Therefore, the speedup $S(P,N)$ is calculated using an initial number of $P_i = 12$ or $P_i = 48$ processes. The results in Figure 3 reveal an almost perfect strong scaling with respect to the number of processes which is almost independent of the expansion order p. Note, that the specific choice of $P_i > 1$ can lead to the artefact that $S(P,N) > P$ and does not imply super-linear scaling.

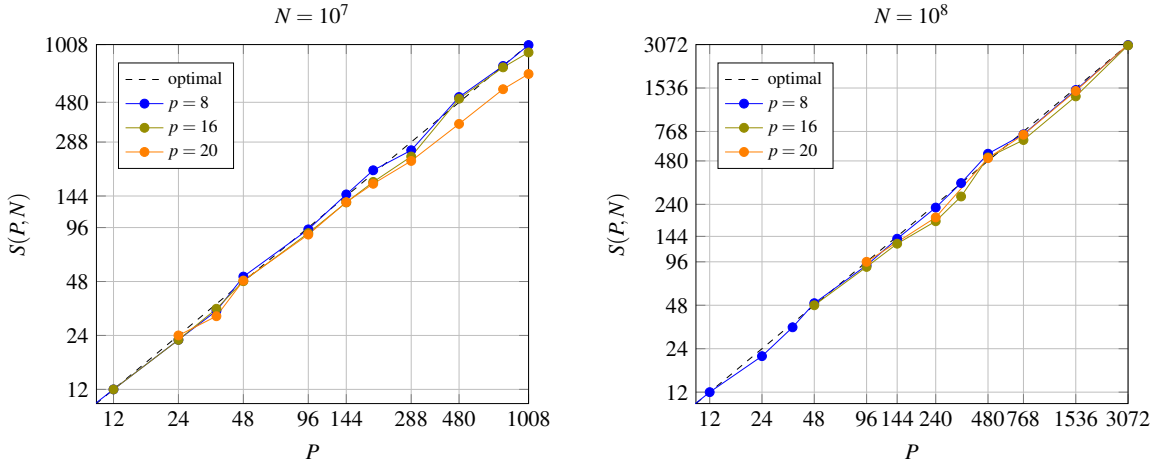


Figure 3. Parallel speedup of the evaluation of the potential with $N = 10^7$ (left) and $N = 10^8$ (right) particles for different FMM expansion degrees p.

2.2. Boundary Element Method

For the parallel FMM-BEM, the complexity of the test is increased: On the one hand, the data management is more involved, e.g. the communication of the partitioned surface grid or the management of degrees of freedom that are attached to the surface elements. On the other hand, a system of equations is created where the nearfield interaction are represented by a distributed sparse matrix. The system is solved iteratively by means of a Conjugate Gradient method. Additionally, a preconditioner is required to keep the number of iterations small. The particular choice of the preconditioner is problem dependent. In this work, a multilevel preconditioning scheme is chosen that operates with the same parallel data structures.

For the benchmark a cube is used as geometry. For the discretisation with the BEM, only surface elements are required and we use linear triangles. The approximation order is $p = 7$ and the minimal leaf size $\ell_{\min}$ has been adapted to the number of elements belonging to each process. Here, we use the fundamental solution $U(x,x^*)$ as given surface potential with a source point x^* outside of the domain. This function fulfils the underlying partial differential equation and allows for an error analysis, because now the calculated charge density is known analytically.

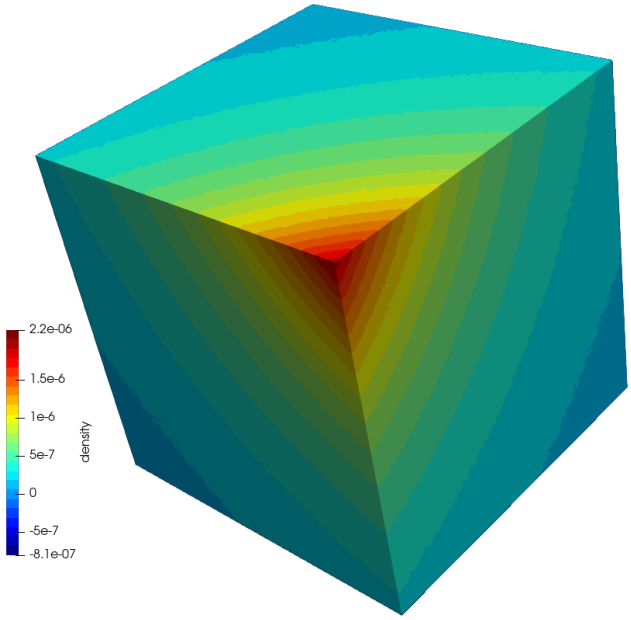
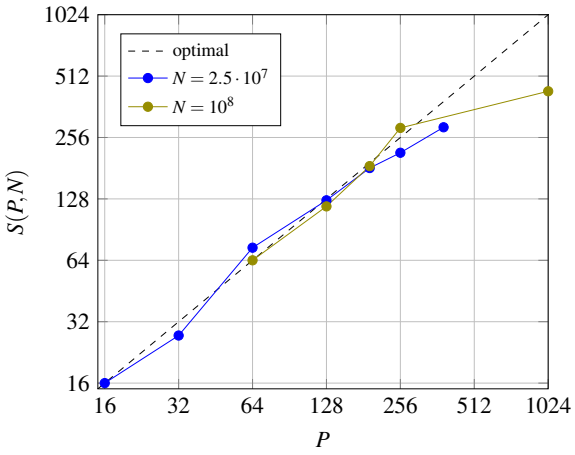


