UCLA
UCLA Previously Published Works

Title

Solution of the Schrödinger equation for quasi-one-dimensional materials using helical waves

Permalink

Authors

Agarwal, Shivang
Banerjee, Amartya S

Publication Date

2024

DOI

Copyright Information

Peer reviewed

Solution of the Schrödinger equation for quasi-one-dimensional materials using helical waves

Shivang Agarwal[a,], Amartya S. Banerjee[b]

[a]*Department of Electrical and Computer Engineering, University of California, Los Angeles, CA 90095, U.S.A*
[b]*Department of Materials Science and Engineering, University of California, Los Angeles, CA 90095, U.S.A*

Abstract

We formulate and implement a spectral method for solving the Schrödinger equation, as it applies to quasi-one-dimensional materials and structures. This allows for computation of the electronic structure of important technological materials such as nanotubes (of arbitrary chirality), nanowires, nanoribbons, chiral nanoassemblies, nanosprings and nanocoils, in an accurate, efficient and systematic manner. Our work is motivated by the observation that one of the most successful methods for carrying out electronic structure calculations of bulk/crystalline systems — the plane-wave method — is a spectral method based on eigenfunction expansion. Our scheme avoids computationally onerous approximations involving periodic supercells often employed in conventional plane-wave calculations of quasi-one-dimensional materials, and also overcomes several limitations of other discretization strategies, e.g., those based on finite differences and atomic orbitals. The basis functions in our method — called *helical waves* (or *twisted waves*) — are eigenfunctions of the Laplacian with symmetry adapted boundary conditions, and are expressible in terms of plane waves and Bessel functions in *helical coordinates*.

We describe the setup of fast transforms to carry out discretization of the governing equations using our basis set, and the use of matrix-free iterative diagonalization to obtain the electronic eigenstates. Miscellaneous computational details, including the choice of eigensolvers, use of a preconditioning scheme, evaluation of oscillatory radial integrals and the imposition of a kinetic energy cutoff are discussed. We have implemented these strategies into a computational package called HelicES (Helical Electronic Structure). We demonstrate the utility of our method in carrying out systematic electronic structure calculations of various quasi-one-dimensional materials through numerous examples involving nanotubes, nanoribbons and nanowires. We also explore the convergence properties of our method, and assess its accuracy and computational efficiency by comparison against reference finite difference, transfer matrix method and plane-wave results. We anticipate that our method will find applications in computational nanomechanics and multiscale modeling, for carrying out transport calculations of interest to the field of semiconductor devices, and for the discovery of novel chiral phases of matter that are of relevance to the burgeoning quantum hardware industry.

Keywords: Helical Waves, Electronic Structure Calculations, Nanomaterials, Nanostructures, Chiral Materials, Spectral Method.

1. Introduction

Low dimensional materials have been intensely investigated in the past few decades due to their remarkable electronic, optical, transport and mechanical characteristics [1, 2]. The properties of these materials often provide sharp contrasts with the bulk phase, and have led to various technological applications, including e.g., new kinds of sensors, actuators and energy harvesting devices [3–8]. Quasi-one-dimensional materials — which include nanotubes, nanoribbons, nanowires, nanocoils, as well as miscellaneous structures of biological origin [9, 10] — are particularly interesting in this regard. This is due to the unique electronic properties that emerge as a result of the availability of a single extended spatial dimension in these structures [11–14], the possibility that they are associated with ferromagnetism, ferroelectricity, and superconductivity [15–18], and the fact that the behavior of these materials may be readily modulated via imposition of mechanical deformation modes such as torsion and/or stretching. [19–21]. Quasi-one-dimensional materials have also been investigated as hardware components for computing platforms — both conventional [22, 23] and quantum [24]. The applications of such materials in the latter case are connected to anomalous transport (the Chiral Induced Spin Selectivity effect [25]) and exotic electronic states [26] that can be observed in such systems.

Given the importance of quasi-one-dimensional materials, it is highly desirable to have available computational methods that can efficiently characterize the unique electronic properties of these systems. However, conventional electronic structure calculation methods — based e.g. on plane-waves [27, 28] — are generally inadequate in handling them. This is a result of the non-periodic symmetries in the atomic arrangements of such materials. As a result of these symmetries, the single particle Schrödinger equation associated with the electronic structure problem exhibits special invariances [29, 30], which plane-waves, being intrinsically periodic, are unable to handle. For example, ground state plane-wave calculations of a twisted nanoribbon (see Fig. 1a) will usually involve making the system artificially periodic along the direction of the twist axis — thus resulting in a periodic supercell containing a very large number of atoms, as well as the inclusion of a substantial amount of vacuum padding in the directions orthogonal to the twist axis, so as to minimize interactions between periodic images. Together, these conditions can make such calculations extremely challenging even on high performance computing platforms, if not altogether impractical. There have been a few attempts to treat quasi-one-dimensional materials using Linear Combination of Atomic Orbitals (LCAO) based techniques [31–36]. However, such methods suffer from basis incompleteness and superposition errors [37–39], which can make it difficult to obtain systematically convergent and improvable results.

In view of these limitations of conventional methods, a series of recent contributions has explored the use of real space techniques to study quasi-one-dimensional materials and their natural deformation modes [20, 30, 40, 41]. Specifically, this line of work incorporates the helical interaction potentials present in such systems using *helical Bloch waves* and employs higher order finite differences to discretize the single particle Schrödinger equation in *helical coordinates*. While this technique shows systematic convergence, and has enabled the exploration of various fascinating electromechanical properties, it also has a number of significant drawbacks. First, due to the curvilinearity of helical coordinates,

---

Email address: asbanerjee@ucla.edu (Amartya S. Banerjee)

2

the discretized Hamiltonian appearing in these calculations is necessarily non-Hermitian [42, 43]. This complicates the process of numerical diagonalization and makes many of the standard iterative eigensolvers [44] unusable. Second, the discretized equations have a coordinate singularity along the system axis which restricts the use of the methods to tubular structures and prevents important nanomaterials such as nanowires and nanoribbons from being studied. The presence of the singularity also tends to ill condition the discretized Hamiltonian, which further restricts the applicability of the method to systems in which the atoms lie far enough away from the system axis (e.g. larger diameter nanotubes). Finally, while the finite difference approach does allow for the simulation of materials with twist (intrinsic or applied), the sparsity pattern of the discretized Hamiltonian worsens upon inclusion of twist, making simulations of such systems significantly more burdensome.

In this work we formulate and implement a novel computational technique that remedies all of the above issues and allows one to carry out systematic numerical solutions of the Schrödinger equation, as it applies to quasi-one-dimensional materials and structures. The technique presented here can be thought of as an analog of the classical plane-wave method, and is similar in spirit to the spectral scheme for clusters presented in [45]. Like the classical plane-wave method, a single parameter (the kinetic energy cutoff) dictates the overall quality of solution of our numerical scheme. We present a derivation of the basis functions of our method — called *helical waves* (or *twisted waves*) — as eigenfunctions of the Laplacian under suitable boundary conditions. We describe how helical waves may be used to discretize the symmetry adapted Schrödinger equation for quasi-one-dimensional materials, and how matrix-free iterative techniques can be used for diagonalization. A key feature of our technique is the handling of convolution sums through the use of fast basis transforms, and we describe in detail how these transforms are formulated and implemented. We also discuss various other computational aspects, including the choice of eigensolvers and preconditioners, and the handling of oscillatory radial integrals that appear in our method. We have implemented these techniques into a MATLAB [46] package called HelicES (**Helic**al **E**lectronic **S**tructure), which we use for carrying out demonstrative electronic structure calculations of various quasi-one-dimensional materials. We also present results related to the convergence, computational efficiency and accuracy properties of our method, while using finite difference, transfer matrix and plane-wave methods for reference data.

We remark that our technique has connections with methods presented in earlier work concerning electronic structure calculations in cylindrical geometries [47–51], but is more general in that the use of helical waves automatically allows both chiral (i.e., twisted) and achiral (i.e., untwisted) structures to be naturally handled. Additionally, some of these earlier studies have employed the strategy of setting up of the discretized Hamiltonian explicitly and then using direct diagonalization techniques, which scales in a significantly worse way (both in memory and computational time) compared to the transform based matrix-free strategies adopted by us. We also note in passing that the basis functions presented here appear to be scalar versions of twisted wave fields explored recently in the x-ray crystallography [52, 53] and elastodynamics [54, 55] literature.

The rest of this paper is organized as follows. In Section 2, we specify the class of systems of interest to this work, formalize the relevant computational problem, and describe our discretization strategy. Numerical techniques and algorithms are presented in Section 3, following which we present results in Section 4. We conclude in Section 5 and

3

also discuss the future outlook of the work. Miscellaneous derivations and computational details are presented in the Appendices.

2. Formulation

In what follows, $\mathbf{e_X}$, $\mathbf{e_Y}$, $\mathbf{e_Z}$ will denote the standard orthonormal basis of $\mathbb{R}^3$. Position vectors will be typically denoted using boldface lower case letters (e.g., $\mathbf{p}$) and rotation matrices using boldface uppercase (e.g., $\mathbf{Q}$).The atomic unit system of $m_e = 1, \hbar = 1, \frac{1}{4\pi\epsilon_0} = 1$ will be used throughout the paper, unless otherwise mentioned. Cartesian and cylindrical coordinates will be typically denoted as (x, y, z) and (r, ϑ, z) respectively. The $\times$ sign will be reserved for denoting dimensions of matrices (e.g. using $M \times N$ to denote the dimensions of a matrix with M rows and N columns), while $*$ will be used to explicitly denote multiplication by or in between scalars, vectors and matrices.

2.1. Description of Physical System and Computational Problem

We consider a quasi-one-dimensional nanostructure of infinite extent aligned along $\mathbf{e_Z}$ (see Fig. 1). We assume the structure to be of limited extent along $\mathbf{e_X}$ and $\mathbf{e_Y}$. Let the atoms of the structure have coordinates:

$$\mathcal{S} = \{\mathbf{p}_1, \mathbf{p}_2, \mathbf{p}_3, \cdots : \mathbf{p}_i \in \mathbb{R}^3\}. \tag{1}$$

Quasi-one-dimensional structures in their undeformed states, or while being subjected to natural deformation modes such as extension, compression or torsion, can often be described using helical (i.e., screw transformation) and cyclic symmetries [10, 20, 29, 30]. Accordingly, we may identify a finite subset of atoms of the structure with coordinates:

$$\mathcal{P} = \{\mathbf{r}_1, \mathbf{r}_2, \mathbf{r}_3, \ldots, \mathbf{r}_M : \mathbf{r}_i \in \mathbb{R}^3\}, \tag{2}$$

and a corresponding set of symmetry operations:

$$\mathcal{G} = \left\{\Upsilon_{\zeta,\mu} = \left(\mathbf{R}_{(2\pi\zeta\alpha+\mu\Theta)} | \zeta\tau\mathbf{e_Z}\right) : \zeta \in \mathbb{Z}, \mu = 0, 1, \ldots, \mathfrak{N} - 1\right\}, \tag{3}$$

such that:

$$\mathcal{S} = \bigcup_{\substack{\zeta\in\mathbb{Z} \\ \mu=0,1,\ldots,\mathfrak{N}-1}} \bigcup_{i=1}^{M} \mathbf{R}_{(2\pi\zeta\alpha+\mu\Theta)}\mathbf{r}_i + \zeta\tau\mathbf{e_Z}. \tag{4}$$

Here, the $\Upsilon_{\zeta,\mu}$ are symmetry operations of the structure — specifically, each $\Upsilon_{\zeta,\mu}$ is an isometry whose action on an arbitrary point $\mathbf{x} \in \mathbb{R}^3$ (denoted as $\Upsilon_{\zeta,\mu} \circ \mathbf{x}$) is to rotate it by the angle $2\pi\zeta\alpha + \mu\Theta$ about $\mathbf{e_Z}$, while simultaneously translating it by $\mu\tau$ about the same axis. The natural number $\mathfrak{N}$ is related to cyclic symmetries in the nanostructure about the axis $\mathbf{e_Z}$, with $\Theta = 2\pi/\mathfrak{N}$ denoting the cyclic symmetry angle. The quantity τ is the pitch of the screw transformation part of $\Upsilon_{\zeta,\mu}$, the parameter α takes values $0 \leq \alpha < 1$, and $\beta = 2\pi\alpha/\tau$ captures the rate of twist (imposed or intrinsic) in the structure. The case $\alpha = 0$ usually represents achiral or untwisted structures (see Fig. 1) .

The electronic properties of a quasi-one-dimensional material under study can be investigated by calculating the spectrum of the single particle Schrödinger operator:

$$\mathfrak{H} = -\frac{1}{2}\Delta + V(\mathbf{x}), \tag{5}$$

4

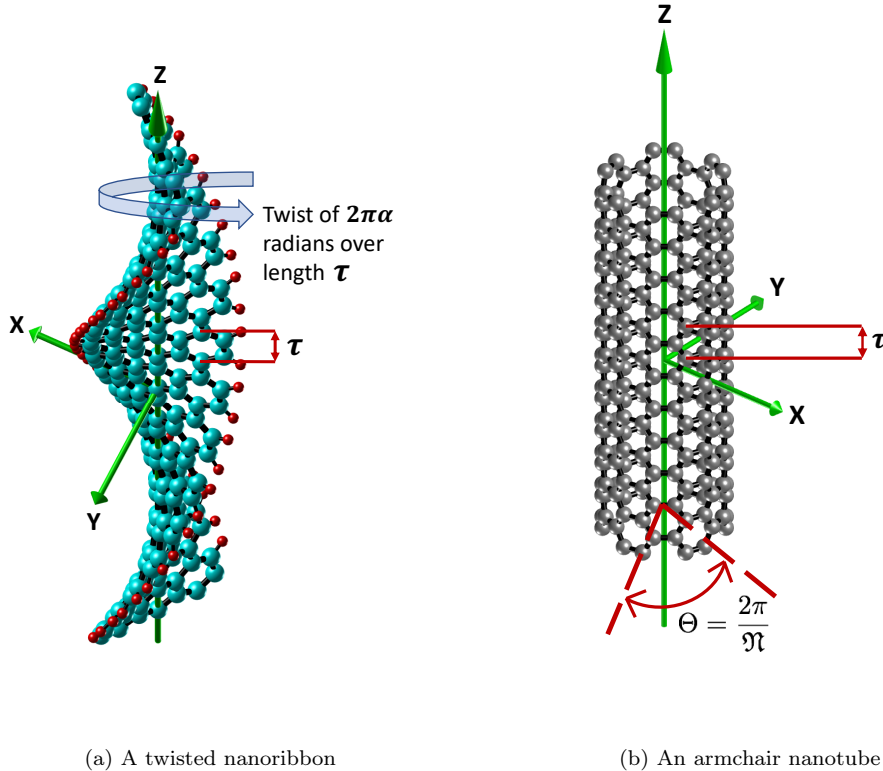(a) A twisted nanoribbon   (b) An armchair nanotube

Figure 1: Examples of the type of nanostructures that can be investigated using the computational framework presented in this work. Helical and cyclic symmetry parameters associated with the geometries of the structures are shown.

associated with the system. Determination of the spectrum in an efficient manner, especially for realistic quasi-one-dimensional nanomaterials serves as the primary computational problem of interest in this work. Here, $V(\mathbf{x})$ represents the "effective potential" as perceived by the electrons. The potential can be computed through self-consistent means (for example, as part of Density Functional Theory calculations [20, 30]), or through the use of empirical pseudopotentials [56, 57], as done here. Due to the presence of global structural symmetries, the potential is expected to obey:

$$V(\mathbf{x}) = V(\Upsilon_{\zeta,\mu} \circ \mathbf{x}), \forall \Upsilon_{\zeta,\mu} \in \mathcal{G}. \tag{6}$$

As a consequence of the quasi-one-dimensional nature of the system, and the above symmetry conditions, the eigenstates of the Hamiltonian can be characterized in terms of *Helical Bloch waves* [29, 30]. Specifically, solutions of the Schrödinger equation:

$$\left(-\frac{1}{2}\Delta + V(\mathbf{x}) \right)\psi = \lambda \psi, \tag{7}$$

can be labeled using band indices $j \in \mathbb{N}$, and symmetry adapted quantum numbers $\eta \in \left[-\frac{1}{2}, \frac{1}{2} \right)$, $\nu \in \{0, 1, 2, \ldots, \mathfrak{N} - 1\}$. Moreover, these solutions obey the following condition for any symmetry operation $\Upsilon_{\zeta,\mu} \in \mathcal{G}$:

$$\psi_j(\Upsilon_{\zeta,\mu} \circ \mathbf{x}; \eta, \nu) = e^{-2\pi i \left(\zeta\eta + \frac{\mu\nu}{\mathfrak{N}} \right)} \psi_j(\mathbf{x}; \eta, \nu). \tag{8}$$

The above relation can be used to reduce the computational problem of determining the eigenstates of the Schrödinger operator over all of space, to a *fundamental domain* or *symmetry-adapted unit cell*.

5

Since the structures considered here have limited spatial extent in the $\mathbf{e_X} - \mathbf{e_Y}$ plane, so does the computational unit cell. We denote the maximum radial coordinate of the points in the computational domain as R. Then, this region of space (see Fig. 3) can be parametrized in cylindrical coordinates as:

$$\mathcal{D} = \left\{ (r, \vartheta, z) : 0 \leq r \leq R, \frac{2\pi\alpha z}{\tau} \leq \vartheta \leq \frac{2\pi\alpha z}{\tau} + \Theta, 0 \leq z \leq \tau \right\}. \tag{9}$$

Due to the decay of the wavefunctions in the radial direction [58, 59], it is often appropriate to enforce Dirichlet boundary conditions on the surface $r = R$, as done here. In practice, the value of R can be chosen so as to ensure a sufficient amount of vacuum exists between the structure under study and this lateral boundary surface [30, 40].

2.2. The Helical Coordinate System and Transformation of Schrödinger's Equation

For computational purposes, it is useful to utilize a coordinate system that describes the computational domain $\mathcal{D}$, and the quasi-one-dimensional system's symmetries more naturally. To this end, we employ *helical coordinates* [29, 60, 61] in this work (Fig. 2). For a point $\mathbf{p} \in \mathbb{R}^3$ with Cartesian coordinates $(x_\mathbf{p}, y_\mathbf{p}, z_\mathbf{p})$, cylindrical coordinates $(r_\mathbf{p}, \vartheta_\mathbf{p}, z_\mathbf{p})$, and helical coordinates $(\theta_{1\,\mathbf{p}}, \theta_{2\,\mathbf{p}}, r_\mathbf{p})$, the following relations hold:

$$r_\mathbf{p} = \sqrt{x_\mathbf{p}^2 + y_\mathbf{p}^2}, \ \theta_{1\,\mathbf{p}} = \frac{z_\mathbf{p}}{\tau},$$
$$\theta_{2\,\mathbf{p}} = \frac{1}{2\pi} \arctan 2 \left(y_\mathbf{p}, x_\mathbf{p} \right) - \alpha \frac{z_\mathbf{p}}{\tau} = \frac{\vartheta_\mathbf{p}}{2\pi} - \alpha \frac{z_\mathbf{p}}{\tau}. \tag{10}$$

Regardless of the amount of twist or cyclic symmetries present in the system, the fundamental domain $\mathcal{D}$ (eq. 9) can be conveniently expressed as a cuboid in helical coordinates, i.e.,

$$\mathcal{D} = \left\{ (\theta_1, \theta_2, r) : 0 \leq \theta_1 \leq 1, 0 \leq \theta_2 \leq \frac{1}{\mathfrak{N}}, 0 \leq r \leq R \right\}. \tag{11}$$

Thus, it is easier to setup a computational mesh over the fundamental domain using helical coordinates. Moreover, the action of the symmetry operations $\Upsilon_{\zeta,\mu} \in \mathcal{G}$ is to simply result in translations of the helical coordinates: if $\mathbf{p} \in \mathbb{R}^3$ has helical coordinates $(\theta_{1\,\mathbf{p}}, \theta_{2\,\mathbf{p}}, r_\mathbf{p})$, then $\Upsilon_{\zeta,\mu} \circ \mathbf{p}$ has helical coordinates $\left(\theta_{1\,\mathbf{p}} + \zeta, \theta_{2\,\mathbf{p}} + \frac{\mu}{\mathfrak{N}}, r_\mathbf{p} \right)$. In particular, this implies that
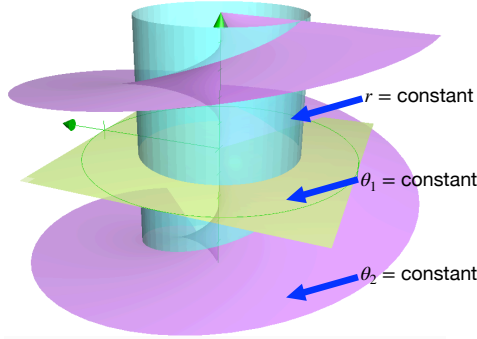

