

A fast multi-resolution lattice Green's function method for elliptic difference equations

Benedikt Dorschner^{a,*}, Ke Yu^a, Gianmarco Mengaldo^a, Tim Colonius^a

^aCalifornia Institute of Technology, 1200 E California Blvd, CA 91125, Pasadena, U.S.A.

ARTICLE INFO

Article history:

Elliptic difference equation, Poisson problem, Lattice Green's function, Fast multipole method, Mesh refinement

ABSTRACT

We propose a mesh refinement technique for solving elliptic difference equations on unbounded domains based on the fast lattice Greens function (FLGF) method. The FLGF method exploits the regularity of the Cartesian mesh and uses the fast multipole method in conjunction with fast Fourier transforms to yield linear complexity and decrease time-to-solution. We extend this method to a multi-resolution scheme and allow for locally refined Cartesian blocks embedded in the computational domain. Appropriately chosen interpolation and regularization operators retain consistency between the discrete Laplace operator and its inverse on the unbounded domain. Second-order accuracy and linear complexity are maintained, while significantly reducing the number of degrees of freedom and hence the computational cost.

© 2019 Elsevier Inc. All rights reserved.

1. Introduction

Elliptic partial differential equations require fast numerical solvers scalable up to trillions of unknowns. In this contribution, we will focus on elliptic difference equations on unbounded domains arising in numerous applications including the incompressible Navier-Stokes equations (NSE), quantum mechanics, random walks, and plasma physics. [1, 2, 3]. In particular, we will use the Poisson equation as a representative example of these. Common, scalable methods to solve these equations include the fast Fourier transform (FFT) [4], the fast multipole method (FMM) [5], multigrid methods and its variants (algebraic (AMG) and geometric (GMG)) [6], domain decomposition methods [7] as well as wavelet transforms [8]. A recent comparison of FFT, FMM as well as AMG and MGM can be found in [9], where the Poisson equation was solved in the unit cube with periodic boundary conditions and scaled up to 600

*Corresponding author
e-mail: bdorschn@caltech.edu (Benedikt Dorschner)

billion degrees of freedom (DoF) and 220,379 cores. The authors conclude that while FFT is the method of choice for smooth source functions that require uniform resolution, it is outperformed by the FMM for localized source distributions where one can take advantage of nonuniform grids. In addition, state-of-the-art algebraic multigrid solvers were found to be an order of magnitude slower than FFT, GMG or FMM. Note that while the authors compare these methods for periodic boundary conditions only, we focus on free-space boundary conditions in the following. In this case, it is expected that the efficiency between both FFT and multigrid method will decrease due to the difficulties of satisfying free-space boundary with the appropriate Dirichlet boundary conditions or zero-padding, respectively.

A fast lattice Green's function method (FLGF) has recently been proposed in [10] and shown extraordinary efficiency in solving the class of elliptic difference equations in unbounded domains. The FLGF method is based on an efficient convolution of the source distribution and its fundamental solution via FMM, the Green's function. In contrast to most existing FMM approaches, the FLGF method exploits the regularity of a Cartesian mesh, and uses a kernel-independent, interpolation-based fast multipole method (FMM) to retain linear computational complexity inherent to FMM. A similar approach has been proposed in [11] for two dimensional infinite lattices using skeleton points (see also [12, 13]). Additionally, the FLGF exploits fast Fourier transforms (FFT) to further reduce the computational costs compared to conventional solvers. It has been successfully applied to solve the incompressible Navier-Stokes equations using a finite volume approach [14]. In addition, accurate simulations of external aerodynamics of complex geometries were enabled by coupling it with the immersed boundary method (IB-LGF)[15]. Owing to the geometrical flexibility of the FMM and the implied free-space boundary of the Lattice Green's functions (LGF), the solver allows for adaptive, possibly disjoint meshes that limit the computational domain to a set of blocks with non-negligible vorticity only, with further computational savings associated with the typical compactness of this region. This is in contrast to most common approaches, which employ spatially truncated domains with approximate free-space boundaries. These approximations introduce blockage errors, which affect accuracy and may even change the dynamics of the flow [16, 17, 18]. Thus, large computational domains in combination with stretched grids [19, 20, 21], local refinement [22, 23] and far-field approximations [24] are required to limit the influence of the approximate free-space boundary condition.

A significant limitation of the FLGF approach, however, is uniform resolution. This limits its ability to reach competitive time-to-solution requirements, commonly required by demanding academic or industrial applications. In fact, while uniform Cartesian meshes can significantly decrease the cost per degree of freedom (DoF), the total number of DoF can be prohibitive for strongly anisotropic or inhomogeneous problems with localized source regions. This issue is particularly prominent for, e.g., high Reynolds number flows or problems where the range of scales is large. The use of uniform meshes for this set of problems requires a resolution that is comparable to the smallest scales in the problem, although these might be confined to a very small region of the computational domain, thereby yielding an unnecessarily high number of DoF. To overcome this limitation, we propose here a multi-resolution extension of the FLGF method. Most popular multi-resolution schemes include multigrid methods as well as FMM, which inherently support local mesh refinement and have been successfully implemented in a variety of publicly available software packages (see, e.g.: [25, 26, 27] or [28] for a review). However, our goal here is to retain the favorable efficiency of

the FLGF method compared to the classical FMM as well as multigrid methods but reduce the number of DoFs by a proposing block-structured mesh refinement algorithm for the FLGF method.

The paper is organized as follows: In section 2, we start by reviewing the fast lattice Green's function method in some detail to facilitate further discussion. Subsequently, the proposed multi-resolution scheme is presented in section 3. In addition, we provide an analysis, discussion and a convergence study of the refinement scheme on a fluid dynamics test problem.

2. Fast lattice Green's function method

The lattice Green's function method [10] solves inhomogeneous, linear, constant-coefficient difference equations, defined on unbounded Cartesian grids, by convolution of the fundamental solution of the discrete operator, so-called lattice Green's functions (LGF), with the equation's source term. LGFs can be obtained from Fourier integrals, and its asymptotic expansions can be used to facilitate numerical or analytical evaluation. The formally infinite mesh may however be truncated such that only cells with non-negligible source are retained. Going by example, for block-structured meshes this allows us to adapt the computational domain by simply adding or removing the corresponding blocks.

