Bergische Universität Wuppertal

Fakultät für Mathematik und Naturwissenschaften

Institute of Mathematical Modelling, Analysis and Computational
Mathematics (IMACM)

Preprint BUW-IMACM 19/40

Karsten Kahl and Nils Kintscher

Automated Local Fourier Analysis (aLFA)

December 17, 2019

http://www.math.uni-wuppertal.de

Automated Local Fourier Analysis (aLFA)*

Karsten Kahl† Nils Kintscher†

December 17, 2019

Abstract

Local Fourier analysis is a commonly used tool to assess the quality and aid in the construction of geometric multigrid methods for translationally invariant operators. In this paper we automate the process of local Fourier analysis and present a framework that can be applied to arbitrary, including non-orthogonal, repetitive structures. To this end we introduce the notion of crystal structures and a suitable definition of corresponding wave functions, which allow for a natural representation of almost all translationally invariant operators that are encountered in applications, e.g., discretizations of systems of PDEs, tight-binding Hamiltonians of crystalline structures, colored domain decomposition approaches and last but not least two- or multigrid hierarchies. Based on this definition we are able to automate the process of local Fourier analysis both with respect to spatial manipulations of operators as well as the Fourier analysis back-end. This automation most notably simplifies the user input by removing the necessity for compatible representations of the involved operators. Each individual operator and its corresponding structure can be provided in any representation chosen by the user.

1 Introduction

Local Fourier analysis (LFA) is a powerful tool used in the construction and analysis of multigrid methods introduced in [4]. The fundamental idea of LFA is to leverage the connection between position space and frequency space via the Fourier transform. That is, in case the involved operators can be described by stencils in position space, meaning that they are translationally invariant, their Fourier transform yields so-called symbols, which can be handled much more easily. In the context of multigrid methods LFA can be used to obtain precise approximations of the asymptotic convergence rate by assessing the spectral radius of the corresponding error propagation operator [23]. This approximation of the convergence rate is (asymptotically) still valid if the translational invariance is slightly violated, as for example when lexicographic Gauss-Seidel type smoothers are used, or in the case of certain non-periodic boundary conditions

*This work was partially funded by Deutsche Forschungsgemeinschaft (DFG) Transregional Collaborative Research Centre 55 (SFB/TRR55)

†School of Mathematics and Natural Sciences, University of Wuppertal, 42097 Germany, {kkahl,kintscher}@math.uni-wuppertal.de

[5, 20, 27]. In other cases a similar convergence rate can usually be obtained by additional processing [12, 29]. Due to these facts, LFA is one of the main tools in the quantitative analysis of two- and multigrid methods.

An introduction to LFA including several examples can be found in [29, 30]. Multigrid methods have first been considered for the solution of the linear systems of equations originating in the discretization of (elliptic) partial differential equations (PDEs) [29]. Due to the fact that the simplest tiling of space is rectangular and discretizations are particularly simple to carry out on such tilings, the usual multigrid components and the LFA have originally been designed and tailored for such discretizations (cf. [29, 30]). Several other geometries, including systems of PDEs, have been considered in the past as well. LFA has been carried out for operators defined on triangular tilings in [7] and on hexagonal tilings in [32]. Further, it has been applied to edge-based quadrilateral discretizations [3], regular Voronoi meshes associated with acute triangular grids [21], edge-based discretizations on triangular grids [22] and jumping coefficients on rectangular grids [2, 18]. These papers do a complete two-grid analysis, and in some cases even a three-grid analysis, which was first introduced in [31].

While the concept of LFA is well understood, its application quickly becomes complex and involved the more frequencies get intermixed, e.g., by block smoothers, in a three-grid analysis, or in higher dimensional problems ($n > 2$). Thus, there exists software that automates the application of the LFA [19, 30]. In contrast to the software described in [30], which basically consists of a collection of templates corresponding to certain smoother and restriction/prolongation strategies for specific problems, the software [19], freely available on GitHub,[1] can be used to analyze arbitrary translationally invariant operators on rectangular grids. This software has for example been used to analyze colored block Jacobi methods in combination with aggressive coarsening applied to PDEs with jumping coefficients in [2, 18]. As the number of frequencies which get intermixed increases with the block-size of the smoother and the growing coarsening rate, a manual analysis of this problem would be laborious.

In contrast to the approach developed in this paper, the bulk of the analysis in the previously mentioned references is mainly carried out in frequency space. LFA with an emphasize on position space is rare in the literature. In [11] under the name *compact Fourier analysis*, a position space oriented LFA is introduced where block Toeplitz matrices are used in order to capture the different classes of unknowns. This work has some aspects in common with the approach to LFA we develop in this paper, but lacks generality as it is limited to simple systems on rectangular grids.

The LFA presented in this paper unifies the position space oriented approaches and allows the treatment of operators on arbitrary repetitive structures. To do so we introduce a mathematical framework for the analysis of translationally invariant operators which alter value distributions on lattices and crystals [1, 24]. These structures can be based on arbitrary sets of primitive vectors, including non-orthogonal ones, i.e., non-rectangular structures. These crystal structures, which naturally occur in the tight-binding descriptions of solid-state physics [1] or discretizations of systems of PDEs, enable the convenient and concise description of the resulting operators and allow for the automatic generation of their representations when enlarging their translational in-

[1] `github.com/hrittich/lfa-lab`

variance, e.g., coarsening in the context of multigrid methods. Furthermore, they are very helpful in the representation of overlapping and non-overlapping block smoothers. Our framework is developed to such an extent that the only task required of the user is to provide a description of the occurring operators with respect to (potentially non-matching) descriptions of the underlying repetitive structures, i.e., each operator can be supplied in the simplest or most convenient representation. The remainder of the analysis can then be carried out automatically. In contrast to previously developed LFA, this is achieved by explicitly including a connection of the operator to its underlying structure. This allows us on one hand to automate the transformation of operators in position space, e.g., by finding a least common lattice of translational invariance of two operators and to rewrite their representations accordingly. On the other hand, this focus on structure yields a natural representation and discretization of the dual space that enables the automation of the frequency space part of the analysis as well. All these tasks can be carried out using basic principles and normal forms of integer linear algebra [24]. In combination these developments alleviate the use of LFA by removing any manual calculations. That is, neither a mixing analysis nor transformations of operators to common (and rectangular) translational invariances have to be carried out by hand. While our automated LFA does not necessarily enlarge the set of methods that are analyzable by conventional LFA, it enables the reliable and easy-to-use analysis of complex methods on complicated structures (e.g., overlapping block smoothers and discretizations of systems of PDEs). An open-source Python implementation of the automated LFA framework [13] is freely available on GitLab.[2]

The automation presented in this paper does have some limitation in terms of the smoothers that can be analyzed. Any sequential, i.e., lexicographic, smoother with overlapping update regions changes values in the overlap multiple times in one application. This cannot be easily translated to the structures introduced in this paper. While such smoothers have been analyzed in frequency space before (cf. [16, 17, 25]), this particular treatment of sequential overlap is momentarily not covered in our framework. Note, that the mere presence of overlap is not the problem here. By introducing a coloring, such that the complete sweep can be split into a sequence of updates where each one of them only changes values at most once, automated LFA can be applied as we are going to demonstrate in this paper. Due to the fact that a coloring in overlapping approaches also favors parallelism over their sequential counterparts, we feel that this limitation is relatively minor when targeting actual applications.

The paper is organized as follows. In section 2 we introduce basic notation to describe the underlying structures of the operators we want to analyze: lattices and crystals. In the context of LFA we are interested in translationally invariant operators which alter value distributions on crystals. These kind of operators and their properties are specified in section 3. After that, section 4 illustrates the introduced notation by means of the discretized Laplacian and the red-black Gauss-Seidel method, which is the simplest method where a difference to the conventional LFA becomes apparent. In here, we manually rewrite the discretized Laplacian with respect to the translational invariance of the red-black splitting, such that the method can be analyzed via the results of section 3. Section 5 contains the general theoretical results which are used to automate the

[2] `gitlab.com/NilsKintscher/alfa`

complete procedure, removing the need to manually modify operators. In here, several results of integer linear algebra are used, which we review in section 2.1. Using these results we obtain the algorithms given in appendix B which, in combination with the arithmetic of multiplication operators given in appendix A, realize the automated LFA. Finally, section 6 contains selected examples to demonstrate the merits of the developed approach. First, we analyze a 4-color overlapping block Gauss-Seidel smoother for the tight-binding Hamiltonian of the carbon allotrope graphene and give some two-grid convergence results. Second, we reproduce the two-level analysis for the curl-curl problem found in [3] as a further illustration of how our approach is applied to complex error propagators and to double-check its results.

2 Lattices and Crystals

In order to be able to automate the process of LFA within a unified framework we first have to review some basic definitions of integer linear algebra and crystallography [1, 24]. An (ideal) crystal is an infinite repetition, defined by a lattice, of a structure element.

Definition 2.1. *Let $\mathfrak{a}_1, \mathfrak{a}_2, \ldots, \mathfrak{a}_n \in \mathbb{R}^n$ be linearly independent. An n-dimensional lattice $\mathbb{L}$ is the set of points*

$$\mathbb{L} = \{x = \sum_{\ell=1}^{n} j_\ell \mathfrak{a}_\ell \in \mathbb{R}^n \,:\, j_1, j_2, \ldots, j_n \in \mathbb{Z}\}.$$

The vectors $\mathfrak{a}_1, \mathfrak{a}_2, \ldots, \mathfrak{a}_n$ are known as the primitive vectors *or* lattice basis. *Using matrix notation, i.e., $\mathcal{A} := \begin{bmatrix} \mathfrak{a}_1 & \mathfrak{a}_2 & \ldots & \mathfrak{a}_n \end{bmatrix}$, we can abbreviate the notation by*

$$\mathbb{L}(\mathcal{A}) := \mathcal{A}\mathbb{Z}^n = \mathbb{L}.$$

Without loss of generality we restrict the definition of the second component of a crystal, the structure element, to primitive cells of the lattice.

Definition 2.2. *A* primitive cell *$\Xi = \Xi(\mathcal{A}) \subset \mathbb{R}^n$ of a lattice $\mathbb{L}(\mathcal{A})$ is a (connected) volume of space that, if translated by all vectors of $\mathbb{L}$, fills up $\mathbb{R}^n$ completely without any overlap, i.e.,*

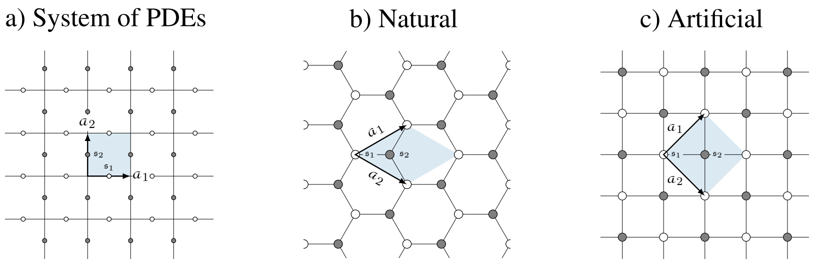
$$\dot{\bigcup}_{x \in \mathbb{L}} \{x + \xi \,:\, \xi \in \Xi\} = \mathbb{R}^n.$$

A common choice of primitive cells is given by

$$\mathcal{P}(\mathcal{A}) := \mathcal{A}[0,1)^n = \{y \in \mathbb{R}^n \,:\, y = \sum_{\ell=1}^{n} \alpha_\ell \mathfrak{a}_\ell, 0 \leq \alpha_\ell < 1\},$$

i.e., parallelotopes spanned by the primitive vectors of $\mathbb{L}(\mathcal{A})$.

The structure element of a crystal is now defined to consist of points $\mathfrak{s}_1, \ldots, \mathfrak{s}_m$ that are contained in a particular primitive cell Ξ.



a) System of PDEs

b) Natural

c) Artificial

Figure 1: Illustration of the crystal representation of three repetitive structures of different origin each consisting of primitive vectors a_1, a_2, a shaded primitive cell $\mathcal{P}(\mathcal{A})$ and a structure element $(\mathfrak{s}_1, \mathfrak{s}_2)$. a) Staggered discretization of the curl-curl system of PDEs with unknowns on the edges of a rectangular lattice. b) Hexagonal lattice of the carbon allotrope graphene. c) Checkerboard coloring of a rectangular lattice as it is encountered in the red-black Gauss-Seidel smoother.

Definition 2.3. *Let $\mathbb{L}(\mathcal{A})$ be a lattice and $\mathfrak{s} \in \Xi(\mathcal{A})^m$, $m \in \mathbb{N}$ be the* structure element*. A* crystal *is defined as the set of tuples*

$$\mathbb{L}^{\mathfrak{s}}(\mathcal{A}) := \{(x+\mathfrak{s}_1, x+\mathfrak{s}_2, \ldots, x+\mathfrak{s}_m) : x \in \mathbb{L}(\mathcal{A}),\ \mathfrak{s} = (\mathfrak{s}_1, \ldots, \mathfrak{s}_m)\}.$$

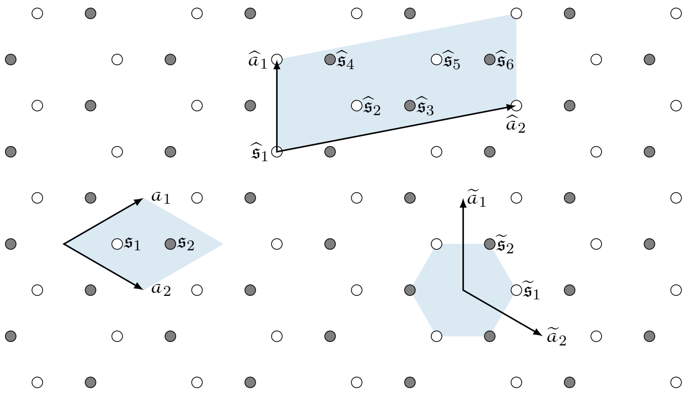
The elements of $\mathbb{L}^{\mathfrak{s}}(\mathcal{A})$ are collectively written as $x + \mathfrak{s} = (x+\mathfrak{s}_1, x+\mathfrak{s}_2, \ldots, x+\mathfrak{s}_m)$.

Remark 1. *We define a crystal and the associated structure element to be a tuple instead of a set as we want to study value distributions on crystals and particular operators which manipulate them. For this purpose, the order of a structure element is of importance.*

To give an idea of the various occurrences of crystal structures in numerical applications we illustrate typical examples in fig. 1 together with their crystal representation. There are three main sources of repetitive structures that are well suited for crystal representations. First, the discretization of systems of PDEs on lattices lead to crystal structures, where the different species of unknowns typically constitute the structure element. Second, tight-binding Hamiltonian formulations in solid state physics for crystalline materials naturally imply a crystal representation based on the atomic structure. Last, colored domain decomposition approaches (e.g., red-black Gauss-Seidel) can be easily represented using crystals, where the smallest structure element typically consists of the union of one domain of each color.

2.1 Sublattices and quotient spaces

There are infinitely many representations of a crystal $\mathbb{L}^{\mathfrak{s}}(\mathcal{A})$. On one hand, the representation of any lattice in $n > 1$ dimensions is non-unique, i.e., there exist different sets of primitive vectors that yield the same lattice structure. On the other hand, different representations of the crystal can be obtained by shifting the structure element or manipulating the underlying lattice structure, e.g., by using integer linear combinations of the primitive vectors and adjusting the structure element accordingly.



$\widehat{a}_1$
$\widehat{\mathfrak{s}}_4$
$\widehat{\mathfrak{s}}_5$
$\widehat{\mathfrak{s}}_6$
$\widehat{\mathfrak{s}}_2$
$\widehat{\mathfrak{s}}_3$
$\widehat{a}_2$
$\widehat{\mathfrak{s}}_1$
a_1
$\widetilde{a}_1$
$\mathfrak{s}_1$
$\mathfrak{s}_2$
$\widetilde{\mathfrak{s}}_2$
a_2
$\widetilde{\mathfrak{s}}_1$
$\widetilde{a}_2$

Figure 2: The structure of graphene: Illustration of three choices of primitive vectors $\mathcal{A}$, $\widehat{\mathcal{A}}$ and $\widetilde{\mathcal{A}}$ with associated structure elements $\mathfrak{s}, \widehat{\mathfrak{s}}$ and $\widetilde{\mathfrak{s}}$ such that $\mathbb{L}^{\mathfrak{s}}(\mathcal{A}) = \mathbb{L}^{\widehat{\mathfrak{s}}}(\widehat{\mathcal{A}}) = \mathbb{L}^{\widetilde{\mathfrak{s}}}(\widetilde{\mathcal{A}})$. Furthermore, three different primitive cells are illustrated: The corresponding parallelotopes $\mathcal{P}(\mathcal{A})$, $\mathcal{P}(\widehat{\mathcal{A}})$ as well as a hexagon corresponding to the Voronoi cell of $\mathbb{L}(\widetilde{\mathcal{A}})$.

In order to cope with this lack of uniqueness of the representation of crystals, illustrated in fig. 2, we introduce basic results from integer linear algebra [24], which resolve the relationship of lattice structures.[3] An important tool in this is the notion of a sublattice.