Figure 2: The helical coordinate system represented as constant surfaces of the parameters r, θ_1, θ_2 (the twist parameter α is nonzero here).
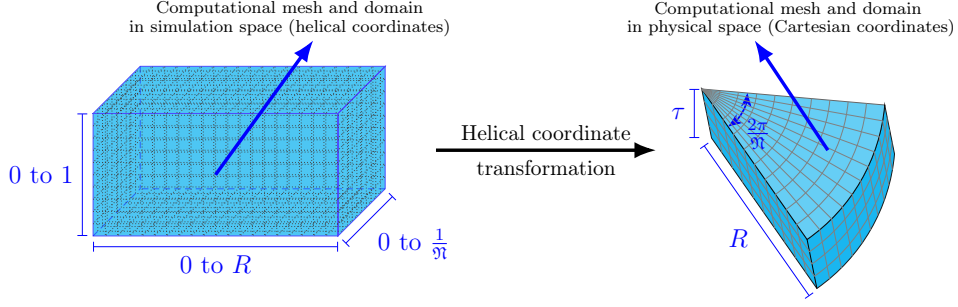
Figure 3: The computational mesh represented in simulation space using helical coordinates (*left*), and physical space using Cartesian coordinates (*right*). The slanted walls of the fundamental domain $\mathcal{D}$ in physical space (*right*) arise due to possibly arbitrary values of twist associated with the system.

a function that is group invariant may be represented over the computational domain by means of periodic boundary conditions along the θ_1 and θ_2 directions. Next, we formulate the governing equations, i.e., Helical-Bloch wave form of Schrodinger's equation over the fundamental domain using helical coordinates. To this end, we first note that:

$$
\begin{aligned}
-\frac{1}{2}\Delta\psi_j + V\psi_j = -\frac{1}{2}\bigg[(\psi_j)_{rr} + \frac{1}{r}(\psi_j)_r + \frac{1}{\tau^2}(\psi_j)_{\theta_1\theta_1} - \frac{2\alpha}{\tau^2}\psi_{\theta_1\theta_2} \\
+ \frac{1}{4\pi^2}\left(\frac{1}{r^2} + \frac{4\pi^2\alpha^2}{\tau^2}\right)(\psi_j)_{\theta_2\theta_2}\bigg] + V\psi_j = \lambda_j\psi_j
\end{aligned}
\tag{12}
$$

Then, we recast eq. 8 to imply that the wavefunctions admit the following Helical Bloch ansatz [20]:

$$
\psi_j(\theta_1,\theta_2,r;\eta,\nu) = e^{-i2\pi(\eta\theta_1+\nu\theta_2)}\phi_j(\theta_1,\theta_2,r;\eta,\nu)\,.
\tag{13}
$$

Here, $\eta \in \left[-\frac{1}{2},\frac{1}{2}\right)$, $\nu \in \{0,1,2,\ldots,\mathfrak{N}-1\}$, and the auxiliary functions $\phi_j(\theta_1,\theta_2,r;\eta,\nu)$ are group invariant. In particular, this implies that these functions obey the conditions:

$$
\begin{aligned}
\phi_j(\theta_1,\theta_2,r;\eta,\nu) &= \phi_j(\theta_1+1,\theta_2,r;\eta,\nu) \\
\phi_j(\theta_1,\theta_2,r;\eta,\nu) &= \phi_j(\theta_1,\theta_2+\frac{1}{\mathfrak{N}},r;\eta,\nu)\,.
\end{aligned}
\tag{14}
$$

Substituting eq. 13 into the Schrödinger equation above (eq. 12) and after some algebra (Appendix A), we arrive at:

$$
\begin{aligned}
\bigg[-\frac{1}{2}\Delta\phi_j - \left(\frac{2\pi^2}{\tau^2}\left\{\nu\alpha(2\eta-\nu\alpha)-\eta^2\right\} - \frac{\nu^2}{2r^2}\right)\phi_j - \frac{2i\pi}{\tau^2}(\nu\alpha-\eta)(\phi_j)_{\theta_1} \\
-2i\pi\left[\frac{\alpha}{\tau^2}(\eta-\nu\alpha) - \frac{\nu}{4\pi^2 r^2}\right](\phi_j)_{\theta_2} + V\phi_j\bigg] = \lambda_j\phi_j\,.
\end{aligned}
\tag{15}
$$

This serves as the governing equation for the computational method in this work. It needs to be discretized and solved over the fundamental domain along with the enforcement of periodic boundary conditions in the θ_1 and θ_2 directions (eq. 14), and the imposition of wavefunction decay in the radial direction, i.e.:

$$
\phi_j(\theta_1,\theta_2,r=R;\eta,\nu) = 0\,.
\tag{16}
$$

Note that due to eq. 6, the effective potential in helical coordinates, $V(\theta_1,\theta_2,r)$, also obeys conditions of the form outlined in eq. 14, although it is generically not expected to obey the decay conditions similar to eq. 16.

7

2.3. Basis Set and Discretization

We now discuss discretization of the governing equations using helical waves. The derivation of these basis functions as symmetry adapted eigenfunctions of the Laplacian is presented in Appendix B. In what follows, we will usually suppress the dependence of $\phi_j(\theta_1, \theta_2, r; \eta, \nu)$ on the band index (j) for the sake of simplicity of notation. Denoting the basis functions as $F_{m,n,k}(\theta_1, \theta_2, r)$ in helical coordinates, we write:

$$\phi(\theta_1, \theta_2, r) = \sum_{(m,n,k) \in \Gamma} \hat{\phi}_{m,n,k} F_{m,n,k}(\theta_1, \theta_2, r)$$

$$= \sum_{(m,n,k) \in \Gamma} \hat{\phi}_{m,n,k} c_{m,n,k} e^{i2\pi(m\theta_1 + n\mathfrak{N}\theta_2)} J_{n\mathfrak{N}}\left(\frac{b_k^{n\mathfrak{N}} r}{R}\right). \tag{17}$$

Here, $\hat{\phi}_{m,n,k}$ are the expansion coefficients, $J_{n\mathfrak{N}}(\cdot)$ denotes Bessel functions of the first kind of order $n\mathfrak{N}$, while $b_k^{n\mathfrak{N}}$ denote the zeros of the Bessel functions. The basis function normalization constants $c_{m,n,k}$ are:

$$c_{m,n,k} = \sqrt{\frac{\mathfrak{N}}{\pi\tau}} \frac{1}{R J_{n\mathfrak{N}+1}\left(b_k^{n\mathfrak{N}}\right)}. \tag{18}$$

The set Γ denotes triplets of integers (m, n, k) such that $m \in [-M_{\max}, M_{\max}]$, $n \in [-N_{\max}, N_{\max}]$ and $k \in [1, K_{\max}]$. The basis set size is $\mathcal{L} = (2M_{\max}+1) * (2N_{\max}+1) * K_{\max}$, i.e., it grows as $\mathcal{O}(M_{\max} N_{\max} K_{\max})$ in terms of the discretization sizes along the θ_1, θ_2, r directions. By design, the basis functions are orthonormal, i.e.:

$$\langle F_{m,n,k}, F_{m',n',k'} \rangle_{\mathsf{L}^2(\mathcal{D})} = \delta_{m,m'} \delta_{n,n'} \delta_{k,k'}, \tag{19}$$

and they satisfy (see Appendix B):

$$-\Delta F_{m,n,k} = \lambda_{m,n,k}^0 F_{m,n,k}, \ \lambda_{m,n,k}^0 = \left(\frac{b_k^{n\mathfrak{N}}}{R}\right)^2 + \left|\frac{2\pi}{\tau}(m - \alpha n\mathfrak{N})\right|^2. \tag{20}$$

The above condition implies that the kinetic energy part of the single particle Schrödinger operator is diagonalized in this basis.

Consistent with the literature, we will refer to the representation of a function in terms of its expansion coefficients (i.e., $\hat{\phi}_{m,n,k}$ in the above) as its *reciprocal space representation*. Furthermore, we will refer to the representation of the function in terms of its values on a discrete set of grid points as its *real space representation*. If the basis functions are also available on these same grid points, the real and reciprocal space representations of the function can be connected via eq. 17. In Appendix C we demonstrate how the gradients of quantities expressed via eq. 17 may be evaluated.

For common systems of interest, the number of basis functions required for discretizing the governing equations can number in the tens or hundreds of thousands (See Section 4). Thus, it can become infeasible to explicitly store the discretized Hamiltonian. This scenario is also encountered in the classical plane-wave method for bulk systems [27, 28], and can be addressed by working with the discretized Hamiltonian implicitly, and using iterative, matrix-free diagonalization techniques to compute the eigenstates [44, 62]. For adopting such strategies, we need to be able to compute action of the Hamiltonian on an

arbitrary vector, such as the wavefunction, as represented in our basis set. To this end, we consider a vector $\phi \in \mathrm{Span}\{F_{m,n,k} : (m,n,k) \in \Gamma\}$, substitute eq. 17 into eq. 15, and use eq. 20, to arrive at:

$$\left(-\frac{1}{2}\Delta + V\right)\phi = \frac{1}{2}\sum_{\Gamma}\hat{\phi}_{m,n,k}\lambda^0_{m,n,k}F_{m,n,k} - a\left(\alpha,\tau,\eta,\nu\right)\phi - b\left(\alpha,\tau,\eta,\nu\right)\phi_{\theta_1}$$

$$-c\left(\alpha,\tau,\eta,\nu\right)\phi_{\theta_2} + \frac{\nu^2}{2r^2}\phi + \frac{i\nu}{2\pi r^2}\phi_{\theta_2} + V\phi = \lambda\phi\,, \qquad (21)$$

which further simplifies to:

$$\frac{1}{2}\sum_{\Gamma}\hat{\phi}_{m,n,k}\lambda^0_{m,n,k}F_{m,n,k} - a\left(\alpha,\tau,\eta,\nu\right)\sum_{\Gamma}\hat{\phi}_{m,n,k}f_{m,n,k}$$

$$- b\left(\alpha,\tau,\eta,\nu\right)\sum_{\Gamma}\hat{\phi}_{m,n,k}\left(i2\pi m\right)F_{m,n,k} - c\left(\alpha,\tau,\eta,\nu\right)\sum_{\Gamma}\hat{\phi}_{m,n,k}\left(i2\pi n\mathfrak{N}\right)F_{m,n,k}$$

$$+ \frac{\nu^2}{2}\sum_{\Gamma}\hat{\phi}_{m,n,k}\frac{F_{m,n,k}}{r^2} + \frac{i\nu}{2\pi}\sum_{\Gamma}\hat{\phi}_{m,n,k}\left(\frac{i2\pi n\mathfrak{N}}{r^2}\right)F_{m,n,k} + V\left(r,\theta_1,\theta_2\right)\sum_{\Gamma}\hat{\phi}_{m,n,k}F_{m,n,k}$$

$$= \lambda\sum_{\Gamma}\hat{\phi}_{m,n,k}F_{m,n,k}\,. \qquad (22)$$

The constants a, b, c in the above are as follows:

$$a\left(\alpha,\tau,\eta,\nu\right) = \frac{2\pi^2}{\tau^2}\left\{\nu\alpha\left(2\eta - \nu\alpha\right) - \eta^2\right\}\,,$$

$$b\left(\alpha,\tau,\eta,\nu\right) = \frac{2i\pi}{\tau^2}\left(\nu\alpha - \eta\right)\,,\ c\left(\alpha,\tau,\eta,\nu\right) = \frac{2i\pi\alpha}{\tau^2}\left(\eta - \nu\alpha\right)\,. \qquad (23)$$

The action of the Hamiltonian on the vector ϕ is simply the left hand side of eq. 22 above. We observe that due to orthonormality of the basis set, the first four terms on the left hand side are easily handled in reciprocal space. Specifically, the second term is simply a scaling of the input vector ϕ with the factor $a\left(\alpha,\tau,\eta,\nu\right)$, while the other three terms can be evaluated as element-wise product operations (Matlab operation $.*$). Thus, these terms can all be evaluated at a cost that scales linearly with the basis set size. The last term on the left hand side is associated with action of the effective potential $V(\mathbf{x})$ on the wavefunction vector. If the expansion coefficients of the potential are available as:

$$V(\theta_1, \theta_2, r) = \sum_{(\tilde{m},\tilde{n},\tilde{k})\in\Gamma}\widehat{V}_{\tilde{m},\tilde{n},\tilde{k}}\,F_{\tilde{m},\tilde{n},\tilde{k}}(\theta_1,\theta_2,r)\,, \qquad (24)$$

then the expansion coefficients of $V(\mathbf{x})\phi(\mathbf{x})$ can be computed as:

$$\left\langle V(\theta_1,\theta_2,r)\,\phi(\theta_1,\theta_2,r)\,,\ F_{m',n',k'}(\theta_1,\theta_2,r)\right\rangle_{\mathsf{L}^2(\mathcal{D})}$$

$$= \left\langle \left(\sum_{\Gamma}\widehat{V}_{\tilde{m},\tilde{n},\tilde{k}}F_{\tilde{m},\tilde{n},\tilde{k}}(\theta_1,\theta_2,r)\right)\left(\sum_{\Gamma}\hat{\phi}_{m,n,k}F_{m,n,k}(\theta_1,\theta_2,r)\right)\,,\ F_{m',n',k'}(\theta_1,\theta_2,r)\right\rangle_{\mathsf{L}^2(\mathcal{D})}$$

$$= \left\langle \left(\sum_{\Gamma}\sum_{\Gamma}\widehat{V}_{\tilde{m},\tilde{n},\tilde{k}}\,\hat{\phi}_{m,n,k}\,F_{\tilde{m},\tilde{n},\tilde{k}}(\theta_1,\theta_2,r)F_{m,n,k}(\theta_1,\theta_2,r)\right)\,,\ F_{m',n',k'}(\theta_1,\theta_2,r)\right\rangle_{\mathsf{L}^2(\mathcal{D})}$$

9

$$= \sum_{\Gamma} \sum_{\Gamma} \hat{V}_{\tilde{m},\tilde{n},\tilde{k}} \, \hat{\phi}_{m,n,k} \left\langle F_{\tilde{m},\tilde{n},\tilde{k}} \, F_{m,n,k} \, , \, F_{m',n',k'} \right\rangle_{\mathsf{L}^2(\mathcal{D})} . \tag{25}$$

There are two problems with the above evaluation strategy. First, the time complexity of the procedure scales in a cubic manner with respect to the basis set size, i.e., $O(M_{\max}^3 N_{\max}^3 K_{\max}^3)$. Moreover, if the coupling coefficients:

$$\left\langle F_{\tilde{m},\tilde{n},\tilde{k}} \, F_{m,n,k} \, , \, F_{m',n',k'} \right\rangle_{\mathsf{L}^2(\mathcal{D})} = c_{\tilde{m},\tilde{n},\tilde{k}} c_{m,n,k} c_{m',n',k'}^* \int_0^1 e^{i2\pi(m+\tilde{m}-m')\theta_1} \, d\theta_1$$

$$\times \int_0^{\frac{1}{\mathfrak{N}}} e^{i2\pi\mathfrak{N}(n+\tilde{n}-n')\theta_2} \, d\theta_2 \int_0^R J_{n\mathfrak{R}}\left(\frac{b_k^{n\mathfrak{N}}r}{R}\right) J_{n'\mathfrak{R}}\left(\frac{b_{k'}^{n'\mathfrak{N}}r}{R}\right) J_{\tilde{n}\mathfrak{R}}\left(\frac{b_{\tilde{k}}^{\tilde{n}\mathfrak{N}}r}{R}\right) 2\pi\tau r \, dr , \tag{26}$$

are to be calculated and stored ahead of time for easier evaluation of eq. 25, the memory complexity of the procedure would also scale cubically with the basis set size. By making use of the fact that the coupling coefficients are non-zero only for $m + \tilde{m} = m'$ and $n + \tilde{n} = n'$, their evaluation, storage and application to eq. 25, can be somewhat simplified. Despite this, the overall complexity still continues to be cubic in the basis set size in both memory and time. Second, the potential $V(\mathbf{x})$ is generally not expected to be equal to zero at $r = R$ and may be slowly decaying due to long range electrostatics effects. Hence, it is inappropriate to express this quantity in terms of helical waves obeying Dirichlet boundary conditions.

Both of the above issues can be remedied by adopting a pseudospectral evaluation strategy [45, 63–66], as we now describe. This is related to the observation that if $V(\mathbf{x})$ and $\phi(\mathbf{x})$ are available in real space, as functions sampled at a common set of grid points, the product $\chi(\mathbf{x}) = V(\mathbf{x})\phi(\mathbf{x})$ can be evaluated with a cost proportional to the size of the grid. Thereafter, the function $\chi(\mathbf{x})$ can be directly expanded in terms of helical waves to yield:

$$\hat{\chi}_{m',n',k'} = \left\langle V(\theta_1,\theta_2,r) \, \phi(\theta_1,\theta_2,r) \, , \, f_{m',n',k'}(\theta_1,\theta_2,r) \right\rangle_{\mathsf{L}^2(\mathcal{D})} . \tag{27}$$

Since $\chi(\mathbf{x})$ obeys Dirichlet boundary conditions and inherits all symmetries of the group $\mathcal{G}$, its expansion using helical waves is appropriate. To put this strategy into practice however, we need access to fast basis transforms so that functions expressed in reciprocal space (i.e., as expansion coefficients) and real space (i.e., on the grid), may be readily interconverted. We describe the formulation and implementation of such transform routines in Sections 3.4.1 and 3.4.2. The overall computational cost of this strategy is the sum total of the costs of the forward and inverse transforms, and the cost of carrying out the real space product. Theoretically, the transforms described here scale in a manner that is slightly worse than the basis set size. However, as we show later, in practice they scale more favorably, in a sub-linear manner (see Fig. 4). Furthermore, the real space grid size is usually a constant multiple of the basis set size, leading to the overall cost of the pseudospectral method scaling in a manner that is close to the first power of this quantity $(= \mathcal{O}(M_{\max}N_{\max}K_{\max}^2))$. The memory complexity is also reduced and scales as the basis set size itself, i.e., $\mathcal{O}(M_{\max}N_{\max}K_{\max})$.

Finally, we discuss the evaluation of the fifth and the sixth terms on the left hand side of eq. 22. The fifth term, i.e.,

$$\ell(\theta_1,\theta_2,r) = \frac{\nu^2}{2} \sum_{\Gamma} \hat{\phi}_{m,n,k} \frac{F_{m,n,k}}{r^2} , \tag{28}$$

10

satisfies $\ell(\theta_1, \theta_2, r = R) = 0$ and invariance under $\mathcal{G}$, since it is a finite linear combination of terms which individually obey these conditions. Thus, the expansion coefficients are:

$$
\begin{aligned}
\widehat{\ell}_{m',n',k'} &= \left\langle \ell(\theta_1, \theta_2, r), F_{m',n',k'}(\theta_1, \theta_2, r) \right\rangle_{\mathsf{L}^2(\mathcal{D})} \\
&= \frac{\nu^2}{2} \sum_{\Gamma} \hat{\phi}_{m,n,k} \int_0^1 \int_0^{\frac{1}{\mathfrak{N}}} \int_0^R \frac{F_{m,n,k} F_{m',n',k'}^*}{r^2} 2\pi\tau r \, dr \, d\theta_2 \, d\theta_1 \\
&= \nu^2 \pi\tau \sum_{\Gamma} \hat{\phi}_{m,n,k} c_{m,n,k} c_{m',n',k'}^* \left[\int_0^1 e^{i2\pi(m-m')\theta_1} d\theta_1 \times \right. \\
&\qquad\qquad \left. \int_0^{\frac{1}{\mathfrak{N}}} e^{i2\pi\mathfrak{N}(n-n')\theta_2} d\theta_2 \int_0^R \frac{J_{n\mathfrak{N}}\left(\frac{b_k^{n\mathfrak{N}} r}{R}\right) J_{n'\mathfrak{N}}\left(\frac{b_{k'}^{n'\mathfrak{N}} r}{R}\right)}{r^2} r \, dr \right] \\
&= \frac{\nu^2}{R^2} \sum_{k=1}^{K_{\max}} \hat{\phi}_{m',n',k} \frac{1}{J_{n'\mathfrak{N}+1}\left(b_k^{n'\mathfrak{N}}\right) J_{n'\mathfrak{N}+1}\left(b_{k'}^{n'\mathfrak{N}}\right)} \int_0^R \frac{J_{n'\mathfrak{N}}\left(\frac{b_k^{n'\mathfrak{N}} r}{R}\right) J_{n'\mathfrak{N}}\left(\frac{b_{k'}^{n'\mathfrak{N}} r}{R}\right)}{r^2} r \, dr \, . \quad (29)
\end{aligned}
$$

In the above, we have made use of the orthonormality of the complex exponentials in the θ_1 and θ_2 directions. We may rewrite eq. 29 as:

$$
\widehat{\ell}_{m',n',k'} = \frac{\nu^2}{R^2} \sum_{k=1}^{K_{\max}} \hat{\phi}_{m',n',k} \, \mathcal{I}(n', k, k') \, , \tag{30}
$$

with:

$$
\mathcal{I}(n', k, k') = \frac{1}{J_{n'\mathfrak{N}+1}\left(b_k^{n'\mathfrak{N}}\right) J_{n'\mathfrak{N}+1}\left(b_{k'}^{n'\mathfrak{N}}\right)} \int_0^1 \frac{J_{n'\mathfrak{N}}\left(b_k^{n'\mathfrak{N}} q\right) J_{n'\mathfrak{N}}\left(b_{k'}^{n'\mathfrak{N}} q\right)}{q^2} q \, dq \, . \tag{31}
$$

Thus, if the quantities $\mathcal{I}(n', k, k')$ are known ahead of time, the coefficients $\ell_{m',n',k'}$ can be readily evaluated at a cost of $\mathcal{O}(M_{\max} N_{\max} K_{\max}^2)$, i.e., quite close to the overall basis set size, and similar in computational complexity to the evaluation of the potential term. Since $\mathcal{I}(n', k, k')$ is problem independent (e.g., it has no dependence on R, α, τ or the potential $V(\mathbf{x})$), we may evaluate and store it as a table for a large range of values of n, k and k'. During program execution, this table of values may be loaded into memory, and each $\ell_{m',n',k'}$ can be evaluated as a vector dot product (eq. 30), after accessing the necessary values of $\mathcal{I}(n', k, k')$. As for the evaluation of the $\mathcal{I}(n', k, k')$ values themselves, we may use the recurrence relation [67]:

$$
\frac{2\kappa}{q} J_\kappa(q) = J_{\kappa-1}(q) + J_{\kappa+1}(q) \, , \tag{32}
$$

to rid the integrand in eq. 31 of it's denominator, and obtain a pair of oscillatory integrals. We may then compute these by using Gauss-Jacobi quadrature as outlined in eq. Appendix D.

