Commun. Comput. Phys.
doi: 10.4208/cicp.OA-2020-0072

Vol. **30**, No. 4, pp. 959-984
October 2021

Distributed and Adaptive Fast Multipole Method in Three Dimensions

Jonathan Bull[1] and Stefan Engblom[1,*,‡]

[1] *Division of Scientific Computing, Department of Information Technology, Uppsala University, SE-751 05 Uppsala, Sweden.*

Received 27 April 2020; Accepted (in revised version) 29 March 2021

Abstract. We develop a general distributed implementation of an adaptive fast multipole method in three space dimensions. We rely on a balanced type of adaptive space discretization which supports a highly transparent and fully distributed implementation. A complexity analysis indicates favorable scaling properties and numerical experiments on up to 512 cores and 1 billion source points verify them. The parameters controlling the algorithm are subject to in-depth experiments and the performance response to the input parameters implies that the overall implementation is well-suited to automated tuning.

AMS subject classifications: 65Y05, 68W10, 65Y10, 65Y20, 68W15

Key words: Adaptive fast multipole method, distributed parallelization, Message Passing Interface (MPI), multipole acceptance criterion, balanced tree.

---

1 Introduction

The N-body problem is one of the most fundamental problems in computational physics and it has attracted considerable interest from researchers in numerical algorithms. For the computation of all pairwise particle interactions, the computational complexity of the immediate double for-loop algorithm scales as $\mathcal{O}(N^2)$. For large enough N and high enough tolerance requirements, it is known that the Fast Multipole Method (FMM) [11] stands out as an optimal algorithm of $\mathcal{O}(N)$ complexity, and which also achieves this optimal performance in practice. There is a steadily growing body of research into the use of FMM for the solution of integral equations and PDEs [1,6,9,14,17,31,32].

*Corresponding author. *Email addresses:* `jonathan.bull84@gmail.com` (J. Bull), `stefane@it.uu.se` (S. Engblom)
‡URL: `http://user.it.uu.se/~stefane`

FMMs offer a linear complexity in N and enjoy sharp *a priori* error estimates, but due to their tree-based nature they are also notoriously hard to implement, particularly so in three spatial dimensions [4]. With modern multicore and distributed computer systems, there is an increased interest in designing effective parallelization strategies [2,7,12,13,15–17,27,28]. Balancing efficiency with implementation transparency is here an important aspect [20,24].

When source points are non-uniformly distributed it becomes necessary to adapt the FMM tree according to the local point density such that the computational effort is balanced across the tree. Mesh adaptivity, while solved in theory already in the early ages of the FMM [26], becomes a major issue when confronted with parallelism due to the associated complicated memory access patterns. Early attempts to mitigate this through post-balancing algorithms [22] are less attractive for complexity reasons, and when data-parallel accelerators made their debut a decade ago, it was in fact suggested that non-adaptive FMMs offer a better performance [12].

In this paper we describe the development of a variant of the adaptive FMM, the balanced tree FMM [8,10,13], in three dimensions and using distributed parallelization on very large modern computer systems. Rather than adapting the number of levels locally as in the classic adaptive FMM, our balanced tree FMM maintains a fixed number of levels and splits boxes at median planes such that the number of points in each subtree is balanced at every level. Otherwise known as orthogonal recursive bisection (ORB) [21,25,29], it is guaranteed to produce a balanced tree [29].

The implementation of the balanced tree FMM with distributed parallelism is relatively straightforward and is shown to scale well up to 512 processes at a respectable absolute efficiency. Importantly, and indeed one of our main messages of the paper, the implementation clearly exposes the various algorithmic parameters to the user thus enabling the performance to be readily optimized for a particular simulation. We show that the performance response is dominantly convex in the parameter space making the balanced tree FMM particularly well-suited to automatic tuning in an online computing context, hence opening up for techniques previously demonstrated in 2D [13].

Since the FMM has been judged to be one of the top 10 most important algorithms of the 20th century [5], it is our hope that insights obtained here is of general value. Therefore, as others have also done previously [17], our implementation of the parallel balanced tree FMM is freely available as `daFMM3D` (distributed adaptive FMM in 3D) C/C++/Matlab code at `www.stenglib.org`.

The structure of the paper is as follows: in Section 2 we describe the balanced tree FMM in three dimensions including the multipole acceptance criterion, adaptive box splitting and computational complexity estimates. The corresponding distributed algorithm is also presented with a description of parallel data structures and communication complexity. In Section 3 convergence test results, parameter response test results and strong scaling results on up to 512 processes are reported. The algorithm's adaptive response is also tested in the more challenging case of a spiral galaxy of 1 billion sources with a multiscale structure.