Here, we will focus on elliptic difference equations on unbounded domains, exemplified by the three dimensional Poisson's problem, which reads

$$\Delta u(\mathbf{x}) = f(\mathbf{x}), \quad \text{supp}(f) \subseteq \Omega, \quad (1)$$

where $\mathbf{x} \in \mathbb{R}^3$ and Ω denotes a bounded domain in \mathbb{R}^3 . Its solution or target field u can be obtained by convolution of the fundamental solution of the Laplace operator $G(\mathbf{x}) = -1/(4\pi|\mathbf{x}|)$ with the source field $f(\mathbf{x})$ such that

$$u(\mathbf{x}) = (G * f)(\mathbf{x}) = \int_{\Omega} G(\mathbf{x} - \boldsymbol{\xi}) f(\boldsymbol{\xi}) d\boldsymbol{\xi}. \quad (2)$$

Correspondingly, the discrete scalar Poisson equation reads

$$L_Q u(\mathbf{n}) = f(\mathbf{n}), \quad \text{supp}(f) \subseteq \Omega_h, \quad (3)$$

where $u, f \in \mathbb{R}^Q$, Ω_h is a bounded domain in \mathbb{Z}^3 , $\mathbf{n} = (n_1, n_2, n_3) \in \mathbb{Z}^3$ and $Q \in \{C\}$ denotes cell-centered values. Its solution is given by the discrete convolution

$$u(\mathbf{n}) = (G * f)(\mathbf{n}) = \sum_{\mathbf{m} \in \Omega_h} G(\mathbf{n} - \mathbf{m}) f(\mathbf{m}), \quad (4)$$

where the fundamental solution or LGF of the discrete 7-pt Laplacian is denoted by G . An expression for $G(\mathbf{n})$ can be obtained by diagonalizing the Laplace operator L_Q in Fourier space, inversion and a back-transform (see, e.g., [29, 30]). In terms of Fourier integrals this eventually yields

$$G(\mathbf{n}) = \frac{1}{8\pi^3} \int_{[-\pi, \pi]^3} \frac{\exp(-in\xi)}{2 \cos(\xi_1) + 2 \cos(\xi_2) + 2 \cos(\xi_3) - 6} d\xi. \quad (5)$$

For evaluation of the LGF it is typically more convenient to rewrite Eq.(5) as a one dimensional, semi-infinite integral as

$$G(\mathbf{n}) = - \int_0^{\infty} \exp(-6t) I_{n_1}(2t) I_{n_2}(2t) I_{n_3}(2t) dt, \quad (6)$$

where $I_k(t)$ denotes the modified Bessel function of first kind and order k . While Eq.(6) is readily evaluated using an adaptive Gauss-Kronrod quadrature or alike, it is more efficient to evaluate the Green's function through its asymptotic expansion in the far-field [31, 13, 32], i.e., large $|\mathbf{n}|$. In particular, the target field u can then be written as

$$u(\mathbf{n}) = u^{\text{near}}(\mathbf{n}) + u^{\text{far}}(\mathbf{n}) + \varepsilon(\mathbf{n}), \quad (7)$$

where

$$u^{\text{near}}(\mathbf{n}) = \sum_{\mathbf{m} \in \Omega_h^{\text{near}}(\mathbf{n})} G(\mathbf{n} - \mathbf{m}) f(\mathbf{m}) \quad (8)$$

$$u^{\text{far}}(\mathbf{n}) = \sum_{\mathbf{m} \in \Omega_h \setminus \Omega_h^{\text{near}}(\mathbf{n})} A_G^q(\mathbf{n} - \mathbf{m}) f(\mathbf{m}), \quad (9)$$

and Ω_h^{near} , $\varepsilon(\mathbf{n})$ are the near field and the error due to approximating $G(\mathbf{n})$ with $A_G^q(\mathbf{n})$ in the far field, respectively. The q -term asymptotic expansion of $G(\mathbf{n})$ is defined such that $G(\mathbf{n}) = A_G^q(\mathbf{n}) + \mathcal{O}(|\mathbf{n}|^{-2q-1})$. For $q = 2$ and $q = 3$ it reads

$$A_G^2(\mathbf{x}) = -\frac{1}{4\pi|\mathbf{x}|} - \frac{x_1^4 + x_2^4 + x_3^4 - 3x_1^2x_2^2 - 3x_1^2x_3^2 - 3x_2^2x_3^2}{16\pi|\mathbf{x}|^7}. \quad (10)$$

and

$$A_G^3(\mathbf{x}) = A_G^2(\mathbf{x}) + \frac{1}{128\pi|\mathbf{x}|^{13}} \left(-228 \left(x_2^2x_3^2x_1^4 + x_2^2x_3^4x_1^2 + x_2^4x_3^2x_1^2 \right) + 621 \left(x_1^4x_2^4 + x_3^4x_2^4 + x_1^4x_3^4 \right) - \right. \\ \left. 244 \left(x_2^2x_1^6 + x_3^2x_1^6 + x_2^6x_1^2 + x_3^6x_1^2 + x_2^2x_3^6 + x_2^6x_3^2 \right) + 23 \left(x_1^8 + x_2^8 + x_3^8 \right) \right), \quad (11)$$

respectively. In our implementation, we use a tabulated integration of Eq.(6) in the near field for $|\mathbf{n}| \leq 100$. The asymptotic expansion is chosen to ensure a conservative error bound of the asymptotic expansion compared to the direct integration of $|\varepsilon| \lesssim 10^{-12}$, while keeping the number of terms to a minimum. Hence, $q = 3$ is used for $100 < |\mathbf{n}| \leq 600$, whereas $q = 2$ suffices for $|\mathbf{n}| > 600$.

2.1. Fast convolutions

The direct evaluation of the discrete convolution in Eq.(4) is prohibitive for large computational domains as it requires $\mathcal{O}(N^2)$ work for N degrees of freedom. The FLGF method [10] on the other hand employs a kernel-independent interpolation-based fast multipole method (FMM) to compute the discrete convolutions in conjunction with block-wise FFT convolution. The FMM achieves linear complexity $\mathcal{O}(N)$ by exploiting the fact that, for an elliptic kernel, the solution in the far-field is much smoother than in the near-field. Thus, a low-rank representation of the kernel is sufficient to accurately compute the contribution of far-field, while only the near field requires full-rank representation of the kernel. A low-rank approximation of a kernel $K(\mathbf{x}, \mathbf{y})$ can be obtained by interpolation using coarse grained sampling points $\mathbf{x}_0, \dots, \mathbf{x}_{n-1}$. With the generic interpolation function $\phi(\mathbf{x})$, this yields

$$\tilde{K}(\mathbf{x}, \mathbf{y}) = \sum_{i=0}^{n-1} \sum_{j=0}^{n-1} \phi(\mathbf{x}) K(\mathbf{x}_i, \mathbf{y}_j) \phi(\mathbf{y}), \quad (12)$$

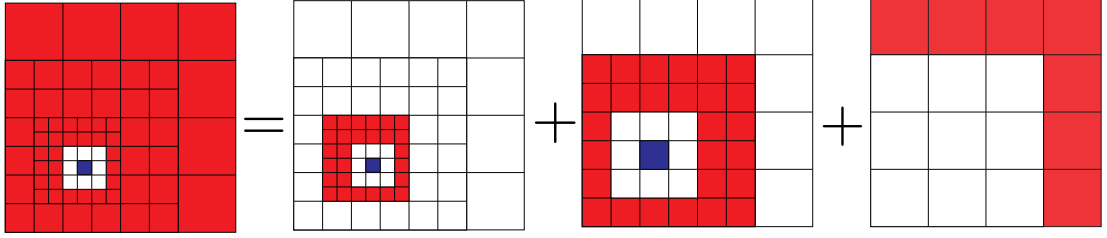


Fig. 1: Schematic of the hierarchical domain composition of the far-field (red, left) for an octant $\mathcal{B}_i^{L_B-1}$ (blue, left). While the near field consist of the nearest neighbors only, the far-field is composed out of the set of influence lists for all levels. For level l the influence list contains the children of the nearest neighbors of \mathcal{B}_i^l 's parent, which are not contained in the near field, i.e., are well separated.