In an analogous manner, the sixth term on the left hand side of eq. 22, i.e.,

$$
\frac{i\nu}{2\pi} \sum_{\Gamma} \hat{\phi}_{m,n,k} \left(i2\pi n\mathfrak{N}\right) \int_0^1 \int_0^{\frac{1}{\mathfrak{N}}} \int_0^R \frac{F_{m,n,k} F_{m',n',k}^*}{r^2} 2\pi\tau r \, dr \, d\theta_2 \, d\theta_1 \, , \tag{33}
$$

11

can be simplified to:

$$= \frac{i\nu}{\pi R^2} \sum_{k=1}^{K_{\max}} \left(\hat{\phi}_{m',n',k} * i2\pi n' \mathfrak{N} \right) \mathcal{I}(n',k,k') . \tag{34}$$

With the quantities $\mathcal{I}(n',k,k')$ available, the above can be evaluated in a manner similar to the evaluation of the fifth term, at a computational cost of $\mathcal{O}(M_{\max}N_{\max}K_{\max}^2)$. The key difference is that instead of the vector $\{\hat{\phi}_{m,n,k}\}_{(m,n,k)\in\Gamma}$, we need to consider an alternate one with entries $\{i2\pi n' \mathfrak{N}\hat{\phi}_{m,n,k}\}_{(m,n,k)\in\Gamma}$. However, this modified vector is already available as part of evaluation of the fourth therm on the left hand side of eq. 22, and therefore, it can be reused.

3. Numerical Implementation

We have implemented the above computational strategies into a MATLAB [46] package called HelicES (**Helic**al **E**lectronic **S**tructure). To ensure efficiency, our code heavily relies on vectorization features of MATLAB. Various details of our implementation are as follows.

3.1. Wave function storage: reciprocal and real space considerations

For any quantity in reciprocal space, there are three indices m, n, k associated with each expansion coefficient, making the collection of coefficients a three-dimensional object of dimensions $(2M_{\max}+1)\times(2N_{\max}+1)\times K_{\max} = \mathcal{L}$. However, it is easier for linear algebra operations to have these coefficients stacked up as vector in $\mathbb{C}^{\mathcal{L}}$. To achieve this, we adopt the following mapping between $(m,n,k) \in \Gamma$ and the linear index $\mathfrak{i} \in \{1,2,\ldots,\mathcal{L}\}$:

$$\mathfrak{i}(m,n,k) = (m + M_{\max}) * (2N_{\max} + 1) * K_{\max} + (n + N_{\max}) * K_{\max} + k . \tag{35}$$

With this, a collection of N_s wavefunctions can be stored as a complex matrix of dimensions $\mathcal{L} \times N_s$.

For real space representation, the number of grid points to be chosen along each helical coordinate θ_1, θ_2 is dictated by the accuracy of the basis transforms (see Section 3.4). We choose to work with Fourier nodes along the θ_1 and θ_2 directions and denote the corresponding number of grid points as N_{θ_1} and N_{θ_2} respectively. Along the radial direction, we choose N_r Gauss-Jacobi nodes [68] over the interval $[0, R]$. This has the advantage that the coordinate singularity at the origin is automatically avoided. In order to accommodate non-linearities and to reduce aliasing errors [45, 69], we typically choose $N_{\theta_1} = 4 * M_{\max} + 1$, $N_{\theta_2} = 4 * N_{\max} + 1$ and $N_r = 4 * K_{\max}$. These choices generally allow transforms to be evaluated accurately up to machine precision. With this setup, quantities such as the wavefunction are available in real space over a three-dimensional grid (i.e., the tensor product grid resulting from the one-dimensional grids along the individual coordinate directions), and each grid point is indexed via $\mathfrak{i} \in \{1, 2, \ldots, N_{\theta_1}\}$, $\mathfrak{j} \in \{1, 2, \ldots, N_{\theta_2}\}$ and $\mathfrak{k} \in \{1, 2, \ldots, N_r\}$. For storage, we stack this three dimensional representation into a complex column vector of size $N_{\theta_1} * N_{\theta_2} * N_r$, for which we use the following ordering:

$$\mathfrak{j}(\mathfrak{i},\mathfrak{j},\mathfrak{k}) = (\mathfrak{i} - 1) * N_r * N_{\theta_2} + (\mathfrak{j} - 1) * N_r + \mathfrak{k} . \tag{36}$$

Since the memory requirement for storage of each wavefunction in real space is much higher than storing it in reciprocal space, we typically avoid storing real space versions of all N_s wave functions simultaneously.

3.2. Imposition of kinetic energy cutoff

In conventional plane-wave calculations, it is common to specify a *kinetic energy cutoff*, i.e., a limit on the H^1 Sobolev norm of the plane-waves to be used for discretization [28, 70]. Once a suitable periodic unit cell has been identified, this criterion automatically provides a recipe for calculating the number of planewaves along each of the Cartesian axes, and in turn, the dimensions of the underlying real space grid to be used for Fast Fourier Transforms (FFTs). In a similar vein, we may wish to retain only helical waves with kinetic energies below a pre-specified cutoff in our calculation, since this has the advantage that the basis set limits $M_{\max}$, $N_{\max}$, and $K_{\max}$ get specified automatically in proportion to the computational domain's geometry parameters. At the gamma point $(\eta = 0, \nu = 0)$ for example, the kinetic energy cutoff criterion requires that all helical waves $f_{m,n,k}$, with m, n, k values satisfying:

$$\frac{1}{2}\lambda_{m,n,k}^0 = \frac{1}{2}\left[\left(\frac{b_k^{n\mathfrak{N}}}{R}\right)^2 + \left|\frac{2\pi}{\tau}\left(m - \alpha n\mathfrak{N}\right)\right|^2 \right] \leq E_{\text{cut}}, \tag{37}$$

be included in our calculations. In our implementation, we first determine the largest absolute values of integers m, n and the largest natural number k consistent with with eq. 37. We set the basis set limits $M_{\max}$, $N_{\max}$, and $K_{\max}$ accordingly. The real space grids used for carrying out fast transforms (described below) are chosen based on these quantities. Within these $(2M_{\max} + 1) * (2N_{\max} + 1) * K_{\max}$ helical waves, however, not every combination of m, n, k would satisfy the kinetic energy criterion. To remedy this, we create a *masking vector* to exclusively retain helical waves which satisfy eq. 37, in various operations of interest (such as the Hamiltonian times wavefunction products). Based on the linear ordering for reciprocal space storage outlined in eq. 35, we may express the masking vector as:

$$\mathcal{M}_{\text{i}(m,n,k)} = 1, \text{ for } \frac{1}{2}\lambda_{m,n,k}^0 \leq E_{\text{cut}}$$
$$= 0, \text{ otherwise.} \tag{38}$$

Element-wise multiplication of a given vector with the masking vector results in only kinetic energy limited helical waves being retained in the calculation. We implement the above strategy at each η, ν point (with the expression for the kinetic energy modified appropriately) to impose the kinetic energy cutoff in HelicES.

3.3. η-space discretization and parallelization

As described earlier, to obtain the helical Bloch states (eq. 8), i.e., solutions to the single electron problem with a symmetry adapted potential (eq. 6), the single electron Hamiltonian has to be diagonalized for every $\eta \in [-\frac{1}{2}, \frac{1}{2})$ and $\nu \in \{0, 1, 2, \ldots, \mathfrak{N} - 1\}$. To make this calculation feasible, we sample η over a discrete set $\{\eta_b\}_{b=1}^{N_\eta} \subset [-\frac{1}{2}, \frac{1}{2})$. The specific choice of the values η_b is based on the Monkhorst-Pack scheme [71]. This procedure akin to "k-point sampling" in conventional periodic codes [27]. With this choice, the Hamiltonian needs to be diagonalized at $N_{\mathcal{K}} = N_\eta \times \mathfrak{N}$ points, and integrals in η can be calculated via quadrature:

$$\int_{-\frac{1}{2}}^{\frac{1}{2}} p(\eta)\, d\eta \approx \sum_{b=1}^{N_\eta} w_b\, p(\eta_b). \tag{39}$$

Here, $\{w_b\}_{b=1}^{N_\eta}$ are the Monkhorst-Pack quadrature weights and are uniformly equal to $1/N_\eta$. Integrals of the above kind appear, for example, while computing the electronic band energy, or the electron density from helical Bloch states [20, 30].

If the single electron Hamiltonian does not include magnetic fields — as is the case here, time reversal symmetry allows further reduction in the number of η, ν points at which the Hamiltonian has to be diagonalized [40, 72]. Specifically, it holds that for any $\eta \in [-\frac{1}{2}, \frac{1}{2})$ the helical Bloch states and the associated electronic bands obey:

$$\left.\begin{array}{l} \psi_j(\mathbf{x}; \eta, \nu) = \overline{\psi_j(\mathbf{x}; -\eta, \mathfrak{N} - \nu)} \\ \lambda_j(\eta, \nu) = \lambda_j(-\eta, \mathfrak{N} - \nu) \end{array}\right\} \text{ for } \nu \in \{0, 1, 2, \ldots, \mathfrak{N} - 1\}, \tag{40}$$

and:

$$\left.\begin{array}{l} \psi_j(\mathbf{x}; \eta, 0) = \overline{\psi_j(\mathbf{x}; -\eta, 0)} \\ \lambda_j(\eta, 0) = \lambda_j(-\eta, 0) \end{array}\right\} \text{ for } \nu = 0. \tag{41}$$

Overall, this reduces the number of discrete points in reciprocal space by a factor of 2.

Since the diagonalization problem arising from distinct sets of η, ν values are independent of one another, they can be dealt with in an embarrassingly parallel manner. In our implementation, we have used MATLAB's Parallel Computing Toolbox (specifically, the parfor function) to carry out this parallelization.

3.4. Fast basis transforms

Since our strategy for carrying out Hamiltonian matrix-vector products involves fast basis transforms, we now elaborate on various aspects of the implementation of such operations within the HelicES code. To arrive at fast transforms, we exploit the separability of the basis functions into radial and θ_1, θ_2 dependence. This allows us to make use of quadrature along the radial direction, and subsequently, efficient two-dimensional fast Fourier transforms (FFTs) along the θ_1, θ_2 directions for each radial grid point, or for each radial basis function. Since the radial part of the basis functions consists of Bessel functions, we have also investigated the use of Hankel and discrete Bessel transforms [73–75]. However, we found that the quadrature approach adopted here resulted in better performance for the basis set sizes considered, consistent with some earlier studies [76].

In what follows, $O_{M \times N}$ is used to denote a zero matrix of size $M \times N$. The typical real space grid point for sampling a function will be denoted as $(\theta_1^i, \theta_2^j, r^k)$, with $\mathsf{i} \in \{1, 2, \ldots, N_{\theta_1}\}$, $\mathsf{j} \in \{1, 2, \ldots, N_{\theta_2}\}$ and $\mathsf{k} \in \{1, 2, \ldots, N_r\}$. We will use the MATLAB commands ifft2 and fft2 to denote two-dimensional fast inverse and forward Fourier Transforms respectively [46]. Additionally, we will use the MATLAB commands ifftshift and fftshift to denote rearrangements of Fourier transform coefficients related to shifting of zero frequency components to matrix center [46]. Finally, we will use the MATLAB ':' (colon) operator notation to denote array/matrix indices with regular increments. Thus, $\mathsf{i} : \mathsf{k} : \mathsf{j}$ will denote indices starting at i, incremented by k and ending at j, $\mathsf{i} : \mathsf{j}$ will denote the indices $\mathsf{i}, \mathsf{i} + 1, \mathsf{i} + 2, \ldots, \mathsf{j} - 1, \mathsf{j}$, and simply ':' will denote all indices along a particular matrix dimension.

3.4.1. Fast Inverse Basis Transform

Given the expansion coefficients $\{\hat{g}_{m,n,k}\}_{(m,n,k) \in \Gamma}$, a naive way of implementing the inverse basis transform would be to calculate each basis function $\{F_{m,n,k}\}_{(m,n,k) \in \Gamma}$ at

every grid point $(\theta_1^i, \theta_2^j, r^k)$, and to then evaluate the sum:

$$g(\theta_1^i, \theta_2^j, r^k) = \sum_{\Gamma} \hat{g}_{m,n,k} F_{m,n,k}(\theta_1^i, \theta_2^j, r^k) \,. \tag{42}$$

The computational complexity of this "naive inverse transform" is easily seen to be $\mathcal{O}(M_{\max} N_{\theta_1} N_{\max} N_{\theta_2} K_{\max} N_r)$, which simplifies to $\mathcal{O}(M_{\max}^2 N_{\max}^2 K_{\max}^2)$. The constant involved in the latter estimate can be seen to be quite large based on the discussion in Section 3.1. To remedy this situation, we express the basis functions as in Appendix B, i.e., $F_{m,n,k}(\theta_1, \theta_2, r) = e^{i2\pi(m\theta_1 + n\mathfrak{N}\theta_2)} \xi_{n,k}(r)$ and also rewrite the eq. 42 as:

$$
\begin{aligned}
g(\theta_1^i, \theta_2^j, r^k) &= \sum_{m=-M_{\max}}^{M_{\max}} \sum_{n=-N_{\max}}^{N_{\max}} \sum_{k=1}^{K_{\max}} \hat{g}_{m,n,k}\, e^{i2\pi(m\,\theta_1^i + n\mathfrak{N}\,\theta_2^j)}\, \xi_{n,k}(r^k) \\
&= \sum_{m=-M_{\max}}^{M_{\max}} \sum_{n=-N_{\max}}^{N_{\max}} e^{i2\pi(m\,\theta_1^i + n\mathfrak{N}\,\theta_2^j)} \left(\sum_{k=1}^{K_{\max}} \hat{g}_{m,n,k}\, \xi_{n,k}(r^k) \right).
\end{aligned}
\tag{43}
$$

Since the quantity in parentheses is independent of the basis function index k, we may rewrite the above as:

$$
\begin{aligned}
g(\theta_1^i, \theta_2^j, r^k) &= \sum_{m=-M_{\max}}^{M_{\max}} \sum_{n=-N_{\max}}^{N_{\max}} e^{i2\pi(m\,\theta_1^i + n\mathfrak{N}\,\theta_2^j)}\, G_{m,n}(r^k) \\
\text{with}: \quad G_{m,n}(r^k) &= \sum_{k=1}^{K_{\max}} \hat{g}_{m,n,k}\, \xi_{n,k}(r^k) \,.
\end{aligned}
\tag{44}
$$

Thus, if the quantities $G_{m,n}(r^k)$ are known, calculation of the inverse basis transform amounts to computing an inverse two-dimensional fast Fourier transform at each radial grid point r^k. Additionally, we observe that at each radial grid point r^k, $G_{m,n}(r^k)$ can be expressed as a vector dot product between two $K_{\max}$ dimensional vectors, i.e., $\{\hat{g}_{m,n,k}\}_{k=1}^{K_{\mathrm{Max}}}$ and $\{\xi_{n,k}(r^k)\}_{k=1}^{K_{\mathrm{Max}}}$. In fact, by grouping the evaluation of $G_{m,n}(r^k)$ for different grid points together, the above operation may be expressed as the product of a $N_r \times K_{\max}$ matrix with a $K_{\max}$ dimensional vector, which allows for the use of Level-2 BLAS [77] operations. If the radial part of the basis functions (i.e., $\xi_{n,k}(r^k)$) are available ahead of time, the above steps provide a convenient recipe of computing the inverse basis transforms with computational complexity $\mathcal{O}\left(M_{\max} N_{\max} K_{\max}\left(K_{\max} + \log\left(M_{\max}\right) + \log\left(N_{\max}\right)\right)\right)$, a significant improvement over the naive algorithm discussed earlier. We outline the overall steps of our implementation in Algorithms 1 and 2 below, and also illustrate some key aspects through Fig. 5.

In Fig. 4 we compare the naive and fast inverse transforms as implemented in HelicES. The starting vectors $\{\hat{g}_{m,n,k}\}_{(m,n,k)\in\Gamma}$ were randomly chosen for the tests. The results from these two methods always agreed with each other to machine precision, guaranteeing consistency of the implementations. However, consistent with the discussion above, the computational time for the naive transforms is found to scale in a quadratic manner with the basis set size, while for the fast transforms, it is close to being linear. The fact that the observed scaling of our fast transform implementation is actually *sublinear*, is almost certainly related to our use of machine optimized linear algebra and Fourier transform routines as available within MATLAB.
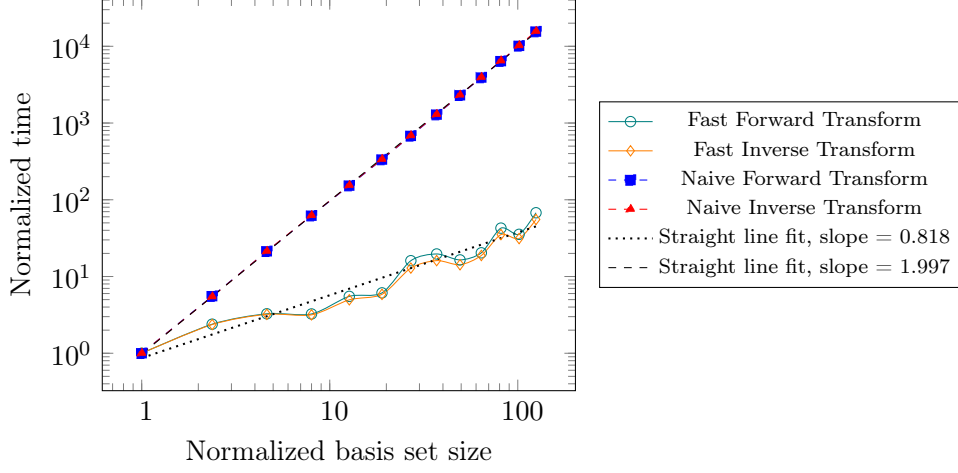
Figure 4: Variation of the normalized time for basis transforms plotted against the basis set size. Both axes are logarithmic. Straight lines were fit using the average of the forward and inverse transform times in each case.