Figure 4. Unit cube and its calculated potential caused by a given fundamental solution

Up to $P = 1024$ processes for $N = 2.5 \cdot 10^7$ and $N = 10^8$ elements are used in this study. The initial evaluation times $T(P_i,N)$ are evaluated for $P_i = 16$ and $P_i = 64$, respectively. The results are shown in Figure 5. The scaling starts to break down if the problem size falls below $\approx 10^5$ elements per process which leaves room for further optimisation. The right table in Figure 5 shows the absolute time required for the setup of the system of equations and its solution. The Conjugate Gradient solver converged within 20 iterations with a relative residual error of $\varepsilon_r = 10^{-5}$.



N	P	T_{setup}	T_{solve}
$2.5 \cdot 10^7$	128	521.18	97.9
	256	271.9	59.1
10^8	256	1065.8	215.5
	1024	271.0	142.3

Figure 5. Parallel speedup for one solver iteration on a unit cube discretised with $N = 2.5 \cdot 10^7$ and $N = 10^8$ boundary elements (left). Measured absolute time (in seconds) to set up the system of equations (T_{setup}) and overall time to solve the system (T_{solve}) (right)

3. Methods

Prior to this project, first steps with respect to distributed memory parallelisation and MPI have only been made on TAILSIT's own 16-core workstation. For this reason, optimisations of the code with respect to shared-memory performance did not have a priority. In general, multi-threading is implemented in the software using C++'s standard `thread` library and is mainly used for the most costly FMM operations. For linear algebra operations the software utilizes standard `BLAS` routines. With this approach the algorithm shows an almost perfect scaling for up to 8 processes on the in-house machine. The use of the `ARM Forge` profiler, provided by VSC, revealed a significant cost bottleneck in the setup of the sparse matrices for the BEM simulation: the initial implementation would simply collect triplets (row and column IDs plus matrix entry) and then sort them in the assembly phase. This sorting became prohibitively slow with large system sizes and we changed the implementation by establishing first the sparsity pattern of the matrix. Further single-node studies have been delayed and the focus was shifted to the distributed parallelism.

The inter-process communication is carried out by resorting to `MPI` routines and their implementation in the FMM algorithm. With the aim to keep the code as generic as possible, we refrained from breaking down the *all-to-all* communications to individual routines tailored to the machine's specific network pattern (e.g. hypercube). Apart from the initial setup phase in which a well-balanced distribution of degrees of freedom to the processes is sought, all later communications effectively run through the `MPI_alltoallv` routine. The studies carried out in this project did not reveal noticeable communication bottlenecks for the tested combinations of problem geometry, size and number of processes. As the main target applications of the code will be on machines with less than 1K processes, we conclude that future improvements are more likely achieved by optimisations in single-node runs with respect to multi-threading.

4. Conclusion

During this SHAPE project TAILSIT's electromagnetic simulation software has been ported to HPC machines using the support and the resources of the VSC. At the regular, sometimes weekly meetings, benchmarks and suggestions for improvement were discussed and, based on this, further measures were set. Thorough performance analyses have led to significant runtime improvements and tests revealed a good overall scaling up to approximately 3000 CPU cores for particle simulations and 400 cores for the BEM. With the improved implementation, up to 10^8 degrees of freedom can be processed within minutes. For TAILSIT, the outcome of this project serves as a basis for future developments of their FEM/BEM-coupling algorithm.

References

[1] L. Kielhorn, T. Rüberg, and J. Zechner. Simulation of electrical machines: a FEM-BEM coupling scheme. *COMPEL-The international journal for computation and mathematics in electrical and electronic engineering*, 2017.

[2] T. Rüberg, L. Kielhorn, and J. Zechner. Electromagnetic devices with moving parts—simulation with FEM/BEM coupling. *Mathematics*, 9(15):1804, 2021.

[3] L. Greengard and V. Rokhlin. A new version of the fast multipole method for the Laplace equation in three dimensions. *Acta Numerica*, 6:229–269, 1997.

[4] Vienna Scientific Cluster (https://vsc.ac.at) .

Acknowledgements

This work was financially supported by the PRACE-6IP project funded in part by the EU's Horizon 2020 Research and Innovation programme (2014-2020) under grant agreement 823767.

Computing cycles for this project were provided by the Vienna Scientific Cluster (https://vsc.ac.at).