and the discrete convolution can then be approximated by

$$u(\mathbf{x}_i) \approx \sum_{j=0}^{M-1} \tilde{K}(\mathbf{x}_i, \mathbf{y}_j) f(\mathbf{y}_j) = \sum_{j=0}^{M-1} \sum_{p=0}^{n-1} \sum_{q=0}^{n-1} \phi(\mathbf{x}_i) K(\mathbf{x}_p, \mathbf{y}_q) \phi(\mathbf{y}_j) f(\mathbf{y}_j) \quad i = 0, \dots, N-1, \quad (13)$$

where N is the number of target points and M the number of source points. The near- and far-field contributions are treated by constructing a hierarchical decomposition of the domain using an octree structure \mathcal{T} (quadtree in two dimensions), for which Eq.(13) is evaluated recursively. The octree is defined to have a depth L_B , where the tree root is assumed to have level 0 and the base level $L_B - 1$ corresponds to physical domain. The tree nodes are also referred to as octants and octants without children are leaf nodes. The set of leafs on level l is indicated by $\mathcal{B}_{\text{Leaf}}^l$. A distinct feature of the FLGF as proposed in [10] is that each tree node corresponds to a region, which is represented by a Cartesian block and contains $N_b = n_b^3$ cells. We further denote the i -th octant or block at level l by \mathcal{B}_i^l and the set of all octants at level l by $\mathcal{B}^l = \bigcup_{i=0}^{N_B^l} \mathcal{B}_i^l$, where N_B^l is the number of octants on level l . The set of children and the parents are denoted by $\mathcal{C}(\mathcal{B}_i^l)$ and $\mathcal{P}(\mathcal{B}_i^l)$, respectively.

Thus, for a given target field $u_i^{L_B-1}$, defined on the octant $\mathcal{B}_i^{L_B-1}$ on level $L_B - 1$, the near-field contribution consists of the interaction, i.e., convolution, with region $\mathcal{N}(\mathcal{B}_i^{L_B-1})$, containing the octant itself and the nearest neighbor octants on the finest tree level $L_B - 1$. The far-field contributions are then evaluated recursively for the levels $l = L_B - 1, \dots, 0$ and are defined as the convolution with octants in the influence region $\mathcal{I}(\mathcal{B}_i^l) = \{\hat{\mathcal{B}}_i^l \in \mathcal{F}(\mathcal{B}_i^l) \setminus \mathcal{F}(\mathcal{B}_i^{l-1})\}$, which includes the well-separated octants, i.e. $\mathcal{F}(\mathcal{B}_i^l) = \bigcup_{i=0}^{L_B-1} \mathcal{B}^l \setminus \mathcal{N}(\mathcal{B}_i^l)$, but excludes the regions well-separated from its parents $\mathcal{F}(\mathcal{P}(\mathcal{B}_i^l))$. Schematically, the domain decomposition in near and far-field regions is depicted in figure 1.

Note that using Cartesian blocks as octants allows for the convolution between each block and its influence list to be computed by a block-wise FFT-based convolution. There, the convolution is converted to a complex Hadamard product in Fourier space. As this is a circular convolution, the original block needs to be zero-padded. FFT-based convolution is a standard technique and we refrain from reporting details in the interest of brevity but refer the reader to [10]. Compared to a direct summation as indicated in Eq.(4) the block-wise FFT decreases computational complexity from $\mathcal{O}(N_b^2)$ to $\mathcal{O}(N_b \log N_b)$ for each block-convolution. It should be clear from the above that given a union of source $\mathcal{B}_s = \bigcup_{i=0}^{N_s-1} \mathcal{B}_{s,i}$ and target blocks $\mathcal{B}_t = \bigcup_{i=0}^{N_t-1} \mathcal{B}_{t,i}$ the convolution can be evaluated as the sum of the individual

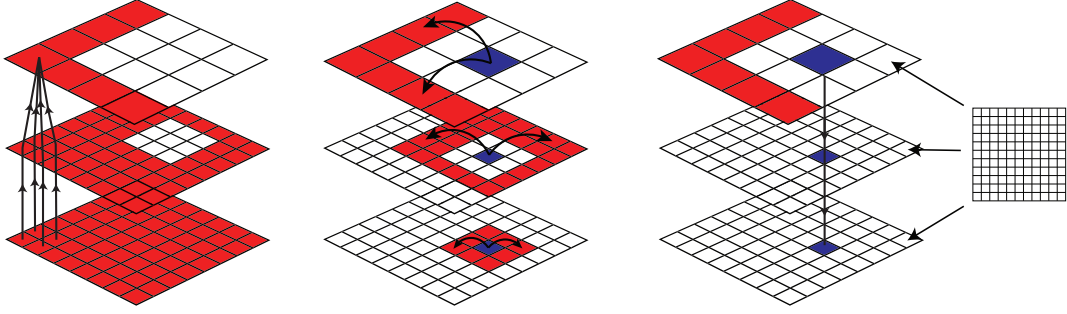


Fig. 2: Schematic of the fast multipole method: Left: *Upward pass* - Source regularization . Middle: *Level interaction* - Convolution of a block (blue) with its influence list (red). Right: *Downward pass* - Compute and accumulate the induced fields at the interpolation nodes. Note that each FMM cell, corresponds to a Cartesian block in the FLGF.

convolutions as

$$u_i = \sum_{j \in \mathcal{B}_s} \text{conv}(G_{j-i}, f_j), \quad \text{for } i = 0, \dots, N_t - 1, \quad (14)$$

where the convolution operator is denoted as conv and G_{j-i} is the vector containing the unique values of $G(\mathbf{m} - \mathbf{n})$ evaluated on the grid points \mathbf{m} and \mathbf{n} of the blocks $\mathcal{B}_{t,j}$ and $\mathcal{B}_{s,i}$, respectively. In the case of FLGF this convolution is evaluated using FFT. The additional advantage of such a block-structured FMM approach is that regions with negligible source can be removed entirely, yielding an adaptive and possibly disjoint mesh.