Algorithm 1: Fast Inverse Basis Transform

Input: The vector of expansion coefficients $\{\hat{g}_{m,n,k}\} \in \mathbb{C}^{\mathcal{L}}$

<u>Prerequisite</u>: The radial basis functions sampled on the grid $\{r^k\}_{k=1}^{N_r}$,
i.e., for each integer $n \in [-N_{\max}, N_{\max}]$, the matrix:

$$\mathfrak{R}_n = \begin{pmatrix} \xi_{n,1}(r^1) & \cdots & \xi_{n,K_{\max}}(r^1) \\ \vdots & \ddots & \vdots \\ \xi_{n,1}(r^{N_r}) & \cdots & \xi_{n,K_{\max}}(r^{N_r}) \end{pmatrix}$$

Initialize $g = O_{N_r N_{\theta_1} N_{\theta_2} \times 1}$, $G = O_{(2M_{\max}+1)(2N_{\max}+1)N_r \times 1}$
Initialize $i, j, p = 1$
for $m, n \in \Gamma$ **do**
 | Calculate $G(i : i + N_r - 1) = \mathfrak{R}_n * \hat{g}_{m,n,k}(j : j + K_{\max} - 1)$
 | $i = i + N_r$
 | $j = j + K_{\max}$
end
for $p \leq N_r$ **do**
 | Set $\hat{v} = G(p : N_r : \text{end})$
 | $g(p : N_r : \text{end}) = \text{AngularInverseTransform}(\hat{v})$
end
Result: The inverse basis transform $g(\theta_1, \theta_2, r)$ of the vector $\{\hat{g}_{m,n,k}\}$

Algorithm 2: AngularInverseTransform
(Fast Angular 2D Inverse Fourier Transform)

Input: Vector $\hat{v} \in \mathbb{C}^{(2M_{\max}+1)*(2N_{\max}+1)}$
Initialize $\mathcal{W} = O_{N_{\theta_1} \times N_{\theta_2}}$ // Note: $N_{\theta_1} = 4M_{\max} + 1$, $N_{\theta_2} = 4N_{\max} + 1$
Reshape $\hat{v}$ into a matrix $\mathcal{V} \in \mathbb{C}^{(2N_{\max}+1) \times (2M_{\max}+1)}$
$\mathcal{W}(N_{\max} + 1 : 3N_{\max} + 1, \ M_{\max} + 1 : 3M_{\max} + 1) = \mathcal{V}$
$\mathcal{U} = \text{ifft2}(\text{ifftshift}(\mathcal{W}))$
Scale $\mathcal{U} = \mathcal{U} * (N_{\theta_1} * N_{\theta_2})$
Reshape $\mathcal{U}$ into a column vector $v \in \mathbb{C}^{N_{\theta_1} * N_{\theta_2}}$
Result: The angular inverse Fourier transform $v(\theta_1, \theta_2)$ of the vector $\hat{v}$
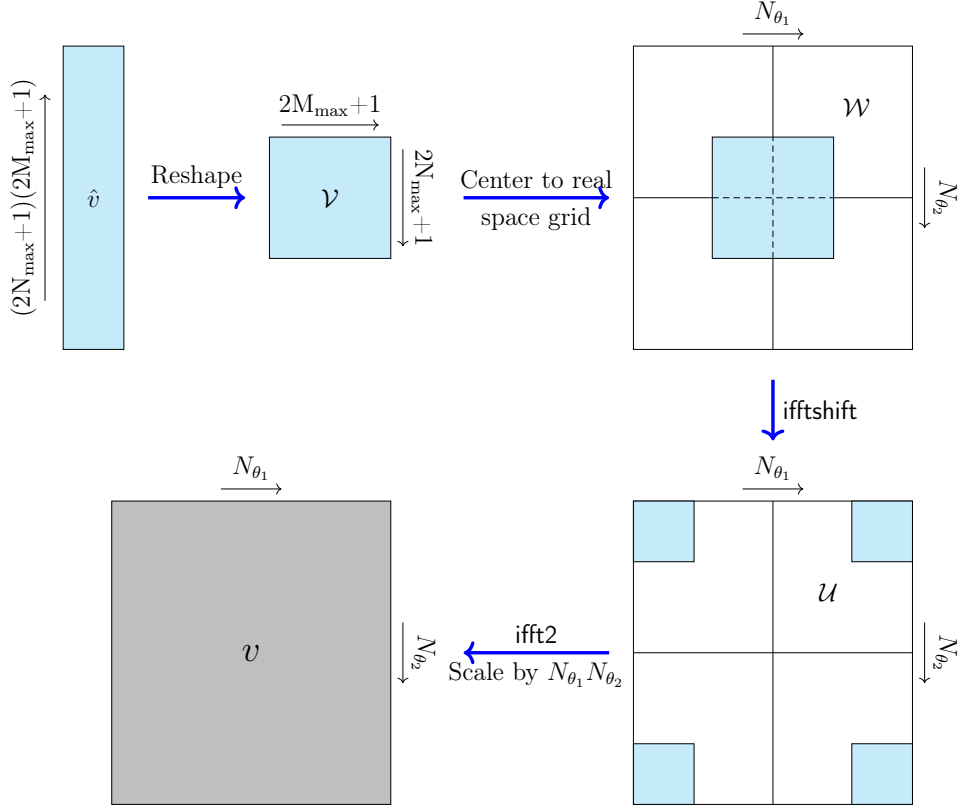
16

Figure 5: Pictorial representation of the workings of the fast angular 2D inverse Fourier transform (Algorithm 2)

3.4.2. Fast Forward Basis Transform

We now discuss the implementation of forward basis transforms within HelicES. Given a function $g(\theta_1, \theta_2, r)$, the forward basis transform is defined as:

$$\hat{g}_{m,n,k} = \langle g, F_{m,n,k} \rangle_{\mathsf{L}^2(\mathcal{D})} = \int_0^1 \int_0^{\frac{1}{\Re}} \int_0^R g(\theta_1, \theta_2, r) \, F^*_{m,n,k}(\theta_1, \theta_2, r) \, 2\pi\tau r \, dr \, d\theta_2 \, d\theta_1 \,. \quad (45)$$

With $F_{m,n,k}$ and g both sampled on the real space grid, this can be approximated via quadrature as:

$$\hat{g}_{m,n,k} \approx 2\pi\tau * \left(\sum_{i=1}^{N_{\theta_1}} \sum_{j=1}^{N_{\theta_2}} \sum_{k=1}^{N_r} g(\theta_1^i, \theta_2^j, r^k) e^{-i2\pi(m\,\theta_1^i + n\Re\theta_2^j)} \, \xi_{n,k}(r^k) \, \omega_r^k \, \omega_{\theta_1}^i \, \omega_{\theta_2}^j \right). \quad (46)$$

Here, the quadrature weights along the θ_1, θ_2 directions are constants, i.e., $\omega_{\theta_1}^i = 1/N_{\theta_1}$ and $\omega_{\theta_2}^j = 1/(\Re N_{\theta_2})$, due to the use of Fourier nodes (i.e., trapezoidal rule). The radial weights $\{\omega_r^k\}_{k=1}^{N_r}$ correspond to Gauss-Jacobi quadrature. We can see that like the case of the inverse transforms, a naive implementation of the above expression will lead to a

computational complexity of $\mathcal{O}(M_{\max}^2 N_{\max}^2 K_{\max}^2)$. Instead, we deal with the evaluation of this expression along the θ_1, θ_2 directions simultaneously at each radial grid point using 2D FFTs, and then perform quadrature in the radial direction. Thus, we compute:

$$H_{m,n}(r^{\mathsf{k}}) = \frac{1}{N_{\theta_1} N_{\theta_2}} * \left(\sum_{i=1}^{N_{\theta_1}} \sum_{j=1}^{N_{\theta_2}} g(\theta_1^i, \theta_2^j, r^{\mathsf{k}}) e^{-i2\pi(m\,\theta_1^i + n\mathfrak{N}\,\theta_2^j)} \right), \qquad (47)$$

followed by:

$$\hat{g}_{m,n,k} = \frac{2\pi\tau}{\mathfrak{N}} * \left(\sum_{k=1}^{N_r} H_{m,n}(r^{\mathsf{k}}) \, \xi_{n,k}(r^{\mathsf{k}}) \, \omega_r^{\mathsf{k}} \right). \qquad (48)$$

The radial quadratures in the above expression can be conveniently cast in terms of Level-2 BLAS [77] operations if the radial basis functions scaled by the corresponding quadrature weights (i.e. $\{\omega_r^{\mathsf{k}} \xi_{n,k}(r^{\mathsf{k}})\}_{k=1}^{K_{\mathrm{Max}}}$) are available as a matrix ahead of time. We outline the steps of our implementation in Algorithms 3 and 4 below, and illustrate key aspects in Figure 6.

Algorithm 3: Fast Forward Basis Transform

Input: Real space representation of function $\{g(\theta_1^i, \theta_2^j, r^{\mathsf{k}})\} \in \mathbb{C}^{N_r N_{\theta_1} N_{\theta_2}}$
Prerequisite: The radial basis functions sampled on the grid $\{r^{\mathsf{k}}\}_{k=1}^{N_r}$, scaled by the corresponding quadrature weights i.e., for each integer $n \in [-N_{\max}, N_{\max}]$, the following matrix:
$$\mathfrak{O}_n = \begin{pmatrix} \omega_r^1 \xi_{n,1}(r^1) & \cdots & \omega_r^{N_r} \xi_{n,1}(r^{N_r}) \\ \vdots & \ddots & \vdots \\ \omega_r^1 \xi_{n,K_{\max}}(r^1) & \cdots & \omega_r^{N_r} \xi_{n,K_{\max}}(r^{N_r}) \end{pmatrix}$$
‒ ‒
Initialize $\hat{g} = O_{\mathcal{L} \times 1}$, $H = O_{N_r \times (2M_{\max}+1)(2N_{\max}+1)}$
Initialize $i, j, p = 1$
for $p \leq N_r$ **do**
$\quad$ Set $u = g(p : N_r : \mathrm{end})$
$\quad$ $H(p, :) = \mathrm{AngularForwardTransform}(u)$
end
for $m, n \in \Gamma$ **do**
$\quad$ Calculate $\hat{g}(i : i + K_{\max} - 1) = \mathfrak{O}_n * H(:, j)$
$\quad$ $i = i + K_{\max}$
$\quad$ $j = j + 1$
end
Scale $\hat{g} = (2\pi\tau/\mathfrak{N}) * \hat{g}$
Result: The forward basis transform $\{\hat{g}_{m,n,k}\}$ of the function $g(r, \theta_1, \theta_2)$

Algorithm 4: AngularForwardTransform
(Fast Angular 2D Forward Fourier Transform)

Input: Vector containing real space representation of 2D angular function on Fourier grid, i.e., $\{u(\theta_1^i, \theta_2^j)\} \in \mathbb{C}^{N_{\theta_1} N_{\theta_2}}$
Reshape u into a matrix $\mathcal{W} \in \mathbb{C}^{N_{\theta_2} \times N_{\theta_1}}$
$\mathcal{V} = \mathsf{fft2}(\mathcal{W})$
Scale $\mathcal{V} = (\frac{1}{N_{\theta_1} N_{\theta_2}}) * \mathcal{V}$
$\mathcal{U} = \mathsf{fftshift}(\mathcal{V})$
$\hat{u} = \mathcal{U}(N_{\max} + 1 : 3N_{\max} + 1, \; M_{\max} + 1 : 3M_{\max} + 1)$
Result: The 2D angular Fourier transform $\{\hat{u}_{m,n}\}$ of the function $u(\theta_1, \theta_2)$

Referring to Fig. 4, we see that like the case of the fast inverse basis transforms, our implementation of the fast forward basis transforms scale in a *sublinear* manner with
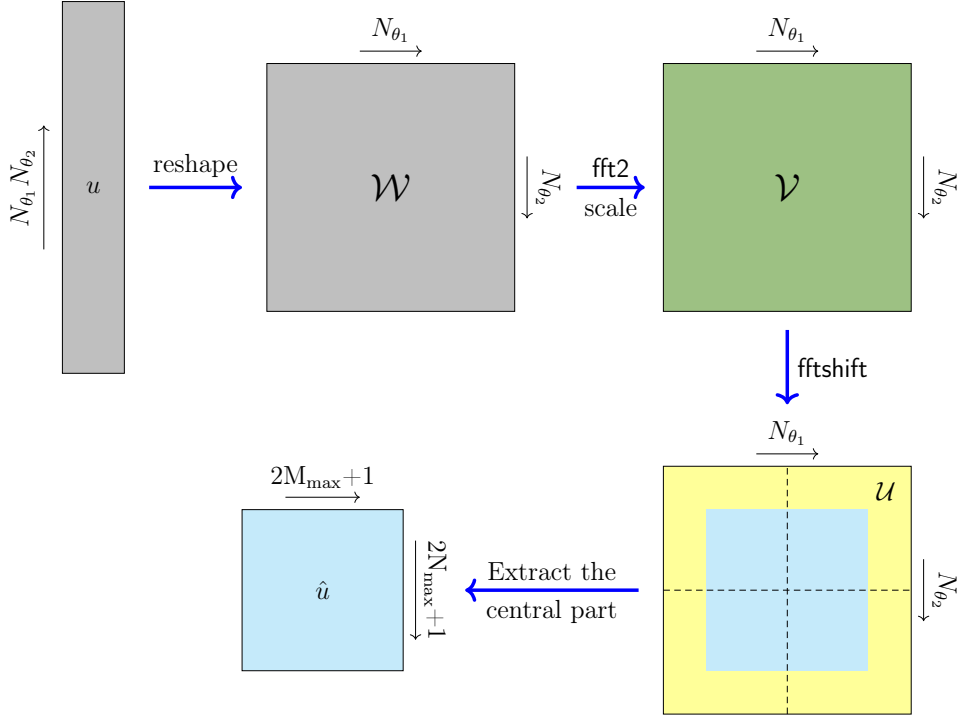
Figure 6: Pictorial representation of the workings of the fast angular 2D forward Fourier transform (Algorithm 4). 'Scale' indicates dividing the result of the 2D FFT by $(N_{\theta_1} N_{\theta_2})$

respect to basis set size increase, although a somewhat worse performance is expected theoretically. In contrast, a naive implementation of the forward transform scales in a quadratic manner with respect to basis set size, although both implementations of the transforms always agree with each other to machine precision.

In practice, the differences between the efficiencies of the fast and the naive transform implementations (both forward and inverse transforms) are not only apparent in terms of their respective scalings with respect to basis set size, but also the actual computational wall times. Indeed, we found that the fast transform implementations can be orders of magnitude faster as compared to the naive ones, even for relatively small basis set sizes. In Algorithm 5, we outline the steps of calculating the product of the Hamiltonian matrix with a wavefunction vector block by use of the fast transforms, as implemented in HelicES.

Algorithm 5: Product of Hamiltonian Matrix with a block vector of wavefunctions

Input: Block of N_s wavefunctions expressed in reciprocal space, i.e., $\hat{X} \in \mathbb{C}^{\mathcal{L} \times N_s}$, real space representation of local potential $V(\theta_1, \theta_2, r)$ as a vector $\mathscr{V} \in \mathbb{C}^{N_{\theta_1} N_{\theta_2} N_r}$, cyclic k-point ν and helical k-point η.
Prerequisites: Indexing function $\mathfrak{i} : \Gamma \to \{1, 2, \ldots, \mathcal{L}\}$ (eq. 35),
for each $n \in [-N_{\max}, N_{\max}]$, the matrix $I_n \in \mathbb{R}^{K_{\max} \times K_{\max}}$ with entries given by $I_n(k, k') = \mathcal{I}(n, k, k')$ (eq. 31),
vector $\Lambda \in \mathbb{R}^{\mathcal{L}}$ with entries corresponding to eq. 20, i.e., $\Lambda(\mathfrak{i}(m, n, k)) = \lambda_{m,n,k}^0$,
vector $\mathscr{M} \in \mathbb{C}^{\mathcal{L}}$ with entries $\mathscr{M}(\mathfrak{i}(m, n, k)) = i 2\pi m$,
and vector $\mathscr{N} \in \mathbb{C}^{\mathcal{L}}$ with entries $\mathscr{N}(\mathfrak{i}(m, n, k)) = i 2\pi \mathfrak{N} n$.

- -

Initialize $a = \frac{2\pi^2}{\tau^2}\left\{ \nu \alpha \left(2\eta - \nu \alpha\right) - \eta^2 \right\}$, $b = \frac{2i\pi}{\tau^2}\left(\nu\alpha - \eta\right)$, $c = \frac{2i\pi\alpha}{\tau^2}\left(\eta - \nu\alpha\right)$

Set $\hat{Y} = O_{\mathcal{L} \times N_s}$ // Result to be stored in this
for $j \leq N_s$ **do**
    $\hat{Z} = \hat{X}(:, j)$ // Work on j^{th} wavefunction.
    $\hat{P} = \mathscr{M} .* \hat{Z}$
    $\hat{Q} = \mathscr{N} .* \hat{Z}$
    $\hat{Y}(:, j) = \frac{1}{2} * (\Lambda .* \hat{Z})$
    $\hat{Y}(:, j) = \hat{Y}(:, j) - a * \hat{Z} - b * \hat{P} - c * \hat{Q}$
    if $\nu \neq 0$ **then**
        Initialize $i = 1$
        for $m, n \in \Gamma$ **do**
            $T = \frac{\nu^2}{R^2} * \hat{Z}(i : i + K_{\max} - 1) + \frac{i\nu}{\pi R^2} * \hat{Q}(i : i + K_{\max} - 1)$
            $\hat{Y}(i : i + K_{\max} - 1, j) = \hat{Y}(i : i + K_{\max} - 1, j) + I_n * T$
            $i = i + K_{\max}$
        end
    end
    $Z = \text{FastInverseBasisTransform}(\hat{Z})$ // Use Algorithm 1.
    $\hat{Y}(:, j) = \hat{Y}(:, j) + \text{FastForwardBasisTransform}(Z .* \mathscr{V})$ // Use Algorithm 3.
end
Result: $\hat{Y} \in \mathbb{C}^{\mathcal{L} \times N_s}$, i.e., the product of the Hamiltonian with $\hat{X}$ at the given values of η, ν.

3.5. Eigensolvers and Preconditioning

As mentioned earlier, we make use of matrix-free iterative eigenvalue solvers for diagonalization of the discretized Hamiltonian. Within HelicES, we have investigated two different diagonalization strategies for this purpose. The first is based on the Krylov-Schur method as implemented in the MATLAB Eigs function [78–80]. The second is based on a MATLAB implementation [81] of the Locally Optimal Block Preconditioned Conjugate Gradient (LOBPCG) scheme [82–84]. LOBPCG requires the use of a preconditioner, for which we have adopted the Teter-Payne-Allan (TPA) recipe [79, 80, 85]. This preconditioner was originally developed in the context of plane-wave calculations of bulk systems, but has also been successfully applied to other spectral methods [45]. During LOBPCG iterations, use of the TPA preconditioner requires the calculation and application of a diagonal matrix $\mathscr{K} \in \mathbb{R}^{\mathcal{L} \times \mathcal{L}}$ to the residual vector. The entries of the matrix are:

$$\mathscr{K}_{i,j} = \frac{27 + 18\, g_i + 12\, g_i^2 + 8\, g_i^3}{27 + 18\, g_i + 12\, g_i^2 + 8\, g_i^3 + 16\, g_i^4}\, \delta_{i,j}, \tag{49}$$

with:

$$g_i = \frac{\text{kinetic energy of basis function } i}{\text{kinetic energy of the residual vector}}. \tag{50}$$

As shown in Fig. 7, the preconditioner can have quite a dramatic effect on the convergence of the diagonalization procedure, especially as the basis set size (and therefore, the size of the discretized Hamiltonian) is increased. Thus, the benefits of the preconditioner are likely to become more apparent when larger systems and/or harder pseudopotentials are considered.
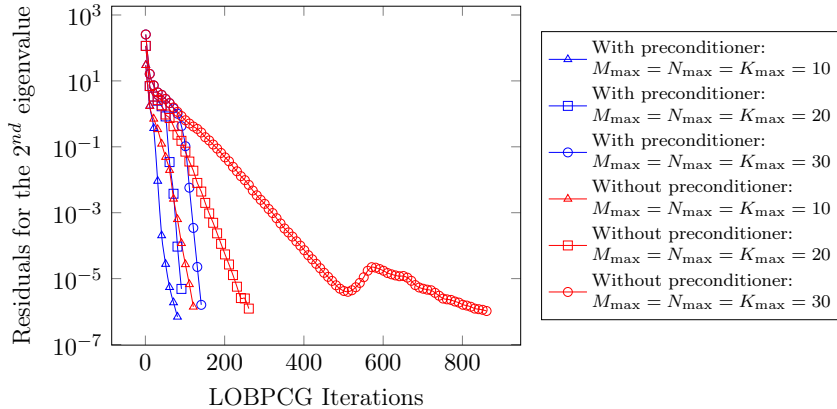

Figure 7: Effect of the Teter-Payne-Allan preconditioner [85] on LOBPCG [82] iterations for diagonalizing the discretized Hamiltonian in HelicES. An untwisted $(6,6)$ armchair carbon nanotube (Mayer pseudopotentials [86]) has been used, and the residual associated with the 2nd eigenvalue for $\eta = 0, \nu = 0$ as been monitored. For clarity, the residual for every 10th iteration has been plotted. Without preconditioning, the number of iterations required to reach a given convergence threshold tends to dramatically increase as the basis set grows larger.

We found that use of LOBPCG along with the TPA preconditioner generally tends to require longer diagonalization wall times as compared to Eigs along with an energy cutoff. Therefore, the latter strategy is adopted for most of the the examples considered in the next section. Implementation of more efficient eigensolvers in HelicES, particularly, ones that work well within self consistent field iterations [87–89], is the scope of future work.

4. Results

We now present results obtained using HelicES and investigate the convergence and accuracy properties of our implementation. All of our calculations have been carried out using smooth empirical pseudopotentials [86, 90]. We have used the planewave code PETRA [91], as well as two separate MATLAB based finite difference codes to generate reference data for comparison purposes. Specifically, we have employed the helical symmetry adapted finite difference code Helical DFT [20, 30] and the Cartesian grid finite difference code RSDFT [92]. The original versions of these finite difference codes were designed for self consistent field calculations, and were modified to work with the empirical pseudopotentials used in HelicES. We have also carried out comparisons of results obtained from HelicES against data obtained from the literature [86, 90]. We have used the WebPlotDigitizer tool [93] for extracting data from published plots.

4.1. Computational Platform

All simulations involving HelicES were carried out using dedicated desktop workstations (Dell Precision 7920 Tower, iMac, and iMac Pro) or on single nodes of the Hoffman2

cluster at UCLA's Institute for Digital Research and Education (IDRE). The Dell Precision workstation has an 18-core Intel Xeon Gold 5220 processor (24.75 L3 MB cache, 2.2 GHz clock speed), 256 GB of RAM and 1 TB of SATA Class 20 Solid State Drive (SSD) storage. The iMac has an 8-core Apple M1 processor (12 MB L2 cache, 3.2 GHz clock speed), 16 GB of RAM and a 2 TB Solid State Drive (SSD). The iMac Pro has an 18-core Intel Xeon W processor (24.75 MB L3 cache, 2.3 GHz clock speed), 256 GB of RAM and a 2 TB SSD. Every compute node of the Hoffman2 cluster has two 18-core Intel Xeon Gold 6140 processors (24.75 MB L3 cache, clock speed of 2.3 GHz), 192 GB of RAM and local SSD storage. MATLAB version 9.7.0 (R2019b) was used for the simulations. Parallelization was achieved by use of using MATLAB's Parallel Computing Toolbox. Reference results generated using Helical DFT [30], RSDFT [92] and PETRA [91] employed the above platforms as well.