Definition 2.4. *Let $\mathbb{L}(\mathcal{A})$ and $\mathbb{L}(\mathcal{C})$ be two lattices, if $\mathbb{L}(\mathcal{A}) \supset \mathbb{L}(\mathcal{C})$ then $\mathbb{L}(\mathcal{C})$ is called* sublattice *of $\mathbb{L}(\mathcal{A})$.*

Lemma 2.1. *A lattice $\mathbb{L}(\mathcal{C})$ is a sublattice of $\mathbb{L}(\mathcal{A})$ if and only if $\mathcal{A}^{-1}\mathcal{C} \in \mathbb{Z}^{n\times n}$.*

A key-role in the computational comparison of lattices play the *Hermite and Smith normal forms*. The Hermite normal form defines a canonical choice of primitive vectors, whereas the Smith normal form allows us to find least common sublattices. Using these canonical forms is a crucial ingredient in both the theoretical analysis and the automation of the LFA.

Definition 2.5. *A matrix $U \in \mathbb{Z}^{n\times n}$ is called* unimodular *if* $\det(U) \in \{\pm 1\}$.

Definition 2.6. *A matrix $H \in \mathbb{R}^{n\times n}$ is in* Hermite normal form *(HNF) if it is upper triangular, elementwise non-negative and its row-wise maximum is located on the diagonal.*

Definition 2.7. *A matrix $S \in \mathbb{R}^{n\times n}$ is in* Smith normal form *(SNF) if it is a diagonal matrix and the diagonal entries satisfy ${}^{s_{i+1}}/_{s_i} \in \mathbb{Z}$ for all $i = 1,\dots,n-1$. These entries are called the* elementary divisors.

Theorem 2.1. *Let $\mathcal{A} \in \mathbb{Q}^{n\times n}$, then*

[3] All proofs of lemma 2.1 and theorems 2.1 and 2.2 can be found in [24].

(i) there exists a unimodular $U \in \mathbb{Z}^{n\times n}$ such that $H = \mathcal{A}U$ is in HNF,

(ii) there exist unimodular $U, V \in \mathbb{Z}^{n\times n}$ such that $S = V\mathcal{A}U$ is in SNF.

In addition both normal forms H and S are unique with respect to $\mathcal{A}$.

Remark 2. *Polynomial algorithms to compute the HNF and SNF can for example be found in [8]. Implementations for the computation of these normalforms are for example part of the PARI software package [28].*

Using these definitions and results one obtains a precise statement about the equality of two lattices.

Theorem 2.2. *Let $\mathbb{L}(\mathcal{A})$ and $\mathbb{L}(\mathcal{C})$ be two lattices then the following statements are equivalent.*

(i) $\mathbb{L}(\mathcal{A}) = \mathbb{L}(\mathcal{C})$.

(ii) $\mathcal{A}^{-1}\mathcal{C} \in \mathbb{Z}^{n\times n}$ and $\mathcal{C}^{-1}\mathcal{A} \in \mathbb{Z}^{n\times n}$.

(iii) There exists a unimodular matrix U, such that $\mathcal{C} = \mathcal{A}U$.

(iv) $\mathcal{A}$ and $\mathcal{C}$ have the same HNF.

Instead of analyzing infinite lattice/crystal structures, we limit ourselves to analyze finite dimensional periodic structures due to the fact that we are ultimately aiming to analyze finite dimensional problems. To this end, another helpful tool is the definition of crystal tori which are defined as quotient groups.

Definition 2.8. *Let $\mathbb{L}^{\mathfrak{s}}(\mathcal{A})$ be a crystal and $\mathbb{L}(\mathcal{C}) \subset \mathbb{L}(\mathcal{A})$ be a sublattice. We define the* crystal torus *$T^{\mathfrak{s}}_{\mathcal{A},\mathcal{C}}$ by*

$$T^{\mathfrak{s}}_{\mathcal{A},\mathcal{C}} := \mathbb{L}^{\mathfrak{s}}(\mathcal{A}) \Big/ \mathbb{L}(\mathcal{C}).$$

For every $x + \mathfrak{s} \in \mathbb{L}^{\mathfrak{s}}(\mathcal{A})$, their equivalence class $[x+\mathfrak{s}]$ is in $T^{\mathfrak{s}}_{\mathcal{A},\mathcal{C}}$. Furthermore, the elements of $T^{\mathfrak{s}}_{\mathcal{A},\mathcal{C}}$ are defined by the equivalence

$$[x+\mathfrak{s}] = [y+\mathfrak{s}] \quad \Longleftrightarrow \quad \text{there exists } z \in \mathbb{L}(\mathcal{C}), \text{ such that } x = y + z.$$

For theoretical and practical reasons, e.g., in theorem 3.2 and algorithm B.3, it is necessary to be able to list all elements of a torus $T^{\mathfrak{s}}_{\mathcal{A},\mathcal{A}M} = \{[x] \in T^{\mathfrak{s}}_{\mathcal{A},\mathcal{A}M} : x \in \mathcal{P}(\mathcal{A}M) \cap \mathbb{L}^{\mathfrak{s}}(\mathcal{A})\}$, $M \in \mathbb{Z}^{n\times n}$, uniquely.
To illustrate this point, consider for example an arbitrary lattice $\mathbb{L}(\mathcal{A})$ with $\mathcal{A} \in \mathbb{R}^{2\times 2}$, and the lattice torus $T_{\mathcal{A},\mathcal{A}M}$ with

$$M = \begin{bmatrix} m_1 & m_2 \end{bmatrix} = \begin{bmatrix} 2 & 3 \\ 2 & -2 \end{bmatrix}, \; m_1, m_2 \in \mathbb{Z}^2,$$

as depicted in fig. 3. Even though we know that the quotient space consists of $|T_{\mathcal{A},\mathcal{A}M}| = |\det(M)| = 10$ different elements, there is no apparent canonical list of these elements. Fortunately, a canonical ordering of the lattice points on a torus can be formulated using the Hermite or Smith normal form of M.

Theorem 2.3. *Let $T_{\mathcal{A},\mathcal{C}}$ be arbitrary, i.e., $\mathcal{C} = \mathcal{A}M$ for some $M \in \mathbb{Z}^{n\times n}$.*

- *Let $H \in \mathbb{Z}^{n\times n}$ be the HNF of M with entries H_{ij} (cf. definition 2.6). Defining the index set $I = I_1 \times I_2 \times \ldots \times I_n$ by $I_\ell := \{0, 1, \ldots, H_{\ell\ell} - 1\}$, we then obtain*

  $$T^{\mathfrak{s}}_{\mathcal{A},\mathcal{C}} = \{[x_j + \mathfrak{s}] : x_j = \mathcal{A}j,\ j \in I\}$$

  with $[x_j + \mathfrak{s}] \neq [x_{j'} + \mathfrak{s}] \ \Leftrightarrow\ j \neq j' \in I$.

- *Let $S = U^{-1}MV \in \mathbb{Z}^{n\times n}$ denote the Smith decomposition of M with diagonal entries s_i (cf. definition 2.7) and unimodular matrices U, V. Defining the index set $\tilde{I} = \tilde{I}_1 \times \tilde{I}_2 \times \ldots \times \tilde{I}_n$ by $\tilde{I}_\ell := \{0, 1, \ldots, s_\ell - 1\}$, we then obtain*

  $$T^{\mathfrak{s}}_{\mathcal{A},\mathcal{C}} = \{[x_j + \mathfrak{s}] : x_j = \tilde{\mathcal{A}}j,\ j \in \tilde{I}\}$$

  with $[x_j + \mathfrak{s}] \neq [x_{j'} + \mathfrak{s}] \ \Leftrightarrow\ j \neq j' \in \tilde{I}$, where $\tilde{\mathcal{A}} := \mathcal{A}U$ denotes the altered lattice basis.

Proof. Both statements are a direct consequence of the triangular or diagonal shape of the normal forms and theorem 2.2, i.e., lattices are not changed by unimodular column transformations. On the one hand we have $T^{\mathfrak{s}}_{\mathcal{A},\mathcal{C}} = T^{\mathfrak{s}}_{\mathcal{A},\mathcal{A}M} = T^{\mathfrak{s}}_{\mathcal{A},\mathcal{A}H}$. The second statement follows from $\tilde{\mathcal{A}}S = \mathcal{C}V$ and hence $T^{\mathfrak{s}}_{\mathcal{A},\mathcal{C}} = T^{\mathfrak{s}}_{\mathcal{A}U,\mathcal{C}V} = T^{\mathfrak{s}}_{\tilde{\mathcal{A}},\tilde{\mathcal{A}}S}$. □

In terms of the example of fig. 3 we obtain:

- The Hermite normal form H of M is given by

  $$H = \begin{bmatrix} h_1 & h_2 \end{bmatrix} = \begin{bmatrix} 5 & 2 \\ 0 & 2 \end{bmatrix},\ h_1, h_2 \in \mathbb{Z}^2.$$

  Thus, a unique list of all representatives of $T_{\mathcal{A},\mathcal{A}M}$ is given by

  $$T_{\mathcal{A},\mathcal{A}M} = T_{\mathcal{A},\mathcal{A}H} = \{[x] = [j_1 a_1 + j_2 a_2] : (j_1, j_2) \in \{0,1,2,3,4\} \times \{0,1\}\}.$$

- The Smith decomposition of M is given by

  $$S = \begin{bmatrix} s_1 & 0 \\ 0 & s_2 \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 10 \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ -4 & -1 \end{bmatrix} M \begin{bmatrix} -1 & -3 \\ 1 & 2 \end{bmatrix}.$$

  Thus, another unique list of all representatives of $T_{\mathcal{A},\mathcal{A}M}$ is given by

  $$T_{\mathcal{A},\mathcal{A}M} = T_{\tilde{\mathcal{A}},\tilde{\mathcal{A}}S} = \{[x] = [j_1\tilde{a}_1 + j_2\tilde{a}_2] : (j_1, j_2) \in \{0\} \times \{0,1,\ldots,9\}\},$$

  where $\tilde{\mathcal{A}} = \begin{bmatrix} \tilde{a}_1 & \tilde{a}_2 \end{bmatrix} = \mathcal{A}\begin{bmatrix} 1 & 0 \\ -4 & -1 \end{bmatrix}$.

All tori representations $T_{\mathcal{A},\mathcal{A}M}$, $T_{\mathcal{A},\mathcal{A}H}$ and $T_{\tilde{\mathcal{A}},\tilde{\mathcal{A}}S}$ are depicted in fig. 3. In the remainder we drop the bracket notation for reasons of readability.

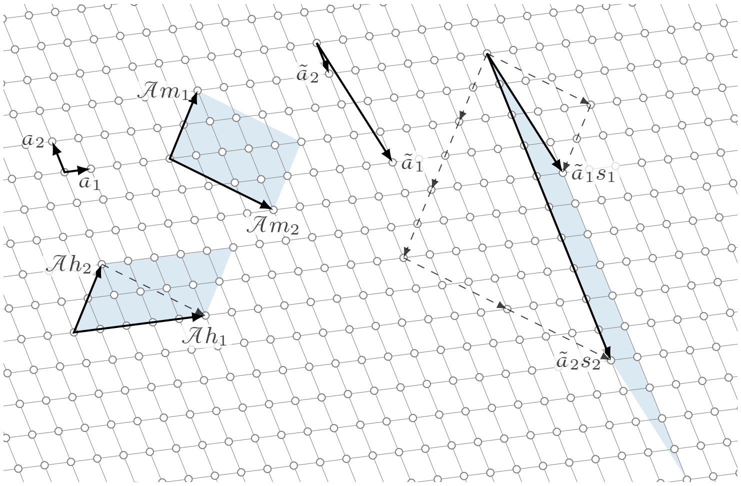


Figure 3: The Hermite normal form H and the Smith normal form S of M yield lattice bases which allow us to define a canonical lexicographic ordering of the lattice points of a crystal torus $T_{\mathcal{A},\mathcal{A}M} = T_{\mathcal{A},\mathcal{A}H} = T_{\tilde{\mathcal{A}},\tilde{\mathcal{A}}S}$.

3 Operators on Crystals

Now that the basic notation of the underlying structure is in place we introduce notation for value distributions and operators on these structures. For aforementioned reasons, we restrict ourselves to the finite dimensional setting by only considering quotient groups of lattices

$$T^{\mathfrak{s}}_{\mathcal{A},\mathcal{Z}} := \mathbb{L}^{\mathfrak{s}}(\mathcal{A}) \Big/ \mathbb{L}(\mathcal{Z})$$

with $\mathbb{L}(\mathcal{Z}) \subset \mathbb{L}(\mathcal{A})$ being an arbitrary sublattice of $\mathbb{L}(\mathcal{A})$. While it is significantly easier to be mathematically precise in this general finite setting than in an infinite setting it is also much closer to the targeted applications, namely finite dimensional approximations of PDEs and Hamiltonians. Eventually we only work with operators as part of numerical simulations, i.e., we face only a finite number of unknowns/lattice points anyway. The quotient group $T^{\mathfrak{s}}_{\mathcal{A},\mathcal{Z}}$, $\mathbb{L}(\mathcal{Z}) \subset \mathbb{L}(\mathcal{A})$, precisely describes such an arbitrarily large but finite torus. To shorten notation we use $T^{\mathfrak{s}}_{\mathcal{A}}$ instead of $T^{\mathfrak{s}}_{\mathcal{A},\mathcal{Z}}$ whenever we do not specify $\mathcal{Z}$ explicitly.

Definition 3.1. *A crystal-operator is a linear function*

$$L : \boldsymbol{\mathcal{L}}(T^{\mathfrak{d}}_{\mathcal{A}}) \longrightarrow \boldsymbol{\mathcal{L}}(T^{\mathfrak{c}}_{\mathcal{A}}),$$

where $T^{\mathfrak{d}}_{\mathcal{A}}$ corresponds to the crystal of the domain and $T^{\mathfrak{c}}_{\mathcal{A}}$ corresponds to the crystal of the codomain. The function spaces are defined by

$$\boldsymbol{\mathcal{L}}(T^{\mathfrak{s}}_{\mathcal{A}}) = \{f = (f_1, \ldots, f_{|\mathfrak{s}|}) : T_{\mathcal{A}} \longrightarrow \mathbb{C}^{|\mathfrak{s}|}\}.$$

A value $f_j(x)$, $x \in \mathbb{L}(\mathcal{A})$, corresponds to a value at the position $x + \mathfrak{s}_j$. The function space is equipped with the scalar product

$$\langle f, g \rangle := \frac{1}{|T_{\mathcal{A},\mathcal{Z}}|} \sum_{x \in T_{\mathcal{A},\mathcal{Z}}} \langle f(x), g(x) \rangle_2,$$

where $\langle f(x), g(x) \rangle_2 := \sum_{\ell=1}^{|\mathfrak{s}|} f_\ell(x) \overline{g_\ell(x)}$ denotes the Euclidean scalar product on $\mathbb{C}^{|\mathfrak{s}|}$.

In the context of LFA we are interested in operators which can be represented in (block) stencil notation. That is, translationally invariant operators that can be written as multiplication operators. As these two properties are in fact equivalent (cf. [26, Theorem 3.16]), we can connect those operators to the notation of crystal structures.

Theorem 3.1. *Let $L : \mathcal{L}(T_{\mathcal{A}}^{\mathfrak{d}}) \longrightarrow \mathcal{L}(T_{\mathcal{A}}^{\mathfrak{c}})$ be a crystal operator. The following statements are equivalent.*

1. *L is a multiplication operator, i.e., there exist matrices $m_L^{(y)} \in \mathbb{C}^{|\mathfrak{c}| \times |\mathfrak{d}|}$ such that for each $x \in T_{\mathcal{A}}$ and $f \in \mathcal{L}(T_{\mathcal{A}}^{\mathfrak{d}})$ we have*

$$(Lf)(x) = \sum_{y \in T_{\mathcal{A}}} m_L^{(y)} f(x+y).$$

2. *L is $(\mathcal{A})$-translationally invariant, i.e.,*

$$L\mathbb{T}_a - \mathbb{T}_a L = 0 \quad \text{for all (primitive) vectors } a \in \mathbb{L}(\mathcal{A}),$$

where the translation operator is defined by $(\mathbb{T}_a f)(x) = f(x + a)$.

For the analysis of such operators the concept of the dual lattice comes in handy as already considered in similar form in [7, 32].

Definition 3.2. *Let $\mathbb{L}(\mathcal{A})$ be a lattice. Its* dual lattice *$\mathbb{L}(\mathcal{B}) = \mathbb{L}(\mathcal{A})^*$ is the set*

$$\mathbb{L}(\mathcal{A})^* := \{k \in \mathbb{R}^n : \langle k, x \rangle_2 \in \mathbb{Z} \text{ for all } x \in \mathbb{L}\}.$$

A lattice basis of the dual lattice is given by $\mathcal{B} = \mathcal{A}^{-T}$. The elements of $\mathbb{L}(\mathcal{A})^*$ may also be referred to as wave vectors.

In addition to the dual space we introduce an orthonormal basis of wave functions that are compatible with the crystal structure introduced in section 2. This basis is an extension of the basis used in [15] to analyze the red-black Gauss-Seidel relaxation. Furthermore, similar basis functions have been used in the context of LFA, e.g. in [2, 3, 6, 9, 14].