With these definitions and the corresponding tree structure, the evaluation of Eq.(13) can be split into three consecutive steps. First, we iterate in bottom-up order through the tree and compute the effective source terms on each level. This is called the *upward pass* in FMM literature and computes $\tilde{f}(\mathbf{y}_q) = \sum_{j=0}^{M-1} \phi(\mathbf{y}_j) f(\mathbf{y}_j)$ from Eq.(13). Second, for each level in the tree the so-called *level interaction* is computed, where the convolution of each octant in the level with its influence region is calculated. This corresponds to $\tilde{u}(\mathbf{x}_p) = \sum_{q=0}^{n-1} K(\mathbf{x}_p, \mathbf{y}_q) \tilde{f}(\mathbf{y}_q)$ in Eq.(13). Finally, iterating from the root to the leaf octants, all contributions are interpolated and accumulated on the next level, which accounts for $u(\mathbf{x}_i) = \sum_{p=0}^{n-1} \phi(\mathbf{x}_i) \tilde{u}(\mathbf{x}_p)$ of Eq.(13). This is called the *downward pass*. Schematically, the FLGF method is shown on figure 2 and can be summarized by the following algorithm:

1. *Upward pass*: Compute effective source terms at interpolation nodes

For $l = L_B - 2, \dots, 0$: For $i = 0, \dots, N_B^l$

$$\tilde{f}_i^l = \sum_{j \in \mathcal{C}(\mathcal{B}_i^l)} R^{l+1} \tilde{f}_j, \quad (15)$$

where the regularization operator R^{l+1} is the adjoint of the interpolation operator J^{l+1} (see below).

2. *Level Interaction* : FFT Convolution with the octant in the influence region

For $l = 0, \dots, L_B - 1$: For $i = 0, \dots, N_B^l$

$$\tilde{v}_i^l = \sum_{j \in \mathcal{I}(\mathcal{B}_i^l)} \text{conv}(G_{i-j}, f_j), \quad (16)$$

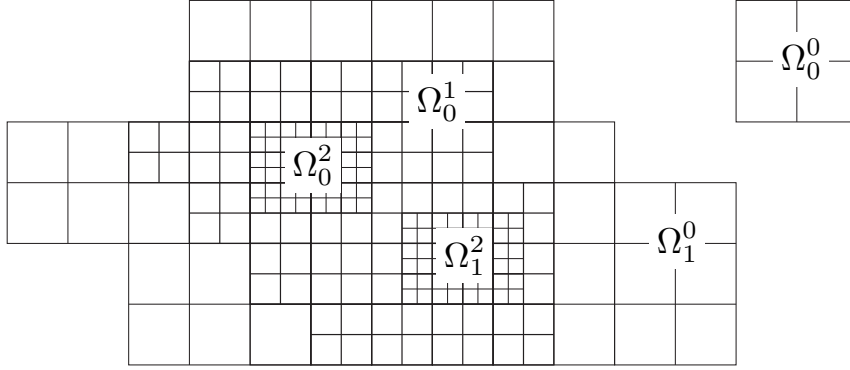


Fig. 3: Example of a block-refined mesh topology with two levels of refinement.

where $\text{conv}(\cdot)$ is the FFT convolution operator.

3. *Downward pass*: Compute and accumulate induced field at interpolation nodes

For $l = 0, \dots, L_B - 1$: For $i = 0, \dots, N_B^l$

$$\tilde{u}_i^l = \tilde{v}_i^l + J_i^{l-1} \tilde{u}_i^{l-1}, \quad (17)$$

where the interpolation operator J^l interpolates from the parent onto the child block.

Owing to the regularity of the Cartesian block mesh, the interpolation operators are implemented using Lagrangian polynomials. As in [10], we typically use $n_l \leq 10$ interpolation nodes to achieve a relative interpolation error of $\epsilon \approx 10^{-12}$ for an analytic function approximation. The regularization operator is given by the adjoint of the interpolation operator and is sometimes called antepolation in FMM literature. In summary the FLGF method combines the fastest methods for regular meshes, while retaining the geometrical flexibility and overall linear complexity inherent to FMM. Excellent computational rates and parallel performance have been reported in [10]. In the following, this methodology will be extended to allow for block-refinement.

3. Block-refined FLGF method

While the FLGF method has shown to be a fast and promising approach for solving the Poisson problem, the methodology is so far limited to block-structured meshes with uniform resolution and the octree is only used to compute the block-structured FMM. Here we propose a multi-resolution, block-refinement strategy and define the computational domain as $\Omega = \bigcup_{l=0}^{L_R-1} \Omega^l = \bigcup_{l=0}^{L_R-1} \bigcup_{m=0}^{M^l} \Omega_m^l$, where Ω_m^l denotes region m of refinement level l and may itself be composed out of N_m^l blocks with $\Omega_m^l = \bigcup_{i=0}^{N_m^l} \mathcal{B}_i^l$ and $\Omega^l = \bigcup_{i=0}^{N_B^l} \mathcal{B}_i^l$ (see figure 3). In this context, the mesh level $l \geq 0$ is defined as physical refinement domain. This is in contrast to the FMM in section 2.1, where the physical domain was defined on the base level $L_B - 1$. This slight abuse of notation will facilitate the discussion below. The refinement method, however can conveniently use the same octree structure as used for the LGF. We will further use the same nomenclature and distinguish between leaf octants and interior, non-leaf octants.

When refining the physical domain by embedding locally refined grid patches within the computational domain, the free-space boundary conditions implied by the lattice Green's functions become problematic since the refinement patches itself do have a well defined boundary condition, which is imposed by the surrounding domain and is not the free space. However, if the source field is projected onto each level within its support by an appropriate coarsening operator, we can, in principle, apply the FLGF method on each level independently. In terms of the octree structure this corresponds to recursive coarsening of the source field on all leaf nodes up to the coarsest refinement level. We further define the convolution of source regions with the LGF such that each target region, which is defined on non-leaf, interior octants only interacts with leaf octants, whereas leaf octants interact with both leaf and non-leaf octants. With a subsequent recursive interpolation procedure of the target fields, the contribution of all octant on all levels to the target region can be accumulated and accounted for. Schematically, this method is depict in figure 4. More concisely, we can summarize the scheme by the following expression for the target field u_m^l

$$u_m^l = \Gamma_m^l \left[\sum_{p < l} J^p G^p f^p + G^l \left(\sum_{p \leq l} \prod_{j > p} C^j f^p - \sum_{p < l} L_Q^{p+1} J^p G^p f^p \right) \right], \quad (18)$$

where $f^l = f^l(\mathbf{x})$, $\forall \mathbf{x} \in \Omega^l$ and the projection operator $\Gamma_m^l f = f_m^l$. In the above, we have also used a shorthand notation for the convolution of the Green's function G^l with a field φ^l on level l , which is given by

$$G^l \varphi^l = G^l(\mathbf{x}, \mathbf{y}) \varphi^l(\mathbf{y}) = \sum_{\mathbf{y} \in \Omega^l} G^l(\mathbf{x} - \mathbf{y}) \varphi^l(\mathbf{y}), \quad \mathbf{x} \in \Omega^l. \quad (19)$$