4.2. Convergence Studies

Using a twisted armchair carbon nanotube as an example system (Mayer pseudopotentials [86]), we first investigate the convergence properties of HelicES. Considering first the case of eigenvalues of the Hamiltonian at $\eta = 0, \nu = 0$, we see in Fig. 8 that as the number of basis functions in HelicES is increased, there is a rapid convergence to the reference values, regardless of which eigenvalue is considered. Consistent with earlier results for electronic structure calculations using spectral basis sets [45, 70], HelicES shows a curvature on a log-log scale, indicative of super-polynomial convergence. In contrast, the finite difference method, also shown on the same figure, shows slower, polynomial convergence. This is consistent with earlier findings for finite difference electronic structure calculations using curvilinear coordinates [20, 40]. Furthermore, when the energy cutoff criterion is engaged, HelicES appears to employ noticeably fewer degrees of freedom than the finite difference method (Helical DFT) in reaching the same levels of convergence.
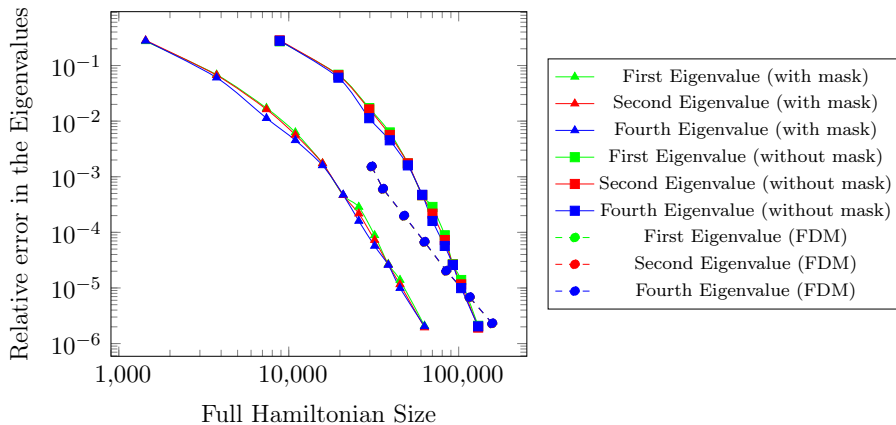

Figure 8: Convergence of the first three non-degenerate eigenvalues of an armchair $(16, 16)$ carbon nanotube with a twist parameter of $\alpha = 0.002$ using HelicES (both with and without the energy cutoff mask implemented) and a finite difference method (FDM), i.e., Helical DFT [20, 30]. The mesh size of the FDM decreases from 0.7 Bohrs to 0.4 Bohrs (in steps of 0.05 Bohrs) as the full Hamiltonian size varies from $30,744$ to $157,440$. The sparsity factor for the FDM Hamiltonian was 0.0055. The reference eigenvalues were taken to be the ones using an energy cutoff of 40 Ha for HelicES and a mesh spacing of 0.10 Bohr for Helical DFT. The $\eta = 0, \nu = 0$ case ("gamma point") is considered here. Note that the errors in the eigenvalues, for different eigenvalues, differ by $\mathcal{O}(10^{-4})$ or less in the FDM case, which makes them indistinguishable in the plot above.

The electronic features of quasi-one-dimensional systems can be characterized by one-dimensional band diagrams [20, 30], and these can be readily calculated for systems of interest using HelicES. As the next step in our studies, we checked the convergence behavior of the code with regard to a few quantities that are associated with the overall features of the one-dimensional band diagram of the aforementioned armchair carbon nanotube system. These include the the electronic band energy — which for an insulating system is simply twice the sum of all occupied state eigenvalues, the valence band maximum eigenvalue, the conduction band minimum eigenvalue and the band gap. As shown in Fig. 9, we see that all these quantities, except for the band gap, show monotonic convergence to reference values. We also note that convergence of the band gap is nearly monotonic until the curve enters regions of very high accuracy ($\mathcal{O}(10^{-6})$ in the figure) and this behavior is likely related to the fact that the band gap is calculated as the difference of two quantities.
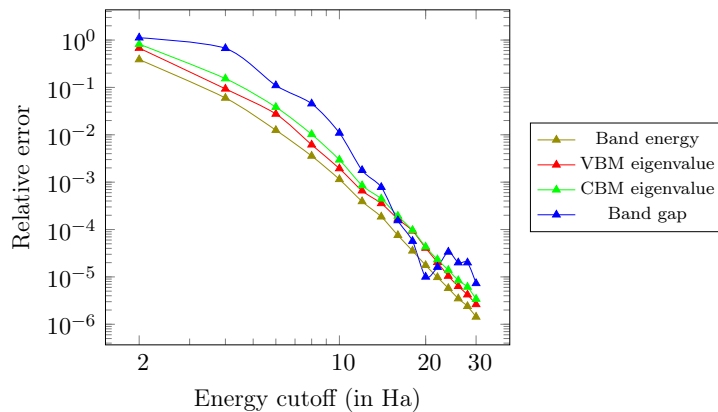
Figure 9: Convergence of the electronic band energy, the Valence Band Maximum (VBM) eigenvalue, the Conduction Band Minimum (CBM) eigenvalue, and the band gap, with respect to the energy cutoff, in the HelicES code. An armchair $(16, 16)$ carbon nanotube with a twist parameter of $\alpha = 0.002$ has been investigated. The reference values were generated using an energy cutoff of 40 Ha.
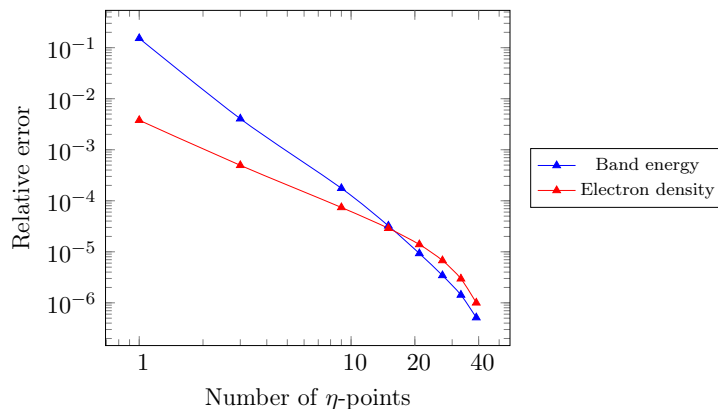
Figure 10: Convergence of the band energy and electron density of a $(16, 16)$ armchair carbon nanotube with a twist parameter of $\alpha = 0.002$. The reference value was taken to be from a calculation with 45 η-points

Within HelicES, the electronic properties of quasi-one-dimensional systems are also expected to exhibit convergence with respect to the number of points used to discretize

23

the η-space (Section 3.3). In Fig. 10, we explore the convergence behavior of the electron density (in terms of the L^1 norm per electron) and the electronic band energy for the aforementioned carbon nanotube system, as the number of η points in the calculation is increased. We see that both the above quantities show excellent convergence. We note that the electron density can be calculated from the wavefunctions $\phi_j(\theta_1, \theta_2, r; \eta, \nu)$, and the corresponding electronic occupation numbers $\varsigma_j(\eta, \nu)$ as:

$$\rho(\theta_1, \theta_2, r) = \frac{2}{\mathfrak{N}} \sum_{j=1}^{N_s} \sum_{b=1}^{N_\eta} \sum_{\nu=0}^{\mathfrak{N}-1} w_b \, \varsigma_j(\eta_b, \nu) \big| \phi_j(\theta_1, \theta_2, r; \eta_b, \nu) \big|^2 . \tag{51}$$

This requires inverse basis transforms to be carried out on each wavefunction vector, at the end of the diagonalization procedure. We also note from Figs. 9,10 that for the Mayer pseudopotential employed in the above calculations, an energy cutoff of 16 Ha and 15 η-points are more than sufficient to reach chemical accuracy.

4.3. Accuracy Studies

While the discussion in Section 4.2 serves to illustrate the systematic convergence properties of HelicES, it does not address the accuracy or correctness of the converged results produced by the code. Therefore, we now carry out a series of systematic tests and compare the results produced by HelicES against solutions produced by other methods, for a variety of systems.

Our first set of tests compares the results produced by HelicES against those computed through the Finite Difference Method (FDM). For these studies, the Mayer pseudopotential [86] was once again employed and the energy cutoff in HelicES was set at 16 Ha. Reference results using the FDM codes were generated using a mesh spacing of 0.2 Bohr, this being the finest mesh that could be uniformly employed for all systems of interest, within computational resource constraints. We first used the RSDFT code [92] for calculating the electronic structure of a variety of finite (cluster-like) systems. The bound state eigenvalues for these same systems, as calculated by HelicES are compared against RSDFT results in Table 1. We see that for these discretization parameters, the agreement between the codes with respect to individual eigenvalues is about 1.3×10^{-4} Ha or better, while the band energies agree to within millihartree range, suggesting excellent accuracy.

Next, we generated the electronic band diagram associated with a deformed quasi-one-dimensional system, namely an armchair nanotube subjected to about $\beta = 2.95°$ of twist per nanometer. Reference calculations were carried out using the Helical DFT code. Both Helical DFT and HelicES were made to use 21 η-points and the Eigs eigensolver in MATLAB. As shown in Figure 11, the band diagrams produced by the two codes are virtually identical, once again suggesting the excellent accuracy of HelicES. Overall, these findings illustrate that HelicES adequately addresses many of the the computational bottlenecks in existing methods for the study of electronic properties of quasi-one-dimensional systems, commensurate with its design goals.
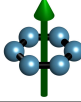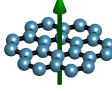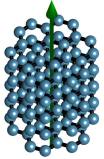
System (# atoms)	Hamiltonian size	Maximum difference in the eigenvalues (in Ha) between HelicES and FDM	Difference in band energy (in Ha/atom) between HelicES and FDM
Carbon dimer (2 atoms)	HelicES without mask: 148625 HelicES with mask: 96353 FDM: 1030301	6.0698×10^{-5}	5.7475×10^{-5}
Carbon ring (6 atoms)	HelicES without mask: 148625 HelicES with mask: 96353 FDM: 8120601	3.8266×10^{-5}	3.7686×10^{-5}
Carbon disk (24 atoms)	HelicES without mask: 240096 HelicES with mask: 152556 FDM: 4173281	7.7064×10^{-5}	4.7145×10^{-5}
Carbon pillar (120 atoms)	HelicES without mask: 129591 HelicES with mask: 85741 FDM: 8120601	1.2604×10^{-4}	3.7118×10^{-5}

Table 1: Accuracy of the HelicES code while studying finite systems (green arrow denotes the $\mathbf{e_Z}$ axis). Reference data was generated using RSDFT [92], a finite difference method (FDM) based MATLAB code. The last two columns show the maximum differences in the eigenvalues and the band energy per atom computed using the two methods.
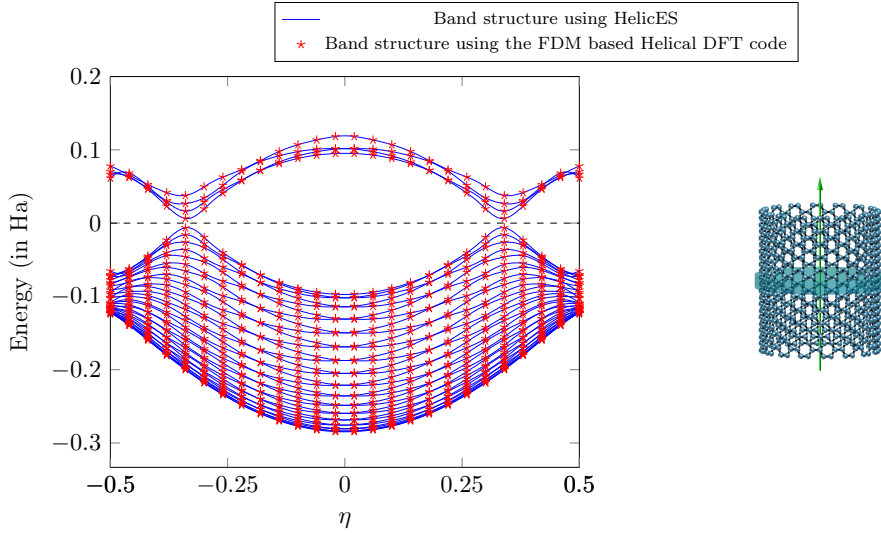

Figure 11: Comparison of band diagrams for a twisted $(16, 16)$ armchair carbon nanotube (diameter = 2.726 nm) with twist parameter of $\alpha = 0.002$, generated using HelicES and the FDM based Helical DFT code [20, 30]. The green shaded region in the structure on the right is the fundamental domain used in HelicES, while the green arrow denotes the $\mathbf{e_Z}$ axis.

Due to inherent design limitations, the aforementioned FDM codes are unable to simulate quasi-one-dimensional nanostructures which have atoms situated near or along the system axis (e.g. nanoribbons, nanowires or small diameter nanotubes). However,

these systems can be conveniently dealt with by HelicES. To carry out accuracy tests for such systems therefore, we compared the band structures calculated by HelicES against those generated through alternate electronic structure calculation techniques. The first of these is based on the transfer matrix method [94–96], often used in electromagnetics calculations. In Figs. 12 and 13 we see that the band structure calculated by HelicES is in nearly perfect agreement with results calculated using this technique in [86]. The systems considered here are carbon nanotubes with radii about 0.3 to 0.4 nanometers. For the $(5,5)$ armchair nanotube, the position of the Dirac cone is correctly predicted to be at $\eta = \pm\frac{1}{3}$. Additionally, the $(10,0)$ zigzag nanotube, the band gap calculated by HelicES is 1.05 eV which is very close to the value of 1.04 eV obtained in [86].
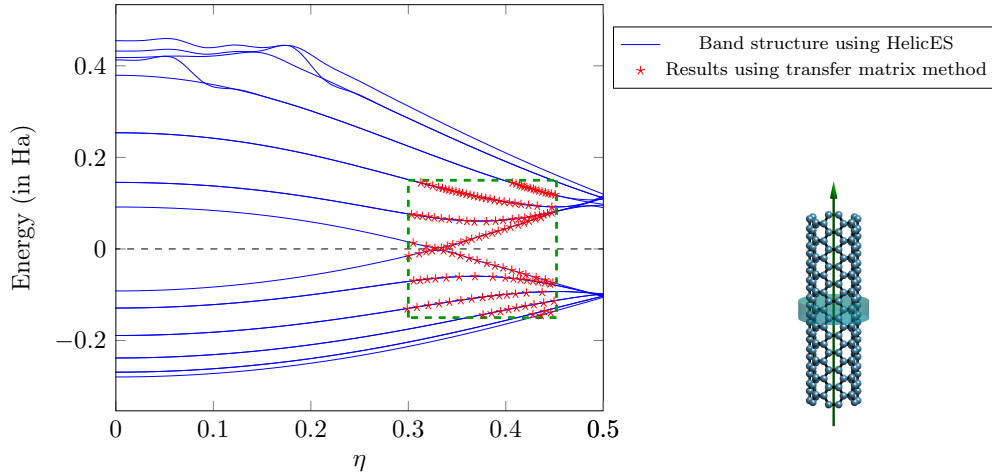
Figure 12: Comparison of band diagrams for a $(5,5)$ armchair carbon nanotube (diameter = 0.851 nm) generated using HelicES and a transfer-matrix technique [86]. The dashed green box in the plot represents the region of the band diagram over which the reference data was available for comparison. The green shaded region in the structure on the right is the fundamental domain used in HelicES and the green arrow denotes the $\mathbf{e_Z}$ axis.
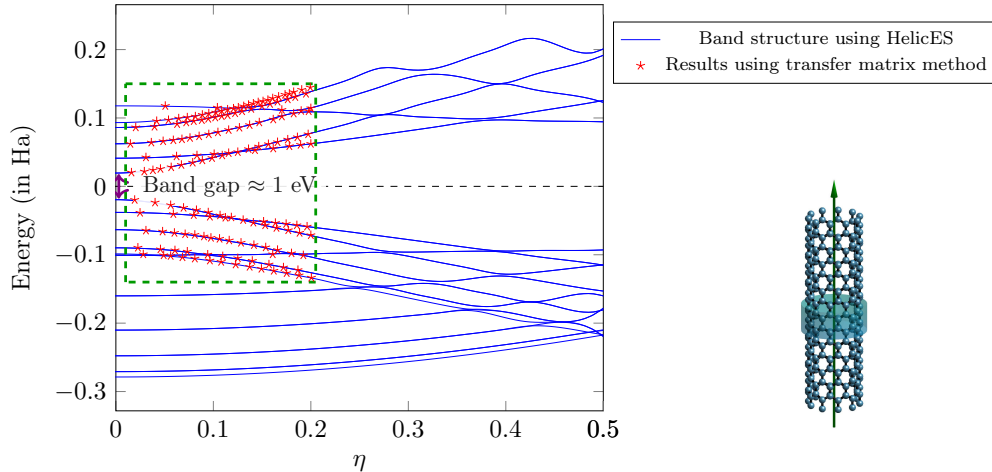

Figure 13: Comparison of band diagrams for a $(10,0)$ zigzag carbon nanotube (diameter = 0.983 nm) generated using HelicES and a transfer-matrix technique [86]. The dashed green box in the plot represents the region of the band diagram over which the reference data was available for comparison. The green shaded region in the structure on the right is the fundamental domain used in HelicES and the green arrow denotes the $\mathbf{e_Z}$ axis.

26

Next, we used the PETRA code for studying an armchair graphene nanoribbon, as well as a silicon nanowire oriented along the $\langle 100 \rangle$ direction. Both these systems were treated using the empirical pseudopotentials developed in [90] and feature hydrogen passivation. Figs. 14 and 15 show that the overall agreement between HelicES and PETRA is excellent, although some minor variations at the edge of the highest energy band for the nanoribbon case may be observed. This is possibly due to the different boundary conditions being employed by PETRA and HelicES in the directions orthogonal to the ribbon axis. We also note that the band gap for the silicon nanowire calculated by HelicES is 3.82 eV, which is very close to the value of 3.84 eV reported in [90].
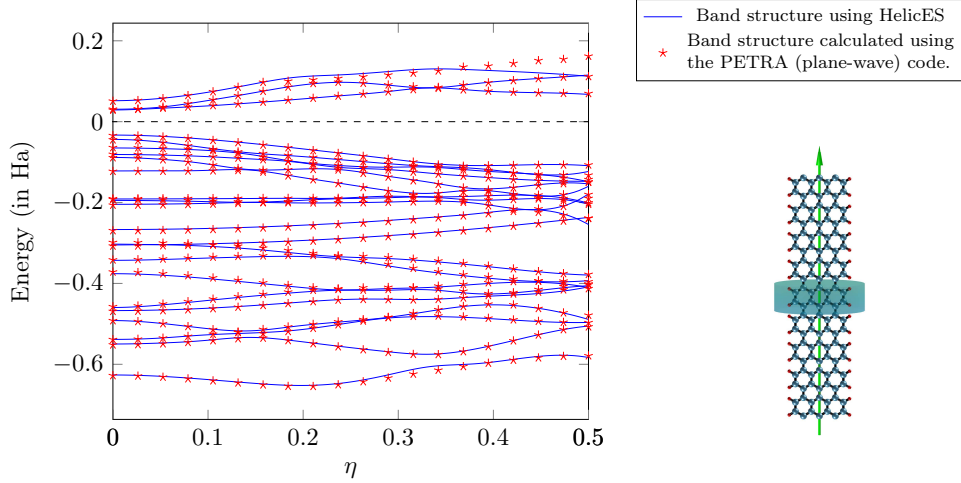

Figure 14: Comparison of band diagrams for a hydrogen passivated armchair graphene nanoribbon generated using HelicES and a plane-wave technique [90, 91]. The green shaded region in the structure on the right is the fundamental domain used in HelicES and the green arrow denotes the $\mathbf{e_Z}$ axis.
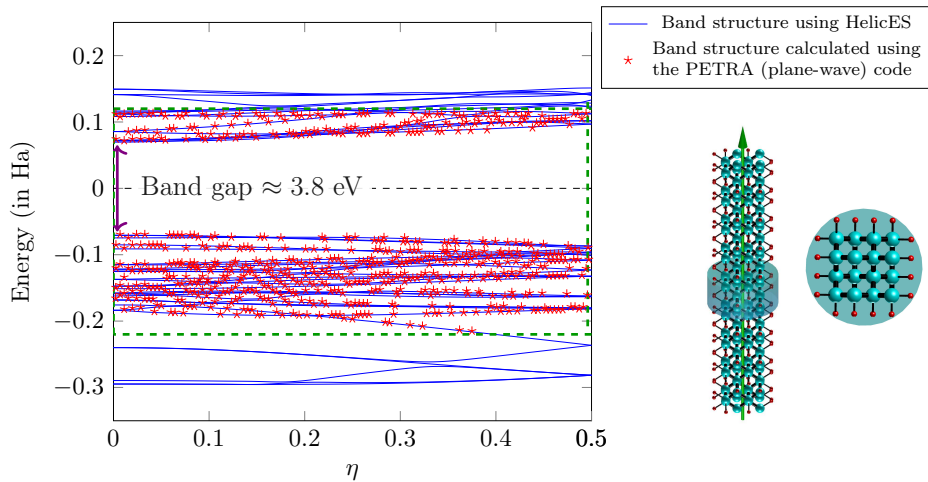

Figure 15: Comparison of band diagrams for a hydrogen passivated, $\langle 100 \rangle$ oriented silicon nanowire generated using HelicES and a plane-wave technique [90, 91]. The dashed green box in the plot represents the region of the band diagram over which the reference data was available for comparison. The green shaded region in the structure in the middle is the fundamental domain used in HelicES, with the green arrow denoting the $\mathbf{e_Z}$ axis. The right image shows a top view of the structure (i.e., looking down along $\mathbf{e_Z}$).