Theorem 3.2. *An orthonormal basis for the function space $\mathcal{L}(T_{\mathcal{A},\mathcal{Z}}^{\mathfrak{s}})$ with a structure element $\mathfrak{s} = (\mathfrak{s}_1, \ldots, \mathfrak{s}_m)$ is given by the wave functions $e_{1,k}, e_{2,k}, \ldots, e_{m,k}$ defined by*

$$\big(e_{\ell,k}(x)\big)_j := \begin{cases} e^{2\pi i \langle k,x \rangle_2} & \text{if } j = \ell, \\ 0 & \text{else,} \end{cases}$$

with $k \in T_{\mathcal{A},\mathcal{Z}}^* := \mathbb{L}(\mathcal{Z})^* \big/ \mathbb{L}(\mathcal{A})^*$.

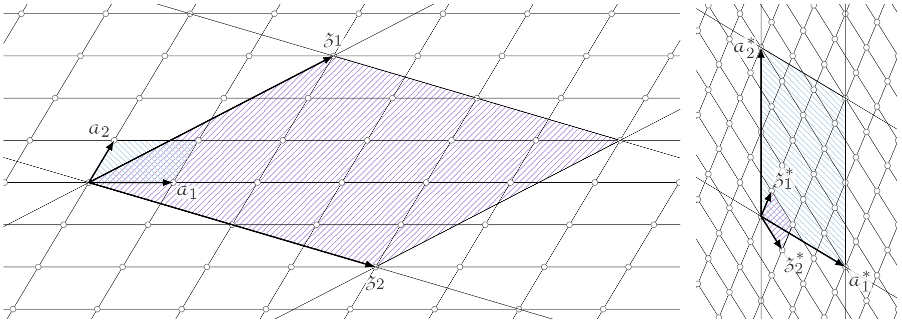


Figure 4: A lattice torus $T_{\mathcal{A},\mathcal{Z}}$ (left) and its dual torus $T^*_{\mathcal{A},\mathcal{Z}}$ (right). In here, the lattices bases are denoted by $\mathcal{A} = \begin{bmatrix} a_1 & a_2 \end{bmatrix}$, $\mathcal{Z} = \begin{bmatrix} \mathfrak{z}_1 & \mathfrak{z}_2 \end{bmatrix}$ and $\mathcal{A}^{-T} = \begin{bmatrix} a_1^* & a_2^* \end{bmatrix}$, $\mathcal{Z}^{-T} = \begin{bmatrix} \mathfrak{z}_1^* & \mathfrak{z}_2^* \end{bmatrix}$.

Proof. The statement follows by a straightforward, but lengthy calculation, by assuming without loss of generality that $\mathcal{A}^{-1}\mathcal{Z}$ is given in Smith normal form, making use of theorem 2.3 and the geometric sum formula. □

An illustration of a lattice torus $T_{\mathcal{A},\mathcal{Z}}$ along with its dual $T^*_{\mathcal{A},\mathcal{Z}}$ is given in fig. 4.
The orthonormal basis of theorem 3.2 can be split into subsets with respect to the wave vector k, i.e., $\mathcal{L}(T^{\mathfrak{s}}_{\mathcal{A},\mathcal{Z}}) = \cup_{k \in T^*_{\mathcal{A},\mathcal{Z}}} \operatorname{span}(H_k)$ with

$$H_k = \operatorname{span}\{e_{\ell,k} : \ell = 1, \ldots, m\}. \tag{1}$$

Theorem 3.3. *Let $L : \mathcal{L}(T^{\mathfrak{s}}_{\mathcal{A},\mathcal{Z}}) \to \mathcal{L}(T^{\mathfrak{s}}_{\mathcal{A},\mathcal{Z}})$ be a multiplication operator. Then the subspaces H_k of eq.* (1) *are L-invariant, i.e., $L(H_k) \subseteq H_k$.*

Proof. Let $e_k(x)$ denote an arbitrary value distribution in H_k. That is, there exist $\alpha_1, \ldots, \alpha_m \in \mathbb{C}$ such that

$$e_k(x) = \sum_\ell \alpha_\ell e_{\ell,k} = (\alpha_1 e^{2\pi i \langle k,x \rangle_2}, \ldots, \alpha_m e^{2\pi i \langle k,x \rangle_2})^T \in \operatorname{span}(H_k).$$

Then we obtain by direct calculation

$$(Le_k)(x) = \sum_y m_L^{(y)} e_k(x+y) = \left(\sum_y m_L^{(y)} e^{2\pi i \langle k,y \rangle_2} \right) e_k(x). \tag{2}$$

□

Due to their L-invariance the subspaces H_k are oftentimes referred to as *spaces of harmonics*. Thus we can easily represent any $\mathcal{A}$-translationally operator via its *symbols*, which are formally defined as follows.

Definition 3.3. *Let $L : \mathcal{L}(T^{\mathfrak{s}}_{\mathcal{A},\mathcal{Z}}) \to \mathcal{L}(T^{\mathfrak{t}}_{\mathcal{A},\mathcal{Z}})$ be a multiplication operator with*

$$(Lf)(x) = \sum_{y \in T_{\mathcal{A},\mathcal{Z}}} m_L^{(y)} f(x+y),\ m_L^{(y)} \in \mathbb{C}^{|\mathfrak{t}| \times |\mathfrak{s}|}.$$

We define the symbol *of L according to eq.* (2) *by*

$$L_k := \sum_{y\in T_{\mathcal{A},\mathcal{Z}}} m_L^{(y)} m_k^{(y)} \text{ with } m_k^{(y)} := e^{2\pi i\langle k,y\rangle_2}.$$

In case $\mathfrak{s} = \mathfrak{t}$ the spectrum of L can then be extracted from its symbols L_k.

Theorem 3.4. *Let $L : \mathcal{L}(T^{\mathfrak{s}}_{\mathcal{A},\mathcal{Z}}) \to \mathcal{L}(T^{\mathfrak{s}}_{\mathcal{A},\mathcal{Z}})$ be a multiplication operator with*

$$(Lf)(x) = \sum_{y\in T_{\mathcal{A},\mathcal{Z}}} m_L^{(y)} f(x+y),\ m_L^{(y)} \in \mathbb{C}^{|\mathfrak{s}|\times|\mathfrak{s}|}.$$

Then $\operatorname{spec}(L) = \cup_{k\in T^*_{\mathcal{A},\mathcal{Z}}} \operatorname{spec}(L_k)$.

Proof. Follows immediately due to the orthonormality of the basis $e_{\ell,k}$ (cf. theorem 3.2) and the L-invariance of the subspaces H_k (cf. theorem 3.3). □

Remark 3. *The main purpose of $\mathcal{Z}$, i.e., the set of primitive vectors that define an arbitrary sublattice $\mathbb{L}(\mathcal{Z})$ of $\mathbb{L}(\mathcal{A})$, is to simplify the theory developed in section 3 by turning an infinite dimensional setting to an (arbitrarily large) finite one. Though there is another interpretation to it as well. In theorem 3.4 $\mathcal{Z}$ explicitly specifies the resolution of the frequency space as seen in fig. 4, i.e.,*

$$T^*_{\mathcal{A},\mathcal{Z}} = \mathbb{L}(\mathcal{Z}^{-T}) \cap \mathcal{P}(\mathcal{A}^{-T}),$$

where the spectrum of the multiplication operator is sampled. Due to the reciprocal nature of the dual space, the larger $|\det(\mathcal{Z})|$ is, the finer the resolution becomes.

With these tools at hand we are able to fully analyze a single multiplication operator, but typically we are interested in analyzing a composition of several operators using LFA. As long as the corresponding domains and codomains of these operators are compatible we can use the rules of computation given in appendix A on the level of multiplication operators and/or on the level of the corresponding symbols. While computing the sum, a product and taking the transpose can easily be done on both levels, taking the (pseudo-)inverse is simple only on the level of symbols. The (pseudo-)inverse of a multiplication operator may have an arbitrarily large number[4] of multipliers $m_{L^{-1}}^{(y)} \neq 0$ and thus there is no simple rule to compute it. In case a pseudo-inverse has to be used in the following we opt to employ the Moore-Penrose pseudo-inverse, which we denote by $S^\dagger$ for a multiplication operator S.

In our framework, which tries to automate as much of the LFA as possible, we would like to allow for user friendly descriptions of all occurring operators. That is, it should be possible to describe operators in terms of their individual translational invariance and ordering of the structure element without having to worry about compatibility issues with other operators on the input level of the analysis. The process how to automatically make crystal representations of operators compatible is explained in detail in section 5. Before diving into the gritty details of this automation process we would like to illustrate the developments made so far with an example in order to convey the introduced notation.

[4] Bounded by the number of lattice points on the (arbitrarily large) torus.

4 An example

Consider the red-black Gauss-Seidel method applied to the discretized Laplacian on the unit square with periodic boundary conditions. The most fundamental representation of the discretized unit square with periodic boundary conditions is given by the torus $T_{\mathcal{A},\mathcal{Z}}$ with

$$\mathcal{Z} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \mathcal{A} = \begin{bmatrix} a_1 & a_2 \end{bmatrix} = \frac{1}{h}\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \text{ for some } h = \frac{1}{N}, N \in \mathbb{N}.$$

Then, the discretized Laplacian $L_h : \boldsymbol{\mathcal{L}}(T^{(0)}_{\mathcal{A},\mathcal{Z}}) \longrightarrow \boldsymbol{\mathcal{L}}(T^{(0)}_{\mathcal{A},\mathcal{Z}})$ using finite differences is given by $(L_h f)(x) := \sum_{y \in \mathbb{L}(\mathcal{A})} m^{(y)}_{L_h} f(x+y)$ with non-zero multipliers:

$$\begin{array}{ccc} & m^{(a_2)}_{L_h} = -\frac{1}{h^2} & \\ m^{(-a_1)}_{L_h} = -\frac{1}{h^2} & m^{(0)}_{L_h} = \frac{4}{h^2} & m^{(a_1)}_{L_h} = -\frac{1}{h^2} \\ & m^{(-a_2)}_{L_h} = -\frac{1}{h^2} & \end{array}$$

The error propagator G of the red-black Gauss-Seidel method can be written with respect to the crystal $T^{(0)}_{\mathcal{A},\mathcal{Z}}$ via

$$G = (I - S_b^\dagger L_h)(I - S_r^\dagger L_h) : \boldsymbol{\mathcal{L}}(T^{(0)}_{\mathcal{A},\mathcal{Z}}) \longrightarrow \boldsymbol{\mathcal{L}}(T^{(0)}_{\mathcal{A},\mathcal{Z}})$$

with

$$(S_r f)(x) := \begin{cases} \frac{4}{h^2} f(x) & x \in X_{\text{red}} \\ 0 & x \in X_{\text{black}} \end{cases} \text{ and } (S_b f)(x) := \begin{cases} 0 & x \in X_{\text{red}} \\ \frac{4}{h^2} f(x) & x \in X_{\text{black}} \end{cases},$$

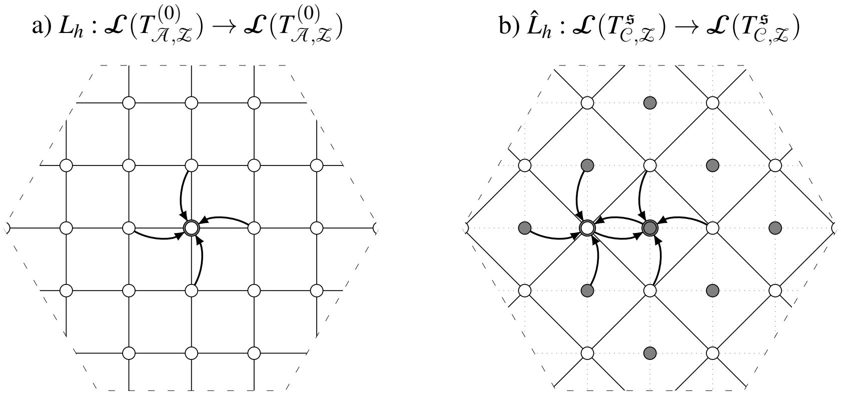
where X_{red} and X_{black} correspond to the red and black unknowns of the torus $T^{(0)}_{\mathcal{A},\mathcal{Z}}$ as illustrated in fig. 5, b). In order to analyze this composite operator in our framework, we write all occurring operators as multiplication operators. It is now important that the red-black splitting of $T^{(0)}_{\mathcal{A},\mathcal{Z}} = X_{\text{red}} \cup X_{\text{black}}$ implies the crystal $T^{\mathfrak{s}}_{\mathcal{C},\mathcal{Z}}$ with

$$c_1 = a_1 + a_2,\; c_2 = a_1 - a_2,\; \mathfrak{s} := (0, a_1) = \mathbb{L}(\mathcal{A}) \cap \mathcal{P}(\mathcal{C})$$

such that $T^{(0)}_{\mathcal{C},\mathcal{Z}} = X_{\text{red}}$, $T^{(a_1)}_{\mathcal{C},\mathcal{Z}} = X_{\text{black}}$ and $T^{(0)}_{\mathcal{A},\mathcal{Z}} = X_{\text{red}} \cup X_{\text{black}} \cong T^{\mathfrak{s}}_{\mathcal{C},\mathcal{Z}}$ (cf. fig. 5, b)). With respect to this crystal the operators S_r and S_b can be written as multiplication operators $\hat{S}_r, \hat{S}_b : \boldsymbol{\mathcal{L}}(T^{\mathfrak{s}}_{\mathcal{C},\mathcal{Z}}) \longrightarrow \boldsymbol{\mathcal{L}}(T^{\mathfrak{s}}_{\mathcal{C},\mathcal{Z}})$ with

$$(\hat{S}_r f)(x) = \begin{bmatrix} \frac{4}{h^2} & 0 \\ 0 & 0 \end{bmatrix} f(x) \text{ and } (\hat{S}_b f)(x) = \begin{bmatrix} 0 & 0 \\ 0 & \frac{4}{h^2} \end{bmatrix} f(x).$$

We now have described all individual operators of G, each one defined with respect to its own (minimal) translational invariance, but the domains and codomains are not



a) $L_h : \mathcal{L}(T^{(0)}_{\mathcal{A},\mathcal{Z}}) \to \mathcal{L}(T^{(0)}_{\mathcal{A},\mathcal{Z}})$

b) $\hat{L}_h : \mathcal{L}(T^{\mathfrak{s}}_{\mathcal{C},\mathcal{Z}}) \to \mathcal{L}(T^{\mathfrak{s}}_{\mathcal{C},\mathcal{Z}})$

Figure 5: The discretized Laplace operator with respect to two different crystal representations. In a) the operator is illustrated with respect to the primitive vectors a_1, a_2 and in b) with respect to c_1, c_2.

identical, i.e., $\mathcal{A} \neq \mathcal{C}$, such that we cannot directly use the computation rules described in appendix A. In order to carry on with the analysis we have to rewrite the operators with respect to a common crystal structure. In this example we construct this structure by hand, but this process can be automated as explained in section 5.

As the crystal $T^{\mathfrak{s}}_{\mathcal{C},\mathcal{Z}}$ is yet another representation of $T^{(0)}_{\mathcal{A},\mathcal{Z}}$, the operator L_h can be rewritten with respect to this crystal (cf. fig. 5) as $\hat{L}_h : \mathcal{L}(T^{\mathfrak{s}}_{\mathcal{C},\mathcal{Z}}) \longrightarrow \mathcal{L}(T^{\mathfrak{s}}_{\mathcal{C},\mathcal{Z}})$ with $(\hat{L}_h f)(x) := \sum_{y \in \mathbb{L}(\mathcal{C})} m^{(y)}_{\hat{L}_h} f(x+y)$ and non-zero multipliers:

$$m^{(-c_2)}_{\hat{L}_h} = \frac{1}{h^2}\begin{bmatrix} 0 & 0 \\ -1 & 0 \end{bmatrix} \qquad m^{(c_1)}_{\hat{L}_h} = \frac{1}{h^2}\begin{bmatrix} 0 & -1 \\ 0 & 0 \end{bmatrix}$$

$$m^{-(c_1+c_2)}_{\hat{L}_h} = \frac{1}{h^2}\begin{bmatrix} 0 & 0 \\ -1 & 0 \end{bmatrix} \qquad m^{(0)}_{\hat{L}_h} = \frac{1}{h^2}\begin{bmatrix} 4 & -1 \\ -1 & 4 \end{bmatrix} \qquad m^{(c_1+c_2)}_{\hat{L}_h} = \frac{1}{h^2}\begin{bmatrix} 0 & -1 \\ 0 & 0 \end{bmatrix}$$

$$m^{(-c_1)}_{\hat{L}_h} = \frac{1}{h^2}\begin{bmatrix} 0 & 0 \\ -1 & 0 \end{bmatrix} \qquad m^{(c_2)}_{\hat{L}_h} = \frac{1}{h^2}\begin{bmatrix} 0 & -1 \\ 0 & 0 \end{bmatrix}$$

Thus, the spectrum of the error propagator of the red-black Gauss-Seidel method applied to the Laplacian

$$\hat{G} = (I - \hat{S}^{\dagger}_{b}\hat{L}_h)(I - \hat{S}^{\dagger}_{r}\hat{L}_h) \tag{3}$$

can now be obtained elementwise for each fixed $k \in T^{*}_{\mathcal{C},\mathcal{Z}} = \mathcal{P}(\mathcal{C}^{-T}) \cap \mathbb{L}(\mathcal{Z}^{-T})$ by first computing the individual *symbols* $I_k, (\hat{S}_r)_k, (\hat{S}_b)_k$ and $(\hat{L}_h)_k$ and assembling the symbols $\hat{G}_k$ according to eq. (3) and the rules in appendix A, followed by the computation of the eigenvalues of the matrices $\hat{G}_k$. The resulting spectral information of the discretized Laplace operator $\hat{L}_h$ and the error propagator $\hat{G}$ is illustrated in fig. 6, where it is sampled on the dual lattice $T^{*}_{\mathcal{C},\mathcal{Z}}$. Note, that one naturally obtains two eigenvalues per sampled wave vector k. In case of the spectrum of the error propagator, one of the two eigenvalues is equal to zero for all wave vectors k.