Further, the interpolation from level l onto level $l + 1$ is denoted by J^l , whereas the coarsening operator C^l projects a field from level l to level $l - 1$. Note that we have included an additional source correction term (last term of Eq. (18)). As will be shown later, this source correction step is not necessary to retain second-order accuracy but is used to ensure consistency between the Laplace operator L_Q and its inverse. While this multi-resolution scheme computes the inverse of the multi-resolution Laplace operator numerically, the corresponding forward Laplace operator is apriori not known. This is due to the fact that the LGF in Eq.(5) corresponds to the inverse of the 7-pt Laplace operator on a uniform mesh with free-space boundaries only. For practical applications however, it is often necessary to apply at least the gradient operator on the target field. Hence, it is desirable to know or at least be able to apply a consistent forward operator and we seek Laplacian free field for each level, where the following equation to holds:

$$L_Q^l \Gamma^l \sum_{p < l} J^l G^l f^l = 0, \quad l = 0, \dots, L_R - 1. \quad (20)$$

The source correction step ensures consistency between the forward operator per level, in this case the 7-pt Laplace, and the computed inverse by numerically evaluating the error, originating from said inconsistency, on the next level and subtracting it from the source on this level. This is indicated in figure 4b for the second and third refinement level with the green region. Naturally, this is only applicable for non-leaf octants.

Using the notation from section 2.1, the algorithm for a target field u_i^l , defined on octant \mathcal{B}_i^l , is given by the following steps:

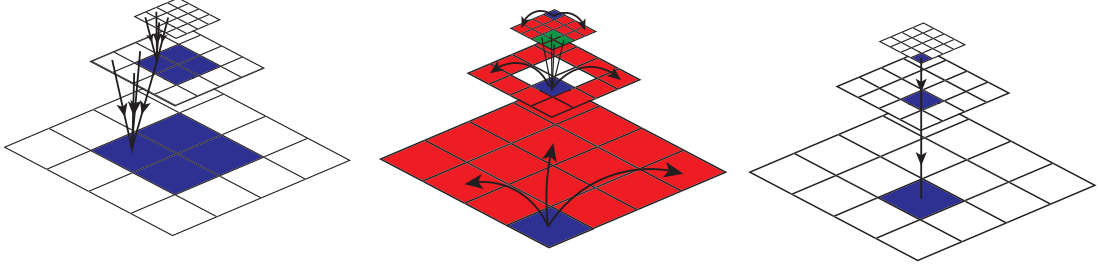


Fig. 4: Schematic of the mesh refinement methodology for the FLGF method. Left: Coarsification of the source field. Middle: Level interaction (blue-red) and source correction (green). Right: Interpolation and accumulation of the target field.

1. Regularization:

For $l = L_R - 2, \dots, 0$: For $i = 0, \dots, N_B^l$

$$\tilde{f}_i^l = \sum_{j \in \mathcal{C}(B_i^l)} C^{l+1} f_j \quad (21)$$

2. Level interaction:

For $l = 0, \dots, L_R - 1$: For $i = 0, \dots, N_B^l$

2..1 Convolution:

$$\tilde{v}_i^l = \begin{cases} \sum_{j \in B_i^l} \text{conv}(G_{i-j}, f_j^l) & \text{if } B_i^l \in B_{\text{Leafs}}^l \\ \sum_{j \in B_i^l} \text{conv}(G_{i-j}, f_j^l) & \text{else.} \end{cases} \quad (22)$$

2..2 Source correction:

$$\tilde{f}_i^{l+1} \leftarrow \tilde{f}_i^{l+1} - L_Q^{l+1} J^l \tilde{f}_i^l \quad (23)$$

3. Accumulation and interpolation:

For $l = 0, \dots, L_R - 1$: For $i = 0, \dots, N_B^l$

$$\tilde{u}_i^l = \tilde{v}_i^l + J^{l-1} \tilde{u}_i^{l-1} \quad (24)$$

Note that the convolution in the second step is computed using the FLGF method as presented in section 2.1. Since, we need to distinguish between the interaction of leaf nodes and the interior octants, we also have to carry out the FMM twice. In practice, we first use the FMM for computing the convolution of the entire refinement tree and a second time using the non-leaf nodes only. Subtracting both is equivalent to Eq.(22) and yields

$$\tilde{v}^l(\mathbf{x}) = G^l(\mathbf{x}, \mathbf{y}) f^l(\mathbf{x}) - G^l(\mathbf{x}, \mathbf{y}_l) f(\mathbf{y}_l) \quad \forall \mathbf{x} \in B^l, \mathbf{y}_l \in B^l \setminus B_{\text{Leafs}}^l. \quad (25)$$

Note that the computational complexity remains linear in the number of DoFs per level and also scales linearly with the number of refinement levels and thus retains linear complexity in the total number of DoFs.

Regularization and Interpolation operators

The FLGF method is a second-order accurate scheme and we thus require the interpolation and regularization operators to be at least second-order accurate to avoid degradation of the overall accuracy of the scheme. The field values are stored at the cell centers, which leads to a staggered storage upon refinement, i.e., none of the field values in any child cell are stored in the same physical location than the field values of the parent cell (see figure 5). This is in contrast to vertex storage, which would lead to field values stored at coinciding locations for different levels. For ease of implementation, we define the interpolation operator to be the second-order Lagrangian polynomials, which already have been used for the FMM. Note that in the procedure above, the interpolant corresponds to the interaction of a non-leaf octant with the leaf octants only. Thus the fields within each non-leaf octant are distinct. We use an additional buffer layer of one cell around each octant and set the source within this region to zero. This allows us to use an unbiased interpolation for each octant.

For the coarsening operator, we use a simple averaging procedure, which is second-order accurate and does not require any neighbor information. In one dimension this yields

$$u(x_i^l) = \frac{u(x_i^{l+1}) + u(x_{i+1}^{l+1})}{2}. \quad (26)$$

3.1. Convergence

For validation of the multi-resolution scheme as presented above, we use the method of manufactured solutions. As a test problem we consider a vortex ring with radius R and its streamfunction Ψ is defined as

$$\Psi(r, z) = f \left(\frac{\sqrt{(r-R)^2 + z^2}}{R} \right) \mathbf{e}_\theta, \quad (27)$$

where

$$f(t) = \begin{cases} c_1 \exp\left(-\frac{c_2}{1-t^2}\right) & \text{if } |t| < 1 \\ 0.0 & \text{else.} \end{cases} \quad (28)$$

The streamfunction is related to vorticity by the Poisson equation $\omega = \Delta\Psi$. This allows us to initialize the source field by analytical evaluation of the Laplace and compare the numerically obtained solution of the streamfunction with the analytical one in Eq.(27). In this setup we place six vortex rings in the unit cube, where a single large vortex ring with $c_1 = 10^3$, $c_2 = 10$ and radius $R = 0.125$ is located in the center of the domain and five smaller ones with $c_1 = 10^6$, $c_2 = 15$ and radius $R = 0.015$ at $z = 0.125$ are arranged as shown in figure 6a.