4.4. Comments on computational efficiency and timing studies

We now discuss issues connected to the computational efficiency of HelicES. By design, the code is meant to overcome the computational limitations of prior approaches in modeling quasi-one-dimensional systems. We highlight this aspect of the code by providing timing comparisons between HelicES and other existing methods, for a few systems of interest. We have focused on the Plane-wave Electronic TRAnsport (PETRA) code [90, 91, 97] which can model periodic systems, and Helical DFT [20, 30] which models quasi-one-dimensional structures within a finite difference framework. For comparisons with PETRA, we chose a twisted hydrogen-passivated graphene nanoribbon. Note that while realistic values of α range from 0.0005 to 0.0025 (i.e., less than about 2.1° per nanometer), it is not feasible to use these values in PETRA. This is because to simulate such a system in a typical plane-wave code like PETRA, we would require $1/\alpha$ times the number of atoms needed for untwisted geometries (for rational values of α). However, the number of atoms required in the fundamental domain in HelicES is independent of the amount of twist. Thus, while a realistic twisted nanoribbon can be studied using only 20 atoms in HelicES, PETRA would require at least $10,000$ atoms in the fundamental domain for the same system. Keeping this in mind, we use larger values of $\alpha = 0.25, 0.2, 0.1$, and 0.05, so that the simulation and timing data from PETRA could be obtained within reasonable wall times. For both codes, we used the same diagonalization technique. The simulations were carried out on dedicated workstations, or on a single node of the Hoffman2 cluster when larger memory was needed. In our studies, we noted factors of $1.26, 1.84, 3.83$, and 12.25 improvement in the total diagonalization wall time of HelicES over PETRA, for $\alpha = 0.25, 0.2, 0.1$, and 0.05 respectively. Based on the above discussion, we anticipate that the performance gap between HelicES and PETRA, as well as the memory requirements of the latter, will only increase when more realistic values of α or more complicated unit cells are considered.

Due to the fundamental limitations of plane-wave codes to efficiently represent helical symmetries, it also makes sense to compare HelicES to Helical DFT, since the right symmetries are incorporated into both these codes, although the latter uses finite differences. For this purpose, we studied a twisted $(16, 16)$ armchair carbon nanotube with a diameter of 2.726 nm and a twist parameter of $\alpha = 0.002$, and we used 21 $\eta-$points. As we showed earlier (Section 4.3), while the two codes produce nearly identical results, the diagonalization wall time for HelicES was about a factor of 27 lower, and the memory footprint was also significantly less. These observations continue to be true when larger values of the energy cutoff are used in HelicES, with the diagonalization wall time of the code being about a factor of 8 lower than Helical DFT, even when an energy cutoff of 40 Ha is employed.

To finish this discussion on computational advantages of HelicES, we now present a system that cannot be simulated in Helical DFT, and one that will require extensive computational resources in typical periodic finite difference or plane-wave codes — an armchair graphene nanoribbon with a twist of $\alpha = 0.02$. Note that this is still a relatively high value of α, but was chosen here for a better visual representation of the system. The band diagram of this system is presented in Fig. 16. Noticeably, in contrast to the untwisted, passivated nanoribbon presented in Fig. 14, this system appears to have a vanishingly small band-gap, indicative of metallic behavior.
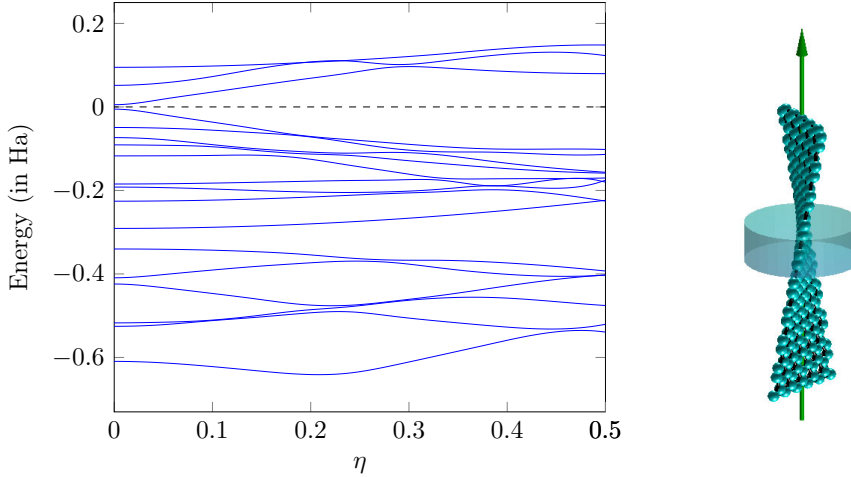
Figure 16: Band diagram for an armchair graphene nanoribbon with a twist parameter of $\alpha = 0.02$ (corresponding to a rate of twist $\beta = 16.9°$ per nanometer) generated using HelicES. The green shaded region in the structure on the right is the fundamental domain used in HelicES. The green arrow denotes the $\mathbf{e_Z}$ axis.

4.5. Application to the study of the electromechanical response of a nanotube

Finally, as a demonstration of the utility of the computational method developed here, we study the the electromechanical response of a quasi-one-dimensional nanomaterial as it undergoes deformations. Specifically, we consider a carbon nanotube with a radius of about 1.0 nanometer (an armchair $(16, 16)$ tube), and subject it to twisting. We start from the untwisted structure and increase the rate of applied twist, considering up to about $\beta = 7.4°$, in our simulations. Fig. 17 shows the variation of the band gap of the material with applied twist. For comparison purposes, results from full self consistent Kohn-Sham DFT calculations using ab initio Troullier Martins pseudopotentials [98] and Local Density Approximation based exchange correlation [99, 100], are also shown (obtained from [20]). It is well known that upon twisting, armchair nanotubes — which are generally metallic in untwisted form — show metal-to-semiconductor transitions, and that these changes manifest themselves as oscillatory behavior in the band gap [20, 101–103]. We see from Fig. 17 that the results from HelicES do reproduce this qualitative behavior correctly, but the actual response curve is quantitatively different from the first principles data. This is very likely due to the lack of inclusion of atomic relaxation effects in HelicES, as well as the general failure of the Mayer pseudopotential to model scenarios where the carbon atoms do not form a perfect honeycomb lattice — a consequence of the shearing distortions that arise from the applied twist in this case. Therefore, these results strongly suggest the need for building in ab initio pseudopotentials and self consistent iterations into HelicES, which constitutes ongoing work [104].
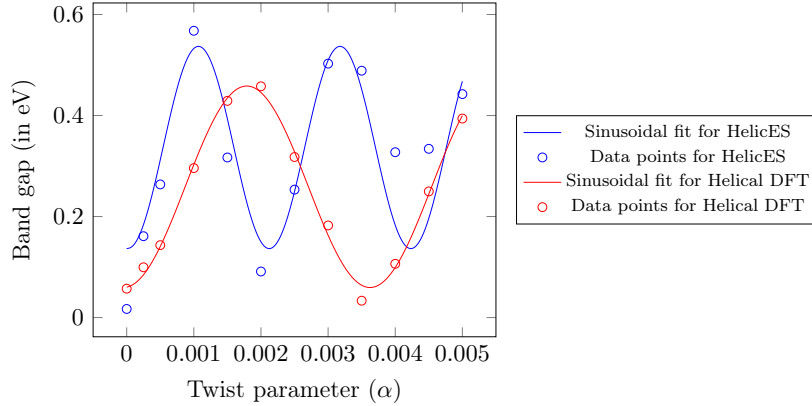
29

Figure 17: Band gap trend as the twist parameter α is varied for a $(16, 16)$ armchair carbon nanotube. Results from HelicES (empirical pseudopotentials) and the Helical DFT code (self consistent calculations with ab initio pseudopotentials and atomic relaxation effects included [20, 30]) are both shown.

5. Conclusions

In summary, we have presented a novel spectral method for efficiently solving the Schrödinger equation for quasi-one-dimensional materials and structures. The basis functions in our method — helical waves — are natural analogs of plane-waves, and allow systematically convergent electronic structure calculations of materials such as nanowires, nanoribbons and nanotubes to be carried out. We have discussed various mathematical, algorithmic and implementation oriented issues of our technique. We have also used our method to carry out a variety of demonstrative calculations and studied its accuracy, computational efficiency, and convergence behaviors.

We anticipate that the method presented here will find utility in the discovery and characterization of new forms of low dimensional matter. It is particularly well suited for coupling with specialized machine learning techniques [105] and for the multiscale modeling of low dimensional systems [106]. Building self-consistency into the method, so as to enable *ab initio* calculations (e.g. using Hartree-Fock or Kohn-Sham Density Functional Theory [107]) remains the scope of ongoing and future work. An important first step in this direction is efficient solution of the associated electrostatics problem [108], towards which we have been making recent progress [104, 109]. Finally, the full power of some of the techniques described here can be brought to bear upon complex materials problems, once a parallel, efficient, hardware optimized version of HelicES is available. Development of such a code constitutes yet another avenue of ongoing and future work.

—

Appendix A. Derivation of the governing equation in helical coordinates

We are interested in solutions of the Schrödinger equation, i.e., $\left(-\frac{1}{2}\Delta + V(\mathbf{x})\right)\psi = \lambda\,\psi$, as it applies to a quasi-one-dimensional structure. For a function $\psi(\theta_1, \theta_2, r)$ expressed in helical coordinates, the Laplacian is given by [29, 30]:

$$\Delta\psi = \psi_{rr} + \frac{1}{r}\psi_r + \frac{1}{\tau^2}\psi_{\theta_1\theta_1} - \frac{2\alpha}{\tau^2}\psi_{\theta_1\theta_2} + \frac{1}{4\pi^2}\left(\frac{1}{r^2} + \frac{4\pi^2\alpha^2}{\tau^2}\right)\psi_{\theta_2\theta_2}. \tag{A.1}$$

30

Considering the helical and cyclic symmetry adapted Bloch ansatz, $\psi(\theta_1, \theta_2, r; \eta, \nu) = e^{-i2\pi(\eta\theta_1 + \nu\theta_2)}\phi(\theta_1, \theta_2, r; \eta, \nu)$, we first note:

$$\begin{aligned}
\psi_r &= e^{-i2\pi(\eta\theta_1+\nu\theta_2)}\phi_r\,, \\
\psi_{rr} &= e^{-i2\pi(\eta\theta_1+\nu\theta_2)}\phi_{rr}\,, \\
\psi_{\theta_1} &= e^{-i2\pi(\eta\theta_1+\nu\theta_2)}\left[-i2\pi\eta\phi + \phi_{\theta_1}\right]\,, \\
\psi_{\theta_2} &= e^{-i2\pi(\eta\theta_1+\nu\theta_2)}\left[-i2\pi\nu\phi + \phi_{\theta_2}\right]\,, \\
\psi_{\theta_1\theta_2} &= e^{-i2\pi(\eta\theta_1+\nu\theta_2)}\left[-4\pi^2\eta\nu\phi - i2\pi\nu\phi_{\theta_1} - i2\pi\eta\phi_{\theta_2} + \phi_{\theta_1\theta_2}\right]\,, \\
\psi_{\theta_1\theta_1} &= e^{-i2\pi(\eta\theta_1+\nu\theta_2)}\left[-i2\pi\eta\left(-i2\pi\eta\phi + 2\phi_{\theta_1}\right) + \phi_{\theta_1\theta_1}\right]\,, \\
\psi_{\theta_2\theta_2} &= e^{-i2\pi(\eta\theta_1+\nu\theta_2)}\left[-4\pi^2\nu^2\phi - 4i\pi\nu\phi_{\theta_2} + \phi_{\theta_2\theta_2}\right]\,.
\end{aligned} \tag{A.2}$$

Thus, we get:

$$\begin{aligned}
\Delta\psi = \bigg[&\phi_{rr} + \frac{1}{r}\phi_r + \frac{1}{\tau^2}\phi_{\theta_1\theta_1} - \frac{4\pi^2\eta^2}{\tau^2}\phi - \frac{i4\pi\eta}{\tau^2}\phi_{\theta_1} - \frac{2\alpha}{\tau^2}\phi_{\theta_1\theta_2} + \frac{8\alpha\pi^2\eta\nu}{\tau^2}\phi \\
&+ \frac{4i\pi\nu\alpha}{\tau^2}\phi_{\theta_1} + \frac{4i\pi\eta\alpha}{\tau^2}\phi_{\theta_2} + \frac{1}{4\pi^2}\left(\frac{1}{r^2} + \frac{4\pi^2\alpha^2}{\tau^2}\right)\phi_{\theta_2\theta_2} \\
&- \nu^2\left(\frac{1}{r^2} + \frac{4\pi^2\alpha^2}{\tau^2}\right)\phi - \frac{i\nu}{\pi}\left(\frac{1}{r^2} + \frac{4\pi^2\alpha^2}{\tau^2}\right)\phi_{\theta_2}\bigg]e^{-i2\pi(\eta\theta_1+\nu\theta_2)}\,,
\end{aligned} \tag{A.3}$$

which simplifies to:

$$\begin{aligned}
\Delta\psi = \bigg[&\Delta\phi + \left(\frac{4\pi^2}{\tau^2}\left[\nu\alpha\left(2\eta - \nu\alpha\right) - \eta^2\right] - \frac{\nu^2}{r^2}\right)\phi + \frac{4i\pi}{\tau^2}\left(\nu\alpha - \eta\right)\phi_{\theta_1} \\
&+ i\left[\frac{4\pi\alpha}{\tau^2}\left(\eta - \nu\alpha\right) - \frac{\nu}{\pi r^2}\right]\phi_{\theta_2}\bigg]e^{-i2\pi(\eta\theta_1+\nu\theta_2)}\,.
\end{aligned} \tag{A.4}$$

Hence the action of the Schrödinger operator on ψ can be expressed as:

$$\begin{aligned}
\left(-\frac{1}{2}\Delta + V\right)\psi = \bigg[&-\frac{1}{2}\Delta\phi - \left(\frac{2\pi^2}{\tau^2}\left\{\nu\alpha\left(2\eta - \nu\alpha\right) - \eta^2\right\} - \frac{\nu^2}{2r^2}\right)\phi \\
&- \frac{2i\pi}{\tau^2}\left(\nu\alpha - \eta\right)\phi_{\theta_1} - 2i\pi\left[\frac{\alpha}{\tau^2}\left(\eta - \nu\alpha\right) - \frac{\nu}{4\pi^2 r^2}\right]\phi_{\theta_2} + V\phi\bigg]e^{-i2\pi(\eta\theta_1+\nu\theta_2)}\,.
\end{aligned} \tag{A.5}$$

Since the phase $e^{-i2\pi(\eta\theta_1+\nu\theta_2)} \neq 0$, canceling it from both sides of the Schrodinger equation in ψ leaves us with the following eigenvalue problem in ϕ:

$$\begin{aligned}
\bigg[&-\frac{1}{2}\Delta\phi - \left(\frac{2\pi^2}{\tau^2}\left\{\nu\alpha\left(2\eta - \nu\alpha\right) - \eta^2\right\} - \frac{\nu^2}{2r^2}\right)\phi - \frac{2i\pi}{\tau^2}\left(\nu\alpha - \eta\right)\phi_{\theta_1} - \\
&2i\pi\left[\frac{\alpha}{\tau^2}\left(\eta - \nu\alpha\right) - \frac{\nu}{4\pi^2 r^2}\right]\phi_{\theta_2} + V\phi\bigg] = \lambda\phi
\end{aligned} \tag{A.6}$$

This equation needs to be discretized and solved over the fundamental domain, along with suitable boundary conditions in ϕ.

Appendix B. Derivation of the Basis Set

In analogy to the classical plane-wave method [27, 28], the basis functions in our scheme are eigenfunctions of the Laplacian. However, instead of periodic boundary conditions obeyed by planewaves, we consider boundary conditions resulting from invariance under helical and cyclic symmetries. The calculation presented below is based on similar results in [29], while a vector version of this calculation appears in [52, 53] in the context of x-ray diffraction patterns of twisted nanomaterials.

Let $F(\theta_1, \theta_2, r)$ be a basis function expressed in helical coordinates. Then, invariance under helical and cyclic symmetries implies that this function must be periodic in θ_1 with a period of 1, and also periodic in θ_2 with a period of $\frac{1}{\mathfrak{N}}$. Assuming $F(\theta_1, \theta_2, r)$ is separable, we characterize the dependence of the function on θ_1 and θ_2 through Fourier modes (i.e., complex exponentials), and write:

$$F_{m,n,k}(\theta_1, \theta_2, r) = e^{i2\pi(m\theta_1 + n\mathfrak{N}\theta_2)} \xi(r) . \tag{B.1}$$

Here $\xi(r)$ is a purely radial function that possibly depends on m, n, k, and incorporates normalization constants. The Laplacian of the above function in the helical coordinates is:

$$\Delta F_{m,n,k} = \xi_r \, e^{i2\pi(m\theta_1 + n\mathfrak{N}\theta_2)} + \frac{1}{r}\xi_{rr} \, e^{i2\pi(m\theta_1 + n\mathfrak{N}\theta_2)} - \frac{4\pi^2 m^2}{\tau^2} F_{m,n,k}$$
$$+ \frac{8\alpha\pi^2 nm\mathfrak{N}}{\tau^2} F_{m,n,k} - \left(\frac{1}{r^2} + \frac{4\pi^2\alpha^2}{\tau^2}\right) n^2\mathfrak{N}^2 F_{m,n,k} , \tag{B.2}$$

which can be rewritten as:

$$\Delta f_{m,n,k} = e^{i2\pi(m\theta_1 + n\mathfrak{N}\theta_2)} \left[\xi_{rr} + \frac{1}{r}\xi_r\right] - F_{m,n,k}\left[\frac{n^2\mathfrak{N}^2}{r^2} + \frac{4\pi^2}{\tau^2}(m - \alpha n\mathfrak{N})^2\right] . \tag{B.3}$$

Now, imposing the condition that $f_{m,n,k}$ is an eigenfunction of the Laplacian, i.e.,

$$-\Delta F_{m,n,k} = \lambda^0_{m,n,k} F_{m,n,k} , \tag{B.4}$$

we get:

$$-e^{i2\pi(m\theta_1 + n\mathfrak{N}\theta_2)}\left[\xi_{rr} + \frac{1}{r}\xi_r\right] + e^{i2\pi(m\theta_1 + n\mathfrak{N}\theta_2)}\left[\frac{n^2\mathfrak{N}^2}{r^2} + \frac{4\pi^2}{\tau^2}(m - \alpha n\mathfrak{N})^2\right]\xi$$
$$= \lambda^0_{m,n,k} e^{i2\pi(m\theta_1 + n\mathfrak{N}\theta_2)}\xi , \tag{B.5}$$

which simplifies to:

$$\xi_{rr} + \frac{1}{r}\xi_r - \xi\left[\frac{n^2\mathfrak{N}^2}{r^2} - \lambda^0_{m,n,k} + \frac{4\pi^2}{\tau^2}(m - \alpha n\mathfrak{N})^2\right] = 0 . \tag{B.6}$$

Denoting $\xi^2_{m,n} = \frac{4\pi^2}{\tau^2}(m - \alpha n\mathfrak{N})^2$ and performing the change of variables:

$$\tilde{r} = r\sqrt{\lambda^0_{m,n,k} - \gamma^2_{m,n}} , \quad \xi(r) = \widetilde{\xi}(\tilde{r}) , \tag{B.7}$$

we see that the above equation reduces to:

$$\tilde{r}^2 \, \widetilde{\xi}_{\tilde{r}\tilde{r}} + \tilde{r} \, \widetilde{\xi}_{\tilde{r}} + (\tilde{r}^2 - n^2\mathfrak{N}^2)\widetilde{\xi} = 0 . \tag{B.8}$$

This is simply Bessel's equation [67, 110] in $\widetilde{\xi}(\tilde{r})$. Since $n\mathfrak{N}$ is real, the general solution of this equation can be expressed in terms of ordinary Bessel functions of the first and second kind as:

$$\widetilde{\xi}(\tilde{r}) = A\,J_{n\mathfrak{N}}(\tilde{r}) + B\,Y_{n\mathfrak{N}}(\tilde{r})\,. \tag{B.9}$$

To evaluate the constants A and B, we need to invoke boundary and normalization conditions. Since the wavefunctions are expected to be finite valued at the origin ($r = 0$), and Bessel functions of the second kind approach infinity near 0, we conclude that $B = 0$. Furthermore, since the wavefunctions obey Dirichlet boundary conditions on the lateral surface of the computational domain ($r = R$), so should the basis functions used to discretize them. Hence, we obtain:

$$\xi\left(R\sqrt{\lambda^0_{m,n,k} - \gamma^2_{m,n}}\right) = A\,J_{n\mathfrak{N}}\left(R\sqrt{\lambda^0_{m,n,k} - \gamma^2_{m,n}}\right) = 0\,. \tag{B.10}$$