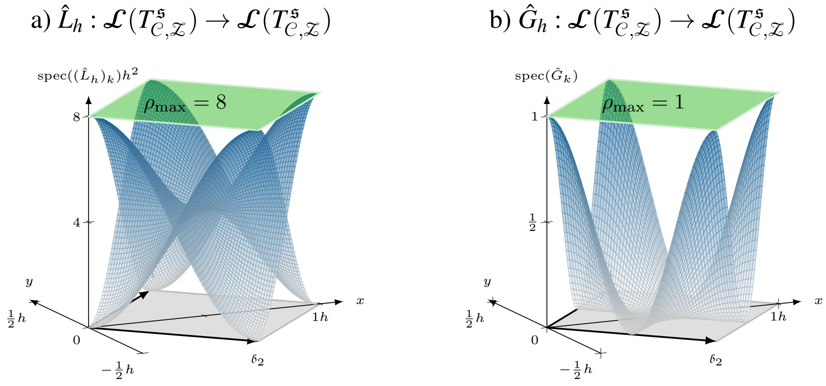


Figure 6: The spectra of a) the discretized Laplace operator and b) the red-black Gauss-Seidel smoother with respect to the red-black crystal structure. In here, $\mathcal{b}_1$ and $\mathcal{b}_2$ denote the columns of $\mathcal{C}^{-T}$.

5 Crystal representations and natural isomorphisms

In general, we are given several multiplication operators which make up the error propagator of an iterative method, each defined with respect to its own (minimal) translational invariance. In order to analyze the method we thus need to find a common denominator, i.e., a lattice basis corresponding to the collective translational invariance, and rewrite the operators accordingly. The following theorem yields a set of primitive vectors of such a collective translational invariance for two arbitrary lattices, if it exists.

Theorem 5.1. *Given two n-dimensional lattices $\mathbb{L}(\mathcal{A})$, $\mathbb{L}(\mathcal{B})$. If there exists an $r \in \mathbb{Z}$, such that $M = r\mathcal{A}^{-1}\mathcal{B} \in \mathbb{Z}^{n\times n}$, then there is a lattice $\mathbb{L}(\mathcal{C})$ with $\mathbb{L}(\mathcal{C}) \subset \mathbb{L}(\mathcal{A})$ and $\mathbb{L}(\mathcal{C}) \subset \mathbb{L}(\mathcal{B})$ with $|\det(\mathcal{C})|$ as small as possible. A lattice basis of $\mathbb{L}(\mathcal{C})$ is given by*

$$\mathcal{C} = \mathcal{B}T^{-1}N_{\mathcal{B}},$$

where $N_{\mathcal{B}}$ is a diagonal matrix with $(N_{\mathcal{B}})_{i,i} := r \cdot \gcd(r,s_i)^{-1}$, where $S = V^{-1}MT^{-1} = \operatorname{diag}(s_1,\dots,s_n)$ is the SNF of M (cf. definition 2.7) and $\gcd(r,s_i)$ denotes the greatest common divisor of r and s_i. Consequently, we write $\mathbb{L}(\mathcal{C}) = \operatorname{lcm}(\mathbb{L}(\mathcal{A}),\mathbb{L}(\mathcal{B}))$ and call it the least common multiple *of $\mathbb{L}(\mathcal{A})$ and $\mathbb{L}(\mathcal{B})$.*

Proof. Due to lemma 2.1 it is sufficient to find integral matrices $N_{\mathcal{A}}, N_{\mathcal{B}}$, such that

$$\mathbb{L}(\mathcal{C}) = \mathbb{L}(\mathcal{A}N_{\mathcal{A}}) = \mathbb{L}(\mathcal{B}N_{\mathcal{B}})$$

with $|\det(N_{\mathcal{A}})|$ and $|\det(N_{\mathcal{B}})|$ as small as possible. Using theorem 2.2, i.e., $\mathbb{L}(\mathcal{B}) = \mathbb{L}(\mathcal{B}V_1)$, $\quad \mathbb{L}(\mathcal{C}) = \mathbb{L}(\mathcal{C}U_1)$, for any unimodular matrices U_1, V_1, we can assume the equality

$$\mathcal{A}N_{\mathcal{A}} = \mathcal{B}UN_{\mathcal{B}}$$

for any unimodular U and $N_{\mathcal{B}}$ in Hermite normal form (cf. theorem 2.1). Plugging in the Smith decomposition VST of $M = r\mathcal{A}^{-1}\mathcal{B}$ and defining $U := T^{-1}$, we find

$$N_{\mathcal{A}} \quad = \quad \mathcal{A}^{-1}\mathcal{B}T^{-1}N_{\mathcal{B}} \quad = \quad \tfrac{1}{r}VSN_{\mathcal{B}}.$$

Both matrices

$$N_{\mathcal{B}} = \begin{bmatrix} (N_{\mathcal{B}})_{1,1} & \cdots & (N_{\mathcal{B}})_{1,n} \\ & \ddots & \vdots \\ & & (N_{\mathcal{B}})_{n,n} \end{bmatrix} \text{ and } \frac{1}{r}SN_{\mathcal{B}} = \begin{bmatrix} \frac{s_1}{r}(N_{\mathcal{B}})_{1,1} & \cdots & (N_{\mathcal{B}})_{1,n}\frac{s_1}{r} \\ & \ddots & \vdots \\ & & (N_{\mathcal{B}})_{n,n}\frac{s_n}{r} \end{bmatrix}$$

have to be integral with

$$|\det(N_{\mathcal{B}})| = |\prod_{i=1}^{n}(N_{\mathcal{B}})_{i,i}| \text{ and } |\det(\frac{1}{r}SN_{\mathcal{B}})| = |\prod_{i=1}^{n}(N_{\mathcal{B}})_{i,i}\frac{s_i}{r}|$$

as small as possible. It can easily be verified that

$$(N_{\mathcal{B}})_{i,i} := \frac{r}{\gcd(r, s_i)}$$

is the optimal choice for the diagonal entries. With this choice, the off-diagonal entries $(N_{\mathcal{B}})_{i,j}$ have to be integral multiples of $(N_{\mathcal{B}})_{i,i}$. Due to the fact that $N_{\mathcal{B}}$ is in Hermite normal form, the off-diagonal entries are zero. □

We now study different representations of the same crystal structure in order to derive a general way to rewrite a multiplication operator with respect to some coarser crystal structure corresponding to a sublattice, as has been done manually in section 4 for the discretized Laplacian.

Theorem 5.2 (Rewriting a crystal with respect to a sublattice)**.** *Let* $\mathbb{L}^{\mathfrak{s}}(\mathcal{A})$ *be a crystal and* $\mathbb{L}(\mathcal{C}) \subset \mathbb{L}(\mathcal{A})$ *a sublattice. Denoting* $T_{\mathcal{A},\mathcal{C}} = \{\mathfrak{t}_1, \ldots, \mathfrak{t}_p\}$*,*[5] *the set*

$$\Xi(\mathcal{C}) := \{x + \delta \in \mathbb{R}^n \, : \, \delta \in \mathcal{P}(\mathcal{A}), x \in \{\mathfrak{t}_1, \ldots, \mathfrak{t}_p\}\}$$

defines a primitive cell of $\mathbb{L}(\mathcal{C})$*, and the tuple*

$$\mathfrak{u} = (\mathfrak{t}_1 + \mathfrak{s}_1, \ldots, \mathfrak{t}_1 + \mathfrak{s}_m, \mathfrak{t}_2 + \mathfrak{s}_1, \ldots, \mathfrak{t}_p + \mathfrak{s}_m) \in \Xi(\mathcal{C})^{p \cdot m}$$

defines a structure element of $\mathbb{L}^{\mathfrak{u}}(\mathcal{C})$ *such that* $\mathbb{L}^{\mathfrak{u}}(\mathcal{C}) \cong \mathbb{L}^{\mathfrak{s}}(\mathcal{A})$*, meant as a one-to-p correspondence.*

Proof. Without loss of generality we may assume $\mathfrak{t}_j \in \mathcal{P}(\mathcal{C})$, $j = 1, \ldots, p$. Then, each element in $\mathbb{L}^{\mathfrak{s}}(\mathcal{A})$ can be written as

$$z = (\mathcal{A}x + \mathfrak{s}_1, \ldots, \mathcal{A}x + \mathfrak{s}_m)$$

and there is a unique y, such that $\mathcal{A}x = \mathcal{C}y = \mathcal{C}\lfloor y \rfloor + \mathcal{C}(y - \lfloor y \rfloor)$ with $\mathcal{C}\lfloor y \rfloor \in \mathbb{L}(\mathcal{C})$ and $\mathcal{C}(y - \lfloor y \rfloor) = \mathfrak{t}_j \in \mathcal{P}(\mathcal{C}) \cap \mathbb{L}(\mathcal{A})$. Thus, we find z as a unique part of the element

$$\mathcal{C}\lfloor y \rfloor + \mathfrak{u} = (\ldots, \mathcal{C}\lfloor y \rfloor + \mathfrak{t}_j + \mathfrak{s}, \ldots) = (\ldots, z, \ldots).$$

This argument works in the other direction in the same way. □

[5] Recall that a unique list of representatives $T_{\mathcal{A},\mathcal{C}} = \{\mathfrak{t}_1, \ldots, \mathfrak{t}_p\}$ can be found via theorem 2.3.

Remark 4. *Note, that $\mathfrak{u}$, as defined in theorem 5.2, is an explicit representation of $T^{\mathfrak{s}}_{\mathcal{A},\mathcal{C}}$, thus using this explicit representation theorem 5.2 implies a congruence of the function spaces*

$$\mathcal{L}(T^{\mathfrak{s}}_{\mathcal{A},\mathcal{Z}}) \cong \mathcal{L}(T^{T^{\mathfrak{s}}_{\mathcal{A},\mathcal{C}}}_{\mathcal{C},\mathcal{Z}})$$

given by the natural isomorphism $\eta : \mathcal{L}(T^{\mathfrak{s}}_{\mathcal{A},\mathcal{Z}}) \to \mathcal{L}(T^{T^{\mathfrak{s}}_{\mathcal{A},\mathcal{C}}}_{\mathcal{C},\mathcal{Z}})$,

$$\mathcal{L}(T^{\mathfrak{s}}_{\mathcal{A},\mathcal{Z}}) \ni f(\cdot) \mapsto (\eta f)(\cdot) = (f(\cdot + \mathfrak{t}_1), \ldots, f(\cdot + \mathfrak{t}_p)) \in \mathcal{L}(T^{T^{\mathfrak{s}}_{\mathcal{A},\mathcal{C}}}_{\mathcal{C},\mathcal{Z}}),$$

as (f) and (ηf) describe the same value distribution on the crystal. This congruence in turn implies that the coarsest possible crystal interpretation, i.e., $\mathfrak{u} \cong T^{\mathfrak{s}}_{\mathcal{A},\mathcal{Z}}$, is simply the complex coordinate space

$$\mathbb{C}^{mn} = \mathcal{L}(T^{T^{\mathfrak{s}}_{\mathcal{A},\mathcal{Z}}}_{\mathcal{Z},\mathcal{Z}}) \cong \mathcal{L}(T^{\mathfrak{s}}_{\mathcal{A},\mathcal{Z}})$$

with $n := |T_{\mathcal{A},\mathcal{Z}}|$. The scalar product on $\mathcal{L}(T^{\mathfrak{s}}_{\mathcal{A},\mathcal{Z}})$ then corresponds to the Euclidean scalar product on $\mathbb{C}^{mn}$ up to a factor of n.

Using the natural isomorphism of function spaces in remark 4, we can derive the transformations of multiplication operators when coarsening the underlying crystal representation corresponding to a sublattice.

Theorem 5.3 (Rewriting a multiplication operator with respect to a sublattice)**.** *Consider crystals $\mathbb{L}^{\mathfrak{d}}(\mathcal{A})$, $\mathbb{L}^{\mathfrak{c}}(\mathcal{A})$, a sublattice $\mathbb{L}(\mathcal{C}) \subset \mathbb{L}(\mathcal{A})$ and a multiplication operator*

$$L : \mathcal{L}(T^{\mathfrak{d}}_{\mathcal{A}}) \to \mathcal{L}(T^{\mathfrak{c}}_{\mathcal{A}}), \quad (Lf)(x) := \sum_{y \in T_{\mathcal{A}}} m^{(y)}_L f(x+y), \quad m^{(y)}_L \in \mathbb{C}^{|\mathfrak{c}| \times |\mathfrak{d}|}.$$

Then, using $T_{\mathcal{A},\mathcal{C}} = \{\mathfrak{t}_1, \ldots, \mathfrak{t}_p\}$, the multiplication operator

$$G : \mathcal{L}(T^{\hat{\mathfrak{d}}}_{\mathcal{C}}) \to \mathcal{L}(T^{\hat{\mathfrak{c}}}_{\mathcal{C}}), \quad (Gg)(x) = \sum_{y \in T_{\mathcal{C}}} m^{(y)}_G g(x+y), \quad m^{(y)}_G \in \mathbb{C}^{p|\mathfrak{c}| \times p|\mathfrak{d}|},$$

with block matrices $(m^{(y)}_G)_{i,k} := m^{(y - \mathfrak{t}_i + \mathfrak{t}_k)}_L \in \mathbb{C}^{|\mathfrak{c}| \times |\mathfrak{d}|}$ fulfills the commutative diagram:

$$\begin{tikzcd}
\mathcal{L}(T^{\mathfrak{d}}_{\mathcal{A}}) \arrow[r, "L"] \arrow[d, "\eta^{\mathfrak{d}}"] & \mathcal{L}(T^{\mathfrak{c}}_{\mathcal{A}}) \arrow[d, "\eta^{\mathfrak{c}}"] \\
\mathcal{L}(T^{\hat{\mathfrak{d}}}_{\mathcal{C}}) \arrow[r, "G"] & \mathcal{L}(T^{\hat{\mathfrak{c}}}_{\mathcal{C}}).
\end{tikzcd}$$

Here, the mappings for $\mathfrak{s} \in \{\mathfrak{d}, \mathfrak{c}\}$,

$$\eta^{\mathfrak{s}} : \mathcal{L}(T^{\mathfrak{s}}_{\mathcal{A}}) \to \mathcal{L}(T^{\hat{\mathfrak{s}}}_{\mathcal{C}}), \quad f(\cdot) \mapsto (f(\cdot + \mathfrak{t}_1), \ldots, f(\cdot + \mathfrak{t}_p)),$$

denote the natural isomorphisms between the congruent crystal representations.

Proof. A straightforward calculation for each block-row i yields

$$\begin{aligned}
[(\eta^{\mathfrak{c}}Lf)(x)]_i &= (Lf)(x+\mathfrak{t}_i) = \sum_{k=1}^{p}\sum_{y\in T_{\mathcal{C}}} m_L^{(y+\mathfrak{t}_k)} f(x+y+\mathfrak{t}_i+\mathfrak{t}_k) \\
&= \sum_{k=1}^{p}\sum_{y\in T_{\mathcal{C}}} m_L^{(y-\mathfrak{t}_i+\mathfrak{t}_k)} f(x+y+\mathfrak{t}_k) = \sum_{y\in T_{\mathcal{C}}}\sum_{k=1}^{p} (m_G^{(y)})_{i,k} f(x+y+\mathfrak{t}_k) \\
&= [G(f(x+\mathfrak{t}_1),\dots,f(x+\mathfrak{t}_p))]_i = [(G\eta^{\mathfrak{d}}f)(x)]_i.
\end{aligned}$$

□

Using theorem 5.3 we now know how to rewrite multiple multiplication operators with respect to some common crystal structure with a coarser translational invariance. Due to the fact that we do not make any assumption on the initial representation of the crystal structures, the resulting structure elements of theorem 2.3 might differ in their orderings and might contain shifts with respect to the common shift invariance. To automatically remove these differences and determine the corresponding transformations of the associated multiplication operators we first define the notion of congruent structure elements.

Definition 5.1. *Two crystal tori $T_{\mathcal{A}}^{\mathfrak{s}} \cong T_{\mathcal{A}}^{\mathfrak{t}}$, $\mathcal{A}\in\mathbb{R}^n$ are* congruent *with respect to $\mathbb{L}(\mathcal{A})$ if the structure elements are of the same size, i.e., $|\mathfrak{s}|=|\mathfrak{t}|=m$, and there is a permutation $\pi:\{1,\dots,m\}\to\{1,\dots,m\}$ as well as shifts $y_j\in\mathbb{L}(\mathcal{A})$, such that*

$$\mathfrak{s}_j = y_j + \mathfrak{t}_{\pi(j)}$$

In order to introduce a unique representation for the sake of automation, we introduce the following *normal form* and the required transformations to transfer any operator to this form.