As a first convergence test, a global mesh refinement study with a fixed mesh topology of three levels near the vortex ring is carried out. An example mesh topology with six levels of refinement is shown in figure 6b. The mesh

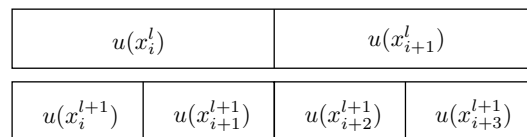


Fig. 5: Mesh refinement in one dimension.

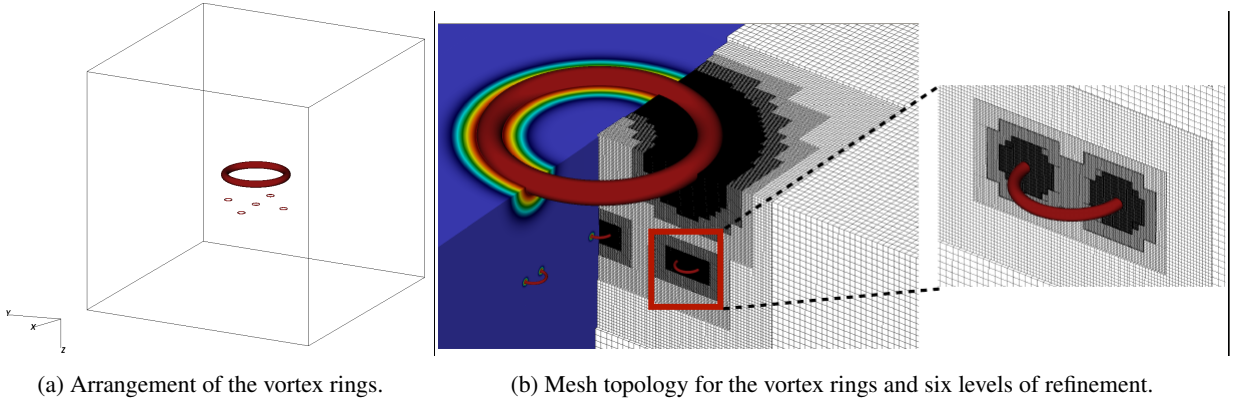


Fig. 6: Mesh topology and vorticity field for the vortex ring and the numerical solution of the streamfunction Ψ_h for six levels of refinement.

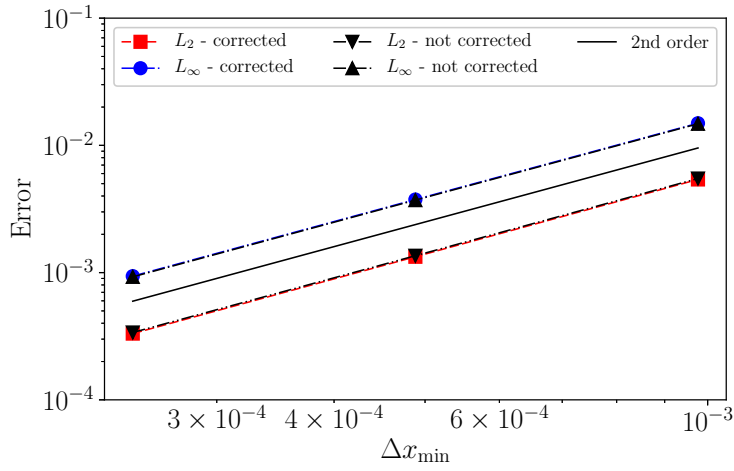


Fig. 7: Convergence of the L_2 and L_∞ norm with respect to the analytical target field Ψ for the block-refined FLGF method for a fixed mesh topology of three refinement levels and an increasing base level resolution Δx_{\max} .

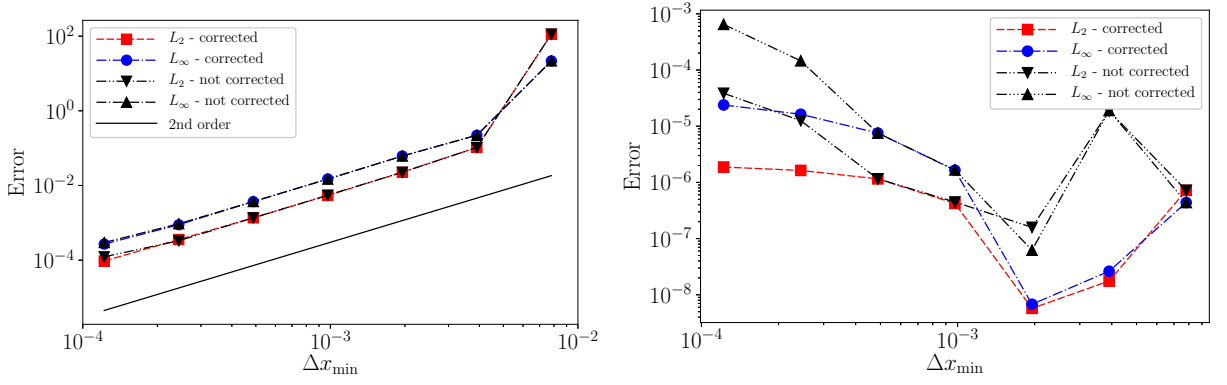
is composed of cubic blocks and we initialize or refine the block if on level l the following criterion is met

$$\omega^l > \alpha^{L_R-l} \omega_{\max}^l, \tag{29}$$

where $\alpha = 1/32$ unless stated otherwise. This criterion is conservative and is offered for the specific test problem shown; it should be reevaluated for different applications.

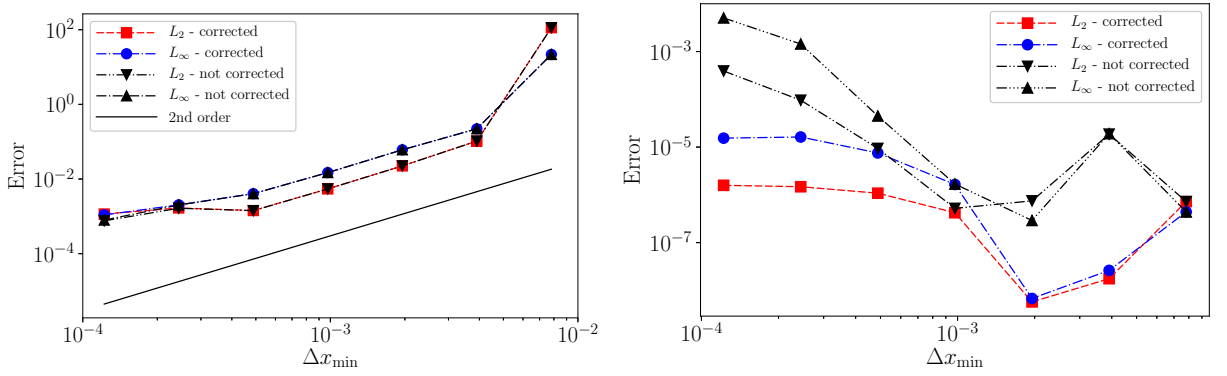
In figure 7, the L_2 and L_∞ norm with respect to the analytical solution are shown with increasing base level resolution. As expected a clear second-order convergence behavior is observed for the proposed multi-resolution scheme. We also plot the target error for the case, where the source correction term is neglected. From the figure it is apparent that the influence of the source correction term on both the convergence and the magnitude of target error is negligible for global refinement and a total of three refinement levels.