This implies that $R\sqrt{\lambda^0_{m,n,k} - \gamma^2_{m,n}}$ must be a root of the the the ordinary Bessel function of the first kind. Denoting the kth root ($k = 1, 2, \ldots$) of the Bessel function of order p, as b^p_k, we see that:

$$b^{n\mathfrak{N}}_k = R\sqrt{\lambda^0_{m,n,k} - \gamma^2_{m,n}}\,, \tag{B.11}$$

from which, it follows that:

$$\lambda^0_{m,n,k} = \left(\frac{b^{n\mathfrak{N}}_k}{R}\right)^2 + \left[\frac{2\pi}{\tau}\left(m - \alpha n\mathfrak{N}\right)\right]^2\,. \tag{B.12}$$

Thus, we have:

$$\xi(r) = A\,J_{n\mathfrak{N}}\left(\frac{b^{n\mathfrak{N}}_k}{R}r\right)\,. \tag{B.13}$$

Finally, to determine the constant A, we apply the orthonormality condition between two distinct basis functions $F_{m,n,k}$ and $F_{m',n',k'}$:

$$\langle F_{m,n,k}, F_{m',n',k'}\rangle_{\mathsf{L}^2(\mathcal{D})} = \delta_{m,m'}\,\delta_{n,n'}\,\delta_{k,k'}\,. \tag{B.14}$$

This requires that:

$$A^2 \int_0^1 e^{i2\pi(m-m')\theta_1}d\theta_1 \times \int_0^{\frac{1}{\mathfrak{N}}} e^{i2\pi\mathfrak{N}(n-n')\theta_2}d\theta_2$$
$$\times \int_0^R J_{n\mathfrak{N}}\left(\frac{b^{n\mathfrak{N}}_k r}{R}\right) J_{n'\mathfrak{N}}\left(\frac{b^{n'\mathfrak{N}}_{k'} r}{R}\right) 2\pi\tau r\,dr = \delta_{m,m'}\,\delta_{n,n'}\,\delta_{k,k'}\,. \tag{B.15}$$

Due to the properties of complex exponentials and Bessel functions, we note that this condition is readily satisfied for distinct basis functions (i.e., when any of the conditions $m \neq m'$, $n \neq n'$, $k \neq k'$ hold). For the case $m = m', n = n', k = k'$, we arrive at:

$$\frac{2\pi\tau A^2}{\mathfrak{N}} \int_0^R J^2_{n\mathfrak{N}}\left(\frac{b^{n\mathfrak{N}}_k r}{R}\right) r\,dr = 1\,, \tag{B.16}$$

i.e.,

$$\frac{2\pi\tau A^2}{\mathfrak{N}}\frac{R^2}{2}J^2_{n\mathfrak{N}+1}\left(b_k^{n\mathfrak{N}}\right) = 1\,. \tag{B.17}$$

Thus it follows that the normalization constant:

$$A = \sqrt{\frac{\mathfrak{N}}{\pi\tau}}\frac{1}{RJ_{n\mathfrak{N}+1}\left(b_k^{n\mathfrak{N}}\right)}\,, \tag{B.18}$$

and that:

$$\xi(r) \equiv \xi_{n,k}(r) = \sqrt{\frac{\mathfrak{N}}{\pi\tau}}\frac{1}{RJ_{n\mathfrak{N}+1}\left(b_k^{n\mathfrak{N}}\right)}J_{n\mathfrak{N}}\left(\frac{b_k^{n\mathfrak{N}}}{R}r\right)\,. \tag{B.19}$$

Hence, the basis functions in our method have the form:

$$F_{m,n,k}\left(\theta_1,\theta_2,r\right) = \sqrt{\frac{\mathfrak{N}}{\pi\tau}}\frac{1}{RJ_{n\mathfrak{N}+1}\left(b_k^{n\mathfrak{N}}\right)}e^{i2\pi(m\theta_1+n\mathfrak{N}\theta_2)}J_{n\mathfrak{N}}\left(\frac{b_k^{n\mathfrak{N}}r}{R}\right)\,. \tag{B.20}$$

Note that if the computational domain were an annular cylinder (as employed in [20, 30]), instead of the solid cylinder considered here, the boundary conditions on the radial part of the wavefunction would be expected to change. For Dirichlet boundary conditions applied to the inner and outer walls of such an annular cylinder — often employed in simulations of large diameter nanotubes — Bessel functions of both kinds would be involved (i.e., $A, B \neq 0$ in eq. B.9) and the zeros of the cross products of Bessel functions [111] would be required.

Appendix C. Calculation of Gradients

In electronic structure calculations, it can sometimes become necessary to compute the derivative of a quantity expressed using a chosen basis set, or over a grid. For instance, evaluation of the Hellmann-Feynman forces [112, 113] on the atoms of a system involves calculation of Cartesian gradients, if atomic pseudopotentials and pseudocharges are used to compute total energies [30, 40, 43]. In this section, we describe how such gradients may be computed for quantities expressed using helical waves.

Let $E(\theta_1, \theta_2, r)$ be a function expressed in helical coordinates over the fundamental domain, and let its expansion using helical waves be:

$$\begin{aligned}E\left(\theta_1,\theta_2,r\right) &= \sum_{\Gamma}\hat{E}_{m,n,k}\,F_{m,n,k}(\theta_1,\theta_2,r)\\ &= \sum_{\Gamma}\hat{E}_{m,n,k}\,c_{m,n,k}\,e^{i2\pi(m\theta_1+n\mathfrak{N}\theta_2)}J_{n\mathfrak{N}}\left(\frac{b_k^{n\mathfrak{N}}r}{R}\right)\,.\end{aligned} \tag{C.1}$$

The Cartesian gradient of this quantity,

$$\nabla E = \frac{\partial E}{\partial x}\mathbf{e}_{\mathrm{X}} + \frac{\partial E}{\partial y}\mathbf{e}_{\mathrm{Y}} + \frac{\partial E}{\partial z}\mathbf{e}_{\mathrm{Z}}\,, \tag{C.2}$$

may be evaluated by using the chain rule, i.e.,:

$$\frac{\partial E}{\partial x} = \frac{\partial E}{\partial \theta_1}\frac{\partial \theta_1}{\partial x} + \frac{\partial E}{\partial \theta_2}\frac{\partial \theta_2}{\partial x} + \frac{\partial E}{\partial r}\frac{\partial r}{\partial x} = -\frac{\sin\left(2\pi\left(\alpha\theta_1 + \theta_2\right)\right)}{2\pi r}E_{\theta_2} + \cos\left(2\pi\left(\alpha\theta_1 + \theta_2\right)\right)E_r\,,$$

$$\frac{\partial E}{\partial y} = \frac{\partial E}{\partial \theta_1}\frac{\partial \theta_1}{\partial y} + \frac{\partial E}{\partial \theta_2}\frac{\partial \theta_2}{\partial y} + \frac{\partial E}{\partial r}\frac{\partial r}{\partial y} = \frac{\cos\left(2\pi\left(\alpha\theta_1 + \theta_2\right)\right)}{2\pi r}E_{\theta_2} + \sin\left(2\pi\left(\alpha\theta_1 + \theta_2\right)\right)E_r\,,$$

$$\frac{\partial E}{\partial z} = \frac{\partial E}{\partial \theta_1}\frac{\partial \theta_1}{\partial z} + \frac{\partial E}{\partial \theta_2}\frac{\partial \theta_2}{\partial z} + \frac{\partial E}{\partial r}\frac{\partial r}{\partial z} = \frac{1}{\tau}\left(E_{\theta_1} - \alpha E_{\theta_2}\right). \tag{C.3}$$

Based on eq. C.1, we note immediately that:

$$E_{\theta_1}(\theta_1, \theta_2, r) = \sum_\Gamma \hat{E}_{m,n,k}\left(i2\pi m\right) F_{m,n,k}(\theta_1, \theta_2, r)\,,$$

$$E_{\theta_2}(\theta_1, \theta_2, r) = \sum_\Gamma \hat{E}_{m,n,k}\left(i2\pi n\mathfrak{N}\right) F_{m,n,k}(\theta_1, \theta_2, r)\,. \tag{C.4}$$

These expressions correspond to the inverse basis transforms of vectors with entries $\{(i2\pi m)\,\hat{E}_{m,n,k}\}_{(m,n,k)\in\Gamma}$ and $\{(i2\pi n\mathfrak{N})\,\hat{E}_{m,n,k}\}_{(m,n,k)\in\Gamma}$ respectively, and so they may be readily computed. To calculate the radial derivative E_r, we first note the following identity [67]:

$$\frac{\partial J_\kappa\left(q\right)}{\partial q} = J_{\kappa-1}\left(q\right) - \frac{\kappa}{q}J_\kappa\left(q\right)\,. \tag{C.5}$$

This expression may be used for computing the radial derivative of all helical waves within the basis set. However, as n varies from $-N_{max}$ to N_{max}, the order of the Bessel functions involved range from $\kappa = -\mathfrak{N}N_{max}$ to $\kappa = \mathfrak{N}N_{max}$, and the above expression results in a Bessel function that lies beyond the range of the basis set. To remedy this, we may use the following alternate expression [67] for the $\kappa = -\mathfrak{N}N_{max}$ case:

$$\frac{\partial J_\kappa\left(q\right)}{\partial q} = \frac{\kappa}{q}J_\kappa\left(q\right) - J_{\kappa+1}\left(q\right)\,. \tag{C.6}$$

Combining eqs. C.5 and C.6 with eq. C.1, we se that the radial derivative E_r may be expressed as:

$$E_r(\theta_1, \theta_2, r) = \sum_\Gamma \hat{E}_{m,n,k}\,c_{m,n,k}\,e^{i2\pi(m\theta_1 + n\mathfrak{N}\theta_2)}\,\mathcal{B}_n(r)\,, \tag{C.7}$$

where the radial functions $\mathcal{B}_n(r)$ are:

$$\mathcal{B}_n(r) = \frac{b_k^{n\mathfrak{N}}}{R}\left[J_{n\mathfrak{N}-1}\left(\frac{b_k^{n\mathfrak{N}}r}{R}\right) - \frac{n\mathfrak{N}R}{b_k^{n\mathfrak{N}}r}J_{n\mathfrak{N}}\left(\frac{b_k^{n\mathfrak{N}}r}{R}\right)\right]\,,\ \text{for } n \neq N_{\max}\,,$$

$$= \frac{b_k^{n\mathfrak{N}}}{R}\left[\frac{n\mathfrak{N}R}{b_k^{n\mathfrak{N}}r}J_{n\mathfrak{N}}\left(\frac{b_k^{n\mathfrak{N}}r}{R}\right) - J_{n\mathfrak{N}+1}\left(\frac{b_k^{n\mathfrak{N}}r}{R}\right)\right]\,,\ \text{for } n = N_{\max}\,. \tag{C.8}$$

With this, the radial derivative may be considered an inverse basis transform of the vector with entries $\{\hat{E}_{m,n,k}\}_{(m,n,k)\in\Gamma}$, provided we use the functions $\mathcal{B}_n(r)$ along the radial direction. These functions may be computed ahead of time and stored, and Algorithm 1 (Section 3.4.1) may then be used for computing $E_r(\theta_1, \theta_2, r)$. With the derivatives

$E_r(\theta_1,\theta_2,r)$, $E_{\theta_1}(\theta_1,\theta_2,r)$ and $E_{\theta_2}(\theta_1,\theta_2,r)$ available on hand as values on a real space grid, we may use eq. C.3 to evaluate the Cartesian derivatives at each point on the same grid.

Instead of obtaining the derivatives as real space quantities as described above, it is also possible to directly obtain them in reciprocal space. The expansion coefficients of E_{θ_1} and E_{θ_2} are immediately seen to be:

$$(\widehat{E_{\theta_1}})_{m',n',k'} = (i2\pi m')\,\hat{E}_{m',n',k'}\,,$$
$$(\widehat{E_{\theta_2}})_{m',n',k'} = (i2\pi n'\mathfrak{N}')\,\hat{E}_{m',n',k'}\,, \qquad (C.9)$$

and they may be evaluated at a cost that is proportional to the basis set size. By considering the inner product of eq. C.7 with the basis functions, the expansion coefficients of the radial derivative may be expressed as :

$$(\widehat{E_r})_{m',n',k'} = \langle E_r,\ F_{m',n',k'}\rangle_{\mathsf{L}^2(\mathcal{D})} = \frac{2}{R}\sum_\Gamma \hat{E}_{m',n',k}\,\mathcal{A}(n',k,k')\,. \qquad (C.10)$$

The numbers $\mathcal{A}(n',k,k')$ can be expressed in terms of oscillatory integrals:

$$\mathcal{A}\left(n',k,k'\right) = \frac{1}{J_{n'\mathfrak{N}+1}\left(b_k^{n'\mathfrak{N}}\right) J_{n'\mathfrak{N}+1}\left(b_{k'}^{n'\mathfrak{N}}\right)}\left[b_k^{n'\mathfrak{N}}\int_0^1 J_{n'\mathfrak{N}-1}\left(b_k^{n'\mathfrak{N}}q\right)\right.$$
$$\left.\times\, J_{n'\mathfrak{N}}\left(b_{k'}^{n'\mathfrak{N}}q\right)q\,dq - n'\mathfrak{N}\int_0^1 \frac{J_{n'\mathfrak{N}}\left(b_k^{n'\mathfrak{N}}q\right)J_{n'\mathfrak{N}}\left(b_{k'}^{n'\mathfrak{N}}q\right)}{q}q\,dq\right],\ \text{for } n'\neq -\mathfrak{N}N_{max} \quad (C.11)$$
$$= \frac{1}{J_{n'\mathfrak{N}+1}\left(b_k^{n'\mathfrak{N}}\right) J_{n'\mathfrak{N}+1}\left(b_{k'}^{n'\mathfrak{N}}\right)}\left[n'\mathfrak{N}\int_0^1 \frac{J_{n'\mathfrak{N}}\left(b_k^{n'\mathfrak{N}}q\right)J_{n'\mathfrak{N}}\left(b_{k'}^{n'\mathfrak{N}}q\right)}{q}q\,dq\right.$$
$$\left.-\, b_k^{n'\mathfrak{N}}\int_0^1 J_{n'\mathfrak{N}+1}\left(b_k^{n'\mathfrak{N}}q\right)J_{n'\mathfrak{N}}\left(b_{k'}^{n'\mathfrak{N}}q\right)q\,dq\right],\ \text{for } n'=-\mathfrak{N}N_{max}\,. \quad (C.12)$$

We may precompute them using the techniques described in Appendix D and store them for use later. Note that such an expansion of the radial derivatives using the basis set implicitly requires these quantities to obey Dirichlet boundary conditions at $r=R$. However, this may not be satisfied in general. The real space expression outlined earlier (eqs. C.7,C.8) skirts this issue.

Appendix D. Evaluation of Oscillatory Radial Integrals

This work has multiple instances in which integrals with oscillatory integrands along the radial direction make an appearance (e.g., eq. 31 and eqs. C.11,C.12). A typical scenario is depicted in Fig. D.18. Techniques for the evaluation of such integrals have been extensively studied in the literature [114–117] and specialized methods for integrands involving Bessel functions are also available [118]. Instead of adopting these more elaborate methods, we choose to evaluate the oscillatory integrals in this work by using the simpler procedure of employing a large number of Gauss-Jacobi quadrature [119] nodes and weights. Thus, denoting $q=r/R$, we write:

$$\int_0^1 f(q)\,q^\sigma\,dq = \sum_{i=1}^{N_q} f(q_i)w_i\,. \qquad (D.1)$$

The values of the weights w_i and the nodes q_i dependent on the the exponent σ, as well as the quadrature order N_q. The weights and nodes can be computed inexpensively [120] even when N_q is of the order of a few thousand.
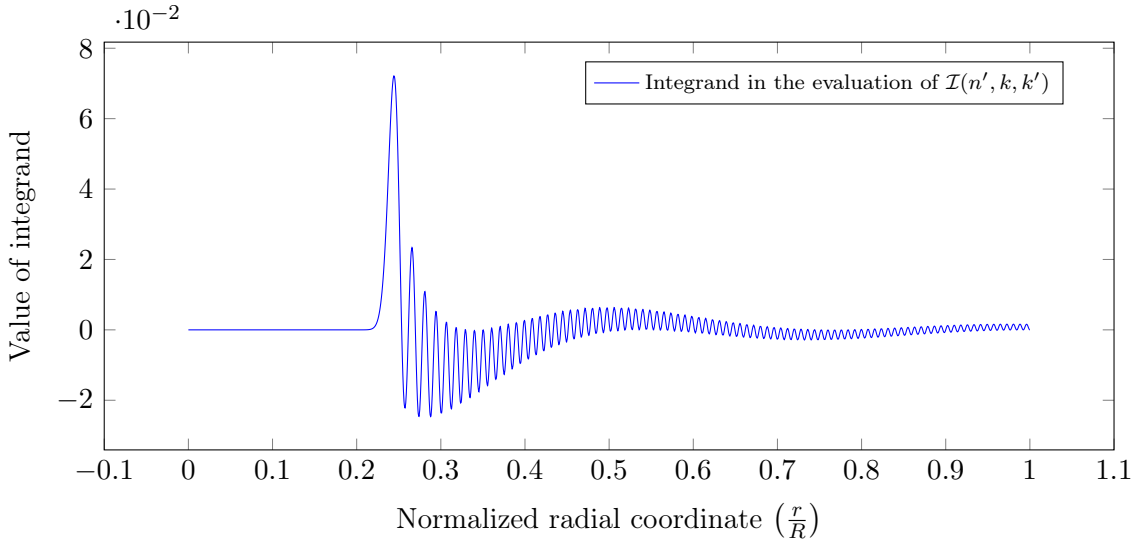

Figure D.18: The highly oscillatory behavior of the integrand involved in the evaluation of the quantity $\mathcal{I}(n', k, k')$ in eq. 31. The behavior increases as the values of n, k, k' increase. For the above plot, we chose $n = -98, k = 89, k' = 85$

For the case of the integrals involved in the evaluation of $\mathcal{I}(n', k, k')$ via eq. 31, the number of oscillations of the integrand is approximately equal to $k + k'$. Thus, within a given basis set, the maximum number of oscillations is $2K_{\max}$. For all the examples considered in this work, $K_{\max}$ does not generally exceed 200, and we have found that choosing N_q to be a few thousand for such cases allows the integrals to be converged to $\mathcal{O}(10^{-14})$. To verify our calculations, we have also used Gauss-Kronrod quadrature [121, 122] as employed within Matlab (`quadgk` function). This allows for automatic adaptive placement of the integration nodes and monitoring of the quadrature error, and we verified that the latter was always $\mathcal{O}(10^{-13})$ or lower, even for the cases involving the most oscillatory integrands.

Acknowledgements

—

References

[1] B. Bhushan, B. Bhushan, Baumann, Springer handbook of nanotechnology, Vol. 2, Springer, 2007.

[2] G. Cao, Nanostructures & nanomaterials: synthesis, properties & applications, Imperial college press, 2004.

[3] I. Kang, M. J. Schulz, J. H. Kim, V. Shanov, D. Shi, A carbon nanotube strain sensor for structural health monitoring, Smart materials and structures 15 (3) (2006) 737.

[4] S. Chopra, K. McGuire, N. Gothard, A. Rao, A. Pham, Selective gas detection using a carbon nanotube sensor, Applied Physics Letters 83 (11) (2003) 2280–2282.

[5] R. H. Baughman, C. Cui, A. A. Zakhidov, Z. Iqbal, J. N. Barisci, G. M. Spinks, G. G. Wallace, A. Mazzoldi, D. De Rossi, A. G. Rinzler, et al., Carbon nanotube actuators, Science 284 (5418) (1999) 1340–1344.

[6] L. Kong, W. Chen, Carbon nanotube and graphene-based bioinspired electrochemical actuators, Advanced materials 26 (7) (2014) 1025–1043.

[7] H. Sun, Y. Zhang, J. Zhang, X. Sun, H. Peng, Energy harvesting and storage in 1d devices, Nature Reviews Materials 2 (6) (2017) 1–12.

[8] F. R. Fan, W. Tang, Z. L. Wang, Flexible nanogenerators for energy harvesting and self-powered electronics, Advanced Materials 28 (22) (2016) 4283–4305.

[9] C. Xu, Chiral Nanoprobes for Biological Applications, John Wiley & Sons, 2022.

[10] R. D. James, Objective structures, Journal of the Mechanics and Physics of Solids 54 (11) (2006) 2354–2390.

[11] T. Giamarchi, Quantum physics in one dimension, Vol. 121, Clarendon press, 2003.

[12] M. Bockrath, D. H. Cobden, J. Lu, A. G. Rinzler, R. E. Smalley, L. Balents, P. L. McEuen, Luttinger-liquid behaviour in carbon nanotubes, Nature 397 (6720) (1999) 598–601.

[13] R. Egger, A. Zazunov, A. L. Yeyati, Helical luttinger liquid in topological insulator nanowires, Physical review letters 105 (13) (2010) 136403.

[14] S. Zaitsev-Zotov, Y. A. Kumzerov, Y. A. Firsov, P. Monceau, Luttinger-liquid-like transport in long insb nanowires, Journal of Physics: Condensed Matter 12 (20) (2000) L303.