Definition 5.2. *Let $L:\mathcal{L}(T_{\mathcal{A}}^{\mathfrak{d}})\to\mathcal{L}(T_{\mathcal{A}}^{\mathfrak{c}})$ be a multiplication operator. We say L is in* normal form *if*

- *the coordinates of the structure elements are found in the primitive cell spanned by the primitive vectors, i.e., $\mathfrak{d}_i,\mathfrak{c}_j\in\mathcal{P}(\mathcal{A})=\mathcal{A}[0,1)^n$ for each i,j,*
- *the structure elements $\mathfrak{d}$ and $\mathfrak{c}$ are sorted lexicographically.*[6]

We now derive the implications of definition 5.1 for multiplication operators when the structure element is element wise shifted or permuted. We do so in two steps, theorem 5.4 and theorem 5.5. First, we show that a shift of an entry of the structure element in the codomain or domain results in a modification of the corresponding row or column of the non-zero multipliers, respectively.

Theorem 5.4 (Shifted structure elements)**.** *Consider the two multiplication operators $L:\mathcal{L}(T_{\mathcal{A}}^{\mathfrak{s}})\to\mathcal{L}(T_{\mathcal{A}}^{\mathfrak{t}})$ and $G:\mathcal{L}(T_{\mathcal{A}}^{\mathfrak{t}})\to\mathcal{L}(T_{\mathcal{A}}^{\mathfrak{u}})$ defined by*

$$(Lf)(x) = \sum_{y\in T_{\mathcal{A}}} m_L^{(y)} f(x+y), \quad m_L^{(y)}\in\mathbb{C}^{|\mathfrak{t}|\times|\mathfrak{s}|},$$

$$(Gg)(x) = \sum_{y\in T_{\mathcal{A}}} m_G^{(y)} g(x+y), \quad m_G^{(y)}\in\mathbb{C}^{|\mathfrak{u}|\times|\mathfrak{t}|}.$$

[6] In case $\mathfrak{d}_i=\mathfrak{d}_j$ or $\mathfrak{c}_i=\mathfrak{c}_j$ for any $i\neq j$ a consistent ordering of i,j has to be defined a priori.

Further let $\hat{\mathfrak{t}}$ be a structure element which is obtained from $\mathfrak{t}$ when shifted element-wise along $\mathbb{L}(\mathcal{A})$, i.e.,

$$\mathfrak{t} = (\mathfrak{t}_1, \dots, \mathfrak{t}_m) = (\hat{\mathfrak{t}}_1 + y_1, \dots, \hat{\mathfrak{t}}_m + y_m) = \hat{\mathfrak{t}} + (y_1, \dots, y_m),$$

where $y_1, \dots, y_m \in \mathbb{L}(\mathcal{A})$ and $m = |\mathfrak{t}|$. Then, the operators $\hat{L}$ and $\hat{G}$ given by

$$\begin{aligned}
(\hat{L}f)(x) &= \sum_{y \in T_{\mathcal{A}}} m_{\hat{L}}^{(y)} f(x+y), \quad m_{\hat{L}}^{(y)} \in \mathcal{C}^{|\mathfrak{t}| \times |\mathfrak{s}|} \quad \textit{with} \quad (m_{\hat{L}}^{(y)})_{i,j} := (m_L^{(y+y_i)})_{i,j}, \\
(\hat{G}f)(x) &= \sum_{y \in T_{\mathcal{A}}} m_{\hat{G}}^{(y)} f(x+y), \quad m_{\hat{G}}^{(y)} \in \mathcal{C}^{|\mathfrak{u}| \times |\mathfrak{t}|} \quad \textit{with} \quad (m_{\hat{G}}^{(y)})_{i,j} := (m_G^{(y-y_j)})_{i,j}
\end{aligned}$$

fulfill the commutative diagram:

$$\begin{tikzcd}
\boldsymbol{\mathcal{L}}(T_{\mathcal{A}}^{\mathfrak{s}}) \arrow[r, "L"] \arrow[dr, "\hat{L}"] & \boldsymbol{\mathcal{L}}(T_{\mathcal{A}}^{\mathfrak{t}}) \arrow[r, "G"] \arrow[d, "\mathbb{T}"] & \boldsymbol{\mathcal{L}}(T_{\mathcal{A}}^{\mathfrak{u}}) \\
& \boldsymbol{\mathcal{L}}(T_{\mathcal{A}}^{\hat{\mathfrak{t}}}) \arrow[ur, "\hat{G}"] &
\end{tikzcd}$$

Proof. The natural isomorphism between the two corresponding function spaces is given by

$$\mathbb{T} : \boldsymbol{\mathcal{L}}(T_{\mathcal{A}}^{\mathfrak{t}}) \to \boldsymbol{\mathcal{L}}(T_{\mathcal{A}}^{\hat{\mathfrak{t}}}),\ f = (f_1, \dots, f_m) \mapsto (f_1(\cdot - y_1), \dots, f_m(\cdot - y_m)) = \mathbb{T}f$$

as f and $(\mathbb{T}f)$ describe the same value distribution on the crystal.[7] Again, a straightforward calculation yields

$$\begin{aligned}
[(\mathbb{T}Lf)(x)]_i &= [(Lf)(x - y_i)]_i \ = \sum_{y \in T_{\mathcal{A}}} \sum_{j=1}^{|\mathfrak{s}|} (m_L^{(y)})_{i,j} f_j(x - y_i + y) \\
&= \sum_{y \in T_{\mathcal{A}}} \sum_{j=1}^{|\mathfrak{s}|} (m_L^{(y+y_i)})_{i,j} f_j(x+y) \ = [\sum_{y \in T_{\mathcal{A}}} m_{\hat{L}}^{(y)} f(x+y)]_i.
\end{aligned}$$

Analogously we find $[(G\mathbb{T}^{-1}g)(x)]_i = [\sum_{y \in T_{\mathcal{A}}} (m_{\hat{G}}^{(y)}) g(x+y)]_i$. □

Finally, we show that permutations of the entries of the structure element result in a transformation of the non-zero multipliers by corresponding permutation matrices.

Theorem 5.5 (Permuted structure elements)**.** *Consider the two multiplication operators $L : \boldsymbol{\mathcal{L}}(T_{\mathcal{A}}^{\mathfrak{s}}) \to \boldsymbol{\mathcal{L}}(T_{\mathcal{A}}^{\mathfrak{t}})$ and $G : \boldsymbol{\mathcal{L}}(T_{\mathcal{A}}^{\mathfrak{t}}) \to \boldsymbol{\mathcal{L}}(T_{\mathcal{A}}^{\mathfrak{u}})$ defined by*

$$\begin{aligned}
(Lf)(x) &= \sum_{y \in T_{\mathcal{A}}} m_L^{(y)} f(x+y), \quad m_L^{(y)} \in \mathcal{C}^{|\mathfrak{t}| \times |\mathfrak{s}|}, \\
(Gg)(x) &= \sum_{y \in T_{\mathcal{A}}} m_G^{(y)} g(x+y), \quad m_G^{(y)} \in \mathcal{C}^{|\mathfrak{u}| \times |\mathfrak{t}|}.
\end{aligned}$$

[7] The left-hand side of $[(\mathbb{T}f)(x)]_i = f_i(x - y_i)$ corresponds to the value at position $x + \hat{\mathfrak{t}}_i = (x - y_i) + \mathfrak{t}_i$ which coincides with the position of the value of the right-hand side.

Let further $\hat{\mathfrak{t}}$ be a structure element which is a permuted version of $\mathfrak{t}$, i.e.,

$$\hat{\mathfrak{t}} = (\hat{\mathfrak{t}}_1, \dots, \hat{\mathfrak{t}}_m) = (\mathfrak{t}_{\pi(1)}, \dots, \mathfrak{t}_{\pi(m)}) = m_\pi \mathfrak{t}$$

where $m = |\mathfrak{t}|$, $\pi : \{1, \dots, m\} \to \{1, \dots, m\}$ is a permutation and $m_\pi \in \{0,1\}^{m \times m}$ the corresponding permutation matrix. Then, the operators $\hat{L}$ and $\hat{G}$ given by

$$\begin{aligned}
(\hat{L}f)(x) &= \sum_{y \in T_{\mathcal{A}}} m_{\hat{L}}^{(y)} f(x+y), \quad m_{\hat{L}}^{(y)} \in \mathcal{C}^{|\mathfrak{t}| \times |\mathfrak{s}|} \quad \textit{with} \quad m_{\hat{L}}^{(y)} := m_\pi m_L^{(y)}, \\
(\hat{G}f)(x) &= \sum_{y \in T_{\mathcal{A}}} m_{\hat{G}}^{(y)} f(x+y), \quad m_{\hat{G}}^{(y)} \in \mathcal{C}^{|\mathfrak{u}| \times |\mathfrak{t}|} \quad \textit{with} \quad m_{\hat{G}}^{(y)} := m_G^{(y)} m_\pi^{-1}
\end{aligned}$$

fulfill the commutative diagram:

$$\begin{tikzcd}
\mathcal{L}(T_{\mathcal{A}}^{\mathfrak{s}}) \arrow[r, "L"] \arrow[dr, "\hat{L}"'] & \mathcal{L}(T_{\mathcal{A}}^{\mathfrak{t}}) \arrow[r, "G"] \arrow[d, "p"] & \mathcal{L}(T_{\mathcal{A}}^{\mathfrak{u}}) \\
& \mathcal{L}(T_{\mathcal{A}}^{\hat{\mathfrak{t}}}) \arrow[ur, "\hat{G}"'] &
\end{tikzcd}$$

Proof. Due to the fact that the natural isomorphism $p : \mathcal{L}(T_{\mathcal{A}}^{\mathfrak{t}}) \to \mathcal{L}(T_{\mathcal{A}}^{\hat{\mathfrak{t}}})$ is a multiplication operator defined by $(pf)(x) = m_\pi f(x)$ for all $x \in \mathbb{L}(\mathcal{A})$, the statement is true due to the rules of computation in lemma A.1. □

Theorem 2.3 and theorems 5.1 to 5.5 allow for the automatic adjustment of crystal representations within the LFA. The corresponding detailed algorithms which make use of these results are given in appendix B.

6 Application

Before demonstrating the application of aLFA to some selected examples let us briefly recapitulate the individual parts of the framework. The introduction of crystal structures in section 2 allow for a canonical description of translationally invariant operators introduced in section 3. Combined with the definition of the dual of a crystal structure in definition 3.2 and a corresponding orthonormal basis in theorem 3.2, the symbol of any single multiplication operator that is translationally invariant with respect to an arbitrary lattice structure, can be expressed (cf. definition 3.3). This combination of choice of basis functions and the explicit connection of operators to their underlying structure enables an automated mixing analysis. This part of the framework can thus be seen on one hand as a unification of positional approaches to LFA by introducing structures that allow for the native treatment of arbitrary translational invariances and on the other hand as a general strategy for the automation of the frequency part back-end of LFA. In order to deal with compositions of operators as encountered in the analysis of iterative methods, the tools provided in section 5 allow for an automatic transformation of the underlying crystal structures into compatible representations and the corresponding transformations of the operators. Thus, the only task remaining for the user is to provide any, i.e., the simplest or most convenient, crystal representation of the operators. The following examples show how such a construction can be carried out and as such serve as a tutorial for the use of the algorithms in appendix B. Annotated Jupyter Notebooks of the presented examples can be found in [13].

6.1 Multicolored block smoother for the tight-binding Hamiltonian of graphene

In [12] a multigrid method for the tight-binding Hamiltonian of the carbon allotrope graphene based on Kaczmarz smoothing is constructed and analyzed using conventional LFA. Due to the hexagonal structure of graphene, the lexicographic ordering of Kaczmarz and the mixing analysis of the coarse grid correction which involved a mixing of eight frequencies, the analysis turned out to be quite lengthy. In this subsection we now want to analyze a two-grid method for this problem where we replace Kaczmarz by an overlapping colored Gauss-Seidel method that allows for better parallelism in the application of the multigrid method. Thus, the goal of this example is two-fold, first we want to show that the tight-binding Hamiltonian can be easily expressed using the native crystal structure of graphene and second that even the analysis of an overlapping block smoother can be carried out with ease using aLFA due to the fact that only a representation of the involved operators is needed.

The graphene structure can be described as a crystal $\mathbb{L}^{\mathfrak{s}}(\mathcal{A})$ where the underlying lattice is triangular, i.e., any three nearby lattice points form an equilateral triangle. We have

$$\mathbb{L}^{\mathfrak{s}}(\mathcal{A}), \quad a_1 = (\frac{3}{2}, \frac{\sqrt{3}}{2})a, \quad a_2 = (\frac{3}{2}, -\frac{\sqrt{3}}{2})a,$$

with the structure element

$$\mathfrak{s} = (\mathfrak{s}_1, \mathfrak{s}_2), \quad \mathfrak{s}_1 = (a, 0) = \frac{1}{3}(a_1 + a_2), \quad \mathfrak{s}_2 = (2a, 0) = \frac{2}{3}(a_1 + a_2).$$

The parameter $a = 1.42\text{Å}$ which denotes the distance of two neighboring atoms can be omitted as it does not occur in the tight-binding formulation. An illustration of the crystal has already been given in fig. 2.

The (nearest neighbor) tight-binding Hamiltonian is a multiplication operator

$$L : \mathcal{L}(T^{\mathfrak{s}}_{\mathcal{A}}) \to \mathcal{L}(T^{\mathfrak{s}}_{\mathcal{A}}), \quad (Lf)(x) := \sum_{y \in T_{\mathcal{A}}} m_L^{(y)} f(x+y)$$

with non-zero multipliers

$$m_L^{(-a_2)} = \begin{bmatrix} 0 & -1 \\ 0 & 0 \end{bmatrix} \qquad m_L^{(a_1)} = \begin{bmatrix} 0 & 0 \\ -1 & 0 \end{bmatrix}$$

$$m_L^{(0)} = \begin{bmatrix} 0 & -1 \\ -1 & 0 \end{bmatrix}$$

$$m_L^{(-a_1)} = \begin{bmatrix} 0 & -1 \\ 0 & 0 \end{bmatrix} \qquad m_L^{(a_2)} = \begin{bmatrix} 0 & 0 \\ -1 & 0 \end{bmatrix}$$

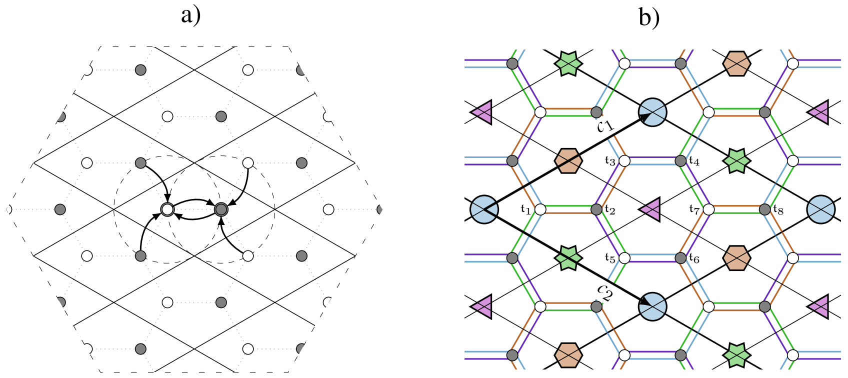


Figure 7: a) Schematic stencil of the tight binding Hamiltonian $L_{[t_0,t_1]}$ of graphene. b) Illustration of the domain decomposition into hexagons. Each unknown belongs to 3 different colors.

as illustrated in fig. 7, a).

Overlapping Hexagons We now present an overlapping block smoother for the tight-binding Hamiltonian L. Consider the non-disjoint splitting (or coloring) of the crystal into hexagons as depicted in fig. 7, b). This splitting has a translational invariance of $\mathcal{C} = 2\mathcal{A}$. Rewriting the graphene crystal $T^{\mathfrak{s}}_{\mathcal{A}}$ with respect to this coarser lattice $\mathbb{L}(\mathcal{C})$, we find $T^{\mathfrak{s}}_{\mathcal{A}} \cong T^{\mathfrak{t}}_{\mathcal{C}}$ with the structure element

$$\mathfrak{t} = (\mathfrak{t}_1, \ldots, \mathfrak{t}_8) = (\mathfrak{s}, \mathfrak{s} + a_1, \mathfrak{s} + a_2, \mathfrak{s} + a_1 + a_2).$$

The splitting is then given by the structure elements

$$\begin{aligned}
&\mathfrak{t}^{(1)} = (\mathfrak{t}^{(1)}_1, \ldots, \mathfrak{t}^{(1)}_6) = (\mathfrak{t}_2, \mathfrak{t}_3, \mathfrak{t}_4, \mathfrak{t}_5, \mathfrak{t}_6, \mathfrak{t}_7),\\
&\mathfrak{t}^{(2)} = \mathfrak{t}^{(1)} + a_1,\ \mathfrak{t}^{(3)} = \mathfrak{t}^{(1)} + a_2, \text{ and } \mathfrak{t}^{(4)} = \mathfrak{t}^{(1)} + a_1 + a_2,
\end{aligned}$$

such that $T^{\mathfrak{t}^{(1)}}_{\mathcal{C}} \mathrel{\widehat{=}}$ ◁, $T^{\mathfrak{t}^{(2)}}_{\mathcal{C}} \mathrel{\widehat{=}}$ ✶, $T^{\mathfrak{t}^{(3)}}_{\mathcal{C}} \mathrel{\widehat{=}}$ ⬡ and $T^{\mathfrak{t}^{(4)}}_{\mathcal{C}} \mathrel{\widehat{=}}$ ●. In the case of the nearest neighbor Hamiltonian any two hexagons of the same color do not interact. Thus, a relaxed sweep on the unknowns of one color is cheap and, in addition, yields a good degree of parallelism.