In addition to changing the base level resolution, we now fix the base level resolution and increase the number of refinement levels successively according to the refinement criterion in Eq.(29). The corresponding convergence of the error in the fine level for both the corrected and the uncorrected version is shown in figure 8a. Note that while second-



(a) L_2 and L_∞ norm in the finest level with respect to the analytical target field Ψ for increasing levels of refinement. (b) L_2 and L_∞ norm of $\omega_h = L_Q \Psi_h$ with respect to the analytical source field ω for increasing levels of refinement.

Fig. 8: Convergence study of six vortex rings with refinement coefficient $\alpha = 1/32$

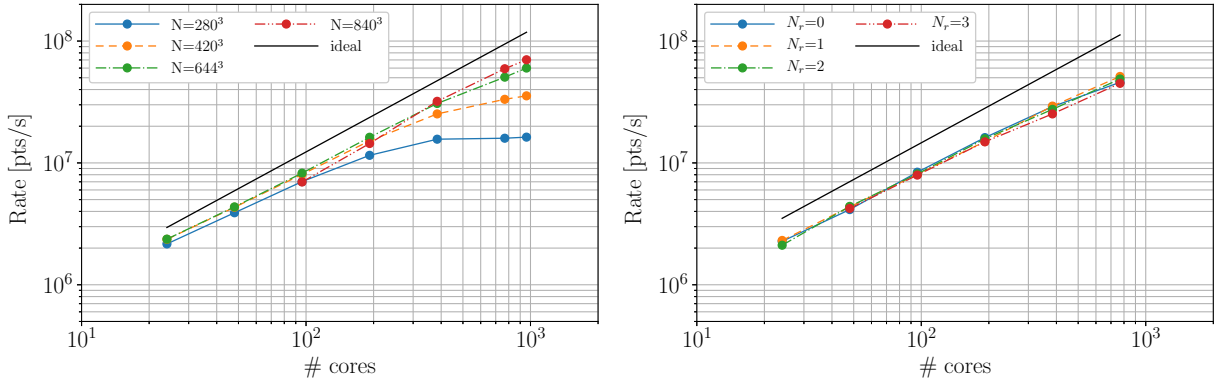


(a) L_2 and L_∞ norm in the finest level with respect to the analytical target field Ψ for increasing levels of refinement. (b) L_2 and L_∞ norm of $\omega_h = L_Q \Psi_h$ with respect to the analytical source field ω for increasing levels of refinement.

Fig. 9: Convergence study of six vortex rings with refinement coefficient $\alpha = 1/8$

order of accuracy is always retained in the proposed scheme when refining a fixed mesh topology on all refinement levels uniformly (see figure 7), the order of accuracy will degrade if non-negligible source is contained in non-refined regions as apparent in figure 8a. The saturation of the error is dependent on the specific refinement criteria chosen as exemplary shown in figure 9a, where the refinement criterion was relaxed and $\alpha = 1/8$ was used.

We also investigate the effect of the source correction term on the consistency between forward Laplace operator L_Q and its numerically computed inverse Ψ_h . To that end, we take the 7-pt Laplacian of the target field for each level and compare it to the source field. On a uniform mesh, the source field is recovered up to the precision of the FMM. For the multi-resolution mesh, the errors are reported in figure 8b. The figure shows, as expected, that the error of the source field $\omega_h = L_Q \Psi_h$ compared to ω becomes large and increases with the number of refinement levels when the appropriate source correction terms are not included. On the other hand, the correction has a relatively minor impact on the target field itself, Ψ_h , as shown in figure 8a, but ensures consistency between the forward Laplacian and its inverse. Analogous behavior is observed in figure 9b for the case of $\alpha = 1/8$.



(a) Computational rates for a uniform mesh of size N .

(b) Computational rates for $N = 644$ and an increasing number of refinement levels.

Fig. 10: Computational rates and parallel performance.

4. Efficiency and parallel performance

To demonstrate the efficiency and parallel performance of the proposed block-refined algorithm, we consider the same test case as above and report its efficiency and parallel performance. We also refer to [10] for performance investigations and the parallel implementation strategy of the uniform solver, which is similar to the one employed here. However, for completeness we also report the computational rates here.

The solver was written in a C++ framework and uses MPI for parallel communications as well as FFTW for fast Fourier transforms. The code implements an octree data structure, where each leaf corresponds to a cubic domain in physical space. A server-client model is used for load balancing of the octree, where the server stores the full octree but does not allocate any data. The clients on the other hand store one or multiple sub-trees including the corresponding data. For load balancing, the anticipated load (mainly the load of fast Fourier transforms) is computed for each octant and the leaf octants are sorted according to their Morton code for each level. Finally, the sorted array of leaf octants is split into chunks with almost equal loads, which are then assigned to each processor. Subsequently, the parents are assigned to the processor with the minimum load in a recursive fashion. Note that the FMMs are sequential in terms of the level due to the correction term, which depends on previous levels and thus necessitates level-wise balancing to avoid an imbalance of load on a particular level. All communication patterns between clients are established using the server and communication costs are almost fully hidden using non-blocking MPI calls.

Note that by far most time is spent in the level interactions ($\sim 99\%$) of the algorithm and in particular the level convolution within each FMM. The time to construct or traverse the octree data structure is negligible due to the block-wise nature of the algorithm, which also allows SIMD vectorization of the Fourier transforms and the Hadamard product for additional speed.

Figure 10a shows the strong scaling for the computational rates of our implementation for various domain sizes N . The parallel efficiencies are in line with the implementation of [10] as well as other kernel-independent FMM solvers and thus verifies our implementation. In addition, in figure 10b, the dependence of the computational rate with the

number of refinement levels is plotted for the case of $N = 644^3$. As stated earlier, the complexity scales linearly with the number of levels and we thus expect that the computational rates are independent of the refinement levels. This is confirmed in 10b. Note however that the parallel efficiency is limited to the efficiency of the individual FMMs.