[15] K. Arutyunov, D. Golubev, A. Zaikin, Superconductivity in one dimension, Physics Reports 464 (1-2) (2008) 1–70.

[16] F. Qin, W. Shi, T. Ideue, M. Yoshida, A. Zak, R. Tenne, T. Kikitsu, D. Inoue, D. Hashizume, Y. Iwasa, Superconductivity in a chiral nanotube, Nature communications 8 (1) (2017) 1–6.

[17] L. Krusin-Elbaum, D. Newns, H. Zeng, V. Derycke, J. Sun, R. Sandstrom, Room-temperature ferromagnetic nanotubes controlled by electron or hole doping, Nature 431 (7009) (2004) 672–676.

[18] M. Liu, J. Wang, Giant electrocaloric effect in ferroelectric nanotubes near room temperature, Scientific reports 5 (1) (2015) 1–7.

[19] D. Zhang, M. Hua, T. Dumitrica, Stability of polycrystalline and wurtzite si nanowires via symmetry-adapted tight-binding objective molecular dynamics, The Journal of Chemical Physics 128 (8) (2008) 084104.

[20] H. M. Yu, A. S. Banerjee, Density functional theory method for twisted geometries with application to torsional deformations in group-iv nanotubes, Journal of Computational Physics 456 (2022) 111023.

[21] J. Ding, X. Yan, J. Cao, Analytical relation of band gaps to both chirality and diameter of single-wall carbon nanotubes, Physical Review B 66 (7) (2002) 073401.

[22] S. J. Tans, A. R. Verschueren, C. Dekker, Room-temperature transistor based on a single carbon nanotube, Nature 393 (6680) (1998) 49–52.

[23] S. Li, Z. Yu, S.-F. Yen, W. Tang, P. J. Burke, Carbon nanotube transistor operation at 2.6 ghz, Nano Letters 4 (4) (2004) 753–756.

[24] C. D. Aiello, J. M. Abendroth, M. Abbas, A. Afanasev, S. Agarwal, A. S. Banerjee, D. N. Beratan, J. N. Belling, B. Berche, A. Botana, J. R. Caram, G. L. Celardo, G. Cuniberti, A. Garcia-Etxarri, A. Dianat, I. Diez-Perez, Y. Guo, R. Gutierrez, C. Herrmann, J. Hihath, S. Kale, P. Kurian, Y.-C. Lai, T. Liu, A. Lopez, E. Medina, V. Mujica, R. Naaman, M. Noormandipour, J. L. Palma, Y. Paltiel, W. Petuskey, J. C. Ribeiro-Silva, J. J. Saenz, E. J. G. Santos, M. Solyanik-Gorgone, V. J. Sorger, D. M. Stemer, J. M. Ugalde, A. Valdes-Curiel, S. Varela, D. H. Waldeck, M. R.

Wasielewski, P. S. Weiss, H. Zacharias, Q. H. Wang, A chirality-based quantum leap, ACS Nano 0 (0) (2022).

[25] R. Naaman, D. H. Waldeck, Spintronics and chirality: Spin selectivity in electron transport through chiral molecules, Annual Review of Physical Chemistry 66 (1) (2015) 263–281.

[26] H. Zhang, D. E. Liu, M. Wimmer, L. P. Kouwenhoven, Next steps of quantum transport in majorana nanowire devices, Nature communications 10 (1) (2019) 1–7.

[27] R. M. Martin, Electronic Structure: Basic Theory and Practical Methods, 1st Edition, Cambridge University Press, 2004.

[28] D. Marx, J. Hutter, Ab initio molecular dynamics: basic theory and advanced methods, 1st Edition, Cambridge University Press, 2009.

[29] A. S. Banerjee, Density functional methods for Objective Structures: Theory and simulation schemes, Ph.D. thesis, University of Minnesota, Minneapolis (2013).

[30] A. S. Banerjee, Ab initio framework for systems with helical symmetry: theory, numerical implementation and applications to torsional deformations in nanostructures, Journal of the Mechanics and Physics of Solids (2021) 104515.

[31] P. D'Arco, Y. Noel, R. Demichelis, R. Dovesi, Single-layered chrysotile nanotubes: A quantum mechanical ab initio simulation, The Journal of chemical physics 131 (20) (2009) 204701.

[32] R. Dovesi, V. Saunders, C. Roetti, R. Orlando, C. Zicovich-Wilson, F. Pascale, B. Civalleri, K. Doll, N. Harrison, I. Bush, et al., Crystal17 (2017).

[33] C. T. White, D. H. Robertson, J. W. Mintmire, Helical and rotational symmetries of nanoscale graphitic tubules, Physical Review B 47 (1993) 5485–5488.

[34] E. Chang, G. Bussi, A. Ruini, E. Molinari, First-principles approach for the calculation of optical properties of one-dimensional systems with helical symmetry: The case of carbon nanotubes, Physical Review B 72 (2005) 195423.

[35] Y. Noel, P. D'arco, R. Demichelis, C. M. Zicovich-Wilson, R. Dovesi, On the use of symmetry in the ab initio quantum mechanical simulation of nanotubes and related materials, Journal of Computational Chemistry 31 (4) (2010) 855–862.

[36] E. Chang, G. Bussi, A. Ruini, E. Molinari, Excitons in carbon nanotubes: An *Ab Initio* symmetry-based approach, Physical Review Letters 92 (2004) 196401.

[37] R. M. Balabin, Communications: Intramolecular basis set superposition error as a measure of basis set incompleteness: Can one reach the basis set limit without extrapolation?, The Journal of chemical physics 132 (21) (2010) 211103.

[38] M. Gutowski, G. Chal/asiński, Critical evaluation of some computational approaches to the problem of basis set superposition error, The Journal of chemical physics 98 (7) (1993) 5540–5554.

[39] S. Simon, M. Duran, J. Dannenberg, How does basis set superposition error change the potential surfaces for hydrogen-bonded dimers?, The Journal of chemical physics 105 (24) (1996) 11024–11031.

[40] S. Ghosh, A. S. Banerjee, P. Suryanarayana, Symmetry-adapted real-space density functional theory for cylindrical geometries: Application to large group-IV nanotubes, Physical Review B 100 (12) (2019) 125143.

[41] A. Sharma, P. Suryanarayana, Real-space density functional theory adapted to cyclic and helical symmetry: Application to torsional deformation of carbon nanotubes, Physical Review B 103 (3) (2021) 035101.

[42] F. Gygi, G. Galli, Real-space adaptive-coordinate electronic-structure calculations, Physical Review B 52 (4) (1995) R2229.

[43] A. S. Banerjee, P. Suryanarayana, Cyclic density functional theory: A route to the first principles simulation of bending in nanostructures, Journal of the Mechanics and Physics of Solids 96 (2016) 605–631.

[44] Y. Saad, Numerical Methods for Large Eigenvalue Problems, Revised Edition, SIAM, 2011.

[45] A. S. Banerjee, R. S. Elliott, R. D. James, A spectral scheme for kohn–sham density functional theory of clusters, Journal of Computational Physics 287 (2015) 226–253.

[46] MATLAB, 9.7.0.1190202 (R2019b), the MathWorks Inc., Natick, Massachusetts (2019).

[47] R. Broglia, G. Coló, G. Onida, H. Roman, Solid State Physiscs of Finite Systems: Metallic Clusters, Fullerenes, Atomic Wires, 1st Edition, Advanced Texts in Physics, Springer, 2004.

[48] P. N. D'yachkov, D. Makaev, Linear augmented cylindrical wave method and its applications to nanotubes electronic structure, Journal of Nanophotonics 4 (1) (2010) 041680.

[49] S. R. Hussain, S. Kharitonov, V. Pavelyev, Calculation of the band structure of a non-chiral semiconductor and metallic carbon nanotubes, in: Journal of Physics: Conference Series, Vol. 1096, IOP Publishing, 2018, p. 012109.

[50] P. D'Yachkov, Cylindrical wave method for pure and doped nanotubes, in: Nanodevices and Nanomaterials for Ecological Security, Springer, 2012, pp. 87–100.

[51] D. Makaev, P. D'yachkov, Linearized augmented cylindrical wave method for chiral nanotubes, in: Doklady Physical Chemistry, Vol. 419, Springer, 2008, pp. 47–52.

[52] D. Jüstel, G. Friesecke, R. D. James, Bragg–von laue diffraction generalized to twisted x-rays, Acta Crystallographica Section A: Foundations and Advances 72 (2) (2016) 190–196.

[53] G. Friesecke, R. D. James, D. Jüstel, Twisted x-rays: incoming waveforms yielding discrete diffraction patterns for helical structures, SIAM Journal on Applied Mathematics 76 (3) (2016) 1191–1218.

[54] G. Chaplain, J. De Ponti, R. Craster, Elastic orbital angular momentum, Physical Review Letters 128 (6) (2022) 064301.

[55] G. J. Chaplain, J. M. De Ponti, The elastic spiral phase pipe, Journal of Sound and Vibration 523 (2022) 116718.

[56] J. P. Walter, M. L. Cohen, Wave-vector-dependent dielectric function for si, ge, gaas, and znse, Physical Review B 2 (6) (1970) 1821.

[57] C. Fong, M. L. Cohen, Energy band structure of copper by the empirical pseudopotential method, Physical Review Letters 24 (7) (1970) 306.

[58] M. Hoffmann-Ostenhof, T. Hoffmann-Ostenhof, R. Ahlrichs, J. Morgan, On the exponential fall off of wavefunctions and electron densities, in: K. Osterwalder (Ed.), Mathematical Problems in Theoretical Physics, Vol. 116 of Lecture Notes in Physics, Springer Berlin / Heidelberg, 1980, pp. 62–67.

[59] R. Ahlrichs, M. Hoffmann-Ostenhof, T. Hoffmann-Ostenhof, J. D. Morgan, Bounds on the decay of electron densities with screening, Physical Review A 23 (5) (1981) 2106–2117.

[60] R. Waldron, A helical coordinate system and its applications in electromagnetic theory, The Quarterly Journal of Mechanics and Applied Mathematics 11 (4) (1958) 438–461.

[61] D. Hochberg, G. Edwards, T. W. Kephart, Representing structural information of helical charge distributions in cylindrical coordinates, Physical Review E 55 (3) (1997) 3765.

[62] Y. Saad, J. R. Chelikowsky, S. M. Shontz, Numerical methods for electronic structure calculations of materials, SIAM Review 52 (1) (2010) 3–54.

[63] S. A. Orszag, Transform method for the calculation of vector-coupled sums: Application to the spectral form of the vorticity equation, Journal of the Atmospheric Sciences 27 (6) (1970) 890–895.

[64] S. A. Orszag, Spectral methods for problems in complex geometries, Journal of Computational Physics 37 (1) (1980) 70–92.

[65] J. Shen, T. Tang, L.-L. Wang, Spectral methods: algorithms, analysis and applications, Springer Series in Computational Mathematics, Springer Berlin, New York, 2011.

[66] D. Gottlieb, S. Orszag, Numerical Analysis of Spectral Methods: Theory and Applications, SIAM-CBMS, Philadelphia, 1977.

[67] M. Abramowitz, I. A. Stegun, R. H. Romer, Handbook of mathematical functions with formulas, graphs, and mathematical tables (1988).

[68] P. P. Teodorescu, N.-D. Stanescu, N. Pandrea, Numerical Analysis with Applications in Mechanics and Engineering, 1st Edition, John Wiley & Sons, 2013.

[69] E. Bylaska, K. Tsemekhman, N. Govind, M. Valiev, Large-scale plane-wave-based density functional theory: Formalism, parallelization, and applications, Computational methods for large systems: electronic structure approaches for biotechnology and nanotechnology (2011) 77–116.

[70] E. Cancès, R. Chakir, Y. Maday, Numerical analysis of the planewave discretization of some orbital-free and Kohn-Sham models, ESAIM: Mathematical Modelling and Numerical Analysis 46 (2012) 341–388.

[71] H. J. Monkhorst, J. D. Pack, Special points for brillouin-zone integrations, Physical review B 13 (12) (1976) 5188.

[72] I. I. Geru, Time-Reversal Symmetry, Springer, 2018.

[73] D. Lemoine, The discrete bessel transform algorithm, The Journal of chemical physics 101 (5) (1994) 3936–3944.

[74] H. Johansen, K. Sørensen, Fast hankel transforms, Geophysical Prospecting 27 (4) (1979) 876–901.

[75] R. Bisseling, R. Kosloff, The fast hankel transform as a tool in the solution of the time dependent schrödinger equation, Journal of Computational Physics 59 (1) (1985) 136–151.

[76] K. Key, Is the fast hankel transform faster than quadrature?, Geophysics 77 (3) (2012) F21–F30.

[77] L. S. Blackford, A. Petitet, R. Pozo, K. Remington, R. C. Whaley, J. Demmel, J. Dongarra, I. Duff, S. Hammarling, G. Henry, et al., An updated set of basic linear algebra subprograms (blas), ACM Transactions on Mathematical Software 28 (2) (2002) 135–151.

[78] G. W. Stewart, A krylov–schur algorithm for large eigenproblems, SIAM Journal on Matrix Analysis and Applications 23 (3) (2002) 601–614.

[79] M. C. Payne, M. P. Teter, D. C. Allan, T. A. Arias, J. D. Joannopoulos, Iterative minimization techniques for *ab initio* total-energy calculations: molecular dynamics and conjugate gradients, Rev. Mod. Phys. 64 (1992) 1045–1097.

[80] Y. Zhou, J. R. Chelikowsky, X. Gao, A. Zhou, On the "preconditioning" function used in planewave dft calculations and its generalization, Communications in Computational Physics 18 (1) (2015) 167–179.

[81] A. Knyazev, Locally Optimal Block Preconditioned Conjugate Gradient, https://github.com/lobpcg/blopex, https://www.mathworks.com/matlabcentral/fileexchange/48-locally-optimal-block-preconditioned-conjugate-gradient, Accessed: 2022-07-21 (2022).

[82] A. Knyazev, Toward the optimal preconditioned eigensolver: Locally optimal block preconditioned conjugate gradient method, SIAM Journal on Scientific Computing 23 (2) (2001) 517–541.

[83] A. V. Knyazev, M. E. Argentati, I. Lashuk, E. Ovtchinnikov, Block locally optimal preconditioned eigenvalue xolvers (BLOPEX) in HYPRE and PETSc, SIAM Journal on Scientific Computing 29 (5) (2007) 2224–2239.

[84] J. A. Duersch, M. Shao, C. Yang, M. Gu, A robust and efficient implementation of lobpcg, SIAM Journal on Scientific Computing 40 (5) (2018) C655–C676.

[85] M. P. Teter, M. C. Payne, D. C. Allan, Solution of Schrödinger's equation for large systems, Physical Review B 40 (1989) 12255–12263.

[86] A. Mayer, Band structure and transport properties of carbon nanotubes using a local pseudopotential and a transfer-matrix technique, Carbon 42 (10) (2004) 2057–2066.

[87] A. S. Banerjee, L. Lin, W. Hu, C. Yang, J. E. Pask, Chebyshev polynomial filtered subspace iteration in the Discontinuous Galerkin method for large-scale electronic structure calculations, arXiv preprint arXiv:1606.03416 (2016).

[88] A. S. Banerjee, L. Lin, P. Suryanarayana, C. Yang, J. E. Pask, Two-level chebyshev filter based complementary subspace method: pushing the envelope of large-scale electronic structure calculations, Journal of chemical theory and computation 14 (6) (2018) 2930–2946.

[89] Y. Zhou, J. R. Chelikowsky, Y. Saad, Chebyshev-filtered subspace iteration method free of sparse diagonalization for solving the Kohn–Sham equation, Journal of Computational Physics 274 (2014) 770–782.

[90] A. A. Laturia, M. L. Van de Put, W. G. Vandenberghe, Generation of empirical pseudopotentials for transport applications and their application to group iv materials, Journal of Applied Physics 128 (3) (2020) 034306.

[91] M. L. Van de Put, M. V. Fischetti, W. G. Vandenberghe, Scalable atomistic simulations of quantum electron transport using empirical pseudopotentials, Computer Physics Communications 244 (2019) 156–169.

[92] J. R. Chelikowsky, Introductory Quantum Mechanics with MATLAB: For Atoms, Molecules, Clusters, and Nanocrystals, John Wiley & Sons, 2019.

[93] A. Rohatgi, Webplotdigitizer: Version 4.5 (2021).
URL https://automeris.io/WebPlotDigitizer

[94] A. Mayer, J.-P. Vigneron, Accuracy-control techniques applied to stable transfer-matrix computations, Physical Review E 59 (4) (1999) 4659.

[95] J. Pendry, A. MacKinnon, Calculation of photon dispersion relations, Physical Review Letters 69 (19) (1992) 2772.

[96] H. Tamura, T. Ando, Conductance fluctuations in quantum wires, Physical Review B 44 (4) (1991) 1792.

[97] M. L. Van de Put, A. A. Laturia, M. V. Fischetti, W. G. Vandenberghe, Efficient modeling of electron transport with plane waves, in: 2018 International Conference on Simulation of Semiconductor

Processes and Devices (SISPAD), 2018, pp. 71–74.

[98] N. Troullier, J. L. Martins, Efficient pseudopotentials for plane-wave calculations, Physical Review B 43 (3) (1991) 1993.

[99] W. Kohn, L. J. Sham, Self-consistent equations including exchange and correlation effects, Phys. Rev. 140 (1965) A1133–A1138.

[100] J. P. Perdew, Y. Wang, Accurate and simple analytic representation of the electron-gas correlation energy, Physical Review B 45 (1992) 13244–13249.

[101] D. Zhang, R. D. James, T. Dumitrica, Electromechanical characterization of carbon nanotubes in torsion via symmetry adapted tight-binding objective molecular dynamics, Physical Review B 80 (11) (2009) 115418.

[102] L. Yang, M. Anantram, J. Han, J. Lu, Band-gap change of carbon nanotubes: Effect of small uniaxial and torsional strain, Physical Review B 60 (19) (1999) 13874.

[103] L. Yang, J. Han, Electronic structure of deformed carbon nanotubes, Physical review letters 85 (1) (2000) 154.

[104] S. Agarwal, A. S. Banerjee, On the solution of the Poisson problem for twisted geometries, (In preparation) (2022).

[105] S. Pathrudkar, H. M. Yu, S. Ghosh, A. S. Banerjee, Machine learning based prediction of the electronic structure of quasi-one-dimensional materials under strain, Phys. Rev. B 105 (2022) 195141.

[106] Y. Hakobyan, E. Tadmor, R. James, Objective quasicontinuum approach for rod problems, Physical Review B 86 (24) (2012) 245435.

[107] C. Le Bris (Ed.), Computational Chemistry, Vol. X of Handbook of Numerical Analysis, North-Holland, 2003.

[108] I. Nikiforov, B. Hourahine, B. Aradi, T. Frauenheim, T. Dumitrică, Ewald summation on a helix: A route to self-consistent charge density-functional based tight-binding objective molecular dynamics, The Journal of Chemical Physics 139 (9) (2013) 094110.

[109] O. Liebman, S. Agarwal, A. Banerjee, Helical waves for self-consistent first principles calculations of chiral one-dimensional nanomaterials, Bulletin of the American Physical Society (2022).

[110] NIST Digital Library of Mathematical Functions, http://dlmf.nist.gov/, Release 1.1.5 of 2022-03-15, f. W. J. Olver, A. B. Olde Daalhuis, D. W. Lozier, B. I. Schneider, R. F. Boisvert, C. W. Clark, B. R. Miller, B. V. Saunders, H. S. Cohl, and M. A. McClain, eds. (2022).
URL http://dlmf.nist.gov/

[111] L. J. Laslett, W. Lewish, Evaluation of the zeros of cross-product bessel functions, Mathematics of Computation 16 (78) (1962) 226–232.

[112] H. Hellmann, Hans Hellmann: Einführung in Die Quantenchemie: Mit Biografischen Notizen von Hans Hellmann Jr., Springer-Verlag, 2015.

[113] R. P. Feynman, Forces in molecules, Physical Review 56 (4) (1939) 340.

[114] A. Deaño, D. Huybrechs, A. Iserles, Computing highly oscillatory integrals, SIAM, 2017.

[115] Y. Ma, Y. Xu, Computing highly oscillatory integrals, Mathematics of Computation 87 (309) (2018) 309–345.

[116] A. Iserles, S. P. Nørsett, Efficient quadrature of highly oscillatory integrals using derivatives, Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences 461 (2057) (2005) 1383–1399.

[117] A. Iserles, S. P. Nørsett, On quadrature methods for highly oscillatory integrals and their implementation, BIT Numerical Mathematics 44 (4) (2004) 755–772.

[118] R. Chen, Numerical approximations to integrals with a highly oscillatory bessel kernel, Applied Numerical Mathematics 62 (5) (2012) 636–648.

[119] A. Ralston, P. Rabinowitz, A first course in numerical analysis, Courier Corporation, 2001.

[120] N. Hale, A. Townsend, Fast and accurate computation of gauss–legendre and gauss–jacobi quadrature nodes and weights, SIAM Journal on Scientific Computing 35 (2) (2013) A652–A674.

[121] D. Laurie, Calculation of gauss-kronrod quadrature rules, Mathematics of Computation 66 (219) (1997) 1133–1145.

[122] L. F. Shampine, Vectorized adaptive quadrature in matlab, Journal of Computational and Applied Mathematics 211 (2) (2008) 131–140.