There are two options to describe the multiplication operators of the multicolored block smoother. On the one hand, we can construct them from scratch by defining the multiplication operator structure as well as the non-zero multipliers. On the other hand, we can derive the structure and non-zero multipliers of the operators corresponding to the colored blocking from the tight-binding Hamiltonian L by exploiting the fact that a colored block corresponds to the operator L restricted to this block. In order to proceed with the latter approach, we first need to find the description $\hat{L}$ of the underlying operator, the tight-binding Hamiltonian L, with respect to the translational invariance of the splitting $\mathcal{C}$. After that the structure element needs to be adjusted such that all unknowns and their couplings among each other are found within the central multiplier $m_{\hat{L}}^{(0)}$. Consider the structure element $\mathfrak{t}$. As can be seen in fig. 7, b), the coupling among the unknowns $\mathfrak{t}^{(\ell)}$ which we want to update simultaneously are found in the multipliers:

$\mathfrak{t}^{(1)}$	$\mathfrak{t}^{(2)}$	$\mathfrak{t}^{(3)}$	$\mathfrak{t}^{(4)}$
$m_{\hat{L}}^{(0)}$	$m_{\hat{L}}^{(0)}$, $m_{\hat{L}}^{(\pm c_1)}$	$m_{\hat{L}}^{(0)}$, $m_{\hat{L}}^{(\pm c_2)}$	$m_{\hat{L}}^{(0)}$, $m_{\hat{L}}^{(\pm c_1)}$, $m_{\hat{L}}^{(\pm c_2)}$, $m_{\hat{L}}^{(\pm(c_1-c_2))}$

Thus, in order to obtain suitable descriptions for $\mathfrak{t}^{(\ell)}$, $\ell \in \{2,3,4\}$, we need to consider shifted versions $\mathfrak{t}+\tau^{(\ell)}$. For example

$$\left[\tau^{(1)}, \quad \tau^{(2)}, \quad \tau^{(3)}, \quad \tau^{(4)}\right] := \left[0, \quad a_1, \quad a_2, \quad a_1+a_2\right]. \tag{4}$$

Finally, the error propagator corresponding to a single color relaxed block Gauss-Seidel update can be written as

$$G^{(\ell)} = (I - \omega(S^{(\ell)})^{\dagger}\hat{L}) : \mathcal{L}(T_{\mathcal{C}}^{\mathfrak{t}+\tau^{(\ell)}}) \to \mathcal{L}(T_{\mathcal{C}}^{\mathfrak{t}+\tau^{(\ell)}})$$

with $(S^{(\ell)}f)(x) := (m_P m_{\hat{L}}^{(0)} m_P) f(x)$ where m_P is the diagonal matrix

$$(m_P)_{ii} = \begin{cases} 1 & \text{for } i \in \{2,3,4,5,6,7\} \\ 0 & \text{else.} \end{cases} \tag{5}$$

We summarize the algorithmical steps to obtain the error propagators.

Obtaining error propagators corresponding to splitting methods

Obtain the description of the tight-binding Hamiltonian L with respect to the translational invariance of the splitting via algorithm B.4:

$$(\hat{L}) = \texttt{LatticeCoarsening}(L, \mathcal{C}).$$

Adjust the structure elements, such that the connections among the unknowns updated simultaneously are found within the central multiplier via algorithm B.6:

$$(\hat{L}^{(\ell)}) = \texttt{ChangeStructureElement}(\hat{L}, \mathfrak{t}+\tau^{(\ell)}, \mathfrak{t}+\tau^{(\ell)}),$$

where $\tau^{(\ell)}$ is defined according to eq. (4). Using m_P as defined in eq. (5) define the operators

$$S^{(\ell)} : \mathcal{L}(T_{\mathcal{C}}^{(\mathfrak{t}+\tau^{(\ell)})}) \to \mathcal{L}(T_{\mathcal{C}}^{(\mathfrak{t}+\tau^{(\ell)})}), \quad m_{S^{(\ell)}}^{(0)} := m_P m_{\hat{L}^{(\ell)}}^{(0)} m_P.$$

The error propagator of a successive update of the four colors corresponds to the product of the error propagators $G^{(\ell)} = (I - \omega(S^{(\ell)})^{\dagger}\hat{L}^{(\ell)})$.

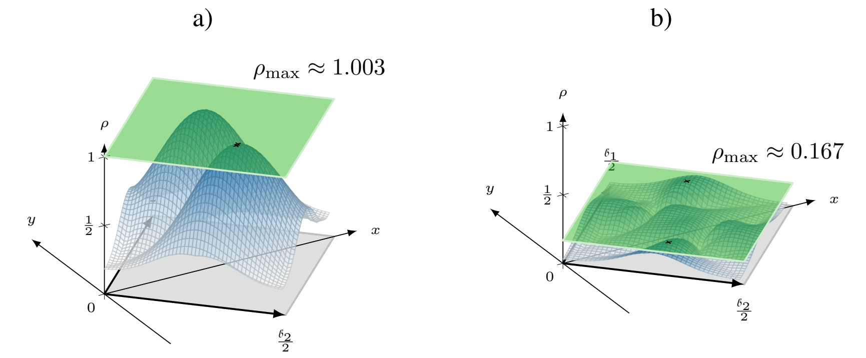


Figure 8: a) Plot of the spectral radii of the four-color hexagonal overlap block smoother $\prod_{\ell=1}^{4} G_{\omega}^{(\ell)}$ with under-relaxation of $\omega = \frac{1}{2}$ for the tight-binding Hamiltonian $L = L_{[0,-1]}$ plotted along $\mathscr{P}(\frac{1}{2}\mathcal{B}) = \mathscr{P}((2\mathcal{A})^{-T})$. b) Plot of the spectral radii of the two-grid method using the overlap smoother.

Now that all operators of the overlapping colored block Gauss-Seidel smoother are defined, we can compute its spectrum. The computation of the eigenvalues of the error propagator is carried out with algorithm B.1 via

$$\texttt{ComputeSpectrum}(g, I, L, S^{(1)}, \ldots, S^{(4)}).$$

The function g denotes the composition of the error propagators

$$(I, L, S^{(1)}, \ldots, S^{(4)}) \overset{g}{\longmapsto} \prod_{\ell=1}^{4} (I - \omega (S^{(\ell)})^{\dagger} L) =: G.$$

Note, that in this algorithm all operators are checked for compatibility and any incompatibility with respect to domains and codomains is dealt with using the transformations introduced in section 5.

For $\omega = \frac{1}{2}$ we obtain the plot in fig. 8, a). As the largest spectral radius is greater than one, this method cannot be used as a standalone solver.

Coarse grid correction In [12] a Galerkin coarse grid correction is used with a corresponding coarse grid correction operator that is defined by

$$E = (I - P(L_c)^{\dagger} RL)$$

with $L_c = RLP$ and $P = R^T$. Just as in the description of the smoother, the shift invariance of the coarse grid is $\mathcal{C} = 2\mathcal{A}$. Defining the coarse structure element by $2\mathfrak{s}$, and with $\mathfrak{t}$ denoting the structure element of the fine crystal according to the coarse lattice basis, the restriction operator can be described as

$$R : \mathcal{L}(T_{2\mathcal{A}}^{\mathfrak{t}}) \to \mathcal{L}(T_{2\mathcal{A}}^{2\mathfrak{s}}).$$

The multipliers of the restriction operator according to [12] then read

$$m_R^{(2\mathscr{a}_1-2\mathscr{a}_2)} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & \frac{1}{4} & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{1}{4} & 0 & 0 & 0 \end{bmatrix}$$

$$m_R^{(-2\mathscr{a}_2)} = \begin{bmatrix} 0 & 0 & 0 & \frac{1}{4} & 0 & -\frac{1}{2} & 0 & -\frac{1}{2} \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \qquad m_R^{(2\mathscr{a}_1)} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -\frac{1}{2} & 0 & \frac{1}{4} & 0 & -\frac{1}{2} & 0 & 0 & 0 \end{bmatrix}$$

$$m_R^{(-2\mathscr{a}_1-2\mathscr{a}_2)} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{1}{4} \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \qquad m_R^{(0)} = \begin{bmatrix} 0 & 1 & 0 & -\frac{1}{2} & 0 & -\frac{1}{2} & 0 & \frac{1}{4} \\ \frac{1}{4} & 0 & -\frac{1}{2} & 0 & -\frac{1}{2} & 0 & 1 & 0 \end{bmatrix} \qquad m_R^{(2\mathscr{a}_1+2\mathscr{a}_2)} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ \frac{1}{4} & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

$$m_R^{(-2\mathscr{a}_1)} = \begin{bmatrix} 0 & 0 & 0 & -\frac{1}{2} & 0 & \frac{1}{4} & 0 & -\frac{1}{2} \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \qquad m_R^{(2\mathscr{a}_2)} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -\frac{1}{2} & 0 & -\frac{1}{2} & 0 & \frac{1}{4} & 0 & 0 & 0 \end{bmatrix}$$

$$m_R^{(-2\mathscr{a}_1+2\mathscr{a}_2)} = \begin{bmatrix} 0 & 0 & 0 & \frac{1}{4} & 0 & 0 & 0 & 0 \\ 0 & 0 & \frac{1}{4} & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

This coarse grid correction is constructed in a way, such that it preserves the kernel of the tight-binding Hamiltonian L, where with $\mathscr{B} = \begin{bmatrix} \mathscr{b}_1 & \mathscr{b}_2 \end{bmatrix} = \mathscr{A}^{-T}$,

$$\ker(L) = \operatorname{span}\{ \begin{bmatrix} e^{2\pi i \langle K, x \rangle_2} \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ e^{2\pi i \langle K, x \rangle_2} \end{bmatrix} : K \in \{ \frac{1}{3}\mathscr{b}_1 + \frac{2}{3}\mathscr{b}_2, \frac{2}{3}\mathscr{b}_1 + \frac{1}{3}\mathscr{b}_2 \} \}.$$

The frequencies corresponding to the kernel modes are known as the *Dirac points*.
The two-grid analysis can now be carried out using algorithm B.1 via

$$\texttt{ComputeSpectrum}(f, I, L, S^{(1)}, \ldots, S^{(4)}, R).$$

The function f denotes the composition of the two-grid error propagator

$$(I, L, S^{(1)}, \ldots, S^{(4)}, R) \xmapsto{f} GEG.$$

A plot of the spectral radii of the two-grid error propagator is given in b) of fig. 8 which shows that the two-grid method converges with a convergence rate of $\rho_{\max} \approx 0.167$. Thus, this new method with overlapping colored block Gauss-Seidel smoothing not only yields opportunities for parallel computations, but also converges faster than the old approach which used Kaczmarz smoothing.
In order to double-check the results of the developed theory, we show in table 1 that the asymptotic convergence rate of the two-grid method with random initial guess x_0 and right-hand-side 0 coincides with the convergence rate obtained in the LFA with a relative accuracy of roughly .002%. This comes as no surprise as the theory is exact for this problem with periodic boundary conditions and we have chosen the sampling of the frequency space in accordance with the problem size (cf. remark 3).

6.2 Two-level analysis for the curl-curl equation

In [3] a complete two-level analysis is carried out for the 2-dimensional curl-curl formulation of Maxwell's equations using conventional LFA. In this subsection we want to reproduce the results of the two-grid method discussed in this paper making use of the native crystal structure of the staggered discretization of the curl-curl equations. The method consists of a so-called half-hybrid smoother, introduced in [10], and a Galerkin coarse grid correction.
The degrees of freedom of the discrete curl-curl equation

$$(\frac{1}{h^2} K_{cc} + \sigma M) x = b \tag{6}$$

iteration i	$\|\|r_i\|\|_2 := \|\|b - Ax_i\|\|_2$	$\rho_i := \frac{\|\|r_i\|\|_2}{\|\|r_{i-1}\|\|_2}$	$\frac{\|\rho_i - \rho_{\text{analytic}}\|}{\rho_{\text{analytic}}}$
398	1.161369e-312	0.16685534	0.00220%
399	1.937806e-313	0.16685539	0.00217%
400	3.233335e-314	0.16685545	0.00213%

Table 1: Convergence history of the two-grid method applied to the tight-binding Hamiltonian of graphene with $(41 \times 41) \cdot 2^3$ unknowns/atoms and periodic boundary conditions. The reported asymptotic convergence rate ρ_i coincides with high precision to the convergence estimate $\rho_{\text{analytic}} = \sup_{k \in T^*_{2\mathcal{A},41\cdot 2\mathcal{A}}} \{|\lambda| : \lambda \text{ eigenvalue of } G_k E_k G_k\} = 0.16685901$ obtained in aLFA.

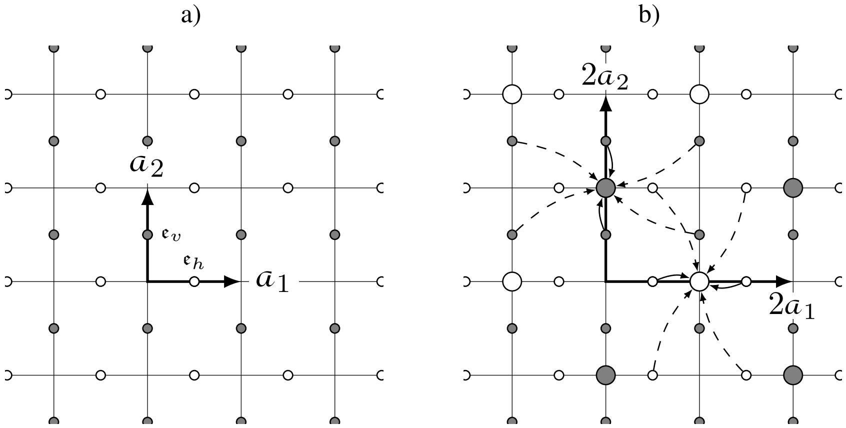


Figure 9: a) Crystal structure of the curl-curl discretization. b) Illustration of the coarse crystal and the restriction operator; dashed lines correspond to restriction weight $\frac{1}{4}$, solid lines to $\frac{1}{2}$.

are associated with the edges of a quadrilateral lattice $\mathbb{L}(\frac{1}{h}\mathcal{A})$ with

$$\mathcal{A} = \begin{bmatrix} a_1 & a_2 \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}.$$

By multiplying eq. (6) with h^2, the grid size can be incorporated into the material parameter $\sigma_h := h^2\sigma$. Then the discrete operator $K = K_{cc} + \sigma_h M$ can be expressed as a multiplication operator by

$$K : \mathcal{L}(T^{\mathfrak{e}}_{\mathcal{A}}) \to \mathcal{L}(T^{\mathfrak{e}}_{\mathcal{A}}) \quad \text{with} \quad \mathfrak{e} = (\mathfrak{e}_h, \mathfrak{e}_v) = (\frac{1}{2}a_1, \frac{1}{2}a_2),$$

where the elements $\mathbb{L}(\mathcal{A}) + \mathfrak{e}_h$ and $\mathbb{L}(\mathcal{A}) + \mathfrak{e}_v$ correspond to the (midpoints of the) horizontal and the vertical edge, respectively (cf. fig. 9). According to [3] the multipliers are given by:

$$
\begin{array}{rclrclrcl}
m_K^{(-a_1+a_2)} &=& \begin{bmatrix} 0 & 0 \\ -1 & 0 \end{bmatrix} & m_K^{(a_2)} &=& \begin{bmatrix} -1+\frac{1}{6}\sigma_h & 0 \\ 1 & 0 \end{bmatrix} & & & \\
m_K^{(-a_1)} &=& \begin{bmatrix} 0 & 0 \\ 1 & -1+\frac{1}{6}\sigma_h \end{bmatrix} & m_K^{(0)} &=& \begin{bmatrix} 2+\frac{2}{3}\sigma_h & -1 \\ -1 & 2+\frac{2}{3}\sigma_h \end{bmatrix} & m_K^{(a_1)} &=& \begin{bmatrix} 0 & 1 \\ 0 & -1+\frac{1}{6}\sigma_h \end{bmatrix} \\
& & & m_K^{(-a_2)} &=& \begin{bmatrix} -1+\frac{1}{6}\sigma_h & 1 \\ 0 & 0 \end{bmatrix} & m_K^{(a_1-a_2)} &=& \begin{bmatrix} 0 & -1 \\ 0 & 0 \end{bmatrix}
\end{array}
$$