5. Discussion and Conclusion

In this paper, we presented a multi-resolution scheme for inhomogeneous, linear, constant-coefficient difference equations on infinite grids based on an extension of the fast lattice Green's function (FLGF) method. The new method retains inherent advantages of the FLGF method, such as the use of uniform mesh interpolations and FFT-based convolutions, and the inherent computational savings associated with satisfaction of the exact far-field boundary conditions for domains truncated snugly around source regions, but allows for an arbitrary block-wise mesh refinement by factors of two. The refinement procedure consists of regularization, level interaction as well as interpolation. By consistent definition of all operators, second-order accuracy is retained, and the complexity of the scheme remains linear in the number of DoFs. This was demonstrated for different refinement criteria using manufactured solutions for the Poisson equation of the streamfunction of multiple vortex rings at different spatial scales. Linear complexity in the number of refinement levels as well as parallel efficiency was shown numerically. Hence, the method is well suited for large-scale computation of multi-scale phenomena that render conventional solvers based on uniform grids or nonlinear complexity impractical.

Acknowledgments

This work was supported by the SNF Grant No. P2EZP2_178436 (B. D.) and the ONR Grant No. N00014-16-1-2734.

References

- [1] E. N. Economou, *Green's functions in quantum physics*, volume 3, Springer, 1983.
- [2] W. McCrea, F. Whipple, Xxii.random paths in two and three dimensions, *Proceedings of the Royal Society of Edinburgh* 60 (1940) 281–298.
- [3] A. Kulikovskiy, Positive streamer between parallel plate electrodes in atmospheric pressure air, *Journal of physics D: Applied physics* 30 (1997) 441.
- [4] J. W. Cooley, J. W. Tukey, An algorithm for the machine calculation of complex fourier series, *Mathematics of computation* 19 (1965) 297–301.
- [5] L. Greengard, V. Rokhlin, A fast algorithm for particle simulations, *Journal of computational physics* 73 (1987) 325–348.
- [6] J. W. Ruge, K. Stüben, Algebraic multigrid, in: *Multigrid methods*, SIAM, 1987, pp. 73–130.
- [7] B. Smith, P. Bjorstad, W. D. Gropp, W. Gropp, *Domain decomposition: Parallel multilevel methods for elliptic partial differential equations*, Cambridge university press, 2004.
- [8] K. Schneider, O. V. Vasilyev, Wavelet methods in computational fluid dynamics, *Annual review of fluid mechanics* 42 (2010) 473–503.
- [9] A. Gholami, D. Malhotra, H. Sundar, G. Biros, FFT, FMM, or multigrid? A comparative study of state-of-the-art poisson solvers for uniform and nonuniform grids in the unit cube, *SIAM Journal on Scientific Computing* 38 (2016) C280–C306.
- [10] S. Liska, T. Colonius, A parallel fast multipole method for elliptic difference equations, *Journal of Computational Physics* 278 (2014) 76–91.
- [11] A. Gillman, P.-G. Martinsson, A fast solver for poisson problems on infinite regular lattices, *Journal of Computational and Applied Mathematics* 258 (2014) 42–56.
- [12] A. Gillman, P.-G. Martinsson, Fast and accurate numerical methods for solving elliptic difference equations defined on lattices, *Journal of Computational Physics* 229 (2010) 9026–9041.
- [13] P.-G. J. Martinsson, *Fast multiscale methods for lattice equations*, Ph.D. thesis, 2002.
- [14] S. Liska, T. Colonius, A fast lattice green's function method for solving viscous incompressible flows on unbounded domains, *Journal of Computational Physics* 316 (2016) 360–384.
- [15] S. Liska, T. Colonius, A fast immersed boundary method for external incompressible viscous flows using lattice green's functions, *Journal of Computational Physics* 331 (2017) 257–279.
- [16] S. V. Tsynkov, Numerical solution of problems on unbounded domains. A review, *Applied Numerical Mathematics* 27 (1998) 465–532.

- [17] T. Colonius, Modeling artificial boundary conditions for compressible Flow, *Annual Review of Fluid Mechanics* 36 (2004) 315–345.
- [18] D. S. Pradeep, F. Hussain, Effects of boundary condition in numerical simulations of vortex dynamics, *Journal of Fluid Mechanics* 516 (2004) 115–124.
- [19] K. Taira, T. Colonius, The immersed boundary method: A projection approach, *Journal of Computational Physics* 225 (2007) 2118–2137.
- [20] G. Yun, D. Kim, H. Choi, Vortical structures behind a sphere at subcritical Reynolds numbers, *Physics of Fluids* 18 (2006) 015102.
- [21] S. Wang, X. Zhang, An immersed boundary method based on discrete stream function formulation for two- and three-dimensional incompressible flows, *Journal of Computational Physics* 230 (2011) 3479–3499.
- [22] a.M. Roma, C. S. Peskin, M. J. Berger, An Adaptive Version of the Immersed Boundary Method, *Journal of Computational Physics* 153 (1999) 509–534.
- [23] B. E. Griffith, R. D. Hornung, D. M. McQueen, C. S. Peskin, An adaptive, formally second order accurate version of the immersed boundary method, *Journal of Computational Physics* 223 (2007) 10–49.
- [24] T. Colonius, K. Taira, A fast immersed boundary method using a nullspace approach and multi-domain far-field boundary conditions, *Computer Methods in Applied Mechanics and Engineering* 197 (2008) 2131–2146.
- [25] I. Lashuk, A. Chandramowlishwaran, H. Langston, T.-A. Nguyen, R. Sampath, A. Shringarpure, R. Vuduc, L. Ying, D. Zorin, G. Biros, A massively parallel adaptive fast-multipole method on heterogeneous architectures, in: *Proceedings of the Conference on High Performance Computing Networking, Storage and Analysis*, ACM, 2009, p. 58.
- [26] L. Ying, G. Biros, D. Zorin, A kernel-independent adaptive fast multipole algorithm in two and three dimensions, *Journal of Computational Physics* 196 (2004) 591–626.
- [27] H. Sundar, G. Biros, C. Burstedde, J. Rudi, O. Ghattas, G. Stadler, Parallel geometric-algebraic multigrid on unstructured forests of octrees, in: *Proceedings of the International Conference on High Performance Computing, Networking, Storage and Analysis*, IEEE Computer Society Press, 2012, p. 43.
- [28] A. Dubey, A. Almgren, J. Bell, M. Berzins, S. Brandt, G. Bryan, P. Colella, D. Graves, M. Lijewski, F. Löffler, et al., A survey of high level frameworks in block-structured adaptive mesh refinement packages, *Journal of Parallel and Distributed Computing* 74 (2014) 3217–3227.
- [29] R. T. Delves, G. S. Joyce, On the green function for the anisotropic simple cubic lattice, *Annals of Physics* 291 (2001) 71–133.
- [30] M. L. Glasser, I. J. Zucker, Extended watson integrals for the cubic lattices, *Proceedings of the National Academy of Sciences* 74 (1977) 1800–1801.
- [31] P.-G. Martinsson, G. J. Rodin, Asymptotic expansions of lattice green’s functions, *Proceedings of the Royal Society of London. Series A: Mathematical, Physical and Engineering Sciences* 458 (2002) 2609–2622.
- [32] M. Mangad, Asymptotic expansions of fourier transforms and discrete polyharmonic greens functions, *Pacific Journal of Mathematics* 20 (1967) 85–98.