Hybrid smoother The (half) hybrid smoother uses an auxiliary space correction. That is, it can be seen as a two-grid method itself where smoothing replaces the exact inversion of the auxiliary space operator. The construction of this operator is as follows. As already stated, we want to reproduce the results in [3]. Due to the fact that lexicographic Gauss-Seidel smoothers are used, we introduce the exact same ordering of the lattice points used in this paper, that is, the coordinates $y \in \mathbb{L}(\mathscr{A})$ are ordered from bottom to top and left to right which leads us to the following definition for $x, y \in \mathbb{L}(\mathscr{A})$:

$$x = (x_1, x_2) < y = (y_1, y_2) :\Leftrightarrow x_2 < y_2 \text{ or } (x_2 = y_2 \text{ and } x_1 < y_1)\,.$$

The error propagator of the smoother on the original crystal $T_{\mathscr{A}}^{\mathfrak{e}}$ is a node-based lexicographic Gauss-Seidel iteration. When updating a horizontal edge $x + \mathfrak{e}_h$, $x \in \mathbb{L}(\mathscr{A})$ it is assumed that all edges $y + \mathfrak{e}_h$ and $y + \hat{\mathfrak{e}}_v$, $\hat{\mathfrak{e}}_v := \mathfrak{e}_v + a_1 - a_2$, with $y < x$ are already updated. When updating the edge $y + \hat{\mathfrak{e}}_v$, it is additionally assumed that $x + \mathfrak{e}_h$ is already updated (cf. [3, Figure 6.1]). In order to obtain the error propagator which exactly represents this ordering, it is convenient to rewrite the operator K with respect to this representation of the structure element, that is

$$K \cong \hat{K} : \boldsymbol{\mathcal{L}}(T_{\mathscr{A}}^{\hat{\mathfrak{e}}}) \to \boldsymbol{\mathcal{L}}(T_{\mathscr{A}}^{\hat{\mathfrak{e}}}) \quad \text{with} \quad \hat{\mathfrak{e}} = (\mathfrak{e}_h, \hat{\mathfrak{e}}_v) \cong \mathfrak{e},$$

which can be obtained with algorithm B.6 via

$$\hat{K} = \texttt{ChangeStructureElement}(K, \hat{\mathfrak{e}}, \hat{\mathfrak{e}}).$$

The corresponding error propagator is then given by

$$G_E = (I - S_E^{\dagger} K) : \boldsymbol{\mathcal{L}}(T_{\mathscr{A}}^{\mathfrak{e}}) \to \boldsymbol{\mathcal{L}}(T_{\mathscr{A}}^{\mathfrak{e}})$$

with the multipliers

$$m_{S_E}^{(y)} = \begin{cases} m_{\hat{K}}^{(y)} & \text{if } y < 0, \\ \texttt{tril}(m_{\hat{K}}^{(y)}) & \text{if } y = 0, \\ 0 & \text{else.} \end{cases}$$

In here, only the lower triangular part $\texttt{tril}(m_{\hat{K}}^{(0)})$ of the central multiplier is used due to the fact that it represents a Gauss-Seidel sweep where a horizontal edge is updated before a vertical edge.

The crystal, where the auxiliary space system is formulated, is given by $T_{\mathscr{A}}^{(0)}$, i.e., the nodal points between the edges. The transfer operator to this crystal is the discrete gradient operator defined by $R_N : \boldsymbol{\mathcal{L}}(T_{\mathscr{A}}^{\mathfrak{e}}) \to \boldsymbol{\mathcal{L}}(T_{\mathscr{A}}^{(0)})$ with non-zero multipliers

$$
\begin{array}{rclrcl}
m_{R_N}^{(-a_1)} &=& \begin{bmatrix} 1 & 0 \end{bmatrix} & m_{R_N}^{(0)} &=& \begin{bmatrix} -1 & -1 \end{bmatrix} \\
& & & m_{R_N}^{(-a_2)} &=& \begin{bmatrix} 0 & 1 \end{bmatrix}.
\end{array}
$$

Using a Galerkin construction, i.e., $P_N = R_N^T$, the coarse grid operator $K_N = R_N K P_N$ can be obtained using the computation rules in lemma A.1. Inversion of this auxiliary space operator is then approximated by Gauss-Seidel. Thus, the auxiliary space correction with a single smoothing step on the auxiliary space is given by

$$G_N = (I - P_N S_N^\dagger R_N K) : \mathcal{L}(T_{\mathcal{A}}^{\mathfrak{e}}) \to \mathcal{L}(T_{\mathcal{A}}^{\mathfrak{e}}),$$

where $S_N : \mathcal{L}(T_{\mathcal{A}}^{(0)}) \to \mathcal{L}(T_{\mathcal{A}}^{(0)})$ consists of the scalar multipliers

$$m_{S_N}^{(y)} = \begin{cases} m_{K_N}^{(y)} & \text{if } y \leq 0, \\ 0 & \text{else.} \end{cases}$$

With this definition of the auxiliary space correction, the half-hybrid smoother consists of the following steps.

1. Smooth on $Kx = b$: $x \leftarrow (I + S_E^\dagger K)x + S_E^\dagger b$,
2. Restrict the residual: $r_N \leftarrow R_N(b - Kx)$,
3. Smooth on $K_N x_N = r_N$ with zero initial guess: $x_N \leftarrow S_N^\dagger r_N$,
4. Prolongate to the primal crystal $x \leftarrow x + P_N x_N$.

The smoother is analyzed analogously to section 6.1 with algorithm B.1 by computing the spectral radii via

$$\texttt{ComputeSpectrum}(g, I, K, S_E, R_N, S_N),$$

where g denotes the composition of the error propagators

$$(I, K, S_E, R_N, S_N) \xmapsto{g} G_N G_E =: G.$$

In fig. 10 a) a contour plot of $\rho(G_k)$ with $\sigma_h = 0.01$, with respect to $k \in \mathcal{A}^{-T}[-\frac{1}{4}, \frac{3}{4})^2$ is given. It corresponds to the result given in [3, Figure 6.2, right].

Coarse grid correction In [3] a Galerkin coarse grid correction is used corresponding to the error propagator

$$E = (I - P(K_c)^\dagger RK)$$

with $K_c = RKP$ and $P = R^T$. In here, the coarse crystal is $\mathbb{L}^{2\mathfrak{e}}(2\mathcal{A})$ and the original crystal $\mathbb{L}^{\mathfrak{e}}(\mathcal{A})$ with respect to the lattice $2\mathcal{A}$ is given by $\mathbb{L}^{\mathfrak{f}}(2\mathcal{A})$ with

$$\mathfrak{f} = (\mathfrak{e}, \mathfrak{e} + a_1, \mathfrak{e} + a_2, \mathfrak{e} + a_1 + a_2).$$

Then, according to [3] the restriction operator is given as $R : \mathcal{L}(T_{2\mathcal{A}}^{\mathfrak{f}}) \to \mathcal{L}(T_{2\mathcal{A}}^{2\mathfrak{s}})$ with the multipliers:

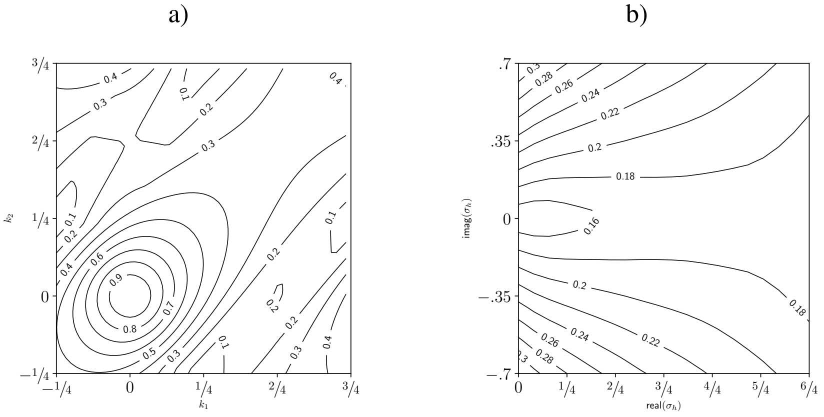


Figure 10: a) Contour plot of $\rho(G_k)$ for the hybrid smoother with respect to $k \in \mathcal{A}^{-T}[-\frac{1}{4}, \frac{3}{4})^2$, $\sigma_h = 0.01$. b) Contour plot of the convergence estimates of the two-grid method $\sup_k \rho(M_k)$ with respect to σ_h. The plots correspond to the results [3, Figure 6.2, right] and [3, Figure 8.1, left], respectively.

$$
\begin{aligned}
m_R^{(-2a_1)} &= \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{1}{4} & 0 & 0 & 0 & \frac{1}{4} \end{bmatrix} \qquad
m_R^{(0)} = \begin{bmatrix} \frac{1}{2} & 0 & \frac{1}{2} & 0 & \frac{1}{4} & 0 & \frac{1}{4} & 0 \\ 0 & \frac{1}{2} & 0 & \frac{1}{4} & 0 & \frac{1}{2} & 0 & \frac{1}{4} \end{bmatrix} \\
m_R^{(-2a_2)} &= \begin{bmatrix} 0 & 0 & 0 & 0 & \frac{1}{4} & 0 & \frac{1}{4} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}
\end{aligned}
$$

Here, each coarse horizontal and vertical edge is connected to its six nearest edges of the same type (cf. fig. 9 b)). The prolongation and coarse grid operator P and K_c can be obtained via lemma A.1.
The spectral radii of the two-grid method can then be obtained via

$$\texttt{ComputeSpectrum}(f, I, K, S_E, R_N, S_N, R)$$

where f denotes the composition of the two-grid error propagator

$$(I, K, S_E, R_N, S_N, R) \xmapsto{f} GEG =: M.$$

Figure 10 b) shows the spectral radii, $\sup_k \rho(M_k)$, of the two-grid error propagator as a function of σ_h. We used exactly the same orderings in pre- and post-smoothing, as described in [3]. Furthermore, we double-checked the results with convergence tests similar to table 1. However, the obtained results show major differences to the one in [3, Figure 8.1, left], which leads us to believe that our assumptions on the lexicographic orderings employed in pre- or post-smoothing in [3] are wrong, as these have a large impact on the convergence rates.

7 Conclusion

In this paper we present an LFA framework that is applicable to arbitrary crystal structures, which are encountered in many applications, e.g., systems of partial differential equations, block smoothers or tight-binding formulations. This was achieved by introducing a rigorous notion of crystal structures and translationally invariant operators that manipulate value distributions on crystal structures. Based on these structures we introduced a complete framework to modify and transform these operators with respect to different crystal representations by using normal forms of integer linear algebra. This allowed us to automate the LFA in both position space, i.e., in terms of multiplication operators/stencils, and frequency space, i.e., in terms of canonical basis functions and samplings. In two examples we showed that the approach can be used for complicated operators, i.e., hexagonal grids, overlapping block smoothers and hybrid smoothers, without requiring insight into the frequency space back-end of LFA. An explicit mixing calculation is no longer needed, and the user only has to provide any representation of the individual operators. The transformation of the individual operators to a compatible representation and the subsequent frequency analysis is then carried out automatically. Even though we have limited ourselves in the examples in this paper to 2-dimensional problems our automated LFA can be applied to operators in higher dimension, where the difference is a larger set of primitive vectors defining the underlying lattice.

The automation presented in this paper does have some limitation. Each individual operator in the analysis is only allowed to change each value of the value distribution at most once. This limitation solely restricts the class of smoothers that can be analyzed with this approach. Any sequential, i.e., lexicographic, smoother with overlapping update regions changes values in the overlap multiple times in one application. This cannot be easily translated to a corresponding local multiplication operator, but it can be dealt with in frequency space (cf. [16, 17, 25]). This particular treatment of sequential overlap is momentarily not covered in our framework. Note, that the mere presence of overlap is not the problem here. By introducing a coloring, such that the complete sweep can be split into a sequence of updates where each one of them only changes values at most once, automated LFA can be applied (cf. section 6.1). Due to the fact that a coloring in overlapping approaches also favors parallelism over their sequential counterparts, we feel that this limitation is relatively minor when targeting actual applications. An open-source implementation of the automated LFA framework [13] is freely available on GitLab[8]. Jupyter Notebooks for all examples from sections 4, 6.1, and 6.2 are included in this software package as well.

A Rules of computation

Calculus of multiplication operators plays a key-role in local Fourier analysis. In this section we list all elementary operations, such as addition and multiplication. Proofs for these rules can be obtained by straightforward calculation.

[8] `gitlab.com/NilsKintscher/alfa`

Lemma A.1. *Let two multiplication operators be given by*

$$L : \mathcal{L}(T^{\mathfrak{s}}_{\mathcal{A},\mathcal{Z}}) \to \mathcal{L}(T^{\mathfrak{t}}_{\mathcal{A},\mathcal{Z}}), \quad (Lf)(x) = \sum_{y \in T_{\mathcal{A},\mathcal{Z}}} m_L^{(y)} f(x+y), \quad m_L^{(y)} \in \mathbb{C}^{|\mathfrak{t}|\times|\mathfrak{s}|},$$
$$G : \mathcal{L}(T^{\mathfrak{u}}_{\mathcal{A},\mathcal{Z}}) \to \mathcal{L}(T^{\mathfrak{v}}_{\mathcal{A},\mathcal{Z}}), \quad (Gf)(x) = \sum_{y \in T_{\mathcal{A},\mathcal{Z}}} m_G^{(y)} f(x+y), \quad m_G^{(y)} \in \mathbb{C}^{|\mathfrak{v}|\times|\mathfrak{u}|}.$$

Then the following operators are multiplication operators as well:

(i) If $\mathfrak{s} = \mathfrak{u}$ and $\mathfrak{t} = \mathfrak{v}$, then $L + G : \mathcal{L}(T^{\mathfrak{s}}_{\mathcal{A},\mathcal{Z}}) \to \mathcal{L}(T^{\mathfrak{t}}_{\mathcal{A},\mathcal{Z}})$ with $m^{(y)}_{L+G} = m^{(y)}_L + m^{(y)}_G$.

(ii) If $\mathfrak{v} = \mathfrak{s}$, then $LG : \mathcal{L}(T^{\mathfrak{u}}_{\mathcal{A},\mathcal{Z}}) \to \mathcal{L}(T^{\mathfrak{t}}_{\mathcal{A},\mathcal{Z}})$ with $m^{(z)}_{LG} = \sum_{y+w=z} m^{(y)}_L \cdot m^{(w)}_G$.

(iii) The adjoint is given by $L^* : \mathcal{L}(T^{\mathfrak{t}}_{\mathcal{A},\mathcal{Z}}) \to \mathcal{L}(T^{\mathfrak{s}}_{\mathcal{A},\mathcal{Z}})$ with $m^{(y)}_{L^*} = (m^{(-y)}_L)^*$.

Theorem A.1. *Let two multiplication operators be given by*

$$L : \mathcal{L}(T^{\mathfrak{s}}_{\mathcal{A},\mathcal{Z}}) \to \mathcal{L}(T^{\mathfrak{t}}_{\mathcal{A},\mathcal{Z}}), \quad (Lf)(x) = \sum_{y \in T_{\mathcal{A},\mathcal{Z}}} m_L^{(y)} f(x+y), \quad m_L^{(y)} \in \mathbb{C}^{|\mathfrak{t}|\times|\mathfrak{s}|},$$
$$G : \mathcal{L}(T^{\mathfrak{u}}_{\mathcal{A},\mathcal{Z}}) \to \mathcal{L}(T^{\mathfrak{v}}_{\mathcal{A},\mathcal{Z}}), \quad (Gf)(x) = \sum_{y \in T_{\mathcal{A},\mathcal{Z}}} m_G^{(y)} f(x+y), \quad m_G^{(y)} \in \mathbb{C}^{|\mathfrak{v}|\times|\mathfrak{u}|}$$

with corresponding symbols L_k and G_k. Then we have the following statements.

(i) Assuming that $\mathfrak{s} = \mathfrak{u}$ and $\mathfrak{t} = \mathfrak{v}$, the symbols of $L + G$ are given by $L_k + G_k$.

(ii) Assuming that $\mathfrak{v} = \mathfrak{s}$, the symbols of $L \cdot G$ are given by $L_k \cdot G_k$.

(iii) The symbols of L^* are given by L_k^*.

(iv) The symbols $(L^\dagger)_k$ are given by $(L_k)^\dagger$.

B Algorithms

Algorithm B.1: Spectrum of a composition of multiplication operators.

```
Input: $L^{(j)} : T^{\mathfrak{s}^{(j)}}_{\mathcal{A}^{(j)}} \to T^{\mathfrak{t}^{(j)}}_{\mathcal{A}^{(j)}}, m^{(x)}_{L^{(j)}} \in \mathbb{C}^{|\mathfrak{t}^{(j)}|\times|\mathfrak{s}^{(j)}|}, x \in \mathbb{L}(\mathcal{A}^{(j)})$ and composition function $f$.
Output: Spectra of $f(L_k^{(1)}, \dots, L_k^{(K)})$, $k \in \mathcal{P}(\mathcal{A}^{-T})$
Function ComputeSpectrum($f, L^{(1)}, \dots, L^{(K)}$)
    $(\hat{L}^{(1)}, \dots, \hat{L}^{(K)})$ = MakeOperatorsCompatible($L^{(1)}, \dots, L^{(K)}$)    ▷ See algorithm B.5
    Sample the dual lattice to obtain $k \in \mathcal{P}(\mathcal{A}^{-T})$    ▷ See remark 3
    Compute the spectra of the composition operator of symbols
        $f(L_k^{(1)}, \dots, L_k^{(K)})$    ▷ See theorem A.1
```

Algorithm B.2: Normalize a multiplication operator.

```
Input: $L : T^{\mathfrak{s}}_{\mathcal{A}} \to T^{\mathfrak{t}}_{\mathcal{A}}, m^{(x)}_L \in \mathbb{C}^{|\mathfrak{t}|\times|\mathfrak{s}|}, x \in \mathbb{L}(\mathcal{A})$.
Output: $G \cong L$ in normal form with $G : T^{\mathfrak{u}}_{\mathcal{A}} \to T^{\mathfrak{v}}_{\mathcal{A}}, m^{(x)}_G \in \mathbb{C}^{|\mathfrak{v}|\times|\mathfrak{u}|}, x \in \mathbb{L}(\mathcal{A})$.
Function $G = \mathtt{Normalize}(L)$                                        ▷ See definition 5.2
    $\mathfrak{u}_j = \mathfrak{s}_j - \mathcal{A}\lfloor\mathcal{A}^{-1}\mathfrak{s}_j\rfloor$ for all $j = 1,\dots,|\mathfrak{s}|$     ▷ Shift into $\mathcal{P}(\mathcal{A}) = \mathcal{A}[0,1)^n$
    $\mathfrak{v}_j = \mathfrak{t}_j - \mathcal{A}\lfloor\mathcal{A}^{-1}\mathfrak{t}_j\rfloor$ for all $j = 1,\dots,|\mathfrak{t}|$     ▷ Shift into $\mathcal{P}(\mathcal{A}) = \mathcal{A}[0,1)^n$
    Sort $\mathfrak{u}$ and $\mathfrak{v}$ lexicographically
    $G = \mathtt{ChangeStructureElement}(L,\mathfrak{u},\mathfrak{v})$              ▷ See algorithm B.6
```

Algorithm B.3: Find all elements in the quotient space of two lattices.

```
Input: Two $n$-dimensional lattices with $\mathbb{L}(\mathcal{C}), \mathbb{L}(\mathcal{A})$ with $\mathbb{L}(\mathcal{C}) \subset \mathbb{L}(\mathcal{A})$.
Output: Structure element $\mathfrak{s} \cong T_{\mathcal{A},\mathcal{C}} = \mathbb{L}(\mathcal{A}) \big/ \mathbb{L}(\mathcal{C})$
Function $\mathfrak{s} = \mathtt{ElementsInQuotientSpace}(\mathcal{A},\mathcal{C})$          ▷ See theorem 2.3
    $H$ = Hermite normal form of $\mathcal{A}^{-1}\mathcal{C}$                       ▷ See theorem 2.1
    $m = \prod_{i=1}^{n} h_{i,i}$                                           ▷ Size of $\mathfrak{s}$
    $\mathfrak{s}_i = 0$ for all $i = 0,\dots,m$                                   ▷ Initialize $\mathfrak{s}$
    for $i = 1,2,\dots,m$
        $k = i - 1$
        for $j = 1,2,\dots,n$
            $t = \mathrm{mod}(k, h_{j,j})$                              ▷ Shift in direction $a_j$
            $k = \frac{k-t}{h_{j,j}}$
            $\mathfrak{s}_i = \mathfrak{s}_i + t a_j$
```

Algorithm B.4: Rewrite a multiplication operator w.r.t. a coarser lattice.

```
Input: $L : T^{\mathfrak{s}}_{\mathcal{A}} \to T^{\mathfrak{t}}_{\mathcal{A}}, m^{(x)}_L \in \mathbb{C}^{|\mathfrak{t}|\times|\mathfrak{s}|}, x \in \mathbb{L}(\mathcal{A})$, and a sublattice $\mathbb{L}(\mathcal{C}) \supset \mathbb{L}(\mathcal{A})$
Output: $G \cong L$ with $G : T^{\mathfrak{u}}_{\mathcal{C}} \to T^{\mathfrak{v}}_{\mathcal{C}}, m^{(x)}_G \in \mathbb{C}^{|\mathfrak{v}|\times|\mathfrak{u}|}, x \in \mathbb{L}(\mathcal{C})$.
Function $G = \mathtt{LatticeCoarsening}(L,\mathcal{C})$                          ▷ See theorem 5.3
    $\mathfrak{e} = \mathtt{ElementsInQuotientSpace}(\mathcal{A},\mathcal{C})$                    ▷ See algorithm B.3
    $\mathfrak{u} = (\mathfrak{e}_1 + \mathfrak{s},\dots,\mathfrak{e}_{|\mathfrak{e}|} + \mathfrak{s}), \mathfrak{v} = (\mathfrak{e}_1 + \mathfrak{t},\dots,\mathfrak{e}_{|\mathfrak{e}|} + \mathfrak{t})$   ▷ Define structure elements
    $(m^{(y)}_G) = 0 \in \mathbb{C}^{|\mathfrak{v}|\times|\mathfrak{u}|}$ for all $y \in \mathbb{L}(\mathcal{C})$          ▷ Initialize new multipliers
    for $m^{(y)}_L \neq 0$
        for $(i,j) \in \{1,\dots,|\mathfrak{e}|\}^2$
            $(m_G^{(\mathcal{C}\lfloor\mathcal{C}^{-1}(y+\mathfrak{e}_i-\mathfrak{e}_j)\rfloor)})_{i,j} = m^{(y)}_L$            ▷ Define multipliers block-wise
```

Algorithm B.5: Rewriting multiplication operators w.r.t. a single lattice.

```
Input: $L^{(j)} : T^{\mathfrak{s}^{(j)}}_{\mathcal{A}^{(j)}} \to T^{\mathfrak{t}^{(j)}}_{\mathcal{A}^{(j)}}, m^{(x)}_{L^{(j)}} \in \mathbb{C}^{|\mathfrak{t}^{(j)}|\times|\mathfrak{s}^{(j)}|}, x \in \mathbb{L}(\mathcal{A}^{(j)})$
Output: $\hat{L}^{(j)} \cong L^{(j)}$ in normal form, $\hat{L}^{(j)} : T^{\mathfrak{u}^{(j)}}_{\mathcal{A}} \to T^{\mathfrak{v}^{(j)}}_{\mathcal{A}}, m^{(x)}_{\hat{L}^{(j)}} \in \mathbb{C}^{|\mathfrak{v}^{(j)}|\times|\mathfrak{u}^{(j)}|}, x \in \mathbb{L}(\mathcal{A})$.
Function $(\hat{L}^{(1)},\dots,\hat{L}^{(K)}) = \mathtt{MakeOperatorsCompatible}(L^{(1)},\dots,L^{(K)})$
    $\mathcal{A} = \mathcal{A}^{(1)}$
    for $j = 2,\dots,K$
        $\mathcal{A} = \mathtt{LeastCommonMultiple}(\mathcal{A},\mathcal{A}^{(j)})$              ▷ See algorithm B.7
    for $j = 1,\dots,K$
        $\hat{L}^{(j)} = \mathtt{LatticeCoarsening}(L^{(j)},\mathcal{A})$                   ▷ See algorithm B.4
        $\hat{L}^{(j)} = \mathtt{Normalize}(\hat{L}^{(j)})$                             ▷ See algorithm B.2
```

Algorithm B.6: Changing the structure elements.

Input: Structure elements $\mathfrak{u} \cong \mathfrak{s}$, $\mathfrak{v} \cong \mathfrak{t}$ w.r.t. $\mathbb{L}(\mathcal{A})$ and $L : T_{\mathcal{A}}^{\mathfrak{s}} \to T_{\mathcal{A}}^{\mathfrak{t}}$, $m_L^{(x)} \in \mathbb{C}^{|\mathfrak{t}|\times|\mathfrak{s}|}, x \in \mathbb{L}(\mathcal{A})$.

Output: $\hat{L} \cong L$ with $\hat{L} : T_{\mathcal{A}}^{\mathfrak{u}} \to T_{\mathcal{A}}^{\mathfrak{v}}, m_{\hat{L}}^{(x)} \in \mathbb{C}^{|\mathfrak{v}|\times|\mathfrak{u}|}, x \in \mathbb{L}(\mathcal{A})$.

```
Function $\hat{L} = \texttt{ChangeStructureElement}(L, \mathfrak{u}, \mathfrak{v})$        ▷ See theorems 5.4 and 5.5
    $m_\pi = 0 \in \{0,1\}^{|\mathfrak{s}|\times|\mathfrak{s}|}$, $m_\sigma = 0 \in \{0,1\}^{|\mathfrak{t}|\times|\mathfrak{t}|}$        ▷ Initialize permutation matrices
    for $(i,j) \in \{1,\ldots,|\mathfrak{s}|\}^2$        ▷ Compute changes in $\mathfrak{d}$
        if $\mathcal{A}^{-1}(\mathfrak{s}_i - \mathfrak{u}_j)$ is integral then
            $\mathfrak{e}_i = \mathfrak{s}_i - \mathfrak{u}_j$        ▷ Save shift
            $(m_\pi)_{j,i} = 1$        ▷ Save permutation
    for $(i,j) \in \{1,\ldots,|\mathfrak{t}|\}^2$        ▷ Compute changes in $\mathfrak{c}$
        if $\mathcal{A}^{-1}(\mathfrak{t}_i - \mathfrak{v}_j)$ is integral then
            $\mathfrak{f}_i = \mathfrak{t}_i - \mathfrak{v}_j$        ▷ Save shift
            $(m_\sigma)_{j,i} = 1$        ▷ Save permutation
    $(m_{\hat{L}}^{(y)}) = 0 \in \mathbb{C}^{|\mathfrak{t}|\times|\mathfrak{s}|}$ for all $y \in \mathbb{L}(\mathcal{A})$        ▷ Initialize new multipliers
    for $m_L^{(y)} \neq 0$
        for $(i,j) \in \{1,\ldots,|\mathfrak{t}|\} \times \{1,\ldots,|\mathfrak{s}|\}$
            $(m_{\hat{L}}^{(y-\mathfrak{e}_j+\mathfrak{f}_i)})_{i,j} = (m_L^{(y)})_{i,j}$        ▷ Incorporate shifts $\mathfrak{e}$ and $\mathfrak{f}$, see theorem 5.4
    $m_{\hat{L}}^{(y)} = m_\sigma \cdot m_{\hat{L}}^{(y)} \cdot m_\pi^{-1}$ for all $y \in \mathbb{L}(\mathcal{A})$        ▷ Incorporate permutations, see theorem 5.5
```

Algorithm B.7: Find a least common multiple of two lattices.

Input: Lattice basis $\mathcal{B}, \mathcal{A} \in \mathbb{R}^{n\times n}$

Output: Lattice basis $\mathcal{C}$, s.t. $\mathbb{L}(\mathcal{C}) \subset \mathbb{L}(\mathcal{A})$ and $\mathbb{L}(\mathcal{C}) \subset \mathbb{L}(\mathcal{B})$

```
Function $\mathcal{C} = \texttt{LeastCommonMultiple}(\mathcal{A}, \mathcal{B})$        ▷ See theorem 5.1
    Find an integer $r$, s.t. $M = r\mathcal{A}^{-1}\mathcal{B}$ is integral
    Compute Smith normal form $S = V^{-1}MT^{-1}$ of $M$        ▷ See theorem 2.1
    $(N_{\mathcal{B}})_{i,i} = r \cdot \gcd(r, s_i)^{-1}$, $\mathcal{C} = \mathcal{B}T^{-1}N_{\mathcal{B}}$        ▷ Define the lattice basis
```

References

[1] Ashcroft, N., Mermin, N.: Solid State Physics. Saunders College (1976)

[2] Bolten, M., Rittich, H.: Fourier Analysis of Periodic Stencils in Multigrid Methods. SIAM J. Sci. Comput. **40**(3), A1642–A1668 (2018)

[3] Boonen, T., Van Lent, J., Vandewalle, S.: Local Fourier Analysis of Multigrid for the Curl-Curl Equation. SIAM J. Sci. Comput. **30**(4), 1730–1755 (2008)

[4] Brandt, A.: Multi-Level Adaptive Solutions to Boundary-Value Problems. Math. Comp. **31**(138), 333–390 (1977)

[5] Brandt, A.: Rigorous Quantitative Analysis of Multigrid, I. Constant Coefficients Two-Level Cycle with $L_2 $-Norm. SIAM J. Numer. Anal. **31**(6), 1695–1730 (1994)

[6] Brown, J., Yunhui, H., MacLachlan, S.: Local Fourier analysis of BDDC-like algorithms (2018)

[7] Gaspar, F., Gracia, J., Lisbona, F.: Fourier Analysis for Multigrid Methods on Triangular Grids. SIAM J. Sci. Comput. **31**(3), 2081–2102 (2009)

[8] Grabmeier, J., Kaltofen, E., Weispfenning, V. (eds.): Computer Algebra Handbook. Springer Berlin Heidelberg (2003)

[9] Greenfeld, D., Galun, M., Kimmel, R., Yavneh, I., Basri, R.: Learning to Optimize Multigrid PDE Solvers (2019)

[10] Hiptmair, R.: Multigrid method for maxwell's equations. SIAM J. Numer. Anal. **36**(1), 204–225 (1998)

[11] Huckle, T.: Compact Fourier Analysis for Designing Multigrid Methods. SIAM J. Sci. Comput. **31**(1), 644–666 (2008)

[12] Kahl, K., Kintscher, N.: Geometric Multigrid for the Tight-Binding Hamiltonian of Graphene. SIAM J. Numer. Anal. **56**(1), 499–519 (2018)

[13] Kahl, K., Kintscher, N.: aLFA: automated local Fourier analysis. `https://gitlab.com/NilsKintscher/alfa` (2019)

[14] Kumar, P., Rodrigo, C., Gaspar, F., Oosterlee, C.W.: On local fourier analysis of multigrid methods for pdes with jumping and random coefficients. SIAM J. Sci. Comput. **41**(3), A1385–A1413 (2019)

[15] Kuo, C.C., Levy, B.C.: Two-color fourier analysis of the multigrid method with red-black Gauss-Seidel smoothing. Applied Mathematics and Computation **29**(1), 69–87 (1989)

[16] MacLachlan, S.P., Oosterlee, C.W.: Local Fourier Analysis for Multigrid with Overlapping Smoothers Applied to Systems of PDEs. Numer. Linear Algebra Appl. **18**(4), 751–774 (2011)

[17] Molenaar, J.: A two-grid analysis of the combination of mixed finite elements and Vanka-type relaxation. In: W. Hackbusch, U. Trottenberg (eds.) Multigrid Methods III, pp. 313–323. Birkhäuser Basel (1991)

[18] Rittich, H.: Extending and Automating Fourier Analysis for Multigrid Methods. PhD Thesis, University of Wuppertal (2017)

[19] Rittich, H.: LFA Lab. `https://github.com/hrittich/lfa-lab` (2018)

[20] Rodrigo, C., Gaspar, F.J., Zikatanov, L.T.: On the validity of the local Fourier analysis. J. Comput. Math. **37**(3), 340–348 (2019)

[21] Rodrigo, C., Salinas, P., Gaspar, F., Lisbona, F.: Local Fourier Analysis for Cell-Centered Multigrid Methods on Triangular Grids. J. Comput. Appl. Math **259**, 35–47 (2014). Proceedings of the Sixteenth International Congress on Computational and Applied Mathematics (ICCAM-2012), Ghent, Belgium, 9-13 July, 2012

[22] Rodrigo, C., Sanz, F., Gaspar, F.J., Lisbona, F.J.: Local Fourier Analysis for Edge-Based Discretizations on Triangular Grids. Numer. Math. Theory Me. **8**(1), 78–96 (2015)

[23] Saad, Y.: Iterative Methods for Sparse Linear Systems, second edn. Society for Industrial and Applied Mathematics (2003)

[24] Schrijver, A.: Theory of Linear and Integer Programming. John Wiley & Sons, Inc. (1986)

[25] Sivaloganathan, S.: The use of local mode analysis in the design and comparison of multigrid methods. Comput. Phys. Commun. **65**(1), 246–252 (1991)

[26] Stein, E.M., Weiss, G.: Introduction to Fourier Analysis on Euclidean Spaces. Princeton University Press (1971)

[27] Stevenson, R.: On the Validity of Local Mode Analysis of Multi-Grid Methods. PhD Thesis, Utrecht University (1990-12)

[28] The PARI Group, Univ. Bordeaux: PARI/GP version `2.9.5` (2018). Available from `http://pari.math.u-bordeaux.fr/`

[29] Trottenberg, U., Oosterlee, C.W., Schüller, A.: Multigrid, *Texts in Applied Mathematics. Bd.*, vol. 33. Academic Press (2001). With contributions by A. Brandt, P. Oswald and K. Stüben

[30] Wienands, R., Joppich, W.: Practical Fourier Analysis for Multigrid Methods. CRC press (2004)

[31] Wienands, R., Oosterlee, C.: On Three-Grid Fourier Analysis for Multigrid. SIAM J. Sci. Comput. **23**(2), 651–671 (2001)

[32] Zhou, G., Fulton, S.: Fourier Analysis of Multigrid Methods on Hexagonal Grids. SIAM J. Sci. Comput. **31**(2), 1518–1538 (2009)